Unsupervised Domain Adaptation using Deep Networks with Cross-Grafted Stacks

Jinyong Hou[1], Xuejie Ding[2], Jeremiah D. Deng[1], Stephen Cranefield[1]
[1]Department of Information Science, University of Otago
[2]Institute of Information Engineering, Chinese Academy of Sciences
houjinyong@gmail.com,dingxuejie@iie.ac.cn, {jeremiah.deng,stephen.cranefield}@otago.ac.nz

Abstract

Current deep domain adaptation methods used in computer vision have mainly focused on learning discriminative and domain-invariant features across different domains. In this paper, we present a novel approach that bridges the domain gap by projecting the source and target domains into a common association space through an unsupervised "cross-grafted representation stacking" (CGRS) mechanism. Specifically, we construct variational auto-encoders (VAE) for the two domains, and form bidirectional associations by cross-grafting the VAEs' decoder stacks. Furthermore, generative adversarial networks (GAN) are employed for domain adaptation (DA), mapping the target domain data to the known label space of the source domain. The overall adaptation process hence consists of three phases: feature representation learning by VAEs, association generation, and association alignment by GANs. Experimental results demonstrate that our CGRS-DA approach outperforms the state-of-the-art on a number of unsupervised domain adaptation benchmarks.

1. Introduction

In machine learning, domain adaptation aims to transfer knowledge learned previously from one or more "source" tasks to a new but related "target" domain. As a special form of transfer learning, it helps to overcome the lack of labelled data in computer vision tasks by utilizing labelled data of the source domain and trying to automatically annotate unlabelled data in the target domain [26]. It may also be used to recognize unfamiliar objects in a dynamically changing environment in robotics. Therefore, in recent years domain adaptation, especially unsupervised domain adaptation, has become an appealing research topic [3, 2, 23, 11, 36, 30, 13].

For domain adaptation to occur, it is assumed that the source and target domains are located in the same label space, but there is a domain bias. The challenge is to extract the domain-invariant representations from the data, and find an effective mechanism to overcome the domain bias and map the unlabelled targets to the label space.

To address the challenge, we propose to recruit different levels of deep unsupervised receptive fields from both the source and target domains and construct grafted representations for domain adaptation. Our approach is inspired by UNIT [20], but we generate the cross-domain association differently, employing grafted deep network layers. Specifically, we construct two parallel variational auto-encoders (VAEs) [16] to extract the latent encodings of the source and target. Then we recruit the different parts of the decoders to construct some cross-grafted representation stacks (CGRS), which produces bi-directional cross associations between the two domains. Furthermore, generative adversarial networks (GANs) [10] are employed to carry out association alignment, so that associations between the source and target contribute to accurate classification.

Due to these treatments our proposed CGRS-DA framework gives a promising direction for domain adaptation. Building cross associations between the domains, feature learning is hence achieved across domains, reducing domain dependency and increasing domain-invariance, while adversarial networks further push feature representations away from the differences between domains, contributing to robust domain adaptation performance. Also, the cross-grafting process is entirely symmetric, leading to similar performance regardless of the adaptation direction, as revealed by our experiment results. Another advantage revealed by our experiments is that the CGRS is transferable across different tasks, which is an attractive trait for developing practical applications.

The rest of the paper is organized as follows. In Section 2, we will briefly review some related work. In Section 3, we outline the overall structure of our proposed model, introduce the CGRS scheme, and present the learning metrics used by the model. The experimental results are presented and discussed in Section 4. We conclude the paper in Section 5, indicating our plan of future work.

2. Related Work

There are existing works that utilize intermediate feature representations to transfer previously learned knowledge to the target tasks. Self-taught learning [27] uses unsupervised learning trained on natural images to construct a sparse coding space, to which targets are projected to complete the recognition. In geodesic flow kernel [9, 12], the source and target datasets are embedded in a Grassman manifold, and a geodesic flow is constructed between the domains. A number of feature subspaces are sampled along the geodesic flow, and a kernel can be defined on the incremental feature vector, allowing a classifier to be built for the target dataset. DLID [6] uses deep sparse learning to extract the interpolated representation from a set of intermediate datasets constructed by combining the source and target datasets using progressively varying proportions, and the features from these intermediates are concatenated to train a classifier.

