Boosting Domain Adaptation by Discovering Latent Domains Supplementary Material

Massimiliano Mancini[1,2], Lorenzo Porzi[3], Samuel Rota Bulò[3], Barbara Caputo[1,4], Elisa Ricci[2,5]
[1]Sapienza University of Rome, [2]Fondazione Bruno Kessler, [3]Mapillary Research,
[4]Italian Institute of Technology, [5]University of Trento
{mancini,caputo}@diag.uniroma1.it, {lorenzo,samuel}@mapillary.com, eliricci@fbk.eu

Abstract

This document provides the following additional contributions to our CVPR 2018 submission:

- *In Section 1, we provide the derivation of the formulas for the forward and backward pass of our mDA layers.*
- *In Section 2, we give additional details about the networks employed in our experimental analysis and the associated training protocols.*
- *In Section 3 we show additional experimental results on the PACS dataset.*

1. mDA layers formulas

From the main text, we have the output of our mDA layer denoted by

$$y_i = \text{mDA}(x_i, \boldsymbol{w}_i; \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}) = \sum_{d \in \mathcal{D}} w_{i,d} \hat{x}_{i,d}, \tag{1}$$

where, for simplicity:

$$\hat{x}_{i,d} = \frac{x_i - \hat{\mu}_d}{\sqrt{\hat{\sigma}_d^2 + \epsilon}}, \tag{2}$$

and the statistics are given by

$$\begin{aligned} \hat{\mu}_d &= \sum_{i=1}^{\mathsf{b}} \hat{w}_{i,d} x_i, \\ \hat{\sigma}_d^2 &= \sum_{i=1}^{\mathsf{b}} \hat{w}_{i,d} (x_i - \hat{\mu}_d)^2, \end{aligned} \tag{3}$$

where $\hat{w}_{i,d} = w_{i,d} / \sum_{j=1}^{\mathsf{b}} w_{j,d}$.

From the previous equations we can derive the partial derivative of the loss function with respect to both the input x_i and the domain assignment probabilities $w_{i,d}$. Let us denote $\frac{\partial L}{\partial y_i}$ the partial derivative of the loss function L with respect to the output y_i of the mDA layer. We have:

$$\begin{aligned} \frac{\partial \hat{x}_{i,d}}{\partial \hat{\sigma}_{d^*}^2} &= -\mathbb{1}_{d=d^*} \frac{1}{2} (x_i - \hat{\mu}_{d^*}) \cdot (\hat{\sigma}_{d^*}^2 + \epsilon)^{-\frac{3}{2}}, \\ \frac{\partial \hat{x}_{i,d}}{\partial \hat{\mu}_{d^*}} &= -\mathbb{1}_{d=d^*} (\hat{\sigma}_{d^*}^2 + \epsilon)^{-\frac{1}{2}}, \end{aligned} \tag{4}$$

and

$$\frac{\partial \hat{\sigma}_d^2}{\partial x_i} = 2\, \hat{w}_{i,d} \cdot (x_i - \hat{\mu}_d), \quad \frac{\partial \hat{\mu}_d}{\partial x_i} = \hat{w}_{i,d}. \tag{5}$$

Thus, the partial derivative of L w.r.t. the input x_i is:

$$\frac{\partial L}{\partial x_{i^*}} = \sum_{d \in \mathcal{D}} \frac{w_{i^*,d}}{\sqrt{\hat{\sigma}_d^2 + \epsilon}} \left[\frac{\partial L}{\partial y_{i^*}} - A_d - \hat{x}_{i^*,d} B_d \right], \tag{6}$$

where:

$$\begin{aligned} A_d &= \sum_{i=1}^{\mathsf{b}} \hat{w}_{i,d} \frac{\partial L}{\partial y_i}, \\ B_d &= \sum_{i=1}^{\mathsf{b}} \hat{w}_{i,d} \hat{x}_{i,d} \frac{\partial L}{\partial y_i}. \end{aligned} \tag{7}$$

