Neural Networks 138 (2021) 98–109

Contents lists available at ScienceDirect

Neural Networks

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

Towards effective deep transfer via attentive feature alignment

Zheng Xie [a,b,1], Zhiquan Wen [a,b,1], Yaowei Wang [b,1], Qingyao Wu [a,*], Mingkui Tan [a,*]

[a] South China University of Technology, China
[b] PengCheng Laboratory, China

ARTICLE INFO

Article history:
Received 23 June 2020
Received in revised form 17 January 2021
Accepted 25 January 2021
Available online 10 February 2021

Keywords:
Deep transfer
Knowledge distillation
Attention mechanism

ABSTRACT

Training a deep convolutional network from scratch requires a large amount of labeled data, which however may not be available for many practical tasks. To alleviate the data burden, a practical approach is to adapt a pre-trained model learned on the large source domain to the target domain, but the performance can be limited when the source and target domain data distributions have large differences. Some recent works attempt to alleviate this issue by imposing feature alignment over the intermediate feature maps between the source and target networks. However, for a source model, many of the channels/spatial-features for each layer can be irrelevant to the target task. Thus, directly applying feature alignment may not achieve promising performance. In this paper, we propose an Attentive Feature Alignment (AFA) method for effective domain knowledge transfer by identifying and attending on the relevant channels and spatial features between two domains. To this end, we devise two learnable attentive modules at both the channel and spatial levels. We then sequentially perform attentive spatial- and channel-level feature alignments between the source and target networks, in which the target model and attentive module are learned simultaneously. Moreover, we theoretically analyze the generalization performance of our method, which confirms its superiority to existing methods. Extensive experiments on both image classification and face recognition demonstrate the effectiveness of our method. The source code and the pre-trained models are available at https://github.com/xiezheng-cs/AFAhttps://github.com/xiezheng-cs/AFA.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

Deep convolutional neural networks (CNNs) (He, Zhang, Ren, & Sun, 2016; Ronneberger, Fischer, & Brox, 2015; Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018; Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016) have been widely applied in various computer vision tasks. As a deep model often contains millions of parameters, training a deep convolutional network from scratch usually requires a large-scale labeled data set (*e.g.,* ImageNet Deng et al., 2009). However, in real-world scenarios, only limited labeled data are available due to expensive labeling costs or scarcity of the data of interest. Fortunately, as a deep model essentially spans the data into a very high-dimensional space, the feature representations derived from a model (*i.e.,* the source model) pre-trained on a large-scale source data set can be transferred to the related target tasks (Yosinski, Clune, Bengio, & Lipson, 2014).

Fine-tuning is a direct method for deep transfer, and it seeks to obtain a target model by directly retraining the source model on the target data. However, the performance is limited due to the discrepancy between the source and target domain data distributions. This issue will be even more severe when the labeled target data are limited. In the last several years, several advanced deep transfer techniques have been proposed to alleviate the domain discrepancy issue. Specifically, L^2-*SP* (Li, Grandvalet, & Davoine, 2018) regularized the parameters between the two networks to encourage the target network to be similar to the source network. However, simply using regularization may not achieve promising transfer performance. In particular, if the regularization is too weak or too strong, it shall hamper the generalization performance (Li, Xiong, et al., 2019). Zhao et al. (2019) proposed Soft Fine-tuning to maintain general discrimination by holding the previous loss and removing it softly. However, this method requires the source data, which often are unavailable.

More recently, motivated by the knowledge distillation paradigm (Romero, Ballas, et al., 2015), Li et al. (2019) proposed a method called DELTA, which attempted to align the feature maps with a channel attention mechanism. However, DELTA has two main limitations. First, it considers only the transferability of features at the channel level but ignores the redundancy in the spatial regions of the feature maps. In fact, since the target domain and source domain are often different, the feature maps generated by the source model may have considerable redundant

* Corresponding authors.
E-mail addresses: sexiezheng@mail.scut.edu.cn (Z. Xie), sewenzhiquan@mail.scut.edu.cn (Z. Wen), wangyw@pcl.ac.cn (Y. Wang), qyw@scut.edu.cn (Q. Wu), mingkuitan@scut.edu.cn (M. Tan).

[1] Equal contribution.

https://doi.org/10.1016/j.neunet.2021.01.022
0893-6080/© 2021 Elsevier Ltd. All rights reserved.

information (such as the background) to the target task, which may affect transfer performance. Second, in DELTA, the channel attention is pre-learned and fixed for all samples when updating the target model. Since the true informative channels may vary for different samples, the shared attention across samples may be suboptimal for deep transfer.

Note that each layer of a deep model often contains many channels and produces many feature maps. As the target domain is often, to some extent, different from the source domain, not all the channels are relevant or informative for the discriminative power in the target domain. Moreover, such a redundancy issue also exists in the spatial regions of the feature maps. Therefore, when performing transfer learning with deep convolutional networks, it is important to identify and exclude irrelevant information and focus more on the informative channels and spatial features for each sample. Based on this intuition, in this paper, we propose a simple but effective method named Attentive Feature Alignment (AFA) to align the relevant feature maps at both the spatial and channel levels. In particular, we exploit the attention mechanism and devise learnable attentive modules at both the channel and spatial levels. Moreover, we propose a two-stage training scheme in which we sequentially conduct attentive spatial- and channel-level feature alignments between the source and target networks. Note that we simultaneously train the target model and the corresponding attentive module by minimizing attentive feature alignment loss and cross-entropy loss in each stage.

Our main contributions are summarized as follows:

- We propose an Attentive Feature Alignment (AFA) method, which seeks to identify the relevant information between the source and target domains at both the channel and spatial levels in the feature maps. By attending and performing matching over relevant features, AFA is able to well match two domains and ensures better deep transfer.
- Different from existing attention-based methods relying on pre-learned attentive modules, we devise learnable attentive channel and spatial modules to identify relevant features between the source and target domains for each sample.
- We theoretically analyze the generalization performance of our proposed method, which confirms its superiority. Moreover, extensive experiments on both image classification and face recognition demonstrate the effectiveness of our method compared with several state-of-the-art methods.

2. Related work

2.1. Transfer learning

Transfer learning (Caruana, 1997; Gligic, Kormilitzin, Goldberg, & Nevado-Holgado, 2020; Nahmias, Cohen, Nissim, & Elovici, 2020; Pan & Yang, 2009; Raghu, Sriraam, Temel, Rao, & Kubben, 2020) aims to transfer knowledge across related tasks. It has several different scenarios, such as domain adaptation (Deng, Dong, Liu, Wang, & Men, 2019; Saenko, Kulis, Fritz, & Darrell, 2010; Yang & Zhong, 2020; Zhang et al., 2019; Zhu et al., 2019), multi-task learning (Caruana, 1997; Dorado-Moreno et al., 2020), continual learning (Kirkpatrick, Pascanu, et al., 2017; Li & Hoiem, 2017), and so forth. Yosinski et al. (2014) argued that the feature representations derived from the source task are transferable to the related target tasks. Recently, various approaches have been proposed to promote the development of transfer learning, such as sparse transfer (Liu et al., 2017), filter distribution constraining (Aygun, Aytar, & Kemal Ekenel, 2017) and filter subset selection (Cui, Song, Sun, Howard, & Belongie, 2018; Ge & Yu, 2017). Specifically, Li and Hoiem (2017) proposed the learning without forgetting (LwF) method, which used the target data to retrain networks while preserving the capabilities of the source task. Inspired by LwF, L^2-*SP* (Li et al., 2018) regularized the parameters between the two networks to encourage the target network to be similar to the source network. Zhao et al. (2019) proposed the Soft Fine-tuning method to maintain general discrimination by holding the previous loss and removing it softly. Different from L^2-*SP*, Li et al. (2019) proposed DELTA to impose alignment on the feature maps with a channel attention mechanism but ignored the redundancy in the spatial regions of features. In addition, all samples shared the same pre-learned and fixed attention weights, which may be suboptimal for deep transfer. In this paper, our AFA imposes feature alignment considering the redundancy at both the channel and spatial levels. Moreover, we devise learnable attentive modules to recognize the relevant features between the source and target domains, where each sample has a unique attention weight.

2.2. Knowledge distillation

Knowledge distillation (KD) (Ba & Caruana, 2014) is widely used to transfer knowledge from a teacher network to a small student network. The original idea of KD is to force the output distribution of a student network to mimic that of a teacher network (Hinton, Vinyals, & Dean, 2015). Furthermore, several methods have been proposed to transfer the knowledge from a teacher network to a student network by aligning the feature maps derived from intermediate layers (Romero et al., 2015; Zhuang et al., 2018). Zhuang et al. (2018) adopted feature alignment to perform channel pruning and obtained a well-performing pruned model. Recently, several methods (Chen, Wang, & Zhang, 2018b; Yim, Joo, Bae, & Kim, 2017; Zhang & Peng, 2018) have been proposed to improve the performance of student networks. Specifically, Chen et al. (2018b) considered the relationship between different samples and then made use of the similarities across samples to improve the performance of student networks. Moreover, many studies have adopted KD to achieve a better transfer performance (Huang, Peng, & Yuan, 2017). For example, to transfer knowledge across several tasks, the source and target networks are regarded as teachers and students respectively (Yim et al., 2017; Zagoruyko & Komodakis, 2017). In this paper, we adopt the concept of feature alignment in KD to perform deep transfer. Note that feature alignment in KD is to transfer the knowledge learned from the large teacher network to the small student network, where the two networks focus on the same task and domain. However, in deep transfer, the source and target networks are usually the same network and often focus on different domains. In this way, we seek to adopt this paradigm to effectively transfer relevant knowledge from the source domain to the target domain.

