Journal of Machine Learning Research 16 (2015) 2147-2205 Submitted 10/13; Revised 10/14; Published 10/15

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

Sofia Triantafillou* STRIANT@ICS.FORTH.GR
Ioannis Tsamardinos* TSAMARD@ICS.FORTH.GR
Institute of Computer Science
Foundation for Research and Technology - Hellas (FORTH)
N. Plastira 100 Vassilika Vouton
GR-700 13 Heraklion, Crete, Greece

Editor: Christopher Meek

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.

Keywords: causality, causal discovery, graphical models, maximal ancestral graphs, semi-Markov causal models, randomized experiments, latent variables

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

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

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 (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, 1935) 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 decreasing 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 SMCMs 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 with respect to the observed independence models; 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 they 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 (2007b) 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 SMCMs by learning the Markov equivalence classes of MAGs from observational data and then designing the experiments necessary to convert it to a SMCM.

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; Richardson, 2003). Both of them are **mixed graphs**, meaning they can contain both directed ($\longrightarrow$) and bi-directed ($\longleftrightarrow$) 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 $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, a path is a sequence of distinct nodes $\langle V_0, V_1, \ldots, V_n \rangle$ 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 $X \dashrightarrow Y$ 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 an **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 Y is a collider on this path, the triple is called an **unshielded collider**.

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.

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 $X \dashrightarrow Y$ denotes that X is a *direct* cause of Y in the context of the variables included in the model. A bi-directed edge $X \dashleftarrow\dashrightarrow Y$ 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 $\mathbf{Z}$, $\mathbf{Z} \subseteq \mathbf{V} \backslash \{A, B\}$ 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 $\mathbf{Z}$. We use the notation $\mathcal{J}_m(\mathcal{G})$ to denote the set of m-separations that hold in $\mathcal{G}$.*

Let $\mathcal{G}$ be a SMCM over a set of variables $\mathbf{O}$, Π the joint probability distribution (JPD) over the same set of variables and $\mathcal{J}(\Pi)$ the independence model, defined as the set of conditional independencies that hold in Π. 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 $\mathcal{J}(\Pi)$ and vice-versa: $\mathcal{J}_m(\mathcal{G}) = \mathcal{J}(\Pi)$.*

In causal Bayesian networks, every missing edge in $\mathcal{G}$ corresponds to a conditional independence in $\mathcal{J}(\Pi)$ (resp. an m-separation in $\mathcal{G}$), meaning there exists a subset of the variables in the model that renders the two non-adjacent variables independent. Respectively, every conditional independence in $\mathcal{J}(\Pi)$ corresponds to a missing edge in the DAG $\mathcal{G}$. This is not always true for SMCMs. Figure 1 illustrates an example of a SMCM where two non-adjacent variables are not independent given any subset of observed variables.

Evans and Richardson (2010, 2011) deal with the factorization and parameterization of SMCMs for discrete variables. Based on this parameterization, 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. The skeleton of a SMCM is not uniquely identifiable by the corresponding conditional independence model on the same variables (see Figure 1 for an example). 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) (Richardson and Spirtes, 2002), are **ancestral** mixed graphs, meaning that they contain no directed or almost directed cycles, where an almost directed cycle occurs if $X \leftrightarrow Y$ and X causes Y. 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 SMCMs, ancestral graphs are also designed to represent marginals of causal Bayesian networks. Thus, under the causal Markov and faithfulness conditions for a MAG $\mathcal{M}$ and a jpd Π, 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 $\mathcal{J}(\Pi)$. Still, like in SMCMs, 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) *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 $\langle X, Y|\mathbf{Z}\rangle \in \mathcal{J}_m(\mathcal{G})$.*

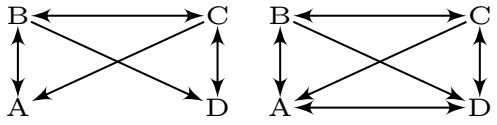

Figure 1: Maximality and primitive inducing paths.(a) Both (i) a semi Markov causal model over variables $\{A,\ B,\ C,\ D\}$; variables A and D are m-connected given any subset of observed variables, but they do not share a direct relationship in the context of observed variables and (ii) a non-maximal 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).

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 Π, then there is a unique MAG $\mathcal{G}'$ faithful to any marginal distribution of Π.

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 Π, $\mathcal{G}[_{\mathbf{L}}$ is the MAG over $\mathbf{O}$ faithful to the marginal joint probability distribution of Π. We use $\mathcal{J}[_{\mathbf{L}}$ to denote the *marginal independence model* of $\mathcal{J}$, i.e. the set of conditional independencies $\{X \perp\!\!\!\perp Y \mid \mathbf{Z} \in \mathcal{J} : (X \cup Y \cup \mathbf{Z}) \cap \mathbf{L} = \emptyset\}$. Obviously, the DAG of a causal Bayesian 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 (Marginal MAG) *(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*

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

The following theorem was proved in Richardson and Spirtes (2002):

Theorem 4 *If $\mathcal{G}$ is a MAG over $\mathbf{V} = \mathbf{O} \cup \mathbf{L}$, then $\mathcal{J}_m(\mathcal{G}[_{\mathbf{L}}) = \mathcal{J}_m(\mathcal{G})[_{\mathbf{L}}$.*

Proof See proof of Theorem 4.18 in Richardson and Spirtes (2002). ■

As mentioned above, every conditional independence in an independence model $\mathcal{J}$ 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 Π that entail a set of conditional independencies $\mathcal{J}(\Pi)$ form a **Markov equivalence class**.

It is well known that two DAGs are Markov equivalent if and only if they share the same adjacencies and unshielded colliders. Markov equivalent MAGs also share adjacencies and unshielded colliders, but this is not sufficient to characterize Markov equivalent graphs. The emergence of bi-directed edges imposes also a set of shielded colliders on the Markov equivalent MAGs. These colliders are discriminated by *discriminating paths*:

Definition 5 (Discriminating path) *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.*

Discriminating paths, their properties and their connection to Markov equivalence is discussed in detail in Ali et al. (2009). Unfortunately, two Markov equivalent MAGs may not share the same discriminating paths. Moreover, a triple may be discriminated to be a collider in MAG $\mathcal{M}_1$ but not in MAG $\mathcal{M}_2$ in the same Markov equivalence class. There exists however, a subset of discriminating paths that (a) are present in all the Markov equivalent MAGs and (b) the colliders discriminated by these paths are necessary and sufficient for Markov equivalence (Ali et al., 2009). The following definition from Ali et al. (2009) is relevant:

Definition 6 (Colliders with order) *Let $\mathfrak{D}_i, i \geq 0$ be a set of triples of order i in MAG $\mathcal{M}$, defined recursively as follows:*

- *Order 0: A triple $\langle X, Y, Z \rangle \in \mathfrak{D}_0$ if X and Z are not adjacent.*
- *Order i: A triple $\langle X, Y, Z \rangle \in \mathfrak{D}_{i+1}$ if,*
  1. *for all $j < i + 1, \langle X, Y, Z \rangle \notin \mathfrak{D}_j$ and*
  2. *There is a discriminating path $\langle W, V_1, \dots, V_n, Y, Q \rangle$ such that either $\langle X, Y, Z \rangle = \langle V_n, Y, Q \rangle$ or $\langle X, Y, Z \rangle = \langle Q, Y, V_n \rangle$ and the n colliders:*

$$\langle W, V_1, V_2 \rangle, \dots, \langle V_{n-1}, V_n, Y \rangle \in \bigcup_{j \leq i} \mathfrak{D}_j$$

*If $\langle X, Y, Z \rangle \in \mathfrak{D}_i$, the triple has order i. If the triple has order i for some i, then we say the triple has order. If $\langle X, Y, Z \rangle$ is a triple with order and $X \star\!\!\rightarrow Y \leftarrow\!\!\star Z$ is in $\mathcal{M}$, then the triple is a **collider with order** i in $\mathcal{M}$. Otherwise, the triple is a **definite non-collider with order** in $\mathcal{M}$. A discriminating path p has order i if all colliders on the path (except from the collider $\langle V_n, Y, Q \rangle$ discriminated by the path) have order at most $i - 1$, and there exists at least one collider with order $i - 1$. If a discriminating path has order i for some i, then we say that the discriminating path has order. In this work we (abusively) call (non) colliders with order ≥ 1 **discriminating (definite non) colliders**.*

Note that not every triple on a mixed graph has order. The order (if any) of a shielded triple is the minimum of the orders of all discriminating paths with order for that triple. Triples with order 0 are the unshielded triples. Discriminating paths with order ≥ 1 are

present in all Markov equivalent MAGs, and therefore colliders with order ≥ 1 are the triples that are colliders in all the Markov equivalent MAGs. Colliders with order, along with adjacencies, are necessary and sufficient to characterize Markov equivalent MAGs:

Theorem 7 *Two MAGs over the same variable set are Markov equivalent if and only if they share the same edges and the same colliders with order.*

Proof See proof of Theorem 3.7 in Ali et al. (2009). ■

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 wild card to denote any of the three marks. We use the notation $\mathcal{M} \in \mathcal{P}$ to denote that MAG $\mathcal{M}$ belongs to the Markov equivalence class represented by PAG $\mathcal{P}$.

For a MAG $\mathcal{M}$ and a probability distribution Π faithful to each other, $\mathcal{J}_m(\mathcal{M}) = \mathcal{J}(\Pi)$. Thus, the set of m-separations entailed in $\mathcal{M}$ are exactly the conditional independencies that hold in Π. **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 Π over variables $\mathbf{O}$ in which a set of conditional independencies $\mathcal{J}(\Pi)$ 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. They both entail the conditional independence structure and the causal ancestry structure of the observed variables. Thus, under CMC and FC, the SMCM $\mathcal{G}$ and the MAG $\mathcal{M}$ over a set of variables $\mathbf{O}$ entail the same independence model: $\mathcal{J}_m(\mathcal{S}) = \mathcal{J}_m(\mathcal{M})$. They also entail the same ancestral relationships: X is an ancestor of Y in $\mathcal{S}$ if and only if X is an ancestor of Y in $\mathcal{M}$.

Nevertheless, SMCMs and MAGs also have significant differences: 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.

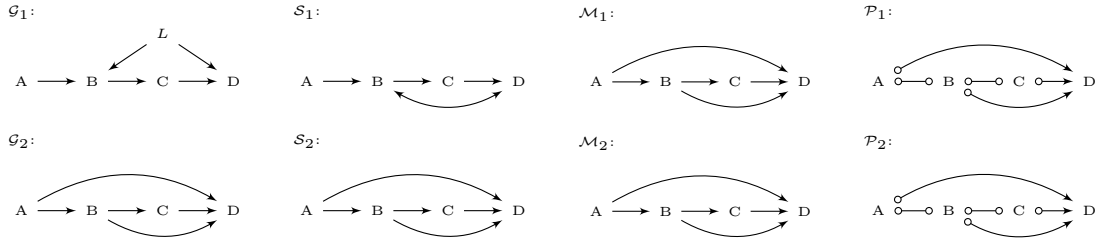

Figure 2: An example two different DAGs and the corresponding mixed causal graphs over observed variables. On the left we can see DAGs $\mathcal{G}_1$ over variables $\{A,\ B,\ C,\ D,\ L\}$ (top) and $\mathcal{G}_2$ over variables $\{A,\ B,\ C,\ D\}$ (bottom). From left to right, on the same row as the 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.

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 $X \dashrightarrow Y$ can be present in a MAG even though X does not directly cause 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 $A \dashrightarrow D$ 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 $B \dashleftarrow\dashrightarrow D$. However, a bi-directed edge between B and D is not allowed in MAG $\mathcal{M}_1$, since it would create an almost directed cycle. Thus, $B \dashrightarrow D$ 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 ($X \rightleftarrows Y$), while the two will share a directed edge from X to Y in the corresponding MAG.

Overall, a SMCM has a subset of the adjacencies (but not necessarily edges) of its MAG counterpart. These extra adjacencies in MAGs 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).

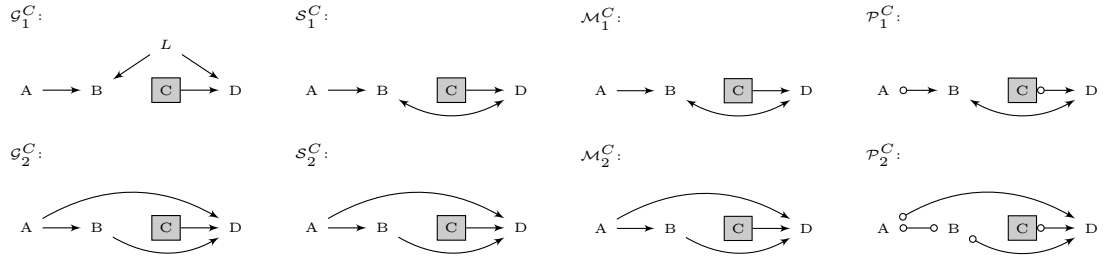

Figure 3: Effect of manipulating variable C on the causal graphs of Figure 2. From right to left we can see the manipulated DAGs $\mathcal{G}_1^C$ (top) and $\mathcal{G}_2^C$ (bottom), the manipulated SMCMs $\mathcal{S}_1^C$ (top) and $\mathcal{S}_2^C$ (bottom) over variables $\{A,\ B,\ C,\ D\}$, the 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 $A \longrightarrow D$ is removed in $\mathcal{M}_1^C$, even though it is not adjacent to the manipulated variable. Moreover, on the same graph, edge $B \longrightarrow D$ is now $B \longleftrightarrow D$.

Definition 8 (Inducing path) *A path $p = \langle V_1, V_2, \ldots, V_n \rangle$ 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 $A \longrightarrow B \longleftarrow L \longrightarrow D$ is an inducing path with respect to L in $\mathcal{G}_1$ of Figure 2, and $A \longrightarrow B \longleftrightarrow D$ 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 9 *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 9:

Theorem 10 *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* $\mathcal{G}$.

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

The following proposition follows from Theorems 9 and 10:

Proposition 12 . *Let* $\mathbf{O}$ *be a set of variables and* $\mathcal{J}$ *the independence model over* $\mathbf{O}$. *Let* $\mathcal{S}$ *be a SMCM over variables* $\mathbf{O}$ *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* $\mathbf{L}$, $\mathbf{L} \subseteq \mathbf{O}$ *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 A. ■

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 $\mathcal{M}_1$. 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.

Algorithm 1 takes as input a SMCM $\mathcal{S}$ and adds the necessary edges to transform it into a MAG $\mathcal{M}$ by looking for primitive inducing paths. The procedure can be viewed as a special case of marginalizing out variables in DAGs, presented in Spirtes and Richardson (1996) and Zhang (2008b). Similar algorithms are also presented in Sadeghi (2012), where the relationship among different types of mixed causal graphs representing the same independence model is discussed in detail. The algorithm is sound, i.e. the output MAG shares the same causal ancestry relations and entails the same independence model as $\mathcal{S}$:

Theorem 13 . *Let* $\mathbf{O}$ *be a set of variables and* $\mathcal{J}$ *the independence model over* $\mathbf{O}$. *Let* $\mathcal{S}$ *be a SMCM over variables* $\mathbf{V}$ *that is faithful to* $\mathcal{J}$. *Let* $\mathcal{M}$ = *SMCMtoMAG(*$\mathcal{S}$*). Then* $\mathcal{S}$ *and* $\mathcal{M}$ *share the same ancestry relations and* $\mathcal{J}_m(\mathcal{S}) = \mathcal{J}_m(\mathcal{M})$.

