Unsupervised Multi-Target Domain Adaptation: An Information Theoretic Approach

Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic

***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.**

***Index Terms*— Domain adaptation, mutual information, variational inference, adversarial learning.**

I. Introduction

IN REAL-WORLD data, the training and test instances often do not come from the same underlying distribution [1]. For example, in the task of object recognition/classification from image data, this 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 [2]–[8] 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 usually tackle the adaptation problems in one of the two settings: (semi)supervised **DA** [9] and unsupervised **DA** [10]. 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/adaptation of a classifier. In 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 address 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 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. This, in turn, 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, most of the **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 domains.

For instance, Zhao *et al.* [11] 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. It is intuitive that the access to multiple *labelled* source domains offers more adaptation flexibility for the target domain (i.e., by efficiently exploring the data labels across multiple source domains that are related to the target domain). Yet, it requires labels for the data from multiple source domains, which can be costly and time/labour-intensive to obtain. On the other hand, a simultaneous adaptation to *multiple* and *unlabelled* target domains may circumvent the

Manuscript received May 9, 2019; revised November 21, 2019; accepted December 15, 2019. Date of publication January 27, 2020; date of current version February 4, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Soma Biswas. *(Corresponding author: Behnam Gholami.)*

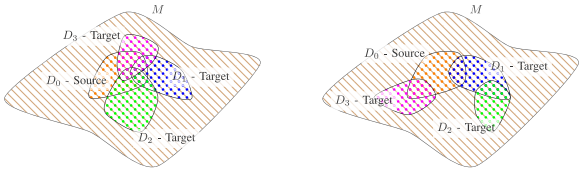
Behnam Gholami, Pritish Sahu, and Vladimir Pavlovic are with the Computer Science Department, Rutgers, The State University of New Jersey, Piscataway, NJ 08854 USA (e-mail: bb510@cs.rutgers.edu; ps851@cs.rutgers.edu; vladimir@cs.rutgers.edu).

Ognjen Rudovic is with the Media Lab, Affective Computing Group, Massachusetts Institute of Technology (MIT), Massachusetts, Cambridge, MA 02139 USA (e-mail: orudovic@mit.edu).

Konstantinos Bousmalis is with Google DeepMind, London EC4A 3TW, U.K. (e-mail: konstantinos@google.com).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TIP.2019.2963389



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

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

need for manual labeling of multiple domains/datasets. This **DA** scenario is important as we usually have access to multiple unlabeled domains; yet, it is also more challenging due to the lack of supervision in the target domains. Still, multi-target **DA** offers 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 pair-wise **DA** and naive-fusion of multiple target domains.

To this end, we propose a Multi-Target DA-Information-Theoretic-Approach (**MTDA-ITA**) for a 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 approach 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 pair-wise 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 [12]. 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 domain labels and the shared (domain-invariant) features. Consequently, 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, when compared to a pair-wise **DA** for source-target domains, a naive combination of multiple target domains, and state-of-the-art models applicable to this task. The contributions of this work are summarized below.

- We propose a novel information theoretic (IT) adversarial framework for disentangling the shared/private features from multiple domains. Different from most **uDA** works to date, our framework takes advantage of the unlabeled target samples during the classifier training (We did an ablation study in Sec. V-D on how much the information in unlabeled target samples is beneficial to the final performance).
- The adaptation from one source to multiple target domains has been relatively unexplored. To the best of our knowledge, this is the first work that enables adaptation to multiple target domains simultaneously, by also leveraging the cross-domain similarities.
- We conducted extensive experiments with detailed ablation studies on three well-known domain adaptation benchmarks to validate our approach on multiple domain adaptation, demonstrating the superiority of jointly adapting multiple unlabeled target domains over the state-of-the-art pair-wise **DA** methods.

II. Preliminaries

A. Information Theory: Background

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ denote a n-dimensional random variable with probability density function (**pdf**) given by $p(\mathbf{x})$. Shannon differential entropy [13] is defined in the usual way as $H(\mathbf{x}) = -\mathbb{E}_{\mathbf{x}}[\ln p(\mathbf{x})]$ where $\mathbb{E}$ denotes the expectation operator. Let $\mathbf{z} = (z_1, z_2, \ldots, z_m)$ denote a m-dimensional random variable with pdf $p(\mathbf{z})$. Then mutual information between two random variables, $\mathbf{x}$ and $\mathbf{z}$, is defined as $I(\mathbf{x};\mathbf{z}) = H(\mathbf{x}) + H(\mathbf{z}) - H(\mathbf{x},\mathbf{z})$. Mutual information can also be viewed as the reduction in uncertainty about one variable given another variable—i.e., $I(\mathbf{x};\mathbf{z}) = H(\mathbf{x}) - H(\mathbf{x}|\mathbf{z}) = H(\mathbf{z}) - H(\mathbf{z}|\mathbf{x})$.

III. Method

In this section, we describe our proposed information theoretic approach that supports domain adaptation for multiple target domains simultaneously, finding factorized latent spaces that are non-redundant, and that can capture explicitly the shared (domain invariant) and the private (domain dependent) features of the data well suited for better generalization for domain adaptation.

A. Problem Formulation

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

The latent space representation of the data point $\mathbf{x}$ 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. By factorizing the joint distribution $p(\mathbf{x}, \mathbf{y}, \mathbf{d}, \mathbf{z}_s, \mathbf{z}_p)$ as

$$p(\mathbf{x}, \mathbf{y}, \mathbf{d}, \mathbf{z}_s, \mathbf{z}_p) = p(\mathbf{x})p(\mathbf{d})p(\mathbf{z}_s|\mathbf{x})p(\mathbf{z}_p|\mathbf{x})p(\mathbf{y}|\mathbf{z}_s), \quad (1)$$

we propose to maximize the following objective function:

$$\mathcal{L}(\theta_s, \theta_p, \theta_c; \mathbf{x}, \mathbf{y}, \mathbf{d}) = \lambda_r I(\mathbf{x};\mathbf{z}) + \lambda_c I(\mathbf{y};\mathbf{z}_s) + \lambda_d \big(I(\mathbf{d};\mathbf{z}_p) - I(\mathbf{d};\mathbf{z}_s)\big), \quad (2)$$

where $p(\mathbf{x})$ and $p(\mathbf{d})$ denote the underlying (true) data distribution and domain label distribution, respectively, $I(\mathbf{x};\mathbf{y})$ denotes the Mutual Information between the random variables $\mathbf{x}$ and $\mathbf{y}$. λ_r, λ_c and λ_d denote the hyper-parameters controlling the weights of the objective terms. The proposed objective function (2) maximizes the three terms described below:

- $I(\mathbf{x};\mathbf{z})$: encourages the latent features (both shared and private) to preserve information about the data samples (that can be used to reconstruct $\mathbf{x}$ from $\mathbf{z}$).
- $I(\mathbf{y};\mathbf{z}_s)$: 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 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 $\mathbf{x}$, we enforce such constraint by jointly maximizing the term: $I(\mathbf{d};\mathbf{z}_p) - I(\mathbf{d};\mathbf{z}_s)$.

B. Optimization

The following lower bound for mutual information is derived using the non-negativity of KL-divergence [14]; i.e., $\Sigma_{\mathbf{x}} p(\mathbf{x}|\mathbf{z}) \ln \frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{x}|\mathbf{z})} \geq 0$ gives:

$$I(\mathbf{x};\mathbf{z}) \geq H(\mathbf{x}) + \mathbb{E}_{p(\mathbf{x},\mathbf{z})}[\ln q(\mathbf{x}|\mathbf{z};\phi)] \tag{3}$$

$q(\mathbf{x}|\mathbf{z};\phi)$ is any arbitrary distribution parameterized by ϕ. We need a variational distribution $q(\mathbf{x}|\mathbf{z};\phi)$ because the posterior distribution $p(\mathbf{x}|\mathbf{z}) = p(\mathbf{z}|\mathbf{x})p(\mathbf{x})/p(\mathbf{z})$ is intractable since the true data distribution $p(\mathbf{x})$ 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 parameterized by ψ.[1] We further drive lower bound for $I(\mathbf{y};\mathbf{z}_s)$ as $I(\mathbf{y};\mathbf{z}_s) \geq H(\mathbf{y}) + \mathbb{E}_{p(\mathbf{y},\mathbf{z}_s)}[\ln q(\mathbf{y}|\mathbf{z}_s;\theta_c)]$, where $q(\mathbf{y}|\mathbf{z}_s;\theta_c)$ is a variational distribution parameterized by θ_c approximating $p(\mathbf{y}|z_s)$.

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

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

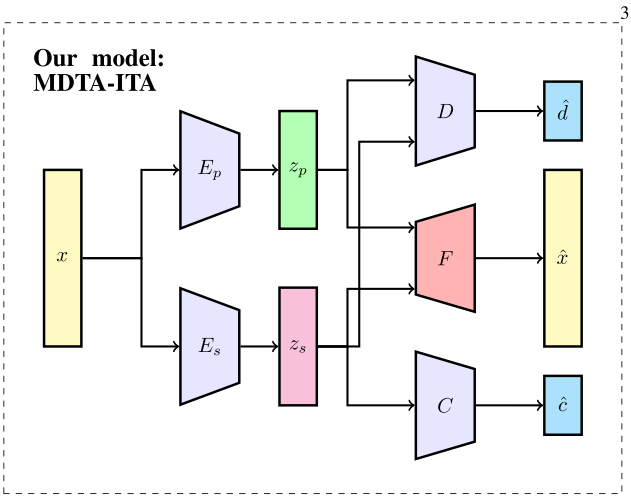


Fig. 2. **MTDA-ITA**: The encoder $E_s(\mathbf{x})$ captures the feature representations (z_s) for a given input sample $\mathbf{x}$ that are shared among domains. $E_p(\mathbf{x})$ captures domain–specific private features ($\mathbf{z}_p$) using the *shared* private encoder. The shared decoder $F(\mathbf{z}_p,\mathbf{z}_s)$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

We represent $p(\mathbf{d})$, $p(\mathbf{x})$, $p(\mathbf{y})$ as the empirical distribution of a finite training set (e.g. $p(\mathbf{d}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{d}-\mathbf{d}_i)$) as in the case of variational autoencoders (VAE) [15], [16], $p(\mathbf{z}_s|\mathbf{x})$, $p(\mathbf{z}_p|\mathbf{x})$ as deterministic functions of $\mathbf{x}$ as $p(\mathbf{z}_s|\mathbf{x}) = \delta\big(\mathbf{z}_s - E_s(\mathbf{x};\theta_s)\big)$ and $p(\mathbf{z}_p|\mathbf{x}) = \delta\big(\mathbf{z}_p - E_p(\mathbf{x};\theta_p)\big)$, and the variational distributions $q(\mathbf{y}|\mathbf{z}_s)$, $q(\mathbf{x}|\mathbf{z})$ and, $q(\mathbf{d}|\mathbf{z})$ as

$$\begin{aligned} q(\mathbf{y}|\mathbf{z}_s) &= \text{SoftMax}(C(\mathbf{z}_s;\theta_c)),\\ q(\mathbf{d}|\mathbf{z}) &= \text{SoftMax}(D(\mathbf{z};\psi)),\\ q(\mathbf{x}|\mathbf{z};\phi) &\propto \exp(\|\mathbf{x} - F(\mathbf{z};\phi)\|_1) \end{aligned} \tag{4}$$

where SoftMax$(\cdot)$ denotes the softmax or normalized exponential function [17], and $\|.\|_1$ denotes the L_1 norm. Then, the optimization task can be posed as a minimax saddle point problem, where we use adversarial training to maximize (2) w.r.t. the parameters $(\theta_s,\theta_p,\theta_c)$, and to minimize (2) w.r.t. the parameters (ϕ,ψ), using Stochastic Gradient Descent (SGD).

1) Optimizing the Parameters ϕ of the Decoder F:

$$\hat{\phi} = \arg\min_{\phi} \mathcal{L}_F = \frac{\lambda_r}{N}\sum_{i=1}^{N}\|\mathbf{x}_i - F\big(E_s(\mathbf{x}_i), E_p(\mathbf{x}_i)\big)\|_1. \tag{5}$$

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.

