TECHNICAL REPORT
R-336
February 2008

Journal of Machine Learning Research 9 (2008) 1941-1979 Submitted 6/07; Revised 2/08; Published 9/08

Complete Identification Methods for the Causal Hierarchy

Ilya Shpitser ILYAS@CS.UCLA.EDU
Judea Pearl JUDEA@CS.UCLA.EDU
Cognitive Systems Laboratory
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA

Editor: Peter Spirtes

Abstract

We consider a hierarchy of queries about causal relationships in graphical models, where each level in the hierarchy requires more detailed information than the one below. The hierarchy consists of three levels: associative relationships, derived from a joint distribution over the observable variables; cause-effect relationships, derived from distributions resulting from external interventions; and counterfactuals, derived from distributions that span multiple "parallel worlds" and resulting from simultaneous, possibly conflicting observations and interventions. We completely characterize cases where a given causal query can be computed from information lower in the hierarchy, and provide algorithms that accomplish this computation. Specifically, we show when effects of interventions can be computed from observational studies, and when probabilities of counterfactuals can be computed from experimental studies. We also provide a graphical characterization of those queries which cannot be computed (by any method) from queries at a lower layer of the hierarchy.

Keywords: causality, graphical causal models, identification

1. Introduction

The human mind sees the world in terms of causes and effects. Understanding and mastering our environment hinges on answering questions about cause-effect relationships. In this paper we consider three distinct classes of causal questions forming a hierarchy.

The first class of questions involves associative relationships in domains with uncertainty, for example, "I took an aspirin after dinner, will I wake up with a headache?" The tools needed to formalize and answer such questions are the subject of probability theory and statistics, for they require computing or estimating some aspects of a joint probability distribution. In our aspirin example, this requires estimating the conditional probability $P(headache|aspirin)$ in a population that resembles the subject in question, that is, sharing age, sex, eating habits and any other traits that can be measured. Associational relationships, as is well known, are insufficient for establishing causation. We nevertheless place associative questions at the base of our causal hierarchy, because the probabilistic tools developed in studying such questions are instrumental for computing more informative causal queries, and serve therefore as an easily available starting point from which such computations can begin.

The second class of questions involves responses of outcomes of interest to outside interventions, for instance, "if I take an aspirin now, will I wake up with a headache?" Questions of this type are normally referred to as *causal effects*, sometimes written as $P(headache|do(aspirin))$. They

differ, of course from the associational counterpart
$P(headache|aspirin)$, because all mechanisms which normally determine aspirin taking behavior, for example, taste of aspirin, family advice, time pressure, etc. are irrelevant in evaluating the effect of a new decision.

To estimate effects, scientists normally perform randomized experiments where a sample of units drawn from the population of interest is subjected to the specified manipulation directly. In our aspirin example, this might involve treating a group of subjects with aspirin and comparing their response to untreated subjects, both groups being selected at random from a population resembling the decision maker in question. In many cases, however, such a direct approach is not possible due to expense or ethical considerations. Instead, investigators have to rely on observational studies to infer effects. A fundamental question in causal analysis is to determine when effects can be inferred from statistical information, encoded as a joint probability distribution, obtained under normal, intervention-free behavior. A key point here is that in order to make causal inferences from statistics, additional causal assumptions are needed. This is because without any assumptions it is possible to construct multiple "causal stories" which can disagree wildly on what effect a given intervention can have, but agree precisely on all observables. For instance, smoking may be highly correlated with lung cancer either because it causes lung cancer, or because people who are genetically predisposed to smoke may also have a gene responsible for a higher cancer incidence rate. In the latter case there will be no effect of smoking on cancer. Distinguishing between such causal stories requires additional, non-statistical language. In this paper, the language that we use for this purpose is the language of graphs, and our causal assumptions will be encoded by a special directed graph called a *causal diagram*.

The use of directed graphs to represent causality is a natural idea that arose multiple times independently: in genetics (Wright, 1921), econometrics (Haavelmo, 1943), and artificial intelligence (Pearl, 1988; Spirtes et al., 1993; Pearl, 2000). A causal diagram encodes variables of interest as nodes, and possible direct causal influences between two variables as arrows. Associated with each node in a causal diagram is a stable causal mechanism which determines its value in terms of the values of its parents. Unlike Bayesian networks (Pearl, 1988), the relationships between variables are assumed to be deterministic and uncertainty arises due to the presence of unobserved variables which have influence on our domain.

The first question we consider is under what conditions the effect of a given intervention can be computed from just the joint distribution over observable variables, which is obtainable by statistical means, and the causal diagram, which is either provided by a human expert, or inferred from experimental studies. This *identification problem* has received consideration attention in the statistics, epidemiology, and causal inference communities (Pearl, 1993a; Spirtes et al., 1993; Pearl and Robins, 1995; Pearl, 1995; Kuroki and Miyakawa, 1999; Pearl, 2000). In the subsequent sections, we solve the identification problem for causal effects by providing a graphical characterization for all non-identifiable effects, and an algorithm for computing all identifiable effects. Note that this identification problem actually involves two "worlds:" the original world where no interventions took place furnishes us with a probability distribution from which to make inferences about the second, post-intervention world. The crucial feature of causal effect queries which distinguishes them from more complex questions in our hierarchy is that they are restricted to the post-intervention world alone.

The third and final class of queries we consider are *counterfactual* or "what-if" questions which arise when we simultaneously ask about multiple hypothetical worlds, with potentially conflicting interventions or observations. An example of such a question would be “I took an aspirin, and my headache is gone; would I have had a headache had I not taken that aspirin?” Unlike questions involving interventions, counterfactuals contain conflicting information: in one world aspirin was taken, in another it was not. It is unclear therefore how to set up an effective experimental procedure for evaluating counterfactuals, let alone how to compute counterfactuals from observations alone. If everything about our causal domain is known, in other words if we have knowledge of both the causal mechanisms and the distributions over unobservable variables, it is possible to compute counterfactual questions directly (Balke and Pearl, 1994b). However, knowledge of precise causal mechanisms is not generally available, and the very nature of unobserved variables means their stochastic behavior cannot be estimated directly. We therefore consider the more practical question of how to compute counterfactual questions from both experimental studies and the structure of the causal diagram.

It may seem strange, in light of what we said earlier about the difficulty of conducting experimental studies, that we take such studies as given. It is nevertheless important that we understand when it is that “what-if” questions involving multiple worlds can be inferred from quantities computable in one world. Our hierarchical approach to identification allows us to cleanly separate difficulties that arise due to multiplicity of worlds from those involved in the identification of causal effects. We provide a complete solution to this version of the identification problem by giving algorithms which compute identifiable counterfactuals from experimental studies, and provide graphical conditions for the class of non-identifiable counterfactuals, where our algorithms fail. Our results can, of course, be combined to give conditions where counterfactuals can be computed from observational studies.

The paper is organized as follows. Section 2 introduces the notation and mathematical machinery needed for causal analysis. Section 3 considers the problem of identifying causal effects from observational studies. Section 4 considers identification of counterfactual queries, while Section 5 summarizes the conclusions. Most of the proofs are deferred to the appendix. This paper consolidates and expands previous results (Shpitser and Pearl, 2006a,b, 2007). Some of the results found in this paper were also derived independently elsewhere (Huang and Valtorta, 2006b,a).

2. Notation and Definitions

The primary object of causal inquiry is a probabilistic causal model. We will denote variables by uppercase letters, and their values by lowercase letters. Similarly, sets of variables will be denoted by bold uppercase, and sets of values by bold lowercase.

Definition 1 *A probabilistic causal model (PCM) is a tuple $M = \langle \mathbf{U}, \mathbf{V}, \mathbf{F}, P(\mathbf{u}) \rangle$, where*

- *$\mathbf{U}$ is a set of background or exogenous variables, which cannot be observed or experimented on, but which affect the rest of the model.*
- *$\mathbf{V}$ is a set $\{V_1, ..., V_n\}$ of observable or endogenous variables. These variables are functionally dependent on some subset of $\mathbf{U} \cup \mathbf{V}$.*
- *$\mathbf{F}$ is a set of functions $\{f_1, ..., f_n\}$ such that each f_i is a mapping from a subset of $\mathbf{U} \cup \mathbf{V} \setminus \{V_i\}$ to V_i, and such that $\bigcup \mathbf{F}$ is a function from $\mathbf{U}$ to $\mathbf{V}$.*
- *$P(\mathbf{u})$ is a joint probability distribution over $\mathbf{U}$.*

The set of functions $\mathbf{F}$ in this definition corresponds to the causal mechanisms, while $\mathbf{U}$ represents the background context that influences the observable domain of discourse $\mathbf{V}$, yet remains outside it. Our ignorance of the background context is represented by a distribution $P(\mathbf{u})$. This distribution, together with the mechanisms in $\mathbf{F}$, induces a distribution $P(\mathbf{v})$ over the observable domain. The causal diagram, our vehicle for expressing causal assumptions, is defined by the causal model as follows. Each observable variable $V_i \in \mathbf{V}$ corresponds to a vertex in the graph. Any two variables $V_i \in \mathbf{U} \cup \mathbf{V}$, $V_j \in \mathbf{V}$ such that V_i appears in the description of f_j are connected by a directed arrow from V_i to V_j. Furthermore, we make two additional assumptions in this paper. The first is that $P(\mathbf{u}) = \prod_{u_i \in \mathbf{u}} P(u_i)$, and each $U_i \in \mathbf{U}$ is used in at most two functions in F.[1] The second is that all induced graphs must be acyclic. Models in which these two assumptions hold are called recursive semi-Markovian. A graph defined as above from a causal model M is said to be a causal diagram *induced* by M. Graphs induced by semi-Markovian models are themselves called semi-Markovian. Fig. 1 and Fig. 2 show some examples of causal diagrams of recursive semi-Markovian models.

The functions in $\mathbf{F}$ are assumed to be *modular* in a sense that changes to one function do not affect any other. This assumption allows us to model how a PCM would react to changes imposed from the outside. The simplest change that is possible for causal mechanisms of a variable set $\mathbf{X}$ would be one that removes the mechanisms entirely and sets $\mathbf{X}$ to a specific value $\mathbf{x}$. This change, denoted by $do(\mathbf{x})$ (Pearl, 2000), is called an *intervention*. An intervention $do(\mathbf{x})$ applied to a model M results in a *submodel* $M_{\mathbf{x}}$. The effects of interventions will be formulated in several ways. For any given $\mathbf{u}$, the effect of $do(\mathbf{x})$ on a set of variables $\mathbf{Y}$ will be represented by *counterfactual variables* $Y_{\mathbf{x}}(\mathbf{u})$, where $Y \in \mathbf{Y}$. As $\mathbf{U}$ varies, the counterfactuals $Y_{\mathbf{x}}(\mathbf{u})$ will vary as well, and their *interventional distribution*, denoted by $P(\mathbf{y}|do(\mathbf{x}))$ or $P_{\mathbf{x}}(\mathbf{y})$ will be used to define the effect of $\mathbf{x}$ on $\mathbf{Y}$. We will denote the event "variable Y attains value y in $M_{\mathbf{x}}$" by the shorthand $y_{\mathbf{x}}$.

Interventional distributions are a mathematical formalization of an intuitive notion of effect of action. We now define joint probabilities on counterfactuals, in multiple worlds, which will serve as the formalization of counterfactual queries. Consider a conjunction of events $\gamma = y^1_{\mathbf{x}^1} \wedge \ldots \wedge y^k_{\mathbf{x}^k}$. If all the subscripts $\mathbf{x}^i$ are the same and equal to $\mathbf{x}$, γ is simply a set of assignments of values to variables in $M_{\mathbf{x}}$, and $P(\gamma) = P_{\mathbf{x}}(y^1, \ldots, y^k)$. However, if the actions $do(\mathbf{x}^i)$ are not the same, and potentially contradictory, a single submodel is no longer sufficient. Instead, γ is really invoking multiple causal worlds, each represented by a submodel $M_{\mathbf{x}^i}$. We assume each submodel shares the same set of exogenous variables $\mathbf{U}$, corresponding to the shared causal context or background history of the hypothetical worlds. Because the submodels are linked by common context, they can really be considered as one large causal model, with its own induced graph, and joint distribution over observable variables. $P(\gamma)$ can then be defined as a marginal distribution in this causal model. Formally, $P(\gamma) = \sum_{\{\mathbf{u}|\mathbf{u} \models \gamma\}} P(\mathbf{u})$, where $\mathbf{u} \models \gamma$ is taken to mean that each variable assignment in γ holds true in the corresponding submodel of M when the exogenous variables $\mathbf{U}$ assume values $\mathbf{u}$. In this way, $P(\mathbf{u})$ induces a distribution on all possible counterfactual variables in M. In this paper, we will represent counterfactual utterances by joint distributions such as $P(\gamma)$ or conditional distributions such as $P(\gamma|\delta)$, where γ and δ are conjunctions of counterfactual events. Pearl (2000) discusses counterfactuals, and their probabilistic representation used in this paper in greater depth.

1. Our results are generalizable to other $P(\mathbf{u})$ distributions which may not have such a simple form, but which can be represented by a set of bidirected arcs in such a way that whenever two sets of $\mathbf{U}$ variables are d-separated from each other, they are marginally independent. However, the exact conditions under which this graphical representation is valid are beyond the scope of this paper.

A fundamental question in causal inference is whether a given causal question, either interventional or counterfactual in nature, can be uniquely specified by the assumptions embodied in the causal diagram, and easily available information, usually statistical, associated with the causal model. To get a handle on this question, we introduce an important notion of *identifiability* (Pearl, 2000).

Definition 2 (identifiability) *Consider a class of models $\boldsymbol{M}$ with a description T, and objects ϕ and θ computable from each model. We say that ϕ is θ-identified in T if ϕ is uniquely computable from θ in any $M \in \boldsymbol{M}$. In this case all models in $\boldsymbol{M}$ which agree on θ will also agree on ϕ.*

If ϕ is θ-identifiable in T, we write $T, \theta \vdash_{id} \phi$. Otherwise, we write $T, \theta \nvdash_{id} \phi$. The above definition leads immediately to the following corollary which we will use to prove non-identifiability results.

Corollary 3 *Let T be a description of a class of models $\boldsymbol{M}$. Assume there exist $M^1, M^2 \in \boldsymbol{M}$ that share objects θ, while ϕ in M^1 is different from ϕ in M^2. Then $T, \theta \nvdash_{id} \phi$.*

In our context, the objects ϕ, θ are probability distributions derived from the PCM, where θ represents available information, while ϕ represents the quantity of interest. The description T is a specification of the properties shared all causal models under consideration, or, in other words, the set of assumptions we wish to impose on those models. Since we chose causal graphs as a language for specifying assumptions, T corresponds to a given graph.

Graphs earn their ubiquity as a specification language because they reflect in many ways the way people store experiential knowledge, especially cause-effect relationships. The ease with which people embrace graphical metaphors for causal and probabilistic notions—ancestry, neighborhood, flow, and so on—are proof of this affinity, and help ensure that the assumptions specified are meaningful and reliable. A consequence of this is that probabilistic dependencies among variables can be verified by checking if the flow of influence is blocked along paths linking the variables. By a path we mean a sequence of distinct nodes where each node is connected to the next in the sequence by an edge. The precise way in which the flow of dependence can be blocked is defined by the notion of d-separation (Pearl, 1986; Verma, 1986; Pearl, 1988). Here we generalize d-separation somewhat to account for the presence of bidirected arcs in causal diagrams.

Definition 4 (d-separation) *A path p in G is said to be d-separated by a set $\mathbf{Z}$ if and only if either*

1. *p contains one of the following three patterns of edges: $I \to M \to J$, $I \leftrightarrow M \to J$, or $I \leftarrow M \to J$, such that $M \in \mathbf{Z}$, or*

2. *p contains one of the following three patterns of edges: $I \to M \leftarrow J$, $I \leftrightarrow M \leftarrow J$, $I \leftrightarrow M \leftrightarrow J$, such that $De(M)_G \cap \mathbf{Z} = \emptyset$.*

Two sets $\mathbf{X}, \mathbf{Y}$ are said to be d-separated given $\mathbf{Z}$ in G if all paths from $\mathbf{X}$ to $\mathbf{Y}$ in G are d-separated by $\mathbf{Z}$. Paths or sets which are not d-separated are said to be d-connected. What allows us to connect this notion of blocking of paths in a causal diagram to the notion of probabilistic independence among variables is that the probability distribution over $\mathbf{V}$ and $\mathbf{U}$ in a causal model can be represented as a product of factors, such that each observable node has a factor corresponding

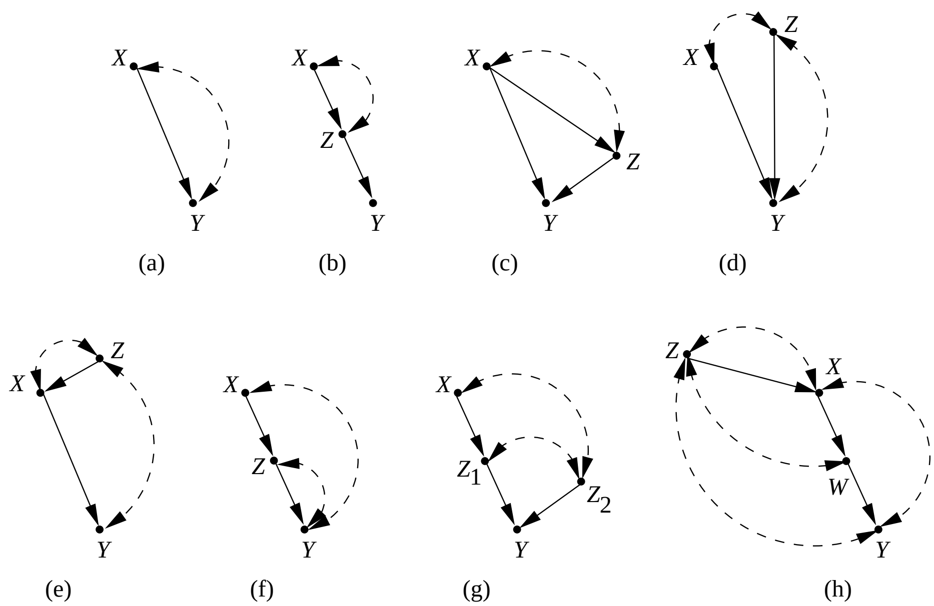

Figure 1: Causal graphs where $P(y|do(\mathbf{x}))$ is not identifiable

to its conditional distribution given the values of its parents in the graph. In other words, $P(\mathbf{v},\mathbf{u}) = \prod_i P(x_i|pa(x_i)_G)$.

