

Simultaneous Deep Transfer Across Domains and Tasks

Eric Tzeng*, Judy Hoffman*, Trevor Darrell
UC Berkeley, EECS & ICSI
{etzeng, jhoffman, trevor}@eecs.berkeley.edu

Kate Saenko
UMass Lowell, CS
saenko@cs.uml.edu

Abstract

Recent reports suggest that a generic supervised deep CNN model trained on a large-scale dataset reduces, but does not remove, dataset bias. Fine-tuning deep models in a new domain can require a significant amount of labeled data, which for many applications is simply not available. We propose a new CNN architecture to exploit unlabeled and sparsely labeled target domain data. Our approach simultaneously optimizes for domain invariance to facilitate domain transfer and uses a soft label distribution matching loss to transfer information between tasks. Our proposed adaptation method offers empirical performance which exceeds previously published results on two standard benchmark visual domain adaptation tasks, evaluated across supervised and semi-supervised adaptation settings.

1. Introduction

Consider a group of robots trained by the manufacturer to recognize thousands of common objects using standard image databases, then shipped to households around the country. As each robot starts to operate in its own unique environment, it is likely to have degraded performance due to the shift in domain. It is clear that, given enough extra supervised data from the new environment, the original performance could be recovered. However, state-of-the-art recognition algorithms rely on high capacity convolutional neural network (CNN) models that require millions of supervised images for initial training. Even the traditional approach for adapting deep models, fine-tuning [14, 29], may require hundreds or thousands of labeled examples for each object category that needs to be adapted.

It is reasonable to assume that the robot’s new owner will label a handful of examples for a few types of objects, but completely unrealistic to presume full supervision in the new environment. Therefore, we propose an algorithm that effectively adapts between the training (source) and test (target) environments by utilizing both generic statistics from

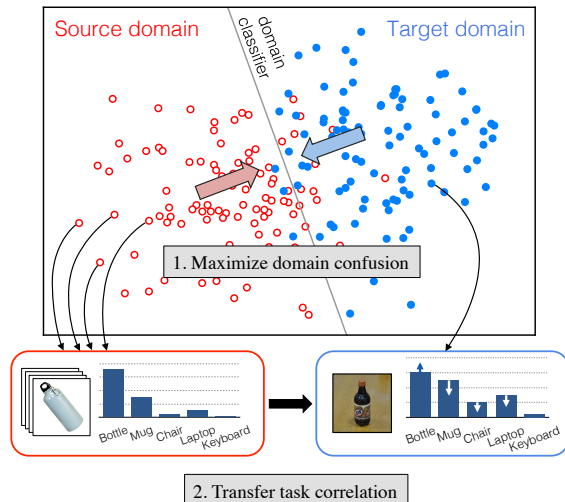


Figure 1. We transfer discriminative category information from a source domain to a target domain via two methods. First, we maximize domain confusion by making the marginal distributions of the two domains as similar as possible. Second, we transfer correlations between classes learned on the source examples directly to the target examples, thereby preserving the relationships between classes.

unlabeled data collected in the new environment as well as a few human labeled examples from a subset of the categories of interest. Our approach performs transfer learning both across domains and across tasks (see Figure 1). Intuitively, domain transfer is accomplished by making the marginal feature distributions of source and target as similar to each other as possible. Task transfer is enabled by transferring empirical category correlations learned on the source to the target domain. This helps to preserve relationships between categories, e.g., *bottle* is similar to *mug* but different from *keyboard*. Previous work proposed techniques for domain transfer with CNN models [12, 24] but did not utilize the learned source semantic structure for task transfer.

To enable domain transfer, we use the unlabeled target data to compute an estimated marginal distribution over the new environment and explicitly optimize a feature repre-

* Authors contributed equally.

sensation that minimizes the distance between the source and target domain distributions. Dataset bias was classically illustrated in computer vision by the “name the dataset” game of Torralba and Efros [31], which trained a classifier to predict which dataset an image originates from, thereby showing that visual datasets are biased samples of the visual world. Indeed, this turns out to be formally connected to measures of domain discrepancy [21, 5]. Optimizing for domain invariance, therefore, can be considered equivalent to the task of learning to predict the class labels while simultaneously finding a representation that makes the domains appear as similar as possible. This principle forms the domain transfer component of our proposed approach. We learn deep representations by optimizing over a loss which includes both classification error on the labeled data as well as a *domain confusion* loss which seeks to make the domains indistinguishable.

However, while maximizing domain confusion pulls the marginal distributions of the domains together, it does not necessarily align the classes in the target with those in the source. Thus, we also explicitly transfer the similarity structure amongst categories from the source to the target and further optimize our representation to produce the same structure in the target domain using the few target labeled examples as reference points. We are inspired by prior work on distilling deep models [3, 16] and extend the ideas presented in these works to a domain adaptation setting. We first compute the average output probability distribution, or “soft label,” over the source training examples in each category. Then, for each target labeled example, we directly optimize our model to match the distribution over classes to the soft label. In this way we are able to perform task adaptation by transferring information to categories with no explicit labels in the target domain.

