An Empirical Study of Domain Adaptation: Are We Really Learning Transferable Representations?

Nicholas Josselyn[1], Biao Yin[1], Ziming Zhang[1,3], and Elke Rundensteiner[1,2]

[1]Department of Data Science, Worcester Polytechnic Institute, Worcester, MA
[2]Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA
[3]Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020767

***Abstract*—Deep learning often relies on the availability of a large amount of high-quality labeled data, which can be very limited in novel domains. To address such data scarcity, domain adaptation is one promising approach that allows for deep networks to leverage large amounts of available data from a source domain to enhance the model's efficacy on the target domain of interest. However, while there is a plethora of alternate models for domain adaptation proposed over many years in the literature, there is a dearth of studies that objectively compare the relative effectiveness of these models in a rigorous, empirical study. To fill this gap, we provide a thorough, unbiased, empirical study of five state-of-the-art (SOTA) deep domain adaptation models proposed over the past 6 years whose codes are publicly available. Models are evaluated on the complex and diverse domain adaptation tasks featured in the DomainNet benchmark dataset as well as the popular Office-31 dataset. Our results suggest that (1) all 5 models perform similarly, on average, and do not even significantly beat the oldest model, and (2) counter to their intended purpose, the transfer loss functions in the literature do not contribute significantly to learning transferable representations. Our observations suggest that domain adaptation research needs to more thoroughly compare newly proposed models against existing works, along with assessing their loss functions' utility thoroughly. Our code and data splits are made public for reproducibility of results by the community.**

***Index Terms*—Domain Adaptation, Transfer Learning**

I. Introduction

Background. A primary influence on the effectiveness of deep learning is the availability of data, specifically high quality labeled data. It could be relatively simple for many industries or organizations to generate a lot of unorganized or unlabeled data, but providing labels for large quantities of data requires costly, both in time and money, human intervention. Worse yet, such human-generated labels are subject to human bias and errors. For each new problem, one would expect that a new model needs to be trained with new data for each task due to dataset bias between the distribution of data used for training *vs.* testing. Over the past few years, transfer learning approaches have thus been developed to exploit large labeled datasets (such as ImageNet) to learn high-level features of images and thereafter fine-tune models for new problems by additional training on new targeted task-specific datasets. However, it is often difficult to have enough data to properly tune a model for specific tasks [1], [2].

Domain adaptation (DA), a subfield in transfer learning, primarily assumes that there are two or more similar but distinct sets of data (domains) called *source* domain and *target* domain. Generally, the task for each of these datasets is the same (e.g. classification of planes) but there exists a *domain shift* between the domains (e.g. photos of planes *v.s.* drawings of planes). A domain shift can exist for several reasons: the data is captured via different modalities (infrared vs visual spectrum images), changes in pose, changes in object background, variations in color, dimensionality differences, or any combination, and more. The goal of DA is two-fold, namely, to learn a model that can both perform a task on the source data well and simultaneously learn a transferable representation of the data such that the source and target domains are indistinguishable. This then would allow for good performance of the task on the target domain as well.

Within the field of DA, we distinguish between different approaches depending on the availability of data: supervised, semi-supervised, and unsupervised. In this work, we focus on unsupervised DA (UDA), arguably the most popular in literature. In UDA, during training, both labeled source examples and unlabeled target examples are utilized. We focus on five core models spanning the 6 recent years in the literature. These models all have two core loss components, namely, a classification loss and a transfer loss. The latter is introduced to align the source and target domains.

Applications. Domain adaptation has potential utility in many application areas, with two important ones being self-driving cars [3]–[5] and medical imaging [6]–[8], as described below.

For self-driving cars, unforeseen environmental factors, different road layouts, and robust pedestrian identification are some of the many hurdles to overcome. Training data for self-driving cars may not be able to capture every potential environmental scenario, roads from one country may differ from another, and pedestrians being in unforeseen, dangerous scenarios in the real-world. All these are examples of *domain shifts* between training and testing and ultimately deployment. Thus, domain adaptation may be a potential solution.

For medical imaging, consensus for diagnoses across patients' anatomy and medical devices (scanners, MRI, CT) is critical. Depending on the cohort of training data, identifying anatomical structures is complicated due to the variety of anatomy present in each dataset. Further, depending on a hospital's scanning parameters and device choices, model

predictions are subject to variability across devices. Finally, if a model is trained on data from one modality (MRI), but it is desired to then perform a diagnosis on another (CT), it is difficult to achieve reliable results. These challenges could potentially be addressed with domain adaptation.

Our Approach and Contributions. We note that a plethora of UDA models have been proposed in the literature, each making claims of better performance than the previous and proposing alternate variations of transfer loss functions to align domains. While categorizations of these DA models have been provided in survey papers [7], [9]–[12], there is a scarcity of unbiased, comprehensive analyses of DA models needed to gain an objective understanding of their relative effectiveness. Given the importance of this task for the community at large, both future model developers and practitioners, we provide an in-depth evaluation in this work. In particular, we conduct a comprehensive experimental study of 5 SOTA UDA models.

Our empirical evaluation study rests upon several important pillars. *First*, we conduct an unbiased, thorough experimental study of popular UDA models on varying amounts of training data (from big to small, more limited data sets) and for a diverse set of domain types and tasks. *Second*, we utilize open-source, benchmark datasets to assure open access to our models and experimental data. *Third*, we provide an analysis of the utility of the proposed transfer loss functions in the DA literature, assessing their relative contribution (or lack thereof) to learning an effective classifier. Our results are two-fold. *(1) All 5 models, regardless of when they were released, perform similarly, on average.* Interestingly, we find that none of the newer models significantly beats even the oldest model. *(2) Surprisingly, the proposed transfer loss functions across these models do not significantly contribute to learning transferable representations.* This result is in contradiction to their proposed functionality. Given our findings, we recommend to the DA research community that a more careful and fair comparison of newly proposed models against the existing literature be conducted, and in particular, proof of domain alignment is shown. Further, we suggest to practitioners looking to adopt DA models that well-established older models are worth to consider and try out on their applications, as they may be more robust to different data scenarios.

II. Related Work

Domain Adaptation Models. There exists a large body of literature on domain adaptation models – all proposing new ways in which we can tackle the problem of domain shift via learning similar representations between domains with new loss functions. These works reflect several areas of domain adaptation, namely, single-source closed-set DA [13]–[25], multi-source DA [26]–[29], multi-target DA [30], open-set DA [31], partial DA [32], [33], and other subset fields of domain adaptation. In our work, we focus on one of the foundational settings of domain adaptation: *unsupervised, single-source, single-target, closed-set domain adaptation with two loss components.* When a new model is proposed in the literature and compared against baselines, they are subject to unintentional bias. In our work, we have no preference to any one model and thus can be fully objective in our analysis.