2.3. Attention mechanism

The attention mechanism (Bahdanau, Cho, & Bengio, 2015; Luong, Pham, & Manning, 2015; Vaswani et al., 2017; Xu et al., 2015) has promoted the development of various tasks, such as image classification (Hu, Shen, & Sun, 2018; Wang, Girshick, Gupta, & He, 2018; Woo, Park, Lee, & Kweon, 2018; Xie et al., 2020; Zhu, Li, Yang, & Ye, 2020), semantic segmentation (Chen, Yang, Wang, Xu, & Yuille, 2016; Fu et al., 2019), natural language processing (Vaswani et al., 2017; Yang et al., 2020) and human trajectory prediction (Fernando, Denman, Sridharan, & Fookes, 2018). To be specific, SENet (Hu et al., 2018) assigned different weights to different channels with an attention mechanism to promote the development of image classification and object detection. Wang et al. (2018) proposed a non-local block embedded

with an attention mechanism to effectively capture pixel-wise contextual information. CBAM (Woo et al., 2018) sequentially assigned intermediate feature maps channel- and spatial-level attention weights for adaptive feature refinement. In addition, Chen et al. (2016) adopted an attention mechanism to identify the importance of different scales of the feature maps, and then aggregated them with different attention weights to improve segmentation performance. Fu et al. (2019) employed a self-attention mechanism in both channel and spatial levels parallelly to capture contextual dependencies in scene segmentation. Moreover, Vaswani et al. (2017) proposed a simple but effective network named Transformer which adopted a self-attention mechanism for machine translation. In this paper, we investigate the attention mechanism and devise two learnable attentive modules focusing on the channel and spatial levels. Specifically, CBAM (Woo et al., 2018) and DANet (Fu et al., 2019) also considered two levels but embedded the two-level attentive modules into networks, which introduces additional parameters and computational cost in both the training and inference phase. In contrast, our attentive modules are only adopted in the training phase, and thus introduces no additional computational overhead in the inference phase. Moreover, the architecture of our attentive module is different from that in CBAM (Woo et al., 2018) and DANet (Fu et al., 2019) because these methods focus on different tasks.

3. Problem definition and motivations

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be the training data for the target task, where N is the number of samples. In many real-world applications, we may not have sufficient labeled target data. Moreover, the data distributions of the source and target domains are different. Fortunately, we may have a pre-trained source model M_s obtained on some related large-scale data set (*e.g.,* ImageNet). Without loss of generality, M_t denotes the target model. Let $\mathbf{W}_s$ and $\mathbf{W}_t$ be the parameters of the source and target networks, respectively. Note that we may need to build new classifiers *w.r.t.* the target task by introducing new FC layers with new parameters $\mathbf{W}_t^{\mathrm{FC}}$. Let $f(\mathbf{x}_i, \mathbf{W}_t)$ be the prediction of M_t.

The task of **deep transfer** is to obtain a promising target model M_t for the target domain by effectively exploiting M_s and the limited data in $\mathcal{D}$. A direct method for deep transfer is to fine-tune the model *w.r.t.* $\mathbf{W}_t$ by minimizing a regularized objective (with $\mathbf{W}_t$ being warm-initialized based on $\mathbf{W}_s$) (Li et al., 2018):

$$\sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}_i, \mathbf{W}_t), y_i) + \gamma \mathcal{R}(\mathbf{W}_t), \tag{1}$$

where $\mathcal{L}(\cdot,\cdot)$ refers to the loss function (*e.g.,* cross-entropy loss function), $\mathcal{R}(\mathbf{W}_t)$ is a regularizer (*e.g.,* $\frac{1}{2}\|\mathbf{W}_t\|_{\mathrm{F}}^2$) and γ is a trade-off parameter. From Eq. (1), the direct fine-tuning based approach does not effectively exploit M_s, and hence it can easily lose useful knowledge learned from the source domain, leading to limited performance (Li et al., 2018).

A feasible approach to effectively exploit the knowledge in M_s is to impose feature alignment on the intermediate feature maps between M_s and M_t. Typically, it minimizes the following objective *w.r.t.* $\mathbf{W}_t$:

$$\sum_{i=1}^{N} (\mathcal{L}(f(\mathbf{x}_i, \mathbf{W}_t), y_i) + \alpha \Omega(\mathbf{W}_t, \mathbf{W}_s, \mathbf{x}_i)) + \gamma \mathcal{R}(\mathbf{W}_t), \tag{2}$$

where $\Omega(\mathbf{W}_t, \mathbf{W}_s, \mathbf{x}_i)$ denotes the feature alignment loss regarding sample $\mathbf{x}_i$, and α is a trade-off parameter. Intuitively, $\Omega(\mathbf{W}_t, \mathbf{W}_s, \mathbf{x}_i)$ can be defined as the distance between the intermediate feature maps derived from M_s and M_t:

$$\Omega(\mathbf{W}_t, \mathbf{W}_s, \mathbf{x}_i) = \frac{1}{2}\|\mathrm{F}(\mathbf{x}_i, \mathbf{W}_t) - \mathrm{F}(\mathbf{x}_i, \mathbf{W}_s)\|_{\mathrm{F}}^2, \tag{3}$$

where $\mathrm{F}(\cdot,\cdot)$ denotes the feature maps generated from some layers of a network and $\|\cdot\|_{\mathrm{F}}$ denotes the F-norm. By minimizing $\Omega(\mathbf{W}_t, \mathbf{W}_s, \mathbf{x}_i)$, the knowledge in M_s is expected to be retained in M_t. For convenience, we denote this paradigm as direct feature alignment without attention (DFA).

Note that the DFA paradigm has been extensively studied in knowledge distillation (KD) (Romero et al., 2015; Zagoruyko & Komodakis, 2017), where M_s and M_t often focus on the same task and domain. However, in deep transfer, the target domain is often different from the source domain. Consequently, some channels can be irrelevant or redundant to new task learning. More critically, such redundancy issues inevitably exist in the spatial regions of the feature maps. Thus, directly applying the DFA paradigm for deep transfer may not obtain promising performance. In the following section, we will address these issues by devising an Attentive Feature Alignment (AFA) paradigm for effective deep transfer.

4. Proposed method

Motivated by the attention mechanism (Xu et al., 2015), we devise two learnable attentive modules to identify the relevant features between the source and target domains at both the channel and spatial levels. The general scheme of our method is shown in Fig. 1. For convenience, we relegate the details of each module to the following subsections.

With the introduction of attentive modules, we perform deep transfer and optimize $\mathbf{W}_t$ and attention parameters $\mathbf{W}_a$ by minimizing the attentive objective below:

$$\sum_{i=1}^{N} (\mathcal{L}(f(\mathbf{x}_i, \mathbf{W}_t), y_i) + \alpha \Omega_A(\mathbf{W}_t, \mathbf{W}_s, \mathbf{W}_a, h, \mathbf{x}_i)) + \gamma \mathcal{R}(\mathbf{W}_t), \tag{4}$$

where h is an attention function and Ω_A (with $A \in \{C, S\}$) denotes the attentive feature alignment (AFA) loss. Here, we consider attention at both the channel level ($A = C$) and the spatial level ($A = S$), which are designed to capture the true useful information at the two levels. With our attentive modules, the relevant features between two domains obtain higher concentrations in feature alignment. In this way, these true informative features can be transferred to the target domain.

Note that directly optimizing the two-level attentions simultaneously (*i.e.,* the one-stage AFA) may incur the following training difficulties. First, the channel and spatial attentions parameters are needed to be optimized simultaneously. Second, the two-level attentions may interfere with each other during the training process of the one-stage AFA due to the following possible reasons. Specifically, since channel features are composed of spatial features, the channel and spatial attentions are closely entangled. In the one-stage AFA, in general, we first apply one forward propagation to update features and losses, and then update the model parameters of the spatial and channel attentions subsequently. However, once we update the parameters of the spatial attention, it will affect the channel features and thus the gradient of the channel attention can be outdated, which may incur the sub-optimal solutions (For more details, please see Section 5.5.4). Thus, we apply the attentive feature alignment in a two-stage training scheme. The overall algorithm called Attentive Feature Alignment (AFA) in Algorithm 1, where $\mathbf{W}_a^s$ and $\mathbf{W}_a^c$ denote the spatial- and channel-level attentions parameters, respectively. Here, we first consider deep transfer at the spatial level (Called attentive spatial transfer (AST)) and then conduct deep transfer at the channel level (Called attentive channel transfer (ACT)). Moreover, to prevent obtaining incorrect guidance from the source model, we initialize the source model based on the target model trained in AST before performing ACT (For more details, please refer to Sections 4.3 and 5.5.7).

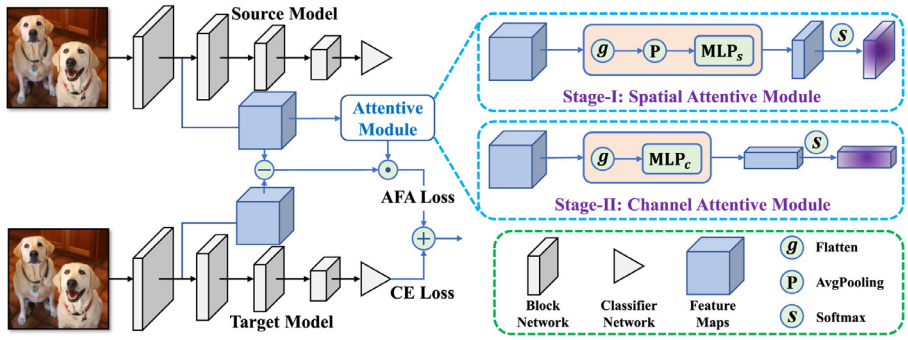


Fig. 1. Overview of the framework of our proposed method. The left part is the framework and the right part is the architecture of attentive modules for two stages. ⊙ denotes element-wise multiplication; ⊖ denotes element-wise subtraction; the AFA loss is the attentive feature alignment loss, and the CE loss is the cross-entropy loss. Note that we evenly take feature maps of four intermediate layers for alignment.

Algorithm 1 Attentive Feature Alignment for Deep Transfer

Input: A pre-trained source model M_s, target training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ and the number of epochs T.
Output: A trained target model M_t.

