https://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Articles in Advance, pp. 1–20
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Discovering Causal Models with Optimization: Confounders, Cycles, and Instrument Validity

Frederick Eberhardt,[a] Nur Kaynar,[b,*] Auyon Siddiq[c]

[a] Division of Humanities and Social Sciences, California Institute of Technology, Pasadena, California 91125; [b] Samuel Curtis Johnson Graduate School of Management, Cornell University, Ithaca, New York 14853; [c] Anderson School of Management, University of California, Los Angeles, Los Angeles, California 90095

*Corresponding author

Contact: fde@caltech.edu (FE); nur.kaynar@cornell.edu, https://orcid.org/0009-0004-5544-0596 (NK); auyon.siddiq@anderson.ucla.edu, https://orcid.org/0000-0003-2977-5558 (AS)

Received: June 23, 2021
Revised: March 9, 2023; October 8, 2023
Accepted: December 19, 2023
Published Online in Articles in Advance: July 17, 2024

https://doi.org/10.1287/mnsc.2021.02066

Abstract. We propose a new optimization-based method for learning causal structures from observational data, a process known as *causal discovery*. Our method takes as input observational data over a set of variables and returns a graph in which causal relations are specified by directed edges. We consider a highly general search space that accommodates latent confounders and feedback cycles, which few extant methods do. We formulate the discovery problem as an integer program, and propose a solution technique that exploits the conditional independence structure in the data to identify promising edges for inclusion in the output graph. In the large-sample limit, our method recovers a graph that is (Markov) equivalent to the true data-generating graph. Computationally, our method is competitive with the state-of-the-art, and can solve in minutes instances that are intractable for alternative causal discovery methods. We leverage our method to develop a procedure for investigating the validity of an instrumental variable and demonstrate it on the influential quarter-of-birth and proximity-to-college instruments for estimating the returns to education. In particular, our procedure complements existing instrument tests by revealing the precise causal pathways that undermine instrument validity, highlighting the unique merits of the graphical perspective on causality.

History: Accepted by J. George Shanthikumar, data science.
Supplemental Material: The online appendix and data files are available at https://doi.org/10.1287/mnsc.2021.02066.

Keywords: programming: integer • networks-graphs: theory • economics: econometrics • statistics

1. Introduction

Establishing causality is central to empirical research in the natural and social sciences. While randomized experiments are often considered the gold standard for determining causal relations, they come with substantial limitations: The experimenter has to be able to fully control the treatment, or adjust for the failure to do so. This often implies that the experiments have to be conducted in artificial environments with small sample sizes, undermining their validity. Further, some interventions are very costly to perform, and some are unethical. Consequently, it is often desirable to learn as much as possible about underlying causal relations from observational data alone, without performing experiments.

Understanding causal relations from data can be naturally framed as two distinct but complementary kinds of inference. The first is *causal inference*, which typically seeks to estimate the causal *effect* (i.e., sign and magnitude) of a treatment or intervention on a given outcome, often from observational data. These methods are typically built upon assumptions regarding the underlying causal *structure* (i.e., whether a cause-effect relationship exists at all between two variables, and in which direction) over the variables of interest. Although these assumptions are necessary for identifying causal effects, it is well known that misspecifications of causal structure can lead to biased estimates and erroneous conclusions. Further, while in some cases the relevant causal structures may follow from contextual knowledge, in others they may be unknown or only partially known, which can undermine the validity of the obtained estimates. This naturally leads to the second kind of inference: How might we learn causal structures directly from the data in the first place? This process, which is the focus of our paper, is known as *causal discovery*.

The framework of causal graphical models (Pearl 2000, Spirtes et al. 2000), discussed in more detail in Section 2, enables inference of causal structure by providing a precise mathematical representation of causal relations (in terms of a directed graph) and the observed data (in terms of a probability distribution associated with the graph). Using this framework, a variety of causal discovery methods have been

developed to infer underlying causal structures from observational data. With a few important exceptions, these methods have relied on two restrictive assumptions, which limit their practical relevance. The first is the absence of latent confounders—referred to as *causal sufficiency*—which means that there are no unmeasured common causes of the measured variables. The second is that there is no feedback, meaning the causal structures can be represented by directed *acyclic* graphs (DAGs). Both of these assumptions can only rarely be justified in practice.

Our main contribution is a new optimization-based method for causal discovery that allows for both unmeasured confounders *and* feedback cycles. Our method takes as input observational data over a set of variables, and returns a graph in which causal relations are specified by directed edges. There are very few extant discovery methods that consider this extremely general search space, and those that do, do not scale well. In contrast, our approach allows us to solve in minutes instances that are outright intractable for recently proposed methods. At a high level, the efficiency of our approach is due to a solution technique that exploits the conditional (in)dependence structure in the data to detect "promising" candidate edges in the underlying graph, which are then assembled into a causal graph by an optimization model. Our main theoretical result is to show that this technique asymptotically recovers a graph that is equivalent to the true, data-generating graph.

The generality of our method combined with its computational efficiency greatly expands the practical relevance of causal discovery to empirical research. We demonstrate this by using our method to develop a graphical procedure for investigating the validity of instrumental variables, which are widely used to estimate causal effects in the presence of unmeasured confounding. In particular, we apply our procedure to the instruments from the seminal papers on the returns to education by Angrist and Krueger (1991) and Card (1993). We find that the causal structures uncovered by our method are consistent with the literature, and that our results are qualitatively consistent with the well-known test for instrument validity proposed by Kitagawa (2015).

The remainder of the paper is organized as follows. Section 2 reviews foundational concepts in causal discovery, which a familiar reader may skip. Section 3 presents an optimization model for causal discovery with latent confounders and feedback cycles. Section 4 develops a solution technique for the model. Section 5 proposes a test for instrument validity within our framework. Section 6 concludes.

2. Causal Graphical Models

In the framework of causal graphical models (Pearl 2000, Spirtes et al. 2000), causal relations are represented by a directed graph $\mathcal{G} = (V, E)$ over a set of nodes V, where the edge set E contains directed edges, representing a direct (relative to V) causal effect of one node on another. We will first introduce the theory and notation using *directed acyclic graphs* (DAGs), as they permit the most intuitive explanation and the simplest causal interpretation (Section 2.1). We then extend these concepts to include graphs that contain cycles and unobserved confounding variables (Section 2.2).

2.1. Acyclic Models Without Unmeasured Confounding

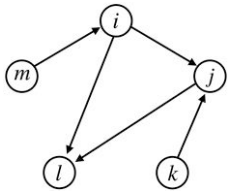
In acyclic causal models without unmeasured confounding, the graph $\mathcal{G} = (V, E)$ over a set of nodes V contains at most one directed edge between any pair of nodes. We define an edge $e \in E$ as a triple (i, t, j) with $i, j \in V$, $i \neq j$, and $t \in \{\rightarrow, \leftarrow\}$.[1] Central to our algorithm is the notion of a *path* between two variables:

Definition 1 ((undirected) Path). Given a node set V and an edge set E, we define a path p_{ij} from node i to node j with $i, j \in V, i \neq j$, as a sequence of edges $p_{ij} = (e_1, \ldots, e_\ell)$ such that $e_k \in E$ for all $1 \leq k \leq \ell$, e_1 starts with node i, e_ℓ ends with node j, consecutive edges are connected, and nodes on the path do not repeat (other than as start- and endpoint of consecutive edges).

A *directed path* from i to j is then a path where all edges point toward j. Any node connected by a directed path from i is a *descendant* of i, any node connected by a directed path to i is an *ancestor* of i. *Parents* and *children* of a node i are the direct causes and effects, respectively, of i in $\mathcal{G}$. A *directed acyclic graph* (DAG) is a directed graph in which there is no pair of distinct nodes (i, j) such that there is a directed path from i to j and an edge $j \rightarrow i$. We say that a node i is a *collider* on a path if its adjacent edges point into i ($\rightarrow i \leftarrow$). A *noncollider* on a path is a node i that is either a *mediator* ($\rightarrow i \rightarrow$) or a *common cause* ($\leftarrow i \rightarrow$). For example, in Figure 1, node j is a collider on the (undirected) path $i \rightarrow j \leftarrow k$ and a mediator on the path $i \rightarrow j \rightarrow l$, and node i is a common cause on the (undirected) path $l \leftarrow i \rightarrow j$.

In causal modeling, a DAG $\mathcal{G}$ is associated with a probability distribution $P_{\mathcal{G}}(V)$, which describes causal relations over the set of nodes V.[2] A standard assumption is that the distribution is generated by the graph structure

Figure 1. Example Directed Acyclic Graph (DAG)



such that it factorizes: $P_{\mathcal{G}}(V) = \prod_{i \in V} P_{\mathcal{G}}(i|Pa(i))$, where $Pa(i)$ are the parents of node i in $\mathcal{G}$ (Spirtes and Zhang 2016, Eberhardt 2017). Based on the connection between the causal structure and the resulting data distribution, many causal discovery algorithms, including the one we present here, exploit the independence structure seen in the data to infer the underlying causal relations. One of the central concepts required for this inference is the notion of *d-separation* (Geiger et al. 1990), which can be thought of as the graphical counterpart to probabilistic independence. It is based on the notion of a *blocked path*:

Definition 2 (Blocked Paths). A path between nodes i and j is *unblocked* with respect to a set of nodes C if every collider k on the path is in C or has a descendant in C, and no noncolliders on the path are in C. Otherwise the path is *blocked* with respect to C (Pearl 2000).

Definition 3 (d-Separation). Two nodes i and j are *d-separated* with respect to a conditioning set C (denoted by $i \perp j|C$) if all paths between them are blocked, otherwise they are *d-connected* given C (denoted by $i \not\perp j|C$) (Pearl 2000).

To illustrate the definitions above, note that there are two paths between m and l in Figure 1: $m \to i \to l$ and $m \to i \to j \to l$. By Definition 2, both of these paths are unblocked with respect to the empty conditioning set $C = \emptyset$, which by Definition 3 implies that m and l are d-connected with respect to $C = \emptyset$. Now consider the conditioning set $C = \{i\}$. By Definition 3, conditioning on node i blocks these paths because node i is a noncollider on both of these paths. Since there does not exist an unblocked path between nodes m and l with respect to the conditioning set $C = \{i\}$, it follows that m and l are d-separated with respect to the conditioning set $C = \{i\}$.

To establish a correspondence between (conditional) d-separation and (conditional) independence, two key assumptions are commonly employed: the *causal Markov* principle and the *faithfulness* condition. These assumptions allow us to relate the notation of d-separation ($\perp$) to probabilistic independence ($\perp\!\!\!\perp$).

If node i is d-separated from node j given conditioning set C in graph $\mathcal{G} = (V, E)$ with $i, j \in V$ and $C \subseteq V \setminus \{i, j\}$, then i is probabilistically independent of j given C in the distribution over the graph $P_{\mathcal{G}}(V)$:

$$i \perp j|C \text{ in } \mathcal{G} \Rightarrow i \perp\!\!\!\perp j|C \text{ in } P_{\mathcal{G}}(V). \tag{1}$$

Assumption 1 (Faithfulness). *If variable i is probabilistically independent of variable j given conditioning set C in the distribution over the graph $P_{\mathcal{G}}(V)$, then i is d-separated from j given C in graph $\mathcal{G} = (V, E)$:*

$$i \perp\!\!\!\perp j|C \text{ in } P_{\mathcal{G}}(V) \Rightarrow i \perp j|C \text{ in } \mathcal{G}. \tag{2}$$

The (global) causal Markov condition, as we have stated it here, follows from how we have defined the probability distribution in terms of the causal structure (Pearl 2000). In contrast, the faithfulness condition, which is the converse to the Markov condition, represents an additional assumption, as it ensures that an independence in the data are actually due to a d-separation (rather than, for example, two causal pathways cancelling each other out (Spirtes et al. 2000, Uhler et al. 2013). Together, the causal Markov and faithfulness conditions provide a tight correspondence between (conditional) probabilistic independence and (conditional) d-separation.

Remark 1. Under the causal Markov and faithfulness conditions, a (conditional) independence in $P_{\mathcal{G}}(V)$ is present if and only if there is a corresponding (conditional) d-separation in DAG $\mathcal{G}$.

This correspondence is the basis for many causal discovery methods, as one can now use the independence structure in the data to constrain the graph structure.

2.2. Extension to Cyclic Models with Latent Confounding

We introduced the key concepts in the context of directed acyclic graphs (DAGs). In the remainder of the paper, we focus on a more general class of graphs that permit cycles and can represent confounding due to unobserved variables. For such graphs, many of the key ideas above can be generalized, but they require much more book-keeping and are much less intuitive. We briefly outline the required adjustments here.

A cyclic model, as its name suggests, permits feedback cycles in the causal structure. In this setting, the edge set E in $\mathcal{G} = (V, E)$ may contain an edge in each direction between a pair of nodes.[3] These cycles do not represent backward-in-time causation, but should instead be understood as shorthand notation for causal feedback that is unraveled over time; for example, $i_t \to j_{t+1} \to i_{t+2}$. One of the simplest and most well-studied cyclic causal models is the linear Gaussian cyclic model given by $\mathbf{x}(t) = \mathbf{B}\mathbf{x}(t-1) + \boldsymbol{\epsilon}$, where $\mathbf{x}$ is a vector representation of the variables in V, $\boldsymbol{\epsilon}$ is a vector of independent errors, and $\mathbf{B}$ is a square matrix representing the (possibly cyclic) causal effects among the variables (Hyttinen et al. 2012). Under appropriate conditions, the model converges to an equilibrium, which allows cycles such as $i_t \to j_{t+1} \to i_{t+2}$ to be represented more simply without time indices as $i \rightleftarrows j$.[4] For this type of linear Gaussian cyclic model, the Markov and faithfulness conditions still imply the correspondence between (conditional) d-separation and (conditional) independence (Remark 1). However, unlike the acyclic case, the correspondence between d-separation and probabilistic independence does not hold in general in cyclic models.

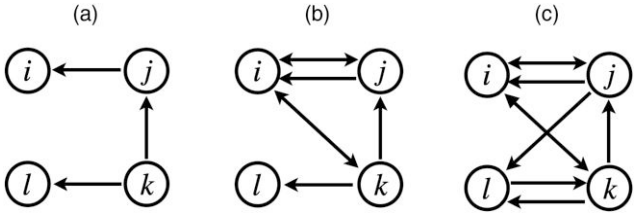
Remark 2. In the cyclic case, the correspondence between d-separation and probabilistic independence holds for linear Gaussian causal models, but not in general (Spirtes 1995).

We will restrict consideration of *cyclic* models to the linear Gaussian case in order to utilize the correspondence described in Remark 1.[5] In the *acyclic* case, our results are not restricted to a particular parameterization.

To represent confounding between a pair of variables (i, j) due to an unobserved common cause c, the graphical framework is extended to include the bidirected edge $i \leftrightarrow j$ (see Figure 2, (b) and (c)). Here, $i \leftrightarrow j$ is shorthand for $i \leftarrow c \rightarrow j$, where c is an unobserved variable ($c \notin V$). The graph $\mathcal{G} = (V, E)$ then consists of a set of variables V and edges E such that every pair (i, j) is permitted to contain directed edges ($\rightarrow$, $\leftarrow$), possibly in both directions, and a bidirected edge ($\leftrightarrow$) to represent unmeasured confounding.[6] Definition 1 (of paths) can be extended to include bidirected edges between variables, that is, $t \in \{\rightarrow, \leftarrow, \leftrightarrow\}$. See Section EC.1 of the electronic companion for formalization. Colliders remain defined as before, but now they can also arise from bidirected edges incident on the "colliding" variable.

The model class that includes bidirected edges but disallows cycles is often referred to as *acyclic directed mixed graphs* (ADMGs) (Figure 2(b)). Our focus will be on the general model class of *directed mixed graphs* (DMGs), where both bidirected edges and cycles are allowed (Figure 2(c)). D-separation can be naturally extended to DMGs, one just has to keep track of a larger set of possible edges, since any pair of variables can now be connected by three different edge types. In the cyclic linear Gaussian model described above, confounding can be represented using a nondiagonal covariance matrix for the error terms, resulting in correlated errors. For such linear Gaussian models with correlated errors the correspondence between d-separation and probabilistic independence (Remark 2) still holds (Spirtes 1995).

Figure 2. Example Graphs with Latent Confounders and Cycles



Notes. Example graphs: (a) A directed acyclic graph (DAG). (b) The more general acyclic directed mixed graph (ADMG), which does not contain cycles, but contains bidirected edges to represent unmeasured confounding. (c) A directed mixed graph (DMG) that allows for both cycles and confounding.

2.3. Constraint-Based Causal Discovery

Our proposed method belongs to a class of causal discovery algorithms known as *constraint-based* methods (see Maathuis et al. (2010), Spirtes and Zhang (2016), and Eberhardt (2017) for reviews). These methods involve performing conditional independence tests on the data to construct conditional (in)dependence "constraints", that are used to search for a causal graph that satisfies these constraints to the extent possible. Here, each input constraint is the statement $i \perp\!\!\!\perp j | C$ or $i \not\perp\!\!\!\perp j | C$ for some $i, j \in V$ and $C \subset V \setminus \{i, j\}$, which implies a d-separation or d-connection that the output graph must satisfy (by Remark 1).

One of the first and best-known constraint-based methods is the PC-algorithm by Spirtes et al. (2000), which is restricted to searching for the equivalence classes of DAGs. Alternative methods generalize the search space to acyclic graphs with unmeasured common causes (Spirtes et al. 2000) and cyclic causal models (albeit without unmeasured confounding) (Richardson 1996). A number of variants of these methods have been developed in the literature with the aim of improving computational efficiency or reliability (Colombo et al. 2012, Teramoto et al. 2014). Constraint-based methods have been shown to be asymptotically correct for their respective background assumptions, meaning in the large-sample limit they discover the true data-generating graph up to an equivalence class (see Section 3.2 for details) (Spirtes et al. 2000, Zhang and Spirtes 2002, Solus et al. 2021).

An advantage of constraint-based causal discovery is that it allows the user to completely separate the statistical challenge of establishing the (conditional) independence constraints from the combinatorial inference of finding graphs that are consistent with them. Thus, one can choose independence tests that are suitable for the particular domain and adopt their preferred correction method for multiple hypothesis testing. Such decisions will be informed by the sample size, the number and dimensionality of the variables, whether the variables are categorical or continuous, or what assumptions one is willing to make about the parametric form of the causal relations.

Our paper is most closely related to constraint-based methods that allow for both cycles and unmeasured confounders. These methods encode d-separation constraints obtained from independence tests in a logical representation, and either use Boolean satisfiability solvers (Hyttinen et al. 2013), *answer set* solvers (Hyttinen et al. 2014, 2017) or custom branch-and-bound algorithms (Rantanen et al. 2018, 2020) to identify a graph that minimizes the (weighted) sum of violated d-separation constraints. Since the class of DMGs dramatically expands the search space of possible graphs, the discovery task can be computationally challenging

for all of these methods, even when the number of variables is modest (i.e., fewer than 10).

To overcome limited scalability, other causal discovery methods have considered simplifications such as (i) only searching for causal ancestry relations, rather than direct causal connections (Magliacane et al. 2016), (ii) only allowing unmeasured confounders, but no cycles (Triantafillou and Tsamardinos 2015), or (iii) by weakening the faithfulness assumption (Zhalama et al. 2017). Instead of imposing restrictive assumptions, our approach maintains tractability by iteratively expanding the search space of possible graphs, and optimizing the solution using two alternating integer programs.

There is a small number of existing methods for causal discovery that are explicitly based on integer optimization (Jaakkola et al. 2010, Cussens 2011, Bartlett and Cussens 2017, Park and Klabjan 2017, Kucukyavuz et al. 2020, Manzour et al. 2021). These methods all focus on DAGs, and thus do not accommodate feedback cycles or unobserved confounders. Chen et al. (2021) develop an integer program for causal discovery with unobserved confounders but their framework does not capture feedback cycles. A second distinction is in the formulation of the problem—ours is a constraint-based approach, and accordingly searches for a graph that satisfies conditional (in)dependencies seen in the data. In contrast, the papers cited above present *score-based* methods, which typically involve maximizing the likelihood of the data for a given DAG. Such a formulation of the search problem is not easily generalized to handle cyclic and confounded models.

3. Path-Based Model for Causal Discovery

Our model takes as input a set of (conditional) independence and dependence relations over the variables V. Assuming the Markov and faithfulness conditions (and for the cyclic case, that the parameterization is linear Gaussian), these relations imply a corresponding set of d-separation and d-connection relations that must be satisfied by the output graph (Remarks 1 and 2). At a high level, our model conceptualizes the search space of DMGs as combinations of paths, and aims to select a set of paths such that the resulting graph minimizes violations of the input d-separation and d-connection constraints.

3.1. Model Formulation

Given a data set over a set of variables V, we denote by $C \subset V$ a generic set of conditioning variables. We then define $A_{ij} = \{C | C \subseteq V \setminus \{i,j\}\}$ to be the set of all possible conditioning sets C for the pair (i, j). We refer to the n^{th} such conditioning set in A_{ij} as $C_{ij}^n \in A_{ij}$. Then, let $D_{ij} = \{C \in A_{ij} | i \not\perp\!\!\!\perp j | C\}$ be the set of all conditioning sets such that i and j are statistically *dependent* conditional on C; similarly, let $I_{ij} = \{C \in A_{ij} | i \perp\!\!\!\perp j | C\}$ be the set of all conditioning sets such that i and j are statistically *independent* conditional on C. We assume that each pair of variables (i, j) is either dependent or independent given a particular conditioning set C. Thus, for all pairs (i, j) we have $D_{ij} \cup I_{ij} = A_{ij}$ and $D_{ij} \cap I_{ij} = \emptyset$. We use N_{ij}, N_{ij}^D and N_{ij}^I as index sets for A_{ij}, D_{ij} and I_{ij}, respectively, where $N_{ij} = N_{ij}^D \cup N_{ij}^I$.

Based on the equivalence established in Remark 1, the sets D_{ij} and I_{ij} encode the d-separation and d-connection relations implied by the distribution $P_{\mathcal{G}}(V)$.[7] A graph $\mathcal{G}$ satisfies the d-connection implied by $C \in D_{ij}$ if $i \not\perp_{\mathcal{G}} j | C$; similarly, $\mathcal{G}$ satisfies the d-separation implied by $C \in I_{ij}$ if $i \perp_{\mathcal{G}} j | C$. Our objective is to find a directed mixed graph $\mathcal{G}$ that minimizes the weighted sum of d-separation and d-connection constraints found in the data that are *not* satisfied in $\mathcal{G}$; that is, we want to find a graph $\mathcal{G}$ that minimizes the following loss function:

$$L(\mathcal{G}) = \sum_{i,j \in V} \sum_{n \in N_{ij}^D} \omega_{ij}^n \cdot \mathbf{1}(i \perp_{\mathcal{G}} j | C_{ij}^n) + \sum_{i,j \in V} \sum_{n \in N_{ij}^I} \omega_{ij}^n \cdot \mathbf{1}(i \not\perp_{\mathcal{G}} j | C_{ij}^n). \quad (3)$$

The parameters $\omega_{ij}^n > 0$ are penalization weights for violating the d-separation or d-connection relation between i and j implied by conditioning set C_{ij}^n. Various weighting schemes have been proposed in the literature (see, e.g., Hyttinen et al. 2014). We formulate the problem of searching for a graph that minimizes $L(\mathcal{G})$ as an integer program (IP). Our model constructs a graph by selecting *paths* that best fit the discovered d-connection and d-separation conditions. Despite the possible cycles in DMGs, we show in Section EC1 of the electronic companion that checking only a finite number of finite length paths is sufficient to determine all d-connection relations.[8] We define a path's *length* ℓ_p as the number of unique edges in p.

