Flexible Deep Transfer Learning by Separate Feature Embeddings and Manifold Alignment

Samuel Rivera[a], Joel Klipfel[b], and Deborah Weeks[c]

[a]Matrix Research, Dayton, USA;
[b]University of Kentucky, Lexington, USA;
[c]George Washington University, Washington DC, USA

ABSTRACT

Object recognition is a key enabler across industry and defense. As technology changes, algorithms must keep pace with new requirements and data. New modalities and higher resolution sensors should allow for increased algorithm robustness. Unfortunately, algorithms trained on existing labeled datasets do not directly generalize to new data because the data distributions do not match. Transfer learning (TL) or domain adaptation (DA) methods have established the groundwork for transferring knowledge from existing labeled source data to new unlabeled target datasets. However, current DA approaches assume similar source and target feature spaces and suffer in the case of massive domain shifts or changes in the feature space. Existing methods assume the data are either the same modality, or can be aligned to a common feature space. Therefore, most methods are not designed to support a fundamental domain change such as visual to auditory data.

We propose a novel deep learning framework that overcomes this limitation by learning separate feature extractions for each domain while minimizing the distance between the domains in a latent lower-dimensional space. The alignment is achieved by considering the data manifold along with an adversarial training procedure. We demonstrate the effectiveness of the approach versus traditional methods with several ablation experiments on synthetic, measured, and satellite image datasets. We also provide practical guidelines for training the network while overcoming vanishing gradients which inhibit learning in some adversarial training settings.

Keywords: transfer learning, domain adaptation, adversarial learning, deep learning, machine learning, automatic target recognition, classification

1. INTRODUCTION

Object recognition, the process of using machines to classify objects from sensor data, is a key enabler across industry and defense that supports automation and situational awareness. Deep learning (DL) algorithms have achieved state-of-the-art performance on current vision tasks by learning good feature representations for large labeled datasets.[1] Labeling datasets relevant to the sensor domain, however, is often prohibitively expensive due to the abundance of new data available for emerging sensor modalities, and the cost of collecting and labeling appropriate data for all relevant operating conditions (environment, target, and sensor state). Using similar, labeled datasets or generating synthetic datasets mitigates some of these issues, but typically fails to generalize well. Namely, classification performance on the *target* dataset of interest suffers when the *source* dataset used for training has a much different probability distribution.[2]

Transfer learning (TL) or domain adaptation (DA) methods have established the groundwork for transferring knowledge from existing labeled source data to these new unlabeled target datasets to overcome the burden of requiring labeled datasets. Unfortunately, most DA approaches assume similar source and target feature spaces and suffer in the case of massive domain shifts or are not meant to handle a fundamental change in the feature space. Therefore, most existing methods assume the data are either the same modality, or can be aligned to a common feature space. We illustrate the challenge in Fig. 1. In this example, the source classes A and B form

Further author information: (Send correspondence to S.R.)
S.R.: E-mail: samuel.rivera@matrixresearch.com

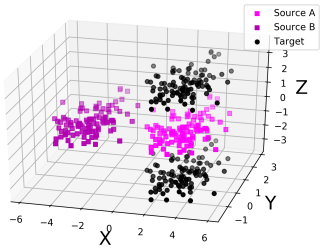

Figure 1. Illustration of a cross-feature-space transfer challenge.

distinct clusters on the xy-plane that are separated at the origin by the yz-plane. The target data also form two distinct clusters, but are separated by the xy-plane and completely overlap with source class B of projects onto the xy-plane. This example shows a case where the discriminative feature information from the source data (x values are informative) has no bearing on category identity for the target samples.

This challenge of learning to transfer across fundamentally different features spaces, or where discriminative information from one domain is not helpful for the other, is referred to as heterogeneous transfer learning (HTL), and it has received less attention than its TL counterpart. See Day and Khoshgoftaar[3] for a recent survey of this approach which has received relatively less attention within the DL literature. We propose a novel deep learning algorithm called direct sum domain adversarial transfer (DiSDAT) that can transfer across very different domains by learning separate feature extractions for each domain while minimizing the distance between the two domains in a latent lower-dimensional space. A key idea behind the DiSDAT network is that source and target feature spaces may be fundamentally different so source and target inputs should be modeled as existing in separate feature spaces as opposed to assuming a common input feature space.

A key assumption is that the source and target data, although residing in different feature spaces, exist on lower dimensional manifolds where the probability distributions are isomorphic, or can be aligned by some transformation of the original feature spaces. Therefore, we align the source and target by learning a feature extraction that preserves the underlying manifold while minimizing the divergence between the source and target distributions in the latent space as illustrated in Fig. 2. We accomplish this by define a suitable objective function that penalizes divergence between the source and target distributions in the latent space. Once aligned in a latent space, we readily classify target data using a classifier trained on labeled source data.

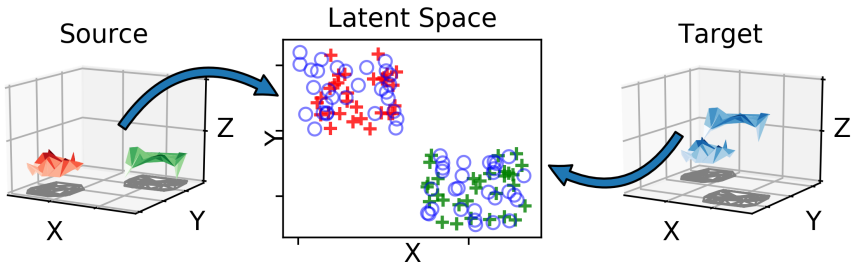

Figure 2. Illustration of the projection from two different feature spaces to a common latent space.

The paper is organized as follows: We describe related work in Sec. 2 before providing DiSDAT details in Sec. 3. We demonstrate the effectiveness of the approach versus traditional methods with several ablation experiments on synthetic, measured, and satellite image datasets in Sec. 4. We also provide practical guidelines for training the network while overcoming vanishing gradients which inhibit learning in some adversarial training settings. Results will show that incorporating a generative manifold-preserving framework within the network architecture alleviates degenerate learning that is often resolved by initializing the target feature extractor with the source extractor.[4] Finally, we close with a discussion and conclusions.

2. RELATED WORK

TL remains an active area of research because of the important applications described above. See Pan & Yang[2] for a thorough review of TL across classification, clustering, and regression methods. Our work falls under the subcategory Pan & Yang refer to as *transductive transfer learning*, where the source and target tasks are the same but over different domains. Furthermore, we consider the case where labels exist for the source data and the algorithm has access to only unlabeled target data. This type of transfer, where we want to learn a classifier from source data but apply it to a different target dataset is also called DA, which has been a very active area of research recently within DL for visual applications.[5,6] In this section we do not attempt to cover all avenues of work, but instead refer to relevant review papers for context and summarize the most relevant work here.

