Data Augmentation with norm-AE and Selective Pseudo-Labelling for Unsupervised Domain Adaptation

Qian Wang, Fanlin Meng, Toby P. Breckon

Department of Computer Science, Durham University, UK.
Department of Mathematical Sciences, University of Essex, UK.
qian.wang173@hotmail.com

Abstract

We address the Unsupervised Domain Adaptation (UDA) problem in image classification from a new perspective. In contrast to most existing works which either align the data distributions or learn domain-invariant features, we directly learn a unified classifier for both the source and target domains in the high-dimensional homogeneous feature space without explicit domain alignment. To this end, we employ the effective Selective Pseudo-Labelling (SPL) technique to take advantage of the unlabelled samples in the target domain. Surprisingly, data distribution discrepancy across the source and target domains can be well handled by a computationally simple classifier (e.g., a shallow Multi-Layer Perceptron) trained in the original feature space. Besides, we propose a novel generative model *norm-AE* to generate synthetic features for the target domain as a data augmentation strategy to enhance the classifier training. Experimental results on several benchmark datasets demonstrate the pseudo-labelling strategy itself can lead to comparable performance to many state-of-the-art methods whilst the use of *norm-AE* for feature augmentation can further improve the performance in most cases. As a result, our proposed methods (i.e. *naive-SPL* and *norm-AE-SPL*) can achieve comparable performance with state-of-the-art methods with the average accuracy of 93.4% and 90.4% on Office-Caltech and ImageCLEF-DA datasets, and achieve competitive performance on Digits, Office31 and Office-Home datasets with the average accuracy of 97.2%, 87.6% and 68.6% respectively.

Key words: Unsupervised domain adaptation, Data augmentation, Variational autoencoder, Selective pseudo-labelling

1. Introduction

In the last decade, impressive progress has been made in supervised image classification with the advancement of deep learning [17] and the availability of large scale image datasets such as ImageNet [8]. One key to the success of deep neural networks in image classification is the access of sufficient annotated images which are usually unavailable in many real-world applications such as image classification in the invisible spectrum, medical image classification, etc.. To address the issues of training data scarcity in practice, a variety of techniques (e.g., semi-supervised learning [69], zero-shot learning [60, 61, 27, 46], domain adaptation [56, 67, 58, 31, 28]) can be employed based on the availability of varied training data resources. Among these, Unsupervised Domain Adaptation (UDA) assumes the access of labelled data only from the *source domain* where the labelled data are easier to obtain but the data distribution is different from that of the *target domain* in which the task of interest resides. As a result, a classifier trained on the labelled source domain suffers from a significant performance drop when directly applied to the target domain. Unsupervised domain adaptation problems are common in real-world applications. For example, recognizing objects in X-ray baggage screening imagery [57] can be a challenging task due to the difficulty of data collection in this domain but regular images are much easier to obtain. In this case, domain adaptation techniques can play a crucial role in making the most of large-scale regular images from the source domain and limited X-ray images from the target domain.

Existing UDA approaches try to align the source and domain data distributions by feature transformation (e.g., projecting features into a subspace) [59, 5, 58] or learning domain-invariant features from images via specially designed deep neural networks [32, 26]. Subsequently, simple classifiers such as Nearest Neighbours (NN) or Support Vector Machines (SVM) can be employed in the learned domain-invariant feature space. Although impressive performance has been achieved in prior works by aligning source and target domains in a learned feature space [58], we argue that *the need for explicit domain alignment before learning a classifier can be relaxed for good performance in UDA problems*. To justify this argument, we demonstrate that a unified classifier can be trained for both source and target domain data in the original high-dimensional homogeneous feature space due to the *blessing of dimensionality* [18] despite the existence of domain shift. From this perspective, the key challenge of UDA problems is the lack of labelled data in the target domain for supervised learning.

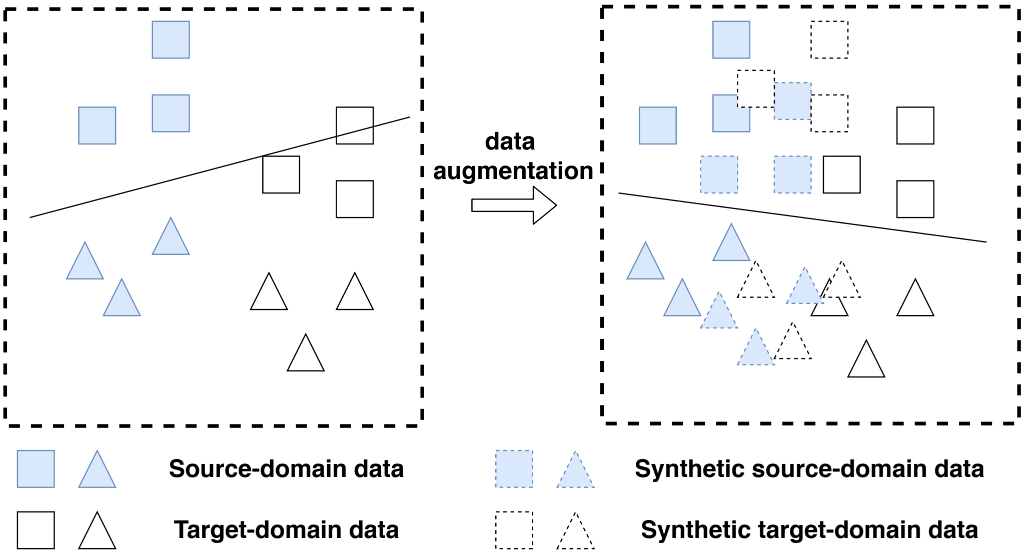


Figure 1: An illustration of how data augmentation by synthesizing source and target domain data can benefit unsupervised domain adaptation. Left: classifier trained with labelled source-domain data only; Right: classifier trained with real and synthetic data from both domains.

In this paper, we address unsupervised domain adaptation for image classification from the perspective of target domain data pseudo-labelling and generation. On one hand, we investigate the effectiveness of pseudo labelling techniques without any explicit source and target data distribution alignment. Pseudo labelling techniques have been employed in prior work [59, 5, 58] but its effectiveness has been underestimated. Our experiments demonstrate surprisingly strong classification performance on UDA benchmark datasets with a simple classifier (i.e. a linear two-layer Multi-Layer Perceptron for image features or a Convolutional Neural Network for digit images) trained on labelled source data and pseudo-labelled target data. Besides, we propose a novel L_2-normalisation regularised Autoencoder (i.e. *norm-AE*) to generate synthetic labelled target samples for training the classifier. The proposed *norm-AE* is characterized by $L2$-normalized parameters (i.e. mean and variance) of latent code distribution as the substitute of the KL-Divergence regularisation in the vanilla VAE. With this data augmentation strategy, the performance of UDA can be enhanced as illustrated in Figure 1.

The contributions of our work can be summarised as follows:

– we demonstrate that a specially designed pseudo-labelling strategy can achieve surprisingly strong performance on commonly used benchmark datasets for unsupervised domain adaptation; the performance is even comparable with or better than many more complex methods on Digits (97.2%), Office-Caltech (92.8%) and ImageCLEF-DA (89.4%).

- we demonstrate the proposed pseudo-labelling strategy is superior to those in [5] and [58] within our proposed framework.
- we propose a generative model adapted from VAE to further improve the performance of unsupervised domain adaptation by generating synthetic features for the target domain; the average accuracy is improved by 0.6%, 1.7%, 1% and 2.9% on Office-Caltech, Office31, ImageCLEF-DA and Office-Home datasets respectively.
- we present a thorough set of comparative experiments and ablation studies to demonstrate the proposed methods can achieve competitive performance on several benchmark datasets (i.e. Digits, Office-Caltech, Office31, ImageCLEF-DA and Office-Home).

2. Related Work

In this section, we review existing work related to ours. We first review existing approaches to UDA problems which fall into two main categories: *feature transformation approaches* [34, 35, 48, 65, 14, 49, 55] and *deep feature learning approaches* [12, 32, 13, 36, 37, 4, 41, 66, 33]. Subsequently, we discuss the use of pseudo-labelling and data augmentation techniques in UDA for image classification.

2.1. Unsupervised Domain Adaptation

Feature transformation approaches aim to transform the source domain and/or target domain features such that transformed source and target domain data can be aligned. As such the classifier learned from labelled source data can be directly applied to target data. Usually, linear transformations are used by learning the projection matrices with different optimization objectives and a kernel trick can help to explore the non-linear relations between source and target domain data if necessary. The most commonly employed objective for unsupervised domain adaptation is to align data distributions in source and target domains [34, 35]. For this purpose, Maximum Mean Discrepancy (MMD) based distribution matching has been used to reduce differences of the marginal distributions [35], conditional distributions or both [34, 55]. Correlation alignment (CORAL) [48] transforms source domain features to minimize domain shift by aligning the second-order statistics of source and target distributions. Manifold Embedded Distribution Alignment (MEDA) [55] learns a domain-invariant classifier based on the transformed features where the transformation aims to align both the marginal and conditional distributions with quantitative account for their relative importance.

