Adapting to Latent Subgroup Shifts via Concepts and Proxies

Ibrahim Alabdulmohsin
Google Research

Nicole Chiou[1]
Stanford University

Alexander D'Amour
Google Research

Arthur Gretton
Gatsby Computational Neuroscience Unit

Sanmi Koyejo
Google Research
Stanford University

Matt J. Kusner[1]
University College London

Stephen R. Pfohl
Google Research

Olawale Salaudeen[1]
University of Illinois Urbana-Champaign

Jessica Schrouff[2]
Google Research

Katherine Tsai[1]
University of Illinois Urbana-Champaign

Abstract

We address the problem of unsupervised domain adaptation when the source domain differs from the target domain because of a shift in the distribution of a latent subgroup. When this subgroup confounds all observed data, neither covariate shift nor label shift assumptions apply. We show that the optimal target predictor can be non-parametrically identified with the help of concept and proxy variables available only in the source domain, and unlabeled data from the target. The identification results are constructive, immediately suggesting an algorithm for estimating the optimal predictor in the target. For continuous observations, when this algorithm becomes impractical, we propose a latent variable model specific to the data generation process at hand. We show how the approach degrades as the size of the shift changes, and verify that it outperforms both covariate and label shift adjustment.

1 INTRODUCTION

Distribution shift is a fact of many real-world machine learning systems. For example, imagine we have trained a prediction model on patients of hospital P and would like to apply it to patients of hospital Q. However, these hospitals differ in their patient populations along socioeconomic, demographic, and other axes (Finlayson et al., 2021). How can we find the optimal predictor for hospital Q, given only labelled data from hospital P and unlabelled data from hospital Q? This is the problem of unsupervised domain adaptation (Huang et al., 2006). Without any assumptions on the shift, this question is impossible to answer: the mapping from features X to labels Y could differ across hospitals in arbitrary ways. To address this, approaches typically assume that certain observed distributions are preserved across the shift, *covariate shift*: $p(Y \mid X) = q(Y \mid X)$ (Shimodaira, 2000) or, *label shift*: $p(X \mid Y) = q(X \mid Y)$ (Gart and Buck, 1966), where p, q are distributions of hospitals P, Q.

However, these assumptions are often restrictive for real-world settings, as the shifts encountered are typically more complex (e.g., 'compound' shifts (Schrouff et al., 2022)). Here, we focus on one such shift that we call *latent subgroup shift*. Subgroup shift occurs when both the source P and target Q distributions are composed of a common set of subgroups $U \in \mathcal{U}$, but the prevalence of these subgroups differs, i.e., $p(U) \neq q(U)$. The subgroup shift is latent if these subgroups are unobserved in both P and Q. Importantly, the relationships between features X and labels Y can differ between subgroups, such that neither the discriminative distribution $p(Y \mid X)$ nor the generative distribution $p(X \mid Y)$ is preserved across the shift. In healthcare settings, these subgroups may differ in their exposure to social

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

Authors listed in alphabetical order

Email any correspondences to m.kusner@ucl.ac.uk, alexdamour@google.com

[1]Work completed while at Google Research

[2]Now at Deepmind

determinants of health, contributing to differences in health outcomes and patterns of comorbidity, care access, delivery, and treatment (Marmot and Wilkinson, 2005).

To tackle latent subgroup shift, we frame learning the optimal $q(Y \mid X)$ as an identification problem. Our identification strategy combines approaches from proximal causal inference (originally designed to identify intervention distributions $p(Y \mid do(X))$ under unobserved confounding using proxy variables) (Kuroki and Pearl, 2014), black box label shift adaptation (Lipton et al., 2018), and concept bottleneck modeling (Koh et al., 2020). We show that it is possible to express $q(Y \mid X)$ in terms of the joint distribution of observables in P and the distribution of unlabeled inputs X in Q. We derive two identification results, one for discrete data and another for continuous data. While the results are constructive, immediately implying an algorithm, estimation requires non-trivial density estimation. Therefore, we describe an alternative approach that leverages stable latent variable (Kingma and Welling, 2013) models to estimate $q(Y \mid X)$. Our approach answers an open question on how to leverage advances in concept bottleneck models (Koh et al., 2020) for distribution shifts in both X and Y. Further, it allows one to learn a single model in the source P which can then be adapted to arbitrary shifts in Q.

Contributions. We propose a new approach for adaptation to latent distribution shifts given concepts and proxies, for cases where existing adaptation methods often fail. We formally identify the target distribution for both discrete and continuous variables, then propose effective estimators. We perform a sensitivity analysis that characterizes how our method changes when shift size and proxy strength are varied. We show that our approach outperforms multiple baselines including covariate and label shift techniques. We provide code to reproduce experiments at https://github.com/google-research/google-research/tree/master/latent_shift_adaptation.

Notation. We denote scalars and functions by lowercase letters (e.g., a, $a(\cdot)$), vectors by bold lowercase ($\mathbf{a}$), random variables by capital letters (A), matrices by bold, capital letters ($\mathbf{A}$), and sets by caligraphic, capital letters ($\mathcal{A}$). Let $[n]$ denote the set $\{1, 2, \ldots, n\}$.

2 RELATED WORK

There has been a flurry of recent work on improving out-of-distribution generalization (see Shen et al. (2021), Wang et al. (2022), and Zhou et al. (2022) for three recent surveys). Largely, this work can be divided into two camps: (a) work that learns a single model to work well across shifts, such as work on invariant predictors (Arjovsky et al., 2019) and, (b) work that adapts a model from a source distribution to a target distribution, given access to limited data in the target. Here we focus on the second class of approaches.

Model adaptation. To obtain an optimal predictor in a new distribution Q, one of the most popular assumptions is to localize the shift between distributions P and Q in the features (covariates) X, i.e., *covariate shift*: $p(X) \neq q(X)$. There has been a large body of work devoted to estimating predictors for Q under this setting (Shimodaira, 2000; Zadrozny, 2004; Huang et al., 2006; Gretton et al., 2009; Bickel et al., 2009; Sugiyama and Kawanabe, 2012; Chen et al., 2016; Schneider et al., 2020). The key assumption in this line of work is that $p(Y \mid X) = q(Y \mid X)$. Therefore, if one makes the source data appear like the target data (e.g., by reweighing the source classifier loss by $q(X)/p(X)$), one can learn an accurate target classifier. The other popular assumption is to localize the shift in the labels Y, i.e., *label shift*: $p(Y) \neq q(Y)$ and $p(X \mid Y) = q(X \mid Y)$ (Gart and Buck, 1966; Manski and Lerman, 1977; Rosenbaum and Rubin, 1983; Saerens et al., 2002; Forman, 2008; Storkey, 2009; du Plessis and Sugiyama, 2012; Zhang et al., 2013; Lipton et al., 2018; Azizzadenesheli et al., 2019; Alexandari et al., 2020; Garg et al., 2020; Tachet des Combes et al., 2020; Wu et al., 2021). Here one can use a similar approach: learn $q(Y)/p(Y)$ and use it to reweigh a source classifier, adapting it to the target distribution. The assumptions of covariate and label shift can be framed as criteria on the causal structure of the data, shown in Figure 1(a)-(b) (Schölkopf et al., 2012). Most theoretical work is on generalization error bounds for covariate shift (Sugiyama and Mueller, 2005; Ben-David et al., 2006; Mansour et al., 2009; Ben-David et al., 2010; Cortes and Mohri, 2011; Johansson et al., 2019) and label shift (Gong et al., 2016).

Causality for domain shift. Recently, a line of work has framed domain shift using causal methods (Zhang et al., 2015; Magliacane et al., 2018; Gong et al., 2018; Chen and Bühlmann, 2020; Teshima et al., 2020). Most related to our approach is the work of Yue et al. (2021). Similar to our setup, they describe a setting where an unobserved latent confounder U shifts the distribution of X and Y. However, different from our work, they target an interventional distribution instead of $q(Y \mid X)$. To do so they learn mappings from $X \sim P$ to $X \sim Q$, and vice-versa. They use these mappings, as well as a variational autoencoder (Kingma and Welling, 2013), to generate two 'proxies', one for X and Y. They assume these proxies are caused by U, and they use the result of Miao et al. (2018) to identify an invariant 'bridge function' to remove the effect of the latent shift. However, this does not guarantee identification of the structural equations mapping U to the proxies, X, and Y, which is necessary for the procedure to correct for U.

3 SETUP AND PRELIMINARIES

Let P be the source distribution and Q be the target, with probability mass/density functions p and q. Our goal is to identify the optimal predictor of Y from X in the target: $q(Y \mid X)$. To do so, we will make two main assumptions. First, to make progress in this setting, we assume that we

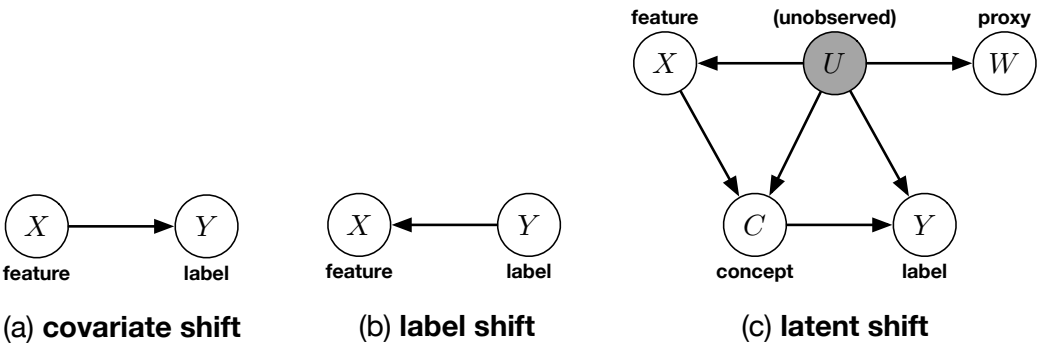

Figure 1: Different domain shift assumptions: (a) $p(X) \neq q(X)$, (b) $p(Y) \neq q(Y)$, (c) $p(U) \neq q(U)$.

have access to some auxiliary variables that play key roles in the source distribution.

A1. *We also observe auxiliary variables C (concept bottleneck) and W (proxy). All data is generated by the process described in Figure 1(c) and is* faithful *and* Markov *(Spirtes et al., 2000) (i.e., conditional independences in the data exist iff they exist in the graph). Crucially, we only observe (X, C, Y, W) in the source P and X in the target Q.*

Formally, the data generation process of Figure 1(c) is a probabilistic graphical model (Pearl, 1988). Given a set of observed variables $\mathcal{V}$ and unobserved variables $\mathcal{U}$, these models define a functional relationship f_i between each $V_i \in' ! \mathcal{V}$ and the variables that generate V_i (also called its direct *parents*) $\mathcal{V}_{\text{pa}(i)}, \mathcal{U}_{\text{pa}(i)}$ i.e., $V_i = f_i(\mathcal{V}_{\text{pa}(i)}, \mathcal{U}_{\text{pa}(i)})$. These relationships can be described by a directed acyclic graph (DAG), e.g., as in Figure 1. A key aspect of these models is that they encode conditional independence relationships between variables in $\mathcal{V}, \mathcal{U}$, that can be derived via d-separation (Pearl et al., 2000). Throughout this work we assume f_i and $\mathcal{U}$ are unknown.

The additional auxiliary variables C and W play specific roles in this graph. C operates as a "concept bottleneck" that mediates the dependence between X and Y within subgroups indexed by U. Meanwhile, W operates as an independent proxy, or noisy observation, of U that is conditionally independent of all other variables. Both of these properties play key roles in our identification strategy.

Our second assumption defines latent subgroup shift.

A2. *The shift between P and Q is located in U, i.e., there is a* latent shift $p(U) \neq q(U)$, *while* $p(\mathcal{V} \mid U) = q(\mathcal{V} \mid U)$, *where* $\mathcal{V} \subseteq \{W, X, C, Y\}$.

Under these assumptions, distributions on U or that have U marginalized (i.e., all observed distributions $p(\mathcal{V}) \neq q(\mathcal{V})$ for $\mathcal{V} \subseteq \{W, X, C, Y\}$) will shift between P and Q, whereas only distributions conditional on U do not shift. This is a direct generalization of the covariate shift invariance, in which $U \rightarrow X \rightarrow Y$ and the label shift invariances, in which $U \rightarrow Y \rightarrow X$.

Our framework is inspired by (a) concept bottleneck models (Koh et al., 2020) and (b) identification via proxies (Kuroki and Pearl, 2014). We briefly review these topics next.

Concept bottleneck models. Data in certain settings may contain information beyond features and labels. For instance, in healthcare it is common to not only have raw electronic health record data X (e.g., temperature, blood cultures, ...) and disease labels Y, but also physician summaries C such as the presence and spread of infection. Koh et al. (2020) formalize this learning setup, calling C *concepts*. In general, concepts C are high-level, often interpretable, pieces of information that mediate the relationship between X and Y. Prior works have used concepts for diagnosing model failures and for covariate shift (Kumar et al., 2009; Lampert et al., 2009; Koh et al., 2020; Chen et al., 2020; Mahinpei et al., 2021). The concept bottleneck model (Koh et al., 2020) was shown to be robust to covariate distribution shifts; here, we show with the appropriate adjustment strategy, such models can also be adapted to subgroup shifts. Another line of work have incorporated concepts into causal models to improve model explanations (Goyal et al., 2019; Bahadori and Heckerman, 2021).

Proxies. Our work leverages results in causal effect estimation with proxy variables (Kuroki and Pearl, 2014; Miao et al., 2018). In these works, W is a proxy of U that allows one to identify the causal effect of C on Y in Figure 1(c). In our running example, a useful W would be the region where a patient lives as this is often a proxy for SDH quantities, such as income U.

4 IDENTIFICATION UNDER LATENT SHIFT

In this section, we report identification results for the optimal target distribution predictor $q(Y \mid X)$ given observed draws from $p(X, C, Y, W)$ and $q(X)$. We first present our central adjustment strategy in the case where U is observed in the source distribution. We then show that, when C and W are observed in the source distribution, we can use this strategy even in cases where U is unobserved. We consider two such cases: one where all observed variables are dis-

crete, and another where X and W are continuous. In these latter two cases, the key challenge is to show that the distributions in our adjustment formula, which involve U, can be identified in the source domain.

4.1 Subgroup Adjustment Formula

To begin, we present our central adjustment formula, considering the case where U is observed in the source distribution P, but not in the target distribution Q. We derive the formula by decomposing our target $q(Y \mid X)$, leveraging A2 and Figure 1(c):

$$\begin{aligned} q(Y|X) &\overset{(a)}{=} \sum_{i=1}^{k_U} q(Y|X, U=i) q(U=i|X) \\ &\overset{(b)}{=} \sum_{i=1}^{k_U} p(Y|X, U=i) \frac{q(X|U=i) q(U=i)}{q(X)} \\ &\overset{(c)}{=} \sum_{i=1}^{k_U} p(Y|X, U=i) \frac{p(U=i|X) p(X) q(U=i)}{p(U=i) q(X)} \\ &\overset{(d)}{\propto} \sum_{i=1}^{k_U} p(Y|X, U=i) p(U=i|X) \frac{q(U=i)}{p(U=i)} \end{aligned} \tag{1}$$

The first equality (a) is given by the chain rule and marginalization. The second (b) is given by A2: since $q(Y|X, U=i)$ conditions on U, we have $q(Y|X, U=i) = p(Y|X, U=i)$. The fractional term is given by Bayes rule. The equality (c) is again given by A2 and Bayes rule: $q(X|U=i) = p(X|U=i) = p(U=i|X)p(X)/p(U=i)$. The proportional (d) is given by the fact that $p(X)/q(X)$ is constant as the left-hand side conditions on these variables.

