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

Bijan Mazaheri[1] **Atalanti Mastakouri**[2] **Dominik Janzing**[2] **Michaela Hardt**[2]

[1]California Institute of Technology, Pasadena, CA, USA
[2]Amazon Causality Lab, Tübingen, Germany

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 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] and 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 capture shifting information from a collection of datasets [Rojas-Carulla et al., 2018, Magliacane et al., 2018]. Such techniques require access to a comprehensive set of datasets that represent all possible shiftings. A causal perspective developed in Storkey et al. [2009] and Pearl and Bareinboim [2011] instead uses graphical modeling via **selection diagrams** to model shifting mechanisms. This approach requires access to multiple datasets to learn these mechanisms, but does not require that those datasets span the entire space of possible shifting. Such approaches also allow the use of domain expert knowledge when building selection diagrams. A detailed comparison of stability in the causal and anticausal scenario is given in 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 abstract concepts that cannot be easily 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

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

for an example. All of these proxies are descendants of $\boldsymbol{U}$ – a case which is common in medicine, where the measured variables are often blood markers (or other tests) that are indicative of an underlying condition.

The proxy setting is difficult to address in standard transportability framework. While previous approaches to partially observed systems suggest restricting model inputs to those on stable paths [Subbaswamy and Saria, 2018], no observed proxies satisfy this condition in our setting. That is, even if probability of Y given its unobserved causes is invariant, the probability of Y given the observed proxies may vary, along with the marginal probability of those proxies.

We will use concepts from causal inference and information theory to define and study the **proxy-based transportability** problem. Our framework will demonstrate that perfection is indeed the enemy of good – some variables (although with an unstable relationship to the target) should still be included as features to build a model with improved stability.

A primary goal of this paper will be to distinguish between proxies that are "helpful" or "hurtful" for stability - a property that they inherit from the causal and anti-causal variables whose information they contain. The stability of these unobserved variables depends on the transportability of their causal structure, which is unobserved. We will present a strategy for feature selection based on properties that propagate from the underlying causal structure to its observed proxies. Specifically, we 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 they will improve or worsen the model's transportability. Inspired by node splitting [Subbaswamy and Saria, 2018], we introduce a method we call **causal information splitting (CIS)**, which can improve stability 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. While theoretical guarantees require a number of assumptions, we demonstrate the surprising ability of CIS to separate stabilizing information from ambiguous variables on synthetic data experiments with relaxed assumptions. 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 the 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 proactively modeling shifts, work on invariant risk minimization [Arjovsky et al., 2019, Bellot and van der Schaar, 2020] has approached this problem when given access to the shifted data on which the models will be used. 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., 2016b, Heinze-Deml et al., 2018, Peters et al., 2016a].

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.

We do not address the tradeoffs associated with robustness and model accuracy in this paper. Such tradeoffs are a natural consequence of restricting the input information for our model, since unstable information is still useful in unperturbed cases. This problem is generally addressed by Oberst et al. [2021] via regularization.

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 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,

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

$\mathbf{PA}^{\mathcal{G}}(V), \mathbf{CH}^{\mathcal{G}}(V)$ denote the parents and children of V in $\mathcal{G}$. Following the uppercase/lowercase convention, $\mathbf{pa}_{\boldsymbol{v}}(V)$

is an assignment to $\mathbf{PA}(V)$ using the values in $\boldsymbol{v}$.[1] $\mathbf{DE}^{\mathcal{G}}(V)$ 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."

