Improve conditional adversarial domain adaptation using self-training

Zi Wang1,2 | Xiaoliang Sun1,2 | Ang Su1,2 | Gang Wang3 | Yang Li1,2 | Qifeng Yu1,2

1 College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, People’s Republic of China
2 Hunan Provincial Key Laboratory of Image Measurement and Vision Navigation, Changsha, People’s Republic of China
3 National Key Laboratory of Human Factor Engineering, China Astronaut Research and Training Centre, Beijing, People’s Republic of China

Correspondence
Xiaoliang Sun, College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, People’s Republic of China.
Email: alexander_sxl@nudt.edu.cn

Funding information
Postgraduate Scientific Research Innovation Project of Hunan Province, Grant/Award Numbers: CX20200024, CX20200025, CX20200088; Natural Science Foundation of Hunan Province, Grant/Award Numbers: 2019JJ50732, 2019JJ50739; National Natural Science Foundation of China, Grant/Award Numbers: 11902349, 61801491

Abstract
Domain adaptation for image classification is one of the most fundamental transfer learning tasks and a promising solution to overcome the annotation burden. Existing deep adversarial domain adaptation approaches imply minimax optimization algorithms, matching the global features across domains. However, the information conveyed in unlabelled target samples is not fully exploited. Here, adversarial learning and self-training are unified in an objective function, where the neural network parameters and the pseudo-labels of target samples are jointly optimized. The model’s predictions on unlabelled samples are leveraged to pseudo-label target samples. The training procedure consists of two alternating steps. The first one is to train the network, while the second is to generate pseudo-labels, and the loop continues. The proposed method achieves mean accuracy improvements of 2% on Office-31, 0.7% on ImageCLEF-DA, 1.8% on Office-Home, and 1.2% on Digits than the baseline, which is superior to most state-of-the-art approaches.

1 | INTRODUCTION

Recently, deep convolutional neural networks (CNN) have achieved many inspiring results in computer vision tasks (e.g. image classification [1], object detection [2]) and have benefited numerous real-world applications. However, CNNs have a considerable number of parameters and can be severely data hungry, which means collecting and labelling large-scale datasets can be time consuming and expensive. The simple idea is to learn transferable knowledge from a small set of labelled samples and many unlabelled samples, then deploy the learned model into the target application. As pointed out in [3], the main challenge is the model performance degradation because of the domain shift between the two datasets.

The basic assumption of machine learning theory is that the training and test data are independent and identically distributed (i.i.d), which is usually violated in real-world deployment due to many factors and can degrade the testing performance. As a particular case of transfer learning, unsupervised domain adaptation (UDA) relaxes the i.i.d constraint and theoretically bounds the generalization error [4, 5]. Given labelled source data and unlabelled target data, the task of UDA is to bridge the gap between the two domains. One of the basic principles of UDA is to align the feature representations extracted from both domains. In the CNN era, adversarial training is a promising approach, where a two-player game strategy is adopted to extract domain-invariant feature representation.

However, there are multimodal distributions that exist in classification problems natively and widely. Aligning feature representations extracted from different domains is a challenging task for deep adversarial domain adaptation approaches. To tackle this problem, Conditional Domain Adversarial Network (CDAN) [6] and its variants condition the feature representation on the classifier prediction to capture the multimodal
structures and have achieved significant results. However, we observe that the classifier is supervised by labelled samples in the source domain alone, and the generalization error is only bounded by feature alignment. Target samples may carry information that is useful for label prediction. The information is not conveyed in source samples and not fully explored, which implies improving the model performance by exploiting unlabelled target samples.

Several semi-supervised learning approaches utilize unlabelled data to promote model performance in domain adaptation tasks. Entropy regularization and minimization are useful for semi-supervised learning [7] and domain adaptation in semantic segmentation [8]. As an improved variant of entropy regularization, the maximum squares loss [9] is proposed to tackle imbalance between the gradients of well-classified and poorly classified target samples. Wrapper methods are among the oldest and most practical approaches for semi-supervised learning and can be used with any supervised-based model without a hitch. The training procedure usually consists of two alternating steps, one of which is training the supervised-based model, and the other is pseudo-labelling unlabelled samples. The prediction probabilities may not reflect the actual confidence ranking because of the influence of many unexpected factors. Hence, the quality and selected procedure of pseudo-labelled samples are of particular importance.

Here, we utilize a self-training framework [10] to exploit information conveyed in unlabelled target samples for domain adaptive image classification tasks. The model’s main components are a feature extractor, a classifier, and a conditional domain discriminator, as shown in Figure 1. The network parameters and pseudo-labels are jointly optimized in an objective function. As a wrapper method, the training procedure contains two alternating steps. In the first step, we fix the pseudo-labels and train the network with labelled source samples and unlabelled target samples. In the second step, we fix the network parameters and update pseudo-labels for target samples. The procedure for selecting pseudo-labels is also presented. Experimental results demonstrate the proposed approach’s promising performance, outperforming the most existing state-of-the-art competitors on four widely used datasets.

The main contribution is that the adversarial training and self-training are unified in one objective function, minimized by a novel self-training scheme. The rest of the article organizes as follows. Section 2 reviews the related work, including adversarial domain adaptation and self-training techniques. Section 3 details the proposed approach. Then, Section 4 conducts comprehensive experiments. Finally, Section 5 concludes the work.

## 2 | RELATED WORKS

This paper aligns the source and target domain in the feature level using adversarial training and exploits unlabelled samples using self-training. This section will first review UDA, then focus on adversarial-domain adaptation and self-training techniques, which are the essential components of the proposed approach.

