UNSUPERVISED MULTI-TARGET DOMAIN ADAPTATION: AN INFORMATION THEORETIC APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

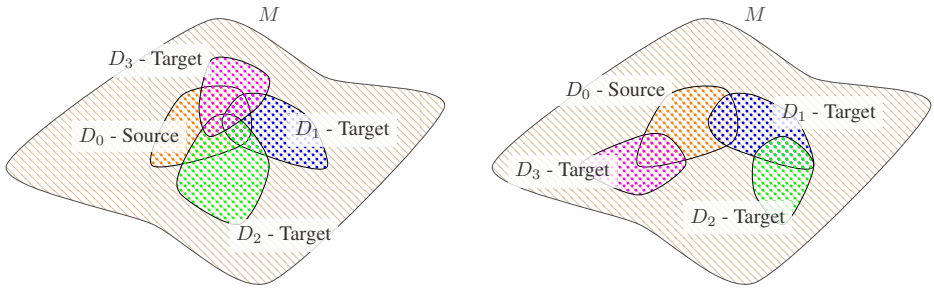
Unsupervised domain adaptation (uDA) models focus on pairwise adaptation settings where there is a single, labeled, source and a single target domain. However, in many real-world settings one seeks to adapt to multiple, but somewhat similar, target domains. Applying pairwise adaptation approaches to this setting may be suboptimal, as they fail to leverage shared information among multiple domains. In this work we propose an information theoretic approach for domain adaptation in the novel context of multiple target domains with unlabeled instances and one source domain with labeled instances. Our model aims to find a shared latent space common to all domains, while simultaneously accounting for the remaining private, domain-specific factors. Disentanglement of shared and private information is accomplished using a unified information-theoretic approach, which also serves to establish a stronger link between the latent representations and the observed data. The resulting model, accompanied by an efficient optimization algorithm, allows simultaneous adaptation from a single source to multiple target domains. We test our approach on three challenging publicly-available datasets, showing that it outperforms several popular domain adaptation methods.

1 INTRODUCTION

In real-world data, the training and test data often do not come from the same underlying distribution Sun et al. (2016). For instance, in the task of object recognition/classification from image data, this is may be due to the image noise, changes in the object view, etc., which induce different biases in the observed data sampled during the training and test stage. Consequently, assumptions made by traditional learning algorithms are often violated, resulting in degradation of the algorithms' performance during inference of test data. Domain Adaptation (DA) approaches (e.g., Fernando et al. (2013); Gong et al. (2012); Kodirov et al. (2015); Yoo et al. (2016)) aim to tackle this by transferring knowledge from a source domain (training data) to an unlabeled target domain (test data) to reduce the discrepancy between the source and target data distributions, typically by exploring domain-invariant data structures.

Existing DA methods can be divided into: (semi)supervised DA, and unsupervised DA Csurka (2017). The former assume that in addition to the labeled data of the source domain, some labeled data from the target domain are also available for training/adapting the classifiers. By contrast, the latter does not require any labels from the target domain but rather explores the similarity in the data distributions of the two domains. In this work, we focus on the unsupervised DA (uDA) scenario, which is more challenging due to the lack of correspondences in source and target labels.

Most works on uDA today focus on a single-source-single-target-domain scenario. However, in many real-world applications, unlabeled data may come from different domains, thus, with different statistical properties but with common task-related content. For instance, we may have access to images of the same class of objects (e.g., cars) recorded by various types of cameras, and/or under different camera views and



(a) Domains with common shared space. **(b)** Domains with pairwise shared spaces.

Figure 1: Illustration of domains with common (a) and pairwise-shared spaces (b). We tackle the domain adaptation task when all domains share a common task/space, which is then leveraged to transfer knowledge across multiple target domains.

at different times, rendering multiple different domains (e.g., datasets). Likewise, facial expressions of emotions, such as joy and surprise, shown by different people and recorded under different views, result in multiple domains with varying data distributions. In most cases, these domains have similar *underlying* data distributions, which can be leveraged to build more effective and robust classifiers for tasks such as the object or emotion recognition across multiple datasets/domains. To this end, traditional uDA methods focus on the single-source-single-target DA scenario. However, in the presence of multiple domains, as typically encountered in real-world settings, this pair-wise adaptation approach may be suboptimal as it fails to leverage simultaneously the knowledge shared across multiple task-related domains. Recently, Zhao et al. (2017) showed that by having access to multiple source domains can facilitate better adaptation to a single target domain, when compared to the pair-wise DA approach. While this is intuitive due to the access to multiple *labelled* source domains, offering more adaptation flexibility for the target domain (i.e., by efficiently exploring the data labels across multiple source domains that are most related to the target domain), it comes at the expense of the data labelling in multiple source domains, which can be costly and time-consuming. In either case, a single source domain or readily available multiple source domains, to the best of our knowledge, a simultaneous adaptation to *multiple* and *unlabelled* target domains remains an unexplored DA scenario. However, this DA scenario is important as we usually have access to multiple unlabeled domains; yet, the adaptation process is also more challenging due to the lack of supervision in the target domains. Still, multi-target DA can have advantages over a single-target DA when: (i) there is direct knowledge sharing between the source and multiple target domains (Fig. 1a), and (ii) the source and a target domain are related through another target domain (Fig. 1b). While this seems intuitive, it is critical how the data from multiple *unlabelled* target domains are leveraged within the multi-target DA approach, in order to improve its performance over the single target DA approaches and naive fusion of multiple target domains.

To this end, we propose a Multi-Target DA-Information-Theoretic-Approach (MTDA-ITA) for single-source-multi-target DA. We exploit a single source domain and focus on multiple target domains to investigate the effects of multi-target DA; however, the proposed can easily be extended to multiple source domains. This approach leverages the data from multiple target domains to improve performance compared to individually learning from pairwise source-target domains. Specifically, we simultaneously factorize the information from each available target domain and learn separate subspaces for modeling the shared (i.e., correlated across the domains) and private (i.e., independent between the domains) subspaces of the data Salzmann et al. (2010). To this end, we employ deep learning to derive an information theoretic approach where we jointly maximize the mutual information between the domain labels and private (domain-specific) features, while minimizing the mutual information between the the domain labels and the shared (domain-invariant) features. Consequently, the more robust feature representations are learned for each target domain by exploiting dependencies between

multiple target domains. We show on benchmark datasets for DA that this approach leads to overall improved performance on each target domain, compared to independent DA for each pair of source-target domains, or the naive combination of multiple target domains, and state-of-the-art models applicable to the target task.

2 THE PROPOSED METHOD

Without loss of generalizability, we consider a multi-class (K-class) classification problem as the running example. Furthermore, let $(\mathbf{X}, \mathbf{Y}, \mathbf{D}) = \{(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{d}_i)\}_{i=0}^{N}$ be a collection of M domains (a labeled source domain, and $M-1$ unlabeled target domains), where $\boldsymbol{x}_i$ denotes the i-th sample, and $\boldsymbol{y}_i = [y_i^1, y_i^2, ..., y_i^K]$ and $\boldsymbol{d}_i = [d_i^1, d_i^2, ..., d_i^M]$ are the K-D and M-D encoding of the class and domain labels for $\boldsymbol{x}_i$, respectively. Note that the class labels are only available for the source samples.

