



**HAL**  
open science

# Adversarial Multi-View Domain Adaptation for Regression

Mehdi Hennequin, Khalid Benabdeslem, Haytham Elghazel

► **To cite this version:**

Mehdi Hennequin, Khalid Benabdeslem, Haytham Elghazel. Adversarial Multi-View Domain Adaptation for Regression. 2022 International Joint Conference on Neural Networks (IJCNN), Jul 2022, Padoue, Italy. 10.1109/IJCNN55064.2022.9892148 . hal-04382606

**HAL Id: hal-04382606**

**<https://hal.science/hal-04382606v1>**

Submitted on 9 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Adversarial Multi-View Domain Adaptation for Regression

Mehdi Hennequin<sup>1,2</sup>

<sup>1</sup> Galilé Group, 28 Bd de la République,  
71100 Chalon-sur-Saône, France

<sup>2</sup> Université Lyon 1, LIRIS,  
UMR CNRS 5205, F-69622, France  
m.hennequin@fondation.galile.fr

Khalid Benabdeslem<sup>2</sup>

<sup>2</sup> Université Lyon 1, LIRIS,  
UMR CNRS 5205, F-69622, France  
khalid.benabdeslem@liris.cnrs.fr

Haytham Elghazel<sup>2</sup>

<sup>2</sup> Université Lyon 1, LIRIS,  
UMR CNRS 5205, F-69622, France  
haytham.elghazel@liris.cnrs.fr

**Abstract**—In this paper, we address the problem of domain adaptation in a regression setting, where only a few labeled samples are available in the source domain, and no labeled samples in the target domain. In addition, we consider that source data have different representations (multiple views). In this work, we investigate an original method to take advantage of different representations using a weighted combination of views. Besides, we use a co-training approach to include information from unlabeled instances by ensuring that models trained on different views make similar predictions. For this purpose, we introduce a novel formulation of the optimization objective for domain adaptation that relies on a discrepancy distance. Then, we develop an adversarial network domain adaptation algorithm adjusting weights given to each view, ensuring that those related to the target receive higher weights. Finally, we evaluate our method on different public datasets and compare it to other domain adaptation baselines to demonstrate the improvement for regression tasks.

**Index Terms**—Domain adaptation, Multi-view, Co-training, Regression

## I. INTRODUCTION

In most industrial problems, data are collected from various domains or devices, or are captured by different processes. Commonly, in industry, data are labeled during a test phase in which the sturdiness of the machine parts is tested. In this context, an algorithm trained for predictive maintenance for one specific machine cannot be generalize correctly to the same machine under different usage conditions. Therefore, it is common practice to retrain the predictive maintenance models. However, this retraining leads to delayed prognostics actions until enough data are available for accurate prediction. To address this issue, predictive models, trained with a specific machine, have to adapt to data with different data distributions and limited or non-existing fault information, *i.e.* different domains. In machine learning, this situation is often referred to as *domain adaptation* or *covariate shift* [1]. In general, domain adaptation methods attempt to solve the learning problem when the main learning task is the same but the domains have different feature spaces or different marginal conditional probabilities [2], [3], [4], [5].

On the other hand, data can be represented by several independent sets of features. For instance, in the example of the aforementioned predictive maintenance, data are collected

from diverse sensors and exhibit heterogeneous properties. Thus, data from different sensors can be naturally partitioned into independent groups [6]. Each group is referred to as a particular view. Multi-view learning [6], [7] aims to improve predictors by taking advantage of the redundancy and consistency between these multiple views.

In the domain adaptation context, views are generally concatenated into one single view to adapt to the learning task. However, this concatenation might cause negative transfer [8], (*i.e.* introduce source domain data/knowledge undesirably) because each view has a specific statistical property. This will result decrease in learning performance in the target domain. Nevertheless, we can find little research on multi-view domain adaptation [9], [10] where considerable attention has been given on the classification problem, while regression remains largely under-studied.

Furthermore, in this same context of predictive maintenance, data labeling is expensive and remains extremely time-consuming and/or cannot be taken for granted. Thus, the amount of labeled source data is very limited and it becomes difficult to training the models. In this case, co-training based approaches could be envisaged to exploit the information conveyed by both labeled and unlabeled data.

In this paper, we propose a novel co-training based approach to improve the alignment of the source domain and the target domain for multi-view domain adaptation in regression where a few data are labeled in the source domain. To the best of the authors' knowledge, no research was conducted for co-training multi-view domain adaptation in the regression context. This work makes two main contributions: first, we adapt a distance between distributions with multi-view, Co-Regression Discrepancy. The second main contribution is a new algorithm optimizing both representations and weights of each view for multi-view domain adaptation. We conduct experiments on real-world datasets and improve on state-of-the-art results for multi-view adversarial domain adaptation for regression.

## II. RELATED WORK

### A. Co-training

The co-training style approach is a technique largely used in multiple-view learning [6]. This method aims to enforce similar outcomes of multiple predictor functions by training them alternately to maximize their mutual agreement on unlabeled examples. The authors in [11] was the first to observe the relationship between the consensus of multiple hypotheses and their error rate for semi-supervised problems. The authors in [12] introduced the co-training algorithm for semi-supervised learning that greedily augments the training sets of two classifiers. Alternatively, a variant of the AdaBoost algorithm was suggested in [13] by boosting the agreement between two views on unlabeled data. Nevertheless, most studies consider classification problems, and the authors in [14] were the first to tackle regression problems by applying co-training with the  $kNN$  regressor. Instead of using two disjoint views, they use distinct distance measures for the two hypotheses. Zhou and Li [14] came up with a single-view variant of co-training that to some extent can be regarded as the bridge to co-regression. Inspired by the co-regularization algorithm [15], Brefeld, Gärtner, Scheffer and Wrobel [16] proposed a co-regression algorithm. Later, the authors in [17] extended this idea to kernel support vector regression and defined co-regularized support vector regression (CoSVR) as an  $\epsilon$ -insensitive version of co-regularization.

