Invariant Policy Learning: A Causal Perspective

Sorawit Saengkyongam, Nikolaj Thams, Jonas Peters, and Niklas Pfister

Abstract—Contextual bandit and reinforcement learning algorithms have been successfully used in various interactive learning systems such as online advertising, recommender systems, and dynamic pricing. However, they have yet to be widely adopted in high-stakes application domains, such as healthcare. One reason may be that existing approaches assume that the underlying mechanisms are static in the sense that they do not change over different environments. In many real-world systems, however, the mechanisms are subject to shifts across environments which may invalidate the static environment assumption. In this paper, we take a step toward tackling the problem of environmental shifts considering the framework of offline contextual bandits. We view the environmental shift problem through the lens of causality and propose multi-environment contextual bandits that allow for changes in the underlying mechanisms. We adopt the concept of invariance from the causality literature and introduce the notion of policy invariance. We argue that policy invariance is only relevant if unobserved variables are present and show that, in that case, an optimal invariant policy is guaranteed to generalize across environments under suitable assumptions.

1 INTRODUCTION

The problem of learning decision-making policies lies at the heart of learning systems. To adopt these learning systems in high-stakes application domains such as personalized medicine or autonomous driving, it is crucial that the learned policies are reliable even in environments that have never been encountered before. In this paper, we consider the problem of learning policies that are robust with respect to shifts across environments. We consider this question in the setup of offline contextual bandits, which provides a mathematical framework for tackling the above learning problems.

While recent studies in offline contextual bandits [3], [11], [18], [30], [64], [65], [77] offer theoretical results and novel methodologies for policy learning from offline data, they primarily focus on a fixed-environment setting (from now on, we will refer to this as the identical distribution assumption) in which the underlying mechanisms do not change over time or over different environments. In practice, however, shifts between environments often occur, possibly invalidating the identical distribution assumption. In healthcare, for example, datasets from different hospitals may not come from the same underlying distribution. As a result, a learning agent that ignores environmental shifts may fail to generalize beyond the environments it was trained on.

In the supervised learning context, the environmental shift problem has been studied under different names, such as domain generalization, covariate shift adaptation, distributional robustness or out-of-distribution generalization [2], [13], [42], [63], [71]. In domain generalization, the goal is to develop learning algorithms that are robust to changes in the test distribution. Thus, a fundamental problem is how to characterize such changes. A promising direction relies on a causal framework to describe the changes through the concept of interventions [2], [13], [40], [54], [57]. A key insight is that while purely predictive methods perform best if test and training distributions coincide, causal models generalize to arbitrarily strong interventions on the covariates because of the modularity property of structural causal models (see e.g., [46]).

In real-world applications knowledge of the underlying causal graph and structural discrepancies between environments may not be available. In recent years, invariance-based methods have been exploited to learn the causal structure from data [25], [49], [51]. In invariant causal prediction [49], for example, one assumes that the data are collected from different environments, each of which describes different underlying data-generating mechanisms, and uses this heterogeneity to learn the causal parents of an outcome variable $Y$. The underpinning assumption is the invariance assumption, which posits the existence of a set of covariates $X$ in which the mechanism between $X$ and $Y$ remains constant. A model based on such invariant covariates is guaranteed to generalize to all unseen environments.

Our paper delineates an explicit connection among causality, invariance, and the environmental shift problem in the context of contextual bandits. We propose a multi-environment contextual bandit framework that represents mechanisms underlying a contextual bandit problem by structural causal models (SCMs; [46]). The framework allows for changes in environments and thereby relaxes the...
The goal of the agent is discussed for all the brief review related work and introduces 2A\text{ð} that maximizes the policy value over \( n \) \( p \)

As in the standard contextual bandit 3

We assume knowledge of the initial policy according to a policy to an agent. From the observed covariates \( X \) \( p \)

environmental shift problem in contextual bandits and, \( 2 \)

ods, however, focus mainly on providing efficient planning (MDP) under the framework of factored MDPs. Such methods, however, focus mainly on providing efficient planning algorithms rather than generalizing to a new environment [23], [24], [27], [33]. Although some recent studies have explored the use of causality and invariance for tackling the environmental shift problem in contextual bandits and, more generally, reinforcement learning [59], [75], the actual roles and benefits of causality and invariance remain unclear and under-explored.

Our framework differs from the framework of causal bandits [17], [36], [38], [72]. While causal bandits exploit causal knowledge (either assumed to be known a priori or estimated from data) for improving the finite sample performance in a single environment, our framework focuses on modeling distributional shifts and the ability to generalize to new environments. Another line of work has addressed the problem of policy evaluation and learning under unobserved confounding between the action and the reward variables [5], [32], [58], [66], [67]. In contrast, we consider the complementary problem of unobserved confounding between the covariates and the reward variables (see Section 3).

1.2 Offline Contextual Bandits

We briefly review the offline contextual bandit problem [9], [61], considering a setup in which some of the covariates (also known as context variables) are unobserved. More precisely, we assume that the covariates can be partitioned into observed and unobserved variables \( X \in \mathcal{X} \) \( U \in \mathcal{U} \). Here, \( \mathcal{X} \) and \( \mathcal{U} \) are metric spaces; the reader may think of \( \mathcal{X} \subseteq \mathbb{R}^d \) and \( \mathcal{U} \subseteq \mathbb{R}^p \). As in the standard contextual bandit setup [35], for each round, we assume that the system generates a covariate vector \( (X, U) \) and reveals only the observable \( X \) to an agent. From the observed covariates \( X \), the agent selects an action \( A \in A \) according to a policy \( \pi : \mathcal{X} \rightarrow \Delta(A) \) that maps the observed covariates to the probability simplex 3 over the set of actions 4. (In this work, we assume \( A \) to be finite). Adapting commonly used notation, we write, for all \( x \in \mathcal{X} \) and \( a \in A \), \( \pi(a|x) := \pi(x)(a) \). The agent then receives a reward \( R \) depending on the chosen action \( A \), and on both the observed and unobserved covariates \( (X, U) \).

In the classical setting, one assumes that the covariates are drawn i.i.d. from a joint distribution \( P_{X,U} \) (an assumption we will relax when introducing multi-environment contextual bandits in Section 2) and that the rewards are drawn from a conditional distribution \( P_{R|X,U,A} \). The agent is evaluated based on the performance of its policy \( \pi \) which is measured by the policy value:

\[
V(\pi) := \mathbb{E}_{(X,U) \sim P_{X,U}} \mathbb{E}_{A \sim \pi(X)} \mathbb{E}_{R \sim P_{R|X,U,A}} [R].
\]

The agent is now given a fixed training dataset that is collected offline: it consists of \( n \) rounds from one or more different policies, i.e., \( D := \{(X_i, A_i, R_i, \pi(X_i))\}_{i=1}^n \), where \( A_i \sim \pi_i(X_i) \) for all \( i \in \{1, \ldots, n\} \). The goal of the agent is then to find a policy \( \pi \) that maximizes the policy value over a given policy class \( \Pi \), i.e., \( \pi^* = \arg \max_{\pi \in \Pi} V(\pi) \).

As mentioned, this setting assumes that the environment in which we deploy the agent is identical to the environment in which the training dataset was collected. Section 2 introduces a causal framework for multi-environment contextual bandits, a framework that relaxes the identical distribution assumption.

1. We assume knowledge of the initial policy \( \pi \), to ease our presentation and focus our contribution on the environmental shifts problem. Our theoretical results and algorithms remain unchanged even if the initial policy is unknown and needs to be estimated from the offline data (see Appendix F, available online for more details).
2 A CAUSAL FRAMEWORK FOR MULTI-ENVIRONMENT CONTEXTUAL BANDITS

Instead of having a fixed distribution \( P_{X,U} \) over the covariates, we introduce a collection \( \mathcal{E} \) of environments such that, in each round, the covariates are drawn from an environment-specific distribution \( P_{X,U}^e \) that depends on the environment \( e \in \mathcal{E} \) in that round.

In practice, the agent only observes part of the environments \( \mathcal{E}^{\pi^e} \subseteq \mathcal{E} \) and is expected to generalize well to all environments in \( \mathcal{E} \) including the unseen environments \( \mathcal{E} \setminus \mathcal{E}^{\pi^e} \). To formalize the problem, we first introduce a model that puts assumptions on how environments change the distributions of \( X, U \) and \( R \). Specifically, an environment \( e \) can only perturb the distribution of the reward \( R \) through altering the distribution of the covariates \( X \) and \( U \). This constraint makes it possible to generalize information learned from one set of environments to another. In this formulation – even though the full conditional distribution of the reward \( P_{R|X,U,A}^{\pi^e} \) is assumed to be fixed across environments – the observable distribution \( P_{R|X,A}^{\pi^e} \) after marginalizing out the unobserved \( U \) may change from one environment to another (see, e.g., Fig. 1b).

2.1 Setting and Notation

Formally, the assumptions are constructed via an underlying class of SCMs indexed by the environment and policy.\(^2\)

**Setting 1 (Multi-environment (acyclic) SCMs for bandits).** Let \( \mathcal{X} = \mathcal{X}^X \times \mathcal{X}^U \) and \( \mathcal{U} = \mathcal{U}^X \times \mathcal{U}^A \) be products of metric spaces, \( \mathcal{A} = \{a_1, \ldots, a_k\} \) a discrete action space, \( \Pi := \{\mathcal{X}, \Delta(A)\} \) the set of all policies, and \( \mathcal{E} \) a collection of environments. For all \( \pi \in \Pi \) and all \( e \in \mathcal{E} \) we consider the following SCMs.