The latent space representation of the data point $\boldsymbol{x}$ is denoted as $\mathbf{z} = [\mathbf{z}_s, \mathbf{z}_p]$, where $\mathbf{z}_s$ and $\mathbf{z}_p$ are the (latent) shared and private features of the data point $\mathbf{x}$, respectively. Let $\mathbf{z}_s$ and $\mathbf{z}_p$ be some stochastic function of ($\boldsymbol{x}$, $\mathbf{d}$,$\boldsymbol{y}$) parameterized by (θ_s, θ_p, θ_c), respectively. We propose to maximize the following objective function:

$$\mathcal{L}(\theta_s, \theta_p, \theta_c; \boldsymbol{x}, \boldsymbol{y}, \mathbf{d}) = \lambda_r I(\boldsymbol{x}; \mathbf{z}) + \lambda_c I(\mathbf{y}; \mathbf{z}_s) + \lambda_d \big(I(\mathbf{d}; \mathbf{z}_p) - I(\mathbf{d}; \mathbf{z}_s) \big). \tag{1}$$

where $I(x; y)$ denotes the Mutual Information between the random variables x and y. λ_r, λ_c and λ_d denote the hyper-parameters controlling the weights of the objective terms. The proposed objective function (1) maximizes the three terms described below:

- $I(\boldsymbol{x}; \mathbf{z})$: encourages the latent features (both shared and private) to preserve information about the data samples (that can be used to reconstruct $\boldsymbol{x}$ from $\boldsymbol{z}$).
- $I(\mathbf{y}; \mathbf{z}_s)$: enables to correctly predict the true class label of the samples out of their common shared features.
- $I(\mathbf{d}; \mathbf{z}_p) - I(\mathbf{d}; \mathbf{z}_s)$: encourages the latent private features to preserve the information about the domain label and penalizes the latent shared features to be domain informative. This also not only reduces the redundancy in the shared and private features, but also, penalizes the redundancy of different private spaces, while preserving the shared information.

An additional term could be used to minimize the mutual information between the shared ($\mathbf{z}_s$) and private ($\mathbf{z}_p$) features. However, computing the mutual information (even approximating it) is intractable due to the highly complex joint distribution $p(\mathbf{z}_s, \mathbf{z}_p)$. Since we want $\mathbf{z}_s$ and $\mathbf{z}_p$ features to encode different aspects of $\boldsymbol{x}$, we enforce such constraint by jointly maximizing the term: $I(\mathbf{d}; \mathbf{z}_p) - I(\mathbf{d}; \mathbf{z}_s)$.

2.1 OPTIMIZATION

The following lower bound for mutual information is derived using the non-negativity of KL-divergence Barber & Agakov (2003); i.e., $\Sigma_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{z}) \ln \frac{p(\boldsymbol{x}|\boldsymbol{z})}{q(\boldsymbol{x}|\boldsymbol{z})} \geq 0$ gives:

$$I(\boldsymbol{x}; \boldsymbol{z}) \geq H(\boldsymbol{x}) + \mathbb{E}_{p(\boldsymbol{x}, \boldsymbol{z})}[\ln q(\boldsymbol{x}|\boldsymbol{z}; \phi)] \tag{2}$$

where $H(\mathbf{x})$ denotes the Shanon Entropy of the random variable $\mathbf{x}$. $q(\boldsymbol{x}|\boldsymbol{z}; \phi)$ is any arbitrary distribution parametrized by ϕ. We need a variational distribution $q(\boldsymbol{x}|\boldsymbol{z}; \phi)$ because the posterior distribution $p(\boldsymbol{x}|\boldsymbol{z}) = p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x})/p(\boldsymbol{z})$ is intractable since the true data distribution $p(\boldsymbol{x})$ is assumed to be unknown. Similarly, we can derive lower bounds for $I(\mathbf{d}; \mathbf{z}_p) \geq H(\mathbf{d}) + \mathbb{E}_{p(\mathbf{d}, \mathbf{z}_p)}[\ln q(\mathbf{d}|\mathbf{z}_p; \psi)]$ and $I(\mathbf{d}; \mathbf{z}_s) \geq H(\mathbf{d}) + \mathbb{E}_{p(\mathbf{d}, \mathbf{z}_s)}[\ln q(\mathbf{d}|\mathbf{z}_s; \psi)]$, where $q(\mathbf{d}|\mathbf{z}_p; \psi)$ is any arbitrary distribution parametrized by ψ.[1] We further compute $I(\mathbf{y}; \mathbf{z}_s)$ as $I(\mathbf{y}; \mathbf{z}_s) = H(\mathbf{y}) + \mathbb{E}_{p(\mathbf{y}, \mathbf{z}_s)}[\ln p(\mathbf{y}|\mathbf{z}_s)]$.

[1]Note that, for simplicity, we shared the parameters ψ between the approximate posterior distributions $q(\mathbf{d}|\mathbf{z}_s, \psi)$ and $q(\mathbf{d}|\mathbf{z}_p; \psi)$.

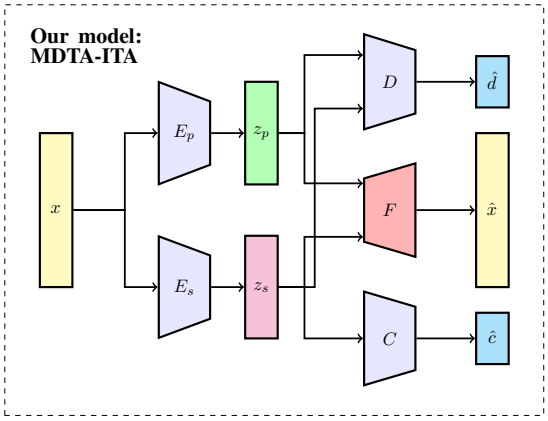


Figure 2: MDTA-ITA: The encoder $E_s(\mathbf{x})$ captures the feature representations (z_s) for a given input sample $\mathbf{x}$ that are shared among domains. $E_p(\mathbf{x})$ captures domain–specific private features (z_p) using the *shared* private encoder. The shared decoder $F(z_p, z_s)$ learns to reconstruct the input sample by using both the private and shared features. The domain classifier D learns to correctly predict the domain labels of the actual samples from both their shared and private features while the classifier C learns to correctly predict the class labels from the shared features.

Let next $E_s(\mathbf{x}; \theta_s)$ be a function parameterized by θ_s that maps a sample $\mathbf{x}$ to the *shared* features $\mathbf{z}_s$, and $E_p(\mathbf{x}; \theta_p)$ be an analogous function which maps $\mathbf{x}$ to $\mathbf{z}_p$, the features that are *private* to each domain (Fig. 2). We also define $F(\mathbf{z}_s, \mathbf{z}_p; \phi)$ as a decoding function mapping the concatenation of the latent features $\mathbf{z}_s$ and $\mathbf{z}_p$ to a sample reconstruction $\hat{\mathbf{x}}$, and $D(\mathbf{z}; \psi)$ as a decoding function mapping $\mathbf{z}_s$ and $\mathbf{z}_p$ to a M-dimensional probability vector: the predictions of the domain label $\hat{\mathbf{d}}$. Finally, $C(\mathbf{z}_s; \boldsymbol{\theta}_c)$ is a task-specific function mapping $\mathbf{z}_s$ to a K-dimensional probability vector of the class label $\hat{\mathbf{y}}$.