For the domain assignment probabilities $w_{i,d}$ we have:

$$\frac{\partial \hat{\mu}_d}{\partial \hat{w}_{i,d^*}} = \mathbb{1}_{d=d^*} x_i, \tag{8}$$

$$\frac{\partial \hat{\sigma}_d^2}{\partial \hat{w}_{i,d^*}} = \mathbb{1}_{d=d^*} (x_i - \hat{\mu}_d)^2, \tag{9}$$

$$\frac{\partial \hat{w}_{i,d}}{\partial w_{i^*,d^*}} = \mathbb{1}_{d=d^*} \frac{\mathbb{1}_{i=i^*} \sum_{j=1}^{\mathsf{b}} w_{j,d} - w_{i,d}}{(\sum_{j=1}^{\mathsf{b}} w_{j,d})^2}. \tag{10}$$

Thus, the partial derivative of L w.r.t. $w_{i,d}$ is:

$$\frac{\partial L}{\partial w_{i^*,d}} = \hat{x}_{i^*,d} \left(\frac{\partial L}{\partial y_{i^*}} - A_d \right) - \frac{1}{2} \left(\hat{x}_{i^*,d}^2 - \frac{\sigma_d^2}{\sigma_d^2 + \epsilon} \right) B_d, \tag{11}$$

where A_d and B_d are defined as in (7).

2. Networks and training protocols

In this section, we provide additional details about the networks and training procedures employed for the experiments on digits datasets, Office31 and Office-Caltech.

The network adopted for the evaluation on digits datasets is the standard architecture from [3]. It is composed by 3 convolutional layers, the first 2 followed by max pooling, and 2 fully-connected layers before the final classifier. The 2 fully-connected layers are followed by dropout [4]. For all the experiments the architecture is trained for 15000 iterations. Following the protocol described in [1, 3], we set the initial learning rate l_0 to 0.01 and we anneal it through a schedule l_p defined by $l_p = \frac{l_0}{(1+\gamma p)^\beta}$ where $\beta = 0.75$, $\gamma = 10$ and p is the training progress increasing linearly from 0 to 1. We rescale the input images to 32×32 pixels, subtract the per-pixel image mean of the dataset and feed the networks with random crops of size 28×28.

For the experiments with the AlexNet architecture, we fix the parameter of all convolutional layers (denoted as `conv1 – conv5`), while fine-tuning the fully-connected ones (denoted as `fc6 – fc8`). mDA-layers are inserted following `fc6`, `fc7` and `fc8` and before their corresponding activation functions. The network is trained for 60 epochs with a batch-size of 256. The batch is split between source and target samples proportionally to the number of images of the different domains. We use stochastic gradient descent as optimizer, with a learning rate of 10^{-3}, a weight-decay of 0.0005 and a momentum of 0.9. The final classifier has an higher learning rate, 10^{-2}, and all the learning-rate values are scaled by 0.1 after 90% of the epochs. We rescale the input images to 256×256 pixels, subtract the per-channel image mean (computed on ImageNet) and feed the networks with random crops of size 227×227.

3. Additional results on the PACS dataset

In this section we provide some additional quantitative and qualitative results on the PACS dataset. The first series of experiments further demonstrates the importance of considering multi-source DA models over single-source ones. In a second series of experiments we evaluate the ability of our approach to discover latent domains.

3.1. Importance of multi-source DA

Similarly to Table 1 of the main paper, we first report the performances of our model for different values of k. Table 1 shows the results. As k increases from 1 to 3 (where 1 is the unified sources model [2] and 3 is the actual number of source domains) the average accuracy also improves. For values of k larger than 3 the performance saturates, obtaining a similar trend to that we observed in the experiments on the digits datasets.