Let $col_{p_{ij}}$ be the set of colliders on a path p_{ij}. To capture possible d-connections obtained by conditioning on descendants of colliders, we define an *appendage of* p_{ij} to be an (acyclic) directed path that has as root a collider in $col_{p_{ij}}$ and does not pass through i or j. We refer to a path with one or more appendages as an *extended path*, and those without any appendages as a *simple path*.

A key aspect of our method, discussed further in Section 4, is controlling the length of *simple* paths that can be constructed using a given set of edges $\tilde{E}$. Accordingly, let $\mathcal{P}(\tilde{E}, \zeta) = \bigcup_{\{(i,j) \in V : i \neq j\}} P_{ij}(\tilde{E}, \zeta)$, where $P_{ij}(\tilde{E}, \zeta)$ stores all simple and extended paths between i and j that can be constructed from the edges in the set $\tilde{E}$ such that the longest *simple* path has length less than or equal to ζ.

For each $p \in \mathcal{P}(\tilde{E},\zeta)$ and $e \in \tilde{E}$, let ϕ_{pe} be a parameter where $\phi_{pe} = 1$ if and only if edge e belongs to path p, and let α^n_{ijp} be a parameter where $\alpha^n_{ijp} = 1$ if and only if the path $p \in P_{ij}(\tilde{E},\zeta)$ is *unblocked* with respect to the conditioning set $C^n_{ij} \in A_{ij}$. Following Definition 2, a simple path p is unblocked with respect to a conditioning set C^n_{ij} if all colliders on p are in C^n_{ij} and no noncolliders on p are in C^n_{ij}. An extended path p is unblocked with respect to C^n_{ij} if for each collider k on p we have $k \in C^n_{ij}$ or p contains an appendage that starts from k and has a node in C^n_{ij}, and no noncolliders on p are in C^n_{ij}.

Next, we define three types of binary decision variables. Let $\mathbf{x} \in \{0,1\}^{|\tilde{E}|}$ determine the presence of edges in the constructed graph $\mathcal{G} = (V,E)$, where $x_e = 1$ if and only if edge $e \in E$. Note that $\tilde{E}$ is an arbitrary set of edges and $E \subseteq \tilde{E}$ are the edges present in the constructed graph. Similarly, let $\mathbf{y} \in \{0,1\}^{|\mathcal{P}(\tilde{E},\zeta)|}$ determine paths in the graph $\mathcal{G}$, where $y_p = 1$ if and only if path p is in the graph $\mathcal{G}$, that is, $p \in \mathcal{P}(E,\zeta)$. Lastly, for each pair $i,j \in V$, we define the error variables $\mathbf{z}_{ij} \in \{0,1\}^{|N_{ij}|}$, where $z^n_{ij} = 1$ if and only if the d-separation relation in $\mathcal{G}$ does *not* correspond to the independence relation found in the data (that is, $i \perp_{\mathcal{G}} j | C^n_{ij}$ but $C^n_{ij} \in D_{ij}$, or $i \not\perp_{\mathcal{G}} j | C^n_{ij}$ but $C^n_{ij} \in I_{ij}$).

We can now define the constraints and objective function of the model. First, we require two constraints that enforce coherence between the edge and path variables $\mathbf{x}$ and $\mathbf{y}$:

$$y_p \geq \sum_{e \in \tilde{E}} \phi_{pe} x_e - (\ell_p - 1), \quad p \in \mathcal{P}(\tilde{E},\zeta), \tag{4a}$$

$$y_p \leq x_e, \quad e \in \{e \in \tilde{E} \,|\, \phi_{pe} = 1\},\ p \in \mathcal{P}(\tilde{E},\zeta). \tag{4b}$$

The first constraint ensures that if all the edges on a path p are selected, then path p is present in the graph. The second constraint ensures that path p can only be present in the graph if all the edges on path p are selected. Next, note that a path p is blocked with respect to a set C^n_{ij} if and only if $\alpha^n_{ijp} = 0$. Thus, to satisfy all d-separation relations, we would ideally like to construct a graph such that for any $i,j \in V$, $\alpha^n_{ijp} y_p = 0$ holds for all $n \in N^I_{ij}$ and $p \in P_{ij}(\tilde{E},\zeta)$. However, because the set of input d-separation and d-connection relations may not be jointly satisfiable, we allow for possible violations by introducing the error variable z^n_{ij}:

$$\alpha^n_{ijp} y_p \leq z^n_{ij}, \quad n \in N^I_{ij},\ p \in P_{ij}(\tilde{E},\zeta),\ i,j \in V. \tag{5}$$

For each conditioning set $C^n_{ij} \in I_{ij}$, constraint (5) forces $z^n_{ij} = 1$ if i and j are *not* d-separated with respect to C^n_{ij}. Similarly, i and j are d-connected with respect to $C^n_{ij} \in D_{ij}$ if and only if there is at least one unblocked path, or equivalently, $\sum_{p \in P_{ij}(\tilde{E},\zeta)} \alpha^n_{ijp} y_p \geq 1$. By again allowing for violations by introducing the error variable z^n_{ij}, we obtain the constraint

$$\sum_{p \in P_{ij}(\tilde{E},\zeta)} \alpha^n_{ijp} y_p \geq 1 - z^n_{ij}, \quad n \in N^D_{ij},\ i,j \in V, \tag{6}$$

which forces $z^n_{ij} = 1$ if i and j are not d-connected with respect to $C^n_{ij} \in D_{ij}$. Note that constraint (5) and (6) are defined over N^I_{ij} and N^D_{ij}, respectively, so that each z^n_{ij} variable appears in one constraint only. Finally, let ω^n_{ij} be the weight associated with the error variables z^n_{ij}. It follows from (5) and (6) that the total weighted violations of d-connection and d-separation relations is given by $\sum_{i,j \in V} \sum_{n \in N_{ij}} \omega^n_{ij} z^n_{ij}$. Minimizing these violations over the constraints (4)–(6) yields the optimization problem:

$$\begin{aligned} &\underset{\mathbf{x},\mathbf{y},\mathbf{z}}{\text{minimize}} \ \sum_{i,j \in V} \sum_{n \in N_{ij}} \omega^n_{ij} z^n_{ij} \\ &\text{subject to (4)–(6),} \\ \text{CausalIP}(\mathcal{P}(\tilde{E},\zeta)): \quad &x_e \in \{0,1\}, \quad e \in \tilde{E}, \\ &y_p \in \{0,1\}, \quad p \in \mathcal{P}(\tilde{E},\zeta), \\ &0 \leq z^n_{ij} \leq 1, \quad n \in N_{ij},\ i,j \in V. \end{aligned} \tag{7}$$

We refer to the formulation above as CausalIP($\mathcal{P}(\tilde{E},\zeta)$), where $\tilde{E}$ and $\mathcal{P}(\tilde{E},\zeta)$ are the set of candidate edges and paths the model has access to, respectively.[9] Note that while ζ is the maximum length in the set of candidate paths $\mathcal{P}(\tilde{E},\zeta)$, the constructed graph may still contain paths longer than ζ through concatenation of shorter paths. Also, while the errors tracked by the z^n_{ij} variables are binary (i.e., whether a d-separation relation is violated), in Section EC2 of the electronic companion we show that each z^n_{ij} is guaranteed to take on a value of 0 or 1 without requiring binary constraints, which greatly reduces the number of integer variables in the model. If $(\mathbf{x},\mathbf{y},\mathbf{z})$ is a solution to CausalIP($\mathcal{P}(\tilde{E},\zeta)$), then the graph returned by the model is given by $\mathcal{G}(\mathbf{x}) = (V,E)$ where $E = \{e | e \in \tilde{E} \text{ and } x_e = 1\}$. In Section EC2 of the electronic companion, we show how our approach can also be restricted to DAG- or ADMG-search, although more scalable methods already exist for those search spaces. Next, let

$$E^c = \{i \leftarrow j,\ i \rightarrow j,\ i \leftrightarrow j,\ \ \forall i,j \in V : i \neq j\} \tag{8}$$

be the set of all possible directed and bidirected edges in a complete graph over V. Also, note that the maximum length of a simple path (i.e., without appendages) over E^c is $|V| - 1$. We now address the performance of the formulation.

Proposition 1. *Let $\mathcal{G}^c$ be the graph returned by* CausalIP($\mathcal{P}(E^c, |V| - 1)$) *for some $\boldsymbol{\omega} > 0$. Then $\mathcal{G}^c \in$ argmin $_{\mathcal{G}} L(\mathcal{G})$, that is, $\mathcal{G}^c$ minimizes the loss function in* (3).

Proposition 1 confirms that CausalIP minimizes the loss function in (3) when it has access to complete set of edges E^c and complete set of paths $\mathcal{P}(E^c, |V| - 1)$. All proofs for the main body are contained in Section EC5 of the electronic companion.

3.2. Discovery Guarantee

In general, the independence structure seen in observational data does not uniquely identify the underlying causal graph. Two graphs that have the same independence structure (and thus the same d-separation relations) are said to be *Markov equivalent*.

Definition 4 (Markov Equivalence). DMG $\mathcal{G}_1 = (V, E)$ is Markov equivalent to

DMG $\mathcal{G}_2 = (V, E')$ if and only if $\mathcal{G}_1$ and $\mathcal{G}_2$ have identical d-separation relations:

$$\mathcal{G}_1 \sim \mathcal{G}_2 \quad \text{if and only if} \quad i \perp_{\mathcal{G}_1} j | C \Leftrightarrow i \perp_{\mathcal{G}_2} j | C \quad \forall i, j \in V \text{ and } C \subseteq V \setminus \{i, j\}. \tag{9}$$

Two graphs that are Markov equivalent cannot be distinguished by their d-separation relations alone.[10] In the context of causal discovery, the Markov equivalence class of the true, data-generating graph is the limit of what can be learned about the causal structure from the independence structure in the data (Geiger and Pearl 1988, Meek 1995).[11]

We now show that, under appropriate conditions, CausalIP correctly uncovers a graph in the Markov equivalence class of the true causal graph. We assume that we have access to the results of all possible independence tests over a given set of variables, and that the test results correctly describe an underlying ground truth DMG $\mathcal{G}_T$:

Assumption 2 (Complete Oracle). *Let $\mathcal{G}_T$ be the true data-generating graph. For all $i, j \in V$ and $C \subseteq V \setminus \{i, j\}$:* (i) *$C \in D_{ij}$ if and only if $i \not\perp\!\!\!\perp j | C$ in $P_{\mathcal{G}_T}(V)$, and* (ii) *$C \in I_{ij}$ if and only if $i \perp\!\!\!\perp j | C$ in $P_{\mathcal{G}_T}(V)$.*

Assumption 2 allows us to separate the discovery task, handled by CausalIP, from the statistical inference of the conditional independence tests. The assumption also describes the model inputs (i.e., I_{ij} and D_{ij}) that would be obtained in the large-sample limit, which we use to prove the asymptotic correctness of our model. We can now state the main result of this section:

Proposition 2. *Let $\mathcal{G}^c$ be the graph returned by* CausalIP *$(\mathcal{P}(E^c, |V| - 1))$ given Assumption 2. Then $\mathcal{G}^c \sim \mathcal{G}_T$ (i.e., $\mathcal{G}^c$ and $\mathcal{G}_T$ are Markov equivalent) for any $\boldsymbol{\omega} > 0$.*

Proposition 2 confirms the asymptotic correctness of CausalIP. When V is small, it is possible to solve CausalIP over the complete set of edges E^c and complete set of paths $\mathcal{P}(E^c, |V| - 1)$ using off-the-shelf integer programming solvers. However, this approach is not viable even for moderately sized graphs ($|V| > 6$), because CausalIP searches over *all* possible paths and extended paths induced by the set E^c, which explodes combinatorially in $|V|$. For larger graphs, a more efficient way of selecting paths is required.

4. Edge Generation Algorithm

Instead of attempting to solve CausalIP over the complete set of edges E^c, we develop an iterative solution algorithm that efficiently constructs a set of *candidate edges* $\tilde{E} \subset E^c$. In broad terms, the algorithm iterates between a master problem that constructs a graph using only the candidate edges in $\tilde{E} \subset E^c$ and a subproblem that leverages the results of the conditional independence tests to select new edges to include in $\tilde{E}$.[12]

4.1. Preliminaries

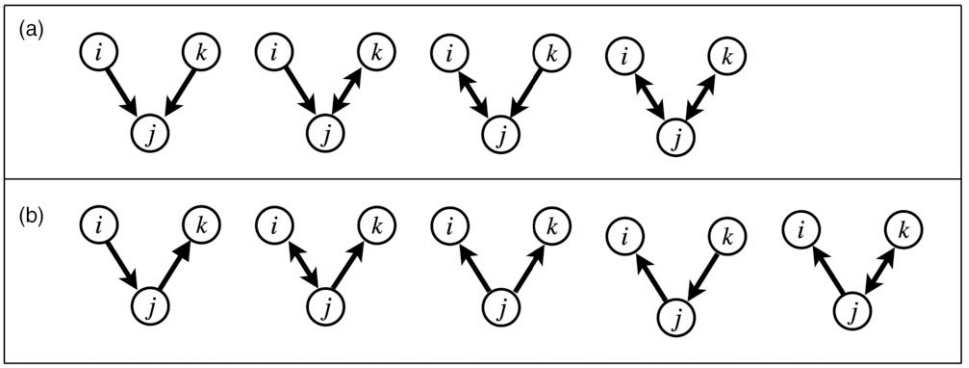
The selection of new edges is based on the observation that every path p (with $\ell_p \geq 2$) can be represented as a concatenation of node triples (i, j, k) that either form a collider at j or a noncollider at j. We refer to such a triple as a *collider chain* or a *noncollider chain*, respectively (see Figure 3). Next, we present necessary conditions for the presence of collider and noncollider chains in the underlying causal graph:

Lemma 1 (Necessary Conditions for Chains). *Suppose Assumption 2 holds and consider a triple of nodes (i, j, k) in a graph $\mathcal{G} \sim \mathcal{G}_T$. Then*

i. *If there exists a collider chain over (i, j, k) in $\mathcal{G}$, then $I_{ij} = I_{jk} = \emptyset$ and $j \notin C$ for all $C \in I_{ik}$.*

ii. *If there exists a noncollider chain over (i, j, k) in $\mathcal{G}$, then $I_{ij} = I_{jk} = \emptyset$ and $j \in C$ for all $C \in I_{ik}$.*

Figure 3. Collider Chains (a) and Noncollider Chains (b)



Lemma 1 formalizes the intuition that every triple along every path of the underlying causal graph may leave its corresponding signature in the independence structure of the data. These signatures can serve as indicators of the presence of the corresponding chains in the graph. The features in Lemma 1 do not *guarantee* that the underlying graph contains the corresponding chain, since they can also arise from other causal structures (see Figure 4). In other words, if a triple (i, j, k) satisfies the conditions in Lemma 1(i) or (ii), we interpret this as strong (but not conclusive) evidence of the corresponding chain's presence in the underlying causal graph. Using this characterization of chains, we construct the following two sets:

$$S = \{(i,j,k) | I_{ij} = I_{jk} = \emptyset \text{ and } j \notin C \text{ for all } C \in I_{ik}\}, \tag{10}$$

$$\overline{S} = \{(i,j,k) | I_{ij} = I_{jk} = \emptyset \text{ and } j \in C \text{ for all } C \in I_{ik}\}. \tag{11}$$

Here, S and $\overline{S}$ contain all triples (i, j, k) whose independence structure is indicative of a collider and noncollider chain, respectively. These sets are not disjoint in general: A triple (i, j, k) where (i, k) are never conditionally independent (i.e., $I_{ik} = \emptyset$) will be included in both S and $\overline{S}$.

Let E_{ijk} and $\overline{E}_{ijk}$ be the sets of all possible edges in a collider and noncollider chain over (i, j, k), respectively:

$$E_{ijk} = \{i \to j,\ i \leftrightarrow j,\ j \leftarrow k,\ j \leftrightarrow k\}, \tag{12a}$$

$$\overline{E}_{ijk} = \{i \leftarrow j,\ i \to j,\ i \leftrightarrow j,\ j \leftarrow k,\ j \to k,\ j \leftrightarrow k\} \tag{12b}$$

Notice that E_{ijk} is the set of edges that are in the collider chains in Figure 3(a), and $\overline{E}_{ijk}$ is the set of edges that are in the noncollider chains in Figure 3(b). The sets S and $\overline{S}$ contain triples that plausibly form chains in the underlying graph, and thus indicate plausible edges as well. We therefore focus on edges contained in the sets S and $\overline{S}$ when constructing the set of candidate edges $\tilde{E}$.

We further refine our search for edges by also considering d-connection or d-separation relations that are unsatisfied by an incumbent solution. For an initial set of candidate edges $\tilde{E}$, let $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ be the solution to CausalIP$(\mathcal{P}(\tilde{E}, \zeta))$ with the corresponding graph $\mathcal{G}(\mathbf{x})$. The error variable $\mathbf{z}$ tracks those d-separations and d-connections that are inconsistent with the independence and dependence findings in the data. Since our method will incrementally add edges to $\tilde{E}$, we focus on those errors where a conditional dependence found in the data are not yet matched by a d-connection in the graph $\mathcal{G}(\mathbf{x})$. Specifically, we are interested in the pairs (i, k) where we have $C^n_{ik} \in D_{ik}$ based on the test results, but the d-connection $i \perp\!\!\!\perp k | C^n_{ik}$ is *not* satisfied in $\mathcal{G}(\mathbf{x})$, and consequently $z^n_{ik} = 1$. Accordingly, for each pair (i, k), we can define the set

$$N^D_{ik}(\mathbf{z}) = \{n \in N^D_{ik} | z^n_{ik} = 1\} \tag{13}$$

to represent the d-connection relations for (i, k) that are implied by the conditional independence tests but violated by the current graph $\mathcal{G}(\mathbf{x})$. Since our goal is to select *new* edges to add to $\tilde{E}$, we have to identify edges that are not already in $\tilde{E}$. We can now identify triples that satisfy two criteria: (i) some of their edges have not yet been considered (i.e., $E_{ijk} \not\subset \tilde{E}$ or $\overline{E}_{ijk} \not\subset \tilde{E}$), and (ii) in the incumbent graph $\mathcal{G}(\mathbf{x})$, i and k are d-separated for some conditioning set C^n_{ik} $(i \perp k | C^n_{ik})$, even though test results indicate that i and k are dependent for that conditioning set $(i \not\perp k | C^n_{ik})$. We now define two sets that contain triples that satisfy these two criteria:

$$\Psi(\mathbf{z}) = \{(i,j,k) | \text{there exists } n \in N^D_{ik}(\mathbf{z}) \text{ such that } j \in C^n_{ik} \text{ and } E_{ijk} \not\subset \tilde{E}\}, \tag{14a}$$

$$\overline{\Psi}(\mathbf{z}) = \{(i,j,k) | \text{there exists } n \in N^D_{ik}(\mathbf{z}) \text{ such that } j \notin C^n_{ik} \text{ and } \overline{E}_{ijk} \not\subset \tilde{E}\}. \tag{14b}$$

The sets $\Psi(\mathbf{z})$ and $\overline{\Psi}(\mathbf{z})$ need not be disjoint. They only differ in their check of whether j belongs to a conditioning set C^n_{ik} that shows i and k to be dependent in the data. This specific check on the role of j ensures that if we now combine the Ψ-sets with the S-sets, the collider/noncollider chains associated with the triples of the respective sets are candidates to address the inconsistencies identified by $\mathbf{z}$. We can finally define the sets that are the focal points of our search for new edges:

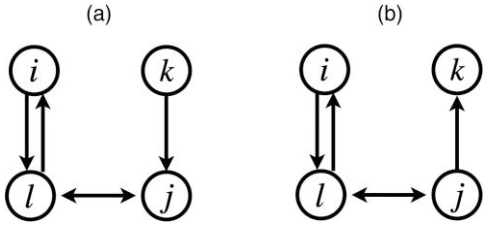
$$S(\mathbf{z}) = S \cap \Psi(\mathbf{z}), \tag{15a}$$

$$\overline{S}(\mathbf{z}) = \overline{S} \cap \overline{\Psi}(\mathbf{z}), \tag{15b}$$

Intuitively, $S(\mathbf{z})$ and $\overline{S}(\mathbf{z})$ represent chains that we have good reason to think exist in the graph (because of S and $\overline{S}$) *and* correspond to edges useful for satisfying a required d-connection that is violated by $\mathcal{G}(\mathbf{x})$ (because of $\Psi(\mathbf{z})$ and $\overline{\Psi}(\mathbf{z})$).

We have thus far established that edges among the triples in $S(\mathbf{z})$ and $\overline{S}(\mathbf{z})$ are strong candidates for inclusion in the candidate set $\tilde{E}$ to send to CausalIP. We

Figure 4. Alternative Causal Relations Corresponding to Observed Independence Relations



Notes. (a) The triple (i, j, k) satisfies all conditions in Lemma 1(i) but does not form a collider chain. (b) The triple (i, j, k) satisfies all conditions in Lemma 1(ii) but does not form a noncollider chain. In both cases the $i \to j$ edge is missing.

now address the problem of how to actually select edges from these sets to pass to CausalIP.

4.2. Generating Candidate Edges

Since the tractability of CausalIP suffers if the edge set $\tilde{E}$ is too large, we would ideally like to select edges that can reconcile unsatisfied d-connections, without needlessly introducing redundant edges. To that end, our approach will be to select the smallest number of edges such that at least one edge from each triple in $S(\mathbf{z})$ and $\overline{S}(\mathbf{z})$ is selected. Proposition 3 below establishes that this minimal edge selection subproblem – which we call NewEdgesIP – is NP-hard; accordingly, we formulate and solve it as an integer program.

There are three types of edges that may exist between every pair of nodes (i, j): $i \to j$, $i \leftarrow j$, and $i \leftrightarrow j$. We index these three edge types by $t \in \{1,2,3\}$, respectively. Let τ_{ij}^t be a binary decision variable where $\tau_{ij}^t = 1$ if a type t edge between nodes i and j is selected to be included in $\tilde{E}$, and $\tau_{ij}^t = 0$ otherwise. Let λ_{ij}^t be a parameter where $\lambda_{ij}^t = 1$ if $\tilde{E}$ contains a type t edge between nodes i and j, and $\lambda_{ij}^t = 0$ otherwise.