1: // **Stage I**
2: Let $M_t \leftarrow M_s$ and randomly initialize $\mathbf{W}_a^s$.
3: **for** $\tau = 1, \ldots, T/2$ **do**
4: Update $\mathbf{W}_t$ and $\mathbf{W}_a^s$ via **Attentive Spatial Transfer**.
5: **end for**
6: // **Stage II**
7: Let $M_s \leftarrow M_t$ and randomly initialize $\mathbf{W}_a^c$.
8: **for** $\tau = 1, \ldots, T/2$ **do**
9: Update $\mathbf{W}_t$ and $\mathbf{W}_a^c$ via **Attentive Channel Transfer**.
10: **end for**

4.1. Attentive spatial transfer

The feature maps derived from the source model may contain considerable irrelevant information for the target domain. We thus devise the **Spatial Attentive Module** (SAM) to focus on the informative spatial regions of the feature maps. To this end, it first averages the feature maps in the channel dimension and then adopts a two-layer perceptron MLP_s with a softmax layer to assign the attention weight to the spatial regions. The details of MLP_s are shown in Table 1. Based on SAM, the attention weights $\mathbf{A}^i$ for the ith sample are computed by

$$\mathbf{A}^i = h_s(\mathbf{x}_i, \mathbf{W}_a^s) = \text{softmax}(\text{MLP}_s(\text{AvgPooling}(g(\text{F}(\mathbf{x}_i, \mathbf{W}_s))), \mathbf{W}_a^s)), \tag{5}$$

where h_s is a spatial-level attention function; $\text{F}(\cdot,\cdot)$ denotes the feature maps derived from some layers of a network; $g(\cdot)$: $\mathbb{R}^{C\times H\times W} \rightarrow \mathbb{R}^{C\times(HW)}$ flattens the feature maps in the spatial dimension; $\text{AvgPooling}(\cdot)$: $\mathbb{R}^{C\times(HW)} \rightarrow \mathbb{R}^{1\times(HW)}$ averages the feature maps in the channel dimension, where H, W, C are the height, width, and number of channels of the feature maps, respectively; and $\mathbf{W}_a^s$ is the parameters of MLP_s.

We then assign the spatial attention values $\mathbf{A}^i$ to the pixel-wise differences between the two feature maps, leading to a spatial-level attentive feature alignment loss:

$$\Omega_S(\mathbf{W}_t, \mathbf{W}_s, \mathbf{W}_a^s, h_s, \mathbf{x}_i) = \sum_{j=1}^{H}\sum_{k=1}^{W} \|\text{A}_{j,k}^i(\text{F}_{j,k}(\mathbf{x}_i, \mathbf{W}_t) - \text{F}_{j,k}(\mathbf{x}_i, \mathbf{W}_s))\|_{\text{F}}^2 . \tag{6}$$

By minimizing Eq. (6), AST forces the feature alignment of the target model M_t over the informative spatial regions to the source model M_s for attentive spatial transfer. Note that since the softmax function is used to normalize the attention weights A in Eq. (5), $\text{A}_{j,k}$ in Eq. (6) will be in the interval (0, 1) and all attention weights will add up to 1. In this sense, not all attention weights are close to 0. Thus, the trivial solution ($\text{A} = 0$) will not happen.

Table 1
Structure of the MLPs. Here, H, W, C are the height, width, and the number of channels of the input feature maps, respectively. FC denotes the fully connected layer.

MLP	Input dimension	Operator	Output dimension
MLP_s	HW	FC+ReLU	H
	H	FC	HW
MLP_c	$(HW)\times C$	FC+ReLU	$H\times C$
	$H\times C$	FC	C

4.2. Attentive channel transfer

Apart from paying attention to the important spatial features, a **Channel Attentive Module** (CAM) is proposed to focus on the informative channels of the feature maps. Similar to AST, we use a two-layer perceptron MLP_c with a softmax layer to identify the informative channels. The details of MLP_c are shown in Table 1. Based on CAM, the attention weight vector $\mathbf{a}^i$ for the ith sample can be formulated as

$$\mathbf{a}^i = h_c(\mathbf{x}_i, \mathbf{W}_a^c) = \text{softmax}(\text{MLP}_c(g(\text{F}(\mathbf{x}_i, \mathbf{W}_s)), \mathbf{W}_a^c)), \tag{7}$$

where h_c is a channel-level attention function and $\mathbf{W}_a^c$ denotes the parameters of MLP_c. We align channel features by minimizing the channel-level attentive feature alignment loss:

$$\Omega_C(\mathbf{W}_t, \mathbf{W}_s, \mathbf{W}_a^c, h_c, \mathbf{x}_i) = \sum_{j=1}^{C} \text{a}_j^i \|\text{F}_j(\mathbf{x}_i, \mathbf{W}_t) - \text{F}_j(\mathbf{x}_i, \mathbf{W}_s)\|_{\text{F}}^2 . \tag{8}$$

By minimizing Eq. (8), ACT aligns the informative channels of the target model M_t with that of the source model M_s for attentive channel transfer.

Note that the recently proposed DELTA (Li et al., 2019) method only considers the redundancy issue in channels and adopts a similar approach to ACT. Moreover, in DELTA, the channel attention is pre-learned and fixed for all samples when updating the target model, while in our method, each sample has a unique attention weight matrix/vector. Last, our method updates $\mathbf{W}_t$ and $\mathbf{W}_a$ simultaneously, which is different from DELTA.

4.3. Algorithm details

As shown in Algorithm 1, we adopt a two-stage scheme to train the target network. In Stage I, we perform AST to conduct deep transfer at the spatial level. In Stage II, we initialize the

source model based on the target model trained in Stage I for the incremental training and then perform ACT. If we preserve the original source model (*e.g.,* pre-trained on ImageNet) to conduct Stage II, the original source model may provide the incorrect initial guidance to the target model. The reason is that for the target domain, the target model trained in Stage I has better performance than the original source model. Due to this limitation, we replace the original source model in Stage II. Note that for each stage of AFA, we apply mini-batch SGD to optimize the target model parameters W_t and attention parameters W_a by minimizing the objective in Eq. (4).

4.4. Training and inference complexity

The training cost of our AFA is close to the existing methods (*e.g.,* DELTA) since our attentive modules only introduce a small number of parameters (See Table 6). In other words, the additional parameters of our attentive modules do not greatly increase the training burden. But if adding too many attentive modules to the target model, it will hamper the transfer performance and increase the training time in each epoch (For more details, please refer to Section 5.5.8). In practice, introducing up to four attentive modules is sufficient and is able to improve the performance. Thus, the increased training complexity is acceptable.

In the inference phase, our AFA does not need to consider the attentive modules since it only participates in the training phase. Therefore, given the same network, the target model trained by our AFA has the same inference cost as that trained by other existing deep transfer methods (such as L^2-*SP*, Li et al., 2018 and DELTA, Li et al., 2019), and achieves better performance (See Table 3).

4.5. Differences between AFA and DELTA

Our AFA differs from DELTA in several aspects: (1) DELTA considers only the redundancy at the channel level but ignores potential redundancy at the spatial level, while AFA considers the redundancy at both the channel and spatial levels. (2) In DELTA, the channel attention is pre-learned and fixed for all samples. Since the relevant channels may vary for different samples, the shared attention across samples shall be suboptimal for deep transfer. In AFA, we devise two kinds of learnable attentive modules, each of which provides a unique attention weight for each sample. (3) The experimental results show that our AFA outperforms DELTA in terms of Top-1 accuracy on five public classification data sets (See Table 3).

4.6. Theoretical analysis

In this section, we analyze the generalization bound of our AFA paradigm to measure the generalization performance on unseen test data. First, we introduce some definitions of the expected loss and Rademacher complexity to analyze our proposed method.

Definition 1. Given functions $f \in \mathcal{F}$ and $h \in \mathcal{H}$ for our AFA paradigm, where $\mathcal{F}$ and $\mathcal{H}$ are function spaces, the expected loss can be defined as:

$$E(f, h) = \mathbb{E}[\mathcal{L}(f(\mathbf{x}), \mathbf{y}) + \alpha\Omega(\mathbf{W}_t, \mathbf{W}_s, \mathbf{W}_a, h, \mathbf{x})]\,. \tag{9}$$

From Definition 1, our aim is to learn the functions f and h such that the expected loss on unseen test data can be small. In practice, given N training samples, the goal of our AFA paradigm is to minimize the empirical loss, which can be defined as follows:

$$\hat{E}(f, h) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}_i), y_i) + \alpha\Omega(\mathbf{W}_t, \mathbf{W}_s, \mathbf{W}_a, h, \mathbf{x}_i)\,. \tag{10}$$

To achieve good generalization performance, the empirical loss can be minimized to be small such that the expected loss can also be small. To this end, following Mohri, Rostamizadeh, and Talwalkar (2012), we define the Rademacher complexity of our AFA paradigm as follows.

Definition 2 (*Rademacher Complexity of AFA*)**.** Given an underlying distribution $\mathcal{Z}$ and its empirical distribution $\mathcal{S} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$, where $\mathbf{z}_i = (\mathbf{x}_i, y_i)$. Then the Rademacher complexity of our AFA paradigm is defined as:

$$\mathbf{R}_N^{AFA}(\Phi) = \mathbb{E}_{\mathcal{S}}\left[\hat{\mathbf{R}}_{\mathcal{S}}(f, h)\right],\ \forall f \in \mathcal{F}, h \in \mathcal{H}\,, \tag{11}$$

where $\mathcal{F}$ and $\mathcal{H}$ are function spaces, $\Phi \in \mathcal{F} \times \mathcal{H}$, and $\hat{\mathbf{R}}_{\mathcal{S}}(f, h)$ is the empirical Rademacher complexity. $\hat{\mathbf{R}}_{\mathcal{S}}(f, h)$ is defined as:

$$\hat{\mathbf{R}}_{\mathcal{S}}(f, h) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{(f,h)\in\Phi} \frac{1}{N}\sum_{i=1}^{N} \sigma_i(\mathcal{L}(f(\mathbf{x}_i), y_i) + \alpha\Omega(\mathbf{W}_t, \mathbf{W}_s, \mathbf{W}_a, h, \mathbf{x}_i))\right], \tag{12}$$

where $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_N]$ are independent uniform random variables valued in $\{-1, +1\}$.