Survey Papers. This work is similar to survey-style work [7], [9]–[12] as it discusses existing models. However, in these works, they only catalogue a large history of domain adaptation and transfer learning models. Results reported in these works are generally taken from existing manuscripts, are not thorough model investigations under varied data scenarios, nor are comparative in nature. In this work, we take an in-depth, empirical approach and provide a thorough evaluation of a set of SOTA domain adaptation models on a diverse set of adaptation tasks. We further deeply investigate the effectiveness of transfer loss functions proposed in the literature.

Experimental Analysis. Other than survey work, there is one notable work that compares UDA models [34]. However, a large portion of this work is spent on the design of a new model and a new dataset, with evaluation on their new dataset. Further, they do not explicitly compare UDA models in the literature. Instead, they restrict their comparison to shallow and deep DA models under different weight sharing strategies.

III. Evaluated Models

In this section, we introduce the 5 UDA models investigated in this study and point out their respective uniqueness as well as commonalities between them. The 5 models, introduced over the past 6 years, were selected based on the following criteria: (1) highly cited, recent or established models in literature, (2) models that have been compared against in a large majority of the domain adaptation literature, and (3) models that match our foundational scenario of being single-source, single-target, closed-set DA with two primary loss components. To implement these models, we use a public DA library [35]. We utilize this framework and 5 of its models and update codes according to the datasets we use and metrics we record. Code for the models we have updated is provided along with access to data for reproducibility [1].

DANN. Domain-Adversarial Training of Neural Networks [15] is a highly cited UDA model introduced in 2016 that is commonly used as a representative baseline. DANN introduces a gradient reversal layer to promote alignment of the source and target domain representations with respect to a domain discriminator. DANN consists of a label predictor that predicts the class label of the source data during training and subsequently target data during testing. Further, DANN incorporates a domain discriminator that discriminates between source and target data during training. The training procedure aims to minimize the loss of the label predictor and maximize the domain confusion loss of the discriminator via an adversarial approach with the gradient reversal layer. With this, there are two main loss components: a classification loss for the label predictor and a transfer loss for the domain discriminator.

JAN. Deep Transfer Learning with Joint Adaptation Networks [13], a UDA model introduced in 2017, aims to reduce the

[1] https://github.com/njosselyn13/Empirical-Study-Domain-Adaptation

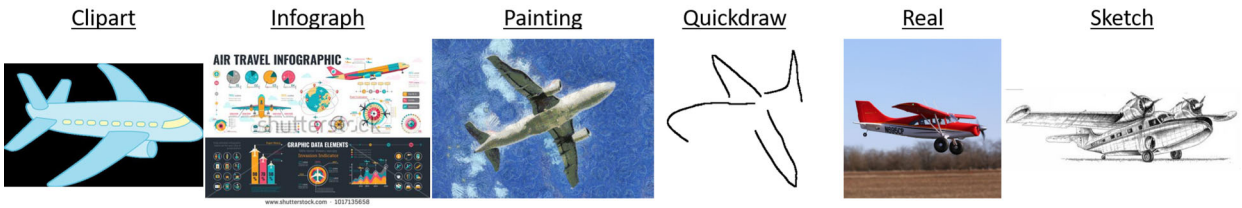


Figure 1: Images from the DomainNet dataset. Airplane class for all 6 domains shown.

Figure 2: Images from the Office-31 dataset. Backpack class for all 3 domains shown.

shifts in joint distributions across domains. Typically, the Maximum Mean Discrepancy [36] is used to measure the discrepancy in marginal distributions between domains. For this, JAN formulates a function to measure the discrepancy between the joint distributions between domains across multiple CNN layers. In measuring the discrepancy of the joint distributions, the authors argue that they can overcome the residual joint distribution shifts not addressed in other domain adaptation models. In this model, a classification loss for assessing the labeled source data remains. The proposed joint distribution alignment corresponds to their metric for transfer loss.

CDAN. Conditional Adversarial Domain Adaptation [18], a UDA model introduced in 2017, was inspired by the advances in conditional generative adversarial networks (CGANs). CGANs make use of discriminative features between real and fake data and incorporate them in a conditional manner into the generator and discriminator networks. In CDAN, a similar approach is taken by conditioning the domain discriminator with the cross-covariance of domain-specific feature representations and classifier predictions. Additionally, the discriminator is conditioned based on the uncertainty of the classifier predictions – thus allowing the discriminator to prioritize easy-to-transfer examples. In this model, there is still a classification loss for assessing the labeled source data. Also, the proposed transfer loss is fairly similar to DANN and its discriminator, simply now with the inclusion of conditioning.

AFN. Larger Norm More Transferable: An Adaptive Feature Norm Approach for Unsupervised Domain Adaptation [14], a UDA model introduced in 2019, proposes a novel method for aligning domains based on statistical criterion different from the literature. That is, they suggest that a better way to align domains is via finding a shared, average feature norm (length of the feature vector) between the two domains. They note that the target domain feature norms are typically much smaller than the source feature norms. They conjecture that this may complicate the adaptation. By adapting the feature norms of both domains to a large range of scalars, they expect they can achieve better adaptation. There is again a classification loss, while transfer loss is the feature norm loss.

MCC. Minimum Class Confusion for Versatile Domain Adaptation [37], a UDA model introduced in 2020, does not aim to explicitly align two domain feature spaces, but instead it aims to reduce class confusion in the label space. MCC is versatile because its approach is suitable for a variety of DA settings such as: closed-set, partial-set, multi-source, and multi-target DA. In our work focused on comparing single-source/single-target solutions, we leverage MCC just for this particular single-source and single-target setup. MCC claims to outperform prior models including the models we study here, namely, DANN, JAN, CDAN, and AFN. In MCC, there are two main loss components, classification loss for the source data and transfer loss designed to minimize the number of misclassifications in the target domain.

IV. Dataset and Experimental Protocol

In this section, we provide details on the experimental methodology to compare the 5 UDA models, including datasets, data preparation, training setup, and more.

A. Datasets and Preparation

In this work, we focus on two benchmark domain adaptation image datasets: the popular Office-31 dataset [38], and the largest, most diverse and complex dataset, DomainNet [26].

DomainNet. Published in 2019, it is the largest domain adaptation benchmark dataset containing 586,575 images across 6 distinct and uniquely challenging domains with 345 classes in each domain. The 6 domains are: clipart, infograph, painting, quickdraw, real, and sketch images. This allows for 30 adaptation tasks to be evaluated (single-source, single-target DA setup). In each domain, diverse classes such as airplane, toothpaste, dragon, rabbit, ear, etc. exist. An example of images for each domain are shown in Figure 1.

