Provable Guarantees on the Robustness of Decision Rules to Causal Interventions[*]

Benjie Wang[†] , **Clare Lyle**[†] and **Marta Kwiatkowska**
University of Oxford
benjie.wang@cs.ox.ac.uk

Abstract

Robustness of decision rules to shifts in the data-generating process is crucial to the successful deployment of decision-making systems. Such shifts can be viewed as interventions on a causal graph, which capture (possibly hypothetical) changes in the data-generating process, whether due to natural reasons or by the action of an adversary. We consider causal Bayesian networks and formally define the *interventional robustness* problem, a novel *model-based* notion of robustness for decision functions that measures worst-case performance with respect to a set of interventions that denote changes to parameters and/or causal influences. By relying on a tractable representation of Bayesian networks as arithmetic circuits, we provide efficient algorithms for computing guaranteed upper and lower bounds on the interventional robustness probabilities. Experimental results demonstrate that the methods yield useful and interpretable bounds for a range of practical networks, paving the way towards provably causally robust decision-making systems.

1 Introduction

As algorithmic decision-making systems become widely deployed, there has been an increasing focus on their safety and robustness, particularly when they are applied to input points outside of the data distribution they were trained on. Much of the work in this area has focused on instance-based robustness properties of classifiers, which guarantee that the prediction does not change in some vicinity of a specific input point [Shih *et al.*, 2018; Narodytska *et al.*, 2018]. However, there are many types of distribution shift that cannot be characterized by robustness against norm-bounded perturbations to individual inputs. Such distribution shifts are often instead characterized by causal interventions on the data-generating process [Quionero-Candela *et al.*, 2009; Zhang *et al.*, 2015; Lipton *et al.*, 2018]. These interventions give rise to a range of different environments (distributions), which can be the effect of natural shifts (e.g. different country) or actions of other agents (e.g. a hospital changing prescription policy).

To assess the impact of such interventions, we must leverage knowledge about the causal structure of the data-generating distribution. This paper concerns itself with a simple question: given a decision-making system and a posited causal model, is the system robust to a set of plausible interventions to the causal model? Defining and verifying such *model-based* notions of robustness requires a formal representation of the decision-making system. For discrete input features and a discrete output class, regardless of how a classifier is learned, its role in decision-making can be unambiguously represented by its *decision function*, mapping features to an output class. This observation has spurred a recent trend of applying logic for meta-reasoning about classifier properties, such as monotonicity and instance-based robustness, by *compiling* the classifier into a tractable form [Shih *et al.*, 2018; Narodytska *et al.*, 2018; Audemard *et al.*, 2020], for example an ordered decision diagram. We extend this approach to causal modelling by combining logical representations of the decision rule and causal model, and compiling this joint representation into an arithmetic circuit, a tractable representation of probability distributions.

Our main technical contributions are as follows. First, we motivate and formalize the robustness of a decision rule with respect to interventions on a causal model, which we call the *interventional robustness problem*, and characterize its complexity. Second, we develop a *joint compilation* technique which allows us to reason about a causal model and decision function simultaneously. Finally, we develop and evaluate algorithms for computing upper and lower bounds on the interventional robustness problem, enabling the verification of robustness of decision-making systems to causal interventions.

1.1 Related Work

The problem of constructing classifiers which are robust to distribution shifts has received much attention from the machine learning perspective [Quionero-Candela *et al.*, 2009; Zhang *et al.*, 2015; Lipton *et al.*, 2018]. Particularly relevant to our work are *proactive* approaches to learning robust classifiers, which aim to produce classifiers that perform well across a range of environments (rather than a specific one) [Rojas-Carulla *et al.*, 2018; Subbaswamy *et al.*, 2019].

[*] Proofs and further details can be found in the Appendices of https://arxiv.org/abs/2105.09108

[†] Equal contribution

A recent line of work analyses the behaviour of machine learning classifiers using symbolic and logical approaches by compiling these classifiers into suitable logical representations [Narodytska *et al.*, 2018; Shi *et al.*, 2020]. Such representations can be used to answer a range of explanation and verification queries [Audemard *et al.*, 2020] about the classifier tractably, depending on the properties of the underlying propositional language . Our work uses this premise to tackle defining and verifying robustness to distribution shift, which involves not only the classifier but also a probablistic causal model such as a causal Bayesian network.

In the Bayesian network literature, *sensitivity analysis* [Chan and Darwiche, 2004] is concerned with examining the effect of (typically small) local changes in parameter values on a target probability. We are concerned with providing worst-case guarantees against a set of possible causal interventions, which can involve changing parameters in multiple CPTs, and even altering the graphical structure of the network. This requires new methods that enable scalability to these large, potentially structural intervention sets. Our causal perspective generalizes and extends the work of [Qin, 2015], considering a richer class of interventions than previous work and using this perspective to prove robustness properties of a decision function.

2 Background and Notation

In the rest of this paper, we use $\boldsymbol{V} = (\boldsymbol{X}, Y, \boldsymbol{H})$ to denote the set of modelled variables, which includes observable features $\boldsymbol{X}$, the prediction target Y, and hidden variables $\boldsymbol{H}$. We use lower case (e.g. $\boldsymbol{x}$) to denote instantiations of variables.

2.1 Decision Functions

Consider the task of predicting Y given $\boldsymbol{X}$. Though many machine learning (ML) techniques exist for this task, once learned, the input-output behaviour of any classifier can be characterized by means of a symbolic decision function F from $\boldsymbol{X}$ to Y. For many important classes of ML methods, including Bayesian network classifiers, binarized neural networks, and random forests, it is possible to encode the corresponding decision function as a Boolean circuit Σ [Audemard *et al.*, 2020; Narodytska *et al.*, 2018; Shih *et al.*, 2019]. Such logical encodings can then be used to reason about the behaviour of the decision function, for instance providing explanations for decisions and verifying properties.

