

Invariant Models for Causal Transfer Learning

Mateo Rojas-Carulla

*Max Planck Institute for Intelligent Systems
Tübingen, Germany*

*Department of Engineering
Univ. of Cambridge, United Kingdom*

MR597@CAM.AC.UK

Bernhard Schölkopf

*Max Planck Institute for Intelligent Systems
Tübingen, Germany*

BS@TUEBINGEN.MPG.DE

Richard Turner

*Department of Engineering
Univ. of Cambridge, United Kingdom*

RET26@CAM.AC.UK

Jonas Peters*

*Department of Mathematical Sciences
Univ. of Copenhagen, Denmark*

JONAS.PETERS@MATH.KU.DK

Editor: Massimiliano Pontil

Abstract

Methods of transfer learning try to combine knowledge from several related tasks (or domains) to improve performance on a test task. Inspired by causal methodology, we relax the usual covariate shift assumption and assume that it holds true for a *subset* of predictor variables: the conditional distribution of the target variable given this subset of predictors is invariant over all tasks. We show how this assumption can be motivated from ideas in the field of causality. We focus on the problem of Domain Generalization, in which no examples from the test task are observed. We prove that in an adversarial setting using this subset for prediction is optimal in Domain Generalization; we further provide examples, in which the tasks are sufficiently diverse and the estimator therefore outperforms pooling the data, even on average. If examples from the test task are available, we also provide a method to transfer knowledge from the training tasks and exploit all available features for prediction. However, we provide no guarantees for this method. We introduce a practical method which allows for automatic inference of the above subset and provide corresponding code. We present results on synthetic data sets and a gene deletion data set.

Keywords: Transfer learning, Multi-task learning, Causality, Domain adaptation, Domain generalization.

1. Introduction

Standard approaches to supervised learning assume that training and test data can be modeled as an i.i.d. sample from a distribution $\mathbb{P} := \mathbb{P}^{(\mathbf{X}, Y)}$. The inputs \mathbf{X} are often vectorial, and the outputs Y may be labels (classification) or continuous values (regression).

*. Most of this work was done while JP was at the Max Planck Institute for Intelligent Systems in Tübingen.

The i.i.d. setting is theoretically well understood and yields remarkable predictive accuracy in problems such as image classification, speech recognition and machine translation (e.g., Schmidhuber, 2015; Krizhevsky et al., 2012). However, many real world problems do not fit into this setting. The field of transfer learning attempts to address the scenario in which distributions may change between training and testing. We focus on two different problems within transfer learning: domain generalization and multi-task learning. We begin by describing these two problems, followed by a discussion of existing assumptions made to address the problem of knowledge transfer, as well as the new assumption we assay in this paper.

1.1 Domain generalization and multi-task learning

Assume that we want to predict a target $Y \in \mathbb{R}$ from some predictor variable $\mathbf{X} \in \mathbb{R}^p$. Consider D training (or source) tasks¹ $\mathbb{P}^1, \dots, \mathbb{P}^D$ where each \mathbb{P}^k represents a probability distribution generating data $(\mathbf{X}^k, Y^k) \sim \mathbb{P}^k$. At training time, we observe a sample $(\mathbf{X}_i^k, Y_i^k)_{i=1}^{n_k}$ for each source task $k \in \{1, \dots, D\}$; at test time, we want to predict the target values of an unlabeled sample from the task T of interest. We wish to learn a map $f : \mathbb{R}^p \rightarrow \mathbb{R}$ with small expected squared loss $\mathcal{E}_{\mathbb{P}^T}(f) = \mathbb{E}_{(\mathbf{X}^T, Y^T) \sim \mathbb{P}^T} (Y^T - f(\mathbf{X}^T))^2$ on the test task T .

In domain generalization (DG) (e.g., Muandet et al., 2013), we have $T = D + 1$, that is, we are interested in using information from the source tasks in order to predict Y^{D+1} from \mathbf{X}^{D+1} in a related yet unobserved test task \mathbb{P}^{D+1} . To beat simple baseline techniques, regularity conditions on the differences of the tasks are required. Indeed, if the test task differs significantly from the source tasks, we may run into the problem of negative transfer (Pan and Yang, 2010) and DG becomes impossible (Ben-David et al., 2010).

If examples from the test task are available during training (e.g., Pan and Yang, 2010; Baxter, 2000), we refer to the problem as asymmetric multi-task learning (AMTL). If the objective is to improve performance in all the training tasks (e.g., Caruana, 1997), we call the problem symmetric multi-task learning (SMTL), see Table 1 for a summary of these settings. In multi-task learning (MTL), which includes both AMTL and SMTL, if infinitely many labeled data are available from the test task, it is impossible to beat a method that learns on the test task and ignores the training tasks.

1.2 Prior work

A first family of methods assumes that **covariate shift** holds (e.g., Quionero-Candela et al., 2009; Schweikert et al., 2009). This states that for all $k \in \{1, \dots, D, T\}$, the conditional distributions $Y^k | \mathbf{X}^k$ are **invariant** between tasks. Therefore, the differences in the joint distribution of \mathbf{X}^k and Y^k originate from a difference in the marginal distribution of \mathbf{X}^k . Under covariate shift, for instance, if an unlabeled sample from the test task is available at training in the DG setting, the training sample can be re-weighted via importance sampling (Gretton et al., 2009; Shimodaira, 2000; Sugiyama et al., 2008) so that it becomes representative of the test task.

1. In this work, we use the expression “task” and “domain” interchangeably.

method	training data from	test domain
Domain Generalization (DG)	$(\mathbf{X}^1, Y^1), \dots, (\mathbf{X}^D, Y^D)$ $(\mathbf{X}^1, Y^1), \dots, (\mathbf{X}^D, Y^D), \mathbf{X}^{D+1}$	$T := D + 1$
Asymm. Multi-Task Learning (AMTL)	$(\mathbf{X}^1, Y^1), \dots, (\mathbf{X}^D, Y^D)$ $(\mathbf{X}^1, Y^1), \dots, (\mathbf{X}^D, Y^D), \mathbf{X}^D$	$T := D$
Symm. Multi-Task Learning (SMTL)	$(\mathbf{X}^1, Y^1), \dots, (\mathbf{X}^D, Y^D)$ $(\mathbf{X}^1, Y^1), \dots, (\mathbf{X}^D, Y^D), \mathbf{X}^1, \dots, \mathbf{X}^D$	all

Table 1: Taxonomy for domain generalization (DG) and multi-task learning (AMTL and SMTL). Each problem can either be used without (first line) or with (second line) additional unlabeled data.

Another line of work focuses on **sharing parameters** between tasks. This idea originates in the hierarchical Bayesian literature (Bonilla et al., 2007; Gao et al., 2008). For instance, Lawrence and Platt (2004) introduce a model for MTL in which the mapping f_k in each task $k \in \{1, \dots, D, T\}$ is drawn independently from a common Gaussian Process (GP), and the likelihood of the latent functions depends on a shared parameter θ . A similar approach is introduced by Evgeniou and Pontil (2004): they consider an SVM with weight vector $w^k = w_0 + v^k$, where w_0 is shared across tasks and v^k is task specific. This allows for tasks to be similar (in which case v^k does not have a significant contribution to predictions) or quite different. Daumé III et al. (2010) use a related approach for MTL when there is one source and one target task. Their method relies on the idea of augmented feature space, which they obtain using two features maps $\Phi^s(\mathbf{X}^s) = (\mathbf{X}^s, \mathbf{X}^s, 0)$ for the source examples and $\Phi^t(\mathbf{X}^t) = (\mathbf{X}^t, 0, \mathbf{X}^t)$ for the target examples. They then train a classifier using these augmented features. Moreover, they propose a way of using available unlabeled data from the target task at training.

An alternative family of methods is based on learning a set of **common features** for all tasks (Argyriou et al., 2007a; Romera-Paredes et al., 2012; Argyriou et al., 2007b; Raina et al., 2007). For instance, Argyriou et al. (2007a,b) propose to learn a set of low dimensional features shared between tasks using L^1 regularization, and then learn all tasks independently using these features. In Raina et al. (2007), the authors construct a similar set of features using L^1 regularization but make use of only unlabeled examples. Chen et al. (2012) proposes to build shared feature mappings which are robust to noise by using autoencoders.

Finally, the assumption introduced in this paper is based on a **causal** view on domain adaptation and transfer.

Schölkopf et al. (2012) relate multi-task learning with the independence between cause and mechanism. This notion is closely related to exogeneity (Zhang et al., 2015b), which roughly states that a causal mechanism mapping a cause X to Y should not depend on the distribution of X . Additionally, Zhang et al. (2013) consider the problem of target and conditional shift when the target variable is causal for the features. They assume that there exists a linear mapping between the covariates in different tasks, and the parameters of this

mapping only depend on the distribution of the target variable. Moreover, Zhang et al. (2015a) argue that the availability of multiple domains is sufficient to drop this previous assumption when the distribution of Y^k and the conditional $\mathbf{X}^k | Y^k$ change independently. The conditional in the test task can then be written as a linear mixture of the conditionals in the source domains. The concept of invariant conditionals and exogeneity can also be used for causal discovery (Peters et al., 2016; Zhang et al., 2015b; Peters et al., 2017).

1.3 Contribution

Taking into account causal knowledge, **our approach** to DG and MTL assumes that covariate shift holds only for a subset of the features. From the point of view of causal modeling (Pearl, 2009), assuming invariance of conditionals makes sense if the conditionals represent causal mechanisms (e.g., Hoover, 1990), see Section 2.3 for details. Intuitively, we expect that a causal mechanism is a property of the physical world, and it does not depend on what we feed into it. If the input (which in this case coincides with the covariates) shifts, the mechanism should thus remain invariant (Hoover, 1990; Janzing and Schölkopf, 2010; Peters et al., 2016). In the anticausal direction, however, a shift of the input usually leads to a changing conditional (Schölkopf et al., 2012). In practice, prediction problems are often not causal — we should allow for the possibility that the set of predictors contains variables that are causal, anticausal, or confounded, i.e., statistically dependent variables without a directed causal link with the target variable. We thus expect that there is a *subset* S^* of predictors, referred to as an **invariant set**, for which the covariate shift assumption holds true, i.e., the conditionals of output given predictor $Y^k | \mathbf{X}_{S^*}^k$ are invariant across $k \in \{1, \dots, D, T\}$. If S^* is a strict subset of all predictors, this relaxes full covariate shift. We prove that knowing S^* leads to robust properties for DG. Once an invariant set is known, traditional methods for covariate shift can be applied as a black box, see Figure 1. In the MTL setting, when labeled or unlabeled examples from the test task are available during training, we might not want to discard the features outside of S^* for prediction. Hence, we also propose a method to leverage the knowledge of the invariant set S^* and the available examples from the test task in order to outperform a method that learns only on the test task.

Finally, note that in this work, we concentrate on the linear setting, keeping in mind that this has specific implications for covariate shift.

1.4 Organization of the paper

Section 2 formally describes our approach and its underlying assumptions; in particular, we assume that an invariant set S^* is known. For DG, we prove in Section 2.1 that predicting using only features in S^* is optimal in an adversarial setting. Moreover, we present an example in which we compare our proposed estimator with pooling the training data, a standard technique for DG. In MTL, when additional labeled examples from T are available, one might want to use all available features for prediction. Section 2.2 provides a method to address this. We discuss a link to causal inference in Section 2.3. Often, an invariant set S^* is not known a priori. Section 3 presents a method for inferring an invariant set from data. Section 4 contains experiments on simulated and real data.

2. Exploiting invariant conditional distributions in transfer learning

Consider a transfer learning regression problem with source tasks $\mathbb{P}^1, \dots, \mathbb{P}^D$, where $(\mathbf{X}^k, Y^k) \sim \mathbb{P}^k$ for $k \in \{1, \dots, D\}$.² We now formulate our main assumptions.

(A1) There exists a subset $S^* \subseteq \{1, \dots, p\}$ of predictor variables such that

$$Y^k | \mathbf{X}_{S^*}^k \stackrel{d}{=} Y^{k'} | \mathbf{X}_{S^*}^{k'} \quad \forall k, k' \in \{1, \dots, D\}. \quad (1)$$

We say that S^* is an **invariant set** which leads to invariant conditionals. Here, $\stackrel{d}{=}$ denotes equality in distribution.

(A1') This invariance also holds in the test task T , i.e., (1) holds for all $k, k' \in \{1, \dots, D, T\}$.

(A2) The conditional distribution of Y given an invariant set S^* is linear: there exists $\alpha \in \mathbb{R}^{|S^*|}$ and a random variable ϵ such that for all $k \in \{1, \dots, D\}$, $[Y^k | \mathbf{X}_{S^*}^k = x] \stackrel{d}{=} \alpha^t x + \epsilon^k$, that is $Y^k = \alpha^t \mathbf{X}_{S^*}^k + \epsilon^k$, with $\epsilon^k \perp \mathbf{X}_{S^*}^k$ and for all $k \in \{1, \dots, D\}$, $\epsilon^k \stackrel{d}{=} \epsilon$.

Assumption (A1') is stronger than (A1) only in the DG setting, where, of course, (A1') and (A2) imply the linearity also in the test task T . While Assumption (A1) is testable from training data, see Section 3, (A1') is not. In covariate shift, one usually assumes that (A1') holds for the set of all features. Therefore, (A1') is a weaker condition than covariate shift, see Figure 1. We regard this assumption as a building block that can be combined with any method for covariate shift, applied to the subset S^* . It is known that it can be arbitrarily hard to exploit the assumption of covariate shift in practice (Ben-David et al., 2010). In a general setting, for instance, assumptions about the support of the training distributions $\mathbb{P}^1, \dots, \mathbb{P}^D$ and the test distribution \mathbb{P}^T must be made for methods such as re-weighting to be expected to work (e.g., Gretton et al., 2009). The aim of our work is not to solve the full covariate shift problem, but to elucidate a relaxation of covariate shift in which it holds given only a subset of the features. We concentrate on linear relations (A2), which circumvents the issue of overlapping supports, for example.

For the remainder of this section, we assume that we are given an invariant subset S^* that satisfies (A1) and (A2). Note that we will also require (A1') for DG. In MTL, the invariance can be tested on the labeled data available from the test task, so (A1) and (A1') are equivalent.

