On the Complexity of Counterfactual Reasoning

Yunqiu Han , **Yizuo Chen** , **Adnan Darwiche**
University of California, Los Angeles
yunqiu21@g.ucla.edu, yizuo.chen@ucla.edu, darwiche@cs.ucla.edu

Abstract

We study the computational complexity of counterfactual reasoning in relation to the complexity of associational and interventional reasoning on structural causal models (SCMs). We show that counterfactual reasoning is no harder than associational or interventional reasoning on fully specified SCMs in the context of two computational frameworks. The first framework is based on the notion of *treewidth* and includes the classical variable elimination and jointree algorithms. The second framework is based on the more recent and refined notion of *causal treewidth* which is directed towards models with functional dependencies such as SCMs. Our results are constructive and based on bounding the (causal) treewidth of twin networks—used in standard counterfactual reasoning that contemplates two worlds, real and imaginary—to the (causal) treewidth of the underlying SCM structure. In particular, we show that the latter (causal) treewidth is no more than twice the former plus one. Hence, if associational or interventional reasoning is tractable on a fully specified SCM then counterfactual reasoning is tractable too. We extend our results to general counterfactual reasoning that requires contemplating more than two worlds and discuss applications of our results to counterfactual reasoning with partially specified SCMs that are coupled with data. We finally present empirical results that measure the gap between the complexities of counterfactual reasoning and associational/interventional reasoning on random SCMs.

1 Introduction

A theory of causality has emerged over the last few decades based on two parallel hierarchies, an *information hierarchy* and a *reasoning hierarchy,* often called the *causal hierarchy* [Pearl and Mackenzie, 2018]. On the reasoning side, this theory has crystallized three levels of reasoning with increased sophistication and proximity to human reasoning: associational, interventional and counterfactual, which are exemplified by the following canonical probabilities. *Associational* $Pr(y|x)$: probability of y given that x was observed. Example: probability that a patient has a flu given they have a fever. *Interventional* $Pr(y_x)$: probability of y given that x was established by an intervention. This is different from $Pr(y|x)$. Example: seeing the barometer fall tells us about the weather but moving the barometer needle won't bring rain. *Counterfactual* $Pr(y_x|\bar{y}, \bar{x})$: probability of y if we were to establish x by an intervention given that neither x nor y are true. Example: probability that a patient who did not take a vaccine and died would have recovered had they been vaccinated. On the information side, these forms of reasoning were shown to require different levels of knowledge, encoded as (1) associational models, (2) causal models and (3) functional (mechanistic) models, respectively, with each class of models containing more information than the preceding one. In the framework of probabilistic graphical models [Koller and Friedman, 2009], this information is encoded by (1) Bayesian networks [Darwiche, 2009; Pearl, 1988], (2) causal Bayesian networks [Pearl, 2009; Peters *et al.*, 2017; Spirtes *et al.*, 2000], and (3) functional Bayesian networks [Balke and Pearl, 1995; Pearl, 2009].

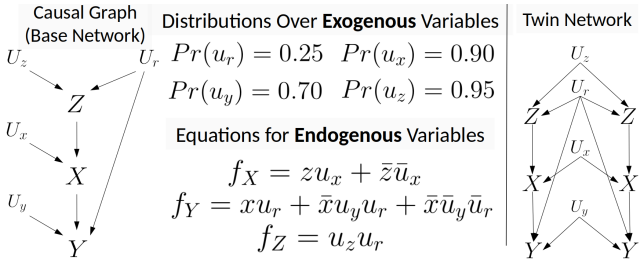


Figure 1: A structural causal model from [Bareinboim *et al.*, 2021] and its twin network. Endogenous variables represent treatment (X), the outcome of (Y), and the presence of (Z), hypertension. Exogenous variables represent natural resistance to disease (U_r) and sources of variation affecting endogenous variables (U_x, U_y, U_z).

Counterfactual reasoning has received much interest as it inspires both introspection and contemplating scenarios that have not been seen before, and is therefore viewed by many as a hallmark of human intelligence. Figure 1 depicts a functional Bayesian network, also known as a *structural causal model (SCM)* [Galles and Pearl, 1998; Halpern, 2000], which can be used to answer counterfactual queries. Variables without causes are called *exogenous* or *root* and variables with

causes are called *endogenous* or *internal.* The only uncertainty in SCMs concerns the states of exogenous variables and this uncertainty is quantified using distributions over these variables. Endogenous variables are assumed to be *functional:* they are functionally determined by their causes where the functional relationships, also known as *causal mechanisms,* are specified by structural equations.[1] These equations and the distributions over exogenous variables define the *parameters* of the causal graph, leading to a fully specified SCM which can be used to evaluate associational, interventional and counterfactual queries. For example, the SCM in Figure 1 has enough information to evaluate the counterfactual query $Pr(y_x|\bar{x},\bar{y})$: the probability that a patient who did not take the treatment and died would have been alive had they been given the treatment (2.17%). A causal Bayesian network contains less information than a functional one (SCM) as it does not require endogenous variables to be functional, but it is sufficient to compute associational and interventional probabilities. A Bayesian network contains even less information as it does not require network edges to have a causal interpretation, only that the conditional independences encoded by the network are correct, so it can only compute associational probabilities.

All three forms of reasoning (and models) involve a directed acyclic graph (DAG) which we call the *base network;* see left of Figure 1. Associational and interventional reasoning can be implemented by applying classical inference algorithms to the base network. The time complexity can be bounded by $n \cdot \exp(w)$, where n is the number of network nodes and w is its treewidth (a graph-theoretic measure of connectivity). Counterfactual reasoning is more sophisticated and is based on a three-step process that involves abduction, intervention and prediction [Balke and Pearl, 1994b]. When contemplating two worlds, this process can be implemented by applying classical inference algorithms to a *twin network* [Balke and Pearl, 1994b], obtained by duplicating endogenous nodes in the base network; see right of Figure 1. To compute the counterfactual query $Pr(y_x|\bar{y},\bar{x})$, one asserts $\bar{y},\bar{x}$ as an observation on one side of the twin network (real world) and computes the interventional query $Pr(y_x)$ on the other side of the network (imaginary world). The time complexity can be bounded by $n^t \cdot \exp(w^t)$, where n^t is the number of nodes in the twin network and w^t is its treewidth. A recent result provides a much tighter bound using the notion of *causal treewidth* [Chen and Darwiche, 2022; Darwiche, 2021], which is no greater than treewidth but applies only when certain nodes in the base network are functional — in SCMs every endogenous node is functional.

