Minimum Class Confusion for Versatile Domain Adaptation

Ying Jin, Ximei Wang, Mingsheng Long (✉), and Jianmin Wang

School of Software, BNRist, Research Center for Big Data, Tsinghua University, China
{jiny18,wxm17}@mails.tsinghua.edu.cn, {mingsheng,jimwang}@tsinghua.edu.cn

Abstract. There are a variety of Domain Adaptation (DA) scenarios subject to label sets and domain configurations, including closed-set and partial-set DA, as well as multi-source and multi-target DA. It is notable that existing DA methods are generally designed only for a specific scenario, and may underperform for scenarios they are not tailored to. To this end, this paper studies **Versatile Domain Adaptation (VDA)**, where one method can handle several different DA scenarios without any modification. Towards this goal, a more general inductive bias other than the *domain alignment* should be explored. We delve into a missing piece of existing methods: *class confusion*, the tendency that a classifier confuses the predictions between the correct and ambiguous classes for target examples, which is common in different DA scenarios. We uncover that reducing such pairwise class confusion leads to significant transfer gains. With this insight, we propose a general loss function: **Minimum Class Confusion (MCC)**. It can be characterized as (1) a *non-adversarial* DA method without explicitly deploying domain alignment, enjoying faster convergence speed; (2) a *versatile* approach that can handle four existing scenarios: Closed-Set, Partial-Set, Multi-Source, and Multi-Target DA, outperforming the state-of-the-art methods in these scenarios, especially on one of the largest and hardest datasets to date (7.3% on DomainNet). Its versatility is further justified by two scenarios proposed in this paper: Multi-Source Partial DA and Multi-Target Partial DA. In addition, it can also be used as a general regularizer that is orthogonal and complementary to a variety of existing DA methods, accelerating convergence and pushing these readily competitive methods to stronger ones. Code is available at https://github.com/thuml/Versatile-Domain-Adaptation.

Keywords: Versatile Domain Adaptation, Minimum Class Confusion

1 Introduction

The scarcity of labeled data hinders deep neural networks (DNNs) from use in real applications. This challenge gives rise to Domain Adaptation (DA) [34,28], an important technology that aims to transfer knowledge from a labeled source domain to an unlabeled target domain in the presence of dataset shift. A rich line of DNN-based methods [44,21,23,24,49,42,8,43,30,22,47,53] have been proposed for Unsupervised DA (UDA), a closed-set scenario with one source domain and

UDA	PDA	MSDA	MTDA	MSPDA	MTPDA

Minimum Class Confusion (MCC), a general loss function for Versatile Domain Adaptation (VDA)
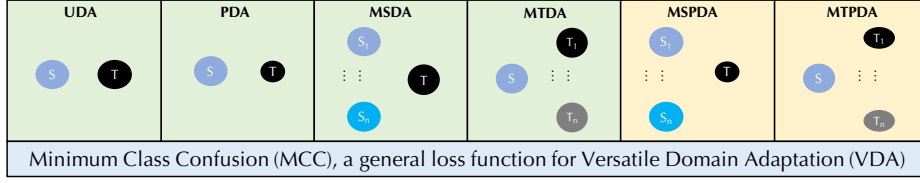
Fig. 1: **Versatile Domain Adaptation (VDA)** subsumes typical domain adaptation scenarios: (1) Unsupervised Domain Adaptation (UDA); (2) Partial Domain Adaptation (PDA); (3) Multi-Source Domain Adaptation (MSDA); (4) Multi-Target Domain Adaptation (MTDA); (5) Multi-Source Partial Domain Adaptation (MSPDA); (6) Multi-Target Partial Domain Adaptation (MTPDA). Note that scenarios (5)–(6) are newly proposed in this paper. Our **Minimum Class Confusion (MCC)** is a *versatile* method towards all these DA scenarios.

one target domain sharing the same label set. Recently, several highly practical scenarios were proposed, such as Partial DA (PDA) [2,51] with the source label set subsuming the target one, Multi-Source DA (MSDA) [54,48] with multiple source domains, and Multi-Target DA (MTDA) [32] with multiple target domains. As existing UDA methods cannot be applied directly to these challenging scenarios, plenty of methods [2,3,51,48,32] have been designed for each specific scenario, which work quite well in each tailored scenario.

In practical applications, however, it is difficult to confirm the label sets and domain configurations in the data acquisition process. Therefore, we may be stuck in choosing a proper method tailored to the suitable DA scenario. The most ideal solution to escape from this dilemma is a *versatile* DA method that can handle various scenarios without any modification. Unfortunately, existing DA methods are generally designed only for a specific scenario and may underperform for scenarios they are not tailored to. For instance, PADA [3], a classic PDA method, excels at selecting out outlier classes but suffers from the internal domain shift in MSDA and MTDA, while DADA [32], an outstanding method tailored to MTDA, cannot be directly applied to PDA or MSDA. Hence, existing DA methods are not versatile enough to handle practical scenarios of complex variations.

In this paper, we define **Versatile Domain Adaptation (VDA)** as a line of *versatile* approaches able to tackle a variety of scenarios without any modification. Towards VDA, a more general inductive bias other than the domain alignment should be explored. In this paper, we delved into the error matrices of the target domain and found that the classifier trained on the source domain may confuse to distinguish the correct class from a similar class, such as `cars` and `trucks`. As shown in Fig. 2(b), the probability that a source-only model misclassifies `cars` as `trucks` on the target domain is over 25%. Further, we analyzed the error matrices in other DA scenarios and reached the same conclusion. These findings give us a fresh perspective to enable VDA: **class confusion**, the tendency that a classifier confuses the predictions between the correct and ambiguous classes

(a) Source   (b) Target   (c) MinEnt   (d) MCC
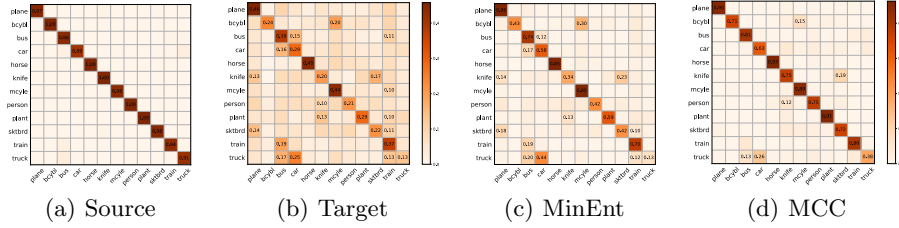
Fig. 2: The error matrices of several models on VisDA-2017 [33]. (a)–(b): Source-only model tested on the source and target domains, showing severe class confusion on target domain examples. (c)–(d): Models trained with entropy minimization (MinEnt) [10] and Minimum Class Confusion (MCC) on target domain examples, respectively. The proposed MCC loss substantially diminishes the class confusion.

for target examples. We uncover that less class confusion leads to more transfer gains for all the domain adaptation scenarios in Fig. 1.

