Bi-Classifier Determinacy Maximization for Unsupervised Domain Adaptation

Shuang Li,[1] Fangrui Lv,[1] Binhui Xie,[1] Chi Harold Liu,[1,*] Jian Liang,[2] Chen Qin[3]
[1]School of Computer Science and Technology, Beijing Institute of Technology, China
[2]Alibaba Group, China
[3] Institute for Digital Communications, School of Engineering, University of Edinburgh, United Kingdom
{shuangli,fangruilv,binhuixie,chiliu}@bit.edu.cn, xuelang.lj@alibaba-inc.com, Chen.Qin@ed.ac.uk

Abstract

Unsupervised domain adaptation challenges the problem of transferring knowledge from a well-labelled source domain to an unlabelled target domain. Recently, adversarial learning with bi-classifier has been proven effective in pushing cross-domain distributions close. Prior approaches typically leverage the disagreement between bi-classifier to learn transferable representations, however, they often neglect the classifier determinacy in the target domain, which could result in a lack of feature discriminability. In this paper, we present a simple yet effective method, namely *Bi-Classifier Determinacy Maximization* (BCDM), to tackle this problem. Motivated by the observation that target samples cannot always be separated distinctly by the decision boundary, here in the proposed BCDM, we design a novel classifier determinacy disparity (CDD) metric, which formulates classifier discrepancy as the class relevance of distinct target predictions and implicitly introduces constraint on the target feature discriminability. To this end, the BCDM can generate discriminative representations by encouraging target predictive outputs to be consistent and determined, meanwhile, preserve the diversity of predictions in an adversarial manner. Furthermore, the properties of CDD as well as the theoretical guarantees of BCDM's generalization bound are both elaborated. Extensive experiments show that BCDM compares favorably against the existing state-of-the-art domain adaptation methods.

Introduction

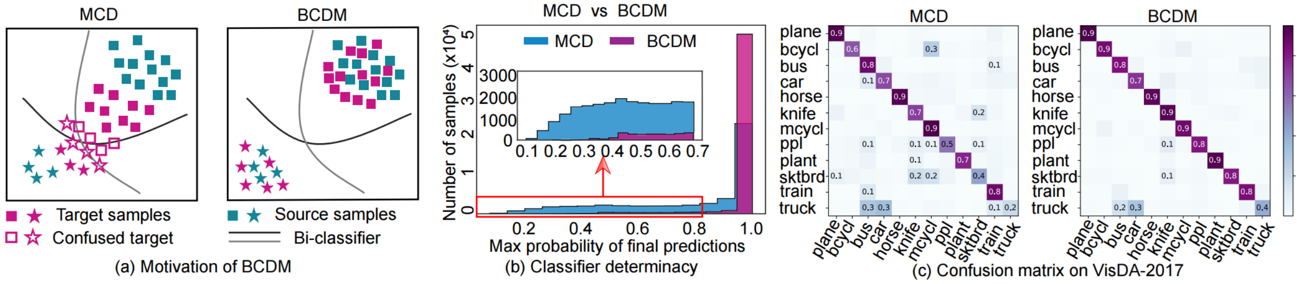
The significant progress on computer vision tasks has been dominated by deep neural networks (DNNs) trained on large-scale datasets, such as image classification (Krizhevsky, Sutskever, and Hinton 2012), semantic segmentation (Chen et al. 2018) and object detection (Ren et al. 2015). Despite their tremendous success, collecting sufficient labelled data in real-world applications is always onerous and extremely costly. In this regard, it is highly desirable to develop unsupervised domain adaptation (UDA) techniques (Pan and Yang 2010; Pan et al. 2011; Gong et al. 2012) to address the domain shift problem existing between unlabelled target domain and labelled source domain.

Latest advances of deep UDA methods, leveraging the extraordinary feature extraction ability of DNNs, have been continuously pushing forward the boundaries of UDA (Cui et al. 2020; Jin et al. 2020; Chen et al. 2019; Xu et al. 2019; Li et al. 2020b). To mitigate the domain shift, some UDA approaches aim at minimizing the well-defined distance metrics between domains (Tzeng et al. 2014; Long et al. 2015). Works along this line are based on the theoretical analysis in (Bendavid et al. 2010) that minimizes the divergence across source and target domains. On par with them, fruitful line of works utilizing generative adversarial nets (GANs) (Goodfellow et al. 2014) have also achieved remarkable performance by learning domain-invariant representations adversarially in a two player min-max game (Ganin and Lempitsky 2015; Saito et al. 2018b).

There exist two popular paradigms to conduct adversarial domain adaptation either by constructing a domain discriminator or by utilizing two distinct classifiers. However, the first category will bring side effects of the deterioration of feature discriminability, though the feature transferability is strengthened (Chen et al. 2019). As for the second paradigm, the selection of classifier discrepancy loss between two task-specific classifiers is critical for expected adaptability. The straightforward ℓ_1 distance between predictions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, i.e., $|\boldsymbol{p}_1 - \boldsymbol{p}_2|_1$, in Maximum Classifier Discrepancy (Saito et al. 2018b), merely measures the bi-classifier discrepancy at the corresponding class positions while ignoring the constraint of the relevance information between classes. Thus the certainty of predictions cannot be guaranteed, which may further hurt the learning behavior. For instance, there could be uncertain predictions such as values of $[0.34, 0.33, 0.33]$ and $[0.34, 0.33, 0.33]$ by minimizing the ℓ_1 distance between bi-classifier, which results in generating confused target features. Such situation can lead to ambiguous decisions and thus should be avoided. As illustrated in Fig. 1, due to the domain shift dilemma, numerous target samples are prone to be around the decision boundary. In other words, the classifier determinacy that can put confident labels on target samples, is lacking, thus impacting the feature discriminability.

To tackle the aforementioned issues, we propose a Bi-Classifier Determinacy Maximization (BCDM) method together with a novel metric called classifier determinacy dis-

[*]Corresponding author.



(a) Motivation of BCDM

(b) Classifier determinacy

(c) Confusion matrix on VisDA-2017

Figure 1: (a): Comparison of previous and the proposed bi-classifier matching methods. (b)(c): Empirically unveil that *higher classifier determinacy* indicates more discriminative representations on VisDA-2017 dataset. We first show the frequency of max probability of each target sample's prediction, where the MCD obtains more uncertain predictions with small values while our BCDM obtains more confident predictions (> 0.9). Similarly, the tendency that bi-classifier in MCD gives ambiguous predictions also results in fewer target samples on the diagonal in (c). It implicitly implies that scarcity of determinacy during bi-classifier adversarial training deteriorates the discriminability in the target domain.

parity (CDD). Specifically, CDD contains all the probabilities that the predictions of bi-classifier are inconsistent, which is defined as the summation of product between bi-classifier different class predictions. Regarding the training procedure, we first maximize the CDD loss which enforces bi-classifier to generate disagreement across classes, so as to explore diverse output spaces under the source supervision. And then we minimize the CDD loss to push the two domains together class-wisely by generating discriminative features that make the predictions consistent and determined. In this way, our method is able to delve into pushing the target samples away from the decision boundary. Consequently, the prediction consistency guarantees target samples to be close to the source regions class-wisely. The bi-classifier determinacy equips target features with categorical discriminability, which further improves the accuracy.

