Abstract

The empirical fact that classifiers, trained on given data collections, perform poorly when tested on data acquired in different settings is theoretically explained in domain adaptation through a shift among distributions of the source and target domains. Alleviating the domain shift problem, especially in the challenging setting where no labeled data are available for the target domain, is paramount for having visual recognition systems working in the wild. As the problem stems from a shift among distributions, intuitively one should try to align them. In the literature, this has resulted in a stream of works attempting to align the feature representations learned from the source and target domains. Here we take a different route. Rather than introducing regularization terms aiming to promote the alignment of the two representations, we act at the distribution level through the introduction of Domain Alignment Layers (DIAL), able to match the observed source and target data distributions to a reference one. Thorough experiments on three different public benchmarks we confirm the power of our approach.

Contents

1. Introduction 2
2. Related Work 3
3. DIAL: Domain Alignment Layers 3
   3.1. Source and Target Predictors 4
   3.2. Training 4
   3.3. Implementation Notes 5
4. Experiments 5
   4.1. Experimental Setup 6
   4.2. Results 7
   4.3. Analysis of the Proposed Method 8
5. Conclusions 9

A. Office 31 with Sampling Protocol [23] 9
B. Office-Caltech with DIAL – Inception-BN 9
C. Cross-Dataset Testbed with DIAL – ResNet 10

*This work was partially supported by the ERC grant 637076 - RoboExNovo (B.C.), and the CHIST-ERA project ALOOF (B.C, F. M. C.).
1. Introduction

Many scientists today believe we are witnessing the golden age of computer vision. The massive adoption of machine learning and, in particular, of deep learning techniques as well as the availability of large fully annotated datasets have enabled amazing progresses in the field. A natural question is if the novel generation of computer vision technologies is robust enough to operate in real world scenarios. One of the main problems is that the central assumption that training and test data are independently and identically drawn from the same distribution does not hold in many real world applications. Indeed, researchers have shown that, even with powerful deep learning models, the so-called domain shift problem can be alleviated but not eliminated. Therefore, solving this issue represents a fundamental challenge for developing computer vision systems working in the wild.

In the last few years the research community has devoted significant efforts in addressing domain shift, and several methods for domain adaptation have emerged. Domain adaptation focuses on improving learning performance in a target task by leveraging knowledge from a related source task. In this context, the specific problem of unsupervised domain adaptation, i.e. no labeled data are available in the target domain, deserves special attention, as in many applications annotating data is not only a tedious operation but may be not possible.

Several approaches have been proposed for unsupervised domain adaptation in the past, both considering handcrafted features and deep models. In particular, recent works based on convolutional neural networks (CNN) have achieved remarkable performance. Most of these methods attempt to reduce the discrepancy among source and target distributions by learning features that are invariant to the domain shift. Two main strategies are typically employed. One is based on the minimization of Maximum Mean Discrepancy (MMD) \cite{18, 19}: the distributions of the learned source and target representations are forced to be as similar as possible by minimizing the distance between their mean embeddings. The other strategy \cite{6, 28} relies on the so-called domain-confusion loss. This loss is introduced to learn an auxiliary classifier predicting if a sample comes from the source or from the target domain. Intuitively, by maximizing this term, i.e. by imposing the auxiliary classifier to exhibit poor performance, domain-invariant features can be obtained. Following these recent approaches, we present a domain adaptation method which simultaneously learns discriminative deep representations while being able to cope with domain shift in the unsupervised setting. Different from previous works, we do not focus on learning domain-invariant features by explicitly optimizing additional loss terms (e.g. MMD, domain-confusion). We argue instead that different predictors should be used for source and target samples and that specific Domain Alignment layers (DA-layers) can be embedded into a deep architecture to align the observed data distributions to a canonical one. Moreover, we propose to use the unlabeled target data to construct a prior distribution on the network parameters, biasing the learned solution towards models that are able to separate well the classes in the target domain. Finally, we show how, under certain conditions, our approach can be completely implemented by employing traditional layers, i.e. batch normalization (BN, \cite{13}) and the cross-entropy loss. We call our algorithm DIAL – Domain Alignment Layers. An extensive experimental evaluation, conducted on three publicly available datasets, demonstrates that DIAL greatly alleviates the domain discrepancy and outperforms state of the art techniques.

**Contributions.** The contribution of this work is three-fold. First, we present an effective approach for unsupervised domain adaptation. Our proposal is based on the introduction of DA-layers to explicitly address the domain shift problem without assuming the existence of a single predictor for source and target domains. Second, we show that, under certain assumptions, our method can be implemented by composing existing CNN layers. Furthermore, in contrast to previous works \cite{6, 18, 19, 28}, our DA-layers do not introduce any additional meta-parameters into the network. Third, we perform an extensive experimental analysis on three different benchmarks. We find that our unsupervised domain adaptation approach outperforms state-of-the-art methods and can be applied to different CNN architectures, consistently improving their performance in domain adaptation problems.
2. Related Work

