Efficient Identification in Linear Structural Causal Models with Auxiliary Cutsets

Daniel Kumor [1] **Carlos Cinelli** [2] **Elias Bareinboim** [3]

Abstract

We develop a polynomial-time algorithm for identification of structural coefficients in linear causal models that subsumes previous efficient state-of-the-art methods, unifying several disparate approaches to identification in this setting. Building on these results, we develop a procedure for identifying total causal effects in linear systems.

1. Introduction

Regression analysis is one of the most popular methods used to understand the relationships across multiple variables throughout the empirical sciences. The most common type of regression is linear, where one attempts to explain the observed data by fitting a line (or hyperplane), minimizing the sum of the corresponding deviations. This method can be traced back at least to the pioneering work of Legendre and Gauss (Legendre, 1805; Gauss, 1809), in the context of astronomical observations (Stigler, 1986). Linear regression and its generalizations have been the go-to tool of a generation of data analysts, and the workhorse behind many recent breakthroughs in the sciences, in businesses, and throughout engineering. Based on modern statistics and machine learning techniques, it's feasible to handle regression instances up to thousands, sometimes even millions of variables at the same time (Hastie et al., 2009).

Despite the power entailed by this family of methods, one of its main drawbacks is that it only explains the association (or correlation) between purported variables, while remaining silent with respect to any possible cause and effect relationship. In practice, however, learning about causation is often the main goal of the exercise, sometimes, the very reason one engaged in the data collection and the subsequent analysis in the first place. For instance, a health scientist may be interested in knowing the effect of a new treatment on the survival of its patients, while an economist may attempt to understand the unintended consequences of a new policy on a nation's gross domestic product. If regression analysis doesn't allow scientists to answer their more precious questions, which framework could legitimize such inferences?

The discipline of causal inference is interested in formalizing precisely these conditions, and, more broadly, providing a principled approach to combining data and partial understanding about the underlying generating processes to support causal claims (Pearl, 2000; Spirtes et al., 2000; Bareinboim & Pearl, 2016). One popular framework used to study this family of problems is known as structural causal models (SCMs, for short). Given the pervasiveness of linear regression in data-driven disciplines, we'll focus on the class of linear structural models, following the treatment provided in Wright (1921) and as discussed more contemporaneously in Pearl (2000, Ch. 5).

In this class of SCMs, the set of observed variables is determined by a linear combination of their direct causes and latent confounders (or errors terms). Formally, this is represented as a system of linear equations $X = \Lambda^T X + \epsilon$, where X is a vector of observed variables, ϵ is a vector of latent variables, and Λ is an upper triangular matrix of *direct effects*, otherwise known as path coefficients, whose ijth element, λ_{ij} gives the magnitude of the direct causal effect of x_i on x_j. The error terms are commonly assumed to be normally distributed, which means that the covariance matrix Σ characterizes the observational distribution. This matrix can be linked to the underlying structural parameters through the system of polynomial equations $\Sigma = XX^T = (I - \Lambda)^{-T}\Omega(I - \Lambda)^{-1}$. Identification then is reduced to finding the elements of Λ that are uniquely determined by the above system. If a structural parameter can be expressed in terms of the elements of Σ alone, it is said to be generically identifiable (Foygel et al., 2012; Drton & Weihs, 2015).

Generic identification can be fully solved using computer algebra as shown in García-Puente et al. (2010). In practice, however, this method has a doubly-exponential computational complexity (Bardet & Chyzak, 2005), becoming

[1]Dept. of Computer Science, Purdue University, West Lafayette, IN, USA [2]Dept. of Statistics, University of California, Los Angeles, CA, USA [3]Dept. of Computer Science, Columbia University, New York, NY, USA. Correspondence to: Daniel Kumor <dkumor@purdue.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

impractical for instances larger than four or five variables (Foygel et al., 2012). It is currently unknown whether the identifiability of an arbitrary structural parameter can be determined in polynomial time.

Instead, most efficient identification algorithms search for patterns in the covariance matrix known to correspond to specific, solvable subsystems of direct effects. The most well-known of such methods is known as instrumental variable (IV) (Wright, 1928). In modern terminology, an "instrument" z relative to a direct effect λ_{xy} needs to be d-separated (Koller & Friedman, 2009) from y, while it cannot be d-separated from x in the modified graph where the target edge $x \to y$ is removed (Pearl, 2000). The existence of such a variable means that $\lambda_{xy} = \frac{\sigma_{zy}}{\sigma_{zx}}$, and, therefore, is uniquely determined by the observational distribution.

The IV and its generalization, the conditional IV (cIV), are heavily exploited in the literature, particularly in the field of econometrics (Fisher, 1966; Bowden & Turkington, 1984). Despite its success, many identifiable effects in a linear system cannot be found with IVs and cIVs. Therefore, the past two decades has witnessed a push in the development of successively more sophisticated identification methods.

Two promising avenues towards efficiently solving generic identification are conditional Auxiliary Variables (cAV) (Chen et al., 2017) and Instrumental Cutsets (IC) (Kumor et al., 2019), both of which provide poly-time algorithms encompassing previous works such as the half-trek criterion (HTC) (Foygel et al., 2012), the generalized half-trek criterion (gHTC) (Chen, 2016; Weihs et al., 2018), auxiliary variable sets (AVS) (Chen et al., 2016), and conditional instrumental variables (cIV) (Van der Zander et al., 2015).

Another class of graphical criteria has no known efficient algorithm to date. These methods currently require an exponential number of steps. One such algorithm, the generalized instrumental set (gIS) (Brito & Pearl, 2002) and its generalization, the quasi-AV set (qAVS) (Chen et al., 2017), have thus far eluded characterization. The perceived difficulty of finding gIS (Tian, 2007) is compounded by a proof that given a candidate set of instruments, finding whether conditioning sets exist to make a gIS is NP-hard (Van der Zander & Liskiewicz, 2016). This was further exasperated when Kumor et al. (2019) proved that finding simplified conditional instrumental sets (scIS) is also NP-hard.

We roughly summarize these methods in Fig. 1, even though it lies outside the scope of this paper to survey this rich literature. It can be seen that the literature is splintered among several competing methods, with the state-of-the-art in poly-time identification being IC or cAV, depending on the setting. It's not currently known how these methods compare to qAVS, which has undetermined complexity.