On top of the proposed CDD metric, the BCDM is able to simultaneously enhance the classifier determinacy and prediction diversity by adversarially optimizing the CDD loss. Our method has been shown to be simple yet effective. In addition, theoretical guarantees about BCDM in the aspect of generalization error bound on the target domain are elaborated. The proposed method is validated on various applications and scenarios including a toy dataset, four cross-domain image classification benchmark datasets, and a prevalent "synthetic-2-real" semantic segmentation set-up. Extensive experiments show that our method achieves significant improvements over state-of-the-art UDA methods.

Related Work

Existing deep domain adaptation methods can be mainly grouped into two categories: discrepancy optimization based methods (Long et al. 2015; Li et al. 2020b,a) and adversarial learning based methods (Ganin and Lempitsky 2015; Li et al. 2019; Tzeng et al. 2015). In the first group, DAN (Long et al. 2015) has proposed to reduce the gap between domains via minimizing a discrepancy metric named maximum mean discrepancy (MMD) (Gretton et al. 2007). Later, JAN (Long et al. 2017) utilizes modified versions of MMD to align joint distributions for more effective transfer.

Adversarial learning based UDA methods represent another line of works, which share the similar spirits with GANs (Goodfellow et al. 2014). They mainly focus on learning indistinguishable representations for data in both source and target domains in an adversarial manner. Specifically, one type of approaches exploits a domain discriminator to distinguish whether the input samples are from source or target domain, while a generator is trained to generate features that can confuse the discriminator (Ganin and Lempitsky 2015; Pei et al. 2018; Tsai et al. 2018). The other group of methods can utilize two distinct task-specific classifiers acting as a domain discriminator (Saito et al. 2018a,b; Lee et al. 2019; Luo et al. 2019; Lu et al. 2020).

Our proposed method falls into the adversarial learning category and inherits the similar bi-classifier training strategy. To name a few, MCD (Saito et al. 2018b) uses the ℓ_1 distance to calculate the classifier discrepancy which only considers the disagreement between the two classifiers with respect to the same class prediction. By comparison, the sliced Wasserstein distance in (Lee et al. 2019) is more powerful to capture the geometrically meaningful dissimilarity between two classifiers through optimal transport, however, this method not only increases the computation complexity but requires a sophisticated hyper-parameter tuning process. Very recently, STAR finds that more classifiers can achieve better adaptation performance and models infinite number of classifiers through reparameterization trick (Lu et al. 2020). CLAN (Luo et al. 2019) applies different adversarial weight to different pixels in semantic segmentation without explicitly incorporating the classes into the model.

Though these approaches have yielded promising results, they actually neglect the determinacy of the classifier, which may result in the decision boundary passing through high data density and misalignment among classes. To address this, here we propose a bi-classifier determinacy maximization (BCDM) algorithm with a novel metric, which formulates the classifier discrepancy as the relevance of different class predictions with theoretical justifications. Extensive experiments have demonstrated that our method can significantly outperform the competing baseline approaches.

Bi-Classifier Determinacy Maximization

Assume a labelled source domain $\mathcal{S} = \{\mathcal{X}_s, \mathcal{Y}_s\} = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s examples, where y_i^s is the corresponding label of $\boldsymbol{x}_i^s$, and an unlabelled target domain $\mathcal{T} = \{\mathcal{X}_t\} = \{\boldsymbol{x}_j^t\}_{j=1}^{n_t}$ with n_t samples, the goal of our work is to learn a deep neural network $\phi : \mathcal{X}_t \to \mathcal{Y}_t$ that can learn discriminative representations with determined predictions on the target domain. The model consists of a feature generator G and two distinct classifiers C_1 and C_2, and θ_g, θ_{c_1} and θ_{c_2} are the corresponding network parameters. To better understand, we show the overview of the BCDM in Fig. 2.

In this section, we first introduce a novel classifier determinacy disparity (CDD) to measure the classifier discrepancy. Then, we report our bi-classifier determinacy maximization (BCDM) and investigate its theoretical guarantees.

Classifier Determinacy Disparity

To measure the classifier discrepancy, we first propose a classifier determinacy disparity (CDD) metric. Given $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ as the two classifiers' softmax probabilities, it can be seen that $\boldsymbol{p}_1$ and $\boldsymbol{p}_2 \in \mathbb{R}^{K\times 1}$ satisfies:

$$\sum_{k=1}^{K} p_j^k = 1, \quad \text{s.t. } p_j^k \geq 0, \ \forall k = 1, \cdots, K, \quad j = 1, 2, \qquad (1)$$

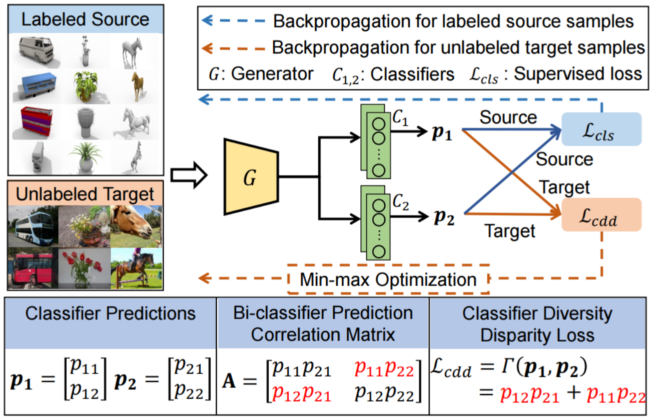
where p_j^k denotes the k-th element of the softmax probabilistic output from C_j, i.e., the probability that C_j classifies the sample into the k-th class, and K is the amount of possible categories. Previous approach, i.e., MCD, deploys ℓ_1 distance $|\boldsymbol{p}_1 - \boldsymbol{p}_2|_1$ to measure the discrepancy between two prediction distributions. However, ℓ_1 distance only considers the similarity between $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ at the same position of the corresponding class, and thus ignores the relevance between classes, which may affect the confidence of predictions. For instance, though the ℓ_1 distance between $\boldsymbol{p}_1 = [0.34, 0.33, 0.33]$ and $\boldsymbol{p}_2 = [0.34, 0.33, 0.33]$ is small to zero, such predictions are rather uncertain across classes and thus the classifiers are less robust to perturbations. Therefore, we investigate the classifier discrepancy by Bi-classifier Prediction Relevance Matrix $\mathbf{A}$:

$$\mathbf{A} = \boldsymbol{p}_1 \boldsymbol{p}_2^\top, \qquad (2)$$

where $\mathbf{A}$ is a square matrix of size $K \times K$. $A_{mn} = p_1^m p_2^n (m, n \in \{1, 2, \cdots, K\})$ is the element in the m-th row and n-th column of $\mathbf{A}$, which represents the probability product of C_1 classifying the sample into the m-th category and C_2 classifying it into the n-th category. In other words, $\mathbf{A}$ could effectively evaluate the bi-classifier prediction relevance across different classes.With the characteristics of $\mathbf{A}$, to minimize the classifier discrepancy with respect to the prediction relevance, we have no alternative but make the two predictions consistent and correlated, i.e., maximizing the diagonal elements of $\mathbf{A}$. This also enables the predictions to be determined with high confidence for the predicted class. At the same time, the non-diagonal elements of $\mathbf{A}$ can be seen as fine-grained confusion information of the two classifiers. Therefore, we define the CDD loss as

$$\Gamma(\boldsymbol{p}_1, \boldsymbol{p}_2) = \sum_{m,n=1}^{K} A_{mn} - \sum_{m=1}^{K} A_{mm} = \sum_{m \neq n}^{K} A_{mn}. \qquad (3)$$

The first item in Eq. (3) equals to 1 since $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are softmax outputs. We can see that CDD contains all the probabilities that the prediction of C_1 is inconsistent with the prediction of C_2, so it can be used as a measure to evaluate the discrepancy between predictions. Notably, $\Gamma(\boldsymbol{p}_1, \boldsymbol{p}_2)$ satisfies the properties of a metric and the proof is provided in the supplementary materials[1]. Moreover, only when the two probabilities are consistent and fully determined, such as [1,0,0] and [1,0,0], can the CDD reach the minimum value of zero, which guarantees the discriminability of features.