Still, we need to address a new challenge that the ground-truth class confusion cannot be calculated if the labels in the target domain are inaccessible. Fortunately, the confusion between different classes can be naturally reflected by an example-weighted inner-product between the classifier predictions and their transposes. And we can define class confusion from this perspective, enabling it to be computed from well-calibrated classifier predictions. To this end, we propose a novel loss function: **Minimum Class Confusion (MCC)**. It can be characterized as a novel and versatile DA approach without explicitly deploying domain alignment [21,8], enjoying fast convergence speed. In addition, it can also be used as a general regularizer that is orthogonal and complementary to existing DA methods, further accelerating and improving those readily competitive methods. Our contributions are summarized as follows:

– We propose a practical setting, **Versatile Domain Adaptation (VDA)**, where one method can tackle many DA scenarios without modification.

– We uncover that the class confusion is a common missing piece of existing DA methods and that less class confusion leads to more transfer gains.

– We propose a novel loss function: **Minimum Class Confusion (MCC)**, which is versatile to handle four existing DA scenarios, including closed-set, partial-set, multi-source, and multi-target, as well as two proposed scenarios: multi-source partial DA and multi-target partial DA.

– We conduct extensive experiments on four standard DA datasets, and show that MCC outperforms the state-of-the-art methods in different DA scenarios, especially on one of the largest and hardest datasets (**7.3**% on DomainNet), enjoying a faster convergence speed than mainstream DA methods.

2   Related Work

Unsupervised Domain Adaptation (UDA). Most of the existing domain adaptation researches focused on UDA, in which numerous UDA methods were proposed based on either *Moment Matching* or *Adversarial Training*.

Moment Matching methods aim at minimizing the distribution discrepancy across domains. Deep Coral [40] aligns second-order statistics between distributions. DDC [44] and DAN [21] utilize Maximum Mean Discrepancy [11], JAN [24] defines Joint Maximum Mean Discrepancy, SWD [18] introduces Sliced Wasserstein Distance and CAN [17] leverages Contrastive Domain Discrepancy.

Adversarial Training methods were inspired by the Generative Adversarial Networks (GANs) [9], aiming at learning domain invariant features in an adversarial manner. DANN [8] introduces a domain discriminator to distinguish source and target features, while the feature extractor strives to fool it. ADDA [43], MADA [30] and MCD [36] extend this architecture to multiple feature extractors and classifiers. Motivated by Conditional GANs [27], CDAN [22] aligns domain features in a class-conditional adversarial game. CyCADA [15] adapts features in both pixel and feature levels. TADA [47] introduces the first transferable attention mechanism. SymNet [52] uses a symmetric classifier, and DTA [19] learns discriminative features with a new adversarial dropout.

There are other approaches to domain adaptation. For instance, SE [7] is based on the teacher-student [41] architecture. TransNorm [46] tackles DA with a new transferable backbone. TAT [20] proposes transferable adversarial training to guarantee the adaptability. BSP [5] balances between the transferability and discriminability. AFN [49] enlarges feature norm to enhance feature transferability. Some methods [39,16,55,56] also utilize the less-reliable self-training or pseudo labeling, *e.g.* TPN [29] is based on pseudo class-prototypes.

Partial Domain Adaptation (PDA). In PDA, the target label set is a subset of the source label set. Representative methods include SAN [2], IWAN [51], PADA [3] and ETN [4], introducing different weighting mechanisms to select out outlier source classes in the process of domain feature alignment.

Multi-Source Domain Adaptation (MSDA). In MSDA, there are multiple source domains of different distributions. MDAN [54] provides theoretical insights for MSDA, while Deep Cocktail Network [48] (DCTN) and M^3SDA [31] extend adversarial training and moment-matching to MSDA, respectively.

Multi-Target Domain Adaptation (MTDA). In MTDA, we need to transfer a learning model to multiple unlabeled target domains. DADA [32] is the first approach to MTDA through disentangling domain-invariant representations.

3   Approach

In this paper, we study **Versatile Domain Adaptation (VDA)** where one method can tackle many scenarios without any modification. We justify the versatility of one method by four existing scenarios: **(1)** Unsupervised Domain Adaptation (**UDA**) [8], the standard scenario with a labeled source domain
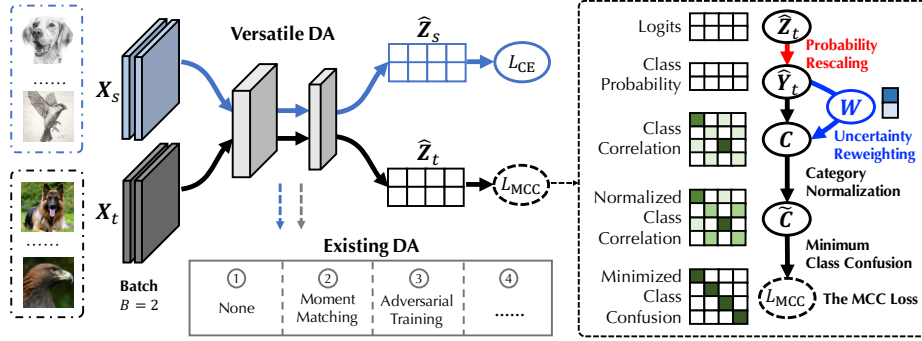
Fig. 3: The schematic of the **Minimum Class Confusion (MCC)** loss function. Given the shared feature extractor F, MCC is defined on the class predictions $\widehat{\mathbf{Y}}_t$ given by the source classifier G on the target data. MCC is versatile to address various domain adaptation scenarios standalone, or to be integrated with existing methods (moment matching, adversarial training, etc). (*Best viewed in color.*)

