Causal Information Splitting: Engineering Proxy Features for Robustness to Distribution Shifts

Abstract

Statistical prediction models are often trained on data that is drawn from different probability distributions than their eventual use cases. One approach to proactively prepare for these shifts harnesses the intuition that causal mechanisms should remain invariant between environments. Here we focus on a challenging setting in which the causal and anticausal variables of the target are unobserved. Leaning on information theory, we develop feature selection and engineering techniques for the observed downstream variables that act as proxies. We identify proxies that help to build stable models and moreover utilize auxiliary training tasks to extract stability-enhancing information from proxies. We demonstrate the effectiveness of our techniques on synthetic and real data.

1 INTRODUCTION

The principle assumption when building any (not necessarily causal) prediction model is access to relevant data for the task at hand. When predicting label Y from inputs $\boldsymbol{X}$, this assumption reads that the data is drawn from a (training) probability distribution $\boldsymbol{X}, Y$ that is identical to the distribution that will generate its use-cases (target distribution).

Unfortunately, the dynamic nature of real-world systems makes obtaining perfectly relevant data difficult. Data-gathering mechanisms can introduce sampling bias, yielding distorted training data. Even in the absence of sampling biases, populations, environments, and interventions give rise to distribution shifts in their own right. For example, Zech et al. [2018] found that convolutional neural networks to detect pneumonia from chest radiographs often relied on site-specific features, including the metallic tokens indicating laterality and image processing techniques. This resulted in poor generalization when these features varied across sites. Understanding these inter-site breakdowns in performance is essential to safety-critical domains such as healthcare.

Transportability and Domain Generalization The first attempts at handling dissociation between training and target distributions involved gathering unlabeled samples of the testing distribution. Within domain generalization (DG), covariate shift handles a shift in the distribution of $\boldsymbol{X}$ [Shimodaira, 2000] and label shift handles a shifting $\Pr(Y)$ [Schweikert et al., 2008]. DG often assumes a stationary label function $\Pr(Y \mid \boldsymbol{X})$, which is extremely limiting in real-life applications.

To address these limitations, one can assume the label function is stationary for a *subset* of the covariates in $\boldsymbol{X}$, called an **invariant set** in Muandet et al. [2013], Rojas-Carulla et al. [2018]. The **transportability** problem concerns itself with finding such an invariant set $\boldsymbol{X}$. One approach to transportability has been to use a rich collection of datasets to capture information about the mechanisms causing the shift Rojas-Carulla et al. [2018], Magliacane et al. [2018].

A causal perspective developed in Storkey et al. [2009] and Pearl and Bareinboim [2011] uses graphical modeling via **selection diagrams** to sidestep these data demands. While the graphical structure could be learned from access to enough data from different environments, aspects of its structure can also be imported from domain expert knowledge. Similarly motivated work has studied stability in the causal and anticausal scenario [Schölkopf et al., 2012].

Contributions The causal perspective to distribution shift is obscured when we lack direct measurements of the causes and effects of Y. Such settings arise from noisy measurements, privacy concerns, as well as concepts that cannot be quantified (such as "work ethic" or "interests"). Instead, we will focus on a setting where we only measure *proxies* for the causes and effects of Y, see Fig. 1 for an example.

We will use concepts from causal inference and information theory to define and study the **proxy-based transporta-**

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

bility problem. While previous approaches to partially observed systems suggest restricting model inputs to those with stable information ([Subbaswamy and Saria, 2018]), no observed variables satisfy this condition in our setting. However, some observed variables are proxies for variables which could have functioned as stable inputs. Our framework will demonstrate that perfection is indeed the enemy of good - restricting models to stable inputs can be deleterious to the model's overall stability. This insight will inform a different strategy for feature selection based on the underlying causal structure of the unobserved variables.

Our paper will build on the observation that post-selecting on a single value of the prediction label Y induces a special independence structure, which the proxies for the causes and effects of Y also inherit. We use this to classify proxies from partial knowledge of a few "seeds" - a technique we call **proxy bootstrapping**.

It is possible that some proxy variables will contain information about both stable and unstable hidden variables. We call these **ambiguous proxies** because it is unclear whether their inclusion will improve or worsen transportability. Inspired by node splitting, we introduce a method we call **causal information splitting (CIS)**, which can improve stability and accuracy of our models at no cost (and even some benefit) to the distribution shift robustness. Again exploiting the inherited independence structure from post-selecting on Y, CIS isolates stabilizing information using seemingly unrelated auxiliary prediction tasks on the covariates. We demonstrate the surprising ability of CIS to separate stabilizing information from ambiguous variables on synthetic data experiments, even when the assumptions of our theory are relaxed. Furthermore, we demonstrate CIS's potential on U.S. Census data which were strongly shifted due to the COVID-19 pandemic. While plenty of experiments have confirmed that techniques for robust models do not consistently provide benefits over empirical risk minimization [Gulrajani and Lopez-Paz, 2021], our proposed technique provides benefits for an income prediction task in a majority of tested states.

2 RELATED WORK

Apart from work on transportability, there is an increasing body of work on domain generalization, see [Quinonero-Candela et al., 2008] for an overview. While we focus on pro-actively modeling shifts, work on invariant risk minimization [Arjovsky et al., 2019, Bellot and van der Schaar, 2020] reacts to shifts assuming access to shifted data. Recent work further generalizes to unseen environments constituting mixtures [Sagawa et al., 2019] and affine combinations [Krueger et al., 2021]. Data from multiple environments can also be used for causal discovery [Peters et al., 2016], [Heinze-Deml et al., 2018].

Another line of work seeks robustness to small adversarial changes in the input that should not change the output (with attacks, e.g. [Croce and Hein, 2020] and defenses, e.g. [Sinha et al., 2018]). Moving from small changes to potentially bigger interventions, work on counterfactual robustness and invariance, introduces additional regularization terms [Veitch et al., 2021], [Quinzan et al., 2022]. Our work differs by allowing for interventions that change the label.

3 BACKGROUND

General Notation Uppercase letters denote random variables, while lowercase letters denote assignments to those random variables. Bold letters denote sets/vectors. The paper will use concepts from information theory, with $\mathcal{H}(A)$ indicating the **entropy** of A, $\mathcal{I}(A:B)$ indicating the **mutual information** between A, B, and $\mathcal{I}(A:B:C)$ indicating the **interaction information** between A, B, C. A short summary of key ideas (including the data processing inequality (DPI) and chain rule) is given in Appendix Section B (see Cover [1999] for more details).