The related ideas of co-training were extended to domain adaptation by Chen, Weinberger and Blitzer [18]. The authors used the idea of co-training for semi-supervised domain adaptation (assuming a few target labeled samples are available) by finding a suitable split of the features into two sets based on the notion of  $\epsilon$ -expandability [19]. A related work [20] used the idea of co-regularization [15] for semi-supervised domain adaptation where they learn different classifiers for source and target, making their predictions agree on the unlabeled target samples. More recently, Saito, Ushiku and Harada [21] proposed asymmetric tri-training for unsupervised domain adaptation where one of the three models is learned only on pseudo-labeled target examples. Asymmetric tri-training, similar to [22], works with a single feature generator which feeds different classifiers. The authors in [23] followed the same idea, but also added a second generator and minimized the variance form of the Jensen-Shannon divergence between source and target feature distributions. Complementary to the agreement between classifiers, the authors encourage source feature distributions induced by the first generator and the second generator to be different from each other.

### B. Discrepancy minimization.

The present work is in line with discrepancy minimization methods, which were first introduced in [24], and further developed in [25], [26], [27], [28], [29]. Discrepancy is the key measure of the difference between two distributions in the

context of domain adaptation and has several advantages over other common divergence measures such as the  $l_1$  distance. Besides, several generalizations bound for adaptation in terms of discrepancy were proposed [25], [26], [27] [30], including pointwise guarantees in the case of kernel-based regularization, kernel ridge regression or support vector regression. Following on from the work on discrepancy, the authors in [31] proposed hypothesis-discrepancy where the bound is tighter. In our work, we use discrepancy distance, but we include the concept of co-regression multi-view learning to enforce this distance.

## III. CO-REGRESSION DISCREPANCY ADVERSARIAL NEURAL NETWORK METHOD

In this section, we present the details of our proposed method. First, we introduce the notation. Second, we give the overall idea of our method, as well as explaining the loss function we used in experiments. Finally, we explain the entire training procedure of our method.

### A. Notation

This section introduces the definitions and concepts needed for the following sections. Given  $\mathcal{X} \in \mathbb{R}^p$  and  $\mathcal{Y} \in \mathbb{R}$ , in the scenario of *multi-view domain adaptation regression setting* that we consider, the data instances from source domain  $X_s$  can be represented in  $M$  different views, where each view is separated into labeled and unlabeled parts. More formally, for  $v \in \{1, \dots, M\}$ ,  $X_s = \{X_v^{(l)}, X_v^{(z)}\}$  where  $X_v^{(l)} = \{(x_{v,1}^{(l)}, y_{v,1}), \dots, (x_{v,m_l}^{(l)}, y_{v,m_l})\} \in (\mathcal{X}, \mathcal{Y})^{m_l}$  of size  $m_l$ . In the same way, we consider an unlabeled source sample of size  $m_z$  such as  $X_v^{(z)} = \{x_{v,1}^{(z)}, \dots, x_{v,m_z}^{(z)}\} \in \mathcal{X}^{m_z}$ . In this setting, the sample size  $m_l$  of  $X_v^{(l)}$  is typically much smaller than  $m_z$  ( $m_l \ll m_z$ ). Similarly, we define unlabeled samples from the target domain,  $\mathcal{T} = X_t^{(z)} = (x_{t,1}^{(z)}, \dots, x_{t,n}^{(z)}) \in \mathcal{X}^n$  of size  $n$ . Unlike the source domain, we have no prior knowledge about views in the target domain.

### B. Overall Idea

The goal of Domain Adaptation is to minimize the target risk  $\mathcal{L}(h, f_t) = \mathbb{E}_{x \sim X_t} [L(h(x), f_t(x))]$ , where  $f_t : \mathcal{X} \rightarrow \mathcal{Y}$  is the true labeling function of the target domain,  $L: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$  a loss function, and a hypothesis class  $\mathcal{H}$  of hypotheses  $h: \mathcal{X} \rightarrow \mathcal{Y}$ . In this way, we aim to align the source and target features, in other words, we want to discover a new representation of the input features where source and target instances cannot be distinguished by any domain predictor. The question is how to find this subspace. We propose to utilize the disagreement of the multiple predictors. Consider  $M$  predictors  $h_v \in \mathcal{H}_v$ , for  $v \in \{1 \dots M\}$  (for reasons of simplification we abbreviate  $h_1 \in \mathcal{H}_1, \dots, h_M \in \mathcal{H}_M$  with  $h_v \in \mathcal{H}_v$ ), we train the  $v^{th}$  predictor on the  $v^{th}$  view source samples. This step is feasible because we have access to a few labeled source samples  $X_v^{(l)}$ . This training phase will allow us to predict the labels of unlabeled samples on both domains with an error for each predictor. Our task is to discover a subspace that minimizes those errors by maximizing their

mutual agreement. Our aim is for models trained on different views to make similar predictions. Therefore, we train a feature generator  $\phi$ , which takes as inputs  $X_s$  or  $X_t^{(z)}$ , and where the predictor networks take features from the generator. The goal of the generator is to generate a subspace feature where the disagreement between predictors is small. Consequently, the generator tries to discover a consensus between the multiple predictors of both domains and multiple views. Thereby, we propose to train discriminators  $h_v$  to maximize their disagreement, before training the generator to fool discriminators by minimizing their disagreement.

### C. Discrepancy Loss

As mentioned above, we are seeking a consensus between predictors for which we need to define an error function which will allow us to measure this consensus, or in another way, measure their disagreement on their outputs. The  $L_p$  losses commonly used in regression, are defined by:  $L_p(y, y') = |y - y'|^p$ ,  $p > 1$ . These losses may be good candidates to measure the disagreement. In [17], [14], [16], we principally use  $L_p$  losses where  $p = 2$ , also mentioned like the squared loss  $\ell_2(y, y') = |y - y'|^2$ . Thus, we define the disagreement loss between the predictors:

**Definition 1.** For  $0 < v < M$ , any  $h_v \in \mathcal{H}_v, h_u \in \mathcal{H}_u$ :

$$d_{\mathcal{H}, L_2}(h_v, h_u; x) = \sum_{v=1}^{M-1} \sum_{x \in \mathcal{X}} \ell_2(h_v(x), h_u(x)) \quad (1)$$