When U is observed under P, all quantities on the final right hand side are directly estimable except $q(U)/p(U)$, because U is not observed under Q. Interestingly, this parallels the label shift problem, where distributions conditional on Y are preserved across the distribution shift, but Y is not observed under Q. In fact, the same label shift adaptation identification arguments and techniques can be applied to adjust for U instead! Here, we adapt the method-of-moments identification argument made in Lipton et al. (2018). For any function $f(X)$, the identity $q(f(X)) = \sum_{i=1}^{k_U} q(f(X) \mid U=i) q(U)$ can be expanded (using Bayes rule and A2):

$$\frac{q(f(X))}{p(f(X))} = \sum_{i=1}^{k_U} p(U=i \mid f(X)) \frac{q(U=i)}{p(U=i)}. \tag{2}$$

These equations define a linear system, and, for appropriate choices of $f(X)$ and rank conditions on $p(U=i \mid X)$ (see A4 and A6 below), we can solve for $q(U=i)/p(U=i)$. For example, Lipton et al. (2018) define $f(X)$ as the decision function of a classifier; in that case the linear system can be written in terms of the confusion matrix of the classifier. Garg et al. (2020) discuss other choices, as well as maximum likelihood approaches to learning this likelihood ratio. Upon solving (2), $q(Y \mid X)$ is identified by (1).

Remark 1. This "observed U" setting is a simplification of the general latent subgroup shift problem, but may be of independent interest. In many applications, especially when U includes sensitive demographic categories, the subgroup label may be collected at training time, but unavailable at deployment time. In such cases, this identification argument would be sufficient for domain adaptation.

Remark 2. The identifying expression (1) enables adaptation to new distributions Q without retraining any models under P. To adapt to a new distribution, we plug in a new estimate of $q(f(X))$ to (2), then evaluate (1) at the solution. This *post hoc* property applies to all identification strategies we discuss.

4.2 The Error of Covariate/Label Adjustment

What if we apply covariate or label shift adjustment to the latent subgroup shift setting?

Covariate shift adjustment. Assume data follows the latent shift setting of Figure 1(c), but we (falsely) believe that the shift between the observed data in P, $\{X, C, W, Y\}$, and that of Q, $\{X'\}$, is due to covariate shift. The covariate shift assumption implies that $p(Y|X) = q(Y|X)$. Given this, we would start by training a model $f : X \to Y$ on the data in P which estimates $P(Y|X)$. We would then use this model on the data X' in Q as an estimate $q(Y|X)$ (we would only use X to train f, and not (C, W), as we only see X' in Q). However, regardless of the amount of data in P and X there would always be an error between $f(X) := p(Y|X)$ and $q(Y|X)$. Specifically, at the population level, the (squared) error under latent shift is:

$$\begin{aligned} &(p(Y|X) - q(Y|X))^2 \\ &\quad = \Big(\sum_u p(Y|X, u) \big[p(u|X) - q(u|X) \big] \Big)^2 \\ &\quad = \left(\sum_u p(Y|X, u) p(u|X) \left[1 - \frac{p(X)}{q(X)} \frac{q(u)}{p(u)} \right] \right)^2. \end{aligned}$$

Label shift adjustment. Imagine we instead assumed the shift was due to label shift which implies $p(X|Y) = q(X|Y)$. Given this, $q(Y|X)$ could be written as:

$$\begin{aligned} q(Y|X) &= q(X|Y) \frac{q(Y)}{q(X)} = p(X|Y) \frac{q(Y)}{q(X)} \\ &= p(Y|X) \frac{p(X)}{q(X)} \frac{q(Y)}{p(Y)}. \end{aligned}$$

All of the terms on the right hand side are estimable, even $q(Y)/p(Y)$. Specifically, given a trained model $f : X \to Y$ on the data in P (estimating $p(Y|X)$), we can estimate $q(Y)/p(Y)$ using a label shift correction

technique. For example, Lipton et al. (2018) shows that $q(f(X)) = \sum_y p(f(X), y)[q(y)/p(y)]$. However, this adjusted estimate also incurs error with respect to the optimal target $q(Y|X)$ in the latent shift setting. The population (squared) error under latent shift is:

$$\left(p(Y \mid X)\frac{p(X)}{q(X)}\frac{q(Y)}{p(Y)} - q(Y \mid X)\right)^2$$
$$= \left(\sum_u p(Y \mid X, u)p(u \mid X)\frac{p(X)}{q(X)}\left[\frac{q(Y)}{p(Y)} - \frac{q(u)}{p(u)}\right]\right)^2.$$

4.3 Discrete Observations

We now state sufficient conditions for identification of $q(Y|X)$ in the latent shift setting. To begin we assume all observable variables $\{X, C, Y, W\}$ are discrete.

A3. *$U \in [k_U]$ is discrete s.t., $k_X, k_W \geq k_U$ (recall k_X, k_W are the number of categories of (discrete) X, W).*

Generally, identification requires some restrictions on how U influences the observed variables $\{W, X, C, Y\}$. The above places such a restriction more generically than restriction functional forms; all we require is that the support of U is smaller than that of observed variables X, W.

A4. *For every $i \in [k_U]$ where $q(U = i) > 0$ we have $p(U = i) > 0$, all linear systems have rank at least k_U, and $p(Y|C, U = 1) \neq p(Y|C, U = 2) \neq \cdots \neq p(Y|C, U = k_U)$ P-almost everywhere.*

The first condition ensures that $q(U = i)/p(U = i)$ is bounded for all i. The remaining two conditions are inherited from Kuroki and Pearl (2014): they ensure that inverses exist and that eigenvectors are unique. Essentially, they require that all variables depend non-trivially on U. Overall these assumptions are of two types: (1) **Structural**: A1 and A2 describe how the data and shifts are structured; (2) **Functional**: S3 and A4 detail conditions on the functions that generate data.

Our main result for discrete data is the following.

Lemma 1. *Given A1–A4, all probability mass functions over discrete $\{W, X, C, Y, \widetilde{U}\}$ in the source P are identifiable, where $\widetilde{U}$ is an unknown permutation of U.*

Theorem 1 (Identifiability for Discrete Observations)**.** *The distribution $q(Y|X)$ is identifiable from discrete $\{W, X, C, Y, \widetilde{U}\} \sim P$ and $X \sim Q$.*

Proof sketches. We give full proofs in the Appendix and give sketches here. The first key observation for Theorem 1 is that all of the steps (a)–(d) in eq. (1) hold when U is replaced with the permutation $\widetilde{U}$. This is because (a) $\widetilde{U}$ satisfies the same independence conditions as U, and (b) $q(Y|X)$ only requires marginalizing over U, making the order of the categories of U irrelevant to identification. Given Lemma 1, the only step remaining is to solve (2) in terms of $\widetilde{U}$. A3 and A4 ensure that the system has a solution.

The proof of Lemma 1 works in two stages: 1. It first demonstrates that $p(W|\widetilde{U})$ can be identified, and 2. It shows that once $p(W|\widetilde{U})$ is identified, all distributions on $W, X, C, Y, \widetilde{U}$ are identified. Stage 1 is done by proving a variation of a result given by Kuroki and Pearl (2014). They demonstrate that when $k_W = k_X = k_U$ and data is generated from the graph of Figure 1(c) then it is possible to identify the causal effect $p(Y|do(C))$ (in Theorem 1 (Kuroki and Pearl, 2014)). Identifying $p(Y|do(C))$ only requires identifying specific distributions involving $\widetilde{U}$, in order to remove its contribution to Y, i.e., $p(Y|do(C)) = \sum_{x,u} P(Y|C, X = x, \widetilde{U} = u)P(X = x, \widetilde{U} = u)$. However, as we show by construction, the result of Kuroki and Pearl (2014) is stronger. In Stage 1, we recover $p(W|\widetilde{U})$ for Figure 1 (c) by contrasting the distributions $p(X, W \mid c)$ and $p(y, X, W \mid c)$. Specifically, $p(W|\widetilde{U})$ can be recovered from the eigendecomposition of $\mathbf{A}^{-1}\mathbf{B}$ where, for fixed values of y and c, these matrices are as follows,

$$\underbrace{\begin{bmatrix} 1 & p(w_1|c) & \cdots & p(w_{k_W-1}|c) \\ p(x_1|c) & p(x_1, w_1|c) & \cdots & p(x_1, w_{k_W-1}|c) \\ \vdots & \vdots & \ddots & \vdots \\ p(x_{k_X-1}|c) & p(x_{k_X-1}, w_1|c), & \cdots & p(x_{k_X-1}, w_{k_W-1}|c) \end{bmatrix}}_{\mathbf{A}}$$

$$\underbrace{\begin{bmatrix} p(y|c) & p(y, w_1|c) & \cdots & p(y, w_{k_W-1}|c) \\ p(y, x_1|c) & p(y, x_1, w_1|c) & \cdots & p(y, x_1, w_{k_W-1}|c) \\ \vdots & \vdots & \ddots & \vdots \\ p(y, x_{k_X-1}|c) & p(y, x_{k_X-1}, w_1|c) & \cdots & p(y, x_{k_X-1}, w_{k_W-1}|c) \end{bmatrix}}_{\mathbf{B}}.$$

In the above w_1 is shorthand for $W = 1$ (similarly for X). In Stage 2, we identify all distributions involving $\widetilde{U}$. The key observation behind this second result is that conditioning on $\widetilde{U}$ d-separates W from the rest of the observed variables. Thus, factorizing observed distributions using $\widetilde{U}, W$ can form linear systems. In these systems, the unknown distributions involving $\widetilde{U}$ can be recovered by some function of $p(W|\widetilde{U})$ (identified in Stage 1) and observables.

Estimation. As both proofs are constructive, we can immediately use them to design an approach to estimate $q(Y|X)$. This is shown in Algorithm 1.

4.4 Continuous Observations

We now consider the case where W, X, C, Y are continuous. This setting turns out to be more challenging, as, unlike in the discrete case, we cannot enumerate all of the states and apply finite dimensional eigendecomposition to estimate the associated probability mass functions. Instead, we must apply functional analysis tools to estimate nonparametric continuous probability density functions, which require more care to ensure existence and estimability. To this end, we make the following assumptions.

Algorithm 1 Estimating $q(Y|X)$.

Require: source $\mathcal{P} = \{(w_i, x_i, c_i, y_i)\}_{i=1}^n$; target $\mathcal{Q} = \{x_j\}_{j=1}^m$; For any variables $G \in [k_G], H \in [k_H]$ let $p(\mathbf{G}|\mathbf{H})$ be a $k_G \times k_H$ matrix of probabilities s.t. $p(\mathbf{G}|\mathbf{H})_{ij} = p(G{=}i|H{=}j)$

1: Using $\mathcal{P}$, form matrices $\mathbf{A}, \mathbf{B}$ described in eq. (3)
2: Decompose $\mathbf{A}^{-1}\mathbf{B} = \mathbf{S}^{-1}\Lambda\mathbf{S}$ to get $p(\mathbf{W}|\widetilde{\mathbf{U}})$ via $\mathbf{S}^{-1}$
3: Compute $p(\widetilde{\mathbf{U}}|\mathbf{X}) = p(\mathbf{W}|\widetilde{\mathbf{U}})^{-1}p(\mathbf{W}|\mathbf{X})$
4: Compute $q(\widetilde{\mathbf{U}})/p(\widetilde{\mathbf{U}}) = p(\widetilde{\mathbf{U}}|\mathbf{X})^{-1}[q(\mathbf{X})/p(\mathbf{X})]$
5: Compute

$$p(\mathbf{Y}|X, \widetilde{\mathbf{U}}) = p(\mathbf{Y}|X, \mathbf{W})\left(\frac{p(\mathbf{W}|\widetilde{\mathbf{U}}) \circ p(\widetilde{\mathbf{U}}|X)}{p(\mathbf{W}|X)}\right)^{-1}$$

6: Compute

$$q(\mathbf{Y}|x_j) \propto p(\mathbf{Y}|x_j, \widetilde{\mathbf{U}})\left[p(\widetilde{\mathbf{U}}|x_j) \circ \frac{q(\widetilde{\mathbf{U}})}{p(\widetilde{\mathbf{U}})}\right], \forall x_j \in \mathcal{Q}.$$

A5. There exists a $c \in \mathrm{Dom}(C)$, such that $p(X \mid U = i, c), p(X \mid U = j, c)$ are linearly independent for all $(i, j) \in [k_U]$ for $i \neq j$. Similarly, $p(W \mid U = i), p(W \mid U = j)$ are linearly independent for all $(i, j) \in [k_U]$ for $i \neq j$.

This assumption allows us to identify the distributions of $p(W \mid U)$ and $p(X \mid U, C)$, which are crucial to the eigendecomposition technique.

A6. There exist distinct points $x_1, \ldots, x_{k_U} \in \mathrm{Dom}(X)$ such that the matrix $[p(U = j \mid x_i)]_{i,j} \in \mathbb{R}^{k_U \times k_U}$ is invertible.

This assumption ensures that the $q(U)/p(U)$ system in eq. (2) has a unique solution. Note this assumption is very weak for continuous X, e.g., $x_1, \ldots, x_{k_U}$ can be chosen to be exemplars of each class $i \in [k_U]$.

With A5, A6 replacing A3, we extend the identification result from Theorem 1 to continuous data.

Theorem 2 (Continuous Observations)**.** *Given A1, A2, A4–6, the distribution $q(Y|X = x)$ is identifiable from continuous $\{W, X, C, Y\} \sim P$ and $x \in X \sim Q$.*

We give a full proof in the Appendix. The steps are similar to the discrete observation case: set up a linear system, eigendecompose it, recover $p(W|\widetilde{U})$ from the eigenvectors, and use $p(W|\widetilde{U})$ to identify all quantities on the right-hand side of eq. (1). However, the specifics of the continuous setting require more technical tools.

Estimation. Implementing a plug-in estimator from Theorem 2 is challenging, as it requires non-parametric conditional density estimation and an eigendecomposition over functions. We implement such an approach, and describe it in detail in the Appendix.

5 ROLES OF CONCEPTS AND PROXIES

Do we really need C and W? And why can't we have additional edges in Figure 1(c), e.g. $X \to Y$? We describe here why the "concept bottleneck" and "proxy" properties of W and C are essential to our identification strategy. Specifically, we discuss at a high level why generalizing the graph by removing observed nodes or adding edges prevents non-parametric identification of simpler causal quantities. While these are not necessary conditions, they are nearly as general as those used in non-parametric identification results in causal inference literature (Miao et al. (2018); Lee and Bareinboim (2021) also allow edge $W \to Y$).

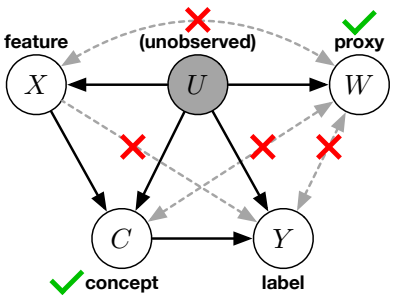

Figure 2: Removing C, W or adding any of the dotted edges prevents non-parametric identification of the full joint distribution $p(\mathcal{V}, \widetilde{U})$ via our approach.

Can C and/or W be removed? Removing C corresponds to the setting of Pearl (2010), where the goal is to estimate $p(Y|do(X))$. This work assumes one can either: (a) observe U without error in a subpopulation (Selén, 1986; Greenland and Lash, 2008), (b) observe $p(W|U)$ (Pearl, 2010), or (c) place a prior distribution on the parameters of $p(W|U)$ to bound $p(Y|do(X))$ (Greenland, 2005). However, these techniques are non-trivial when U is complex. Here we will not assume that it is possible to observe $U, p(W|U)$ or derive a prior for $p(W|U)$. Keeping C but removing W leads to a generalization of the front-door graph (Pearl et al., 2000) for which causal effects are not non-parametrically identifiable. If we remove both C and W, we can only identify $p(Y|do(X))$ if U is observed, an assumption called 'ignorability' (Imbens and Rubin, 2015).