Whenever the above factor decomposition holds for a distribution $P(\mathbf{v},\mathbf{u})$ and a graph G, we say G is an I-map of $P(\mathbf{v},\mathbf{u})$. The following theorem links d-separation of vertex sets in an I-map G with the independence of corresponding variable sets in P.

Theorem 5 *If sets $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G, then $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$ in every P for which G is an I-map. Furthermore, the causal diagram induced by any semi-Markovian PCM M is an I-map of the distribution $P(\mathbf{v},\mathbf{u})$ induced by M.*

Note that it's easy to rephrase the above theorem in terms of ordinary directed acyclic graphs, since each semi-Markovian graph is really an abbreviation where each bidirected arc stands for two directed arcs emanating from a hidden common cause. We will abbreviate this statement of d-separation, and corresponding independence by $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_G$, following the notation of Dawid (1979). For example in the graph shown in Fig. 6 (a), $X \not\!\perp\!\!\!\perp Y$ and $X \perp\!\!\!\perp Y|Z$, while in Fig. 6 (b), $X \perp\!\!\!\perp Y$ and $X \not\!\perp\!\!\!\perp Y|Z$.

Finally we consider the axioms and inference rules we will need. Since PCMs contain probability distributions, the inference rules we would use to compute queries in PCMs would certainly include the standard axioms of probability. They also include a set of axioms which govern the behavior of counterfactuals, such as Effectiveness, Composition, etc. (Galles and Pearl, 1998; Halpern, 2000; Pearl, 2000). However, in this paper, we will concentrate on a set of three identities applicable to interventional distributions known as do-calculus (Pearl, 1993b, 2000):

- Rule 1: $P_{\mathbf{x}}(\mathbf{y}|\mathbf{z},\mathbf{w}) = P_{\mathbf{x}}(\mathbf{y}|\mathbf{w})$ if $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X},\mathbf{W})_{G_{\overline{\mathbf{X}}}}$
- Rule 2: $P_{\mathbf{x},\mathbf{z}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{x}}(\mathbf{y}|\mathbf{z},\mathbf{w})$ if $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X},\mathbf{W})_{G_{\overline{\mathbf{X}},\underline{\mathbf{Z}}}}$

- Rule 3: $P_{\mathbf{X},\mathbf{Z}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$ if $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X},\mathbf{W})_{G_{\overline{x},\overline{z(\mathbf{W})}}}$

where $Z(\mathbf{W}) = \mathbf{Z} \setminus An(\mathbf{W})_{G_{\overline{\mathbf{X}}}}$, and $G_{\overline{\mathbf{X}},\underline{\mathbf{Y}}}$ stands for a directed graph obtained from G by removing all incoming arrows to $\mathbf{X}$ and all outgoing arrows from $\mathbf{Y}$. The rules of do-calculus provide a way of linking ordinary statistical distributions with distributions resulting from various manipulations.

In the remainder of this section we will introduce relevant graphs and graph-theoretic terminology which we will use in the rest of the paper. First, having defined causal diagrams induced by natural causal models, we consider the graphs induced by models derived from interventional and counterfactual queries. We note that in a given submodel $M_{\mathbf{X}}$, the mechanisms determining $\mathbf{X}$ no longer make use of the parents of $\mathbf{X}$ to determine their values, but instead set them independently to constant values $\mathbf{x}$. This means that the induced graph of $M_{\mathbf{X}}$ derived from a model M inducing graph G can be obtained from G by removing all arrows incoming to $\mathbf{X}$, in other words $M_{\mathbf{X}}$ induces $G_{\overline{\mathbf{X}}}$. A counterfactual $\gamma = y^1_{\mathbf{X}^1} \wedge ... \wedge y^k_{\mathbf{X}^k}$, as we already discussed invokes multiple hypothetical causal worlds, each represented by a submodel, where all worlds share the same background context $\mathbf{U}$. A naive way to graphically represent these worlds would be to consider all the graphs $G_{\overline{\mathbf{X}^i}}$ and have them share the $\mathbf{U}$ nodes. It turns out this representation suffers from certain problems. In Section 4 we discuss this issue in more detail and suggest a more appropriate graphical representation of counterfactual situations.

We denote $Pa(.)_G, Ch(.)_G, An(.)_G, De(.)_G$ as the sets of parents, children, ancestors, and descendants of a given set in G. We denote $G_{\mathbf{X}}$ to be the subgraph of G containing all vertices in $\mathbf{X}$, and edges between these vertices, while the set of vertices in a given graph G is given by $\mathbf{ver}(G)$. As a shorthand, we denote $G_{\mathbf{ver}(G)\setminus\mathbf{ver}(G')}$ as $G \setminus G'$ or $G \setminus \mathbf{X}$, if $\mathbf{X} = \mathbf{ver}(G')$, and G' is a subgraph of G. We will call the set $\{X \in G | De(X)_G = \emptyset\}$ the *root set* of G. A path connecting X and Y which begins with an arrow pointing to X is called a *back-door path* from X, while a path beginning with an arrow pointing away from X is called a *front-door path* from X.

The goal of this paper is a complete characterization of causal graphs which permit the answering of causal queries of a given type. This characterization requires the introduction of certain key graph structures.

Definition 6 (tree) *A graph G such that each vertex has at most one child, and only one vertex (called the root) has no children is called a tree.*

Note that this definition reverses the usual direction of arrows in trees as they are generally understood in graph theory. If we ignore bidirected arcs, graphs in Fig. 1 (a), (b), (d), (e), (f), (g), and (h) are trees.

Definition 7 (forest) *A graph G such that each vertex has at most one child is called a forest.*

Note that the above two definitions reverse the arrow directionality usual for these structures.

Definition 8 (confounded path) *A path where all directed arrowheads point at observable nodes, and never away from observable nodes is called a confounded path.*

The graph in Fig. 1 (g) contains a confounded path from Z_1 to Z_2.

Definition 9 (c-component) *A graph G where any pair of observable nodes is connected by a confounded path is called a c-component (confounded component).*

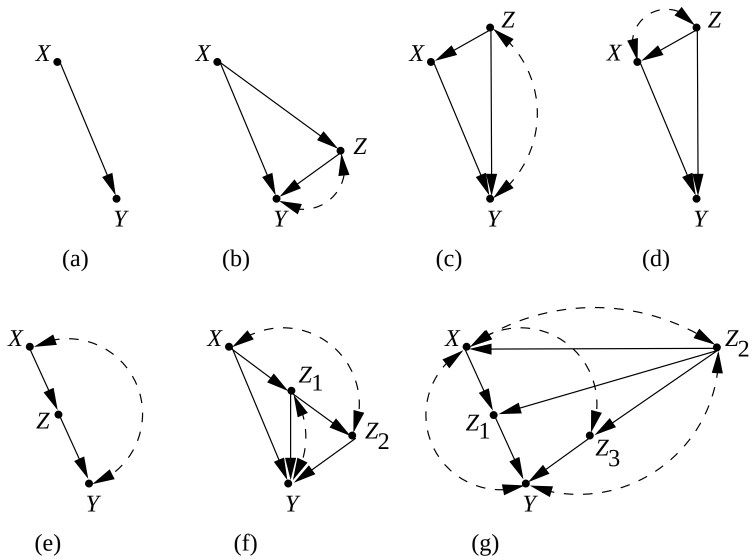

Figure 2: Causal graphs admitting identifiable effect $P(y|do(x))$

Graphs in Fig. 1 (a), (d), (e), (f), and (h) are c-components. Some graphs contain multiple c-components, for example the graph in Fig. 1 (b) has two maximal c-components: $\{Y\}$, and $\{X,Z\}$. We will denote the set of maximal c-components of a given graph G by $C(G)$. The importance of c-components stems from the fact that that the observational distribution $P(\mathbf{v})$ can be expressed as a product of factors $P_{\mathbf{v}\setminus\mathbf{s}}(\mathbf{s})$, where each $\mathbf{s}$ is a set of nodes forming a c-component. This important property is known as *c-component factorization*, and we will this property extensively in the remainder of the manuscript to decompose identification problems into smaller subproblems.

In the following sections, we will show how the graph structures we defined in this section are key for characterizing cases when $P_{\mathbf{x}}(\mathbf{y})$ and $P(\gamma)$ can be identified from available information.

3. Identification of Causal Effects

Like probabilistic dependence, the notion of causal effect of X on Y has an interpretation in terms of flow. Intuitively, X has an effect on Y if changing X causes Y to change. Since intervening on X cuts off X from the normal causal influences of its parents in the graph, we can interpret the causal effect of X on Y as the flow of dependence which leaves X via outgoing arrows only.

Recall that our ultimate goal is to express distributions of the form $P(\mathbf{y}|do(\mathbf{x}))$ in terms of the joint distribution $P(\mathbf{v})$. The interpretation of effect as downward dependence immediately suggests a set of graphs where this is possible. Specifically, whenever all d-connected paths from $\mathbf{X}$ to $\mathbf{Y}$ are front-door from $\mathbf{X}$, the causal effect $P(\mathbf{y}|do(\mathbf{x}))$ is equal to $P(\mathbf{y}|\mathbf{x})$. In graphs shown in Fig. 2 (a) and (b) causal effect $P(y|do(x))$ has this property.

In general, we don't expect acting on $\mathbf{X}$ to produce the same effect as observing $\mathbf{X}$ due to the presence of back-door paths between $\mathbf{X}$ and $\mathbf{Y}$. However, d-separation gives us a way to block undesirable paths by conditioning. If we can find a set $\mathbf{Z}$ that blocks all back-door paths from $\mathbf{X}$ to $\mathbf{Y}$, we obtain the following: $P(\mathbf{y}|do(\mathbf{x})) = \sum_{\mathbf{Z}} P(\mathbf{y}|\mathbf{z}, do(\mathbf{x}))P(\mathbf{z}|do(\mathbf{x}))$. The term $P(\mathbf{y}|\mathbf{z}, do(\mathbf{x}))$ is reduced to $P(\mathbf{y}|\mathbf{z},\mathbf{x})$ since the influence flow from $\mathbf{X}$ to $\mathbf{Y}$ is blocked by $\mathbf{Z}$. However, the act of adjusting for $\mathbf{Z}$ introduced a new effect we must compute, corresponding to the term $P(\mathbf{z}|do(\mathbf{x}))$. If it so happens that no variable in $\mathbf{Z}$ is a descendant of $\mathbf{X}$, we can reduce this term to $P(\mathbf{z})$ using the intuitive argument that acting on effects should not influence causes, or a more formal appeal to rule 3 of do-calculus. Computing effects in this way is always possible if we can find a set $\mathbf{Z}$ blocking all back-door paths which contains no descendants of $\mathbf{X}$. This is known as the *back-door criterion* (Pearl, 1993a, 2000). Figs. 2 (c) and (d) show some graphs where the node z satisfies the back-door criterion with respect to $P(y|do(x))$, which means $P(y|do(x))$ is identifiable.

The back-door criterion can fail—a common way involves a confounder that is unobserved, which prevents adjusting for it. Surprisingly, it is sometimes possible to identify the effect of $\mathbf{X}$ on $\mathbf{Y}$ even in the presence of such a confounder. To do so, we want to find a set $\mathbf{Z}$ located downstream of $\mathbf{X}$ but upstream of $\mathbf{Y}$, such that the downward flow of the effect of $\mathbf{X}$ on $\mathbf{Y}$ can be decomposed into the flow from $\mathbf{X}$ to $\mathbf{Z}$, and the flow from $\mathbf{Z}$ to $\mathbf{Y}$. Clearly, in order for this to happen $\mathbf{Z}$ must d-separate all front-door paths from $\mathbf{X}$ to $\mathbf{Y}$. However, in order to make sure that the component effects $P(\mathbf{z}|do(\mathbf{x}))$ and $P(\mathbf{y}|do(\mathbf{z}))$ are themselves identifiable, and combine appropriately to form $P(\mathbf{y}|do(\mathbf{x}))$, we need two additional assumptions: there are no back-door paths from $\mathbf{X}$ to $\mathbf{Z}$, and all back-door paths from $\mathbf{Z}$ to $\mathbf{Y}$ are blocked by $\mathbf{X}$. It turns out that these three conditions imply that $P(\mathbf{y}|do(\mathbf{x})) = \sum_{\mathbf{Z}} P(\mathbf{y}|do(\mathbf{z}))P(\mathbf{z}|do(\mathbf{x}))$, and the latter two conditions further imply that the first term is identifiable by the back-door criterion and equal to $\sum_{\mathbf{Z}} P(\mathbf{y}|\mathbf{z},\mathbf{x})P(\mathbf{x})$, while the second term is equal to $P(\mathbf{z}|\mathbf{x})$. Whenever these three conditions hold, the effect of $\mathbf{X}$ on $\mathbf{Y}$ is identifiable. This is known as the *front-door criterion* (Pearl, 1995, 2000). The front-door criterion holds in the graph shown in Fig. 2 (e).

Unfortunately, in some graphs neither the front-door, nor the back-door criterion holds. The simplest such graph, known as the bow arc graph due to its shape, is shown in Fig. 1 (a). The back-door criterion fails since the confounder node is unobservable, while the front-door criterion fails since no intermediate variables between X and Y exist in the graph. While the failure of these two criteria does not imply non-identification, in fact the effect $P(y|do(x))$ is identifiable in Fig. 2 (f), (g) despite this failure, a simple argument shows that $P(y|do(x))$ is not identifiable in the bow arc graph.

Theorem 10 *$P(\mathbf{v}), G \nvdash_{id} P(y|do(x))$ in G shown in Fig. 1 (a).*

Since we are interested in completely characterizing graphs where a given causal effect $P(\mathbf{y}|do(\mathbf{x}))$ is identifiable, it would be desirable to list difficult graphs like the bow arc graph which prevent identification of causal effects, in the hope of eventually making such a list complete and finding a way to identify effects in all graphs not on the list. We start constructing this list by considering graphs which generalize the bow arc graph since they can contain more than two nodes, but which also inherit its difficult structure. We call such graphs C-trees.

Definition 11 (C-tree) *A graph G which is both a C-component and a tree is called a C-tree.*

We call a C-tree with a root node Y Y-rooted. The graphs in Fig. 1 (a), (d), (e), (f), and (h) are Y-rooted C-trees. It turns out that in any Y-rooted C-tree, the effect of any subset of nodes, other than Y, on the root Y is not identifiable.

Theorem 12 *Let G be a Y-rooted C-tree. Let $\mathbf{X}$ be any subset of observable nodes in G which does not contain Y. Then $P(\mathbf{v}), G \nvdash_{id} P(y|do(\mathbf{x}))$.*

C-trees play a prominent role in the identification of *direct effects*. Intuitively, the direct effect of X on Y exists if there is an arrow from X to Y in the graph, and corresponds to the flow of influence along this arrow. However, simply considering changes in Y after fixing X is insufficient for isolating direct effect, since X can influence Y along other, longer front-door paths than the direct arrow. In order to disregard such influences, we also fix all other parents of Y (which as noted earlier removes all arrows incoming to these parents and thus to Y). The expression corresponding to the direct effect of X on Y is then $P(y|do(pa(y)))$. The following theorem links C-trees and direct effects.

Theorem 13 *$P(\mathbf{v}), G \not\vdash_{id} P(y|do(pa(y)))$ if and only if there exists a subgraph of G which is a Y-rooted C-tree.*

This theorem might suggest that C-trees might play an equally strong role in identifying arbitrary effects on a single variable, not just direct effects. Unfortunately, this turns out not to be the case, due to the following lemma.

Lemma 14 (downward extension lemma) *Let $\mathbf{V}$ be the set of observable nodes in G, and $P(\mathbf{v})$ the observable distribution of models inducing G. Assume $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{y}|do(\mathbf{x}))$. Let G' contain all the nodes and edges of G, and an additional node Z which is a child of all nodes in $\mathbf{Y}$. Then if $P(\mathbf{v}, z)$ is the observable distribution of models inducing G', then $P(\mathbf{v}, z), G' \not\vdash_{id} P(z|do(\mathbf{x}))$.*

Proof Let $|Z| = \prod_{Y_i \in \mathbf{Y}} |Y_i| = n$. By construction, $P(z|do(\mathbf{x})) = \sum_{\mathbf{y}} P(z|\mathbf{y})P(\mathbf{y}|do(\mathbf{x}))$. Due to the way we set the arity of Z, $P(Z|\mathbf{Y})$ is an n by n matrix which acts as a linear map which transforms $P(\mathbf{y}|do(\mathbf{x}))$ into $P(z|do(\mathbf{x}))$. Since we can arrange this linear map to be one to one, any proof of non-identifiability of $P(\mathbf{y}|do(\mathbf{x}))$ immediately extends to the proof of non-identifiability of $P(z|do(\mathbf{x}))$. ■

What this lemma shows is that identification of effects on a singleton is not any simpler than the general problem of identification of effect on a set. To find difficult graphs which prevent identification of effects on sets, we consider a multi-root generalization of C-trees.