Now we have defined a loss for the disagreement, we can define a distance to measure the difference between the disagreement on labeled instances and the disagreement on unlabeled instances. Inspired by [24], the discrepancy methods, coupled with disagreement loss can give a variant measure distance named Co-Regression Discrepancy, CoDisc:

**Definition 2.** For any  $h_v \in \mathcal{H}_v, h_u \in \mathcal{H}_u$ :

$$CoDisc_{H, L_2}(h_v, h_u; x^{(l)}, x^{(z)}) = \max_{h_v \in \mathcal{H}, h_u \in \mathcal{H}} |d_{\mathcal{H}, L}(h_v, h_u; x^{(l)}) - d_{\mathcal{H}, L}(h_v, h_u; x^{(z)})| \quad (2)$$

### D. Adversarial algorithm for optimization

To sum up the earlier discussion, we need to train multiple predictors, which take inputs from the generator and maximize CoDisc, and the generator which tries to mimic the predictors. We will show how this can be achieved. Similarly to most other adversarial methods, we sequentially optimize different parameters of our networks according to different objectives. At a given iteration, losses are minimized/maximized sequentially:

**Step 1:** First, we train the predictors and generator on labeled source data with the different views. Our aim is for the  $v^{th}$  predictor to predict correctly the  $v^{th}$  view, for  $v \in \{1, \dots, M\}$ :

$$\min_{h_v \in \mathcal{H}_v, \phi_\theta} \mathcal{L}(X_v^{(l)}, Y_v). \quad (3)$$

$$\mathcal{L}(X_v^{(l)}, Y_v) = \sum_{i=1}^{m_l} \ell_2(h_v \circ \phi(x_{v,i}^{(l)}), y_{v,i}) \quad (4)$$

**Step 2:** In this second step, our aim is to maximize the mutual agreement between the predictors on source unlabeled instances and find a subspace between different views. Thus, we train the predictors  $h_v$  as a discriminator for a fixed generator. We increase the CoDisc loss between  $X_v^{(l)}$  and  $X_v^{(z)}$ . Then, we train the generator to minimize the discrepancy loss for fixed predictors:

$$\min_{\phi} \max_{\substack{h_v \in \mathcal{H}_v, \\ h_u \in \mathcal{H}_u}} CoDisc_{H, L_2}(h_u, h_v; X_v^{(l)}, X_v^{(z)}). \quad (5)$$

$$CoDisc_{H, L_2}(h_u, h_v; X_v^{(l)}, X_v^{(z)}) = \left| \sum_{v=1}^{M-1} \sum_{i=1}^{m_l} \ell_2(h_v \circ \phi(x_{v,i}^{(l)}), h_u \circ \phi(x_{u,i}^{(l)})) - \sum_{v=1}^{M-1} \sum_{i=1}^{m_z} \ell_2(h_v \circ \phi(x_{v,i}^{(z)}), h_u \circ \phi(x_{u,i}^{(z)})) \right|. \quad (6)$$

**Step 3:** In this third step, our aim is to maximize the mutual agreement between the predictors on target unlabeled instances and to align feature representation between target and source domain. Thus, we train the predictors  $h_v$  as a discriminator for fixed generators. We increase the CoDisc loss between  $X_v^{(l)}, X_t^{(z)}$  and  $X_v^{(z)}, X_t^{(z)}$ . We add a trade-off parameter  $\beta$  to moderate the adaptation between  $X_v^{(z)}$  and  $X_t^{(z)}$ . During the adaptation between target domain and source domain, we note that the generator struggles to reach a consensus between all predictors. We will thus weight each predictor according to its contribution with the parameter  $\alpha$ . We will see in the next step how to train that parameter. Then, we train the generator to minimize the discrepancy loss for fixed predictors:

$$\min_{\phi} \max_{\substack{h_v \in \mathcal{H}, \\ h_u \in \mathcal{H}}} CoDisc_{H, L_2}(h_u, \alpha_v h_v; x_v^{(l)}, x_t^{(z)}) - \beta CoDisc_{H, L_2}(h_u, \alpha_v h_v; x_v^{(z)}, x_t^{(z)}). \quad (7)$$

$$CoDisc_{H, L_2}(h_u, \alpha_v h_v; x_v^{(l)}, x_t^{(z)}) = \left| \sum_{v=1}^{M-1} \sum_{i=1}^{m_l} \alpha_v \ell_2(h_v \circ \phi(x_{v,i}^{(l)}), h_u \circ \phi(x_{u,i}^{(l)})) - \sum_{v=1}^{M-1} \sum_{i=1}^n \ell_2(h_v \circ \phi(x_{t,i}^{(z)}), h_u \circ \phi(x_{t,i}^{(z)})) \right| \quad (8)$$

$$CoDisc_{H, L_2}(h_u, \alpha_v h_v; x_v^{(z)}, x_t^{(z)}) = \left| \sum_{v=1}^{M-1} \sum_{i=1}^{m_z} \alpha_v \ell_2(h_v \circ \phi(x_{v,i}^{(z)}), h_u \circ \phi(x_{u,i}^{(z)})) - \sum_{v=1}^{M-1} \sum_{i=1}^n \ell_2(h_v \circ \phi(x_{t,i}^{(z)}), h_u \circ \phi(x_{t,i}^{(z)})) \right| \quad (9)$$

**Step 4:** In this step we train the parameter  $\alpha$  to select the best predictor:

$$\min_{\|\alpha\|_1=1} \mathcal{L}(X_v^{(l)}, X_t, \alpha_v). \quad (10)$$

$$\mathcal{L}(X_v^{(l)}, X_t, \alpha_v) = |\alpha_v \ell_2(h_v \circ \phi(x_v^{(l)}), h_u \circ \phi(x_u^{(l)})) - \ell_2(h_v \circ \phi(x_t), h_u \circ \phi(x_t))| + \lambda \|\alpha\|_2. \quad (11)$$

#### IV. EXPERIMENTS

In this section, we evaluate our CoDisc method. It should be noted that unsupervised Domain Adaptation with multi-view for regression is hard to evaluate as we have no real public database that corresponds entirely to the problem we described in the introduction. Consequently, we build scenarios and, for each one, we will describe the protocol.