Unsupervised domain adaptation focuses on the scenario where labeled data are only available in the source domain. Traditional methods addressed the problem of reducing the discrepancy between the source and the target distributions by considering two main strategies. The first is based on instance re-weighting \cite{2, 7, 12, 50, 51}. Initially, source samples are assigned different importance according to their similarity with the target data. Then, the re-weighted instances are used to learn a classification/regression model for the target domain. Following this scheme, Huang et al. \cite{12} introduced Kernel Mean Matching, a nonparametric method to set source sample weights without explicitly estimating the data distributions. Gong et al. \cite{2} proposed to automatically discover landmark datapoints, i.e. the subset of source instances being more similar to target data, and used them to create domain-invariant features. Chu et al. \cite{2} formalized the two tasks of sample selection and classifier learning within a single optimization problem. While these works considered hand-crafted features, recently similar ideas have been applied to deep models. For instance, Zeng et al. \cite{51} described an unsupervised domain adaptation approach for pedestrian detection using deep autoencoders to weight the importance of source training samples.

A second strategy for unsupervised domain adaptation is based on feature alignment, i.e. source and target data are projected in a common subspace as to reduce the distance among the associated distributions. This approach attracted considerable interest in the past years and several different methods have been proposed, both considering shallow models \cite{5, 8, 17} and deep architectures \cite{6, 18, 28}. Focusing on recent deep domain adaptation methods, two different schemes are typically considered for aligning feature representations: (i) multiple adaptation schemes are introduced in order to reduce Maximum Mean Discrepancy \cite{18, 19, 25} or (ii) deep features are learned in a domain-adversarial setting, i.e. maximizing a domain confusion loss \cite{6, 28}. Also our approach belongs to the category of methods employing deep learning for domain adaptation. However, we significantly depart from previous works and we reduce the discrepancy between source and target distributions by introducing a novel domain alignment approach based on our DA-layers. At the time of our submission, a work by Li et al. \cite{16} exploiting batch normalization for domain adaptation appeared online. While our approach develops from a similar intuition, we show in Section 3 that our method can be regarded as a generalization of \cite{16}, as we consider arbitrary transformation in our DA layers and we also introduce a prior over the network parameters in order to benefit from the target samples during training. Experiments presented in Section 4 show the significant added value of our idea.

3. DIAL: Domain Alignment Layers

Let $\mathcal{X}$ be the input space (e.g. images) and $\mathcal{Y}$ the output space (e.g. image categories) of our learning task. In unsupervised domain adaptation, we have a source domain and a target domain that are identified via probability distributions $p^s_{xy}$ and $p^t_{xy}$, respectively defined over $\mathcal{X} \times \mathcal{Y}$. The two distributions are in general different and unknown, but we are provided with a source dataset $S = \{(x'_1, y'_1), \ldots, (x'_m, y'_m)\}$ of $i.i.d.$ observations from $p^s_{xy}$ and an unlabeled target dataset $T = \{x'_1, \ldots, x'_m\}$ of $i.i.d.$ observations from the marginal $p^t_x$. The goal is to estimate a predictor from $S$ and $T$ that can be used to classify sample points from the target domain. This task is particularly challenging because on one hand we lack direct observations of labels from the target domain and on the other hand the discrepancy between the source and target domain distributions prevents a predictor trained on $S$ to be readily applied to the target domain.

A number of state-of-the-art methods try to reduce the domain discrepancy by performing some form of alignment at the feature or classifier level. In particular, the recent, most successful methods try to couple the training process and the domain adaptation step within deep neural architectures \cite{6, 18, 19}, as this solution enables alignments at different levels of abstraction. The approach we propose embraces the same philosophy, but we depart from the majority of the methods making the assumption that the domain alignment suffices to apply the same predictor to the source and target domains. This is motivated by an impossibility theorem \cite{1}, which intuitively states that no learner relying on the covariate shift hypothesis, i.e. $p^s_{xy} = p^t_{xy}$, and achieving a low discrepancy between the source and target unlabeled distributions $p^s_x$ and $p^t_x$, can succeed in domain adaptation without further relatedness assumptions between training and target distributions. For this reason, we assume that the source and target predictors are in general different functions. Nonetheless, both predictors depend on a common parameter $\theta$ belonging to a set $\Theta$, which represents the hypothesis space of our learning task. The common hypothesis couples explicitly the two predictors, but it is not directly involved in the alignment of the source and target domains. This contrasts with the majority of state-of-the-art methods that augment the loss function used to train their predictors with a regularization term penalizing discrepancies between source and target representations (see, e.g. \cite{6, 18, 19}). The perspective we take is different and consists in hardcoding, in a sense, the desired domain-invariance properties into the source and target predictors through the introduction of so-called Domain-Alignment layers. The rest of this section is devoted to providing the details of our method.
3.1. Source and Target Predictors

The source and target predictors are implemented as two deep neural networks being almost identical, as they share the same structure and the same weights (given by the parameter $\theta$). However, the two networks contain also a number of special layers, called Domain-Alignment layers (DA-layers), which implement a domain-specific operation. Indeed, the role of such layers is to apply a data transformation that aligns in a sense the observed input distribution with a reference distribution (see, Figure [1]). Since in general the input distributions of the source and target predictors differ, while the reference distribution stays the same, we have that the two predictors undergo different transformations in the corresponding DA-layers. Consequently, the source and target predictors implement de facto different functions. More details about the neural network architectures are given in Section [4].