2) Optimizing the Parameters ψ of the Domain Classifier D:

$$\begin{aligned} \hat{\psi} = \arg\min_{\psi} \mathcal{L}_D = &-\frac{\lambda_d}{N}\sum_{i=1}^{N}\mathbf{d}_i^{\top}\ln D\big(E_s(\mathbf{x}_i)\big)\\ &-\frac{\lambda_d}{N}\sum_{i=1}^{N}\mathbf{d}_i^{\top}\ln D\big(E_p(\mathbf{x}_i)\big). \end{aligned} \tag{6}$$

$D(\mathbf{z};\psi)$ 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. 6 encourage D to correctly predict the domain labels from the shared and private features, respectively.

3) Optimizing the Parameters θ_c of the Label Classifier C:

$$\hat{\theta}_c = \arg\min_{\theta_c} \big\{ - H(\mathbf{y}) - \mathbb{E}_{p(\mathbf{y},\mathbf{z}_s)}\big[\ln q(\mathbf{y}|\mathbf{z}_s)\big]\big\}. \tag{7}$$

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

$$\begin{aligned}\hat{\theta}_c = \arg\min_{\theta_c} \mathcal{L}_C = &-\frac{1}{N}\sum_{i=1}^{N_s} \mathbf{y}_i^T \ln C\big(E_s(\mathbf{x}_i)\big) \\ &-\frac{\lambda_c}{N-N_s}\sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)^\top \ln C\big(E_s(\mathbf{x}_i)\big) + \frac{\lambda_c}{N-N_s} \\ &\times \sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)^\top \ln\left(\frac{1}{N-N_s}\sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)\right),\end{aligned} \tag{8}$$

where N_s denotes the number of source samples. Intuitively, we enforce the classifier C to correctly predict the class labels of the source samples by the first term in Eq. 8. We use the second term to minimize the entropy of $q(\mathbf{y}|\mathbf{z}_s)$ for the target samples; effectively, reducing the effects of "confusing" labels of target samples, as given by $p(\mathbf{y}|\mathbf{z}_s)$ 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 by minimizing only the entropy (second term), we may arrive at a degenerate solution where every target point $\mathbf{x}_t$ is assigned to the same class. Hence, the last term encourages the classifier C to have balanced labeling for the target samples where it reaches its minimum, $\ln K$, when each class is selected with uniform probability.

4) Optimizing the Parameter θ_s of the Shared Encoder E_s:

$$\begin{aligned}\hat{\theta}_s = \arg\min_{\theta_s} \mathcal{L}_S = &\frac{\lambda_r}{N}\sum_{i=1}^{N} \|\mathbf{x} - F\big(E_s(\mathbf{x}_i), E_p(\mathbf{x}_i)\big)\|_1 \\ &-\frac{\lambda_c}{N}\sum_{i=1}^{N_s} \mathbf{y}_i^T \ln C\big(E_s(\mathbf{x}_i)\big) + \frac{\lambda_d}{N}\sum_{i=1}^{N} \mathbf{d}_i^\top \ln D\big(E_s(\mathbf{x}_i)\big) \\ &-\frac{\lambda_c}{N-N_s}\sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)^\top \ln C\big(E_s(\mathbf{x}_i)\big) + \frac{\lambda_c}{N-N_s} \\ &\times \sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)^\top \ln\left(\frac{1}{N-N_s}\sum_{i=N_s+1}^{N} C\big(E_s(\mathbf{x}_i)\big)\right).\end{aligned} \tag{9}$$

The first term in Eq. 9 encourages the shared encoder E_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 when predicting the domain labels $\mathbf{d}$, given the shared features $\mathbf{z}_s$. The 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 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.

Algorithm 1 MTDA-ITA Algorithm

Require: $\{\mathbf{X}, \mathbf{Y}, \mathbf{D}\}$:M domain datasets.
$\lambda_r, \lambda_c, \lambda_d$: Model hyper-parameters.
η: Learning rate.
Ensure: $\theta_s, \theta_p, \theta_c, \phi, \psi$: Model parameters.
1: Initialize $\theta_s, \theta_p, \theta_c, \phi, \psi$; t=0;
2: **repeat**
3: Sample a mini-batch from each of source/target domain datasets.
4: $\theta_s^{(t+1)} \leftarrow \theta_s^{(t)} - \eta\frac{\partial \mathcal{L}_s}{\partial \theta_s}$ through eq. 9.
5: $\theta_p^{(t+1)} \leftarrow \theta_p^{(t)} - \eta\frac{\partial \mathcal{L}_p}{\partial \theta_p}$ through eq. 10.
6: $\theta_c^{(t+1)} \leftarrow \theta_c^{(t)} - \eta\frac{\partial \mathcal{L}_c}{\partial \theta_c}$ through eq. 8
7: $\phi^{(t+1)} \leftarrow \phi^{(t)} - \eta\frac{\partial \mathcal{L}_\phi}{\partial \phi}$ through eq. 6.
8: $\psi^{(t+1)} \leftarrow \psi^{(t)} - \eta\frac{\partial \mathcal{L}_s}{\partial \psi}$ through eq. 5.
9: $t \leftarrow t+1$
10: **until** $t <$ # epochs;
11: return $\{\theta_s, \theta_p, \theta_c, \phi, \psi\}$.

5) Optimizing the Parameter θ_p of the Private Encoder E_p:

$$\begin{aligned}\hat{\theta}_p = \arg\min_{\theta_p} \mathcal{L}_P = &\frac{\lambda_r}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - F\big(E_s(\mathbf{x}_i), E_p(\mathbf{x}_i)\big)\|_1 \\ &-\frac{\lambda_d}{N}\sum_{i=1}^{N} \mathbf{d}_i^\top D\big(E_p(\mathbf{x}_i)\big).\end{aligned} \tag{10}$$

The first term in Eq. 10 encourages the private encoder E_p to preserve the recovery ability of the private features. The second term 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.

To train our model, we alternate between updating the shared encoder E_s, the private encoder E_p, the decoder F, the classifier C, and the domain classifier D using the SGD algorithm (see Algorithm 1 for more details).