$\mathcal{S} = \{(\mathbf{x}_s^i, \mathbf{y}_s^i)\}_{i=1}^{n_s}$ and an unlabeled target domain $\mathcal{T} = \{\mathbf{x}_t^i\}_{i=1}^{n_t}$, where $\mathbf{x}^i$ is an example and $\mathbf{y}^i$ is the associated label; **(2)** Partial Domain Adaptation (**PDA**) [3], which extends UDA by relaxing the source domain label set to subsume the target domain label set; **(3)** Multi-Source Domain Adaptation (**MSDA**) [31], which extends UDA by expanding to S labeled source domains $\{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_S\}$; **(4)** Multi-Target Domain Adaptation (**MTDA**) [32], which extends UDA by expanding to T unlabeled target domains $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T\}$. We further propose two scenarios to confirm the versatility: **(5)/(6)** Multi-Source/Multi-Target Partial Domain Adaptation (**MSPDA/MTPDA**), which extend PDA to multi-source/multi-target scenarios. Tailored to a specific scenario, existing methods fail to readily handle these scenarios. We propose **Minimum Class Confusion (MCC)** as a generic loss function for VDA. Hereafter, we denote by $\mathbf{y}_{i\cdot}$, $\mathbf{y}_{\cdot j}$ and $\mathbf{Y}_{ij}$ the i-row, the j-th column and the ij-th entry of matrix $\mathbf{Y}$, respectively.

3.1 Minimum Class Confusion

To enable versatile domain adaptation, we need to find out a proper criterion to measure the pairwise class confusion on the target domain. Unlike previous methods such as CORAL [40] that focus on features, we explore the class predictions. Denote the classifier output on the target domain as $\widehat{\mathbf{Y}}_t = G(F(\mathbf{X}_t)) \in \mathbb{R}^{B \times |\mathcal{C}|}$, where B is the batch size of the target data, $|\mathcal{C}|$ is the number of source classes, F is the feature extractor and G is the classifier. In our method, we focus on the classifier predictions $\widehat{\mathbf{Y}}$ and omit the domain subscript t for clarity.

Probability Rescaling. According to [12], DNNs tend to make overconfident predictions, hindering them from directly reasoning about the class confusion. Therefore, we adopt temperature rescaling [14,12], a simple yet effective technique,

to alleviate the negative effect of overconfident predictions. Using temperature scaling, the probability $\widehat{Y}_{ij}$ that the i-th instance belongs to the j-th class can be recalibrated as

$$\widehat{Y}_{ij} = \frac{\exp\left(Z_{ij}/T\right)}{\sum_{j'=1}^{|\mathcal{C}|}\exp\left(Z_{ij'}/T\right)}, \tag{1}$$

where Z_{ij} is the logit output of the classifier layer and T is the temperature hyper-parameter for probability rescaling. Obviously, Eq. (1) boils down to the vanilla softmax function when $T = 1$.

Class Correlation. As $\widehat{Y}_{ij}$ reveals the relationship between the i-th instance and the j-th class, we define the class correlation between two classes j and j' as

$$\mathbf{C}_{jj'} = \widehat{\mathbf{y}}_{\cdot j}^{\mathsf{T}}\widehat{\mathbf{y}}_{\cdot j'}. \tag{2}$$

It is a coarse estimation of the class confusion. Lets delve into the definition of the class correlation in Eq. (2). Note that $\widehat{\mathbf{y}}_{\cdot j}$ denotes the probabilities that the B examples in each batch come from the j-th class. The class correlation measures the possibility that the classifier simultaneously classifies the B examples into the j-th and the j'-th classes. It is noteworthy that such pairwise class correlation is relatively safe: for false predictions with high confidence, the corresponding class correlation is still low. In other words, highly confident false predictions will negligibly impact the class correlation.

Uncertainty Reweighting. We note that examples are not equally important for quantifying class confusion. When the prediction is closer to a uniform distribution, showing no obvious peak (obviously larger probabilities for some classes), we consider the classifier as ignorant of this example. On the contrary, when the prediction shows several peaks, it indicates that the classifier is reluctant between several ambiguous classes (such as `car` and `truck`). Obviously, these examples that make the classifier ambiguous across classes will be more suitable for embodying class confusion. As defined in Eq. (2), these examples can be naturally highlighted with higher probabilities on the several peaks. Further, we introduce a *weighting* mechanism based on uncertainty such that we can quantify class confusion more accurately. Here, those examples with higher certainty in class predictions given by the classifier are more reliable and should contribute more to the pairwise class confusion. We use the *entropy* function $H(p) \triangleq -\mathbb{E}_p \log p$ in information theory as an uncertainty measure of distribution p. The entropy (uncertainty) $H(\widehat{\mathbf{y}}_{i\cdot})$ of predicting the i-th example by the classifier is defined as

$$H(\widehat{\mathbf{y}}_{i\cdot}) = -\sum_{j=1}^{|\mathcal{C}|}\widehat{Y}_{ij}\log\widehat{Y}_{ij}. \tag{3}$$

While the entropy is a measure of uncertainty, what we want is a probability distribution that places larger probabilities on examples with larger certainty of class predictions. A *de facto* transformation to probability is the softmax function

$$W_{ii} = \frac{B\left(1 + \exp(-H(\widehat{\mathbf{y}}_{i\cdot}))\right)}{\sum_{i'=1}^{B}\left(1 + \exp(-H(\widehat{\mathbf{y}}_{i'\cdot}))\right)}, \tag{4}$$

where W_{ii} is the probability of quantifying the importance of the i-th example for modeling the class confusion, and $\mathbf{W}$ is the corresponding diagonal matrix. Note that we take the opposite value of the entropy to reflect the *certainty*. Laplace Smoothing [38] (*i.e.* adding a constant 1 to each addend of the softmax function) is used to form a *heavier-tailed* weight distribution, which is suitable for highlighting more certain examples as well as avoiding overly penalizing the others. For better scaling, the probability over the examples in each batch of size B is rescaled to sum up to B such that the average weight for each example is 1. With this weighting mechanism, the preliminary definition of *class confusion* is

$$\mathbf{C}_{jj'} = \widehat{\mathbf{y}}_{\cdot j}^{\mathsf{T}} \mathbf{W} \widehat{\mathbf{y}}_{\cdot j'}. \tag{5}$$

