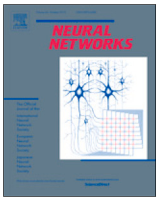
Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Generalized zero-shot domain adaptation via coupled conditional variational autoencoders

Qian Wang [a,*], Toby P. Breckon [a,b]

[a] Department of Computer Science, Durham University, UK
[b] Department of Engineering, Durham University, UK

ARTICLE INFO

Article history:
Received 15 May 2022
Received in revised form 31 January 2023
Accepted 22 March 2023
Available online 28 March 2023

Keywords:
Generalized zero-shot learning
Domain adaptation
Generalized zero-shot domain adaptation
Conditional variational autoencoder

ABSTRACT

Domain adaptation aims to exploit useful information from the source domain where annotated training data are easier to obtain to address a learning problem in the target domain where only limited or even no annotated data are available. In classification problems, domain adaptation has been studied under the assumption all classes are available in the target domain regardless of the annotations. However, a common situation where only a subset of classes in the target domain are available has not attracted much attention. In this paper, we formulate this particular domain adaptation problem within a generalized zero-shot learning framework by treating the labelled source-domain samples as semantic representations for zero-shot learning. For this novel problem, neither conventional domain adaptation approaches nor zero-shot learning algorithms directly apply. To solve this problem, we present a novel Coupled Conditional Variational Autoencoder (CCVAE) which can generate synthetic target-domain image features for unseen classes from real images in the source domain. Extensive experiments have been conducted on three domain adaptation datasets including a bespoke X-ray security checkpoint dataset to simulate a real-world application in aviation security. The results demonstrate the effectiveness of our proposed approach both against established benchmarks and in terms of real-world applicability.

1. Introduction

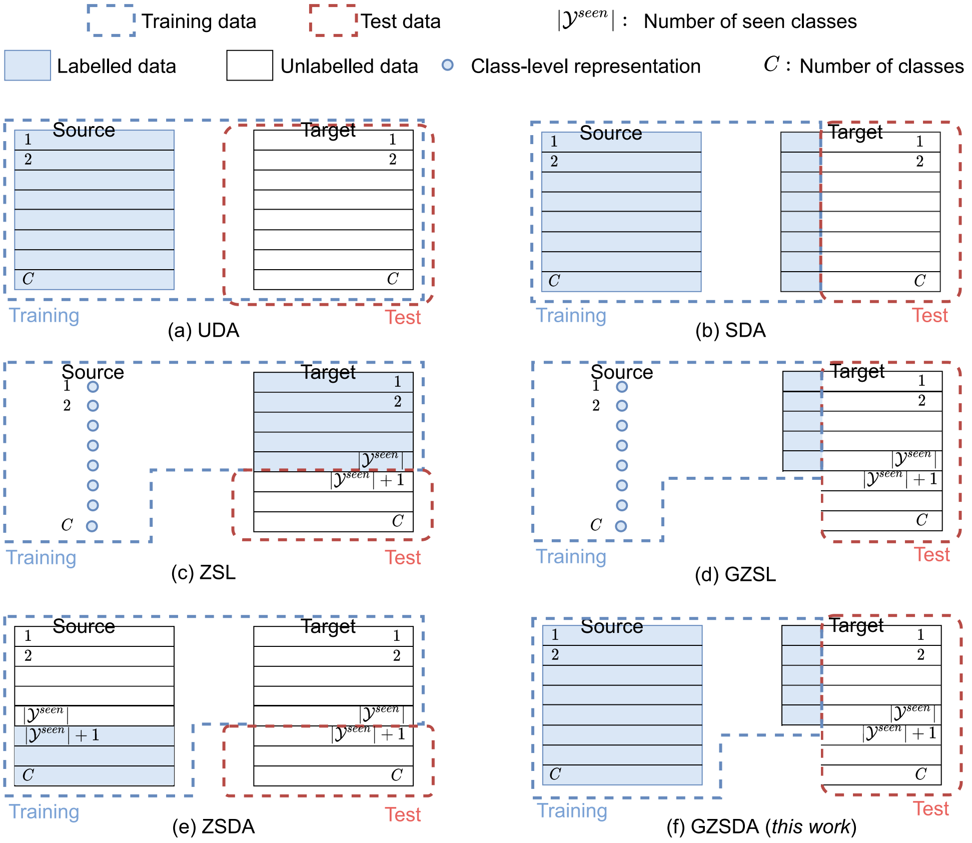
The success of deep learning in the recent decade relies on the availability of abundant annotated data for training (Deng et al., 2009). In real-world applications, the acquisition of sufficient training data can be difficult or even impossible. One technique to address the training data sparsity issue is transfer learning which aims to explore and transfer knowledge learned from the source domain to the target domain Tan et al. (2018). There are usually more annotated data in the source domain than those in the target domain within which the task to solve resides. *Zero-shot learning* (Guo & Guo, 2020; Pourpanah et al., 2022; Wang & Chen, 2017a, 2017b; Xian, Sharma, Schiele, & Akata, 2019) and *domain adaptation* (Deng et al., 2021; Ma, Zhang, & Xu, 2019; Wang & Breckon, 2020; Wang, Bu, & Breckon, 2019) are two well-formulated transfer learning problems that have attracted much attention in the recent decade.

Traditional supervised learning methods have the limitation in that they can only recognize *seen classes* (observed) for which labelled samples are available during training. By contrast, zero-shot learning (ZSL, Fig. 1c) aims to recognize samples from novel unseen classes (unobserved) for which no training samples are available during training (Ji, Wang, et al., 2021; Ji, Yan, Wang, Pang, & Li, 2021; Ji, Yu, Yu, Pang, & Zhang, 2021; Wang & Chen, 2017b, 2020; Xian et al., 2019). To this end, side information of both seen and unseen classes from a *source domain* (as opposed to the *target domain* where the recognition task resides) is needed to model the between-class relations. In zero-shot visual recognition, class-level semantic representations (e.g., attributes or word vectors) are usually adopted as the side information in the source domain (i.e. semantic representation space) whilst the image classification task is addressed in the target domain (i.e. visual representation space) (Wang & Chen, 2017a). In domain adaptation problems (Fig. 1 a–b), we have plenty of labelled data in the source domain but limited or even no **labelled** data in the target domain.

The problem definitions of zero-shot learning and domain adaptation can be unified into one framework as shown in Fig. 1. By exploring the capabilities of zero-shot learning and domain adaptation, an emerging type of problem within the same framework, zero-shot domain adaptation (Fig. 1e), can be addressed and has been studied in Peng, Wu, and Ernst (2018), Wang, Cheng, and Jiang (2021), Wang and Jiang (2019, 2021). ZSDA assumes

[*] Correspondence to: Department of Computer Science, Durham Univeristy, Durham, UK.
E-mail addresses: qian.wang173@hotmail.com (Q. Wang), toby.breckon@durham.ac.uk (T.P. Breckon).

https://doi.org/10.1016/j.neunet.2023.03.033
0893-6080/



(a) UDA

(b) SDA

(c) ZSL

(d) GZSL

(e) ZSDA

(f) GZSDA (*this work*)

Fig. 1. A comparison of generalized zero-shot domain adaptation (GZSDA) problem (f) with related ones including: (a) Unsupervised Domain Adaptation (UDA); (b) Supervised Domain Adaptation (SDA); (c) Zero-Shot Learning (ZSL); (d) Generalized ZSL (GZSL); (e) Zero-Shot Domain Adaptation (ZSDA). C is the number of classes and $|\mathcal{Y}^{seen}|$ is the number of seen classes. As opposed to ZSDA, the GZSDA problem aims to classify all classes including both seen and unseen ones whilst ZSDA only focuses on the classification of unseen classes; the GZSDA problem also does not require the paired training data across two domains for the seen classes which is a limitation of ZSDA for practical use.

that there are unlabelled but **paired** source–target samples for $|\mathcal{Y}^{seen}|$ classes (i.e. irrelevant classes in some literature) and plenty of labelled samples in the source domain for unseen classes $|\mathcal{Y}^{seen}|+1,\ldots,C$ (i.e. relevant classes in some literature) during training. However, these studies aim at classifying unseen classes only and assume the test samples are only from unseen classes. As agreed in the zero-shot learning literature, generalized zero-shot learning (Fig. 1d), in which the recognition of both seen and unseen classes in the target domain is required, is more practically useful (Xian et al., 2019). In the same spirit, we take one step further in this paper to address a novel Generalized Zero-Shot Domain Adaptation (GZSDA, Fig. 1f) problem arising from many real-world applications. As illustrated in Fig. 1f, GZSDA is a variant of GZSL by replacing the class-level representations in the source domain with labelled sample-level data. Compared with ZSDA, GZSDA does not require paired source–target data for training but labelled samples for all classes in the source domain and labelled samples for seen classes in the target domain. As a result, the formulated GZSDA problem is essentially different from ZSDA which is an unsupervised learning problem (i.e. no labelled data in the target domain) whilst GZSDA requires supervision from seen classes in the target domain.

ZSL and GZSL face the challenge of data imbalance across seen and unseen classes (Ji, Yu, et al., 2021). The learned model tends to overfit data belonging to seen classes and hence performs unsatisfactorily for unseen classes. This challenge is even more significant for GZSL (Pourpanah et al., 2022) since both seen and unseen classes need to be classified. Domain adaptation problems including ZSDA face the challenge of data imbalance across source and target domains (Kouw & Loog, 2019). The learned model tends to overfit data from the source domain and degrades the performance on the target domain. Typical Unsupervised Domain Adaptation (UDA) (Wang & Breckon, 2020) approaches usually fight off this challenge by taking advantage of the unlabelled target-domain data for feature alignment or pseudo-labelling. However, the target-domain data to be classified (i.e. data from unseen classes) are not available in ZSDA which poses a more significant challenge. As a composition of GZSL and domain adaptation, the GZSDA problem faces challenges from both, i.e., the learned models bias to seen classes (Ji, Yu, et al., 2021; Kumar Verma, Arora, Mishra, & Rai, 2018) and the source domain (Kouw & Loog, 2019), due to training data imbalance across classes and domains.

To address the data imbalance issue in the GZSDA problem, we present a novel Coupled Conditional Variational Autoencoder (CCVAE) solution by generating unseen data in the target domain to re-balance the training data. Specifically, the proposed CCVAE can transform source-domain samples into their associated projections within the target domain without loss of class information and vice versa. As a result, target-domain samples of unseen classes can be generated from the corresponding source-domain samples. Subsequently, the generated target-domain samples for unseen classes together with real training data can be used to train a classifier for all classes in a supervised learning manner. The CCVAE works in the feature space rather than the image pixel space to reduce the complexity and challenge of image generation since the goal of GZSDA is image classification rather than image generation. Following this outline, the contributions of this paper can be summarized as follows:

- a novel Coupled Conditional Variational Autoencoder (CCVAE) model is proposed to address the GZSDA problem extending and outperforming the prior work of Wang et al. (2019); the proposed CCVAE integrates the benefits of feature transformation and feature generation in one framework.
- a new multi-domain dataset arising from real-world applications is collected, annotated and released for domain adaptation research; it comprises of cross-spectral image domains (i.e. dual-energy colour-mapped X-ray and regular colour photograph) which are not present in other datasets.
- extended experimentation is performed on three benchmark datasets in addition to a bespoke X-ray security checkpoint dataset to validate the effectiveness of the proposed CCVAE in GZSDA problems both against established benchmarks and in terms of real-world applicability, and also its superiority to a variety of contemporary methods in the field.

2. Related work

We review closely related work to our study from the perspective of zero-shot learning, domain adaptation and zero-shot domain adaptation and summarize the relationship to existing research topics and approaches in Fig. 1 and Table 1.

2.1. Domain adaptation

Domain adaptation aims to effectively transfer knowledge learned from the source domain to the target domain and has been applied in weakly supervised image classification problems (Kim & Kim, 2021; Wang & Breckon, 2020; Wang et al., 2019). Existing domain adaptation approaches (Wang & Breckon, 2020; Wang et al., 2019) try to align the marginal distributions across the source and target domains (Wang & Breckon, 2020) or to learn domain-invariant representations (Pei, Cao, Long, & Wang, 2018) so that labelling information available in the source domain can be explored to guide the learning of a classifier in the target domain or a latent common space. However, aligning the marginal distributions is not sufficient for distinguishing different classes in the target domain. Fine-grained class-wise adaptation across domains has been employed by promoting the alignment of conditional distributions as an additional constraint (Long, Cao, Wang, & Jordan, 2018). This class-wise adaptation is feasible for supervised domain adaptation where labelled samples for all classes in the target domain are available. For unsupervised domain adaptation, this can be implemented by pseudo-labelling (Chen et al., 2019; Wang & Breckon, 2020) given access to unlabelled target-domain samples for all classes. However, in the scenario of zero-shot domain adaptation, class-wise adaptation forms the primary challenge that we address in this work due to the lack of samples for unseen classes in the target domain regardless of labelled or unlabelled.

