Data-Driven Approach to Multiple-Source Domain Adaptation

Petar Stojanov
Computer Science Department,
Carnegie Mellon University

Mingming Gong
University of Pittsburgh
Carnegie Mellon University

Jaime G. Carbonell
Language Technologies Institute,
Carnegie Mellon University

Kun Zhang
Philosophy Department,
Carnegie Mellon University

Abstract

A key problem in domain adaptation is determining what to transfer across different domains. We propose a data-driven method to represent these changes across multiple source domains and perform unsupervised domain adaptation. We assume that the joint distributions follow a specific generating process and have a small number of identifiable changing parameters, and develop a data-driven method to identify the changing parameters by learning low-dimensional representations of the changing class-conditional distributions across multiple source domains. The learned low-dimensional representations enable us to reconstruct the target-domain joint distribution from unlabeled target-domain data, and further enable predicting the labels in the target domain. We demonstrate the efficacy of this method by conducting experiments on synthetic and real datasets.

1 INTRODUCTION

In recent years machine learning techniques have become ubiquitous in solving real-world problems. For many of these applications obtaining new labeled data can be difficult, time-consuming, or expensive. Moreover, the training and test data are collected during different time periods and/or under different conditions, often yielding a shift in the distribution across datasets. For example, the distribution of medical data regarding a particular disease may vary from patient to patient because of heritable factors and different laboratory and measurement conditions. Furthermore, image datasets are collected in more than one setting, with different viewpoints and illumination conditions. Suppose we have one or more labeled datasets called source domains, and a new unlabeled dataset called target domain which has a different distribution from the source domain(s). Domain adaptation is the problem of accounting for the shift in distribution across domains such that relevant information is transferred from source domain(s) to the target domain so as to predict the target domain labels.

Let X denote the features and Y denote the labels. In the multiple-source domain adaptation setting, there are $M > 1$ source domains in the training data generated from multiple respective joint distributions $P_{XY}^{(1)}, ..., P_{XY}^{(M)}$. The goal is to learn a classifier for a new target domain with unlabeled data generated from $P_X^{\mathcal{T}}$. To enable successful domain transfer, one needs to make some assumptions about the joint distribution and take into account the generating process of the data. Following [1, 2, 3], we assume that the causal direction is $Y \to X$; then $P_{X|Y}$ corresponds to the causal mechanism that generates features from the label. According to the modularity property of a causal model, $Y \to X$ implies that P_Y and $P_{X|Y}$ change independently across domains [4, 5, 6]. The generating process is illustrated on Figure 1. For example, in image classification, the class label can be considered as the cause of images. If we change the label distribution, this would not change the causal mechanism $P_{X|Y}$ that generates images from labels. The change of $P_{X|Y}$ can be due to other factors such as illumination and viewpoint. Thus, the factorization of the joint distribution following the causal direction (given by$P_Y P_{X|Y}$) is more favorable, because the other factorization yields factors P_X and $P_{Y|X}$ which arise from independent modules P_Y and $P_{X|Y}$ (via Bayes rule), and are thus coupled and change dependently across domains in the generic case.

Determination of what information to transfer from source domains to the target is a crucial issue in domain adaptation. In this paper, we propose a nonparametric approach to capture distribution changes

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

and recover the target domain joint distribution. Since the causal direction is $Y \to X$, it is not surprising that the changes in the data generating process, $P_{X|Y}$, are usually simple and relatively easy to model. More specifically, we assume an infinite-dimensional non-parametric paradigm for the causal mechanism of all domains, i.e., $\{P_{X|Y;\Theta} : \Theta \in \Theta^{\infty}\}$, where Θ^{∞} is an infinite-dimensional space of parameters. We show that if the number of changing parameters in $P_{X|Y}$ is small, $P^{(1)}_{X|Y}, \cdots, P^{(M)}_{X|Y}$ lie in a low-dimensional manifold. Given enough source domains, we can identify the manifold of the d changing parameters by learning low-dimensional representations of the distributions $P^{(1)}_{X|Y}, \cdots, P^{(M)}_{X|Y}$. Furthermore, we can make use of the low-dimensional representation to reconstruct the target-domain causal mechanism $P^{\mathcal{T}}_{X|Y}$, which can then be used to construct the target-domain classifier.

Therefore, the motivation of our approach is two-fold: **(1)** Working with a plausible representation for the generating process of the data allows us to observe a low-dimensional change across domains. **(2)** When factorized according to the generative process, the factors of the distribution (i.e. P_Y and $P_{X|Y}$) change independently, and their respective low-dimensional changes across domains can be learned separately. The proposed method leverages these properties to extract the low-dimensional representations of the changing parameters across domains, and make use of it for predicting target-domain labels.

1.1 RELATED WORK

Classical single-source domain adaptation focuses on the setting where there are two domains, one labeled training dataset and one unlabeled test dataset (termed source and target domain), arising from two joint distributions $P^{\mathcal{S}}_{X,Y}$ and $P^{\mathcal{T}}_{X,Y}$, respectively. In order for classification in the test domain to be feasible, there must be some connection between the source and target domains, and this connection is reflected in the respective joint distributions. Therefore, domain adaptation approaches generally focus on understanding what aspects of the joint distribution change and leveraging this knowledge to account for the difference and construct an appropriate hypothesis in the target domain. Single-source domain adaptation has been extensively studied, with some of its theoretical underpinnings analyzed in [7, 8] and [9].

When considering the change of P_{XY} across domains, prior approaches generally make some assumptions of changes in its factors. Namely, a large body of work has focused on the setting in which it is assumed that P_X changes and $P_{Y|X}$ remains the same. Approaches in this setting focus on minimizing the discrepancy

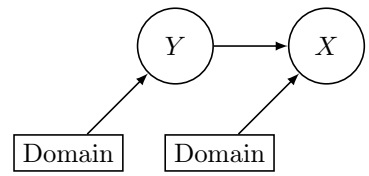

Figure 1: Generating process $Y \to X$ across domains with domain index variable $D = 1, ..., M$

in P_X between the weighted source domain and the target. This setting is called covariate shift or sample selection bias [10], and has been thoroughly studied in [11, 12, 13]. However, in practice both P_X and $P_{Y|X}$ can change across domains. An alternative assumption to address this is the setting in which P_Y changes and $P_{X|Y}$ remains the same, a setting termed target shift [1] or prior probability shift [14, 15].