Definition 15 (c-forest) *A graph G which is both a C-component and a forest is called a C-forest.*

If a given C-forest has a set of root nodes $\mathbf{R}$, we call it $\mathbf{R}$-rooted. Graphs in Fig. 3 (a), (b) are $\{Y1, Y2\}$-rooted C-forests. A naive way to generalize Theorem 12 would be to state that if G is an $\mathbf{R}$-rooted C-forest, then the effect of any set $\mathbf{X}$ that does not intersect $\mathbf{R}$ is not identifiable. However, as we later show, this is not true. Specifically, we later prove that $P(y1, y2|do(x))$ in the graph in Fig. 3 (a) is identifiable. To formulate the correct generalization of Theorem 12, we must understand what made C-trees difficult for the purposes of identifying effects on the root Y. It turned out that for particular function choices, the effects of ancestors of Y on Y precisely cancelled themselves out so even though Y itself was dependent on its parents, it was observationally indistinguishable from a constant function. To get the same canceling of effects with C-forests, we must define a more complex graphical structure.

Definition 16 (hedge) *Let $\mathbf{X}, \mathbf{Y}$ be sets of variables in G. Let F, F' be $\mathbf{R}$-rooted C-forests in G such that F' is a subgraph of F, $\mathbf{X}$ only occur in F, and $\mathbf{R} \in An(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$. Then F and F' form a hedge for $P(\mathbf{y}|do(\mathbf{x}))$.*

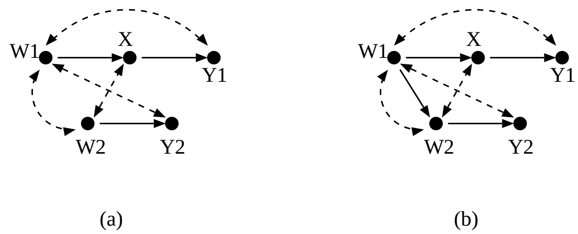

Figure 3: (a) A graph hedge-less for $P(y|do(x))$. (b) A graph containing a hedge for $P(y|do(x))$.

The graph in Fig. 3 (b) contains a hedge for $P(y1,y2|do(x))$. The mental picture for a hedge is as follows. We start with a C-forest F'. Then, F' grows new branches, while retaining the same root set, and becomes F. Finally, we "trim the hedge," by performing the action $do(\mathbf{x})$ which has the effect of removing some incoming arrows in $F \setminus F'$ (the subgraph of F consisting of vertices not a part of F'). Note that any Y-rooted C-tree and its root node Y form a hedge. The right generalization of Theorem 12 can be stated on hedges.

Theorem 17 *Let F, F' be subgraphs of G which form a hedge for $P(\mathbf{y}|do(\mathbf{x}))$. Then $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{y}|do(\mathbf{x}))$.*

Proof outline As before, assume binary variables. We let the causal mechanisms of one of the models consists entirely of bit parity functions. The second model also computes bit parity for every mechanism, except those nodes in F' which have parents in F ignore the values of those parents. It turns out that these two models are observationally indistinguishable. Furthermore, any intervention in $F \setminus F'$ will break the bit parity circuits of the models. This break will be felt at the root set $\mathbf{R}$ of the first model, but not of the second, by construction. ∎

Unlike the bow arc graph, and C-trees, hedges prevent identification of effects on multiple variables at once. Certainly a complete list of all possible difficult graphs must contain structures like hedges. But are there other kinds of structures that present problems? It turns out that the answer is "no," any time an effect is not identifiable in a causal model (if we make no restrictions on the type of function that can appear), there is a hedge structure involved. To prove that this is so, we need an algorithm which can identify any causal effect lacking a hedge. This algorithm, which we call **ID**, and which can be viewed as a simplified version of the identification algorithm due to Tian (2002), appears in Fig. 4.

We will explain why each line of **ID** makes sense, and conclude by showing the operation of the algorithm on an example. The formal proof of soundness of **ID** can be found in the appendix. The first line merely asserts that if no action has been taken, the effect on $\mathbf{Y}$ is just the marginal of the observational distribution $P(\mathbf{v})$ on $\mathbf{Y}$. The second line states that if we are interested in the effect on $\mathbf{Y}$, it is sufficient to restrict our attention on the parts of the model ancestral to $\mathbf{Y}$. One intuitive argument for this is that descendants of $\mathbf{Y}$ can be viewed as 'noisy versions' of $\mathbf{Y}$ and so any information they may impart which may be helpful for identification is already present in $\mathbf{Y}$. On the other hand, variables which are neither ancestors nor descendants of $\mathbf{Y}$ lie outside the relevant causal chain entirely, and have no useful information to contribute.

Line 3 forces an action on any node where such an action would have no effect on $\mathbf{Y}$—assuming we already acted on $\mathbf{X}$. Since actions remove incoming arrows, we can view line 3 as simplifying

function **ID**(**y**, **x**, P, G)
INPUT: **x**,**y** value assignments, P a probability distribution, G a causal diagram.
OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y})$ in terms of P or **FAIL**(F,F').

1 if $\mathbf{x} = \emptyset$ return $\sum_{\mathbf{v} \setminus \mathbf{y}} P(\mathbf{v})$.

2 if $\mathbf{V} \setminus An(\mathbf{Y})_G \neq \emptyset$
return $\mathbf{ID}(\mathbf{y}, \mathbf{x} \cap An(\mathbf{Y})_G, \sum_{\mathbf{V} \setminus An(\mathbf{Y})_G} P, G_{An(\mathbf{Y})})$.

3 let $\mathbf{W} = (\mathbf{V} \setminus \mathbf{X}) \setminus An(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$.
if $\mathbf{W} \neq \emptyset$, return $\mathbf{ID}(\mathbf{y}, \mathbf{x} \cup \mathbf{w}, P, G)$.

4 if $C(G \setminus \mathbf{X}) = \{S_1, ..., S_k\}$
return $\sum_{\mathbf{v} \setminus (\mathbf{y} \cup \mathbf{x})} \prod_i \mathbf{ID}(s_i, \mathbf{v} \setminus s_i, P, G)$.

if $C(G \setminus \mathbf{X}) = \{S\}$

5 if $C(G) = \{G\}$, throw $\mathbf{FAIL}(G, G \cap S)$.

6 if $S \in C(G)$ return $\sum_{s \setminus \mathbf{y}} \prod_{\{i | V_i \in S\}} P(v_i | v_\pi^{(i-1)})$.

7 if $(\exists S') S \subset S' \in C(G)$ return $\mathbf{ID}(\mathbf{y}, \mathbf{x} \cap S'$,
$\prod_{\{i | V_i \in S'\}} P(V_i | V_\pi^{(i-1)} \cap S', v_\pi^{(i-1)} \setminus S'), G_{S'})$.

Figure 4: A complete identification algorithm. **FAIL** propagates through recursive calls like an exception, and returns the hedge which witnesses non-identifiability. $V_\pi^{(i-1)}$ is the set of nodes preceding V_i in some topological ordering π in G.

the causal graph we consider by removing certain arcs from the graph, without affecting the overall answer. Line 4 is the key line of the algorithm, it decomposes the problem into a set of smaller problems using the key property of *c-component factorization* of causal models. If the entire graph is a single C-component already, further problem decomposition is impossible, and we must provide base cases. **ID** has three base cases. Line 5 fails because it finds two C-components, the graph G itself, and a subgraph S that does not contain any **X** nodes. But that is exactly one of the properties of C-forests that make up a hedge. In fact, it turns out that it is always possible to recover a hedge from these two c-components. Line 6 asserts that if there are no bidirected arcs from **X** to the other nodes in the current subproblem under consideration, then we can replace acting on **X** by conditioning, and thus solve the subproblem. Line 7 is the most complex case where **X** is partitioned into two sets, **W** which contain bidirected arcs into other nodes in the subproblem, and **Z** which do not. In this situation, identifying $P(\mathbf{y}|do(\mathbf{x}))$ from $P(\mathbf{v})$ is equivalent to identifying $P(\mathbf{y}|do(\mathbf{w}))$ from $P(\mathbf{V}|do(\mathbf{z}))$, since $P(\mathbf{y}|do(\mathbf{x})) = P(\mathbf{y}|do(\mathbf{w}), do(\mathbf{z}))$. But the term $P(\mathbf{V}|do(\mathbf{z}))$ is identifiable using the previous base case, so we can consider the subproblem of identifying $P(\mathbf{y}|do(\mathbf{w}))$.

We give an example of the operation of the algorithm by identifying $P_x(y_1, y_2)$ from $P(\mathbf{v})$ in the graph shown in in Fig. 3 (a). Since $G = G_{An(\{Y_1,Y_2\})}, C(G \setminus \{X\}) = \{G\}$, and $\mathbf{W} = \{W_1\}$, we invoke line 3 and attempt to identify $P_{x,w}(y_1, y_2)$. Now $C(G \setminus \{X, W\}) = \{Y_1, W_2 \to Y_2\}$, so we invoke line

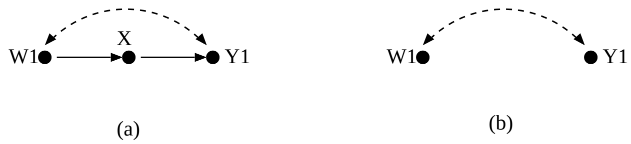

Figure 5: Subgraphs of G used for identifying $P_x(y_1, y_2)$.

4. Thus the original problem reduces to identifying $\sum_{w_2} P_{x,w_1,w_2,y_2}(y_1)P_{w,x,y_1}(w_2,y_2)$. Solving for the second expression, we trigger line 2, noting that we can ignore nodes which are not ancestors of W_2 and Y_2, which means $P_{w,x,y_1}(w_2,y_2) = P(w_2,y_2)$. Solving for the first expression, we first trigger line 2 also, obtaining $P_{x,w_1,w_2,y_2}(y_1) = P_{x,w}(y_1)$. The corresponding G is shown in Fig. 5 (a). Next, we trigger line 7, reducing the problem to computing $P_w(y_1)$ from $P(Y_1|X,W_1)P(W_1)$. The corresponding G is shown in Fig. 5 (b). Finally, we trigger line 2, obtaining $P_w(y_1) = \sum_{w_1} P(y_1|x,w_1)P(w_1)$. Putting everything together, we obtain: $P_x(y_1,y_2) = \sum_{w_2} P(y_1,w_2)\sum_{w_1} P(y_1|x,w_1)P(w_1)$.

As mentioned earlier, whenever the algorithm fails at line 5, it is possible to recover a hedge from the C-components S and G considered for the subproblem where the failure occurs. In fact, it can be shown that this hedge implies the non-identifiability of the original query with which the algorithm was invoked, which implies the following result.

Theorem 18 ***ID*** *is complete.*

The completeness of **ID** implies that hedges can be used to characterize all cases where effects of the form $P(\mathbf{y}|do(\mathbf{x}))$ cannot be identified from the observational distribution $P(\mathbf{v})$.

Theorem 19 (hedge criterion) *$P(\mathbf{v}), G \nvdash_{id} P(\mathbf{y}|do(\mathbf{x}))$ if and only if G contains a hedge for some $P(\mathbf{y}'|do(\mathbf{x}'))$, where $\mathbf{y}' \subseteq \mathbf{y}$, $\mathbf{x}' \subseteq \mathbf{x}$.*

We close this section by considering identification of *conditional effects* of the form $P(\mathbf{y}|do(\mathbf{x}),\mathbf{z})$ which are defined to be equal to $P(\mathbf{y},\mathbf{z}|do(\mathbf{x}))/P(\mathbf{z}|do(\mathbf{x}))$. Such expressions are a formalization of an intuitive notion of "effect of action in the presence of non-contradictory evidence," for instance the effect of smoking on lung cancer incidence rates in a particular age group (as opposed to the effect of smoking on cancer in the general population). We say that evidence $\mathbf{z}$ is non-contradictory since it is conceivable to consider questions where the evidence $\mathbf{z}$ stands in logical contradiction to the proposed hypothetical action $do(\mathbf{x})$: for instance what is the effect of smoking on cancer among the non-smokers. Such counterfactual questions will be considered in the next section. Conditioning can both help and hinder identifiability. $P(y|do(x))$ is not identifiable in the graph shown in Fig. 6 (a), while it is identifiable in the graph shown in Fig. 6 (b). Conditioning reverses the situation. In Fig. 6 (a), conditioning on Z renders Y independent of any changes to X, making $P_x(y|z)$ equal to $P(y|z)$. On the other hand, in Fig. 6 (b), conditioning on Z makes X and Y dependent, resulting in $P_x(y|z)$ becoming non-identifiable.

We would like to reduce the problem of identifying conditional effects to the familiar problem of identifying causal effects without evidence for which we already have a complete algorithm. Fortunately, rule 2 of do-calculus provides us with a convenient way of converting the unwanted evidence $\mathbf{z}$ into actions $do(\mathbf{x})$ which we know how to handle. The following convenient lemma allows us to remove as many evidence variables as possible from a conditional effect.

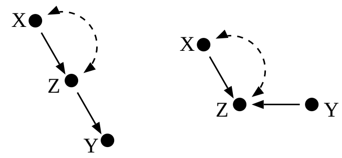

Figure 6: (a) Causal graph with an identifiable conditional effect $P(y|do(x),z)$. (b) Causal graph with a non-identifiable conditional effect $P(y|do(x),z)$.

function **IDC**(**y**, **x**, **z**, P, G)
INPUT: **x**,**y**,**z** value assignments, P a probability distribution, G a causal diagram (an I-map of P).
OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y}|\mathbf{z})$ in terms of P or **FAIL**(F,F').

1 if $(\exists Z \in \mathbf{Z})(\mathbf{Y} \perp\!\!\!\perp Z|\mathbf{X},\mathbf{Z}\setminus\{Z\})_{G_{\overline{\mathbf{X}},\underline{Z}}}$,
  return $\mathbf{IDC}(\mathbf{y},\mathbf{x}\cup\{z\},\mathbf{z}\setminus\{z\},P,G)$.

2 else let $P' = \mathbf{ID}(\mathbf{y}\cup\mathbf{z},\mathbf{x},P,G)$.
  return $P'/\sum_{\mathbf{y}} P'$.

Figure 7: A complete identification algorithm for conditional effects.

Theorem 20 *For any G and any conditional effect $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$ there exists a unique maximal set $\mathbf{Z} = \{Z \in \mathbf{W}|P_{\mathbf{X}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{X},z}(\mathbf{y}|\mathbf{w}\setminus\{z\})\}$ such that rule 2 applies to $\mathbf{Z}$ in G for $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$. In other words, $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{X},\mathbf{Z}}(\mathbf{y}|\mathbf{w}\setminus\mathbf{z})$.*

Of course Theorem 20 does not guarantee that the entire set **z** can be handled in this way. In many cases, even after rule 2 is applied, some set of evidence will remain in the expression. Fortunately, the following result implies that identification of unconditional causal effects is all we need.

Theorem 21 *Let $\mathbf{Z} \subseteq \mathbf{W}$ be the maximal set such that $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{X},\mathbf{Z}}(\mathbf{y}|\mathbf{w}\setminus\mathbf{z})$. Then $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$ is identifiable in G if and only if $P_{\mathbf{X},\mathbf{Z}}(\mathbf{y},\mathbf{w}\setminus\mathbf{z})$ is identifiable in G.*

The previous two theorems suggest a simple addition to **ID**, which we call **IDC**, shown in Fig. 7, which handles identification of conditional causal effects.

Theorem 22 ***IDC** is sound and complete.*

Proof This follows from Theorems 20 and 21. ■

We conclude this section by showing that our notion of a causal theory as a set of independencies embodied by the causal graph, together with rules of probability and do-calculus is complete for computing causal effects, if we also take statistical data embodied by $P(\mathbf{v})$ as axiomatic.

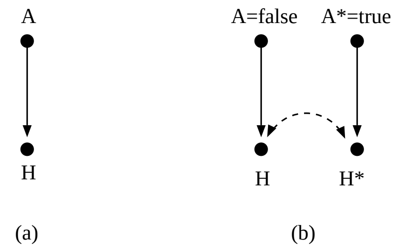

Figure 8: (a) A causal graph for the aspirin/headache domain (b) A corresponding twin network graph for the query $P(H^*_{a^*=true}|A = false)$.

Theorem 23 *The rules of do-calculus are complete for identifying effects of the form* $P(\boldsymbol{y}|do(\boldsymbol{x}), \boldsymbol{z})$, *where* $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ *are arbitrary sets.*

Proof The proofs of soundness of **ID** and **IDC** in the appendix use do-calculus. This implies every line of the algorithms we presented can be rephrased as a sequence of do-calculus manipulations. But **ID** and **IDC** are also complete, which implies the conclusion. ■

4. Identification of Counterfactuals

While effects of actions have an intuitive interpretation as downward flow, the interpretation of counterfactuals, or what-if questions is more complex. An informal counterfactual statement in natural language such as "would I have a headache had I taken an aspirin" talks about multiple worlds: the actual world, and other, hypothetical worlds which differ in some small respect from the actual world (e.g., the aspirin was taken), while in most other respects are the same. In this paper, we represent the actual world by a causal model in its natural state, devoid of any interventions, while the alternative worlds are represented by submodels $M_{\mathbf{x}}$ where the action $do(\mathbf{x})$ implements the hypothetical change from the actual state of affairs considered. People make sense of informal statements involving multiple, possibly conflicting worlds because they expect not only the causal rules to be invariant across these worlds (e.g., aspirin helps headaches in all worlds), but the worlds themselves to be similar enough where evidence in one world has ramifications in another. For instance, if we find ourselves with a headache, we expect the usual causes of our headache to also operate in the hypothetical world, interacting there with the preventative influence of aspirin. In our representation of counterfactuals, we model this interaction between worlds by assuming that the world histories or background contexts, represented by the unobserved **U** variables are shared across all hypothetical worlds.