Can we remove/add any additional edges? First note that if we remove edges from our assumed graph this limits the possible data distributions that it could have generated. This is because when edges are removed, conditional independences may be introduced. For example, if we remove the edge from $U \to C$ then $W \perp\!\!\!\perp C \mid X$, which is not the case for our original graph in Figure 1 (c). Another way to see this is that we can recover the covariate shift graph of Figure 1 (a) from ours if we remove all edges starting from U, then remove $X \to C$, and finally relabel C as X. Recall that the covariate shift graph implies $p(Y|X) = q(Y|X)$ which does not hold in our original graph. What about adding edges? Identifying $p(W|U)$ (i.e., Stage 1 in the proof of Lemma 1) requires that both $W \perp\!\!\!\perp \{X, C, Y\} \mid U$ and $Y \perp\!\!\!\perp \{W, X\} \mid \{U, C\}$. The first conditional independence is broken if there are any arrows from X, C, Y to or

from W. We do not prove here that this is necessary, but we suspect that it is: currently the only edge that can be added for identifying the simpler causal quantity $p(Y|do(C))$ is $W \to Y$ (Miao et al., 2018; Lee and Bareinboim, 2021). Crucially, these methods do not identify $p(W|U)$. The only other edge that could be added to the graph and it still be a DAG is $X \to Y$. However, this would break the second conditional independence statement as it would make $Y \not\perp\!\!\!\perp X \mid \{U, C\}$. This edge would also render the causal effect unidentifiable under the most generic non-parametric methods (Lee and Bareinboim, 2021).

6 ESTIMATION WITH LATENT VARIABLE MODELS

Algorithm 1 and its associated continuous version (described in the Appendix C) become impractical as the dimension increases (due to the need for probability mass/density estimation). Here, we propose an alternative approach based in deep latent variable modelling that can be useful for adapting to latent subgroup shifts with high-dimensional data. Note that the identification arguments in the previous section imply that any joint distribution $p(\widetilde{U}, C, X, Y, W)$ that satisfies our assumptions and matches the observed marginal distribution $p(C, X, Y, W)$ can be used to identify $q(Y \mid X)$. We propose approximating such a joint distribution using a model based on the Wasserstein Auto-Encoder (WAE; Tolstikhin et al., 2018). In this section, we describe modifications to the standard WAE to customize the learned joint distribution to our assumptions.

Formally, we approximate the true posterior $p(\widetilde{U}|X, C, Y, W)$ with a recognition model or encoder $\widehat{p}(\widetilde{U}|X, C, Y, W)$ with parameters ϕ. Given observed variables $\mathcal{V} = \{X, Y, C, W\}$, reconstruction loss ℓ, decoder f with parameters θ, divergence D, and prior distribution $\overline{p}(\widetilde{U})$, the form of the training objective is

$$\min_{\phi,\theta} \mathbb{E}_{p(\mathcal{V})} \mathbb{E}_{p(\widetilde{U}|\mathcal{V})} \big[\ell(\mathcal{V}, f(\widetilde{U}))\big] + D(\widehat{p}(\widetilde{U}) \,||\, \overline{p}(\widetilde{U})). \quad (3)$$

To encourage the inference network to learn a posterior distribution that conforms to Figure 1(c) we impose the following factorization on the joint probability

$$p(\mathcal{V}, \widetilde{U}) = p(Y|C, \widetilde{U})p(C|X, \widetilde{U})p(X|\widetilde{U})p(W|\widetilde{U})p(\widetilde{U}).$$

Given this, the reconstruction (log) loss decomposes

$$\begin{aligned}\ell(\mathcal{V}, f(\widetilde{U})) \; &= \beta_Y \ell_Y(Y, f_Y(C, \widetilde{U})) + \beta_C \ell_C(C, f_C(X, \widetilde{U})) \\ &+ \beta_X \ell_X(X, f_X(\widetilde{U})) + \beta_W \ell_W(W, f_W(\widetilde{U})).\end{aligned}$$

where the above subscripts indicate variable-specific decoders, loss functions, and scalar hyperparameter weights β. As $\widetilde{U}$ is discrete, to allow training with the reparameterization trick we model $\widehat{p}(\widetilde{U}|X, C, Y, W)$ using a Gumbel-Softmax distribution (Jang et al., 2016; Maddison et al., 2016). We set the prior $\widetilde{p}(\widetilde{U})$ to be a uniform categorical distribution over the categories of $\widetilde{U}$.

Given a trained WAE model, we can generate joint samples $\{(x_i, c_i, y_i, w_i, \widetilde{u}_i)\}_{i=1}^n$ by the encoder $\widehat{p}(\widetilde{U} \mid X, C, Y, W)$. Lemma 1, which establishes identification of this joint distribution under our assumptions, provides some justification for this approach. All that remains to estimate are $p(\widetilde{U}|X), q(\widetilde{U})/p(\widetilde{U}), p(Y|X, \widetilde{U})$ and Equation (5). Each of these is readily estimable using standard classification models, as we have joint samples. We discuss our implementation of this estimation strategy in the Appendix.

7 SIMULATION STUDY

We now describe demonstrate our identification results in a simulated numerical examples. These examples serve as a proof of concept that our identification strategies can serve as the basis for estimation methods. In particular, we aim to show that (a) plug-in estimators based on our constructive proofs can be used to estimate $q(Y \mid X)$ in simple contexts, and (b) modifying deep latent variable models to respect the conditional independence structure in our setting can be an effective strategy for estimation in more complex settings. We also show that estimators based on our adjustment strategy can succeed where standard covariate shift and label shift adaptation techniques, or naive applications of latent variable models, fail.

The simulations are structured as follows. We have one source distribution P, and several target distributions Q, generated by latent subgroup shifts. We train several models on the source distribution, some of which use unlabeled examples from Q for adaptation, then measure their performance on the target distribution. In each case, we compare performance to two endpoints: the performance of an unadapted model trained by ERM on the source (ERM-SOURCE), which should be a lower bound on performance, and an oracle model trained directly on data from the target distribution (ERM-TARGET), which should be an upper bound. We also compare to an oracle model that adjusts for U using (1), as if it were observed (LSA-ORACLE).

For these simulations, we fix a set of parameters that instantiate a case where standard empirical risk minimization (ERM-SOURCE) fails in a predictable way, while oracle adjustments for U (LSA-ORACLE) recover the optimal target predictor $q(Y \mid X)$. We do so by constructing a setting where the subgroup specific conditional expectation $E[Y \mid X, U]$ is sufficiently different across subgroups, thus producing a different ordering of predictions over examples from the target $q(Y \mid X)$. Furthermore, we ensure that neither U nor Y can be perfectly reconstructed from X. If either were the case $p(Y \mid X) = p(Y \mid X, U) = q(Y \mid X, U) = q(Y \mid X)$, and the optimal predictor under Q would simply correspond to the optimal predictor under P. We then evaluate several estimation approaches based on

Table 1: Results of discrete simulation study ($\alpha_w = 1$, $n = 10^4$, $p(U=1) = 0.1$, $q(U=1) = 0.9$). Results shown are the RMSE between estimated and true $q(Y \mid X)$ across categories of discretized X.

	RMSE
$p(Y\|X)$	0.194
ours	**0.056**
$q(Y\|X)$	0.004

our identification strategy (from which U is hidden).

We sample datasets of size 10,000, and divide training, validation, and test sets into 70%, 20%, and 10% splits. For all experiments, we consider a fixed setting for the source distribution such that $p(U=1) = 0.1$. The target distribution varies over a range of settings of $q(U=1) \in \{0.1, 0.2, \ldots, 0.9\}$. Further details regarding the experimental procedure are provided in Appendix B.

To evaluate the discrete eigendecomposition approach (Algorithm 1), we first apply K-means with two clusters to discretize X. The results in Table 1 verify that the algorithm is capable of improving on estimates derived from the source domain in a setting where the magnitude of the distribution shift is large ($p(U=1) = 0.1$ vs. $q(U=1) = 0.9$) and W is a noisy proxy of U ($\alpha_w = 1$).

For the case where X is continuous, we compare the proposed adaptation approach to alternatives. In the main text, we primarily evaluate performance using the area under the ROC curve (AUROC), but include analagous results in the appendix for the cross-entropy loss and accuracy (Supplementary Tables 3 and 4). In a setting analogous to the experiment conducted in the discrete case (Table 2; $p(U=1) = 0.1$, $q(U=1) = 0.9$, $\alpha_w = 1$)), models learned with ERM on the source domain (ERM-SOURCE) using a multilayer perceptron perform poorly in the target domain relative to those learned in the target domain (ERM-TARGET). Furthermore, standard approaches to accounting for distribution shift, including covariate shift weighting (COVAR; Shimodaira (2000)), label shift weighting (LABEL; weighting by oracle $q(Y)/p(Y)$), and black box shift estimation (BBSE; Lipton et al. (2018)) do not outperform ERM-SOURCE. However, we note that the latent shift adaptation approach with oracle access to U (LSA-ORACLE; (1)) is able to perform on-par with ERM-TARGET without access to labeled data in the target domain. Our main WAE-based approach that leverages the structured decoder and reconstruction loss (LSA-WAE-S) does not match LSA-ORACLE, but does partially mitigate the gap in performance between ERM-SOURCE and ERM-TARGET. We compare to an alternative WAE specification that does not leverage a structured decoder (LSA-WAE-V) and find that it is does not improve on ERM-SOURCE. This highlights the key role played by that the structural properties of the auxiliary variables C and W.

Table 2: Results of continuous simulation study ($\alpha_w = 1$, $n = 10^4$, $p(U=1) = 0.1$, $q(U=1) = 0.9$), mean $\pm$ std AUROC over 10 random training replicates.

Method	Source	Target
ERM-SOURCE	0.9560 ± 0.0001	0.6856 ± 0.0010
COVAR	0.9113 ± 0.0216	0.3274 ± 0.1351
LABEL	0.9561 ± 0.0001	0.6848 ± 0.0014
BBSE	0.9550 ± 0.0001	0.6789 ± 0.0005
LSA-WAE-S	0.9429 ± 0.0083	$\mathbf{0.8131 \pm 0.0365}$
LSA-WAE-V	0.9550 ± 0.0006	0.6730 ± 0.0138
LSA-ORACLE	0.7843 ± 0.0254	0.9167 ± 0.0012
ERM-TARGET	0.7611 ± 0.0011	0.9194 ± 0.0001

We further evaluate the proposed WAE approach over varying degrees of distribution shift and levels of noise in the proxy variable W, and compare it to the continuous eigendecomposition (spectral) method (appendix C) suggested by the proof of Theorem 2. We observe that ERM-SOURCE performance degrades smoothly as a function of the degree of distribution shift (Figure 3). Both the WAE-based adaptation approach and the continuous eigendecomposition approach are capable of mitigating the performance degradation when the level of noise in W is low ($\alpha_w \in \{2, 3\}$). Overall, the WAE approach outperforms the continuous eigendecomposition approach and is less sensitive noise in W. In the high-noise setting ($\alpha_w = 1$), the eigendecomposition approach is worse than the ERM-source but has similar performance to the eigendecomposition method without adaptation (Spectral-ERM-source in Figure 3(b)).

8 DISCUSSION

We presented a strategy for unsupervised domain adaptation under latent subgroup shift, which generalizes the standard settings of covariate and label shift. Our strategy leverages auxiliary data in the source domain (concepts C and a proxy W), and generalizes identification results from the causal inference literature to derive an identification strategy for the optimal predictor $q(Y|X)$ under the target distribution. Our identification results are amenable to deep latent variable modeling, and suggest constraints that can be imposed on these models to make them effective for domain adaptation under this particular shift. We demonstrated these claims in a carefully designed numerical example.

Limitations and future work While a latent variable model has been shown promising to estimate the quantities of interest, such models are tricky to tune in practice, and have many known failure modes when used in causal contexts (see, e.g., Rissanen and Marttinen (2021), who critique the method proposed in Louizos et al. (2017)). The

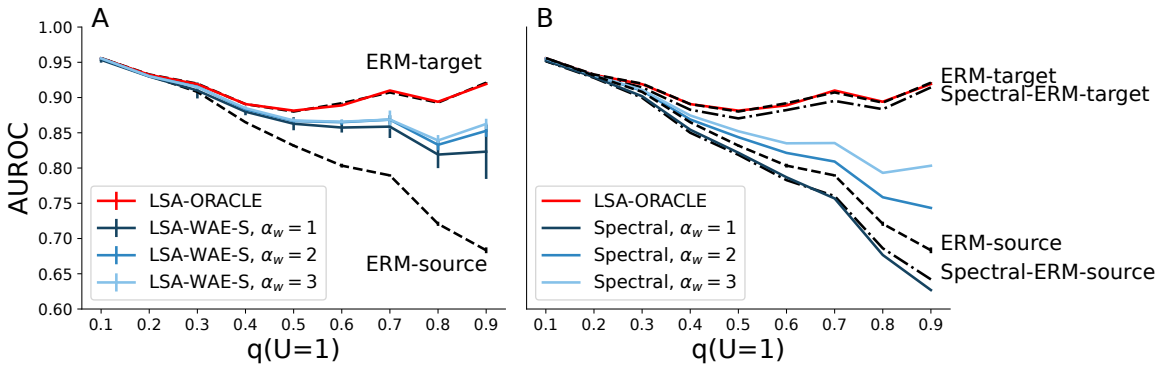

Figure 3: Simulation study: adaptation over target environments and varying levels of noise in the proxy variable W ($\alpha_w = [1, 2, 3]$ for high, medium, and low noise). We plot the mean $\pm$ std AUROC for different shift levels $q(U=1) \in [0.1, 0.9]$ across ten trials. All models trained in a fixed source domain ($p(U=1)=0.1$). **Panel A** compares adaptation with Wassertein Autoencoders with structured decoders (LSA-WAE-S) and **Panel B** compares adaptation with the continuous eigendecomposition (spectral) approach.

identification arguments and corresponding modifications we make to the latent variable model may address some of these concerns, but practical challenges still remain. For example, in practice, we observed that the dimensionality of the latent space mattered (the higher, the better) and that multiple preprocessing and training choices influenced the fit of the model.

Our approach requires the availability of mediating concepts C and of a proxy variable W at training time. This information might not be readily available, or it may not satisfy all the assumptions (e.g. C such that $p(Y|C, U, X) = p(Y|C, U)$). Furthermore, these assumptions are typically not testable as U is not observed. However, we hope that our identification results can serve as motivation for careful collection of richer data, in which concepts and proxies may be present by design.

It is also worth deriving estimation guarantees (i.e., consistency guarantees, error bounds) for estimators of $q(Y|X)$. This would help understand if further data in Q could improve estimation. For example, if we also observed C in Q would this more tightly bound the error of $q(Y|X)$?

We study the case where U is discrete and other variables $\{W, Y, X, C\}$ can be either discrete or continuous. It is interesting to study the identification in the case that U is continuous. In addition, our identification results require additional assumptions, i.e., A4–A6, that potentially limit the the class of distributions. These assumptions arise from the eigendecomposition technique used to show identification. It would interesting to understand whether these assumptions can be relaxed, perhaps incorporating results from proximal causal inference and missing data methods that do not need to identify the full joint distribution of observables and latent variables (see, e.g., Tchetgen Tchetgen et al., 2020; Kallus et al., 2021; Li et al., 2021).

Acknowledgements We would like to thank Victor Veitch and Alexander Brown for valuable discussions and feedback. This work was funded by Google and supported by the Gatsby charitable foundation.

References

Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning*, pages 222–232. PMLR, 2020.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2019.

Mohammad Taha Bahadori and David Heckerman. Debiasing concept-based explanations with causal analysis. In *International Conference on Learning Representations*, 2021.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in Neural Information Processing Systems*, 19, 2006.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(9), 2009.

Xiangli Chen, Mathew Monfort, Anqi Liu, and Brian D

Ziebart. Robust covariate shift regression. In *Artificial Intelligence and Statistics*, pages 1270–1279. PMLR, 2016.