One highly relevant method by Tzeng and colleagues, called adversarial discriminative domain adaptation (ADDA),[4] establishes a framework for DA where separate source and target feature extraction mappings are learned that minimize the *distance* between source and target while being discriminative for the source data. A classifier learned with the source data then extends to target data after the mapping step. This general framework allows for choice of i) whether source and target feature extractions should share weights, ii) similarity (or divergence) measure between source and target, and iii) whether the algorithm employs a generative or purely discriminative architecture. Importantly, many of the recent DA methods can be described as a variant of this framework. Our approach also fits within this general framework of aligning the source and target in a discriminative latent feature space, but with important algorithmic distinctions for each major component. Therefore, we describe the key related literature in the context of the ADDA framework.

Network Weight Sharing: A major drawback of ADDA and related methods is that they use the source mapping parameters to initialize the target mapping parameters, then fine-tune the target mapping parameters[7] to prevent the network from learning a degenerate target mapping. This means that source and target must be pre-aligned to the same feature space. In our early experiments we also found that learning a separate target mapping when using a domain adversarial loss[8] for source and target alignment led to poor target mapping solutions when starting with random initial weights. We overcame this by using an autoencoder to preserve the underlying structure of the original target data while learning the mappings. This allows us to learn a target mapping from random initializations and remove the requirement of having mirrored network architectures for source and target feature mapping. Consequently, this allows for much larger domain shifts and completely different modality shifts, such as in HTL, that would not be possible by initializing with the source mapping because the target mapping does not depend on the source feature mapping.

A notable example of HTL within the DL family is the Hybrid Heterogeneous Transfer Learning (HHTL) algorithm of Zhou *et al.*[9] HHTL first learns separate higher-level representations of source and target data using marginalized stacked denoised autoencoder (mSDA),[10] a particular variant of auto-encoders that learn increasingly higher-level data representations. Then, target samples in the latent space are mapped to the source latent space where a source classifier applies across both source and target. Unfortunately, learning the mapping between source and target latent representations requires corresponding samples between source and target, which limits the practical application of HHTL. Our method does not require corresponding samples between source and target, so it is more applicable in practice.

Divergence Measure: Another important consideration within the general DA framework is the choice of divergence measure between source and target as well as the training procedure. ADDA uses a *GAN-loss*, a domain classifier loss with reversed labels, to adapt the target to the source in the latent space. The GAN-loss builds on the seminal work of Ganin *et al.*[8] that uses the domain classification loss along with *gradient reversal* to maximize the domain classification error, essentially learning feature mappings such that the source and target are indistinguishable in the latent space. Using the GAN-loss instead of the gradient reversal achieves the same goal but avoids vanishing gradients that occur in practice when training DA networks using the gradient reversal trick.[4]

Unfortunately, the domain classification error only acts as a *proxy* for the similarity between the source and target distributions in the latent space and does not explicitly measure the distance between the source and target distributions. An alternative discrepancy measure, the maximum mean discrepancy (MMD),[11,12] more directly measures the difference between source and target distributions through the distance in a reproducing

kernel Hilbert space (RKHS). Long *et al.*[13,14] applied MMD as a key component of the deep adaptation network (DAN) algorithm for minimizing the discrepancy between source and target in higher level network activation layers. The idea has been extended to the joint maximum mean discrepancy (JMMD) in joint adaptation network (JAN)[15] to better minimize the joint discrepancy across higher level layers and lower level feature extraction layers.

Taking inspiration from the subspace transfer learning work of Si Si *et al.*[16] and Mendoza *et al.*,[17] we use a quadratic form of Bregman Divergence (BD)[18] as the measure of dissimilarity between source and target distributions in the latent space. This allows us to explicitly model differences between source and target distributions in the latent space and learn feature mappings such that source and target distributions match. Unlike DAN, we do not try to match network outputs for source and target across the network, but instead focus on aligning distributions after feature extraction. Once source and target data are aligned in a common latent feature space, then the higher classification layers do not need to be further altered for the target.

Other researchers have used the Kullback–Leibler (KL)-divergence,[19] another special case of BD, to penalize the divergence between two vectors that represent probability mass functions. This type of penalty has been used as an alternative to the standard cross-entropy loss for multi-class classification. It has also been applied by Zhuang *et al.* in the transfer learning with deep autoencoders (TLDA) algorithm[20] to penalize the distance between source and target in the embedded space. A major shortcoming of their method is that the distributions are not actually aligned. Instead, the difference between the *sample means* of source and target in the embedded space are penalized without penalizing the difference between the overall distributions. TLDA also deserves mention because it applies an autoencoder similarly to our algorithm, but unlike our algorithm assumes tied input source and target feature mapping as with most algorithms.

Architecture: The final point of discussion is the specific network architecture. Tzeng *et al.*[4] frame the discussion in terms of whether networks are purely discriminative, or whether they include a generative modeling component. Generative models typically use a variant of generative adversarial networks (GANs)[21] or autoencoders. Tzeng and colleagues argue that it is not always necessary to learn a generative data model when learning a discriminative model for classification. We sympathize with this view, but acknowledge that cases exist where generative models can support discriminate model learning. Sankaranarayanan *et al.*,[22] for example, use a GAN along with the learned embedding for DA. They argue that learning the generative model produces more meaningful gradients for backpropagation.

GANs have also been applied in the coupled generative adversarial nets (CoGAN) algorithm[23] to learn shared high-level representations that are common across domains. This is similar to the idea of using mSDA to learn higher level representations of data[10] that work across domains. Alternatively, GANs can be used along with *cycle consistency* as in cycle-consistent adversarial domain adaptation (CyCADA)[24] to adapt data across domains such that both feature-level and higher-level semantic information is preserved.

Besides learning high-level representations, a generative model can learn the underlying data structure, or manifold. The approach by Gong *et al.* projects the source and target in a Grassmann manifold, then aligns the data using the *geodesic flow kernel* approach.[25] Our approach is similar to this in that it learns and aligns the underlying source and target data manifolds, but the manifold is learned through an autoencoder and the alignment is achieved by divergence minimization. Others have used graph convolutional network (GCN)[26] to model the underlying data structure for semi-supervised DL.