A particular line of work in single-source domain adaptation , [1, 3] makes assumptions on the generative process across domains. With the assumption that the labels generate the features ($Y \to X$) (which is reasonable for many real-world scenarios including digit recognition and medical)diagnosis , the authors work with the factors $P_{X|Y}$ and P_Y, which are assumed to change independently under this causal generative process assumption [16]. This property is leveraged to model the changes in $P_{X|Y}$ and P_Y separately, naturally reducing the complexity of the problem.

Recently there have also been deep learning approaches to single-source domain adaptation. They are based on constructing domain adaptation layers with the aim of learning transferable representations for classification under the covariate shift setting [17, 18, 19, 20, 21]. Recent work by [22] focuses on the setting when multiple components of the joint distribution may change, and presents an architecture that extracts transferable representations that reduce the discrepancy of the joint distribution across domains.

There is a diverse body of work in multiple-source domain adaptation. Similar to single domain adaptation, [23] learns domain invariant components that are shared by all domains and uses them for prediction in the target domain. Other approaches focus on combining multiple hypotheses from the source domains and weighing them based on the source-domain marginal distributions, $P^{(1)}_X, ..., P^{(M)}_X$, [24], where the weights are determined in various ways [25, 26, 27]. Another approach [28] focuses on incorporating the marginal distribution P_X as an additional input of the classifier. However, P_X is an infinite-dimensional object, and performing direct comparisons on it across domains may lead to high estimation error and overfitting.

This fact is addressed by a method which assumes

that the generating process is $Y \to X$ and that the change across domains follows the Conditional-Target Shift setting (described above) [2]. This approach performs domain adaptation by assuming that the target conditional distribution $P^{\mathcal{T}}_{X|Y}$ is a linear mixing of the conditional distributions in the source domains $P^{(1)}_{X|Y}, ..., P^{(M)}_{X|Y}$, and solving for the mixing weights. However, the linear mixture assumption imposes a rather strong constraint on the type of low-dimensional changes that can be modeled and accounted for across domains.

In our approach, we follow the same domain adaptation setting, and we aim to automatically discover the (potentially nonlinear) low-dimensional changes across domains from data. This work consists of the following main contributions:
(1) We present a data-driven approach to capturing the low-dimensional manifold of the changes in the distribution across domains.
(2) We show that if the source- and target-domain joint distributions lie on a low-dimensional manifold, then the joint distribution in the target domain $P^{\mathcal{T}}_{XY}$ can be identified from the marginal distribution $P^{\mathcal{T}}_{X}$.
(3) We provide an algorithm that makes use of the low-dimensional manifold in order to reconstruct the joint distribution in the target domain and perform classification.

2 THEORETICAL FOUNDATION

Closely following the multiple-source settings of binary classification of [28] and [23], we let $\mathcal{X}$ be the input feature space, and let $\mathcal{Y} = \{-1, 1\}$ be the output space. Let $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$ be the family of joint distributions over $\mathcal{X} \times \mathcal{Y}$. Also, let $\mathcal{P}_{\mathcal{Y}}$ and $\mathcal{P}_{\mathcal{X}|\mathcal{Y}}$ be the respective families of distributions. Let there be a distribution μ on $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$, where $P^{(1)}_{XY}, ..., P^{(M)}_{XY}$ are independent and identically distributed (i.i.d.) realizations from this family for the source domains, and $P^{(\mathcal{T})}_{XY}$ is the realization for the target domain. In what follows, we shall describe some of the mathematical tools required to represent and make use of the changing parameters across source domains.

2.1 REPRESENTING DISTRIBUTIONS IN HILBERT SPACE

To perform domain adaptation, one needs to compare probability distributions. Kernel mean embeddings provide a convenient way to represent probability distributions as points in a Reproducing Kernel Hilbert Space (RKHS) associated with some positive semi-definite kernel, where the distance between them can be easily computed [29].

random variable	X	Y
domain	$\mathcal{X}$	$\mathcal{Y}$
feature map	$\psi(x)$	$\rho(y)$
kernel	$k(x, x')$	$l(y, y')$
i-th domain data point	$\mathbf{x}^{(i)}$	$\mathbf{y}^{(i)}$
empirical estimates of $P_X(x)$ and $P_Y(y)$	$\hat{P}_X(x)$	$\hat{P}_Y(y)$
kernel mean embedding on i-th domain	$\mu^{(i)}_X$	$\mu^{(i)}_Y$
feature map on kernel mean embedding	$\Phi(\mu_X)$	

Table 1: Notation used

Given a positive semi-definite kernel function k with corresponding RKHS $\mathcal{H}_k$ and a feature map $\psi : \mathcal{X} \to \mathcal{H}_k$ (s.t. for $x_1, x_2 \in \mathcal{X}$, $k(x_1, x_2) = \langle \psi(x_1), \psi(x_2) \rangle_{\mathcal{H}_k}$), the kernel mean embedding of the marginal distribution P_X is given by:

$$\mu_X := \int_{\mathcal{X}} k(x, \cdot) dP_X(x) = \mathbb{E}_{P_X}[\psi(x)]. \qquad (1)$$

When k is a characteristic kernel (such as the Gaussian kernel), μ_X is a point in $\mathcal{H}_k$ that captures all the moments of P_X. A computationally convenient distance metric between two distributions $P^{(1)}_X$ and $P^{(2)}_X$ is their Euclidean distance in the high-dimensional embedding space, given by $d(P^{(1)}_X, P^{(2)}_X) \equiv ||\mu^{(1)}_X - \mu^{(2)}_X||^2$. It is also known as the Maximum Mean Discrepancy (MMD) [30]. A consistent estimator of the kernel mean embedding with finite n data points is $\hat{\mu}_X = \frac{1}{n} \sum_{i=1}^{n} \psi(x_i)$.