In contrast to the above-mentioned approaches that learn one feature transformation matrix for either source domain or both domains, Joint Geometrical and Statistical Alignment (JGSA) [65] learns two coupled projections that project the source and target domain data into a joint subspace where the geometrical and distribution shifts are reduced simultaneously. Apart from the distribution alignment, recent feature transformation based approaches also promote the discriminative properties in the transformed features. Scatter Component Analysis (SCA) [14] aims to learn a feature transformation such that the transformed data from different domains have similar scattering and the labelled data are well separated. A Linear Discriminant Analysis (LDA) framework was proposed in [38] by learning class-specific projections. Similarly, Li et al. [30] proposed an approach to feature transformation towards Domain Invariant and Class Discriminative (DICD) features.

Deep feature learning approaches to domain adaptation were inspired by the success of deep Convolutional Neural Networks (CNN) in visual recognition [29]. Attempts have been made to take advantage of the powerful representation learning capability of CNN combined with a variety of feature learning objectives. Most deep feature learning approaches aim to learn domain-invariant features from raw image data in source and target domains in an end-to-end framework. Specifically, the objectives of feature transformation approaches have been incorporated in the deep learning models. To learn the domain-invariant features through a deep CNN, the gradient reversal layer was proposed in [12] and used in other deep feature learning approaches [13, 41, 66] as well. The gradient reversal layer connects the feature extraction layers and the domain classifier layers. During backpropagation, the gradients of this layer multiply a certain negative constant to ensure the feature distributions over two domains are made similar (as indistinguishable as possible for the domain classifier). Deep Adaptation Networks (DAN) [32] and Residual Transfer Network (RTN) [36] aim to learn transferable features from two domains by matching the domain distributions of multiple hidden layer features based on MMD. Deep CORAL [50] integrates the idea of CORAL [48] into a deep CNN framework to learn features with favoured properties (i.e. aligned correlations over source and target distributions for multiple layer activations). These approaches only consider the alignment of marginal distributions and cannot ensure the separability of target data. Deep Reconstruction Classification Network (DRCN) [15] trains a feature learning model using labelled source data and unlabelled target data in the supervised and unsupervised learning manners respectively. More recently, the prevalent Generative Adversarial Network (GAN) loss has been employed in Adversarial Discriminative Domain Adaption (ADDA) [51] with promising results.

2.2. UDA With Pseudo-Labelling

To address the issue of lack of labelled data in the target domain, pseudo-labelling has been used by many existing approaches. Pseudo-labels are assigned to unlabelled samples in the target domain by a classifier. Hard labelling assigns a pseudo-label $\hat{y}$ to each unlabelled sample without considering the confidence [34, 65, 55]. The pseudo-labelled target samples together with labelled source samples are used to learn an improved classifier. By repeating these two steps, the classifier and accuracy of pseudo-labels can be improved gradually. Hard pseudo-labelling relies heavily on good initialisation otherwise it is likely to be stuck in local optima. To address this issue, soft labelling was employed in [41]. Instead of assigning a hard label to a sample, soft labelling assigns the probability of belonging to each class to a sample. In the Multi-Adversarial Domain Adaptation (MADA) approach [41], the soft pseudo-label of a target sample is used to determine how much this sample should be attended to different class-specific domain discriminators.

Selective pseudo-labelling is the other way to alleviate the mislabelling issue [66, 59, 5]. Similar to the soft labelling strategy, selective pseudo-labelling also takes into consideration the confidence in target sample labelling but a different manner. Selective pseudo-labelling picks up a subset of target samples and assigns them with pseudo labels with high confidence to avoid potential mislabelling. The idea is that at the beginning the classifier is weak so that only a small fraction of the target samples can be correctly classified. When the classifier gets stronger after each iteration of learning, more target samples can be correctly classified hence should be pseudo-labelled and participate in the learning process. An easy-to-hard strategy was employed in [5]. Target samples whose similarity scores are higher than a threshold are selected for pseudo-labelling and this threshold is updated after each iteration of learning so that more unlabelled target samples can be selected. A class-wise sample selection strategy was proposed in [59, 58]. Samples are selected for each class independently so that pseudo-labelled target samples will contribute to the alignment of conditional distribution for each class during learning. In this paper, we propose a novel pseudo-label selection strategy that is superior to those used in [58] within the proposed framework.

2.3. UDA With Data Augmentation

Data augmentation has drawn attention in existing works for UDA. For example, Hsu et al. [23] proposed a novel augmentation-based method to generate labelled data with a similar distribution to the target domain for robust speech recognition. A vanilla VAE was trained in an unsupervised way to learn a disentangled latent representation of speech which can be modified for generating expected target domain data. However, the disentangled image attributes in the latent space are a challenging goal to achieve. Instead, we employ a conditional AutoEncoder (AE) and the domain information can be incorporated and fed into the decoder for target domain sample generation. Volpi et al. [54] performed data augmentation in the feature space by devising a feature generator trained with a Conditional Generative Adversarial Network (CGAN). Our approach is similar to this in the sense of feature augmentation whilst we aim to augment data by feature transformation across domains rather than from random noises. Huang et al. [24] proposed GAN based models for image-to-image translation and evaluated the performance in object detection rather than image classification which is our focus in this work. Lv et al. [39] also utilised GAN to generate target domain data given class labels to improve the classifier training. Following these studies, in our work, a novel *norm-AE* is proposed to generate target domain samples by feature transformation across domains and its effectiveness is demonstrated through comparative experiments.

Variational Autoencoder (VAE) has been a prevalent generative model for data generation and it has been used for UDA in literature [23, 22, 62, 25, 64, 6]. Hou et al. [22] aim to generate synthetic target-domain data with VAEs trained domain-wisely. Subsequently, the higher-level and lower-level layers of the decoders for source and target domains are cross-stacked to form new VAEs which can be used to transform images from one domain to the other. However, the effectiveness of the idea was only validated on digits data in [22] and is questionable for more complicated image classification tasks. In contrast to pixel-level image generation, a more reliable alternative is employed in our work which aims to generate image features with a simplified VAE model. Wang et al.[62] also used VAE in the feature space for speech signal representation learning. However, their work focused on the latent code vectors z generated by the encoder of VAE whilst our goal is to generate synthetic features in the original feature space. We also investigated the effect of latent code vectors in our preliminary experiments but did not observe favourable performance enhancement in the image classification tasks. Chen et al. [6] utilized two-stream Wasserstein Autoencoders to map the data from four domains (i.e. real source, real target, synthetic source and synthetic target) into a common subspace towards better classification performance. By contrast, our work also concern data from these four domains whilst the classification is carried out in the original feature space without the need of learning a latent space.

3. Problem Formulation

Before presenting our method, we describe the standard problem formulation of UDA for image classification. Given a labelled dataset $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}, i = 1, 2, ..., n_s$ from the source domain $\mathcal{S}$, $\boldsymbol{x}_i^s \in \mathbb{R}^{d^x}$ represents the feature vector of i-th labelled sample in the source domain, d^x is the feature dimension and $y_i^s \in \mathcal{Y}^s$ denotes the corresponding label. UDA aims to classify an unlabelled data set $\mathcal{D}^t = \{\boldsymbol{x}_i^t\}, i = 1, 2, ..., n_t$ from the target domain $\mathcal{T}$, where $\boldsymbol{x}_i^t \in \mathbb{R}^{d^x}$ represents the feature vector in the target domain. The target label space $\mathcal{Y}^t$ is equal to the source label space $\mathcal{Y}^s$. It is assumed that both the labelled source domain data $\mathcal{D}^s$ and the unlabelled target domain data $\mathcal{D}^t$ are available for model learning. As a result, most existing UDA approaches are evaluated in the transductive learning setting. Cases of inductive learning settings where evaluation on new target data that are not accessed during training are also considered in the literature [51, 33, 5, 43]. Our proposed methods apply to both settings.

4. Proposed Method

In this section, we first present a computationally simple approach to UDA for classification problems. The approach is based on the hypothesis *a unified classifier for both source and target domains can be trained in the original homogeneous feature space despite the domain* *shift across domains by supervised learning*. Pseudo-labelled target domain data are combined with labelled source domain data to train the unified classifier for both source and target domains. Subsequently, we describe our proposed generative model *norm-AE* which is used to generate synthetic features to augment the training data for classifier training.

