
Finding Invariant Predictors Efficiently via Causal Structure (Supplementary Material)

Kenneth Lee¹

Md Musfiqur Rahman¹

Murat Kocaoglu¹

¹School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA

A PROOFS OF LEMMAS AND THEOREMS

A.1 PROOF OF THEOREM 3.2

Theorem A.1. *There exists a hedge for $P(\mathbf{Y}|do(\mathbf{X}), \mathbf{W})$ according to the generalized hedge condition if and only if $P(\mathbf{Y}|do(\mathbf{X}), \mathbf{W})$ is unidentifiable in G .*

Proof. Let $\mathbf{Z} \subseteq \mathbf{W}$ be the maximal set such that $P(Y|do(\mathbf{X}), \mathbf{W}) = P(Y|do(\mathbf{X}, \mathbf{Z}), \mathbf{W} \setminus \mathbf{Z})$. By Theorem 21 in [4], $P(\mathbf{Y}|do(\mathbf{X}), \mathbf{W})$ is identifiable in G if and only if $P(Y, \mathbf{X} \setminus \mathbf{Z}|do(\mathbf{X}, \mathbf{Z}))$ is identifiable in G . By Theorem 3.2, there exists a hedge for $P(Y, \mathbf{W} \setminus \mathbf{Z}|do(\mathbf{X}, \mathbf{Z}))$ if and only if $p(y, \mathbf{W} \setminus \mathbf{Z}|do(\mathbf{X}, \mathbf{Z}))$ is unidentifiable in G . Therefore, we can apply the definition of hedge for $P(Y, \mathbf{W} \setminus \mathbf{Z}|do(\mathbf{X}, \mathbf{Z}))$ to formulate definition 3.1 such that there exists a hedge for $P(\mathbf{Y}|do(\mathbf{X}), \mathbf{W})$ according to definition 3.1 if and only if $P(\mathbf{Y}|do(\mathbf{X}), \mathbf{W})$ is unidentifiable in G . \square

A.2 PROOF OF LEMMA 3.5

Lemma A.2. *Let $\mathbf{Y} = \{Y\}$. The output of **Find-MACS-on-set**(G, \mathbf{Y}) is the MACS of Y . The MACS of Y is a Y -rooted C-tree.*

Proof. The line 3 of the Algorithm 2 in Section B.2, first gets an induced subgraph of G over $An(Y)$. That implies every variable in the resulting graph has a directed path to Y and Y does not have any child in it. Then, in step 4, it recursively calls on **Find-MACS-on-set** with $G_{An(Y)}$. Then, every variable in $G_{An(Y)}$ must be in $An(Y)$. The execution of the algorithm will then move to step 6 to get an induced subgraph of $G_{An(Y)}$ over $C(Y)$.

If there is no bidirected path from any variable to Y , then the algorithm will return Y , which is a Y -rooted C-tree. Suppose otherwise that there is a bidirected path from some variables to Y , then we have two cases: Case i.) The variables \mathbf{M} have bidirected paths to Y , but $De(\mathbf{M}) \cap An(Y) \setminus \mathbf{M}$ are not in $C(Y)$ such that $G_{C(Y)}$ is not a Y -rooted C-tree. Case ii.) The variables \mathbf{M} have bidirected paths to Y and $De(\mathbf{M}) \cap An(Y) \setminus \mathbf{M}$ are in $C(Y)$ such that $G_{C(Y)}$ is a Y -rooted C-tree. For case i, when the algorithm recursively call on itself at Step 7. Since $De(\mathbf{M}) \cap An(Y) \setminus \mathbf{M}$ are not in $C(Y)$, $\mathbf{M} \notin An(Y)$ in $G_{C(Y)}$. Then it will return Y as G . For case ii, the result trivially follows. \square

A.3 PROOF OF THEOREM 3.6

Theorem A.3. *For some $W \in Ch(S)$, if there exists a hedge for $P(Y|do(W))$, then for any $\mathbf{H}, \mathbf{J} \subseteq \mathbf{V}$, we have $(Y \not\perp\!\!\!\perp S|\mathbf{J})_{G_{\overline{H}}}$ or $P(Y|do(\mathbf{H}), \mathbf{J})$ is unidentifiable in G .*

Proof. Suppose there exists a Y -rooted C-tree F in G such that there exists a hedge for $P(Y|do(W))$ for some $W \in Ch(S)$. We will show that for any $\mathbf{H}, \mathbf{J} \subseteq \mathbf{V}$, we have $(Y \not\perp\!\!\!\perp S|\mathbf{J})_{G_{\overline{H}}}$ or $P(Y|do(\mathbf{H}), \mathbf{J})$ is unidentifiable in G .

First, suppose \mathbf{H} does not contain any member in c-forest F . We will show that there exists an inducing path from S to Y such that $(Y \not\perp\!\!\!\perp S|\mathbf{J})_{G_{\overline{\mathbf{H}}}}$. Since every member in $F \setminus Y$ must have only one child and Y does not any children in G_F , every member in F is in $An(Y)$. By definition of hedge, $W \in F$. As F is a C-component by the definition of C-tree and $W \in Ch(S)$ and $W \in F$, for a member S_W of S that is $Pa(W)$, we can have a path from S_W to Y through the directed path from W to Y in F along which every variable on that path is a collider. Therefore, there exists an inducing path from S_W to F . By theorem 4.2 in [1], S_W cannot be d-separated from Y in G if and only if there exists an inducing path from S_W to Y in G . Therefore, $(Y \not\perp\!\!\!\perp S|\mathbf{J})_{G_{\overline{\mathbf{H}}}}$ for any $\mathbf{J} \subseteq \mathbf{V}$. Next, suppose \mathbf{H} contains some members of F . By lemma A.17, that $P(Y|do(\mathbf{H}), \mathbf{J})$ is unidentifiable in G for any $\mathbf{J} \subseteq \mathbf{V}$. \square

A.4 PROOF OF THEOREM 3.7

Theorem A.4. *If the selection variable S is a parent of MACS T_Y , then there is no graph surgery estimator in G .*

Proof. Let S be a parent of T_Y , where T_Y is the MACS of Y in G . By Lemma 3.5, T_Y is a Y -rooted C-tree. Since $S \in Pa(T_Y)$, there exists some members $W \in Ch(S)$ that are in T_Y such that there exists a hedge for $P(Y|do(W))$ by the following construction: let $F = T_Y$, $F' = \{Y\}$ to be two \mathbf{R} -rooted C-forest, where $\mathbf{R} = \{Y\}$. By Theorem 3.6, the result follows implying that there is no graph surgery estimator in G . \square

A.5 PROOF OF THEOREM 3.8

Theorem A.5. *Let T_Y be the MACS of Y in G , $\mathcal{H} := \{H : H \in Ch(Y), Pa(T_H) \not\supseteq S\}$ and $T_J := \bigcup_{H \in \mathbf{K}} T_H$ for any $\mathbf{K} \subseteq \mathcal{H}$, where T_H is the MACS with respect to the variable H . Let $\mathbf{D} = Pa(T_Y \cup T_J)$. If S is not a parent of T_Y , then $P(Y|do(\mathbf{D}), \mathbf{K}, \mathbf{W})$ is identifiable in G and $(Y \perp\!\!\!\perp S|\mathbf{W}, \mathbf{K})_{G_{\overline{\mathbf{D}}}}$ for any $\mathbf{W} \subseteq (T_Y \cup T_J) \setminus (Y \cup \mathbf{K})$.*

Proof. Let T_Y be the MACS of Y in G . Let $\mathcal{H} := \{H : H \in Ch(Y), Pa(T_H) \not\supseteq S\}$, $T_J := \bigcup_{H \in \mathbf{K}} T_H$ for some $\mathbf{K} \subseteq \mathcal{H}$. Let $\mathbf{D} = Pa(T_Y \cup T_J)$. Recall that by the conditions of the theorem, S is not a parent of T_Y .

Consider any $\mathbf{W} \subseteq (T_Y \cup T_J) \setminus (Y \cup \mathbf{K})$. We first prove the following claim.

Claim: $P(Y|do(\mathbf{D}), \mathbf{K}, \mathbf{W}) \neq P(Y|do(\mathbf{D}), \mathbf{Z}, (\mathbf{K} \cup \mathbf{W}) \setminus \mathbf{Z})$ for any $\mathbf{Z} \subset \mathbf{K} \cup \mathbf{W}, \mathbf{Z} \neq \emptyset$.

