# Learning Robust Models Using the Principle of Independent Causal Mechanisms

Jens Müller[1,2](✉), Robert Schmier[2,3], Lynton Ardizzone[1,2], Carsten Rother[1,2], and Ullrich Köthe[1,2]

[1] Heidelberg Collaboratory for Image Processing, Heidelberg University, Heidelberg, Germany
jens.mueller@iwr.uni-heidelberg.de
[2] Computer Vision and Learning Lab, Heidelberg University, Heidelberg, Germany
[3] Bosch Center for Artificial Intelligence, Renningen, Germany

**Abstract.** Standard supervised learning breaks down under data distribution shift. However, the principle of independent causal mechanisms (ICM, [31]) can turn this weakness into an opportunity: one can take advantage of distribution shift between different environments during training in order to obtain more robust models. We propose a new gradient-based learning framework whose objective function is derived from the ICM principle. We show theoretically and experimentally that neural networks trained in this framework focus on relations remaining invariant across environments and ignore unstable ones. Moreover, we prove that the recovered stable relations correspond to the true causal mechanisms under certain conditions, turning domain generalization into a causal discovery problem. In both regression and classification, the resulting models generalize well to unseen scenarios where traditionally trained models fail.

**Keywords:** Domain generalization · Principle of independent causal mechanisms

## 1 Introduction

Standard supervised learning has shown impressive results when training and test samples follow the same distribution. However, many real world applications do not conform to this setting, so that research successes do not readily translate into practice [20]. *Domain Generalization* (DG) addresses this problem: it aims at training models that generalize well under domain shift. In contrast to domain *adaption*, where a few labeled and/or many unlabeled examples are provided for each target test domain, in DG absolutely no data is available from the test domains' distributions making the problem unsolvable in general.

In this work, we view the problem of DG specifically using ideas from causal discovery. This viewpoint makes the problem of DG well-posed: we assume that

C. Bauckhage et al. (Eds.): DAGM GCPR 2021, LNCS 13024, pp. 79–110, 2021.
https://doi.org/10.1007/978-3-030-92659-5_6

there exists a feature vector $h^\star(\mathbf{X})$ whose relation to the target variable $Y$ is invariant across all environments. Consequently, the conditional probability $p(Y \mid h^\star(\mathbf{X}))$ has predictive power in each environment. From a causal perspective, changes between domains or environments can be described as interventions; and causal relationships – unlike purely statistical ones – remain invariant across environments unless explicitly changed under intervention. This is due to the fundamental principle of "Independent Causal Mechanisms" which will be discussed in Sect. 3. From a causal standpoint, finding robust models is therefore a *causal discovery* task [4,24]. Taking a causal perspective on DG, we aim at identifying features which (i) have an invariant relationship to the target variable $Y$ and (ii) are maximally informative about $Y$. This problem has already been addressed with some simplifying assumptions and a discrete combinatorial search by [22,35], but we make weaker assumptions and enable gradient based optimization. The later is attractive because it readily scales to high dimensions and offers the possibility to *learn* very informative features, instead of merely selecting among predefined ones. Approaches to invariant relations similar to ours were taken by [10], who restrict themselves to linear relations, and [2,19], who consider a weaker notion of invariance. Problems (i) and (ii) are quite intricate because the search space has combinatorial complexity and testing for conditional independence in high dimensions is notoriously difficult. Our main contributions to this problem are the following: *First*, by connecting invariant (causal) relations with normalizing flows, we propose a differentiable two-part objective of the form $I(Y; h(\mathbf{X})) + \lambda_I \mathcal{L}_I$, where $I$ is the mutual information and $\mathcal{L}_I$ enforces the invariance of the relation between $h(\mathbf{X})$ and $Y$ across all environments. This objective operationalizes the ICM principle with a trade-off between feature informativeness and invariance controlled by parameter $\lambda_I$. Our formulation generalizes existing work because our objective is not restricted to linear models. *Second*, we take advantage of the continuous objective in three important ways: (1) We can learn invariant new features, whereas graph-based methods as in e.g. [22] can only select features from a pre-defined set. (2) Our approach does not suffer from the scalability problems of combinatorial optimization methods as proposed in e.g. [30] and [35]. (3) Our optimization via normalizing flows, i.e. in the form of a density estimation task, facilitates accurate maximization of the mutual information. *Third*, we show how our objective simplifies in important special cases and under which conditions its optimal solution identifies the true causal parents of the target variable $Y$. We empirically demonstrate that the new method achieves good results on two datasets proposed in the literature.

## 2 Related Work

Different types of invariances have been considered in the field of DG. One type is defined on the feature level, i.e. features $h(\mathbf{X})$ are invariant across environments if they follow the same distribution in all environments (e.g. [5,8,27]). However, this form of invariance is problematic since the distribution of the target variable might change between environments, which induces a corresponding change in the distribution of $h(\mathbf{X})$. A more plausible and theoretically justified assumption is the invariance of *relations* [22,30,35]. The relation between a target $Y$ and features $h(\mathbf{X})$ is invariant across environments, if the conditional distribution $p(Y \mid h(\mathbf{X}))$ remains unchanged in all environments. Existing approaches exhaustively model conditional distributions for all possible feature selections and check for the invariance property [22,30,35], which scales poorly for large feature spaces. We derive a theoretical result connecting *normalizing flows* and *invariant relations*, which enables gradient-based learning of an invariant solution. In order to exploit our formulation, we also use the Hilbert-Schmidt-Independence Criterion that has been used for robust learning by [11] in the one environment setting. [2,19,38] also propose gradient-based learning frameworks, which exploit a weaker notion of invariance: They aim to match the conditional expectations across environments, whereas we address the harder problem of matching the entire conditional distributions. The connection between DG, invariance and causality has been pointed out for instance by [24,35,39]. From a causal perspective, DG is a causal discovery task [24]. For studies on causal discovery in the purely observational setting see e.g. [6,29,36], but they do not take advantage of variations across environments. The case of different environments has been studied by [4,9,15,16,22,26,30,37]. Most of these approaches rely on combinatorial optimization or are restricted to linear mechanisms, whereas our continuous objective efficiently optimizes very general non-linear models. The distinctive property of causal relations to remain invariant across environments in the absence of direct interventions has been known since at least the 1930s [7,13]. However, its crucial role as a tool for causal discovery was – to the best of our knowledge– only recently recognized by [30]. Their estimator – *Invariant Causal Prediction* (ICP) – returns the intersection of all subsets of variables that have an invariant relation w.r.t. $Y$. The output is shown to be the set of the direct causes of $Y$ under suitable conditions. Again, this approach requires linear models and exhaustive search over all possible variable sets $\mathbf{X}_S$. Extensions to time series and non-linear additive noise models were studied in [14,33]. Our treatment of invariance is inspired by these papers and also discusses identifiability results, i.e. conditions when the identified variables are indeed the direct causes, with two key differences: Firstly, we propose a formulation that allows for a gradient-based learning and does not need strong assumptions on the underlying causal model. Second, while ICP tends to exclude features from the parent set when in doubt, our algorithm prefers to err towards best predictive performance in this situation.

## 3 Preliminaries

In the following we introduce the basics of this work as well as the connection between DG and causality. Basics on causality are presented in Appendix A. We first define our notation as follows: We denote the set of all variables describing the system under study as $\widetilde{\mathbf{X}} = \{X_1, \ldots, X_D\}$. One of these variables will be singled out as our prediction target, whereas the remaining ones are observed and may serve as predictors. To clarify notation, we call the target variable $Y \equiv X_i$ for some $i \in \{1, \dots, D\}$, and the remaining observations are $\mathbf{X} = \widetilde{\mathbf{X}} \setminus \{Y\}$. Realizations of a random variable (RV) are denoted with lower case letters, e.g. $x_i$. We assume that observations can be obtained in different environments $e \in \mathcal{E}$. Symbols with superscript, e.g. $Y^e$, refer to a specific environment, whereas symbols without refer to data pooled over all environments. We distinguish known environments $e \in \mathcal{E}_{\text{seen}}$, where training data are available, from unknown ones $e \in \mathcal{E}_{\text{unseen}}$, where we wish our models to generalize to. The set of all environments is $\mathcal{E} = \mathcal{E}_{\text{seen}} \cup \mathcal{E}_{\text{unseen}}$. We assume that all RVs have a density $p_A$ with probability distribution $P_A$ (for some variable or set $A$). We consider the environment to be a RV $E$ and therefore a system variable similar to [26]. This gives an additional view on causal discovery and the DG problem. Independence and dependence of two variables $A$ and $B$ is written as $A \perp B$ and $A \not\perp B$ respectively. Two RVs $A, B$ are conditionally independent given $C$ if $P(A, B \mid C) = P(A \mid C)P(B \mid C)$. This is denoted with $A \perp B \mid C$. It means $A$ does not contain any information about $B$ if $C$ is known (see e.g. [31]). Similarly, one can define independence and conditional independence for sets of RVs.

### 3.1 Invariance and the Principle of ICM

DG is in general unsolvable because distributions between seen and unseen environments could differ arbitrarily. In order to transfer knowledge from $\mathcal{E}_{\text{seen}}$ to $\mathcal{E}_{\text{unseen}}$, we have to make assumptions on how seen and unseen environments relate. These assumptions have a close link to causality. We assume certain relations between variables remain invariant across all environments. A subset $\mathbf{X}_S \subset \mathbf{X}$ of variables *elicits an invariant relation* or *satisfies the invariance property* w.r.t. $Y$ over a subset $W \subset \mathcal{E}$ of environments if

$$\forall e, e' \in W: \quad P(Y^e \mid \mathbf{X}_S^e = u) = P(Y^{e'} \mid \mathbf{X}_S^{e'} = u) \tag{1}$$

for all $u$ where both conditional distributions are well-defined. Equivalently, we can define the invariance property by $Y \perp E \mid \mathbf{X}_S$ and $I(Y; E \mid \mathbf{X}_S) = 0$ for $E$ restricted to $W$. The *invariance property* for computed features $h(\mathbf{X})$ is defined analogously by the relation $Y \perp E \mid h(\mathbf{X})$. Although we can only test for Eq. 1 in $\mathcal{E}_{\text{seen}}$, taking a causal perspective allows us to derive plausible conditions for an invariance to remain valid in all environments $\mathcal{E}$. In brief, we assume that environments correspond to interventions in the system and invariance arises from the principle of *independent causal mechanisms* [31]. We specify these conditions later in Assumption 1 and 2. At first, consider the joint density $p_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{X}})$.The chain rule offers a combinatorial number of ways to decompose this distribution into a product of conditionals. Among those, the *causal factorization*

$$p_{\widetilde{\mathbf{X}}}(x_1, \dots, x_D) = \textstyle\prod_{i=1}^{D} p_i(x_i \mid \mathbf{x}_{pa(i)}) \tag{2}$$

is singled out by conditioning each $X_i$ onto its *direct causes* or *causal parents* $\mathbf{X}_{pa(i)}$, where $pa(i)$ denotes the appropriate index set. The special properties of this factorization are discussed in [31]. The conditionals $p_i$ of the causal factorization are called *causal mechanisms*. An *intervention* onto the system is defined by replacing one or several factors in the decomposition with different (conditional) densities $\overline{p}$. Here, we distinguish *soft-interventions* where $\overline{p}_j(x_j \mid \mathbf{x}_{pa(j)}) \neq p_j(x_j \mid \mathbf{x}_{pa(j)})$ and *hard-interventions* where $\overline{p}_j(x_j \mid \mathbf{x}_{pa(j)}) = \overline{p}_j(x_j)$ is a density which does not depend on $x_{pa(j)}$ (e.g. an atomic intervention where $x_j$ is set to a specific value $\overline{x}$). The resulting joint distribution for a single intervention is

$$\overline{p}_{\widetilde{\mathbf{X}}}(x_1, \dots, x_D) = \overline{p}_j(x_j \mid \mathbf{x}_{pa(j)}) \textstyle\prod_{i=1, i \neq j}^{D} p_i(x_i \mid \mathbf{x}_{pa(i)}) \tag{3}$$

and extends to multiple simultaneous interventions in the obvious way. The principle of *independent causal mechanisms* (ICM) states that every mechanism acts independently of the others [31]. Consequently, an intervention replacing $p_j$ with $\overline{p}_j$ has no effect on the other factors $p_{i \neq j}$, as indicated by Eq. 3. This is a crucial property of the causal decomposition – alternative factorizations do not exhibit this behavior. Instead, a coordinated modification of several factors is generally required to model the effect of an intervention in a non-causal decomposition. We utilize this principle as a tool to train *robust* models. To do so, we make two additional assumptions, similar to [30] and [14]:

**Assumption** ***(1)*** *Any differences in the joint distributions $p^e_{\widetilde{\mathbf{X}}}$ from one environment to the other are fully explainable as interventions: replacing factors $p^e_i(x_i \mid \mathbf{x}_{pa(i)})$ in environment $e$ with factors $p^{e'}_i(x_i \mid \mathbf{x}_{pa(i)})$ in environment $e'$ (for some subset of the variables) is the only admissible change.* ***(2)*** *The mechanism $p(y \mid \mathbf{x}_{pa(Y)})$ for the target variable $Y$ is invariant under changes of environment, i.e. we require conditional independence $Y \perp E \mid \mathbf{X}_{pa(Y)}$.*

Assumption 2 implies that $Y$ must not directly depend on $E$. Consequences in case of omitted variables are discussed in Appendix B. If we knew the causal decomposition, we could use these assumptions directly to train a robust model for $Y$ – we would simply regress $Y$ on its parents $\mathbf{X}_{pa(Y)}$. However, we only require that a causal decomposition with these properties exists, but do not assume that it is known. Instead, our method uses the assumptions indirectly – by simultaneously considering data from different environments – to identify a stable regressor for $Y$. We call a regressor stable if it solely relies on predictors whose relationship to $Y$ remains invariant across environments, i.e. is not influenced by any intervention. By assumption 2, such a regressor always exists. However, predictor variables beyond $\mathbf{X}_{pa(Y)}$, e.g. children of $Y$ or parents of children, may be included into our model as long as their relationship to $Y$ remains invariant across all environments. We discuss this and further illustrate Assumption 2 in Appendix B. In general, prediction accuracy will be maximized when all suitable predictor variables are included into the model. Accordingly, our algorithm will asymptotically identify the full set of stable predictors for $Y$. In addition, we will prove under which conditions this set contains exactly the parents of $Y$.

