Chapter

Transfer Learning and Deep Domain Adaptation

Wen Xu, Jing He and Yanfeng Shu

Abstract

Transfer learning is an emerging technique in machine learning, by which we can solve a new task with the knowledge obtained from an old task in order to address the lack of labeled data. In particular deep domain adaptation (a branch of transfer learning) gets the most attention in recently published articles. The intuition behind this is that deep neural networks usually have a large capacity to learn representation from one dataset and part of the information can be further used for a new task. In this research, we firstly present the complete scenarios of transfer learning according to the domains and tasks. Secondly, we conduct a comprehensive survey related to deep domain adaptation and categorize the recent advances into three types based on implementing approaches: fine-tuning networks, adversarial domain adaptation, and sample-reconstruction approaches. Thirdly, we discuss the details of these methods and introduce some typical real-world applications. Finally, we conclude our work and explore some potential issues to be further addressed.

Keywords: transfer learning, deep domain adaptation, fine-tuning, adversarial domain adaptation, sample-reconstruction

1. Introduction

Inspired by the biological neurons, deep neural networks are well known for their ability to learn data representation from a huge amount of labeled data such as the famous convolutional neural networks (CNNs). Specifically, given a specific task such as image classification, we usually need to train a deep neural network from scratch with enough training data so that our model can achieve acceptable performance. However, sufficient training data for a new task is not always available as manually collecting and annotating data are labor-intensive and expensive. Especially in some specific domains such as healthcare, a privacy concern is also raised. Meanwhile, training a deep network with a large dataset is usually time-consuming and involves huge computational resources. Intuitively, it is not realistic and practical to learn from zero, because the real way we humans learn is that we usually try to solve a new task based on the knowledge obtained from past experiences. For example, once we have learned a programming language (e.g., Java), we can easily learn a new one (e.g., Python) as the basic programming fundamentals are the same.

Transfer Learning is an inspiring method that can help apply the knowledge gained from a source task to a new/target task. Specifically, the goal of transfer learning is to obtain some transferable representations between the source domain

and target domain and utilize the stored knowledge to improve the performance on the target task. Note that transfer learning is an extensive research topic that involves many learning methods. In particular, deep domain adaptation gets the most attention in recent years among these methods. Therefore, after briefly introducing the transfer learning in this research, we pay our attention to analyzing and discussing the recent advances in deep domain adaptation.

The rest of this chapter is structured as follows. In Section 2, we give an overview and specific definitions of transfer learning. In Section 3, we summarize the main approaches for deep domain adaptation. In Section 4, 5 and 6, we discuss the details for conducting deep domain adaptation. The recent applications based on deep domain adaptation methods are also introduced in Section 7. Finally, we conclude this research and discuss future trends in Section 8.

2. Overview

We first give some notations and definitions which match those from the survey paper written by Pan et al. [1], and these notations are also widely adopted in many other survey papers such as [2, 3].

Definition 1 (**Domain** [1]) Given a specific dataset $X = \{X_1, \ldots, X_n\} \in \mathcal{X}$, where $\mathcal{X}$ denotes the feature space, and a marginal probability distribution on the dataset $P(X)$. A domain can be defined as $\mathcal{D} = \{\mathcal{X}, P(X)\}$. Therefore, a domain consists of two components: the feature space and the marginal probability distribution on the dataset.

Definition 2 (**Task** [1]) Given a specific dataset $X = \{X_1, \ldots, X_n\} \in \mathcal{X}$ and their labels $Y = \{Y_1, \ldots, Y_n\} \in \mathcal{Y}$, where $\mathcal{Y}$ denotes the label space. A task can be defined as $\mathcal{T} = \{\mathcal{Y}, \mathcal{F}(X)\}$, where $\mathcal{F}$ is an objective predictive function to learn, which can be seen as a conditional distribution $P(Y|X)$.

Definition 3 (**Transfer Learning** [1]) Given a source domain $\mathcal{D}_s$ and its corresponding task $\mathcal{T}_s$, where the learned function $\mathcal{F}_s$ can be interpreted as some knowledge obtained in $\mathcal{D}_s$ and $\mathcal{T}_s$. Our goal is to get the target predictive function $\mathcal{F}_t$ for target task $\mathcal{T}_t$ with target domain $\mathcal{D}_t$. Transfer learning aims to help improve the performance of $\mathcal{F}_t$ by utilizing the knowledge $\mathcal{F}_s$, where $\mathcal{D}_s \neq \mathcal{D}_t$ or $\mathcal{T}_s \neq \mathcal{T}_t$.

In short, transfer learning can be simply denoted as

$$\mathcal{D}_s, \mathcal{T}_s \rightarrow \mathcal{D}_t, \mathcal{T}_t \tag{1}$$

Transfer learning is a very broad research subject in machine learning. In this research, we mainly focus on transfer learning based on deep neural networks (i.e., deep learning). Therefore, as shown in **Figure 1**, based on $\mathcal{D}_s \neq \mathcal{D}_t$ or $\mathcal{T}_s \neq \mathcal{T}_t$, we can have three scenarios when applying transfer learning. Note that when $\mathcal{D}_s = \mathcal{D}_t$ and $\mathcal{T}_s = \mathcal{T}_t$, the problem becomes a traditional deep learning task. In such case, a dataset is usually divided into a training dataset $\mathcal{D}_s$ and a test training dataset $\mathcal{D}_t$, then we can train a neural network $\mathcal{F}$ on $\mathcal{D}_s$ and apply the pre-trained model $\mathcal{F}$ to $\mathcal{D}_t$.

When $\mathcal{D}_s = \mathcal{D}_t$ and $\mathcal{T}_s \neq \mathcal{T}_t$, transfer learning is usually transformed into a multi-task learning problem. Since the source domain and the target domain share the same feature space, we can utilize one giant neural network to solve different types of tasks at the same time. For example, multi-task learning is widely used in the autopilot system. Given an input image, we can utilize a deep neural network that has enough capacity to recognize the cars, the pedestrians, traffic signs, and the locations of these objectives in the image.