Proof. For the sake of contradiction, suppose there exists a non-empty subset \mathbf{Z} where the equality holds. Under the extended faithfulness assumption, this would only be true if do-calculus Rule 2 is applicable through the following graphical condition:

$$(Y \perp\!\!\!\perp \mathbf{Z}|\mathbf{K} \cup \mathbf{W} \setminus \mathbf{Z})_{G_{\mathbf{Z}, \overline{\mathbf{D}}}} \quad (1)$$

Now observe that any node U in $T_Y \cup T_J$ must belong to a T_H for some $H \in \mathbf{K}$ or T_Y . By definition of T_H (or T_Y), there must be a bidirected path from U to H (or Y) that only goes through nodes in T_H (or T_Y). By definition of T_H (or T_Y), any node along this bidirected path must be an ancestor of H (or Y). Therefore, there is a d-connecting path from U to H (or Y) that starts with an arrow into U . Note that further conditioning cannot break this path since the path only consists of colliders. Finally, for the case where U belongs to a T_H , as the conditioning set contains H , we can concatenate this path with the edge $Y \rightarrow H$ to obtain a d-connecting path to Y , since H is a collider along this concatenated path. This contradicts with the d-separation statement above. \square

For the sake of contradiction, suppose for some $\mathbf{W} \subseteq (T_Y \cup T_J) \setminus (Y \cup \mathbf{K})$, either $P(Y|do(\mathbf{D}), \mathbf{K}, \mathbf{W})$ is unidentifiable in G or $(Y \not\perp\!\!\!\perp S|\mathbf{W}, \mathbf{K})_{G_{\overline{\mathbf{D}}}}$.

Suppose $P(Y|do(\mathbf{D}), \mathbf{K}, \mathbf{W})$ is unidentifiable in G .

By Theorem 3.2, there must exists a hedge for $P(Y|do(\mathbf{D}), \mathbf{K}, \mathbf{W})$. The claim above implies that the maximal set \mathbf{Z} such that $P(Y|do(\mathbf{D}), \mathbf{K}, \mathbf{W}) = P(Y|do(\mathbf{D}), \mathbf{Z}, (\mathbf{K} \cup \mathbf{W}) \setminus \mathbf{Z})$ is an empty set by rule 2 of do-calculus.

Since $\mathbf{Z} = \emptyset$, by Definition 3.1, there exists a hedge for $P(Y|do(\mathbf{D}), \mathbf{K}, \mathbf{W})$ only if there exists two \mathbf{R} -rooted C-forests F, F' such that $F \cap \mathbf{D} \neq \emptyset$ and $F' \cap \mathbf{D} = \emptyset$ for some $\mathbf{R} \subset An(Y \cup \mathbf{K} \cup \mathbf{W})_{G_{\overline{\mathbf{D}}}}$. Consider any such \mathbf{R}, F, F' . Since $F' \cap \mathbf{D} = \emptyset$ and F' is a \mathbf{R} -rooted C-forest, it must be the case that $\mathbf{D} \cap \mathbf{R} = \emptyset$. Since $\mathbf{D} \cap \mathbf{R} = \emptyset$ and intervening on \mathbf{D} removes the incoming edges of \mathbf{D} , any member of $An(Y \cup \mathbf{K} \cup \mathbf{W})$ will be in $T_Y \cup T_J$ such that $\mathbf{R} \subseteq T_Y \cup T_J$. But this is a contradiction due to the following:

Suppose $\mathbf{R} \subseteq T_Y \cup T_J$. Since $\mathbf{D} \cap \mathbf{R} = \emptyset$ and $F \cap \mathbf{D} \neq \emptyset$, that implies some members of \mathbf{D} must have a directed path to some members in \mathbf{R} . If that is the case, then that member of \mathbf{D} must also be in $T_Y \cup T_H$, where T_Y is a Y -rooted C-tree and T_H is a H -rooted C-tree for some $H \in Ch(Y)$, which is also in T_J . But this is a contradiction as any member of \mathbf{D} cannot be in $T_Y \cup T_J$ by the definition of \mathbf{D} . Thus, $\mathbf{R} \not\subseteq (T_Y \cup T_J)$.

Therefore, there is no hedge for $P(Y|do(\mathbf{D}), \mathbf{K}, \mathbf{W})$. By Theorem 3.2, $P(Y|do(\mathbf{D}), \mathbf{K}, \mathbf{W})$ is identifiable in G .

Suppose $(Y \perp\!\!\!\perp S|\mathbf{W}, \mathbf{K})_{G_{\overline{\mathbf{D}}}}$.

Next, we will show $(Y \perp\!\!\!\perp S|\mathbf{W}, \mathbf{K})_{G_{\overline{\mathbf{D}}}}$ for any $\mathbf{W} \subseteq (T_Y \cup T_J) \setminus (Y \cup \mathbf{K})$. Given that S is not a parent of T_Y , we will consider two cases separately: *i.)* $S \in An(Y)$; *ii.)* $S \notin An(Y)$

$S \in An(Y)$: Suppose there exists a d-connecting path from S to Y by conditioning on $\mathbf{W} \cup \mathbf{K}$ and intervening on \mathbf{D} for some $\mathbf{W} \subseteq (T_Y \cup T_J) \setminus (Y \cup \mathbf{K})$. First, there is no directed path from S to Y in $G_{\overline{\mathbf{D}}}$ since S is not a parent of T_Y by the conditions of the theorem and we intervene on \mathbf{D} , which is a superset of $Pa(T_Y)$ and $Y \in T_Y$. For some $\mathbf{W} \subseteq (T_Y \cup T_J) \setminus (Y \cup \mathbf{K})$, conditioning on \mathbf{W} must have opened paths with colliders that are in $An(\mathbf{W})$. However, since any member of \mathbf{W} is in $T_Y \cup T_J$ and any incoming edges of $Pa(T_Y \cup T_J)$ are removed in $G_{\overline{\mathbf{D}}}$ and S is not a parent of any member of T_J by definition of T_J , there cannot be a d-connecting path from S to Y in $G_{\overline{\mathbf{D}}}$. Therefore, a contradiction. We have $(Y \perp\!\!\!\perp S|\mathbf{W}, \mathbf{K})_{G_{\overline{\mathbf{D}}}}$ for any $\mathbf{W} \subseteq (T_Y \cup T_J) \setminus (Y \cup \mathbf{K})$ when $S \in An(Y)$.

$S \notin An(Y)$: we will show *i.)* there is no d-connecting path from S to Y ends with a member of $Ch(Y)$ and *ii.)* there is no d-connecting path from S to Y ends with a bidirected neighbor of Y .

Show that there is no d-connecting path from S to Y that ends with a member of $Ch(Y)$. Any child of Y where parents of its MACS do not contain S will be in $T_J = \bigcup_{H \in \mathbf{K}} T_H$ for some $\mathbf{K} \subseteq \mathcal{H}$ and we intervene on $Pa(T_Y \cup T_J)$ so that there is no path from S to any such child of Y . Suppose the d-connecting path from S to Y ends with some other members of $Ch(Y)$. However, for any such path, we must have conditioned on the descendants of those children of Y to open the path from S to Y through some descendants of Y , but any child of Y that is in \mathbf{K} , its parents are also intervened such that any path from S to Y through those children are blocked. For any other children that are not in \mathbf{K} either form a collider or their descendants form a collider to block any other active paths from S to Y . Therefore, there is no d-connecting path from S to Y that ends with a member of $Ch(Y)$.

Show that there is no d-connecting path from S to Y ends with a bidirected neighbor of Y . Suppose further that bidirected neighbor of Y is a child of Y . From above, we have proved there is no d-connecting path from S to Y ends with a member of $Ch(Y)$. Suppose that bidirected neighbor of Y is not a child of Y and there exists some d-connecting paths from S to Y that ends with those bidirected neighbor of Y , for the case where there is no any descendant of those bidirected neighbors of Y is in \mathbf{W} , any path from S to Y along that bidirected neighbor of Y is blocked as there exists a collider along any such path and that bidirected neighbor is not in T_J . Therefore, there is no d-connecting path from S to Y . For the case where there exists some descendants of those bidirected neighbors \mathbf{X} of Y is in \mathbf{W} , but any parent of those descendants must be in \mathbf{D} such that all the incoming edges of any such parent are removed so that any d-connecting path from S to any of those parents will be blocked. Any d-connecting path from S to the members along the upstream path of those parents to \mathbf{X} will be blocked by a collider along the path by concatenating the path $Y \rightarrow X$ or $M \rightarrow L$ for some descendants M, L of \mathbf{X} , where $M \neq L$.

Therefore, we have $(Y \perp\!\!\!\perp S|\mathbf{W}, \mathbf{K})_{G_{\overline{\mathbf{D}}}}$. □

A.6 PROOF OF THEOREM 3.10

Theorem A.6. Let T_Y be the MACS of Y in G , T_H be the MACS of a child H of Y in G . Define $T_C := \bigcup_{H \in Ch(Y)} T_H$, $\mathcal{Z} := \{Z : Z \in (C(Y) \cap Nbr(Y)) \setminus (T_Y \cup T_C) \text{ s.t. } Pa(T_{Y \cup Z}) \not\subseteq S\}$ and $T_B := \bigcup_{Z \in \mathbf{M}} T_{Y \cup Z}$ for any $\mathbf{M} \subseteq \mathcal{Z}$ where $T_{Y \cup Z}$ is the MACS for the set $(Y \cup Z)$. Let $\mathbf{D} = Pa(T_B)$. If S is not a parent of T_Y , then $P(Y|do(\mathbf{D}), \mathbf{M}, \mathbf{W})$ is identifiable in G and $(Y \perp\!\!\!\perp S|\mathbf{W}, \mathbf{M})_{G_{\overline{\mathbf{D}}}}$ for any $\mathbf{W} \subseteq (T_B) \setminus (Y \cup \mathbf{M})$.