4.1. Revisiting Selective Pseudo-Labelling

We aim to learn a unified classifier $y = f(\boldsymbol{x})$ for both source and target domains. The classifier $f(\boldsymbol{x})$ can be implemented as a shallow CNN model for image classification when the input $\boldsymbol{x}$ are raw images or a linear two-layer Multi-Layer Perceptron (MLP, containing an input layer and an output layer) when the input $\boldsymbol{x}$ are image features. As the first step, we train the classifier with labelled source domain data. The trained classifier is subsequently used to classify unlabelled target domain samples and get their pseudo labels $\hat{y}_i^t, i = 1, 2, ..., n_t$. The confidence score $s(\hat{y}_i^t)$ of the pseudo label $\hat{y}_i^t$ can also be obtained from the softmax layer of the classifier. The pseudo-labelled target domain samples are combined with the labelled source domain samples to re-train the classifier so that the classifier can gain the capability of separating target domain samples. The updated classifier is again used to update the pseudo-labels of target domain samples. This process can be repeated for multiple iterations towards an optimal classifier and better classification performance.

One key to the above pseudo-labelling strategy is the selection of pseudo-labelled target domain samples for training in each iteration. Instead of using all the pseudo-labelled target samples for classifier training, it has been proved that progressively selecting a fraction of the target domain samples for training is beneficial [5, 58]. Following the previous works in [59] and [58], we select pseudo-labelled target samples with top confidence scores class-wisely and add them to the training data set in each iteration. Specifically, we consider the pseudo-labels class-wisely and select top-K confident pseudo-labelled target domain samples for each class.

Distinct from existing selective pseudo-labelling in [59] and [58], the number of selected pseudo-labelled target domain samples $N(c,k)$ for c-th class in k-th iteration is determined as follows:

$$N(c,k) = \min\{\frac{k}{T}\frac{n_t}{C}, \hat{n}_t(c,k)\} \tag{1}$$

where T is the number of iterations empirically set as 10 in our experiments; n_t is the number of target domain samples; C is the number of classes and $\hat{n}_t(c,k)$ denotes the number of target

Algorithm 1 The method of naive Selective Pseudo-Labelling (naive-SPL)

```
Input: Labelled source data set $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}, i = 1,2,...,n_s$ and unlabelled target data set $\mathcal{D}^t = \{\boldsymbol{x}_i^t\}, i = 1,2,...,n_t$, number of iteration $T$.
Output: A unified classifier $f(\boldsymbol{x})$ and predicted labels $\{\hat{y}^t\}$ for target domain samples.
initialise $k = 0$;
Training the classifier $f(\boldsymbol{x})$ using only source data $\mathcal{D}^s$;
Assign pseudo labels for all target data;
while $k < T$ do
    $k \leftarrow k + 1$;
    Select a subset of pseudo-labelled target data $\mathcal{S}_k \in \hat{\mathcal{D}}^t$ using Eq. (1);
    Re-training the classifier using $\mathcal{D}^s$ and $\mathcal{S}_k$;
    Update pseudo labels for all target data.
end while
```

domain samples predicted to be from c-th classes in k-th iteration. In contrast, $N(c,k)$ is set as $(k\hat{n}_t(c,k))/T$ in previous work [58]. That is, the number of selected pseudo-labelled samples $N(c,k)$ is proportional to the number of predicted pseudo-labels $\hat{n}_t(c,k)$ for a specific class. As a result, there can be a large number of selected pseudo-labelled samples for some classes whilst very limited pseudo-labelled samples for other classes. Our pseudo-label selection strategy indicated in Eq.(1) allows balanced pseudo-labelled target samples across different classes. This naive Selective Pseudo-Labelling (naive-SPL) approach is summarized in Algorithm 1.

4.2. Data Augmentation Using norm-AE

As opposed to the existing methods of UDA, our proposed *naive-SPL* does not aim to explicitly address the distribution discrepancy. Instead, it focuses on the issue of training data scarcity. Following this direction, we propose a novel norm-AE model to further address the training data scarcity issue in the target domain by generating synthetic target domain features from labelled source domain ones.

Our proposed generative model is inspired by conditional VAE (CVAE) [47] and is conditioned on domain labels rather than class labels. As illustrated in Figure 2, given an input sample $\boldsymbol{x}$ from the source or target domain, the encoder aims to learn a posterior distribution $q_\Phi(\boldsymbol{z}|\boldsymbol{x},d)$

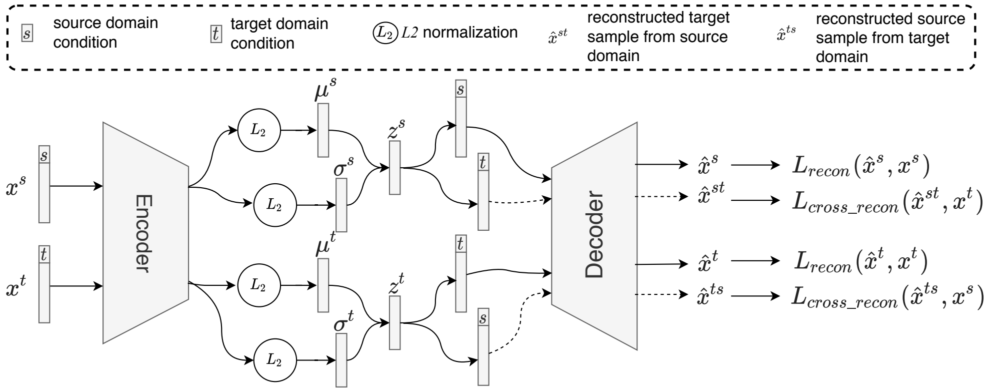

s source domain condition
t target domain condition
L_2 L2 normalization
$\hat{x}^{st}$ reconstructed target sample from source domain
$\hat{x}^{ts}$ reconstructed source sample from target domain
x^s
x^t
s
t
Encoder
L_2
μ^s
σ^s
μ^t
σ^t
z^s
z^t
Decoder
$\hat{x}^s \longrightarrow L_{recon}(\hat{x}^s, x^s)$
$\hat{x}^{st} \longrightarrow L_{cross\_recon}(\hat{x}^{st}, x^t)$
$\hat{x}^t \longrightarrow L_{recon}(\hat{x}^t, x^t)$
$\hat{x}^{ts} \longrightarrow L_{cross\_recon}(\hat{x}^{ts}, x^s)$

Figure 2: The diagram of norm-AE used for data augmentation. The encoder and decoder are conditioned on the domain label s or t. Given a source domain sample x^s as the input, the model generates reconstructed samples $\hat{x}^s$ and $\hat{x}^{st}$ in the source and target domains respectively. Similarly, the model can take a target domain sample x^t as the input and generates $\hat{x}^t$ and $\hat{x}^{ts}$.

from which the latent encoding vector $\boldsymbol{z}$ can be sampled and subsequently fed into the decoder to reconstruct the input feature $\hat{\boldsymbol{x}}$, where d denotes the domain label condition (i.e. $d \in \{s,t\}$). The decoder can be parameterized by $p_\theta(\boldsymbol{x}|\boldsymbol{z},d)$. As a result, the model is expected to generate synthetic target domain samples from those in the source domain and vice versa. To this end, we make some essential modifications to the traditional CVAE in two aspects: *replacing the Kullback-Leibler divergence regularization by L_2 normalization* and *training the model using paired source and target domain samples*.

In traditional CVAE, the loss function is composed of two components as follows:

$$\begin{aligned}\mathcal{L}_{CVAE}(\Phi,\theta;\boldsymbol{x}) =& \mathcal{L}_{recon}\big(\boldsymbol{x},\hat{\boldsymbol{x}}\big)\\ &+D_{KL}\big(\mathcal{N}(\mu_{\boldsymbol{x}},\sigma_{\boldsymbol{x}})||\mathcal{N}(\boldsymbol{0},\boldsymbol{I})\big)\end{aligned} \tag{2}$$

where the first terms represents the reconstruction error $\mathcal{L}_{recon}\big(\boldsymbol{x},\hat{\boldsymbol{x}}\big) = ||\boldsymbol{x}-\hat{\boldsymbol{x}}||_2^2$ and the second term is the KL-divergence between the learned posterior distribution and the standard Normal distribution. The KL-divergence is a regularization term forcing the learned latent codes $\boldsymbol{z}$ to follow the standard Normal distribution. This regularization enables the learned model to gain the capability of generating meaningful data from a random latent code $\boldsymbol{z}$ sampled from the standard Normal distribution.

One limitation of VAE is the approximation of the posterior to a Gaussian prior [7]. Although convenient, the Gaussian prior encourages points to cluster close to the origin. This is particularly problematic when the data are from multiple classes [7]. An ideal prior would only stimulate the variance of the posterior without forcing its mean to be close to the origin. For this purpose, we can simply remove the KL-divergence loss from Eq.(2) and the model degrades into an AutoEncoder with deterministic latent code (i.e. the variance tends to be zero for good reconstruction of $\boldsymbol{x}$). To promote the discriminative property of the learned latent code, we apply L_2 normalisation to the outputs of the encoder μ and $\log(\sigma^2)$.