To better understand how the domain-alignment transformation works, we consider a single DA-layer in isolation. The desired output distribution, namely the reference distribution, is decided a priori and thus known. The input distribution instead is unknown, but we can rely on a distribution, is decided a priori and thus known. The information works, we consider a single DA-layer in isolation the input distributions of the source and target predictors, using the observations provided by the source dataset $S$ and the target dataset $T$.

As we stick to a discriminative model, the unlabeled target dataset cannot be employed to express the data likelihood. However, we can exploit $T$ to construct a prior distribution of the parameter $\theta$. Accordingly, we shape a posterior distribution of $\theta$ given the observations $S$ and $T$ as

$$\pi(\theta|S, T) \propto \pi(y_S|x_S, \theta)\pi(\theta|T),$$

where $y_S = \{y_1, \ldots, y_n\}$ and $x_S = \{x_1, \ldots, x_n\}$ collect the labels and data points of the observations in $S$, respectively. The posterior distribution is maximized over $\Theta$ to obtain a maximum a posteriori estimate $\hat{\theta}$ of the parameter used in the source and target predictors:

$$\hat{\theta} \in \arg \max_{\theta \in \Theta} \pi(\theta|S, T).$$

The term $\pi(y_S|x_S, \theta)$ in (1) represents the likelihood of $\theta$ with respect to the source dataset, while $\pi(\theta|T)$ is the prior term depending on the target dataset, which acts as a regularizer in the classical learning theory sense. The likelihood decomposes into the following product over sample points, due to the data sample i.i.d. assumption:

$$\pi(y_S|x_S, \theta) = \prod_{i=1}^{n} f_{y_i|x_i}^\theta(x_i; x_S),$$

where $f_{y_i|x_i}^\theta(x_i; x_S)$ is the probability that sample point $x_i$ takes label $y_i$ according to the source predictor.

Before delving into the details of the prior term, we would like to remark on the absence of an explicit component in the probabilistic model that tries to align the source

1. There is an unpublished work [16] which is in parallel to ours, that uses batch normalization for a-posteriori domain adaptation.
2. We use notation $x_T$ and $T$ interchangeably.
and target distributions. This is because in our model the domain-alignment step is taken over by each predictor, independently, via the domain-alignment layers as shown in the previous subsection.

Prior distribution. The prior distribution of the parameter $\theta$ shared by the source and target predictors is constructed from the observed, target data distribution. This choice is motivated by the theoretical possibility of squeezing more bits of information from unlabeled data points insofar as they exhibit low levels of class overlap [21]. Accordingly, it is reasonable to bias a priori a predictor based on the degree of label uncertainty that is observed when the same predictor is applied to the target samples. Uncertainty in this sense can be measured for an hypothesis $\theta$ in terms of the empirical entropy of $y|\theta$ conditioned on $x$ as follows

$$h(\theta|T) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{y \in Y} f_{y}^{\theta}(x_i^t; x_T) \log f_{y}^{\theta}(x_i^t; x_T) , \quad (4)$$

where $f_{y}(x_i^t; T)$ represents the probability that sample $x_i^t$ takes label $y$ according to the target predictor.

It is now possible to derive a prior distribution $\pi(\theta|T)$ in terms of the label uncertainty measure $h(\theta|T)$ by requiring the prior distribution to maximize the entropy under the constraint $\int h(\theta|T) \pi(\theta|T) d\theta = \epsilon$, where the constant $\epsilon > 0$ specifies how small the label uncertainty should be on average. This yields a concave, variational optimization problem with solution:

$$\pi(\theta|T) \propto \exp (-\lambda h(\theta|T)) , \quad (5)$$

where $\lambda$ is the Lagrange multiplier corresponding to $\epsilon$. The resulting prior distribution satisfies the desired property of preferring models that exhibit well separated classes (i.e., having lower values of $h(\theta|T)$), thus enabling the exploitation of the information content of unlabeled target observations within a discriminative setting [9].

Inference Once we have estimated the optimal network parameters $\hat{\theta}$ by solving (2), we can remove the dependence of the target predictor $f_{y}^{\theta}(x; x_T)$ from $x_T$. In fact, after fixing $\hat{\theta}$, the input distribution to each DA-layer also becomes fixed, and we can thus compute and store the required transformation once and for all. For instance, for the special case discussed in Section 3.1 this boils down to store the values of $\mu(D)$ and $\sigma(D)$.

3.3. Implementation Notes

As mentioned in Section 1 our method can be implemented with layers commonly available in deep learning toolkits. In particular, when considering channel-wise linear transformations and a standard normal distribution as reference, a DA-layer can be built from Batch Normalization, concatenation and split layers as in Figure 2. During training, the batches contain a fixed number of source samples, followed by a fixed number of target samples. Inside each DA-layer we separate source and target data simply by slicing the batch along its first dimension, and give each slice as input to a separate BN layer. The outputs of the BNs are then concatenated again and fed to the following layer (usually a scale / bias layer). At test time we remove the split, source BN and concatenation layers and directly connect the target BN to the DA-layer’s input and output.