One would expect the more sophisticated counterfactual reasoning with twin networks to be more expensive than associational/interventional reasoning with base networks since the former networks are larger and have more complex topologies. But the question is: How much more expensive? For example, can counterfactual reasoning be intractable on a twin network when associational/interventional reasoning is tractable on its base network? We address this question in the context of reasoning algorithms whose complexity is exponential only in the (causal) treewidth, such as the jointree algorithm [Lauritzen and Spiegelhalter, 1988], the variable elimination algorithm [Zhang and Poole, 1994; Dechter, 1996] and circuit-based algorithms [Darwiche, 2003; Darwiche, 2022]. In particular, we show in Sections 3 & 4 that the (causal) treewidth of a twin network is at most twice the (causal) treewidth of its base network plus one. Hence, the complexity of counterfactual reasoning on fully specified SCMs is no more than quadratic in the complexity of associational and interventional reasoning, so the former must be tractable if the latter is tractable. We extend our results in Section 5 to counterfactual reasoning that requires contemplating more than two worlds, where we also discuss a class of applications that require this type of reasoning and for which fully specified SCMs can be readily available. Our results apply directly to counterfactual reasoning on fully specified SCMs but we also discuss in Section 6 how they can sometimes be used in the context of counterfactual reasoning on data and a partially specified SCM. We finally present empirical results in Section 7 which reveal that, on average, the complexity gap between counterfactual and associational/interventional reasoning on fully specified SCMs can be smaller than what our worst-case bounds may suggest.

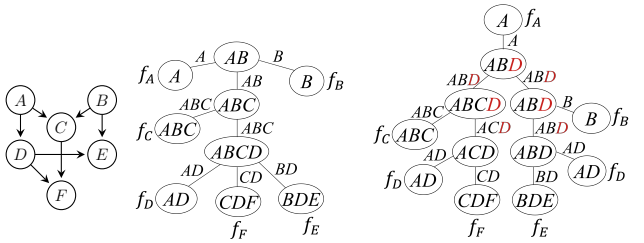
2 Technical Preliminaries

We next review the notions of treewidth [Robertson and Seymour, 1986] and causal treewidth [Chen and Darwiche, 2022; Darwiche, 2021; Darwiche, 2020] which we use to characterize the computational complexity of counterfactual reasoning on fully specified SCMs. We also review the notions of elimination orders, jointrees and thinned jointrees which are the basis for defining (causal) treewidth and act as data structures that characterize the computational complexity of various reasoning algorithms. We use these notions extensively when stating and proving our results (proofs of all results are in the Appendix found in [Han *et al.*, 2023]). We assume all variables are discrete. A variable is denoted by an uppercase letter (e.g. X) and its values by a lowercase letter (e.g. x). A set of variables is denoted by a bold uppercase letter (e.g. $\mathbf{X}$) and its instantiations by a bold lowercase letter (e.g. $\mathbf{x}$).

2.1 Elimination Orders and Treewidth

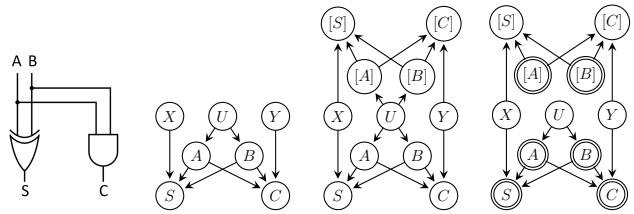
These are total orders of the network variables which drive, and characterize the complexity of, the classical variable elimination algorithm when computing associational, interventional and counterfactual queries. Consider a DAG G where every node represents a variable. An *elimination order* π for G is a total ordering of the variables in G, where $\pi(i)$ is the i^{th} variable in the order, starting from $i = 1$. An elimination order defines an elimination process on the moral graph of DAG G which is used to define the treewidth of G. The *moral graph* G_m is obtained from G by adding an undirected edge between every pair of common parents and then removing directions from all directed edges. When we *eliminate* variable $\pi(i)$ from G_m, we connect every pair of neighbors of $\pi(i)$ in G_m and remove $\pi(i)$ from G_m. This elimination process induces a *cluster sequence* $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_n$,

[1] These equations can also be specified using conditional probability tables (CPTs) that are normally used in Bayesian networks, but the CPTs will contain only deterministic distributions.



(a) DAG (b) jointree (width 3) (c) thinned jointree (width 2)

Figure 2: A family f appears next to a jointree node i iff f is hosted by i ($i \in \mathcal{H}(f)$). D is functional and red variables are thinned.



(a) half adder (b) base network (c) twin network (d) mutilated

Figure 3: Internal nodes in the base network (Figure (b)) are functional. Double-circled nodes have evidence.

where $\mathbf{C}_i$ is $\pi(i)$ and its neighbors in G_m just before eliminating $\pi(i)$. The *width* of an elimination order is the size of its largest induced cluster minus 1. The *treewidth* for DAG G is the minimum width of any elimination order for G. The variable elimination algorithm computes queries in $O(n \cdot \exp(w))$ time where n is the number of nodes in the (base or twin) network and w is the width of a corresponding elimination order. Elimination orders are usually constructed using heuristics that aim to minimize their width. We use the popular *min-fill* heuristic [Kjaerulff, 1990] in our experiments while noting that more effective heuristics may exist as shown in [Kjærulff, 1994; Larrañaga *et al.*, 1997].

2.2 Jointrees and Treewidth

These are data structures that drive, and characterize the complexity of, the classical jointree algorithm; see Figure 2b. Let the *family* of variable X in DAG G be the set f_X containing X and its parents in G. A *jointree* for DAG G is a pair $\langle \mathcal{T}, \mathcal{H} \rangle$ where $\mathcal{T}$ is a tree and $\mathcal{H}$ is a function that maps each family f of G into nodes $\mathcal{H}(f)$ in $\mathcal{T}$ called the *hosts* of family f. The requirements are: only nodes with a single neighbor (called *leaves*) can be hosts; each leaf node hosts exactly one family; and each family must be hosted by at least one node.[2] This induces a *cluster* $\mathbf{C}_i$ for each jointree node i and a *separator* $\mathbf{S}_{ij}$ for each jointree edge (i, j) which are defined as follows. Separator $\mathbf{S}_{ij}$ is the set of variables hosted at both sides of edge (i, j). If jointree node i is a leaf then cluster $\mathbf{C}_i$ is the family hosted by i; otherwise, $\mathbf{C}_i$ is the union of separators adjacent to node i. The *width* of a jointree is the size of its largest cluster minus 1. The minimum width attained by any jointree for G corresponds to the treewidth of G. The jointree algorithm computes queries in $O(n \cdot \exp(w))$ time where n is the number of nodes and w is the width of a corresponding jointree. Jointrees are usually constructed from elimination orders, and there are polytime, width-preserving transformations between elimination orders and jointrees; see [Darwiche, 2009, Ch 9] for details.