We now define the constraints and objective of NewEdgesIP. To select edges from $S(\mathbf{z})$, we include the following constraints:

$$\sum_{t\in\{1,3\}} (\tau_{ij}^t + \lambda_{ij}^t) \geq 1, \quad (i,j,k) \in S(\mathbf{z}), \tag{16a}$$

$$\sum_{t\in\{2,3\}} (\tau_{jk}^t + \lambda_{jk}^t) \geq 1, \quad (i,j,k) \in S(\mathbf{z}), \tag{16b}$$

$$\sum_{t\in\{1,3\}} \tau_{ij}^t + \sum_{t\in\{2,3\}} \tau_{jk}^t \geq 1, \quad (i,j,k) \in S(\mathbf{z}). \tag{16c}$$

The first two constraints construct a collider chain over (i, j, k): the first ensures that there is either a new or existing edge between i and j with an arrowhead at j, and the second ensures there is either a new or existing edge between k and j with an arrowhead at j. Then, the third constraint forces at least one new edge to be selected from the candidate collider chain. Similarly, to select edges from $\overline{S}(\mathbf{z})$, we include:

$$\sum_{t\in\{1,2,3\}} (\tau_{ij}^t + \lambda_{ij}^t) \geq 1, \quad (i,j,k) \in \overline{S}(\mathbf{z}), \tag{17a}$$

$$\sum_{t\in\{1,2,3\}} (\tau_{jk}^t + \lambda_{jk}^t) \geq 1, \quad (i,j,k) \in \overline{S}(\mathbf{z}), \tag{17b}$$

$$\tau_{ij}^2 + \lambda_{ij}^2 + \tau_{jk}^1 + \lambda_{jk}^1 \geq 1, \quad (i,j,k) \in \overline{S}(\mathbf{z}), \tag{17c}$$

$$\sum_{t\in\{1,2,3\}} (\tau_{ij}^t + \tau_{jk}^t) \geq 1, \quad (i,j,k) \in \overline{S}(\mathbf{z}). \tag{17d}$$

Analogous to (16a) and (16b), the first three constraints above construct a noncollider chain over (i, j, k), using either existing or new edges: The first two constraints ensure an edge exists between both (i, j) and (j, k), and the third constraint ensures that j is a noncollider. Then, the fourth constraint forces at least one new edge to be selected from the constructed noncollider chain.

The final group of constraints we include are:

$$\tau_{ij}^t + \lambda_{ij}^t \leq 1, \quad i,j \in V,\ t \in \{1,2,3\}, \tag{18a}$$

$$\tau_{ij}^1 = \tau_{ji}^2, \quad i,j \in V, \tag{18b}$$

$$\tau_{ij}^3 = \tau_{ji}^3, \quad i,j \in V. \tag{18c}$$

The first constraint ensures we do not select an edge that is already included in $\tilde{E}$. The second constraint enforces that $i \to j$ and $j \leftarrow i$ are the same edge, and the third constraint enforces that $i \leftrightarrow j$ and $j \leftrightarrow i$ are the same edge.

Our objective is to minimize the total number of new edges added to the set of candidate edges $\tilde{E}$. Combining this objective with the constraints (16)–(18) and forcing each τ_{ij}^t to be binary yields the following formulation:

$$\underset{\boldsymbol{\tau}}{\text{minimize}} \quad \sum_{\substack{i,j\in V:\\ j<i}} \sum_{t\in\{1,2,3\}} \tau_{ij}^t$$

$$\text{NewEdgesIP}(S(\mathbf{z}), \overline{S}(\mathbf{z}), \boldsymbol{\lambda}) : \text{subject to} \quad (16)\text{–}(18)$$
$$\tau_{ij}^t \in \{0,1\}, \quad i,j \in V,\ t \in \{1,2,3\}.$$

Next, we analyze the computational complexity of NewEdgesIP.

Proposition 3. *The integer programming problem NewEdgesIP is NP-hard.*

The output of this formulation is a set of new edges E_{new} to be added to $\tilde{E}$, where E_{new} contains a type t edge between nodes i and j if and only if $\tau_{ij}^t = 1$ at an optimal solution to NewEdgesIP. In summary, NewEdgesIP generates edges efficiently by searching for edges that satisfy the following criteria:

i. the edges belong to collider or noncollider chains for which we have strong evidence of their presence in the true graph based on the observed independence and dependence relations (i.e., the chains belong to S or $\overline{S}$), and

ii. The edges belong to collider or noncollider chains whose inclusion in the graph would satisfy a d-connection relation violated by the incumbent solution (i.e., the chains belong to $\Psi(\mathbf{z})$ or $\overline{\Psi}(\mathbf{z})$).

Having defined the key components of our method, we now present a summary of the algorithm and prove its correctness.

4.3. Algorithm Summary and Main Result

Algorithm 1 (EdgeGen) provides an overview of the main steps. Let $\mathcal{P}^-(\tilde{E}, \zeta) = \bigcup_{\{(i,j)\in V: i\neq j\}} P_{ij}^-(\tilde{E}, \zeta)$, where $P_{ij}^-(\tilde{E}, \zeta)$ is the set of all simple paths (i.e., without appendages) between i and j that have maximum length ζ and can be constructed from the edges in $\tilde{E}$. In step 1, for an initial set of candidate edges $\tilde{E}$ and value of ζ, EdgeGen solves CausalIP over $\mathcal{P}^-(\tilde{E}, \zeta)$ and returns a graph $\mathcal{G}_s = (V, E_s)$. By ignoring simple paths

longer than ζ and all extended paths in step 1, it dramatically reduces the number of candidate paths that are considered by CausalIP, which allows the model to quickly return an approximate solution. But it also implies that the loss function in CausalIP may undercount the number of violated input relations. Thus, in step 2, we call the $\mathcal{G}$-Postprocess subalgorithm (Algorithm 2) to manually check *all* paths (including appendages) in the returned graph $\mathcal{G}_s$ to precisely identify the d-separation or d-connection relations violated in $\mathcal{G}_s$. The errors are tracked in $\boldsymbol{\epsilon}$: For conditioning set C_{ij}^n that makes i and j dependent, we have $\epsilon_{ij}^n = 0$ if there exists a d-connecting path in $\mathcal{G}_s$, otherwise $\epsilon_{ij}^n = 1$. Similarly, for conditioning set C_{ij}^n that make i and j independent, we have $\epsilon_{ij}^n = 1$ if there exists a d-connecting path in $\mathcal{G}_s$, otherwise $\epsilon_{ij}^n = 0$.

If $\mathcal{G}$-Postprocess finds that $\mathcal{G}_s$ violates any input relations, new simple paths are introduced in step 3.1 of EdgeGen. Specifically, it calls the UpdateEdges subalgorithm (Algorithm 3) to generate new simple paths by either adding new edges to $\tilde{E}$ by solving NewEdgesIP, or by increasing the maximum path length ζ. If $S(\mathbf{z}) \cup \overline{S}(\mathbf{z}) \neq \emptyset$, we solve NewEdgesIP to obtain new edges E_{new}, otherwise we randomly pick triples from S and $\overline{S}$ to pass to NewEdgesIP. When $E_{\text{new}} = \emptyset$, we increment the path length ζ while $\zeta < |V| - 1$, otherwise we randomly select a new edge to add to $\tilde{E}$.

EdgeGen terminates and returns the graph $\mathcal{G}^*$ when all d-connection and d-separation relations are satisfied, or all possible candidate paths have been generated. Note that as the initial edge set in EdgeGen, we use

$$\tilde{E}_0 = \{i \to j, i \leftarrow j, i \leftrightarrow j \mid I_{ij} = \emptyset, D_{ik} = D_{jk} = \emptyset \text{ for all } k \in V \setminus \{i,j\}\},$$

which captures edges that may not belong to any chains implied by dependencies (i.e., those that do not correspond to any member of S or $\overline{S}$).

Next we show that EdgeGen always terminates.

Algorithm 1 (EdgeGen)

Input: $V, S, \overline{S}, \boldsymbol{\omega}, \zeta$.
Output: $\mathcal{G}^*$.
Initialize: $\tilde{E} = \tilde{E}_0$, $\sigma = \emptyset$, $s = 1$.
1. Solve CausalIP($\mathcal{P}^-(\tilde{E}, \zeta)$) and get $\mathcal{G}_s = (V, E_s)$ where $E_s = \{e \in \tilde{E} \mid x_e = 1\}$.
2. Call $\mathcal{G}$-Postprocess to obtain $\boldsymbol{\epsilon}_s$ associated with $\mathcal{G}_s$. Set $\sigma_s = \boldsymbol{\omega}^\top \boldsymbol{\epsilon}_s$.
3. **if** $\boldsymbol{\omega}^\top \boldsymbol{\epsilon}_s > 0$:
   3.1. Call UpdateEdges to update ζ and $\tilde{E}$. Update $s = s + 1$.
   3.2. **if** $\tilde{E} \neq E^c$ **or** $\zeta < |V| - 1$:
   Go to Step 1.
   3.3 **else**:
   Solve CausalIP($\mathcal{P}(\tilde{E}, \zeta)$) and get $\mathcal{G}_s = (V, E_s)$. Set $\sigma_s = \boldsymbol{\omega}^\top \boldsymbol{\epsilon}_s$. Go to Step 4.
4. Return $\mathcal{G}^* = \mathcal{G}_{s^*}$ where $s^* = \text{argmin}_s \sigma_s$.

Algorithm 2 ($\mathcal{G}$-Postprocess Subalgorithm)

Input: E_s, I_{ij} and D_{ij} for $i, j \in V$.
Output: ϵ_{ij}^n for $i, j \in V, n \in N_{ij}$.
1. **for** $n \in N_{ij}^I$ and $i, j \in V$:
   If $\sum_{p \in \mathcal{P}_{ij}(E_s, |V|-1)} \alpha_{ijp}^n > 0$, then $\epsilon_{ij}^n = 1$. Otherwise, $\epsilon_{ij}^n = 0$.
2. **for** $n \in N_{ij}^D$ and $i, j \in V$:
   If $\sum_{p \in \mathcal{P}_{ij}(E_s, |V|-1)} \alpha_{ijp}^n > 0$, then $\epsilon_{ij}^n = 0$. Otherwise, $\epsilon_{ij}^n = 1$.
3. Return ϵ_{ij}^n for $i, j \in V, n \in N_{ij}$.

Algorithm 3 (UpdateEdges Subalgorithm)

Input: $S(\mathbf{z}), \overline{S}(\mathbf{z}), S, \overline{S}, \tilde{E}, \tilde{E}_0, \zeta$.
Output: $\zeta, \tilde{E}$.
1. **if** $S(\mathbf{z}) \cup \overline{S}(\mathbf{z}) \neq \emptyset$:
   Solve NewEdgesIP($S(\mathbf{z}), \overline{S}(\mathbf{z}), \boldsymbol{\lambda}$) to obtain E_{new}.
   else:
   Pick any $(i, j, k) \in S$ such that $E_{ijk} \notin \tilde{E}$. Set $R(\mathbf{z}) = (i, j, k)$.
   Pick any $(i, j, k) \in \overline{S}$ such that $\overline{E}_{ijk} \notin \tilde{E}$. Set $\overline{R}(\mathbf{z}) = (i, j, k)$.
   Solve NewEdgesIP($R(\mathbf{z}), \overline{R}(\mathbf{z}), \boldsymbol{\lambda}$) to obtain E_{new}.
2. Update $\tilde{E} \leftarrow \tilde{E} \cup E_{\text{new}}$.
3. **if** $E_{\text{new}} = \emptyset$ and $\zeta < |V| - 1$:
   Update $\zeta = \zeta + 1$ and $\tilde{E} \leftarrow \tilde{E}_0$.
   else if $E_{\text{new}} = \emptyset$ and $\zeta = |V| - 1$:
   Select a random edge $e \in E^c \setminus \tilde{E}$. Update $\tilde{E} \leftarrow \tilde{E} \cup e$.
4. Return $\zeta, \tilde{E}$.

Proposition 4. *EdgeGen is guaranteed to terminate. Further, if Assumption 2 holds, the objective in (3) is equal to zero at termination.*

We now present our main result:

Theorem 1. *Let $\mathcal{G}^*$ be the graph returned by EdgeGen for $\boldsymbol{\omega} > 0$.*

i. *$\mathcal{G}^c \in \text{argmin}_{\mathcal{G}} L(\mathcal{G})$, that is, $\mathcal{G}^c$ minimizes the objective in (3).*

ii. *Further, if Assumption 2 holds, $\mathcal{G}^* \sim \mathcal{G}_T$ (i.e., $\mathcal{G}^*$ and $\mathcal{G}_T$ are Markov equivalent).*

Theorem 1(ii) states that EdgeGen, which is far more scalable than a brute-force solution of the full CausalIP($\mathcal{P}(\tilde{E}, \zeta)$) model, preserves the same discovery guarantees given in Proposition 2. Similar to Proposition 2, Theorem 1(ii) is an asymptotic guarantee, due to its dependence on Assumption 2. In the more realistic finite-sample setting where Assumption 2 does not hold, the input relations may not be jointly satisfiable, in which case we obtain the weaker result Theorem 1(i). In this setting, most constraint-based methods that also allow for cycles and confounders aim to find a graph that minimizes (weighted) violations of the input constraints (Hyttinen et al. 2013, 2014, 2017; Rantanen et al. 2020). However, *exactly* minimizing such violations requires searching over the entire space of DMGs. As a consequence, in the

finite-sample setting where Assumption 3 does not hold, our approach can be viewed as a way to prioritize certain edges for minimizing the number of unsatisfied dependence and independence relations. As demonstrated in the numerical results below, the advantage of this heuristic approach is that it scales to instances that are intractable for provably optimal methods from the literature, while maintaining reasonable accuracy. Further, in the setting where Assumption 2 *does* hold, our method outperforms appropriate benchmark algorithms by an order of magnitude with respect to solution time, without sacrificing optimality.

4.4. Computational Performance

In this section we examine the computational performance of EdgeGen using synthetic data. To serve as performance benchmarks, we also implemented the causal discovery methods described in Hyttinen et al. (2013) and Hyttinen et al. (2014), both of which also allow for feedback loops and latent confounders.[13] Hyttinen et al. (2013) only address the conflict-free setting, that is, where the joint set of conditional independence and dependence constraints are consistent with some DMG. They solve the discovery problem using a Boolean satisfiability solver. In contrast, Hyttinen et al. (2014) allow for conflicts among the constraints and propose a solution method based on answer set programming. For conciseness, we will refer to these two approaches as SAT and ASP, respectively. We also created an additional hybrid benchmark by combining the logical encoding developed in Hyttinen et al. (2013) and the solver used in Hyttinen et al. (2014), which we refer to as SAT+ASP. All algorithms are available in the code package at https://github.com/nkaynar/causal-IP.

We conducted three sets of numerical experiments. First, we considered a *conflict-free* setting in which we supply as input the (conditional) independence and dependence relations that hold for the ground truth graph $\mathcal{G}_T$. This allows us to separate the statistical question of how to handle errors that arise from finite samples from the combinatorial challenge of identifying the graph given the set of constraints. All four methods are guaranteed to return a graph that is Markov equivalent to $\mathcal{G}_T$ in this setting, so we focus our comparison exclusively on solution times. Second, we considered a *conflicted* setting where the input constraints are estimated from finite samples generated by $\mathcal{G}_T$ and therefore may not be jointly satisfiable (i.e., Assumption 2 does not hold). In this setting, the SAT algorithm does not apply, so we compare EdgeGen with ASP with respect to solution time and accuracy. Third, we test how well EdgeGen scales to large graphs when we limit the conditioning set size. We report solution times and accuracy.

The full graph and data generating procedure is discussed in Section EC3 of the electronic companion. In all simulations, we simulated 25 DMGs for each setting (allowing for cycles and unobserved confounders) and varied the graph density by controlling the *maxDegree* of nodes. Directed and bidirected edges were sampled with equal probability. All experiments were run on an Intel Xeon E5-2680 machine with 3.0 GHz×24 processors and 20 GB of memory, and used Gurobi v8.0 to solve CausalIP and NewEdgesIP.

4.4.1. Conflict-Free Setting. For the conflict-free case, we considered graphs over $|V| = 6, 8 \ldots, 18$ nodes and varied the *maxDegree* from 2 through 5. As required by default for all four methods, we computed all (conditional) dependence and independence relations for each graph. In fact, generating all ground truth constraints from each graph ultimately constituted the limiting factor for scaling this simulation beyond 18 nodes. Since we used the true independence and dependence relations as input to the method, no data generation procedure is needed.

Following Hyttinen et al. (2014), for ASP we used uniform weights for all constraints. SAT and SAT+ASP do not have a weighting scheme, but effectively also treat the constraints with equal weight. Since EdgeGen is sensitive to being flooded with too many edges, we weighted the dependence constraints with $\omega_{ij}^n = 1$ and the independence constraints with $\omega_{ij}^n = M$ where M is a large integer.[14] Since the set of constraints is satisfiable in this setting, this weighting scheme only amounts to prioritizing independence constraints in the search. Since all methods require the same precomputation of independence and dependence relations, we isolate the performance of each of the four methods by reporting the solution times of the discovery task only.

Table 1 shows the results: The median solution times (without any time-outs) across 25 instances for each algorithm to identify a DMG that is Markov equivalent to the ground truth graph $\mathcal{G}_T$. EdgeGen is 1-2 orders of magnitude faster than the competing methods and can solve large instances for which the other methods exceeded our 20 GB memory capacity.

4.4.2. Conflicted Setting. For the conflicted case, we generated the 25 DMGs for each $|V| = 5, 6, \ldots, 10, 15$ with *maxDegree* varying from 2 through 5 as before. We then used the code from Hyttinen et al. (2014) to parameterize the graphs as linear Gaussian models and generated 5,000 observations from each graph. We computed weights for each constraint using their most successful weighting scheme, the pseudo-Bayesian "log-weights". These weights approximate the posterior probability that a particular (conditional) independence holds in the data starting from a

Table 1. Median Solution Times (Nearest CPU Second) over 25 Random Instances in the Conflict-Free Setting

$\|V\|$	EdgeGen	SAT	ASP	SAT+ASP
Panel A. Maximum node degree 2				
6	0	0	0	0
8	0	2	1	0
10	1	19	15	5
12	2	115	166	34
14	9	—	—	242
16	56	—	—	—
18	316	—	—	—
Panel B. Maximum node degree 3				
6	0	1	0	0
8	0	12	1	1
10	1	108	15	5
12	2	—	182	38
14	20	—	—	273
16	61	—	—	—
18	339	—	—	—
Panel C. Maximum node degree 4				
6	0	1	0	0
8	1	17	1	1
10	2	293	17	6
12	3	—	192	39
14	14	—	—	265
16	211	—	—	—
18	1,428	—	—	—
Panel D. Maximum node degree 5				
6	0	1	0	0
8	2	41	2	1
10	3	825	22	8
12	15	—	206	43
14	34	—	—	278
16	490	—	—	—
18	—	—	—	—

Note. Dashes indicate that the instances could not be solved due to insufficient memory (20 GB).

uniform prior on independence and dependence (for details, see section 4.3 and appendix B of Hyttinen et al. (2014)).[15] Due to the size of this simulation and the complexity of this inference task, we enforced a 500 seconds time limit per instance on both methods.

Table 2 shows the results: For each set of 25 DMGs over a particular number of nodes $|V|$ and edge density controlled by *maxDegree*, we show the median fraction of incorrect independence and dependence constraints (relative to the ground truth graph) implied by the output graph returned by each method. To provide some calibration of the quality of the set of input constraints, the second column ("Test") gives the fraction of incorrect constraints in the input. Since ASP can solve small instances optimally, we mark them with an asterisk (*) when this occurred within the 500 s time limit for all 25 instances. We use a dash ($-$) to indicate ASP was not able to return even an initial graph after 500 s.

EdgeGen achieves almost optimal results for the small instances where we can compute the optimal answers using ASP (the $*$-cases). For larger instances, EdgeGen returns more accurate results than ASP within the 500 s time limit, and is close to or below the baseline error rate from the input tests. For most of these instances it is not known how long ASP would take to return the optimal solution, or even just a solution with lower error rates than EdgeGen.

4.4.3. Large Graphs and Limited Conditioning Sets.

For the scalability simulation, we generated graphs for $|V| \in \{20, 30, 40, 50\}$ with *maxDegree* for each node again varying from 2 to 5. The graph generation, as well as the weighting scheme for the constraints,

Table 2. Median Errors (Fraction of Incorrect (Conditional) Independence and Dependence Constraints in the Output Graph Relative to the Ground Truth Graph) over 25 Random Instances in the Conflicted Setting, Using a 500-Second Time Limit

	Error		
$\|V\|$	Test	EdgeGen	ASP
Panel A. Maximum node degree 2			
5	0.19	0.19	0.18[a]
6	0.16	0.18	0.13[a]
7	0.17	0.09	0.13
8	0.12	0.10	0.21
9	0.13	0.10	0.35
10	0.13	0.12	0.53
15	0.08	0.06	—
Panel B. Maximum node degree 3			
5	0.23	0.19	0.18[a]
6	0.19	0.14	0.10[a]
7	0.21	0.18	0.18
8	0.21	0.18	0.25
9	0.22	0.21	0.41
10	0.21	0.20	0.43
15	0.12	0.08	—
Panel C. Maximum node degree 4			
5	0.25	0.16	0.16[a]
6	0.23	0.17	0.17[a]
7	0.22	0.16	0.21
8	0.25	0.24	0.25
9	0.26	0.21	0.30
10	0.28	0.22	0.34
15	0.21	0.21	—
Panel D. Maximum node degree 5			
5	0.19	0.14	0.13[a]
6	0.31	0.28	0.25[a]
7	0.24	0.18	0.16
8	0.27	0.21	0.21
9	0.31	0.21	0.25
10	0.31	0.24	0.30
15	0.25	0.27	—

Notes. Dashes indicate that no solution could be returned within the time limit. Column "Test" reports the fraction of incorrect constraints (relative to ground truth) in the input.

[a]Indicates that all 25 instances solved optimally within the time limit.

followed that of the unconflicted setting above. However, for graphs of this size we cannot efficiently generate all the ground truth (in)dependence relations as there are $\binom{|V|}{2}2^{|V|-2}$ such relations for each graph. We therefore limited the input constraints to conditioning sets of size 0 or 1, and again imposed a time limit of 500 s. Even though all the input constraints are correct with respect to the ground truth graph, errors can arise in the output, as only a subset of all the constraints are given to EdgeGen. Since we do not have all the ground truth constraints, we cannot use the same error measure as in the conflicted setting, so we report Type 1 (false positive) and Type 2 errors (false negative) and the False Discovery Rate (FDR) for the undirected adjacency relations in the output graph compared with the ground truth (two nodes are adjacent in a graph if they have an edge between them).

Table 3 shows the results. Unsurprisingly, for *maxDegree* = 2, the unconflicted constraints with a maximum conditioning set size of 1 are sufficient to correctly identify the adjacency relations (panel A), and EdgeGen solves these sparse graphs within a few seconds. As these large graphs get denser, solution times quickly increase, but the adjacency errors remain low. For graphs with *maxDegree* = 5 (panel D), we were able to solve all 25 instances to optimality within the 500 s time limit for $|V| = 20$; for $|V| \geq 30$, our method did not return any graph at all within 500 s. Nevertheless, it should be noted that we are not aware of any other discovery method that can handle graphs with cycles and unmeasured confounders over this many nodes.

Table 3. Median Solution Times and Errors over 25 Random Instances in the Unconflicted Setting with a Maximum Conditioning Set Size of 1

$\|V\|$	Solution time	Type 1	Type 2	FDR
Panel A. Maximum node degree 2				
20	0	0	0	0
30	1	0	0	0
40	3	0	0	0
50	7	0	0	0
Panel B. Maximum node degree 3				
20	1	0.006	0	0.042
30	2	0.002	0	0.029
40	4	0.001	0	0.023
50	25	0.002	0	0.038
Panel C. Maximum node degree 4				
20	15	0.043	0	0.213
30	77	0.021	0	0.157
40	—	—	—	—
50	—	—	—	—
Panel D. Maximum node degree 5				
20	177	0.049	0.034	0.235
30	—	—	—	—
40	—	—	—	—
50	—	—	—	—

Note. Dashes indicate that no instance could produce a solution within 500 seconds.