2.2. Zero-shot learning

Zero-shot learning in visual recognition has been extensively studied in literature (Mishra et al., 2018; Wang & Chen, 2017b; Xian et al., 2019). The most popular approaches to zero-shot learning are based on a generative model such as Generative Adversarial Networks (GAN) (Xian et al., 2019) and Variational Autoencoders (VAE) (Mishra et al., 2018). The generative models are trained to generate image features for specific classes given the corresponding class-level semantic representations (i.e. attributes or word vectors). Subsequently, a classifier can be trained using the combined real and generated data covering both seen and unseen classes. Although recent advances in zero-shot learning have achieved impressive performance in several benchmark image classification datasets (Xian et al., 2019), an intrinsic drawback arising from the semantic gap between the source (semantic) and target (visual) domains has been overlooked. One intrinsic limitation of ZSL is that the class-level attributes or word vectors in the source domain restrict the capability of representing the intra-class variability. As a result, the quality of class-level semantic representations plays a significant role in the success of zero-shot learning (Wang & Chen, 2017b). Attempts have been made to improve the class-level semantic representations so that the semantic gap can be mitigated fundamentally (Wang & Chen, 2017a). Alternatively, the class-level semantic representations in zero-shot learning can be replaced by more informative labelled samples in a source domain where such labelled samples are easy to collect and annotate. This leads to the very novel zero-shot learning problem we focus on in this paper. Existing zero-shot learning methods (Mishra et al., 2018; Xian et al., 2019) cannot be directly applied to this problem since the source-domain information appears in a different modality, whilst the ideas of generating synthetic image features for unseen classes will be employed and extended in our approach (Section 3).

2.3. Zero-shot domain adaptation

Very limited prior work has addressed zero-shot domain adaptation problems. According to our definition in Fig. 1, the problems addressed in Blitzer, Foster, and Kakade (2009), Ishii et al. (2019), Kumagai and Iwata (2018) should be categorized as unsupervised domain adaptation though the papers were entitled as zero-shot domain adaptation. Yang and Hospedales (2015) attempted to address the issue where multiple source domains and the target domain are determined by a vector of continuous variables. Here there is no data available for the target domain but the corresponding control variables are known as prior knowledge. The transfer learning across the source and target domains can be explicitly modelled by such control variables. Similar assumptions are made by Ishii et al. (2019) which assumes that prior knowledge of attribute information exists (e.g., time, angle, gender, age, etc.) characterizing the difference between source and target domains. By contrast, we aim to address a more generic problem without the need for these control variables relating to the source and target domains. The problem we try to address in this work is also related to that in Peng et al. (2018), Wang et al. (2021), Wang and Jiang (2019, 2021) which however restrict the recognition to unseen classes (Fig. 1e). Moreover, paired task-irrelevant data (i.e. seen class data in this context) from source and target domains are required during training in Jhoo and Heo (2021), Peng et al. (2018), Wang et al. (2021), Wang and Jiang (2021) whilst such correspondences may not be available in most real cases. We lift these restrictions and focus on the GZSDA problem without the need for either control variables or paired training samples.

During the preparation of this manuscript, Li, Fang, and Chen (2022) propose a target unseen class prototype learning method to address the GZSDA problem following our definition in Fig. 1. Our proposed method in this paper employ a generative model to directly generate missing data in the target domain for classifier learning.

3. Method

In this section, we first describe the problem settings of Generalized Zero-Shot Domain Adaptation and subsequently our proposed solution to this problem.

Table 1
A comparison of generalized zero-shot domain adaptation with other related research topics.

Research problem	Training		Testing
	Source	Target	Target
Unsupervised Domain Adaptation (UDA, Wang & Breckon, 2020; Wang et al., 2019)	labelled samples for all classes	unlabelled samples for all classes (same as testing)	all classes
Supervised Domain Adaptation (SDA, Motiian, Piccirilli, Adjeroh, & Doretto, 2017)	labelled samples for all classes	labelled samples for all classes (a small number)	all classes
Zero-Shot Learning (ZSL, Wang & Chen, 2017b)	per-class representations for all classes	labelled samples for seen classes	unseen classes
Generalized ZSL (GZSL, Mishra, Krishna Reddy, Mittal, & Murthy, 2018)	per-class representations for all classes	labelled samples for seen classes	all classes
Zero-Shot Domain Adaptation (ZSDA, Ishii, Takenouchi, & Sugiyama, 2019; Peng et al., 2018; Wang & Jiang, 2019)	paired samples for seen classes and labelled samples for unseen classes	paired samples for seen classes	unseen classes
Generalized Zero-Shot Domain Adaptation (GZSDA, this paper)	labelled samples for all classes	labelled samples for seen classes	all classes

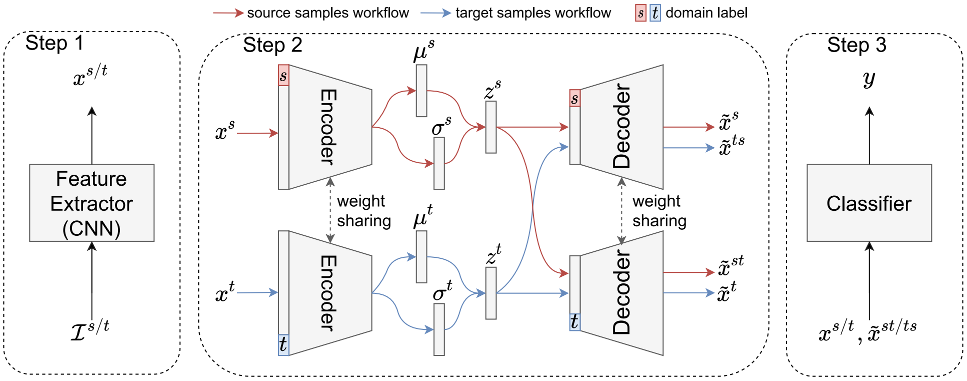


Fig. 2. Our proposed Coupled Conditional Variational Autoencoder (CCVAE) framework.

Our proposed framework consists of three steps as illustrated Fig. 2 (full details in Section 3.3). Step one trains a feature extractor using all the labelled training data from both domains. In the second step, a Coupled Conditional Variational Autoencoder is trained using image features extracted in step one and will be used to generate synthetic features in the target domain. With the combination of these synthetic features and features extracted from real training images, a classifier is trained and used for image classification in the target domain.

3.1. Problem formulation of generalized zero-shot domain adaptation

Given a labelled dataset $\mathcal{D}^s = \{(x_i^s, y_i^s)\}, i = 1, 2, \ldots, n^s$ from the source domain $\mathcal{S}$, x_i^s represents the ith training sample (e.g., an image in our case) in the source domain, and $y_i^s \in \mathcal{Y} = \{1, 2, \ldots, C\}$ denotes the corresponding label, and C is the number classes. In the target domain, a labelled dataset $\mathcal{D}^t = \{(x_i^t, y_i^t)\}, i = 1, 2, \ldots, n^t$ from the target domain $\mathcal{T}$. x_i^t and $y_i^t \in \mathcal{Y}^{seen}$ are the ith labelled sample and its label respectively. Note that $\mathcal{Y}^{seen} \subset \mathcal{Y}$, that is, labelled samples are available for only a subset of classes in the target domain. The label space $\mathcal{Y} = \mathcal{Y}^{seen} \cup \mathcal{Y}^{unseen}$ is shared by source and target domains. The task is to classify any given new instance x from the target domain by learning an inference model $y = f(x) \in \mathcal{Y}$.

3.2. Feature extraction

The key to our approach to the GZSDA problem is the generation of synthetic data for unseen classes in the target domain. Given the challenge of image generation in the pixel space (Xian et al., 2019), we choose to generate image features since the ultimate goal is image classification rather than image generation. To this end, we extract image features in the first step. As shown in Fig. 2, a shared deep Convolutional Neural Network (CNN) model is employed to extract features for images from both source and target domains. We use ResNet50 (He, Zhang, Ren, & Sun, 2016) pre-trained on the ImageNet (Deng et al., 2009) as the feature extractor for object images in our experiments and AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) trained from scratch using $\mathcal{D}^s$ and $\mathcal{D}^t$ for digits data. We will use $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ to denote the real and synthetically generated image features in the following sections.

3.3. Coupled conditional variational autoencoder

Variational Autoencoder The Variational Autoencoder (Kingma & Welling, 2013) encodes an input feature $\boldsymbol{x}$ into a distribution $p_\theta(\boldsymbol{z})$ (approximated by $q_\Phi(\boldsymbol{z}|\boldsymbol{x})$) from which the latent encoding vector $\boldsymbol{z}$ can be sampled and subsequently fed into the decoder to reconstruct the input feature $\tilde{\boldsymbol{x}}$. The decoder can be parameterized by $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. According to Kingma and Welling (2013), the objective function for the VAE can be written as follows:

$$\mathcal{J}_{VAE}(\Phi, \theta; \boldsymbol{x}) = -D_{KL}(q_{\Phi}(\boldsymbol{z}|\boldsymbol{x})||p_{\theta}(\boldsymbol{z})) + \mathbb{E}_{q_{\Phi}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\theta}(\boldsymbol{x}|\boldsymbol{z})] \tag{1}$$

where $D_{KL}(p||q)$ is the Kullback–Leibler (KL) divergence between two distributions p and q. The VAE is trained by maximizing $\mathcal{J}_{VAE}(\Phi, \theta; \boldsymbol{x})$ which can be interpreted as minimizing the reconstruction error and the KL divergence.

Conditional VAE Conditional VAE (CVAE) was first proposed in Sohn, Lee, and Yan (2015). It allows for modelling multiple modes (e.g., classes) in conditional distribution of the target variable (e.g., reconstructed input $\tilde{\boldsymbol{x}}$) given input $\boldsymbol{x}$ and the condition c. The objective function of CVAE can be adapted from Eq. (1) as follows:

$$\mathcal{J}_{CVAE}(\Phi, \theta; \boldsymbol{x}, c) = -D_{KL}(q_{\Phi}(\boldsymbol{z}|\boldsymbol{x}, c)||p_{\theta}(\boldsymbol{z})) + \mathbb{E}_{q_{\Phi}(\boldsymbol{z}|\boldsymbol{x}, c)}[\log p_{\theta}(\boldsymbol{x}|\boldsymbol{z}, c)] \tag{2}$$

In existing CVAE models (Mishra et al., 2018; Yan, Yang, Sohn, & Lee, 2016), both the encoder $q_{\Phi}(\boldsymbol{z}|\boldsymbol{x}, c)$ and the decoder $p_{\theta}(\boldsymbol{x}|\boldsymbol{z}, c)$ are conditioned on the class information (i.e. the condition c represents class-wise attributes in this case). In ZSL problem (Mishra et al., 2018), the CVAE is trained using target-domain data in the condition of class-wise attributes from the source domain. Each training sample in the target domain has its corresponding attribute vector as the condition in the CVAE model. However, in our GZSDA problem information from the source domain is represented by labelled samples rather than class-level representations. Although such labelled samples (e.g. their extracted features) can be aggregated into class-level representations to enable the application of conventional ZSL methods, the feature aggregation suffers from significant information loss as demonstrated in our experiments. In addition, the cross-domain correspondence in the sample level is unavailable hence the conventional CVAE and other ZSL methods do not apply to this problem.

By contrast, our CVAE is conditioned on the domain label (i.e. the condition c denotes the domain label) so that the decoder can generate features for a specified domain given a sampled $\boldsymbol{z}$ from the distribution $q_{\Phi}(\boldsymbol{z}|\boldsymbol{x}, c)$ and the domain label as the condition.

Coupled Conditional VAE The challenge of the GZSDA problem originates from the missing labelled samples for unseen classes in the target domain. We attempt to learn a generative model based on CVAE to generate synthetic features for unseen classes in the target domain. The generated features are required to be both class discriminative and domain discriminative. To these ends, the decoder $p(\boldsymbol{x}|\boldsymbol{z}, c)$ in the CVAE is conditioned on domain labels to generate domain discriminative features whilst the latent codes $\boldsymbol{z}$ need to be class discriminative to generate class discriminative features.

The proposed CCVAE is illustrated in Fig. 2 (step 2). It is composed of a pair of CVAE for the source and target domains respectively. In our work, we model both the encoders and decoders using fully connected neural networks. We force two CVAE to have identical architectures with shared weights. As a result, the model degenerates into one coupled CVAE trained on both source and target-domain data.