By replacing the optimization problem in (2) with the equivalent minimization of the negative logarithm of $\pi(\theta|S, T)$ and combining (1), (3), (4) and (5) we obtain a loss function $L(\theta) = L^s(\theta) + \lambda L^t(\theta)$, where:

$$L^s(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \log f_{y_i}^{\theta}(x_i^s; x_S) ,$$

$$L^t(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{y \in Y} f_{y}^{\theta}(x_i^t; x_T) \log f_{y}^{\theta}(x_i^t; x_T) .$$

The term $L^s(\theta)$ is the standard log-loss applied to the source samples, while $L^t(\theta)$ is an entropy loss applied to the target samples. The second term can be implemented by feeding $f_{y}^{\theta}(x_i^t; x_T)$ to both inputs of a cross-entropy loss layer, where supported by the deep learning toolkit of choice. In our implementation, based on Caffe [14], we obtain it by slightly modifying the existing SigmoidLoss layer. The code will be made publicly available.

4. Experiments

In this section we extensively evaluate our approach and compare it with state-of-the-art unsupervised domain-adaptation methods. We also provide a detailed analysis of the proposed framework, performing a sensitivity study and demonstrating empirically the effect of our domain-alignment strategy.
4.1. Experimental Setup

Datasets

We evaluate the proposed approach on three publicly-available datasets.

The Office 31 [23] dataset is a standard benchmark for testing domain-adaptation methods. It contains 4652 images organized in 31 classes from three different domains: Amazon (A), DSLR (D) and Webcam (W). Amazon images are collected from amazon.com, Webcam and DSLR images were manually gathered in an office environment. In our experiments we consider all possible source/target combinations of these domains and adopt the full protocol setting [7], i.e. we train on the entire labeled source and unlabeled target data and test on annotated target samples.

The Office-Caltech [8] dataset is obtained by selecting the subset of 10 common categories in the Office31 and the Caltech256 [10] datasets. It contains 2533 images of which about half belong to Caltech256. Each of Amazon (A), DSLR (D), Webcam (W) and Caltech256 (C) are regarded as separate domains. In our experiments we only consider the source/target combinations containing C as either the source or target domain.

To further perform an analysis on a large-scale dataset, we also consider the recent Cross-Dataset Testbed introduced in [27] and specifically the Caltech-ImageNet setting. This dataset was obtained by collecting the images corresponding to the 40 classes shared between the Caltech256 (C) and the Imagenet (I) [3] datasets. To facilitate comparison with previous works [24, 26, 28] we perform experiments in two different settings. The first setting, adopted in [26, 28], considers 5 splits obtained by selecting 5534 images from ImageNet and 4366 images from Caltech256 across all 40 categories. The second setting, adopted in [24], uses 3847 images for Caltech256 and 4000 images for ImageNet.

Networks and Training

We apply the proposed method to two state-of-the-art CNNs, i.e. AlexNet [15] and Inception-BN [13]. We train our networks using mini-batch stochastic gradient descent with momentum, as implemented in the Caffe library, using the following meta-parameters: weight decay $5 \times 10^{-4}$, momentum 0.9, initial learning rate $10^{-3}$. We augment the input data by scaling all images to $256 \times 256$ pixels, randomly cropping $227 \times 227$ pixels (for AlexNet) or $224 \times 224$ pixels (Inception-BN) patches and performing random flips. In all experiments we choose the parameter $\lambda$ by cross-validation.

AlexNet [15] is a well-known architecture with five convolutional and three fully-connected layers, denoted as fc6, fc7 and fc8. The outputs of fc6 and fc7 are commonly used in the domain-adaptation literature as pre-trained feature representations [4, 24] for traditional machine learning approaches. In our experiments we modify AlexNet by appending a DA-layer to each fully-connected layer. Differently from the original AlexNet, we do not perform dropout on the outputs of fc6 and fc7. We initialize the network parameters from a publicly-available model trained on the ILSVRC-2012 data, we freeze all the convolutional layers, and increase the learning rate of fc8 by a factor of 10. During training, each mini-batch contains a number of source and target samples proportional to the size of the corresponding dataset, while the batch size remains fixed at 256. We train for a total of 60 epochs (where “epoch” refers to a complete pass over the source set), reducing the learning rate by a factor 10 after 40 epochs.