Proof See Appendix A. ■

Algorithm 1: SMCMtoMAG

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

$\mathcal{M} \leftarrow \mathcal{S}$;
foreach *ordered pair of variables* X, Y *not adjacent in* $\mathcal{S}$ **do**
  if $\exists$ *primitive inducing path from* X *to* Y *in* $\mathcal{S}$ **then**
    if $X \in \mathbf{An}_{\mathcal{S}}(Y)$ **then**
      add $X \dashrightarrow Y$ to $\mathcal{M}$;
    else if $Y \in \mathbf{An}_{\mathcal{S}}(X)$ **then**
      add $Y \dashrightarrow X$ to $\mathcal{M}$;
    else
      add $Y \dashleftarrow\dashrightarrow X$ to $\mathcal{M}$;
    end
  end
end
foreach $X \overleftrightarrows Y$ *in* $\mathcal{M}$ **do**
  remove $X \dashleftarrow\dashrightarrow Y$;
end

The algorithm is also complete, since there only exists one such MAG. 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, in the next section we 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**.

Naturally, DAGs are closed under manipulation. We use the term **intervention target** to denote a set of manipulated variables. For a DAG $\mathcal{G}$ and an intervention target $\mathbf{I}$, we use $\mathcal{G}^{\mathbf{I}}$ to denote the manipulated DAG. 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. We use $\Pi^{\mathbf{I}}$ to denote this **manipulated joint probability distribution**, and $\mathcal{J}^{\mathbf{I}}$ to denote the corresponding **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 $\mathbf{I}$. In contrast, 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 marginals of 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.

We must point out here that we use $\mathcal{P}^{\mathbf{I}}$ as the representative of the Markov equivalence class of models that are faithful to the manipulated conditional independence model $\mathcal{J}(\Pi^{\mathbf{I}})$, as opposed to the representative of the *interventional Markov equivalence class* of manipulated MAGs. The information on manipulations, not included in the present use of $\mathcal{P}^{\mathbf{I}}$, defines a smaller Markov equivalence class: For example, in Figure 3, MAGs in the interventional Markov equivalence class of $\mathcal{M}_1^C$ share the additional invariant characteristic of a tail into C on the edge $C \circ\!\!\rightarrow D$. This invariant feature however is not oriented in $\mathcal{P}_1^C$. To the best of our knowledge, no sound and complete algorithm for identifying the maximally informative PAG for the *interventional Markov equivalence class of* $\mathcal{M}^{\mathbf{I}}$ exists (however, orienting all edges out of the manipulated variables is a trivially sound method).

By observing PAGs $\{\mathcal{P}^{\mathbf{I}_i}\}$ that stem from known, different manipulations of the same underlying distribution, we can infer some 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 $A \circ\!\!-\!\!\circ D$ 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 experiment i is also provided and is denoted with $\mathbf{I}_i$.

In the rest of this paper, we make the following assumptions:

A1 We assume that there exists an underlying causal mechanism over a set of variables $\mathbf{O}$ that can be described with a semi Markov causal model $\mathcal{G}$ over $\mathbf{O}$. If Π is the joint probability distribution over $\mathbf{O}$, we assume that Π and $\mathcal{G}$ are faithful to each other, i.e. $\mathcal{J}_m(\mathcal{G}) = \mathcal{J}(\Pi)$. We also say that $\mathcal{G}$ is faithful to the independence model $\mathcal{J}(\Pi)$.

A2 We assume that we collect data sets in N different experiments, where in experiment i we observe variables $\mathbf{O}_i \subseteq \mathbf{O}$, while variables $\mathbf{L}_i = \mathbf{O} \setminus \mathbf{O}_i$ are latent and variables $\mathbf{I}_i \subset \mathbf{O}$ are manipulated. We also assume $\mathbf{O} = \bigcup_{i=1}^N \mathbf{O}_i$. We assume that manipulations are ideal hard interventions and that they result in removal of all edges in $\mathcal{G}$ that are incoming to the manipulated variables.

A3 We assume faithfulness for the manipulated SMCMs and distributions, i.e. $\mathcal{J}_m(\mathcal{G}^{\mathbf{I}_i}) = \mathcal{J}(\Pi^{\mathbf{I}_i})$.

Unless mentioned otherwise, the following notation is used:

- $\mathbf{O}_i$ denotes the set of observed variables in experiment i.
- $\mathbf{I}_i$ denotes the set of manipulated variables in experiment i.
- $\mathbf{O} = \cup_i \mathbf{O}_i$ denotes the union of observed variables.
- $\mathbf{L}_i = \mathbf{O} \setminus \mathbf{O}_i$ denotes the set of latent variables (with respect to the union of observed variables) in experiment i.
- $\mathbf{D}_i$ denotes a data set for experiment i, sampled from the mechanism described by $(\mathcal{G}^{\mathbf{I}_i}, \Pi^{\mathbf{I}_i})$, measuring variables in $\mathbf{O}_i$.
- $\mathcal{J}_i$ denotes the independence model that holds in data set $\mathbf{D}_i$. In the sample limit, $\mathcal{J}_i$ is equal to the set of m-separations that hold for sets of variables in $\mathbf{O}_i$ after manipulating $\mathbf{I}_i$ in the underlying causal model: $\mathcal{J}_i = \mathcal{J}(\Pi^{\mathbf{I_i}})[_{\mathbf{L_i}} = \mathcal{J}_m(\mathcal{G}^{\mathbf{I_i}})[_{\mathbf{L_i}}$.
- $\mathcal{P}_i$ denotes the maximally informative PAG for the (observational) Markov equivalence class of MAGs faithful to $\mathcal{J}_i$. Thus, for any MAG $\mathcal{M}_i \in \mathcal{P}_i$, $\mathcal{J}_m(\mathcal{M}_i) = \mathcal{J}_i$. Notice that, since SMCMs and MAGs over the same variables represent the same independence model, for an oracle of conditional independence, $\mathcal{P}_i = [\text{SMCMtoMAG}(\mathcal{G}^{\mathbf{I}_i})[_{\mathbf{L}_i}]$.

Under the assumptions described above, we are interested in combining information across data sets collected from different manipulations and marginalizations of the same system under study, to identify features of the possible underlying causal mechanism. If $\mathcal{S}$ is a SMCM that describes this underlying causal mechanism, then this SMCM must agree with all the observed independence models $\{\mathcal{J}_i\}_{i=1}^N$. This means that for each experiment, the respective manipulated $\mathcal{S}^{\mathbf{I}_i}$ must entail all and only the conditional independencies that hold in data set $\mathbf{D}_i$ (in the sample limit $\mathcal{J}_i$ can be obtained correctly from the data). For the family of independence models $\{\mathcal{J}_i\}_{i=1}^N$, and a family of intervention targets $\{\mathbf{I}_i\}_{i=1}^N$ a **possibly underlying SMCM** is defined as follows:

Definition 11 (Possibly underlying SMCM) *If $\{\mathcal{J}_i\}_{i=1}^N$ is a family of independence models over variable sets $\{\mathbf{O}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ is a family of intervention targets such that $\mathbf{I}_i \subseteq \mathbf{O}_i \quad \forall i$, then a SMCM $\mathcal{S}$ is a **possibly underlying SMCM** for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ iff:*

$$\forall X, Y, \mathbf{Z} \subseteq \mathbf{O}_i,\ [X \text{ is } m\text{-separated from } Y \text{ given } \mathbf{Z} \text{ in } \mathcal{S}^{\mathbf{I}_i}] \Leftrightarrow X \perp\!\!\!\perp Y \mid \mathbf{Z} \in \mathcal{J}_i,$$

Intuitively, $\mathcal{S}$ is a SMCM such that once the effects of manipulations are modeled (i.e. $\mathcal{S}^{\mathbf{I}_i}$ is constructed), it entails all and only the independencies $\mathcal{J}_i$ observed in the corresponding data set. Thus, $\mathcal{S}$ is a possible causal model that explains all data. Since each independence model $\mathcal{J}_i$ can be graphically represented by a PAG $\mathcal{P}_i$, one can recast this definition in graph-theoretic terms: $\mathcal{S}$ is a possibly underlying SMCM if, after graph surgery, results in a marginal MAG that belongs in $\mathcal{P}_i$:

Theorem 14 *If $\mathcal{S}$ is a SMCM, $\{\mathcal{J}_i\}_{i=1}^N$ is a family of independence models, $\{\mathbf{I}_i\}_{i=1}^N$ is a family of intervention targets and $\mathcal{P}_i$ is the PAG of the Markov equivalence class of MAGs faithful to $\mathcal{J}_i$, the following statements are equivalent:*

- *$\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*
- *$\mathcal{M}_i \in \mathcal{P}_i \quad \forall i$, where $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i}$.*

Proof See Appendix A. ■

As mentioned above, PAGs $\mathcal{P}_i$ here denote the maximally informative representatives of the Markov equivalence class of MAGs that entail independence models $\mathcal{J}_i$, instead of the *interventional* Markov equivalence class of MAGs that entail both $\mathcal{J}_i$ and the interventional constraints following the manipulation of targets $\mathbf{I_i}$. Hence, this graphical criterion may seem incomplete, since the actual MAGs belong to thinner equivalence classes, which include some additional orientations: tails towards any manipulated variable and additional orientations stemming from the combination of m-separation and acyclicity with these aforementioned tails. However, MAGs $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i}$ are constructed after graph surgery has been applied to the (candidate) possibly underlying SMCM and abide by definition the constraints that correspond to interventional information (i.e. tail orientations towards manipulated variables), since $\mathcal{S}^{\mathbf{I}_i}$ and $\text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})$ share the same ancestral relations. Thus, the resulting MAGs $\mathcal{M}_i$ belong (by construction) to the thinner *interventional* Markov equivalence class of MAGs, and testing Markov equivalence

in the observational sense is a sound and complete graphical criterion to determine whether a SMCM is possibly underlying for a family of independence models coupled with a family of intervention targets.

Notice that PAG $\mathcal{P}_i$ can be learnt with a sound and complete algorithm such as FCI. We can now benefit by the compact representation of Markov equivalence classes of MAGs described in Theorem 7, to check whether a SMCM $\mathcal{S}$ is possibly underlying for a family of independence models $\{\mathcal{J}_i\}_{i=1}^N$ and a family of intervention targets $\{\mathbf{I}_i\}_{i=1}^N$: Instead of checking *all* conditional dependencies (resp. independencies) in $\mathcal{J}_i$ to be m-connections (resp. m-separations) in the corresponding SMCM $\mathcal{S}^{\mathbf{I}_i}$, we can construct the corresponding MAGs $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i}$ for each experiment and check whether they belong to the Markov equivalence class represented by $\mathcal{P}_i$. By Theorem 7, we only need to check adjacencies and colliders with order.

In the next section, we present an algorithm that converts the problem of identifying a SMCM $\mathcal{S}$ that is possibly underlying for a family of observed independence models $\{\mathcal{J}_i\}_{i=1}^N$ and a family of intervention targets $\{\mathbf{I}_i\}_{i=1}^N$ into a constraint satisfaction problem. Specifically, we will create a satisfiability instance s.t. a SMCM is possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ if and only if it corresponds to a truth-setting assignment for the SAT instance. For a family of independence models $\{\mathcal{J}_i\}_{i=1}^N$ and a family of intervention targets $\{\mathbf{I}_i\}_{i=1}^N$, several SMCMs may be possibly underlying. We can then use the equivalent SAT instance to query properties shared by all possibly underlying SMCMs, or to identify a single possibly underlying SMCM with some desirable characteristics. In this work, we use the equivalent SAT instance to identify all edges and endpoints that are invariant in all possibly underlying SMCMs.

4.1 Conversion to SAT

Theorem 14 implies that $\mathcal{M}_i$ has the same edges (adjacencies), and the same colliders with order (unshielded colliders and discriminating colliders with order) as any MAG in $\mathcal{P}_i$, 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 orientations in any possibly underlying SMCM $\mathcal{S}$.

- edge(X, Y): true if X and Y are adjacent in $\mathcal{S}$, false otherwise.
- tail(X, Y): true if there exists an edge between X and Y in $\mathcal{S}$ that is out of Y, false otherwise.
- 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 $X \leftrightarrows Y$ edges when $tail(Y, X) \wedge arrow(Y, X)$. 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 9 and 10 and Proposition 12).

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

/* X and Y are **adjacent** in $\mathcal{P}_i$ iff
there exists an inducing path from X to Y with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$. */

$$\textbf{collider}(\langle X, Y, Z\rangle, \mathcal{P}_i) \leftrightarrow \neg ancestor(Y, X, i) \wedge \neg ancestor(Y, Z, i)$$

/* Triple $\langle X, Y, Z\rangle$ is **collider** in $\mathcal{P}_i$ iff
Y is not an ancestor of X or Z in $\mathcal{S}^{\mathbf{I}_i}$. */

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

/* Triple $\langle X, Y, Z\rangle$ is **unshielded** in $\mathcal{P}_i$ iff
(X, Y), (Y, Z) are adjacent in $\mathcal{P}_i$, (X, Z) are not adjacent in $\mathcal{P}_i$. */

$$\begin{aligned}\textbf{discriminating}(\langle V_0, \ldots, V_{n-1}, V_n, V_{n+1}\rangle, V_n, \mathcal{P}_i) \leftrightarrow \\ \neg adjacent(V_0, V_{n+1}, \mathcal{P}_i) \wedge \forall j \in [0, \ldots, n] adjacent(V_j, V_{j+1}, \mathcal{P}_i) \wedge \\ \forall j \in [1, \ldots, n-1]\big(collider(\langle V_{j-1}, V_j, V_{j+1}\rangle, \mathcal{P}_i) \\ \wedge adjacent(V_j, V_{n+1}, \mathcal{P}_i) \wedge ancestral(V_j, V_{n+1}, i)\big)\end{aligned}$$

/* Path $\langle V_0, \ldots, V_{n+1}\rangle$ is **discriminating** for V_n in $\mathcal{P}_i$ iff
V_0, V_{n+1} are not adjacent in $\mathcal{P}_i$, $V_0, \ldots, V_{n+1}$ is a path in $\mathcal{P}_i$,
every node between V_0 and V_n is a collider on the path
and a parent of V_{n+1} in $\mathcal{P}_i$. */

Figure 4: Formulae relating properties of observed PAGs to the underlying SMCM $\mathcal{S}$. In each PAG, all features that are necessary and sufficient for Markov equivalence impose constraints on possibly underlying SMCMs. Constraints are expressed using the literals and formulae introduced here. Index i is used to denote properties of an underlying SMCM in experiment i, where variables $\mathbf{L}_i$ are latent and variables $\mathbf{I}_i$ are manipulated. We use use p_{XY} to denote a path between X and Y in $\mathcal{S}$. Conjunction and disjunction are assumed to have precedence over implication with regard to bracketing. Each formula is followed by an explanation in natural language (in star-slash comments).

$$\textbf{inducing}(\langle V_0,\ldots,V_{n+1}\rangle, i) \leftrightarrow$$
$$(n=0 \rightarrow edge(V_0, V_{n+1})) \wedge$$
$$\big(n>0 \rightarrow (\forall j \in [1,\ldots,n]\ unblocked(\langle V_{j-1}, V_j, V_{j+1}\rangle, V_0, V_{n+1}, i))\big) \wedge$$
$$(V_0 \in \mathbf{I}_i \rightarrow tail(V_1, V_0)) \wedge (Y \in \mathbf{I}_i \rightarrow tail(V_n, V_{n+1}))$$

