Unsupervised Domain Adaptation with Implicit Smoothness Constraints

Anonymous CVPR submission

Paper ID 4502

Abstract

Unsupervised Domain Adaptation (UDA) aims to leverage the labeled source data and unlabeled target data to generalize better in target domain. Some regularization based methods have achieved remarkable performances in UDA due to exploiting the properties of target data. These methods face with the accumulation of misclassification issue, where samples of the same category are prone to be classified into several specific categories, which is impossible to be settled by explicitly enforcing the label consistency on unlabeled target domain samples within the same category. In this paper, we propose a simple yet effective smoothness constraint based on the Minimum Class Confusion framework. It implicitly reduces the model sensitivity to perturbations for each target sample, and the instance class confusion is further used as the weighting mechanism to promote the effectiveness of the constraints. Extensive experiments show that our method brings great improvements, and yields results outperforming or at least comparable to the state-of-the-arts on four public datasets.

1. Introduction

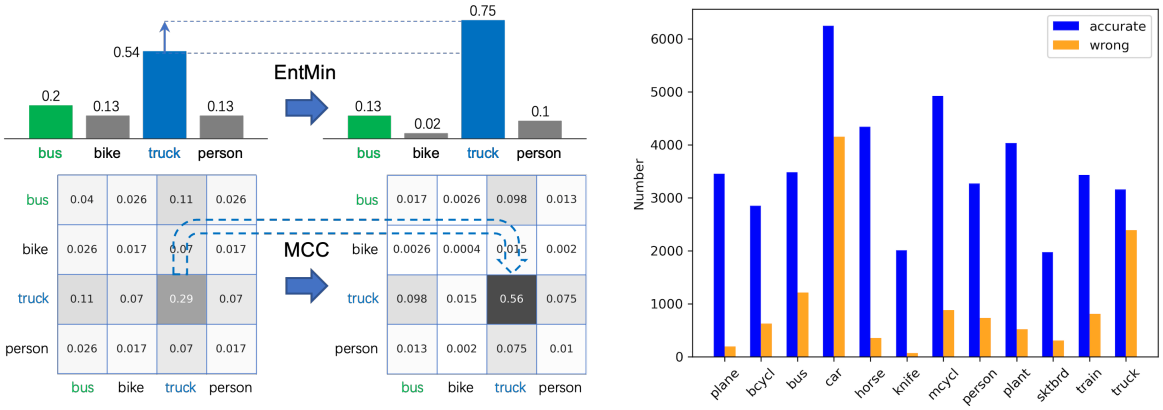
Deep learning methods have achieved great success on a wide variety of tasks. However, when the training set and test set are drawn from different data distributions, the deep learning model would always have poor generalization performance on the test set. To handle this problem, the model is trained to transfer knowledge from a labeled source domain to an unlabeled target domain, based on popular assumptions such as Covariate Shift [32].

On the above Unsupervised Domain Adaptation (UDA) scenario, deep UDA methods have almost dominated this field with promising results [5, 6, 14, 23, 39, 43]. To learn domain-invariant feature, methods like [23, 43] imposed adversarial training leading to better domain distribution alignment. Recently, researchers found that by adding the specific regularization items on target data could obtain a striking performance [5, 14, 39] due to exploiting the properties of target data. For example, AFN [39] added regularization on target features, while BNM [5] and Minimum Class Confusion (MCC) [14] added regularization on classification responses. MCC estimated and minimized the class confusion for target classification responses, which achieved surprising performances on several public datasets. However, it still faces with serious accumulation of misclassification on target domain similar to Entropy Minimization (EntMin) [9].

To dive into the misclassification issue of a series of regularization based methods such as EntMin and MCC, we simulated a toy example to show how the probability would change for EntMin or MCC. For simplicity, we assume that the classifier is trained to classify four classes. For a target sample of the *bus* class, we assume that it is misclassified into *truck* with a high confidence. After adding EntMin constraint, the model tends to produce a higher probability for *truck* and lower probabilities for other classes, including the ground truth label *bus*, as shown at the top of Fig. 1a. Similar circumstance occurs when utilizing MCC to train the model that the values except the diagonals get smaller but the value of the diagonal corresponding to *truck* becomes larger. The consequence is that the probability of *truck* goes up, and this sample will be misclassified to the *truck* category with higher confidence.

In regularization based methods, MCC is far superior to others for UDA. However, the accumulation of misclassification issue misleads the optimization of MCC. As shown in Fig. 1b, the statistics show predictions of MCC on VisDA-C [28]. The values of blue bins represent the accurate predictions, while the orange corresponds to the wrong. We note that misclassification of the *car* and *truck* classes are more serious, and most of the classes are affected.

The accumulation of misclassification leads the model to obtain unsatisfactory results. We aim to deal with this issue by equipping the model with high *smoothness*. The high smoothness guarantees that predictions of two samples of the same class would not differ too much, preventing them being misclassified into two different classes with high confidence. However, as the labels in the target domain are not available, it is impossible to explicitly enforce the label consistency on unlabeled target samples within the same cate-



(a) Simulated training process

(b) Predictions of MCC on VisDA-C

Figure 1. Illustration of (a) the accumulation of misclassification exists in both EntMin [9] and MCC [14] in simulated training process, the ground truth label of this sample is *bus*, and (b) Prediction statistics of MCC [14] on VisDA-C [28]. (Best viewed in color)

gory. Fortunately, we can borrow some ideas from Semi-Supervised Learning (SSL) methods [3, 33]. The key in our method is that reduce the sensitivity of the model over image perturbations of target samples, which encourages the model to generate consistent predictions of target samples within the same category.