While the marginal distribution is fully represented as a single point in Hilbert space, the conditional distribution P_{XY} is represented by a set of a family of points in RKHS indexed by the conditioning variable Y [31]. Namely, given a kernel l corresponding to a feature map $\rho : \mathcal{Y} \to \mathcal{H}_l$ and kernel k corresponding to feature map $\psi : \mathcal{X} \to \mathcal{H}_k$, the conditional kernel mean embedding is given by the operator $\mathcal{U}_{X|Y}$, a mapping from $\mathcal{H}_l$ to $\mathcal{H}_k$. Using this operator, the kernel sum rule [31] can be used to express the embedding of the marginal distribution P_X in terms of the independently changing factors $P_{X|Y}$ and P_Y: $\mu_X = \mathcal{U}_{X|Y}\mu_Y$. For a fixed value of the conditioning variable $Y = c$, the kernel mean embedding of $P_{X|Y=c}$ is given by:

$$\mu_{X|Y=c} := \int_{\mathcal{X}} k(x, \cdot) dP_{X|Y=c}(x) = \mathbb{E}_{P_{X|Y=c}}[\psi(x)].$$

It can be shown that when Y is discrete and $l(y_1, y_2) = \delta(y_1, y_2)$ is the Kronecker delta kernel, $\mathcal{U}_{X|Y} = [\mu_{X|Y=1}, ..., \mu_{X|Y=C}]^T$. Furthermore, similarly to the marginal case, the conditional kernel mean embedding for a fixed $Y = c$ can be estimated by $\hat{\mu}_{X|Y=c} =$

$\frac{1}{n_c}\sum_{i=1}^{n_c}\psi(x_i)$, where n_c is the number of observations which have class c.

2.2 IDENTIFYING LOW-DIMENSIONAL CHANGING PARAMETERS

The goal of our method is to mathematically express and utilize the identifiable changing parameters in P_{XY} across domains via its independent factors, in our case $P_{X|Y}$ and P_Y. We work with $P_{X|Y}$ and demonstrate how this can be achieved. When performing kernel mean embedding of the conditional distributions of the features given a class label c in the the source domains, $P^{(1)}_{X|Y=c}, ..., P^{(M)}_{X|Y=c}$, we obtain M points in $\mathcal{H}_k$ given by $\mu^{(1)}_{X|Y=c}, ..., \mu^{(M)}_{X|Y=c}$. Let there be a kernel k_μ with an RKHS $\mathcal{H}_{k_\mu}$ and a corresponding feature map $\Phi : \mathcal{H}_k \to \mathcal{H}_{k_\mu}$. To extract the nonstationary components (parameters) of these distributions, one needs to find the transformations of the distributions with maximal variability. This can be achieved by performing Kernel Principcal Component Analysis (KPCA) [32] on $\mu^{(1)}_{X|Y=c}...,\mu^{(M)}_{X|Y=c}$, using an additional kernel k_μ, resulting in a centered kernel Gram Matrix: $\tilde{\mathbf{K}}_{ij} = k_\mu(\mu^{(i)}_{X|Y=c}, \mu^{(j)}_{X|Y=c})$. To show that this is the case, we first need the following lemma regarding linear PCA.

Lemma 2 *Let points $\phi_1, ..., \phi_M$ be p-dimensional vectors (where p could be infinite). Let $\lambda_1, ..., \lambda_q$ be the set of all non-zero eigenvalues after performing PCA on these vectors. If $P_\lambda(\phi_i)$ is the projection of ϕ_i on the principal eigenvectors corresponding to $\lambda_1, ..., \lambda_q$, then $\phi_i \neq \phi_j \iff P_\lambda(\phi_i) \neq P_\lambda(\phi_j)$.*

The proof can be found in the supplementary materials. Using this lemma we are now ready to formally establish the connection between the changing parameters of distributions and the outcome of KPCA performed on their kernel mean embeddings:

Theorem 1 *Let $P^{(1)}_{X|Y=c}, ..., P^{(M)}_{X|Y=c}$ be probability distributions with d identifiable changing parameters $\Theta_d = \theta_1, .., \theta_d$, and $\xi_1, ..., \xi_q$ be principal components resulting from KPCA with kernel k_μ on kernel mean embeddings $\mu^{(1)}_{X|Y=c}...\mu^{(M)}_{X|Y=c}$. If k and k_μ are characteristic kernels, then $\xi_1, ..., \xi_q$ are a one-to-one mapping of the d changing parameters (i.e. $\xi_1, .., \xi_q = f(\theta_1, ..., \theta_d)$, where f is a bijective mapping.)*

Proof: The characteristic property of kernel k and identifiability of parameters $\theta_1, ..., \theta_d$ imply that $\Theta^{(1)}_d \neq \Theta^{(2)}_d \implies \mu^{(1)}_{X|Y=c} \neq \mu^{(2)}_{X|y=c}$ for the KMEs of two distributions $P^{(1)}_{X|Y=c}$ and $P^{(2)}_{X|Y=c}$, where $\Theta^{(1)}_d$ and $\Theta^{(2)}_d$ are their respective realizations of the d parameters. Performing KPCA on $\mu^{(1)}_{X|y=c}...\mu^{(M)}_{X|y=c}$ using kernel k_μ results in non-zero eigenvalues $\lambda_1, ..., \lambda_q$. An important observation is that for a particular $\mu^{(i)}_{X|Y=c}$, its principal components corresponding to the non-zero eigenvalues, given by $\xi^{(i)}_{1,c}, ..., \xi^{(i)}_{q,c}$ are a function of $\mu^{(i)}_{X|Y=c}$ (i.e. $\xi^{(i)}_{1,c}, ..., \xi^{(i)}_{q,c} = g(\mu^{(i)}_{X|Y=c})$). Therefore, it suffices to show that g is a one-to-one function. By the characteristic property of k_μ, it follows that $\Theta^{(1)}_d \neq \Theta^{(2)}_d \implies \Phi(\mu^{(1)}_{X|Y=c}) \neq \Phi(\mu^{(2)}_{X|Y=c})$, and $\Phi(\mu^{(1)}_{X|Y=c}), ..., \Phi(\mu^{(M)}_{X|Y=c})$ are points in a q-dimensional subspace in $\mathcal{H}_{k_\mu}$. Since KPCA performs linear PCA on infinite-dimensional points $\Phi(\mu^{(1)}_{X|Y=c}), ..., \Phi(\mu^{(M)}_{X|Y=c})$, by Lemma 2, $\Phi(\mu^{(1)}_{X|Y=c}) \neq \Phi(\mu^{(2)}_{X|Y=c}) \iff \xi^{(1)}_{1,c}, ..., \xi^{(1)}_{q,c} \neq \xi^{(2)}_{1,c}, ..., \xi^{(2)}_{q,c}$, so g is a one-to-one function. This means that $\Theta^{(1)}_d \neq \Theta^{(2)}_d \implies \xi^{(1)}_{1,c}, ..., \xi^{(1)}_{q,c} \neq \xi^{(2)}_{1,c}, ..., \xi^{(2)}_{q,c}$, implying that f is a composition of one-to-one functions, and is itself a one-to-one function.□