### 3.2 Domain Generalization

To exploit the principle of ICM for DG, we formulate the DG problem as follows

$$h^\star := \underset{h \in \mathcal{H}}{\arg\max} \left\{ \min_{e \in \mathcal{E}} I(Y^e; h(\mathbf{X}^e)) \right\} \qquad \text{s.t.} \quad Y \perp E \mid h(\mathbf{X}) \tag{4}$$

The optimization problem in Eq. 4 asks to find features $h(\mathbf{X})$ which are maximally informative in the worst environment subject to the invariance constraint. where $h \in \mathcal{H}$ denotes a learnable feature extraction function $h\colon \mathbb{R}^D \to \mathbb{R}^M$ where $M$ is a hyperparameter. This optimization problem defines a maximin objective: The features $h(\mathbf{X})$ should be as informative as possible about the response $Y$ even in the most difficult environment, while conforming to the ICM constraint that the relationship between features and response must remain invariant across all environments. In principle, our approach can also optimize related objectives like the average mutual information over environments. However, very good performance in a majority of the environments could then mask failure in a single (outlier) environment. We opted for the maximin formulation to avoid this. On the other hand there might be scenarios where the maxmin formulation is limited. For instance when the training signal is very noisy in one environment, the classifier might discard valuable information from the other environments. As it stands, Eq. 4 is hard to optimize, because traditional independence tests for the constraint $Y \perp E \mid h(\mathbf{X})$ cannot cope with conditioning variables selected from a potentially infinitely large space $\mathcal{H}$. A re-formulation of the DG problem to circumvent these issues is our main theoretical contribution.

### 3.3 Normalizing Flows

Normalizing flows form a class of probabilistic models that has recently received considerable attention, see e.g. [28]. They model complex distributions by means of invertible functions $T$ (chosen from some model space $\mathcal{T}$), which map the densities of interest to latent normal distributions. Normalizing flows are typically built with specialized neural networks that are invertible by construction and have tractable Jacobian determinants. We represent the conditional distribution $P(Y \mid h(\mathbf{X}))$ by a *conditional* normalizing flow (see e.g. [1]). The literature typically deals with Structural Causal Models restricted to additive noise. With normalizing flows, we are able to lift this restriction to the much broader setting of arbitrary distributions (for details see Appendix C). The corresponding loss is the negative log-likelihood (NLL) of $Y$ under $T$, given by

$$\mathcal{L}_{\mathrm{NLL}}(T,h) := \mathbb{E}_{h(\mathbf{X}),Y}\left[\|T(Y;h(\mathbf{X})\|^2/2 - \log|\det \nabla_y T(Y;h(\mathbf{X}))|\right] + C \quad (5)$$

where $\det \nabla_y T$ is the Jacobian determinant and $C = \dim(Y)\log(\sqrt{2\pi})$ is a constant that can be dropped [28]. Equation 5 can be derived from the change of variables formula and the assumption that $T$ maps to a standard normal distribution [28]. If we consider the NLL on a particular environment $e \in \mathcal{E}$, we denote this with $\mathcal{L}^e_{\mathrm{NLL}}$. Lemma 1 shows that normalizing flows optimized by NLL are indeed applicable to our problem:

**Lemma 1.** *(proof in Appendix C) Let* $h^\star, T^\star := \arg\min_{h\in\mathcal{H},T\in\mathcal{T}} \mathcal{L}_{\mathrm{NLL}}(T,h)$ *be the solution of the NLL minimization problem on a sufficiently rich function space* $\mathcal{T}$*. Then the following properties hold for any set* $\mathcal{H}$ *of feature extractors:*

*(a)* $h^\star$ *also maximizes the mutual information, i.e.* $h^\star = \arg\max_{g\in\mathcal{H}} I(g(\mathbf{X});Y)$

*(b) $h^\star$ and the latent variables $R = T^\star(Y; h^\star(\mathbf{X}))$ are independent: $h^\star(\mathbf{X}) \perp R$*

Statement (a) guarantees that $h^\star$ extracts as much information about $Y$ as possible. Hence, the objective (4) becomes equivalent to optimizing (5) when we restrict the space $\mathcal{H}$ of admissible feature extractors to the subspace $\mathcal{H}_\perp$ satisfying the invariance constraint $Y \perp E \mid h(\mathbf{X})$: $\arg\min_{h\in\mathcal{H}_\perp} \max_{e\in\mathcal{E}} \min_{T\in\mathcal{T}} \mathcal{L}^e_{\mathrm{NLL}}(T; h) = \arg\max_{h\in\mathcal{H}_\perp} \min_{e\in\mathcal{E}} I(Y^e; h(\mathbf{X}^e))$ (Appendix C). Statement (b) ensures that the flow indeed implements a valid structural equation, which requires that $R$ can be sampled independently of the features $h(\mathbf{X})$.

## 4 Method

In the following we propose a way of indirectly expressing the constraint in Eq. 4 via normalizing flows. Thereafter, we combine this result with Lemma 1 to obtain a differentiable objective for solving the DG problem. We also present important simplifications for least squares regression and softmax classification and discuss relations of our approach with causal discovery.

### 4.1 Learning the Invariance Property

The following theorem establishes a connection between invariant relations, prediction residuals and normalizing flows. The key consequence is that a suitably trained normalizing flow translates the statistical independence of the latent variable $R$ from the features and environment $(h(\mathbf{X}), E)$ into the desired invariance of the mechanism $P(Y \mid h(\mathbf{X}))$ under changes of $E$. We will exploit this for an elegant reformulation of the DG problem (4) into the objective (7) below.

**Theorem 1.** *Let $h$ be a differentiable function and $Y, \mathbf{X}, E$ be RVs. Furthermore, let $R = T(Y; h(\mathbf{X}))$ be a continuous, differentiable function that is a diffeomorphism in $Y$. Suppose that $R \perp (h(\mathbf{X}), E)$. Then, it holds that $Y \perp E \mid h(\mathbf{X})$.*

*Proof.* The decomposition rule for the assumption (i) $R \perp (h(\mathbf{X}), E)$ implies (ii) $R \perp h(\mathbf{X})$. To simplify notation, we define $Z := h(\mathbf{X})$. Because $T$ is invertible in $Y$ and due to the change of variables (c.o.v.) formula, we obtain

$$p_{Y|Z,E}(y \mid z, e) \overset{(c.o.v.)}{=} p_{R|Z,E}(T(y,z) \mid z, e) \left|\det \frac{\partial T}{\partial y}(y,z)\right|$$
$$\overset{(i)}{=} \; p_R(r) \left|\det \frac{\partial T}{\partial y}(y,z)\right| \overset{(ii)}{=} p_{R|Z}(r \mid z) \left|\det \frac{\partial T}{\partial y}(y,z)\right| \overset{(c.o.v.)}{=} p_{Y|Z}(y \mid z).$$

This implies $Y \perp E \mid Z$. The theorem states in particular that if there exists a suitable diffeomorphism $T$ such that $R \perp (h(\mathbf{X}), E)$, then $h(\mathbf{X})$ satisfies the invariance property w.r.t. $Y$. Note that if Assumption 2 is violated, the condition $R \perp (h(\mathbf{X}), E)$ is unachievable in general and therefore the theorem is not applicable (see Appendix B). We use Theorem 1 in order to *learn* features

$h$ that meet this requirement. In the following, we denote a conditional normalizing flow parameterized via $\theta$ with $T_\theta$. Furthermore, $h_\phi$ denotes a feature extractor implemented as a neural network parameterized via $\phi$. We can relax condition $R \perp (h_\phi(\mathbf{X}), E)$ by means of the Hilbert Schmidt Independence Criterion (HSIC), a kernel-based independence measure (see Appendix D for the definition and [12] for details). This loss, denoted as $\mathcal{L}_I$, penalizes dependence between the distributions of $R$ and $(h_\phi(\mathbf{X}), E)$. The HSIC guarantees that

$$\mathcal{L}_I\big(P_R, P_{h_\phi(\mathbf{X}),E}\big) = 0 \quad \iff \quad R \perp (h_\phi(\mathbf{X}), E) \tag{6}$$

where $R = T_\theta(Y; h_\phi(\mathbf{X}))$ and $P_R, P_{h_\phi(\mathbf{X}),E}$ are the distributions implied by the parameter choices $\phi$ and $\theta$. Due to Theorem 1, minimization of $\mathcal{L}_I(P_R, P_{h_\phi(\mathbf{X}),E})$ w.r.t. $\phi$ and $\theta$ will thus approximate the desired invariance property $Y \perp E \mid h_\phi(\mathbf{X})$, with exact validity upon perfect convergence. When $R \perp (h_\phi(\mathbf{X}), E)$ is fulfilled, the decomposition rule implies $R \perp E$ as well. However, if the differences between environments are small, empirical convergence is accelerated by adding a Wasserstein loss which enforces the latter (see Appendix D and Sect. 5.2).

### 4.2 Exploiting Invariances for Prediction

Equation 4 can be re-formulated as a differentiable loss using a Lagrange multiplier $\lambda_I$ on the HSIC loss. $\lambda_I$ acts as a hyperparameter to adjust the trade-off between the invariance property of $h_\phi(\mathbf{X})$ w.r.t. $Y$ and the mutual information between $h_\phi(\mathbf{X})$ and $Y$. See Appendix F for algorithm details. In the following, we consider normalizing flows in order to optimize Eq. 4. Using Lemma 1(a), we maximize $\min_{e\in\mathcal{E}} I(Y^e; h_\phi(\mathbf{X}^e))$ by minimizing $\max_{e\in\mathcal{E}}\{\mathcal{L}_{\text{NLL}}(T_\theta; h_\phi)\}$ w.r.t. $\phi, \theta$. To achieve the described trade-off between goodness-of-fit and invariance, we therefore optimize

$$\arg\min_{\theta,\phi}\Big(\max_{e\in\mathcal{E}}\Big\{\mathcal{L}^e_{\text{NLL}}(T_\theta, h_\phi)\Big\} + \lambda_I\mathcal{L}_I(P_R, P_{h_\phi(\mathbf{X}),E})\Big) \tag{7}$$

where $R^e = T_\theta(Y^e, h_\phi(\mathbf{X}^e))$ and $\lambda_I > 0$. The first term maximizes the mutual information between $h_\phi(\mathbf{X})$ and $Y$ in the environment where the features are least informative about $Y$ and the second term aims to ensure an invariant relation. In the special case that the data is governed by additive noise, Eq. 7 simplifies: Let $f_\theta$ be a regression function, then solving for the noise term gives $Y - f_\theta(\mathbf{X})$ which corresponds to a diffeomorphism in $Y$, namely $T_\theta(Y; X) = Y - f_\theta(\mathbf{X})$. Under certain assumptions (see Appendix E) we obtain an approximation of Eq. 7 via

$$\arg\min_{\theta}\Big(\max_{e\in\mathcal{E}_{\text{seen}}}\Big\{\mathbb{E}\big[(Y^e - f_\theta(\mathbf{X}^e))^2\big]\Big\} + \lambda_I\mathcal{L}_I(P_R, P_{f_\theta(\mathbf{X}),E})\Big) \tag{8}$$

where $R^e = Y^e - f_\theta(\mathbf{X}^e)$ and $\lambda_I > 0$. Here, $\arg\max_\theta I(f_\theta(\mathbf{X}^e), Y^e)$ corresponds to the argmin of the L2-Loss in the corresponding environment. Alternatively we can view the problem as to find features $h_\phi\colon \mathbb{R}^D \to \mathbb{R}^m$ such that $I(h_\phi(\mathbf{X}), Y)$ gets maximized under the assumption that there exists a model $f_\theta(h_\phi(\mathbf{X})) + R = Y$ where $R$ is independent of $h_\phi(\mathbf{X})$ and is Gaussian. In this case we obtain the learning objective

$$\arg\min_{\theta,\phi}\left(\max_{e\in\mathcal{E}_{\text{seen}}}\left\{\mathbb{E}\left[(Y^e - f_\theta(h_\phi(\mathbf{X}^e)))^2\right]\right\} + \lambda_I \mathcal{L}_I(P_R, P_{h_\phi(\mathbf{X}),E})\right) \tag{9}$$

For the classification case, we consider the expected cross-entropy loss

$$-\mathbb{E}_{\mathbf{X},Y}\left[f(\mathbf{X})_Y - \log\left(\sum_c \exp\left(f(\mathbf{X})_c\right)\right)\right] \tag{10}$$

where $f\colon \mathcal{X} \to \mathbb{R}^m$ returns the logits. Minimizing the expected cross-entropy loss amounts to maximizing the mutual information between $f(\mathbf{X})$ and $Y$ [3,34, Eq. 3]. We set $T(Y; f(\mathbf{X})) = Y \cdot \mathrm{softmax}(f(\mathbf{X}))$ with component-wise multiplication. Then $T$ is invertible in $Y$ conditioned on the softmax output and therefore Theorem 1 is applicable. Now we can apply the same invariance loss as above in order to obtain a solution to Eq. 4.