5. Empirical Study: Investigating the Validity of an Instrumental Variable

In this section, we show how our discovery method can be used to investigate the validity of an instrumental variable. As a point of reference, we compare our method to the test proposed in Kitagawa (2015) by applying both approaches to the data used in the landmark studies on the returns to education by Angrist and Krueger (1991) and Card (1993).

5.1. Instrumental Variables and Graphical Criteria

Instrumental variables (Bowden and Turkington 1990, Angrist et al. 1996, Angrist and Krueger 2001) are one of the most widely used techniques to adjust causal effect estimates for bias due to unmeasured confounding. In order to isolate the causal effect of the treatment on the outcome from the spurious dependence due to confounding, the *instrument* must satisfy a set of structural assumptions. Pearl (2000) formalizes these conditions in graphical terms as follows:

Definition 5 (Instrumental Variables). Given a graph $\mathcal{G}$, a node i is an *instrument* for the effect of j on k if (1) $i \not\perp\!\!\!\perp_{\mathcal{G}} j$ and (2) $i \perp\!\!\!\perp_{\bar{\mathcal{G}}} k$, where $\bar{\mathcal{G}}$ is the graph obtained by removing $j \to k$ from $\mathcal{G}$.

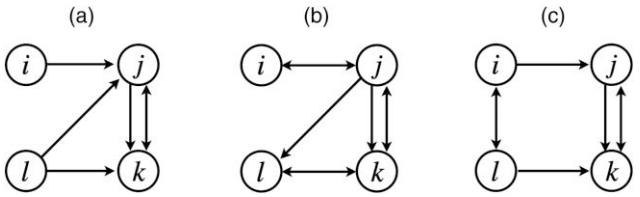
Condition (1) is often referred to as the *relevance* criterion—the instrument must be correlated with the treatment—while condition (2) combines the *exclusion* and *exogeneity* criteria: the instrument can only affect the outcome via the treatment and there cannot be any confounding of the instrument and outcome. Figure 5 provides examples of valid and invalid instruments for different causal structures.

When a variable violating condition (2) can be turned into a valid instrument using a suitable conditioning set W, then it is said to be a *conditional instrument*, and Definition 5 can be slightly generalized to capture the case:

Definition 6 (Conditional Instrumental Variables). Given a graph $\mathcal{G}$, a node i is an *conditional instrument* for the effect of j on k if there exists a conditioning set W such that (1) $i \not\perp\!\!\!\perp_{\mathcal{G}} j | W$, (2) $i \perp\!\!\!\perp_{\bar{\mathcal{G}}} k | W$, where $\bar{\mathcal{G}}$ is the graph obtained by removing the edge $j \to k$ from $\mathcal{G}$, and (3) W contains only nondescendants of k. (Pearl 2000).

In both definitions above, verifying that an instrument satisfies condition (1) is easily done by ensuring it is sufficiently correlated with the treatment.[16] In contrast, establishing condition (2) poses a major challenge (Angrist et al. 1996, Stock 2002). As a consequence, the validity of instruments is often justified by context-specific arguments.[17] A recent line of work has

Figure 5. Examples of Valid and Invalid Instruments in Various Causal Structures



Notes. Node i is a valid instrument for the effect of j on k in (a) and (b), but an invalid instrument in (c) because the path $i \leftrightarrow l \rightarrow k$ violates Definition 5(2). Conditioning on $W = \{l\}$ in (c) makes i a valid conditional instrument.

developed tests for instrument validity under a variety of assumptions (e.g., Kitagawa 2015, Mourifié and Wan 2017, Kédagni and Mourifié 2020). Kitagawa's test, which we will compare with in Sections 5.3 and 5.4, assumes a binary treatment and a discrete instrument. The test builds on the "testable implications" of valid instruments from Balke and Pearl (1997) and Heckman and Vytlacil (2005), and uses a Kolmogorov-Smirnov statistic to compare the empirical outcome distributions for each value of the instrument, for both treated and control units.

5.2. A Path-Based Procedure for Investigating Instrument Validity

In contrast to Kitagawa's test, our procedure does not make any parametric assumptions and instead directly evaluates the extent to which a candidate instrument conforms to the graphical criteria in Definition 5 (or 6). In particular, given observational data for the relevant variables (i.e., the instrument, treatment, outcome, and additional covariates), we solve a causal discovery problem over two search spaces: the full space of directed mixed graphs, and the subspace of graphs in which the graphical criteria in Definition 5 (or 6) are enforced. If forcing the instrument to be valid degrades model fit – as measured by the loss function $L(\mathcal{G})$ in CausalIP—we take that to constitute evidence against the instrument's validity, because it implies that the instrument is invalid in the graphs most consistent with the data. Obviously, failure to reject validity does not imply the validity of the instrument, especially for our procedure, since causal structures with valid instruments can be Markov equivalent to structures with invalid instruments (see Section EC4.1 of the electronic companion for an example). In spirit, our procedure is similar to likelihood ratio tests commonly used to test for model misspecification, except that we use the loss function from an integer program to evaluate model fit instead of a likelihood function, which is not well defined in this model space.

In CausalIP the graphical criteria in Definitions 5 can be directly expressed as the presence or absence of particular paths. Specifically, enforcing the exclusion criterion amounts to simply precomputing all paths that violate condition (2) in Definition 5 and prohibiting them from appearing in the output graph. Formally, let i, j, and k denote the instrument, treatment, and outcome nodes, respectively. Then Definition 5(2) can be enforced by adding to CausalIP the constraints

$$y_p = 0, \quad p \in \mathcal{P}_{\text{IV}}, \tag{20}$$

where $\mathcal{P}_{\text{IV}}$ contains all unblocked paths from i to k that do not contain the edge $j \rightarrow k$ (i.e., all paths that violate the exclusion criterion). Similarly, the relevance criterion (Definition 5(1)) can be enforced by ensuring the graph contains at least one edge (of any type) between the instrument and treatment, which can be done by adding the constraint

$$\sum_{e \in E_{ij}} x_e \geq 1, \tag{21}$$

where $E_{ij} = \{i \rightarrow j, i \leftarrow j, i \leftrightarrow j\}$. When the constraints (20) and (21) are added to CausalIP, we refer to the resulting space of graphs as the *restricted* model space; analogously, we refer to the full space of directed mixed graphs as the *unrestricted* model space. A similar approach can be used to enforce the criteria for conditional instruments from Definition 6, given appropriate adjustments to the set of violating paths $\mathcal{P}_{\text{IV}}$.

Our procedure for checking instrument validity is based on the bootstrap percentile method (Efron and Tibshirani 1994) and is summarized in Algorithm 4. The *bootstrap percentile value* generated by Algorithm 4 is the fraction of bootstrap repetitions in which enforcing the instrument criteria did not degrade model fit; accordingly, we interpret a small bootstrap percentile value as evidence against the candidate instrument's validity. Note that obtaining valid p-values in this setting requires an asymptotic characterization of the loss difference $L(\mathcal{G}_{IV}) - L(\mathcal{G})$, which is beyond the scope of this paper. We therefore rely on bootstrap percentile values to informally measure the extent to which a variable satisfies the graphical criteria of an instrumental variable, and caution against interpreting them as p-values.

Algorithm 4 (Path-Based Procedure for Instrument Validity)

Input: Bootstrap repetitions B, BIC complexity parameter c.

Output: Bootstrap percentile value.

1. **for** $b = 1, 2, \ldots, B$:
   - 1.1 Solve CausalIP on bootstrap sample b over unrestricted model space and get graph $\mathcal{G}^b$.
   - 1.2 Repeat for restricted model space and get graph $\mathcal{G}^b_{\text{IV}}$.
   - 1.3 Set $\delta^b = L(\mathcal{G}^b_{\text{IV}}) - L(\mathcal{G}^b)$.
2. Compute bootstrap percentile value $= (1/B)\sum_{b=1}^{B} \mathbf{1}(\delta^b \leq 0)$.

To explore our procedure, we apply it to two well-known instruments for estimating the returns to

education: the quarter-of-birth instrument from Angrist and Krueger (1991) and the proximity-to-college instrument from Card (1993).

5.3. Example 1: Quarter-of-Birth Instrument from Angrist and Krueger (1991)

The causal effect of education on income is a classical question in economics with significant policy implications, but one that is challenging to measure due to unobserved confounders (Card 1999). The remedy for confounding proposed by Angrist and Krueger (1991) is to use *quarter-of-birth* as an instrument for years of education completed. The justification for the validity of this instrument is given as follows: Because students are born year-round, the age at which students start school varies. Further, compulsory schooling laws in many states prohibit students from dropping out before they reach a certain age (e.g., their 16^{th} birthday). The combination of variability in starting ages and compulsory schooling laws effectively forces some students to complete more schooling than others, making quarter-of-birth correlated with years of education. Further, Angrist and Krueger (1991) argue that there is little reason to think quarter-of-birth would be correlated with income beyond its effect on education, thereby ensuring validity of the instrument to estimate the effect of education on income.

Angrist and Krueger (1991)'s pioneering use of quarter-of-birth as an instrument for education has led to its adoption in numerous other studies (Buckles and Hungerman 2013). Meanwhile, the validity of this instrument has been the subject of extensive debate and discussion (e.g., Bound et al. 1995, Card 1999, Staiger and Stock 1997, Angrist and Krueger 2001, Imbens and Rosenbaum 2005, Buckles and Hungerman 2013). We investigate the validity of the quarter-of-birth instrument by applying the path-based procedure described in Section 5.2, and by implementing the test proposed by Kitagawa (2015) for comparison.

5.3.1. Data and Experimental Setup. We focus on a subset of the data used in Angrist and Krueger (1991) containing information about 329,509 individuals taken from the 1980 U.S. Census.[18] There are six available variables: QOB (quarter-of-birth, an integer value between 1 and 4), EDU (years of education completed), WAGE (weekly wage), RACE (race, 1 = Black) MAR (marital status, 1 = married), and SMSA (location of residence, 1 = Metropolitan Statistical Area). To remove the effect of year-of-birth, we de-trended the data following the steps described in Angrist and Krueger (1991) and used the Bayesian Information Criterion (BIC) (see Section EC3.2 of the electronic companion) to determine the independence relations and their weights.[19] Since the outcomes of the conditional independence tests depend critically on the BIC complexity parameter c, we repeat the procedure for three different values of c. In particular, we consider $c \in \{c_{0.95}, c_1, c_{1.05}\}$, where $c_1 = 1$, $c_{0.95}$ is the value that generates 5% fewer d-separation conditions than c_1, and $c_{1.05}$ is the value that generates 5% more d-separation conditions than c_1.

For each value of the complexity parameter, we generate a bootstrap percentile value by running Algorithm 4 with $B = 50$. The graphs $\mathcal{G}$ and $\mathcal{G}_{\text{IV}}$ are generated by solving CausalIP using the EdgeGen algorithm described in Section 4, terminating after 500 seconds in each bootstrap iteration. Since quarter-of-birth is proposed as an "unconditional" instrument in Angrist and Krueger (1991), we apply Definition 5 when constructing the restricted model space.

Kitagawa's test assumes a binary treatment, so we binarized EDU by assuming that individuals with 12 or more years of education are treated, that is, the treatment is considered as obtaining high school degree, and implemented the test using the R functions used in Kitagawa (2015), again with 50 bootstrap repetitions. Kitagawa's test is parameterized by a user-defined "trimming constant" ξ, which is a regularization parameter that stabilizes a variance-based weighting scheme for the proposed test statistic. We implement Kitagawa's test using the same values for ξ used in Kitagawa (2015): $\xi \in \{0.07, 0.3, 1\}$.

5.3.2. Results. Table 4 presents the bootstrap percentile values obtained by applying our path-based procedure to the quarter-of-birth instrument from Angrist and Krueger (1991). We find the values to be highly sensitive to the specification of the complexity parameter c, and obtain a range of 0.12 to 0.52. Table 5 shows the results from applying Kitagawa's test, which generates p-values showing similar variability (0.16 to 0.9).

Our graph-based procedure also allows us to take the additional step of probing where potential violations of the criteria for valid instruments may lie. In particular, a graph constructed over the variables from Angrist and Krueger (1991) may reveal a causal path that violates the exclusion criterion, or show a weak relation to the treatment. A natural approach is to count the frequency with which each edge appears over all bootstrapped graphs in Algorithm 4, where

Table 4. Bootstrap Percentile Values from Path-Based Procedure (Algorithm 4) Applied to Quarter-of-Birth Instrument from Angrist and Krueger (1991), Based on 50 Bootstrap Repetitions

BIC penalty, c	$c_{0.95}$	$\boldsymbol{c_1}$	$c_{1.05}$
Bootstrap percentile value	0.52	0.62	0.12

Note. $c_{1.05}$ corresponds to the strongest penalty on model complexity.

Table 5. *p*-Values from Kitagawa's Test Applied to Quarter-of-Birth Instrument from Angrist and Krueger (1991), Based on 50 Bootstrap Repetitions

Trimming constant, ξ	0.07	0.3	1
p-value	0.9	0.53	0.16

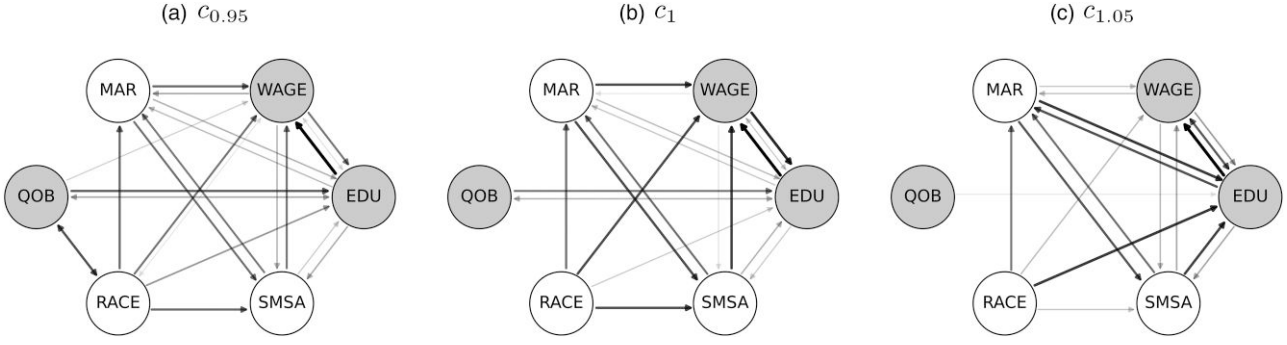
higher frequencies denote greater confidence in the causal relation implied by the edge. Figure 6 visualizes the edge frequencies over 50 bootstrap repetitions for the quarter-of-birth instrument, where thicker lines are used for higher frequency edges. Section EC4.2 of the electronic companion contains detailed tables with all edge frequencies.[20]

The results in Figure 6 provide some evidence of confounding between QOB and RACE. The bidirected edge QOB ↔ RACE appears with a frequency of 0.68 for $c = c_{0.95}$; notably, no other edge appears more frequently at this value of c (see Section EC4.2 of the electronic companion). However, the edge QOB ↔ RACE vanishes at higher values of c. Although QOB ↔ RACE only appears at low values of c, the presence of this edge threatens the validity of quarter-of-birth as an instrument for education. Because the edge RACE → WAGE appears with moderate frequency (0.54 under $c = c_{0.95}$), the edge QOB ↔ RACE violates condition (2) in Definition 5 by creating a causal path from QOB to WAGE that does not pass through EDU. Our results also show that the edge QOB → EDU appears with a frequency of 0.64 and 0.54 for $c = c_{0.95}$ and $c = c_1$, respectively, indicating that condition (1) in Definition 5 is not satisfied in half of the output graphs. For $c = c_{1.05}$, the frequency of QOB → EDU drops to 0.10, which speaks to potential weakness of QOB as an instrument for EDU, and we conjecture is responsible for the sharp drop in the bootstrap percentile value in Table 4 at $c = c_{1.05}$.

These concerns about the validity of quarter-of-birth as an instrument are not new—two well-known critiques of this instrument are presented by Bound et al. (1995) (hereafter BJB-95) and Buckles and Hungerman (2013) (BH-13). The main criticism in BJB-95 is that quarter-of-birth's association with education is so weak that even minimal confounding may lead to biased estimates, despite the large sample sizes in Angrist and Krueger (1991). As noted above, this weak association between quarter-of-birth and education is reflected in our results by the edge QOB → EDU dropping to a frequency of 0.1 for $c = c_{1.05}$, while other edges persist. More interestingly, BJB-95 also suggest that quarter-of-birth may be associated with family characteristics that are predictive of an individual's income. In particular, they argue that race may be associated with quarter-of-birth, and also point to research that finds families with high incomes are less likely to have children in the winter months (Kestenbaum 1987). These conjectures about potential confounding due to family background are rigorously examined by BH-13, who propose that maternal characteristics can explain a significant share of the association between quarter-of-birth and income. Using birth certificate and U.S. Census data, BH-13 find that children born in the winter are more likely to have mothers that are nonwhite, teenagers, and lacking a high school diploma. BH-13 state that because quarter-of-birth is associated with family background (which is itself related to income), the quarter-of-birth instrument violates the critical *exclusion* criterion. Notably, our detection of a causal relationship between RACE and QOB is well-aligned with both BJB-95 and BH-13's claims of potential confounding between quarter-of-birth and income due to race.

Beyond the effect of race, BJB-95 also argue that it is plausible that quarter-of-birth has a direct effect on income, and point to research in psychology and education for possible mechanisms. Our results do not support this claim, because we do not find quarter-of-birth to be a direct cause of income, nor do we detect any confounding between those variables other than

Figure 6. Visualization of Edge Frequencies from EdgeGen on Angrist and Krueger (1991) Data Over 50 Repetitions



Notes. Edge frequencies over 50 bootstrap repetitions of EdgeGen applied to Angrist and Krueger (1991) data for three levels of complexity penalty. Only edges with frequency ≥ 0.1 are shown.

race. Although our findings suggest that confounding between QOB and WAGE is predominantly captured by RACE, this does not rule out the possibility that there exists a weak and undetected edge between QOB and WAGE, which may still meaningfully bias estimates of the effect of EDU on WAGE.[21]

5.4. Example 2: Proximity-to-College Instrument from Card (1993)

Card (1993) proposes the presence of a college near an individual's childhood home as a binary instrument for the effect of education on income. The argument put forth is that living near a college reduces the cost of education by allowing students to live at home, making students with nearby colleges more likely to pursue higher education. Further, the exclusion criterion is claimed to be satisfied because proximity to a college should be independent of a student's unobserved ability.

5.4.1. Data and Experimental Setup. The data from Card (1993) is taken from the National Longitudinal Survey of Young Men, which conducted surveys of men aged 14–24 from 1966 through 1981. In the data set provided by Card (1993), the survey respondents' place of residence is taken from the 1966 survey, and years of education and income are taken from the 1976 survey.

Kitagawa (2015) analyses Card (1993)'s proximity-to-college instrument using his test of validity. We follow his set-up by using the following variables: PROX (proximity to 4 year college, 0 or 1), EDU (years of education completed), WAGE (wage), RACE (race, 1 = Black), SOUTH (location in 1966, 1 = lives in a Southern state), and SMSA (location in 1966, 1 = Metropolitan Statistical Area). We remove observations with missing entries, which yields 1,600 observations in total. We implement our path-based procedure in an identical manner to Section 5.3 using Algorithm 4, and reproduce the table from Kitagawa (2015) below.

5.4.2. Results. Table 6 presents the results of applying our procedure to the proximity-to-college instrument

Table 6. Bootstrap Percentile Values from Path-based Procedure (Algorithm 4) Applied to Proximity-to-College Instrument from Card (1993), Based on 50 Bootstrap Repetitions

	No covariates			With covariates		
BIC penalty, c	$c_{0.95}$	c_1	$c_{1.05}$	$c_{0.95}$	c_1	$c_{1.05}$
Bootstrap percentile value	0.26	0.08	0.10	0.88	0.72	0.64

Note. "No covariates" refers to testing proximity as an unconditional instrument; "With covariates" refers to testing proximity as a conditional instrument with set $W = \{\text{RACE}, \text{SOUTH}, \text{SMSA}\}$.

Table 7. p-Values from Kitagawa's Test Applied to Proximity-to-College Instrument from Card (1993), Based on 500 Bootstrap Repetitions

	No covariates			With covariates		
Trimming constant, ξ	0.07	0.3	1	0.07	0.3	1
p-value	0.00	0.00	0.00	0.89	0.71	0.91

Note. Table reproduced from Kitagawa (2015).

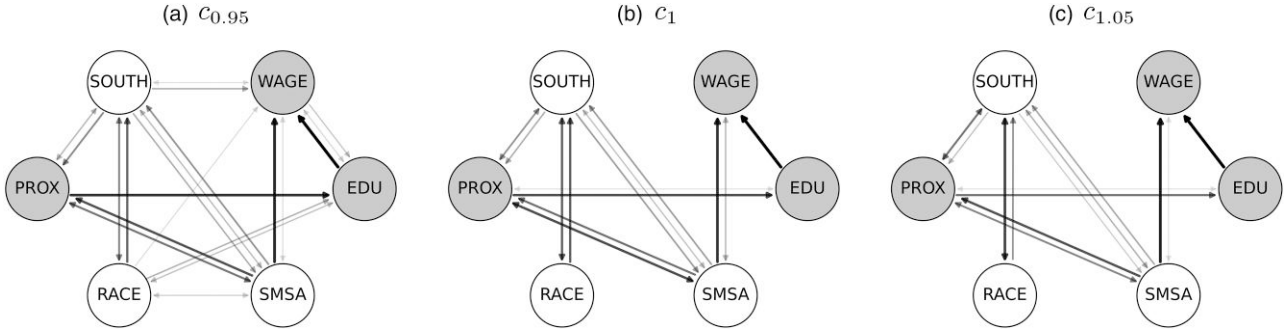
from Card (1993). Following Kitagawa (2015), we consider two settings: one in which proximity-to-college is assumed to be an unconditional instrument (i.e., Definition 5 is enforced in the restricted model space), and one in which proximity is used as a conditional instrument with conditioning set $W = \{\text{RACE}, \text{SOUTH}, \text{SMSA}\}$ (Definition 6 is enforced). As in Section 5.3, our results are qualitatively consistent with those from Kitagawa (2015), reproduced here as Table 7. In particular, when testing the validity of proximity-to-college as an unconditional instrument, Kitagawa's test produces p-values of 0.00 for all values of the trimming constant ξ, rejecting the validity of the instrument. We obtain bootstrap percentile values of 0.08 to 0.26 depending on the value of the BIC penalty c. When investigating proximity-to-college as a conditional instrument, the bootstrap percentile values returned by our approach (0.64–0.88) are comparable to the p-values from Kitagawa's test (0.71–0.91). Notably, the values from both our procedure and Kitagawa's test are relatively stable with respect to the tuning parameters c and ξ, in sharp contrast to the wide range of values obtained from testing quarter-of-birth in Section 5.3.

