

Deep Domain Confusion: Maximizing for Domain Invariance

Eric Tzeng, Judy Hoffman, Ning Zhang
 UC Berkeley, EECS & ICSI
 {etzeng, jhoffman, nzhang}@eecs.berkeley.edu

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

Trevor Darrell
 UC Berkeley, EECS & ICSI
 trevor@eecs.berkeley.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 on a standard benchmark. Fine-tuning deep models in a new domain can require a significant amount of data, which for many applications is simply not available. We propose a new CNN architecture which introduces an adaptation layer and an additional domain confusion loss, to learn a representation that is both semantically meaningful and domain invariant. We additionally show that a domain confusion metric can be used for model selection to determine the dimension of an adaptation layer and the best position for the layer in the CNN architecture. Our proposed adaptation method offers empirical performance which exceeds previously published results on a standard benchmark visual domain adaptation task.

1. Introduction

Dataset bias is a well known problem with traditional supervised approaches to image recognition [32]. A number of recent theoretical and empirical results have shown that supervised methods' test error increases in proportion to the difference between the test and training input distribution [3, 5, 29, 32]. In the last few years several methods for visual domain adaptation have been suggested to overcome this issue [10, 33, 2, 29, 25, 22, 17, 16, 19, 20], but were limited to shallow models. The traditional approach to adapting deep models has been fine-tuning; see [15] for a recent example.

Directly fine-tuning a deep network's parameters on a small amount of labeled target data turns out to be problematic. Fortunately, pre-trained deep models do perform well in novel domains. Recently, [11, 21] showed that using the deep mid-level features learned on ImageNet, instead of the more conventional bag-of-words features, effectively removed the bias in some of the domain adaptation settings in the Office dataset [29]. These algorithms transferred the representation from a large scale domain, ImageNet, as well

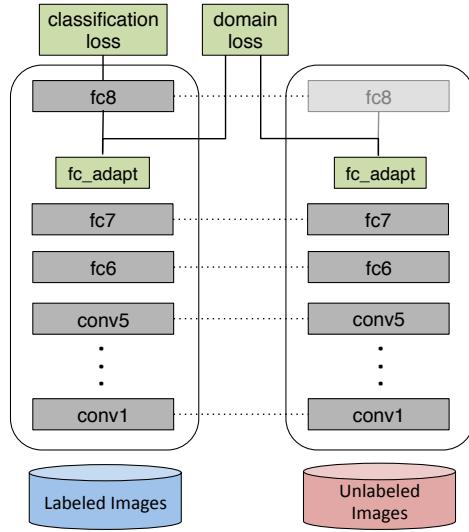


Figure 1: Our architecture optimizes a deep CNN for both classification loss as well as domain invariance. The model can be trained for *supervised* adaptation, when there is a small amount of target labels available, or *unsupervised* adaptation, when no target labels are available. We introduce domain invariance through *domain confusion* guided selection of the depth and width of the adaptation layer, as well as an additional domain loss term during fine-tuning that directly minimizes the distance between source and target representations.

as using all of the data in that domain as source data for appropriate categories. However, these methods have no way to select a representation from the deep architecture and instead report results across multiple layer selection choices.

Dataset bias was classically illustrated in computer vision by way of the “name the dataset” game of Torralba and Efros [32]. Indeed, this turns out to be formally connected to measures of domain discrepancy [23, 6]. 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 do-

mains appear as similar as possible. This principle forms the essence 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.

We propose a new CNN architecture, outlined in Figure 1, which uses an adaptation layer along with a domain confusion loss based on maximum mean discrepancy (MMD) [6] to automatically learn a representation jointly trained to optimize for classification and domain invariance. We show that our domain confusion metric can be used both to select the dimension of the adaptation layers, choose an effective placement for a new adaptation layer within a pre-trained CNN architecture, and fine-tune the representation.