### 2.1 | Unsupervised domain adaptation

Over the past two decades, numerous shallow UDA approaches have been proposed and been reviewed in [11, 12], including instance re-weighting [13], re-sampling [14] and subspace alignment [15]. As an efficient tool to extract and represent feature representations, CNNs have been leveraged to learn domain-invariant feature representations for various UDA tasks. According to the research from Wang et al. [16], the approaches for deep UDA divide into three main classes. The discrepancy-based approaches minimize the discrepancy between source and target feature distributions, such as maximum mean discrepancy (MMD) [17], correlation alignment (CORAL) [18] and Wasserstein distance [19]. The second case is adversarial domain adaptation [6, 20], which adopts the two-player game utilized in Generative Adversarial Networks (GANs) [21] to train a domain-invariant feature extractor. The last case refers to reconstruction-based approaches, where encoder–decoder structures are applied [22, 23].
2.2 | UDA and feature representation

Aligning feature representations between the two domains is one of the most important approaches for UDA. Recently, GAN [21] achieved great success in estimating generative models via adversarial training. GAN consists of two networks. A generative model $G$ that extracts the feature representations in terms of data distributions. And a discriminator model $D$ that distinguishes which dataset the feature representations belong to. The model $G$ tries to generate invariant features in order to fool the discriminator $D$. Inspired by GAN, the domain adversarial neural network (DANN) [20] introduced a gradient reversal layer (GRL) into the discriminator to align the feature distributions of source and target domain. Hoffman et al. [24] performed a global feature alignment using adversarial training and constrained the adversarial model with category statistics. The variants of GAN are also explored to improve the transfer learning performance of feature extractors. Instead of classifying domain labels, Shen et al. [19] applied a discriminator to estimate Wasserstein distance [25] between source and target data, which is minimized by optimizing the feature extractor network. Motivated by conditional GAN [26, 27], Long et al. [6] utilized discriminative information conveyed in the classifier prediction to condition the adversarial domain adaptation network. Luo et al. [28] trained two classifiers to align each class with an adaptive adversarial loss for semantic segmentation tasks. Based on the feature representations spectral analysis, Chen et al. [29] found that the eigenvectors with the largest singular values dominated the feature transferability.

2.3 | UDA and self-training

Self-training is a suitable technique to learn a model from large amounts of unlabelled data and a small quantity of labelled data and has been well researched for vision applications [10, 30, 31]. A complimentary survey of self-labelled techniques was reported in [32]. Several studies have shown that entropy minimization is useful for semi-supervised learning and self-training [8, 33, 34]. In the Mean Teacher method [35], the exponential moving average of the model was adopted as the teacher model, whose prediction was used to guide the model. Another simple and efficient method for self-training is the pseudo-labelled (self-labelled) technique [36], which generates pseudo-labels for unlabelled samples. Confidence measures are investigated as an essential property of self-labeling [37]. In [38], entropy was applied as the confidence indicator for producing more accurate pseudo-labels. Instead of using the classifier’s prediction output, Zhang et al. [39] resorted to feature distance to generate pseudo-labels via category anchors. Most self-labelled approaches enlarge pseudo-labelled sample set iteratively. Zou et al. [10] formulated the UDA semantic segmentation as a latent variable loss minimization problem. And the problem was solved by alternatively pseudo-labeling unlabelled target samples in a class-balanced way and re-training the model with enlarged datasets. In their follow-up work [40], the algorithm was reformulated under a continuous framework and several regularization terms were explored. Based on assumptions on the feature space, Song et al. [41] proposed a novel self-training scheme for unsupervised domain adaptive re-ID tasks. Similar to our work, Adversarial-Learned Loss [29] introduced the confusion matrix learned from the discriminator prediction to bridge the gap between pseudo-label and the ground truth.

These methods do not explore the adversarial training and self-training together. This paper unified both training approaches in an objective function, which is vital to capture the latent information conveyed in unlabelled samples.

3 | METHOD

3.1 | Problem settings

This paper focuses on the problem of UDA in image classification, where we are given a source dataset that consists of $n_s$ labelled samples $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and a target dataset consists of $n_t$ unlabelled samples $\{(x_i^t)\}_{i=1}^{n_t}$. The training images $X^s = \{x_i^s\}_{i=1}^{n_s}$ and $X^t = \{x_i^t\}_{i=1}^{n_t}$ are sampled from marginal probability distributions $P(X^s)$ and $P(X^t)$, respectively, and the assumption is $P(X^s) \neq P(X^t)$, which is called data distribution shift. Source knowledge $P(Y^s|X^s)$ can be learned in a supervised way from labelled data $(X^s, Y^s)$, where $Y^s = \{y_i^s\}_{i=1}^{n_s}$. The goal is to learn $P(Y^t|X^t)$ from source data $(X^s, Y^s)$ and unlabelled data $X^t$ in an unsupervised way.

To fully exploit beneficial information carried in unlabelled target samples, we align feature distribution globally using adversarial training and label target samples using self-training. We then re-train the model with labelled source samples and pseudo-labelled target samples. CDAN is chosen as the supervised base model, on which the wrapper method is applied. Section 3.2 introduces basic knowledge about DANN and CDAN. Then, Section 3.3 extends CDAN with self-training and presents the unified objective function and solver. Section 3.4 gives the selection procedure for pseudo-labels and the entire training procedure.

