Knowledge Distillation with Attention for Deep Transfer Learning of Convolutional Networks

XINGJIAN LI, Baidu, Inc., China and University of Macau
HAOYI XIONG and ZEYU CHEN, Baidu, Inc.
JUN HUAN, StylingAI Inc.
JI LIU, Baidu, Inc.
CHENG-ZHONG XU, University of Macau
DEJING DOU, Baidu, Inc.

Transfer learning through fine-tuning a pre-trained neural network with an extremely large dataset, such as ImageNet, can significantly improve and accelerate training while the accuracy is frequently bottlenecked by the limited dataset size of the new target task. To solve the problem, some regularization methods, constraining the outer layer weights of the target network using the starting point as references (SPAR), have been studied. In this article, we propose a novel regularized transfer learning framework DELTA, namely *DEep Learning Transfer using Feature Map with Attention*. Instead of constraining the weights of neural network, DELTA aims at preserving the outer layer outputs of the source network. Specifically, in addition to minimizing the empirical loss, DELTA aligns the outer layer outputs of two networks, through constraining a subset of feature maps that are precisely selected by attention that has been learned in a supervised learning manner. We evaluate DELTA with the state-of-the-art algorithms, including L^2 and L^2-*SP*. The experiment results show that our method outperforms these baselines with higher accuracy for new tasks. Code has been made publicly available.[1]

CCS Concepts: • **Computing methodologies** → **Neural networks**; **Transfer learning**; *Supervised learning by classification;*

Additional Key Words and Phrases: Transfer learning, framework, algorithms, knowledge distillation

[1]Source codes of the proposed algorithms and experiments are available online at https://github.com/lixingjian/DELTA. An early version of this manuscript has been published as [1].

X. Li and H. Xiong contributed equally to this work.

This paper is supported by National Key Research and Development Program of China (No. 2019YFB2102100), the Science and Technology Development Fund of Macau SAR (File no. 0015/2019/AKP), GuangDong Basic and Applied Basic Research Foundation No. 2020B515130004, and Key-Area Research and Development Program of Guangdong Province (No. 2020B010164003).

Authors' addresses: X. Li, Baidu, Inc., Baidu Technology Park, Haidian, Beijing, China, University of Macau, Tapia, Macau, China; email: lixingjian@baidu.com; H. Xiong (corresponding author), Z. Chen, J. Liu, and D. Dou are with Baidu, Inc., Baidu Technology Park, Haidian, Beijing, China; emails: haoyi.xiong.fr@ieee.org, {chenzeyu01, liuji04, doudejing}@baidu.com; J. Huan, Styling AI Inc., Haidian, Beijing, China; email: lukehuan@shenshangtech.com; C.-Z. Xu, University of Macau, Tapia, Macau, China; email: czxu@um.edu.mo.

1556-4681/2021/10-ART42 $15.00
https://doi.org/10.1145/3473912

ACM Reference format:
Xingjian Li, Haoyi Xiong, Zeyu Chen, Jun Huan, Ji Liu, Cheng-Zhong Xu, and Dejing Dou. 2021. Knowledge Distillation with Attention for Deep Transfer Learning of Convolutional Networks. *ACM Trans. Knowl. Discov. Data.* 16, 3, Article 42 (October 2021), 20 pages.
https://doi.org/10.1145/3473912

1 INTRODUCTION

Deep neural networks (**DNNs**), especially **deep convolutional neural networks** (**deep CNNs**) show up enormous advantages in tasks of various modalities including images [2–4], audios [5, 6], and videos [7, 8]. While in many real-world applications, deep learning practitioners often have limited number of training instances. Training a DNN with a small training data set and random weights usually results in the so-called *over-fitting* problem and the quality of the obtained model is low. A simple yet effective approach to obtain high-quality deep learning models is to perform weight fine-tuning [9].

1.1 Summary of Existing Works

To enable fine-tuning, a DNN is first trained using a large (and possibly irrelevant) source dataset (e.g., ImageNet). Then, the pre-trained weights are further fine-tuned using the data from the target application domain. Due to the simplicity and effectiveness, the fine-tuning strategy is widely applied in a large variety of tasks such as image/video classification [10, 11], visual tracking [12], action recognition [13], human head pose estimation [14], and so on. Intuitively, the weights pre-trained by the source dataset with a sufficiently large number of instances usually provide a better initialization for the target task than random initialization [15]. Fine-tuning with pre-trained weights could largely improve the performance of deep learning, as part of DNN weights would be reused [16]. More specifically, strategies proposed for fine-tuning could be categorized in three folders as follows.

- *Fine-tuning with Weight Decay (L^2) [9].* A popular way for fine-tuning is to use the weight decay as a L^2-norm regularizer and the pre-trained model as the initialization for optimization. Though this method is simple and efficient, it however suffers from the phenomenon of catastrophic forgetting [17]. Specifically, weights of the target model may be driven far away from initial values and converge to some point on a L^2-ball [18], which leads to losses of source knowledge and causes over-fitting in transfer learning scenarios.
- *Fine-tuning with Start Point as Reference (L^2-SP) [19].* In addition to use L^2-norm regularization around the origin point, Li et al. [19] proposed L^2-*SP* that incorporates the Euclid distance between the target weights and the starting point (i.e., weights of source network) as the regularizer. L^2-*SP* aims at minimizing the empirical loss of deep learning while reducing the distance of weights between source and target networks. They achieved significant improvement compared with standard practice of using the weight decay.
- *Knowledge Distillation on Feature Maps [20].* In addition to learning from pre-trained weights, yet another way is to learn from pre-trained features. Thus, rather than constraining the differences between weights of source and target models, Yim et al. [20] proposed to use quadratic loss between feature maps, so as to bound the divergence between the feature maps generated by source and target models. With constrained feature maps, the generalization capacity of the target model could be improved through aligning the "behaviors" paid by the

(a) Source Model (b) Fine-tuning (L^2) (c) L^2-*SP* (d) Know. Distill.

Fig. 1. Visual attentions to images—activation maps extracted from ResNet-50 models pre-trained with ImageNet and fine-tuned to adapt Stanford Dog 102 datasets using various transfer learning methods. Know. Distill.: Knowledge distillation from the feature maps of pre-trained models.

outer CNN layers on feature maps[2] of the target network to the source one, which has been pre-trained using an extremely large dataset.

In addition to above methods, Aygun et al. [21] proposed to model the weights of source network using a probabilistic distribution. With such distribution as prior, a Bayesian transfer learning mechanism was implemented to fit the weights of target network in a similar distribution.

1.2 Technical Challenges and Research Issues

Existing regularization methods however may be able to deliver the best performance for transfer learning in many cases. On one side, if the regularization is not strong, even with fine-turning, the weights may still be driven far away from the initial position, leading to the lose of useful knowledge, i.e., catastrophic memory loss. On the other side, if the regularization is too strong, newly obtained model is constrained to a local neighborhood of the original model, which may be suboptimal to the target dataset. Although aforementioned methods demonstrated the power of regularization in deep transfer learning, we argue that we need to perform research on at least the following two aspects in order to further improve current regularization methods.

- ***Attention vs. Discrimination.*** We argue that the pre-trained features are not discriminative in the target domain, as they "pay attentions" [22] to inappropriate visual concepts in the image for classification. Figure 1 demonstrates an example of images with visual attentions/activation maps extracted from various DNN models (Please refer to Section 4.4 for detail settings). All these models are pre-trained using ImageNet [23] and fine-tuned to adapt Stanford Dog 102 datasets [24]. It is obvious that the source model pre-trained with ImageNet would activate at inappropriate parts of images. Instead of activating on the dog, filters in the source model as well as the fine-tuned models all significantly activate on pixels of flowers and grass, as flowers and grass are two important visual concepts in ImageNet datasets. Thus, there needs to pay attention to the discriminative parts of images.
- ***Adaptation vs Generalization.*** The goal of fine-tuning is to update pre-trained weights to adapt the target dataset. On the other hand, as the pre-trained model is based on an extremely large dataset from source domain, it could provide good generalization performance through reusing the pre-trained weights [9]. To achieve the best performance, there might need to optimally select a subset of weights from the pre-trained model and reuse the

[2]In CNNs, which we focus on exclusively in this article, an *outer CNN layer* is a convolution layer and *outputs* of an outer layer are feature maps.

selected weights as the initialization of fine-tuning [16]. Such subset selection is a combinatoric problem subject to the target datasets, while fine-tuning and validation procedures are required to evaluate every possible selection. Thus, there needs a low-complexity method to surrogate the solution for subset selection of pre-trained weights for optimal fine-tuning.

