International Journal of Approximate Reasoning 151 (2022) 101–129

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

A Survey on Causal Discovery: Theory and Practice

Alessio Zanga [a,*], Elif Ozkirimli [b], Fabio Stella [a,c]

[a] Department of Informatics, Systems and Communication, University of Milano-Bicocca, Viale Sarca, 336, 20126 Milano, Italy
[b] Data and Analytics Chapter, F. Hoffmann - La Roche Ltd, Basel, Switzerland
[c] Bicocca Bioinformatics, Biostatistics and Bioimaging Centre (B4), Milano, Italy

ARTICLE INFO

Article history:
Received 16 March 2022
Received in revised form 6 September 2022
Accepted 7 September 2022
Available online 13 September 2022

Keywords:
Causality
Causal models
Causal discovery
Structural learning

ABSTRACT

Understanding the laws that govern a phenomenon is the core of scientific progress. This is especially true when the goal is to model the interplay between different aspects in a causal fashion. Indeed, causal inference itself is specifically designed to quantify the underlying relationships that connect a cause to its effect. Causal discovery is a branch of the broader field of causality in which causal graphs are recovered from data (whenever possible), enabling the identification and estimation of causal effects. In this paper, we explore recent advancements in causal discovery in a unified manner, provide a consistent overview of existing algorithms developed under different settings, report useful tools and data, present real-world applications to understand why and how these methods can be fruitfully exploited.

1. Introduction

1.1. A general overview

One of the mantras that is repeated in every statistical course is that *correlation does not imply causation*. This is also observed in several disciplines, such as economics [1], biology [2], computer science [3,4] and philosophy [5]. Following [6], the main goal of a research study is often to assess the effect, if any, of an action on some outcome and not measuring a mere correlation. For example, this is true when it comes to decision making, since deciding which intervention must be taken is not straightforward and must be addressed properly to avoid any potential side effects. In order to identify and quantify a causal effect, the set of tools provided by causal discovery should be used accordingly. Here, the final goal is to decompose the total effect of an action into the causal and non-causal effects.

Causal inference, i.e. the task of quantifying the impact of a cause on its effect, relies heavily on a formal description on the interactions between the observed variables, i.e. a casual graph. Such graphical representation is naïve in its concept, yet so effective when it comes to *explainability*. Following [7], it boils down to connect a cause to an effect (outcome) by drawing arrows from the former to the latter, to obtain a qualitative description of the system under study. This is in stark contrast with black-box techniques, where predictions about an outcome are made with a pure data-driven approach. Indeed, these methods fall short both in terms of explainability and decision making, as stated in [8,9,6]. Therefore, when causality is empowered through the instrument of graphical models, it is possible to overcome the current limitations of machine learning and deep learning tools, enabling the researcher to reach a higher level of understanding.

* Corresponding author.
E-mail addresses: alessio.zanga@unimib.it (A. Zanga), elif.ozkirimli@roche.com (E. Ozkirimli), fabio.stella@unimib.it (F. Stella).

https://doi.org/10.1016/j.ijar.2022.09.004
0888-613X/

Table 1
Comparison of recent surveys on causal discovery in terms of covered contents.

Related Works	Theoretical Definitions	Evaluation Datasets	Evaluation Metrics	Software Packages	Practical Applications
A Survey of Learning Causality with Data: Problems and Methods [11]	✓		✓		✓
Causal Inference for Time Series Analysis: Problems, Methods and Evaluation [12]	✓	✓	✓		
Review of Causal Discovery Methods based on Graphical Models [5]	✓				✓
Causal Discovery Algorithms: A Practical Guide [13]	✓		✓	✓	
D'ya like DAGs? A survey on structure learning and causal discovery [14]	✓	✓	✓	✓	
Causal Discovery in Machine Learning: Theories and Applications [15]	✓		✓	✓	
Toward Causal Representation Learning [16]	✓				

When the causal graph is unknown, one may recover the cause-effect pairs by combining available data together with prior knowledge, whenever possible. The process of learning graphical structures with a causal interpretation is known as *causal discovery*. Recently, causal discovery has gained significant traction, especially when experimental data are available. However, this growth fragmented the landscape into multiple fields that differ for assumptions, problems and solutions, while aiming to the same goal. For this reason, this work summarizes the current status of causal discovery from both a theoretical and practical point of view, unifying shared concepts and addressing differences in the algorithms made available by the specialized scientific literature.

This survey is structured as follows. In Section 1, the reader is provided with a general introduction to the causal discovery problem, along with an overview of previous works on the same topic. Section 2 is devoted to provide concepts, definitions and problems that are common across different approaches presented in the following pages. Section 3 explores the first set of algorithms in the observational setting, while Section 4 relaxes the acyclicity assumption. In Section 5, the scope is extended to cover the experimental scenario, where multiple interactions with the system of interest are taken into account. Sections 6 and 7 report respectively on evaluation techniques and on practical applications of the discussed methodologies. Finally, Section 8 draws conclusions about the current landscape of causal discovery.

1.2. Related works

To the best of our knowledge, seven different surveys on causal discovery were published from 2019 to 2022. In particular, [10] acted as a *meta-survey* by checking the contents covered by the others concerning five topics, namely: theory, data, software, metrics and examples. A modified version of this checklist can be found in Table 1, which was adapted for a direct comparison with the structure of our survey.

While every contribution provided adequate background knowledge and theoretical definitions involving the fundamental aspects of causal discovery, only a few examples [5,12,15] reported evaluation datasets or metrics, and just two of them listed both [12,10]. The landscape is even more fragmented when observed from a practical point of view: only two contributions [15,10] presented and discussed the availability of software tools to perform the described procedures, thus hindering the applicability of causal discovery to researchers approaching this topic for the first time.

In particular, the contributions from [11,14] provide insights on the discovery procedure using machine learning, deep learning and reinforcement learning approaches. Authors in [12] tackled the problem of recovering the causal graph from time-series datasets, while [5] restricted its attention to the most famous techniques. Moreover, [15] presented a general survey on the topic without a proper *interventional* section, as for [16] in the latent case. Researchers in [13] focused on a high-level overview of the methods to provide a general guide for practical applications. Finally, if the reader is interested in gaining a high-level perspective that investigates the interplay between causal inference and causal discovery, the content of [10] has to be preferred, opposed to our in-depth approach exclusively focused on causal discovery.

This survey is designed in the light of the above considerations and aims to guide the inexpert reader through the forest of causal graphs to avoid common pitfalls when comparing and assessing the quality of results obtained by different causal discovery algorithms. It is worthwhile to mention that this survey is different from those published from 2019 to 2022 with respect to both theory and practice. Indeed, existing surveys introduce theoretical aspects of causal discovery while only a few go into additional details. Another lack of existing surveys in term of theory is that very few of them discuss the difference between observational and interventional data. This survey has also many differences in terms of practice: *i*) we provide a description on evaluation datasets and metrics, *ii*) we discuss how to tune strategies for choosing values of algorithm's hyperparameters, *iii*) we report on software packages, and *iv*) we discuss practical applications of causal discovery methods.

2. Definitions and notation

This section gives the main definitions, concepts and assumptions on causality, together with the associated notation. In particular, we give the definition of causal model together with the definition of causal discovery problem.

2.1. Notation

We denote mathematical objects with capital letters, such as random variable X, and collections of objects with capital boldface letters, such as set $\mathbf{X}$.

Definition 2.1 *(Graph).* A graph $G = (\mathbf{V}, \mathbf{E})$ is a mathematical object represented by a tuple of two sets: a finite set of vertices $\mathbf{V}$ and a finite set of edges $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$. If not specified otherwise, this graph is intended as an *undirected graph*, where the *undirected edge* (X, Y) is identical to the edge (Y, X) and its graphical representation is $X - Y$.

Definition 2.2 *(Directed graph).* A directed graph (DG) G is a graph where the edge (X, Y) is distinct from the edge (Y, X).

In particular, a directed edge (X, Y) is graphically represented by an arrow as $X \to Y$, and induces a set of relationships between the vertices of the graph G. Given a vertex X, we denote by $Pa(X)$ its *parents*, i.e. the set of vertices that have an arrow into X, while we denote by $Ch(X)$ its *children*, i.e. the set of vertices that have an arrow out of X. Recursively, any parent and parent of a parent (child and child of a child) of X is an *ancestor* $An(X)$ (*descendant* $De(X)$) of X.

The vertices connected to X are said to be *adjacent* to X and denoted by $Adj(X)$, while the vertices connected with an undirected edge are the *neighbors* $Ne(X)$. These two sets of vertices are identical in undirected graphs, but may be different in graphs with other mixed orientations.

Definition 2.3 *(Path).* A path $\pi = (X - \cdots - Y)$ is a tuple of non repeating vertices, where each vertex is connected to the next in the sequence with an undirected edge.

Definition 2.4 *(Directed path).* A directed path $\pi = (X \to \cdots \to Y)$ is a tuple of non repeating vertices, where each vertex is connected to the next in the sequence with a directed edge.

Definition 2.5 *(Cycle).* A cycle is a path that starts and ends at the same vertex.

Definition 2.6 *(Directed acyclic graph).* A directed acyclic graph (DAG) is a directed graph G that has no cycles.

2.2. Causal model

Definition 2.7 *(Causal graph).* A *causal graph* $G = (\mathbf{V}, \mathbf{E})$ [8] is a graphical description of a system in terms of cause-effect relationships, i.e. the *causal mechanism*.

For instance, the key difference between a Bayesian network (BN) [17] and a causal Bayesian network (CBN) is the semantic interpretation of the edges. In particular, in a CBN a directed edge $X \to Y$ establishes a cause-effect relationship between the cause X and its effect Y. Hence, reversing an edge in a BN that is not defined by a causal graph might result in the same underlying probability distribution for the pair of variables X and Y due to the Bayes theorem, while reversing an edge in a CBN where the graph is causal means changing the interpretation of the data generating mechanism, i.e. the causal mechanism. This concept is formalized by the following definitions.

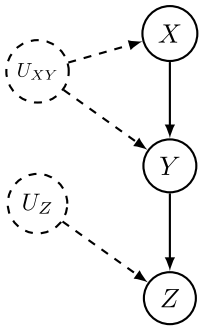
Definition 2.8 *(Direct and indirect cause).* For each directed edge $(X, Y) \in \mathbf{E}$, X is a *direct cause* of Y and Y is a *direct effect* of X. Recursively, every cause of X that is not a direct cause of Y, is an *indirect cause* of Y.

This definition is formally enforced by the *causal edge assumption* [9], as follows:

Definition 2.9 *(Causal edge assumption).* The value assigned to each X is completely determined by the function f given its parents:

$$X := f(Pa(X)) \qquad \forall X \in \mathbf{V}. \tag{2.1}$$

As natural consequence of Definition 2.9, we can define models that entail both the structural representation and the set of functions, i.e. the underlying causal mechanism.



(a)

$$M = (\mathbf{V}, \mathbf{U}, \mathbf{F}, P)$$
$$\mathbf{V} = \{X, Y, Z\}$$
$$\mathbf{U} = \{U_{XY}, U_Z\}$$
$$\mathbf{F} = \begin{cases} f_X : X := 2U_{XY}, \\ f_Y : Y := X + U_{XY}, \\ f_Z : Z := 3Y + U_Z \end{cases}$$
$$P = \begin{cases} U_{XY} \sim \mathcal{N}(0,1), \\ U_Z \sim \mathcal{N}(0,1) \end{cases}$$

(b)

Fig. 2.1. The causal graph G (a) of the related SCM M (b). In (a) X is a direct cause of Y and an indirect cause of Z, while Y is an effect, and in particular a direct effect, of X. An example of associated SCM is reported in (b), where the functional set **F** follows the causal edge assumption.

Definition 2.10 *(Structural causal model).* A *structural causal model* (SCM) [9,18] is defined by the tuple $M = (\mathbf{V}, \mathbf{U}, \mathbf{F}, P)$, where:

- **V** is a set of *endogenous* variables, i.e. *observable* variables,
- **U** is a set of *exogenous* variables, i.e. *unobservable*[1] variables, where $\mathbf{V} \cap \mathbf{U} = \varnothing$,
- **F** is a set of *functions*, where each function $f \in \mathbf{F}$ is defined as $f_i : (\mathbf{V} \cup \mathbf{U})^p \to \mathbf{V}$, with p the ariety of f, so that f determines completely the value of V_i,
- P is a joint probability distribution over the exogenous variables $P(\mathbf{U}) = \prod_i P(U_i)$.

Structural Causal Models are also known as Structural Equation Models (SEMs).

The joint *exogenous distribution* P is responsible for the non-deterministic nature of the model, adding a layer of uncertainty through a set of independent *noise* distributions. The unobserved terms **U** are represented in Fig. 2.1 as dashed vertices with dashed edges.

2.3. The causal discovery problem

The causal discovery problem [19] consists in selecting a causal graph as a possible explanation for a given dataset.

Formally, let **G** be the set of graphs defined over the variables **V** of a dataset **D** and $G^* \in \mathbf{G}$ be the *true but unknown* graph from which **D** has been generated.

Definition 2.11 *(Causal discovery problem).* The causal discovery problem [7] consists in recovering the *true* graph G^* from the given dataset **D**.

A causal discovery algorithm is said to *solve* the causal discovery problem if and only if it *converges* to the true graph G^* in the limit of the sample size of the dataset **D**.

Definition 2.12 *(Soundness and completeness).* A causal discovery algorithm is *sound* if it is able to solve the causal discovery problem, and it is *complete* if it outputs the *most informative* causal graph G that can be recovered from the input dataset **D**, without making further assumptions.

While the criterion by which a graph is said to be "most informative" depends on the context, in general it refers to the case in which neither the presence of an edge nor its orientation can be modified further without proving more information, i.e. prior knowledge or making additional assumptions.

Definition 2.13 *(Consistency of a causal graph).* A causal discovery algorithm is *consistent* [5,7] if it outputs a graph G that induces a probability distribution consistent with the input dataset **D**.

Definition 2.14 *(Identifiability of a causal graph).* A causal discovery algorithm is said to *identify* [9] a graph G if it is able to determine the direction of any edge in G.

[1] Authors in [8] define variables as exogenous if they are determined by factors *outside* the model, for which we choose not to explain the causes, without the precise connotation of unobservable, although typically they are.

In the following pages we will see that some algorithms are able to identify the causal graph *up-to its equivalence class* (Definition 2.21), meaning that setting the direction of any of the remaining undirected edges would not induce a different probability distribution, i.e. it is not possible to choose a specific direction for that edge without further assumptions.

Moreover, some of these methods are able to exploit only *observational* distributions, i.e. probability distributions that are induced by observation dataset, while others are capable of taking advantage of *interventional* distributions, i.e. probability distributions that are generated by experimental data, where we intervene on the system of interest.

Finally, even though the general formulation of the causal discovery problem is focused on the causal graph only, causal discovery algorithms are usually designed to find a solution w.r.t. a specific set of functions [20–23], e.g. non-linear equations.

2.4. Acyclicity and faithfulness

Definition 2.15 *(Markov property).* A graph $G = (\mathbf{V}, \mathbf{E})$ is said to satisfy the *Markov property* if the associated joint probability distribution $P(\mathbf{V})$ can be decomposed *recursively* as:

$$P(\mathbf{V}) = \prod_{X \in \mathbf{V}} P(X | Pa(X)) \tag{2.2}$$

The probability factorization expressed in Equation (2.2) relies on the assumption that the relationships encoded by the graph match exactly the underlying conditional probability independencies:

$$X \perp\!\!\!\perp_P Y \,|\, \mathbf{Z} \implies X \perp\!\!\!\perp_G Y \,|\, \mathbf{Z} \tag{2.3}$$

where $\mathbf{Z}$ is a subset of $\mathbf{V} \backslash \{X, Y\}$.

Essentially, it is assumed that the probability independence ($\perp\!\!\!\perp_P$) implies the graphical independence ($\perp\!\!\!\perp_G$), as stated in Equation (2.3).