Proof. Let T_Y be the MACS of Y in G , T_H be the MACS of a child H of Y in G . Define $T_C := \bigcup_{H \in Ch(Y)} T_H$. Let $\mathcal{Z} := \{Z : Z \in (C(Y) \cap Nbr(Y)) \setminus (T_Y \cup T_C) \text{ s.t. } Pa(T_{Y \cup Z}) \not\subseteq S\}$, $T_B := \bigcup_{Z \in \mathbf{M}} T_Z$ for some $\mathbf{M} \subseteq \mathcal{Z}$. Let $\mathbf{D} = Pa(T_B)$. Recall that by the conditions of the theorem, S is not a parent of T_Y .

Consider any $\mathbf{W} \subseteq (T_B) \setminus (Y \cup \mathbf{M})$. We first prove the following claim.

Claim: $P(Y|do(\mathbf{D}), \mathbf{M}, \mathbf{W}) \neq P(Y|do(\mathbf{D}, \mathbf{Q}), (\mathbf{M} \cup \mathbf{W}) \setminus \mathbf{Q})$ for any $\mathbf{Q} \subset \mathbf{M} \cup \mathbf{W}$, $\mathbf{Q} \neq \emptyset$.

Proof. For the sake of contradiction, suppose there exists a non-empty subset \mathbf{Q} where the equality holds. Under the extended faithfulness assumption, this would only be true if do-calculus Rule 2 is applicable through the following graphical condition:

$$(Y \perp\!\!\!\perp \mathbf{Q} | \mathbf{M} \cup \mathbf{W} \setminus \mathbf{Q})_{G_{\mathbf{Q}, \overline{\mathbf{B}}}} \quad (2)$$

Now observe that any node U in T_B must belong to a $T_{Y \cup Z}$ for some $Z \in \mathbf{M}$. By definition of $T_{Y \cup Z}$, there must be a bidirected path from U to Z and Y that only goes through nodes in $T_{Y \cup Z}$. By definition of $T_{Y \cup Z}$, any node along this bidirected path must be an ancestor of Z or Y . Therefore, there is a d-connecting path from U to Z or Y that starts with an arrow into U . Note that further conditioning cannot break this path since the path only consists of colliders. Finally, for the case where there is a d-connecting path from U to Z , as the conditioning set contains Z , we can concatenate this path with the edge $Y \leftrightarrow Z$ to obtain a d-connecting path to Y , since Z is a collider along this concatenated path. This contradicts with the d-separation statement above. \square

For the sake of contradiction, suppose for some $\mathbf{W} \subseteq (T_B) \setminus (Y \cup \mathbf{M})$, either $P(Y|do(\mathbf{D}), \mathbf{M}, \mathbf{W})$ is unidentifiable in G or $(Y \not\perp\!\!\!\perp S | \mathbf{W}, \mathbf{M})_{G_{\overline{\mathbf{B}}}}$.

Suppose $P(Y|do(\mathbf{D}), \mathbf{M}, \mathbf{W})$ is unidentifiable in G .

By Theorem 3.2, there must exist a hedge for $P(Y|do(\mathbf{D}), \mathbf{M}, \mathbf{W})$. The claim above implies that the maximal set \mathbf{Q} such that $P(Y|do(\mathbf{D}), \mathbf{M}, \mathbf{W}) = P(Y|do(\mathbf{D}, \mathbf{Q}), \mathbf{M} \cup \mathbf{W} \setminus \mathbf{Q})$ is an empty set by rule 2 of do-calculus.

Since $\mathbf{Q} = \emptyset$, by Definition 3.1, there exists a hedge for $P(Y|do(\mathbf{D}), \mathbf{M}, \mathbf{W})$ only if there exists two \mathbf{R} -rooted C-forests F, F' such that $F \cap \mathbf{D} \neq \emptyset$ and $F' \cap \mathbf{D} = \emptyset$ for some $\mathbf{R} \subset An(Y \cup \mathbf{M} \cup \mathbf{W})_{G_{\overline{\mathbf{B}}}}$. Consider any such \mathbf{R}, F, F' . Since $F' \cap \mathbf{D} = \emptyset$ and F' is a \mathbf{R} -rooted C-forest, it must be the case that $\mathbf{D} \cap \mathbf{R} = \emptyset$. Since $\mathbf{D} \cap \mathbf{R} = \emptyset$ and intervening on \mathbf{D} removes the incoming edges of \mathbf{D} , any member of $An(Y \cup \mathbf{M} \cup \mathbf{W})$ will be in T_B such that $\mathbf{R} \subseteq T_B$. But this is a contradiction due to the following:

Suppose $\mathbf{R} \subseteq T_B$. Since $\mathbf{D} \cap \mathbf{R} = \emptyset$ and $F \cap \mathbf{D} \neq \emptyset$, that implies some members of \mathbf{D} must have a directed path to some members in \mathbf{R} . If that is the case, then some members of \mathbf{D} must also be in $T_{Y \cup Z}$, which is also in T_B . But this is a contradiction as any member of \mathbf{D} cannot be in T_B by the definition of \mathbf{D} . Thus, $\mathbf{R} \not\subseteq T_B$.

Therefore, there is no hedge for $P(Y|do(\mathbf{D}), \mathbf{M}, \mathbf{W})$. By Theorem 3.2, $P(Y|do(\mathbf{D}), \mathbf{M}, \mathbf{W})$ is identifiable in G .

Suppose $(Y \not\perp\!\!\!\perp S | \mathbf{W}, \mathbf{M})_{G_{\overline{\mathbf{B}}}}$.

Next, we will show $(Y \perp\!\!\!\perp S | \mathbf{W}, \mathbf{M})_{G_{\overline{\mathbf{B}}}}$ for any $\mathbf{W} \subseteq (T_B) \setminus (Y \cup \mathbf{M})$. Given that S is not a parent of T_Y , we will consider two cases separately: *i.*) $S \in An(Y)$; *ii.*) $S \notin An(Y)$

$S \in An(Y)$: Suppose there exists a d-connecting path from S to Y by conditioning on $\mathbf{W} \cup \mathbf{M}$ and intervening on \mathbf{D} for some $\mathbf{W} \subseteq T_B \setminus (Y \cup \mathbf{M})$. First, there is no directed path from S to Y in $G_{\overline{\mathbf{D}}}$ since S is not a parent of T_Y by the conditions of the theorem and we intervene on \mathbf{D} , which is a superset of $Pa(T_Y)$ and $Y \in T_Y$ since $T_Y \subset T_B$. For some $\mathbf{W} \subseteq T_B \setminus (Y \cup \mathbf{M})$, conditioning on \mathbf{W} must have opened paths with colliders that are in $An(\mathbf{W})$. However, since any member of \mathbf{W} is in T_B and any incoming edges of $Pa(T_B)$ are removed in $G_{\overline{\mathbf{D}}}$ and S is not a parent of any member of $T_{Y \cup Z}$ by definition of T_B , there cannot be a d-connecting path from S to Y in $G_{\overline{\mathbf{D}}}$. Therefore, a contradiction. We have $(Y \perp\!\!\!\perp S | \mathbf{W}, \mathbf{M})_{G_{\overline{\mathbf{B}}}}$ for any $\mathbf{W} \subseteq T_B \setminus (Y \cup \mathbf{M})$ when $S \in An(Y)$.

$S \notin An(Y)$: we will show *i.*) there is no d-connecting path from S to Y ends with a member of $Ch(Y)$ and *ii.*) there is no d-connecting path from S to Y ends with a bidirected neighbor of Y .

Show that there is no d-connecting path from S to Y that ends with a member of $Ch(Y)$. For the case where there is no descendant of any children of Y are in \mathbf{W} . Since \mathbf{W} does not contain any child of Y , any d-connecting path from S to any child J of Y , we can concatenate this path with the edge $Y \rightarrow J$ to obtain a blocked path to Y as J is a collider along this concatenated path. Suppose there exists some descendants of some children of Y that are in \mathbf{W} . Note that for any member in \mathbf{W} , its parents are in \mathbf{D} such that all incoming edges of those parents are removed and there cannot be a d-connecting path from S to Y by concatenating the path $Y \rightarrow J$ for any child J of Y as any J is not in \mathbf{W} nor such path can be opened by conditioning on \mathbf{W} .