Yuansi Chen and Peter Bühlmann. Domain adaptation under structural causal models. *arXiv preprint arXiv:2010.15764*, 2020.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, December 2020.

Corinna Cortes and Mehryar Mohri. Domain adaptation in regression. In *International Conference on Algorithmic Learning Theory*, pages 308–323. Springer, 2011.

Marthinus Christoffel du Plessis and Masashi Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. In *ICML*, 2012.

Samuel G Finlayson, Adarsh Subbaswamy, Karandeep Singh, John Bowers, Annabel Kupke, Jonathan Zittrain, Isaac S Kohane, and Suchi Saria. The clinician and dataset shift in artificial intelligence. *N. Engl. J. Med.*, 385(3): 283–286, July 2021.

George Forman. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2): 164–206, 2008.

Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary Lipton. A unified view of label shift estimation. *Advances in Neural Information Processing Systems*, 33: 3290–3300, 2020.

JJ Gart and AA Buck. Comparison of a screening test and a reference test in epidemiologic studies. ii. a probabilistic model for the comparison of diagnostic tests. *American journal of epidemiology*, 83(3):593–602, 1966.

Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International Conference on Machine Learning*, pages 2839–2848. PMLR, 2016.

Mingming Gong, Kun Zhang, Biwei Huang, Clark Glymour, Dacheng Tao, and Kayhan Batmanghelich. Causal generative domain adaptation networks. *arXiv preprint arXiv:1804.04333*, 2018.

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

S Greenland and TL Lash. Bias analysis in modern epidemiology. *Philadelphia, PN: Lippincott Williams & Wilkins*, pages 345–380, 2008.

Sander Greenland. Multiple-bias modelling for analysis of observational data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 168(2):267–306, 2005.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in Neural Information Processing Systems*, 19, 2006.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 527–536. PMLR, 2019.

Nathan Kallus, Xiaojie Mao, and Masatoshi Uehara. Causal inference under unmeasured confounding with negative controls: A minimax learning approach. *arXiv preprint arXiv:2103.14029*, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th international conference on computer vision*, pages 365–372. IEEE, 2009.

Manabu Kuroki and Judea Pearl. Measurement bias and effect restoration in causal inference. *Biometrika*, 101(2): 423–437, 2014.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pages 951–958. IEEE, 2009.

Sanghack Lee and Elias Bareinboim. Causal identification with matrix equations. *Advances in Neural Information Processing Systems*, 34, 2021.

Wei Li, Wang Miao, and Eric Tchetgen Tchetgen. Nonparametric inference about mean functionals of nonignorable

nonresponse data without identifying the joint distribution. *arXiv preprint arXiv:2110.05776*, 2021.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning*, pages 3122–3130. PMLR, 2018.

Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *Advances in Neural Information Processing Systems*, 30, 2017.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Sara Magliacane, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of Black-Box concept learning models. June 2021.

Charles F Manski and Steven R Lerman. The estimation of choice probabilities from choice based samples. *Econometrica: Journal of the Econometric Society*, pages 1977–1988, 1977.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Michael Marmot and Richard Wilkinson. *Social determinants of health*. Oup Oxford, 2005.

Wang Miao, Zhi Geng, and Eric J Tchetgen Tchetgen. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 105(4):987–993, 2018.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

Judea Pearl. On measurement bias in causal inference. In *UAI*, 2010.

Judea Pearl et al. Causality: Models, reasoning and inference. *Cambridge University Press*, 19:2, 2000.

Severi Rissanen and Pekka Marttinen. A critical look at the consistency of causal estimation with deep latent variable models. *Advances in Neural Information Processing Systems*, 34:4207–4217, 2021.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33:11539–11551, 2020.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *International Conference on Machine Learning*, pages 459–466, 2012.

Jessica Schrouff, Natalie Harris, Oluwasanmi Koyejo, Ibrahim Alabdulmohsin, Eva Schnider, Krista Opsahl-Ong, Alex Brown, Subhrajit Roy, Diana Mincu, Christina Chen, et al. Maintaining fairness across distribution shift: do we have viable solutions for real-world applications? *arXiv preprint arXiv:2202.01034*, 2022.

Jan Selén. Adjusting for errors in classification and measurement in the analysis of partly and purely categorical data. *Journal of the American Statistical Association*, 81 (393):75–81, 1986.

Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2): 227–244, 2000.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Amos Storkey. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, 30:3–28, 2009.

Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

Masashi Sugiyama and K Mueller. Generalization error estimation under covariate shift. In *Workshop on Information-Based Induction Sciences*, pages 21–26. Citeseer, 2005.

Masashi Sugiyama, Ichiro Takeuchi, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Daisuke Okanohara. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 781–788. JMLR Workshop and Conference Proceedings, 2010.

Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33: 19276–19289, 2020.

Eric J Tchetgen Tchetgen, Andrew Ying, Yifan Cui, Xu Shi, and Wang Miao. An introduction to proximal causal learning. *arXiv preprint arXiv:2009.10982*, 2020.

Takeshi Teshima, Issei Sato, and Masashi Sugiyama. Few-shot domain adaptation by causal mechanism transfer. In *International Conference on Machine Learning*, pages 9458–9469. PMLR, 2020.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkL7n1-0b.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Ruihan Wu, Chuan Guo, Yi Su, and Kilian Q Weinberger. Online adaptation to label distribution shift. *Advances in Neural Information Processing Systems*, 34:11340–11351, 2021.

Zhongqi Yue, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Transporting causal mechanisms for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8599–8608, 2021.

Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first International Conference on Machine Learning*, page 114, 2004.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827. PMLR, 2013.

Kun Zhang, Mingming Gong, and Bernhard Schölkopf. Multi-source domain adaptation: A causal view. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

APPENDIX

A PROOFS

A.1 Proof of Lemma 1

Recall Lemma 1:

Lemma 1. *Given that the above assumptions hold, all probability mass functions over discrete $\{W, X, C, Y, \widetilde{U}\}$ in the source P are identifiable, where $\widetilde{U}$ is an unknown sorting of U.*

Before we prove this we will prove a variant of Theorem 1 of Kuroki and Pearl (2014).

Lemma 2 (variant of Theorem 1 of Kuroki and Pearl (2014))**.** *Given A1-A4, $p(W|\widetilde{U})$ is identifiable.*

Proof. First, fix a k_U. Without any additional information the easiest is to set $k_U = k_W$. However, if you believe that $k_U < k_W$, coarsen W by dropping categories to ensure that the new dimensionality k'_W is equal to k_U. Next notice that, given A1 (Figure 1 (c)) we can factorize the joint of W, X, Y conditional on C as:

$$p(Y, X, W \mid C) = \sum_{k=1}^{k_U} p(Y \mid C, U = k)p(X \mid C, U = k)p(W \mid U = k)p(U = k \mid C).$$

Next, construct the following matrices based on the decomposition of $p(Y, X, W|C)$ and of its marginal distributions:

$$\mathbf{A} := \begin{bmatrix} 1 & p(W=1|C) & \cdots & p(W=k_W-1|C) \\ p(X=1|C) & p(X=1,W=1|C) & \cdots & p(X=1,W=k_W-1|C) \\ \vdots & \vdots & \ddots & \vdots \\ p(X=k_X-1|C) & p(X=k_X-1,W=1|C), & \cdots & p(X=k_X-1,W=k_W-1|C) \end{bmatrix}$$

$$\mathbf{B} := \begin{bmatrix} p(Y|C) & p(Y,W=1|C) & \cdots & p(Y,W=k_W-1|C) \\ p(Y,X=1|C) & p(Y,X=1,W=1|C) & \cdots & p(Y,X=1,W=k_W-1|C) \\ \vdots & \vdots & \ddots & \vdots \\ p(Y,X=k_X-1|C) & p(Y,X=k_X-1,W=1|C) & \cdots & p(Y,X=k_X-1,W=k_W-1|C) \end{bmatrix}$$

$$\mathbf{R} := \begin{bmatrix} 1 & p(X=1|C,U=1) & \cdots & p(X=k_X-1|C,U=1) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & p(X=1|C,U=k_U) & \cdots & p(X=k_X-1|C,U=k_U) \end{bmatrix}$$

$$\mathbf{M} := \begin{bmatrix} p(U=1|C) & 0 & \cdots & 0 \\ & \ddots & & \\ 0 & \cdots & 0 & p(U=k_U|C) \end{bmatrix}$$

$$\Lambda := \begin{bmatrix} p(Y|C,U=1) & 0 & \cdots & 0 \\ & \ddots & & \\ 0 & \cdots & 0 & p(Y|C,U=k_U) \end{bmatrix}$$

$$\mathbf{S} := \begin{bmatrix} 1 & p(W=1|U=1) & \cdots & p(W=k_W-1|U=1) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & p(W=1|U=k_U) & \cdots & p(W=k_W-1|U=k_U) \end{bmatrix}.$$

Then note that

$$\mathbf{A} = \mathbf{R}^\top \mathbf{M}\mathbf{S} \qquad \mathbf{B} = \mathbf{R}^\top \mathbf{M}\Lambda\mathbf{S}. \tag{4}$$

We then have that,

$$\begin{aligned}\mathbf{A}^\dagger\mathbf{B} &= \left[\left(\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top\right]\mathbf{R}^\top\mathbf{M}\Lambda\mathbf{S} \\ &= \left(\mathbf{S}^\top\mathbf{M}\mathbf{R}\mathbf{R}^\top\mathbf{M}\mathbf{S}\right)^{-1}\mathbf{S}^\top\mathbf{M}\mathbf{R}\left(\mathbf{R}^\top\mathbf{M}\Lambda\mathbf{S}\right) \\ &\underset{(a)}{=} (\mathbf{S})^{-1}\Lambda\mathbf{S},\end{aligned} \tag{5}$$

where $\mathbf{A}^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{A}$ (recall all pseudoinverses are unique and exist). Recall that above we have ensured that the dimensionality of W is equal to the dimensional U. Thus, $\mathbf{S}$ is square. Further, both $\mathbf{S}$ and $\mathbf{R}$ have rank at least k_U by A4. So $\mathbf{S}$ and $\mathbf{R}\mathbf{R}^\top$ are invertible. Because we have to marginalize U in order to obtain observed distributions, it is only possible to identify U up to an arbitrary permutation. Specifically, let $\widetilde{U}$ be a sorting of U such that $p(Y|C,\widetilde{U}=1) > p(Y|C,\widetilde{U}=2) > \cdots > p(Y|C,\widetilde{U}=k_U)$.

Now we need to show that we can obtain $p(W|\widetilde{U})$ from eigendecomposition of $\mathbf{A}^\dagger\mathbf{B}$. To do so we first must solve $|\mathbf{A}^\dagger\mathbf{B} - \lambda\mathbf{I}| = 0$ for λ, to obtain the eigenvalues of $\mathbf{A}^\dagger\mathbf{B}$. Note that $|\mathbf{A}^\dagger\mathbf{B} - \lambda\mathbf{I}| = |(\mathbf{S})^{-1}\Lambda\mathbf{S} - \lambda\mathbf{I}| = |\Lambda - \lambda\mathbf{I}| = 0$ where the second-to-last equality uses the Weinstein–Aronszajn identity $|(\mathbf{S})^{-1}\Lambda\mathbf{S} - \lambda\mathbf{I}| = |\mathbf{S}(\mathbf{S})^{-1}\Lambda - \lambda\mathbf{I}| = |\Lambda - \lambda\mathbf{I}|$. Therefore, if we define $\lambda_1 > \cdots \lambda_{k_U}$ as the eigenvalues of $\mathbf{A}^\dagger\mathbf{B}$, it must be that $\lambda_i = p(Y|C,\widetilde{U}=i)$ for $i = 1,\ldots,k_U$.

Now that we have identified $p(Y|C,\widetilde{U})$ we will show we can obtain $p(W|\widetilde{U})$ from λ_i and the eigenvectors η_i of $\mathbf{A}^\dagger\mathbf{B}$. Define the matrix of eigenvectors as $\mathbf{H} = [\eta_1,\ldots,\eta_{k_U}]$. To obtain this we must solve the linear system $\mathbf{A}^\dagger\mathbf{B}\mathbf{H} = \mathbf{H}\Lambda$. Note that $\mathbf{H}$ is determined up to a multiplicative constant as $\lambda_1 \neq \cdots \neq \lambda_{k_U}$ from A4. Define a matrix of non-zero multiplicative constants $\mathbf{E} = \text{diag}(\alpha_1,\ldots,\alpha_{k_U})$ and the shifted matrix $\mathbf{F} = \mathbf{S}^{-1}\mathbf{E}$. Note that $\mathbf{A}^\dagger\mathbf{B}\mathbf{F} = \mathbf{S}^{-1}\Lambda\mathbf{S}\mathbf{S}^{-1}\mathbf{E} = \mathbf{S}^{-1}\Lambda\mathbf{E} = \mathbf{S}^{-1}\mathbf{E}\Lambda = \mathbf{F}\Lambda$. Therefore, $\mathbf{F}$ is also a matrix of eigenvectors of $\mathbf{A}^\dagger\mathbf{B}$, and that $\mathbf{F} = \mathbf{S}^{-1}\mathbf{E} = \mathbf{H}$ for certain values of $\alpha_1,\ldots,\alpha_{k_U}$. To recover these, note that,

$$\mathbf{S} = \begin{bmatrix} 1 & p(W=1|U=1) & \cdots & p(W=k_W-1|U=1) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & p(W=1|U=k_U) & \cdots & p(W=k_W-1|U=k_U) \end{bmatrix} = \mathbf{E}\mathbf{H}^{-1} = \begin{bmatrix} \alpha_1 h_{11} & \cdots & \alpha_1 h_{1k_U} \\ \vdots & \ddots & \vdots \\ \alpha_{k_U} h_{k_U 1} & \cdots & \alpha_{k_U} h_{k_U k_U} \end{bmatrix}.$$

Equating the first column of both sides of the equation we have that $\alpha_1 = 1/h_{11},\ldots,\alpha_{k_U} = 1/h_{k_U 1}$. This means that $\mathbf{S}$ is identifiable from $\mathbf{E}\mathbf{H}^{-1}$ as $\mathbf{H}^{-1}$ is what we estimate from eigendecomposition of $\mathbf{A}^\dagger\mathbf{B}$. Therefore, every element of $p(W|\widetilde{U})$ is identifiable. □

Now that we have obtained $p(W|\widetilde{U})$, we can prove Lemma 1.

Proof. As distributions that only involve $\{W,X,C,Y\}$ are observable, all we need to prove is that we can identify all distributions involving $\widetilde{U}$. Let $\mathcal{V} \subseteq \{W,X,C,Y\}$ and $\mathcal{V}' \subseteq \{W,X,C,Y\} \setminus \mathcal{V}$. All we need to identify are

(a) $p(\widetilde{U})$;

(b) $p(\mathcal{V} \mid \widetilde{U})$;

(c) $p(\widetilde{U} \mid \mathcal{V})$;

(d) $p(\mathcal{V} \mid \widetilde{U}, \mathcal{V}')$.

Note that proving above identities are sufficient because (e) $p(\widetilde{U},\mathcal{V} \mid \mathcal{V}') = p(\mathcal{V} \mid \widetilde{U},\mathcal{V}')p(\widetilde{U} \mid \mathcal{V}')$ (given by (d) and (c)).

Identifying (a) $p(\widetilde{U})$. The identification is straightforward: note that $p(\widetilde{\mathbf{U}}) = p(\mathbf{W}|\widetilde{\mathbf{U}})^\dagger p(\mathbf{W})$.