Inception-BN [13] is a very deep architecture obtained by concatenating “inception” blocks. Each block is composed of several parallel convolutions with batch normalization and pooling layers. To apply the proposed method to Inception-BN, we replace each batch-normalization layer with a DA-layer. Similarly to AlexNet, we initialize the net-

| Method | Source Target | Amazon | Webcam | DSLR | Amazon | Webcam | DSLR | Average |
|--------|---------------|--------|--------|------|--------|--------|------|---------|
| AlexNet ~ source [15] | | 62.7 | 66.8 | 45.4 | 94.9 | 59.6 | 95.9 | 76.2 |
| DCC [29] | | 61.8 | 64.4 | 52.2 | 98.5 | 54.0 | 96.0 | 72.9 |
| DAN [18] | | 68.5 | 67.0 | 53.1 | 99.0 | 51.1 | 95.4 | 70.1 |
| ReverseGrad [6] | | 73.0 | – | – | 99.2 | – | 96.4 | – |
| DIAL – AlexNet-No DA | | 62.7 | 66.8 | 45.4 | 99.4 | 42.3 | 95.4 | 68.7 |
| DIAL – AlexNet | | 73.2 | 71.7 | 57.2 | 99.3 | 59.6 | 95.9 | 76.2 |
| Inception-BN ~ source [13] | | 70.3 | 70.5 | 57.9 | 100.0 | 60.1 | 94.3 | 75.5 |
| AdaBN [16] | | 77.4 | 73.1 | 57.4 | 99.8 | 59.8 | 95.7 | 76.7 |
| AdaBN + CORAL [16] | | 75.4 | 72.7 | 60.5 | 99.6 | 59.0 | 96.2 | 77.2 |
| DIAL – Inception-BN | | 82.9 | 87.3 | 62.6 | 99.9 | 63.1 | 98.2 | 82.4 |

Table 1: Results on the Office-31 dataset using the full protocol.
Comparing our approach, applied to both AlexNet and Inception-BN, with several state-of-the-art methods on the Office-31 dataset. In particular, we consider: two shallow methods, i.e. Geodesic Flow Kernel (GFK) [8] and Subspace Alignment (SA) [5]; several deep methods based on AlexNet-like architectures, i.e. Deep Adaptation Networks (DAN) [18], Deep Domain Confusion (DDC) [29], the ReverseGrad network [6]; a recent deep method based on the Inception-BN architecture, i.e. AdaBN [16] with and without CORAL feature alignment [24]. We compare these baselines to the AlexNet and Inception-BN networks modified with our approach as explained in Section 4.1. In the table our approach is denoted as DIAL – AlexNet and DIAL – Inception-BN. As a reference, we further report the results obtained considering standard AlexNet and Inception-BN networks trained only on source data, and with AlexNet trained using the proposed approach but without domain-alignment layers (DIAL – AlexNet-No DA).

Unsurprisingly, the classification accuracy of the two methods based on hand-crafted features is greatly inferior to all other approaches. Among the deep methods based on the AlexNet architecture, DIAL – AlexNet shows the best average performance. On the other hand, the results obtained with DIAL – AlexNet-No DA are the lowest among the AlexNet-derived architectures, highlighting the effectiveness of the proposed domain-alignment technique. Among the methods based on Inception-BN, our approach considerably outperforms the others, with an average accuracy of five points higher than the second best, and improvements on the single experiments as high as ten points. It is interesting to note that the relative increase in accuracy from the source-only Inception-BN to DIAL – Inception-BN is higher than that from the source only AlexNet to DIAL – AlexNet. The considerable success of our method in conjunction with Inception-BN can be attributed to the fact that, differently from AlexNet, this network is pre-trained with batch normalization, and thus initialized with weights that are already calibrated for normalized features. In appendix A we also provide DIAL – AlexNet results on Office 31, using the classical sampling protocol and show that our

### Table 2: Results on the Office-Caltech dataset using the full protocol.

| Method                  | Source | Target    | Amazon | Caltech | Webcam | DSLR | Caltech | Amazon | Webcam | DSLR | Average |
|-------------------------|--------|-----------|--------|---------|--------|------|---------|--------|--------|------|---------|
| AlexNet – source [15]   |         |           | 83.8   | 76.1    | 80.8   | 91.1 | 83.1    | 89.0   | 84.0   |      |         |
| DDC [29]                |         |           | 85.0   | 78.0    | 81.1   | 91.9 | 85.4    | 88.8   | 85.0   |      |         |
| DAN [18]                |         |           | 85.1   | 84.3    | 82.4   | 92.0 | 90.6    | 90.5   | 87.5   |      |         |
| RTN [19]                |         |           | 88.5   | **88.4**| 84.3   | **94.4**| **96.6**| **92.9**| 90.9   |      |         |
| RTN (no RES) [19]       |         |           | 88.0   | 87.3    | 82.4   | 93.5 | 96.3    | 91.4   | 89.8   |      |         |
| DIAL – AlexNet-No DA    |         |           | 83.4   | 75.3    | 66.7   | 92.8 | 82.0    | 83.6   | 80.6   |      |         |
| DIAL – AlexNet          | **89.3**|           | 87.8   | **88.2**| **94.5**| 96.6 | 88.8    | 90.9   |      |      |         |