Show that there is no d-connecting path from S to Y ends with a bidirected neighbor of Y . Suppose further that bidirected neighbor of Y is a child of Y . From above, we have proved there is no d-connecting path from S to Y ends with a member of $Ch(Y)$. Any bidirected neighbor of Y where it is not a child of Y and the parents of its MACS do not contain S will be in $T_B = \bigcup_{Z \in \mathbf{M}} T_{Y \cup Z}$ for some $\mathbf{M} \subseteq \mathcal{Z}$ and we intervene on $Pa(T_B)$ so that there is no path from S to any such bidirected

neighbor of Y . Suppose the d-connecting path from S to Y ends with some other members of bidirected neighbors that is not child of Y . Since those bidirected neighbors \mathbf{A} are not in \mathbf{W} , there exists some descendants of \mathbf{A} are in \mathbf{W} . However, for any such descendant, its parent must be in \mathbf{D} such that all the incoming edges of that parent are removed. That implies any such path is blocked. Since no member of \mathbf{A} are in \mathbf{W} , any d-connecting path will be blocked by concatenating $Y \rightarrow A$ and having a collider $A \in \mathbf{A}$ along that concatenated path. Thus, there is no d-connecting path from S to Y ends with a bidirected neighbor of Y .

Therefore, we have $(Y \perp\!\!\!\perp S | \mathbf{W}, \mathbf{M})_{G_{\overline{\mathbf{D}}}}$. □

A.7 PROOF OF THEOREM 3.11

We will first prove the following lemma.

Lemma A.7. *If $S \notin An(Y)$ and **ID4IP** (Algorithm 3) returns **FAIL**, then there is no parent of Y in G .*

Proof. For the sake of contradiction, assume there exists some parents of Y in G . Given as the conditions of the lemma, we know that $S \notin An(Y)$ and **ID4IP** returns **FAIL**. Since $S \notin An(Y)$, **ID4IP** will only return **FAIL** at line 13 since line 6 requires that $S \in An(Y)$. $P(Y|do(Pa(T_Y)), T_Y \setminus Y)$ will be a graph surgery estimator by Corollary 3.9. This is a contradiction because **ID4IP** returns **FAIL** at line 13 due to $P_{set} = \emptyset$. Therefore, if $S \notin An(Y)$ and **ID4IP** returns **FAIL**, there is no parent of Y in G . □

Theorem A.8. *If there exists a graph surgery estimator, **ID4IP** outputs a graph surgery estimator.*

Claim: Given $S \in An(Y)$, if there exists a graph surgery estimator, **ID4IP** (Algorithm 3) outputs a graph surgery estimator.

We will first prove the above claim

Proof. As given by the condition, $S \in An(Y)$ and there exists a graph surgery estimator. By Theorem 3.7, $S \notin Pa(T_Y)$, where T_Y is the MACS of Y . S has no incoming edges by problem set up so S cannot be in T_Y . Also, since there is a directed path from S to Y and S is not a parent of T_Y , $S \in An(M)$ for some $M \in Pa(T_Y)$. By Corollary 3.9, **ID4IP** will output a graph surgery estimator at line 7 of the algorithm. □

Next, we will prove the theorem.

Proof. With the claim above, we only need to consider the case where $S \notin An(Y)$. Suppose $S \notin An(Y)$. Note that since $S \notin An(Y)$, this **FAIL** is returned by line 13 of the algorithm.

We will prove it by using contraposition. Suppose **ID4IP** returns **FAIL**, for the sake of contradiction, suppose also there exists a graph surgery estimator $P(Y|do(\mathbf{Q}), \mathbf{W})$ for some $\mathbf{Q} \subseteq \mathbf{V}$ and some $\mathbf{W} \subseteq \mathbf{V} \setminus \mathbf{Q}$.

We will consider the following cases where i.) $\mathbf{Q} = \emptyset, \mathbf{W} \neq \emptyset$; ii.) $\mathbf{Q} \neq \emptyset, \mathbf{W} = \emptyset$; iii.) $\mathbf{Q} \neq \emptyset, \mathbf{W} \neq \emptyset$

i.) $\mathbf{Q} = \emptyset, \mathbf{W} \neq \emptyset$. Given $\mathbf{Q} = \emptyset, \mathbf{W} \neq \emptyset$, it must be the case that there exists some $W \in \mathbf{W}$ that has a d-connecting path to Y . By Lemma A.7, Y has no parents in G . It implies that any such path must end at a child of Y or a bidirected neighbor of Y . If such path ends at a child of Y , for any such child K of Y , there exists an inducing path from S to K since $S \in Pa(T_K)$ and T_K is a K -rooted C-tree. It implies that there exists no subset in \mathbf{V} such that Y and S can be d-separated by Theorem 4.2 in [1]. Since $\mathbf{Q} = \emptyset$, we have $(Y \not\perp\!\!\!\perp S | \mathbf{W})_G$, which is a contradiction. Similarly, if such path ends at a bidirected neighbor of Y that is not a child of Y , for any such bidirected neighbor Z of Y , $S \in Pa(T_{Y \cup Z})$, which implies $(Y \not\perp\!\!\!\perp S | \mathbf{W})_G$. It is because **ID4IP** returns **FAIL** implies that S is a parent of the MACS of $\{Y, Z\}$ for any bidirected neighbor Z of Y that is not a child of Y . Observe that any node U in $T_{Y \cup Z}$ for any Z , U has a bidirected path to Z . By the definition of $T_{Y \cup Z}$, any node along this bidirected path must be an ancestor of Z . Therefore, there is a d-connecting path from U to Z that starts with an arrow into U . Note that further conditioning cannot break this path since the path only consists of colliders. As $\mathbf{W} \neq \emptyset$, if Z is not in \mathbf{W} , then any such member will be d-separated from Y by concatenating any d-connecting path from U to Z with $Y \leftrightarrow Z$ such that $P(Y | \mathbf{W}) = P(Y)$. If Z is in \mathbf{W} , then $(Y \not\perp\!\!\!\perp S | \mathbf{W})_G$. Therefore, we have reached a contradiction for the case when $\mathbf{Q} = \emptyset, \mathbf{W} \neq \emptyset$.

ii.) $\mathbf{Q} \neq \emptyset, \mathbf{W} = \emptyset$. By Lemma A.7, there is no parent of Y in G . Any member of \mathbf{Q} must be non-ancestors of Y . Then, either Y has a directed path to some $Q \in \mathbf{Q}$ or there is no directed path to any $Q \in \mathbf{Q}$ such that $(Y \perp\!\!\!\perp \mathbf{Q})_{G_{\overline{\mathbf{Q}}}}$. By rule 3 of do-calculus, that implies $P(Y|do(\mathbf{Q})) = P(Y)$, which is a contradiction.

iii.) $\mathbf{Q} \neq \emptyset, \mathbf{W} \neq \emptyset$. We consider the case where $P(Y|do(\mathbf{Q}), \mathbf{W}) \neq P(Y|\mathbf{W})$ and $P(Y|do(\mathbf{Q}), \mathbf{W}) \neq P(Y|do(\mathbf{Q}))$ as we have reached contradiction for those cases. Since there is no parent of Y in G by Lemma A.7, both \mathbf{Q} and \mathbf{W} must be non-ancestors of Y . Also, since **ID4IP** returns **FAIL**, there is no children H of Y where $S \notin Pa(T_H)$ and there is also no bidirected neighbor Z of Y such that $S \notin Pa(T_{Y \cup Z})$.

If S is not connected with Y and **ID4IP** return **FAIL**, then there will be no children or bidirected neighbors of Y . It is because S is disconnected with Y implies $S \notin Pa(T_H)$ and $S \notin Pa(T_{Y \cup Z})$ for any child and bidirected neighbor of Y . To see this, suppose there exist some children of Y and bidirected neighbor of Y in G , **ID4IP** will not return **FAIL** as any of those children and bidirected neighbors will be in either T_H or $T_{Y \cup Z}$ for some children of Y , H and bidirected neighbors Z of Y . We can have predictors return at line 8 and 10 of Algorithm 3. Thus, it is a contradiction. Since there are no children of Y and bidirected neighbors of Y and no parents of Y , any variable in \mathbf{V} will be d-separated from Y in G so that any query will be equal to $P(Y)$ such that there is no graph surgery estimator.

Suppose S is connected with Y . Since $S \notin An(Y)$, for S to be connected with Y , any path from S to Y must have a collider. Let \mathbf{J} be the largest set such that $P(Y|do(\mathbf{Q}), \mathbf{W}) = P(Y|do(\mathbf{Q} \cup \mathbf{J}), \mathbf{W} \setminus \mathbf{J})$ by rule 2 of do-calculus. In addition, let \mathbf{M} be the largest set such that $P(Y|do(\mathbf{Q} \cup \mathbf{J}), \mathbf{W} \setminus \mathbf{J}) = P(Y|do((\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}), \mathbf{W} \setminus \mathbf{J})$ by rule 3 of do-calculus. If $(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M} = \emptyset$, then we reach the same contradiction as (i.). Suppose $(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M} \neq \emptyset$, it must be the case that any member in $(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}$ is d-connected with Y when conditioning on $\mathbf{W} \setminus \mathbf{J}$ in $G_{(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}}$ as implied by rule 3 of do-calculus.

