Causal Regularization Using Domain Priors

Abbavaram Gowtham Reddy
Indian Institute of Technology Hyderabad, India
cs19resch11002@iith.ac.in

Sai Srinivas Kancheti
Indian Institute of Technology Hyderabad, India
cs21resch01004@iith.ac.in

Vineeth N Balasubramanian
Indian Institute of Technology Hyderabad, India
vineethnb@iith.ac.in

Amit Sharma
Microsoft Research India
amshar@microsoft.com

Abstract

Neural networks leverage both causal and correlation-based relationships in data to learn models that optimize a given performance criterion, such as classification accuracy. This results in learned models that may not necessarily reflect the true causal relationships between input and output. When domain priors of causal relationships are available at the time of training, it is essential that a neural network model maintains these relationships as causal, even as it learns to optimize the performance criterion. We propose a causal regularization method that can incorporate such causal domain priors into the network and which supports both direct and total causal effects. We show that this approach can generalize to various kinds of specifications of causal priors, including monotonicity of causal effect of a given input feature or removing a certain influence for purposes of fairness. Our experiments on eleven benchmark datasets show the usefulness of this approach in regularizing a learned neural network model to maintain desired causal effects. On most datasets, domain-prior consistent models can be obtained without compromising on accuracy.

Keywords: Causality, Regularization, Deep Learning

1. Introduction

There has been a growing interest in integrating causal inference into machine learning models at different levels. Recent efforts have focused on post-hoc explanations of a trained neural network (NN) model’s decisions in terms of causal effect [10, 18], using counterfactuals for explanations or augmentations [19, 50, 29], causal discovery [49], or embedding causal structures in disentangled representation learning [41, 45]. None of these efforts however consider the setting when a user, before training a NN model, may have knowledge (complete or partial) of causal relationships between input and output variables. In this work, we propose a new methodology – to the best of our knowledge, the first of its kind - to regularize NN models during training in order to maintain such causal priors known a priori.

Causal domain priors can come in different forms across fields such as algorithmic fairness, economics, health and physical sciences, and can be integrated into a NN model during training using our method. Specific use cases include: (i) Fairness constraints that require a protected attribute’s influence on output to be zero – e.g. skin color in a classifier based on

* Equal contribution
facial images [12], or race in a loan approval prediction model in a bank [24]; (ii) Monotonic relationship between a specific input feature and output - e.g., a student’s score on a test may be required to be monotonically and causally related to prediction of admission into a college program, increasing the number of employees should have a positive effect on productivity. The nature of these real-world relationships is often a submodular prior [23] and can be enforced by our method; (iii) Arbitrary non-linear functions causally relating input and output - e.g., in medicine, U-shaped curves have been identified from randomized controlled trials (RCTs) for factors such as cholesterol level and diastolic blood pressure [33] and the dose-response curve for drugs [9]; or a J-shaped relationship between exposure and outcome, such as between alcohol and coronary heart disease [17], or between mortality and Body Mass Index [16]. Further discussion on priors is included in Section 4 and the Appendix.

We propose a method that can embed these different kinds of user-defined causal priors into a NN model during training (see Fig. 1) as detailed in Section 3. Importantly, we show that the final model learns causal effects that are consistent with these domain priors. The efforts closest to ours are those on enforcing monotonicity [38, 11, 20, 39], which however only focus on monotonicity (and not on other kinds of priors), and also do not study their models in terms of the causal effects learned. We study our causal regularizer both in the context of direct and total causal effects [28], depending on what information is available in the domain prior. Existing work does not make such distinction between these effects when training a NN model. Our contributions are:

• We propose a new method for Causal REgularization using DOmain priors (CREDO) and show formally that it can maintain consistency of the learned causal effect with the given domain prior. The method is conceptually simple and is easy to implement.

• The method supports regularization for all three kinds of effects—controlled direct, natural direct, and total causal effects. We show that estimation of natural direct and total effects requires partial knowledge of the underlying causal graph over input features while controlled direct effect does not.

• On 17 real-world and synthetic datasets, we show that CREDO can be used to enforce different kinds of domain priors, including zero effect, monotonic effect, and arbitrary functional priors. We show that the causal effects learned by NN model trained using CREDO is indeed closer to the true prior, with no significant impact on test accuracy.

2. Related Work

Our work focuses on maintaining priorly known causal relationships between input and output variables while training a NN model. One could view the earliest related efforts as [3] and [38], which attempted to maintain monotonicity in the functions learned by NNs by constraining the NN weights to be positive or the derivative of the output to be strictly positive. This was subsequently extended to the case of partially monotone problems [11][14] or specifically
Causal Regularization Using Domain Priors

in Bayesian networks [46]. However, these efforts did not consider the relationships to be causal; besides, these efforts focused on shallow models, where in case of NNs, it may be sufficient to control the parameters of just one layer of the NN.

More recent efforts related to our work are those that attempted to influence the feature attributions of a learned model to be either monotonic [21, 47, 20, 39] or zero [31, 30]; these efforts did not consider the causal implications either. [30] proposed a method to penalize model explanations that did not align with prior knowledge; this method is applicable for enforcing simple constraints like encouraging a feature’s attribution to be zero (e.g., spurious correlated feature) but not for monotonicity or other complex priors. Besides, the method uses associational attribution strategies and hence, may not conform with study of causal effect of input on output. [40] matched the Jacobian of a student network with a teacher network to transfer knowledge and feature influences from teacher to student. [36] proposed an active learning algorithm that samples new counterfactual [28] points by changing a single feature, obtaining a label, and then training a classifier on the augmented training set, with the goal of matching feature influences w.r.t. the oracle. Our key objective of regularizing a model to maintain priors for causal effect is different from these efforts. In contrast to existing methods, our method’s advantage is its generality: our method can not only force a feature’s attribution to be zero or monotonic but can also be used to enforce any differentiable function as a causal prior on the feature attribution. Importantly, we validate our causal regularization method by studying the causal effect of the learned model, which none of the earlier methods consider to the best of our knowledge. Moreover, we consider three kinds of effects—controlled direct, natural direct and total causal effects [27], and provide methods to regularize for any of them, which has not been done hitherto too.

Related efforts with causal implications have had different objectives such as removing confounding features [28] through regularization [4, 36, 22], or using causal discovery to infer stable features for prediction [25]. [4] used a form of weighted $L_1$ regularization to penalize non-causal attributes in a linear model, and show how this can be used in tandem with DNNs to view latent representations in DNNs as causal hypotheses. [22] related confounding and overfitting in shallow regression models, and thereby proposed a regularizer that also learns a co-efficient in $L_1$ or $L_2$ regularization. Both the aforementioned methods do not consider directly regularizing DNN models for causal effect. In contrast to these few existing related methods, our method focuses on training NN models whose causal attributions maintain prior causal influences that are provided as domain knowledge priors.

3. Defining Causal Effect in Neural Networks

As a motivating example, consider the task of predicting the body mass index (BMI) of a person based on features such as miles run each week, calorie intake per day, number of dogs one owns, and so on. From domain knowledge, we may expect a negative causal effect of miles run per week on BMI (higher the miles, lower the BMI). Calorie intake, on the other hand, has a positive effect on BMI, and may have a complex correlation with miles run — e.g., high calorie intake may be correlated both with people who run less as well as with people who run a lot (to support their exercise). While both these features directly cause BMI, a feature may affect BMI indirectly too. For instance, having dogs may enable more miles run or walked in a week, even though owning a dog does not directly affect the BMI.
The above example shows that features may have both direct and indirect effects on the outcome. Without modeling them correctly, a NN model may rely on correlations and produce less generalizable predictions or provide non-causal explanations. For example, a NN model may learn a non-zero direct effect of dogs owned by an individual on his/her BMI, which may lead to incorrect predictions (e.g., even with all other identical features, the model may predict a higher BMI for a person who owns dogs than another person). Similarly, on a training dataset with only high-activity runners who are both fit and have a high calorie intake, a NN model may learn that high calorie intake is associated with a lower BMI. While this may be a valid correlation in a training dataset, it may not reflect the true causal effect between these variables, and can lead to incorrect predictions on new data where the correlation no longer holds. It is hence important to match the feature-to-output relationship learned by a model to a known ground-truth causal relationship from feature to outcome, and distinguish between the kind of effect—direct or indirect.

To this end, we present a method for regularizing a NN that differentiate between different kinds of effects and respect them in the learned model $f$. This is achieved through a gradient matching approach, which enables the NN to learn the correct structural equations underlying the data generation process, whenever the nature of such relationships is known, even if partially only on some input variables. We begin by describing how a NN can be considered as a Structural Causal Model (SCM) and then define causal effects w.r.t the NN SCM.

Neural Network as a Structural Causal Model
Consider a NN $f : \mathbb{R}^d \rightarrow \{1, 2, \ldots, C\}$ (for classification) or $f : \mathbb{R}^d \rightarrow \mathbb{R}$ (for regression), with inputs $X = \{X_1, X_2, \ldots, X_d\}$ and output $\hat{Y} = f(X)$. Since NNs are function approximators, for theoretical analysis, w.l.o.g. one can marginalize the hidden layers of $f$ and consider only input and output nodes. Such NNs can be viewed as Structural Causal Models (SCMs) [10], encoding the interactions between input and output nodes. Thus, a NN can be considered as an SCM $S = (V, U, F)$, corresponding to a causal directed acyclic graph (DAG) $G = (V, U, E_G)$, where $V = \{X_1, X_2, \ldots, X_d, \hat{Y}\}$ is the set of observed variables, $U$ is the set of (unobserved) mutually independent exogenous noise variables and $E_G$ is the set of edges in $G$. $F$ is the set of structural equations operating on $U, V$. $G$ allows us to consider interactions among input nodes and output nodes of a NN by marginalizing hidden layers.