3.2 | Conditional adversarial training

In deep domain adaptation networks, the feature representation extracted from an input image $x$ by the backbone $G$ is denoted by $f = G(x)$ and the probability predicted by the classifier $T$ with the input of $f$ is denoted by $g = softmax(T(f))$. The key of the deep UDA is to learn domain-invariant feature extractor $G$. Inspired by GANs, adversarial training is used to align the feature representations between source and target domain. DANN is a two-player game: the discriminator $D$ with GRL is used to classify which domain the input feature belongs to while the backbone $G$ tries to extract domain-invariant feature representations to confuse the discriminator. The general loss function of DANN for image classification can be formulated as a min–max problem:

$$
\min_{G,T} \max_D \mathcal{L}_{CE}(T(G(X^s)), Y^s) + \lambda \mathcal{L}_{adv}(D(G(X^s), Y^D)) \tag{1}
$$
where $\lambda$ denotes a balancing parameter and $L_{CE}(\cdot, \cdot)$ is the cross entropy loss with $C$ categories:

$$L_{CE} = -\frac{1}{n_t} \sum_{j=1}^{n_t} \sum_{c=1}^{C} y^*_j \log g^*_c$$  

(2)

and $Y^*_{Dj}$ is domain label with 1 for source domain and 0 for target domain:

$$L_{adv} = -\frac{1}{n_t} \sum_{j=1}^{n_t} \log D(g^*_j) - \frac{1}{n_t} \sum_{j=1}^{n_t} \log(1 - D(g^*_j)).$$  

(3)

As pointed in CDAN [6], DANN fails to capture multimodal structures that are native in classification problems. Because even if the discriminator is fully confused, the two feature distributions are only guaranteed by global alignment instead of sufficient similarity [42]. What is more, the adversarial adaptation of feature representations $f$ is not enough to bridge the domain gap. Motivated by the observation that classifier prediction $g$ conveys discriminative information potentially revealing the multimodal structures, CDAN tackled the two bottlenecks above by condition the discriminator $D$ on $g$ through the joint variable $h = (f, g)$. Thus, the two-player game is formulated as:

$$\min_{G,T,D} \max_{G,T,D} L_{CE}(T(G(X')), Y') + \lambda L_{adv}(D(M(h')), Y^*_{Dj}).$$  

(4)

The condition strategy $M$ is a mapping function:

$$M(h) = \begin{cases} M_{\oplus}(f, g) & \text{if } d_f \times d_g \leq 4096, \\ M_{\ominus}(f, g) & \text{else ,} \end{cases}$$  

(5)

where $M_{\oplus}$ and $M_{\ominus}$ denote multilinear map and explicit randomized multilinear map, $d_f$ and $d_g$ denote the dimensions of feature vectors.

### 3.3 CDAN with self-training

Even armed with conditional adversarial-training strategy, we observed that the deep adversarial adaptation approaches could not fully utilize the target samples’ information. The main bottleneck is that classification supervision is provided by source samples alone because the target ground truth labels are unavailable in unsupervised domain adaptation. The simple idea is to exploit target samples using self-training techniques.

Zou et al. [10] proposed an iterative class-balanced self-training framework with self-paced learning. In their framework and the follow-up one [40], the adversarial training is ignored, and the model structure only consists of a generator and a classifier. We extended their framework with conditional adversarial training and reformulated Equation (4) as:

$$\min_{G,T,D} \max_{G,T,D} L_{CE}(T(G(X')), Y') + \lambda L_{adv}(D(M(h')), Y^*_{Dj})$$

$$+ \alpha (L_{CE}(T(G(X')), \hat{Y}') - L_{normal}(\hat{Y}'))$$

(6)

s.t.  

$$\hat{Y}_{j} \in \Delta^{C-1} \cup 0, j = 1, 2, \ldots, n_t,$$

where $\hat{Y}' = \{\hat{y}'_j\}_{j=1}^{n_t}$ is pseudo-labels set of target samples, $\hat{y}'_j$ is one-hot vectors and $\alpha$ is a balancing parameter. If $\hat{y}'_j$ is a zero vector, the target samples $X'_{j}$ is not used when training the classifier. $L_{normal}$ is the regularization term to prevent the trivial solution of ignoring all target samples. In Equation (6), the network parameters and pseudo-labels are optimized jointly, which is undoubtedly complicated because it is hard to guarantee pseudo-labels’ quality. What is more, the network parameters can be learned continuously using SGD while pseudo-labels are discrete. So we optimize the networks parameters and pseudo-labels alternatively with alternating two steps:

1. Fix pseudo-labels $\hat{Y}'$ and minimize Equation (6) with respect to the network parameters $G$, $T$ and $D$.

2. Fix the network parameters $G$, $T$, and minimize Equation (6) to generate the pseudo-labels of target samples, that is, $\hat{Y}'$.

One step of 1 followed by one step of 2 is called one training round and repeats the two steps for multiple rounds. When pseudo-labels are fixed, it is equal to minimizing Equation (7) using SGD.

$$\min_{G,T,D} \max_{G,T,D} L_{CE}(T(G(X')), Y') + \lambda L_{adv}(D(M(h')), Y^*_{Dj})$$

$$+ \alpha L_{CE}(T(G(X')), \hat{Y}').$$

(7)

### 3.4 Pseudo-label generating

When the network parameters are fixed, Equation (6) is a discrete optimization problem. The regularization term is defined as $\sum_{j=1}^{n_t} \sum_{c=1}^{C} k \cdot \hat{y}'_{j,c}$, where $k \in \mathbb{R}^C$ are hyper-parameters. Equation (6) can be reformulated as:

$$\min_{\hat{y}'_{j,c}} \sum_{j=1}^{n_t} \sum_{c=1}^{C} y^*_j \log g^*_c - \lambda \sum_{j=1}^{n_t} \sum_{c=1}^{C} \hat{y}'_{j,c} \log g^*_c$$

$$- \lambda \sum_{j=1}^{n_t} \sum_{c=1}^{C} \hat{y}'_{j,c} \exp(-k_c)$$

s.t.  

$$\hat{y}'_{j,c} \in \Delta^{C-1} \cup 0, j = 1, 2, \ldots, n_t,$$

(8)

where $\exp(-k_c)$ controls the proportion of pseudo-labels for each category and can be denoted as $g^*_c$. If $g^*_c$ is fixed, Equation (8) can be optimized by each target samples, leading to the following solver:

$$\hat{y}'_{j,c}^* = \begin{cases} 1, & \text{if } \epsilon = \arg \max_{\hat{y}'_{j,c}} / g^*_c \text{ and } \hat{y}'_{j,c} > g^*_c, \\ 0, & \text{otherwise.} \end{cases}$$

(9)

The quality of pseudo-labels $\hat{Y}'$ is essential to model performance but hard to guarantee. A self-paced learning strategy...
Datasets

EXPERIMENTS

The proposed approach is compared with several state-of-the-art algorithms on four publicly used datasets.