Based on the MCC framework, we define an implicit model smoothness constraint that some perturbations are imposed to target images, and model predictions of perturbed image and the original one are constrained to be similar. This constraint increases the smoothness of model on target domain. Then even if samples within the same class may vary, their outputs should not change drastically, preventing samples of the same class from being misclassified into different classes with high confidence.

In order to better prevent the misclassifications of target samples within the same class, we should pay more attention to more reliable samples. Therefore, we adopt the instance class confusion to reweight the smoothness constraints, where a sample with smaller instance class confusion is assigned with larger weight. By the weighting mechanism, our method achieves the state-of-the-art performance on DomainNet [27], and yields the results comparable to the state-of-the-arts on VisDA-C, Office-Home [37], and Office-31 [29].

2. Related Work

Domain adaptation aims to transfer source domain knowledge to the related target domain, and there are various settings of this field, such as Unsupervised Domain Adaptation [22, 43], Unsupervised Model Adaptation [18, 20], Semi-Supervised Domain Adaptation [15, 30, 40], etc. Most of works focus on UDA which is adopted in this paper. We also review the related regularization methods.

2.1. Unsupervised Domain Adaptation

The deep unsupervised domain adaptation methods have made a success without any target supervision. These methods can be mainly divided into domain-invariant learning methods and regularization methods. For the domain-invariant learning methods, the early methods [22, 24, 25, 35, 42] are based on feature distribution matching. [35] firstly used the MMD [10] in the deep neural network to deal with the domain adaptation. And DAN [22] aimed to better align the distribution by utilizing the multiple kernel variant of MMD over multiple layer representations. JAN [25] considered the joint distributions of features and logits, which made distribution alignment more effective. CMD [42] proposed a new domain discrepancy metric called central moment discrepancy, and this metric can be estimated by several numbers of central moments.

Due to the potential of Generative Adversarial Network (GAN) [8], various works achieved better performances by using GANs or the adversarial learning. And domain-invariant learning has been promoted by utilizing the adversarial training. DANN [7] designed a novel adversarial pipeline to handle the domain adaptation. It imposed a domain classifier to discriminate the samples both from source and target domain, and encouraged the model to generate representations that confuse the domain classifier and learn domain-invariant representations. CDAN [23] conducted adversarial learning on the covariance of feature representations and classifier predictions, which makes the domain-invariant feature learning more elaborate. MCD [31] explicitly utilized two task-specific classifiers to measure the domain discrepancy. According to the $\mathcal{H}\Delta\mathcal{H}$ theory [2], MCD first approximately calculated the domain $\mathcal{H}\Delta\mathcal{H}$ distance by the supremum of the expected disagreement of two classifiers' predictions, and the generator tried to minimize

this discrepancy. MDD [43] proposed a novel margin disparity discrepancy that firstly leveraged the scoring function and margin loss to bound the gap caused by domain shift.

2.2. Regularization Based Methods

Semi-Supervised Learning (SSL) is a foundational area and becomes active recently. Contrast learning [11] or data augmentation [3, 33] methods that have achieved good performance in this field, as well as regularization methods [9, 16]. Entropy minimization (EntMin) [9] can enrich the discriminability of unlabeled data to acquire considerable improvement of performance. Temporal ensembling [16] by constraining the consistency of different training epochs enables model to better learn unlabeled data. Also, regularization methods are widely used in various fields because of their simple yet effective implementations.

Domain adaptation can be considered as a special case of SSL, where labeled and unlabeled data are drawn from different distributions. In UDA, EntMin is considered as an orthogonal technique that could cooperate well with other methods such as CDAN [23] or SHOT [20]. AFN [39] investigated that those task-specific features with larger norms are more transferable, so it proposed a feature norm regularization. BNM [5] proved that the batch nuclear-norm maximization can lead to the improvement on both the prediction discriminability and diversity, which works well in domain adaptation, SSL and open domain recognition [44]. MCC [14] can be considered as a regularization method, and it restrained the inter-class confusion of unlabeled data.

The regularization item is not only defined on target features or classification responses, but also can be the consistency regularization in self-ensembling (SE) [6]. SE [6] enforced the prediction consistency between the student network and teacher network, where the weights of teacher network are the exponential moving average (EMA) of those of the student network, and this regularization achieved superior performance in VisDA-C [28]. And [1] conceptually explored the regularization methods by comparing the gradient norms between regularization loss and cross-entropy loss in SSL. Additionally, it conducted experiments to prove that the model ensembling and weight averaging in regularization methods are effective.

3. Methodology

In UDA, we are given a source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ of n_s labeled samples and a target domain $D_t = \{(x_i^t)\}_{i=1}^{n_t}$ of n_t unlabeled samples. The two domains share the same K categories but their distributions follow the Covariate Shift [32] assumption. Specifically, the input marginal distribution $p(x)$ changes but the conditional $p(y|x)$ remains the same.

The overall framework is shown in Fig. 2, which is equipped with a feature extractor and a classifier. Here, the feature extractor $\mathcal{G}$ consists of the deep convolution network and a bottleneck layer that is introduced to reduce the dimension of features, and the features are passed through the classifier $\mathcal{F}$ to generate predictions.