Classifier Predictions	Bi-classifier Prediction Correlation Matrix	Classifier Diversity Disparity Loss
$\boldsymbol{p_1} = \begin{bmatrix} p_{11} \\ p_{12} \end{bmatrix}$ $\boldsymbol{p_2} = \begin{bmatrix} p_{21} \\ p_{22} \end{bmatrix}$	$\mathbf{A} = \begin{bmatrix} p_{11}p_{21} & p_{11}p_{22} \\ p_{12}p_{21} & p_{12}p_{22} \end{bmatrix}$	$\mathcal{L}_{cdd} = \Gamma(\boldsymbol{p_1}, \boldsymbol{p_2})$ $= p_{12}p_{21} + p_{11}p_{22}$

Figure 2: Approach overview. In BCDM, step 1 optimizes all the networks by minimizing the source classification loss. Under the source supervision, step 2 maximizes the proposed classifier determinacy disparity (CDD) loss on target domain by updating bi-classifier parameters. The generator seeks to generate transferable and discriminative features by minimizing CDD loss on target domain in step 3. An example of calculating CDD loss is also provided in the bottom, where p_i^j denotes the probability of classifier C_i classifying the input sample into the j-th category.

Bi-Classifier Determinacy Maximization

Based on the introduced CDD metric, here we propose a bi-classifier determinacy maximization (BCDM) algorithm for adversarial domain adaptation. In UDA problems, it is a prerequisite to ensure that the classifiers can correctly classify source data. Thus, to fully exploit the supervision from source domain, we first train the whole network to minimize the standard supervised loss on source data as follows:

$$\min_{\theta_g, \theta_{c_1}, \theta_{c_2}} \frac{1}{n_s} \sum_{i=1}^{n_s} \mathcal{L}_{cls}(\boldsymbol{x}_i^s, y_i^s) = \frac{1}{2n_s} \sum_{i=1}^{n_s} \sum_{j=1}^{2} \mathcal{L}_{ce}(\boldsymbol{p}_{j_i}, y_i^s), \qquad (4)$$

where $\boldsymbol{p}_{j_i} = C_j(G(\boldsymbol{x}_i^s))$ and $\mathcal{L}_{ce}(\cdot, \cdot)$ is the cross entropy loss. With the full supervision on source data, the feature discriminability for source data can be well preserved. However, it is worth noting that the learned decision boundary on the source domain cannot be directly transferred to the target domain due to the distribution divergence between the two domains. Therefore, we further propose to train the two distinct classifiers C_1 and C_2 on target domain in an adversarial

[1] The supplementary materials can be found in the corresponding arXiv version (https://arxiv.org/abs/2012.06995).

manner, which aims to learn transferable features and discriminative decision boundary on target domain. To achieve this, here, we utilize the designed metric CDD to measure the classifier discrepancy. Specifically, the two classifiers are trained to maximize CDD on target domain data,

$$\max_{\theta_{c_1},\theta_{c_2}} \frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}_{cdd}(\boldsymbol{x}_i^t) = \frac{1}{n_t}\sum_{i=1}^{n_t}\Gamma(\boldsymbol{p}_{1_i},\boldsymbol{p}_{2_i})\,. \tag{5}$$

Here, $\boldsymbol{p}_{1_i}$ and $\boldsymbol{p}_{2_i}$ are the softmax outputs of C_1 and C_2 for target sample $\boldsymbol{x}_i^t$, respectively. Training the classifiers via maximizing Eq. (5) can effectively detect target samples that are far from the support of the source domain. In essence, the CDD loss encourages two classifiers to perform differently across categories rather than that of the same category in MCD (Saito et al. 2018b). Besides, Eq. (5) could equip classifiers with more diverse probabilities in a quite simple way that potentially facilitates the discovery of more ambiguous target data near the decision boundary.

Through maximizing Eq. (5), target samples are mostly located near the decision boundary so as to be detected with diverse predictions, which brings uncertainty to target feature learning. To encourage yielding representations that are discriminative and achieve bi-classifier determinacy, we further train the generator G on target domain to minimize CDD while fixing the classifiers,

$$\min_{\theta_g} \frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}_{cdd}(\boldsymbol{x}_i^t) = \frac{1}{n_t}\sum_{i=1}^{n_t}\Gamma(\boldsymbol{p}_{1_i},\boldsymbol{p}_{2_i}). \tag{6}$$

Considering the specific property of CDD that it can reach the minimum value only when the two predictions are consistent and fully determined, i.e., 100% certainty, the generator G is capable of generating target representations that possess great discriminability by minimizing Eq. (6), and thus further benefits the learning task. In contrast, MCD (Saito et al. 2018b) and SWD (Lee et al. 2019) cannot guarantee whether the decision boundary separates the categorical clusters in the target domain since they only focus on the consistency of two predictions. Obviously, the worst case is that the generator is still able to generate confused target features even if the output probabilities are consistent, which severely hinders the feature discriminability.

Remarks. On one hand, to maximize the CDD, we need to minimize the summation of the diagonal elements of matrix $\mathbf{A}$, i.e., $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ need to be as inconsistent as possible with totally different categorical predictions. It enables us to explore more possibilities of classification for each target data, which increases prediction diversity. **On the other hand**, to minimize $\Gamma(\boldsymbol{p}_1,\boldsymbol{p}_2)$ on the target set, the bi-classifier predictions on target data need to be highly correlated and with high certainty, i.e., ideally $\boldsymbol{p}_1 = \boldsymbol{p}_2$ and with probability 1 for the predicted class. Via this minimization, the predictions should be fully determined and thus push the target domain distribution far away from the decision boundary, which stimulates G to generate discriminative representations. Through such adversarial learning of the CDD loss between the two prediction outputs, we promote the feature discriminability and preserve the diversity of predictions simultaneously. To be clear, we summarize the training process of BCDM based on the above discussions as Algorithm 1. Deep Embedded Validation (You et al. 2019) is conducted to select hyper-parameters, and then we fix $\alpha = 0.01$ in all experiments.