Applying L_2 to the latent code of AutoEncoder has been proved to be beneficial to the clustering accuracy [1]. On the other hand, to avoid learning deterministic latent code, we also need to constrain the variance of the posterior. There exist various options for constraints on the variances. For example, we can force the variance close to 1 for each dimension in the latent space. We choose to apply L_2 normalisation to the log of variance vectors to allow for more flexibility. As a result, the variance of each dimension in the latent space is forced within the range $[1/e, e]$. In practice, however, there might be little difference between these two choices of constraints on variances as shown in our ablation study in Section 5.4.

To summarise, the encoder learns a posterior probability distribution $q_\Phi(\boldsymbol{z}|\boldsymbol{x}, d) = \mathcal{N}(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}})$. From the posterior distribution, we can sample a latent code $\boldsymbol{z}$ given a sample $\boldsymbol{x}$ and the decoder try to reconstruct the sample by learning the probability distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z}, d)$. During training, the objective of our generative model is to maximise the probability of the training data X [10]:

$$\log p(X) = E_{z\sim q}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z}, d)] \tag{3}$$

In practice, $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is chosen depending on the modeling of the input data but is often taken as a simple distribution (e.g., fixed variance Gaussian) [3]. In the case of fixed variance Gaussian, we have

$$\mathcal{L}_{recon} = ||\boldsymbol{x} - \mu_\theta(\boldsymbol{z})||_2^2 = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2^2 \tag{4}$$

To enable the capability of generating synthetic data across domains, we train the norm-AE in a novel way. Specifically, we use paired data $\{\boldsymbol{x}^s, \boldsymbol{x}^t\}$ from source and target domains that belong to the same class. The class information for unlabelled target domain data can be obtained by pseudo-labelling as described in the previous section. The paired data are fed into the norm-AE and a set of reconstructions are generated as $\{\hat{\boldsymbol{x}}^s, \hat{\boldsymbol{x}}^{st}, \hat{\boldsymbol{x}}^t, \hat{\boldsymbol{x}}^{ts}\}$ (c.f. Figure 2). The loss function is formulated as:

$$\begin{aligned}\mathcal{L}_{norm-AE}(\Phi,\theta;\boldsymbol{x}) = &\left(\mathcal{L}_{recon}(\boldsymbol{x^s},\hat{\boldsymbol{x}}^{\boldsymbol{s}}) + \mathcal{L}_{recon}(\boldsymbol{x^t},\hat{\boldsymbol{x}}^{\boldsymbol{t}})\right) \\ &+ \left(\mathcal{L}_{cross\_recon}(\boldsymbol{x^s},\hat{\boldsymbol{x}}^{ts}) + \mathcal{L}_{cross\_recon}(\boldsymbol{x^t},\hat{\boldsymbol{x}}^{st})\right)\end{aligned} \tag{5}$$

The first two terms measure the reconstruction errors for source and target domain samples respectively. The last two terms are cross-domain reconstruction errors, i.e., $\mathcal{L}_{cross\_recon}(\boldsymbol{x},\hat{\boldsymbol{x}}) = ||\boldsymbol{x}-\hat{\boldsymbol{x}}||_2^2$. Although the samples in the pair of $\{\boldsymbol{x^s},\hat{\boldsymbol{x}}^{ts}\}$ or $\{\boldsymbol{x^t},\hat{\boldsymbol{x}}^{st}\}$ are from the same class, they are not necessarily two views of the same image. To reduce the cross-domain reconstruction errors, the encoder has to preserve class information in the latent code space. As a result, the use of cross-domain reconstruction loss $\mathcal{L}_{cross\_recon}$ facilitates the model to generate class discriminative synthetic data across domains.

The norm-AE model is incorporated into the selective pseudo-labelling framework described in the previous section so that the classifier training can be enhanced by combining the real training data and synthetic data generated by norm-AE. The norm-AE is trained with labelled source domain data and pseudo-labelled target domain data in each iteration. The method of our proposed norm-AE-SPL is summarized in Algorithm 2 where the differences from Algorithm 1 are highlighted in **bold**.

4.3. Model Architectures and Computational Complexity

The computational cost of Algorithm 1 depends on the classifier training itself and the number of iterations T. In our experiments, we use a CNN architecture from [43] as the classifier which consists of two convolutional layers and three fully connected layers for digit classification. For image classification datasets, we use deep features (i.e. ResNet50) and a linear MLP consisting of only the input and output layers which are computationally efficient.

The method *norm-AE-SPL* in Algorithm 2 involves one additional step of training the *norm-AE* model. The encoder and decoder are implemented as 3-layer MLP (i.e. $d^x \rightarrow 512 \rightarrow d^z$ and $d^z \rightarrow 512 \rightarrow d^x$) with ReLU layers and a dropout rate of 0.5 applied on the intermediate activations. d^x and d^z are the dimensionality of input $\boldsymbol{x}$ and latent code $\boldsymbol{z}$ respectively. The value of d^z is set as 64 in our experiments.

5. Experiments and Results

In this section, we describe our experiments on commonly used datasets for unsupervised domain adaptation for image classification (i.e. Digits, Office-Caltech [16], Office31 [42],

Algorithm 2 The method of SPL with data augmentation by norm-AE (norm-AE-SPL)

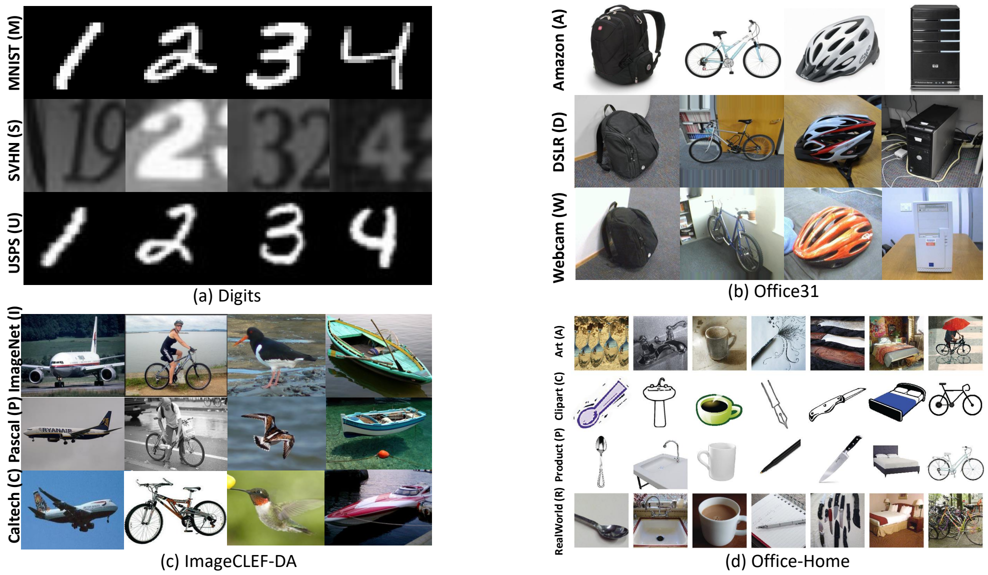
```
Input: Labelled source data set $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}, i = 1,2,...,n_s$ and unlabelled target data set
    $\mathcal{D}^t = \{\boldsymbol{x}_i^t\}, i = 1,2,...,n_t$, number of iteration $T$.
Output: A unified classifier $f(x)$ and predicted labels $\{\hat{y}^t\}$ for target domain samples.
initialise $k = 0$;
Training the classifier $f(x)$ using only source data $\mathcal{D}^s$;
Assign pseudo labels for all target data;
while $k < T$ do
    $k \leftarrow k + 1$;
    Select a subset of pseudo-labelled target data $\mathcal{S}_k \in \hat{\mathcal{D}}^t$ using Eq. (1);
    Training the norm-AE model using $\mathcal{D}^s$ and $\mathcal{S}_k$ by minimizing the loss in Eq.(5);
    Re-training the classifier using real data from $\mathcal{D}^s$ and $\mathcal{S}_k$, and their corresponding synthetic data generated by norm-AE;
    Update pseudo labels for all target data.
end while
```

ImageCLEF-DA [2] and Office-Home [53]). Our approach is firstly compared with state-of-the-art UDA approaches to evaluate its effectiveness. An ablation study is conducted to demonstrate the effects of different components and hyper-parameters in our approach. Finally, we investigate how different hyper-parameters affect performance.

5.1. Datasets

To make a thorough evaluation, we conduct experiments on five commonly used datasets including one digit classification dataset and four image classification datasets. Exemplar images from different domains are shown in Figure 3 for four datasets. The Office-Caltech dataset is not shown since it consists of the same 3 domains as those in Office31 and the Caltech domain in ImageCLEF-DA. More details of these datasets are described as follows.