3. METHODS

3.1 Direct Sum Domain Adversarial Transfer Network (DiSDAT)

We model the input feature space mathematically as the algebraic sum of the source and target feature spaces. More formally the source space $\mathcal{S}$ and target space $\mathcal{T}$ give rise to a combined input space, $\mathcal{S} \bigoplus \mathcal{T}$. Separate feature extraction modules ($\mathcal{F}_\mathcal{S}$, $\mathcal{F}_\mathcal{T}$) embed data from either $\mathcal{S}$ or $\mathcal{T}$ into a common latent feature space, $\mathcal{H}$. Separate feature embeddings allow the source and target to differ arbitrarily–even in dimensionality–as long as the feature embeddings preserve isomorphisms between the class conditional probability distributions of the source and target data. As indicated in Figure 3.1, DiSDAT learns the latent feature space embeddings $\mathcal{F}_\mathcal{S}$, $\mathcal{F}_\mathcal{T}$ using separate auto-encoders. The auto-encoder used to train $\mathcal{F}_\mathcal{S}$ (or $\mathcal{F}_\mathcal{T}$) first maps from $\mathcal{S}$ (or $\mathcal{T}$) into the common

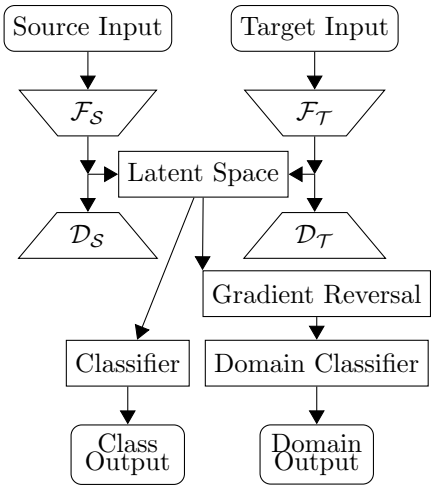

Figure 3. Illustration of the DiSDAT Network.

latent feature space, and then back out to $\mathcal{S}$ (or $\mathcal{T}$) with the decoders $\mathcal{D}_\mathcal{S}$ (or $\mathcal{D}_\mathcal{T}$). We learn the embedding $\mathcal{F}_\mathcal{S}$(or $\mathcal{F}_\mathcal{T}$) from $\mathcal{S}$ (or $\mathcal{T}$) into the common latent feature space and the corresponding decoder by applying standard backprop procedure with the mean square error (MSE) of the reconstruction loss between the input and output of an autoencoder. Using an autoencoder to train the feature extraction modules $\mathcal{F}_\mathcal{S}$ and $\mathcal{F}_\mathcal{T}$ allows DiSDAT to account for the underlying geometric structure of $\mathcal{S}$ and $\mathcal{T}$, when considered as Grassmannian manifolds. Once features are embedded in a common latent space, DiSDAT assigns to a category label using a classifier $\mathcal{C}$ that is trained using backprop updates along with the standard cross-entropy loss.

The key feature of DiSDAT is that it learns an embedding into the common latent feature space $\mathcal{H}$ that aligns the source and target data probability distributions. DiSDAT does so by employing the quadratic-type Bregman divergence discussed in Si *et al.*[16] and Mendoza *et al.*[17] to measure differences between the respective probability distributions $P_\mathcal{S}$ and $P_\mathcal{T}$ for the embedded source and target data, and then updates the source and target data via gradient descent. The quadratic-type Bregman divergence from Si *et al.*[16] and Mendoza *et al.*[17] is given by

$$D(P_\mathcal{S} \| P_\mathcal{T}) = \int (P_\mathcal{S}(\mathbf{y}) - P_\mathcal{T}(\mathbf{y}))^{\mathbf{2}} \, \mathrm{d}\mathbf{y}, \tag{1}$$

where the above integral is taken over the common latent feature space. By using Gaussian kernel density estimation (KDE), Si shows that (1) can be approximated on the embedded source and target data by

$$\begin{aligned} D(P_\mathcal{S} \| P_\mathcal{T}) \approx & \frac{1}{(n_\mathcal{S})^2} \sum_{j=1}^{n_\mathcal{S}} \sum_{k=1}^{n_\mathcal{S}} G_{\Sigma_{\mathcal{S},\mathcal{S}}}(\mathbf{y}_k^\mathcal{S} - \mathbf{y}_j^\mathcal{S}) \\ & + \frac{1}{(n_\mathcal{T})^2} \sum_{j=1}^{n_\mathcal{T}} \sum_{k=1}^{n_\mathcal{T}} G_{\Sigma_{\mathcal{T},\mathcal{T}}}(\mathbf{y}_k^\mathcal{T} - \mathbf{y}_j^\mathcal{T}) \\ & - \frac{2}{n_\mathcal{S}\, n_\mathcal{T}} \sum_{j=1}^{n_\mathcal{S}} \sum_{k=1}^{n_\mathcal{T}} G_{\Sigma_{\mathcal{S},\mathcal{T}}}(\mathbf{y}_k^\mathcal{T} - \mathbf{y}_j^\mathcal{S}). \end{aligned} \tag{2}$$

In Equation (2), $\{\mathbf{y}_k^\mathcal{S}\}_{k=1}^{n_\mathcal{S}}$ and $\{\mathbf{y}_k^\mathcal{T}\}_{k=1}^{n_\mathcal{T}}$ are respectively used to denote the embedded source and target data, where $n_\mathcal{S}$ and $n_\mathcal{T}$ are the respective cardinalities of the source and target datasets. We further use G_{Σ_*} to denote the Gaussian kernel

$$G_{\Sigma_*}(\mathbf{y}) = \exp\left(-\frac{1}{2}\mathbf{y}^T \Sigma_*^{-1} \mathbf{y}\right)$$

with covariance matrix Σ_*, where $\Sigma_{\mathcal{S}}$ and $\Sigma_{\mathcal{T}}$ respectively represents the covariance matrices for the embedded source and target data, and the notation $\mathbf{y}^T$ denotes the transpose vector $\mathbf{y}$. In order to simplify notation, we follow Si's lead and define $\Sigma_{*,*} := \Sigma_* + \Sigma_*$. For example, under this notational convention, $\Sigma_{\mathcal{S},\mathcal{T}}$ represents the sum $\Sigma_{\mathcal{S},\mathcal{T}} = \Sigma_{\mathcal{S}} + \Sigma_{\mathcal{T}}$. The loss function L_{Breg} used in DiSDAT to penalize for differences data probability distributions is precisely (2). To perform backpropagation, one can readily show that the derivatives of $D(P_{\mathcal{S}} \| P_{\mathcal{T}})$ with respect to the embedded source and target data are given by