Identifying (b) $p(\mathcal{V} \mid \widetilde{U})$. Recall we have already identified $p(\mathbf{W}|\widetilde{\mathbf{U}})$. zlet $\mathcal{V}_{\setminus W} = \mathcal{V} \setminus W$. Note that $p(\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}}, \mathbf{W}|\widetilde{\mathbf{U}}) = p(\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}}|\widetilde{\mathbf{U}})p(\mathbf{W}|\widetilde{\mathbf{U}})$ because $\mathcal{V}_{\setminus W} \perp\!\!\!\perp W \mid \widetilde{U}$. Hence, we have

$$p(\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}} \mid \mathbf{W}) = p(\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}} \mid \widetilde{\mathbf{U}})p(\widetilde{\mathbf{U}} \mid \mathbf{W}).$$

By multiplying $p(\widetilde{\mathbf{U}} \mid \mathbf{W})^\dagger$ on both side, we can obtain

$$p(\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}} \mid \widetilde{\mathbf{U}}) = p(\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}} \mid \mathbf{W})p(\widetilde{\mathbf{U}} \mid \mathbf{W})^\dagger$$

Note that this is identified because the first term on the right-hand side is observed and the second term can be identified via Bayes rule $p(\widetilde{U}|W) = p(W|\widetilde{U})p(\widetilde{U})/p(W)$, where $p(W|\widetilde{U})$ is identifiable as shown in Lemma 2.

Identifying (c) $p(\widetilde{U} \mid \mathcal{V})$**.** We have identified $p(\widetilde{\mathbf{U}}|\mathbf{W})$ in the previous step using Bayes rule. We then have that $p(\mathbf{W}|\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}}) = p(\mathbf{W}|\widetilde{\mathbf{U}})p(\widetilde{\mathbf{U}}|\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}}) \Rightarrow p(\widetilde{\mathbf{U}}|\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}}) = p(\mathbf{W}|\widetilde{\mathbf{U}})^{\dagger}p(\mathbf{W}|\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}})$, which is identifiable. Finally we have via Bayes rule $p(\widetilde{U}|\mathcal{V}_{\setminus W}, W) = p(\mathcal{V}_{\setminus W}, W|\widetilde{U})p(\widetilde{U})/p(\mathcal{V}_{\setminus W}, W)$ all of which we can identify (via (a) and (b)).

Identifying (d) $p(\mathcal{V} \mid \widetilde{U}, \mathcal{V}')$**.** Note that $p(\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}}|\mathcal{V}'_{\setminus W}, \mathbf{W}) = p(\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}}|\widetilde{\mathbf{U}}, \mathcal{V}'_{\setminus W})p(\widetilde{\mathbf{U}}|\mathcal{V}'_{\setminus W}, \mathbf{W})$, which implies that

$$p(\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}}|\widetilde{\mathbf{U}}, \mathcal{V}'_{\setminus W}) = p(\boldsymbol{\mathcal{V}}_{\setminus \boldsymbol{W}}|\mathcal{V}'_{\setminus W}, \mathbf{W})p(\widetilde{\mathbf{U}}|\mathcal{V}'_{\setminus W}, \mathbf{W})^{\dagger}.$$

The first term on the right-hand side is observed and the second is identified via (c). Finally note that $p(\mathcal{V}_{\setminus W}, W|\widetilde{U}, \mathcal{V}'_{\setminus W}) = p(W|\mathcal{V}_{\setminus W}, \widetilde{U}, \mathcal{V}'_{\setminus W})p(\mathcal{V}_{\setminus W}|\widetilde{U}, \mathcal{V}'_{\setminus W}) = p(W|\widetilde{U})p(\mathcal{V}_{\setminus W}|\widetilde{U}, \mathcal{V}'_{\setminus W})$ (as $W \perp\!\!\!\perp \mathcal{V}_{\setminus W}|\widetilde{U}$), where all right-hand terms are identified. Also that $p(\mathcal{V}_{\setminus W}, |W, \widetilde{U}, \mathcal{V}'_{\setminus W}) = p(\mathcal{V}_{\setminus W}, |\widetilde{U}, \mathcal{V}'_{\setminus W})$ which is identified. □

A.2 Proof of Theorem 1

Theorem 1 (Discrete Observations). *The distribution $q(Y|X)$ is identifiable from discrete $\{W, X, C, Y, \widetilde{U}\} \sim P$ and $X \sim Q$.*

Proof. The first observation is that we can replace U with $\widetilde{U}$ everywhere. This is because $\widetilde{U}$ has the exact same conditional independences as U that are required in the factorization of $q(Y|X)$ in eq. (1) (as there are no requirements on the ordering of the categories of U). Further, we can replace U with $\widetilde{U}$ in eq. (1) without changing anything, i.e.,

$$q(Y|X) \propto \sum_{i=1}^{k_U} p(Y|X, \widetilde{U} = i)p(\widetilde{U} = i|X)\frac{q(\widetilde{U} = i)}{p(\widetilde{U} = i)}. \tag{6}$$

This is because we are summing over categories of U and so it makes no difference to change the order of categories of U, as in $\widetilde{U}$. The only remaining thing to show is that $\frac{q(\widetilde{U}=i)}{p(\widetilde{U}=i)}$ can be identified. Note that

$$\frac{q(X)}{p(X)} = \sum_{k=1}^{k_U} p(\widetilde{U} = i|X)\frac{q(\widetilde{U} = i)}{p(\widetilde{U} = i)}.$$

Define the vector $\mathbf{v}_X = [q(X = 1)/p(X = 1), \ldots, q(X = k_X)/p(X = k_X)]$, the matrix $\mathbf{N}_{ij} = p(\widetilde{U} = i|X = j)$, and the vector $\mathbf{v}_U = [q(U = 1)/p(U = 1), \ldots, q(U = k_X)/p(U = k_U)]$. We have that $\mathbf{v}_U = \mathbf{N}^{\dagger}\mathbf{v}_X$. Note that $\frac{q(\widetilde{U}=i)}{p(\widetilde{U}=i)}$ is identified because $\mathbf{N}, \mathbf{v}_X$ are identified, and $\mathbf{N}^{\dagger} = (\mathbf{N}^{\top}\mathbf{N})^{-1}\mathbf{N}^{\top}$ because $k_X \geq k_U$ by A3 and (b) all linear systems have rank at least k_U by A4. □

A.3 Proof of Theorem 2

We first restate Theorem 2:

Theorem 2 (Continuous Observations). *The distribution $q(Y|X)$ is identifiable from continuous $\{W, X, C, Y\} \sim P$ and $X \sim Q$, and discrete $\widetilde{U} \sim P$.*

Proof. The proof steps is similar to the proof of Theorem 1: we can factorize the probability as (6). We identify each component as follows.

Identifying $p(W \mid \widetilde{U})$**.** We first show the continuous version of Lemma 2. As in the discrete case, given A1 we can factorize $p(Y, X, W|C)$ as written above. We rewrite it here in order to define functions $\boldsymbol{\psi}_i(X), \boldsymbol{\phi}_i(W)$ and quantities s_i, m_i as follows,

$$p(Y, X, W \mid C) = \sum_{k=1}^{k_U} \overbrace{p(Y \mid C, U = k)}^{m_i} \overbrace{p(X \mid C, U = i)}^{\boldsymbol{\psi}_i(X)} \overbrace{p(W \mid U = i)}^{\boldsymbol{\phi}_i(W)} \overbrace{p(U = i \mid C)}^{s_i}.$$

To construct the integral operators for A, B let $\mathcal{W}, \mathcal{X}$ be the domains of X, W, respectively. Let $L_2(\mathcal{W}, \mu)$ be the space of L_2-integrable functions on $\mathcal{W}$ with Lebesgue measure μ (and similarly for $\mathcal{X}$). Let $A : L_2(\mathcal{W}, \mu) \to L_2(\mathcal{X}, \mu)$ and

$B : L_2(\mathcal{W}, \mu) \to L_2(\mathcal{X}, \mu)$ be the integral operators associated with kernel functions $p(X, W|C)$ and $p(Y, X, W|C)$, respectively. They are defined as

$$A := \sum_{i=1}^{k_U} s_i \psi_i(X) \otimes \phi_i(W) \qquad B := \sum_{i=1}^{k_U} s_i m_i \psi_i(X) \otimes \phi_i(W)$$

Note that these operators operate on any function $h \in L_2(\mathcal{W}, \mu)$ in the following way, e.g., for A,

$$Ah = \sum_{i=1}^{k_U} s_i \psi_i(X) \langle \phi_i(W), h \rangle, \qquad s.t., \qquad \langle \phi_i(W), h \rangle := \int \phi_i(W) h(W) dW.$$

Next we will describe how we can identify functions $p(Y|C, \widetilde{U})$ and $p(W|\widetilde{U})$ from eigendecomposition of the operator $A^\dagger B$. We begin by collecting functions into vectors/matrices that will make up this decomposition. Define the row vectors of functions

$$\boldsymbol{\psi} := [\psi_1(X), \ldots, \psi_{k_U}(X)];$$
$$\boldsymbol{\phi} := [\phi_1(W), \ldots, \phi_{k_U}(W)].$$

To fix the scale of the decomposition, we will apply the Gram–Schmidt process to $\boldsymbol{\psi}, \boldsymbol{\phi}$ to create the set of orthonormal functions

$$\overline{\boldsymbol{\psi}} := [\overline{\psi_1}(X), \ldots, \overline{\psi_{k_U}}(X)]$$
$$\overline{\boldsymbol{\phi}} := [\overline{\phi_1}(W), \ldots, \overline{\phi_{k_U}}(W)].$$

Note that the Gram-Schmidt process is well-defined as the inner product between functions is defined. This process also creates upper triangular matrices $\mathbf{R}_\psi, \mathbf{R}_\phi \in \mathbb{R}^{k_U \times k_U}$ that map the orthonormal functions back to their originals $\boldsymbol{\psi} = \overline{\boldsymbol{\psi}}\mathbf{R}_\psi$ and $\boldsymbol{\phi} = \overline{\boldsymbol{\phi}}\mathbf{R}_\phi$. Finally define the diagonal matrices $\Lambda_s := \text{diag}(s_1, \ldots, s_{k_U})$ and $\Lambda_m := \text{diag}(m_1, \ldots, m_{k_U})$ Now note the following decompositions:

$$A = \boldsymbol{\psi}\Lambda_s\boldsymbol{\phi}^\top = \overline{\boldsymbol{\psi}}\mathbf{R}_\psi\Lambda_s\mathbf{R}_\phi^\top(\overline{\boldsymbol{\phi}})^\top \qquad B = \boldsymbol{\psi}\Lambda_s\Lambda_m\boldsymbol{\phi}^\top = \overline{\boldsymbol{\psi}}\mathbf{R}_\psi\Lambda_s\Lambda_m\mathbf{R}_\phi^\top(\overline{\boldsymbol{\phi}})^\top \tag{7}$$

Notice that the operator A is a finite-rank operator mapping between two finite dimensional spaces $A : \mathbb{H}_\phi \to \mathbb{H}_\psi$, as $\mathbb{H}_\phi, \mathbb{H}_\psi$ are closed subspaces spanned by $\boldsymbol{\psi}, \boldsymbol{\phi}$. Then we can write the inverse of A as:

$$A^{-1} = \overline{\boldsymbol{\phi}}(\mathbf{R}_\phi^\top)^{-1}\Lambda_s^{-1}\mathbf{R}_\psi^{-1}(\overline{\boldsymbol{\psi}})^\top.$$

It follows that,

$$A^{-1}B = \overline{\boldsymbol{\phi}}(\mathbf{R}_\phi^\top)^{-1}\Lambda_s^{-1}\mathbf{R}_\psi^{-1}(\overline{\boldsymbol{\psi}})^\top\overline{\boldsymbol{\psi}}\mathbf{R}_\psi\Lambda_s\Lambda_m\mathbf{R}_\phi^\top(\overline{\boldsymbol{\phi}})^\top = \overline{\boldsymbol{\phi}}(\mathbf{R}_\phi^\top)^{-1}\Lambda_m\mathbf{R}_\phi^\top(\overline{\boldsymbol{\phi}})^\top.$$

Given the above decomposition, we now show that we can identify $\Lambda_m, \boldsymbol{\phi}$ and thus $p(Y|C, \widetilde{U}), p(W|\widetilde{U})$ via eigendecomposition. First notice that eigendecomposition of $A^{-1}B$ gives $\overline{\boldsymbol{\phi}}(\mathbf{R}_\phi^\top)^{-1}\Lambda_m\mathbf{R}_\phi^\top(\overline{\boldsymbol{\phi}})^\top$. As in the discrete observation setting we have that the eigenvalues $\lambda_1, \ldots, \lambda_{k_U}$ must satisfy $|A^{-1}B - \lambda\mathbf{I}| = |\Lambda_m - \lambda\mathbf{I}| = 0$. Therefore, $\lambda_i = p(Y|C, \widetilde{U} = i)$. Using the same argument as we use in Theorem 2, it follows that column of $\overline{\boldsymbol{\phi}}(\mathbf{R}_\phi^\top)^{-1}$ are eigenfunctions of $A^{-1}B$. Applying the Gram–Schmidt process to $\overline{\boldsymbol{\phi}}(\mathbf{R}_\phi^\top)^{-1}$, we recover $\overline{\boldsymbol{\phi}}$ and $(\mathbf{R}_\phi^\top)^{-1}$. We can then invert $(\mathbf{R}_\phi^\top)^{-1}$ to identify $\boldsymbol{\phi}$ via $\boldsymbol{\phi} = \overline{\boldsymbol{\phi}}\mathbf{R}_\phi$, and thus $p(W|\widetilde{U})$.

All that is left to show is how to identify $p(\widetilde{U}|X = x), p(Y| = x, \widetilde{U}), q(\widetilde{U})/p(\widetilde{U})$.

Identifying $p(\widetilde{U}|X = x)$**.** Under the A1 , we have $W \perp\!\!\!\perp X \mid U$. Hence, we can write

$$p(W \mid X = x) = \sum_{k=1}^{k_U} p(W \mid \widetilde{U} = i) p(\widetilde{U} = k \mid X = x). \tag{8}$$

By the linear independence condition stated in A5, we know that $f(W \mid X = x)$ is uniquely represented by $f(W \mid \widetilde{U} = 1), \ldots, f(W \mid \widetilde{U} = k_U)$. This implies for any $x \in \text{Dom}(X)$, we can identify $p(\widetilde{U} = i \mid X = x)$.

Identifying $p(Y|x,\widetilde{U})$. Note that

$$\begin{aligned}
p(W,Y \mid x) &= \sum_{k=1}^{k_U} p(Y \mid x, \widetilde{U}=i)p(W \mid \widetilde{U}=k)p(\widetilde{U}=k \mid x) \\
&= \boldsymbol{\phi} \begin{bmatrix} p(\widetilde{U}=1 \mid x) & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & p(\widetilde{U}=k \mid x) \end{bmatrix} \begin{bmatrix} p(Y \mid x, \widetilde{U}=1) \\ \vdots \\ p(Y \mid x, \widetilde{U}=k) \end{bmatrix} \\
&= \overline{\boldsymbol{\phi}} R_\phi \begin{bmatrix} p(\widetilde{U}=1 \mid x) & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & p(\widetilde{U}=k \mid x) \end{bmatrix} \begin{bmatrix} p(Y \mid x, \widetilde{U}=1) \\ \vdots \\ p(Y \mid x, \widetilde{U}=k) \end{bmatrix}.
\end{aligned}$$

Since $\overline{\boldsymbol{\phi}}_1, \ldots, \overline{\boldsymbol{\phi}}_k$ are pairwise orthonormal, it follows that for any $i \in \{1, \ldots, k\}$

$$\langle p(W,Y \mid X=x), \overline{\boldsymbol{\phi}}_i \rangle = \int_{\mathcal{W}} p(W,Y \mid X=x)\overline{\boldsymbol{\phi}}_i dw = z_i(Y,x,\widetilde{U}=i). \tag{9}$$

Then, we can obtain

$$\begin{bmatrix} p(Y \mid x, \widetilde{U}=1) \\ \vdots \\ p(Y \mid x, \widetilde{U}=k) \end{bmatrix} = R_\phi^{-1} \begin{bmatrix} \frac{1}{p(\widetilde{U}=1|x)} & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \frac{1}{p(\widetilde{U}=k|x)} \end{bmatrix} \begin{bmatrix} z_i(Y,x,\widetilde{U}=1) \\ \vdots \\ z_k(Y,x,\widetilde{U}=k) \end{bmatrix}$$

as the inverse of R_ϕ exists given A5.

