

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 $p_{ij}^- = (e_1, \dots, e_\ell)$ 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 l 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: $i \cdots * \rightarrow l \leftarrow * \cdots * \rightarrow l \leftarrow * \dots j$, where $*$ represents that an edge can have an arrow end or a tail end. This implies that path p^* must have the following form $i \cdots * \rightarrow l \leftarrow * \dots j$, 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: $i \cdots * \rightarrow l \rightarrow \dots * \leftarrow l \leftarrow * \dots j$. 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: $i \cdots * \rightarrow l \leftarrow * \dots j$. 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 . \square

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

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_i^k 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_i^k 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:

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

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

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

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

Constraint (EC.1a) ensures R_i^k can only be the parent set of i if the edge $j \rightarrow i$ is present for each $j \in R_i^k$. Constraint (EC.1b) ensures that if $j \rightarrow i$ is present for all $j \in R_i^k$ and if $j \rightarrow i$ 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 $\{1, \dots, \text{maxDegree}\}$ 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.

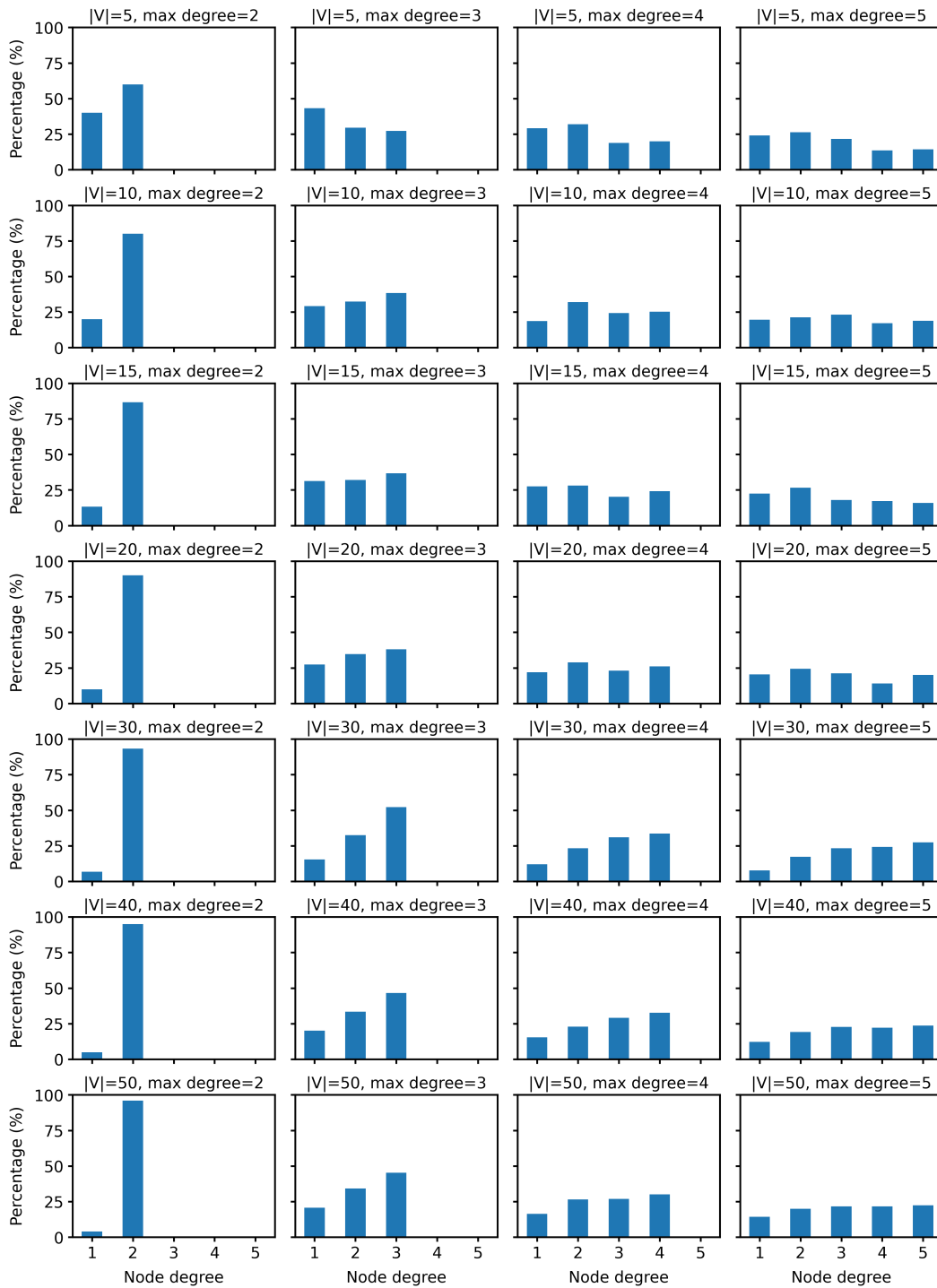


Figure EC.1 Empirical distributions of node degree for randomly generated graphs in §4.4.