Our architecture can be used to solve both *supervised adaptation*, when a small amount of target labeled data is available, and *unsupervised adaptation*, when no labeled target training data is available. We provide a comprehensive evaluation on the popular Office benchmark for classification across visually distinct domains [29]. We demonstrate that by jointly optimizing for domain confusion and classification, we are able to significantly outperform the current state-of-the-art visual domain adaptation results. In fact, for the case of minor pose, resolution, and lighting changes, our algorithm is able to achieve 96% accuracy on the target domain, demonstrating that we have in fact learned a representation that is invariant to these biases.

2. Related work

The concept of visual dataset bias was popularized in [32]. There have been many approaches proposed in recent years to solve the visual domain adaptation problem. 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 [6, 13, 23, 26, 28]. A large number of methods have sought to overcome this difference by learning a feature space transformation to align the source and target representations [29, 25, 13, 16]. 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 [33, 2, 1]. Others have sought to both learn a feature transformation and regularize a target classifier simultaneously [20, 12].

Recently, supervised convolutional neural network (CNN) based feature representations have been shown to be extremely effective for a variety of visual recognition tasks [24, 11, 15, 30]. In particular, using deep representations dramatically reduce the effect of resolution and lighting on domain shifts [11, 21].

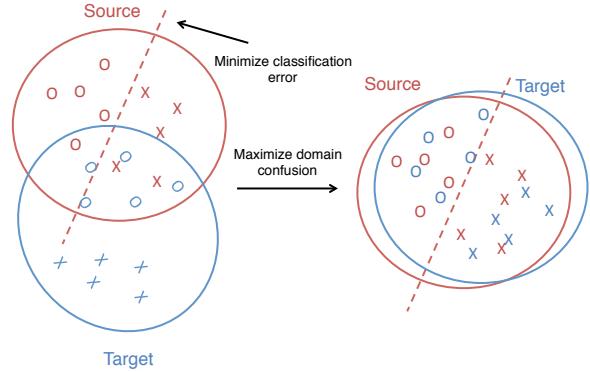


Figure 2: For biased datasets (left), classifiers learned in a source domain do not necessarily transfer well to target domains. By optimizing an objective that simultaneously minimizes classification error and maximizes *domain confusion* (right), we can learn representations that are discriminative and domain invariant.

Parallel CNN architectures such as Siamese networks have been shown to be effective for learning invariant representations [7, 9]. However, training these networks requires labels for each training instance, so it is unclear how to extend these methods to unsupervised settings.

Multimodal deep learning architectures have also been explored to learn representations that are invariant to different input modalities [27]. 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 [8], but was limited to two layers and so was significantly outperformed by the methods which used a deeper architecture [24], pre-trained on a large auxiliary data source (ex: ImageNet [4]).

[14] 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.

3. Training CNN-based domain invariant representations

We introduce a new convolutional neural network (CNN) architecture which we use to learn a visual representation that is both domain invariant and which offers strong semantic separation. It has been shown that a pre-trained CNN can be adapted for a new task through fine-tuning [15,

[30, 18]. However, in the domain adaptation scenario there is little, or no, labeled training data in the target domain so we can not directly fine-tune for the categories of interest, C in the target domain, T . Instead, we will use data from a related, but distinct source domain, S , where more labeled data is available from the corresponding categories, C .

Directly training a classifier using only the source data often leads to overfitting to the source distribution, causing reduced performance at test time when recognizing in the target domain. Our intuition is that if we can learn a representation that minimizes the distance between the source and target distributions, then we can train a classifier on the source labeled data and directly apply it to the target domain with minimal loss in accuracy.

To minimize this distance, we consider the standard distribution distance metric, Maximum Mean Discrepancy (MMD) [6]. This distance is computed with respect to a particular representation, $\phi(\cdot)$. In our case, we define a representation, $\phi(\cdot)$, which operates on source data points, $x_s \in X_S$, and target data points, $x_t \in X_T$. Then an empirical approximation to this distance is computed as followed:

$$\text{MMD}(X_S, X_T) = \left\| \frac{1}{|X_S|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \phi(x_t) \right\| \quad (1)$$

As Figure 2 shows, not only do we want to minimize the distance between domains (or maximize the *domain confusion*), but we want a representation which is conducive to training strong classifiers. Such a representation would enable us to learn strong classifiers that readily transfer across domains. One approach to meeting both these criteria is to minimize the loss:

$$\mathcal{L} = \mathcal{L}_C(X_L, y) + \lambda \text{MMD}^2(X_S, X_T) \quad (2)$$

where $\mathcal{L}_C(X_L, y)$ denotes classification loss on the available labeled data, X_L , and the ground truth labels, y , and $\text{MMD}(X_S, X_T)$ denotes the distance between the source data, X_S , and the target data, X_T . The hyperparameter λ determines how strongly we would like to confuse the domains.