We show how the knowledge of S^* can be exploited for the DG problem (Section 2.1) and in the MTL case (Section 2.2). Here and below, we focus on linear regression using squared loss

$$\mathcal{E}_{\mathbb{P}^T}(\beta) = \mathbb{E}_{(\mathbf{X}^T, Y^T) \sim \mathbb{P}^T} (Y^T - \beta^t \mathbf{X}^T)^2 \quad (2)$$

(the superscript T corresponds to the test task, not to be confused with the transpose, indicated by superscript t). We denote by $\mathcal{E}_{\mathbb{P}^1, \dots, \mathbb{P}^D}(\beta)$ the squared error averaged over the training tasks $k \in \{1, \dots, D\}$.

2. We assume throughout this work the existence of densities and that random variables have finite variance.

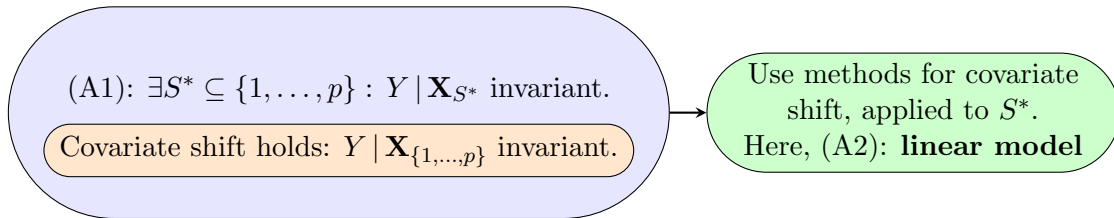


Figure 1: Assumption (A1) (blue) is a relaxation of covariate shift (orange): the covariate shift assumption is a special case of (A1) with $S^* = \{1, \dots, p\}$. Given the invariant set S^* , methods for covariate shift can be applied.

2.1 Domain generalization (DG): no labels from the test task

We first study the DG setting in which we receive no labeled examples from the test task during training time. Throughout this subsection, we assume that additionally to (A1) and (A2), assumption (A1') holds. It is important to appreciate that (A1') is a strong assumption that is not testable on the training data: it is an assumption about the test task. We believe no nontrivial statement about DG is possible without an assumption of this type.

Now, we introduce our proposed estimator, which uses the conditional mean of the target variable given the invariant set in the training tasks. We prove that this estimator is optimal in an adversarial setting.

Proposed estimator. The optimal predictor obtained by minimizing (2) is the conditional mean

$$\beta^{opt} := \arg \min_{\beta \in \mathbb{R}^p} \mathcal{E}_{\mathbb{P}^T}(\beta), \quad (3)$$

which is not available during training time. Given an invariant set S^* satisfying (A1), (A1') and (A2), we propose to use the corresponding conditional expectation as an estimator. In other words, let $\beta^{S^*} = \arg \min_{\beta \in \mathbb{R}^{|S^*|}} (Y^1 - \beta^t \mathbf{X}_{S^*}^1)^2$ be the vector obtained by minimizing the squared loss in the training tasks using only predictors in S^* . We propose as a predictor the vector $\beta^{CS(S^*)} \in \mathbb{R}^p$ obtained by adding zeros to β^{S^*} in the dimensions corresponding to covariates outside of S^* . More formally, we propose to use as a predictor

$$\begin{array}{l} \mathbb{R}^p \rightarrow \mathbb{R} \\ \mathbf{x} \mapsto \mathbb{E}[Y^1 | \mathbf{X}_{S^*}^1 = \mathbf{x}_{S^*}] \end{array} \quad \text{and write} \quad \mathbb{E}[Y^1 | \mathbf{X}_{S^*}^1 = \mathbf{x}_{S^*}] = (\beta^{CS(S^*)})^t \mathbf{x}. \quad (4)$$

Because of (A1), the conditional expectation in (4) is the same in all training tasks. In the limit of infinitely many data, given a subset S , $\beta^{CS(S)}$ is obtained by pooling the training tasks and regressing using only features in S . In particular, $\beta^{CS} := \beta^{CS(\{1, \dots, p\})}$ is the estimator obtained when assuming traditional covariate shift.

Optimality in an adversarial setting. In an adversarial setting, predictor (4) satisfies the following optimality condition; as for the other results, the proof is provided in Appendix A. We state and prove a more general, nonlinear version of Theorem 1 in Appendix A.1.

Theorem 1 (Adversarial) Consider $(\mathbf{X}^1, Y^1) \sim \mathbb{P}^1, \dots, (\mathbf{X}^D, Y^D) \sim \mathbb{P}^D$ and an invariant set S^* satisfying (A1) and (A2). The proposed estimator satisfies an optimality statement over the set of distributions such that (A1') holds: we have

$$\beta^{CS(S^*)} \in \arg \min_{\beta \in \mathbb{R}^p} \sup_{\mathbb{P}^T \in \mathcal{P}} \mathcal{E}_{\mathbb{P}^T}(\beta),$$

where $\beta^{CS(S^*)}$ is defined in (4) and \mathcal{P} contains all distributions over (\mathbf{X}^T, Y^T) , $T = D + 1$, that are absolutely continuous with respect to the same product measure μ and satisfy $Y^T | \mathbf{X}_{S^*}^T \stackrel{d}{=} Y^1 | \mathbf{X}_{S^*}^1$.

Unlike the optimal predictor β^{opt} , the proposed estimator (4) can be learned from the data available in the training tasks. Given a sample $(\mathbf{X}_1^k, Y_1^k), \dots, (\mathbf{X}_{n_k}^k, Y_{n_k}^k)$ from tasks $k \in \{1, \dots, D\}$, we can estimate the conditional mean in (4) by regressing Y^k on $\mathbf{X}_{S^*}^k$. Due to (A1), we may also pool the data over the different tasks and use

$$(\mathbf{X}_1^1, Y_1^1), \dots, (\mathbf{X}_{n_1}^1, Y_{n_1}^1), (\mathbf{X}_1^2, Y_1^2), \dots, (\mathbf{X}_{n_D}^D, Y_{n_D}^D)$$

as a training sample for this regression.

One may also compare the proposed estimator with pooling the training tasks, a standard baseline in transfer learning which corresponds to assuming that usual covariate shift holds. Focusing on a specific example, Proposition 2 in the following paragraph shows that when the test tasks become diverse, predicting using (4) outperforms pooling on average over all tasks.

Comparison against pooling the data. We proved that the proposed estimator (4) does well on an adversarial setting, in the sense that it minimizes the largest error on a task in \mathcal{P} . The following result provides an example in which we can analytically compare the proposed estimator with the estimator obtained from pooling the training data, which is a benchmark in transfer learning. We prove that in this setting, the proposed estimator outperforms pooling the data on average over test tasks when the tasks become more diverse.

Let $\mathbf{X}_{S^*}^k$ be a vector of independent Gaussian variables in task k . Let the target Y^k satisfy

$$Y^k = \alpha^t \mathbf{X}_{S^*}^k + \epsilon^k, \tag{5}$$

where for each $k \in \{1, \dots, D\}$, ϵ^k is Gaussian and independent of $\mathbf{X}_{S^*}^k$. We have $\mathbf{X}^k = (\mathbf{X}_{S^*}^k, Z^k)$, where

$$Z^k = \gamma^k Y^k + \eta^k,$$

for some $\gamma^k \in \mathbb{R}$ and where η^k is Gaussian and independent of Y^k .³ Moreover, assume that the training tasks are balanced. We compare properties of estimator $\beta^{CS(S^*)}$ defined in Equation (4) against the least squares estimator obtained from pooling the training data. In this setting, the tasks differ in coefficients γ^k , which are randomly sampled. We prove that the squared loss averaged over unseen test tasks is always larger for the pooled approach, when coefficients γ^k are centered around zero. In the case where they are centered around a non-zero mean, we prove that when the variance between tasks (in this case, for coefficients γ^k) becomes large enough, the invariant approach also outperforms pooling the data.

3. Using the notation introduced later in Section 2.3, this corresponds to a Gaussian SEM with DAG shown in Fig. 3.

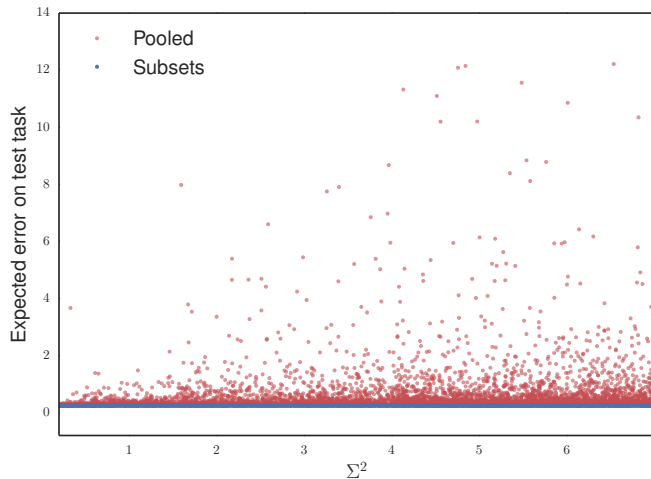


Figure 2: The figure shows expected errors for the pooled approach and the proposed method, see Equation (6). $\mu = 0$. We consider two training tasks over 10,000 simulations. In each, we randomly sample the variance of each covariate in \mathbf{X} , the variance of η , and γ . σ^2 is the same in all tasks. As predicted by Proposition 2 observe that the error from the pooled approach (red) is systematically higher than the error from the prediction using only the invariant subset (blue), and both the error and its variance become large as the variance Σ^2 of coefficients γ^k increases.

Proposition 2 (Average performance) *Consider the model described previously. Moreover, assume that the tasks differ as follows: the coefficients $\gamma^1, \dots, \gamma^D, \gamma^T = \gamma^{D+1}$ are i.i.d. with mean zero and variance $\Sigma^2 > 0$. The tasks do not differ elsewhere. In particular, the distribution of $X_{S^*}^k$ is the same for all tasks. Then the least squares predictor obtained from pooling the D training tasks $\beta^{CS} = (\beta_{S^*}^{CS}, \beta_Z^{CS})$ satisfies:*

$$\mathbb{E}_{\gamma^T} (\mathcal{E}_{\mathbb{P}^T} (\beta^{CS})) \geq \mathbb{E}_{\gamma^T} (\mathcal{E}_{\mathbb{P}^T} (\beta^{CS(S^*)})) = \sigma^2. \quad (6)$$

In particular, this implies the following:

$$\mathbb{E}_{\gamma^1, \dots, \gamma^D, \gamma^T} (\mathcal{E}_{\mathbb{P}^T} (\beta^{CS})) \geq \mathbb{E}_{\gamma^1, \dots, \gamma^D, \gamma^T} (\mathcal{E}_{\mathbb{P}^T} (\beta^{CS(S^*)})) = \sigma^2. \quad (7)$$

Moreover, if the coefficients $\gamma^1, \dots, \gamma^D, \gamma^T$ are i.i.d. with non-zero mean μ , (6) holds for fixed $\gamma^1, \dots, \gamma^D$ if $\Sigma^2 \geq P(\mu)$, where P is a polynomial in μ , see Appendix A.2 for details.

The proof of Proposition 2 can be found in Appendix A.2. Figure 2 visualizes Proposition 2 for two training tasks, it shows the expected errors for the pooled and invariant approaches, see (6), as the variance Σ^2 increases. Recall that Σ^2 corresponds to the variance of coefficients γ^k , and thus indicates how different the tasks are. The expected errors are computed using the analytic expression found in the proof of Proposition 2. As predicted by Proposition 2, the expected error of the pooled approach always exceeds the one of the proposed method (the coefficients γ^k are centered around zero), see Equation (6). As Σ^2 tends to

zero, γ^k is close to zero in all tasks, which explains the equality of both the pooled and invariant errors for the limit case Σ approaching 0. For coefficients γ^k centered around a non zero value, Equation (6) does not necessarily hold for small Σ^2 .

Proposition 2 presents a setting in which the invariant approach outperforms pooling the data when the test errors are averaged over γ , i.e., $\mathbb{E}_{\gamma^T} (\mathcal{E}_{\mathbb{P}^T} (\beta^{CS})) \geq \mathbb{E}_{\gamma^T} (\mathcal{E}_{\mathbb{P}^T} (\beta^{CS}))$. It is also clear to see that the equality of the distribution of ϵ^k in Equation (5) for all $k \in \{1, \dots, D\}$ leads to $\text{Var}_{\gamma} (\mathcal{E}_{\mathbb{P}^T} (\beta^{CS(S^*)})) = 0$, thus our invariant estimator minimizes the variance of the test errors across all related tasks.

2.2 Multi-task learning (MTL): combining invariance and task-specific information

In MTL, a labeled sample $(\mathbf{X}_i^T, Y_i^T)_{i=1}^{n_T}$ is available from the test task and the goal is to transfer knowledge from the training tasks. As before, we are given an invariant set S^* satisfying (A1) and (A2). Can we combine the invariance assumption with the new labeled sample and perform better than a method that trains only on the data in the test task? According to (A1) and (A2), the target satisfies $Y^k = \alpha^t \mathbf{X}_{S^*}^k + \epsilon^k$, where the noise ϵ^k has zero mean and finite variance, is independent of $\mathbf{X}_{S^*}^k$ and has the same distribution in the different tasks $k \in \{1, \dots, D, T\}$. Our objective is to use the knowledge gained from the training tasks to get a better estimate of β^{opt} defined in Equation (3). We describe below a way to tackle this using missing data methods.