Identifying $q(\widetilde{U})/p(\widetilde{U})$. Note that

$$\begin{aligned}
q(X) = \sum_{k=1}^{k_U} q(X \mid \widetilde{U}=k)q(\widetilde{U}=k) &= \sum_{i=k}^{k_U} p(X \mid \widetilde{U}=k)q(\widetilde{U}=k) \\
&= \sum_{i=k}^{k_U} p(\widetilde{U}=k, X)\frac{q(\widetilde{U}=k)}{p(\widetilde{U}=k)} \\
&= \sum_{i=k}^{k_U} p(\widetilde{U}=k|X)p(X)\frac{q(\widetilde{U}=k)}{p(\widetilde{U}=k)}.
\end{aligned}$$

This implies that for all $x \in \mathrm{Dom}(X)$

$$\frac{q(x)}{p(x)} = \sum_{k=1}^{k_U} p(\widetilde{U}=k|x)\frac{q(\widetilde{U}=k)}{p(\widetilde{U}=k)}.$$

Now select observed $x_1, \ldots, x_k$ that satisfies A6. Then, we can write

$$\underbrace{\begin{bmatrix} \frac{q(x_1)}{p(x_1)} \\ \frac{q(x_2)}{p(x_2)} \\ \vdots \\ \frac{q(x_k)}{p(x_k)} \end{bmatrix}}_{\mathbf{v}_{q,p,X}} = \underbrace{\begin{bmatrix} p(\widetilde{U}=1 \mid x_1) & p(\widetilde{U}=2 \mid x_1) & \cdots & p(\widetilde{U}=k \mid x_1) \\ p(\widetilde{U}=1 \mid x_2) & p(\widetilde{U}=2 \mid x_2) & \cdots & p(\widetilde{U}=k \mid x_2) \\ \vdots & \ddots & & \vdots \\ p(\widetilde{U}=1 \mid x_k) & p(\widetilde{U}=2 \mid x_k) & \cdots & p(\widetilde{U}=k \mid x_k) \end{bmatrix}}_{\mathbf{M}_{\widetilde{U},X}} \underbrace{\begin{bmatrix} \frac{q(\widetilde{U}=1)}{p(\widetilde{U}=1)} \\ \frac{q(\widetilde{U}=2)}{p(\widetilde{U}=2)} \\ \vdots \\ \frac{q(\widetilde{U}=k)}{p(\widetilde{U}=k)} \end{bmatrix}}_{\mathbf{v}_{q,p,\widetilde{U}}}.$$

By A6, the confusion matrix $\mathbf{M}_{\widetilde{U},X}$ is invertible and hence we can obtain $q(\widetilde{U})/p(\widetilde{U})$ via $\mathbf{v}_{q,p,\widetilde{U}} = \mathbf{M}_{\widetilde{U},X}^{-1}\mathbf{v}_{q,p,X}$, and we are done. □

Table 3: Cross-entropy for continuous observations ($\alpha_w = 1$, $n = 10^4$), mean $\pm$ std over 10 training replicates.

Method	Source	Target
ERM-SOURCE	0.1683 ± 0.0002	0.3979 ± 0.0007
COVAR	0.2489 ± 0.0107	0.4206 ± 0.0711
LABEL	0.1726 ± 0.0021	0.3635 ± 0.0111
BBSE	0.2372 ± 0.0065	0.8461 ± 0.0233
LSA-WAE-S	0.1962 ± 0.0112	0.2530 ± 0.0248
LSA-WAE-V	0.1751 ± 0.0112	0.3929 ± 0.0409
LSA-ORACLE	0.3300 ± 0.0250	0.1637 ± 0.0008
ERM-TARGET	0.3415 ± 0.0010	0.1660 ± 0.0003

Table 4: Accuracy for continuous observations ($\alpha_w = 1$, $n = 10^4$), mean $\pm$ std over 10 training replicates.

Method	Source	Target
ERM-SOURCE	0.9179 ± 0.0006	0.7972 ± 0.0009
COVAR	0.8807 ± 0.0011	0.9199 ± 0.0140
LABEL	0.9153 ± 0.0011	0.8294 ± 0.0115
BBSE	0.8935 ± 0.0030	0.5875 ± 0.0083
LSA-WAE-S	0.8994 ± 0.0093	0.8924 ± 0.0181
LSA-WAE-V	0.9121 ± 0.0113	0.7942 ± 0.0489
LSA-ORACLE	0.8555 ± 0.0225	0.9320 ± 0.0008
ERM-TARGET	0.8653 ± 0.0030	0.9342 ± 0.0005

B Experimental details

Here we describe the construction of the simulation study considered in Section 7. We let $k_U = 2, k_X = 2, k_C = 3, k_Y = 2, k_W = 2$, where X is continuous and U, Y, C, and W are discrete. We generate C as a multilabel variable where each dimension C_j takes on a value of either 0 or 1, giving a discrete variable with 2^{k_C} states. Let $\mathbf{o}(v)$ be the $|V|$-dimensional one-hot representation of a sample from a categorical variable $v \in V$. Let V_j designate the j-th dimension of a categorical random variable V. Let $\mathbf{I}_k$ be the identity matrix of size $k \times k$. Let sign be the function such that $\text{sign}(z) = 1$ if $z > 0$ and $\text{sign}(z) = 0$ otherwise. For a vector $\boldsymbol{\pi}$ drawn from the $(k_U - 1)$-dimensional simplex, the data are simulated as

$$
\begin{aligned}
U &\sim \text{Categorical}(\boldsymbol{\pi}) \\
W \mid U = u &\sim \text{sign}\big(\mathcal{N}(\mathbf{o}(u)\mathbf{M}_{W|U}, 1)\big) \\
X \mid U = u &\sim \mathcal{N}(\mathbf{o}(u)\mathbf{M}_{X|U}, \mathbf{I}_{k_X}) \\
C_j \mid X = x, U = u &\sim \text{Bernoulli}\Big(\text{logit}^{-1}\big(x\mathbf{M}_{C|X,U=u} + \mathbf{o}(u)\mathbf{M}_{C|U}\big)\Big) \\
Y \mid C = c, U = u &\sim \text{Bernoulli}\Big(\text{logit}^{-1}\big(c\mathbf{M}_{Y|C,U=u} + \mathbf{o}(u)\mathbf{M}_{Y|U}\big)\Big),
\end{aligned}
$$

where the matrices are defined as

$$
\begin{aligned}
&\mathbf{M}_{W|U} := \alpha_w \begin{bmatrix} -1 & 1 \end{bmatrix}^\top \quad \mathbf{M}_{X|U} := \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \quad \mathbf{M}_{C|U} := \begin{bmatrix} -2 & 2 & 2 \\ -1 & 1 & 2 \end{bmatrix} \\
&\mathbf{M}_{C|X,U=u_0} := 3 \begin{bmatrix} -2 & 2 & -1 \\ 1 & -2 & -3 \end{bmatrix} \quad \mathbf{M}_{C|X,U=u_1} := 3 \begin{bmatrix} 2 & -2 & 1 \\ -1 & 2 & 3 \end{bmatrix} \\
&\mathbf{M}_{Y|U} := \begin{bmatrix} 2 & 2 \end{bmatrix}^\top \mathbf{M}_{Y|C,U=u_0} := \begin{bmatrix} 3 & -2 & -1 \end{bmatrix}^\top \mathbf{M}_{Y|C,U=u_1} := \begin{bmatrix} 3 & -1 & -2 \end{bmatrix}^\top.
\end{aligned}
$$

To construct the setting used for the simulation experiments, we draw a sample from a source domain where $\boldsymbol{\pi}$ is such that $p(U = 1) = 0.1$. We further draw several target distributions where $\boldsymbol{\pi}$ is such that $q(U = 1) \in \{0.1, 0.2, \ldots, 0.9\}$. We vary the noisiness of the proxy W by generating three copies of the target domain datasets where $\alpha_w \in \{1, 2, 3\}$ such that greater values for α_w indicate less noise.

For the ERM baselines considered for the experiment presented in Tables 2, 3, and 4, and Figure 3 we use a multilayer perceptron (MLP) with one hidden layer of size 100 with ReLU activations. We train for 200 epochs with a batch size of 128 using stochastic gradient descent (SGD) with a learning rate of 0.01 that is reduced by a factor of ten if the training loss has not improved by at least 0.01 in the last 20 epochs, with a minimum learning rate of 10^{-7}. We use a weight decay of 10^{-6}. The training procedure is implemented using Tensorflow 2.12.0.

For the covariate shift adjustment baseline, we fit a domain classifier, using the same model architecture and training procedure in the model for Y, derive instance weights following Shimodaira (2000), and apply weighted ERM with the same procedure as in the unweighted case. The label shift baseline with oracle access to labels in the target domain (LABEL) applies weighted ERM with learned class weights $q(Y)/p(Y)$ based on observed frequencies in the validation set in the

Algorithm 2 Estimating Continuous $q(Y|x_{\text{new}})$.

Require: source $\{(w_i, x_i, c_i, y_i)\}_{i=1}^n$; target $\{\widetilde{x}_j\}_{j=1}^m$; Given x_{new}

1: $\widehat{p}(W|\widetilde{U}) \leftarrow$ Algorithm 3$(\{(w_i, x_i, c_i, y_i)\}_{i=1}^n)$
2: $[\widehat{q}(\widetilde{\mathbf{U}})/\widehat{p}(\widetilde{\mathbf{U}})] \leftarrow$ Algorithm 4$(\{(w_i, x_i)\}_{i=1}^n,\ \{\widetilde{x}_j\}_{j=1}^m,\ \{\widehat{p}(W|\widetilde{U}=k)\}_{k=1}^{k_U})$
3: $\widehat{p}(\widetilde{\mathbf{U}} \mid x_{\text{new}})$ is obtained by solving (17)
4: $\widehat{p}(\mathbf{Y} \mid \widetilde{\mathbf{U}}, x_{\text{new}}) \leftarrow$ Algorithm 5$(\{(w_i, x_i, y_i)\}_{i=1}^n,\ \{\widehat{p}(W|\widetilde{U}=k)\}_{k=1}^{k_U}, \widehat{p}(\widetilde{\mathbf{U}} \mid x_{\text{new}}))$
5: **for** $y = 1, \ldots, k_Y$ **do**
6: $\quad \widehat{q}(y \mid x_{\text{new}}) \leftarrow \sum_{i=1}^{k_U} \widehat{p}(y \mid x_{\text{new}}, \widetilde{U}=i)\widehat{p}(\widetilde{U}=i \mid x_{\text{new}})\frac{q(\widetilde{U})}{p(\widetilde{U})}$
7: $\widehat{q}(\mathbf{Y} \mid x_{\text{new}}) \leftarrow \widehat{q}(\mathbf{Y} \mid x_{\text{new}}) / \sum_{i=1}^{k_y} \widehat{q}(y \mid x_{\text{new}})$

source domain and the training set in the target domain. For the BBSE approach (Lipton et al., 2018), where the labels Y are not available in the target domain, we first fit an auxiliary model with ERM to estimate $p(Y \mid X)$ in the source domain, using the same procedure as before, and use its predictions on the source validation set and target training set to estimate $q(Y)/p(Y)$ using the soft confusion matrix approach of (Garg et al., 2020). We clip weights derived from the confusion matrix approach to the range $[0.01, 15]$. For the adjustment procedure that implements equation (1) with oracle access to U (LSA-ORACLE), we fit auxiliary models for $p(Y \mid X, U)$ and $p(U \mid X)$, using the model for $p(U \mid X)$ directly in equation (1) and as the predictor used to derive $q(U)/p(U)$ with the soft confusion matrix approach. We apply temperature scaling as an additional calibration step to each auxiliary model and the final result of each procedure. The temperature scaling procedure is implemented as a uniform scaling of the output logits by a scalar learned on the validation data using SGD with a fixed learning rate of 0.001.

For the WAE-based adaptation approach, we use an encoder with one hidden layer of size 100 and set the dimensionality of the learned latent space over $\widetilde{U}$ to be 10. Following the construction in section 6, we use a model architecture and objective function that reflects the factorization of the joint distribution implied by the causal graph (LSA-WAE-S). For this approach, we use separate decoder networks $\{f_Y, f_C, f_X, f_W\}$ of one hidden layer of size 100 for each of the observed variables. We use categorical cross-entropy losses over the reconstruction of Y and W and the elementwise binary cross-entropy loss over the elements of C. The loss ℓ_X over X is given by $\log(\sigma_X) + \frac{1}{\sigma_X}(X - f_X(\widetilde{U}))^2$, where σ_X is a learned parameter. The weight β on reconstruction loss associated with each of C, W, and Y is the reciprocal of the entropy of the variable, estimated on the training data of the source domain, and the weight β_X is analogously the reciprocal of the variance of X. The KL divergence term in the loss is weighted by a factor of 3. The WAE is fit using the RMSprop optimizer for 200 epochs using a learning rate of 10^{-4}, annealed with the same strategy as in the baseline approaches. We anneal the temperature of the Gumbel-softmax distribution used for sampling $\widetilde{U}$ by a factor of 0.9999 at each training iteration, starting from an initial temperature of 1 to a minimum temperature of 0.01.

C ESTIMATION PROCEDURE FOR CONTINUOUS RANDOM VARIABLES

In this section, we introduce the estimation procedure for continuous random variables, an extension of Algorithm 1. The continuous setting requires an additional step to select k_U points from the domain of X such that the constructed confusion matrix is invertible. While there exist various density function estimators, the main challenge is finding a reliable density estimator for computing of the underlying eigenfunctions. To this end, we employ the Least-Squares Conditional Density Estimator (LS-CDE) (Sugiyama et al., 2010), where the set of basis functions are pre-defined by users. This method allows us to easily compute the eigenfunctions of the underlying density operators, which, in turn are a finite set of basis functions. The complete estimation procedure is presented in Algorithm 2. The algorithm is implemented for discrete U, Y, C and continuous X, W, which matches the simulation setting in Section 7. We first briefly introduce the LS-CDE method and discuss the selection of basis functions, followed by the details of each step.

