

Constraint-based Causal Discovery from Multiple Interventions over Overlapping Variable Sets

Sofia Triantafillou*

Ioannis Tsamardinos*

Institute of Computer Science

Foundation for Research and Technology - Hellas (FORTH)

N. Plastira 100 Vassilika Vouton

GR-700 13 Heraklion, Crete, Greece

STRIANT@ICS.FORTH.GR

TSAMARD@ICS.FORTH.GR

Abstract

Scientific practice typically involves repeatedly studying a system, each time trying to unravel a different perspective. In each study, the scientist may take measurements under different experimental conditions (interventions, manipulations, perturbations) and measure different sets of quantities (variables). The result is a collection of heterogeneous data sets coming from different data distributions. In this work, we present algorithm COMBINE, which accepts a collection of data sets over overlapping variable sets under different experimental conditions; COMBINE then outputs a summary of all causal models indicating the invariant and variant structural characteristics of all models that simultaneously fit all of the input data sets. COMBINE converts estimated dependencies and independencies in the data into path constraints on the data-generating causal model and encodes them as a SAT instance. The algorithm is sound and complete in the sample limit. To account for conflicting constraints arising from statistical errors, we introduce a general method for sorting constraints in order of confidence, computed as a function of their corresponding p-values. In our empirical evaluation, COMBINE outperforms in terms of efficiency the only pre-existing similar algorithm; the latter additionally admits feedback cycles, but does not admit conflicting constraints which hinders the applicability on real data. As a proof-of-concept, COMBINE is employed to co-analyze 4 real, mass-cytometry data sets measuring phosphorylated protein concentrations of overlapping protein sets under 3 different interventions.

1. Introduction

Causal discovery is an abiding goal in almost every scientific field. In order to discover the causal mechanisms of a system, scientists typically have to perform a series of experiments (interchangeably: manipulations, interventions, or perturbations). Each experiment adds to the existing knowledge of the system and sheds light to the sought-after mechanism from a different perspective. In addition, each measurement may include a different set of quantities (variables), when for example the technology used allows only a limited number of measured quantities.

However, for the most part, machine learning and statistical methods focus on analyzing a single data set. They are unable to make joint inferences from the complete collection of available heterogeneous data sets, since each one is following a different data distribution

*. Also in Department of Computer Science, University of Crete.

(albeit stemming from the same system under study). Thus, data sets are often analyzed in isolation and independently of each other; the resulting knowledge is typically synthesized ad hoc in the researcher’s mind.

The proposed work tries to automate the above inferences. We propose a general, constraint-based algorithm named COmbINE for learning causal structure characteristics from the integrative analysis of collections of data sets. The data sets can be heterogeneous in the following manners: they may be measuring different overlapping sets of variables \mathbf{O}_i under different hard manipulations on a set of observed variables \mathbf{I}_i . A hard manipulation on a variable I , corresponds to a Randomized Controlled Trial (Fisher, 1922) where the experimentation procedure completely eliminates any other causal effect on I (e.g., randomizing mice to two groups having two different diets; the effect of all other factors on the diet is completely eliminated).

What connects together the available data sets and allows their co-analysis is the assumption that *there exists a single underlying causal mechanism that generates the data*, even though it is measured with a different experimental setting each time. A causal model is plausible as an explanation if it simultaneously fits all data-sets when the effect of manipulations and selection of measured variables is taken into consideration.

COmbINE searches for the set of causal models that simultaneously fits all available data-sets in the sense given above. The algorithm outputs a summary network that includes all the variant and invariant pairwise causal characteristics of the set of fitting models. For example, it indicates the causal relations upon which all fitting models agree, as well as the ones for which conflicting explanations are plausible. As our formalism of choice for causal modeling, we employ Semi-Markov Causal Models (**SMCMs**). SMCMs (Tian and Pearl, 2003) are extensions of Causal Bayesian Networks (**CBNs**) that can account for latent confounding variables, but do not admit feedback cycles. Internally, the algorithm also makes heavy use of the theory and learning algorithm for Maximal Ancestral Graphs (**MAGs**) (Richardson and Spirtes, 2002).

The algorithm builds upon the ideas in Triantafillou et al. (2010) to convert the observed statistical dependencies and independencies in the data to path constraints on the plausible data generating structures. The constraints are encoded as a SAT instance and solved with modern SAT engines, exploiting the efficiency of state-of-the-art solvers. However, due to statistical errors in the determination of dependencies and independencies, conflicting constraints may arise. In this case, the SAT instance is unsolvable and no useful information can be inferred. COmbINE includes a technique for sorting constraints according to confidence: The constraints are added to the SAT instance in increasing order of confidence, and the ones that conflict with the set of higher-ranked constraints are discarded. The technique is general and the ranking score is a function of the p-values of the statistical tests of independence. It can therefore be applied to any type of data, provided an appropriate test exists.

COmbINE is empirically compared against a similar, recently developed algorithm by Hyttinen et al. (2013). The latter is also based on conversion to SAT and is able to additionally deal with cyclic structures, but assumes lack of statistical errors and corresponding conflicts. It can therefore not be directly applied to typical real problems that may generate such conflicts. COmbINE proves to be more efficient than Hyttinen et al. (2013) and scales to larger problem sizes, due to an inherently more compact representation of the path-

constraints. The empirical evaluation also includes a quantification of the effect of sample size, number of data-sets co-analyzed, and other factors on the quality and computational efficiency of learning. In addition, the proposed conflict resolution technique’s superiority is demonstrated over several other alternative conflict resolution methods. Finally, we present a proof-of-concept computational experiment by applying the algorithm on 5 heterogeneous data sets from Bendall et al. (2011) and Bodenmiller et al. (2012) measuring overlapping variable sets under 3 different manipulations. The data sets measure protein concentrations in thousands of human cells of the autoimmune system using mass-cytometry technologies. Mass cytometers can perform single-cell measurements with a rate of about 10,000 cells per second, but can currently only measure up to circa 30 variables per run. Thus, they seem to form a suitable test-bed for integrative causal analysis algorithms.

The rest of this paper is organized as follows: Section 2 presents the related literature on learning causal models and combining multiple data sets. Section 3 reviews the necessary theory of MAGs and SMCs and discusses the relation between the two and how hard manipulations are modeled in each. Section 4 is the core of this paper, and it is split in three subsections; presenting the conversion to SAT; introducing the algorithm and proving soundness and completeness; introducing the conflict resolution strategy. Section 5 is devoted to the experimental evaluation of the algorithm: testing the algorithm’s performance in several settings and presenting an actual case study where the algorithm can be applied. Finally, Section 6 summarizes the conclusions and proposes some future directions of this work.

2. Related Work

Methods for causal discovery have been, for the most part, limited to the analysis of a single data set. However, the great advancement of intervention and data collection technology has led to a vast increase of available data sets, both observational and experimental. Therefore, over the last few years, there have been a number of works that focus on causal discovery from multiple sources. Algorithms in that area may differ in the formalism the use to model causality or in the type of heterogeneity in the studies they co-analyze. In any case, the goal is always to discover the single underlying data-generating causal mechanism.

One group of algorithms focuses on combining observational data that measure overlapping variables. Tillman et al. (2008) and Triantafillou et al. (2010) both provide sound and complete algorithms for learning the common characteristics of MAGs from data sets measuring overlapping variables. Tillman et al. (2008) handles conflicts by ignoring conflicting evidence, while the method presented in Triantafillou et al. (2010) only works with an oracle of conditional independence. Tillman and Spirtes (2011) present an algorithm for the same task that handles a limited type of conflicts (those concerning p-values for the same pair of variables stemming from different data sets) by combining the p-values for conditional independencies that are testable in more than one data sets. Claassen and Heskes (2010b) present a sound, but not complete, algorithm for causal structure learning from multiple independence models over overlapping variables by transforming independencies into a set of causal ancestry rules.

Another line of work deals with learning causal models from multiple experiments. Cooper and Yoo (1999) use a Bayesian score to combine experimental and observational

data in the context of causal Bayesian networks. Hauser and Bühlmann (2012) extend the notion of Markov equivalence for DAGs to the case of interventional distributions arising from multiple experiments, and propose a learning algorithm. Tong and Koller (2001) and Murphy (2001) use Bayesian network theory to propose experiments that are most informative for causal structure discovery. Eberhardt and Scheines (2007) and Eaton and Murphy (2007a) discuss how some other types of interventions can be modeled and used to learn Bayesian networks. Hyttinen et al. (2012a) provides an algorithm for learning linear cyclic models from a series of experiments, along with sufficient and necessary conditions for identifiability. This method admits latent confounders but uses linear structural equations to model causal relations and is therefore inherently limited to linear relations. Meganck et al. (2006) propose learning SMCs by learning the Markov equivalence classes of MAGs from observational data and then designing the experiments necessary to convert it to a SMC.

Finally, there is a limited number of methods that attempt to co-analyze data sets measuring overlapping variables under different experimental conditions. In Hyttinen et al. (2012b) the authors extend the methods of Hyttinen et al. (2012a) to handle overlapping variables, again under the assumption of linearity. Hyttinen et al. (2013) propose a constraint-based algorithm for learning causal structure from different manipulations of overlapping variable sets. The method works by transforming the observed m -connection and m -separation constraints into a SAT instance. The method uses a path analysis heuristic to reduce the number of tests translated into path constraints. Causal insufficiency is allowed, as well as feedback cycles. However, this method cannot handle conflicts and therefore relies on an oracle of conditional independence. Moreover, the method can only scale up to about 12 variables. Claassen and Heskes (2010a) present an algorithm for learning causal models from multiple experiments; the experiments here are not hard manipulations, but general experimental conditions, modeled like variables that have no parents in the graph but can cause other variables in some of the conditions.

To the best of our knowledge, COMBINE is the first algorithm to address both overlapping variables and multiple interventions for acyclic structures without relying on specific parametric assumptions or requiring an oracle of conditional independence. While the limits of COMBINE in terms of input size have not been exhaustively checked, the algorithm scales up to networks of up to 100 variables for relatively sparse networks (maximum number of parents equals 5).

3. Mixed Causal Models

Causally insufficient systems are often described using Semi-Markov causal models (SMCMs) (Tian and Pearl, 2003) or Maximal Ancestral Graphs (MAGs) (Richardson and Spirtes, 2002). Both of them are **mixed graphs**, meaning they can contain both directed (\rightarrow) and bi-directed (\leftrightarrow) edges. We use the term **mixed causal graph** to denote both. In this section, we will briefly present their common and unique properties. First, let us review the basic mixed graph notation:

In a mixed graph \mathcal{G} , a path is a sequence of distinct nodes $\langle V_0, V_1, \dots, V_n \rangle$ s.t for $0 \leq i < n$, V_i and V_{i+1} are adjacent in \mathcal{G} . X is called a **parent** of Y and Y a **child** of X in \mathcal{G} if $X \rightarrow Y$ in \mathcal{G} . A path from V_0 to V_n is **directed** if for $0 \leq i < n$, V_i is a parent V_{i+1} . X is called a **ancestor** of Y and Y is called a **descendant** of X in \mathcal{G} if $X = Y$ in \mathcal{G} or there exists a

directed path from X to Y in \mathcal{G} . We use the notation $\mathbf{Pa}_{\mathcal{G}}(\mathbf{X})$, $\mathbf{Ch}_{\mathcal{G}}(\mathbf{X})$, $\mathbf{An}_{\mathcal{G}}(\mathbf{X})$, $\mathbf{Desc}_{\mathcal{G}}(\mathbf{X})$ to denote the set of parents, children, ancestors and descendants of nodes \mathbf{X} in \mathcal{G} . A **directed cycle** in \mathcal{G} occurs when $X \rightarrow Y \in \mathbf{E}$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$. An **almost directed cycle** in \mathcal{G} occurs when $X \leftrightarrow Y \in \mathbf{E}$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$. Given a path p in a mixed graph, a non-endpoint node V on p is called a **collider** if the two edges incident to V on p are both into V . Otherwise V is called a **non-collider**. A path $p = \langle X, Y, Z \rangle$, where X and Z are not adjacent in \mathcal{G} is called an **unshielded triple**. If Z is a collider on this path, the triple is called an **unshielded collider**. A path $p = \langle X \dots W, V, Y \rangle$ is called **discriminating** for V if X is not adjacent to Y and every node on the path from X to V is a collider and a parent of Y .

MAGs and SMCMs are graphical models that represent both causal relations and conditional independencies among a set of measured (observed) variables \mathbf{O} , and can be viewed as generalizations of causal Bayesian networks that can account for latent confounders. MAGs can also account for selection bias, but in this work we assume selection bias is not present.

sufficient. We call this hypothetical extended model the **underlying causal DAG**.

3.1 Semi-Markov Causal Models

Semi-Markov causal models (SMCMs), introduced by Tian and Pearl (2003), often also reported as Acyclic Directed Mixed Graphs (ADMGs), are causal models that implicitly model hidden confounders using bi-directed edges. A directed edge $X \rightarrow Y$ denotes that X is a *direct* cause of Y in the context of the variables included in the model. A bi-directed edge $X \leftrightarrow Y$ denotes that X and Y are confounded by an unobserved variable. Two variables can be joined by at most two edges, one directed and one bi-directed.

Semi-Markov causal models are designed to represent marginals of causal Bayesian networks. In DAGs, the probabilistic properties of the distribution of variables included in the model can be determined graphically using the criterion of d -separation. The natural extension of d -separation to mixed causal graphs is called m -separation:

Definition 1 (*m -connection, m -separation*) In a mixed graph $\mathcal{G} = (\mathbf{E}, \mathbf{V})$, a path p between A and B is **m -connecting** given (conditioned on) a set of nodes \mathbf{Z} , $\mathbf{Z} \subseteq \mathbf{V} \setminus \{A, B\}$ if

1. Every non-collider on p is not a member of \mathbf{Z} .
2. Every collider on the path is an ancestor of some member of \mathbf{Z} .

A and B are said to be **m -separated** by \mathbf{Z} if there is no m -connecting path between A and B relative to \mathbf{Z} . Otherwise, we say they are **m -connected** given \mathbf{Z} .

Let \mathcal{G} be a SMCM over a set of variables \mathbf{O} , \mathcal{P} the joint probability distribution (JPD) over the same set of variables and \mathcal{J} the independence model, defined as the set of conditional independencies that hold in \mathcal{P} . We use $\langle \mathbf{X}, \mathbf{Y} | \mathbf{Z} \rangle$ to denote the conditional independence of variables in \mathbf{X} with variables in \mathbf{Y} given variables in \mathbf{Z} . Under the Causal Markov (CMC) and Faithfulness (FC) conditions (Spirtes et al., 2001), *every m -separation present in \mathcal{G} corresponds to a conditional independence in \mathcal{J} and vice-versa*.

In causal Bayesian networks, every missing edge in \mathcal{G} corresponds to a conditional independence in \mathcal{J} , meaning there exists a subset of the variables in the model that renders the

two non-adjacent variables independent. Respectively, every conditional independence in \mathcal{J} corresponds to a missing edge in the DAG \mathcal{G} . This is not always true for SMCs. Figure 1 illustrates an example of a SMC where two non-adjacent variables are not independent given any subset of observed variables.

Evans and Richardson (2010, 2011) deal with the factorization and parametrization of SMCs for discrete variables. Based on this parametrization, score-based methods have also recently been explored Richardson et al. (2012); Shpitser et al. (2013), but are still limited to small sets of discrete variables. To the best of our knowledge, there exists no constraint-based algorithm for learning the structure of SMCs, probably due to the fact that the lack of conditional independence for a pair of variables does not necessarily mean non-adjacency. Richardson and Spirtes (2002) overcome this obstacle by introducing a causal mixed graph with slightly different semantics, the maximal ancestral graph.

3.2 Maximal Ancestral Graphs

Maximal ancestral graphs (MAGs) are **ancestral** mixed graphs, meaning that they contain no directed or almost directed cycles. Every pair of variables X, Y in an ancestral graph is joined by at most one edge. The orientation of this edge represents (non) causal ancestry: A bi-directed edge $X \leftrightarrow Y$ denotes that X does not cause Y and Y does not cause X , but (under the faithfulness assumption) the two share a latent confounder. A directed edge $X \rightarrow Y$ denotes causal ancestry: X is a *causal ancestor* of Y . Thus, if X causes Y (not necessarily directly in the context of observed variables) and they are also confounded, there is an edge $X \rightarrow Y$ in the corresponding MAG. Undirected edges can also be present in MAGs that account for selection bias. As mentioned above, we assume no selection bias in this work and the theory of MAGs presented here is restricted to MAGs with no undirected edges.

Like SMCs, ancestral graphs are also designed to represent marginals of causal Bayesian networks. Thus, under the causal Markov and faithfulness conditions, X and Y are m -separated given \mathbf{Z} in an ancestral graph \mathcal{M} if and only if $\langle X, Y | \mathbf{Z} \rangle$ is in the corresponding independence model \mathcal{J} . Still, like in SMCs, a missing edge does not necessarily correspond to a conditional independence. The following definition describes a subset of ancestral graphs in which every missing edge (non-adjacency) corresponds to a conditional independence:

Definition 2 (*Maximal Ancestral Graph, MAG*) (Richardson and Spirtes, 2002) *A mixed graph is called ancestral if it contains no directed and almost directed cycles. An ancestral graph \mathcal{G} is called maximal if for every pair of non-adjacent nodes (X, Y) , there is a (possibly empty) set \mathbf{Z} , $X, Y \notin \mathbf{Z}$ such that $\langle X, Y | \mathbf{Z} \rangle \in \mathcal{J}$.*

Figure 1 illustrates an ancestral graph that is not maximal, and the corresponding maximal ancestral graph. MAGs are closed under marginalization (Richardson and Spirtes, 2002). Thus, if \mathcal{G} is a MAG faithful to \mathcal{P} , then there is a unique MAG \mathcal{G}' faithful to any marginal distribution of \mathcal{P} .