IV. Related Work

There has been extensive prior work on domain adaptation [10], [18]–[24]. 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 [25]–[33]. 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 [33]–[35]. **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) [36] applied **MMD** to layers embedded in a **RKHS**, effectively matching higher order statistics of the two distributions. The deep Correlation Alignment (**CORAL**) method [37] attempts to match the mean and covariance of the two distributions. Deep Transfer Network (**DTN**) [38] 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 [27]–[30] 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 and Lempitsky [39] 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**) [40] 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.* [41] makes a shared-latent space assumption and proposes an unsupervised image-to-image translation (**UNIT**) framework based on Coupled GANs [42]. Another example is the pixel-level **DA** models that perform the distribution alignment not in the feature space but directly in raw pixel space. **PixelDA** [33] 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.* [11] 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. Although both our model and **MDAN** use the similar notion of the domain classifier to minimize the domain mismatch in shared space, the domain classifier induced by our information-theoretic (IT) loss also acts to separate domains in the private space (see Eqs. 6 & 10 for more details), improving the essential reconstruction ability, similar to [43]. We provided how our model is related to multiple domain transfer networks, and Information Theoretic representation learning approaches, in the Supplementary Material (SM). We also clearly contrasted our model with **DSN** model which also uses 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 in the SM.

V. 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** [44], **MNIST-M** [45], **SVHN** [46], **USPS** [40], **Multi-PIE** expression recognition dataset [47], and three multi-domain object recognition datasets namely **PACS** [48], **Office-Home** [49], and **Domain-Net** [50] datasets (the details of the experiments for **Office-Home** and **Domain-Net** are available in the SM). Fig. 3 illustrate image samples from different datasets and domains. We evaluate the performance of all methods with classification accuracy metric. We repeated each experiment 5 times and report the average and the standard deviation of the accuracy.

We used ADAM [51] for training; the learning rate was set to 0.0002 and momentum parameters 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 $\lambda_r = 1.0, \lambda_c = 0.01, \lambda_d = 0.20$. All the used architectures replicate those of state-of-the-art methods. Specifically, we use the network structure similar to **UNIT** [41]. Precisely, our private/shared encoders consisted of three convolutional layers as the front-end and four basic residual blocks as the back-end.

The decoder consisted of four basic residual blocks as the front-end and four 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 available in the SM.

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 our model 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** ([37]), **DANN** [39], **ADDA** [40], **DTN** [38], **UNIT** [41], **PixelDA** [33], **MCDA** [18], and **DSN** [43]. 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**, **c-DSN**, and **c-MCDA**) as well. We also extend **DSN** to multiple domains by (i) having one private encoder for all domains denoted by (**1p-DSN**), (ii) adding multiple private

TABLE I

CLASSIFICATION RESULTS ON DIGIT DATASETS. 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. **1P-DSN**: EXTENDED **DSN** WITH SINGLE PRIVATE ENCODER.**MP-DSN**: EXTENDED **DSN** WITH MULTIPLE PRIVATE ENCODER. LAST COLUMN SHOWS THE AVERAGE RANK OF EACH METHOD OVER ALL ADAPTATION PAIRS.*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 HYPER-PARAMETERS

method	S → M	S → MM	S → U	M → S	M → MM	M → U	Ave. ranking
Source Only	62.10 ± 0.60	40.43 ± 0.70	39.90 ± 0.60	30.29 ± 0.59	55.98 ± 0.48	78.30 ± 0.38	14.00
1-NN	35.86	18.21	29.31	28.01	12.58	41.22	15.00
CORAL [37]	63.10 ± 0.61	54.37 ± 0.53	50.15 ± 0.63	33.40 ± 0.74	57.70 ± 0.69	81.05 ± 0.80	11.33
DANN [45]	73.80 ± 0.49	61.05 ± 0.80	62.54 ± 0.91	35.50 ± 0.65	77.40 ± 0.73	81.60 ± 0.60	8.75
ADDA [40]	77.68 ± 0.92	64.23 ± 0.70	64.10 ± 0.79	30.04 ± 0.98	91.47 ± 1.0	90.51 ± 0.80	6.43
c-ADDA	80.10 ± 0.69	56.80 ± 0.79	64.80 ± 0.88	27.50 ± 0.86	83.30 ± 0.90	84.10 ± 0.98	8.95
DTN [38]	81.40 ± 0.42	63.70 ± 0.39	60.12 ± 0.52	40.40 ± 0.50	85.70 ± 0.39	85.80 ± 0.46	6.04
c-DTN	82.10 ± 0.62	59.30 ± 0.59	56.87 ± 0.65	38.32 ± 0.50	80.90 ± 0.80	79.31 ± 0.78	7.96
MCDA [18]	**96.20** ± 0.47	60.80 ± 0.49	65.42 ± 0.32	44.60 ± 0.40	94.90 ± 0.41	**96.50** ± 0.46	2.74
c-MCDA	94.30 ± 0.62	56.42 ± 0.59	61.87 ± 0.35	40.51 ± 0.40	92.10 ± 0.40	93.51 ± 0.38	3.84
PixelDA [33]	–	–	–	–	**98.10***	94.10*	–
UNIT ([41])	90.6*	–	–	–	–	92.90	–
DSN [43]	82.70 ± 0.37	64.80 ± 0.40	65.30 ± 0.28	49.30 ± 0.30	83.20 ± 0.30	91.65 ± 0.40	2.85
c-DSN	83.10 ± 0.20	60.56 ± 0.36	60.35 ± 0.59	46.80 ± 0.45	80.49 ± 0.40	88.21 ± 0.38	4.84
1p-DSN	81.00 ± 0.47	58.22 ± 0.68	58.06 ± 0.48	45.11 ± 0.33	77.33 ± 0.52	85.16 ± 0.63	4.90
mp-DSN	83.40 ± 0.30	61.00 ± 0.50	58.10 ± 0.64	47.35 ± 0.40	79.30 ± 0.59	86.45 ± 0.71	5.33
s-MTDA-ITA	82.90 ± 0.13	63.10 ± 0.28	63.54 ± 0.30	49.60 ± 0.25	87.42 ± 0.19	89.21 ± 0.28	2.88
c-MTDA-ITA	79.20 ± 0.28	59.90 ± 0.30	63.70 ± 0.26	45.30 ± 0.30	82.12 ± 0.22	87.47 ± 0.25	4.25
MTDA-ITA	87.70 ± 0.24	**68.30** ± 0.15	**70.03** ± 0.20	**56.01** ± 0.21	93.50 ± 0.18	94.20 ± 0.20	**1.16**

encoders to it denoted by (**mp-DSN**) and contrast them with our model.