Office-31 [53] is the most commonly used dataset for image classification in UDA, with 31 categories and 4652 images. The images are collected from three domains: Amazon(A), Webcam(W) and Dslr(D). The image numbers of the three domains are 2817, 795 and 498, respectively. In experiments, we build six transfer tasks: A → W, A → D, W → A, W → D, D → A, D → W.

ImageCLEF-DA [49] consists of 12 common classes shared by three domains: Caltech-256(C), ImageNet ILSVRC 2012(I) and Pascal VOC 2012(P). Each domain has about 600 images. As in Office-31, we build six transfer tasks: C → I, C → P, I → C, I → P, P → C, P → I.

Office-Home [43] consists of 65 categories and 15,500 images, which is a more difficult dataset than Office-31. The images are collected from the office and home environments and four distinct domains: Art(Ar), Clipart(Cl), Product(Pr) and Real World(Rw). Figure 2 shows some example images. The image

4 | EXPERIMENTS

4.1 | Datasets

Algorithm 1: Determine \( \{ \hat{g}_t \}_{t=1}^C \) in each training round

Require: Neural Network \( G, T \), target samples \( X^t \) and the proportion of pseudo labels \( p \)

Ensure: \( \{ \hat{g}_t \}_{t=1}^C \)

\[
\text{label} = \text{zeros}(n_t, 1), \quad \text{prob} = \text{zeros}(n_t, 1), \quad P = [\ldots] \times C
\]

for \( j = 1 \) to \( n_t \) do

\[
g_j^t = \text{softmax}(T(G(s_j^t)))
\]

\[
\text{label}[j] = \arg \max (g_j^t)
\]

\[
\text{prob}[j] = \max (g_j^t)
\]

\[
P[\text{label}[j]].\text{append} (\text{prob}[j])
\]

end for

for \( c = 1 \) to \( C \) do

\[
P[c] = \text{sort}(P[c], \text{order} = \text{descending})
\]

\[
\text{Len}_c = \text{int}(\text{length}(P[c]) \times p)
\]

if \( \text{Len}_c < 1 \) then

\[
\text{Len}_c = 1
\]

end if

\[
g^c = P[\text{Len}_c]
\]

end for

return \( \{ \hat{g}_t \}_{t=1}^C \)

Algorithm 2: Training process

Require: Neural Network \( G, T, D \), source samples \( (X^s, Y^s) \), target samples \( (X^t) \), the proportion of pseudo labels \( p \) and step \( \Delta p \) and the number of training round \( K \)

Ensure: Neural Network \( G, T, D \) and pseudo labels \( \hat{Y}^t \)

Initialization: \( \hat{Y}^t = \emptyset \)

for \( k = 1 \) to \( K \) do

fix \( \hat{Y}^t \) and unfix \( G, T \) step 1

train \( G, T, D \) with \( (X^s, Y^s) \) and \( (X^t, \hat{Y}^t) \) step 1

unfix \( \hat{Y}^t \) and fix \( G, T \) step 2

update \( \hat{Y}^t \) using Equation (9) step 2

\[
p = p + \Delta p
\]

end for

return \( G, T, D \) and \( \hat{Y}^t \)

FIGURE 2 Example images from Office-Home [43]
4.2 Implementation details

The proposed method is implemented in PyTorch [55], and adopt the ResNet-50 [55] model pretrained on ImageNet [1] as the backbone and train the classifier and the discriminator from scratch. The discriminator is composed of a GRL and three fully-connected layers with dropout and ReLu between each other. Mini-batch stochastic gradient descent (SGD) is applied with momentum of 0.9, basic weight decay of 0.0005, and basic learning rate of 0.001. The batchsize is set to 36, and the model is trained on a NVIDIA RTX 2080Ti GPU. The multiplier factor for weight decay and learning rate of the classifier and discriminator are 2 and 10, respectively. The learning rate is scheduled by $\eta_p = \eta_0(1 + \alpha \text{iter})^{-\gamma}$, where $\eta_0 = 0.001$, $\alpha = 0.001$, $\gamma = 0.75$ and iter is the iteration number. Linear Warmup strategy with 200 steps starting from 0 for learning rate is also adopted. For the self-paced learning policy, we set $\rho_{\text{start}} = 0.5$ for two easier datasets Office-31 and ImageCLEF-D-A, and $\rho_{\text{start}} = 0.2$ for Office-Home. The step $\Delta \rho$ is 0.05 for all four datasets. The model is trained in 6 rounds and 5000 iterations for each round.

### Results

In this section, several experiments are conducted to analyze the performance of the proposed method. We compare with state-of-art domain adaptation methods: Deep Adaptation Network (DAN) [17], Residual Transfer Network (RTN) [45], Domain Adversarial Neural Network (DANN) [20], Adversarial Discriminative Domain Adaptation (ADDA) [46], Joint Adaptation Network (JAN) [49], Generate to Adapt (GTA) [50], Conditional Adversarial Domain Adaptation with Entropy (CDAN + E) [6], Conditional Adversarial Domain Adaptation with Batch Spectral Penalization (BSP + CDAN) [52], Multi-adversarial Domain Adaptation (MADA) [47], Similarity Networks (SimNet) [48], Transferable Adversarial Training (TAT) [51], Adversarial-Learned Loss for Domain Adaptation (ALDA) [29]. All labelled source samples and unlabelled target samples are used in the training process. We follow the protocol of CDAN [6] and compare the average classification accuracy of three random experiments.