2. Readers familiar with the standard notion of SCMs may think about an SCM with a source node \( E \). \( S(\pi, e) \) then corresponds to an intervention on the action variable (change of policy) and on some of the observed covariates variables (change of environment). Here, we consider fixed environments, so that we do not have to consider them as random draws from an underlying distribution; see also [16].

\[ S(\pi, e) : \]
\[ \begin{align*}
U &:= s_c(X, U, e_U) \\
X &:= h_e(X, U, e_X) \\
A &:= g_\pi(X, e_A) \\
R &:= f(X, U, A, e_R),
\end{align*} \] (1)

where \((X, U, A, R) \in \mathcal{X} \times \mathcal{U} \times \mathcal{A} \times \mathcal{R}\) such that for all \( \pi \in \Pi \) and all \( e \in \mathcal{E} \) the SCM \( S(\pi, e) \) induces a unique distribution \( P_{X,U,A}^{\pi,e} \) over \((X, U, A, R)\) (see [10] for details) which is dominated by \( \mu \) and marginally has full support on \( \mathcal{X} \). We denote the corresponding density by \( P_{X,U,A}^{\pi,e} \) and the corresponding expectations by \( E_{\pi,e} \). Whenever a probability, density, or expectation does not depend on \( \pi \), we omit \( \pi \) and write \( E_{\pi,e} \) rather than \( E_{\pi,e}^{\pi,e} \), for example.

Some remarks regarding Setting 1 are provided below.

**Remark 1.** (1) We only use the SCMs as a flexible way of modeling the changes in the joint distribution with respect to the environment \( e \) and the policy \( \pi \). In particular, we do not use it to model any further intervention distributions that do not correspond to a change of policy or environment. (2) In practice, the precise form of the SCMs is unknown. Indeed, we will neither assume knowledge of the structural equations nor complete knowledge of the graph structure, except that the constraints induced by (1) hold. (3) The assumption of a dominating measure for all environments ensures that we can always assume the
We are now ready to define contextual bandits with multiple environments in the subsequent section.

2.2 Multi-Environment Contextual Bandits

Definition 1 (Multi-environment Contextual Bandits). Assume Setting 1. In a multi-environment contextual bandit setup, before the beginning of each round, the system is in an environment \( e \in \mathcal{E} \). Then, the system generates a covariate vector \((X, U)\) and reveals only the observable \( X \) and the environment label \( e \) to the agent. Based on the observed covariates \( X, e \), the agent selects an action \( A \) according to the policy \( \pi: \mathcal{X} \rightarrow \Delta(A) \). The agent then receives a reward \( R \), depending on the chosen action \( A \) and on both the observed and unobserved covariates \((X, U)\). More precisely, we assume for all \( i \in \{1, \ldots, n\} \) that \((X_i, U_i, A_i, R_i)\) are sampled independently according to \( \mathbb{P}_{X_i\mid U_i\mid A_i\mid R_i} \) (see Setting 1). The training data contains data from environments in \( \mathcal{E}^\text{tr} \). When \( \mathcal{E}^\text{tr} = \mathcal{E} = 1 \), the setup reduces to a standard contextual bandit setup.

In the multi-environment contextual bandit setup, the covariates on different rounds are not identically distributed due to changes in the environments. We can thus use this framework to model situations, where the test environments differ from training environments. We illustrate this setting with the following example, which we will refer back to several times throughout the paper.

Example 1. Consider a linear multi-environment contextual bandit with the following underlying SCMs

\[
\begin{align*}
S(\pi, e) : & \\
U := & e_U \\
X^1 := & \gamma_U X + \epsilon_{X^1} \\
X^2 := & \alpha_X + \epsilon_{X^2} \\
A := & g_A(X^1, X^2, \epsilon_A) \\
R := & \begin{cases} 
\beta_1 X^2 + U + \epsilon_R & \text{if } A = 0 \\
\beta_2 X^2 - U + \epsilon_R & \text{if } A = 1,
\end{cases}
\end{align*}
\]

where \( \epsilon_U, \epsilon_{X^1}, \epsilon_{X^2}, \epsilon_R, \epsilon_A, \epsilon_{X^1}, \epsilon_{X^2} \) are jointly independent noise variables with zero mean, \( \gamma_U, \alpha_X \in \mathcal{E} \) for all \( e \in \mathcal{E} \), \( \beta_1, \beta_2 \in \mathcal{R} \), and \( A = \{0, 1\} \). Fig. 1b depicts the induced graph \( \mathcal{G} \). In this example, the environments influence the observed covariates in two ways: (a) they change the mean of \( X^2 \) via \( \alpha_X \) and (b) they change the conditional mean of \( X^1 \) given \( U \) via \( \gamma_U \), while the rest of the components remain fixed across different environments. Here, the environment-specific coefficient \( \gamma_U \) modifies the correlation between the observable \( X^1 \) and the unobserved variable \( U \), and consequently between \( X^1 \) and the reward \( R \). Thus, an agent that uses information from \( X^1 \) to predict the reward \( R \) in the training environments may fail to generalize to other environments. To see this, consider a training environment \( e = 1 \) and a test environment \( e = 2 \) and let \( \gamma_1 = 1, \gamma_2 = -1 \) be the environment-specific coefficients in the training and test environments, respectively. In the training environment, we have a large positive correlation between \( X^1 \) and \( U \), and consequently the agent will learn that the action \( A = 0 \) yields a higher expected reward when observing a positive value of \( X^1 \) (and \( A = 1 \) otherwise). However, the correlation between \( X^1 \) and \( U \) becomes negative (and large in absolute value) in the test environment, which means
that the policy that the agent learned from the training environment will now be harmful. We will see in Section 3 that a policy that depends on a \( d \)-invariant set \((X^S)\) in this example) does not suffer from this generalization problem and is guaranteed to generalize across different environments.

A similar structure appears in the medical example discussed in Section 6. There, \( A \) is the dose of a drug, \( R \) is a response variable, \( X \) are observed patient features and \( U \) are unobserved genetic factors. The environment \( e \) is (a proxy of) the continent on which the data was collected.

### 2.3 Distributionally Robust Policies

To evaluate the performance of an agent across different environments, we define a policy value that takes into account environments. In particular, we focus on the worst-case performance of an agent over all environments.

**Definition 2 (Robust Policy Value).** For a fixed policy \( \pi \in \Pi \), and a set of environments \( \mathcal{E} \), we define the robust policy value \( V^\mathcal{E}(\pi) \in \mathbb{R} \) as the worst-case expected reward

\[
V^\mathcal{E}(\pi) := \inf_{e \in \mathcal{E}} E^\pi_{X|A,R}[R].
\]  

Intuitively, an agent that maximizes the robust policy value is expected to perform well (relative to other agents) in the most harmful environment. The idea of optimizing worst-case performance has been suggested in the reinforcement learning literature [1], [20] to ensure safe behavior of an agent and prevent catastrophic events and has also been used to formulate adversarial training [4] as well as out-of-distribution generalization [73].

We now assume that, for several training environments, we are given independent samples from a multi-environment contextual bandit, see Definition 1. More precisely, we assume to observe \( D := \{(X_i, A_i, R_i, \pi(X_i), e_i)\}_{i=1}^n \), where \( e_i \in \mathcal{E}^\pi, A_i \sim \pi(X_i), (X_i, A_i, R_i) \overset{\text{i.i.d.}}{\sim} P_{X|A,R}^e \) for all \( i \in \{1, \ldots, n\} \). Using only \( D \), we aim to solve the following maximin problem:

\[
\arg \max_{\pi \in \Pi} V^\mathcal{E}(\pi).
\]  

If we do not observe all environments, solving the maximin problem (3) is impossible without further assumptions. A baseline approach to this problem is to pool the data from all training environments and learn a policy that maximizes the policy value ignoring the environment structure. We show in Appendix B, available in the online supplemental material, that this is indeed optimal if the observed covariates explain all of the environment-based distributional shifts in \( R_i \), e.g., if all relevant covariates have been observed. However, if for example, hidden variables are present, the pooling approach does not necessarily yield an optimal policy and the learned policy may fail to generalize to unseen test environments.

### 3 Invariant Policies for Distributional Robustness

We now consider the general case in which the environment shifts may not be explained by the observed covariates. In what follows, we introduce \( d \)-invariant sets and policies, and show that, under Setting 1, the maximin problem (3) can be reduced to finding an optimal \( d \)-invariant policy given certain assumptions, see Proposition 1 and Theorem 1. This becomes particularly relevant if important variables remain unobserved. If all variables are observed, it suffices to pool the training environments.

**Remark 2.** If there are no hidden variables, one can solve the objective (3) by a standard policy optimization using all covariates \( X \), without taking into account further concepts such as invariance or causality. This statement is made precise and proved as Proposition 5 in Appendix B, available in the online supplemental material.

Nevertheless, in more realistic cases (see e.g., Figs. 1b and 1c), \( d \)-invariant sets and policies (introduced below) play a central role in solving the distributionally robust objective (3).

**Definition 3 (\( d \)-invariant Sets).** A subset \( S \subseteq \{1, \ldots, d\} \) is said to be \( d \)-invariant if the following \( d \)-separation statement holds:

\[
R \indep_{\mathcal{G}^S} e \mid X^S.
\]  

Our approach relies on the existence of a \( d \)-invariant set. We therefore make this assumption explicit.

**Assumption 1.** There exists a subset \( S \subseteq \{1, \ldots, d\} \) such that \( S \) is \( d \)-invariant.

Under faithfulness [46], Assumption 1 is testable from the observed data (see Section 4.1). Next, we define \( d \)-invariant policies. For all subsets \( S \subseteq \{1, \ldots, d\} \), let us denote the set of all policies that depend only on \( X^S \) by \( \Pi^S := \{\pi \in \Pi \mid \exists \pi^S : X^S \rightarrow \Delta(A) \text{ s.t. } \forall x \in \mathcal{X} \cdot \pi^S(x^S) = \pi^S(x^S) \} \subseteq \Pi \).

**Definition 4 (\( d \)-invariant Policies).** A policy \( \pi \) is said to be \( d \)-invariant with respect to a subset \( S \subseteq \{1, \ldots, d\} \) if \( \pi \) is a \( d \)-invariant set and \( \pi \in \Pi^S \).

