DACH: Domain Adaptation Without Domain Information

Ruichu Cai, *Member, IEEE*, Jiahao Li, Zhenjie Zhang, Xiaoyan Yang, and Zhifeng Hao, *Member, IEEE*

***Abstract*— Domain adaptation is becoming increasingly important for learning systems in recent years, especially with the growing diversification of data domains in real-world applications, such as the genetic data from various sequencing platforms and video feeds from multiple surveillance cameras. Traditional domain adaptation approaches target to design transformations for each individual domain so that the twisted data from different domains follow an almost identical distribution. In many applications, however, the data from diversified domains are simply dumped to an archive even without clear domain labels. In this article, we discuss the possibility of learning domain adaptations even when the data does not contain domain labels. Our solution is based on our new model, named domain adaption using cross-domain homomorphism (DACH in short), to identify intrinsic homomorphism hidden in mixed data from all domains. DACH is generally compatible with existing deep learning frameworks, enabling the generation of nonlinear features from the original data domains. Our theoretical analysis not only shows the universality of the homomorphism, but also proves the convergence of DACH for significant homomorphism structures over the data domains is preserved. Empirical studies on real-world data sets validate the effectiveness of DACH on merging multiple data domains for joint machine learning tasks and the scalability of our algorithm to domain dimensionality.**

***Index Terms*— Causality, deep learning, domain adaptation, homomorphism operator.**

I. Introduction

DATA is the core of learning systems, and *data adaptation* is the key to effective machine learning over massive data from multiple domains. The diversity of the data domain is constantly growing, mainly due to quickly increasing capability and lowering the cost of data collection in the big data era. In the genomic area, for example, genomic sequencing now only costs hundreds of dollars such that a huge number of platforms are now providing services for human genome sequencing and analysis. By combining all genomic data from those platforms, it could fulfil the vision of a better understanding of genetic diseases and the design of new genetic therapies. In computer vision area, millions of surveillance cameras are generating billions of image frames for analysis in every hour, requiring generic processing techniques to support real-time response to suspicious articles and person [1]. Domain adaption is the key to the success of data integration for business in these areas, which aims to eliminate the biases linked to the domains and identify meaningful features for joint learning tasks over massive data.

The core idea behind existing studies on both unsupervised and semisupervised domain adaptation is to design a transformation for each individual domain such that these transformations on the original data generate an almost identical distribution. For example, [2] and [3] assume that different domains have the same prior in their data generation procedure. The bias with respect to the domain is reflected by the representation of the data by an unknown transformation. It is, therefore, sufficient to reconstruct the reversed transformation to align the data distributions across the domains. Similarly, other studies [4], [5] imply that distribution alignment remains effective if the priors over the domains are different, under the mild assumption on the (possibly different) linear feature generation beneath the observations over the domains.

In this article, we unveil the new possibility of domain adaptation even when the data do not contain any domain information, i.e., there is no label describing the domain of the records. Domain adaptation under this new setting is slightly different from existing studies, in the sense, that the goal is to extract common features from the tuples which are invariant across domains but retain distinguishing power for machine learning tasks, e.g., regression and classification. Such problem setting is common in real-world applications, on archived genetic data without platform information and the video data from a huge number of surveillance cameras without the position and angle information. Note that our setting (only need the task label) is much more general and has wider application scenarios than the existing setting (usually need both the task label and domain label). Moreover, while existing approaches of domain adaptation are applicable to

Manuscript received August 30, 2018; revised January 15, 2019, May 19, 2019, August 21, 2019, and December 5, 2019; accepted December 25, 2019. This work was supported in part by the National Natural Science Foundation of China (NSFC)-Guangdong Joint Fund under Grant U1501254, in part by the NSFC under Grant 61876043 and Grant 61572143, in part by the Natural Science Foundation of Guangdong under Grant 2014A030306004 and Grant 2014A030308008, in part by the Guangdong High-level Personnel of Special Support Program under Grant 2015TQ01X140, and in part by the Science and Technology Planning Project of Guangzhou under Grant 201902010058. *(Corresponding author: Ruichu Cai.)*

Ruichu Cai and Jiahao Li are with the School of Computer, Guangdong University of Technology, Guangzhou 510006, China (e-mail: cairuichu@gdut.edu.cn; jiahaoli.gdut@gmail.com).

Zhenjie Zhang and Xiaoyan Yang are with the Singapore Research and Development Centre, Yitu Technology Singapore, Singapore 018960 (e-mail: zhenjie.zhang@yitu-inc.com; xiaoyan.yang@yitu-inc.com).

Zhifeng Hao is with the School of Mathematics and Big Data, Foshan University, Foshan, China 528100, and also with the School of Computer, Guangdong University of Technology, Guangzhou 510006, China (e-mail: zfhao@gdut.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2962817

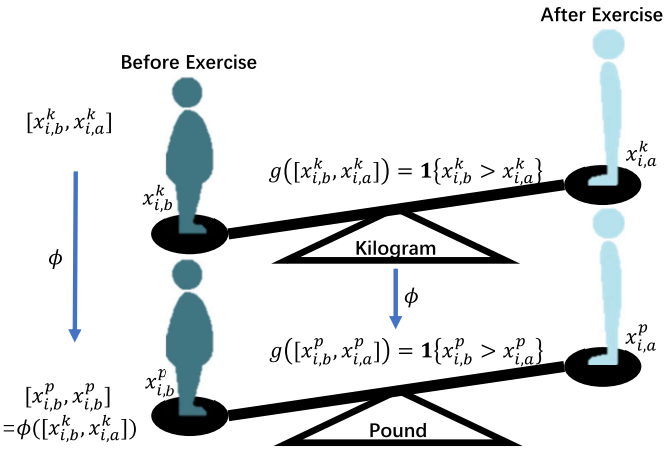

Fig. 1. Homomorphism between the kilogram and pound domain. In this example, $x^k_{i,b}$, $x^k_{i,a}$, $x^p_{i,b}$, and $x^p_{i,a}$ are the weights of the ith sample on different domains and exercise states, $g(x_{i,b}, x_{i,a}) = \mathbf{1}\{x_{i,b} > x_{i,a}\}$ is a homomorphism operator, results of which are consistent across the domains.

a few domains, usually two in the experiment setup, our problem usually involves dozens or hundreds of domains in one single-mixed data set.

While data without domain labels raise huge challenges to the reliability and extensibility on feature selection for regression and classification tasks, fortunately, there is usually strong consistency among indicating features from various domains, based on the observations in existing studies on particular areas, e.g., bioinformatics and medical science. Fig. 1 gives a toy example of such consistency. In this example, we want to study the effect of exercise on the weight, given the ith sample evaluated on the kilogram domain and the pound domain, respectively, and the results of the operator g are consistent across the domains. Formally, there is a homomorphism between the two domains, and ϕ is the corresponding homomorphic mapping. Homomorphism provides a way to preserve the structures, but not only aligning the distribution of different domains. Homomorphism phenomenon exists in many fields. For example, the relationship between genes is usually independent of the measurement platforms. Similarly, the image content is usually independent of its source camera.

Motivated by these observations, we believe the key of the solution is to explore the feature space and identify effective homomorphic features hidden the mixed data without a domain label. We, therefore, propose a domain adaptation approach using such cross-domain homomorphism (named DACH), with a suite of new and enticing properties as solutions to the problem. First, DACH is easily extensible to an arbitrary number of domains for adaptation. Second, DACH theoretically preserves the statistics behind the causal structures in the original domain, i.e., the connection between the variables behind the unified data generation procedure in our assumption. Third, DACH supports both unsupervised and semisupervised learning over the domains, in the sense that any labeled data from the domains can be used to enhance the adaptation model accuracy at any time. Fourth, DACH is generally compatible with existing deep learning framework. It can be implemented on top of any mainstream deep learning framework, to generate features under any specified deep neural network structures. All these features make DACH a generic and effective tool for general domain adaptation tasks for all purposes. Our empirical evaluations verify the advantages of DACH over the state-of-the-art solutions by a huge margin.

II. Related Work

Aimed at transferring knowledge from source domains to target ones, a large number of domain adaptation approaches have been proposed recently. Unsupervised and semisupervised approaches are two mainstream approaches.

For unsupervised approaches, there is no labeled data on the target domain, and the existing works focus on the alignment of the feature distributions between the source and target domains. Some approaches achieve the alignment by selecting or reweighting source samples [6]–[8], while the others try to find the alignment by feature space transformation [4], [9], [10]. An effective distribution measure is necessary for these alignment algorithms. One popular measure is evaluating their distribution discrepancy in the kernel-reproducing Hilbert space [11]–[13]. Some new measures are developed from the second-order or higher order statistics for estimating distribution disagreement [14], [15]. In practice, these measures could be applied in deep learning as regularizations, to ensure the neural network trained from source domains also works well on the target domain. For examples, Maximum Mean Discrepancy (MMD) and multikernel-based variant are embedded in two subnets related to different domains [14], [16], [17]. Similarly, total domain distinctiveness has been introduced in multisource setting [18]. Furthermore, the distribution measurement could be viewed as an output of subnet predicting the domain label, which brings a set of domain-specified alignment methods, e.g., the domain invariant distribution alignment [2], [3] and domain feature normalization-based methods [19], [20].

For semisupervised approaches, there exist a small number of labeled samples on the target domain, and the key is to build a bridge from sources to targets based on the labeled samples of the target domain. In feature-based series, the most classical trick is seeking a linear transformation from the labeled sample in sources and targets [21], [22], and the popular approach is mapping different domains to a domain-invariant feature space [23]–[25]. Among the mapping-based methods, semisupervised clustering is one of the most used techniques [26], [27]. In sample-based series, [28] confirms the correct tags of weakly annotated data using some labeled samples in the target domain before model training. Similarly, some AdaBoost-based approaches also borrow a small amount of target labeled data to select helpful source instances [29], [30]. In some additional scenarios, the model performs much better by involving the inherent properties hold in unlabeled target samples [31], [32]. In recent years, the combinations of semisupervised domain adaptation and deep learning are also very significant. Some novel studies state that semisupervised domain adaptation can be implemented with coupled neural networks and generative adversarial nets [10], [33]–[35].