We solve the two problems jointly using a new CNN architecture, outlined in Figure 2. We combine a domain confusion and softmax cross-entropy losses to train the network with the target data. Our architecture can be used to solve *supervised adaptation*, when a small amount of target labeled data is available from each category, and *semi-supervised adaptation*, when a small amount of target labeled data is available from a subset of the categories. We provide a comprehensive evaluation on the popular Office benchmark [28] and the recently introduced cross-dataset collection [30] for classification across visually distinct domains. We demonstrate that by jointly optimizing for domain confusion and matching soft labels, we are able to outperform the current state-of-the-art visual domain adaptation results.

2. Related work

There have been many approaches proposed in recent years to solve the visual domain adaptation problem, which is also commonly framed as the visual dataset bias prob-

lem [31]. All recognize that there is a shift in the distribution of the source and target data representations. In fact, the size of a domain shift is often measured by the distance between the source and target subspace representations [5, 11, 21, 25, 27]. A large number of methods have sought to overcome this difference by learning a feature space transformation to align the source and target representations [28, 23, 11, 15]. For the *supervised* adaptation scenario, when a limited amount of labeled data is available in the target domain, some approaches have been proposed to learn a target classifier regularized against the source classifier [32, 2, 1]. Others have sought to both learn a feature transformation and regularize a target classifier simultaneously [18, 10].

Recently, supervised CNN based feature representations have been shown to be extremely effective for a variety of visual recognition tasks [22, 9, 14, 29]. In particular, using deep representations dramatically reduces the effect of resolution and lighting on domain shifts [9, 19]. Parallel CNN architectures such as Siamese networks have been shown to be effective for learning invariant representations [6, 8]. However, training these networks requires labels for each training instance, so it is unclear how to extend these methods to unsupervised or semi-supervised settings. Multimodal deep learning architectures have also been explored to learn representations that are invariant to different input modalities [26]. However, this method operated primarily in a generative context and therefore did not leverage the full representational power of supervised CNN representations.

Training a joint source and target CNN architecture was proposed by [7], but was limited to two layers and so was significantly outperformed by the methods which used a deeper architecture [22], pre-trained on a large auxiliary data source (ex: ImageNet [4]). [13] proposed pre-training with a denoising auto-encoder, then training a two-layer network simultaneously with the MMD domain confusion loss. This effectively learns a domain invariant representation, but again, because the learned network is relatively shallow, it lacks the strong semantic representation that is learned by directly optimizing a classification objective with a supervised deep CNN.

Using classifier output distributions instead of category labels during training has been explored in the context of model compression or distillation [3, 16]. However, we are the first to apply this technique in a domain adaptation setting in order to transfer class correlations between domains.

Other works have contemporaneously explored the idea of directly optimizing a representation for domain invariance [12, 24]. However, they either use weaker measures of domain invariance or make use of optimization methods that are less robust than our proposed method, and they do not attempt to solve the task transfer problem in the semi-supervised setting.

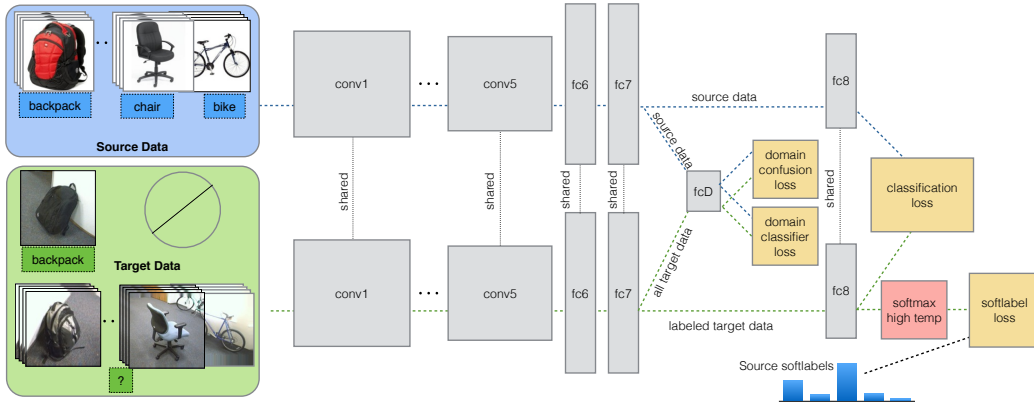


Figure 2. Our overall CNN architecture for domain and task transfer. We use a domain confusion loss over all source and target (both labeled and unlabeled) data to learn a domain invariant representation. We simultaneously transfer the learned source semantic structure to the target domain by optimizing the network to produce activation distributions that match those learned for source data in the source only CNN. *Best viewed in color.*

3. Joint CNN architecture for domain and task transfer

We first give an overview of our convolutional network (CNN) architecture, depicted in Figure 2, that learns a representation which both aligns visual domains and transfers the semantic structure from a well labeled source domain to the sparsely labeled target domain. We assume access to a limited amount of labeled target data, potentially from only a subset of the categories of interest. With limited labels on a subset of the categories, the traditional domain transfer approach of fine-tuning on the available target data [14, 29, 17] is not effective. Instead, since the source labeled data shares the label space of our target domain, we use the source data to guide training of the corresponding classifiers.