When $\mathcal{D}_s \neq \mathcal{D}_t$ and $\mathcal{T}_s = \mathcal{T}_t$, deep domain adaptation technique is usually used to transfer the knowledge from the source to the target. In general, the goal of domain adaptation is to learn a mapping function $\mathcal{F}$ to reduce the domain

DOI: http://dx.doi.org/10.5772/intechopen.94072

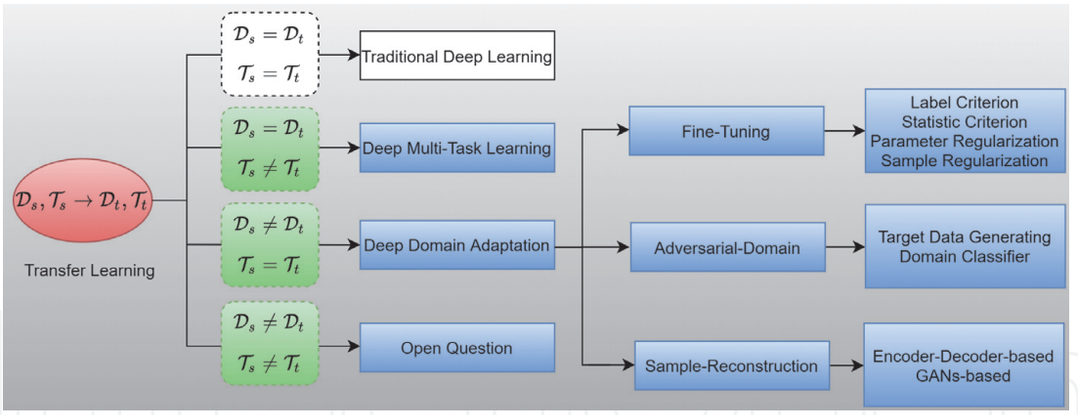

Figure 1.
Hierarchically-structured taxonomy of transfer learning in this survey.

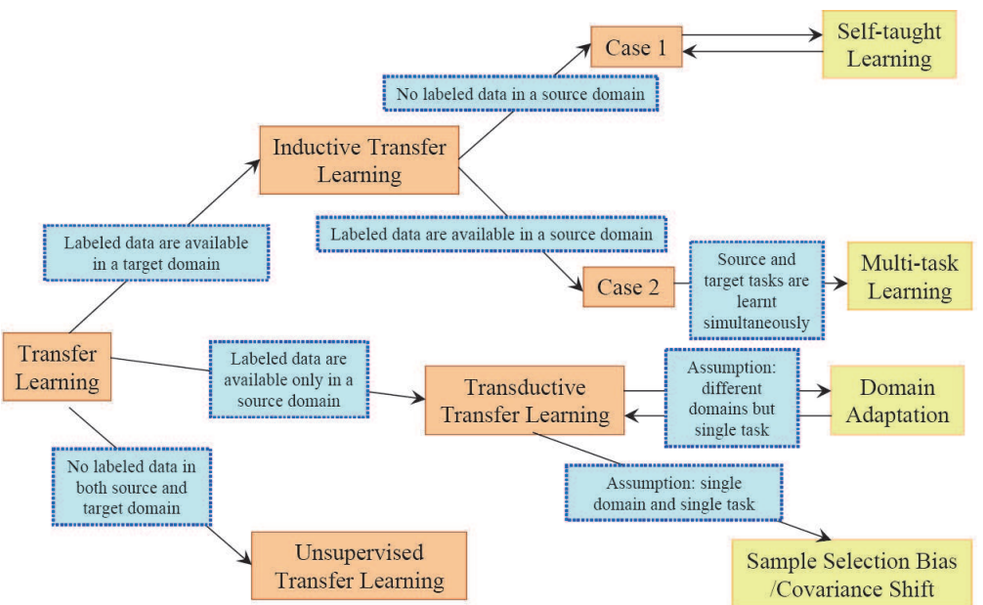

Figure 2.
Categorization of transfer learning based on labels. (The image is from Pan [1]).

divergence between $\mathcal{D}_s$ and $\mathcal{D}_t$ including distribution shift and different feature spaces. Formally, the definition of domain adaptation can be defined as.

Definition 4 (**Domain Adaptation**) Given a source domain $\mathcal{D}_s$ for task $\mathcal{T}_s$ and a target domain $\mathcal{D}_t$ for task $\mathcal{T}_t$, where $\mathcal{D}_s \neq \mathcal{D}_t$. Domain adaptation aims to learn a predictive function $\mathcal{F}_t$ so that the knowledge obtained from $\mathcal{D}_s$ and $\mathcal{T}_s$ can be used for enhancing $\mathcal{F}_t$. In other words, the domain divergence is adapted in $\mathcal{F}_t$.

When $\mathcal{D}_s \neq \mathcal{D}_t$ and $\mathcal{T}_s \neq \mathcal{T}_t$, transfer learning should be conducted carefully. If the data in source domain $\mathcal{D}_s$ is very different from that in target domain $\mathcal{D}_t$, brute-force transfer may hurt the performance of predictive function $\mathcal{F}_t$, not to mention the scenario when source task $\mathcal{T}_s$ and target task $\mathcal{T}_t$ are also different. From a literature review of deep learning, we notice that there is little research in this scenario and it is still an open question.

In summary, the above definitions give us the answer to what to transfer, and the four scenarios demonstrate the research issue of when to transfer. As shown in **Figure 2**, in contrast to the categorization of transfer learning that is introduced in the survey paper [1], our discussions mainly focus on transfer learning in deep neural networks. In the following sections, we pay our attention to how to transfer. Specifically, we will introduce and summarize three main methods for deep domain adaptation.

3. Deep domain adaptation

According to the definition of domain adaptation, we assume that the tasks of the source domain and target domain are the same, and the data in the source domain and target domain are different but related (i.e., $\mathcal{D}_s \neq \mathcal{D}_t$ and $\mathcal{T}_s = \mathcal{T}_t$). In general, the goal of domain adaptation is to reduce the domain distribution discrepancy between the source domain and the target domain so that the knowledge learned from the source domain can be further applied to the target domain.