Missing data approach In this section, we specify how we propose to tackle MTL by framing it as a missing data problem. While the idea is presented in the context of AMTL, it can be used for SMTL in the same way. In order to motivate the method, assume that for each $k \in \{1, \dots, D, T\}$, there exists another probability distribution \mathbb{Q}^k with density q^k having the following properties: (i) when restricted to $(\mathbf{X}_{S^*}^k, Y^k)$, \mathbb{Q}^k coincides with \mathbb{P}^k , (ii) the conditional $q^T(y | \mathbf{x}_{S^*}, \mathbf{x}_N)$ coincides with $p^T(y | \mathbf{x}_{S^*}, \mathbf{x}_N)$ on the test task and (iii) $q(y | \mathbf{x}_{S^*}, \mathbf{x}_N) := q^k(y | \mathbf{x}_{S^*}, \mathbf{x}_N)$ is the same in all tasks (which is not satisfied by \mathbb{P}^k , of course). The goal of learning the regression model from Y on \mathbf{X}_{S^*} and \mathbf{X}_N in \mathbb{P}^T coincides with the task of learning the same regression model in \mathbb{Q}^T . Property (iii) implies that we can pool the data from all tasks \mathbb{Q}^k . This is not possible, of course, for the given data, which we have received from the distributions \mathbb{P}^k . But now assume that in all training tasks, we only have access to the marginal $(\mathbf{X}_{S^*}^k, Y^k)$ from \mathbb{Q}^k . Any method that addresses the regression under these constraints be used with the data available because of (i). We first prove the existence of such distributions \mathbb{Q}^k :

Proposition 3 (Correctness of transfer) *Let S^* be an invariant set verifying (A1) and (A2). For $k \in \{1, \dots, D, T\}$, denote by $(\mathbf{x}, y) \mapsto p^k(\mathbf{x}, y)$ the density of \mathbb{P}^k . Then there exists a function $q : \mathbb{R}^p \rightarrow \mathbb{R}^+$ such that for each $k \in \{1, \dots, D, T\}$, there exists a distribution \mathbb{Q}^k with density q^k such that for all $(\mathbf{x}, y) \in \mathbb{R}^{d+1}$, for all $k \in \{1, \dots, D, T\}$,*

- i) $q^k(\mathbf{x}_{S^*}, y) = p^k(\mathbf{x}_{S^*}, y)$,
- ii) $q^T(y | \mathbf{x}_{S^*}, \mathbf{x}_N) = p^T(y | \mathbf{x}_{S^*}, \mathbf{x}_N)$,
- iii) $q^k(y | \mathbf{x}_{S^*}, \mathbf{x}_N) = q(y | \mathbf{x}_{S^*}, \mathbf{x}_N)$.

The proof for Proposition 3 can be found in Appendix A.3. Following the previous intuition, for the training tasks $k \in \{1, \dots, D\}$, we hide the data of \mathbf{X}_N^k and pretend the data in each

task $k \in \{1, \dots, D, T\}$ come from \mathbb{Q}^k . Note that some of the data are only missing for the training tasks. More precisely, \mathbf{X}_N^k is missing for $k \in \{1, \dots, D\}$, while because of (i) in Proposition 3, $(\mathbf{X}_{S^*}^k, Y^k)$ is available for all tasks $k \in \{1, \dots, D, T\}$. We thus pool the data and learn a regression model of Y versus $(\mathbf{X}_{S^*}, \mathbf{X}_N)$ by maximizing the likelihood of the observed data.

We formalize the problem as follows. Let $(\mathbf{Z}_i)_{i=1}^n = (\mathbf{X}_{S^*,i}, \mathbf{X}_{N,i}, Y_i)_{i=1}^n$ be a pooled sample of the available data from the training tasks and the test task, in which $\mathbf{X}_{N,i}$ is considered missing if \mathbf{X}_i is drawn from one of the training tasks. Here, $n = \sum_{k=1}^T n_k$ is the total number of training and test examples. Denote by $\mathbf{Z}_{obs,i}$ the components of \mathbf{Z}_i which are not missing. In particular, $\mathbf{Z}_{obs,i} = \mathbf{Z}_i$ if i is drawn from the test task and $\mathbf{Z}_{obs,i} = (\mathbf{X}_{S^*,i}, Y_i)$ otherwise. Moreover, let Σ be a $(p+1) \times (p+1)$ positive definite matrix, and Σ_i is the submatrix of Σ which corresponds to the observed features for example i . If example i is drawn from a training task, Σ_i is of size $(|S^*|+1) \times (|S^*|+1)$, and $(p+1) \times (p+1)$ otherwise. The log-likelihood based on the observed data for matrix Σ satisfies:

$$\ell(\Sigma) = \text{const} - \frac{1}{2} \sum_{i=1}^n \det(\Sigma_i) - \frac{1}{2} \mathbf{Z}_{obs,i}^T \Sigma_i^{-1} \mathbf{Z}_{obs,i}, \quad (8)$$

and our goal is to find Σ which maximizes (8). This model for the likelihood assumes that the data is multi-variate Gaussian with covariance matrix Σ .

When all data are observed, the least squares estimator β^{opt} can be seen as the result of a two step procedure. First, (8) is maximized for the sample covariance matrix. Then, one computes the conditional mean $\mathbb{E}[Y | \mathbf{X} = \mathbf{x}]$ of the estimated joint distribution of (\mathbf{X}, Y) . In the case of missing data, however, the sample covariance matrix does no longer maximize (8), see paragraph ‘*A naive estimator for comparison*’ below. Instead, we maximize (8) using EM.

Chapter 11 in Little and Rubin (1986) provides the update equations for optimizing Equation (8) using EM. More precisely, given an estimate Σ^r of the covariance matrix at step r , the algorithm goes as follows.

E step: For an example i , we define

$$\mathbf{Z}_i^r := \begin{cases} \mathbf{Z}_i & \text{if example } i \text{ is from the test task,} \\ (\mathbf{X}_{S^*,i}, \mathbb{E}(\mathbf{X}_N^r | \mathbf{Z}_{obs,i}), Y_i) & \text{otherwise.} \end{cases}$$

Here, we are essentially imputing the data for \mathbf{X}_N in the training tasks by the conditional mean given the observed data, using the current estimate of the covariance matrix Σ^r . The conditional expectation is computed using the current estimate Σ^r and the Gaussian conditioning formula:

$$\mathbb{E}(\mathbf{X}_N^r | \mathbf{Z}_{obs,i}) = \Sigma_{N Z_{obs}}^r (\Sigma_{Z_{obs}}^r)^{-1} \mathbf{Z}_{obs,i},$$

where $\Sigma_{N Z_{obs}}^r$ is the submatrix of Σ^r corresponding to the cross-covariance between \mathbf{X}_N and (\mathbf{X}_{S^*}, Y) , and $\Sigma_{Z_{obs}}^r$ is the submatrix corresponding to the covariance of (\mathbf{X}_{S^*}, Y) . For examples from the test task, we simply copy the example, since $\mathbb{P}^T = \mathbb{Q}^T$. Moreover, define

$$C_{N,i}^r := \begin{cases} 0 & \text{if example } i \text{ is from the test task,} \\ \text{Cov}(\mathbf{X}_N^r | \mathbf{Z}_{obs,i}) = \Sigma_N^r - \Sigma_{N Z_{obs}}^r (\Sigma_{Z_{obs}}^r)^{-1} \Sigma_{Z_{obs} N}^r & \text{otherwise.} \end{cases}$$

M step: compute the sample covariance given the imputed data:

$$\Sigma^{r+1} = \frac{1}{n} \mathbb{E} \left(\sum_{i=1}^n \mathbf{Z}_i^r (\mathbf{Z}_i^r)^t \mid \mathbf{Z}_{obs,i}, \Sigma^r \right) = \frac{1}{n} \sum_{i=1}^n \mathbf{Z}_i^r (\mathbf{Z}_i^r)^t + C_i^r,$$

where C_i^r is a $(p+1) \times (p+1)$ matrix whose submatrix corresponding to features in N is $C_{N,i}^r$, and the remaining elements are 0. The intuition for the M step is simple: we compute the sample covariance with the values imputed for \mathbf{X}_N . Since these values are being imputed, matrix C adds uncertainty for the corresponding values.

Once the algorithm has converged, we can read off the regression coefficient from the joint covariance matrix as $\mathbb{E}[Y \mid \mathbf{X}_{S^*} = \mathbf{x}_{S^*}]$. The whole procedure is initialized with the sample covariance matrix computed with the available labeled sample from T .

Incorporating unlabeled data The previous method also allows us to incorporate unlabeled data from the test task. Indeed, assume that an unlabeled sample $\mathbf{X}^T = (\mathbf{X}_{S^*}^T, \mathbf{X}_N^T)$ from the test task is also available at training time. This can be incorporated in the previous framework since the label Y can be considered to be missing (as opposed to \mathbf{X}_N^T previously). We can then write $\mathbf{Z}_i^r = (\mathbf{X}_{S^*,i}, \mathbf{X}_{N,i}, \mathbb{E}(Y_i^r \mid \mathbf{Z}_{obs,i}))$ for the unlabeled data, thus imputing the value of Y in the E-step by the conditional mean given $(\mathbf{X}_{S^*,i}, \mathbf{X}_{N,i})$. The added covariance is then $C_{Y,i}^r = \text{Var}(Y)^r - \Sigma_{Y Z_{obs}}^r (\Sigma_{Z_{obs}}^r)^{-1} \Sigma_{Z_{obs} Y}^r$. The rest of the algorithm remains unchanged.

A naive estimator for comparison In the population setting, Proposition 5 in Appendix A.4 provides an expression for β^{opt} as a function of α and ϵ from Assumption (A2). As in the previous paragraph, one could try to estimate the covariance matrix of (\mathbf{X}, Y) using the knowledge of α and ϵ from the training tasks, and then read off the regression coefficients. In the presence of a finite amount of labeled and unlabeled data from the test task, a naive approach would thus plug in the knowledge of α and ϵ as follows: the entries of $\hat{\Sigma}_{\mathbf{X}, Y}$ that correspond to the covariances between \mathbf{X}_{S^*} and Y are replaced with $\hat{\Sigma}_{\mathbf{X}_{S^*}} \cdot \alpha$, and the entry corresponding to the variance of Y is replaced by $\alpha^t \hat{\Sigma}_{\mathbf{X}_{S^*}} \alpha + \text{Var}(\epsilon)$. This, however, often performs worse than forgetting about α and using the data in the test domain only, see Figure 5 (left). Why is this the case? The naive solution described above leads to a matrix Σ that does not only *not* maximize (8) but that often is not even positive definite. One needs to optimize over the free parameters of Σ , which corresponds to the covariance between \mathbf{X}_N and Y , given the constraint of positive definiteness. For comparison, we modified the naive approach as follows. First, we find a positive definite matrix satisfying the desired constraints. In order to do this, we solve a semi-definite Program (SDP) with a trivial objective which always equals zero. Then, we maximize the likelihood (8) over the free parameters of Σ with a Nelder-Mead simplex algorithm. The constrained optimization problem can be shown to be convex in the neighborhood of the optimum (Zwiernik et al., 2017, Sec. 3) if the number of data in the test domain grows. While gradients can be computed for this problem, gradient-based methods seem to perform poorly in practice (experiments are not shown for gradient based methods).

In an idealized scenario, infinite amount of unlabeled data in the test and labeled data in the training tasks could provide us with $\Sigma_{\mathbf{X}}$, $\Sigma_{(\mathbf{X}_{S^*}, Y)}$ and $\text{Var}(Y)$. We could then plug in these values into Σ and optimize over the remaining parameters, see $\beta^{CS(cau+,i.d.)}$ in

Figure 5 (left). In practice, we have to estimate $\Sigma_{\mathbf{X}}$, $\Sigma_{(\mathbf{x}_{S^*}, Y)}$ and $\text{Var}(Y)$ from data. Thus, the EM approach mentioned above constitutes the more principled approach.

2.3 Relation to causality

In this section, we provide a brief introduction to causal notions in order to motivate our method. More specifically, we show that under some conditions, the set S^* of causal parents verifies Assumptions (A1) and (A1'). Structural equation models (SEMs) (Pearl, 2009) are one possibility to formalize causal statements. We say that a distribution over random variables $\mathbf{X} = (X_1, \dots, X_p)$ is induced by a structural equation model with corresponding graph \mathcal{G} if each variable X_j can be written as a deterministic function of its parents $\mathbf{PA}_j^{\mathcal{G}}$ (in \mathcal{G}) and some noise variable N_j :

$$X_j = f_j(X_{\mathbf{PA}_j^{\mathcal{G}}}, N_j), \quad j = 1, \dots, p. \quad (9)$$

Here, the graph is required to be acyclic and the noise variables are assumed to be jointly independent. An SEM comes with the ability to describe *interventions*. Intervening in the system corresponds to replacing one of the structural equations (9). The resulting joint distribution is called an intervention distribution. Changing the equation for variable X_j usually affects the distribution of its children for example, but never the distribution of its parents. Consider now an SEM over variables (\mathbf{X}, Y) . Here, we do not specify the graphical relation between Y and the other nodes: Y may or may not have children or parents. Suppose further that the different tasks $\mathbb{P}^1, \dots, \mathbb{P}^D$ are intervention distributions of an underlying SEM with graph structure \mathcal{G} . If the target variable has not been intervened on, then the set $S^* := \mathbf{PA}_Y^{\mathcal{G}}$ satisfies Assumptions (A1) and (A1'). This means that as long as the interventions will not take place at the target variable, the set S^* of causal parents will satisfy Assumptions (A1) and (A1').

Recently, Peters et al. (2016) have given several sufficient conditions for the identifiability of the causal parents in the linear Gaussian framework. E.g., if the interventions take place at informative locations, or if we see sufficiently many different interventions, the set of causal parents is the *only* set S^* that satisfies Assumptions (A1) and (A1'). If there exists more than one set leading to invariant predictions, they consider the intersection of all such subsets. In this sense, seeing more environments helps for identifying the causal structure. In this work, we are interested in prediction rather than causal discovery. Therefore, we try to find a trade-off between models that predict well and invariant models that generalize well to other domains. That is, in the DG setting, we are interested in the subset which leads to invariant conditionals and minimizes the prediction error across training tasks.

If the tasks \mathbb{P}^k correspond to interventions in an SEM, we may construct an extended SEM with a parent-less environment variable E that points into the intervened variables. Then, \mathbb{P}^k equals the distribution of $(\mathbf{X}, Y) | E = k$, see (Peters et al., 2016, Appendix C). If the distribution of (\mathbf{X}, Y, E) is Markov and faithful w.r.t. the extended graph, the smallest set S that leads to invariant conditionals and to best prediction is a subset of the Markov blanket of Y : certainly, it contains all parents of Y ; if it includes a descendant of Y , this must be a child of Y (which yields better prediction and still blocks any path from Y to E); analogously, any contained ancestor of a child of Y must be a parent of that child.

3. Learning invariant conditionals

In the previous section, we have seen how a known invariant subset $S^* \subseteq \{1, \dots, p\}$ of predictors leading to invariant conditionals $Y^k | \mathbf{X}_{S^*}^k$, see Assumptions (A1) and (A1'), can be beneficial in the problems of DG and MTL. In practice, such a set S^* is often unknown. We now present a method that aims at *inferring* an invariant subset from data. Throughout this paper, we denote by S any subset of features, while S^+ is an invariant set (which is not necessarily unique) for which (A1) holds. Such a subset S^+ *does not necessarily satisfy both Assumptions (A1) and (A1')*. Indeed, in DG, only (A1) is testable in the training data. More precisely, if several invariant sets which satisfy (A1) are found, and only some of them satisfy (A1'), we cannot find these from data. We therefore have to add a criterion allowing us to select among several invariant sets. The method we propose provides an estimator \hat{S} for an invariant subset S^+ , which is chosen as the subset satisfying Assumption (A1) which maximizes predictive accuracy on a validation set. In MTL, we still write S^+ , even if we could then write S^* as (A1') becomes testable. It is summarized in Algorithm 1, code is provided in https://github.com/mrojascarulla/causal_transfer_learning.