Existing methods [9, 19–21] cannot tackle above two technical challenges simultaneously. All these methods intend to learn from pre-trained weights/features without fine-grained weights/features selection subject to the source and target domains.

1.3 Our Contributions

In this article, we propose a novel regularization approach **DEep Learning Transfer using Feature Maps with Attention** (DELTA) to address above two technical issues. In general, the contributions made in this manuscript could be summarized as follows:

- In this work, we study the problem of fine-tuning pre-trained models to adapt target domains. Specifically, we focus on the technical challenges on (1) preserving discriminative features from the pre-trained models, while (2) adapting the target dataset in a generalizable manner. In summary, our key insight is what we call "***Attentional Knowledge Distillation***". Specifically our approach identifies those "transferable channels", which could extract discriminative features from the target datasets, and preserves such filters through knowledge distillation. On the other hand, the proposed mechanism also identifies those "untransferable channels" and reuses them for fine-tuning. In this way, the fine-tuning procedure could pay attentions to those discriminative features among all pre-trained ones.
- In this work, we propose DELTAalgorithms. Specifically, DELTA selects the discriminative features from outer layer outputs, and learns from these pre-trained features through weighted knowledge distillation on feature maps. Specifically, DELTA re-weights the L^2-norm error terms of feature-wise knowledge distillation with a novel supervised attention mechanism. Through paying attention to discriminative parts of feature maps, DELTA characterizes the distance between source/target networks using their outer layer outputs, and incorporates such distance as the regularization term of the loss function. With the back-propagation, such regularization finally affects the optimization for weights of DNN and awards the target network generalization capacity inherited from the source network.
- We have conducted extensive experiments using a wide range of source/target datasets and compared DELTA to the existing deep transfer learning algorithms that are in pursuit of weight similarity. The experiment results show that DELTA significantly outperformed the state-of-the-art regularization algorithms including L^2, L^2-SP, and feature-wise knowledge distllation with higher accuracy on a wide group of image classification data sets.

Organization of the Article. The rest of the article is organized as follows. In Section 2, backgrounds and preliminaries on fine-tuning pre-trained models are summarized. In Section 3, the proposed DELTA algorithms are introduced. In Section 4, experimental results are presented and discussed. In Section 5, we present the related work and discuss our contributions. Finally in Section 6 the article is concluded.

2 BACKGROUNDS

In this section, we first review the technical backgrounds of the proposed research.

2.1 General Regularization

To achieve good performance, deep CNNs usually consist of a great number of parameters that can describe an amazingly wide range of phenomena and can fit any amount of data available.

For example, ResNet-110 has more than 1 million free parameters. These models are over-parameterized for their tasks and causes over-fitting easily. Regularization is the technique to reduce this risk by constraining the parameters within a limited space. The general regularization problem is usually formulated as follow.

Let's denote the dataset for the desired task as $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), \ldots, (\mathbf{x}_n, y_n)\}$, where totally n tuples are offered and each tuple $(\mathbf{x}_i, y_i)$ refers to an input image and its label in the dataset. We further denote $\omega \in \mathbb{R}^d$ be the d-dimensional parameter vector containing all d parameters of the target model. The optimization object with regularization is to obtain

$$\min_{w} \sum_{i=1}^{n} L(z(\mathbf{x}_i, \omega), y_i) + \lambda \cdot \Omega(\omega), \tag{1}$$

where the first term $\sum_{i=1}^{n} L(z(\mathbf{x}_i, \omega), y_i)$ refers to the empirical loss of data fitting while the second term is a general form of regularization. The tuning parameter $\lambda > 0$ balances the trade-off between the empirical loss and the regularization loss. Larger values of λ correspond to more regularization. Without any explicit information (such as other datasets) given, one can easily use the $L^0/L^1/L^2$-norm of the parameter vector ω as the regularization to penalize the weights while they are updated.

2.2 Regularization for Transfer Learning

Given a pre-trained network with parameter ω^* based on an extremely large dataset as the source, one can estimate the parameter of target network through the transfer learning paradigms. Using the ω^* as the initialization to solve the problem in Equation (1) can accelerate the training of target network through knowledge transfer [25, 26]. However, the accuracy of the target network would be bottlenecked in such settings. To further improve the transfer learning, novel regularized transfer learning paradigms that constrain the divergence between target and source networks has been proposed, such that

$$\min_{w} \sum_{i=1}^{n} L(z(\mathbf{x}_i, \omega), y_i) + \lambda \cdot \Omega(\omega, \omega^*), \tag{2}$$

where the regularization term $\Omega(\omega, \omega^*)$ characterizes the regularization effects. In this way, we summarize the existing deep transfer learning approaches as the solution of the regularized learning problem listed in Equation (2), where the regularizer aims at constraining the divergence of parameters of the two networks while ignoring the behavior of the networks with the training dataset $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$. More specifically, the regularization terms used by the existing deep transfer learning approaches neither consider how the network with certain parameters would behave with the new data (images) nor leverage the supervision information from the labeled data (images) to improve the transfer performance.

As was mentioned, three common regularization-based deep transfer learning algorithms studied in this article are fine-tuning with weight decay [9], L^2-SP [19] and knowledge distillation-based regularization [20]. Specifically, these three algorithms can be implemented with the general regularization-based deep transfer learning procedure with objective function listed in Equation (2) using following three regularizers:

— Fine-tuning with Weight Decay Regularization [9]—In terms of regularizer, this algorithm uses the squared–euclidean distance between the target weights (i.e., optimization objective ω) and the origin point (listed in Equation (3)) to constrain the learning procedure.

$$\Omega(\omega, \omega^*) = \|\omega\|_2^2. \tag{3}$$

In terms of optimization procedure, fine-tuning with weight decay uses ω^* to initialize the learning procedure.

— Fine-tuning with L^2-SP Regularization [19]—In terms of regularizer, this algorithm uses the squared–euclidean distance between the target weights (i.e., optimization objective ω) and the pre-trained weights ω_s of source network (listed in Equation (4)) to constrain the learning procedure.

$$\Omega(\omega, \omega^*) = \|\omega - \omega^*\|_2^2. \tag{4}$$

In terms of optimization procedure, L^2-SP makes the learning procedure start from the pre-trained weights ω^*.

— Fine-tuning with Knowledge Distillation-based Regularization [20]—Given the target dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ and N filters in the target/source networks for knowledge transfer, this algorithm models the regularization as the aggregation of squared–euclidean distances between feature maps outputted by the N filters of the source/target networks, such that

$$\Omega(\omega, \omega^*) = \frac{1}{n} \sum_{j=1}^{N} \sum_{i=1}^{n} \|\mathrm{FM}_j(\omega, \mathbf{x}_i) - \mathrm{FM}_j(\omega^*, \mathbf{x}_i)\|_2^2, \tag{5}$$

where $\boldsymbol{F}_j(\omega, \mathbf{x}_i))$ refers to the feature map outputted by the jth filter ($1 \le j \le N$) of the target network based on weight ω using input image $\mathbf{x}_i$ ($1 \le i \le n$). The optimization algorithm starts from ω^s as the initialization of learning.

In the rest of this work, we presented a strategy DELTA to improve the general form of knowledge distillation-based deep transfer learning shown in Equation (5), then evaluated and compared DELTA using above two regularizers with common deep transfer learning benchmarks.

3 LEARNING FRAMEWORK AND ALGORITHMS

In this section, we detail the regularization term for fine-tuning. Different with learning from scratch, regularization in transfer learning aims at making the best use of the knowledge learned on the source tasks and avoid over-fitting when training on the target dataset. We will first present a general form of regularizations for fine-tuning. Next, we describe our feature map- based regularization with a learnable attention component in detail.

3.1 Overall Framework

In our research, instead of bounding the difference of weights, we intend to regulate the network behaviors and force some layers of the target network to behave similarly to the source ones. Specifically, we define the "behaviors" of a layer as its output, which are with semantics-rich and discriminative information.