C.1 Brief introduction of least-squares conditional density estimator

Given a pair of random variables (X, Y), the Least-Squares Conditional Density Estimator (LS-CDE)) (Sugiyama et al., 2010) assumes the following form

$$p(Y \mid X) = \frac{p(X, Y)}{p(X)} := r(X, Y),$$

where $r(X, Y)$ is the density ratio function. Let $\{g_1(x, y), \ldots, g_m(x, y)\}$ to be a set of basis functions such that (1) $g_i(x, y) \geq 0$ for every $i \in [m]$ and $x \in \text{Dom}(X)$ and $y \in \text{Dom}(Y)$. To estimate $r(X, Y)$, we consider the estimate

Algorithm 3 Estimate Continuous $p(W|U)$, details provided in Section C.2

Require: source $\mathcal{P}=\{(w_i,x_i,c_i,y_i)\}_{i=1}^n$; Given $c\in\mathrm{Dom}(C)$ and $y\in\mathrm{Dom}(Y)$
1: $\widehat{p}(W,X\mid c)$ and $\widehat{p}(W,X,y\mid c)$ via least-squares density estimator (12)–(13)
2: Find the decomposition $\widehat{p}(W,X\mid c)$ and $\widehat{p}(W,X,y\mid c)$ (14)
3: Eigendecompose $\widehat{A}^{-1}\widehat{B}'$ (15) and obtain the eigenfunctions
4: Compute the inverse of the eigenfunctions to obtain $\{\widehat{p}(W|\widetilde{U}=1),\ldots,\widehat{p}(W|\widetilde{U}=k_U)\}$

$\widehat{r}_{\boldsymbol{\alpha}}(X,Y)$ that lies in the linear subspace of $r(x,y)\in\{\boldsymbol{\alpha}^\top\mathbf{g}(x,y):\boldsymbol{\alpha}\in\mathbb{R}^m\}$ with $\mathbf{g}(x,y)=(g_1(x,y),\ldots,g_m(x,y))$. Hence, the goal is to estimate the coefficient vector $\boldsymbol{\alpha}$ from data. To this end, Sugiyama et al. (2010) proposed the following objective functional:

$$\arg\min\frac{1}{2}\int\int\left(\boldsymbol{\alpha}^\top\mathbf{g}(x,y)-\frac{p(x,y)}{p(x)}\right)^2p(x)dxdy.$$

With simple algebraic manipulation, the above objective function is equivalent as the following:

$$\widehat{\boldsymbol{\alpha}}=\arg\min\frac{1}{2}\boldsymbol{\alpha}^\top\mathbf{H}\boldsymbol{\alpha}-\mathbf{h}^\top\boldsymbol{\alpha},$$

where

$$\mathbf{H}:=\int\int\mathbf{g}(x,y)\mathbf{g}(x,y)^\top p(x)dydx,\quad\mathbf{h}:=\int\int\mathbf{g}(x,y)p(x,y)dxdy.$$

Since the density functions $p(x,y)$ and $p(x)$ are unknown, we can compute the empirical estimators of $\mathbf{H}$ and $\mathbf{h}$ from independent samples $\{(x_i,y_i)\}_{i=1}^n$ as follows:

$$\widehat{\mathbf{H}}:=\frac{1}{n}\sum_{i=1}^n\int\mathbf{g}(x_i,y)\mathbf{g}(x_i,y)^\top dy,\quad\widehat{\mathbf{h}}:=\frac{1}{n}\sum_{i=1}^n\mathbf{g}(x_i,y_i).$$

To stabilize the empirical estimator, we additionally add a regularizer $\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha}$ with $\lambda>0$. The overall objective function is summarized as

$$\widetilde{\boldsymbol{\alpha}}:=\arg\min\frac{1}{2}\boldsymbol{\alpha}^\top\widehat{\mathbf{H}}\boldsymbol{\alpha}-\widehat{\mathbf{h}}^\top\boldsymbol{\alpha}+\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha}.\tag{10}$$

Note that (10) is a quadratic program and yields an analytical solution

$$\widetilde{\boldsymbol{\alpha}}=(\widehat{\mathbf{H}}+\lambda\mathbf{I})^{-1}\widehat{\mathbf{h}}.$$

To ensure the the estimated conditional density is non-negative everywhere, we output $\widehat{\boldsymbol{\alpha}}=(\widehat{\alpha}_1,\ldots,\widehat{\alpha}_m)$ such that $\widehat{\alpha}_i=\max(0,\widetilde{\alpha}_i)$ for $i\in[m]$. In our simulations, we found that choosing $\lambda=10^{-2}$ suffices to provide good results.

Choosing the candidate basis functions requires knowledge of the underlying distributions. When the class of distribution is unknown, Gaussian kernel functions can often be used as a basis to provide a good approximation of the distribution(s). In addition, the Gaussian kernel function yields and analytical result for the integral $\widehat{\mathbf{H}}$. Specifically, let $g_\ell(x,y)=\exp(-\|(x-x_\ell\|^2+\|y-y_\ell\|^2)/2\sigma^2)$ and $g_{\ell'}(x,y)=\exp(-(\|x-x_{\ell'}\|^2+\|y-y_{\ell'}\|^2)/2\sigma^2)$ for some $x_\ell,x_{\ell'}\in\mathrm{Dom}(X)$, $y_\ell,y_{\ell'}\in\mathrm{Dom}(Y)$ and $\sigma>0$, we have

$$\int g_\ell(x,y)g_{\ell'}(x,y)dy=(\sqrt{\pi}\sigma)^{d_y}\exp\left(-\frac{\|y_\ell-y_{\ell'}\|^2}{4\sigma^2}\right)\exp\left(-\frac{\|x-x_{\ell'}\|^2+\|x-x_\ell\|^2}{2\sigma^2}\right),$$

where d_y is the dimension of Y. Hence, we do not need to resort to numerical methods to compute $\widehat{\mathbf{H}}$.

C.2 Implementation details of Algorithm 3

In this section, we introduce the implementation details of Algorithm 3 step-by-step.

Step 1 of Algorithm 3. Since both Y,C are discrete random variables, for any fixed $c\in\mathrm{Dom}(C),y\in\mathrm{Dom}(Y)$, the conditional density functions $\widehat{p}(w,x\mid c)$ and $\widehat{p}(w,x,y\mid c)$ can be estimated by marginal density estimators. We use the

least-squares density estimator to estimate both $\widehat{p}(w, x \mid c)$ and $\widehat{p}(w, x, y \mid c)$ with Gaussian kernel basis functions of length-scale 1:

$$\left\{ g_\ell(w,x) = \varphi_\ell(w)\vartheta_\ell(x) : \varphi_\ell(w) = \exp\left(-\frac{\|w-\bar{w}_\ell\|^2}{2}\right), \vartheta_\ell(x) = \exp\left(-\frac{\|x-\bar{x}_\ell\|^2}{2}\right), \ell = 1, \ldots, k_U \right\}, \tag{11}$$

where the centers $\bar{x}_\ell$, $\bar{w}_\ell$ for $\ell = 1, \ldots, m$ are chosen to match the means of the mixture models from the data generation process, namely $\mathbf{M}_{W|U}$ and $\mathbf{M}_{X|U}$ defined in Section B. Here, we assume the density functions have the following form

$$p(w, x|c) = \boldsymbol{\alpha}_c^\top \mathbf{g}(w,x), \quad p(w, x, y \mid c) = p(w, x \mid y, c)p(y \mid c) = \boldsymbol{\beta}_{y,c}^\top \mathbf{g}(w,x)p(y \mid c),$$

where the conditional probability $p(y \mid c)$ can be seen as a constant given that y, c are fixed. Hence, it is natural to assume that the empirical marginal density estimator has the following form

$$\widehat{p}(w, x|c) = \widehat{\boldsymbol{\alpha}}_c^\top \mathbf{g}(w,x), \quad \widehat{p}(w, x \mid y, c) = \widehat{\boldsymbol{\beta}}_{y,c}^\top \mathbf{g}(w,x),$$

We obtain coefficient vectors $\widehat{\boldsymbol{\alpha}}_c$ and $\widehat{\boldsymbol{\beta}}_{y,c}$ by solving a similar objective function as LS-CDE:

$$\boldsymbol{\alpha}_c = \operatorname{argmin} \frac{1}{2} \int\int \left(\boldsymbol{\alpha}_c^\top \mathbf{g}(w,x) - p(w, x \mid c)\right)^2 dwdx;$$
$$\boldsymbol{\beta}_{y,c} = \operatorname{argmin} \frac{1}{2} \int\int \left(\boldsymbol{\beta}_{y,c}^\top \mathbf{g}(w,x) - p(w, x \mid y, c)\right)^2 dwdx.$$

Given subsets of samples $\{(x_i, w_i, c_i)\}_{i \in \mathcal{N}_c}$ with $\mathcal{N}_c = \{i \in [n] : c_i = c\}$ and $\{(x_i, y_i, w_i, c_i)\}_{i \in \mathcal{N}_{y,c}}$ with $\mathcal{N}_{y,c} = \{i \in [n] : y_i = y, c_i = c\}$ from the original sample set $\{(x_i, y_i, w_i, c_i)\}_{i=1}^n$, we can construct the associated regularized empirical estimators. Define $\widetilde{\mathbf{H}} = \int\int \mathbf{g}(x,y)\mathbf{g}(x,y)^\top dxdy$, $\widehat{\boldsymbol{\alpha}}_c$, $\widehat{\boldsymbol{\beta}}_{y,c}$ are obtained by solving the following function

$$\widehat{\boldsymbol{\alpha}}_c = \arg\min \frac{1}{2}\boldsymbol{\alpha}^\top \widetilde{\mathbf{H}}\boldsymbol{\alpha} - \widehat{\mathbf{h}}_c^\top \boldsymbol{\alpha} + \lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha}, \quad \widehat{\mathbf{h}}_c = \frac{1}{|\mathcal{N}_c|} \sum_{i \in \mathcal{N}_c} \mathbf{g}(w_i, x_i); \tag{12}$$

$$\widehat{\boldsymbol{\beta}}_{y,c} = \arg\min \frac{1}{2}\boldsymbol{\beta}^\top \widetilde{\mathbf{H}}\boldsymbol{\beta} - \widehat{\mathbf{h}}_{y,c}^\top \boldsymbol{\beta} + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta}, \quad \widehat{\mathbf{h}}_{y,c} = \frac{1}{|\mathcal{N}_{y,c}|} \sum_{i \in \mathcal{N}_{y,c}} \mathbf{g}(w_i, x_i). \tag{13}$$

It is worth noting that the integral of Gaussian kernel functions $\widetilde{\mathbf{H}} = \int\int \mathbf{g}(w,x)\mathbf{g}(w,x)^\top dxdw$ has an analytical form and the objective function is quadratic, yielding analytical forms $\widehat{\boldsymbol{\alpha}}_c = (\widetilde{\mathbf{H}} + \lambda \mathbf{I})^{-1}\widehat{\mathbf{h}}_c$ and $\widehat{\boldsymbol{\beta}}_{y,c} = (\widetilde{\mathbf{H}} + \lambda \mathbf{I})^{-1}\widehat{\mathbf{h}}_{y,c}$.

Step 2–3 of Algorithm 3. With the estimated from *Step 1*, we can construct the empirical integral operator $\widehat{A}$ and $\widehat{B}'$ with respect to the kernel functions $\widehat{p}(w, x \mid c)$ and $\widehat{p}(w, x \mid y, c)$, respectively, as

$$\widehat{A} = \sum_{i=1}^{k_U} \widehat{\alpha}_{c,i}\vartheta_i(X) \otimes \varphi_i(W), \quad \widehat{B}' = \sum_{i=1}^{k_U} \widehat{\beta}_{y,c,i}\vartheta_i(X) \otimes \varphi_i(W). \tag{14}$$

To find the inverse of $\widehat{A}$, we first run the Gram-Schmidt procedure on $\{\vartheta_1, \ldots, \vartheta_{k_U}\}$ and $\{\varphi_1, \ldots, \varphi_{k_U}\}$ respectively to orthonormalize the basis functions. Since we are using Guassian kernels, the Gram-Schmidt procedure can be obtained analytically. We provide the example of constructing the first two orthonormal components and the rest of them can be constructed similarly. We have

$$\overline{\vartheta}_1 = \frac{\vartheta_1}{\|\vartheta_1\|}, \qquad \langle \vartheta_1, \vartheta_1 \rangle = \int \vartheta_1(x)\vartheta_1(x)dx = (\sqrt{\pi})^{d_x};$$
$$\overline{\vartheta}_2 = \frac{\vartheta_2 - \langle \vartheta_2, \overline{\vartheta}_1 \rangle}{\|\vartheta_2 - \langle \vartheta_2, \overline{\vartheta}_1 \rangle\|}, \qquad \langle \vartheta_2, \overline{\vartheta}_1 \rangle = \int \overline{\vartheta}_1(x)\vartheta_2(x)dx = (\sqrt{\pi})^{d_x/2} \exp\left(-\frac{\|\bar{x}_1 - \bar{x}_2\|^2}{4}\right).$$

Let $\mathbf{R}_\vartheta \in \mathbb{R}^{k_U \times k_U}$ be a coefficient matrix whose ij-th entry is $\langle \overline{\vartheta}_i, \vartheta_j \rangle$ and $\mathbf{R}_\varphi \in \mathbb{R}^{k_U \times k_U}$ be a coefficient matrix whose ij-th entry is $\langle \overline{\varphi}_i, \varphi_j \rangle$. Then, it follows that

$$\widehat{A}^{-1} = \overline{\boldsymbol{\varphi}}(\mathbf{R}_\varphi^\top)^{-1} \begin{bmatrix} \widehat{\alpha}_{c,1} & & \\ & \ddots & \\ & & \widehat{\alpha}_{c,k_U} \end{bmatrix}^{-1} \mathbf{R}_\vartheta^{-1} \overline{\boldsymbol{\vartheta}}^\top, \quad \widehat{B}' = \overline{\boldsymbol{\vartheta}}\mathbf{R}_\vartheta \begin{bmatrix} \widehat{\beta}_{y,c,1} & & \\ & \ddots & \\ & & \widehat{\beta}_{y,c,k_U} \end{bmatrix} \mathbf{R}_\varphi \overline{\boldsymbol{\varphi}}^\top.$$

Algorithm 4 Estimate $q(\widetilde{U})/p(\widetilde{U})$, details provided in Section C.3

Require: source $\mathcal{P}=\{(w_i,x_i)\}_{i=1}^n$; target $\mathcal{Q}=\{x_j\}_{j=1}^m$; $\{\widehat{p}(W|U=1),\ldots,\widehat{p}(W|U=k_U)\}$;
1: Compute $\widehat{p}(W \mid X)$ via LS-CDE (Sugiyama et al., 2010)
2: Run K-means clustering to select k_U centers: $x_1,\ldots,x_{k_U}$
3: **for** x in $\{x_1,\ldots,x_{k_U}\}$ **do**
4: $\quad\widehat{p}(\widetilde{\mathbf{U}}|X=x)$ is obtained by solving (17)
5: $[\widehat{q}(\widetilde{\mathbf{U}})/\widehat{p}(\widetilde{\mathbf{U}})]$ is obtained by solving (18)

Hence, we can obtain

$$\widehat{A}^{-1}\widehat{B}' = \overline{\boldsymbol{\varphi}}\,(\mathbf{R}_{\varphi}^{\top})^{-1}\underbrace{\begin{bmatrix}\widehat{\alpha}_{c,1} & & \\ & \ddots & \\ & & \widehat{\alpha}_{c,k_U}\end{bmatrix}^{-1}\begin{bmatrix}\widehat{\beta}_{y,c,1} & & \\ & \ddots & \\ & & \widehat{\beta}_{y,c,k_U}\end{bmatrix}}_{\mathbf{M}}\mathbf{R}_{\varphi}\,\overline{\boldsymbol{\varphi}}^{\top}, \tag{15}$$

where the eigenfunctions of $\widehat{A}^{-1}\widehat{B}'$ are obtained by first computing the eigenvectors of $\mathbf{M}$, denoted as $\widehat{\boldsymbol{\eta}}_1,\ldots,\widehat{\boldsymbol{\eta}}_{k_U}$ and then projecting them to the basis functions $\{\overline{\varphi}_1,\ldots,\overline{\varphi}_{k_U}\}$. That is, the j-th eigenfunction of $\widehat{A}^{-1}\widehat{B}'$ is $\sum_{i=1}^{k_U}\widehat{\eta}_{j,i}\overline{\varphi}_i(w)$.

Step 4 of Algorithm 3. Let $\widehat{\mathbf{D}} = \left[\widehat{\mathbf{d}}_1\cdots\widehat{\mathbf{d}}_{k_U}\right] = \left[\widehat{\boldsymbol{\eta}}_1\cdots\widehat{\boldsymbol{\eta}}_{k_U}\right]^{-1}$, by proof of Theorem 2, the estimate of $p(w \mid \widetilde{U}=j)$ is $\sum_{i=1}^{k_U}\widehat{d}_{j,i}\overline{\varphi}_i(w)/\|\sum_{i=1}^{k_U}\widehat{d}_{j,i}\overline{\varphi}_i\|_{L_1}$.