### 4.3 Relation to Causal Discovery

Under certain conditions, solving Eq. 4 leads to features which correspond to the direct causes of $Y$ (identifiability). In this case we obtain the causal mechanism by computing the conditional distribution of $Y$ given the direct causes. Hence Eq. 4 can be seen as an approximation of the causal mechanism when the identifiability conditions are met. The following Proposition states the conditions when the direct causes of $Y$ can be found by exploiting Theorem 1.

**Proposition 1.** *We assume that the underlying causal graph $G$ is faithful with respect to $P_{\widetilde{\mathbf{X}},E}$. We further assume that every child of $Y$ in $G$ is also a child of $E$ in $G$. A variable selection $h(\mathbf{X}) = \mathbf{X}_S$ corresponds to the direct causes if the following conditions are met: (i) $T(Y; h(\mathbf{X})) \perp E, h(\mathbf{X})$ is satisfied for a diffeomorphism $T(\cdot; h(\mathbf{X}))$, (ii) $h(\mathbf{X})$ is maximally informative about $Y$ and (iii) $h(\mathbf{X})$ contains only variables from the Markov blanket of $Y$.*

The Markov blanket of $Y$ is the only set of vertices which are necessary to predict $Y$ (see Appendix A). We give a proof of Proposition 1 as well as a discussion in Appendix G. To facilitate explainability and explicit causal discovery, we employ the same gating function and complexity loss as in [17]. The gating function $h_\phi$ is a 0-1 mask that marks the selected variables, and the complexity loss $\mathcal{L}(h_\phi)$ is a soft counter of the selected variables. Intuitively speaking, if we search for a variable selection that conforms to the conditions in Proposition 1, the complexity loss will exclude all non-task relevant variables. Therefore, if $\mathcal{H}$ is the set of gating functions, then $h^\star$ in Eq. 4 corresponds to the direct causes of $Y$ under the conditions listed in Proposition 1. The complexity loss as well as the gating function can be optimized by gradient descent.

## 5 Experiments

The main focus of this work is on the theoretical and methodological improvements of causality-based domain generalization using information theoretical concepts. A complete and rigorous quantitative evaluation is beyond the scope of this work. In the following we demonstrate proof-of-concept experiments.

### 5.1 Synthetic Causal Graphs

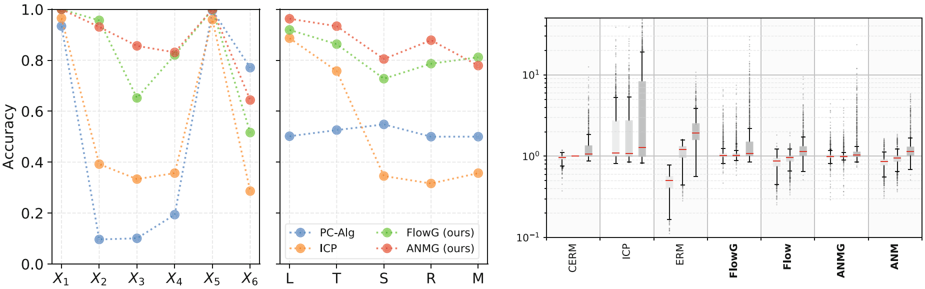


**Fig. 1.** (a) Detection accuracy of the direct causes for baselines and our gating architectures, broken down for different target variables (left) and mechanisms (right: **L**inear, **T**anhshrink, **S**oftplus, **R**eLU, **M**ultipl. Noise) (b) Logarithmic plot of L2 errors, normalized by CERM test error. For each method (ours in bold) from left to right: training error, test error on seen environments, domain generalization error on unseen environments.

To evaluate our methods for the regression case, we follow the experimental design of [14]. It rests on the causal graph in Fig. 2. Each variable $X_1, ..., X_6$ is chosen as the regression target $Y$ in turn, so that a rich variety of local configurations around $Y$ is tested. The corresponding structural equations are selected among four model types of the form $f(\mathbf{X}_{pa(i)}, N_i) = \sum_{j \in pa(i)} \texttt{mech}(a_j X_j) + N_i$, where `mech` is either the identity (hence we get a linear Structural Causal Model (SCM)), Tanhshrink, Softplus or ReLU, and one multiplicative noise mechanism of the form $f_i(\mathbf{X}_{pa(i)}, N_i) = (\sum_{j \in pa(i)} a_j X_j) \cdot (1 + (1/4)N_i) + N_i$, resulting in 1365 different settings. For each setting, we define one observational environment (using exactly the selected mechanisms) and three interventional ones, where soft or do-interventions are applied to non-target variables according to Assumptions 1 and 2 (full details in Appendix H). Each inference model is trained on 1024 realizations of three environments, whereas the fourth one is held back for DG testing. The tasks are to identify the parents of the current target variable $Y$, and to train a transferable regression model based on this parent hypothesis. We measure performance by the accuracy of the detected parent sets and by the L2 regression errors relative to the regression function using the ground-truth parents. We evaluate four models derived from our theory: two normalizing flows as in Eq. 7 with and without gating mechanisms (FlowG, Flow) and two additive noise models, again with and without gating mechanism (ANMG, ANM), using a feed-forward network with the objective in Eq. 9 (ANMG) and Eq. 8 (ANM).

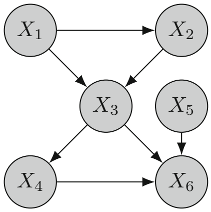


**Fig. 2.** Directed graph of our SCM. Target variable $Y$ is chosen among $X_1, \ldots, X_6$ in turn.

For comparison, we train three baselines: ICP (a causal discovery algorithm also exploiting ICM, but restricted to linear regression, [30]), a variant of the PC-Algorithm (PC-Alg, see Appendix H.4) and standard empirical-risk-minimization ERM, a feed-forward network minimizing the L2-loss, which ignores the causal structure by regressing $Y$ on all other variables. We normalize our results with a ground truth model (CERM), which is identical to ERM, but restricted to the true causal parents of the respective $Y$. The accuracy of parent detection is shown in Fig. 1a. The score indicates the fraction of the experiments where the exact set of all causal parents was found and all non-parents were excluded. We see that the PC algorithm performs unsatisfactorily, whereas ICP exhibits the expected behavior: it works well for variables without parents and for linear SCMs, i.e. exactly within its specification. Among our models, only the gating ones explicitly identify the parents. They clearly outperform the baselines, with a slight edge for ANMG, as long as its assumption of additive noise is fulfilled. Figure 1b and Table 1 report regression errors for seen and unseen environments, with CERM indicating the theoretical lower bound. The PC algorithm is excluded from this experiment due to its poor detection of the direct causes. ICP wins for linear SCMs, but otherwise has largest errors, since it cannot accurately account for non-linear mechanisms. ERM gives reasonable test errors (while overfitting the training data ), but generalizes poorly to unseen environments, as expected. Our models perform quite similarly to CERM. We again find a slight edge for ANMG, except under multiplicative noise, where ANMG's additive noise assumption is violated and Flow is superior. All methods (including CERM) occasionally fail in the domain generalization task, indicating that some DG problems are more difficult than others, e.g. when the differences between seen environments are too small to reliably identify the invariant mechanism or the unseen environment requires extrapolation beyond the training data boundaries. Models without gating (Flow, ANM) seem to be slightly more robust in this respect. A detailed analysis of our experiments can be found in Appendix H.

### 5.2 Colored MNIST

To demonstrate that our model is able to perform DG in the classification case, we use the same data generating process as in the colored variant of the MNIST-dataset established by [2], but create training instances online rather than upfront. The response is reduced to two labels – 0 for all images with digit $\{0, \ldots, 4\}$ and 1 for digits $\{5, \ldots 9\}$ – with deliberate label noise that limits the achievable shape-based classification accuracy to 75%. To confuse the classifier, digits are additionally colored such that colors are spuriously associated with the true labels at accuracies of 90% resp. 80% in the first two environments, whereas the association is only 10% correct in the third environment. A classifier naively trained on the first two environments will identify color as the best predictor,

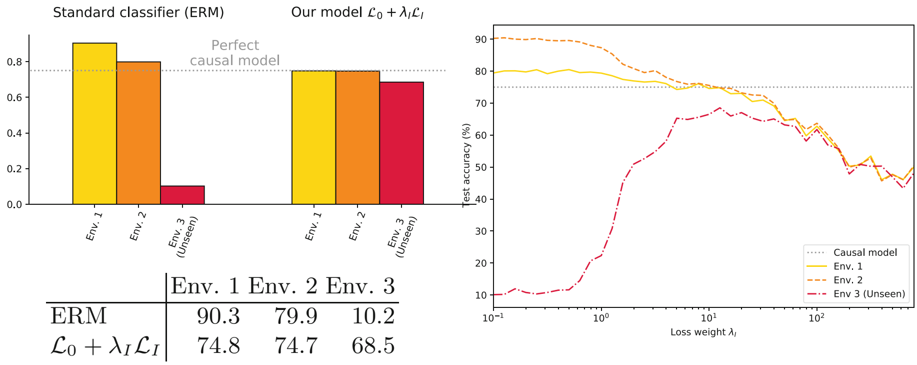
**Table 1.** Medians and upper 95% quantiles for domain generalization L2 errors (i.e. on unseen environments) for different model types and data-generating mechanisms (lower is better).

| Models | Linear | Tanhshrink | Softplus | ReLU | Mult. Noise |
|---|---|---|---|---|---|
| FlowG (ours) | 1.05...4.2 | 1.08...4.8 | 1.09...5.52 | 1.08...5.7 | 1.55...8.64 |
| ANMG (ours) | 1.02...1.56 | **1.03**...2.23 | **1.04**...4.66 | **1.03**...4.32 | 1.46...4.22 |
| Flow (ours) | 1.08...1.61 | 1.14...1.57 | 1.14...1.55 | 1.14...1.54 | **1.35**...4.07 |
| ANM (ours) | 1.05...1.52 | 1.15...1.47 | 1.14...1.47 | 1.15...1.54 | 1.48...4.19 |
| ICP (Peters et al. 2016) | **0.99**...25.7 | 1.44...20.39 | 3.9...23.77 | 4.37...23.49 | 8.94...33.49 |
| ERM | 1.79...3.84 | 1.89...3.89 | 1.99...3.71 | 2.01...3.62 | 2.08...5.86 |
| CERM (true parents) | 1.06...1.89 | 1.06...1.84 | 1.06...2.11 | 1.07...2.15 | 1.37...5.1 |

but will perform terribly when tested on the third environment. In contrast, a robust model will ignore the unstable relation between colors and labels and use the invariant relation, namely the one between digit shapes and labels, for prediction. We supplement the HSIC loss with a Wasserstein term to explicitly enforce $R \perp E$, i.e. $\mathcal{L}_I = \text{HSIC} + \text{L2}(\text{sort}(R^{e_1}), \text{sort}(R^{e_2}))$ (see Appendix D). This gives a better training signal as the HSIC alone, since the difference in label-color association between environments 1 and 2 (90% vs. 80%) is deliberately chosen very small to make the task hard to learn. Experimental details can be found in Appendix I. Figure 3a shows the results for our model: Naive training ($\lambda_I = 0$, i.e. invariance of residuals is not enforced) gives accuracies corresponding to the association between colors and labels and thus completely fails in test environment 3. In contrast, our model performs close to the best possible rate for invariant classifiers in environments 1 and 2 and still achieves 68.5% in environment 3. This is essentially on par with preexisting methods. For instance, IRM achieves 71% on the third environment for this particular dataset, although the dataset itself is not particularly suitable for meaningful quantitative comparisons. Figure 3b demonstrates the trade-off between goodness of fit in the training environments 1 and 2 and the robustness of the resulting classifier: the model's ability to perform DG to the unseen environment 3 improves as $\lambda_I$ increases. If $\lambda_I$ is too large, it dominates the classification training signal and performance breaks down in all environments. However, the choice of $\lambda_I$ is not critical, as good results are obtained over a wide range of settings.

