Learning to Cluster under Domain Shift: Supplementary Material

Willi Menapace[1], Stéphane Lathuilière[3], and Elisa Ricci[1,2]

[1] University of Trento, Trento, Italy
[2] Fondazione Bruno Kessler, Trento, Italy
[3] LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France
willi.menapace@gmail.com

1 Additional Implementation Details

We use a ResNet-18 backbone for all the experiments and report the hyperparameters used for each in Table 1. For all datasets we compose $\mathcal{T}$ using random crops, random horizontal flips and random hue, saturation and brightness changes. The input resolution to the network is 64×64px and on our GPUS with 8GB of memory allows the use of a maximum batch size of 162 images. The value of α is a function of the number of ground truth classes and the number of source domains in each dataset. A larger number of ground truth classes C causes a larger probability matrix $P_{cc'}$ to be estimated, while a higher number of source domains empirically causes more instability, probably due to a higher variance of features despite alignment. In particular, the PACS [3] dataset with $C = 7$ does not suffer much from training instability problems, so a value of $\alpha = 0.7$ is chosen. On the other hand, the Office31 [4] and the Office-Home [5] datasets with respectively $C = 31$ and $C = 65$ pose more stability problems. As we note in our Ablation Study on the main paper, a value of $\alpha = 0.1$ produces the best results on the Office31 dataset when training on all the dataset domains. Since in the standard experimental setting, however, we consider one domain as the target and train on one less source domain, we decide to use a higher value of $\alpha = 0.2$. The number of classes produced by the overclustering head C_{oc} is chosen following [2] which obtains the best results when choosing a value from 5 to 7 times C. Note that on the Office-Home dataset, due to the high number of ground truth classes, a factor of 2 is used. We use the Adam optimizer with learning rate 10^{-4} in all the experiments.

2 Additional Details about Baselines

As stated in the main paper, we make use of the IIC [2] and DeepCluster [1] algorithms as baselines for the evaluation of our method. With regards to the IIC baseline, we make use of the same ResNet-18 backbone and the same hyperparameters reported in Table 1 to guarantee fairness in the evaluation. Note that we do not employ the α-smoothing strategy in the IIC baseline in order to follow their exact implementation. For DeepCluster, we use a ResNet-18 backbone and

Table 1: Hyperparameter values used during the experiments. s denotes the number of times each head is replicated to improve training stability, C_{oc} represents the number of output classes used in the auxiliary overclustering head.

Dataset	Task	BS	α	s	C	C_{oc}
PACS	C,P,S→A	162	0.7	5	7	49
PACS	A,P,S→C	162	0.7	5	7	49
PACS	A,C,S→P	162	0.7	5	7	49
PACS	A,C,P→S	162	0.7	5	7	49
Office31	D,W→A	162	0.2	5	31	155
Office31	A,W→D	162	0.2	5	31	155
Office31	A,D→W	162	0.2	5	31	155
Office-Home	C,P,R→A	162	0.2	5	65	130
Office-Home	A,P,R→C	162	0.2	5	65	130
Office-Home	A,C,R→P	162	0.2	5	65	130
Office-Home	A,C,P→R	162	0.2	5	65	130

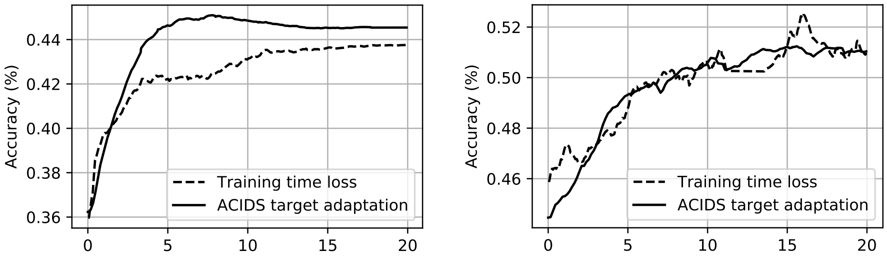
train it using an SGD optimizer with learning rate 10^{-2} for all the experiments. Despite [1] suggests the use of a number of clusters for self-supervision during training equal to 10 times the number of ground truth classes, we employed smaller factors due to the small number of samples in the datasets which does not allow the intermediate K-means procedure to work effectively with a high number of clusters.

3 Additional Target Adaptation Experimental Results

In this section, we propose to further investigate the effectiveness of the ACIDS target domain adaptation procedure. We perform training on the source domains and, starting from the same network parameters, we perform two different target adaptation procedures. The first uses the ACIDS adaptation procedure described in the main paper, the second performs adaptation using the same mutual information loss used at training time, computed on the target domain. In order to illustrate the stable convergence of the proposed model, we show in Fig.1 the evolution of the accuracy on the target domain while performing adaptation on two domains of the PACS dataset. The proposed target adaptation procedure leads to faster convergence and higher accuracy on the Cartoon domain (Fig.1-left), while it produces no appreciable effects on the Sketch domain (Fig.1-right).