We denote by \( S_{\text{inv}} := \{S \subseteq \{1, \ldots, d\} \mid S \text{ is } d\text{-invariant}\} \) the collection of all \( d \)-invariant sets and \( \Pi_{\text{inv}} := \{\pi \in \Pi \mid S_{\text{inv}} \cap S \text{ s.t. } \pi \text{ is } d\text{-invariant w.r.t. } S\} \) the collection of \( d \)-invariant policies. For now, we assume to have access to the set of \( d \)-invariant policies \( \Pi_{\text{inv}} \). Section 4 discusses when and how we can learn \( \Pi_{\text{inv}} \) from the observed data.

Because of the hidden variables \( U \), the conditional mean \( \mathbb{E}^\pi_{X|A}[R \mid X = x] \) is not ensured to be stable across the environments. Nevertheless, a \( d \)-invariant policy ensures that parts of the conditional mean are unchanged across environments.

4. The notion of \( d \)-invariant sets is related to \( S \)-admissibility in [48]. We use the term ‘\( d \)-invariant’ to emphasize that the definition is based on the \( d \)-separation statement (5) and involves the unseen environments. In related contexts, sometimes the term ‘generalizing’ is used [52]. Section 4 introduces the invariance hypothesis (11) that is testable from the observed data and discusses the assumptions required to connect the two conditions.
Lemma 1. Let $S \in S_{\text{inv}}$ be a $d$-invariant set and $\pi \in \Pi_S$. It holds for all $e, f \in E$ and $x \in X$ that

$$E^{\pi_e} [R | X^S = x] = E^{\pi_f} [R | X^S = x]. \quad (5)$$

Proof. See Appendix D.3, available in the online supplemental material.

For $S \in S_{\text{inv}},$ Lemma 1 implies that if a policy $\pi \in \Pi^S$ is optimal among $\Pi^S$ in the training environments, then $\pi$ is also optimal among $\Pi^S$ in all environments (Proposition 1). With the following assumption, we show in Proposition 1 that the same holds when replacing $\Pi^S$ by $\Pi_{\text{inv}}$.

Assumption 2. Let $G$ be the graph of the SCMs in Setting 1. Then, for all $\ell \in \{1, \ldots, p\}$, $G$ must contain an edge from $U^{\ell}$ to $R$.

Proposition 1. Assume Setting 1 and Assumption 1. Then the following statements hold.

i) Let $S \in S_{\text{inv}}$ and $\pi^S_{\text{opt}} \in \arg\max_{\pi \in \Pi^S} \sum_{e \in E^{\prime}} E^{\pi_e} [R]$. We then have

$$\forall \pi \in \Pi^S : \quad V^E (\pi) \leq V^E (\pi^S_{\text{opt}}). \quad (6)$$

ii) Let $\pi^* \in \arg\max_{\pi \in \Pi_{\text{inv}}} \sum_{e \in E^{\prime}} E^{\pi_e} [R]$. If Assumption 2 holds, we have

$$\forall \pi \in \Pi_{\text{inv}} : \quad V^E (\pi) \leq V^E (\pi^*). \quad (7)$$

Proof. See Appendix D.5, available in the online supplemental material.

The key argument in the proof of Proposition 1 is the identifiability of the optimal $d$-invariant set. Assumption 2 is necessary for this identifiability: if the assumption is violated and there are multiple $d$-invariant sets, one can, in general, not say which of those $d$-invariant sets is optimal with respect to all environments $E$ (see Appendix L, available in the online supplemental material, for a more detailed discussion). While, without Assumption 2, the $d$-invariant set that is most predictive on $E^{\pi^*}$ is no longer guaranteed to be worst-case optimal, it still satisfies a weaker guarantee shown in Theorem 1 below.

Proposition 1 shows that a $d$-invariant policy that is optimal under the training environments outperforms all other $d$-invariant policies, even on the test environments. But what about other policies that are not $d$-invariant? We will see in Theorem 1 that under certain assumptions on the set $E$ of environments, they cannot perform better than the above $\pi^*$ either.

We now outline the assumptions on the set $E$ of environments facilitating this result. As seen in Example 1, the crucial difference between a $d$-invariant policy $\pi^E$ (a policy that only depends on $X^d$) and a non-$d$-invariant policy $\pi^{(1:2)}$ (a policy that depends on both $X^1$ and $X^2$) is that $\pi^E$ can use information related to variables confounded with the reward ($X^1$ in this example) that may change across environments. In cases where the environments do not change the system ‘too strongly’ it can therefore happen that using such information is beneficial across all environments. In practice, however, one might not know how strong the test environments can change the system in which case such information can become useless or even harmful. Intuitively, this happens, for example, if environments exist where the non-$d$-invariant variables no longer contain any information about the reward. Formally, we make the following definition.

Definition 5 (Confounding Removing Environments). For $j \in \{1, \ldots, d\}$, we say that the variable $X^j$ is strongly non-$d$-invariant if for all $S \subseteq \{1, \ldots, d\}$

$$R \not\perp_{G^S} e \mid X^{S \setminus \{j\}}. \quad (8)$$

An environment $e \in E$ is said to be a confounding removing environment if for all $\pi \in \Pi$ it holds that

$$X^j \perp_{G^e} U,$$

for all strongly non-$d$-invariant variables $X^j$, where $G^e$ is the graph induced by the SCM $S(\pi, e)$.

The two $d$-separation statements in Definition 5 are in different graphs: Both graphs $G^S$ and $G^e$ are subgraphs of $G$. The distinction that is important for this definition is that while $G^S$ contains all edges between the covariates $(X, U)$ that appear in at least one environment, the graph $G^e$ only contains the edges that are active in the environment $e \in E$. Furthermore, to provide more understanding of the strongly non-$d$-invariant variables, we characterize a graphical criterion for such variables in Appendix D.4, available in the online supplemental material. There we show that the strongly non-$d$-invariant variables are the variables that are directly affected by $e$ and are confounded with $R$ through $U$, and descendants of such variables. These strongly non-$d$-invariant variables should not be included if one wants to find $d$-invariant sets. For example in Fig. 1c, the variable $X^1$ is strongly non-$d$-invariant and the $d$-invariant sets $\{X^2, X^3\}$ and $\{X^2, X^3\}$ are the sets that do not contain $X^1$.

Next, we give an example of a confounding removing environment, consider the graph $G^e$ in Example 1 (see Fig. 1b). For any subset $S$ where $\{1\} \subseteq S$ the path $e \rightarrow X^1 \rightarrow U \rightarrow R$ is open, and therefore $X^1$ is strongly non-$d$-invariant. A confounding removing environment is an environment that removes the incoming edge from $U$ to $X^1$. In such an environment, the variable $X^1$ does not contain any information about the reward $R$. A similar notion of confounding removing environments is used in [13] in the setting of prediction.

The existence of confounding removing environments implies that at least in some of the environments it is impossible to benefit from a non-$d$-invariant policy. To ensure that one cannot benefit in the worst-case, we therefore introduce the following assumption.

Assumption 3. For all $e \in E$, there exists $f \in E$ such that $f$ is a confounding removing environment and it holds that $P^e_X = P_f^X$.

To give an example, let $I \subseteq \{1, \ldots, d\}$ index the variables $X^I$ for which there is an edge from $e$ to $X^I$ in the graph $G$. If the set $E$ of environments consists of arbitrary interventions on $X^I$, then Assumption 3 is satisfied.
Theorem 1. Assume Setting 1 and Assumption 1. Let \( \pi^* \) be an optimal \( d \)-invariant policy that maximizes the pooled policy value under the training environments, \( \pi^* \in \arg \max_{\pi \in \Pi_{\text{inv}}} \sum_{e \in \mathcal{E}^*} \mathbb{E}^{\pi^*}[R] \). We then have the following statements.

i) We have for all \( e \in \mathcal{E} \) that
\[
\max_{a \in \mathcal{A}} \mathbb{E}^{\pi_a, \pi}_e[R] \leq \mathbb{E}^{\pi^*}_e[R],
\]
where \( \pi_a \) is the policy that always chooses action \( a \in \mathcal{A} \).

ii) If Assumptions 2 and 3 hold, we have
\[
\forall \pi \in \Pi : \quad V^{\mathcal{E}^*}(\pi) \leq V^{\mathcal{E}^*}(\pi^*). \tag{10}
\]

Proof. See Appendix D.6, available in the online supplemental material.

Theorem 1 implies that if we consider a policy class containing only the \( d \)-invariant policies, the maximin problem reduces to a standard policy optimization problem. Theorem 1 shows that an optimal \( d \)-invariant policy, under Assumption 3, is a solution to the distributionally robust objective. In other words, given a training dataset \( D \), we seek to operationalize the following two steps: (a) find the set \( \Pi_{\text{inv}} \) of all \( d \)-invariant policies (Section 4.1 discusses under which assumptions this is possible), (b) use offline policy optimization to solve \( \arg \max_{\pi \in \Pi_{\text{inv}}} V^{\mathcal{E}^*}(\pi) \) on the dataset \( D \).

One of the key components of the proposed method is to test whether a policy \( \pi \), which may be different from the policy generating the data, is \( d \)-invariant using data obtained from the training environments \( \mathcal{E}^* \). The following section proposes such a test, discusses the assumptions required to learn the set of \( d \)-invariant policies, and gives a detailed description of the whole procedure.