Our method takes as input the labeled source data $\{x_S, y_S\}$ (blue box Figure 2) and the target data $\{x_T, y_T\}$ (green box Figure 2), where the labels y_T are only provided for a subset of the target examples. Our goal is to produce a category classifier θ_C that operates on an image feature representation $f(x; \theta_{\text{repr}})$ parameterized by representation parameters θ_{repr} and can correctly classify target examples at test time.

For a setting with K categories, let our desired classification objective be defined as the standard softmax loss

$$\mathcal{L}_C(x, y; \theta_{\text{repr}}, \theta_C) = - \sum_k \mathbb{1}[y = k] \log p_k \quad (1)$$

where p is the softmax of the classifier activations, $p = \text{softmax}(\theta_C^T f(x; \theta_{\text{repr}}))$.

We could use the available source labeled data to train our representation and classifier parameters according to Equation (1), but this often leads to overfitting to the source distribution, causing reduced performance at test time when recognizing in the target domain. However, we note that

if the source and target domains are very similar then the classifier trained on the source will perform well on the target. In fact, it is sufficient for the source and target data to be similar under the learned representation, θ_{repr} .

Inspired by the “name the dataset” game of Torralba and Efros [31], we can directly train a domain classifier θ_D to identify whether a training example originates from the source or target domain given its feature representation. Intuitively, if our choice of representation suffers from domain shift, then they will lie in distinct parts of the feature space, and a classifier will be able to easily separate the domains. We use this notion to add a new *domain confusion* loss $\mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}})$ to our objective and directly optimize our representation so as to minimize the discrepancy between the source and target distributions. This loss is described in more detail in Section 3.1.

Domain confusion can be applied to learn a representation that aligns source and target data without any target labeled data. However, we also presume a handful of sparse labels in the target domain, y_T . In this setting, a simple approach is to incorporate the target labeled data along with the source labeled data into the classification objective of Equation (1)¹. However, fine-tuning with hard category labels limits the impact of a single training example, making it hard for the network to learn to generalize from the limited labeled data. Additionally, fine-tuning with hard labels is ineffective when labeled data is available for only a subset of the categories.

For our approach, we draw inspiration from recent network distillation works [3, 16], which demonstrate that a large network can be “distilled” into a simpler model by replacing the hard labels with the softmax activations from the original large model. This modification proves to be critical, as the distribution holds key information about the relation-

¹We present this approach as one of our baselines.

ships between categories and imposes additional structure during the training process. In essence, because each training example is paired with an output distribution, it provides valuable information about not only the category it belongs to, but also each other category the classifier is trained to recognize.

Thus, we propose using the labeled target data to optimize the network parameters through a *soft label* loss, $\mathcal{L}_{\text{soft}}(x_T, y_T; \theta_{\text{repr}}, \theta_C)$. This loss will train the network parameters to produce a “soft label” activation that matches the average output distribution of source examples on a network trained to classify source data. This loss is described in more detail in Section 3.2. By training the network to match the expected source output distributions on target data, we transfer the learned inter-class correlations from the source domain to examples in the target domain. This directly transfers useful information from source to target, such as the fact that *bookshelves* appear more similar to *filing cabinets* than to *bicycles*.

Our full method then minimizes the joint loss function

$$\begin{aligned} \mathcal{L}(x_S, y_S, x_T, y_T, \theta_D; \theta_{\text{repr}}, \theta_C) = & \\ & \mathcal{L}_C(x_S, y_S, x_T, y_T; \theta_{\text{repr}}, \theta_C) \\ & + \lambda \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}) \\ & + \nu \mathcal{L}_{\text{soft}}(x_T, y_T; \theta_{\text{repr}}, \theta_C). \end{aligned} \quad (2)$$

where the hyperparameters λ and ν determine how strongly domain confusion and soft labels influence the optimization.

Our ideas of domain confusion and soft label loss for task transfer are generic and can be applied to any CNN classification architecture. For our experiments and for the detailed discussion in this paper we modify the standard Krizhevsky architecture [22], which has five convolutional layers (conv1–conv5) and three fully connected layers (fc6–fc8). The representation parameter θ_{repr} corresponds to layers 1–7 of the network, and the classification parameter θ_C corresponds to layer 8. For the remainder of this section, we provide further details on our novel loss definitions and the implementation of our model.

3.1. Aligning domains via domain confusion

In this section we describe in detail our proposed *domain confusion* loss objective. Recall that we introduce the domain confusion loss as a means to learn a representation that is domain invariant, and thus will allow us to better utilize a classifier trained using the labeled source data. We consider a representation to be domain invariant if a classifier trained using that representation can not distinguish examples from the two domains.