2.2 Causal Bayesian Networks

In this paper, we are interested in robust performance of decision functions under distribution shift caused by changes in the data-generating process (DGP). In order to reason about this, we first need a *causal model* of the DGP which enables such changes to be represented. We first define Bayesian networks, which are a convenient way to specify a joint distribution over the set of variables $\boldsymbol{V} = \{V_1, V_2, ..., V_n\}$:

Definition 1 (Bayesian Network)**.** *A (discrete) Bayesian network (BN) $\mathcal{N}$ over variables $\boldsymbol{V}$ is a pair $(\mathcal{G}, \boldsymbol{\Theta})$. $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ is a directed acyclic graph (DAG) whose nodes correspond to the random variables $\boldsymbol{V}$ and whose edges indicate conditional dependence. $\boldsymbol{\Theta}$ denotes the set of conditional probability tables (CPTs) $\boldsymbol{\theta}_{V_i|\boldsymbol{U}_i}$ with parameters $\theta_{v_i|\boldsymbol{u}_i} = P(V_i = v_i|\boldsymbol{U}_i = \boldsymbol{u}_i)$ which specify the distribution, where $\boldsymbol{U}_i = \mathrm{pa}_{\mathcal{G}}(V_i)$ are the parents of V_i in $\mathcal{G}$. We will denote by $p_{\mathcal{N}}$ the distribution defined by the BN $\mathcal{N}$.*

Causal Bayesian networks (CBNs) are defined similarly to Bayesian networks, with the addition of *causal*, or *interventional*, semantics to the joint distribution. Intuitively, an edge (V, V') in a causal Bayesian network indicates that V *causes* V', and the CPTs correspond to causal mechanisms. An intervention can be defined to be a change to some of these mechanisms, replacing $\boldsymbol{\Theta}$ with $\boldsymbol{\Theta}'$. A CBN can thus be characterized as representing a set of distributions, each of which is generated by a different intervention.

We now define a representation of a (causal) Bayesian network, called the *network polynomial*, based on the seminal work of [Darwiche, 2003]. This is a multi-linear function of *indicator* variables, encoding the BN variables $\boldsymbol{V}$, and *parameter* variables, encoding the BN parameters.

Definition 2 (Network Polynomial)**.** *The network polynomial of causal BN $\mathcal{N}$ is defined to be:*

$$l_{\mathcal{N}}[\boldsymbol{\lambda}, \boldsymbol{\Theta}] = \sum_{v_1,\ldots,v_n} \prod_{i=1}^{n} \lambda_{v_i} \theta_{v_i|\boldsymbol{u}_i} \tag{1}$$

where λ_{v_i} denotes an indicator variable *for each value v_i in the support of each random variable V_i, and $\theta_{v_i|\boldsymbol{u}_i}$ denotes each element of a CPT in $\boldsymbol{\Theta}$. Each component of the addition $l_{\boldsymbol{v}}[\boldsymbol{\lambda}, \boldsymbol{\Theta}] := \prod_{i=1}^{n} \lambda_{v_i} \theta_{v_i|\boldsymbol{u}_i}$ is called a **term**, and is associated with an instantiation $\boldsymbol{V} = \boldsymbol{v}$.*

2.3 Arithmetic Circuits

Arithmetic circuits (AC) are computational graphs used to encode probability distributions over a set of discrete variables $\boldsymbol{V}$, which can tractably answer a broad range of probabilistic queries, depending on certain structural properties (called decomposability, smoothness and determinism). They were first introduced by [Darwiche, 2003] as a means of *compiling* Bayesian networks for the purposes of efficient inference. Subsequently they have been considered as objects of study in their own right, with proposals for directly learning ACs from data [Lowd and Domingos, 2008] and extensions relaxing determinism [Poon and Domingos, 2011].

Definition 3 (Arithmetic Circuit)**.** *An arithmetic circuit $\mathcal{AC}$ over variables $\mathbf{V}$ and with parameters $\boldsymbol{\Phi}$ is a rooted directed acyclic graph (DAG), whose internal nodes are labelled with with $+$ or $\times$ and whose leaf nodes are labelled with indicator variables λ_v, where v is the value of some variable $V \in \boldsymbol{V}$, or non-negative parameters ϕ.*

Crucially, evaluating an arithmetic circuit can be done in time linear in the size (number of edges) of the circuit. When an AC represents a probability distribution, this means that marginals can be computed efficiently.

Like Bayesian networks, arithmetic circuits can be represented as polynomials over indicator and parameter variables, based on subcircuits [Choi and Darwiche, 2017]:

Definition 4 (Complete Subcircuit)**.** *A complete subcircuit α of an AC is obtained by traversing the AC top-down, choosing one child of every visited $+$-node and all children of every*

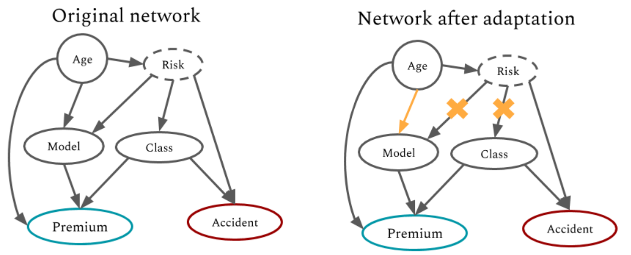

Figure 1: An example causal model describing accident risk for a car insurance problem, and illustrating how strategic adaptation to a classifier can be characterized as a change to a causal model describing how the data was generated.

visited $\times$-node. The ***term*** $term(\alpha)$ *of α is the product of all leaf nodes visited (i.e. all indicator and parameter variables). The set of all complete subcircuits is denoted $\boldsymbol{\alpha}_{\mathcal{AC}}$.*

Definition 5 (AC Polynomial)**.** *The AC polynomial of arithmetic circuit $\mathcal{AC}$ is defined to be:*

$$l_{\mathcal{AC}}[\boldsymbol{\lambda}, \boldsymbol{\Phi}] = \sum_{\alpha \in \boldsymbol{\alpha}_{\mathcal{AC}}} term(\alpha)$$

3 The Intervention Robustness Problem

Many distribution shifts faced by decision-making systems can be characterized by an *intervention* on the data-generating process. For example, if an insurance company offers reduced premiums to drivers who take a driving class, the way ‘risk aversion’ affects ‘class’ in Figure 1 may change in response as more risk-seeking drivers take driving classes to benefit from reduced premiums. The company therefore seeks to determine whether this policy will be *robust* to changes in this relationship before it deploys the policy.