2.3 Thinned Jointrees and Causal Treewidth

To *thin* a jointree is to remove some variables from its separators (and hence clusters, which are defined in terms of separators); see Figure 2c. Thinning can reduce the jointree width quite significantly, leading to exponential savings in reasoning time. Thinning is possible only when some variables in the network are functional, even without knowing the specific functional relationships (i.e., structural equations). The *causal treewidth* is the minimum width for any thinned jointree. Causal treewidth is no greater than treewidth and the former can be bounded when the latter is not. While this notion can be applied broadly as in [Darwiche, 2020], it is particularly relevant to counterfactual reasoning since every internal node in an SCM is functional so the causal treewidth for SCMs can be much smaller than their treewidth. There are alternate definitions of thinned jointrees. The next definition is based on thinning rules [Chen and Darwiche, 2022].

A thinning rule removes a variable from a separator under certain conditions. There are two thinning rules which apply only to functional variables. The first rule removes variable X from a separator $\mathbf{S}_{ij}$ if edge (i, j) is on the path between two leaf nodes that host the family of X and every separator on that path contains X. The second rule removes variable X from a separator $\mathbf{S}_{ij}$ if no other separator $\mathbf{S}_{ik}$ contains X, or no other separator $\mathbf{S}_{kj}$ contains X. A thinned jointree is obtained by applying these rules to exhaustion. Figure 2 depicts an optimal, classical jointree and a thinned jointree for the same DAG (the latter has smaller width).

The effectiveness of thinning rules depends on the number of jointree nodes that host a family f, $|\mathcal{H}(f)|$, and the location of these nodes in the jointree. One can enable more thinnings by increasing the number of jointree nodes that host each family f. This process is called *replication* where $|\mathcal{H}(f)|$ is called the number of *replicas* for family f. Replication comes at the expense of increasing the number of jointree nodes so the definition of causal treewidth limits this growth by requiring the jointree size to be a polynomial in the number of nodes in the underlying DAG; see [Chen and Darwiche, 2022] for details.[3]

3 The Treewidth of Twin Networks

Consider Figure 3a which depicts a 2-bit half adder. Suppose the binary inputs A and B are randomly sampled from some

[2] The standard definition of jointrees allows any node to be a host of any number of families. Our definition facilitates the upcoming treatment and does not preclude optimal jointrees.

[3] Thinning rules will not trigger if families are not replicated ($|\mathcal{H}(f)| = 1$ for all f). Replication usually increases the width of a jointree from w to w_r with the goal of having thinning rules reduce width w_r to width $w_t < w \leq w_r$. The replication strategy may sometimes not be effective on certain networks, leading to $w < w_t \leq w_r$. We use the strategy in [Chen and Darwiche, 2022] which exhibits this behavior on certain networks.

distribution and the gates may not be functioning properly. This circuit can be modeled using the network in Figure 3b. Variables A, B, S, C represent the inputs and outputs of the circuit; X, Y represent the health of the XOR gate and the AND gate; and U represents an unknown external random sampler that decides the state of inputs A and B. Suppose that currently input A is high, input B is low, yet both outputs C and S are low which is an abnormal circuit behavior. We wish to know whether the half adder would still behave correctly when we turn both inputs A and B on. This question can be formulated using the following counterfactual query: $Pr((c, \bar{s})_{a,b}|a, \bar{b}, \bar{c}, \bar{s})$. This query can be answered using a twin network as shown in Figure 3c, where each non-root variable V has a duplicate $[V]$. The current evidence $a, \bar{b}, \bar{c}, \bar{s}$ is asserted on the variables A, B, C, S representing the real world and the interventional query $Pr((c, \bar{s})_{a,b})$ is computed on the duplicate variables $[A], [B], [C], [S]$ representing the imaginary world. This is done by removing the edges incoming into the intervened upon variables $[A], [B]$, asserting evidence $[a], [b]$ and finally computing the probability of $[c], [\bar{s}]$ as shown in Figure 3d; see [Pearl, 2009] for an elaborate discussion of these steps. This basically illustrates how a counterfactual query can be computed using algorithms for associational queries, like variable elimination, but on a mutilated twin network instead of the base network.

We next show that the treewidth of a twin network is at most twice the treewidth of its base network plus one, which allows us to relate the complexities of assocational, interventional and counterfactual reasoning on fully specified SCMs. We first recall the definition of twin networks as proposed by [Balke and Pearl, 1994b].

Definition 1. *Given a base network G, its twin network G^t is constructed as follows. For each internal variable X in G, add a new variable labeled $[X]$. For each parent P of X, if P is an internal variable, make $[P]$ a parent of $[X]$; otherwise, make P a parent of $[X]$. We will call X a base variable and $[X]$ a duplicate variable.*

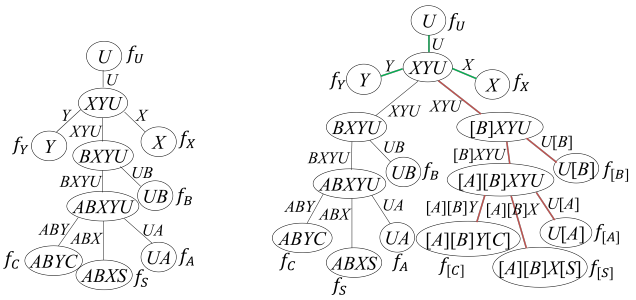
For convenience, we use $[U] = U$ when U is root. For variables $\mathbf{X}$, we use $[\mathbf{X}]$ to denote $\{[X]|X \in \mathbf{X}\}$. Figure 3c depicts the twin network for the base network in Figure 3b.