$$\begin{aligned}\frac{\partial}{\partial \mathbf{y}_i^{\mathcal{S}}} D(P_{\mathcal{S}} \| P_{\mathcal{T}}) = \frac{2}{(n_{\mathcal{S}})^2} \sum_{k=1}^{n_{\mathcal{S}}} G_{\Sigma_{\mathcal{S},\mathcal{S}}}(\mathbf{y}_k^{\mathcal{S}} - \mathbf{y}_i^{\mathcal{S}}) \left(\Sigma_{\mathcal{S},\mathcal{S}}\right)^{-1} (\mathbf{y}_k^{\mathcal{T}} - \mathbf{y}_i^{\mathcal{T}}) \\ - \frac{2}{n_{\mathcal{S}}\, n_{\mathcal{T}}} \sum_{k=1}^{n_{\mathcal{T}}} G_{\Sigma_{\mathcal{S},\mathcal{T}}}(\mathbf{y}_k^{\mathcal{T}} - \mathbf{y}_i^{\mathcal{S}}) \left(\Sigma_{\mathcal{S},\mathcal{T}}\right)^{-1} (\mathbf{y}_k^{\mathcal{T}} - \mathbf{y}_i^{\mathcal{S}}) \end{aligned} \tag{3}$$

and

$$\begin{aligned}\frac{\partial}{\partial \mathbf{y}_i^{\mathcal{T}}} D(P_{\mathcal{S}} \| P_{\mathcal{T}}) = \frac{2}{(n_{\mathcal{T}})^2} \sum_{k=1}^{n_{\mathcal{T}}} G_{\Sigma_{\mathcal{T},\mathcal{T}}}(\mathbf{y}_k^{\mathcal{T}} - \mathbf{y}_i^{\mathcal{T}}) \left(\Sigma_{\mathcal{S},\mathcal{S}}\right)^{-1} (\mathbf{y}_k^{\mathcal{T}} - \mathbf{y}_i^{\mathcal{T}}) \\ - \frac{2}{n_{\mathcal{S}}\, n_{\mathcal{T}}} \sum_{k=1}^{n_{\mathcal{S}}} G_{\Sigma_{\mathcal{S},\mathcal{T}}}(\mathbf{y}_k^{\mathcal{S}} - \mathbf{y}_i^{\mathcal{T}}) \left(\Sigma_{\mathcal{S},\mathcal{T}}\right)^{-1} (\mathbf{y}_k^{\mathcal{T}} - \mathbf{y}_i^{\mathcal{S}}). \end{aligned} \tag{4}$$

An alternative approach to divergence minimization for DA is the domain adversarial (DAd) technique of Ganin *et al.*,[27] where the network is optimized to *increase* the domain classification loss by using gradient descent on the reversed domain classifier loss. As noted above, DAd is distinct from BD regularization as DAd does not align probability distributions, but instead uses the domain classifier loss as a proxy for distribution divergence. To compare this popular approach to BD, we implemented DAd within our framework using the standard cross-entropy loss and a gradient reversal layer approach[27] after the embedding layer as shown in Fig. 3.1. In following section we will describe how to toggle between the regularizations by altering the total cost function.

3.2 Network Sculpting Through Loss Customization

To understand the effect of the different features of DiSDAT, we performed a series of ablation studies on various transfer tasks and datasets. Each study consists of 12 separate experiments which are based on modifying the parameters of the DiSDAT loss function:

$$L = \lambda_{\text{AE}_{\mathcal{S}}} L_{\text{AE}_{\mathcal{S}}} + \lambda_{\text{AE}_{\mathcal{T}}} L_{\text{AE}_{\mathcal{T}}} + \lambda_{\text{class}} L_{\text{class}} - \alpha_{\text{DA}} L_{\text{DA}_{\mathcal{S}}} - \alpha_{\text{DA}} L_{\text{DA}_{\mathcal{T}}} + \lambda_{\text{Breg}} L_{\text{Breg}} \tag{5}$$

Here we denote the source and target space auto-encoder loss by L_{AE_*}, the classifier loss by L_{class}, the DAd loss by L_{DA_*}, and the BD loss by L_{Breg}. By setting particular loss parameter values, we can control the specific network components involved during network optimization. The parameter values for each experiment type are shown in Table 1. Note that the final column of Table 1 indicates whether or not a direct sum is used to separately embed $\mathcal{S}$ and $\mathcal{T}$ into the common latent feature space.

Experiment	α_{DA}	$\lambda_{\text{AE}_{\mathcal{S}}}$	$\lambda_{\text{AE}_{\mathcal{T}}}$	λ_{class}	λ_{Breg}	Sep Embedding
Baseline	0	0	0	1	0	No
Domain Adversarial (DA)	0.1	0	0	1	0	No
Bregman Divergence(BD)	0	0	0	1	1	No
Auto-Encoder (AE)	0	1	1	1	0	No
DA, AE	0.1	1	1	1	0	No
BD, AE	0	1	1	1	1	No
Direct Sum (DS)	0	0	0	1	0	Yes
DS, DA	0.1	0	0	1	0	Yes
DS, BD	0	0	0	1	1	Yes
DS, DA, AE	0.1	1	1	1	0	Yes
BD, BS, AE	0	1	1	1	1	Yes
Everything	0.1	1	1	1	1	Yes

Table 1. Experiment Types for Ablation Study

4. EXPERIMENTS

4.1 Network Implementation Details

We used a small convolutional network to establish a proof of concept for DiSDAT. Source and target feature encoders ($\mathcal{F}_\mathcal{S}$ and $\mathcal{F}_\mathcal{T}$) began with a 2D convolution layer having 1 input channel (grayscale images), 16 output channels, a 3×3 kernel, a stride size of 1, and 1 pixel padding. This was fallowing by 2D batch normalization, a ReLU activation, and 2D max pooling with a $2\times$ 2 kernel a stride of 2. Next, we applied a 2nd 2D convolution with 16 input channels, 8 output channels, a 3×3 kernel, a stride of 2 and 1 pixel padding. The second convolution was also followed with batch normalization, ReLU activation, and max pooling but with a stride of 1 pixel. We closed the feature extraction with a fully connected layer having 32 inputs and either 3 or 10 outputs depending on the experiment.

The decoders ($\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$) reversed the convolution procedure, with slight adjustments. We began with a fully connected layer with 32 output channels. Then we applied a 2D transpose convolution operation with 8 input channels, 16 output channels, a 3×3 kernel, a stride size of 2 and no padding. This was followed by ReLU activation and 2D batch normalization before another transpose convolution. The 2nd transpose convolution had 16 input channels, 8 output channels, a 5×5 kernel, a stride size of 3, and a 1 pixel padding. Next we applied ReLU activation and batch normalization. The final 2D transpose convolution had 8 input channels, 1 output channel, a 2×2 kernel, a stride of 2, and 1 pixel padding. We finally applied the hyperbolic tangent.

The classifier network operated on the output of the feature encoders ($\mathcal{F}_\mathcal{S}$ and $\mathcal{F}_\mathcal{T}$). We applied a fully connected layer with 5 output nodes followed by ReLU activation and a final fully connected layer with a number of output nodes equal to the number of classes. The domain classifier was a single fully connected layer with 2 output nodes.