From Definition 2, the Rademacher complexity captures the capacity of a family of functions f and h by measuring how they fit random noises. Based on Definitions 1 and 2, we derive a generalization bound of our AFA paradigm as follows:

Theorem 1 (*Generalization Performance of AFA*)**.** *Let* $\mathcal{L}(f(\mathbf{x}), \mathbf{y}) + \alpha\Omega(\mathbf{W}_t, \mathbf{W}_s, \mathbf{W}_a, h, \mathbf{x})$ *be a mapping from* $\mathcal{X}\times\mathcal{Y}$ *to* $[0, U]$ *with upper bound* U *and the function space* Φ *be infinite. For any* $\delta > 0$*, all* $(f, h)\in\Phi$*, with probability at least* $1-\delta$*, the generalization error* $E(f, h)$*, i.e., the expected loss, satisfies*

$$E(f, h) \le \hat{E}(f, h) + 2\mathbf{R}_N^{AFA}(\Phi) + U\sqrt{\frac{1}{2N}\log\left(\frac{1}{\delta}\right)}\,, \tag{13}$$

where $\mathbf{R}_N^{AFA}$ *is the Rademacher complexity of our AFA paradigm. Let* $\mathcal{B}(f, h)$ *be the generalization bound of AFA, i.e.,* $\mathcal{B}(f, h) = 2\mathbf{R}_N^{AFA}(\Phi) + U\sqrt{\frac{1}{2N}\log\left(\frac{1}{\delta}\right)}$*, we thus have*

$$\mathcal{B}(f, h) \le \mathcal{B}(f)\,, \tag{14}$$

where $f \in \Phi$*,* $\mathcal{B}(f)$ *is the generalization bound of the direct feature alignment without attention (DFA) paradigm w.r.t. the Rademacher complexity* $\mathbf{R}_N^{DFA}(\Phi)$*.*

Proof. See Appendix for the detailed proof.

This theorem shows the generalization bound of our AFA paradigm, which relies on the Rademacher complexity of a function space Φ. From Theorem 1, our AFA paradigm has a smaller generalization bound than the DFA paradigm, which helps to achieve better generalization performance. To justify this, we further conduct experiments in Section 5.5.1. The results in Fig. 3 demonstrate that our AFA paradigm has smaller test loss and cross-entropy loss than the DFA paradigm. Therefore, our AFA paradigm achieves better generalization performance than the DFA paradigm. For further discussion, please refer to Remark 1.

Remark 1. Based on the definition of the Rademacher complexity, the capacity of function space $\Phi \in \mathcal{F}\times\mathcal{H}$ is smaller than that of function space $\Phi \in \mathcal{F}$ in the DFA paradigm, i.e., $\mathbf{R}_N^{AFA} \le \mathbf{R}_N^{DFA}$, where $\mathbf{R}_N^{DFA}$ is the Rademacher complexity of the DFA paradigm. In other words, the generalization bound of our AFA paradigm

Table 2
Characteristics of the target data sets: the name, number of classes, and number of samples in the training set and validation set.

Target data sets	# Classes	# Training samples	# Validation samples
MIT Indoors 67	67	5,360	1,340
Stanford Dogs 120	120	12,000	8,580
Caltech 256–30	257	7,710	5,140
Caltech 256–60	257	15,420	5,140
CUB-200–2011	200	5,994	5,794

is smaller than that of the DFA paradigm. Thus, compared with the DFA paradigm, our proposed AFA paradigm is able to achieve better generalization performance.

5. Experiments on image classification

5.1. Source and target data sets

For the image classification task, we choose two large-scale data sets as the source data, namely, ImageNet (Deng et al., 2009) for object classification and Places 365 (Zhou, Lapedriza, Khosla, Oliva, & Torralba, 2017) for scene classification. The small-scale target data sets are five public data sets with different domains: MIT Indoor 67 (Quattoni & Torralba, 2009), Stanford Dogs 120 (Khosla, Jayadevaprakash, Yao, & Li, 2011), Caltech-UCSD Birds-200-2011 (CUB-200-2011) (Wah, Branson, Welinder, Perona, & Belongie, 2011), Caltech 256-30 and Caltech 256-60 (Griffin, Holub, & Perona, 2007). We show the details of these target data sets in Table 2. Note that the target data set MIT Indoor 67 corresponds to the source data set Places 365 and the rest of the target data sets correspond to the source data set ImageNet. Besides, for a fair comparison, we pre-process these data sets following DELTA (Li et al., 2019). For ResNet-101 and MobileNetV2, we first resize input images to 256×256, and then process them to 224×224. For Inception-V3, we first resize input images to 320×320, and finally crop them to 299×299. Note that we use data augmentation operations of random flip and random crop for all networks.

5.2. Implementation details

We use ResNet-101 (He et al., 2016), Inception-V3 (Szegedy et al., 2016) and MobileNetV2 (Sandler et al., 2018) as base networks. For fair comparisons, we follow experimental settings in DELTA[2] (Li et al., 2019). We take pre-trained models from torchvision[3] or CSAILVision[4] as source models. We use mini-batch SGD with a mini-batch size of 64 for optimization, where the momentum term is set to 0.9. Moreover, the number of training iterations is 9k, and the learning rate is divided by 10 at the 6kth iteration. The initial learning rates of ResNet-101 and MobileNetV2 are 0.01, while that for Inception-V3 is 0.001. Moreover, for the LwF and Soft Fine-tuning, the initial learning rate of the three networks is 0.001. For each stage of our AFA, the number of training iterations is 4.5k. In stage I of our AFA, the learning rate is divided by 10 at the 3kth iteration. In stage II, the initial learning rate is the same as the decayed learning rate of stage I. Then, the learning rate of stage II is divided by 10 at the 3kth iteration. For the hyper-parameters α and γ in Eq. (4), we follow the experimental settings of DELTA (Li et al., 2019) and L^2-*SP* (Li et al., 2018). Then, we fix γ to 0.01 and use cross-validation to search for the best α for each experiment. Note that

[2] https://github.com/lixingjian/DELTA
[3] https://pytorch.org/docs/stable/torchvision/index.html
[4] https://github.com/CSAILVision/places365

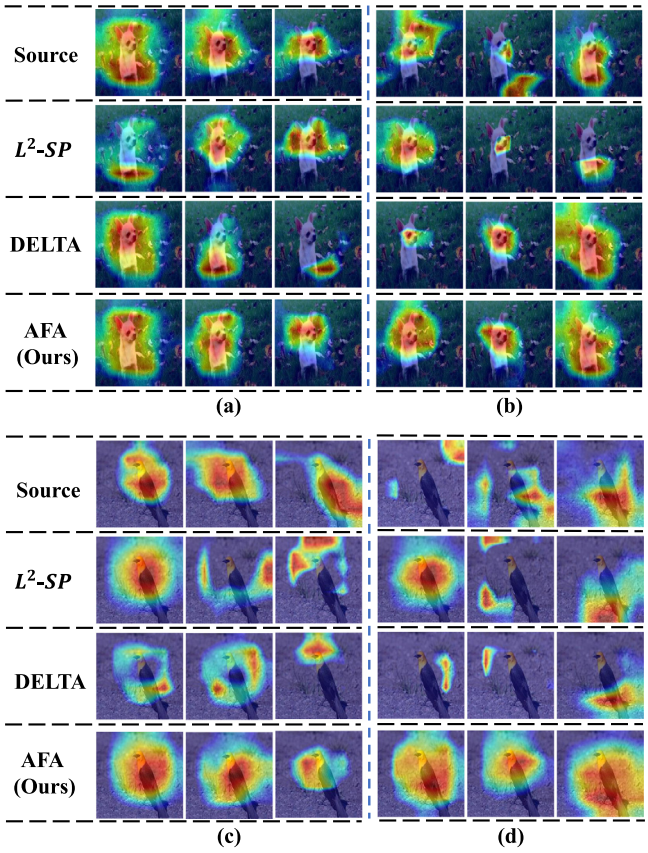


Fig. 2. Visualization of the activation maps of the channels of the block conv5_2 in ResNet-101 for various methods. Examples are taken from the target validation set of Stanford Dogs 120 and CUB-200-2011. Source model: ResNet-101 pre-trained on ImageNet.

for all networks, we take the feature maps from four intermediate layers evenly for feature alignment. The compared methods are L^2, L^2-*FE*, L^2-*SP* (Li et al., 2018), LwF (Li & Hoiem, 2017), Soft Fine-tuning (Soft FT) (Zhao et al., 2019) and DELTA (Li et al., 2019). We repeat each experiment five times to report the average Top-1 accuracy and the standard deviation.

5.3. Comparisons with state-of-the-art methods

5.3.1. Quantitative evaluation

We report the results of our AFA and other compared methods in Table 3. From these results, we have the following observations. First, the four compared methods (L^2-*SP*, LwF, Soft Fine-tuning and DELTA) outperform the naïve L^2 and L^2-*FE* baselines with higher accuracy in most experiments. Second, our AFA achieves much better performance than all the compared methods in three networks on five public image classification data sets. Specifically, for ResNet-101, our AFA outperforms the state-of-the-art method DELTA by 1.7% on CUB-200-2011. For Inception-V3, our AFA surpasses DELTA by 2.9% on Stanford Dogs 120. Moreover, for MobileNetV2, our AFA exceeds DELTA by 1.5% on MIT Indoors 67. These results demonstrate the superiority of our AFA for the image classification task. Third, the state-of-the-art method DELTA outperforms L^2-*SP* by 0.5% $\sim$ 1.5% on five public data sets. Our AFA surpasses L^2-*SP* by 1.0% $\sim$ 3.0%, and outperforms DELTA by 0.5% $\sim$ 2.0%. In this sense, compared with the state-of-the-art method DELTA, the improvements of our AFA are significant.