/* Path $\langle V_0,\ldots,V_{n+1}\rangle$ is **inducing** with respect to $\mathbf{L_i}$ in $\mathcal{S}^{\mathbf{I}_i}$ iff
if the path has only two variables, V_0 is adjacent to V_n in $\mathcal{S}$
else each triple is **unblocked** for the endpoints with respect to $\mathbf{L_i}$,
if V_0 (V_{n+1}) is manipulated in i then the path is out of V_0 (V_{n+1}) in $\mathcal{S}$. */

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

/* Triple $\langle Z, V, W\rangle$ is **unblocked** for X, Y with respect to $\mathbf{L_i}$ iff
(Z, V) (V, W) are adjacent in $\mathcal{S}$
if V is latent, if V is **head2head** then it is an ancestor of X or Y in $\mathcal{S}^{\mathbf{I}_i}$
if V is not latent, V is a **head2head** and an ancestor of X or Y in $\mathcal{S}^{\mathbf{I}_i}$. */

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

/* Triple $\langle X, Y, Z\rangle$ is **head2head** in $\mathcal{S}^{\mathbf{I}_i}$ iff
Y is not manipulated in experiment i, X is into Y, Z is into Y in $\mathcal{S}$. */

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

/* X is an **ancestor** of Y in experiment i iff
there exists an ancestral path from X to Y in $\mathcal{S}^{\mathbf{I}_i}$. */

$$\textbf{ancestral}(\langle V_0,\ldots,V_{n+1}\rangle, i) \leftrightarrow$$
$$\forall j \in [1,\ldots,n+1]\big(V_j \notin \mathbf{I}_i \wedge (edge(V_{j-1}, V_j) \wedge tail(V_j, V_{j-1}) \wedge arrow(V_{j-1}, V_j))\big)$$

/* Path $\langle V_0,\ldots,V_{n+1}\rangle$ is **ancestral** in experiment i iff
every variable (apart from possibly V_0) is not manipulated in $\mathcal{S}^{\mathbf{I}_i}$
every variable is a parent of the next in $\mathcal{S}$. */

Figure 5: Formulae reducing path properties of the graphs $\mathcal{S}^{\mathbf{I}_i}$ to the core variables: Graph properties of $\mathcal{S}$ in each experiment, inferred by the observed PAGs using the formulae in Figure 4, are now expressed as boolean formulae using the "core" variables *edge*, *arrow* and *tail*. Index i is used to denote properties of an underlying SMCM in experiment i, where variables $\mathbf{L}_i$ are latent and variables $\mathbf{I}_i$ are manipulated. Conjunction and disjunction are assumed to have precedence over implication with regard to bracketing. Each formula is followed by an explanation in in natural language (in star-slash comments).

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 $\langle W, \ldots, X, Y, Z\rangle$ is a discriminating collider in $\mathcal{P}_i$, then $\langle W \ldots, X, Y, Z\rangle$ 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).

5. If $\langle W, \ldots, X, Y, Z\rangle$ is a discriminating definite non collider in $\mathcal{P}_i$, then $\langle W \ldots, X, Y, Z\rangle$ 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).

These constraints are expressed using the core variables (edges, tails and arrows), as described in Figures 4 and 5. Figure 4 describes how features of a PAG are imposed as path constraints in a possibly underlying SMCM. More specifically, an adjacency, a tail and an arrowhead in a PAG $\mathcal{P}_i$ correspond to an inducing path, a causal ancestry and the lack of causal ancestry on any possibly underlying SMCM, respectively. Unshielded triples and discriminating paths are expressed on the basis of these basic PAG features. In each PAG, the observed features depend on the latent and manipulated variables. When constraints are imposed on the candidate underlying SMCMs, the latent and manipulated variables in the experiment are taken under consideration: If an adjacency is observed in $\mathcal{P}_i$ in experiment i, where variables $\mathbf{L}_i$ are latent and $\mathbf{I}_i$ are manipulated, then any path on $\mathcal{S}$ that explains this adjacency must be inducing with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$. 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 $tail(X, Y)$ and $tail(Y, X)$ can be true (no selection bias).

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 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, heterogeneous 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. Each data set entails an independence model $\mathcal{J}_i$. FCI is run on each data set and the corresponding PAGs $\{\mathcal{P}_i\}_{i=1}^N$ are produced. The algorithm then creates a candidate underlying SMCM $\mathcal{H}_{in}$. Subsequently, for each PAG $\mathcal{P}_i$, the features of $\mathcal{P}_i$ are translated into

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** $\mathcal{P}_i \leftarrow$ `FCI`$(\mathbf{D}_i, \textit{params})$;
2 $\mathcal{H}_{in} \leftarrow$ `initializeSMCM` $(\{\mathcal{P}_i\}_{i=1}^N)$;
3 $(\Phi, \mathcal{F}) \leftarrow$ `addConstraints` $(\mathcal{H}, \{\mathcal{P}_i\}_{i=1}^N, \{\mathbf{I}_i\}_{i=1}^N, \textit{mpl})$;
4 $\mathcal{F}' \leftarrow$ select a subset of non-conflicting literals $\mathcal{F}'$ according to strategy *str*;
5 $\mathcal{H} \leftarrow$ `backBone` $(\Phi \wedge \mathcal{F}')$

constraints, expressed in terms of edges and endpoints in $\mathcal{H}_{in}$, using the formulae in Figures 4 and 5 . 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 possibly underlying SMCMs for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}$. 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 $\mathcal{F}'$ 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 possibly 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 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 $\mathcal{H}_{in}$ is initialized: $\mathcal{H}_{in}$ is the graph upon which all path constraints will be imposed. Path constraints are realized on the basis of the *plausible configurations* of $\mathcal{H}_{in}$. We say that a path p in $\mathcal{H}_{in}$ is **possibly inducing with respect to L**, if we can create a graph $\mathcal{H}'_{in}$ by orienting circle endpoints in $\mathcal{H}_{in}$ such that path p is inducing with respect to $\mathbf{L}$ in $\mathcal{H}'_{in}$. We say that a path p in $\mathcal{H}_{in}$ is **possibly ancestral**, if we can create a graph $\mathcal{H}'_{in}$ by orienting circle endpoints in $\mathcal{H}_{in}$ such that path p is ancestral $\mathcal{H}'_{in}$. To ensure the soundness of the algorithm, if p is an inducing (ancestral) path in $\mathcal{S}$, it must be a possibly inducing (ancestral) path in $\mathcal{H}_{in}$. Thus, $\mathcal{H}_{in}$ must have at least a superset of edges and at most a subset of orientations of any possibly underlying SMCM $\mathcal{S}$.

An obvious–yet not very smart–choice for $\mathcal{H}_{in}$ would be the complete unoriented graph. However, looking for possibly inducing and possibly ancestral paths on the complete unoriented graph over the union of variables could make the problem intractable even for small input sizes. To reduce the number of possibly inducing and possibly ancestral paths, we use Algorithm 3 to construct $\mathcal{H}_{in}$.

Algorithm 3 constructs a graph $\mathcal{H}_{in}$ 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

Algorithm 3: initializeSMCM

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

1 $\mathcal{H}_{in} \leftarrow$ empty graph over $\cup \mathbf{O}_i$;
2 **foreach** i **do**
3 | $\mathcal{H}_{in} \leftarrow$ add all edges in $\mathcal{P}_i$ unoriented;
4 **end**
5 Orient only arrowheads that are present in every $\mathcal{P}_i$;
/* Add edges between variables never measured unmanipulated together */
6 **foreach** *pair* X, Y *of non-adjacent nodes* **do**
7 | **if** $\nexists i$ *s.t.* $X, Y \in \mathbf{O}_i \setminus \mathbf{I}_i$ **then**
8 | | add $X \circ\!\!-\!\!\circ Y$ to $\mathcal{H}_{in}$;
9 | | **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;
10 | | **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;
11 | **end**
12 **end**

appearance in a data set. For example, variables $X9$ and $X15$ in Figure 6 are measured together in two data sets: $\mathbf{D}_2$ and $\mathbf{D}_3$. If $X9 \dashrightarrow X15$ in the underlying SMCM, this edge would be present in $\mathcal{P}_3$. Similarly, if $X15 \dashrightarrow X9$ 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 $X15$ and $X9$ 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 $\mathcal{H}_{in}$. In addition, in Line 5, Algorithm 3 adds all the orientations found so far in all $\mathcal{P}_i$'s that are invariant.[1] The resulting graph has, in the sample limit, a superset of edges and a subset of orientations compared to the actual underlying SMCM. Lemma 15 in Appendix A 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 14, 13, 20 and 19), (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 8 and 9 respectively). Literal *col* (resp. *dnc*) is used to represent both unshielded and discriminating colliders (resp. unshielded and discriminating non colliders).

The constraints are realized on the basis of the *plausible* configurations of $\mathcal{H}_{in}$: Thus, for the constraints corresponding to $adjacent(X, Y, i)$ the algorithm finds all paths between

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

$\Phi \leftarrow \emptyset$ **foreach** X, Y **do**
  foreach i **do**
    posIndPaths$\leftarrow$ paths in $\mathcal{H}_{in}$ of maximum length *mpl* that are possibly inducing with respect to $\mathbf{L}_i$;
    $\Phi \leftarrow \Phi \wedge \big[adjacent(X, Y, \mathcal{P}_i) \leftrightarrow \exists p_{XY} \in$**posIndPaths** s. t. $inducing(p_{XY}, i)\big]$;
    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 $\mathcal{F}$;
  end
  $\Phi \leftarrow \Phi \wedge \big[\neg ancestor(X, Y) \vee \neg ancestor(Y, X)\big]$;
  $\Phi \leftarrow \Phi \wedge \big[\neg tail(X, Y) \vee \neg tail(Y, X)\big] \wedge \big[(arrow(X, Y) \vee tail(X, Y)\big]$;
end
foreach i **do**
  foreach *unshielded triple in* $\mathcal{P}_i$ **do**
    $\Phi \leftarrow \Phi \wedge \big[col(X, Y, Z, \mathcal{P}_i) \rightarrow unshielded(X, Y, Z, \mathcal{P}_i) \wedge collider(X, Y, Z, \mathcal{P}_i)\big]$;
    $\Phi \leftarrow \Phi \wedge \big[dnc(X, Y, Z, \mathcal{P}_i) \rightarrow unshielded(X, Y, Z, \mathcal{P}_i) \wedge \neg collider(X, Y, Z, \mathcal{P}_i)\big]$;
    if $\langle X, Y, Z\rangle$ *is a collider in* $\mathcal{P}_i$ **then** add $col(X, Y, Z, \mathcal{P}_i)$ to $\mathcal{F}$;
    else add $dnc(X, Y, Z, \mathcal{P}_i)$ to $\mathcal{F}$;
  end
  foreach *discriminating path* $p_{WZ} = \langle W, \ldots, X, Y, Z\rangle$ **do**
    $\Phi \leftarrow \Phi \wedge \big[col(X, Y, Z, \mathcal{P}_i) \rightarrow discriminating(p_{WZ}, Y, \mathcal{P}_i) \wedge collider(X, Y, Z, \mathcal{P}_i)\big]$;
    $\Phi \leftarrow \Phi \wedge \big[dnc(X, Y, Z, \mathcal{P}_i) \rightarrow discriminating(p_{WZ}, Y, \mathcal{P}_i) \wedge \neg collider(X, Y, Z, \mathcal{P}_i)\big]$;
    if *X, Y, Z is a collider in* $\mathcal{P}_i$ **then** add $col(X, Y, Z, \mathcal{P}_i)$ to $\mathcal{F}$;
    else add $dnc(X, Y, Z, \mathcal{P}_i)$ to $\mathcal{F}$;
  end
end

X and Y in $\mathcal{H}_{in}$ 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.

As an example, consider the following variation of the instance presented in Figure 7. Assume that variable X is manipulated in experiment 1, and no variable is manipulated in experiment 2. Since no information concerning experiments is employed up to the initialization of the search graph, the resulting PAGs are the $\mathcal{P}_1$ and $\mathcal{P}_2$ shown in Figure 7. Thus, in the initial search graph $\mathcal{H}_{in}$, $X \circ\!\!-\!\!\circ Y$ and $X \circ\!\!-\!\!\circ Z \circ\!\!-\!\!\circ Y$ are the two possibly inducing paths for X and Y in experiment i. Then the following constraint will be imposed:

$$adjacent(X, Y, 1) \leftrightarrow inducing(\langle X, Y\rangle, 1) \vee inducing(\langle X, Z, Y\rangle, 1)$$

For path $\langle X, Y\rangle$, the corresponding constraint is reduced to the properties of $\mathcal{S}$ as follows:

$$inducing(\langle X, Y\rangle, 1) \leftrightarrow \\ (X \in \mathbf{I}_1 \rightarrow tail(Y, X)) \wedge (Y \in \mathbf{I}_1 \rightarrow tail(X, Y)) \wedge edge(X, Y)$$

which is then added in Φ as $inducing(\langle X, Y\rangle, 1) \leftrightarrow tail(Y, X) \wedge edge(X, Y)$ since $X \in \mathbf{I}_1$ is true and $Y \in \mathbf{I}_1$ is false. For the path $\langle X, Z, Y\rangle$ the corresponding constraint finally added in Φ is

$$inducing(\langle X, Z, Y\rangle) \leftrightarrow \\ tail(Z, X) \wedge [\neg head2head(\langle X, Z, Y\rangle) \vee ancestral(Z, X) \vee ancestral(Z, Y)]$$

Thus, in a SMCM that satisfies the final formula, if $inducing(\langle X, Y\rangle, i)$ is true, there will be an inducing path from X to Y consistent with the manipulation information.

Also notice how this constraint is *propagated* in the SAT: For example, $X \star\!\!-\!\!\star Z \star\!\!-\!\!\star Y \star\!\!-\!\!\star W$ is a plausible skeleton for a possibly underlying SMCM. By the constraints mentioned above, $X \rightarrow Z \star\!\!-\!\!\star Y$ is the inducing path for X and Y with respect to $L_1 = Z$. By the constraints added for the definite non collider $\langle X, Z, W\rangle$ for $\mathcal{P}_2$, Z has to be an ancestor of either X or Y in $\mathcal{S}^{\emptyset}$. Therefore, the path $Z \star\!\!-\!\!\star Y \star\!\!-\!\!\star W$ has to be an ancestral path in $\mathcal{S}$, which implies that $Y \dashrightarrow Z$ in $\mathcal{S}$. Thus, the orientation $Y \dashrightarrow Z$ is imposed by a combination of constraints stemming from different PAGs, for two variables never jointly measured.

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 uses 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 $\exists p_{XY} : inducing(X, Y, i)$ to Φ, we add the bi-conditional $adjacent(X, Y, \mathcal{P}_i) \leftrightarrow \exists p_{XY} : inducing(X, Y, i)$. 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 $\mathcal{F}'$ that results in a satisfiable SAT formula $\Phi \wedge \mathcal{F}'$. The formula $\Phi \wedge \mathcal{F}'$ is expressed in Conjunctive Normal Form (CNF) so it can be input to standard SAT solvers.