For the BD implementation, parameter choices for the KDE have important consequences regarding accuracy and computation. For this implementation we made a simplifying assumption that latent features are independent, which leads to diagonal covariance matrices that are estimated by simply calculating the feature variances. This leads to trivial inverse covariance matrix estimation by inverting the diagonal values. For numerical reasons, we found it necessary to apply Tikhonov regularization[28] for matrix inversion by adding 0.001 to the variance values before inverting.

4.2 Experiment Setup

Both cross class and cross dataset transfer tasks were used to validate the DiSDAT architecture. For cross class transfer tasks, we took disjoint two class subsets from the same dataset as the source and target domains. Cross dataset transfer tasks involving taking two distinct datasets as source and target domains.

Each experiment for every architecture evaluated was performed using 5 Monte Carlo iterations. Reported results are the average accuracy over the Monte Carlo iterations, and the reported errors for each experiment are the standard deviations across all Monte Carlo iterations. For the two cross-class experiments, it is possible that the transfer network reversed the target class labels. That is to say that the transfer network will assign a 0 or 1 arbitrarily to either of the two possible class options. Therefore, for two-class transfer experiments we transformed accuracy as $\max(\text{accuracy}, 1 - \text{accuracy})$ for each Monte Carlo trial to account for a possible label reversal.

4.3 Databases

The transfer tasks we use to benchmark DiSDAT employ the Fashion-MNIST (FMNIST),[29] MNIST,[30] USPS,[31] xView,[32] and SAMPLE[33] datasets. Both MNIST and USPS consist of gray-scale images of the handwritten digits 0 through 9 that we scaled to 28×28 pixels. The MNIST training set has 60,000 images that were downsampled to a common 7,000 images across our experiments. The USPS training set contains 7,291 images that were downsampled to 7,000 images for our experiments.

Due to the ease with which convolutional neural networks can classify the MNIST images, FMNIST was designed by Zalando to allow researchers to better showcase improvements in neural network architectures by providing researchers with a dataset that is more difficult for networks to classify. FMNIST consists of 28×28

pixel grayscale images of articles of clothing and footwear from ten distinct classes. The FMINST training set has 60,000 images that are evenly split over the 10 classes. Particular subsets were used in our experiments. The 10,000 test images were not used.

xView is a particularly large dataset with approximately 1.0 million satellite images taken at 0.3 meter resolution. The object training images in xView are annotated using bounding boxes, and are labeled according to 60 distinct classes. We generated our xView images by cropping square regions centered at the bounding box locations then scaling to 28×28 pixels. We used a subset of 1000 images from each class of interest across our experiments. A detailed description of the xView dataset is found in a paper by Darius Lam *et al.*[32]

The SAMPLE dataset is a publicly available synthetic and measured SAR imagery dataset of 10 target classes. This set is particularly small, with just 1345 total images across all classes in the synthetic set and 1345 in the measured set. We used the *decibel* format images for our experiments.

4.4 Exp 1A: Tops to Shoes

Table 2 shows the results of using images of dresses and shirts from the FMNIST dataset as the source domain, with FMNIST images of sandals and ankle boots as the target. Results show a best performance when using BD and DAd regularization along with direct sum (DS) feature embeddings and the autoencoder for preserving the underlying data structure. Just using BD regularization had the second best results. Looking more closely at the other components, the two other conditions applying the separate feature extractions along with BD were 3^{nd} and 4^{th} in accuracy. In addition, conditions using only DAd regularization or DAd along with the DS embeddings performed worse than Baseline. These results together suggest that the BD along with the DS together were responsible for the large transfer accuracy over the baseline. Using the DS without the appropriate distribution regularization is not sufficient for transfer.

Experiment Type	Source Accuracy	Target Accuracy
Baseline (3 dims)	94.1%±1.9%	65.3%±5.0%
Domain Adversarial (DAd) (3 dims)	95.8%±0.7%	57.6%±7.5%
Bregman Divergence(BD) (3 dims)	91.3%±2.1%	*80.7%±9.2%*
Auto-Encoder (AE) (3 dims)	95.1%±1.7%	58.5%±5.8%
DA, AE (3 dims)	95.6%±0.6%	58.7%±5.1%
BD, AE (3 dims)	94.2%±1.8%	76.1%±5.4%
Direct Sum (DS) (3 dims)	95.6%±0.9%	52.8%±4.1%
DS, DA (3 dims)	95.4%±1.3%	50.1%±0.2%
DS, BD (3 dims)	96.3%±0.5%	*79.9%±15.4%*
DS, DA, AE (3 dims)	95.7%±0.5%	64.0%±4.9%
DS, BD, AE (3 dims)	96.2%±0.3%	*79.6%±8.7%*
Everything* (3 dims)	95.5%±1.0%	**88.0%±3.1%**

Table 2. FMNIST Transfer Task: source images are of dresses and shirts; target images are of sandals and ankle boots. All experiments used a 3-dimensional latent layer

4.5 Exp 1B: Shoes to Tops

Table 3 shows the results of reversing the source and target domains. In this case, the BD again lead to a large accuracy improvement over the baseline. However, this was the case without needing the DS feature embeddings. Using the DS feature embeddings along with BD or DAd regularization improved transfer accuracy over the baseline, but not as much as just using BD regularization. These results together suggest that for this experiment scenario, a common feature embedding could be found to match the source and target distributions. More work would need to be done to identify the particular important features.

Experiment Type	Source Accuracy	Target Accuracy
Baseline	98.1%±0.8%	56.7%±5.1%
Domain Adversarial (DA)	98.7%±0.4%	55.3%±4.8%
Bregman Divergence(BD)	98.5%±0.7%	*75.1%±2.5%*
Auto-Encoder (AE)	99.1%±0.2%	57.2%±7.3%
DA, AE	99.1%±0.2%	56.8%±9.9%
BD, AE	98.6%±0.4%	**75.7%±1.9%**
Direct Sum (DS)	99.1%±0.4%	51.2%±1.5%
DS, DA	99.1%±0.3%	55.4%±5.8%
DS, BD	99.1%±0.2%	66.3%±5.7%
DS, DAd, AE	99.1%±0.3%	69.6%±14.7%
DS, BD, AE	99.3%±0.1%	67.4%±3.4%
Everything	99.1%±0.3%	69.8%±2.9%

Table 3. FMNIST Transfer Task: source images are of sandals and ankle boots; target images are of dresses and shirts. All algorithms used a 3-dimensional latent space.