Algorithm 1 The Algorithm of BCDM for UDA.

```
Input: Source samples $\{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{n_s}$; Target samples $\{\boldsymbol{x}_j^t\}_{j=1}^{n_t}$; Trade-off parameter: $\alpha$ Batch size: $B$; max_iteration.
Output: Optimal parameters $\widehat{\theta}_g, \widehat{\theta}_{c_1}, \widehat{\theta}_{c_2}$ of feature generator $G$ and task-specific classifiers $C_1, C_2$.
Initialize $\theta_g$ using pre-trained model on ImageNet, and randomly initialize $\theta_{c_1}, \theta_{c_2}$.
repeat
    Randomly sample mini-batch of $B$ source samples and $B$ target samples.
    Update $\theta_g, \theta_{c_1}, \theta_{c_2}$ under the source supervision: $\min_{\theta_g,\theta_{c_1},\theta_{c_2}} \frac{1}{B}\sum_{i=1}^{B}\mathcal{L}_{cls}(\boldsymbol{x}_i^s, y_i^s)$.
    Update $\theta_{c_1}, \theta_{c_2}$ by maximizing classifier determinacy disparity: $\min_{\theta_{c_1},\theta_{c_2}} \frac{1}{B}\sum_{i=1}^{B}\mathcal{L}_{cls}(\boldsymbol{x}_i^s, y_i^s) - \alpha\mathcal{L}_{cdd}(\boldsymbol{x}_i^t)$.
    Update $\theta_g$ by minimizing classifier determinacy disparity: $\min_{\theta_g} \frac{1}{B}\sum_{i=1}^{B}\alpha\mathcal{L}_{cdd}(\boldsymbol{x}_i^t)$.
until max_iteration is reached.
```

Generalization Bound of BCDM

In this subsection, we further provide the theoretical guarantees for BCDM. Limited by space, all the proofs are given in the supplementary materials.

First, we propose a novel discrepancy to measure the difference between two distributions based on our classifier determinacy disparity. Assuming $\mathcal{H}$ is a hypothesis space, given two hypotheses $h_1, h_2 \in \mathcal{H}$, we define the expected Determinacy Disparity on some distribution D as: $dis_D(h_1,h_2) \triangleq \mathbb{E}_D\Gamma(\boldsymbol{p_1},\boldsymbol{p}_2)$.

Definition 1. *Given a hypothesis set $\mathcal{H}$ and a specific hypothesis $h \in \mathcal{H}$, we define the Determinacy Disparity Discrepancy as:*

$$\begin{aligned} d_{h,\mathcal{H}}(\mathcal{S},\mathcal{T}) &\triangleq \sup_{h'\in\mathcal{H}} \left(dis_{\mathcal{T}}(h',h) - dis_{\mathcal{S}}(h',h)\right) \\ &= \sup_{h'\in\mathcal{H}} \left(\mathbb{E}_{\mathcal{T}}\Gamma(\boldsymbol{p'},\boldsymbol{p}) - \mathbb{E}_{\mathcal{S}}\Gamma(\boldsymbol{p'},\boldsymbol{p})\right). \end{aligned} \tag{7}$$

Compared with $\mathcal{H}\Delta\mathcal{H}-$distance (Bendavid et al. 2010), the Determinacy Disparity Discrepancy (DDD) we proposed depends upon a hypothesis space $\mathcal{H}$ and a specific classifier h, which is easier to be optimized. Moreover, since the DDD is well-designed, we verify that it satisfies symmetrical characteristic and trigonometric inequality through strict proof. Thus, according to the theory proposed in (Mansour, Mohri, and Rostamizadeh 2009), we can deduce that the DDD can effectively measure the difference between distributions on domain adaptation, the detail of which is given in the supplementary materials. Besides, we can obtain an upper bound of the expected target error $e_t(h)$ through the DDD.

Proposition 1. *For every classifier $h \in \mathcal{H}$,*

$$e_t(h) \leq e_s(h) + d_{h,\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda. \tag{8}$$

where $e_s(h)$ stands for the expected error on source samples, and λ denotes the error of ideal joint hypothesis, which can be considered rather small if the hypothesis space is sufficient enough (Zhang et al. 2019).

Then we introduce the Rademacher complexity which is commonly used in the generalization theory as a measurement of richness for a particular hypothesis space (Mohri, Rostamizadeh, and Talwalkar 2012), and provide a generalization bound for our method BCDM based on the Determinacy Disparity Discrepancy defined above. To begin with, we introduce a Rademacher complexity bound as follows.

Lemma 1. *Suppose that $\mathcal{G}$ is a class of function maps $\mathcal{X} \to [0, 1]$. Then for any $\delta > 0$, with probability at least $1 - \delta$ and sample size n, the following holds for all $g \in \mathcal{G}$:*

$$|\mathbb{E}_D g - \mathbb{E}_{\widehat{D}} g| \leq 2\mathfrak{R}_{n,D}(\mathcal{G}) + \sqrt{\frac{log\frac{2}{\delta}}{2n}}. \tag{9}$$

Further, we define a new function class $\mathcal{G}_\Gamma \mathcal{H}$ mapping $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ to [0,1], which serves as a "CDD" version of the symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H}$.

Definition 2. *Given a hypothesis space $\mathcal{H}$ and a Classifier Disparity Discrepancy function Γ, the $\mathcal{G}_\Gamma \mathcal{H}$ is defined as*

$$\mathcal{G}_\Gamma \mathcal{H} = \{\boldsymbol{x} \to \Gamma(h_1(\boldsymbol{x}), h_2(\boldsymbol{x}))|h_1, h_2 \in \mathcal{H}\}. \tag{10}$$

Based on this definition, we induce a Rademacher complexity bound for the difference between the Determinacy Disparity and its empirical version.

Proposition 2. *For any $\delta \geq 0$, with probability at least $1-\delta$, the following holds for all $h, h' \in \mathcal{H}$:*

$$|dis_D(h', h) - dis_{\widehat{D}}(h', h)| \leq 2\mathfrak{R}_{n,D}(\mathcal{G}_\Gamma \mathcal{H}) + \sqrt{\frac{log\frac{2}{\delta}}{2n}}. \tag{11}$$

Finally, we can come to the final generalization bound by combining the Rademacher complexity bound of CDD and Proposition 1.

Theorm 1. *For any $\delta \geq 0$, with probability $1 - 3\delta$, we have the following generalization bound for any classifier $h \in \mathcal{H}$:*

$$err_{\mathcal{T}}(h) \leq err_{\widehat{\mathcal{S}}}(h) + d_{h,\mathcal{H}}(\widehat{\mathcal{S}}, \widehat{\mathcal{T}}) + \lambda + \Omega, \tag{12}$$

where $\Omega = 2\mathfrak{R}_{n,\mathcal{S}}(\mathcal{G}_\Gamma \mathcal{H}) + 2\mathfrak{R}_{n,\mathcal{S}}(\mathcal{H}) + 2\sqrt{\frac{log\frac{2}{\delta}}{2n}} + 2\mathfrak{R}_{m,\mathcal{T}}(\mathcal{G}_\Gamma \mathcal{H}) + \sqrt{\frac{log\frac{2}{\delta}}{2m}}$. n, m are the amount of samples in source domain and target domain respectively, and $\mathcal{H}$ is the hypothesis set.

Reviewing our approach, in essence, we minimize the first term $err_{\widehat{\mathcal{S}}}(h)$ in **Step 1**. Later, the min-max process including **Step 2** and **Step 3** implements the minimization of $d_{h,\mathcal{H}}(\widehat{\mathcal{S}}, \widehat{\mathcal{T}})$, since $dis_{\mathcal{S}}(h', h)$ is close to zero after the first step. In summary, our method has a rigorous generalization upper bound which can guarantee the effectiveness of BCDM. Through the training process we can achieve the minimization of the major terms in the upper bound, by which the generalization of classifier on target domain will be improved significantly.