4 LEARNING AN OPTIMAL INVARIANT POLICY

4.1 Learning Invariant Sets

Our theoretical results (Proposition 1 and Theorem 1) in the previous section assume that the set of all \( d \)-invariant policies \( \Pi_{\text{inv}} \) is given. We now turn to the task of learning \( \Pi_{\text{inv}} \) which boils down to searching for the collection of all \( d \)-invariant sets \( S_{\text{inv}} \) using data obtained from the training environments \( \mathcal{E}^* \). To this end, we first define, for all \( S \subseteq \{1, \ldots, d\}, \pi \in \Pi \) and \( \mathcal{E}' \subseteq \mathcal{E} \), the null hypothesis
\[
H_0(S, \pi, \mathcal{E}') : \mathbb{P}^{\pi, \mathcal{E}_S}_R | X^S \text{ is the same for all } e \in \mathcal{E}'. \tag{11}
\]

In the case \( \mathcal{E}' = \mathcal{E}^* \), we refer to \( H_0(S, \pi, \mathcal{E}^*) \) as \( \mathcal{E}^* \)-invariance (which does not consider the unseen environments). Furthermore, we call a set \( S \) invariant if there exists \( \pi \in \Pi^{S} \) such that \( H_0(S, \pi, \mathcal{E}^*) \) holds and a policy \( \pi \) invariant with respect to \( S \) if \( \pi \in \Pi^{S} \) and \( S \) is invariant. We now state our core assumptions that make learning possible.

Assumption 4. For all \( S \subseteq \{1, \ldots, d\} \), the following holds:

i) \( \exists \pi \in \Pi^{S} : H_0(S, \pi, \mathcal{E}) \Rightarrow R \cup_{S^{G}} e \mid X^S \)

ii) \( \forall \pi \in \Pi^{S} : H_0(S, \pi, \mathcal{E}^*) \Rightarrow H_0(S, \pi, \mathcal{E}^*) \) true

Assumption 4 connects the conditional distribution invariance used in the null hypothesis (11) to the \( d \)-invariance condition given in (4) (The reversed implication follows by Lemma 3, Appendix D.3, available in the online supplemental material). This assumption is a special case of the faithfulness assumption [46] which is a fundamental assumption in causal discovery methods (e.g., [21]) that, in linear SCMs, holds with probability one if the coefficients are drawn from a distribution that is absolutely continuous with respect to Lebesgue measure [41], [60]. Assumption 4 ensures that any invariance found in the training environments \( \mathcal{E}^* \) can be generalized to all environments in \( \mathcal{E} \). Implicitly, it requires that the training environments are sufficiently heterogeneous. This type of assumption is also at the core of other invariance-based methods [2], [40], [53], [54].

At first glance, Assumption 4 suggests that we have to check the hypothesis \( H_0(S, \pi, \mathcal{E}) \) for all \( \pi \in \Pi^{S} \) to conclude whether or not \( S \) is \( d \)-invariant. Fortunately, as shown in Proposition 2, we actually only need to check the null hypothesis for a single policy \( \pi \in \Pi^{S} \).

Proposition 2. Assume Setting 1 and Assumption 4. Then, for all subsets \( S \subseteq \{1, \ldots, d\} \) and for all policies \( \pi, \tilde{\pi} \in \Pi^{S} \), it holds that
\[
H_0(S, \pi, \mathcal{E}) \text{ true} \iff H_0(S, \tilde{\pi}, \mathcal{E}) \text{ true}. \tag{12}
\]

Proof. See Appendix D.9, available in the online supplemental material.

Assumption 4 and Proposition 2 make the learning problem tractable. The task of testing whether a set \( S \) is \( d \)-invariant boils down to testing the \( \mathcal{E}^* \)-invariance hypothesis \( H_0(S, \pi^S, \mathcal{E}^*) \) for a single policy \( \pi^S \in \Pi^{S} \). We therefore have the flexibility to choose any \( \pi^S \) from \( \Pi^{S} \) to test the hypothesis (called the test policy). We discuss strategies for choosing the test policy in Section 4.4.

Testing \( H_0(S, \pi^S, \mathcal{E}^*) \) for \( \pi^S \in \Pi^{S} \) by directly checking for a change in the conditional distributions across environments in the observed data is, however, not in general possible. This is because the observed data may have been generated based on an initial policy \( \pi^{(0)} \) that does not satisfy

\[5\] The conditional expectation under \( \pi_a \) can also be written in terms of do-notation [46], that is, \( \forall a \in \mathcal{A}, x \in X : \mathbb{E}^{\pi_a, \pi}_e[R | X = x] = \mathbb{E}^{\pi}_e[R | X = x, \text{do}(A = a)] \). We use the \( \pi_a \) notation to make our presentation consistent.

\[6\] For example, if the training environments are identical, we clearly would not be able to distinguish \( d \)-invariant sets from other sets using the observed data. Assumption 4 prevents such cases.
\( \pi^0 \in \Pi^S \). It can therefore happen that \( H_0(S, \pi^S, \mathcal{E}^{tr}) \) is true but \( H_0(S, \pi^0, \mathcal{E}^{ct}) \) is not.

We illustrate this point using the example graph \( G \) given in Fig. 1b. For a policy depending only on \( S = \{2\} \) the environment \( e \) is d-separated from \( R \) given \( X^{[2]} \) in \( \mathcal{G}^{[2]} \), which implies that \( \{2\} \) is \( d \)-invariant, and in particular that \( H_0(\{2\}, \pi(\cdot), \mathcal{E}^{ct}) \) is true by the Markov property (see Lemma 3 in Appendix D.3, available in the online supplemental material). However, if the initial policy \( \pi^0 \) depends on both \( X^1 \) and \( X^2 \), then the path \( e \rightarrow X^1 \rightarrow A \rightarrow R \) in Fig. 1b is open, which implies, by Assumption 4, that \( H_0(\{2\}, \pi(\cdot), \mathcal{E}^{ct}) \) is not true. \(^7\)

Thus, in general, we cannot directly test the \( \mathcal{E}^{tr} \)-invariance hypothesis of a set \( S \) by using the observed data that were generated by the initial policy. Instead, we need to test \( H_0(S, \pi^S, \mathcal{E}^{ct}) \) for a policy \( \pi^S \in \Pi^S \) that is different from the data-generating policy \( \pi^0 \). As we detail in the following section, we can do so by applying an off-policy test for invariance by resampling the data to mimic the test policy \( \pi^S \).

### 4.2 Testing Invariance Under Distributional Shifts

Consider a fixed set \( S \subseteq \{1, \ldots, d\} \) and a pre-specified test policy \( \pi^S \in \Pi^S \) (see Section 4.4 for how to choose \( \pi^S \)). To test the hypothesis \( H_0(S, \pi^S, \mathcal{E}^{ct}) \), we apply the off-policy test from [68], which draws a target sample from \( \pi^S \) by resampling the offline data — drawn from \( \pi^0 \) — and then tests the invariance in this target sample. More formally, let \( \mathcal{E}^{ct} := \{e_1, \ldots, e_L\} \) and suppose that for every \( e_i \in \mathcal{E}^{ct} \) a dataset \( D_i \) consisting of \( n_e \) observations \( D_i^S = \{(X^i_j, A^i_j, R^i_j, \pi^S(A^i_j|X^i_j))\}_{j=1}^{n_e} \) is available. For each environment \( e_i \), we draw a weighted resampling \( D_i^{S, \pi^S} \) of \( D_i \) using the weighted resampling procedure introduced in [68].\(^8\) Then we apply an invariance test \( \varphi^{S}(D^{S, \pi^S}, \ldots, D^{S, \pi^S}) \) to the resampled data, to test the \( \mathcal{E}^{ct} \)-invariance hypothesis \( H_0(S, \pi^S, \mathcal{E}^{ct}) \). An invariance hypothesis test \( \varphi^{S} \) is a function (into \( \{0, 1\} \)) that takes data from environments \( e_1, \ldots, e_L \), each of size \( n_{e_i} \), and tests whether \( S \) is invariant. Here, \( \varphi^{S} = 1 \) indicates that we reject the \( \mathcal{E}^{ct} \)-invariance hypothesis. We detail a concrete test \( \varphi^{S} \) in Section 4.4. In Appendix F, available in the online supplemental material, we provide details on the resampling scheme, that is, a formal definition of \( D^{S, \pi^S} \) and show that the theoretical guarantees on the asymptotic level proved in [68] also extend to our application.

### 4.3 Algorithm for Invariant Policy Learning

The previous sections discuss finding invariant subsets \( S \). We now discuss how to employ this in an algorithm that learns an optimal invariant policy. We assume that we are given an off-policy optimization algorithm \( \text{off}_{\text{opt}} \) that takes as input a sample \( D := (D^1, \ldots, D^L) \) and a policy space \( \Pi^S \), and returns an optimal policy \( \pi^* \) and its estimated expected reward \( \bar{\mathbb{E}}^\pi[R] \).

Here, we present one choice of \( \text{off}_{\text{opt}} \) that we use in the experimental section; our approach can also be applied with other off-policy optimization algorithms. Given an invariant set \( S \) and a policy space \( \Pi^S \), we consider an optimal policy of the form

\[
\pi^S(a | x) := \arg\max_{a \in A} Q^S(x, a,),
\]

where \( Q^S(x, a) := \frac{1}{L} \sum_{l=1}^{L} \mathbb{E}_{\pi^S}[R | X^S = x] \) denotes the pooled conditional mean under the policy that always selects action \( a \).\(^9\)

Let \( \pi^0 \) be an initial policy generating the sample \( D \). By our assumption in Setting 1, the policy \( \pi^0 \) depends only on the observed covariates \( X \). We therefore have that for all \( S \subseteq \{1, \ldots, d\} \) the pooled conditional mean \( Q^S(x, a) \) is identifiable for all \( a \in A \) and \( x \in X^S \) as shown in Lemma 2 below.

**Lemma 2.** Let \( S \in \mathcal{S}_{\text{inv}} \) be a \( d \)-invariant set. It holds for all \( x \in X^S \) and all \( a \in A \) that

\[
Q^S(x, a) = \frac{1}{L} \sum_{l=1}^{L} \mathbb{E}_{\pi^S_{\text{off}}}[\frac{1}{\pi^0(A | X)} R | X^S = x].
\]

**Proof.** See Appendix D.7, available in the online supplemental material. \( \square \)

Here, we express the causal quantity \( Q^S(x, a) \) entirely in terms of expectations under the observed policy \( \pi^0 \) by using reweighting. Equivalently, one can also express \( Q^S(x, a) \) with the backdoor adjustment formula [46]. While the two formulations are equivalent, the resulting estimators are different (see the discussion in Appendix C, available in the online supplemental material).