4 Additional Parameter Ablation

We perform an evaluation of the effect of the α parameter described in Sec.3.3 on the main paper. Since it would be computationally expensive to run separate



Cartoon

Sketch

Fig. 1: Evolution of accuracy on the target domain during the target adaptation phase using Cartoon (left) and Sketch (right) as target domains. The solid line refers to the ACIDS target adaptation procedure, the dashed one refers to adaptation using the same mutual information maximization procedure used during training. Time on the x-axis is expressed in thousands of optimization steps.

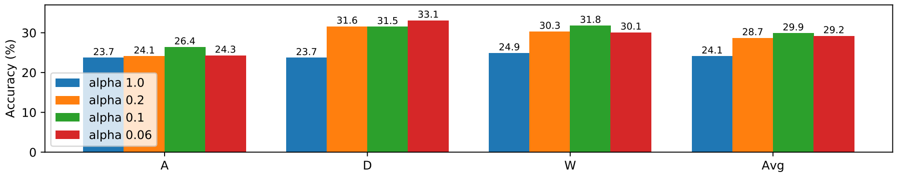


Fig. 2: Ablation of the α parameter on the Office 31 dataset. Labels express the source domain (A, D or W). Results expressed in accuracy (%).

training processes for every value of α and every domain, we adopt an evaluation protocol where we perform training without target adaptation, considering every domain as a source domain. For these experiments, we choose the Office31 dataset that is especially challenging in terms of optimization because of its high number of classes C.

Fig.2 reports the numerical evaluation results. Without using our stabilization method ($\alpha = 1.0$), we obtain degraded results due to noise in the estimation of $P_{cc'}$. Lowering the value to $\alpha = 0.1$ improves the estimation, achieving $+5.8\%$ average accuracy with respect to $\alpha = 0$. When further decreasing the value, however, accuracy starts to decrease. The estimation, in this case, becomes incorrect because it is influenced by network parameters that differ too much from the current ones.

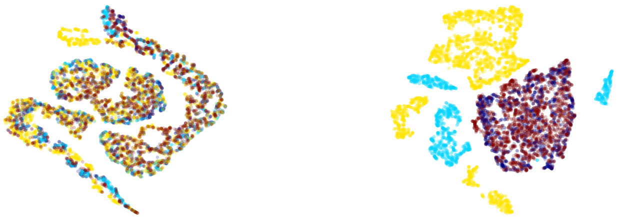


Fig. 3: t-SNE visualizations of the feature space extracted before the classification heads for the proposed method (left) and the same method without the mutual information loss for feature alignment Eq.(4) on the main paper (right), using PACS Cartoon as the target domain. Colors represent the different target domains. While in the proposed method the distributions of the domains align (left), when the feature alignment loss is removed (right) the method produces clusters based on the style rather than the content information (best viewed in color).

5 Feature Alignment via Mutual Information Qualitative Evaluation

In Fig.3 we report a qualitative analysis of the effect of the proposed mutual information minimization procedure for feature alignment on the feature space. The analysis shows that without the proposed procedure, the method produces clusters based on style rather than the image semantics, while the desired domain alignment is obtained when employing the proposed method.

6 Qualitative Clustering Results

In this section, we present qualitative clustering results produced by our method. Each cluster visualization corresponds to the results produced by the model after adaptation to the corresponding target domain. A visual inspection of the produced clusters reveals that classes with the most distinctive features such as “Giraffe” in PACS (Fig.6) or “Bike” in Office31 (Fig.8) tend to be clustered best, while classes with shapes similar to others tend to be confused like the “Desktop Computer” class in Office31. (Fig.12).

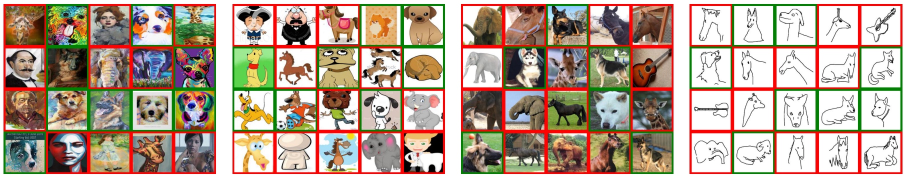


Fig. 4: Clusters corresponding to the PACS "Dog" class from Art, Cartoon, Photo and Sketch domains.

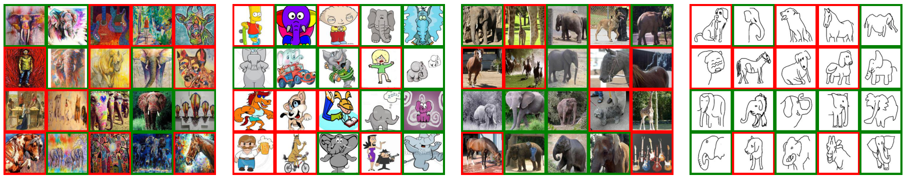


Fig. 5: Clusters corresponding to the PACS "Elephant" class from Art, Cartoon, Photo and Sketch domains.