Experiment

Dataset and Setup

We evaluate BCDM against many state-of-the-art algorithms on four domain adaptation datasets and two semantic segmentation datasets. **DomainNet** (Peng et al. 2019) is the largest and hardest dataset to date for visual domain adaptation and consists of about 0.6 million images with 345 classes that spread in 6 distinct domains: Clipart (clp), infograph (inf), Painting (pnt), Quickdraw (qdr), Real (rel) and Sketch (skt); **VisDA-2017** (Peng et al. 2017) is a large-scale synthetic-to-real dataset contains over 280K images across 12 categories; **Office-31** (Saenko et al. 2010) is widely adopted by adaptation methods involving three distinct domains: Amazon (A), DSLR (D) and Webcam (W); **ImageCLEF**[2] is composed of 12 common categories shared by three popular datasets: Caltech-256 (C), ImageNet ILSVRC2012 (I), and PASCAL VOC2012 (P); **Cityscapes** (Cordts et al. 2016) is a real-word dataset with 5,000 urban scenes which are divided into training, validation and test sets; **GTA5** (Richter et al. 2016) contains 24,966 synthesized frames captured from GTA5 game engine. Similar to (Tsai et al. 2018; Lee et al. 2019), we use the 19 classes in common with Cityscapes dataset and report the results on the Cityscapes validation set. Following the standard protocols in (Long et al. 2018; Luo et al. 2019), we use the average classification accuracy and PASCAL VOC intersection-over-union (Everingham et al. 2015), i.e., IoU and mean IoU (mIoU) as evaluation metrics for image classification and semantic segmentation tasks respectively.

Implementation Details

For image classification, we use pre-trained ResNet-50/101 (He et al. 2016) from ImageNet (Deng et al. 2009) as the base feature extractor G and replace the last fully-connected (FC) layer with a bottleneck layer as (Long et al. 2018). The architecture of task-specific classifier is identical to (Saito et al. 2018b): three FC layers with random initialization is attached to the bottleneck layer. The FC layers are trained from the scratch with learning rate 10 times that of the feature generator layers. We adopt Stochastic Gradient Descent optimizer (SGD) (Bottou 2010) with learning rate 3×10^{-4}, momentum 0.9 and weight decay 5×10^{-4}. To schedule the learning rate, we follow the annealing procedure mentioned in (Pei et al. 2018). Besides, we exploit the entropy loss as (Saito et al. 2018b; Long et al. 2016) to make the training procedure more stable in all experiment.

For semantic segmentation, we utilize the DeepLab-v2 (Chen et al. 2018) with ResNet-101 pre-trained on ImageNet as the base architecture G. Also, Atrous Spatial Pyramid Pooling (ASPP) is used for classifier and applied on the $conv5$ feature outputs. Following (Tsai et al. 2018; Luo et al. 2019), sampling rates are fixed as {6, 12, 18, 24} and we modify the stride and dilation rate of the last layers to produce denser feature maps. To train the network, we use the SGD with Nesterov acceleration where initial learning rate is set as 2.5×10^{-4} with momentum 0.9 and weight decay

[2] http://imageclef.org/2014/adaptation

Accuracy(%) on DomainNet for UDA (ResNet-50).

ResNet‡	clp	inf	pnt	qdr	rel	skt	Avg.	CDAN‡	clp	inf	pnt	qdr	rel	skt	Avg.	BNM‡	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	14.2	29.6	9.5	43.8	34.3	26.3	clp	-	13.5	28.3	9.3	43.8	30.2	25.0	clp	-	12.1	33.1	6.2	50.8	40.2	28.5
inf	21.8	-	23.2	2.3	40.6	20.8	21.7	inf	18.9	-	21.4	1.9	36.3	21.3	20.0	inf	26.6	-	28.5	2.4	38.5	18.1	22.8
pnt	24.1	15.0	-	4.6	45.0	29.0	23.5	pnt	29.6	14.4	-	4.1	45.2	27.4	24.2	pnt	39.9	12.2	-	3.4	54.5	36.2	29.2
qdr	12.2	1.5	4.9	-	5.6	5.7	6.0	qdr	11.8	1.2	4.0	-	9.4	9.5	7.2	qdr	17.8	1.0	3.6	-	9.2	8.3	8.0
rel	32.1	17.0	36.7	3.6	-	26.2	23.1	rel	36.4	18.3	40.9	3.4	-	24.6	24.7	rel	48.6	13.2	49.7	3.6	-	33.9	29.8
skt	30.4	11.3	27.8	3.4	32.9	-	21.2	skt	38.2	14.7	33.9	7.0	36.6	-	26.1	skt	54.9	12.8	42.3	5.4	51.3	-	33.3
Avg.	24.1	11.8	24.4	4.7	33.6	23.2	20.3	Avg.	27.0	12.4	25.7	5.1	34.3	22.6	21.2	Avg.	37.6	10.3	31.4	4.2	40.9	27.3	25.3
MCD‡	clp	inf	pnt	qdr	rel	skt	Avg.	SWD‡	clp	inf	pnt	qdr	rel	skt	Avg.	**BCDM**	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	15.4	25.5	3.3	44.6	31.2	24.0	clp	-	14.7	31.9	10.1	45.3	36.5	27.7	clp	-	17.2	35.2	10.6	50.1	40.0	30.6
inf	24.1	-	24.0	1.6	35.2	19.7	20.9	inf	22.9	-	24.2	2.5	33.2	21.3	20.0	inf	29.3	-	29.4	3.8	41.3	25.0	25.8
pnt	31.1	14.8	-	1.7	48.1	22.8	23.7	pnt	33.6	15.3	-	4.4	46.1	30.7	26.0	pnt	39.2	17.7	-	4.8	51.2	34.9	29.6
qdr	8.5	2.1	4.6	-	7.9	7.1	6.0	qdr	15.5	2.2	6.4	-	11.1	10.2	9.1	qdr	19.4	2.6	7.2	-	13.6	12.8	11.1
rel	39.4	17.8	41.2	1.5	-	25.2	25.0	rel	41.2	18.1	44.2	4.6	-	31.6	27.9	rel	48.2	21.5	48.3	5.4	-	36.7	32.0
skt	37.3	12.6	27.2	4.1	34.5	-	23.1	skt	44.2	15.2	37.3	10.3	44.7	-	30.3	skt	50.6	17.3	41.9	10.6	49.0	-	33.9
Avg.	28.1	12.5	24.5	2.4	34.1	21.2	20.5	Avg.	31.5	13.1	28.8	6.4	36.1	26.1	23.6	Avg.	37.3	15.3	32.4	7.0	41.0	29.9	**27.2**

Accuracy(%) on DomainNet for UDA (ResNet-101).