Table 1 reveals that self-training with pseudo-labels positively affects adversarial domain adaptation. It is desirable that the proposed approach substantially outperforms other methods in five tasks and gets state-of-the-art performance on Office-31. Note that, on the two hardest tasks, that is, D → A and W → A, our approach gets the biggest improvement of both 4.4 compared with CDAN + E. The reason is that the Amazon domain has more samples than the other two domains, and other approaches cannot utilize the category information conveyed in the pseudo-labels. It makes sense that the target domain’s supervising signal promotes adaptation performance when a small number of labelled source samples and a large number of unlabelled target samples are provided.

Table 2 shows the evaluation results on ImageCLEF-DA. The proposed method obtains the best performance on five transfer tasks.

| Method       | A → W | D → W | W → D | A → D | D → A | W → A | Average |
|--------------|-------|-------|-------|-------|-------|-------|---------|
| ResNet-50    | 68.4 ± 0.2 | 96.7 ± 0.2 | 99.3 ± 0.1 | 68.9 ± 0.2 | 62.5 ± 0.3 | 60.7 ± 0.3 | 76.1    |
| DAN [17]     | 80.5 ± 0.4 | 97.1 ± 0.2 | 99.6 ± 0.1 | 78.6 ± 0.2 | 63.6 ± 0.3 | 62.8 ± 0.2 | 80.4    |
| RTN [45]     | 84.5 ± 0.2 | 96.8 ± 0.1 | 99.4 ± 0.1 | 77.5 ± 0.3 | 66.2 ± 0.2 | 64.8 ± 0.3 | 81.6    |
| DANN [20]    | 82.0 ± 0.4 | 96.9 ± 0.2 | 99.1 ± 0.1 | 79.7 ± 0.4 | 68.2 ± 0.4 | 67.4 ± 0.5 | 82.2    |
| ADDA [46]    | 86.2 ± 0.5 | 96.2 ± 0.3 | 98.4 ± 0.3 | 77.8 ± 0.3 | 69.5 ± 0.4 | 68.9 ± 0.5 | 82.9    |
| MADA [47]    | 90.0 ± 0.1 | 97.4 ± 0.1 | 99.6 ± 0.1 | 87.8 ± 0.2 | 70.3 ± 0.3 | 66.4 ± 0.3 | 85.2    |
| SimNet [48]  | 88.6 ± 0.5 | 98.2 ± 0.2 | 99.7 ± 0.2 | 85.3 ± 0.3 | 73.4 ± 0.8 | 71.6 ± 0.6 | 86.2    |
| JAN [49]     | 85.4 ± 0.3 | 97.8 ± 0.2 | 99.8 ± 0.2 | 84.7 ± 0.3 | 68.6 ± 0.3 | 70.0 ± 0.4 | 84.3    |
| GTA [50]     | 89.5 ± 0.5 | 97.9 ± 0.3 | 99.8 ± 0.4 | 87.7 ± 0.5 | 72.8 ± 0.3 | 71.4 ± 0.4 | 86.5    |
| CDAN + E [6] | 94.1 ± 0.1 | 98.6 ± 0.1 | 100.0 ± 0.0 | 92.9 ± 0.2 | 71.0 ± 0.3 | 69.3 ± 0.3 | 87.7    |
| TAT [51]     | 92.5 ± 0.3 | 99.3 ± 0.1 | 100.0 ± 0.0 | 93.2 ± 0.2 | 73.1 ± 0.3 | 72.1 ± 0.3 | 88.4    |
| BSP + CDAN [52] | 93.3 ± 0.2 | 98.2 ± 0.2 | 100.0 ± 0.0 | 93.0 ± 0.2 | 73.6 ± 0.3 | 72.6 ± 0.3 | 88.5    |
| ALDA [29]    | 95.6 ± 0.5 | 97.7 ± 0.1 | 100.0 ± 0.0 | 94.0 ± 0.4 | 72.2 ± 0.4 | 72.5 ± 0.2 | 88.7    |
| Ours         | 94.8 ± 0.5 | 98.9 ± 0.2 | 100.0 ± 0.0 | 95.3 ± 0.1 | 75.4 ± 0.2 | 73.7 ± 0.2 | 89.7    |

Bold values indicate the best results of the current experimental group.


### Table 2: Accuracy (%) on ImageCLEF-D4 for unsupervised domain adaptation

| Method       | I → P | P → I | I → C | C → I | C → P | P → C | Average |
|--------------|-------|-------|-------|-------|-------|-------|---------|
| ResNet-50 [44] | 74.8 ± 0.3 | 83.9 ± 0.1 | 91.5 ± 0.3 | 78.0 ± 0.2 | 65.5 ± 0.3 | 91.2 ± 0.3 | 76.1 |
| DAN [17]      | 74.5 ± 0.4 | 82.2 ± 0.2 | 92.8 ± 0.2 | 86.3 ± 0.4 | 69.2 ± 0.4 | 89.8 ± 0.4 | 80.4 |
| DANN [20]     | 75.0 ± 0.6 | 86.0 ± 0.3 | 96.2 ± 0.4 | 87.0 ± 0.5 | 74.3 ± 0.5 | 91.5 ± 0.6 | 82.2 |
| JAN [49]      | 76.8 ± 0.4 | 88.0 ± 0.2 | 94.7 ± 0.2 | 89.5 ± 0.3 | 74.5 ± 0.3 | 91.7 ± 0.3 | 84.3 |
| MADA [47]     | 75.0 ± 0.3 | 87.9 ± 0.2 | 96.0 ± 0.3 | 88.8 ± 0.3 | 75.2 ± 0.2 | 92.2 ± 0.3 | 85.8 |
| CDAN + E [6]  | 76.8 ± 0.4 | 90.7 ± 0.2 | 97.7 ± 0.2 | 91.3 ± 0.3 | 74.2 ± 0.2 | 94.3 ± 0.3 | 87.7 |
| Ours          | 78.7 ± 0.4 | 92.6 ± 0.3 | 96.6 ± 0.1 | 92.8 ± 0.4 | 75.8 ± 0.2 | 93.9 ± 0.2 | 88.4 |

Bold values indicate the best results of the current experimental group.