3.1. Regularization with Batch Class Confusion

In UDA, target samples can be learned in a variety of ways, such as aligning the target features with the source features [22, 23, 25], or directly imposing constraints on target samples [5, 39]. Researchers in [14] proposed a novel method by minimizing the pair-wise class confusion in mini-batch training. Suppose that there are source samples $X^s = \{x_i^s\}_{i=1}^b$ and corresponding labels $Y^s = \{y_i^s\}_{i=1}^b$ and target samples $X^t = \{x_i^t\}_{i=1}^b$. X^s and X^t are fed through the model. Then the logit outputs of source batch X^s and target batch X^t are defined as follows:

$$Z^s = \mathcal{F}(\mathcal{G}(X^s)), Z^t = \mathcal{F}(\mathcal{G}(X^t)) \tag{1}$$

and the source logits pass through the softmax function σ to obtain the source classification responses $\hat{Y}^s$ via $\hat{Y}^s = \sigma(Z^s)$. And the standard classification loss is defined as below:

$$\mathcal{L}_s = \frac{1}{b}\sum_{i=1}^{b} H\left(\hat{Y}_i^s, Y_i^s\right) \tag{2}$$

where $H(\cdot,\cdot)$ represents the cross entropy loss, and Y^s represents the ground truth labels.

To alleviate the negative effect of overconfident predictions, the temperature rescaling is adopted to obtain rescaled probability $\tilde{Y}^t$ of target batch as below:

$$\tilde{Y}^t = \sigma\left(Z^t/\mathcal{T}\right) \tag{3}$$

where the division of Z^t with respect to $\mathcal{T}$ is the element level operation, and the $\mathcal{T}$ is the hyperparameter of temperature.

To train a model that performs well in target domain, MCC [14] estimates the confusion among classes and minimizes the confusion. The class correlation C reflects the pair-wise class correlation on target batch, the C with respect to class i and class j is defined as below:

$$C_{i,j} = \tilde{Y}_{\cdot i}^{t\top} \tilde{Y}_{\cdot j}^t \tag{4}$$

where $\tilde{Y}_{\cdot i}^t$ represents the probabilities of target batch come from the i-th class, and $\top$ is the matrix transposing.

Confusion among classes is not completely symmetric, because many samples of a class may be misclassified into several classes, but the reverse is not necessarily true. So, the category normalization [38] adopted by [14] is defined as follow:

$$\tilde{C}_{i,j} = \frac{C_{i,j}}{\sum_{j=1}^{K} C_{i,j}} \tag{5}$$

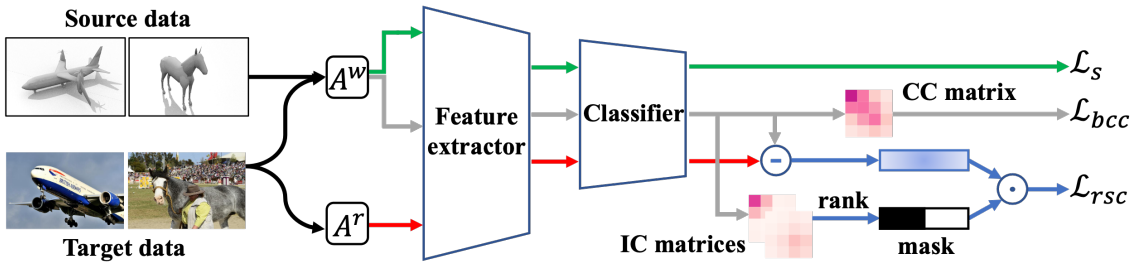


Figure 2. Overall framework. The weakly augmented source data with *green flow* are used to calculate the standard classification loss $\mathcal{L}_s$ with labels. The weakly augmented target data with *gray flow* generate the **C**lass **C**onfusion (CC) matrix and **I**nstance **C**lass confusion (IC) matrices. CC matrix is utilized to calculate the batch class confusion loss $\mathcal{L}_{bcc}$. The strongly augmented target data with *red flow*, which is used to measure the distance between predictions of the weak augmented one via $\ominus$ unit. We rank multiple IC matrices to obtain the weight mask, and calculate the reweighted implicit smoothness constraints loss $\mathcal{L}_{rsc}$ via $\odot$ unit. (Best viewed in color)

where the K is the class number in both two domains. So, we obtain the **C**lass **C**onfusion (CC) matrix $\tilde{C}$, where the class confusion between class i and class j is $\tilde{C}_{i,j}$.

Minimizing all cross-class confusion can be achieved via the **B**atch **C**lass **C**onfusion (BCC) loss as below:

$$\mathcal{L}_{bcc} = \frac{1}{K}\sum_{i=1}^{K}\sum_{\substack{j\neq i\\ j=1}}^{K}\tilde{C}_{i,j} \tag{6}$$

Combining the cross entropy loss on labeled source domain and the batch class confusion loss leads to surprising performances in several public datasets.

3.2. Implicit Smoothness Constraint

We notice that the MCC [14] has the problem of misclassification accumulation. The core problem is that the model is sensitive to image perturbations in target domain, and it is prone to misclassify images of the same class to different classes. Therefore, to reduce the sensitivity of the model, we directly impose the model to generate more consistent predictions from original and perturbed images. Although the idea is simple, it can effectively contribute to the more consistent predictions of different samples within a class.