By establishing this one-to-one correspondence between the d changing parameters and the q principal components of KPCA, we have shown that the resulting q-dimensional manifold contains valuable low-dimensional information regarding the change of a particular factor of the joint distribution (in the proof treated as $P_{X|Y=c}$) across source domains $i = 1, ..., M$.

3 ALGORITHM

Now that we have a way of representing the changes of distributions across domains, we can use them to reconstruct the factors of the joint distribution in the target domain that will be used for classification. Given class labels $c = 1, ..., C$, the first step of the algorithm is to reconstruct the marginal distribution $P^{\mathcal{T}}_X$ by using the q-dimensional manifold of change across domains, such that the relevant factors are identified. The second step uses the reconstructed components $P^{\mathcal{T}}_Y$ and $P^{\mathcal{T}}_{X|Y}$ from the reconstructed marginal distribution in order to calculate $P^{\mathcal{T}}_{Y|X}$ and thus do classification in the target domain.

3.1 RECONSTRUCTION IN THE TARGET DOMAIN

The main objective of our method is to identify the two factors of the joint distribution: $P^{\mathcal{T}}_{X|Y=c}$ and $P^{\mathcal{T}}_{Y=c}$, $\forall c$. All of the information about these two factors is contained in the marginal distribution of the target-domain $P^{\mathcal{T}}_X = \sum_{c=1}^{C} P^{\mathcal{T}}(X|y=c)P^{\mathcal{T}}(Y=c)$. Since we have access to unlabeled data points $\mathbf{x}^{\mathcal{T}}_1, ..., \mathbf{x}^{\mathcal{T}}_{n_t}$ in the target domain, we can estimate the marginal distribution $\hat{P}^{\mathcal{T}}_X$, and search for factors $\hat{P}^{new}(X|y=c)$ and $\hat{P}^{new}(Y=c)$ that best reconstruct the marginal distribution estimate

in terms of: $\hat{P}_X^{new} = \sum_{c=1}^{C} \hat{P}^{new}(X|y=c)\hat{P}^{new}(Y=c)$. Thus, we aim to find the respective factors which minimize a distance metric $d(\hat{P}_X^{\mathcal{T}}, \hat{P}_X^{new})$

A computationally and statistically efficient procedure for minimization of the distance between the reconstructed and true marginal distribution is via Maximum Mean Discrepancy (MMD): $||\mu_X^{\mathcal{T}} - \mu_X^{new}||^2$, where $\mu_X^{\mathcal{T}}$ and μ_X^{new} are the kernel mean embeddings of $P_X^{\mathcal{T}}$ and P_X^{new} respectively [30].

We parameterize conditional distribution mean embedding in the target domain as $\mu_{X|Y=c}^{new} = \mathbb{E}_{X\sim P_X^{\mathcal{T}}}[\beta_c(x)\psi(x)]$, where $\beta_c(x)$ represents the class-specific density ratio $P_{X|Y=c}^{\mathcal{T}}/P_X^{\mathcal{T}}$ which needs to be learned. Here, ψ is the feature transform into Hilbert space corresponding to the Gaussian kernel or another characteristic kernel, while for the label Y we have a feature map $\rho(y)$ corresponding to the Kronecker Delta kernel $k(x,y) = \delta(x,y)$, so the kernel mean embedding for the label is $\mu_Y = \mathbb{E}_{Y\sim P_Y^{\mathcal{T}}}[\rho(y)]$. For possible labels $y = 1,..,C$, the feature map of this kernel is the standard basis $\rho(y) = e_Y$ and the corresponding kernel mean embedding is: $\mu_Y = \mathbb{E}_{Y\sim P_Y^{\mathcal{T}}}[\rho(y)] = [P_{Y=1}, ..., P_{Y=C}]$.

In addition to this parameterization of the target domain, we are also given a q-dimensional manifold in $\mathcal{H}_k$ of the changing parameters of $P_{X|Y=c}$ across domains. We minimize the maximum mean discrepancy (MMD) [30] between the marginal distribution of the target domain and its reconstruction $\mu_{X|Y=c}^{new}$, such that the reconstruction is as close as possible to the q-dimensional manifold. For this purpose, we introduce the following minimization criterion, given in population version:

$$\min_{\beta,\mu_Y} ||\mu_X^{\mathcal{T}} - \mathcal{U}_{X|Y}^{new}\mu_Y^{new}||^2$$

$$\iff \min_{\beta,\mu_Y} ||\mu_X^{\mathcal{T}} - \sum_{c=1}^{C} \mu_{X|Y=c}^{new}(\mu_Y)_c||^2 \tag{2}$$

$$\iff \min_{\beta,\mu_Y} ||\mu_X^{\mathcal{T}} - \sum_{c=1}^{C} \mathbb{E}_{X\sim P_X^{\mathcal{T}}}[\beta_c(x)\psi(x)](\mu_Y)_c||^2 \tag{3}$$

$$\text{s.t. } \sum_{c=1}^{C} ||\Phi(\mu_{X|Y=c}^{new}) - P_q\Phi(\mu_{X|Y=c}^{new})||^2 \leq \epsilon \tag{4}$$

$$\beta_c(x) \geq 0, \mathbb{E}_{X\sim P_X^{\mathcal{T}}}[\beta_c(x)] = 1 \ \forall c \tag{5}$$