Analogous to Section 5.3, Figure 7 shows the edge frequencies for each value of the complexity parameter c. For all values of c, we observe the path from $\text{PROX} \leftrightarrow \text{SMSA} \rightarrow \text{WAGE}$ with moderate frequency, with $\text{PROX} \leftrightarrow \text{SMSA}$ appearing with frequency 0.48–0.54 across the three values of c and $\text{SMSA} \rightarrow \text{WAGE}$ appearing with frequency 0.82–0.88. The robustness of this path underscores the invalidity of proximity-to-college as an unconditional instrument, and thus the importance of conditioning on the location variable SMSA. Similarly, at $c = c_{0.95}$, the edges in the path $\text{PROX} \leftrightarrow \text{SOUTH} \rightarrow \text{WAGE}$ appears with frequency 0.28 and 0.32, which violates Definition 5(2), although the edge $\text{SOUTH} \rightarrow \text{WAGE}$ vanishes at higher values of c. The edge $\text{PROX} \rightarrow \text{EDU}$ appears with frequencies of $0.56 - 0.8$ across the values of c tested. This edge thus appears to be more durable than $\text{QOB} \rightarrow \text{EDU}$ in Section 5.3, which drops to a frequency of 0.1 under $c = c_{1.05}$.

5.5. Discussion

We have outlined a graph-based procedure for investigating instrument validity and applied it to the instruments from Angrist and Krueger (1991) and

Figure 7. Visualization of Edge Frequencies from EdgeGen on Card (1993) Data Over 50 Repetitions



Notes. Edge frequencies over 50 bootstrap repetitions of EdgeGen applied to Card (1993) data for three levels of complexity penalty. Only edges with frequency ≥ 0.1 are shown.

Card (1993). Our results are qualitatively consistent with the test proposed in Kitagawa (2015), although both methods are sensitive to user-specified tuning parameters, and our approach relies on the more informal bootstrap percentile method. When applied to the quarter-of-birth instrument from Angrist and Krueger (1991), Kitagawa's test produces highly variable p-values depending on the tuning parameter, but does not conclusively rejects its validity; similarly, we observe a wide range of bootstrap percentile values from our procedure. For the proximity-to-college instrument from Card (1993), both approaches generate large values when location and race variables are conditioned on; in the unconditional setting, Kitagawa's test generates p-values close to 0, while the lowest bootstrap percentile value under our test is 0.08.

For invalid instruments, our graph-based procedure can identify specific causal pathways that violate the critical exclusion criterion. Identifying these paths may also reveal covariates that can restore the validity of a proposed instrument once conditioned on, as demonstrated by the results for the Card (1993) data. While the additional insights regarding exclusion-violating paths are not obtained under Kitagawa's test, we note that they come at the cost of the higher computational burden associated with causal discovery.

Our approach has limitations. As discussed above, the output graphs are sensitive to the choice of conditional independence test and the weighting scheme. We tested sensitivity to the BIC complexity penalty c, but not to the choice of independence test; the use of other tests may alter our results. Further, there may exist multiple graphs within the Markov equivalence class of graphs implied by the input independence relations, and we only considered one of them per run in this analysis. Lastly, we terminated the algorithm after 500 seconds in order to complete the many bootstraps and settings in reasonable time. Adjusting this criterion may also affect our findings.

Our procedure is also not unique to our causal discovery method. In principle, one could develop a similar test using other causal discovery techniques. However, an effective way of encoding the instrument conditions and a representation of possible latent confounding is essential, and none of the existing causal discovery methods for this search space scale well.

6. Conclusion

In this article, we presented a new optimization-based method for *causal discovery*: the problem of learning causal structures from observational data. The key to our method is an iterative solution algorithm that makes use of the conditional independence structure in the data to identify promising edges and paths to include in the output graph. Our method is one of the few in the literature that allows for both feedback cycles and unmeasured confounding, and performs favorably compared with those that do.

We also proposed a procedure for investigating instrument validity built upon our causal discovery method. Our approach complements instrument tests from the literature by revealing the precise causal pathways that invalidate an instrumental variable, which in turn identifies conditioning variables that can transform an otherwise invalid instrument into a valid conditional instrument.

There are numerous potential directions for future research. On the algorithmic side, natural candidates include methods that efficiently select a subset of conditional independence tests to further improve scalability, and systematically exploring different weighting schemes for the input relations. Regarding the estimation of treatment effects, our focus was on instrumental variables; however, our results suggest that causal discovery methods may provide valuable structural justification for many other causal inference methods. This is especially the case for our integer programming-based framework, which can easily

incorporate structural assumptions and background information. Lastly, a sound understanding of causal structures can play a vital role in correctly evaluating counterfactuals and the development of data-driven decision-making models. As causal discovery methods and computational power more generally both continue to advance, opportunities to tractably blend the inference of causal structure with prescriptive models will invariably arise.

Endnotes

[1] We will extend the edge types to accommodate unmeasured confounding in Section 2.2.

[2] Given the correspondence between graphical structure and probability distribution, we will use the terms "node" and "variable" interchangeably.

[3] In principle there can also be edges from a node to itself, but such self-loops are redundant for linear Gaussian cyclic models that we consider here (Hyttinen et al. 2012).

[4] In the general case with arbitrary initial conditions, a sufficient and necessary condition for convergence to an equilibrium is for all eigenvalues of $\mathbf{B}$ to be less than 1 (Hyttinen et al. 2012).

[5] An extension to more general parameterizations for cyclic models is beyond the scope of this paper—see the notion of σ-separation in Forré and Mooij (2018) for a thorough treatment.

[6] Only one bi-directed edge is used to represent all possible confounders between a pair of variables. A confounder of n observed variables is represented by $\binom{n}{2}$ bi-directed edges among the n variables. See Evans (2016) for important subtle issues about this representation that, however, do not matter to the independence structure we consider here.

[7] In practice, the conditional independence relations in $P_{\mathcal{G}}(V)$ are unknown and inferred through statistical testing. Our numerical results in Section 4 considers the performance of our method with and without errors in testing.

[8] A DMG may technically contain paths with repeating nodes, due to cycles. However, as we show in Section EC1 of the electronic companion, all d-separation relations can be accurately captured by paths that conform to Definition 1. While this is a fairly technical result, it has important consequences for the correctness of our method, because it allows us to restrict attention to paths with non-repeating nodes without loss of inferential power.

[9] In the case where all input d-separation and d-connection relations are known to be jointly satisfiable, CausalIP can be expressed equivalently as a feasibility problem, that is, with no objective and constraints $z^n_{ij} = 0$, $n \in N_{ij}$, $i, j \in V$.

[10] In the acyclic, causally sufficient case (i.e., for DAGs), the features shared by Markov equivalent DAGs can be easily characterized: two DAGs are Markov equivalent if and only if they share the same *skeleton* (unoriented adjacency structure) and the same *unshielded colliders* (Verma and Pearl 1990). For the general class of DMGs that we are considering here, no such compact characterization of Markov equivalent graphs exists.

[11] To further distinguish Markov equivalent graphs requires experimental intervention, stronger background assumptions, or that one can make assumptions about the data distribution that go beyond its independence structure (e.g., about particular parameterizations; see, e.g., Eberhardt (2017) for an overview).

[12] Our solution technique is related to constraint-and-column generation methods, which are widely-used for solving large-scale integer programs, and for which many variations exist, including the branch-price-and-cut algorithm (Barnhart et al. 1998). A distinct feature of our algorithm is that the subproblem uses results of conditional independence tests to identify the columns and constraints to be introduced into the master problem. As with all integer-linear programs, branch-price-and-cut can be applied to solve CausalIP as well. See Lubbecke and Desrosiers (2005) for a review of column generation methods.

[13] Hyttinen et al. (2017) and Rantanen et al. (2020) are also relevant benchmarks here, but we do not compare against them because implementable code is not publicly available for those methods.

[14] We set M to $\sum_{i,j \in V} |A_{ij}|$ where $|A_{ij}|$ is the cardinality of set A_{ij}.

[15] Computing the weights this way is computationally intensive, so for the largest instances with $|V| = 15$ we followed the suggestion by Hyttinen et al. (2014) to use the Bayesian Information Criterion (BIC) to determine the weights more efficiently.

[16] Relatedly, there is extensive work on the pitfalls of using weak (i.e., low correlation) instruments and potential remedies (Bound et al. 1995, Staiger and Stock 1997, Stock et al. 2002, Murray 2006).

[17] Angrist and Krueger (2001) write "good instruments often come from detailed knowledge of the economic mechanism and institutions", and Imbens and Rosenbaum (2005) write "finding instruments is an art rather than a science."

[18] Angrist and Krueger (1991) repeat their analysis for three cohorts separately—those born in the 1930s, 1940s, and 1950s—obtaining similar results across all three cohorts. For conciseness in our presentation, we use data from just the first cohort.

[19] As suggested in Hyttinen et al. (2014), BIC weights are a fast approximation of the log-weights discussed in Section 4.4, which are computationally intensive to compute for large data sets like in Angrist and Krueger (1991).

[20] Strobl et al. (2019) develop analytical upper bounds on the p-values for individual edges in DAGs, which depend on the significance level of the underlying conditional independence tests. However, their method of generating edge-specific p-values does not hold for the more general class of DMGs, and we are unaware of any that do.

[21] See Belloni et al. (2014) for a general related discussion.

References

Angrist JD, Krueger AB (1991) Does compulsory school attendance affect schooling and earnings? *Quart. J. Econom.* 106(4):979–1014.

Angrist JD, Krueger AB (2001) Instrumental variables and the search for identification: From supply and demand to natural experiments. *J. Econom. Perspect.* 15(4):69–85.

Angrist JD, Imbens GW, Rubin DB (1996) Identification of causal effects using instrumental variables. *J. Amer. Statist. Assoc.* 91(434):444–455.

Balke A, Pearl J (1997) Bounds on treatment effects from studies with imperfect compliance. *J. Amer. Statist. Assoc.* 92(439):1171–1176.

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* 46(3):316–329.

Bartlett M, Cussens J (2017) Integer linear programming for the Bayesian network structure learning problem. *Artificial Intelligence* 244:258–271.

Belloni A, Chernozhukov V, Hansen C (2014) Inference on treatment effects after selection among high-dimensional controls. *Rev. Econom. Stud.* 81(2):608–650.

Bound J, Jaeger DA, Baker RM (1995) Problems with instrumental variables estimation when the correlation between the instruments and the endogenous explanatory variable is weak. *J. Amer. Statist. Assoc.* 90(430):443–450.

Bowden RJ, Turkington DA (1990) *Instrumental Variables*, vol. 8 (Cambridge University Press, Cambridge, UK).

Buckles KS, Hungerman DM (2013) Season of birth and later outcomes: Old questions, new answers. *Rev. Econom. Statist.* 95(3):711–724.

Card D (1993) Using geographic variation in college proximity to estimate the return to schooling. NBER Working Paper No. 4483, National Bureau of Economic Research, Cambridge, MA.

Card D (1999) The causal effect of education on earnings. *Handbook Labor Econom.* 3:1801–1863.

Chen R, Dash S, Gao T (2021) Integer programming for causal structure learning in the presence of latent variables. *Internat. Conf. Machine Learn.* (PMLR, New York), 1550–1560.

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

Cussens J (2011) Bayesian network learning with cutting planes. *Proc. Twenty-Eighth Conf. Uncertainty Artificial Intelligence* (AUAI Press, Arlington, VA), 153–160.

Eberhardt F (2017) Introduction to the foundations of causal discovery. *Internat. J. Data Sci. Anal.* 3(2):81–91.

Efron B, Tibshirani RJ (1994) *An Introduction to the Bootstrap* (Chapman and Hall/CRC, Norwell, MA).

Evans RJ (2016) Graphs for margins of Bayesian networks. *Scand. J. Statist.* 43(3):625–648.

Forré P, Mooij JM (2018) Constraint-based causal discovery for nonlinear structural causal models with cycles and latent confounders. Preprint, submitted July 9, https://arxiv.org/abs/1807.03024.

Geiger D, Pearl J (1988) *On the Logic of Influence Diagrams* (University of California (Los Angeles), Computer Science Department, Los Angeles).

Geiger D, Verma T, Pearl J (1990) Identifying independence in Bayesian networks. *Networks* 20(5):507–534.

Heckman JJ, Vytlacil E (2005) Structural equations, treatment effects, and econometric policy evaluation. *Econometrica* 73(3):669–738.

Hyttinen A, Eberhardt F, Hoyer PO (2012) Learning linear cyclic causal models with latent variables. *J. Machine Learn. Res.* 13(1): 3387–3439.

Hyttinen A, Eberhardt F, Järvisalo M (2014) Constraint-based causal discovery: Conflict resolution with answer set programming. *Proc. Thirtieth Conf. Uncertainty Artificial Intelligence* (AUAI Press, Corvallis, OR), 340–349.

Hyttinen A, Saikko P, Järvisalo M (2017) A core-guided approach to learning optimal causal graphs. *Proc. 26th Internat. Joint Conf. Artificial Intelligence (IJCAI 2017)* (AAAI Press, Palo Alto, CA).

Hyttinen A, Hoyer PO, Eberhardt F, Järvisalo M (2013) Discovering cyclic causal models with latent variables: A general SAT-based procedure. *Proc. Twenty-Ninth Conf. Uncertainty Artificial Intelligence* (AUAI Press, Corvallis, OR), 301–310.

Imbens GW, Rosenbaum PR (2005) Robust, accurate confidence intervals with a weak instrument: Quarter of birth and education. *J. Royal Statist. Soc. Ser. A* 168(1):109–126.

Jaakkola T, Sontag D, Globerson A, Meila M (2010) Learning Bayesian network structure using LP relaxations. *Proc. Thirteenth Internat. Conf. Artificial Intelligence Statist.* (PMLR, New York), 358–365.

Kédagni D, Mourifié I (2020) Generalized instrumental inequalities: Testing the instrumental variable independence assumption. *Biometrika* 107(3):661–675.

Kestenbaum B (1987) Seasonality of birth: Two findings from the decennial census. *Soc. Biol.* 34(3–4):244–248.

Kitagawa T (2015) A test for instrument validity. *Econometrica* 83(5): 2043–2063.

Kucukyavuz S, Shojaie A, Manzour H, Wei L, Wu H-H (2020) Consistent second-order conic integer programming for learning Bayesian networks. Preprint, submitted May 29, https://arxiv.org/abs/2005.14346.

Lubbecke ME, Desrosiers J (2005) Selected topics in column generation. *Oper. Res.* 53(6):1007–1023.

Maathuis MH, Colombo D, Kalisch M, Bühlmann P (2010) Predicting causal effects in large-scale systems from observational data. *Nat. Methods* 7(4):247–248.

Magliacane S, Claassen T, Mooij JM (2016) Ancestral causal inference. *Adv. Neural Inform. Processing Systems* 29:4466–4474.

Manzour H, Küçükyavuz S, Wu H-H, Shojaie A (2021) Integer programming for learning directed acyclic graphs from continuous data. *Inform. J. Optim.* 3(1):46–73.

Meek C (1995) Strong completeness and faithfulness in Bayesian networks. *Proc. Eleventh Conf. Uncertainty Artificial Intelligence* (Morgan Kaufmann Publishers, San Francisco), 411–418.

Mourifié I, Wan Y (2017) Testing local average treatment effect assumptions. *Rev. Econom. Statist.* 99(2):305–313.

Murray MP (2006) Avoiding invalid instruments and coping with weak instruments. *J. Econom. Perspect.* 20(4):111–132.

Park YW, Klabjan D (2017) Bayesian network learning via topological order. *J. Machine Learn. Res.* 18(1):3451–3482.

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

Rantanen K, Hyttinen A, Järvisalo M (2020) Discovering causal graphs with cycles and latent confounders: An exact branch-and-bound approach. *Internat. J. Approx. Reason.* 117:29–49.

Rantanen K, Hyttinen A, Järvisalo M (2018) Learning optimal causal graphs with exact search. *Internat. Conf. Probabilistic Graphical Models* (PMLR, New York), 344–355.

Richardson T (1996) Feedback models: Interpretation and discovery. PhD thesis, Carnegie Mellon, Pittsburgh.

Solus L, Wang Y, Uhler C (2021) Consistency guarantees for greedy permutation-based causal inference algorithms. *Biometrika* 108(4): 795–814.

Spirtes P (1995) Directed cyclic graphical representations of feedback models. *Proc. Eleventh Conf. Uncertainty Artificial Intelligence* (Morgan Kaufmann Publishers, Inc., San Francisco), 491–498.

Spirtes P, Zhang K (2016) Causal discovery and inference: Concepts and recent methodological advances. *Applied Informatics*, vol. 3 (SpringerOpen, Cham, Switzerland), 1–28.

Spirtes P, Glymour CN, Scheines R, Heckerman D, Meek C, Cooper G, Richardson T (2000) *Causation, Prediction, and Search* (MIT Press, Cambridge, MA).

Staiger D, Stock JH (1997) Instrumental variables regression with weak instruments. *Econometrica* 65(3):557–586.

Stock J (2002) Instrumental variables in economics and statistics. *International Encyclopedia of the Social Sciences* (Macmillan Reference USA, New York).

Stock JH, Wright JH, Yogo M (2002) A survey of weak instruments and weak identification in generalized method of moments. *J. Bus. Econom. Statist.* 20(4):518–529.

Strobl EV, Spirtes PL, Visweswaran S (2019) Estimating and controlling the false discovery rate of the pc algorithm using edge-specific p-values. *ACM Trans. Intell. Syst. Tech.* 10(5):1–37.

Teramoto R, Saito C, Shin-ichi F (2014) Estimating causal effects with a non-paranormal method for the design of efficient intervention experiments. *BMC Bioinformatics* 15(1):1–14.

Triantafillou S, Tsamardinos I (2015) Constraint-based causal discovery from multiple interventions over overlapping variable sets. *J. Machine Learn. Res.* 16(1):2147–2205.

Uhler C, Raskutti G, Bühlmann P, Yu B (2013) Geometry of the faithfulness assumption in causal inference. *Ann. Statist.* 41(2):436–463.

Verma T, Pearl J (1990) *Equivalence and Synthesis of Causal Models* (UCLA, Computer Science Department, Los Angeles).

Zhalama JZ, Eberhardt F, Mayer W (2017) SAT-based causal discovery under weaker assumptions. *Proc. Thirty-Third Conf. Uncertainty Artificial Intelligence* (AUAI Press, Corvallis, OR).

Zhang J, Spirtes P (2002) Strong faithfulness and uniform consistency in causal inference. *Proc. Nineteenth Conf. Uncertainty Artificial Intelligence* (Morgan Kaufmann Publishers, San Francisco), 632–639.

E-companion to Discovering Causal Models with Optimization: Confounders, Cycles, and Instrument Validity

Frederick Eberhardt

Division of Humanities and Social Sciences, California Institute of Technology

fde@caltech.edu

Nur Kaynar

Samuel Curtis Johnson Graduate School of Management, Cornell University

nur.kaynar@cornell.edu

Auyon Siddiq

Anderson School of Management, University of California, Los Angeles

auyon.siddiq@anderson.ucla.edu

EC.1. Formalization of Paths

EC.1.1. Paths Without Cycles

We provide here a more complete definition of paths in directed mixed graphs, which may contain paths with repeating nodes due to cycles. In the lemma that follows, we show that such paths can be eliminated from consideration without altering d-separation relations between nodes. This result serves as justification for the simplified path definition given in Definition 1.

DEFINITION EC.1 (PATH). Given a node set V, a set of edge-types $T = \{\rightarrow, \leftarrow, \leftrightarrow\}$ and an edge set E of triples (v_1, t, v_2) with $v_1, v_2 \in V$ and $t \in T$, we define a path p_{ij}^{-} between node i and node j with $i, j \in V, i \neq j$, as a sequence of edges $p_{ij}^{-} = (e_1, \ldots, e_\ell)$ such that

(i) the edges in the path are in the edge set: $e_k \in E$ for all $1 \leq k \leq \ell$,

(ii) the path starts with node i: $e_1 = (i, t, v)$ for some $v \in V \setminus \{i\}$ and $t \in T$,

(iii) the path ends with node j: $e_\ell = (v, t, j)$ for some $v \in V \setminus \{j\}$ and $t \in T$,

(iv) consecutive edges on the path are connected: for all $e_k, e_{k+1} = (v_1, t, v_2)(u_1, t', u_2) \in p_{ij}^{-}$ with $1 \leq k < \ell$, we have $v_2 = u_1$.

A *directed path* from i to j is a path that only has edge-type $T = \{\rightarrow\}$, i.e. all edges point away from i and towards j along the path. So, a node j is a *descendant* of i if there is a directed path from i to j.

We say that an occurrence of a node $v \in V \setminus \{i, j\}$ on a path p_{ij} or on an appendage is when v is the endpoint of one edge and the starting point of the subsequent edge. In addition, i and j occur once at the beginning and end of the path, respectively. A node repeats on a path (appendage) if

it occurs more than once on the path (appendage). In that case the path (appendage) is said to contain a cycle.

We now show that with respect to the d-separation and d-connection relations, we can ignore cycles on a path. That is, if a path with a cycle is d-connecting, then there is a path without the cycle that is also d-connecting.

LEMMA EC.1. *Let path p_{ij} be a path between nodes $i, j \in V$ according to Definition EC.1. If path p_{ij} is unblocked with respect to conditioning set $C \subseteq V \setminus \{i, j\}$ in directed mixed graph $\mathcal{G} = (V, E)$ and has repeating nodes, then there exists a path p^*_{ij} without any repeating nodes in $\mathcal{G}$ that is also unblocked with respect to C.*

Proof of Lemma EC.1. Suppose there is a path p with repeating nodes between variables $i, j \in V$ in graph $\mathcal{G} = (V, E)$ that is unblocked with respect to conditioning set $C \subseteq V \setminus \{i, j\}$. Following from Definition 2, every collider k on the path p is in C or has a descendant in C, and no other nodes on the path are in C. Note that since path p is unblocked with respect to C, Definition 2 implies that the same node cannot be a collider and a noncollider at the same time on path p.

Now we show we can find a shorter path p^* in $\mathcal{G}$ that is also unblocked with respect to C. Note that path p includes the same node more than once by construction. Without loss of generality, let node l repeat on path p more than once. Let l_1 represent the *first* occurrence of node l on path p and let l_2 represent the *last* occurrence of node k on path p. Let p_{i-l_1} represent the subpath between i and l_1 on path p. Similarly, let p_{l_2-j} represent the subpath between l_2 and j on path p. Note that both the last node on path p_{i-l_1} and the first node on path p_{l_2-j} are node l. Hence, we can obtain a path p^* where node l does not repeat by combining p_{i-l_1} and p_{l_2-j} together.

Next we show that path p^* is unblocked with respect to C. To do so, we need to show every collider k on the path p^* is in C or has a descendant in C, and no other nodes on the path p^* are in C by Definition 2. Notice that if node $k \neq l$ is a collider on path p^*, then it must be a collider on path p by construction. Since path p is unblocked with respect to C, i.e. every collider k on the path p is in C or has a descendant in C, it follows that every collider $k \neq l$ on the path p^* is in C or has a descendant in C. Similarly, if node $k \neq l$ is a noncollider on path p^*, then it is a noncollider on path p^* by construction. Since path p is unblocked with respect to C, i.e. none of the noncolliders on path p is in C, it follows that if node $k \neq l$ is a noncollider on path p^*, then $k \notin C$.