Table 3
Comparisons of different methods in terms of Top-1 accuracy (%) on five public classification data sets. Compared methods: L^2, L^2-*FE*, L^2-*SP* (Li et al., 2018), LwF (Li & Hoiem, 2017), Soft Fine-tuning (Soft FT) (Zhao et al., 2019) and DELTA (Li et al., 2019). L^2-*FE*: Using the pre-trained model as a feature extractor.

ResNet-101	L^2	L^2-*FE*	L^2-*SP*	LwF	Soft FT	DELTA	AFA (Ours)
MIT Indoors 67	83.7 ± 0.3	80.4 ± 0.2	85.1 ± 0.1	83.9 ± 0.2	84.6 ± 0.3	85.5 ± 0.3	**85.9 ± 0.1**
Stanford Dogs 120	83.3 ± 0.2	84.7 ± 0.1	88.3 ± 0.2	89.4 ± 0.1	88.3 ± 0.1	88.7 ± 0.1	**90.1 ± 0.0**
Caltech 256–30	84.7 ± 0.3	82.9 ± 0.2	85.4 ± 0.2	85.7 ± 0.1	86.3 ± 0.1	86.6 ± 0.1	**87.2 ± 0.1**
Caltech 256–60	87.2 ± 0.3	85.3 ± 0.2	87.2 ± 0.1	88.2 ± 0.1	88.5 ± 0.1	88.7 ± 0.1	**89.4 ± 0.1**
CUB-200–2011	78.4 ± 0.1	61.5 ± 0.1	79.5 ± 0.1	79.5 ± 0.1	78.2 ± 0.3	80.5 ± 0.1	**82.2 ± 0.1**
Inception-V3	L^2	L^2-*FE*	L^2-*SP*	LwF	Soft FT	DELTA	AFA (Ours)
MIT Indoors 67	74.8 ± 0.4	74.9 ± 0.2	74.6 ± 0.4	76.8 ± 0.3	76.2 ± 0.7	78.1 ± 0.4	**79.3 ± 0.1**
Stanford Dogs 120	88.6 ± 0.2	84.1 ± 0.1	89.4 ± 0.1	88.8 ± 0.2	89.2 ± 0.2	88.7 ± 0.1	**91.6 ± 0.1**
Caltech 256–30	83.6 ± 0.3	82.5 ± 0.2	83.3 ± 0.2	84.1 ± 0.1	84.5 ± 0.1	84.9 ± 0.2	**85.6 ± 0.1**
Caltech 256–60	85.8 ± 0.3	84.1 ± 0.1	85.3 ± 0.1	86.4 ± 0.1	86.5 ± 0.1	86.8 ± 0.1	**87.6 ± 0.1**
CUB-200–2011	74.3 ± 0.2	57.6 ± 0.1	75.2 ± 0.1	76.1 ± 0.2	76.5 ± 0.2	76.5 ± 0.1	**77.1 ± 0.1**
MobileNetV2	L^2	L^2-*FE*	L^2-*SP*	LwF	Soft FT	DELTA	AFA (Ours)
MIT Indoors 67	76.0 ± 0.2	65.0 ± 0.4	75.7 ± 0.5	76.6 ± 0.4	73.3 ± 0.7	77.1 ± 0.3	**78.6 ± 0.1**
Stanford Dogs 120	75.5 ± 0.2	78.3 ± 0.2	81.1 ± 0.1	80.6 ± 0.1	79.9 ± 0.1	81.3 ± 0.1	**82.1 ± 0.1**
Caltech 256–30	78.5 ± 0.2	77.8 ± 0.3	81.0 ± 0.3	80.5 ± 0.2	80.8 ± 0.2	81.2 ± 0.1	**81.9 ± 0.1**
Caltech 256–60	80.8 ± 0.1	80.9 ± 0.1	83.0 ± 0.1	82.7 ± 0.1	82.8 ± 0.1	83.3 ± 0.1	**83.9 ± 0.1**
CUB-200–2011	77.3 ± 0.2	57.5 ± 0.2	76.5 ± 0.4	75.8 ± 0.4	73.4 ± 0.2	77.1 ± 0.3	**77.6 ± 0.1**

Table 4
Effect of attentive modules in terms of Top-1 accuracy (%) of ResNet-101 on five public classification data sets. Here, DFA denotes direct feature alignment without attention for deep transfer.

ResNet-101	w/o Attention (DFA)	AST only	ACT only	AFA (Ours)
MIT Indoors 67	85.3 ± 0.2	85.6 ± 0.2	85.6 ± 0.2	**85.9 ± 0.1**
Stanford Dogs 120	88.3 ± 0.2	90.1 ± 0.1	90.1 ± 0.1	**90.1 ± 0.0**
Caltech 256–30	85.7 ± 0.3	86.9 ± 0.1	87.0 ± 0.1	**87.2 ± 0.1**
Caltech 256–60	87.6 ± 0.2	89.2 ± 0.1	**89.4 ± 0.1**	**89.4 ± 0.1**
CUB-200–2011	78.9 ± 0.1	81.7 ± 0.1	81.7 ± 0.3	**82.2 ± 0.1**

5.3.2. Qualitative evaluation

To further demonstrate the superiority of our AFA, following DELTA, we visualize the activation maps of L^2-*SP*, DELTA and our AFA in Fig. 2. The activation maps of various methods are shown in each column, which are taken from the same channel of a block (*i.e.,* conv5_2) in ResNet-101. In addition, the first row presents the visualization results of ResNet-101 pre-trained on ImageNet.

As shown in Fig. 2 (a) and (c), some activation maps of the source ResNet-101 (pre-trained on ImageNet) are able to approximately cover the target object regions. Our AFA identifies these relevant and informative feature maps between the source and target domains, and then aligns them well. However, L^2-*SP* and DELTA fail to focus on the target object. As shown in Fig. 2 (b) and (d), some activation maps of the source ResNet-101 are unable to cover the target object regions. Nevertheless, our AFA significantly improves these activation maps to focus on the target object. In a word, these visualization results show that our AFA achieves obviously better concentration on the target objects than L^2-*SP* and DELTA, which verifies our motivations and further demonstrates the effectiveness of our AFA.

5.4. Ablation studies

To evaluate our attentive modules, we conduct experiments on ResNet-101 with and without attentive modules. Note that DFA means directly using feature alignment without attention for deep transfer. From the results in Table 4, we have the following observations. First, our AST and ACT achieve much better performance than DFA, which shows the effectiveness of our spatial and channel attentive modules for deep transfer. Second, the performance of our AFA is better than that of AST and ACT. These results demonstrate that our two-stage AFA further improves the performance. In addition, the state-of-the-art DELTA method (See Table 3) outperforms DFA in terms of the Top-1 accuracy by approximately 1.0%. Our AFA surpasses DELTA by approximately 0.5% $\sim$ 2.0%. Thus, compared with the state-of-the-art method DELTA, the improvements of our AFA are significant.

5.5. More discussions

5.5.1. Analysis of our AFA and the DFA paradigm

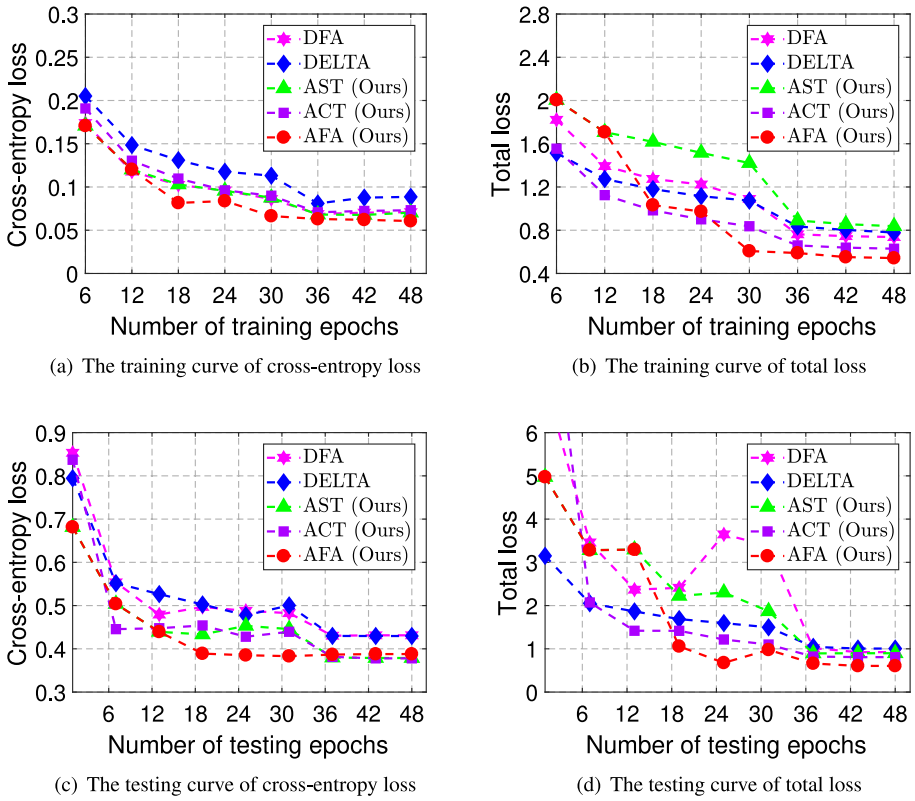
To evaluate our AFA and the DFA paradigms, we show several loss curves in Fig. 3. From Fig. 3, we have the following observations. First, as the training epoch increases, all the methods are guaranteed to converge. Second, our AFA paradigm (*i.e.,* AST, ACT and AFA) has lower testing losses than the DFA paradigm (*i.e.,* DFA and DELTA). Specifically, compared to the DFA paradigm, our AFA paradigm has a smaller cross-entropy loss in testing, which means better generalization performance for the target model (See more results in Tables 3 and 4). Both the theoretical analysis (See Section 4.6) and these results demonstrate the effectiveness of our AFA paradigm.