Recent works have shown that deep networks involved in domain adaptation have achieved impressive performance due to their strong feature learning capacity. This provides a considerable improvement for some cross-domain recognition tasks [34, 22, 31, 24, 28, 20, 5, 8]. A number of deep domain adaptation models have applied the adversarial training strategy [32, 33, 7, 20, 4, 19, 21]. DANN [7] employs a gradient reversal layer between the feature layer and the domain discriminator, causing feature representation to anti-learn the domain difference and hence adapt well to the target domain. ADDA [32] firstly trains a convolutional based classifier using source dataset. And an additional features extractor is built for target. Then a discriminator is utilized to confuse the features extracted by features extractor of source and target. Finally the target encoder is combined with the source classifier to achieve the adaptation.

Using generative adversarial networks (GAN), the PixelDA framework [4] generates synthetic images from source-domain images that are mapped to the target domain. A task classifier then is trained by the source and synthetic images using the source labels. UNIT [20] introduces an unsupervised image-to-image translation framework based on couple of variational auto-encoders (VAEs) and GANs. To achieve this, a pair of corresponding images in different domains are mapped to a shared latent representation space.

Inspired by these previous works, our proposed CGRS-DA framework combines two ideas: constructing cross-domain feature representations, and employing adversarial networks for association alignment. Specifically, it incorporates VAEs to learn feature representations, a cross-grafting step to generate bidirectional cross-domain associations, and a generative adversarial approach that carries out classification on source-target associations. A detailed description of our framework is given next.

3. The CGRS-DA Framework

3.1. Model Description

For domain adaptation, we consider two domains: one is a source domain $\mathcal{D}_s$, which is constructed by n_s images $\mathbf{X}_s = \{\boldsymbol{x}_i^s\}_{i=1}^{n_s}$ and their correspond labels $\boldsymbol{y}_s = \{y_i^s\}_{i=1}^{n_s}$; the other is a target domain $\mathcal{D}_t = \{\mathbf{X}_t, \boldsymbol{y}_t\}$, where $\mathbf{X}_t = \{\boldsymbol{x}_i^t\}_{i=1}^{n_t}$ and their labels $\boldsymbol{y}_t = \{y_i^t\}_{i=1}^{n_t}$ are not available during adaptation. Our goal is to learn some representations bearing similarity to both domains, i.e. some joint distribution between source distribution $\mathcal{P}$ and target's $\mathcal{Q}$ as a bridge for the adaptation.

Our framework is shown in Figure 1, split into five modular sub-tasks based on the ideas outlined as above. Firstly, in module *A*, a pair of VAEs are implemented by CNNs. Both the encoders and decoders are divided into high and low level stacks. The high-level layers of the encoders are shared between domains. We assume that they have the same latent space with normal prior $\mathcal{N}(0, I)$.

Secondly, the latent encodings pass through the cross-grafted stacks, forming cross-domain associations that are aligned to the association space. In module *B*, we construct two parallel CGRS by grafting the decoder stacks of the source and the target. Therefore, the cross-domain association images $(\mathbf{X}_s^{st}, \mathbf{X}_t^{st}, \mathbf{X}_s^{ts}, \mathbf{X}_t^{ts})$ are generated when the latent encodings from different domains (indicated by subscripts) pass through the CGRS (order indicated by superscripts). In the domain alignment module *C*, G_1 and G_2 are two adversarial generators for associations. They are used to generate the target association adversarial to the source's association, and vice versa. The situation when the source associations works as the "real player" for the adversarial generation is shown in Figure 1[1]. Here the adversarials of the corresponding target associations are $\widetilde{\mathbf{X}}_t^{st}$, and $\widetilde{\mathbf{X}}_t^{ts}$. The discriminators D_1, D_2 are used to distinguish associations of $\mathbf{X}_s^{st}$ from $\widetilde{\mathbf{X}}_t^{st}$, and $\mathbf{X}_s^{ts}$ from $\widetilde{\mathbf{X}}_t^{ts}$ respectively.