In our study, we distinguish between 3 training scenarios based on the amount of training data provided, namely, large, medium, and small datasets. We investigate varying training data scenarios in order to assess the robustness of each model to data availability. Using the original split for DomainNet

	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Total
Subset-20	5,516	5,953	8,286	19,320	19,455	7,851	66,381
Subset-50	13,485	14,483	20,270	48,300	48,432	19,354	164,324
Full	26,820	28,818	40,358	96,600	96,725	38,570	327,891

Table I: Amount of *training* data for each of the 6 domains in DomainNet for each of the 3 data subsets used.

defined in [26] and the VisDA-2019 competition (with small modifications due to some classes not being assigned images in the original release), 30% of the data is partitioned into a held-out test set. The remaining 70% of data is designated for training (and validation). We break this remaining 70% of data into 3 distinct datasets: the full amount of data (full), 50% of the data per class per domain (subset-50), and 20% of the data per class per domain (subset-20). For this data subsetting, a hard minimum of 6 images per class was enforced to allow for cross-validation to be performed. Then, for each of the 3 subsets of training data, 5 stratified cross-validated folds of training and validation data are generated with 20% being held for validation and 80% for training. In Table I we provide statistics on how many images are available for training for each of the 6 domains over all 345 classes. All data subsets are released to assure reproducibility.

Office-31. This dataset consists of 3 domains of office-space image data taken from Amazon, a DSLR camera, and a webcam camera, allowing for 6 adaptation tasks (single-source, single-target DA setup). There are a total of 4,110 images across all 3 domains. In each domain, there are 31 classes, including, back packs, scissors, trash cans, etc. Examples can be seen in Figure 2. In the Amazon domain, there are 2,817 images. The Amazon domain provides images with clean backgrounds and a uniform scale. In the DSLR domain, there is a total of 498 images. The DSLR domain provides images with low noise and high resolution. The webcam domain, with a total of 795 images, presents images with low resolution, significant noise, and color and white balance artifacts. The Office-31 dataset is considered a small-scale domain adaptation dataset with small domain shifts, particularly between webcam and DSLR domains.

To assure a comprehensive analysis of model performance, we assess all 5 models on the full Office-31 dataset, which differs widely in tasks and data types from DomainNet. Unlike DomainNet, we do not subset the Office-31 dataset to smaller training scenarios as it is already a smaller-scale dataset.

We split the Office-31 dataset into training, validation, and testing sets. We split 15% of the data to a held-out test set with the remaining 85% of data being split into 5 cross-validated folds; 70% held for training and 30% held for validation.

B. Experimental Protocols

In this section, we first outline our methodology for establishing a lower bound for domain adaptation performance, i.e., the performance of models without domain adaptation on a set of transfer tasks. Then, using the 3 subsets of DomainNet data and the Office-31 dataset, we describe our experimental methodology for comparing and tuning the 5 UDA models and for the assessment of the impact transfer loss has on learning.

Baseline Transfer Tasks. As a first step, we conduct baseline experiments to confirm that indeed a classifier trained on one domain of data with no domain adaptation, when applied to a similar but different domain of data, will perform poorer than when using domain adaptation models.

For this, we train 6 classifiers on the full DomainNet dataset using classical single-domain models. That is, no domain adaptation is used. This is repeated for all 6 domains. For each of these classifiers, we tune the learning rate and weight decay and select the model with highest validation accuracy. We sample learning rate and weight decay values of 0.1, 0.01, and 0.001. Once an optimal model for each classifier is chosen, it is then applied to the 5 other target domain *test* data sets.

Model Comparison. For each of the 5 UDA models, 5-fold cross-validation experiments are conducted on the Office-31 dataset and each of the 3 subsets of the DomainNet dataset. Experiments are conducted for all 30 domain adaptation tasks in the DomainNet dataset and all 6 adaptation tasks in the Office-31 dataset. When a domain (e.g. clipart) is selected as the source domain for a model and dataset pair (e.g. JAN and full DomainNet dataset), the labels are made available to the model at train time. When a domain (e.g. infograph) is selected as the target domain for a model and dataset pair, the labels are not available to the model during training. The target domain data for the corresponding validation and test sets are then used to be tuned and evaluated on, respectively.

For each model and dataset experiment pair, and each adaptation task, test set accuracies are reported over the 5 cross-validation folds. Additionally, we compute for each model and dataset pair an average test accuracy over all adaptation tasks. Code and data for reproducibility are made available.

Hyperparameter Tuning. Extensive hyperparameter tuning was conducted to give each model a fair chance at maximizing their performance on the DomainNet or Office-31 datasets. All experiments run for 30 epochs. The test accuracy where the max validation accuracy is observed is reported, thus implementing an early stopping approach. We experimented with longer training times (100 epochs) with a subset of 5 adaptation tasks from DomainNet (clipart as source). But we did not observe significant improvement in accuracy to warrant running longer than 30 epochs. Additionally, we tuned each model's learning rate, weight decay, and the loss trade-off hyperparameters. The loss trade-off weighs how much to focus on classification loss vs each model's respective transfer loss. This trade-off is seen in Equation 1.

$$loss = cls\_loss + transfer\_loss * trade\_off \quad (1)$$

DANN [15]	clp	inf	pnt	qdr	rel	skt
clp	-	27.0	37.3	18.7	47.6	50.8
inf	19.1	-	18.6	3.9	22.7	19.3
pnt	34.3	25.9	-	7.9	47.7	41.4
qdr	12.7	5.5	5.8	-	6.5	12.4
rel	50.4	36.9	51.8	12.7	-	50.8
skt	40.7	22.6	33.5	13.0	37.4	-

JAN [13]	clp	inf	pnt	qdr	rel	skt
clp	-	28.1	38.5	15.3	46.6	48.5
inf	17.2	-	17.3	2.4	20.6	16.5
pnt	33.9	26.6	-	6.7	46.6	39.2
qdr	11.4	4.7	4.8	-	4.6	10.1
rel	49.5	39.4	51.5	11.7	-	47.0
skt	40.2	24.0	35.7	10.9	36.5	-

CDAN [18]	clp	inf	pnt	qdr	rel	skt
clp	-	26.6	37.2	19.9	49.3	51.0
inf	18.7	-	18.4	4.0	22.7	18.8
pnt	35.0	25.2	-	7.1	49.0	42.0
qdr	10.1	4.7	4.7	-	4.5	9.3
rel	52.6	37.0	51.6	14.4	-	51.1
skt	40.9	22.0	34.8	13.8	38.7	-

AFN [14]	clp	inf	pnt	qdr	rel	skt
clp	-	29.1	38.9	17.3	43.9	49.6
inf	15.8	-	16.5	2.8	17.6	15.6
pnt	36.0	32.5	-	6.6	46.6	42.6
qdr	11.3	3.1	4.7	-	4.8	11.1
rel	51.9	45.7	53.9	11.8	-	51.9
skt	41.2	25.0	36.4	9.3	34.4	-