Digit classification is a commonly used benchmark for unsupervised domain adaptation. We follow existing works [51, 33, 5, 43] and consider three domain adaptation tasks (i.e. MNIST → USPS, USPS → MNIST and MNIST → SVHN) on three digit datasets: MNIST, USPS and SVHN. There are 60,000/10,000 images for training/testing in MNIST, 7,291/2,007 in USPS,



(a) Digits

(b) Office31

(c) ImageCLEF-DA

(d) Office-Home

Figure 3: Exemplar images from different domains of four datasets used in our experiments (The Office-Caltech dataset consists of the same domains as Office31 and one additional Caltech domain; exemplar images for the Office-Home dataset (d) originate from [53]; best viewed in color).

and 73,257/26,032 in SVHN. In each dataset, there are 10 classes of digit 0–9.

Office-Caltech [16] consists of four domains: Amazon (A, images downloaded from online merchants), Webcam (W, low-resolution images by a web camera), DSLR (D, high-resolution images by a digital SLR camera) and Caltech-256 (C). Ten common classes from all four domains are used: backpack, bike, calculator, headphone, computer-keyboard, laptop-101, computer-monitor, computer-mouse, coffee-mug, and video-projector. There are 2533 images in total with 8 to 151 images per category per domain.

Office31 [42] consists of three domains: Amazon (A), Webcam (W) and DSLR (D). There are 31 common classes for all three domains containing 4,110 images in total.

ImageCLEF-DA [2] consists of four domains. We follow the existing works [67] using three of them in our experiments: Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P). There are 12 classes and 50 images for each class in each domain.

Office-Home [53] is another dataset recently released for evaluation of domain adaptation algorithms. It consists of four different domains: Artistic images (A), Clipart (C), Product images (P) and Real-World images (R). There are 65 object classes in each domain with a total number

Table 1: Classification Accuracy (%) of UDA on Digits dataset (M: MNIST, U: USPS, S: SVHN).

Method	M→ U	U → M	M → S	Average
ADDA [51]	89.4	90.1	76.0	85.2
GTA [44]	95.3	90.8	92.4	92.8
MCD [43]	96.5	94.1	96.2	95.6
MCD+CAT [9]	96.3	95.2	97.1	96.3
rRevGrad+CAT [9]	94.0	96.0	98.8	96.3
CTSN [70]	96.1	97.3	-	-
CAN [68]	95.8	94.6	-	-
SHOT [31]	**98.0**	**98.4**	**98.9**	**98.4**
Baseline (w/o selection)	30.1	51.3	83.1	54.8
naive-SPL* (overall selection)	88.2	91.7	90.7	90.2
naive-SPL (Ours)	95.8	97.7	98.0	97.2
norm-AE-SPL (Ours)	95.8	97.7	98.0	97.2

of 15,588 images.

5.2. Experimental Setting

The algorithm is implemented in PyTorch[1]. For digit classification, we use the same CNN model designed by [43]. In each domain adaptation task, the labelled training data from the source domain and the unlabelled training data from the target domain are used to train the classifier which is subsequently evaluated on the test data from the target domain. As a result, the evaluation on this dataset is done in an inductive learning setting. For the Office-Caltech dataset, we use deep features Decaf6 [11] (activations of the $6th$ fully connected layer of a convolutional neural network trained on ImageNet, $d = 4096$) which were commonly used in existing works for

[1] Code is available: https://github.com/hellowangqian/UDA-norm-AE

Table 2: Classification Accuracy (%) on Office-Caltech dataset using Decaf6 features. Each column displays the results of a pair of source → target setting.

Method	C→A	C→W	C→D	A→C	A→W	A→D	W→C	W→A	W→D	D→C	D→A	D→W	Average
DDC[52]	91.9	85.4	88.8	85.0	86.1	89.0	78.0	84.9	**100.0**	81.1	89.5	98.2	88.2
DAN[32]	92.0	90.6	89.3	84.1	91.8	91.7	81.2	92.1	**100.0**	80.3	90.0	98.5	90.1
DCORAL[50]	92.4	91.1	91.4	84.7	-	-	79.3	-	-	82.8	-	-	-
CORAL[49]	92.0	80.0	84.7	83.2	74.6	84.1	75.5	81.2	**100.0**	76.8	85.5	99.3	84.7
SCA[14]	89.5	85.4	87.9	78.8	75.9	85.4	74.8	86.1	**100.0**	78.1	90.0	98.6	85.9
JGSA[65]	91.4	86.8	93.6	84.9	81.0	88.5	85.0	90.7	**100.0**	86.2	92.0	99.7	90.0
MEDA[55]	93.4	95.6	91.1	87.4	88.1	88.1	**93.2**	**99.4**	99.4	87.5	93.2	97.6	92.8
CAPLS [59]	90.8	85.4	95.5	86.1	87.1	**94.9**	88.2	92.3	**100.0**	88.8	93.0	**100.0**	91.8
SPL [58]	92.7	93.2	**98.7**	87.4	95.3	89.2	87.0	92.0	**100.0**	88.6	92.9	98.6	93.0
Han et al. [19]	90.8	87.5	89.8	87.4	81.0	86.6	85.0	91.3	99.4	85.8	90.4	99.0	89.5
DS-c [20]	92.5	81.0	89.8	85.3	81.7	87.3	81.4	78.0	97.5	85.0	90.6	99.0	87.4
Baseline (w/o selection)	91.8	80.5	87.5	84.7	76.1	84.3	74.1	78.3	**100.0**	77.3	85.3	98.3	84.9
naive-SPL* (overall selection)	92.6	89.2	95.4	87.5	89.5	93.0	87.9	91.9	**100.0**	**89.1**	92.9	99.3	92.4
naive-SPL (Ours)	**94.1**	92.9	88.5	86.9	95.6	91.3	87.5	94.1	**100.0**	88.7	**94.1**	99.3	92.8
norm-AE-SPL (Ours)	94.0	**97.6**	90.8	**88.1**	**97.3**	92.0	88.4	93.0	99.4	87.9	93.0	99.3	**93.4**

a fair comparison with the state of the arts. For the other three datasets, ResNet50 [21] features ($d = 2048$) are used in our experiments to allow a direct comparison with other methods.

5.3. Comparison with State-of-the-Art Approaches

We compare our approaches with the most competitive methods including those based on deep features (extracted using deep models such as ResNet50 pre-trained on ImageNet) and deep learning models using pre-trained ResNet50 as the backbones. The classification accuracy of our approaches and the comparative ones are shown in Tables 1-5 in terms of each combination of "source" → "target" domains and the average accuracy over all different combinations. The classification accuracy is calculated as the number of correctly predicted samples over the total number of test samples (i.e. per-image accuracy). For all experiments in this section, each task is repeated five times with random seeds set as 0-4 to calculate the mean accuracy for this task. We use **bold** and underlined fonts to indicate the best and the second-best results respectively in each column of the tables.

Our approaches without and with data augmentation are denoted as ***naive-SPL*** and ***norm-AE-SPL***, respectively. Besides, we conduct an ablation study to investigate the effect of different pseudo-label selection strategies. For this purpose, we consider two more related methods in our experiments. One is denoted as ***Baseline (w/o selection)*** which uses all pseudo-labelled target

Table 3: Classification Accuracy (%) on Office31 dataset using either ResNet50 features or ResNet50 based deep models.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
RTN[36]	84.5	96.8	99.4	77.5	66.2	64.8	81.6
MADA[41]	90.0	97.4	99.6	87.8	70.3	66.4	85.2
MEDA[55]	86.2	97.2	99.4	85.3	72.4	74.0	85.7
GTA [45]	89.5	97.9	99.8	87.7	72.8	71.4	86.5
iCAN[66]	92.5	98.8	**100.0**	90.1	72.1	69.9	87.2
CDAN-E[33]	94.1	98.6	**100.0**	92.9	71.0	69.3	87.7
JDDA[4]	82.6	95.2	99.7	79.8	57.4	66.7	80.2
SymNets[67]	90.8	98.8	**100.0**	**93.9**	74.6	72.5	88.4
TADA [63]	**94.3**	98.7	99.8	91.6	72.9	73.0	88.4
CAPLS [59]	90.6	98.6	99.6	88.6	75.4	76.3	88.2
SPL [58]	92.7	98.7	99.8	93.0	**76.4**	**76.8**	**89.6**
CTSN [70]	90.6	98.6	99.9	89.3	73.7	74.1	81.9
Han et al. [19]	77.0	92.1	95.8	81.1	62.7	63.6	78.7
DS-c [20]	71.6	95.7	99.6	76.9	67.8	67.3	79.8
Baseline (w/o selection)	73.3	97.5	99.6	75.8	68.0	67.6	80.3
naive-SPL* (overall selection)	84.6	**98.9**	99.8	81.4	70.4	70.6	84.3
naive-SPL (Ours)	88.6	98.1	99.9	82.0	73.6	73.4	85.9
norm-AE-SPL (Ours)	88.6	98.7	97.1	93.0	73.8	74.2	87.6

domain samples without selection for classifier training. The other dubbed as ***naive-SPL* (overall selection)*** is adapted from our proposed ***naive-SPL*** by replacing the pseudo-label selection strategy in Eq.(1) with that used in [59]. This pseudo label selection strategy selects the most confident pseudo-labelled target samples without considering the balance across different classes hence may lead to sub-optimal self-training performance.