TABLE I
NOTATIONS

Symbol	Description
Z	domain-related random variable
$\mathbb{Z}$, $\mathbb{Z}_S$, $\mathbb{Z}_T$	full, source and target domain set, respectively
m, m_S, m_T	number of full, source and target domains, respectively
z_j	jth domain realization of Z
$\boldsymbol{x}_i^j$	ith sample from source domain z_j
x_{iv}^j	$\boldsymbol{x}_i^j$'s value on the vth component
y_i^j	label of $\boldsymbol{x}_i^j$
$\mathcal{G}$	homomorphic feature extractor space
$\mathcal{F}$	classifier function space
g	homomorphic feature extractor from space $\mathcal{G}$
f	classifier function from space $\mathcal{F}$

Our work is also related to the multisource domain adaptation without domain information and the domain generalization. For the multisource setting without domain information, the recent studies either measure a distinctiveness on all domains [18], [36] or use the clustering and so on techniques to predict the latent domain labels [20], [27]. LatentDA [20] follows the line of domain label prediction-based method and achieves the best performance among the multisource domain adaptation. For the domain generalization setting, typical methods are designed with low-rank constraints [36], [37] or common hidden distribution process [11]. In the domain generalization setting, the target domain data are not used in the training phase, which is different from the problem setting of our method.

As a summary, most of the existing methods explore the domain information to align the distribution or feature space, which are usually failed to work when the data do not contain any domain information. Inspired by the causal mechanism analysis of the domain adaptations [38] and the homomorphism phenomenon in the real-world applications [39]–[41], our work focuses on exploring the homomorphism among the data to unveil the new possibility of domain adaptation even when the domain information is unavailable.

III. HOMOMORPHISM

Assume there exists a domain-related random variable Z controlling the generation of *homogeneous* data (e.g., human genome sequences) with *heterogeneous* representations (e.g., results from different sequencing platform). Let $\mathbb{Z}_S = \{z_j\}_{j=1}^{m_S}$ is the source domain set with m_S realizations of Z. Let $\mathbb{Z}_T = \{z_j\}_{j=m_S+1}^{m_S+m_T}$ is the target domain set with m_T realizations of Z. $\boldsymbol{X}$ is the unified observation data space and Y is the label space, such that each $(\boldsymbol{x}^j, y^j) \in \boldsymbol{X} \times Y$ is a tuple generated by the domain z_j. The goal of domain adaptation is to transform every tuple $(\boldsymbol{x}, y)$ to a new representation $(\widehat{\boldsymbol{x}}, y)$ and run joint machine learning over the complete data set from all domains under the new representations. All the related notations are summarized in Table I.

A. Homomorphism Operator

We now begin with the definition of homomorphism, then apply the concept to the domain adaptation task.

Definition 1 (Group Homomorphism [42]): Let $(A_1, *)$ and $(A_2, \cdot)$ be two groups. A mapping $\phi : A_1 \to A_2$ is called a group homomorphism if for all $x_1, x_2 \in A_1$, $\phi(x_1 * x_2) = \phi(x_1) \cdot \phi(x_2)$ is valid.

By extending the binary operators ("$*$" and "$\cdot$") to the n-ary operator and further assuming A_1 and A_2 share a common n-ary operator g, the homomorphism with n-ary operations is as follows.

Definition 2 (Homomorphism With n-Ary Operation [43]): Let (A_1, g) and (A_2, g) be two algebras and g be a n-ary operator defined over the n-dimension vector $\boldsymbol{x}$ on both A_1 and A_2. A mapping $\phi : A_1 \to A_2$ is called a homomorphism if $\phi(g(\boldsymbol{x})) = g(\phi(\boldsymbol{x}))$ is valid.

In the above-mentioned homomorphism with n-ary operation, the mapping ϕ and the operator g are generally called *homomorphism map* and *homomorphism operator*, respectively.

In the following, we will consider the interdomain homomorphism mapping over the feature space in the domain adaptation task. Let $\{\boldsymbol{x}_{\mathcal{V}}^{j_1}\}$ and $\{\boldsymbol{x}_{\mathcal{V}}^{j_2}\}$ be two feature sets from the unified observation data space $\boldsymbol{X}$, but from domains z_{j_1} and z_{j_2}, respectively. Assume there exists a *homomorphism map* $\phi : \boldsymbol{x}_{\mathcal{V}}^{j_1} \to \boldsymbol{x}_{\mathcal{V}}^{j_2}$ and the associated *homomorphism operator* g on the subspace $\mathcal{V}$, then

$$g\left(\boldsymbol{x}_{i_1\mathcal{V}}^{j_2}\right) = g\left(\phi\left(\boldsymbol{x}_{i_1\mathcal{V}}^{j_1}\right)\right) = \phi\left(g\left(\boldsymbol{x}_{i_1\mathcal{V}}^{j_1}\right)\right) \tag{1}$$

holds for every pair $\boldsymbol{x}_{i_1\mathcal{V}}^{j_1}$ and $\boldsymbol{x}_{i_1\mathcal{V}}^{j_2}$.

In the domain adaptation task, we focus on extracting the label-specified but domain-invariant subspace from the original feature space. Thus, it is reasonable to further assume the result feature space of the homomorphism operator is invariant to he mapping across the domains, i.e., $g(\boldsymbol{x}_{i_1\mathcal{V}}^{j_1})$ is invariant to the interdomain mapping ϕ. From the view of abstract algebra, the result space of g is corresponding to the kernel of the two algebras, which is almost universal in various algebras [43]. Such assumption is formally presented as follows:

$$\phi\left(g\left(\boldsymbol{x}_{i_1\mathcal{V}}^{j_1}\right)\right) = g\left(\boldsymbol{x}_{i_1\mathcal{V}}^{j_1}\right). \tag{2}$$

Combining (1) and (2), we finally have

$$g\left(\boldsymbol{x}_{i_1\mathcal{V}}^{j_1}\right) = g\left(\boldsymbol{x}_{i_1\mathcal{V}}^{j_2}\right). \tag{3}$$

The results of the homomorphism operator g are invariant structures among the domains. Such property is attractive in the domain adaptation tasks. Beside the invariant to the domain, we also want the extracted features are discriminant to the task labels. We can further derive the following label specified homomorphism. Data from different domains are label-specified homomorphic if there exists a homomorphism map ϕ between the two domains and the associated operator $g \in \mathcal{G}$ over a subspace of the feature space, such that the results of g are similar when the labels are identical (regardless of the domains).

B. Universality of Homomorphism Operator

For a better understanding of the homomorphism operator, some widely used homomorphism operators are introduced in

the following text. These operators not only show us the universality of the homomorphism phenomenon on the real-world data but also provide us with some ideas of designing a new homomorphism operator.

1) Pairwise Comparison Operator: This operator is first explored in gene data [39], [40] for cancer classification. Gene pairs $x^j_{\mathrm{iv}_1} < x^j_{\mathrm{iv}_2}$ are selected for classification purpose, i.e., the subspace $\{v_1, v_2\}$ are selected to be $\mathcal{V}$ if the comparison result $x^j_{\mathrm{iv}_1} < x^j_{\mathrm{iv}_2}$ remains consistent for any domain z_j. The corresponding homomorphism operator g on subspace $\{v_1, v_2\}$ is as follows. Here, $\mathbf{1}\{\cdot\}$ is the indicator function

$$g\left(\boldsymbol{x}^j_i\right) = \mathbf{1}\left\{x^j_{iv_1} < x^j_{iv_2}\right\}. \tag{4}$$

2) Pairwise Ratio Operator: This operator explores pairwise ratio on gene expression pairs for classification [41]. In detail, the feature pairs with stable pairwise ratio are selected for classification purpose, i.e., $x^j_{\mathrm{iv}_1}/x^j_{\mathrm{iv}_2}$ is stable for any domains z_j. The corresponding homomorphism operator g is defined as

$$g\left(\boldsymbol{x}^j_i\right) = x^j_{iv_1}/x^j_{iv_2}. \tag{5}$$

3) Causal Mechanism Operator: This operator is based on the causal mechanism behind the data. Ingenerally, it is a common assumption that samples with a certain label are generated by its corresponding causal mechanism [44]–[46]. The causal mechanism is invariant across the domains but different among the labels. Typical examples include, in the gene sample classification task, the gene regulation network are generally the same through the samples obtained from different platforms; in the image classification tasks, the eyes are always inside the face region [44], though the images are taken using different cameras or with various angles. Assume the causal mechanism $x^j_{\mathrm{iv}_1} = \psi(x^j_{\mathrm{iv}_2}, x^j_{\mathrm{iv}_3}, \ldots, x^j_{\mathrm{iv}_r})$ holds for all the samples with $y^j_i = y_0$, the corresponding homomorphism operator is given as follows:

$$g\left(\boldsymbol{x}^j_i\right) = x^j_{iv_1} - \psi\left(x^j_{iv_2}, x^j_{iv_3}, \ldots, x^j_{iv_r}\right). \tag{6}$$

IV. Model

A. Learning Framework

Based on the above-mentioned observations of homomorphism among domains, the task of domain adaptation is to learn the homomorphism operator $g \in \mathcal{G}$ and the classifier $f \in \mathcal{F}$, by solving the following constrained optimization problem:

$$\begin{aligned} &\arg\min_{f\in\mathcal{F}, g\in\mathcal{G}} \frac{1}{m_S n}\sum_{j=1}^{m_S}\sum_{i=1}^{n}\mathcal{L}\left(f\left(g\left(\boldsymbol{x}^j_i\right)\right), y^j_i\right) \\ &\text{s.t.}\quad g\left(\boldsymbol{x}^{j_1}_{i_1}\right) = g\left(\boldsymbol{x}^{j_2}_{i_2}\right) \text{ if } y^{j_1}_{i_1} = y^{j_2}_{i_2} \\ &\quad\text{for}\quad \forall i_1, i_2 \in \left\{1, 2, \ldots n\right\}, j_1, j_2 \in \left\{1, 2, \ldots m\right\} \end{aligned} \tag{7}$$

where $m = m_S + m_T$ is the total number of domains. In this problem, the objective function is the prediction loss, and the constraint is based on the homomorphism of g.

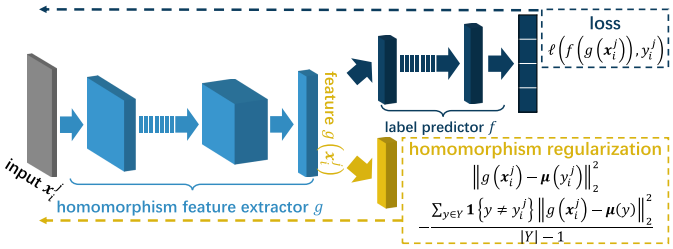

Fig. 2. Architecture of the proposed neural network solution consists of three parts: 1) the homomorphic feature extractor g; 2) the classifier f; and 3) the homomorphism regularization.