In the first constraint, (4), Φ represents an additional feature map corresponding to the Gaussian kernel k_μ (which we also use to perform Kernel PCA), and we use $P_q\Phi(\hat{\mu}_{X|Y=c}^{new})$ to represent the reconstruction of $\mu_{X|Y=c}^{new}$ onto the q-dimensional manifold described by the principal components of the source domains $(\mu_{X|Y=c}^1, ..., \mu_{X|Y=c}^M)$ in the Gaussian Kernel feature space. Namely, if $\mathbf{v}_1, ..., \mathbf{v}_q$ are the eigenvectors corresponding to the nonzero eigenvalues in that feature space, then we let $\xi_{k,c}^{new} = (\mathbf{v}_k \cdot \Phi(\hat{\mu}_{X|Y=c}^{new})) = \sum_{i=1}^{M} \alpha_{i,c}^k k_\mu(\hat{\mu}_{X|Y=c}^{new}, \hat{\mu}_{X|Y=c}^{(i)})$ be the projection of $\hat{\mu}_{X|Y=c}^{new}$ on the k-th principal component, where $\boldsymbol{\alpha}_C$ vectors are eigenvectors of the centered Gaussian Kernel Gram Matrix $\tilde{\mathbf{K}}$ which was used to perform Kernel PCA on the source domains [33]. Then $P_q\Phi(\mu_{X|Y=c}^{new}) = \sum_{k=1}^{n} \xi_k \mathbf{v}_k$, and the k-th eigenvector $\mathbf{v}_k$ can be expressed as the following linear combination: $\mathbf{v}_k = \sum_{l=1}^{M} \alpha_{i,c}^k \Phi(\mu_{X|Y=c}^i)$. One should note that for each class label c we try to identify a separate low-dimensional manifold corresponding to the conditional distributions $P_{X|Y=c}^{(i)}$ of the source domains, and the regularizer penalizes the sum of reconstruction errors across all label-specific manifolds.

The last two constraints, given in (5), ensure that $P_{X|Y=c}^{\mathcal{T}} = \beta_c(x)P_X^{\mathcal{T}}$ is a valid distribution. The empirical version of the objective is:

$$\min_{\mathbf{B},\boldsymbol{\gamma}} ||\hat{\mu}_X^{\mathcal{T}} - \hat{\mathcal{U}}_{X|Y}^{new}\hat{\mu}_Y^{new}||^2 \tag{6}$$

$$\iff \min_{\mathbf{B},\boldsymbol{\gamma}} ||\hat{\mu}_X^{\mathcal{T}} - \sum_{c=1}^{C} \boldsymbol{\gamma}_j \frac{1}{n_{\mathcal{T}}} \sum_{i=1}^{n_{\mathcal{T}}} \mathbf{B}_{ic}\psi(x_i^{\mathcal{T}})||^2 \tag{7}$$

$$\text{s.t. } \sum_{c=1}^{C} ||\Phi(\hat{\mu}_{X|Y=c}^{new}) - P_q\Phi(\hat{\mu}_{X|Y=c}^{new})||^2 \leq \epsilon \tag{8}$$

$$\mathbf{B}_{ic} \in [0, B_{max}] \text{ and } |\sum_{i=1}^{n_{\mathcal{T}}} \mathbf{B}_{ic}| = n_{\mathcal{T}}, \tag{9}$$

$$\forall c \in 1, 2, \ldots, C. \tag{10}$$

Here, $\mathbf{B} \in \mathbb{R}^{n_{\mathcal{T}}\times C}$ contain the re-weighting coefficients that help reconstruct (estimate) the target conditional distribution given a specific class c: $\hat{P}_{X|Y=c}^{\mathcal{T}} = \mathbf{B}_{:,c}\hat{P}_X^{\mathcal{T}}$.

$\boldsymbol{\gamma}$ is used to estimate class probabilities (given by $\hat{P}_Y^{\mathcal{T}}$, as a result of applying the Kronecker Delta Kernel feature map) across all source domains, resulting in a new estimated marginal class probability in the target domain: $\hat{P}_{Y=c}^{new} = \boldsymbol{\gamma}_c$.

In order to make sure that $\mathbf{B}$ is a smooth function of the data, we reparameterize it as in [1]; namely, we let $\mathbf{B} = \mathbf{RA}$ where $\mathbf{R} = \mathbf{K}_B(\mathbf{K}_B + \lambda_B\mathbf{I})^{-1}$, where $\mathbf{K}_B$ is calculated using the Gaussian kernel with a separate width parameter σ_B, and regularized with a separate λ_B. We then minimize over $\mathbf{A} \in \mathbb{R}^{n_{\mathcal{T}}\times\kappa}$ instead. After incorporating this reparameterization and putting the objective in Lagrange form, we have:

$$\min_{\mathbf{A},\boldsymbol{\gamma}} ||\hat{\mu}_X^{\mathcal{T}} - \hat{\mathcal{U}}_{X|Y}^{new}\hat{\mu}_Y^{new}||^2 +$$

$$\lambda_f(\sum_{c=1}^{C} ||\Phi(\hat{\mu}_{X|Y=c}^{new}) - P_q\Phi(\hat{\mu}_{X|Y=c}^{new})||^2)$$

$$\iff \min_{\mathbf{A},\boldsymbol{\gamma}} ||\hat{\mu}_X^{\mathcal{T}} - \sum_{c=1}^{C} \boldsymbol{\gamma}_c \frac{1}{n_{\mathcal{T}}} \sum_{i=1}^{n_{\mathcal{T}}} (\mathbf{RA})_{ic}\psi(x_i^{\mathcal{T}})||^2 + \tag{11}$$

$$\lambda_f(\sum_{c=1}^{C} ||\Phi(\hat{\mu}_{X|Y=c}^{new}) - P_q\Phi(\hat{\mu}_{X|Y=c}^{new})||^2) \tag{12}$$

$$\text{s.t. } (\mathbf{RA})_{ic} \in [0, B_{max}] \text{ and } |\sum_{i=1}^{n_{\mathcal{T}}} (\mathbf{RA})_{ic}| = n_{\mathcal{T}}.$$

We use alternating optimization for this task; optimizing w.r.t $\boldsymbol{\gamma}$ is a straight-forward quadratic programming problem and optimizing w.r.t $\mathbf{A}$ can be done using a barrier method. (For details see the supplementary materials). The procedure is outlined in Algorithm 1.