We propose to estimate \( Q^S \) by a weighted least squares approach in which we consider a parameterized function class \( \{f_\theta : \mathcal{X}^S \times A \rightarrow \mathbb{R} | \theta \in \Theta^S\} \) and assume that there exists a unique \( \theta^0 \in \Theta^S \) such that for all \( x \in X^S \) and \( a \in A \) it holds that \( Q^S(x, a) = f_{\theta^0}(x, a) \). That is, we consider \( \hat{\theta}^0_n := \arg\min_{\theta \in \Theta^S} \frac{1}{n} \sum_{l=1}^{L} \sum_{i=1}^{n_e} \frac{(f_\theta(A^i_l, X^i_l) - R^i_l)^2}{\pi^0(A^i_l | X^i_l)} \),

where \( n := (n_{e_1}, \ldots, n_{e_L}) \). We then plug the estimate \( \hat{Q}^S := f_{\hat{\theta}^0_n} \) into (13) to obtain our (estimated) optimal policy. Proposition 3 shows that, under some regularity conditions, \( \hat{\theta}^0_n \) is a consistent estimate of \( \theta^0 \).

**Proposition 3.** Assume Setting 1 and Assumption 1. Let \( S \in \mathcal{S}_{\text{inv}} \) be a \( d \)-invariant set. Assume that

i. \( \Theta^S \) is compact,

ii. there exists a unique \( \theta^0 \in \Theta^S \) s.t. \( \forall x \in X^S, \forall a \in A : Q^S(x, a) = f_{\theta^0}(x, a) \) \( \mu \)-a.s.,

iii. \( \forall x \in X^S, \forall a \in A : 0 \rightarrow f_{\theta^0}(x, a) \) is continuous on \( \Theta^S \),

iv. \( \forall d \in \mathcal{E}^{ct} : \mathbb{E}_{\pi^S}[\sup_{\theta \in \Theta^S} (R - f_\theta(X, A))^2] < \infty \),

v. \( \exists \delta > 0 \text{ s.t. } \forall x \in X^S, \forall a \in A : \pi^S(a | x) \geq \delta. \)

7. In the same example, when conditioning on \( \{1, 2\} \), the path \( e \rightarrow X^1 \rightarrow U \rightarrow R \) is also open, which shows that \( S = \{1, 2\} \) is not a \( d \)-invariant set.

8. Importance weighting is not applicable here because the test statistics of an invariance test cannot be expressed in terms of weighted averages. See also the discussion in [68].

9. In our framework, changing the policy corresponds to intervening on the underlying SCM (see Setting 1). The expression \( Q^S(x, a) \) is derived from expectations under such interventions and can thus be considered a causal quantity.
Algorithm 1. Learning an Optimal Invariant Policy

**Input:** data $D = (D^1, \ldots, D^t)$, off-policy optimization $\text{off}_\text{opt}$, hypothesis tests and test policies $\{(\psi^s, \pi^s)\}_{S \subseteq \{1, \ldots, d\}}$

1. Initialize maximum reward $\text{maxR} \leftarrow -\infty$.
2. Initialize optimal invariant policy $\pi^\text{inv} \leftarrow \text{null}$.

**Loop over all subsets** $S \subseteq \{1, \ldots, d\}$

- **Test for invariance**
  - $\text{is_inv} \leftarrow \text{test_inv}(D, \pi^S, \psi^S, S)$; 
  - (see Algorithm 2)

- **Update best invariant set**
  - if $\text{is_inv}$ then
    - $\pi^S_\text{inv}, \tilde{E}^S(R) \leftarrow \text{off}_\text{opt}(D, \Pi^S)$;
  - if $\text{maxR} < \tilde{E}^S(R)$ then
    - $\text{maxR} \leftarrow \tilde{E}^S(R)$;
    - $\pi^\text{inv} \leftarrow \pi^S_\text{inv}$.

**Output:** optimal invariant policy $\pi^\text{inv}$

Algorithm 2. Testing the Invariance of a Set $S$ With Given Test Policy $\pi^S$

**Function:** test_inv data $D = (D^1, \ldots, D^t)$, test policy $\pi^S$, hypothesis test $\psi^S$, target set $S$

1. // resampling according to $\pi^S$
2. for $e = e_1, \ldots, e_t$ do
3.   for $i = 1$ to $D^e$ do
4.     compute weights: $r_i^e \leftarrow \frac{\pi^S(e_i|\pi^S)}{\pi^S(S|e_i)}$;
5.   end
6. choose resampling size $m_i$ with GOF-heuristic in [68];
7. draw $D^{e-x} := (D^1_{\text{tr}}, \ldots, D^e_{\text{tr}})$ from $D^e$ with prob. $\propto \prod_{i=1}^{m_i} r_i^e$;
8. // verifying invariance condition is_invariant := $\psi^S(D^e)$;
9. return is_invariant

4.4 Specifications of the Target Test

The resampling procedure detailed in Algorithm 2 requires a hypothesis test for the $\mathcal{E}^e$-invariance null hypothesis that has power against the alternatives. We discuss one such test in Section 4.4.1 below. Moreover, in Sections 4.4.2 and 4.4.3, we discuss two choices of the test policy that aim to improve the power of the resampling test.

4.4.1 Invariant Residual Distribution Test

We now detail a test $\varphi^S$ to test $\mathcal{E}^e$-invariance in the resampled data. We first pool data from all environments into one dataset and estimate the conditional $E^e\pi^S(R|X^S)$ using any prediction method (such as linear regression or a neural network). We then test whether the residuals $R - E^e\pi^S(R|X^S)$ are equally distributed across the environments $e \in \mathcal{E}$, i.e., we split the sample back into $L$ groups (corresponding to the environments) and test whether the residuals in these groups are equally distributed (see also [49], for example). We then define $\varphi^S$ to be the composition of these operations, that is, $\varphi^S$ returns 1 if the test for equal distribution of the residuals is rejected.

In the simulation and the warfarin case study (Section 5 and 6), we use the Kruskal-Wallis test [34] to test whether the residuals have the same mean across environments; this test holds pointwise asymptotic level for all $\alpha \in (0, 1)$ (see Proposition 8 in Appendix F, available in the online supplemental material). To obtain power against more alternatives, one could also use other tests, such as a two-sample kernel test with maximum mean discrepancy [22] and then correct for the multiple testing using Bonferroni-corrections (see also [54], for example).

4.4.2 Optimizing the Test Policy for Power

To check whether a subset $S$ is invariant, we only need to test the $\mathcal{E}^e$-invariance for a single policy $\pi \in \Pi^S$ (see Proposition 2). This provides us with a degree of freedom that we can leverage. Intuitively, the non-invariance may be more easily detectable in some test policies compared to others. We can therefore try to find a policy that gives us the strongest signal for detecting non-invariance. We approximate
maximizing the power of the test by minimizing the p-value of the test. In a population setting, this would return small p-values for non-invariant sets, whereas for invariant sets one would not be able to make the p-values arbitrarily small, since they are uniformly distributed. In a finite sample setting, this type of power optimization can lead to overfitting (which would break any level guarantees); to avoid this we use sample splitting.

As presented in Section 4.2, for each environment \( e \), we obtain a target sample \( D^e \) from a test policy \( \pi^e \) by resampling the sample \( D^e \) that was generated under the policy \( \pi^e \), and then test \( \mathcal{E} \)-invariance in the target sample. The probabilities for obtaining the resampled sample conditioned on the original sample are given by the importance weights, see Appendix F, available in the online supplemental material. Here, we optimize the ability to detect non-invariance over a parameterized subclass of \( \Pi^S \),

\[
\Pi^S_\theta := \{ \pi^e_\theta \mid \theta \in \Theta \},
\]

where \( \Theta = \{ a \in A \mid \pi^e_{a} \in \Pi^S \} \) and \( \pi^e_\theta \) is a linear softmax policy, i.e., for all \( x \in \mathbb{R}^{|S|} \) and \( a \in A \):

\[
\pi^e_\theta(a \mid x) = \frac{\exp(\theta^e_a x^e)}{\sum_a \exp(\theta^e_a x^e)}.
\]

To check for the \( \mathcal{E} \)-invariance condition of a subset \( S \), the idea is then to find a policy \( \pi^e_\theta \in \Pi^S_\theta \) such that, in expectation, the p-value of the test is minimized, i.e., we aim to solve the following optimization problem

\[
\arg\min_{\theta \in \Theta} \mathbb{E} \left[ p_v(D^S_\theta) \mid D \right],
\]

where \( D := (D^1, \ldots, D^e) \), \( D^S_\theta := (D^1 \pi^e_{a_1}, \ldots, D^e \pi^e_{a_e}) \) are the observed and resampled data, respectively, and \( p_v \) is a function that takes as input the resampled sample \( D^S_\theta \) and outputs the p-value of the test. Since we condition on \( D \), the expectation is only with respect to the resampling of \( D^S_\theta \). We then employ a gradient-based optimization algorithm to solve the above optimization problem, where the gradient is derived using the log-derivative. More precisely, let \( J(\theta) := \mathbb{E}[p_v(D^S_\theta) \mid D] \) be our objective function which now depends on the parameters \( \theta \). The gradient of the objective function \( J(\theta) \) can be derived as follows

\[
\nabla J(\theta) = \nabla \mathbb{E} \left[ p_v(D^S_\theta) \mid D \right] = \mathbb{E} \left[ \nabla \log p_v(D^S_\theta) \mid D \right] - \mathbb{E} \left[ \nabla \log p_v(D^S_\theta) \mid D \right].
\]

This expectation can be estimated by drawing repeated resamples \( D^S_\theta \), where \( p_v(D^S_\theta \mid D) \) is determined by the resampling weights, corresponding to sampling with replacement instead of distinct weights.