Considering the difficulty of solving the above-mentioned constrained minimization problem, we derive the unconstrained form

$$\arg\min_{f\in\mathcal{F}, g\in\mathcal{G}} \frac{1}{m_S n}\sum_{j=1}^{m_S}\sum_{i=1}^{n}\mathcal{L}\left(f\left(g\left(\boldsymbol{x}^j_i\right)\right), y^j_i\right) + \lambda_1\mathcal{R}_1 - \lambda_2\mathcal{R}_2 \tag{8}$$

where $\mathcal{R}_1$ and $\mathcal{R}_2$ are two regularization terms ensuring the homomorphism of operator g, defined as follows:

$$\mathcal{R}_1 = \frac{1}{2mn}\sum_{j=1}^{m}\sum_{i=1}^{n}\left\|g\left(\boldsymbol{x}^j_i\right) - \boldsymbol{\mu}\left(y^j_i\right)\right\|_2^2 \tag{9}$$

$$\mathcal{R}_2 = \frac{1}{2mn\left(|Y|-1\right)}\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{y\in Y}\mathbf{1}\left\{y \neq y^j_i\right\}\left\|g\left(\boldsymbol{x}^j_i\right) - \boldsymbol{\mu}(y)\right\|_2^2 \tag{10}$$

where $n(y)$ is the total number of samples with label y, and $\boldsymbol{\mu}(y)$ is defined as $\boldsymbol{\mu}(y) = \frac{1}{n(y)}\sum_{j=1}^{m}\sum_{i=1}^{n}\mathbf{1}\{y^j_i = y\}g(x^j_i)$, which represents the center of the homomorphism representation of samples with label y.

The regularization term $\mathcal{R}_1$ is introduced to enforce homomorphism by minimizing the intraclass variance, while $\mathcal{R}_2$ is incorporated to ensure the discriminative ability by maximizing the interclass variance of features. λ_1 and λ_2 are scaler parameters to control the impact of $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. Although it is not necessary to incorporate $\mathcal{R}_2$ for enforcement of homomorphism in the objective function, it has two merits to have R_2 in the objective function. First, it helps prevent feature congestion in the feature space. Second, it is useful for initialization in the training phase. These two regularization terms are also closely related to the minimum intraclass distance and maximum interclass distance used in various machine learning tasks [47], [48].

Considering the various forms of g and f, a flexible and feasible realization of the above-mentioned learning framework is to build a neural network with three subnets, as shown in Fig. 2. The first subnet on the left is designed for the homomorphism operator g. The second subnet on the upper right is devised for the classifier function f. The last subnet (lower right) estimates the homomorphism regularization terms. The whole network is trained with the objective function in (8).

The training algorithm of the above-mentioned network is given in Algorithm 1. It consists of a forward estimation procedure and a backward updating procedure. The forward procedure estimates the pseudolabel of the unlabeled samples

in the training set, and this step is used only when we want to use the information of target domains. The backward procedure updates the parameters θ_g and θ_f using the stochastic gradient descent. Here, θ_g is the parameters of feature extractor g and θ_f is the parameters of classifier f. The learning procedure is controlled by the learning rate η. In the parameter updating step, θ_f is updated only by the prediction loss, while θ_g is updated by both the prediction loss and the regularization term since g exerts both the prediction loss and the regularization term, while f only affects the prediction loss.

Algorithm 1 Stochastic gradient descent training of deep neural network with homomorphism constraint.

Input: samples $\{\boldsymbol{x}_i^j\}_{j=1}^m$ and labels $\{y_i^j\}_{j=1}^{m_S}$
Output: θ_g and θ_f
for number of training iterations **do**
  $Y \leftarrow \{\}$
  for $i = 1, \ldots n,\ \ j = m_S + 1, \ldots m$ **do**
    $y_i^j \leftarrow f(g(\boldsymbol{x}_i^j))$
  end for
  for $i = 1, \ldots n,\ \ j = 1, \ldots m$ **do**
    if $y_i^j \notin Y$ **then**
      $Y \leftarrow Y \cup \{y_i^j\}$
      $\boldsymbol{\mu}(y_i^j) \leftarrow \mathbf{0}$
      $n(y_i^j) \leftarrow 0$
    end if
    $\boldsymbol{\mu}(y_i^j) \leftarrow \frac{n(y_i^j)}{n(y_i^j)+1}\boldsymbol{\mu}(y_i^j) + \frac{1}{n(y_i^j)+1} g(\boldsymbol{x}_i^j)$
    $n(y_i^j) \leftarrow n(y_i^j) + 1$
  end for
  Update θ_f and θ_g

$$\theta_f \leftarrow \theta_f - \eta \frac{\partial \mathcal{L}}{\partial \theta_f}$$

$$\theta_g \leftarrow \theta_g - \eta\left(\frac{\partial \mathcal{L}}{\partial \theta_g} + \lambda_1 \frac{\partial \mathcal{R}_1}{\partial \theta_g} - \lambda_2 \frac{\partial \mathcal{R}_2}{\partial \theta_g}\right)$$

end for

Note that the unsupervised approach proposed in Section IV-A can be easily adapted to the semisupervised case. In detail, for the semisupervised setting, we can use the samples from the source domain and only the labeled sample from the target domain in our model [turn OFF the step $y_i^j \leftarrow f(g(\boldsymbol{x}_i^j))$], or we can also use the unlabeled samples from the target domain by using the predicted pseudolabels as the labels of the corresponding samples.

B. Generalization Bound

In this section, we establish the generalization bound for the proposed framework, with respect to the number of domains and samples.

We start with a probabilistic form of the homomorphism. According to the definition of homomorphism, for any domain $z_j \in \mathbb{Z}$ and any extractor $g \in \mathcal{G}$

$$\mathbb{E}_{P(\boldsymbol{X}Y|z_j)}[\mathbf{1}\{g(\boldsymbol{x}) = \boldsymbol{\mu}(y)\}] = 1 \tag{11}$$

where $\boldsymbol{\mu}(y)$ represents the homomorphic feature with label y. An equivalent expression of (11) is as follows:

$$\mathbb{E}_{P(\boldsymbol{X}Y|z_j)}[\mathbf{1}\{g(\boldsymbol{x}) \neq \boldsymbol{\mu}(y)\}] = 0. \tag{12}$$

With the help of (12), we further assume the existing of the homomorphic operator space.

Assumption 1 (Homomorphic Operator Space): For any domain $z_j \in \mathbb{Z}$ and any operator $g \in \mathcal{G}$, the expected value $\mathbb{E}_{P(\mathbf{XY}|z_j)}[\mathbf{1}\{g(\boldsymbol{x}) \neq \boldsymbol{\mu}(y)\}]$ is independently drawn from a distribution with the expectation $\mathbb{E}_{P(\mathbf{XYZ})}[\mathbf{1}\{g(\boldsymbol{x}) \neq \boldsymbol{\mu}(y)\}]$, where $\boldsymbol{\mu}(y)$ is the homomorphic features with label y.

This assumption implies that for any operator $g \in \mathcal{G}$, $\mathbb{E}_{P(\mathbf{XY}|z_S)}[\mathbf{1}\{g(\boldsymbol{x}) \neq \boldsymbol{\mu}(y)\}]$ and $\mathbb{E}_{P(\mathbf{XY}|z_T)}[\mathbf{1}\{g(\boldsymbol{x}) \neq \boldsymbol{\mu}(y)\}]$ have the same expectations, where $z_S \in \mathbb{Z}_S$ and $z_T \in \mathbb{Z}_T$ are the source domain and target domain, respectively. In another word, the results of the homomorphic operator $g \in \mathcal{G}$ is consistent on all the source and target domains.

Before delving into the theoretical analysis of the generalization bound, the following two lemmas are introduced as the preliminaries.

Lemma 1 (Hoeffding's Inequality [49]): Let $\xi_1, \xi_2, \ldots, \xi_m$ be the independent random variables bounded by $[0, 1]$, and $\widehat{\xi} = (1/m0)\sum_{j=1}^m \xi_j$ be their empirical mean. For any $\delta \in (0, 1)$

$$P\left(\mathbb{E}\left[\widehat{\xi}\right] - \widehat{\xi} \leq \sqrt{\frac{\log \frac{1}{\delta}}{2m}}\right) \geq 1 - \delta$$

where $\mathbb{E}[\widehat{\xi}]$ is the expectation of empirical mean.

The Hoeffding inequality shows that if all the variables take values in the identical space, the mean of them converges to the expectation of their mean in probability, with the increasing number of variables. This property can be used for the proof of our main theorem and the establishment of the following lemma.

Lemma 2 (VC Generalization Bound [50]): Let $h \in \mathcal{H}$ be a hypothesis defined on the space with Vapnik-Chervonenkis (VC) dimension $d_{\mathcal{H}}$, $E(h) := \mathbb{E}_{P(\mathbf{XY})}[\mathbf{1}\{h(\boldsymbol{x}) \neq y\}]$ be the expected error, and $\widehat{E}(h) := \frac{1}{n}\sum_{i=1}^n \mathbf{1}\{h(\boldsymbol{x}_i) \neq y_i\}$ be the empirical error of n instances. If all the observed instances are independent and identically distributed, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$

$$E(h) - \widehat{E}(h) \leq \sqrt{\frac{2}{n}\log 2 + \frac{2d_{\mathcal{H}}}{n}\log\left(\frac{2ne}{d_{\mathcal{H}}}\right) + \frac{2}{n}\log\frac{1}{\delta}}.$$

The VC bound implies that the empirical error is asymptotically and uniformly close to the expected error in probability, with the increasing number of samples. Assuming the existing of sufficient data and each domain contains n samples. We could only focus on the optimality of the empirical error and obtain the following generalization bound of DACH.