By approximating $p(\mathbf{d}), p(\boldsymbol{x}), p(\boldsymbol{y}), p(\mathbf{z}_s|\mathbf{x})$ and $p(\mathbf{z}_p|\mathbf{x})$ with their empirical distributions, and modeling the variational distributions as $\ln q(\mathbf{x}|\mathbf{z}; \psi) = \|\mathbf{x} - F(\mathbf{z}; \phi)\|_1$, $\ln q(\mathbf{d}|\mathbf{z}) = \mathbf{d}^\top \ln D(\mathbf{z}; \psi)$, and $\ln p(\mathbf{y}|\mathbf{z}_s) = \mathbf{y}^\top \ln C(\mathbf{z}_s; \theta_c)$, where $\|.\|_1$ denotes the L_1 norm, the optimization task can be posed as a minimax saddle point problem, where we use adversarial training to maximize (1) w.r.t. the stochastic parameters $(\theta_s, \theta_p, \theta_c)$, and to minimize (1) w.r.t. the variational parameters (ϕ,ψ), using Stochastic Gradient Descent (SGD).

Optimizing the parameters ϕ of the decoder F:

$$\hat{\phi} = \arg\min_{\phi} \mathcal{L}_F = \frac{\lambda_r}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - F\big(E_s(\mathbf{x}_i), E_p(\mathbf{x}_i)\big)\|_1. \tag{3}$$

The decoder $F(\mathbf{z}_s, \mathbf{z}_p; \phi)$ is trained in such a way so as to minimize the difference between original input $\mathbf{x}$ and its decoding from corresponding shared and private features via the decoder $F(\cdot)$.

Optimizing the parameters ψ of the domain classifier D:

$$\hat{\psi} = \arg\min_{\psi} \mathcal{L}_D = -\frac{\lambda_d}{N} \sum_{i=1}^{N} \mathbf{d}_i^\top \ln D\big(E_s(\mathbf{x}_i)\big) - \frac{\lambda_d}{N} \sum_{i=1}^{N} \mathbf{d}_i^\top \ln D\big(E_p(\mathbf{x}_i)\big). \tag{4}$$

$D(\mathbf{z}; \psi)$ can be considered as a classifier whose task is to distinguish between the shared/private features of the different domains. More precisely, the two terms in Eq. 4 encourage D to correctly predict the domain labels from the shared and private features, respectively.

Optimizing the parameters θ_c of the label classifier C:

$$\hat{\theta}_c = \arg\min_{\theta_c} \big\{ -H(\mathbf{y}) - \mathbb{E}_{p(\mathbf{y}, \mathbf{z}_s)} \big[\ln p(\mathbf{y}|\mathbf{z}_s) \big] \big\}. \tag{5}$$

Since we have access to the source labels, $H(\mathbf{y})$ is a constant for source samples. we can approximate $H[\boldsymbol{y}]$ for the target samples using the output of the classifier C, leading to the following optimization problem:

$$\hat{\theta_c} = \arg\min_{\theta_c} \mathcal{L}_C = -\frac{1}{N}\sum_{i=1}^{N_s} \mathbf{y}_i^T \ln C\big(E_s(\mathbf{x}_i)\big) - \frac{\lambda_c}{N-N_s}\sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)^\top \ln C\big(E_s(\mathbf{x}_i)\big) + \frac{\lambda_c}{N-N_s}\sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)^\top \ln\left(\frac{1}{N-N_s}\sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)\right). \tag{6}$$

where N_s denotes the number of source samples. Intuitively, we enforce the classifier $C(\mathbf{z}_s;\theta_c)$ to correctly predict the class labels of the source samples by the first term in Eq. 6. We use the second term to minimize the entropy of $p(\mathbf{y}|\boldsymbol{z}_s)$ for the target samples; effectively, reducing the effects of "confusing" labels of target samples, as given by $p(\mathbf{y}|\boldsymbol{z}_s)$ that leads to decision boundaries occur far away from target data-dense regions in the feature space. The intuition behind the last term is that if we minimize the entropy only(second term), we may arrive at a degenerate solution where every point x_t is assigned to the same class. Hence, the last term encourages the classifier $C(\cdot)$ to have balanced labeling for the target samples where it reaches its minimum, $\ln K$, when each class is selected with uniform probability.

Optimizing the parameter θ_p of the private encoder E_p:

$$\hat{\theta_p} = \arg\min_{\theta_p} \mathcal{L}_P = \frac{\lambda_r}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - F\big(E_s(\mathbf{x}_i), E_p(\mathbf{x}_i)\big)\|_1 - \frac{\lambda_d}{N}\sum_{i=1}^{N} \mathbf{d}_i^\top D\big(E_p(\mathbf{x}_i)\big) \tag{7}$$

The first term in Eq. 7 encourages the private encoder $E_p(\mathbf{x};\theta_p)$ to preserve the recovery ability of the private features. The second term, $E_p(\cdot)$, enforces distinct private features be produced for each domain by penalizing the representation redundancy in different private spaces. This, in turn, encourages moving this common information from multiple domains to their shared space.

Optimizing the parameter θ_s of the shared encoder E_s:

$$\hat{\theta_s} = \arg\min_{\theta_s} \mathcal{L}_S = \frac{\lambda_r}{N}\sum_{i=1}^{N} \|\mathbf{x} - F\big(E_s(\mathbf{x}_i), E_p(\mathbf{x}_i)\big)\|_1 - \frac{\lambda_c}{N}\sum_{i=1}^{N_s} \mathbf{y}_i^T \ln C\big(E_s(\mathbf{x}_i)\big) - \frac{\lambda_d}{N}\sum_{i=1}^{N} \mathbf{d}_i^\top \ln D\big(E_s(\mathbf{x}_i)\big) - \frac{\lambda_c}{N-N_s}\sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)^\top \ln C\big(E_s(\mathbf{x}_i)\big) + \frac{\lambda_c}{N-N_s}\sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)^\top \ln\left(\frac{1}{N-N_s}\sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)\right) \tag{8}$$

The first term in Eq. 8 encourages the shared encoder $E_s(\mathbf{x};\theta_s)$ to preserve the recovery ability of the shared features. The second term is the source domain classification loss penalty that encourages E_s to produce discriminative features for the labeled source samples. The third term simulates the adversarial training by trying to fool the domain classifier $D(\cdot)$ when predicting the domain labels $\mathbf{d}$, given the shared features $\mathbf{z}_s$. This effect of this is two-fold: (i) the rendered shared features are more distinct from the corresponding private features, (ii) the shared features of different domains are encouraged to be similar to each other. The last two terms encourage $E_s(\cdot)$ to produce the shared features for target samples so that the classifier is confident on the unlabeled target data, driving the shared features away from the decision boundaries. To train our model, we alternate between updating the shared encoder $E_c(\cdot)$, the private encoder $E_p(\cdot)$, the decoder $F(\cdot)$, the classifier $C(\cdot)$, and the domain classifier $D(\cdot)$ using the SGD algorithm (see Alg.1 in Appendix).