Assume that target batch samples $X^t = \{x_i^t\}_{i=1}^b$ are fed into the model. Firstly, they pass through the weak augmentation A^w and strong augmentation A^r to generate the two augmented images W^t and R^t via $W^t = A^w(X^t)$ and $R^t = A^r(X^t)$. The strong augmentation is adopted by RandAug [4] with randomly selected n of the 14 fixed transformations and the specific magnitude m.

Then, both the two augmented batches are passed through the feature extractor and classifier to generate the classification responses respectively, as follows:

$$\hat{Y}^{tW} = \sigma\left(\mathcal{F}(\mathcal{G}(W^t))\right) \tag{7}$$

$$\hat{Y}^{tR} = \sigma\left(\mathcal{F}(\mathcal{G}(R^t))\right) \tag{8}$$

The difference between $\hat{Y}^{tW}$ and $\hat{Y}^{tR}$ for a target sample X_i^t can be measured as below:

$$D_i = \frac{1}{K}\sum_{j=1}^{K}\left\|\hat{Y}_{ij}^{tW} - \hat{Y}_{ij}^{tR}\right\|_2 \tag{9}$$

where the $\hat{Y}_{ij}^{tW}$ is the probability that i-th weakly augmented instance belongs to the j-th class, and $\hat{Y}_{ij}^{tR}$ corresponds to the strongly augmented instance.

We aim to reduce the sensitivity of model over sample perturbations, alleviating the misclassification accumulation. We propose the implicit **S**moothness **C**onstraint (SC) loss defined as below:

$$\mathcal{L}_{sc} = \frac{1}{b}\sum_{i=1}^{b}D_i \tag{10}$$

where the b is the sample number of a batch data, and the SC loss is defined as the average difference of instance level.

3.3. Weighting Mechanism with Instance Class Confusion

Do we really need to impose such constraints on all samples? In fact, samples are not equally important. Some methods like CDAN [23] and MCC [14] used the entropy of samples to reweight the loss function. In order to better prevent misclassification, we should pay more attention to reliable samples. We noticed that the class confusion of a sample can be a measure of the uncertainty of it. Given a target sample X_i^t and the corresponding classification response $\tilde{Y}_i^t$ after temperature rescaling, the **I**nstance class **C**onfusion (IC) matrix of i-th instance is defined as below:

$$IC_{i(p,q)} = \tilde{Y}_{i,p}^t \cdot \tilde{Y}_{i,q}^t \tag{11}$$

where $\tilde{Y}_{i,p}^t$ represents the probability that i-th instance belongs to p-th class. We sum the off-diagonal elements to

get the confusion value at the instance level. The instance confusion value of i-th instance is defined as below:

$$\mathcal{V}_i = \sum_{p=1}^{K} \sum_{\substack{q=1 \\ q \neq p}}^{K} IC_{i(p,q)} \tag{12}$$

The confusion value reflects the uncertainty of the sample. And then we rank the instance confusion values to get the ascendingly sorted indexes array $Id = argsort\{\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_b\}$. We construct a mask M to reweight the implicit smoothness constraint loss, and the mask values are defined as follows:

$$M_i = \begin{cases} 1, & if\ i \in \{\underbrace{Id_1, Id_2, \cdots, Id_{N_m}}_{N_m}\} \\ 0, & else \end{cases} \tag{13}$$

where the N_m represents the total number of elements in the mask whose value is 1. The N_m is the crucial hyperparameter in our method. The **R**eweighted implicit **S**moothness **C**onstraint (RSC) loss is defined as below:

$$\mathcal{L}_{rsc} = \frac{1}{N_m} \sum_{i=1}^{b} D_i \cdot M_i \tag{14}$$

So, given the labeled source data and unlabeled target data, we integrate the classification loss $\mathcal{L}_s$, the Batch Class Confusion (BCC) loss $\mathcal{L}_{bcc}$, and the Reweighted implicit Smoothness Constraint (RSC) loss $\mathcal{L}_{rsc}$ together. The final objective is defined as follow:

$$\mathcal{L} = \mathcal{L}_s + \alpha \cdot \mathcal{L}_{bcc} + \beta \cdot \mathcal{L}_{rsc} \tag{15}$$

where α and β are trade-off hyperparameters between different loss functions, and the overall networks $\{\mathcal{G}, \mathcal{F}\}$ are optimized by minimizing the final objective.

3.4. Insight Analysis

We give an analysis of our method following the $\mathcal{H}\Delta\mathcal{H}$ theory [2]. Our method is similar to MCD [31], and tries to minimize the $\mathcal{H}\Delta\mathcal{H}$ divergence between source and target domains. With random image augmentations, the predictions of the network change. It can be seen as we are given two different hypotheses over a single sample by weak and strong augmentations, respectively. Due to the randomness of strong augmentations, we can widely search the upper bound of $\mathcal{H}\Delta\mathcal{H}$ divergence, by calculating the disagreements of weak hypothesis and strong hypothesis on target samples. And we minimize every disagreement of two hypotheses during training. Additionally, the more reliable target samples selected by weak hypothesis are useful in approximating the $\mathcal{H}\Delta\mathcal{H}$ divergence, which bounds the target error more effectively.