We use $[\mathbf{L}]$ to denote the act of marginalizing out variables \mathbf{L} , thus, if \mathcal{G} is a MAG over variables $\mathbf{O} \cup \mathbf{L}$ faithful to a joint probability distribution \mathcal{P} , $\mathcal{G}_{[\mathbf{L}]}$ is the MAG over \mathbf{O} faithful to the marginal joint probability distribution $\mathcal{P}_{[\mathbf{L}]}$. Obviously, the DAG of a causal Bayesian

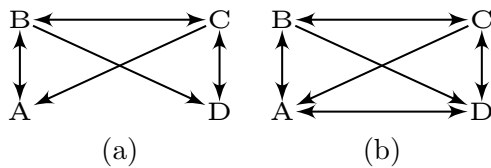


Figure 1: **Maximality and primitive inducing paths.** (a) Both (i) a semi Markov causal model over variables $\{A, B, C, D\}$. Variables A and D are m -connected given any subset of observed variables, but they do not share a direct relationship in the context of observed variables and (ii) a non-maximal ancestral graph over variables $\{A, B, C, D\}$. (b) The corresponding MAG. A and D are adjacent, since they cannot be m -separated given any subset of $\{B, C\}$. Path $\langle A, B, C, D \rangle$ is a primitive inducing path. This example was presented in Zhang (2008b).

network is also a MAG. For a MAG \mathcal{G} over \mathbf{O} and a set of variables $\mathbf{L} \subset \mathbf{O}$, the marginal MAG $\mathcal{G}_{[\mathbf{L}]}$ is defined as follows:

Definition 3 (Richardson and Spirtes, 2002) *MAG $\mathcal{G}_{[\mathbf{L}]}$ has node set $\mathbf{O} \setminus \mathbf{L}$ and edges specified as follows: If X, Y are s.t. $\forall \mathbf{Z} \subset \mathbf{O} \setminus \mathbf{L} \cup \{X, Y\}$, X and Y are m -connected given \mathbf{Z} in \mathcal{G} , then*

$$\text{if } \left\{ \begin{array}{l} X \notin \mathbf{An}_{\mathcal{G}}(Y); Y \notin \mathbf{An}_{\mathcal{G}}(X) \\ X \in \mathbf{An}_{\mathcal{G}}(Y); Y \notin \mathbf{An}_{\mathcal{G}}(X) \\ X \notin \mathbf{An}_{\mathcal{G}}(Y); Y \in \mathbf{An}_{\mathcal{G}}(X) \end{array} \right\} \text{ then } \left\{ \begin{array}{l} X \leftrightarrow Y \\ X \rightarrow Y \\ X \leftarrow Y \end{array} \right\} \text{ in } \mathcal{G}_{[\mathbf{L}]}$$

As mentioned above, every conditional independence in an independence model \mathcal{I} corresponds to a missing edge in the corresponding faithful MAG \mathcal{G} . Conversely, if X and Y are dependent given every subset of observed variables, then X and Y are adjacent in \mathcal{G} . Thus, given an oracle of conditional independence it is possible to learn the skeleton of a MAG \mathcal{G} over variables \mathbf{O} from a data set. Still, some of the orientations of \mathcal{G} are not distinguishable by mere observations. The set of MAGs \mathcal{G} faithful to distributions \mathcal{P} that entail a set of conditional independencies form a **Markov equivalence class**. The following result was proved in Spirtes and Richardson (1996):

Proposition 4 *Two MAGs over the same variable set are Markov equivalent if and only if:*

1. *They share the same edges.*
2. *They share the same unshielded colliders.*
3. *If a path p is discriminating for a node V in both graphs, V is a collider on the path on one graph if and only if it is a collider on the path on the other.*

We use $[\mathcal{G}]$ to denote the class of MAGs that are Markov equivalent to \mathcal{G} . A **partial ancestral graph (PAG)** is a representative graph of this class, and has the skeleton shared by all the graphs in $[\mathcal{G}]$, and all the orientations invariant in all the graphs in $[\mathcal{G}]$. Endpoints

that can be either arrows or tails in different MAGs in \mathcal{G} are denoted with a circle “ \circ ” in the representative PAG. We use the symbol \star as a wildcard to denote any of the three marks. We use the notations $\mathcal{M} \in \mathcal{P}$ to denote that MAG \mathcal{M} belongs to the Markov equivalence class represented by PAG \mathcal{P} , and we use the notation $\mathcal{M} \in \mathcal{J}$ to denote that MAG \mathcal{M} is faithful to the conditional independence model \mathcal{J} . **FCI** Algorithm (Spirtes et al., 2001; Zhang, 2008a) is a sound and complete algorithm for learning the complete (maximally informative) PAG of the MAGs faithful to a distribution \mathcal{P} over variables \mathbf{O} in which a set of conditional independencies \mathcal{J} hold. An important advantage of FCI is that it employs CMC, faithfulness and some graph theory to reduce the number of tests required to identify the correct PAG.

3.3 Correspondence between SMCMs and MAGs

Semi Markov Causal Models and Maximal Ancestral Graphs both represent causally insufficient causal structures, but they have some significant differences. While they both entail the conditional independence and causal ancestry structure of the observed variables, SMCMs describe the causal relations among observed variables, while MAGs encode independence structure with partial causal ordering. Edge semantics in SMCMs are closer to the semantics of causal Bayesian networks, whereas edge semantics in MAGs are more complicated. On the other hand, unlike in DAGs and MAGs, a missing edge in a SMCM does not necessarily correspond to a conditional independence (SMCMs do not obey a pairwise Markov property).

Figure 2 summarizes the main differences of SMCMs and MAGs. It shows two different DAGs, and the corresponding marginal SMCMs and MAGs over four observed variables. SMCMs have a many-to-one relationship to MAGs: For a MAG \mathcal{M} , there can exist more than one SMCMs that entail the same probabilistic and causal ancestry relations. On the other hand, for any given SMCM there exists only one MAG entailing the same probabilistic and causal ancestry relations. This is clear in Figure 2, where a unique MAG, $\mathcal{M}_1 = \mathcal{M}_2$ entails the same information as two different SMCMs, \mathcal{S}_1 and \mathcal{S}_2 in the same figure.

Directed edges in a SMCM denote a causal relation that is *direct* in the context of observed variables. In contrast, a directed edge in a MAG merely denotes causal ancestry; the causal relation is not necessarily direct. An edge $X \rightarrow Y$ can be present in a MAG even though X does not directly causes Y ; this happens when X is a causal ancestor of Y and the two cannot be rendered independent given any subset of observed variables. Depending on the structure of latent variables, this edge can be either missing or bi-directed in the respective SMCM.

In Figure 2 we can see examples of both cases. For example, A is a causal ancestor of D in DAG \mathcal{G}_1 , but not a direct cause (in the context of observed variables). Therefore, the two are not adjacent in the corresponding SMCM \mathcal{S}_1 over $\{A, B, C, D\}$. However, the two cannot be rendered independent given any subset of $\{B, C\}$, and therefore $A \rightarrow D$ is in the respective MAG \mathcal{M}_1 .

On the same DAG, B is another causal ancestor (but not a direct cause) of D . The two variables share the common cause L . Thus, in the corresponding SMCM \mathcal{S}_1 over $\{A, B, C, D\}$ we can see the edge $B \leftrightarrow D$. However, a bi-directed edge between B and D is

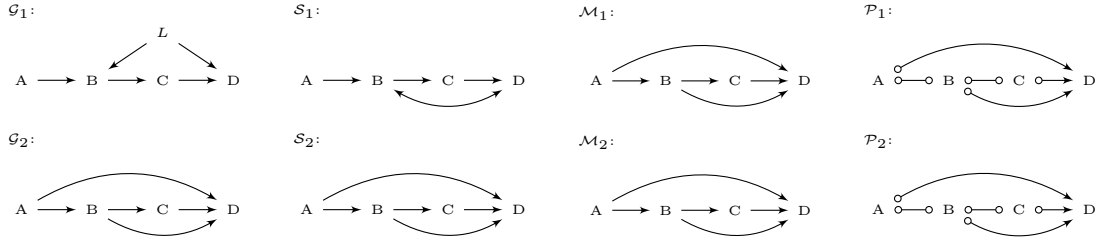


Figure 2: **An example two different DAGs and the corresponding mixed causal graphs over observed variables.** On the right we can see DAGs \mathcal{G}_1 over variables $\{A, B, C, D, L\}$ (top) and \mathcal{G}_2 over variables $\{A, B, C, D\}$ (bottom). From left to right, on the same row as the underlying causal DAG, we can see the respective SMCMs \mathcal{S}_1 and \mathcal{S}_2 over $\{A, B, C, D\}$; the respective MAGs $\mathcal{M}_1 = \mathcal{G}_1[L]$ and $\mathcal{M}_2 = \mathcal{G}_2$ over variables $\{A, B, C, D\}$; finally, the respective PAGs \mathcal{P}_1 and \mathcal{P}_2 . Notice that, \mathcal{M}_1 and \mathcal{M}_2 are identical, despite representing different underlying causal structures.

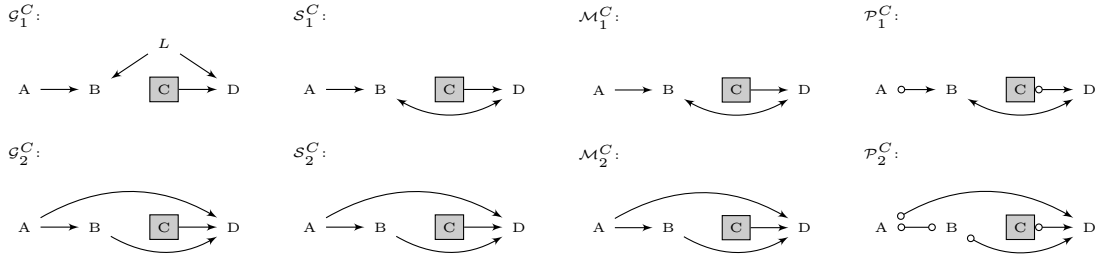


Figure 3: **Effect of manipulating variable C on the causal graphs of Figure 2.** From right to left we can see the manipulated DAGs \mathcal{G}_1^C (top) and \mathcal{G}_2^C (bottom), the manipulated SMCMs \mathcal{S}_1^C (top) and \mathcal{S}_2^C (bottom) over variables $\{A, B, C, D\}$, the manipulated MAGs $\mathcal{M}_1^C = \mathcal{G}_1^C[L]$ (top) and $\mathcal{M}_2^C = \mathcal{G}_2^C$ (bottom) over the same set of variables, and the corresponding PAGs \mathcal{P}_1^C (top) and \mathcal{P}_2^C (bottom). Notice that edge $A \rightarrow D$ is removed in \mathcal{M}_1^C , even though it is not adjacent to the manipulated variable. Moreover, on the same graph, edge $B \rightarrow D$ is now $B \leftrightarrow D$.

not allowed in MAG \mathcal{M}_1 , since it would create an almost directed cycle. Thus, $B \rightarrow D$ is in \mathcal{M}_1 .

We must also note that unlike SMCMs, MAGs only allow one edge per variable pair. Thus, if X directly causes Y and the two are also confounded, both edges will be in a relevant SMCM ($X \leftrightarrow Y$), while the two will share a directed edge from X to Y in the corresponding MAG.

Overall, a SMCM has a subset of adjacencies (but not necessarily edges) of its MAG counterpart. These extra adjacencies correspond to pairs of variables that cannot be m -separated given any subset of observed variables, but neither directly causes the other, and

the two are not confounded. These adjacencies can be checked in a SMCM using a special type of path, called **inducing path** (Richardson and Spirtes, 2002).

Definition 5 (*inducing path*) A path $p = \langle V_1, V_2, \dots, V_n \rangle$ on a mixed causal graph \mathcal{G} over a set of variables $\mathbf{V} = \mathbf{O} \dot{\cup} \mathbf{L}$ is called **inducing** with respect to \mathbf{L} if every non-collider on the path is in \mathbf{L} and every collider is an ancestor of either V_1 or V_n . A path that is inducing with respect to the empty set is called a **primitive inducing path**.

Obviously, an edge joining X and Y is a primitive inducing path. Intuitively, an inducing path with respect to \mathbf{L} is m -connecting given any subset of variables that does not include variables in \mathbf{L} . Path $A \rightarrow B \leftarrow L \rightarrow D$ is an inducing path with respect to L in \mathcal{G}_1 of Figure 2, and $A \rightarrow B \leftrightarrow D$ is an inducing path with respect to the empty set in \mathcal{S}_1 of the same figure. Inducing paths are extensively discussed in Richardson and Spirtes (2002), where the following theorem is proved:

Theorem 6 If \mathcal{G} is an ancestral graph over variables $\mathbf{V} = \mathbf{O} \dot{\cup} \mathbf{L}$, and $X, Y \in \mathbf{O}$ then the following statements are equivalent:

1. X and Y are adjacent in $\mathcal{G}[\mathbf{L}]$.
2. There is an inducing path with respect to \mathbf{L} in \mathcal{G} .
3. $\forall \mathbf{Z}, \mathbf{Z} \subseteq \mathbf{V} \setminus \mathbf{L} \cup \{X, Y\}$, X and Y are m -connected given \mathbf{Z} in \mathcal{G} .

Proof See proof of Theorem 4.2 in Richardson and Spirtes (2002). ■

This theorem links inducing paths in an ancestral graph to m -separations in the same graph and to adjacencies in any marginal ancestral graph. The equivalence of (ii) and (iii) can also be proved for SMCMs, using the proof presented in Richardson and Spirtes (2002) for Theorem 6:

Theorem 7 If \mathcal{G} is a SMCM over variables $\mathbf{V} = \mathbf{O} \dot{\cup} \mathbf{L}$, and $X, Y \in \mathbf{O}$ then the following statements are equivalent:

1. There is an inducing path with respect to \mathbf{L} in \mathcal{G} .
2. $\forall \mathbf{Z}, \mathbf{Z} \subseteq \mathbf{V} \setminus \mathbf{L} \cup \{X, Y\}$, X and Y are m -connected given \mathbf{Z} in \mathcal{G} .

Primitive inducing paths are connected to the notion of maximality in ancestral graphs: Every ancestral graph can be transformed into a maximal ancestral graph with the addition of a finite number of bi-directed edges. Such edges are added between variables X, Y that are m -connected through a **primitive inducing path** (Richardson and Spirtes, 2002). Path $A \leftrightarrow B \leftrightarrow C \leftrightarrow D$ in Figure 1 is an example of a primitive inducing path.

Inducing paths are crucial in this work because adjacencies and non-adjacencies in marginal ancestral graphs can be translated into existence or absence of inducing paths in causal graphs that include some additional variables. For example, path $A \rightarrow B \leftarrow L \rightarrow D$ is an inducing path w.r.t. L in \mathcal{G}_1 in Figure 2, and therefore A and D are adjacent in

Algorithm 1: SMCMtoMAG

```
input : SMCM  $\mathcal{S}$ 
output: MAG  $\mathcal{M}$ 

1  $\mathcal{M} \leftarrow \mathcal{S}$ ;
2 foreach ordered pair of variables  $X, Y$  not adjacent in  $\mathcal{S}$  do
3   | if  $\exists$  primitive inducing path from  $X$  to  $Y$  in  $\mathcal{S}$  then
4   |   | if  $X \in \mathbf{An}_{\mathcal{S}}(Y)$  then
5   |   |   | add  $X \rightarrow Y$  to  $\mathcal{M}$ ;
6   |   | else if  $Y \in \mathbf{An}_{\mathcal{S}}(X)$  then
7   |   |   | add  $Y \rightarrow X$  to  $\mathcal{M}$ ;
8   |   | else
9   |   |   | add  $Y \leftrightarrow X$  to  $\mathcal{M}$ ;
10  |   | end
11  | end
12 end
13 foreach  $X \leftrightarrow Y$  in  $\mathcal{M}$  do
14 | remove  $X \leftrightarrow Y$ ;
15 end
```

\mathcal{M}_1 . Thus, inducing paths are useful for combining causal mixed graphs over overlapping variables.

Inducing paths are also necessary to decide whether two variables in an SMCM will be adjacent in a MAG over the same variables without having to check all possible m -separations. Algorithm 1 describes how to turn a SMCM into a MAG over the same variables. To prove the algorithm's soundness, we first need to prove the following:

Proposition 8 *Let \mathbf{O} be a set of variables and \mathcal{J} the independence model over \mathbf{V} . Let \mathcal{S} be a SMCM over variables \mathbf{V} that is faithful to \mathcal{J} and \mathcal{M} be the MAG over the same variables that is faithful to \mathcal{J} . Let $X, Y \in \mathbf{O}$. Then there is an inducing path between X and Y with respect to \mathbf{L} , $\mathbf{L} \subseteq \mathbf{V}$ in \mathcal{S} if and only if there is an inducing path between X and Y with respect to \mathbf{L} in \mathcal{M} .*

Proof See Appendix 6.. ■

Algorithm 1 takes as input a SMCM and adds the necessary edges to transform it into a MAG by looking for primitive inducing paths. The soundness of the algorithm is a direct consequence of Proposition 8. The inverse procedure, converting a MAG into the underlying SMCM, is not possible, since we cannot know in general which of the edges correspond to direct causation or confounding and which are there because of a (non-trivial) primitive inducing path. Note though that, there exist sound and complete algorithms that identify all edges for which such a determination is possible (Borboudakis et al., 2012). In addition, we later show that co-examining manipulated distributions can indicate that some edges stand for indirect causality (or indirect confounding).

3.4 Manipulations under causal insufficiency

An important motivation for using causal models is to predict causal effects. In this work, we focus on hard manipulations, where the value of the manipulated variables is set exclusively by the manipulation procedure. We also adopt the assumption of locality, denoting that the intervention of each manipulated variable should not directly affect any variable other than its direct target, and more importantly, local mechanisms for other variables should remain the same as before the intervention (Zhang, 2006). Thus, the intervention is merely a local surgery with respect to causal mechanisms. These assumptions may seem a bit restricting, but this type of experiment is fairly common in several modern fields where the technical capability for precise interventions is available, such as, for example, molecular biology. Finally, we assume that the manipulated model is faithful to the corresponding manipulated distributions.

In the context of causal Bayesian networks, hard interventions are modeled using what is referred to as “graph surgery”, in which all edges incoming to the manipulated variables are removed from the graph. The resulting graph is referred to as the **manipulated graph**. Parameters of the distribution that refer to the probability of manipulated variables given their parents are replaced by the parameters set by the manipulation procedure, while all other parameters remain intact. Naturally, DAGs are closed under manipulation. We use the term **intervention target** to denote a set of manipulated variables. For a DAG \mathcal{D} and an intervention target \mathbf{I} , we use $\mathcal{D}^{\mathbf{I}}$ to denote the manipulated DAG. The same notation (the intervention targets as a superscript) is used to denote a manipulated independence model.

Graph surgery can be easily extended to SMCMs: One must simply remove edges into the manipulated variables. Again, we use the notation $\mathcal{S}^{\mathbf{I}}$ to denote the graph resulting from a SMCM \mathcal{S} after the manipulation of variables in \mathbf{I} . On the contrary, predicting the effect of manipulations in MAGs is not trivial. Due to the complicated semantics of the edges, the manipulated graph is usually not unique.

This becomes more obvious by looking at Figures 2 and 3. Figure 2 shows two different causal DAGs and the corresponding SMCMs and MAGs, and Figure 3 shows the effect of a manipulation on the same graphs. In Figure 2 the marginal DAGs \mathcal{D}_1 and \mathcal{D}_2 are represented by the same MAG $\mathcal{M}_1 = \mathcal{M}_2$. However, after manipulating variable C , the resulting manipulated MAGs \mathcal{M}_1^C and \mathcal{M}_2^C do not belong to the same equivalence class (they do not even share the same skeleton). We must point out, that the indistinguishability of \mathcal{M}_1 and \mathcal{M}_2 refers to m -separation only; the absence of a direct causal edge between A and D could be detected using other types of tests, like the Verma constraint (Verma and Pearl, 2003).

While we cannot predict the effect of manipulations on a MAG \mathcal{M} , given a data set measuring variables \mathbf{O} when variables in $\mathbf{I} \subset \mathbf{O}$ are manipulated, we can obtain (assuming an oracle of conditional independence) the PAG representative of the actual manipulated MAG $\mathcal{M}^{\mathbf{I}}$. We use $\mathcal{P}^{\mathbf{I}}$ to denote this PAG. Moreover, by observing PAGs $\{\mathcal{P}^{\mathbf{I}_i}\}_i$ that stem from different manipulations of the same underlying distribution, we can infer some more refined information for the underlying causal model.

Let’s suppose, for example, that \mathcal{G}_1 in Figure 2 is the true underlying causal graph for variables $\{A, B, C, D, L\}$ and that we have the learnt PAGs \mathcal{P}_1^A and \mathcal{P}_1^C from relevant data

sets. Graph \mathcal{P}_1^A is not shown, but is identical to \mathcal{P}_1 in Figure 2 since A has no incoming edges in the underlying DAG (and SMCM). \mathcal{P}_1^C is illustrated in Figure 3. Edge $A \circ\text{---}D$ is present in \mathcal{P}_1^A , but is missing in \mathcal{P}_1^C even though neither A nor D are manipulated in \mathcal{P}_1^C . By reasoning on the basis of both graphs, we can infer that edge $A \rightarrow D$ in \mathcal{P}_1^A cannot denote a *direct* causal relation among the two variables, but must be the result of a primitive, non-trivial inducing path.

4. Learning causal structure from multiple data sets measuring overlapping variables under different manipulations

In the previous section we described the effect of manipulation on MAGs and saw an example of how co-examining PAGs faithful to different manipulations of the same underlying distribution can help classify an edge between two variables as not direct.

In this section, we expand this idea and present a general, constraint-based algorithm for learning causal structure from overlapping manipulations. The algorithm takes as input a set of data sets measuring overlapping variable sets $\{\mathbf{O}_i\}_{i=1}^N$; in each data set, some of the observed variables can be manipulated. The set of manipulated variables in data set i is also provided and is denoted with \mathbf{I}_i .