ResNet	clp	inf	pnt	qdr	rel	skt	Avg.	CDAN‡	clp	inf	pnt	qdr	rel	skt	Avg.	BNM‡	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	19.3	37.5	11.1	52.2	41.0	32.2	clp	-	17.8	35.7	15.3	51.3	37.2	31.4	clp	-	19.4	35.6	16.1	49.8	36.3	31.4
inf	30.2	-	31.2	3.6	44.0	27.9	27.4	inf	25.4	-	28.9	5.8	38.2	22.8	24.2	inf	24.6	-	27.8	7.9	35.0	22.0	23.5
pnt	39.6	18.7	-	4.9	54.5	36.3	30.8	pnt	37.1	17.9	-	7.9	51.4	34.0	29.7	pnt	36.0	20.2	-	9.7	51.8	34.2	30.4
qdr	7.0	0.9	1.4	-	4.1	8.3	4.3	qdr	20.5	2.3	7.7	-	14.6	12.6	11.5	qdr	21.3	3.8	10.5	-	14.0	12.9	12.5
rel	48.4	22.2	49.4	6.4	-	38.8	33.0	rel	43.6	19.4	46.1	8.3	-	33.2	30.1	rel	43.4	21.7	47.0	9.9	-	32.9	31.0
skt	46.9	15.4	37.0	10.9	47.0	-	31.4	skt	45.4	18.3	40.4	14.5	48.3	-	33.4	skt	43.1	19.1	39.5	15.6	47.0	-	32.7
Avg.	34.4	15.3	31.3	7.4	40.4	30.5	26.5	Avg.	34.4	15.1	31.7	10.4	40.8	27.9	26.7	Avg.	33.7	16.8	32.1	11.8	39.6	27.7	26.9
MCD	clp	inf	pnt	qdr	rel	skt	Avg.	SWD‡	clp	inf	pnt	qdr	rel	skt	Avg.	**BCDM**	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	14.2	26.1	1.6	45.0	33.8	24.1	clp	-	16.6	35.3	12.8	48.7	41.0	30.9	clp	-	19.9	38.5	15.1	53.2	43.9	34.1
inf	23.6	-	21.2	1.5	36.7	18.0	20.2	inf	26.9	-	27.6	2.7	38.1	25.4	24.1	inf	31.9	-	32.7	6.9	44.7	28.5	28.9
pnt	34.4	14.8	-	1.9	50.5	28.4	26.0	pnt	37.3	16.9	-	5.9	48.7	34.6	28.7	pnt	42.5	19.8	-	7.9	54.5	38.5	32.6
qdr	15.0	3.0	7.0	-	11.5	10.2	9.3	qdr	19.3	3.0	8.1	-	14.2	13.3	11.6	qdr	23.0	4.0	9.5	-	16.9	16.2	13.9
rel	42.6	19.6	42.6	2.2	-	29.3	27.2	rel	47.0	19.9	47.1	6.1	-	36.8	31.4	rel	51.9	24.9	51.2	8.7	-	40.6	35.5
skt	41.2	13.7	27.6	3.8	34.8	-	24.2	skt	48.8	17.3	41.1	12.2	49.1	-	33.7	skt	53.7	20.5	46.0	13.1	53.4	-	37.1
Avg.	31.4	13.1	24.9	2.2	35.7	23.9	21.9	Avg.	35.9	14.7	31.8	7.9	39.8	30.2	26.7	Avg.	40.6	17.8	35.6	10.3	44.3	33.5	**30.4**

Table 1: Accuracy(%) on DomainNet for unsupervised domain adaptation. In each sub-table, the column-wise domains are selected as the source domain and the row-wise domains are selected as the target domain. ‡: Results we obtained using the publicly released codes by the authors without any change and others are borrowed from (Peng et al. 2019).

10^{-4}. And we use the polynomial annealing procedure as mentioned in (Chen et al. 2018). Regarding the training procedure, the network is first trained with $\mathcal{L}_{cls}$ for 20k iterations and then fine-tune using Algorithm 1 for 40k iterations.

Experimental Results

DomainNet. As illustrated in Table 1, our BCDM significantly outperforms the comparison methods (i.e., CDAN, BNM, SWD, MCD) in terms of final mean accuracy. Note that the mainstream domain adaptation baseline, i.e., MCD, suffers from negative transfer (Pan and Yang 2010). An interpretation is that the MCD model is category agnostic and will bring side effects of the deterioration of the representation discriminability, especially when there exist notable domain gaps and a large number of categories on this dataset. In spite of that, our method still yields large improvement over other baselines for the scenarios of using both ResNet-50/101 backbones, which highlights its superiority in aligning distinct domains with large class variation and its suitability for this imbalanced dataset (Peng et al. 2019).

VisDA-2017. Table 2 compares various methods with the pretrained ResNet-101 backbone. Clearly, we outperform other approaches by large margins on 7 out of 12 tasks. Furthermore, our BCDM brings a rise of improvement by **31.0**% over the source only model (ResNet-101). Compared with MCD and SWD, which explicitly utilize adversarial learning with bi-classifier to align features across source domain and target domain, our model achieves extra gains of **11.5**% and **7.0**% respectively. Besides, the BCDM surpasses them on bicycle, knife, and sktbrd classes by large margins. Moreover, we achieve decent gains compared to the best baseline STAR. The considerable results reflect that the proposed classifier determinacy disparity metric is effective in associating two different but related distributions of data.

Office-31 and ImageCLEF. Experimental results are reported in Table 3. On both datasets, our BCDM obtains superior results over other popular adaptation methods and achieves the best average accuracies (**89.0**% on Office-31 and **89.3**% on ImageCLEF). The results show that BCDM is beneficial to promote the adaptation capability, especially

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
ResNet	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DAN	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
MinEnt	80.3	75.5	75.8	48.3	77.9	27.3	69.7	40.2	46.5	46.6	79.3	16.0	57.0
DANN	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
CDAN	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
JADA	91.9	78.0	81.5	68.7	90.2	84.1	84.0	73.6	88.2	67.2	79.0	38.0	77.0
AFN	93.6	61.3	84.1	70.6	**94.1**	79.0	**91.8**	79.6	89.9	55.6	89.0	24.4	76.1
BNM	89.6	61.5	76.9	55.0	89.3	69.1	81.3	65.5	90.0	47.3	**89.1**	30.1	70.4
MCD	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
SWD	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
STAR	95.0	84.0	**84.6**	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	**42.2**	82.7
BCDM	**95.1**	**87.6**	81.2	**73.2**	92.7	**95.4**	86.9	**82.5**	**95.1**	**84.8**	88.1	39.5	**83.4**

Table 2: Accuracy(%) on VisDA-2017 for unsupervised DA (ResNet-101).

Method	Office-31							ImageCLEF						
	A → W	D → W	W → D	A → D	D → A	W → A	Avg.	I → P	P → I	I → C	C → I	C → P	P → C	Avg.
ResNet	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4	74.5	82.2	92.8	86.3	69.2	89.8	82.5
MinEnt	86.8±0.2	**98.6±0.1**	**100.0±.0**	88.7±0.3	67.2±0.5	63.4±0.4	84.1	76.2	85.7	93.5	83.5	69.3	89.7	83.0
DANN	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2	75.0	86.0	96.2	87.0	74.3	91.5	85.0
CDAN	94.1±0.1	98.6±0.1	**100.0±.0**	92.9±0.2	71.0±0.3	69.3±0.3	87.7	77.7	90.7	**97.7**	91.3	74.2	94.3	87.7
JADA	90.5±0.1	97.5±0.3	**100.0±.0**	88.2±0.1	70.9±0.5	70.6±0.4	86.1	78.2	90.1	95.9	90.8	76.8	94.1	87.7
AFN	90.1±0.8	98.6±0.2	99.8±.0	90.7±0.5	73.0±0.2	70.2±0.3	87.1	79.3	**93.3**	96.3	**91.7**	77.6	95.3	88.9
BNM	91.5±.0	98.5±.0	**100.0±.0**	90.3±.0	70.9±.0	71.6±.0	87.1	77.2	91.2	96.2	**91.7**	75.7	**96.7**	88.1
MCD	88.6±0.2	98.5±0.1	**100.0±.0**	92.2±0.2	69.5±0.1	69.7±0.3	86.5	77.3	89.2	92.7	88.2	71.0	92.3	85.1
SWD	90.4±0.4	**98.7±0.2**	**100.0±.0**	**94.7±0.4**	70.3±0.2	70.5±0.5	87.4	78.1	89.6	95.2	89.3	73.4	92.8	86.4
STAR	92.6±0.1	**98.7±0.1**	**100.0±.0**	93.2±0.1	71.4±0.3	70.8±0.1	87.8	78.8	90.5	96.2	91.2	75.5	93.8	87.7
BCDM	**95.4±0.3**	98.6±0.1	**100.0±.0**	93.8±0.3	**73.1±0.3**	**73.0±0.2**	**89.0**	**79.5**	93.2	96.8	91.3	**78.9**	95.8	**89.3**

Table 3: Classification Accuracy (%) on Office-31 and ImageCLEF Datasets (ResNet-50).