One approach to minimizing this loss is to take a fixed CNN, which is already a strong classification representation, and use MMD to decide which layer to use activations from to minimize the domain distribution distance. We can then use this representation to train another classifier for the classes we are interested in recognizing. This can be viewed as coordinate descent on Eqn. 2: we take a network that was trained to minimize \mathcal{L}_C , select the representation that minimizes MMD, then use that representation to again minimize \mathcal{L}_C .

However, this approach is limited in that it cannot directly adapt the representation—instead, it is constrained to

selecting from a set of fixed representations. Thus, we propose creating a network to directly optimize the classification and domain confusion objectives, outlined in Figure 1.

We begin with the Krizhevsky architecture [24], which has five convolutional and pooling layers and three fully connected layers with dimensions {4096, 4096, $|C|$ }. We additionally add a lower dimensional, “bottleneck,” adaptation layer. Our intuition is that a lower dimensional layer can be used to regularize the training of the source classifier and prevent overfitting to the particular nuances of the source distribution. We place the domain distance loss on top of the “bottleneck” layer to directly regularize the representation to be invariant to the source and target domains.

There are two model selection choices that must be made to add our adaptation layer and the domain distance loss. We must choose where in the network to place the adaptation layer and we must choose the dimension of the layer. We use the MMD metric to make both of these decisions. First, as previously discussed, for our initial fixed representation we find the layer which minimizes the empirical MMD distance between all available source and target data, in our experiments this corresponded to placing the layer after the fully connected layer, $fc7$.

Next, we must determine the dimension for our adaptation layer. We solve this problem with a grid search, where we fine-tune multiple networks using various dimensions and compute the MMD in the new lower dimension representation, finally choosing the dimension which minimizes the source and target distance.

Both the selection of which layer’s representation to use (“depth”) and how large the adaptation layer should be (“width”) are guided by MMD, and thus can be seen as descent steps on our overall objective.

Our architecture (see Figure 1) consists of a source and target CNN, with shared weights. Only the labeled examples are used to compute the classification loss, while all data is used from both domains to compute the domain confusion loss. The network is jointly trained on all available source and target data.

The objective outlined in Eqn. 2 is easily represented by this convolutional neural network where MMD is computed over minibatches of source and target data. We simply use a fork at the top of the network, after the adaptation layer. One branch uses the labeled data and trains a classifier, and the other branch uses all the data and computes MMD between source and target.

After fine-tuning this architecture, owing to the two terms in the joint loss, the adaptation layer learns a representation that can effectively discriminate between the classes in question due to the classification loss term, while still remaining invariant to domain shift due the MMD term. We expect that such a representation will thus enable increased adaptation performance.

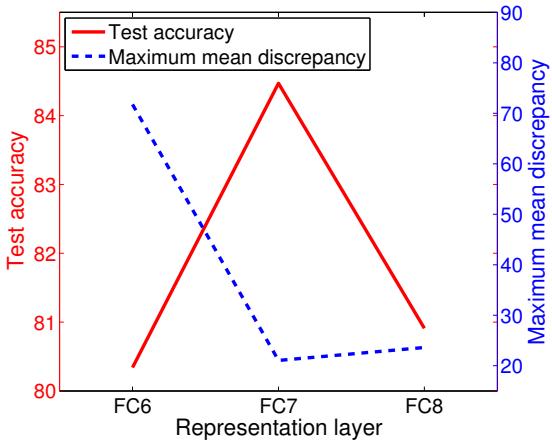


Figure 3: Maximum mean discrepancy and test accuracy for different choices of representation layer. We observe that MMD between source and target and accuracy on the target domain test set seem inversely related, indicating that MMD can be used to help select a layer for adaptation.