Theorem 1: Let $f \circ g \in \mathcal{F} \circ \mathcal{G}$ be a compound function defined on the hypothesis space with VC dimension $d_{\mathcal{F}\circ\mathcal{G}}$, $\ell_{\text{fg}}(\boldsymbol{x}, y)$ be the zero-one loss function

$\mathbf{1}\{f \circ g(\boldsymbol{x}) \neq y\}$, $E(f,g) := \mathbb{E}_{P(\mathbf{XYZ})}\left[\ell_{\text{fg}}(\boldsymbol{x}, y)\right]$ be the expected error on the joint distribution $P(\mathbf{XYZ})$, and $\widehat{E}(f,g) := \frac{1}{\text{mn}}\sum_{j=1}^{m}\sum_{i=1}^{n}\ell_{\text{fg}}(\boldsymbol{x}_i^j, y_i^j)$ be the empirical error of the full domain set. If all the instances are independent and identically distributed from their respective domains, then for any $\delta \in (0,1)$

$$\begin{aligned} &E(f,g) - \widehat{E}(f,g) \\ &\leq \mathbb{E}_{P(Z)}\left[P\left(\mathcal{C}_g|z\right)\right] - \mathbb{E}\left[\frac{1}{m}\sum_{j=1}^{m}P\left(\mathcal{C}_g|z_j\right)\right] \\ &\quad + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} + \sqrt{\frac{2}{n}\log 2 + \frac{2d_{\mathcal{F}\circ\mathcal{G}}}{n}\log\left(\frac{2ne}{d_{\mathcal{F}\circ\mathcal{G}}}\right) + \frac{2}{n}\log\frac{1}{\delta}} \end{aligned}$$

holds with probability at least $(1-\delta)^2$. Here, $P(\mathcal{C}_g|z) := \mathbb{E}_{P(\mathbf{XY}|z)}\left[\mathcal{C}_g(\boldsymbol{x}, \boldsymbol{\mu}(y))\right]$ and $P(\mathcal{C}_g|z_j) := \mathbb{E}_{P(\mathbf{XY}|z_j)}\left[\mathcal{C}_g(\boldsymbol{x}, \boldsymbol{\mu}(y))\right]$.

Proof: Recall the definitions of expected error $E(f,g)$ and empirical error $\widehat{E}(f,g)$. The gap consists of the interdomain subgap $\mathcal{B}_1 + \mathcal{B}_2$, and the intradomain subgap $\mathcal{B}_3$

$$\begin{aligned} &E(f,g) - \widehat{E}(f,g) \\ &= \mathbb{E}_{P(XYZ)}\left[\ell_{fg}(\boldsymbol{x}, y)\right] - \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n}\ell_{fg}(\boldsymbol{x}_i^j, y_i^j) \\ &= \mathbb{E}_{P(XYZ)}\left[\ell_{fg}(\boldsymbol{x}, y)\right] - \mathbb{E}\left[\frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_{P(XY|z_j)}\left[\ell_{fg}(\boldsymbol{x}, y)\right]\right] \\ &\quad + \mathbb{E}\left[\frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_{P(XY|z_j)}\left[\ell_{fg}(\boldsymbol{x}, y)\right]\right] \\ &\quad - \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_{P(XY|z_j)}\left[\ell_{fg}(\boldsymbol{x}, y)\right] \\ &\quad + \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_{P(XY|z_j)}\left[\ell_{fg}(\boldsymbol{x}, y)\right] - \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n}\ell_{fg}(\boldsymbol{x}_i^j, y_i^j) \\ &= \mathcal{B}_1 + \mathcal{B}_2 + \mathcal{B}_3. \end{aligned} \tag{13}$$

Here, $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$, are the first to third lines of the last equation, respectively.

For the subgap $\mathcal{B}_1$, we first define the one-zero loss between $\boldsymbol{x}$ and $\boldsymbol{\mu}(y)$ as $\mathcal{C}_g(\boldsymbol{x}, \boldsymbol{\mu}(y)) := \mathbf{1}\{g(\boldsymbol{x}) \neq \boldsymbol{\mu}(y)\}$. Since the event $g(\boldsymbol{x}) = \boldsymbol{\mu}(y)$ is a sufficient condition for $f \circ g(\boldsymbol{x}) = y$, the term $\ell_{\text{fg}}(\boldsymbol{x}, y)$ can be replaced by $\mathcal{C}_g(\boldsymbol{x}, \boldsymbol{\mu}(y))$. Thus, $\mathcal{B}_1$ satisfies the following inequality:

$$\begin{aligned} \mathcal{B}_1 &\leq \mathbb{E}_{P(XYZ)}\left[\mathcal{C}_g(\boldsymbol{x}, \boldsymbol{\mu}(y))\right] - \mathbb{E}\left[\frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_{P(XY|z_j)}\left[\mathcal{C}_g(\boldsymbol{x}, \boldsymbol{\mu}(y))\right]\right] \\ &= \mathbb{E}_{P(Z)}\left[P(\mathcal{C}_g|z)\right] - \mathbb{E}\left[\frac{1}{m}\sum_{j=1}^{m}P(\mathcal{C}_g|z_j)\right]. \end{aligned} \tag{14}$$

For the subgap $\mathcal{B}_2$, we replace ξ_j with $\mathbb{E}_{P(\mathbf{XY}|z_j)}\left[\ell_{\text{fg}}(\boldsymbol{x}, y)\right]$ in Lemma 1. Therefore, for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have

$$\mathcal{B}_2 \leq \sqrt{\frac{\log\frac{1}{\delta}}{2m}}. \tag{15}$$

For the subgap $\mathcal{B}_3$, we have the following transformation:

$$\begin{aligned} \mathcal{B}_3 &= \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_{P(XY|z_j)}\left[\ell_{fg}(\boldsymbol{x}, y)\right] - \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n}\ell_{fg}\left(\boldsymbol{x}_i^j, y_i^j\right) \\ &= \mathbb{E}_{\frac{1}{m}\sum_{j=1}^{m}P(XY|z_j)}\left[\ell_{fg}(\boldsymbol{x}, y)\right] - \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{m}\sum_{j=1}^{m}\ell_{fg}\left(\boldsymbol{x}_i^j, y_i^j\right)\right]. \end{aligned}$$

It is easy to see that $1/m\sum_{j=1}^{m}\ell_{\text{fg}}(\boldsymbol{x}_i^j, y_i^j)$ is independent and identically drawn from a distribution with expected value $\mathbb{E}_{1/m\sum_{j=1}^{m}P(\mathbf{XY}|z_j)}[\ell_{\text{fg}}(\boldsymbol{x}, y)]$. By Lemma 2, we could replace the term $E(h)$ with $\mathbb{E}_{\frac{1}{m}\sum_{j=1}^{m}P(\mathbf{XY}|z_j)}[\ell_{\text{fg}}(\boldsymbol{x}, y)]$, and $\widehat{E}(h)$ with $\frac{1}{n}\sum_{i=1}^{n}[\frac{1}{m}\sum_{j=1}^{m}\ell_{\text{fg}}(\boldsymbol{x}_i^j, y_i^j)]$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$

$$\mathcal{B}_3 \leq \sqrt{\frac{2}{n}\log 2 + \frac{2d_{\mathcal{F}\circ\mathcal{G}}}{n}\log\left(\frac{2ne}{d_{\mathcal{F}\circ\mathcal{G}}}\right) + \frac{2}{n}\log\frac{1}{\delta}}. \tag{16}$$

Plugging (14)–(16) into (13) completes the proof of Theorem 1. ■

Theorem 1 implies that the generalization bound are determined by two parts: 1) *domain effect*, i.e., $\mathbb{E}_{P(Z)}[P(\mathcal{C}_g|z)] - \mathbb{E}[(1)/(m)\sum_{j=1}^{m}P(\mathcal{C}_g|z_j)] + ((\log(1)/(\delta))/(2m))^{1/2}$, and 2) *sample effect*, i.e., $((2)/(n)\log 2 + (2d_{\mathcal{F}\circ\mathcal{G}})/(n)\log((2\text{ne})/(d_{\mathcal{F}\circ\mathcal{G}})) + (2)/(n)\log(1)/(\delta))^{1/2}$. The sample effect is the same as the traditional VC-theory, i.e., the bound converges with the increasing number of samples. Thus, the sample effect is not discussed in this article. Following, we will focus on the convergence of the domain effect, considering whether the homomorphism assumption holds.

When the homomorphism Assumption 1 holds, and further assume the training algorithms can find the correct g, then the $\mathbb{E}_{P(Z)}[P(\mathcal{C}_g|z)] - \mathbb{E}[(1)/(m)\sum_{j=1}^{m}P(\mathcal{C}_g|z_j)]$ is zero, the domain effect is reduced to $((\log(1)/(\delta))/(2m))^{1/2}$, and it convergences to 0 with the increasing number of domains. We will also test this point in the experiments.

Even when the homomorphism Assumption 1 does not hold, the bound also gives some interesting results. In detail, if the number of domains is sufficient, the upper bound could be minimized by approximating $\mathbb{E}_{P(Z)}[P(\mathcal{C}_g|z)]$ with $\mathbb{E}[(1)/(m)\sum_{j=1}^{m}P(\mathcal{C}_g|z_j)]$. Thus, the bound also convergences to 0 with the increasing number of domains, with the help of the constraint $g(\boldsymbol{x}) = \boldsymbol{\mu}(y)$.

V. Concrete Algorithms

To illuminate the application of the above-mentioned proposed abstract framework in the real-world problems, two concrete algorithms are devised for two typical domain adaptation tasks, i.e., sequence data classification and image data classification. As a general framework, we only need to specify the concrete homomorphism operator and the deep learning structure for each task. The other details are the same as the general framework and are ignored in this section.

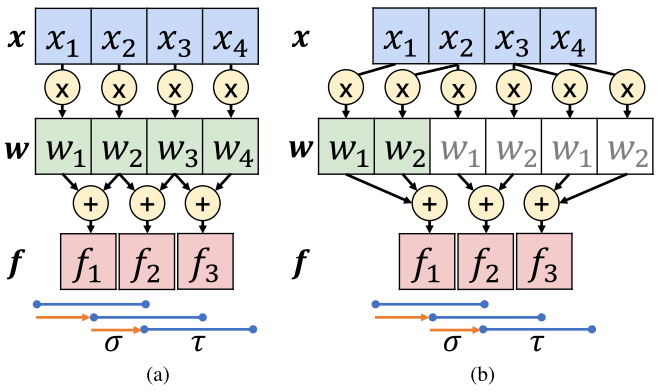

Fig. 3. Gridding operator is a homomorphism operator designed for sequence data. In this example, the input sequence is $\boldsymbol{x} = x_1 \dots x_3$ ($L = 3$), the sliding window size τ is 2, and the step size σ is 1. For comparison, the convolution operator is shown in (b). (a) Gridding 1-D. (b) Convolution 1-D.