The main goal of this paper is to provide an unifying treatment of the threads and corresponding algorithms found in this literature, under the umbrella of a single, efficient algorithm. In particular, our contributions are:

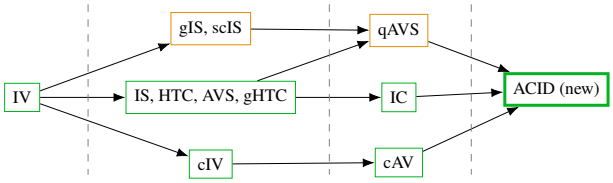

Figure 1. Summary of the discussed identification methods. $a \to b$ means all methods in b subsume all methods in a. Green boxes represent existence of polynomial-time algorithms, orange ones are undetermined or NP-hard. ACID is the newly proposed algorithm.

- We develop the Auxiliary Cutset Identification Algorithm (ACID), which runs in polynomial-time, and unifies and *strictly subsumes* existing efficient identification methods (such as IC and cAV) as well as conditioning-based methods with unknown complexity (qAVS).

- We design a strategy for identification of total effects based on the decomposition of the target query into smaller, more manageable effects that can be effectively and systematically solved by algorithms designed for direct effects.

2. Preliminaries

The causal graph of an SCM is defined as a triple $G = (V, D, B)$, representing the nodes, directed, and bidirected edges, respectively. A linear SCM has a node v_i for each variable x_i, a directed edge between v_i and v_j for each nonzero λ_{ij}, and a bidirected edge between v_i and v_j whenever there is latent confounding between the variables, i.e., nonzero $\epsilon_{ij} = \sigma_{\epsilon_i \epsilon_j}$ (Fig. 2a). When clear from the context, we will use λ_{ij} and ϵ_{ij} to refer to the corresponding directed and bidirected edges in the graph. In keeping with other works, we define $Pa(x_i)$ as the set of parents of x_i, $An(x_i)$ as ancestors of x_i, $De(x_i)$ as descendants of x_i, and $Sib(x_i)$ as variables connected to x_i with bidirected edges (i.e., variables with latent common causes).

A path in the graph is said to be "active" conditioned on a (possibly empty) set W if it contains a collider at node b ($\cdots \to b \leftarrow \cdots$) only when $b \in W \cup An(W)$, and if it does not otherwise contain vertices from W (see d-separation (Koller & Friedman, 2009)). Active paths without conditioning do not contain colliders, and are referred to as treks (Sullivant et al., 2010). The covariances of observed variables have a graphical interpretation in terms of a sum over all treks between nodes in the causal graph, namely

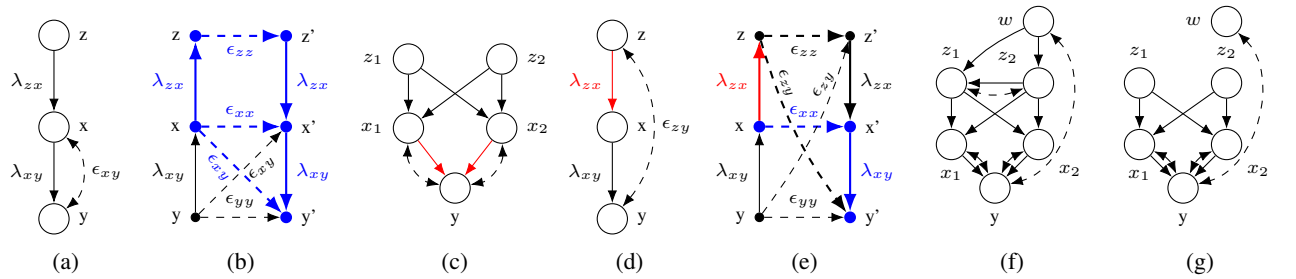

Figure 2. In (a), z is an instrument for λ_{xy}. (b) shows a directed flow graph encoding the covariances between variables, with σ_{xy} highlighted in blue. (c) λ_{x_1y} cannot be solved using a conditional instrument, but can be approached using the instrumental set $\{z_1, z_2\}$. (d) knowledge of λ_{zx} can be used to remove the backdoor path from x to y, shown in (e)'s flow graph (f) λ_{x_1y} cannot be solved by either IC nor cAV, but can be solved using gIS. (g) After applying Tian's model decomposition to the graph in (f), it becomes equivalent to the one in c, making the desired parameter efficiently solvable.

$\sigma_{xy} = \sum \pi(x, y)$, where π is the product of structural parameters along the trek.

While methods such as Wright's rules of path analysis are commonly used for reading covariances directly from the causal graph, we follow the identification literature and define a "flow graph", which explicitly encodes the treks between nodes in its directed paths, allowing a direct algorithmic approach to path analysis.

Definition 2.1. *(Sullivant et al., 2010; Weihs et al., 2018; Kumor et al., 2019) The flow graph of $G = (V, D, B)$ is the graph with vertices $V \cup V'$ containing the edges:*

- $j \to i$ *and* $i' \to j'$ *with weight* λ_{ij} *if* $i \to j \in D$
- $i \to i'$ *with weight* ϵ_{ii} *for all* $i \in V$
- $i \to j'$ *with weight* ϵ_{ij} *if* $i \leftrightarrow j \in B$

This graph is referred to as G_{flow}. The nodes without $'$ are called "source", or "top" nodes, and the nodes with $'$ are called "sink" or "bottom" nodes.

As an example, consider the instrumental variable graph of Fig. 2a. Each trek corresponds to a directed path in the flow graph shown in Fig. 2b. Summing over all paths from x to y' (in blue), we obtain the covariance between these two variables,

$$\sigma_{xy} = \lambda_{zx}\epsilon_{zz}\lambda_{zx}\lambda_{xy} + \epsilon_{xx}\lambda_{xy} + \epsilon_{xy} = \sigma_{xx}\lambda_{xy} + \epsilon_{xy}$$

Reading covariances from the flow graph can help in visualizing the IV: $\frac{\sigma_{zy}}{\sigma_{zx}} = \frac{\epsilon_{zz}\lambda_{zx}\lambda_{xy}}{\epsilon_{zz}\lambda_{zx}} = \lambda_{xy}$. Finally, in Fig. 2c, there is no single instrument, but one can use the instrumental set $\{z_1, z_2\}$ to construct a solvable system of equations (Brito & Pearl, 2002),

$$\begin{aligned} \sigma_{z_1y} &= \sigma_{z_1x_1}\lambda_{x_1y} + \sigma_{z_1x_2}\lambda_{x_2y} \\ \sigma_{z_2y} &= \sigma_{z_2x_1}\lambda_{x_1y} + \sigma_{z_2x_2}\lambda_{x_2y} \end{aligned} \tag{1}$$

To simplify discussion of paths in the graph, we define the *partial effect* of a on b avoiding set C, denoted as $\delta_{ab.C}$, as the causal effect of a on b when holding all variables in C constant ($\delta_{ab.C} = \frac{\partial}{\partial a}\mathbf{E}[b \mid do(a), do(C)]$). This corresponds to the sum of all products of direct effects along the directed paths from a to b that do not cross any nodes in C. In particular, two special cases are worth noting. First, when $C = \emptyset$, then $\delta_{ab.C} = \delta_{ab}$ is the *total effect* of a on b. Second, when $a \in Pa(b)$ and $C = Pa(b) \setminus \{a\}$, we recover the direct effect $\delta_{ab.C} = \lambda_{ab}$.