DELTA intends to incorporate a new regularizer $\Omega'(\omega, \omega^*, \mathbf{x})$. Given a pre-trained parameter ω^* and any input image $\mathbf{x}$, the regularizer $\Omega'(\omega, \omega^*, \mathbf{x})$ measures the distance between the behaviors of target network with parameter ω and the source one based on ω^*. With such regularizer, the transfer learning problem can be reduced to learning problem as follows:

$$\min_{w} \sum_{i=1}^{n} L(z(\mathbf{x}_i, \omega), y_i) + \sum_{i=1}^{n} \Omega(\omega, \omega^*, \mathbf{x}_i, y_i, z), \tag{6}$$

where $\sum_{i=1}^{n} \Omega(\omega, \omega^*, \mathbf{x}_i, y_i, z)$ characterizes the aggregated difference between the source and target network over the whole training dataset using the model z. Note that, with the input tuples $(\mathbf{x}_i, y_i)$ and for $1 \le i \le n$, the proposed regularizer $\Omega(\omega, \omega^*, \mathbf{x}_i, y_i, z)$ is capable of regularizing the behavioral differences of network model z based on each labeled sample $(\mathbf{x}_i, y_i)$ in the dataset, using the parameters ω and ω^*, respectively.

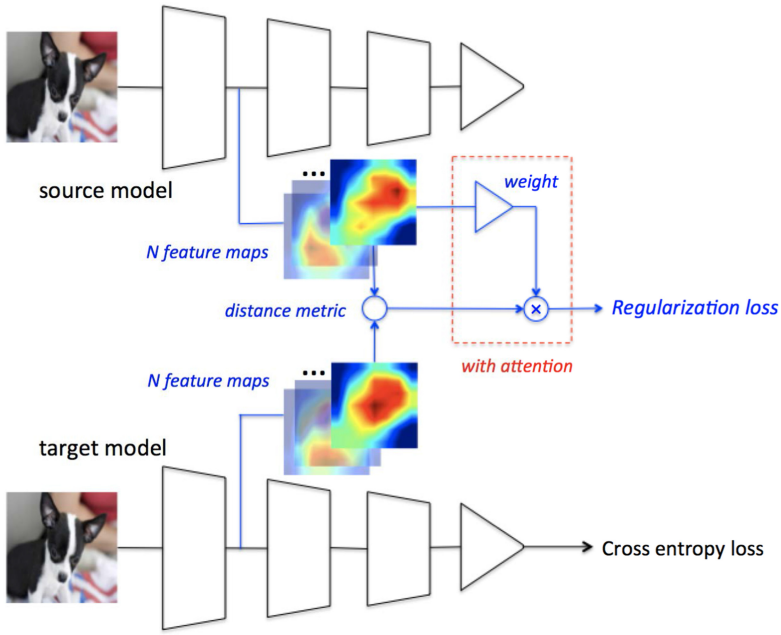

Fig. 2. Behavior-based Regularization using feature maps with attentions.

Furthermore, inspired by the **starting point as reference** (**SPAR**) method, DELTA accelerates the optimization procedure of the regularizer through incorporating a parameter-based proximal term, such that

$$\Omega(\omega, \omega^*, \mathbf{x}, y, z) = \alpha \cdot \Omega'(\omega, \omega^*, \mathbf{x}, y, z) + \beta \cdot \Omega''(\omega \backslash \omega^*), \tag{7}$$

where α, β are two non-negative tuning parameters to balance two terms. On top of the *behavioral regularizer* $\Omega'(\omega, \omega^*, \mathbf{x}, y, z)$, DELTA includes a term $\Omega''(\omega \backslash \omega^*)$ regularizing a subset of parameters that are privately owned by the target network ω only but not exist in the source network ω^*. Specifically, $\Omega''(\omega \backslash \omega^*)$ constrains the L^2-norm of the private parameters in ω, so as to improve the consistency of inner layer parameters estimation. Note that, when using ω^* as the initialization of ω for optimization, DELTA indeed adopts SPAR strategy [19] to accelerate the optimization and gains better generalizability.

3.2 Attentional Regularization

To regularize the behavior of the networks, DELTA considers the distance between the outer layer outputs of the two networks. Figure 2 illustrates the concepts of proposed method. Specifically, the outer layer of the network consists of a large set of convolutional filters. Given an input $\mathbf{x}_i$ (for $\forall 1 \leq i \leq n$ in training set), each filter generates a feature map. Thus, DELTA characterizes the outer layer output of the network model z based on input $\mathbf{x}_i$ and parameter ω using a set of feature maps, such as $\mathrm{FM}_j(z, \omega, \mathbf{x}_i)$ and $1 \leq j \leq N$ for the N filters in networks. In this way, the behavioral regularizer is defined as follows:

$$\Omega'(\omega, \omega^*, \mathbf{x}_i, y_i, z) = \sum_{j=1}^{N} \left(\mathrm{W}_j \cdot \|\mathrm{FM}_j(z, \omega, \mathbf{x}_i) - \mathrm{FM}_j(z, \omega^*, \mathbf{x}_i)\|_2^2 \right), \tag{8}$$

ALGORITHM 1: Supervised Attention Algorithm of Filter Importance Estimation

Input:
The target function represented by a neural network z;
The pre-trained source model parameterized with ω^*;
The target dataset $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), \ldots, (\mathbf{x}_n, y_n)\}$;
The number of channels of the target feature map N;
Output: The importance of the filters of the target feature map W;
Initialize the feature extractor of z with the pre-trained parameter ω^*;
Randomly initialize the FC layer of z;
Fix the feature extractor of z;
while not convergence **do**
  Sample a mini-batch of examples M_D from D;
  Update the FC layer of z by backpropagation over M_D;
end while
Initialize W with 0;
for $j = 1$ to N **do**
  $W_j = 0$;
  for $i = 1$ to n **do**
    $L_{i*} = L(z(\mathbf{x}_i, \omega^*), y_i)$;
    Obtain $\omega^{*\backslash \mathrm{j}}$ by setting all elements of the j^{th} filter in ω^* to 0;
    $L_{i*\backslash j} = L(z(\mathbf{x}_i, \omega^{*\backslash \mathrm{j}}), y_i)$;
    $\mathrm{W}_j(z, \omega^*, \mathbf{x}_i, y_i) = L_{i*} - L_{i*\backslash j}$;
    $\mathrm{W}_j = \mathrm{W}_j + \mathrm{W}_j(z, \omega^*, \mathbf{x}_i, y_i)$;
  end for
end for
W = softmax(W);
return W

where W_j refers to the weight assigned to the jth filter (for $\forall 1 \leq j \leq N$) and the behavioral difference between the two feature maps, i.e., $\mathrm{FM}_j(z, \omega, \mathbf{x}_i)$ and $\mathrm{FM}_j(z, \omega^*, \mathbf{x}_i)$, is measured using their Euclid distance (denoted as $\|\cdot\|_2$).

In following sections, we are going to present (1) the design and implementation of feature map extraction $\mathrm{FM}_j(z, \omega, \mathbf{x})$ for $1 \leq j \leq N$, as well as (2) the the attention model that assigns the weight W_j to each filter.

3.3 Feature Map Extraction from Convolution Layers

Given each filter of the network with parameter ω and the input x_i drawn from the target dataset, DELTA first uses such filter to get the corresponding output based on x, then adopts **Rectified Linear Units** (**ReLU**) to rectify the output as a matrix. Furthermore, DELTA formats the output matrices into vectors through concatenation. In this way, DELTA obtains $\mathrm{FM}_j(z, \omega, \mathbf{x}_i)$ for $1 \leq j \leq N$ and $1 \leq i \leq n$ that have been used in Equation (8).

3.4 Weighting Feature Maps with Supervised Attention Models

In DELTA, the proposed regularizer measures the distance between the feature maps generated by the two networks, then aggregates the distances using non-negative weights. Our aim is to pay more attention to those features with greater capacity of discrimination through supervised learning. To obtain such weights for feature maps, we propose a supervised attention method

derived from the backward variable selection, where the weights of features are characterized by the potential performance loss when removing these features from the network.

For clear description, following common conventions, we first define a convolution filter as follow. The parameter of a conv2d layer is a four-dimensional tensor with the shape of (c_{i+1}, c_i, k_h, k_w), where c_i and c_{i+1} represent for the number of channels of the i_{th} and $(i+1)_{th}$ layer, respectively. c_{i+1} filters are contained in such a convolutional layer, each of which with the kernel size of $c_i * k_h * k_w$, taking the feature maps with the size of $c_i * h_i * w_i$ of the ith layer as input, and outputing the feature map with the size of $h_{i+1} * w_{i+1}$.