To model this, we formulate an *intervention robustness* problem, which considers the worst-case drop in performance of a classifier in response to changes to a *subset* of the causal mechanisms (CPTs) of the Bayesian network. This is inspired by the principle of *independent causal mechanisms* (ICM) [Peters *et al.*, 2017], which states that causal mechanisms do not inform or influence each other; that is, even as some mechanisms are changed, other mechanisms tend to remain invariant. In the insurance example, this is reflected in that we would not necessarily expect the way ’risk aversion’ or ’accident’ is generated to change, for instance.

While many related notions of robustness exist in the literature, none accurately captures this notion of robustness to causal mechanism changes. Many popular definitions of robustness measure the size of a perturbation necessary to change an input’s classification, without taking into account that such perturbations may change the value which the classifier tries to predict [Shih *et al.*, 2018]. [Miller *et al.*, 2020] highlight the connection between causal inference and robustness to distribution shifts caused by ‘gaming’ in the *strategic classification* [Hardt *et al.*, 2016] regime. However, [Miller *et al.*, 2020] does not assume access to a known causal model, and its focus is on identifying classifiers which are robust to gaming, whereas our objective is to verify robustness to a much richer collection of distribution shifts.

3.1 Intervention Classes

To reason about the effects of changes to a causal model, we need a formal description of these interventions. We consider interventions as *actions* that modify the mechanisms of a causal Bayesian network $\mathcal{N} = (\mathcal{G}, \boldsymbol{\Theta})$, thereby changing its joint distribution. In particular, we consider two types of interventions: the first concerns changes to the parameters of the causal model, while the second concerns changes to the existence of cause-effect relationships themselves.

Typically, we might expect that only mechanisms for a subset of variables $\boldsymbol{W} \subseteq \boldsymbol{V}$ will change. In what follows, given a subset of variables $\boldsymbol{W} \subseteq \boldsymbol{V}$, we will use $\boldsymbol{\theta}^{(\mathcal{G})}_{\boldsymbol{W}} \subseteq \boldsymbol{\Theta}$ to denote the parameters associated with the CPTs for variables $W \in \boldsymbol{W}$, where the parents of W are given by graph $\mathcal{G}$.

Definition 6 (Parametric Interventions)**.** *A parametric intervention on variables $\boldsymbol{W}$ substitutes a subset of parameters $\boldsymbol{\theta}^{(\mathcal{G})}_{\boldsymbol{W}}$ for new values $\boldsymbol{\theta}^{(\mathcal{G})\prime}_{\boldsymbol{W}}$ obtaining a new parameter set $\boldsymbol{\Theta}'$, which yields the BN:*

$$\mathcal{N}[\boldsymbol{\theta}^{(\mathcal{G})\prime}_{\boldsymbol{W}}] := (\mathcal{G}, \boldsymbol{\Theta}') \tag{2}$$

Parametric interventions encompass the do-interventions discussed by [Qin, 2015], but allow us to express more complex changes to causal mechanisms than fixing a variable to a set value. We can further consider changes not just to the parameters of the network, but also to its edge structure; such changes to a set of variables $\boldsymbol{W}$ can be described by a *context function* $C_{\boldsymbol{W}} : \boldsymbol{W} \to \mathcal{P}(\boldsymbol{V})$, which replaces the parents of $W \in \boldsymbol{W}$ in $\mathcal{G}$ with $C_{\boldsymbol{W}}(W)$, producing a new graph $\mathcal{G}'$. We refer to such interventions as *structural* interventions. In this work we restrict ourselves to context sets which preserve the acyclicity of the DAG.

Definition 7 (Structural Interventions)**.** *A structural intervention on variables $\boldsymbol{W}$ modifies the edges $\boldsymbol{E}$ of the graph $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ according to a context function $C_{\boldsymbol{W}}$, obtaining a new graph $\mathcal{G}' = (\boldsymbol{V}, \boldsymbol{E}')$, and substitutes parameters $\boldsymbol{\theta}^{(\mathcal{G})}_{\boldsymbol{W}}$ for new values $\boldsymbol{\theta}^{(\mathcal{G}')\prime}_{\boldsymbol{W}}$, obtaining a new parameter set $\boldsymbol{\Theta}'$, which yields the BN:*

$$\mathcal{N}[\boldsymbol{\theta}^{(\mathcal{G}')\prime}_{\boldsymbol{W}}, C_{\boldsymbol{W}}] := (\mathcal{G}', \boldsymbol{\Theta}') \tag{3}$$

We will often be interested in considering all of the possible interventions of a given class on some subset $\boldsymbol{W} \subseteq \boldsymbol{V}$ of the variables in the causal graph. Letting $\mathbb{P}_{\mathcal{G}}(\boldsymbol{W})$ denote the set of valid parameter sets for $\boldsymbol{W} \subseteq \boldsymbol{V}$ in graph $\mathcal{G}$, we will write for parametric interventions:

$$\mathcal{I}_{\mathcal{N}}[\boldsymbol{W}] := \{\mathcal{N}[\boldsymbol{\theta}^{(\mathcal{G})\prime}_{\boldsymbol{W}}] \mid \boldsymbol{\theta}^{(\mathcal{G})\prime}_{\boldsymbol{W}} \in \mathbb{P}_{\mathcal{G}}(\boldsymbol{W})\} \tag{4}$$

and for structural interventions,

$$\mathcal{I}_{\mathcal{N}}[\boldsymbol{W}, C_{\boldsymbol{W}}] := \{\mathcal{N}[\boldsymbol{\theta}^{(\mathcal{G}')\prime}_{\boldsymbol{W}}, C_{\boldsymbol{W}}] \mid \boldsymbol{\theta}^{(\mathcal{G}')\prime}_{\boldsymbol{W}} \in \mathbb{P}_{\mathcal{G}'}(\boldsymbol{W})\} \tag{5}$$

3.2 Problem Definition and Complexity

Accurately assessing the robustness of a decision function F requires an understanding of the causal data-generating process. We propose to model this DGP using a causal Bayesian network $\mathcal{N}$ on all variables $\boldsymbol{V}$, thus enabling causal *model-based* analysis of classifiers. In order to reason about the causal structure $\mathcal{N}$ and decision rule F simultaneously, we add an additional node to the CBN $\mathcal{N}$.