2.1. Auxiliary Variables

One can leverage direct effects that were previously identified to create new variables to help with the identification of further edges. For example, in Fig. 2d, λ_{zx} can be trivially identified with the regression coefficient of z on x, $\lambda_{zx} = \frac{\sigma_{zx}}{\sigma_{zz}}$. Once λ_{zx} is known, one can create an **auxiliary variable** (AV) $x^* = x - \lambda_{zx}z$ subtracting out the direct effect of z. This new variable behaves as if the edge λ_{zx} did not exist in the graph, eliminating the backdoor path from x to y, and allowing the identification of the direct effect of x on y with the regression coefficient of x^* on y, i.e, $\lambda_{xy} = \frac{\sigma_{x^*y}}{\sigma_{x^*x}}$ (Chen et al., 2016). This phenomenon can also be observed in the flow graph of Fig. 2d, shown in Fig. 2e. The covariance of x^* with y reads $\sigma_{x^*y} = \sigma_{xy} - \lambda_{zx}\sigma_{zy} = \epsilon_{xx}\lambda_{xy}$. That is, the operation to create x^* effectively removes the source edge λ_{zx} (red) from the flow graph. Next, dividing σ_{x^*y} by $\sigma_{x^*x} = \sigma_{xx} - \lambda_{zx}\sigma_{zx} = \epsilon_{xx}$ gives λ_{xy}.

2.2. Trek Systems & Determinants

Instrumental variables and instrumental sets can be generalized in the flow graph by exploiting properties of determinants of the minors of the covariance matrix (Sullivant et al., 2010). Denote by $\Sigma_{z_1z_2,x_1x_2}$ the minor of the covariance matrix with columns z_1, z_2 and rows x_1x_2. This gives us $\det \Sigma_{z_1z_2,x_1x_2} = \sigma_{z_1x_1}\sigma_{z_2x_2} - \sigma_{z_2x_1}\sigma_{z_1x_2}$. As

shown by Gessel & Viennot (1989), the value of this determinant can be read directly from the flow graph as the sum of non-intersecting sets of paths (i.e, paths which do not share any vertices) between z_1, z_2 and x'_1, x'_2, multiplied by the sign of their permutations (see Gessel & Viennot (1989) for details). This is due to the fact that terms that come from intersecting paths cancel out when computing the determinant, leaving only terms corresponding to non-intersecting paths. We therefore have $\det \Sigma_{z_1 z_2, x_1 x_2}$ as the product of non-intersecting paths between $z_1 \to x'_1$ and $z_2 \to x'_2$, with the non-intersecting paths between $z_1 \to x_2$ and $z_2 \to x_1$ subtracted out. If any such path set exists, the determinant can be shown to be generically non-zero.

For illustration, consider the flow graph of Fig. 2c, which is shown in Fig. 3. There is only one non-intersecting path set between $z_1 \to x_1$ and $z_2 \to x_2$, namely $\epsilon_{z_1 z_1} \lambda_{z_1 x_1}$ and $\epsilon_{z_2 z_2} \lambda_{z_2 x_2}$. Similarly, the non-intersecting path set between $z_1 \to x_2$ and $z_2 \to x_1$ consists of the paths $\epsilon_{z_1 z_1} \lambda_{z_1 x_2}$ and $\epsilon_{z_2 z_2} \lambda_{z_2 x_1}$. The determinant is given by

$$\begin{aligned} \det \Sigma_{z_1 z_2, x_1 x_2} = \; & (\epsilon_{z_1 z_1} \lambda_{z_1 x_1})(\epsilon_{z_2 z_2} \lambda_{z_2 x_2}) \\ & - (\epsilon_{z_1 z_1} \lambda_{z_1 x_2})(\epsilon_{z_2 z_2} \lambda_{z_2 x_1}) \end{aligned} \quad (2)$$

Weihs et al. (2018) realized that this property can be exploited for the identification of structural parameters. Solving for $\det \Sigma_{z_1 z_2, y x_2}$, the same non-intersecting paths exist between the sets as shown in Fig. 3, but all paths to x'_1 are now extended to y'. This yields

$$\det \Sigma_{z_1 z_2, y x_2} = \lambda_{x_1 y} \det \Sigma_{z_1 z_2, x_1 x_2} \quad (3)$$

allowing identification of $\lambda_{x_1 y} = \frac{\det \Sigma_{z_1 z_2, y x_2}}{\det \Sigma_{z_1 z_2, x_1 x_2}}$, which formally recovers Cramer's rule solution of Eq. (1) for $\lambda_{x_1 y}$.

2.3. Model Decomposition

One can also *decompose* a graph into smaller sub-graphs. First proposed by Tian (2005), the decomposition hinges upon the concept of a **c-component**, consisting of a set of variables connected through paths consisting solely of bidirected edges. Consider Fig. 2f. Here, x_1 has a bidirected edge to y, which in turn is connected to x_2 and w via bidirected edges. This makes $\{w, x_1, x_2, y\}$ a c-component. Likewise, $\{z_1, z_2\}$ is also a c-component.

Given a c-component, we can define a subgraph consisting of the nodes in the c-component and its direct parents, with all other variables and edges removed. This subgaph is called a "mixed component" of G.