3.2 GENERATIVE CLASSIFIER

Once we have the key components of the joint distribution in the target domain, we can perform classification in the target domain. The probability of the label given the data is

$$\hat{P}_Y^{\mathcal{T}}(\mathbf{y}_i = c|\mathbf{x}^{\mathcal{T}}) = \frac{P_Y^{\hat{n}ew}(\mathbf{y}_i = c)\hat{P}^{new}(\mathbf{x}^{\mathcal{T}}|\mathbf{y}_i = c)}{P_X^{\mathcal{T}}(\mathbf{x}^{\mathcal{T}})}$$
$$= P_Y^{\hat{n}ew}(\mathbf{Y_i} = c)\mathbf{B}_{ic} \tag{13}$$

We test the efficacy of this approach in the following section.

Algorithm 1 Classification Routine for Data-Driven Multi-Source Domain Adaptation

Input: **(1)** M source domains with n_i labeled training data-points: $(x_1, y_1), ..., (x_{n_i}, y_{n_i}) \sim P_{XY}^{(i)}$ $\forall i \in 1, ..., M$.
(2) A target domain with unlabeled data-points: $x_1, ..., x_{n_{\mathcal{T}}} \sim P_X^{\mathcal{T}}$
Output: predicted class labels in target domain: $\hat{\mathbf{y}}$

```
while not converged do
    Solve MMD problem given by (11) for γ using quadratic programming.
    Solve MMD problem given by (11) for A using barrier method.
end while
B = RA,
Return P̂_Y^T(y_i = c|x_i) = γ_c B_ic, ∀i ∈ 1,...,n_T, ∀c ∈ 1,...,C.
```

3.3 IDENTIFIABILITY OF TARGET CONDITIONAL DISTRIBUTION

The above algorithm identifies the separate components $P_{X|Y}$ and P_Y while reconstructing the marginal distribution P_X. Before presenting the identifiability result, we make some assumptions:

$\mathbf{A}_1$: For each value of c, the distribution $P_{X|Y=c}$ has only a finite number of parameters that change across possible domains. Suppose we have enough source domains, and let q be the number of non-zero eigenvalues of Gram matrix on $\mu_{X|Y=c}$ across all source domains.

Assumption $\mathbf{A}_1$ implies that there exists a nonlinear one-to-one transformation $h : \mathcal{P}_{\mathcal{X}|\mathcal{Y}} \to \mathbb{R}^q$. Then, the conditional distribution in each domain j is a linear combination of the other domains after such a transformation: $h(P_{X|Y=c}^{(j)}) = \sum_{i=1, i\neq j}^{M} \eta_{ic}^j h(P_{X|Y=c}^{(i)})$ for some weights $\eta_{1c}^j, .., \eta_{Mc}^j$. Furthermore, for the target domain $\mathcal{T}$, $\exists\ \boldsymbol{\eta}_c^*$ s.t. $h(P_{X|Y=c}^{\mathcal{T}}) = \sum_{i=1, i\neq j}^{M} \eta_{ic}^* h(P_{X|Y=c}^{(i)})$. In other words, all domain-specific conditional distributions for label c lie in a q-dimensional subspace of $\mathcal{H}_\mu$. This means that each conditional distribution corresponding to domain j can be uniquely determined by the mixture weights $\eta_{1c}^j, .., \eta_{Mc}^j$.

$\mathbf{A}_2$: Let $P_{X|Y=c}^{\boldsymbol{\eta}_c}$ be a distribution determined by weights $\boldsymbol{\eta}_c$, and $P_{X|Y=c}^{\boldsymbol{\eta}_c'}$ be determined by $\boldsymbol{\eta}_c'$. Then the elements of the set $\{p_{1c}P_{X|Y=c}^{\boldsymbol{\eta}_c} + p_{2c}P_{X|Y=c}^{\boldsymbol{\eta}_c'}; c = 1, .., C\}$ are linearly independent for $\forall \boldsymbol{\eta}_c, \boldsymbol{\eta}_c', p_{1c}, p_{2c}, p_{1c}^2 + p_{2c}^2 \neq 0$.

We can now state the following identifiability theorem:

Theorem 2 *Let $\mathbf{A}_1$ and $\mathbf{A}_2$ hold, and $\hat{\eta}_c$ be the weights such that $P_{X|Y=c}^{new} = P_{X|Y=c}^{\hat{\boldsymbol{\eta}}_c}$ is the reconstructed distribution, namely $P_{X|Y=c}^{new} = \mathbf{B}_{:,c}P_X^{\mathcal{T}}$. If $\exists\ \hat{\boldsymbol{\eta}}_c$ s.t $P_X^{\mathcal{T}} = \sum_{c=1}^{C} P_Y^{new}(Y = c)P_{X|Y=c}^{\hat{\boldsymbol{\eta}}_c} = \sum_{c=1}^{C} \boldsymbol{\gamma}_c P_{X|Y=c}^{\hat{\boldsymbol{\eta}}_c}$, then we have $\forall$ c, $P_Y^{\mathcal{T}}(Y = c) = \boldsymbol{\gamma}_c$ and $P_{X|Y=c}^{\hat{\boldsymbol{\eta}}_c} = P_{X|Y=c}^{\mathcal{T}}$.*

4 EMPIRICAL RESULTS

4.1 BASELINES

We consider several baselines that can be used to perform classification in the target domain using data from multiple source domains:
(1) The simplest and most straightforward approach is to combine the data from all source domains and treat it as if it arose from a single joint distribution P_{XY} and use it for training via SVM. This approach is called "poolSVM".