Defining $T, Z, W$. Since our goal is to identify different kinds of causal effect of a feature on output $\hat{Y}$, we divide input features $X$ into three disjoint subsets for convenience: $T, Z, W$. Here $T$ denotes the feature(s) for which we want to enforce a causal domain prior. Aligning with literature in causality [28], $T$ is akin to the treatment variable and $\hat{Y}$ is the target variable. $Z = \{Z_1, Z_2, \ldots, Z_K\}$ is the set of features that lie on a causal path between $T$ and $\hat{Y}$ in $G$ (mediators). $W$ denotes the set of remaining features.

Figure 1 shows the NN causal graph $G$ (exogenous variables not shown) among input features $T, Z, W$, and NN’s prediction $\hat{Y}$. Note that $G$ denotes the graph corresponding to the NN model and is different from the true causal graph of the underlying data-generating process $G^\circ$ that connects $X = \{T, Z, W\}$ to ground-truth $Y$. $X, U_X$ (set of exogenous variables which are ancestors of $X$) are variables present in both $G$ and $G^\circ$, and the relationships among them remain invariant across both graphs. All causal effects henceforth discussed in this paper are w.r.t. $G$ corresponding to the NN $f$, which we influence during training.
Following \cite{28}, we use the counterfactual notation $Y_t(u)$ to denote the value that $Y$ would attain given $U = u$ and the intervention $do(T = t)$, under the SCM $S$. For the treatment variable $T$, $t^*$ refers to a baseline treatment relative to which causal effects are computed. We consider three kinds of causal effects, as in \cite{27, 42}, of an input feature on model output $\hat{Y}$: controlled direct effect, natural direct effect, and (natural) total effect.

**Controlled Direct, Natural Direct, and Total Effect** Prior efforts assume all input features of a NN to be independent; this is a strong assumption which hinders us from considering indirect causal effect that is in turn necessary to estimate total causal effect. Such an assumption may not hold in practice for most real-world datasets. Moreover, the lack of this distinction creates chances of misinterpretation of a human-provided domain prior. Hence, we consider and define different kinds of causal effect priors. Throughout, we make the following assumptions, common in causal inference \cite{34, 37}.

**Assumption 1 Unconfoundedness.** There are no unobserved confounders \cite{28} in $G$. That is, observed features block all backdoor paths between any two features.

Note that unconfoundedness is always satisfied between $\hat{Y}$ and any feature, since parents of $\hat{Y}$ are the input features, which are all observed. While we specify Assumption 1 for clarity, our method can also work under certain weaker assumptions, provided in the Appendix.

**Assumption 2 Positivity.** $p(T = t|W, Z) > 0$ almost surely for all values of $T$, wherever $p(W, Z) > 0$.

where $T$, $W$ and $Z$ are the feature subsets as defined above. Specifically, $T$ refers to the feature subset whose effect we need to estimate (and regularize).

**Definition 3.1 (Controlled Direct Effect).** The Controlled Direct Effect (CDE) measures the causal effect of $T$ at $t$ (i.e. $T = t$) on $\hat{Y}$ when all parents of $\hat{Y}$ except $T$ ($Z, W$ in this case) are intervened to pre-defined control values $z, w$ respectively. CDE is defined as: $CDE^Y_t(u) := \hat{Y}_{t,z,w}(u) - \hat{Y}_{r,z,w}(u)$. The Average Controlled Direct Effect (ACDE) is defined as: $ACDE_t := \mathbb{E}_U[\hat{Y}_{t,z,w}] - \mathbb{E}_U[\hat{Y}_{r,z,w}] = \hat{Y}_{t,z,w} - \hat{Y}_{r,z,w}$.

While the expectation is taken over exogenous noise variables $U$, it can be removed since the neural network $\hat{Y} = f(T, W, Z)$ is a deterministic function which does not depend on the values the exogenous variables take. $t^*$ is a baseline treatment value, as stated earlier. The above definition of ACDE is defined for a particular intervention on $\{Z, W\}$ (i.e. all parents of $\hat{Y}$ except $T$) \cite{27}. By our construction, however, the domain priors are expressed only in terms of $T$ and $Y$, so we propose a modified definition for ACDE that marginalizes over $\{Z, W\}$. That is, we take the expectation over $\{Z, W\}$ (average of ACDE for all interventions on $\{Z, W\}$) along with $U$. Our version of ACDE is hence: $ACDE_t := \mathbb{E}_{Z,W,U}[\hat{Y}_{t,z,w}] - \mathbb{E}_{Z,W,U}[\hat{Y}_{r,z,w}]$

**Definition 3.2 (Natural Direct Effect).** The Natural Direct Effect (NDE) measures the causal effect of $T$ at $t$ on $\hat{Y}$ when the nodes of mediating variables $Z$ are fixed to their natural values $Z_{r}(u)$ under baseline treatment $do(T = t^*)$. NDE is defined as: $NDE^Y_t(u) := \hat{Y}_{t,Z_{r}(u)}(u) - \hat{Y}_{r,Z_{r}(u)}(u)$. The Average Natural Direct Effect (ANDE) is defined as: $ANDE_t := \mathbb{E}_U[\hat{Y}_{t,Z_{r}}] - \mathbb{E}_U[\hat{Y}_{r,Z_{r}}]$. 


Definition 3.3 (Total Causal Effect). The Total Causal Effect (TCE) of $T$ at $t$ on $\hat{Y}$ is given by $TCE_t^T(u) := \hat{Y}_t(u) - \hat{Y}_{t,z,u}(u) = \hat{Y}_{t,z,u}(u) - \hat{Y}_{t,z,u}(u)$. The Average Total Causal Effect (ATCE) is defined as: $ATCE_t^T := E_{Z}[\hat{Y}_{t,Z}] - E_{Z}[\hat{Y}_{t',Z,r}]$.

4. Identify & Regularize Causal Effect in NNs

Domain Prior. Let $g^c_i : \mathbb{R} \rightarrow \mathbb{R}$ be a function that encodes prior causal knowledge of how the $i$th input feature influences output class $c$ (note that this notation trivially extends to a regression setting with one or multiple outputs too). For example, $g^c_i$ may be a monotonically increasing function. While we assume $g$ as the domain prior here, in practice, our method can also work if only the shape of $g$ is available. More information on practical availability of $g$ is provided at the end of this section.

Matching gradients to domain prior. Given a known domain prior $g^c_i$, our objective is to ensure that the causal effects learned by the NN match the prior. Instead of directly matching causal effects, we rather match the gradient of NN’s causal effect with the gradient of $g^c_i$ for two reasons: (i) It is natural to get prior causal knowledge as a shape (or relative values) rather than exact values in many applications (see end of this section for examples), making gradient matching more natural than matching absolute values; and (ii) There is no closed form expression for the interventional expectation terms (Defns 3.1-3.3), thus making gradient matching more computationally efficient. Besides, it avoids having to choose a particular baseline treatment value. We hence match the gradient of $g^c_i$ with the gradient of ACE of $i$th input feature on $\hat{Y}$ under the graph $G$.

Next, we show how we use gradient matching to regularize for each of the aforementioned causal effects. For each of the causal effects considered—controlled direct, natural direct and total, we first prove that these effects are identifiable in NNs and then provide regularization procedures.

Identifying and Regularizing ACDE

Given a NN $f$, a datapoint $(t, z, w) \sim (T, Z, W)$ and its prediction $f(t, z, w)$, we can intervene on $T$ with $t'$ and compute $f(t', z, w)$. The expression $f(t', z, w) - f(t, z, w)$ gives an intuitive formula for the direct effect of $T$ on output. Below we show that this formula captures CDE as in Defn 3.1.

Proposition 1 (ACDE Identifiability in Neural Networks) For a neural network with output $\hat{Y}$, the ACDE of a feature $T$ at $t$ on $\hat{Y}$ is identifiable and given by $\text{ACDE}_t^T = E_{Z,W}[\hat{Y}|t,Z,W] - E_{Z,W}[\hat{Y}|t,Z,W]$.

Proofs of all propositions are in the Appendix. Since the ACDE is measured as the change in interventional expectation w.r.t. a baseline, the expected gradient of $\hat{Y}$ w.r.t. $t$ at $(t,z,w)$ is equivalent to the gradient of ACDE w.r.t. $t$.

Proposition 2 (ACDE Regularization in Neural Networks) The $n$th partial derivative of ACDE of $T$ at $t$ on $\hat{Y}$ is equal to the expected value of $n$th partial derivative of $\hat{Y}$ w.r.t. $T$ at $t$, that is: $\frac{\partial^n \text{ACDE}_t^T}{\partial t^n} = E_{Z,W}[\frac{\partial^n \hat{Y}}{\partial t^n}|t,Z,W]$.

Propn 2 allows us to enforce causal priors in a NN model by matching gradients. Let $x^j$ (instance of random variable $X$) denote the $j$th $d$-dimensional input to the NN. For a given
data point \( x^j \), let \( \delta G^j \) represent the matrix (of dimension \( C \times d \)) of derivatives of all available priors \( g^j \) w.r.t. \( x^j \), i.e. \( \delta G^j_{c,i} \) denotes the derivative of function \( g^j_c \) w.r.t. \( i^{th} \) feature of \( x^j \). To enforce \( f \) to maintain known prior causal knowledge (in terms of gradients), we define our regularizer \( R \) as:

\[
R(f, G, M) = \frac{1}{N} \sum_{j=1}^{N} \max\{0, ||\nabla_j f \odot M - \delta G^j||_1 - \epsilon\} \tag{1}
\]

where \( \nabla_j f \) is the \( C \times d \) Jacobian of \( f \) w.r.t. \( x^j \); \( M \) is a \( C \times d \) binary matrix that acts as an indicator of features for which prior knowledge is available; \( \odot \) represents the element-wise (Hadamard) product; \( N \) is the size of training data; and \( \epsilon \) is a hyperparameter to allow a margin of error. For the case where we wish to make gradient of ACDE of a feature to be zero, we set \( M_{c,i} = 1 \) and \( \delta G_{c,i} \) to be 0 and the regularizer hence simplifies into:

\[
R(f, G, M) = \frac{1}{N} \sum_{j=1}^{N} \max\{0, ||\nabla_j f \odot M||_1 - \epsilon\},
\]

which is equivalent to the loss function defined in fairness literature [31, 20].