In this section, we report the results of the CoDisc algorithm compared to other domain adaptation methods. The experiments are conducted on two public datasets. The following competitors are selected to compare the performance of the CoDisc algorithm:

- Weighting Adversarial Neural Network (WANN) [32] is a semi-supervised domain adaptation method based on the empirical  $\mathcal{Y}$ -discrepancy [26].
- Discriminative Adversarial Neural Network (DANN) [33] is a unsupervised domain adaptation method based on the  $\mathcal{H}$ -divergence. It is used here for regression tasks by considering the mean squared error as task loss instead of the binary cross-entropy proposed in the original algorithm.
- Adversarial Discriminative Domain Adaptation (ADDA) [34] performs a DANN algorithm in two-stage: it first learns a source encoder and a task hypothesis using labeled data and then learns the target encoder with adversarial training.
- Deep Correlation Alignment (Deep-CORAL) [35] is unsupervised domain adaptation method that aligns the second-order statistics of the source and target distributions with a linear transformation.
- Margin Disparity Discrepancy (MDD) [36] is a unsupervised domain adaptation, it learns a new feature representation by minimizing the disparity discrepancy.
- TrAdaBoostR2 [37] is based on a reverse-boosting principle where the weight of source instances poorly predicted are decreased at each boosting iteration. A 5 fold cross-validation is performed at each first stage and the best hypothesis is returned.
- Kullback-Leibler Importance Estimation Procedure (KLIIEP) [38] is a sample bias correction method minimizing the KL-divergence between a reweighted source and target distributions.
- Kernel Mean Matching (KMM) [39] reweights source instances in order to minimize the MMD between domains.
- Multiple Source Domain Adaptation (MDANN) [40] is a multi-source domain adaptation based on the  $d_{\mathcal{H}}$  distance

proposed in [22]. It gives high weights to sources far from the target (we used the soft version).

- Adversarial Hypothesis-Discrepancy Multi-Source Domain Adaptation (AHD-MSDA) [31] is a multi-source unsupervised domain adaptation based on the hypothesis-discrepancy.

#### A. Experiments on real datasets

1) *Dataset Superconductivity:* We propose here to demonstrate the efficiency of CoDisc on the UCI dataset superconductivity [41], [42]. The goal is to predict the critical temperature of superconductors. This is a common regression problem in industry, as industrialists are particularly interested in modeling the relationship between a material and its properties. The dataset contains two views: the first view contains 81 features extracted from 21263 superconductors, while the second view contains the chemical formula broken up for all the 21263 superconductors, whose format is binary. We divide this dataset into separate domains as per the setup of [37]. We select an input feature with a moderate correlation factor with the output (0.3). We then sort the set according to this feature and split it into four parts: low (l), middle-low (ml), middle-high (mh), high (h). Each part defines a domain with around 5000 instances. We conduct an experiment for each pair of domains which leads to 12 experiments. For each pair of domains we also randomly select different features from the two views. Therefore, the source domain and the target domain do not have the same features. We conduct the experiments for 10% and 15% source labeled instances. 10 target labeled instances are used in the training except for our method CoDisc, AHD-MSDA and MDAN, which benefit multi-view/multi-source learning method. The other target data are used to compute the results. For the multi-source methods such as AHD-MSDA and MDAN, we consider a view to be a source, while for the other baseline methods that do not consider multi-source learning, we merge the views. We reported the results in tables I, II, We also report the average MSE over the 12 experiments. In the second experiments we fix the size of source labeled instances at 5% and a pair of domains. Then, we gradually augment the target labeled instances used in training. The results are exported to table III. The presented results of this section are computed on a (2.7 GHz, 16 G RAM) computer. For all base line methods implementation except AHD-MSDA, the python library ADAPT is used [43]. The optimization parameters used in the presented experiments for baseline methods are  $\text{lr} = 0.001$ , and the loss function is the mean squared error (MSE). The base hypothesis used to learn the task is a neural network with two hidden fully-connected layers of 100 neurons each, ReLU activation functions, weights clipping  $C = 1$  and Adam optimizer; 200 epochs with a batch size of 128 are performed. Cross-validation is also applied to select best parameters and best scores for each baseline method. Our method and AHD-MSDA have been implemented using the Pytorch library, and the network architecture is:

- Feature extractor:
  - Linear(size(input features),50,ReLU)

TABLE I  
SUPERCONDUCTIVITY EXPERIMENTS MSE (10% SOURCE LABELED INSTANCES)

Expe.	l → ml	l → mh	l → h	ml → l	ml → mh	ml → h	mh → l
<b>CoDisc</b>	0.0938	<b>0.0337</b>	<b>0.0301</b>	<b>0.0215</b>	<b>0.0369</b>	<b>0.0318</b>	<b>0.0192</b>
WANN	0.0914	0.0564	0.0636	0.0343	0.0577	0.0495	0.0344
DANN	0.0738	0.0484	0.0400	0.0417	0.0693	0.1391	0.0417
ADDA	0.1046	0.0346	0.0307	0.0484	0.1004	0.0654	0.0729
<b>Deep-CO.</b>	0.0717	0.0379	0.0310	0.0438	0.0454	0.0435	0.0355
MDD	0.0718	0.0656	0.0443	0.0623	0.0507	0.0320	0.0572
TrAda.	0.0946	0.0420	0.0572	0.0577	0.0492	0.0290	0.0417
KLIEP	0.0764	0.0736	0.0312	0.0561	0.0400	0.0555	0.0324
KMM	<b>0.0710</b>	0.0399	0.0768	0.0269	0.0384	0.0286	0.0503
MDANN	0.0783	0.0464	0.0326	0.0408	0.0638	0.0984	0.0398
<b>AHD-MSDA</b>	0.1053	0.0537	0.0416	0.0415	0.0505	0.0418	0.0392
Expe.	mh → ml	mh → h	h → l	h → ml	h → mh	Avg MSE	
<b>CoDisc</b>	<b>0.0414</b>	<b>0.0206</b>	0.0236	<b>0.0561</b>	<b>0.0314</b>	<b>0.0367</b>	
WANN	0.0717	0.0354	0.0542	0.0758	0.0433	0.0556	
DANN	0.0625	0.0611	0.0237	0.1190	0.0554	0.0651	
ADDA	0.0576	0.0302	0.0255	0.0662	0.0320	0.0557	
<b>Deep-CO.</b>	0.0507	0.0308	<b>0.0247</b>	0.0730	0.0510	0.0449	
MDD	0.0742	0.0807	0.0456	0.0787	0.0530	0.0597	
TrAda.	0.0816	0.0660	0.0336	0.0795	0.0524	0.0570	
KLIEP	0.0717	0.0354	0.0542	0.0758	0.0433	0.0556	
KMM	0.0748	0.0686	0.0480	0.0889	0.0406	0.0544	
MDANN	0.0525	0.0540	0.0236	0.0906	0.0454	0.0569	
<b>AHD-MSDA</b>	0.0450	0.0305	0.0335	0.0660	0.0414	0.0433	