We assume that there exists an underlying causal mechanism over the union of observed variables $\mathbf{O} = \bigcup_i \mathbf{O}_i$ that can be described with a probability distribution \mathcal{P} over \mathbf{O} and a semi Markov causal model \mathcal{S} such that \mathcal{P} and \mathcal{S} are faithful to each other. We denote with \mathcal{J} the independence model of \mathcal{P} . Every manipulation is then performed on \mathcal{S} and only on variables observed in the corresponding data set. In addition, we assume Faithfulness holds for the manipulated graphs as well. The data are then sampled from the manipulated distribution. In each data set i , the set $\mathbf{L}_i = \mathbf{O} \setminus \mathbf{O}_i$ is latent. We denote the independence model of each data set i as $\mathcal{J}_i \equiv \mathcal{J}^{\mathbf{I}_i}[\mathbf{L}_i]$. We now define the following problem:

Definition 9 (Identify a consistent SMCM) *Given sets $\{\mathbf{O}_i\}_{i=1}^N$, $\{\mathbf{I}_i\}_{i=1}^N$, and $\{\mathcal{J}_i\}_{i=1}^N$ identify a SMCM \mathcal{S} , such that:*

$$\mathcal{M}_i \in \mathcal{J}_i, \forall i \text{ where } \mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[\mathbf{L}_i]$$

*that is, \mathcal{M}_i is the MAG corresponding to the manipulated marginal of \mathcal{S} , for each data set i . We call such a graph \mathcal{S} a **possibly underlying SMCM** for $\{\mathcal{J}_i\}_{i=1}^N$.*

We present an algorithm that converts the problem above into a satisfiability instance s.t. a SMCM is consistent iff it corresponds to a truth-setting assignment of the SAT instance. Notice that, an independence model \mathcal{J} corresponds to a PAG \mathcal{P} over the same variables when they represent the same Markov equivalence class of MAGs. Thus, in what follows we use the corresponding set of manipulated marginal PAGs $\{\mathcal{P}_i\}_{i=1}^N$ instead of the independence models $\{\mathcal{J}_i\}_{i=1}^N$. Notice that, PAGs $\{\mathcal{P}_i\}_{i=1}^N$ can be learnt with a sound and complete algorithm such as FCI.

In the following section, we discuss converting the problem presented above into a constraint satisfaction problem.

Formulae relating properties of observed PAGs to the underlying SMCM \mathcal{S} :

$$\mathbf{adjacent}(X, Y, \mathcal{P}_i) \leftrightarrow \exists p_{XY} : \mathit{inducing}(p_{XY}, i)$$

$$\mathbf{unshielded_dnc}(X, Y, Z, \mathcal{P}_i) \rightarrow \\ \mathit{unshielded}(\langle X, Y, Z \rangle, \mathcal{P}_i) \wedge (\mathit{ancestor}(Y, X, i) \vee \mathit{ancestor}(Y, Z, i))$$

$$\mathbf{discriminating_dnc}(\langle W, \dots, X, Y, Z \rangle, Y, \mathcal{P}_i) \rightarrow \\ (\mathit{discriminating}(\langle W, \dots, X, Y, Z \rangle, Y, \mathcal{P}_i) \wedge \mathit{ancestor}(Y, X, i) \vee \mathit{ancestor}(Y, Z, i))$$

$$\mathbf{unshielded_collider}(X, Y, Z, \mathcal{P}_i) \rightarrow \\ \mathit{unshielded}(\langle X, Y, Z \rangle, \mathcal{P}_i) \wedge (\neg \mathit{ancestor}(Y, X, i) \wedge \neg \mathit{ancestor}(Y, Z, i))$$

$$\mathbf{discriminating_collider}(\langle W, \dots, X, Y, Z \rangle, Y, \mathcal{P}_i) \rightarrow \\ (\mathit{discriminating}(\langle W, \dots, X, Y, Z \rangle, Y, \mathcal{P}_i) \wedge (\neg \mathit{ancestor}(Y, X, i) \wedge \neg \mathit{ancestor}(Y, Z, i)))$$

$$\mathbf{unshielded}(\langle X, Y, Z \rangle, \mathcal{P}_i) \leftrightarrow \\ \mathit{adjacent}(X, Y, \mathcal{P}_i) \wedge \mathit{adjacent}(Y, Z, \mathcal{P}_i) \wedge \neg \mathit{adjacent}(X, Z, \mathcal{P}_i)$$

$$\mathbf{discriminating}(\langle V_0, V_1, \dots, V_{n-1}, V_n, V_{n+1} \rangle, V_n, \mathcal{P}_i) \leftrightarrow \\ \forall j [V_j \notin \mathbf{I}_i \wedge \mathit{adjacent}(V_{j-1}, Y, \mathcal{P}_i) \wedge \mathit{ancestor}(V_j, V_{n+1}, i) \wedge \\ \mathit{adjacent}(V_{j-1}, V_j, \mathcal{P}_i) \wedge \neg \mathit{ancestor}(V_j, V_{j-1}, i) \wedge \neg \mathit{ancestor}(V_{j-1}, V_j, i)]$$

Formulae reducing path properties of \mathcal{S} to the core variables:

$$\mathbf{inducing}(p_{XY}, i) \leftrightarrow \\ \forall j V_j \notin \mathbf{I}_i \wedge (X \in \mathbf{I}_i \rightarrow \mathit{tail}(V_1, X)) \wedge (Y \in \mathbf{I}_i \rightarrow \mathit{tail}(V_n, Y)) \wedge \\ (|p_{XY}| = 2 \rightarrow \mathit{edge}(X, Y)) \wedge (|p_{XY}| > 2 \rightarrow \forall j \mathit{unblocked}(\langle V_{j-1}, V_j, V_{j+1} \rangle, X, Y, i))$$

$$\mathbf{unblocked}(\langle Z, V, W \rangle, X, Y, i) \leftrightarrow \\ \mathit{edge}(Z, V) \wedge \mathit{edge}(V, W) \wedge \\ [V \in \mathbf{I}_i \rightarrow \neg \mathit{head2head}(Z, V, W, i) \vee \mathit{ancestor}(V, X, i) \vee \mathit{ancestor}(V, Y, i)] \wedge \\ [V \notin \mathbf{I}_i \rightarrow \mathit{head2head}(Z, V, W, i) \wedge (\mathit{ancestor}(V, X, i) \vee \mathit{ancestor}(V, Y, i))]$$

$$\mathbf{head2head}(X, Y, Z, i) \leftrightarrow Y \notin \mathbf{I}_i \wedge \mathit{arrow}(X, Y) \wedge \mathit{arrow}(Z, Y)$$

$$\mathbf{ancestor}(X, Y, i) \leftrightarrow \exists p_{XY} : \mathit{ancestral}(p_{XY}, i)$$

$$\mathbf{ancestral}(p_{XY}, i) \leftrightarrow \\ \forall j [V_j \notin \mathbf{I}_i \wedge (\mathit{edge}(V_{j-1}, V_j) \wedge \mathit{tail}(V_j, V_{j-1}) \wedge \mathit{arrow}(V_{j-1}, V_j))]$$

Figure 4: Graph properties expressed as boolean formulae using the variables *edge*, *arrow* and *tail*. In all equations, we use p_{XY} to denote a path of length $n+2$ between X and Y in \mathcal{S} : $p_{XY} = \langle V_0 = X, V_1, \dots, V_j, \dots, V_n, V_{n+1} = Y \rangle$. Index i is used to denote experiment i , where variables \mathbf{I}_i are latent and variables \mathbf{O}_i are manipulated. Conjunction and disjunction are assumed to have precedence over implication with regard to bracketing.

4.1 Conversion to SAT

Definition 9 implies that each \mathcal{M}_i has the same edges (adjacencies), the same unshielded colliders and the same discriminating colliders as \mathcal{P}_i , for all i . We impose these constraints on \mathcal{S} by converting them to a SAT instance. We express the constraints in terms of the following **core** variables, denoting edges and orientation orientations in any consistent SMCM \mathcal{S} .

- $\text{edge}(X, Y)$: true if X and Y are adjacent in \mathcal{S} , false otherwise.
- $\text{tail}(X, Y)$: true if there exists an edge between X and Y in \mathcal{S} that is out of Y , false otherwise.
- $\text{arrow}(X, Y)$: true if there exists an edge between X and Y in \mathcal{S} that is into Y , false otherwise.

Variables tail and arrow are not mutually exclusive, enabling us to represent $X \leftrightarrow Y$ edges when $\text{tail}(Y, X) \wedge \text{arrow}(Y, X)$. Each independence model \mathcal{J}_i is entailed by the (non) adjacencies and (non) colliders in each observed PAG \mathcal{P}_i . These structural characteristics correspond to paths in any possibly underlying SMCM as follows:

1. $\forall X, Y \in \mathbf{O}_i$, X and Y are adjacent in \mathcal{P}_i if and only if there exists an inducing path between X and Y with respect to \mathbf{L}_i in $\mathcal{S}^{\mathbf{I}_i}$ (by Theorems 6 and 7 and Proposition 8).
2. If $\langle X, Y, Z \rangle$ is an unshielded definite non collider in \mathcal{P}_i , then $\langle X, Y, Z \rangle$ is an unshielded triple in \mathcal{P}_i and Y is an ancestor of either X or Z in $\mathcal{S}^{\mathbf{I}_i}$ (by the semantics of edges in MAGs).
3. If $\langle X, Y, Z \rangle$ is an unshielded collider in \mathcal{P}_i , then $\langle X, Y, Z \rangle$ is an unshielded triple in \mathcal{P}_i and Y is not an ancestor of X nor Z in $\mathcal{S}^{\mathbf{I}_i}$ (by the semantics of edges in MAGs).
4. If $\langle W, \dots, X, Y, Z \rangle$ is a discriminating collider in \mathcal{P}_i , then $\langle W, \dots, X, Y, Z \rangle$ is a discriminating path for Y in \mathcal{P}_i and Y is an ancestor of either X or Z in $\mathcal{S}^{\mathbf{I}_i}$ (by the semantics of edges in MAGs).
5. If $\langle W, \dots, X, Y, Z \rangle$ is a discriminating definite non collider in \mathcal{P}_i , then $\langle W, \dots, X, Y, Z \rangle$ is a discriminating path for Y in \mathcal{P}_i and Y is not an ancestor of X nor Z in $\mathcal{S}^{\mathbf{I}_i}$ (by the semantics of edges in MAGs).

These constraints are expressed using the core variables (edges, tails and arrows), as described in Figure 4. For example, if X and Y are adjacent in \mathcal{P}_i , in a consistent SMCM \mathcal{S} there must exist an inducing path p between X and Y in $\mathcal{S}^{\mathbf{I}_i}$ with respect to variables \mathbf{L}_i . Any truth-assignment to the core variables that does not entail the presence of such an inducing path should not satisfy the SAT instance. The following constraints are added to ensure that the graphs satisfying constraints 1-5 above are SMCMs:

6. $\forall X, Y \in \mathbf{O}$, either X is not an ancestor of Y or Y is not an ancestor of X in \mathcal{S} (no directed cycles).
7. $\forall X, Y \in \mathbf{O}$, at most one of $\text{tail}(X, Y)$ and $\text{tail}(Y, X)$ can be true (no selection bias).

Algorithm 2: COmbINE

input : data sets $\{\mathbf{D}_i\}_{i=1}^N$, sets of intervention targets $\{\mathbf{I}_i\}_{i=1}^N$, FCI parameters $params$, maximum path length mpl , conflict resolution strategy str
output: Summary graph \mathcal{H}

- 1 **foreach** i **do** $\mathcal{P}_i \leftarrow \text{FCI}(\mathbf{D}_i, params)$ $\mathcal{H} \leftarrow \text{initializeSMCM}(\{\mathcal{P}_i\}_{i=1}^N)$;
- 2 $(\Phi, \mathcal{F}) \leftarrow \text{addConstraints}(\mathcal{H}, \{\mathcal{P}_i\}_{i=1}^N, \{\mathbf{I}_i\}_{i=1}^N, mpl)$;
- 3 $\mathcal{F}' \leftarrow$ select a subset of non-conflicting literals \mathcal{F}' according to strategy str ;
- 4 $\mathcal{H} \leftarrow \text{backBone}(\Phi \wedge \mathcal{F}')$

8. $\forall X, Y \in \mathbf{O}$, at least one of $tail(X, Y)$ and $arrow(Y, X)$ must be true.

Naturally, Constraints 7 and 8 are meaningful only if X and Y are adjacent (if $\text{edge}(X, Y)$ is true), and redundant otherwise.

4.2 Algorithm COmbINE

We now present algorithm **COmbINE** (Causal discovery from Overlapping INtErventions) that learns causal features from multiple, heterogenous data sets. The algorithm takes as input a set of data sets $\{\mathbf{D}_i\}_{i=1}^N$ over a set of overlapping variable sets $\{\mathbf{O}_i\}_{i=1}^N$. In each data set, a (possibly empty) subset of the observed variables $\mathbf{I}_i \subset \mathbf{O}_i$ may be manipulated. FCI is run on each data set and the corresponding PAGs $\{\mathcal{P}_i\}_{i=1}^N$ are produced. The algorithm then creates an candidate underlying SMCM \mathcal{H} . Subsequently, for each PAG \mathcal{P}_i , the features of \mathcal{P}_i are translated into constraints, expressed in terms of edges and endpoints in \mathcal{H} , using the formulae in Figure 4. In the sample limit (and under the assumptions discussed above), the SAT formula $\Phi \wedge \mathcal{F}'$ produced by this procedure is satisfied by all and only the possible underlying SMCMs for $\{\mathcal{P}_i\}_{i=1}^N$. In the presence of statistical errors, however, $\Phi \wedge \mathcal{F}'$ may be unsatisfiable. To handle conflicts, the algorithm takes as input a strategy for selecting a non-conflicting subset of constraints and ignores the rest. Finally, COmbINE queries the SAT formula for variables that have the same truth-value in all satisfying assignments, translates them into graph features, and returns a graph that summarizes the invariant edges and orientations of all possible underlying SMCMs. In the rest of this paper we call the graphical output of COmbINE a **summary graph**.

The pseudocode for COmbINE is presented in Algorithm 2. Apart from the set of data sets described above, COmbINE takes as input the chosen parameters for FCI (threshold α , maximum conditioning set $maxK$), the maximum length of possible inducing paths to consider and a strategy for selecting a subset of non-conflicting constraints.

Initially, the algorithm runs FCI on each data set \mathbf{D}_i and produces the corresponding PAG \mathcal{P}_i . Then the candidate SMCM \mathcal{H} is initialized: \mathcal{H} is the graph upon which all path constraints will be imposed. Therefore, \mathcal{H} must have at least a superset of edges and at most a subset of orientations of any consistent SMCM \mathcal{S} : If p is an inducing (ancestral) path in \mathcal{S} , it must be a possibly inducing (ancestral) path in \mathcal{H} . An obvious–yet not very smart–choice for \mathcal{H} would be the complete unoriented graph. However, looking for possible inducing and ancestral paths on the complete unoriented graph over the union of variables

Algorithm 3: initializeSMCM

input : PAGs $\{\mathcal{P}_i\}_{i=1}^N$
output: initial graph \mathcal{H}

- 1 $\mathcal{H} \leftarrow$ empty graph over $\cup \mathbf{O}_i$;
- 2 **foreach** i **do** $\mathcal{H} \leftarrow$ add all edges in \mathcal{P}_i unoriented Orient only arrowheads that are present in every \mathcal{P}_i ;
- /* Add edges between variables never measured unmanipulated together */
- 3 **foreach** pair X, Y of non-adjacent nodes **do**
- 4 **if** $\nexists i$ s.t. $X, Y \in \mathbf{O}_i \setminus \mathbf{I}_i$ **then**
- 5 add $X \circ \circ Y$ to \mathcal{H} ;
- 6 **if** $\exists i$ s.t. $X, Y \in \mathbf{O}_i, X \in \mathbf{I}_i, Y \notin \mathbf{I}_i$ **then** add arrow into X **if** $\exists i$ s.t. $X, Y \in \mathbf{O}_i, Y \in \mathbf{I}_i, X \notin \mathbf{I}_i$ **then** add arrow into Y
- 7 **end**
- 8 **end**

could make the problem intractable even for small input sizes. To reduce the number of possible inducing and ancestral paths, we use Algorithm 3 to construct \mathcal{H} .

Algorithm 3 constructs a graph \mathcal{H} that has all edges observed in any PAG \mathcal{P}_i as well as some additional edges that would not have been observed even if they existed: Edges connecting variables that have never been observed together, and edges connecting variables that have been observed together, but at least one of them was manipulated in each joint appearance in a data set. For example, variables X_9 and X_{15} in Figure 5 are measured together in two data sets: \mathbf{D}_2 and \mathbf{D}_3 . If $X_9 \rightarrow X_{15}$ in the underlying SMCM, this edge would be present in \mathcal{P}_3 . Similarly, if $X_{15} \rightarrow X_9$ in the underlying SMCM, the variables would be adjacent in \mathcal{P}_2 . We can therefore rule out the possibility of a directed edge between the two variables in \mathcal{S} . However, it is possible that X_{15} and X_9 are confounded in \mathcal{S} , and the edge disappears by the manipulation procedure in both \mathcal{P}_2 and \mathcal{P}_3 . Thus, Algorithm 3 will add these possible edges in \mathcal{H} . In addition, in Line 2, Algorithm 3 adds all the orientations found so far in all \mathcal{P}_i 's that are invariant¹. The resulting graph has, in the sample limit, a superset of edges and a subset of orientations compared to the actual underlying SMCM. Lemma 10 formalizes and proves this property.

Having initialized the search graph, Algorithm 2 proceeds to generate the constraints. This procedure is described in detail in Algorithm 4, that is the core of COMBINE. These are: (i) the bi-conditionals regarding the presence/absence of edges (Line 4), (ii) conditionals regarding unshielded and discriminating colliders (Lines 12, 13, 17 and 18), (iii) constraints that ensure that any truth-setting assignment is a SMCM, i.e., it has no directed cycles and that every edge has at least one arrowhead (Lines 7 and 8 respectively).

1. Other options would be to keep all non-conflicting arrows, or keep non-conflicting arrows and tails after some additional analysis on definitely visible edges (see Zhang (2008b), Borboudakis et al. (2012) for more on this subject). These options are asymptotically correct and would constrain search even further. Nevertheless, orientation rules in FCI seem to be prone to error propagation and we chose a more conservative strategy giving a chance to the conflict resolution strategy to improve the learning quality. Naturally, if an oracle of conditional independence is available or there is a reason to be confident on certain features, one can opt to make additional orientations.