This assumption is known as *d-faithfulness* or "directed faithfulness". In fact, the graphical model is required to rely on a DAG in order to satisfy the Markov property. More recently, extensions of the faithfulness assumption to the cyclic setting have been taken into consideration, e.g. *σ-faithfulness* [24,25], enabling the discovery of general non-acyclic DGs.

In order to test whether a variable X is conditionally independent from Y given a set $\mathbf{Z}$ in any probability distribution P faithful to G, one can use the *d-separation* criterion, which is based on the concept of *blocked path*.

In particular, when $\mathbf{Z}$ *blocks* every path between X and Y, we say that X and Y are *d-separated* by $\mathbf{Z}$. A path π is blocked depending on the presence of specific graphical patterns in it, as given in the following two definitions.

Definition 2.16 *(Fork, chain & collider).* Let $G = (\mathbf{V}, \mathbf{E})$ be a DG and π be a path on G. Then, given three vertices X, Y and Z in π, we have the following:

- $X \leftarrow Y \rightarrow Z$ is a *fork* on π,
- $X \rightarrow Y \rightarrow Z$ is a *chain* on π, and
- $X \rightarrow Y \leftarrow Z$ is a *collider* on π.

Definition 2.17 *(d-separation).* Let $G = (\mathbf{V}, \mathbf{E})$ be a DG, π be a path on G and $\mathbf{Z}$ a subset of $\mathbf{V}$. The path π is *blocked* [9] by $\mathbf{Z}$ if and only if π contains:

- a fork $X \leftarrow Y \rightarrow Z$ or a chain $X \rightarrow Y \rightarrow Z$ such that the middle vertex Y is in $\mathbf{Z}$, or
- a collider $X \rightarrow Y \leftarrow Z$ such that middle vertex Y, or any descendant of it, is not in $\mathbf{Z}$.

The set $\mathbf{Z}$ d-separates X from Y if it blocks every path between X and Y.[2] (Fig. 2.2.)

2.5. Equivalence classes

In the previous paragraphs we introduced the concept of causal graph as natural consequence of the causal edge assumption, where the functional set $\mathbf{F}$ is mapped to a directed graph G.

The naïve representation of a DAG does not allow to convey the (lack of) knowledge that typically arises during a discovery procedure. Here, we define formally other graphical representations, along with their interpretations.

[2] In a more general setting, d-separation can be extended to set of vertices rather than just singletons [26].

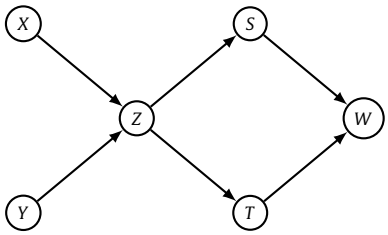


Fig. 2.2. In this figure, X and Y are d-separated without conditioning on Z, since they form a collider. The same does not hold for X and S, given that they form a chain by means of Z, and therefore conditioning (i.e. setting its value) on the middle vertex Z d-separates X from S.

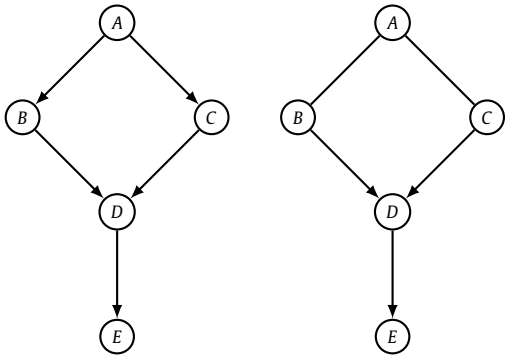


Fig. 2.3. A DAG on the left and its CPDAG (Definition 2.23) on the right. As we can see, both graphs have the same underlying structure (i.e. skeleton), but differ from the orientation of some of the edges. Specifically, the edges connecting A to B and C can be rearranged to form different chains or a fork. This is not true for the others edges in the CPDAG, since they are compelled. In fact, modifying the orientation of one of them would either remove the v-structure formed by $B \to D \leftarrow C$ or introduce a new one.

Definition 2.18 *(Partially DAG).* The graph G is a *partially-directed acyclic graph* (PDAG) if it can contain both undirected ($-$) and directed ($\to$) edges.

This alternative representation allows to distinguish a cause-effect pair ($X \to Y$) from a yet unknown relationship ($X - Y$), where there is still uncertainty about the direction of the edge. PDAGs are also called *patterns* [7].

Definition 2.19 *(Skeleton).* Let G be a PDAG. The skeleton of G is the undirected graph resulting from changing any directed edge of G to undirected.

Definition 2.20 *(V-structure).* Let G be a PDAG. A v-structure in G is a triple $X \to Y \leftarrow Z$ where X and Z are not adjacent. V-structures are also called *unshielded colliders* [27].

In the context of PDAGs, v-structures encode the conditional independencies that shape the associated probability distribution. Any edge that, when reversed, would either add or remove a v-structure is said to be a *compelled* edge, as in Fig. 2.3. Any compelled edge, along with the underlying skeleton, is a constraint for the set of observational distributions consistent with the given PDAG. Any non-compelled edge is called *reversible*.

Definition 2.21 *(Observational equivalence).* Two DAGs G and H are *observationally Markov equivalent* [27] if they have the same skeleton and the same v-structures, denoted as $G \equiv H$.

Henceforth, the definition of equivalence stems from an observational point of view, where graphs are compared in terms of the observational probability that is faithful to the given structure. In fact, changing the orientation of an reversible edge leads to a different structure with an equivalent factorization of the associated probability distribution.

Definition 2.22 *(Observational equivalence class).* Two DAGs G and H belong to the same *observational Markov equivalence class* (MEC) [25,28,29] if they are Markov equivalent. As generalization, the MEC of a graph G, denoted by $[G]$, represents the set of possible DAGs that are observationally equivalent.

Since MECs are defined in terms of skeletons and v-structures only, edges that are not part of any v-structure remain undirected, meaning that, given the limited knowledge, it is not possible to disentangle the relationship between the two variables.

Definition 2.23 *(Completed PDAG).* A PDAG G is said to be *completed* [7] if any directed edge is compelled and any undirected edge is reversible w.r.t. its MEC $[G]$.

The usual representation of a MEC is a *complete partially-directed acyclic graph* (CPDAG), also called *essential graph* [30] or *maximally oriented graph* [31]. Although the discovery problem is focused on recovering the true graph G^* from a dataset $\mathbf{D}$, it is not always possible to retrieve a *specific* instance, but rather its MEC $[G^*]$.

2.6. Sufficiency vs. insufficiency

In many applications, the collected variables are assumed to be sufficient to find the causes of a system of interest. This condition rarely holds true in real world scenarios [32].

Definition 2.24 *(Causally sufficient set).* The set of variables $\mathbf{V}$ is said to be *causally sufficient* if and only if every cause of any subset of $\mathbf{V}$ is contained in $\mathbf{V}$ itself.

That is, there are no *unobserved* variables $\mathbf{U}$ that affect the behavior of the causal mechanism generating the dataset $\mathbf{D}$. If at least one latent cause exists, then $\mathbf{V}$ is *causally insufficient*, which means that there exists a non-empty set of unobserved variables $\mathbf{U}$ that contains at least a cause of $\mathbf{V}$. In this case, G is only a sub-graph of the *augmented graph* G^a [24,33] defined over $\mathbf{V} \cup \mathbf{U}$, as depicted in Fig. 2.1a. However, listing which variables are going to be included in $\mathbf{U}$ is not an easy task. Unless prior knowledge is available, it is generally assumed [9] that, for each variable $V_i \in \mathbf{V}$, there exists one and only one variable $U_i \in \mathbf{U}$ that is parent of V_i, i.e. there is an edge $U_i \to V_i$ in G. Hence, each endogenous variable will be influenced by one exogenous variable in the simplest scenario, while in complex settings an observed variable may share one or more unobserved parents with others variables in $\mathbf{V}$, i.e. what are usually called *common latent causes*. On the algorithmic side, causal discovery methods, such as the Fast Causal Inference (FCI) [34], limit the search space by testing for potential unobserved variables that are common parents of two observed variables, as graphically explained in the figures in the following pages.

The equivalence class related to constraint-based causal insufficient methods relies on the concept of *mixed graph* and its properties.

Definition 2.25 *(Mixed graph).* The graph G is a *mixed graph* (MG) [35,34] if it contains undirected ($-$), directed ($\to$) and bidirected ($\leftrightarrow$) edges.

In mixed graphs the focus is on the *edge endpoints*, also called *marks*, rather than on the edge itself. For example, the directed edge $X \to Y$ is decomposed in two marks: the one insisting on $X-\cdots$ and the one insisting on $\cdots\!\to Y$. For this reason, we refer to the former as the *tail* mark ($-$) and the latter as the *arrowhead* mark ($>$). Therefore, a bidirected edge is an edge with both marks set to arrowheads.

In a bidirected edge $X \leftrightarrow Y$, X is a *spouse* of Y and vice versa. Therefore, the set of vertices connected with a bidirected edge to X is the *spouse set* $Sp(X)$. The graphical relationships inherited from partially directed graphs remain the same.

The fork, chain and collider patterns must be revised in the context of bidirectional edges. Let G be a MG and π a path on G. The pattern $X{*\!\!\to} Y {\leftarrow\!\!*} Z$ is a collider on Y, where '$*$' stands for a generic mark. Any other pattern is a *non-collider*.

Definition 2.26 *(M-separation).* Let G be a MG, π be a path on G and $\mathbf{Z}$ a subset of $\mathbf{V}$. The path π is *blocked* [36] by $\mathbf{Z}$ if and only if π contains:

- a non-collider such that the middle vertex is in $\mathbf{Z}$, or
- a collider such that middle vertex, or any descendant of it, is not in $\mathbf{Z}$.

The set $\mathbf{Z}$ m-separates X from Y if it blocks every path between X and Y.

Definition 2.27 *(Ancestral graph).* A mixed graph G is *ancestral* if:

- G has no (directed) cycles, and
- $X \in Sp(Y)$, then $X \notin An(Y)$, and
- $X \in Ne(Y)$, then $Pa(X) = \varnothing \wedge Sp(X) = \varnothing$.

These conditions allow an insightful interpretation of arrowheads in mixed graphs. In particular, in ancestral graphs, an arrowhead implies non-ancestorship, which explains why these representations are particularly useful in defining causal relationships.

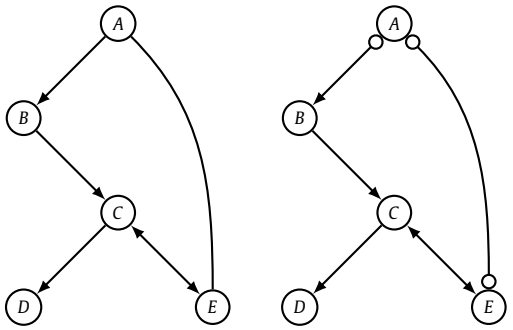


Fig. 2.4. A mixed graph on the left and one of its possible PAGs on the right.

Definition 2.28 *(Maximal ancestral graph).* An ancestral graph is *maximal* (MAG) if any pair of non adjacent vertices are *graphically separated* (in terms of m-separation).

As for the previous definition of the Markov equivalence class of DAGs using a CPDAG, the MEC of a set of MAGs is represented using a *partial ancestral graph* (PAG). A mark that is present in the same location in any MAG of a MEC is called *invariant*.

Definition 2.29 *(Partial ancestral graph).* The graph G is a *partial ancestral graph* (PAG) if it contains any combination of the following edge marks: tail ($-$), arrowhead ($\rightarrow$) and circle ($\circ$). Moreover, let $[G]$ be the MEC associated to G, then:

- G has the same adjacencies of $[G]$, and
- any arrowhead mark in G is invariant in $[G]$, and
- any tail mark in G is invariant in $[G]$.

As a direct consequence of this PAG definition, any circle mark present in G represents a variant mark in $[G]$, as for reversible edges of CPDAGs. Thus, PAGs are *the most informative* (Definition 2.12) representation of MECs for MAGs, hence, they satisfy the same *completed* definition of CPDAG.

The interpretation of PAGs can be tricky:

1. $(X \rightarrow Y)$: X causes Y and Y does *not* cause X, there may be an unobserved confounder,
2. $(X \leftrightarrow Y)$: Neither X causes Y nor Y causes X, there is an unobserved confounder that causes both X and Y,
3. $(X \circ\!\!\rightarrow Y)$: Either X causes Y, or there is an unobserved confounder that causes X and Y. In this case, Y does *not* cause X.
4. $(X \circ\!\!-\!\!\circ Y)$: Exactly one of the following holds: X causes Y or vice versa; there is an unobserved confounder that causes X and Y; or both (1) and (3) hold; or both (2) and (3) hold.

Understanding which causal statement is implied by each equivalence class is fundamental for a coherent interpretation of its graphical representation, as made explicit in Fig. 2.4.

Depending on additional assumptions, such as homoscedasticity or non-linearity, some algorithms are able to identify the causal graph beyond its equivalence class and recover a single graph instance [37,23,38].

2.7. Adding prior knowledge

Sometimes a cause-effect pair is known to exist (or to not exist) *a priori*, e.g. through expert's elicitation. Following the causal edge assumption, we can explicitly represent pairs as directed edges, defining a knowledge base composed of required (or forbidden) causal statements.

Definition 2.30 *(Knowledge base).* A *knowledge base* K is defined as an ordered pair $(\mathbf{R}, \mathbf{F})$, where $\mathbf{R}$ is the set of required directed edges, while $\mathbf{F}$ is the set of forbidden directed edges.

The knowledge base K is a valid representation for the given *background knowledge*. There exists a class of algorithms that are capable of taking advantage of this prior knowledge [31,28], either by integrating such knowledge before the actual discovery step or by checking if the resulting graph is consistent *a posteriori*.

Table 2
Algorithms classified by supported (✓) and unsupported settings.

Algorithm	Year	Category	Output	Non-Linear	Insufficient	Cyclic	Intervention
PC [39]	1991	Constraint	CPDAG				
FCI [34]	2008	Constraint	PAG		✓		
GES [40]	2013	Score	CPDAG				
FGES [41]	2017	Score	CPDAG				
ARGES [42]	2018	Hybrid	CPDAG				
GFCI [43]	2016	Hybrid	PAG		✓		
HCR [44]	2018	Score	DAG				
bQCD [45]	2020	Asymmetric	PDAG				
LiNGAM [46,23]	2014	Asymmetric	DAG		✓		
NOTEARS [47]	2018	Score	DAG				
CCD [48]	1996	Constraint	PAG			✓	
LiNG [49]	2012	Asymmetric	DG			✓	
dseptor [50]	2017	Exact	MG		✓	✓	
bcause [51]	2020	Exact	MG		✓	✓	
σ-CG [52]	2018	Constraint	σ-CG	✓	✓	✓	
GIES [53]	2012	Score	CPDAG				✓
IGSP [29]	2018	Score	CPDAG				✓
UT-IGSP [54]	2020	Score	CPDAG				✓
FCI-JCI [28]	2020	Constraint	PAG		✓	✓	✓
Ψ-PC [55]	2020	Constraint	CPDAG				✓
Ψ-FCI [55]	2020	Constraint	PAG		✓		✓
backShift [56]	2015	Asymmetric	MG		✓	✓	✓
bcause+ [57]	2020	Exact	MG	✓	✓	✓	✓
DCDI [58]	2020	Asymmetric	DAG	✓			✓

3. Causal discovery

In this section we introduce the first class of causal discovery algorithms. Here, the hypothetical dataset is represented by static observational data samples, neither interventional information nor time dependencies are taken into account. A summary of the explored algorithms can be found in Table 2.

3.1. Constraint-based algorithms

Constraint-based algorithms try to recover the causal graph by exploiting a set of *conditional independence statements* (CISs) obtained from a sequence of statistical tests. This class of methods translates conditional probability independence into graphical separation by assuming *faithfulness* (Subsection 2.4) of the underlying distribution.