### Table 3: Accuracy (%) on Office-Home for unsupervised domain adaptation

| Method       | Ar → Cl | Ar → Pr | Ar → Rw | Cl → Ar | Cl → Pr | Cl → Rw | Pr → Ar | Pr → Cl | Pr → Rw | Rw → Ar | Rw → Cl | Rw → Pr | Average |
|--------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| ResNet-50 [44] | 34.9    | 50.0    | 58.0    | 37.4    | 41.9    | 46.2    | 38.5    | 31.2    | 60.4    | 53.9    | 41.2    | 59.9    | 46.1    |
| DAN [17]      | 43.6    | 57.0    | 67.9    | 45.8    | 56.5    | 60.4    | 44.4    | 43.6    | 67.7    | 63.1    | 51.5    | 74.3    | 56.3    |
| DANN [20]     | 45.6    | 59.3    | 70.1    | 47.0    | 58.5    | 60.9    | 46.1    | 43.7    | 68.5    | 63.2    | 51.8    | 76.8    | 57.6    |
| JAN [49]      | 49.0    | 61.3    | 68.9    | 50.4    | 59.7    | 61.0    | 45.8    | 43.4    | 70.3    | 63.9    | 52.4    | 76.8    | 58.3    |
| CDAN + E [6]  | 50.7    | 70.6    | 76.0    | 57.6    | 70.0    | 70.0    | 57.4    | 50.9    | 77.3    | 70.9    | 56.7    | 81.6    | 65.8    |
| TAT [51]      | 51.6    | 69.5    | 75.4    | 59.4    | 69.5    | 68.6    | 59.0    | 50.5    | 76.8    | 70.9    | 56.6    | 81.6    | 65.8    |
| BSP + CDAN [52] | 52.0    | 68.6    | 76.1    | 58.0    | 70.3    | 70.2    | 58.6    | 50.2    | 77.6    | 72.2    | 59.3    | 81.9    | 66.3    |
| ALDA [29]     | 53.7    | 70.1    | 76.4    | 60.2    | 72.6    | 71.5    | 56.8    | 51.9    | 77.1    | 70.2    | 56.3    | 82.1    | 66.6    |
| Ours          | 55.6    | 71.6    | 78.4    | 62.3    | 71.9    | 72.8    | 60.7    | 50.3    | 79.3    | 71.2    | 54.4    | 82.5    | 67.6    |

Bold values indicate the best results of the current experimental group.

### Table 4: Accuracy (%) on Digits for unsupervised domain adaptation

| Method       | M → U | M → U | S → M | Average |
|--------------|-------|-------|-------|---------|
| UNIT [58]    | 96.0  | 93.6  | 90.5  | 93.4    |
| CyCADA [54]  | 95.6  | 96.5  | 90.4  | 94.2    |
| CDAN + E [6] | 95.6  | 98.0  | 89.2  | 94.3    |
| Ours         | 96.4  | 98.2  | 92.0  | 95.5    |

Bold values indicate the best results of the current experimental group.

The most significant improvement occurs on the hardest task $S \rightarrow M$.

### Analysis

#### 4.4.1 Pseudo-label proportion

Table 5 shows the influence of the pseudo-label proportion on model performance. $p_{\text{start}}$ is increased from 0.2 to 0.7 by a step of 0.1 and three random experiments on the dataset Office-31. The average accuracy increases slightly when $p_{\text{start}}$ is smaller than 0.4. The best performance is obtained at $p_{\text{start}} = 0.5$. Note that the main contributions to larger accuracy increases come from the task $D \rightarrow A$ and $W \rightarrow A$, where $n_y$ is much bigger than $n_x$.

#### 4.4.2 Self-training methods

The performances of different self-training strategies are compared on the dataset Office-31 in Table 6. "MinEnt" and "MaxSquare" are the CDAN + E baseline with entropy minimization and max square loss, respectively. The balance parameters for "MinEnt" and "MaxSquare" are 0.01. Entropy minimization contributes to improving performance by 0.5, which illustrates the need for exploiting the information of tasks, but with a smaller average improvement than the results on Office-31. A possible reason is that the three domains in this dataset have about the same number of samples and smaller gaps between them, which is different from Office-31.

Table 3 shows the evaluation results on Office-Home. The four domains of Office-Home have more categories and greater gaps between each other, leading to harder transfer tasks and decreased model performance. Nevertheless, the model significantly outperforms the comparison methods on most transfer tasks, with a larger improvement room than Table 2. The most considerable improvement occurs on the task $Ar \rightarrow Cl$, where the number of the target domain samples is about twice the source domain, proving our method has a more significant advantage than others with a larger number of unlabelled samples.