MCC [37]	clp	inf	pnt	qdr	rel	skt
clp	-	31.1	43.0	11.6	51.2	53.6
inf	13.5	-	14.8	1.1	17.7	13.8
pnt	33.4	28.7	-	2.6	46.1	39.8
qdr	12.9	2.7	3.7	-	4.0	11.7
rel	49.0	43.7	52.5	7.0	-	49.1
skt	39.3	22.7	36.1	7.9	35.1	-

Averages:

DANN (2016)	**27.2**
CDAN (2017)	**27.2**
AFN (2019)	26.9
JAN (2017)	26.2
MCC (2020)	26.0

Table II: Mean target domain test accuracy over 5-cross-val folds for all models for all 30 adaptation tasks on the *full* dataset. Average test accuracies over all 30 tasks reported in bottom right. (Columns = source domain, rows = target domain)

	clp	inf	pnt	qdr	rel	skt
clp	-	29.2	29.7	8.1	44.4	46.3
inf	15.9	-	15.7	0.6	19.0	13.4
pnt	35.3	27.4	-	1.0	45.1	32.4
qdr	9.1	2.6	2.9	-	4.1	11.1
rel	49.6	43.9	44.4	2.4	-	45.3
skt	37.0	22.8	27.2	5.9	32.0	-

Table III: Baseline transfer task test accuracies on the full DomainNet dataset. (Columns = source, rows = target)

For each model, we tune hyperparameters on the full DomainNet dataset, for a subset of 5 adaptation tasks (clipart as source). For the Office-31 dataset, we perform tuning for each model over all 6 adaptation tasks. Learning rate and weight decay combinations explored for each model were: 0.1, 0.01, 0.001, 0.0001, and original code defaults. Loss trade-off values explored for each model were: 0.25, 0.5, 0.75, 2, 5, 10, 20, 50, and original code defaults. Larger values mean giving more relative weight to the transfer loss. All final hyperparameters are available in the released code repository.

Finally, for the most recent MCC model, after much experimentation, we found it to be one of the poorer performing models on DomainNet. Thus, we reached out to the MCC authors and took their advice to tune the temperature hyperparameter with suggested values of 0.25, 0.5, 1, 2, and 3.

Transfer Loss Analysis. To assess the utility of proposed transfer loss functions in DA models and their contribution to learning an effective target domain classifier, we examine training loss curves for each model for each loss trade-off hyperparameter we worked with. We observe how the classification and transfer losses evolve over the 30 training epochs independently. We then conclude whether any amount of weighting for the transfer loss (via the loss trade-off hyperparameter) leads to significant learning during training with respect to the transfer loss. We show, for each model, the lack of learning occurring with respect to the transfer loss and discuss potential reasons for this observation.

For these experiments, we use optimal hyperparameters determined previously on the DomainNet dataset. We only vary the loss trade-off value. We observe loss curves for the full DomainNet dataset and the first cross validation fold.

Next, we use the trained models with optimal hyperparameters, from the first cross-validation fold, to generate T-SNE plots. These plots show the learned distributions of each model-adaptation task pair for one class (airplane). We extract the features before the final classification layer for either the source or target data and reduce them to 2 dimensions via T-SNE. This allows us to visualize how much the representations between two domains have been transformed to be similar.

V. Experimental Results

In this section, we establish a lower bound baseline (no domain adaptation) for the DomainNet dataset transfer tasks. Then, we discuss results for our two proposed contributions in Section I: (1) model comparisons and robustness, and (2) the utility of transfer loss functions proposed in the DA literature.

A. Baseline Transfer Tasks

Results using the full DomainNet dataset with no domain adaptation are presented in Table III. On average, over all 30 adaptation tasks, this baseline approach achieves only 23.4% test accuracy. We see later in Table II (test results also using the full DomainNet dataset) that this performance is lower than that for all 5 domain adaptation models. In fact, this

DANN [15]	clp	inf	pnt	qdr	rel	skt
clp	-	22.9	34.1	17.9	45.3	45.9
inf	17.1	-	17.0	3.9	21.6	17.5
pnt	32.0	23.0	-	7.5	46.2	38.7
qdr	11.8	5.2	5.3	-	6.6	11.2
rel	46.8	33.6	49.7	12.8	-	48.1
skt	37.3	19.8	30.9	12.5	35.4	-

JAN [13]	clp	inf	pnt	qdr	rel	skt
clp	-	24.0	35.4	14.8	44.6	44.4
inf	15.4	-	16.0	2.6	19.5	14.7
pnt	31.5	23.3	-	6.9	45.5	36.2
qdr	10.5	4.3	4.2	-	4.6	9.5
rel	46.6	35.2	50.1	12.1	-	44.6
skt	36.5	20.0	33.1	11.2	34.6	-

CDAN [18]	clp	inf	pnt	qdr	rel	skt
clp	-	22.4	34.7	19.1	47.1	46.2
inf	16.9	-	17.1	4.0	21.8	17.1
pnt	32.2	21.7	-	6.9	47.8	39.0
qdr	8.4	4.3	3.9	-	3.6	8.5
rel	49.1	33.9	50.3	13.8	-	48.5
skt	37.1	18.7	31.8	13.4	36.4	-

AFN [14]	clp	inf	pnt	qdr	rel	skt
clp	-	27.0	36.5	17.3	43.2	47.2
inf	14.8	-	15.8	2.8	17.2	15.0
pnt	34.2	29.6	-	6.7	45.9	41.0
qdr	10.5	3.2	4.2	-	4.5	10.6
rel	49.9	42.5	52.8	12.2	-	50.3
skt	38.2	22.0	34.3	9.2	33.7	-

MCC [37]	clp	inf	pnt	qdr	rel	skt
clp	-	26.4	38.1	11.9	47.2	49.4
inf	11.7	-	13.2	1.2	16.8	12.4
pnt	29.6	24.5	-	2.9	43.3	37.0
qdr	12.0	2.7	4.0	-	4.7	11.2
rel	45.1	38.6	49.0	6.6	-	46.1
skt	34.9	19.0	32.5	7.4	33.2	-

Averages:

AFN (2019)	**25.7**
DANN (2016)	25.2
CDAN (2017)	25.2
JAN (2017)	24.4
MCC (2020)	23.8

Table IV: Mean target domain test accuracy over 5-cross-val folds for all models for all 30 adaptation tasks on the *subset-50* dataset. Average test accuracies over all 30 tasks reported in bottom right. (Columns = source domain, rows = target domain)

performance is lower than all 5 models when only 50% of the training data is used. *This confirms what has been observed in the literature: domain adaptation outperforms the simple application of trained models to related but different domains.*

B. Model Comparison Analysis

We present results obtained on the DomainNet dataset for all 5 UDA models, followed by results on Office-31.