3.1 Twin Elimination Orders

Our result on the treewidth of twin networks is based on converting every elimination order for the base network into an elimination order for its twin network while providing a guarantee on the width of the latter in terms of the width of the former. We provide a similar result for jointrees that we use when discussing the causal treewidth of twin networks.

Definition 2. *Consider an elimination order π for a base network G. The twin elimination order π^t is an elimination order for its twin network G^t constructed by replacing each non-root variable X in π by $X, [X]$.*

Consider the base network in Figure 3b and its elimination order $\pi = A, B, X, Y, S, C, U$. The twin elimination order will be $\pi^t = A, [A], B, [B], X, Y, S, [S], C, [C], U$. Recall that eliminating variables $\pi(i), \ldots, \pi(n)$ from a base network G induces a cluster sequence $\mathbf{C}_1, \ldots, \mathbf{C}_n$. We use



(a) base jointree (b) twin jointree using Algorithm 1

Figure 4: A family f appears next to a jointree node i iff the family is hosted by that node ($i \in \mathcal{H}(f)$).

$\mathbf{C}(X)$ to denote the cluster of eliminated variable X. Similarly, eliminating variables from a twin network G^t induces a cluster sequence and we use $\mathbf{C}^t(X)$ to denote the cluster of eliminated variable X and $\mathbf{C}^t([X])$ to denote the cluster of its eliminated duplicate $[X]$.

Theorem 1. *Suppose we are eliminating variables from base network G using an elimination order π and eliminating variables from its twin network G^t using the twin elimination order π^t. For every variable X in G, we have $\mathbf{C}^t(X) \subseteq \mathbf{C}(X) \cup [\mathbf{C}(X)]$ and $\mathbf{C}^t([X]) \subseteq \mathbf{C}(X) \cup [\mathbf{C}(X)]$.*

This theorem has two key corollaries. The first relates the widths of an elimination order and its twin elimination order.

Corollary 1. *Let w be the width of elimination order π for base network G and let w^t be the width of twin elimination order π^t for twin network G^t. We then have $w^t \leq 2w + 1$.*

The above bound is tight as shown in the Appendix. The next corollary gives us our first major result.

Corollary 2. *If w is the treewidth of base network G and w^t is the treewidth of its twin network G^t, then $w^t \leq 2w + 1$.*

3.2 Twin Jointrees

We will now provide a similar result for jointrees. That is, we will show how to convert a jointree $\langle \mathcal{T}, \mathcal{H} \rangle$ for a base network G into a jointree $\langle \mathcal{T}^t, \mathcal{H}^t \rangle$ for its twin network G^t while providing a guarantee on the width/size of the twin jointree in terms of the width/size of the base jointree. This may seem like a redundant result given Corollary 1 but the provided conversion will actually be critical for our later result on bounding the causal treewidth of twin networks. It can also be significantly more efficient than constructing a jointree by operating on the (larger) twin network.

Our conversion process operates on a *jointree* after directing its edges away from some node r, call it a *root*. This defines a single parent for each jointree node $i \neq r$, which is the neighbor of i closest to root r, with all other neighbors of i being its children. These parent-child relationships are invariant when running the algorithm. We also use a subroutine for *duplicating the jointree nodes rooted at some node i*. This subroutine duplicates node i and its descendant while also duplicating the edges connecting these nodes. If a duplicated node j hosts a family f, this subroutine will make $[j]$ host the duplicate family $[f]$ (so $j \in \mathcal{H}(f)$ iff $[j] \in \mathcal{H}([f])$).

Algorithm 1 Jointree to Twin Jointree

```
1: procedure MAKE-TWIN-JOINTREE(⟨T, H⟩, r, p)
2:     Σ ← leaf nodes at or below node r
3:     if nodes in Σ only host families for root variables then
4:         return
5:     if nodes in Σ only host families for internal variables then
6:         duplicate the jointree nodes rooted at node r
7:         add [r] as a child of p
8:     else
9:         for each child k of node r do
10:            MAKE-TWIN-JOINTREE(⟨T, H⟩, k, r)
```

The conversion process is given in Algorithm 1 which should be called initially with a root r that does not host a family for an internal DAG node and $p = null$. The twin jointree in Figure 4b was obtained from the base jointree in Figure 4a by this algorithm which simply adds nodes and edges to the base jointree. If an edge (i, j) in the base jointree is duplicated by Algorithm 1, we call (i, j) a *duplicated edge* and $([i], [j])$ a *duplicate edge*. Otherwise, we call (i, j) an *invariant edge*. In Figure 4b, duplicate edges are shown in red and invariant edges are shown in green. We now have the following key result on these twin jointrees.

Theorem 2. *If the input jointree to Alg. 1 has separators* $\mathbf{S}$ *and the output jointree has separators* $\mathbf{S}^t$*, then for duplicated edges* (i, j)*,* $\mathbf{S}^t_{ij} = \mathbf{S}_{ij}$*; for duplicate edges* $([i], [j])$*,* $\mathbf{S}^t_{[i][j]} = [\mathbf{S}_{ij}]$*; and for invariant edges* (i, j)*,* $\mathbf{S}^t_{ij} = \mathbf{S}_{ij} \cup [\mathbf{S}_{ij}]$*.*

One can verify that the separators in Figure 4 satisfy these properties. The following result bounds the width and size of twin jointrees generated by Algorithm 1.

Corollary 3. *Let* w *be the width of a jointree for base network* G *and let* n *be the number of jointree nodes. Calling Algorithm 1 on this jointree will generate a jointree for twin network* G^t *whose width is no greater than* $2w + 1$ *and whose number of nodes is no greater than* $2n$*.*

The above bound on width is tight as shown in the Appendix. Since treewidth can be defined in terms of jointree width, the above result leads to the same guarantee of Corollary 2 on the treewidth of twin networks. However, the main role of the construction in this section is in bounding the causal treewidth of twin networks. This is discussed next.

4 The Causal Treewidth of Twin Networks

Recall that causal treewidth is a more refined notion than treewidth as it uses more information about the network. In particular, this notion is relevant when we know that some variables in the network are functional, without needing to know the specific functions (equations) of these variables. By exploiting this information, one can construct thinned jointrees that have smaller separators and clusters compared to classical jointrees, which can lead to exponential savings in reasoning time [Chen and Darwiche, 2022; Darwiche, 2021; Darwiche, 2020]. As mentioned earlier, the causal treewidth corresponds to the minimum width of any thinned jointree. This is guaranteed to be no greater than treewidth and can be bounded when treewidth is not [Darwiche, 2021]. We next show that the causal treewidth of a twin network is also at most twice the causal treewidth of its base network plus one.