Compared with the traditional shallow method, deep domain adaptation mainly focuses on utilizing deep neural networks to improve the performance of the predictive function $\mathcal{F}_t$. Formally, a neural network can be denoted as

$$\hat{Y} = \mathcal{F}(X; \Theta) \tag{2}$$

where $\mathcal{F}$ denotes a neural network and Θ is a set of parameters, $\hat{Y}$ represents the predicted label of input X. The deep neural architecture is usually specifically designed to learn representation with back-propagation from the source and target data for domain adaptation. The intuition behind domain adaptation is that we can find some domain-invariant schemes or sharing features from related datasets. In other words, we ensure that the internal representations learned from related domains in deep neural networks are indiscriminating. In this section, based on the published works in recent years, we discuss how to reduce the domain divergence in deep neural networks and categorize deep domain adaptation approaches into three main ways, including fine-tuning networks, domain-adversarial learning, and sample-reconstruction approach.

3.1 Categorization based on implementing approaches

3.1.1 Fine-tuning networks

A natural way to reduce the domain shift is to fine-tune the pre-trained networks with the data in the target domain, as the past researches show that the internal representations of deep convolutional neural networks learned from large datasets, such as ImageNet, can be effectively used for solving a variety of tasks in computer vision. Specifically, for a pre-trained model such as VGG [4] or ResNet [5], we can keep its earlier layers fixed/frozen and only fine-tune the weights in the high-level portion of the network by continuing back-propagation. Or we can fine-tune all the layers if needed. The main idea behind this is that the learned low-level representations in the earlier layers mainly consist of generic features such as the edge detector. During fine-tuning the networks, the discrepancy between the source domain and target domain is usually measured by a criterion such as class labels based criterion, and statistic criterion. Instead of directly using the measurement as a criterion to adjust networks, regularization techniques can also be used for fine-tuning, which mainly includes parameter regularization and sample regularization.

3.1.2 Adversarial domain adaptation

Generative Adversarial Networks (GANs) are a promising method and get the most attention due to its unsupervised learning approach and the flexibility of generator design. Since the first version of GANs is proposed by Goodfellow et al. [6], many variants based on it have been proposed for solving different types of tasks. Specifically, there are normally two networks in GANs, namely a generator

and a discriminator. The generator can synthesize fake examples from an input space called latent space and the discriminator can distinguish real samples from fake. By alternately training these two players, both of them can enhance their abilities. The fundamental idea behind GANs is that we want the data distribution learned by the generator is close to the true data distribution. And this is very similar to the principle of domain adaptation, which is that the learned data distribution between the source domain and the target domain is close to each other (i.e., domain confusion). For example, a representative work related to adversarial domain adaptation is [7], in which a generalized framework based on GANs is introduced. Instead of using GANs for domain-adversarial learning, a more simple but powerful method is to add a domain classifier into a general deep network for encouraging domain confusion [8].

3.1.3 Data-reconstruction approaches

Data-reconstruction approaches are a type of deep domain adaptation method that utilizes the deep encoder-decoder architectures, where the encoder networks are used for the tasks and the decoder network can be treated as an auxiliary task to ensure that the learned features between the source domain and target domain are invariant or sharing. There are mainly two types of methods to conduct data reconstruction: (1) A typical way is by utilizing an encoder-decoder deep network for domain adaptation such as [9]; (2) Another way is to conduct sample reconstruction based on GANs such as cycle GANs [10].

3.1.4 Hybrid approaches

In general, the core idea of deep domain adaptation is to learn indiscriminating internal representations from the source domain and target domain with deep neural networks. Therefore, we can combine different kinds of approaches discussed above to enhance the overall performance. For example, in [11], they adopt both the encoder-decoder reconstruction method and the statistic criterion method.

3.2 Categorization based on learning methods

Based on whether there are labels in the target domain datasets, we can further divide the above approaches into supervised learning and unsupervised learning. Note that the unsupervised learning methods can be generalized and applied to semi-supervised cases, therefore, we mainly discuss these two methods in this research. **Table 1** shows the categorization of deep domain adaptation based on whether the labels are needed in the target domain. A similar categorization is also introduced in [12].

3.3 Categorization based on data space

In some survey papers, the domain adaptation methods can also be categorized into two main methods based on the similarity of data space. (1) Homogeneous domain adaptation represents that the source data space and the target data space is the same (i.e., $\mathcal{X}_s = \mathcal{X}_t$). E.g., the source dataset consists of some images of cars from open public datasets, and the images of cars in the target dataset are manually collected from the real world. (2) Heterogeneous domain adaptation represents that the datasets are from different data space (i.e., $\mathcal{X}_s \neq \mathcal{X}_t$). E.g., text vs. images. **Figure 3** presents the topology that is introduced in [12].

		Supervised	Unsupervised
Fine-tuning	Label criterion	✓	
	Statistic criterion		✓
	Parameter regularization	✓	✓
	Sample regularization	✓	✓
Adversarial-domain	Domain classifier		✓
	Target data generating		✓
Sample-reconstruction	Encoder-decoder-based		✓
	GAN-based		✓

Table 1.
Categorization of deep domain adaptation based on whether the labels in the target domain are available.

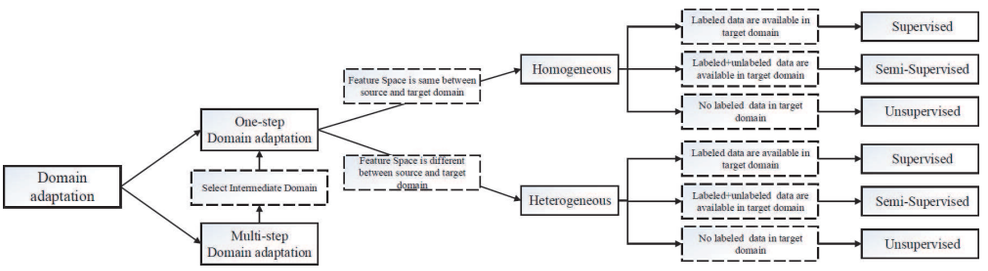

Figure 3.
Categorization of domain adaptation based on feature space. (The image is from Wang [12]).