Table 1 shows the performance of different UDA approaches on the Digits dataset. The classifier is implemented by a shallow CNN model (c.f. Section 4.3) and is trained on raw images. Our proposed *naive-SPL* achieves an average accuracy of 97.2% over three commonly

Table 4: Classification Accuracy (%) on ImageCLEF-DA dataset using either ResNet50 features or ResNet50 based deep models.

Method	I→P	P→I	I→C	C→I	C→P	P→C	Avg
RTN[36]	75.6	86.8	95.3	86.9	72.7	92.2	84.9
MADA[41]	75.0	87.9	96.0	88.8	75.2	92.2	85.8
iCAN[66]	79.5	89.7	94.7	89.9	78.5	92.0	87.4
CDAN-E[33]	77.7	90.7	**97.7**	91.3	74.2	94.3	87.7
SymNets[67]	80.2	93.6	97.0	93.4	78.7	**96.4**	89.9
MEDA[55]	79.7	92.5	95.7	92.2	78.5	95.5	89.0
SPL [58]	78.3	**94.5**	96.7	**95.7**	**80.5**	96.3	90.3
Han et al. [19]	76.8	80.8	93.2	89.8	72.8	85.3	83.1
DS-c [20]	78.7	86.7	92.8	87.3	70.4	91.3	84.5
CAN [68]	78.5	91.8	95.5	91.6	76.4	95.2	88.2
Baseline (w/o selection)	79.1	89.2	94.0	86.0	69.9	92.4	85.1
naive-SPL* (overall selection)	77.0	90.7	95.9	92.1	73.4	93.7	87.1
naive-SPL (Ours)	80.0	91.5	96.2	94.3	79.1	95.6	89.4
norm-AE-SPL (Ours)	**80.3**	93.9	96.9	94.6	80.4	96.3	**90.4**

Table 5: Classification Accuracy (%) on Office-Home dataset using either ResNet50 features or ResNet50 based deep models.

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Average
JAN[37]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN-E [33]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
MEDA[55]	54.6	75.2	77.0	56.5	72.8	72.3	59.0	51.9	78.2	67.7	57.2	81.8	67.0
SymNets [67]	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	**74.5**	52.6	82.7	67.6
TADA [63]	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	**60.0**	82.9	67.6
CAPLS [59]	**56.2**	**78.3**	80.2	**66.0**	75.4	78.4	**66.4**	**53.2**	81.1	71.6	56.1	84.3	70.6
SPL [58]	54.5	77.8	**81.9**	65.1	**78.0**	**81.1**	66.0	53.1	**82.8**	69.9	55.3	**86.0**	**71.0**
CAN [68]	50.4	69.8	75.9	58.7	70.4	69.8	57.8	47.6	76.0	70.2	54.5	79.9	65.2
Baseline (w/o selection)	43.1	65.1	73.7	50.7	64.4	64.6	53.8	41.6	73.8	62.7	44.3	77.4	59.6
naive-SPL* (overall selection)	44.1	72.5	77.7	52.0	70.0	70.5	52.4	39.4	77.3	62.3	44.3	79.4	61.8
naive-SPL (Ours)	52.0	74.2	79.1	56.1	74.4	74.1	56.8	49.0	78.1	61.4	52.4	80.5	65.7
norm-AE-SPL (Ours)	51.6	76.0	80.6	63.0	77.0	78.4	62.9	50.7	81.2	66.3	52.8	82.9	68.6

used domain adaptation tasks which is better than all the comparative methods except SHOT [31] with an average accuracy of 98.4%. The method *norm-AE-SPL* in this experiment is based on the features extracted by the classifier trained for *naive-SPL*. As we can see, there is no performance improvement when data augmentation by norm-AE is employed in this case. This is because our selective pseudo-labelling strategy described in Section 4.1 enables the CNN model to learn domain-invariant features from source and target domains. As a result, the domain shift between the source- and target- domain features used to train the *norm-AE* model is negligible hence the synthetic features generated by *norm-AE* can not provide additional information for classifier training.

In many real-world applications, however, much deeper and more complicated CNN models than the one used for digit classification are required to extract image features. Training large CNN models on both source and target domain data can be computationally expensive and unnecessary [58]. In the following experiments on real-world image datasets, we use pre-trained (on ImageNet only) ResNet50 as the feature extractor to extract features of source and target domain images. As shown in Figure 5, a data distribution shift can be observed between source and target domain data. In these cases, our proposed approach including the data augmentation model *norm-AE* demonstrates its effectiveness in improving the classification accuracy as described in the following paragraphs.

Tables 2-5 demonstrate the results on image classification datasets. Our proposed *naive-SPL* can already achieve quite good performance with the average accuracy of 92.8% on Office-Caltech, 85.9% on Office31, 89.4% on ImageCLEF-DA and 65.7% on Office-Home. These results are comparable with many more complex UDA approaches, especially on Office-Caltech and ImageCLEF-DA datasets. This validates the effectiveness of our proposed selective pseudo-labelling strategy since *naive-SPL* is no more than a simple classifier trained with labelled source and pseudo-labelled target domain samples iteratively. With the use of our proposed data augmentation method, *norm-AE-SPL* improves the performance consistently on all four image classification datasets. As a result, our proposed *norm-AE-SPL* achieves the best average accuracy of 93.4% and 90.4% on Office-Caltech and ImageCLEF-DA datasets, respectively. On the other two datasets, *norm-AE-SPL* also performs comparably well with most approaches except CAPLS [59] and SPL [58]. It is noteworthy that both of them employ the dimensionality reduction algorithm Locality Preserving Projection (LPP) to learn a latent subspace where source- and target-domain data can be well aligned. These methods require to solve the eigenvalue problems and the computational cost is subject to the number of samples in both domains. As a result, they are not suitable for large-scale applications whilst our proposed method does not have such constraints. Besides, our method is intrinsically different from those based on the domain alignment in that we assume a unified classifier can be learned in the original homogeneous feature space despite the existence of the domain shift across the source and target domains.

We conduct an ablation study by comparing the performance of *Baseline (w/o selection)*, *naive-SPL* (overall selection)* and *naive-SPL* and some consistent conclusions can be drawn from this ablation study. Firstly, the *Baseline* method using all pseudo-labelled target-domain data without selection is always inferior to the other two methods with selective pseudo-labelling. Specifically, the performance gaps between the *Baseline (w/o selection)* and *naive-SPL* (overall selection)* methods in terms of the average classification accuracy are 7.5%, 4.0%, 2.0% and 2.2% on Office-Caltech, Office31, ImageCLEF-DA and Office-Home datasets respectively. These results demonstrate that selecting the most confident pseudo-labels progressively is of vital importance to classifier training. On the other hand, the pseudo-label selection strategy used in [58] is inferior to the proposed alternative described in Section 4.1 as *naive-SPL* outperforms *naive-SPL* (overall selection)* by margins of 0.4%, 1.6%, 2.3% and 3.9% on four image classification datasets respectively.

5.4. Ablation studies on model architecture

We conduct additional ablation studies to validate the effectiveness of the proposed norm-AE architecture. To this end, we investigate following variants of our proposed method. They share the architecture (i.e. an encoder-decoder architecture with cross-domain and within-domain reconstruction flows) but differ in the forms of latent code regularisation. They also employ the same selective pseudo labelling strategy as the proposed *norm-AE-SPL* does.

- **AE**: a vanilla AutoEncoder is employed together with the selective pseudo labelling. The latent code output by the encoder μ_x (concatenated with the domain label $d \in \{s,t\}$) is directly fed into the decoder for the reconstructed input $\hat{\boldsymbol{x}}$.

- **AE w/ L2-norm**: inspired by [1], we apply L_2 normalisation to the latent code of the AutoEncoder in this method and keep other settings the same as *AE*.

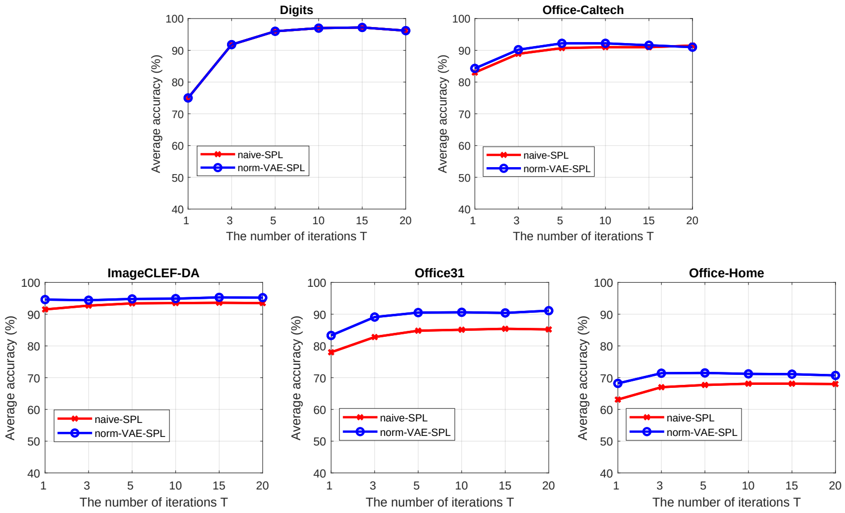


Figure 4: Effect of the number of iterations T.