A. Digits Datasets

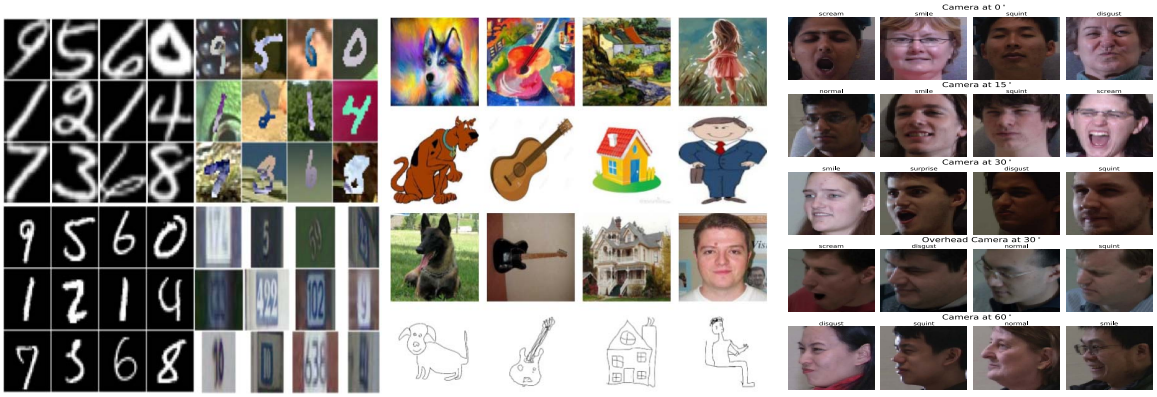
We combine four popular digits datasets (**MNIST**, **MNIST-M**, **SVHN**, and **USPS**) to build the multi-target domain dataset. All images were uniformly rescaled to 32×32. We take each of **MNIST-M**, **SVHN**, **USPS**, and **MNIST** as source domain in turn, and the rest as targets. We use all labeled source images and all unlabeled target images, following the standard evaluation protocol for unsupervised domain adaptation [45], [52]. We show the accuracy of different methods in Tab. I (additional results are available in the SM). The results show that first of all **c-MTDA-ITA** has worse performance than **s-MTDA-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. 8 for updating the classifier C) during the learning process, compared to when no data from target domain is used (See the ablation study section 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 both **mp-DSN**, and **1p-DSN**. This can be due to (i) having multiple private encoders increase the number of parameters that may lead to **mp-DSN** 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**.

B. 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. Tab. II show the classification accuracy of different methods (additional results are available in the SM). 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.

C. PACS Dataset

This dataset contains 9991 images ($227 \times 227 \times 3$ dimension) across 7 categories ('dog', 'elephant', 'giraffe','guitar',



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

Fig. 3. Exemplary images from different datasets. a) Digits datasets (top left: USPS, top right: MNIST-M, bottom left: MNIST, bottom right: SVHN), 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).

TABLE II

CLASSIFICATION RESULTS ON MULTI-PIE DATASET. LAST COLUMN SHOWS THE AVERAGE RANK OF EACH METHOD OVER ALL ADAPTATION PAIRS. THE BEST IS SHOWN IN RED