DomainNet. Addressing the first objective of model robustness in our work, we first present 5-fold cross-validation results on the test dataset (for each target domain) for all 5 UDA models for a diverse set of 30 adaptation tasks from the DomainNet dataset in Tables II, IV, and V. Each column designates a source domain and each row a target domain. Table II reports results on the full dataset, Table IV on the subset-50 dataset, and Table V on the subset-20 dataset. Additionally, each table presents the average accuracy over all adaptation tasks for each model as is typically done in the DA literature to provide a summary of model performance overall.

In Table II, results for all models on the *full* dataset are shown. We observe that for 9/30 adaptation tasks, DANN performs best, for 9/30 tasks AFN performs best, for 8/30 tasks CDAN performs best, for 5/30 tasks MCC performs best, and for 0 tasks JAN performs best. Note that for the real-to-infograph task, DANN and CDAN tie. Over all 30 adaptation tasks, for all models, the standard deviations over the 5 cross-validation folds ranges from 0.1 to 1.1. With respect to averages over all 30 adaptation tasks, DANN and CDAN beat the other 3 models with 27.2% test accuracy, with AFN as a close second with 26.9%. We note that *the oldest method (DANN) is performing best for a diverse set of challenging adaptation tasks; with the most recently published method (MCC) performing the poorest*. This is of particular interest as it raises concerns over the current trend in proposing new domain adaptation methods, and the rigor with which new models should be tested against baselines to assure their robustness in performance in practice.

In Table IV, results for all models on the *subset-50* dataset are shown. We observe that for 7/30 adaptation tasks DANN performs best, for 11/30 tasks AFN performs best, for 8/30 tasks CDAN performs best, for 5/30 tasks MCC performs best (tied with DANN for sketch to quickdraw task), and for 0 tasks JAN performs best. Over all 30 adaptation tasks, for all models, the standard deviations over the 5 cross-validation folds ranges from 0.0 to 0.9. With respect to averages over all 30 adaptation tasks, AFN just slightly wins with 25.7% test accuracy, but is closely followed by DANN and CDAN with 25.2%. Again we observe that the oldest of the methods (DANN) is performing near the top, and the newest of the methods (MCC) is performing at the bottom. We begin to observe a *trend of model robustness across training data availability;* DANN, CDAN, and AFN perform well with the full amount of training data and with 50% of the training data.

In Table V, results for all models on the *subset-20* dataset are shown. We observe that for 10/30 adaptation tasks, DANN performs best, for 15/30 tasks, AFN performs best, for 1 task, MCC performs best, and JAN and CDAN never perform best. For the painting-to-infograph task, DANN and CDAN tie. Over all 30 adaptation tasks, for all models, the standard deviation over the 5 cross-validation folds ranges from 0.1 to 0.9. With respect to averages over all 30 adaptation tasks, AFN performs best with 23.2% test accuracy, with DANN coming in second again with 22.1%. Again we see MCC perform towards the bottom of the model list. We further observe the *same trend*

DANN [15]	clp	inf	pnt	qdr	rel	skt
clp	-	17.5	29.0	16.4	41.0	39.0
inf	14.1	-	14.9	3.8	19.9	14.8
pnt	27.1	18.3	-	7.2	43.0	34.2
qdr	9.9	3.9	5.0	-	6.1	9.6
rel	42.2	28.3	46.3	12.4	-	43.5
skt	30.8	15.0	27.1	11.2	32.7	-

JAN [13]	clp	inf	pnt	qdr	rel	skt
clp	-	18.5	30.5	13.7	40.3	37.6
inf	12.8	-	14.0	2.6	17.8	12.3
pnt	26.7	19.9	-	6.6	43.0	31.6
qdr	7.9	3.1	4.1	-	3.9	7.7
rel	40.8	31.0	47.3	11.3	-	39.5
skt	29.7	15.4	28.4	9.9	30.8	-

CDAN [18]	clp	inf	pnt	qdr	rel	skt
clp	-	16.7	30.3	16.1	42.3	38.9
inf	13.7	-	14.9	3.6	20.0	14.0
pnt	25.3	18.9	-	6.2	44.6	34.1
qdr	4.8	2.2	2.5	-	3.3	5.4
rel	42.1	29.8	47.8	12.2	-	44.1
skt	28.1	12.3	27.4	11.7	33.0	-

AFN [14]	clp	inf	pnt	qdr	rel	skt
clp	-	21.4	32.7	17.0	40.7	41.2
inf	13.2	-	14.7	2.8	16.7	13.0
pnt	30.3	24.8	-	6.2	44.7	36.8
qdr	8.9	2.4	3.3	-	3.8	9.5
rel	45.3	35.9	50.7	11.8	-	45.9
skt	32.5	17.5	30.4	8.7	31.8	-

MCC [37]	clp	inf	pnt	qdr	rel	skt
clp	-	19.0	30.6	11.6	40.6	40.3
inf	8.7	-	10.7	1.2	15.0	9.6
pnt	22.5	18.4	-	3.1	39.6	31.0
qdr	9.6	2.9	4.3	-	4.9	9.8
rel	37.8	30.6	44.3	7.2	-	39.6
skt	26.4	13.5	26.1	7.9	27.9	-

Averages:

AFN (2019)	**23.2**
DANN (2016)	22.1
CDAN (2017)	21.6
JAN (2017)	21.3
MCC (2020)	19.8

Table V: Mean target domain test accuracy over 5-cross-val folds for all models for all 30 adaptation tasks on the *subset-20* dataset. Average test accuracies over all 30 tasks reported in bottom right. (Columns = source domain, rows = target domain)

DANN [15]	Amazon	DSLR	Webcam
Amazon	-	69.8	70.0
DSLR	91.7	-	99.7
Webcam	89.2	97.5	-

JAN [13]	Amazon	DSLR	Webcam
Amazon	-	70.1	68.6
DSLR	92.5	-	100.0
Webcam	90.0	97.5	-

CDAN [18]	Amazon	DSLR	Webcam
Amazon	-	69.9	65.9
DSLR	91.7	-	100.0
Webcam	89.7	97.3	-

AFN [14]	Amazon	DSLR	Webcam
Amazon	-	70.3	70.2
DSLR	93.1	-	100.0
Webcam	89.2	98.2	-

MCC [37]	Amazon	DSLR	Webcam
Amazon	-	74.7	77.2
DSLR	94.7	-	100.0
Webcam	92.2	98.3	-

Averages:

MCC (2020)	**89.5**
AFN (2019)	86.8
JAN (2017)	86.5
DANN (2016)	86.3
CDAN (2017)	85.8

Table VI: Mean target domain test accuracy over 5-cross-val folds for all models for all 6 adaptation tasks on the Office-31 dataset. Average test accuracies over all 6 tasks reported in far right. (Columns = source domain, rows = target domain)

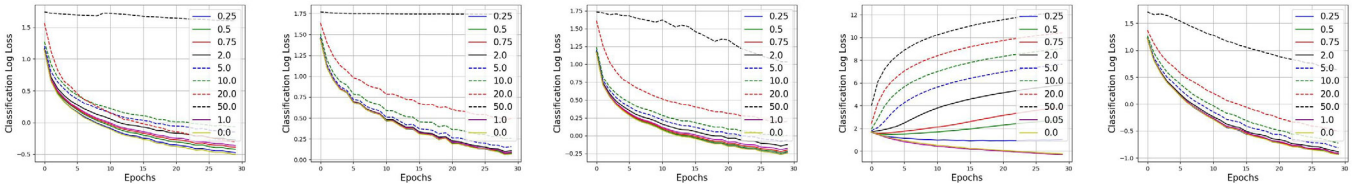
that AFN and the oldest model DANN perform best for a wide variety of adaptation tasks and for all 3 sizes of training data.

Office-31. To further evaluate model robustness, we present results on the Office-31 dataset. We show 5-fold cross-validation results on the test set (for each target domain) for all 5 models for all 6 adaptation tasks in Table VI. Each column denotes a source domain, and each row a target domain. We present the average performance over all 6 tasks for each model to provide a summary of the model performance.

In Table VI we observe that for all 6 adaptation tasks, the MCC model performs best and has the highest overall average performance of 89.5%. For all 5 models and for all 6 adaptation tasks, the standard deviations over the 5 cross-validation folds range from 0.0 to 2.9. We observe that with Office-31, MCC is now the top performing model and DANN is toward the bottom. *However, all 5 models still perform close on average, with the AFN model still a top performing model.*

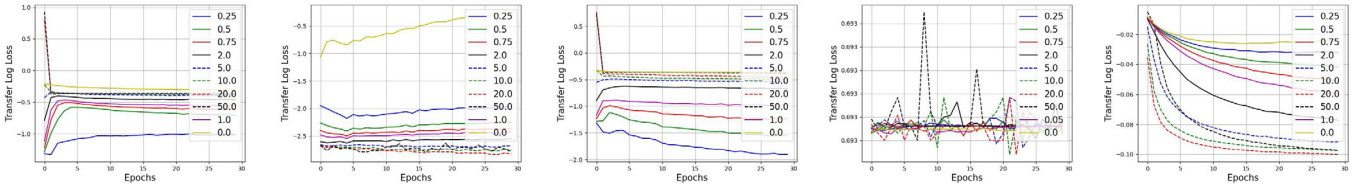
This relative change in model performance order points at the importance of needing to tune the MCC model significantly. As another set of experiments, we also run all 5 models for all 6 adaptation tasks from the Office-31 dataset without any tuning of hyperparameters. Instead, we simply take the best hyperparameters from the DomainNet tuning and apply them when using the Office-31 dataset. When doing this, we observe that the DANN, JAN, CDAN, and AFN models still obtain fairly comparable average test performances of: 86.9%, 86.5%, 88.1%, and 86.4%, respectively. However, the MCC model only achieved an average test accuracy of 24.1%; drastically less than the tuned average test accuracy.

Additionally, over all learning rates and weight decay combinations explored for each model and all 6 adaptation tasks, we observe that for 4 out of 6 of the tasks, the MCC model has the highest variability in performance when varying learning rates and weight decays. *These two pieces of evidence lead us to conclude that the MCC model is fairly unstable across datasets and requires careful, time-consuming*



(a) DANN (b) JAN (c) CDAN (d) AFN (e) MCC

Figure 3: *Classification* loss curves for varied loss trade-off values (y-axis log-scale). Clipart to painting task for full dataset.



(a) DANN (b) JAN (c) CDAN (d) AFN (e) MCC

Figure 4: *Transfer* loss curves for varied loss trade-off values (y-axis log-scale). Clipart to painting task for full dataset.

hyperparameter tuning for each new dataset in order to assure that it outperforms other models in the case of Office-31. However, in the case of DomainNet as previously noted, even with careful hyperparameter tuning, MCC performs the poorest in our study.

C. Transfer Loss Analysis

Next, we present results on classification loss, transfer loss, and classification accuracy curves for all 5 models. Then, we visualize learned feature spaces for a set of adaptation tasks.

Loss Curve Analysis. Addressing the second objective of transfer loss utility in our work, we present classification and transfer loss curves in Figures 3 and 4, respectively, for a variety of loss trade-off hyperparameter values as mentioned in Section IV-B. Loss curves are shown for just the clipart-to-painting adaptation task for the full dataset. The default trade-off value for all models except AFN is 1.0, AFN had a default value of 0.05. All default values are plotted in magenta in Figures 3 and 4. Loss values are plotted on a log-scale. In Figure 4, for the AFN model, the y-axis is all 0.693 as changes in the transfer loss vary only to the seventh decimal place over the 30 epochs; meaning, very little learning occurring. In Figure 5, we also present the target domain (painting) test accuracy plots for each model for each loss trade-off value.

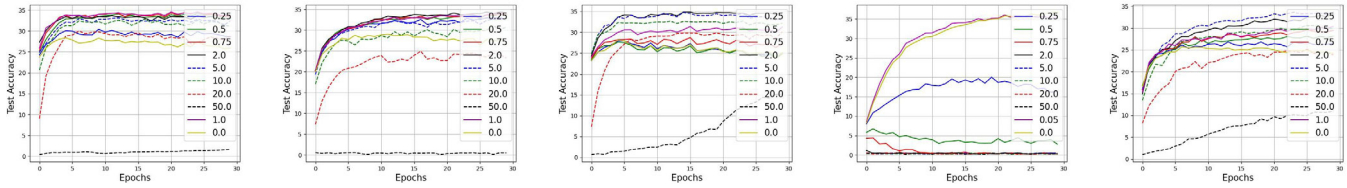
In Figure 3, across all 5 models, we generally see the classification loss steadily decreasing and flattening out during training. The exception is with trade-off value of 50 (heavily weighting transfer loss) for JAN and CDAN where the classification loss increases or stays flat. For AFN, for the majority of trade-off values, the classification loss increases except for the default trade-off value and small trade-off values. Even for larger trade-off values that weigh transfer loss more of 10.0 and 20.0 for DANN, JAN, CDAN, and MCC and 50.0 for MCC, we still observe a decreasing classification loss, indicating learning with respect to the classification loss.