4. Evaluation

We evaluate our adaptation algorithm on a standard domain adaptation dataset with small-scale source domains. We show that our algorithm is effectively able to adapt a deep CNN representation to a target domain with limited or no target labeled data.

The Office [29] dataset is a collection of images from three distinct domains: Amazon, DSLR, and Webcam. The 31 categories in the dataset consist of objects commonly encountered in office settings, such as keyboards, file cabinets, and laptops. The largest domain has 2817 labeled images.

We evaluate our method across 5 random train/test splits for each of the 3 transfer tasks commonly used for evaluation (Amazon→Webcam, DSLR→Webcam, and Webcam→DSLR) and report averages and standard errors for each setting. We compare in both supervised and unsupervised scenarios against the numbers reported by six recently published methods.

We follow the standard training protocol for this dataset of using 20 source examples per category for the Amazon source domain and 8 images per category for Webcam or DSLR as the source domains [29, 16]. For the supervised adaptation setting we assume 3 labeled target examples per category.

4.1. Evaluating adaptation layer placement

We begin with an evaluation of our representation selection strategy. Using a pre-trained convolutional neural network, we extract features from source and target data using the representations at each fully connected layer. We can

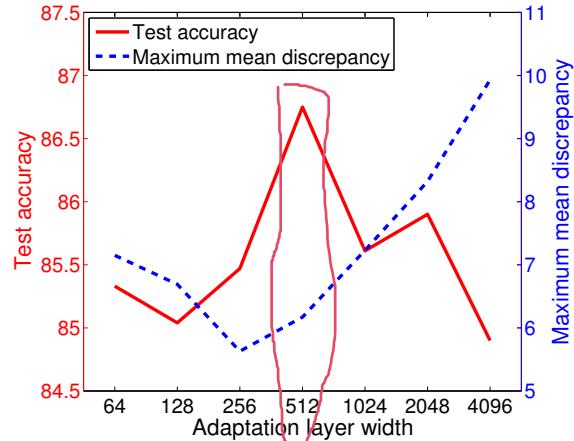


Figure 4: Maximum mean discrepancy and test accuracy for different values of adaptation layer dimensionality. We observe that MMD between source and target and accuracy on the target domain test set seem inversely related, indicating that MMD can be used to help select a dimensionality to use.

then compute the MMD between source and target at each layer. Since a lower MMD indicates that the representation is more domain invariant, we expect the representation with the lowest MMD to achieve the highest performance after adaptation.

To test this hypothesis, for one of the Amazon→Webcam splits we apply a simple domain adaptation baseline introduced by Daumé III [10] to compute test accuracy for the target domain. Figure 3 shows a comparison of MMD and adaptation performance across different choices of bridge layers. We see that MMD correctly ranks the representations, singling out *fc7* as the best performing layer and *fc6* as the worst. Therefore, we add our adaptation layer after *fc7* for the remaining experiments.

4.2. Choosing the adaptation layer dimension

Before we can learn a new representation via our proposed fine-tuning method, we must determine how wide this representation should be. Again, we use MMD as the deciding metric.

In order to determine what dimensionality our learned adaptation layer should have, we train a variety of networks with different widths on the Amazon→Webcam task, as this is the most challenging of the three. In particular, we try different widths varying from 64 to 4096, stepping by a power of two each time. Once the networks are trained, we then compute MMD between source and target for each of the learned representations. Our method then selects the dimensionality that minimizes the MMD between the source and target data.

	$A \rightarrow W$	$D \rightarrow W$	$W \rightarrow D$	Average
GFK(PLS,PCA) [16]	46.4 ± 0.5	61.3 ± 0.4	66.3 ± 0.4	53.0
SA [13]	45.0	64.8	69.9	59.9
DA-NBNN [31]	52.8 ± 3.7	76.6 ± 1.7	76.2 ± 2.5	68.5
DLID [8]	51.9	78.2	89.9	73.3
DeCAF ₆ S+T [11]	80.7 ± 2.3	94.8 ± 1.2	—	—
DaNN [14]	53.6 ± 0.2	71.2 ± 0.0	83.5 ± 0.0	69.4
Ours	84.1 ± 0.6	95.4 ± 0.4	96.3 ± 0.3	91.9

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 [29]. Here, we compare against six state-of-the-art domain adaptation methods.