Any such d-connecting path from a member A of $(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}$ to Y must begin with an outgoing edge of some members N in $(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}$ in $G_{(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}}$. Since Y does not have any parents, this d-connecting path must end at some children of Y or some bidirected neighbors of Y that are in $\mathbf{W} \setminus \mathbf{J}$. However, for any such child H or bidirected neighbor Z of Y , it must be that $S \in Pa(T_H)$ and $S \in Pa(T_{Y \cup Z})$ as implied by the condition that **ID4IP** returns **FAIL**.

For the case where the d-connecting path ends at H , without loss of generality, we consider the following three cases: *i.* N is in T_H or *ii.* $N \in An(T_H) \setminus T_H$ or *iii.* $N \in De(T_H) \setminus T_H$. Note that the case $N \in T_H$ subsumes the case when H is also a bidirected neighbor of Y .

i. $N \in T_H$: If N is in T_H , then $P(Y|do(\mathbf{Q}), \mathbf{W})$ is unidentifiable in G by the following construction of hedge condition: for a query $P(Y|do((\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}), \mathbf{W} \setminus \mathbf{J})$, we can let $\mathbf{R} = \{H\}$, which is a proper subset of $An(Y \cup (\mathbf{W} \setminus \mathbf{J}))$. Also, we can let $F' = \{H\}$ and $F = T_H$ as T_H is a H -rooted C-tree. The result then follows Theorem 3.2.

For the next two cases, we will make use of this observation: observe that for any node U in T_H . By the definition of T_H , there must be a bidirected path from U to H that only goes through nodes in T_H and any node along this bidirected path must be an ancestor of H . Therefore, there is a d-connecting path from U to H that starts with an arrow into U . Note that further conditioning cannot break this path since the path only consists of colliders.

ii. $N \in An(T_H) \setminus T_H$: If N is in the $An(T_H) \setminus T_H$ and $H \in \mathbf{W} \setminus \mathbf{J}$, then $(Y \not\perp\!\!\!\perp S|\mathbf{W} \setminus \mathbf{J})_{G_{(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}}}$. It is because $S \in Pa(T_H)$ such that we can construct a d-connecting path from S to Y by concatenating $Y \rightarrow H$ with the bidirected path as outlined in the observation. Note that no member in $(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}$ can be in T_H due to unidentifiability of the query as shown previously. Therefore, we have a contradiction. If $H \notin \mathbf{W} \setminus \mathbf{J}$, then any such path is blocked by H , which contradicts the fact that $(\mathbf{Q} \cup \mathbf{J} \setminus \mathbf{M})$ is non-empty by rule 3 of do-calculus.

iii. $N \in De(T_H) \setminus T_H$: If $N \in De(T_H) \setminus T_H$ and $H \in \mathbf{W} \setminus \mathbf{J}$, then we can obtain a d-connecting path from S to Y by concatenating $Y \rightarrow H$ with the bidirected path as outlined in the observation, which is a contradiction. If $N \in De(T_H) \setminus T_H$ and $H \notin \mathbf{W} \setminus \mathbf{J}$, then the such path is blocked by H , which contradict to the fact that $(\mathbf{Q} \cup \mathbf{J} \setminus \mathbf{M})$ is non-empty by rule 3 of do-calculus.

For the case where the d-connecting path ends at a bidirected neighbor Z of Y that is not a child of Y , without loss of generality, we also consider three cases: *i.* N is in $T_{Y \cup Z}$ or *ii.* $N \in An(T_{Y \cup Z}) \setminus T_{Y \cup Z}$ or *iii.* $N \in De(T_{Y \cup Z})$.

i. $N \in T_{Y \cup Z}$: If N is in $T_{Y \cup Z}$, we can construct two \mathbf{R} -rooted C-forests $F = T_{Y \cup Z}, F' = \{Y, Z\}$ such that $F \cap N \neq \emptyset, F' \cap N = \emptyset$, where $\mathbf{R} = \{Y, Z\}$, which is the proper subset of $An(Y \cup (\mathbf{W} \setminus \mathbf{J}))$ such that there exists a hedge for $P(Y|do((\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}), \mathbf{W} \setminus \mathbf{J})$ by the characterization of generalized hedge condition. By Theorem 3.2, $P(Y|do((\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}), \mathbf{W} \setminus \mathbf{J})$ is unidentifiable in G , which is a contradiction.

For the next two cases, we will make use of this observation: observe that any node U in $T_{Y \cup Z}$, there must be a bidirected path from U to Z and Y that only goes through nodes in $T_{Y \cup Z}$. By definition of $T_{Y \cup Z}$, any node along this bidirected path must be an ancestor of Z or Y . Therefore, there is a d-connecting path from U to Z or Y that starts with an arrow into U . Note that further conditioning cannot break this path since the path only consists of colliders.

ii. $N \in An(T_{Y \cup Z}) \setminus T_{Y \cup Z}$: If N is in $An(T_{Y \cup Z}) \setminus T_{Y \cup Z}$ and $Z \in \mathbf{W} \setminus \mathbf{J}$, then $(Y \not\perp\!\!\!\perp S | \mathbf{W} \setminus \mathbf{J})_{G_{\mathbf{Q} \cup \mathbf{J} \setminus \mathbf{M}}}$. It is because $S \in Pa(T_{Y \cup Z})$ such that we can construct a d-connecting path from S to Y by concatenating $Y \leftrightarrow Z$ with the bidirected path as outlined in the observation. Also, no member in $(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}$ can be in $T_{Y \cup Z}$ due to unidentifiability of the query as shown previously. If $Z \notin \mathbf{W} \setminus \mathbf{J}$, then any such path is blocked by Z , which contradicts the fact that $(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}$ is non-empty by rule 3 of do-calculus

iii. $N \in De(T_{Y \cup Z}) \setminus T_{Y \cup Z}$: If $N \in De(T_{Y \cup Z}) \setminus T_{Y \cup Z}$ and $Z \in \mathbf{W} \setminus \mathbf{J}$, then we can obtain a d-connecting path from S to Y by concatenating $Y \leftrightarrow Z$ with the bidirected path as outlined in the observation. If $N \in De(T_{Y \cup Z}) \setminus T_{Y \cup Z}$ and $Z \notin \mathbf{W} \setminus \mathbf{J}$, then such path is blocked by Z , which contradict to the fact that $(\mathbf{Q} \cup \mathbf{J}) \setminus \mathbf{M}$ is non-empty by rule 3 of do-calculus.

Therefore, we reach a contradiction to the case where $\mathbf{Q} \neq \emptyset$ and $\mathbf{W} \neq \emptyset$.

Thus, there exists no graph surgery estimator. By contraposition, if there exists a graph surgery estimator, **ID4IP** (Algorithm 3) outputs a graph surgery estimator. \square

A.8 PROOF OF THEOREM 3.12

Theorem A.9. (Soundness of Algorithm 3:ID4IP) *When Algorithm 3:ID4IP returns an estimator, it is a graph surgery estimator with respect to the given target and the selection variable in G .*

Proof. For line 5- 6 in Algorithm 3, the soundness follows Theorem 3.7 where we describe how selection variable being a parent of T_y implies there is no graph surgery estimator. Then, the soundness of line 7 will follow Corollary 3.9 which shows how **ID4IP** can get graph surgery estimators by utilizing the parents of the MACS of Y .

In addition, Theorem 3.8 ensures the correctness of line 8 which utilizes the parents of the MACS of some children of the target that are not selection variables. For line 10, the soundness follows Theorem 3.10 which shows how **ID4IP** can find graph surgery estimators if any by utilizing the parents of the MACS of the bidirected neighbors of Y . Lastly, line 13 follows Theorem 3.11 which guarantees **ID4IP** to find at least one graph surgery estimator or show that there is no graph surgery estimator. \square

A.9 PROOF OF THEOREM 4.1

In this proof, we will make use of the following theorems for the proof.