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

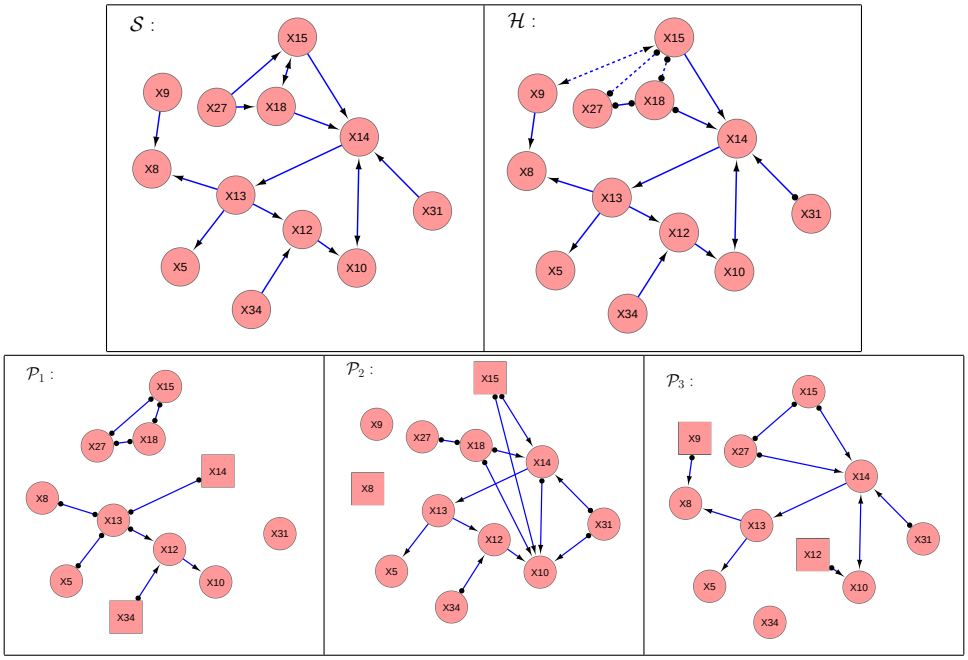

Figure 6: An example of COmbINE input - output. Graph $\mathcal{S}$ is the actual, data-generating, underlying SMCM over 12 variables. PAGs $\mathcal{P}_1, \mathcal{P}_2$ and $\mathcal{P}_3$ are the output of FCI ran with an oracle of conditional independence on three different marginals of $\mathcal{G}$. $\mathcal{H}$ 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 ($X31 \longrightarrow X14 \longleftrightarrow X10$ 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).

in all the possible truth-setting assignments of $\Phi \wedge \mathcal{F}'$, meaning the respective features are invariant in all possibly underlying SMCMs. Such variables are called **backbone** variables of $\Phi \wedge \mathcal{F}'$ (Hyttinen 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 $\Phi \wedge \mathcal{F}'$ and has polarity 0/1. Based on the backbone of $\Phi \wedge \mathcal{F}'$, the final step of COmbINE is to construct the summary graph $\mathcal{S}$. $\mathcal{S}$ has the following types of edges and endpoints:

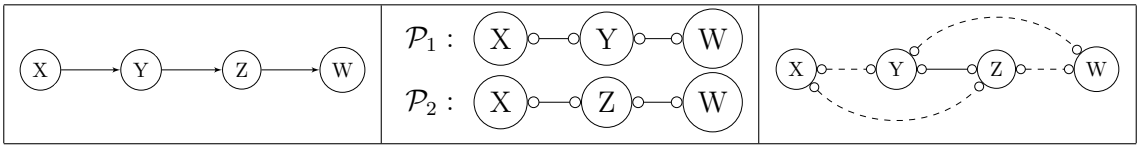

Figure 7: A detailed example of a non-trivial inference. From left to right: The true underlying SMCM over variables X, Y, Z, W; PAGs $\mathcal{P}_1$ and $\mathcal{P}_2$ over $\{X, Y, W\}$ and $\{X, Z, W\}$, respectively; The output $\mathcal{H}$ of Algorithm 2 ran with an oracle of conditional independence. 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 possibly underlying 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).

- **Solid Edges:** Edges in $\mathcal{H}$ that have polarity 1 in $\Phi \wedge \mathcal{F}'$, meaning that they are present in all possibly underlying SMCMs.
- **Absent Edges:** Edges that are not in $\mathcal{H}$ or edges in $\mathcal{H}$ that have polarity 0 in $\Phi \wedge \mathcal{F}'$, meaning that they are absent in all possibly underlying SMCMs.
- **Dashed Edges:** Edges in $\mathcal{H}$ that are not backbone variables in $\Phi \wedge \mathcal{F}'$, meaning that there exists at least one possibly 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 possibly 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 possibly underlying 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 6 and 7 illustrate the output of Algorithm 2, along with the corresponding input PAGs.

We claim that, given an oracle of conditional independence, the SAT-generating procedure described in Algorithm 4 results in a SAT instance $\Phi \wedge \mathcal{F}$ that is satisfied by all

and only the possibly underlying SMCMs for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ (i.e., every SMCM that entails the exact same conditional independencies as those obtained by the oracle for every experiment, after the removal of edges incoming to the manipulated variables). Lemma 17 proves that the every possibly underlying SMCM satisfies $\Phi \wedge \mathcal{F}$, while Lemma 19 proves that if $\mathcal{S}$ is a mixed graph satisfying $\Phi \wedge \mathcal{F}$, $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.

In all subsequent lemmas, theorems and proofs we employ the assumptions A1-A3 and the notation presented in the beginning of Section 4. We also assume the algorithms are run with an oracle of conditional independence and infinite maximum conditioning set size and maximum path length. We only present the main lemmas and theorems here. Auxiliary lemmas and all proofs can be found in Appendix A.

Lemma 17 *For an oracle of conditional independence, if $\mathcal{S}$ is a possibly underlying model for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, and $\Phi \wedge \mathcal{F}$ is the conjunction of the outputs of Algorithm 4, $\mathcal{S}$ satisfies $\Phi \wedge \mathcal{F}$.*

Proof See Appendix A. ■

Lemma 19. *For an oracle of conditional independence, if $\Phi \wedge \mathcal{F}$ is the conjunction of the outputs of Algorithm 4, and $\mathcal{S}$ a mixed graph that satisfies $\Phi \wedge \mathcal{F}$, then $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*

Proof See Appendix A. ■

Soundness and completeness of Algorithm 2 stems from the Lemmas 17 and 19: For the summary graph that is the output of COmbINE soundness means that if a feature is solid in $\mathcal{H}$, the feature is present in all possibly underlying SMCMs for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$. Completeness means that if a feature is dashed in $\mathcal{H}$, there exists at least two possibly underlying SMCM where this feature has different truth values. Since $\Phi \wedge \mathcal{F}$ implicitly represents the entire solution space, and it is satisfied by all and only the possibly underlying SMCMs, soundness and completeness of Algorithm 2 easily follows.

Theorem 20 ***(Soundness and completeness of Algorithm 2)*** *If $\mathcal{H}$ is the output of Algorithm 2, then the following hold:*
Soundness*: If a feature (edge, absent edge, endpoint) is* solid *in $\mathcal{H}$, then this feature is present in* all *SMCMs that are possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*
Completeness*: If a feature is present in* all *SMCMs that are possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, the feature is solid in $\mathcal{H}$.*

Proof See Appendix A. ■

4.3 A Strategy for Conflict Resolution Based on the Maximum Posterior 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 not independent given any subset of $\mathbf{O}_i$.
2. $\neg adjacent(X, Y, \mathcal{P}_i)$: X and Y are independent given some $\mathbf{Z} \subset \mathbf{O}_i$
3. $col(\langle X, Y, Z\rangle, \mathcal{P}_i)$: Y is in no subset of $\mathbf{O}_i$ that renders X and Z independent.
4. $dnc(\langle X, Y, Z\rangle, \mathcal{P}_i)$: 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 colliders and non-colliders with order to the same rank as the non-adjacency of the corresponding discriminating 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:

$$H_0 : \exists \mathbf{Z} \subset \mathbf{O}_i : X \perp\!\!\!\perp Y \mid \mathbf{Z} \text{ than the alternative } H_1 : \nexists \mathbf{Z} \subset \mathbf{O}_i : X \perp\!\!\!\perp Y \mid \mathbf{Z} \tag{1}$$

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

$$H_0 : X \perp\!\!\!\perp Y \mid \mathbf{Z}_{XY} \text{ against the alternative } H_1 : X \not\perp\!\!\!\perp Y \mid \mathbf{Z}_{XY}$$

as a surrogate for the set of hypotheses in Equation 1. Under the null hypothesis, the p-values follow a uniform $\mathcal{U}([0, 1])$ distribution,[2] 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

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

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

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

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

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

For given estimates $\hat{\pi_0}$ and $\hat{\xi}$, the posterior ratio of H_0 against H_1 is

$$E_0(p) = \frac{P(p|H_0)P(H_0)}{P(p|H_1)P(H_1)} = \frac{P(p|p \sim Beta(1, 1))P(p \sim Beta(1, 1))}{P(p|p \sim Beta(\hat{\xi}, 1))P(p \sim Beta(\hat{\xi}, 1))} = \frac{\hat{\pi_0}}{\hat{\xi} p^{\hat{\xi}-1}(1 - \hat{\pi_0})}.$$

$E_0(p) > 1$ 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 posterior 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 posterior ratio (MPR)** for a p-value p to be the maximum between the two:

$$E(p) = max\Big\{\frac{\hat{\pi_0}}{\hat{\xi} p^{\hat{\xi}-1}(1 - \hat{\pi_0})}, \frac{\hat{\xi} p^{\hat{\xi}-1}(1 - \hat{\pi_0})}{\hat{\pi_0}}\Big\}. \tag{4}$$

MPR 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.

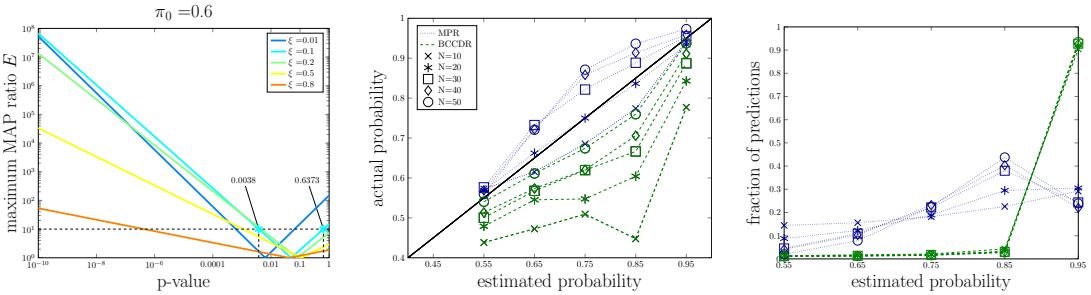

Figure 8: Behaviour and calibration of MPR estimates. (left) Log of the maximum posterior ratio $E(p)$ versus log of the p-value p for $\hat{\pi_0} = 0.6$ and various $\hat{\xi}$. For $\hat{\xi} = 0.1$, an adjacency supported by a maximum p-value of 0.0038 corresponds to the same E as a non-adjacency supported by a p-value of 0.6373. The intersection point of the line with the x axis is the p for which $E_0(p) = E_1(p) = 1$. (center) Probability calibration plots for confidence estimates obtained using MPR estimates $(1/(1 + E_0(p))$ for adjacencies, $E_0(p)/(1 + E_0(p))$ for non-adjacencies). For each interval of length 0.1 in [0.5, 1], the estimated confidences are plotted against the actual frequency of correctness of the corresponding constraints. The green lines correspond to estimates obtained using BCCDR (see Section 5) The confidence estimates correspond to the experiments presented in Figure 10. (right). Number of confidences in each interval.

Notice that, it is possible that for a p-value $E_0(p_{XY}) > E_1(p_{XY})$ (i.e., MPR 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 MPR, independence seems more probable. The reverse situation is also possible. The pseudo-code in Algorithm 5 (Lines 6—10) accepts the MPR decisions for dependencies and independencies; *this implies that some of the decisions made by FCI will be reversed*. Nevertheless, in anecdotal experiments we found that 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 MPR usually does not have a large impact on the output of the algorithm.

Figure 8 shows how the MPR varies with the p-value for $\hat{\pi_0} = 0.6$ and several $\hat{\xi}$'s. The lowest possible value of the MPR 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 8 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 MPR 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 $\{p_j\}_{j=1}^{M}$

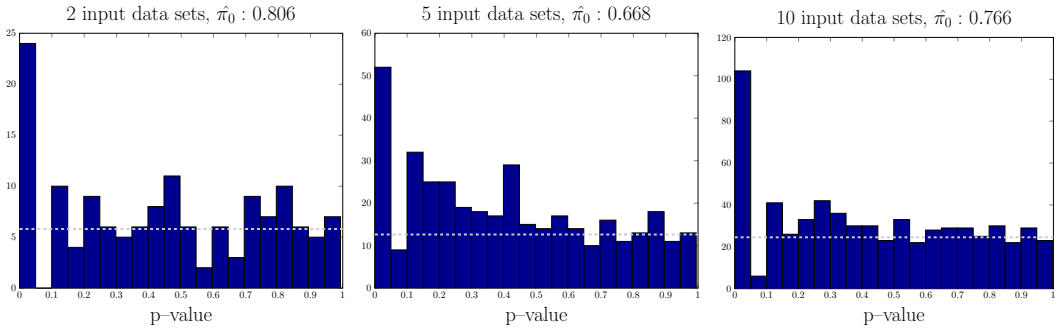

Figure 9: Distribution of p-values and estimated $\hat{\pi_0}$. We used the method of Storey and Tibshirani (2003) to estimate $\hat{\pi_0}$ for a sample of p-values corresponding to 2 (left), 5 (center) and 10 (right) input data sets. We generated networks by manipulating a marginal of the ALARM network (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 $\hat{\pi_0}$.

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 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 9 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 MPR is presented in Algorithm 5. MPR 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

Algorithm 5: MPRstrategy

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

$\mathcal{F}' \leftarrow \emptyset$;
Estimate $\hat{\pi_0}$ from $\{p_j\}$ using the method described in Storey and Tibshirani (2003);
Find $\hat{\xi}$ that minimizes $-\sum_j log(\hat{\pi_0} + (1-\hat{\pi_0})\xi p_j^{\xi-1})$;
foreach *literal* $(\neg)adjacent(X, Y, \mathcal{P}_i) \in \mathcal{F}$ *with p-value* p_j **do**
    $E_0(p_j) \leftarrow \frac{\hat{\pi_0}}{\hat{\xi} p_j^{\hat{\xi}-1}(1-\hat{\pi_0})}$, $E_1(p_j) \leftarrow \frac{\hat{\xi} p_j^{\hat{\xi}-1}(1-\hat{\pi_0})}{\hat{\pi_0}}$;
    if $E_1(p_j) < E_0(p_j)$ **then**
        add $\neg adjacent(X, Y, \mathcal{P}_i)$ to $\mathcal{F}$
    else
        add $adjacent(X, Y, \mathcal{P}_i)$ in $\mathcal{F}$
    end
    $Score(literal) \leftarrow max\{E_0(p_j), E_1(p_j)\}$;
end
foreach *literal* $collider(X, Y, Z, \mathcal{P}_i)$, $dnc(X, Y, Z, \mathcal{P}_i)$ **do**
    if X, Y, Z *is an unshielded triple in* $\mathcal{P}_i$ **then**
        $Score(literal) \leftarrow Score(X, Z, \mathcal{P}_i)$;
    else if $\langle W \ldots X, Y, Z \rangle$ *is discriminating for* Y *in* $\mathcal{P}_i$ **then**
        $Score(literal) \leftarrow Score(W, Z, \mathcal{P}_i)$;
    end
end
$\mathcal{F} \leftarrow$ sort $\mathcal{F}$ by descending score;
foreach $\phi \in \mathcal{F}$ **do**
    if $\Phi \wedge \phi$ *is satisfiable* **then**
        $\Phi \leftarrow \Phi \wedge \phi$;
        Add ϕ to $\mathcal{F}'$;
    end
end

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 posterior 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 another alternative

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.

algorithm in the literature. We also present a comparative evaluation of conflict resolution methods, including the one based on the proposed MPR 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 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 correct output, 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 as the correct output for comparison.* 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 7 (middle), the correct output, shown in Figure 7 (right), has one solid edge out of 5, no solid endpoint, one absent, and four dashed edges. Dashed edges and endpoints in the output of the algorithm can only be evaluated if one knows this correct output. Unfortunately, the correct output cannot be recovered in a timely fashion in most problems involving more than 15 variables, as shown in Section 5.1.