4. Fine-tuning networks

In the last section, we categorize the main methods to conduct domain adaptation with deep neural networks and give some high-level information. In this section, we firstly discuss the details of four approaches for fine-tuning networks in **Table 1**.

4.1 Label criterion

The most basic approach to conduct domain adaptation is to fine-tune a pre-trained network with labeled data from the target domain. Hence, we assume that the labels in the target dataset are available and we can utilize a supervised learning approach to adjust the weights/parameters in the network. Based on the definition of the task, our target task $\mathcal{T}_t$ based on label criterion approach is

$$\mathcal{T}_t = \mathcal{L}\left(Y_t, \hat{Y}_t\right) = \mathcal{L}(Y_t, \mathcal{F}_t(X_t; \Theta)) \tag{3}$$

where $\mathcal{L}$ denotes a loss function, such as the cross-entropy loss $\mathcal{L}\left(Y, \hat{Y}\right) = -Ylog\left(\hat{Y}\right) - (1-Y)\,log\left(1-\hat{Y}\right)$, which is commonly used in many works. Note that Θ is a set of parameters which is normally initialized with weights from the pre-trained model.

As discussed in Section 3.1, a question is that how many layers in the neural network we should freeze. In general, there are two main factors that can influence the fine-tuning procedure: the size of the target dataset and its similarity to the source domain. Based on the two factors, some common rules of thumb are introduced in [13]. One typical work is [14], in which a unified supervised method for

deep domain adaptation is proposed. Another problem is that what if there are no labels in the target dataset. Therefore, an unsupervised learning method must be applied to the target dataset for domain confusion.

4.2 Statistic criterion

From the definition of domain adaptation, we see that the fundamental goal is to reduce the domain divergence between the source domain and target domain so that the function $\mathcal{F}_t$ can achieve good performance on the target domain. Therefore, it's important and valuable to use a criterion to measure the divergence between different domains. In other words, we need to have a measurement of the difference of probability distributions from different datasets.

Maximum Mean Discrepancy (MMD) [15] is a well-known criterion that is widely adopted in deep domain adaptation such as [16, 17]. Specifically, MMD computes the mean squared difference between the two datasets, which can be defined as

$$\begin{aligned}\mathcal{D}_{MMD}(X^s, X^t) &= \left\| \frac{1}{n}\sum_{i=1}^{n}\phi\left(x_i^s\right) - \frac{1}{m}\sum_{j=1}^{m}\phi\left(x_i^t\right) \right\|^2 \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\phi\left(x_i^s\right)^{\mathrm{T}}\phi\left(x_j^s\right) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\phi\left(x_i^s\right)^{\mathrm{T}}\phi\left(x_i^t\right) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\phi\left(x_i^t\right)^{\mathrm{T}}\phi\left(x_j^t\right) \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}k\left(x_i^s, x_j^s\right) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}k\left(x_i^s, x_j^t\right) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}k\left(x_i^t, x_j^t\right)\end{aligned} \tag{4}$$

where $\phi()$ denotes the feature space map. In practice, we can use the kernel method $k()$ to make MMD be computed easily (i.e., Gaussian kernel).

$\mathcal{H}$-divergence [18] is a more general theory to measure the domain divergence, which is defined as

$$d_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) = 2 \sup_{h \in \mathcal{H}} \left| \Pr_{x_s \sim \mathcal{D}_s}[h(x_s) = 1] - \Pr_{x_t \sim \mathcal{D}_t}[h(x_t) = 1] \right| \tag{5}$$

where $h \in \mathcal{H}$ is a binary classifier (i.e., hypothesis). For example, in [19], domain-adversarial networks are proposed based on this statistic criterion (note that this method can belong to the approach of domain-adversarial learning).

4.3 Parameter regularization

Note that for fine-tuning networks with the label criterion or the statistic criterion, the weights in the networks are usually shared between the source domain and target domain. In contrast to these methods, some researchers argue that the weights for each domain should be related but not shared. Based on this idea, the authors in [20] propose a two-stream architecture with a weight regularization method. Two types of regularizers are introduced: L_2 norm or in an exponential form.

$$\begin{aligned} r_w\left(\theta_j^s, \theta_j^t\right) &= \left\| a_j\theta_j^s + b_j - \theta_j^t \right\| \\ or = \ exp&\left(\left\| a_j\theta_j^s + b_j - \theta_j^t \right\| \right) - 1 \end{aligned} \tag{6}$$

where a_j and b_j are different parameters in each layer. Rather than using two networks for domain adaptation, in [21], they introduce a domain guided method to drop some weights in the networks directly.

4.4 Sample regularization

Alternatively, instead of adapting the parameters in the networks, we can re-weight the data in each layer of feed-forward neural networks. The typical method to reduce internal covariate shit in deep neural networks is to conduct batch normalization during training [22].

$$\hat{x}_i = \gamma \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{7}$$

Note that x_i usually denotes the hidden activation of input sample x_i in each layer of a neural network (e.g., the output feature map of each convolutional layer). $\mu = \frac{1}{B}\sum_{i=1}^{B} x_i$ and $\sigma = \frac{1}{B}\sum_{i=1}^{B}(x_i - \mu)^2$. B is the batch size, γ and β are two hyper-parameters to learn. Based on this method, [23] propose a revised method for practical domain adaptation. And in [24], researchers adopt instance normalization for stylization.

5. Adversarial domain adaptation

Instead of directly fine-tuning networks, adversarial domain adaptation is an appealing alternative to unsupervised learning. It mainly addresses the problem that there are abundant labeled data in the source domain but sparse/limited unlabeled samples in the target domain. The core idea of the adversarial domain adaptation is based on GANs. Specifically, a generalized architecture to implement this idea is proposed in [7]. In this section, we detail two main ideas: target data generating and domain classifier.