Finally, L_G and L_T in modules *D* and *E* are the learning metrics for domain confusion and task classification. Module *C* combines the learning metric modules to align the label space of the source and target images, and complete the adaptation. The training process adopts standard backpropagation. In contrast to the conventional domain adaptation framework in which the classifier input is $\{\mathbf{X}_s, \boldsymbol{y}_s\}$ and output is $\{\mathbf{X}_t, \widehat{\boldsymbol{y}_t}\}$, our model's classifier is trained by $\{\mathbf{X}_s^{st}, \boldsymbol{y}_s\}$, $\{\mathbf{X}_s^{ts}, \boldsymbol{y}_s\}$ and tested by $\{\widetilde{\mathbf{X}}_t^{st}, \boldsymbol{y}_t\}$, $\{\widetilde{\mathbf{X}}_t^{ts}, \boldsymbol{y}_t\}$. In short, the associations of the source data are used for training, and the adversarial generation of the target data are used in testing.

[1] The arrangement can be flexible, i.e. it also works if the target association is used as the real player.

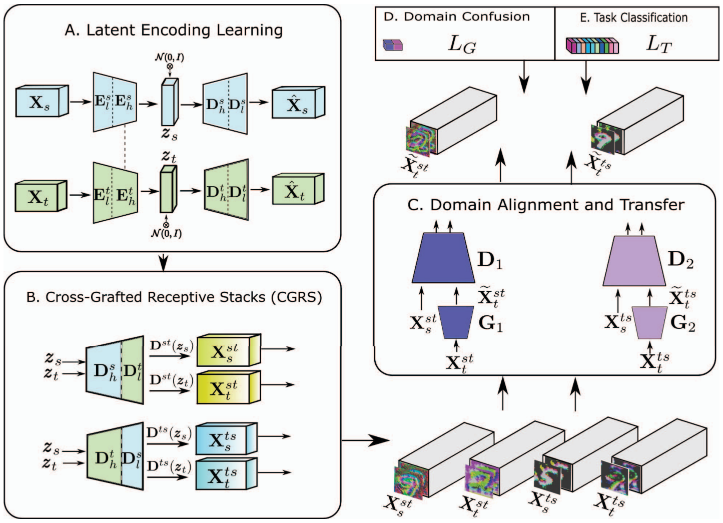


Figure 1: Overview of the the proposed model. There are 5 modules in it. In module *A*, the high-level layers of encoders E_h^s, E_h^t are shared (demonstrated by the dashed line). The outputs of D_h^s and D_h^t are the high-level representations of the source and target, whereas D_l^s, D_l^t are the low-level ones. The $\mathbf{X}_s^{st}$, $\mathbf{X}_t^{ts}$, $\mathbf{X}_s^{ts}$, $\mathbf{X}_t^{ts}$ in module *B* are the association images reproduced by CGRS ($D^{st} \equiv [D_h^s \circ D_l^t]$ and $D^{ts} \equiv [D_h^t \circ D_l^s]$) from latent encodings. In module *C*, G_1 and G_2 are adversarial generators, D_1, D_2 are discriminators. L_G and L_T are learning metrics for the domain and task respectively. Best viewed in colors.

3.2. Learning

To train our model, we jointly solve the learning problems of the subnetworks.

First, we learn the representations of the source and target domains from encoders and decoders. Here, we minimize the within-domain VAEs loss functions. The loss function of our VAEs consists of both reconstruction error L_{rec} and prior regularization L_{prior}:

$$L_{VAEs} = L_{rec} + L_{prior}, \tag{1}$$

given by

$$\begin{aligned} L_{rec} = -\lambda_1\{&\mathbb{E}_{q_s(\boldsymbol{z}_s|\boldsymbol{x}_s)}[\log p_s(\boldsymbol{x}_s|\boldsymbol{z}_s)] \\ &+ \mathbb{E}_{q_t(\boldsymbol{z}_t|\boldsymbol{x}_t)}[\log p_t(\boldsymbol{x}_t|\boldsymbol{z}_t)]\}, \end{aligned} \tag{2}$$