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:

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

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.

# 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.

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 data sets 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 $X \star\!\!-\!\!\star Y$ 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 suggested.[3] We denote this version of the algorithm as SBCSD′. Also note, that the available implementation of

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.

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 acyclicity 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 (**MPR**) against three other alternatives: A ranking strategy where constraints are sorted based on Bayesian probabilities as proposed in Claassen and Heskes (2012) (**BCCDR**), as well as a Max-SAT (**MaxSAT**) and a weighted max-SAT (**wMaxSAT**) approach.

MPR: 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 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 $P(f) = \sum_{\mathcal{G} \vdash \mathrm{f}} P(\mathbf{D}|\mathcal{G})P(\mathcal{G})$. Scoring all DAGs is practically infeasible for networks with

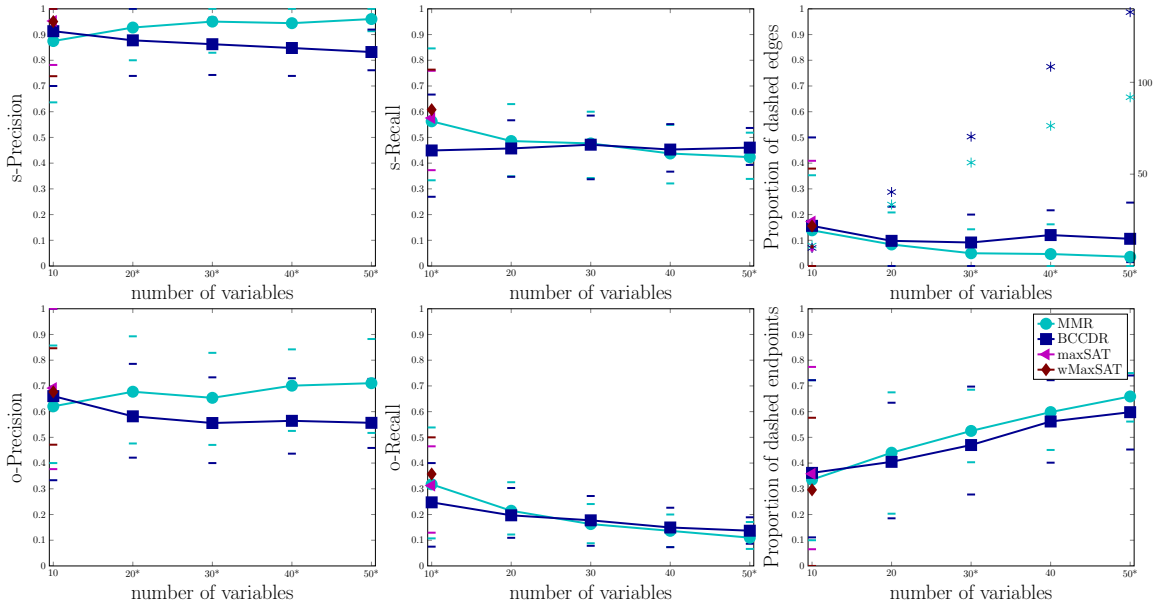

Figure 10: Learning performance of COmbINE with various conflict resolution strategies. From left to right: Median s-Precision, s-Recall, proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints (bottom) for networks of several sizes for various conflict resolution 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. Coloured bars indicate 5 and 95 percentiles. Asterisks in the top right figure show the absolute number of literals rejected by each strategy (y axis on the right). Asterisks on x tick labels indicate cases where the behaviour of MPR and BCCDR are significantly different (paired t-test of equality of means with unknown but equal variances).

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 $X \star\!\!-\!\!\star Y$ 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. (Non) colliders are assigned the same score as the non adjacency of their endpoints.

We use the BGE metric for Gaussian distributions (Geiger and Heckerman, 1994) as implemented in the BDAGL package Eaton and Murphy (2007a) 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 MPR. Again, each hard-constraint 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 MPR method (calculating MPRs 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 10, where we can see the median performance of both algorithms over 50 iterations. Overall, **MPR** 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 **MPR** is not provided with sufficient number of p-values to estimate $\hat{\pi}_0$ and $\hat{\xi}$. In terms of computational complexity, for networks of 50 variables, estimating the **BCCDR** ratios takes about 150 seconds on average, while estimating the **MPR** 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 **MPR** is a reasonable and relatively efficient conflict resolution strategy.

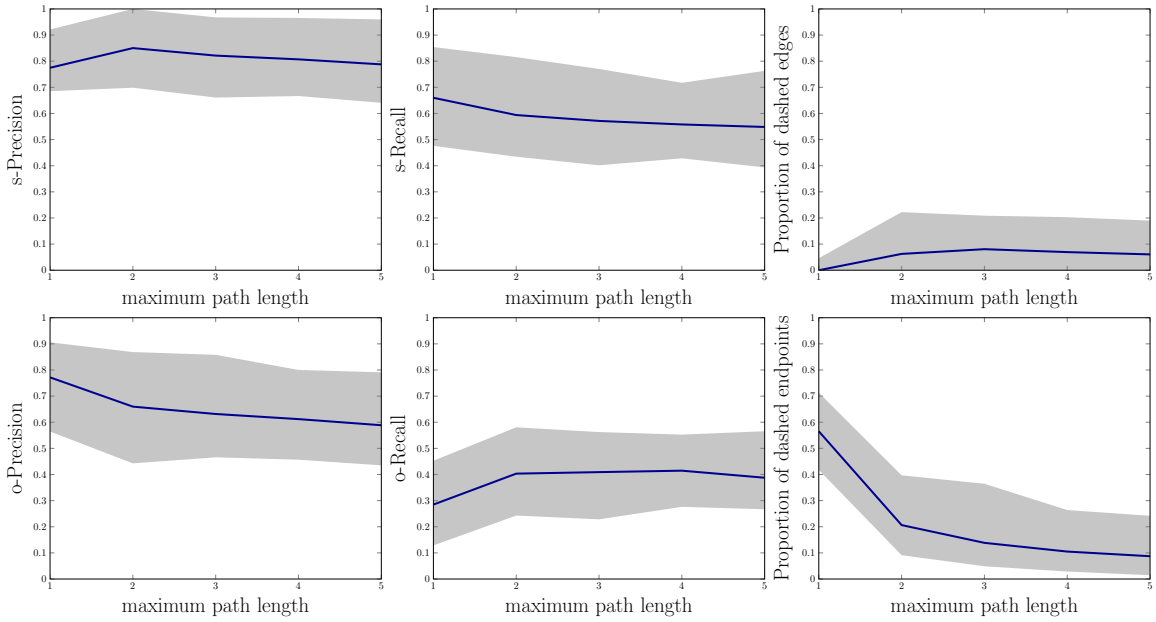

Figure 11: Learning performance of COmbINE against maximum path length. From left to right: s-Precision, s-Recall, percentage dashed edges and o-Precision, o-Recall and percentage of dashed endpoints (bottom) for varying maximum path length, averaged over all networks. Shaded area ranges from the 5 to the 95 percentile. Maximum path length 3 seems to be a be a reasonable trade-off between performance, percentage of dashed features, and efficiency.

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 11. 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 5 of Algorithm 3). When the maximum length increases, adjacencies in one data set, can 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.

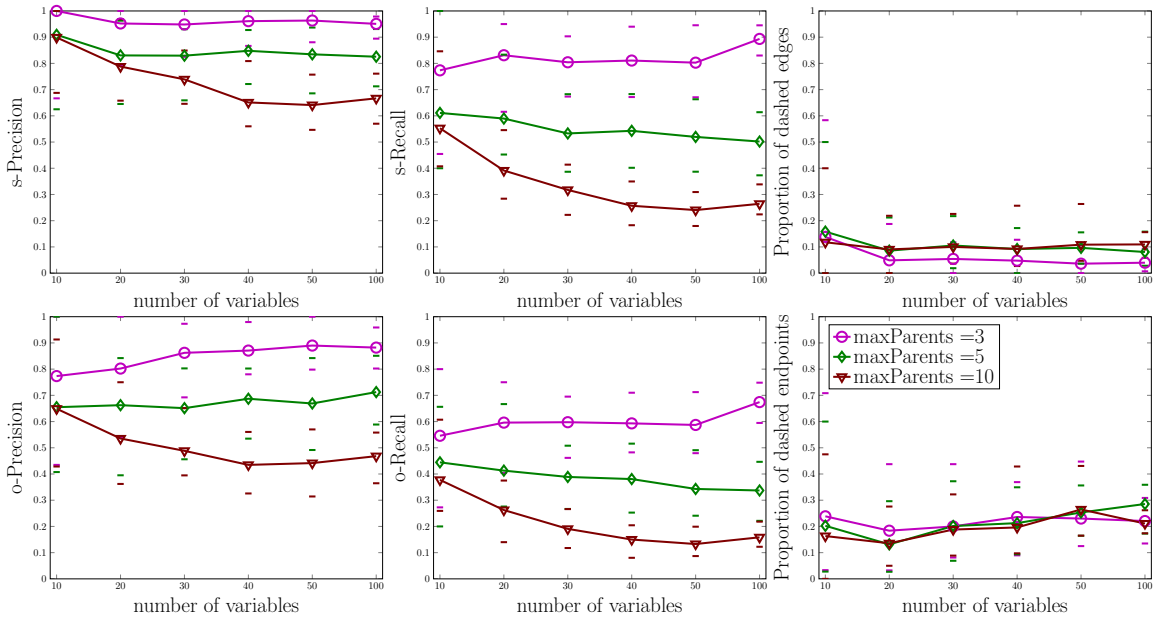

Figure 12: Learning performance of COmbINE for various network sizes and densities. From left to right: Median s-Precision, s-Recall, proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints (bottom) for varying network size and density. Density is controlled by limiting the number of possible parents per variable. Coloured bars indicate 5 and 95 percentiles. As expected, the performance deteriorates as networks become denser.

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

We also calculated confusion matrices for edges and endpoints inferred by COmbINE against the *correct output* structure $\mathcal{H}$ for networks of 10 variables, where $\mathcal{H}$ can be obtained by running COmbINE with an oracle of conditional independence and unrestricted path length and conditioning set size. Table 3 shows the resulting confusion matrices for

		Actual $\mathcal{H}$					
		maxParents 3			maxParents 5		
	Edges	solid	dashed	absent	solid	dashed	absent
$\hat{\mathcal{H}}$	solid	**8.0** (4.0, 12.0)	**0.0** (0.0, 5.0)	**0.0** (0.0, 4.0)	**9.0** (3.0, 13.0)	**1.0** (0.0, 10.0)	**1.0** (0.0, 5.0)
	dashed	**0.0** (0.0, 3.0)	**0.0** (0.0, 4.0)	**0.0** (0.0, 2.0)	**0.5** (0.0, 4.0)	**0.5** (0.0, 3.0)	**1.0** (0.0, 2.0)
	absent	**1.0** (0.0, 4.0)	**0.0** (0.0, 3.0)	**31.0** (24.0, 36.0)	**2.5** (0.0, 8.0)	**1.5** (0.0, 9.0)	**24.0** (14.0, 34.0)
	Endpoints	arrow	circle	tail	arrow	circle	tail
	arrow	**8.0** (4.0, 12.0)	**1.0** (0.0, 5.0)	**0.0** (0.0, 3.0)	**8.0** (4.0, 13.0)	**3.0** (0.0, 8.0)	**2** (0.0, 5.0)
	circle	**1.0** (0.0, 3.0)	**3.0** (0.0, 14.0)	**0.0** (0.0, 2.0)	**1.0** (0.0, 5.0)	**3.0** (0.0, 8.0)	**1.0** (0.0, 4.0)
	tail	**0.0** (0.0, 2.0)	**0.0** (0.0, 5.0)	**4.0** (0.0, 8.0)	**1.0** (0.0, 5.0)	**1** (0.0, 54.0)	**3.0** (1.0, 6.0)

Table 3: Confusion matrices reporting edge and endpoint counts of the output of COmbINE $\hat{\mathcal{H}}$ versus the actual summary graph $\mathcal{H}$. Results are shown for 10 variables and 5 data sets of 1000 samples each. $\mathcal{H}$ was obtained using COmbINE with an oracle of conditional independence, and unconstrained maxK and maximum path length parameters. The table reports median values (bold) along with the 5 and 95 percentiles (in parenthesis). Results are in agreement with the metrics used for larger networks.

maxParents 3 and 5 and 5 data sets of sample size 1000. Overall, the results are in concordance with the metrics used for larger networks, and confirm that the method works best for sparser networks. Notice that for dense networks (for N=10 and maxParents =5, the networks have about 40% of all possible edges), there are cases where the actual correct output includes a large proportion of dashed edges, while constricting the maximum path length forces the algorithm to accept more solid features (hence the wide percentiles).

5.5 COmbINE Performance over Sample Size and Number of Input Data Sets

Figure 13 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 14 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 15 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 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 12).

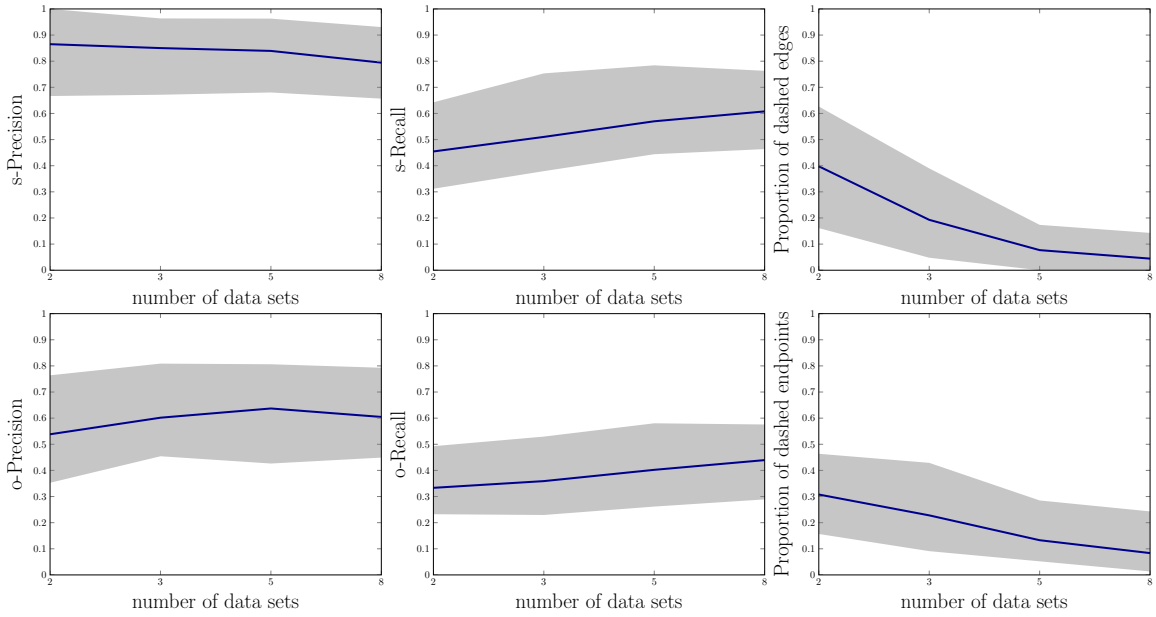

Figure 13: Learning performance of COmbINE for varying number of input data sets. From left to right: Median s-Precision, s-Recall, Proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints of (bottom) for varying number of input data sets. Shaded area ranges from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

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 16 illustrates the running time of COmbINE against these factors. As we can see in Figure 16, 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 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.