In particular, we evaluate the weight of a filter as the performance reduction when the filter is disabled in the network. Intuitively, removing a filter with greater capacity of discrimination usually causes higher performance loss. In this way, such channels should be constrained more strictly since a useful representation for the target task is already learned by the source task. Given the pre-trained parameter ω^* and an input image x_i, DELTA sets the weight of the jth channel, using the gap between the empirical losses of the networks on the labeled sample (x_i, y_i) with and without the jth channel, as follows:

$$\mathrm{W}_j(z, \omega^*, \mathbf{x}_i, y_i) = L(z(\mathbf{x}_i, \omega^{*\backslash \mathrm{j}}), y_i) - L(z(\mathbf{x}_i, \omega^*), y_i), \tag{9}$$

where $\omega^{*\backslash \mathrm{j}}$ refers to the modification of original parameter ω^* with all elements of the jth filter set to zero (i.e., removing the jth filter from the network). We aggregate the weights over the entire training set for each channel and then use softmax to normalize the result to ensure all weights are non-negative. The aforementioned supervised attention mechanism yields a filter a higher weight for a specific image if and only if the corresponding feature map in the pre-trained source network is with higher discrimination power—i.e., paying more attention to such filter on that image might bring higher performance gain.

Note that, to calculate $L(z(\mathbf{x}_i, \omega^{*\backslash \mathrm{j}}), y_i)$ and $L(z(\mathbf{x}_i, \omega^*), y_i)$ for supervised attention mechanism, we introduce a baseline algorithm L^2-FE that fixes the feature extractor (with all parameters copied from source networks) and only trains the discriminators using the target task. The L^2-FE model can be viewed as an adaption of the source network (weights) to the target tasks, without further modifications to the outer layer parameters. In our work, we use L^2-FE to evaluate $L(z(\mathbf{x}_i, \omega^{*\backslash \mathrm{j}}), y_i)$ and $L(z(\mathbf{x}_i, \omega^*), y_i)$ using the target datasets. The entire procedure of the supervised attention method is presented in Algorithm 1.

4 EXPERIMENTS AND RESULTS

We have conducted a comprehensive experimental study of the proposed DELTA method. Below we first briefly review the used datasets, followed by a description of experimental procedure and finally our observations.

4.1 Datasets

We evaluate the performance using three benchmarks with different tasks: Caltech 256 for general object recognition, Stanford Dogs 120 for fine-grained object recognition, and MIT Indoors 67 for scene classification. For the first two benchmarks, we used ImageNet as the source domain and Places 365 for the last one.

Caltech 256. Caltech 256 is a dataset with 256 object categories containing a total of 30,607 images. Different numbers of training examples are used by researchers to validate the generalization of proposed algorithms. In this article, we create two configurations for Caltech 256, which have 30 and 60 random sampled training examples, respectively, for each category, following the procedure used in [19].

Stanford Dogs 120. The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. There are exactly 100 examples per category in the training set. It is used for the task of fine-grained image categorization. We do not use the bounding box annotations.

MIT Indoors 67. MIT Indoors 67 is a scene classification task containing 67 indoor scene categories, each of which consists of 80 images for training and 20 for testing. Indoor scene recognition is challenging because both spatial properties and object characters are expected to be extracted.

Caltech-UCSD Birds-200-2011. CUB-200-2011 contains 11,788 images of 200 bird species. Each species is associated with a wikipedia article and organized by scientific classification. Each image is annotated with bounding box, part location, and attribute labels. We use only classification labels during training. While part location annotations are used in a quantitative evaluation of show cases, to explain the transferring effect of our algorithm.

Food-101. Food-101 is a large scale dataset of 101 food categories, with 101,000 images, for the task of fine-grained image categorization. 750 training images and 250 test images are provided for each class. This dataset is challenging because the training images contain some amount of noise.

4.2 Experimental Procedure

We implement our method with ResNet-101 and Inception-V3 as the base networks. For experiment set up we follow almost the same procedure in [19] due to the close relationship between our work and theirs. After training with the source dataset and before fine-tuning the network with the target dataset, we replace the last layer of the base network with random initialization in suit for the target dataset.

For ResNet-101, the input images are resized to 256*256 and normalized to zero mean for each channel, following with data augmentation operations of random mirror and random crop to 224*224. For Inception-V3, images are resized to 320*320 and finally cropped to 299*299. We use a batch size of 64. SGD with the momentum of 0.9 is used for optimizing all models. The learning rate for the base model starts with 0.01 for ResNet-101 and 0.001 for Inception-V3, and is divided by 10 after 6,000 iterations. The training is finished at 9,000 iterations.

The network parameters are regularized as described in Section 4. We use five-fold cross validation for searching the best configurations of the hyperparameter α and β for each experiment by grid search. We set α and β to be 0.0001, 0.001, 0.01, 0.1, and 0.2. As observed from Figure 4, the validation accuracy varies consistently with increasing of β when α differs across experiments. Increasing the value of β improves the performance at the beginning, however, their performance is degraded sharply when the value keeps increasing. The classification accuracy reaches to the best performance 88.7% when $\alpha = 0.01$ and $\beta = 0.01$. The hyperparameters α and β are fixed to 0.01 in the following experiments for DELTA. As was mentioned, our experiments compare DELTA to several key baseline algorithms including L^2, L^2-SP [19], and L^2-FE (see also in Section 3.4), all under the same settings. Each experiment is repeated five times. The average top-1 classification accuracy and standard division are reported.

For our task, we make the following changes to the state-of-the-art architecture L^2-SP. We add attention strategy on the convolutional layers weights corresponding to DELTA, which is called L^2-SP-ATT. For a further exploration of the attention strategy, two methods DELTA with attention and DELTA(w/o ATT) without attention are compared in the following experiments.

4.3 Results and Comparisons

In Figure 3 we plot a sample learning curve of training with different regularization techniques. Comparing these regularization techniques, we observe that our proposed DELTA shows faster convergence than the simple L^2-SP regularization with both step decay (StepLR) and exponential

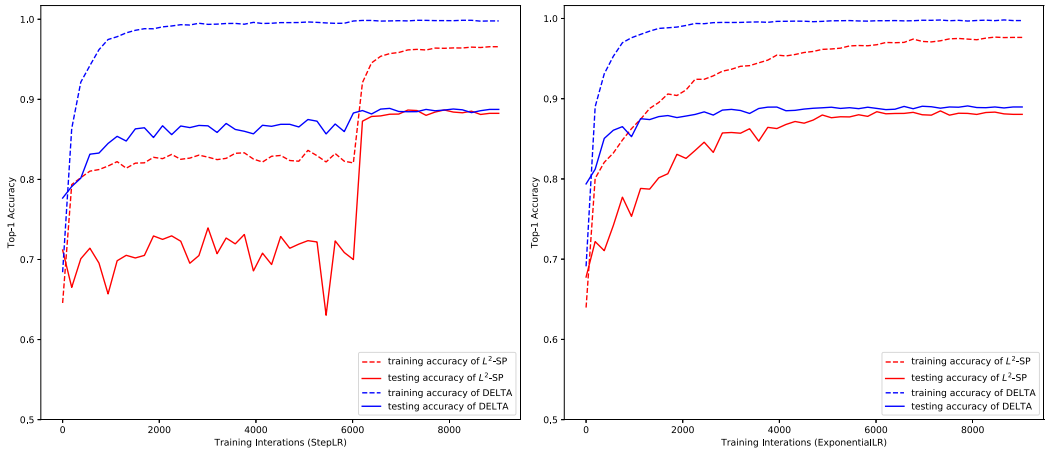

Fig. 3. Learning curves of the proposed feature map-based regularization (DELTA) compared with weight based-regularization (L^2-SP) on the Stanford Dog 120 benchmark using different methods to adjust the learning rate. StepLR: setting the learning rate to the initial value decayed by 0.1 after 6,000 iterations (32 epochs for the Stanford Dogs dataset). ExponentialLR: setting the learning rate to the initial value decayed by 0.93 every epoch.