and

$$\begin{aligned} L_{prior} = \lambda_2\{&D_{KL}(q_s(\boldsymbol{z}_s|\boldsymbol{x}_s)||p(\boldsymbol{z})) \\ &+ D_{KL}(q_t(\boldsymbol{z}_t|\boldsymbol{x}_t)||p(\boldsymbol{z}))\}, \end{aligned} \tag{3}$$

where D_{KL} is the Kullback-Leibler divergence, λ_1 and λ_2 are the trade-off hyper-parameters to control the priorities of variational encoding and reconstruction.

To align the source and target domains, we use the adversarial training for the association spaces $\mathcal{P}_{st}$ and $\mathcal{P}_{ts}$. The adversarial objective of $\mathcal{P}_{st}$ is

$$\begin{aligned} L_G^{st} = \lambda_0\{&\mathbb{E}_{x_s,z_s}[\log D_1(\mathbf{X}_s^{st})] \\ &+ \mathbb{E}_{x_t,z_t}[\log(1 - D_1(G_1(\mathbf{X}_t^{st}))]\}, \end{aligned} \tag{4}$$

where $D_1(\cdot)$ is the probability function assigned by the discriminator network, which tries to distinguish the generated source-based associations by $G_1(\cdot)$ from the target-based ones. For $\mathcal{P}_{ts}$, the adversarial objective L_G^{ts} is similarly defined. At last, the overall adversarial generative cost function is:

$$L_G = L_G^{st} + L_G^{ts}. \tag{5}$$

For the training stability, we introduce a content similarity metric for the associations [4, 14]. Both the $L1$ and $L2$ penalty can be used to regularize the associations, such as MSE, pairwise MSE, and Huber loss. Here we simply use MSE. The MSE loss for associations $\mathbf{X}^{st}$ is given as follows:

$$L_s^{st} = \mathbb{E}_{\mathbf{X}^{st}}(||\mathbf{X}_s^{st} - \mathbf{X}_t^{st}||_2^2) \tag{6}$$

and for $\mathbf{X}^{ts}$, L_s^{ts} has a similar style. So the overall content objective for associations is:

$$L_s = \lambda_3(L_s^{st} + L_s^{ts}). \tag{7}$$

For classification, we use a typical soft-max cross-entropy loss:

$$L_T = \mathbb{E}[-\mathbf{y}_s^T \log T_s(\mathbf{X}_s^{st}) - \mathbf{y}_s^T \log T_s(\mathbf{X}_s^{ts})], \tag{8}$$

where $\mathbf{y}_s$ is the class label for source $\mathbf{X}_s$, and T_s is the task classifier. Finally, the overall loss function of our model is:

$$L^* = \min_{E,D,G} \max_{D_1,D_2} (L_{VAEs} + L_G + L_s + L_T). \tag{9}$$

We solve this minimax optimization by three alternating steps. First, the latent encodings are learned by the self-mapped process, which updates (E_s, E_t, D_s, D_t). Then,

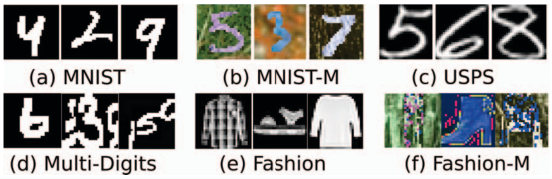


Figure 2: Examples of the Datasets used for Experiments.

we apply a gradient ascent step to update two discriminators D_1, D_2 and the classifier T. Finally, a gradient descent step is applied to update (E_s, E_t, G_1, G_2).

4. Experiments and Results

4.1. Datasets and Adaptation Scenarios

We use six popular datasets to construct four domain adaptation scenarios:

MNIST ⇄ MNIST-M: This is a scenario where the image content is the same, but the target data are polluted by noise. The MNIST handwritten dataset [18] has a training set of 60,000 binary images, and a test set of 10,000. There are 10 classes in the dataset. MNIST-M [7] is a modified version of MNIST, with random RGB backgrounds cropped from the Berkeley Segmentation Dataset[2]. In our experiments, we use the standard split of the dataset.