Definition 2.2. *(Tian, 2005; Drton & Weihs, 2015) Given $G = (V, D, B)$, let $C_1, ..., C_k \subset V$ be the unique partitioning of V where $v, w \in C_i$ iff $\exists$ path from v to w composed only of bidirected edges, and let $V_i = C_i \cup Pa(C_1)$, $D_i = \{v \to w \in G | v \in V_i, w \in C_i\}$ and $B_i = \{v \leftrightarrow w \in G | v, w \in C_i\}$. Then $G_i = (V_i, D_i, B_i)$, $i = 1...k$ are the **mixed components** of G.*

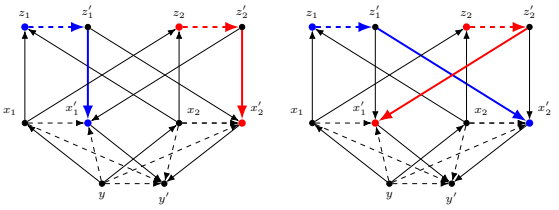

Figure 3. The flow graph of Fig. 2c, showing the two non-intersecting path sets from z_1, z_2 to x_1, x_2

The mixed component of $\{w, x_1, x_2, y\}$ in Fig. 2f is shown in Fig. 2g. Whereas existing efficient methods fail to identify $\lambda_{x_1 y}$ in Fig. 2f, it is easily identifiable in Fig. 2g. As shown by Tian (2005), this means the effect is also identifiable in the original graph.

3. Identification with the Auxiliary Cutset

In this section, we develop a polynomial-time algorithm that subsumes the state-of-the-art for efficient identification in linear SCM. We begin by defining a new type of auxiliary variable, which can help with the identification of new coefficients in the model. We then devise an identification criterion for *partial effects*, and show how to use it to efficiently create these new AVs. Finally, we show how our results can be used to recursively identify direct effects.

3.1. Auxiliary variables using total and partial effects

Standard auxiliary variables enable the use of previously identified direct effects to remove edges from the flow graph. In some cases, such as in Fig. 2d, this allows us to directly identify a target parameter, bypassing the need to search for a conditioning set. However, in many cases, auxiliary variables using only identifiable direct effects are not sufficient for this task.

An example of such a model is given in Fig. 2f. Here, although the generalized instrumental set $\{z_1, z_2\}$ *conditional* on w is sufficient for identifying $\lambda_{x_1 y}$ and $\lambda_{x_2 y}$, we cannot achieve the same result with AVs. While $z_2^* = z_2 - \lambda_{w z_2} w$ can be computed (because $\lambda_{w z_2}$ is identified), the AV $z_1^* = z_1 - \lambda_{w z_1} w - \lambda_{z_1 z_2} z_2$ cannot, since that requires the identification of both $\lambda_{w z_1}$ and $\lambda_{z_1 z_2}$. This suggests that there is something missing from AVs that rely solely on direct effects.

The source of the issue can be revealed in the mixed component of the model (Fig. 2g). Here, w is disconnected from z_1 and z_2, which becomes equivalent to the model in Fig. 2c. In the mixed component, $\lambda_{x_1 y}$ and $\lambda_{x_2 y}$ can be easily identified using an *unconditional* instrumental set $\{z_1, z_w\}$. This leads to the realization that it may not be necessary to identify the direct effects $\lambda_{w z_1}$ and $\lambda_{z_1 z_2}$ to

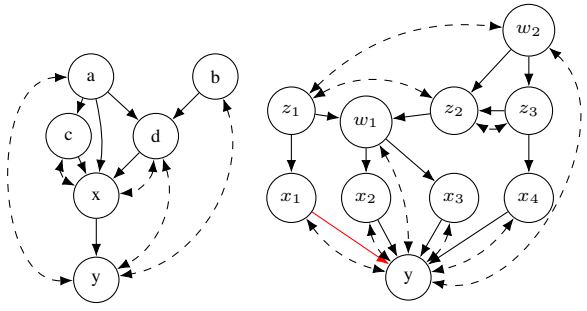

Figure 4. (a) A naïve application of total-effect AVs can't be used here. (b) Only ACID can solve for $\lambda_{x_1 y}$.

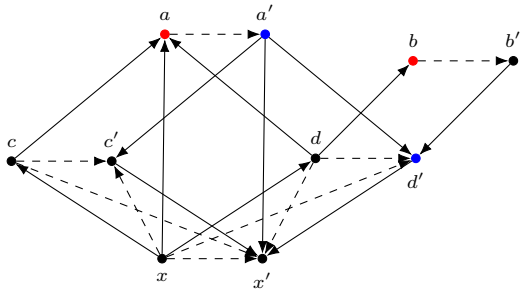

Figure 5. The flow graph of Fig. 4a, excluding y. To find the partial effects $\delta_{ax.d}$ and $\delta_{dx.a}$ (a', d' in blue), we can use the partial-effect instrumental set a, b (red).

disconnect z_1 from w—it suffices to identify the *total effect* of w on z_1. As it happens, this effect is easily computable with the regression coefficient of w on z_1, i.e, $\delta_{wz_1} = \frac{\sigma_{wz_1}}{\sigma_{ww}}$. We can now create a new type of AV, $z_1^\dagger = z_1 - \delta_{wz_1} w$, which indeed behaves as if z_1 had no path to w. Combining $z_1^\dagger$ with the standard AV z_2^* leads to a system of equations that can be solved for $\lambda_{x_1 y}$ and $\lambda_{x_2 y}$,

$$\sigma_{z_1^\dagger y} = \sigma_{z_1 y} - \delta_{wz_1}\sigma_{wy} = \lambda_{x_1 y}\sigma_{z_1^\dagger x_1} + \lambda_{x_2 y}\sigma_{z_1^\dagger x_2}$$
$$\sigma_{z_2^* y} = \sigma_{z_2 y} - \lambda_{wz_2}\sigma_{wy} = \lambda_{x_1 y}\sigma_{z_2^* x_1} + \lambda_{x_2 y}\sigma_{z_2^* x_2}$$

One needs to be careful when generalizing the results of this example—sometimes, a naïve subtraction of total effects can backfire. Consider, for instance, the model of Fig. 4a, and suppose we want to use x as an AV for the target query λ_{xy}. Note we can identify the total effects δ_{ax} and δ_{dx}. But also note that in the "naïve AV" $x^\ddagger = x - \delta_{dx} d - \delta_{ax} a$, subtracting out *both* total effects, *does not* remove the backdoor paths of x with y,

$$\sigma_{x^\ddagger y} = \sigma_{xy} - \delta_{ax}\sigma_{ay} - \delta_{dx}\sigma_{dy} = \lambda_{xy}\sigma_{x^\ddagger x} - \delta_{dx}\lambda_{ad}\epsilon_{ay}$$

This happens because the total effect of a already includes parts of the total effect of d, and thus when constructing the "naïve AV" $x^\ddagger$ we subtracted the path $\lambda_{ad}\lambda_{dx}$ twice.