During training, the encoder takes the concatenation of a feature vector $\boldsymbol{x}^{\boldsymbol{s}}/\boldsymbol{x}^{\boldsymbol{t}}$ from the source/target domain and its corresponding domain label $c(\boldsymbol{x}) = s/t$ (represented by a one-hot 2-dimensional vector) as the input to estimate the latent code distribution $q(\boldsymbol{z}|\boldsymbol{x}, c) = \mathcal{N}(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}})$. $\mu_{\boldsymbol{x}}$ and $\Sigma_{\boldsymbol{x}}$ are the outputs of the encoder given the input $\boldsymbol{x}$. Subsequently, a latent code $\boldsymbol{z}$ is sampled from $\mathcal{N}(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}})$ and fed into the decoder with the same domain label s/t as the condition to reconstruct the input as $\tilde{\boldsymbol{x}}^{s}/\tilde{\boldsymbol{x}}^{t}$. On the other hand, the sampled latent code $\boldsymbol{z}$ can also be decoded with the condition of the other domain label t/s to generate the synthetic feature in a different domain as $\tilde{\boldsymbol{x}}^{st}/\tilde{\boldsymbol{x}}^{ts}$.

The model is trained by feeding paired source and target-domain samples $\{\boldsymbol{x}^{s}, \boldsymbol{x}^{t}\}$ randomly selected from the same class. The loss function to minimize can be formulated as:

$$\begin{aligned}
&\mathcal{L}_{CCVAE}(\Phi, \theta; \boldsymbol{x}^{\boldsymbol{s}}, \boldsymbol{x}^{\boldsymbol{t}}) = \\
&(\mathcal{L}_{recon}\left(\boldsymbol{x}^{\boldsymbol{s}}, \tilde{\boldsymbol{x}}^{s}\right) + \mathcal{L}_{recon}(\boldsymbol{x}^{\boldsymbol{t}}, \tilde{\boldsymbol{x}}^{t})) \\
&+ \left(\mathcal{L}_{cross\_recon}(\boldsymbol{x}^{\boldsymbol{s}}, \tilde{\boldsymbol{x}}^{ts}) + \mathcal{L}_{cross\_recon}(\boldsymbol{x}^{\boldsymbol{t}}, \tilde{\boldsymbol{x}}^{st})\right) \\
&+ \lambda D_{KL}\left(\mathcal{N}(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}})||\mathcal{N}(\mathbf{0}, \boldsymbol{I})\right)
\end{aligned} \tag{3}$$

The first terms measure the reconstruction errors for both source and target-domain samples. The second terms measure the cross-domain reconstruction errors. Although the samples in the pair of $\{\boldsymbol{x}^{\boldsymbol{s}}, \tilde{\boldsymbol{x}}^{ts}\}$ or $\{\boldsymbol{x}^{\boldsymbol{t}}, \tilde{\boldsymbol{x}}^{st}\}$ 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 $\mathcal{L}_{cross\_recon}$ facilitate the model to generate class discriminative features across domains. For those $\boldsymbol{x}^{s}$ belonging to unseen classes, there is no valid target-domain samples $\boldsymbol{x}^{t}$ from the same class. We use dummy features in practice and exclude the loss terms involving these dummy features. The third term aims to reduce the KL divergence between the distributions of the latent code and a normal distribution. It serves as a regularization term in the same way as in the VAE model. λ is a hyper-parameter balancing the KL divergence and reconstruction errors. We use the mean squared error (MSE) loss for both $\mathcal{L}_{recon}$ and $\mathcal{L}_{cross\_recon}$. The effectiveness of different terms in Eq. (3) will be further investigated and discussed in Section 4.

3.4. Target domain image classification

Once the CCVAE is trained, we can use it to generate synthetic features by the cross-domain reconstruction pipelines. Specifically, given a feature vector $\boldsymbol{x}^{s}$ (or $\boldsymbol{x}^{t}$), the model can generate $\tilde{\boldsymbol{x}}^{st}$ (or $\tilde{\boldsymbol{x}}^{ts}$) which should have the same class label as the input. In this way, we can generate synthetic features for unseen classes in the target domain with the source-domain samples. We use real data $\mathcal{D}^{s}$ and $\mathcal{D}^{t}$ together with synthetically generated features from them to train a unified neural network classifier for all classes and both domains. The classifier is then used to classify test images.

The proposed CCVAE can be summarized in Algorithm 1.

3.5. Relation to VAE based methods for ZSL

Our CCVAE framework for the GZSDA problem is distinct from those VAE networks used for ZSL (Schonfeld, Ebrahimi, Sinha, Darrell, & Akata, 2019) in at least two aspects. Firstly, for ZSL, the synthetic data are generated from noise sampled from a standard Normal distribution by the trained decoder conditioned on the class-level semantic representations; our CCVAE generate synthetic data from the latent codes which are encoded from real data (i.e. cross-domain data transformation). Secondly, our CCVAE generate data for both source and target domains and trains a unified classifier for both domains; whilst in ZSL, the data generation is only required from the semantic space (i.e. source domain) to the visual space (i.e. target domain).

4. Experiments and results

As the first attempt to address the GZSDA problem, we present a benchmark on GZSDA with extensive experiments on three datasets. We compare our proposed CCVAE with baseline methods and state-of-the-art methods for zero-shot learning (Mishra et al., 2018; Wang et al., 2019; Wang & Chen, 2017b) which have been adapted to the GZSDA problem.

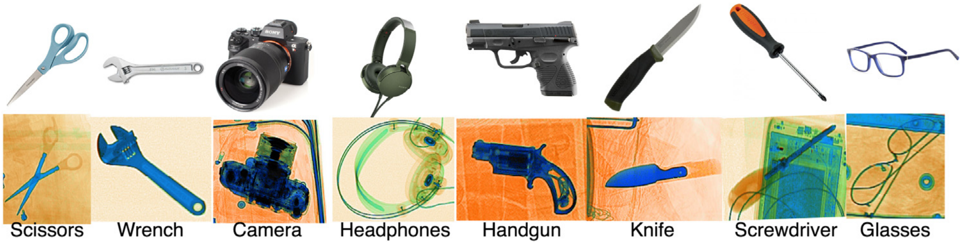


Fig. 3. Sample images from the BaggageXray-20 dataset (upper: regular; bottom: X-ray).

Algorithm 1 Coupled Conditional Variational AutoEncoder (CCVAE) for Generalised Zero-Shot Domain Adaptation (GZSDA)

```
Input: Labelled source data set $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}, i = 1, 2, ..., n_s$ and
    labelled target data set $\mathcal{D}^t = \{\boldsymbol{x}_i^t, y_i^t\}, i = 1, 2, ..., n_t$, number
    of training iterations $T_1$ and $T_2$ for CCVAE and the classifier.
Output: A unified classifier $f(x; \theta)$.
Training the CCVAE model:
$k \leftarrow 0$;
while $k < T_1$ do
    $k \leftarrow k + 1$;
    Randomly sample a batch of source-domain samples $B^s$;
    For each sample $\boldsymbol{x}_i^s$ in $B^s$, randomly choose a target-domain
        sample $\boldsymbol{x}_i^t$ from the same class $y_i^s$ to pair with $\boldsymbol{x}_i^s$ if this class
        belongs to seen classes, otherwise create a dummy sample
        to pair with $\boldsymbol{x}_i^s$;
    Combine $B^s$ and samples chosen from the target domain (or
        dummy samples) to form a batch of paired training samples
        and feed them into the model for one forward pass;
    Compute the loss in Eq. (3) (the loss contributed by the
        dummy features will be excluded) and update the model
        parameters.
end while
Training the unified classifier $f(x; \theta)$:
$k \leftarrow 0$;
while $k < T_2$ do
    $k \leftarrow k + 1$;
    Randomly sample a batch of training samples $\{\boldsymbol{X}^s, \boldsymbol{Y}^s\}$ from
        the source domain; feed them into the learned CCVAE
        model to get the cross-domain reconstructed $\tilde{\boldsymbol{X}}^{st}$;
    Randomly sample a batch of training samples $\{\boldsymbol{X}^t, \boldsymbol{Y}^t\}$ from
        the target domain; ; feed them into the learned CCVAE
        model to get the cross-domain reconstructed $\tilde{\boldsymbol{X}}^{ts}$;
    Combine $\boldsymbol{X}^s, \boldsymbol{X}^t, \tilde{\boldsymbol{X}}^{st}, \tilde{\boldsymbol{X}}^{ts}$ and their corresponding labels
        $\boldsymbol{Y}^s, \boldsymbol{Y}^t, \boldsymbol{Y}^s, \boldsymbol{Y}^t$ to form a training batch for the classifier
        $f(x; \theta)$;
    Compute the cross-entropy loss and update the classifier
        parameters $\theta$.
end while
```

4.1. Dataset

BaggageXray-20 This dataset is collected for automatic object recognition in aviation security baggage screening.[1] The dataset consists of images from two domains: dual-energy colour-mapped X-ray and regular colour photograph (denoted as *X-ray* and *regular* domains respectively). Compared with the prevalence of regular RGB images, X-ray images are rarely available and have significantly different appearances as shown in Fig. 3. We consider 20 object classes: *binder-clip, bottle, bullet, camera, fork, glasses, handgun, headphones, keys, knife, lock, mug, pliers, rifle, scissors, screwdriver, spoon, tableknife, wrench-big, wrench-small.*

For each object class, we use 3–10 different physical instances for X-ray scanning. To diversify the data, we simulate the baggage scanning in a checkpoint at the airport. Specifically, the objects are arbitrarily put in different baggage (e.g., backpacks, suitcases and plastic trays) containing non-target objects (e.g., clothes, books, laptops, etc.). The baggage containing both target objects and non-target clutter are scanned with a Gilardoni dual-energy X-ray scanner (FEP ME 640 AMX) and the resultant colour-mapped X-ray images of baggage can have a cluttered background and overlap between target objects and non-target objects (see Fig. 4). Subsequently, we manually crop and annotate the target object patches from the whole baggage scan images using the open-source tool LabelImg.[2]

The regular domain images were collected using the open-source image scrapping tool GoogleScrapper.[3] We use the class names as keywords to search images using image search engines (e.g., Yahoo Image) and automatically download the search results. The corrupted and irrelevant results were manually removed to reduce the noise in the dataset.

The statistics of the dataset are shown in Fig. 5.

There are 4620 and 3444 images in the X-ray and regular domains, respectively. In our preliminary experiments, ResNet101 features were proved to be more discriminative than its counterparts such as VGG16 and ResNet50, hence we use ResNet101 pre-trained on the ImageNet to extract 2048-dim features for images from both domains. To simulate the GZSDA problem setting, the 20 classes were randomly split into two subsets: 10 classes as the seen classes and the rest 10 classes as the unseen classes. Five random seen/unseen class splits were used in our experiments to get statistics of the experimental results. Two domain adaptation tasks (i.e. regular → X-ray and X-ray → regular) were employed in the experiments. When a domain serves as the source domain, all images are used to form the labelled source dataset $\mathcal{D}^s$ whilst for the domain serving as the target domain, we randomly reserve 50% of the images from each class for testing and the rest 50% from seen classes form the labelled target dataset $\mathcal{D}^t$.

Office31 (Saenko, Kulis, Fritz, & Darrell, 2010) is an image dataset which consists of three domains: Amazon (A), Webcam (W) and DSLR (D). There are 31 common classes for all three domains containing 4110 images in total. In our experiments, we divide 31 classes into 16 seen classes and 15 unseen classes randomly and generate 5 different splits for calculating the statistics of results.

Office–Home Office–Home (Venkateswara, Eusebio, Chakraborty, & Panchanathan, 2017) is a dataset commonly used for domain adaptation. It consists of four domains: artistic images (Art),

[1] The dataset has been publicly and openly released including annotations, at https://github.com/hellowangqian/gzsda.

[2] https://github.com/tzutalin/labelImg

[3] https://github.com/NikolaiT/GoogleScraper

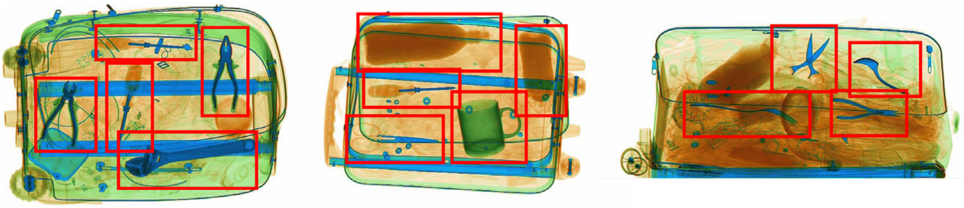


Fig. 4. The X-ray domain images are manually cropped from the baggage scan images with cluttered backgrounds.