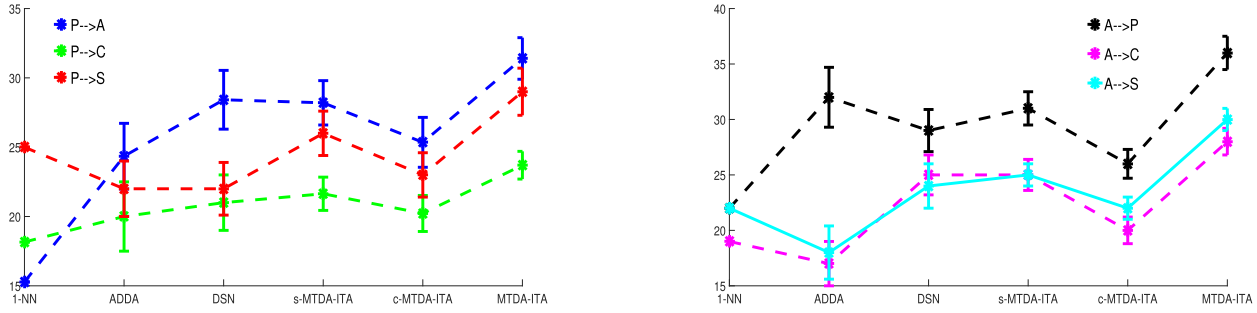
method	C13 → C05	C13 → C08	C13 → C09	C13 → C14	C14 → C05	C14 → C08	C14 → C09	C14 → C13	Ave. ranking
Source Only	50.79 ± 0.33	45.90 ± 0.50	40.04 ± 0.40	59.68 ± 0.29	60.03 ± 0.55	36.80 ± 0.61	40.11 ± 0.50	60.57 ± 0.36	16.08
1-NN	33.21	37.01	34.45	48.79	47.44	28.24	30.86	44.86	17.00
CORAL [37]	54.89 ± 0.52	48.90 ± 0.48	40.30 ± 0.53	68.90 ± 0.35	59.98 ± 0.45	40.63 ± 0.55	40.80 ± 0.53	65.11 ± 0.45	11.95
DANN [45]	57.86 ± 0.41	50.30 ± 0.43	45.30 ± 0.50	70.68 ± 0.35	57.20 ± 0.45	40.22 ± 0.55	40.77 ± 0.45	70.50 ± 0.55	9.92
ADDA [40]	64.83 ± 0.69	63.20 ± 0.45	55.48 ± 0.65	74.25 ± 0.55	73.62 ± 0.75	43.56 ± 0.95	38.68 ± 0.95	72.84 ± 0.75	9.33
c-ADDA	59.20 ± 0.25	30.70 ± 0.63	53.20 ± 0.40	68.33 ± 0.35	65.88 ± 0.38	30.60 ± 0.61	45.34 ± 0.48	64.30 ± 0.40	11.50
DTN [38]	63.78 ± 0.29	60.45 ± 0.35	60.55 ± 0.35	72.60 ± 0.25	70.67 ± 0.30	41.55 ± 0.65	41.45 ± 0.45	70.67 ± 0.45	8.75
c-DTN	57.53 ± 0.42	55.24 ± 0.45	57.14 ± 0.39	65.16 ± 0.35	63.80 ± 0.42	38.97 ± 0.71	39.80 ± 0.65	62.10 ± 0.45	10.92
MCDA [18]	71.68 ± 0.29	60.35 ± 0.35	62.75 ± 0.35	80.20 ± 0.25	**88.17** ± 0.30	53.15 ± 0.35	55.15 ± 0.35	84.17 ± 0.42	2.55
c-MCDA	65.38 ± 0.39	57.15 ± 0.35	60.35 ± 0.25	80.00 ± 0.25	83.07 ± 0.40	50.25 ± 0.35	54.05 ± 0.28	80.02 ± 0.40	4.75
PixelDA [43]	45.68 ± 0.52	44.95 ± 0.42	44.45 ± 0.55	**90.50** ± 0.25	46.28 ± 0.60	45.89 ± 0.61	44.45 ± 0.51	69.15 ± 0.45	9.95
UNIT [41]	44.14 ± 0.10	44.47 ± 0.11	44.21 ± 0.12	44.47 ± 0.11	43.03 ± 0.1	44.44 ± 0.15	44.47 ± 0.15	44.47 ± 0.05	11.07
DSN [43]	64.15 ± 0.30	57.70 ± 0.38	49.15 ± 0.45	80.75 ± 0.27	82.20 ± 0.28	38.75 ± 0.53	45.00 ± 0.25	80.50 ± 0.35	5.15
c-DSN	57.34 ± 0.45	31.63 ± 0.60	51.17 ± 0.40	74.52 ± 0.37	82.01 ± 0.35	34.25 ± 0.58	42.63 ± 0.55	79.42 ± 0.35	8.20
1p-DSN	55.84 ± 0.50	30.03 ± 0.50	49.06 ± 0.38	72.11 ± 0.50	81.22 ± 0.45	33.33 ± 0.58	42.03 ± 0.24	78.78± 0.57	8.63
mp-DSN	55.20 ± 0.46	30.40 ± 0.50	47.80 ± 0.35	75.30 ± 0.25	80.75 ± 0.20	30.20 ± 0.55	43.00 ± 0.35	79.02 ± 0.40	8.88
s-MTDA-ITA	70.10 ± 0.27	58.90 ± 0.25	58.10 ± 0.27	80.12 ± 0.15	82.05 ± 0.18	45.90 ± 0.30	52.67 ± 0.30	81.60 ± 0.24	3.65
c-MTDA-ITA	60.34 ± 0.17	55.67 ± 0.21	57.10 ± 0.23	73.50 ± 0.20	76.80 ± 0.10	43.10 ± 0.12	48.10 ± 0.14	80.90 ± 0.11	5.01
MTDA-ITA	**78.40** ± 0.2	**66.70** ± 0.17	**70.30** ± 0.14	85.49 ± 0.11	87.20 ± 0.10	**61.40** ± 0.14	**60.05** ± 0.13	**86.70** ± 0.10	**1.20**

'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 [48]. Instead, we tackle a significantly more challenging problem of single-source to multiple target adaptation. Fig. 4 shows the mean classification accuracy averaged over 5 trials (with error bars) 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 are low due to the extreme difficulty of the transfer task, induced by large differences among domains.

D. Ablation Studies

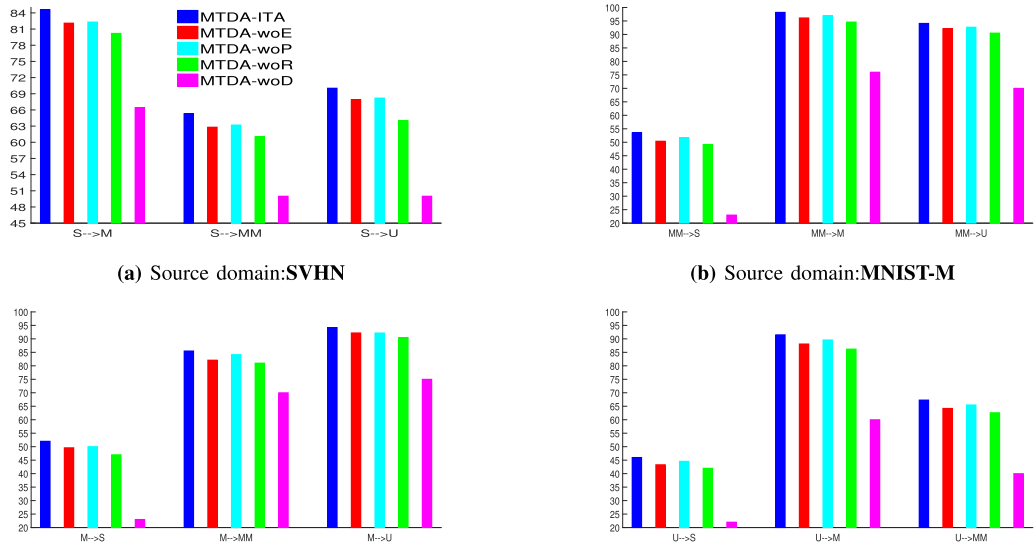
We performed an ablation study on the proposed model measuring impact of various terms on the model's performance. To this end, we conducted additional experiments for the digit datasets with different components ablation, i.e., training without the reconstruction loss (denoted as **MTDA-woR**) by setting $\lambda_r = 0$, training without the classifier entropy loss (denoted as **MTDA-woE**) by setting $\lambda_c = 0$, training without the multi-domain separation loss (denoted as **MTDA-woD**) by setting $\lambda_d = 0$.