Definition 3.1 *(Perfect map).* A graph G is said to be a *perfect map* [59,60] for a probability distribution P if every CIS derived from G can also be derived from P and vice versa:

$$X \perp\!\!\!\perp_P Y \,|\, \mathbf{Z} \iff X \perp\!\!\!\perp_G Y \,|\, \mathbf{Z} \tag{3.1}$$

where $\mathbf{Z}$ is a subset of $\mathbf{V}$.

Definition 3.2 *(Conditional independence test).* The null H_0 and alternative hypotheses H_1 are defined as $H_0 : X \perp\!\!\!\perp_P Y \,|\, \mathbf{Z}$ and $H_1 : X \not\perp\!\!\!\perp_P Y \,|\, \mathbf{Z}$, let $I(X, Y|\mathbf{Z})$ denote a *conditional independence* (CI) test. The null hypothesis H_0 is not rejected if and only if the estimated p-value $\hat{I}(X, Y|\mathbf{Z})$ is higher than a chosen significance level α:

$$\hat{I}(X, Y|\mathbf{Z}) > \alpha \implies X \perp\!\!\!\perp_P Y \,|\, \mathbf{Z} \tag{3.2}$$

where $\mathbf{Z}$ is a subset of $\mathbf{V}$.

When faithfulness is assumed, probability independence implies graphical separation.[3] The main limitation of this approach is related to the exponential growth of the *conditioning set* $\mathbf{Z}$. Indeed, given the pair (X, Y), in the worst case scenario where X is dependent on Y (or vice versa), the algorithm is required to test for $2^{|\mathbf{V}\setminus\{X,Y\}|}$ conditioning sets.

Constraint-based methods are generally capable of integrating prior knowledge into the learning process.

Conditional independence and data types. Constraint-based techniques are essentially *agnostic* of the specific conditional independence test that is being used. Indeed, it is possible to take advantage of such approaches in a wide variety of scenarios, as long as the assumptions of the said test are satisfied. While the main focus of causal discovery studies has been into either discrete or continuous settings, recent advances in conditional independence testing [61,62] extend existing tests to mixed-data.

3.1.1. Peter-Clark (PC)

One of the most studied algorithm that leverages the CISs is the *Peter-Clark* (PC) algorithm [7] together with its variants [39,63].

The first step of the procedure consists in defining a complete undirected graph over the variables of the given dataset $\mathbf{D}$. Subsequently, a sequence of CI tests are performed following an heuristic strategy [39], in order to minimize the number of tests needed. For instance, it is known that the power of CI test decreases when the size of the conditioning set increases [64], due to the curse of dimensionality. A common approach consists in selecting an upper limit to the size of the conditioning set, discarding computational-intensive time-wasting tests with low significance levels.

The obtained independence statements are then used to remove the associated edges and to identify the underlying skeleton. Finally, the remaining edges are oriented according to a set of rules [31] that leverage the identified v-structures and acyclicity property.

The resulting equivalence class is returned as a CPDAG, where the remaining undirected edges are reversible for the given observational distribution that arises from the data.

3.1.2. Fast Causal Inference (FCI)

A first extension of the PC algorithm to the causal insufficient setting (Subsection 2.6) is represented by the Fast Causal Inference (FCI) algorithm [65,34]. Specifically, the FCI algorithm relaxes both the assumption of no latent confounding [6] and no selection bias [66] in the observational setting, pushing the causal discovery problem a step closer to real-world scenarios. In this context, the authors leverage the definition of *discriminating path* to derive a new set of orientation rules.

Definition 3.3 *(Discriminating path).* Let G be an ancestral graph, a path $\pi = (X - \cdots - W - Z - Y)$ between X and Y is a discriminating path for Z if i) π contains at least three edges, ii) X is not adjacent to Y, iii) Z is adjacent to Y, and iv) every vertex between X and Z is a collider on π and parent of Y.

Discriminating paths are closely related to the *separation sets* identified by the PC algorithm: if a path π between X and Y is discriminating for Z, then Z is a collider on π iff every set that separates X and Y does not contain Z, otherwise it is a non-collider iff every set that separates X and Y contains Z.

3.2. Score-based algorithms

Score-based algorithms are usually structured around the maximization of a measure of fitness of a graph G through a space of possible graphs $\mathbb{G}$ for the observed samples $\mathbf{D}$, following a defined *scoring criterion* $\mathcal{S}(G, \mathbf{D})$ [67]:

$$G^* = \underset{G \in \mathbb{G}}{\operatorname{argmax}}\, \mathcal{S}(G, \mathbf{D}) \tag{3.3}$$

In the next few paragraphs, a set of properties for scoring criteria are introduced, before shifting the focus on an optimal two-step procedure for the causal sufficient scenario (Definition 2.6).

Definition 3.4 *(Decomposable score).* A scoring criterion $\mathcal{S}(G, \mathbf{D})$ is *decomposable* if it can be defined as a sum of the scores over a vertex and its parents:

$$\mathcal{S}(G, \mathbf{D}) = \sum_{X \in \mathbf{V}} \mathcal{S}(X, Pa(X), \mathbf{D}) \tag{3.4}$$

[3] Here the term *separation* is used as a placeholder for a generic graphical separation, which is intended as d-separation for directed graphs and m-separation for mixed graphs.

As direct consequence of this property, during the discovery procedure, the score computation can be simplified in terms of local differences of the causal graph.

Moreover, the comparison of scores of two DAGs G and H can be handled by taking into account only the vertices that have different parent sets.

Definition 3.5 *(Equivalent score).* A scoring criterion $\mathcal{S}(G, \mathbf{D})$ is *score equivalent* if $\mathcal{S}(G, \mathbf{D}) = \mathcal{S}(H, \mathbf{D})$, for each pair of graphs G and H in the same equivalence class.

A graph G is said to *contain* a probability distribution P if there exists an independence model associated with G that represents P exactly, i.e. G is a perfect map of P (Definition 3.1).

Definition 3.6 *(Consistent score).* Let $\mathbf{D}$ be a dataset associated with a probability distribution P, and let G and H be two graphs. A scoring criterion $\mathcal{S}$ is said to be *consistent* in the limit of the number of samples if and only if:

- If only G contains P, then $\mathcal{S}(G, \mathbf{D}) > \mathcal{S}(H, \mathbf{D})$,
- If both G and H contain P and the model associated with H has fewer parameters that the one with G, then $\mathcal{S}(G, \mathbf{D}) < \mathcal{S}(H, \mathbf{D})$.

If a scoring criterion is both decomposable and consistent, then it is *locally consistent*.

Definition 3.7 *(Locally consistent score).* Let G be a graph and H the graph resulting from addition of the edge $X \rightarrow Y$ to G. A scoring criterion $\mathcal{S}(G, \mathbf{D})$ is said to be *locally consistent* if and only if:

- $X \not\perp\!\!\!\perp_P Y \mid Pa(X) \implies \mathcal{S}(H, \mathbf{D}) > \mathcal{S}(G, \mathbf{D})$, and
- $X \perp\!\!\!\perp_P Y \mid Pa(X) \implies \mathcal{S}(H, \mathbf{D}) < \mathcal{S}(G, \mathbf{D})$.

Explicitly, if a scoring criterion is locally consistent then the score:

- Increases when any edge that eliminates an independence constraint that does not hold in the generative distribution is added, and
- Decreases when any edge that does not eliminate such a constraint is added.

This property guarantees that any deletion of an unnecessary edge will produce a higher score value, allowing the definition of an optimal greedy search algorithm.

One of the most commonly used score criterion is the Akaike Information Criterion (AIC) [68]:

$$AIC = 2k - 2\ln \hat{L} \tag{3.5}$$

where k is the number of parameters of the model and $\hat{L}$ is the maximum value of the likelihood for the given model. Models achieving a lower value of AIC are preferred, i.e. they explain better the observed data. Another common scoring criterion is offered by the Bayesian Information Criterion (BIC) [69], also known as the Schwarz Information Criterion:

$$BIC = k\ln n - 2\ln \hat{L} \tag{3.6}$$

which differs from AIC due to the parameters penalty term that takes into account the number of observations n. Others commonly used scoring criteria are the Bayesian Dirichlet equivalent uniform (BDeu) [70] and the Bayesian Dirichlet sparse (BDs) [71].

Definition 3.8 *(Optimal equivalence class).* Let $[G]^*$ be the equivalence class that is a perfect map of the probability distribution P and $\mathbf{D}$ the associated dataset. $[G]^*$ is said to be the *optimal* equivalence class if and only if:

$$\mathcal{S}([G]^*, \mathbf{D}) > \mathcal{S}([G], \mathbf{D}) \qquad \forall [G] \neq [G]^* \tag{3.7}$$

in the limit of the number of samples, for any consistent scoring criterion $\mathcal{S}$, following Definition 3.6.

3.2.1. Greedy Equivalent Search (GES)

The Greedy Equivalence Search (GES) [40,72] is optimal in the limit of the number of samples [67]. The first step of the algorithm consists in the initialization of the empty graph G. The algorithm is composed by two phases: the *forward search* and the *backward search*. In the forward search phase, i) G is modified by repeatedly adding the edge that has the highest difference in score (i.e. delta score), until there is no such edge that increases the score. In the backward search phase, ii)

the edge that again achieves the highest delta score is repeatedly removed. The algorithm terminates once it reaches a local maximum during the backward search phase.

This algorithm is designed to work under causal sufficiency. When this assumption no longer holds, the procedure is known to introduce extra edges as a compensation behavior for the unobserved relationships. For example, when a fork ($X \leftarrow Y \rightarrow Z$) is present and the middle vertex is indeed latent, GES will likely add an edge between the other two observed vertices of the structure, even if such edge is not present in the true graph. Any algorithm that is based on this technique and does not address the issue directly displays such pattern.

3.2.2. Fast GES (FGES)

Score-based algorithms are as fast as the computation of the chosen scoring criterion is. Leveraging the properties of the score function, it is possible to minimize the number of computations needed by storing previous intermediate evaluations. Not only this optimization reduces the computation time considerably, but also allows the application of these methods to high-dimensional datasets [61,41]. This "fast" variant of GES (FGES) caches partial graph scores, significantly increasing the memory usage, since relevant fragments of the graph may be considered. Moreover, computationally expensive sections of the algorithm can be parallelized, taking advantage of high performance computing (HPC) settings.

3.3. Hybrid algorithms

With the term "hybrid" algorithms we refer to the class of methods that combine constraint-based and score-based approaches to mitigate their drawbacks.

3.3.1. Adaptively Restricted GES (ARGES)

Consistency of constraint- and score-based algorithms is usually proved in low-dimensional use cases, where the number of samples is orders of magnitude greater than the number of variables. Hybrid approaches generally lacks a formal and rigorous proof of consistency, leading to undefined behavior. For this reason, an adaptively restricted variant of GES (ARGES) [42] has been developed, targeting specifically the consistency weakness in both low- and high-dimensional spaces.

The novelty of this hybrid version of GES stems from the concept of *admissible edge*. Let G be a CPDAG and X and Y be a pair of non adjacent vertices on it. Adding an edge between X and Y is admissible for the graph G if i) X and Y are adjacent in the (estimated) skeleton of G, or ii) there exists a node Z such that $X \rightarrow Z \leftarrow Y$ is a v-structure in G.

From the definition of admissible edge, an equal admissible move consists in adding such edge to the graph and obtain a new equivalent CPDAG. This point is sufficient to prove that the resulting forward phase of ARGES is consistent when restricted to admissible moves (i.e. it is an independence map for the given observational probability distribution [67]).

3.3.2. Greedy FCI (GFCI)

Score-based causal discovery algorithms such as GES and FGES are asymptotically correct, but are not designed to work in a causal insufficient scenario (Definition 2.6), where unmeasured confounders are present in the true graph. Constraint-based causal search algorithms, such as FCI, are asymptotically correct even with unmeasured confounders, but often perform poorly on small samples. The Greedy Fast Causal Inference (GFCI) [43] algorithm combines score-based and constraint-based algorithms improving over the previous results while being asymptotically correct (Definition 2.12) under causal insufficiency.

Specifically, the initial skeleton is obtained by un-orienting the CPDAG resulting from the execution of FGES. Then, the orientation rules of FCI are applied, with only a few slight modifications that rely on original FGES output. This approach leads to an improved accuracy over the distinct constraint- and score-based approaches. As a side effect, additional requirements arise from the union of these methods. For example, not only the conditional independence test is required to be consistent by FCI, but also the associated score must be *locally* consistent due to FGES. This constraint reduces the practical applications to settings where indeed such score exists.

3.4. Other methods

3.4.1. Hidden Compact Representation (HCR)

Causal discovery methods for discrete and mixed variables have gained renovated interested in the last few years [61,62]. Although additive noise models have been widely used in the context of continuous variables, it is difficult to justify their application with categorical data, where the addition operator between the levels of variables is not well defined.

For this reason, authors in [44] developed a new low-dimensional embedding for discrete variables, allowing a (hidden) compact representation (HCR) of the discrete states of such variables. The method follows a two-stage procedure: at first, a discrete variable is deterministically mapped into a low-cardinality representation (e.g. binary), which acts as a proxy for the information contained in the original variable; then, a set of samples are drawn for the new proxy variable using a probabilistic mapping. The overall complexity of the model in controlled using the BIC score, balancing between total fitness and size of parameters.

The authors address the problem of identifiability (Definition 2.14) of the model and prove that, under mild conditions, the causal graph recovered from observational data is identifiable. The method is tested against both synthetic and real-

world data, providing reference values for performance metrics. In these experiments, HCR outperforms linear models in terms of accuracy and sensitivity, especially when the additive noise assumption does not hold.

3.4.2. Bivariate Quantile Causal Discovery (bQCD)

The bivariate quantile causal discovery (bQCD) [45] technique is designed to uncover cause-effect pairs in the bivariate setting. By re-expressing independence statements in light of the minimum description length (MDL) [73], the authors build a discovery procedure by using *quantile scoring*.

Following [74], let X and Y be two random variables with joint, marginal and conditional distributions denoted by F_{XY}, F_X and $F_{X|Y}$, respectively. The key concept here is that a lower complexity follows from a correct causal orientation of the (X, Y) pair, since it is a more informative representation of the associated data.

Hence, the Kolmogorov complexity $K(F)$ is defined as the length of the shortest program F that outputs $F(X)$. Since $K(F)$ measures the information contained in F, authors in [75] state that if X causes Y, then $K(F_X) + K(F_{Y|X}) \leq K(F_Y) + K(F_{X|Y})$. The problem is that $K(F)$ cannot be computed in practice. Therefore, the authors rely on the MDL principle as a proxy for the Kolmogorov complexity. Such an approximation can be performed by estimating the population quantiles through nonparametric quantile regression.

The resulting procedure is robust to outliers and can be generalized to a wide range of distributions, although it requires that *all* population quantiles are computable, which could be a limiting factor in real-world applications.

3.4.3. Linear Non-Gaussian Acyclic Model (LiNGAM)

In the context of linear causal models, when causal sufficiency (Definition 2.24) holds, the observed variables can be expressed as a linear combination of the noise terms:

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e} \tag{3.8}$$

where $\mathbf{B}$ is matrix consisting of the coefficients of the variables in the associated set of equations, $\mathbf{x}$ is the vector of variables and $\mathbf{e}$ is the vector of noises.

Here, the exogenous distribution is assumed to be made of mutually independent (possibly non-Gaussian) variables. Solving for $\mathbf{x}$ reduces to the identification of the matrix $\mathbf{A}$ such that:

$$\mathbf{x} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{e} = \mathbf{A}\mathbf{e} \tag{3.9}$$

LiNGAM [23,38] relies on Independent Component Analysis (ICA) [76] to identify a possible solution for $\mathbf{A}$. In fact, multiple mixing matrices $\mathbf{A}$ are feasible solutions for the given joint probability distribution. This technique is essentially focused on discovering asymmetries in the sample distribution to determine the correct causal ordering. Once such ordering has been discovered, the causal graph is built by recovering all and only the edges coherent with the order.