	$A \rightarrow W$	$D \rightarrow W$	$W \rightarrow D$	Average
GFK(PLS,PCA) [16]	15.0 ± 0.4	44.6 ± 0.3	49.7 ± 0.5	36.4
SA [13]	15.3	50.1	56.9	40.8
DA-NBNN [31]	23.3 ± 2.7	67.2 ± 1.9	67.4 ± 3.0	52.6
DLID [8]	26.1	68.9	84.9	60.0
DeCAF ₆ S [11]	52.2 ± 1.7	91.5 ± 1.5	—	—
DaNN [14]	35.0 ± 0.2	70.5 ± 0.0	74.3 ± 0.0	59.9
Ours	59.4 ± 0.8	92.5 ± 0.3	91.7 ± 0.8	81.2

Table 2: Multi-class accuracy evaluation on the standard unsupervised adaptation setting with the *Office* dataset. We evaluate on all 31 categories using the standard experimental protocol from [16]. Here, we compare against six state-of-the-art domain adaptation methods.

To verify that MMD makes the right choice, again we compare MMD with performance on a test set. Figure 4 shows that we select 256 dimensions for the adaptation layer, and although this setting is not the one that maximizes test performance, it appears to be a reasonable choice. In particular, using MMD avoids choosing either extreme, near which performance suffers. It is worth noting that the plot has quite a few irregularities—perhaps finer sampling would allow for a more accurate choice.

4.3. Fine-tuning with domain confusion regularization

Once we have settled on our choice of adaptation layer dimensionality, we can begin fine-tuning using the joint loss described in Section 3. However, we need to set the regularization hyperparameter λ . Setting λ too low will cause the MMD regularizer have no effect on the learned representation, but setting λ too high will regularize too heavily and learn a degenerate representation in which all points are too close together. We set the regularization hyperparameter to $\lambda = 0.25$, which makes the objective primarily weighted towards classification, but with enough regularization to avoid overfitting.

We use the same fine-tuning architecture for both unsu-

pervised and supervised. However, in the supervised setting, the classifier is trained on data from both domains, whereas in the unsupervised setting, due to the lack of labeled training data, the classifier sees only source data. In both settings, the MMD regularizer sees all of the data, since it does not require labels.

Finally, because the adaptation layer and classifier are being trained from scratch, we set their learning rates to be 10 times higher than the lower layers of the network that were copied from the pre-trained model. Fine-tuning then proceeds via standard backpropagation optimization.

The supervised adaptation setting results are shown in Table 1 and the unsupervised adaptation results are shown in Table 2. We notice that our algorithm dramatically outperforms all of the competing methods. The distinct improvement of our method demonstrates that the adaptation layer learned via MMD regularized fine-tuning is able to successfully transfer to a new target domain.

In order to determine how MMD regularization affects learning, we also compare the learning curves with and without regularization on the Amazon → Webcam transfer task in Figure 5. We see that, although the unregularized version is initially faster to train, it quickly begins overfitting, and test accuracy suffers. In contrast, using MMD