(2) The method introduced by [24], in which the target-domain conditional distribution $P_{Y|X}^{\mathcal{T}}$ is represented as a linear mixture of the source-domain marginal distributions, $P_{Y|X}^{\mathcal{T}} = \sum_{i=1}^{M} \lambda_i P_{Y|X}^{(i)}$, where the weights are functions of the marginal distributions of the source domain, namely $\lambda_i = \frac{\tilde{\alpha}_i P_X^{(i)}}{\sum_{q=1}^{M} \tilde{\alpha}_q P_X^{(q)}}$. When they introduced the method, [24] used uniform weights $\tilde{\alpha}_i = \frac{1}{M}$ $\forall i \in 1, ..., M$. As described in [2], the weights can also be learned using a kernel mean matching approach, such that $\sum_{i=1}^{M} \tilde{\alpha}_i P_X^{(i)}$ is as close to $P_X^{\mathcal{T}}$ as possible (we refer to this approach as "dist-weight").

(3) Treating the target conditional distribution as a uniform mixture of the source-domain conditional distributions, $P_{X|Y}^{\mathcal{T}} = \frac{1}{M}\sum_{i=1}^{M} P_{X|Y}^{(i)}$. We refer to this baseline as "uniform". This method is proven to be optimal when $X \to Y$ and P_X stays the same (shown in [2, Proposition 1]).

(4) The algorithm proposed by [2] which, like our approach, assumes the generative process $Y \to X$, and aims to use the relevant low-dimensional factors P_Y and $P_{X|Y}$, where the kernel mean embedding of $P_{X|Y}$ in the target domain is a linear mixture of the kernel mean embeddings in the source domains, namely: $\mu^{\mathcal{T}}_{X|Y=c} = \sum_{i=1}^{M} \lambda_i \mu^{(i)}_{X|Y=c}$. The mixing weights are learned jointly with $P^{\mathcal{T}}_Y$, and this information is used to do distribution-weighted combination of the classifiers in the source domains, like in the method by [24]. We denote this method by "dist-comb".

(5) The method proposed by [28], which uses a kernel SVM approach. The authors used the canonical SVM framework with a product kernel which, in addition to comparing data points, also compares marginal distributions across domains. This kernel is given by a product of two kernel functions: $k_B((P^{(i)}_X X_{iq}), (P^{(j)}_X, X_{jl})) = k_P(P^{(i)}_X, P^{(j)}_X) k_X(X_{iq}, X_{jl})$ between two points X_{iq} and X_{jl} of domains i and j. Here, k_P is a characteristic kernel that operates on probability distributions, and k_X is a kernel applied directly on the data points. We refer to this method as "marg-kernel".

4.2 SYNTHETIC DATASETS

In order to test the effectiveness of our proposed method, we perform the task of handwritten digit recongition on the MNIST [34] dataset. This task satisfies the assumption of the generative process $Y \to X$, and is thus suitable for application of our approach. We performed two classification tasks; in the first one we classify digits 4 and 9, and in the second one we try to discern between digits 1 and 7. For each task, we create a multiple-source domain adaptation setting, where each domain represents a rotation of a digit with a different angle. We establish 20 such angles, with the difference of two adjacent domains (angles) being 18 degrees . Thus, in this setting, rotation is the only changing parameter across domains. Because of the choice of the changing parameter, this dataset violates the commonly required assumption that the target domain must be contained in the support of the source-domain joint distributions. We conduct 20 experiments, where each angle is treated as a target domain, and 10 other source angles are sampled randomly, while ensuring that the nearest source angle is at least 36 degrees away from the target. We sample 350 points for each source domain and the target domain. Because the dimensionality of images is high and we used a very simple approach to reduce it, we fixed $P(Y = c)$ to range between 0.2 and 0.8 for the two classes in order to prevent instability when estimating $P_{X|Y}/P_X$ in our generative approach via MMD.

We present the accuracies of all the baselines and the proposed method (termed "generative") in Figures 2 and 3, for the classification of digits "4" vs. "9" and "1" vs. "7" respectively. In addition, we also provide the average accuracy together with standard deviations and Wilcoxon signed rank tests in Table 2.

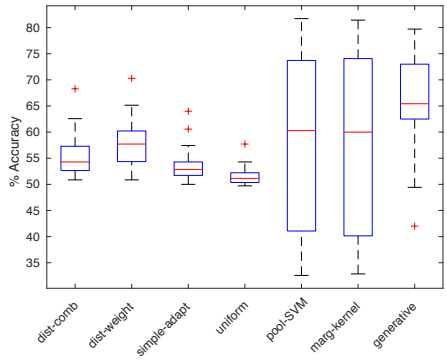

Figure 2: Accuracies of the baselines and the proposed method for the task of classifying between digits 4 and 9, for handwritten digit recognition

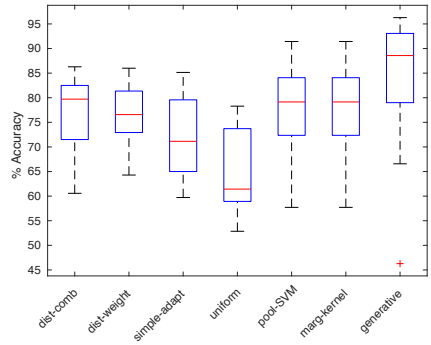

Figure 3: Accuracies of the baselines and the proposed method for the task of classifying between digits 1 and 7, for handwritten digit recognition

From the figures, one can see that the proposed method outperforms all the baselines. In particular, this performance gap is drastic in the task with digits 4 and 9, where the digits are difficult to classify and there is no possibility for support overlap between the source and target domains because of reflexion. It demonstrates that our method is capable of utilizing the one-dimensional rotational changes across domains to perform classification on datasets with a complex decision boundary.