To further demonstrate the advantages of adopting a multi-source domain adaptation approach and the benefit of considering different statistics for aligning features of different domains, we conduct an additional analysis. Specifically, we consider the Multi-source DA method and analyze how the performance of the target classifier varies when adopting the batch normalization statistics of different domains. In other words, we substitute the statistics of DA layers for the target model with those associated to one of the source domains, observing how this change influences the performances. Of course, we also consider the case where the "right" statistics, *i.e.* those corresponding to target domain, are used. We repeat this analysis for each possible combination of sources/target domains. Figure 1 shows the results. Each group of four color bars indicates a different experiment (*i.e.* a different target domain).

As expected, when the statistics of Sketch are used, there is a huge drop in performances in every scenario (*i.e.* when Photo, Art and Cartoon are considered as target) due to the distance of the feature distributions of this domain from those of the other domains. When Sketch is considered as target, the accuracy decreases significantly if the statistics of Photo or Art domains are employed. Instead, using the statistics derived from the Cartoon domain, higher accuracy is obtained. This is not surprising due to the similarity

Table 1: PACS dataset: performances of our model for different values of k with the ResNet architecture. The first row indicates the target domain, while all the others are considered as sources.

Method	Sketch	Photo	Art	Cartoon	Mean
DIAL [2]	66.8	97.0	87.3	85.5	84.2
Ours k=2	68.1	96.9	87.2	86.4	84.7
Ours k=3	69.6	97.0	**87.7**	86.9	85.3
Ours k=4	70.4	**97.1**	**87.7**	**87.3**	**85.6**
Ours k=5	**72.1**	96.9	86.5	87.1	**85.6**

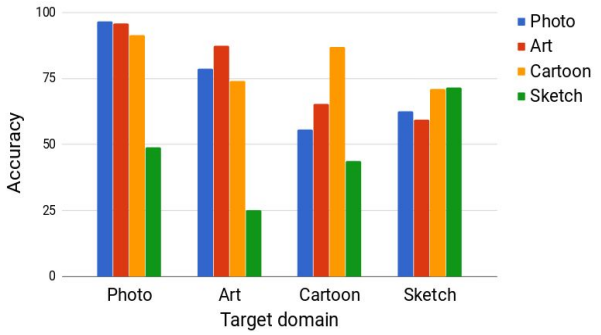


Figure 1: PACS dataset: classification accuracies of the Multi-source DA model as CNN feature representations of a domain are aligned using statistics of a different domain.

in visual appearance between images of the Cartoon and the Sketch domains. Importantly, only a multi-source DA model can exploit this similarity, while traditional single-source approaches will not suffice.

3.2. Latent domain discovery

To demonstrate the ability of our approach to discover latent domains we provide some additional results.

We first show how our approach assigns source samples to different latent domains. Assignments are computed considering the soft-max scores obtained with the domain prediction branch. Figure 2 reports the percentage of source samples assigned to each latent domain. The four plots correspond to a single run of the experiments reported in Table 2 of the main paper. Each plot is associated to a different target domain. Different colors are associated to the original source domains, while the x-axis indicates the latent domain. Interestingly, when either Cartoon (Figure 2c) or Sketch (Figure 2d) are considered as target domains, samples from Photo and Art tend to be associated to the same latent domain. Similarly, when either Photo (Figure 2a) or Art (Figure 2b) are considered as target domains, samples from Cartoon and Sketch tend to be grouped together. These results confirms the ability of our approach to assign images of similar visual appearance to the same latent distribution, in order to build stronger target classifiers.

Additionally, in Figure 3 we show the top-10 images associated to each latent domain for each sources/target setting. In each plot each row corresponds to a different latent domain. It is possible to notice how images associated to the same latent domain have similar appearance, while there is high dissimilarity between images associated to different latent domains. Moreover, images assigned to the same latent domain tend to be associated with one of the original domains. For instance, in Figure 3a, the first row contains only images of domain Art, while the third contains only images of domain Sketch.