Category Normalization. The batch-based definition of the class confusion in Eq.(5) is native for the mini-batch SGD optimization. However, when the number of classes is large, it will run into a severe *class imbalance* in each batch. To tackle this problem, we adopt a category normalization technique widely used in Random Walk [26]:

$$\widetilde{\mathbf{C}}_{jj'} = \frac{\mathbf{C}_{jj'}}{\sum_{j''=1}^{|\mathcal{C}|} \mathbf{C}_{jj''}}. \tag{6}$$

Taking the idea of Random Walk, the normalized class confusion in Eq.(6) has a neat interpretation: It is probable to walk from one class to another (resulting in the wrong classification) if the two classes have a high class confusion.

Minimum Class Confusion. Given the aforementioned derivations, we can formally define the loss function to enable Versatile Domain Adaptation (VDA). Recall that $\widetilde{\mathbf{C}}_{jj'}$ well measures the confusion between each class pair j and j'. We only need to minimize the cross-class confusion, *i.e.* $j \neq j'$. Namely, the ideal situation is that no examples are ambiguously classified into two classes at the same time. In this sense, the Minimum Class Confusion (MCC) loss is defined as

$$L_{\mathrm{MCC}}(\widehat{\mathbf{Y}}_t) = \frac{1}{|\mathcal{C}|} \sum_{j=1}^{|\mathcal{C}|} \sum_{j' \neq j}^{|\mathcal{C}|} \left| \widetilde{\mathbf{C}}_{jj'} \right|. \tag{7}$$

Since the class confusion in Eq. (6) has been normalized, minimizing the *between-class* confusion in Eq. (7) implies that the *within-class* confusion is maximized. Note that Eq. (7) is a general loss that is pluggable to existing approaches.

We want to emphasize that the inductive bias of *class confusion* in this work is more general than that of *domain alignment* in previous work [8,21,25,22,36]. As discussed in Section 2, many previous methods explicitly align features from the source and target domains, at the risk of deteriorating the feature discriminability and impeding the transferability [5]. Further, the inductive bias of *class confusion* is general and applicable to a variety of domain adaptation scenarios, while that of *domain alignment* will suffer when the domains cannot be aligned naturally (*e.g.* the partial-set DA scenarios [2,3,51]).

3.2   Versatile Approach to Domain Adaptation

The main motivation of this work is to design a versatile approach to a variety of DA scenarios. As the class confusion is a common inductive bias of many DA scenarios, combining the cross-entropy loss on the source labeled data and the MCC loss on the target unlabeled data will enable these DA scenarios.

Denote by $\widehat{\mathbf{y}}_s = G(F(\mathbf{x}_s))$ the class prediction for a source example $\mathbf{x}_s$, and by $\widehat{\mathbf{Y}}_t = G(F(\mathbf{X}_t))$ the class predictions for a batch of B target examples $\mathbf{X}_t$. The versatile approach (also termed by **MCC** for clarity) proposed in this paper for a variety of domain adaptation scenarios is formulated as

$$\min_{F,G} \mathbb{E}_{(\mathbf{x}_s, \mathbf{y}_s) \in \mathcal{S}} L_{\mathrm{CE}}\left(\widehat{\mathbf{y}}_s, \mathbf{y}_s\right) + \mu \, \mathbb{E}_{\mathbf{X}_t \subset \mathcal{T}} L_{\mathrm{MCC}}(\widehat{\mathbf{Y}}_t), \tag{8}$$

where L_{CE} is the cross-entropy loss and μ is a hyper-parameter for the MCC loss. With this joint loss, feature extractor F and classifier G of the deep DA model can be trained end-to-end by back-propagation. Note that, Eq. (8) is a *versatile* approach to many DA scenarios without any modifications to the loss.

- **Unsupervised Domain Adaptation (UDA).** Since Eq. (8) is formulated natively for this vanilla domain adaptation scenario, MCC can be directly applied to this scenario without any modification.
- **Partial Domain Adaptation (PDA).** Without explicit domain alignment, we need not to worry about the *misalignment* between source outlier classes and target classes, which is the technical bottleneck of PDA [3]. Meanwhile, compared to the confusion between the target classes, the confusion between the source outlier classes on the target domain is negligible in the MCC loss. Therefore, we can directly apply Eq. (8) to PDA.
- **Multi-Source Domain Adaptation (MSDA).** Prior methods of MSDA consider multiple source domains as different domains, capturing the internal source domain shifts, and a simple merge of source domains proves fragile. However, since MCC is based on class confusion instead of domain alignment, we can safely merge S source domains as $\mathcal{S} \leftarrow \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_S$.
- **Multi-Target Domain Adaptation (MTDA).** Though a simple merge of target domains is risky for existing methods, for MCC applied to MTDA, we can safely merge T target domains as $\mathcal{T} \leftarrow \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_T$.
- **Multi-Source/Multi-Target Partial Domain Adaptation (MSPDA / MTPDA).** As MCC can directly tackle PDA and MSDA/MTDA, it can handle these derived scenarios by simply merging multiple sources or targets.

3.3   Regularizer to Existing DA Methods

Since the inductive bias of *class confusion* is orthogonal to the widely-used *domain alignment*, our method is well complementary to the previous methods. The MCC loss Eq. (7) can serve as a regularizer pluggable to existing methods.