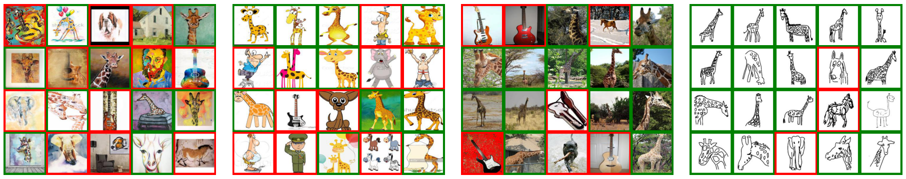


Fig. 6: Clusters corresponding to the PACS "Giraffe" class from Art, Cartoon, Photo and Sketch domains.

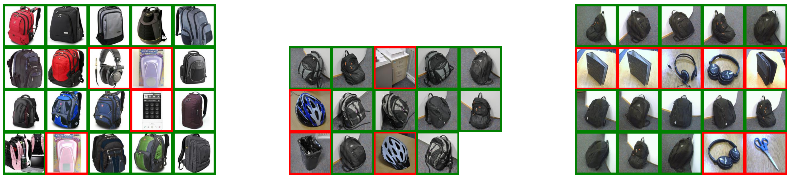


Fig. 7: Clusters corresponding to the Office31 "Back Pack" class from Amazon, DSLR and Webcam domains.

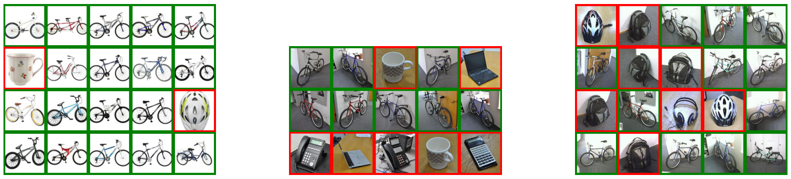


Fig. 8: Clusters corresponding to the Office31 "Bike" class from Amazon, DSLR and Webcam domains.

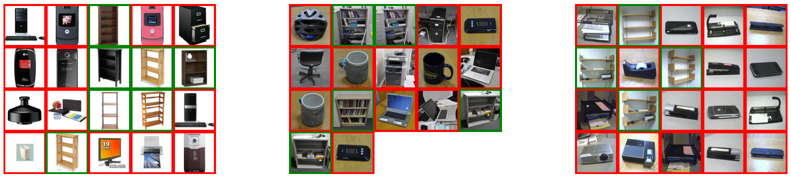


Fig. 9: Clusters corresponding to the Office31 “Bookshelf” class from Amazon, DSLR and Webcam domains.

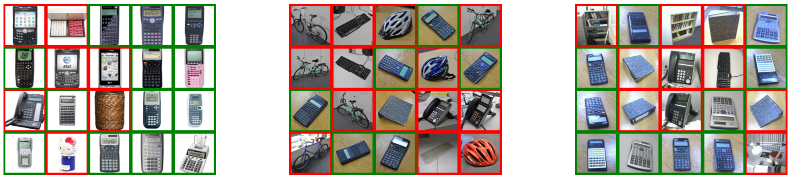


Fig. 10: Clusters corresponding to the Office31 “Calculator” class from Amazon, DSLR and Webcam domains.

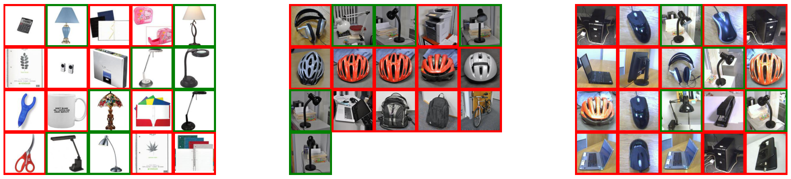


Fig. 11: Clusters corresponding to the Office31 “Desk Lamp” class from Amazon, DSLR and Webcam domains.

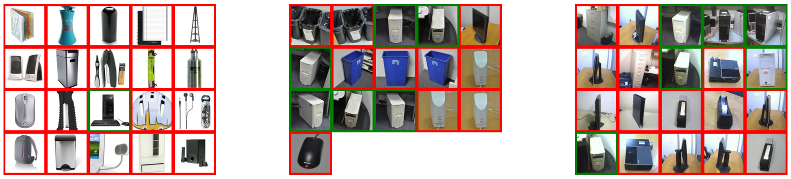


Fig. 12: Clusters corresponding to the Office31 “Desktop Computer” class from Amazon, DSLR and Webcam domains.

References

1. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Computer Vision – European Conference of Computer Vision (ECCV) 2018. pp. 139–156 (2018)
2. Ji, X., Vedaldi, A., Henriques, J.F.: Invariant information clustering for unsupervised image classification and segmentation. IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9864–9873 (2019)
3. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 5542–5550 (2017)
4. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Computer Vision – European Conference of Computer Vision (ECCV) 2010. pp. 213–226 (2010)
5. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5018–5027 (2017)