Algorithm 4: addConstraints

input : \mathcal{H} , $\{\mathcal{P}_i\}_{i=1}^N$, $\{\mathbf{I}_i\}_{i=1}^N$, mpl
output: Φ , list of literals \mathcal{F}

```
1  $\Phi \leftarrow \emptyset$  foreach  $X, Y$  do
2   posIndPaths  $\leftarrow$  paths in  $\mathcal{H}$  of maximum length  $mpl$  that are possibly inducing
   with respect to  $\mathbf{I}_i$ ;
3   foreach  $i$  do
4      $\Phi \leftarrow \Phi \wedge [adjacent(X, Y, \mathcal{P}_i) \leftrightarrow \exists p_{XY} \in \mathbf{posIndPaths} \text{ s. t. } inducing(p_{XY}, i)];$ 
5     if  $X, Y$  are adjacent in  $\mathcal{P}_i$  then add  $adjacent(X, Y, \mathcal{P}_i)$  to  $\mathcal{F}$  else add
      $\neg adjacent(X, Y, \mathcal{P}_i)$  to  $\mathcal{F}$ 
6   end
7    $\Phi \leftarrow \Phi \wedge [\neg ancestor(X, Y) \vee \neg ancestor(Y, X)];$ 
8    $\Phi \leftarrow \Phi \wedge [\neg tail(X, Y) \vee \neg tail(Y, X)] \wedge [(arrow(X, Y) \vee tail(X, Y));$ 
9 end
10 foreach  $i$  do
11   foreach unshielded triple in  $\mathcal{P}_i$  do
12      $\Phi \leftarrow \Phi \wedge [dnc(X, Y, Z, \mathcal{P}_i) \rightarrow unshielded\_dnc(X, Y, Z, \mathcal{P}_i)];$ 
13      $\Phi \leftarrow \Phi \wedge [collider(X, Y, Z, \mathcal{P}_i) \rightarrow unshielded\_collider(X, Y, Z, \mathcal{P}_i)];$ 
14     if  $\langle X, Y, Z \rangle$  is a collider in  $\mathcal{P}_i$  then add  $collider(X, Y, Z, \mathcal{P}_i)$  to  $\mathcal{F}$  else add
      $dnc(X, Y, Z, \mathcal{P}_i)$  to  $\mathcal{F}$ 
15   end
16   foreach discriminating path  $p_{WZ} = \langle W, \dots, X, Y, Z \rangle$  do
17      $\Phi \leftarrow \Phi \wedge [dnc(X, Y, Z, \mathcal{P}_i) \rightarrow discriminating\_dnc(p_{WZ}, Y, \mathcal{P}_i)];$ 
18      $\Phi \leftarrow \Phi \wedge [collider(X, Y, Z, \mathcal{P}_i) \rightarrow discriminating\_collider(p_{WZ}, Y, \mathcal{P}_i)];$ 
19     if  $X, Y, Z$  is a collider in  $\mathcal{P}_i$  then add  $collider(X, Y, Z, \mathcal{P}_i)$  to  $\mathcal{F}$  else add
      $dnc(X, Y, Z, \mathcal{P}_i)$  to  $\mathcal{F}$ 
20   end
21 end
```

The constraints are realized on the basis of the *plausible* configurations of \mathcal{H} : Thus, for the constraints corresponding to $adjacent(X, Y, i)$ the algorithm finds all paths between X and Y in \mathcal{H} that are possibly inducing. Then, for the literal $adjacent(X, Y, i)$ to be true, at least one of these paths is constrained to be inducing; for the opposite, none of these paths is allowed to be inducing. This step is the most computationally expensive part of the algorithm. The parameter mpl controls the length of the possibly inducing paths; instead of finding *all* paths between X and Y that are possibly inducing, the algorithm looks for all paths of length at most mpl . This plays a major part in the ability of the algorithm to scale up, since finding all possible paths between every pair of variables can blow up even in relatively small networks, particularly in the presence of unoriented cliques or in relatively dense networks.

Notice that the information on manipulations is included in the satisfiability instance through the encoding of the constraints: For every adjacency between X and Y observed in \mathcal{P}_i , the plausible inducing paths are consistent with the respective intervention targets: No

inducing path is allowed to include an edge that is incoming to a manipulated variable. For example, in Figure 5 $X15$ and $X14$ are adjacent in \mathcal{P}_3 , where $X15$ is manipulated. Since no information concerning experiments is employed up to the initialization of the search graph, $X15 \circ \rightarrow X14$ is in the initial search graph \mathcal{H} , and the edge is a possible inducing path for $X15$ and $X14$ in \mathcal{P}_3 . However, since $X15$ is manipulated in \mathcal{P}_3 , the edge cannot have an arrow into $X15$. This is imposed by the constraint:

$$\begin{aligned} & \text{inducing}(\langle X15, X14 \rangle, 3) \leftrightarrow \\ & (X15 \in \mathbf{I}_3 \rightarrow \text{tail}(X14, X15)) \wedge (X14 \in \mathbf{I}_3 \rightarrow \text{tail}(X15, X14)) \wedge \text{edge}(X14, X15) \end{aligned}$$

which is then added to Φ as

$$\text{inducing}(X15, X14, 3) \leftrightarrow \text{tail}(X14, X15) \wedge \text{edge}(X14, X15).$$

Thus, in any SMCM \mathcal{S} that satisfies the final formula of Algorithm 2, if $\text{inducing}(\langle X15, X14 \rangle, 3)$ is true, the edge will be consistent with the manipulation information.

As mentioned above, in the absence of statistical errors, all the constraints stemming from all PAGs \mathcal{P}_i are simultaneously satisfiable. In practical settings however, it is possible that some of the PAGs have some erroneous features due to statistical errors, and these features can lead to conflicting constraints. To tackle this problem, Algorithm 4 using the following technique: For every observed feature, instead of imposing the implied constraints on the formula Φ , the algorithm adds a bi-conditional connecting the feature to the constraints. For example, if X and Y are found adjacent in \mathcal{P}_i , then instead of adding the constraints $\exists p_{XY} : \text{inducing}(X, Y, i)$ to Φ , we add the bi-conditional $\text{adjacent}(X, Y, \mathcal{P}_i) \leftrightarrow \exists p_{XY} : \text{inducing}(X, Y, i)$. The antecedents of the conditionals are stored in a list of literals \mathcal{F} . The conflict resolution strategy is then imposed on this list of literals, selecting a subset F' that results in a satisfiable SAT formula $\Phi \wedge F'$. The formula $\Phi \wedge F'$ is expressed in Conjunctive Normal Form (CNF) so it can be input to standard SAT solvers.

without imposing the antecedent. These semantics should always be guaranteed and thus, Φ forms a set of hard-constraints. In contrast, if the list of antecedents in \mathcal{F} leads to a conflict, one can select only a subset of antecedents to satisfy (soft-constraints).

Recall that the propositional variables of Φ correspond to the features of the actual underlying SMCM (its edges and endpoints). Some of these variables have the same value in all the possible truth-setting assignments of $\Phi \wedge F'$, meaning the respective features are invariant in all possible underlying SMCMs. Such variables are called **backbone** variables of $\Phi \wedge F'$ (Hytinen et al., 2013). The actual value of a backbone variable is called the polarity of the variable. For sake of brevity, we say an edge or endpoint has polarity 0/1 if the corresponding variable is a backbone variable in $\Phi \wedge F'$ and has polarity 0/1. Based on the backbone of $\Phi \wedge F'$, the final step of COMBINE is to construct the summary graph \mathcal{S} . \mathcal{S} has the following types of edges and endpoints:

- **Solid Edges:** in \mathcal{H} that have polarity 1 in $\Phi \wedge F'$, meaning that they are present in all possible underlying SMCM.
- **Absent Edges:** Edges that are not in \mathcal{H} or edges in \mathcal{H} that have polarity 0 in Φ , meaning that they are absent in all possible underlying SMCM.

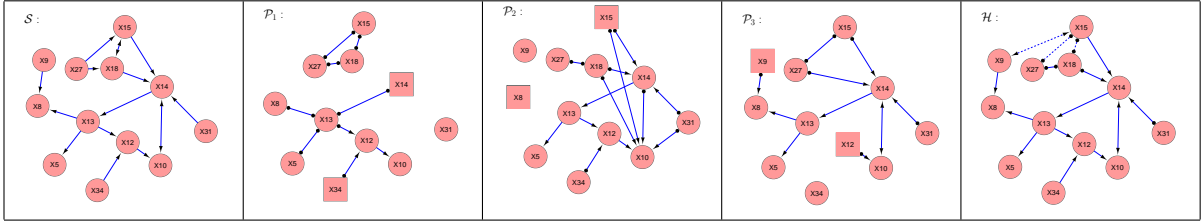


Figure 5: **An example of COMBINE input - output.** Graph \mathcal{S} is the actual, data-generating, underlying SMCM over 12 variables. PAGs $\mathcal{P}_1, \mathcal{P}_2$ and \mathcal{P}_3 are the output of FCI run with an oracle of conditional independence on three different marginals of \mathcal{G} . \mathcal{H} is the output of COMBINE algorithm. The sets of latent variables (with respect to the union of observed variables) per data set are: $\mathbf{L}_1 = \{X9\}$, $\mathbf{L}_2 = \{\emptyset\}$, $\mathbf{L}_3 = \{X18\}$. The sets of manipulated variables (annotated as rectangle nodes instead of circles in the respective graphs) are: $\mathbf{I}_1 = \{X14, X34\}$, $\mathbf{I}_2 = \{X15, X8\}$, $\mathbf{I}_3 = \{X9, X12\}$. Notice that $X10$ and $X31$ are adjacent in \mathcal{P}_2 , but not in \mathcal{P}_1 or \mathcal{P}_3 . This happens because there exists an inducing path in the underlying SMCM ($X31 \rightarrow X14 \leftrightarrow X10$ in \mathcal{S}) that is “broken” by the manipulation of $X14$ and $X12$, respectively. Also notice a dashed edge between $X9$ and $X15$, which cannot be excluded since the variables have never been observed unmanipulated together. Even if the link existed, it would be destroyed in both \mathcal{P}_2 and \mathcal{P}_3 , where both variables are observed. All graphs were visualized in Cytoscape (Smoot et al., 2011).

- **Dashed Edges:** Edges in \mathcal{H} that are not backbone variables in $\Phi \wedge \mathcal{F}'$, meaning that there exists at least one possible underlying SMCM where this edge is present and one where this edge is absent.
- **Solid Endpoints:** Endpoints in \mathcal{H} that are backbone variables in $\Phi \wedge \mathcal{F}'$, meaning that this orientation is invariant in all possible underlying SMCMs.
- **Dashed (circled) Endpoints:** Endpoints in \mathcal{H} that are not backbone variables in $\Phi \wedge \mathcal{F}'$, meaning that there exists at least one SMCM where this orientation does not hold.

We use the term **solid features** of the summary graph to denote the set of solid edges, absent edges and solid endpoints of the summary graph.

Overall, *Algorithm 2* takes as input a set of data sets and a list of parameters and outputs a summary graph that has all invariant edges and orientations of the SMCMs that satisfy as many constraints as possible (according to some strategy). The algorithm is capable of non-trivial inferences, like for example the presence of a solid edge among variables never measured together. Figures 5 and 6 illustrate the output of Algorithm 2, along with the corresponding input PAGs. For an oracle of conditional independence, Algorithm 2 is sound and complete in the manner described in Theorem 13. Lemmas 10 to 12 are necessary for the proof of soundness and completeness: Lemma 10 proves that the possibly inducing and ancestral paths employed by COMBINE are complete: for any consistent \mathcal{S} , if p is a path

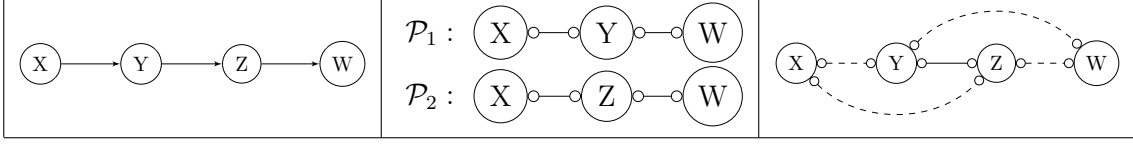


Figure 6: **A detailed example of a non-trivial inference.** From left to right: The true underlying SMCM over variables X, Y, Z, W ; PAGs \mathcal{P}_1 and \mathcal{P}_2 over $\{X, Y, W\}$ and $\{X, Z, W\}$, respectively; The output \mathcal{H} of Algorithm 2 ran with an oracle of conditional independence. Notice that, the edges in \mathcal{P}_1 can not both simultaneously occur in a consistent SMCM \mathcal{S} : This would make $X \circ - \circ Y \circ - \circ W$ an inducing path for X and W with respect to $\mathbf{L}_2 = \{Y\}$ and contradict the features of \mathcal{P}_2 , where X and W are not adjacent. Similarly, $X \circ - \circ Z \circ - \circ W$ cannot occur in any consistent SMCM \mathcal{S} . The only possible edge structures that explain all the observed adjacencies and definite non colliders are $X \circ - \circ Y \circ - \circ Z \circ - \circ W$ or $X \circ - \circ Z \circ - \circ Y \circ - \circ W$. Either way, Y and Z share an edge in all consistent SMCMs, and the algorithm will predict a solid edge between Y and Z , even if the two have not been measured in the same data set. This example is discussed in detail in (Tsamardinos et al., 2012).

that is inducing with respect to a set \mathbf{L} (ancestral) in \mathcal{S} , p is possibly inducing with respect to \mathbf{L} (possibly ancestral) in the initial search graph \mathcal{H} , and will therefore be considered during Algorithm 4. This also implies that if there exists no possibly inducing (ancestral) path in \mathcal{H} there exists no inducing (ancestral) in \mathcal{S} . Lemma 11 proves that any consistent SMCM \mathcal{S} satisfies the final formula $\Phi \wedge F'$ of Algorithm 2, and Lemma 12 proves that any truth-setting assignment of the final formula $\Phi \wedge F'$ corresponds to a consistent SMCM \mathcal{S} .

Lemma 10 *Let $\{\mathcal{P}_i\}_{i=1}^N$ be a set of PAGs and \mathcal{S} a SMCM such that \mathcal{S} is possibly underlying for $\{\mathcal{P}_i\}_{i=1}^N$, and let \mathcal{H} be the initial search graph returned by Algorithm 3 for $\{\mathcal{P}_i\}_{i=1}^N$. Then, if p is an ancestral path in \mathcal{S} , then p is a possibly ancestral path in \mathcal{H} . Similarly, if p is a possibly inducing path with respect to \mathbf{L} in \mathcal{S} , then p is a possibly inducing path with respect to \mathbf{L} in \mathcal{H} .*

Proof See Appendix 6. ■

Lemma 11 *Let $\{\mathbf{D}_i\}_{i=1}^N$ be a set of data sets over overlapping subsets of \mathbf{O} , and $\{\mathbf{I}_i\}_{i=1}^N$ be a set of (possibly empty) intervention targets such that $\mathbf{I}_i \subset \mathbf{O}_i$ for each i . Let \mathcal{P}_i be output PAG of FCI for data set \mathbf{D}_i , $\Phi \wedge \mathcal{F}'$ be the final formula of Algorithm 2, and \mathcal{S} be a possibly underlying SMCM for $\{\mathcal{P}_i\}_{i=1}^N$. Then \mathcal{S} satisfies $\Phi \wedge \mathcal{F}'$.*

Proof See Appendix 6. ■

Lemma 12 *Let $\{\mathbf{D}_i\}_{i=1}^N$, $\{\mathbf{I}_i\}_{i=1}^N$, $\{\mathcal{P}_i\}_{i=1}^N$, $\Phi \wedge \mathcal{F}'$ be defined as in Lemma 11. If graph \mathcal{S} satisfies $\Phi \wedge \mathcal{F}'$, then \mathcal{S} is a possibly underlying SMCM for $\{\mathcal{P}_i\}_{i=1}^N$.*

Proof See Appendix 6. ■

Theorem 13 (*Soundness and completeness of Algorithm 2*) Let $\{\mathbf{D}_i\}_{i=1}^N$, $\{\mathbf{I}_i\}_{i=1}^N$, $\{\mathbf{P}_i\}_{i=1}^N$, $\Phi \wedge \mathcal{F}'$ be defined as in Lemma 11. Finally, let \mathcal{H} be the summary graph returned by COMBINE. Then the following hold:

Soundness: If a feature (edge, absent edge, endpoint) is solid in \mathcal{H} , then this feature is present in all consistent SMCMs.

Completeness: If a feature is present in all consistent SMCMs, the feature is solid in \mathcal{H} .

Proof *Soundness:* Solid features correspond to backbone variables. By Lemma 11 every possible underlying SMCM \mathcal{S} for $\{\mathbf{P}_i\}_{i=1}^N$ satisfies the final formula $\Phi \wedge \mathcal{F}'$. Thus, if a core variable has the same value in all the possible truth-setting assignments of $\Phi \wedge \mathcal{F}'$, this feature is present in all possible underlying SMCMs. *Completeness:* By Lemma 12 the final formula $\Phi \wedge \mathcal{F}'$ of Algorithm 2 is satisfied only by possibly underlying SMCMs. Thus, if a core variable is present in *all* consistent SMCMs, the corresponding core variable will be a backbone variable for $\Phi \wedge \mathcal{F}'$. ■

4.3 A strategy for conflict resolution based on the Maximum MAP Ratio

In this section, we present a method for assigning a measure of confidence to every literal in list \mathcal{F} described in Algorithm 2, and a strategy for selecting a subset of non-conflicting constraints. List \mathcal{F} includes four types of literals, expressing different statistical information:

1. $adjacent(X, Y, \mathcal{P}_i)$: X and Y are independent given some $\mathbf{Z} \subset \mathbf{O}_i$
2. $\neg adjacent(X, Y, \mathcal{P}_i)$: X and Y are not independent given any subset of \mathbf{O}_i .
3. $collider(X, Y, Z, \mathcal{P}_i)$: Y is in no subset of \mathbf{O}_i that renders X and Z independent.
4. $dnc(X, Y, Z, \mathcal{P}_i)$: Y is in every subset of \mathbf{O}_i that renders X and Z independent.

For the scope of this work, we will focus on ranking the first two types of antecedents: Adjacencies and non-adjacencies. We will then assign unshielded colliders and non-colliders to the same rank as the non-adjacency of the triple's endpoints; similarly, discriminating colliders and non-colliders will be assigned to the same rank as the non-adjacency of the path's endpoints. Naturally, this criterion of sorting colliders and non-colliders is merely a heuristic, as more than one tests of independence are involved in deciding that a triple is a (non) collider.

Assigning a measure of likelihood or posterior probability to every single (non) adjacency would enable their comparison. A non-adjacency in a PAG corresponds to a conditional independence given some subset of the observed variables. In contrast, an adjacency corresponds to the lack of such a subset. Thus, an edge between X and Y should be present in \mathcal{P}_i if the evidence (data) is less in favor of hypothesis:

$$H_0 : \exists \mathbf{Z} \subset \mathbf{O}_i : Ind(X, Y | \mathbf{Z}) \text{ than the alternative } H_1 : \nexists \mathbf{Z} \subset \mathbf{O}_i : Ind(X, Y | \mathbf{Z}) \quad (1)$$

This is a complicated set of hypotheses, that involves multiple tests of independence. We try to approximate testing by using a single test of independence as a surrogate: During FCI, several conditioning sets are tested for every pair of variables X and Y . Let \mathbf{Z}_{XY} be the conditioning test for which the highest p-value is identified for the given pair of variables. Notice that it is this maximum p-value that is employed in FCI and similar algorithms to determine whether an edge is included in the output or not. We use the set of hypotheses

$$H_0 : Ind(X, Y | \mathbf{Z}_{XY}) \text{ against the alternative } H_1 : \neg Ind(X, Y | \mathbf{Z}_{XY})$$

as a surrogate for the set of hypotheses in Equation 1. Under the null hypothesis, the p-values follow a uniform $\mathcal{U}([0, 1])$ distribution², also known as the $Beta(1, 1)$ distribution. Under the alternative hypothesis, the density of the p-values should be decreasing in p . One class of decreasing densities is the $Beta(\xi, 1)$ distribution for $0 < \xi < 1$, with density $f(p|\xi) = \xi p^{\xi-1}$. Thus, we can approximate the null and alternative hypotheses in terms of the p-value as

$$H_0 : p_{XY.Z} \sim Beta(1, 1) \text{ against } H_1 : p_{XY.Z} \sim Beta(\xi, 1) \text{ for some } \xi \in (0, 1). \quad (2)$$

Taking the Beta alternatives was presented as a method for calibrating p-values in Sellke et al. (2001). For the purpose of this work, we use them to estimate whether dependence is more probable than independence for a given p-value p , by estimating which of the Beta alternatives it is most likely to follow.