We illustrate the representation method for counterfactuals we introduced in Section 2 by modeling our example question "would I have a headache had I taken an aspirin?" The actual world referenced by this query is represented by a causal model containing two variables, headache and aspirin, with aspirin being a parent of headache, see Fig. 8 (a). In this world, we observe that aspirin has value false. The hypothetical world is represented by a submodel where the action $do(aspirin = true)$ has been taken. To distinguish nodes in this world we augment their names with an asterisk. The two worlds share the background variables **U**, and so can be represented by a single causal model with the graph shown in Fig. 8 (b). Our query is represented by the distribution $P(H^*_{a^*=true}|A = false)$, where H is headache, and A is aspirin. Note that the nodes $A^* = true$ and $A = false$ in Fig. 8 (b) do not share a bidirected arc. This is because an intervention $do(a^* = true)$ removes all incoming arrows to A^*, which removes the bidirected arc between A^* and A.

The graphs representing two hypothetical worlds invoked by a counterfactual query like the one shown in Fig. 8 (b) are called *twin network graphs*, and were first proposed as a way to represent counterfactuals by Balke and Pearl (1994b) and Balke and Pearl (1994a). In addition, Balke and Pearl (1994b) proposed a method for evaluating expressions like $P(H^*_{a^*=true}|A = false)$ when all parameters of a causal model are known. This method can be explained as follows. If we forget the causal and counterfactual meaning behind the twin network graph, and simply view it as a Bayesian network, the query $P(H^*_{a^*=true}|A = false)$ can be evaluated using any of the standard inference algorithms available, provided we have access to all conditional probability tables generated by $\mathbf{F}$ and $\mathbf{U}$ of a causal model which gave rise to the twin network graph. In practice, however, complete knowledge of the model is too much to ask for; the functional relationships as well as the distribution $P(\mathbf{u})$ are not known exactly, though some of their aspects can be inferred from the observable distribution $P(\mathbf{v})$.

Instead, the typical state of knowledge of a causal domain is the statistical behavior of the observable variables in the domain, summarized by the distribution $P(\mathbf{v})$, together with knowledge of causal directionality, obtained either from expert judgment (e.g., we know that visiting the doctor does not make us sick, though disease and doctor visits are highly correlated), or direct experimentation (e.g., it's easy to imagine an experiment which establishes that wet grass does not cause sprinklers to turn on). We already used these two sources of knowledge in the previous section as a basis for computing causal effects. Nevertheless, there are reasons to consider computing counterfactual quantities from experimental, rather than observational studies. In general, a counterfactual can posit worlds with features contradictory to what has actually been observed. For instance, questions resembling the headache/aspirin question we used as an example are actually frequently asked in epidemiology in the more general form where we are interested in estimating the effect of a treatment x on the outcome variable Y for the patients that were not treated (x'). In our notation, this is just our familiar expression $P(Y_x|X = x')$. The problem with questions such as these is that no experimental setup exists in which someone is both given and not given treatment. Therefore, it makes sense to ask under what circumstances we can evaluate such questions even if we are given as input every experiment that is possible to perform in principle on a given causal model. In our framework the set of all experiments is denoted as P_*, and is formally defined as $\{P_{\mathbf{x}}|\mathbf{x}$ is any set of values of $\mathbf{X} \subseteq \mathbf{V}\}$. The question that we ask in this section, then, is whether it is possible to identify a query $P(\gamma|\delta)$, where γ, δ are conjunctions of counterfactual events (with δ possibly empty), from the graph G and the set of all experiments P_*. We can pose the problem in this way without loss of generality since we already developed complete methods for identifying members of P_* from G and $P(\mathbf{v})$. This means that if for some reason using P_* as input is not realistic we can combine the methods which we will develop in this section with those in the previous section to obtain identification results for $P(\gamma|\delta)$ from G and $P(\mathbf{v})$.

Before tackling the problem of identifying counterfactual queries from experiments, we extend our example in Fig. 8 (b) to a general graphical representation for worlds invoked by a counterfactual query. The twin network graph is a good first attempt at such a representation. It is essentially a causal diagram for a model encompassing two potential worlds. Nevertheless, the twin network graph suffers from a number of problems. Firstly, it can easily come to pass that a counterfactual

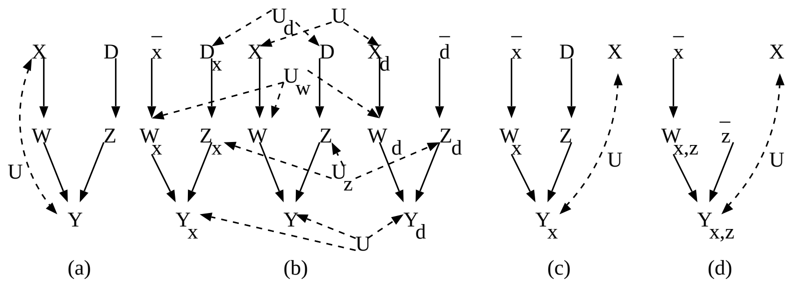

Figure 9: Nodes fixed by actions denoted with an overline, signifying that all incoming arrows are cut. (a) Original causal diagram (b) Parallel worlds graph for $P(y_x|x', z_d, d)$ (the two nodes denoted by U are the same). (c) Counterfactual graph for $P(y_x|x', z_d, d)$. (d) Counterfactual graph for $P(y_{x,z}|x')$.

query of interest would involve three or more worlds. For instance, we might be interested in how likely the patient would be to have a symptom Y given a certain dose x of drug X, assuming we know that the patient has taken dose x' of drug X, dose d of drug D, and we know how an intermediate symptom Z responds to treatment d. This would correspond to the query $P(y_x|x', z_d, d)$, which mentions three worlds, the original model M, and the submodels M_d, M_x. This problem is easy to tackle—we simply add more than two submodel graphs, and have them all share the same **U** nodes. This simple generalization of the twin network model was considered by Avin et al. (2005), and was called there the parallel worlds graph. Fig. 9 shows the original causal graph and the parallel worlds graph for $\gamma = y_x \wedge x' \wedge z_d \wedge d$.

The other problematic feature of the twin network graph, which is inherited by the parallel worlds graph, is that multiple nodes can sometimes correspond to the same random variable. For example, in Fig. 9 (b), the variables Z and Z_x are represented by distinct nodes, although it's easy to show that since Z is not a descendant of X, $Z = Z_x$. These equality constraints among nodes can make the d-separation criterion misleading if not used carefully. For instance, $Y_x \perp\!\!\!\perp D_x|Z$ even though using d-separation in the parallel worlds graph suggests the opposite. This sort of problem is fairly common in causal models which are not *faithful* (Spirtes et al., 1993) or *stable* (Pearl, 2000), in other words in models where d-separation statements in a causal diagram imply independence in a distribution, but not vice versa. However, lack of faithfulness usually arises due to "numeric coincidences" in the observable distribution. In this case, the lack of faithfulness is "structural," in a sense that it is possible to refine parallel worlds graphs in such a way that the node duplication disappears, and the attendant independencies not captured by d-separation are captured by d-separation in refined graphs.

This refinement has two additional beneficial side effects. The first is that by removing node duplication, we also determine which syntactically distinct counterfactual variables correspond to the same random variable. By identifying such equivalence classes of counterfactual variables, we guarantee that syntactically different variables are in fact different, and this makes it simpler to reason about counterfactuals in order to identify them. For instance, a counterfactual $P(y_x, y')$ may either be non-identifiable or inconsistent (and so identifiable to equal 0), depending on whether Y_x and Y are the same variable. The second benefit of this refinement is that resulting graphs are generally much smaller and less cluttered than parallel worlds graphs, and so are easier to understand. Compare, for instance, the graphs in Fig. 9 (b) and Fig. 9 (c). To rid ourselves of duplicates, we need a formal way of determining when variables from different submodels are in fact the same. The following lemma does this.

Lemma 24 *Let M be a model inducing G containing variables α, β with the following properties:*

- *α and β have the same domain of values.*
- *There is a bijection f from $Pa(\alpha)$ to $Pa(\beta)$ such that a parent γ and $f(\gamma)$ have the same domain of values.*
- *The functional mechanisms of α and β are the same (except whenever the function for α uses the parent γ, the corresponding function for β uses $f(\gamma)$).*

Assume an observable variable set $\mathbf{Z}$ was observed to attain values $\mathbf{z}$ in $M_{\mathbf{X}}$, the submodel obtained from M by forcing another observable variable set $\mathbf{X}$ to attain values $\mathbf{x}$. Assume further that for each $\gamma \in Pa(\alpha)$, either $f(\gamma) = \gamma$, or γ and $f(\gamma)$ attain the same values (whether by observation or intervention). Then α and β are the same random variable in $M_{\mathbf{X}}$ with observations $\mathbf{z}$.

Proof This follows from the fact that variables in a causal model are functionally determined from their parents. ■

If two distinct nodes in a causal diagram represent the same random variable, the diagram contains redundant information, and the nodes must be merged. If two nodes, say corresponding to $Y_{\mathbf{X}}, Y_{\mathbf{Z}}$, are established to be the same in G, they are merged into a single node which inherits all the children of the original two. These two nodes either share their parents (by induction) or their parents attain the same values. If a given parent is shared, it becomes the parent of the new node. Otherwise, we pick one of the parents arbitrarily to become the parent of the new node. This operation is summarized by the following lemma.

Lemma 25 *Let $M_{\mathbf{X}}$ be a submodel derived from M with set $\mathbf{Z}$ observed to attain values $\mathbf{z}$, such that Lemma 24 holds for α, β. Let M' be a causal model obtained from M by merging α, β into a new node ω, which inherits all parents and the functional mechanism of α. All children of α, β in M' become children of ω. Then $M_{\mathbf{X}}, M'_{\mathbf{X}}$ agree on any distribution consistent with $\mathbf{z}$ being observed.*

Proof This is a direct consequence of Lemma 24. ■

The new node ω we obtain from Lemma 25 can be thought of as a new counterfactual variable. As mentioned in section 2, such variables take the form $Y_{\mathbf{x}}$ where Y is the variable in the original causal model, and $\mathbf{x}$ is a subscript specifying the action which distinguishes the counterfactual. Since we only merge two variables derived from the same original, specifying Y is simple. But what about the subscript? Intuitively, the subscript of ω contains those fixed variables which are ancestors of ω in the graph G' of M'. Formally the subscript is $\mathbf{w}$, where $\mathbf{W} = An(\omega)_{G'} \cap \mathbf{sub}(\gamma)$, where the $\mathbf{sub}(\gamma)$ corresponds to those nodes in G' which correspond to subscripts in γ. Since we replaced α, β by ω, we replace any mention of α, β in our given counterfactual query $P(\gamma)$ by ω.

function **make-cg**(G,γ)
INPUT: G a causal diagram, γ a conjunction of counterfactual events
OUTPUT: A counterfactual graph G_γ, and either a set of events γ' s.t. $P(\gamma') = P(\gamma)$ or **INCONSISTENT**

- Construct a submodel graph $G_{\mathbf{x}_i}$ for each action $do(\mathbf{x}_i)$ mentioned in γ. Construct the parallel worlds graph G' by having all such submodel graphs share their corresponding U nodes.
- Let π be a topological ordering of nodes in G', let $\gamma' := \gamma$.
- Apply Lemmas 24 and 25, in order π, to each observable node pair α,β derived from the same variable in G. For each α,β that are the same, do:
  - Let G' be modified as specified in Lemma 25.
  - Modify γ' by renaming all occurrences of β to α.
  - If $\mathbf{val}(\alpha) \neq \mathbf{val}(\beta)$, return G', **INCONSISTENT**.
- return $(G'_{An(\gamma')},\gamma')$, where $An(\gamma')$ is the set of nodes in G' ancestral to nodes corresponding to variables mentioned in γ'.

Figure 10: An algorithm for constructing counterfactual graphs

Note that since α,β are the *same*, their value assignments must be the same (say equal to y). The new counterfactual ω inherits this assignment.

We summarize the inductive applications of Lemma 24, and 25 by the **make-cg** algorithm, which takes γ and G as arguments, and constructs a version of the parallel worlds graph without duplicate nodes. We call the resulting structure the *counterfactual graph* of γ, and denote it by G_γ. The algorithm is shown in Fig. 10.

There are three additional subtleties in **make-cg**. The first is that if variables $Y_{\mathbf{X}},Y_{\mathbf{Z}}$ were judged to be the same by Lemma 24, but γ assigns them different values, this implies that the original set of counterfactual events γ is inconsistent, and so $P(\gamma)=0$. The second is that if we are interested in identifiability of $P(\gamma)$, we can restrict ourselves to the ancestors of γ in G'. We can justify this using the same intuitive argument we used in Section 3 to justify Line 2 in **ID**. The formal proof for line 2 we provide in the appendix applies with little change to **make-cg**. Finally, because the algorithm can make an arbitrary choice picking a parent of ω each time Lemma 25 is applied, both the counterfactual graph G', and the corresponding modified counterfactual γ' are not unique. This does not present a problem, however, as any such graph is acceptable for our purposes.

We illustrate the operation of **make-cg** by showing how the graph in Fig. 9 (c) is derived from the graph in Fig. 9 (b). We start the application of Lemma 24 from the topmost observable nodes, and conclude that the node pairs D_x,D, and X_d,X have the same functional mechanisms, and the same parent set (in this case the parents are unobservable nodes U_d for the first pair, and U for the second). We then use Lemma 25 to obtain the graph shown in Fig. 11 (a). Since the node pairs are the same, we pick the name of one of the nodes of the pair to serve as the name of the new node. In our case, we picked D and X. Note that for this graph, and all subsequent intermediate graphs we generate, we use the convention that if a merge creates a situation where an unobservable variable

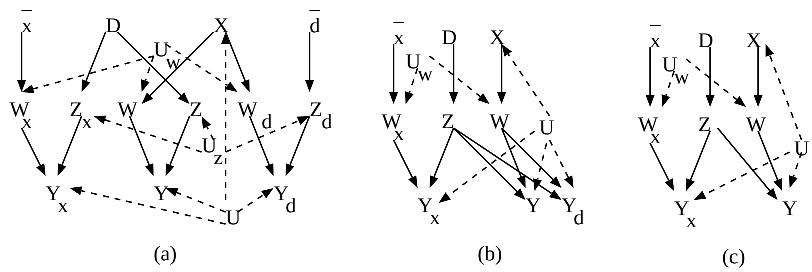

Figure 11: Intermediate graphs obtained by **make-cg** in constructing the counterfactual graph for $P(y_x|x',z_d,d)$ from Fig. 9 (b).

has a single child, that variable is omitted from the graph. For instance, in Fig. 11 (a), the variable U_d, and its corresponding arrow to D omitted.

Next, we apply Lemma 24 for the node pair W_d,W. In this case, the functional mechanisms are once again the same, while the parents of W_d,W are X and U_w. We can also apply Lemma 24 twice to conclude that Z,Z_x and Z_d are in fact the same node, and so can be merged. The functional mechanisms of these three nodes are the same, and they share the parent U_z. As far as the parents of this triplet, the U_z parent is shared by all three, while Z,Z_x share the parent D, and Z_d has a separate parent d, fixed by intervention. However, in our counterfactual query, which is $P(y_x|x',z_d,d)$, the variable D happens to be observed to attain the value d, the same as the intervention value for the parent of Z_d. This implies that for the purposes of the Z,Z_x,Z_d triplet, their D-derived parents share the same value, which allows us to conclude they are the same random variable. The intuition here is that while intervention and observation are not the same operation, they have the same effect if the relevant U variables happen to react in the same way to both the given intervention, and the given observation (this is the essence of the Axiom of Composition discussed by Pearl (2000).) In our case, U variables react the same way because the parallel worlds share all unobserved variables.

There is one additional subtlety in performing the merge of the triplet Z,Z_x,Z_d. If we examine our query $P(y_x|x',z_d,d)$, we notice that Z_d, or more precisely its value, appears in it. When we merge nodes, we only use one name out of the original two. It's possible that some of the old names appear in the query, which means we must replace all references to the old, pre-merge nodes with the new post-merge name we picked. Since we picked the name Z for the newly merged node, we replace the reference to Z_d in our query by the reference to Z, so our modified query is $P(y_x|x',z,d)$. Since the variables were established to be the same, this is a safe syntactic transformation.

After W_d,W, and the Z,Z_x,Z_d triplet are merged, we obtain the graph in Fig. 11 (b). Finally, we apply Lemma 24 one more time to conclude Y and Y_d are the same variable, using the same reasoning as before. After performing this final merge, we obtain the graph in Fig. 11 (c). It's easy to see that Lemma 24 no longer applies to any node pair: W and W_x differ in their X-derived parent, and Y, and Y_x differ on their W-derived parent, which was established inductively. The final operation which **make-cg** performs is restricting the graph in Fig. 11 (b) to variables actually relevant for computing the (potentially syntactically modified) query it was given as input, namely $P(y_x|x',z,d)$. These relevant variables are ancestral to variables in the query in the final intermediate graph we

function $\mathbf{ID^*}(G,\gamma)$
INPUT: G a causal diagram, γ a conjunction of counterfactual events
OUTPUT: an expression for $P(\gamma)$ in terms of P_* or **FAIL**

1 if $\gamma = \emptyset$, return 1

2 if $(\exists x_{x'..} \in \gamma)$, return 0

3 if $(\exists x_{x..} \in \gamma)$, return $\mathbf{ID^*}(G, \gamma \setminus \{x_{x..}\})$

4 $(G', \gamma') = \mathbf{make\text{-}cg}(G, \gamma)$

5 if $\gamma' = \mathbf{INCONSISTENT}$, return 0

6 if $C(G') = \{S^1, \ldots, S^k\}$,
return $\sum_{V(G') \setminus \gamma'} \prod_i \mathbf{ID^*}(G, s^i_{v(G') \setminus s^i})$

7 if $C(G') = \{S\}$ then,

  8 if $(\exists \mathbf{x}, \mathbf{x}')$ s.t. $\mathbf{x} \neq \mathbf{x}', \mathbf{x} \in \mathbf{sub}(S), \mathbf{x}' \in \mathbf{ev}(S)$,
  throw **FAIL**

  9 else, let $\mathbf{x} = \bigcup \mathbf{sub}(S)$
  return $P_{\mathbf{x}}(\mathbf{var}(S))$

function $\mathbf{IDC^*}(G,\gamma,\delta)$
INPUT: G a causal diagram, γ, δ conjunctions of counterfactual events
OUTPUT: an expression for $P(\gamma|\delta)$ in terms of P_*, **FAIL**, or **UNDEFINED**

1 if $\mathbf{ID^*}(G, \delta) = 0$, return **UNDEFINED**

2 $(G', \gamma' \wedge \delta') = \mathbf{make\text{-}cg}(G, \gamma \wedge \delta)$

3 if $\gamma' \wedge \delta' = \mathbf{INCONSISTENT}$, return 0

4 if $(\exists y_{\mathbf{x}} \in \delta')$ s.t. $(Y_{\mathbf{x}} \perp\!\!\!\perp \gamma')_{G'_{\underline{y_{\mathbf{x}}}}}$,
return $\mathbf{IDC^*}(G, \gamma'_{y_{\mathbf{x}}}, \delta' \setminus \{y_{\mathbf{x}}\})$

5 else, let $P' = \mathbf{ID^*}(G, \gamma' \wedge \delta')$. return $P'/P'(\delta)$

Figure 12: Counterfactual identification algorithms.

obtained. In our case, we remove nodes W and Y (and their adjacent edges) from consideration, to finally obtain the graph in Fig. 9 (c), which is a counterfactual graph for our query.