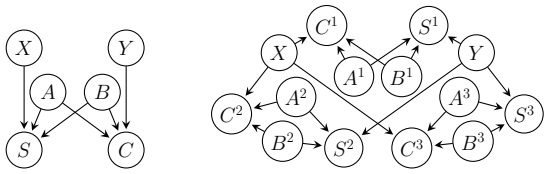


Figure 5: A base network and its 3-world network.

Theorem 3. *Consider a twin jointree constructed by Algorithm 1 from a base jointree with thinned separators* $\mathbf{S}$*. The following are valid thinned separators for this twin jointree: for duplicated edges* (i, j)*,* $\mathbf{S}^t_{ij} = \mathbf{S}_{ij}$*; for duplicate edges* $([i], [j])$*;* $\mathbf{S}^t_{[i][j]} = [\mathbf{S}_{ij}]$*; and for invariant edges* (i, j)*,* $\mathbf{S}^t_{ij} = \mathbf{S}_{ij} \cup [\mathbf{S}_{ij}]$*.*

This theorem shows that a thinned, base jointree can be easily converted into a thinned, twin jointree. This is significant for two reasons. First, this method avoids the explicit construction of thinned jointrees for twin networks which can be quite expensive computationally [Chen and Darwiche, 2022]. Second, we have the following guarantee on the width of thinned, twin jointrees constructed by Theorem 3.

Corollary 4. *Consider the thinned, base and twin jointrees in Theorem 3. If the thinned, base jointree has width* w*, then the thinned, twin jointree has width no greater than* $2w + 1$*.*

Due to space constraints, we include a thinned jointree for the base network and the corresponding thinned, twin jointree constructed by Algorithm 1 and Theorem 3 in the Appendix. We can now bound the causal treewidth of twin networks.

Corollary 5. *If* w *and* w^t *are the causal treewidths of a base network and its twin network, then* $w^t \leq 2w + 1$*.*

5 Counterfactual Reasoning Beyond Two Worlds

Standard counterfactual reasoning contemplates two worlds, one real and another imaginary, while assuming that exogenous variables correspond to causal mechanisms that govern both worlds. This motivates the notion of a twin network as it ensures that these causal mechanisms are invariant. We can think of counterfactual reasoning as a kind of *temporal reasoning* where endogenous variables can change their states over time. A more general setup arises when we allow some exogenous variables to change their states over time. For example, consider again the half adder in Figure 3a and its base network in Figure 5. Suppose we set inputs A and B to high and low and observe outputs S and C to be high and low, which is a normal behavior. We then set both inputs to low and observe that the outputs do not change, which is an abnormal behavior. We then aim to predict the state of outputs if we were to set both inputs to high. This scenario involves three time steps (worlds). Moreover, while the health of gates X and Y are invariant over time, we do not wish to make the same assumption about the inputs A and B. We can model this situation using the network in Figure 5, which is a more general type of networks that we call N-world networks.

Definition 3. *Consider a base network G and let $\mathbf{R}$ be a subset of its roots and $N \geq 1$ be an integer. The N-world network G^N of G is constructed as follows. For each variable X in G that is not in $\mathbf{R}$, replace it with N duplicates of X, labeled $X^1, X^2, \ldots, X^N$. For each parent P of X, if P is in $\mathbf{R}$, make P a parent of X^i for all $i \in 1, 2, \ldots, N$. Otherwise, make P^i a parent of X^i for all $i \in 1, 2, \ldots, N$.*

This definition corresponds to the notion of a *parallel worlds model* [Avin *et al.*, 2005] when $\mathbf{R}$ contains all roots in the base network. Moreover, twin networks fall as a special case when $N = 2$ and $\mathbf{R}$ contains all roots of the base network. We next bound the (causal) treewidth of N-world networks by the (causal) treewidth of their base networks.

Theorem 4. *If w and w^t are the (causal) treewidths of a base network and its N-world network, then $w^t \leq N(w+1) - 1$.*

The class of N-world networks is a subclass of *dynamic Bayesian networks* [Dean and Kanazawa, 1989] and is significant for a number of reasons. First, as illustrated above, it arises when reasoning about the behavior of systems consisting of function blocks (e.g., gates) [Hamscher *et al.*, 1992]. These kinds of physical systems can be easily modeled using fully specified SCMs, where the structural equations correspond to component behaviors and the distributions over exogenous variables correspond to component reliabilities; see [Darwiche, 2009, Ch 5] for a textbook discussion and [Mengshoel *et al.*, 2010] for a case study of a real-world electrical power system. More broadly, N-world networks allow counterfactual reasoning that involves conflicting observations and actions that arise in multiple worlds as in the *unit selection problem* [Li and Pearl, 2022]—for example, [Huang and Darwiche, 2023] used Theorem 4 to obtain bounds on the complexity of this problem. See also [Avin *et al.*, 2005; Shpitser and Pearl, 2007; Shpitser and Pearl, 2008] for further applications of N-world networks in the context of counterfactual reasoning. The Appendix shows that the treewidth bound of Theorem 4 holds for a generalization of N-world networks that permits the duplication of only a subset of base nodes and allows certain edges that extend between worlds.

Our complexity bounds thus far apply to any counterfactual query. For a specific counterfactual query, we can further reduce the complexity of inference by pruning nodes and edges as in [Darwiche, 2009, Ch 6] and merging nodes which leads to *counterfactual graphs* as in [Shpitser and Pearl, 2007].

6 Counterfactual Reasoning with Partially Specified SCMs

The results we presented on N-world networks, which include twin networks, apply directly to fully specified SCMs. In particular, in the context of variable elimination and jointree algorithms, these results allow us to bound the complexity of computing counterfactual queries in terms of the complexity of computing associational/interventional queries. Moreover, they provide efficient methods for constructing elimination orders and jointrees that can be used for computing counterfactual queries based on the ones used for answering associational/interventional queries, while ensuring that the stated bounds will be realized. Recall again that our bounds and constructions apply to both traditional treewidth and the more recent causal treewidth.