We will rely on the concepts of d**-separation** and **active paths** to discuss the independence properties of Bayesian networks, which are discussed in Appendix 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 sets of variables that act as "switches" for d-separation. $\boldsymbol{A} \multimap\!\!\!\!\multimap \boldsymbol{C} \multimap\!\!\!\!\multimap \boldsymbol{B}$ 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 $\boldsymbol{A} \multimap\!\!\!\!\multimap \boldsymbol{C} \multimap\!\!\!\!\multimap \boldsymbol{B}$ 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), $\mathcal{G}^+ = (\boldsymbol{V} \cup \boldsymbol{U} \cup \boldsymbol{M}, \boldsymbol{E} \cup \boldsymbol{E_M})$, 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

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.[2] We are given access to a list of "visible proxy variables" $\boldsymbol{V} \setminus \{Y\}$ which are descendants 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 $U_i \neq U_j \in \boldsymbol{U}$. Systemic sparsity guarantees that a discoverable causal structure exists within the unobserved variables and simplifies the interactions between the proxies.

We will build our theory on 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., 2016b].

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 set of features $\boldsymbol{X}$ that 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**. To allow for feature engineering, we define these features to be the output of a function, $\boldsymbol{X} = F(\boldsymbol{V} \setminus \{Y\})$ which can capture higher-level representations of $\boldsymbol{V} \setminus \{Y\}$.

Challenges in 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.

4.1 INVERTIBLE DROPOUT FUNCTIONS

We will demonstrate the failure of existing transportability approaches in this setting using a counterexample built on structural equations models with cleanly interpretable entropic relationships. This construction will show the cost of restricting features to those with stable paths to the prediction variable Y, and serve as a framework for understanding the problem in general. 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.7.

Our restricted structural equations give edges from A to B

[1] $\mathbf{PA}(V) \subseteq \boldsymbol{V}$

[2] This assumption is not necessary but allows us to focus on more difficult questions that have not been answered by previous work. Namely, direct causes and effects can be visible or have perfect proxies without changing the results of the paper.

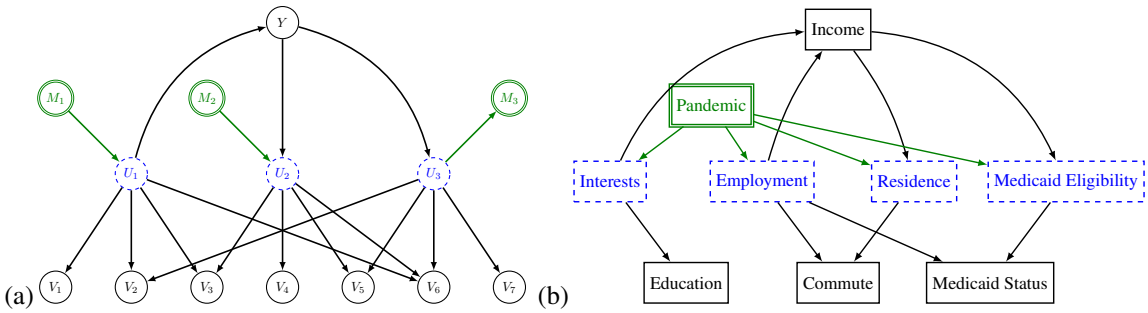

Figure 1: Examples of the $\mathcal{G}^+$ considered for the paper. (a) shows a generic setup where U_1 is a hidden cause of Y, and U_2, U_3 are hidden effects. (b) shows a *plausible* model explaining the success of our real-data experiment in Section 7.2.

described by an invertible function with "dropout" noise,

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

$\mathcal{T}_{A,B}(\cdot)$ 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 $\alpha_{A,B} \in [0,1]$. We will refer to $B^{(A)}(A) \neq \phi$ as "transmission," and $\alpha_{A,B}$ as the "probability of transmission."[3] 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, the probability that at least one of their children transmits is

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

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 variables 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* $U_1 \rightsquigarrow V \leftsquigarrow U_2$ *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 $U_1 \perp\!\!\!\perp U_2 \mid V$.

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

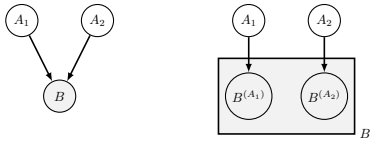

Figure 2: A diagram showing separability.