5.5.2. Differences among DELTA, ACT and AST

In this section, we discuss the differences among DELTA, ACT and AST in detail. DELTA considers the redundancy at the channel level and then adopts feature alignment with pre-learned and fixed channel attention. However, the shared attention across samples is suboptimal for deep transfer. Different from DELTA, our ACT devises a learnable channel attentive module, where each sample has a unique attention weight. As shown in Table 5, our ACT outperforms DELTA on five public data sets. These results demonstrate the effectiveness of our ACT compared with DELTA.

Furthermore, apart from considering the redundancy at the channel level, we also focus on the redundancy at the spatial level. In this way, we devise a learnable spatial attentive module to recognize the true informative spatial regions for feature alignment. From the results in Table 5, our AST achieves comparable performance to our ACT, which demonstrates that identifying the informative spatial regions and channels is equally important for effective deep transfer.

Last, during the training phase, our attentive modules only introduce a small number of parameters. As shown in Table 6, for



(a) The training curve of cross-entropy loss

(b) The training curve of total loss

(c) The testing curve of cross-entropy loss

(d) The testing curve of total loss

Fig. 3. The training and testing curve of the cross-entropy and total loss. Note that we conduct experiments on different methods with ResNet-101 on Stanford Dogs 120. Here, DFA denotes direct feature alignment without attention for deep transfer.

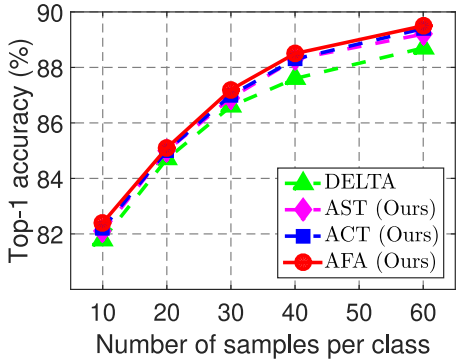


Fig. 4. Comparisons of DELTA and our method in terms of Top-1 accuracy (%) of ResNet-101 on different scales of Caltech 256.

ResNet-101, MLP_c and MLP_s introduce 0.20M and 0.41M parameters respectively, which is negligible for ResNet-101 (44.55M). In practice, for MobileNetV2, the training speed of our AFA (19.9s per epoch) is similar to that of the DELTA method (19.4s per epoch) on CUB-200-2011. Thus, the increased training burden introduced by these additional parameters is acceptable.

5.5.3. Performance on different scales of data set

To evaluate our method and DELTA on different scales of the target data set, we train ResNet-101 on Caltech 256 with different numbers of training samples for each class, *i.e.,* from 10 to 60. The results of Fig. 4 show that our methods (*i.e.,* AST, ACT and AFA) and DELTA achieve higher accuracy with the increasing number of training samples. Moreover, our AFA performs better than the DELTA method. Meanwhile, the performance gap between our AFA and DELTA gradually increases with the increasing number of training samples. These results demonstrate the effectiveness of our AFA on different scales of Caltech 256.

Table 5
Comparisons of DELTA, AST and ACT in terms of Top-1 accuracy (%) on five public data sets.

ResNet-101	DELTA	AST only	ACT only
MIT Indoors	85.5 ± 0.3	**85.6 ± 0.2**	**85.6 ± 0.2**
Stanford Dogs 120	88.7 ± 0.1	**90.1 ± 0.1**	**90.1 ± 0.1**
Caltech 256–30	86.6 ± 0.1	86.9 ± 0.1	**87.0 ± 0.1**
Caltech 256–60	88.7 ± 0.1	89.2 ± 0.1	**89.4 ± 0.1**
CUB-200–2011	80.5 ± 0.1	**81.7 ± 0.1**	81.7 ± 0.3
Inception-V3	**DELTA**	**AST only**	**ACT only**
MIT Indoors 67	78.1 ± 0.4	78.2 ± 0.2	**79.3 ± 0.4**
Stanford Dogs 120	88.7 ± 0.1	**91.5 ± 0.1**	90.3 ± 0.1
Caltech 256–30	84.9 ± 0.2	**85.3 ± 0.1**	85.1 ± 0.1
Caltech 256–60	86.8 ± 0.1	**87.2 ± 0.1**	87.1 ± 0.1
CUB-200–2011	76.5 ± 0.1	**76.8 ± 0.2**	76.7 ± 0.3
MobileNetV2	**DELTA**	**AST only**	**ACT only**
MIT Indoors 67	77.1 ± 0.3	77.8 ± 0.3	**77.9 ± 0.1**
Stanford Dogs 120	81.3 ± 0.1	**81.8 ± 0.1**	81.7 ± 0.1
Caltech 256–30	81.2 ± 0.1	**81.6 ± 0.1**	**81.6 ± 0.1**
Caltech 256–60	83.3 ± 0.1	83.7 ± 0.1	**83.8 ± 0.1**
CUB-200–2011	77.1 ± 0.3	77.0 ± 0.1	**77.3 ± 0.1**

Table 6
Number of parameters of networks and the MLPs. Note that for each network, we devise four MLP_s and four MLP_c for the proposed AST and ACT, respectively.

Network	# Parameters		
	Network	$4 \times MLP_s$	$4 \times MLP_c$
ResNet-101	44.55M	0.41M	0.20M
Inception-V3	27.16M	0.82M	0.41M
MobileNetV2	3.50M	0.06M	0.03M

Table 7
Effect of the one-stage AFA and two-stage AFA in terms of Top-1 accuracy (%) of ResNet-101 on five public classification data sets.

ResNet-101	One-stage AFA ACT+AST	Two-stage AFA AST→ACT
MIT Indoors 67	85.5 ± 0.1	**85.9 ± 0.1**
Stanford Dogs 120	90.0 ± 0.1	**90.1 ± 0.0**
Caltech 256–30	87.0 ± 0.1	**87.2 ± 0.1**
Caltech 256–60	89.4 ± 0.2	**89.4 ± 0.1**
CUB-200–2011	81.7 ± 0.2	**82.2 ± 0.1**

Table 8
Effect of the order of AST and ACT in terms of Top-1 accuracy (%) of ResNet-101 on five public classification data sets.

ResNet-101	Two-stage AFA	
	ACT→AST	AST→ACT
MIT Indoors 67	85.8 ± 0.1	**85.9 ± 0.1**
Stanford Dogs 120	90.1 ± 0.1	**90.1 ± 0.0**
Caltech 256–30	87.0 ± 0.1	**87.2 ± 0.1**
Caltech 256–60	**89.4 ± 0.1**	**89.4 ± 0.1**
CUB-200–2011	**82.2 ± 0.1**	**82.2 ± 0.1**

5.5.4. Effect of the one-stage AFA and two-stage AFA

To evaluate the one-stage AFA and two-stage AFA, we conduct experiments on ResNet-101. Specifically, the one-stage AFA ("ACT+AST") means that we jointly train the target model and the two-level attentive modules by minimizing the objective function in Eq. (4), where the feature alignment loss is the sum of Eqs. (6) and (8). In this way, compared with the two-stage AFA, the one-stage AFA requires to optimize the channel attention and spatial attention parameters simultaneously. Thus, the training complexity of the two-stage AFA is lower than that of the one-stage AFA. From Table 7, the two-stage AFA achieves better performance than the one-stage AFA (*e.g.,* 0.5% improvement on CUB-200-2011, 0.4% improvement on MIT Indoors 67, and 0.2% improvement on Caltech 256-30), which demonstrates the effectiveness of the two-stage AFA. Moreover, ACT (only) and AST (only) achieve comparable performance with the one-stage AFA but perform worse than the two-stage AFA. To some extent, these results demonstrate that the two-level attentions may interfere with each other during the training process. Thus, these results demonstrate the necessity of our two-stage AFA.

To further evaluate the one-stage AFA and two-stage AFA, we show the loss curves of them on CUB-200-2011. As shown in Fig. 5, the two-stage AFA converges to a lower loss value compared with the one-stage AFA in both training and testing phases, which indicates the superiority of the two-stage AFA.

5.5.5. Effect of the order of AST and ACT

To demonstrate the effect of the order of AST and ACT in a two-stage scheme, we conduct experiments on ResNet-101. Specifically, "AST → ACT" scheme denotes that we perform AST in the first stage and then perform ACT in the second stage. As shown in Table 8, "ACT → AST" achieves comparable performance with "AST → ACT", which demonstrates that our AFA is insensitive to the order of AST and ACT. Therefore, in the two-stage AFA, both "AST → ACT" and "ACT → AST" training schemes are appropriate for deep transfer.

Table 10
Comparisons between AFA with and without changing source models in terms of Top-1 accuracy (%) of ResNet-101 on five public classification data sets.

ResNet-101	AFA (w/o change)	AFA (Ours) (with change)
MIT Indoors 67	85.6 ± 0.1	**85.9 ± 0.1**
Stanford Dogs 120	90.0 ± 0.1	**90.1 ± 0.1**
Caltech 256–30	86.5 ± 0.1	**87.2 ± 0.1**
Caltech 256–60	88.6 ± 0.1	**89.4 ± 0.1**
CUB-200–2011	80.8 ± 0.1	**82.2 ± 0.1**

5.5.6. Effect of learnable and fixed attentive modules

Since the informative channels and spatial regions of the feature maps may vary for different samples, the fixed attention across samples may be suboptimal for deep transfer. Thus, we devise learnable attentive modules to find informative channels and spatial regions for each sample. In our AFA, each sample has a unique attention weight. We perform the experiments about the fixed channel and spatial attentions with ResNet-101 on five public data sets. Specifically, we obtain the fixed channel attention following DELTA. We adopt our pre-trained attentive spatial module to forward input samples, and then average the spatial attention weights of these samples to obtain the fixed spatial attention. From Table 9, our AFA performs better than that with fixed channel and spatial attentions (*e.g.,* 0.9% improvement on MIT Indoors 67, 0.7% improvement on Caltech 256-60) and DELTA. These results demonstrate the effectiveness of learnable attentive modules in our AFA compared with these fixed attention methods.