Table 4 shows the evaluation results on Digits. On all three tasks, the proposed method gets the highest accuracy. Again,
TABLE 5  Experiments of pseudo-label proportion on Office-31 for unsupervised domain adaptation

| p   | A → W | D → W | W → D | A → D | D → A | W → A | Average |
|-----|-------|-------|-------|-------|-------|-------|---------|
| 0.0 | 94.1 ± 0.1 | 98.6 ± 0.1 | **100.0 ± 0.0** | 92.9 ± 0.2 | 71.0 ± 0.3 | 69.3 ± 0.3 | 87.7 |
| 0.2 | 93.3 ± 0.5 | 98.9 ± 0.3 | **100.0 ± 0.0** | 93.8 ± 0.4 | 71.1 ± 0.8 | 70.1 ± 0.3 | 87.9 |
| 0.3 | 93.5 ± 0.5 | **99.3 ± 0.0** | **100.0 ± 0.0** | 94.0 ± 0.4 | 72.1 ± 0.4 | 70.7 ± 1.2 | 88.3 |
| 0.4 | **95.0 ± 0.9** | **99.3 ± 0.0** | **100.0 ± 0.0** | **95.3 ± 0.3** | 75.4 ± 0.6 | 72.9 ± 0.3 | 89.6 |
| 0.5 | 94.8 ± 0.5 | 98.9 ± 0.2 | **100.0 ± 0.0** | 95.3 ± 0.1 | 75.4 ± 0.2 | 73.7 ± 0.2 | **89.7** |
| 0.6 | 94.9 ± 0.6 | 98.6 ± 0.0 | **100.0 ± 0.0** | 95.3 ± 0.1 | 75.6 ± 0.2 | 73.7 ± 0.2 | **89.7** |
| 0.7 | 94.1 ± 0.3 | 98.7 ± 0.0 | **100.0 ± 0.0** | 93.6 ± 0.2 | 73.9 ± 0.3 | 75.2 ± 1.0 | 89.3 |

Bold values indicate the best results of the current experimental group.

TABLE 6  Experiments on Office-31 with different self-training strategies

| Task   | CDAN+E [6] | MinEnt [8] | MaxSquare [9] | Ours |
|--------|-------------|------------|---------------|------|
| A → W  | 94.1        | 93.6       | **95.0**      | 94.8 |
| A → D  | 92.9        | 93.5       | 92.8          | **95.3** |
| D → A  | 71.0        | 72.8       | 73.0          | **75.4** |
| D → W  | 98.6        | **99.3**   | 98.7          | 98.6 |
| W → A  | 69.3        | 70.0       | 71.5          | **73.7** |
| W → D  | **100.**    | **100.**   | **100.**      | **100.** |
| Mean   | 87.7        | 88.2       | 88.5          | **89.7** |

Bold values indicate the best results of the current experimental group.

unlabelled target samples. With the balanced gradient of highly confident classes, MaxSquare increases the average accuracy by 0.8, compared with the baseline. Finally, the proposed approach benefits from the scenario with much more unlabelled target samples and significant improvement by 2.0.

4.4.3  Feature visualization

The feature representations of task A → W are visualized in Figure 3 using t-SNE [57], by ResNet, DANN, CDAN + E and Ours. ResNet cannot align the source and target well, suggesting that a large domain gap between the two domains exists. DANN aligns the representations much better than ResNet but shows poor performance in some categories. The source and target are well aligned, and classes are discriminated better by CDAN + E, while the proposed approach obtains better aligned centers. This shows the benefit of exploiting the information conveyed in the target samples.

4.4.4  Convergence

As shown in Figure 4, ResNet, DANN, CDAN + E, and Ours’ convergence performance is testified, with the test accuracy

![Figures and tables](image-url)
on the task $A \rightarrow W$. As the training proceeds, ResNet gradually overfits the source domain and underfit it to the target domain, which illustrates the domain shift. DANN obtains better accuracy performance, but the growth is much unstable and stops earlier compared with CDAN + E. The proposed method enjoys the same faster convergence as CDAN + E at the first training round and steadily improves after adding classification supervision with pseudo-labels.

5 CONCLUSION

In this paper, a self-training framework is utilized to wrap deep adversarial domain adaptation method to exploit the information conveyed in unlabelled target samples fully. The domain adaptive image classification problem is reformulated as a min-max objective function, where the network parameters and the pseudo-labels are jointly optimized. The solver consists of two alternating steps as usually used in wrapper methods. The first step trains the network parameters with labelled source samples and pseudo-labelled target samples. The second step generates and selects the pseudo-labels according to the predicted probabilities output from the network. Experiments on standard benchmark datasets show state-of-the-art performance. Analysis experiments testify to the feasibility and effectiveness of the proposed method.

In future work, we will extend this work for other visual domain adaptation tasks, such as semantic segmentation and object detection.

ACKNOWLEDGEMENTS

This research was supported in part by the National Natural Science Foundation of China under Grant 11902349, Grant 61801491, in part by the Hunan Provincial Natural Science Foundation of China under Grant 2019JJ50732, Grant 2019JJ50739, and in part by the Postgraduate Scientific Research Innovation Project of Hunan Province under Grants CX20200024, CX20200025, and CX20200088.

ORCID

Zi Wang https://orcid.org/0000-0001-9081-7185
Xiaoliang Sun https://orcid.org/0000-0003-3018-8043
Ang Su https://orcid.org/0000-0003-0965-8717
Gang Wang https://orcid.org/0000-0003-0114-0649
Yang Li https://orcid.org/0000-0003-0817-098X
Qifeng Yu https://orcid.org/0000-0002-1101-1093