5.1 Target data generating

To overcome the limitation of sparse unlabeled data, target data generating is an approach to directly generate samples with labels for the target domain so that we can utilize them to train a classifier for the new task. One representative work is the CoGANs [25], in which there are two GANs involved: one for processing the labeled data in the source domain and another for processing the unlabeled data in the target domain. Part of the weights in the two generators is shared/tied in order to reduce the domain divergence. In addition to two discriminators for classifying the fake and real samples, there is also an extra classifier to classify the samples based on the information of labels in the source domain. By jointly training these two GANs, we can generate unlimited pairs of data, in which each pair consists of a synthetic source sample and a synthetic target sample and each pair shares the same label. Therefore, after finishing jointly training the two GANs, the pre-trained extra classifier is the function $\mathcal{F}_t$ that we need for solving the new task. Similar work can also be found in [26], in which a transformation in the pixel space is introduced.

In summary, target data generating is a domain adaptation approach that focuses on generating target data, which can also be treated as an auxiliary task to reduce domain shift by a weight sharing mechanism between two GANs. The main disadvantage is that the training cost for generating synthesized samples with two GANs

is expensive especially when the target datasets consist of large-size samples such as high-resolution images.

5.2 Domain classifier

Instead of directly synthesizing labeled data for domain adaptation, an alternative way is to add an extra domain classifier to enough domain confusion. The role of domain classifier is similar to that of the discriminator in GANs, it can distinguish the data between the source domain and target domain (the discriminator in GANs is responsible for recognizing the fake from the real data). With the help of an adversarial learning approach, the domain classifier can help the network learn domain-invariant representation from the source domain and the target domain. In other words, the trained model can be directly used for the target/new task.

Therefore, the key is how to conduct adversarial learning with the domain classifier. In [8], a gradient reversal layer (GRL) before domain classifier is introduced to maximize the gradients for encouraging domain confusion (we normally minimize the gradients for reducing the scalar value of a loss function). In [27], a domain confusion loss is proposed beside the domain classifier loss.

6. Sample-reconstruction approaches

The core idea of the data-reconstruction approach is to utilize the reconstruction as an auxiliary task for encouraging domain confusion in an unsupervised manner. In this section, we discuss two types of approaches that are mainly addressed in recent years, including the encoder-decoder-based method and the GANs-based method.

6.1 Encoder-decoder-based approaches

To reconstruct the samples, the basic method is that we can adopt an auto-encoder framework, in which there is an encoder network and decoder network. The encoder can map an input sample into a hidden representation and the decoder can reconstruct the input sample based on the hidden representation. In particular, the encoder-decoder networks for domain adaptation typically involve a shared encoder between the source domain and target domain so that the encoder can learn some domain-invariant representation. An earlier work can be found in [9], in which the stacked denoising auto-encoder is adopted for sentiment classification.

Recently, a typical work called deep reconstruction-classification networks is introduced in [11], in which the encoder and decoder are both implemented with convolutional networks. Specifically, the convolutional encoder is used for supervised classification of the labeled data from the source domain. Meanwhile, it also maps the unlabeled data from the target domain into hidden representation, which is further decoded by the convolutional encoder for reconstructing the input. By jointly training these networks with the data from the source and target domains, the shared encoder can learn some common representations from both datasets, which results in domain adaptation. Other similar work based on auto-encoder can also be found in [11, 28].

6.2 GAN-based approaches

Traditionally, the GANs [6] consists of a generator and discriminator, where the generator can be seen as a decoder network which can decode some random noise

into a fake sample and the discriminator can be treated as an encoder network which is used to encode the sample into some high-level features for classification (i.e., fake or real). Instead of just using a decoder network as the generator, a typical work known as Cycle GANs is proposed in [10], in which the generator is implemented with an encoder-decoder network. Specifically, this encoder-decoder generator is used for dual learning: $G(x_s) \rightarrow x_t$ and $F(x_t) \rightarrow x_s$. And the discriminator also has two roles: to distinguish between the fake x_t and real x_t, and to distinguish between the fake x_s and real x_s. By alternatively training these two players in GANs, the encoder-decoder generator can lean a reversible mapping function. In other words, the domain-invariant features are obtained from two different datasets. However, one remaining problem is that the encoder-decoder network usually consists of millions of parameters, with enough capacity, it can map an input image from the source domain to any random image which is close to the target domain. Therefore, in addition to using the standard adversarial loss for training the GANs, the consistency loss (i.e., L1 norm) is also proposed to make sure that $F(G(x)) \approx x$.

$$\mathcal{L}_{cyc}(G,F) = \mathbb{E}_{x_s \sim data(x_s)} \|F(G(x_s)) - x_s\|_1 + \mathbb{E}_{x_t \sim data(x_t)} \|F(G(x_t)) - x_t\|_1 \quad (8)$$

where $G(x_s)$ denotes fake x_t and $F(G(x_s))$ is reconstructed x_s (i.e., $F(x_t) \rightarrow x_s$). Inspired by the Cycle GANs, many variants based on encoder-decoder generator are proposed for domain adaptation, such as the Disco GANs [29] and the Dual GANs [30].

7. Applications

As shown in **Figure 1**, the scope of transfer learning is far beyond traditional machine learning. Theoretically, the problems addressed by deep learning can also be solved by transfer learning. In this section, we narrow the discussion to the typical real-world applications based on deep domain adaptation. In Section 7.1, we summarize the most methods discussed above for computer vision. In Section 7.2, we discuss the applications beyond the context of image processing, including natural language processing, speech recognition and other real-world applications based on processing time-serial data.

7.1 Applications in computer vision

7.1.1 Image classification and recognition

Classification is a fundamental and most basic problem in machine learning, most of the above methods are introduced to address this problem. Therefore, we pay our attention to the advances that deep domain adaptation can bring for image classification, rather than repeatedly introducing them. Probably the most well-known example is fine-tuning a giant network that is pre-trained with the ImageNet dataset for real-world applications such as pet recognition. Despite the fact that manually collecting data is time-consuming and expensive, the data collected from the real-world is usually imbalanced (e.g., there are only 100 images of class A but 10,000 images of class B). If we train a classifier from scratch, the performance can be poor because it cannot learn enough knowledge from the limited samples (e.g., class A). However, if we utilize a pre-trained model based on the well-collected ImageNet and fine-tune it, the problem caused by an imbalance dataset will be

reduced because the model has already obtained rich knowledge from the source domain.