- **AE w/ noises**: we add random noise to the latent code of the AutoEncoder and keep other settings the same as *AE*.
- **VAE**: a VAE is employed together with the selective pseudo labelling. Particularly, we use the same flow to generate synthetic data as our proposed *norm-AE*, i.e., the latent code $\boldsymbol{z}$ is sampled from a Normal distribution $\mathcal{N}(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}})$ for any given input $\boldsymbol{x}$. The difference between this *VAE* method and *norm-AE* lies in how the posterior distribution $\mathcal{N}(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}})$ is constrained during training. Note that we do not sample $\boldsymbol{z}$ from a standard normal distribution for data generation since the data generated in this way has no labels for subsequent supervised classifier training.
- **mean-Norm-AE**: this is a variant of *norm-AE* with the L_2 normalisation applied only to the mean $\mu_{\boldsymbol{x}}$ and the variances $\sigma_{\boldsymbol{x}}$ are not constrained during training.
- **mean-Norm-AE w/ variance loss**: this is a variant of *norm-AE* with the mean $\mu_{\boldsymbol{x}}$ L_2 normalised and the variances $\sigma_{\boldsymbol{x}}$ forced to be close to ones.

The results of the ablation studies are shown in Table 6. In general, all the investigated variants of our *norm-AE* method can achieve reasonably good performance thanks to the selective

Table 6: Results of the ablation study on the *norm-AE* architecture.

Method	L_2 mean	L_2 $\log(\sigma^2)$	z sampling	others	Office-Caltech	Office31	Image-CLEF	Office-Home
AE	✘	✘	✘	-	92.6 ± 0.3	87.2 ± 0.2	89.9 ± 0.1	67.4 ± 0.1
AE w/ L2-norm	✔	✘	✘	-	92.6 ± 0.2	87.4 ± 0.3	89.9 ± 0.2	67.5 ± 0.1
AE w/ noises	✘	✘	✘	noised z	92.2 ± 0.1	84.2 ± 0.2	87.1 ± 0.1	62.6 ± 0.1
VAE	✘	✘	✔	KLD loss	93.0 ± 0.5	87.3 ± 0.1	90.4 ± 0.1	68.7 ± 0.1
mean-Norm-AE	✔	✘	✔	-	92.6 ± 0.2	86.7 ± 0.2	89.8 ± 0.1	68.6 ± 0.1
mean-Norm-AE w/ variance loss	✔	✘	✔	$\log(\sigma^2)$ loss	92.8 ± 0.2	87.7 ± 0.3	90.4 ± 0.1	68.7 ± 0.1
norm-AE	✔	✔	✔	-	93.4 ± 0.1	87.6 ± 0.1	90.4 ± 0.1	68.6 ± 0.1

pseudo-labeling strategy. By a closer look at the results in Table 6, we can draw some interesting conclusions. Firstly, *AE* and its variants (i.e. *AE w/ L2-norm* and *AE w/ noises*) perform consistently worse than our *norm-AE*. It is likely to be that *AE* learns deterministic latent codes, hence the cross-domain reconstruction is restricted. In contrast, our *norm-AE* learns the posterior distribution for the latent codes and has higher capacity of modeling domain invariant latent codes for each class. Secondly, by comparing *VAE* with *norm-AE*, we can see that *norm-AE* performs slightly better on two out of four datasets. As we have discussed in Section 4.2, *VAE* has the limitation of forcing latent codes of all classes close to the origin whilst our *norm-AE* projects them onto the sphere for better separability. Finally, we can see that applying regularisation on the variances of the learned posterior distribution is necessary. Without regularisation on the variances, the method *mean-Norm-AE* performs worse than *norm-AE* on three out of four datasets. This is because the model tends to learn near zero variance for good reconstruction and degrades into a vanilla *AE*. Besides, an alternative regularisation term (e.g., $\log(\sigma^2) \to 0$) can lead to comparably good performance with *norm-AE* which applies the L_2 normalisation on variances.

5.5. *Effects of the Hyper-parameter T*

In Algorithms 1 and 2, the number of iterations T is a hyper-parameter which was set as 10 throughout our main experiments. In this experiment, we investigate how the value of T affects the performance of *naive-SPL* and *norm-AE-SPL*. To this end, we set the value of T to be 1, 3, 5, 10, 15, 20 respectively and calculate the average accuracy over some representative domain adaptation tasks. Specifically, we consider all three tasks for Digits, three tasks $C \to A/D/W$ for Office-Caltech, two tasks $A \to W/D$ for Office31, two tasks $P \to I/C$ for ImageCLEF-DA and three tasks $A \to C/P/R$ for Office-Home. Each task is repeated three times with random seeds

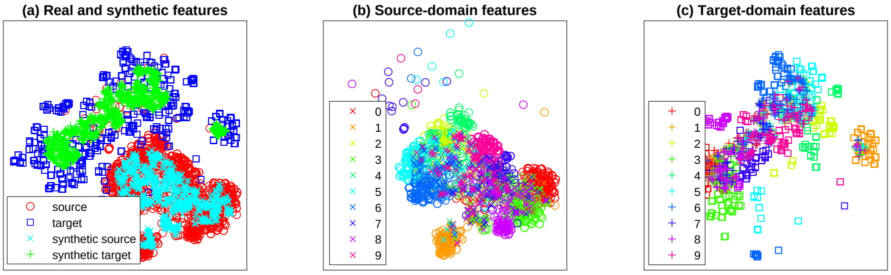


Figure 5: Visualization of real and synthetic features using t-SNE (best viewed in color). (a) data distribution of four domains (i.e. real source, real target, synthetic source, synthetic target); (b) real and synthetic data distribution in the source domain (colours represent different classes); (c) real and synthetic data distribution in the target domain (colours represent different classes).

set as 0, 1 and 2.

The results are shown in Figure 4 in which the average accuracy over considered tasks are reported for five datasets. As we can see, the number of iterations T has a negligible effect on the performance when it is greater than 5 for both *naive-SPL* and *norm-AE-SPL*. For the Digits dataset, significant performance improvement can be observed when T increases from 1 to 5 whilst for other image classification datasets, the optimal value of T varies from 1, 3 to 5 with subtle differences. To summarize, our approaches are not sensitive to hyper-parameters and perform well enough with a relatively small number of iterations.

5.6. Data Visualization

For qualitative evaluation, we use the t-SNE technique [40] to visualize the real and synthetic features in Figure 5. The domain adaptation task $C \to W$ in the Office-Caltech dataset is taken as an exemplar. The 4096-dimensional features of real and synthetic data from both domains are mapped into 2-dimensional projections in an unsupervised way by preserving data distributions [40].

Firstly, we visualize real data points from the source (*red circles*) and target (*blue squares*) domains in Figure 5(a). It is clear data from source and target domains are distributed in different regions. In the same plot, we also visualize the synthetic features generated by our proposed model for the source (*cyan crosses*) and target (*green* +) domains. We can see the synthetic data points generated for the source/target domain are well aligned with the real data points in the corresponding domain thanks to the domain conditions of the decoder in our norm-AE model.

Secondly, we examine the class discriminative property of synthetic data in the source and target domains in Figure 5(b) and (c) respectively. In the source domain, we use *circles* and *crosses* to represent the real and synthetic data points respectively whilst different colours are used for ten classes. Similarly, *squares* and *crosses* are used for real and synthetic data points and colours represent different classes in the target domain. We can see that real data points from the same class are distributed in a cluster thanks to the discriminative features extracted by deep CNN models pre-trained on ImageNet. The synthetic data generated by our proposed model are also distributed in clusters of different classes. This demonstrates our proposed method can generate synthetic data which are both domain and class discriminative.

Finally, a closer inspection of Figure 5(b) and (c) also tells us that the synthetic data clusters are not perfectly aligned with their corresponding clusters of real data (i.e. circles/squares and crosses of the same colour are not well aligned). Such misalignment is more severe in the target domain due to the fact there is no labelled data in this domain. We believe slight misalignment leads to over-complete data distribution [27] and is beneficial to learning a more robust classifier. However, significant distribution shifts can hurt the performance. This demonstrates the limitation of our proposed method in generating reliable class-discriminative synthetic data and leads us to improve the model in our future work.