3.1 Our method.

Algorithm 1: Subset search	
Inputs:	Sample $(\mathbf{x}_i^k, y_i^k)_{i=1}^{n_k}$ for tasks $k \in \{1, \dots, D\}$, threshold δ for independence test.
Outputs:	Estimated invariant subset \hat{S} .
1	Set $S_{acc} = \{\}$, $MSE = \{\}$.
2	for $S \subseteq \{1, \dots, p\}$ do
3	linearly regress Y on \mathbf{X}_S and compute the residuals $R_{\beta^{CS(S)}}$ on a validation set.
4	compute $H = \text{HSIC}_b \left((R_{\beta^{CS(S)}, i}, K_i)_{i=1}^n \right)$ and the corresponding p-value p^* (or the p-value from an alternative test, e.g., Levene test.).
5	if $p^* > \delta$ then
6	compute $\hat{\mathcal{E}}_{\mathbb{P}^1, \dots, D}(\beta^{CS(S)})$, the empirical estimate of $\mathcal{E}_{\mathbb{P}^1, \dots, D}(\beta^{CS(S)})$ on a validation set.
7	$S_{acc} \cdot \text{add}(S)$, $MSE \cdot \text{add}(\hat{\mathcal{E}}_{\mathbb{P}^1, \dots, D}(\beta^{CS(S)}))$
8	end
9	end
10	Select \hat{S} according to <i>RULE</i> , see Section 3.4.

Consider a set of D tasks, a target variable Y^k and a vector \mathbf{X}^k of p predictor variables in task k . For $\beta \in \mathbb{R}^p$, we define the residual in task k as:

$$R_{\beta}^k = Y^k - \beta^t \mathbf{X}^k, \quad k \in \{1, \dots, D\}. \quad (10)$$

By Assumptions (A1) and (A2), there exists a subset S^+ and some vector $\beta^{CS(S^+)}$ such that for all $j \notin S^+$, $\beta_j^{CS(S^+)} = 0$ and $R_{\beta^{CS(S^+)}}^1 \stackrel{d}{=} \dots \stackrel{d}{=} R_{\beta^{CS(S^+)}}^D$. Such a set S^+ is not necessarily unique. As stated in (Peters et al., 2016), the number of invariant subsets decreases as more

Algorithm 2: Greedy subset search	
	Inputs: Sample $(\mathbf{x}_i^k, y_i^k)_{i=1}^{n_k}$ for tasks $k \in \{1, \dots, D\}$, threshold δ for independence test.
	Outputs: Estimated invariant set \hat{S}_{greedy} .
1	Set $S_{acc} = \{\}$, $\hat{S}_{current} = \{\}$, $MSE = \{\}$.
2	for $i \in \{1, \dots, n_{iters}\}$ do
3	Set $stat_{min} = \infty$.
4	for $S \in \mathcal{S}_{\hat{S}_{current}}$ do
5	linearly regress Y on \mathbf{X}_S and compute the residuals $R_{\beta^{CS}(S)}$ on a validation set.
6	compute $H = \text{HSIC}_b \left((R_{\beta^{CS}(S),i}, K_i)_{i=1}^n \right)$ and the corresponding p-value p^* (or the p-value from an alternative test, e.g. Levene test.).
7	if $p^* > \delta$ then
8	compute $\hat{\mathcal{E}}_{\mathbb{P}^{1,\dots,D}}(\beta^{CS}(S))$, the empirical estimate of $\mathcal{E}_{\mathbb{P}^{1,\dots,D}}(\beta^{CS}(S))$ on a validation set.
9	$S_{acc}.\text{add}(S)$, $MSE.\text{add}(\hat{\mathcal{E}}_{\mathbb{P}^{1,\dots,D}}(\beta^{CS}(S)))$,
10	set $\hat{S}_{current} = S$.
11	end
12	else if $H < stat_{min}$ then
13	set $\hat{S}_{current} = S$, $stat_{min} = H$.
14	end
15	end
16	end
17	Select \hat{S} according to <i>RULE</i> , see Section 3.4.

different tasks are observed at training time. We propose to do an exhaustive search over subsets S of predictors and statistically test for equality of the distribution of the residuals in the training tasks, see the section below. Among the accepted subsets, we select the subset \hat{S} which leads to the smallest error on a validation set. This is a fundamental difference to the method proposed by Peters et al. (2016). Indeed, while our method addresses the transfer problem, Peters et al. (2016) is about causal discovery. Algorithm 1 finds an invariant subset which also leads to the lowest validation error. This subset may contain covariates which are non causal, see Section 4.3 for further details. On the other hand, Peters et al. (2016) estimate the causal parents (with coverage guarantee). Such an approach has a different purpose and performs very badly both in DG and MTL: e.g., when all tasks are identical, it uses the empty set as predictors, while our method selects the full set of predictors.

In Section 3.3, we propose two solutions for when the number of predictors p is too large for an exhaustive search: a greedy method and variable selection. While the algorithms are presented using linear regression, the extension to a nonlinear framework is straightforward. In particular, linear regression can be replaced by a nonlinear regression method.

3.2 Statistical tests for equality of distributions.

In order to test whether a subset S leads to invariant conditionals, we can use a statistical test to check whether the residuals $R_{\beta^{CS(S)}}^k$ have the same distribution in all tasks $k \in \{1, \dots, D\}$. We propose two possible methods.

For Gaussian data, one can use a Levene test (Levene, 1960) to test whether the residuals have the same variance in all tasks; their means are zero as long as an intercept is included in the regression model.

As an alternative, we propose a nonparametric D -sample test by testing whether the residuals are independent of the task index. This test is a direct application of HSIC (Gretton et al., 2007) but to our knowledge, is novel. Suppose that the index of the task can be considered as a random variable K . We consider the sample $Z = (R_{\beta^S, i}, K_i)_{i=1}^n$ as drawn from a joint distribution over residuals and task indices, where $n = \sum_{k=1}^D n_k$ and $K_i \in \{1, \dots, D\}$ is a discrete value indicating the index of the corresponding task. The residuals have the same distribution in all training tasks if and only if R_{β^S} and K are independent. Two characteristic kernels are used: a kernel κ is used for embedding the residuals and a trivial kernel d such that $d(i, j) = \delta_{ij}$ is used for K . Let therefore $\text{HSIC}(R_{\beta^S}, K)$ denote the value of the HSIC (Gretton et al., 2007) between R_{β^S} and K , and let $\text{HSIC}_b(Z)$ be the corresponding test statistic. A subset S is accepted if it leads to accepting the null hypothesis of independence between R_{β^S} and K at level δ .

Both in the case of the Levene test and the D -sample test, the test outputs a p-value p^* , and we accept the null H_0 if $p^* > \delta$. Among these accepted subsets, we output the set \hat{S} which leads to the smallest loss on a validation set. The test level δ is given as an input to our method and allows for a trade-off between predictive accuracy and exploiting invariance. As δ tends to zero, the null is accepted for all subsets and we then select all features, which is equivalent to covariate shift. When δ approaches one, no subset is accepted as invariant. Our method then reduces to the mean prediction. In order to compute p-values, a Gamma approximation is used for the distribution of $\text{HSIC}_b(Z)$ under the null.

For non-additive models, one may even apply a conditional independence test (e.g., Zhang et al., 2011; Fukumizu et al., 2008) to test whether K is independent of $Y | \mathbf{X}_S$.

3.3 Scalability to a large number of predictors

When the number of features p is large, full subset search is computationally not feasible. We propose two solutions for this scenario. If one has reasons to believe that the signal is sparse, that is the true set S^* is small, one may use a **variable selection** technique such as the Lasso (Tibshirani, 1996) as a first step. Under the assumptions described in Section 2.3, we know that the invariant set with the best prediction in the training tasks can be assumed to be a subset of the relevant features (which here equals the Markov blanket of Y). Thus, if variable screening is satisfied, i.e., one selects all relevant variables and possibly more, the pre-selection step does not change the result of Algorithm 1 in the limit of infinitely many data. For linear models with ℓ_1 penalization, variable screening is a well studied problem, see, e.g., compatibility and β_{min} conditions (Bühlmann and van de Geer, 2011, Chapter 2.5).

Alternatively, one may perform a **greedy search** over subsets when full subset search is not feasible. Denote by \mathcal{S}_S the collection of neighboring sets of a set S obtained by adding or

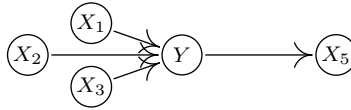


Figure 3: Example of a directed acyclic graph, see Section 2.3. If Y is not intervened on, the conditional $Y | X_1, X_2, X_3$ remains invariant.

removing exactly one predictor in S . If no subset has been accepted at a given iteration, we select the neighbor leading to the smallest test statistic. If a neighbor is accepted, we select the one which leads to the smallest training error. We start with the p subsets with only one element, and allow to add or remove a single predictor at each step, see Algorithm 2. As often for greedy methods, there is no theoretical guarantee.

3.4 Subset selection in MTL

In DG, among the accepted subsets, we select the set \hat{S} which leads to the lowest validation error. In MTL, however, a labeled sample from the test task T is available at training time. Therefore, Algorithm 1 is slightly modified. First, we get all the sets for which H_0 is accepted. Then, we select the accepted set \hat{S} which leads to the smallest 5 fold cross validation error. For each subset, we compute the least squares coefficients using the procedure described in Section 2.2, and measure the prediction error on the held out validation set. Using the notation of Algorithm 1, let S_{acc} be the set of subsets accepted as invariant, and let MSE be the set of their corresponding squared errors on the validation set. The following rules are used for selecting an invariant set in DG and MTL.

- i) **RULE for DG:** Return $\hat{S} = S_{acc}[\arg \min MSE]$.
- ii) **RULE for MTL:** Define $CV_{acc} = \{\}$. For each set $S \subseteq S_{acc}$, do $CV_{acc}.add(CV_S)$, where CV_S is the 5-fold cross validation error over the labeled test data obtained by optimizing (8) using EM with subset S .
Return $\hat{S} = S_{acc}[\arg \min CV_{acc}]$.

Given a set of $k \in \{1, \dots, T\}$ training tasks, a collection of sets $\hat{S}_1, \dots, \hat{S}_u$ (eventually empty) is obtained, all of which lead to accepting the null hypothesis of invariance between the training tasks in DG. Our methods use the MSE on a validation set as a criterion for selecting a subset among these u candidates. This is a design choice which is dependent on the specific application, and can be modified. For instance, if being conservative is important, the MSE may be an inappropriate choice. One may be then interested in combining confidence intervals for the accepted sets. One idea is to consider all accepted sets at the same time, one of which is, with probability $1 - \alpha$, the set S^* from Assumption (A1'). These sets yield different predictions, one of which stems from S^* , again, with probability $1 - \alpha$. In some settings, it might be helpful to output the whole set of predictions. If one is interested in confidence intervals, these may be combined by taking its union. Heinze-Deml et al. (2018) discuss this idea in the context of prediction under interventions.

estimator	description
$\beta^{CS(cau)}$	Linear regr. with true causal predictors (often unknown in practice).
$\beta^{CS(\hat{S})}$	Finding the invariant set \hat{S} using full subset search and performing lin. regr. using predictors in \hat{S} . \hat{S}_{greedy} corresponds to finding the invariant set using a greedy procedure. \hat{S}_{Lasso} corresponds to doing variable selection using Lasso as a first step, then doing full subset search on the selected features.
β^{CS}	Pooling the training data and using linear regr.
$\beta^{CS(\hat{S}+)}$	Finding the invariant set \hat{S} using full subset search and solve the optimization problem described in 'A naive estimator for comparison'.
$\beta^{CS(\hat{S}\#)}$	Finding the invariant set \hat{S} using full subset search and maximizing (8) for MTL using EM.
β^{mean}	Pooling the training data and outputting the mean of the target.
β^{dom}	Linear regression using only the available labeled sample from T .
β^{MTL}	Multi-task feature learning estimator (Argyriou et al., 2007a).
β^{DICA}	DICA (Muandet et al., 2013) with rbf kernel.
β^{mDA}	Pooling the training data and an unlabeled sample from T , learning features using mSDA (Chen et al., 2012) with one layer and linear output, then using linear regr.

Table 2: Estimators used in the numerical experiments. A '+' next to a subset S corresponds to the method for MTL described in the last paragraph of Section 2.2.

4. Experiments

We compare our estimator to different methods, which are summarized in Table 2. $\beta^{CS(cau)}$ uses the ground truth for S^* when it is available, $\beta^{CS(\hat{S})}$ corresponds to full search using Algorithm 1, β^{CS} uses the pooled training data, β^{MTL} performs the Multi-task feature learning algorithm (Argyriou et al., 2007a) for the MTL setting and β^{DICA} performs DICA (Muandet et al., 2013) for DG. For DICA, which is a nonlinear method, the kernel matrices are constructed using an rbf kernel, and the length-scale of the kernel is selected according to the median heuristic. In the MTL setting, we combine the invariance with task specific information by optimizing (8) using EM, resulting in regression coefficients $\beta^{CS(\hat{S}\#)}$ and $\beta^{CS(cau\#)}$ when the ground truth is known. Finally, $\beta^{CS(cau\#,UL)}$ indicates that unlabeled data from T was also available. For reference, Figure 5 (left) provides results for $\beta^{CS(\hat{S}+)}$ and $\beta^{CS(cau+)}$, which correspond to the estimators obtained by solving the constrained optimization problem described in the paragraph 'A naive estimator for comparison' of Section 2.2 ($\beta^{CS(cau+)}$ uses the ground truth for S^* and α), while β^{naive} imputes the covariance matrices but does not optimize the free parameters. $\beta^{CS(cau+,i.d.)}$ (infinite data) also assumes that we know the ground truth for the entries of the covariance matrix for the test task corresponding to the covariance of \mathbf{X} , the covariance between \mathbf{X}_{S^*} and Y , and the variance of Y .