Theorem A.10. (Bayes-ball Complexity) [2] *Given a causal graph $G = (\mathbf{V}, \mathbf{E})$, the time complexity of Bayes-ball algorithm is $O(|\mathbf{V}| + |\mathbf{E}_{\mathbf{V}}|)$, where $\mathbf{E}_{\mathbf{V}}$ are the edges incident to the nodes marked during the algorithm. In the worst case, it is linear time in the size of the graph.*

Theorem A.11. [5] *Find-MACS-on-set(G, Y) outputs the MACS of Y in polynomial time in the size of graph.*

Proof. Let $|Ch(S)| = C$. the Graph Surgery Estimator algorithm first finds all supersets of $Ch(S)$. Getting all supersets of $Ch(S)$ takes the complexity of $O(2^{|\mathbf{V}| - C})$. Then, for each superset \mathbf{M} , the Graph Surgery Estimator algorithm finds the power set of $\mathbf{V} \setminus (\mathbf{M} \cup Y)$, which takes $O(2^{|\mathbf{V}| - (|\mathbf{M}| + |Y|)})$. Asymptotically, the complexity of finding power set for each superset becomes $O(2^{|\mathbf{V}| - (C+1)})$ as $Ch(S)$ is the smallest superset of $Ch(S)$. Then, for each member \mathbf{Q} of each power set, two major operations attribute to the complexity of the algorithm are:

1. Using a for-loop to search through each member in \mathbf{Q} . Then, it checks for d-separation condition within each loop, resulting in the complexity of $O(|\mathbf{Q}| \times (|\mathbf{V}| + |\mathbf{E}_{\mathbf{V}}|))$ with the use of Bayes-ball algorithm [2].
2. Calling **ID** algorithm for checking the identifiability of the returned unconditional query, which takes the complexity of $O(B)$

\square

Theorem A.12. (GSE Complexity) Let $|Ch(S)| = C$, $\mathbf{M} = Ch(S)$, $\mathbf{Q} = \mathbf{V} \setminus (\mathbf{M} \cup Y)$. Given a causal graph $G = (\mathbf{V}, \mathbf{E})$ and disjoint variables $\mathbf{X}, \mathbf{Y} \subset V$, the time complexity of Graph Surgery Estimator (GSE) (Algorithm 5 in Section B.4) is: $O(2^{2(|\mathbf{V}|-C)-1} \times B)$, where B represents the time complexity of **ID** algorithm.

Proof. From Theorem A.11 and A.10, we can derive the the complexity of the Graph Surgery Estimator as $O(2^{|\mathbf{V}|-C+|\mathbf{V} \setminus (\mathbf{M} \cup Y)|} \times |\mathbf{Q}| \times (|\mathbf{V}| + |\mathbf{E}_V|) \times B)$, which we can simplify to $O(2^{2(|\mathbf{V}|-C)-1} \times B)$. \square

A.10 PROOF OF THEOREM 4.3

Theorem A.13. (ID4IP Complexity) Given a causal graph $G = (\mathbf{V}, E)$ and disjoint variables $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$, the complexity of **ID4IP** (Algorithm 3) is $O((|C(Y) \cap Nbr(Y)| \setminus (T_Y \cup T_C)| + |Ch(Y)| + 1)K + (|T_Y| - 1 + |T_J| + |T'_J| - |\mathcal{H}'| - |\mathcal{H}|)B)$, where K represents the time complexity of **Find-MACS-on-set** and B represents the time complexity of **ID** algorithm, T_Y be the MACS of Y in G , T_H be the MACS of a child H of Y in G , and $T_C := \bigcup_{H \in Ch(Y)} T_H$.

Proof. By Theorem 4.2, **Find-MACS-on-set** outputs the MACS of a set in polynomial time in the size of the graph. We let $O(K)$ be the complexity of **Find-MACS-on-set** so that line 4 takes $O(K)$. Let $O(B)$ be the time complexity of **ID** algorithm. At line 7 of **ID4IP**, it takes $O(|T_Y| - 1)$ to search through the sets \mathbf{W} and each time it calls on **ID** algorithm so that each **Greedy-Eval** takes $O((|T_Y| - 1)B)$. The line 8 takes $O(|Ch(Y)|K + (|T_J| - |\mathcal{H}|)B)$ because we call on **Find-MACS-on-set** $|Ch(Y)|$ many times and each time **Find-MACS-on-set** takes $O(K)$. Now we have, $\mathcal{H} := \{H : H \in Ch(Y), Pa(T_H) \not\supseteq S\}$ and $T_J := \bigcup_{H \in \mathbf{K}} T_H$ for any $\mathbf{K} \subseteq \mathcal{H}$, where T_H is the MACS with respect to the variable H . Therefore, after finding the MACS of each child of Y , we use **Greedy-Eval**, which calls on **ID** algorithm $|T_J| - |\mathcal{H}|$ times.

Line 10, similar to line 8, it finds the MACS of each bidirected neighbor of Y that is not child nor parent of Y , which results in $O((|C(Y) \cap Nbr(Y)| \setminus (T_Y \cup T_C)|K)$. Here, we have, $\mathcal{H}' := \{H' : H' \in (C(Y) \cap Nbr(Y)) \setminus (T_Y \cup T_C), Pa(T_{H'}) \not\supseteq S\}$ and $T'_J := \bigcup_{H' \in \mathbf{K}'} T_{H'}$ for any $\mathbf{K}' \subseteq \mathcal{H}'$, where $T_{H'}$ is the MACS with respect to the variable H' . Then, we will use **Greedy-Eval**, which calls on **ID** algorithm $|T'_J| - |\mathcal{H}'|$ times, where $T'_J \neq T_J$ and $\mathcal{H}' \neq \mathcal{H}$. Therefore, line 10 takes $O((|C(Y) \cap Nbr(Y)| \setminus (T_Y \cup T_C)|K + (|T'_J| - |\mathcal{H}'|)B)$. Therefore, **ID4IP** takes $O(K + (|T_Y| - 1)B + |Ch(Y)|K + (|T_J| - |\mathcal{H}|)B + (|C(Y) \cap Nbr(Y)| \setminus (T_Y \cup T_C)|K + (|T'_J| - |\mathcal{H}'|)B)$, which can be simplified to $O((|C(Y) \cap Nbr(Y)| \setminus (T_Y \cup T_C)| + |Ch(Y)| + 1)K + (|T_Y| - 1 + |T_J| + |T'_J| - |\mathcal{H}'| - |\mathcal{H}|)B)$ \square

A.11 PROOF OF LEMMA A.14

Lemma A.14. If there is no hedge for $P(\mathbf{Y}|\text{do}(\mathbf{X}))$, then $P(\mathbf{Y}|\text{do}(\mathbf{X}))$ is identifiable in G .

Proof. Suppose there is no hedge for $P(\mathbf{Y}|\text{do}(\mathbf{X}))$. Therefore, for any $\mathbf{R} \subseteq An(Y)_{G_{\overline{\mathbf{X}}}}$, there does not exist two \mathbf{R} -rooted C forests $F' \subset F \subseteq V$ such that $F \cap \mathbf{X} \neq \emptyset$ and $F' \cap \mathbf{X} = \emptyset$. Equivalently, it must be the case that for any $\mathbf{R} \subseteq An(Y)_{G_{\overline{\mathbf{X}}}}$ and \mathbf{R} -rooted C forests $F' \subset F \subseteq V$, either $F \cap \mathbf{X} = \emptyset$ or $F' \cap \mathbf{X} \neq \emptyset$.

We consider two cases: *i.*) there is no bidirected path from \mathbf{X} to any of its children *ii.*) there is a bidirected path from \mathbf{X} to some of its children.

Suppose there is no bidirected path from any of the nodes in \mathbf{X} to any of their children in $G_{An(\mathbf{Y})}$. By Theorem 4 of [7], the query $P(\mathbf{Y}|\text{do}(\mathbf{X}))$ is then identifiable.

Suppose there is a bidirected path from some of the nodes in \mathbf{X} to some of their children in $G_{An(\mathbf{Y})}$. For case *ii*), suppose there is a bidirected path from \mathbf{X} to some of its children in $G_{An(\mathbf{Y})}$. For any $\mathbf{R} \subseteq An(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$, either there is \mathbf{R} -rooted C-forest F' such that $F' \cap \mathbf{X} = \emptyset$ but $F \cap \mathbf{X} \neq \emptyset$ or there is a \mathbf{R} -rooted C-forest F such that $F \cap \mathbf{X} \neq \emptyset$ but $F' \cap \mathbf{X} = \emptyset$ for any \mathbf{R} -rooted C-forest F' . Given these conditions, we will show that by soundness of **ID** algorithm [4], we will have $p(\mathbf{y}|\text{do}(\mathbf{x}))$ being identifiable in G .

We will briefly describe the **ID** algorithm (Algorithm 3) here. At step 6, the **ID** algorithm takes the induced subgraph of G over $An(\mathbf{Y})$, then it partitions $G_{An(\mathbf{Y})}$ into various induced subgraphs of $G_{An(\mathbf{Y})}$ over all possible C-components at step 13 of the algorithm. Since there is a hedge for $p(\mathbf{y}|\text{do}(\mathbf{x}))$ if and only if the **ID** algorithm returns at step 18 of the algorithm by the soundness of **ID** algorithm, we will proceed by showing we will never run into step 18 of the algorithm given the conditions described in the previous paragraph.

When **ID** algorithm returns FAIL, the graph G at step 17 may not necessarily refer to the original causal graph G that has been passed into **ID**, but rather a subgraph of G after taking step 13 of the algorithm along with the other potential recursive steps 6 through 11. We use $G' \subseteq G$ in the rest of the argument for the sake of clarity.