TABLE II  
SUPERCONDUCTIVITY EXPERIMENTS MSE (15% SOURCE LABELED INSTANCES)

Expe.	l → ml	l → mh	l → h	ml → l	ml → mh	ml → h	mh → l
<b>CoDisc</b>	0.1218	0.0368	<b>0.0221</b>	<b>0.0208</b>	<b>0.0323</b>	<b>0.0212</b>	<b>0.0246</b>
WANN	0.0734	0.0489	0.0446	0.0443	0.0501	0.0537	0.0279
DANN	0.0849	0.0597	0.0269	0.0491	0.0568	0.0678	0.0456
ADDA	0.2755	0.0388	0.0256	0.0360	0.1940	0.0466	0.0552
<b>Deep-CO.</b>	0.0756	<b>0.0331</b>	0.0306	0.0649	0.0417	0.0446	0.0406
MDD	<b>0.0696</b>	0.0609	0.0456	0.0569	0.0513	0.0382	0.0454
TrAda.	0.0902	0.0414	0.0562	0.0623	0.0504	0.0288	0.0434
KLIEP	0.0705	0.0422	0.0409	0.0463	0.0482	0.0396	0.0500
KMM	0.0744	0.0398	0.0724	0.0247	0.0387	0.0279	0.0509
MDANN	0.0819	0.0517	0.0259	0.0411	0.0518	0.0618	0.0426
<b>AHD-MSDA</b>	0.0918	0.0388	0.0321	0.0308	0.0423	0.0412	0.0346
Expe.	mh → ml	mh → h	h → l	h → ml	h → mh	Avg MSE	
<b>CoDisc</b>	0.0625	0.0371	<b>0.0250</b>	<b>0.0620</b>	0.0420	<b>0.0424</b>	
WANN	0.0939	0.0484	0.0464	0.0756	0.0513	0.0549	
DANN	0.0646	0.0681	0.0321	0.1346	0.0450	0.0613	
ADDA	0.5448	0.0536	1.4011	0.3490	0.0341	0.2545	
<b>Deep-CO.</b>	<b>0.0464</b>	<b>0.0305</b>	0.0256	0.0725	<b>0.0332</b>	0.0449	
MDD	0.0725	0.0711	0.0471	0.0768	0.0501	0.0571	
TrAda.	0.0825	0.0666	0.0313	0.0781	0.0522	0.0570	
KLIEP	0.0991	0.0317	0.0438	0.0845	0.0732	0.0558	
KMM	0.0715	0.0630	0.0507	0.0857	0.0401	0.0533	
MDANN	0.0625	0.0540	0.0260	0.0800	0.0444	0.0555	
<b>AHD-MSDA</b>	0.0522	0.0405	0.0335	0.0660	0.0414	0.454	

- Linear(50,25,ReLU)
- Linear(25,10,ReLU)
- Linear(10,25,ReLU)
- Linear(25,50,ReLU)

- Predictors/Discriminators:

- Linear(50,15)
- Linear(15,1)

We use ADAM optimizer with lr = 0,0001 for CoDisc and lr = 0.001 for AHD-MSDA. The batch size is set to 128 with 400 epochs for CoDisc and 200 epochs for AHD-MSDA.

2) *Dataset Ligand prediction*: In the second scenario we use the dataset Ligand prediction from [17]. Ligand affinity prediction is an important learning task in the chemoinformatics sector since many drugs act as protein ligands. The aim of affinity prediction is to determine binding affinities for small molecular compounds with respect to a bigger protein using computational methods. Besides a few labeled protein-ligand pairs, millions of small compounds are gathered in molecular databases as ligand candidates. Many different data representations or views exist that can be used for learning.

TABLE III  
SUPERCONDUCTIVITY EXPERIMENTS MSE (5% SOURCE LABELED INSTANCES), PAIR DOMAINS:  $h \rightarrow mh$

Number of target labeled	10	20	30	40	50	60	70	80	90	100
<b>CoDisc</b>	<b>0.0216</b>	<b>0.0216</b>	<b>0.0216</b>	<b>0.0216</b>	<b>0.0216</b>	<b>0.0216</b>	<b>0.0216</b>	<b>0.0216</b>	<b>0.0216</b>	0.0216
<b>WANN</b>	0.0793	0.0740	0.0718	0.0656	0.0604	0.0565	0.0486	0.0326	0.0233	<b>0.0208</b>
<b>DANN</b>	0.0766	0.0764	0.0715	0.0689	0.0674	0.0669	0.0639	0.0634	0.0626	0.0616
<b>ADDA</b>	0.4925	0.4915	0.4906	0.4856	0.4833	0.4821	0.4807	0.4679	0.4569	0.4548
<b>Deep-CO.</b>	0.0673	0.0651	0.0634	0.0593	0.0571	0.0541	0.0532	0.0469	0.0406	0.0377
<b>MDD</b>	0.0796	0.0768	0.0748	0.0698	0.0779	0.0628	0.0602	0.0538	0.0530	0.0508
<b>TrAda.</b>	0.0707	0.0667	0.0642	0.0596	0.0577	0.0563	0.0524	0.0448	0.0389	0.0364
<b>KLIEP</b>	0.0937	0.0913	0.0860	0.0780	0.0769	0.0763	0.0759	0.0735	0.0720	0.0682
<b>KMM</b>	0.0818	0.0801	0.0772	0.0772	0.0698	0.0675	0.0649	0.0626	0.0581	0.0561
<b>MDANN</b>	0.0668	0.0668	0.0668	0.0668	0.0668	0.0668	0.0668	0.0668	0.0668	0.0668
<b>AHD-MSDA</b>	0.0459	0.0459	0.0459	0.0459	0.0459	0.0459	0.0459	0.0459	0.0459	0.0459