## 6 Discussion

In this paper, we have introduced a new method to find invariant and causal models by exploiting the principle of ICM. Our method works by gradient descent in contrast to combinatorial optimization procedures. This circumvents scalability issues and allows us to extract invariant features even when the raw data representation is not in itself meaningful (e.g. we only observe pixel values). In comparison to alternative approaches, our use of normalizing flows places fewer restrictions on the underlying true generative process. We have also shown under



| | Env. 1 | Env. 2 | Env. 3 |
|---|---|---|---|
| ERM | 90.3 | 79.9 | 10.2 |
| $\mathcal{L}_0 + \lambda_I \mathcal{L}_I$ | 74.8 | 74.7 | 68.5 |

**Fig. 3.** (a) Accuracy of a standard classifier and our model (b) Performance of the model in the three environments, depending on the hyperparameter $\lambda_I$.

which circumstances our method guarantees to find the underlying causal model. Moreover, we demonstrated theoretically and empirically that our method is able to learn robust models w.r.t. distribution shift. Future work includes ablations studies in order to improve the understanding of the influence of single components, e.g. the choice of the maxmin objective over the average mutual information or the Wasserstein loss and the HSIC loss. As a next step, we will examine our approach in more complex scenarios where, for instance, the invariance assumption may only hold approximately.

**Acknowledgments.** JM received funding by the Heidelberg Collaboratory for Image Processing (HCI). LA received funding by the Federal Ministry of Education and Research of Germany project High Performance Deep Learning Framework (No 01IH 17002). CR and UK received financial support from the European Research Council (ERC) under the European Unions Horizon2020 research and innovation program (grant agreement No647769).

Furthermore, we thank our colleagues Felix Draxler, Jakob Kruse and Michael Aichmüller for their help, support and fruitful discussions.

# Appendix

## A Causality: Basics

*Structural Causal Models.* (SCM) allow us to express causal relations on a functional level. Following [31] we define a SCM in the following way:

**Definition 1.** *A Structural Causal Model (SCM) $\mathcal{S} = (S, P_{\boldsymbol{N}})$ consists of a collection $S$ of $D$ (structural) assignments*

$$X_j := f_j(\widetilde{\mathbf{X}}_{pa(j)}, N_j), \quad j = 1, \ldots, D \tag{11}$$

*where $pa(j) \subset \{1, \ldots, j-1\}$ are called parents of $X_j$. $P_{\boldsymbol{N}}$ denotes the distribution over the noise variables $\boldsymbol{N} = (N_1, \ldots, N_D)$ which are assumed to be jointly independent.*

An SCM defined as above produces an acyclic graph $G$ and induces a probability distribution over $P_{\widetilde{\mathbf{X}}}$ which allows for the *causal factorization* as in Eq. 3 [31]. Children of $X_i$ in $G$ are denoted as $ch(i)$ or $ch(X_i)$. An SCM satisfies the *causal sufficiency* assumption if all the noise variables in Definition 1 are indeed jointly independent. A random variable $H$ in the SCM is called *confounder* between two variables $X_i, X_j$ if it causes both of them. If a confounder is not observed, we call it hidden confounder. If there exists a hidden confounder, the causal sufficiency assumption is violated.

The random variables in an SCM correspond to vertices in a graph and the structural assignments $S$ define the edges of this graph. Two sets of vertices $\boldsymbol{A}, \boldsymbol{B}$ are said to be $d$-separated if there exists a set of vertices $\boldsymbol{C}$ such that every path between $\boldsymbol{A}$ and $\boldsymbol{B}$ is blocked. For details see e.g. [31]. The subscript $\perp_d$ denotes $d$-separability which in this case is denoted by $\boldsymbol{A} \perp_d \boldsymbol{B}$. An SCM generates a probability distribution $P_{\widetilde{\mathbf{X}}}$ which satisfies the *Causal Markov Condition*, that is $\boldsymbol{A} \perp_d \boldsymbol{B} \mid \boldsymbol{C}$ results in $\boldsymbol{A} \perp \boldsymbol{B} \mid \boldsymbol{C}$ for sets or random variables $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \subset \widetilde{\mathbf{X}}$. The Causal Markov Condition can be seen as an inherent property of a causal system which leaves marks in the data distribution.

A distribution $P_{\widetilde{\mathbf{X}}}$ is said to be *faithful* to the graph $G$ if $\boldsymbol{A} \perp \boldsymbol{B} \mid \boldsymbol{C}$ results in $\mathbf{A} \perp_d \boldsymbol{B} \mid \boldsymbol{C}$ for all $\mathbf{A}, \mathbf{B}, \mathbf{C} \subset \widetilde{\mathbf{X}}$. This means from the distribution $P_{\widetilde{\mathbf{X}}}$ statements about the underlying graph $G$ can be made.

Assuming both, faithfulness and the Causal Markov condition, we obtain that the $d$-separation statements in $G$ are equivalent to the conditional independence statements in $P_{\widetilde{\mathbf{X}}}$. These two assumptions allow for a whole class of causal discovery algorithms like the PC- or IC-algorithm [29,36].

The smallest set $\boldsymbol{M}$ such that $Y \perp_d \mathbf{X} \setminus (\{Y\} \cup \boldsymbol{M})$ is called *Markov Blanket*. It is given by $\boldsymbol{M} = \mathbf{X}_{pa(Y)} \cup \mathbf{X}_{ch(Y)} \cup \mathbf{X}_{pa(ch(Y))} \setminus \{Y\}$. The *Markov Blanket* of $Y$ is the only set of vertices which are necessary to predict $Y$.

# B Discussion and Illustration of Assumptions

## B.1 Causal Sufficiency and Omitted Variables

Assumption 2 implies that $Y$ must not directly depend on $E$. In addition, it has important consequences when there exist omitted variables $\mathbf{W}$, which influence $Y$ but have not been measured. Specifically, if the omitted variables depend on the environment (hence $\mathbf{W} \not\perp E$) or $\mathbf{W}$ contains a hidden confounder of $\mathbf{X}_{pa(Y)}$ and $Y$ while $\mathbf{X}_{pa(Y)} \not\perp E$ (the system is not causally sufficient and $\mathbf{X}_{pa(Y)}$ becomes a "collider", hence $\mathbf{W} \not\perp E \mid \mathbf{X}_{pa(Y)}$), then $Y$ and $E$ are no longer $d$-separated by $\mathbf{X}_{pa(Y)}$ and Assumption 2 is unsatisfiable. Then our method will be unable to find an invariant mechanism.

### B.2 Using Causal Effects for Prediction

Our estimator might use predictor variables beyond $\mathbf{X}_{pa(Y)}$ as well, e.g. children of $Y$ or parents of children, provided their relationships to $Y$ do not depend on the environment. The case of children is especially interesting: Suppose $X_j$ is a noisy measurement of $Y$, described by the causal mechanism $P(X_j \mid Y)$. As long as the measurement device works identically in all environments, including $X_j$ as a predictor of $Y$ is desirable, despite it being a child.

### B.3 Examples

Domain generalization is in general impossible without strong assumptions (in contrast to classical supervised learning). In our view, the interesting question is "Which strong assumptions are the most useful in a given setting?". For instance, [14] use Assumption 2 to identify causes for birth rates in different countries. If all variables mediating the influence of continent/country (environment variable) on birth rates (target variable) are included in the model (e.g. GDP, Education), this assumption is reasonable. The same may hold for other epidemiological investigations as well. [33] suppose Assumption 2 in the field of finance.

Another reasonable example are data augmentations in computer vision. Deliberate image rotations, shifts and distortions can be considered as environment interventions that preserve the relation between semantic image features and object classes (see e.g. [25]), i.e. verify assumption 2. In general, assumption 2 may be justified when one studies a fundamental mechanism that can reasonably be assumed to remain invariant across environments, but is obscured by unstable relationships between observable variables.

### B.4 Robustness Example

To illustrate the impact of causality on robustness, consider the following example: Suppose we would like to estimate the gas consumption of a car. In a sufficiently narrow setting, the total amount of money spent on gas might be a simple and accurate predictor. However, gas prices vary dramatically between countries and over time, so statistical models relying on it will not be robust, even if they fit the training data very well. Gas costs are an *effect* of gas consumption, and this relationship is unstable due to external influences. In contrast, predictions on the basis of the *causes* of gas consumption (e.g. car model, local speed limits and geography, owner's driving habits) tend to be much more robust, because these causal relations are intrinsic to the system and not subjected to external influences. Note that there is a trade-off here: Including gas costs in the model will improve estimation accuracy when gas prices remain sufficiently stable, but will impair results otherwise. By considering the same phenomenon in several environments simultaneously, we hope to gain enough information to adjust this trade-off properly.

In the gas example, countries can be considered as environments that "intervene" on the relation between consumed gas and money spent, e.g. by applying different tax policies. In contrast, interventions changing the impact of motor properties or geography on gas consumption are much less plausible - powerful motors and steep roads will always lead to higher consumption. From a causal standpoint, finding robust models is therefore a causal discovery task [24].

## C Normalizing Flows

Normalizing flows are a specific type of neural network architecture which are by construction invertible and have a tractable Jacobian. They are used for density estimation and sampling of a target density (for an overview see [28]). This in turn allows optimizing information theoretic objectives in a convenient and mathematically sound way.

Similarly as in the paper, we denote with $\mathcal{H}$ the set of feature extractors $h\colon \mathbb{R}^D \to \mathbb{R}^M$ where $M$ is chosen a priori. The set of all one-dimensional (conditional) normalizing flows is denoted by $\mathcal{T}$. Together with a reference distribution $p_{ref}$, a normalizing flow $T$ defines a new distribution $\nu_T = (T(\cdot; h(\mathbf{x})))^{-1}_{\#} p_{ref}$ which is called the push-forward of the reference distribution $p_{ref}$ [23]. By drawing samples from $p_{ref}$ and applying $T$ on these samples we obtain samples from this new distribution. The density of this so-obtained distribution $p_{\nu_T}$ can be derived from the change of variables formula:

$$p_{\nu_T}(y \mid h(\mathbf{x})) = p_{ref}(T(y; h(\mathbf{x})))|\nabla_y T(y; h(\mathbf{x}))| \tag{12}$$

The KL-divergence between the target distribution $p_{Y|h(\mathbf{X})}$ and the flow-based model $p_{\nu_T}$ can be written as follows:

$$\begin{aligned}
&\mathbb{E}_{h(\mathbf{X})}[D_{\mathrm{KL}}(p_{Y|h(\mathbf{X})} \| p_{\nu_T})] \\
=&\mathbb{E}_{h(\mathbf{X})}\left[\mathbb{E}_{Y|h(\mathbf{X})}\left[\log\left(\frac{p_{Y|h(\mathbf{X})}}{p_{\nu_T}}\right)\right]\right] \\
=&-H(Y \mid h(\mathbf{X})) - \mathbb{E}_{h(\mathbf{X}),Y}[\log p_{\nu_T}(Y \mid h(\mathbf{X}))] \\
=&-H(Y \mid h(\mathbf{X})) + \mathbb{E}_{h(\mathbf{X}),Y}[-\log p_{ref}(T(y; h(\mathbf{x})) \\
&-\log|\nabla_y T(y; h(\mathbf{x}))|]
\end{aligned} \tag{13}$$

The last two terms in Eq. 13 correspond to the negative log-likelihood (NLL) for conditional flows with distribution $p_{ref}$ in latent space. If the reference distribution is assumed to be standard normal, the NLL is given as in Sect. 3.

We restate Lemma 1 with a more general notation. Note that the argmax or argmin is a set.

**Lemma 1.** *Let* $\mathbf{X}, Y$ *be random variables. We furthermore assume that for each* $h \in \mathcal{H}$ *there exists one* $T \in \mathcal{T}$ *with* $\mathbb{E}_{h(\mathbf{X})}[D_{KL}(p_{Y|h(\mathbf{X})} \| p_{\nu_T})] = 0$. *Then, the following two statements are true*

*(a) Let*

$$h^\star, T^\star = \arg\min_{h \in \mathcal{H}, T \in \mathcal{T}} -\mathbb{E}_{h(\mathbf{X}),Y}[\log p_{\nu_T}(Y \mid h(\mathbf{X}))]$$

*then it holds* $h^\star = g^\star$ *where* $g^\star = \arg\max_{g \in \mathcal{H}} I(g(\mathbf{X}); Y)$

*(b) Let*

$$T^\star = \arg\min_{T\in\mathcal{T}} \mathbb{E}_{h(\mathbf{X})}[D_{KL}(p_{Y|h(\mathbf{X})}\|p_{\nu_T})]$$

*then it holds* $h(\mathbf{X}) \perp T^\star(Y; h(\mathbf{X}))$

*Proof.* (a) From Eq. 13, we obtain $-\mathbb{E}_{h(\mathbf{X}),Y}[\log p_{\nu_T}(Y \mid h(\mathbf{X}))] \geq H(Y \mid h(\mathbf{X}))$ for all $h \in \mathcal{H}, T \in \mathcal{T}$. We furthermore have $\min_{T\in\mathcal{T}} -\mathbb{E}_{h(\mathbf{X}),Y}[\log p_{\nu_T}(Y \mid h(\mathbf{X}))] = H(Y \mid h(\mathbf{X}))$ due to our assumptions on $\mathcal{T}$.