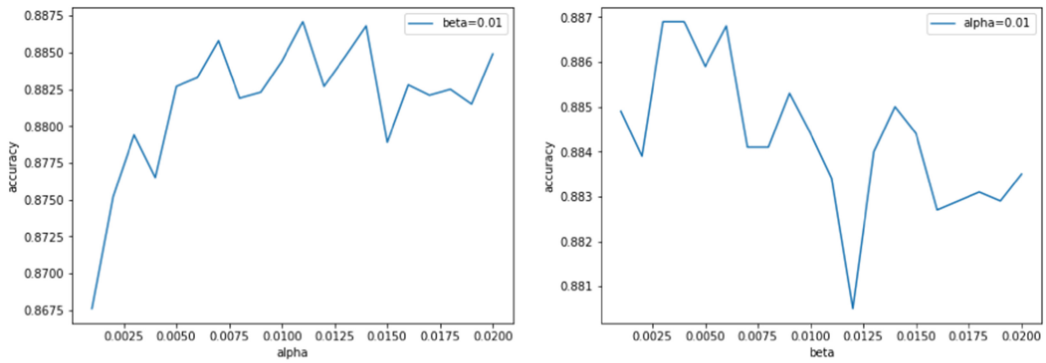

Fig. 4. Classification accuracy (in %) on Caltech 60 for DELTA. According to the two regularization hyperparameters α and β, respectively, applied to the feature maps and layers privately owned by the target network (see Equation (7)).

decay (ExponentialLR) learning rate scheduler. In addition, we find that the learning curve of DELTA is smoother than L^2-SP and it is not sensitive to the learning rate decay happened at the 6,000th iteration when using StepLR.

In Table 1 we show the results of our proposed method DELTA with and without attention, compared to the baseline of L^2-SP reported in [19] and also the naive L^2-FE, L^2, and L^2-SP-ATT methods. We find that on some datasets, fine-tuning using L^2 normalization does not perform significantly better than directly using the pre-trained model as a feature extractor(L^2-FE), while L^2-SP outperforms the naive methods without SPAR. We observe that greater benefits are gained using our proposed attention mechanism. However, L^2-$SP(ATT)$ does not perform better than L^2-SP, indicating that directly imposing the attention mechanism on parameters does not benefit knowledge transfer.

Table 1. Comparison of Top-1 Accuracy with Different Methods

ResNet-101	L^2-FE	L^2	L^2-SP	L^2-$SP(ATT)$	DELTA(w/o ATT)	DELTA
MIT Indoors 67	80.4 ± 0.2	83.7 ± 0.3	85.1 ± 0.1	84.2 ± 0.2	85.3 ± 0.2	**85.5 ± 0.3**
Stanford Dogs 120	84.7 ± 0.1	83.3 ± 0.2	88.3 ± 0.2	88.1 ± 0.3	88.3 ± 0.2	**88.7 ± 0.1**
Caltech 256-30	82.9 ± 0.2	84.7 ± 0.3	85.4 ± 0.2	84.5 ± 0.3	85.7 ± 0.3	**86.6 ± 0.1**
Caltech 256-60	85.3 ± 0.2	87.2 ± 0.3	87.2 ± 0.1	87.1 ± 0.2	87.6 ± 0.2	**88.7 ± 0.1**
CUB-200-2011	61.5 ± 0.1	78.4 ± 0.1	79.5 ± 0.1	77.8 ± 0.1	78.9 ± 0.1	**80.5 ± 0.1**
Food-101	64.3 ± 0.1	85.3 ± 0.1	**86.4 ± 0.1**	85.8 ± 0.1	85.9 ± 0.1	86.3 ± 0.2
Inception-V3	L^2-FE	L^2	L^2-SP	L^2-$SP(ATT)$	DELTA(w/o ATT)	DELTA
MIT Indoors 67	74.9 ± 0.2	74.8 ± 0.4	74.6 ± 0.4	76.8 ± 0.3	76.9 ± 0.3	**78.1 ± 0.4**
Stanford Dogs 120	84.1 ± 0.1	88.6 ± 0.2	**89.4 ± 0.1**	86.4 ± 0.3	88.7 ± 0.1	88.7 ± 0.1
Caltech 256-30	82.5 ± 0.2	83.6 ± 0.3	83.3 ± 0.2	84.4 ± 0.3	83.4 ± 0.3	**84.9 ± 0.2**
Caltech 256-60	84.1 ± 0.1	85.8 ± 0.3	85.3 ± 0.1	84.8 ± 0.2	85.1 ± 0.2	**86.8 ± 0.1**
CUB-200-2011	57.6 ± 0.1	74.3 ± 0.2	75.2 ± 0.1	74.1 ± 0.2	74.5 ± 0.1	**76.5 ± 0.1**
Food-101	55.9 ± 0.1	76.9 ± 0.2	75.9 ± 0.2	75.3 ± 0.4	76.2 ± 0.2	**80.8 ± 0.2**

L^2-FE: Using the pre-trained model as a feature extractor. Baselines: L^2-FE, L^2, and L^2-SP.

Table 2. Comparing Top-1 Accuracy Using Data Augmentation for Three Regularization Methods

ResNet-101	L^2	L^2-SP	L^2-$SP(ATT)$	DELTA
MIT Indoors 67	84.4 ± 0.5	85.2 ± 0.3	84.9 ± 0.3	**85.9 ± 0.3**
Stanford Dogs 120	85.7 ± 0.2	90.8 ± 0.2	89.97 ± 0.2	**91.2 ± 0.2**
Caltech 256-30	85.1 ± 0.4	86.4 ± 0.2	86.2 ± 0.2	**87.1 ± 0.2**
Caltech 256-60	87.4 ± 0.2	88.3 ± 0.1	87.6 ± 0.4	**89.1 ± 0.1**
CUB-200-2011	81.7 ± 0.2	82.3 ± 0.2	82.2 ± 0.1	**82.6 ± 0.2**
Food-101	86.7 ± 0.1	87.2 ± 0.2	87.1 ± 0.1	**87.5 ± 0.1**
Inception-V3	L^2	L^2-SP	L^2-$SP(ATT)$	DELTA
MIT Indoors 67	75.5 ± 0.4	76.5 ± 0.3	76.1 ± 0.4	**78.7 ± 0.3**
Stanford Dogs 120	91.2 ± 0.1	91.9 ± 0.1	88.3 ± 0.2	**92.1 ± 0.1**
Caltech 256-30	84.7 ± 0.2	84.5 ± 0.2	83.5 ± 0.1	**85.5 ± 0.2**
Caltech 256-60	86.1 ± 0.2	86.0 ± 0.1	85.8 ± 0.2	**87.0 ± 0.2**
CUB-200-2011	76.3 ± 0.3	76.3 ± 0.2	76.0 ± 0.3	**77.6 ± 0.3**
Food-101	78.2 ± 0.1	77.2 ± 0.2	80.6 ± 0.2	**82.1 ± 0.2**

Data augmentation is a widely used technique to improve image classification. Following [19], we use a simple data augmentation method and a post-processing technique. First, we keep the original aspect ratio of input images by resizing them with the shorter edge, being 256, instead of ignoring the aspect ratio and directly resizing them to 256*256. Second, we apply 10-crop testing to further improve the performance. In Table 2, we report the experimental results using these techniques with different regularization methods. We observe a clear pattern that with additional data augmentation, all the four evaluated methods L^2, L^2-SP, and L^2-$SP(ATT)$, DELTA have improved classification accuracies while our method still delivers the best one.

4.4 A Case Study and Discussions

To better understand the performance gain of DELTA we perform an experiment where we analyze how parameters of the convolution filters change after fine-tuning. Towards that purpose we randomly sample images from the testing set of Stanford Dogs 120. For ResNet-101, which we use exclusively in this article, we group filters into stages as described in [27]. These stages

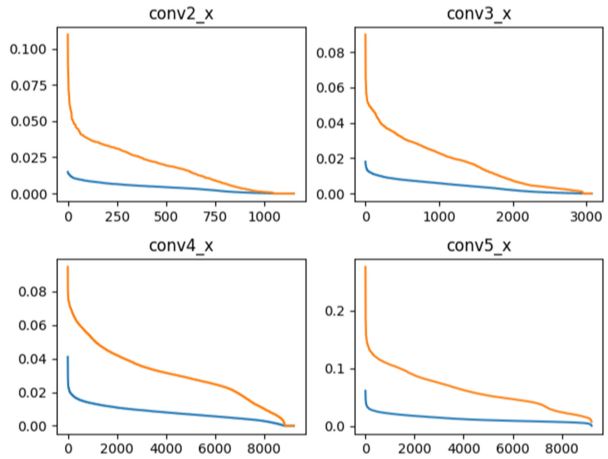

Fig. 5. Distribution of the distance of parameters from the starting point. In ResNet-101, conv2_x, conv3_x, conv4_x, and conv5_x represent for four main stages each of which has stacked convolution layers. The blue line represents for the result of L^2-SP, and the orange line for DELTA.

are conv2_x, conv3_x, conv4_x, and conv5_x. Each stage contains a few stacked blocks and each block is a basic bottleneck unit consisting of three conv2d layers. One conv2d layer is composed of a number of output filters. We flatten each filter into a one-dimension parameter vector for convenience. The Euclidian distance between the parameter vectors before and after fine-tuning is calculated. All distances are sorted as shown in Figure 5.