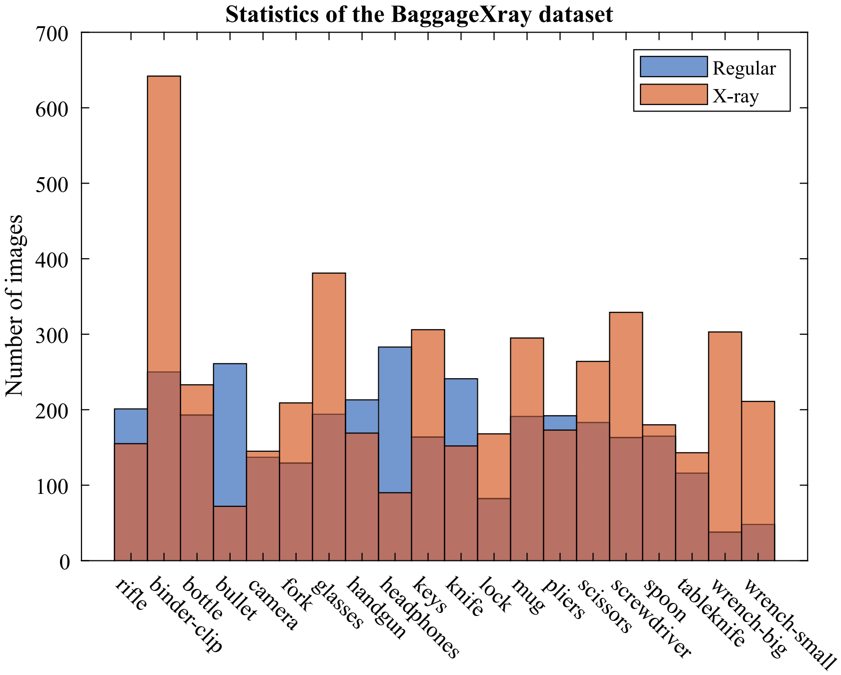


Fig. 5. Statistics of the BaggageXray-20 dataset (the heights of blue and orange bars indicate the numbers of images belonging to 20 classes from the Regular domain and the X-ray domain, respectively).

Clipart, Product images and Real-World images. There are 65 object classes in each domain with a total number of 15,588 images. We follow prior works to extract ResNet50 features in our experiments and divide these 65 classes into 35 seen classes and 30 unseen classes randomly and generate 5 different splits for calculating the statistics of results. Given four domains, there are 12 different domain adaptation tasks. The training and test dataset creation strategy is identical to the BaggageXray-20 dataset.

MNIST/Fashion-MNIST/EMNIST (*X*MNIST) We follow previous works on ZSDA (Peng et al., 2018; Wang & Jiang, 2019) and conducted experiments using MNIST (LeCun, Bottou, Bengio, & Haffner, 1998), Fashion-MNIST (FMNIST) (Xiao, Rasul, & Vollgraf, 2017) and EMNIST (Cohen, Afshar, Tapson, & Van Schaik, 2017) (denoted collectively as *X*MNIST).

MNIST contains 70,000 grey images of digits 0–9 which were divided into two subsets: 60,000 for training and 10,000 for testing. FMNIST is a similar dataset containing 70,000 grey images of 10 classes, i.e. *T-shirt, trouser, pullover, dress, coat, sandals, shirt, sneaker, bag* and *ankle boot*. Similarly, a fixed 6:1 split for training and testing is available along with the dataset. EMNIST is an extension of MNIST containing 26-class English letters (the uppercase and lowercase letters are merged as one class). A fixed 6:1 split of the total 20,800 images is available for training and testing respectively.

All these three datasets contain grey images of the same size of 28 × 28. We consider these grey images as in the *Grey* domain from which we create another two domains *Colour* and *Negative*. The *Colour* domain images were created using the method proposed in Ganin and Lempitsky (2015). Specifically, for a given image I, a random patch P of the same size was cropped from a colour image in BSDS500 (Arbelaez, Maire, Fowlkes, & Malik, 2010) and the colour version of I is created by $I_c = |I - P|$ for all channels. The *Negative* domain images are obtained by $I_n = 255 - I$. Exemplar images of each domain from three datasets are shown in Fig. 6.

There are 6 different combinations of 3 domains to form 6 domain adaptation tasks among which we report the representative results of *Grey* → *Colour*, *Colour* → *Grey* and *Negative* → *Colour*. In each domain adaptation task, we choose any two datasets as the seen and unseen classes respectively. As a result, for each domain adaptation task there can be 6 sub-tasks with different combinations of seen and unseen datasets. AlexNet (Krizhevsky et al., 2012) was trained from scratch using the training data $\mathcal{D}^s$ and $\mathcal{D}^t$ to extract features for a specific adaptation task.

4.2. Implementation details

The proposed method was implemented in PyTorch.[4] (Paszke et al., 2019) Both the encoder and decoder were three-layer fully connected neural networks. For BaggageXray-20 and Office–Home datasets, the VAE share the same architecture of $2048 - 512 - 64 - 64 - 512 - 2048$ where 64 is the dimension of the latent code $\boldsymbol{z}$. For *X*MNIST datasets, the VAE has an architecture

[4] https://github.com/hellowangqian/gzsda

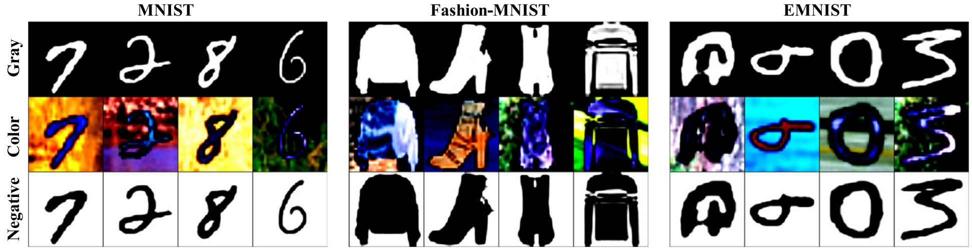


Fig. 6. Exemplar images from the XMNIST dataset.

Table 2
Experimental results (%) on BaggageXray dataset with 10 unseen classes (mean and standard error of the mean (SEM) over five different seen/unseen class splits are reported).

Method	Regular → Xray			Xray → Regular		
	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H
Source Only	23.4 ± 3.0	20.4 ± 3.0	20.2 ± 0.7	47.9 ± 4.0	42.7 ± 4.0	43.7 ± 1.3
Baseline (1NN)	75.0 ± 2.4	1.9 ± 0.5	3.6 ± 0.9	93.8 ± 1.5	12.6 ± 1.3	22.1 ± 2.0
Baseline (NN)	84.3 ± 1.9	2.5 ± 0.4	4.8 ± 0.7	$\mathbf{95.0 \pm 0.8}$	20.4 ± 3.8	32.6 ± 5.5
BiDiLEL (Wang & Chen, 2017b)	80.8 ± 2.2	8.2 ± 0.4	14.9 ± 0.7	94.6 ± 0.9	2.8 ± 0.6	5.5 ± 1.1
CADA-VAE (Schonfeld et al., 2019)	47.3 ± 6.7	24.3 ± 5.0	31.6 ± 4.3	73.0 ± 7.3	26.1 ± 4.1	38.3 ± 4.9
LPP (Wang et al., 2019)	$\mathbf{85.7 \pm 1.6}$	10.2 ± 1.1	18.1 ± 1.7	92.9 ± 1.3	30.4 ± 2.4	45.6 ± 2.9
CCVAE	77.2 ± 1.9	$\mathbf{29.0 \pm 1.5}$	$\mathbf{42.1 \pm 1.8}$	90.8 ± 1.0	$\mathbf{52.3 \pm 2.5}$	$\mathbf{66.1 \pm 1.9}$

Table 3
Experimental results (%) on the Office31 dataset with 15 unseen classes (mean and standard error of the mean (SEM) over five different seen/unseen class splits are reported).

Method	A → D			A → W		
	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H
Source Only	80.8 ± 1.5	80.7 ± 1.6	80.6 ± 0.1	73.3 ± 1.5	77.9 ± 1.6	75.4 ± 0.1
Baseline (1NN)	90.5 ± 1.2	67.5 ± 1.8	77.2 ± 0.9	91.0 ± 0.8	60.6 ± 1.3	72.7 ± 0.9
Baseline (NN)	$\mathbf{92.9 \pm 1.2}$	66.4 ± 3.2	77.2 ± 1.9	$\mathbf{94.2 \pm 0.6}$	57.6 ± 2.0	71.4 ± 1.5
BiDiLEL (Wang & Chen, 2017b)	90.5 ± 1.4	26.5 ± 4.7	40.2 ± 5.5	89.7 ± 1.2	18.8 ± 2.1	30.9 ± 2.9
CADA-VAE (Schonfeld et al., 2019)	79.9 ± 2.1	39.7 ± 5.0	52.8 ± 4.5	80.4 ± 2.6	41.3 ± 9.0	54.0 ± 7.4
LPP (Wang et al., 2019)	90.0 ± 1.8	73.9 ± 2.4	80.9 ± 1.1	91.0 ± 1.2	65.6 ± 1.9	76.1 ± 1.0
CCVAE	89.1 ± 2.4	$\mathbf{86.4 \pm 2.7}$	$\mathbf{87.4 \pm 1.0}$	86.8 ± 1.4	$\mathbf{82.8 \pm 2.0}$	$\mathbf{84.6 \pm 0.8}$
	D → A			D → W		
	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H
Source Only	66.1 ± 2.5	$\mathbf{60.4 \pm 2.7}$	62.7 ± 0.3	99.2 ± 0.2	$\mathbf{99.5 \pm 0.3}$	99.4 ± 0.0
Baseline (1NN)	$\mathbf{86.6 \pm 0.7}$	36.0 ± 0.5	50.9 ± 0.6	$\mathbf{99.8 \pm 0.1}$	99.2 ± 0.3	$\mathbf{99.5 \pm 0.1}$
Baseline (NN)	85.5 ± 1.4	39.9 ± 1.3	54.3 ± 1.1	98.2 ± 0.1	90.6 ± 1.9	94.2 ± 1.0
BiDiLEL (Wang & Chen, 2017b)	86.1 ± 1.1	8.1 ± 1.3	14.6 ± 2.1	88.7 ± 1.1	42.8 ± 3.3	57.3 ± 2.8
CADA-VAE (Schonfeld et al., 2019)	82.2 ± 1.9	32.8 ± 6.5	46.5 ± 7.1	77.4 ± 2.6	62.6 ± 5.0	69.1 ± 3.7
LPP (Wang et al., 2019)	84.3 ± 1.4	58.8 ± 2.3	$\mathbf{69.1 \pm 1.2}$	99.1 ± 0.5	95.2 ± 1.3	97.1 ± 0.5
CCVAE	83.6 ± 1.4	59.0 ± 1.5	$\mathbf{69.1 \pm 0.6}$	97.4 ± 0.5	95.2 ± 0.9	96.3 ± 0.3
	W → A			W → D		
	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H
Source Only	62.0 ± 2.5	59.0 ± 2.6	60.0 ± 0.2	$\mathbf{99.8 \pm 0.2}$	$\mathbf{99.3 \pm 0.2}$	$\mathbf{99.6 \pm 0.0}$
Baseline (1NN)	$\mathbf{86.9 \pm 0.7}$	36.4 ± 1.2	51.3 ± 1.2	$\mathbf{99.8 \pm 0.1}$	$\mathbf{99.3 \pm 0.2}$	$\mathbf{99.6 \pm 0.1}$
Baseline (NN)	86.3 ± 1.3	39.5 ± 1.6	54.0 ± 1.4	98.6 ± 0.3	97.0 ± 0.8	97.8 ± 0.3
BiDiLEL (Wang & Chen, 2017b)	86.4 ± 1.1	6.8 ± 1.5	12.4 ± 2.5	89.0 ± 1.7	56.3 ± 5.0	68.4 ± 3.8
CADA-VAE (Schonfeld et al., 2019)	82.1 ± 2.7	30.3 ± 3.6	44.1 ± 4.1	79.1 ± 3.6	46.9 ± 8.9	58.3 ± 7.1
LPP (Wang et al., 2019)	84.8 ± 1.2	57.5 ± 2.5	68.3 ± 1.5	99.5 ± 0.2	97.6 ± 0.5	98.6 ± 0.3
CCVAE	83.5 ± 1.2	$\mathbf{60.4 \pm 1.9}$	$\mathbf{69.9 \pm 0.9}$	98.5 ± 0.2	99.2 ± 0.2	98.8 ± 0.1

of $512-128-32-32-128-512$ where 512 is the dimension of features extracted in the first step and 32 is the dimension of the latent space. The ReLU layer was employed after each fully connected layer for non-linearity. For the classifier in step 3, we used a simple two-layer linear neural network (no hidden layer) across all experiments. We used the Adam optimizer to train the CCVAE with the learning rate of $1e-3$ for a fixed number of epochs (50 epochs for BaggageXray20 and Office–Home, 10 epochs for XMNIST datasets). The value of λ was dynamically adjusted by a gradual warm-up strategy (Goyal et al., 2017) from 0 up to 0.2 to facilitate the model training.