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).

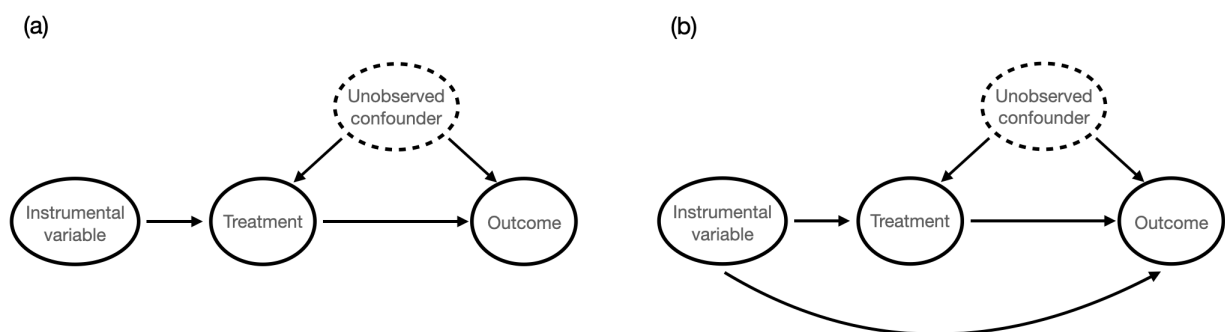


Figure EC.2 Two Markov equivalent graphs. In (a) the instrument is valid, in (b) it is invalid.

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($\mathcal{P}(E^c, |V| - 1)$) for some $\omega > 0$. 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 $\omega > 0$. The proof proceeds in two steps. First, we show $\tilde{z}_{ij}^n = 1$ if and only if $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C_{ij}^n) = 1$ for $i, j, \in V, n \in N_{ij}^D$. Second, we show $\tilde{z}_{ij}^n = 1$ if and only if $\mathbf{1}(i \not\perp_{\mathcal{G}^c} j | C_{ij}^n) = 1$ for $i, j, \in V, n \in N_{ij}^I$.

Step 1. We first show that $\tilde{z}_{ij}^n = 1$ implies that $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C_{ij}^n) = 1$ for $i, j, \in V, n \in N_{ij}^D$. Suppose $\tilde{z}_{i^*j^*}^{n^*} = 1$ for some fixed $i^*, j^*, \in V, n^* \in N_{ij}^D$. Note that the variable $\tilde{z}_{i^*j^*}^{n^*}$ only appears in constraint (6). Since CAUSALIP($\mathcal{P}(E^c, |V| - 1)$) minimizes $\sum_{i,j \in V} \sum_{n \in N_{ij}} \omega_{ij}^n z_{ij}^n$, in order for $\tilde{z}_{i^*j^*}^{n^*}$ to take value 1, left hand-side of constraint (6) must be equal to 0, i.e. $\sum_{p \in P_{i^*j^*}(\bar{E}, \zeta)} \alpha_{i^*j^*p}^{n^*} y_p = 0$. 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 $\alpha_{i^*j^*p}^{n^*} = 1$. Following from the definition of $\alpha_{i^*j^*p}^{n^*}$, we must have $\mathbf{1}(i^* \perp_{\mathcal{G}^c} j^* | C_{i^*j^*}^{n^*}) = 1$. Next we show $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C_{ij}^n) = 1$ implies that $\tilde{z}_{ij}^n = 1$ for $i, j, \in V, n \in N_{ij}^D$. Suppose $\mathbf{1}(i^* \perp_{\mathcal{G}^c} j^* | C_{i^*j^*}^{n^*}) = 1$ for some fixed $i^*, j^*, \in V, n^* \in N_{ij}^D$. Then \mathcal{G}^c does not include an unblocked path between i^* and j^* with respect to conditioning set $C_{i^*j^*}^{n^*}$. This implies $\sum_{p \in P_{i^*j^*}(\bar{E}, \zeta)} \alpha_{i^*j^*p}^{n^*} y_p = 0$. By constraint (6), we must have $\tilde{z}_{i^*j^*}^{n^*} = 1$. Hence $\tilde{z}_{ij}^n = 1$ if and only if $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C_{ij}^n) = 1$ for $i, j, \in V, n \in N_{ij}^D$.