### Table 3: Results on the Cross-Dataset Testbed using the experimental setup in [27]

| Method                  | Source | Target    | Caltech | Imagenet | Caltech |
|-------------------------|--------|-----------|---------|----------|---------|
| SDT [28]                |         |           | –       | 73.6     |         |
| Tommasi et al. [26]     |         |           | –       | 75.4 ± 0.6|        |
| Inception-BN source     | 82.1 ± 0.3|         | 88.4 ± 0.7|          |         |
| AdaBN [16]              | 82.2 ± 0.6|         | 87.3 ± 0.5|          |         |
| DIAL – Inception-BN     | 84.6 ± 0.7|         | 90.2 ± 0.3|          |         |

### Table 4: Results on the Cross-Dataset Testbed using the experimental setup in [24]

| Method                  | Source | Target    | Caltech | Imagenet | Caltech |
|-------------------------|--------|-----------|---------|----------|---------|
| SA [5]                  | 43.7   | 52.0      |         |          |         |
| GFK [8]                 | 52.0   | 58.5      |         |          |         |
| TCA [22]                | 48.6   | 54.0      |         |          |         |
| CORAL [24]              | 66.2   | 74.7      |         |          |         |
| Inception-BN – source   | 82.1 ± 0.7|         | 88.4 ± 0.8|          |         |
| AdaBN [16]              | 81.9 ± 0.6|         | 86.5 ± 0.8|          |         |
| DIAL – Inception-BN     | 84.6 ± 1.0|         | 89.7 ± 0.4|          |         |

4.2. Results

#### Comparison with State-of-the-art Methods

In our first series of experiments, summarized in Table 1, we compare our approach, applied to both AlexNet and Inception-BN, with several state-of-the-art methods on the Office-31 dataset. In particular, we consider: two shallow methods, i.e. Geodesic Flow Kernel (GFK) [8] and Subspace Alignment (SA) [5]; several deep methods based on AlexNet-like architectures, i.e. Deep Adaptation Networks (DAN) [18], Deep Domain Confusion (DDC) [29], the ReverseGrad network [6]; a recent deep method based on the Inception-BN architecture, i.e. AdaBN [16] with and without CORAL feature alignment [24]. We compare these baselines to the AlexNet and Inception-BN networks modified with our approach as explained in Section 4.1. In the table our approach is denoted as DIAL – AlexNet and DIAL – Inception-BN. As a reference, we further report the results obtained considering standard AlexNet and Inception-BN networks trained only on source data, and with AlexNet trained using the proposed approach but without domain-alignment layers (DIAL – AlexNet-No DA).

Unsurprisingly, the classification accuracy of the two methods based on hand-crafted features is greatly inferior to all other approaches. Among the deep methods based on the AlexNet architecture, DIAL – AlexNet shows the best average performance. On the other hand, the results obtained with DIAL – AlexNet-No DA are the lowest among the AlexNet-derived architectures, highlighting the effectiveness of the proposed domain-alignment technique. Among the methods based on Inception-BN, our approach considerably outperforms the others, with an average accuracy of five points higher than the second best, and improvements on the single experiments as high as ten points. It is interesting to note that the relative increase in accuracy from the source-only Inception-BN to DIAL – Inception-BN is higher than that from the source only AlexNet to DIAL – AlexNet. The considerable success of our method in conjunction with Inception-BN can be attributed to the fact that, differently from AlexNet, this network is pre-trained with batch normalization, and thus initialized with weights that are already calibrated for normalized features. In appendix A we also provide DIAL – AlexNet results on Office 31, using the classical sampling protocol and show that our
We also report results obtained with the recent Residual Network (ResNet). In our second set of experiments, we analyze the performance of several approaches on the Office-Caltech dataset. The results are reported in Table 2. We restrict our attention to methods based on deep architectures and, for a fair comparison, we consider all AlexNet-based approaches. In addition to the methods used in the Office31 experiments, we also report results obtained with the recent Residual Transfer Network (RTN) in [19]. From the table, it is evident that the performance of the proposed method and RTN are roughly the same (average accuracy 90.7% vs 90.9%) and they significantly outperform all the other baselines. A closer look at the table reveals the advantages of the proposed domain-alignment technique. Specifically, our approach mostly differs from the method in [19] in the fact that RTN adopts MMD to reduce the domain shift and introduces a residual function, and thus a considerable number of additional parameters, to model the difference between the source and the target classifiers. Therefore, by analyzing the accuracy of RTN without the residual term, RTN (noRES) - we can directly compare two different domain-alignment strategies: our domain-aware normalization and MMD minimization. The results in Table 2 clearly show the advantages of our contribution. We would also like to point out that, differently from MMD-based approaches [18, 19, 29], our DA-layers add no hand-tuned meta-parameters to the network. This is a key advantage over previous works, as it is well-known that parameter selection is a major problem in unsupervised domain adaptation [7]. Furthermore, as can be seen in appendix B, our DIAL – Inception-BN results strongly improve on the current SOTA.