Another typical real-world application that we can gain benefits from domain adaptation is face recognition. A general approach to solve this problem is to train a model based on a dataset of labeled face images. In contrast, the large-scale unlabeled video datasets are always available. However, the divergence of data in the video is usually limited and there remains a clear gap between these two different domains. In order to utilize the rich information from video and overcome these challenges, the authors in [31] propose a framework for face recognition in unlabeled video based on the adversarial domain adaptation approach.

7.1.2 Object detection

The recent object detection methods are mainly driven by two approaches: Faster R-CNN [32] and YOLO [33]. Specifically, two tasks are mainly involved in object detection: The first one is to detect whether there are objects in an input image (i.e., to output the bounding box of each object in the image); Meanwhile, the object in each bounding box is also classified. Object detection is a very common learning task in many real-world applications such as intelligent surveillance systems [34]. By utilizing domain adaptation approaches for the new task of object detection in the wild, the Domain Adaptive Faster R-CNN is introduced in [35]. And the core idea is also to utilize domain classifier with GRL to encourage domain confusion (i.e., in Section 5.2). Another recent similar work is also discussed in [36], in which the GRL is also adopted and the process of conducting domain adaptation is divided into two stages called progress domain adaptation.

7.1.3 Image segmentation

The convolutional encoder-decoder architecture has achieved great success for image segmentation in recent years. Specifically, given an input image, the convolutional encoder-decoder network can map this image into a pixel-level classification image (i.e., each pixel is classified with a label). The problem of domain shifts can also appear in this task, which results in poor performance on a new domain. In [37], the researchers introduce a domain adversarial learning method which includes both global and category-specific techniques. They argue that two factors can cause domain shift: the global changes between the two distinct domains and the category-specific changes. (i.e., the distribution of cars from two different cities may be different.) Based on this assumption, two new loss functions are introduced, one is used for reducing the global distribution shift between the source images and target images and the other is used for adapting the category-specific divergence between the target images and the transferring label statistics. Instead of just using a simple adversarial objective, the authors in [38] propose an iterative optimization procedure based on GANs for addressing domain shift.

7.1.4 Image-to-image translation

As mentioned in Section 6.2, Cycle GANs [10] is a typical method for image-to-image translation based on deep domain adaptation. In general, image-to-image translation denotes that we can map an image from the source domain to the target domain and vice versa. One real task that is also addressed in Cycle GANs is the style transfer application. To our best knowledge, the algorithm of neural style transfer is firstly proposed in [39], the core idea in this paper is how to define the content loss and style loss between the source data and the target data. Actually, it

can be treated as a statistic criterion approach which is discussed in Section 4.2. In the paper of demystifying neural style transfer [40], the authors show that matching Gram matrices (i.e., style loss) is equivalent to minimize the MMD (i.e., Eq. 4). Based on this argument, they introduce several style transfer methods by utilizing different types of kernel functions in the MMD and achieve impressive results.

7.1.5 Image caption

An interesting but challenging task is to utilize deep neural networks to describe an input image with natural language, which is well known as the image caption. Specifically, the goal of image caption is to learn a mapping function $\mathcal{F}_t$, so that we can get $\mathcal{F}_t(Image) \rightarrow Text$ and vice versa. Note that there are two different data space involved in this task: a dataset with images vs. a dataset with text. Therefore, based on the categorization methods which are discussed in Section 3.3, image caption belongs to heterogeneous domain adaptation. A general method to implement this idea is to utilize a CNN-RNN architecture (i.e., recurrent neural network), where the CNN is used for encoding an input image to some hidden representation and the RNN can decode the representation to some sentences which can describe the content of this image. In particular, the CNN is usually pre-trained based on the ImageNet and then we can re-train it in the CNN-RNN [41].

When we apply an image-caption model which is trained from image dataset A on image dataset B, the performance will degrade due to the distribution change or domain shift of two datasets. To address this problem, the work in [42] introduces an adversarial learning method to address unpaired data in the target domain for image caption (i.e., adversarial domain adaptation approach in Section 5). In [43], the authors propose a dual learning method for addressing this problem, which involves two steps: (1) A CNN-RNN model is trained with sufficient labeled data in the source domain. (2) The model is then fine-tuned with limited target data. The core idea of dual learning mechanism involved a reverse mapping process: the model firstly maps an input target image to text (i.e., $CNN - RNN(Image) \rightarrow Text$) and the text is then mapped back to an image by a generator network, which is further distinguished by a discriminator network. Therefore, the work in [43] belongs to sample-reconstruction approach (i.e., in Section 6).

7.2 Applications beyond computer vision

7.2.1 Natural language processing

Deep domain adaptation technique is also used for solving a variety of tasks in processing natural language. In [44], an effective domain mixing method for machine translation is introduced. The core idea is to jointly train domain discrimination and translation networks. The authors in [45] propose aspect-augmented adversarial networks for text classification. The main idea is to adopt a domain classifier, which has been discussed in Section 5.2. Recently, an interesting research area is to utilize neural models to automatically generate answers based on the input questions, which is also known as questions answering. However, the main challenge to train models is that it is usually difficult to collect a large dataset of labeled question-answer pairs. Therefore, domain adaptation is a natural choice to address this problem. E.g., in [46], a framework called generative domain-adaptive nets is introduced. Specifically, a generative model is used to generate questions from the unlabeled text for enhancing the model performance. Other applications of domain

adaptation can also be found in sentence specificity prediction [47], where the specificity denotes the quality of a sentence that belongs to a specific subject.

7.2.2 Speech recognition

A typical real-world application is to transcribe speech into text, which is also known as automatic speech recognition. Domain adaptation is also suitable for addressing the training-testing mismatch of speech recognition that is caused by the shift of data distribution between different datasets. For example, a neural model trained on a manually collected dataset may generalize poorly in the real-world application of speech recognition due to the environmental noises. In [48], an adaptive teacher-student learning method is proposed for domain adaptation in speech recognition systems. In [49], the domain classifier that is discussed above is also adopted for robust speech recognition. Similar work can also be found in [50], in which the adversarial learning method for domain adaptation is also used for addressing the unseen recoding conditions.