Step 2. Now we show $\tilde{z}_{ij}^n = 1$ if and only if $\mathbf{1}(i \not\perp_{\mathcal{G}^c} j | C_{ij}^n) = 1$ for $i, j, \in V, n \in N_{ij}^I$. We first show that $\tilde{z}_{ij}^n = 1$ implies that $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C_{ij}^n) = 1$ for $i, j, \in V, n \in N_{ij}^I$. Suppose $\tilde{z}_{i^*j^*}^{n^*} = 1$ for some fixed $i^*, j^*, \in V, n^* \in N_{ij}^I$. Note that the variable $\tilde{z}_{i^*j^*}^{n^*}$ only appears in constraint (5). Since CAUSALIP($\mathcal{P}(E^c, |V| - 1)$) minimizes $\sum_{i,j \in V} \sum_{n \in N_{ij}} \omega_{ij}^n z_{ij}^n$, in order for $\tilde{z}_{i^*j^*}^{n^*}$ to take value 1, left hand-side of constraint (6) must be equal to 1 for at least one $n \in N_{ij}^I$, i.e. $\exists n^* \in N_{ij}^I$ such that $\alpha_{i^*j^*p}^{n^*} y_p \leq z_{i^*j^*}^{n^*}$. This implies that there exists an unblocked path between i^* and j^* with respect to $C_{i^*j^*}^{n^*}$ in \mathcal{G}^c . Hence we must have $\mathbf{1}(i^* \not\perp_{\mathcal{G}^c} j^* | C_{i^*j^*}^{n^*}) = 1$. Next we show $\mathbf{1}(i \not\perp_{\mathcal{G}^c} j | C_{ij}^n) = 1$ implies that $\tilde{z}_{ij}^n = 1$ for $i, j, \in V, n \in N_{ij}^I$. Suppose $\mathbf{1}(i^* \not\perp_{\mathcal{G}^c} j^* | C_{i^*j^*}^{n^*}) = 1$ for some fixed $i^*, j^*, \in V, n^* \in N_{ij}^I$. Then \mathcal{G}^c include an unblocked path between i^* and j^* with respect to conditioning set $C_{i^*j^*}^{n^*}$. 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 $\alpha_{i^*j^*p^*}^{n^*} y_{p^*} = 1$. By constraint (5), we must have $\tilde{z}_{i^*j^*}^{n^*} = 1$. Hence $\tilde{z}_{ij}^n = 1$ if and only if $\mathbf{1}(i \perp_{\mathcal{G}^c} j | C_{ij}^n) = 1$ for $i, j, \in V, n \in N_{ij}^I$. \square

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 $\omega > 0$. Then we must have $\sum_{i,j \in V} \sum_{n \in N_{ij}} \tilde{z}_{ij}^n = 0$.

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) $\sum_{i,j \in V} \sum_{n \in N_{ij}} \tilde{z}_{ij}^n = 0$, (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 $\mathcal{G}(\tilde{\mathbf{x}}) = \mathcal{G}_T$, 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 $\omega > 0$, 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. \square

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 $\omega > 0$.*

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 \mathcal{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 $\mathcal{G}^c \sim \mathcal{G}_T$. \square

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. \square

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 $j \in C$. \square

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:** $V, S(\mathbf{z}), \bar{S}(\mathbf{z}), \lambda$.**Initialize:** $V = \{1, 2, \dots, |W| + 1\}, S(\mathbf{z}) = \emptyset, \bar{S}(\mathbf{z}) = \emptyset, \lambda_{ij}^3 = 1, \lambda_{ij}^t = 0$ 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 \mathbf{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 $\tau_{ij}^t = 0$ for all $i, j \in V, t \in \{1, 2, 3\}$. Next, set $\tau_{|V|,w}^2 = 1$ and $\tau_{w,|V|}^1 = 1$ 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 $\boldsymbol{\tau}'$. Constraint (16a) is satisfied because $\lambda_{ij}^3 = 1$ for all $(i, j, k) \in S(\mathbf{z})$ by construction. Similarly, constraint (16b) is satisfied because $\lambda_{jk}^3 = 1$ 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 $\tau_{i^*,|V|}^1 = 1$, 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 $\tau_{k^*,|V|}^1 = 1$, and hence constraint (17c) is satisfied for $(i^*, j^*, k^*) \in S(\mathbf{z})$. Lastly, since we have $\tau_{w,|V|}^1 = 1$ and $\tau_{|V|,w}^2 = 1$ for $w \in W'$ only, and only one of $\tau_{w,|V|}^1$ and $\tau_{|V|,w}^2$ is included in $\sum_{\substack{i,j \in V: \\ j < i}} \sum_{t \in \{1,2,3\}} \tau_{ij}^t = m$. Therefore, $\boldsymbol{\tau}'$ is a solution to NEWEDGESIP.