Causal Graphical Models Graphically modeling distribution shift makes use of causal DAGs. For a causal DAG $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$, the joint probability distribution factorizes according to the local Markov condition,

$$\Pr(\boldsymbol{v}) = \prod_{v \in \boldsymbol{v}} \Pr(v \mid \mathbf{pa}^{\mathcal{G}}_{\boldsymbol{v}}(V)).$$

$\mathbf{PA}^{\mathcal{G}}(V), \mathbf{CH}^{\mathcal{G}}(V)$ denote the parents and children of V in $\mathcal{G}$. Similarly, $\mathbf{DE}^{\mathcal{G}}(V)$ and $\mathbf{AN}^{\mathcal{G}}(V)$ denote the descendants and ancestors respectively. $\mathbf{FM}(V) = \mathbf{PA}(V) \cup \mathbf{CH}(V)$ denotes the "family."

Our paper will rely on the concepts of d**-separation** and **active paths** to discuss the independence properties of Bayesian networks, which are discussed in Appendix Section A. See Pearl [2009] for a more extensive study.

Active Path Notation In addition to using $A \perp\!\!\!\perp_d B \mid C$ to indicate d-separation conditioned on C, we will develop a notation to refer to denote sets of variables which act as "switches" for d-separation. $\boldsymbol{A} \leftrightsquigarrow \boldsymbol{C} \rightsquigarrow \boldsymbol{B}$ means that we have both $\boldsymbol{A} \not\perp\!\!\!\perp_d \boldsymbol{B}$ and $\boldsymbol{A} \perp\!\!\!\perp_d \boldsymbol{B} \mid \boldsymbol{C}$. Conversely, we have $\boldsymbol{A} \leftrightsquigarrow \boldsymbol{C} \rightsquigarrow \boldsymbol{B}$ if $\boldsymbol{A} \perp\!\!\!\perp_d \boldsymbol{B}$, but $\boldsymbol{A} \not\perp\!\!\!\perp_d \boldsymbol{B} \mid \boldsymbol{C}$ (i.e. conditioning on $\boldsymbol{C}$ renders $\boldsymbol{A}$ and $\boldsymbol{B}$ d-connected).

Graphically Modeling Distribution Shift Borrowing terms from Magliacane et al. [2018], we will begin with a graphical model $\mathcal{G} = (\boldsymbol{V} \cup \boldsymbol{U})$, calling $\boldsymbol{U} \cup \boldsymbol{V}$ the **system variables** with (un-)observed variables. In addition, we are also given a set of context variables $\boldsymbol{M}$, which model the mechanisms that shift our distribution. The augmentation of $\mathcal{G}$ with $\boldsymbol{M}$ gives what we call the **distribution shift diagram** (DSD), $\mathcal{G}^{+} = (\boldsymbol{V} \cup \boldsymbol{U} \cup \boldsymbol{M}, \boldsymbol{E} \cup \boldsymbol{E}_{\boldsymbol{M}})$, for which

$\mathcal{G}$ is a subgraph, with additional vertices $\boldsymbol{M}$ introducing shifts along $\boldsymbol{E_M}$. The transportability problem [Pearl and Bareinboim, 2011] involves finding an input set $\boldsymbol{X} \subseteq \boldsymbol{V}$ such that $\Pr(Y \mid \boldsymbol{X}) = \Pr(Y \mid \boldsymbol{X}, \boldsymbol{M})$. Such a set $\boldsymbol{X}$, called an "invariant set" in Magliacane et al. [2018], blocks all possible influence from the mechanisms of the dataset shift. Pearl and Bareinboim [2011] shows this framework is capable of modeling sampling bias and population shift.

4 SETTING

4.1 PROXY-BASED TRANSPORTABILITY

This paper will consider the **proxy-based transportability (PBT)** setting. PBT focuses on the role of proxy variables in feature selection by assuming all of the causes and effects $\boldsymbol{U} = \mathbf{FM}(Y)$ are unobserved.[1] We are given access to a list of "visible proxy variables" $\boldsymbol{V} \setminus \{Y\}$ which are children of at least one $U \in \boldsymbol{U}$. Hence, $\boldsymbol{V}$ can be thought of as the union of overlapping subsets $\mathbf{CH}(U)$ for each $U \in \boldsymbol{U}$.

We will assume that there are no edges directly within $\boldsymbol{U}$ or within $\boldsymbol{V}$, which we call **systemic sparsity**. See Figure 1 for an example of this setting. This assumption enforces two useful independence properties: (1) $V_i \perp\!\!\!\perp V_j \mid U$ for $V_i, V_j \in \mathbf{CH}(U)$) and (2) $U_i \perp\!\!\!\perp U_j \mid Y$ for $U_i \neq U_j \in \boldsymbol{U}$.

We will consider distribution shift diagrams $\mathcal{G}^+ = (\boldsymbol{V} \cup \boldsymbol{U} \cup \boldsymbol{M}, \boldsymbol{E} \cup \boldsymbol{E_M})$ with one $M_i \in \boldsymbol{M}$ connected to a corresponding $U_i \in \boldsymbol{U}$. Each M_i models a different shifting mechanism for each unobserved cause and effect of Y. It is common to assume there is no direct shifting mechanism acting on Y - which comes without loss of generality since such a mechanism can be thought of as another unobserved cause Pearl and Bareinboim [2011], Peters et al. [2016].

In this setting, a perfect invariant set $\boldsymbol{X}$ in which $Y \perp\!\!\!\perp_d \boldsymbol{M} \mid \boldsymbol{X}$ does not exist. Proxy-based transportability will instead seek to minimize the influence of the context variables on our label function. Borrowing concepts from information theory, the task in the proxy-based transportability problem corresponds to finding a function $\boldsymbol{X} = F(\boldsymbol{V} \setminus \{Y\})$ whose output minimizes the conditional mutual information between the label and the environment. We call this quantity, $\mathcal{I}(Y : \boldsymbol{M} \mid \boldsymbol{X})$, the **context sensitivity**. $F(\cdot)$ is broadly defined to include feature selection ($\boldsymbol{X} \subseteq \boldsymbol{V}$), as well as feature engineering via higher-level representations.

Difficulties of PBT The PBT setting is difficult to address using existing methods for transportability. Building a model on the causes $\mathbf{PA}(Y)$ as in Schölkopf et al. [2012] is impossible because all of the causes are unobserved. Furthermore, finding a separating set as in Magliacane et al. [2018], Pearl and Bareinboim [2011] is also impossible for the same reason. Proxies can contain combinations of both stable and unstable information when they are connected to multiple $U \in \boldsymbol{U}$. Introduced in Subbaswamy and Saria [2018], "node splitting" requires knowledge of the structural equations that govern a vertex to remove unstable information from ambiguous variables, which can only be learned if the causes of the split node are observed. This requirement limits node splitting's power in the proxy setting.