Therefore, $\min_{h\in\mathcal{H},T\in\mathcal{T}} -\mathbb{E}_{h(\mathbf{X}),Y}[\log p_{\nu_T}(Y \mid h(\mathbf{X}))] = \min_{h\in\mathcal{H}} H(Y \mid h(\mathbf{X}))$. Since we have $I(Y; h(\mathbf{X})) = H(Y) - H(Y \mid h(\mathbf{X}))$ and only the second term depends on $h$, statement (a) holds true.

(b) For convenience, we denote $T(Y; h(\mathbf{X})) = R$ and $h(\mathbf{X}) = Z$. We have $\mathbb{E}_Z[D_{\mathrm{KL}}(p_{Y|Z}\|p_{\nu_{T^\star}})] = 0$ and therefore $p_{Y|Z}(y \mid z) = p_{ref}(T(y;z))|\nabla_y T^{-1}(y;z)|$.

Then it holds

$$\begin{aligned} p_{R|Z}(r \mid z) &= p_{Y|Z}(T^{-1}(r;z)|z) \cdot |\nabla_y T^{-1}(r;z)| \\ &= p_{ref}(T(T^{-1}(r;z);z)) \cdot |\nabla_y T(y;z)| \\ &\quad \cdot |\nabla_y T^{-1}(r;z)| \\ &= p_{ref}(r) \cdot 1 \end{aligned}$$

Since the density $p_{ref}$ is independent of $Z$, we obtain $R \perp Z$ which concludes the proof of (b)

Statement (a) describes an optimization problem that allows to find features which share maximal information with the target variable $Y$. Due to statement (b) it is possible to draw samples from the conditional distribution $P(Y \mid h(\mathbf{X}))$ via the reference distribution.

Let $\mathcal{H}_\perp$ the set of features which satisfy the invariance property, i.e. $Y \perp E \mid h(\mathbf{X})$ for all $h \in \mathcal{H}_\perp$. In the following, we sketch why

$\arg\min_{h\in\mathcal{H}_\perp} \max_{e\in\mathcal{E}} \min_{T\in\mathcal{T}} \mathcal{L}^e_{\mathrm{NLL}}(T;h) = \arg\max_{h\in\mathcal{H}_\perp} \min_{e\in\mathcal{E}} I(Y^e; h(\mathbf{X}^e))$ follows from Lemma 1.

Let $h \in \mathcal{H}_\perp$. Then, it is easily seen that there exists a $T^\star \in \mathcal{T}$ with (1) $\mathcal{L}_{\mathrm{NLL}}(T^\star; h) = \min_{T\in\mathcal{T}} \mathcal{L}_{\mathrm{NLL}}(T, h)$ and (2) $\mathcal{L}^e_{\mathrm{NLL}}(T^\star, h) = \min_{T\in\mathcal{T}} \mathcal{L}^e_{\mathrm{NLL}}(T, h)$ for all $e \in \mathcal{E}$ since the conditional densities $p(y \mid h(\mathbf{X}))$ are invariant across all environments. Hence we have $H(Y^e \mid h(\mathbf{X}^e)) = \mathcal{L}^e_{\mathrm{NLL}}(T^\star; h)$ for all $e \in \mathcal{E}$. Therefore, $\arg\min_{h\in\mathcal{H}_\perp} \max_{e\in\mathcal{E}} \min_{T\in\mathcal{T}} \mathcal{L}^e_{\mathrm{NLL}}(T; h) = \arg\max_{h\in\mathcal{H}_\perp} \min_{e\in\mathcal{E}} I(Y^e; h(\mathbf{X}^e))$ due to $I(Y^e; h(\mathbf{X}^e)) = H(Y^e) - H(Y^e \mid h(\mathbf{X}^e))$.

## C.1 Normalizing Flows and Additive Noise Models

In our case, we represent the conditional distribution $P(Y \mid h(\mathbf{X}))$ using a *conditional* normalizing flow (see e.g. [1]). In our work, we seek a mapping $R = T(Y; h(\mathbf{X}))$ that is diffeomorphic in $Y$ such that $R \sim \mathcal{N}(0,1) \perp h(\mathbf{X})$ when

$Y \sim P(Y \mid h(\mathbf{X}))$. This is a generalization of the well-studied additive Gaussian noise model $R = Y - f(h(\mathbf{X}))$, see Appendix E. The inverse $Y = F(R; h(\mathbf{X}))$ takes the role of a structural equation for the mechanism $p(Y \mid h(\mathbf{X}))$ with $R$ being the corresponding noise variable.[1]

## D HSIC and Wasserstein Loss

The Hilbert-Schmidt Independence Criterion (HSIC) is a kernel based measure for independence which is in expectation 0 if and only if the compared random variables are independent [12]. An empirical estimate of HSIC$(A, B)$ for two random variables $A, B$ is given by

$$\widehat{\text{HSIC}}(\{a_j\}_{j=1}^n, \{b_j\}_{j=1}^n) = \frac{1}{(n-1)^2}\text{tr}(KHK'H) \tag{14}$$

where tr is the trace operator. $K_{ij} = k(a^i, a^j)$ and $K'_{ij} = k'(b^i, b^j)$ are kernel matrices for given kernels $k$ and $k'$. The matrix $H$ is a centering matrix $H_{i,j} = \delta_{i,j} - 1/n$.

The one dimensional Wasserstein loss compares the similarity of two distributions [18]. This loss has expectation 0 if both distributions are equal. An empirical estimate of the one dimensional Wasserstein loss for two random variables $A, B$ is given by

$$\mathcal{L}_W = \|\text{sort}(\{a_j\}_{j=1}^n) - \text{sort}(\{b_j\}_{j=1}^n)\|_2$$

Here, the two batches are sorted in ascending order and then compared in the L2-Norm. We assume that both batches have the same size.

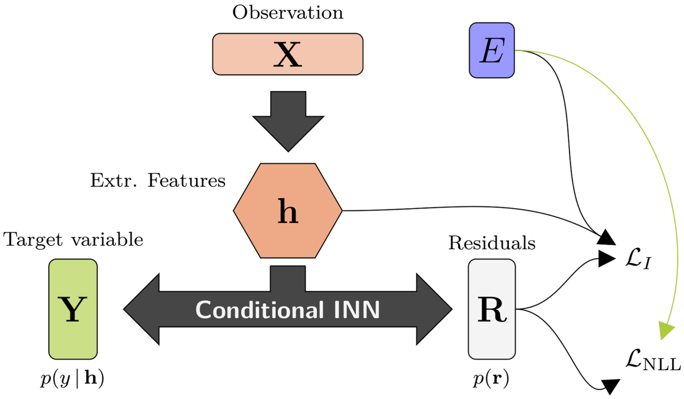


**Fig. 4.** Illustration of Architecture of Conditional Invertible Neural Network (Conditional INN) which implements Eq. 7. $h$ is a feature extractor implemented as feed forward neural network. $\mathcal{L}_I$ is the invariance loss that measures the dependence between residuals $R$ and $(E, h(\mathbf{X}))$ and $\mathcal{L}_{\text{NLL}}$ is the negative log-likelihood as in Eq. 5.

[1] $F$ is the concatenation of the normal CDF with theinverse CDF of $P(Y \mid h(\mathbf{X}))$, see [32].

# E Additive Noise Models and Robust Prediction

Let $f_\theta$ be a regression function. Solving for the noise term gives $R = Y - f_\theta(\mathbf{X})$ which corresponds to a diffeomorphism in $Y$, namely $T_\theta(Y; X) = Y - f_\theta(\mathbf{X})$. If we make two simplified assumptions: (i) the noise is Gaussian with zero mean and (ii) $R \perp f_\theta(\mathbf{X})$, then we obtain

$$\begin{aligned} I(Y; f_\theta(\mathbf{X})) &= H(Y) - H(Y \mid f_\theta(\mathbf{X})) = H(Y) - H(R \mid f_\theta(\mathbf{X})) \\ &\overset{(ii)}{=} H(Y) - H(R) \overset{(i)}{=} H(Y) - 1/2 \log(2\pi e \sigma^2) \end{aligned}$$

where $\sigma^2 = \mathbb{E}[(Y - f_\theta(\mathbf{X}))^2]$. In this case maximizing the mutual information $I(Y; f_\theta(\mathbf{X}))$ amounts to minimizing $\mathbb{E}[(Y - f_\theta(\mathbf{X}))^2]$ w.r.t. $\theta$, i.e. the standard L2-loss for regression problems. From this, we obtain an approximation of Eq. 7 via

$$\arg\min_\theta \Big( \max_{e \in \mathcal{E}_{\text{seen}}} \Big\{ \mathbb{E}\big[(Y^e - f_\theta(\mathbf{X}^e))^2\big] \Big\} + \lambda_I \mathcal{L}_I(P_R, P_{f_\theta(\mathbf{X}),E}) \Big) \tag{15}$$

where $R^e = Y - f_\theta(\mathbf{X}^e)$ and $\lambda_I > 0$. Under the conditions stated above, the objective achieves the mentioned trade-off between information and invariance.

Alternatively we can view the problem as to find features $h_\phi \colon \mathbb{R}^D \to \mathbb{R}^m$ such that $I(h_\phi(\mathbf{X}), Y)$ gets maximized under the assumption that there exists a model $f_\theta(h_\phi(\mathbf{X})) + R = Y$ where $R$ is independent of $h_\phi(\mathbf{X})$ and $R$ is gaussian. In this case we obtain similarly as above the learning objective

$$\arg\min_{\theta,\phi} \Big( \max_{e \in \mathcal{E}_{\text{seen}}} \Big\{ \mathbb{E}\big[(Y^e - f_\theta(h_\phi(\mathbf{X}^e)))^2\big] \Big\} + \lambda_I \mathcal{L}_I(P_R, P_{h_\phi(\mathbf{X}),E}) \Big) \tag{16}$$

# F Algorithm

In order to optimize the DG problem in Eq. 4, we optimize a normalizing flow $T_\theta$ and a feed forward neural network $h_\phi$ as described in Algorithm 1. There is an inherent trade-off between robustness and goodness-of-fit. The hyperparameter $\lambda_I$ describes this trade-off and is chosen a priori.

If we choose a gating mechanisms $h_\phi$ as feature extractor similar to [17], then a complexity loss is added to the loss in the gradient update step. The architecture is illustrated in Fig. 4. Figure 5 shows the architecture with gating function.

In case we assume that the underlying mechanisms elaborates the noise in an additive manner, we could replace the normalizing flow $T_\theta$ with a feed forward neural network $f_\theta$ and execute Algorithm 2.