Method	road	side.	buil.	wall	fence	pole	light	sign	veg	terr.	sky	pers.	rider	car	truck	bus	train	mbike	bike	mIoU
Source only	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	**36.0**	36.6
AdaSegNet	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
AdvEnt	89.9	36.5	81.6	29.2	25.2	28.5	32.3	22.4	83.9	34.0	77.1	57.4	27.9	83.7	29.4	39.1	1.5	28.4	23.3	43.8
SWD	**92.0**	**46.4**	82.4	24.8	24.0	**35.1**	33.4	**34.2**	83.6	30.4	80.9	56.9	21.9	82.0	24.4	28.7	6.1	25.0	33.6	44.5
CLAN	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	**33.1**	36.7	6.7	**31.9**	31.4	43.2
STAR	88.4	27.9	80.8	27.3	**25.6**	26.9	31.6	20.8	83.5	34.1	76.6	**60.5**	27.2	84.2	32.9	38.2	1.0	30.2	31.2	43.6
BCDM	90.5	37.3	**83.7**	**39.2**	22.2	28.5	**36.0**	17.0	**84.2**	**35.9**	**85.8**	59.1	**35.5**	**85.2**	31.1	**39.3**	**21.1**	26.7	27.5	**46.6**

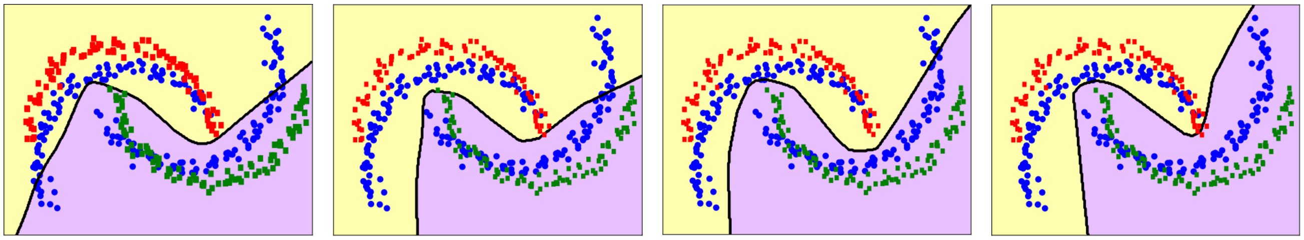
Table 4: Semantic segmentation performance mIoU (%) on Cityscapes validation set.

on the difficult scenarios where the baseline accuracy is relatively low, e.g., D → A, W → A and C → P.

GTA5 → Cityscapes. Unlike image classification, segmentation task generally requires more effort of human labor to annotate each pixel in an image. Here we extend our BCDM to perform domain adaptive semantic segmentation task. We present quantitative results of transferring GTA5 to Cityscapes in Table 4 with the best results highlighted in bold. Even with a large domain shift between synthetic-to-real scenes, the model trained on our BCDM has been shown to outperform the model trained on Source only by a significant gain of **10.0**%. Additionally, our method consistently outperforms other recent approaches that utilize bi-classifier (Lee et al. 2019; Luo et al. 2019; Lu et al. 2020).

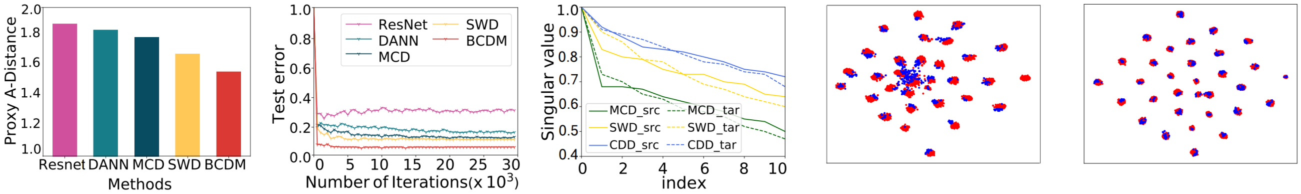
Insight Analysis

Toy Experiment. We perform a toy experiment on the inter twinning moons 2D problems (Pedregosa et al. 2011) to provide analysis on the learned decision boundaries of ResNet-50, MCD, SWD and BCDM. As indicated in Fig. 3, we generate two classes of source samples, one is an upper moon, the other is a lower moon, labeled 0 and 1, respectively. For the target samples, we enhance domain gap by rotating 30 degrees of the distribution of the source samples to generate them. We generate 100 samples per class as the training samples. Both feature generator and classifiers employ a 3-layered fully-connected network. After convergence, ResNet-50 (Fig. 3(a)) correctly classifies the source samples under the supervision of source labels but does badly in target samples. MCD (Fig 3(b)) is able to re-



(a) ResNet-50 (b) MCD (c) SWD (d) BCDM

Figure 3: Comparisons of four decision boundaries on a toy dataset. Red points and green points denote the class 0 and 1 of source data, respectively. Blue points are target data generated from the same distribution as the source data but with domain shift by rotation. The yellow and magenta regions are classified as class 0 and 1 by the final decision boundary, respectively.



(a) $\mathcal{A}$-distance (b) Convergence (c) SVD analysis (d) MCD (e) BCDM

Figure 4: Visualization of $\mathcal{A}$-distance, convergence, SVD analysis, t-SNE on task A (red) $\rightarrow$ W (blue) of Office-31.

duce the distribution shift between domains to some extent, but it is not good enough to perform properly in the whole target samples. SWD (Fig. 3(c) is capable of perfectly aligning its decision boundary in yellow region (class 0) but fails in magenta region (class 1). In contrast, BCDM (Fig. 3(d)) achieves the best matching to the target domain samples and obtains the right decision boundary for all classes.

Generalization Results. To further demonstrate the generalization of our metric, classifier determinacy disparity (cdd) can be used as an additive module to the latest bi-classifier UDA method STAR (Lu et al. 2020), and we compare its performance against ℓ_1 and Wasserstein metric (swd). As shown in Table 5, using cdd achieves the best performance compared to ℓ_1 and swd. Specifically, our proposed metric improves the baseline approach by up to 1.4%.

Proxy $\mathcal{A}$-distance. As shown in Fig. 4(a), we compute the proxy $\mathcal{A}$-distance of ResNet-50, DANN, MCD, SWD and BCDM task-specific feature representations for task A $\rightarrow$ W of Office-31. We observe that BCDM has the smallest $\mathcal{A}$-distance among the compared approaches which justifies that the representations of BCDM are more indistinguishable and can reduce the domain divergence effectively.

Convergence. We present the test error convergence curve with respect to the number of iterations on task A $\rightarrow$ W as shown in Fig 4(b). It can be observed that BCDM achieves comparable convergence with smaller errors.