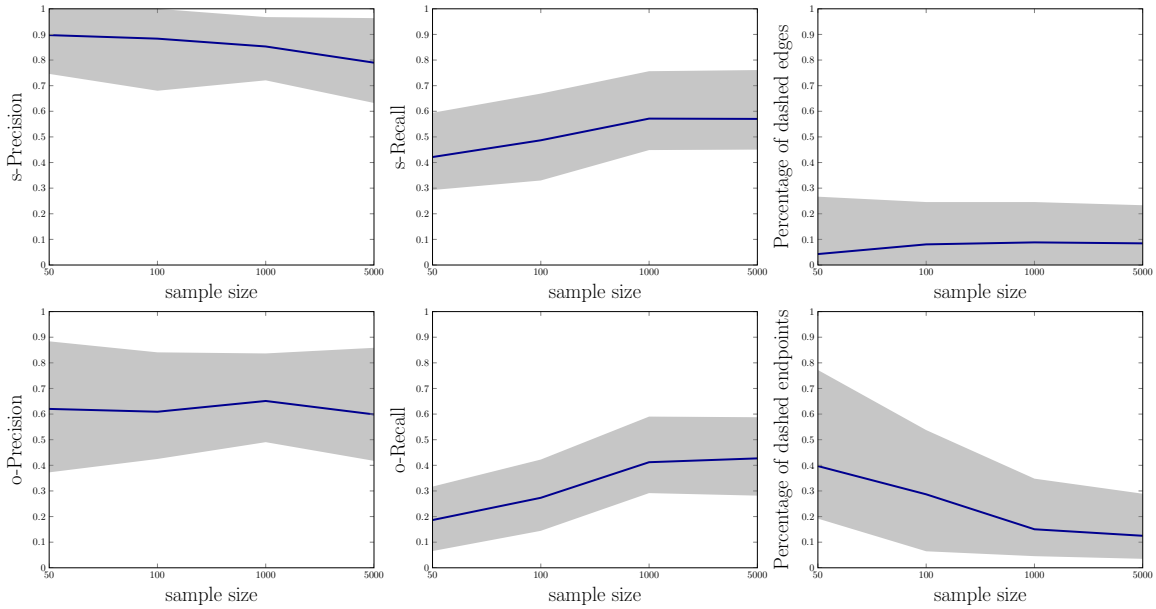

Figure 14: Learning performance of COmbINE for varying sample size per data set. From left to right: s-Precision, s-Recall, Proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints of (bottom) for varying sample size per data set. Shaded area ranges from the 5 to the 95 percentile. Increasing the sample size improves the performance of the algorithm.

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 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 (PBMC) 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.

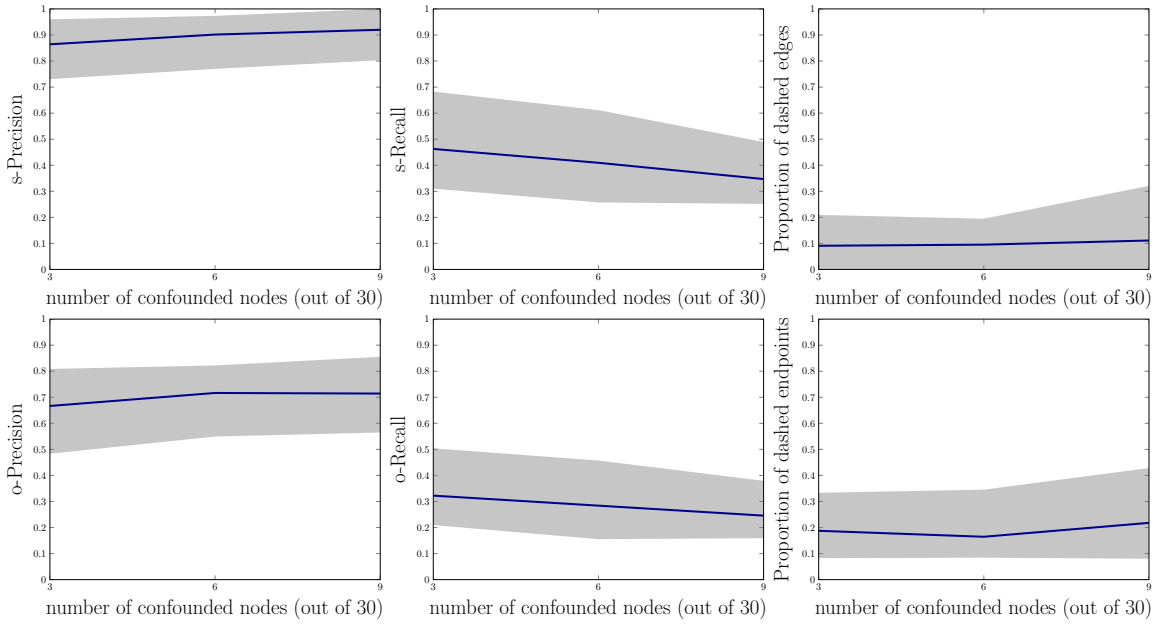

Figure 15: Learning performance of COmbINE for varying percentage of confounded variables. From left to right: s-Precision, s-Recall, percentage of dashed edges (top) and o-Precision, o-Recall and percentage of dashed endpoints (bottom) for varying number of confounded nodes for networks of 30 variables. 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.

We focused on two sub-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, pSHP2 and pPlcg2 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. (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. Specific inhibitors can be abundance inhibitors, which affect the level of measured protein, and activity inhibitors, which affect the function of measured proteins. The former are closer to ideal hard interventions. Activity inhibitors have been modelled in several ways in the literature. Sachs et al. (2005) model them as ideal interventions by manually setting the values to the lowest discretization level. Itani et al. (2010) propose splitting the target variable in two nodes, one used to represent the inhibition and the other

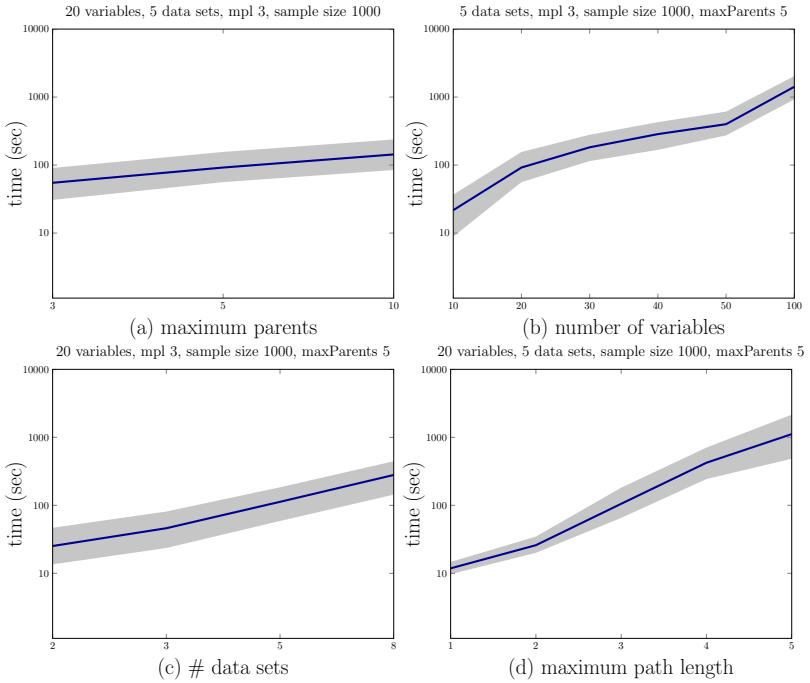

Figure 16: Running time of COmbINE. From left to right: Running time (in seconds) is plotted in logarithmic scale against maximum parents per variable and number of variables (top row); number of data sets and maximum path length (bottom row). Shaded area ranges from the 5 to the 95 percentile. The number 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.

used to represent the abundance. Mooij and Heskes (2013) propose modelling activity inhibitions by removing outgoing edges of the target variable. Notice that this type of modelling can be easily encoded in a SAT representation.

We used abundance inhibitors that we believe 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). The maximum dosage of each inhibitor was used. For all three interventions, the distribution of the target variable under zero dosage is differs significantly (according to a Kolmogorov-Smirnov test with significance threshold 0.05) from the distribution of the target variable for the maximum dosage, indicating that the inhibitor has an effect on the abundance of the target protein. Nevertheless, we must point out that the interventions may not be entirely ideal. 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 4. 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

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 4: 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.

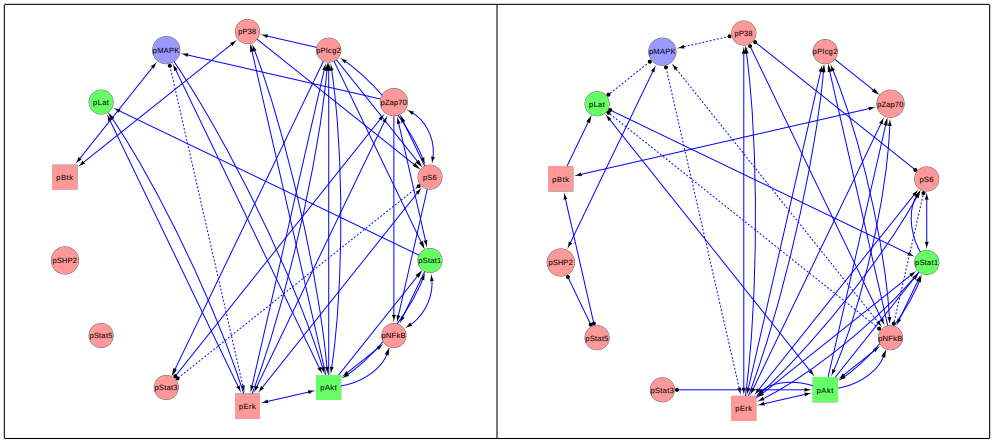

Figure 17: A case study for COmbINE: Mass cytometry data. COmbINE was run on 4 different mass cytometry data for two different cell populations: CD4+ T-cells (left) and CD8+ T-cells (right). In each data set, one variable was manipulated (pAkt, pBTk, pErk, pErk respectively). Variables pAkt, pLat and pStat1 are only measured in data sets 1-3, while pMAPK is only measured in data set 4.

FCI with $\alpha = 0.05$ and maxK=5. We used Cytoscape (Smoot et al., 2011) to visualize the summary graphs produced by COmbINE, illustrated in Figure 17.

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 of a direct relation of pAkt and pMAPK in CD4+ cells, *even though the variables are not jointly measured in any of the input data sets*. 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. In addition to the potential imperfect manipulations described above, COmbINE also 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).

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.

Acknowledgements

We thank the anonymous reviewers and the action editor for their constructive comments, their thorough reviews really helped improve the manuscript. We also thank Vincenzo Lagani and Giorgos Borboudakis for comments and suggestions on early versions of this work, and Tom Claassen for providing clarifications on the BCCD algorithm. ST and IT were funded by the STATegra EU FP7 project, No 306000. IT was partially funded by the ERC Consolidator Grant No 617393 CAUSALPATH, as well as the EPILOGEAS GSRT ARISTEIA II project, No 3446, which is part of the NSRF 2007-2013 Education and Lifelong Learning Program, co-financed by the European Union (European Social Fund) and national resources.

Appendix A. Proofs

We now present proofs for propositions and theorems presented in the main section.

Proposition 12 *Let* $\mathbf{O}$ *be a set of variables and* $\mathcal{J}$ *the independence model over* $\mathbf{O}$*. Let* $\mathcal{S}$ *be a SMCM over variables* $\mathbf{O}$ *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* $\mathbf{L}$*,* $\mathbf{L} \subseteq \mathbf{O}$ *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 10 there exists no $\mathbf{Z} \subseteq \mathbf{O} \setminus \mathbf{L} \cup \{X, Y\}$ 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 $\mathbf{Z} \subseteq \mathbf{O} \setminus \mathbf{L} \cup \{X, Y\}$ such that X and Y are m-separated given $\mathbf{Z}$ in $\mathcal{M}$. Thus, by Theorem 9 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 9 there exists no $\mathbf{Z} \subseteq \mathbf{O} \setminus \mathbf{L} \cup \{X, Y\}$ 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 $\mathbf{Z} \subseteq \mathbf{O} \setminus \mathbf{L} \cup \{X, Y\}$ such that X and Y are m-separated given $\mathbf{Z}$ in $\mathcal{S}$. Thus, by Theorem 10 there exists an inducing path between X and Y with respect to $\mathbf{L}$ in $\mathcal{S}$. ■

Theorem 13 *Let* $\mathbf{O}$ *be a set of variables and* $\mathcal{J}$ *the independence model over* $\mathbf{O}$*. Let* $\mathcal{S}$ *be a SMCM over variables* $\mathbf{O}$ *that is faithful to* $\mathcal{J}$*. Let* $\mathcal{M}$ *= SMCMtoMAG(*$\mathcal{S}$*). Then* $\mathcal{S}$ *and* $\mathcal{M}$ *share the same ancestry relations and* $\mathcal{J}_m(\mathcal{S}) = \mathcal{J}_m(\mathcal{M})$*, hence the two graphs entail the same independence model.*

Proof $\mathcal{S}$ and $\mathcal{M}$ share the same ancestry relations, since during Algorithm 1 a directed edge $X \rightarrow Y$ is added only if X is an ancestor of Y in $\mathcal{S}$, and no directed edges are removed. To prove that the $\mathcal{J}_m(\mathcal{S}) = \mathcal{J}_m(\mathcal{M})$, consider a DAG $\mathcal{G}$ constructed from $\mathcal{S}$ as follows: For every bi-directed edge $X \leftrightarrow Y$, introduce a new node L_{XY}. Remove $X \leftrightarrow Y$ and add $X \leftarrow L_{XY} \rightarrow Y$. Let $\{L_{V_i V_j}\}$ be the set of nodes added by this procedure. Obviously, $\mathcal{G}$ is a DAG and $\mathcal{G}$ and $\mathcal{S}$ share the same ancestry relations and the same m-separations for variables in $\mathbf{O}$, thus $\mathcal{J}_m(\mathcal{S}) = \mathcal{J}_m(\mathcal{G})[_{\mathbf{L}}$. If $\langle X, V_1, \ldots, V_n, Y \rangle$ is a primitive inducing path

in $\mathcal{S}$, then $\langle X, L_{XV_1}, V_1, \ldots, L_{V_{n-1}V_n}, V_n, L_{V_nY}, Y\rangle$ is an inducing path with respect to $\mathbf{L}$ in $\mathcal{G}$ and vice versa. Thus, X and Y are adjacent in $\mathcal{G}[_{\mathbf{L}}$ if only if there exists a primitive inducing path between X and Y in $\mathcal{S}$, and $\mathcal{G}$ shares the same ancestry relations with $\mathcal{S}$ for variables in $\mathbf{O}$, thus by Definition 3, $\mathcal{G}[_{\mathbf{L}} = \mathcal{M}$. By Theorem 4 (Richardson and Spirtes, 2002) $\mathcal{J}_m(\mathcal{M}) = \mathcal{J}_m(\mathcal{G}[_{\mathbf{L}}) = \mathcal{J}_m(\mathcal{G})[_{\mathbf{L}} = \mathcal{J}_m(\mathcal{S})$. ■

In all subsequent lemmas, theorems and proofs we employ the assumptions and notation presented in Section 4 (Assumptions A1-A3 and notation presented beneath them). We also assume the algorithms are run with an oracle of conditional independence and infinite maximum conditioning set size and maximum path length.