6. Conclusion

In this paper, we proposed novel approaches to the unsupervised domain adaptation problem from a novel perspective and achieved impressive experimental results with the average classification accuracy of 97.2%, 93.4%, 87.6%, 90.4% and 68.6% on Digits, Office-Caltech, Office31, ImageCLEF-DA and Office-Home datasets, respectively. Instead of pursuing explicit domain alignment, we train a unified classifier for both source and target domain data in a high-dimensional feature space despite the existence of distribution discrepancy across domains. We proposed a novel pseudo-label selection strategy outperforming the existing ones in the literature [5, 59, 58]. With this specially designed pseudo-labelling strategy, our method *naive-SPL* can achieve strong performance which is impressive given that it only uses a typical shallow CNN for digit classification and a linear two-layer MLP for image classification. Moreover, our proposed *norm-AE-SPL* can improve the performance by generating synthetic features for training data augmentation. To conclude, our work provides fresh insights into unsupervised domain adaptation for the community.

References

[1] Aytekin, C., Ni, X., Cricri, F., and Aksu, E. (2018). Clustering and unsupervised anomaly detection with l2 normalized deep auto-encoder representations. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE.

[2] Caputo, B., Müller, H., Martinez-Gomez, J., Villegas, M., Acar, B., Patricia, N., Marvasti, N., Üsküdarlı, S., Paredes, R., Cazorla, M., et al. (2014). Imageclef 2014: Overview and analysis of the results. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 192–211. Springer.

[3] Chadebec, C. and Allassonnière, S. (2022). A geometric perspective on variational autoencoders. *arXiv preprint arXiv:2209.07370*.

[4] Chen, C., Chen, Z., Jiang, B., and Jin, X. (2019a). Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *AAAI Conference on Artificial Intelligence*.

[5] Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., Xu, T., and Huang, J. (2019b). Progressive feature alignment for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 627–636.

[6] Chen, Z., Chen, C., Jin, X., Liu, Y., and Cheng, Z. (2019c). Deep joint two-stream wasserstein auto-encoder and selective attention alignment for unsupervised domain adaptation. *Neural Computing and Applications*, pages 1–14.

[7] Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*.

[8] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE.

[9] Deng, Z., Luo, Y., and Zhu, J. (2019). Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9944–9953.

[10] Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

[11] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, pages 647–655.

[12] Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189.

[13] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

[14] Ghifary, M., Balduzzi, D., Kleijn, W. B., and Zhang, M. (2016a). Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1414–1430.

[15] Ghifary, M., Kleijn, W. B., Zhang, M., Balduzzi, D., and Li, W. (2016b). Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer.

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

[17] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[18] Gorban, A. N., Makarov, V. A., and Tyukin, I. Y. (2020). High-dimensional brain in a high-dimensional world: Blessing of dimensionality. *Entropy*, 22(1):82.

[19] Han, C., Lei, Y., Xie, Y., Zhou, D., and Gong, M. (2020). Visual domain adaptation based on modified a- distance and sparse filtering. *Pattern Recognition*, 104:107254.

[20] Han, C., Lei, Y., Xie, Y., Zhou, D., and Gong, M. (2021). Learning smooth representations with generalized softmax for unsupervised domain adaptation. *Information Sciences*, 544:415–426.

[21] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, pages 770–778.

[22] Hou, J., Ding, X., Deng, J. D., and Cranefield, S. (2019). Unsupervised domain adaptation using deep networks with cross-grafted stacks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0.

[23] Hsu, W.-N., Zhang, Y., and Glass, J. (2017). Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 16–23. IEEE.

[24] Huang, S.-W., Lin, C.-T., Chen, S.-P., Wu, Y.-Y., Hsu, P.-H., and Lai, S.-H. (2018). Auggan: Cross domain adaptation with gan-based data augmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 718–731.

[25] Ilse, M., Tomczak, J. M., Louizos, C., and Welling, M. (2020). Diva: Domain invariant variational autoencoders. In *Medical Imaging with Deep Learning*, pages 322–348. PMLR.

[26] Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. (2019). Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902.

[27] Keshari, R., Singh, R., and Vatsa, M. (2020). Generalized zero-shot learning via over-complete distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13300–13308.

[28] Kim, Y., Cho, D., Han, K., Panda, P., and Hong, S. (2021). Domain adaptation without source data. *IEEE Transactions on Artificial Intelligence*.

[29] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

[30] Li, S., Song, S., Huang, G., Ding, Z., and Wu, C. (2018). Domain invariant and class discriminative feature learning for visual domain adaptation. *IEEE Transactions on Image Processing*, 27(9):4260–4273.

[31] Liang, J., Hu, D., and Feng, J. (2020). Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of International Conference on Machine Learning*.

[32] Long, M., Cao, Y., Wang, J., and Jordan, M. I. (2015). Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105. JMLR. org.

[33] Long, M., Cao, Z., Wang, J., and Jordan, M. I. (2018). Conditional adversarial domain adaptation. In *Advances in*

Neural Information Processing Systems, pages 1647–1657.

[34] Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. S. (2013). Transfer feature learning with joint distribution adaptation. In *International Conference on Computer Vision*, pages 2200–2207.

[35] Long, M., Wang, J., Ding, G., Sun, j., and Yu, P. S. (2014). Transfer joint matching for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition*, pages 1410–1417.

[36] Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2016). Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144.

[37] Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2017). Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, pages 2208–2217.

[38] Lu, H., Shen, C., Cao, Z., Xiao, Y., and van den Hengel, A. (2018). An embarrassingly simple approach to visual domain adaptation. *IEEE Transactions on Image Processing*, 27(7):3403–3417.

[39] Lv, F., Zhu, J., Yang, G., and Duan, L. (2019). Targan: Generating target data with class labels for unsupervised domain adaptation. *Knowledge-Based Systems*, 172:123–129.

[40] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

[41] Pei, Z., Cao, Z., Long, M., and Wang, J. (2018). Multi-adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence*.

[42] Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). Adapting visual category models to new domains. In *European Conference on Computer Vision*, pages 213–226. Springer.

[43] Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. (2018). Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732.

[44] Sankaranarayanan, S., Balaji, Y., Castillo, C. D., and Chellappa, R. (2018a). Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512.

[45] Sankaranarayanan, S., Balaji, Y., Castillo, C. D., and Chellappa, R. (2018b). Generate to adapt: Aligning domains using generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[46] Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., and Akata, Z. (2019). Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8247–8255.

[47] Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Proceedings of the Advances in neural information processing systems*, pages 3483–3491.

[48] Sun, B., Feng, J., and Saenko, K. (2016). Return of frustratingly easy domain adaptation. In *AAAI Conference on Artificial Intelligence*, volume 6, page 8.

[49] Sun, B., Feng, J., and Saenko, K. (2017). Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, pages 153–171.

[50] Sun, B. and Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer.

[51] Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. In *IEEE*

Conference on Computer Vision and Pattern Recognition, volume 1, page 4.

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

[53] Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027.

[54] Volpi, R., Morerio, P., Savarese, S., and Murino, V. (2018). Adversarial feature augmentation for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5495–5504.

[55] Wang, J., Feng, W., Chen, Y., Yu, H., Huang, M., and Yu, P. S. (2018). Visual domain adaptation with manifold embedded distribution alignment. In *ACM Multimedia Conference on Multimedia Conference*, pages 402–410. ACM.

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

[57] Wang, Q. and Breckon, T. P. (2020a). Generalized zero-shot domain adaptation via coupled conditional variational autoencoders. *arXiv preprint arXiv:2008.01214*.

[58] Wang, Q. and Breckon, T. P. (2020b). Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. In *AAAI Conference on Artificial Intelligence*.

[59] Wang, Q., Bu, P., and Breckon, T. P. (2019a). Unifying unsupervised domain adaptation and zero-shot visual recognition. In *International Joint Conference on Neural Networks*.

[60] Wang, Q. and Chen, K. (2017). Zero-shot visual recognition via bidirectional latent embedding. *International Journal of Computer Vision*, 124(3):356–383.

[61] Wang, Q. and Chen, K. (2020). Multi-label zero-shot human action recognition via joint latent ranking embedding. *Neural Networks*, 122:1–23.

[62] Wang, X., Li, L., and Wang, D. (2019b). Vae-based domain adaptation for speaker verification. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 535–539. IEEE.

[63] Wang, X., Li, L., Ye, W., Long, M., and Wang, J. (2019c). Transferable attention for domain adaptation. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[64] Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., and Zhang, W. (2020). Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6502–6509.

[65] Zhang, J., Li, W., and Ogunbona, P. (2017). Joint geometrical and statistical alignment for visual domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5150–5158. IEEE.

[66] Zhang, W., Ouyang, W., Li, W., and Xu, D. (2018). Collaborative and adversarial network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3801–3809.

[67] Zhang, Y., Tang, H., Jia, K., and Tan, M. (2019). Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5031–5040.

[68] Zhou, Q., Wang, S., et al. (2021). Cluster adaptation networks for unsupervised domain adaptation. *Image and Vision Computing*, 108:104137.

[69] Zhu, X. and Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130.

[70] Zuo, L., Jing, M., Li, J., Zhu, L., Lu, K., and Yang, Y. (2021). Challenging tough samples in unsupervised domain adaptation. *Pattern Recognition*, 110:107540.