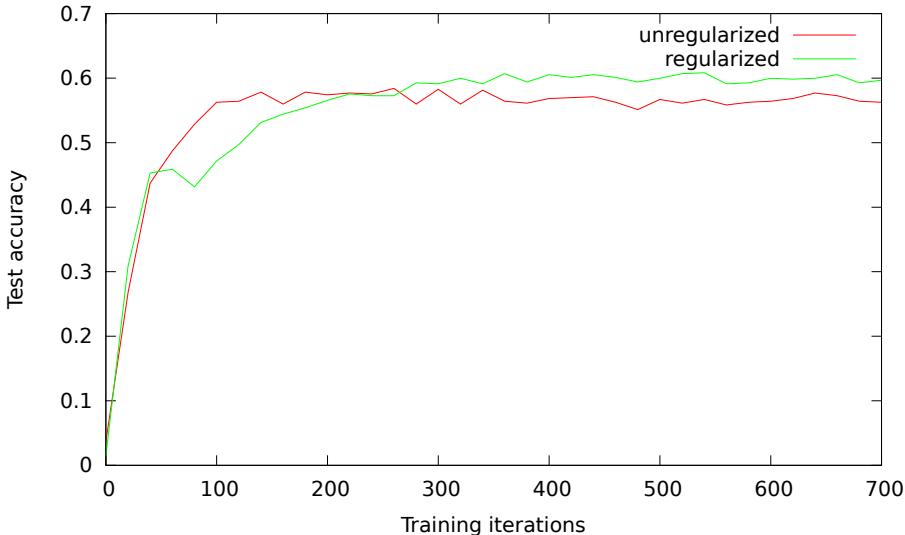


Figure 5: A plot of the test accuracy on an unsupervised Amazon→Webcam split during the first 700 iterations of fine-tuning for both regularized and unregularized methods. Although initially the unregularized training achieves better performance, it overfits to the source data. In contrast, using regularization prevents overfitting, so although initial learning is slower we ultimately see better final performance.

regularization prevents the network from overfitting to the source data, and although training takes longer, the regularization results in a higher final test accuracy.

To further demonstrate the domain invariance of our learned representation, we plot in Figure 6 a t-SNE embedding of Amazon and Webcam images using our learned representation and compare it to an embedding created with *fc7* in the pretrained model. Examining the embeddings, we see that our learned representation exhibits tighter class clustering while mixing the domains within each cluster. While there is weak clustering in the *fc7* embedding, we find that most tight clusters consist of data points from one domain or the other, but rarely both.

4.4. Historical Progress on the Office Dataset

In Figure 7 we report historical progress on the standard Office dataset since it's introduction. We indicate methods which use traditional features (ex: SURF BoW) with a blue circle and methods which use deep representations with a red square. We show two adaptation scenarios. The first scenario is a supervised adaptation task for visually distant domains (Amazon→Webcam). For this task our algorithm outperforms DeCAF by 3.4% multiclass accuracy. Finally, we show the hardest task of unsupervised adaptation for that same shift. Here we show that our method provides the most significant improvement of 5.5% multiclass accuracy.

5. Conclusion

In this paper, we presented an objective function for learning domain invariant representations for classification. This objective makes use of an additional domain confusion term to ensure that domains are indistinguishable in the learned representation. We then presented a variety of ways to optimize this objective, ranging from simple representation selection from a fixed pool to a full convolutional architecture that directly minimizes the objective via back-propagation.

Our full method, which uses MMD both to select the depth and width of the architecture while using it as a regularizer during fine-tuning, achieves state-of-the-art performance on the standard visual domain adaptation benchmark, beating previous methods by a considerable margin.

These experiments show that incorporating a domain confusion term into the discriminative representation learning process is an effective way to ensure that the learned representation is both useful for classification and invariant to domain shifts.

Acknowledgments

This work was supported in part by DARPA's MSEE and SMISC programs, NSF awards IIS-1427425, IIS-1212798, and IIS-1116411, Toyota, and the Berkeley Vision and Learning Center.