Definition 8 (Augmented BN)**.** *For a CBN $\mathcal{N}$ over variables $\boldsymbol{V}$ and a classifier $F : \boldsymbol{X} \to Y$, we define the augmented BN $\mathcal{N}_F$ based on $\mathcal{N}$ as follows: $\boldsymbol{V}_F = \boldsymbol{V} \cup \{\hat{Y}\}$ with $\mathrm{pa}(\hat{Y}) = \boldsymbol{X}$ and deterministic CPT $\theta_{\hat{y}|\boldsymbol{x}} = \mathbb{1}[\hat{y} = F(\boldsymbol{x})]$.*

This produces a well-defined joint distribution over the variables $\boldsymbol{V}$ and $\hat{Y} = F(\boldsymbol{X})$, which allows us to specify performance metrics as probabilities of events $\boldsymbol{e}$. For instance, a classifier's false positive probability can be expressed as the probability $p_{\mathcal{N}_F}(\boldsymbol{e})$ of event $\boldsymbol{e} = (\hat{Y} = 1) \wedge (Y = 0)$. More importantly, we can consider how these metrics change as the joint distribution changes, due to hypothetical or observed interventions on the causal model. This provides a basis for *model-based* notions of robustness of decision rules.

We use intervention sets to represent all interventions which the modeller considers plausible. The *interventional robustness* problem then concerns the worst-case performance of the decision rule over interventions in that set.

Definition 9 (IntRob)**.** *Given CBN $\mathcal{N}$ and decision rule F, let $\mathcal{I}_{\mathcal{N}_F}$ be an intervention set for the augmented BN $\mathcal{N}_F$, $\boldsymbol{e}$ be an assignment of a subset of the variables in $\boldsymbol{V}$, and $\epsilon > 0$. The interventional robustness problem is that of computing:*

$$\texttt{IntRob}(\mathcal{I}_{\mathcal{N}_F}, \boldsymbol{e}) := \max_{\mathcal{N}' \in \mathcal{I}_{\mathcal{N}_F}} p_{\mathcal{N}'}(\boldsymbol{e}) \ .$$

We also have the corresponding decision problem:

$$\texttt{IntRob}(\mathcal{I}_{\mathcal{N}_F}, \boldsymbol{e}, \epsilon) := \max_{\mathcal{N}' \in \mathcal{I}_{\mathcal{N}_F}} p_{\mathcal{N}'}(\boldsymbol{e}) > \epsilon \ .$$

We will be particularly interested in problem instances where $\mathcal{I}_{\mathcal{N}}$ is of the form $\mathcal{I}_{\mathcal{N}}[\boldsymbol{W}]$, in which case we can view the problem instance as $\texttt{IntRob}((\mathcal{N}, \boldsymbol{W}), \boldsymbol{e})$. Our next result shows that the causal semantics of $\texttt{IntRob}$ do not increase the computational hardness of the problem beyond that of MAP inference.

Theorem 1. *Let $\mathcal{N} = (\mathcal{G}, \boldsymbol{\Theta})$ be a causal Bayesian network, with n nodes and maximal in-degree d. Then an instance of MAP can be reduced to an instance of $\texttt{IntRob}$ on a BN $\mathcal{N}'$ of size linear in $|\mathcal{N}|$, and of treewidth $w' \leq w+2$. An instance of $\texttt{IntRob}$ can be reduced to an instance of MAP on a BN $\mathcal{N}'$ whose CPT $\boldsymbol{\Theta}'$ has size polynomial in the size of $\boldsymbol{\Theta}$, and with treewidth $w' \leq 2w$.*

4 Verification of Intervention Robustness

In this section, we present our approach to verifying interventional robustness. Due to the difficulty of the problem, we seek to approximate $\texttt{IntRob}(\mathcal{I}, \boldsymbol{e})$ by providing guaranteed upper and lower bounds that can be efficiently computed.

4.1 Joint Compilation

Our first goal is to *compile* $\mathcal{N}_F$ into an equivalent arithmetic circuit $\mathcal{AC}$. To do so, we make use of a standard CNF encoding $\Delta_{\mathcal{N}}$ of the causal BN $\mathcal{N}$, defined over the indicator and parameter variables $\boldsymbol{\lambda_V}, \boldsymbol{\Theta}$ [Chavira and Darwiche, 2005], and additionally an encoding of the decision function F.

A naïve encoding of F is to explicitly enumerate all instantiations of features $\boldsymbol{x}$ and prediction $\hat{y}$, and encode these directly as CNF clauses. However, this approach is very inefficient for larger feature sets $\boldsymbol{X}$. We instead assume access to an encoding of the classifier as a Boolean circuit Σ over input features $\boldsymbol{X}$ and prediction $\hat{Y}$. Such a circuit can be converted to CNF through the Tseitin transformation, introducing additional intermediate variables $\boldsymbol{T}$, obtaining a CNF formula Δ_F over $\boldsymbol{\lambda_X}, \boldsymbol{\lambda}_{\hat{Y}}, \boldsymbol{T}$. We then combine the encodings of F and $\mathcal{N}$ simply by conjoining the CNF formulae, to produce a new formula $\Delta_{joint} = \Delta_{\mathcal{N}} \wedge \Delta_F$, over $\boldsymbol{\lambda_V}, \boldsymbol{\lambda}_{\hat{Y}}, \boldsymbol{\Theta}, \boldsymbol{T}$.

To construct an AC $\mathcal{AC}$, we now *compile* this CNF encoding into d-DNNF (deterministic decomposable negation normal form), using the C2D compiler [Darwiche, 2004], and then replace $\vee$-nodes with $+$, $\wedge$-nodes with $\times$, and set all negative literals and literals corresponding to $\boldsymbol{T}$ to 1. This produces an AC with polynomial $l_{\mathcal{AC}}[\boldsymbol{\lambda}, \boldsymbol{\Theta}]$, where $\boldsymbol{\lambda} := \boldsymbol{\lambda_V} \cup \boldsymbol{\lambda}_{\hat{Y}}$. Crucially, this AC is equivalent to the augmented BN, in the following sense:

Proposition 1. *$l_{\mathcal{AC}}[\boldsymbol{\lambda}, \boldsymbol{\Theta}]$ is equivalent to $l_{\mathcal{N}_F}[\boldsymbol{\lambda}, \boldsymbol{\Theta}]$. Further, $\mathcal{AC}$ can be used to faithfully evaluate marginal probabilities $p_{\mathcal{N}'}(\boldsymbol{e})$ under any parametric intervention $\mathcal{N}'$.*

The time and space complexity of this procedure is $O(nw2^w)$, where n is the number of CNF variables and w the treewidth, a measure of the connectivity of the CNF. When jointly compiling a BN and a decision function, we can bound n, w in terms of the individual encodings $\Delta_{\mathcal{N}}, \Delta_F$.