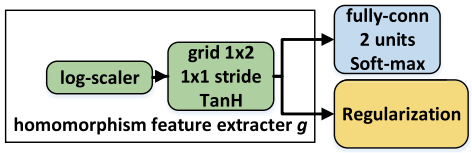

Fig. 4. Architecture of neural network for gender prediction on the sequence data.

A. Sequence Classification

1) Homomorphism Operator Design: For the sequence data, the homomorphism operator usually exists among the neighborhoods of the features, and the following gridding layer is proposed to extract the homomorphic subspaces among the domains. As shown in the Fig. 3, the operator scans over the sequence with window size τ and step size σ, and the output is a linear combination of the features insides the sliding windows. The output of the operator is defined as follows:

$$\begin{aligned} f_u &= \mathrm{g}_u(x_{(u-1)\sigma+1} \dots x_{(u-1)\sigma+\tau}) \\ &= \sum_{\alpha=1}^{\tau} w_{(u-1)\sigma+\alpha} \cdot x_{(u-1)\sigma+\alpha} + b_u. \end{aligned} \tag{17}$$

The difference between the gridding layer and conventional convolution layer is that the parameters of the gridding layer are different across the sliding windows, while the parameters of the convolution layer are shared across the sliding windows. The gridding operator allows different filters to be applied to different subspaces of the gene expression data. Meanwhile, weighing every variable in the scalable region helps the model to select meaningful variables in data.

2) Network Structure Design: The proposed domain adaptation approach is applied to sequence data for classification. The realization of the proposed architecture (Fig. 4) contains four components. In detail, the log-scaler layer and the gridding layer are used as the homomorphism feature extractor, and the homomorphism regularization layer is inserted before the last fully connected prediction layer. Note that normalization is necessary due to huge differences in value ranges of samples from different domains. Given an input variable x_v, the following normalization function is applied:

$$x'_v = \mathrm{sign}(x_v) \log(1 + |x_v|). \tag{18}$$

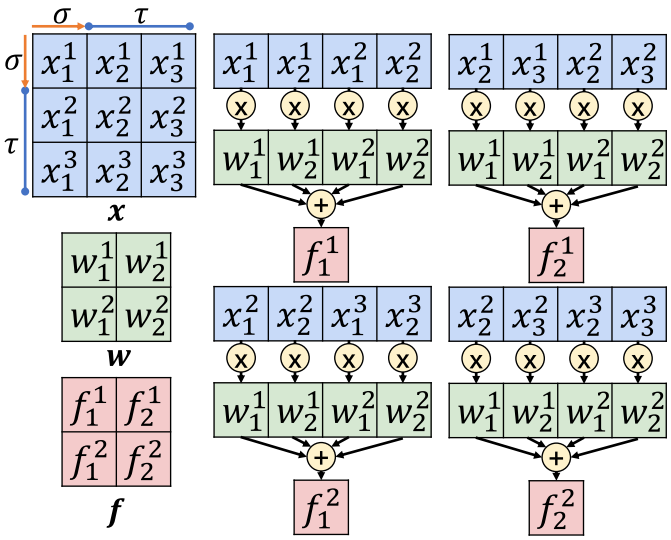

Fig. 5. Convolution-based homomorphism operator on the image data. In this example, the input sample is $\boldsymbol{x} = [[x_1^1 \dots x_3^1], \dots, [x_1^3 \dots x_3^3]]$ ($L \times L = 3 \times 3$), the kernel size τ is 2, and the stride size is $\sigma = 1$.

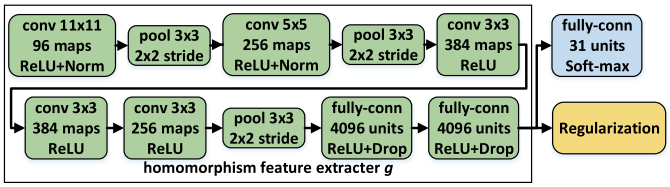

Fig. 6. Architecture of the neural network for label prediction on image data.

B. Image Classification

1) Homomorphism Operator Design: Different from the sequence data, the image data shows homomorphic properties among the 2-D neighborhood. As is known in the existing studies, convolution layer is designed to mine the homomorphic subspace between domains [2], [16], [17], [51]. The convolution operator could be viewed as the fully weight-shared version of 2-D gridding layer, as shown in Fig. 5. Let τ denote the kernel size of g, which is the dimension of the local receptive field that each operator will be applied on. σ is the stride size at which the sliding kernel is moving. Given input samples $\boldsymbol{x} \in \mathbb{R}^{L \times L}$, the output features $\boldsymbol{f}$ are defined as

$$\begin{aligned} f_v^u &= \mathrm{g}_v^u \left(x_{(v-1)\sigma+1}^{(u-1)\sigma+1} \dots x_{(v-1)\sigma+\tau}^{(u-1)\sigma+\tau} \right) \\ &= \left(\sum_{\alpha=1}^{\tau} \sum_{\beta=1}^{\tau} w_\beta^\alpha \cdot x_{(v-1)\sigma+\beta}^{(u-1)\sigma+\alpha} \right) + b_v^u. \end{aligned} \tag{19}$$

2) Network Structure Design: The architecture of the underlying neural network extended from AlexNet [52] is shown in Fig. 6. In detail, the feature extractor layers are used as the homomorphism feature extractor, and the homomorphism regularization layer is inserted before the last fully connected prediction layer. For suitable use in our image data set, the final fully connected predictor is modified with 31 units. Normalization, established by the literature through long-social practice, is that all samples are subtracted by the mean of ImageNet on ILSVRC12.

VI. Experiments

The two proposed concrete algorithms are tested on the sequence data and image data, respectively. Some shared experiment setup is given as follows. The classification accuracy is the main metric for the evaluation of the algorithms. The concrete algorithms are implemented on the CAFFE [53] platform. The experiments are conducted on a server with Intel Xeon, Nvidia Tesla K80, and CentOS Linux 7.3.1611.

The baseline algorithms include the following state-of-the-art algorithms, LatentDA [20], gradient reversal layer (GRL) [2], DRCN [51], DAN [17], and DDC [16]. Among them, LatentDA normalizes the data using different parameters for each domain. GRL and DRCN focus on finding a domain invariant feature subspace, while DAN and DDC try to align the distributions of the data using the MMD. Note that some methods are slightly modified in the multisource domain experiments. In detail, GRL maximizes the domain prediction error by the softmax loss instead of the logistic loss. DAN and DDC maximize the mean discrepancy for any two domains. The hyperparameters of the baseline algorithms are based on the optimal setting of the original work. The hyperparameters of DACH are as follows, the learning rate η is 0.01 and 0.001 for the sequence data and the image data, respectively, the corresponding annealing police is "inv," the weights λ_1 and λ_2 are 0.2 and 0.1, respectively, and the maximal iteration is 500 and 5000 for the sequence data and the image data, respectively. Furthermore, since the pseudolabel may be noisy, the gradients of the regularization terms are scaled according to the gradient of the label predictor, as follows, $\frac{\partial \mathcal{R}_1}{\partial \theta_g} \leftarrow \frac{\partial \mathcal{R}_1}{\partial \theta_g} \cdot \left(\left\| \frac{\partial \mathcal{L}}{\partial \theta_g} \right\|_2 / \left\| \frac{\partial \mathcal{R}_1}{\partial \theta_g} \right\|_2 \right)$ and $\frac{\partial \mathcal{R}_2}{\partial \theta_g} \leftarrow \frac{\partial \mathcal{R}_2}{\partial \theta_g} \cdot \left(\left\| \frac{\partial \mathcal{L}}{\partial \theta_g} \right\|_2 / \left\| \frac{\partial \mathcal{R}_2}{\partial \theta_g} \right\|_2 \right)$, where $\|\cdot\|_2$ is the L_2 norm.

A. Results on Sequence Data

For the domain adaptation on the sequence data, we test the algorithms on the gene expression omnibus (GEO) data. GEO [54] is a public repository that archives and distributes comprehensive sets of gene expression data submitted by the scientific community. GEO now consists of more than 300K samples. Each sample includes the gene expression level and its corresponding sample information. The data are collected from more than 300 different platforms, and a lot of them are *without* platform information. Viewing each platform as one domain, we randomly select seven domains from GEO with a total of 36 033 samples for the following experiments, details of which are listed in Table II. In all the following experiment, the normalization process is given in 18 is used for all the compared algorithms.

Extensive experiments are conducted by varying the number of source domains and the ratio of labels in each domain. Detailed results with different number of source domains and ratio of labels are given in Tables III and IV, respectively. All the algorithms and two homomorphism operators are compared under the above-mentioned two sets of configurations. The experimental results are analyzed in the following five different aspects.

TABLE II
Statistics of GEO Data Set

Domain	Male	Female	Total
GPL570	10273	9098	19371
GPL4133	1119	882	2001
GPL6102	383	464	847
GPL6480	1203	980	2183
GPL6884	1008	928	1936
GPL6947	3046	2533	5579
GPL10558	1898	2218	4116

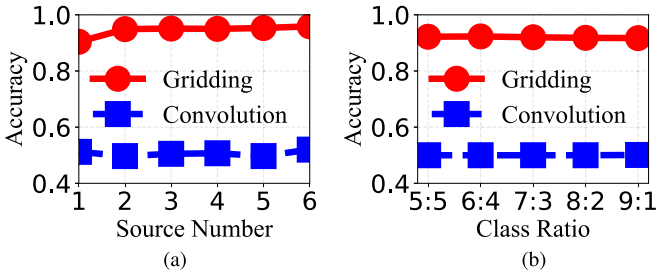

Fig. 7. Comparison of the homomorphism operator. On the sequence data, the gridding operator outperforms the convolution operator, regardless of the source domain number and class ratio. (a) Varying the source number. (b) Varying the class ratio.

1) Effect of Homomorphism Operator: Fig. 7 plots the accuracy of the two operators with a different number of source domains and a different ratio of labels. The original results can be found in Tables III and IV, respectively. As shown in Fig. 7, the gridding operator constantly outperforms the convolution operator over all the configurations. This reflects that the gridding operator is an effective homomorphism operator on the sequence data. This result is consistent with the existing results on the gene expression data analysis, e.g., the frequently used pairwise order among the neighborhoods of the genes [40]. The result of this experiment also shows the importance of homomorphism operator design, which is essential to the extraction of invariant information among the domains.