The following theorem proves that a $\mathcal{S}$ is possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^n$ and $\{\mathbf{I}_i\}_{i=1}^N$ if and only if the result of manipulating $\mathbf{I}_i$, adding necessary edges to create a Markov equivalent MAG and then marginalizing out variables in $\mathbf{L}_i$ produces a MAG $\mathcal{M}_i$ that belongs to the Markov equivalence class represented by $\mathcal{P}_i$ for all experiments.

Theorem 14 *If $\mathcal{S}$ is a SMCM, $\{\mathcal{J}_i\}_{i=1}^N$ is a family of independence models, $\{\mathbf{I}_i\}_{i=1}^N$ is a family of intervention targets and $\mathcal{P}_i$ is the PAG of the Markov equivalence class of MAGs faithful to $\mathcal{J}_i$, the following statements are equivalent:*

- *$\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*
- *$\forall i, \mathcal{M}_i \in \mathcal{P}_i$, where $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i}$.*

Proof The following hold:

$$\mathcal{S} \text{ is a possibly underlying SMCM for } \{\mathcal{J}_i\}_{i=1}^N \text{ and } \{\mathbf{I}_i\}_{i=1}^N \Leftrightarrow \mathcal{J}_m(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} = \mathcal{J}_i \quad \forall i$$

(by definition)

$$\mathcal{J}_m(\text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i}))[_{\mathbf{L}_i} = \mathcal{J}_m(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} = \mathcal{J}_i \quad \forall i \quad \text{(by Theorem 13)}$$

$$\mathcal{J}_m(\text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i}) = \mathcal{J}_m(\text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i}))[_{\mathbf{L}_i} = \mathcal{J}_i \quad \forall i \quad \text{(by Theorem 4)}$$

$$\mathcal{J}_m(\mathcal{M}_i) = \mathcal{J}_i \quad \forall i, \text{ and by definition of } \mathcal{P}_i, \quad \mathcal{M}_i \in \mathcal{P}_i \quad \forall i.$$

■

The following Lemma proves that no inducing and ancestral paths present in the true underlying SMCM are ruled out during the construction of the initial search graph, and is necessary for subsequent proofs. We prove that $\mathcal{H}_{in}$ has a superset of edges and a subset of orientations compared to $\mathcal{S}$.

Lemma 15 *If $\mathcal{H}_{in}$ is the initial search graph returned by Algorithm 3 for $\{\mathcal{P}_i\}_{i=1}^N$, and $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, then the following hold: If p is an ancestral path in $\mathcal{S}$, then p is a possibly ancestral path in $\mathcal{H}_{in}$. Similarly, if p is an inducing path with respect to $\mathbf{L}$ in $\mathcal{S}$, then p is a possibly inducing path with respect to $\mathbf{L}$ in $\mathcal{H}_{in}$.*

Proof We will first prove that $\mathcal{H}_{in}$ has a superset of edges compared to $\mathcal{S}$, and therefore any path in $\mathcal{S}$ is a path also in $\mathcal{H}_{in}$. 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 $\mathcal{H}_{in}$ in Line 3 of Algorithm 3.

2. $\nexists i$ s.t. $X, Y \in \mathbf{O}_i \setminus \mathbf{I}_i$. Then the edge is added to $\mathcal{H}_{in}$ in Line 8 of Algorithm 3.

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

Arrowheads are added to $\mathcal{H}_{in}$ in Lines 5, 9 or 10 of the Algorithm. Arrowheads added in Line 5 occur in all $\mathcal{P}_i$. If $X \ast\!\!\rightarrow Y$ in any $\mathcal{P}_i$, this means that Y is not an ancestor of X in $\mathcal{S}^{\mathbf{I}_i}$. Assume that $X \leftarrow\!\!\ast Y$ 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 $\mathcal{S}$, $X \ast\!\!\rightarrow Y$ in $\mathcal{S}$.

Arrows added to $\mathcal{H}_{in}$ in Lines 9 and 10 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 experiment 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, $\mathcal{H}_{in}$ 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 $\mathcal{H}_{in}$. 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 $\mathcal{H}_{in}$. ■

We can now prove that if a SMCM $\mathcal{S}$ entails all and only the observed conditional independencies for all experiments (and is therefore a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$), then $\mathcal{S}$ satisfies $\Phi \wedge \mathcal{F}$. We say that $\mathcal{S}$ *satisfies a constraint* ϕ if the truth-values assigned to *edge*, *arrow* and *tail* variables by their corresponding configuration in $\mathcal{S}$ satisfies ϕ. To simplify the proof, we first prove the following lemma:

Lemma 16 *If $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, and $X \rightarrow Y$ is in $\mathcal{P}_i$, then $\mathcal{S}$ satisfies $ancestor(X, Y, i)$. Similarly, if $X \circ\!\!\rightarrow Y$ is in $\mathcal{P}_i$, then $\mathcal{S}$ satisfies $\neg ancestor(Y, X, i)$.*

Proof By Theorem 14 SMCMtoMAG $(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} \in \mathcal{P}_i$. Thus, if $X \rightarrow Y$ is in $\mathcal{P}_i$, then X is an ancestor of Y in $\mathcal{S}^{\mathbf{I}_i}$ (there exists an ancestral path from X to Y in $\mathcal{S}^{\mathbf{I}_i}$). Let $p_1, \ldots, p_M$ be the possibly ancestral paths (there exists at least one: if $X \rightarrow Y$ in $\mathcal{P}_i$, then $X \ast\!\!\ast Y$ is a possibly inducing path in $\mathcal{H}_{in}$) from X to Y in $\mathcal{H}_{in}$. The constraint $ancestor(X, Y, i)$ is realized in $\Phi \wedge \mathcal{F}$ as $ancestor(Y, X, i) \wedge [ancestor(Y, X, i) \leftrightarrow ancestral(p_1, i) \vee ancestral(p_2, i) \cdots \vee ancestral(p_M, i)]$. This is equivalent to $ancestral(p_1, i) \vee ancestral(p_2, i) \cdots \vee ancestral(p_M, i)$. If a path is ancestral in $\mathcal{S}^{\mathbf{I}_i}$, the path is also ancestral in $\mathcal{S}$. By Lemma 15, if a path is ancestral in $\mathcal{S}$, the path is possibly ancestral in $\mathcal{H}_{in}$. Hence, at least one of $p_1, \ldots, p_M$ is ancestral in $\mathcal{S}^{\mathbf{I}_i}$, and $\mathcal{S}$ satisfies $ancestor(X, Y, i)$.

If $X \circ\!\!\rightarrow Y$ is in $\mathcal{P}_i$, then, since SMCMtoMAG $(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} \in \mathcal{P}_i$, there can be no ancestral path from Y to X in $\mathcal{S}^{\mathbf{I}_i})$. Let $p_1, \ldots, p_M$ be the possibly ancestral paths (if any) from Y to X in $\mathcal{H}_{in}$. The constraint $\neg ancestral(Y, X, i)$ is realized in $\Phi \wedge \mathcal{F}$ as $\neg ancestor(Y, X, i) \wedge [ancestor(Y, X, i) \leftrightarrow ancestral(p_1, i) \vee ancestral(p_2, i) \cdots \vee ancestral(p_M, i)]$. This is equivalent to $\neg ancestral(p_1, i) \wedge \neg ancestral(p_2, i) \cdots \wedge \neg ancestral(p_M, i)$. None of these paths are ancestral in $\mathcal{S}^{\mathbf{I}_i}$, therefore $\mathcal{S}$ satisfies $ancestor(X, Y, i)$. ■

We can now prove that any possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ satisfies $\Phi \wedge \mathcal{F}$.

Lemma 17 *For an oracle of conditional independence, if $\mathcal{S}$ is a possibly underlying model for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, and $\Phi \wedge \mathcal{F}$ is the conjunction of the outputs of Algorithm 4, $\mathcal{S}$ satisfies $\Phi \wedge \mathcal{F}$.*

Proof By Theorem 14, since $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} \in \mathcal{P}_i \quad \forall i$.

1. **Constraints added in Lines 8, 9 of Algorithm 4**. These constraints are satisfied since $\mathcal{S}$ is an acyclic mixed graph.

2. **Adjacency constraints added in Lines 4, 5, 6 of Algorithm 4**. Assume that for a pair of variables X, Y adjacent in $\mathcal{P}_i$, there exist M possibly inducing paths in $\mathcal{H}_{in}$, namely $p_1, , \ldots, p_M$. For this adjacency, the following constraint is added in $\Phi \wedge \mathcal{F}$ in Lines 4 and 5 of Algorithm 4:

   $$adjacent(X, Y, \mathcal{P}_i) \wedge [adjacent(X, Y, \mathcal{P}_i) \leftrightarrow inducing(p_1, i) \vee \cdots \vee inducing(p_M, i)],$$

   which is equivalent to

   $$inducing(p_1, i) \vee \cdots \vee inducing(p_M, i).$$

   Since $\mathcal{M}_i \in \mathcal{P}_i$, X and Y are adjacent in $\mathcal{M}_i$. By Proposition 12 there exists an inducing path p^* between X and Y with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$. By Lemma 15, this path is a possibly inducing path in $\mathcal{H}_{in}$, thus, $\exists i \in [1, \ldots, M]$ such that $p^* = p_i$. Thus, the constraint $inducing(p_1, i) \vee \cdots \vee inducing(p_M, i)$ is satisfied by $\mathcal{S}$.

   Similarly, if X and Y are not adjacent in $\mathcal{P}_i$, the constraint

   $$\neg adjacent(X, Y, \mathcal{P}_i) \wedge [adjacent(X, Y, \mathcal{P}_i) \leftrightarrow inducing(p_1, i) \vee \cdots \vee inducing(p_M, i)]$$

   is added to $\Phi \wedge \mathcal{F}$ in Lines 4 and 6 of Algorithm 4. The constraint is equivalent to

   $$\neg inducing(p_1, i) \wedge \cdots \wedge \neg inducing(p_M, i).$$

   Since X and Y are not adjacent in $\mathcal{M}_i$, by Proposition 12 there exists no inducing path with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$. Thus, none of the paths (if any) $p_1, \ldots, p_M$ is inducing with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$, and the constraint $\neg inducing(p_1, i) \wedge \cdots \wedge \neg inducing(p_M, i)$ is satisfied by $\mathcal{S}$.

3. **Unshielded (non) collider constraints added in Lines 13,14, 15,16 of Algorithm 4.** For an unshielded collider $X \star\!\!-\!\!\star Y \star\!\!-\!\!\star Z$ in $\mathcal{P}_i$, the constraint

$$col(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge \big[col(\langle X,Y,Z\rangle,\mathcal{P}_i) \rightarrow unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge collider(\langle X,Y,Z\rangle,\mathcal{P}_i)\big],$$

which is equivalent to

$$unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge collider(\langle X,Y,Z\rangle,\mathcal{P}_i)$$

is added in Lines 14 and 15. As shown in Figure 4,

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

and

$$collider(\langle X,Y,Z\rangle,\mathcal{P}_i) \leftrightarrow \neg ancestor(Y,X,i) \wedge \neg ancestor(Y,Z,i)$$

. Since $\mathcal{M}_i \in \mathcal{P}_i$, $X \star\!\!\rightarrow Y \leftarrow\!\!\star Z$ is an unshielded triple in $\mathcal{M}_i$, $adjacent(X,Y,\mathcal{P}_i) \wedge adjacent(Y,Z,\mathcal{P}_i) \wedge \neg adjacent(X,Z,\mathcal{P}_i)$ is satisfied (as described above for adjacency constraints). Since $X \star\!\!\rightarrow Y \leftarrow\!\!\star Z$ in $\mathcal{P}_i$, by Lemma 16 constraints $\neg ancestor(Y,X,i) \wedge \neg ancestor(Y,Z,i)$ are satisfied by $\mathcal{S}$.

For an unshielded definite non collider $X \star\!\!-\!\!\star Y \star\!\!-\!\!\star Z$ in $\mathcal{P}_i$, the constraint

$$dnc(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge \big[dnc(\langle X,Y,Z\rangle,\mathcal{P}_i) \rightarrow unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge \neg collider(\langle X,Y,Z\rangle,\mathcal{P}_i)\big],$$

is added in Lines 13 and 16 of Algorithm 4, which is equivalent to

$$unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge \neg collider(\langle X,Y,Z\rangle,\mathcal{P}_i).$$

Since $\mathcal{M}_i \in \mathcal{P}_i$, $X \star\!\!-\!\!\star Y \star\!\!-\!\!\star Z$ is an unshielded triple in $\mathcal{M}_i$, so $unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i)$ is satisfied by $\mathcal{S}$ as described above. Moreover, since either $Y \rightarrow X$ in $\mathcal{M}_i$, or $Y \rightarrow Z$ in $\mathcal{M}_i$, by Lemma 16 $ancestor(Y,X,i) \vee ancestor(Y,Z,i)$ is satisfied by $\mathcal{S}$.

4. **Discriminating (non) collider constraints added in Lines 19, 20,21, 22 of Algorithm 4**. If $\langle W,\ldots,X,Y,Z\rangle$ is a discriminating path for Y in $\mathcal{P}_i$, and Y is a collider on the path in $\mathcal{P}_i$, the following constraint is added in $\Phi \wedge \mathcal{F}$ and in Lines 19 and 21 of Algorithm 4:

$$col(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge \big[col(\langle X,Y,Z\rangle,\mathcal{P}_i) \rightarrow discriminating(p_{WZ},Y,\mathcal{P}_i) \wedge collider(\langle X,Y,Z\rangle,\mathcal{P}_i)\big],$$

which is equivalent to

$$discriminating(p_{WZ},Y,\mathcal{P}_i) \wedge collider(\langle X,Y,Z\rangle,\mathcal{P}_i).$$

Since $\mathcal{M}_i \in \mathcal{P}_i$, the path is discriminating for Y in $\mathcal{M}_i$ and the triple is a collider in $\mathcal{M}_i$. The constraint for the discriminating path is analyzed as a conjunction of the individual features ((non) adjacencies and endpoints) of the path as shown in Figure

4. Since the path is discriminating in $\mathcal{M}_i$, all these adjacency and ancestry constraints are satisfied by $\mathcal{S}$, by the proof for adjacency constraints and Lemma 16. In addition, the triple is a collider in $\mathcal{M}_i$, thus $collider(\langle X, Y, Z\rangle, \mathcal{P}_i)$ is satisfied by $\mathcal{S}$ as described for unshielded colliders.

Similarly, if $\langle W, \ldots, X, Y, Z\rangle$ is a discriminating path for Y in $\mathcal{P}_i$, and Y is a definite non collider on the path in $\mathcal{P}_i$, the following constraint is added in $\Phi \wedge \mathcal{F}$ and in Lines 20 and 22 of Algorithm 4:

$$dnc(\langle X, Y, Z\rangle, \mathcal{P}_i) \wedge \\ \big[dnc(\langle X, Y, Z\rangle, \mathcal{P}_i) \rightarrow discriminating(p_{WZ}, Y, \mathcal{P}_i) \wedge \neg collider(\langle X, Y, Z\rangle, \mathcal{P}_i)\big],$$

which is equivalent to

$$discriminating(p_{WZ}, Y, \mathcal{P}_i) \wedge \neg collider(\langle X, Y, Z\rangle, \mathcal{P}_i).$$

Since $\mathcal{M}_i \in \mathcal{P}_i$, the path is discriminating for Y in $\mathcal{M}_i$ and the triple is a non-collider in $\mathcal{M}_i$. The constraint for the discriminating path satisfied by $\mathcal{S}$ as described above. In addition, the triple is a non-collider in $\mathcal{M}_i$, thus $\neg collider(\langle X, Y, Z\rangle, \mathcal{P}_i)$ is satisfied by $\mathcal{S}$ as described for unshielded definite non colliders.