[1] This assumption is not necessary but allows us to focus on more difficult questions that have not been answered by previous work.

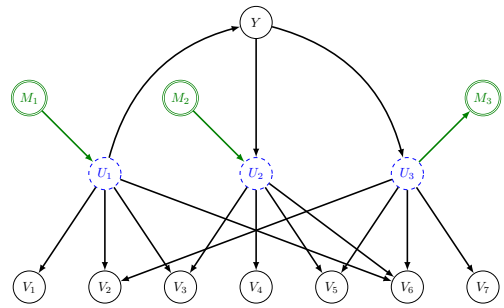

Figure 1: An example of the $\mathcal{G}^+$ considered for the paper. U_1 is a hidden cause of Y, and U_2, U_3 are hidden effects. .

4.2 INVERTIBLE DROPOUT FUNCTIONS

In order to demonstrate key concepts and intuition, we will introduce a few more simplifying assumptions about the structural equations in our causal models. For a discussion of relaxations, see Sec. 8 and for a demonstration that our method can work in real-world settings (where the assumption does not hold), see Sec.E.2.

We will study edges from A to B given by an invertible function with "dropout" noise formalized as

$$B^{(A)}(A) = \begin{cases} \mathcal{T}_{A,B}(A) & \text{with probability } \alpha \\ \phi & \text{with probability } 1-\alpha \end{cases}. \quad (1)$$

$\mathcal{T}_{A,B}(\cdot)$ is a function that is invertible, with $\mathcal{T}_{A,B}(\phi) = \phi$. The probability that information from the parent is preserved is given by $\alpha \in [0, 1]$. We will refer to $B^{(A)}(A) \neq \phi$ as "transmission," and $\alpha_{A,B}$ as the "probability of transmission."[2] ϕ, called "null", is a value that represents the dropout, or the failure of the edge to "transmit".

The structural equation for a vertex B given its parents is a deterministic function of these $B^{(A)}$,

$$B = \mathcal{T}_B(\{B^{(A)}(A) \text{ for } A \in \mathbf{PA}(B)\}), \quad (2)$$

where $\mathcal{T}_B$ is not necessarily an invertible function.

For functions with many children, we have the probability

[2] The direction of the edge for these $\alpha_{A,B}$ will sometimes be arbitrary, in which case the ordering of the vertices is unimportant.

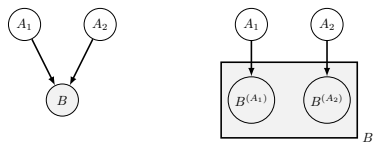

Figure 2: A diagram showing separability.

that at least one of their children transmits,

$$\alpha_{A,\mathbf{CH}(A)} := 1 - \prod_{B \in \mathbf{CH}(A)} (1 - \alpha_{A,B}). \tag{3}$$

Separability and Faithfulness If $\mathcal{T}_B$ is invertible, we say that B is a separable variable, which means that a child B with more than one parent can be split into separate disconnected vertices $B^{(A)}$ for $A \in \mathbf{PA}(B)$, each with the structural equation given by Equation 1 (See Figure 2). Separable functions make up a special violation of faithfulness in that conditioning on separable colliders no longer opens up active paths, illustrated by Lemma 1.

Lemma 1 (Separability violates faithfulness)**.** *If $U_1 \leftrightsquigarrow V \leftrightsquigarrow U_2$ and V is separable, then $U_1 \not\perp\!\!\!\perp_d U_2 \mid V$, but $U_1 \perp\!\!\!\perp U_2 \mid V$.*

The proof follows from the definition of mutual information and the fact that $U_1 \perp\!\!\!\perp U_2 \mid V$.

Our setting will rely on the assumption of faithfulness of the sub-graph on the $\boldsymbol{U} \cup \{Y\}$ vertices for proxy bootstrapping, as is the case for algorithms attempting any degree of structure learning. Specifically, we will require that any active path between two proxies V_i, V_j that does not travel through any other vertices in $\boldsymbol{V}$ must imply statistical dependence (we call this "partial faithfulness"). When we move to causal information splitting, we will allow specific violations of faithfulness that come from splittable proxies $\boldsymbol{V}$ in order to illustrate an ideal use-case of our method. This does not contradict partial faithfulness.

Transmitting Active Paths A convenient aspect of these structural equations is that α_{AB} controls the mutual information between A and its child $B^{(A)}$,

$$\begin{aligned}\mathcal{I}(A : B^{(A)}) &= \mathcal{H}(A) - \mathcal{H}(A \mid B^{(A)}) \\ &= \mathcal{H}(A) - \Pr(B^{(A)} = \phi)\, \mathcal{H}(A \mid B^{(A)} = \phi) \\ &\quad - \Pr(B^{(A)} \neq \phi)\, \mathcal{H}(A \mid B^{(A)} \neq \phi)\end{aligned}$$

An important insight is that $\mathcal{H}(A \mid B^{(A)} = \phi) = 0$ and $\mathcal{H}(A \mid B^{(A)} \neq \phi) = \mathcal{H}(A)$. Applying this gives,

$$\mathcal{I}(A : B^{(A)}) = \mathcal{H}(A) - (1 - \alpha_{A,B})\, \mathcal{H}(A) = \alpha_{A,B}\, \mathcal{H}(A).$$

This aspect generalizes to active single paths. For a length-2 path $A \to B \to C$, $\mathcal{I}(A : C) = \mathcal{I}(A : C^{(B)}) = \mathcal{H}(A) - \mathcal{H}(A \mid C^{(B)})$. Again, we can break up $\mathcal{H}(A \mid C^{(B)})$ into $\mathcal{H}(A \mid C^{(B)} = \phi) = 0$ and $\mathcal{H}(A \mid C^{(B)} \neq \phi) = \mathcal{H}(A)$. Hence, reasoning about mutual information reduces to the task of determining the probability that one of the endpoints is null. In our setup, the dropout events of different edges are independent events. Hence, $\mathcal{I}(A : C) = \alpha_{A,B}\alpha_{B,C}\, \mathcal{H}(A)$.

Conditioning adds an additional complication. Notice that transmitting active paths can "transfer" a conditioning. That is, $\mathcal{H}(A \mid x) = 0$ when there is only one active path between A and X (or X to A) and it transmits. In the next section, we will study two cases that emerge in the PBT problem: colliders and non-colliders.

5 CONTEXT SENSITIVITY

We quantify robustness through the dependence on environmental mechanisms and the label function.

Definition 1 (Context sensitivity)**.** Context sensitivity of a mechanism $M \in \boldsymbol{M}$ is defined as $\mathcal{I}(Y : M \mid \boldsymbol{X})$.