If we choose a gating mechanism, minor adjustments have to be made to Algorithm 2 such that we optimize Eq. 9. The classification case can be obtained similarly as described in Sect. 4.

```
Data: Samples from $P_{\mathbf{X}^e,Y^e}$ in different environments $e \in \mathcal{E}_{\text{seen}}$.
Initialize: Parameters $\theta, \phi$;
for number of training iterations do
    for $e \in \mathcal{E}_{seen}$ do
        Sample minibatch $\{(y_1^e, \mathbf{x}_1^e), \dots, (y_m^e, \mathbf{x}_m^e)\}$ from $P_{Y,\mathbf{X}|E=e}$ for $e \in \mathcal{E}_{\text{seen}}$;;
        Compute $r_j^e = T_\theta(y_j^e; h_\phi(\mathbf{x}_j^e))$;;
    end
    Update $\theta, \phi$ by descending alongside the stochastic gradient
        $\nabla_{\theta,\phi}\Big( \max_{e \in \mathcal{E}_{\text{seen}}} \Big\{ \sum_{i=1}^{m} \big[\tfrac{1}{2}\|T_\theta(y_i^e; h_\phi(\mathbf{x}_i^e))\|^2$
            $- \log \nabla_y T_\theta(y_i^e; h_\phi(\mathbf{x}_i^e))\big] \Big\}$
            $+ \lambda_I \mathcal{L}_I(\{r_j^e\}_{j,e}, \{h_\phi(\mathbf{x}_j^e), e\}_{j,e})\Big)$;
end
Result: In case of convergence, we obtain $T_{\theta^\star}, h_{\phi^\star}$ with
    $\theta^\star, \phi^\star =$
    $\arg\min_{\theta,\phi} \Big( \max_{e \in \mathcal{E}_{\text{seen}}} \Big\{ \mathbb{E}_{\mathbf{X}^e,Y^e} \big[\tfrac{1}{2}\|T_\theta(Y^e; h_\phi(\mathbf{X}^e))\|^2$
        $- \log \nabla_y T_\theta(Y^e; h_\phi(\mathbf{X}^e))\big] \Big\}$
        $+ \lambda_I \mathcal{L}_I(P_R, P_{h_\phi(\mathbf{X}),E})\Big)$
```

**Algorithm 1:** DG training with normalizing flows

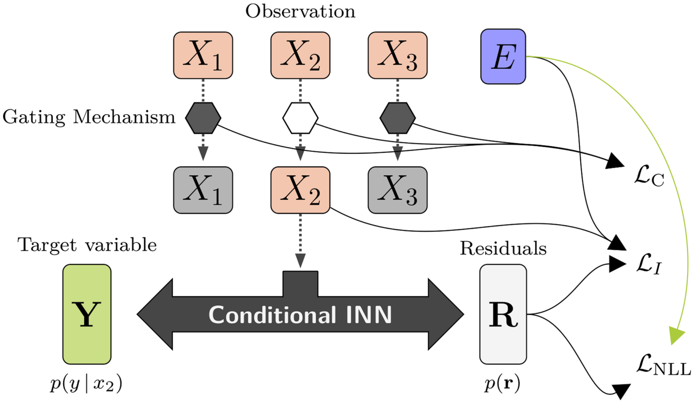


**Fig. 5.** Illustration of Architecture of Conditional Invertible Neural Network (Conditional INN) model which implements Eq. 7 where the feature extractor $h$ is a gating mechanism. Architecture is depicted for three input variables. $\mathcal{L}_I$ is the invariance loss that measures the dependence between residuals $R$ and $(E, h(\mathbf{X}))$, $\mathcal{L}_\text{NLL}$ is the negative log-likelihood as in Eq. 5 and $\mathcal{L}_\text{C}$ is the complexity loss as described in Sect. 4.3.

```
Data: Samples from $P_{\mathbf{X}^e,Y^e}$ in different environments $e \in \mathcal{E}_{\text{seen}}$.
Initialize: Parameters $\theta, \phi$;
for number of training iterations do
    for $e \in \mathcal{E}_{seen}$ do
        Sample minibatch $\{(y_1^e, \mathbf{x}_1^e), \dots, (y_m^e, \mathbf{x}_m^e)\}$ from $P_{Y,\mathbf{X}|E=e}$ for $e \in \mathcal{E}_{\text{seen}}$;;
        Compute $r_j^e = y_j^e - f_\theta(\mathbf{x}_j^e)$;;
    end
    Update $\theta$ by descending alongside the stochastic gradient
        $\nabla_\theta \Big( \max_{e \in \mathcal{E}_{\text{seen}}} \Big\{ \sum_{i=1}^{m} |r_j^e|^2 \Big\} + \lambda_I \mathcal{L}_I(\{r_j^e\}_{j,e}, \{f_\theta(\mathbf{x}_j^e), e\}_{j,e}) \Big);$
end
Result: In case of convergence, we obtain $f_{\theta^\star}$ with
    $\theta^\star = \arg\min_\theta \Big( \max_{e \in \mathcal{E}_{\text{seen}}} \Big\{ \mathbb{E}_{\mathbf{X}^e,Y^e} \big[ |Y^e - f_\theta(\mathbf{X}^e)|^2 \big] \Big\} + \lambda_I \mathcal{L}_I(P_R, P_{f_\theta(\mathbf{X}),E}) \Big)$
```

**Algorithm 2:** DG training under the assumption of additive noise

## G Identifiability Result

Under certain conditions on the environment and the underlying causal graph, the direct causes of $Y$ become identifiable:

**Proposition 1.** *We assume that the underlying causal graph $G$ is faithful with respect to $P_{\widetilde{\mathbf{X}},E}$. We further assume that every child of $Y$ in $G$ is also a child of $E$ in $G$. A variable selection $h(\mathbf{X}) = \mathbf{X}_S$ corresponds to the direct causes if the following conditions are met: (i) $T(Y; h(\mathbf{X})) \perp E, h(\mathbf{X})$ are satisfied for a diffeomorphism $T(\cdot; h(\mathbf{X}))$, (ii) $h(\mathbf{X})$ is maximally informative about $Y$ and (iii) $h(\mathbf{X})$ contains only variables from the Markov blanket of $Y$.*

*Proof.* Let $S(\mathcal{E}_{\text{seen}})$ denote a subset of $\mathbf{X}$ which corresponds to the variable selection due to $h$. Without loss of generality, we assume $S(\mathcal{E}_{\text{seen}}) \subset \mathbf{M}$ where $\mathbf{M}$ is the Markov Blanket. This assumption is reasonable since we have $Y \perp \mathbf{X} \setminus \mathbf{M} \mid \mathbf{M}$ in the asymptotic limit.

Since $pa(Y)$ cannot contain colliders between $Y$ and $E$, we obtain that $Y \perp E \mid S(\mathcal{E}_{\text{seen}})$ implies $Y \perp E \mid (S(\mathcal{E}_{\text{seen}}) \cup pa(Y))$. This means using $pa(Y)$ as predictors does not harm the constraint in the optimization problem. Due to faithfulness and since the parents of $Y$ are directly connected to $Y$, we obtain that $pa(Y) \subset S(\mathcal{E}_{\text{seen}})$.

For each subset $\mathbf{X}_S \subset \mathbf{X}$ for which there exists an $X_i \in \mathbf{X}_S \cap \mathbf{X}_{ch(Y)}$, we have $\mathbf{X}_S \not\perp Y \mid E$. This follows from the fact that $X_i$ is a collider, in particular $E \rightarrow X_i \leftarrow Y$. Conditioning on $X_i$ leads to the result that $Y$ and $E$ are not

$d$-separated anymore. Hence, we obtain $Y \not\perp \mathbf{X}_S \mid E$ due to the faithfulness assumption. Hence, for each $\mathbf{X}_S$ with $Y \perp E \mid \mathbf{X}_S$ we have $\mathbf{X}_S \cap \mathbf{X}_{ch(Y)} = \emptyset$ and therefore $\mathbf{X}_{ch(Y)} \cap S(\mathcal{E}_{\text{seen}}) = \emptyset$.

Since $\mathbf{X}_{pa(Y)} \subset S(\mathcal{E}_{\text{seen}})$, we obtain that $Y \perp \mathbf{X}_{pa(ch(Y))} \mid \mathbf{X}_{pa(Y)}$ and therefore the parents of $ch(Y)$ are not in $S(\mathcal{E}_{\text{seen}})$ except when they are parents of $Y$.

Therefore, we obtain that $S(\mathcal{E}_{\text{seen}}) = \mathbf{X}_{pa(Y)}$.

One might argue that the conditions are very strict in order to obtain the true direct causes. But the conditions set in Proposition 1 are necessary if we do not impose additional constraints on the true underlying causal mechanisms, e.g. linearity as done by [30]. For instance if $E \to X_1 \to Y \to X_2$, a model including $X_1$ and $X_2$ as predictor might be a better predictor than the one using only $X_1$. From the Causal Markov Condition we obtain $E \perp Y \mid X_1, X_2$ which results in $X_1, X_2 \in S(\mathcal{E}_{\text{seen}})$. Under certain conditions however, the relation $Y \to X_2$ might be invariant across $\mathcal{E}$. This is for instance the case when $X_2$ is a measurement of $Y$. In this cases it might be useful to use $X_2$ for a good prediction.

### G.1 Gating Architecture

We employ the same gating architecture as in [17] which was first proposed in [21] as a Bernoulli reparameterization trick. They use this reparameterization trick in their original work in order to train neural networks with L0-Regularization in a gradient based manner. [17] apply the L0-Regularization on the input to learn a gating mechanism. Similarly we use the L0-Regularization to learn a gating mechanism.

The gating architecture $h_\phi$ is parameterized via $\phi = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_D)$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_D)$. Let $\gamma < 0$ and $\zeta > 0$ be fixed. Then we map $\boldsymbol{u} \sim \mathcal{U}[0,1]^D$ via $\boldsymbol{s}(\boldsymbol{u}) = \text{Sigmoid}((\log \boldsymbol{u} - \log(1-\boldsymbol{u}) + \boldsymbol{\alpha})/\boldsymbol{\beta})$, to $\boldsymbol{z} = \min(1, \max(0, \boldsymbol{s}(\boldsymbol{u})(\zeta - \gamma) + \gamma))$. This is how we sample the gates for each batch during training. The gates are then multiplied element-wise with the input $\boldsymbol{z} \odot \mathbf{X}$. In principle we could sample many $u \sim \mathcal{U}[0,1]$, but we observe that one sample of $u \sim \mathcal{U}[0,1]$ per batch suffices for our examples. At test time we use the following estimator for the gates:

$$\hat{\boldsymbol{z}} = \min(1, \max(0, \text{Sigmoid}(\boldsymbol{\alpha})(\zeta - \gamma) + \gamma))$$

Similarly as during training time, we multiply $\hat{\boldsymbol{z}}$ with the input. After sufficient training $\hat{\boldsymbol{z}}$ is a hard 0-1 mask. The complexity loss is defined via

$$\mathcal{L}(h_\theta) = \sum_{j=1}^{D} \text{Sigmoid}\Big(\alpha_j - \beta_j \log \frac{-\gamma}{\zeta}\Big). \tag{17}$$

For a detailed derivation of the reparameterization and complexity loss, see [21].

# H Experimental Setting for Synthetic Dataset

## H.1 Data Generation

In Sect. 5 we described how we choose different Structural Causal Models (SCM). In the following we describe details of this process.

We simulate the datasets in a way that the conditions in Proposition 1 are met. We choose different variables in the graph shown in Fig. 2 as target variable. Hence, we consider different "topological" scenarios. We assume the data is generated by some underlying SCM. We define the structural assignments in the SCM as follows

$$\text{(a)} \quad f_i^{(1)}(\mathbf{X}_{pa(i)}, N_i) = \sum_{j \in pa(i)} a_j X_j + N_i \quad \text{[Linear]}$$

$$\text{(b)} \quad f_i^{(2)}(\mathbf{X}_{pa(i)}, N_i) = \sum_{j \in pa(i)} a_j X_j - \tanh(a_j X_j) + N_i$$

[Tanhshrink]

$$\text{(c)} \quad f_i^{(3)}(\mathbf{X}_{pa(i)}, N_i) = \sum_{j \in pa(i)} \log(1 + \exp(a_j X_j)) + N_i$$

[Softplus]

$$\text{(d)} \quad f_i^{(4)}(\mathbf{X}_{pa(i)}, N_i) = \sum_{j \in pa(i)} \max\{0, a_j X_j)\} + N_i$$

[ReLU]

$$\text{(e)} \quad f_i^{(5)}(\mathbf{X}_{pa(i)}, N_i) = \Big( \sum_{j \in pa(i)} a_j X_j \Big) \cdot (1 + \frac{1}{4} N_i) + N_i$$

[Mult. Noise]

with $N_i \sim \mathcal{N}(0, c_i^2)$ where $c_i \sim \mathcal{U}[0.8, 1.2]$, $i \in \{0, \ldots, 5\}$ and $a_i \in \{-1, 1\}$ according to Fig. 6. Note that the mechanisms in (b), (c) and (d) are non-linear with additive noise and (e) elaborates the noise in a non-linear manner.

We consider hard- and soft-interventions on the assignments $f_i$. We either intervene on all variables except the target variable at once *or* on all parents and children of the target variable (Intervention Location). We consider three types of interventions:

- *Hard-Intervention* on $X_i$: Force $X_i \sim e_1 + e_2 \mathcal{N}(0, 1)$ where we sample for each environment $e_2 \sim \mathcal{U}([1.5, 2.5])$ and $e_1 \sim \mathcal{U}([0.5, 1.5] \cup [-1.5, -0.5])$
- *Soft-Intervention* I on $X_i$: Add $e_1 + e_2 \mathcal{N}(0, 1)$ to $X_i$ where we sample for each environment $e_2 \sim \mathcal{U}([1.5, 2.5])$ and $e_1 \sim \mathcal{U}([0.5, 1.5] \cup [-1.5, -0.5])$
- *Soft-Intervention* II on $X_i$: Set the noise distribution $N_i$ to $\mathcal{N}(0, 2^2)$ for $E = 2$ and to $\mathcal{N}(0, 0.2^2)$ for $E = 3$

Per run, we consider one environment without intervention ($E = 1$) and two environments with either both soft- or hard-interventions ($E = 2, 3$). We also create a fourth environment to measure a models' ability for out-of-distribution generalization:

- *Hard-Intervention*: Force $X_i \sim e + \mathcal{N}(0, 4^2)$ where $e = e_1 \pm 1$ with $e_1$ from environment $E = 1$. The sign $\{+, -\}$ is chosen once for each $i$ with equal probability.
- *Soft-Intervention* I: Add $e + \mathcal{N}(0, 4^2)$ to $X_i$ where $e = e_1 \pm 1$ with $e_1$ from environment $E = 1$. The sign $\{+, -\}$ is chosen once for each $i$ with equal probability as for the *do-intervention* case.
- *Soft-Intervention* II: Half of the samples have noise $N_i$ distributed due to $\mathcal{N}(0, 1.2^2)$ and the other half of the samples have noise distributed as $\mathcal{N}(0, 3^2)$

We randomly sample causal graphs as described above. Per environment, we consider 1024 samples.