We observe a sharp difference between the two distance distributions. Our hypothesis of possible cause of the difference is that simply using L^2-SP regularization all convolution filters are forced to be similar to the original ones. Using attention, we allow "unactivated" convolution filters to be re-used for learning more target-adapted deep features. About 90% parameter vectors of DELTA have larger distance than L^2-SP. We also observe that a small number of filters is driven very far away from the initial value (as shown at the left end of the curves in Figure 5). We call such an effect as "unactivated channel re-usage".

To further understand the effect of attention and the implication of "unactivated channel re-usage", we "attribute" the visual attention to the original image to identify the set of pixels having high contributions in the activated feature maps. We select some convolution filters on which the source model (the initialization before fine-tuning) has low activation. For the convenience of analyzing the effect of regularization methods, each element a_i of the original activation map is normalized with

$$a_i = (a_i - \min_j a_j)/(\max_j a_j - \min_j a_j),$$

where the min and max terms in the formula represent for the minimum and maximum value of the whole activation map, respectively. Activation maps of these convolution filter for various regularization methods are presented on each row.

As shown in Figure 6, our first observation is that without attention, DELTA in different images has more or less the same activation maps with other regularization methods. This partially explains the fact that we do not observe significant improvement of DELTA without attention.

Using attention, however, changes the activation map significantly. Regularization of DELTA with attention show obviously improved concentration. With attention (the right-most column in

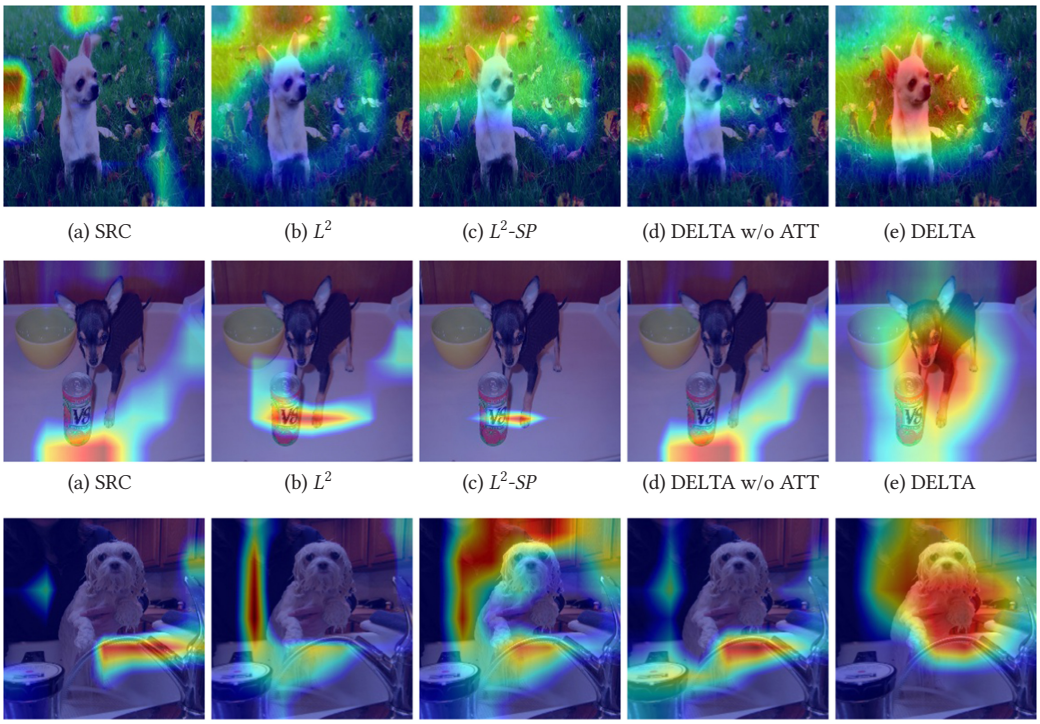

Fig. 6. Illustration of the effect of the attention mechanism for fine-tuning on **Stanford Dog 120** Datasets. DELTA w/o ATT: DELTA without Attentions (equivalent to Knowledge Distillation).

Table 3. Comparing Average Activations on 15 Discriminate Parts of CUB-200-2011 Datasets for Different Regularization Methods

	SRC	L^2	L^2-SP	DELTA(w/o ATT)	DELTA
Average Activations	5.298	5.392	6.258	6.241	**6.367**

Figure 6), we observe a large set of pixels that have high activation at important regions around the head of the animals. We believe this phenomenon provides additional evidence to support our intuition of "unactivated channel re-usage" as discussed in previous paragraphs. Examples with different regularization methods from CUB-200-2011 are shown in Figure 7.

In addition, we include new statistical results of activations on part locations of CUB-200-2011 supporting the above qualitative cases. The CUB-200-2011 datasets defined 15 discriminative parts of birds, e.g., the forehead, tail, beak, and so on. Each part is annotated with a pixel location representing for its center position if it is visible. So for each image, we get several key points which are very important to discriminate its category. Using all testing examples of CUB-200-2011, we calculate normalized activations on these key points of these different regularization methods. As shown in Table 3, DELTA get the highest average activations on those key points, demonstrating that DELTA focused on more discriminate features for bird recognition.

With pixel-wise semantic annotations available, **Intersection over Union** (**IoU**) is used as an evaluation metric to quantify model interpretability [28]. Six types of semantics for CNN filters, i.e., objects, parts, scenes, textures, materials, and colors are selected by [28] to evaluate model

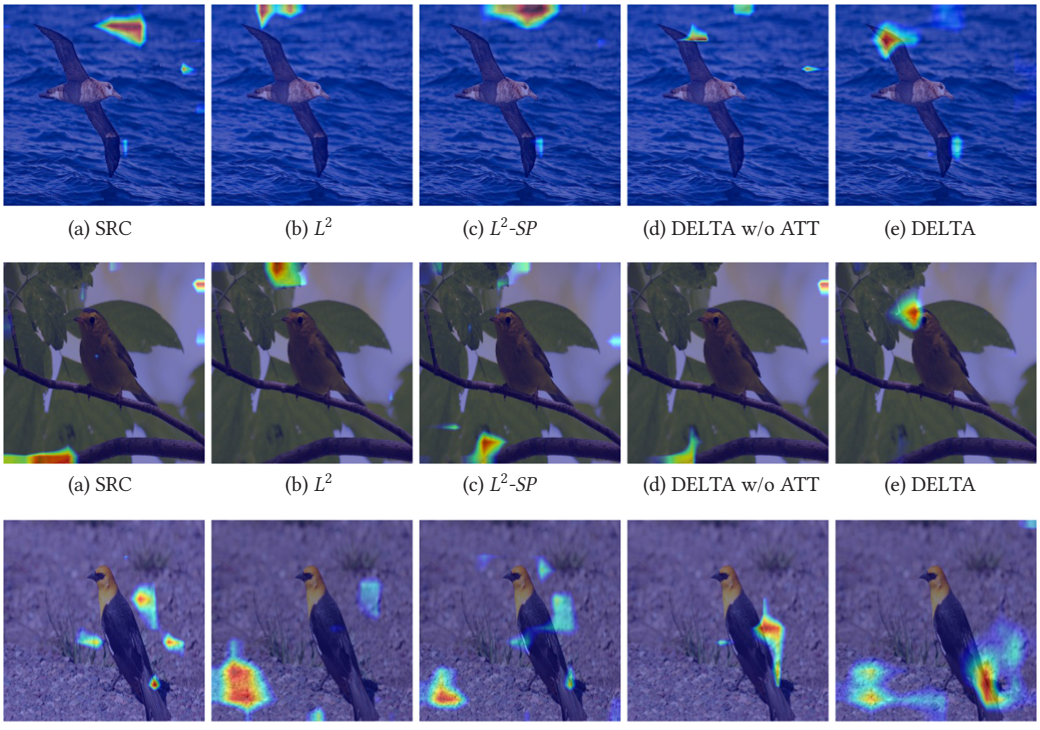

Fig. 7. Illustration of the effect of the attention mechanism for fine-tuning on **Caltech-UCSD Birds-200-2011** Datasets. DELTA w/o ATT: DELTA without Attentions (equivalent to Knowledge Distillation).