To this end, we add an additional domain classification layer, denoted as fcD in Figure 2, with parameters θ_D . This layer simply performs binary classification using the domain corresponding to an image as its label. For a particular feature representation, θ_{repr} , we evaluate its domain invariance

by learning the best domain classifier on the representation. This can be learned by optimizing the following objective, where y_D denotes the domain that the example is drawn from:

$$\mathcal{L}_D(x_S, x_T, \theta_{\text{repr}}; \theta_D) = - \sum_d \mathbb{1}[y_D = d] \log q_d \quad (3)$$

with q corresponding to the softmax of the domain classifier activation: $q = \text{softmax}(\theta_D^T f(x; \theta_{\text{repr}}))$.

For a particular domain classifier, θ_D , we can now introduce our loss which seeks to “maximally confuse” the two domains by computing the cross entropy between the output predicted domain labels and a uniform distribution over domain labels:

$$\mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}) = - \sum_d \frac{1}{D} \log q_d. \quad (4)$$

This domain confusion loss seeks to learn domain invariance by finding a representation in which the best domain classifier performs poorly.

Ideally, we want to simultaneously minimize Equations (3) and (4) for the representation and the domain classifier parameters. However, the two losses stand in direct opposition to one another: learning a fully domain invariant representation means the domain classifier must do poorly, and learning an effective domain classifier means that the representation is not domain invariant. Rather than globally optimizing θ_D and θ_{repr} , we instead perform iterative updates for the following two objectives given the fixed parameters from the previous iteration:

$$\min_{\theta_D} \mathcal{L}_D(x_S, x_T, \theta_{\text{repr}}; \theta_D) \quad (5)$$

$$\min_{\theta_{\text{repr}}} \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}). \quad (6)$$

These losses are readily implemented in standard deep learning frameworks, and after setting learning rates properly so that Equation (5) only updates θ_D and Equation (6) only updates θ_{repr} , the updates can be performed via standard backpropagation. Together, these updates ensure that we learn a representation that is domain invariant.

3.2. Aligning source and target classes via soft labels

While training the network to confuse the domains acts to align their marginal distributions, there are no guarantees about the alignment of classes between each domain. To ensure that the relationships between classes are preserved across source and target, we fine-tune the network against “soft labels” rather than the image category hard label.

We define a soft label for category k as the average over the softmax of all activations of source examples in category k , depicted graphically in Figure 3, and denote this average as $l^{(k)}$. Note that, since the source network was trained

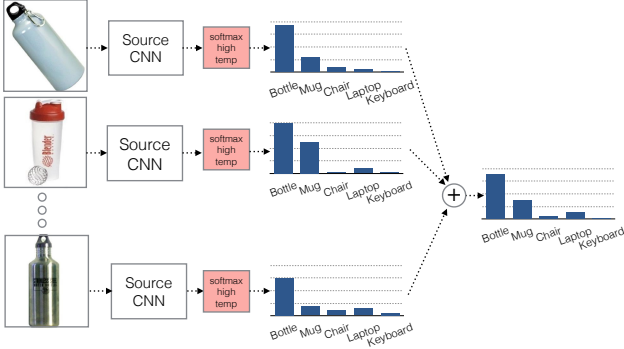


Figure 3. Soft label distributions are learned by averaging the per-category activations of source training examples using the source model. An example, with 5 categories, depicted here to demonstrate the final soft activation for the bottle category will be primarily dominated by bottle and mug with very little mass on chair, laptop, and keyboard.

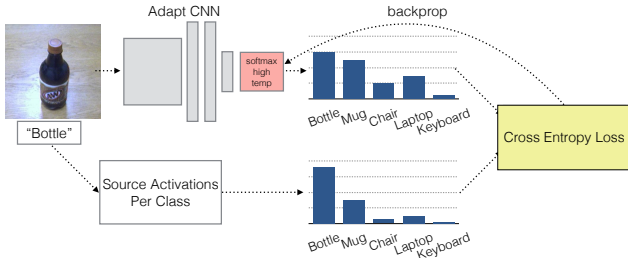


Figure 4. Depiction of the use of source per-category soft activations with the cross entropy loss function over the current target activations.

purely to optimize a classification objective, a simple softmax over each z_S^i will hide much of the useful information by producing a very peaked distribution. Instead, we use a softmax with a high temperature τ so that the related classes have enough probability mass to have an effect during fine-tuning. With our computed per-category soft labels we can now define our soft label loss:

$$\mathcal{L}_{\text{soft}}(x_T, y_T; \theta_{\text{repr}}, \theta_C) = - \sum_i l_i^{(y_T)} \log p_i \quad (7)$$

where p denotes the soft activation of the target image, $p = \text{softmax}(\theta_C^T f(x_T; \theta_{\text{repr}}) / \tau)$. The loss above corresponds to the cross-entropy loss between the soft activation of a particular target image and the soft label corresponding to the category of that image, as shown in Figure 4.