If $\boldsymbol{X}$ d-separates $\boldsymbol{M}$ from Y, the context sensitivity is 0 and training on $\boldsymbol{X}$ to predict Y yields a model that is robust across environments $\boldsymbol{M}$.

We are usually most concerned with the success of our prediction models, something that is limited by the "relevance", $\mathcal{I}(Y : \boldsymbol{X})$, of our input. More generally, we can rewrite the sensitivity in terms of the expected relevance across environments.

$$\begin{aligned}\mathcal{I}(Y : M \mid \boldsymbol{X}) &= \mathcal{I}(Y : M) - \mathcal{I}(Y : M : \boldsymbol{X}) \\ &= \mathcal{I}(Y : M) - \mathcal{I}(Y : \boldsymbol{X}) + \mathcal{I}(Y : \boldsymbol{X} \mid M).\end{aligned}$$

5.1 REDUNDANCY

Recall that in our setting we assume that all direct causes and effects are unobserved. This unobserved boundary gives rise to an invariant set $\boldsymbol{S} \subseteq \boldsymbol{U}$. We seek to identify a subset of visible proxies $\boldsymbol{X} \subseteq \boldsymbol{V}$ to extract information about $\boldsymbol{S}$.

Definition 2. For a specific U, we call $\mathcal{I}(U : \boldsymbol{X}) = \mathcal{H}(U) - \mathcal{H}(U \mid \boldsymbol{X})$ the **redundancy** between U and $\boldsymbol{X}$.

Lemma 2. *In our setting, let $\mathbf{CH}_{\boldsymbol{X}}(U) := \mathbf{CH}(U) \cap \boldsymbol{X}$.*

$$\mathcal{I}(U : \boldsymbol{X}) = \alpha_{U,\mathbf{CH}_{\boldsymbol{X}}(U)}\, \mathcal{H}(U).$$

We sometimes also refer to $\alpha_{U,\mathbf{CH}_{\boldsymbol{X}}(U)}$, the probability of transmission to at least one child, as the redundancy.

The assumptions of our setting ensure that only one potential active path exists between each $M \in \boldsymbol{M}$ and Y - hence each vertex acts as either a collider or a non-collider in the interaction of M and Y (and does not do both). We now demonstrate that redundancy has different effects on context sensitivity for these two cases.

"Good" $\boldsymbol{U}$ If M_i and Y do not form a collider at $U_i \in \boldsymbol{U}$, we say $U_i \in \boldsymbol{U}^{\text{GOOD}}$. From d-separation, we have that $M_i \perp\!\!\!\perp_d Y \mid U_i$ for all $U_i \in \boldsymbol{U}^{\text{GOOD}}$. For an example, $\boldsymbol{U}^{\text{GOOD}} = \{U_1, U_3\}$ in Figure 1. Let $\mathbf{CH}_{\boldsymbol{X}}(U_i) = \mathbf{CH}(U_i) \cap \boldsymbol{X}$.

Lemma 3 (Redundancy with $\boldsymbol{U}^{\text{GOOD}}$)**.** *For some $U_i \in \boldsymbol{U}$, if corresponding $M_i \leftrightsquigarrow U_i \leftrightsquigarrow Y$, then*

$$\mathcal{I}(M_i : Y \mid \boldsymbol{X}) = \alpha_{M_i,U_i}(1-\alpha_{U_i,\mathbf{CH}_{\boldsymbol{X}}(U_i)})\alpha_{U_i,Y}\,\mathcal{H}(M_i).$$

Analogous to including $\boldsymbol{U}^{\text{GOOD}}$ in an invariant set, a high redundancy with $\boldsymbol{X}$ is beneficial for the context sensitivity.

"Bad" $\boldsymbol{U}$ The inclusion of $\mathbf{CH}(U_i)$ in $\boldsymbol{X}$ could open up active paths via colliders of the form $M_i \to U_i \leftarrow Y$. We call the set of these variables $\boldsymbol{U}^{\text{BAD}}$. For an example, $\boldsymbol{U}^{\text{BAD}} = \{U_2\}$ in Figure 1.

Lemma 4 (Redundancy with $\boldsymbol{U}^{\text{BAD}}$)**.** *For $U_i \in \boldsymbol{U}, \boldsymbol{X} \subseteq \boldsymbol{V}$, if $M_i \leftrightsquigarrow U_i \leftrightsquigarrow Y$ then*

$$\mathcal{I}(M_i : Y \mid \boldsymbol{X}) = \alpha_{U_i,\mathbf{CH}_{\boldsymbol{X}}(U_i)}\,\mathcal{I}(M_i : Y \mid U_i)$$

Analogous to not including $\boldsymbol{U}^{\text{BAD}}$ in an invariant set, a high redundancy with $\boldsymbol{X}$ is detrimental to the context sensitivity. Similar concepts can be demonstrated via upper bounds when we relax our assumptions on the structural equations - given in Appendix Section C.

5.2 FEATURE SELECTION IMPLICATIONS

Our setup always has $\boldsymbol{X} \leftrightsquigarrow \boldsymbol{U} \leftrightsquigarrow Y$, meaning the relevance of our input is upper bounded by the redundancy in U, $\mathcal{I}(\boldsymbol{X} : Y) \leq \mathcal{I}(\boldsymbol{U} : \boldsymbol{X})$. Lemma 3 shows that, when $U \in \boldsymbol{U}^{\text{GOOD}}$, the goal of maximizing redundancy with U aligns with that of maximizing relevance. When $U \in \boldsymbol{U}^{\text{BAD}}$ in Lemma 4, the incentives clash: increasing the redundancy increases the relevance, but it also increases the context sensitivity. This suggests multiple classes of proxy variables.

Definition 3.

$$\boldsymbol{V}^{\text{GOOD}} := \mathbf{CH}(\boldsymbol{U}^{\text{GOOD}}) \setminus \mathbf{CH}(\boldsymbol{U}^{\text{BAD}}) \tag{4}$$

$$\boldsymbol{V}^{\text{BAD}} := \mathbf{CH}(\boldsymbol{U}^{\text{BAD}}) \setminus \mathbf{CH}(\boldsymbol{U}^{\text{GOOD}}) \tag{5}$$

$$\boldsymbol{V}^{\text{AMBIG}} := \mathbf{CH}(\boldsymbol{U}^{\text{BAD}}) \cap \mathbf{CH}(\boldsymbol{U}^{\text{GOOD}}) \tag{6}$$

$\boldsymbol{V}^{\text{GOOD}}$ do not have robust relationships with Y; they are not causes of Y and Subbaswamy and Saria [2018] would say to exclude $\boldsymbol{V}^{\text{GOOD}}$ from model building because they have an unstable path to Y. However, the inclusion of $\boldsymbol{V}^{\text{GOOD}}$ in $\boldsymbol{X}$ still improves the input's context sensitivity. Hence, the existence of $\boldsymbol{V}^{\text{GOOD}}$ suggests a break from the traditional fixation with only including features with fully robust information about Y.