4.3 REAL DATASET

We also applied our method to lung phenotype data (CT images) from the COPDGene cohort, which is a public dataset for lung disease study. The task here is to detect the fissure between two lung lobes, which is

	dist-comb	dist-weight	simple-adapt	uniform	poolSVM	marg-kernel	generative
MNIST 4/9							
% accuracy	55.6391(4.43)	58.0(4.8)	54.0 (3.4)	51.4 (1.8)	57.93 (15.9)	58.0 (17.0)	**65.8 (9)**
p-value	0.0006	0.0033	0.0003	0.0001	0.0795	0.0766	–
MNIST 1/7							
% accuracy	76.81 (7.4)	76.76 (7.7)	72.0 (8)	64.96 (8.8)	77.74 (8.8)	77.72(12.3)	**84.4 (2.7)**
p-value	0.01	0.009	0.001	0.0003	0.035	0.035	–
Medical							
% accuracy	75.07(14.4)	79.39 (15.0)	81.76 (14.2)	81.75 (13)	76.7 (13.9)	81.41 (14.7)	**85.62 (7.4)**
p-value	0.0002	0.015	0.07	0.05	0.0005	0.08	–

Table 2: Accuracies and p values for Wilcoxon signed rank test across the baselines and the proposed method performed on: the MNIST dataset where we classify 4 vs. 9 (top), he MNIST dataset where we classify 1 vs. 7 (top), and the real dataset from lung lobe images (bottom). The p-values displayed are comparing the proposed method with each respective baseline.

a binary classification problem. This task is an important intermediate step towards understanding which genes are responsible for certain lung diseases. The fissure is represented by a 3D point set obtained by the method proposed in [35] and further refined by manual annotations. The goal is to classify whether the 3D points belong to one fissure region or another (represented by the positive and negative labels). Since the lung and fissure shape varies from patient to patient, the distributions of the points for the two fissures change across different patients. Furthermore, since labeling the points is costly and expensive, it would be very useful to be able to learn an optimal classifier on lung image data for a target patient by using existing labeled data from a few other patients by applying our method.

We conducted 40 experiments, in which randomly picked 7 source patients, and for each experiment we randomly sampled a target patient. We then subsampled 250 points for each patient (domain), such that $P(Y = 1)$ varies uniformly between 0.2 and 0.8 across all patients (both sources and target) for each experiment. We then performed classification using the generative method in each of the 40 target patients, and we present the accuracies in Figure 4 and Table 2. From these real dataset experiments, we see that our method outperforms all of the baselines. We also note that all of the baselines have a much higher variance probably due to larger differences between the distributions of the target patient and source patients in some of the experiments.

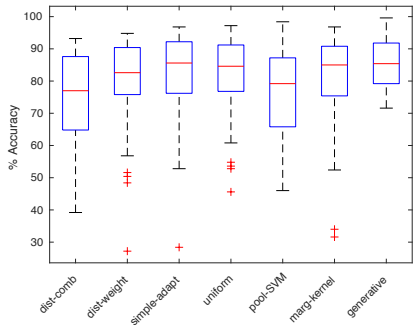

Figure 4: Accuracies of the baselines and the proposed method for the task of classifying between two different lung fissures in the real dataset

5 CONCLUSION

We developed a data-driven method to discover and utilize low-dimensional changes of the joint distribution across domains for the purpose of domain adaptation. We did so by representing and exploiting the low-dimensionality of the change of the causal mechanism $P_{X|Y}$ across source domains. Out approach consists of two steps: (1) reconstructing the marginal distribution in the target-domain $P_X^{\mathcal{T}}$ such that $P_{X|Y=c}^{\mathcal{T}}$ and $P_Y^{\mathcal{T}}$ can be identified, and (2) using the reconstructed joint distribution in the target domain to perform classification. We have proven that this method is theoretically well grounded and have demonstrated its increased efficacy compared to the baselines via synthetic and real data experiments. We believe that this method opens the door for more flexible and principled data-driven approaches to domain adaptation.

Acknowledgements We would like to acknowledge the supported by the United States Air Force under Contract No. FA8650-17-C-7715, by National Institutes of Health under Contract No. NIH-1R01EB022858-01, FAINR01EB022858, NIH-1R01LM012087, NIH-5U54HG008540-02, and FAIN-U54HG008540, and by National Science Foundation EAGER Grant No. IIS-1829681. The National Institutes of Health, the U.S. Air Force, and the National Science Foundation are not responsible for the views reported in this article. We would also like to thank Kayhan Batmanghelich for providing us with the lung phenotype data.

References

[1] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *ICML (3)*, pages 819–827, 2013.

[2] Kun Zhang, Mingming Gong, and Bernhard Schölkopf. Multi-source domain adaptation: A causal view. In *AAAI*, pages 3150–3157, 2015.

[3] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2839–2848, 2016.

[4] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2000.

[5] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2nd edition, 2001.

[6] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij. On causal and anticausal learning. In *Proc. 29th International Conference on Machine Learning (ICML 2012)*, Edinburgh, Scotland, 2012.

[7] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[8] Jing Jiang. A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, 3, 2008.

[9] 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.

[10] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114. ACM, 2004.

[11] 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.

[12] Shai Ben-David and Ruth Urner. Domain adaptation–can quantity compensate for quality? *Annals of Mathematics and Artificial Intelligence*, 70(3):185–202, 2014.

[13] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440, 2008.

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

[15] Marthinus Christoffel Du Plessis and Masashi Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119, 2014.

[16] James Woodward. *Making things happen: A theory of causal explanation*. Oxford university press, 2005.

[17] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015.

[18] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[19] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[20] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7, 2017.

[21] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[22] Mingsheng Long, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. *arXiv preprint arXiv:1605.06636*, 2016.

[23] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *Proceedings of the 30th*

International Conference on Machine Learning (ICML-13), pages 10–18, 2013.

[24] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pages 1041–1048, 2009.

[25] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 283–291. ACM, 2008.

[26] Lixin Duan, Ivor W Tsang, Dong Xu, and Tat-Seng Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 289–296. ACM, 2009.

[27] Rita Chattopadhyay, Qian Sun, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):18, 2012.

[28] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in neural information processing systems*, pages 2178–2186, 2011.

[29] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pages 13–31. Springer, 2007.

[30] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[31] Le Song, Kenji Fukumizu, and Arthur Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.

[32] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.

[33] Sebastian Mika, Bernhard Schölkopf, Alexander J Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. In *NIPS*, volume 11, pages 536–542, 1998.

[34] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[35] James C Ross, Gordon L Kindlmann, Yuka Okajima, Hiroto Hatabu, Alejandro A Díaz, Edwin K Silverman, George R Washko, Jennifer Dy, and Raúl San José Estépar. Pulmonary lobe segmentation based on ridge surface sampling and shape model fitting. *Medical physics*, 40(12), 2013.