MNIST ⇄ USPS: For this scenario, source and target domains have different contents but the same background. USPS is a handwritten zip digits dataset [17]. It contains 9298 binary images (16 × 16), 7291 of which are used as the training set, while the remaining 2007 are used as the test set. The USPS samples are resized to 28 × 28, the same as MNIST.

Fashion ⇄ Fashion-M: Fashion-MNIST [35] contains 60,000 images for training, and 10,000 for testing. All the images are grayscale by 28 × 28 in size.The samples are collected from 10 fashion categories There are some complex textures in the images. In addition, following the protocol in [7], we add random noise to the Fashion images to generate the Fashion-M dataset.

MNIST ⇄ M-Digits In this scenario, we design a multi-digits dataset to evaluate the proposed model, denoted as M-Digits. The MNIST digits are cropped first, and then are randomly selected, combined and randomly aligned in a new image, limited to 3 digits maximum. The label for the new image is decided by the central digit. Finally, the new dataset is resized to 28 × 28.

4.2. Implementation Details

All the models are implemented using TensorFlow [1] and are trained with Mini-Batch Gradient Descent using the Adam optimizer [15]. The initial learning rate is 0.0002. Then it adopts an annealing method, with a decay of 0.95 after every 20,000 mini-batch steps. The mini-batch size for both the source and target domains is 64 samples, and the input images are rescaled to [-1, 1]. The hyper-parameters are $\lambda_0 = 1$, $\lambda_1 = 10$, $\lambda_2 = 0.01$, $\lambda_3 = 1$.

In our implementation, the latent space is sampled from a normal distribution $\mathcal{N}(0, I)$, and is achieved by the convolution encoders. The transpose convolution [37] is used in the decoder to build the reconstruction image space. A similar structure to that of [20] is used, but we modify the padding strategy to 'same' for convolution layers. For sake of convenience in experiments, we add another 32-kernel layer before the last layer in the decoders. The stride is 2 for down-sampling in the encoders, and their counterpart in decoders is also 2 so as to get the same dimensionality of the original image. The encoders for source and target domains share their high-level layers. We add batch normalization between each layer in the encoders and the decoders. The CGRS of associations is the composition of different levels of the source and target representations. The stride step is 1 for all the dimensions in the adversarial generator, and the kernel is 3 × 3. This adopts the structure of PixelDA [4], which uses a ResNet architecture. The discriminator confuses the domains, and also plays the role of a task classifier for the label space learning. It follows the design as of [20]. However, we do not share the layers of discriminators of $\mathbf{X}^{st}$ and $\mathbf{X}^{ts}$ channels. Also, we replace the max-pooling with a stride of 2 × 2 steps.

4.3. Results

4.3.1 Quantitative Results

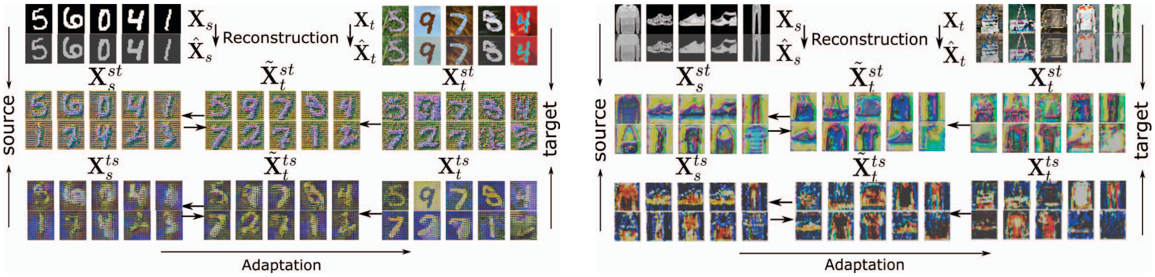
Now we report the classification performance of our proposed model. During the experiments, associations $\mathbf{X}^{st}_s$ and $\mathbf{X}^{ts}_t$ are used to train the classifier, and the adversarial generation of $\mathbf{X}^{st}_t$ and $\mathbf{X}^{ts}_t$ are used for testing. The accuracy of the target domain classification after domain adaptation is listed in Table 1, presenting the result of 8 methods (4 versions of our model, and 4 state-of-the-art methods) across 4 tasks (each in two directions). Our proposed model outperforms the state-of-the-art in most of the scenarios, especially when content similarity is considered (denoted by CGRS-DA). Also, it can be seen that the adaptation performance is usually asymmetric for the methods in comparison, e.g. the accuracies for MNIST→M-Digits and M-Digits→MNIST are quite different for DANN and PixelDA. The CGRS-DA models, however, perform almost equally well on both directions for these adaptation tasks.