One way to avoid this problem is to use only the portions of the effect of a on x that do not pass through d. That is, instead of subtracting out the total effect δ_{ax}, we subtract out the *partial effect* $\delta_{ax.d}$, leading to $x^\dagger = x - \delta_{dx} d - \delta_{ax.d} a$. Indeed, as desired, this removes all backdoor paths,

$$\sigma_{x^\dagger y} = \sigma_{xy} - \delta_{ax.d}\sigma_{ay} - \delta_{dx}\sigma_{dy} = \lambda_{xy}\sigma_{x^\dagger x}$$

In other words, by using only paths that do not intersect any other variable subtracted in the AV, we can avoid subtracting any path more than once, allowing us to effectively remove all of x's backdoor paths to both a and d at the same time. We formalize this idea in Theorem 3.1.

Theorem 3.1. *Given a variable x, and a subset $\mathbf{C}$ of the ancestors of x, the covariance of the auxiliary variable $x^\dagger = x - \sum_i \delta_{c_i x.C} \times c_i$ with variable v can be determined by the sum of paths from x to v' in G_{flow} with source edges $\lambda_{c_i d_j}$ removed where $c_i \in \mathbf{C}$ and $d_j \in An(x)$.*

Proof. Proofs are given in (Kumor et al., 2020). □

3.2. Instrumental sets for partial-effects

Theorem 3.1 gives us a principled way to incorporate knowledge of partial effects into the flow graph in order to help existing identification algorithms. A natural question now arises: how can we identify those partial effects? In this subsection, we demonstrate how a modified version of instrumental sets can solve this task. In particular, we exploit the same property that was used to identify a direct effect in the example of Eq. (3).

Continuing with the model of Fig. 4a, we want to identify $\delta_{dx.a} = \delta_{dx}$ and $\delta_{ax.d}$. To help with understanding the general approach, we will operate on the flow graph of Fig. 4a excluding y, shown in Fig. 5. We have placed a', d' (blue) in the sink nodes. Notice that the paths from a' to x' form δ_{ax}. All paths from a' to x' that do not intersect with d' form $\delta_{ax.d}$. Likewise, the paths from d' to x' form δ_{dx}, which in this case is the same as $\delta_{dx.a}$. Our goal is to exploit the non-intersection property of paths in the determinant of a trek system to automatically find all paths from a to x that do not pass through d.

Observe that a and b are two **candidate instruments** for x, that is, they are non-descendants of $\{x\} \cup Sib(x)$. Furthermore, all paths from the source nodes a and b (red) to x' in the flow graph cross either a' or d'. Computing the determinant between a, b and a', d' gives $\det \Sigma_{ab,ad} = (\epsilon_{aa})(\epsilon_{bb}\lambda_{bd})$, since there is only one valid path set, $a \to a'$ and $b \to d'$.

Next, replace a' with x' in the determinant. That is:

$$\begin{aligned}\det \Sigma_{ab,xd} &= (\epsilon_{aa}(\lambda_{ax} + \lambda_{ac}\lambda_{cx}))(\epsilon_{bb}\lambda_{bd}) \\ &= \delta_{ax.d} \det \Sigma_{ab,ad} \end{aligned} \tag{4}$$

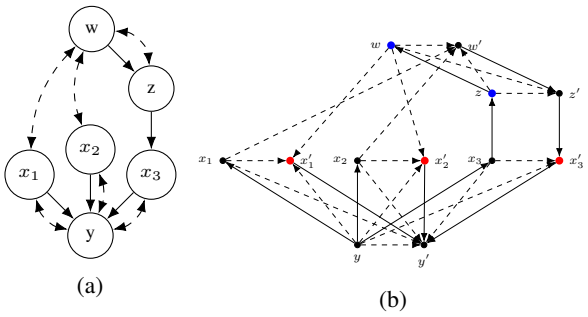

Figure 6. In this graph, w and z are both candidate instruments, but the min-cut between w, z (blue) and x'_1, x'_2, x'_3 (red) is w, z' - with w as a source node.

We have therefore found that $\delta_{ax.d} = \frac{\det \Sigma_{ab,xd}}{\det \Sigma_{ab,ad}}$. Similarly, $\delta_{dx.a} = \frac{\det \Sigma_{ab,ax}}{\det \Sigma_{ab,ad}}$. What we have created here is the analog of a standard instrumental set, which uses the set $\{a, b\}$ as instruments to identify the partial effects of a and d on x.

The formalization of instrumental sets for identifying partial effects is given in Definition 3.1 and Theorem 3.2.

Definition 3.1. *Given target node x, a set $C \subset An(x)$, and a set Z such that $|Z| = |C|$, then, if there is a non-intersecting path-set between source nodes of Z and sink nodes of C in G_{flow}, and all paths from source nodes Z to sink node x' cross through the sink nodes of C, the set Z is said to be a **Partial-Effect Instrumental Set** (PEIS) relative to C and x.*

Theorem 3.2. *If Z is a PEIS relative to C and x, then the partial effects of C on x are identified, and given by $\delta_{c_i x.C} = \frac{\det \Sigma_{Z,C^x_{-i}}}{\det \Sigma_{Z,C}}$, where C^x_{-i} is the ordered set C with c_i replaced by x.*

Whenever there exists a PEIS Z relative to C and x, we call C a **feasible ancestral cutset** of x, because C cuts x from Z in the flow graph. Finally, we would like to emphasize that, while in this work we mainly use the PEIS for the purpose of constructing AVs, this is a general criterion for finding partial effects, and can be used independently for other purposes.

3.3. Auxiliary cutset: "best" feasible ancestral cutset

In general, given a target node x for which we want to create an auxiliary variable $x^\dagger$, there will be multiple feasible ancestral cutsets we can choose from. For instance, in Fig. 4a, d alone is a feasible ancestral cutset (using b as an instrument). Clearly, however, an auxiliary variable constructed by subtracting the effect of d alone cuts x from strictly less variables than using $\{a, d\}$. Moreover, an $x^\dagger$ constructed in this way is not independent of y (a path through a still exists), and can no longer be used as an AV to identify λ_{xy}. It is useful, therefore, to define a notion of the *best* feasible ancestral cutset C for generating an auxiliary variable $x^\dagger$. This is called the **auxiliary cutset**.