### H.2 Training Details

All used feed forward neural networks have two internal layers of size 256. For the normalizing flows we use a 2 layer *MTA-Flow* described in Appendix H.3 with K=32. As optimizer we use Adam with a learning rate of $10^{-3}$ and a L2-Regularizer weighted by $10^{-5}$ for all models. Each model is trained with a batch size of 256. We train each model for 1000 epochs and decay the learning rate every 400 epochs by 0.5. For each model we use $\lambda_I = 256$ and the HSIC $\mathcal{L}_I$ employs a Gaussian kernel with $\sigma = 1$. The gating architecture was trained without the complexity loss for 200 epochs and then with complexity loss weighted by 5. For the Flow model without gating architecture we use a feed forward neural network $h_\phi$ with two internal layers of size 256 mapping to an one dimensional vector. In total, we evaluated our models on 1365 created datasets as described in H.1.

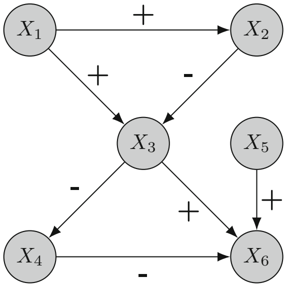


**Fig. 6.** The signs of the coefficients $a_j$ for the mechanisms of the different SCMs

Once the normalizing flow $T$ is learned, we predict $y$ given features $h(\mathbf{x})$ using 512 normally distributed samples $u_i$ which are mapped to samples from $p(y|h(\mathbf{x}))$ by the trained normalizing flow $T(u_i; h(\mathbf{x}))$. As prediction we use the mean of these samples.

### H.3 One-Dimensional Normalizing Flow

We use as one-dimension normalizing flow the *More-Than-Affine-Flow (MTA-Flow)*, which was developed by us. An overview of different architectures for one-dimensional normalizing flows can be found in [28]. For each layer of the flow, a conditioner network C maps the conditional data $h(\mathbf{X})$ to a set of parameters $a, b \in \mathbb{R}$ and $\mathbf{w}, \mathbf{v}, \mathbf{r} \in \mathbb{R}^K$ for a chosen $K \in \mathbb{N}$. It builds the transformer $\tau$ for each layer as

$$\begin{aligned} z &= \tau(y \mid h(\mathbf{X})) \\ &:= a\left(y + \frac{1}{N(\mathbf{w}, \mathbf{v})} \sum_{i=1}^{K} w_i f(v_i y + r_i)\right) + b, \end{aligned} \tag{18}$$

where $f$ is any almost everywhere smooth function with a derivative bounded by 1. In this work we used a gaussian function with normalized derivative for $f$. The division by

$$N(\mathbf{w}, \mathbf{v}) := \varepsilon^{-1} \left(\sum_{i=1}^{K} |w_i v_i| + \delta\right), \tag{19}$$

with numeric stabilizers $\varepsilon < 1$ and $\delta > 0$, assures the strict monotonicity of $\tau$ and thus its invertibility $\forall x \in \mathbb{R}$. We also used a slightly different version of the *MTA-Flow* which uses the ELU activation function and – because of its monotonicity – can use a relaxed normalizing expression $N(\mathbf{w}, \mathbf{v})$.

## H.4 PC-Variant

Since we are interested in the direct causes of $Y$, the widely applied PC-Algorithm gives not the complete answer to the query for the parents of $Y$. This is due to the fact that it is not able to orient all edges. To compare the PC-Algorithm we include the environment as system-intern variable and use a conservative assignment scheme where non-oriented edges are thrown away. This assignment scheme corresponds to the conservative nature of the ICP.

For further interest going beyond this work, we consider diverse variants of the PC-Algorithm. We consider two orientation schemes: A *conservative* one, where non-oriented edges are thrown away and a *non-conservative* one where non-oriented edges from a node $X_i$ to $Y$ are considered parents of $Y$.

We furthermore consider three scenarios: (1) the samples across all environments are pooled, (2) only the observational data (from the first environment) is given, and (3) the environment variable is considered as system-intern variable and is seen by the PC-Algorithm (similar as in [26]). Results are shown in Fig. 7. In order to obtain these results, we sampled 1500 graphs as described above and applied on each of these datasets a PC-Variant. Best accuracies are achieved if we consider the environment variable as system-intern variable and use the non-conservative orientation scheme (EnvIn).

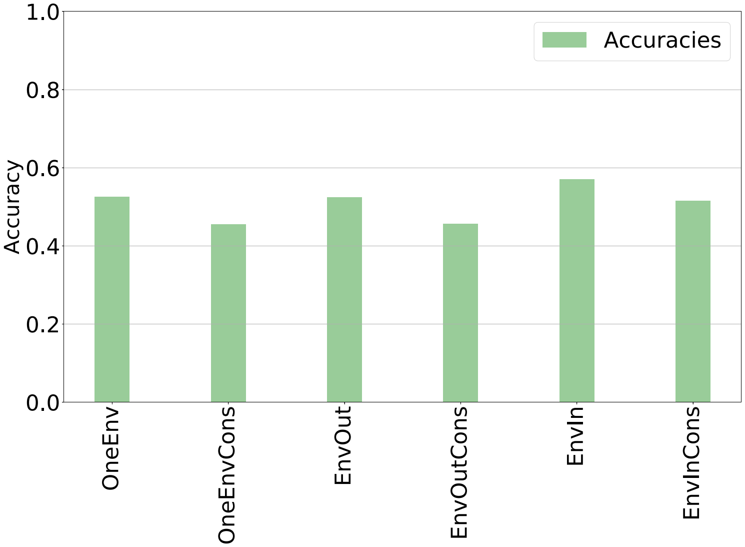


**Fig. 7.** Detection accuracies of direct causes for different variants of the PC-Algorithm. EnvOut means we pool over all environments and EnvIn means the environment is treated as system intern variable $E$. The suffix Cons means we us the conservative assignment scheme. OneEnv means we only consider the observational environment for inference.

### H.5 Variable Selection

We consider the task of finding the direct causes of a target variable $Y$. Our models based on the gating mechanism perform a variable selection and are therefore compared to the PC-Algorithm and ICP. In the following we show the accuracies of this variable selection according to different scenarios.

Figure 8 shows the accuracies of ICP, the PC-Algorithm and our models pooled over all scenarios. Our models perform comparably well and better than the baseline in the causal discovery task.

In the following we show results due to different mechanisms, target variables, intervention types and intervention locations. Figure 9a shows the accuracies of all models across different target variables. Parentless target variables, i.e. $Y = X_4$ or $Y = X_0$ are easy to solve for ICP due to its conservative nature. All our models solve the parentless case quite well. Performance of the PC-variant depends strongly on the position of the target variable in the SCM indicating that its conservative assignment scheme has a strong influence on its performance. As expected, the PC-variant deals well with with $Y = X_6$ which is a childless collider. The causal discovery task seems to be particularly hard for variable $Y = X_6$ for all other models. This is the variable which has the most parents.

The type of intervention and its location seem to play a minor role as shown in Fig. 9a and Fig. 9a.

Figure 9b shows that ICP performs well if the underlying causal model is linear, but degrades if the mechanism become non-linear. The PC-Algorithm

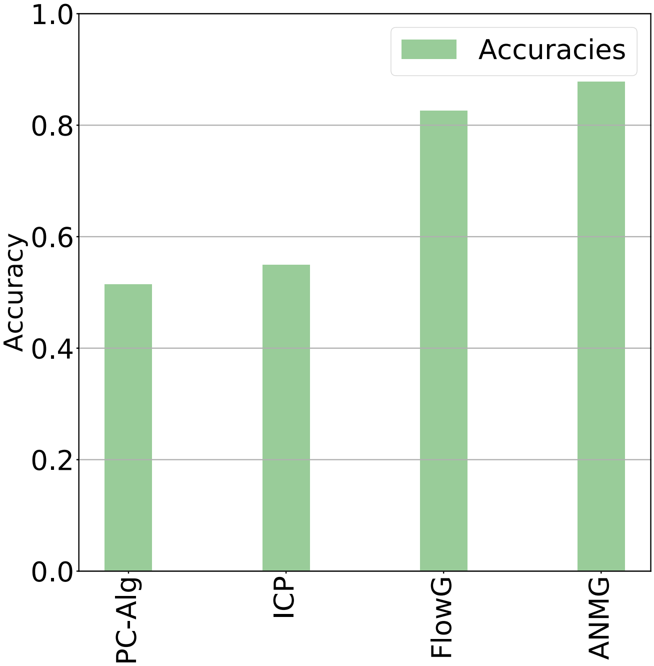
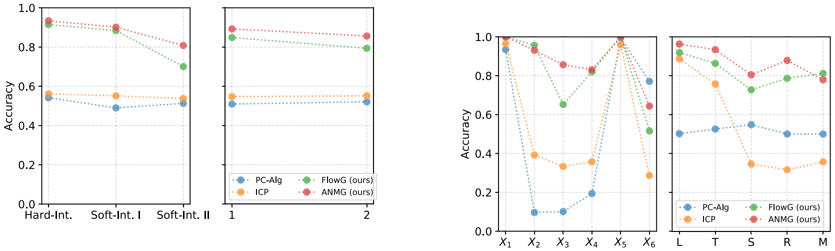


**Fig. 8.** Accuracies for different models across all scenarios. FlowG and ANMG are our models.



(a) Accuracies of different models for different intervention types and locations. 1 stands for intervention on all variables except $Y$ and 2 stands for interventions on parents and children only.

(b) Accuracies of different models according to target variables and mechanisms of the underlying SCM.

**Fig. 9.** Comparison of models across different scenarios in the causal discovery task

performs under all mechanisms comparably, but not well. ANMG performs quite well in all cases and even slightly better than FlowG in the cases of additive noise. However in the case of non-additive noise FlowG performs quite well whereas ANMG perform slightly worse – arguably because their requirements (additive noise) on the underlying mechanisms are not met.

## H.6 Transfer Study

In the following we show the performance of different models on the training set, a test set of the same distribution and a set drawn from an unseen environment for different scenarios. As in Sect. 5, we use the L2-Loss on samples of an unseen

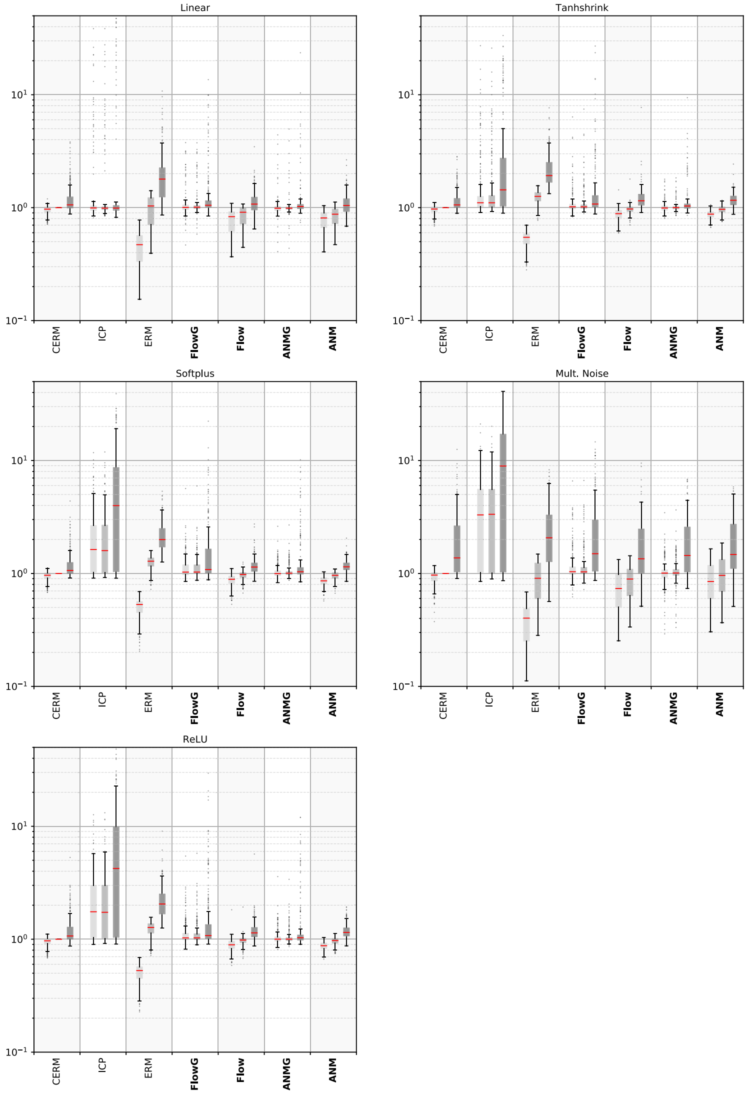


**Fig. 10.** Logarithmic plot of L2 errors, normalized by CERM test error. For each method (ours in bold) from left to right: training error, test error on seen environments, domain generalization error on unseen environments. Scenarios for different mechanisms are shown.