For feature selection, an obvious strategy is to choose $\boldsymbol{X} = \boldsymbol{V}^{\text{GOOD}}$, avoid $\boldsymbol{V}^{\text{BAD}}$, and potentially try using some elements in $\boldsymbol{V}^{\text{AMBIG}}$. In the next section we will explore how we can use non-invertible functions to transform these $\boldsymbol{V}^{\text{AMBIG}}$ into $\boldsymbol{V}^{\text{GOOD}}$.

5.3 PROXY BOOTSTRAPPING

Given the robustness implications of the different classes of V, their partitioning into good, bad, and ambiguous partitions will be important. We will now demonstrate how to harness partial information to determine these partitions and classify proxies. We begin with an observation about the independence structure of the conditional probability distribution on Y.

Lemma 5 (Linking related proxies)**.** *In the given framework, if $V_i \not\perp\!\!\!\perp_d V_j \mid Y$, then either they have a shared parent ($\mathbf{PA}(V_i) \cap \mathbf{PA}(V_j) \neq \emptyset$) or they both have at least one parent that is a cause of Y (i.e. $\mathbf{PA}(V_i) \cap \mathbf{PA}(Y) \neq \emptyset$ and $\mathbf{PA}(V_j) \cap \mathbf{PA}(Y) \neq \emptyset$).*

Definition 4. For a DSD $\mathcal{G}^+ = \{\boldsymbol{V} \cup \boldsymbol{U} \cup \boldsymbol{M}, \boldsymbol{E}\}$, define the dependence graph $\mathcal{G}_Y = (\boldsymbol{V}, \boldsymbol{E}_Y)$ to be an undirected graph with edges $(V_i, V_j) \in \boldsymbol{E}_Y$ iff $V_i \not\perp\!\!\!\perp_d V_j \mid Y$.

Lemma 5 tells us that $\mathcal{G}_Y$ will have a cliques on the sets $\mathbf{CH}^{\mathcal{G}}(U)$ for $U \in \boldsymbol{U}$. Furthermore, conditioning on Y links its causes, so $\mathcal{G}_Y$ has one large clique on $\mathbf{CH}^{\mathcal{G}}(\mathbf{PA}(Y))$. This clique structure can be utilized to bootstrap partial knowledge of $\mathbf{CH}(\boldsymbol{U}^{\text{GOOD}})$ and $\mathbf{CH}(\boldsymbol{U}^{\text{BAD}})$. In this sense, "birds of a feather flock together," which can be used to bootstrap information about unknown proxies.

Lemma 6 (Information about seed proxies spreads)**.** *If $V_i \in \boldsymbol{V}^{\text{GOOD}}$ then all neighbors of $V_j \in \mathbf{NB}^{\mathcal{G}_Y}(V_i)$ are not in $\boldsymbol{V}^{\text{BAD}}$ - i.e. $V_j \in \boldsymbol{V}^{\text{GOOD}} \cap \boldsymbol{V}^{\text{AMBIG}}$. If $V_i \in \boldsymbol{V}^{\text{BAD}}$ then all neighbors of $V_j \in \mathbf{NB}^{\mathcal{G}_Y}(V_i)$ are not in $\boldsymbol{V}^{\text{GOOD}}$ - i.e. $V_j \in \boldsymbol{V}^{\text{BAD}} \cap \boldsymbol{V}^{\text{AMBIG}}$.*

Lemma 6 suggests a simple algorithm for bootstrapping the sets $\boldsymbol{V}^{\text{GOOD}}, \boldsymbol{V}^{\text{BAD}}, \boldsymbol{V}^{\text{AMBIG}}$ from a set of "seed" vertices $\boldsymbol{V}^* \subseteq \boldsymbol{V}$ with known assignments to $\boldsymbol{V}^{\text{GOOD}}, \boldsymbol{V}^{\text{BAD}}, \boldsymbol{V}^{\text{AMBIG}}$.

1. Construct $\mathcal{G}_Y$ according to Definition 4 using conditional independence tests.
2. For each $V^* \in \boldsymbol{V}^*$, if $V^* \in \boldsymbol{V}^{\text{GOOD}}$ then add a "good" label to $\mathbf{NB}(V^*)$. If $V^* \in \boldsymbol{V}^{\text{BAD}}$ then add a "bad" label to $\mathbf{NB}(V^*)$.
3. All $V \in \boldsymbol{V} \setminus \boldsymbol{V}^*$ with both "good" and "bad" labels receive an "ambigious" label instead.

Theorem 1 (Proxy bootstrapping works)**.** *Upon termination of proxy bootstrapping all vertices with a single label are correctly described if :*

(i) Partial faithfulness holds.

(ii) $\boldsymbol{V}^*$ *has at least one* $V^* \in \boldsymbol{V}^* \cap \mathbf{CH}(U)$ *for each* $U \in \boldsymbol{U}^{\text{GOOD}} \cap \mathbf{CH}(Y)$.

(iii) $\boldsymbol{V}^*$ *has at least one* $V^* \in \boldsymbol{V}^* \cap \mathbf{CH}(\mathbf{PA}(Y))$.

(iv) $\boldsymbol{V}^*$ *has at least one* $V^* \in \boldsymbol{V}^*$ *for each* $U \in \boldsymbol{U}^{\text{BAD}}$.

6 CAUSAL INFORMATION SPLITTING

This section will expand our theory into **feature engineering**, which allows us to build inputs on functions of $\boldsymbol{V}$.

Lemma 7 (Independence Inheritance)**.** *For* $V_i, V_j \in \boldsymbol{V}$ *and engineered feature* $F(V_i)$*, if for some* $\boldsymbol{C} \subseteq \boldsymbol{V} \cup \boldsymbol{U}$ *we have* $V_i \perp\!\!\!\perp V_j \mid \boldsymbol{C}$ *then we also have* $F(V_i) \perp\!\!\!\perp V_j \mid \boldsymbol{C}$*.*

Lemma 7 follows directly from the DPI.

In the feature engineering setting, the equality in Lemma 3 becomes an upper bound by the DPI

$$\mathcal{I}(U_i : \boldsymbol{X}) \leq \mathcal{I}(U_i : F(\boldsymbol{X}))$$

for some function $F(\cdot)$). Let $\tilde{\mathbf{CH}}_{\boldsymbol{X}}(U_i)$ be the children or functions of children of U_i in $\boldsymbol{X}$.