4.3. Experimental results

We compare the performance of CCVAE with three baseline models, two ZSL methods (i.e. BiDiLEL (Wang & Chen, 2017b) and CADA-VAE (Schonfeld et al., 2019)) adapted for GZSDA and one existing GZSDA method LPP (Wang et al., 2019). We do not consider the ZSDA methods in Kutbi, Peng, and Wu (2021), Peng et al. (2018), Wang et al. (2021), Wang and Jiang (2019, 2021) because they require **paired** images and cannot discriminate seen and unseen classes in the target domain.

Table 4
Experimental results (%) on Office–Home dataset with 30 unseen classes (mean and standard error of the mean (SEM) over five different seen/unseen class splits are reported).

Method	Art → ClipArt			Art → Product			Art → RealWorld		
	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H
Source Only	36.8 ± 0.4	34.2 ± 0.4	35.4 ± 0.1	57.0 ± 0.6	56.5 ± 0.7	56.7 ± 0.1	62.9 ± 0.9	61.0 ± 1.0	61.9 ± 0.1
Baseline (1NN)	70.4 ± 0.5	19.3 ± 0.6	30.3 ± 0.7	87.5 ± 0.4	39.7 ± 0.6	54.6 ± 0.5	80.1 ± 1.0	52.0 ± 1.2	63.0 ± 0.6
Baseline (NN)	71.7 ± 0.5	28.1 ± 0.5	40.3 ± 0.5	89.6 ± 0.2	53.3 ± 1.1	66.8 ± 0.8	85.6 ± 0.9	63.0 ± 0.6	72.6 ± 0.3
BiDiLEL (Wang & Chen, 2017b)	**74.3 ± 0.7**	5.8 ± 0.7	10.7 ± 1.1	$\underline{89.8 \pm 0.3}$	6.3 ± 0.8	11.7 ± 1.4	**87.5 ± 0.9**	5.8 ± 0.3	10.9 ± 0.5
CADA-VAE (Schonfeld et al., 2019)	55.2 ± 1.4	27.3 ± 2.5	36.4 ± 2.2	77.7 ± 1.5	43.5 ± 1.6	55.7 ± 1.1	72.3 ± 2.5	53.5 ± 2.0	61.4 ± 1.3
LPP (Wang et al., 2019)	$\underline{73.8 \pm 0.6}$	$\underline{40.7 \pm 0.9}$	**52.4 ± 0.8**	**90.0 ± 0.3**	$\underline{60.5 \pm 0.9}$	$\underline{72.4 \pm 0.6}$	$\underline{85.7 \pm 0.7}$	**68.8 ± 0.4**	**76.3 ± 0.2**
CCVAE	$\underline{66.7 \pm 0.5}$	**41.3 ± 0.7**	$\underline{51.0 \pm 0.4}$	87.4 ± 0.3	**65.3 ± 0.9**	**74.7 ± 0.5**	84.1 ± 0.8	**68.8 ± 0.7**	$\underline{75.6 \pm 0.3}$
	ClipArt → Art			ClipArt → Product			ClipArt → RealWorld		
	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H
Source Only	48.0 ± 0.7	44.2 ± 0.9	46.0 ± 0.1	55.6 ± 0.8	57.2 ± 0.9	56.3 ± 0.1	59.3 ± 1.2	59.3 ± 1.4	59.2 ± 0.1
Baseline (1NN)	61.0 ± 0.2	32.3 ± 0.9	42.2 ± 0.8	85.3 ± 0.5	44.6 ± 1.1	58.6 ± 0.9	81.0 ± 1.6	45.3 ± 1.1	58.0 ± 0.7
Baseline (NN)	$\underline{72.6 \pm 0.4}$	33.0 ± 1.3	45.3 ± 1.1	$\underline{88.2 \pm 0.3}$	50.0 ± 1.6	63.8 ± 1.3	$\underline{86.7 \pm 0.6}$	48.5 ± 1.6	62.1 ± 1.2
BiDiLEL (Wang & Chen, 2017b)	**74.7 ± 0.8**	4.5 ± 0.4	8.4 ± 0.8	**89.5 ± 0.3**	6.0 ± 0.7	11.2 ± 1.2	**87.3 ± 0.8**	5.0 ± 0.6	9.4 ± 1.0
CADA-VAE (Schonfeld et al., 2019)	53.3 ± 1.5	27.9 ± 3.0	36.5 ± 2.5	77.7 ± 1.0	37.7 ± 1.6	50.7 ± 1.6	73.0 ± 2.4	43.7 ± 1.5	54.7 ± 1.1
LPP (Wang et al., 2019)	72.1 ± 0.8	$\underline{48.1 \pm 0.8}$	$\underline{57.7 \pm 0.7}$	87.6 ± 0.4	$\underline{58.8 \pm 1.5}$	$\underline{70.3 \pm 1.1}$	86.0 ± 0.9	$\underline{59.4 \pm 2.0}$	$\underline{70.1 \pm 1.2}$
CCVAE	69.2 ± 0.8	**51.8 ± 0.5**	**59.2 ± 0.3**	85.4 ± 0.5	**63.6 ± 1.7**	**72.8 ± 1.0**	83.3 ± 0.7	**65.1 ± 1.8**	**73.0 ± 0.9**
	Product → Art			Product → ClipArt			Product → RealWorld		
	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H
Source Only	51.8 ± 1.4	47.3 ± 1.6	49.2 ± 0.4	40.4 ± 1.0	40.2 ± 1.1	40.2 ± 0.1	68.4 ± 1.6	66.3 ± 1.8	67.2 ± 0.2
Baseline (1NN)	63.4 ± 0.9	39.5 ± 1.4	48.5 ± 0.9	70.3 ± 0.4	29.3 ± 0.9	41.3 ± 0.9	81.1 ± 1.2	61.9 ± 1.6	70.1 ± 0.6
Baseline (NN)	$\underline{72.0 \pm 0.7}$	35.8 ± 1.4	47.7 ± 1.2	72.0 ± 0.4	25.7 ± 0.9	37.9 ± 1.0	**88.1 ± 0.6**	63.0 ± 1.7	73.4 ± 1.0
BiDiLEL (Wang & Chen, 2017b)	**74.3 ± 0.9**	6.4 ± 0.8	11.7 ± 1.3	**74.2 ± 0.7**	5.4 ± 0.4	10.0 ± 0.7	$\underline{87.3 \pm 0.9}$	9.3 ± 1.3	16.7 ± 2.0
CADA-VAE (Schonfeld et al., 2019)	52.5 ± 2.4	30.1 ± 2.7	38.1 ± 1.5	55.9 ± 3.7	25.1 ± 2.8	34.5 ± 2.0	70.8 ± 1.3	51.8 ± 2.2	59.8 ± 1.7
LPP (Wang et al., 2019)	69.9 ± 0.8	$\underline{50.2 \pm 0.9}$	$\underline{58.4 \pm 0.5}$	$\underline{72.6 \pm 0.5}$	$\underline{41.0 \pm 0.7}$	$\underline{52.4 \pm 0.6}$	86.2 ± 0.7	$\underline{71.4 \pm 1.0}$	**78.1 ± 0.4**
CCVAE	67.6 ± 0.6	**52.3 ± 0.6**	**59.0 ± 0.3**	66.9 ± 0.2	**43.8 ± 1.0**	**52.9 ± 0.8**	84.7 ± 0.8	**72.0 ± 1.6**	$\underline{77.7 \pm 0.6}$
	RealWorld → Art			RealWorld → ClipArt			RealWorld → Product		
	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H
Source Only	57.9 ± 0.6	55.9 ± 0.8	56.9 ± 0.1	43.2 ± 0.6	43.5 ± 0.7	43.3 ± 0.1	71.7 ± 0.9	72.2 ± 1.0	71.9 ± 0.1
Baseline (1NN)	64.7 ± 0.7	52.3 ± 0.7	57.8 ± 0.2	71.6 ± 0.4	32.5 ± 0.7	44.7 ± 0.7	87.0 ± 0.6	67.5 ± 1.0	76.0 ± 0.5
Baseline (NN)	$\underline{73.7 \pm 0.5}$	54.3 ± 0.5	62.5 ± 0.5	72.2 ± 0.3	33.5 ± 0.8	45.7 ± 0.8	**90.0 ± 0.2**	72.6 ± 1.3	80.3 ± 0.8
BiDiLEL (Wang & Chen, 2017b)	**73.9 ± 1.1**	9.0 ± 1.2	15.9 ± 1.8	**74.3 ± 0.8**	8.3 ± 0.6	14.9 ± 1.0	$\underline{89.4 \pm 0.3}$	14.0 ± 1.4	24.2 ± 2.0
CADA-VAE (Schonfeld et al., 2019)	52.1 ± 2.5	41.4 ± 2.3	46.0 ± 1.3	55.7 ± 3.0	32.1 ± 2.5	40.6 ± 2.0	74.8 ± 2.9	59.0 ± 3.0	65.8 ± 1.2
LPP (Wang et al., 2019)	72.2 ± 0.7	**64.6 ± 0.9**	**68.1 ± 0.5**	$\underline{73.8 \pm 0.9}$	$\underline{46.5 \pm 1.0}$	**57.0 ± 0.7**	89.3 ± 0.2	**78.6 ± 0.5**	**83.6 ± 0.2**
CCVAE	71.1 ± 0.7	$\underline{62.1 \pm 1.1}$	$\underline{66.2 \pm 0.4}$	67.5 ± 0.3	**47.8 ± 0.6**	$\underline{55.9 \pm 0.4}$	87.2 ± 0.3	$\underline{76.9 \pm 1.3}$	$\underline{81.7 \pm 0.6}$

- **Source Only** uses only source-domain data and the 1 Nearest Neighbor (1NN) classifier. Due to the domain shift, applying the classifier trained on the source-domain data directly to the target-domain data will suffer a significant performance drop when the domain shift is large.
- **Baseline (1NN/NN)** uses training data from both domains (i.e. $\mathcal{D}^s$ and $\mathcal{D}^t$) and a simple classifier 1NN or Neural Networks (NN) with the same architecture as that used in step 3 of CCVAE. Compared with the **Source Only** method, the addition of target-domain data may lead to the overfitting the seen classes since there exist data for only seen classes in the target domain $\mathcal{D}^t$.
- **BiDiLEL** (Wang & Chen, 2017b) is a representative method for zero-shot learning based on common subspace learning. To use such a method for the GZSDA problem, we compute the class means of source samples as the class-level side information for generalized zero-shot learning.
- **CADA-VAE** (Schonfeld et al., 2019) is another representative method for generalized zero-shot learning based on a generative model VAE. Again we take the class means of source samples as the class-level semantic representations.
- **LPP** (Wang et al., 2019) is the only approach to GZSDA in the literature. The approach uses the locality preserving projection algorithm (Wang & Chen, 2017b; Wong & Zhao, 2012) and iterative pseudo-labelling to learn a common subspace from the source and target domains so that two domains are aligned in the subspace. In contrast to learning a common subspace that is difficult to scale, our proposed method learns a unified classifier in the original feature space by augmenting the training data using the proposed generative model CCVAE.

Following the generalized ZSL works (Xian et al., 2019), we report the mean per-class classification accuracy for seen and unseen classes and their harmony mean (Acc_{seen}, Acc_{unseen} and H). Our experimental results are shown in Tables 2–5. The best and second best results are highlighted in **bold** and $\underline{\text{underlined}}$, respectively. It can be seen from Table 2 that the discrepancy of data distribution between regular and X-ray domains is significant as the *Source Only* method achieves low accuracy on both adaptation tasks (Table 2). When labelled target samples from seen classes are employed, the two baseline methods achieve much better performance on seen classes at the sacrifice of accuracy on the unseen classes. The ZSL method BiDiLEL generally performs well on seen classes but poorly on unseen classes due to the notorious issue of overfitting the seen classes in GZSL. The GZSL method CADA-VAE can balance the performance on seen and unseen classes and hence achieves higher values of H but is still outperformed by LPP and our proposed CCVAE since the traditional ZSL/GZSL methods cannot take advantage of the source-domain data properly. The proposed CCVAE achieves the highest H values by improving the recognition accuracy of unseen classes whilst maintaining the accuracy of seen classes.