**Identifying and Regularizing ANDE**

To identify natural effects in NNs, we need to know: (i) which features belong to the mediating variables set \( Z \) (it is enough to know partial causal graph involving \( Z \)); and (ii) the structural equations of how \( Z \) changes when \( T \) changes (Defs 3.2 and 3.3). If the variables \( Z \) from the true causal graph are not known, it is not possible to learn them from training data because there exists a set of causal graphs that are Markov-equivalent w.r.t. given training data distribution, each leading to different causal effects [28]. When \( Z \) is an empty set, the unbiased ANDE, obtained by controlling for \( \{Z, W\} \) will be ACDE itself [48]. Below we show that when \( Z \) is known, we can identify and regularize the ANDE in \( f \) w.r.t. a baseline \( t^* \).

**Proposition 3 (ANDE Identifiability in Neural Networks)** Given the DAG \( G' \) corresponding to a neural network \( f \), the ANDE of \( T \) at \( t \) on \( \hat{Y} \) is identifiable and is given by \( \text{ANDE}_{f} = \mathbb{E}_{Z_r, W}[\hat{Y}|t, Z_r, W] - \mathbb{E}_{Z_r, W}[\hat{Y}|t^*, Z_r, W] \) under the assumption that the set \( W \) forms a valid adjustment set [28].

**Proposition 4 (ANDE Regularization in Neural Networks)** The \( n^{th} \) partial derivative of ANDE of \( T \) at \( t \) on \( \hat{Y} \) is equal to the expected value of \( n^{th} \) partial derivative of \( \hat{Y} \) w.r.t. \( T \) at \( t \), that is,

\[
\frac{\partial^n \text{ANDE}^f}{\partial t^n} = \mathbb{E}_{Z_r, W} \left[ \frac{\partial^n \hat{Y}|t, Z_r, W}{\partial t^n} \right].
\]

In this case, \( R(f, G, M) \) is same as Equation 1 with the only difference that \( \nabla_j f \) is evaluated at \( (t, Z_r, W) \).

**Identifying and Regularizing ATCE**

Given known \( Z \), similar to ANDE, the ATCE of \( T \) at \( t \) on \( \hat{Y} \) is identifiable under assumptions similar to the previous case, as shown below.

**Proposition 5 (ATCE Identifiability in Neural Networks)** Given the DAG \( G' \) corresponding to a neural network \( f \), the total causal effect of \( T \) at \( t \) on \( \hat{Y} \) is identifiable and is given by \( \text{ATCE}^f = \mathbb{E}_{Z_r, W}[\hat{Y}|t, Z_r, W] - \mathbb{E}_{Z_r, W}[\hat{Y}|t^*, Z_r, W] \), under the assumption that the set \( W \) forms a valid adjustment set.
Proposition 6 (ATCE Regularization in Neural Networks) The gradient of the Average Total Causal Effect (ATCE) of $T$ at $t$ on $Y$ is equal to the expected value of the total gradient of $Y$ w.r.t. $T$ at $t$, that is,
\[
\frac{d\text{ATCE}_T}{dt} = \mathbb{E}_{Z, W} \left[ \frac{d[Y_t(Z,W)]}{dt} \right].
\]

As stated earlier, we provide all proofs in the Appendix. To regularize the total causal effect, we match the total derivative of $Y$ w.r.t. $t$ with the gradient of a given total causal effect prior. The regularizer $R(f, G, M)$ that enforces a NN model to maintain known total causal effect is then:
\[
R(f, G, M) = \frac{1}{N} \sum_{j=1}^{N} \max\{0, \|\nabla_j f \odot M - \delta G_j\|_1 - \epsilon\} \tag{2}
\]
where $\nabla_j f$ is the $C \times d$ matrix of total derivatives at input $x^j$. The computation of the total derivative is described in Algorithm 1. Other variables are as defined in Eqn. 1.

Final Algorithm: CREDO
To summarize, the overall optimization problem with the proposed CREDO regularizer to train the NN is given by:
\[
\arg \min_{\theta} ERM + \lambda_1 R(f, G, M) \tag{3}
\]
where $\theta$ are parameters of the NN $f$, $ERM$ stands for traditional Empirical Risk Minimizer over the given dataset (based on loss functions such as cross-entropy loss). The regularizer $R(f, G, M)$ is defined differently for each desired causal effect, as described above, and is summarized in Algorithm 1.

Algorithm 1: CREDO Regularizer

**Result:** Regularizer for ACDE, ANDE, ATCE of $f$.

**Input:** $D = \{(x^j, y^j)\}_{j=1}^{N}, y^j \in \{0,1, \ldots, C\}, x^j \in X^j; Q = \{i| \exists g^c_i \text{ for some } c\}; G = \{g^c_i\}$ prior for $i^{th}$ feature is available w.r.t. class $c$; $\mathbb{F} = \{f^1, \ldots, f^K\}$ is the set of structural equations of the underlying causal model s.t. $f^i$ describes $Z^i$; $\epsilon$ is a hyperparameter

**Initialize:** $j = 1, \delta G^j = 0_{c \times d} g_j^1 = 1, \ldots, N, M = 0_{c \times d}$

**while** $j \leq N$ **do**

**foreach** $i \in Q$ **do**

**foreach** $g^c_i \in G$ **do**

$\delta G^j[c, i] = \nabla g^c_i |_{x^j}; M[c, i] = 1$

**case 1:** regularizing ACDE do

$\nabla f[c, i] = \frac{\partial g^c_i}{\partial x^j}|_{x^j}$

**case 2:** regularizing ANDE do

/* causal graph is known */

$t = x_i$

$\nabla f[c, i] = \frac{\partial g^c_i}{\partial x^j}(t, z^j[i, w])$

**case 3:** regularizing ATCE do

/* causal graph is known */

$\nabla f[c, i] = \left[ \frac{\partial g^c_i}{\partial x^j} + \sum_{k=1}^{K} \frac{\partial g^c_k}{\partial x^j} \right]|_{x^j}$

end

end

$j = j + 1$

end

return $\frac{1}{N} \sum_{j=1}^{N} \max\{0, \|\nabla f \odot M - \delta G^j\|_1 - \epsilon\}$
### Availability of Causal Domain Priors

Our work is based on priors for the causal effect of a feature provided by users as input. In Table 1, we briefly list where such domain priors could come from and what forms they could assume. We highlight the fact that validating the correctness of the prior is not in the scope of this work; we assume the user-provided prior to be correct (i.e., representing the true causal mechanisms connecting features to true $Y$). When only the general shape is available, we can choose the shape parameters for which we obtain best validation accuracy, and then apply our method. We provide a more detailed discussion of this hyperparameter search in the Appendix. Going forward, we believe that methods like ours will encourage users to arrange for causal priors in their respective domains.

| Domain        | Example Prior                                      | Functional form          |
|---------------|----------------------------------------------------|--------------------------|
| Fairness      | Zero causal effect                                 | $y_x = \alpha; \forall x$ |
| Healthcare    | U-shape of drug effect                             | $y_x = ax^2$             |
| Healthcare    | J-shape of BMI                                      | $y_x = ae^{bx^2}; x > 0$ |
| Economics     | Diminishing returns                                 | $y_x = -ax^3 + bx^2$     |

Table 1: Examples of availability of domain priors ($a, b \in \mathbb{R}^+$).

### 5. Experiments and Results

We performed an extensive suite of experiments to study the proposed CREDO method, presented across this section and the Appendix. We divide this section into two parts: (i) when we know the partial or complete causal graph along with domain priors, we match any of ACDE, ANDE or ATCE; and (ii) when we do not know the causal graph generating the data and we can only match ACDE with the provided prior. We studied the performance of CREDO on synthetic and several well-known real-world datasets including COMPAS [1]; Medical Expenditure Panel Survey (MEPS) [2]; Law School Admission [44]; AutoMPG, Adult, Car Evaluation, Titanic, Boston Housing from the UCI repository [13]; and SANGIOVESE, MEHRA, SACHS, and Asia (Lung Cancer) from the BNLearn repository [35, 32, 43]. The BNLearn datasets have causal graphs provided, and hence help evaluate the first setting mentioned above. W.l.o.g., we set the baseline intervention $t^*$ as 0. Other experimental details including NN architectures and hyperparameters are provided in the Appendix.

#### Evaluation Metrics

To measure the conformity of the learned causal effect of the trained NN models with the given prior, we provide both qualitative and quantitative results. Qualitative results are obtained using ACE plots proposed in [10] that can measure the causal effect of any feature on the NN model’s output; it computes the interventional expectation via a second-order Taylor approximation of the NN. Quantitative results include Root Mean Square Error (RMSE), Frechet score and Pearson correlation coefficient, measured between ACE of the trained model and the given prior. We also report test accuracies to confirm that our regularizer does not affect model performance while incorporating causal priors. We observed that gradients on output logits work better than softmax or logsoftmax activated outputs. A grid search is performed to fix the regularization coefficient $\lambda_1$. We report the specific values chosen for each dataset in the corresponding result.

#### Baselines

We compare against vanilla Empirical Risk Minimization (ERM) without our regularizer in all our experiments. In a setting of making CDE zero, our method becomes the same as [31]; we hence get the same results under such a setting and hence do not compare
explicitly. In a setting of making ACE match a monotonic prior, we compare our results with Point Wise Loss (PWL) [20] and Deep Lattice Networks (DLN) [47]. In all results, “GT” (prefixed with the variable of interest) refers to the ground-truth prior provided. All experiments were conducted on one NVIDIA GeForce 1080Ti.