LiNGAM has been extended later for causally insufficient settings [46]. Let $\mathbf{f}$ be the vector of latent variables and $\mathbf{\Lambda}$ the matrix of the connections strength between $\mathbf{f}$ and $\mathbf{x}$, then:

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{\Lambda}\mathbf{f} + \mathbf{e} \tag{3.10}$$

The proposed model can be solved with a variant of ICA, called *overcomplete* ICA, which takes into account the presence of unobserved effects.

LiNGAM consistently estimates the connection matrix $\mathbf{B}$. While standard ICA does not scale well in high-dimensional settings, approximated variants of ICA can be used to compute the components with a predefined fixed number of iterations with reasonable precision. This leads to an efficient solution in presence of non-Gaussian noise and causally insufficient datasets.

3.4.4. Continuous optimization (NOTEARS)

In the "DAGs with NO TEARS" [47] algorithm, the causal discovery problem is reduced to a continuous optimization problem. The acyclicity constraint is expressed as an equality constraint $h(\mathbf{W}) = 0$, where h is a smooth differentiable function that measures the "DAG-ness" (i.e. a quantification of the acyclicity violations) of a given adjacency matrix $\mathbf{W}$:

$$h(\mathbf{W}) = tr(e^{\mathbf{W}\circ\mathbf{W}}) - n = 0 \tag{3.11}$$

where tr, is the trace operator, $\circ$ is the Hadamard product, e^* is the matrix exponential and n the size of $\mathbf{W}$. Moreover, this function has a rather simple associated gradient:

$$\nabla h(\mathbf{W}) = (e^{\mathbf{W}\circ\mathbf{W}})^T \circ 2\mathbf{W} \tag{3.12}$$

Coefficients in $\mathbf{W}$ smaller than a fixed threshold $\omega > 0$ are set to zero, rounding the solution with an arbitrary precision. The evaluation of the matrix exponential is $O(n^3)$, i.e. cubic in the number of vertices. Given the low computational complexity, NOTEARS outperforms existing methods when both the in-degree and the sample size are large. However, even if the authors compared their approach to other methods (e.g. PC, FGES, LiNGAM), this algorithm is focused on exploring the DAG space, rather than addressing Definition 2.9 directly, i.e. an *acyclic* graph may not necessarily be a *causal graph*.

3.5. Comparison between methods

Now that different classes of methods have been explored, one might be tempted to ask which approach performs better or is faster. Unfortunately, it is difficult to restrict the focus on a specific algorithm due to both theoretical and practical limitations, but still, one could at least try to compare advantages and limitations across classes of algorithms.

On the theoretical side, only a few contributions [7,47] provide proofs of the asymptotic time complexity of such algorithms, making the comparison impossible due to the lack of references. On the practical side, there exist cases in which statistical criteria enforced during the evaluation steps of different algorithms do not match [77,78], and others where computational time is recorded using absolute measurements [78,79], i.e. seconds. These approaches not only make the comparison across different hardware configurations unfair, but are also affected by the size of the input. Furthermore, these issues are exacerbated by the fragmentation of the software packages across multiple programming languages, which are known to have different degrees of efficiency.

Nonetheless, a recent contribution [80] addresses these experimental pitfalls and conducts a rigorous analysis based on both synthetic and real-world data. While the authors do not exclude the existence of others potentially confounding factors, they essentially conclude that there is no systematic difference across classes of algorithms in terms of accuracy and speed, even taking into account a wide range of scenarios. Other works [81,60] state that constraint-based algorithms are fast in general, but early mistakes undermine the construction of the final structure due to the chained evaluation of conditional independence tests, especially in high dimensional settings.

These last considerations seem to push towards score-based approaches, also considering the possibility of leveraging parallelization during the score evaluation, but further experiments are needed [80] in order to take into account others methods properly.

4. Causal discovery with cycles

4.1. Cyclic SCM

In a SCM (Definition 2.10), the causal graph induces a functional set $\mathbf{F}$ where equations follow the decomposition enforced by the causal edge assumption, Subsection 2.9. If the causal graph is acyclic, then the SCM itself is called acyclic, or *recursive* SEM. The concept of recursion is linked to the hierarchical order that arises from the topological ordering of the underlying DAG. Indeed, it is possible to define a sequence $V_1, V_2, \dots, V_n$ of vertices over $\mathbf{V}$ such that for any V_i and V_j where $i < j$, V_j is not a cause of V_i [82].

Therefore, in a *non-recursive* SEM, or cyclic SCM, some endogenous variables are connected to each other, forming cycles that do not allow a recursive decomposition. Still, the causal edge assumption is satisfied, since its definition is consistent even in the presence of cycles.

4.2. No acyclicity assumption

Conditional independencies arising from cyclic SCMs are entailed by the cyclic graph [48]. It can be shown that, in general, there is no DAG encoding the conditional independencies that hold in such SCM [83]. Nonetheless, cyclic SCMs are widely used to model systems with *feedback*, and are applied in sociology, economics and biology, making this class of models a relevant target of interest for causal discovery techniques.

To test for such independencies, d-separation can be adapted to the cyclic setting under the assumption of causal sufficiency (Definition 2.6) [84]. In causally insufficient scenarios, d-separation can be replaced with σ-separation [52,33] applied to directed mixed graphs (DMGs), i.e. mixed graph (Subsection 2.25) without undirected edges.

Definition 4.1 *(Strongly connected component).* Let G be a DG and X a vertex in G. The *strongly connected component* [52] of a vertex X is defined as:

$$SCC(X) = An(X) \cap De(X) \tag{4.1}$$

that is, the set of vertices that are both ancestors and descendants of X, including X itself.

Definition 4.2 *(σ-separation).* Let G be a DMG, π be a path on G and $\mathbf{Z}$ a subset of $\mathbf{V}$. The path π is *blocked* [52,33] by $\mathbf{Z}$ if and only if π contains:

- a collider $X *\!\!\rightarrow Y \leftarrow\!\!* Z$ where $Y \notin An(\mathbf{Z})$, or
- a non-collider $X \leftarrow Y *\!\!-\!\!* Z$ (or $X *\!\!-\!\!* Y \rightarrow Z$) where $Y \in An(\mathbf{Z})$ and X (respectively Z) is part of $SCC(Y)$ (Equation (4.1)).

The set $\mathbf{Z}$ σ-separates X from Y if it blocks every path between X and Y.

The above graphical criterion implies d-separation and reduces to it in the case of DAGs.

4.2.1. Cyclic Causal Discovery (CCD)

The Cyclic Causal Discovery (CCD) algorithm [48] has been the only provably sound (Subsection 2.12) approach to general directed graphs until the development of LiNG [49] (Subsection 4.2.2). CCD is a constraint-based algorithm that follows the same initial procedure as the one of the PC algorithm, with five different orientation rules. CCD outputs a PAG G that differs from the output of FCI for a couple of additional patterns:

- underlining triples ($X \ast\!\!-\!\!\ast \underline{Y} \ast\!\!-\!\!\ast Z$), where Y is an ancestor of *at least* one of X or Z in every graph in $[G]$, and
- dotted underlining triples ($X \ast\!\!\rightarrow \underline{Y}_{\cdot\cdot} \leftarrow\!\!\ast Z$), where Y is not a descendant of a common child of X and Z.

These additional patterns arise from a fundamental problem: the algorithms are *not complete* (Definition 2.12), and, therefore, there may be features common to all graphs in the same equivalence class that are not present in the output PAG (i.e. it is not the most informative PAG). While not being complete in the same sense of the previous algorithms, CCD is *d-separation complete*, meaning that the resulting PAG represents an equivalence class with a single graph, i.e. it encodes all the needed conditional independencies. Therefore, CCD is useful when one is interested in querying the resulting graph about dependencies, but lacks the capability to represent every causal edge by definition, in contrast to others algorithms. This limitation makes CCD less suitable for the definition of SCMs, especially when one is interested in the form of the functional set.

4.2.2. Linear Non-Gaussian (LiNG)

The LiNGAM algorithm can be adapted to the cyclic setting by weakening the acyclicity assumption. Specifically, instead of targeting a DAG, LiNG (or LiNG-D family) [49] tries to recover a simple graph (i.e. without self-loops) by forcing all entries on the diagonal of the $\mathbf{B}$ matrix to be zero.

While LiNGAM output could be seen as a set of admissible models that contains a single model (i.e. the model is *identifiable*), the cyclic variant usually admits more than one causal graph at a time. In fact, the acyclicity assumption that allowed to find the row-permutation of $\mathbf{B}$ that best fits the given dataset is missing. The authors then suggest to limit the discovery procedure to the k-th best assignment, following the intuition that permutations associated to inadmissible models would score poorly asymptotically. This approach selects one single model from the equivalent class (i.e. returning set).

LiNG inherits both limits and strengths of the original method: approximate (or sparse) ICA can be a valid alternative if running the full ICA is computationally expensive for the considered task.

4.2.3. σ-connection graphs

From the concept of σ-separation, one can derive a MG (Definition 2.25) where conditional independencies are expressed in the presence of cycles and latent variables, namely a σ-Connection Graph (σ-CG). An algorithm to learn this structure from data has been developed [52] as a natural extension of the work presented in [85]. The causal discovery problem is re-casted as a continuous optimization problem based on the following loss function:

$$\mathcal{L}(G, \mathbf{S}) = \sum_{i=1}^{n} \lambda_i (\mathbb{1}_{\lambda_i > 0} - \mathbb{1}_{X_i \perp\!\!\!\perp_G Y_i | \mathbf{Z}_i}) \tag{4.2}$$

where $\mathbf{S}$ is a set of conditional independence statements expressed as $\mathbf{S} = \left((X_i, Y_i, \mathbf{Z}_i, \lambda_i)\right)_{i=1}^{n}$, X_i, Y_i and $\mathbf{Z}_i$ are variables in $\mathbf{V}$, $\lambda_i \in \mathbb{R} \cup \{-\infty, +\infty\}$ encodes the confidence of probabilistic conditional independence $X_i \perp\!\!\!\perp_P Y_i | \mathbf{Z}_i$ as a constraint and $\mathbb{1}$ is the indicator function which assumes the value one when the constraint is satisfied.

The λ_i weights are evaluated using the indicator function $\mathbb{1}$ to constrain the conditional dependence between variables. Therefore, Equation (4.2) quantifies the observations against the proposed causal graph based on the observed data. During the experimental evaluation, authors relied on weights proposed in [86]:

$$\lambda_i = \log p_i - \log \alpha \tag{4.3}$$

with p_i representing the p-value of a statistical test for conditional independence and α being a significance level.

Minimizing the loss function may lead to multiple optimal solutions, where each solution G is an instance of the actual equivalence class $[G]$. Indeed, as for d-separation and CPDAGs, the σ-separation criterion and the associated σ-CGs take into account possible undirected edges that are invariant for any causal graph belonging to the same equivalence class $[G]$.

This algorithm has been benchmarked against synthetic data in a low-dimensional setting. While the recovery metrics show consistent performance across the experiments, especially when increasing the number of interventions, it is clear that the main limitation of this approach is linked with the σ-separation encoding, as noted in [57]. Indeed, the separation checks are performed using Answer Set Programming (ASP), a declarative logic programming language, which slows down the learning procedure.

Table 3
Layers of causation with associated questions, practical examples and methods.

Layer	Question	Method
Observational	How would *seeing* X change my belief in Y?	Un/Supervised Learning
Interventional	What happens to Y if I *do* X?	Reinforcement Learning
Counterfactual	What would have happened to Y if I *had done* X' instead of X?	Structural Causal Model

4.2.4. bcause

The procedures described so far are essentially *approximate algorithms* that reduce the search space (i.e. the number of conditional independence tests) by using previously computed test results. In fact, edges that are tested in later phases rely on adjacent vertices that are selected in earlier steps of the algorithm. During the last few years, *exact* search approaches have been developed in a *branch-and-bound* fashion.

The *bcause* algorithm [51] explores the search space in a tree-like visit guided by an objective function that determines the *weight* of a potential solution. During the discovery phase, any edge of an intermediate result G is either *absent, present* or *undecided*. Before the actual branching step, the lower bound of the given objective function for the current partial solution G' is computed. If such bound is higher than the weight obtained by the previous solution G, the branch can be closed and the algorithm backtracks. Otherwise, if G' contains at least one undecided edge, the procedure branches recursively in two directions: one in which said edge is set as present and the other marked as absent. Finally, if the branch cannot be closed and G has no undecided edges, then the current solution G' is updated if and only if the evaluation of the objective function results in a lower weight. The search procedure will return G as a globally optimal solution.

Since the causal discovery problem is inherently exponential, an exact search algorithm is unfeasible in the general setting. However, if both the objective function and its lower bound can be efficiently evaluated, a constrained space for a low dimensional problem can be effectively explored. For example, the authors benchmark their method under different conditions, showing that assuming acyclicity results in a lower execution time. Moreover, the algorithm maintains a set of constraints satisfied by the local solution and updates them incrementally. Therefore, any incompatible extension of the current solution is ruled out by leveraging a linear programming solver, reducing the total number of evaluations needed.

5. Causal discovery with interventions

This section is focused on the difference between learning causal models using either observational or interventional data. While the former setting has been explored extensively in the past decades, only recently solutions for properly handling experimental data have been proposed.

5.1. Observational vs. interventional

In order to grasp the added value of experimental data, we will introduce the concept of *ladder of causality* [8,87] as a reference framework.

The ladder of causation. The ladder of causation, also called the *causal hierarchy*, is an ordered structure composed by three layers, where each layer is mapped to a *cognition* level: observational, interventional and counterfactual. A level inherently defines the set of potential queries that can be answered with the given information associated to it.

In practice, the observational layer is composed by associational or *factual* data, while the interventional layer is related to data that are generated by an *intervention* on the system, i.e. an experiment. Interacting with the system itself is the reason why these two levels are different. The counterfactual layer is the highest level of cognition, where one may ask what would have happened if a different intervention had been performed, opposed to the one that factually altered the system. This hypothetical scenario is strongly opposed to the observational one, being in the *counter-factual* space.

Even if the three layers represent different information levels, they are not distinct. In fact, each layer is a generalization of the previous one, e.g. the observational setting can be seen as a special case of the interventional scenario, where no intervention is performed. Therefore, the interventional layer *subsumes* the observational one. The same happens with the counterfactual layer w.r.t. the interventional one, provided that the former allows to define hypothetical actions that were not present in the latter, as expressed in Table 3.

At this point, one may ask how to formally represent the concepts expressed by this hierarchy, to operatively exploit the informative gap between the layers. The answer is provided by *do-calculus* [26].

do-calculus. Queries that are usually expressed in natural language can be rephrased in terms of probability distribution by introducing the *do* operator, whenever possible.[4] The *do* operator represents an intervention on a given variable, e.g.

[4] We restrict ourselves to a minimal introduction of the *do*-calculus, aiming to formally represent the set of concepts that are essential for the causal discovery scenario. For a broader discussion on *identification* and *estimation* of the causal effect, refer to [88].

$do(X = x)$ sets the value of X to x. This notation is usually overloaded by extending the operator over sets of variables, e.g. $do(\mathbf{X} = \mathbf{x})$ with $\mathbf{x}$ a vector of values. Finally, the $\mathbf{x}$ vector can be omitted entirely for brevity.