2) Effect of the Source Domain Number: In accordance with the second column of Table III, Fig. 7(a) plots the accuracy of DACH with a different number of source domains. As shown in the red curve, the performance of DACH improves with the increasing number of domains. The above-mentioned phenomenon also verifies the theoretical results given in Theorem 1. In detail, the domain effect part of the generalization bound is $((\log(1)/(\delta))/(2m))^{1/2}$, and it convergences to 0 with the increasing number of domains. Intuitively, this is because a large number of domains are helpful for the algorithm to eliminate the domain specified information while preserving the shared information across the domains.

3) Existence of Homomorphism Subspace: The right part of Table III shows the top-five variable blocks in different domain configurations. As shown in the table, the top-five variable blocks are very stable across all the configurations with the different number of source domains. In detail, the blocks coded 6519, 6518, 4412, 4413, 9050, and 9051 form a subspace, which preserves the homomorphism among the domains. Fig. 8 also shows the t-SNE-based visualization of

TABLE III

RESULTS ON THE SEQUENCE DATA WITH VARYING NUMBER OF SOURCE DOMAINS. THE PERFORMANCE OF DACH-GRID INCREASES WITH THE INCREASING OF SOURCE DOMAIN NUMBERS AND THE TOP FIVE VARIABLE BLOCKS ARE STABLE, BUT RESULTS OF DACH-CONV DO NOT HAVE SUCH PROPERTY

Domain Configuration	**DACH-GRID**	**Top 5 variable blocks**				
570→10558	0.904±0.025	6519	4413	9050	4412	6518
6947 570→10558	0.949±0.004	6519	6518	4412	4413	9051
6480 6947 570→10558	0.951±0.003	6519	6518	4413	4412	9050
4133 6480 6947 570→10558	0.950±0.003	6519	6518	4413	4412	9050
6884 4133 6480 6947 570→10558	0.953±0.002	6519	6518	4413	4412	6520
6102 6884 4133 6480 6947 570→10558	**0.959±0.002**	6519	6518	4413	9050	9051
Domain Configuration	**DACH-CONV**	**Top 5 variable blocks**				
570→10558	0.512±0.026	9865	6865	13783	9035	13874
6947 570→10558	0.497±0.031	8965	8350	9871	12736	13513
6480 6947 570→10558	0.505±0.028	7432	9050	6518	4412	6679
4133 6480 6947 570→10558	0.507±0.030	9050	9051	5644	7433	4413
6884 4133 6480 6947 570→10558	0.497±0.031	9050	7041	9051	5644	6221
6102 6884 4133 6480 6947 570→10558	**0.520±0.027**	9050	7432	7041	6221	9051

TABLE IV

RESULTS ON THE SEQUENCE DATA WITH VARYING RATIO OF LABELS. THE PERFORMANCE OF DACH-GRID IS STABLE WITH THE RATIO OF LABELS AND SIGNIFICANTLY BETTER THAN THAT OF DACH-CONV

Ratio	DACH-GRID	LatentDA-GRID	GRL-GRID	DRCN-GRID	DAN-GRID	DDC-GRID
GPL6947:5:5 GPL10558:5:5 → GPL570	**0.922±0.001**	0.896±0.001	0.891±0.012	0.843±0.020	0.883±0.028	0.901±0.016
GPL6947:6:4 GPL10558:4:6 → GPL570	**0.923±0.001**	0.872±0.003	0.910±0.003	0.819±0.037	0.874±0.039	0.905±0.010
GPL6947:7:3 GPL10558:3:7 → GPL570	**0.920±0.002**	0.822±0.002	0.807±0.019	0.799±0.043	0.852±0.029	0.879±0.014
GPL6947:8:2 GPL10558:2:8 → GPL570	**0.918±0.002**	0.785±0.003	0.677±0.025	0.762±0.038	0.838±0.038	0.868±0.016
GPL6947:9:1 GPL10558:1:9 → GPL570	**0.917±0.001**	0.742±0.004	0.580±0.054	0.752±0.038	0.828±0.047	0.826±0.023
Ratio	DACH-CONV	LatentDA-CONV	GRL-CONV	DRCN-CONV	DAN-CONV	DDC-CONV
GPL6947:5:5 GPL10558:5:5 → GPL570	0.500±0.002	0.515±0.001	0.504±0.001	0.502±0.005	0.510±0.004	0.513±0.001
GPL6947:6:4 GPL10558:4:6 → GPL570	0.500±0.002	0.526±0.004	0.500±0.001	0.501±0.004	0.506±0.005	0.507±0.004
GPL6947:7:3 GPL10558:3:7 → GPL570	0.500±0.003	0.532±0.006	0.500±0.002	0.500±0.005	0.504±0.007	0.503±0.004
GPL6947:8:2 GPL10558:2:8 → GPL570	0.500±0.003	0.509±0.002	0.499±0.004	0.501±0.005	0.503±0.004	0.503±0.001
GPL6947:9:1 GPL10558:1:9 → GPL570	0.501±0.004	0.509±0.001	0.502±0.003	0.500±0.005	0.502±0.004	0.504±0.003

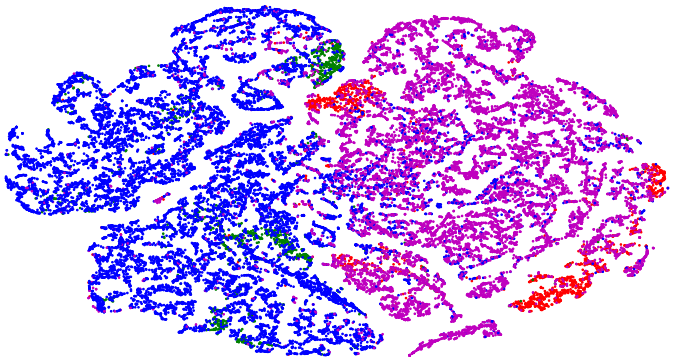

Fig. 8. t-SNE visualization on top-five variable blocks in different domains. Blue and magenta points represent the male and female samples from six source domains, respectively. Green and red points indicate the male and female samples from target domain, respectively.

the distribution on the last domain configuration. In Fig. 8, the blue and magenta points represent the male and female samples from the source domains, respectively, while green and red points represent the male and female samples from the target domain, respectively. It is easy to see that male and female are almost separated by the central axis, which implies the existence of homomorphic space. Furthermore, among the six blocks, the block 6519 arises most frequently in all experiments. A visualization of block 6519 is given in Fig. 9 to help gain a deep insight into the homomorphism operator. In Fig. 9, the blue and green histograms are the strength distributions of the male, while the red and magenta ones are for the female. As shown in Fig. 9, the distribution is stable across the configurations, which constantly reflects the existence of the homomorphism phenomenon among the domains.

4) Coupling Effect Between the Label and the Domain: In this part, we investigate the coupling effect between the label and the domain. In this set of experiments, by unbalanced sampling over the domains, the label information and domain information is coupled. As shown in Table IV, the ratio of labels (i.e., the ratio between the sample sizes belonging to two different labels) is changed to increase the coupling between the label and the domain. Take the fifth setting (GPL6947:9:1 GPL10558:1:9) as an example, 90% samples of GPL6947 belongs to class "male," while 90% samples of GPL10558 belong to the class "female" such that the domain and label are highly coupled.

As shown in Fig. 10, with the increasing of the coupling effect, the performance of the baseline methods decreases, while the performance of DACH is stable across all the configurations. This is because DACH is based on the homomorphism among the domains and is invariant to the domain-label coupling effect, while the baseline methods are not intended for this unbalanced scenery, and their constraints ignore classification mechanism and blindly pursuit feature and distribution

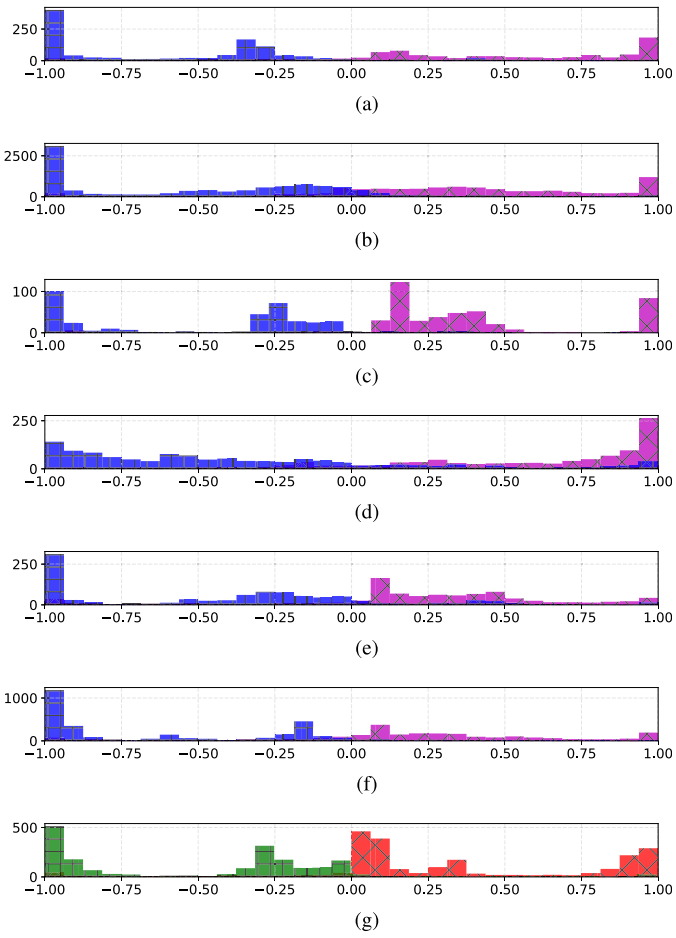

Fig. 9. Frequency histogram on top variable block in different domains. Blue and magenta histograms indicate the value distribution of the male and female samples of each source domain, respectively. Green and red histograms indicate the value distribution of the male and female samples from target domain, respectively. (a) Source: GPL4133. (b) Source: GPL570. (c) Source: GPL6102. (d) Source: GPL6480. (e) Source: GPL6884. (f) Source: GPL6947. (g) Target: GPL10558.