Proposition 2. *Suppose $\Delta_{\mathcal{N}}$ has n variables and treewidth w, and Δ_F has n' variables and treewidth w'. Then Δ_{joint} has exactly $n + n' - |\boldsymbol{\lambda_X}|$ variables, and treewidth at most $\max(w, w', \min(w, w') + |\boldsymbol{\lambda_X}|)$.*

4.2 Orderings

For the correctness of our upper bounding algorithm, it is necessary to impose some structural constraints on the circuit.

Firstly, any circuit compiled using the procedure described above has the property that every $+$-node t has two children, and is associated with some CNF variable c, such that one child branch has c true, and the other has c false (information on the identity of this variable for each $+$-node is provided by the C2D compiler). We need to ensure that the arithmetic circuit only contains $+$-nodes associated with indicators $\boldsymbol{\lambda}$, and not intermediate variables $\boldsymbol{T}$. Provided this is the case, the branches of each $+$-node t will contain contradicting indicators for some unique variable V. We can thus say that t 'splits' variable V, as each of its child branches corresponds to different values of V, and we write $split(t)$ to denote this splitting variable.

Secondly, provided the above holds, we require our circuit to satisfy some constraints of the following form.

Definition 10 (Ordering Constraint)**.** *An arithmetic circuit $\mathcal{AC}$ satisfies the ordering constraint (V_j, V_i) if:*

$$\forall t, t', (split(t) = V_i \wedge split(t') = V_j) \implies t' \text{ is not a descendant of } t \text{ in } \mathcal{AC} \quad (6)$$

Intuitively, our algorithm requires that for BN variables in the intervention set $\boldsymbol{W}$, the relative position of splitting $+$-nodes in the AC agrees with the causal ordering in the BN. More formally, we say that $\mathcal{AC}$ **satisfies the ordering constraints associated with intervention set** $\mathcal{I}_{\mathcal{N}_F}$, if for all

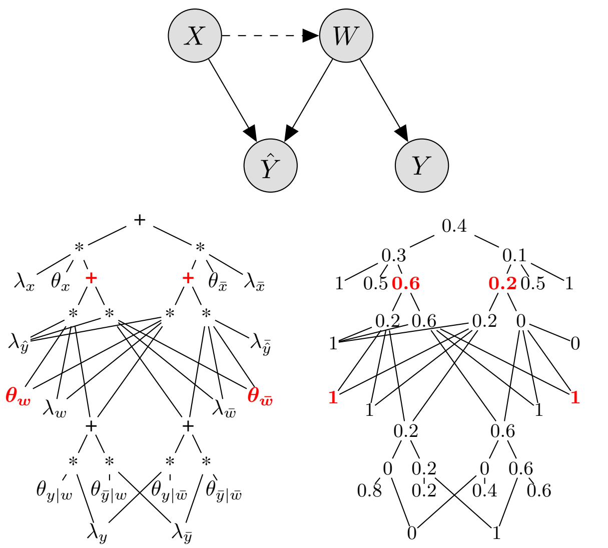

Figure 2: Example augmented BN (top), corresponding AC (bottom left), and execution of Algorithm 1 (bottom right). Nodes which differ from standard AC evaluation are highlighted in red and bold.

```
Algorithm 1: $UB(\mathcal{AC}, \boldsymbol{e}, \boldsymbol{W})$ (Upper Bounding)
  Input: $\mathcal{AC}$, the AC; evidence $\boldsymbol{e}$; intervenable variables $\boldsymbol{W} \subseteq \boldsymbol{V}$;
  Result: Output probability $p$
1 for node $c \in \mathcal{AC}$ (children before parents) do
2     switch type($c$) do
3         case Indicator $\lambda_v$ do
4             $p[c] := 0$ if $v'$ not consistent with $\boldsymbol{e}$ else 1
5         case Parameter $\theta_{v|\boldsymbol{u}}$ do
6             $p[c] := 1$ if $V \in \boldsymbol{W}$else $\theta_{v|\boldsymbol{u}}$
7         case $\times$ do
8             $p[c] := \prod_d p[d]$
              where $d$ are the children of $c$
9         case $+$ do
10            if $c$ splits on some $W \in \boldsymbol{W}$ then
11                $p[c] := \max_d p[d]$
                  where $d$ are the children of $c$
12            else
13                $p[c] := \sum_d p[d]$
                  where $d$ are the children of $c$
14 Return $p[c_{root}]$, where $c_{root}$ is the root node of $\mathcal{AC}$
```

$V_i \in \boldsymbol{W}$, and all V_j such that $V_j \in \text{pa}_{\mathcal{G}}(V_i)$ (parametric intervention set) or $V_j \in \text{pa}_{\mathcal{G}'}(V_i)$ (structural intervention set), $\mathcal{AC}$ satisfies the ordering constraint (V_j, V_i). In practice, when computationally feasible, we compile ACs with *topological* and *structural topological* orderings, which satisfy these constraints for all V_i, not just $V_i \in \boldsymbol{W}$; such orderings have the advantage of being valid for any intervention sets $\boldsymbol{W}$.

We enforce these constraints by enforcing corresponding constraints on the *elimination ordering* π over the CNF variables, which is used to construct the *dtree* that is used in the compilation process, and affects the time and space taken by the compilation. Such an elimination ordering is usually chosen using a heuristic such as min-fill. We instead find a elimination ordering by using a *constrained* min-fill heuristic, which ensures that these constraints are satisfied, but may produce an AC which is much larger than can be achieved with an unconstrained heuristic in practice.