Definition 3.2. *The **auxiliary cutset** (AC) is the feasible ancestral cutset C of x such that sink nodes of C intersect all paths from sink nodes of C' to x' for all feasible ancestral cutsets C' for x in G.*

Definition 3.2 ensures that the AC of x results in an auxiliary variable $x^\dagger$ that has all the removed ancestral paths of any other possible auxiliary varible $x^{\dagger'}$ removed. In other words, for every other feasible auxiliary variable $x^{\dagger'}$, we know that the $x^\dagger$ constructed using the AC is the "best", meaning that, if a path is removed using any other feasible ancestral cutset, it is also removed using the auxiliary cutset.

To find the AC, we follow a procedure similar to the one used in Kumor et al. (2019). The core idea can once again be demonstrated on Fig. 5. In this flow graph, only a and b are source nodes whose paths to x all go through the sink nodes of $Pa(x)$. This means that a and b are both "candidate instruments"—only candidate instruments can possibly be instruments of a feasible ancestral cutset. We then run a vertex min-cut algorithm (Picard & Queyranne, 1982) from $\{a, b\}$ to the sink nodes of parents of x, namely $\{c', d'\}$, and find the min-cut that is *closest* to x. In this case, the min-cut is a', d', meaning that $\{a, d\}$ is the auxiliary cutset of x.

By using the closest vertex min-cut C between candidate instruments Z and sink nodes of parents of x, we guarantee all the conditions of Definition 3.1 are satisfied automatically by Z' and C, with $Z' \subseteq Z$, except for the requirement that C consists entirely of sink nodes. An example where this is violated is given in Fig. 6—the min-cut there includes source node w. In such cases, we know that the associated node is not part of any possible AC, so we can remove it from candidacy, and rerun the min-cut algorithm. After removing w, and then z, we are left with no possible AC in Fig. 6. The general version of this procedure (which includes concepts from Section 3.4) is implemented in Algorithm 1, and always finds the AC if it exists.

Theorem 3.3. *If there exists a feasible ancestral cutset for x in G, then the AC exists, and is found by Algorithm 1.*

3.4. Putting everything together: the *ACID* Algorithm

The final question remaining is: how can we use this newly generated AV $x^\dagger$ for identification? We want to both use it recursively in Algorithm 1, and within our identification algorithm (IC), to identify direct effects. When the set of candidate instruments consists of original variables we can easily find the AC using the standard flow graph, as it was done in the example in Fig. 5. However, once the candidate instruments have their own auxiliary cutsets, with each candidate's cutset being specific to that candidate, we run into difficulties trying to encode non-intersecting paths.

Algorithm 1 - AC: is given a graph, target vertex x, a set of candidate instruments (which can themselves be AVs), a set of identified structural parameters, and returns the Auxiliary Cutset for x

Input: G, x, Z, Λ
$G_{hf} \leftarrow$ AUXHALFFLOWGRAPH(G, x, Λ)
$C \leftarrow \emptyset$
repeat
  $Z \leftarrow \{z \in Z :$ UNREMOVEDANCESTORS$(z) \cap C = \emptyset\}$
  $C \leftarrow$ vertex min-cut closest to x between $Pa(x)$ and Z
until UNREMOVEDANCESTORS$(Z) \cap C = \emptyset$
$Z' \leftarrow \{z_i \in Z : z_i$ has non-zero flow to $c_i \in C\}$
return (Z', C)

To understand why, we will use the concept of "unremoved ancestors"

Definition 3.3. *Let y be a target node with auxiliary cutset C. The **unremoved ancestors** of y are all $v \in An(y)$ such that there exists a directed path from v to y in G that does not cross any $c \in C$.*

When using multiple AVs at once, e.g, $z_1^\dagger$ and $z_2^\dagger$, where each AV has source-paths in the flow graph to its own unremoved ancestors, some of which might be shared, it is unclear how to efficiently find a path set from both AVs at once that does not intersect. Theorem 3.4 shows we do not need to encode paths in the source nodes at all.

Theorem 3.4. *Let A be the unremoved ancestors of the AV $x^\dagger$. If $x^\dagger$ is a candidate instrument for y, then all elements of the AV $a^\dagger$ of $a \in A$ are also candidate instruments for y.*

Therefore, since all the unremoved ancestors of a candidate instrument are candidate instruments themselves, any candidate with a path in the source nodes can be switched with the candidate at which the path crosses over to the sink nodes. This means we only need to encode edges from source nodes to sink nodes, which can be done with the Auxiliary Half-Flow Graph (Definition 3.4).

Definition 3.4. *The **auxiliary half-flow graph** of $G = (V, D, B)$ given target node y and set of identified edges Λ is the graph with vertices $V \cup V'$ containing the edges:*

- *$i' \to j'$ with weight λ_{ij} if $i \to j \in V$ and $j \neq y$*
- *$i' \to y'$ with weight λ_{iy} if $i \to y \in V$ and $\lambda_{iy} \notin \Lambda$*
- *$i \to i'$ with weight ϵ_{ii} for all $i \in V$*
- *$i \to j'$ with weight ϵ_{ij} if $i \leftrightarrow j \in B$*

This graph is referred to as G_{hf}. The nodes without $'$ are called "source" nodes, and the nodes with $'$ are called "sink" or "bottom" nodes.

Algorithm 2 - ACID: Given a graph, returns a set of identifiable structural parameters.

Input: G
$\Lambda_{id} \leftarrow \emptyset$
$v^\dagger = v \; \forall$ vertices $v \in G$
repeat
  for all vertices $v \in G$ in topological order **do**
    $Z \leftarrow \{z^\dagger :$ UNREMOVEDANCESTORS$(z^\dagger) \cap (Sib(y) \cup \{y\}) = \emptyset, \; \forall z \in G\}$
    $G_{hf} \leftarrow$ AUXHALFFLOWGRAPH(G, x, Λ_{id})
    $\Lambda_{id} \leftarrow \Lambda_{id} \cup IC(G_{hf}, v, Z)$
    $v^* = v - \sum_{\lambda_{av} \in \Lambda_{id}} \lambda_{av}$
    $(Z', C) \leftarrow AC(G, v, Z, \Lambda_{id})$
    $v^\dagger = v^* - \sum_{c_i \in C} \frac{\det \Sigma_{Z', C_i^{v^*}}}{\Sigma_{Z', C}} c_i$
  end for
until no change in this iteration
return Λ_{id}

This completes the tools needed to specify the ACID algorithm (Algorithm 2), which internally uses a version of the IC algorithm (Kumor et al., 2019) adapted to make use of partial-effect AVs and Half-Flow graphs.[1]

ACID unifies the state-of-the-art for identification in linear SCMs. Moreover, the AC's ability to block certain ancestors turns out to be the missing piece needed for auxiliary variable methods to finally overtake methods based on conditioning. Methods built upon conditioning such as the gIS and qAVS have undetermined complexity, with several NP-Hardness results for similar methods. ACID is the first efficient identification algorithm that subsumes these approaches, obviating the discussion of their computational complexity.