Lastly, we need to consider node l. Note that node l is on path p^* by construction. Case 1. Node l is a collider on both path p and p^*. If node l is a collider on path p, then l is in C or has a descendant in C. Therefore, having node l as a collider on path p^* does not block path p^*. Case 2. Node l is a noncollider on both path p and p^*. If node l is a noncollider on path p, then l is not in C. Therefore, having node l as a noncollider on path p^* does not block path p^*. Case 3. Node l is a collider on

path p and a noncollider on path p^*. If node l is a collider on path p, then path p must have the following form: $i \cdots * \rightarrow l \leftarrow * \cdots * \rightarrow l \leftarrow * \ldots j$, where $*$ represents that an edge can have an arrow end or a tail end. This implies that path p^* must have the following form $i \cdots * \rightarrow l \leftarrow * \ldots j$, hence l cannot be a noncollider on path p^*. Case 4. Node l is a noncollider on path p and a collider on path p^*. Then path p must have the following form: $i \cdots * \rightarrow l \rightarrow \cdots * \leftarrow l \leftarrow * \ldots j$. This path must contain a collider between the instances of l. Let node c be this collider on path p. Note that it follows that c is a descendant of l. Since node c is a collider on path p and since path p is unblocked with respect to conditioning set C, then it follows either (i) $c \in C$ or c has a descendant in C. Using this, we now show p^* is unblocked with respect to C. By construction, p^* has the following forms: $i \cdots * \rightarrow l \leftarrow * \ldots j$. Since l is a collider on path p^* and c is a descendant of l, following from (i) and (ii), C includes a descendant of l. Therefore, having node l as a collider on path p^* does not block path p^*. If there is more than one collider between the instances of l in p, this reasoning can be repeated for each collider.

Similarly, we can repeat the above procedure until there are no repeating nodes. Hence it follows we can construct a path p^* without any repeating nodes between (i, j) in $\mathcal{G}$ such that p^* is unblocked with respect to C. □

EC.1.2. Appendages Without Cycles

Here we describe how appendages without cycles are sufficient for capturing all possible d-connections between variables.

Definition EC.2 (appendage). Given a path p from i to j defined according to Definition 1, let $col_{p_{ij}}$ store the colliders on path p_{ij}. An appendage of p_{ij} is a directed path from a collider $c \in col_p$ to another node $k \in V \setminus \{i, j, c\}$.

We only need to consider directed paths as appendages because the definition of blocked paths only considers descendants of colliders. In order to capture d-connections due to conditioning on descendants of a collider, we define the notion of an *extended path*.

Definition EC.3 (extended path). Let path p be a path generated according to Definition 1. The set of extended paths of p contains p and all combinations of p with its appendages.

Using similar arguments to Lemma EC.1, one can easily show it is enough to consider appendages without cycles. Let path p_{ij}^+ be an extended path between nodes $i, j \in V$ according to Definition EC.3. Suppose extended path p_{ij}^+ is unblocked with respect to conditioning set $C \subseteq V \setminus \{i, j\}$ in directed mixed graph $\mathcal{G} = (V, E)$ and one of its appendages has a cycle. Then there exists another unblocked extended $p_{ij}^{+^*}$ with appendages without cycles in $\mathcal{G}$. We formalize these observations that are rooted in Lemma EC.1 in Corollary EC.1 below (stated without proof).

COROLLARY EC.1. *Let $P_{ij}(E,|V|-1)$ store all extended paths with acyclic appendages between variables i and j generated according to Definition EC.3 over $\mathcal{G}=(V,E)$ with finite E. Then, $P_{ij}(E,|V|-1)$ is finite, the paths and extended paths it contains are of finite length, and it captures all possible d-connections between i and j.*

EC.2. CausalIP Formulation

EC.2.1. Continuous Relaxation of Error Variables z

LEMMA EC.2. *Let $(\tilde{\mathbf{x}},\tilde{\mathbf{y}},\tilde{\mathbf{z}})$ be an optimal solution to* CausalIP$(\mathcal{P}(\tilde{E},\zeta))$. *Then $\tilde{z}^n_{ij}\in\{0,1\}$ for $i,j\in V, n\in N_{ij}$.*

Proof of Lemma EC.2. Notice variable z appears only on constraints (5) and (6) and the same variable z^n_{ij} does not repeat in both constraints (5) and (6) as $N^I_{ij}\cap N^D_{ij}=\emptyset$. Let's first focus on z^n_{ij} for $i,j\in V, n\in N^I_{ij}$ that appears on constraint (5). Since $y_p\in\{0,1\}$ and $\alpha^n_{ijp}\in\{0,1\}$, $\alpha^n_{ijp}y_p\in\{0,1\}$ for all $i,j\in V, n\in N^I_{ij}$, $p\in P_{ij}(\tilde{E},\zeta)$. Since CausalIP$(\mathcal{P}(\tilde{E},\zeta))$ minimizes $\sum_{i,j\in V}\sum_{n\in N_{ij}}\omega^n_{ij}z^n_{ij}$ and $\boldsymbol{\omega}>0$, constraint (5) forces z^n_{ij} to take value 0 if $\alpha^n_{ijp}y_p=0$ for all $p\in P_{ij}(\tilde{E},\zeta)$, and to take value 1 if $\alpha^n_{ijp}y_p=1$ for at least one $p\in P_{ij}(\tilde{E},\zeta)$. Let's now focus on z^n_{ij} for $i,j\in V, n\in N^D_{ij}$ that appears on constraint (6). Since $y_p\in\{0,1\}$ and $\alpha^n_{ijp}\in\{0,1\}$, $\sum_{p\in P_{ij}(\tilde{E},\zeta)}\alpha^n_{ijp}y_p$ will be an integer for all $i,j\in V, n\in N^I_{ij}$. This implies that the left hand side of constraint (6) is always an integer. Let's now consider two scenarios where $\sum_{p\in P_{ij}(\tilde{E},\zeta)}\alpha^n_{ijp}y_p$ is (i) equal to zero or (ii) greater than or equal to one. (i) If $\sum_{p\in P_{ij}(\tilde{E},\zeta)}\alpha^n_{ijp}y_p$ is equal to 0, then constraint (6) becomes $z^n_{ij}\geq 1$. When this combined with the objective function with $\boldsymbol{\omega}>0$ will ensure $z^n_{ij}=1$. (ii) If $\sum_{p\in P_{ij}(\tilde{E},\zeta)}\alpha^n_{ijp}y_p$ is greater than or equal to 1, then constraint (6) becomes $z^n_{ij}\geq 1-K$ where $K=\sum_{p\in P_{ij}(\tilde{E},\zeta)}\alpha^n_{ijp}y_p\geq 1$. Hence the constraint (6) combined with the objective function with $\boldsymbol{\omega}>0$ will ensure $z^n_{ij}=0$. □

EC.2.2. Enforcing Acyclicity and Causal Sufficiency

Our main formulation presented in §3 assumes the presence of latent confounders and feedback cycles. However, we can naturally assume causal sufficiency or exclude cycles within our modeling framework. Let $E^s=\{i\leftarrow j, i\rightarrow j,\ \forall i,j\in V: i\neq j\}$ be the set of all possible directed edges under the assumption of causal sufficiency (i.e., no bi-directed edges). Then solving CausalIP(E^s) ensures that unobserved confounders are not allowed.

To exclude cycles, we follow the constraints provided in Cussens (2012) and Jaakkola et al. (2010). These constraints are based on the observation that if cycles are not permitted, any subset of vertices in a graph must contain at least one node that has no parent in that subset. Let $R_i=\{C\mid C\subseteq V\setminus i\}$ be all possible subsets of V that exclude i. Let R^k_i be the k^{th} set in R_i, and let K_i index the sets in R_i. Note that exactly one set in R_i must be the set of parent nodes of i. Accordingly, let τ^k_i be

a binary decision variable where $\tau_i^k = 1$ if $R_i^k \in R_i$ is the parent set of node i, and $\tau_i^k = 0$ otherwise. Next, let E_i^k be the set of all incoming edges to i from nodes in R_i^k, and let ρ_i^k be the number of nodes in R_i^k. Then, we can eliminate cycles by adding the following constraints to CausalIP:

$$\tau_i^k \leq x_e, \quad e \in E_i^k,\ k \in K_i,\ i \in V, \tag{EC.1a}$$

$$\tau_i^k \geq \sum_{e \in E_i^k} x_e - \sum_{e' \in E \setminus E_i^r} x_{e'} - \rho_i^k + 1, \quad i \in V,\ k \in K_i, \tag{EC.1b}$$

$$\sum_{k \in K_i} \tau_i^k = 1, \quad i \in V, \tag{EC.1c}$$

$$\sum_{i \in C} \sum_{\substack{k \in K_i: \\ R_i^k \cap C = \emptyset}} \tau_i^k \geq 1, \quad C \subseteq V. \tag{EC.1d}$$

Constraint (EC.1a) ensures R_i^k can only be the parent set of i if the edge $j \to i$ is present for each $j \in R_i^k$. Constraint (EC.1b) ensures that if $j \to i$ is present for all $j \in R_i^k$ and if $j \to i$ is not present for all j such that $j \notin R_i^k$, then R_i^k must be the set of parents of node i. Constraint (EC.1c) ensures that only one set in R_i can be the parent set of node $i \in V$. Constraint (EC.1d) is the directed cycle elimination constraint, which ensures that all subsets of V must contain at least one node who has no parent in that subset.

Our formulation also allows for integrating background knowledge; the presence or absence of specific edges or paths can be easily encoded via constraints on the $\mathbf{x}$ variable. Furthermore, sparsity constraints, such as maximum degree of nodes (see Claassen et al. (2013) for an example) can be easily incorporated into the model. This flexibility may be especially useful in applications where significant domain knowledge is available.

EC.3. Graph Generation, Node Degree Distributions, and Weighting Scheme

Here we provide additional details regarding the experiments in §4.4 and §5.

EC.3.1. Generation of Random Graphs

For each set of experiments in §4.4, we generate 25 directed mixed graphs. In each case, the graph over $|V|$ nodes is generated by assigning node degrees to each node in the graph uniformly from $\{1, \ldots, maxDegree\}$ and then randomly adding edges to the graph by selecting uniformly among all possible directed and bi-directed edges in the graph until each node degree is satisfied while not exceeding $maxDegree$. If needed, the node degree of one node that is not at $maxDegree$ is increased by 1 to accommodate the last edge. Figure EC.1 shows the empirical distribution over node degree for the instances with $|V| \in \{5, 10, 15, 20, 30, 40, 50\}$.

EC.3.2. Bayesian Information Criterion (BIC) Weighting Scheme

When the number of nodes or the sample size is large, the log-weights used in Hyttinen et al. (2014) become computationally prohibitive to compute. A fast approximation used in the causal discovery literature, including Hyttinen et al. (2014), is the Bayesian Information Criterion (BIC). We use BIC-based weights for all instances in §4.4 where $|V| \geq 15$ and for all experiments in §5 (due to the large sample size in Angrist and Krueger (1991)).

For a linear model that predicts the variable i from covariates C, the BIC is given by

$$BIC = -2\ln(\hat{L}) + ck\ln(n),$$

where $\hat{L}$ is the maximum likelihood of the model, $k = |C|$ is the number of predictors, and n is the sample size. The parameter c is an additional complexity parameter, where setting $c = 1$ recovers the classical BIC score, and larger values of c correspond to stronger penalty on complexity. Thus, the BIC of a given model is a composite score that balances model fit and complexity, with lower BIC values being preferred. In §5, we examine how adjusting the parameter c influences our proposed procedure for investigating instrument validity.

For each pair of variables (i, j) and conditioning set $C_{ij}^n \in A_{ij}$, we use BIC scores to simultaneously determine d-separation (i.e., whether $i \perp j | C_{ij}^n$ or $i \not\perp j | C_{ij}^n$) and the associated weight ω_{ij}^n. Specifically, given variables (i, j) and covariates C_{ij}^n, we compute two BIC scores: one associated with predicting i from C_{ij}^n and another from predicting i from $\{j \cup C_{ij}^n\}$. If the latter score is worse (i.e., higher), we conclude that $i \perp j | C_{ij}^n$ and $C_{ij}^n \in I_{ij}$; otherwise, we conclude $i \not\perp j | C_{ij}^n$ and $C_{ij}^n \in D_{ij}$. The intuition is that a degradation in BIC after adding j as a predictor suggests j carries little additional information about i that is not already carried by C_{ij}^n. The weight ω_{ij}^n is then given by the magnitude of the difference in the two BIC scores.

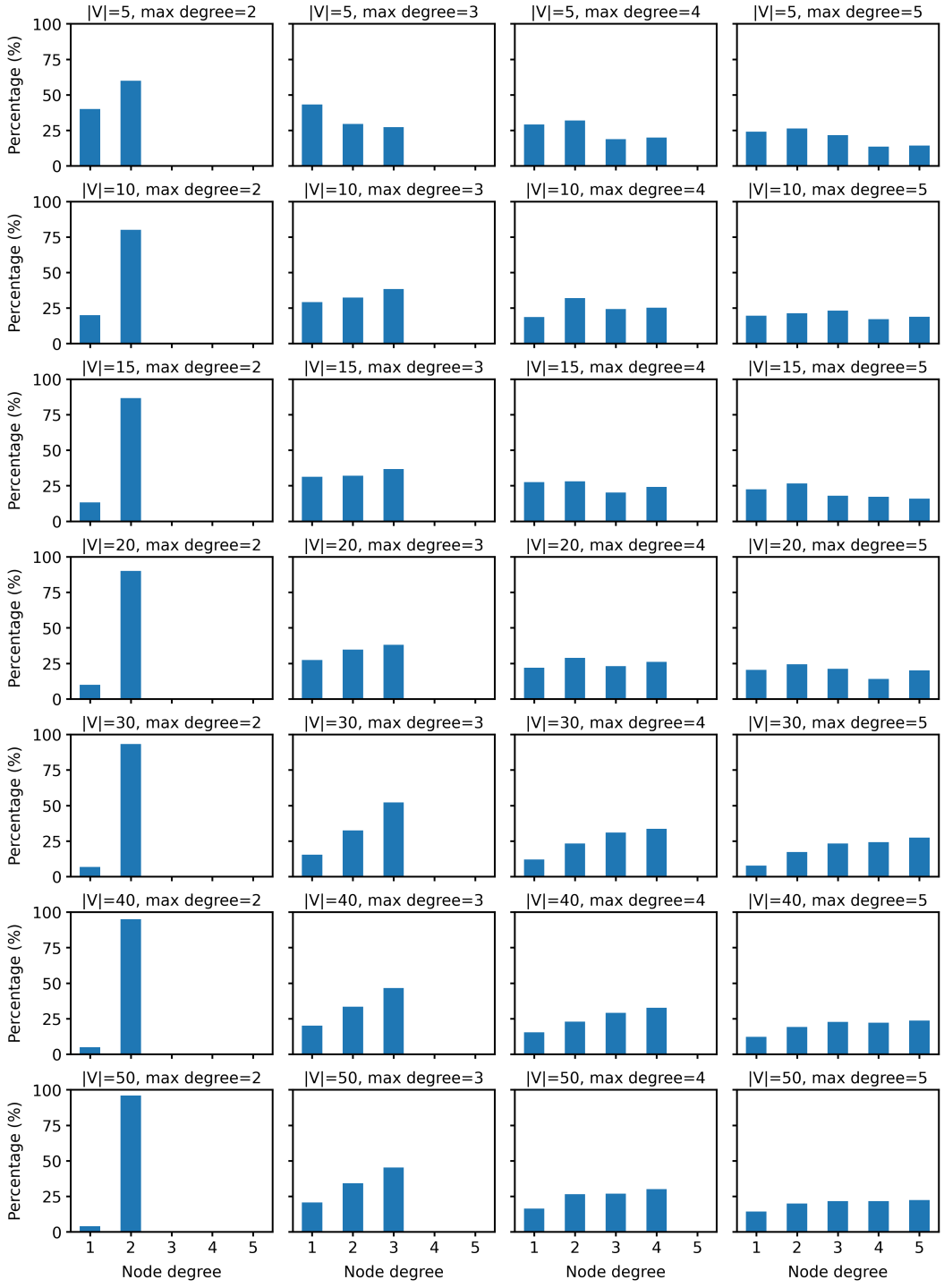


Figure EC.1 Empirical distributions of node degree for randomly generated graphs in §4.4.

EC.4. Additional Results for §5

EC.4.1. Instrument Validity and Markov Equivalence

In general, a failure to reject the null hypothesis in our path-based procedure does not imply validity of the instrument. Figure EC.2 illustrates one such instance – graphs (a) and (b) are Markov equivalent, where the instrument in graph (a) is valid, but the instrument in graph (b) is invalid as a result of violating the exclusion restriction (condition (2) of Definition 5). Since our method returns a graph within the Markov equivalence class of the data-generating graph, it may infer graph (a) when the true graph is (b).

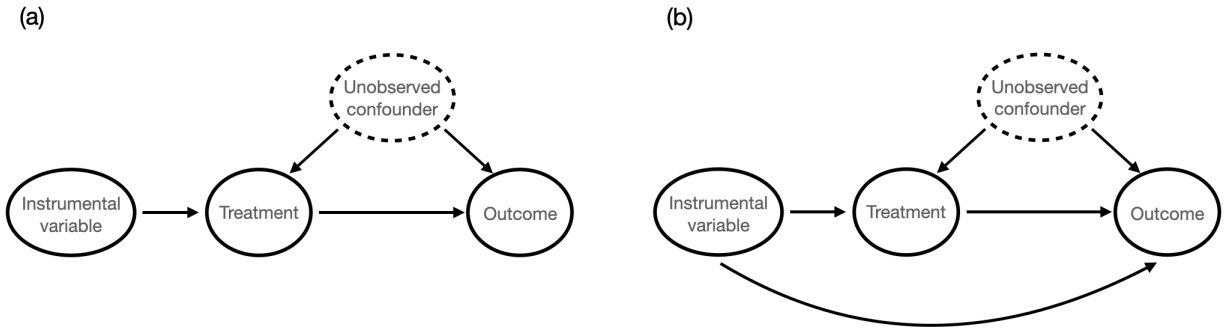


Figure EC.2 Two Markov equivalent graphs. In (a) the instrument is valid, in (b) it is invalid.

EC.4.2. Edge Frequency Tables

This section presents edge frequency tables corresponding to the graphs in Figures 6 and 7, separated into directed and bi-directed edges. All frequencies are normalized so that 1 indicates that the edge appeared in the output graph of all 50 bootstrap repetitions. Note that higher values of the BIC complexity penalty c leads to sparser graphs, where $c = c_{0.95}$ is the smallest penalty (that yields 5% fewer d-separation conditions than $c = c_1$) and $c = c_{1.05}$ is the largest penalty (that yields 5% more d-separations than $c = c_1$).

	EDU	WAGE	MAR	QOB	RACE	SMSA
EDU	-	1	0.26	0	0	0.32
WAGE	0.54	-	0.34	0	0	0.3
MAR	0.3	0.6	-	0	0	0.56
QOB	0.64	0.18	0.06	-	0	0.04
RACE	0.4	0.54	0.6	0	-	0.6
SMSA	0.2	0.54	0.52	0	0	0

(a) Directed edge frequency ($c = c_{0.95}$).

	EDU	WAGE	MAR	QOB	RACE	SMSA
EDU	-	0.12	0	0.34	0.06	0.02
WAGE	0.12	-	0.04	0	0.10	0.04
MAR	0.00	0.04	-	0	0.06	0.02
QOB	0.34	0.00	0	-	0.68	0
RACE	0.06	0.10	0.06	0.68	-	0.04
SMSA	0.02	0.04	0.02	0	0.04	-

(b) Bi-directed edge frequency ($c = c_{0.95}$).

	EDU	WAGE	MAR	QOB	RACE	SMSA
EDU	-	1	0.22	0	0	0.2
WAGE	0.84	-	0.1	0	0	0.12
MAR	0.26	0.7	-	0	0	0.7
QOB	0.54	0.02	0	-	0	0
RACE	0.2	0.7	0.58	0	-	0.7
SMSA	0.3	0.74	0.56	0	0	-

(c) Directed edge frequency ($c = c_1$).

	EDU	WAGE	MAR	QOB	RACE	SMSA
EDU	-	0.2	0	0.3	0	0
WAGE	0.2	-	0	0	0	0
MAR	0	0	-	0	0	0
QOB	0.3	0	0	-	0	0
RACE	0	0	0	0	-	0
SMSA	0	0	0	0	0	-

(d) Bi-directed edge frequency ($c = c_1$).

	EDU	WAGE	MAR	QOB	RACE	SMSA
EDU	-	1	0.68	0	0	0.34
WAGE	0.38	-	0.24	0	0	0.28
MAR	0.78	0.22	-	0	0	0.66
QOB	0.1	0	0	-	0	0
RACE	0.78	0.26	0.54	0	-	0.22
SMSA	0.74	0.34	0.56	0	0	-

(e) Directed edge frequency ($c = c_{1.05}$).

	EDU	WAGE	MAR	QOB	RACE	SMSA
EDU	-	0.64	0	0.04	0	0
WAGE	0.64	-	0	0	0	0
MAR	0	0	-	0	0	0
QOB	0.04	0	0	-	0	0
RACE	0	0	0	0	-	0
SMSA	0	0	0	0	0	-

(f) Bi-directed edge frequency ($c = c_{1.05}$).

TABLE 8. Normalized edge frequencies for Angrist and Krueger (1991) data.

	PROX	EDU	SOUTH	SMSA	WAGE	RACE
PROX	-	0.8	0	0	0.04	0
EDU	0	-	0.04	0.02	1	0
SOUTH	0.46	0.02	-	0.32	0.32	0
SMSA	0.68	0.04	0.42	-	0.86	0
WAGE	0	0.22	0	0	-	0
RACE	0	0.3	0.62	0.1	0.14	-

(a) Directed edge frequency ($c = c_{0.95}$).

	PROX	EDU	SOUTH	SMSA	WAGE	RACE
PROX	-	0.08	0.28	0.54	0.04	0
EDU	0.08	-	0	0	0.16	0.24
SOUTH	0.28	0	-	0.26	0.12	0.48
SMSA	0.54	0	0.26	-	0.14	0.18
WAGE	0.04	0.16	0.12	0.14	-	0.08
RACE	0	0.24	0.48	0.18	0.08	-

(b) Bi-directed edge frequency ($c = c_{0.95}$).

	PROX	EDU	SOUTH	SMSA	WAGE	RACE
PROX	-	0.7	0	0	0.02	0
EDU	0	-	0.02	0.02	1	0
SOUTH	0.32	0	-	0.38	0.06	0
SMSA	0.56	0.02	0.38	-	0.82	0
WAGE	0	0.08	0	0	-	0
RACE	0	0.06	0.64	0	0.02	-

(c) Directed edge frequency for ($c = c_1$).

	PROX	EDU	SOUTH	SMSA	WAGE	RACE
PROX	-	0.12	0.42	0.7	0.02	0
EDU	0.12	-	0.02	0	0.08	0.06
SOUTH	0.42	0.02	-	0.24	0.1	0.64
SMSA	0.7	0	0.24	-	0.28	0
WAGE	0.02	0.08	0.1	0.28	-	0
RACE	0	0.06	0.64	0	0	-

(d) Bi-directed edge frequency ($c = c_1$).

	PROX	EDU	SOUTH	SMSA	WAGE	RACE
PROX	-	0.56	0	0	0	0
EDU	0	-	0.04	0.04	1	0
SOUTH	0.24	0	-	0.2	0.02	0
SMSA	0.68	0.02	0.34	-	0.88	0
WAGE	0	0.02	0	0	-	0
RACE	0	0	0.42	0	0	-

(e) Directed edge frequency ($c = c_{1.05}$).

	PROX	EDU	SOUTH	SMSA	WAGE	RACE
PROX	-	0.12	0.54	0.48	0	0
EDU	0.12	-	0.04	0.02	0	0
SOUTH	0.54	0.04	-	0.3	0.04	0.76
SMSA	0.48	0.02	0.3	-	0.12	0
WAGE	0	0	0.04	0.12	-	0
RACE	0	0	0.76	0	0	-

(f) Bi-directed edge frequency ($c = c_{1.05}$).