We take as an example the standard domain alignment framework [8,22] based on domain-adversarial training. Integrating the MCC loss as a regularizer

yields

$$\min_{F,G} \max_{D} \ \mathbb{E}_{(\mathbf{x}_s, \mathbf{y}_s) \in \mathcal{S}} L_{\mathrm{CE}}(\widehat{\mathbf{y}}_s, \mathbf{y}_s) + \mu \, \mathbb{E}_{\mathbf{X}_t \subset \mathcal{T}} L_{\mathrm{MCC}}(\widehat{\mathbf{Y}}_t) - \lambda \, \mathbb{E}_{\mathbf{x} \in \mathcal{S} \cup \mathcal{T}} L_{\mathrm{CE}}(D(\widehat{\mathbf{f}}), \mathbf{d}),$$

(9)

where the third term is the domain-adversarial loss for the domain discriminator D striving to distinguish the source from the target, and $\mathbf{d}$ is the domain label, $\widehat{\mathbf{f}} = F(\mathbf{x})$ is the feature representation learned to confuse the domain discriminator. The overall framework is a *minimax* game between two players F and D, in which λ and μ are trade-off hyper-parameters between different loss functions. Generally, the MCC loss is also readily pluggable to other representative domain adaptation frameworks, *e.g.* moment matching [21] and large norm [49].

4   Experiments

We evaluate MCC as a standalone approach with many methods for six domain adaptation scenarios (UDA, MSDA, MTDA, PDA, MSPDA and MTPDA). We also evaluate MCC as a regularizer to existing domain adaptation methods.

4.1   Setup

We use four standard datasets: **(1) Office-31** [35]: a classic domain adaptation dataset with 31 categories and 3 domains: *Amazon* (**A**), *Webcam* (**W**) and *DSLR* (**D**); **(2) Office-Home** [45]: a more difficult dataset (larger domain shift) with 65 categories and 4 domains: *Art* (**A**), *Clip Art* (**C**), *Product* (**P**) and *Real World* (**R**); **(3) VisDA-2017** [33]: a dataset with 12 categories and over 280,000 images; **(4) DomainNet** [31]: the largest and hardest domain adaptation dataset, with approximately 0.6 million images from 345 categories and 6 domains: *Clipart* (**c**), *Infograph* (**i**), *Painting* (**p**), *Quickdraw* (**q**), *Real* (**r**) and *Sketch* (**s**).

Our methods are implemented based on **PyTorch**. ResNet [13] pre-trained on ImageNet [6] is used as the network backbone. We use Deep Embedded Validation (DEV) [50] to select hyper-parameter T and provide parameter sensitivity analysis. A balance between the cross-entropy and MCC, *i.e.* $\mu = 1.0$ works well for all experiments. We run each experiment for 5 times and report the average results.

4.2   Results and Discussion

Multi-Target Domain Adaptation (MTDA). We evaluate the MTDA tasks following the protocol of DADA [32], which provides six tasks on DomainNet, the most difficult dataset to date. We adopt the strategy that directly merges multiple target domains. As shown in Table 1, many competitive methods are not effective in this challenging scenario. However, our simple method outperforms the current state-of-the-art method DADA [32] by a big margin (**7.3%**). Note that the source-only accuracy is rather low on this dataset, validating that our method, with well-designed mechanisms, is sufficiently robust to wrong predictions.

Multi-Source Domain Adaptation (MSDA). When running our method for MSDA, we similarly merge multiple source domains in MCC and compare it to existing DA algorithms that are specifically designed for MSDA on DomainNet. As shown in Table 1, based on the inductive bias of minimizing the class confusion, MCC significantly outperforms M^{3}SDA [31], the state-of-the-art method by a big margin (**5.0**%). Note that these specific methods are of very complex architecture and loss designs and may be hard to use in practical applications.

Table 1: Accuracy (%) on DomainNet for MTDA and MSDA (ResNet-101).

(a) MTDA

Method	c:	i:	p:	q:	r:	s:	Avg
ResNet [13]	25.6	16.8	25.8	9.2	20.6	22.3	20.1
SE [7]	21.3	8.5	14.5	13.8	16.0	19.7	15.6
MCD [36]	25.1	19.1	27.0	10.4	20.2	22.5	20.7
DADA [32]	26.1	20.0	26.5	12.9	20.7	22.8	21.5
MCC	**33.6**	**30.0**	**32.4**	**13.5**	**28.0**	**35.3**	**28.8**

(b) MSDA

Method	:c	:i	:p	:q	:r	:s	Avg
ResNet [13]	47.6	13.0	38.1	13.3	51.9	33.7	32.9
MCD [36]	54.3	22.1	45.7	7.6	58.4	43.5	38.5
DCTN [48]	48.6	23.5	48.8	7.2	53.5	47.3	38.2
M^{3}SDA [31]	58.6	26.0	52.3	6.3	62.7	49.5	42.6
MCC	**65.5**	**26.0**	**56.6**	**16.5**	**68.0**	**52.7**	**47.6**

Partial Domain Adaptation (PDA). Due to the existence of source outlier classes, PDA is known as a challenging scenario because of the misalignment between the source and target classes. For a fair comparison, we follow the protocol of PADA [3] and AFN [49], where the first 25 categories (in alphabetic order) of the Office-Home dataset are taken as the target domain. As shown in Table 2, on this dataset, MCC outperforms AFN [49], the *ICCV'19 honorable-mention* entry and the state-of-the-art method for PDA, by a big margin (**3.3**%).

Table 2: Accuracy (%) on Office-Home for PDA (ResNet-50).

Method (S:T)	A:C	A:P	A:R	C:A	C:P	C:R	P:A	P:C	P:R	R:A	R:C	R:P	Avg
ResNet [13]	38.6	60.8	75.2	39.9	48.1	52.9	49.7	30.9	70.8	65.4	41.8	70.4	53.7
DAN [21]	44.4	61.8	74.5	41.8	45.2	54.1	46.9	38.1	68.4	64.4	51.5	74.3	56.3
JAN [24]	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
PADA [3]	51.2	67.0	78.7	52.2	53.8	59.0	52.6	43.2	78.8	73.7	56.6	77.1	62.0
AFN [49]	58.9	76.3	81.4	70.4	**73.0**	77.8	72.4	55.3	80.4	75.8	60.4	79.9	71.8
MCC	**63.1**	**80.8**	**86.0**	**70.8**	72.1	**80.1**	**75.0**	**60.8**	**85.9**	**78.6**	**65.2**	**82.8**	**75.1**

Unsupervised Domain Adaptation (UDA). We evaluate MCC for the most common UDA scenario on several datasets. (1) *VisDA-2017*. As reported in Table 3, MCC surpasses state-of-the-art UDA algorithms and yields the highest accuracy to date among methods of no complex architecture and loss designs. (2) *Office-31*. As shown in Table 4, MCC performs the best. (3) *Two Moon* [20]. We train a shallow MLP from scratch and plot the decision boundaries of MCC and Minimum Entropy (MinEnt) [10]. MCC yields much better boundaries in Fig. 4.