4. Experiments

4.1. Setup

Datasets. We use four standard benchmarks for unsupervised domain adaptation to evaluate various methods.

DomainNet [27] is the largest and hardest benchmark for UDA by far. It contains approximately 0.6 million images with 345 categories and 6 domains: *Clipart* (Clp), *Infograph* (Inf), *Painting* (Pnt), *Quickdraw* (Qdr), *Real* (Rel), and *Sketch* (Skt). Here we focus on the tasks between Clp, Pnt, Rel, and Skt, following the settings in [13], which contains 12 relatively challenging transfer tasks.

VisDA-C [28] is a large-scale benchmark that contains the images from 12 categories of two very distinct domains: *synthetic* domain and *real-world* domain. The synthetic domain contains 152,397 images and the latter contains 55,388 images, and we focus on the synthetic-to-real transfer task.

Office-Home [37] is a more difficult benchmark that consists of images from four domains: *Art* (Ar), *Clipart* (Cl), *Product* (Pr), and *Real-world* (Rw), totally around 15,500 images from 65 different categories. All 12 transfer tasks are selected for evaluation.

Office-31 [29] is a classic benchmark with 31 categories and three domains: *Amazon* (A) with 2,817 images, *Dslr* (D) 498 images, and *Webcam* (W) with 795 images. This benchmark contains 6 transfer tasks.

Implementation Details. These methods are implemented based on **PyTorch**, and for a fair comparison, we use the ResNet-50/ResNet-101 [12] as the backbone of network, fixing the batch-size at 36, α at 1, and N_m at 20 for all experiments. For the crucial hyperparameter β, we use the DEV [41] to select the optimal value for each dataset. We set β as 60 for DomainNet, 50 for VisDA-C, and 20 for Office-Home and Office-31. We use ResNet-101 as the backbone for DomainNet and VisDA-C, and ResNet-50 for Office-Home and Office-31. For training convergence, we set empirical settings following the previous works [13, 21], including SGD optimizer, learning rate scheduler, momentum (0.9), weight decay ($1e^{-3}$), and bottleneck size (256). We report the accuracy tested after all iterations are done, and our method is named **ISC** (**I**mplicit **S**moothness **C**onstraints) in the following experimental results. And our codebase will be released.

4.2. Results

DomainNet. Following the settings in [13], we compare various methods for 12 tasks between Clp, Pnt, Rel, and Skt domains on original DomainNet. As shown in Tab. 1, in the such challenging dataset, results of the domain alignment based method do not differ too much. However, our method significantly outperforms the compared methods, and achieves the state-of-the-art results. It is worth noting

Table 1. Accuracy (%) On DomainNet for vanilla UDA using the ResNet-101 backbone. All methods follow the same settings, so the results are directly borrowed from [13]. †: Reproduced result from our codebase. Best (**bold red**), second best (*italic blue*)

Method	Clp→Pnt	Clp→Rel	Clp→Skt	Pnt→Clp	Pnt→Rel	Pnt→Skt	Rel→Clp	Rel→Pnt	Rel→Skt	Skt→Clp	Skt→Pnt	Skt→Rel	Avg
ResNet-101 [12]	32.7	50.6	39.4	41.1	56.8	35.0	48.6	48.8	36.1	49.0	34.8	46.1	43.3
DANN [7]	37.9	54.3	44.4	41.7	55.6	36.8	50.7	50.8	40.1	55.0	45.0	54.5	47.2
BCDM [19]	38.5	53.2	43.9	42.5	54.5	38.5	51.9	51.2	40.6	53.7	46.0	53.4	47.3
MCD [31]	37.5	52.9	44.0	44.6	54.5	41.6	52.0	51.5	39.7	55.5	44.6	52.0	47.5
ADDA [34]	38.4	54.1	44.1	43.5	56.7	39.2	52.8	51.3	40.9	55.0	45.4	54.5	48.0
DAN [22]	38.8	55.2	43.9	45.9	59.0	40.8	50.8	49.8	38.9	56.1	45.9	55.5	48.4
MCC [14]	37.7	55.7	42.6	45.4	*59.8*	39.9	54.4	53.1	37.0	58.1	46.3	*56.2*	48.9
CDAN [23]	40.4	*56.8*	46.1	45.1	58.4	40.5	55.6	53.6	43.0	57.2	46.4	55.7	49.9
JAN [25]	40.5	56.7	45.1	47.2	**59.9**	*43.0*	54.2	52.6	41.9	56.6	46.2	55.5	50.0
MDD [43]	*42.9*	**59.5**	*47.5*	*48.6*	59.4	42.6	*58.3*	*53.7*	*46.2*	*58.7*	*46.5*	**57.7**	*51.8*
MCC† [14]	40.1	56.5	44.9	46.9	57.7	41.4	56.0	*53.7*	40.6	58.2	45.1	55.9	49.7
ISC (Ours)	**44.1**	55.3	**48.5**	**49.4**	57.5	**45.5**	**58.8**	**55.4**	**46.8**	**61.3**	**51.1**	**57.7**	**52.6**

Table 2. Accuracy (%) On VisDA-C for vanilla UDA using the ResNet-101 backbone. All methods follow the same settings, so the results are directly copied from the original works or borrowed from [21]. †: Reproduced result from our codebase. Best (**bold red**), second best (*italic blue*)