To see why this will help, consider the soft label for a particular category, such as *bottle*. The soft label $l^{(\text{bottle})}$ is a K -dimensional vector, where each dimension indicates the similarity of bottles to each of the K categories. In this example, the bottle soft label will have a higher weight on *mug* than on *keyboard*, since bottles and mugs are more visually similar. Thus, soft label training with this particular soft label directly enforces the relationship that bottles and

mugs should be closer in feature space than bottles and keyboards.

One important benefit of using this soft label loss is that we ensure that the parameters for categories without any labeled target data are still updated to output non-zero probabilities. We explore this benefit in Section 4, where we train a network using labels from a subset of the target categories and find significant performance improvement even when evaluating only on the unlabeled categories.

4. Evaluation

To analyze the effectiveness of our method, we evaluate it on the Office dataset, a standard benchmark dataset for visual domain adaptation, and on a new large-scale cross-dataset domain adaptation challenge.

4.1. Adaptation on the Office dataset

The Office dataset is a collection of images from three distinct domains, Amazon, DSLR, and Webcam, the largest of which has 2817 labeled images [28]. The 31 categories in the dataset consist of objects commonly encountered in office settings, such as keyboards, file cabinets, and laptops.

We evaluate our method in two different settings:

- **Supervised adaptation** Labeled training data for all categories is available in source and sparsely in target.
- **Semi-supervised adaptation (task adaptation)** Labeled training data is available in source and sparsely for a subset of the target categories.

For all experiments we initialize the parameters of conv1-fc7 using the released CaffeNet [20] weights. We then further fine-tune the network using the source labeled data in order to produce the soft label distributions and use the learned source CNN weights as the initial parameters for training our method. All implementations are produced using the open source Caffe [20] framework, and the network definition files and cross entropy loss layer needed for training will be released upon acceptance. We optimize the network using a learning rate of 0.001 and set the hyper-parameters to $\lambda = 0.01$ (confusion) and $\nu = 0.1$ (soft).

For each of the six domain shifts, we evaluate across five train/test splits, which are generated by sampling examples from the full set of images per domain. In the source domain, we follow the standard protocol for this dataset and generate splits by sampling 20 examples per category for the Amazon domain, and 8 examples per category for the DSLR and Webcam domains.

We first present results for the supervised setting, where 3 labeled examples are provided for each category in the target domain. We report accuracies on the remaining unlabeled images, following the standard protocol introduced with the dataset [28]. In addition to a variety of baselines, we

	$A \rightarrow W$	$A \rightarrow D$	$W \rightarrow A$	$W \rightarrow D$	$D \rightarrow A$	$D \rightarrow W$	Average
DLID [7]	51.9	-	-	89.9	-	78.2	-
DeCAF ₆ S+T [9]	80.7 ± 2.3	-	-	-	-	94.8 ± 1.2	-
DaNN [13]	53.6 ± 0.2	-	-	83.5 ± 0.0	-	71.2 ± 0.0	-
Source CNN	56.5 ± 0.3	64.6 ± 0.4	42.7 ± 0.1	93.6 ± 0.2	47.6 ± 0.1	92.4 ± 0.3	66.22
Target CNN	80.5 ± 0.5	81.8 ± 1.0	59.9 ± 0.3	81.8 ± 1.0	59.9 ± 0.3	80.5 ± 0.5	74.05
Source+Target CNN	82.5 ± 0.9	85.2 ± 1.1	65.2 ± 0.7	96.3 ± 0.5	<u>65.8 ± 0.5</u>	93.9 ± 0.5	81.50
Ours: dom confusion only	82.8 ± 0.9	<u>85.9 ± 1.1</u>	<u>64.9 ± 0.5</u>	97.5 ± 0.2	66.2 ± 0.4	<u>95.6 ± 0.4</u>	82.13
Ours: soft labels only	<u>82.7 ± 0.7</u>	84.9 ± 1.2	65.2 ± 0.6	98.3 ± 0.3	<u>66.0 ± 0.5</u>	95.9 ± 0.6	82.17
Ours: dom confusion+soft labels	<u>82.7 ± 0.8</u>	86.1 ± 1.2	<u>65.0 ± 0.5</u>	97.6 ± 0.2	66.2 ± 0.3	<u>95.7 ± 0.5</u>	82.22

Table 1. Multi-class accuracy evaluation on the standard supervised adaptation setting with the *Office* dataset. We evaluate on all 31 categories using the standard experimental protocol from [28]. Here, we compare against three state-of-the-art domain adaptation methods as well as a CNN trained using only source data, only target data, or both source and target data together.