For the sake of contradiction, assume there exists a C-component in some subgraphs of $G' \subseteq G$ such that $C(G' \setminus \mathbf{X}) = \{S\}$ and $C(G') = \{G'\}$. By definition, we can construct a **R**-rooted C-forest F' as S such that $F' \cap \mathbf{X} = \emptyset$. Now, given $C(G') = \{G'\}$ and by definition of hedge, $\mathbf{X} \notin \mathbf{R}$, we can also construct another **R**-rooted C-forest F such that $F \cap \mathbf{X} \neq \emptyset$. Then, there exists a hedge for $P(\mathbf{Y}|do(\mathbf{X}))$, which is a contradiction. Therefore, we will never run into step 5 of the **ID** algorithm. Then, **ID** will return an identifiable query. Therefore, we have that $P(\mathbf{Y}|do(\mathbf{X}))$ is identifiable in G . \square

A.12 PROOF OF THEOREM A.15

Theorem A.15. *There exists a hedge for $P(\mathbf{Y}|do(\mathbf{X}))$ if and only if $P(\mathbf{Y}|do(\mathbf{X}))$ is unidentifiable in G*

Proof. By Lemma A.14 and Theorem 4 in [3], the result follows. \square

A.13 PROOF OF THEOREM A.16

Theorem A.16. *$P(Y|do(\mathbf{X}))$ is identifiable if and only if there is no hedge for $P(Y|do(\mathbf{X}'))$ where \mathbf{X}' is the smallest subset of \mathbf{X} such that $P(Y|do(\mathbf{X}')) = P(Y|do(\mathbf{X}))$.*

Proof. (\Leftarrow) Suppose there is no hedge for $P(Y|do(\mathbf{X}'))$, where \mathbf{X}' is the smallest subset of \mathbf{X} such that $P(Y|do(\mathbf{X}')) = p(y|do(\mathbf{x}))$. Since $P(Y|do(\mathbf{X}')) = P(Y|do(\mathbf{X}))$, there also exists no hedge for $P(Y|do(\mathbf{X}))$. By Lemma A.14, we have that $P(Y|do(\mathbf{X}))$ is identifiable in G .

(\Rightarrow) Suppose there exists a hedge for $P(Y|do(\mathbf{X}'))$, where \mathbf{X}' is the smallest subset of \mathbf{X} such that $P(Y|do(\mathbf{X}')) = P(Y|do(\mathbf{X}))$. Then, by Theorem 4 in [3], we have that $P(Y|do(\mathbf{X}'))$ is unidentifiable so that $P(Y|do(\mathbf{X}))$ is also unidentifiable. By contraposition, $p(y|do(\mathbf{x}))$ is identifiable implies there is no hedge for $P(Y|do(\mathbf{X}'))$, where \mathbf{X}' is the smallest subset of \mathbf{X} such that $P(Y|do(\mathbf{X}')) = P(Y|do(\mathbf{X}))$. \square

A.14 PROOF OF LEMMA A.17

Lemma A.17. *Let F be a Y -rooted C-tree in $G = (\mathbf{V}, \mathbf{E})$. For any $K \in F \setminus \{Y\}$ such that $K \subseteq \mathbf{J} \subseteq \mathbf{V} \setminus \{Y\}$ and for any $\mathbf{W} \subseteq \mathbf{V} \setminus (\mathbf{J} \cup Y)$, $P(Y|do(\mathbf{J}), \mathbf{W})$ is unidentifiable in G .*

Proof. We will show that there exists a hedge for $p(y|do(\mathbf{j}), \mathbf{w})$ according to the definition 3.1. By Theorem 20 in [4], there exists a unique maximal set \mathbf{Z} such that $P(Y|do(\mathbf{J}), \mathbf{W}) = P(Y|do(\mathbf{J}, \mathbf{Z}), \mathbf{W} \setminus \mathbf{Z})$. Since $K \in F \setminus \{Y\}$ and $K \subseteq \mathbf{J}$, we have that $F \cap (\mathbf{J} \cup \mathbf{Z}) \neq \emptyset$. Next, we let $F' = \{Y\}$ such that $F' \cap (\mathbf{J} \cup \mathbf{Z}) = \emptyset$. By definition 3.1, there exists a hedge for $P(Y|do(\mathbf{J}), \mathbf{W})$. By Theorem 3.2, $P(Y|do(\mathbf{J}), \mathbf{W})$ is unidentifiable in G . \square

B ALGORITHMS

In this section, we provide the pseudo-codes of the algorithms that we call as sub-routines from the algorithms in the main paper.

B.1 COMPUTELOSS

- 1: **Input:** A set of targets \mathbf{Y} , an intervention set \mathbf{X}
- 2: **Output:** the value of $P(\mathbf{Y}|do(\mathbf{X}))$
- 3: $P = \mathbf{ID}(\mathbf{Y}, \mathbf{X}, G)$ {Algorithm 3}
- 4: $P_s = P / \sum_Y P$
- 5: $L = \text{Compute validation loss } l(P_s)$
- 6: **Return** L

Algorithm 1: computeLoss(\mathbf{Y}, \mathbf{X})

B.2 FIND-MACS-ON-SET

- 1: **Input:** A causal graph G , an AC-component \mathbf{Y} in G
- 2: **Output:** $T_{\mathbf{Y}}$, a subgraph of G , the maximal ancestral confounded set for \mathbf{Y} in G .
- 3: **if** $\exists X \notin An(\mathbf{Y})_G$ **then**
- 4: **Return** Find-MACS-on-set($G_{An(\mathbf{Y})}, \mathbf{Y}$)
- 5: **if** $\exists Y \in \mathbf{Y}, \exists X \notin C(Y)_G$ **then**
- 6: **Return** Find-MACS-on-set($G_{C(Y)}, \mathbf{Y}$)
- 7: **else**
- 8: **Return** G

Algorithm 2: Find-MACS-on-set(G, \mathbf{Y})[5]

B.3 ID ALGORITHM

- 1: **Input:** a set of target variables \mathbf{Y} , a set of random variables for intervention \mathbf{X} , a probability distribution P , a causal graph G
- 2: **Output:** Expression for $P(Y|do(\mathbf{X}))$ in terms of P or **FAIL**(F, F')
- 3: **if** $\mathbf{X} = \emptyset$ **then**
- 4: **Return** $\sum_{\mathbf{V} \setminus \mathbf{Y}} P(\mathbf{v})$
- 5: **if** $\mathbf{V} \setminus An(\mathbf{Y})_G \neq \emptyset$ **then**
- 6: **Return** ID($\mathbf{Y}, \mathbf{X} \cap An(\mathbf{Y})_G, \sum_{\mathbf{V} \setminus An(\mathbf{Y})_G} P, G_{An(\mathbf{Y})}$)
- 7: Let $\mathbf{W} = (\mathbf{V} \setminus \mathbf{X}) \setminus An(\mathbf{Y})_{G_{\mathbf{X}}}$
- 8: **if** $\mathbf{W} \neq \emptyset$ **then**
- 9: **Return** ID($\mathbf{Y}, \mathbf{X} \cup \mathbf{W}, P, G$)
- 10: **if** $C(G \setminus \mathbf{X}) = \{S_1, \dots, S_k\}$ **then**
- 11: **Return** $\sum_{\mathbf{V} \setminus (\mathbf{Y} \cup \mathbf{X})} \prod_i ID(S_i, \mathbf{V} \setminus S_i, P, G)$
- 12: **if** $C(G \setminus \mathbf{X}) = \{S\}$ **then**
- 13: **if** $C(G) = \{G\}$ **then**
- 14: **Return** **FAIL**($G, G \cap S$)
- 15: **if** $S \in C(G)$ **then**
- 16: **Return** $\sum_{S \setminus \mathbf{Y}} \prod_{\{i|V_i \in S\}} P(V_i|V_{\pi}^{i-1})$
- 17: **if** $\exists S$ s.t. $S \subset S' \in C(G)$ **then**
- 18: **Return** ID($\mathbf{Y}, \mathbf{X} \cup S', \prod_{\{i|V_i \in S\}} P(V_i|V_{\pi}^{(i-1)} \setminus S), G_{S'}$)

Algorithm 3: ID($\mathbf{Y}, \mathbf{X}, P, G$) [4]

B.4 GRAPH SURGERY ESTIMATOR ALGORITHM

In this section, we present the main algorithms in [6].