4.3 Upper Bounds on Intervention Robustness

In order to compute upper bounds on the interventional robustness quantity, we propose Algorithm 1, which sets parameters in the AC for the CPTs of variables in $\boldsymbol{W}$ to 1, and applies maximization instead of addition at $+$-nodes splitting on variables in $\boldsymbol{W}$, when evaluating the (appropriately ordered) AC. Algorithm 1 somewhat resembles the well-known MPE algorithm on ACs, introduced by [Chan and Darwiche, 2006] and used as an upper bound on the MAP problem in [Huang *et al.*, 2006]. However, our algorithm maximizes over parameters rather than variables and makes use of specific AC structure ensured by our ordering constraints; the reason it produces correct upper bounds is thus also different.

Intuitively, the maximizations represent decision points, where choosing a child branch corresponds to intervening to set a particular parameter $\theta_{w|\boldsymbol{u_W}}$ to 1 (and others to 0). For example, consider the augmented Bayesian network in Figure 2, where all variables are binary, $\hat{Y} = X \lor W$, and the context $C(W)$ is $\{X\}$ (represented by the dashed line, which is not part of the original BN). Figure 2 also shows execution of Algorithm 1 for false positive probability, i.e. $\boldsymbol{e} = (\hat{Y} = 1) \land (Y = 0)$. At the two $+$-nodes where maximizations occur, the value of X is already "decided", and the adversary can effectively choose to set $\theta_{\bar{w}|x} = 1$ and $\theta_{w|\bar{x}} = 1$. In this case, the result 0.4 turns out to be exactly equal to the interventional robustness quantity $\texttt{IntRob}(\mathcal{I}_{\mathcal{N}_F}, \boldsymbol{e})$.

We might ask whether this intuition is correct in general. Our next result shows that, while the algorithm cannot always compute $\texttt{IntRob}(\mathcal{I}_{\mathcal{N}_F}, \boldsymbol{e})$ exactly, it does produce guaranteed upper bounds:

Theorem 2. *Given a parametric/structural intervention set $\mathcal{I}_{\mathcal{N}_F}$, let $\mathcal{AC}$ be an arithmetic circuit with the same polynomial as $\mathcal{N}_F$, and satisfying the ordering constraints associated with the intervention set. Then, applying the UB algorithm $UB(\mathcal{AC}, \boldsymbol{e}, \boldsymbol{W})$ returns a quantity B_U which is an upper bound on the interventional robustness quantity $\texttt{IntRob}(\mathcal{I}_{\mathcal{N}_F}, \boldsymbol{e})$.*

This result is quite surprising; it shows that it is possible, through a very simple and inexpensive procedure requiring just a single linear time pass through the AC, to upper bound the worst-case marginal probability over an exponentially sized set of interventions. That this also holds for structural intervention sets, which alter the structure of the Bayesian network which the AC was compiled from, is even more surprising. Further, a compiled AC can be used for *any* intervention set given that it satisfies the appropriate ordering constraints. For instance, an AC compiled using a topological ordering allows us to derive upper bounds for parametric intervention sets involving any subset (of any size) $\boldsymbol{W} \subseteq \boldsymbol{V}$, simply by setting the appropriate parameter nodes in the AC to 1 (Line 5). This allows us to amortize the cost of evaluating robustness against multiple intervention sets.

Algorithm 2: Lower Bounding

```
Input: $\mathcal{N} = (\mathcal{G}, \boldsymbol{\Theta})$, a Bayesian network; evidence $\boldsymbol{e}$
       whose probability will be maximized;
       intervenable variables $\boldsymbol{W} \subseteq \boldsymbol{V}$.
Result: Output probability
        $p(e) \leq \max_{\mathcal{N}' \in \mathcal{I}_{\mathcal{N}}[\boldsymbol{W}]} P_{\mathcal{N}'}(\boldsymbol{e})$
begin
    $v \leftarrow 0$;
    while $p_{\mathcal{N}[\boldsymbol{\Theta}_{\boldsymbol{W}}]}(\boldsymbol{e}) > v$ do
        $v \leftarrow p_{\mathcal{N}[\boldsymbol{\Theta}_{\boldsymbol{W}}]}(\boldsymbol{e})$;
        for CPT $\boldsymbol{\theta}^{(\mathcal{G})}_{W|u} \in \boldsymbol{\Theta}_{\boldsymbol{W}}$ do
            $\boldsymbol{\theta}^{(\mathcal{G})}_{W|\boldsymbol{u}} \leftarrow \arg\max_{\boldsymbol{\theta}'_{W|\boldsymbol{u}}} p_{\mathcal{N}[\boldsymbol{\theta}'_{W|\boldsymbol{u}}]}(\boldsymbol{e})$;
```

4.4 Lower Bounds via Best-Response Dynamics

In addition to an upper bound on $\texttt{IntRob}(\mathcal{I}_{\mathcal{N}_F}, \boldsymbol{e})$, we can also straightforwardly lower bound this quantity using any witness, in the case of parametric interventions $\mathcal{I}_{\mathcal{N}_F}[\boldsymbol{W}]$. That is, we search for an intervention in the set which approximately maximizes the probability of evidence $\boldsymbol{e}$.

We obtain such an approach by formalizing the problem of finding an intervention which maximizes $p_{\mathcal{N}'}(\boldsymbol{e})$ as a multiplayer game, where each instantiation $\boldsymbol{u}_W$ of parents $\text{pa}_{\mathcal{G}'}(W)$ for each $W \in \boldsymbol{W}$ specifies a player, and where all players share a utility function given by $p_{\mathcal{N}[\boldsymbol{\Theta}]}(\boldsymbol{e})$. Each player's strategy set consists of the set of deterministic conditional distributions $\boldsymbol{\theta}_{W|\boldsymbol{u}}$ (we note w.l.o.g. that, by the multilinearity of the network polynomial, the optimal value of $P_{\mathcal{N}[\boldsymbol{\Theta}']}(e)$ is obtained by at least one deterministic interventional distribution). A Nash equilibrium in this game then corresponds to an interventional distribution for which no change in a single parameter can increase $p_{\mathcal{N}'}(\boldsymbol{e})$. Algorithm 2 follows best-response dynamics in this game. We provide an analysis of the time complexity and convergence of this approach in the proof of the following proposition.