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	trunk	Mean
ResNet-101 [12]	67.7	27.4	50.0	61.7	69.5	13.7	85.9	11.5	64.4	34.4	84.2	19.2	49.1
DANN [7]	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
DAN [22]	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
MCD [31]	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
CDAN [23]	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
STAR [26]	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
EntMin [9]	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
BNM [5]	91.1	69.0	76.7	64.3	89.8	61.2	*90.8*	74.8	90.9	66.6	*88.1*	46.1	75.8
AFN [39]	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
MCC [14]	88.1	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
ATDOC [21]	93.7	83.0	76.9	58.7	89.7	95.1	84.4	71.4	89.4	80.0	86.7	55.1	80.3
DTA [17]	93.7	82.2	**85.6**	*83.8*	93.0	81.0	90.7	82.1	*95.1*	78.1	86.4	32.1	81.5
SHOT [20]	94.3	**88.5**	80.1	57.3	93.1	94.9	80.7	80.3	91.5	*89.1*	86.3	**58.2**	82.9
BCDM [19]	*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*
MCC† [14]	94.7	82.0	74.2	60.1	92.5	**96.7**	84.9	81.7	88.4	86.6	80.9	*57.0*	81.6
ISC (Ours)	**96.8**	*88.3*	84.5	**89.5**	**96.1**	*96.1*	90.1	**84.0**	**95.5**	**92.0**	83.0	40.7	**86.4**

that we achieve the best accuracy on 10 out of 12 tasks, although all 12 tasks are more challenging.

As shown in the second part in Tab. 1, we reproduce the results of MCC as the MCC†, and the results are better than that in [13]. It is worth noting that our method brings substantial improvements on most of the tasks against MCC, which demonstrates the effectiveness of our method. Specifically, compared with the reproduced MCC, our method surprisingly obtains more than 4% accuracy improvement on tasks including Clp→Pnt, Pnt→Skt, Rel→Skt, and Skt→Pnt. Compared with the results of MDD, which is based on domain alignment and shows the superiority on this dataset, our method surpasses it on most of tasks, and achieve the best average accuracy up to **52.6**%. The promising results in such a hard dataset with 345 categories demonstrate the effectiveness of our method, and the potential for large categories classification.

VisDA-C. We focus on the classification accuracy in the synthetic-to-real transfer task. As shown in Tab. 2, we compare various methods with ResNet-101 backbone. The domain alignment based methods including DANN [7], DAN [22], MCD [31], CDAN [23], and STAR [26] perform much better benefited from their carefully designed metrics or training patterns to minimize the discrepancy between source and target domain. And those regularization based methods including BNM [5], AFN [39], MCC [14], ATDOC [21], DTA [17], SHOT [20], and BCDM [19] obtain more promising average accuracy.

All methods follow the same settings, and MCC† represents the reproduced accuracy. Our method brings 4.8% improvement against MCC, and achieves 86.4% accuracy on average, which demonstrates the effectiveness of our

Table 3. Accuracy (%) On Office-Home for vanilla UDA using the ResNet-50 backbone. All methods follow the same settings, so the results are directly copied from the original works or borrowed from [21]. †: Reproduced result from our codebase. Best (**bold red**), second best (*italic blue*)

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
ResNet-50 [12]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [22]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [7]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [25]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [23]	54.6	74.1	78.1	63.0	72.2	74.1	61.6	52.3	79.1	72.3	57.3	82.8	68.5
EntMin [9]	51.0	71.9	77.1	61.2	69.1	70.1	59.3	48.7	77.0	70.4	53.0	81.0	65.8
AFN [39]	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
MCC [14]	56.3	77.3	80.3	67.0	77.1	77.0	66.2	55.1	81.2	73.5	57.4	84.1	71.0
BNM [5]	56.7	77.5	81.0	67.3	76.3	77.1	65.3	55.1	82.0	73.6	57.0	84.3	71.1
SHOT [20]	57.1	78.1	81.5	68.0	**78.2**	*78.1*	*67.4*	54.9	82.2	73.3	58.8	84.3	71.8
ATDOC [21]	*58.3*	*78.8*	**82.3**	**69.4**	**78.2**	**78.2**	67.1	*56.0*	**82.7**	72.0	58.2	*85.5*	*72.2*
MCC† [14]	57.6	76.8	*81.8*	67.9	*77.5*	77.8	66.9	55.3	82.1	*75.5*	*61.9*	85.2	*72.2*
ISC (Ours)	**59.2**	**79.4**	**82.3**	*69.1*	76.6	**78.2**	**68.2**	**57.6**	*82.6*	**75.7**	**63.0**	**85.8**	**73.1**

Table 4. Accuracy (%) on Office-31 for vanilla UDA using the ResNet-50 backbone. All methods follow the same settings, so the results are directly copied from the original works or borrowed from [21]. Best (**bold red**), second best (*italic blue*)