Finally, we perform some experiments on the Caltech-ImageNet subset of the Cross-Dataset Testbed of [27]. As explained above, to facilitate comparison with previous works, we perform experiments in two different settings. Table 3 and Table 4 show our results. The proposed approach significantly outperforms previous methods and sets the new state of the art on this dataset. The higher performance of our method is not only due to the use of Inception-BN but also due to the effectiveness of our contribution. In appendix C, we also successfully evaluate a DIAL – ResNet on the same task. Indeed, the unique combination of the proposed regularization term (see Equation (5)) with our DA-layers makes our approach more effective than previous adaptation techniques based on Inception-BN (i.e. AdaBN [16]).

4.3. Analysis of the Proposed Method

DIAL only introduces one additional meta-parameter: the Lagrange multiplier $\lambda$ in [5], i.e. the weight assigned to the target loss term $L_t(\theta)$. Figure 3 shows the accuracy we obtain on the Office-Caltech dataset when training DIAL – AlexNet with different settings of $\lambda$. Note that when $\lambda = 0$ the network ignores the target samples at training time, updating its weights on the basis of the source data only. In most settings accuracy increases for increasing values of $\lambda$, with an asymptotic behavior for $\lambda > 1$. Not showed in this plot, we also experimented with bigger values of $\lambda$, observing however progressively higher training instability and variance in the final accuracy.

To further shed some light on the proposed domain-alignment approach, as in previous works [18, 19], we show some qualitative results considering t-SNE [20] to compute a low-dimensional embedding of the source and target features learned by our networks. We visualize them in Figure 4. In particular, we focus on the $W \rightarrow C$ task of the Office-Caltech dataset and compare the feature transformation described in Section 3.1. In contrast, some previous works [18, 19] proposed
to explicitly minimize the MMD between source and target distributions. In our last series of experiments we show how our approach is able to implicitly reduce the MMD, without introducing additional loss terms. Specifically, we evaluate the empirical MMD between source and target features at various levels of DIAL – AlexNet and DIAL – Inception-BN, when considering, respectively, the A→W task of Office-31 and the C→I task of the Cross-Dataset Testbed. The results, reported in Figure 5, show a consistent decrease in MMD when considering the features before and after our DA-layers. Note that the MMD values are only comparable within a layer, and not across layers.

5. Conclusions

We presented DIAL, a novel framework for unsupervised, deep domain adaptation. The core of our contribution is the introduction of novel, domain-alignment layers, which reduce domain shift by matching source and target distributions to a canonical one. Our framework also exploits unlabeled target data by introducing a prior distribution on the network parameters, which promotes classification models with high confidence on unlabeled samples. We evaluated the proposed approach devising a simple implementation of our DA-layers based on batch normalization. The results of our experiments demonstrate that our approach outperforms state-of-the-art domain-adaptation methods.

While this paper focuses on the challenging problem of unsupervised domain-adaptation and considers a single source/single task scenario, our approach can be trivially extended to a semi-supervised setting and to multiple domains. Future works will investigate this research direction.

Appendix

A. Office 31 with Sampling Protocol [23]

In this section we report the results obtained by DIAL – AlexNet on the Office31 dataset adopting the sampling protocol [23]. Similarly to the the full protocol experiments, we modify AlexNet by appending a DA-layer to each fully-connected layer. Differently from the original AlexNet, we do not perform dropout on the outputs of fc6 and fc7. We initialize the network parameters from a publicly-available model trained on the ILSVRC-2012 data, we freeze all the convolutional layers, and increase the learning rate of fc8 by a factor of 10. During training, each mini-batch contains a number of source and target samples proportional to the size of the corresponding dataset, while the batch size remains fixed at 256. We train for a total of 35 epochs (where “epoch” refers to a complete pass over the source set), reducing the learning rate by a factor 10 after 2/3 of training.

As shown in Table 5 and consistently with our results using the full protocol and described in the paper, our method outperforms all the baseline approaches. In particular, it achieves higher accuracy than RTN [19] with no residual layers and it is comparable, on average, with RTN.

B. Office-Caltech with DIAL – Inception-BN

In this section we evaluate the DIAL – Inception-BN architecture on the Office-Caltech dataset. We initialize the networks parameters from a publicly-available model trained on the ILSVRC-2012 data and freeze the first three inception blocks. Each batch is composed of 32 source images and 16 target images. As with the Office-31 experiments, we train for 20 epochs, reducing the learning rate by a factor 10 every 33% of the total number of iterations.

As shown in Table 6, the DIAL – Inception-BN architecture proves to be superior to all previous methods and
achieves state-of-the-art results on this dataset.

C. Cross-Dataset Testbed with DIAL – ResNet

In this section we apply DIAL to an additional very deep network architecture: ResNet-50 [11]. Similarly to DIAL – Inception-BN, we replace each batch normalization layer in ResNet-50 with a DA-layer, initialize the parameters from a publicly-available model trained on the ILSVRC-2012 data and freeze the first three network blocks. Training is performed for a total of 25 epochs, decreasing the learning rate by a factor 10 every 10 epochs. Each batch is composed of 16 source and 16 target images. 