Definition 5.1 *(Rules of do-calculus).* Let G be a causal graph and P the probability distribution induced by G. For any disjoint subset of variables $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ and $\mathbf{W}$, the following three rules apply:

1. Insertion and deletion of observations:

$$P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{Z}, \mathbf{W}) = P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{W}) \tag{5.1}$$

   if $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{X}, \mathbf{W})$ holds true in $G_{\overline{\mathbf{X}}}$,
2. Exchange of observations and interventions:

$$P(\mathbf{Y} \mid do(\mathbf{X}), do(\mathbf{Z}), \mathbf{W}) = P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{Z}, \mathbf{W}) \tag{5.2}$$

   if $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{X}, \mathbf{W})$ holds true in $G_{\overline{\mathbf{X}},\underline{\mathbf{Z}}}$,
3. Insertion and deletion of interventions:

$$P(\mathbf{Y} \mid do(\mathbf{X}), do(\mathbf{Z}), \mathbf{W}) = P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{W}) \tag{5.3}$$

   if $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{X}, \mathbf{W})$ holds true in $G_{\overline{\mathbf{X},\mathbf{Z}(\mathbf{W})}}$,

where $G_{\overline{\mathbf{X}}}$ is the subgraph of G where the incoming edges into $\mathbf{X}$ are removed, $G_{\underline{\mathbf{Z}}}$ is the analogous for the outgoing edges from $\mathbf{Z}$, and finally $\mathbf{Z}(\mathbf{W})$ is $\mathbf{Z} \setminus An(\mathbf{W})$ w.r.t. the subgraph $G_{\overline{\mathbf{X}}}$.

With these rules, which are *correct and complete*, a causal effect can be identified if there exists a finite sequence of applications of such rules leading to a *do*-free expression of the considered probability distribution.

5.2. Types of interventions

Definition 5.2 *(Perfect intervention).* An intervention is said to be *perfect* (or *hard*) if it removes the causal dependencies (i.e. the incoming causal edges, as in Subsection 2.9) that affect the intervention target.

Indeed, *do*-calculus enables us to express perfect interventions in an operative framework, but there are other types of interventions that cannot be expressed using this notation.

Definition 5.3 *(Imperfect intervention).* An intervention is said to be *imperfect* (or *parametric*, *soft*) [89] if it does not remove the causal dependence that affects the intervention target, but alters the causal mechanism that represents such dependence.

For instance, an imperfect intervention on an SCM could be a change in the parameters that quantify the strength of the causal relationships, while a perfect intervention would result in hard setting them to zero. In this sense, perfect interventions are a subset of imperfect interventions, where some variables are removed from the equations of the functional set (Definition 2.10) as a special case.

Mechanism change. Imperfect interventions itself are a formal definition of a broader concept called *mechanism change* [90]. For a SCM M with a causal graph G and a set of parameters Θ associated to the function set $\mathbf{F}$. A mechanism change is a mapping from M to M', where the new set of parameters is defined as $\Theta' = \Psi' \cup (\Theta \setminus \Psi)$, with the new subset Ψ' that differs from the original subset Ψ. The change affects the behavior of the function set $\mathbf{F}$, inducing a set $\mathbf{F}'$.

5.3. Defining the intervention target

We can rephrase perfect and imperfect interventions under a single unified framework through the concept of intervention target [53].

Definition 5.4 *(Intervention target).* Let G be a causal graph. A subset $\mathbf{I} \subset \mathbf{V}$ is said to be an *intervention target* if it contains all and only the variables associated to an *intervention* over G.

Therefore, a single-variable intervention is an intervention target that contains only one variable, while in a multi-variable intervention contains more than one. As a special case, when $\mathbf{I} = \varnothing$ the intervention target represents the observational case. A set of multiple intervention targets $\{\mathbf{I}_0, \mathbf{I}_1, \ldots, \mathbf{I}_n\}$ is called an *intervention family* and it is denoted with the calligraphic letter $\mathcal{I}$.

Definition 5.5 *(Conservative family).* A family of targets $\mathcal{I}$ is *conservative* if for each vertex X in $\mathbf{V}$ there exists at least one intervention target in $\mathcal{I}$ that does not contain X:

$$\exists \mathbf{I} \in \mathcal{I} : X \notin \mathbf{I}, \quad \forall X \in \mathbf{V} \tag{5.4}$$

Essentially, a conservative family is a family that allows the existence of at least one intervention target that does not intervene on a specific variable. This property guarantees that there is at least one experiment in the family that does not alter the behavior of such variable if performed.

In this setting, a conservative family allows to observe the influence of a (known) set of targets on at least one unaffected variable, enabling the possibility of disentangling such effect, especially when compared to the other experiments in the whole family.

Definition 5.6 *(Intervention graph).* Let G be a causal graph and $\mathbf{I}$ be an intervention target defined over G. The *intervention graph* $G^{(\mathbf{I})} = (\mathbf{V}, \mathbf{E}^{(\mathbf{I})})$ is the causal graph obtained by removing any directed edge that points to a vertex in $\mathbf{I}$ from G:

$$\mathbf{E}^{(\mathbf{I})} = \{(X, Y) \mid (X, Y) \in \mathbf{E} \ \wedge \ Y \notin \mathbf{I}\} \tag{5.5}$$

This definition of intervention graph is coherent with the intervened graph resulting from a *do*-intervention [26], also known as *graph surgery* or *graph manipulation*.

We can now formally express the interventional distribution associated to an intervention graph.

Definition 5.7 *(Interventional distribution).* Let G be a causal graph and $\mathbf{I}$ be an intervention target. The *interventional distribution* $P^{(\mathbf{I})}$ can be expressed using the factorization formula:

$$P^{(\mathbf{I})} = \prod_{X \in \mathbf{I}} P^{(\mathbf{I})}\big(X | Pa(X)\big) \prod_{X \notin \mathbf{I}} P^{(\varnothing)}\big(X | Pa(X)\big) \tag{5.6}$$

where $P^{(\varnothing)}$ is the observational distribution of the variables that were not included in the intervention target, if any.

In case of perfect interventions, the interventional distribution can also be expressed using the *do*-notation:

$$P^{(\mathbf{I})} = \prod_{X \in \mathbf{I}} P^{(\mathbf{I})}\big(X \mid do(\mathbf{I})\big) \prod_{X \notin \mathbf{I}} P^{(\varnothing)}\big(X | Pa(X)\big) \tag{5.7}$$

Definition 5.8 *(Interventional equivalence).* Let G and H be two causal graphs and $\mathcal{I}$ be an intervention family. G and H are *interventionally Markov equivalent* w.r.t. the family $\mathcal{I}$ (i.e. $\mathcal{I}$-equivalent) if the associated intervention graphs $G^{(\mathbf{I})}$ and $H^{(\mathbf{I})}$ have the same skeleton and the same v-structures for each intervention target of the family:

$$G \equiv_{\mathcal{I}} H \implies G^{(\mathbf{I})} \equiv H^{(\mathbf{I})}, \quad \forall \mathbf{I} \in \mathcal{I} \tag{5.8}$$

In other terms, interventional equivalence can be decomposed in a set of equivalence statements of intervention graphs, where each observational equivalence statement is formulated against a single intervention target contained in the given family.

Definition 5.9 *(Interventional equivalence class).* Two causal graphs G and H belong to the same *interventional Markov equivalence class* w.r.t. the intervention family $\mathcal{I}$ ($\mathcal{I}$-MEC) [55,91] if they are $\mathcal{I}$-equivalent. As for the observational setting, the $\mathcal{I}$-MEC of a graph G, denoted by $[G]_{\mathcal{I}}$, represents the set of possible causal graphs that are interventionally equivalent.

An intervention family $\mathcal{I}$ induces a classification of the edges of an intervention graphs depending on the effect on the underlying interventional distribution.

Definition 5.10 *($\mathcal{I}$-covered edge).* An edge $(X \to Y)$ in G is *$\mathcal{I}$-covered* if:

$$Pa(X) = Pa(Y) \setminus \{X\} \ \wedge \ P^{(\{X\})}(Y) = P^{(\varnothing)}(Y)$$

when the intervention target $\{X\}$ is in $\mathcal{I}$.

Definition 5.11 *($\mathcal{I}$-contradictory edge).* An edge $(X \to Y)$ in G is *$\mathcal{I}$-contradictory* if at least one of the following conditions holds:

- $\exists \mathbf{S} \subset Ne(Y) \setminus \{X\}$ such that $\forall \mathbf{I} \in \mathcal{I}_{X \setminus Y}$ we observe $P^{(\mathbf{I})}(Y \mid \mathbf{S}) = P^{(\varnothing)}(Y \mid \mathbf{S})$, or
- $\forall \mathbf{S} \subset Ne(X) \setminus \{Y\}$ such that $\exists \mathbf{I} \in \mathcal{I}_{Y \setminus X}$ we observe $P^{(\mathbf{I})}(X \mid \mathbf{S}) \neq P^{(\varnothing)}(X \mid \mathbf{S})$.

$\mathcal{I}$-contradictory edges are particularly of interest since they differ among interventional equivalence classes, i.e. they violate the $\mathcal{I}$-Markov property, highlighting the possibility for a consistent exploitation during the discovery procedure.

5.4. Learning with interventions

Sometimes researchers want to observe the effect of an intervention on one single variable at a time, but there are settings in which this is not possible or it is inconvenient. Therefore, multi-variable interventions must be addressed as a special case of a generic intervention target.

Single vs. multi-variable interventions. When each intervention target contains a single variable at a time, the number of experiments needed to collect enough evidence to identify the causal graph is $n - 1$, with n the number of variables [92]. Indeed, if one intervention would enable the identification of the causal edges incoming into the only variable contained in the intervention target, then the n-th intervention would be redundant.

In the case of intervention targets with more than one variable, only $\lfloor \log(n) \rfloor + 1$ interventions[5] are necessary and sufficient in the worst case scenario [92], where the causal graph is the complete graph. Since this worst case is improbable, $O(\log \log(n))$ can be achieved as lower bound with high probability in the multi-variable setting with a randomized intervention scheme [93], that is, it is possible to plan the experimental design in advance to minimize the number of interventions.

Unknown intervention targets. An other problem that one may face during structural learning with interventional data is the uncertainty related to the interventional targets [54–56]. There are scenarios in which it is known that an intervention has been performed, but it is unclear which is the exact set of variables that has been affected by such intervention. In this case, an additional layer of complexity is added in order to properly handle the less informative setting of *unknown intervention targets*.

5.5. Interventional algorithms

5.5.1. Interventional GES (GIES)

By leveraging the similarity between observational causal graphs and their interventional counterparts, authors in [53] proposed a generalization of the GES algorithm to the interventional setting. This new score-based variant, called Greedy Interventional Equivalence Search (GIES), follows the same two step approach of the original procedure, traversing the search space using forward- and backward-phases, until a (local) maximum score is reached.

A major contribution of this work is related to the formalization of the interventional setting. Indeed, while the algorithm itself does not differ significantly from the observational one in terms of overall design, the performance improvements are relevant, as expected by transitioning from the first to the second layer of the causal hierarchy. This is an interesting example of how observational techniques can be adapted to the interventional setting with ease, once the theoretical aspects of both the intervention distribution and the intervention targets are addressed properly.

5.5.2. Interventional Greedy Permutation (IGSP)

While GIES focuses its attention on perfect interventions, a first extension to general interventions is presented in [29], with the *Interventional Greedy Sparsest Permutations* (IGSP), an interventional variant of the GSP [94]. In this case, the *greedy* approach consists in the optimization of a score function, coupled with a permutation-based strategy that guides the traversal of the $\mathcal{I}$-MECs space.

Formally, let ρ be a permutation of vertices of a causal graph G. The space on which such permutation lays is a polytope called *permutahedron*. A possible representation of this mathematical object is indeed another graph, where each vertex corresponds to a permutation ρ and each edge between two permutations encodes a transposition of the vertices. The goal of a permutation-based causal discovery algorithm is to find a permutation ρ^*, consistent with the topological order of the true causal graph G^*, that optimizes a given score function. The search procedure traverses the permutahedron using a depth-first approach starting from an initial permutation ρ. For each permutation τ visited, if G_τ yields a better score than G_ρ then ρ is set to τ. The traversal is restarted from the updated ρ, until no such τ is found.

In order to leverage the advantages of the interventional data, IGSP limits the vertices transposition to the neighbors that are connected by $\mathcal{I}$-covered edges, restricting the search space to permutations that are coherent with the intervention

[5] Where $\lfloor x \rfloor$ denotes the *floor* function that maps a real number x to the greatest integer less than or equal to x.

targets. An other characteristic of this search strategy is given by the prioritization of $\mathcal{I}$-covered edges that are also $\mathcal{I}$-contradictory, given that they represent a transition of $\mathcal{I}$-MEC, which could lead to an improvement of the total score.

An extended version of this algorithm, named *UT-IGSP*, has been presented in [54] in order to tackle the *unknown target* scenario. The main contribution of this work is linked to the new definition of $\mathcal{I}$-covered edges in light of partially unknown intervention targets.

IGSP (and later UT-IGSP) has been compared to GIES under different conditions, showing that the former achieves better performances than the latter when the dimensionality of the problem is limited (i.e. lower than 10 vertices). This limit is coherent with others traversal-based approaches: although GIES is not consistent in general, its score function is more efficient in pooling together the various interventional datasets when it comes to high-dimensionality spaces.

5.5.3. Joint Causal Inference with FCI (FCI-JCI)

Another formal approach, similar to the one introduced in the previous subsection, is presented under the name of *Joint Causal Inference* (JCI) [28]. This method aims to pool together multiple observations collected during different experiments (i.e. *contexts*), hence, the name *joint* causal inference.

In this framework, the set of observed variables is split into two disjoint sets: *system variables* $\mathbf{X}$ and *context variables* $\mathbf{C}$. While the former set contains the variables that have been observed during an experiment, the latter set describes under which conditions such system has been observed, following the classical distinction between endogenous and exogenous variables, respectively.

Context variables can be used as *intervention variables*, even if this might not always be the case: here the term is related to the notion of *change of context*, which is a broader scope than simply intervene on the system. Doing so, it is possible to obtain a more flexible representation of the system of interest, where external forces are represented as internal characteristics of a *meta-system*. This approach relaxes the boundary between experiments performed under different conditions, allowing researchers to join data with a coherent causal description.

Before diving into JCI itself, there are a couple of assumptions that can be (optionally) taken into consideration to understand the purpose of the entire context framework:

0. The underlying mechanism that generates the data is represented by a SCM M, where the observed variables are split in system variables and context variables.
1. No system variable is cause of any context variable, i.e. *exogeneity assumption*.
2. No system variable is confounded with any context variable, i.e. *randomized context*.
3. Let $G_{\mathbf{C}}$ be the *context* graph induced by the context variables $\mathbf{C}$ over the causal graph G associated with the SCM M. For each pair of context variables (C_i, C_j) in the context graph the following holds true:

$$(C_i \leftrightarrow C_j) \in G_{\mathbf{C}} \wedge (C_i \rightarrow C_j) \notin G_{\mathbf{C}} \tag{5.9}$$

that is, no context variable is a direct cause of another context variable, but there is a hidden confounder between each pair of context variables, i.e. *generic context*.

While assumptions (0), (1) and (2) are usually considered mild in the interventional setting, assumption (3) might need to be clarified further: if the goal of the causal discovery is to disentangle the causal relationships *using* the context variables as guidance, rather than *focusing on* the connections of the context graph, then assumption (3) can be enforced if (1) and (2) were also assumed. This approach allows the algorithm to restrict the search space to the graphs that satisfy this last assumption, speeding-up the learning process.

The generic JCI procedure can be *adapted to any* observational causal discovery algorithm by following four steps: i) add the context variables, ii) pool data together by setting the values of the context variables, iii) address faithfulness violations between contexts, if any, iv) execute the selected observational learning algorithm. Authors provide reference adaptations for multiple algorithms, such as FCI.

The FCI-JCI variant is particularly of interest, provided that it inherits the strong points of FCI in the causally insufficient setting. Various combinations of the three assumptions were tested, showing that FCI123 (i.e. all three assumptions made) is less accurate in general, but significantly faster than others solutions, allowing its application in more complex scenarios with a sensible number of variables.

5.5.4. Unknown intervention targets using Ψ-FCI

Authors in [55] adapted both PC and FCI algorithms to the causal discovery setting under imperfect interventions with unknown intervention targets. The fundamental contribution of this work is the extension of the $\mathcal{I}$-MEC to a more general Ψ-MEC that is capable of representing intervention graphs with unknown intervention targets.