	$A \rightarrow W$	$A \rightarrow D$	$W \rightarrow A$	$W \rightarrow D$	$D \rightarrow A$	$D \rightarrow W$	Average
MMDT [18]	-	44.6 ± 0.3	-	58.3 ± 0.5	-	-	-
Source CNN	54.2 ± 0.6	63.2 ± 0.4	34.7 ± 0.1	94.5 ± 0.2	36.4 ± 0.1	89.3 ± 0.5	62.0
Ours: dom confusion only	55.2 ± 0.6	63.7 ± 0.9	41.1 ± 0.0	96.5 ± 0.1	41.2 ± 0.1	91.3 ± 0.4	64.8
Ours: soft labels only	56.8 ± 0.4	65.2 ± 0.9	38.8 ± 0.4	96.5 ± 0.2	41.7 ± 0.3	89.6 ± 0.1	64.8
Ours: dom confusion+soft labels	59.3 ± 0.6	68.0 ± 0.5	40.5 ± 0.2	97.5 ± 0.1	43.1 ± 0.2	90.0 ± 0.2	66.4

Table 2. Multi-class accuracy evaluation on the standard semi-supervised adaptation setting with the *Office* dataset. We evaluate on 16 held-out categories for which we have no access to target labeled data. We show results on these unsupervised categories for the source only model, our model trained using only soft labels for the 15 auxiliary categories, and finally using domain confusion together with soft labels on the 15 auxiliary categories.

report numbers for both soft label fine-tuning alone as well as soft labels with domain confusion in Table 1. Because the Office dataset is imbalanced, we report multi-class accuracies, which are obtained by computing per-class accuracies independently, then averaging over all 31 categories.

We see that fine-tuning with soft labels or domain confusion provides a consistent improvement over hard label training in 5 of 6 shifts. Combining soft labels with domain confusion produces marginally higher performance on average. This result follows the intuitive notion that when enough target labeled examples are present, directly optimizing for the joint source and target classification objective (Source+Target CNN) is a strong baseline and so using either of our new losses adds enough regularization to improve performance.

Next, we experiment with the semi-supervised adaptation setting. We consider the case in which training data and labels are available for some, but not all of the categories in the target domain. We are interested in seeing whether we can transfer information learned from the labeled classes to the unlabeled classes.

To do this, we consider having 10 target labeled examples per category from only 15 of the 31 total categories, following the standard protocol introduced with the *Office* dataset [28]. We then evaluate our classification performance on the remaining 16 categories for which no data was avail-

able at training time.

In Table 2 we present multi-class accuracies over the 16 held-out categories and compare our method to a previous domain adaptation method [18] as well as a source-only trained CNN. Note that, since the performance here is computed over only a subset of the categories in the dataset, the numbers in this table should not be directly compared to the supervised setting in Table 1.

We find that all variations of our method (only soft label loss, only domain confusion, and both together) outperform the baselines. Contrary to the fully supervised case, here we note that both domain confusion and soft labels contribute significantly to the overall performance improvement of our method. This stems from the fact that we are now evaluating on categories which lack labeled target data, and thus the network can not implicitly enforce domain invariance through the classification objective alone. Separately, the fact that we get improvement from the soft label training on related tasks indicates that information is being effectively transferred between tasks.

In Figure 5, we show examples for the Amazon→Webcam shift where our method correctly classifies images from held out object categories and the baseline does not. We find that our method is able to consistently overcome error cases, such as the notebooks that were previously confused with letter trays, or the black



Figure 5. Examples from the Amazon→Webcam shift in the semi-supervised adaptation setting, where our method (the bottom turquoise label) correctly classifies images while the baseline (the top purple label) does not.

mugs that were confused with black computer mice.

4.2. Adaptation between diverse domains

For an evaluation with larger, more distinct domains, we test on the recent testbed for cross-dataset analysis [30], which collects images from classes shared in common among computer vision datasets. We use the dense version of this testbed, which consists of 40 categories shared between the ImageNet, Caltech-256, SUN, and Bing datasets, and evaluate specifically with ImageNet as source and Caltech-256 as target.

We follow the protocol outlined in [30] and generate 5 splits by selecting 5534 images from ImageNet and 4366 images from Caltech-256 across the 40 shared categories. Each split is then equally divided into a train and test set. However, since we are most interested in evaluating in the setting with limited target data, we further subsample the target training set into smaller sets with only 1, 3, and 5 labeled examples per category.

Results from this evaluation are shown in Figure 6. We compare our method to both CNNs fine-tuned using only source data using source and target labeled data. Contrary to the previous supervised adaptation experiment, our method significantly outperforms both baselines. We see that our full architecture, combining domain confusion with the soft label loss, performs the best overall and is able to operate in the regime of no labeled examples in the target (corresponding to the red line at point 0 on the x -axis). We find that the most benefit of our method arises when there are few labeled training examples per category in the target domain. As we increase the number of labeled examples in the target, the standard fine-tuning strategy begins to approach the performance of the adaptation approach. This