Having constructed a graphical representation of worlds mentioned in counterfactual queries, we can turn to identification. We construct two algorithms for this task, the first is called **ID*** and works for unconditional queries, while the second, **IDC***, works on queries with counterfactual evidence and calls the first as a subroutine. These are shown in Fig. 12.

These algorithms make use of the following notation: **sub**$(.)$ returns the set of subscripts, **var**$(.)$ the set of variables, and **ev**$(.)$ the set of values (either set or observed) appearing in a given counterfactual conjunction (or set of counterfactual events), while **val**$(.)$ is the value assigned to a given counterfactual variable. This notation is used to extract variables and values present in the original causal model from a counterfactual which refers to parallel worlds. As before, $C(G')$ is the set of maximal C-components of G', except we don't count nodes in G' fixed by interventions as part of any C-component. $V(G')$ is the set of observable nodes of G' not fixed by interventions. Following Pearl (2000), $G'_{\underline{y_{\mathbf{x}}}}$ is the graph obtained from G' by removing all outgoing arcs from $Y_{\mathbf{x}}$; $\gamma'_{y_{\mathbf{x}}}$ is obtained from γ' by replacing all descendant variables $W_{\mathbf{z}}$ of $Y_{\mathbf{x}}$ in γ' by $W_{\mathbf{z},y}$. A counterfactual $\mathbf{s_r}$, where $\mathbf{s},\mathbf{r}$ are value assignments to sets of nodes, represents the event "the node set $\mathbf{S}$ attains values $\mathbf{s}$ under intervention $do(\mathbf{r})$." For instance, the term $s^i_{v(g')\setminus s^i}$ stands for the event "the node set S^i attains values s^i under the intervention $do(v(g')\setminus s^i)$," in other words under the intervention where we fix the values of all observable nodes in G' except those in S^i. Finally, we take $x_{x..}$ to mean some counterfactual variable derived from X where x appears in the subscript (the rest of the subscript can be arbitrary), which also attains value x.

The notation used in these algorithms is somewhat intricate, so we give an intuitive description of each line. We start with **ID***. The first line states that if γ is an empty conjunction, then its probability is 1, by convention. The second line states that if γ contains a counterfactual which violates the Axiom of Effectiveness (Pearl, 2000), then γ is inconsistent, and we return probability 0. The third line states that if a counterfactual contains its own value in the subscript, then it is a tautological event, and it can be removed from γ without affecting its probability. Line 4 invokes **make-cg** to construct a counterfactual graph G', and the corresponding relabeled counterfactual γ'. Line 5 returns probability 0 if an inconsistency was found during the construction of the counterfactual graph, for example, if two variables found to be the same in γ had different value assignments. Line 6 is analogous to Line 4 in the **ID** algorithm, it decomposes the problem into a set of subproblems, one for each C-component in the counterfactual graph. In the **ID** algorithm, the term corresponding to a given C-component S_i of the causal diagram was the effect of all variables not in S_i on variables in S_i, in other words $P_{\mathbf{v}\setminus s_i}(s_i)$, and the outermost summation on line 4 was over values of variables not in $\mathbf{Y},\mathbf{X}$. Here, the term corresponding to a given C-component S^i of the counterfactual graph G' is the conjunction of counterfactual variables where each variable contains in its subscript all variables not in the C-component S^i, in other words $\mathbf{v}(G')\setminus s^i$, and the outermost summation is over observable variables not in γ', that is over $\mathbf{v}(G')\setminus\gamma'$, where we interpret γ' as a set of counterfactuals, rather than a conjunction. Line 7 is the base case, where our counterfactual graph has a single C-component. There are two cases, corresponding to line 8 and line 9. Line 8 says that if γ' contains a "conflict," that is an inconsistent value assignment where at least one value is in the subscript, then we fail. Line 9 says if there are no conflicts, then its safe to take the union of all subscripts in γ', and return the effect of the subscripts in γ' on the variables in γ'.

The **IDC***, like its counterpart **IDC**, is shorter. The first line fails if δ is inconsistent. **IDC** did not have an equivalent line, since we can assume $P(\mathbf{v})$ is positive. The problem with counterfactual distributions is there is no simple way to prevent non-positive distributions spanning multiple worlds from arising, even if the original $P(\mathbf{v})$ was positive—hence the explicit check. The second line constructs the counterfactual graph, except since **make-cg** can only take conjunctions, we provide it with a joint counterfactual $\gamma\wedge\delta$. Line 3 returns 0 if an inconsistency was detected. Line 4 of **IDC*** is the central line of the algorithm and is analogous to line 1 of **IDC**. In **IDC**, we moved a value assignment $Z = z$ from being observed to being fixed if there were no back-door paths from Z to the outcome variables $\mathbf{Y}$ given the context of the effect of $do(\mathbf{x})$. Here in **IDC***, we move a counterfactual value assignment $Y_{\mathbf{x}} = y$ from being observed (that is being a part of δ), to being fixed (that is appearing in every subscript of γ') if there are no back-door paths from $Y_{\mathbf{x}}$ to the counterfactual of interest γ'. Finally, line 5 of **IDC*** is the analogue of line 2 of **IDC**, we attempt to identify a joint counterfactual probability, and then obtain a conditional counterfactual probability from the result.

We illustrate the operation of these algorithms by considering the identification of a query $P(y_x|x', z_d, d)$ we mentioned earlier. Since $P(x', z_d, d)$ is not inconsistent, we proceed to construct the counterfactual graph on line 2. Suppose we produce the graph in Fig. 9 (c), where the corresponding modified query is $P(y_x|x', z, d)$. Since $P(y_x, x', z, d)$ is not inconsistent we proceed to the next line, which moves z, d (with d being redundant due to graph structure) to the subscript of y_x, to obtain $P(y_{x,z}|x')$, and calls **IDC*** with this query recursively. Note that since the subscripts in one of the variables of our query changed, the counterfactual graph generated will change as well. In particular, the invocation of **make-cg** with the joint distribution from which $P(y_{x,z}|x')$ is derived, namely $P(y_{x,z}, x')$, will result in the graph shown in Fig. 9 (d). Since X' has a back-door path to $Y_{x,z}$ in this graph, we can no longer call **IDC*** recursively, so we invoke **ID*** with the query $P(y_{x,z}, x')$.

The first interesting line in **ID*** is line 6, which first computes $P(y_{x,z}, w_{x,z}, x')$ by C-component factorization, and then computes $P(y_{x,z}, x')$ from $P(y_{x,z}, w_{x,z}, x')$ by marginalizing over $W_{x,z}$.[2] Since the counterfactual graph for this query (Fig. 9 (d)) has two C-components, $\{Y_{x,z}, X\}$ and $\{W_{x,z}\}$, $P(y_{x,z}, w_{x,z}, x') = P(y_{x,z,w}, x'_w)P(w_{x,z})$, which can be simplified by removing redundant subscripts to $P(y_{z,w}, x')P(w_x)$. Line 6 then recursively calls **ID*** with $P(y_{x,z,w}, x')$ and $P(w_x)$, multiplies the results and marginalizes over W_x. The first recursive call reaches line 9 with $P(y_{z,w}, x')$, which is identifiable as $P_{z,w}(y, x')$ from P_*. The second term is trivially identifiable as $P_x(w)$, which means our query is identifiable as $P' = \sum_w P_{z,w}(y, x')P_x(w)$, and the conditional query is equal to $P'/P'(x')$.

The definitions of **ID***, and **IDC*** reveal their close similarity to algorithms **ID** and **IDC** in the previous section. The major differences lie in the failure and success base cases, and slightly different subscript notation. This is not a coincidence, since a counterfactual graph can be thought of as a causal graph for a particular large causal model which happens to have some distinct nodes share the same causal mechanisms. This means that all the theorems and definitions used in the previous sections for causal diagrams transfer over without change to counterfactual graphs. Using this fact, we will show that **ID***, and **IDC*** are sound and complete for identifying $P(\gamma)$, and $P(\gamma|\delta)$ respectively.

Theorem 26 (soundness) *If **ID*** succeeds, the expression it returns is equal to $P(\gamma)$ in a given causal graph. Furthermore, if **IDC*** does not output **FAIL**, the expression it returns is equal to $P(\gamma|\delta)$ in a given causal graph, if that expression is defined, and **UNDEFINED** otherwise.*

Proof outline The first line merely states that the probability of an empty conjunction is 1, which is true by convention. Lines 2 and 3 follow by the Axiom of Effectiveness (Galles and Pearl, 1998). The soundness of **make-cg** has already been established, which implies the soundness of line 4. Line 6 decomposes the problem using c-component factorization. The soundness proof for this decomposition, also used in the previous section, is in the appendix. Line 9 asserts that if a set

2. Note that since $W_{x,z}$ is a counterfactual variable derived from W, it shares its domain with W. Therefore it makes sense when marginalizing to operate over the values of W, denoted by w in the subscript of the summation.

of counterfactual events does not contain conflicting value assignments to any variable, obtained either by observation or intervention, then taking the union of all actions of the events results in a consistent action. The probability of the set of events can then be computed from a submodel where this consistent action has taken place. A full proof of this is in the appendix. ■

To show completeness, we follow the same strategy we used in the previous section. We catalogue all difficult counterfactual graphs which arise from queries which cannot be identified from P_*. We then show these graphs arise whenever **ID*** and **IDC*** fail. This, together with the soundness theorem we already proved, implies that these algorithms are complete.

The simplest difficult counterfactual graph arises from the query $P(y_x, y'_{x'})$ named "probability of necessity and sufficiency" by Pearl (2000). This graph, shown in Fig. 8 (b) with variable relabeling, is called the "w-graph" due to its shape (Avin et al., 2005). This query is so named because if $P(y_x, y'_{x'})$ is high, this implies that if the variable X is forced to x, variable Y is likely to be y, while if X is forced to some other value, Y is likely to not be y. This means that the action $do(x)$ is likely a necessary and sufficient cause of Y assuming value y, up to noise. The w-graph starts our catalogue of bad graphs with good reason, as the following lemma shows.

Lemma 27 *Assume X is a parent of Y in G. Then $P_*, G \not\vdash_{id} P(y_x, y'_{x'}), P(y_x, y')$ for any value pair y, y'.*

Proof See Avin et al. (2005). ■

The intuitive explanation for this result is that $P(y_x, y'_{x'})$ is derived from the joint distribution over the counterfactual variables in the w-graph, while if we restrict ourselves to P_*, we only have access to marginal distributions—one marginal for each possible world. Because counterfactual variables Y_x and $Y_{x'}$ share an unobserved parent U, they are dependent, and their joint distribution cannot be decomposed into a product of marginals. This means that the information encoded in the marginals is insufficient to uniquely determine the joint we are interested in. This intuitive argument can be generalized to a counterfactual graph with more than two nodes, the so-called "zig-zag graphs" an example of which is shown in Fig. 13 (b).

Lemma 28 *Assume G is such that X is a parent of Y and Z, and Y and Z are connected by a bidirected path with observable nodes $W^1, \ldots, W^k$ on the path. Then $P_*, G \not\vdash_{id} P(y_x, w^1, \ldots, w^k, z_{x'})$, $P(y_x, w^1, \ldots, w^k, z)$ for any value assignments $y, w^1, \ldots, w^k, z$.*

The w-graph in Fig. 8 (b) and the zig-zag graph in Fig. 13 (b) have very special structure, so we don't expect our characterization to be complete with just these graphs. In order to continue, we must provide two lemmas which allow us to transform difficult graphs in various ways by adding nodes and edges, while retaining the non-identifiability of the underlying counterfactual from P_*.

Lemma 29 (downward extension lemma) *Assume $P_*, G \not\vdash_{id} P(\gamma)$. Let $\{y^1_{\mathbf{x}^1}, \ldots, y^n_{\mathbf{x}^m}\}$ be a subset of counterfactual events in γ. Let G' be a graph obtained from G by adding a new child W of $Y^1, \ldots, Y^n$, and let P'_* be the set of all interventional distributions in models inducing G'. Let $\gamma' = (\gamma \setminus \{y^1_{\mathbf{x}^1}, \ldots, y^n_{\mathbf{x}^m}\}) \cup \{w_{\mathbf{x}^1}, \ldots, w_{\mathbf{x}^m}\}$, where w is an arbitrary value of W. Then $P'_*, G' \not\vdash_{id} P(\gamma')$.*

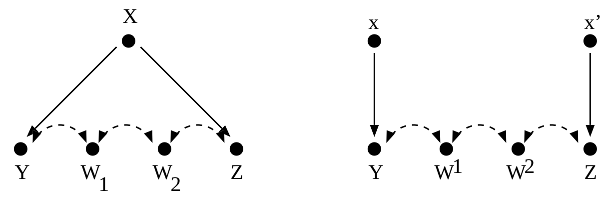

Figure 13: (a) Causal diagram (b) Corresponding counterfactual graph for the non-identifiable query $P(Y_x, W^1, W^2, Z_{x'})$.

The first result states that non-identification on a set of parents (causes) translates into non-identification on children (effects). The intuitive explanation for this is that it is possible to construct a one-to-one function from the space of distributions on causes to the space of distributions on effects. If a given $P(\gamma)$ cannot be identified from P_*, this implies that there exist two models which agree on P_*, but disagree on $P(\gamma)$, where γ is a set of counterfactual causes. It is then possible to augment these models using the one-to-one function in question to obtain disagreement on $P(\delta)$, where δ is a set of counterfactual effects of γ. A more detailed argument is found in the appendix.

Lemma 30 (contraction lemma) *Assume $P_*, G \nvdash_{id} P(\gamma)$. Let G' be obtained from G by merging some two nodes X, Y into a new node Z where Z inherits all the parents and children of X, Y, subject to the following restrictions:*

- *The merge does not create cycles.*
- *If $(\exists w_{\mathbf{s}} \in \gamma)$ where $x \in \mathbf{s}$, $y \notin \mathbf{s}$, and $X \in An(W)_G$, then $Y \notin An(W)_G$.*
- *If $(\exists y_{\mathbf{s}} \in \gamma)$ where $x \in \mathbf{s}$, then $An(X)_G = \emptyset$.*
- *If $(Y_{\mathbf{w}}, X_{\mathbf{s}} \in \gamma)$, then $\mathbf{w}$ and $\mathbf{s}$ agree on all variable settings.*

Assume $|X| \times |Y| = |Z|$ and there's some isomorphism f assigning value pairs x, y to a value $f(x, y) = z$. Let γ' be obtained from γ as follows. For any $w_{\mathbf{s}} \in \gamma$:

- *If $W \notin \{X, Y\}$, and values x, y occur in $\mathbf{s}$, replace them by $f(x, y)$.*
- *If $W \notin \{X, Y\}$, and the value of one of X, Y occur in $\mathbf{s}$, replace it by some z consistent with the value of X or Y.*
- *If X, Y do not occur in γ, leave γ as is.*
- *If $W = Y$ and $x \in \mathbf{s}$, replace $w_{\mathbf{s}}$ by $f(x, y)_{\mathbf{s} \setminus \{x\}}$.*
- *otherwise, replace every variable pair of the form $Y_{\mathbf{r}} = y, X_{\mathbf{s}} = x$ by $Z_{\mathbf{r}, \mathbf{s}} = f(x, y)$.*

Then $P_*, G' \nvdash_{id} P(\gamma')$.

This lemma has a rather complicated statement, but the basic idea is very simple. If we have a causal model with a graph G where some counterfactual $P(\gamma)$ is not identifiable, then a coarser, more "near-sighted" view of G which merges two distinct variables with their own mechanisms into a single variable with a single mechanism will not render $P(\gamma)$ identifiable. This is because merging nodes in the graph does not alter the model, but only our state of knowledge of the model. Therefore, whatever model pair was used to prove $P(\gamma)$ non-identifiable will remain the same in the new, coarser graph. The complicated statement of the lemma is due to the fact that we cannot allow arbitrary node merges, we must satisfy certain coherence conditions. For instance, the merge cannot create directed cycles in the graph.

It turns out that whenever **ID*** fails on $P(\gamma)$, the corresponding counterfactual graph contains a subgraph which can be obtained by a set of applications of the previous two lemmas to the w-graph and the zig-zag graphs. This allows an argument that shows $P(\gamma)$ cannot be identified from P_*.

Theorem 31 (completeness) *If **ID*** or **IDC*** fail, then the corresponding query is not identifiable from P_*.*

Since **ID*** is complete for $P(\gamma)$ queries, we can give a graphical characterization of counterfactual graphs where $P(\gamma)$ cannot be identified from P_*.