The key idea is that a pair of intervention targets $\mathbf{I}, \mathbf{J} \in \mathcal{I}$ can be used to identify a unique interventional mechanism that encompasses both targets. Let G be a causal graph and $\mathcal{I}$ an intervention family. The induced set of interventional probability distributions $P^{(\mathcal{I})} = \{P^{(\mathbf{I}_0)}, P^{(\mathbf{I}_1)}, \dots, P^{(\mathbf{I}_n)}\}$ satisfies the Ψ-Markov property if the following holds true for any $\mathbf{Y}$, $\mathbf{Z}$ and $\mathbf{W}$ disjoint subsets of variables of $\mathbf{V}$:

1. Insertion and deletion of observations:

$$P^{(\mathbf{I})}(\mathbf{Y} \mid \mathbf{Z}, \mathbf{W}) = P^{(\mathbf{I})}(\mathbf{Y} \mid \mathbf{W}) \tag{5.10}$$

if $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{W})$ holds true in G for all $\mathbf{I} \in \mathcal{I}$,

2. Invariance of interventions:

$$P^{(\mathbf{I})}(\mathbf{Y} \mid \mathbf{W}) = P^{(\mathbf{J})}(\mathbf{Y} \mid \mathbf{W}) \tag{5.11}$$

if $\left(\mathbf{Y} \perp\!\!\!\perp \mathbf{K} \mid (\mathbf{W} \setminus \mathbf{W_K})\right)$ holds true in $G_{\overline{\mathbf{W_K}\mathbf{R}(\mathbf{W})}}$ for all $\mathbf{I}, \mathbf{J} \in \mathcal{I}$, where $\mathbf{K}$ is the symmetric difference of $\mathbf{I}$ and $\mathbf{J}$, $\mathbf{W_K} = \mathbf{W} \cap \mathbf{K}$, $\mathbf{R} = \mathbf{K} \setminus \mathbf{W_K}$ and $\mathbf{R}(\mathbf{W}) = \mathbf{R} \setminus An(\mathbf{W})$ w.r.t. G.

While Equation (5.10) is essentially derived from observational Markov equivalence, Equation (5.11) is related to the distributional invariances across pairs of intervention targets w.r.t. the associated intervention graph. Indeed, if $\mathbf{I}$ and $\mathbf{J}$ are the *true* intervention targets for $P^{(\mathbf{I})}$ and $P^{(\mathbf{J})}$, they must satisfy the invariance for the interventional distributions when separation holds in a given intervention graph.

Moreover, the Ψ-Markov property does not require any assumption about the experimental setting in which such interventions are performed. Specifically, it could happen that a subset of experiments were not carried out exactly in the same way, e.g. not in a *controlled* environment. Therefore, even if interventions targets were known a priori, Ψ-Markov would still be more general than the $\mathcal{I}$-Markov property.

The authors then recast the augmented graph proposed by [24,33], adding a set of *utility* vertices that are analogous to the context vertices proposed by [28]. Therefore, the output of the former can be compared to the latter using the related augmented graph, showing that the accuracy of the edge orientations recovered by their Ψ-FCI variant is superior than the one proposed by FCI-JCI.

5.5.5. backShift

Continuing in the unknown targets setting, the *backShift* [56] algorithm is a causal discovery approach that recovers linear (possibly cyclical) models under causal insufficiency. It focuses on *shift interventions* with unknown targets, a subset of imperfect interventions where the effect of such perturbation yields a fixed shift of the intervened variable. Both the targets and the shift value can be estimated from the data.

The key idea of this technique is to represent the target SCM M as:

$$(\mathbf{I} - \mathbf{B})\mathbf{x} = \mathbf{c} + \mathbf{e} \tag{5.12}$$

where $\mathbf{x}$ is a random vector, $\mathbf{B}$ is the adjacency matrix of the casual graph associated to M, $\mathbf{e}$ is the noise vector and $\mathbf{c}$ is the random shift vector that models the shift intervention on the system. Then, a joint matrix diagonalization is applied to the differences between the covariance matrices $\Delta\mathbf{\Sigma}$ of each experiment $\mathbf{I} \in \mathcal{I}$:

$$\widetilde{\mathbf{D}} = \underset{\mathbf{D} \in \mathcal{D}}{\operatorname{argmin}} \sum_{\mathbf{I} \in \mathcal{I}} \mathcal{L}(\mathbf{D} \Delta \mathbf{\Sigma}^{(\mathbf{I})} \mathbf{D}^T) \tag{5.13}$$

where $\mathbf{D} = \mathbf{I} - \mathbf{B}$, $\mathcal{L}$ is the sum-of-squared loss function, $\mathcal{I}$ the family of targets.

This approach assumes that data represent observation at the equilibrium, the $\mathbf{D}$ matrix is invertible and the *cycle product* [56] is strictly smaller than one. Moreover, noises, interventions between variables and between experiments are assumed to be uncorrelated.

Authors compare their solution to the LiNG observational alternative, taking advantage of the interventional asymmetries arising from the additional information contained in the data. The results show that *backShift* is capable of dealing with both interventions and latent variables under mild assumptions, outperforming LiNG in both the observational and interventional settings. Moreover, the computational complexity is $O(|\mathcal{I}| \cdot n^2 \cdot m)$, with n representing the number of variables and m representing the sample size, which allows its application in high-dimensional settings.

5.5.6. bcause+

An extension of the *bcause* algorithm to interventional data, called *bcause+*, is proposed in [57]. When multiple experimental datasets are available, the core-base estimation of the lower bound of each branch of the exact search can be improved by taking into account the variables affected by the intervention.

In particular, the graphical separation statements checks by the observational variant (using either d-separation or σ-separation) are extended to consider the constraints induced by an intervention target. By assuming the absence of edges oriented into vertices that are part of an intervention target, the search procedure can avoid to check for separation, e.g. in case of perfect interventions. In this sense, intervention targets can be used to derive linear programming constraints by considering the subsets of intervened variables that affect the separation statements.

The improved version of the previous algorithm is also evaluated on non-linear cyclic causal models, showing its capability to deal with non-linear relationships. However, even with the added constraints, the exponentially-increasing execution time prohibits its application in high-dimensional contexts, which is a well known limitation for exact search methods.

Table 4
Static datasets by source, type and availability. Each dataset originates from (R)eal-world or (S)ynthetic experiments, which may contain (O)bservational, (I)nterventional or (M)ixed data, i.e. both observational and interventional instances.

Dataset	Source	Type	URL
Tuebingen [95]	R	O	Here
CausalWorld [96]	S	I	Here
Sachs [97]	R	M	Here
Klein [98]	R	M	Here
Perturb-Seq [100]	R	I	Here
SynTReN [101]	S	O	Here
DREAM4 [102]	S	M	Here

5.5.7. Differentiable Causal Discovery with Interventions (DCDI)

Under regularity assumption, authors in [58] propose a general differentiable causal discovery algorithm that is capable of learning causal graphs from interventional data with both perfect and imperfect interventional targets, even in the case of unknown interventions.

The key idea of this algorithm is to maximize a score function defined as follows:

$$\mathcal{S}_{\mathcal{I}}(G) = \sup_{\phi} \sum_{\mathbf{I}\in\mathcal{I}} \mathbb{E}_X \log f^{(\mathbf{I})}(X; \mathbf{B}, \mathbf{R}, \phi) - \lambda|G| \tag{5.14}$$

where ϕ are the weights of the estimator used to maximize the score function (i.e. neural networks in this case), X follows the interventional distribution $P^{(\mathbf{I})}$, $f^{(\mathbf{I})}$ is the interventional density function, $\mathbf{B}$ the binary adjacency matrix of G, $\mathbf{R}$ the binary interventional matrix (i.e. $R_{ij} = 1$ if $X_i \in \mathbf{I}_j$) and λ a penalty coefficient.

Essentially, the score function is built upon the conditional interventional distribution to recover the invariant edges across interventions. In fact, vertices that are not in any intervention target are characterized by a conditional probability distribution that is invariant across interventions, as for conservative families of interventions. Relying on conditional invariance, the causal graph $\hat{G} = \text{argmax}_{G\in\mathbb{G}} \mathcal{S}_{\mathcal{I}}(G)$ is $\mathcal{I}$-equivalent (Subsection 5.8) to the true graph G^*, for $\lambda > 0$ small enough. In case of unknown intervention targets, an additional $-\lambda_R|\mathcal{I}|$ regularization term is added to the score function.

The DCDI algorithm has been tested against IGSP and GIES with known interventions and JCI-PC and UT-IGSP for unknown interventions, showing marginal advantages in terms of structural recovery. As for others continuous optimization methods [47], the major strength is represented by its scalability: it takes $O(n^3)$, with n the number of variables, to compute the matrix exponential during each training step, making it the only causal discovery algorithm that supports non-linear relationships in the interventional setting in a high-dimensional setting.

6. Evaluation and tuning

This section tackles the evaluation and tuning step typical of any practical application. A collection of reference datasets is listed in Table 4, both real-world and synthetic generated ones, serving as benchmarking resources for discovery methods. In order to evaluate different solutions resulting from a set of configurations (i.e. hyperparameters) we report comparison metrics found in the specialized literature, both in terms of structure and entailed causal statements. Finally, tuning strategies and software packages are explored as support for new developed techniques.

6.1. Evaluation datasets

Cause-effect pairs (tuebingen). Ever-growing dataset [95,45] designed to benchmark discovery algorithms against bi-variate settings with known ground truth. The latest version reported by the change-log (December 20, 2017) includes 108 pairs. Each pair is composed by a data file with two columns (cause and effect, respectively), a short description of the data sampling procedure, and a 2D scatter plot.

Robotic manipulation (CausalWorld). Simulator [96] for causal structure and transfer learning in a robotic manipulation environment. The environment is a simulation of an open-source robotic platform capable of constructing 3D shapes from a given set of blocks.

Single-cell flow cytometry (Sachs). Flow cytometry measurements [97] of 11 proteins and phospholipids. The dataset is split into different experiments, with nine stimulatory or inhibitory interventions. The study compares new learned model against ground truth obtained by reference literature on signaling networks with intervention points.

Single-cell RNA-sequencing (Klein). Single-cell RNA-sequencing (scRNA-seq) dataset [98] of mouse embryonic stem cells after leukemia inhibitory factor (LIF) withdrawal. The ground truth model is obtained by querying the TRRUST database [99] for the related causal relationships.

Single-cell gene expression (perturb-seq). Measurements of gene expression [100] composed by 992 observational and 13.435 interventional observations from eight close-to-perfect interventions, each corresponding to a gene deletion using the CRISPR/Cas9 technique applied to bone marrow-derived dendritic cells.

Synthetic gene expression (SynTReN). Network generator [101] that creates synthetic transcriptional regulatory networks. The models are paired with kinetics simulations in order to sample gene expression data that approximate observed experimental data.

Synthetic mRNA expression (DREAM4). The DREAM4 challenge [102] provides five datasets simulated from five biologically plausible gene regulatory networks with 10 genes [103]. Each dataset is composed by both observational and interventional data sampled by applying small random noise perturbations, single-gene knockdowns and single-gene knockouts, resulting in time-series with unknown interventions.

6.2. Evaluation metrics

In the context of causal discovery, the definitions of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) have the same interpretation of the metrics referred to a binary classifier, which tries to predict the edge orientation.

Adjacency precision (AP) & recall (AR). A first set of evaluation metrics for graphical models is made of the adjacency precision (AP) and adjacency recall (AR) [104]. These metrics are computed as the ratio between the number of correctly predicted adjacent vertices over the total predicted ones for AP, and true predicted ones for AR. Formally, once the confusion matrix associated with the presence of edges is computed, the two metrics are defined as follows:

$$AP = \frac{TP}{TP + FP} \tag{6.1}$$

$$AR = \frac{TP}{TP + FN} \tag{6.2}$$

Arrowheads precision (AHP) & recall (AHR). While metrics related to adjacency can deliver insights on the general structure (i.e. skeleton) quality, arrowheads metrics [61,104] focus on highlighting inferred relationships performance. This class of metrics is particularly useful when there are multiple arrowhead marks that encode different causal statements, such as in PAGs. Here, classical adjacency metrics fail to account for invariant marks that might be interpreted as a head or a tail, overestimating the algorithm performance.

As for adjacency metrics, arrowheads precision (AHP) and recall (AHR) are defined as the ratio between correctly predicted arrowheads over total predicted arrowheads, and correctly predicted arrowheads over true arrowheads:

$$AHP = \frac{TP}{TP + FP} \tag{6.3}$$

$$AHR = \frac{TP}{TP + FN} \tag{6.4}$$

where TP, FP and FN refer to the confusion matrix entries computed over the predicted arrowheads, not only the presence/absence of an edge.

Structural Hamming Distance (SHD). It measures the differences between two graphical models in terms of their edges. Formally, let G and H be two graphs and $\mathbf{E}(G, U)$ the symmetric difference between the edge sets $\mathbf{E}(G)$ and $\mathbf{E}(U)$, the SHD [77] counts the number of necessary operations to transform G into H:

$$SHD(G, H) = \sum_{(X,Y),\, X<Y}^{\mathbf{v}^2} \begin{cases} 1 & (X, Y) \in \mathbf{E}(G, U), \\ 1 & (Y, X) \in \mathbf{E}(U, G), \\ 0 & \text{otherwise}, \end{cases} \tag{6.5}$$

where the allowed operations consist in addition, deletion and reversal of an edge.

Structural Intervention Distance (SID). It is a pre-metric defined over the interventional distributions. Formally, the SID [105] counts the number of wrongly inferred interventional distributions. This measure relies on the notion of *adjustment set* [88] and it is strongly related to the SHD.

6.3. Parameters tuning

Strategies to perform parameters tuning are rarely found in surveys, even tough casual discovery algorithms may have multiple parameters that regulate the search procedure. Here, we report three general and flexible practices described in the specialized literature that can be applied to any technique described so far.

Minimizing model complexity (BIC & AIC). A first approach for parameters tuning is related to model complexity. The goal is to find the parameters configuration that minimizes the complexity of the associated causal graph. As a measure of complexity one can rely on the Akaike Information Criterion (AIC) (Equation (3.5)) or the Bayesian Information Criterion (BIC) (Equation (3.6)). This tuning strategy is particularly effective when coupled with score-based approaches that are able to exploit the same function, allowing to reuse the intermediate scores for a faster evaluation. The most general form of model complexity minimization is implemented as a grid search over all parameters configurations for the ranges.

Stability Approach to Regularization Selection (StARS). The StARS [106] approach is based on selecting the parameters configuration that minimizes the graph *instability* when small perturbations are applied to the data. The instability of an edge is the probability of presence of said edge when the causal graph is learned from a subsample of the data (without replacement). Hence, the graph instability of a given parameters' configuration h is the average of the edge instabilities computed w.r.t. h. In order to avoid configurations that lead to trivial graphs, e.g. the empty graph or the complete graph, the authors introduce a β parameter that acts as a threshold for the acceptable level of instability. In the end, this method measures the sensitivity of a specific parameters' configuration h as a function of the underlying data distribution.

Out-of-sample Causal Tuning (OCT). While previous approaches focus on metrics related to the causal structure alone, authors in [107] propose to employ the resulting model for its prediction capabilities, reducing the problem into an evaluation of a predictor. This approach works in a out-of-sample fashion, hence the name *Out-of-sample Causal Tuning* (OCT). The main advantages of such method are i) the lack of parametric assumptions about the distribution of the data, and ii) the generalization to cases where the BIC and AIC scores are not defined, i.e. discrete models with hidden variables.

6.4. Software packages

Stable and reliable implementations of discovery methods are fundamental to achieve reproducibility of the experimental results. In the following paragraphs, a list of notable tools is presented.