Lemma 8 (Engineering redundancy for $\boldsymbol{U}^{\text{GOOD}}$)**.** *For some* $U_i \in \boldsymbol{U}$*, if* $M_i \circ\!\!\leftrightarrow U_i \leftrightarrow\!\!\circ Y$ *then*

$$\mathcal{I}(M_i : Y \mid \boldsymbol{X}) \leq \alpha_{M_i,U_i} \mathcal{H}(U_i \mid \tilde{\mathbf{CH}}(\boldsymbol{X}))\alpha_{U_i,Y}.$$

This bound is tight when features are engineered with invertible functions.

6.1 SPLITTABLE AMBIGUOUS PROXIES

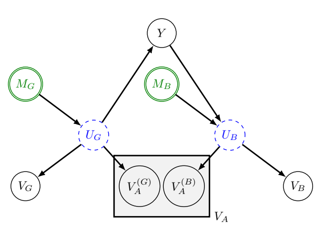

Figure 3: $V_G \in \boldsymbol{V}^{\text{GOOD}}$, $V_B \in \boldsymbol{V}^{\text{BAD}}$. $V_A \in \boldsymbol{V}^{\text{AMBIG}}$, but V_A is actually a linear transformation of two components, $V_A^{(G)}, V_A^{(B)}$, which are good and bad respectively.

Consider the setup in Figure 3, where $V_G \in \boldsymbol{V}^{\text{GOOD}}$, $V_B \in \boldsymbol{V}^{\text{BAD}}$, and $V_A \in \boldsymbol{V}^{\text{AMBIG}}$. V_A is splittable, and hence generated by invertible $\mathcal{T}_A$.[3] Splitting V_A into components allows us to understand where its ambiguity comes from. We will call these features **splittable ambiguous proxies** **(SAPs)**. This section will develop a technique to isolate the good information from ambiguous proxies without paying the penalty for their bad information.

[3] While we may still be able to gain useful information from non-splittable proxies, the tradeoffs are difficult to quantify and hence beyond the scope of this paper.

Definition 5. We define an **isolation function** of V_i on V_A

$$\begin{aligned} F_{\text{ISO}(V_i)}(V_A) &:= \arg\min_F \mathcal{H}(F(V_A)) \\ \text{such that } &\mathcal{I}(F(V_A) : V_i) = \mathcal{I}(V_A : V_i) \end{aligned} \quad (7)$$

Note that isolation sufficient statistic for V_i [Cover, 1999]. Isolation involves maintaining the information about V_i while removing all possible excess information.

6.2 TRANSPORTABILITY GUARANTEES

An optimal isolation function would be $F_{\text{ISO}(\boldsymbol{U}^{\text{GOOD}} \cap \mathbf{PA}(V_A))}(V_A)$. Again, the primary challenge is that we do not have access to $\boldsymbol{U}^{\text{BAD}}$ and $\boldsymbol{U}^{\text{GOOD}}$. Fortunately, the relationships between the visible proxies give insight into these hidden U.

Lemma 9 (Unobserved common-cause information)**.** *Given a causal DAG* $\mathcal{G} = \{\boldsymbol{V} \cup \boldsymbol{U}, \boldsymbol{E}\}$ *and an observed subgraph* $\mathcal{G}' = \{\boldsymbol{V}, \boldsymbol{E}'\}$*, for any* $U \in \boldsymbol{U}$ *and* $V_i, V_j \in \boldsymbol{V}$ *that satisfy, (i)* $V_i \perp\!\!\!\perp_d^{\mathcal{G}'} V_j$*, (ii)* $V_i \not\perp\!\!\!\perp_d^{\mathcal{G}} V_j$*, and (iii)* $V_i \circ\!\!\leftrightarrow U \leftrightarrow\!\!\circ V_j$*, then* $\mathcal{I}(V_i, V_j : U) \geq \mathcal{I}(V_i : V_j)$*.*

We can lower bound the degree to which $F_{\text{ISO}(V_G)}(U_A)$ approximates good information in $U_A^{(G)}$ using Lemma 9,

$$\mathcal{I}((V_i, F_{\text{ISO}(V_i)}(V_A)) : \mathbf{PA}(V_i)) \geq \mathcal{I}(V_A : V_i). \quad (8)$$

The key advantage of isolation functions is that applying them to SAPs forces their outputs to have the independence properties of the isolated vertex. Recall the structure of the conditional dependence graph $\mathcal{G}_Y$ in Definition 4: if $V_G \in \boldsymbol{V}^{\text{GOOD}}$, then $\mathbf{PA}(V_G) \in \boldsymbol{U}^{\text{GOOD}}$ and $U_B \perp\!\!\!\perp V_G$ for every $U_B \in \boldsymbol{U}^{\text{BAD}}$. This suggests that V_G is a perfect candidate for isolation because including $F_{\text{ISO}(V_B)}(V_A)$ would ensure redundancy with $\mathbf{PA}(V_i) \subset \boldsymbol{U}^{\text{GOOD}}$ while not increasing redundancy with $\boldsymbol{U}_B$!

Lemma 10 (Isolating $\boldsymbol{V}^{\text{GOOD}}$ behaves like $\boldsymbol{V}^{\text{GOOD}}$)**.** *For* $V_G \in \boldsymbol{V}^{\text{GOOD}}$ *and* $U_B \in \boldsymbol{U}^{\text{BAD}}$ *and* $F_{\text{ISO}(V_G)}(V_A)$ *defined on the distribution conditioned on* Y*,*

$$\mathcal{I}(U_B : F_{\text{ISO}(V_G)}(V_A) \mid Y) = 0. \quad (9)$$

In the previous section, the independence relationships between good and bad proxies allowed us to bootstrap additional good and bad vertices. We can now these relationships to engineer good proxies from ambiguous ones.

Theorem 2 (CIS costs and benefits)**.** *Consider comparing* $\boldsymbol{X} = \{V_G\}$ *to* $\boldsymbol{X}^+ = \{V_G, F_{\text{ISO}(V_G)}(V_A))\}$ *for*

$V_G \in \boldsymbol{V}^{\text{GOOD}}$ and $V_A \in \boldsymbol{V}^{\text{AMBIG}}$ where V_A is a SAP. Also consider isolation functions defined on the conditional distribution on Y. We claim that including the output of this isolation function does not worsen any context sensitivity, i.e. for all $M \in \boldsymbol{M}$

$$\mathcal{I}(M : Y \mid \boldsymbol{X}^+) \leq \mathcal{I}(M : Y \mid \boldsymbol{X}). \tag{10}$$

Furthermore, if $\mathcal{I}(F_{\text{ISO}(V_G)}(V_A) : V_G) < \mathcal{I}(V_A : V_G \mid Y)$, then the relevance improves:

$$\mathcal{I}(Y : \boldsymbol{X}^+) > \mathcal{I}(Y : \boldsymbol{X}). \tag{11}$$

Theorem 2 tells us that using an isolation function helps when the function is more predictive of the isolated variable in the post-selected Y distribution than it is in the full distribution. This condition is sufficient but loose because it does not take into account the direct $\mathcal{I}(Y : F_{\text{ISO}(V_G)}(V_A))$ term (for which we have no guaranteed bounds). The proof relies on Lemma 8, see Appendix Sec. D.