TABLE IV  
LIGAND PREDICTION EXPERIMENTS MSE (10% SOURCE LABELED INSTANCES)

Expe.	l ml	l mh	l h	ml l	ml mh	ml h	mh l
<b>CoDisc</b>	0.0273	<b>0.0223</b>	<b>0.0217</b>	0.0466	<b>0.0349</b>	<b>0.0365</b>	<b>0.0240</b>
<b>WANN</b>	0.0419	0.0417	0.0447	0.0738	0.0798	0.0821	0.0539
<b>DANN</b>	0.0390	0.0396	0.0355	0.0745	0.0751	0.0745	0.0485
<b>ADDA</b>	0.0231	0.0264	0.0250	0.0534	0.2585	0.0626	0.0753
<b>Deep-CO.</b>	<b>0.0229</b>	0.0249	0.0262	0.0461	0.0461	0.0521	0.0330
<b>MDD</b>	0.0402	0.0385	0.0405	0.0745	0.0768	0.0731	0.0550
<b>TrAda.</b>	0.0422	0.0420	0.0422	0.0761	0.0792	0.0786	0.0555
<b>KLIEP</b>	0.0414	0.0398	0.0410	0.0767	0.0753	0.0739	0.0498
<b>KMM</b>	0.0382	0.0444	0.0334	0.0661	0.0709	0.0646	0.0422
<b>MDANN</b>	0.0819	0.0517	0.0259	<b>0.0411</b>	0.0518	0.0618	0.0426
<b>AHD-MSDA</b>	0.0373	0.0322	0.0317	0.0570	0.0459	0.0465	0.0340
Expe.	mh ml	mh h	h l	h ml	h mh	Avg MSE	
<b>CoDisc</b>	<b>0.0231</b>	<b>0.0243</b>	<b>0.0183</b>	<b>0.0180</b>	<b>0.0209</b>	<b>0.0265</b>	
<b>WANN</b>	0.0533	0.0537	0.0378	0.0384	0.0404	0.0535	
<b>DANN</b>	0.0517	0.0491	0.0377	0.0390	0.0432	0.0506	
<b>ADDA</b>	0.0944	0.0345	0.2749	0.0343	0.0762	0.0866	
<b>Deep-CO.</b>	0.0316	0.0321	0.0266	0.0236	0.0244	0.0325	
<b>MDD</b>	0.0529	0.0540	0.0399	0.0428	0.0380	0.0522	
<b>TrAda.</b>	0.0536	0.0565	0.0444	0.0422	0.0449	0.0548	
<b>KLIEP</b>	0.0490	0.0551	0.0429	0.0413	0.0415	0.0523	
<b>KMM</b>	0.0360	0.0761	0.0315	0.0334	0.0275	0.0470	
<b>MDANN</b>	0.0517	0.0451	0.0337	0.0360	0.0392	0.0476	
<b>AHD-MSDA</b>	0.0331	0.0443	0.0383	0.0380	0.0409	0.0399	

Affinity prediction and other applications suffer from scant label information and the need to choose the most appropriate view for learning. The strength of a protein-compound binding interaction is characterized by the real valued binding affinity. Our aim is to predict binding strength of binding using regression techniques from machine learning. In this section we evaluate the performance of CoDisc for predicting the affinity values of small compounds. Our experiments are performed on 24 datasets consisting of ligands and their affinity to one particular human protein per dataset, gathered from BindingDB [44]. Every ligand is a single molecule in the sense of a connected graph, and all ligands are available in the standard molecular fingerprint formats ECFP4, GpiDAPH3, and Maccs. All three formats are binary and high-dimensional, and each format is considered to be a view. To create domains we follow the same method described in the superconductivity experiments and we also repeat the same experiments with the same parameters optimisation. The results are exported in tables IV, V and VI.

3) *Discussion:* Overall, we find that our method performs better than baseline methods in the target domain with 10% and 15% source labeled instances. However, with 15% source labeled instances, our method performs as well as or less well than some baseline methods such as Deep-CORAL for a pair of domains. The reason for this is that some baseline methods leverage information from a few target labeled instances during training, thus penalizing our performance. We can make the same observation with table III, VI, thus, the more we increase the number of target labeled instances, the less efficient we are. However, our approach outperforms, for the different settings, the baseline methods which not benefit from the information from target domain such as AHD-MSDA, DANN (considered as our main competitor). As far as the ligand prediction dataset considered, we observe similar result as for the superconductivity dataset. Our method performs well for 10% and 15% source labeled instances and decrease when the baseline method receive more target labeled instances.

TABLE V  
LIGAND PREDICTION EXPERIMENTS MSE (15% SOURCE LABELED INSTANCES)