4.1 Synthetic data set

In this section, we generate a synthetic data set in which the causal structure of the problem is known. For all experiments, we choose $\delta = 0.05$ as a rejection level for the statistical test in Algorithms 1 and 2. Moreover, we use 40% of the training examples to fit the linear

models in Algorithms 1 and 2, and the remaining data as validation. The sensitivity to the choice of δ is discussed in Section 4.2.

Generative process of the data For each task $k \in \{1, 2, \dots, D, T\}$, we sample a set of causal variables from a multivariate Gaussian

$$\mathbf{X}_{S^*}^k \sim \mathcal{N}(0, \Sigma_{S^*}^k)$$

where the covariance matrix $\Sigma_{S^*}^k$ is drawn from a Wishart distribution $\mathcal{W}(U_{S^*}^k, p)$, where $U_{S^*}^k$ is computed as $V^k(V^k)^t$. Here, V^k is a $(|S|, |S|)$ matrix of standard Gaussian random variables.

The target variable Y^k is drawn as

$$Y^k = \alpha \mathbf{X}_{S^*}^k + \epsilon^k$$

where $\epsilon^k \sim \mathcal{N}(0, 2)$ (the standard deviation of ϵ^k is 6 for the non sparse DG experiment with 30 predictors, see the bottom of Figure 4).

We sample the remaining predictor variables as

$$\mathbf{X}_N^k = \gamma^k Y^k + \beta^k (\mathbf{X}_{S^*}^k)_C + \eta^k$$

where $\eta^k \sim \mathcal{N}(0, \Sigma_N^k)$. $(\mathbf{X}_{S^*}^k)_C$ is a subset of $\mathbf{X}_{S^*}^k$ of size $|C|$ which generates both the target Y^k and $\mathbf{X}_N T^k$. γ^k of size $|N|$ is computed as $\gamma^k = (1 - \lambda)\gamma_0 + \lambda g^k$, where $\lambda \in [0, 1]$, γ_0 is the same in all tasks while g^k is task dependent. Both γ_0 and g^k are drawn from a standard Gaussian. Similarly to γ^k , β^k is a $(|C|, |N|)$ matrix computed as $\beta^k = (1 - \lambda)\beta_0 + \lambda b^k$. Σ_N^k is sampled similarly to $\Sigma_{S^*}^k$. Finally, α is sampled from a standard Gaussian distribution.

The generative process and hyper-parameters are the same for all the experiments (DG and MTL).

Results Our goal is to linearly predict target Y^T using predictors $\mathbf{X}^T = (\mathbf{X}_{S^*}^T, \mathbf{X}_N^T)$ on the test task. Given regression coefficient β , we measure the performance in the test task using the logarithm of the empirical estimator of $\mathcal{E}_{\mathbb{P}^T}(\beta)$.

In Figure 4, we are in the DG setting (thus, no labeled examples from T are observed at training). 4000 examples per training task are available for the top left and right plots, while only 1000 examples per task are available on the bottom because of computational reasons. We report the log average empirical MSE over left out test tasks. We study both sparse and non sparse settings (in which full search is not feasible). On the upper left and upper right, we see that when more than four training tasks are available, both the full search and greedy approaches are able to recover an invariant set, and outperform pooling the data for any number of training tasks. When more than five training tasks are observed, $\beta^{CS(\hat{S})}$ performs like $\beta^{CS(cau)}$, which uses knowledge of the ground truth. On the bottom, full search is not feasible, and $\beta^{CS(\hat{S}_{greedy})}$ outperforms other approaches.

In Figure 5 (top left), we consider an AMTL setting, in which large amounts of labeled data (36000) from the training tasks and unlabeled data from the test task (50000) are available. Both S and N are of size 3, such that \mathbf{X} is 6-dimensional. For all MTL experiments, 6 training tasks are available. We report the percentage of simulations for which the population MSE of a given approach outperforms β^{dom} . We see that $\beta^{CS(cau+, i.d.)}$ systematically outperforms β^{dom} . Moreover, $\beta^{CS(cau+)}$ and $\beta^{CS(\hat{S}+)}$ also perform well, and positive

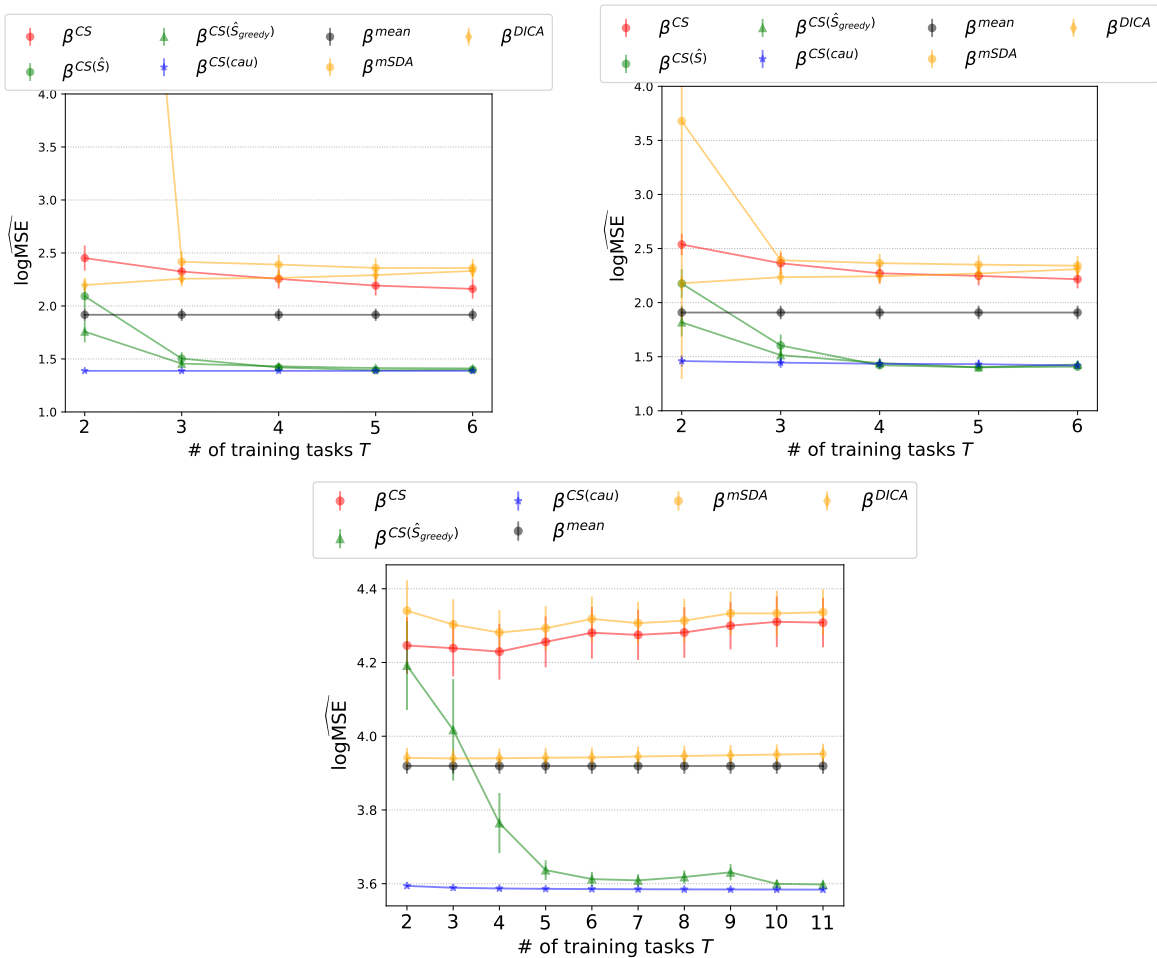


Figure 4: DG setting. Logarithm of the empirical squared error in the test task for the different estimators in the DG setting. The results show averages and 95% confidence intervals for the mean performance over 100 repetitions. We vary the number of tasks D available at training time. Upper left: both S and N are of size 3, such that \mathbf{X} is 6-dimensional. $|C|$ is of size one. Upper right: 30 noise variables are added to \mathbf{X} . Variable selection using the Lasso is used prior to computing $\beta^{CS(\hat{s})}$, while $\beta^{CS(\hat{s}_{greedy})}$ uses all predictors. Bottom: both S and N are of size 15. Full search is not computationally feasible in this setting and only the greedy procedure can be used. Other methods such as β^{CS} , β^{mSDA} and β^{DICA} often perform badly, which explains why in comparison β^{mean} appears to performs well.

transfer is effective. However, a prohibitively large amount of labeled and unlabeled data is needed for these approaches, and the differences become non-significant for all methods except $\beta^{CS(cau+,i.d.)}$. This shows the limitation of this family of approaches. In a setting with only 900 examples per training task in SMTL, we plot in Figure 6 the histogram of the error difference $\Delta = \mathcal{E}(\beta^{dom}) - \mathcal{E}(\beta)$ for $\beta^{CS(cau\#)}$. Figure 5 (top right) corresponds to the same setting, but we vary the number of unlabeled data available (we only plot methods

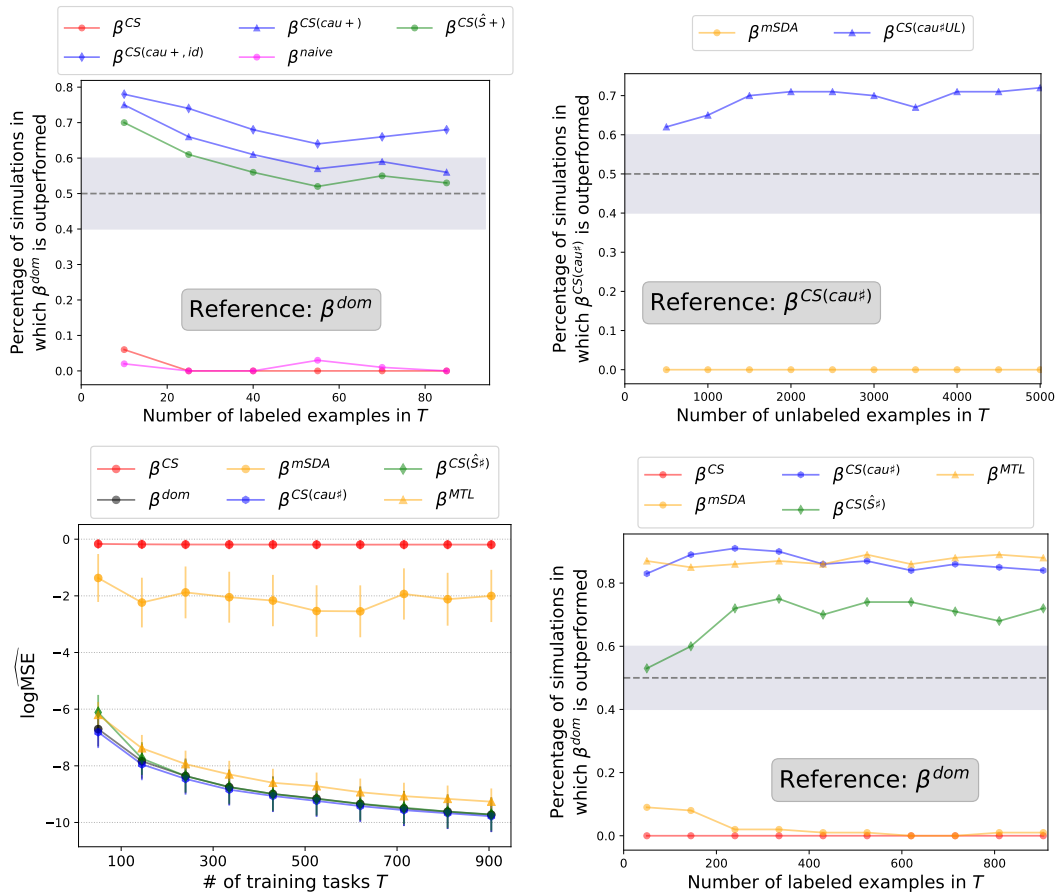


Figure 5: MTL setting. Percentage of repetitions (out of 100) for which the corresponding method outperforms β^{dom} (or $\beta^{CS(cau\#)}$ for the top right plot). Both S and N are of size 3, such that \mathbf{X} is 6-dimensional. Upper left: AMTL setting. This plot shows that the methods $\beta^{CS(\hat{S}+)}$ and $\beta^{CS(cau+)}$ presented in Section 2.2 perform well, but a large amount of data is necessary: 50000 unlabeled examples from T and 36000 training examples are available. The naive method β^{naive} performs poorly. Upper right: in the SMTL setting, we fix the number of training data (500 per task) and vary the amount of unlabeled data available from the test task. We report the percentage of scenarios in which the corresponding method outperforms $\beta^{CS(cau\#)}$ this time (which uses no unlabeled data). While β^{mDA} always performs worse than $\beta^{CS(cau\#)}$ and does not exploit the unlabeled data, we see that $\beta^{CS(cau\#,UL)}$ performs better as the amount of unlabeled data increases. Bottom: SMTL setting, and we vary the number of labeled examples available in each training task. Here, significantly less labeled data was available in the training tasks (from 50 to 1000 per task). In this setting, the methods using unlabeled data were given 100 unlabeled examples. Bottom left: logarithm of the empirical squared error in the test task for different estimators. Bottom right: percentage of repetitions (out of 100) for which the corresponding method outperforms β^{dom} .

that use unlabeled data, and $\beta^{CS(cau\#)}$ is used as reference instead of β^{dom}). In Figure 5 (bottom) we consider an SMTL setting in which only 100 unlabeled data points are avail-

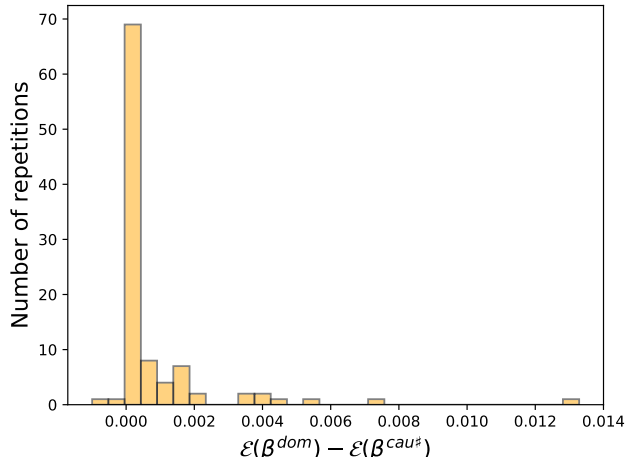


Figure 6: In the SMTL setting, 900 examples from each of the training tasks are available (this corresponds to the data point furthest to the right in the bottom plot of Figure 5). We run 100 repetitions and plot the histograms of $\Delta = \mathcal{E}(\beta^{dom}) - \mathcal{E}(\beta^{CS(cau\#)})$. The proposed estimator outperform β^{dom} : for a large proportion of the repetitions, $\Delta > 0$. More importantly, the distribution of Δ is heavily skewed in the positive values. In other words, when β^{dom} outperforms $\beta^{CS(cau\#)}$, the difference in performance is small, while the difference is often larger for the converse.

able, and only few labeled examples are available in each task. Here, we see that $\beta^{CS(cau\#)}$, $\beta^{CS(\hat{S}\#)}$ and β^{MTL} perform well, while other methods do not. In terms of MSE (bottom left), the difference in performance between the top competing methods is not statistically significant.