3 RELATED WORK

There has been extensive prior work on domain adaptation Csurka (2017). Recent papers have focused on transferring deep neural network representations from a labeled source dataset to an unlabeled target domain, where the main strategy is to find a feature space such that the confusion between source and target distributions in that space is maximized (Rebuffi et al. (2017); Benaim & Wolf (2017); Courty et al. (2017); Motiian et al. (2017); Saito et al. (2017); Zhang et al. (2017); Yan et al. (2017); Bousmalis et al. (2017a)). For this, it is critical to first define a measure of divergence between source and target distributions. For instance, several methods have used the Maximum Mean Discrepancy (MMD) loss for this purpose (e.g., Bousmalis et al. (2017b); Zellinger et al. (2017); Long et al. (2014)). MMD computes the norm of the difference between two domain means in the reproducing Kernel Hilbert Space (RKHS) induced by a pre-specified kernel. The Deep Adaptation Network (DAN) Long et al. (2015) applied MMD to layers embedded in a RKHS, effectively matching higher order statistics of the two distributions. The deep Correlation Alignment (CORAL) method Sun & Saenko (2016) attempts to match the mean and covariance of the two distributions. Deep Transfer Network (DTN) Zhang et al. (2015) achieved source/target distribution alignment via two types of network layers based on MMD distance: the shared feature extraction layer, which learns a subspace that matches the marginal distributions of the source and the target samples, and the discrimination layer, which matches the conditional distributions by classifier transduction.

Recently proposed unsupervised DA methods (Rebuffi et al. (2017); Benaim & Wolf (2017); Courty et al. (2017); Motiian et al. (2017); Saito et al. (2017); Zhang et al. (2017)) operate by training deep neural networks using adversarial training, which allows the learning of feature representations that are simultaneously discriminative of source labels, and indistinguishable between the source and target domain. For instance, Ganin & Lempitsky (2015) proposed a DA mechanism called Domain-Adversarial Training of Neural Networks (DANN), which enables the network to learn domain invariant representations in an adversarial way by adding a domain classifier and back-propagating inverse gradients. Adversarial Discriminative Domain Adaptation (ADDA) Tzeng et al. (2017) learns a discriminative feature subspace using the source labels, followed by a separate encoding of the target data to this subspace using an asymmetric mapping learned through a domain-adversarial loss. Liu et al. (2017) makes a shared-latent space assumption and proposes an unsupervised image-to-image translation (UNIT) framework based on Coupled GANs Liu & Tuzel (2016). Another example is the pixel-level domain adaptation models that perform the distribution alignment not in the feature space but directly in raw pixel space. PixelDA Bousmalis et al. (2017a) uses adversarial approaches to adapt source-domain images as if drawn from the target domain while maintaining the original content.

While these approaches have shown success in DA tasks with single source-target domains, they are not designed to leverage information from multiple domains simultaneously. More recently, Zhao et al. (2017) introduced an adversarial framework called MDAN for multiple source single target domain adaptation where a domain classifier, induced by minimizing the H-divergence between multiple source and a target domain, is used to align their feature distributions in a shared space. Instead, in our approach we focus on multi-target DA where we perform adaptation of multiple *unlabelled* target domains. From the probabilistic point of view, our model is VAE-inspired models Bowman et al. (2015). However, we propose to tackle the task using an information-theoretic (IT) approach instead of the traditional ELBO optimization. One of the main drawbacks of ELBO-based approaches is that they can result in poor latent representation, when a powerful decoder effectively ignores the latent space Bowman et al. (2015). In the spirit of recent works on improved representational learning Alemi et al. (2018), we seek to directly focus on recovering a good latent representation by replacing the ELBO objective with an IT-driven loss. In contrast to the unsupervised representation learning approaches, our setting allows us to further improve the latent representation using the labeled data in the source domain while leveraging the sharing of dependencies across different target domains.

3.1 CONNECTION TO DOMAIN SEPARATION NETWORKS

The method closest to our work is Domain Separation Networks (DSN) Bousmalis et al. (2016), which use the notion of auto-encoders to explicitly separate the feature representations private to each source/target domain from those that are shared between the domains. However, DSN were not experimentally tested in the multi-domain setting. Although such an extension might seem trivial, DSNs require an autoencoder per domain, making the model impractical in the case of more than a couple of domains.

The overall loss of DSN consists of a reconstruction loss for each domain modeled by a shared decoder, a similarity loss such as MMD, which encourages domain invariance modeled by a shared encoder, and a dissimilarity loss modeled by two private encoders: one for the source domain and one for the target domain. While one could attempt to generalize DSN to multiple target domains by having individual per-target domain private encoders, doing so would prove problematic when the number of target domains is large — each private encoder would require a large "private" dataset to learn the private parameters. Precisely, for multiple (M) target domains, we could train a DSN model with one shared encoder, $M + 1$ private encoder (one for each domain), and one shared decoder. This leads to $M + 3$ models to train that implies the number of models increases linearly with the number of domains, as does the required training time. Second, DSN uses an orthogonality constraint among the private representations which may not be strong enough to remove redundancy and enforce disentangling among different private spaces. More precisely, DSN defines the loss via a soft subspace orthogonality constraint between the private and shared representation of each domain. However, it does not enforce the private representation of different domains to be different that may result in redundancy of different private spaces.

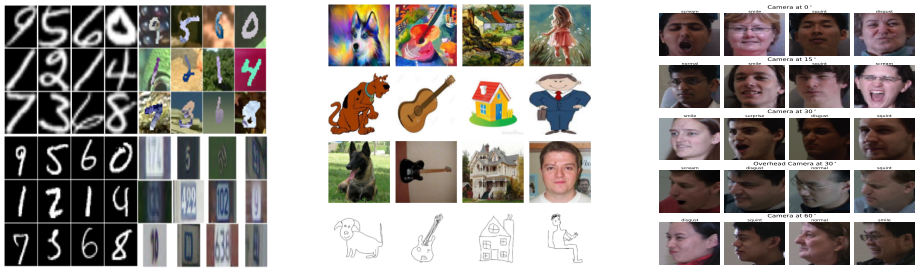
In addition, DSN enforces separation of spaces using the notion of Euclidean orthogonality, e.g., $\|z_s - z_p\|^2$. In case of multiple target domains, this would result in learning of all pairs of private spaces independently. To address those deficiencies, we first explicitly couple different private encoders into a single private encoder model, E_{θ_p} of Fig. 2 , which allows us to generalize to an arbitrary number of target domains. To assure that the information among the private and shared spaces is not shared (i.e., "orthogonal"), we define an information-theoretic criteria enforced by a domain classifier,D_Ψ of Fig. 2, which aims to segment the private space into clusters that correspond to individual target domains. By using D within the adversarial framework, **MTDA-ITA** learns simultaneously the shared and private features from different domains. (see Fig. 4 in Appendix) We also show in Sec. 4 that our model performs better than the trivial extension of DSNs to the multi-domain case.