TABLE 9. Normalized edge frequencies for Card (1993) data.

EC.5. Proofs

PROPOSITION 1. *Let $\mathcal{G}^c$ be the graph returned by* CAUSALIP$\big(\mathcal{P}(E^c, |V|-1)\big)$ *for some $\boldsymbol{\omega} > 0$. Then $\mathcal{G}^c \in \underset{\mathcal{G}}{\operatorname{argmin}}\, L(\mathcal{G})$, i.e., $\mathcal{G}^c$ minimizes the loss function in (3).*

Proof of Proposition 1. Let $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ be an optimal solution to CAUSALIP$(\mathcal{P}(E^c, |V|-1))$ where $\boldsymbol{\omega} > 0$. The proof proceeds in two steps. First, we show $\tilde{z}^n_{ij} = 1$ if and only if $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C^n_{ij}) = 1$ for $i, j, \in V, n \in N^D_{ij}$. Second, we show $\tilde{z}^n_{ij} = 1$ if and only if $\mathbf{1}(i \not\perp_{\mathcal{G}^c} j | C^n_{ij}) = 1$ for $i, j, \in V, n \in N^I_{ij}$.

Step 1. We first show that $\tilde{z}^n_{ij} = 1$ implies that $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C^n_{ij}) = 1$ for $i, j, \in V, n \in N^D_{ij}$. Suppose $\tilde{z}^{n^*}_{i^*j^*} = 1$ for some fixed $i^*, j^*, \in V, n^* \in N^D_{ij}$. Note that the variable $\tilde{z}^{n^*}_{i^*j^*}$ only appears in constraint (6). Since CAUSALIP$(\mathcal{P}(E^c, |V|-1))$ minimizes $\sum_{i,j \in V} \sum_{n \in N_{ij}} \omega^n_{ij} z^n_{ij}$, in order for $\tilde{z}^{n^*}_{i^*j^*}$ to take value 1, left hand-side of constraint (6) must be equal to 0, i.e. $\sum_{p \in P_{i^*j^*}(\tilde{E}, \zeta)} \alpha^{n^*}_{i^*j^*p} y_p = 0$. Note $y_p = 1$ if and only if $p \in \mathcal{P}(E, \zeta)$ by constraints (4a) (4b). By Corollary EC.1, it follows that there does not exist a path in $\mathcal{G}^c$ with $\alpha^{n^*}_{i^*j^*p} = 1$. Following from the definition of $\alpha^{n^*}_{i^*j^*p}$, we must have $\mathbf{1}(i^* \perp_{\mathcal{G}^c} j^* | C^{n^*}_{i^*j^*}) = 1$. Next we show $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C^n_{ij}) = 1$ implies that $\tilde{z}^n_{ij} = 1$ for $i, j, \in V, n \in N^D_{ij}$. Suppose $\mathbf{1}(i^* \perp_{\mathcal{G}^c} j^* | C^{n^*}_{i^*j^*}) = 1$ for some fixed $i^*, j^*, \in V, n^* \in N^D_{ij}$. Then $\mathcal{G}^c$ does not include an unblocked path between i^* and j^* with respect to conditioning set $C^{n^*}_{i^*j^*}$. This implies $\sum_{p \in P_{i^*j^*}(\tilde{E}, \zeta)} \alpha^{n^*}_{i^*j^*p} y_p = 0$. By constraint (6), we mush have $\tilde{z}^{n^*}_{i^*j^*} = 1$. Hence $\tilde{z}^n_{ij} = 1$ if and only if $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C^n_{ij}) = 1$ for $i, j, \in V, n \in N^D_{ij}$.

Step 2. Now we show $\tilde{z}^n_{ij} = 1$ if and only if $\mathbf{1}(i \not\perp_{\mathcal{G}^c} j | C^n_{ij}) = 1$ for $i, j, \in V, n \in N^I_{ij}$. We first show that $\tilde{z}^n_{ij} = 1$ implies that $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C^n_{ij}) = 1$ for $i, j, \in V, n \in N^I_{ij}$. Suppose $\tilde{z}^{n^*}_{i^*j^*} = 1$ for some fixed $i^*, j^*, \in V, n^* \in N^I_{ij}$. Note that the variable $\tilde{z}^{n^*}_{i^*j^*}$ only appears in constraint (5). Since CAUSALIP$(\mathcal{P}(E^c, |V|-1))$ minimizes $\sum_{i,j \in V} \sum_{n \in N_{ij}} \omega^n_{ij} z^n_{ij}$, in order for $\tilde{z}^{n^*}_{i^*j^*}$ to take value 1, left hand-side of constraint (6) must be equal to 1 for at least one $n \in N^I_{ij}$, i.e. $\exists n^* \in N^I_{i^*j^*}$ such that $\alpha^n_{i^*j^*p} y_p \le z^{n^*}_{i^*j^*}$. This implies that there exists an unblocked path between i^* and j^* with respect to $C^{n^*}_{i^*j^*}$ in G^c. Hence we must have $\mathbf{1}(i^* \not\perp_{\mathcal{G}^c} j^* | C^{n^*}_{i^*j^*}) = 1$. Next we show $\mathbf{1}(i \not\perp_{\mathcal{G}^c} j | C^n_{ij}) = 1$ implies that $\tilde{z}^n_{ij} = 1$ for $i, j, \in V, n \in N^I_{ij}$. Suppose $\mathbf{1}(i^* \not\perp_{\mathcal{G}^c} j^* | C^{n^*}_{i^*j^*}) = 1$ for some fixed $i^*, j^*, \in V, n^* \in N^I_{ij}$. Then $\mathcal{G}^c$ include an unblocked path between i^* and j^* with respect to conditioning set $C^{n^*}_{i^*j^*}$. Note $y_p = 1$ if and only if $p \in \mathcal{P}(E, \zeta)$ by constraints (4a) (4b). By Corollary EC.1, it follows that there exists p^* such that $\alpha^{n^*}_{i^*j^*p^*} y^*_p = 1$. By constraint (5), we mush have $\tilde{z}^{n^*}_{i^*j^*} = 1$. Hence $\tilde{z}^n_{ij} = 1$ if and only if $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C^n_{ij}) = 1$ for $i, j, \in V, n \in N^I_{ij}$. □

LEMMA EC.3. *Suppose Assumption 2 holds. Let $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ be an optimal solution to* CAUSALIP$(\mathcal{P}(E^c, |V|-1))$ *where $\boldsymbol{\omega} > 0$. Then we must have $\sum_{i,j \in V} \sum_{n \in N_{ij}} \tilde{z}^n_{ij} = 0$.*

Proof of Lemma EC.3. Our approach is to construct a solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ such that the following three conditions are satisfied: (i) $\sum_{i,j \in V} \sum_{n \in N_{ij}} \tilde{z}^n_{ij} = 0$, (ii) $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ is a feasible solution

for CausalIP($\mathcal{P}(E^c, |V|-1)$), and (*iii*) all other solutions for CausalIP($\mathcal{P}(E^c, |V|-1)$) where $\sum_{i,j\in V}\sum_{n\in N_{ij}} \tilde{z}_{ij}^n \neq 0$ cannot be optimal. The proof proceeds in two steps. First, we construct $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$. Second, we show it satisfies conditions (*i*), (*ii*), and (*iii*) above.

Step 1. Let $\mathcal{G}_T = (V, E_T)$ be the true graph. First, construct $\tilde{\mathbf{x}}$ so that for all $e \in E^c$, $\tilde{x}_e = 1$ if and only if $e \in E_T$. It follows that $\mathcal{G}(\tilde{\mathbf{x}}) = \mathcal{G}_T$. Next, it is straightforward to show that for any fixed $\mathbf{x}$, there exists a solution to the inequalities (4) over $\mathbf{y}$ that satisfies $\mathbf{y} \in \{0,1\}^{|P(E^c)|}$. Let $\tilde{\mathbf{y}}$ be a solution to (4) under $\tilde{\mathbf{x}}$. Lastly, let $\tilde{z}_{ij}^n = 0$ for all $n \in N_{ij}$, $i, j \in V$.

Step 2. Note $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ trivially satisfies condition (*i*) above. Now let's show that $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ is feasible to CausalIP($\mathcal{P}(E^c, |V|-1)$). By construction, $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ satisfies the constraints (4) as well as the constraints that ensure $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ are all binary-valued. It remains to show that $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ satisfies constraints (5) and (6). To see that $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ satisfies (5), pick any $n \in N_{ij}^I$, $p \in P_{ij}(E^c, |V|-1)$ and $i, j \in V$. If $p \notin P_{ij}(E_T, |V|-1)$, then $y_p = 0$ by constraints (4a) and (4b). Hence such $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ satisfies (5). Now suppose $p \in P_{ij}(E_T, |V|-1)$. Because $n \in N_{ij}^I$ and Assumption 2 holds, we have $i \perp j | C_{ij}^n$ in graph $\mathcal{G}_T$. Because p is a path between i to j in $\mathcal{G}_T$, $i \perp j | C_{ij}^n$ implies that p must be blocked with respect to $C_{ij}^n \in I_{ij}$ (Definition 3). It follows that $\alpha_{ijp}^n = 0$, which implies $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ satisfies (5). Next, we show $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ satisfies (6). Pick any $n \in N_{ij}^D$ and $i, j \in V$. Note that by constraints (4a) and (4b), because $\mathcal{G}(\mathbf{x}) = \mathcal{G}_T$, for all $p \in \mathcal{P}(E^c, |V|-1)$, we have $\tilde{y}_p = 1$ if and only if $p \in \mathcal{P}(E_T, |V-1|)$. It follows that the left hand side of constraint (6) can be re-written as $\sum_{p\in P_{ij}(E^c,|V|-1)} \alpha_{ijp}^n \tilde{y}_p = \sum_{p\in P_{ij}(E_T,|V|-1)} \alpha_{ijp}^n$. Because $z_{ij}^n = 0$, it remains to show $\sum_{p\in P_{ij}(E_T,|V|-1)} \alpha_{ijp}^n \geq 1$. Because Assumption 2 holds, $n \in N_{ij}^D$ implies that $i \not\perp j | C_{ij}^n$ in $\mathcal{G}_T$. It follows from Definition 3 that there must exist at least one path between i and j that is unblocked with respect to C_{ij}^n. By definition of α_{ijp}^n, it follows that there exists at least one path p in G_T such that $\alpha_{ijp}^n = 1$. By Lemma EC.1, we have each path p in G_T must satisfy $p \in P_{ij}(E_T, |V|-1)$, which then implies $\sum_{p\in P_{ij}(E_T,|V|-1)} \alpha_{ijp}^n \geq 1$. With these we prove (*ii*) holds. (*iii*) Suppose there exists an optimal solution to CausalIP($\mathcal{P}(E^c, |V|-1)$) with $\sum_{i,j\in V}\sum_{n\in N_{ij}} \tilde{z}_{ij}^n \neq 0$. Since $\boldsymbol{\omega} > 0$, objective value corresponding to this solution must be positive. However, since we showed in Step 2 that there exists feasible solutions where $\sum_{i,j\in V}\sum_{n\in N_{ij}} \tilde{z}_{ij}^n = 0$ holds, and such solutions would make the objective value equal to 0. Hence a solution with $\sum_{i,j\in V}\sum_{n\in N_{ij}} \tilde{z}_{ij}^n \neq 0$ cannot be optimal. □

Proposition 2. *Let $\mathcal{G}^c$ be the graph returned by* CausalIP$\big(\mathcal{P}(E^c, |V|-1)\big)$ *given Assumption 2. Then $\mathcal{G}^c \sim \mathcal{G}_T$ (i.e., $\mathcal{G}^c$ and $\mathcal{G}_T$ are Markov equivalent) for any $\boldsymbol{\omega} > 0$.*

Proof of Proposition 2. Let $(\mathbf{x}^c, \mathbf{y}^c, \mathbf{z}^c)$ be an optimal solution to CausalIP($\mathcal{P}(E^c, |V|-1)$), where $\mathcal{G}^c := (V, E)$ where $E = \{e | x_e^c = 1\}$. Our approach will be to show that $\mathcal{G}^c$ satisfies the following two conditions for all pairs (i, j) such that $i \neq j$: (*a*) for each $C \in D_{ij}$, the nodes i and j

are d-connected with respect to C in $\mathcal{G}^c$, and (b) for each $C \in I_{ij}$, nodes i and j are d-separated with respect to C in $\mathcal{G}^c$. Note that if these two conditions hold, then it follows immediately from Assumption 2 that G^c and $\mathcal{G}_T$ are Markov equivalent. Pick any (i,j), and let it be fixed in the remainder of the proof. We first prove condition (a) holds. Note i and j are d-connected with respect to a conditioning set C if and only if there exists an unblocked path from i to j with respect to C (Definition 3). Suppose by way of contradiction that there exists a conditioning set $C_{ij}^{\bar{n}} \in D_{ij}$ such that all paths between i and j in $\mathcal{G}^c$ are blocked with respect to $C_{ij}^{\bar{n}}$. Then by definition, $\alpha_{ijp}^{\bar{n}} = 0$ for all $p \in P_{ij}(E, |V|-1)$. It follows that $\sum_{p \in P_{ij}(E,|V|-1)} \alpha_{ijp}^{\bar{n}} y_p^c = 0$. Next, note $\bar{n} \in N_{ij}^D$ because $C_{ij}^{\bar{n}} \in D_{ij}$. Because $\sum_{p \in P_{ij}(E,|V|-1)} \alpha_{ijp}^{\bar{n}} y_p^c = 0$, it follows from constraint (6) that $z_{ij}^{\bar{n}c} = 1$, and thus $\sum_{i,j \in V} \sum_{n \in N_{ij}} z_{ij}^{nc} > 0$. However, by Lemma EC.3, there exists a solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ to CausalIP(E^c) such that $\sum_{i,j \in V} \sum_{n \in N_{ij}} \tilde{z}_{ij}^n = 0$, which yields a contradiction. We conclude that condition (a) holds. We now prove condition (b) holds. Pick any conditioning set $C_{ij}^{\bar{n}} \in I_{ij}$. It suffices to show that all paths between i and j are blocked with respect to $C_{ij}^{\bar{n}}$ in $\mathcal{G}^c$. By definition, $\alpha_{ijp}^{\bar{n}} = 1$ for path $p \in P_{ij}(E, |V|-1)$ that is unblocked with respect to $C_{ij}^{\bar{n}}$. Therefore, it remains to show $\alpha_{ijp}^{\bar{n}} y_p^c = 0$ for all $p \in P_{ij}(E, |V|-1)$. Note $\bar{n} \in N_{ij}^I$ because $C_{ij}^{\bar{n}} \in I_{ij}$. Because $\bar{n} \in N_{ij}^I$, it follows from constraint (5) that $\alpha_{ijp}^{\bar{n}} y_p^c \leq z_{ij}^{\bar{n}c}$ for all $p \in P_{ij}(E, |V|-1)$. By Lemma EC.3, there exists a solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ to CausalIP(E^c) such that $\sum_{i,j \in V} \sum_{n \in N_{ij}} \tilde{z}_{ij}^n = 0$. Lemma EC.3 and constraint (5) imply $\alpha_{ijp}^{\bar{n}} y_p^c = 0$ for all $p \in P_{ij}(E, |V|-1)$. It follows that every path between i and j in $\mathcal{G}^c$ is blocked with respect to $C_{ij}^{\bar{n}}$, as desired. Because conditions (a) and (b) both hold, we conclude $\mathcal{G}^c \sim \mathcal{G}_T$. □

Lemma EC.4. *Suppose $\mathcal{G} \sim \mathcal{G}_T$ and Assumption 2 holds. For each $i, j \in V$, if $\mathcal{G}$ contains an edge between (i,j), then $I_{ij} = \emptyset$.*

Proof of Lemma EC.4. We show that if $I_{ij} \neq \emptyset$, then $\mathcal{G}$ cannot contain an edge between (i,j). Suppose by way of contradiction that it does. This edge is also a path – call it path p. Next, pick any conditioning set $C \in I_{ij}$. Because p does not contain any colliders or non-colliders, the path p is unblocked with respect to C (Definition 2). Then by Definition 3, the path p is d-connected with respect to C. Because $\mathcal{G} \in \mathcal{M}$ and the path p is d-connected with respect to C, it follows that $i \not\perp\!\!\!\perp j | C$ (Remark 1). However, because $C \in I_{ij}$, by definition of I_{ij} we have $i \perp\!\!\!\perp j | C$ – a contradiction. The result follows. □

Lemma 1. *Suppose Assumption 2 holds and consider a triple of nodes (i,j,k) in a graph $\mathcal{G} \sim \mathcal{G}_T$. Then*

(i) If there exists a collider chain over (i,j,k) in $\mathcal{G}$, then $I_{ij} = I_{jk} = \emptyset$ and $j \notin C$ for all $C \in I_{ik}$.

(ii) If there exists a non-collider chain over (i,j,k) in $\mathcal{G}$, then $I_{ij} = I_{jk} = \emptyset$ and $j \in C$ for all $C \in I_{ik}$.

Proof of Lemma 1. We prove the statements in order. ***(i)***. If (i,j,k) forms a collider chain, then the pairs (i,j) and (j,k) both contain an edge. It follows from Lemma EC.4 that $I_{ij} = I_{jk} = \emptyset$. We now show that if (i,j,k) form a collider chain, then $j \notin C$ for all $C \in I_{ik}$. Suppose by way of contradiction that (i,j,k) form a collider chain and there exists $C \in I_{ik}$ such that $j \in C$. Because node j is the only collider on the chain, and $j \in C$, the path from i to k is unblocked with respect to C (Definition 2), which implies i and k are d-connected with respect to C (Definition 3). Because the chain is d-connected with respect to C, and $\mathcal{G} \sim \mathcal{G}_T$, by Assumption 2 we have $C \in D_{ik}$. However, this is a contradiction because $C \in I_{ik}$. ***(ii)***. The result follows by parallel argument to (i), where (i,j,k) form a non-collider chain instead of a collider chain, and $j \notin C$ is used in place of $j \in C$. □

PROPOSITION 3. *The integer programming problem* NEWEDGESIP *is NP-hard.*

Proof of Proposition 3. The proof proceeds in two steps. First, we show a reduction from the vertex cover problem to NEWEDGESIP. Then we verify that the instance of NEWEDGESIP constructed by the reduction can arise within the EDGEGEN algorithm by appropriately constructing corresponding ground-truth graphs.

Step 1. We prove NEWEDGESIP is NP-hard through a reduction from the vertex cover problem. The vertex cover problem is an optimization problem defined on an undirected graph $H = (W, F)$ with vertices W and undirected edges F. The objective is to find a vertex cover $W' \subseteq W$ that has the minimum possible size. A vertex cover is a subset of vertices such that every edge in F has at least one of its endpoints in W'. The vertex cover problem is NP-hard (Garey and Johnson 1979).

The integer programming formulation for this optimization problem is as follows:

$$\underset{\mathbf{a}}{\text{minimize}} \quad \sum_{w \in W} a_w \tag{EC.2a}$$

$$\text{subject to} \quad a_w + a_{w'} \geq 1, \quad (w—w') \in F, \tag{EC.2b}$$

$$a_w \in \{0,1\}, \quad w \in W. \tag{EC.2c}$$

In this formulation, a_w is a binary variable associated with each vertex $w \in W$, which takes on the value 1 if w is included in the vertex cover (i.e., $w \in W'$), and 0 otherwise.

We can construct an instance of NEWEDGESIP for this vertex cover problem using Algorithm 5. Note that this is a polynomial time reduction as we just iterate over all pairs of vertices in W.

Algorithm 5: Reduction from Vertex Cover.

Input: $H = (W, F)$.
Output: V, $S(\mathbf{z})$, $\bar{S}(\mathbf{z})$, $\boldsymbol{\lambda}$.
Initialize: $V = \{1, 2, \ldots, |W| + 1\}, S(\mathbf{z}) = \emptyset, \bar{S}(\mathbf{z}) = \emptyset, \lambda^3_{ij} = 1, \lambda^t_{ij} = 0$ for all $i, j \in V, t \in \{1, 2\}$.
for $w, w' \in W$:
  if $(w$—$w') \in F$:
    Update $S(\mathbf{z}) = S(\mathbf{z}) \cup \{(w, |V|, w')\}$.

We now prove the reduction is correct. Suppose $\boldsymbol{a}'$ is a solution to the vertex cover problem over $H = (W, F)$ with the objective value m. Define W' as the corresponding vertex cover where $W' = \{w | a'_w = 1,\ w \in W\}$. Now let us construct a solution $\boldsymbol{\tau}'$ to the version of NewEdgesIP constructed by Algorithm 5. We start by initializing $\tau'^t_{ij} = 0$ for all $i, j \in V, t \in \{1, 2, 3\}$. Next, set $\tau'^2_{|V|,w} = 1$ and $\tau'^1_{w,|V|} = 1$ for $w \in W'$.

Next we show $\boldsymbol{\tau}'$ is a solution to NewEdgesIP with the objective value m. Since $\bar{S}(\mathbf{z}) = \emptyset$, constraint set (17) is satisfied trivially. Similarly, constraint set (18) is satisfied by the construction in the reduction algorithm and by how we constructed τ'. Constraint (16a) is satisfied because $\lambda^3_{ij} = 1$ for all $(i, j, k) \in S(z)$ by construction. Similarly, constraint (16b) is satisfied because $\lambda^3_{jk} = 1$ for all $(i, j, k) \in S(\mathbf{z})$. Next, we consider constraint (16c). Without loss of generality, consider a specific $(i^*, j^*, k^*) \in S(\mathbf{z})$. Note that we have $j^* = |V|$ and $(i^*$—$k^*) \in F$ for all $(i^*, j^*, k^*) \in S(\mathbf{z})$ by the reduction algorithm. This implies we must have $a'_{i^*} + a'_{k^*} \geq 1$ by constraint (EC.2b). Suppose $a'_{i^*} = 1$. This implies $i^* \in W'$. Then, by construction $\tau'^1_{i^*,|V|} = 1$, and hence constraint (17c) is satisfied for $(i^*, j^*, k^*) \in S(\mathbf{z})$. Now suppose $a'_{k^*} = 1$. This implies $k^* \in W'$. Then, by construction $\tau'^1_{k^*,|V|} = 1$, and hence constraint (17c) is satisfied for $(i^*, j^*, k^*) \in S(\mathbf{z})$. Lastly, since we have $\tau'^1_{w,|V|} = 1$ and $\tau'^2_{|V|,w} = 1$ for $w \in W'$ only, and only one of $\tau'^1_{w,|V|}$ and $\tau'^2_{|V|,w}$ is included in $\sum_{\substack{i,j \in V: \\ j<i}} \sum_{t \in \{1,2,3\}} \tau^t_{ij} = m$. Therefore, $\boldsymbol{\tau}'$ is a solution to NewEdgesIP.