Conversely, suppose that $\boldsymbol{\tau}^*$ is a solution with $\sum_{\substack{i,j \in V: \\ j < i}} \sum_{t \in \{1,2,3\}} \tau_{ij}^t = m$ to the version of NEWEDGESIP constructed by Algorithm 5. We construct a solution \mathbf{a}^* for the vertex cover problem over (H, F) . Let $a_w^* = 1$ if $\tau_{w,|V|}^1 = 1$ 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 \mathbf{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 $(w, |V|, w') \in S(\mathbf{z})$ by the reduction algorithm. By the reduction algorithm, $\lambda_{ij}^3 = 1$ for all $i, j \in V$. Hence by constraint (18a), $\tau_{ij}^{*3} = 0$ for all $i, j \in V$. Hence in order to satisfy constraint (16c), we must have $\tau_{w,|V|}^{*1} + \tau_{|V|,w}^{*2} \geq 1$ for $(w, |V|, w') \in S(\mathbf{z})$. Suppose $\tau_{w,|V|}^{*1} = 1$. 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 $\tau_{|V|,w'}^{*2} = 1$. Note that by constraint (18b), we must have $\tau_{w',|V|}^{*1} = 1$ whenever $\tau_{|V|,w'}^{*2} = 1$. 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 \mathbf{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 $V = \{1, \dots, |W| + 1\}$ and edges $E^* = \{(w \rightarrow |W| + 1), (w' \rightarrow |W| + 1) \text{ for each } (w, w') \in F\}$. 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 $i \rightarrow |W| + 1 \leftarrow j$ is the only path between i and j , and node $|W| + 1$ is a collider on it, we have $D_{ij} = \{C' \mid C' = C \cup \{|W| + 1\}, C \subset \{1, \dots, |W|\} \setminus \{i, j\}\}$ and $I_{ij} = \{C \mid C \subset \{1, \dots, |W|\} \setminus \{i, j\}\}$ for $i, j \in \{1, \dots, |W|\}$. Also, we have $D_{i, |W| + 1} = \{C \mid C \subset \{1, \dots, |W| + 1\} \setminus \{i, |W| + 1\}\}$ and $I_{i, |W| + 1} = \emptyset$ for $i \in \{1, \dots, |W|\}$. 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 $S^* = \{(i, |W| + 1, j) \mid \forall i, j \in \{1, \dots, W\}\}$ 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 $\tilde{E}_0 = \{i \rightarrow j, i \leftarrow j, i \leftrightarrow j \mid I_{ij} = \emptyset, D_{ik} = D_{jk} = \emptyset \text{ for all } k \in V \setminus \{i, j\}\}$, 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($\mathcal{P}^-(\tilde{E}, \zeta)$) 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., $E^c = \{i \leftarrow j, i \rightarrow j, \forall i, j \in V : i \neq j\}$. 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 $E_{i, |W| + 1, j} = \{i \rightarrow |W| + 1, |W| + 1 \leftarrow j\}$ 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 λ^* 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 λ^* is equivalent to the λ 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 $\lambda = \lambda^*$. \square

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 $t \in \{1, 2, 3\}$. \square

LEMMA EC.6. CAUSALIP($\mathcal{P}^-(\tilde{E}, \zeta)$) 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($\mathcal{P}^-(\tilde{E}, \zeta)$) 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 $\mathbf{y} = \mathbf{0}$ 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 $n \in N_{ij}^I$, $p \in \mathcal{P}_{ij}(\tilde{E}, |V| - 1)$ 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. \square

LEMMA EC.7. Let $\mathcal{G}_s = (V, E_s)$ be an input graph to \mathcal{G} -POSTPROCESS at iteration t and let ϵ 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 ϵ 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, $i \not\perp_{\mathcal{G}_s} j | C_{ij}^n$. \square

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) $\omega^\top \epsilon = 0$ 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., $\omega^\top \epsilon = 0$, 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. \square

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

- (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) $\omega^\top \epsilon = 0$ or (2) $\tilde{E} = E^c$ and $\zeta = |V| - 1$ (by Step 3.3 of EDGEGEN). Note that by Lemma EC.7, $\omega^\top \epsilon$ is equivalent to the loss function $L(\mathcal{G})$ given in (3):

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

Note that the loss function is lower bounded by 0. Suppose condition (1) holds at iteration s^* and EDGEGEN terminates. Because $\omega > \mathbf{0}$, this implies $\sigma_s > 0$ for all $s < s^*$. Hence, by Step 4 of EDGEGEN, the graph returned is \mathcal{G}_{s^*} . Because $\omega^\top \epsilon = 0$ 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($\mathcal{P}(E^c, |V| - 1)$) 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) $\omega^\top \epsilon = 0$ or (2) $\tilde{E} = E^c$ and $\zeta = |V| - 1$. First, suppose condition (1) holds at s^* for some $\omega > 0$. Then by the proof of part (i), EDGEGEN returns graph \mathcal{G}_{s^*} . Because condition (1) holds and $\omega > 0$, 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($\mathcal{P}(E^c, |V| - 1)$) 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 . \square