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 separable 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 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. This concept is related to context sensitivity, and 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 set of parents gives rise to an invariant set $\boldsymbol{S} \subseteq \boldsymbol{U}$[4]. 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 the dropout function setting, let* $\mathbf{CH}_{\boldsymbol{X}}(U) := \mathbf{CH}(U) \cap \boldsymbol{X}$.

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

Redundancy in the dropout function setting is controlled by our choice of $\boldsymbol{X}$ via $\alpha_{U,\mathbf{CH}_{\boldsymbol{X}}(U)}$, the probability of transmission to at least one child.

Our graphical assumptions 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 with stable (non-collider) variables generally improves our context sensitivity, whereas redundancy with unstable (collider) variables worsens it.

[4]The Markov boundary of Y would also give an invariant set, but could include vertices in $\boldsymbol{M}$ that are parents of effects of Y.

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

Lemma 3 (Redundancy with $\boldsymbol{U}^{\mathrm{GOOD}}$)**.** *In the dropout function setting, for some $U_i \in \boldsymbol{U}$, if corresponding $M_i \multimap U_i \multimap Y$, then*

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

Lemma 3 comes from multiplying the probability of transmission of each edge along the path M_i, U_i, Y. We also pick up a term requiring that the $U_i, \boldsymbol{X}$ edges do not transmit, in which case conditioning on $\boldsymbol{X}$ would reduce the entropy of U to nothing and close off the path.

"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 $\boldsymbol{U}^{\mathrm{BAD}}$. For an example, $\boldsymbol{U}^{\mathrm{BAD}} = \{U_2\}$ in Figure 1.

Lemma 4 (Redundancy with $\boldsymbol{U}^{\mathrm{BAD}}$)**.** *In the dropout function setting, $U_i \in \boldsymbol{U}, \boldsymbol{X} \subseteq \boldsymbol{V}$, if $M_i \multimap U_i \multimap Y$ then*

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

Lemma 4 demonstrates that there are still proxies for which inclusion hurts our model's robustness. Similar concepts can be demonstrated via upper bounds when we allow arbitrary sets of structural equations - given in Appendix C. Optimizing these upper bounds does not give a guarantee of optimality, but can still point towards a general improvement.

5.2 FEATURE SELECTION IMPLICATIONS

The proxy graphical setup requires $\boldsymbol{X} \multimap \boldsymbol{U} \multimap Y$, meaning the relevance of our input is upper bounded by the redundancy with U, $\mathcal{I}(\boldsymbol{X} : Y) \leq \mathcal{I}(\boldsymbol{U} : \boldsymbol{X})$.

Lemma 3 shows that proxies of $\boldsymbol{U}^{\mathrm{GOOD}}$ help build accurate and universal models, while Lemma 4 shows that proxies of $\boldsymbol{U}^{\mathrm{BAD}}$ can trade universality for domain-specific accuracy. Of course, proxies need not lie neatly in these two classes - many proxies contain a combination of universally-relevant and domain-relevant features. This suggests multiple classes of proxy variables.

Definition 3.

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

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

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

The behavior of $\boldsymbol{V}^{\mathrm{GOOD}}$ in the dropout function setting shows how restricting models to invariant features fails; a high redundancy with $\boldsymbol{U}^{\mathrm{GOOD}}$ is beneficial for the context sensitivity even though the paths from the proxies are unstable. Inclusion of $\boldsymbol{V}^{\mathrm{GOOD}}$ in $\boldsymbol{X}$ improves context sensitivity even though $\boldsymbol{V}^{\mathrm{GOOD}}$ is not made up of direct causes (as suggested by Schölkopf et al. [2012]) or invariant features (as suggested by Magliacane et al. [2018] and [Subbaswamy and Saria, 2018]).