An interesting observation we make is with respect to the transfer loss curves for each model depicted in Figure 4. *For JAN, CDAN, AFN, and DANN we observe little to no learning occurring with respect to the transfer losses proposed for each model.* Each model claims that their transfer loss function would be aligning domains. We should observe this as an improvement in learning via the transfer loss curve. However, even though we vary the weighting of the two losses during training, we consistently observe that transfer loss changes are minimal and sometimes even experience increases, even when heavily weighting it with values up to 50. Given this observation, for a variety of loss trade-off values and models, *the utility of DA transfer loss functions may not be achieving the goal they are designed for: a full alignment of domains.* Given that the inclusion of transfer loss functions in domain adaptation models leads to improvement over traditional transfer learning, *we speculate that these transfer loss functions may be simply acting as a regularization technique to generalize better to the target domain data.*

The only model whose transfer loss curves appear to be decreasing to some degree is the MCC model. However, the actual decrease over 30 epochs is approximately 0.08, a small fraction of the overall loss decrease that occurs during training. Although, even with the transfer loss decreasing only slightly for MCC, we have consistently seen MCC perform the poorest.

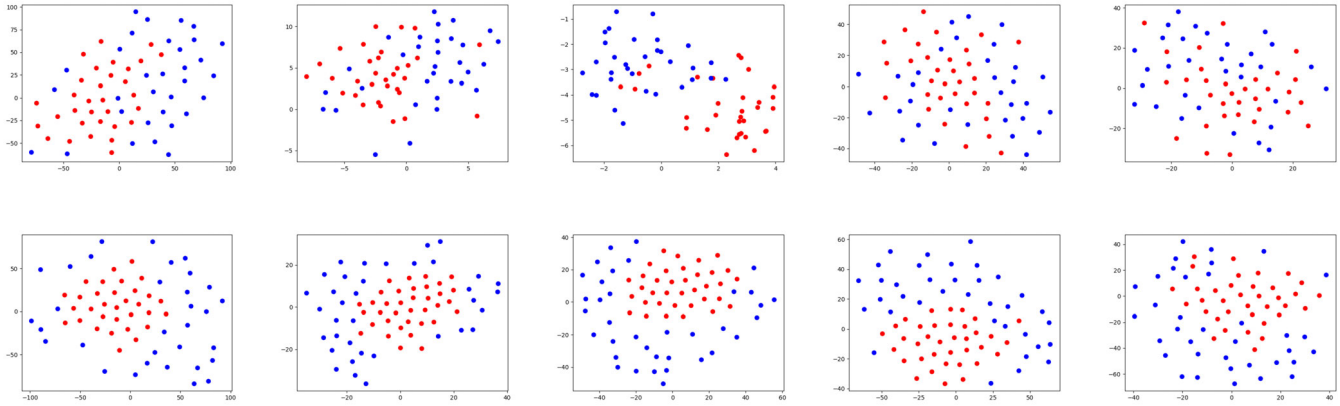
In Figures 3 and 4 we demonstrate that classification loss is the dominating factor in learning, while transfer loss is not contributing to learning during training. *This raises concern over the utility of the proposed transfer loss functions in that they are not helping much to learn a transferable representation between source and target domains.*

Further, when we compare the loss curves to what we observe with respect to the target domain test accuracy curves in Figure 5, we see that for a majority of trade-off values, the test accuracies continue to increase even when the corresponding transfer loss curves remain unchanged or even start to increase. This further supports the observation that the classification loss, regardless of trade-off weight, is the dominating factor in learning. To summarize, given that the



(a) DANN (b) JAN (c) CDAN (d) AFN (e) MCC

Figure 5: Target domain test accuracies for varied loss trade-off values. Clipart to painting task for full dataset.



(a) C2I (b) C2P (c) C2Q (d) C2R (e) C2S

Figure 6: T-SNE plots for the airplane class. Top: DANN, Bottom: MCC. Red is the clipart source domain, blue is the target domain. a-e are 5 of 30 adaptation tasks. (C=clipart, I=infograph, P=painting, Q=quickdraw, R=real, S=sketch)

inclusion of a transfer loss has been shown to still improve target domain classification over traditional transfer learning, this leads us to suggest that these transfer loss functions may be acting simply as regularization techniques.

Feature space alignment. We present T-SNE plots showing feature distributions for 5 adaptation tasks for 2 models in Figure 6. Due to space constraints, we select the DANN model, a consistently top performing model, and the MCC model, the newest albeit poorest performing model. We show the 5 adaptation tasks with clipart used as the source domain and all other domains used as target. Additional plots for other models are available in our Github repository. For each adaptation task, we plot the learned representations of data for each domain, with red indicating source domain, and blue the target domain, for a single class (airplane).

When transferring knowledge between domains, we anticipate overlapping distributions of data representations for the same class. With Figure 6, we can check on the effectiveness of each transfer loss function for a variety of transfer tasks by observing if the two domains have overlapping distributions. *For a majority of the tasks we see a visual separation of the source (red) and target (blue) domain representations.* The test accuracies for these 5 tasks, left to right, for DANN were: 19.1%, 34.3%, 12.7%, 50.4%, and 40.7%. And for MCC they were: 13.5%, 33.4%, 12.9%, 49.0%, and 39.3%. We see in particular that for the clipart-to-quickdraw (C2Q) transfer task, a fairly noticeable separation between domains remains. Correspondingly, we see fairly low accuracies of $\sim 12\%$. For the clipart-to-infograph (C2I) task, DANN has a higher performance of 19.1% compared to MCC's 13.5%. Correspondingly in Figure 6 there is a slightly larger overlap of distributions for DANN than observed for MCC for the C2I task. *With this separation of domains remaining even after applying domain adaptation, this is further evidence that supports that the transfer loss functions are not necessarily achieving their goal of aligning domains.* However, since accuracies are boosted with domain adaptation, transfer loss functions are indeed having some impact, possibly as regularization.

VI. Discussion and Conclusion

In this work, we provide a detailed, unbiased, empirical evaluation of 5 state-of-the-art deep unsupervised DA models to assess model robustness across different benchmark domain adaptation tasks and datasets. This work provides a valuable objective empirical analysis for model choice for a variety of data scenarios. Surprisingly, we observe that AFN and DANN, the oldest model, are consistently top performers for all 3 training data availability scenarios on the DomainNet benchmark. Interestingly, one of the oldest UDA methods, DANN, outperforms newer models; most notably the 2020 MCC model, with the latter performing at or near the bottom for a majority of adaptation tasks. This observation suggests that future DA research thoroughly evaluate baseline models under a rich variety of conditions when comparing them to newly proposed models.