Causal reasoning can also be conducted on partially specified SCMs and data, which is a more common and challenging task. A partially specified SCM typically includes the SCM structure and some information about its parameters (i.e., its structural equations and the distributions over its exogenous variables). For example, we may not know any of the SCM paramaters, or we may know the structural equations but not the distributions over exogenous variables as assumed in [Zaffalon *et al.*, 2021]. A central question in this setup is whether the available information, which includes data, is sufficient to obtain a point estimate for the causal query of interest, in which case the query is said to be identifiable. A significant amount of work has focused on characterizing conditions under which causal queries (both counterfactual and interventional) are identifiable; see, [Pearl, 2009; Spirtes *et al.*, 2000] for textbook discussions of this subject and [Shpitser and Pearl, 2008; Correa *et al.*, 2021] for some results on the identification of counterfactual queries.

When a query is identifiable, the classical approach for estimating it is to derive an estimand using techniques such as the do-calculus for interventional queries [Pearl, 1995; Tian and Pearl, 2002; Shpitser and Pearl, 2006].[4] Some recent approaches take a different direction by first estimating the SCM parameters to yield a fully specified SCM that is then used to answer (identifiable) interventional and counterfactual queries using classical inference algorithms [Zaffalon *et al.*, 2022; Zaffalon *et al.*, 2021; Darwiche, 2021]. Our results on twin and N-world networks apply directly in this case as they can be used when conducting inference on the fully parameterized SCM. For unidentifiable queries, the classical approach is to derive a closed-form bound on the query; see, for example, [Balke and Pearl, 1994a; Pearl, 1999; Tian and Pearl, 2000; Dawid *et al.*, 2017; Rosset *et al.*, 2018; Evans, 2018; Zhang *et al.*, 2021; Mueller *et al.*, 2022]. Some recent approaches take a different direction for establishing bounds, such as reducing the problem into one of polynomial programming [Duarte *et al.*, 2021; Zhang *et al.*, 2022] or inference on credal networks [Zaffalon *et al.*, 2020; Cozman, 2000; Mauá and Cozman, 2020]. Another recent direction is to establish (approximate) bounds by estimating SCM parameters and then using classical inference algorithms on the fully specified SCM to obtain point estimates [Zaffalon *et al.*, 2021; Zaffalon *et al.*, 2022]. Since the query is not identifiable, different parametrizations can lead to different point estimates which are employed to improve (widen) the computed bounds. Our results can also be used in this case for computing point estimates based on a particular parametrization (fully specified SCM) within the overall process of establishing bounds.

7 Experimental Results

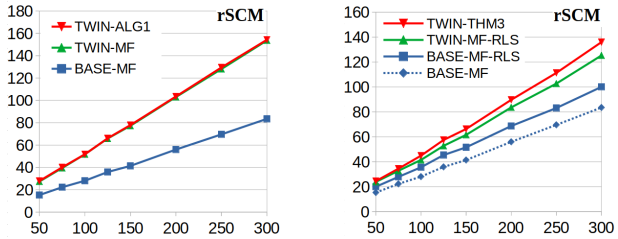
We consider experiments that target random networks whose structures emulate the structures of SCMs used in counter-

[4] See [Jung *et al.*, 2021b; Jung *et al.*, 2021a; Jung *et al.*, 2020] for some recent work on estimating identifiable interventional queries from finite data.

factual reasoning. We have a few objectives in mind. First, we wish to compare the widths of base and twin jointrees, with and without thinning. These widths do not correspond to (causal) treewidth since the jointrees are constructed using heuristics (finding optimal jointrees is NP-hard). Next, we want to compare the quality of twin jointrees constructed by Algorithm 1 (TWIN-ALG1), which operates directly on a base jointree, to the quality of twin jointrees obtained by applying the minfill heuristic to a twin network (TWIN-MF). Recall that the former method is more efficient than the latter method. Finally, we wish to conduct a similar comparison between the thinned, twin jointrees constructed according to Theorem 3 (TWIN-THM3) and the thinned, twin jointrees obtained by applying the minfill heuristic and thinning rules to a twin network (TWIN-MF-RLS). Again, the former method is more efficient than the latter. The widths of these jointrees will be compared to the widths of base jointrees constructed by minfill (BASE-MF) and thinned, base jointrees constructed by minfill and thinning rules (BASE-MF-RLS).

We generated random networks according to the method used in [Darwiche, 2020]. Given a number of nodes n and a maximum number of parents p, the method chooses the parents of node X_i randomly from the set $X_1, \ldots, X_{i-1}$. The number of parents for node X_i is chosen randomly from the set $0, \ldots, min(p, i-1)$. We refer to these networks as rNET. We then consider each internal node N and add a unique root R as parent for N. This is meant to emulate the structure of SCMs as the exogenous variable R can be viewed as representing the different causal mechanisms for endogenous variable N. We refer to these modified networks as rSCM. The twin networks of rSCM are more complex than those for rNET since more variables are shared between the two slices representing the real and imaginary worlds (i.e., more information is shared between the two worlds). We used $n \in \{50, 75, 100, 125, 150, 200, 250, 300\}$ and $p \in \{3, 5, 7\}$. For each combination of n and p, we generated 50 random, base networks and reported averages of two properties for the constructed jointrees: width and *normalized width*. If a jointree has clusters $\mathbf{C}_1, \ldots, \mathbf{C}_n$, then normalized width is $\log_2 \sum_{i=1}^{n} 2^{|\mathbf{C}_i|}$. This accounts for all clusters in the jointree (instead of just the largest one) and the jointree size. The data we generated occupies significant space so we included it in the Appendix while providing representative plots in Figure 6 for jointree widths under $p = 5$. We next discuss patterns exhibited in these plots and the full data in the Appendix, which also includes experiments using random networks generated according to the method in [Ide and Cozman, 2002].

First, the widths of twin jointrees are always less than twice the widths of their base jointrees and often significantly less than that. This is not guaranteed by our theoretical bounds as those apply to (causal) treewidth not to the widths of jointrees produced by heuristics — the latter widths are an upper bound on the former. Second, constructing a twin jointree by directly applying Algorithm 1 to a base jointree (TWIN-ALG1) is relatively comparable to constructing the twin jointree by operating on the twin network (TWIN-MF), as would normally be done. This also holds for thinned jointrees (TWIN-THM3 vs TWIN-MF-RLS) and is encouraging since the former methods are much more efficient than the latter ones.