Proposition 3. *Algorithm 2 converges to a locally optimal parametric intervention in finite time. Further, if the algorithm is stopped before termination, the current value v will be a lower bound on* $\max_{\mathcal{N}' \in \mathcal{I}[\boldsymbol{W}]} P_{\mathcal{N}'}(\boldsymbol{e})$.

5 Case Study: Insurance

In this case study, we look at an extended version of the car insurance example, using the Bayesian network model shown in Figure 3 [Binder *et al.*, 1997].

Suppose an insurance company wishes to predict `MedCost` (the medical cost of an insurance claim), given an insurant's `Age`, `DrivHist`, and `MakeModel` (categorical variables with 3-5 values). `MedCost` is either *BelowThousand* (0) or *AboveThousand* (1). They fit a Naïve Bayes classifier to historical data, obtaining a decision function F. This is then used as part of their decision-making policy determining what premiums to offer to customers.

The company is particularly concerned about false negatives, as the company could lose a lot of money in payouts. Based on the original Bayesian network model (Figure 3) and their new classifier, this should occur 2.5% of the time. However, insurants may attempt to game the classifier to predict *BelowThousand* (so that they get lower premiums), while actually being likely to have a high medical cost. In our framework, we model this using structural interventions, assuming that insurants can causally intervene on some of `DrivHist` (hide some accident history), `MakeModel` (choose a different type of car than they would normally choose), and `Cushioning` (upgrade/downgrade the degree of protection). We use structural intervention sets (with appropriately designed context function) because insurants will have access to some non-parent variables when adapting; for instance, they will know their `Age` when choosing the `MakeModel` of a new car. The company would like to understand how *robust* their classifier is to these adaptations.

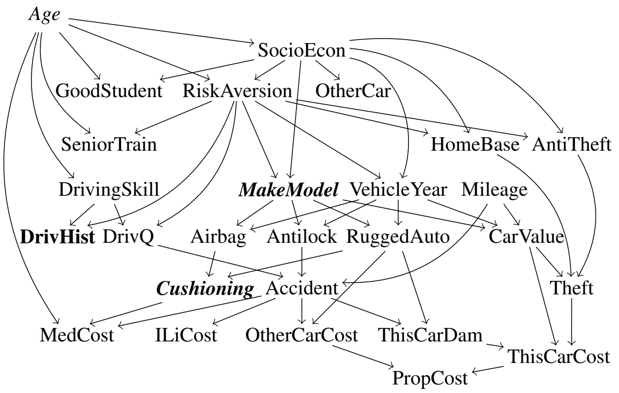

Figure 3: INSURANCE Bayesian network. Classifier features $\boldsymbol{X}$ are italicized, and (potential) interventions are shown in bold.

We consider a number of structural intervention sets $\mathcal{I}_{\mathcal{N}_F}$, given by intervenable variables $\boldsymbol{W}$, which may be any subset of $\{$`DrivHist, MakeModel, Cushioning`$\}$. We use structural sets because we assume the insurant has access to other variables when choosing how to adapt, such as `Age` or `Mileage`, which are not parents of these variables in the original BN. Under each of these intervention sets, we seek to obtain guaranteed upper bounds on these two quantities:

- **FN**: The probability of a false negative $p(F = 0, \texttt{MedCost} = 1)$, i.e. predicted low medical cost, but high actual medical cost.
- **P**: The probability of a positive $p(\texttt{MedCost} = 1)$, i.e. high actual medical cost.

The results are shown in Table 1. The insurance company can use these bounds to assess risk, and improve their classifier's robustness if they deem the false negative rate under intervention unacceptable.

The bounds can also provide further insight. We notice that whenever `DrivHist` is intervenable, the percentage of false negatives is the same as positives, i.e. the classifier always predicts wrong when `MedCost` is 1. This turns out to be because the Naïve Bayes classifier always predicts 0 whenever `DrivHist` is *None*, regardless of the other input variables. Thus, an insurant who can change their `DrivHist` can always fool the classifier to predict 0. In addition, the percentage of positives doesn't increase from the original BN: this

Intervenable Variables W	**FN**	**P**
Empty Set	2.5%	7.2%
{DrivHist}	7.2%	7.2%
{MakeModel}	5.7%	10.0%
{Cushioning}	6.1%	12.9%
{DrivHist, MakeModel}	10.0%	10.0%
{DrivHist, Cushioning}	12.9%	12.9%
{MakeModel, Cushioning}	13.0%	13.9%
{DrivHist, MakeModel, Cushioning}	13.9%	13.9%

Table 1: Guaranteed upper bounds on FN and P, under different structural intervention sets

Net	**CSize**	**Ord**	**TW**	**AC size**	**Time (s)**
insurance	3 (41)	N	24	167121	0.5
	3 (41)	T	31	794267	4
	3 (41)	S	33	1270075	8
win95pts	16 (799)	N	51	1210072	3
	16 (799)	T	58	52266950	77
hepar2	12 (946)	N	53	8096874	49
	12 (946)	T	51	123108407	73
	12 (946)	S	51	123164181	75

Table 2: Results for the joint compilations used in the UB and LB algorithms. Shown are the number of input features d and the sizes of the Boolean circuits representing the classifier, ordering constraints (none, topological, or structural topological), treewidth of the combined CNF encoding, and AC size and compilation time.

can be seen from the causal graph, where `DrivHist` has no causal influence on `MedCost`.

On the other hand, `Cushioning` significantly increases the positive rate. Notice that, in the graph, intervening on `Cushioning` will not have any influence on the inputs to the classifier; thus, the increase in FN to 6.1% is not due to fooling the classifier, but rather making high medical expenses generally more likely, by downgrading the quality of cushioning. In this way, the intervention is ”taking advantage” of the classifier not having full information about cushioning.

6 Evaluations

6.1 Compilation Performance

In Table 2 we show the performance of our joint compilation approach on a number of benchmark Bayesian networks, where we jointly compile the network and a decision rule. We observe that the sizes of the compiled ACs are significantly smaller than the worst-case bounds would suggest (exponential in treewidth). Further, when we enforce a topological or structural topological ordering, the size of the compilation increases, but not by more than ~ 100. Our results provide evidence that our methods can scale to fairly large networks and classifiers, including networks compiled with topological and structural topological orderings.