4.6 Exp 2A: Cross Digit (01-23) and Cross Class (MNIST-USPS)

The results of the cross-dataset transfer task consisting of images of 0's and 1's from MNIST as the source domain and images of 2's and 3's from the USPS dataset can be found in Table 4. In this experiment we also experimented with the dimensionality of the latent space by using both a 3 and 10-dimensional latent space. This allowed us to investigate the effect of varying the manifold dimension. The intuition is that a higher dimensional latent space can represent more class variability, but also requires more parameters to learn. However, higher fidelity representations do not necessary imply better class discrimination. As in the previous ablation studies, the results shown are based on running 5 Monte Carlo iterations for each experiment. Overall, the smaller 3-dimensional latent space showed the best transfer performance. The architecture applying the DS embeddings with both BD and DAd regularization with the autoencoder did the best, while removing just DAd regularization from the best architecture only led to a small performance drop (97.9% to 96.9% accuracy). As was found in previous experiments, BD regularization without the DS embeddings also gave performance well above the baseline. These results are consistent with the previous, and show that the DS embeddings give a performance boost above a single embedding with divergence minimization, but the particular divergence minimization is important. BD regularization again shows an advantage over DAd regularization. Furthermore, for this two-class transfer case we found a clear advantage using a smaller latent space.

Experiment Type	Source Acc (3 dim)	Target Acc (3 dim)	Source Acc (10 dim)	Target Acc (10 dim)
Baseline	97.7%±1.4%	73.6%±12.6%	99.5%±0.2%	89.5%±6.2%
Domain Adversarial (DA)	97.1%±3.4%	82.4%±14.5%	98.3%±1.4%	82.8%±11.4%
Bregman Divergence(BD)	94.5%±1.8%	94.6%±2.6%	98.8%±0.7%	91.4%±8.0%
Auto-Encoder (AE)	99.1%±0.6%	65.4%±9.4%	97.2%±2.7%	74.4%±16.5%
DA, AE	97.7%±3.2%	69.2%±10.6%	98.4%±1.0%	80.6%±10.4%
BD, AE	95.4%±2.6%	*95.5%±2.4%*	97.0%±1.9%	92.2%±4.3%
Direct Sum (DS)	98.2%±2.0%	65.8%±17.1%	99.1%±0.3%	66.8%±12.8%
DS, DA	99.2%±0.4%	63.6%±17.4%	99.1%±0.3%	63.7%±17.7%
DS, BD	98.7%±0.3%	92.0%±10.7%	97.7%±1.9%	75.7%±13.6%
DS, DA, AE	97.8%±2.0%	65.8%±17.3%	98.5%±0.6%	66.0%±14.4%
DS, BD, AE	98.3%±1.0%	*96.9%±2.5%*	98.7%±0.8%	78.9%±19.9%
Everything	97.9%±1.0%	**97.9%±0.8%**	98.0%±1.0%	73.5%±15.0%

Table 4. MNIST to USPS cross dataset transfer task for digits 0 and 1 to 2 and 3. We experimented with both a 3 and 10-dimensional latent space.

4.7 Exp 2B: Cross Digit (45-39) and Cross Class (USPS-MNIST)

Reversing the study of the previous cross dataset transfer task and trying different number classes, we take images of 4's and 5's from the USPS dataset as the source domain and images of 3's and 9's from MNIST as the target. The results of this study are shown in Table 5. In this experiment, unlike the previous digits experiment

of Sec. 4.6, we found that a simpler architecture with just a single feature embedding and BD with or without the autoencoder gave the best transfer performance. Furthermore, unlike the previous experiment, the larger 10-dimensional latent space worked the best.

Experiment Type	Source Acc (3 dim)	Target Acc (3 dim)	Source Acc (10 dim)	Target Acc (10 dim)
Baseline	99.5%±0.6%	82.5%±3.6%	99.5%±0.3%	79.7%±7.8%
Domain Adversarial (DA)	99.3%±0.9%	81.9%±4.4%	98.7%±1.1%	86.8%±2.6%
Bregman Divergence(BD)	99.5%±0.4%	*92.2%±1.2%*	99.6%±0.3%	**93.4%±0.8%**
Auto-Encoder (AE)	98.4%±1.5%	87.1%±2.0%	99.2%±0.8%	88.4%±1.8%
DA, AE	99.8%±0.1%	86.3%±6.3%	99.4%±0.5%	88.1%±2.2%
BD, AE	99.3%±0.7%	92.1%±1.3%	99.4%±0.4%	**93.4%±1.0%**
Direct Sum (DS)	99.4%±0.8%	52.7%±3.0%	99.5%±0.4%	53.8%±2.2%
DS, DA	99.7%±0.2%	51.4%±0.1%	99.1%±1.0%	53.9%±5.2%
DS, BD	99.4%±0.4%	81.5%±9.4%	99.4%±0.5%	67.3%±15.3%
DS, DA, AE	99.8%±0.1%	55.2%±5.9%	99.5%±0.2%	53.9%±4.9%
DS, BD, AE	99.5%±0.3%	83.9%±8.7%	99.0%±0.9%	53.7%±4.8%
Everything	99.4%±0.3%	77.7%±13.1%	98.8%±1.0%	61.4%±16.4%

Table 5. USPS to MNIST cross dataset transfer task for digits 4 and 5 to 3 and 9. We experimented with both a 3 and 10-dimensional latent space.

4.8 Experiment 3: 10-digit transfer (MNIST and USPS)

Table 6 shows the results of 10 class digit transfer. Surprisingly, on the MNIST to USPS transfer, the BD regularization along with the autoencoder only very slightly improved accuracy over the baseline. This is in contrast to the USPS to MNIST transfer, where BD regularization led to a 10% absolute improvement over the baseline. Another initially surprising result was that dramatic reduction in performance when using the DS embedding. We inspected confusion matrices to more precisely understand the error source. The confusion matrices demonstrated that most samples were classified into a small subset of classes. This reveals that the network had learned to *collapse* multiple classes, an issue called "mode collapse" in the literature. This results from initializing the network with random weights and not having any labels from the target set to separate classes. It also highlights the utility of initializing the target feature extraction network with the source network as in ADDA.[4]

Experiment Type	Source Acc (U→M)	Target Acc (U→M)	Source Acc (M→U)	Target Acc (M→U)
Baseline	89.9%±2.4%	47.3%±3.6%	87.3%±2.2%	73.1%±2.0%
Domain Adversarial (DA)	88.2%±3.1%	50.6%±2.0%	81.8%±5.0%	64.2%±5.0%
Bregman Divergence(BD)	89.8%±5.9%	**57.8%±2.1%**	84.5%±3.9%	69.0%±8.5%
Auto-Encoder (AE)(10 dims)	90.8%±0.9%	49.4%±2.1%	86.8%±3.6%	68.0%±4.3%
DA, AE	88.4%±4.0%	46.2%±2.9%	88.5%±2.4%	69.8%±3.8%
BD, AE	89.1%±2.4%	55.2%±3.7%	80.7%±9.2%	**73.2%±2.5%**
Direct Sum (DS)(10 dims)	91.5%±3.2%	8.5%±2.2%	88.3%±2.7%	8.2%±1.2%
DS, DA	91.6%±3.3%	9.4%±0.6%	87.6%±2.4%	7.8%±1.8%
DS, BD	91.5%±1.7%	8.3%±2.4%	87.7%±1.9%	9.3%±5.0%
DS, DA, AE	91.9%±1.3%	10.4%±1.5%	84.9%±9.8%	9.5%±4.0%
DS, BD, AE	90.8%±3.0%	13.0%±4.1%	87.6%±1.2%	13.7%±5.4%
Everything	93.1%±1.1%	14.9%±1.9%	84.7%±6.0%	8.2%±3.8%

Table 6. USPS (U) to MNIST (M) on left and M to U on right. All experiments used a 10-dimensional latent space.