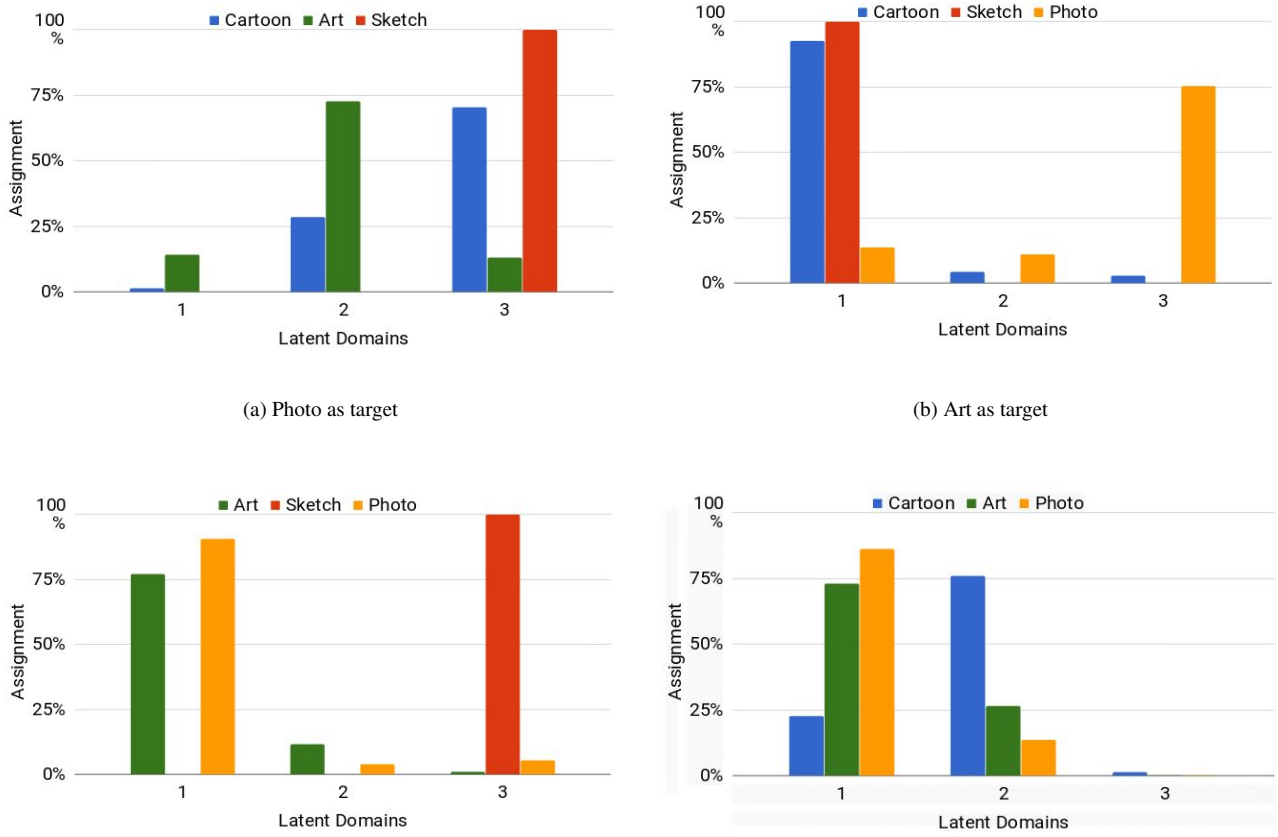
References

[1] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulò. Autodial: Automatic domain alignment layers. *ICCV*, 2017. 2

[2] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulò. Just dial: Domain alignment layers for unsupervised domain adaptation. *arXiv preprint arXiv:1702.06332*, 2017. 2