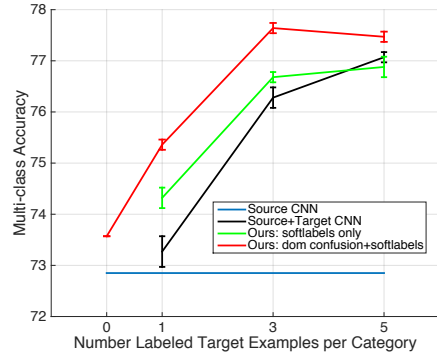


Figure 6. ImageNet→Caltech supervised adaptation from the Cross-dataset [30] testbed with varying numbers of labeled target examples per category. We find that our method using soft label loss (with and without domain confusion) outperforms the baselines of training on source data alone or using a standard fine-tuning strategy to train with the source and target data. *Best viewed in color.*

indicates that direct joint source and target fine-tuning is a viable adaptation approach when you have a reasonable number of training examples per category. In comparison, fine-tuning on the target examples alone yields accuracies of 36.6 ± 0.6 , 60.9 ± 0.5 , and 67.7 ± 0.5 for the cases of 1, 3, and 5 labeled examples per category, respectively. All of these numbers underperform the source only model, indicating that adaptation is crucial in the setting of limited training data.

Finally, we note that our results are significantly higher than the 24.8% result reported in [30], despite the use of much less training data. This difference is explained by their use of SURF BoW features, indicating that CNN features are a much stronger feature for use in adaptation tasks.

5. Analysis

Our experimental results demonstrate that our method improves classification performance in a variety of domain adaptation settings. We now perform additional analysis on our method by confirming our claims that it exhibits domain invariance and transfers information across tasks.

5.1. Domain confusion enforces domain invariance

We begin by evaluating the effectiveness of domain confusion at learning a domain invariant representation. As previously explained, we consider a representation to be domain invariant if an optimal classifier has difficulty predicting which domain an image originates from. Thus, for our representation learned with a domain confusion loss, we expect a trained domain classifier to perform poorly.

We train two support vector machines (SVMs) to classify images into domains: one using the baseline CaffeNet fc7 representation, and the other using our fc7 learned with

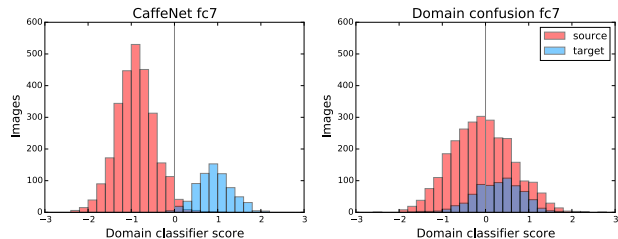


Figure 7. We compare the baseline CaffeNet representation to our representation learned with domain confusion by training a support vector machine to predict the domains of Amazon and Webcam images. For each representation, we plot a histogram of the classifier decision scores of the test images. In the baseline representation, the classifier is able to separate the two domains with 99% accuracy. In contrast, the representation learned with domain confusion is domain invariant, and the classifier can do no better than 56%.

domain confusion. These SVMs are trained using 160 images, 80 from Amazon and 80 from Webcam, then tested on the remaining images from those domains. We plot the classifier scores for each test image in Figure 7. It is obvious that the domain confusion representation is domain invariant, making it much harder to separate the two domains—the test accuracy on the domain confusion representation is only 56%, not much better than random. In contrast, on the baseline CaffeNet representation, the domain classifier achieves 99% test accuracy.

5.2. Soft labels for task transfer

We now examine the effect of soft labels in transferring information between categories. We consider the Amazon→Webcam shift from the semi-supervised adaptation experiment in the previous section. Recall that in this setting, we have access to target labeled data for only half of our categories. We use soft label information from the source domain to provide information about the held-out categories which lack labeled target examples. Figure 8 examines one target example from the held-out category *monitor*. No labeled target monitors were available during training; however, as shown in the upper right corner of Figure 8, the soft labels for *laptop computer* was present during training and assigns a relatively high weight to the *monitor* class. Soft label fine-tuning thus allows us to exploit the fact that these categories are similar. We see that the baseline model misclassifies this image as a *ring binder*, while our soft label model correctly assigns the *monitor* label.

6. Conclusion

We have presented a CNN architecture that effectively adapts to a new domain with limited or no labeled data per target category. We accomplish this through a novel CNN architecture which simultaneously optimizes for domain in-