Let \mathcal{F} be a set of M literals corresponding to adjacencies and non-adjacencies, and $\{p_j\}_{j=1}^M$ the respective maximum p-values: If the j -th literal in \mathcal{F} is $(\neg)adjacent(X, Y, \mathcal{P}_i)$, then p_j is the maximum p-value obtained for X, Y during FCI over \mathbf{D}_i . We assume that this population of p-values follows a mixture of $Beta(\xi, 1)$ and $Beta(1, 1)$ distribution. If π_0 is the proportion of p-values following $Beta(\xi, 1)$, the probability density function is

$$f(p|\xi, \pi_0) = \pi_0 + (1 - \pi_0)\xi p^{\xi-1}$$

and the likelihood for a set of p-values $\{p_j\}_{j=1}^M$ is

$$L(\xi, \pi_0) = \prod_j (\pi_0 + (1 - \pi_0)\xi p_j^{\xi-1}).$$

The respective negative log likelihood is

$$-LL(\xi, \pi_0) = -\sum_j \log(\pi_0 + (1 - \pi_0)\xi p_j^{\xi-1}). \quad (3)$$

For given estimates $\hat{\pi}_0$ and $\hat{\xi}$, the MAP ratio of H_0 against H_1 is

$$E_0(p) = \frac{P(p|H_0)P(H_0)}{P(p|H_1)P(H_1)} = \frac{P(p|p \sim Beta(1, 1))P(p \sim Beta(1, 1))}{P(p|p \sim Beta(\hat{\xi}, 1))P(p \sim Beta(\hat{\xi}, 1))} = \frac{\hat{\pi}_0}{\hat{\xi} p^{\hat{\xi}-1} (1 - \hat{\pi}_0)}.$$

2. This is actually an approximation in this case, since these p-values are maximum p-values over several tests.

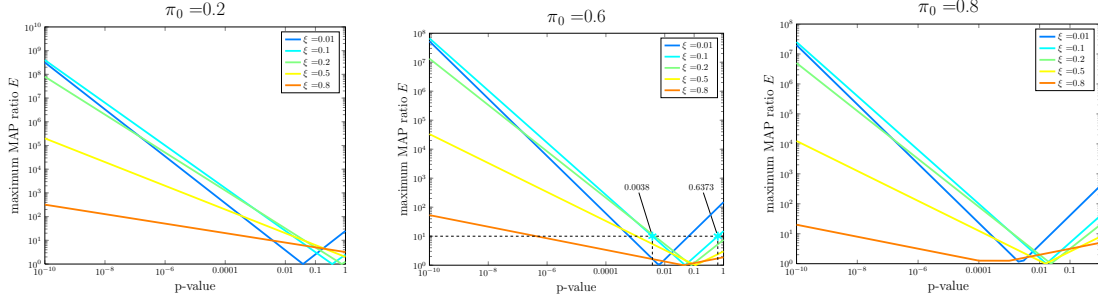


Figure 7: **Log of the maximum map ratio $E(p)$ versus log of the p-value p for various $\hat{\pi}_0$ and $\hat{\xi}$.** For $\hat{\pi}_0 = 0.6$ and $\hat{\xi} = 0.1$, an adjacency supported by a maximum p-value of 0.0038 corresponds to the same E as a non-adjacency supported by a p-value of 0.6373. The intersection point of the line with the x axis is the p for which $E_0(p) = E_1(p) = 1$.

$E_0(p) > 1$ implies that for the test of independence represented by the p-value p , independence is more probable than dependence, while $E_0(p) < 1$ implies the opposite. Moreover, the value of $E_0(p)$ *quantifies* this belief. Conversely, the corresponding MAP ratio of H_1 against H_0 is

$$E_1(p) = \frac{\hat{\xi} p^{\hat{\xi}-1} (1 - \hat{\pi}_0)}{\hat{\pi}_0}.$$

We define the **maximum MAP ratio (MMR)** for a p-value p to be the maximum between the two:

$$E(p) = \max\left\{\frac{\hat{\pi}_0}{\hat{\xi} p^{\hat{\xi}-1} (1 - \hat{\pi}_0)}, \frac{\hat{\xi} p^{\hat{\xi}-1} (1 - \hat{\pi}_0)}{\hat{\pi}_0}\right\}. \quad (4)$$

MMR estimates heuristically quantify our confidence in the observed adjacencies and non-adjacencies and are employed to create a list of literals as follows: Let X and Y be a pair of observed variables, and p_{XY} be the maximum p-value reported during FCI for these variables. Then, if $E_0(p_{XY}) > E_1(p_{XY})$, the literal $\neg adjacent(X, Y, i)$ is added to \mathcal{F} with confidence estimate $E(p_{XY})$. Otherwise, the literal $adjacent(X, Y, i)$ is added to \mathcal{F} with a confidence estimate $E(p_{XY})$. The list can then be sorted in order of confidence, and the literals can be satisfied incrementally. Whenever a literal in the list is encountered that cannot be satisfied in conjunction with the ones already selected, it is ignored.

Notice that, it is possible that for a p-value $E_0(p_{XY}) > E_1(p_{XY})$ (i.e., MMR determines independence is more probable), even though p_{XY} is smaller than the FCI threshold used. In other words, given a fixed FCI threshold, dependence maybe accepted; but, when analyzing the set of p-values encountered to compute MMR, independence seems more probable. The reverse situation is also possible. The pseudo-code in Algorithm 5 (Lines 6–10) accepts the MMR decisions for dependencies and independencies; *this is equivalent to dynamically readjusting the decisions made by FCI*. Nevertheless, in anecdotal experiments we found that

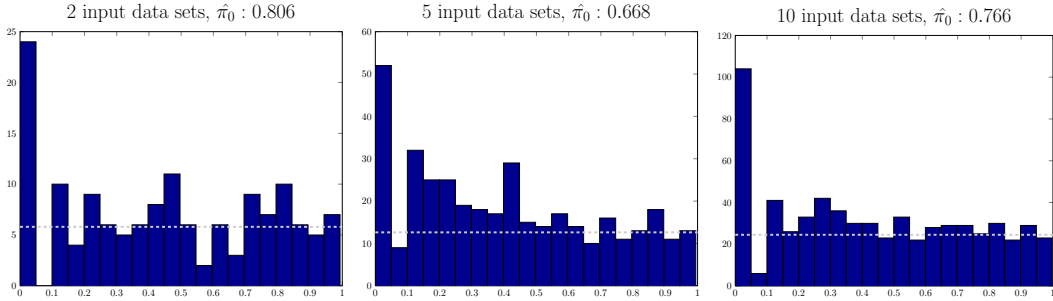


Figure 8: **Distribution of p-values and estimated $\hat{\pi}_0$.** We used the method of Storey and Tibshirani (2003) to estimate $\hat{\pi}_0$ for a sample of p-values corresponding to 2 (left), 5 (center) and 10 (right) input data sets. We generated networks by manipulating a marginal of the ALARM network (Beinlich et al., 1989) consisting of 14 variables. In each experiment, at most 3 variables were latent and at most 2 variables were manipulated. We simulated data sets of 100 samples each from the resulting manipulated graphs. We ran FCI on each data set with $\alpha = 0.1$ and $maxK = 5$ and cached the maximum p-value reported for each pair of variables. We used the p-values from all data sets to estimate $\hat{\pi}_0$. The dashed line corresponds to the proportion of p-values that come from the null distribution based on the estimated $\hat{\pi}_0$.

the literals for which this situation occurs are near the end of the sorted list; thus, whether one accepts the initial decisions of FCI based on a fixed threshold, or a dynamic threshold based on MMR usually does not have a large impact on the output of the algorithm.

Figure 7 shows how the MMR varies with the p-value for several combinations of $\hat{\pi}_0$ and $\hat{\xi}$. The lowest possible value of the MMR is 1, and corresponds to the p-value p for which $E_0(p) = E_1(p)$. Naturally, for the same ξ , this p-value (where the odds switch in favor of non-adjacency) is larger for a lower π_0 . In Figure 7 for $\pi_0 = 0.6$ we can see an example of two p-values that correspond to the same E : An adjacency represented by a p-value of 0.0038 (0.0038 being the *maximum* p-value of any test performed by FCI for the pair of variables) is as likely as a non-adjacency represented by a p-value of 0.6373 (0.6373 being the p-value based on which FCI removed this edge).

To obtain MMR estimates, we need to estimate π_0 and ξ . We used the method described in Storey and Tibshirani (2003) to estimate π_0 on the pooled (maximum) p-values $\{p_j\}_{j=1}^M$ over all data-sets obtained during FCI. For a given $\hat{\pi}_0$, Equation 3 can then be easily optimized for ξ .

The method used to obtain $\hat{\pi}_0$ assumes independent p-values, which is of course not the case since the test schedule of FCI depends on previous decisions. In addition, each p-value may be the maximum of several p-values; these maximum p-values may not follow a uniform distribution even when the non-adjacency (null) hypothesis is true. Finally, given that p-values stem from tests over different conditioning set sizes, p-values corresponding

Algorithm 5: MMRstrategy

input : SAT formula Φ , list of literals \mathcal{F} , their corresponding p-values $\{p_j\}$

output: List of non conflicting literals \mathcal{F}'

```
1  $\mathcal{F}' \leftarrow \emptyset$ ;  
2 Estimate  $\hat{\pi}_0$  from  $\{p_j\}$  using the method described in Storey and Tibshirani (2003);  
3 Find  $\hat{\xi}$  that minimizes  $-\sum_j \log(\hat{\pi}_0 + (1 - \hat{\pi}_0)\xi p_j^{\xi-1})$ ;  
4 foreach literal  $(\neg)adjacent(X, Y, \mathcal{P}_i) \in \mathcal{F}$  with p-value  $p_j$  do  
5    $E_0(p_j) \leftarrow \frac{\hat{\pi}_0}{\hat{\xi} p_j^{\hat{\xi}-1} (1-\hat{\pi}_0)}$ ,  $E_1(p_j) \leftarrow \frac{\hat{\xi} p_j^{\hat{\xi}-1} (1-\hat{\pi}_0)}{\hat{\pi}_0}$ ;  
6   if  $E_1(p_j) < E_0(p_j)$  then  
7     | add  $\neg adjacent(X, Y, \mathcal{P}_i)$  to  $\mathcal{F}$   
8   else  
9     | add  $adjacent(X, Y, \mathcal{P}_i)$  in  $\mathcal{F}$   
10  end  
11   $Score(literal) \leftarrow \max\{E_0(p_j), E_1(p_j)\}$ ;  
12 end  
13 foreach literal collider  $(X, Y, Z, \mathcal{P}_i)$ ,  $dnc(X, Y, Z, \mathcal{P}_i)$  do  
14   | if  $X, Y, Z$  is an unshielded triple in  $\mathcal{P}_i$  then  
15     |  $Score(literal) \leftarrow Score(X, Z, \mathcal{P}_i)$ ;  
16   else if  $\langle W \dots X, Y, Z \rangle$  is discriminating for  $Y$  in  $\mathcal{P}_i$  then  
17     |  $Score(literal) \leftarrow Score(W, Z, \mathcal{P}_i)$ ;  
18   end  
19 end  
20  $\mathcal{F} \leftarrow$  sort  $\mathcal{F}$  by descending score;  
21 foreach  $\phi \in \mathcal{F}$  do  
22   | if  $\Phi \wedge \phi$  is satisfiable then  
23     |  $\Phi \leftarrow \Phi \wedge \phi$ ;  
24     | Add  $\phi$  to  $\mathcal{F}'$   
25   end  
26 end
```

to adjacencies do not necessarily follow the same beta distribution. Thus, the approach presented here is at best an approximation.

In the algorithm as presented, a single beta is fit from the pooled p-values of FCI runs over all data-sets. This strategy is perhaps more appropriate when individual data-sets have a small number of p-values, so the pooled set provides a larger sample size for the fitting. Other strategies though, are also possible. One could instead fit a different beta for each data-set and its corresponding set of p-values. This approach could perhaps be more appropriate in case the PAG structures \mathcal{P}_i vary greatly in terms of sparseness. In addition, one could also fit different beta distributions for each conditioning set size. Figure 8 shows the empirical distribution of p-values and the estimated $\hat{\pi}_0$ based on the p-values returned from FCI on 2, 5 and 10 input data sets, simulated from a network of 14 variables.

The strategy for selecting non-conflicting constraints based on the MMR strategy is presented in Algorithm 5. MMR is a general criterion that can be used to compare confidence in dependencies and independencies. The method is based on p-values and thus, can be applied in different types of data (e.g., continuous and discrete) in conjunction with any appropriate test of independence. Moreover, since it is based on cached p-values, and fitting a beta distribution is efficient, it adds minimal computational complexity. On the other hand, the estimation of maximum MAP ratios is based on heuristic assumptions and approximations. Nevertheless, experiments presented in the following section showcase that the method works similarly if not better than other conflict resolution methods, while being orders of magnitude computationally more efficient.

5. Experimental Evaluation

We present a series of experiments to characterize how the behavior of COmbINE is affected by the characteristics of the problem instance and compare it against other alternative algorithm in the literature. We also present a comparative evaluation of conflict resolution methods, including the one based on the proposed MMR estimation technique. Finally, we present a proof-of-concept application on real mass cytometry data on human T-cells. In more detail, we initially compare the complete version of COmbINE (i.e., without restrictions on the maximum path length or the conditioning set) against SBCSD (Hyttinen et al., 2013) in ideal conditions (i.e., both algorithms are provided with an independence oracle). We perform a series of experiments to explore the (a) learning accuracy of COmbINE as a function of the maximum path length considered by the algorithm, the density and size of the network to reconstruct, the number of input data sets, the sample size, and the number of latent variables, and (b) the computational time as a function of the above factors.

All experiments were performed on data simulated from randomly generated networks as follows. The graph of each network is a DAG with a specified number of variables and maximum number of parents per variable. Variables are randomly sorted topologically and for each variable the number of parents is uniformly selected between 0 and the maximum allowed. The parents of each variable are selected with uniform probability from the set of preceding nodes. Each DAG is then coupled with random parameters to generate conditional linear gaussian networks. To avoid very weak interactions, minimum absolute conditional correlation was set to 0.2. Before generating a data set, the variables of the graph are partitioned to unmanipulated, manipulated, and latent. Mean value and standard deviation for the manipulated variables were set to 0 and 1, respectively. Subsequently, data instances are sampled from the network distribution, considering the manipulations and removing the latent variables. All experiments are performed on **conservative** families of targets; the term was introduced in Hauser and Bühlmann (2012) to denote families of intervention targets in which all variables have been observed unmanipulated at least once.

For each invocation of the algorithm, the problem instance (set of data sets) is generated using the parameters shown in Table 1. COmbINE default parameters were set as follows: maximum path length = 3, $\alpha = 0.1$ and maximum conditioning set $maxK = 5$, and the Fisher z-test of conditional independence. As far as orientations are concerned, in our experience, FCI is very prone to error propagation, we therefore used the rule in (Ramsey et al., 2006) for *conservative* colliders. Unless otherwise stated, Algorithm 5 is employed to

Problem attribute	Default value used
Number of variables in the generating DAG	20
Maximum number of parents per variable	5
Number of input data sets	5
Maximum number of latent variables per data set	3
Maximum number of manipulated variables per data set	2
Sample size per data set	1000

Table 1: **Default values used in generating experiments in each iteration of COMbINE.** Unless otherwise stated, the input data sets of COMbINE were generated according to these values.

resolve conflicts. SAT instances were solved using MINISAT2.0 (Eén and Sörensson, 2004) along with the modifications presented in Hyttinen et al. (2013) for iterative solving and computing the backbone with some minor modifications for sequentially performing literal queries. In the subsequent experiments, *one of the problem parameters in Table 1 is varied each time, while the others retain the values above.*

To measure learning performance, ideally one should know the ground truth, i.e., the structure that the algorithm would learn if ran with an oracle of conditional independence, and unrestricted infinite maxK and maximum path length parameters. Notice, that *the original generating DAG structure cannot serve directly as the ground truth.* This is because the presence of manipulated and latent variables implies that not all structural features of the generating DAG can be recovered. For example, for the problem instance presented in Figure 6(middle), the ground truth structure has one solid edge out of 5, no solid endpoint 6(right), one absent, and four dashed edges. Dashed edges and endpoints in the output of the algorithm can only be evaluated if one knows the ground truth structure. Unfortunately, the ground truth structure cannot be recovered in a timely fashion in most problems involving more than 15 variables.

As a surrogate, we defined metrics that do not consider dashed edges or endpoints and can be directly computed by comparing the “solid” features of the output with the original data generating graph. Specifically, we used two types of precision and recall; one for edges (s-Precision/s-Recall) and one for orientations (o-Precision/o-Recall). Let \mathcal{G} be the graph that generated the data (the SMCM stemming from the initial random DAG after marginalizing out variables latent in all data sets), and \mathcal{H} be the summary graph returned by COMbINE. s-Precision and s-Recall were then calculated as follows:

$$\text{s-Precision} = \frac{\# \text{ solid edges in } \mathcal{H} \text{ that are also in } \mathcal{G}}{\# \text{ solid edges in } \mathcal{H}}$$

and

$$\text{s-Recall} = \frac{\# \text{ solid edges in } \mathcal{H} \text{ that are also in } \mathcal{G}}{\# \text{ edges in } \mathcal{G}}.$$

Similarly, orientation precision and recall are calculated as follows:

$$\text{o-Precision} = \frac{\# \text{ endpoints in } \mathcal{G} \text{ correctly oriented in } \mathcal{H}}{\# \text{ of orientations(arrows/tails) in } \mathcal{H}}$$

# variables	# max parents	Running time Median (5 %ile, 95 %ile)			Completed instances/ total instances		
		COmbINE	SBCSD	SBCSD*	COmbINE	SBCSD	SBCSD'
10	3	17 (1, 113)	149 (14, 470)*	91 (30, 369)*	50/50	30/50	48/50
	5	80 (4, 1192)	365 (133, 500)*	264 (68, 554)*	50/50	16/50	32/50
14	3	28 (4, 6361)*	–	451 (407, 492)*	49/50	0/50	4/50
	5	272 (23, 16107)*	–	–	43/50	0/50	0/50

Table 2: **Comparison of running times for COmbINE and SBCSD for networks of 10 and 14 variables.** The table reports the median running time along with the 5 and 95 percentiles, as well as the number of instances (problem inputs) in which each algorithm managed to complete; *numbers are computed only on the problems for which the algorithm completed.

and

$$\text{o-Recall} = \frac{\# \text{ endpoints in } \mathcal{G} \text{ correctly oriented in } \mathcal{H}}{\# \text{ endpoints in } \mathcal{G}}.$$

Since dashed edges and endpoints do not contribute to these metrics, precision in particular could be favorable for conservative algorithms that tend to categorize all edges (endpoints) as dashed. To alleviate this problem, we accompany each precision / recall figure with the percentage of dashed edges out of all edges in the output graph to indicate how conservative is the algorithm. Similarly, we present the percentage of dashed (circled) endpoints out of all endpoints in the output graph. Finally, we note that in the experiments that follow, unless otherwise stated, we report the median, 5, and 95 percentile over 100 runs of the algorithm with the same settings.

5.1 COmbINE vs. SBCSD

Hyttinen et al. (2013) presented a similar algorithm, named SAT-based causal structure discovery (SBCSD). SBCSD is also capable of learning causal structure from manipulated datasets over overlapping variable sets. In addition, if linearity is assumed, it can admit feedback cycles. SBCSD also uses similar techniques for converting conditional (in)dependencies into a SAT instance. However, the algorithm requires all m -connections to constrain the search space (at least the ones that guarantee completeness), while COmbINE uses inducing paths to avoid that. For each adjacency $X \star \rightarrow Y$ in a data set, COmbINE creates a constraint specifying that at least one path between the variables is inducing with respect to \mathbf{L}_i . In contrast, SBCSD creates a constraint specifying that at least one path between the variables is m -connecting path given each possible conditioning set. So, both algorithms are forced to check every possible path, yet COmbINE examines each path once (with respect to \mathbf{L}_i), while SBCSD examines it for multiple possible conditioning sets. The latter choice may be necessary to deal with cyclic structures, but leads to significantly larger SAT problems when acyclicity is assumed.