Theorem 32 *Let G_γ, γ' be obtained from **make-cg**(G,γ). Then $P_*, G \not\vdash_{id} P(\gamma)$ if and only if there exists a C-component $S \subseteq An(\gamma')_{G_\gamma}$ where some $X \in Pa(S)$ is set to x while at the same time either X is also a parent of another node in S and is set to another value x', or S contains a variable derived from X which is observed to be x'.*

Proof This follows from Theorem 31 and the construction of **ID***. ∎

5. Conclusions

This paper considers a hierarchy of queries about relationships among variables in graphical causal models: associational relationships which can be obtained from observational studies, cause-effect relationships obtained by experimental studies, and counterfactuals, which are derived from parallel worlds resulting from hypothetical actions, possibly conflicting with available evidence. We consider the identification problem for this hierarchy, the task of computing a query from the given causal diagram and available information lower in the hierarchy.

We provide sound and complete algorithms for this identification problem, and a graphical characterization of non-identifiable queries where these algorithms must fail. Specifically, we provide complete algorithms for identifying causal effects and conditional causal effects from observational studies, and show that a graphical structure called a *hedge* completely characterizes all cases where causal effects are non-identifiable. As a corollary, we show that the three rules of do-calculus are complete for identifying effects. We also provide complete algorithms for identifying counterfactual queries (possibly conditional) from experimental studies. If we view the structure of the causal graph as experimentally testable, as is often the case in practice, this result can be viewed as giving a full characterization of testable counterfactuals assuming structural semantics.

These results settle important questions in causal inference, and pave the way for computing more intricate causal queries which involve nested counterfactuals, such as those defining direct and indirect effects (Pearl, 2001), and path-specific effects (Avin et al., 2005). The characterization of non-identifiable queries we provide defines precisely the situations when such queries cannot be computed precisely, and must instead by approximated using methods such as bounding (Balke and Pearl, 1994a), instrumental variables (Pearl, 2000), or additional assumptions, such as linearity, which can make identification simpler.

Acknowledgments

The authors would like to thank Eleazar Eskin and Eun Yong Kang for discussing earlier versions of this work. This work was supported in part by AFOSR grant #F49620-01-1-0055, NSF grant #IIS-0535223, MURI grant #N00014-00-1-0617, and NLM grant #T15 LM07356.

Appendix A.

Here, we augment the intuitive proof outlines we gave in the main body of the paper with more formal arguments. We start with a set of results which were used to classify graphs with non-identifiable effects. In the proofs presented here, we will construct the distributions which make up our set of premises to be positive. This is because non-positive distributions present a number of technical difficulties, for instance d-separation and independence are not related in a straightforward way in such distributions, and conditional distributions may not be defined. We should mention, however, that distributions which span multiple hypothetical worlds which we discussed in Section 4 may be non-positive by definition.

Theorem 5 *If sets $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G, then $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$ in every P for which G is an I-map. Furthermore, the causal diagram induced by any semi-Markovian PCM M is a semi-Markovian I-map of the distribution $P(\mathbf{v},\mathbf{u})$ induced by M.*

Proof It is not difficult to see that if we restrict d-separation queries to a subset of variables $\mathbf{W}$ in some graph G, the corresponding independencies in $P(\mathbf{w})$ will only hold whenever the d-separation statements hold. Furthermore, if we replace G by a latent projection L (Pearl, 2000), where we view variables $\mathbf{V}\setminus\mathbf{W}$ as hidden, independencies in $P(\mathbf{w})$ will only hold whenever the corresponding d-separation statement (extended to include bidirected arcs) holds in L. ■

Theorem 10 *$P(\mathbf{v}), G \nvdash_{id} P(y|do(x))$ in G shown in Fig. 1 (a).*

Proof We construct two causal models M^1 and M^2 such that $P^1(X,Y) = P^2(X,Y)$, and $P^1_x(Y) \neq P^2_x(Y)$. The two models agree on the following: all 3 variables are boolean, U is a fair coin, and $f_X(u) = u$. Let $\oplus$ denote the exclusive or (XOR) function. Then the value of Y is determined by the function $u \oplus x$ in M^1, while Y is set to 0 in M^2. Then $P^1(Y=0) = P^2(Y=0) = 1$, $P^1(X=0) = P^2(X=0) = 0.5$. Therefore, $P^1(X,Y) = P^2(X,Y)$, while $P^2_x(Y=0) = 1 \neq P^1_x(Y=0) = 0.5$. Note that while P is non-positive, it is straightforward to modify the proof for the positive case by letting f_Y functions in both models return 1 half the time, and the values outlined above half the time. ■

Theorem 12 *Let G be a Y-rooted C-tree. Let $\mathbf{X}$ be any subset of observable nodes in G which does not contain Y. Then $P(\mathbf{v}), G \not\vdash_{id} P(y|do(\mathbf{x}))$.*

Proof We generalize the proof for the bow arc graph. We can assume without loss of generality that each unobservable U in G has exactly two observable children. We construct two models with binary nodes. In the first model, the value of all observable nodes is set to the bit parity (sum modulo 2) of the parent values. In the second model, the same is true for all nodes except Y, with the latter being set to 0 explicitly. All $\mathbf{U}$ nodes in both models are fair coins. Since G is a tree, and since every $U \in \mathbf{U}$ has exactly two children in G, every $U \in \mathbf{U}$ has exactly two distinct downward paths to Y in G. It's then easy to establish that Y counts the bit parity of every node in $\mathbf{U}$ twice in the first model. But this implies $P^1(Y = 1) = 0$.

Because bidirected arcs form a spanning tree over observable nodes in G, for any set of nodes $\mathbf{X}$ such that $Y \notin \mathbf{X}$, there exists $U \in \mathbf{U}$ with one child in $An(\mathbf{X})_G$ and one child in $G \setminus An(\mathbf{X})_G$. Thus $P^1_{\mathbf{X}}(Y = 1) > 0$, but $P^2_{\mathbf{X}}(Y = 1) = 0$. It is straightforward to generalize this proof for the positive $P(\mathbf{v})$ in the same way as in Theorem 10. ■

Theorem 13 *$P(\mathbf{v}), G \not\vdash_{id} P(y|do(pa(y)))$ if and only if there exists a subgraph of G which is a Y-rooted C-tree.*

Proof From Tian (2002), we know that whenever there is no subgraph G' of G, such that all nodes in G' are ancestors of Y, and G' is a C-component, $P_{pa(Y)}(Y)$ is identifiable. From Theorem 12, we know that if there is a Y-rooted C-tree containing a non-empty subset S of parents of Y, then $P_s(Y)$ is not identifiable. But it is always possible to extend the counterexamples which prove non-identification of $P_s(Y)$ with additional variables which are independent. ■

Theorem 17 *Let F, F' be subgraphs of G which form a hedge for $P(\mathbf{y}|do(\mathbf{x}))$. Then $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{y}|do(\mathbf{x}))$.*

Proof We first show $P_{\mathbf{X}}(\mathbf{r})$ is not identifiable in F. As before, we assume each U has two observable children. We construct two models with binary nodes. In M^1 every variable in F is equal to the bit parity of its parents. In M^2 the same is true, except all nodes in F' disregard the parent values in $F \setminus F'$. All $\mathbf{U}$ are fair coins in both models.

As was the case with C-trees, for any C-forest F, every $U \in \mathbf{U} \cap F$ has exactly two downward paths to $\mathbf{R}$. It is now easy to establish that in M^1, $\mathbf{R}$ counts the bit parity of every node in $\mathbf{U}^1$ twice, while in M^2, $\mathbf{R}$ counts the bit parity of every node in $\mathbf{U}^2 \cap F'$ twice. Thus, in both models with no interventions, the bit parity of $\mathbf{R}$ is even.

Next, fix two distinct instantiations of $\mathbf{U}$ that differ by values of $\mathbf{U}^*$. Consider the topmost node $W \in F$ with an odd number of parents in $\mathbf{U}^*$ (which exists because bidirected edges in F form a spanning tree). Then flipping the values of $\mathbf{U}^*$ once will flip the value W once. Thus the function from $\mathbf{U}$ to $\mathbf{V}$ induced by a C-forest F in M^1 and M^2 is one to one.

The above results, coupled with the fact that in a C-forest, $|\mathbf{U}| + 1 = |\mathbf{V}|$ implies that any assignment where $\sum \mathbf{r} \pmod 2 = 0$ is equally likely, and all other node assignments are impossible in both F and F'. Since the two models agree on all functions and distributions in $F \setminus F'$, $\sum_{f'} P^1 = \sum_{f'} P^2$. It follows that the observational distributions are the same in both models.

As before, we can find $U \in \mathbf{U}$ with one child in $An(\mathbf{X})_F$, and one child in $F \setminus An(\mathbf{X})_F$, which implies the probability of odd bit parity of $\mathbf{R}$ is 0.5 in M^1, and 0 in M^2.

Next, we note that the construction so far results in a non-positive distribution P. To rid our proof of non-positivity, we "soften" our two models with new unobservable binary U_R for every $R \in \mathbf{R}$ which assumes value 1 with very small probability p. Whenever U_R is 1, the node R flips its value, otherwise it keeps the value as defined above. Note that $P(\mathbf{v})$ will remain the same in both models because our augmentation is the same, and the previous unsoftened models agreed on $P(\mathbf{v})$. It's easy to see that the bit parity of R in both models will be odd only when an odd number of U_R assume values of 1. Because p is arbitrarily small, the probability of an odd parity is far smaller than the probability of even parity. Now consider what happens after $do(\mathbf{x})$. In M^2, the probability of odd bit parity stays the same. In M^1 before the addition of U_R, the probability was 0.5. But it's easy to see that U_R nodes change the bit parity of $\mathbf{R}$ in a completely symmetric way, so the probability of even parity remains 0.5.

This implies $P_{\mathbf{X}}(\mathbf{r})$ is not identifiable. Finally, to see that $P_{\mathbf{X}}(\mathbf{y})$ is not identifiable, augment our counterexample by nodes in $\mathbf{I} = An(\mathbf{Y}) \cap De(\mathbf{R})$. Without loss of generality, assume every node in $\mathbf{I}$ has at most one child. Let each node I in $\mathbf{I}$ be equal to the bit parity of its parents. Moreover, each I has an exogenous parent U_I independent of the rest of $\mathbf{U}$ which, with small probability p causes it to flip it's value. Then the bit parity of $\mathbf{Y}$ is even if and only if an odd number of $\mathbf{U_I}$ turn on. Moreover, it's easy to see $P(\mathbf{I}|\mathbf{R})$ is positive by construction. We can now repeat the previous argument. ■

Next, we provide the proof of soundness of **ID** and **IDC** using do-calculus. This both simplifies the proofs and allows us to infer do-calculus is complete from completeness of our algorithms. We will invoke do-calculus rules by just using their number, for instance "by rule 2." First, we prove that a joint distribution in a causal model can be represented as a product of interventional distributions corresponding to the set of c-component in the graph induced by the model.

Lemma 33 (c-component factorization) *Let M be a causal model with graph G. Let $\boldsymbol{y}, \boldsymbol{x}$ be value assignments. Let $C(G \setminus \boldsymbol{X}) = \{S_1, ..., S_k\}$. Then $P_{\boldsymbol{x}}(\boldsymbol{y}) = \sum_{\boldsymbol{v} \setminus (\boldsymbol{y} \cup \boldsymbol{x})} \prod_i P_{\boldsymbol{v} \setminus s_i}(s_i)$.*

Proof A proof of this was derived by Tian (2002). Nevertheless, we reprove this result using do-calculus to help with our subsequent completeness results. Assume $\mathbf{X} = \emptyset$, $\mathbf{Y} = \mathbf{V}$, $C(G) = \{S_1, ..., S_k\}$, and let $A_i = An(S_i)_G \setminus S_i$. Then

$$\prod_i P_{\mathbf{v} \setminus s_i}(s_i) = \prod_i P_{a_i}(s_i) = \prod_i \prod_{V_j \in S_i} P_{a_i}(v_j | v_\pi^{(j-1)} \setminus a_i)$$

$$= \prod_i \prod_{V_j \in S_i} P(v_j | v_\pi^{(j-1)}) = \prod_i P(v_i | v_\pi^{(i-1)}) = P(\mathbf{v}).$$

The first identity is by rule 3, the second is by chain rule of probability. To prove the third identity, we consider two cases. If $A \in A_i \setminus V_\pi^{(j-1)}$, we can eliminate the intervention on A from the expression $P_{a_i}(v_j | v_\pi^{(j-1)})$ by rule 3, since $(V_j \perp\!\!\!\perp A | V_\pi^{(j-1)})_{G_{\overline{a_i}}}$.

If $A \in A_i \cap V_\pi^{(j-1)}$, consider any back-door path from A_i to V_j. Any such path with a node not in $V_\pi^{(j-1)}$ will be d-separated because, due to recursiveness, it must contain a blocked collider. Further, this path must contain bidirected arcs only, since all nodes on this path are conditioned or fixed.

Because $A_i \cap S_i = \emptyset$, all such paths are d-separated. The identity now follows from rule 2. The last two identities are just grouping of terms, and application of chain rule.

Having proven that c-component factorization holds for $P(\mathbf{v})$, we want to extend the result to $P_{\mathbf{x}}(\mathbf{y})$. First, let's consider $P_{\mathbf{x}}(\mathbf{v} \setminus \mathbf{x})$. This is just the distribution of the submodel $M_{\mathbf{x}}$. But $M_{\mathbf{x}}$ is just an ordinary causal model inducing $G \setminus \mathbf{X}$, so we can apply the same reasoning to obtain $P_{\mathbf{x}}(\mathbf{v} \setminus \mathbf{x}) = \prod_i P_{\mathbf{v} \setminus s_i}(s_i)$, where $C(G \setminus \mathbf{X}) = \{S_1, ..., S_k\}$. As a last step, it's easy to verify that $P_{\mathbf{x}}(\mathbf{y}) = \sum_{\mathbf{v} \setminus (\mathbf{x} \cup \mathbf{y})} P_{\mathbf{x}}(\mathbf{v} \setminus \mathbf{x})$. ■

Lemma 34 *Let $\mathbf{X}' = \mathbf{X} \cap An(\mathbf{Y})_G$. Then $P_{\mathbf{x}}(\mathbf{y})$ obtained from P in G is equal to $P'_{\mathbf{x}'}(\mathbf{y})$ obtained from $P' = P(An(\mathbf{Y}))$ in $An(\mathbf{Y})_G$.*

Proof Let $\mathbf{W} = \mathbf{V} \setminus An(\mathbf{Y})_G$. Then the submodel $M_{\mathbf{W}}$ induces the graph $G \setminus \mathbf{W} = An(\mathbf{Y})_G$, and its distribution is $P' = P_{\mathbf{W}}(An(\mathbf{Y})) = P(An(\mathbf{Y}))$ by rule 3. Now $P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x}'}(\mathbf{y}) = P_{\mathbf{x}',\mathbf{w}}(\mathbf{y}) = P'_{\mathbf{x}'}(\mathbf{y})$ by rule 3. ■

Lemma 35 *Let $\mathbf{W} = (\mathbf{V} \setminus \mathbf{X}) \setminus An(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$. Then $P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x},\mathbf{w}}(\mathbf{y})$, where $\mathbf{w}$ are arbitrary values of $\mathbf{W}$.*

Proof Note that by assumption, $\mathbf{Y} \perp\!\!\!\perp \mathbf{W} | \mathbf{X}$ in $G_{\overline{\mathbf{X}},\overline{\mathbf{W}}}$. The conclusion follows by rule 3. ■

Lemma 36 *When the conditions of line 6 are satisfied, $P_{\mathbf{x}}(\mathbf{y}) = \sum_{s \setminus \mathbf{y}} \prod_{V_i \in S} P(v_i | v_\pi^{(i-1)})$.*

Proof If line 6 preconditions are met, then G local to that recursive call is partitioned into S and $\mathbf{X}$, and there are no bidirected arcs from $\mathbf{X}$ to S. The conclusion now follows from the proof of Lemma 33. ■

Lemma 37 *Whenever the conditions of the last recursive call of **ID** are satisfied, $P_{\mathbf{x}}$ obtained from P in the graph G is equal to $P'_{\mathbf{x} \cap S'}$ obtained from $P' = \prod_{V_i \in S'} P(V_i | V_\pi^{(i-1)} \cap S', v_\pi^{(i-1)} \setminus S')$ in the graph S'.*

Proof It is easy to see that when the last recursive call executes, $\mathbf{X}$ and S partition G, and $\mathbf{X} \subset An(S)_G$. This implies that the submodel $M_{\mathbf{x} \setminus S'}$ induces the graph $G \setminus (\mathbf{X} \setminus S') = S'$. The distribution $P_{\mathbf{x} \setminus S'}$ of $M_{\mathbf{x} \setminus S'}$ is equal to P' by the proof of Lemma 33. It now follows that $P_{\mathbf{x}} = P_{\mathbf{x} \cap S', \mathbf{x} \setminus S'} = P'_{\mathbf{x} \cap S'}$. ■

Theorem 38 (soundness) *Whenever **ID** returns an expression for $P_{\mathbf{x}}(\mathbf{y})$, it is correct.*

Proof If $\mathbf{x} = \emptyset$, the desired effect can be obtained from P by marginalization, thus this base case is clearly correct. The soundness of all other lines except the failing line 5 has already been established. ■

Having established soundness, we show that whenever **ID** fails, we can recover a hedge for an effect involving a subset of variables involved in the original effect expression $P(\mathbf{y}|do(\mathbf{x}))$. This in turn implies completeness.

Theorem 39 *Assume **ID** fails to identify $P_{\mathbf{X}}(\mathbf{y})$ (executes line 5). Then there exist $\mathbf{X}' \subseteq \mathbf{X}$, $\mathbf{Y}' \subseteq \mathbf{Y}$ such that the graph pair G, S returned by the fail condition of **ID** contain as edge subgraphs C-forests F, F' that form a hedge for $P_{\mathbf{X}'}(\mathbf{y}')$.*

Proof Consider line 5, and G and $\mathbf{y}$ local to that recursive call. Let $\mathbf{R}$ be the root set of G. Since G is a single C-component, it is possible to remove a set of directed arrows from G while preserving the root set $\mathbf{R}$ such that the resulting graph F is an $\mathbf{R}$-rooted C-forest.