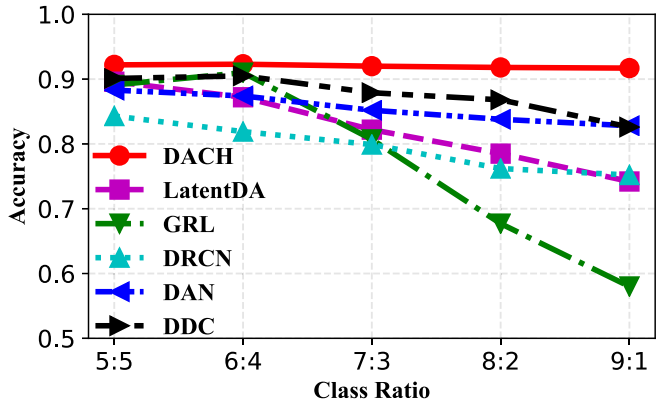

Fig. 10. Comparison of the methods with various coupling effect.

alignment. For example, GRL employs the domain label to search the domain invariant features. In such cases, the coupling effect raises the competition between the label and the domain and reduces the performance of GRL. Furthermore, LatentDA fits different distributions on different domains,

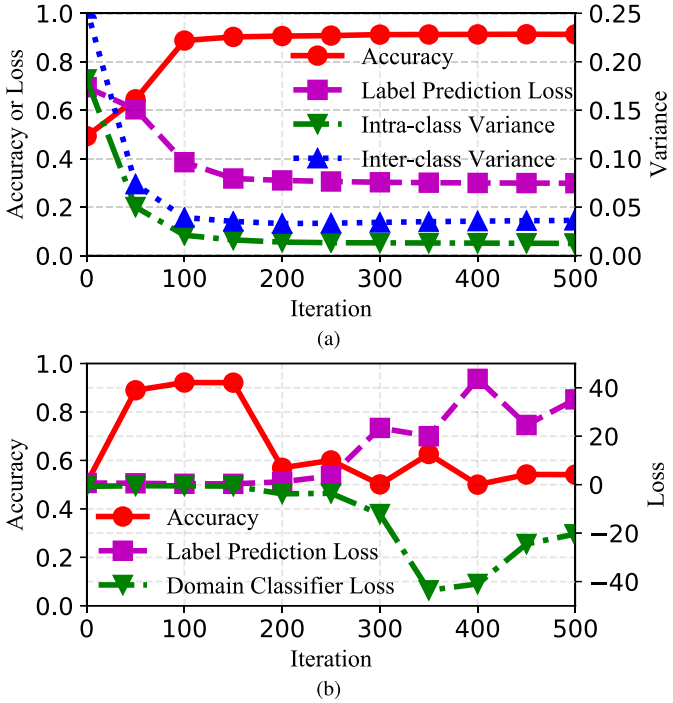

Fig. 11. Convergence of DACH and GRL with various coupling effect. (a) DACH. (b) GRL.

TABLE V
COMPARISON WITH THE STATE-OF-THE-ART ALGORITHMS

Method	Accuracy
DACH	**0.959±0.002**
LatentDA	0.917±0.007
GRL	0.921±0.003
DRCN	0.841±0.004
DAN	0.838±0.023
DDC	0.857±0.014
Train on Target	0.964±0.002

TABLE VI
STATISTICS OF OFFICE DATA SET

Domain	# of images	#of classes
AMAZON	2817	31
DSLR	498	31
WEBCAM	795	31

which further causes the difficulty in transferring the source domain information to the target domains.

To obtain a deep insight into the coupling effect, we plot the convergence performance of DACH and GRL experiments with label ratio 1:9 in Fig. 11. As shown in Fig. 11, DACH converges smoothly, while GRL does not converge. In detail, the label prediction loss and the domain classifier loss are competing with each other, which results in the nonconvergence of GRL.

5) Comparison With the State-of-the-Art Algorithms: Finally, we compare our approach with the state-of-the-art algorithms on the default configuration in Table V, on the configuration, adapting source 6102, 6884, 4133, 6480, 6947, and 570 to target 10 558. In Table V, the last row is the

TABLE VII

RESULTS ON THE IMAGE DATA WITH VARYING RATIO OF LABELS. THE PERFORMANCE OF DACH IS STABLE WITH THE RATIO OF LABELS AND SIGNIFICANTLY BETTER THAN THAT OF THE OTHER BASELINES

Ratio	DACH	LatentDA	GRL	DRCN	DAN	DDC
AMAZON:5:5 DSLR:5:5 → WEBCAM	**0.945±0.002**	0.817±0.001	0.926±0.004	0.938±0.002	0.751±0.010	0.907±0.009
AMAZON:6:4 DSLR:4:6 → WEBCAM	**0.944±0.001**	0.820±0.001	0.925±0.003	0.934±0.003	0.742±0.009	0.911±0.008
AMAZON:7:3 DSLR:3:7 → WEBCAM	**0.941±0.002**	0.814±0.001	0.787±0.013	0.923±0.005	0.722±0.009	0.901±0.009
AMAZON:8:2 DSLR:2:8 → WEBCAM	**0.941±0.001**	0.814±0.001	0.754±0.024	0.916±0.007	0.705±0.009	0.901±0.009
AMAZON:9:1 DSLR:1:9 → WEBCAM	**0.941±0.002**	0.801±0.001	0.731±0.012	0.915±0.010	0.702±0.013	0.903±0.012
AMAZON:5:5 WEBCAM:5:5 → DSLR	**0.979±0.002**	0.852±0.001	0.973±0.003	0.978±0.003	0.876±0.010	0.962±0.007
AMAZON:6:4 WEBCAM:4:6 → DSLR	**0.981±0.002**	0.841±0.001	0.970±0.004	0.975±0.003	0.864±0.008	0.959±0.009
AMAZON:7:3 WEBCAM:3:7 → DSLR	**0.982±0.002**	0.835±0.001	0.812±0.015	0.973±0.007	0.833±0.009	0.957±0.004
AMAZON:8:2 WEBCAM:2:8 → DSLR	**0.982±0.002**	0.831±0.001	0.761±0.021	0.972±0.008	0.814±0.008	0.953±0.005
AMAZON:9:1 WEBCAM:1:9 → DSLR	**0.978±0.002**	0.832±0.001	0.743±0.016	0.972±0.005	0.811±0.011	0.942±0.004
DSLR:5:5 WEBCAM:5:5 → AMAZON	**0.617±0.002**	0.616±0.003	0.552±0.005	-	0.332±0.009	0.423±0.016
DSLR:6:4 WEBCAM:4:6 → AMAZON	0.619±0.003	**0.622±0.002**	0.534±0.006	-	0.328±0.007	0.397±0.009
DSLR:7:3 WEBCAM:3:7 → AMAZON	0.619±0.002	**0.621±0.002**	0.518±0.008	-	0.318±0.008	0.382±0.016
DSLR:8:2 WEBCAM:2:8 → AMAZON	**0.618±0.002**	**0.618±0.002**	0.498±0.005	-	0.297±0.008	0.375±0.006
DSLR:9:1 WEBCAM:1:9 → AMAZON	**0.618±0.004**	0.617±0.003	0.479±0.005	-	0.268±0.011	0.373±0.033

results of "Train on Target," which corresponds to the upper performance bound (i.e., if no adaptation is performed). As shown in Table V, our method outperforms the state-of-the-art algorithms, and the performance of DACH is close to the upper bound of the adaptation "Train on Target."

B. Results on Image Data

For the domain adaptation of the image data, we test the proposed concrete algorithm on the OFFICE data set. The OFFICE data set is a commonly used data set for evaluating domain adaptation methods [3]. It includes images from three domains, AMAZON, DSLR, and WEBCAM, with 31 different classes. Table VI lists the statistics of the OFFICE data set.

Due to the small number of domains, we do not vary the number of domains in this experiment. We report the evaluation results using the convolution operator only as it performs excellently on the image data. Thus, only the following two aspects, *coupling effect between the label and the domain* and *comparison with the state-of-the-art algorithms*, are analyzed in this set of experiments.

1) Coupling Effect Between the Label and the Domain: Similar to the experiments in the sequence data, we also explored the unbalanced sampling method to control the coupling effect between the label and the domain. The main difference is that there are 31 classes in the image data, while there are only two classes in the sequence data. Here, we divide the 31 classes into two sets and perform the unbalanced sampling over the two sets.

Table VII shows the comparison of the methods with varying coupling effect. As shown in Table VII, DACH keeps a good performance in all ratio settings while the performance of the other methods decreases with the increase of the coupling effect. From the perspective of assumptions of these methods, we can find that only our method DACH is intended for searching a unified classification decision space. LatentDA extracts the source-specified class information, and such information is not shared with the target domain in an extremely biased ratio setting. GRL assumes there exists a feature without domain-related bias for classification and may eliminate useful feature in unbalance setting. DRCN assumes the feature is useful if it is not only useful for prediction but also helpful for reconstruction, which inevitably causes the class-related feature to be biased and inclined to predict as the majority class. DAN and DDC use MMD and inherit the assumption that the two distributions with the same expectation are identical, and therefore, they face the same problem as DRCN. Note that DRCN collapses in the setting "DSLR & WEBCAM to AMAZON" (DW2A in short). This is because the variation between class-related features and reconstruction features is too huge, and the explosive reconstruction does not converge.

TABLE VIII

COMPARISON WITH THE STATE-OF-THE-ART ALGORITHMS ON THE IMAGE DATA. DACH ACHIEVES THE BEST PERFORMANCE ON AD2W AND AW2D AND THE SECOND BEST PERFORMANCE ON DW2A

Method	AD2W	AW2D	DW2A
DACH	**0.940±0.003**	**0.981±0.002**	0.585±0.003
LatentDA	0.931±0.003	0.943±0.002	**0.603±0.003**
TDD	0.918	0.946	0.489
GRL	0.934±0.002	0.967±0.001	0.538±0.002
DRCN	0.834±0.004	0.846±0.004	0.479±0.003
DAN	0.885±0.011	0.943±0.011	0.486±0.003
DDC	0.851±0.004	0.873±0.003	0.455±0.004
Train on Target	0.976±0.002	0.965±0.004	0.840±0.003

2) Comparison With the State-of-the-Art Algorithms: Finally, we compare our approach with the state-of-the-art algorithms on the default configuration in Table VIII. Similar to the experiments of the sequence data, the results of "Train on Target" is regarded as the upper bound of the domain adaptation. As shown in Table VIII, our method still works well and outperforms the state-of-the-art algorithms in all cases. In detail, the performance of DACH is close to the upper bound of the adaptation, in the configurations AD2W and AW2D. Note that all the methods have low accuracy in DW2A; this is because the AMAZON domain has a more complex distribution than the other two domains. However, DACH still achieves the suboptimal performance in the DW2A, which

shows DACH is able to catch the invariant information in the noisy case.