SBCSD is not presented with a conflict resolution strategy and so it can only be tested by using an oracle of conditional independence. Equipping SBCSD with such a strategy is possible, but it may not be straightforward: SBCSD computes the SAT backbone incrementally

for efficiency, which complicates pre-ranking constraints according to some criterion. Since SBCSD cannot handle conflicts, we compared it to the complete version of our algorithm (infinite maxK and maximum path length) using an oracle of conditional independence. Since no statistical errors are assumed, the initial search graph for COmbINE includes all observed arrows. Both algorithms are sound and complete, hence we only compare running time. SBCSD uses a path-analysis heuristic to limit the number of tests to perform. However, the authors suggest that in cases of acyclic structures, this heuristic could be substituted with the FCI test schedule. To better characterize the behavior of SBCSD on acyclic structures, we equipped the original implementation as suggested³. We denote this version of the algorithm as SBCSD'. Also note, that the available implementation of SBCSD by its authors has an option to restrict the search to acyclic structures, which was employed in the comparative evaluation. Finally, we note that SBCSD is implemented in C, while COmbINE is implemented in Matlab.

For the comparative evaluation, we simulated random acyclic networks with 10 and 14 variables. The default parameters were used to generate 50 problem instances for networks with 3 and 5 maximum parents per variable. Both algorithms were run on the same computer, with 4GB of available memory. SBCSD reached maximum memory and aborted without concluding in several cases for networks of 10 variables, and *in all cases for networks of 14 variables*. SBCSD' slightly improves the running time over SBCSD. Median running time along with the 5 and 95 percentiles as well as number of cases completed are reported in Table 2. The metrics for each algorithm were calculated only on the cases where the algorithm completed.

The results in Table 2 indicate that COmbINE is more time-efficient than SBCSD and SBCSD'. While the running times do depend on implementation, the fact that SBCSD have much higher memory requirements indicates that the results must be at least in part due to the more compact representation of constraints by COmbINE. COmbINE managed to complete all cases for networks of 10 and most cases for 14 variables, while SBCSD completed less than 50% and 0%, respectively. SBCSD' completed most cases for 10 variables but only 4% of cases for 14 variables. Interestingly, the percentiles for COmbINE are quite wide spanning two orders of magnitude for problems with maxParents equal to 5 (we cannot compute the actual 95 percentile for SBCSD since it did not complete for most problems). Thus, performance highly depends on the input structure. Such heavy-tailed distributions are well-noted in the constraint satisfaction literature (Gomes et al., 2000). We also note the fact that COmbINE seems to depend more on the sparsity and less on the number of variables, while SBCSD's time increases monotonically with the number of variables. Based on these results, we would suggest the use of COmbINE for problems where acyclity is a reasonable assumption and the number of variables is relatively high.

5.2 Evaluation of Conflict Resolution Strategies

In this section we evaluate our Maximum Map Ratio strategy (MMR) against three other alternatives: A ranking strategy where constraints are sorted based on Bayesian prob-

3. However, we do not include the Possible d-Separating step of FCI; this step hardly influences the quality of the algorithm Colombo et al. (2012). Thus, the timing results of Table 2 are a lower bound on the execution time of the SBCSD algorithm.

abilities as proposed in Claassen and Heskes (2012) (**BCCDR**), as well as a Max-SAT (**MaxSAT**) and a weighted max-SAT (**wMaxSAT**) approach.

MMR: This strategy sorts constraints according to the Maximum Map Ratio (Algorithm 5) and greedily satisfies constraints in order of confidence; whenever a new constraint is not satisfiable given the ones already selected, it is ignored (lines 21- 25 in Algorithm 5).

BCCDR: BCCDR sorts constraints according to Bayesian probability estimates of the literals in \mathcal{F} as presented in Claassen and Heskes (2012). The same greedy strategy for satisfying constraints in order is employed. Briefly, the authors of (Claassen and Heskes, 2012) propose a method for calculating Bayesian probabilities for any feature of a causal graph (e.g. adjacency, m -connection, causal ancestry). To estimate the probability of a feature, for a given data set \mathbf{D} , the authors calculate the score of all DAGs of N variables. Let $\mathcal{G} \vdash f$ denote that a feature f is present in DAG \mathcal{G} . The probability of the feature is then calculated as $P(f) = \sum_{\mathcal{G} \vdash f} P(\mathbf{D}|\mathcal{G})P(\mathcal{G})$. Scoring all DAGs is practically infeasible for networks with more than 5 or 6 variables. Thus, for data sets with more variables, a subset of variables must be selected for the calculation of the probability of a feature. Following (Claassen and Heskes, 2012), we use 5 as the maximum N attempted.

The literals in \mathcal{F} represent information on adjacencies: $(\neg)adjacent(X, Y, \mathcal{P}_i)$ and colliders: $(\neg)collider(X, Y, Z, \mathcal{P}_i)$. To apply the method above for a given feature, we have to select the variables used in the DAGs, a suitable scoring function, and suitable DAG priors. For (non) adjacencies $X \star \rightarrow Y$ in PAG \mathcal{P}_i , we scored the DAGs over variables X, Y and \mathbf{Z} , for the conditioning set \mathbf{Z} maximizing the p-value of the tests $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ performed by FCI. Since the total number of variables cannot exceed 5, the maximum conditioning set for FCI is limited to 3 in all experiments in this section for a fair comparison. For a (non) collider $X \star \rightarrow Y \star \rightarrow Z$, we score all networks over X, Y and Z .

We use the BGE metric for gaussian distributions (Geiger and Heckerman, 1994) as implemented in the BDAGL package Eaton and Murphy (2007b) to calculate the likelihoods of the DAGs. This metric is score equivalent, so we pre-computed representatives of the Markov equivalent networks of up to 5 nodes, and scored only one network per equivalence class to speed up the method. Priors for the DAGs were also pre-computed to be consistent with respect to the maximum attempted number of nodes (i.e. 5) as suggested in Claassen and Heskes (2012).

MaxSAT: This approach tries to satisfy as many literals in \mathcal{F} as possible. Recall that the SAT problem consists of a set of hard-constraints (conditionals, no cycles, no tail-tail edges), which should always be satisfied (hard constraints), and a set of literals \mathcal{F} . Maximum SAT solvers cannot be directly applied to the entire SAT formula since they do not distinguish between hard and soft constraints. To maximize the number of literals satisfied, while ensuring all hard-constraints are satisfied we resorted to the following technique: we use the `akmaxsat` (Kuegel, 2010) *weighted* max SAT solver that tries to maximize the sum of the weights of the satisfied clauses. Each literal is assigned a weight of 1, and each hard-constraint is assigned a weight equal to the sum of all weights in \mathcal{F} plus 10000. The summary graph returned by Algorithm 2 is based on the backbone of the subset of literals selected by `akmaxsat`.

wMaxSAT: Finally, we augmented the above technique with a different weighted strategy that considers the importance of each literal. Specifically, each literal was weighted proportionally to the logarithm of the corresponding MMR. Again, each hard-constraint

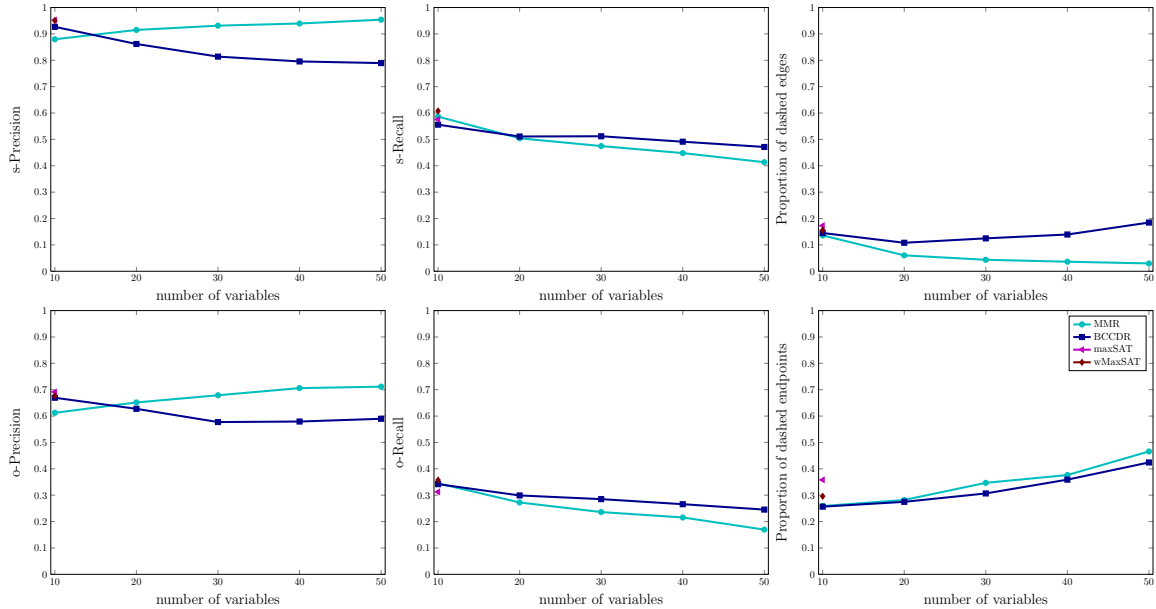


Figure 9: **Learning performance of COMBINE with various conflict resolution strategies.** From left to right: Median s-Precision, s-Recall, proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints (bottom) for networks of several sizes for various conflict resolution strategies. Each data set consists of 100 samples. The numbers for wMaxSAT and maxSAT correspond to 22 and 23 cases, respectively, in which the algorithms managed to return a solution within 500 seconds.

was assigned a weight equal to the sum of all weights in \mathcal{F} plus 10000, to ensure that the solver will always satisfy these statements. The summary graph returned by Algorithm 2 is based on the backbone of the subset of literals selected by akmaxsat.

We ran all methods for networks of 10, 20, 30, 40 and 50 variables for data sets of 100 samples to test them on cases where statistical errors are common. For each network size we performed 50 iterations. **MaxSAT** and **wMaxSAT** often failed to complete in a timely fashion; to complete the experiments we aborted the solver after 500 seconds. We note that this amount of time corresponds to more than 10 times the maximum running time of the MMR method (calculating MMRs and solving the SAT instance), and more than twice times the maximum running time of the BCCDR-based method (for 50 variables). Cases where the solver did not complete were not included in the reported statistics. Unfortunately, *the methods using weighted max SAT solving failed to complete in most cases for 10 variables, and all cases for more than 10 variables.*

The results are shown in Figure 9, where we can see the median performance of both algorithms over 50 iterations. Overall, **MMR** exhibits better Precision and identifies more solid edges, while **BCCDR** exhibits slightly better Recall. **BCCDR** is better for variable size equal to 10, which could be explained from the fact that **MMR** is not provided with sufficient number of p-values to estimate $\hat{\pi}_0$ and $\hat{\xi}$. In terms of computational complexity,

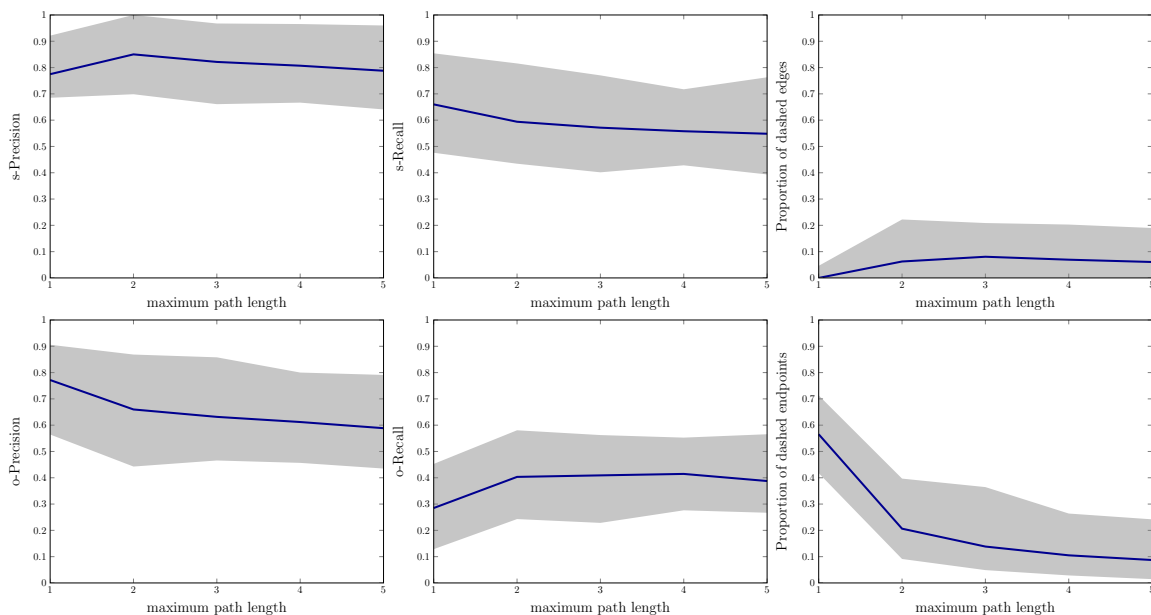


Figure 10: **Learning performance of COmbINE against maximum path length.** From left to right: s-Precision, s-Recall, percentage dashed edges and o-Precision, o-Recall and percentage of dashed endpoints (bottom) for varying maximum path length, averaged over all networks. Shaded area ranges from the 5 to the 95 percentile. Maximum path length 3 seems to be a reasonable trade-off between performance, percentage of dashed features, and efficiency.

for networks of 50 variables, estimating the **BCCDR** ratios takes about 150 seconds on average, while estimating the **MMR** ratios takes less than a second. The more sophisticated search strategies **MaxSAT** and **wMaxSAT** do not seem to offer any significant quality benefits, at least for the single variable size for which we could evaluate them. Based on these results, we believe that **MMR** is a reasonable and relatively efficient conflict resolution strategy.

5.3 COmbINE performance with increasing maximum path length

In this section, we examine the behavior of the algorithm when the length of the paths considered is limited, in which case the output is an approximation of the actual solution. The COmbINE pseudo-code in Algorithm 2 accepts the maximum path length as a parameter. Learning performance as a function of the maximum path length is shown in Figure 10. Notice that when the path length is increased from 1 to 2 there is drop in the percentage of dashed endpoints, implying more orientations are possible. For length equal to 1, only unshielded and discriminating colliders are identified, while for length larger than 2 further orientations become possible thanks to reasoning with the inducing paths. When length is 1, notice that there are almost no dashed edges (except for the edges added in line 2 of Algorithm 3). When the maximum length increases, adjacencies in one data set, can

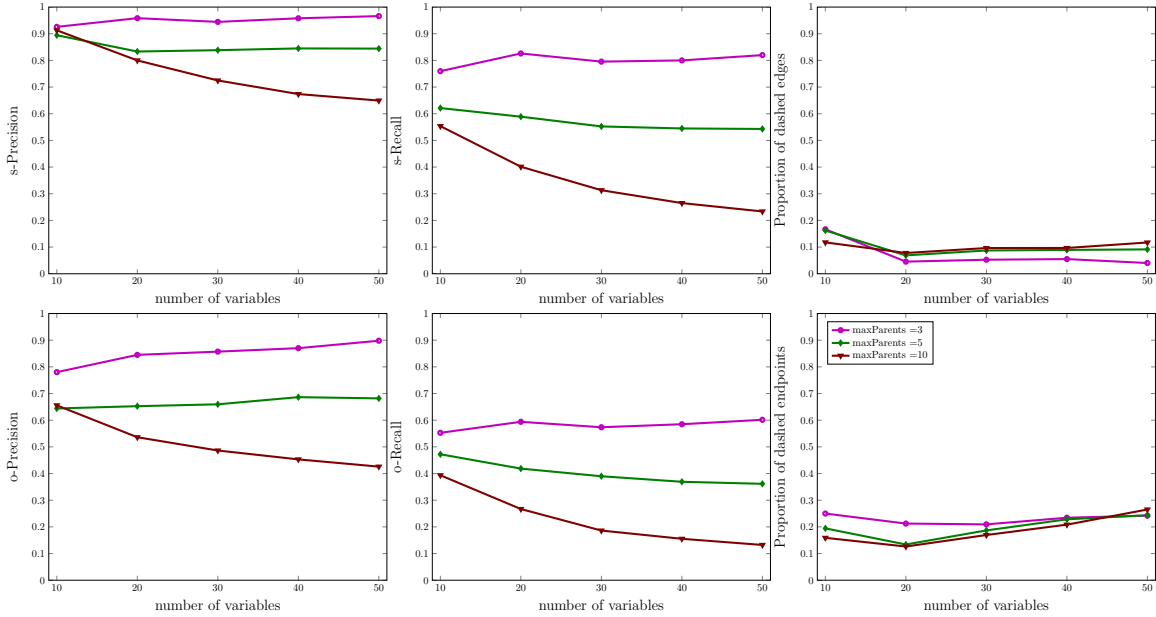


Figure 11: **Learning performance of COMBINE for various network sizes and densities.** From left to right: Median s-Precision, s-Recall, proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints (bottom) for varying network size and density. Density is controlled by limiting the number of possible parents per variable. As expected, the performance deteriorates as networks become denser.

be explained with longer inducing paths in the underlying graph and more dashed edges appear. The learning performance of the algorithm is not monotonic with the maximum length. Explaining an association (adjacency) through the presence of a long inducing path may be necessary for asymptotic correctness. However, in the presence of statistical errors, allowing such long paths could lead to complicated solutions or the propagation of errors.

Overall, it seems any increase of the maximum path length above 3 does not significantly affect performance. It seems that a maximum path length of 3 is a reasonable trade-off among learning performance (precision and recall), percentage of uncertainties, and computational efficiency. These experiments justify our choice of maximum length 3 as the default parameter value of the algorithm.

5.4 COMBINE performance as a function of network density and size

In Figure 11 the learning performance of the algorithm is presented as a function of network density and size. Density was controlled by the maximum parents allowed per variable, set by parameter `maxParents` during the generation of the random networks. For all network sizes, learning performance monotonically decreases with increased density, while the percentage of dashed features does not significantly vary. The size of the network has a smaller

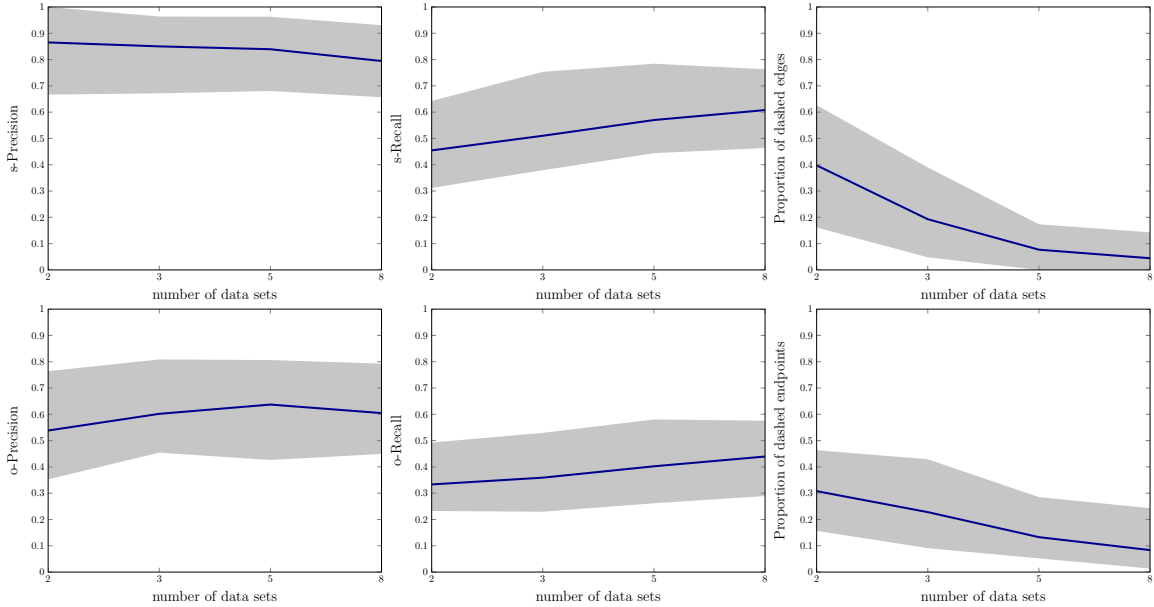


Figure 12: **Learning performance of COMbine for varying number of input data sets.** From left to right: Median s-Precision, s-Recall, Proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints of (bottom) for varying number of input data sets. Shaded area ranges from the 5 to the 95 percentile. Increasing the number of input data sets improves the performance of the algorithm.

impact on the performance, particularly for the sparser networks. For dense networks, performance is relatively poor and becomes worse with larger sizes.