For MNIST⇄MNIST-M and MNIST⇄USPS, the mean classification accuracy nearly reaches the upper bound, suggesting these are easier tasks. On the other hand, we can see the adaptation task between Fashion and Fashion-M is more difficult than others. For this task, our method again not only achieves the best performance but also demonstrates

[2] URL https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

Table 1: Mean classification accuracy comparison. The ”source only” row is the accuracy for target without domain adaptation training only on the source. And the ”target only” is the accuracy of the full adaptation training on the target. For each source-target task the best performance is in bold.

Source	MNIST	MNIST-M	MNIST	USPS	MNIST	M-Digits	Fashion	Fashion-M
Target	MNIST-M	MNIST	USPS	MNIST	M-Digits	MNIST	Fashion-M	Fashion
Source Only	0.561	0.633	0.634	0.625	0.603	0.651	0.527	0.612
GtA [29]	-	-	0.953	0.908	-	-	-	-
DANN [7]	0.766	0.851	0.774	0.833	0.864	0.920	0.604	0.822
PixelDA [4]	0.982	0.922	0.959	0.942	0.734	0.913	0.805	0.762
UNIT [20]	0.920	0.932	0.960	0.951	0.903	0.910	0.796	0.805
CGRS-DA-noC ($\mathbf{X}^{st}$)	0.821	0.935	0.946	0.938	0.895	0.902	0.735	0.805
CGRS-DA-noC ($\mathbf{X}^{ts}$)	0.923	0.840	0.902	0.930	0.853	0.851	0.792	0.760
CGRS-DA ($\mathbf{X}^{st}$)	0.890	**0.983**	**0.961**	0.956	**0.916**	**0.923**	0.766	**0.825**
CGRS-DA ($\mathbf{X}^{ts}$)	**0.983**	0.871	0.943	0.953	0.883	0.892	**0.813**	0.811
Target Only	0.983	0.985	0.980	0.985	0.982	0.985	0.920	0.942



(a) MNIST → MNIST-M

(b) Fashion → Fashion-M

Figure 3: The visualization of association generations. For each scenario, the leftmost column is the source and its association, and the rightmost is for target. During the experiments, the associations of source are real player. The adversarial generations for target associations are in the middle column.

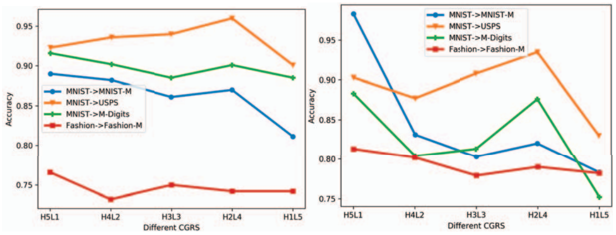
balanced performance in two directions.

4.3.2 Qualitative Results

Since our model adopts a generative approach, we can have direct visual evaluation of the associations generated by CGRS. The generative associations obtained by CGRS are shown in Figure 3, obtained after 100k mini-batch steps for the Fashion scenario and 50k for other three scenarios. CGRS generates the associations with very similar appearance for the source and target domains. Then the GANs are employed to move them closer. During association generation, CGRS eliminates the strong noise of MNIST-M and Fashion-M. Though there are more complex textures in the Fashion task, the proposed model still performs well to produce reasonable visualizations of the associations. The associations of the Fashion scenario seem to suffer some information loss, possibly due to the complex textures and strongly polluted images. However, they still look reasonable upon visualization.