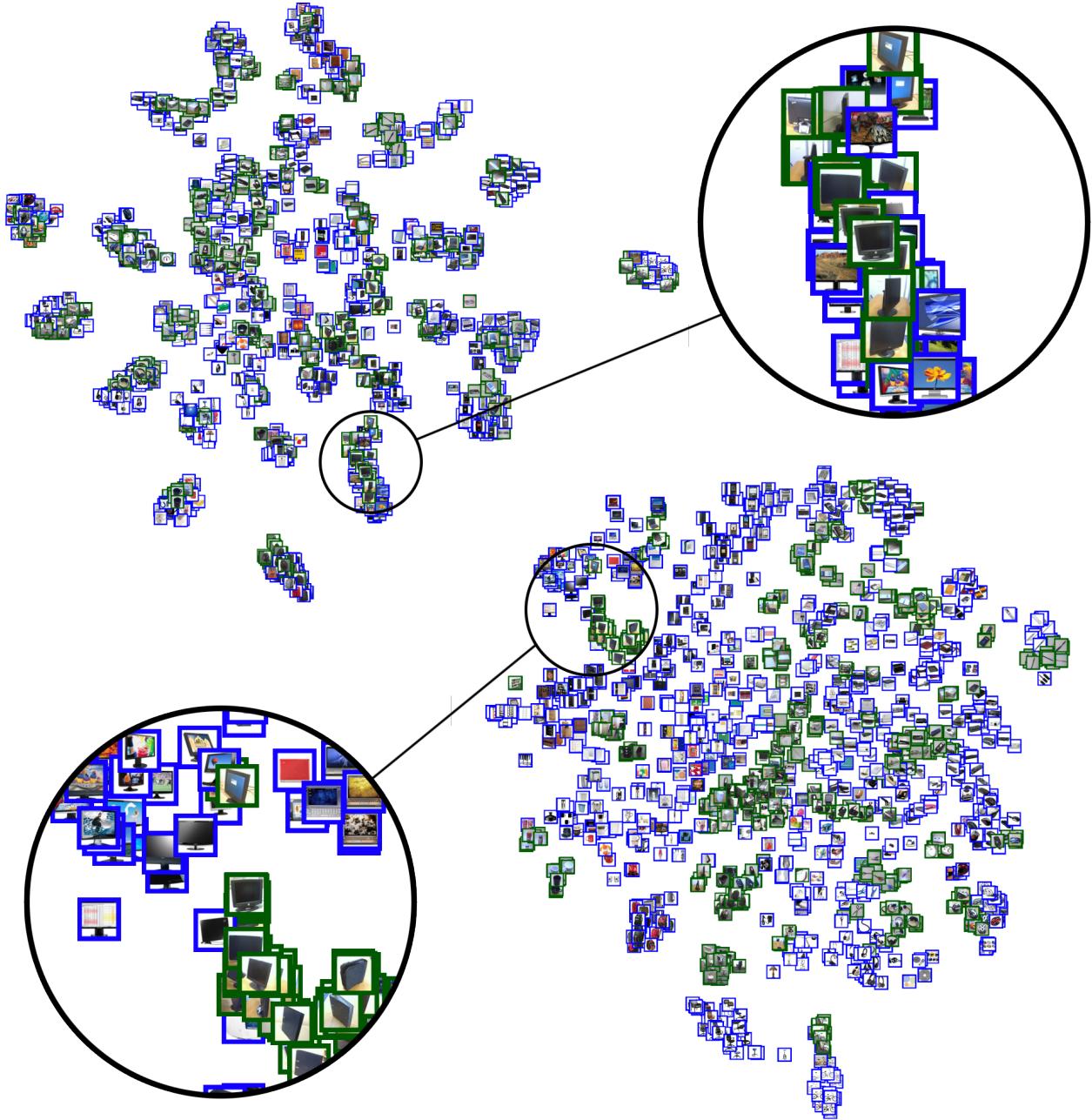


Figure 6: t-SNE embeddings of Amazon (blue) and Webcam (green) images using our supervised 256-dimensional representation learned with MMD regularization (top left) and the original *fc7* representation from the pre-trained model (bottom right). Observe that the clusters formed by our representation separate classes while mixing domains much more effectively than the original representation that was not trained for domain invariance. For example, in *fc7*-space the Amazon monitors and Webcam monitors are separated into distinct clusters, whereas with our learned representation all monitors irrespective of domain are mixed into the same cluster.