6.3 AUXILIARY TRAINING TASKS

In the infinite sample regime, consider an "optimal" model $F(\cdot)$ that predicts V_i using input V_A. Optimal models should utilize all of the information available for prediction in their inputs, meaning $I(F(V_A) : V_i) = \mathcal{I}(V_A : V_i)$. In addition, training tasks often try to find the simplest prediction function that maintains the accuracy - a task often referred to as regularization. This corresponds information theoretically to minimizing $\mathcal{H}(F_{V_i}(V_A))$.

An optimal model with sufficient regularization can therefore be thought of as an isolation function. Hence, we can get an approximate isolation function, $\tilde{F}_{\text{ISO}(V_i)}(V_{\text{SAP}})$, by training a model to predict V_i using input V_{SAP}.

The conditions for guaranteed success also have a nice interpretation. $\mathcal{I}(\tilde{F}_{\text{ISO}(V_G)}(V_A) : V_G) < \mathcal{I}(V_A : V_G \mid Y)$ means that the accuracy of a V_G predictor using input V_A trained on post-conditioned Y must degrade when moving to the full dataset. More precisely, the conditions for improvement now translate to

$$\begin{aligned} &\min_F \mathbb{E}[\text{Error}(F(V_A), V_G)] \\ &\quad < \sum_y \Pr(y) \min_F (\mathbb{E}[\text{Error}(F(V_A), V_G) \mid y]), \end{aligned} \tag{12}$$

which can easily be checked on our training data.

6.4 SUGGESTED OVERALL PROCEDURE

We propose the following procedure for building robust (low context-sensitivity) models in the PBT problem.

1. Split the data into partitions based on Y and study the independence of every pair of variables in these partitions to determine cliques of dependence.
2. Using domain knowledge, identify seeds in $\boldsymbol{V}^{\text{GOOD}}, \boldsymbol{V}^{\text{BAD}}$ for proxy bootstrapping (Sec.5.3).
3. Perform CIS on $\boldsymbol{V}^{\text{AMBIG}}$ (see Sec. 6.3).
4. Build a prediction model for Y using $\boldsymbol{V}^{\text{GOOD}}$ and the CIS-engineered $\boldsymbol{V}^{\text{AMBIG}}$.

7 EXPERIMENTS

We will now demonstrate the effectiveness of these methods on synthetic and real world data.

7.1 SYNTHETIC EXPERIMENTS

We generate data for the DAG in Figure 3 based on normal distributions, see details of the setup in the Appendix E.1. We vary the standard deviations of normally distributed M_G and M_B. The training data is drawn from $\sigma(M_G) = \sigma(M_B) = 1$, while the testing data varies both quantities and thus the influence of the context. We measure the accuracy of our feature engineering based on CIS, $\hat{Y}^{(3)}(V_G, \tilde{F}_{\text{ISO}(V_G)}(V_A))$, that utilizes the auxiliary task approximation to isolate V_A's predictive information about V_G. We compare it to $\hat{Y}^{(1)}(V_G)$ trained on only $\boldsymbol{V}^{\text{GOOD}}$ and $\hat{Y}^{(2)}(V_G, V_A)$ trained on $\boldsymbol{V}^{\text{GOOD}} \cup \boldsymbol{V}^{\text{AMBIG}}$. For a theoretical limit of CIS we also compare to $\hat{Y}^{(4)}(V_G, V_A^{(G)})$ although access to $V_A^{(G)}$ is usually not possible.

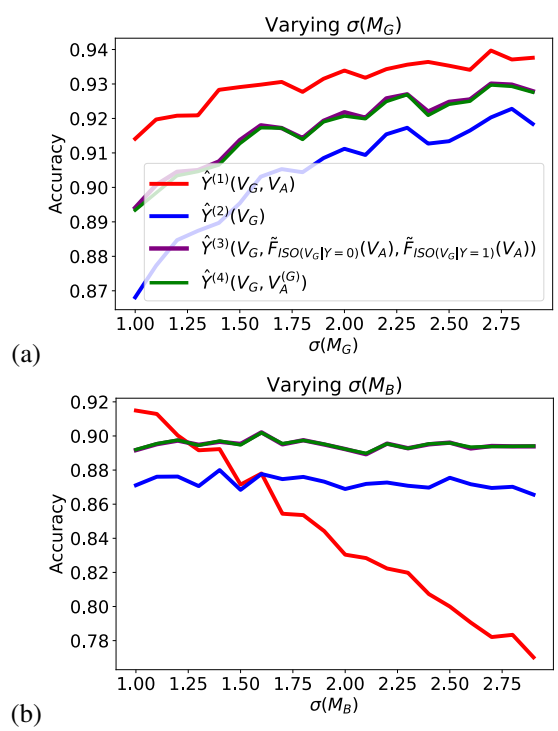

Figure 4: Results from our experiments on synthetic data.

Results When comparing feature selection approaches, we observe in Figure 4 that including V_A results in higher accuracy of $\hat{Y}^{(1)}$ over $\hat{Y}^{(2)}$ when the shift acts on $\boldsymbol{U}^{\text{GOOD}}$ (a) or is small for $\boldsymbol{U}^{\text{BAD}}$ (b). However, the accuracy of $\hat{Y}^{(2)}$ deteriorates with bigger shifts in $\boldsymbol{U}^{\text{BAD}}$.

Our proposed method based on causal information splitting offers a middle ground. $\hat{Y}^{(3)}$ is able to maintain the same robustness as $\hat{Y}^{(2)}$ while taking advantage of some of the gains enjoyed by $\hat{Y}^{(1)}$ in (a). In fact, $\hat{Y}^{(3)}$ performs very similarly to $\hat{Y}^{(4)}$, which had a-priori knowledge of the SAP components and used only $V_A^{(G)}$. These improvements were achieved despite not meeting the sufficient condition for increasing relevance in Theorem 2.