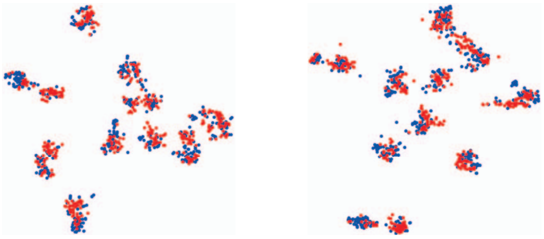
4.3.3 Model Analysis

Some further experiments are done to evaluate our model.
Ablation Study: To evaluate the potential effect of employing the content similarity strategy in our model, we conduct the adaptation tasks without content similarity, denoted by CGRS-DA-noC. From the Table 1, we can see that the model with content similarity outperforms its CRGS-DA-noC counterpart. This confirms the effectiveness of incorporating content similarity.
Sensitivity of CGRS: CGRS plays a critical role in the proposed model. We evaluated the performance of diverse structures of CGRS. During the experiments, we used a fixed depth of network (6 layers) for the generation process. We applied various settings for splitting the high-level and low-level decoder stacks. For example, H5L1 denotes the scheme using 5 layers for high-level and 1 layer as low-level. added between layers. The results of evaluation are shown in Figure 4. It can be seen that for the channel $\mathbf{X}^{st}$ in MNIST→MNIST-M and Fashion→Fashion-M tasks, the highest accuracies are at the point H5L1, and



(a) $\mathbf{X}^{st}$ channel (b) $\mathbf{X}^{ts}$ channel

Figure 4: The Adaptation Accuracy of Different CGRS.



(a) $\mathbf{X}^{st}$ channel (b) $\mathbf{X}^{ts}$ channel

Figure 5: The visualization of top association features embedded by t-SNE w.r.t. source and target. The blue dots are for source patterns and red ones for target patterns.

for MNIST→USPS and MNIST→M-Digits tasks, there is a peak value at the point H2L4. The $\mathbf{X}^{ts}$ channel somehow seems more sensitive to varying CGRS settings.

Generalization of CGRS: Can we utilize the trained CGRS in one scenario to another adaptation task? In this evaluation, we use our pre-trained CGRS from one scenario to adapt to a different task. These models are trained with a trade-off H4L2 CGRS according to the sensitivity analysis. During the experiments, we kept the CGRS fixed, then fine-tuned the associations adversarial alignment parts. The performance of adaptation to other tasks remains reasonable. For example, the CGRS trained by task Fashion→Fashion-M can get the performance of 0.983, 0.958, 0.915 for MNSIT→MNIST-M, MNIST→USPS, and MNIST→M-Digits respectively.

Visualization of Extracted Features: We also evaluated the features of the top, fully connected layers in the discriminator for task MNIST→USPS. The features were embedded in Euclidean space by the t-SNE algorithm [25]. Figure 5 shows that the two domains can be aligned well on both channels after adaptation.

4.3.4 Discussion

To sum up, our method can maintain stable performance when we vary the settings of CGRS for stack splitting. There seems to be a tendency to favour a higher ratio of high-level to low-level layers when the domains contain similar contents but different background, while adaptation tasks with similar background but different content favour more low-level layers.

Another interesting observation is that CGRS representations have very good generalization ability. The CGRS trained by one task can be employed for domain adaptation in another task. This demonstrates a merit of our method for practical applications, that is the CGRS representations are transferable.

Finally, while the both association channels are well aligned, from our experiment it seems $\mathbf{X}^{ts}$ claims better classification performance more often. In practical applications, it may be possible to design a classification combination method so that an optimal final decision can be developed from both association channels.