[3] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *ICML*, 2015. 2

[4] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 2



(a) Photo as target

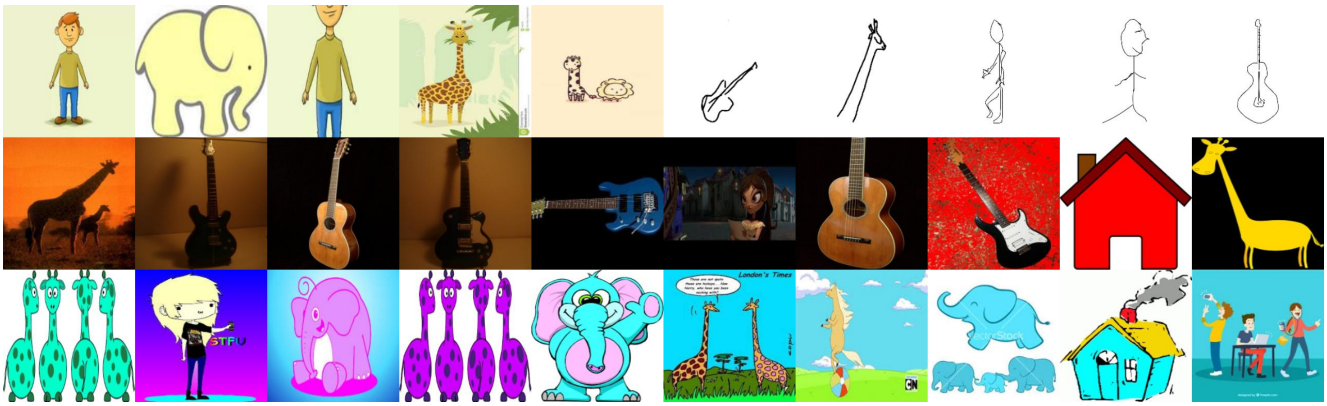
(b) Art as target

(c) Cartoon as target

(d) Sketch as target

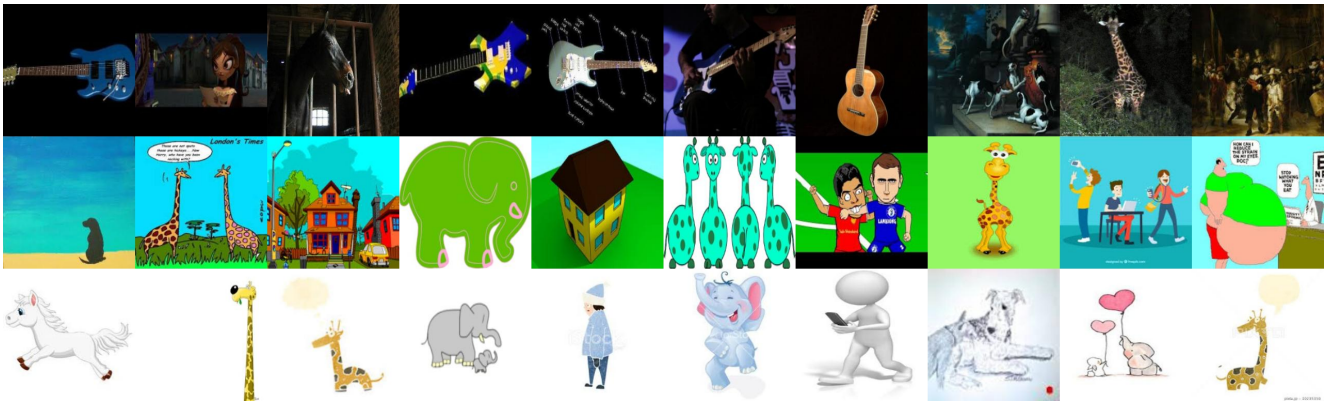
Figure 2: Distribution of the assignments produced by the domain prediction branch for each latent domain in all possible settings of the PACS dataset. Different colors denote different source domains.

(a) Photo as target



(b) Art as target

(c) Cartoon as target



(d) Sketch as target

Figure 3: Top-10 images associated to each latent domain for the different sources/target combinations. Each row corresponds to a different latent domain.