Expe.	l ml	l mh	l h	ml l	ml mh	ml h	mh l
<b>CoDisc</b>	<b>0.0159</b>	<b>0.0203</b>	<b>0.0230</b>	<b>0.0402</b>	<b>0.0352</b>	<b>0.0324</b>	<b>0.0258</b>
WANN	0.0427	0.0406	0.0436	0.0747	0.0739	0.0740	0.0495
DANN	0.0376	0.0408	0.0403	0.0714	0.0745	0.0753	0.0539
ADDA	0.0395	0.0375	0.0332	0.2063	0.1357	0.1051	0.0564
<b>Deep-CO.</b>	0.0233	0.0243	0.0274	0.0467	0.0454	0.0498	0.0325
MDD	0.0413	0.0418	0.0414	0.0787	0.0779	0.0818	0.0610
TrAda.	0.0404	0.0397	0.0401	0.0787	0.0771	0.0788	0.0524
KLIEP	0.0417	0.0414	0.0435	0.0758	0.0745	0.0765	0.0524
KMM	0.0401	0.0253	0.0324	0.0875	0.0659	0.0692	0.0522
MDANN	0.0306	0.0348	0.0303	0.0614	0.0645	0.0653	0.0539
<b>AHD-MSDA</b>	0.0259	0.0303	0.0330	0.0502	0.0452	0.0424	0.0458
Expe.	mh ml	mh h	h l	h ml	h mh	Avg MSE	
<b>CoDisc</b>	<b>0.0206</b>	0.0391	<b>0.0181</b>	<b>0.0202</b>	<b>0.0177</b>	<b>0.0257</b>	
WANN	0.0502	0.0481	0.0448	0.0433	0.0404	0.0522	
DANN	0.0536	0.0511	0.0408	0.0458	0.0424	0.0523	
ADDA	0.0384	0.0323	0.0279	0.0433	0.0240	0.0650	
<b>Deep-CO.</b>	0.0317	<b>0.0313</b>	0.0254	0.0239	0.0253	0.0322	
MDD	0.0541	0.0574	0.0423	0.0476	0.0446	0.0558	
TrAda.	0.0503	0.0520	0.0459	0.0451	0.0460	0.0539	
KLIEP	0.0510	0.0507	0.0454	0.0454	0.0427	0.0534	
KMM	0.0749	0.0338	0.0327	0.0437	0.0255	0.0486	
MDANN	0.0436	0.0411	0.0308	0.0358	0.0414	0.0444	
<b>AHD-MSDA</b>	0.0406	0.0491	0.0381	0.0402	0.0377	0.0399	

TABLE VI  
LIGAND PREDICTION EXPERIMENTS MSE (5% SOURCE LABELED INSTANCES), PAIR DOMAINS:  $h \rightarrow mh$

Number of target labeled	10	20	30	40	50	60	70	80	90	100
<b>CoDisc</b>	<b>0.0187</b>	<b>0.0187</b>	<b>0.0187</b>	<b>0.0187</b>	<b>0.0187</b>	<b>0.0187</b>	<b>0.0187</b>	<b>0.0187</b>	<b>0.0187</b>	0.0187
WANN	0.0492	0.0409	0.0391	0.0322	0.0321	0.0277	0.0227	0.0189	<b>0.0187</b>	<b>0.0169</b>
DANN	0.0505	0.0495	0.0424	0.0412	0.0395	0.0365	0.0352	0.0300	0.0249	0.0222
ADDA	0.0626	0.0623	0.0618	0.0601	0.0592	0.0583	0.0532	0.0509	0.0493	0.0448
<b>Deep-CO.</b>	0.0437	0.0426	0.0370	0.0352	0.0333	0.0281	0.0237	0.0221	0.0209	0.0177
MDD	0.0519	0.0506	0.0498	0.0473	0.0441	0.0439	0.0424	0.0388	0.0362	0.0344
TrAda.	0.0497	0.0427	0.0416	0.0375	0.0348	0.0328	0.0269	0.0245	0.0224	0.0211
KLIEP	0.0599	0.0595	0.0586	0.0579	0.0556	0.0549	0.0525	0.0515	0.0513	0.0492
KMM	0.0495	0.0487	0.0478	0.0464	0.0445	0.0439	0.0426	0.0392	0.0381	0.0361
MDANN	0.0449	0.0449	0.0449	0.0449	0.0449	0.0449	0.0449	0.0449	0.0449	0.0449
<b>AHD-MSDA</b>	0.0405	0.405	0.405	0.405	0.405	0.405	0.405	0.405	0.405	0.405

## V. CONCLUSION

In this work, we proposed an adversarial domain adaptation algorithm based on a new discrepancy, CoDisc, tailored for multi-view regression where a few data are labeled in the source domain.

We demonstrated the efficiency of our method in real datasets especially with a few source labeled instances. Nevertheless, the main problem we face in our work stems from the aggregation of multiple predictors. To control the synchronization of the generator, we must choose the appropriate learning rate of the discriminator for the context, as well as empirically adapt the parameter  $\beta$ . In our future work, we intend to investigate the self-supervised learning and active learning settings, to try to label target data with a high degree of confidence.

## REFERENCES

- [1] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, Oct. 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/B6V0M-4136355-5/1/6432c256e0be03b1503bbf79e4e91d1a>
- [2] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [3] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, Jul. 2020. [Online]. Available: <https://doi.org/10.1145/3400066>
- [4] I. Redko, E. Morvant, A. Habrard, M. Sebban, and Y. Bennani, "A survey on domain adaptation theory," *CoRR*, vol. abs/2004.11829, 2020. [Online]. Available: <https://arxiv.org/abs/2004.11829>
- [5] L. Zhong, Z. Fang, F. Liu, B. Yuan, G. Zhang, and J. Lu, "Bridging the theoretical bound and deep algorithms for open set domain adaptation," *CoRR*, vol. abs/2006.13022, 2020.
- [6] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *CoRR*, vol. abs/1304.5634, 2013. [Online]. Available: <http://arxiv.org/abs/1304.5634>
- [7] Y. Li, M. Yang, and Z. Zhang, "Multi-view representation learning: A survey from shallow methods to deep methods," *CoRR*, vol. abs/1610.01206, 2016. [Online]. Available: <http://arxiv.org/abs/1610.01206>
- [8] W. Zhang, L. Deng, and D. Wu, "Overcoming negative transfer: A survey," *CoRR*, vol. abs/2009.00909, 2020. [Online]. Available: <https://arxiv.org/abs/2009.00909>
- [9] P. Yang, W. Gao, Q. Tan, and K.-F. Wong, "Information-theoretic multi-view domain adaptation," in *Proceedings of the 50th Annual Meeting of*