The optimization procedure yields a policy \( \pi_\theta^* \) that approximately satisfies \( \pi_\theta^* \in \arg\min_{\pi \in \Pi^S} J(\theta) \). We can then use \( \pi_\theta^* \) as the test policy for testing whether \( S \) is invariant. Lastly, to preserve the level of the statistical test, we split the original sample into two halves, perform the power optimization procedure on one half, and verify the invariance condition on the other half. The algorithm is presented in Algorithm 4 in Appendix I, available in the online supplemental material. We only use the approximation of the resampling weights for the power optimization and use the actual resampling weights for the final resampling, so the level guarantee of Proposition 8 in Appendix F, available in the online supplemental material, still holds.

4.4.3 Using a Uniform Target Distribution

Since the procedure in Section 4.4.2 may be computationally challenging, especially if the algorithm is repeated many times as in Section 5. A computationally simpler approach is for each \( a \in A \) to test invariance under the test policy \( \pi_{a} \), which always chooses the action \( a \), and then combine the resulting p-values using Bonferroni corrections [19]. Beyond computational simplicity, this has an additional benefit: Across environments there may be a cancelling effect of the difference in means due to different dependencies on the action in each environment. By testing the invariance of the conditional mean of the reward in each action, such cancelling effects are accounted for.

4.5 Learning Causal Ancestors Under Distributional Shifts

Sections 4.1 and 4.2 discuss an approach to learn invariant sets from offline data. The learned invariant sets are then used to find an optimal invariant policy as discussed in Section 4.3. Besides learning an optimal invariant policy, one can further use the proposed off-policy invariance test to analyze the causal structure. More specifically, the learned invariant sets allow us to look for potential observed causal ancestors \( \text{AN}(R)^{10} \) of \( R \) by taking the intersection of the accepted sets. This approach is similar to invariant causal prediction [49], except that here, we employ the off-policy invariance test to account for the distributional shift between the initial and the test policies, and allow for hidden variables.

Now we outline a method for finding \( \text{AN}(R) \) from the offline data obtained from multiple environments \( D^1, \ldots, D^e \). Recall that we denote by \( D^S_{\psi} \) a weighted resample of \( D^e \), and \( \psi^S \) an invariance test for the \( \mathcal{E} \)-invariance hypothesis \( H_0(S, \pi^e, \mathcal{E}) \), as discussed in Section 4.2 and Appendix F, available in the online supplemental material. For ease of presentation, we assume that \( n_{e_1} = \ldots = n_{e_l} := n \). Then, we propose to estimate the causal ancestors of \( R \) by

\[
\hat{S}_{\text{AN}}^S := \bigcap_{S \ni \bar{S}_i \in \{ D^1, \ldots, D^e \}} S_{i=0}. \tag{17}
\]

We detail the whole procedure in Algorithm 3 in Appendix G, available in the online supplemental material. Proposition 4

10. Formally, \( \text{AN}(R) \subseteq \{ 1, \ldots, d \} \) is defined as the set of indices \( j \) for which there is a directed path from \( X^j \) to \( R \) in \( G \).
shows that this method controls the probability of wrongly selecting a non-causal-ancestor variable.

**Proposition 4.** Assume Setting 1, and that \( S_{\text{inv}} \) is non-empty. Let \( \hat{S}_{\text{AN}} \) be the estimated set of causal ancestors given in (17) and assume that the invariance tests \( \psi_{e} \) used in (17) have pointwise asymptotic level \( \alpha \in (0, 1) \). It then holds that

\[
\liminf_{n \to \infty} \mathbb{P}(\hat{S}_{\text{AN}} \subseteq \Lambda(R)) \geq 1 - \alpha.
\]

**Proof.** See Appendix D.10, available in the online supplemental material.

## 5 SIMULATION EXPERIMENTS

To verify our theoretical findings we perform two simulation experiments, where we consider a linear multi-environment contextual bandit setting similar to Example 1 with the following SCM \( S(\pi, e) \) (which induces the graph shown in Fig. 1b):

\[
U := \epsilon_{U}, \quad X_{1} := \gamma_{U} + \epsilon_{X_{1}}, \quad X_{2} := \alpha_{e} + \epsilon_{X_{2}}, \quad A \sim \pi(A \mid X_{1}, X_{2}), \quad R := \beta_{A,1} X_{2} + \beta_{A,2} U + \epsilon_{R},
\]

where \( \epsilon_{U}, \epsilon_{X_{1}}, \epsilon_{X_{2}}, \epsilon_{R} \sim \mathcal{N}(0, 1) \), \( A \) takes values in the space \( \{a_{1}, \ldots, a_{3}\} \), \( \gamma_{U} \) and \( \alpha_{e} \) are parameters that depend on the environment \( e \), and \( \beta_{0,1}, \ldots, \beta_{0,1}, \beta_{0,2}, \ldots, \beta_{0,2} \) are parameters that are fixed across environments. Appendix J.1, available in the online supplemental material, contains details on how the parameters are chosen in the experiments. The code for all the experiments is available at https://github.com/sorawitj/invariant-policy-learning.

### 5.1 Generalization and Invariance

We first consider an oracle setting, where we know a priori which subsets are invariant. From our data-generating process, it follows that \( \{X_{2}\} \) is the only invariant set. We then compare an invariant policy which depends only on \( X_{2} \) with a policy that uses both \( X_{1} \) and \( X_{2} \). We train both policies on a dataset of size 100000 obtained from multiple training environments under a fixed initial policy \( \pi^{0} \) (see Appendix J.2, available in the online supplemental material). In both cases, we employ a weightless squares to estimate the expected reward \( \mathbb{E}[R \mid A, X^{S}] \), where \( S \) is the subset that the policy uses. The policy then takes a greedy action w.r.t. the estimated expected reward, i.e.,

\[
\arg\max_{a} \mathbb{E}[R \mid A = a, X^{S}]
\]

(see Section 4.3). Then we evaluate both policies on multiple unseen environments and compute the regret with respect to the policy that is optimal in each of the unseen environments. Fig. 2 shows the results. Each data point represents the evaluation on an unseen environment. The y-axes show the regret value and the x-axes display the distance from each unseen environment to the training environments (the distance is computed as the \( \ell^{2} \)-distance between the average value of the pairs \( (Y_{e}, \alpha_{e}) \) in the training environments and the pair \( (Y_{e}, \alpha_{e}) \) in the unseen test environment). The plot shows that the worst-case regret of the invariant policy is smaller than that of the non-invariant policy. In particular, for environments different from the training environments the gain can be significant. This empirically supports our result of Theorem 1.

### 5.2 Learning Invariant Policies

In practice, we do not know in advance which sets are invariant. We now aim to find an invariant policy from a dataset generated under an initial policy \( \pi^{0} \) which takes both \( X_{1} \) and \( X_{2} \) as input. To do so, we employ the method proposed in Section 4.2 for testing invariance under distributional shifts. More precisely, we generate a dataset of size \( n \) from multiple training environments under the initial policy \( \pi^{0} \) and apply the off-policy invariance test (see Section 4.4) to verify the invariance property of each subset in \( \{\emptyset, \{X_{1}\}, \{X_{2}\}, \{X_{1}, X_{2}\}\} \). We repeat the experiment 500 times and plot the acceptance rates at various sample sizes \( (n = 1000, 3000, 9000, 27000, 81000) \) (these numbers denote the total sample size, that is, number of observations, summed over all environments). The resulting acceptance rates are shown in Fig. 3. Our method yields high acceptance rates for the set \( \{X_{2}\} \), which indeed is invariant, while the acceptance rates for other sets gradually decrease as the sample size increases. Furthermore, we can see that our test is more powerful when the number of training environments increases (keeping the total number of observations fixed). Our test is conservative (the acceptance rate is above the 95% level in the left plot) because the target test is not exact (the true conditional expectation is not given). In Appendix J.3, available in the online supplemental material, we conduct the same experiment with an exact test, using the true conditional expectation, which shows the correct level.

## 6 WARFARIN DOSING CASE STUDY

We evaluate our proposed approach on the clinical task of warfarin dosing. Warfarin is a blood thinner medicine prescribed to patients at risk of blood clots. The appropriate dose of warfarin varies from patient to patient depending
on various factors such as demographic and genetic information [14]. Our case study is based on the International Warfarin Pharmacogenetics Consortium (IWPC) dataset [14] which consists of 5'700 patients who were treated with warfarin, collected from 21 research groups on 4 continents. The IWPC dataset contains the optimal dose of warfarin for each of the patients as well as their information on demographic characteristics, clinical and genetic factors. The warfarin dosing problem has been used in a number of previous works evaluating off-policy learning algorithms [8], [31], [74]. Similarly to these works, we formulate the warfarin dosing problem as a multi-environment contextual bandit problem as follows.

- The covariates \( (X) \) are patient-level features including demographic, clinical and genetic factors.
- The actions \( (A) \) are recommended warfarin doses output by a policy. We discretize the actions into three equal-sized buckets (low, medium, high) based on the quantiles of the optimal warfarin dose.
- The reward \( (R) \) depends on the recommended dose and the optimal dose: For each patient \( i \), the reward \( R_i(a) \) for an action \( a \in \{ \text{low, medium, high} \} \) is computed as

\[
R_i(a) := Y_i - m(a),
\]

where \( Y_i \) is the optimal warfarin dose for a patient \( i \) and \( m(a) \) is a median value of the optimal warfarin doses within the bucket \( a \). Here, we assume that neither the reward function nor the optimal warfarin doses are known to the agent. Instead, for each patient \( i \), only the reward for the action \( A_i \) is observed, i.e., \( R_i := R_i(A_i) \).
- The environments \( (E) \) are proxies for continents. The continent information is not directly contained in the dataset, but we create proxies for the continent by clustering the 21 research groups into 4 clusters based on their proportion of the patients’ race within each group. We believe that the resulting clusters roughly correspond to 4 different continents.