Conversely, suppose that $\boldsymbol{\tau}^*$ is a solution with $\sum_{\substack{i,j \in V: \\ j<i}} \sum_{t \in \{1,2,3\}} \tau^t_{ij} = m$ to the version of NewEdgesIP constructed by Algorithm 5. We construct a solution $\boldsymbol{a}^*$ for the vertex cover problem over (H, F). Let $a^*_w = 1$ if $\tau^{*1}_{w,|V|} = 1$ and $a^*_w = 0$ otherwise. In other words, let $W' = \{w : a^*_w = 1,\ w \in W\}$ be a vertex cover of H. We want to show $\boldsymbol{a}^*$ is feasible for the vertex cover problem over $H = (W, F)$. Without loss of generality, let's select an edge $(w$—$w')$ in F. Then we must have $(w, |V|, w') \in S(z)$ by the reduction algorithm. By the reduction algorithm, $\lambda^3_{ij} = 1$ for all $i, j \in V$. Hence by constraint (18a), $\tau^{*3}_{ij} = 0$ for all $i, j \in V$. Hence in order to satisfy constraint (16c), we must have $\tau^{*1}_{w,|V|} + \tau^{*2}_{|V|,w} \geq 1$ for $(w, |V|, w') \in S(z)$. Suppose $\tau^{*1}_{w,|V|} = 1$. This implies $a^*_w = 1$ by construction, i.e. $w \in W'$. Hence the edge $(w$—$w')$ is covered by the end point $w \in W'$. Similarly, suppose $\tau^{*2}_{|V|,w'} = 1$. Note that by constraint (18b), we must have $\tau^{*1}_{w',|V|} = 1$ whenever $\tau^{*2}_{|V|,w'} = 1$. This implies $a^*_{w'} = 1$ by construction, i.e. $w' \in W'$. Hence the edge $(w$—$w')$ is covered by the end point $w' \in W'$. Note

that we have $\sum_{\substack{i,j\in V:\\ j<i}} \sum_{t\in\{1,2,3\}} \tau_{ij}^{*t} = m$.Since $a_w^* = 1$ if $\tau_{|V|,w}^{*1} = 1$, and $a_w^* = 0$ otherwise, we have $\sum_{w\in W} a_w = m$. Therefore, we establish $\boldsymbol{a}^*$ is a solution with objective value m to the vertex cover problem of $H = (W, F)$. With this, we have shown NewEdgesIP is NP-hard.

Step 2. Now we show the specific instances of $S(\mathbf{z})$, $\bar{S}(\mathbf{z})$, and $\boldsymbol{\lambda}$ constructed using Algorithm 5 can arise in NewEdgesIP when used within the EdgeGen algorithm in the causally sufficient setting. We do so by constructing a true, causally sufficient, data-generating graph given an instance of the vertex cover problem, stepping through the EdgeGen algorithm, and showing that the constructed instance of NewEdgesIP is realized within the algorithm. Graph construction. Consider a vertex cover problem over $H = (W, F)$. Let $\mathcal{G}^* = (V, E^*)$ be the corresponding data-generating graph with vertices $V = \{1, \ldots, |W|+1\}$ and edges $E^* = \big\{(w \to |W|+1), (w' \to |W|+1)$ for each $(w, w') \in F\big\}$. Constructing oracle independence/dependence relations. Let us now consider the oracle setting (i.e., when Assumption 2 holds) and construct the dependence and independence relations implied by this graph. Since $i \to |W|+1 \leftarrow j$ is the only path between i and j, and node $|W|+1$ is a collider on it, we have $D_{ij} = \big\{C' \mid C' = C \cup \{|W|+1\},\ C \subset \{1, \ldots, |W|\} \setminus \{i, j\}\big\}$ and $I_{ij} = \big\{C \mid C \subset \{1, \ldots, |W|\} \setminus \{i, j\}\big\}$ for $i, j \in \{1, \ldots, |W|\}$. Also, we have $D_{i,|W|+1} = \big\{C \mid C \subset \{1, \ldots, |W|+1\} \setminus \{i, |W|+1\}\big\}$ and $I_{i,|W|+1} = \emptyset$ for $i \in \{1, \ldots, |W|\}$. EdgeGen algorithm. EdgeGen takes five inputs $V, S, \bar{S}, \boldsymbol{\omega}, \zeta$. Let S^* and $\bar{S}^*$ be the corresponding collider and non-collider chains implied by the established dependence structure over $\mathcal{G}^* = (V, E^*)$. By equations (10) and (11), we have $S^* = \{(i, |W|+1, j)\ \forall i, j \in \{1, \ldots, W\}\}$ and $\bar{S}^* = \emptyset$. Let $\boldsymbol{\omega}$ be any weight such that $\boldsymbol{\omega} > 0$ and initialize $\zeta = |W|$, which is the longest simple path length over $|W|+1$ nodes. Since we define the initial set of edges $\tilde{E}_0$ as $\tilde{E}_0 = \{i \to j, i \leftarrow j, i \leftrightarrow j \mid I_{ij} = \emptyset, D_{ik} = D_{jk} = \emptyset$ for all $k \in V \setminus \{i, j\}\}$, we have $\tilde{E}_0 = \emptyset$. Let us also initialize $\tilde{E} = \tilde{E}_0$, $\sigma = \emptyset$, and $s = 1$. Step 1 of EdgeGen. Note that $\tilde{E} = \emptyset$ implies $\mathcal{P}^-(\tilde{E}, \zeta) = \emptyset$. Hence when we solve CausalIP$\big(\mathcal{P}^-(\tilde{E}, \zeta)\big)$ in iteration s, we will get $\mathcal{G}_s = (V, E_s)$ where $E_s = \emptyset$ and $z_{ij}^n = 0$ for all $n \in N_{ij}^I, i, j \in V$ and $z_{ij}^n = 1$ for all $n \in N_{ij}^D, i, j \in V$. Step 2 of EdgeGen. Since $E_s = \emptyset$, we will have $\epsilon_{ij}^n = 1$ for all $n \in N_{ij}^D$ and $i, j \in V$. Step 3 of EdgeGen. The condition $\boldsymbol{\omega}^\top \boldsymbol{\epsilon}_s > 0$ is satisfied and UpdateEdges sub-algorithm is called in Step 3.1. UpdateEdges sub-algorithm. Since we consider the causally sufficient setting, complete edge set doesn't include bi-directed edges, i.e., $E^c = \{i \leftarrow j,\ i \to j,\ \forall i, j \in V : i \neq j\}$. UpdateEdges takes $S(\mathbf{z}), \bar{S}(\mathbf{z}), S, \bar{S}, \tilde{E}, \tilde{E}_0, \zeta$ as inputs. The sets $S^*(\mathbf{z})$ and $\bar{S}^*(\mathbf{z})$ are then constructed following the steps in Section 4.1. We have $E_{i,|W|+1,j} = \big\{i \to |W|+1,\ |W|+1 \leftarrow j\big\}$ for all $(i, |W|+1, j) \in S^*$ by equation (12a). Because $\epsilon_{ij}^n = 1$ for all $n \in N_{ij}^D$, we have $N_{ij}^D(\mathbf{z}) = N_{ij}^D$ for all $i, j \in V$. This implies $\Psi(\mathbf{z}) = S^*$ by equation (14a) and $S^*(\mathbf{z}) = S^*$ by equation (15a). Lastly, note that we have $\bar{S}^*(\mathbf{z}) = \emptyset$ by equation (15b) because $\bar{S}^* = \emptyset$. In summary, the sets $S^*(\mathbf{z})$ and $\bar{S}^*(\mathbf{z})$ are equal to those constructed through the steps of Algorithm 5 over $H = (W, F)$. Step 1 of UpdateEdges. Since $S^*(\mathbf{z}) \cup \bar{S}^*(\mathbf{z}) \neq \emptyset$, we solve NewEdgesIP$(S^*(\mathbf{z}), \bar{S}^*(\mathbf{z}), \boldsymbol{\lambda})$. We

construct $\boldsymbol{\lambda}^*$ in the causally sufficient setting as follows: $\lambda_{ij}^{*t} = 1$ if $\tilde{E}$ contains a type t edge between nodes i and j for $t \in \{1,2\}$, and $\lambda_{ij}^t = 0$ otherwise, and $\lambda_{ij}^{*3} = 1$ for all i and j pairs. This way we ensure bi-directed edges will never be selected in NewEdgesIP because of the constraint (18a). Since we already showed we have $S^*(\mathbf{z}) = S(\mathbf{z})$ and $\bar{S}^*(\mathbf{z}) = \bar{S}(\mathbf{z})$, it remains to show that $\boldsymbol{\lambda}^*$ is equivalent to the $\boldsymbol{\lambda}$ constructed in Algorithm 5. Since $\tilde{E} = \emptyset$, we have $\lambda_{ij}^{*1} = 0$ and $\lambda_{ij}^{*2} = 0$ for all $i, j \in V$ and we have $\lambda_{ij}^{*3} = 1$ for all $i, j \in V$ as a result of considering the causally sufficient setting. Hence $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$. □

Lemma EC.5. NewEdgesIP *is always feasible when called by* UpdateEdges.

Proof of Lemma EC.5. The formulation NewEdgesIP is called in two cases: $S(\mathbf{z}) \cup \bar{S}(\mathbf{z}) \neq \emptyset$ and $S(\mathbf{z}) \cup \bar{S}(\mathbf{z}) = \emptyset$. The proofs for the second case is identical to the first, where $R(\mathbf{z})$ and $\bar{R}(\mathbf{z})$ is used in place of $S(\mathbf{z})$ and $\bar{S}(\mathbf{z})$. We therefore focus on the case where $S(\mathbf{z}) \cup \bar{S}(\mathbf{z}) \neq \emptyset$. To prove the result, it suffices to construct a solution $\tilde{\mathbf{w}}$ that satisfies each constraint in NewEdgesIP. Accordingly, let $\tilde{\tau}_{ij}^t = 1 - \lambda_{ij}^t$ for all $i, j \in V$ and $t \in \{1,2,3\}$. With a slight abuse of notation, let e_{ij}^t be the edge corresponding to the variable τ_{ij}^t. The proof proceeds in three steps. First, we show $\tilde{\mathbf{w}}$ satisfies constraints (16a)–(16c); second, that it satisfies (17a)–(17d); and third, (18a)–(18c).

Step 1. We first show that $\tilde{\mathbf{w}}$ satisfies constraints (16a)–(16c). If $S(\mathbf{z}) = \emptyset$, the result holds trivially. Suppose $S(\mathbf{z}) \neq \emptyset$. Note $\tilde{\tau}_{ij}^t + \lambda_{ij}^t = 1$ for all $t \in \{1,2,3\}$ by construction. It immediately follows that (16a) and (16b) are satisfied for all $(i,j,k) \in S(\mathbf{z})$. Next we show $\tilde{\mathbf{w}}$ satisfies (16c). Pick any $(i,j,k) \in S(\mathbf{z})$. Note $(i,j,k) \in S(\mathbf{z})$ implies there exists $e \in E_{ijk}$ such that $e \notin \tilde{E}$. By definition of E_{ijk}, we must have either $e = e_{ij}^t$ for some $t \in \{1,3\}$ or $e = e_{jk}^t$ for some $t \in \{2,3\}$. Suppose $e = e_{ij}^t$ for some $t \in \{1,3\}$. Because $e \notin \tilde{E}$, at least one of $\lambda_{ij}^1 = 0$ or $\lambda_{ij}^3 = 0$ must hold. By construction of $\tilde{\mathbf{w}}$, it follows that at least one of $\tilde{\tau}_{ij}^1 = 1$ or $\tilde{\tau}_{ij}^3 = 1$ must hold. Therefore, $\sum_{t\in\{1,3\}} \tau_{ij}^t \geq 1$, which implies constraint (16c) is satisfied by $\tilde{\mathbf{w}}$. The case where $e = e_{jk}^t$ for some $t \in \{2,3\}$ follows by parallel argument.

Step 2. We now show $\tilde{\mathbf{w}}$ satisfies constraints (17a)–(17d). If $\bar{S}(\mathbf{z}) = \emptyset$, the result holds trivially. Suppose $\bar{S}(\mathbf{z}) \neq \emptyset$. It is straightforward to verify that constraints (17a)– (17c) are immediately satisfied because $\tau_{ij}^t + \lambda_{ij}^t = 1$ for all $i, j \in V$ and $t \in \{1,2,3\}$. We now show $\tilde{\mathbf{w}}$ satisfies constraint (17d). Pick any $(i,j,k) \in \bar{S}(\mathbf{z})$. By definition of $\bar{S}(\mathbf{z})$, $(i,j,k) \in \bar{S}(\mathbf{z})$ implies there exists $e \in e_{ijk}$ such that $e \notin \tilde{E}$. By definition of $\bar{E}_{ijk}$, we must have either $e = e_{ij}^t$ or $e = e_{jk}^t$ for some $t \in \{1,2,3\}$. Suppose $e = e_{ij}^t$ for some $t \in \{1,2,3\}$. Because $e \notin \tilde{E}$, at least one of $\lambda_{ij}^1 = 0$, $\lambda_{ij}^2 = 0$, or $\lambda_{ij}^3 = 0$ must hold. It follows that at least one of $\tilde{\tau}_{ij}^1 = 1$, $\tilde{\tau}_{ij}^2 = 1$, or $\tilde{\tau}_{ij}^3 = 1$ must hold. Therefore, $\sum_{t\in\{1,2,3\}} \tau_{ij}^t \geq 1$, which implies constraint (17d) is satisfied by $\tilde{\mathbf{w}}$. The case where $e = e_{jk}^t$ for some $t \in \{1,2,3\}$ follows by parallel argument.

Step 3. Constraints (18a)–(18c) hold because $\tilde{\tau}_{ij}^t + \lambda_{ij}^t = 1$ by construction of $\tilde{\mathbf{w}}$, and $\lambda_{ij}^1 = \lambda_{ji}^2$ and $\lambda_{ij}^3 = \lambda_{ji}^3$ by definition of λ_{ij}^t, for all $i, j \in V$ and $t \in \{1, 2, 3\}$. □

LEMMA EC.6. CAUSALIP$\big(\mathcal{P}^-(\tilde{E}, \zeta)\big)$ *is always feasible when called by* EDGEGEN.

Proof of Lemma EC.6. Our approach is to construct a feasible solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ for CAUSALIP$\big(\mathcal{P}^-(\tilde{E}, \zeta)\big)$ when it is called by EDGEGEN. First, let's start with constructing a solution where $x_e = 0$ for all $e \in \tilde{E}$. Next, it is straightforward to show that for any fixed $\mathbf{x}$, there exists a solution to the inequalities (4). When $\mathbf{x} = \mathbf{0}$, this solution is $\mathbf{y} = 0$ by (4). Lastly, let $\tilde{z}_{ij}^n = 1$ for all $n \in N_{ij}^D$, $i, j \in V$ and $\tilde{z}_{ij}^n = 0$ for all $n \in N_{ij}^I$, $i, j \in V$. It remains to show that $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ satisfies constraints (5) and (6). To see that $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ satisfies (5), pick any $n \in N_{ij}^I$, $p \in P_{ij}(\tilde{E}, |V| - 1)$ and $i, j \in V$. Since $y_p = 0$, such $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ satisfies (5). Next, we show $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ satisfies (6). Pick any $n \in N_{ij}^D$ and $i, j \in V$. Note that the left hand side of constraint (6) is equal to 0 as $y_p = 0$ for all $p \in \mathcal{P}(\tilde{E}, |V| - 1)$. Then, constraint (6) can be written as $z_{ij}^n \geq 1$ for all $i, j \in V, n \in N_{ij}^D$. As we set $z_{ij}^n = 1$ for all $i, j \in V, n \in N_{ij}^D$, constraint (6) is satisfied. □

LEMMA EC.7. *Let $\mathcal{G}_s = (V, E_s)$ be an input graph to* $\mathcal{G}$-POSTPROCESS *at iteration t and let $\boldsymbol{\epsilon}$ be the output constructed by* $\mathcal{G}$-POSTPROCESS. *Then (i) for each $i, j \in V$, $n \in N_{ij}^I$, $\epsilon_{ij}^n = 0$ if and only if $i \perp_{\mathcal{G}_s} j | C_{ij}^n$, and (ii) for each $i, j \in V$, $n \in N_{ij}^D$, $\epsilon_{ij}^n = 0$ if and only if $i \not\perp_{\mathcal{G}_s} j | C_{ij}^n$.*

Proof of Lemma EC.7. By definition of $\boldsymbol{\epsilon}$ in $\mathcal{G}$-POSTPROCESS, for each $i, j \in V$ and $n \in N_{ij}^I$, $\epsilon_{ij}^n = 0$ if and only if $\sum_{p \in \mathcal{P}_{ij}(E_s, |V|-1)} \alpha_{ijp}^n = 0$. By definition, $\alpha_{ijp}^n = 0$ if and only if the path p between i and j is blocked with respect to $C_{ij}^n \in A_{ij}$. Note $\mathcal{P}_{ij}(E_s, |V| - 1)$ is the complete set of paths (i.e., including appendages) up to length $|V| - 1$ in graph $\mathcal{G}_s$. Therefore, for each $i, j \in V$, $n \in N_{ij}^I$, $\epsilon_{ij}^n = 0$ if and only if all paths between i and j are blocked with respect to $C_{ij}^n \in A_{ij}$ in the graph $\mathcal{G}_s$. It follows from Definition 3 that for $i, j \in V$, $n \in N_{ij}^I$, $\epsilon_{ij}^n = 0$ if and only if i and j are d-separated with respect to $C_{ij}^n \in A_{ij}$ in $\mathcal{G}_s$, or equivalently, $i \perp_{\mathcal{G}_s} j | C_{ij}^n$. By a similar argument, for each $i, j \in V$, $n \in N_{ij}^D$, $\epsilon_{ij}^n = 0$ if and only if $\sum_{p \in \mathcal{P}_{ij}(E_s, |V|-1)} \alpha_{ijp}^n > 0$, which implies there exists at least one unblocked path between i and j. It follows that for $i, j \in V$, $n \in N_{ij}^D$, $\epsilon_{ij}^n = 0$ if and only if i and j are d-connected with respect to $C_{ij}^n \in A_{ij}$ in $\mathcal{G}_s$, or equivalently, $i \not\perp_{\mathcal{G}_s} j | C_{ij}^n$. □

PROPOSITION 4. EDGEGEN *is guaranteed to terminate. Further, if Assumption 2 holds, the objective in (3) is equal to zero at termination.*

Proof of Proposition 4. We first prove that the algorithm terminates. We start by noting that EDGEGEN terminates if (*i*) $\boldsymbol{\omega}^\top \boldsymbol{\epsilon} = 0$ or (*ii*) $\tilde{E} = E^c$ and $\zeta = |V| - 1$ (by Step 3.3 of EDGEGEN).

Suppose condition (i) never holds. We show that condition (ii) eventually holds. By Lemma EC.5 and Lemma EC.6, EdgeGen returns a graph $\mathcal{G}_s = (V, E_s)$ at each iteration t. At each iteration of Algorithm 1, either $\tilde{E}$ or ζ strictly increases. Because both E^c and $|V|-1$ are finite, we arrive at the termination condition $\tilde{E} = E^c$ and $\zeta = |V|-1$ in a finite number of iterations. Therefore, if condition (i) never holds, condition (ii) eventually holds and EdgeGen terminates. If condition (i) holds, the algorithm terminates by construction. Next we prove $L(\mathcal{G}_s) = 0$ at termination if Assumption 2 holds. First, suppose condition (i) holds, i.e., $\boldsymbol{\omega}^\top \boldsymbol{\epsilon} = 0$, at termination. Then $L(\mathcal{G}_s)$ must be equal to zero by Lemma EC.7. Now suppose condition (ii) holds, i.e., $\tilde{E} = E^c$ and $\zeta = |V|-1$, at termination. This implies we have $\sum_{i,j\in V}\sum_{n\in N_{ij}} \tilde{z}_{ij}^n = 0$ by Lemma EC.3. Then $L(\mathcal{G}_s) = 0$ as a result of the correspondence between the loss function in (3) and the objective of CausalIP$(\mathcal{P}(E^c, |V|-1))$ we established in Proposition 1. □

Theorem 1. *Let $\mathcal{G}^*$ be the graph returned by* EdgeGen *for $\boldsymbol{\omega} > 0$.*

(i) $\mathcal{G}^c \in \underset{\mathcal{G}}{\operatorname{argmin}}\, L(\mathcal{G})$, i.e., $\mathcal{G}^c$ minimizes the objective in (3).

(ii) Further, if Assumption 2 holds, $\mathcal{G}^* \sim \mathcal{G}_T$ (i.e., $\mathcal{G}^*$ and $\mathcal{G}_T$ are Markov equivalent).

Proof of Theorem 1. **(*i*)**. By Proposition 4, EdgeGen is guaranteed to terminate. Note that EdgeGen terminates only if (1) $\boldsymbol{\omega}^\top \boldsymbol{\epsilon} = 0$ or (2) $\tilde{E} = E^c$ and $\zeta = |V|-1$ (by Step 3.3 of EdgeGen). Note that by Lemma EC.7, $\boldsymbol{\omega}^\top \boldsymbol{\epsilon}$ is equivalent to the loss function $L(\mathcal{G})$ given in (3):

$$\boldsymbol{\omega}^\top \boldsymbol{\epsilon} = \sum_{i,j\in V}\sum_{n\in N_{ij}^D} \omega_{ij}^n \cdot \mathbf{1}(i \perp_{\mathcal{G}} j | C_{ij}^n) + \sum_{i,j\in V}\sum_{n\in N_{ij}^I} \omega_{ij}^n \cdot \mathbf{1}(i \not\perp_{\mathcal{G}} j | C_{ij}^n).$$

Note that the loss function is lower bounded by 0. Suppose condition (1) holds at iteration s^* and EdgeGen terminates. Because $\boldsymbol{\omega} > \mathbf{0}$, this implies $\sigma_s > 0$ for all $s < s^*$. Hence, by Step 4 of EdgeGen, the graph returned is $\mathcal{G}_{s^*}$. Because $\boldsymbol{\omega}^\top \boldsymbol{\epsilon} = 0$ holds at iteration s^*, $\mathcal{G}_{s^*}$ minimizes the loss function $L(\mathcal{G})$. Suppose condition (1) never holds. Then condition (2) must eventually hold at the termination iteration s^*. This implies that EdgeGen solves CausalIP$\big(\mathcal{P}(E^c, |V|-1)\big)$ at iteration s^{**} and returns corresponding graph $\mathcal{G}_{s^*}$. By Proposition 1, $\mathcal{G}_{s^*}$ minimizes the loss function $L(\mathcal{G})$.

(*ii*). Similar to part (i), at the termination iteration s^* of EdgeGen, at least one of the following two conditions must hold: (1) $\boldsymbol{\omega}^\top \boldsymbol{\epsilon} = 0$ or (2) $\tilde{E} = E^c$ and $\zeta = |V|-1$. First, suppose condition (1) holds at s^* for some $\boldsymbol{\omega} > 0$. Then by the proof of part (i), EdgeGen returns graph $\mathcal{G}_{s^*}$. Because condition (1) holds and $\boldsymbol{\omega} > 0$, we have $\epsilon_{ij}^n = 0$ for all $i, j \in V$ and $n \in N_{ij}^I \cup N_{ij}^D$. It follows from Assumption 2 and Lemma EC.7 that $\mathcal{G}_{s^*}$ must be Markov equivalent to the true graph $\mathcal{G}_T$. Now suppose condition (2) holds at the termination iteration s^*. Then EdgeGen solves CausalIP$\big(\mathcal{P}(E^c, |V|-1)\big)$ at iteration s^*, and let $\mathcal{G}_{s^*}$ be the returned graph. Because Assumption 2 holds, it follows from Proposition 2 that $\mathcal{G}_{s^*}$ is Markov equivalent to $\mathcal{G}_T$. □