Theorem 3.5. *If λ_{ab} is identifiable with either the cAV, IC, or qAVS criteria, then it is identifiable with ACID.*

4. Decomposition of total effects

The ACID algorithm identifies individual direct effects. It does not include total effects, which play an important role in virtually all causal inference tasks, such as policy-making, model testing, z-identification, and sensitivity analysis (Pearl, 2000; Bareinboim & Pearl, 2012; Chen et al., 2017; Cinelli et al., 2019; Lee et al., 2019; Cinelli & Hazlett, 2020). Unlike direct effects, the identification of total effects in linear models has not received as much attention and existing approaches fall broadly into two categories.

The first approach is to appeal to the foundational methods of identification, such as the instrumental variable, the front-door, or the back-door criterion (or, more generally,

[1]For details of the modified IC, refer to the appendix.

the do-calculus) (Pearl, 2000; Tian, 2004). While sound, these methods ignore recent advances in the linear identification literature. A second approach is to decompose the total effect of x on y into the sum of structural parameters along all directed paths from x to y, and use state-of-the-art identification algorithms for direct effects to identify *all* structural parameters along these paths. This is sub-optimal, as it misses cases in which the total effect is identifiable, but some individual parameters are not.

To bridge the gap between these two extremes, we derive a decomposition of total effects that relies on the identification of only some of the direct effects of which it is composed. The method makes use of ancestor and c-component decompositions to recursively break the total effect into a set of partial effects and direct effects, which can then be attacked with current identification algorithms for linear models.

Suppose our target query is the total effect δ_{xy} in Fig. 7a. This effect is *not* identifiable non-parametrically, as there are bidirected edges between x and its children. Second, an approach based on current state-of-the-art methods that relies on identifying all direct effects of δ_{xy} would equally fail, since no existing method can identify λ_{ce}.

We begin by defining the "top boundary" of a mixed component as the parents of variables in a c-component that are in other c-components.

Definition 4.1. *Given graph G, with c-components $C_1, ..., C_k$, the **top boundary** $Tb(C_i)$ of the c-component C_i is defined as $Tb(C_i) = Pa(C_i) \setminus C_i$.*

The concept of top boundary is useful for two reasons: first, we can always identify the "total effect" in the mixed component G' of its top boundary on any other node in the mixed component (this "total effect" in G' may correspond to a partial effect in the original model G); second, the total effect of x on y can be decomposed as the sum of the total effect of x in the nodes of the top boundary (in G) times the "total effect" of the top boundary nodes in y in the mixed component. This is formalized in Theorem 4.1.

Theorem 4.1. *Let $G_{An(y)}$ be the graph G with the non ancestors of y removed. Let C_y be the c-component of y in $G_{An(y)}$ and G' its corresponding mixed component. Then, if $x \in C_y$, the total effect of x on y, δ_{xy}, can be decomposed as, $\delta_{xy} = \delta'_{xy} + \sum_{b \in Tb(C_y)} \delta_{xb}\delta'_{by}$ otherwise, if $x \notin C_y$, we have, $\delta_{xy} = \sum_{b \in Tb(C_y)} \delta_{xb}\delta'_{by}$ where $Tb(C_y)$ is the top boundary of the c-component C_y and δ'_{by} is the total effect of node $b \in Tb(C_y)$ on y in the mixed component G'. Moreover, all δ'_{by} for $b \in Tb(C_y)$ are identified.*

The recursive application of this idea enables us to iteratively identify parts of the total effect via a combination of ancestral and c-component decompositions, leaving only a portion of the path to be identified using algorithms specialized for direct effects. In our example, note that h is not an ancestor of y, therefore $G_{An(y)}$ does not include h. This allows us to decompose the pruned graph into the the mixed component G' of the c-component $\{y, g\}$. Since the total effect of x on y needs to necessarily pass through the top boundary of G', we have that, as per Theorem 4.1, the total effect can be decomposed as $\delta_{xy} = \delta_{xe}\delta'_{ey} + \delta_{xd}\delta'_{dy} + \delta_{xf}\delta'_{fy}$, and all direct effects starting from the top boundary in the mixed component (δ') can be identified. The query δ_{xy} is then broken down into three smaller subqueries, δ_{xe}, δ_{xd}, δ_{xf}.

Now we can apply Theorem 4.1 for each of the remaining queries (shown in Fig. 7b). Pruning the non-ancestors of f leaves us with just f, and $\delta_{xf} = 0$. Looking at e, note that f is not its ancestor, and the c-component decomposition allows identification of δ'_{ae} and δ'_{be}, with resulting subqueries δ_{xa} and δ_{xb}. Applying the same logic to d leads us to identify δ'_{ad} and δ'_{bd}, with identical remaining subqueries. This leaves only two directed edges left to be identified $\delta_{xa} = \lambda_{xa}$ and $\delta_{xb} = \lambda_{xb}$. We have thus reduced the identification of the total effect of δ_{xy} to the identification of λ_{xa} and λ_{xb} only, both of which can be solved using f as an instrumental variable. The full algorithm for the total effect decomposition is given in Algorithm 3, which given a target total-effect δ_{xy}, returns a set of direct effects that are sufficient for identifying the desired quantity.

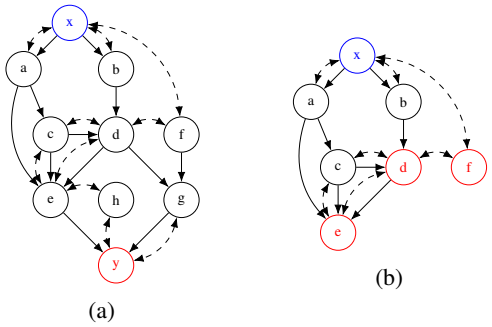

Figure 7. In (a), we can reduce δ_{xy} into queries on $\delta_{xe}, \delta_{xd}, \delta_{xf}$ in (b), and then to λ_{xa} and λ_{xb} by iteratively decomposing the model.