Causal Discovery Toolbox (CDT). The Causal Discovery Toolbox [108] is a Python front-end that acts as a bridge between different subpackages, pooling together multiple discovery algorithms. For example, one may find constraint-based algorithms such as PC, Max-Min Parents & Children (MMPC) [77], score-based algorithms as GES and variants (GIES), and non linear approaches as LiNGAM, Causal Additive Models (CAM) [109], and others.

bnlearn. The bnlearn [110] package is an R package developed for bayesian inference and structural learning. While both PC and MMPC are implemented, algorithms such as Incremental Association Markov Blanket (IAMB) [111] and its variants are present too. Moreover, the underlying implementation is well suited for large scale application due to the optimized support for parallel computing [112].

pcalg. The pcalg [113] package is a R utility for causal discovery and causal inference using graphical models. The algorithms provided here are PC and variants (CPC, PC Select), FCI and variants (RFCI [114], Anytime FCI [115], Adaptive Anytime FCI [114,115], FCI-JCI), GES and variants (AGES [42], ARGES, GIES) and LiNGAM. Given the wide variety of FCI-based algorithms and the integrated tools for causal inference, this package is particularly well suited for causal insufficient settings.

TETRAD. While previous packages were intended for command line usage, TETRAD [116] is a causal discovery package developed in Java with a graphical user interface. It follows a pipeline paradigm, where the user can drag & drop boxes from the side bar and connect them together to form data pipelines. The user can choose from a wide range of options, such as PC (PCStable, CPC & CPCStable [39], PCMax), FCI (RFCI, RFCI-BSC [117], GFCI), GES (FGES, FGES-MB, IMaGES), LiNGAM, and others. Given the simplicity of the interface, it is well suited for researchers with limited programming experience.

7. Practical applications

7.1. Causal discovery in economics

Emissions, production and energy use. Authors in [118] explore the interactions between growth in CO_2 emissions, economic production and energy use, both at the global and multi-regional levels over the period 1990–2014. In order to recover the causal relationship between variables, a modified version of the PC algorithm for time-series is used.

Using PC for multi-variate analysis in an economic application lead to a high-level understanding of the influences present in the global economic cycle. Moreover, authors highlight the advantages of applying causal discovery approaches w.r.t. other methodologies, e.g. "Granger causality", especially with the selected algorithms. In fact, the PC algorithm is specifically designed to avoid conditioning on irrelevant variables, resulting in larger effect size and lower dimensionality.

The output of the discovery step showed that CO_2 emissions, energy and economic activity are linked by a set of non-linear dependencies. At the global level, this graph suggests that a too rapid transition to net-zero emissions in the energy sector may hinder the global economic growth. When the regional level is taken into account, it is shown that regions are fully integrated into the system, which argues for coordinated policies across regions.

7.2. Causal discovery in medicine

Alzheimer's pathophysiology. Researchers in [119] employed data made available by the *Alzheimer's Disease Neuroimaging Initiative* (ADNI) coupled with biological markers and clinical assessment to study the biological mechanism behind the Alzheimer's Disease. Two causal discovery algorithms (FCI and FGES) were compared against the *gold standard* graph retrieved from literature.

Moreover, performing the discovery step alternatively under causal sufficient and insufficient assumptions provides useful hints to isolate the influence of unmeasured external variables, e.g. the presence of a bi-directed edge instead of a directed one. These differences in the recovered structures guide researches during the interpretation of the collected data, and ease the evaluation of competing hypotheses that are consistent with the causal graph.

The methods were executed both with and without trivial background knowledge, e.g. patient's age is not affected by any biomarker. A significant improvement was observed with the addition of the knowledge base. Finally, longitudinal data were included, discovering more edges and removing the incorrect ones. The performance of the constraint-based approach was lower and less stable across the bootstrap samples than the score-based one.

Unmet treatments in schizophrenia. Authors in [120] selected the GFCI algorithm to identify the causes of functional outcomes of patients affected by schizophrenia during the *critical window* for early intervention. The algorithm was applied to the *Recovery After an Initial Schizophrenia Episode Early Treatment Program* (RAISE-ETP) trial at two time-points, i.e. baseline and after 6-months. Social and occupational functioning metrics were derived from the *Quality of Life Scale* (QLS). The retrieved causal graph was used to build a SCM in order to quantify the magnitude of the effects.

Researchers explicitly selected this approach in order to rule-out direct and indirect effects of latent variables, while taking advantage of the hybrid structure of the algorithm to speed-up the discovery step. In fact, the GFCI algorithm leverages an initial phase that exploits the score-based approach described in FGES, which inherently caches intermediate evaluations of the score function to reduce the execution time.

The estimated effects shed light over the interaction between both social and occupational functioning with the socio-affective capacity, which in turn affects the motivation of the subject. Moreover, an extended analysis of time dependencies revealed several causal cycles over the 6-months time-frame.

7.3. Causal discovery in psychology

Alcohol use and anxiety disorder. Psychopathology researchers in [121] used graphical modeling algorithms to identify causal relationships within and between manifestations of psychiatric disorders. In this context, the GFCI algorithm was employed to identify symptoms that are part of a causal chain of "mediators". The main target of the study was to test whether drinking motivated by the goal of reducing negative affect (i.e. *drinking to cope*, DTC) served as a mediator in comorbid alcohol use and anxiety disorder.

One of the reasons why GFCI is well suited for this scenario is due to the capability of this algorithm to tackle the causally insufficient assumption. This property constitutes a major advantage, enabling researches to explore the effect of potential hidden variables during the assessment of the drinking-anxiety interplay in general.

The resulting graph showed that the most important causal influence of drinking was drinking craving, which was in turn influenced by DTC. However, there was still a degree of ambiguity in the direction of depression's associations with social anxiety and stress, suggesting the possible presence of latent variables.

8. Conclusions and discussion

8.1. A brief summary

Causal inference depends heavily on the construction of a reference model that crystallizes the acquired knowledge. To meet such requirement, causal discovery provides a set of methods that are able to recover a graphical description of the underlying mechanism, exploiting both collected data and prior knowledge. In this work, we presented a list of algorithms, evaluation criteria and software tools, trying to cover a wide range of theoretical and practical scenarios in a coherent and unified manner. Moreover, we compared these resources against challenging problems, such as the presence of unobserved variables, cyclical dependencies, non-linear relationships and unknown interventions, highlighting the strengths

and weaknesses of each solution. Finally, we reported a set of parameters tuning strategies and publicly available datasets to explore properly the described techniques and to test new ones.

8.2. Future directions

In terms of opportunities for future extensions, in this contribution we did not explore the implications of applying such method to time-series, which would add complexity. Indeed, the representation of the causal dependencies in time are different from the one expressed in a static scenario and deserves a separated discussion on its own, especially when combined with the other topics introduced during the discussion.

From a more general perspective, there is a set of issues that have yet to be solved in order to shorten the distance between these techniques and real-world applications. For instance, causal discovery in presence of missing data is an under explored field. In fact, imputing missing data when the missingness mechanism is *not at random* results in sub-optimal solutions due to the introduction of bias component. There have been some recent advances w.r.t. this topic [122,123], but these new contributions are restricted to constraint-based approaches only.

Another aspect that is worth mentioning is the application of causal discovery to heterogeneous data [124]. The "heterogeneity" of the data is referred to the collection process that happens under different observational contexts. Naïvely pooling different data sources together into a single one results in poor performances. Essentially, this problem stems from the presence of *distribution shifts*, which arise when context variables differ significantly, e.g. in case of environmental changes. Still, learning a causal representation in the presence of multiple heterogeneous sources is a relevant issue that has yet to be solved.

Finally, there are other scenarios in which causal discovery is limited by the current state-of-the-art, such as non-identical overlapping sets of variables [28,125], learning from streaming data [126] and federated learning [127].

CRediT authorship contribution statement

Alessio Zanga: Conceptualization, Writing – original draft, Writing – review & editing. **Elif Ozkirimli:** Supervision, Writing – review & editing. **Fabio Stella:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Alessio Zanga reports financial support was provided by F. Hoffmann-La Roche Ltd.

Data availability

No data was used for the research described in the article.

Acknowledgements

We thank the two anonymous reviewers for their insightful comments. We also thank professor Daniela Besozzi for her thoughtful suggestions.

Funding

Alessio Zanga was granted a Ph.D. scholarship by F. Hoffmann-La Roche Ltd.

References

[1] G.W. Imbens, Nonparametric estimation of average treatment effects under exogeneity: a review, Rev. Econ. Stat. 86 (2004) 4–29.
[2] G.C.-D. Psychiatric Genomics Consortium, et al., Identification of risk loci with shared effects on five major psychiatric disorders: a genome-wide analysis, Lancet 381 (2013) 1371–1379.
[3] J.L. Hill, Bayesian nonparametric modeling for causal inference, J. Comput. Graph. Stat. 20 (2011) 217–240.
[4] J. Pearl, Theoretical impediments to machine learning with seven sparks from the causal revolution, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018.
[5] C. Glymour, K. Zhang, P. Spirtes, Review of causal discovery methods based on graphical models, Front. Genet. 10 (2019) 1–15.
[6] M. Hernán, J. Robins, Causal Inference: What If, Chapman & Hall/CRC, Boca Raton, 2020.
[7] P. Spirtes, C.N. Glymour, R. Scheines, D. Heckerman, Causation, Prediction, and Search, MIT Press, 2000.
[8] E. Bareinboim, J.D. Correa, D. Ibeling, T.F. Icard, On pearl's hierarchy and the foundations of causal inference, in: Probabilistic and Causal Inference, 2022.
[9] M. Glymour, J. Pearl, N.P. Jewell, Causal Inference in Statistics: A Primer, John Wiley & Sons, 2016.
[10] A.R. Nogueira, A. Pugnana, S. Ruggieri, D. Pedreschi, J. Gama, Methods and Tools for Causal Discovery and Causal Inference, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2022, p. e1449.
[11] R. Guo, L. Cheng, J. Li, P.R. Hahn, H. Liu, A survey of learning causality with data: problems and methods, ACM Comput. Surv. 53 (2021) 1–37, https://doi.org/10.1145/3397269.

[12] R. Moraffah, P. Sheth, M. Karami, A. Bhattacharya, Q. Wang, A. Tahir, A. Raglin, H. Liu, Causal inference for time series analysis: problems, methods and evaluation, Knowl. Inf. Syst. (2021) 1–45.
[13] D. Malinsky, D. Danks, Causal discovery algorithms: a practical guide, Philos. Compass 13 (2018) e12470.
[14] M.J. Vowels, N.C. Camgoz, R. Bowden, D'ya like DAGs? A survey on structure learning and causal discovery, ACM Comput. Surv. (2021).
[15] A.R. Nogueira, J. Gama, C.A. Ferreira, Causal discovery in machine learning: theories and applications, J. Dyn. Games 8 (2021) 203–231.
[16] B. Schölkopf, F. Locatello, S. Bauer, N.R. Ke, N. Kalchbrenner, A. Goyal, Y. Bengio, Toward causal representation learning, Proc. IEEE 109 (2021) 612–634.
[17] J. Pearl, Bayesian networks, in: Encyclopedia of Social Network Analysis and Mining, 2nd ed., 2018.
[18] A. Massmann, P. Gentine, J. Runge, Causal inference for process understanding in Earth sciences, arXiv preprint arXiv:2105.00912, 2021.
[19] P. Spirtes, K. Zhang, Causal discovery and inference: concepts and recent methodological advances, in: Applied Informatics, vol. 3, 2016, pp. 1–28.
[20] S. Bongers, J.M. Mooij, From random differential equations to structural causal models: the stochastic case, arXiv:1803.08784, 2018.
[21] P.K. Rubenstein, S. Bongers, J.M. Mooij, B. Schölkopf, From deterministic odes to dynamic structural causal models, in: UAI, 2018.
[22] A. Shahbazinia, S. Salehkaleybar, M. Hashemi, Paralingam: parallel causal structure learning for linear non-Gaussian acyclic models, arXiv preprint arXiv:2109.13993, 2021.
[23] S. Shimizu, Lingam: non-Gaussian methods for estimating causal structures, Behaviormetrika 41 (2014) 65–98.
[24] S. Bongers, P. Forré, J. Peters, J.M. Mooij, Foundations of structural causal models with cycles and latent variables, arXiv:1611.06221, 2021.
[25] J.M. Mooij, T. Claassen, Constraint-based causal discovery using partial ancestral graphs in the presence of cycles, in: Conference on Uncertainty in Artificial Intelligence, in: PMLR, 2020, pp. 1159–1168.
[26] J. Pearl, Causal diagrams for empirical research, Biometrika 82 (1995) 669–688.
[27] T. Verma, J. Pearl, Equivalence and synthesis of causal models, in: Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence, UAI '90, Elsevier Science Inc., USA, 1990, pp. 255–270.
[28] J.M. Mooij, S. Magliacane, T. Claassen, Joint causal inference from multiple contexts, J. Mach. Learn. Res. 21 (2020) 99.
[29] K. Yang, A. Katcoff, C. Uhler, Characterizing and learning equivalence classes of causal DAGs under interventions, in: International Conference on Machine Learning, in: PMLR, 2018, pp. 5541–5550.
[30] S.A. Andersson, D. Madigan, M.D. Perlman, A characterization of Markov equivalence classes for acyclic digraphs, Ann. Stat. 25 (1997) 505–541.
[31] C. Meek, Causal inference and causal explanation with background knowledge, arXiv preprint arXiv:1302.4972, 2013.
[32] M. Kocaoglu, K. Shanmugam, E. Bareinboim, Experimental design for learning causal graphs with latent variables, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 7021–7031.
[33] P. Forré, J.M. Mooij, Markov properties for graphical models with cycles and latent variables, arXiv:1710.08775, 2017.
[34] J. Zhang, On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias, Artif. Intell. 172 (2008) 1873–1896.
[35] T. Richardson, P. Spirtes, Ancestral graph Markov models, Ann. Stat. 30 (2002) 962–1030.
[36] M. Drton, T.S. Richardson, Iterative conditional fitting for Gaussian ancestral graph models, in: UAI, 2004.
[37] J. Peters, D. Janzing, B. Schölkopf, Elements of Causal Inference: Foundations and Learning Algorithms, The MIT Press, 2017.
[38] S. Shimizu, P. Blöbaum, Recent Advances in Semi-Parametric Methods for Causal Discovery, 2020, pp. 111–130, https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119523024.ch5, https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119523024.ch5.
[39] D. Colombo, M.H. Maathuis, Order-independent constraint-based causal structure learning, arXiv:1211.3295, 2013.
[40] J.I. Alonso-Barba, J.A. Gámez, J.M. Puerta, et al., Scaling up the greedy equivalence search algorithm by constraining the search space of equivalence classes, Int. J. Approx. Reason. 54 (2013) 429–451.
[41] J. Ramsey, M. Glymour, R. Sanchez-Romero, C. Glymour, A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images, Int. J. Data Sci. Anal. 3 (2017) 121–129.
[42] P. Nandy, A. Hauser, M.H. Maathuis, High-dimensional consistency in score-based and hybrid structure learning, Ann. Stat. (2018).
[43] J.M. Ogarrio, P. Spirtes, J. Ramsey, A hybrid causal search algorithm for latent variable models, in: Conference on Probabilistic Graphical Models, in: PMLR, 2016, pp. 368–379.
[44] R. Cai, J. Qiao, K. Zhang, Z. Zhang, Z. Hao, Causal discovery from discrete data using hidden compact representation, Adv. Neural Inf. Process. Syst. 32 (2018) 2671–2679.
[45] N. Tagasovska, V. Chavez-Demoulin, T. Vatter, Distinguishing cause from effect using quantiles: bivariate quantile causal discovery, in: International Conference on Machine Learning, in: PMLR, 2020, pp. 9311–9323.
[46] P.O. Hoyer, S. Shimizu, A.J. Kerminen, Estimation of linear, non-Gaussian causal models in the presence of confounding latent variables, in: Probabilistic Graphical Models, 2006.
[47] X. Zheng, B. Aragam, P. Ravikumar, E.P. Xing, DAGs with no tears: continuous optimization for structure learning, arXiv preprint arXiv:1803.01422, 2018.
[48] T.S. Richardson, A discovery algorithm for directed cyclic graphs, arXiv preprint arXiv:1302.3599, 2013.
[49] G. Lacerda, P.L. Spirtes, J. Ramsey, P.O. Hoyer, Discovering cyclic causal models by independent components analysis, in: UAI, 2008.
[50] A. Hyttinen, P. Saikko, M. Järvisalo, et al., A core-guided approach to learning optimal causal graphs, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), International Joint Conferences on Artificial Intelligence, 2017.
[51] K. Rantanen, A. Hyttinen, M. Järvisalo, Discovering causal graphs with cycles and latent confounders: an exact branch-and-bound approach, Int. J. Approx. Reason. 117 (2020) 29–49.
[52] P. Forré, J.M. Mooij, Constraint-based causal discovery for non-linear structural causal models with cycles and latent confounders, arXiv preprint arXiv:1807.03024, 2018.
[53] A. Hauser, P. Bühlmann, Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs, J. Mach. Learn. Res. 13 (2012) 2409–2464.
[54] C. Squires, Y. Wang, C. Uhler, Permutation-based causal structure learning with unknown intervention targets, arXiv:1910.09007, 2020.
[55] A. Jaber, M. Kocaoglu, K. Shanmugam, E. Bareinboim, Causal discovery from soft interventions with unknown targets: characterization and learning, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 9551–9561, https://proceedings.neurips.cc/paper/2020/file/6cd9313ed34ef58bad3fdd504355e72c-Paper.pdf.
[56] D. Rothenhäusler, C. Heinze, J. Peters, N. Meinshausen, Backshift: learning causal cyclic graphs from unknown shift interventions, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 28, Curran Associates, Inc., 2015, https://proceedings.neurips.cc/paper/2015/file/92262bf907af914b95a0fc33c3f33bf6-Paper.pdf.
[57] K. Rantanen, A. Hyttinen, M. Järvisalo, Learning optimal cyclic causal graphs from interventional data, in: International Conference on Probabilistic Graphical Models, in: PMLR, 2020, pp. 365–376.
[58] P. Brouillard, S. Lachapelle, A. Lacoste, S. Lacoste-Julien, A. Drouin, Differentiable causal discovery from interventional data, arXiv:2007.01754, 2020.
[59] E. Castillo, J.M. Gutierrez, A.S. Hadi, Expert Systems and Probabilistic Network Models, Springer Science & Business Media, 2012.
[60] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.
[61] B. Andrews, J. Ramsey, G.F. Cooper, Learning high-dimensional directed acyclic graphs with mixed data-types, in: The 2019 ACM SIGKDD Workshop on Causal Discovery, in: PMLR, 2019, pp. 4–21.