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

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. This step is optional if the role of each proxy is already understood (as is the case when the DAG is known). The results in this subsection will only require the graphical assumptions of the PBT setting - i.e. systemic sparsity, partial faithfulness, and an independent shifting mechanism M_i for each $U_i \in \boldsymbol{U}$.

We begin with an observation about the independence structure of the conditional probability distribution on Y.

Lemma 5 (Linking related proxies)**.** *Within the graphical constraints of PBT, 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 clique 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 enhance partial knowledge of $\mathbf{CH}(\boldsymbol{U}^{\mathrm{GOOD}})$ and $\mathbf{CH}(\boldsymbol{U}^{\mathrm{BAD}})$. In this sense, "birds of a feather flock together" – information about each clique's proxies can be a determined from understanding a single member of that clique.

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

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

1. Construct $\mathcal{G}_Y$ according to Definition 4 using conditional independence tests.
2. For each $V^* \in \boldsymbol{V}^*$, if $V^* \in \boldsymbol{V}^{\mathrm{GOOD}}$ then add a "good" label to $\mathbf{NB}(V^*)$. If $V^* \in \boldsymbol{V}^{\mathrm{BAD}}$ 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 :*

1. *Partial faithfulness holds.*
2. *$\boldsymbol{V}^*$ has at least one $V^* \in \boldsymbol{V}^* \cap \mathbf{CH}(U)$ for each $U \in \boldsymbol{U}^{\mathrm{GOOD}} \cap \mathbf{CH}(Y)$.*
3. *$\boldsymbol{V}^*$ has at least one $V^* \in \boldsymbol{V}^* \cap \mathbf{CH}(\mathbf{PA}(Y))$.*
4. *$\boldsymbol{V}^*$ has at least one $V^* \in \boldsymbol{V}^*$ for each $U \in \boldsymbol{U}^{\mathrm{BAD}}$.*

6 CAUSAL INFORMATION SPLITTING

This section will expand our theory into **feature engineering**, which allows us to build inputs on functions of $\boldsymbol{V}$. A main takeaway from Section 5 was that we should build models using proxies for $\boldsymbol{U}^{\mathrm{GOOD}}$ and avoid using features that are proxies for $\boldsymbol{U}^{\mathrm{BAD}}$. The extension of this to engineered features is to build a model on functions of proxies for which the output of those functions is related to $\boldsymbol{U}^{\mathrm{GOOD}}$ and not related to $\boldsymbol{U}^{\mathrm{BAD}}$. We present two lemmas to formalize this notion.

Let $\widetilde{\mathbf{CH}}_{\boldsymbol{X}}(U_i)$ be the children or functions of children of U_i in $\boldsymbol{X}$. Lemma 7 shows that building models with more redundancy with $\boldsymbol{U}^{\mathrm{GOOD}}$ (i.e. lower $\mathcal{H}(U_i \mid \widetilde{\mathbf{CH}}_{\boldsymbol{X}}(U_i))$ improves our context sensitivity in the dropout function setting.[5]

Lemma 7 (Engineering redundancy for $\boldsymbol{U}^{\mathrm{GOOD}}$)**.** *In the dropout function setting, if $U_i \in \boldsymbol{U}^{\mathrm{GOOD}}$ then*

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

Of course, even good proxies are related to $\boldsymbol{U}^{\mathrm{BAD}}$ through their connection to Y, so $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{U}^{\mathrm{BAD}}$ is impossible. Instead, Lemma 8 tells us that if we avoid redundancy with $\boldsymbol{U}^{\mathrm{BAD}}$ after conditioning on Y, we do not pick up any context sensitivity from the associated shifting mechanisms.