Table 3: Accuracy (%) on VisDA-2017 for UDA (ResNet-101).

Method	plane	bcybl	bus	car	horse	knife	mcyle	persn	plant	sktb	train	truck	mean
ResNet [13]	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
MinEnt [10]	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 [8]	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 [21]	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 [36]	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 [22]	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
AFN [49]	**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	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**

Table 4: Accuracy (%) on Office-31 for UDA (ResNet-50).

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet [13]	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
DAN [21]	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
RTN [23]	84.5±0.2	96.8±0.1	99.4±0.1	77.5±0.3	66.2±0.2	64.8±0.3	81.6
DANN [8]	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
JAN [24]	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
GTA [37]	89.5±0.5	97.9±0.3	99.8±0.4	87.7±0.5	72.8±0.3	71.4±0.4	86.5
CDAN [22]	94.1±0.1	**98.6**±0.1	**100.0**±0.0	92.9±0.2	71.0±0.3	69.3±0.3	87.7
AFN [49]	88.8±0.5	98.4±0.3	99.8±0.1	87.7±0.6	69.8±0.4	69.7±0.4	85.7
MDD [53]	94.5±0.3	98.4±0.3	**100.0**±0.0	93.5±0.2	**74.6**±0.3	72.2±0.1	88.9
MCC	**95.5**±0.2	**98.6**±0.1	**100.0**±0.0	**94.4**±0.3	72.9±0.2	**74.9**±0.3	**89.4**

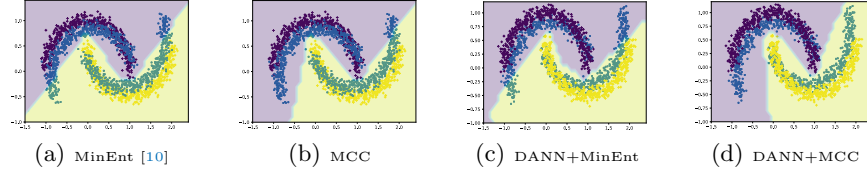(a) MinEnt [10]   (b) MCC   (c) DANN+MinEnt   (d) DANN+MCC

Fig. 4: Decision boundaries on the Two Moon dataset. Blue points indicate target data, and different classes of the source data are depicted in purple and yellow.

Multi-Source/Multi-Target Partial Domain Adaptation (MSPDA / MTPDA). Table 5 shows that MCC is versatile to handle these hard scenarios.

Table 5: Accuracy (%) on Office-Home for MSPDA and MTPDA.

(a) MSPDA

Method	:A	:C	:P	:R	Avg
DANN [8]	58.3	43.6	60.7	71.2	58.5
PADA [3]	62.8	51.8	71.7	79.2	66.4
M^{3}SDA [31]	67.4	55.3	72.2	80.4	68.8
AFN [49]	77.1	61.2	79.3	82.5	75.0
MCC	**79.6**	**67.5**	**80.6**	**85.1**	**78.2**

(b) MTPDA

Method	A:	C:	P:	R:	Avg
DANN [8]	44.6	44.8	39.1	44.1	43.1
PADA [3]	59.9	53.7	51.1	61.4	56.5
DADA [32]	65.1	63.0	60.4	63.0	62.9
AFN [49]	68.7	65.6	63.4	67.5	66.3
MCC	**73.1**	**72.1**	**69.4**	**68.3**	**70.7**

4.3   Empirical Analyses

General Regularizer. MCC can be used as a general regularizer for existing DA methods. We compare its performance with entropy minimization (MinEnt) [10] and Batch Spectral Penalization (BSP) [5]. As shown in Tables 6 and 7, MCC yields larger improvements than MinEnt and BSP to a variety of DA methods.

Table 6: Accuracy (%) on VisDA-2017 as *regularizer* for UDA (ResNet-101).

Method	plane	bcybl	bus	car	horse	knife	mcyle	persn	plant	sktb	train	truck	mean
DANN [8]	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
DANN + MinEnt [10]	87.4	55.0	75.3	**63.8**	87.4	43.6	**89.3**	72.5	82.9	**78.6**	85.6	27.4	70.7
DANN + BSP [5]	**92.2**	72.5	**83.8**	47.5	87.0	54.0	86.8	72.4	80.6	66.9	84.5	37.1	72.1
DANN + MCC	90.4	**79.8**	72.3	55.1	**90.5**	**86.8**	86.6	**80.0**	**94.2**	76.9	**90.0**	**49.6**	**79.4**
CDAN [22]	85.2	66.9	**83.0**	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38	73.9
CDAN + MinEnt [10]	90.5	65.8	79.1	62.2	89.8	28.7	**92.8**	75.4	86.8	65.3	85.2	35.3	71.4
CDAN + BSP [5]	92.4	61.0	81.0	57.5	89.0	**80.6**	90.1	77.0	84.2	77.9	82.1	38.4	75.9
CDAN + MCC	**94.5**	**80.8**	78.4	**65.3**	**90.6**	79.4	87.5	**82.2**	**94.7**	**81.0**	**86.0**	**44.6**	**80.4**

Table 7: Accuracy (%) on Office-31 as *regularizer* for UDA (ResNet-50).