To reduce the search space, we select the top 10 features that are most predictive for the optimal warfarin dose using the permutation feature importance method [12]. The top 10 features include 4 demographic variables, 4 clinical factors, and 2 genetic factors.

We consider two experimental setups to illustrate the benefits of our invariant learning approach. In the first setup, we directly apply our method to the IWPC dataset. Here, including invariance does not seem necessary in that our method performs similarly to other baselines (but not worse). It does, however, generate some causal insight into the problem. The second setup is a semi-real setting, where we introduce an artificial, non-invariant confounder.

We now outline our first experimental setup and the results. We first generate training data \( \{(X_i, A_i, R_i, e_i)\}_{i=1}^{n_{train}} \) by drawing actions \( A_i \) from a policy \( \pi^0 \in \Pi^{BMI} \) that is constructed from linear regression \( Y_i \approx f(X_i^{BMI}) \) of the optimal dose onto the BMI (see Appendix K.1, available in the online supplemental material, for more details).

### 6.1 Candidate Methods

Using the generated training data, we empirically compare the performance of the following policy learning methods:

- Invariant Policy Learning (Inv): This is our proposed method. We first perform the off-policy invariance test using the test described in Section 4.4.3 to search for potential invariant sets. We then take the top 20 sets with the largest p-values \( S_i^{inv} \), as the candidate invariant sets. For each \( S \) in \( S_i^{inv} \), we fit the policy optimization algorithm described in Section 4.3 with \( X^S \) as the covariates (the same algorithm is also used in other candidate methods below). Lastly, we select the top 3 sets that yield the largest expected rewards (computed using 5-fold cross-validation).

- Predictive Policy Learning (Pred): This method serves as a baseline for policy learning that solely maximizes the expected reward. For each subset \( S \), we fit the policy optimization algorithm with \( X^S \) as the covariates. We then take the policies corresponding to the top 3 sets with the largest expected rewards.

- All Set Policy Learning (All): This method serves as another baseline where we take all of the patient’s features and fit the policy optimization algorithm.

### 6.2 Evaluation Setup & Results

We compare the policy learning methods using the following ‘leave-one-environment-out’ evaluation procedure.

1. Select \( e \in E = \{1, \ldots, 4\} \) as a test environment. Split the training data into \( D_{test} := \{(X_i, A_i, R_i, e_i)\}_{i=1}^{n_{test}} \), where \( e_i = e \) and \( D_{train} := \{(X_i, A_i, R_i, e_i)\}_{i=1}^{n_{train}} \), where \( e_i \in \{1, \ldots, 4\} \setminus \{e\} \).
2. Using \( D_{train} \), train the policies with candidate methods detailed in Section 6.1.
3. Evaluate the fitted policies by computing the expected reward on \( D_{test} \) using the true reward function \( 19 \).

We repeat the above procedure for each \( e \in E \) and display the evaluation result in Fig. 4. The performances of all candidate methods are similar. Even though the proposed invariant approach does not yield a higher reward compared with the baselines, it does not worsen the performance, either. This suggests that we can gain the stability benefit of an invariant policy without having to sacrifice predictiveness. Indeed, the stability benefit could prevent the learned invariant policy from being suboptimal when a new test environment is sufficiently different from the training environments as we show in Section 6.4.

### 6.3 Analyzing Invariant Sets

In addition to learning an optimal invariant policy, we can use the invariance-based approach to further analyze the dependence between the patient’s features and the reward as discussed in Section 4.5. In particular, we apply the off-policy invariant causal prediction algorithm (see Algorithm 3 in Appendix G, available in the online supplemental material) to find potential causal ancestors of the reward. On this dataset, with a confidence level of 5%, the algorithm returns the empty set, which can happen if the covariates
are highly correlated, for example [25]. Nonetheless, we can still extract more information by obtaining the defining sets (see Section 2.2 in [25]). The resulting defining set of size 2 is \{Race, VKORC1\} (see Appendix K.2, available in the online supplemental material, for more details on the variables). These variables are potential causal ancestors in the sense that at least one variable in these sets is a causal ancestor with high probability.

### 6.4 Semi-Real Experiment

To further illustrate the benefits of the invariance-based learning approach, we consider a semi-real setup where we introduce hidden variables and a non-invariant predictor. We remove the two genetic factors from the patient’s features and create a non-invariant predictor that depends on those two factors as follows.

We first fit a linear regression to estimate the optimal warfarin dose from the genetic factors and denote the resulting coefficients by \(\beta\). To mimic environmental perturbations, we perturb \(\beta\) depending on an environment \(e \in E\) resulting in \(\beta_e \coloneqq \gamma_e \beta\), where \(\gamma_e\) is an environment-specific parameter. We define the non-invariant predictor in the environment \(e \in E\) as \(X^{n-inv} \coloneqq X^G \beta_e\), where \(X^G\) are the two genetic features. We then add \(X^{n-inv}\) as part of the patient’s features and remove \(X^G\). The training data are generated in a similar fashion as in the first setup, except that the initial policy does not only depend on the BMI score \(X_{BMI}\) but also on the non-invariant predictor \(X^{n-inv}\).

In addition to the candidate methods described in Section 6.1, we introduce an additional baseline for this setup.

- **Oracle Invariant Policy (Oracle-Inv):** By construction, we know that \(X^{n-inv}\) is a strongly non-\(d\)-invariant variable (see Definition 5). This method serves as an oracle version of the invariant policy learning method by searching for the top 3 sets that do not contain \(X^{n-inv}\) such that their corresponding policies yield the largest expected reward (the procedure is similar to the Pred method with \(X^{n-inv}\) being removed).

We evaluate the candidate methods using a similar procedure as described in Section 6.2. Fig. 5 illustrates the evaluation result. Our proposed method (Inv) yields a higher expected reward than the two baselines on most of the test environments. This is because the two baselines ignore the environment structure and use information from \(X^{n-inv}\) in their resulting policies, while the invariant method uses the invariance test to remove this non-invariant proxy variable. Furthermore, the performance of our proposed method is almost on par with the invariant oracle (Oracle-Inv), except for the test environment \(e = 3\), in which our approach is unable to ignore the non-invariant predictor, possibly because the non-invariance that would be implied by Assumption 4 may not be strong enough (for our test) when \(\mathcal{E}_{tr} = \{1, 2, 4\}\).

### 7 Conclusion

This paper tackles the problem of environmental shifts in offline contextual bandits from a causal perspective. We introduce a framework for multi-environment contextual bandits that is based on structural causal models and frame the environmental shift problem as a distributionally robust objective over environments that are induced by different perturbations on the covariates. We prove that if all relevant covariates have been observed, taking into account causality and invariance is not necessary for obtaining the distributionally robust policies. However, causality and invariance can become relevant when not all variables are observed. To tackle settings with unobserved covariates, we adapt invariance-based ideas from causal inference to the proposed framework and introduce the notion of invariant policies.

Our theoretical results show that under certain assumptions an invariant policy that is optimal on the training environments is also optimal on all unseen environments, and therefore distributionally robust. We further provide a method for finding invariant policies based on an off-policy invariance test. It can be combined with any existing policy optimization algorithm to learn an optimal invariant policy.

For future work, there are several directions that would be interesting to investigate. One direction is to explore the use of invariance-based ideas in the adaptive setting, in which the goal of an agent is to optimally adapt to a changing environment. Learning agents may require fewer and safer explorations in a new environment if they carry over invariance information from previous environments. It may
further be possible to extend invariance-based ideas from the contextual bandit setting to the full reinforcement learning problem with long-term consequences and state dynamics. Although some previous works have explored this direction [59, 75], we believe that the connections with respect to causality and invariance are not yet fully understood. In the supervised learning setting, recent work has investigated trading off invariance and predictability [26, [44], [53], [55], [56]. We believe that a similar idea can be applied to contextual bandit and reinforcement learning problems. Lastly, if one can gain additional knowledge of the test environments, one may aim to optimize objectives other than the worst-case performance which could lead to a different class of generalization guarantees.

This paper considers invariance as a dichotomous property and could be a first step towards using invariance-based ideas for building safer and more robust adaptive learning systems.  

ACKNOWLEDGMENTS

We thank Steffen Lauritz for helpful discussions. Finally, we thank an anonymous associate editor and anonymous referees for their helpful comments on the manuscript.