Method	A→D	A → W	D→A	D→W	W→A	W→D	Avg
ResNet-50 [12]	78.3	70.4	57.3	93.4	61.5	98.1	76.5
DAN [22]	78.6	80.5	63.6	97.1	62.8	99.6	80.4
DANN [7]	79.7	82.0	68.2	96.9	67.4	99.1	82.2
MCD [31]	92.2	88.6	69.5	98.5	69.7	**100.0**	86.5
CDAN [23]	92.9	94.1	71.0	*98.6*	69.3	**100.0**	87.7
MDD [43]	93.5	94.5	74.6	98.4	72.2	**100.0**	89.4
EntMin [9]	86.0	87.9	67.0	98.4	63.7	**100.0**	83.8
AFN [39]	87.7	88.8	69.8	98.4	69.7	99.8	85.7
BNM [5]	90.3	91.5	70.9	98.5	71.6	**100.0**	87.1
SHOT [20]	94.0	90.1	*74.7*	98.4	74.3	*99.9*	88.6
BCDM [19]	93.8	95.4	73.1	*98.6*	73.0	**100.0**	89.0
MCC [14]	**95.6**	*96.1*	73.5	98.1	73.6	**100.0**	89.5
ATDOC [21]	94.4	94.3	**75.6**	**98.9**	*75.2*	99.6	*89.7*
ISC (Ours)	*95.4*	**96.2**	73.8	98.5	**75.9**	**100.0**	**90.0**

method. It is worth noting that our method performs against other approaches by large margins on 6 out of 12 categories, and outperforms the BCDM by 3.0% accuracy on average. Compared with the baseline MCC, we found that our method achieves surprising improvements in most of tasks. The results prove that our method could also maintain high performances when there are few categories.

Office-Home. We evaluated various methods on total of 12 tasks. As shown in Tab. 3, most of the regularization based methods including MCC [14], BNM [5], SHOT [20], and ATDOC [21] perform better than domain alignment based methods. All methods follow the same settings, and we also reproduced the results of MCC as MCC†, which achieved 72.2% accuracy on average. Our method achieves 73.1% accuracy, and perform the best accuracy on 9 out of 12 tasks. Besides, our method obtains much improvement on the difficult scenarios such as Ar→Cl and Pr→Cl.

Office-31. We compare various methods in this classic dataset, as shown in Tab. 4. Domain alignment based methods obtain results comparable to those of regularization based methods including MCC. However, our method brings 0.5% accuracy improvement against MCC. Our method achieves the best accuracy on A→W, W→A, and W→D, and obtains 90% average accuracy.

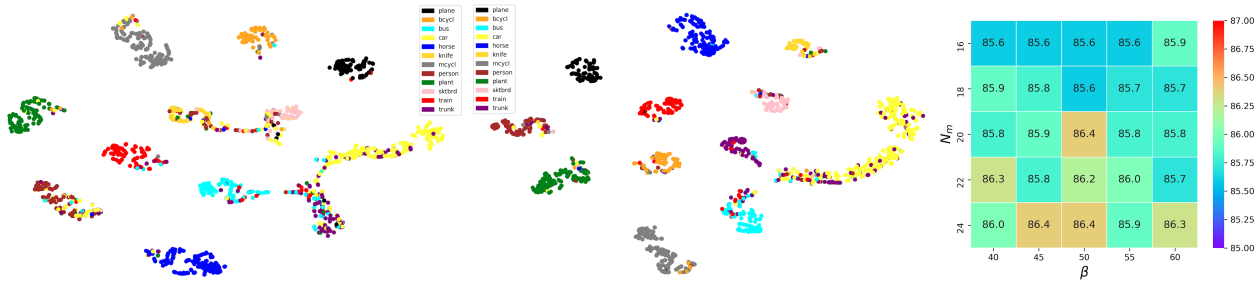
The above experimental results show our methods surprisingly achieves promising results on various datasets, and demonstrate the advantage and necessity of considering the smoothness constraints to models.

4.3. Empirical Analysis

Ablation Study. We conduct the ablation study in VisDA-C [28] to investigate the necessity of the implicit smoothness constraints and weight mechanism in our method. The results are shown in Tab. 5. Here we use SC to represent the implicit smoothness constraints without any weighting mechanism, and EW denotes the entropy weighting mechanism, and IW denotes the instance class confusion weighting mechanism. The MCC with only SC obtains substantial improvements on this dataset, which demonstrates that our designed implicit smoothness constraints are effective to correct the misclassification accumulation. Furthermore, consider the weighting mechanism, and we evaluated SC with different weighting mechanisms from entropy or instance class confusion. The results show that SC with EW obtains 0.5% improvement on average accuracy. But SC with IW outperforms the original SC by 1.4% margin. Our full method performs the best accuracy on 8 out of 12 categories. So it verifies our claim that instance class confusion is a better measure of the uncertainty. Clearly, our main contributions, i.e., implicit smoothness constraints and instance class confusion weighting mechanism, show promising benefits for unsupervised domain adaptation.

Table 5. Ablation study On VisDA-C for vanilla UDA using the ResNet-101 backbone. †: Reproduced result from our codebase. EW: Entropy Weight mechanism, IW: Instance class confusion Weight mechanism. Best (**bold red**).