**Setting 1: Causal Graph is Known**

We first consider datasets from the BN-Learn repository where the causal graph/SCM is known. In this setting, the datasets used to train the NN models are generated from the given SCM. Here, we regularize for ACDE, ANDE, ATCE of a NN to match a desired prior w.r.t. the outcome. In some scenarios, it is possible that only partial knowledge of the SCM may be available, in which case one can perform a search for the exact functional form of the prior. More implementation details are provided in the Appendix.

**SANGIOVESE:** We generate data from the SANGIOVESE conditional Linear Gaussian Bayesian network. We convert the output into a categorical variable and train a NN that predicts if the yield of grapes is better than average. We choose three priors: a linear decreasing ACE of Acid feature on the output; and zero ACE for Potass (potassium content) and Polyph (total polyphenolic content) on the output. Fig 2 and Table 2 show the ACE plots and quantitative metrics learned by the trained model. Evidently, CREDO helps conform to the given causal prior (red and blue lines coincide); moreover, in this case, it even helps improve the accuracy slightly.

**MEHRA:**

We generate data from the conditional linear Gaussian network that models Multidimensional Environment-Health Risk Analysis (MEHRA). We convert the output, total precipitation (TP), into a categorical variable and train a NN that predicts if it is greater than average. We choose three priors: a nonlinear (inverted-V shaped) ACE of Latitude on TP; and a

**Figure 2:** Results on SANGIOVESE: Comparison of ACDE, ANDE, ATCE learned by ERM and CREDO models.

**Figure 3:** Results on MEHRA: Comparison of ACDE, ANDE, ATCE learned by ERM and CREDO models.
Causal Regularization Using Domain Priors

### Table 2: Results on SANGIOVESE: CREDO with ACDE ($\lambda_1 = 1.9$), ANDE ($\lambda_1 = 1.7$), ATCE ($\lambda_1 = 1.9$) and on MEHRA: CREDO with ACDE ($\lambda_1 = 2.2$), ANDE ($\lambda_1 = 2.1$), ATCE ($\lambda_1 = 1.5$)

| Feature | RMSE | Frechet | Corr | Feature | RMSE | Frechet | Corr |
|---------|------|---------|------|---------|------|---------|------|
| **ACDE** |      |         |      | **ACDE** |      |         |      |
| Potass  | 0.161 | 0.000   | 0.417| 0.000   | 0.366 | 0.449   | 0.115| 0.997 |
| PolypH  | 0.233 | 0.001   | 0.492| 0.002   | -     | -       | -    | -     |
| Acid    | 0.333 | 0.059   | 1.589| 1.055   | 0.997 | 0.998   | 0.998| 1.000 |
| Avg.    | 0.242 | 0.020   | 0.833| 0.352   | 0.997 | 0.998   | 0.998| 1.000 |

---

| Feature | RMSE | Frechet | Corr | Feature | RMSE | Frechet | Corr |
|---------|------|---------|------|---------|------|---------|------|
| **ERM CREDO** |      |         |      | **ERM CREDO** |      |         |      |
| Potass  | 0.309 | 0.045   | 0.366| 0.449   | 0.115| 0.997   | 0.997| 1.000 |
| PolypH  | 0.588 | 0.011   | 0.062| 0.991   | -0.494| 0.094   | 0.995| 1.000 |
| Acid    | 0.790 | 0.196   | 3.057| 1.810   | 0.983| 0.995   | 0.995| 1.000 |
| Avg.    | 0.562 | 0.084   | 1.162| 1.093   | -0.494| 0.094   | 0.995| 1.000 |

---

**Test Acc**

| Feature | RMSE | Frechet | Corr | Feature | RMSE | Frechet | Corr |
|---------|------|---------|------|---------|------|---------|------|
| **ERM CREDO-ACDE: 83.05%** |      |         |      | **ERM CREDO-ANDE: 83.00%** |      |         |      |
| **ERM CREDO: 82.00%** |      |         |      | **ERM CREDO-ATCE: 82.75%** |      |         |      |
| **ERM CREDO-ANDE: 80.05%** |      |         |      | **ERM CREDO-ATCE: 79.3%** |      |         |      |

---

**Setting 2: Causal Graph Unknown**

As discussed in Section 4, when we do not know the causal graph, the best we can do is regularize for ACDE. We study the performance of CREDO when regularizing for ACDE on synthetic and real-world datasets with different domain priors as detailed below. In each case, we make the assumption that the parametric form of the prior function as well as the search spaces are given (which is reasonable to assume in this context). We then choose the best parameters for the prior function by tuning for highest validation-set accuracy. More details about this procedure are in the Appendix.

**COMPAS:**

In the COMPAS dataset, the task is to predict the likelihood of two-year recidivism (re-offending in next two years) given a set of features. This dataset is studied in fairness literature [6, 5, 7, 15]. From a fairness perspective, we expect NN models to have zero causal effect of race while predicting recidivism. Table 3 and Fig 4 show our results. CREDO is able to force linearly increasing ACE of $O3$, $SO2$ on $TP$. The ACE plot in Fig 3 and quantitative results in Table 2 show that CREDO helps conform to the causal prior in this case too. Our results on SACHS and Asia datasets showed similar trends, and are reported in the Appendix.

Figure 4: Results on COMPAS, BOSTON, and AutoMPG datasets
the ACDE of race on two-year recidivism to be close to zero with insignificant effect on model accuracy. Note the significant reduction in the RMSE, permutation and Frechet scores, which measure the alignment of ACE of the trained model w.r.t. the domain prior. Fig 4 shows ACDE across the range of the race variable on the recidivism outcome, and shows that CREDO brings ACDE significantly closer to the given prior.

**Boston Housing:** This dataset concerns home values in the suburbs of Boston. We convert the output attribute home value into a categorical output, and learn a classifier that can predict if the home value is better than average. As the domain prior, we want crime rate and concentration of Nitric Oxides (above a threshold) to have a decreasing causal effect and number of rooms (RM) to have an increasing causal effect w.r.t. housing prices. Fig 4 and Table 3 show that CREDO helps align the ACDE of the trained model with the prior in all these cases.

**AutoMPG:** The AutoMPG dataset contains the mileage of various cars given attributes such as weight, horsepower, displacement, etc. We learn a classifier that can predict if the mileage is better than average. We want displacement, weight and horsepower to have a decreasing causal effect w.r.t. mileage. Results shown in Fig 4 and Table 3 once again support the usefulness of our method.

We obtain similar results on enforcing domain priors over the MEPS, Law School Admission, Adult, Car Evaluation, and Titanic datasets, as shown in the Appendix. More studies including time complexity, effect of the regularization co-efficient and the effect of use of incorrect priors are provided in the Appendix owing to space constraints.

### Table 3: Enforcing Causal Effects (ACDE) of Multiple Variables: Results on COMPAS, BOSTON Housing and AutoMPG datasets

| Feature         | RMSE ERM | Frechet Score ERM | Corr. Coeff. ERM | RMSE CREDO | Frechet Score CREDO | Corr. Coeff. CREDO |
|-----------------|----------|-------------------|------------------|------------|---------------------|-------------------|
| COMPAS ($\lambda = 5$) ERM test accuracy is 67.90%, CREDO test accuracy is 67.09% |
| African American | 0.055    | 0.016             | 0.088            | 0.025      | -                   | -                 |
| Asian           | 0.092    | 0.018             | 0.162            | 0.021      | -                   | -                 |
| Native American | 0.059    | 0.011             | 0.109            | 0.025      | -                   | -                 |
| Boston ($\lambda = 2.2$) ERM test accuracy is 88.2%, CREDO test accuracy is 85.30% |
| Crime           | 0.52     | 0.145             | 0.181            | 1.951      | 0.996               | 0.999             |
| Nitric Oxide    | 0.165    | 0.080             | 1.265            | 0.994      | 0.957               | 0.991             |
| Num. of Rooms   | 0.994    | 0.036             | 3.786            | 2.009      | 0.993               | 1.000             |
| AutoMPG ($\lambda = 1.5$) ERM test accuracy is 88.6%, CREDO test accuracy is 87.34% |
| Displacement    | 1.144    | 0.212             | 0.566            | 1.524      | -0.945              | 0.977             |
| Horsepower      | 1.036    | 0.081             | 6.978            | 3.908      | 0.922               | 0.999             |
| Weight          | 1.780    | 0.25              | 9.453            | 5.510      | 0.986               | 0.992             |

6. Conclusion

In this work, we propose a new causal regularization method, CREDO, that can learn neural network models whose learned causal effects match prior domain knowledge as provided by a user. We show that the method can work with any differentiable prior representing complete or partial understanding of the domain. Importantly, we make the distinction between direct and total causal effects, and show how both can be considered in CREDO. We theoretically analyze our method and show why the regularizer helps align the learned causal effect with the prior. We perform extensive experiments on various datasets, with known and unknown causal graphs, to show the effectiveness of our method under different kinds of priors. CREDO shows promising performance in matching causal domain priors with no significant impact on model accuracy/training time.
Appendix A.

In this appendix, we include the following additional information.

- Proofs of propositions
- Discussion on sources of causal domain priors for using CREDO in practice
- Details on the assumptions of causal inference used and implementation details of CREDO
- More experimental results:
  - Enforcing fairness constraints
  - Comparison of CREDO with pointwise loss (PWL) [20], deep lattice networks (DLN) [47]
  - Experiments on BNLearn datasets
- Ablation studies
- Description of computation of ACE (Average Causal Effect), as in [10]
- Causal graphs/DAGs for the BNLearn datasets
- Architectures/training details of our models for all the datasets

Appendix A. Proofs of Propositions

We begin writing the proofs of our propositions by recollecting two key results from [28] which we use in our proofs—(i) When there is no backdoor path from treatment $T$ to the outcome $\hat{Y}$, interventional distribution is equal to the conditional distribution i.e., $p(\hat{Y}|do(T = t)) = p(\hat{Y}) = p(\hat{Y}|T = t)$. (ii) If there exist a set of nodes $W$ that satisfy the backdoor criteria relative to the causal effect of $T$ on $\hat{Y}$, the causal effect of $T$ on $\hat{Y}$ can be evaluated using the adjustment formula $E[\hat{Y}_t] = E[\hat{Y}|do(T = t)] = \sum_{\hat{Y}} \hat{Y}p(\hat{Y}|do(T = t)) = \sum_{w} \sum_{\hat{Y}} \hat{Y}p(\hat{Y}|T = t, W = w)p(W = w)$. Note that this is equivalent to $E[W|\hat{Y}, t, W]$ in our notation, since the inner expectation over $\hat{Y}$ vanishes due to the deterministic nature of the NN. (We use a similar notation for expectations over gradients in Propositions 2, 4 and 6.)