Lemma 8 (Avoiding redundancy with $\boldsymbol{U}^{\mathrm{BAD}}$)**.** *For some $U_i \in \boldsymbol{U}^{\mathrm{BAD}}$, if we maintain $\mathcal{I}(U_i : \boldsymbol{X} \mid Y) = 0$, then $\mathcal{I}(M_i : Y \mid \boldsymbol{X}) = 0$.*

[5] Appendix C shows that redundancy with $\boldsymbol{U}^{\mathrm{GOOD}}$ lowers an upper bound on context sensitivity in more general cases

Recall that ambiguous proxies contain information about both $\boldsymbol{U}^{\text{GOOD}}$ and $\boldsymbol{U}^{\text{BAD}}$. The inclusion of an ambiguous proxy V_A improves context sensitivity because of its redundancy with $\boldsymbol{U}^{\text{GOOD}}$ via Lemma 7. This section will develop a technique for filtering V_A into $F(V_A)$, which will satisfy the conditions in Lemma 8. To do this, we will require separability.

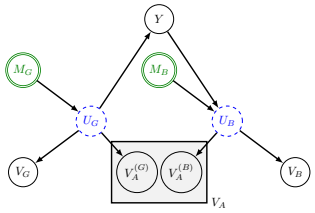

Figure 3: $V_G \in \boldsymbol{V}^{\text{GOOD}}$, $V_B \in \boldsymbol{V}^{\text{BAD}}$. $V_A \in \boldsymbol{V}^{\text{AMBIG}}$ is a linear transformation of two components, $V_A^{(G)}, V_A^{(B)}$, which are good and bad respectively.

Separable Ambigious Proxies 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 generated by invertible $\mathcal{T}_A$, making it a **separable ambiguous proxy (SAP)**.[6] Splitting V_A into components allows us to isolate the origins of its ambiguity - the mixing of good information from $V_A^{(G)}$ and bad information from $V_A^{(B)}$.

6.1 ISOLATION FUNCTIONS

We would like to isolate $V_A^{(G)}$ from V_A to avoid paying the penalty for $V_A^{(B)}$. We will do this using **isolation functions**.

Definition 5. We define an **isolation function** of V_i on V_A, with optional conditioning on y, to be

$$F_{\text{ISO}(V_i)}(V_A \mid y) := \arg\min_F \mathcal{H}(F(V_A \mid y)) \quad \text{such that } \mathcal{I}(F(V_A) : V_i \mid y) = \mathcal{I}(V_A : V_i \mid y). \tag{7}$$

$F_{\text{ISO}(V_i)}(V_A \mid Y)$ gives a vector of functions with an entry for each $y \in Y$.

Note that isolation functions are sufficient statistics for V_i [Cover, 1999]. Isolation involves maintaining the information about V_i while removing excess noise.

Recall from Lemma 8 that in order to avoid worsening context sensitivity, we want to ensure $\mathcal{I}(F(V_A) : \boldsymbol{U}^{\text{BAD}} \mid Y) = 0$. Isolation functions on SAPs are well designed for this purpose, because they enforce the independence properties of the isolated vertex on their outputs. In order to achieve $\mathcal{I}(F(V_A) : \boldsymbol{U}^{\text{BAD}} \mid Y) = 0$ while preserving as much information about $\boldsymbol{U}^{\text{GOOD}}$ as possible, an optimal isolation function would be to isolate $\boldsymbol{U}^{\text{GOOD}}$ using $F_{\text{ISO}(\boldsymbol{U}^{\text{GOOD}})}(V_A \mid Y)$.

Of course, we do not have access to $\boldsymbol{U}^{\text{GOOD}}$, so our next best option is to isolate $\boldsymbol{V}^{\text{GOOD}}$ using $F_{\text{ISO}(\boldsymbol{V}^{\text{GOOD}})}(V_A \mid Y)$, since $\boldsymbol{U}^{\text{BAD}} \perp\!\!\!\perp \boldsymbol{V}^{\text{GOOD}} \mid Y$. Lemma 9 shows that the output of $F_{\text{ISO}(V_G)}(V_A \mid Y)$ behaves like a good proxy if $V_G \in \boldsymbol{V}^{\text{GOOD}}$ and V_A is a SAP.