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	trunk	Mean
MCC† [14]	94.7	82.0	74.2	60.1	92.5	96.7	84.9	81.7	88.4	86.6	80.9	**57.0**	81.6
MCC + SC	96.5	82.7	79.8	79.2	95.1	**97.4**	91.0	**86.7**	93.9	89.8	82.2	45.7	85.0
MCC + SC+ EW	96.5	83.3	82.2	84.8	95.6	**97.4**	**91.5**	**86.7**	94.6	89.8	82.5	40.8	85.5
MCC + SC+ IW	**96.8**	**88.3**	**84.5**	**89.5**	**96.1**	96.1	90.1	84.0	**95.5**	**92.0**	**83.0**	40.7	**86.4**



(a) t-SNE: MCC (b) t-SNE: Ours (c) Accuracy sensitivity w.r.t β and N_m

Figure 3. (a) and (b) correspond to the t-SNE embedding visualization of MCC and our models on VisDA-C. Different colors indicate different categories. (c): The accuracy sensitivity of our method with regard to hyperparameters β and N_m. (Best viewed in color)

Feature Visualization. To better illustrate that our method can improve the discriminability of target features, we visualize the features learned by MCC and our method on VisDA-C. For feature visualization, we employ the t-SNE method [36]. Here we randomly select 2000 samples across 12 categories from real-world domain in VisDA-C. As shown in Fig. 3, compare with the MCC, our method better separates the target samples in the feature space. We can observe that the features within the classes of our method are more centralized and compact, and the feature distances between classes are increased. And the misclassification of features in every cluster is reduced. Especially in ***plane*** and *horse* clusters, our method achieves almost perfect classification. So, the feature visualization result proves that, compare with MCC, our method can reduce the misclassification accumulation effectively, thereby the model generalizes better in target domain.

Parameter Sensitivity. We aim to analyze the parameter sensitivity of proposed implicit smoothness constraints on VisDA-C. Two hyperparameters are crucial to the effectiveness of our method. One is N_m, the total number of valid values in weight mask, which controls the weighting of implicit smoothness constraints, and the other is β, which controls the importance of entire smoothness constraints loss. The α is fixed at 1, and we perform a grid search for N_m and β. We empirically choose a representative stable region for each parameter, $\{16, 18, 20, 22, 24\}$ for N_m and $\{40, 45, 50, 55, 60\}$ for β. As shown in Fig. 3c, our method is not very sensitive to both two hyperparameters and maintains fair performances when β and N_m are relatively large. We find that when N_m is relatively low, the accuracy is around 85.6%, and when N_m is not lower than 20, our method achieves better accuracies. And we can see that appropriate combination of β and N_m contributes to good transfer performance in UDA, which justifies the motivation of imposing the instance class confusion as the weighting mechanism for smoothness constraints.

5. Limitations

In our method, we introduce the strong perturbation to target samples to construct the implicit smoothness constraints. However, the perturbed samples may not be able to simulate sample differences within each category on the target domain, such that our constraints may not help. For example, our approach brings marginal improvement on Office-Home due to the larger differences of samples within each category and the wide variation between domains.

6. Conclusion

In this paper, we aim to handle the accumulation of misclassification issue of regularization methods for UDA. Accordingly, we design an implicit smoothness constraint based on the Minimum Class Confusion framework, which contributes to reducing the model sensitivity to perturbations for each target sample. Furthermore, we utilize the instance class confusion as the weight to enhance the effectiveness of the constraints. Extensive experiments show that our method surprisingly yields the results surpassing or comparable to the state-of-the-arts on four public datasets.

References

[1] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 3

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 2, 5

[3] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5050–5060, 2019. 2, 3

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 4

[5] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3941–3950, 2020. 1, 3, 6, 7

[6] Geoffrey French, Michal Mackiewicz, and Mark H. Fisher. Self-ensembling for visual domain adaptation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 1, 3

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189, 2015. 2, 6, 7

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[9] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005. 1, 2, 3, 6, 7

[10] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19:513–520, 2006. 2

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 6, 7

[13] Junguang Jiang, Baixu Chen, Bo Fu, and Mingsheng Long. Transfer-learning-library. https://github.com/thuml/Transfer-Learning-Library, 2020. 5, 6

[14] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *European Conference on Computer Vision*, pages 464–480. Springer, 2020. 1, 2, 3, 4, 6, 7, 8

[15] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *European Conference on Computer Vision*, pages 591–607. Springer, 2020. 2

[16] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 3

[17] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 91–100, 2019. 6

[18] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020. 2

[19] Shuang Li, Fangrui Lv, Binhui Xie, Chi Harold Liu, Jian Liang, and Chen Qin. Bi-classifier determinacy maximization for unsupervised domain adaptation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 8455–8464. AAAI Press, 2021. 6, 7

[20] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039, 2020. 2, 3, 6, 7

[21] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2021. 5, 6, 7

[22] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105, 2015. 2, 3, 6, 7

[23] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1647–1657, 2018. 1, 2, 3, 4, 6, 7

[24] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information*

Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 136–144, 2016. 2

[25] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217, 2017. 2, 3, 6, 7

[26] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9111–9120, 2020. 6

[27] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019. 2, 5

[28] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 1, 2, 3, 5, 7

[29] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 2, 5

[30] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019. 2

[31] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018. 2, 5, 6, 7

[32] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000. 1, 3

[33] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2, 3

[34] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 6

[35] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2

[36] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8

[37] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 2, 5

[38] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 3

[39] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1426–1435, 2019. 1, 3, 6, 7

[40] Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2142–2150, 2015. 2

[41] Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 7124–7133. PMLR, 2019. 5

[42] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 2

[43] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413, 2019. 1, 2, 3, 6, 7

[44] Junbao Zhuo, Shuhui Wang, Shuhao Cui, and Qingming Huang. Unsupervised open domain recognition by semantic discrepancy minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 750–759, 2019. 3