Network	**IntSet**	**LBound**	**UBound**	**Δ**
insurance	P1	0.1181	0.1276	**0.0095**
	P2	0.3275	0.3433	**0.0158**
	S1	0.1181	0.1297	**0.0116**
win95pts	P1	0.2111	0.2111	**0.0000**
	P2	0.2163	0.2191	**0.0028**
hepar2	P1	0.09445	0.09445	**0.0000**
	P2	0.09585	0.09585	**0.0000**
	S1	0.1029	0.1029	**0.0000**

Table 3: Analysis of the tightness of bounds (on probability of false negatives) produced by Algorithms 1 and 2. For each network, we have different intervention sets (P/S indicates the intervention set is parametric/structural respectively). Lower and upper bounds, along with the difference, are shown for each intervention set.

6.2 Lower and Upper Bound Tightness

In Table 3 we analyse the quality of our upper and lower bounds on interventional robustness. We compute bounds on false negative probability under different intervention sets. Overall, we find small or nonexistent gaps between the lower and upper bounds across all networks and intervention sets evaluated, suggesting that in many settings of interest it is possible to obtain tight guarantees using our algorithms.

Further, both bounding algorithms are very fast to execute, taking no more than a few seconds for each run. This is remarkable given the sizes of the intervention sets. For instance, for the insurance network, the parametric intervention set P2 covers 6 variables ($|\boldsymbol{W}| = 6$), 248 parameters, and $\sim 10^{36}$ different interventions, making brute-force search clearly infeasible. For worst-case (interventional robustness) analysis, the sensitivity analysis method of [Chan and Darwiche, 2004] requires $\sim 10^7$ passes through the AC in this case. On the other hand, our upper bounding algorithm requires an ordered AC (which is ~ 5 times larger in this case), but requires just a *single* pass through the AC, making it $\sim 10^6$ faster. Further, our algorithm is uniquely able to provide guarantees for structural intervention sets.

7 Conclusions

In this work, we have motivated and formalized the interventional robustness problem, developed a compilation technique to produce efficient joint representations for classifiers and DGPs, and provided tractable upper and lower bounding algorithms which we have shown empirically to be tight on a range of networks and intervention sets. The techniques presented here provide ample opportunity for further work, such as extending the upper and lower bounding technique to networks where the modeller has uncertainty over the parameters, and developing learning algorithms for arithmetic circuits which permit reasoning about causal structure.

Acknowledgments

This project was funded by the ERC under the European Union’s Horizon 2020 research and innovation programme (FUN2MODEL, grant agreement No. 834115).

References

[Audemard *et al.*, 2020] Gilles Audemard, Frédéric Koriche, and Pierre Marquis. On Tractable XAI Queries based on Compiled Representations. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*, pages 838–849, 2020.

[Binder *et al.*, 1997] John Binder, Daphne Koller, Stuart Russell, and Keiji Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244, 1997.

[Chan and Darwiche, 2004] Hei Chan and Adnan Darwiche. Sensitivity analysis in bayesian networks: From single to multiple parameters. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, page 67–75, July 2004.

[Chan and Darwiche, 2006] Hei Chan and Adnan Darwiche. On the robustness of most probable explanations. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, page 63–71, July 2006.

[Chavira and Darwiche, 2005] Mark Chavira and Adnan Darwiche. Compiling bayesian networks with local structure. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, page 1306–1312, 2005.

[Choi and Darwiche, 2017] Arthur Choi and Adnan Darwiche. On relaxing determinism in arithmetic circuits. In *Proceedings of the 34th International Conference on Machine Learning*, page 825–833, August 2017.

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

[Darwiche, 2004] Adnan Darwiche. New advances in compiling cnf to decomposable negation normal form. In *Proceedings of the 16th European Conference on Artificial Intelligence*, page 318–322, August 2004.

[Hardt *et al.*, 2016] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, page 111–122, January 2016.

[Huang *et al.*, 2006] Jinbo Huang, Mark Chavira, and Adnan Darwiche. Solving map exactly by searching on compiled arithmetic circuits. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, page 1143–1148, July 2006.

[Lipton *et al.*, 2018] Zachary C. Lipton, Yu-Xiang Wang, and Alexander J. Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3128–3136, July 2018.

[Lowd and Domingos, 2008] Daniel Lowd and Pedro Domingos. Learning arithmetic circuits. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, page 383–392, July 2008.

[Miller *et al.*, 2020] John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *Proceedings of 37th International Conference on Machine Learning*, pages 6917–6926, July 2020.

[Narodytska *et al.*, 2018] Nina Narodytska, Shiva Kasiviswanathan, Leonid Ryzhyk, Mooly Sagiv, and Toby Walsh. Verifying properties of binarized deep neural networks. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 6615–6624, 2018.

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

[Poon and Domingos, 2011] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, page 337–346, July 2011.

[Qin, 2015] Biao Qin. Differential semantics of intervention in bayesian networks. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 710–716, July 2015.

[Quionero-Candela *et al.*, 2009] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[Rojas-Carulla *et al.*, 2018] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *Journal of Machine Learning Research*, 19(36):1–34, 2018.

[Shi *et al.*, 2020] Weijia Shi, Andy Shih, Adnan Darwiche, and Arthur Choi. On tractable representations of binary neural networks. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*, pages 882–892, September 2020.

[Shih *et al.*, 2018] Andy Shih, Arthur Choi, and Adnan Darwiche. Formal verification of bayesian network classifiers. In *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*, pages 427–438. PMLR, September 2018.

[Shih *et al.*, 2019] Andy Shih, Arthur Choi, and Adnan Darwiche. Compiling bayesian network classifiers into decision graphs. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 7966–7974, January 2019.

[Subbaswamy *et al.*, 2019] Adarsh Subbaswamy, Peter Schulam, and Suchi Saria. Preventing failures due to dataset shift: Learning predictive models that transport. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, pages 3118–3127, 2019.

[Zhang *et al.*, 2015] Kun Zhang, Mingming Gong, and Bernhard Scholkopf. Multi-source domain adaptation: A causal view. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, page 3150–3157, January 2015.