4 EXPERIMENTAL RESULTS

We compare the proposed method with state-of-the-art methods on standard benchmark datasets: a digit classification task that includes 4 datasets: MNIST LeCun et al. (1998), MNIST-M Ganin et al. (2016), SVHN Netzer et al. (2011), USPS Tzeng et al. (2017), Multi-PIE expression recognition dataset[2], and PACS multi-domain image recognition benchmark Li et al. (2017), a new dataset designed for the cross-domain recognition problems (The details for this experiment is available in the Appendix). Figure 3 illustrates image samples from different datasets and domains. We evaluate the performance of all methods with classification accuracy metric.

We used ADAM Kingma & Ba (2015) for training; the learning rate was set to 0.0002 and momentums to 0.5 and 0.999. We used batches of size 16 from each domain, and the input images were mean-centered/rescaled to $[-1, 1]$. The hyper-parameters are empirically set as $\lambda_r = 1.0, \lambda_c = 0.01, \lambda_d = 0.20$. For the network architecture, our private/shared encoders consisted of 3 convolutional layers as the front-end and 4 basic residual blocks as the back-end. The decoder consisted of 4 basic residual blocks as the front-end and 3

[2]http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html



(a) Digit datasets **(b)** PACS dataset **(c)** Multi-PIE dataset

Figure 3: Exemplary images from different datasets. a) Digits datasets, b) PACS datatset (first row: Art-painting, second row: Cartoon, Third row: Photo, last row: Sketch), c) Multi-PIE dataset (each row corresponds to a different camera angle and each subject depicts an expression("normal", "smile", "surprise", "squint", "disgust", "scream") at every camera position).

transposed convolutional layers as the back-end. The discriminator and the classifier consisted of stacks of convolutional layers. We used ReLU for nonlinearity. TanH function is used as the activation function of the last layer in the decoder F for scaling the output pixels to $[-1, 1]$. The details of the networks are given in Appendix. The quantitative evaluation involves a comparison of the performance of our model to previous work and to "Source Only" and "1-NN" baselines that do not use any domain adaptation. For "Source Only" baseline, we train models only on the unaltered source training data and evaluate on the target test data. We compare the proposed method **MTDA-ITA** with several related methods designed for pair-wise source-target adaptation: CORAL Sun & Saenko (2016), DANN Ganin & Lempitsky (2015), ADDA Tzeng et al. (2017), DTN Zhang et al. (2015), UNIT Liu et al. (2017), PixelDA Bousmalis et al. (2017b), and DSN Bousmalis et al. (2016). We reported the results of two following baselines: (i) one is to combine all the target domains into a single one and train it using **MTDA-ITA**, which we denote as (**c-MTDA-ITA**). (ii) the other one is to train multiple **MTDA-ITA** separately, where each one corresponds to a source-target pair which we denote as (**s-MTDA-ITA**). For completeness, we reported the results of the competing methods by combining all the target domains into a single one (denoted by c-DTN, c-ADDA, and c-DSN) as well. We also extend DSN to multiple domains by adding multiple private encoder to it (denoted by mp-DSN) and contrast it with our model. The methods are also compared with the results obtained by the Nearest Neighbor (1-NN) classifier on the source domain data without adaptation.

4.1 Digits Datasets

We combine four popular digits datasets (MNIST, MNIST-M, SVHN, and USPS) to build the multi-target domain dataset. All images were uniformly rescaled to 32×32. We take each of MNIST-M, SVHN, USPS, and MNIST as source domain in turn, and the rest as targets. We use all labeled source images and all unlabeled target images, following the standard evaluation protocol for unsupervised domain adaptation Ganin et al. (2016); Long et al. (2016). We show the accuracy of different methods in Tab. 1. The results show that first of all **cMTDA-ITA** has worse performance than **sMTDA-ITA** and **MTDA-ITA**. We have similar observations for ADDA, DTN, and DSN that demonstrates a naive combination of different target datasets can sometimes even decrease the performance of the competing methods. Furthermore, **MTDA-ITA** outperforms the state-of-the-art methods in most of domain transformations. The higher performance of **MTDA-ITA** compared to other methods is mainly attributed to the joint adaptation of related domains where each domain could benefit of other related domains. Furthermore, from the results obtained, we see that it is beneficial to use information coming from unlabeled target data (see Eq. 6 for updating the classifier C_{θ_c}) during the learning process, compared to when no data from target domain is used (See the ablation study section

Table 1: Mean classification accuracy on digit classification. M: MNIST; MM: MNIST-M, S: SVHN, U: USPS. The best is shown in red. c-X: combining all target domains into a single one and train it using X. s-MTDA-ITA: training multiple MTDA-ITA where each one correspond to a source-target pair. mp-DSN: extended DSN with multiple private encoder. *UNIT trains with the extended SVHN (> 500K images vs ours 72K). *PixelDA uses ($\approx 1,000$) of labeled target domain data as a validation set for tuning the hyperparameters.

method	S → M	S → MM	S → U	M → S	M → MM	M → U	MM → S	MM → M	MM → U	U → S	U → M	U → MM
Source Only	62.10	40.43	39.90	30.29	55.98	78.30	40.00	84.46	80.43	23.41	50.64	41.45
1-NN	35.86	18.21	29.31	28.01	12.58	41.22	21.45	82.13	36.90	15.34	38.45	18.54
CORAL Sun & Saenko (2016)	63.10	54.37	50.15	33.40	57.70	81.05	40.20	84.90	87.54	38.90	85.01	60.45
DANNGanin et al. (2016)	73.80	61.05	62.54	35.50	77.40	81.60	51.80	61.05	85.34	35.50	77.40	61.60
ADDATzeng et al. (2017)	77.68	64.23	64.10	30.04	91.47	90.51	40.64	92.82	80.70	41.23	90.10	56.21
c-ADDA	80.10	56.80	64.80	27.50	83.30	84.10	35.43	88.47	74.19	39.36	84.67	52.54
DTNZhang et al. (2015)	81.40	63.70	60.12	40.40	85.70	85.80	48.80	88.80	90.68	42.43	89.04	55.78
c-DTN	82.10	59.30	56.87	38.32	80.90	79.31	44.21	83.60	84.98	39.75	85.04	48.86
PixelDABousmalis et al. (2017b)	–	–	–	–	**98.10***	**94.10***	–	–	–	–	–	–
UNITLiu et al. (2017)	**90.6***	–	–	–	–	92.90	–	–	–		90.60	–
DSNBousmalis et al. (2016)	82.70	64.80	65.30	49.30	83.20	91.65	51.50	90.20	89.95	**48.20**	**91.40**	60.45
c-DSN	83.10	60.56	60.35	46.80	80.49	88.21	47.10	84.60	84.80	40.50	86.05	56.25
mp-DSN	83.40	61.00	58.10	47.35	79.30	78.45	47.15	85.51	83.24	38.30	87.40	55.47
s-MTDA-ITA	82.90	63.10	63.54	49.60	82.42	89.21	50.55	94.82	89.05	40.13	87.10	61.01
c-MTDA-ITA	79.20	59.90	63.70	45.30	77.12	87.47	47.32	90.20	90.01	41.10	85.35	60.31
MTDA-ITA	**84.60**	**65.30**	**70.03**	**52.01**	85.50	**94.20**	**53.50**	**98.20**	**94.10**	46.00	**91.50**	**67.30**