Lemma 9 (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 an isolation function $F_{\text{ISO}(V_G)}(V_A \mid Y)$,*

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

The benefit from $F_{\text{ISO}(V_G)}(V_A \mid Y)$'s information about $\boldsymbol{U}^{\text{GOOD}}$ is difficult to quantify for use with Lemma 7, but lower bounds are obtained in Appendix D.

Even without a quantification of improvement, Theorem 2 shows that isolation functions can avoid worsening the context sensitivity, while certain conditions can guarantee relevance gains for predicting Y.

Theorem 2 (CIS costs and benefits)**.** *Consider $V_G \in \boldsymbol{V}^{\text{GOOD}}$ and $V_A \in \boldsymbol{V}^{\text{AMBIG}}$ where V_A is a SAP. Also consider the isolation function $F_{\text{ISO}(V_G)}(V_A \mid Y)$. We will compare the context sensitivity of inputs $\boldsymbol{X} := \{V_G\}$ and $\boldsymbol{X}^+ := \{V_G, F_{\text{ISO}(V_G)}(V_A \mid Y))\}$. We claim that $\mathcal{I}(M : Y \mid \boldsymbol{X}^+) \leq \mathcal{I}(M : Y \mid \boldsymbol{X})$ for all $M \in \boldsymbol{M}$. Furthermore, if*

$$\mathcal{I}(F_{\text{ISO}(V_G)}(V_A \mid Y) : V_G) < \mathcal{I}(F_{\text{ISO}(V_G)}(V_A \mid Y) : V_G \mid Y), \tag{8}$$

then the relevance improves: $\mathcal{I}(Y : \boldsymbol{X}^+) > \mathcal{I}(Y : \boldsymbol{X})$.

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 direct effects from $\mathcal{I}(Y : F_{\text{ISO}(V_G)}(V_A \mid Y))$ (for which we have no guaranteed bounds). The proof is given in Appendix E.

6.2 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 $I(F(V_A) : V_i) = \mathcal{I}(V_A : V_i)$. Information theoretically, minimizing $\mathcal{H}(F_{V_i}(V_A))$ corresponds to reducing the outputs of $F_{V_i}(V_i)$ to equivalence classes wherein $\Pr(V_A \mid F_{V_i}(V_i) = f)$ is constant. This minimization corresponds to ensuring $F_{V_i}(V_i)$ does not over-fit to

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

the empirical values of V_A using noise that is orthogonal to $\mathbf{PA}(V_A)$.

Auxiliary training tasks can therefore be used in place of isolation functions: we can get an approximate isolation function, $\tilde{F}_{\mathrm{ISO}(V_i)}(V_{\mathrm{SAP}})$, by training a model to predict V_i using input V_{SAP}. We do not give any theoretical results beyond intuition for this interpretation, but will support our claims with experiments in the next section.

Equation 8 in Theorem 2 also has a nice interpretation within the training context – the accuracy of the predictor must degrade when moving from the post-selected data to the full dataset. More precisely, the conditions for improvement now translate to

$$\min_F \mathbb{E}[\mathrm{Error}(F(V_A), V_G)] > \sum_y \Pr(y) \min_F (\mathbb{E}[\mathrm{Error}(F(V_A), V_G) \mid y]), \quad (9)$$

which can easily be checked on our training data.

6.3 SUGGESTED OVERALL PROCEDURE

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

1. Partition the data into constant $Y = y$ and determine cliques of dependence.
2. Using domain knowledge, identify seeds in $\boldsymbol{V}^{\mathrm{GOOD}}, \boldsymbol{V}^{\mathrm{BAD}}$ for proxy bootstrapping (Sec. 5.3).
3. Perform CIS on $\boldsymbol{V}^{\mathrm{AMBIG}}$ (Sec. 6.2).
4. Build a prediction model for Y using $\boldsymbol{V}^{\mathrm{GOOD}}$ and the CIS-engineered $\boldsymbol{V}^{\mathrm{AMBIG}}$.