C.3 Implementation details of Algorithm 4

The first step of Algorithm 4 is implemented by LS-CDE introduced in Appendix C.1 with the set of basis functions defined in (11) and $\lambda = 10^{-2}$. The second step is straightforward. Hence, we only discuss the implementation details of Step 4 in Algorithm 4. The identification result (8) suggests the construction of the following program

$$\begin{aligned}\widehat{p}(\widetilde{\mathbf{U}} \mid X=x) = \arg\min \quad & \left\|\widehat{p}(W \mid X=x) - \sum_{k=1}^{k_U}\widehat{p}(W \mid \widetilde{U}=i)p(\widetilde{U}=k \mid X=x)\right\|_{L_2}^2 \\ \text{subject to} \quad & 0 \le p(\widetilde{U}=i \mid x) \le 1, \quad i=1,\ldots,k_U; \\ & \sum_{i=1}^{k_U} p(\widetilde{U}=i \mid x) = 1.\end{aligned} \tag{16}$$

Define the design matrix $\mathbf{G} \in \mathbb{R}^{k_U\times k_U}$:

$$\mathbf{G} = \begin{bmatrix}\langle\widehat{p}(W \mid \widetilde{U}=1),\widehat{p}(W \mid \widetilde{U}=1)\rangle & \cdots & \langle\widehat{p}(W \mid \widetilde{U}=1),\widehat{p}(W \mid \widetilde{U}=k_U)\rangle \\ \vdots & \ddots & \vdots \\ \langle\widehat{p}(W \mid \widetilde{U}=k_U),\widehat{p}(W \mid \widetilde{U}=1)\rangle & \cdots & \langle\widehat{p}(W \mid \widetilde{U}=k_U),\widehat{p}(W \mid \widetilde{U}=k_U)\rangle\end{bmatrix}.$$

Given $x \in \mathrm{Dom}\,X$, since $p(\widetilde{U} \mid x)$ is discrete random variable with k_U states, we can reformulate (16) as

$$\begin{aligned}\widehat{p}(\widetilde{\mathbf{U}} \mid x) = \arg\min \quad & \left\|\begin{bmatrix}\langle\widehat{p}(W \mid x),\widehat{p}(W \mid \widetilde{U}=1)\rangle \\ \vdots \\ \langle\widehat{p}(W \mid x),\widehat{p}(W \mid \widetilde{U}=k_U)\rangle\end{bmatrix} - \mathbf{G}\begin{bmatrix}p(\widetilde{U}=1 \mid x) \\ \vdots \\ p(\widetilde{U}=k_U \mid x)\end{bmatrix}\right\|_F^2, \\ \text{subject to} \quad & 0 \le p(\widetilde{U}=i \mid x) \le 1, \quad i=1,\ldots,k_U; \\ & \sum_{i=1}^{k_U} p(\widetilde{U}=i \mid x) = 1,\end{aligned} \tag{17}$$

which is a constrained least-squares problem and can be optimized efficiently by sequential least-squares programming.

Algorithm 5 Estimate $p(Y \mid x_{\text{new}}, \widetilde{U})$, details provided in Section C.4

Require: source $\mathcal{P} = \{(w_i, y_i, x_i)\}_{i=1}^n$, $\widehat{p}(\widetilde{\mathbf{U}} \mid x_{\text{new}})$
1: Compute $\widehat{p}(Y \mid X)$ using MLP
2: **for** $y = 1, \ldots, k_Y$ **do**
3: Estimate $p(W \mid X, y)$ via LS-CDE (Sugiyama et al., 2010)
4: Compute $\widehat{p}(W, y \mid x_{\text{new}}) = \widehat{p}(W \mid X, y)\widehat{p}(y \mid x_{\text{new}})$
5: Compute $\widehat{p}(\mathbf{Y} \mid \widetilde{\mathbf{U}}, x_{\text{new}})$ by solving (19).

Finally, to compute the vector $q(\widetilde{\mathbf{U}})/p(\widetilde{\mathbf{U}})$, we need to estimate the marginal density $p(x)$ and $q(x)$. This can be implemented through a similar approach as introduced in *Step 1* in Appendix C.2. We briefly introduce the procedure to estimate $p(x)$; The estimation procedure of $q(x)$ follows similarly. Consider the subspace spanned by Gaussian kernel basis functions of length-scale 1,

$$\left\{\vartheta_\ell(x) : \vartheta_\ell(x) = \exp\left(-\frac{\|x - \bar{x}_\ell\|^2}{2}\right), \ell = 1, \ldots, k_U\right\}.$$

We assume that the distribution is of the form $p(x) = \boldsymbol{\alpha}^\top \boldsymbol{\vartheta}(x)$ with $\boldsymbol{\vartheta}(x) = \big[\vartheta_1(x) \cdots \vartheta_{k_U}(x)\big]$. Hence, it follows that $\boldsymbol{\alpha}$ minimizes $\int (\boldsymbol{\alpha}^\top \boldsymbol{\vartheta}(x) - p(x))^2 dx$. Then, it is natural to formulate the empirical estimator as $\widehat{p}(x) = \widehat{\boldsymbol{\alpha}}^\top \boldsymbol{\vartheta}(x)$, where we can obtain $\widehat{\boldsymbol{\alpha}}$ by solving the following problem:

$$\widetilde{\boldsymbol{\alpha}} = \arg\min \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H}_x \boldsymbol{\alpha} - \widehat{\mathbf{h}}_x^\top \boldsymbol{\alpha} + \lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha},$$

where $\mathbf{H}_x = \int \boldsymbol{\vartheta}(x)\boldsymbol{\vartheta}(x)^\top dx$, and $\widehat{\mathbf{h}}_x = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\vartheta}(x_i)$. Then we set $\widehat{\alpha}_i = \max(0, \widetilde{\alpha}_i)$ for $i = 1, \ldots, k_U$ to ensure the non-negativity of the distribution.

After estimating $\widehat{q}(x)$ and $\widehat{p}(x)$, we construct the vector $[\widehat{q}(\mathbf{x})/\widehat{p}(\mathbf{x})]^\top = [\widehat{q}(x_1)/\widehat{p}(x_1) \cdots \widehat{q}(x_{k_U})/\widehat{p}(x_{k_U})]^\top$ by querying $x_1, \ldots x_k$ from $\widehat{q}(x)$ and $\widehat{p}(x)$. Then we can obtain $\widehat{q}(\widetilde{\mathbf{U}})/\widehat{p}(\widetilde{\mathbf{U}})$ by solving the following constrained least-squares problem:

$$[\widehat{q}(\widetilde{\mathbf{U}})/\widehat{p}(\widetilde{\mathbf{U}})] = \arg\min \left\| \begin{bmatrix} \frac{\widehat{q}(x_1)}{\widehat{p}(x_1)} \\ \vdots \\ \frac{\widehat{q}(x_{k_U})}{\widehat{p}(x_{k_U})} \end{bmatrix} - \begin{bmatrix} p(\widetilde{U}=1 \mid x_1) & p(\widetilde{U}=2 \mid x_1) & \cdots & p(\widetilde{U}=k \mid x_1) \\ p(\widetilde{U}=1 \mid x_2) & p(\widetilde{U}=2 \mid x_2) & \cdots & p(\widetilde{U}=k \mid x_2) \\ \vdots & \ddots & & \vdots \\ p(\widetilde{U}=1 \mid x_k) & p(\widetilde{U}=2 \mid x_k) & \cdots & p(\widetilde{U}=k \mid x_k) \end{bmatrix} [q(\widetilde{\mathbf{U}})/p(\widetilde{\mathbf{U}})]^\top \right\|_F^2 \tag{18}$$

$$\text{subject to} \quad [q(\widetilde{\mathbf{U}})/p(\widetilde{\mathbf{U}})]_i \geq 0, \quad i = 1, \ldots, k_U.$$

C.4 Implementation details of Algorithm 5

In this section, we introduce the implementation details of Algorithm 5. First, we learn the distribution of $p(y|x)$ by fitting a Multi-Layer Perceptron classifier (MLP) with 'ReLU' activation function attached at the output of the hidden layers. Given a fixed y, we estimate $p(W \mid X, y)$ by first constructing the subset of samples $\{x_i, w_i\}_{i \in \mathcal{N}_y}$ such that $\mathcal{N}_y = \{i \in [n] : y_i = y\}$. Then, $p(W \mid X, y)$ is estimated by fitting LS-CDE (Sugiyama et al., 2010) with $\{x_i, w_i\}_{i \in \mathcal{N}_y}$ and the basis functions (11).

To estimate $p(Y \mid \widetilde{U}, X)$, we first recall the relation of $p(W \mid \widetilde{U})$, $p(\widetilde{U} \mid X)$, $p(W, Y \mid X)$ and $p(Y \mid \widetilde{U}, X)$ defined in (9). Computing the inverse of the matrix might lead to numerical instability in practice and hence we solves a constrained least-squares problem as an alternative. Given estimated $\{\widehat{\phi}_i = \widehat{p}(W \mid \widetilde{U} = i)\}_{i=1}^{k_U}$, we run the Gram-Schmidt procedure to obtain $\widehat{\boldsymbol{\phi}}_{\text{ortho}} = [\widehat{\phi}_{\text{ortho},1}, \ldots, \widehat{\phi}_{\text{ortho},k_U}]$ and $\widehat{\mathbf{R}}_\phi$. Let $\widehat{\mathbf{R}} = \widehat{\mathbf{R}}_\phi \text{diag}(\widehat{p}(\widetilde{U}=1 \mid x_{\text{new}}), \ldots, \widehat{p}(\widetilde{U}=k_U \mid x_{\text{new}}))$. Then, we

estimate $p(\mathbf{Y} \mid \widetilde{\mathbf{U}}, x_{\text{new}}) \in \mathbb{R}^{k_U \times k_Y}$ by solving the following constrained optimization problem

$$\widehat{p}(\mathbf{Y} \mid \widetilde{\mathbf{U}}, x_{\text{new}}) =$$

$$\arg\min \left\| \begin{bmatrix} \langle p(Y=1, W \mid x_{\text{new}}), \widehat{\phi}_{\text{ortho},1} \rangle & \cdots & \langle p(Y=k_Y, W \mid x_{\text{new}}), \widehat{\phi}_{\text{ortho},1} \rangle \\ \vdots & \ddots & \vdots \\ \langle p(Y=1, W \mid x_{\text{new}}), \widehat{\phi}_{\text{ortho},k_U} \rangle & \cdots & \langle p(Y=k_Y, W \mid x_{\text{new}}), \widehat{\phi}_{\text{ortho},k_U} \rangle \end{bmatrix} - (\mathbf{I} \otimes_k \widehat{\mathbf{R}}) p(\mathbf{Y} \mid \widetilde{\mathbf{U}}, x_{\text{new}}) \right\|_F^2 \tag{19}$$

$$\text{subject to} \quad 0 \le p(Y = y \mid \widetilde{U} = i, x_{\text{new}}) \le 1, \quad y = 1, \dots k_Y, i = 1, \dots, k_U;$$

$$\sum_{y=1}^{k_Y} p(Y = y \mid \widetilde{U} = i) = 1, \quad i = 1, \dots, k_U,$$

where $\otimes_k$ denotes the Kroneker product. This completes the procedure.

D DEEP LATENT VARIABLE MODEL SETUP

As described in Section 6, we approximate the joint distribution $p(X, Y, C, W, \widetilde{U})$ using a model based on the Wasserstein Auto-Encoder (WAE; Tolstikhin et al., 2018). The overall algorithm is broken down into five main steps as shown below:

High-Level Pseudo-code

1. Train the WAE.
2. Use the WAE's encoder to append $\widetilde{U}$ to the source dataset: $\{(x_i, y_i, c_i, w_i)\}_{i=1}^n \to \{(x_i, y_i, c_i, w_i, \widetilde{u}_i)\}_{i=1}^n$.
3. Train $p(\widetilde{U} \mid X)$ and $p(Y \mid X, \widetilde{U})$ using the dataset $\{(x_i, y_i, c_i, w_i, \widetilde{u}_i)\}_{i=1}^n$.
4. Estimate the likelihood ratios $q(\widetilde{U})/p(\widetilde{U})$ using the confusion matrix approach of (Lipton et al., 2018) and $p(\widetilde{U} \mid X)$.
5. Predict $q(Y \mid X)$ using (1).

We describe, next, how this approach works in detail.

D.1 Training the WAE

First, we approximate the latent variable $\widetilde{U}$. For that, we construct a variant of WAE, in which the assumptions of the graph in Figure 1(c) are imposed. Specifically, while the encoder $p(\widetilde{U} \mid X, C, Y, W)$ is an MLP $(X, Y, C, W) \to \widetilde{U}$ with parameters ϕ, the decoder has the structure: $\widetilde{U} \to X$, $\widetilde{U} \to W$, $(\widetilde{U}, X) \to C$, and $(\widetilde{U}, C) \to Y$, where each arrow is a separate MLP model with its own parameters, leading to the factorization:

$$p(\mathcal{V}, \widetilde{U}) = p(Y \mid C, \widetilde{U})\, p(C \mid X, \widetilde{U})\, p(X \mid \widetilde{U})\, p(W \mid \widetilde{U})\, p(\widetilde{U}),$$

where $\mathcal{V} = (X, Y, C, W)$ as discussed in Section 6. The WAE is trained to minimize the reconstruction loss and the KL-divergence between $p(\widetilde{U})$ and its prior $\overline{p}(\widetilde{U})$ as shown in (3), where p is averaged over the entire batch. In our experiments, the reconstruction loss is the mean square error (MSE) for X, cross-entropy for Y and W (because both are one-hot encoded), and the binary cross-entropy for every concept in C (because C is multi-label). We set the number of latent categories $|\widetilde{U}|$ in the WAE to 10. All MLPs follow the architecture described in Appendix B.

As $\widetilde{U}$ is discrete, to allow training with the reparameterization trick, we model $p(\widetilde{U} \mid X, C, Y, W)$ using a Gumbel-Softmax distribution (Jang et al., 2016; Maddison et al., 2016). We set the prior $\overline{p}(\widetilde{U})$ to be a uniform categorical distribution over the categories of $\widetilde{U}$.

D.2 Append the latent category $\widetilde{U}$

Given a trained WAE model, we next generate joint samples $\{(x_i, c_i, y_i, w_i, \widetilde{u}_i)\}_{i=1}^n$ using the encoder $p(\widetilde{U} \mid X, C, Y, W)$. Specifically, for every tuple in the training set (x, y, c, w), we generate $\widetilde{u} \sim p(\widetilde{U} \mid X = x, C = c, Y = y, W = w)$ and append $\widetilde{u}$ to the tuple (x, y, c, w).

D.3 Training $p(\widetilde{U} \mid X)$ and $p(Y \mid X, \widetilde{U})$

Given the dataset $\{(x_i, c_i, y_i, w_i, \widetilde{u}_i)\}_{i=1}^n$, we train a model $p(\widetilde{U} \mid X)$ and another model $p(Y \mid X, \widetilde{U})$. In our experiments, both models are MLPs (of the same architecture specified in Appendix B), which are trained by minimizing the cross-entropy loss. After training, we calibrate on the separate hold-out dataset using temperature scaling (Guo et al., 2017).

D.4 Likelihood Ratios

Next, we employ the confusion matrix approach of (Lipton et al., 2018) to estimate the likelihood ratios $q(\widetilde{U})/p(\widetilde{U})$ by applying it on the model $p(\widetilde{U}|X)$. Specifically, since $p(\widetilde{U}|X)$ is trained on source data, we calculate the confusion matrix on source. Then, we run the model on unlabeled X from the target domain q and calculate its mean predictions. After that, we use Proposition 2 in (Lipton et al., 2018) to estimate the likelihood ratios.

D.5 Inference

Finally, during inference, we use the two models $p(\widetilde{U} \mid X)$ and $p(Y \mid X, \widetilde{U})$ trained in the third step and the likelihood ratios $q(\widetilde{U})/p(\widetilde{U})$ obtained in the forth step, and predict $q(Y|X)$ using (1).