5.5.7. Necessity of changing the source model in Stage II

To demonstrate the necessity of changing the source model in Stage II, we conduct experiments on preserving the original source model in Stage II with ResNet-101 on five public data sets. From Table 10, our AFA with changing the source model performs better than that without changing the source model, especially on Caltech 256-30, Caltech 256-60, CUB-200-2011, and MIT Indoors 67. These results demonstrate the necessity of changing the source model in Stage II.

5.5.8. Effect of the number of attentive modules

To demonstrate the effect of the number of attentive modules, we conduct experiments on ResNet-101 with different numbers of attentive modules, such as 4, 8, 12. From Table 11, our AFA with 4 attentive modules (AMs) achieves the best performance, which demonstrates that introducing too many attentive modules hampers the transfer performance. Moreover, the training time of AFA with 4 AMs (136.5s per epoch) is lower than that with 8 AMs (138.5s per epoch) and 12 AMs (139.4s per epoch). Thus, introducing too many attentive modules increases the training time in each epoch.

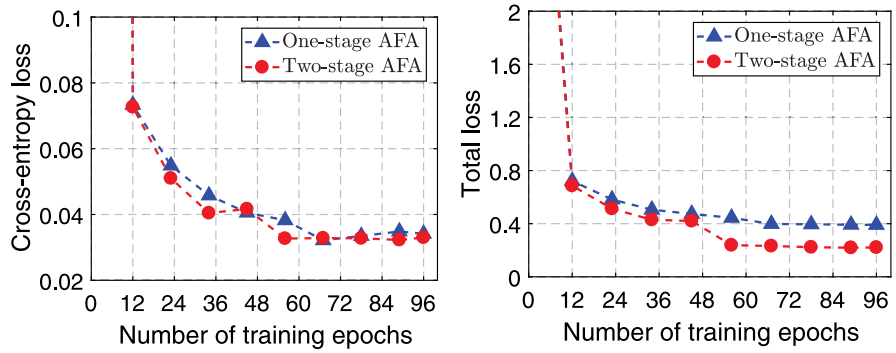
6. Experiments on VIS to NIR face recognition

6.1. Source and target data sets

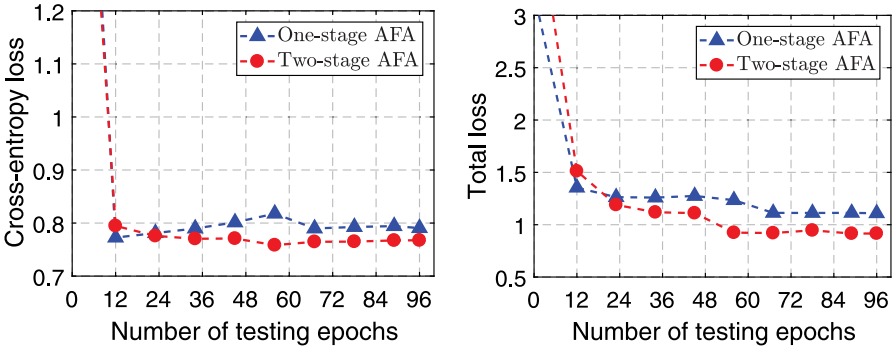
To further demonstrate the effectiveness of our method, we conduct experiments on the face recognition task. The source

Table 9
Comparison between DELTA, AFA and AFA with fixed channel and spatial attentions in terms of Top-1 accuracy (%) of ResNet-101 on five public classification data sets.

ResNet-101	DELTA (Fixed channel attentions)	AFA (Fixed all attentions)	AFA (Ours)
MIT Indoors 67	85.5 ± 0.3	85.0 ± 0.1	**85.9 ± 0.1**
Stanford Dogs 120	88.7 ± 0.1	89.8 ± 0.1	**90.1 ± 0.0**
Caltech 256–30	86.6 ± 0.1	86.9 ± 0.1	**87.2 ± 0.1**
Caltech 256–60	88.7 ± 0.1	88.7 ± 0.1	**89.4 ± 0.1**
CUB-200–2011	80.5 ± 0.1	82.1 ± 0.1	**82.2 ± 0.1**



(a) The training curve of cross-entropy loss (b) The training curve of total loss



(c) The testing curve of cross-entropy loss (d) The testing curve of total loss

Fig. 5. The training and testing curves of the cross-entropy and total loss. Here, we conduct experiments on the one-stage AFA and two-stage AFA with ResNet-101 and report the results on CUB-200-2011.

Table 11
Effect of the number of attentive modules in terms of Top-1 accuracy (%) and training time (seconds) of ResNet-101 on Stanford Dogs 120. Here, #AMs indicates the number of attentive modules.

ResNet-101	AFA		
	4 #AMs	8 #AMs	12 #AMs
Top-1 accuracy (%)	**90.1 ± 0.0**	89.9 ± 0.1	89.8 ± 0.0
Training time (s)/epoch	**136.5**	138.5	139.4

Table 12
Performance comparisons of different methods on PolyU-NIRFD. "TAR" refers to True Accepted Rate and "FAR = 1e−5" refers to the False Accepted Rate at 1e−5.

MobileFaceNet	Verification TAR (%)		
	@FAR = 1e−5	@FAR = 1e−6	@FAR = 1e−7
L^2	94.4	89.1	87.6
L^2-*FE*	93.6	91.2	87.8
L^2-*SP*	96.4	94.9	94.4
LwF	96.9	95.8	95.0
Soft FT	97.4	96.3	94.8
DELTA	97.3	96.1	94.9
AFA (Ours)	**98.1**	**97.2**	**96.7**
LResNet34E-IR	Verification TAR (%)		
	@FAR = 1e−5	@FAR = 1e−6	@FAR = 1e−7
L^2	93.5	92.3	91.6
L^2-*FE*	97.4	96.4	95.5
L^2-*SP*	96.6	95.7	95.2
LwF	97.1	95.2	94.5
Soft FT	96.8	95.5	94.7
DELTA	97.6	96.8	96.2
AFA (Ours)	**98.6**	**97.9**	**97.5**

data set is the visible light (VIS) face data set and the target data set is the near infrared ray (NIR) face data set. We use the refined MS-Celeb-1M (Guo, Zhang, Hu, He, & Gao, 2016) released by Deng, Guo, Xue, and Zafeiriou (2019) as the source data set and the PolyU NIR face database (PolyU-NIRFD) (Zhang, Zhang, Zhang, & Shen, 2010) as the target data set. PolyU-NIRFD contains 38,430 NIR face images from 335 identities. We randomly select 80 identities of PolyU-NIRFD as the validation set and the others as the training data. The evaluation metric is the true accept rate (TAR) given the false accept rate (FAR).

6.2. Implementation details

We adopt MobileFaceNet (Chen, Liu, Gao, & Han, 2018a) and LResNet34E-IR (Deng, Guo, Xue, & Zafeiriou, 2019) as base networks. Following the settings in Deng, Guo, Xue, and Zafeiriou (2019), we use the two base models pre-trained on the refined MS-Celeb-1M as the source models. Moreover, we adopt L^2, L^2-*FE*, L^2-*SP* (Li et al., 2018), LwF (Li & Hoiem, 2017), Soft Fine-tuning (Soft FT) (Zhao et al., 2019) and DELTA (Li et al., 2019) as compared methods. The initial learning rate is set to 0.001 for LwF and 0.0001 for Soft Fine-tuning.

6.3. Comparisons with state-of-the-art methods

From the results of Table 12, we have the following observations. First, for the lightweight network (*i.e.,* MobileFaceNet), the four compared methods (L^2-*SP*, LwF, Soft Fine-tuning and DELTA) surpass the naïve L^2 and L^2-*FE* baselines on the TAR by more than 2.0%. However, three compared methods (L^2-*SP*, LwF, Soft Fine-tuning) perform worse than the naïve L^2-*FE* baseline on the TAR for the large network (*i.e.,* LResNet34E-IR). These results demonstrate that L^2-*SP*, LwF and Soft Fine-tuning methods are

sensitive to the networks in the face recognition task. Second, our AFA outperforms the four compared methods on the TAR by at least 1.0% for the two networks. These results demonstrate the effectiveness of AFA in the face recognition task. In addition, these results indicate that AFA is insensitive to the networks in contrast to the other state-of-the-art methods.

7. Conclusion

In this paper, we have proposed a two-stage deep transfer method, named Attentive Feature Alignment (AFA), for effective domain knowledge transfer by identifying and attending on those relevant channels and spatial features between two domains. To this end, we devise two learnable attentive modules to recognize the relevant features at both the channel and spatial levels. Specifically, in the first stage of AFA, we adopt Attentive Spatial Transfer (AST) to perform spatial-level feature alignment. In the second stage, we take the target model trained in the first stage as the source model and then employ Attentive Channel Transfer (ACT) to perform channel-level feature alignment for better knowledge transfer. In this way, our AFA is able to well match the source and target domains and ensures better deep transfer. More critically, we theoretically analyze the generalization performance of AFA, which confirms its superiority to existing methods. Extensive experiments on both image classification and face recognition demonstrate the effectiveness of AFA compared with several state-of-the-art methods. In the future, we will extend our proposed method to other tasks, such as scene segmentation and fine-grained image recognition.

Declaration of competing interest

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

Acknowledgments

This work was partially supported by Key-Area Research and Development Program of Guangdong Province, China (2018B010107001, 2019B010155002, 2019B010155001), National Natural Science Foundation of China (NSFC) 61836003 (key project), Program for Guangdong Introducing Innovative and Entrepreneurial Teams, China (2017ZT07X183), Tencent AI Lab Rhino-Bird Focused Research Program, China (No. JR201902), Fundamental Research Funds for the Central Universities, China D2191240.