Time complexity The most expensive component of our method is the estimation of the invariant subset. In the DG experiment in Figure 4, with $n = 4000$ examples available for each of the 6 tasks, and $p = 6$ predictors, full subset search takes 0.067 seconds and greedy search 0.037, where the results are averaged over 100 repetitions. With $p = 10$, full search averages at 1.57 seconds, and greedy search 0.0396. With $p = 30$, where full search is not feasible, greedy search averages at 1.21 seconds. In the MTL experiment in Figure 5, the EM algorithm runs for 0.00105 seconds on average over 100 repetitions. As a reference, in MTL, linear regression averages at 0.000301 seconds and mSDA at 0.0547 seconds.

4.2 Sensitivity to the acceptance level δ

Both Algorithm 1 and its greedy version Algorithm 2 receive an acceptance level δ as input for the statistical test. In our other experiments, we chose the standard value of $\delta = 0.05$. Figure 8 shows the error on the test tasks in the DG setting for both methods for different values of δ . The setting is the same as in the left of Figure 4 for three training tasks. β^{CS} and $\beta^{CS(cau)}$ are provided as reference. For $\delta = 0$, all subsets are accepted as invariant,

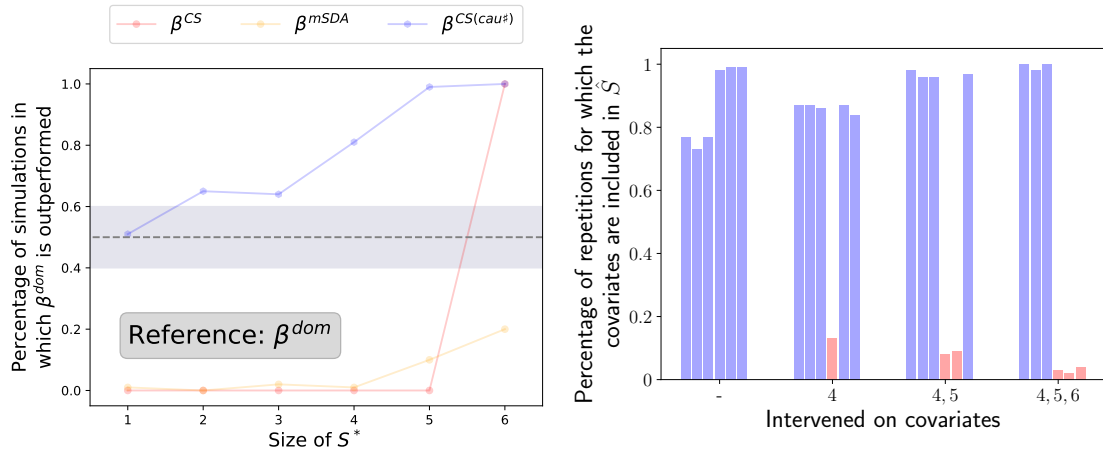


Figure 7: *Left*: SMTL setting with 6 tasks and 900 examples per task. We plot the percentage of repetitions (over 100) for which the given methods outperform β^{dom} , as a function of the size of the invariant set S^* . We see that as S^* becomes larger, more information is transferred from the training tasks, and as such the performance of $\beta^{CS(cau\#)}$ improves. When S^* is the full set, our method behaves like pooling the data. *Right*: Covariates selected by Algorithm 1 when the training tasks contain interventions only on some of the covariates. The bars represent the percentage of repetitions (out of 100) for which the corresponding covariates were selected. When there are no interventions in the training tasks, meaning that all the training tasks follow the same distribution, Algorithm 1 systematically selects *all* covariates for prediction. When more interventions are performed, however, the corresponding covariates (in red) are excluded in a large number of the repetitions.

thus both methods behave like pooling the data. After a critical value of δ , no subset is accepted, and both algorithms return the subset with the largest p-value.

4.3 Informativeness and subset estimation

The estimation of an invariant subset involves finding a subset for which the residuals have the same distribution across tasks. It is desirable, however, that the selected subset is one which explains the data best. This is ensured by selecting the subset which leads to the smallest error on a validation set. Therefore, some covariates in N may be included in a selected subset *if there are no interventions on this covariates in the training tasks*. More precisely, if including a covariate does not lead to a statistically measurable difference in the distribution of the residuals between the training tasks, it is advantageous in general to include it in the selected subset since the data is better explained.

We illustrate this in Figure 7 (right) in the setting previously described with $p = 6$. We estimate an invariant subset using Algorithm 1 over 100 repetitions in the following scenarios: i) all the covariates have the same distribution across tasks, ii) one, two or three covariates in N are subject to interventions between the tasks. Figure 7 (right) show the proportion of repetitions for which each covariate is included in the selected subset. We see that, as expected, covariates in N for which there are no interventions are included

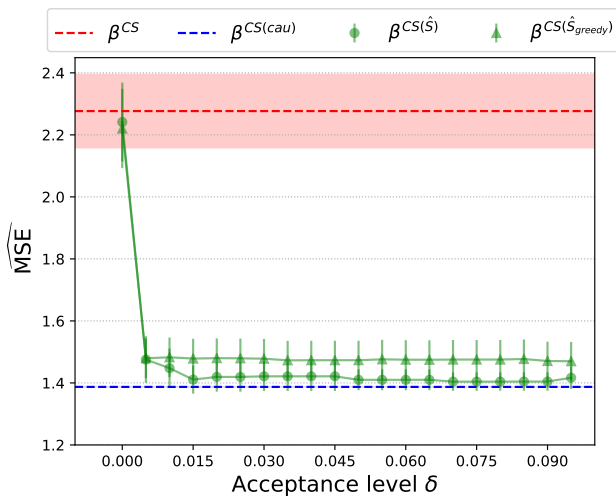


Figure 8: Logarithm of the empirical squared error in the test task in the DG setting as a function of the acceptance level of the statistical test δ in Algorithm 1. The setup corresponds to $t = 3$ in Figure 4 (left), also over 100 repetitions. For $\delta = 0$, all subsets are accepted, so the full set of predictors, which minimizes the validation squared error, is selected. Algorithm 1 then returns β^{CS} . As δ increases, no subset is accepted, and Algorithm 1 returns the subset with the largest p-value.

in the selected subset in a large portion of the repetitions, while the other covariates are excluded. This highlights that Algorithm 1 can only exclude covariates whose distribution shifts between training tasks. If being conservative is important for the problem at hand, one can modify Algorithm 1 accordingly, see the end of Section 3.4.

Moreover, in Figure 7 (left) we consider a similar setting, and we compute the performance against β^{dom} in an SMTL setting as the size of the invariant set increases. We see that as the size of the invariant set increases, the performance of $\beta^{CS(cau\#)}$ improves, since more information is being transferred from the training tasks. When $p = 6$, traditional covariate shift holds, and $\beta^{CS(cau\#)}$ performs on par with β^{pool} .

4.4 Gene perturbation experiment

We apply our method to gene perturbation data provided by Kemmeren et al. (2014). This data set consists of the m-RNA expression levels of $p = 6170$ genes X_1, \dots, X_p of the *Saccharomyces cerevisiae* (yeast). It contains both $n_{obs} = 160$ observational data points and $n_{int} = 1479$ data points from intervention experiments. In each of these interventions, one known gene (out of p genes) is deleted. In the following, we consider two different tasks. The observational sample is drawn from the first task, and the pooled n_{int} interventions are drawn from the second task.

Motivation In order to gain an intuition about the experiments we are presenting, consider Figure 9. We select as a target a gene Y out of the p genes, and our goal is to predict the activity of Y given the remaining $p - 1$ genes as features. Some of these $p - 1$ genes are

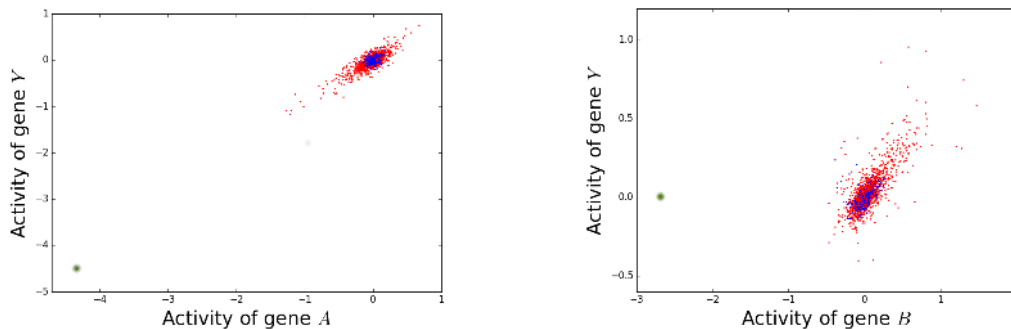


Figure 9: Example of the expression of pairs of genes, where A is causal (left) and B is non-causal (right) of target Y . The blue points are from the observational sample (task 1), the red dots are the interventional sample (task 2), and the green point corresponds to the single interventions in which A and B are intervened on respectively. On the left, a model learned on the data in red and blue would still perform well on the intervention point, which is not the case on the right.

causal of the activation of Y . For example, Figure 9 shows on the x-axis the activity of two genes (gene A on the left, gene B on the right) such that:

- The expressions of A and B are strongly correlated with the expression of Y .
- A is causal of Y (here, we use the definition of a causal effect proposed by Peters et al. (2016)).
- B is non-causal of Y (anticausal or confounded).

In Figure 9 (left), the blue points correspond to the 160 data points from the observational sample, which corresponds to the first task. The red dots are the 1478 data points from the interventional sample, except for the single data point for which A is intervened on, and constitute the second task. The plot on Figure 9 (right) is constructed analogously for B . We can indeed see that in the pooled sample from task 1 and 2, A and B are both strongly correlated with target Y .

The key difference between both plots are the green points. On Figure 9 (left), the green dot corresponds to the single intervention experiment in which gene A is intervened on. Similarly, the green dot on Figure 9 (right) is the single point in which B is intervened on. Our goal is to consider the DG setting in which the test task consists on this single intervention point.

For the causal gene A , one expects that a change in the activity of A should translate into a proportional change in the activity of Y . We observe that, in the particular example of the left plot, a linear regression model from A to Y trained only on the pooled data from tasks 1 and 2 (blue and red in Figure 9) would lead to a small prediction error on the intervened point (in green). That is, $S^* = \{A\}$ might be a good candidate for a set satisfying Assumptions (A1), (A1') and (A2). For the non-causal gene B , however, intervening on B leaves the activity of Y unchanged, and the linear model learned on the data from tasks 1 and 2 performs badly on the test point in green. In such case, a candidate set is the empty set $S^* = \{\}$, leading to prediction using the mean of the target in the training data. A model which is aiming to test in these challenging intervention points should therefore

include causal genes as features, but exclude non-causal genes. In these experiments, we aim at testing whether we can exclude non-causal genes such as B automatically.

Setup We address the problem of predicting the activity of a given gene from the remaining genes. We are looking at the following:

- We consider p different **problems**. In each problem $j \in \{1, \dots, p\}$, we aim at predicting the activity $Y = X_j$ of gene j using $(X_\ell)_{\ell \neq j}$ as features.
- In each problem $j \in \{1, \dots, p\}$, two **training tasks** $k \in \{1, 2\}$ are available. The data from the first task is the observational sample, and the data from the second task are all the n_{int} interventions (we shall subsequently remove some points for testing, see below).

The goal is now to apply our method to each of the problems and estimate an invariant subset. Due to the large number of predictors, we first select the 10 top predictor variables using the Lasso and then apply Algorithm 1 to select a set of invariant predictors \hat{S} , see $\beta^{\hat{S}Lasso}$ in Table 2. We denote the indices of the features selected using Lasso by $L = (L_1, \dots, L_{10})$.

The procedure is then evaluated as follows: for each problem $j \in \{1, \dots, p\}$, we first find the genes in $(X_{L_1}, \dots, X_{L_{10}})$ for which an interventional example is available. Note that this might not hold for all selected genes, since only $n_{int} < p$ interventions are available. We then iterate the following procedure (this is within the context of *the same problem*): for each gene in $(X_{L_1}, \dots, X_{L_{10}})$ for which an intervention is available,

- we put aside the example corresponding to this intervention from the training data (in the motivation example, this would correspond to the green point).
- we estimate an invariant subset $\hat{S} \subseteq L$ using Algorithm 1 with the remaining observational and interventional data.
- we test all methods on the single intervention point which was put aside.

We expect two different scenarios, as explained in the motivation paragraph above: (1) if the intervened gene is a *cause* of the target gene, it should still be a good predictor (see Section 2.3); then, it should be beneficial to have this gene included in the set of predictors \hat{S} . (2) if the intervened gene is anticausal or confounded (we refer to this scenario as *non-causal*), the statistical relation to the target gene might change dramatically after the intervention and therefore, one may not want to base the prediction on this gene. In order to see this effect and understand how the different approaches for DG in Table 2 handle the problem, we consider two groups of experiments.

- (1) we select the target genes Y for which one of the features in L is causal for the activity of Y and for which an intervention experiment is available. 39 problems fall in this causal scenario.
- (2) out of the remaining problems we chose target genes with (non-causal) predictors that have been intervened on and — in order to increase the difficulty of the problem — that are strongly correlated with the target gene. We therefore select 269 cases for which a Pearson correlation test (the null hypothesis corresponds to no correlation) outputs a p-value equal to zero.