SVD analysis. We plot the singular values (max-normalized) of features extracted by MCD, SWD and BCDM in Fig. 4(c). BCDM keeps high values while successfully reducing the big difference between the largest and the rest. It implies that more dimensions corresponding to smaller singular values pose positive influence on classification, which intuitively boosts the feature discriminability.

Method	STAR w/ ℓ_1	STAR w/ swd	STAR w/ cdd
Avg. (Δ)	87.8 (-)	88.2 (0.4 $\uparrow$)	**89.2 (1.4 $\uparrow$)**

Table 5: Generalized Results (%) on Office-31 Dataset.

t-SNE Visualization. We visualize features of the last fc-layer of ResNet backbone using t-SNE (Maaten and Hinton 2008) in Fig. 4(d)-4(e). The target data are not aligned well with source data using MCD, while BCDM can learn highly discriminative features and keep clear class boundaries.

Conclusion

In this paper, we investigate that the representation discriminability and classifier determinacy imply more transferability, which is a general exploration for unsupervised domain adaptation. Accordingly, we introduce a new metric, classifier determinacy disparity (CDD), to felicitously evaluate the discrepancy across distinct target predictions. With the merits of CDD and its theoretical guarantees, we propose Bi-Classifier Determinacy Maximization (BCDM) to adversarially optimize the CDD loss, achieving significant performance. Extensive results convincingly demonstrate that the BCDM outperforms the state-of-the-arts on a wide range of unsupervised domain adaptation scenarios including both image classification and semantic segmentation tasks.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61902028).

Ethics Statement

Similar to some automation-related research which has an impact on job loss, our work, which focuses on the application of automated annotation for an unseen domain, is no exception. Specifically, this work has a positive impact on society and community to save the cost and time of data annotation, which can greatly improve the efficiency. However, this work suffers from some negative consequences, which worth us carrying out further researches to explore. For example, more jobs of classification or object detection for rare or variable conditions may be canceled. Furthermore, we should be cautious of the result of the failure of the system, which could render people believe that classification was unbiased. Still, it might be not, which might be misleading. Finally, this work does leverage biases in the data, which is the primary task of this work.

References

Bendavid, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *MLJ* 79(1-2): 151–175.

Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, 177–186.

Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *TPAMI* 40(4): 834–848.

Chen, X.; Wang, S.; Long, M.; and Wang, J. 2019. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, 1081–1090.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 3213–3223.

Cui, S.; Wang, S.; Zhuo, J.; Li, L.; Huang, Q.; and Qi, T. 2020. Towards Discriminability and Diversity: Batch Nuclear-norm Maximization under Label Insufficient Situations. In *CVPR*, 3941–3950.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Feifei, L. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Everingham, M.; Eslami, S. M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2015. The Pascal Visual Object Classes Challenge: A Retrospective. *IJCV* 111(1): 98–136.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised Domain Adaptation by Backpropagation. In *ICML*, 1180–1189.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2066–2073. IEEE.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*, 2672–2680.

Gretton, A.; Borgwardt, K. M.; Rasch, M.; Schölkopf, B.; and Smola, A. J. 2007. A kernel method for the two-sample-problem. In *NeurIPS*, 513–520.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Jin, Y.; Wang, X.; Long, M.; and Wang, J. 2020. Minimum Class Confusion for Versatile Domain Adaptation. In *ECCV*, volume 12366, 464–480. Springer.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NeurIPS*, 1097–1105.

Lee, C.-Y.; Batra, T.; Baig, M. H.; and Ulbricht, D. 2019. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, 10285–10295.

Li, S.; Liu, C. H.; Lin, Q.; ; Wen, Q.; Su, L.; Huang, G.; and Ding, Z. 2020a. Deep Residual Correction Network for Partial Domain Adaptation. *TPAMI* 1–1.

Li, S.; Liu, C. H.; Xie, B.; Su, L.; Ding, Z.; and Huang, G. 2019. Joint Adversarial Domain Adaptation. In *ACM MM*, 729–737. ACM.

Li, S.; Liu, H. C.; Lin, Q.; Xie, B.; Ding, Z.; Huang, G.; and Tang, J. 2020b. Domain Conditioned Adaptation Network. In *AAAI*, 11386–11393.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning transferable features with deep adaptation networks. In *ICML*, 97–105. ACM.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional adversarial domain adaptation. In *NeurIPS*, 1647–1657.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2016. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*, 136–144.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2017. Deep Transfer Learning with Joint Adaptation Networks. In *ICML*, 2208–2217. ACM.

Lu, Z.; Yang, Y.; Zhu, X.; Liu, C.; Song, Y.-Z.; and Xiang, T. 2020. Stochastic Classifiers for Unsupervised Domain Adaptation. In *CVPR*, 9111–9120.

Luo, Y.; Zheng, L.; Guan, T.; Yu, J.; and Yang, Y. 2019. Taking A Closer Look at Domain Shift: Category-level Adversaries for Semantics Consistent Domain Adaptation. In *CVPR*, 2502–2511.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR* 9(Nov): 2579–2605.

Mansour, Y.; Mohri, M.; and Rostamizadeh, A. 2009. Domain Adaptation: Learning Bounds and Algorithms. *CoRR, abs/0902.3430* .

Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2012. *Foundations of Machine Learning*. The MIT Press. ISBN 026201825X.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2011. Domain adaptation via transfer component analysis. *TNNLS* 22(2): 199–210.

Pan, S. J.; and Yang, Q. 2010. A survey on transfer learning. *TKDE* 22(10): 1345–1359.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine Learning in Python. *JMLR* 12: 2825–2830.

Pei, Z.; Cao, Z.; Long, M.; and Wang, J. 2018. Multi-adversarial domain adaptation. In *AAAI*, 3934–3941.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment Matching for Multi-Source Domain Adaptation. In *ICCV*, 1406–1415.

Peng, X.; Usman, B.; Kaushik, N.; Hoffman, J.; Wang, D.; and Saenko, K. 2017. Visda: The visual domain adaptation challenge. *CoRR, abs/1710.06924* .

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 91–99.

Richter, S. R.; Vineet, V.; Roth, S.; and Koltun, V. 2016. Playing for data: Ground truth from computer games. In *ECCV*, 102–118.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *ECCV*, 213–226. Springer.

Saito, K.; Ushiku, Y.; Harada, T.; and Saenko, K. 2018a. Adversarial Dropout Regularization. In *ICLR*.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018b. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 3723–3732.

Tsai, Y.; Hung, W.; Schulter, S.; Sohn, K.; Yang, M.; and Chandraker, M. 2018. Learning to Adapt Structured Output Space for Semantic Segmentation. In *CVPR*, 7472–7481.

Tzeng, E.; Hoffman, J.; Darrell, T.; and Saenko, K. 2015. Simultaneous deep transfer across domains and tasks. In *ICCV*, 4068–4076.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep domain confusion: Maximizing for domain invariance. *CoRR, abs/1412.3474* .

Xu, R.; Li, G.; Yang, J.; and Lin, L. 2019. Larger Norm More Transferable: An Adaptive Feature Norm Approach for Unsupervised Domain Adaptation. In *ICCV*, 1426–1435.

You, K.; Wang, X.; Long, M.; and Jordan, M. 2019. Towards accurate model selection in deep unsupervised domain adaptation. In *ICML*, 7124–7133.

Zhang, Y.; Liu, T.; Long, M.; and Jordan, M. 2019. Bridging Theory and Algorithm for Domain Adaptation. In *ICML*, 7404–7413.