The experimental results on the image classification dataset Office31 are shown in Table 3. Our proposed CCVAE can achieve the best performance in terms of H values on 4 out of 6 tasks and comparably good performance with the best performance on the remaining 2 tasks. Since the data distribution shift between domains in this dataset is relatively small, we can see that the *Source*

Table 5
Experimental results on *X*MNIST datasets (mean and standard deviation of *H* over five trials are reported).

Domains	Method	Seen: MNIST		Seen: FMNIST		Seen: EMNIST	
		FMNIST	EMNIST	MNIST	EMNIST	MNIST	FMNIST
Grey → *Colour*	Source Only	7.3 ± 0.1	2.5 ± 0.1	6.1 ± 0.2	3.5 ± 0.1	2.2 ± 0.1	3.3 ± 0.1
	Baseline (1NN)	46.3 ± 0.6	30.2 ± 0.4	38.2 ± 0.5	17.1 ± 0.2	51.8 ± 0.7	47.6 ± 0.4
	Baseline (NN)	50.0 ± 0.1	36.2 ± 0.1	47.9 ± 0.1	27.5 ± 0.1	42.7 ± 0.1	42.5 ± 0.1
	BiDiLEL (Wang & Chen, 2017b)	39.9 ± 2.0	35.8 ± 1.1	30.6 ± 1.9	13.9 ± 1.4	52.8 ± 1.9	37.6 ± 1.6
	CADA-VAE (Schonfeld et al., 2019)	39.2 ± 1.9	35.5 ± 1.3	30.0 ± 3.5	19.3 ± 1.6	46.3 ± 1.1	43.3 ± 0.6
	LPP (Wang et al., 2019)	$\underline{61.3 \pm 0.5}$	$\underline{43.5 \pm 0.6}$	$\underline{58.8 \pm 0.4}$	$\underline{39.1 \pm 0.4}$	$\underline{68.8 \pm 0.3}$	**58.8 ± 0.2**
	CCVAE	**63.9 ± 0.3**	**61.3 ± 0.1**	**69.1 ± 0.8**	**45.3 ± 1.2**	**71.4 ± 0.9**	$\underline{56.0 \pm 1.7}$
Colour → *Grey*	Source Only	86.5 ± 0.6	89.0 ± 0.3	87.1 ± 0.6	81.4 ± 0.6	89.6 ± 0.3	80.8 ± 0.3
	Baseline (1NN)	85.6 ± 0.5	87.6 ± 0.3	90.9 ± 0.2	85.2 ± 0.2	90.7 ± 0.1	82.8 ± 0.5
	Baseline (NN)	**89.6 ± 0.0**	$\underline{89.1 \pm 0.0}$	**92.5 ± 0.0**	**87.8 ± 0.0**	$\underline{91.0 \pm 0.0}$	**86.9 ± 0.0**
	BiDiLEL (Wang & Chen, 2017b)	29.0 ± 2.6	31.8 ± 2.1	18.7 ± 2.9	10.1 ± 3.4	53.7 ± 4.3	38.9 ± 2.7
	CADA-VAE (Schonfeld et al., 2019)	39.1 ± 2.2	38.9 ± 2.7	47.3 ± 4.5	31.2 ± 5.1	52.9 ± 2.0	45.8 ± 4.4
	LPP (Wang et al., 2019)	86.6 ± 0.2	80.3 ± 0.2	90.9 ± 0.1	81.3 ± 0.3	85.2 ± 0.4	81.5 ± 0.3
	CCVAE	$\underline{88.8 \pm 0.0}$	**90.4 ± 0.0**	$\underline{92.1 \pm 0.0}$	$\underline{87.3 \pm 0.1}$	**92.1 ± 0.0**	$\underline{86.3 \pm 0.1}$
Neg. → *Colour*	Source Only	5.1 ± 0.3	1.9 ± 0.1	5.0 ± 0.4	1.4 ± 0.0	1.8 ± 0.1	2.7 ± 0.0
	Baseline (1NN)	41.4 ± 0.9	28.8 ± 0.6	25.5 ± 0.5	9.6 ± 0.1	64.9 ± 0.4	46.9 ± 1.1
	Baseline (NN)	49.7 ± 0.1	36.7 ± 0.2	44.5 ± 0.1	23.9 ± 0.2	55.9 ± 0.0	40.8 ± 0.1
	BiDiLEL (Wang & Chen, 2017b)	36.6 ± 2.6	31.3 ± 1.1	27.0 ± 2.1	13.5 ± 0.8	50.9 ± 1.7	36.6 ± 1.7
	CADA-VAE (Schonfeld et al., 2019)	39.3 ± 1.7	34.5 ± 1.8	26.7 ± 1.4	18.4 ± 4.6	42.7 ± 1.1	41.0 ± 0.9
	LPP (Wang et al., 2019)	$\underline{68.1 \pm 0.6}$	$\underline{45.2 \pm 0.6}$	$\underline{54.1 \pm 0.8}$	$\underline{30.5 \pm 0.4}$	$\underline{69.2 \pm 0.2}$	**66.7 ± 0.2**
	CCVAE	**70.2 ± 0.5**	**63.9 ± 0.3**	**68.3 ± 1.1**	**47.1 ± 0.8**	**73.6 ± 0.9**	$\underline{62.8 \pm 1.3}$
Grey → *Neg*.	Source Only	20.6 ± 1.7	0.0 ± 0.0	23.5 ± 0.5	6.2 ± 0.7	0.0 ± 0.0	7.8 ± 0.9
	Baseline (1NN)	39.9 ± 1.4	20.4 ± 0.5	43.4 ± 1.8	9.1 ± 0.9	62.7 ± 0.5	65.9 ± 0.9
	Baseline (NN)	37.3 ± 0.1	34.5 ± 0.2	56.9 ± 0.5	$\underline{29.9 \pm 0.2}$	46.8 ± 0.1	60.9 ± 0.1
	BiDiLEL (Wang & Chen, 2017b)	24.7 ± 3.9	34.7 ± 2.2	25.8 ± 2.6	8.9 ± 2.4	46.4 ± 1.9	28.7 ± 6.1
	CADA-VAE (Schonfeld et al., 2019)	43.5 ± 2.0	$\underline{48.8 \pm 1.2}$	39.1 ± 6.8	22.0 ± 1.7	63.1 ± 0.8	45.1 ± 4.3
	LPP (Wang et al., 2019)	$\underline{55.2 \pm 0.4}$	29.0 ± 0.6	$\underline{61.7 \pm 1.5}$	24.3 ± 1.2	$\underline{71.4 \pm 0.4}$	**69.2 ± 0.5**
	CCVAE	**58.6 ± 0.3**	**56.7 ± 0.4**	**74.1 ± 0.6**	**49.6 ± 0.9**	**77.9 ± 0.3**	$\underline{66.3 \pm 0.9}$
Colour → *Neg*.	Source Only	85.5 ± 0.2	87.1 ± 0.2	84.8 ± 0.1	79.0 ± 0.2	88.6 ± 0.3	79.7 ± 0.4
	Baseline (1NN)	85.5 ± 0.6	88.6 ± 0.3	90.8 ± 0.1	84.3 ± 0.2	90.8 ± 0.1	81.8 ± 0.6
	Baseline (NN)	$\underline{89.3 \pm 0.0}$	$\underline{90.2 \pm 0.0}$	**92.7 ± 0.0**	**87.4 ± 0.0**	$\underline{91.1 \pm 0.0}$	$\underline{86.0 \pm 0.0}$
	BiDiLEL (Wang & Chen, 2017b)	33.2 ± 1.5	29.3 ± 1.3	13.1 ± 4.1	11.9 ± 2.0	52.3 ± 4.3	35.3 ± 3.2
	CADA-VAE (Schonfeld et al., 2019)	36.7 ± 3.6	43.5 ± 2.1	41.6 ± 4.5	19.5 ± 2.3	66.4 ± 1.2	43.0 ± 7.6
	LPP (Wang et al., 2019)	87.0 ± 0.1	82.1 ± 0.2	90.7 ± 0.1	80.1 ± 0.3	86.6 ± 0.4	81.5 ± 0.4
	CCVAE	**89.4 ± 0.0**	**90.3 ± 0.0**	$\underline{92.3 \pm 0.0}$	$\underline{86.8 \pm 0.1}$	**92.3 ± 0.0**	**86.1 ± 0.1**
Neg. → *Grey*	Source Only	33.1 ± 1.0	0.0 ± 0.0	30.1 ± 2.0	12.7 ± 1.2	0.0 ± 0.0	16.2 ± 0.8
	Baseline (1NN)	41.8 ± 1.2	25.4 ± 0.5	63.9 ± 0.9	28.4 ± 0.8	67.4 ± 0.6	$\underline{67.2 \pm 0.8}$
	Baseline (NN)	38.7 ± 0.1	33.3 ± 0.1	63.7 ± 0.4	$\underline{38.2 \pm 0.1}$	51.0 ± 0.0	60.9 ± 0.1
	BiDiLEL (Wang & Chen, 2017b)	28.7 ± 1.0	29.2 ± 2.4	22.5 ± 5.1	13.9 ± 1.0	53.3 ± 2.7	37.0 ± 2.4
	CADA-VAE (Schonfeld et al., 2019)	44.1 ± 3.0	$\underline{47.6 \pm 0.7}$	44.5 ± 3.2	24.9 ± 3.5	62.7 ± 1.2	43.2 ± 1.8
	LPP (Wang et al., 2019)	$\underline{52.9 \pm 0.9}$	30.6 ± 1.0	$\underline{65.1 \pm 1.8}$	33.4 ± 0.4	$\underline{68.9 \pm 0.4}$	**68.4 ± 0.3**
	CCVAE	**55.2 ± 0.2**	**57.4 ± 0.7**	**74.9 ± 0.4**	**52.6 ± 1.1**	**76.9 ± 0.4**	66.4 ± 1.1

Only method can achieve very decent performance on all 6 tasks and particularly on tasks between the *D* and *W* domains where it achieves the best performance. The ZSL/GZSL methods, however, perform significantly worse than others due to the information loss when compressing the sample-level source domain data into class-level semantic representations. These results demonstrate our proposed CCVAE performs consistently well regardless of the extent of domain shift.

The experimental results on the Office–Home dataset (Table 4) show a similar phenomenon observed on the BaggageXray dataset. CCVAE outperforms other comparative methods on 10 out of 12 tasks in terms of Acc_{unseen}. In terms of H, CCVAE performs the best on 6 tasks but is inferior to LPP (Wang et al., 2019) on the other 6 tasks due to the lower accuracy of seen classes. These results demonstrate CCVAE has an advantage in recognizing unseen classes which is beneficial in cases with large-scale unseen classes. On the other hand, CCVAE as a neural network is more tractable when training with a large number of training samples whilst LPP suffers from the computational complexity of eigenvalue decomposition (Wang et al., 2019).

In Table 5, H values are reported for 36 sub-tasks for XMNIST datasets. The tasks of *Colour* → *Grey* and *Colour* → *Neg* are relatively easy so that even using source data only can achieve as good performance as other more advanced methods except for the ZSL methods which, again, suffer from the issue of overfitting to seen classes hence result in low H values. In terms of the other four tasks (i.e. *Grey* → *Colour*, *Neg* → *Colour*, *Grey* → *Neg* and *Neg* → *Grey*) reported in Table 5, our proposed CCVAE significantly outperforms the comparative methods in most (20 out of 24) sub-tasks especially when there are more unseen classes (i.e. EMNIST with 26 classes serving as the unseen dataset).

To give a closer look into the accuracy of seen and unseen classes on the XMNIST datasets, we expand the first two columns in Table 5 and report the details of Acc_{seen} and Acc_{unseen} in Table 6. The performance difference among variant methods mainly relies on the varying capabilities of recognizing unseen classes whilst no significant difference exists in the accuracy of seen classes (Acc_{seen}) except *Source Only* and *CADA-VAE*. This is expected since the feature extraction models are trained with seen class data from both source and target domains except those used by *Source Only*.

In summary, our proposed CCVAE can handle the GZSDA problem effectively in varying settings across different datasets and outperforms contemporary methods consistently and more significantly in the most challenging scenarios.

Table 6
Detailed experimental results with MNIST as the seen dataset and FMNIST/EMNIST as the unseen dataset (mean and standard deviation of Acc_{seen}, Acc_{unseen} and H over five trials are reported).