REFERENCES

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classifi cation with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
2. Ren, S., et al.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
3. Donahue, J., et al.: Decaf: A deep convolutional activation feature for generic visual recognition. In: International Conference on Machine Learning, pp. 647–655 (2014)
4. Ben-David, S., et al.: Analysis of representations for domain adaptation. In: Advances in Neural Information Processing Systems, pp. 137–144 (2007)
5. Zhang, Y., et al.: Bridging theory and algorithm for domain adaptation. In: International Conference on Machine Learning, pp. 7404–7413 (2019)
6. Long, M., et al.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems, pp. 1640–1650 (2018)
7. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Advances in Neural Information Processing Systems 17, pp. 529–536 (2005)
8. Vu, T.H., et al.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2517–2526 (2019)
9. Chen, M., Xue, H., Cai, D.: Domain adaptation for semantic segmentation with maximum squares loss. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2090–2099 (2019)
10. Zou, Y., et al.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 289–305 (2018)
11. Csurka, G.: Domain adaptation for visual applications: A comprehensive survey. arXiv:1702.05374 (2017)
12. Patel, V.M., et al.: Visual domain adaptation: A survey of recent advances. IEEE Signal Process Mag. 32(3), 53–69 (2015)
13. Dai, W., et al.: Boosting for transfer learning. In: Proceedings of the 24th International Conference on Machine Learning, pp. 193–200 (2007)
14. Chu, W.S., De la Torre, F., Cohn, J.F.: Selective transfer machine for personalized facial action unit detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3515–3522 (2013)
15. Fernando, B., et al.: Unsupervised visual domain adaptation using subspace alignment. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2960–2967 (2013)
16. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing 312, 135–153 (2018)
17. Long, M., et al.: Transferable representation learning with deep adaptation networks. IEEE Trans. Pattern Anal. Mach. Intell. 41(12), 3071–3085 (2019)
18. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: European Conference on Computer Vision, pp. 443–450. Springer, Berlin (2016)
19. Shen, J., et al.: Wasserstein distance guided representation learning for domain adaptation. In: AAAI (2018)
20. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. 17(1), 2096–2103 (2016)
21. Goodfellow, I.J., et al.: Generative adversarial networks. Adv. Neural Inf. Process Syst. 3, 2672–2680 (2014)
22. Bousmalis, K., et al.: Domain separation networks. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, pp. 343–351 (2016)
23. Chang, W., et al.: All about structure: Adapting structural information across domains for boosting semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1900–1909 (2019)
24. Hoffman, J., et al.: FCNS in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv:1612.02649 (2016)
25. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv:1701.07875 (2017)
26. Isola, P., et al.: Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5967–5976 (2017)
27. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier GANs. In: International Conference on Machine Learning, pp. 2642–2651 (2017)
28. Luo, Y., et al.: Taking a closer look at domain shift: Category-level adversarial networks for semantics consistent domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2507–2516 (2019)
29. Chen, M., et al.: Adversarial-learned loss for domain adaptation (pp. 3521–3528). AAAI Press, Palo Alto (2020)
30. Yan, H., et al.: Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2272–2281 (2017)
31. Kim, M., Byun, H.: Learning texture invariant representation for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12975–12984 (2020)
32. Triguero, I., García, S., Herrera, F.: Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. Knowl. Inf. Syst. 42(2), 245–284 (2015)
33. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Advances in Neural Information Processing Systems, pp. 529–536 (2005)
34. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. arXiv:151106390 (2015)
35. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: NIPS, pp. 1195–1204 (2017)
36. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML vol. 3 (2013)
37. Goldman, S.A., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 327–334 (2000)
38. Saporta, A., et al.: ESL: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation. arXiv:2006.08658 (2020)
39. Zhang, Q., et al.: Category anchor-guided unsupervised domain adaptation for semantic segmentation. In: Advances in Neural Information Processing Systems, pp. 435–445 (2019)
40. Zou, Y., et al.: Confidence regularized self-training. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5982–5991 (2019)
41. Song, L., et al.: Unsupervised domain adaptive re-identification: Theory and practice. Pattern Recognit. 102, 107173 (2020)
42. Arora, S., et al.: Generalization and equilibrium in generative adversarial nets (GANs). In: ICML. Vol. 70 of Proceedings of Machine Learning Research, pp. 224–232 (2017)
43. Hoffman, J., et al.: Simultaneous deep transfer across domains and tasks. In: Csurka, G. (ed.) Domain Adaptation in Computer Vision Applications. Advances in Computer Vision and Pattern Recognition, pp. 173–187. Springer, Berlin (2017)
44. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
45. Long, M., et al.: Unsupervised domain adaptation with residual transfer networks. In: Lee, D.D., Sugiyma, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, pp. 136–144 (2016)
46. Tzeng, E., et al.: Adversarial discriminative domain adaptation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. Honolulu, July 21–26. IEEE Computer Society, pp. 2962–2971 (2017)
47. Pei, Z., et al.: Multi-adversarial domain adaptation. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18). AAAI Press, pp. 3934–3941 (2018)
48. Pinheiro, P.O.: Unsupervised domain adaptation with similarity learning. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City. IEEE Computer Society, pp. 8004–8013 (2018)
49. Long, M., et al.: Deep transfer learning with joint adaptation networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning (ICML 2017). Vol. 70 of Proceedings of Machine Learning Research, pp. 2208–2217 (2017)
50. Sankaranarayanan, S., et al.: Generate to adapt: Aligning domains using generative adversarial networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018). IEEE Computer Society, pp. 8503–8512 (2018)
51. Liu, H., et al.: Transferable adversarial training: A general approach to adapting deep classifiers. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning (ICML 2019). Vol. 97 of Proceedings of Machine Learning Research, pp. 4013–4022 (2019)
52. Chen, X., et al.: Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning (ICML 2019). Vol. 97 of Proceedings of Machine Learning Research, pp. 1081–1090 (2019)
53. Saenko, K., et al.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision - ECCV 2010, 11th European Conference on Computer Vision. Proceedings, Part IV. Vol. 6314 of Lecture Notes in Computer Science. Springer, Berlin, pp. 213–226 (2010)
54. Hoffman, J., et al.: Cycada: Cycle-consistent adversarial domain adaption. In: ICML. Vol. 80 of Proceedings of Machine Learning Research, pp. 1994–2003 (2018)
55. Paszke, A., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, pp. 8026–8037 (2019)
56. Liu, M., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: NIPS, pp. 700–708 (2017)
57. Maaten, L.V.D., Hinton, G.E.: Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 2579–2605 (2008)