(a) classical jointrees (b) thinned jointrees

Figure 6: Width of jointrees (y-axis) against number of base network nodes (x-axis) for maximum number of parents $p = 5$.

Third, the employment of thinned jointrees can lead to significant reduction in width and hence an exponential reduction in reasoning time. This can be seen by comparing the widths of twin jointrees TWIN-THM3 and TWIN-ALG1 since the former is thinned but the latter is not (similarly for TWIN-MF-RLS and TWIN-MF). Fourth, the twin jointrees of rSCM have larger widths than those of rNET. Recall that in rSCM, every endogenous variable has its own exogenous variable as a parent. Therefore, the distribution over exogenous variables has a larger space in rSCM compared to rNET. Since this distribution needs to be shared between the real and imaginary worlds, counterfactual reasoning with rSCM is indeed expected to be more complex computationally than reasoning with rNET. Finally, consider Figure 6b for a bottom-line comparison between the complexity of counterfactual reasoning and the complexity of associational/interventional reasoning in practice. Jointrees BASE-MF have the smallest widths for base networks so these are the jointrees one would use for associational/interventional reasoning. The best twin jointrees are TWIN-MF-RLS which are thinned. This is what one would use for counterfactual reasoning. The widths of latter jointrees are always less than twice the widths of the former, and quite often significantly much less.[5]

8 Conclusion

We studied the complexity of counterfactual reasoning on fully specified SCMs in relation to the complexity of associational and interventional reasoning on these models. Our basic finding is that in the context of algorithms based on (causal) treewidth, the former complexity is no greater than quadratic in the latter when counterfactual reasoning involves only two worlds. We extended these results to counterfactual reasoning that requires multiple worlds, showing that the gap in complexity is bounded polynomially by the number of needed worlds. Our empirical results suggest that for two types of random SCMs, the complexity of counterfactual reasoning is closer to that of associational and interventional reasoning than our worst-case theoretical analysis may suggest. While our results directly target counterfactual reasoning on fully specified SCMs, we also discussed cases when they can be applied to counterfactual reasoning on partially specified SCMs that are coupled with data.

[5] See footnote 3 for why BASE-MF is better than BASE-MF-RLS for rSCM.

Acknowledgements

We wish to thank Haiying Huang, Scott Mueller, Judea Pearl, Ilya Shpitser, Jin Tian and Marco Zaffalon for providing useful feedback on an earlier version of this paper. This work has been partially supported by ONR grant N000142212501.

References

[Avin *et al.*, 2005] Chen Avin, Ilya Shpitser, and Judea Pearl. Identifiability of path-specific effects. In *IJCAI*, pages 357–363. Professional Book Center, 2005.

[Balke and Pearl, 1994a] Alexander Balke and Judea Pearl. Counterfactual probabilities: Computational methods, bounds and applications. In *UAI*, pages 46–54. Morgan Kaufmann, 1994.

[Balke and Pearl, 1994b] Alexander Balke and Judea Pearl. Probabilistic evaluation of counterfactual queries. In *AAAI*, pages 230–237. AAAI Press / The MIT Press, 1994.

[Balke and Pearl, 1995] Alexander Balke and Judea Pearl. Counterfactuals and policy analysis in structural models. In *UAI*, pages 11–18. Morgan Kaufmann, 1995.

[Bareinboim *et al.*, 2021] Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. On Pearl's hierarchy and the foundations of causal inference. 2021. Technical Report, R-60, Colombia University.

[Chen and Darwiche, 2022] Yizuo Chen and Adnan Darwiche. On the definition and computation of causal treewidth. In *UAI*, volume 180 of *Proceedings of Machine Learning Research*, pages 368–377. PMLR, 2022.

[Correa *et al.*, 2021] Juan D. Correa, Sanghack Lee, and Elias Bareinboim. Nested counterfactual identification from arbitrary surrogate experiments. In *NeurIPS*, pages 6856–6867, 2021.

[Cozman, 2000] Fábio Gagliardi Cozman. Credal networks. *Artif. Intell.*, 120(2):199–233, 2000.

[Darwiche, 2003] Adnan Darwiche. A differential approach to inference in Bayesian networks. *J. ACM*, 50(3):280–305, 2003.

[Darwiche, 2009] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[Darwiche, 2020] Adnan Darwiche. An advance on variable elimination with applications to tensor-based computation. In *ECAI*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2559–2568. IOS Press, 2020.

[Darwiche, 2021] Adnan Darwiche. Causal inference with tractable circuits. In *WHY Workshop, NeurIPS*, 2021. https://arxiv.org/abs/2202.02891.

[Darwiche, 2022] Adnan Darwiche. Tractable Boolean and arithmetic circuits. In Pascal Hitzler and Md Kamruzzaman Sarker, editors, *Neuro-symbolic Artificial Intelligence: The State of the Art*, volume 342, chapter 6. Frontiers in Artificial Intelligence and Applications. IOS Press, 2022.

[Dawid *et al.*, 2017] Philip Dawid, Monica Musio, and Rossella Murtas. The probability of causation. *Law, Probability and Risk*, (16):163–179, 2017.

[Dean and Kanazawa, 1989] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5, 1989.

[Dechter, 1996] Rina Dechter. Bucket elimination: A unifying framework for probabilistic inference. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 211–219, 1996.

[Duarte *et al.*, 2021] Guilherme Duarte, Noam Finkelstein, Dean Knox, Jonathan Mummolo, and Ilya Shpitser. An automated approach to causal inference in discrete settings. *CoRR*, abs/2109.13471, 2021.

[Evans, 2018] Robin J. Evans. Margins of discrete Bayesian networks. *The Annals of Statistics*, 46(6A):2623 – 2656, 2018.

[Galles and Pearl, 1998] David Galles and Judea Pearl. An axiomatic characterization of causal counterfactuals. *Foundations of Science*, 3(1):151–182, 1998.

[Halpern, 2000] Joseph Y. Halpern. Axiomatizing causal reasoning. *Journal of Artificial Intelligence Research*, 12:317–337, 2000.

[Hamscher *et al.*, 1992] Walter Hamscher, Luca Console, and Johan de Kleer, editors. *Readings in Model-Based Diagnosis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[Han *et al.*, 2023] Yunqiu Han, Yizuo Chen, and Adnan Darwiche. On the complexity of counterfactual reasoning. *CoRR*, abs/2211.13447, 2023.