[62] M. Tsagris, G. Borboudakis, V. Lagani, I. Tsamardinos, Constraint-based causal discovery with mixed data, Int. J. Data Sci. Anal. 6 (2018) 19–30.
[63] T.D. Le, T. Hoang, J. Li, L. Liu, H. Liu, S. Hu, A fast PC algorithm for high dimensional causal discovery with multi-core pcs, IEEE/ACM Trans. Comput. Biol. Bioinform. 16 (2019) 1483–1495, https://doi.org/10.1109/TCBB.2016.2591526.
[64] C. Li, X. Fan, On nonparametric conditional independence tests for continuous variables, Wiley Interdiscip. Rev.: Comput. Stat. 12 (2020) e1489.
[65] P.L. Spirtes, C. Meek, T.S. Richardson, Causal inference in the presence of latent variables and selection bias, arXiv:1302.4983, 2013.
[66] S. Lee, J. Correa, E. Bareinboim, Generalized transportability: synthesis of experiments from heterogeneous domains, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI Press, New York, NY, 2020.
[67] D.M. Chickering, Optimal structure identification with greedy search, J. Mach. Learn. Res. 3 (2002) 507–554.
[68] H. Akaike, A new look at the statistical model identification, IEEE Trans. Autom. Control 19 (1974) 716–723, https://doi.org/10.1109/TAC.1974.1100705.
[69] G. Schwarz, Estimating the dimension of a model, Ann. Stat. (1978) 461–464.
[70] D. Geiger, D. Heckerman, Learning Gaussian networks, in: Uncertainty Proceedings 1994, Elsevier, 1994, pp. 235–243.
[71] M. Scutari, An empirical-Bayes score for discrete Bayesian networks, in: Conference on Probabilistic Graphical Models, in: PMLR, 2016, pp. 438–448.
[72] C. Meek, Graphical Models: Selecting causal and statistical models, Ph.D. thesis, Carnegie Mellon University, 1997.
[73] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 465–471.
[74] D. Janzing, B. Schölkopf, Causal inference using the algorithmic Markov condition, IEEE Trans. Inf. Theory 56 (2010) 5168–5194.
[75] O. Stegle, D. Janzing, K. Zhang, J.M. Mooij, B. Schölkopf, Probabilistic latent variable models for distinguishing between cause and effect, Adv. Neural Inf. Process. Syst. 23 (2010) 1687–1695.
[76] P. Comon, Independent component analysis, a new concept?, Signal Process. 36 (1994) 287–314.
[77] I. Tsamardinos, L.E. Brown, C.F. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, Mach. Learn. 65 (2006) 31–78.
[78] T. Niinimaki, P. Parviainen, Local structure discovery in Bayesian networks, in: UAI, 2012.
[79] K. Natori, M. Uto, Y. Nishiyama, S. Kawano, M. Ueno, Constraint-based learning Bayesian networks using Bayes factor, in: Workshop on Advanced Methodologies for Bayesian Networks, Springer, 2015, pp. 15–31.
[80] M. Scutari, C.E. Graafland, J.M. Gutiérrez, Who learns better Bayesian network structures: accuracy and speed of structure learning algorithms, Int. J. Approx. Reason. 115 (2019) 235–253.
[81] P. Spirtes, Introduction to causal inference, J. Mach. Learn. Res. 11 (2010) 1643–1662, http://jmlr.org/papers/v11/spirtes10a.html.
[82] W.D. Berry, Nonrecursive Causal Models, vol. 37, Sage, 1984.
[83] M. Nagase, Y. Kano, Identifiability of nonrecursive structural equation models, Stat. Probab. Lett. 122 (2017) 109–117.
[84] P.L. Spirtes, Directed cyclic graphical representations of feedback models, arXiv:1302.4982, 2013.
[85] A. Hyttinen, F. Eberhardt, M. Järvisalo, Constraint-based causal discovery: conflict resolution with answer set programming, in: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI'14, AUAI Press, Arlington, Virginia, USA, 2014, pp. 340–349.
[86] S. Magliacane, T. Claassen, J.M. Mooij, Ancestral causal inference, arXiv:1606.07035, 2017.
[87] J. Pearl, D. Mackenzie, The Book of Why: The New Science of Cause and Effect, 1st ed., Basic Books, Inc., USA, 2018.
[88] I. Shpitser, J. Pearl, Complete identification methods for the causal hierarchy, J. Mach. Learn. Res. 9 (2008).
[89] F. Markowetz, S. Grossmann, R. Spang, Probabilistic soft interventions in conditional Gaussian networks, in: R.G. Cowell, Z. Ghahramani (Eds.), Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, PMLR, vol. R5, 2005, pp. 214–221, https://proceedings.mlr.press/r5/markowetz05a.html, reissued by PMLR on 30 March 2021.
[90] J. Tian, J. Pearl, Causal discovery from changes, arXiv:1301.2312, 2013.
[91] M. Kocaoglu, A. Jaber, K. Shanmugam, E. Bareinboim, Characterization and learning of causal graphs with latent variables from soft interventions, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019, https://proceedings.neurips.cc/paper/2019/file/c3d96fbd5b1b45096ff04c04038fff5d-Paper.pdf.
[92] F. Eberhardt, C. Glymour, R. Scheines, On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables, arXiv:1207.1389, 2012.
[93] H. Hu, Z. Li, A.R. Vetta, Randomized experimental design for causal graph discovery, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, vol. 27, Curran Associates, Inc., 2014, https://proceedings.neurips.cc/paper/2014/file/e53a0a2978c28872a4505bdb51db06dc-Paper.pdf.
[94] L. Solus, Y. Wang, C. Uhler, Consistency guarantees for greedy permutation-based causal inference algorithms, arXiv:1702.03530, 2021.
[95] J.M. Mooij, J. Peters, D. Janzing, J. Zscheischler, B. Schölkopf, Distinguishing cause from effect using observational data: methods and benchmarks, J. Mach. Learn. Res. 17 (2016) 1103–1204.
[96] O. Ahmed, F. Träuble, A. Goyal, A. Neitz, Y. Bengio, B. Schölkopf, M. Wüthrich, S. Bauer, CausalWorld: a robotic manipulation benchmark for causal structure and transfer learning, arXiv:2010.04296, 2020.
[97] K. Sachs, O. Perez, D. Pe'er, D.A. Lauffenburger, G.P. Nolan, Causal protein-signaling networks derived from multiparameter single-cell data, Science 308 (2005) 523–529.
[98] A.M. Klein, L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D.A. Weitz, M.W. Kirschner, Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells, Cell 161 (2015) 1187–1201.
[99] H. Han, J.-W. Cho, S. Lee, A. Yun, H. Kim, D. Bae, S. Yang, C.Y. Kim, M. Lee, E. Kim, et al., TRRUST v2: an expanded reference database of human and mouse transcriptional regulatory interactions, Nucleic Acids Res. 46 (2018) D380–D386.
[100] A. Dixit, O. Parnas, B. Li, J. Chen, C.P. Fulco, L. Jerby-Arnon, N.D. Marjanovic, D. Dionne, T. Burks, R. Raychowdhury, et al., Perturb-seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens, Cell 167 (2016) 1853–1866.
[101] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, K. Marchal, SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms, BMC Bioinform. 7 (2006) 1–12.
[102] P. Shannon, Dream4: Synthetic Expression Data for Gene Regulatory Network Inference from the 2009 DREAM4 Challenge, 2021, R package version 1.30.0.
[103] D. Marbach, T. Schaffter, C. Mattiussi, D. Floreano, Generating realistic in silico gene networks for performance assessment of reverse engineering methods, J. Comput. Biol. 16 (2009) 229–239.
[104] R. Scheines, J. Ramsey, Measurement Error and Causal Discovery, CEUR Workshop Proceedings, vol. 1792, NIH Public Access, 2016, p. 1.
[105] J. Peters, P. Bühlmann, Structural intervention distance for evaluating causal graphs, Neural Comput. 27 (2015) 771–799.
[106] H. Liu, K. Roeder, L.A. Wasserman, Stability approach to regularization selection (stars) for high dimensional graphical models, Adv. Neural Inf. Process. Syst. 24 (2) (2010) 1432–1440.
[107] K. Biza, I. Tsamardinos, S. Triantafillou, Tuning causal discovery algorithms, in: M. Jaeger, T.D. Nielsen (Eds.), Proceedings of the 10th International Conference on Probabilistic Graphical Models, in: Proceedings of Machine Learning Research, PMLR, vol. 138, 2020, pp. 17–28, https://proceedings.mlr.press/v138/biza20a.html.
[108] D. Kalainathan, O. Goudet, Causal discovery toolbox: uncover causal relationships in Python, arXiv:1903.02278, 2019.
[109] P. Bühlmann, J. Peters, J. Ernest, CAM: causal additive models, high-dimensional order search and penalized regression, Ann. Stat. 42 (2014) 2526–2556.

[110] M. Scutari, Learning Bayesian networks with the bnlearn R package, J. Stat. Softw. 35 (2010) 1–22, https://doi.org/10.18637/jss.v035.i03.
[111] I. Tsamardinos, C.F. Aliferis, A.R. Statnikov, E. Statnikov, Algorithms for large scale Markov blanket discovery, in: FLAIRS Conference, vol. 2, St. Augustine, FL, 2003, pp. 376–380.
[112] M. Scutari, Bayesian network constraint-based structure learning algorithms: parallel and optimized implementations in the bnlearn R package, J. Stat. Softw. 77 (2017) 1–20, https://doi.org/10.18637/jss.v077.i02.
[113] M. Kalisch, M. Mächler, D. Colombo, M.H. Maathuis, P. Bühlmann, Causal inference using graphical models with the R package pcalg, J. Stat. Softw. 47 (2012) 1–26, https://doi.org/10.18637/jss.v047.i11.
[114] D. Colombo, M.H. Maathuis, M. Kalisch, T.S. Richardson, Learning high-dimensional directed acyclic graphs with latent and selection variables, Ann. Stat. 40 (2012), https://doi.org/10.1214/11-AOS940.
[115] P. Spirtes, An anytime algorithm for causal inference, in: International Workshop on Artificial Intelligence and Statistics, in: PMLR, 2001, pp. 278–285.
[116] J.D. Ramsey, K. Zhang, M. Glymour, R.S. Romero, B. Huang, I. Ebert-Uphoff, S. Samarasinghe, E.A. Barnes, C. Glymour, TETRAD—a toolbox for causal discovery, in: 8th International Workshop on Climate Informatics, 2018.
[117] F. Jabbari, J. Ramsey, P. Spirtes, G. Cooper, Discovery of causal models that contain latent variables through Bayesian scoring of independence constraints, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2017, pp. 142–157.
[118] P.M. Addo, C. Manibialoa, F. McIsaac, Exploring nonlinearity on the CO2 emissions, economic production and energy use nexus: a causal discovery approach, Energy Rep. 7 (2021) 6196–6204, https://doi.org/10.1016/j.egyr.2021.09.026, https://www.sciencedirect.com/science/article/pii/S2352484721008313.
[119] X. Shen, S. Ma, P. Vemuri, G. Simon, Challenges and opportunities with causal discovery algorithms: application to Alzheimer's pathophysiology, Sci. Rep. 10 (2020) 1–12.
[120] K. Miley, P. Meyer-Kalos, S. Ma, D.J. Bond, E. Kummerfeld, S. Vinogradov, Causal pathways to social and occupational functioning in the first episode of schizophrenia: uncovering unmet treatment needs, Psychol. Med. (2021) 1–9, https://doi.org/10.1017/S0033291721003780.
[121] J.J. Anker, E. Kummerfeld, A. Rix, S.J. Burwell, M.G. Kushner, Causal network modeling of the determinants of drinking behavior in comorbid alcohol use and anxiety disorder, Alcohol. Clin. Exp. Res. 43 (2019) 91–97, https://doi.org/10.1111/acer.13914, https://onlinelibrary.wiley.com/doi/abs/10.1111/acer.13914, https://onlinelibrary.wiley.com/doi/pdf/10.1111/acer.13914.
[122] E.V. Strobl, S. Visweswaran, P.L. Spirtes, Fast causal inference with non-random missingness by test-wise deletion, Int. J. Data Sci. Anal. 6 (2018) 47–62.
[123] J. Witte, R. Foraita, V. Didelez, Multiple imputation and test-wise deletion for causal discovery with incomplete cohort data, arXiv preprint arXiv:2108.13331, 2021.
[124] B. Huang, K. Zhang, J. Zhang, J.D. Ramsey, R. Sanchez-Romero, C. Glymour, B. Schölkopf, Causal discovery from heterogeneous/nonstationary data, J. Mach. Learn. Res. 21 (2020) 1–53.
[125] S. Triantafillou, I. Tsamardinos, Constraint-based causal discovery from multiple interventions over overlapping variable sets, J. Mach. Learn. Res. 16 (2015) 2147–2205.
[126] K. Yu, X. Wu, H. Wang, W. Ding, Causal discovery from streaming features, in: 2010 IEEE International Conference on Data Mining, IEEE, 2010, pp. 1163–1168.
[127] E. Gao, J. Chen, L. Shen, T. Liu, M. Gong, H. Bondell, Federated causal discovery, arXiv preprint arXiv:2112.03555, 2021.