Thus, $\mathcal{S}$ satisfies all constraints in $\Phi \wedge \mathcal{F}$. ■

To prove completeness for Algorithm 4, we must show that the opposite also holds: If $\mathcal{S}$ is a truth-setting assignment of $\Phi \wedge \mathcal{F}$, $\mathcal{S}$ entails all and only the conditional independencies observed in $\{\mathcal{J}_i\}_{i=1}^N$ for each experiment. According to Theorem 14, we need to show that any truth setting assignment of $\Phi \wedge \mathcal{F}$ results, in each experiment i (after the respective procedures of manipulation, conversion to MAG and marginalization) in a MAG $\mathcal{M}_i$ that belongs to the Markov equivalence class represented by $\mathcal{P}_i$. Thus, we need to show that $\mathcal{M}_i$ has the same adjacencies and colliders with order as any MAG $\mathcal{M}' \in \mathcal{P}_i$. Proving that $\mathcal{M}_i$ and any $\mathcal{M}' \in \mathcal{P}_i$ have the same adjacencies is straight-forward. We then use induction to the order of the triple to show that the two MAGs also share the same colliders with order. The following lemma proves that discriminating paths with order are present in all members of the equivalence class, and therefore they are (definite) discriminating paths with order in $\mathcal{P}_i$ (Lemma 18.) Thus, all (non) colliders with order in $\mathcal{P}_i$ are identified and added to the SAT formula in Lines 19 and 20 of Algorithm 4.

Lemma 18 *If $p = \langle W, V_1, \ldots, V_n, Y, Q\rangle$ is a discriminating path with order r in $\mathcal{M}$, then the path is a discriminating path with order r in $\mathcal{P} = [\mathcal{M}]$.*

Proof We will show that the path is a discriminating path with order r in any $\mathcal{M}' \in \mathcal{P}$. Since $\mathcal{M}'$ and $\mathcal{M}$ are Markov equivalent, the two share the same colliders with order. Thus, every triple $\langle V_{i-1}, V_i, V_{i+1}\rangle$ is a collider with order in $\mathcal{M}$. Lemma 3.10 in Ali et al. (2009) states that if a path $\langle W, V_1, \ldots, V_n, Y, Q\rangle$ is discriminating for Y in a MAG $\mathcal{M}$, then in any Markov equivalent MAG $\mathcal{M}'$ in which V_i are colliders on the same path, $V_i \rightarrow Q$ in $\mathcal{M}'$ for $i = 1, \ldots, N$, and therefore the path is discriminating with order r in $\mathcal{M}'$. Thus, the path is discriminating with order r in all members of $[\mathcal{M}]$. It is therefore a discriminating path

with order r in $\mathcal{P}$. ■

We can now prove that any truth-setting assignment for $\Phi \wedge \mathcal{F}$ corresponds to a SMCM $\mathcal{S}$ that is possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.

Lemma 19 *For an oracle of conditional independence, if $\Phi \wedge \mathcal{F}$ is the conjunction of the outputs of Algorithm 4, and $\mathcal{S}$ a mixed graph that satisfies $\Phi \wedge \mathcal{F}$, then $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*

Proof We need to prove that (a) $\mathcal{S}$ is an acyclic mixed graph and (b) $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} \in \mathcal{P}_i \quad \forall i$. To prove the latter, we need to prove that for each i, if $\mathcal{M}' \in \mathcal{P}_i$, $\mathcal{M}_i$ and $\mathcal{M}'$ are Markov equivalent. Thus, we must show that $\mathcal{M}_i$ and $\mathcal{M}'$ share the same edges and colliders with order.

- **$\mathcal{S}$ is a SMCM:** $\mathcal{S}$ satisfies the constraints added in Lines 8 and 9 respectively. Therefore, $\mathcal{S}$ has no tail-tail edges, every endpoint is an arrow or a tail (not exclusively) and $\mathcal{S}$ has no directed cycles.

- **$\mathcal{M}_i$ and $\mathcal{M}'$ share the same edges**: If X and Y are adjacent in $\mathcal{M}'$, then X and Y are adjacent in $\mathcal{P}_i$. $\mathcal{S}$ satisfies the constraints added in Line 4 of Algorithm 4, therefore there exists an inducing path with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$. Thus, X and Y are adjacent in $\mathcal{M}_i$. If X and Y are not adjacent in $\mathcal{M}'$, X and Y are not adjacent in $\mathcal{P}_i$ and 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 $\mathcal{M}'$ share the same colliders with order:** We will prove this by induction to order r: For order $= 0$, if $\langle X, Y, Z\rangle$ is an unshielded collider in $\mathcal{M}'$, the triple is an unshielded collider in $\mathcal{P}_i$. Since $\mathcal{M}'$ and $\mathcal{M}_i$ share the same edges, $X \star\!\!\rightarrow\!\!\star Y \star\!\!\leftarrow\!\!\star Z$ is an unshielded triple in $\mathcal{M}_i$. $\mathcal{S}$ satisfies the constraints added in Line 13 of Algorithm 4, and therefore Y is not an ancestor of X nor Z in $\mathcal{S}^{\mathbf{I}_i}$. Thus, $X \star\!\!\rightarrow Y \leftarrow\!\!\star Z$ in $\mathcal{M}_i$. If the triple is an unshielded collider in $\mathcal{M}_i$, then the triple is unshielded in $\mathcal{M}'$. If the triple is a non-collider in $\mathcal{M}'$, then $\mathcal{S}$ satisfies the constraints added in Line 14 of Algorithm 4, and Y is an ancestor of either X or Z in $\mathcal{S}^{\mathbf{I}_i}$. But then the triple is a non-collider in $\mathcal{M}_i$, which is a contradiction. Thus, $\mathcal{M}_i$ and $\mathcal{M}'$ share the same colliders with order 0.

  For the induction step, we assume that $\mathcal{M}_i$ and $\mathcal{M}'$ share the same colliders with order $s < r$. We will show that the two MAGs also share the same colliders with order r. We will first show that a path $\langle W, V_1, \ldots, V_n, Y, Q\rangle$ is discriminating for $\langle V_n, Y, Q\rangle$ with order r in $\mathcal{M}_i$ iff the path is discriminating for $\langle V_n, Y, Q\rangle$ with order r in $\mathcal{M}'$.

  If $\langle W, V_1, \ldots, V_n, Y, Q\rangle$ is discriminating with order r in $\mathcal{M}'$, by Lemma 18 the path is discriminating with order r in $\mathcal{P}_i$. $\mathcal{S}$ satisfies the constraints added in Lines 20 and 19 and therefore the path is discriminating in $\mathcal{M}_i$. Moreover, every triple on the path is a collider with order $< r$ in $\mathcal{M}'$ and by the induction hypothesis $\mathcal{M}'$ and $\mathcal{M}_i$ share the same colliders with order $< r$, thus the path has order r in $\mathcal{M}_i$.

  If $\langle W, V_1, \ldots, V_n, Y, Q\rangle$ is discriminating with order r in $\mathcal{M}_i$, then, by the induction hypothesis, every triple on the path is a collider with the same order $< r$ in $\mathcal{M}'$.

We will show that $V_i \to Q \quad \forall i$, and therefore $\langle W, V_1, \ldots, V_n, Y, Q\rangle$ is a discriminating path with order r in $\mathcal{M}'$.

The proof is similar to that of Lemma 3.10 in Ali et al. (2009). We will use induction on i. First, consider the (V_1, Q) edge in $\mathcal{M}'$. If $V_1 \leftarrow\!\star Q$, then $W \star\!\rightarrow V_1 \leftarrow\!\star Q$ forms a collider with order 0 in $\mathcal{M}'$, but an non-collider with order 0 in $\mathcal{M}_i$, which is a contradiction. Thus, $V_1 \rightarrow Q$ in $\mathcal{M}'$.

Suppose that $V_j \rightarrow Q$ for $1 \leq j \leq i$ in $\mathcal{M}'$. Then, the path $\langle W, V_1, \ldots, V_i, Q\rangle$ forms a discriminating path for V_i with the same order $< r$ in both graphs, and $\langle V_{i-1}, V_i, Q\rangle$ is a non-collider in $\mathcal{M}_i$. By Lemma 18, the path is a discriminating path with order in $\mathcal{P}_i$, and therefore $\Phi \wedge \mathcal{F}$ includes discriminating path constraints for this path added in Lines 19 and 21 or 20 and 22 of Algorithm 4. Thus, the triple can only be a non-collider in $\mathcal{M}_i$ if it is a non-collider in $\mathcal{M}'$. Since $V_{i-1} \leftrightarrow V_i$ in $\mathcal{M}'$, $V_i \rightarrow Q \quad \forall i$ and the path is discriminating in $\mathcal{M}'$ with order r.

We have shown that $\mathcal{M}_i$ and $\mathcal{M}'$ share the same discriminating paths with order r. It is now easy to show that a triple is a collider with order r in $\mathcal{M}'$ iff it is a collider with order r in $\mathcal{M}_i$. If $\langle V_n, Y, Z\rangle$ is a collider with order r in $\mathcal{M}'$, then there exists a discriminating path with order r in both graphs and in $\mathcal{P}_i$. Thus, $\mathcal{S}$ satisfies the constraints added in Lines 19 and 21 of Algorithm 4, by which Y is not an ancestor of V_n nor Q in $\mathcal{S}^{\mathbf{I}_i}$, and therefore the triple is a collider in $\mathcal{M}_i$, and it has order at most r. But by the induction hypothesis, the $\mathcal{M}'$ and $\mathcal{M}_i$ share the same colliders with order $< r$, thus the triple has order r in $\mathcal{M}_i$. Similarly, if the triple is a collider with order r in $\mathcal{M}_i$, there exists a discriminating path with order r in $\mathcal{M}$; and therefore in $\mathcal{P}_i$. Thus, $\mathcal{S}$ satisfies the constraints added in Lines 19 and 21 of Algorithm 4 or in Lines 20 and 22 of Algorithm 4. Hence, the triple must be in $\mathcal{M}'$, otherwise the triple would be a non-collider in $\mathcal{M}_i$. In addition, the triple has order at most r in $\mathcal{M}'$ and by the induction hypothesis the triple can not have order $< r$ in $\mathcal{M}'$, so the triple has order r in $\mathcal{M}'$. Thus, $\mathcal{M}'$ and $\mathcal{M}_i$ share the same colliders with order.

Thus, if $\mathcal{S}$ a mixed graph that satisfies $\Phi \wedge \mathcal{F}$, then $\mathcal{S}$ is a SMCM and $\text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[\mathbf{L}_i \in \mathcal{P}_i \quad \forall i$, so by Theorem 14, $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$. ■

We can now prove soundness and completeness of Algorithm 2:

Theorem 20 (Soundness and completeness of Algorithm 2) *If $\mathcal{H}$ is the output of Algorithm 2, then the following hold:*
***Soundness**: If a feature (edge, absent edge, endpoint) is* solid *in $\mathcal{H}$, then this feature is present in* all *SMCMs that are possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*
***Completeness**: If a feature is present in* all *SMCMs that are possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, the feature is solid in $\mathcal{H}$.*

Proof *Soundness:* Solid features correspond to backbone variables. By Lemma 17 every possibly underlying SMCM $\mathcal{S}$ for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ 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 possibly underlying SMCMs. *Completeness:* By Lemma 19 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 $\Phi \wedge \mathcal{F}$. ■

References

RA Ali, TS Richardson, and P Spirtes. Markov equivalence for ancestral graphs. *The Annals of Statistics*, 37(5B):2808–2837, 2009.

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*, 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 Tanner, 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*, 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 *Twenty-fourth Annual Conference on Neural Information Processing Systems*, 2010a.

T Claassen and T Heskes. Learning causal network structure from multiple (in) dependence models. In *Fifth European Workshop on Probabilistic Graphical Models*, 2010b.

T Claassen and T Heskes. A Bayesian approach to constraint based causal inference. In *Twenty-eighth Conference on Uncertainty in Artificial Intelligence*, 2012.

D Colombo, MH Maathuis, M Kalisch, and TS Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40 (1):294–321, 2012.

GF Cooper and Ch Yoo. Causal discovery from a mixture of experimental and observational data. In *Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999.

D Eaton and K Murphy. BDAGL: Bayesian DAG learning. `http://www.cs.ubc.ca/~murphyk/Software/BDAGL/`, 2007a.

D Eaton and KP Murphy. Exact bayesian structure learning from uncertain interventions. In *Eleventh International Conference on Artificial Intelligence and Statistics*, 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*, 2004.

RJ Evans and TS Richardson. Maximum likelihood fitting of acyclic directed mixed graphs to binary data. In *Twenty-sixth 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. *The Design of Experiments*. Hafner Publishing, New York, 1935.

D Geiger and D Heckerman. Learning Gaussian networks. In *Tenth Conference on Uncertainty in Artificial Intelligence*, 1994.

CP Gomes, B Selman, N Crato, and H 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. *Journal of Machine Learning Research*, 13 (1):2409–2464, 2012.

A Hyttinen, F Eberhardt, and PO Hoyer. Learning linear cyclic causal models with latent variables. *Journal of Machine Learning Research*, 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 *Twenty-eighth Conference on 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 *Twenty-ninth Conference on Uncertainty in Artificial Intelligence*, 2013.

S Itani, M Ohannessian, K Sachs, GP Nolan, and MA Dahleh. Structure learning in causal cyclic networks. In *Journal of Machine Learning Research Workshop and Conference Proceedings*, volume 6, pages 165 – 176, 2010.

A 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*, 2006.

JM Mooij and T Heskes. Cyclic causal discovery from continuous equilibrium data. In *Twenty-ninth Conference on Uncertainty in Artificial Intelligence*, 2013.

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 *Twenty-second Conference on Uncertainty in Artificial Intelligence*, 2006.

TS Richardson. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30(1):145–157, 2003.

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 *Twenty-eighth Conference on Uncertainty in Artificial Intelligence*. 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.

K Sadeghi. *Graphical Representation of Independence Structures*. PhD thesis, Oxford University, 2012.

T Sellke, MJ Bayarri, and JO Berger. Calibration of p-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 *Twenty-ninth Conference on Uncertainty in Artificial Intelligence*. 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 *Sixth International Workshop on Artificial Intelligence and Statistics*, 1996.

P Spirtes, C Glymour, and R Scheines. *Causation, Prediction, and Search*. The MIT Press, second edition, 2001.

JD Storey and R Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 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 *Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011.

RE Tillman, D Danks, and C Glymour. Integrating locally learned causal structures with overlapping variables. In *Twenty-Second Annual Conference on Neural Information Processing Systems*, 2008.

S Tong and D Koller. Active learning for structure in Bayesian networks. In *Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

S Triantafillou, I Tsamardinos, and IG Tollis. Learning causal structure from overlapping variable sets. In *Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

I Tsamardinos, S Triantafillou, and V Lagani. Towards integrative causal analysis of heterogeneous data sets and studies. *Journal of Machine Learning Research*, 13:1097–1157, 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, 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.