Table 4. Interpretability across Different Datasets for DELTA and L^2-SP in Basis of the Representation of Resnet101

DELTA	objects	scenes	parts	textures	colors	sum
MIT Indoors67	57	83	23	32	1	**196**
Stanford Dogs 120	63	70	25	44	1	**203**
Caltech 256-30	51	64	20	42	1	**178**
CUB-200-2011	49	51	18	44	1	**163**
L^2-SP	objects	scenes	parts	textures	colors	sum
MIT Indoors67	50	76	14	39	0	179
Stanford Dogs 120	56	64	19	45	0	184
Caltech 256-30	49	56	15	45	0	165
CUB-200-2011	46	40	15	42	1	144

interpretability. To deeper diagnose the model behaviors by understanding image semantics of representations, we apply network dissection to DELTA and L^2-SP. For comparing the interpretability of units, the experiments focus on the last convolutional layer of each CNN, where semantic detectors emerge most [28]. We calculate the number of unique detectors and the results are listed in Table 4. We observe that, on all four evaluated datasets, DELTA shows a higher overall degree of interpretability in deep representations than L^2-SP. Specifically, DELTA has significant better results for most of the semantic types including objects, scenes, parts, and colors.

While they show comparable effects for textures possibly because lower level representations are more general and tend to be unchanged during fine-tuning.

5 RELATED WORK AND DISCUSSION

In this section, we compare DELTA with the related works, and discuss contributions made in this work.

5.1 Transfer Learning

Transfer learning is a type of machine learning paradigm aiming at transferring the knowledge obtained in a source task to a target task [29, 30]. Our work primarily focuses on inductive transfer learning for DNNs, where the label space of the target task differs from that of the source task. For example, Donahue et al. [31] proposed to train a classifier based on feature extracted from a pre-trained CNN, where a large number of parameters, such as filters, of the source network are reused directly in the target one. This method may overload the target network with tons of irrelevant features (without discrimination power) involved, while the key features of the target task might be ignored. To understand whether a feature can be transferred to the target network, Yosinki et al. [9] quantified the transferability of features from each layer considering the performance gain. Moreover, to understand the factors that may affect deep transfer learning performance, Huh et al. [10] empirically analyzed the features obtained by the ImageNet pre-trained source network to a wide range of computer vision tasks. Recently, more studies to improve the inductive transfer learning from a diverse set of angles have been proposed, such as filter subset selection [32, 33], parameter transfer [21, 34]. Cui et al. [32, 33] demonstrated one could benefit from transfer learning based on a selected subset of the source dataset, which is similar to the target dataset. Moreover, the work [21, 34] studied how to transfer parameters or their statistical distributions of the source and target task. Later, the work [35–37] introduced algorithms to prevent regularizers such as L^2-SP from the hurts to transfer learning, where [35] introduced Batch Spectral Shrinkage to truncate the tail spectrum, [36] proposed to truncate the ill-posed direction of the aggregated gradients, while [37] proposed to deepen back-propagation by incorporating randomness to the FC layer.

For deep transfer learning problems, the most relevant work to our study is [19], where authors investigated regularization schemes to accelerate deep transfer learning while preventing fine-tuning from over-fitting. Their work showed that a simple L^2-norm regularization on top of the "Starting Point as a Reference" optimization can significantly outperform a wide range of regularization-based deep transfer learning mechanisms, such as the standard L^2-norm regularization. Compared to above work, the key contributions made in this article include (1) rather than regularizing the distance between the parameters of source network and target network, DELTA constrains the L^2-norm of the difference between their behaviors (i.e., the feature maps of outer layer outputs in the source/target networks); and (2) the regularization term used in DELTA incorporates a supervised attention mechanism, which re-weights regularizers according to their performance gain/loss.

5.2 Knowledge Distillation

In terms of methodologies, our work is also related to the knowledge distillation for model compression [38, 39]. Generally, knowledge distillation focuses on teacher–student network training, where the teacher and student networks are usually based on the same task [38]. Buciluǎ et al. [40] proposed to guide the student model to learn the logits output softened by a temperature factor of the teacher model. This extra supervision facilitates more information flowing to the model parameters. Ba and Caruana [41] use internal feature maps matching instead as supervision. Romero

et al. [42] proposed to compress the knowledge of a teacher network into a student network by reinforcement learning.

Some recent work also demonstrated the performance of knowledge distillation for transfer learning. But their target task differs from us. Refs. [20, 42] proposed learning to mimic some varieties of internal feature maps, which are similar to us. They finally aimed at teaching the student network to learn the same task as the teacher, one of which is fine-tuning imagenet pre-trained weights to adapt another dataset. They claimed that besides the performance of the origin task, capacity of transferability is also learned by the student. The evidence is that the performance of fine-tuning the smaller student network on a new task is comparable with fine-tuning the larger teacher on the same one. However, the purpose of this article is to improve the transfer learning performance within a fixed network architecture. So the key difference is that they transfer knowledge between two different architectures on identical tasks for purpose of compressing, but we transfer knowledge between two identical architectures on different tasks.

Particularly, we note that [17] proposed to prevent catastrophic forgetting in multi-task learning scenario. They distill the final output of the source network with input of target data as a regularization to force the new shared network to remember old knowledge. This spirit inspired us, but such constraining is too serve for transfer learning, since we only care about the target task. Also, the final output is high-level knowledge, which is difficult to learn for deeper networks with smaller samples and not flexible to be imposed on with attention weights. These work frequently intends to transfer the knowledge in the teacher network to the student one through aligning their outputs of some layers [42]. The most close works to this article are [20, 43], where knowledge distillation technique has been studied to improve transfer learning. Recent works have also studied knowledge transfer crossing modalities, such as image understanding through semantic concepts [44] and text-to-image synthesis [45].

Compared to above work, our work, including other transfer learning studies, intends to transfer knowledge between different source/target image classification tasks (i.e., source and target tasks), though the source/target networks can be viewed as teachers and students, respectively. We follow the conceptual ideas of knowledge distillation to regularize the outer layer outputs of the network (i.e., feature maps), yet further extend such regularization to a supervised transfer learning mechanism by incorporating the labels of target task (which is different from the source task/network). Moreover, a supervised attention mechanism has been adopted to regularize the feature maps according to the importance of filters.

5.3 Attention

Other works relevant to our methodology include: continual learning [17, 46], attention mechanism for CNN models [47–50], among others. Early work on attention was motivated by human perception process, which uses top information to guide bottom-up feedforward process [47, 51]. Attention mechanism is widely and successfully applied in natural language processing and computer vision tasks. Bahdanau et al. [52] adapted attention to for neural machine translation with a bidirectional recurrent networks. Wang et al. [53] proposed an attention-based long short-term memory network for aspect-level sentiment classification. Refs. [48, 54–56] exploited attention mechanism in multi-modality tasks, such as image captioning, video captioning, action recognition, and visual question–answering. Zheng et al. [57] proposed a part learning approach by a multi-attention CNN, where part generation and feature learning can reinforce each other. Fu et al. [58] proposed a recurrent attention CNN which recursively learns discriminative region attention and region-based feature representation at multiple scales. Zhao et al. [49] propose a **diversified visual attention network (DVAN)** aiming at the problem of fine-grained object

classification. Rodríguez et al. [50] instead applied attention at convolutional feature activations aiming at learning better lower level features.

Attention in CNN has been widely used in network visualization. Zagoruyko et al. [43] used activation-based and gradient-based spatial attention maps, to improve the performance of knowledge distillation by force the student network to mimic them. There attention weights are estimated along channel dimensions and operated over different spatial positions, while attention weights for DELTA are estimated on top of feature maps over different channels.