As can be seen from Fig. 5, disabling each of the above components leads to degraded performance. More precisely, the average drop by disabling the classifier entropy loss is $\approx$ 3.5%. Similarly, by disabling the reconstruction loss and the multi-domain separation loss, we have $\approx$ 4.5% and $\approx$ 22% average drop in performance, respectively. Clearly, by disabling the multi-domain separation loss, the accuracy drops significantly 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,



(a) Source domain:**Art-painting** **(b)** Source domain:**Photo**

Fig. 4. Classification results on PACS dataset. A:Art-painting, C:Cartoon, S:Sketch, P:Photo.



(a) Source domain:**SVHN** **(b)** Source domain:**MNIST-M**

(c) Source domain: **MNIST** **(d)** Source domain: **USPS**

Fig. 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.

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 order to examine the effect of the private features on the model's classification performance, we took the **MTDA-ITA** and trained it without the private encoder (denoted as **MTDA-woP**). As Fig. 5 shows, without the private features, the model performed consistently worse ($\approx 2\%$ average drop in performance) in all scenarios. This demonstrates explicitly modeling what is unique to each domain can improve the model's ability to extract domain-invariant features. In summary, this ablation study showed that the individual components bring complimentary information to achieve the best classification results.

E. Analysis of Shared/Private Space Embedding

In the experiments conducted, we showed that our approach is able to achieve better performance than the competing methods including the extended **DSN** with one private encoder (**1p-DSN**) which is the most similar method to ours.

We use t-SNE [53] on digit datasets to compare the visualization of the shared and private features of **MTDA-ITA** with **DSN**.

Fig. 6 depicts the embedding of the **MTDA-ITA** learned private/shared features using those of **1p-DSN** and the original features from different domains for Digit datasets (**SVHN** is the source).

Notice that both **MTDA-ITA** and **1p-DSN** reduces the domain mismatch for the shared features (circle markers in Fig. 6) and separate the shared features from private features. On the other hand, **MTDA-ITA** increases the domain separation for the private features (triangle markers, pure and well-separated domain clusters in Fig. 6c) while **1p-DSN** is unable to enforce the private representation of different domains to be different (Fig. 6e) that may result in redundancy of different private spaces. This is partially due to the proposed multi-domain separation loss through the use of the domain classifier D, which penalizes the domain mismatch for the shared features and rewards the mismatch for the private features, something the **1p-DSN** fails to account for. Moreover, as supported by the quantitative results in Tab. I, the class label

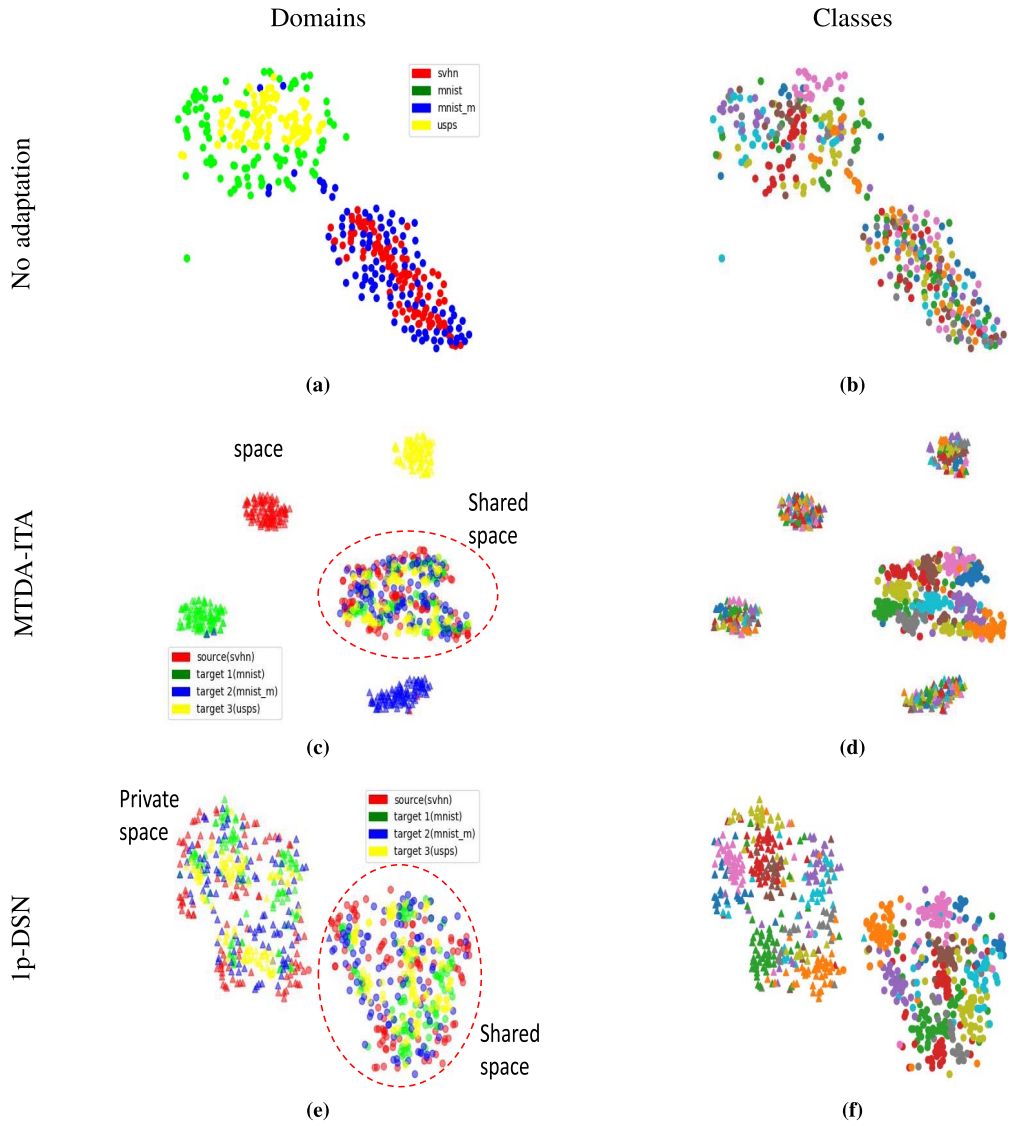


Fig. 6. Feature visualization for embedding of digit datasets using t-SNE algorithm. The first and the second columns show the domains and classes, respectively, with color indicating domain and class membership. (a), (b) Original features. (c), (d) learned features for **MTDA-ITA** (triangle marker: 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. (e), (f) learned features for **1p-DSN**.

separation in the shared space for **1p-DSN** (Fig. 6f), is still evident but not as strong as in the **MTDA-ITA** (Fig. 6d). This can be attributed to the lack of redundancy in the private space that helps **MTDA-ITA** to learn more disentangled shared features and usage of the target samples during training, something the **1p-DSN** fails to account for.