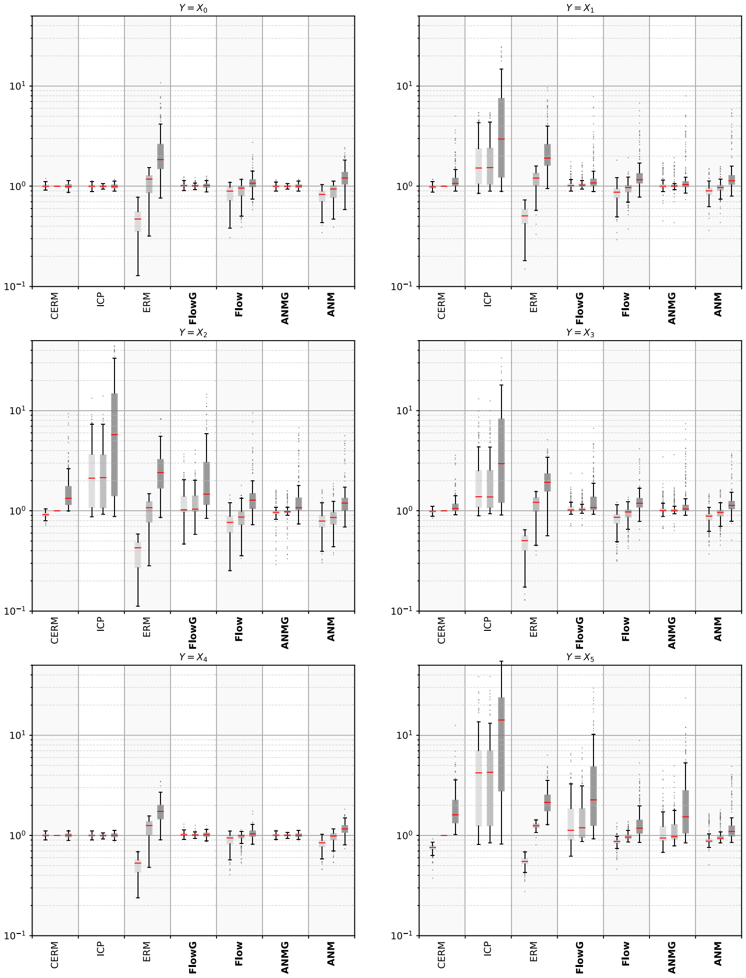


**Fig. 11.** Logarithmic plot of L2 errors, normalized by CERM test error. For each method (ours in bold) from left to right: training error, test error on seen environments, domain generalization error on unseen environments. Scenarios for different target variables are shown.

environment to measure out-of-distribution generalization. Figure 10, 11 and 12 show results according to the underlying mechanisms, target variable or type of intervention respectively. The boxes show the quartiles and the upper whiskers ranges from third quartile to $1.5 \cdot IQR$ where $IQR$ is the interquartile range. Similar for the lower whisker.

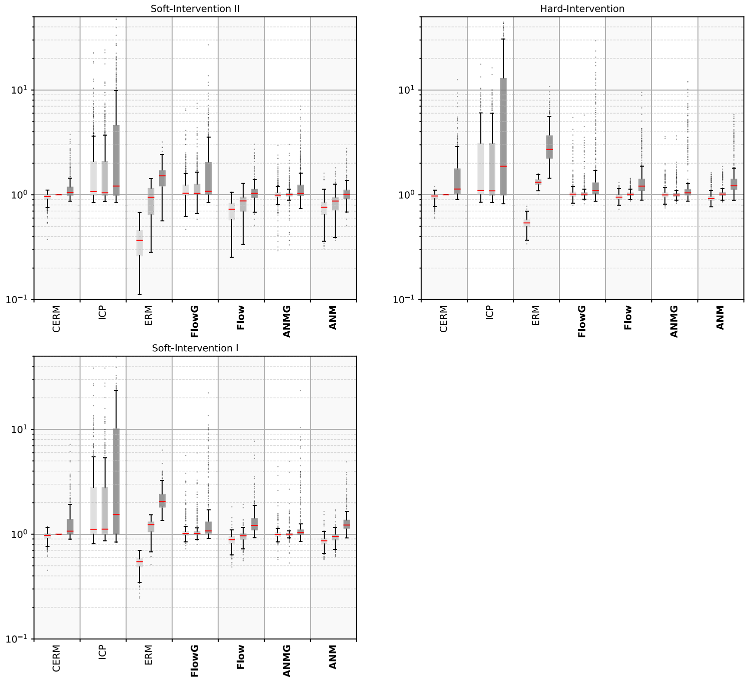


**Fig. 12.** Logarithmic plot of L2 errors, normalized by CERM test error. For each method (ours in bold) from left to right: training error, test error on seen environments, domain generalization error on unseen environments. Scenarios for different intervention types are shown.

# I Experimental Details Colored MNIST

For the training, we use a feed forward neural network consisting of a feature selector followed by a classificator. The feature selector consists of two convolutional layers with a kernel size of 3 with 16 respectively 32 channels followed by a max pooling layer with kernel size 2, one dropout layer ($p = 0.2$) and a fully connected layer mapping to 16 feature dimensions. After the first convolutional layer and after the pooling layer a PReLU activation function is applied. For the classification we use a PReLU activation function followed by a Dropout layer ($p = 0.2$) and a linear layer which maps the 16 features onto the two classes corresponding to the labels.

We use the data generating process from [2]. 50 000 samples are used for training and 10 000 samples as test set. For training, we choose a batch size of 1000 and train our models for 60 epochs. We choose a starting learning rate of $6 \cdot 10^{-3}$. The learning rate is decayed by 0.33 after 20 epochs. We use an L2-Regularization loss weighted by $10^{-5}$. After each epoch we randomly reassign the colors and the labels with the corresponding probabilities. The one-dimensional Wasserstein loss is applied dimension-wise and the maximum over dimensions is computed in order to compare residuals. For the HSIC we use a cauchy kernel with $\sigma = 1$. The invariance loss $\mathcal{L}_I$ is simply the sum of the HSIC and Wasserstein term. For Fig. 3a we trained our model with $\lambda_I \approx 13$. This hyperparameter is chosen from the best run in Fig. 3b. For stability in the case of large $\lambda_I$, we divide the total loss by $\lambda_I$ during training to produce the results in Fig. 3b. For the reported accuracy of IRM, we train with the same network architecture on the dataset where we created training instances online.

## References

1. Ardizzone, L., Lüth, C., Kruse, J., Rother, C., Köthe, U.: Guided image generation with conditional invertible neural networks. arXiv preprint arXiv:1907.02392 (2019)
2. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019)
3. Barber, D., Agakov, F.V.: The IM algorithm: a variational approach to information maximization. In: Advances in Neural Information Processing Systems (2003)
4. Bareinboim, E., Pearl, J.: Causal inference and the data-fusion problem. Proc. Natl. Acad. Sci. **113**(27), 7345–7352 (2016)
5. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: Advances in Neural Information Processing Systems, pp. 137–144 (2007)
6. Chickering, D.M.: Optimal structure identification with greedy search. J. Mach. Learn. Res. **3**(Nov), 507–554 (2002)
7. Frisch, R.: Statistical versus theoretical relations in economic macrodynamics. In: Hendry, D.F., Morgan, M.S. (eds.) Paper given at League of Nations (1995). The Foundations of Econometric Analysis (1938)
8. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(1), 1–35 (2016). 2096–2030
9. Ghassami, A., Kiyavash, N., Huang, B., Zhang, K.: Multi-domain causal structure learning in linear systems. In: Advances in Neural Information Processing Systems, pp. 6266–6276 (2018)
10. Ghassami, A., Salehkaleybar, S., Kiyavash, N., Zhang, K.: Learning causal structures using regression invariance. In: Advances in Neural Information Processing Systems, pp. 3011–3021 (2017)
11. Greenfeld, D., Shalit, U.: Robust learning with the Hilbert-Schmidt independence criterion. arXiv preprint arXiv:1910.00270 (2019)
12. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with Hilbert-Schmidt norms. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 63–77. Springer, Heidelberg (2005). https://doi.org/10.1007/11564089_7
13. Heckman, J.J., Pinto, R.: Causal analysis after haavelmo. Technical report, National Bureau of Economic Research (2013)
14. Heinze-Deml, C., Peters, J., Meinshausen, N.: Invariant causal prediction for nonlinear models. J. Causal Inference **6**(2) (2018)
15. Hoover, K.D.: The logic of causal inference: econometrics and the conditional analysis of causation. Econ. Philos. **6**(2), 207–234 (1990)
16. Huang, B., et al.: Causal discovery from heterogeneous/nonstationary data. J. Mach. Learn. Res. **21**(89), 1–53 (2020)
17. Kalainathan, D., Goudet, O., Guyon, I., Lopez-Paz, D., Sebag, M.: Sam: structural agnostic model, causal discovery and penalized adversarial learning. arXiv preprint arXiv:1803.04929 (2018)

18. Kolouri, S., Pope, P.E., Martin, C.E., Rohde, G.K.: Sliced-Wasserstein autoencoder: an embarrassingly simple generative model. arXiv preprint arXiv:1804.01947 (2018)
19. Krueger, D., et al.: Out-of-distribution generalization via risk extrapolation (REx). arXiv preprint arXiv:2003.00688 (2020)
20. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. Behav. Brain Sciences, **40** (2017)
21. Louizos, C., Welling, M., Kingma, D.P.: Learning sparse neural networks through $l\_0$ regularization. arXiv preprint arXiv:1712.01312 (2017)
22. Magliacane, S., van Ommen, T., Claassen, T., Bongers, S., Versteeg, P., Mooij, J.M.: Domain adaptation by using causal inference to predict invariant conditional distributions. In: Advances in Neural Information Processing Systems, pp. 10846–10856 (2018)
23. Marzouk, Y., Moselhy, T., Parno, M., Spantini, A.: Sampling via measure transport: an introduction. In: Ghanem, R., Higdon, D., Owhadi, H. (eds.) Handbook of Uncertainty Quantification, pp. 1–41. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-11259-6_23-1
24. Meinshausen, N.: Causality from a distributional robustness point of view. In: 2018 IEEE Data Science Workshop (DSW), pp. 6–10. IEEE (2018)
25. Mitrovic, J., McWilliams, B., Walker, J., Buesing, L., Blundell, C.: Representation learning via invariant causal mechanisms. arXiv preprint arXiv:2010.07922 (2020)
26. Mooij, J.M., Magliacane, S., Claassen, T.: Joint causal inference from multiple contexts. arXiv preprint arXiv:1611.10351 (2016)
27. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Trans. Neural Netw. **22**(2), 199–210 (2010)
28. Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. arXiv preprint arXiv:1912.02762 (2019)
29. Pearl, J.: Causality. Cambridge University Press, Cambridge (2009)
30. Peters, J., Bühlmann, P., Meinshausen, N.: Causal inference by using invariant prediction: identification and confidence intervals. J. R. Stat. Soc.: Ser. B (Stat. Methodol.) **78**(5), 947–1012 (2016)
31. Peters, J., Janzing, D., Schölkopf, B.: Elements of Causal Inference: Foundations and Learning Algorithms. MIT Press, Cambridge (2017)
32. Peters, J., Mooij, J.M., Janzing, D., Schölkopf, B.: Causal discovery with continuous additive noise models. J. Mach. Learn. Res. **15**(1), 2009–2053 (2014)
33. Pfister, N., Bühlmann, P., Peters, J.: Invariant causal prediction for sequential data. J. Am. Stat. Assoc. **114**(527), 1264–1276 (2019)
34. Qin, Z., Kim, D.: Rethinking softmax with cross-entropy: neural network classifier as mutual information estimator. arXiv preprint arXiv:1911.10688 (2019)
35. Rojas-Carulla, M., Schölkopf, B., Turner, R., Peters, J.: Invariant models for causal transfer learning. J. Mach. Learn. Res. **19**(1), 1309–1342 (2018)
36. Spirtes, P., Glymour, C.: An algorithm for fast recovery of sparse causal graphs. Soc. Sci. Comput. Rev. **9**(1), 62–72 (1991)
37. Tian, J., Pearl, J.: Causal discovery from changes. In: Uncertainty in Artificial Intelligence (UAI), pp. 512–521 (2001)
38. Xie, C., Chen, F., Liu, Y., Li, Z.: Risk variance penalization: from distributional robustness to causality. arXiv preprint arXiv:2006.07544 (2020)
39. Zhang, K., Gong, M., Schölkopf, B.: Multi-source domain adaptation: a causal view. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)