Domains	Method	Seen: MNIST; Unseen: FMNIST			Seen: MNIST; Unseen: EMNIST		
		Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H
Grey → Colour	Source Only	4.4 ± 0.1	20.6 ± 1.0	7.3 ± 0.1	1.4 ± 0.0	14.3 ± 1.6	2.5 ± 0.1
	Baseline (1NN)	94.4 ± 0.1	30.7 ± 0.5	46.3 ± 0.6	94.0 ± 0.1	18.0 ± 0.3	30.2 ± 0.4
	Baseline (NN)	**95.6 ± 0.0**	33.8 ± 0.1	50.0 ± 0.1	**95.7 ± 0.0**	22.4 ± 0.1	36.2 ± 0.1
	BiDiLEL (Wang & Chen, 2017b)	92.6 ± 0.2	25.5 ± 1.7	39.9 ± 2.0	89.7 ± 0.4	22.4 ± 0.9	35.8 ± 1.1
	CADA-VAE (Schonfeld et al., 2019)	62.0 ± 2.9	28.8 ± 2.5	39.2 ± 1.9	52.1 ± 3.4	27.0 ± 1.2	35.5 ± 1.3
	LPP (Wang et al., 2019)	93.7 ± 0.1	$\underline{45.5 \pm 0.5}$	$\underline{61.3 \pm 0.5}$	94.3 ± 0.1	$\underline{28.2 \pm 0.5}$	$\underline{43.5 \pm 0.6}$
	CCVAE	$\underline{95.2 \pm 0.0}$	**48.1 ± 0.2**	**63.9 ± 0.2**	$\underline{94.9 \pm 0.0}$	**45.4 ± 0.1**	**61.4 ± 0.0**
Colour → Grey	Source Only	96.9 ± 0.2	78.2 ± 0.9	86.5 ± 0.6	94.4 ± 0.5	**84.2 ± 0.5**	89.0 ± 0.3
	Baseline (1NN)	**98.3 ± 0.1**	75.9 ± 0.8	85.6 ± 0.5	**98.4 ± 0.0**	79.0 ± 0.4	87.6 ± 0.3
	Baseline (NN)	$\underline{98.2 \pm 0.0}$	**82.4 ± 0.0**	**89.6 ± 0.0**	**98.4 ± 0.0**	81.4 ± 0.1	$\underline{89.1 \pm 0.0}$
	BiDiLEL (Wang & Chen, 2017b)	97.4 ± 0.1	17.0 ± 1.8	29.0 ± 2.6	96.8 ± 0.3	19.0 ± 1.5	31.8 ± 2.1
	CADA-VAE (Schonfeld et al., 2019)	82.9 ± 1.8	25.7 ± 2.0	39.1 ± 2.2	67.5 ± 3.4	27.5 ± 3.2	38.9 ± 2.7
	LPP (Wang et al., 2019)	97.3 ± 0.1	78.0 ± 0.3	86.6 ± 0.2	97.5 ± 0.0	68.3 ± 0.3	80.3 ± 0.2
	CCVAE	98.0 ± 0.0	$\underline{80.8 \pm 0.1}$	$\underline{88.6 \pm 0.1}$	$\underline{98.0 \pm 0.0}$	$\underline{83.5 \pm 0.1}$	**90.2 ± 0.1**
Neg. → Colour	Source Only	2.9 ± 0.2	21.1 ± 1.4	5.1 ± 0.3	1.0 ± 0.1	9.7 ± 1.5	1.9 ± 0.1
	Baseline (1NN)	94.4 ± 0.1	26.5 ± 0.7	41.4 ± 0.9	93.9 ± 0.2	17.0 ± 0.5	28.8 ± 0.6
	Baseline (NN)	**95.5 ± 0.1**	33.6 ± 0.1	49.7 ± 0.1	**95.3 ± 0.0**	22.7 ± 0.1	36.7 ± 0.2
	BiDiLEL (Wang & Chen, 2017b)	92.9 ± 0.2	22.8 ± 2.0	36.6 ± 2.6	89.7 ± 0.1	18.9 ± 0.8	31.3 ± 1.1
	CADA-VAE (Schonfeld et al., 2019)	68.7 ± 0.7	27.5 ± 1.6	39.3 ± 1.7	43.9 ± 4.4	28.6 ± 1.2	34.5 ± 1.8
	LPP (Wang et al., 2019)	93.5 ± 0.1	$\underline{53.6 \pm 0.8}$	$\underline{68.1 \pm 0.6}$	93.9 ± 0.1	$\underline{29.7 \pm 0.5}$	$\underline{45.2 \pm 0.6}$
	CCVAE	$\underline{95.3 \pm 0.0}$	**56.1 ± 0.5**	**70.6 ± 0.4**	$\underline{94.9 \pm 0.1}$	**47.3 ± 0.5**	**63.2 ± 0.5**
Grey → Neg.	Source Only	21.4 ± 0.9	29.3 ± 2.6	24.7 ± 1.4	0.0 ± 0.0	10.4 ± 1.0	0.0 ± 0.0
	Baseline (1NN)	**98.3 ± 0.1**	25.0 ± 1.1	39.9 ± 1.4	**98.5 ± 0.1**	11.4 ± 0.3	20.4 ± 0.5
	Baseline (NN)	$\underline{98.2 \pm 0.0}$	23.1 ± 0.1	37.3 ± 0.1	$\underline{98.4 \pm 0.0}$	20.9 ± 0.1	34.5 ± 0.2
	BiDiLEL (Wang & Chen, 2017b)	97.4 ± 0.1	14.2 ± 2.5	24.7 ± 3.9	96.1 ± 0.4	21.2 ± 1.6	34.7 ± 2.2
	CADA-VAE (Schonfeld et al., 2019)	88.5 ± 2.6	28.9 ± 2.0	43.5 ± 2.0	64.8 ± 1.2	**39.2 ± 1.5**	$\underline{48.8 \pm 1.2}$
	LPP (Wang et al., 2019)	97.2 ± 0.1	$\underline{38.6 \pm 0.4}$	$\underline{55.2 \pm 0.4}$	97.5 ± 0.0	17.1 ± 0.4	29.0 ± 0.6
	CCVAE	98.1 ± 0.0	**41.4 ± 0.3**	**58.2 ± 0.3**	98.0 ± 0.0	$\underline{38.9 \pm 0.5}$	**55.7 ± 0.5**
Colour → Neg.	Source Only	97.0 ± 0.2	78.4 ± 1.0	86.7 ± 0.6	94.8 ± 0.4	82.9 ± 0.4	88.4 ± 0.3
	Baseline (1NN)	**98.4 ± 0.1**	75.6 ± 0.9	85.5 ± 0.6	$\underline{98.2 \pm 0.1}$	80.6 ± 0.5	88.6 ± 0.3
	Baseline (NN)	**98.4 ± 0.0**	$\underline{81.8 \pm 0.0}$	$\underline{89.3 \pm 0.0}$	**98.5 ± 0.0**	$\underline{83.2 \pm 0.0}$	$\underline{90.2 \pm 0.0}$
	BiDiLEL (Wang & Chen, 2017b)	97.5 ± 0.0	20.0 ± 1.0	33.2 ± 1.5	96.8 ± 0.2	17.3 ± 0.9	29.3 ± 1.3
	CADA-VAE (Schonfeld et al., 2019)	86.8 ± 0.7	23.3 ± 2.8	36.7 ± 3.6	69.8 ± 1.5	31.7 ± 2.4	43.5 ± 2.1
	LPP (Wang et al., 2019)	97.3 ± 0.1	78.7 ± 0.2	87.0 ± 0.1	97.4 ± 0.1	71.0 ± 0.3	82.1 ± 0.2
	CCVAE	$\underline{98.2 \pm 0.0}$	**82.6 ± 0.1**	**89.7 ± 0.0**	98.0 ± 0.1	**84.3 ± 0.0**	**90.7 ± 0.0**
Neg. → Grey	Source Only	32.7 ± 1.0	30.3 ± 2.0	31.4 ± 1.1	0.0 ± 0.0	14.0 ± 1.5	0.0 ± 0.0
	Baseline (1NN)	**98.2 ± 0.2**	26.5 ± 0.9	41.8 ± 1.2	**98.3 ± 0.1**	14.6 ± 0.3	25.4 ± 0.5
	Baseline (NN)	$\underline{98.1 \pm 0.0}$	24.1 ± 0.1	38.7 ± 0.1	$\underline{98.2 \pm 0.0}$	20.0 ± 0.1	33.3 ± 0.1
	BiDiLEL (Wang & Chen, 2017b)	97.3 ± 0.1	16.9 ± 0.7	28.7 ± 1.0	96.4 ± 0.3	17.3 ± 1.7	29.2 ± 2.4
	CADA-VAE (Schonfeld et al., 2019)	92.9 ± 1.0	29.0 ± 2.6	44.1 ± 3.0	64.8 ± 3.6	$\underline{37.7 \pm 1.4}$	$\underline{47.6 \pm 0.7}$
	LPP (Wang et al., 2019)	97.2 ± 0.1	$\underline{36.3 \pm 0.8}$	$\underline{52.9 \pm 0.9}$	97.4 ± 0.1	18.1 ± 0.7	30.6 ± 1.0
	CCVAE	97.9 ± 0.0	**38.4 ± 0.2**	**55.2 ± 0.3**	97.8 ± 0.0	**40.0 ± 0.7**	**56.8 ± 0.8**

4.4. Ablation study

To investigate the contribution of different terms in the loss function Eq. (3), we conduct an ablation study by removing different components from the complete loss function. The ablation study is carried out and evaluated on the BaggageXray-20 dataset. Experimental results are shown in Table 7. When only the reconstruction loss $\mathcal{L}_{recon}$ is used, the model is equivalent to an Autoencoder which gives a recognition accuracy of 8.7% for unseen classes (i.e. $Acc_{unseen} = 8.7\%$) which is slightly higher than the random guess (i.e. 5%) in the *Regular* → *Xray* adaptation task. Similarly, in the *Xray* → *Regular* task, such a setting gives the lowest accuracy of 21.4% overall the unseen classes. Adding the KL divergence as a regularization term, the VAE model improves the average H over two adaptation tasks from 24.3% to 45.8%. The introduction of $\mathcal{L}_{cross\_recon}$ can significantly improve the performance in recognizing unseen classes as well as the harmonic mean H. Using the cross-domain reconstruction loss $\mathcal{L}_{cross\_recon}$ solely can achieve fairly good performance although the complete form of the loss function (3) achieves the best average H of 54.1%. These ablation study results demonstrate our model gains the capabilities of transforming features across domains by adding the cross-domain reconstruction loss to the conventional VAE framework.

A second ablation study is conducted to evaluate how different sets of data contribute to classifier training and recognition accuracy in step 3 of the proposed framework. In the GZSDA problem, we have access to the labelled real data sets $\mathcal{D}^s$ and $\mathcal{D}^t$ from the source domain and the target domain respectively. By leveraging the CCVAE model trained in step 2 (c.f. Fig. 2), we can generate synthetic data $\tilde{\mathcal{D}}^{st}$ from the source-domain data set $\mathcal{D}^s$ and synthetic data set $\tilde{\mathcal{D}}^{ts}$ from the target-domain data set $\mathcal{D}^t$. In our experiments, we investigate different combinations of these four sets of data for the classifier training and the results are shown in Table 8. When $\tilde{\mathcal{D}}^{st}$ is not used for classifier training (the first two rows of Table 8), there is no training data belonging to unseen classes from the target domain. As expected, the target-domain test samples belonging to the unseen classes are mostly mistakenly classified as seen classes hence leading to very low Acc_{unseen} and high Acc_{seen}. Adding the synthetic data set $\tilde{\mathcal{D}}^{st}$ generated by our trained CCVAE can improve the performance significantly (e.g., the 2nd row vs the 4th row of Table 8). This demonstrates our trained CCVAE can effectively transform the source-domain data $\boldsymbol{x}^s$ to synthetic target-domain data $\tilde{\boldsymbol{x}}^{st}$ for both seen and unseen classes. Note that during the training of CCVAE, there exist only labelled target-domain data belonging to the seen classes and a lack of target-domain data for the unseen classes. CCVAE can learn the cross-domain relation from the seen

Table 7
Ablation study results (%) using different combinations of loss terms on the BaggageXray dataset with 10 unseen classes (mean and standard error of the mean (SEM) over five different seen/unseen class splits are reported).

Method			Regular → Xray			Xray → Regular			Average
$\mathcal{L}_{recon}$	$\mathcal{L}_{cross\_recon}$	KLD	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H	H
✓	✗	✗	**81.1** ± **2.7**	8.7 ± 4.3	14.1 ± 5.9	**95.0** ± **0.8**	21.4 ± 3.2	34.4 ± 4.5	24.3
✓	✗	✓	70.9 ± 2.4	21.9 ± 2.5	33.1 ± 3.0	87.1 ± 1.2	44.6 ± 4.2	58.5 ± 3.9	45.8
✗	✓	✗	74.1 ± 2.1	27.2 ± 1.5	39.6 ± 1.7	88.4 ± 0.7	52.6 ± 2.0	65.8 ± 1.5	52.7
✓	✓	✗	79.9 ± 1.8	23.4 ± 2.8	35.8 ± 3.5	89.8 ± 0.5	**55.0** ± **3.4**	**67.9** ± **2.7**	51.9
✓	✓	✓	77.2 ± 1.9	**29.0** ± **1.5**	**42.1** ± **1.8**	90.8 ± 1.0	52.3 ± 2.5	66.1 ± 1.9	**54.1**

Table 8
Ablation study results (%) using different combinations of data to train the classifier (step 3 in Fig. 2) on the BaggageXray-20 dataset with 10 unseen classes (mean and standard error of the mean (SEM) over five different seen/unseen class splits are reported).