VI. Conclusion

This paper presented an information theoretic end-to-end approach to **uDA** 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. Our use of a single private domain encoder results in a highly scalable model, easily optimized using established back-propagation approaches. Results on three 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

[1] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. 30th AAAI Conf. Artif. Intell.*, Dec. 2016, pp. 901–907.

[2] S. Li, S. Song, G. Huang, Z. Ding, and C. Wu, "Domain invariant and class discriminative feature learning for visual domain adaptation," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4260–4273, Sep. 2018.

[3] H. Lu, C. Shen, Z. Cao, Y. Xiao, and A. Van Den Hengel, "An embarrassingly simple approach to visual domain adaptation," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3403–3417, Jul. 2018.

[4] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2960–2967.

[5] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.

[6] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2452–2460.

[7] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, "Pixel-level domain transfer," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 517–532.

[8] B. Geng, D. Tao, and C. Xu, "DAML: Domain adaptation metric learning," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2980–2989, Oct. 2011.

[9] Z. Ding, N. M. Nasrabadi, and Y. Fu, "Semi-supervised deep domain adaptation via coupled neural networks," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5214–5224, Nov. 2018.

[10] G. Csurka, "A comprehensive survey on domain adaptation for visual applications," *Domain Adaptation Comput. Vis. Appl.*, vol. 17, pp. 1–35, Jun. 2017.

[11] H. Zhao, S. Zhang, G. Wu, J. P. Costeira, J. Moura, and G. J. Gordon, "Multiple source domain adaptation with adversarial training of neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR) Workshops*, 2017, pp. 37–47.

[12] M. Salzmann, C. H. Ek, R. Urtasun, and T. Darrell, "Factorized orthogonal latent spaces," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 701–708.

[13] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, Jan. 1991.

[14] D. Barber and F. Agakov, "The IM algorithm: A variational approach to information maximization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2003, pp. 201–208.

[15] M. E. Abbasnejad, A. Dick, and A. van den Hengel, "Infinite variational autoencoder for semi-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 781–790.

[16] Y. Pu *et al.*, "Adversarial symmetric variational autoencoder," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 4330–4339.

[17] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*. Berlin, Germany: Springer, 1990, pp. 227–236.

[18] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732.

[19] D. Das and C. G. Lee, "Sample-to-sample correspondence for unsupervised domain adaptation," *Eng. Appl. Artif. Intell.*, vol. 73, pp. 80–91, Aug. 2018.

[20] J.-T. Zhou, I. W. Tsang, and S. J. Pan, "Multi-class heterogeneous domain adaptation," *J. Mach. Learn. Res.*, vol. 20, no. 57, pp. 1–31, 2019.

[21] D. Das and C. G. Lee, "Unsupervised domain adaptation using regularized hyper-graph matching," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3758–3762.

[22] H. Zhao, J. Hu, Z. Zhu, A. Coates, and G. Gordon, "Deep generative and discriminative domain adaptation," in *Proc. 18th Int. Conf. Auto. Agents MultiAgent Syst.*, 2019, pp. 2315–2317.

[23] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1853–1865, Sep. 2017.

[24] D. Das and C. G. Lee, "Graph matching and pseudo-label guided deep unsupervised domain adaptation," in *Proc. Int. Conf. Artif. Neural Netw.*, 2018, pp. 342–352.

[25] B. Gholami *et al.*, "PUnDA: Probabilistic unsupervised domain adaptation for knowledge transfer across visual categories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 3581–3590.

[26] M. Kim, P. Sahu, B. Gholami, and V. Pavlovic, "Unsupervised visual domain adaptation: A deep max-margin Gaussian process approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4380–4390.

[27] S. Benaim and L. Wolf, "One-sided unsupervised domain mapping," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 752–762.

[28] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 3733–3742.

[29] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 321–330.

[30] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 730–738.

[31] B. Gholami, P. Sahu, M. Kim, and V. Pavlovic, "Task-discriminative domain alignment for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Oct. 2019, pp. 100–108.

[32] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 69–77.

[33] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, vol. 1, no. 2, pp. 7–15.

[34] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (CMD) for domain-invariant representation learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 872–882.

[35] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1410–1417.

[36] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1002–1010.

[37] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 443–450.

[38] X. Zhang, F. X. Yu, S.-F. Chang, and S. Wang, "Deep transfer network: Unsupervised domain adaptation," 2015, *arXiv:1503.00591*. [Online]. Available: https://arxiv.org/abs/1503.00591

[39] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 456–464.

[40] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, vol. 1, no. 2, pp. 4–12.

[41] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 700–708.

[42] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 469–477.

[43] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 343–351.

[44] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[45] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, May 2015.

[46] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, vol. 2011, no. 2, p. 5, 2011.

[47] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image Vis. Comput.*, vol. 28, no. 5, pp. 807–813, May 2010.

[48] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5543–5551.

[49] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5018–5027.

[50] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," 2018, *arXiv:1812.01754*. [Online]. Available: https://arxiv.org/abs/1812.01754

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 43–53.

[52] M. Long, J. Wang, Y. Cao, J. Sun, and P. S. Yu, "Deep learning of transferable representation for scalable domain adaptation," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2027–2040, Aug. 2016.

[53] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.