Table 2: Mean classification accuracy on Multi-PIE classification. The best is shown in red.

method	C05 → C08	C05 → C09	C05 → C13	C05 → C14	C13 → C05	C13 → C08	C13 → C09	C13 → C14	C14 → C05	C14 → C08	C14 → C09	C14 → C13
Source Only	31.56	40.67	39.89	54.70	50.79	45.90	40.04	59.68	60.03	36.80	40.11	60.57
1-NN	27.28	31.22	33.66	47.04	33.21	37.01	34.45	48.79	47.44	28.24	30.86	44.86
CORAL Sun & Saenko (2016)	36.55	38.60	40.60	55.29	54.89	48.90	40.30	68.90	59.98	40.63	40.80	65.11
DANNGanin et al. (2016)	40.30	41.20	40.12	58.90	57.86	50.30	45.30	70.68	57.20	40.22	40.77	70.50
ADDATzeng et al. (2017)	33.21	30.8 6	52.44	70.18	64.83	63.20	55.48	74.25	73.62	43.56	38.68	72.84
c-ADDA	46.88	36.38	39.14	65.41	59.20	30.70	53.20	68.33	65.88	30.60	45.34	64.30
DTNZhang et al. (2015)	38.50	30.56	55.78	68.90	63.78	60.45	60.55	72.60	70.67	41.55	41.45	70.67
c-DTN	41.70	31.10	50.19	60.34	57.53	55.24	57.14	65.16	63.80	38.97	39.80	62.10
PixelDABousmalis et al. (2017b)	44.93	44.75	45.18	46.88	45.68	44.95	44.45	**90.50**	46.28	45.89	44.45	69.15
UNITLiu et al. (2017)	44.47	44.47	44.47	44.51	44.14	44.47	44.21	44.47	43.03	44.44	44.47	44.47
DSNBousmalis et al. (2016)	45.12	44.35	48.12	75.00	64.15	57.70	49.15	80.75	82.20	38.75	45.00	80.50
c-DSN	42.52	38.54	34.15	69.45	57.34	31.63	51.17	74.52	82.01	34.25	42.63	79.42
mp-DSN Tzeng et al. (2017)	41.30	35.14	34.40	65.70	55.20	30.40	47.80	75.30	80.75	30.20	43.00	79.02
s-MTDA-ITA	44.40	44.60	47.65	80.20	70.10	58.90	58.10	80.12	82.05	45.90	52.67	81.60
c-MTDA-ITA	40.49	40.70	42.80	71.60	60.34	55.67	57.10	73.50	76.80	43.10	48.10	80.90
MTDA-ITA	**49.01**	**48.23**	**53.13**	**84.29**	**78.40**	**66.70**	**70.30**	85.49	**87.20**	**61.40**	**60.05**	**86.70**

for more information). Indeed, using our scheme, we find a representation space in which embeds the knowledge from the target domain into the learned classifier. By contrast, the competing methods do not provide a principled way of sharing information across all domains, leading to overall lower performance. The results also verify the superiority of **MTDA-ITA** over mp-DSN. This can be due to (i) having multiple private encoders increase the number of parameters that may lead to model overfitting, (ii) superiority of the **MTDA-ITA**'s domain adversarial loss over the DSN's MMD loss to separate the shared and private features, (iii) utilization of the unlabeled target data to regularize the classifier in **MTDA-ITA**.

Feature Visualization: we use t-SNE Maaten & Hinton (2008) to visualize shared and private feature representations from different domains. Fig. 4 (Appendix) shows shared and private features from source (SVHN) and target domains before (a) and after adaptation (b). **MTDA-ITA** significantly reduces the domain mismatch for the shared features (circle markers) and increases it for the private features (squares). This is partially due to the use of the domain classifier D, which penalizes the domain mismatch for the shared features and rewards the mismatch for the private features. Moreover, as supported by the quantitative results in Tab. 1, joint adaptation of related domains and the classifier, accomplished through the model, leads to superior class separability, compared to original features. This is depicted in Fig. 4(d) (Appendix), where the points in the shared space (large cluster) are grouped into class-specific subgroups (color indicates class label), while they are mixed in private spaces (smaller clusters). This is in contrast to Fig. 4(c) (Appendix), where original features show no class-specificity.

4.2 MULTI-PIE DATASET

The Multi-PIE dataset includes face images of 337 individuals captured from different expressions, views, and illumination conditions (Fig. 3(c)). For this experiment, we use 5 different camera views (positions) $C05$, $C08$, $C09$, $C13$, and $C14$ as different domains (Fig. 3(c)) and the face expressions (normal, smile, surprise, squint, disgust, scream) as labels. Each domain contains 27120 images of size $64 \times 64 \times 3$. We used each view as the source domain, in turn, and the rest as targets. We expect the face inclination angle to reflect the complexity of transfer learning. Table 2 shows the classification accuracy for $C05, C13$ and $C14$ as source domain (the results for views $C08$ and $C09$ as source domain are available in the Appendix). As can be seen, MTDA-ITA achieves the best performances as well as the best scores in most settings that verifies the effectiveness of MTDA-ITA for multi-target domain adaptation. Clearly, with the increasing camera angle, the image structure changes up to a certain extent (the views become heterogeneous). However, our method produces better results even under such very challenging conditions.

4.3 ABLATION STUDIES

We performed an ablation study on the proposed model measuring impact of various terms on the model's performance. To this end, we conducted additional experiments for the digit datasets with different components ablation, i.e., training without the reconstruction loss (denoted as MTDA-woR) by setting $\lambda_r = 0$, training without the classifier entropy loss (denoted as MTDA-woE) by setting $\lambda_c = 0$, training without the domain separation loss (denoted as MTDA-woD) by setting $\lambda_d = 0$. As can be seen from Fig. 5 in Appendix, by disabling each of the above components, the performance degrades. More precisely, the average drop by disabling the classifier entropy loss is $\approx 3.5\%$. Similarly, by disabling the reconstruction loss and the domain discriminator loss, we have $\approx 4.5\%$ and $\approx 22\%$ average drop in performance respectively. Clearly, by disabling the multi-domain adversarial loss, the accuracy significantly drops due to the severe data distribution mismatch between different domains. The figure also demonstrates that leveraging the unlabeled data from multiple target domains during training enhances the generalization ability of the model that leads to higher performance. In addition, the performance drop caused by removing the reconstruction loss , i.e., without the private encoder/decoder, indicates (i) the benefit of modeling the latent features as the combination of shared and private features, (ii) the ability of the model's domain adversarial loss to effectively learn those features. In summary, this ablation study showed that the individual components bring complimentary information to achieve the best classification results.