4.9 Experiment 4: Synthetic to Measured SAR

For the next transfer task, we consider the synthetic-to-measured transfer task where our source domain consists of 10 classes of synthetic SAR images from the SAMPLE dataset[33] and our target domain consists of corresponding measured SAR images. The results are shown in Table 7. Much like in the 10-category experiment of Sec. 4.8, DS embeddings caused mode collapse and poor performance for transfer learning. However, unlike the

previous experiments, DAd outperformed BD regularization, with BD regularization only narrowly outperforming the baseline. Best performance was achieved by a single autoencoder with DAd regularization. The simple autoencoder did nearly as well (40.1% versus 37.7% accuracy).

Experiment Type	Source Accuracy	Target Accuracy
Baseline	85.7%±9.9%	31.4%±5.4%
Domain Adversarial (DA)	84.1%±6.0%	31.7%±8.1%
Bregman Divergence(BD)	67.8%±19.7%	33.2%±5.1%
Auto-Encoder (AE)(10 dims)	61.7%±16.5%	37.7%±9.3%
DA, AE	78.4%±8.8%	**40.1%±3.2%**
BD, AE	49.5%±14.7%	33.3%±12.0%
Direct Sum (DS)(10 dims)	78.8%±11.1%	10.3%±2.2%
DS, DA	82.6%±17.4%	10.2%±1.3%
DS, BD	92.3%±8.0%	11.3%±3.8%
DS, DA, AE	80.7%±15.1%	8.5%±4.4%
DS, BD, AE	89.8%±5.8%	10.7%±3.1%
Everything	79.7%±12.1%	13.3%±0.9%

Table 7. SAMPLE Synthetic to Measured Cross Dataset Transfer Task. All experiments used a 10-dimensional latent space.

4.10 Experiment 5: Cross Vehicle Satellite Imagery

In the next experiment of Table 8 we consider a two cross category transfer experiment on the xView satellite imagery dataset.[32] As in the Experiments of Rivera *et al.*,[34] for source we used 'bus' and 'cargo truck' while we used 'passenger vehicle' and 'utility truck' as the target classes. As with previous 2 cross class transfer experiments, the DS embedding gave a slight performance edge over the single feature embedding. However, the results are only slightly better than chance while being better than the baseline.

Experiment Type	Source Accuracy	Target Accuracy
Baseline (3 dims)	75.6%±3.1%	52.1%±0.6%
Domain Adversarial (DA) (3 dims)	78.0%±1.0%	52.9%±1.1%
Bregman Divergence(BD) (3 dims)	66.6%±9.3%	53.5%±1.9%
Auto-Encoder (AE)(3 dims)	74.1%±2.2%	52.6%±1.5%
DA, AE	74.6%±5.8%	53.0%±1.1%
BD, AE (3 dims)	64.2%±5.3%	53.7%±3.2%
Direct Sum (DS)(3 dims)	73.8%±7.6%	51.6%±3.0%
DS, DA (3 dims)	76.5%±4.1%	52.6%±3.1%
DS, BD (3 dims)	74.0%±2.9%	54.9%±3.8%
DS, DA, AE (3 dims)	72.7%±4.1%	54.4%±3.9%
DS, BD, AE (3 dims)	74.3%±2.1%	55.5%±1.9%
Everything (3 dims)	71.3%±6.9%	54.3%±2.8%

Table 8. xView Transfer Task. All experiments used a 3-dimensional latent space.

5. DISCUSSION

The results of Sec. 4 show a few consistent patterns:

1. Separate feature extractions supported better transfer across two classes in some cases, but multi-class sets favored a single feature extractor,
2. In all cases except for the synthetic-to-measured experiment, BD outperformed DAd regularization.

We discuss those patterns below.

5.1 Separate versus Single Feature Extraction

One of the key contributions of this work is to show that by treating the source and target data as existing in separate feature spaces, we can transfer more flexibly across object classes than in the baseline case. We demonstrated this in the experiment of Sec. 4.6, where we achieved 97.9% accuracy on both source and target recognition for a two-digit transfer task where the baseline achieved 73.6%. However, this was not always the case. In some cases, a single feature common extraction gave better performance, but it typically was coupled with either BD regularization or the autoencoder for preserving data structure. The DS shortcoming becomes increasingly evident for the multiple class transfer experiments where mode collapse caused chance performance for all architectures using the DS feature extraction. Taken together, the results show that learning separate feature embeddings can provide increased flexibility, but comes at the cost of being more challenging to train. It may require a different type of training procedure or some supervision. Such questions pose directions for future work.

5.2 BD versus DAd Regularization

Another important contribution is the application of BD within this network learning framework. Unlike the popular DAd regularization that acts as a proxy for the difference between source and target distributions, the BD explicitly penalizes the distribution differences. The experiments overwhelmingly favored BD, except for the synthetic-to-measured study in Sec. 4.9. The drawback of BD is the increased computational burden of KDE and the associated gradient calculation. Fortunately, the burden happens during training so it can be done offline. DAd regularization also has training drawbacks and is prone to vanishing or exploding gradients. In our preliminary experiments we found that batch normalization helped with the vanishing gradient problem. We also found that values larger than $\alpha_{\mathrm{DA}} = 0.1$ destabilized network training.

6. CONCLUSION

We have presented the DiSDAT network for transfer learning from a labeled source dataset to an unlabeled target dataset. The key idea is to learn separate feature embeddings to a common latent space by conceptualizing the source and target as existing in separate regions of a combined source and target direct sum space. By taking this view, we can more flexible transfer across different object classes and feature spaces. The important constraint is that we must match the target data to the source data in the latent space through appropriate regularization. Our experiments show that minimizing the distribution divergence while preserving the lower dimensional data manifold gives good results. The major drawback is mode collapse which occurs for multi-class sets and will be addressed with future work.