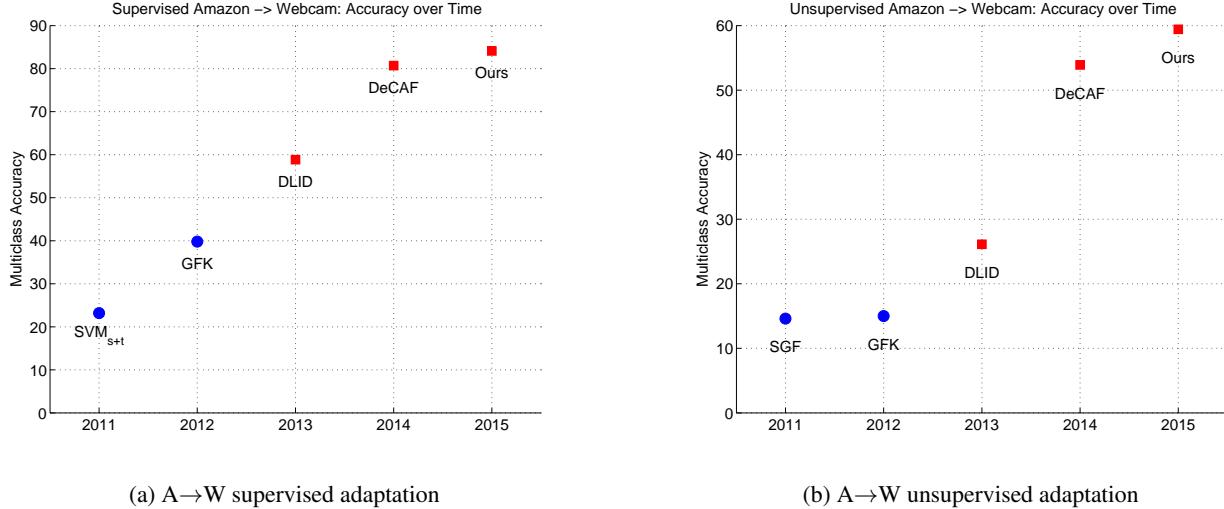


Figure 7: Rapid progress over the last few years on a standard visual domain adaptation dataset, *Office* [29]. We show methods on Amazon→Webcam that use traditional hand designed visual representations with blue circles and methods that use deep representations are depicted with red squares. For the supervised task, our method achieves 84% multiclass accuracy, an increase of 3%. For the unsupervised task, our method achieves 60% multiclass accuracy, an increase of 6%.

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. 1, 2
- [3] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, et al. Analysis of representations for domain adaptation. *Proc. NIPS*, 2007. 1
- [4] A. Berg, J. Deng, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge 2012. 2012. 2
- [5] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. In *Proc. NIPS*, 2007. 1
- [6] 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. 1, 2, 3
- [7] 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
- [8] 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, 5
- [9] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition*, 2005.
- [10] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007. 1, 4
- [11] 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. 1, 2, 5
- [12] L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for heterogeneous domain adaptation. In *Proc. ICML*, 2012. 2
- [13] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proc. ICCV*, 2013. 2, 5
- [14] M. Ghifary, W. B. Kleijn, and M. Zhang. Domain adaptive neural networks for object recognition. *CoRR*, abs/1409.6041, 2014. 2, 5
- [15] 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
- [16] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proc. CVPR*, 2012. 1, 2, 4, 5
- [17] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proc. ICCV*, 2011. 1
- [18] 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. 2
- [19] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *Proc. ECCV*, 2012. 1

- [20] J. Hoffman, E. Rodner, J. Donahue, K. Saenko, and T. Darrell. Efficient learning of domain-invariant image representations. In *Proc. ICLR*, 2013. [1](#), [2](#)
- [21] 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. [1](#), [2](#)
- [22] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *Proc. ECCV*, 2012. [1](#)
- [23] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proc. VLDB*, 2004. [1](#), [2](#)
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. [2](#), [3](#)
- [25] 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. [1](#), [2](#)
- [26] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009. [2](#)
- [27] 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](#)
- [28] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *IJCA*, 2009. [2](#)
- [29] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proc. ECCV*, 2010. [1](#), [2](#), [4](#), [5](#), [8](#)
- [30] 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. [2](#)
- [31] T. Tommasi and B. Caputo. Frustratingly easy NBNN domain adaptation. In *Proc. ICCV*, 2013. [5](#)
- [32] A. Torralba and A. Efros. Unbiased look at dataset bias. In *Proc. CVPR*, 2011. [1](#), [2](#)
- [33] J. Yang, R. Yan, and A. Hauptmann. Adapting SVM classifiers to data with shifted distributions. In *ICDM Workshops*, 2007. [1](#), [2](#)