5 CONCLUSION

This paper presented an information theoretic end-to-end approach to unsupervised domain adaptation in the context of a single source and multiple target domains that share a common task or properties. The proposed method learns feature representations invariant under multiple domain shifts and simultaneously discriminative for the learning task. This is accomplished by explicitly separating representations private to each domain and shared between source and target domains using a novel discrimination strategy that avoids explicit encoding of orthogonality. Our use of a single private domain encoder results in a highly scalable model, easily optimized using established back-propagation approaches. Results on two benchmark datasets for image classification show superiority of the proposed method compared to the state-of-the-art methods for unsupervised domain adaptation of visual domain categories.

REFERENCES

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In *ICML*, pp. 159–168, 2018.

David Barber and Felix Agakov. The im algorithm: a variational approach to information maximization. In *NIPS*, pp. 201–208. MIT Press, 2003.

Sagie Benaim and Lior Wolf. One-sided unsupervised domain mapping. In *NIPS*, pp. 752–762, 2017.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NIPS*, pp. 343–351, 2016.

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, July 2017a.

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, volume 1, pp. 7, 2017b.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *NIPS*, pp. 3733–3742, 2017.

Gabriela Csurka. A comprehensive survey on domain adaptation for visual applications. pp. 1–35. Springer, 2017.

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, pp. 2960–2967, 2013.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *ICML*, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pp. 2066–2073, 2012.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

Elyor Kodirov, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, pp. 2452–2460, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *iccv*, pp. 5543–5551. IEEE, 2017.

Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 469–477. Curran Associates, Inc., 2016. URL `http://papers.nips.cc/paper/6544-coupled-generative-adversarial-networks.pdf`.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NIPS*, pp. 700–708, 2017.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *CVPR*, pp. 1410–1417, 2014.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *ICML*, 2015.

Mingsheng Long, Jianmin Wang, Yue Cao, Jiaguang Sun, and S Yu Philip. Deep learning of transferable representation for scalable domain adaptation. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2027–2040, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *NIPS*, pp. 6673–6683, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 5, 2011.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *NIPS*, pp. 506–516, 2017.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. *ICML*, 2017.

Mathieu Salzmann, Carl Henrik Ek, Raquel Urtasun, and Trevor Darrell. Factorized orthogonal latent spaces. In *AISTATS*, pp. 701–708, 2010.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, pp. 443–450. Springer, 2016.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, volume 1, pp. 4, 2017.

Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *CVPR*, July 2017.

Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S Paek, and In So Kweon. Pixel-level domain transfer. In *ECCV*, pp. 517–532, 2016.

Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *ICLR*, 2017.

Jing Zhang, Wanqing Li, and Philip Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. In *CVPR*, July 2017.

Xu Zhang, Felix Xinnan Yu, Shih-Fu Chang, and Shengjin Wang. Deep transfer network: Unsupervised domain adaptation. *arXiv preprint arXiv:1503.00591*, 2015.

Han Zhao, Shanghang Zhang, Guanhang Wu, JoÃ£o P. Costeira, Jose Moura, and Geoffrey J. Gordon. Multiple source domain adaptation with adversarial training of neural networks. *ICLR workshops*, 05 2017.

Appendix

Table 3: Network architecture for the experiments.

Layer	Encoders (shared, private)
1	CONV-(N16,K7,S1), ReLU
2	CONV-(N32,K3,S2), ReLU
3	CONV-(N64,K3,S2), ReLU
4	RESBLK-(N64,K3,S1)
5	RESBLK-(N64,K3,S1)
6	RESBLK-(N64,K3,S1)
7	RESBLK-(N64,K3,S1)
Layer	**Decoder**
1	RESBLK-(N64,K3,S1)
2	RESBLK-(N64,K3,S1)
3	RESBLK-(N64,K3,S1)
4	RESBLK-(N64,K3,S1)
5	DCONV-(N32,K3,S2), ReLU
6	DCONV-(N16,K3,S2), ReLU
7	DCONV-(N1,K1,S1), TanH
Layer	**Discriminator**
1	CONV-(N4,K3,S1), ReLU
2	CONV-(N8,K3,S1), ReLU
3	CONV-(N16,K3,S1), ReLU
4	CONV-(N32,K3,S1), ReLU
5	CONV-(N1,K3,S1), ReLU
6	DENSE-(ND), Softmax
Layer	**Classifier**
1	CONV-(N4,K3,S1), ReLU
2	CONV-(N8,K3,S1), ReLU
3	CONV-(N16,K3,S1), ReLU
4	CONV-(N32,K3,S1), ReLU
5	CONV-(N1,K3,S1), ReLU
6	DENSE-(NC), Softmax

.1 Network Architecture

The network architecture used for the experiments is given in Table 3. We use the following abbreviation for ease of presentation: N=Neurons, K=Kernel size, S=Stride size, D=Number of Domains, C=number of Classes. The transposed convolutional layer is denoted by DCONV. The residual basic block is denoted as RESBLK.

.2 Proposed model's Algorithm

The detailed optimization process of the proposed model is shown in Algorithm 1.

.3 Additional Experiments for Multi-PIE dataset

The additional experiments for Multi-PIE dataset where we set $C08$ and $C09$ as source domain is available in Table 4.

Algorithm 1 MDTA-ITA Algorithm

Require: $\{\mathbf{X}, \mathbf{Y}, \mathbf{D}\}$:M domain datasets.
$\lambda_r, \lambda_c, \lambda_d$: Model hyper-parameters.
Ensure: $\theta_s, \theta_p, \theta_c, \phi, \psi$: Model parameters.
Initialize $\theta_s, \theta_p, \theta_c, \phi, \psi$;
repeat
Sample a mini-batch from each of source/target domain datasets.
Update $\{\theta_s\}$ by minimizing $\mathcal{L}_s$ in Eq.(8) through the gradient descent: $\theta_s = \theta_s - \eta \frac{\partial \mathcal{L}_s}{\partial \theta_s}$.
Update $\{\theta_p\}$ by minimizing $\mathcal{L}_p$ in Eq.(7) through the gradient descent:$\theta_p = \theta_p - \eta \frac{\partial \mathcal{L}_p}{\partial \theta_p}$.
Update $\{\theta_c\}$ by minimizing $\mathcal{L}_c$ in Eq.(6) through the gradient descent:$\theta_c = \theta_c - \eta \frac{\partial \mathcal{L}_c}{\partial \theta_c}$.
Update $\{\phi\}$ by minimizing $\mathcal{L}_\phi$ in Eq.(4) through the gradient descent:$\phi = \phi - \eta \frac{\partial \mathcal{L}_\phi}{\partial \phi}$.
Update $\{\psi\}$ by minimizing $\mathcal{L}_\psi$ in Eq.(3) through the gradient descent:$\psi = \psi - \eta \frac{\partial \mathcal{L}_s}{\partial \psi}$.
until Convergence;
return $\{\theta_s, \theta_p, \theta_c, \phi, \psi\}$.

Table 4: Mean classification accuracy on Multi-PIE classification. The best (red).