6 CONCLUSION

In this article, we studied a regularization technique that transfers the behaviors and semantics of the source network to the target one through constraining the difference between the feature maps generated by the convolution layers of source/target networks with attentions. Specifically, we designed a regularized learning algorithm DELTA that models the difference of feature maps with attentions between networks, where the attention models are obtained through supervised learning. Moreover, we further accelerate the optimization for regularization using SPAR. Our extensive experiments evaluated DELTA using several real-world datasets based on commonly used CNNs. The experiment results show that DELTA significantly outperforms the state-of-the-art transfer learning methods.

REFERENCES

[1] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, and Jun Huan. 2019. DELTA: Deep learning transfer using feature map with attention for convolutional networks. In *Proceedings of the International Conference on Learning Representations*.

[2] C. Hsu and C. Lin. 2018. CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data. *IEEE Transactions on Multimedia* 20, 2 (2018), 421–429.

[3] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang. 2015. Rating image aesthetics using deep learning. *IEEE Transactions on Multimedia* 17, 11 (2015), 2021–2034.

[4] N. Kumar and A. Sethi. 2016. Fast learning-based single image super-resolution. *IEEE Transactions on Multimedia* 18, 8 (2016), 1504–1515.

[5] D. Guo, W. Li, and X. Fang. 2018. Fully convolutional network for multi-scale temporal action proposals. *IEEE Transactions on Multimedia* 20, 12 (2018), 3428–3438.

[6] X. Lin, J. Liu, and X. Kang. 2016. Audio recapture detection with convolutional neural networks. *IEEE Transactions on Multimedia* 18, 8 (2016), 1480–1487.

[7] N. Takahashi, M. Gygli, and L. Van Gool. 2018. AENet: Learning deep audio features for video analysis. *IEEE Transactions on Multimedia* 20, 3 (2018), 513–524.

[8] V. E. Liong, J. Lu, Y. Tan, and J. Zhou. 2017. Deep video hashing. *IEEE Transactions on Multimedia* 19, 6 (2017), 1209–1219.

[9] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Proceedings of the Advances in Neural Information Processing Systems*. 3320–3328.

[10] Minyoung Huh, Pulkit Agrawal, and Alexei A. Efros. 2016. What makes ImageNet good for transfer learning? arXiv:1608.08614. Retrieved from https://arxiv.org/abs/1608.08614.

[11] M. Soltanian, S. Amini, and S. Ghaemmaghami. 2020. Spatio-temporal VLAD encoding of visual events using temporal ordering of the mid-level deep semantics. *IEEE Transactions on Multimedia* 22, 7 (2020), 1769–1784.

[12] Y. Zha, T. Ku, Y. Li, and P. Zhang. 2020. Deep position-sensitive tracking. *IEEE Transactions on Multimedia* 22, 1 (2020), 96–107.

[13] P. Wang, W. Li, Z. Gao, C. Tang, and P. O. Ogunbona. 2018. Depth pooling based large-scale 3-D action recognition with convolutional neural networks. *IEEE Transactions on Multimedia* 20, 5 (2018), 1051–1061.

[14] S. S. Mukherjee and N. M. Robertson. 2015. Deep head pose: Gaze-direction estimation in multi-modal video. *IEEE Transactions on Multimedia* 17, 11 (2015), 2094–2107.

[15] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. 2019. Towards understanding the transferability of deep representations. arXiv:1909.12031. Retrieved from https://arxiv.org/abs/1909.12031.

[16] Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proceedings of the International Conference on Learning Representations*.

[17] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 12 (2017) 2935–2947.
[18] Phillippe Rigollet and Jan-Christian Hütter. 2015. High dimensional statistics. *Lecture notes for course 18S997* 813 (2015), 814.
[19] Xuhong Li, Yves Grandvalet, and Franck Davoine. 2018. Explicit inductive bias for transfer learning with convolutional networks. In *Proceedings of the 35th International Conference on Machine Learning*.
[20] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[21] Mehmet Aygun, Yusuf Aytar, and Hazim Kemal Ekenel. 2017. Exploiting convolution filter patterns for transfer learning. In *Proceedings of the International Conference on Computer Vision Workshops*. 2674–2680.
[22] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. 2019. Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10705–10714.
[23] Jun Deng, Wei Dong, Richard Socher, Li-Jia Li, Kuntai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.
[24] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. 2011. Novel dataset for fine-grained image categorization. In *Proceedings of the First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*. Colorado Springs, CO.
[25] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 7 (2006), 1527–1554.
[26] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Proceedings of the Advances in Neural Information Processing Systems*. 153–160.
[27] Kaiming he, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
[28] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network Dissection: Quantifying interpretability of deep visual representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 3319–3327.
[29] Rich Caruana. 1997. Multi-task learning. *Machine Learning* 28, 1 (1997), 41–75.
[30] Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.
[31] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the International Conference on Machine Learning*. 647–655.
[32] Weifeng Ge and Yizhou Yu. 2017. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10–19.
[33] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. 2018. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4109–4118.
[34] Yinghua Zhang, Yu Zhang, and Qiang Yang. 2019. Parameter transfer unit for deep neural networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 82–95.
[35] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. 2019. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. In *Proceedings of the Advances in Neural Information Processing Systems*. 1906–1916.
[36] R. Wan, H. Xiong, X. Li, Z. Zhu, and J. Huan. 2019. Towards making deep transfer learning never hurt. In *Proceedings of the 2019 IEEE International Conference on Data Mining*. 578–587.
[37] Xingjian Li, Haoyi Xiong, Haozhe An, Cheng-Zhong Xu, and Dejing Dou. 2020. RIFLE: Backpropagation in depth for deep transfer learning through re-initializing the fully-connected LayEr. In *Proceedings of the International Conference on Machine Learning*. 6010–6019.
[38] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arXiv:1503.02531. Retrieved from https://arxiv.org/abs/1503.02531.
[39] S. Lin, R. Ji, C. Chen, D. Tao, and J. Luo. 2019. Holistic CNN compression via low-rank decomposition with knowledge transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 12 (2019), 2889–2905.
[40] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 535–541.
[41] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Proceedings of the Advances in Neural Information Processing Systems*. 2654–2662.

[42] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. arXiv:1412.6550. Retrieved from https://arxiv.org/abs/1412.6550.

[43] Sergey Zagoruyko and Nikos Komodakis. 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR'17)*.

[44] C. Yan, L. Li, C. Zhang, B. Liu, Y. Zhang, and Q. Dai. 2019. Cross-modality bridging and knowledge transferring for image understanding. *IEEE Transactions on Multimedia* 21, 10 (2019), 2675–2685.

[45] M. Yuan and Y. Peng. 2020. CKD: Cross-task knowledge distillation for text-to-image synthesis. *IEEE Transactions on Multimedia* 22, 8 (2020), 1955–1968.

[46] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America* 114, 13 (2017), 3521–3526.

[47] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Proceedings of the Advances in Neural Information Processing Systems*. 2204–2212.

[48] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*. 2048–2057.

[49] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan. 2017. Diversified visual attention networks for fine-grained object classification. *IEEE Transactions on Multimedia* 19, 6 (2017), 1245–1256.

[50] P. Rodríguez, D. Velazquez, G. Cucurull, J. M. Gonfaus, F. X. Roca, and J. Gonzàlez. 2020. Pay Attention to the Activations: A modular attention mechanism for fine-grained image recognition. *IEEE Transactions on Multimedia* 22, 2 (2020), 502–514.

[51] Ronald A. Rensink. 2000. The dynamic representation of scenes. *Visual Cognition* 7, 1–3 (2000), 17–42.

[52] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR'15)*.

[53] Yequan Wang, Minlie Huang, Li Zhao, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 606–615.

[54] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai. 2020. Corrections to "STAT: Spatial-Temporal Attention Mechanism for Video Captioning". *IEEE Transactions on Multimedia* 22, 3 (2020), 830–830.

[55] Z. Yang, Y. Li, J. Yang, and J. Luo. 2019. Action Recognition With Spatio-Temporal Visual Attention on Skeleton Image Sequences. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 8 (2019), 2405–2415.

[56] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 21–29.

[57] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. 2017. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the International Conference on Computer Vision*.

[58] Jianlong Fu, Heliang Zheng, and Tao Mei. 2017. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

Received February 2021; accepted July 2021