Training data				Regular → Xray			Xray → Regular			Average
$\mathcal{D}^s$	$\mathcal{D}^t$	$\tilde{\mathcal{D}}^{st}$	$\tilde{\mathcal{D}}^{ts}$	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H	H
✓	✓	✗	✗	**84.3** ± **1.9**	2.5 ± 0.4	4.8 ± 0.7	**95.0** ± **0.8**	20.4 ± 3.8	32.6 ± 5.5	18.7
✓	✓	✗	✓	83.6 ± 1.9	2.3 ± 0.3	4.4 ± 0.6	94.5 ± 1.0	18.9 ± 3.5	30.8 ± 5.0	17.6
✗	✗	✓	✗	60.0 ± 3.9	24.3 ± 0.7	34.4 ± 1.1	85.1 ± 2.4	32.1 ± 3.0	46.2 ± 3.0	40.3
✓	✓	✓	✗	79.2 ± 2.1	26.9 ± 1.7	40.0 ± 2.1	92.0 ± 1.0	50.7 ± 2.5	65.1 ± 2.0	52.6
✓	✓	✓	✓	77.2 ± 1.9	**29.0** ± **1.5**	**42.1** ± **1.8**	90.8 ± 1.0	**52.3** ± **2.5**	**66.1** ± **1.9**	**54.1**

Table 9
Ablation study results (%) with varying parameters of the latent code distribution on the BaggageXray dataset with 10 unseen classes (mean and standard error of the mean (SEM) over five different seen/unseen class splits are reported).

Parameter		Regular → Xray			Xray → Regular			Average
μ	σ^2	Acc_{seen}	Acc_{unseen}	H	Acc_{seen}	Acc_{unseen}	H	H
0.0	1.0	77.2 ± 1.9	29.0 ± 1.5	42.1 ± 1.8	90.8 ± 1.0	52.3 ± 2.5	66.1 ± 1.9	54.1
0.1	1.0	77.8 ± 1.9	29.7 ± 1.7	42.9 ± 1.8	90.9 ± 0.8	51.2 ± 2.5	65.3 ± 2.0	54.1
1.0	1.0	77.7 ± 1.8	29.3 ± 1.6	42.5 ± 1.9	90.3 ± 0.9	52.7 ± 2.4	66.3 ± 1.9	54.4
10.0	1.0	77.1 ± 1.8	30.3 ± 1.5	43.4 ± 1.7	90.1 ± 1.0	52.7 ± 2.6	66.3 ± 1.9	54.8
100.0	1.0	77.1 ± 1.9	29.9 ± 1.8	42.9 ± 1.9	90.7 ± 0.8	51.8 ± 2.6	65.7 ± 2.1	54.3
0.0	0.01	78.0 ± 1.9	27.5 ± 1.9	40.6 ± 2.1	90.1 ± 1.0	51.6 ± 2.6	65.4 ± 2.1	53.0
0.0	0.1	77.4 ± 1.8	29.1 ± 1.9	42.1 ± 2.1	90.7 ± 0.8	51.3 ± 2.4	65.3 ± 1.9	53.7
0.0	10.0	77.7 ± 2.0	28.9 ± 1.3	42.1 ± 1.6	90.4 ± 1.0	52.2 ± 2.5	65.9 ± 1.9	54.0
0.0	100.0	77.6 ± 2.0	29.2 ± 1.8	42.3 ± 2.1	90.3 ± 1.0	51.2 ± 2.7	65.1 ± 2.2	53.7

classes and generalize to the unseen classes hence enabling the generalized zero-shot domain adaptation. Using all four sets of data for the classifier training (last row in Table 8) achieves the best performance in Acc_{unseen} and H due to the additional synthetic data set $\tilde{\mathcal{D}}^{ts}$. In summary, the success of generalized zero-shot domain adaptation relies on the generation of effective synthetic target-domain data for unseen classes and our proposed CCVAE can realize this with training data from the source domain $\mathcal{D}^s$ and target-domain data $\mathcal{D}^t$ for seen classes only.

A third ablation study is conducted to investigate the effect of Gaussian distribution parameters for the latent code $\boldsymbol{z}$. Specifically, we replace the standard Normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ in Eq. (3) with a Normal distribution parameterized by varying $\boldsymbol{\mu}$ or varying $\boldsymbol{\sigma}^2$. We conduct the experiments on the BaggageXray-20 dataset and the results are shown in Table 9. By varying the normal distribution parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$, we do not observe significant performance change compared with the case where the standard normal distribution (i.e. the first row in Table 9) is employed.

5. Discussion and conclusion

The key to GZSDA is to overcome the overfitting to seen classes so that ZSDA methods such as (Wang & Jiang, 2019) and generic generative models such as CycleGAN (Zhu, Park, Isola, & Efros, 2017) do not apply. LPP achieves this goal by mapping the source and target data into a common subspace of lower dimension with a unified linear projection whilst the generative models (e.g., CADA-VAE and CCVAE) address the overfitting issue by generating synthetic data for the unseen classes.

Our proposed CCVAE was inspired by CADA-VAE and has a similar framework but is essentially different from it. Firstly, CADA-VAE generates features from class-wise attribute vectors, restricting the intra-class variations of the synthetic features whilst CCVAE generates features from individual samples. Secondly, CADA-VAE employs domain-specific VAE for source and target domains whilst CCVAE uses a unified VAE to promote the preserving of class information in the latent space. As a result, to generate both domain and class discriminative features, the generative model in CADA-VAE is conditioned on class information whilst CCVAE is conditioned on domain information. Finally, CCVAE is used to generate not only target-domain features but also source-domain features to augment the training data and a unified classifier is trained for both domains in CCVAE (Step 3).

In conclusion, our proposed CCVAE is an effective approach to the GZSDA problems. In addition, our proposed BaggageXray dataset provides a challenging testbed for future researches in GZSDA as well as other domain adaptation problems given the fact it arises from a real-world application in aviation security screening and the unique spectral X-ray imagery.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2010). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 898–916.

Blitzer, J., Foster, D. P., & Kakade, S. M. (2009). *Zero-shot domain adaptation: A multi-view approach*: *Tech. rep. TTI-TR-2009-1*.

Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., et al. (2019). Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 627–636).

Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017). EMNIST: Extending MNIST to handwritten letters. In *Proceedings of the international joint conference on neural networks* (pp. 2921–2926). IEEE.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 248–255). IEEE.

Deng, W., Zhao, L., Liao, Q., Guo, D., Kuang, G., Hu, D., et al. (2021). Informative feature disentanglement for unsupervised domain adaptation. *IEEE Transactions on Multimedia*.

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *Proceedings of the international conference on machine learning* (pp. 1180–1189).

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., et al. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677.

Guo, J., & Guo, S. (2020). A novel perspective to zero-shot learning: Towards an alignment of manifold structures via semantic feature expansion. *IEEE Transactions on Multimedia*, *23*, 524–537.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Ishii, M., Takenouchi, T., & Sugiyama, M. (2019). Zero-shot domain adaptation based on attribute information. In W. S. Lee, & T. Suzuki (Eds.), *Proceedings of machine learning research*: *vol. 101*, *Proceedings of the eleventh Asian conference on machine learning* (pp. 473–488). Nagoya, Japan: PMLR.

Jhoo, W. Y., & Heo, J.-P. (2021). Collaborative learning with disentangled features for zero-shot domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8896–8905).

Ji, Z., Wang, Q., Cui, B., Pang, Y., Cao, X., & Li, X. (2021). A semi-supervised zero-shot image classification method based on soft-target. *Neural Networks*, *143*, 88–96.

Ji, Z., Yan, J., Wang, Q., Pang, Y., & Li, X. (2021). Triple discriminator generative adversarial network for zero-shot image classification. *Science China. Information Sciences*, *64*(2), 1–14.

Ji, Z., Yu, X., Yu, Y., Pang, Y., & Zhang, Z. (2021). Semantic-guided class-imbalance learning model for zero-shot image classification. *IEEE Transactions on Cybernetics*.

Kim, Y., & Kim, C. (2021). Semi-supervised domain adaptation via selective pseudo labeling and progressive self-training. In *2020 25th international conference on pattern recognition* (pp. 1059–1066). IEEE.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. In *International conference on learning representations*.

Kouw, W. M., & Loog, M. (2019). A review of domain adaptation without target labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the advances in neural information processing systems* (pp. 1097–1105).

Kumagai, A., & Iwata, T. (2018). Zero-shot domain adaptation without domain semantic descriptors. arXiv preprint arXiv:1807.02927.

Kumar Verma, V., Arora, G., Mishra, A., & Rai, P. (2018). Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4281–4289).

Kutbi, M., Peng, K.-C., & Wu, Z. (2021). Zero-shot deep domain adaptation with common representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, X., Fang, M., & Chen, B. (2022). Generalized zero-shot domain adaptation with target unseen class prototype learning. *Neural Computing and Applications*, *34*(20), 17793–17807.

Long, M., Cao, Z., Wang, J., & Jordan, M. I. (2018). Conditional adversarial domain adaptation. In *Proceedings of advances in neural information processing systems* (pp. 1647–1657).

Ma, X., Zhang, T., & Xu, C. (2019). Deep multi-modality adversarial networks for unsupervised domain adaptation. *IEEE Transactions on Multimedia*, *21*(9), 2419–2431.

Mishra, A., Krishna Reddy, S., Mittal, A., & Murthy, H. A. (2018). A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of IEEE conference on computer vision and pattern recognition workshops* (pp. 2188–2196).

Motiian, S., Piccirilli, M., Adjeroh, D. A., & Doretto, G. (2017). Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision* (pp. 5715–5725).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the advances in neural information processing systems* (pp. 8024–8035).

Pei, Z., Cao, Z., Long, M., & Wang, J. (2018). Multi-adversarial domain adaptation. In *Proceedings of AAAI conference on artificial intelligence*.

Peng, K.-C., Wu, Z., & Ernst, J. (2018). Zero-shot deep domain adaptation. In *Proceedings of the European conference on computer vision* (pp. 764–781).

Pourpanah, F., Abdar, M., Luo, Y., Zhou, X., Wang, R., Lim, C. P., et al. (2022). A review of generalized zero-shot learning methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting visual category models to new domains. In *European conference on computer vision* (pp. 213–226). Springer.

Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., & 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* (pp. 8247–8255).

Sohn, K., Lee, H., & Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Proceedings of the advances in neural information processing systems* (pp. 3483–3491).

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In *International conference on artificial neural networks* (pp. 270–279). Springer.

Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5018–5027).

Wang, Q., & Breckon, T. P. (2020). Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. In *Proceedings of AAAI conference on artificial intelligence*.

Wang, Q., Bu, P., & Breckon, T. P. (2019). Unifying unsupervised domain adaptation and zero-shot visual recognition. In *Proceedings of international joint conference on neural networks* (pp. 1–8). IEEE.

Wang, Q., & Chen, K. (2017a). Alternative semantic representations for zero-shot human action recognition. In *Proceedings of joint European conference on machine learning and knowledge discovery in databases* (pp. 87–102). Springer.

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

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

Wang, J., Cheng, M.-M., & Jiang, J. (2021). Domain shift preservation for zero-shot domain adaptation. *IEEE Transactions on Image Processing*.

Wang, J., & Jiang, J. (2019). Conditional coupled generative adversarial networks for zero-shot domain adaptation. In *Proceedings of the IEEE international conference on computer vision* (pp. 3375–3384).

Wang, J., & Jiang, J. (2021). Learning across tasks for zero-shot domain adaptation from a single source domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wong, W. K., & Zhao, H. (2012). Supervised optimal locality preserving projection. *Pattern Recognition*, *45*(1), 186–197.

Xian, Y., Sharma, S., Schiele, B., & Akata, Z. (2019). f-VAEGAN-D2: A feature generating framework for any-shot learning. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 10275–10284).

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

Yan, X., Yang, J., Sohn, K., & Lee, H. (2016). Attribute2image: Conditional image generation from visual attributes. In *Proceedings of the European conference on computer vision* (pp. 776–791). Springer.

Yang, Y., & Hospedales, T. (2015). Zero-shot domain adaptation via kernel regression on the grassmannian. In *British machine vision conference*.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223–2232).