Algorithm 3 - TED: Given a graph and a target total-effect δ_{xy}, returns a set of direct effects that suffice to identify δ_{xy}

Input: G, δ_{xy}
$G_{An} \leftarrow An(y)$
$G' \leftarrow$ MIXEDCOMPONENT(G_{An}, y)
$B \leftarrow$ TOPBOUNDARY$(G') \cap An(y, G')$
if $x \in G'$ and $x \notin B$ **then**
  return $\{\lambda_{ab} : \forall \lambda_{ab} \in \delta'_{xy}\} \cup \bigcup_{b \in B}$ TED(G, δ_{xb})
else
  return $\bigcup_{b \in B}$ TED(G, δ_{xb})
end if

5. Conclusion

We developed an efficient algorithm for linear identification that subsumes the current state-of-the-art, unifying disparate approaches found in the literature. In doing so, we also introduced a new method for identification of partial effects, as well as a method for exploiting those partial effects via auxiliary variables. Finally, we devised a novel decomposition of total effects allowing previously incompatible methods to be combined, leading to strictly more powerful results.

Acknowledgements

Kumor and Bareinboim are supported in parts by grants from NSF IIS-1704352 and IIS-1750807 (CAREER).

References

Bardet, M. and Chyzak, F. On the complexity of a gröbner basis algorithm. In *Algorithms Seminar*, pp. 85–92, 2005.

Bareinboim, E. and Pearl, J. Causal inference by surrogate experiments: z-identifiability. *Uncertainty in Artificial Intelligence*, 2012.

Bareinboim, E. and Pearl, J. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113:7345–7352, 2016.

Bowden, R. J. and Turkington, D. A. *Instrumental Variables*. Number no. 8 in Econometric Society Monographs in Quantitative Economics. Cambridge University Press, Cambridge [Cambridgeshire] ; New York, 1984. ISBN 978-0-521-26241-5.

Brito, C. and Pearl, J. Generalized instrumental variables. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pp. 85–93. Morgan Kaufmann Publishers Inc., 2002.

Chen, B. Identification and Overidentification of Linear Structural Equation Models. *Advances in Neural Information Processing Systems 29*, pp. 1579–1587, 2016.

Chen, B., Pearl, J., and Bareinboim, E. Incorporating Knowledge into Structural Equation Models Using Auxiliary Variables. *IJCAI 2016, Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 7, 2016.

Chen, B., Kumor, D., and Bareinboim, E. Identification and model testing in linear structural equation models using auxiliary variables. *International Conference on Machine Learning*, pp. 757–766, 2017.

Cinelli, C. and Hazlett, C. Making sense of sensitivity: extending omitted variable bias. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82 (1):39–67, 2020.

Cinelli, C., Kumor, D., Chen, B., Pearl, J., and Bareinboim, E. Sensitivity analysis of linear structural causal models. *International Conference on Machine Learning*, 2019.

Drton, M. and Weihs, L. Generic Identifiability of Linear Structural Equation Models by Ancestor Decomposition. *arXiv:1504.02992 [stat]*, April 2015.

Fisher, F. M. *The Identification Problem in Econometrics*. McGraw-Hill, 1966.

Foygel, R., Draisma, J., and Drton, M. Half-trek criterion for generic identifiability of linear structural equation models. *The Annals of Statistics*, pp. 1682–1713, 2012.

García-Puente, L. D., Spielvogel, S., and Sullivant, S. Identifying causal effects with computer algebra. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pp. 193–200, 2010.

Gauss, C. Theoria motus, corporum coelesium, lib. 2, sec. iii, perthes u. *Besser Publ*, pp. 205–224, 1809.

Gessel, I. M. and Viennot, X. Determinants, paths, and plane partitions. *preprint*, 1989.

Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

Kumor, D., Chen, B., and Bareinboim, E. Efficient identification in linear structural causal models with instrumental cutsets. In *Advances in Neural Information Processing Systems*, pp. 12477–12486, 2019.

Kumor, D., Cinelli, C., and Bareinboim, E. Efficient identification in linear structural causal models with auxiliary cutsets. Technical Report R-56, 2020. https://causalai.net/r56.pdf.

Lee, S., Correa, J. D., and Bareinboim, E. General identifiability with arbitrary surrogate experiments. In *Proceedings of the Thirty-Fifth Conference Annual Conference on Uncertainty in Artificial Intelligence*, Corvallis, OR, 2019. AUAI Press.

Legendre, A. M. *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot, 1805.

Pearl, J. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

Picard, J.-C. and Queyranne, M. On the structure of all minimum cuts in a network and applications. *Mathematical Programming*, 22(1):121–121, December 1982. ISSN 1436-4646. doi: 10.1007/BF01581031.

Spirtes, P., Glymour, C. N., and Scheines, R. *Causation, prediction, and search*. MIT press, 2000.

Stigler, S. M. *The history of statistics: The measurement of uncertainty before 1900*. Harvard University Press, 1986.

Sullivant, S., Talaska, K., and Draisma, J. Trek separation for Gaussian graphical models. *The Annals of Statistics*, pp. 1665–1685, 2010.

Tian, J. Identifying linear causal effects. In *AAAI*, pp. 104–111, 2004.

Tian, J. Identifying direct causal effects in linear models. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, pp. 346. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

Tian, J. A Criterion for Parameter Identification in Structural Equation Models. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, pp. 8, Vancouver, BC, Canada, 2007.

Van der Zander, B. and Liskiewicz, M. On Searching for Generalized Instrumental Variables. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS-16)*, 2016.

Van der Zander, B., Textor, J., and Liskiewicz, M. Efficiently Finding Conditional Instruments for Causal Inference. *IJCAI 2015, Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2015.

Weihs, L., Robinson, B., Dufresne, E., Kenkel, J., McGee II, K. K. R., Reginald, M. I., Nguyen, N., Robeva, E., and Drton, M. Determinantal generalizations of instrumental variables. *Journal of Causal Inference*, 6(1), 2018.

Wright, P. G. *Tariff on Animal and Vegetable Oils*. Macmillan Company, New York, 1928.

Wright, S. Correlation and causation. *J. agric. Res.*, 20: 557–580, 1921.