7.2.3 Time-series data processing

Domain adaptation can also enhance the performance of processing many other time-series datasets such as healthcare time-series datasets [51], in which the authors present a variational recurrent adversarial method for domain adaptation. The main idea is to learn domain-invariant temporal latent representations of multivariate time-series data. Another real-world task that involves time-series data is to build diver assistant systems. In [52], an auxiliary domain classifier is also adopted to enhance the performance of recurrent neural networks for driving maneuvers anticipation. And the core idea in this paper is also to learn sharing features from different datasets by the domain classifier. An interesting work related to inertial information processing is introduced in [53], in which a novel framework called MotionTransformer is proposed for extracting domain-invariant features of raw sequences.

8. Conclusion

In this chapter, we firstly introduce the background and explain why transfer learning is important for helping learn real-world tasks. Then we give a strict definition of transfer learning and its scope. In particular, we pay our attention to deep domain adaptation, which is a subset of transfer learning and it mainly addresses the situation where we have different but related datasets for a common learning task. Next, we categorize the deep domain adaptation based on three aspects: the specific implementing approaches, the learning methods, and the data space. In general, deep domain adaptation is one type of method that mainly utilizes deep neural networks to reduce the domain shift or data distribution so that we can enhance the performance of the target task with the help of the knowledge obtained from the source domain. Specifically, we mainly discuss the recent advanced methods for domain adaptation from the deep learning community, including fine-tuning networks, adversarial domain adaptation, and data-reconstruction approaches. Finally, we introduce and summarize the typical real-world applications in computer vision from recently published articles, from which we can see that the unsupervised learning approach based on GANs gets the most attention. In addition, we discuss many other applications beyond the context of image processing. And we notice that many deep domain adaptation methods that are

initially proposed for processing images are also suitable for addressing a variety of tasks in natural language processing, speech recognition, and time-series data processing.

Although deep domain adaptation has been successfully used for solving various types of tasks, we should be careful to conduct transfer learning, as brute-force transfer may hurt the performance of our model. The above applications mainly focus on homogeneous domain adaptation, which means that the data between the source domain and the target domain is related and we assume that deep neural networks can find some shared representation from these two domains. However, the data collected from real-world may not always meet this requirement. Therefore, the future challenge is how to apply a heterogeneous domain adaptation method effectively. From the above analyses, we notice that transfer learning has been mainly applied to a limited scale of applications. Therefore, more challenges are also needed to address in the future such as logical inference and graph neural networks based tasks.

Acknowledgements

This work is supported by China Scholarship Council and Data61 from CSIRO, Australia.

Conflict of interest

The authors declare no conflict of interest.

Author details

Wen Xu[1,2], Jing He[1*] and Yanfeng Shu[2]

1 Swinburne University of Technology, Australia

2 Data61, CSIRO, Australia

*Address all correspondence to: jinghe@swin.edu.au

References

[1] Pan SJ, Yang Q. A survey on transfer learning. IEEE Transactions on knowledge and data engineering. 2009 Oct 16;22(10):1345–59.

[2] Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. Journal of Big data. 2016 Dec 1;3(1):9.

[3] Zhang J, Li W, Ogunbona P. Transfer learning for cross-dataset recognition: a survey. arXiv preprint arXiv: 1705.04396. 2017 May.

[4] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409.1556. 2014 Sep 4.

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

[6] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. InAdvances in neural information processing systems 2014 (pp. 2672–2680).

[7] Tzeng E, Hoffman J, Saenko K, Darrell T. Adversarial discriminative domain adaptation. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 7167–7176).

[8] Ganin Y, Lempitsky V. Unsupervised domain adaptation by backpropagation. InInternational conference on machine learning 2015 Jun 1 (pp. 1180–1189).

[9] Ghifary M, Kleijn WB, Zhang M, Balduzzi D, Li W. Deep reconstruction-classification networks for unsupervised domain adaptation. InEuropean Conference on Computer Vision 2016 Oct 8 (pp. 597–613). Springer, Cham.

[10] Zhu JY, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. InProceedings of the IEEE international conference on computer vision 2017 (pp. 2223–2232).

[11] Bousmalis K, Trigeorgis G, Silberman N, Krishnan D, Erhan D. Domain separation networks. InAdvances in neural information processing systems 2016 (pp. 343–351).

[12] Wang M, Deng W. Deep visual domain adaptation: A survey. Neurocomputing. 2018 Oct 27;312: 135–53.

[13] Chu B, Madhavan V, Beijbom O, Hoffman J, Darrell T. Best practices for fine-tuning visual classifiers to new domains. InEuropean conference on computer vision 2016 Oct 8 (pp. 435–442). Springer, Cham.

[14] Motiian S, Piccirilli M, Adjeroh DA, Doretto G. Unified deep supervised domain adaptation and generalization. InProceedings of the IEEE International Conference on Computer Vision 2017 (pp. 5715–5725).

[15] Borgwardt KM, Gretton A, Rasch MJ, Kriegel HP, Schölkopf B, Smola AJ. Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics. 2006 Jul 15;22(14):e49–57.

[16] Long M, Zhu H, Wang J, Jordan MI. Unsupervised domain adaptation with residual transfer networks. InAdvances in neural information processing systems 2016 (pp. 136–144).

[17] Long M, Zhu H, Wang J, Jordan MI. Deep transfer learning with joint adaptation networks. InInternational conference on machine learning 2017 Jul 17 (pp. 2208–2217).

[18] Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW. A theory of learning from different domains. Machine learning. 2010 May 1; 79(1–2):151–75.

[19] Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V. Domain-adversarial training of neural networks. The Journal of Machine Learning Research. 2016 Jan 1;17(1):2096–30.

[20] Rozantsev A, Salzmann M, Fua P. Beyond sharing weights for deep domain adaptation. IEEE transactions on pattern analysis and machine intelligence. 2018 Mar 8;41(4):801–14.