7 EXPERIMENTS

We will now demonstrate the effectiveness of these methods on synthetic and real world data. Full code for both of these experiments is available at `https://zenodo.org/badge/latestdoi/651823136`.

7.1 EXPERIMENTS ON SYNTHETIC DATA

We generate data for the DAG in Figure 3 based on normal distributions, see details of the setup in Appendix F. 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, $\hat{Y}^{(3)}(V_G, \tilde{F}_{\mathrm{ISO}(V_G)}(V_A))$, that utilizes the auxiliary task approximation to isolate V_A's predictive information about V_G. We compare it to $\hat{Y}^{(1)}(V_G, V_A)$ trained on $\boldsymbol{V}^{\mathrm{GOOD}} \cup \boldsymbol{V}^{\mathrm{AMBIG}}$ and $\hat{Y}^{(2)}(V_G)$ trained on only $\boldsymbol{V}^{\mathrm{GOOD}}$. 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.

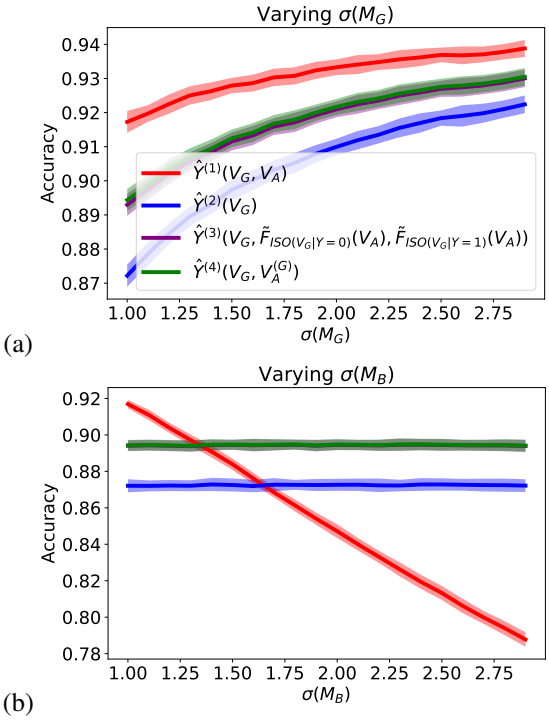

Figure 4: Results from our experiments on synthetic data. Single standard deviation confidence intervals are shaded in the corresponding colors.

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 $\boldsymbol{U}^{\mathrm{GOOD}}$ (a) or is small for $\boldsymbol{U}^{\mathrm{BAD}}$ (b). However, the accuracy of $\hat{Y}^{(2)}$ deteriorates with bigger shifts in $\boldsymbol{U}^{\mathrm{BAD}}$.

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 EXPERIMENTS ON CENSUS DATA

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]. To test out-of-domain generalization, prediction models were built on 2019 pre-pandemic data and evaluated on 2021 data during

the pandemic.[7] As model inputs, 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). This small feature set was purposefully selected to see a starker effect of including/excluding individual features, including 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. 6.2, referred to as engineered features, does not use HINS4 and JWMNP directly as input features to predict the income level. Instead it uses HINS4 and JWMNP to train two models predicting the education-level: One trained on examples with high income and one trained on examples with low income. These predictions based on HINS4 and JWMNP together with the actual education-level serve as input features to the final model. We compare the model built on these engineered features to ones 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.

Table 1: Comparison of out-of-domain (2021) performance via mean of accuracy.