5.5 COMbine performance over sample size and number of input data sets

Figure 12 shows the performance of the algorithm with increasing the number of input data sets. As expected, the percentage of uncertainties (dashed features) is steadily decreasing with increased number of input data sets. Recall also steadily improves, while Precision is relatively unaffected. Figure 13 holds the number of input data set constant to the default value 5, while increasing the sample size per data set. Recall in particular improves with larger sample sizes, while the percentage of dashed endpoints drops.

5.6 COMbine performance for increasing number of latent variables

We also examine the effect of confounding to the performance of COMbine. To do so, we generated semi-Markov causal models instead of DAGs in the generation of the experiments: We generated random DAG networks of 30 variables and then marginalized out a percentage of the variables. Figure 14 depicts COMbine’s performance against 3, 6, and 9 of latent variables, corresponding to 10%, 20% and 30% of the total number of variables in the graph, respectively. Overall, confounding does not seem to greatly affect the performance

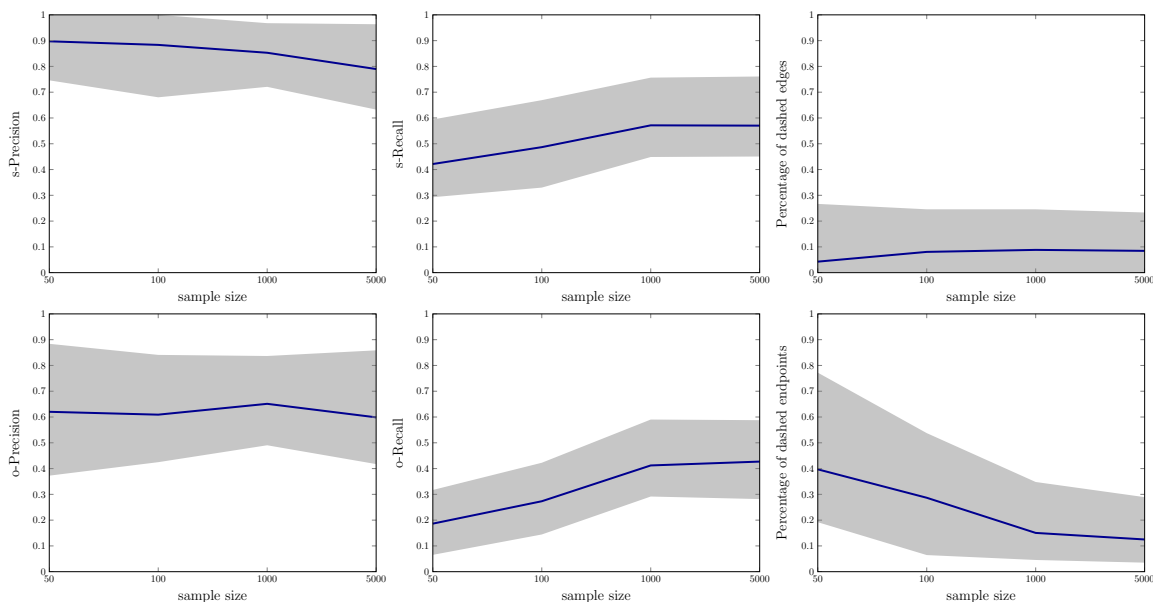


Figure 13: **Learning performance of COmbINE for varying sample size per data set.** From left to right: s-Precision, s-Recall, Proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints of (bottom) for varying sample size per data set. Shaded area ranges from the 5 to the 95 percentile. Increasing the sample size improves the performance of the algorithm.

of COmbINE. We must point out however, that s-Recall is lower than the s-Recall with no confounded variables for the same network size (see Figure 11).

5.7 Running Time for COmbINE

The running time of COmbINE depends on several factors, including the ones examined in the previous experiments: Maximum path length, number of input data sets and sample size, and, naturally, the number of variables. Figure 15 illustrates the running time of COmbINE against these factors. Figure 15 (b) presents the running time of COmbINE against number of variables for networks of 5 maximum parents per variable. The experiments regarding 10 to 50 variables have also been presented in terms of learning performance in section 5.4. To further examine the scalability of the algorithm, *we also ran COmbINE in networks of 100 variables*, with 5 maximum parents per variable. The experiments were ran with the default parameter values. As we can see in Figure 15, the restriction on the maximum path length is the most critical factor for the scalability of the algorithm.

5.8 A case study: Mass Cytometry data

Mass cytometry (Bendall et al., 2011) is a recently introduced technique that enables measuring protein activity in cells, and its main use is to classify hematopoietic cells and identify signaling profiles in the immune system. Therefore, the proteins are usually measured in

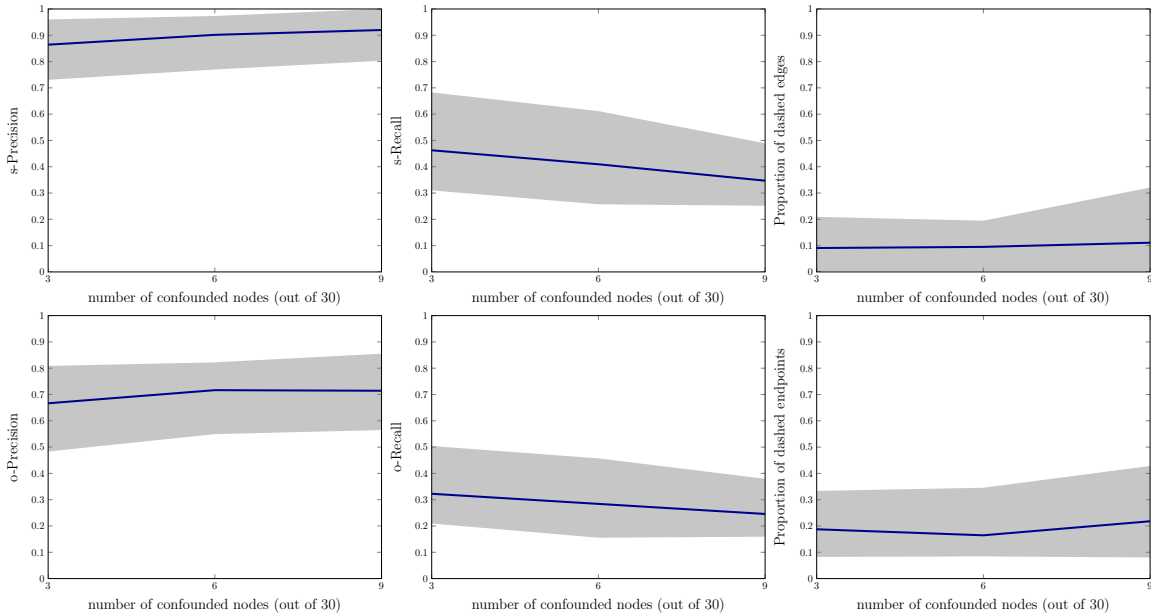


Figure 14: **Learning performance of COMBINE for varying percentage of confounded variables.** From left to right: s-Precision, s-Recall, percentage of dashed edges (top) and o-Precision, o-Recall and percentage of dashed endpoints (bottom) for varying number of confounded nodes for networks of 30 variables. Shaded area ranges from the 5 to the 95 percentile. Overall, the number of confounding variables does not seem to greatly affect the algorithm’s performance.

a sample of cells and then in a different sample of the same (type of) cells after they have been stimulated with a compound that triggers some kind of signaling behavior. Identifying the causal succession of events during cell signaling is crucial to designing drugs that can trigger or suppress immune reaction. Therefore in several studies both stimulated and un-stimulated cells are treated with several perturbing compounds to monitor the potential effect on the signaling pathway.

Mass cytometry data seem to be a suitable test-bed for causal discovery methods: The proteins are measured in single cells instead of representing tissue averages, the latter being known to be problematic for causal discovery (Chu et al., 2003), and the samples range in thousands. However, the mass cytometer can measure only up to 34 variables, which may be too low a number to measure all the variables involved in a signaling pathway. Moreover, about half of these variables are surface proteins that are necessary to distinguish (gate) the cells into sub-populations, but are not functional proteins involved in the signaling pathway. It is therefore reasonable for scientists to perform experiments measuring overlapping variable sets.

Bendall et al. (2011) and Bodenmiller et al. (2012) both use mass cytometry to measure protein abundance in cells of the immune system. In both studies, the samples were treated

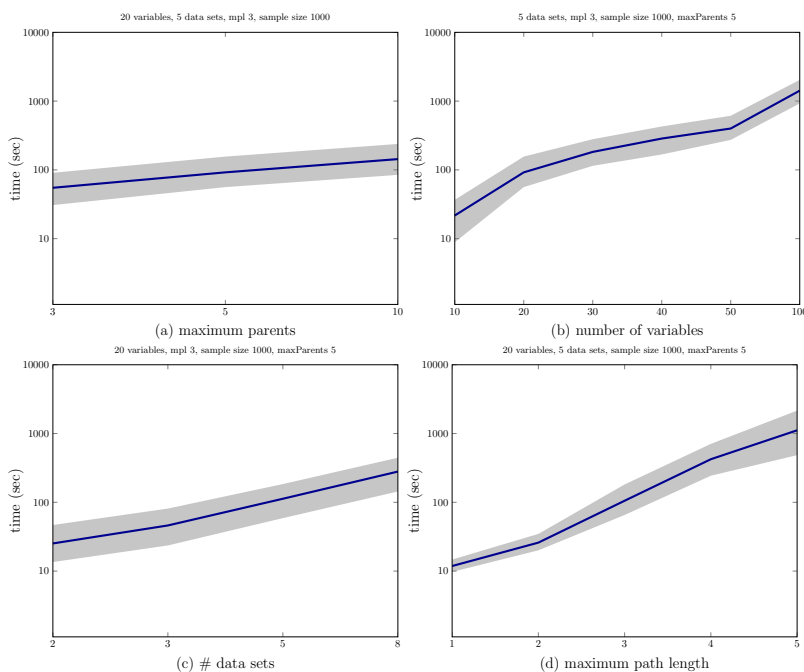


Figure 15: **Running time of COMBINE** . From left to right: Running time (in seconds) is plotted in logarithmic scale against maximum parents per variable and number of variables (top row); number of data sets and maximum path length (bottom row). Shaded area ranges from the 5 to the 95 percentile. The number of variables and the maximum path length seem to be the most critical factors of computational performance. Notice that, COMBINE scales up to problems with 100 total variables for limited path length and relatively sparse networks.

with several different signaling stimuli. Some of the stimuli were common in both studies. After stimulation with each activating compound, Bodenmiller et al. (2012) also test the cell's response to 27 inhibitors. One of these inhibitors is also used in Bendall et al. (2011). For this inhibitor, Bendall et al. (2011) measured bone marrow cell samples of a single donor. In Bodenmiller et al. (2012), measurements were taken from Peripheral blood mononuclear cell samples of a (different) single donor. Despite differences in the experimental setup, the signaling pathway of every stimulus and every sub-population of cells is considered universal across (healthy) donors, so the data should reflect the same underlying causal structure.

We focused on two sup-populations of the cells, CD4+ and CD8+ T-cells, which are known to play a central role in immune signaling. The data were manually gated by the researchers in the original studies. We also focused on one of the stimuli present in both studies, PMA-Ionomycin, which is known to have prominent effects on T-cells. Proteins pBtk, pStat3, pStat5, pNfkb, pS6, pp38, pErk, pZap70 and pSHP2 are measured in both data sets (initial p denotes that the concentration of the phosphorylated protein is measured). Four additional variables were included in the analysis, pAkt, pLat and pStat1 measured only in Bodenmiller et al. (2012) and pMAPK measured only in Bendall et al.

Data set	Source	latent (\mathbf{L}_i):	manipulated(\mathbf{I}_i)	Donor
\mathbf{D}_1	Bodenmiller et al. (2012)	pMAPK	pAkt	1
\mathbf{D}_2	Bodenmiller et al. (2012)	pMAPK	pBtk	1
\mathbf{D}_3	Bodenmiller et al. (2012)	pMAPK	pErk	1
\mathbf{D}_4	Bendall et al. (2011)	pAkt, pLat, pStat1	pErk	2

Table 3: **Summary of the mass cytometry data sets co-analyzed with COMBINE.** The procedure was repeated for two sub-populations of cells, CD4+ cells and CD8+ cells.

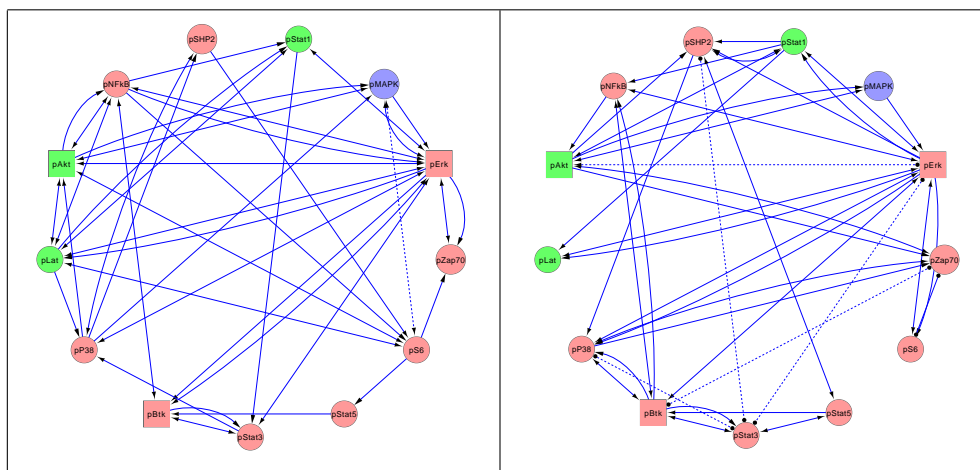


Figure 16: **A case study for COMBINE: Mass cytometry data.** COMBINE was run on 4 different mass cytometry data for two different cell populations: CD4+ T-cells (left) and CD8+ T-cells (right). In each data set, one variable was manipulated (pAkt, pBtk, pErk, pErk respectively). Variables pAkt, pLat and pStat1 are only measured in data sets 1-3, while pMAPK is only measured in data set 4. Notice that pAkt is predicted to be a direct cause of pMAPK in both CD4+ and CD8+ cells, even though the two variables have never been measured together.

(2011). To be able to detect signaling behavior, we formed data sets that contain both stimulated and unstimulated samples. As mentioned above, the cells were treated with several inhibitors. Some of these inhibitors target a specific protein, and some of them perturb the system in a more general or unidentified way. We used three target specific compounds that can be modeled as hard interventions (i.e. the compounds used to target these proteins are known to be specific and to have an effect in the phosphorylation levels of the target). More information on the specific compounds can be found in the respective publications. We ended up with four data sets for each sub-population. Details can be found in Table 3.

Protein interactions are typically non-linear, so we discretized the data into 4 bins. We ran Algorithm 2 with maximum path length 3. We used the G^2 test of independence for

FCI with $\alpha = 0.1$ and $\max K=5$. We used Cytoscape (Smoot et al., 2011) to visualize the summary graphs produced by COmbINE, illustrated in Figure 16.

Unfortunately, the ground truth for this problem is not known for a full quantitative evaluation of the results. Nevertheless, this set of experiments demonstrates the availability of real and important data sets and problems that are suited integrative causal analysis. Second, these experiments provide a proof-of-concept for the specific algorithm. One type of interesting type of inference possible with COmbINE and similar algorithms is the prediction that pAkt is a direct cause of pMAPK in both CD4+ and CD8+ cells, *even though the variables are not jointly measured in any of the input data sets*. Evidence of a direct protein interaction between the two proteins does exist in the literature Rane et al. (2001). Thus, methods for learning causal structure from multiple manipulations over overlapping variables potentially constitute a powerful tool in the field of mass cytometry.

We do not make any claims for the validity of the output graphs and they are presented only as a proof-of-concept, as there are several potential pitfalls. COmbINE assumes lack of feedback cycles, which is not guaranteed in this system (we note however, that acyclic networks have been successfully used for reverse engineering protein pathways in the past (Sachs et al., 2005)). Causal discovery methods that allow cycles Hyttinen et al. (2013) on the other hand rely on the assumption of linearity, which is also known to be heavily violated in such networks. Thus, which set of assumptions best approximates the specific system is unknown.

6. Conclusions and Future Work

We have presented COmbINE, a sound and complete algorithm that performs causal discovery from multiple data sets that measure overlapping variable sets under different interventions in acyclic domains. COmbINE works by converting the constraints on inducing paths in the sought out semi Markov causal model (SMCMs) that stem from the discovered (in)dependencies into a SAT instance. COmbINE outputs a summary of the structural characteristics of the underlying SMCM, distinguishing between the characteristics that are identifiable from the data (e.g., causal relations that are postulated as present), and the ones that are not (e.g., relations that could be present or not). In the empirical evaluation the algorithm outperforms in efficiency a recently published similar one (Hyttinen et al., 2013) that, given an oracle of conditional independence, performs the same inferences by checking all m -connections necessary for completeness.

COmbINE is equipped with a conflict resolution technique that ranks dependencies and independencies discovered according to confidence as a function of their p-values. This technique allows it to be applicable on real data that may present conflicting constraints due to statistical errors. To the best of our knowledge, COmbINE is the only implemented algorithm of its kind that can be applied on real data.

The algorithm is empirically evaluated in various scenarios, where it is shown to exhibit high precision and recall and reasonable behavior against sample size and number of input data sets. It scales up to networks with up to 100 variables for relatively sparse networks. Moreover, it is possible for the user to trade the number of inferences for improved computational efficiency by limiting the maximum path length considered by the algorithm. As a proof-of-concept application, we used COmbINE to analyze a real set of experimental

mass-cytometry data sets measuring overlapping variables under three different interventions.

COMBINE outputs a summary of the characteristics of the underlying SMCM that can be identified by computing the backbone of the corresponding SAT instance. The conversion of a causal discovery problem to a SAT instance makes COMBINE easily extendable to other inference tasks. One could instead produce all SAT solutions and obtain all the SMCMs that are plausible (i.e., fit all data sets). In this case, COMBINE with input a single PAG would output all SMCMs that are Markov Equivalent with the PAG; there is no other known procedure for this task. Alternatively, one could easily query whether there are solution models with certain structural characteristics of interest (e.g., a directed path from A to B); this is easily done by imposing additional SAT clauses expressing the presence of these features. Incorporating certain types of prior knowledge such as causal precedence information can also be achieved by imposing additional path constraints. Future work includes extending this work for admitting soft interventions and known instrumental variables. The conflict resolution technique proposed could be employed to standard causal discovery algorithms that learn from single data sets, in an effort to improve their learning quality.

Appendix A.

Proof of Proposition 8

Proposition

Let \mathbf{O} be a set of variables and \mathcal{J} the independence model over \mathbf{V} . Let \mathcal{S} be a SMCM over variables \mathbf{V} that is faithful to \mathcal{J} and \mathcal{M} be the MAG over the same variables that is faithful to \mathcal{J} . Let $X, Y \in \mathbf{O}$. Then there is an inducing path between X and Y with respect to \mathbf{L} , $\mathbf{L} \subseteq \mathbf{V}$ in \mathcal{S} if and only if there is an inducing path between X and Y with respect to \mathbf{L} in \mathcal{M} .