- 1: **Input:** Acyclic Directed Mixed Graph (ADMG) $G = (\mathbf{V}, \mathbf{E})$, disjoint variable sets $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{V}$
- 2: **Output:** Unconditional query $P(\mathbf{Y}|do(\mathbf{X}), \mathbf{Z})$
- 3: $\mathbf{X}' = \mathbf{X}$
- 4: $\mathbf{Y}' = \mathbf{Y}$
- 5: $\mathbf{Z}' = \mathbf{Z}$
- 6: **while** $\exists Z \in \mathbf{Z}$ s.t. $(\mathbf{Y} \perp\!\!\!\perp Z|\mathbf{X}, \mathbf{Z} \setminus \{Z\})_{G_{\mathbf{X}, \mathbf{Z}}}$ **do**
- 7: $\mathbf{X}' = \mathbf{X}' \cup Z$
- 8: $\mathbf{Z}' = \mathbf{Z}' \setminus \{Z\}$
- 9: $\mathbf{Y}' = \mathbf{Y} \cup \mathbf{Z}$
- 10: **Return** \mathbf{X}', \mathbf{Y}' of unconditional query $P(\mathbf{Y}'|do(\mathbf{X}'))$

Algorithm 4: Unconditional Query: UQ($\mathbf{X}, \mathbf{Y}, \mathbf{Z}; G$)[6]

```

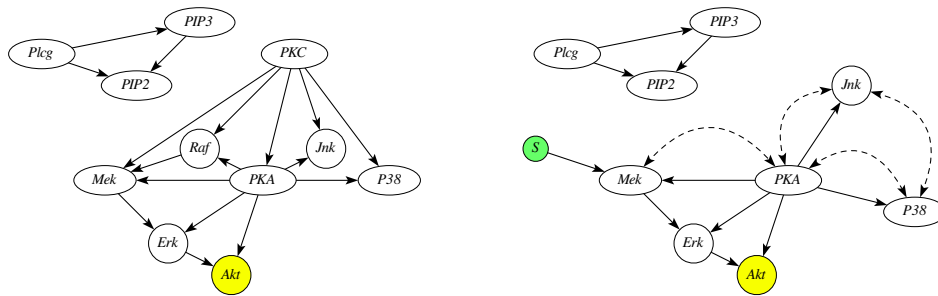
1: Input: ADMG  $G$ , mutable variables  $\mathbf{M}$ , target  $T$ 
2: Output: Expression for the surgery estimator or FAIL if there is no stable estimator.
3:  $S_{ID} = \emptyset$ 
4:  $Loss = \emptyset$ 
5: for  $\mathbf{Z} \in \mathcal{P}(\mathbf{V} \setminus (\mathbf{M} \cup \{T\}))$  do
6:   if  $T \notin \mathbf{M}$  then
7:      $\mathbf{X}, \mathbf{Y} = UQ(\mathbf{M}, \{T\}, \mathbf{Z}, G)$ 
8:     try
9:        $P = ID(\mathbf{X}, \mathbf{Y}, G)$ 
10:       $P_s = P / \sum_Y P$ 
11:      Compute the validation loss  $l(P_s)$ 
12:       $S_{ID}.append(P_s); Loss.append(l(P_s))$ 
13:     catch
14:       continue
15:    $\mathbf{X}, \mathbf{Y} = UQ(\mathbf{M}, \{T\}, \mathbf{Z}; G_T)$ 
16:    $\mathbf{X} = \mathbf{X} \cup \{T\}$ 
17:    $\mathbf{Y} = \mathbf{Y} \setminus \{T\}$ 
18:   if  $\mathbf{Y} \cap (T \cup Ch(T)) = \emptyset$  then
19:     continue
20:   try
21:      $P = ID(\mathbf{X}, \mathbf{Y}, G)$ 
22:      $P_s = P / \sum_Y P$ 
23:     Compute the validation loss  $l(P_s)$ 
24:      $S_{ID}.append(P_s); Loss.append(l(P_s))$ 
25:   catch
26:     continue
27: if  $S_{ID} = \emptyset$  then
28:   Return FAIL
29: Return  $P_s \in S_{ID}$  with lowest corresponding  $Loss$ 

```

Algorithm 5: Graph Surgery Estimator $(G, \mathbf{M}, Y)[6]$

C SEMI-SYNTHETIC CAUSAL GRAPHS

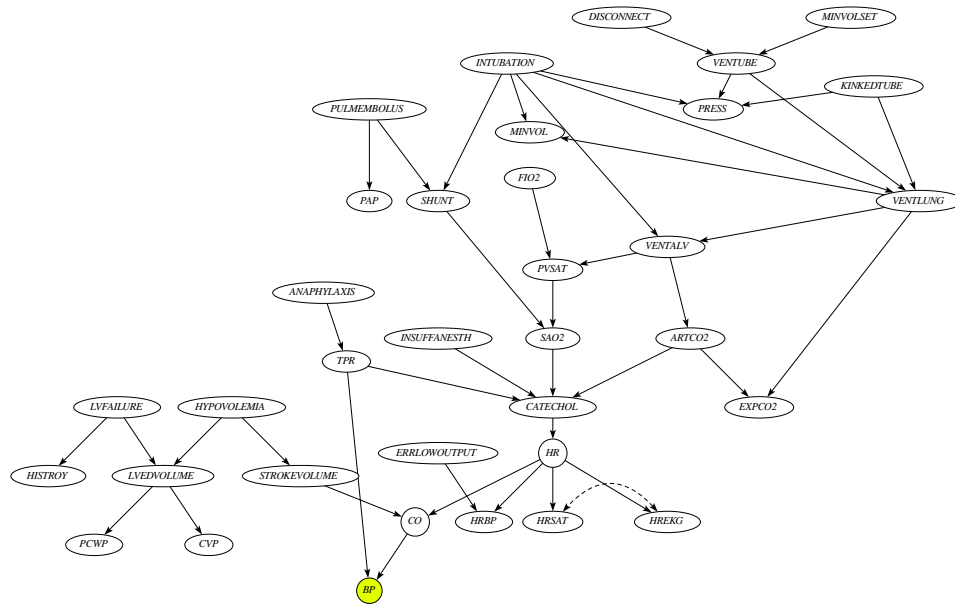
C.1 SACHS CAUSAL GRAPH



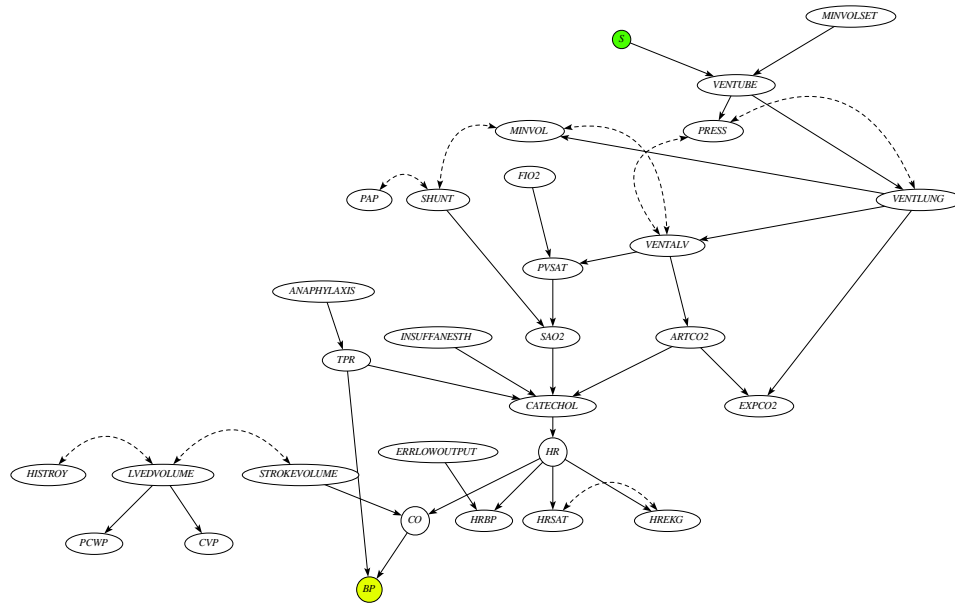
(a) Original Sachs causal graph with the target variable (b) Modified Sachs causal graph with the selection variable (green)

Figure 1: Semi-synthetic experimental results

C.2 ALARM CAUSAL GRAPH



(a) Original Alarm causal graph with the target variable (yellow)



(b) Modified Alarm causal graph with the selection variable (green)

Figure 2: Semi-synthetic causal graph: Alarm

REFERENCES

- [1] Thomas Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

- [2] Ross D Shachter. Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). *arXiv preprint arXiv:1301.7412*, 2013.
- [3] Ilya Shpitser and Judea Pearl. *Identification of joint interventional distributions in recursive semi-Markovian causal models*. eScholarship, University of California, 2006.
- [4] Ilya Shpitser and Judea Pearl. Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research*, 9:1941–1979, 2008.
- [5] Ilya Shpitser and Judea Pearl. *Dormant independence*. AAAI Press, 2008.
- [6] Adarsh Subbaswamy, Peter Schulam, and Suchi Saria. Preventing failures due to dataset shift: Learning predictive models that transport. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3118–3127. PMLR, 2019.
- [7] Jin Tian and Judea Pearl. *A general identification condition for causal effects*. eScholarship, University of California, 2002.