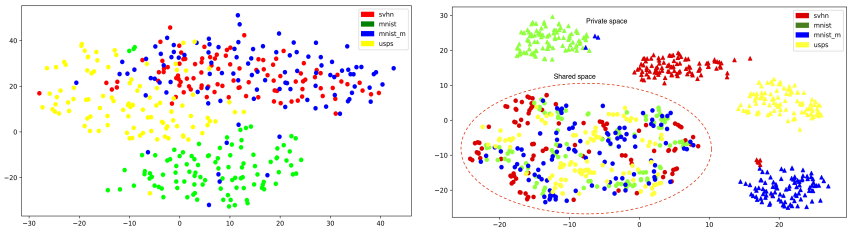
method	C08 → C05	C08 → C09	C08 → C13	C08 → C14	C09 → C05	C09 → C08	C09 → C13	C09 → C14
Source Only	33.70	50.10	50.80	40.13	33.32	48.24	49.24	36.19
1-NN	28.75	35.39	39.79	32.13	26.82	35.30	34.26	28.41
CORAL Sun & Saenko (2016)	35.89	55.79	60.00	40.67	35.89	51.56	50.45	40.67
DANNGanin et al. (2016)	40.20	56.89	55.83	43.25	50.63	**58.40**	55.81	48.90
ADDATzeng et al. (2017)	37.40	58.40	60.40	42.10	29.40	53.30	45.30	38.30
c-ADDA	41.60	39.65	50.00	46.25	45.01	52.14	37.43	43.26
DTNZhang et al. (2015)	44.13	57.42	55.89	45.76	44.53	57.34	52.43	51.55
c-DTN	45.10	49.78	47.43	45.79	49.80	55.69	50.10	52.31
PixelDABousmalis et al. (2017b)	**46.45**	44.33	44.87	46.83	45.63	16.37	45.43	47.00
UNITLiu et al. (2017)	43.88	43.99	44.47	44.47	44.47	43.95	44.64	44.47
DSNBousmalis et al. (2016)	**46.25**	47.50	**62.15**	39.72	45.85	56.65	56.5	42.87
c-DSN	45.82	44.64	45.60	46.32	45.18	45.52	44.79	47.37
mp-DSN	42.19	44.70	42.47	40.50	45.00	43.80	45.79	42.39
s-MTDA-ITA	44.77	45.61	60.00	46.70	49.06	55.33	59.90	50.64
c-MTDA-ITA	44.35	42.67	58.90	44.32	46.74	54.11	56.89	49.64
MTDA-ITA	**46.30**	**60.60**	60.50	**50.40**	**55.59**	57.80	**64.20**	**56.34**

.4 PACS DATASET

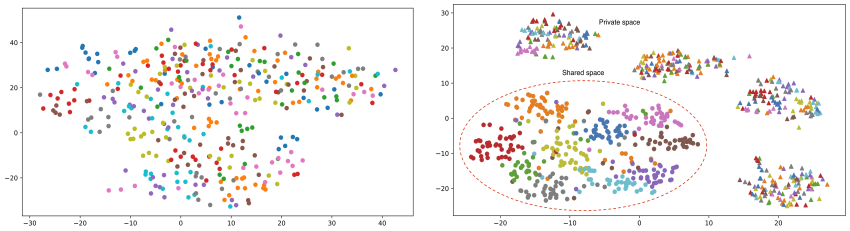
This dataset contains 9991 images ($227 \times 227 \times 3$ dimension) across 7 categories ('dog', 'elephant', 'giraffe','guitar', 'house', 'horse' and 'person') and 4 domains of different stylistic depictions ('Photo', 'Art painting', 'Cartoon' and 'Sketch'). The very diverse depiction styles provide a significant gap between domains, coupled with the small number of data samples, making it extremely challenging for domain adaptation. Consequently, the dataset was originally used for multi-source to single target domain adaptation Li et al. (2017). Instead, we tackle a significantly more challenging problem of single-source to multiple target adaptation. Tab. 5 shows the classification accuracy of various methods. **MTDA-ITA** consistently achieves the best performance for all transfer tasks. Evaluations were obtained by training all models (ADDA, DSN, and ours) from scratch on the PACS dataset. Note that the overall performance figures are low due to the extreme difficulty of the transfer task, induced by large differences among domains.

Table 5: Mean classification accuracy on PACS dataset classification. A:Art-painting, C:Cartoon, S:Sketch, P:Photo. The best (red).

method	P → A	P → C	P → S	A → P	A→ C	A → S
1-NN	15.28	18.16	25.60	22.70	19.75	22.70
ADDATzeng et al. (2017)	24.35	20.12	22.45	32.57	17.68	18.90
DSNBousmalis et al. (2016)	28.42	21.14	25.64	29.54	25.89	24.69
s-MTDA-ITA	28.02	21.64	26.24	31.06	25.09	25.89
c-MTDA-ITA	25.35	20.24	23.64	26.54	20.30	22.38
MTDA-ITA	**31.40**	**23.05**	**28.24**	**35.74**	**27.00**	**28.90**

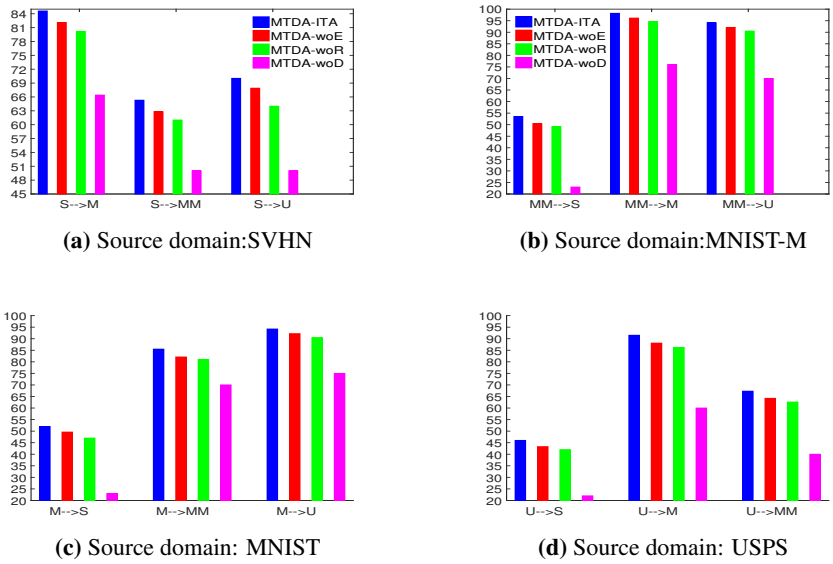


(a) Original features **(b)** MTDA-ITA private and shared features



(c) Original features **(d)** MTDA-ITA private and shared features

Figure 4: Feature visualization for embedding of digit datasets using t-sne algorithm. The top and bottom rows show the domains and classes, respectively, with color indicating domain and class membership. a,c) Original features. b,d) learned features (triangle marker: private features, circle marker: shared features). Large clusters in the right column represent points from the shared space, while the smaller ones are from the private spaces.



(a) Source domain:SVHN

(b) Source domain:MNIST-M

(c) Source domain: MNIST

(d) Source domain: USPS

Figure 5: Ablation of MTDA-ITA on Digit dataset. We show that each component of our method, Reconstruction loss, Classifier entropy loss with separating shared/private features, contributes to the overall performance.