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
DANN [8]	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
DANN + MinEnt [10]	91.7±0.3	98.3±0.1	**100.0**±0.0	87.9±0.3	68.8±0.3	68.1±0.3	85.8
DANN + BSP [5]	93.0±0.2	98.0±0.2	**100.0**±0.0	90.0±0.4	71.9±0.3	73.0±0.3	87.7
DANN + MCC	**95.6**±0.3	**98.6**±0.1	99.3±0.0	**93.8**±0.4	**74.0**±0.3	**75.0**±0.4	**89.4**
CDAN [22]	94.1±0.1	**98.6**±0.1	**100.0**±0.0	92.9±0.2	71.0±0.3	69.3±0.3	87.7
CDAN + MinEnt [10]	91.7±0.2	98.5±0.1	**100.0**±0.0	90.4±0.3	72.3±0.2	69.5±0.2	87.1
CDAN + BSP [5]	93.3±0.2	98.2±0.2	**100.0**±0.0	93.0±0.2	**73.6**±0.3	72.6±0.3	88.5
CDAN + MCC	**94.7**±0.2	**98.6**±0.1	**100.0**±0.0	**95.0**±0.1	73.0±0.2	**73.6**±0.3	**89.2**
AFN [49]	88.8±0.5	98.4±0.3	99.8±0.1	87.7±0.6	69.8±0.4	69.7±0.4	85.7
AFN + MinEnt [10]	90.3±0.4	**98.7**±0.2	**100.0**±0.0	92.1±0.5	73.4±0.3	71.2±0.3	87.6
AFN + BSP [5]	89.7±0.4	98.0±0.2	99.8±0.1	91.0±0.4	71.4±0.3	71.4±0.2	86.9
AFN + MCC	**95.4**±0.3	98.6±0.2	**100.0**±0.0	**96.0**±0.2	**74.6**±0.3	**75.2**±0.2	**90.0**

Ablation Study. It is interesting to investigate the contribution of each part of the MCC loss: Class Correlation **(CC)**, Probability Rescaling **(PR)**, and Uncertainty Reweighting **(UR)**. Results in Table 8 justify that each part has its indispensable contribution. To enable ease of use, we seamlessly integrate these parts into a coherent loss and reduce the number of hyper-parameters.

Further, we analyze how the specially designed Uncertainty Reweighting (UR) mechanism works. Fig. 5 shows three typical examples as well as their weights and the confusion values before and after reweighting. The classifier prediction on the first image shows no obvious peak, while the one on the third image shows two obvious peaks on classes `calculator` and `phone`. The third image is more suitable for embodying class confusion. Naturally, its confusion value is higher

Table 8: Ablation study of MCC on Office-31 for UDA (ResNet-50).

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
MCC (CC Only)	92.2	96.9	**100.0**	88.6	73.2	64.5	85.9
MCC (CC + PR)	93.1	98.5	**100.0**	91.6	70.9	69.0	87.2
MCC (CC + PR + UR)	93.7	**98.6**	**100.0**	93.2	72.1	73.7	88.4
MCC (All)	**95.5**	**98.6**	**100.0**	**94.4**	**72.9**	**74.9**	**89.4**

than the first one, and our reweighting mechanism further highlights the suitable one. On the other hand, as the reweighting mechanism is defined with entropy, we recognize that it will improperly assign high weights to examples with highly confident predictions, including the wrong ones. As shown in the second image, its ground truth label is a `lamp`, but it is classified as a `bike`. In our method, the confusion value of such an example is so low that the influence of higher weight can be neglected. Therefore, our reweighting mechanism is effective and reliable.
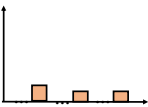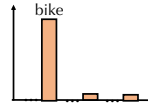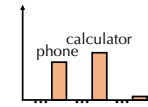
Distribution	No obvious peak	One obvious peak	Two obvious peaks
Classifier Prediction		bike	phone / calculator
Example			
Weighting Value (W)	0.230	0.470	0.340
Confusion (w/o W)	0.141	0.011	0.848
Confusion (w/ W)	0.098	0.016	0.886

Fig. 5: Three typical samples and the corresponding weights and confusion values.

Theoretical Insight. Ben-David *et al.* [1] derived the expected error $\mathcal{E}_{\mathcal{T}}(h)$ of a hypothesis h on the target domain $\mathcal{E}_{\mathcal{T}}(h) \leq \mathcal{E}_{\mathcal{S}}(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \epsilon_{\mathrm{ideal}}$ by: **(a)** expected error of h on the source domain, $\mathcal{E}_{\mathcal{S}}(h)$; **(b)** the A-distance $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$, a measure of domain discrepancy; and **(c)** the error $\epsilon_{\mathrm{ideal}}$ of the ideal joint hypothesis h^* on both source and target domains. As shown in Fig. 6, MCC has the lowest A-distance [1], which is close to the oracle one (*i.e.* supervised learning on both domains). In Fig. 7, the $\epsilon_{\mathrm{ideal}}$ value of MCC is also lower than that of mainstream DA methods. Both imply better generalization.

Parameter Sensitivity. Temperature factor T and MCC coefficient μ are the two hyper-parameters of MCC and MinEnt [10] when applying them standalone or with existing methods. We traverse hyper-parameters around their optimal values $[T^*, \mu^*]$, as shown in Fig. 6, MCC is much less sensitive to its hyper-parameters.
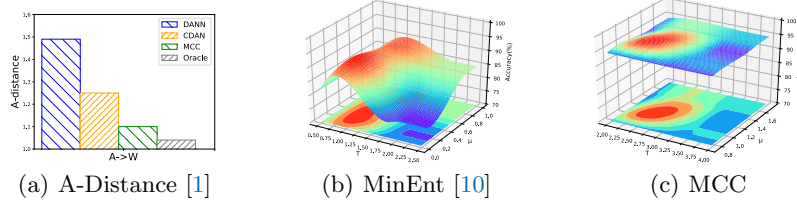
(a) A-Distance [1]   (b) MinEnt [10]   (c) MCC

Fig. 6: (a): A-Distance of the last *fc*-layer features of task $A \to W$ on Office-31 (UDA); (b)–(c): Hyper-parameter sensitivity of task $A \to W$ on Office-31 (UDA).

Convergence Speed. We show the training curves throughout iterations in Fig. 7. Impressively, MCC takes only 1000 iterations to reach an accuracy of 95%, while at this point the accuracies of CDAN and DANN are below 85%. When used as a regularizer for existing domain adaptation methods, MCC largely accelerates convergence. In general, MCC is $3\times$ faster than the existing methods.
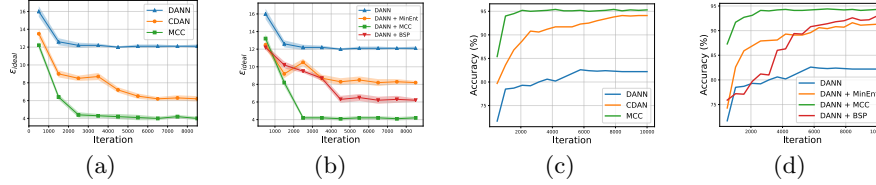

(a)   (b)   (c)   (d)

Fig. 7: The ϵ_{ideal} error values (%) and training curves throughout iterations.