**Proposition 7** (ADCE Identifiability in Neural Networks) For a neural network with output $\hat{Y}$, the ACDE of a feature $T$ at $t$ on $\hat{Y}$ is identifiable and given by $ACDE_t^\hat{Y} = E[Z, W][\hat{Y}_t, Z, W] - E[Z, W][\hat{Y}|T = t, Z, W]$.

**Proof** Starting with the definition of $ACDE$ of $T$ at $t$ on $\hat{Y}$, we get

$$ACDE_t^\hat{Y} = E[Z, W, u][\hat{Y}_t, Z, W] - E[Z, W, u][\hat{Y}_r, Z, W]$$

$$= E[Z, W][\hat{Y}_t, Z, W] - E[Z, W][\hat{Y}_r, Z, W]$$

$$= E[Z, W][\hat{Y}|t, Z, W] - E[Z, W][\hat{Y}|r, Z, W]$$
Since a NN $f$ is a deterministic function of its inputs, expectation over $\mathbf{U}$ can be discarded once we condition on all features (second equality above). Further, once all parents of $\hat{Y}$ (i.e., $T, Z, W$) are intervened, there cannot be an unobserved confounder that causes both $\hat{Y}$ and $T$. Hence unconfoundedness is valid for the effect from $T$ to $\hat{Y}$. For this reason, we can replace the intervention with conditioning in the last step. ■

**Proposition 8 (ACDE Regularization in Neural Networks)** The $n^{th}$ partial derivative of ACDE of $T$ at $t$ on $\hat{Y}$ is equal to the expected value of $n^{th}$ partial derivative of $\hat{Y}$ w.r.t. $T$ at $t$, that is: \[ \frac{\partial^n \text{ACDE}_i}{\partial t^n} = \mathbb{E}_{Z, W}[\frac{\partial^n (\hat{Y}_{t, Z, W})}{\partial t^n}] \]

**Proof** Using the identifiability result from Proposition 1,

\[ \text{ACDE}_i^\hat{Y} = \mathbb{E}_{Z, W}[\hat{Y}_{t, Z, W}] - \mathbb{E}_{Z, W}[\hat{Y}_{t^*, Z, W}] \]

Now, taking the $n^{th}$ partial derivative w.r.t $t$ on both sides,

\[
\frac{\partial^n \text{ACDE}_i^\hat{Y}}{\partial t^n} = \frac{\partial^n [\mathbb{E}_{Z, W}[\hat{Y}_{t, Z, W}] - \mathbb{E}_{Z, W}[\hat{Y}_{t^*, Z, W}]]}{\partial t^n} \\
= \frac{\partial^n [\mathbb{E}_{Z, W}[\hat{Y}_{t, Z, W}]]}{\partial t^n} (\because t^* is a constant) \\
= \mathbb{E}_{Z, W}[\frac{\partial^n [\hat{Y}_{t, Z, W}]}{\partial t^n}] 
\]

**Proposition 9 (ANDE Identifiability in Neural Networks)** Given the DAG $\mathcal{G}$ corresponding to a neural network $f$, the ANDE of $T$ at $t$ on $\hat{Y}$ is identifiable and is given by $\text{ANDE}_i^\hat{Y} = \mathbb{E}_{Z^t, W}[\hat{Y}_{t, Z^t, W}] - \mathbb{E}_{Z^t, W}[\hat{Y}_{t^*, Z^t, W}]$ under the assumption that the set $W$ forms a valid adjustment set \cite{28}.

**Proof** Assuming that the set $W$ forms a valid adjustment set in the calculation of causal effect of $T$ on $\hat{Y}$, from the backdoor adjustment formula \cite{28}, we get $\mathbb{E}_{\mathcal{U}}[\hat{Y}_{t, Z^t}] = \mathbb{E}_{W, \mathcal{U}}[\hat{Y}_{t, Z^t}, W]$. However, the value $Z_t$ not only depends on the intervention $do(T = t)$ but also on $\mathcal{U}, W$ \cite{27}, and hence we can write this as: $\mathbb{E}_{\mathcal{U}}[\hat{Y}_{t, Z^t}] = \mathbb{E}_{Z^t, W, \mathcal{U}}[\hat{Y}_{t, Z^t}, W]$. Therefore,

\[
\text{ANDE}_i^\hat{Y} = \mathbb{E}_{\mathcal{U}}[\hat{Y}_{t, Z^t}] - \mathbb{E}_{\mathcal{U}}[\hat{Y}_{t^*, Z^t}] \\
= \mathbb{E}_{Z^t, W, \mathcal{U}}[\hat{Y}_{t, Z^t}, W] - \mathbb{E}_{Z^t, W, \mathcal{U}}[\hat{Y}_{t^*, Z^t}, W] \\
= \mathbb{E}_{Z^t, W}[\hat{Y}_{t, Z^t}, W] - \mathbb{E}_{Z^t, W}[\hat{Y}_{t^*, Z^t}, W] 
\]

The second equality is because of the adjustment formula as described above. Further, similar to the ACDE case, the third equality is because once we condition on all input features, $f$ is a deterministic function of its inputs and hence expectation over noise variables can be omitted. Further, $Z_{t^*}$ is identified because all parents of $Z$ are observed as per Assumption 1. ■
Proposition 10 (ANDE Regularization in Neural Networks) The $n^{th}$ partial derivative of \( \text{ANDE} \) of $T$ at $t$ on $\hat{Y}$ is equal to the expected value of $n^{th}$ partial derivative of $\hat{Y}$ w.r.t. $T$ at $t$, that is, 
\[
\frac{\partial^n \text{ANDE}_t}{\partial t^n} = E_{Z_t, W} \left[ \frac{\partial^n [\hat{Y}|t, Z_t, W]}{\partial t^n} \right].
\]

Proof Using the identifiability result from Proposition 3,
\[
\text{ANDE}_t \dot{Y} = E_{Z_t, W} [\hat{Y}|t, Z_t, W] - E_{Z_t, W} [\hat{Y}|t^*, Z_t, W]
\]
Now, taking the $n^{th}$ partial derivative w.r.t $t$ on both sides,
\[
\frac{\partial^n \text{ANDE}_t}{\partial t^n} = \frac{\partial^n \left[ E_{Z_t, W} [\hat{Y}|t, Z_t, W] - E_{Z_t, W} [\hat{Y}|t^*, Z_t, W] \right]}{\partial t^n} = \frac{\partial^n \left[ E_{Z_t, W} [\hat{Y}|t, Z_t, W] \right]}{\partial t^n} (\because t^* \text{ is a constant}) = E_{Z_t, W} \left[ \frac{\partial^n \hat{Y}|t, Z_t, W}{\partial t^n} \right]
\]

Proposition 11 (ATCE Identifiability in Neural Networks) Given the DAG $\mathcal{G}$ corresponding to a neural network $f$, the total causal effect of $T$ at $t$ on $\hat{Y}$ is identifiable and is given by 
\[
\text{ATCE}_t \dot{Y} = E_{T, W} [\hat{Y}|t, Z_t, W] - E_{Z_t, W} [\hat{Y}|t^*, Z_t, W],
\]
under the assumption that the set $W$ forms a valid adjustment set.

Proof Assuming that the set $W$ forms a valid adjustment set in the calculation of causal effect of $T$ on $\hat{Y}$, similar to the proofs above, from the backdoor adjustment formula [28], we get:
\[
\text{ATCE}_t \dot{Y} = E_{T, W} [\hat{Y}|t, Z_t, -\hat{Y}_t, Z_t] = E_{Z_t, W} [\hat{Y}|t, Z_t, W] - E_{Z_t, W} [\hat{Y}|t^*, Z_t, W]
\]
Similar to ANDE identifiability in Proposition 3, we can reason about the explicit expectation over $Z_t, Z_t$ in the second equality. Similar to the ACDE, ANDE identifiability in Propositions 1-3, once we condition on all input features, $f$ is a deterministic function of its inputs and hence expectation over noise variables can be omitted.

Note that the values of $Z_t, Z_t$ in both cases (ANDE, ATCE) are decided by the causal mechanism that generates $Z$ which are learned as separate regressors in our implementation.