Furthermore, we show that for Office-31, while MCC becomes the top-performing model, all 5 models on average

remain close in performance to one another. We observe, however, that the MCC model is more sensitive to hyperparameter choice across datasets, requires careful tuning for new datasets, and is not guaranteed to be the best model even when tuned (i.e. on DomainNet). This suggests that the newest model is not the most robust choice for practitioners and more careful analysis needs to be done when proposing new models.

Another important take-away of our study is that the dominating factor for learning is the classification loss, even when weighting the two losses (classification and transfer loss) with a wide range of weight values. The transfer loss curves indicate that little to no learning occurs during training. We thus speculate that their effectiveness in target domain classification is due more to them acting as a regularizer instead of truly aligning domains. Future work is thus needed to confirm the usefulness of the proposed transfer loss functions.

Further, as illustrated by visual inspection of T-SNE plots, the learned representations for a (source, target) pair of domains are not mapped to a significantly overlapping feature space - as would have been expected from DA models.

Lastly, in the future, newer approaches, such as, two-staged training where classification and transfer loss are optimized separately and not concurrently, should be compared against alternate approaches, such as, the ones studied by our work.

ACKNOWLEDGMENTS

This research is supported by the US ARMY, ACC-APG-RTP, Cooperative Agreement W911NF-19-2-0112, US. CDCC Army Research Lab and NSF grants CCF-2006738 and NRT-HDR-2021871.

REFERENCES

[1] A. Torralba and A. Efros, "Unbiased look at dataset bias," in *IEEE Conf. on Comput. Vision and Pattern Recognition*, 2011, pp. 1521–1528.

[2] J. Donahue *et al.*, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Int. Conf. on Mach. Learn.* PMLR, 2014, pp. 647–655.

[3] K. Saleh *et al.*, "Domain adaptation for vehicle detection from bird's eye view lidar point cloud data," in *IEEE/CVF Int. Conf. on Comput. Vision Workshops*, Oct 2019.

[4] Y. You *et al.*, "Exploiting playbacks in unsupervised domain adaptation for 3d object detection in self-driving cars," in *2022 Int. Conf. on Robotics and Autom.*, 2022, pp. 5070–5077.

[5] T. Sun *et al.*, "Shift: A synthetic driving dataset for continuous multi-task domain adaptation," in *IEEE/CVF Conf. on Comput. Vision and Pattern Recognition*, 2022, pp. 21 371–21 382.

[6] C. S. Perone, P. Ballester, R. C. Barros, and J. Cohen-Adad, "Unsupervised domain adaptation for medical imaging segmentation with self-ensembling," *NeuroImage*, vol. 194, pp. 1–11, 2019.

[7] H. Guan and M. Liu, "Domain adaptation for medical image analysis: a survey," *arXiv:2102.09508*, 2021.

[8] R. Wang, P. Chaudhari, and C. Davatzikos, "Embracing the disharmony in medical imaging: A simple and effective framework for domain adaptation," *Med. Image Analysis*, vol. 76, p. 102309, 2022.

[9] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent adv." *IEEE signal processing magazine*, vol. 32, no. 3, pp. 53–69, 2015.

[10] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *arXiv:1702.05374*, 2017.

[11] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[12] D. Saunders, "Domain adaptation and multi-domain adaptation for neural mach. translation: A survey," *arXiv:2104.06951*, 2021.

[13] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learn. with joint adaptation networks," in *Int. Conf. on Mach. Learn.* PMLR, 2017, pp. 2208–2217.

[14] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in *IEEE/CVF Int. Conf. on Comput. Vision*, 2019, pp. 1426–1435.

[15] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *The journal of Mach. Learn. research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[16] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Int. Conf. on Mach. Learn.* PMLR, 2015, pp. 1180–1189.

[17] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *IEEE Conf. on Comput. Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[18] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *arXiv:1705.10667*, 2017.

[19] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv:1412.3474*, 2014.

[20] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learn. transferable features with deep adaptation networks," in *Int. Conf. on Mach. Learn.* PMLR, 2015, pp. 97–105.

[21] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," *Adv. in neural inform. processing syst.*, vol. 29, 2016.

[22] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *IEEE Int. Conf. on Comput. vision*, 2015, pp. 4068–4076.

[23] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *AAAI Conf. on Artif. Intell.*, 2018.

[24] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, "Transferable attention for domain adaptation," in *AAAI Conf. on Artif. Intell.*, vol. 33, no. 01, 2019, pp. 5345–5352.

[25] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *Int. Conf. on Mach. Learn.* PMLR, 2019, pp. 7404–7413.

[26] X. Peng *et al.*, "Moment matching for multi-source domain adaptation," in *IEEE/CVF Int. Conf. on Comput. Vision*, 2019, pp. 1406–1415.

[27] Y. Li, L. Yuan, Y. Chen, P. Wang, and N. Vasconcelos, "Dynamic transfer for multi-source domain adaptation," in *IEEE/CVF Comput. Vision and Pattern Recognition*, 2021, pp. 10 998–11 007.

[28] H. Zhao *et al.*, "Adversarial multiple source domain adaptation," *Adv. in neural inform. processing syst.*, vol. 31, 2018.

[29] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, "Deep cocktail network: Multi-source unsupervised domain adaptation with category shift," in *IEEE Comput. Vision and Pattern Recognition*, 2018, pp. 3964–3973.

[30] X. Peng, Z. Huang, X. Sun, and K. Saenko, "Domain agnostic learn. with disentangled representations," in *Int. Conf. on Mach. Learn.* PMLR, 2019, pp. 5102–5112.

[31] P. Panareda Busto and J. Gall, "Open set domain adaptation," in *IEEE Int. Conf. on Comput. Vision*, 2017, pp. 754–763.

[32] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *Eur. Conf. on Comput. Vision*, 2018, pp. 135–150.

[33] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *IEEE Comput. Vision and Pattern Recognition*, 2018, pp. 8156–8164.

[34] G. Csurka, F. Baradel, B. Chidlovskii, and S. Clinchant, "Discrepancy-based networks for unsupervised domain adaptation: a comparative study," in *IEEE Int. Conf. on Comput. Vision Workshops*, 2017, pp. 2630–2636.

[35] J. Jiang, B. Chen, B. Fu, and M. Long, "Transfer-learning-library," https://github.com/thuml/Transfer-Learning-Library, 2020.

[36] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Mach. Learn. Research*, vol. 13, no. 1, pp. 723–773, 2012.

[37] Y. Jin, X. Wang, M. Long, and J. Wang, "Minimum class confusion for versatile domain adaptation," in *Eur. Conf. on Comput. Vision*. Springer, 2020, pp. 464–480.

[38] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Eur. Conf. on Comput. Vision*. Springer, 2010, pp. 213–226.