Proof (\Rightarrow) Assume there exists a path p in \mathcal{S} that is inducing w.r.t. \mathbf{L} . Then by theorem 7 there exists no $\mathbf{Z} \subseteq \mathbf{V} \setminus \mathbf{L} \cup \{X, Y\}$ such that X and Y are m -separated given \mathbf{Z} in \mathcal{S} , and since \mathcal{S} and \mathcal{M} entail the same m -separations there exists no $\mathbf{Z} \subseteq \mathbf{V} \setminus \mathbf{L} \cup \{X, Y\}$ such that X and Y are m -separated given \mathbf{Z} in \mathcal{M} . Thus, by Theorem 6 there exists an inducing path between X and Y with respect to \mathbf{L} in \mathcal{M} .

(\Leftarrow) Similarly, assume there exists a path p in \mathcal{M} that is inducing w.r.t. \mathbf{L} . Then by theorem 6 there exists no $\mathbf{Z} \subseteq \mathbf{V} \setminus \mathbf{L} \cup \{X, Y\}$ such that X and Y are m -separated given \mathbf{Z} in \mathcal{M} , and since \mathcal{S} and \mathcal{M} entail the same m -separations there exists no $\mathbf{Z} \subseteq \mathbf{V} \setminus \mathbf{L} \cup \{X, Y\}$ such that X and Y are m -separated given \mathbf{Z} in \mathcal{S} . Thus, by Theorem 7 there exists an inducing path between X and Y with respect to \mathbf{L} in \mathcal{S} . ■

Proof of Lemma 10

Lemma

Let $\{\mathcal{P}_i\}_{i=1}^N$ be a set of PAGs and \mathcal{S} a SMCM such that \mathcal{S} is possibly underlying for $\{\mathcal{P}_i\}_{i=1}^N$, and let \mathcal{H} be the initial search graph returned by Algorithm 3 for $\{\mathcal{P}_i\}_{i=1}^N$. Then, if p is an ancestral path in \mathcal{S} , then p is a possibly ancestral path in \mathcal{H} . Similarly, if p is a possibly inducing path with respect to \mathbf{L} in \mathcal{S} , then p is a possibly inducing path with respect to \mathbf{L} in \mathcal{H} .

Proof We will first prove that any path in \mathcal{S} is a path also in \mathcal{H} , i.e. \mathcal{H} has a superset of edges compared to \mathcal{S} . If X and Y are adjacent in \mathcal{S} , then one of the following holds:

1. $\exists i$ s.t. $X, Y \in \mathbf{O}_i \setminus \mathbf{I}_i$. Then the edge is present in $\mathcal{S}^{\mathbf{I}_i}$, and X and Y are adjacent in \mathcal{P}_i : the edge is added to \mathcal{H} in Lines 2 of Algorithm 3.
2. $\nexists i$ s.t. $X, Y \in \mathbf{O}_i \setminus \mathbf{I}_i$. Then the edge is added in \mathcal{H} in Line 5 of Algorithm 3.

Therefore, every edge in \mathcal{S} is present also in \mathcal{H} . We must also prove that no orientation in \mathcal{H} is oriented differently in \mathcal{S} : \mathcal{H} has only arrowhead orientations, so we must prove that, if $X \leftarrow^* Y$ in \mathcal{H} and X and Y are adjacent in both graphs, $X \leftarrow^* Y$ in \mathcal{S} .

Arrows are added to \mathcal{H} in Line 2 or in Lines 6 of the Algorithm. Arrowheads added in Line 2 occur in all \mathcal{P}_i . If $X \star \rightarrow Y$ in \mathcal{P}_i , this means that Y is not an ancestor of X in $\mathcal{S}^{\mathbf{I}_i}$. Assume that $X \leftarrow Y$ in \mathcal{S} : If X in \mathbf{I}_i , the edge would be absent in $\mathcal{S}^{\mathbf{I}_i}$ and \mathcal{P}_i . If $X \notin \mathbf{I}_i$, X would be ancestor of Y in $\mathcal{S}^{\mathbf{I}_i}$, which is a contradiction. Therefore, if X and Y are adjacent in \mathcal{S} , $X \star \rightarrow Y$ in \mathcal{S} .

The latter type of arrows correspond to cases where an edge is not present in any \mathcal{P}_i , $\nexists i$ s.t. $X, Y \in \mathbf{O}_i \setminus \mathbf{I}_i$, but $\exists i$ s.t. $X, Y \in \mathbf{O}_i$, $X \in \mathbf{I}_i$ and $Y \notin \mathbf{I}_i$. Then an arrow is added towards X . Assume the opposite holds: $X \rightarrow Y$ in \mathcal{S} , then $X \rightarrow Y$ in $\mathcal{S}^{\mathbf{I}_i}$, and since both variables are observed in i the edge would be present in \mathcal{P}_i , which is a contradiction. Thus, if the edge is present in \mathcal{S} , the edge is oriented into X .

Thus, \mathcal{H} has a superset of edges of \mathcal{S} , and for any edge present in both graphs, the orientations are the same. Thus, if p is an ancestral path in \mathcal{S} , then p is a possibly ancestral path in \mathcal{H} . Similarly, if p is a possibly inducing path with respect to \mathbf{L} in \mathcal{S} , then p is a possibly inducing path with respect to \mathbf{L} in \mathcal{H} . \blacksquare

Proof of Lemma 11

Lemma

Let $\{\mathbf{D}_i\}_{i=1}^N$ be a set of data sets over overlapping subsets of \mathbf{O} , and $\{\mathbf{I}_i\}_{i=1}^N$ be a set of (possibly empty) intervention targets such that $\mathbf{I}_i \subset \mathbf{O}_i$ for each i . Let \mathcal{P}_i be output PAG of FCI for data set \mathbf{D}_i , $\Phi \wedge \mathcal{F}'$ be the final formula of Algorithm 2, and \mathcal{S} be a possibly underlying SMCM for $\{\mathbf{P}_i\}_{i=1}^N$. Then \mathcal{S} satisfies $\Phi \wedge \mathcal{F}'$.

Proof Constraints in Lines 7 and 8 of Algorithm 4 are satisfied since \mathcal{S} is a semi-Markov causal model.

Since $\mathcal{M}_i \in \mathcal{P}_i \forall i$, \mathcal{M}_i and \mathcal{P}_i share the same adjacencies and non-adjacencies. If X and Y are adjacent in \mathcal{P}_i , X and Y are adjacent in \mathcal{M}_i , and by Proposition 8 there exists an inducing path with respect to \mathbf{L}_i in $\mathcal{S}^{\mathbf{I}_i}$, and by Lemma 10 this path is a possibly inducing path in the initial search graph. If X and Y are not adjacent in \mathcal{P}_i , X and Y are not adjacent in \mathcal{M}_i , and by Proposition 8 there exists no inducing path with respect to \mathbf{L}_i in $\mathcal{S}^{\mathbf{I}_i}$. Thus, constraints added in Line 4 of Algorithm 4 along with the corresponding literals $(\neg)adjacent(X, Y, \mathcal{P}_i)$ are satisfied by \mathcal{S} .

If $X \star \rightarrow Y \star \rightarrow Z$ is an unshielded triple in \mathcal{P}_i , $X \star \rightarrow Y \star \rightarrow Z$ is an unshielded triple in \mathcal{M}_i . If Y is a collider on the triple on \mathcal{P}_i then Y is a collider on the triple on \mathcal{M}_i and by the semantics of edges in MAGs Y is not an ancestor of X nor Z in $\mathcal{S}^{\mathbf{I}_i}$. Thus, constraints added to Φ in Line 13 along with the corresponding literal $collider(X, Y, Z, \mathcal{P}_i)$ are satisfied by \mathcal{S} . Similarly, if Y is not a collider on the triple, Y is an ancestor of either X or Z in \mathcal{M}_i

and there exists a relative ancestral path p_{YX} or p_{YZ} in $\mathcal{S}^{\mathbf{I}_i}$. By Lemma 10, this path is a possibly ancestral path in the initial \mathcal{H} . Thus, \mathcal{S} satisfies the constraints added to Φ in in Line 12 along with the corresponding literal $dnc(X, Y, Z, \mathcal{P}_i)$.

If $\langle W, \dots, X, Y, Z \rangle$ is a discriminating path for V in \mathcal{P}_i and \mathcal{M}_i and Y is a collider on the path in \mathcal{P}_i , then Y is a collider on the path in \mathcal{M}_i , therefore Y is not an ancestor of either X or Z in $\mathcal{S}^{\mathbf{I}_i}$, so \mathcal{S} satisfies the constraints added to Φ in Line 18 of Algorithm 4 along with the corresponding literal $collider(X, Y, Z, \mathcal{P}_i)$. Similarly, if Y is not a collider on the triple, Y is an ancestor of either X or Z in \mathcal{M}_i and there exists a relative ancestral path p_{YX} or p_{YZ} in $\mathcal{S}^{\mathbf{I}_i}$. By Lemma 10, this path is a possibly ancestral path in the initial \mathcal{H} . Thus, \mathcal{S} satisfies the constraints added to Φ in in Line 17 along with the corresponding literal $dnc(X, Y, Z, \mathcal{P}_i)$. ■

Proof of Lemma 12

Lemma

Let $\{\mathbf{D}_i\}_{i=1}^N$, $\{\mathbf{I}_i\}_{i=1}^N$, $\{\mathbf{P}_i\}_{i=1}^N$, $\Phi \wedge \mathcal{F}'$ be defined as in Lemma 11. If graph \mathcal{S} satisfies $\Phi \wedge \mathcal{F}'$, then \mathcal{S} is a possibly underlying SMCMM for $\{\mathbf{P}_i\}_{i=1}^N$.

Proof \mathcal{S} is a SMCMM: \mathcal{S} is by construction a mixed graph, and it satisfies constraints in Lines 7 and 8 of Algorithm 4, so it has no directed cycles, and at most one tail per edge.

\mathcal{M}_i and \mathcal{P}_i share the same edges: If X and Y are adjacent in \mathcal{P}_i , then by the constraints in Line 4 of Algorithm 4 there exists an inducing path with respect to \mathbf{L}_i in $\mathcal{S}^{\mathbf{I}_i}$, therefore X and Y are adjacent in \mathcal{M}_i . If X and Y are not adjacent in \mathcal{P}_i then by the same constraints there exists no inducing path with respect to \mathbf{L}_i in $\mathcal{S}^{\mathbf{I}_i}$, therefore X and Y are not adjacent in \mathcal{M}_i .

\mathcal{M}_i and \mathcal{P}_i share the same unshielded colliders: Let $X \star \rightarrow Y \star \rightarrow Z$ be an unshielded triple in \mathcal{P}_i . Since \mathcal{P}_i and \mathcal{M}_i share the same edges, $X \star \rightarrow Y \star \rightarrow Z$ is an unshielded triple in \mathcal{M}_i . If the triple is an unshielded collider in \mathcal{P}_i then by the constraints in Line 13 of Algorithm 4 Y is not an ancestor of either X or Z in $\mathcal{S}^{\mathbf{I}_i}$, thus $X \star \rightarrow Y \leftarrow \star Z$ in \mathcal{M}_i . If on the other hand the triple is a definite non-collider in \mathcal{P}_i , then by the constraints in Line 12 of Algorithm 4 Y is an ancestor of either X or Z in $\mathcal{S}^{\mathbf{I}_i}$, therefore either $Y \rightarrow X$ or $Y \rightarrow Z$ in \mathcal{M}_i , thus, the triple is an unshielded non-collider in \mathcal{M}_i .

If $\langle W, \dots, X, Y, Z \rangle$ is a discriminating path for V in both \mathcal{M}_i and \mathcal{P}_i , and Y is a collider on the path, then by the constraints in Line 18 of Algorithm 4 Y is not an ancestor of X or Z in $\mathcal{S}^{\mathbf{I}_i}$, therefore Y is a collider on the same path in \mathcal{M}_i . If, conversely, Y is not a collider on the path, then by the constraints in Line 17 of Algorithm 4, Y is an ancestor of either X or Z , thus, X is not a collider on the same path in \mathcal{M}_i . ■

References

- IA Beinlich, HJ Suermondt, RM Chavez, and GF Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Second European Conference on Artificial Intelligence in Medicine*, volume 38, pages 247–256. Springer-Verlag, Berlin, 1989.
- SC Bendall, EF Simonds, P Qiu, El-ad D Amir, PO Krutzik, R Finck, RV Bruggner, R Melamed, A Trejo, OI Ornatsky, RS Balderas, SK Plevritis, K Sachs, D Peér, SD Tan-

- ner, and GP Nolan. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science*, 332(6030):687–696, 2011.
- B Bodenmiller, ER Zunder, R Finck, TJ Chen, ES Savig, RV Bruggner, EF Simonds, SC Bendall, K Sachs, PO Krutzik, et al. Multiplexed mass cytometry profiling of cellular states perturbed by small-molecule regulators. *Nature biotechnology*, 30(9):858–867, 2012.
- G Borboudakis, S Triantafillou, and I Tsamardinos. Tools and algorithms for causally interpreting directed edges in maximal ancestral graphs. In *Sixth European Workshop on Probabilistic Graphical Models(PGM)*, 2012.
- T Chu, C Glymour, R Scheines, and P Spirtes. A statistical problem for inference to regulatory structure from associations of gene expression measurements with microarrays. *Bioinformatics*, 19(9):1147–1152, 2003.
- T Claassen and T Heskes. Causal discovery in multiple models from different experiments. In *Advances in Neural Information Processing Systems (NIPS 2010)*, volume 23, pages 1–9, 2010a.
- T Claassen and T Heskes. Learning causal network structure from multiple (in) dependence models. In *Proc. of the Fifth European Workshop on Probabilistic Graphical Models (PGM)*, pages 81–88, 2010b.
- T Claassen and T Heskes. A Bayesian Approach to Constraint Based Causal Inference. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI2012)*, pages 207–217, 2012.
- Diego Colombo, Marloes H. Maathuis, Markus Kalisch, and Thomas S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40(1):294–321, 02 2012.
- GF Cooper and Ch Yoo. Causal discovery from a mixture of experimental and observational data. In *Proceedings of Uncertainty in Artificial Intelligence (UAI 1999)*, volume 10, pages 116–125, 1999.
- D Eaton and K Murphy. Belief net structure learning from uncertain interventions. *J Mach Learn Res*, 1:1–48, 2007a.
- D Eaton and K Murphy. Bdagl: Bayesian dag learning. <http://www.cs.ubc.ca/~murphyk/Software/BDAGL/>, 2007b.
- F Eberhardt and R Scheines. Interventions and causal inference. *Philosophy of science*, 74(5):981–995, 2007.
- N Eén and N Sörensson. An extensible SAT-solver. In *Theory and Applications of Satisfiability Testing*, pages 333–336, 2004.
- RJ Evans and TS Richardson. Maximum likelihood fitting of acyclic directed mixed graphs to binary data. In *Proceedings of the 26th International Conference on Uncertainty in Artificial Intelligence*, 2010.

- RJ Evans and TS Richardson. Marginal log-linear parameters for graphical markov models. *arXiv preprint arXiv:1105.6075*, 2011.
- RA Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.
- Dan Geiger and David Heckerman. Learning gaussian networks. In *Proceedings of the Tenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 235–243, San Francisco, CA, 1994. Morgan Kaufmann.
- Carla P Gomes, Bart Selman, Nuno Crato, and Henry Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of automated reasoning*, 24(1-2):67–100, 2000.
- A Hauser and P Bühlmann. Characterization and Greedy Learning of Interventional Markov Equivalence Classes of Directed Acyclic Graphs. *JMLR*, 13, August 2012.
- A Hyttinen, F Eberhardt, and PO Hoyer. Learning linear cyclic causal models with latent variables. *JMLR*, 13:3387–3439, 2012a.
- A Hyttinen, F Eberhardt, and PO Hoyer. Causal discovery of linear cyclic models from multiple experimental data sets with overlapping variables. In *Uncertainty in Artificial Intelligence*, 2012b.
- A Hyttinen, PO Hoyer, F Eberhardt, and M Järvisalo. Discovering cyclic causal models with latent variables: A general sat-based procedure. In *Uncertainty in Artificial Intelligence*, 2013.
- Adrian Kuegel. Improved exact solver for the weighted max-sat problem. In *Workshop Pragmatics of SAT*, 2010.
- S Meganck, S Maes, P Leray, and B Manderick. Learning semi-markovian causal models using experiments. In *Third European Workshop on Probabilistic Graphical Models(PGM)*, 2006.
- K Murphy. Active learning of causal bayes net structure. Technical report, UC Berkeley, 2001.
- J Ramsey, P Spirtes, and J Zhang. Adjacency faithfulness and conservative causal inference. In *Proceedings of Uncertainty in Artificial Intelligence*, 2006.
- MJ Rane, PY Coxon, DW Powell, R Webster, JB Klein, W Pierce, P Ping, and KR McLeish. p38 kinase-dependent mapkapk-2 activation functions as 3-phosphoinositide-dependent kinase-2 for akt in human neutrophils. *Journal of Biological Chemistry*, 276(5):3517–3523, 2001.
- TS Richardson and P Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

- TS Richardson, JM Robins, and I Shpitser. Nested markov properties for acyclic directed mixed graphs. In *Proceedings of the Twenty Eighth Conference on Uncertainty in Artificial Intelligence*, page 13. AUAI Press, 2012.
- K Sachs, O Perez, D Pe’er, DA Lauffenburger, and GP Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.
- T Sellke, MJ Bayarri, and JO Berger. Calibration of ρ values for testing precise null hypotheses. *The American Statistician*, 55(1):62–71, 2001.
- I Shpitser, R Evans, TS Richardson, and JM Robins. Sparse nested markov models with log-linear parameters. In *Proceedings of the Twenty Ninth Conference on Uncertainty in Artificial Intelligence (UAI-13)*, pages 576–585. AUAI Press, 2013.
- ME Smoot, K Ono, J Ruscheinski, PL Wang, and T Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27(3):431–432, 2011.
- P Spirtes and TS Richardson. A polynomial time algorithm for determining DAG equivalence in the presence of latent variables and selection bias. In *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics*, pages 489–500, 1996.
- P Spirtes, C Glymour, and R Scheines. *Causation, Prediction, and Search*. The MIT Press, second edition, January 2001.
- JD Storey and R Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America*, 100(16):9440, 2003.
- J Tian and J Pearl. On the identification of causal effects. Technical Report R-290-L, UCLA Cognitive Systems Laboratory, 2003.
- RE Tillman and P Spirtes. Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, volume 15, pages 3–15, 2011.
- RE Tillman, D Danks, and C Glymour. Integrating locally learned causal structures with overlapping variables. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.
- S Tong and D Koller. Active learning for structure in bayesian networks. In *International joint conference on artificial intelligence*, pages 863–869, 2001.
- S Triantafillou, I Tsamardinos, and IG Tollis. Learning causal structure from overlapping variable sets. In *Proceedings of Artificial Intelligence and Statistics*, volume 9, 2010.
- Ioannis Tsamardinos, Sofia Triantafillou, and Vincenzo Lagani. Towards integrative causal analysis of heterogeneous data sets and studies. *The Journal of Machine Learning Research*, 98888:1097–1157, 2012.
- TS Verma and J Pearl. Equivalence and Synthesis of Causal Models. Technical Report R-150, UCLA Department of Computer Science, 2003.

- J Zhang. *Causal inference and reasoning in causally insufficient systems*. PhD thesis, PhD thesis, Carnegie Mellon University, 2006.
- J Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896, 2008a.
- J Zhang. Causal Reasoning with Ancestral Graphs. *Journal of Machine Learning Research*, 9(1):1437–1474, 2008b.