Table 7 reports the results we obtain on the Caltech-ImageNet subset of the Cross-Dataset Testbed, using the experimental setup of [24]. Here, DIAL – ResNet denotes our ResNet-50 network with DA-layers, while ResNet – source denotes a ResNet-50 network trained on source data only. As for Inception-BN, we observe a notable increase in performance from ResNet – source to DIAL – ResNet, giving further support to the generality of our approach. In the C→I setting, DIAL – Inception-BN achieves slightly higher accuracy than DIAL – ResNet. This is somehow surprising, as many recent works have found the ResNet model more effective in image recognition tasks.

References

[1] S. Ben-David, T. Lu, T. Luu, and D. Pl. Impossibility theorems for domain adaptation. In AISTATS, 2010.
[2] W.-S. Chu, F. De la Torre, and J. F. Cohn. Selective transfer machine for personalized facial action unit detection. In CVPR, 2013.
[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009.
[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In ICML, 2014.
[5] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In ICCV, 2013.
[6] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. ICML, 2015.
[7] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In ICML, 2013.
[8] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In CVPR, 2012.
[9] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In NIPS, 2004.
[10] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.
[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.
[12] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola. Correcting sample selection bias by unlabeled data. In NIPS, 2006.
[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.
[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.
[16] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. Revisiting batch normalization for practical domain adaptation. arXiv preprint arXiv:1603.04779, 2016.
[17] M. Long, G. Ding, J. Wang, J. Sun, Y. Guo, and P. S. Yu. Transfer sparse coding for robust image representation. In CVPR, 2013.
[18] M. Long and J. Wang. Learning transferable features with deep adaptation networks. In ICML, 2015.
### Table 6: Results on the Office-Caltech dataset using the full protocol.

| Method           | Source Target | Amazon Caltech | Webcam Caltech | DSLR Caltech | Caltech Amazon | Caltech Webcam | Caltech DSLR | Average |
|------------------|---------------|----------------|----------------|--------------|----------------|----------------|--------------|---------|
| AlexNet – source | Caltech       | 83.8           | 76.1           | 80.8         | 91.1           | 83.1           | 89.0         | 84.0    |
| DDC [29]         | Caltech       | 85.0           | 78.0           | 81.1         | 91.9           | 85.4           | 88.8         | 85.0    |
| DAN [18]         | Caltech       | 85.1           | 84.3           | 82.4         | 92.0           | 90.6           | 90.5         | 87.5    |
| RTN [19]         | Caltech       | 88.5           | 88.4           | 84.3         | 94.4           | 96.6           | 92.9         | 90.9    |
| RTN (no RES) [19]| Caltech       | 88.0           | 87.3           | 82.4         | 93.5           | 96.3           | 91.4         | 89.8    |
| DIAL – AlexNet-No DA | Caltech | 83.4           | 75.3           | 66.7         | 92.8           | 82.0           | 83.6         | 80.6    |
| DIAL – AlexNet   | Caltech       | 89.3           | 87.8           | 88.2         | 94.5           | 96.6           | 88.8         | 90.9    |
| DIAL – Inception-BN | Caltech       | **93.2**       | **90.5**       | **90.0**     | **95.1**       | **98.5**       | **96.9**     | **94.0** |

### Table 7: Results on the Cross-Dataset Testbed using the experimental setup in [24]

| Method           | Source Target | CaltechImagenet | Imagenet Caltech |
|------------------|---------------|-----------------|-----------------|
| SA [5]           | Caltech       | 43.7            | 52.0            |
| GFK [8]          | Caltech       | 52.0            | 58.5            |
| TCA [22]         | Caltech       | 48.6            | 54.0            |
| CORAL [24]       | Caltech       | 66.2            | 74.7            |
| Inception-BN – source | Caltech | 82.1 ± 0.7     | 88.4 ± 0.8     |
| AdaBN [10]       | Caltech       | 81.9 ± 0.6     | 86.5 ± 0.8     |
| DIAL – Inception-BN | Caltech       | **84.6 ± 1.0** | 89.7 ± 0.4     |
| ResNet – source  | Caltech       | 78.9 ± 0.7     | 88.2 ± 0.8     |
| DIAL – ResNet    | Caltech       | 83.8 ± 1.0     | **91.1 ± 0.6** |

[19] M. Long, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. *NIPS*, 2016.

[20] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[21] T. J. O’Neill. Normal discrimination with unclassified observations. *Journal of the American Statistical Association*, 73(364):821–826, 1978.

[22] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[23] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[24] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

[25] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. *arXiv preprint arXiv:1607.01719*, 2016.

[26] T. Tommasi, M. Lanzì, P. Russo, and B. Caputo. Learning the roots of visual domain shift. *arXiv preprint arXiv:1607.06144*, 2016.

[27] T. Tommasi and T. Tuytelaars. A testbed for cross-dataset analysis. In *ECCV*, 2014.

[28] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015.

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

[30] M. Yamada, L. Sigal, and M. Raptis. No bias left behind: Covariate shift adaptation for discriminative 3d pose estimation. In *ECCV*, 2012.

[31] X. Zeng, W. Ouyang, M. Wang, and X. Wang. Deep learning of scene-specific classifier for pedestrian detection. In *ECCV*, 2014.