Moreover, since $F' = F \cap S$ is closed under descendants, and since only single directed arrows were removed from S to obtain F', F' is also a C-forest. $F' \cap \mathbf{X} = \emptyset$, and $F \cap \mathbf{X} \neq \emptyset$ by construction. $\mathbf{R} \subseteq An(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$ by lines 2 and 3 of the algorithm. It's also clear that $\mathbf{y}, \mathbf{x}$ local to the recursive call in question are subsets of the original input. ■

Theorem 18 ***ID** is complete.*

Proof By the previous theorem, if **ID** fails, then $P_{\mathbf{X}'}(\mathbf{y}')$ is not identifiable in a subgraph $H = G_{An(\mathbf{Y}) \cap De(F)}$ of G. Moreover, $\mathbf{X} \cap H = \mathbf{X}'$, by construction of H. As such, it is easy to extend the counterexamples in Theorem 39 with variables independent of H, with the resulting models inducing G, and witnessing the non-identifiability of $P_{\mathbf{X}}(\mathbf{y})$. ■

Next, we prove the results necessary to establish completeness of **IDC**.

Lemma 40 *If rule 2 of do-calculus applies to a set $\mathbf{Z}$ in G for $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$ then there are no d-connected paths to $\mathbf{Y}$ that pass through $\mathbf{Z}$ in neither $G_1 = G \setminus \mathbf{X}$ given $\mathbf{Z}, \mathbf{W}$ nor in $G_2 = G \setminus (\mathbf{X} \cup \mathbf{Z})$ given $\mathbf{W}$.*

Proof Clearly, there are no d-connected paths through $\mathbf{Z}$ in G_2 given $\mathbf{W}$. Consider a d-connected path through $Z \in \mathbf{Z}$ to $\mathbf{Y}$ in G_1, given $\mathbf{Z}, \mathbf{W}$. Note that this path must either form a collider at Z or a collider which is an ancestor of Z. But this must mean there is a back-door path from $\mathbf{Z}$ to $\mathbf{Y}$, which is impossible, since rule 2 is applicable to $\mathbf{Z}$ in G for $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$. Contradiction. ■

Theorem 20 *For any G and any conditional effect $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$ there exists a unique maximal set $\mathbf{Z} = \{Z \in \mathbf{W} | P_{\mathbf{X}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{X},z}(\mathbf{y}|\mathbf{w} \setminus \{z\})\}$ such that rule 2 applies to $\mathbf{Z}$ in G for $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$. In other words, $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{X},\mathbf{Z}}(\mathbf{y}|\mathbf{w} \setminus \mathbf{z})$.*

Proof Fix two maximal sets $\mathbf{Z}_1, \mathbf{Z}_2 \subseteq \mathbf{W}$ such that rule 2 applies to $\mathbf{Z}_1, \mathbf{Z}_2$ in G for $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$. If $\mathbf{Z}_1 \neq \mathbf{Z}_2$, fix $Z \in \mathbf{Z}_1 \setminus \mathbf{Z}_2$. By Lemma 40, rule 2 applies for $\{Z\} \cup \mathbf{Z}_2$ in G for $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$, contradicting our assumption.

Thus if we fix G and $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$, any set to which rule 2 applies must be a subset of the unique maximal set $\mathbf{Z}$. It follows that $\mathbf{Z} = \{Z \in \mathbf{W} | P_{\mathbf{X}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{X},z}(\mathbf{y}|\mathbf{w} \setminus \{z\})\}$. ■

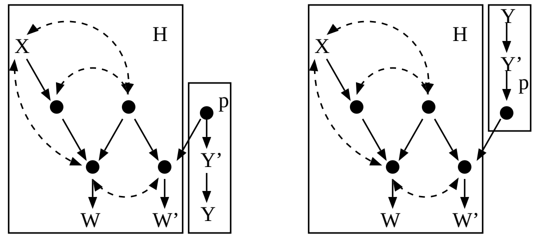

Figure 14: Inductive cases for proving non-identifiability of $P_x(y|w,w')$.

Lemma 41 *Let F, F' form a hedge for $P_{\mathbf{X}}(\mathbf{y})$. Then $F \subseteq F' \cup \mathbf{X}$.*

Proof It has been shown that **ID** fails on $P_{\mathbf{X}}(\mathbf{y})$ in G and returns a hedge if and only if $P_{\mathbf{X}}(\mathbf{y})$ is not identifiable in G. In particular, edge subgraphs of the graphs G and S returned by line 5 of **ID** form the C-forests of the hedge in question. It is easy to check that a subset of $\mathbf{X}$ and S partition G. ■

We rephrase the statement of Theorem 21 somewhat, to reduce "algebraic clutter."

Theorem 21 *Let $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$ be such that every $W \in \mathbf{W}$ has a back-door path to $\mathbf{Y}$ in $G \setminus \mathbf{X}$ given $\mathbf{W} \setminus \{W\}$. Then $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$ is identifiable in G if and only if $P_{\mathbf{X}}(\mathbf{y},\mathbf{w})$ is identifiable in G.*

Proof If $P_{\mathbf{X}}(\mathbf{y},\mathbf{w})$ is identifiable in G, then we can certainly identify $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$ by marginalization and division. The difficult part is to prove that if $P_{\mathbf{X}}(\mathbf{y},\mathbf{w})$ is not identifiable then neither is $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$.

Assume $P_{\mathbf{X}}(\mathbf{w})$ is identifiable. Then if $P_{\mathbf{X}}(\mathbf{y}|\mathbf{w})$ were identifiable, we would be able to compute $P_{\mathbf{X}}(\mathbf{y},\mathbf{w})$ by the chain rule. Thus our conclusion follows.

Assume $P_{\mathbf{X}}(\mathbf{w})$ is not identifiable. We also know that every $W \in \mathbf{W}$ contains a back-door path to some $Y \in \mathbf{Y}$ in $G \setminus \mathbf{X}$ given $\mathbf{W} \setminus \{W\}$. Fix such W and Y, along with a subgraph p of G which forms the witnessing back-door path. Consider also the hedge F, F' which witnesses the non-identifiability of $P_{\mathbf{X}'}(\mathbf{w}')$, where $\mathbf{X}' \subseteq \mathbf{X}, \mathbf{W}' \subseteq \mathbf{W}$.

Let $H = G_{De(F) \cup An(\mathbf{W}')_{G_{\overline{\mathbf{X}'}}}}$. We will attempt to show that $P_{\mathbf{X}'}(Y|\mathbf{w})$ is not identifiable in $H \cup p$. Without loss of generality, we make the following three assumptions. First, we restrict our attention to $\mathbf{W}'' \subseteq \mathbf{W}$ that occurs in $H \cup p$. Second, we assume p is a path segment which starts at H and ends at Y, and does not intersect H. Third, we assume all observable nodes in H have at most one child.

Consider the models M^1, M^2 from the proof of Theorem 17 which induce H. We extend the models by adding to them binary variables in p. Each variable $X \in p$ is equal to the bit parity of its parents, if it has any. If not, X behaves as a fair coin. If $Y \in H$ has a parent $X \in p$, the value of X is added to the bit parity computation Y makes.

Call the resulting models M^1_*, M^2_*. Because M^1, M^2 agreed on $P(H)$, and variables and functions in p are the same in both models, $P^1_* = P^2_*$. We will assume $\mathbf{w}''$ assigns 0 to every variable in $\mathbf{W}''$. What remains to be shown is that $P^1_{*\mathbf{X}}(y|\mathbf{w}'') \neq P^2_{*\mathbf{X}}(y|\mathbf{w}'')$. We will prove this by induction on the path structure of p. We handle the inductive cases first. In all these cases, we fix a node Y' that is between Y and H on the path p, and prove that if $P_{\mathbf{X}'}(y'|\mathbf{w}'')$ is not identifiable, then neither is $P_{\mathbf{X}'}(y|\mathbf{w}'')$.

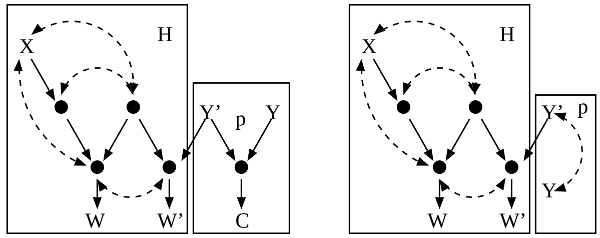

Figure 15: Inductive cases for proving non-identifiability of $P_x(y|w,w')$.

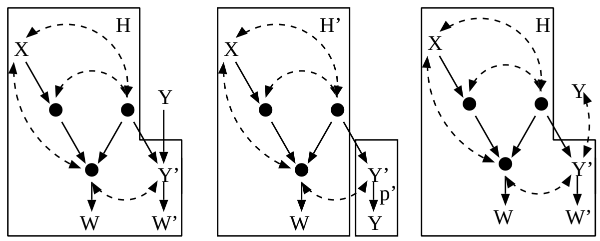

Figure 16: Base cases for proving non-identifiability of $P_x(y|w,w')$.

Assume neither Y nor Y' have descendants in $\mathbf{W}''$. If Y' is a parent of Y as in Fig. 14 (a), then $P_{\mathbf{X}'}(y|\mathbf{w}'') = \sum_{y'} P(y|y')P_{\mathbf{X}'}(y'|\mathbf{w}'')$. If Y is a parent of Y', as in Fig. 14 (b) then the next node in p must be a child of Y'. Therefore, $P_{\mathbf{X}'}(y|\mathbf{w}'') = \sum_{y'} P(y|y')P_{\mathbf{X}'}(y'|\mathbf{w}'')$. In either case, by construction $P(Y|Y')$ is a 2 by 2 identity matrix. This implies that the mapping from $P_{\mathbf{X}'}(y'|\mathbf{w}'')$ to $P_{\mathbf{X}'}(y|\mathbf{w}'')$ is one to one. If Y' and Y share a hidden common parent U as in Fig. 15 (b), then our result follows by combining the previous two cases.

The next case is if Y and Y have a common child C which is either in $\mathbf{W}''$ or has a descendant in $\mathbf{W}''$, as in Fig. 15 (a). Now $P_{\mathbf{X}'}(y|\mathbf{w}'') = \sum_{y'} P(y|y',c)P_{\mathbf{X}'}(y'|\mathbf{w}'')$. Because all nodes in $\mathbf{W}''$ were observed to be 0, $P(y|y',c)$ is again a 2 by 2 identity matrix.

Finally, we handle the base cases of our induction. In all such cases, Y is the first node not in H on the path p. Let Y' be the last node in H on the path p.

Assume Y is a parent of Y', as shown in Fig. 16 (a). By Lemma 41, we can assume $Y \not\in An(F \setminus F')_H$. By construction, $(\sum \mathbf{W}'' = Y + 2 * \sum \mathbf{U}) \pmod 2$ in M^1_*, and $(\sum \mathbf{W}'' = Y + 2 * \sum(\mathbf{U} \cap F')) \pmod 2$ in M^2_*. If every variable in $\mathbf{W}''$ is observed to be 0, then $Y = (2 * \sum \mathbf{U}) \pmod 2$ in M^1_*, and $Y = (2 * \sum(\mathbf{U} \cap F')) \pmod 2$ in M^2_*. If an intervention $do(\mathbf{x})$ is performed, $(\sum \mathbf{W}'' = Y + 2 * \sum(\mathbf{U} \cap F')) \pmod 2$ in $M^2_{*\mathbf{X}}$, by construction. Thus if $\mathbf{W}''$ are all observed to be zero, $Y = 0$ with probability 1. Note that in $M^1_{\mathbf{X}}$ as constructed in the proof of Theorem 17, $(\sum \mathbf{w}'' = \mathbf{x} + \sum \mathbf{U}') \pmod 2$, where $\mathbf{U}' \subseteq \mathbf{U}$ consists of unobservable nodes with one child in $An(\mathbf{X})_F$ and one child in $F \setminus An(\mathbf{X})_F$.

Because $Y \not\in An(F \setminus F')_H$, we can conclude that if $\mathbf{W}''$ are observed to be 0, $Y = (\mathbf{x} + \sum \mathbf{U}') \pmod 2$ in $M^1_{*\mathbf{X}'}$. Thus, $Y = 0$ with probability 0.5. Therefore, $P^1_{*\mathbf{X}'}(y|\mathbf{w}'') \neq P^2_{*\mathbf{X}'}(y|\mathbf{w}'')$ in this case.

Assume Y is a child of Y'. Now consider a graph G' which is obtained from $H \cup p$ by removing the (unique) outgoing arrow from Y' in H. If $P_{\mathbf{X}'}(Y|\mathbf{w}'')$ is not identifiable in G', we are done.

Assume $P_{\mathbf{X}'}(Y|\mathbf{w}'')$ is identifiable in G'. If $Y' \in F$, and $\mathbf{R}$ is the root set of F, then removing the Y'-outgoing directed arrow from F results in a new C-forest, with a root set $\mathbf{R} \cup \{Y'\}$. Because Y is a child of Y', the new C-forests form a hedge for $P_{\mathbf{X}'}(y,\mathbf{w}'')$. If $Y' \in H \setminus F$, then removing the Y'-outgoing directed arrow results in substituting Y for $W \in \mathbf{W}'' \cap De(Y')_H$. Thus in G', F, F' form a hedge for $P_{\mathbf{X}'}(y,\mathbf{w}'' \setminus \{w\})$. In either case, $P_{\mathbf{X}'}(y,\mathbf{w}'')$ is not identifiable in G'.

If $P_{\mathbf{X}'}(\mathbf{w}'')$ is identifiable in G', we are done. If not, consider a smaller hedge $H' \subset H$ witnessing this fact. Now consider the segment p' of p between Y and H'. We can repeat the inductive argument for H', p' and Y. See Fig. 16 (b).

If $P_{\mathbf{X}'}(\mathbf{w}'')$ is identifiable in G', we are done. If not, consider a smaller hedge $H' \subset H$ witnessing this fact. Now consider the segment p' of p between Y and H'. We can repeat the inductive argument for H', p' and Y. See Fig. 16 (b). If Y and Y' have a hidden common parent, as is the case in Fig. 16 (c), we can combine the first inductive case, and the first base case to prove our result.

We conclude the proof by introducing a slight change to rid us of non-positivity in the distributions P^1, P^2 in our counterexamples. Specifically, for every node I in $p \cup (De(\mathbf{R}) \cap An(\mathbf{Y}))$, add a new binary exogenous parent U_I which is independent of other nodes in $\mathbf{U}$, and has an arbitrarily small probability of assuming the value 1, and causing its child to flip its current value. We let P_{odd} be the probability an odd number of U_I nodes assume the value 1. Because $P(U_I = 1)$ is vanishingly small for every I, P_{odd} is much smaller than 0.5. It's easy to see that P is positive in counterexamples augmented in this way. In the base case when Y is a parent of Y', we modify our equations to account for the addition of U_I. Specifically, $(\sum \mathbf{W}'' = Y + 2 * \sum \mathbf{U} + \sum \mathbf{U_I}) \pmod 2$ in M^1_*, and $(\sum \mathbf{W}'' = Y + 2 * \sum(\mathbf{U} \cap F') + \sum \mathbf{U_I}) \pmod 2$ in M^2_*, where $U_{\mathbf{U}}$ is the set of nodes added. If every variable in $\mathbf{W}''$ is observed to be 0, then $Y = (2 * \sum \mathbf{U} + \sum \mathbf{U_I}) \pmod 2$ in M^1_*, and $Y = (2 * \sum(\mathbf{U} \cap F') + \sum \mathbf{U_I}) \pmod 2$ in M^2_*. So prior to the intervention, $P(Y = 1|\mathbf{w}'') = P_{odd}$. But because $P^1_{\mathbf{X}'}(Y = 1|\mathbf{w}'') = 0.5$, adding U_I nodes to the model does not change this probability. Because $P^2(Y = 1|\mathbf{w}'') = P^2_{\mathbf{X}}(Y = 1|\mathbf{w}'')$, our conclusion follows.

In the inductive cases above, we showed that $P_{\mathbf{X}}(Y' = Y|\mathbf{W}'') = 1$ in our counterexamples. It's easy to see that with the addition of U_I, $P_{\mathbf{X}}(Y' = Y|\mathbf{W}'') = P_{odd}$. This implies that if $P^1_{\mathbf{X}}(Y'|\mathbf{W}'') \neq P^2_{\mathbf{X}}(Y'|\mathbf{W}'')$, then $P^1_{\mathbf{X}}(Y|\mathbf{W}'') \neq P^2_{\mathbf{X}}(Y|\mathbf{W}'')$.

This completes the proof. ■

What remains for us to show are the theorems which imply the soundness and completeness results in Section 4. The most important point in these proofs is that counterfactual graphs are generally no different from causal diagrams discussed in Sections 2 and 3, with their only special feature being that by construction, some nodes in the graph happen to share functions. This means that a lot of results we already proved for Section 3 can be reused without change.

Lemma 42 *If the preconditions of line 7 are met,* $P(S) = P_{\mathbf{x}}(\boldsymbol{var}(S))$*, where* $\mathbf{x} = \bigcup \boldsymbol{sub}(S)$.