Proposition 12 (ATCE Regularization in Neural Networks) The gradient of the Average Total Causal Effect (ATCE) of $T$ at $t$ on $\hat{Y}$ is equal to the expected value of the total gradient of $\hat{Y}$ w.r.t. $T$ at $t$, that is, 
\[
\frac{d\text{ATCE}_t}{dt} = E_{Z_t, W} \left[ \frac{d[Y|t, Z_t, W]}{dt} \right].
\]

Proof Let $f^i$ denote the structural equation of the underlying causal model for variable $Z^i$ (a total of $K$ such equations, which are learned separately). W.l.o.g., assuming $Z$'s are topologically ordered, we have $Z^i = f^i(T, Z^{i-1}, W)$ (we add $W$ for generality;
it is not necessary for all variables in \( W \) to cause \( Z^i \). Consequently, at \( T = t \), we have \( Z^i_t = f^i(t, Z^1_t, Z^2_t, \ldots, Z^{i-1}_t, W) \). Consider the first-order Taylor expansion of \( Z^k_{t+\Delta t} \):

\[
Z^k_{t+\Delta t} \approx Z^k_t + \Delta t \left[ \frac{d f^k}{d t} \right], \quad \text{where} \quad \frac{d f^k}{d t} = \frac{\partial f^k}{\partial t} + \sum_{j=1}^{k-1} \frac{\partial f^k}{\partial Z^j} \frac{df^j}{dt}
\]

We can prove Eqn 4 by induction over \( k \). Since we are interested in the behavior of TCE w.r.t. the interventional value, we consider the first-order Taylor expansion of \( \hat{Y}_{t+\Delta t} \):

\[
\hat{Y}_{t+\Delta t} = f(t + \Delta t, Z^1_{t+\Delta t}, \ldots, Z^K_{t+\Delta t}, W)
\]

\[
\approx f(t + \Delta t, Z^1_t + \Delta t \left[ \frac{df^1}{dt} \right], \ldots, Z^K_t + \Delta t \left[ \frac{df^K}{dt} \right], W)\]

\[
\approx \hat{Y}_t + \Delta t \left[ \frac{\partial f}{\partial t} + \sum_{j=1}^{K} \frac{\partial f}{\partial Z^j} \frac{df^j}{dt} \right]
\]

Taking \( \hat{Y}_t \) to the left, and adding-subtracting \( \hat{Y}_t \), we get the LHS of Equation 5 as \( \Delta \text{TCE}^{\hat{Y}}_t = \text{TCE}^{\hat{Y}}_{t+\Delta t} - \text{TCE}^{\hat{Y}}_t \). In the limit that the perturbation \( \Delta t \) is very small, we get rid of the error introduced by the first-order Taylor approximations. Thus we have:

\[
\lim_{\Delta t \to 0} \frac{\Delta \text{TCE}^{\hat{Y}}_t}{\Delta t} = \frac{d \text{TCE}^{\hat{Y}}_t}{dt} = \frac{\partial f}{\partial t} + \sum_{j=1}^{K} \frac{\partial f}{\partial Z^j} \frac{df^j}{dt}
\]

Finally, taking expectation on both sides completes the proof (by Leibniz integral rule).

\[
\frac{d \Delta \text{TCE}^{\hat{Y}}_t}{dt} = \mathbb{E}_{Z_t, W} \left[ \frac{d [\hat{Y}^t|t, Z_t, W]}{dt} \right]
\]

where \( \frac{d [\hat{Y}^t|t, Z_t, W]}{dt} = \frac{\partial [\hat{Y}^t|t, Z_t, W]}{\partial t} + \sum_{j=1}^{K} \frac{\partial [\hat{Y}^t|t, Z_t, W]}{\partial Z^j} \frac{df^j}{dt} \)

\[\blacksquare\]

Appendix B. Availability of Causal Domain Priors

Our work is based on priors for the causal effect of a feature provided by users. Extending our discussion on the availability of such causal domain priors in the main paper (Section 4 and Table 1), we provide below a more detailed discussion and list examples of different kinds of priors that are commonly known in fields such as algorithmic fairness, economics, health and physical sciences, and can be used in CREDO for building robust prediction models.

- **U-shaped causal effects:** There are many situations where it is known that a feature’s effect follows a U-shaped pattern, i.e. increasing the feature may increase the outcome
up to a point, after which it starts decreasing the outcome. In medicine, U-shaped curves have been found for various factors, such as cholesterol level, diastolic blood pressure [33], and the dose-response curve for drugs [9]. Another example is the relationship of mortality with age for certain diseases, where infants and elderly may experience the highest mortality (although sometimes more complex relationships are observed) [26]. All such effects can be modeled by the proposed CREDO algorithm.

- **J-shaped causal effects:** [17] studied the J-shaped relationship between exposure and outcome, such as between alcohol with coronary heart disease. Similarly, [16] observed a J-shaped relationship between mortality and Body Mass Index. CREDO allows the use of such non-linear priors while training a model and thus retain these causal relationships in the NN model.

- **Zero (direct) causal effect:** In many cases, a feature may have a spurious correlation with the outcome and an ML model should not learn such correlations. For example, skin color should not matter in a classifier based on facial images [12], race/caste should not matter in a loan approval prediction model in a bank, and so on. A special case comes up in algorithmic fairness where we may allow total effect of a sensitive feature to be non-zero, but require the direct effect to be zero [48]. For example, sensitive demographic features may affect a college admission decision mediated by the test score, but not directly. By formally distinguishing between direct and total effect regularization, our method makes it possible to regularize for such constraints.

- **General monotonicity (e.g. “diminishing returns”):** In addition to plain monotonicity as done by [20, and 47], domain experts often have additional information. The relationship may be super-linear (gradient of gradient is positive) or sub-linear. A special case of sub-linear monotonicity is the diminishing returns observation [8] in economics (e.g., increasing the number of employees has a positive effect on productivity, but the effect decreases with the number of people already added). This is a popular submodular prior for many scenarios [23] and can be enforced by our method, while prior methods on monotonicity do not provide any guarantees on specific functions.

**Appendix C. Assumption of Causal Inference**

Our work relies on the assumptions of Unconfoundedness and Positivity (Section 3 of main paper). For simplicity and clarity, we used unconfoundedness assumption directly in the main paper. We now list a set of weaker assumptions/conditions than Unconfoundedness which are enough to support all the theoretical analysis in this work. For the treatment $T$, whose effect we want to regularize, we assume that “conditioning on all other features is sufficient to block the backdoor paths from $T$ to $\hat{Y}$”. This assumption holds true whenever one of the following conditions hold in the causal graph $G$.

- Whenever $T$ does not cause any other feature with whom it shares an unobserved confounding.
- Whenever $T$ does not cause any other feature.
- Whenever $T$ does not have any unobserved confounder w.r.t. $\hat{Y}$. 

17
Appendix D. Implementation Details of CREDO

D.1 Evaluating $Z_t, Z_{t'}$

The definitions of ANDE and ATCE require the values $Z_t, Z_{t'}$ to evaluate the causal effect of $T$ on $\hat{Y}$. To find $Z_t, Z_{t'}$, we need to know the structure of the causal graph. If complete causal graph is not available, we at least need to know the partial causal graph involving $Z$. Since ANDE and ATCE are regularized for the setting where we have access to causal graph, we chose BNLearn datasets for our study: SANGIOVESE, MEHRA, ASIA (Lung Cancer), and SACHS. We assume that the underlying SCM is linear with Gaussian noise — this assumption holds true for SANGIOVESE and MEHRA— and we model each structural equation of $Z_i \in Z$ as a function of its parents in the form of separate regressors. These regressors are learned independently from the model being trained and regularized using CREDO.

D.2 How is the exact functional form of the prior determined?

If the exact functional form of the prior is provided, we can use it as it is in our method. However, we often get causal domain prior as a shape rather than an exact function. When the true parameters of such a function are not known, we search over possible values they can take (within a range) and choose the ones with the highest validation-set classification accuracy (this would function like any other hyperparameter search done for neural network models). For example, if the prior shape is linear w.r.t. a feature $t$, we search for a hyperparameter value for the slope $\alpha$ such that prior function is $\alpha t + c$ and choose the value with the highest validation accuracy. Performing a simple linear search over 10-20 values of $\alpha$ (e.g., 0.1 to 2 with 0.1 increment) suffices to achieve better/equal performance compared to ERM. We observed this in most of our experiments. For non-linear priors, a similar search can be performed as long as a parametric form and a reasonable search-space can be assumed. We, in general, assume that a prior shape and a search space are provided as the domain priors in this work.

Appendix E. More Experimental Results

Along with the 5 dataset results in the main paper, we herein present additional experimental results on 8 real-world and 4 synthetic datasets covering real-world use cases of CREDO as well as ablation studies.

E.1 Enforcing Fairness Constraints

Law School Admission: In the Law School Admission dataset, the task is to predict whether a student gets admission into a law school based on a set of features. We expect a model to have no influence of protected attributes like gender, race on admission. Table 5 and Fig 4 reiterate our claims of CREDO's usefulness in forcing the ACDE of gender, race on admission to be zero without drop in model accuracy.

MEPS: In the MEPS (Medical Expenditure Panel Survey) dataset, usually the task is to predict the utilization score of an individual. Utilization can be thought of as requiring additional care for an individual, and such decisions should be made independent of the race
Table 4: Enforcing Zero Causal Effect: Plot of ACDE of sensitive attributes on outcome in Law School, MEPS datasets.

Table 5: Results on Law School, MEPS datasets

Table 6: Results on ADULT, Titanic datasets

E.2 Comparison with PWL and DLN

In order to compare CREDO with Pointwise Loss (PWL) [20] and Deep Lattice Networks (DLN) [47], which incorporate monotonicity domain priors into neural networks without causal motivation, we create two synthetic datasets and perform experiments as described below.
Table 7: Enforcing Monotonicity and Zero Causal Effect: Plot of causal effects learned by models trained on Adult dataset.

Table 8: Enforcing Monotonicity on age: Comparison of ACE plots of ERM, CREDO on Titanic dataset.

Table 9: ACDE plots of ERM, CREDO, PWL and DLN on Synthetic Tabular 1 dataset. With known prior, CREDO performs better than other methods.

**Synthetic Tabular-1:** We consider a synthetic dataset of the form $z = \log(1+2x), x \in [0, 1]$. Now, to regularize the causal effect of $x$ on $z$ in a neural network with input $x$ and output $z$, we employ CREDO, PWL, and DLN independently. We observe that CREDO matches the prior shape better when compared to PWL and DLN (Fig 9, Table 10).