ACKNOWLEDGMENTS

We would like to thank Dr. Olga Mendoza-Schrock and Mr. Christopher Menart for their input and feedback during this research project. This work was supported by Air Force Research Laboratory (AFRL), Air Force Office of Scientific Research (AFOSR), Dynamic Data Driven Application Systems (DDDAS) Program, Autonomy Technology Research Center (ATRC), and Dr. Erik Blasch.

REFERENCES

[1] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," in [*International Conference on Learning Representations*], (2015).

[2] Pan, S. J. and Yang, Q., "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering* **22**, 1345–1359 (Oct 2010).

[3] Day, O. and Khoshgoftaar, T. M., "A survey on heterogeneous transfer learning," *Journal of Big Data* **4**, 29 (Sep 2017).

[4] Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T., "Adversarial discriminative domain adaptation," in [*IEEE Computer Society Conference on Computer Vision and Pattern Recognition*], 2962–2971 (2017).

[5] Csurka, G., [*A Comprehensive Survey on Domain Adaptation for Visual Applications*], 1–35, Springer International Publishing, Cham (2017).

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

[7] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H., "How transferable are features in deep neural networks?," in [*Advances in Neural Information Processing Systems 27*], Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., eds., 3320–3328, Curran Associates, Inc. (2014).

[8] Ganin, Y. and Lempitsky, V., "Unsupervised domain adaptation by backpropagation," in [*Proceedings of the 32nd International Conference on Machine Learning*], Bach, F. and Blei, D., eds., *Proceedings of Machine Learning Research* **37**, 1180–1189, PMLR, Lille, France (07–09 Jul 2015).

[9] Zhou, J. T., Pan, S. J., Tsang, I. W., and Yan, Y., "Hybrid heterogeneous transfer learning through deep learning," in [*Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*], *AAAI'14*, 2213–2219, AAAI Press (2014).

[10] Chen, M., Xu, Z., Weinberger, K. Q., and Sha, F., "Marginalized denoising autoencoders for domain adaptation," in [*Proceedings of the 29th International Coference on International Conference on Machine Learning*], *ICML'12*, 1627–1634, Omnipress, USA (2012).

[11] Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., and Smola, A. J., "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics* **22**, e49–e57 (July 2006).

[12] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A., "A kernel two-sample test," *Journal of Machine Learning Research* **13**, 723–773 (Mar. 2012).

[13] Long, M., Cao, Y., Wang, J., and Jordan, M. I., "Learning transferable features with deep adaptation networks," in [*Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*], *ICML'15*, 97–105, JMLR.org (2015).

[14] Long, M., Cao, Y., Cao, Z., Wang, J., and Jordan, M. I., "Transferable representation learning with deep adaptation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 1–1 (2018).

[15] Long, M., Zhu, H., Wang, J., and Jordan, M. I., "Deep transfer learning with joint adaptation networks," in [*Proceedings of the 34th International Conference on Machine Learning - Volume 70*], *ICML'17*, 2208–2217, JMLR.org (2017).

[16] Si, S., Tao, D., and Geng, B., "Bregman divergence-based regularization for transfer subspace learning," *IEEE Transactions on Knowledge and Data Engineering* **22**, 929–942 (July 2010).

[17] Mendoza-Schrock, O., Rizki, M. M., Raymer, M. L., and Velten, V. J., "Manifold transfer subspace learning for high dimensional data — applications to handwritten digits and health informatics," *in Proceedings of the International Conference on IP, Comp. Vision, and Pattern Recognition (IPCV'17)* , 3–9 (2017).

[18] Bregman, L., "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics* **7**(3), 200 – 217 (1967).

[19] Kullback, S. and Leibler, R. A., "On information and sufficiency," *Ann. Math. Statist.* **22**(1), 79–86 (1951).

[20] Zhuang, F., Cheng, X., Luo, P., Pan, S. J., and He, Q., "Supervised representation learning: Transfer learning with deep autoencoders," (2015).

[21] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial nets," in [*Advances in Neural Information Processing Systems 27*], Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., eds., 2672–2680, Curran Associates, Inc. (2014).

[22] Sankaranarayanan, S., Balaji, Y., Castillo, C. D., and Chellappa, R., "Generate To Adapt: Aligning Domains using Generative Adversarial Networks," in [*CVPR*], (2018).

[23] Liu, M.-Y. and Tuzel, O., "Coupled generative adversarial networks," in [*Advances in Neural Information Processing Systems 29*], Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., eds., 469–477, Curran Associates, Inc. (2016).

[24] Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T., "CyCADA: Cycle-consistent adversarial domain adaptation," in [*Proceedings of the 35th International Conference on Machine Learning*], Dy, J. and Krause, A., eds., *Proceedings of Machine Learning Research* **80**, 1989–1998, PMLR, Stockholmsmässan, Stockholm Sweden (10–15 Jul 2018).

[25] Gong, B., Shi, Y., Sha, F., and Grauman, K., "Geodesic flow kernel for unsupervised domain adaptation," in [*2012 IEEE Conference on Computer Vision and Pattern Recognition*], 2066–2073 (June 2012).

[26] Kipf, T. N. and Welling, M., "Semi-supervised classification with graph convolutional networks," in [*International Conference on Learning Representations (ICLR)*], (2017).

[27] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V., "Domain-adversarial training of neural networks," *Journal of Machine Learning Research* **17**(59), 1–35 (2016).

[28] Tikhonov, A. N. and Arsenin, V. I., [*Solutions of ill-posed problems / Andrey N. Tikhonov and Vasiliy Y. Arsenin ; translation editor, Fritz John*], Winston ; distributed solely by Halsted Press Washington : New York (1977).

[29] Xiao, H., Rasul, K., and Vollgraf, R., "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," (2017).

[30] LeCun, Y. and Cortes, C., "MNIST handwritten digit database," (2010).

[31] Hull, J. J., "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**, 550–554 (May 1994).

[32] Lam, D., Kuzma, R., McGee, K., Dooley, S., Laielli, M., Klaric, M., Bulatov, Y., and McCord, B., "xview: Objects in context in overhead imagery," *CoRR* **abs/1802.07856** (2018).

[33] Lewis, B., Scarnati, T., Sudkamp, E., Nehrbass, J., Rosencrantz, S., and Zelnio, E., "A SAR dataset for ATR development: the Synthetic and Measured Paired Labeled Experiment (SAMPLE)," in [*Algorithms for Synthetic Aperture Radar Imagery XXVI*], Zelnio, E. and Garber, F. D., eds., **10987**, 39 – 54, International Society for Optics and Photonics, SPIE (2019).

[34] Rivera, S., Mendoza-Schrock, O., and Diehl, A., "Transfer learning for aided target recognition: comparing deep learning to other machine learning approaches," in [*Automatic Target Recognition XXIX*], Hammoud, R. I. and Overman, T. L., eds., **10988**, 200 – 209, International Society for Optics and Photonics, SPIE (2019).