Proof Let $\mathbf{x} = \bigcup \mathbf{sub}(S)$. Since the preconditions are met, $\mathbf{x}$ does not contain conflicting assignments to the same variable, which means $do(\mathbf{x})$ is a sound action in the original causal model. Note that for any variable $Y_{\mathbf{W}}$ in S, any variable in $(Pa(S) \setminus S) \cap An(Y_{\mathbf{W}})_S$ is already in $\mathbf{w}$, while any variable in $(Pa(S) \setminus S) \setminus An(Y_{\mathbf{W}})_S$ can be added to the subscript of $Y_{\mathbf{W}}$ without changing the variable. Since $Y \cap \mathbf{X} = \emptyset$ by assumption, $Y_{\mathbf{W}} = Y_{\mathbf{X}}$. Since $Y_{\mathbf{W}}$ was arbitrary, our result follows. ■

For convenience, we show the soundness of **ID*** and **IDC*** asserted in Theorem 26 separately.

Theorem 26 (a) *If **ID*** succeeds, the expression it returns is equal to $P(\gamma)$ in a given causal graph.*

Proof The proof outline in Section 3 is sufficient for everything except the base cases. In particular, line 6 follows by Lemma 33. For soundness, we only need to handle the positive base case, which follows from Lemma 42. ■

The soundness of **IDC*** is also fairly straightforward.

Theorem 26 (b) *If **IDC*** does not output **FAIL**, the expression it returns is equal to $P(\gamma|\delta)$ in a given causal graph, if that expression is defined, and **UNDEFINED** otherwise.*

Proof Theorem 20 shows how an operation similar to line 4 is sound by rule 2 of do-calculus (Pearl, 1995) when applied in a causal diagram. But we know that the counterfactual graph is just a causal diagram for a model where some nodes share functions, so the same reasoning applies. The rest is straightforward. ■

To show completeness of **ID*** and **IDC***, we first prove a utility lemma which will make it easier to construct counterexamples which agree on P_* but disagree on a given counterfactual query.

Lemma 43 *Let G be a causal graph partitioned into a set $\{S_1,...,S_k\}$ of C-components. Then two models M_1, M_2 which induce G agree on P_* if and only if their submodels $M^1_{\mathbf{v}\setminus s_i}$, $M^2_{\mathbf{v}\setminus s_i}$ agree on P_* for every C-component S_i, and value assignment $\mathbf{v}\setminus s_i$.*

Proof This follows from C-component factorization: $P(\mathbf{v}) = \prod_i P_{\mathbf{v}\setminus s_i}(s_i)$. This implies that for every $do(\mathbf{x})$, $P_{\mathbf{x}}(\mathbf{v})$ can be expressed as a product of terms $P_{\mathbf{v}\setminus(s_i\setminus\mathbf{x})}(s_i\setminus\mathbf{x})$, which implies the result. ■

The next result generalizes Lemma 27 to a wider set of counterfactual graphs which result from non-identifiable queries.

Lemma 28 *Assume G is such that X is a parent of Y and Z, and Y and Z are connected by a bidirected path with observable nodes $W^1,...,W^k$ on the path. Then $P_*, G \not\vdash_{id} P(y_x, w^1,...,w^k, z_{x'})$, $P(y_x, w^1,...,w^k, z)$ for any value assignments $y, w^1,...,w^k, z$.*

Proof We construct two models with graph G as follows. In both models, all variables are binary, and $P(\mathbf{u})$ is uniform. In M^1, each variable is set to the bit parity of its parents. In M^2, the same is true except Y and Z ignore the values of X. To prove that the two models agree on P_*, we use Lemma 43. Clearly the two models agree on $P(X)$. To show that the models also agree on $P_x(\mathbf{V}\setminus X)$ for all values of x, note that in M_2 each value assignment over $\mathbf{V}\setminus X$ with even bit parity is equally likely, while no assignment with odd bit parity is possible. But the same is true in M^1 because any value of x contributes to the bit parity of $\mathbf{V}\setminus X$ exactly twice. The agreement of M^1_x, M^2_x on P_* follows by the graph structure of G.

To see that the result is true, we note firstly that $P(\Sigma_i W^i + Y_x + Z_{x'} \pmod 2 = 1) = P(\Sigma_i W^i + Y_x + Z \pmod 2 = 1) = 0$ in M^2, while the same probabilities are positive in M^1, and secondly that in both models distributions $P(y_x, w^1,...,w^k, z_{x'})$ and $P(y_x, w^1,..,w^k, z)$ assign equal probabilities to outcomes with positive probabilities, while we just established that the set of these possible outcomes differs in M_1 and M_2. Note that the proof is easy to generalize for positive P_* by adding a small probability for Y to flip its normal value. ■

To obtain a full characterization of non-identifiable counterfactual graphs, we augment the difficult graphs we obtained from the previous two results using certain graph transformation rules which preserve non-identifiability. These rules are given in the following two lemmas.

Lemma 29 *Assume $P_*, G \not\vdash_{id} P(\gamma)$. Let $\{y^1_{\mathbf{x}^1}, ..., y^n_{\mathbf{x}^m}\}$ be a subset of counterfactual events in γ. Let G' be a graph obtained from G by adding a new child W of $Y^1, ..., Y^n$, and let P'_* be the set of all interventional distributions in models inducing G'. Let $\gamma' = (\gamma \setminus \{y^1_{\mathbf{x}^1}, ..., y^n_{\mathbf{x}^m}\}) \cup \{w_{\mathbf{x}^1}, ..., w_{\mathbf{x}^m}\}$, where w is an arbitrary value of W. Then $P'_*, G' \not\vdash_{id} P(\gamma')$.*

Proof Let M^1, M^2 witness $P_*, G \not\vdash_{id} P(\gamma)$. We will extend these models to witness $P'_*, G' \not\vdash_{id} P(\gamma')$. Since the function of a newly added W will be shared, and M^1, M^2 agree on P_* in G, the extensions will agree on P'_* by Lemma 43. We have two cases.

Assume there is a variable Y^i such that $y^i_{\mathbf{x}^j}, y^i_{\mathbf{x}^k}$ are in γ. By Lemma 27, $P_*, G \not\vdash_{id} P(y^i_{\mathbf{x}^j}, y^i_{\mathbf{x}^k})$. Then let W be a child of just Y^i, and assume $|W| = |Y^i| = c$. Let W be set to the value of Y^i with probability $1-\varepsilon$, and otherwise it is set to a uniformly chosen random value of Y^i among the other $c-1$ values. Since ε is arbitrarily small, and since $W_{\mathbf{x}^j}$ and $W_{\mathbf{x}^k}$ pay attention to the same U variable, it is possible to set ε in such a way that if $P^1(Y^i_{\mathbf{x}^j}, Y^i_{\mathbf{x}^k}) \neq P^2(Y^i_{\mathbf{x}^j}, Y^i_{\mathbf{x}^k})$, however minutely, then $P^1(W_{\mathbf{x}^j}, W_{\mathbf{x}^k}) \neq P^2(W_{\mathbf{x}^j}, W_{\mathbf{x}^k})$.

Otherwise, let $|W| = \prod_i |Y^i|$, and let $P(W|Y^1, ..., Y^n)$ be an invertible stochastic matrix. Our result follows. ■

Lemma 30 *Assume $P_*, G \not\vdash_{id} P(\gamma)$. Let G' be obtained from G by merging some two nodes X, Y into a new node Z where Z inherits all the parents and children of X, Y, subject to the following restrictions:*

- *The merge does not create cycles.*
- *If $(\exists w_{\mathbf{s}} \in \gamma)$ where $x \in \mathbf{s}$, $y \notin \mathbf{s}$, and $X \in An(W)_G$, then $Y \notin An(W)_G$.*
- *If $(\exists y_{\mathbf{s}} \in \gamma)$ where $x \in \mathbf{s}$, then $An(X)_G = \emptyset$.*
- *If $(Y_{\mathbf{w}}, X_{\mathbf{s}} \in \gamma)$, then $\mathbf{w}$ and $\mathbf{s}$ agree on all variable settings.*

Assume $|X| \times |Y| = |Z|$ and there's some isomorphism f assigning value pairs x, y to a value $f(x, y) = z$. Let γ' be obtained from γ as follows. For any $w_{\mathbf{s}} \in \gamma$:

- *If $W \notin \{X, Y\}$, and values x, y occur in $\mathbf{s}$, replace them by $f(x, y)$.*
- *If $W \notin \{X, Y\}$, and the value of one of X, Y occur in $\mathbf{s}$, replace it by some z consistent with the value of X or Y.*
- *If X, Y do not occur in γ, leave γ as is.*
- *If $W = Y$ and $x \in \mathbf{s}$, replace $w_{\mathbf{s}}$ by $f(x, y)_{\mathbf{s} \setminus \{x\}}$.*

- *otherwise, replace every variable pair of the form $Y_{\boldsymbol{r}} = y, X_{\mathbf{s}} = x$ by $Z_{\boldsymbol{r},\mathbf{s}} = f(x,y)$.*

Then $P_*, G' \nvdash_{id} P(\gamma')$.

Proof Let Z be the Cartesian product of X, Y, and fix f. We want to show that the proof of non-identification of $P(\gamma)$ in G carries over to $P(\gamma')$ in G'.

We have five modification conditions which can apply to a variable $w_{\mathbf{s}} \in \gamma$. However, since γ is left alone if X, Y do not occur in γ (the third condition), only the remaining four of these conditions result in an actual modification of a counterfactual variable in γ.

We go through these remaining conditions one by one. The first clearly results in the same counterfactual variable. For the second, due to the restrictions we imposed, $w_{\mathbf{z}} = w_{\mathbf{z},y,x}$, which means we can apply the first modification.

For the fourth, we have $P(\gamma) = P(\delta, y_{x,\mathbf{z}})$. By our restrictions, and rule 2 of do-calculus (Pearl, 1995), this is equal to $P(\delta, y_{\mathbf{z}}|x_{\mathbf{z}})$. Since this is not identifiable, then neither is $P(\delta, y_{\mathbf{z}}, x_{\mathbf{z}})$. Now it's clear that our modification is equivalent to one applied after the fifth condition.

The fifth modification is simply a merge of events consistent with a single causal world into a conjunctive event, which does not change the overall expression. ■

We are now ready to show the main completeness results for counterfactual identification algorithms. Again, we prove this results separately for **ID*** and **IDC*** for convenience.

Theorem 31 (a) ***ID**** *is complete.*

Proof We want to show that if line 8 fails, the original $P(\gamma)$ cannot be identified. There are two broad cases to consider. If G_γ contains the w-graph, the result follows by Lemmas 27 and 29. If not, we argue as follows.

Fix some X which witnesses the precondition on line 8. We can assume X is a parent of some nodes in S. Assume no other node in $\mathbf{sub}(S)$ affects S (effectively we delete all edges from parents of S to S except from X). Because the w-graph is not a part of G_γ, this has no ramifications on edges in S. Further, we assume X has two values in S.

If $X \notin S$, fix $Y, W \in S \cap Ch(X)$. Assume S has no directed edges at all. Then $P_*, G \nvdash_{id} P(S)$ by Lemma 28. The result now follows by Lemma 29, and by construction of G_γ, which implies all nodes in S have some descendant in γ.

If S has directed edges, we want to show $P_*, G \nvdash_{id} P(R(S))$, where $R(S)$ is the subset of S with no children in S. We can recover this from the previous case as follows. Assume S has no edges as before. For a node $Y \in S$, fix a set of childless nodes $\mathbf{X} \in S$ which are to be their parents. Add a virtual node Y' which is a child of all nodes in $\mathbf{X}$. Then $P_*, G \nvdash_{id} P((S \setminus \mathbf{X}) \cup Y')$ by Lemma 29. Then $P_*, G \nvdash_{id} P(R(S'))$, where S' is obtained from S by adding edges from $\mathbf{X}$ to Y by Lemma 30, which applies because no w-graph exists in G_γ. We can apply this step inductively to obtain the desired forest (all nodes have at most one child) S while making sure $P_*, G \nvdash_{id} P(R(S))$.

If S is not a forest, we can simply disregard extra edges so effectively it is a forest. Since the w-graph is not in G_γ this does not affect edges from X to S.

If $X \in S$, fix $Y \in S \cap Ch(X)$. If S has no directed edges at all, replace X by a new virtual node Y, and make X be the parent of Y. By Lemma 28, $P_*, G \nvdash_{id} P((S \setminus x) \cup y_x)$. We now repeat the same steps as before, to obtain that $P_*, G \nvdash_{id} P((R(S) \setminus x) \cup y_x)$ for general S. Now we use Lemma 30 to obtain $P_*, G \not\vdash_{id} P(R(S))$. Having shown $P_*, G \not\vdash_{id} P(R(S))$, we conclude our result by inductively applying Lemma 29. ■

Theorem 31 (b) ***IDC**** *is complete.*

Proof The difficult step is to show that after line 5 is reached, if $P_*, G \not\vdash_{id} P(\gamma, \delta)$ then $P_*, G \not\vdash_{id} P(\gamma|\delta)$. If $P_*, G \vdash_{id} P(\delta)$, this is obvious. Assume $P_*, G \not\vdash_{id} P(\delta)$. Fix the S which witnesses that for $\delta' \subseteq \delta$, $P_*, G \not\vdash_{id} P(\delta')$. Fix some Y such that a back-door, that is, starting with an incoming arrow, path exists from δ' to Y in $G_{\gamma,\delta}$. We want to show that $P_*, G \not\vdash_{id} P(Y|\delta')$. Let $G' = G_{An(\delta') \cap De(S)}$.

Assume Y is a parent of a node $D \in \delta'$, and $D \in G'$. Augment the counterexample models which induce counterfactual graph G' with an additional binary node for Y, and let the value of D be set as the old value plus Y modulo $|D|$. Let Y attain value 1 with vanishing probability ε. That the new models agree on P_* is easy to establish. To see that $P_*, G \not\vdash_{id} P(\delta')$ in the new model, note that $P(\delta')$ in the new model is equal to $P(\delta' \setminus D, D = d) * (1 - \varepsilon) + P(\delta' \setminus D, D = (d-1) \pmod{|D|}) * \varepsilon$. Because ε is arbitrarily small, this implies our result. To show that $P_*, G \not\vdash_{id} P(Y = 1|\delta')$, we must show that the models disagree on $P(\delta'|Y = 1)/P(\delta')$. But to do this, we must simply find two consecutive values of D, $d, d+1 \pmod{|D|}$ such that $P(\delta' \setminus D, d+1 \pmod{|D|})/P(\delta' \setminus D, d)$ is different in the two models. But this follows from non-identification of $P(\delta')$.

If Y is not a parent of $D \in G'$, then either it is further along on the back-door path or it's a child of some node in G'. In case 1, we must construct the distributions along the back-door path in such a way that if $P_*, G \not\vdash_{id} P(Y'|\delta')$ then $P_*, G \not\vdash_{id} P(Y|\delta')$, where Y' is a node preceding Y on the path. The proof follows closely the one in Theorem 21. In case 2, we duplicate the nodes in G' which lead from Y to δ', and note that we can show non-identification in the resulting graph using reasoning in case 1. We obtain our result by applying Lemma 30. ■

References

Chen Avin, Ilya Shpitser, and Judea Pearl. Identifiability of path-specific effects. In *International Joint Conference on Artificial Intelligence*, volume 19, pages 357–363, 2005.

Alexander Balke and Judea Pearl. Counterfactual probabilities: Computational methods, bounds and applications. In *Proceedings of UAI-94*, pages 46–54, 1994a.

Alexander Balke and Judea Pearl. Probabilistic evaluation of counterfactual queries. In *Proceedings of AAAI-94*, pages 230–237, 1994b.

Alexander Philip Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society*, 41:1–31, 1979.

David Galles and Judea Pearl. An axiomatic characterization of causal counterfactuals. *Foundation of Science*, 3:151–182, 1998.

Trygve Haavelmo. The statistical implications of a system of simultaneous equations. *Econometrica*, 11: 1–12, 1943.

Joseph Halpern. Axiomatizing causal reasoning. *Journal of A.I. Research*, pages 317–337, 2000.

Yimin Huang and Marco Valtorta. Pearl's calculus of interventions is complete. In *Twenty Second Conference On Uncertainty in Artificial Intelligence*, 2006a.

Yimin Huang and Marco Valtorta. Identifiability in causal bayesian networks: A sound and complete algorithm. In *Twenty-First National Conference on Artificial Intelligence*, 2006b.

Manabu Kuroki and Masami Miyakawa. Identifiability criteria for causal effects of joint interventions. *Journal of Japan Statistical Society*, 29:105–117, 1999.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000. ISBN 0-521-77362-8.

Judea Pearl. Direct and indirect effects. In *Proceedings of UAI-01*, pages 411–420, 2001.

Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29:241–288, 1986.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan and Kaufmann, San Mateo, 1988.

Judea Pearl. Graphical models, causality, and intervention. *Statistical Science*, 8:266–9, 1993a.

Judea Pearl. A probabilistic calculus of actions. In *Uncertainty in Artificial Intelligence (UAI)*, volume 10, pages 454–462, 1993b.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–709, 1995. URL `citeseer.ist.psu.edu/55450.html`.

Judea Pearl and James M. Robins. Probabilistic evaluation of sequential plans from causal models with hidden variables. In *Uncertainty in Artificial Intelligence*, volume 11, pages 444–453, 1995.

Ilya Shpitser and Judea Pearl. Identification of joint interventional distributions in recursive semi-markovian causal models. In *Twenty-First National Conference on Artificial Intelligence*, 2006a.

Ilya Shpitser and Judea Pearl. Identification of conditional interventional distributions. In *Uncertainty in Artificial Intelligence*, volume 22, 2006b.

Ilya Shpitser and Judea Pearl. What counterfactuals can be tested. In *Twenty Third Conference on Uncertainty in Artificial Intelligence, forthcoming*. Morgan Kaufmann, 2007.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Springer Verlag, New York, 1993.

Jin Tian. *Studies in Causal Reasoning and Learning*. PhD thesis, Department of Computer Science, University of California, Los Angeles, 2002.

Thomas S. Verma. Causal networks: semantics and expressiveness. Technical Report R-65, Cognitive Systems Laborator, University of California, Los Angeles, 1986.

Sewall Wright. Correlation and causation. *Journal of Agricultural Research*, 20:557–585, 1921.