- the Association for Computational Linguistics (Volume 2: Short Papers)*. Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 270–274. [Online]. Available: <https://aclanthology.org/P12-2053>
- [10] Y. Xia, D. Yang, Z. Yu, F. Liu, J. Cai, L. Yu, Z. Zhu, D. Xu, A. L. Yuille, and H. R. Roth, “Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation,” *Medical image analysis*, vol. 65, p. 101766, 2020.
- [11] D. S. Virginia R., “Learning classification with unlabeled data,” in *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*. Morgan Kaufmann, 1993, pp. 112–119.
- [12] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, ser. COLT’98. New York, NY, USA: Association for Computing Machinery, 1998, p. 92–100.
- [13] M. Collins and Y. Singer, “Unsupervised models for named entity classification,” in *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP 1999, College Park, MD, USA, June 21-22, 1999*. Association for Computational Linguistics, 1999.
- [14] Z. Zhou and M. Li, “Semi-supervised regression with co-training,” in *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*. Professional Book Center, 2005, pp. 908–916.
- [15] V. Sindhwani, P. Niyogi, and M. Belkin, “A co-regularization approach to semi-supervised learning with multiple views,” 2005.
- [16] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel, “Efficient co-regularised least squares regression,” in *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, ser. ACM International Conference Proceeding Series, vol. 148. ACM, 2006, pp. 137–144.
- [17] K. Ullrich, M. Kamp, T. Gärtner, M. Vogt, and S. Wrobel, “Co-regularised support vector regression,” in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part II*, ser. Lecture Notes in Computer Science, vol. 10535. Springer, 2017, pp. 338–354.
- [18] M. Chen, K. Q. Weinberger, and J. Blitzer, “Co-training for domain adaptation,” in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., vol. 24. Curran Associates, Inc., 2011.
- [19] M.-f. Balcan, A. Blum, and K. Yang, “Co-training and expansion: Towards bridging theory and practice,” in *Advances in Neural Information Processing Systems*, L. Saul, Y. Weiss, and L. Bottou, Eds., vol. 17. MIT Press, 2005.
- [20] A. Kumar, A. Saha, and H. Daume, “Co-regularization based semi-supervised domain adaptation,” in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010.
- [21] K. Saito, Y. Ushiku, and T. Harada, “Asymmetric tri-training for unsupervised domain adaptation,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 2988–2997. [Online]. Available: <https://proceedings.mlr.press/v70/saito17a.html>
- [22] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, “Maximum classifier discrepancy for unsupervised domain adaptation,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 3723–3732.
- [23] A. Kumar, P. Sattigeri, K. Wadhawan, L. Karlinsky, R. S. Feris, B. Freeman, and G. W. Wornell, “Co-regularized alignment for unsupervised domain adaptation,” in *Advances in Neural Information Processing Systems 31: NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018, pp. 9367–9378.
- [24] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain adaptation: Learning bounds and algorithms,” in *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009.
- [25] C. Cortes and M. Mohri, “Domain adaptation in regression,” in *Algorithmic Learning Theory - 22nd International Conference, ALT 2011, Espoo, Finland, October 5-7, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6925. Springer, 2011, pp. 308–323.
- [26] M. Mohri and A. Muñoz Medina, “New analysis and algorithm for learning with drifting distributions,” in *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, ser. ALT’12. Berlin, Heidelberg: Springer-Verlag, 2012, p. 124–138.
- [27] C. Cortes and M. Mohri, “Domain adaptation and sample bias correction theory and algorithm for regression,” *Theoretical Computer Science*, vol. 519, 2014.
- [28] S. Kuroki, N. Charoenphakdee, H. Bao, J. Honda, I. Sato, and M. Sugiyama, “Unsupervised domain adaptation based on source-guided discrepancy,” 2018.
- [29] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, “Bridging theory and algorithm for domain adaptation,” *CoRR*, vol. abs/1904.05801, 2019. [Online]. Available: <http://arxiv.org/abs/1904.05801>
- [30] C. Cortes, M. Mohri, and A. M. Medina, “Adaptation based on generalized discrepancy,” *Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1–30, 2019. [Online]. Available: <http://jmlr.org/papers/v20/15-192.html>
- [31] G. Richard, A. de Mathelin, G. Hébrail, M. Mougeot, and N. Vayatis, “Unsupervised multi-source domain adaptation for regression,” in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 12457. Springer, 2020, pp. 395–411.
- [32] A. de Mathelin, G. Richard, M. Mougeot, and N. Vayatis, “Adversarial weighting for domain adaptation in regression,” *CoRR*, vol. abs/2006.08251, 2020. [Online]. Available: <https://arxiv.org/abs/2006.08251>
- [33] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, “Domain-adversarial training of neural networks,” in *J. Mach. Learn. Res.*, 2016.
- [34] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” *CoRR*, vol. abs/1702.05464, 2017. [Online]. Available: <http://arxiv.org/abs/1702.05464>
- [35] B. Sun and K. Saenko, “Deep coral: Correlation alignment for deep domain adaptation,” in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, pp. 443–450.
- [36] Y. Zhang, T. Liu, M. Long, and M. Jordan, “Bridging theory and algorithm for domain adaptation,” in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 7404–7413. [Online]. Available: <https://proceedings.mlr.press/v97/zhang19i.html>
- [37] D. Pardoe and P. Stone, “Boosting for regression transfer,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML’10. Madison, WI, USA: Omnipress, 2010, p. 863–870.
- [38] M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe, “Direct importance estimation with model selection and its application to covariate shift adaptation,” in *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20. Curran Associates, Inc., 2008.
- [39] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola, “Correcting sample selection bias by unlabeled data,” in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman, Eds., vol. 19. MIT Press, 2007.
- [40] H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J. Gordon, “Adversarial multiple source domain adaptation,” in *Advances in Neural Information Processing Systems 31: NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018, pp. 8568–8579.
- [41] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [42] K. Hamidieh, “A data-driven statistical model for predicting the critical temperature of a superconductor,” *Computational Materials Science*, vol. 154, pp. 346–354, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0927025618304877>
- [43] A. de Mathelin, F. Deheeger, G. Richard, M. Mougeot, and N. Vayatis, “ADAPT : Awesome domain adaptation python toolbox,” *CoRR*, vol. abs/2107.03049, 2021. [Online]. Available: <https://arxiv.org/abs/2107.03049>
- [44] M. K. Gilson, T. Liu, M. Baitaluk, G. Nicola, L. Hwang, and J. Chong, “BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology,” *Nucleic Acids Research*, vol. 44, no. D1, pp. D1045–D1053, 10 2015.