7.2 REAL EXPERIMENTS

We use US Census data processed through folktables Ding et al. [2021] to predict whether the income of a person exceeds 50k following Dua and Graff [2017]. We use 2019 pre-pandemic data for training and 2021 data during the pandemic for evaluation, skipping the experimental release of 2020. We select a small feature set to see a starker effect of including/excluding individual features. We consider commute time (coded as JWMNP in the dataset), a flag whether the person received Medicaid, Medical Assistance, or any kind of government-assistance plan for those with low incomes or a disability (coded as HINS4) and education level (SCHL). We purposefully selected a small feature set to see a starker effect of including/excluding individual features. They include a feature with relatively stable predictive power (education level) and two features heavily affected by the pandemic through increased work-from-home and medicaid's continuous enrollment provision.

Our auxiliary task from Sec. 7, referred to as engineered features, uses HINS4 and JWMNP indirectly to predict education conditioned on the label. The actual education-level with its two predictions (for high and low income) serve as input features. We compare it to using all three features directly, all features, or using just the stable education feature limited features. We use logistic regression from sklearn with l1 regularization to build models based on the different feature sets that the three methods created. l1 regularization yielded better generalization than l2 regularization.

Results Table 1 shows a mixed picture of the accuracies for the three tested methods across states. Using all features leads to the best in-domain performance (see Table 2 in the Appendix E.2), but not necessarily the best out-of-domain performance. How much dropping the ambiguous features hurts predictive power but helps with robustness varies across the states: For some states it leads to a overall higher accuracy on the 2021 data compared to using all features while for others using all features is better. Our proposed feature engineering using CIS has the best mean out-of-domain accuracy of 0.698 across states compared to these two baselines. It also achieves close to the best out-of-domain accuracy for 6 out of 9 states.

state	all features	engineered features	limited features
CA	0.712 ± 0.0011	0.711 ± 0.0014	0.692 ± 0.0014
FL	0.683 ± 0.0012	0.678 ± 0.0018	0.68 ± 0.0013
GA	0.689 ± 0.0025	0.707 ± 0.0055	0.709 ± 0.0029
IL	0.662 ± 0.0026	0.689 ± 0.0033	0.684 ± 0.0019
NY	0.707 ± 0.0022	0.702 ± 0.0025	0.687 ± 0.008
NC	0.691 ± 0.0031	0.684 ± 0.0034	0.683 ± 0.003
OH	0.689 ± 0.0022	0.703 ± 0.004	0.696 ± 0.0029
PA	0.672 ± 0.0017	0.695 ± 0.0023	0.688 ± 0.0022
TX	0.69 ± 0.0029	0.712 ± 0.0028	0.712 ± 0.0027
avg	0.688	0.698	0.692

Table 1: Out-of-domain accuracy on the 2021 American Community Survey for the largest states in the U.S. and the average across the states. Mean of the held-out data from 10 random data splits with standard deviation.

8 CONCLUSION

In this paper we studied the challenging problem of building robust models that are robust to distribution shift when causes and effects of the target variable are unmeasured. Among the observed noisy proxies, we showed how to perform feature selection based on conditional independence tests and knowledge about some seed nodes.

After bootstrapping, we often have a significant number of ambiguous proxies, which have components that are both helful and hurtful to our model's robustness. Through feature engineering, however, we showed how to isolate robust predictive power from these ambiguous proxies using auxiliary learning tasks. We proved that these engineered features do not reduce robustness but can help with accuracy. In our experiments on real census data under shifts due to the pandemic, we showed the engineered features provided benefits for most states over using the ambiguous features directly or completely ignoring them. While our theoretical framework is involved, these experiments demonstrate improvements outside of our assumptions.

We hope that our information theoretic perspective can be applied to other settings to quantify robustness. In future work, we seek to relax our stringent sparsity assumptions by building on ideas from mixtures of DAG structures like Gordon et al. [2021]. In addition, the notion of separability and its relationship to inevitability in structural equation models appears to be a key in understanding when information can be successfully split and filtered.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. arXiv preprint arXiv:1907.02893, 2019.

Alexis Bellot and Mihaela van der Schaar. Generalization and invariances in the presence of unobserved confounding. arXiv preprint arXiv:2007.10653, 4, 2020.

Thomas M Cover. Elements of information theory. John Wiley & Sons, 1999.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. 2020.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. Advances in Neural Information Processing Systems, 34, 2021.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Spencer L Gordon, Bijan Mazaheri, Yuval Rabani, and Leonard J Schulman. Identifying mixtures of bayesian network distributions. arXiv preprint arXiv:2112.11602, 2021.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=lQdXeXDoWtI.

Christina Heinze-Deml, Jonas Peters, and Nicolai Meinshausen. Invariant causal prediction for nonlinear models. Journal of Causal Inference, 6(2), 2018.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In International Conference on Machine Learning, pages 5815–5826. PMLR, 2021.

Sara Magliacane, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M. Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, page 10869–10879, Red Hook, NY, USA, 2018. Curran Associates Inc.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In International conference on machine learning, pages 10–18. PMLR, 2013.

Judea Pearl. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan kaufmann, 1988.

Judea Pearl. Causality. Cambridge university press, 2009.

Judea Pearl and Elias Bareinboim. Transportability of causal and statistical relations: A formal approach. In Twenty-fifth AAAI conference on artificial intelligence, 2011.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. Journal of the Royal Statistical Society. Series B (Statistical Methodology), 78(5):947–1012, 2016. ISSN 13697412, 14679868. URL http://www.jstor.org/stable/44682904.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Elements of causal inference: foundations and learning algorithms. The MIT Press, 2017.

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. Dataset shift in machine learning. Mit Press, 2008.

Francesco Quinzan, Cecilia Casolo, Krikamol Muandet, Niki Kilbertus, and Yucen Luo. Learning counterfactually invariant predictors, 2022. URL https://arxiv.org/abs/2207.09768.

Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. The Journal of Machine Learning Research, 19 (1):1309–1342, 2018.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. arXiv preprint arXiv:1911.08731, 2019.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. arXiv preprint arXiv:1206.6471, 2012.

Gabriele Schweikert, Gunnar Rätsch, Christian Widmer, and Bernhard Schölkopf. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. Advances in neural information processing systems, 21, 2008.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of statistical planning and inference, 90(2):227–244, 2000.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. In International Conference on Learning

Representations, 2018. URL https://openreview.net/forum?id=Hk6kPgZA-.

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. Causation, prediction, and search. MIT press, 2000.

Amos Storkey et al. When training and test sets are different: characterizing learning transfer. Dataset shift in machine learning, 30:3–28, 2009.

Adarsh Subbaswamy and Suchi Saria. Counterfactual normalization: Proactively addressing dataset shift using causal mechanisms. In UAI, pages 947–957, 2018.

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations in text classification. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, 2021. URL https://openreview.net/forum?id=BdKxQp0iBi8.

John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. PLOS Medicine, 15(11):1–17, 11 2018. doi: 10.1371/journal.pmed.1002683. URL https://doi.org/10.1371/journal.pmed.1002683.