5. Conclusion

In this paper, we have proposed a novel unsupervised domain adaptation model based on cross-domain association generation, and label alignment using adversarial networks. In particular, cross-grafted representation stacks between different domains are constructed for bi-directional associations. The domain adaptation task hence is transformed to construct an effective mapping of the cross-domain associations onto the label space of the original source domain, a methodology we believe contributes to its robust performance in domain adaptation tasks. This is verified by the empirical results we have obtained from a number of tasks involving 6 benchmark tasks, which also demonstrate that the proposed CGRS models has strong cross-task generalization abilities. For future work, we would like to explore the extension of the framework for continual learning with cross-task adaptation.

Acknowledgment This work is supported by the University of Otago Research Grant, and the Natural Science Foundation of China (No.61601459).

References

[1] M. Abadi et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, 2016. 1603.04467. 4

[2] T. Adel and A. Wong. A probabilistic covariate shift assumption for domain adaptation. In *Proceeding of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2476–2482, 2015. 1

[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010. 1

[4] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, 2017. 2, 3, 4, 5

[5] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 343–351, 2016. 2

[6] S. Chopra, S. Balakrishnan, and R. Gopalan. Dlid: Deep learning for domain adaptation by interpolating between domains. ICML Workshop on Challenges in Representation Learning (WREPL), 2013. 2

[7] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research (JMLR)*, 17(1):2096–2030, 2016. 2, 4, 5

[8] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 597–613. Springer, 2016. 2

[9] B. Gong, K. Grauman, and F. Sha. Geodesic flow kernel and landmarks: Kernel methods for unsupervised domain adaptation. In G. Csurka, editor, *Domain Adaptation in Computer Vision Applications.*, Advances in Computer Vision and Pattern Recognition, pages 59–79. Springer, 2017. 2

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems (NeurIPS)*, pages 2672–2680, 2014. 1

[11] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proceeding of the IEEE International Conference on Computer Vision (ICCV)*, pages 999–1006, 2011. 1

[12] R. Gopalan, R. Li, and R. Chellappa. Unsupervised adaptation across domain shifts by generating intermediate data representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(11):2288–2302, 2014. 2

[13] S. Herath, M. T. Harandi, and F. Porikli. Learning an invariant hilbert space for domain adaptation. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3956–3965, 2017. 1

[14] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, (ICML)*, volume 70 of *Proc. of Machine Learning Research*, pages 1857–1865, 2017. 3

[15] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 4

[16] D. P. Kingma and M. Welling. Auto-encoding Variational-Bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[17] Y. Le Cun, L. Jackel, B. Boser, J. Denker, H. Graf, I. Guyon, D. Henderson, R. Howard, and W. Hubbard. Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine*, 27(11):41–46, 1989. 4

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4

[19] M. Liu and O. Tuzel. Coupled generative adversarial networks. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 469–477, 2016. 2

[20] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 700–708, 2017. 1, 2, 4, 5

[21] Y.-C. Liu, Y.-Y. Yeh, T.-C. Fu, S.-D. Wang, W.-C. Chiu, and Y.-C. F. Wang. Detach and adapt: Learning cross-domain disentangled deep representation. *arXiv preprint arXiv:1705.01314*, 2017. 2

[22] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015. 2

[23] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer joint matching for unsupervised domain adaptation. In *Proceeding of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1410–1417, 2014. 1

[24] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 136–144, 2016. 2

[25] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research (JMLR)*, 9:2579–2605, Nov 2008. 6

[26] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering (TKDE)*, 22(10):1345–1359, 2010. 1

[27] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 227, pages 759–766, 2007. 2

[28] A. Rozantsev, M. Salzmann, and P. Fua. Beyond sharing weights for deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 2

[29] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8503–8512, 2018. 5

[30] O. Sener, H. O. Song, A. Saxena, and S. Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2110–2118, 2016. 1

[31] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceeding of the IEEE International Conference on Computer Vision, (ICCV)*, pages 4068–4076, 2015. 2

[32] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017. 2

[33] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2

[34] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5018–5027, 2017. 2

[35] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. 4

[36] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems (NeurIPS)*, pages 3320–3328, 2014. 1

[37] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 2528–2535, 2010. 4