Appendix. Proof of Theorem 1

Proof. We extend Mohri et al. (2012) to a case of the expected loss:

$$E(f,h) = \mathbb{E}[\mathcal{L}(f(\mathbf{x}),\mathbf{y}) + \alpha\Omega(\mathbf{W}_t,\mathbf{W}_s,\mathbf{W}_a,h,\mathbf{x})]\,. \tag{A.1}$$

For any sample $\mathcal{S} = \{\mathbf{z}_1,\ldots,\mathbf{z}_N\}$ and any $(f,h) \in \Phi$, let $\hat{E}_{\mathcal{S}}(f,h)$ be the empirical loss of f and h over $\mathcal{S}$. Applying McDiarmid's inequality (Hoeffding, 1994) to function Ψ defined for any sample $\mathcal{S}$ by

$$\Psi(\mathcal{S}) = \sup_{(f,h)\in\Phi} E(f,h) - \hat{E}_{\mathcal{S}}(f,h)\,. \tag{A.2}$$

Based on McDiarmid's inequality (Hoeffding, 1994) and Theorem 3.1 in Mohri et al. (2012), for any $\delta > 0$, with probability at least $1-\delta$, we have

$$\begin{aligned}\Psi(\mathcal{S}) &\le E_{\mathcal{S}}[\Psi(\mathcal{S})] + U\sqrt{\frac{1}{2N}\log\left(\frac{1}{\delta}\right)}\\ &\le 2\mathbf{R}_N^{AFA}(\Phi) + U\sqrt{\frac{1}{2N}\log\left(\frac{1}{\delta}\right)}\,.\end{aligned} \tag{A.3}$$

Last, by combining Eq. (A.2) and Inequalities (A.3), we have

$$E(f,h) \le \hat{E}(f,h) + 2\mathbf{R}_N^{AFA}(\Phi) + U\sqrt{\frac{1}{2N}\log\left(\frac{1}{\delta}\right)}\,, \tag{A.4}$$

with probability at least $1-\delta$.

Based on Rademacher complexity, we have

$$\mathbf{R}_N^{AFA} \le \mathbf{R}_N^{DFA}\,, \tag{A.5}$$

since the capacity of the function space $\Phi \in \mathcal{F}\times\mathcal{H}$ is smaller than the capacity of the function space $\Phi \in \mathcal{F}$. With the same number of training samples, we thus have

$$\mathcal{B}(f,h) \le \mathcal{B}(f)\,. \tag{A.6}$$

References

Aygun, M., Aytar, Y., & Kemal Ekenel, H. (2017). Exploiting convolution filter patterns for transfer learning. In *ICCV* (pp. 2674–2680).
Ba, J., & Caruana, R. (2014). Do deep nets really need to be deep? In *NeurIPS* (pp. 2654–2662).
Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR*.
Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*(1), 41–75.
Chen, S., Liu, Y., Gao, X., & Han, Z. (2018). MobileFaceNets: Efficient cnns for accurate real-time face verification on mobile devices. In *CCBR* (pp. 428–438).
Chen, Y., Wang, N., & Zhang, Z. (2018). DarkRank: Accelerating deep metric learning via cross sample similarities transfer. In *AAAI* (pp. 2852–2859).
Chen, L., Yang, Y., Wang, J., Xu, W., & Yuille, A. L. (2016). Attention to scale: Scale-Aware semantic image segmentation. In *CVPR* (pp. 3640–3649).
Cui, Y., Song, Y., Sun, C., Howard, A., & Belongie, S. (2018). Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR* (pp. 4109–4118).
Deng, W., Dong, Y., Liu, G., Wang, Y., & Men, J. (2019). Multiclass heterogeneous domain adaptation via bidirectional ECOC projection. *Neural Networks*, *119*, 313–322.
Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F. (2009). ImageNet: A large-scale hierarchical image database. In *CVPR* (pp. 248–255).
Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). ArcFace: Additive angular margin loss for deep face recognition. In *CVPR* (pp. 4690–4699).
Dorado-Moreno, M., Navarin, N., Gutiérrez, P. A., Prieto, L., Sperduti, A., Salcedo-Sanz, S., et al. (2020). Multi-task learning for the prediction of wind power ramp events with deep neural networks. *Neural Networks*, *123*, 401–411.
Fernando, T., Denman, S., Sridharan, S., & Fookes, C. (2018). *Soft + hardwired* attention: An LSTM framework for human trajectory prediction and abnormal event detection. *Neural Networks*, *108*, 466–478.
Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., et al. (2019). Dual attention network for scene segmentation. In *CVPR* (pp. 3146–3154).
Ge, W., & Yu, Y. (2017). Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *CVPR* (pp. 1086–1095).
Gligic, L., Kormilitzin, A., Goldberg, P., & Nevado-Holgado, A. J. (2020). Named entity recognition in electronic health records using transfer learning bootstrapped neural networks. *Neural Networks*, *121*, 132–139.
Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 object category dataset*. California Institute of Technology.
Guo, Y., Zhang, L., Hu, Y., He, X., & Gao, J. (2016). MS-Celeb-1M: A dataset and benchmark for large-scale face recognition. In *ECCV* (pp. 87–102).
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR* (pp. 770–778).
Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.
Hoeffding, W. (1994). *Probability inequalities for sums of bounded random variables* (pp. 409–426).
Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *CVPR* (pp. 7132–7141).
Huang, X., Peng, Y., & Yuan, M. (2017). Cross-modal common representation learning by hybrid transfer network. In *IJCAI* (pp. 1893–1900).

Khosla, A., Jayadevaprakash, N., Yao, B., & Li, F. -F. (2011). Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPRW*.
Kirkpatrick, J., Pascanu, R., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, *114*(13), 3521–3526.
Li, X., Grandvalet, Y., & Davoine, F. (2018). Explicit inductive bias for transfer learning with convolutional networks. In *ICML* (pp. 2830–2839).
Li, Z., & Hoiem, D. (2017). Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(12), 2935–2947.
Li, X., Xiong, H., et al. (2019). DELTA: Deep learning transfer using feature map with attention for convolutional networks. In *ICLR*.
Liu, J., et al. (2017). Sparse deep transfer learning for convolutional neural network. In *AAAI* (pp. 2245–2251).
Luong, T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *EMNLP* (pp. 1412–1421).
Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. MIT Press.
Nahmias, D., Cohen, A., Nissim, N., & Elovici, Y. (2020). Deep feature transfer learning for trusted and automated malware signature generation in private cloud environments. *Neural Networks*, *124*, 243–257.
Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.
Quattoni, A., & Torralba, A. (2009). Recognizing indoor scenes. In *CVPR* (pp. 413–420).
Raghu, S., Sriraam, N., Temel, Y., Rao, S. V., & Kubben, P. L. (2020). EEG based multi-class seizure type classification using convolutional neural network and transfer learning. *Neural Networks*, *124*, 202–212.
Romero, A., Ballas, N., et al. (2015). FitNets: Hints for thin deep nets. In *ICLR*.
Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI* (pp. 234–241).
Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting visual category models to new domains. In *ECCV* (pp. 213–226).
Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. -C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR* (pp. 4510–4520).
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *CVPR* (pp. 2818–2826).
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *NeurIPS* (pp. 5998–6008).
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *NeurIPS* (pp. 6000–6010).
Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). *The caltech-ucsd birds-200-2011 dataset*. California Institute of Technology.
Wang, X., Girshick, R. B., Gupta, A., & He, K. (2018). Non-Local neural networks. In *CVPR* (pp. 7794–7803).
Woo, S., Park, J., Lee, J., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. In *ECCV* (pp. 3–19).
Xie, Z., Wen, Z., Liu, J., Liu, Z., Wu, X., & Tan, M. (2020). Deep transferring quantization. In *ECCV* (pp. 625–642).
Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *ICML* (pp. 2048–2057).
Yang, S., Lu, H., Kang, S., Xue, L., Xiao, J., Su, D., et al. (2020). On the localness modeling for the self-attention based end-to-end speech synthesis. *Neural Networks*, *125*, 121–130.
Yang, L., & Zhong, P. (2020). Robust adaptation regularization based on within-class scatter for domain adaptation. *Neural Networks*, *124*, 60–74.
Yim, J., Joo, D., Bae, J., & Kim, J. (2017). A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR* (pp. 4133–4141).
Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *NeurIPS* (pp. 3320–3328).
Zagoruyko, S., & Komodakis, N. (2017). Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*.
Zhang, Y., Chen, H., Wei, Y., Zhao, P., Cao, J., Fan, X., et al. (2019). From whole slide imaging to microscopy: Deep microscopy adaptation network for histopathology cancer image classification. In *MICCAI* (pp. 360–368).
Zhang, C., & Peng, Y. (2018). Better and faster: Knowledge transfer from multiple self-supervised learning tasks via graph distillation for video classification. In *IJCAI* (pp. 1135–1141).
Zhang, B., Zhang, L., Zhang, D., & Shen, L. (2010). Directional binary code with application to polyu near-infrared face database. *Pattern Recognition Letters*.
Zhao, Z., Zhang, B., Jiang, Y., Xu, L., Li, L., & Ma, W. -Y. (2019). Effective domain knowledge transfer with soft fine-tuning. arXiv preprint arXiv:1909.02236.
Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(6), 1452–1464.
Zhu, Y., Li, R., Yang, Y., & Ye, N. (2020). Learning cascade attention for fine-grained image classification. *Neural Networks*, *122*, 174–182.
Zhu, Y., Zhuang, F., Wang, J., Chen, J., Shi, Z., Wu, W., et al. (2019). Multi-representation adaptation network for cross-domain image classification. *Neural Networks*, *119*, 214–221.
Zhuang, Z., Tan, M., Zhuang, B., Liu, J., Guo, Y., Wu, Q., et al. (2018). Discrimination-aware channel pruning for deep neural networks. In *NeurIPS* (pp. 875–886).