VII. Conclusion

We have proposed a new approach to adapt unlabeled target domain-based multiple labeled source domains by exploring the shared homomorphism property among the domains. Similar to many previous works in domain adaptation, the key idea is to search for a hidden space which is helpful for label prediction in all different domains. In a sense, the adaptation is mainly achieved by aligning the distributions in this hidden space. However, most of these works do not consider the relationship between label prediction and domain independence. If domain information contains label information, domain independence eradicates useful label information. Some models, such as [2], pursue domain independence blindly and do not work well in extremely unbalanced scenery.

In order to avoid repeated mistakes, our method learns a homomorphism space to predict the label. In this homomorphism space, features may be independent with the domain or not, but can help model more easily classify data. As long as the rules that try to minimize intraclass similarity and maximize interclass dissimilarity, the proposed model is able to select an appropriate feature space as final homomorphism space.

Acknowledgment

The authors would like to thank Dr. K. Zhang for his insightful discussions and suggestions.

References

[1] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. ICCV*, 2015, pp. 4068–4076.

[2] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.

[3] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.

[4] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *Proc. ICCV*, 2013, pp. 769–776.

[5] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, "Domain adaptation with conditional transferable components," in *Proc. ICML*, 2016, pp. 2839–2848.

[6] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Scholkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, Jul. 2006.

[7] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Proc. NIPS*, 2007, pp. 601–608.

[8] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *Proc. ICML*, 2013, pp. 222–230.

[9] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[10] G. Cai, Y. Wang, L. He, and M. Zhou, "Unsupervised domain adaptation with adversarial residual transform networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, pp. 1–14, 2019, doi: 10.1109/TNNLS.2019.2935384.

[11] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *Proc. ICML*, 2013, pp. 10–18.

[12] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, Jul. 2017.

[13] S. Li, S. Song, and G. Huang, "Prediction reweighting for domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1682–1695, Jul. 2017.

[14] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI*, 2016, pp. 2058–2065.

[15] P. Koniusz, Y. Tas, and F. Porikli, "Domain adaptation by mixture of alignments of second-or higher-order scatter tensors," 2016, *arXiv:1611.08195*. [Online]. Available: https://arxiv.org/abs/1611.08195

[16] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*. [Online]. Available: https://arxiv.org/abs/1412.3474

[17] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML*, 2015, pp. 97–105.

[18] B. Gong, K. Grauman, and F. Sha, "Reshaping visual datasets for domain adaptation," in *Proc. NIPS*, 2013, pp. 1286–1294.

[19] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "Just dial: Domain alignment layers for unsupervised domain adaptation," in *Proc. ICIAP*. Cham, Switzerland: Springer, 2017, pp. 357–369.

[20] M. Mancini, L. Porzi, S. R. Bulò, B. Caputo, and E. Ricci, "Boosting domain adaptation by discovering latent domains," 2018, *arXiv:1805.01386*. [Online]. Available: https://arxiv.org/abs/1805.01386

[21] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. ECCV*. Berlin, Germany: Springer, 2010, pp. 213–226.

[22] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko, "Efficient learning of domain-invariant image representations," 2013, *arXiv:1301.3224*. [Online]. Available: https://arxiv.org/abs/1301.3224

[23] H. Daumé III, A. Kumar, and A. Saha, "Frustratingly easy semi-supervised domain adaptation," in *Proc. Workshop Domain Adaptation Natural Lang. Process.*, 2010, pp. 53–59.

[24] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, "Domain adaptive dictionary learning," in *Proc. ECCV*. Berlin, Germany: Springer, 2012, pp. 631–645.

[25] T. Yao, Y. Pan, C.-W. Ngo, H. Li, and T. Mei, "Semi-supervised Domain Adaptation with Subspace Learning for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2142–2150.

[26] S. Basu, I. Davidson, and K. Wagstaff, *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Boca Raton, FL, USA: CRC Press, 2008.

[27] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko, "Discovering latent domains for multisource domain adaptation," in *Proc. ECCV*. Berlin, Germany: Springer, 2012, pp. 702–715.

[28] A. Bergamo and L. Torresani, "Exploiting weakly-labeled Web images to improve object classification: A domain adaptation approach," in *Proc. NIPS*, 2010, pp. 181–189.

[29] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1855–1862.

[30] Z. Xu and S. Sun, "Multi-view transfer learning with adaboost," in *Proc. IEEE 23rd Int. Conf. Tools Artif. Intell.*, Nov. 2011, pp. 399–402.

[31] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 668–675.

[32] L. Cheng and S. J. Pan, "Semi-supervised domain adaptation on manifolds," *IEEE Trans. Neural Netw. Learn.Syst.*, vol. 25, no. 12, pp. 2240–2249, Dec. 2014.

[33] Z. Ding, N. M. Nasrabadi, and Y. Fu, "Semi-supervised deep domain adaptation via coupled neural networks," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5214–5224, Nov. 2018.

[34] A. Madani, M. Moradi, A. Karargyris, and T. Syeda-Mahmood, "Semi-supervised learning with generative adversarial networks for chest X-ray classification with ability of data domain adaptation," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 1038–1042.

[35] Z. Ding and Y. Fu, "Deep transfer low-rank coding for cross–domain learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1768–1779, Jun. 2019.

[36] Z. Xu, W. Li, L. Niu, and D. Xu, "Exploiting low-rank structure from latent domains for domain generalization," in *Proc. ECCV*. Cham, Switzerland: Springer, 2014, pp. 628–643.

[37] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5543–5551.

[38] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," *J. Volcanol. Geothermal Res.*, vol. 173, no. 3, pp. 185–195, 2014.

[39] D. Geman, C. D'avignon, D. Q. Naiman, and R. L. Winslow, "Classifying gene expression profiles from pairwise mRNA comparisons," *Stat. Appl. Genet. Mol. Biol.*, vol. 3, no. 1, pp. 1–19, Jan. 2004.
[40] A. C. Tan, D. Q. Naiman, L. Xu, R. L. Winslow, and D. Geman, "Simple decision rules for classifying human cancers from gene expression profiles," *Bioinformatics*, vol. 21, no. 20, pp. 3896–3904, Oct. 2005.
[41] Y. Yap, X. Zhang, M. Ling, X. Wang, Y. Wong, and A. Danchin, "Classification between normal and tumor tissues based on the pair-wise gene expression ratio," *BMC Cancer*, vol. 4, no. 1, p. 72, Dec. 2004.
[42] I. N. Bronshtein and K. A. Semendyayev, *Handbook Mathematics*, 5th ed. Berlin, Germany: Springer, 2013.
[43] H. P. Sankappanavar and S. Burris, *A Course in Universal Algebra*. Berlin, Germany: Springer-Verlag, 2012.
[44] D. Lopez-Paz, R. Nishihara, S. Chintala, B. Scholkopf, and L. Bottou, "Discovering causal signals in images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017.
[45] K. Zhang, B. Schölkopf, P. Spirtes, and C. Glymour, "Learning causality and causality-related learning: Some recent progress," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 26–29, Jan. 2018.
[46] M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters, "Invariant models for causal transfer learning," *J. Mach. Learn. Res.*, to be published.
[47] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, 2016, vol. 2, no. 3, p. 7.
[48] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2686–2694.
[49] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, 1963.
[50] M. J. Kearns and U. V. Vazirani, *An Introduction to Computational Learning Theory*. Cambridge, MA, USA: MIT Press, 1994.
[51] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. ECCV*. Cham, Switzerland: Springer, 2016, pp. 597–613.
[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.
[53] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[54] (2016). *Geo Datasets*. [Online]. Available: https://www.ncbi.nlm.nih.gov/gds

Ruichu Cai (Member, IEEE) received the B.S. degree in applied mathematics and the Ph.D. degree in computer science from the South China University of Technology, Guangzhou, China, in 2005 and 2010, respectively.

He was a visiting student with the National University of Singapore, Singapore, from 2007 to 2009, and a Research Fellow at the Advanced Digital Sciences Center, Illinois at Singapore Pte, from 2013 to 2014. He is currently a Professor with the School of Computer, Guangdong University of Technology, Guangzhou. His research interests cover a variety of different topics, including causality, machine learning, and their applications.

Jiahao Li received the B.S. degree in network engineering and the M.S. degree in computer science and technology from the Guangdong University of Technology, Guangzhou, China, from 2016 to 2019.

His current research interests include transfer learning, deep learning, and their optimization theory.

Zhenjie Zhang received the B.S. degree from Fudan University, Shanghai, China, in 2004, and the Ph.D. degree from the National University of Singapore, Singapore, in 2010.

From 2010 to 2018, he worked as a Research Scientist and Senior Research Scientist at the Advanced Digital Sciences Center, University of Illinois, Champaign, IL, USA. He is currently the Singapore R&D Head of Yitu Technology.

Dr. Zhang is a recipient of the early-career Honourable Mention Award from the IEEE Technical Committee of Data Engineering (TCDE) in 2015, and the Best Paper Award from the IEEE International Conference on Cloud Engineering (IC2E) in 2013. He is the program committee Co-Chair for the International Workshop on Data Privacy (PrivData) in 2014, the program committee Co-Chair for the Asia-Pacific Web Conference (APWeb) in 2015, and the program committee member for top machine learning conferences, including ICML, NIPS, IJCAI, AAAI, and so on.

Xiaoyan Yang received the B.S. degree from the Department of Computer Science and Engineering, Fudan University, Shanghai, China, and the Ph.D. degree in computer science from the School of Computing, National University of Singapore, Singapore.

She was a Post-Doctoral Fellow at the Advanced Digital Sciences Center, Illinois at Singapore Pte. She is currently a Senior Data Scientist at Singapore R&D, Yitu Technology Pte Ltd.

Zhifeng Hao (Member, IEEE) received the B.Sc. degree in mathematics from Sun Yat-sen University, Guangzhou, China, in 1990, and the Ph.D. degree in mathematics from Nanjing University, Nanjing, China, in 1995.

He is currently a Professor with the School of Computer, Guangdong University of Technology, Guangzhou, and the School of Mathematics and Big Date, Foshan University, Foshan, China. His research interests involve various aspects of algebra, machine learning, data mining, and evolutionary algorithms.