**Synthetic Tabular-2:** We also consider a synthetic dataset of the form $f(x, y) = \sin(x) + e^y, x \in [0, 1]$, $y \in [0, 1]$. $f(x, y)$ is monotonically increasing w.r.t. $y$ when $x$ is kept constant; specifically the domain prior is $e^y$ when we regularize the model w.r.t. $y$. The results are shown in Fig 5 and Table 10, where we compare with PWL and DLN. While all models learn monotonic ACE of $y$ on outcome, our regularizer is closer to the true prior.

| RMSE  | Frechet Score | Corr. Coeff | Test Loss |
|-------|---------------|-------------|-----------|
| Synthetic Tabular-1 | | | |
| ERM   | 0.10          | 1.7e-3      | 0.99      | 1e-2         |
| PWL   | 0.10          | 1.7e-3      | 0.99      | 1e-2         |
| DLN   | 0.27          | 0.31        | 0.91      | 1e-3         |
| CREDO | **0.04**      | **1.6e-3**  | **1.0**   | **1e-2**     |

| Synthetic Tabular-2 | | | |
| ERM   | 0.10          | 1e - 3      | 0.98      | 2.05e – 1    |
| PWL   | 0.10          | 1e - 3      | 0.98      | 1.45e – 1    |
| DLN   | 0.25          | 0.26        | 0.90      | 2.7e – 1     |
| CREDO | **0.01**      | **1e - 3**  | **0.99**  | **1.04e – 1** |

Table 10: Comparison of ERM, CREDO, PWL, DLN on Synthetic Tabular-1, Synthetic Tabular-2 datasets.
Causal Regularization Using Domain Priors

Table 11: Causal Bayesian network that generates ASIA/Lung Cancer dataset

Table 12: Enforcing Monotonic Effects: Results on ASIA dataset

| Feature | RMSE | Frechet Score | Corr. Coeff. |
|---------|------|---------------|--------------|
| ERM     | CREDO | ERM           | CREDO        |
| Tub     | 0.387 | 0.279         | 0.729        | 0.579         | 0.874 | 0.983 |
| Lung    | 0.858 | 0.265         | 1.55         | 0.620         | -0.989 | 0.964 |

Table 13: Causal Bayesian network that generates SACHS dataset

Table 14: Enforcing Zero Causal Effects: Results on SACHS

| Feature | RMSE | Frechet Score |
|---------|------|---------------|
| ERM     | CREDO | ERM           | CREDO        |
| Jnk     | 0.030 | 0.000         | 0.038        | 0.000         |
| P38     | 0.026 | 0.000         | 0.036        | 0.000         |
| PIP2    | 0.073 | 0.000         | 0.199        | 0.000         |
| PIP3    | 0.103 | 0.001         | 0.149        | 0.001         |
| Pleg    | 0.020 | 0.000         | 0.035        | 0.000         |

E.3 Known Causal Graph: BNLearn Datasets

Asia/Lung Cancer: Since ASIA/Lung Cancer dataset is taken from the BNLearn repository, we have access to the true causal graph that generated the dataset (Fig 11). In this case, we regularize for tuberculosis (tub) and lung cancer (lung) to have monotonically increasing relationship on the outcome dyspnoea (dysp). Table 12 shows the results where the regularized model, without any change in accuracy, learns to incorporate prior knowledge.

SACHS: In the SACHS dataset (generated using underlying causal graph shown in Figure 13), without going into the semantic meanings of the features, it is evident that Jnk, PIP2, PIP3, Pleg, P38 are non-causal predictors of Akt. With CREDO, we get good accuracy while maintaining the zero causal effect of the non-causal predictors on the outcome (Table 14).

Appendix F. Ablation Studies and Analysis

We now report our results from ablation studies that study various aspects of CREDO, such as how CREDO behaves under different priors for the same problem, time complexity analysis, etc.
Enforcing Monotonicity with Different Priors Using heart disease dataset [13], we show how our method can be used when it is known that the prior shape is monotonic, but the exact slope of the monotonic prior is not known. In particular, we show that we are able to match any assumed shape of the monotonic prior in this setting. (This raises questions on how a given prior can be validated for correctness, which is an interesting direction of future work by itself.)

Fig 6 shows the results, where CREDO is able to match any of the different priors. In such scenarios, one can choose a parametrization of the prior that maximizes generalization performance, while respecting domain knowledge.

Is CREDO always useful? When the domain prior also matches the most significant correlations in a dataset, ERM can perform well by itself. We note however that this is not common especially in most real-world datasets. As an example, in the Car evaluation dataset from the UCI ML repository, the task is to predict the acceptability of a car given a set of features. Intuitively, safety levels of a car should have monotonic causal effect on its acceptability. Fig 7 and Table 15 show the results of running ERM and CREDO on this dataset. On closer analysis of the dataset (Fig 8), we observe that among all features in the dataset: Buying, Maintenance, Doors, Persons, Lug_boot, Safety, high correlation is observed between Safety and acceptability. ERM captures this and performs comparably to CREDO under such a scenario.

Time Complexity: We compared the training time of ERM and CREDO on a single NVidia GTX 1080 Ti GPU. On the Boston Housing dataset, ERM takes 36.02secs while CREDO takes 40.40secs to train for 100 epochs. On AutoMPG, ERM takes 16.39secs while
CREDO takes 17.70 secs to train for 50 epochs. CREDO trains in almost the same time as ERM with a marginal increase, while providing the benefit of causal regularization.

**Effect of Choice of $\lambda_1$, Regularization Coefficient:** To study the effect of regularization coeff $\lambda_1$, we again consider the function $f(x, y) = \sin(x) + e^y$, now with inputs $x \in [-2, -1], y \in [-2, -1]$ (Synthetic Tabular 3) (note the change in interval, this allows the domain prior to be an arbitrary shape close to $\sin(x)$ rather than monotonic). Fig 9 shows the plots of ACDE of $x$ learned by the model with and without CREDO with different co-efficients. We notice that while a specific value ($\lambda_1 = 1$) provides the best match with the GT, most other choices for $\lambda_1$ do better than ERM in matching the prior.

**Effect of Incorrect Prior:** To understand the behavior of regularization with an incorrect prior on model performance, we study the MEPS dataset and observe that while making the model match incorrect priors reduces the model accuracy expectedly as presented in Table 16. Our method may not work well when the provided priors are substantially different from the true causal relationships. This may happen in case of a mismatch between domain knowledge and creation of the causal graph, or a faulty understanding of the causal relationships. While we focused our efforts in this work with the assumption that the provided causal priors are true, studying the faulty nature of priors under/using our framework is an interesting direction of future work.

**Correct Shape and Arbitrary Slope** In this experiment, we ask CREDO the question "If one knows that the causal prior is monotonic but not the exact slope, how well CREDO matches the results with true prior?" To this end, we performed the following experiment. We create a synthetic tabular dataset (Synthetic Tabular 4) using the structural equations given below so that the true gradient of the causal effect of $X$ on $Y$ is known, which is 2.

$$W := N_{\mathbb{R}}(0, 1)$$
$$Z := -2W + N_{\mathbb{R}}(4, 1)$$
$$X := 0.5Z + N_{\mathbb{R}}(2, 1)$$
$$Y := 2X + Z + W + N_{\mathbb{R}}(0, 0.1)$$

| Method  | Prior Slope of Race | Test Acc |
|---------|--------------------|----------|
| ERM     | -                  | 86.27    |
| CREDO   | 0                  | 86.04    |
| CREDO   | -2                 | 85.00    |
| CREDO   | 2                  | 83.50    |
| CREDO   | 3                  | 83.40    |

Table 16: Effect of modulating Race prior on MEPS dataset
In the dataset generated using these equations, we use $X, Z, W$ as inputs and $Y$ as output to a neural network, and the input given to CREDO is that the slope is linearly monotonic. We provide different assumed domain priors (slope values) to CREDO, to simulate a setting where the incorrect slope value is provided as input. We make two observations: (i) As we get closer to the true slope, the CREDO classifier’s accuracy improves (Table 17). The highest accuracy is for the assumed slope=2, which is the true slope. Note that our method here has no information about the true slope. (ii) Assuming that only the linear monotonicity property of the gradient was input to CREDO, if we use this simple linear search to find the gradient hyperparameter $\alpha$, our method would return the correct assumed gradient prior=2 since that achieves the highest classification accuracy. With these results, it is evident that the closer our assumed gradient is to the true gradient, the better the accuracy is.

### Appendix G. ACE Algorithm

For all our experiments, qualitative results are obtained using ACE plots. We use the algorithm proposed in [10] for computation of ACE. For completeness, we briefly present the ACE calculation in Algorithm 2. The algorithm summarizes the method to find $E(\hat{Y}_t)$; it is easy to find ACE subsequently as $ACE_T = E(\hat{Y}_T) - E(\hat{Y}_t)$. For more details please refer to [10]. This algorithm depends on a Taylor’s series expansion of neural network output, and hence the use of first-order and second-order gradients in the method.

