Distance-based Hyperspherical Classification for Multi-source Open-Set Domain Adaptation
Supplementary Material

Silvia Bucci* Francesco Cappio Borlino* Barbara Caputo Tatiana Tommasi
Politecnico di Torino, Italy Italian Institute of Technology
{silvia.bucci, francesco.cappio, barbara.caputo, tatiana.tommasi}@polito.it

1. Qualitative Analysis

We visualize the distribution of source and target data in the feature space (output of the contrastive head) with the t-sne [12] plots in Figure 1. In particular, we focus on the Ar.Pr.Rw → Cl case of the Office-Home dataset: the red dots represent the source domain, the blue dots are the known samples of the target domain, and the green dots the unknown ones. We take three snapshots of the data on the hyperspherical embedding: at the beginning when the backbone network is inherited from SupClr [6] pre-trained on ImageNet, immediately before the first break-point (i.e. before the application of self-training), and at the end of the training process. By observing the intermediate plot we can state that source balancing and style transfer already favor a good alignment of most of the known (blue) target classes with the respective source known clusters (red). The last plot indicates that self-training further improves the alignment while the unknown samples (green) remain in the regions among the clusters.

Randomly zooming on a known sample (the bike) and on an unknown sample (the speaker) we observe how their position change during training. The first moves from an isolated region where its top five neighbors show high class confusion, towards the correct bike class. The second starts from a neighborhood populated by several samples of classes webcam and fan and finally appears in a different region shared mostly by other instances of the class speaker.

2. Further experiments

Complete results with additional metrics In Table 1 we present the same results of the main paper including also additional metrics: the average class accuracy over known classes $OS^*$, the accuracy on the unknown class $UNK$ and the average accuracy over all classes $OS$ defined as $OS = \frac{|C_s|}{|C_s|+1} \times OS^* + \frac{1}{|C_s|+1} \times UNK$.

Robustness to temperature variation The temperature $\tau$ in the contrastive loss (main paper Eq. (1)) is kept fixed to the default value 0.07 as suggested in [11]. We verified experimentally that the results are stable even when tuning $\tau$ and remain always higher than ROS (65.3) (see Figure 2).

3. Implementation Details

We implemented HyMOS with an architecture composed of the ResNet-50 [4] backbone that corresponds to the encoder and two fully connected layers of dimension 2048 and 128 which define the contrastive head. The overall network is trained by minimizing the contrastive loss (see the main paper, Eq. (1)), setting $\tau = 0.07$ as in [11]. Our distance-based classifier lives in the hyperspherical space produced by the model, whose dimension is not constrained by the number of classes. As a consequence, the architecture remains exactly the same for all our experiments.

We initialize the backbone network with the ImageNet pre-trained SupClr model [6] and train HyMOS for 40k iterations with a balanced data mini-batch which contains one sample for each class of every source domain. The learning rate grows from 0 to 0.05 (at iteration 2500) with a linear warm-up schedule, to then decrease back to 0 at the end of training (iteration 40k) through a cosine annealing schedule. We use LARS optimizer [13] with momentum 0.9 and weight decay $10^{-6}$. For the first 20k iterations we train only on source data, using target data exclusively for the style transfer based data augmentation for the supervised contrastive learning objective. We then perform an eval step that we call self-training break-point in order to start including confident known target samples in the learning objective. We perform break-point eval steps every 5K iterations till the end of the training.

For style transfer data augmentation we use the standard VGG19-based AdaIN model with default hyperparameters [5], trained with content data from the available source domains and target samples as style data.

For what concerns the instance transformations, we ap-
HyMOS 40K

72.7 74.1 10.7 18.6
71.5 71.5 70.6
Ar,Pr,Rw
83.4 84.6 26.2 40.0
57.1 57.1 57.6 57.3
Cl,Ar,Rw

W,A

clusters. The zooms show how the neighborhood of a known (bike) and unknown (speaker) target samples change during training.

Figure 2. Sensitivity analysis for the temperature value $\tau$.

Table 1. Accuracy (%) averaged over three runs for each method on the Office31, DomainNet and Office-Home datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Office-Home</th>
<th>DomainNet</th>
<th>Office31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Al. Pr. Rw $\rightarrow$ Cl</td>
<td>OS</td>
<td>OSA*</td>
<td>UNK</td>
</tr>
<tr>
<td>Inheritable [1]</td>
<td>28.4</td>
<td>35.5</td>
<td>71.8</td>
</tr>
<tr>
<td>ROS [1]</td>
<td>69.9</td>
<td>69.8</td>
<td>76.9</td>
</tr>
<tr>
<td>CMU [3]</td>
<td>62.9</td>
<td>62.5</td>
<td>81.5</td>
</tr>
<tr>
<td>DANCE [10]</td>
<td>83.9</td>
<td>85.6</td>
<td>4.5</td>
</tr>
<tr>
<td>PGL [8]</td>
<td>83.4</td>
<td>84.6</td>
<td>26.2</td>
</tr>
<tr>
<td>Multi-Source MDSiNET [9]</td>
<td>78.4</td>
<td>79.4</td>
<td>55.0</td>
</tr>
<tr>
<td>HyMOS</td>
<td>69.5</td>
<td>69.4</td>
<td>72.7</td>
</tr>
</tbody>
</table>

Figure 2. Sensitivity analysis for the temperature value $\tau$ on Office-Home.

Applied the same data augmentations originally proposed for SimClr [2], extending them with style transfer. Specifically, we used random resized crop with scale in $\{0.08, 1\}$ and random horizontal flip. The style transfer is applied with probability $p = 0.5$ on the source images, while the remaining not-stylized images are transformed via color jittering with probability $p = 0.8$ and grayscale with probability $p = 0.2$.

The final evaluation procedure of HyMOS is summarized in Algorithm 1.

Algorithm 1 HyMOS evaluation procedure

Input: $T$; trained $Enc$ and $Proj$
Output: Predictions on $T$

procedure finalEval()
$\alpha \leftarrow$ (main paper Eq. (4))
for each $x_t$ in $T$ do
$z^t = Proj(Enc(x_t))$
$h_{y^t} \leftarrow$ nearest prototype to $z^t$
if $d_{h_{y^t}}(z^t) < \alpha$ then
$\hat{y}^t = y^*$
else
$\hat{y}^t = unknown$

procedure MAIN()
finalEval()

References
[1] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set