5   Conclusion

This paper studies a more practical paradigm, Versatile Domain Adaptation (VDA), where one method tackles many scenarios. We uncover that less class confusion implies more transferability, which is the key insight to enable VDA. Based on this, we propose a new loss function: Minimum Class Confusion (MCC). MCC can be applied as a versatile domain adaptation approach to a variety of DA scenarios. Extensive results justify that our method, without any modification, outperforms state-of-the-art scenario-specific domain adaptation methods with much faster convergence. Further, MCC can also be used as a general regularizer for existing DA methods, further improving accuracy and accelerating training.

Acknowledgments

References

1. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine learning **79**(1-2), 151–175 (2010)
2. Cao, Z., Long, M., Wang, J., Jordan, M.I.: Partial transfer learning with selective adversarial networks. In: CVPR (2018)
3. Cao, Z., Ma, L., Long, M., Wang, J.: Partial adversarial domain adaptation. In: ECCV (2018)
4. Cao, Z., You, K., Long, M., Wang, J., Yang, Q.: Learning to transfer examples for partial domain adaptation. In: CVPR (2019)
5. Chen, X., Wang, S., Long, M., Wang, J.: Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In: ICML (2019)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
7. French, G., Mackiewicz, M., Fisher, M.: Self-ensembling for visual domain adaptation. In: ICLR (2018)
8. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. JMLR **17**(1), 2096–2030 (2016)
9. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
10. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: NeurIPS (2005)
11. Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K., Sriperumbudur, B.K.: Optimal kernel choice for large-scale two-sample tests. In: NeurIPS (2012)
12. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: ICML (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
14. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
15. Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: ICML (2018)
16. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: CVPR (2018)
17. Kang, G., Jiang, L., Yang, Y., Hauptmann, A.: Contrastive adaptation network for unsupervised domain adaptation. In: CVPR (2019)
18. Lee, C., Batra, T., Baig, M.H., Ulbricht, D.: Sliced wasserstein discrepancy for unsupervised domain adaptation. In: CVPR (2019)
19. Lee, S., Kim, D., Kim, N., Jeong, S.G.: Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In: ICCV (2019)
20. Liu, H., Long, M., Wang, J., Jordan, M.: Transferable adversarial training: A general approach to adapting deep classifiers. In: ICML (2019)
21. Long, M., Cao, Y., Wang, J., Jordan, M.I.J.: Learning transferable features with deep adaptation networks. In: ICML (2015)
22. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: NeurIPS (2018)

23. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: NeurIPS (2016)
24. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: ICML (2017)
25. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: ICML (2017)
26. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing **17**(4), 395–416 (Dec 2007)
27. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
28. Pan, S.J., Yang, Q.: A survey on transfer learning. TKDE **22**(10), 1345–1359 (2010)
29. Pan, Y., Yao, T., Li, Y., Wang, Y., Ngo, C.W., Mei, T.: Transferrable prototypical networks for unsupervised domain adaptation. In: CVPR (2019)
30. Pei, Z., Cao, Z., Long, M., Wang, J.: Multi-adversarial domain adaptation. In: AAAI (2018)
31. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: ICCV (2019)
32. Peng, X., Huang, Z., Sun, X., Saenko, K.: Domain agnostic learning with disentangled representations. In: ICML (2019)
33. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924 (2017)
34. Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset Shift in Machine Learning. The MIT Press (2009)
35. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV (2010)
36. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: CVPR (2018)
37. Sankaranarayanan, S., Balaji, Y., Castillo, C.D., Chellappa, R.: Generate to adapt: Aligning domains using generative adversarial networks. In: CVPR (2018)
38. Schutze, H., Manning, C.D., Raghavan, P.: Introduction to information retrieval. In: Proceedings of the international communication of association for computing machinery conference (2008)
39. Shu, R., Bui, H.H., Narui, H., Ermon, S.: A DIRT-T Approach to Unsupervised Domain Adaptation. In: ICLR (2018)
40. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: ECCV (2016)
41. Tarvainen, A., Valpola, H.: Weight-averaged consistency targets improve semi-supervised deep learning results. In: NeurIPS (2017)
42. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: ICCV (2015)
43. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017)
44. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. CoRR **abs/1412.3474** (2014)
45. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep Hashing Network for Unsupervised Domain Adaptation. In: CVPR (2017)
46. Wang, X., Jin, Y., Long, M., Wang, J., Jordan, M.I.: Transferable normalization: Towards improving transferability of deep neural networks. In: Advances in Neural Information Processing Systems. pp. 1953–1963 (2019)
47. Wang, X., Li, L., Ye, W., Long, M., Wang, J.: Transferable attention for domain adaptation. In: AAAI (2019)

48. Xu, R., Chen, Z., Zuo, W., Yan, J., Lin, L.: Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In: CVPR (2018)
49. Xu, R., Li, G., Yang, J., Lin, L.: Larger norm more transferable: An adaptive feature norm approach. In: ICCV (2019)
50. You, K., Wang, X., Long, M., Jordan, M.: Towards accurate model selection in deep unsupervised domain adaptation. In: ICML (2019)
51. Zhang, J., Ding, Z., Li, W., Ogunbona, P.: Importance weighted adversarial nets for partial domain adaptation. In: CVPR (2018)
52. Zhang, Y., Tang, H., Jia, K., Tan, M.: Domain-symmetric networks for adversarial domain adaptation. In: CVPR (2019)
53. Zhang, Y., Liu, T., Long, M., Jordan, M.I.: Bridging theory and algorithm for domain adaptation. In: ICML (2019)
54. Zhao, H., Zhang, S., Wu, G., Moura, J.M., Costeira, J.P., Gordon, G.J.: Adversarial multiple source domain adaptation. In: NeurIPS (2018)
55. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: ECCV (2018)
56. Zou, Y., Yu, Z., Liu, X., Kumar, B.V., Wang, J.: Confidence regularized self-training. In: ICCV (2019)