[Huang and Darwiche, 2023] Haiying Huang and Adnan Darwiche. An algorithm and complexity results for causal unit selection. In *2nd Conference on Causal Learning and Reasoning*, Proceedings of Machine Learning Research, 2023.

[Ide and Cozman, 2002] Jaime Shinsuke Ide and Fábio Gagliardi Cozman. Random generation of bayesian networks. In *SBIA*, volume 2507 of *Lecture Notes in Computer Science*, pages 366–375. Springer, 2002.

[Jung *et al.*, 2020] Yonghan Jung, Jin Tian, and Elias Bareinboim. Learning causal effects via weighted empirical risk minimization. In *NeurIPS*, 2020.

[Jung *et al.*, 2021a] Yonghan Jung, Jin Tian, and Elias Bareinboim. Estimating identifiable causal effects on markov equivalence class through double machine learning. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 5168–5179. PMLR, 2021.

[Jung *et al.*, 2021b] Yonghan Jung, Jin Tian, and Elias Bareinboim. Estimating identifiable causal effects through double machine learning. In *AAAI*, pages 12113–12122. AAAI Press, 2021.

[Kjaerulff, 1990] Uffe Kjaerulff. Triangulation of graphs – algorithms giving small total state space. 1990. Technical

Report R-90-09, Department of Mathematics and Computer Science, University of Aalborg, Denmark.

[Kjærulff, 1994] Uffe Kjærulff. Reduction of computational complexity in bayesian networks through removal of weak dependences. In *UAI*, pages 374–382. Morgan Kaufmann, 1994.

[Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[Larrañaga *et al.*, 1997] Pedro Larrañaga, Cindy M. H. Kuijpers, Mikel Poza, and Roberto H. Murga. Decomposing bayesian networks: triangulation of the moral graph with genetic algorithms. *Statistics and Computing*, 7, 1997.

[Lauritzen and Spiegelhalter, 1988] Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):157–224, 1988.

[Li and Pearl, 2022] Ang Li and Judea Pearl. Unit selection with nonbinary treatment and effect. *CoRR*, abs/2208.09569, 2022.

[Mauá and Cozman, 2020] Denis Deratani Mauá and Fabio Gagliardi Cozman. Thirty years of credal networks: Specification, algorithms and complexity. *International Journal of Approximate Reasoning*, 126:133–157, 2020.

[Mengshoel *et al.*, 2010] Ole J. Mengshoel, Mark Chavira, Keith Cascio, Scott Poll, Adnan Darwiche, and N. Serdar Uckun. Probabilistic model-based diagnosis: An electrical power system case study. *IEEE Trans. Syst. Man Cybern. Part A*, 40(5):874–885, 2010.

[Mueller *et al.*, 2022] Scott Mueller, Ang Li, and Judea Pearl. Causes of effects: Learning individual responses from population data. In *IJCAI*, pages 2712–2718. ijcai.org, 2022.

[Pearl and Mackenzie, 2018] Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, 2018.

[Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[Pearl, 1995] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

[Pearl, 1999] Judea Pearl. Probabilities of causation: Three counterfactual interpretations and their identification. *Synth.*, 121(1-2):93–149, 1999.

[Pearl, 2009] Judea Pearl. *Causality*. Cambridge University Press, second edition, 2009.

[Peters *et al.*, 2017] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

[Robertson and Seymour, 1986] Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of treewidth. *J. Algorithms*, 7(3):309–322, 1986.

[Rosset *et al.*, 2018] Denis Rosset, Nicolas Gisin, and Elie Wolfe. Universal bound on the cardinality of local hidden variables in networks. *Quantum Info. Comput.*, 18(11–12):910–926, sep 2018.

[Shpitser and Pearl, 2006] Ilya Shpitser and Judea Pearl. Identification of joint interventional distributions in recursive semi-markovian causal models. In *AAAI*, pages 1219–1226. AAAI Press, 2006.

[Shpitser and Pearl, 2007] Ilya Shpitser and Judea Pearl. What counterfactuals can be tested. In *UAI*, pages 352–359. AUAI Press, 2007.

[Shpitser and Pearl, 2008] Ilya Shpitser and Judea Pearl. Complete identification methods for the causal hierarchy. *J. Mach. Learn. Res.*, 9:1941–1979, 2008.

[Spirtes *et al.*, 2000] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search, Second Edition*. Adaptive computation and machine learning. MIT Press, 2000.

[Tian and Pearl, 2000] Jin Tian and Judea Pearl. Probabilities of causation: Bounds and identification. *Ann. Math. Artif. Intell.*, 28(1-4):287–313, 2000.

[Tian and Pearl, 2002] Jin Tian and Judea Pearl. A general identification condition for causal effects. In *AAAI/IAAI*, pages 567–573. AAAI Press / The MIT Press, 2002.

[Zaffalon *et al.*, 2020] Marco Zaffalon, Alessandro Antonucci, and Rafael Cabañas. Structural causal models are (solvable by) credal networks. In *International Conference on Probabilistic Graphical Models, PGM*, volume 138 of *Proceedings of Machine Learning Research*, pages 581–592. PMLR, 2020.

[Zaffalon *et al.*, 2021] Marco Zaffalon, Alessandro Antonucci, and Rafael Cabañas. Causal Expectation-Maximisation. In *WHY Workshop, NeurIPS*, 2021. https://arxiv.org/abs/2011.02912.

[Zaffalon *et al.*, 2022] Marco Zaffalon, Alessandro Antonucci, Rafael Cabañas, David Huber, and Dario Azzimonti. Bounding counterfactuals under selection bias. In *PGM*, volume 186 of *Proceedings of Machine Learning Research*, pages 289–300. PMLR, 2022.

[Zhang and Poole, 1994] Nevin Lianwen Zhang and David L. Poole. Intercausal independence and heterogeneous factorization. In *UAI*, pages 606–614. Morgan Kaufmann, 1994.

[Zhang *et al.*, 2021] Junzhe Zhang, Jin Tian, and Elias Bareinboim. Partial counterfactual identification from observational and experimental data. *CoRR*, abs/2110.05690, 2021.

[Zhang *et al.*, 2022] Junzhe Zhang, Jin Tian, and Elias Bareinboim. Partial counterfactual identification from observational and experimental data. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 26548–26558. PMLR, 2022.