**Algorithm 2: ACE learned by the neural network**

- **Result:** $E(\hat{Y}_t)$ for each $t = \alpha$
- **Inputs:** $f, t, t$’s range: [low, high], number of interventions: $n$,
  - data mean : $\mu$, data covariance matrix : $\text{cov}$
- **Initialize:** $\text{cov}[t][:] := 0, \text{cov}[:][t] := 0, \alpha = \text{low}, IE := []$
- **while** $\alpha \leq \text{high}$ **do**
  - $\mu[t] = \alpha$
  - $IE.append(f(\mu) + \frac{1}{2}\text{trace(matmul(\nabla^2 f(\mu), \text{cov}))})$
  - $\alpha = \alpha + \frac{\text{high-low}}{n}$
- **end**
- return $IE$

### Appendix H. BNLearn Datasets: Causal Graphs

For the experiments in Sec 4.1 of the main paper (where we know the causal graph), we used SANGIOVESE and MEHRA dagsets from the BNLearn repository [35]. For completeness of understanding, Figs 18, 19 show the causal DAGs that generate these datasets.
Appendix I. Architectural/Training Details

We use a multi-layer perceptron with ReLU non-linearity across our experiments, each trained using ADAM optimizer with Dropout and $L_2$ weight decay. Table 20 shows the details of neural network architectures and training details of our models for various datasets. 80% of the dataset is used for training and remaining 20% for testing.

| S.No. | Dataset          | Dataset Size | Input Size, Output Size | Learning Rate | Batch Size | $\lambda_1$ (ACDE) | Number of Layers | Size of Each Layer |
|-------|------------------|--------------|-------------------------|---------------|------------|-------------------|------------------|--------------------|
| 1     | COMPAS           | 6,172        | 11.2                    | 1e-3          | 64         | 50                | 3                | 16,32,32           |
| 2     | MEPS             | 15,830       | 138.2                   | 1e-3          | 64         | 2                 | 3                | 128,256,256        |
| 3     | Law School       | 20,797       | 15.2                    | 1e-3          | 64         | 2                 | 3                | 64,32,64           |
| 4     | AutoMPG          | 398          | 7.2                     | 1e-2          | 32         | 1.5               | 4                | 16,16,16,16        |
| 5     | Boston Housing   | 506          | 13.2                    | 1e-2          | 64         | 1                 | 4                | 20,20,20,20        |
| 6     | Titanic          | 1,309        | 10.2                    | 1e-3          | 64         | 3                 | 2                | 64,128             |
| 7     | Car Evaluation   | 1,728        | 6.2                     | 1e-3          | 64         | 2                 | 3                | 64,128,128         |
| 8     | Heart Disease    | 303          | 13.2                    | 1e-2          | 64         | 1                 | 3                | 16,16,16           |
| 9     | Adult            | 48842        | 14.2                    | 1e-2          | 1024       | 0.2               | 3                | 64,64,64           |
| 10    | SANGIOVESE       | 10,000       | 29.2                    | 1e-2          | 64         | 2.3               | 3                | 16,16,16           |
| 11    | SACHS            | 10,000       | 10.3                    | 1e-3          | 64         | 10                | 2                | 16,32              |
| 12    | ASIA             | 10,000       | 7.2                     | 1e-3          | 64         | 1                 | 2                | 32,64              |
| 13    | MEHRA            | 10,000       | 152.2                   | 1e-3          | 64         | 2.2               | 3                | 32,32,32           |
| 14    | Synthetic Tabular 1 $z = \log(1 + 2^x), x \in [0, 1]$ | 1,000 | 1.1 | 1e-2 | 64 | 10 | 2 | 4.8 |
| 15    | Synthetic Tabular 2 $z = \sin x + e^y, x, y \in [0, 1]$ | 1,000 | 2.1 | 1e-2 | 64 | 0.5 | 2 | 8.16 |
| 16    | Synthetic Tabular 3 $z = \sin x + e^y, x, y \in [-2, -1]$ | 1,000 | 2.1 | 1e-3 | 64 | 20 | 2 | 8.16 |
| 17    | Synthetic Tabular 4 $W = N(0, 1)$ | 10,000 | 3.2 | 1e-2 | 64 | 1.0 | 3 | 12,12,12 |

Table 20: Architectural and training details of all datasets.
References

[1] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. How we analyzed the compas recidivism algorithm. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm, 2016.

[2] Anonymous. Medical expenditure panel survey. https://meps.ahrq.gov/mepsweb/, 2011.

[3] Norman P Archer and Shouhong Wang. Application of the back propagation neural network algorithm with monotonicity constraints for two-group classification problems. Decision Sciences, 24(1):60–75, 1993.

[4] Mohammad Taha Bahadori, Krzysztof Chalupka, Edward Choi, Robert Chen, Walter F Stewart, and Jimeng Sun. Causal regularization. arXiv preprint arXiv:1702.02604, 2017.

[5] Richard Berk. Accuracy and fairness for juvenile justice risk assessments. Journal of Empirical Legal Studies, 16(1):175–194, 2019.

[6] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. Sociological Methods & Research, 50(1):3–44, 2021.

[7] Tim Brennan and William L Oliver. Emergence of machine learning techniques in criminology: implications of complexity in our data and in research questions. Criminology & Pub. Pol’y, 12:551, 2013.

[8] Stanley L Brue. Retrospectives: The law of diminishing returns. Journal of Economic Perspectives, 7(3):185–192, 1993.

[9] Edward J Calabrese and Linda A Baldwin. U-shaped dose-responses in biology, toxicology, and public health. Annual review of public health, 22(1):15–33, 2001.

[10] Aditya Chattopadhyay, Piyushi Manupriya, Anirban Sarkar, and Vineeth N Balasubramanian. Neural network attributions: A causal perspective. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 981–990, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[11] Hennie Daniels and Marina Velikova. Monotone and partially monotone neural networks. IEEE Transactions on Neural Networks, 21(6):906–917, 2010.

[12] Saloni Dash, Vineeth N Balasubramanian, and Amit Sharma. Evaluating and mitigating bias in image classifiers: A causal perspective using counterfactuals, 2021.

[13] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[14] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating functional knowledge in neural networks. Journal of Machine Learning Research, 10(6), 2009.
[15] Andrew Guthrie Ferguson. Big data and predictive reasonable suspicion. U. Pa. L. Rev., 163:327, 2014.

[16] W. D. Flanders and L. B. Augestad. Adjusting for reverse causality in the relationship between obesity and mortality. International Journal of Obesity (2005), 32 Suppl 3:S42–46, August 2008.

[17] Abigail Fraser, Debbie A. Lawlor, and Laura D. Howe. Nonlinear Exposure-Outcome Associations and Public Health Policy. JAMA, 315(12):1286–1287, March 2016.

[18] Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). arXiv preprint arXiv:1907.07165, 2019.

[19] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. arXiv preprint arXiv:1904.07451, 2019.

[20] Akhil Gupta, Naman Shukla, Lavanya Marla, Arinbjörn Kolbeinsson, and Kartik Yellepeddi. How to incorporate monotonicity in deep networks while preserving flexibility? arXiv preprint arXiv:1909.10662, 2019.

[21] Maya Gupta, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Alexander Mangylov, Wojciech Moczydlowski, and Alexander Van Esbroeck. Monotonic calibrated interpolated look-up tables. J. Mach. Learn. Res., 17(1):3790–3836, January 2016.

[22] Dominik Janzing. Causal regularization. In Advances in Neural Information Processing Systems, pages 12704–12714, 2019.

[23] Daniel Kahneman and Amos Tversky. Choices, values, and frames. In Handbook of the fundamentals of financial decision making: Part I, pages 269–278. World Scientific, 2013.

[24] Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. arXiv preprint arXiv:1706.10244, 2017.

[25] Trent Kyono, Yao Zhang, and Mihaela van der Schaar. Castle: Regularization via auxiliary causal graph discovery. arXiv preprint arXiv:2009.13180, 2020.

[26] Jeffrey Luk, Peter Gross, and William W Thompson. Observations on mortality during the 1918 influenza pandemic. Clinical Infectious Diseases, 33(8):1375–1378, 2001.

[27] Judea Pearl. Direct and indirect effects. In Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence, pages 411–420, 2001.

[28] Judea Pearl. Causality. Cambridge university press, 2009.

[29] Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. arXiv preprint arXiv:2007.02863, 2020.
[30] Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. Interpretations are useful: Penalizing explanations to align neural networks with prior knowledge. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8116–8126, Virtual, 13–18 Jul 2020. PMLR.

[31] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2662–2670, 2017.

[32] Karen Sachs, Omar Perez, Dana Pe’er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

[33] N J Salkind. *Encyclopedia of research design (Vols. 1-0): U-shaped curve*. SAGE Publications, Inc., 2010.

[34] Patrick Schwab, Lorenz Linhardt, Stefan Bauer, Joachim M Buhmann, and Walter Karlen. Learning counterfactual representations for estimating individual dose-response curves. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5612–5619, 2020.

[35] M. Scutari and J.B. Denis. *Bayesian Networks: With Examples in R*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2014.

[36] Shayak Sen, Piotr Mardziel, Anupam Datta, and Matthew Fredrikson. Supervising feature influence. *arXiv preprint arXiv:1803.10815*, 2018.

[37] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

[38] Joseph Sill. Monotonic networks. In *Advances in neural information processing systems*, pages 661–667, 1998.

[39] Aishwarya Sivaraman, Golnoosh Farnadi, Todd Millstein, and Guy Van den Broeck. Counterexample-guided learning of monotonic neural networks. *arXiv preprint arXiv:2006.08852*, 2020.

[40] Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. *arXiv preprint arXiv:1803.00443*, 2018.

[41] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, pages 6056–6065. PMLR, 2019.

[42] Tyler J VanderWeele. Controlled direct and mediated effects: definition, identification and bounds. *Scandinavian Journal of Statistics*, 38(3):551–563, 2011.
Causal Regularization Using Domain Priors

[43] C. Vitolo, M. Scutari, Mohamed Ghalaieny, A. Tucker, and Andrew Russell. Modeling air pollution, climate, and health data using bayesian networks: A case study of the english regions. *Earth and Space Science*, 5:76–88, 2018.

[44] L.F. Wightman, H. Ramsey, and Law School Admission Council. *LSAC National Longitudinal Bar Passage Study*. LSAC research report series. Law School Admission Council, 1998.

[45] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Structured causal disentanglement in variational autoencoder. *arXiv preprint arXiv:2004.08697*, 2020.

[46] Shuo Yang and Sriraam Natarajan. Knowledge intensive learning: Combining qualitative constraints with causal independence for parameter learning in probabilistic models. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 580–595, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[47] Seungil You, David Ding, Kevin Canini, Jan Pfeifer, and Maya Gupta. Deep lattice networks and partial monotonic functions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 2981–2989. Curran Associates, Inc., 2017.

[48] Junzhe Zhang and Elias Bareinboim. Fairness in decision-making—the causal explanation formula. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[49] Shengyu Zhu, Ignavier Ng, and Zhitang Chen. Causal discovery with reinforcement learning. In *International Conference on Learning Representations*, 2020.

[50] Ran Zmigrod, Sabrina J Mielke, Hanna Wallach, and Ryan Cotterell. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. *arXiv preprint arXiv:1906.04571*, 2019.