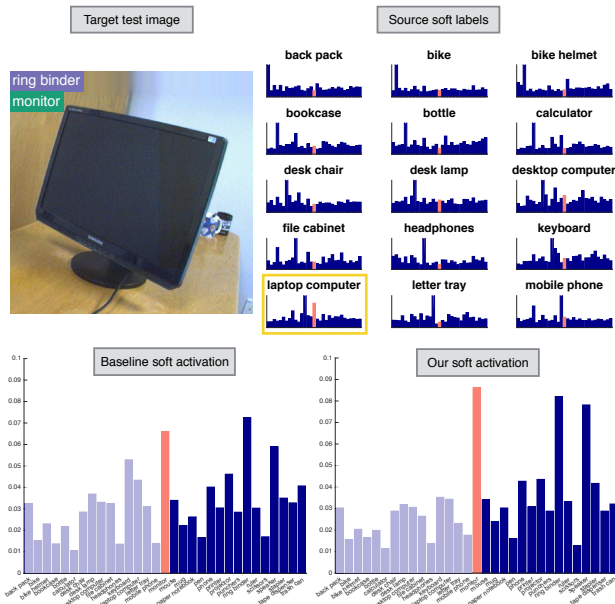


Figure 8. Our method (bottom turquoise label) correctly predicts the category of this image, whereas the baseline (top purple label) does not. The source per-category soft labels for the 15 categories with labeled target data are shown in the upper right corner, where the x -axis of the plot represents the 31 categories and the y -axis is the output probability. We highlight the index corresponding to the *monitor* category in red. As no labeled target data is available for the correct category, *monitor*, we find that in our method the related category of *laptop computer* (outlined with yellow box) transfers information to the *monitor* category. As a result, after training, our method places the highest weight on the correct category. Probability score per category for the baseline and our method are shown in the bottom left and right, respectively, training categories are opaque and correct test category is shown in red.

variance, to facilitate domain transfer, while transferring task information between domains in the form of a cross entropy soft label loss. We demonstrate the ability of our architecture to improve adaptation performance in the *supervised* and *semi-supervised* settings by experimenting with two standard domain adaptation benchmark datasets. In the semi-supervised adaptation setting, we see an average relative improvement of 13% over the baselines on the four most challenging shifts in the Office dataset. Overall, our method can be easily implemented as an alternative fine-tuning strategy when limited or no labeled data is available per category in the target domain.

Acknowledgements This work was supported by DARPA; AFRL; DoD MURI award N000141110688; NSF awards 113629, IIS-1427425, and IIS-1212798; and the Berkeley Vision and Learning Center.

References

- [1] L. T. Alessandro Bergamo. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *Neural Information Processing Systems (NIPS)*, Dec. 2010. [2](#)
- [2] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *Proc. ICCV*, 2011. [2](#)
- [3] J. Ba and R. Caruana. Do deep nets really need to be deep? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2654–2662. Curran Associates, Inc., 2014. [2](#), [3](#)
- [4] A. Berg, J. Deng, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge 2012. 2012. [2](#)
- [5] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. In *Bioinformatics*, 2006. [2](#)
- [6] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993. [2](#)
- [7] S. Chopra, S. Balakrishnan, and R. Gopalan. DLID: Deep learning for domain adaptation by interpolating between domains. In *ICML Workshop on Challenges in Representation Learning*, 2013. [2](#), [6](#)
- [8] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005. [2](#)
- [9] 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 *Proc. ICML*, 2014. [2](#), [6](#)
- [10] L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for heterogeneous domain adaptation. In *Proc. ICML*, 2012. [2](#)
- [11] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proc. ICCV*, 2013. [2](#)
- [12] Y. Ganin and V. Lempitsky. Unsupervised Domain Adaptation by Backpropagation. *ArXiv e-prints*, Sept. 2014. [1](#), [2](#)
- [13] M. Ghifary, W. B. Kleijn, and M. Zhang. Domain adaptive neural networks for object recognition. *CoRR*, abs/1409.6041, 2014. [2](#), [6](#)
- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv e-prints*, 2013. [1](#), [2](#), [3](#)
- [15] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proc. CVPR*, 2012. [2](#)
- [16] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2014. [2](#), [3](#)
- [17] J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. LSDA: Large scale detection through adaptation. In *Neural Information Processing Systems (NIPS)*, 2014. [3](#)
- [18] J. Hoffman, E. Rodner, J. Donahue, K. Saenko, and T. Darrell. Efficient learning of domain-invariant image representations. In *Proc. ICLR*, 2013. [2](#), [6](#)
- [19] J. Hoffman, E. Tzeng, J. Donahue, , Y. Jia, K. Saenko, and T. Darrell. One-shot learning of supervised deep convolutional models. In *arXiv 1312.6204; presented at ICLR Workshop*, 2014. [2](#)
- [20] 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. [5](#)
- [21] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proc. VLDB*, 2004. [2](#)
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. [2](#), [4](#)
- [23] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proc. CVPR*, 2011. [2](#)
- [24] M. Long and J. Wang. Learning transferable features with deep adaptation networks. *CoRR*, abs/1502.02791, 2015. [1](#), [2](#)
- [25] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009. [2](#)
- [26] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 689–696, 2011. [2](#)
- [27] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *IJCA*, 2009. [2](#)
- [28] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proc. ECCV*, 2010. [2](#), [5](#), [6](#)
- [29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013. [1](#), [2](#), [3](#)
- [30] T. Tommasi, T. Tuytelaars, and B. Caputo. A testbed for cross-dataset analysis. In *TASK-CV Workshop, ECCV*, 2014. [2](#), [7](#)
- [31] A. Torralba and A. Efros. Unbiased look at dataset bias. In *Proc. CVPR*, 2011. [2](#), [3](#)
- [32] J. Yang, R. Yan, and A. Hauptmann. Adapting SVM classifiers to data with shifted distributions. In *ICDM Workshops*, 2007. [2](#)