[21] Xiao T, Li H, Ouyang W, Wang X. Learning deep feature representations with domain guided dropout for person re-identification. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 1249–1258).

[22] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv: 1502.03167. 2015 Feb 11.

[23] Li Y, Wang N, Shi J, Liu J, Hou X. Revisiting batch normalization for practical domain adaptation. arXiv preprint arXiv:1603.04779. 2016 Mar 15.

[24] Ulyanov D, Vedaldi A, Lempitsky V. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017 (pp. 6924–6932).

[25] Liu MY, Tuzel O. Coupled generative adversarial networks. InAdvances in neural information processing systems 2016 (pp. 469–477).

[26] Bousmalis K, Silberman N, Dohan D, Erhan D, Krishnan D. Unsupervised pixel-level domain adaptation with generative adversarial networks. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 3722–3731).

[27] Tzeng E, Hoffman J, Darrell T, Saenko K. Simultaneous deep transfer across domains and tasks. InProceedings of the IEEE International Conference on Computer Vision 2015 (pp. 4068–4076).

[28] Ghifary M, Bastiaan Kleijn W, Zhang M, Balduzzi D. Domain generalization for object recognition with multi-task autoencoders. InProceedings of the IEEE international conference on computer vision 2015 (pp. 2551–2559).

[29] Kim T, Cha M, Kim H, Lee JK, Kim J. Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192. 2017 Mar 15.

[30] Yi Z, Zhang H, Tan P, Gong M. Dualgan: Unsupervised dual learning for image-to-image translation. InProceedings of the IEEE international conference on computer vision 2017 (pp. 2849–2857).

[31] Sohn K, Liu S, Zhong G, Yu X, Yang MH, Chandraker M. Unsupervised domain adaptation for face recognition in unlabeled videos. InProceedings of the IEEE International Conference on Computer Vision 2017 (pp. 3210–3218).

[32] Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. InAdvances in neural information processing systems 2015 (pp. 91–99).

[33] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. InProceedings of the IEEE conference

on computer vision and pattern recognition 2016 (pp. 779–788).

[34] Xu W, He J, Zhang HL, Mao B, Cao J. Real-time target detection and recognition with deep convolutional networks for intelligent visual surveillance. InProceedings of the 9th International Conference on Utility and Cloud Computing 2016 Dec 6 (pp. 321–326).

[35] Chen Y, Li W, Sakaridis C, Dai D, Van Gool L. Domain adaptive faster r-cnn for object detection in the wild. InProceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 3339–3348).

[36] Hsu HK, Yao CH, Tsai YH, Hung WC, Tseng HY, Singh M, Yang MH. Progressive domain adaptation for object detection. InThe IEEE Winter Conference on Applications of Computer Vision 2020 (pp. 749–757).

[37] Hoffman J, Wang D, Yu F, Darrell T. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv: 1612.02649. 2016 Dec 8.

[38] Sankaranarayanan S, Balaji Y, Jain A, Nam Lim S, Chellappa R. Learning from synthetic data: Addressing domain shift for semantic segmentation. InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018 (pp. 3752–3761).

[39] Gatys LA, Ecker AS, Bethge M. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576. 2015 Aug 26.

[40] Li Y, Wang N, Liu J, Hou X. Demystifying neural style transfer. arXiv preprint arXiv:1701.01036. 2017 Jan 4.

[41] Johnson J, Karpathy A, Fei-Fei L. Densecap: Fully convolutional localization networks for dense captioning. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 4565–4574).

[42] Chen TH, Liao YH, Chuang CY, Hsu WT, Fu J, Sun M. Show, adapt and tell: Adversarial training of cross-domain image captioner. InProceedings of the IEEE international conference on computer vision 2017 (pp. 521–530).

[43] Zhao W, Xu W, Yang M, Ye J, Zhao Z, Feng Y, Qiao Y. Dual learning for cross-domain image captioning. InProceedings of the 2017 ACM on Conference on Information and Knowledge Management 2017 Nov 6 (pp. 29–38).

[44] Britz D, Le Q, Pryzant R. Effective domain mixing for neural machine translation. InProceedings of the Second Conference on Machine Translation 2017 Sep (pp. 118–126).

[45] Zhang Y, Barzilay R, Jaakkola T. Aspect-augmented adversarial networks for domain adaptation. Transactions of the Association for Computational Linguistics. 2017 Dec;5:515–28.

[46] Yang Z, Hu J, Salakhutdinov R, Cohen WW. Semi-supervised qa with generative domain-adaptive nets. arXiv preprint arXiv:1702.02206. 2017 Feb 7.

[47] Ko WJ, Durrett G, Li JJ. Domain agnostic real-valued specificity prediction. InProceedings of the AAAI Conference on Artificial Intelligence 2019 Jul 17 (Vol. 33, pp. 6610–6617).

[48] Meng Z, Li J, Gaur Y, Gong Y. Domain adaptation via teacher-student learning for end-to-end speech recognition. In2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) 2019 Dec 14 (pp. 268–275). IEEE.

[49] Sun S, Zhang B, Xie L, Zhang Y. An unsupervised deep domain adaptation approach for robust speech recognition. Neurocomputing. 2017 Sep 27;257:79–87.

[50] Denisov P, Vu NT, Font MF. Unsupervised domain adaptation by adversarial learning for robust speech recognition. InSpeech Communication; 13th ITG-Symposium 2018 Oct 10 (pp. 1–5). VDE.

[51] Purushotham S, Carvalho W, Nilanon T, Liu Y. Variational recurrent adversarial deep domain adaptation.

[52] Tonutti M, Ruffaldi E, Cattaneo A, Avizzano CA. Robust and subject-independent driving manoeuvre anticipation through Domain-Adversarial Recurrent Neural Networks. Robotics and Autonomous Systems. 2019 May 1;115:162–73.

[53] Chen C, Miao Y, Lu CX, Xie L, Blunsom P, Markham A, Trigoni N. Motiontransformer: Transferring neural inertial tracking between domains. InProceedings of the AAAI Conference on Artificial Intelligence 2019 Jul 17 (Vol. 33, pp. 8009–8016).