REFERENCES

[1] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, “Concrete problems in ai safety,” 2016, arXiv:1606.06565.
[2] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, “Invariant risk minimization,” 2019, arXiv: 1907.02833.
[3] S. Athey and S. Wager, “Policy learning with observational data,” Econometrica, vol. 89, no. 1, pp. 133–161, 2021.
[4] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, “Recent advances in adversarial training for adversarial robustness,” in Proc. 30th Int. Joint Conf. Artif. Intell., 2021, pp. 4312–4321.
[5] E. Bareinboim, A. Forney, and J. Pearl, “Bandits with unobserved confounders: A causal approach,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2015, pp. 1342–1350.
[6] E. Bareinboim and J. Pearl, “Transportability from multiple environments with limited experiments: Completeness results,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2014, pp. 280–288.
[7] E. Bareinboim and J. Pearl, “Causal inference and the data-fusion problem,” Proc. Nat. Acad. Sci. USA, vol. 113, no. 27, pp. 7345–7352, 2016.
[8] D. Bertsimas and C. McCord, “Optimization over continuous and multi-dimensional decisions with observational data,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2018, pp. 2966–2974.
[9] A. Beygelzimer and J. Langford, “The offset tree for learning with partial labels,” in Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2009, pp. 129–138.
[10] S. Bongers, P. Forr, J. Peters, and J. M. Mooij, “Foundations of structural causal models with cycles and latent variables,” Ann. Statist., vol. 49, no. 5, pp. 2885–2915, 2021.
[11] L. Bottou et al., “Counterfactual reasoning and learning systems: The example of computational advertising,” J. Mach. Learn. Res., vol. 14, no. 65, pp. 3207–3260, 2013.
[12] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
[13] R. Christiansen, N. Pister, M. E. Jakobsen, N. Gnecco, and J. Peters, “A causal framework for distribution generalization,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 10, pp. 6614–6630, Oct. 2022.
[14] I. W. P. Consortium, “Estimation of the warfare dose with clinical and pharmacogenetic data,” New England J. Med., vol. 360, no. 8, pp. 753–764, 2009.
[15] S. Correa and E. Bareinboim, “General transportability of soft interventions: Completeness results,” Adv. Neural Inf. Process. Syst., vol. 33, pp. 10 902–10 912, 2020.
[16] A. P. Dawid, “Influence diagrams for causal modelling and inference,” Int. Statist. Rev., vol. 70, no. 2, pp. 161–189, 2002.
[17] A. A. de Kroon, D. Belgrave, and J. M. Mooij, “Causal discovery for causal bandits utilizing separating sets,” 2020, arXiv: 2009.07916.
[18] M. Dudík, J. Langford, and L. Li, “ Doubly robust policy evaluation and learning,” in Proc. 28th Int. Conf. Mach. Learn., 2011, pp. 1097–1104.
[19] Ó. J. Dunn, “Multiple comparisons among means,” J. Amer. Statist. Assoc., vol. 56, no. 293, pp. 52–64, 1961.
[20] J. Garcia and F. Fernández, “A comprehensive survey on safe reinforcement learning,” J. Mach. Learn. Res., vol. 16, no. 1, pp. 1437–1480, 2015.
[21] C. Glymour, K. Zhang, and P. Spirtes, “Review of causal discovery methods based on graphical models,” Front. Genet., vol. 10, 2019, Art. no. 524.
[22] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test,” J. Mach. Learn. Res., vol. 13, no. 25, pp. 723–773, 2012.
[23] C. Guostrin, D. Koller, and R. Parr, “Multiagent planning with factored MDPs,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2002, pp. 1523–1530.
[24] C. Guostrin, D. Koller, R. Parr, and S. Venkataaraman, “Efficient solution algorithms for factored MDPs,” J. Artif. Int. Res., vol. 19, no. 1, pp. 399–468, 2003.
[25] C. Heinze-Deml, J. Peters, and N. Meinshausen, “Invariant causal prediction for nonlinear models,” J. Causal Inference, vol. 6, no. 2, pp. 1–35, 2018.
[26] M. E. Jakobsen and J. Peters, “Distributional robustness of K-class estimators and the PULSE,” Econometrics J., vol. 25, no. 2, pp. 404–432, 2021.
[27] A. Jonsson and A. Barto, “Causal graph based decomposition of factored MDPs,” J. Mach. Learn. Res., vol. 7, no. 81, pp. 2259–2301, 2006.
[28] Y. Jung, J. Tian, and E. Bareinboim, “Estimating causal effects using weighting-based estimators,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 10186–10193.
[29] Y. Jung, J. Tian, and E. Bareinboim, “Learning causal effects via weighted empirical risk minimization,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 12609–12620.
[30] N. Kallus, “Balanced policy evaluation and learning,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 8909–8920.
[31] N. Kallus and A. Zhou, “Policy evaluation and optimization with continuous treatments,” in Proc. Int. Conf. Artif. Intell. Statist., 2018, pp. 1243–1251.
[32] N. Kallus and A. Zhou, “Confounding-robust policy evaluation in infinite-horizon reinforcement learning,” in Adv. Neural Inf. Process. Syst., 2020, pp. 22 293–22 304.
[33] M. Kearns and D. Koller, “Efficient reinforcement learning in factored MDPs,” in Proc. 16th Int. Joint Conf. Artif. Intell., 1999, pp. 740–747.
[34] W. H. Kruskal and W. A. Wallis, “Use of ranks in one-criterion variance analysis,” J. Amer. Statist. Assoc., vol. 47, no. 260, pp. 583–621, 1952.
[35] J. Langford and T. Zhang, “The epoch-greedy algorithm for multi-armed bandits with side information,” in Proc. Adv. Neural Inf. Process. Syst., 2008, Art. no. 1.
[36] F. Lattimore, T. Lattimore, and M. D. Reid, “Causal bandits: Learning good interventions via causal inference,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 1199–1197.
[37] T. Lakkaraju and C. Guestrin, “Generalized causal discovery with independent interventions,” in Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2009, pp. 129–138.
[38] J. Langford and T. Zhang, “Multi-armed bandits with side information,” in Proc. Adv. Neural Inf. Process. Syst., 2008, Art. no. 1.
[39] S. Lee, J. D. Correa, and E. Bareinboim, “Domain adaptation by using causal inference to predict invariant conditional distributions,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 10869–10879.
[40] S. Magliacane, T. van Ommen, T. Claassen, S. Bongers, P. Versteeg, and J. M. Mooij, “Domain adaptation via invariant feature representation,” in Proc. 30th Int. Conf. Mach. Learn., 2013, pp. 10–18.
[41] W. K. Newey and D. McFadden, “Chapter 36 large sample estimation and hypothesis testing,” Handbook of Econometrics, vol. 4, Amsterdam, The Netherlands: Elsevier, 1994, pp. 2111–2245.
M. Oberst, N. Thams, J. Peters, and D. Sontag, “Regularizing towards causal invariance: Linear models with proxies,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 8260–8270.

J. Pearl, “Causal diagrams for empirical research,” Biometrika, vol. 82, no. 4, pp. 669–688, 1995.

P. Spirtes, C. Glymour, and R. Scheines, Causality. Cambridge, U.K.: Cambridge Univ. Press, 2009.

J. Pearl, Causal Inference in Statistics : A Primer. Hoboken, NJ, USA: Wiley, 2016.

J. Pearl and E. Bareinboim, “Transportability of causal and statistical relations: A formal approach,” in Proc. 25th AAAI Conf. Artif. Intell., 2011, pp. 247–254.

J. Peters, P. Bühlmann, and N. Meinshausen, “Causal inference using invariant prediction: Identification and confidence intervals,” J. Roy. Statist. Soc. Ser. B, vol. 78, pp. 947–910, 2016.

J. Peters, D. Janzing, and B. Schölkopf, Elements of Causal Inference: Foundations and Learning Algorithms. Cambridge, MA, USA: MIT Press, 2017.

N. Pfister, P. Bühlmann, and J. Peters, “Invariant causal prediction for sequential data,” J. Amer. Statist. Assoc., vol. 114, no. 527, pp. 1264–1276, 2018.

S. Saengkyongam and J. Peters, “Learning stable and predictive structures in kinetic systems,” Proc. Nat. Acad. Sci., vol. 116, no. 51, pp. 25 405–25 411, 2019.

N. Pfister, E. G. Williams, J. Peters, R. Aebersold, and P. Bühlmann, “Stabilizing variable selection and regression,” Ann. Appl. Statist., vol. 15, no. 3, pp. 1220–1246, 2021.

M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters, “Invariant models for causal transfer learning,” J. Mach. Learn. Res., vol. 19, pp. 1309–1342, 2018.

D. Rothenhäusler, N. Meinshausen, P. Bühlmann, and J. Peters, “Anchor regression: Heterogeneous data meet causality,” J. Roy. Statist. Soc., vol. 83, no. 2, pp. 215–246, 2021.

S. Saengkyongam, L. Henckel, N. Pfister, and J. Peters, “Exploiting independent instruments: Identification and distribution generalization,” in Proc. 39th Int. Conf. Mach. Learn., 2022, pp. 18 935–18 958.

B. Schölkopf, D. Janzing, J. Peters, E. Shroff, K. Zhang, and J. M. Mooij, “On causal and anticausal learning,” in Proc. 29th Int. Conf. Mach. Learn., 2012, pp. 1255–1262.

R. Sen, K. Sharmugam, M. Kocaoglu, A. Dimakis, and S. Shakkottai, “Contextual bands with latent confounders: An NMF approach,” in Proc. Conf. Artif. Intell. Statist., 2017, pp. 518–527.

A. Sonar, V. Pacelli, and A. Majumdar, “Invariant policy optimization towards stronger generalization in reinforcement learning,” in Proc. Conf. Learn. Dyn. Control, 2021, pp. 21–33.

P. Spirtes, C. Glymour, and R. Scheines, Causation, Prediction, and Search, 2nd ed. Cambridge, MA, USA: MIT Press, 2000.

A. Strehl, J. Langford, L. Li, and S. M. Kakade, “Learning from logged implicit exploration data,” in Proc. Int. Conf. Mach. Learn., 2010, pp. 2217–2225.

A. Subbaswamy, P. Schulam, and S. Saria, “Preventing failures due to dataset shift: Learning predictive models that transport,” in Proc. 22nd Int. Conf. Artif. Intell. Statist., 2019, pp. 3118–3127.

M. Sugiyama and M. Kawanabe, Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation. Cambridge, MA, USA: MIT Press, 2012.

A. Swaminathan and T. Joachims, “Counterfactual risk minimization: Learning from logged bandit feedback,” in Proc. 32nd Int. Conf. Mach. Learn., 2015, pp. 814–823.

A. Swaminathan and T. Joachims, “The self-normalized estimator for counterfactual learning,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 3231–3239.

G. Tennenholtz, U. Shalit, and S. Mannor, “Off-policy evaluation in partially observable environments,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 10 276–10 283.

G. Tennenholtz, U. Shalit, S. Mannor, and Y. Efroni, “Bandits with partially observable confounded data,” in Proc. Conf. Uncertainty Artif. Intell., 2021, pp. 430–439.

N. Thams, S. Saengkyongam, N. Pfister, and J. Peters, “Statistical testing under distributional shifts,” 2021, arXiv:2105.10821.

J. Tian, A. Paz, and J. Pearl, “Finding minimal d-separators,” University of California, Los Angeles, Tech. Rep. R-254, 1998.