Results Figure 10 shows box plots for the errors of the different methods for the causal problems (1) on the top left and for the non-causal problems (2) in the top right. We do not plot outliers in order to improve presentation. Figure 10 (top left) presents the causal

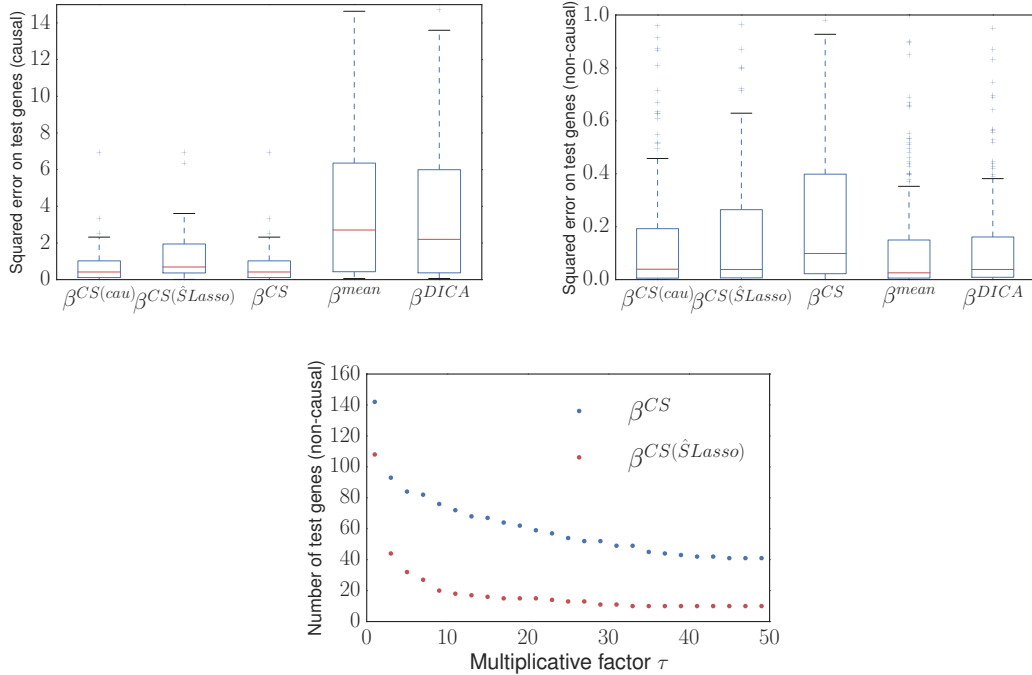


Figure 10: In the causal problems (top left), interventions are performed on causal genes. As expected, the input genes continue to be good predictors, and β^{CS} works well. In the non-causal problems (top right), one of the inputs is intervened upon and becomes a poor predictor, impairing the performance of β^{CS} . The mean predictor β^{mean} uses none of the predictors, and therefore works comparatively well in this scenario. Our proposed estimator $\beta^{CS(\hat{S})}$ provides reasonable estimates in both the causal and non-causal settings, while other methods only perform well in one of the scenarios. β^{DICA} performs similarly to β^{mean} in both scenarios, and is therefore outperformed by other methods in the causal problems (note that β^{DICA} uses all available features). Bottom: in the non-causal scenario (2), we plot the number of test genes for which the squared error for β^{CS} is larger than τ times the squared error for $\beta^{CS(\hat{S})}$, and vice-versa, where τ is plotted on the x-axis. This plot shows the number of genes for which one of the method does significantly worse than the other. By this measure, $\beta^{CS(\hat{S}Lasso)}$ outperforms β^{CS} for all values of τ .

scenario. As expected, pooling does well in this setting. Figure 10 (bottom) shows that in the non-causal problems (2), prediction using an invariant subset leads to less severe mistakes on test genes compared to pooling the tasks.

For comparison, since we know which predictors are being intervened on at test time, we included a method that makes use of causal knowledge: $\beta^{CS(cau)}$ uses all 10 predictors in the causal problems (1) and all but the intervened gene in the non-causal problems (2). In practice, this causal knowledge is often not available. We regard it as promising that the fully automated procedure $\beta^{CS(\hat{S}Lasso)}$ performs comparably to $\beta^{CS(cau)}$.

5. Conclusions and further directions

We propose a method for transfer learning that is motivated by causal modeling and exploits a set of invariant predictors. If the underlying causal structure is known and the tasks correspond to interventions on variables other than the target variable, the causal parents of the target variable constitute such a set of invariant predictors. We prove that predicting using an invariant set is optimal in an adversarial setting in DG. If the invariant structure is not known, we propose an algorithm that automatically detects an invariant subset, while also focusing on good prediction. In practice, we see that our algorithm successfully finds a set of predictors leading to invariant conditionals when enough training tasks are available. Our method can incorporate additional data from the test task (MTL) and yields good performance on synthetic data. Although an invariant set may not always exist, our experiment on real data indicates that exploiting invariance leads to methods which are robust against transfer.

As we saw in the DG and MTL experiments, $\beta^{\hat{S}}$ does not always perform as well as β^{cau} , which uses the ground truth. We believe that alternative methods for estimating the set \hat{S} may close this gap. Furthermore, extending our framework to nonlinearities seems straightforward and may prove to be useful in many applications. For instance, we provide a general, nonlinear version of Theorem 1 in Appendix A. Moreover, Algorithms 1 and 2 are presented in a linear setting. However, the extension to a nonlinear framework is straightforward. In particular, the linear regression can be replaced by a nonlinear regression method. We expect that there may be feature maps leading to invariant conditionals that are different from a subset.

We expect our method to be favorable in (adversarial-like) situations with strong differences between the tasks, such as the gene experiment in Section 4.4. We also evaluated our method on the School dataset (Bakker and Heskes, 2003), but found that we do not do better than pooling the data (we also do not do worse, the results are not shown). We believe this may be due to the fact that the difference between the tasks in this dataset are not too large.

We believe, finally, that the link to causal assumptions and the exploitation of causal structure may lend itself well to proving additional theoretical results on transfer learning.

Appendix A.

In this Appendix, we provide proofs for the theoretical results in the paper, as well as an extension of Theorem 1.

A.1 A nonlinear extension of Theorem 1

The extension of Theorem 1 to a nonlinear setting is straightforward. Given a subset S^* leading to invariant predictions, the proposed predictor is defined as the conditional expectation

$$f_{S^*} : \begin{array}{l} \mathbb{R}^P \rightarrow \mathbb{R} \\ \mathbf{x} \mapsto \mathbb{E}[Y^1 | \mathbf{X}_{S^*}^1 = \mathbf{x}_{S^*}]. \end{array} \quad (11)$$

The following theorem states that f_{S^*} is optimal over the set of continuous functions \mathcal{C}^0 in an adversarial setting.

Theorem 4 *Consider D tasks $(\mathbf{X}^1, Y^1) \sim \mathbb{P}^1, \dots, (\mathbf{X}^D, Y^D) \sim \mathbb{P}^D$ that satisfy Assumption (A1). Then the estimator f_{S^*} in (11) satisfies*

$$f_{S^*} \in \arg \min_{f \in \mathcal{C}^0} \sup_{\mathbb{P}^T \in \mathcal{P}} \mathbb{E}_{(\mathbf{X}^T, Y^T) \sim \mathbb{P}^T} (Y^T - f(\mathbf{X}^T))^2,$$

where \mathcal{P} contains all distributions over (\mathbf{X}^T, Y^T) that are absolutely continuous with respect to the same product measure μ and satisfy $Y^T | \mathbf{X}_{S^*}^T \stackrel{d}{=} Y^1 | \mathbf{X}_{S^*}^1$.

Proof

Consider a function f that is possibly different from f_{S^*} , see (11). For each distribution $\mathbb{Q} \in \mathcal{P}$, we will now construct a distribution $\mathbb{P} \in \mathcal{P}$ such that

$$\int (y - f(\mathbf{x}))^2 d\mathbb{P} \geq \int (y - f_{S^*}(\mathbf{x}))^2 d\mathbb{Q}.$$

In this proof, we assume that the probability distributions in \mathcal{P} are absolutely continuous with respect to Lebesgue measure. The extension to the case where they are absolutely continuous with respect to a same product measure μ is straightforward. Let us therefore assume that \mathbb{Q} has a density $(\mathbf{x}, y) \mapsto q(\mathbf{x}, y)$. Define \mathbb{P} to be the distribution that corresponds to $p(\mathbf{x}, y) := q(\mathbf{x}_{S^*}, y) \cdot q(\mathbf{x}_N)$, where \mathbf{x}_N contains all components of \mathbf{x} that are not in S^* . In the distribution \mathbb{P} , the random vector \mathbf{X}_N is independent of (\mathbf{X}_{S^*}, Y) . But then

$$\begin{aligned} & \int (y - f(\mathbf{x}))^2 d\mathbb{P} \\ &= \int_{\mathbf{x}_N} \int_{\mathbf{x}_{S^*}, y} (y - f(\mathbf{x}_{S^*}, \mathbf{x}))^2 p(\mathbf{x}_{S^*}, y) d\mathbf{x}_{S^*} dy p(\mathbf{x}_N) d\mathbf{x}_N \\ &\geq \int_{\mathbf{x}_N} \int_{\mathbf{x}_{S^*}, y} (y - f_{S^*}(\mathbf{x}_{S^*}))^2 p(\mathbf{x}_{S^*}, y) d\mathbf{x}_{S^*} dy p(\mathbf{x}_N) d\mathbf{x}_N \\ &= \int_{\mathbf{x}, y} (y - f_{S^*}(\mathbf{x}_{S^*}))^2 q(\mathbf{x}_{S^*}, \mathbf{x}_N, y) d\mathbf{x}_{S^*} dy d\mathbf{x}_N \\ &= \int (y - f_{S^*}(\mathbf{x}))^2 d\mathbb{Q}. \end{aligned}$$

■

A.2 Proof of Proposition 2

We consider three variables and the following generative process: $Y^k = \alpha^t \mathbf{X}_{S^*}^k + \epsilon^k$, $Z^k = \gamma^k Y^k + \eta^k$, where $\epsilon^k \sim \mathcal{N}(0, \sigma^2)$, $\eta^k \sim \mathcal{N}(0, \sigma_\eta^2)$ and $(\mathbf{X}_{S^*}^k)_j \sim \mathcal{N}(0, (\sigma_X)_j^2)$. In this model, γ^k is the parameter responsible for the difference between the tasks, while the other parameters are shared between the tasks.

At training time, D tasks are available. We first aim to obtain an explicit formula for the linear regression coefficients $\beta^{CS} = (\beta_{S^*}^{CS}, \beta_Z^{CS})$ obtained from pooling all the training tasks together. Denote by \mathbf{X} , Y and Z the pooled training data. For fixed $\gamma^1, \dots, \gamma^D$, the expected loss in the training data satisfies for coefficient β verifies:

$$\begin{aligned} \mathbb{E} \left((Y - (\beta_X)^t \mathbf{X} - \beta_Z Z)^2 \right) &= \frac{1}{D} \sum_{k=1}^D \mathbb{E} \left(Y^k - (\beta_X)^t \mathbf{X}^k - \beta_Z Z^k \right)^2 \\ &= \beta_X^t \text{diag}(\sigma_X^2) \beta_X + \frac{\beta_Z^2}{D} \left(\sigma_\eta^2 D + V_Y \bar{\gamma}^2 \right) + 2(\beta_Z \frac{\bar{\gamma}}{D} - 1) \alpha^t \text{diag}(\sigma_X^2) \beta_X + V_Y - 2 \frac{\bar{\gamma}}{D} V_Y \beta_Z, \end{aligned} \quad (12)$$

where $V_Y = \alpha^t \text{diag}(\sigma_X^2) \alpha + \epsilon^2$. By differentiating (12) with respect to β , we obtain the following expression for the pooled coefficients:

$$\beta_Z^{CS} = \frac{\bar{\gamma} \sigma^2}{V_Y^2 \bar{\gamma}^2 + D \sigma_\eta^2 - \frac{\bar{\gamma}^2}{D} \alpha^t \text{diag}(\sigma_X^2) \alpha} \quad \text{and} \quad \beta_{S^*}^{CS} = \left(1 - \frac{\bar{\gamma}}{D} \beta_Z^{CS} \right) \alpha,$$

where $\bar{\gamma}^2 = \sum_{k=1}^D (\gamma^k)^2$ and $\bar{\gamma} = \sum_{k=1}^D \gamma^k$. Consider now an unseen test task with coefficient γ^T . The expected loss on the test task using the pooled coefficients is:

$$\begin{aligned} \mathcal{E}_{\mathbb{P}^T}(\beta^{CS}) &= \mathbb{E} \left((Y^T - (\beta_X^{CS})^t \mathbf{X}^T - \beta_Z^{CS} Z^T)^2 \right) = (\beta_X^{CS})^t \text{diag}(\sigma_X^2) \beta_X^{CS} + (\beta_Z^{CS})^2 (V_Y (\gamma^T)^2 + \sigma_\eta^2) \\ &\quad + 2 \beta_Z^{CS} \gamma^T \alpha^t \text{diag}(\sigma_X^2) \beta_X^{CS} + V_Y \\ &\quad - 2 \alpha^t \text{diag}(\sigma_X^2) \beta_X^{CS} - 2 \beta_Z^{CS} V_Y \gamma^T. \end{aligned} \quad (13)$$

Therefore, the expectation with respect to γ^T is:

$$\mathbb{E}_{\gamma^T} \left(\mathcal{E}_{\mathbb{P}^T}(\beta^{CS}) \right) = (\beta_X^{CS})^t \text{diag}(\sigma_X^2) \beta_X^{CS} + (\beta_Z^{CS})^2 (V_Y \Sigma^2 + \sigma_\eta^2) + V_Y - 2 \alpha^t \text{diag}(\sigma_X^2) \beta_X^{CS}$$

Denote by $\mathcal{E}_{\mathbb{P}^T}(\beta^S) = \sigma^2$ the expected loss when using the invariant conditional predictor $\beta^{S^*} = (\alpha, 0)$. Then:

$$\begin{aligned} \mathbb{E}_{\gamma^T} \left(\mathcal{E}_{\mathbb{P}^T}(\beta^{CS}) \right) &\geq \mathbb{E}_{\gamma^T} \left(\mathcal{E}_{\mathbb{P}^T}(\beta^{S^*}) \right) \\ &\Leftrightarrow (\beta_X^{CS})^t \text{diag}(\sigma_X^2) (\beta_X^{CS}) + (\beta_Z^{CS})^2 (V_Y \Sigma^2 + \sigma_\eta^2) + V_Y - 2 \alpha^t \text{diag}(\sigma_X^2) \beta_X^{CS} \geq \sigma^2 \\ &\Leftrightarrow (\beta_Z^{CS})^2 (V_Y \Sigma^2 + \sigma_\eta^2) \geq 2 \alpha^t \text{diag}(\sigma_X^2) \beta_X^{CS} - (\beta_X^{CS})^t \text{diag}(\sigma_X^2) \beta_X^{CS} - \alpha^t \text{diag}(\sigma_X^2) \alpha \\ &\Leftrightarrow (\beta_Z^{CS})^2 (V_Y \Sigma^2 + \sigma_\eta^2) \geq - \frac{\bar{\gamma}^2}{D^2} (\beta_Z^{CS})^2 \alpha^t \text{diag}(\sigma_X^2) \alpha, \end{aligned} \quad (14)$$

by replacing $\beta_X^{CS} = \alpha - \alpha \frac{\bar{\gamma}}{D} \beta_Z^{CS}$. This inequality holds true for any value of the variance Σ^2 , and the pooled coefficient leads to larger error in expectation.