State	All Features	Engineered Features	Limited Features
CA	**0.712** $\pm$ 0.0011	**0.711** $\pm$ 0.0014	0.692 $\pm$ 0.0014
FL	**0.683** $\pm$ 0.0012	0.678 $\pm$ 0.0018	0.68 $\pm$ 0.0013
GA	0.689 $\pm$ 0.0025	**0.707** $\pm$ 0.0055	**0.709** $\pm$ 0.0029
IL	0.662 $\pm$ 0.0026	**0.689** $\pm$ 0.0033	0.684 $\pm$ 0.0019
NY	**0.707** $\pm$ 0.0022	**0.702** $\pm$ 0.0025	0.687 $\pm$ 0.008
NC	**0.691** $\pm$ 0.0031	**0.684** $\pm$ 0.0034	**0.683** $\pm$ 0.003
OH	0.689 $\pm$ 0.0022	**0.703** $\pm$ 0.004	**0.696** $\pm$ 0.0029
PA	0.672 $\pm$ 0.0017	**0.695** $\pm$ 0.0023	0.688 $\pm$ 0.0022
TX	0.69 $\pm$ 0.0029	**0.712** $\pm$ 0.0028	**0.712** $\pm$ 0.0027
avg	0.688	**0.698**	0.692

Results Table 1 reports the mean and standard deviation of accuracies for 10 different test splits. For the F1 scores of the same experiment, see Appendix F. Using all features leads to the best in-domain performance (see Appendix F), but not necessarily the best out-of-domain performance. Dropping the ambiguous features hurts predictive power in limited feature models, but helps with robustness varies across the states: these limited models even perform better on 2021 data. Our proposed feature engineering using CIS achieves the best of both worlds, with the best mean out-of-domain accuracy of 0.698. It also achieves close to the best out-of-domain accuracy for 8 out of 9 states.

[7] We ignored the experimental release of 2020 data to ensure a starker distribution shift.

8 DISCUSSION

In this paper we studied the challenging problem of building 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 helpful and hurtful to our model's robustness. Through CIS, however, we showed how to isolate robust predictive power from these ambiguous proxies using auxiliary learning tasks. We proved that including these engineered features safely increases robustness in our setting, while also improving accuracy. In our experiments on real census data under shifts due to the pandemic, we showed that 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.

Relaxation of Assumptions A number of our assumptions can be softened. One softening of systemic sparsity would involve allowing edges within $\boldsymbol{U}$ so long as their dependence is relatively weak. Such a relaxation would involve using mutual information (or correlation) thresholds instead of independence tests. Sparsity assumptions may also be relaxed by building on ideas from mixtures of DAG structures like [Gordon et al., 2021].

The strongest assumption is that of separable ambiguous proxies. Under a softening of the separability assumption, we cannot guarantee that we have isolated only robust information from our ambiguous proxy – some unstable information associated with $\boldsymbol{U}^{\mathrm{BAD}}$ may slip through. However, degrees of separability may still guarantee the benefit of the engineered feature.

While separability corresponds to invertability with linear functions, there are many examples of nonlinear that are separable. For example, when the effects of two causes have significantly different magnitudes they can be easily disentangled, such as fine and hyper-fine structures in atomic energy levels. Work on data fission [Leiner et al., 2022] may provide valuable insights to help understand the degrees of separability for different choices of functions.

Acknowledgements

This work was completed at the Amazon Causality Lab in Tübingen, Germany. We would like to thank Dr. Leena Chennuru Vankadara for providing valueble feedback on the paper.

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. In Proceedings of the 37th International Conference on Machine Learning, ICML, 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.

James Leiner, Boyan Duan, Larry Wasserman, and Aaditya Ramdas. Data fission: splitting a single data point. arXiv preprint arXiv:2112.11079, 2022.

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.

Michael Oberst, Nikolaj Thams, Jonas Peters, and David Sontag. Regularizing towards causal invariance: Linear models with proxies. In International Conference on Machine Learning, pages 8260–8270. PMLR, 2021.

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), pages 947–1012, 2016a.

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, 2016b. ISSN 13697412, 14679868. URL http://www.jstor.org/stable/44682904.

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-.

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.