Consider now that the coefficients γ^k are fixed and centered around a non-zero value μ . Then the expectation with respect to γ^T of the loss in the test task is the following:

$$\begin{aligned} \mathbb{E}_{\gamma^T} (\mathcal{E}_{\mathbb{P}^T}(\beta^{CS})) &= (\beta_X^{CS})^t \text{diag}(\sigma_X^2) \beta_X^{CS} + (\beta_Z^{CS})^2 (V_Y(\Sigma^2 + \mu^2) + \sigma_\eta^2) \\ &\quad + 2\beta_Z^{CS} \alpha^t \text{diag}(\sigma_X^2) \beta_X^{CS} \mu + V_Y - 2\alpha^t \text{diag}(\sigma_X^2) \beta_X^{CS} - 2\beta_Z^{CS} V_Y \mu. \end{aligned} \quad (15)$$

Then, if $\bar{\gamma} \neq 0$ (if $\bar{\gamma} = 0$, both estimators coincide):

$$\mathbb{E}_{\gamma^T} (\mathcal{E}_{\mathbb{P}^T}(\beta^{CS})) \geq \mathbb{E}_{\gamma^T} (\mathcal{E}_{\mathbb{P}^T}(\beta^{S^*})) \Leftrightarrow \Sigma^2 \geq P(\mu), \quad (16)$$

where $P(\mu) = -\mu^2 - \frac{2}{\beta_Z^{CS}} \left((1 - \frac{\bar{\gamma}}{D} \beta_Z^{CS}) \frac{\alpha^t \text{diag}(\sigma_X^2) \alpha}{V_Y} - 1 \right) \mu - \frac{\bar{\gamma}^2}{V_Y D^2} \alpha^t \text{diag}(\sigma_X^2) \alpha + \frac{\sigma_\eta}{V_Y}$.

A.3 Proof of Proposition 3

Proof For $k \in \{1, \dots, D, T\}$, let \mathbb{Q}^k be the probability distribution with density:

$$q^k(\mathbf{x}_{S^*}, \mathbf{x}_N, y) := p^k(\mathbf{x}_{S^*}, y) p^T(\mathbf{x}_N | \mathbf{x}_{S^*}, y). \quad (17)$$

In the test task T , we trivially have $q^T = p^T$. First, it is easy to see that q^k and p^k have the same marginal distribution over \mathbf{x}_{S^*} and y . Indeed:

$$\begin{aligned} q^k(\mathbf{x}_{S^*}, y) &= \int_{\mathbb{R}^{|\mathcal{N}|}} q^k(\mathbf{x}_{S^*}, \mathbf{x}_N, y) d\mathbf{x}_N \\ &= \int_{\mathbb{R}^{|\mathcal{N}|}} p^k(\mathbf{x}_{S^*}, y) p^T(\mathbf{x}_N | \mathbf{x}_{S^*}, y) d\mathbf{x}_N \\ &= p^k(\mathbf{x}_{S^*}, y) \int_{\mathbb{R}^{|\mathcal{N}|}} p^T(\mathbf{x}_N | \mathbf{x}_{S^*}, y) d\mathbf{x}_N = p^k(\mathbf{x}_{S^*}, y). \end{aligned} \quad (18)$$

Second, we prove that the conditional $q^k(y | \mathbf{x}_{S^*}, \mathbf{x}_N)$ is the same in all tasks. Indeed, by applying Bayes' rule:

$$\begin{aligned} q^k(y | \mathbf{x}_{S^*}, \mathbf{x}_N) &= q^k(\mathbf{x}_N | y, \mathbf{x}_{S^*}) \frac{q^k(y, \mathbf{x}_{S^*})}{q^k(\mathbf{x}_{S^*}, \mathbf{x}_N)} \\ &= p^T(\mathbf{x}_N | y, \mathbf{x}_{S^*}) \frac{q^k(y | \mathbf{x}_{S^*})}{q^k(\mathbf{x}_N | \mathbf{x}_{S^*})} \\ &= p^T(\mathbf{x}_N | y, \mathbf{x}_{S^*}) \frac{p^k(y | \mathbf{x}_{S^*})}{\int_{\mathbb{R}} q^k(y, \mathbf{x}_N | \mathbf{x}_{S^*}) dy} \\ &= p^T(\mathbf{x}_N | y, \mathbf{x}_{S^*}) \frac{p^k(y | \mathbf{x}_{S^*})}{\int_{\mathbb{R}} q^k(\mathbf{x}_N | y, \mathbf{x}_{S^*}) q^k(y | \mathbf{x}_{S^*}) dy} \\ &= p^T(\mathbf{x}_N | y, \mathbf{x}_{S^*}) \frac{p^k(y | \mathbf{x}_{S^*})}{\int_{\mathbb{R}} p^T(\mathbf{x}_N | y, \mathbf{x}_{S^*}) p^k(y | \mathbf{x}_{S^*}) dy}. \end{aligned}$$

We have used the fact that $q^k(\mathbf{x}_N | y, \mathbf{x}_{S^*}) = p^T(\mathbf{x}_N | y, \mathbf{x}_{S^*})$, which follows from (18). Since the last equality leads to a term which is equal in all tasks (indeed, Assumption (A1) ensures that $p^k(y | \mathbf{x}_{S^*})$ is the same for all $k \in \{1, \dots, D, T\}$), we have the desired result. \blacksquare

A.4 Statement and proof of Proposition 5

In this Section, we provide an analytic expression for β^{opt} from (3) in terms of α and ϵ .

Proposition 5 *Assume that \mathbf{X}_{S^*} follows an arbitrary distribution and that Assumptions (A1) and (A2) hold. Let $\gamma \in \mathbb{R}^{|N|}$ be the solution of an L^2 regression from \mathbf{X}_N^T on Y^T . Therefore, we can write $\mathbf{X}_N^T = \gamma Y^T + \eta$, with η uncorrelated to Y^T , and the components of η can be correlated. Then the regression coefficients $\beta^{opt} = (\beta_{S^*}^{opt}, \beta_N^{opt})$ minimizing the expected squared loss in the test task satisfy*

$$\beta_N^{opt} = \mathbb{E}(\epsilon^2)M^{-1}\gamma, \quad (19)$$

$$\beta_{S^*}^{opt} = \alpha(1 - (\gamma^T)^t \beta_N^T) - \Sigma_{X,S^*}^{-1} \Sigma_{X,N} \beta_N, \quad (20)$$

where $M = \mathbb{E}(\epsilon^2)\gamma\gamma^t + \Sigma_N - \Sigma_{X,N}^t \Sigma_{X,S^*}^{-1} \Sigma_{X,N}$, and $\Sigma_N := \mathbb{E}(\eta\eta^t)$, $\Sigma_{X,S^*} := \mathbb{E}(\mathbf{X}_{S^*} \mathbf{X}_{S^*}^t)$, $\Sigma_{X,N} := \mathbb{E}(\mathbf{X}_{S^*} \eta^t)$ are the corresponding Gram matrices.⁴

Proof To simplify notation, we write Y^T , $\mathbf{X}_{S^*}^T$ and \mathbf{X}_N^T as Y , \mathbf{X}_{S^*} and \mathbf{X}_N . We compute the gradients of the expected squared loss after replacing the expression for Y and \mathbf{X}_{S^*} :

$$\begin{aligned} L &= \mathbb{E}(Y - \beta_{S^*}^t \mathbf{X}_{S^*} - \beta_N^t \mathbf{X}_N)^2 \\ &= (\alpha(1 - \gamma^t \beta_N) - \beta_{S^*})^t \Sigma_{X,S^*} (\alpha(1 - \gamma^t \beta_N) - \beta_{S^*}) \\ &\quad + (1 - \beta_N^t \gamma)^2 \mathbb{E}(\epsilon^2) + \beta_N^t \Sigma_N \beta_N - 2(\alpha(1 - \gamma^t \beta_N) - \beta_{S^*})^t \Sigma_{X,N} \beta_N \end{aligned}$$

The gradients satisfy

$$\begin{aligned} \frac{\partial L}{\partial \beta_{S^*}} &= -2\Sigma_{X,S^*} (\alpha(1 - \gamma^t \beta_N) - \beta_{S^*}) + 2\Sigma_{X,N} \beta_N \\ \frac{1}{2} \frac{\partial L}{\partial \beta_N} &= \Sigma_N \beta_N - (1 - \gamma^t \beta_N) \mathbb{E}(\epsilon^2) \gamma + \gamma \alpha^t \Sigma_{X,N} \beta_N \\ &\quad - \gamma \alpha^t \Sigma_{X,S^*} (\alpha(1 - \gamma^t \beta_N) - \beta_{S^*}) - \Sigma_{X,N}^t (\alpha(1 - \gamma^t \beta_N) - \beta_{S^*}) \end{aligned}$$

By setting these to zero, we find the stated values for $\beta_{S^*}^{opt}$ and β_N^{opt} . ■

Appendix B.

The code to reproduce the experiments in the paper can be found in https://github.com/mrojascaull/causal_transfer_learning.

4. We dropped the superscript T to lighten the notation.

References

- A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19 (NIPS)*, pages 41 – 48, 2007a.
- A. Argyriou, M. Pontil, Y. Ying, and C. Micchelli. A spectral regularization framework for multi-task structure learning. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 25 – 32, 2007b.
- B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4:83 – 99, 2003.
- J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149 – 198, 2000.
- S. Ben-David, T. Lu, T. Luu, and D. Pál. Impossibility theorems for domain adaptation. In *Proc. of the 13th Intern. Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 129 – 136, 2010.
- E. Bonilla, K. Chai, and C. Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 153 – 160, 2007.
- P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Series in Statistics. Springer, New York, NY, 2011.
- R. Caruana. Multitask learning. *Machine Learning*, 28:41 – 75, 1997.
- M. Chen, Z. Xu, K. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. In *Proc. of the 29th Intern. Conference on Machine Learning (ICML)*, pages 767 – 774, 2012.
- H. Daumé III, A. Kumar, and A. Saha. Frustratingly easy semi-supervised domain adaptation. In *Proc. of the 2010 Workshop on Domain Adaptation for NLP*, pages 53 – 59, 2010.
- T. Evgeniou and M. Pontil. Regularized multi-task learning. In *Proc. of the 10th Intern. Conference on Knowledge Discovery and Data Mining*, pages 109 – 117, 2004.
- K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 489–496, 2008.
- J. Gao, W. Fan, J. Jiang, and J. Han. Knowledge transfer via multiple model local structure mapping. In *Proc. of the 10th Intern. Conference on Knowledge Discovery and Data Mining*, pages 283 – 291, 2008.
- A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems 19 (NIPS)*, pages 585 – 592, 2007.

- A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3:131 – 160, 2009.
- C. Heinze-Deml, J. Peters, and N. Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 2018.
- K. D. Hoover. The logic of causal inference. *Economics and Philosophy*, 6:207 – 234, 1990.
- D. Janzing and B. Schölkopf. Causal inference using the algorithmic Markov condition. *IEEE Transactions on Information Theory*, 56(10):5168 – 5194, 2010.
- P. Kemmeren, K. Sameith, L. van de Pasch, J. Benschop, T. Lenstra, T. Margaritis, E. O’Duibhir, E. Apweiler, S. van Wageningen, C. Ko, et al. Large-scale genetic perturbations reveal regulatory networks and an abundance of gene-specific repressors. *Cell*, 157(3):740 – 752, 2014.
- A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1097 – 1105, 2012.
- N. Lawrence and J. Platt. Learning to learn with the informative vector machine. In *Proc. of the 21st Intern. Conference on Machine learning (ICML)*, pages 65–72, 2004.
- H. Levene. Robust tests for equality of variances. *Contributions to probability and statistics: Essays in honor of Harold Hotelling*, 2:278 – 292, 1960.
- R. Little and D. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, Inc., 1986. ISBN 0-471-80254-9.
- K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *Proc. of the 30th Intern. Conference on Machine Learning (ICML)*, pages 10 – 18, 2013.
- S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345 – 1359, 2010.
- J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge Univ. Press, USA, 2nd edition, 2009.
- J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (with discussion)*, 78(5):947 – 1012, 2016.
- J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.
- J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

- R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proc. of the 24th Intern. Conference on Machine Learning (ICML)*, pages 759 – 766, 2007.
- B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil. Exploiting unrelated tasks in multi-task learning. In *Proc. of the 15th Intern. Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 951 – 959, 2012.
- J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117, 2015.
- B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. M. Mooij. On causal and anticausal learning. In *Proc. of the 29th Intern. Conference on Machine Learning (ICML)*, pages 1255 – 1262, 2012.
- G. Schweikert, C. Widmer, B. Schölkopf, and G. Rätsch. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 1433 – 1440, 2009.
- H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227 – 244, 2000.
- M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 1433 – 1440, 2008.
- R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, 58:267 – 288, 1996.
- K. Zhang, J. Peters, D. Janzing, and B. Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *Proc. of the 27th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 804 – 813, 2011.
- K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In *Proc. of the 30th Intern. Conference on Machine Learning (ICML)*, pages 819 – 827, 2013.
- K. Zhang, M. Gong, and B. Schölkopf. Multi-source domain adaptation: A causal view. In *Proc. of the 29th AAAI Conference on Artificial Intelligence*, pages 3150 – 3157, 2015a.
- K. Zhang, J. Zhang, and B. Schölkopf. Distinguishing cause from effect based on exogeneity. In *Proc. of the 15th conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 261 – 271, 2015b.
- P. Zwiernik, C. Uhler, and D. Richards. Maximum likelihood estimation for linear gaussian covariance models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):1269 – 1292, 2017.