2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) | 978-1-6654-0915-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/WACV51458.2022.00110

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

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

Abstract

Vision systems trained in closed-world scenarios fail when presented with new environmental conditions, new data distributions, and novel classes at deployment time. How to move towards open-world learning is a long-standing research question. The existing solutions mainly focus on specific aspects of the problem (single domain Open-Set, multi-domain Closed-Set), or propose complex strategies which combine several losses and manually tuned hyperparameters. In this work, we tackle multi-source Open-Set domain adaptation by introducing HyMOS: a straightforward model that exploits the power of contrastive learning and the properties of its hyperspherical feature space to correctly predict known labels on the target, while rejecting samples belonging to any unknown class. HyMOS includes style transfer among the instance transformations of contrastive learning to get domain invariance while avoiding the risk of negative-transfer. A self-paced threshold is defined on the basis of the observed data distribution and updates online during training, allowing to handle the known-unknown separation. We validate our method over three challenging datasets. The obtained results show that HyMOS outperforms several competitors, defining the new state-of-the-art. Our code is available at `https://github.com/silvia1993/HyMOS`.

1. Introduction

Artificial intelligent systems face many operational challenges when moving from the controlled lab environment to the real-world. First of all the annotated data available to train a model might be the result of asynchronous multi-agent collection processes. For vision tasks, this means dealing with datasets composed of labeled images that share the same class set, but with sub-groups of instances showing significant differences in appearance and style among each other. Moreover, at deployment time the learned model will inevitably encounter new environmental conditions, with distribution shift and novel classes not present during training. In standard Closed-Set domain adaptation [8], the main focus is on how to reduce the gap between the training (labeled source) and test (unlabeled target) data when the latter covers exactly the same class set of the former. Open-Set domain adaptation [35] aims at bridging the domain gap while also rejecting target samples of *unknown* classes. Indeed, in the case of category shift, the application of naïve adaptive solutions may lead to *negative-transfer* and unrecoverable class misalignment [24]. Although dealing with multiple sources is more the rule than an exception in real-world conditions, only one recent work has started to peek into the multi-source Open-Set domain adaptation task [38]. This highlights the difficulty of learning a feature space shared among domains, while also maximizing the sepa-

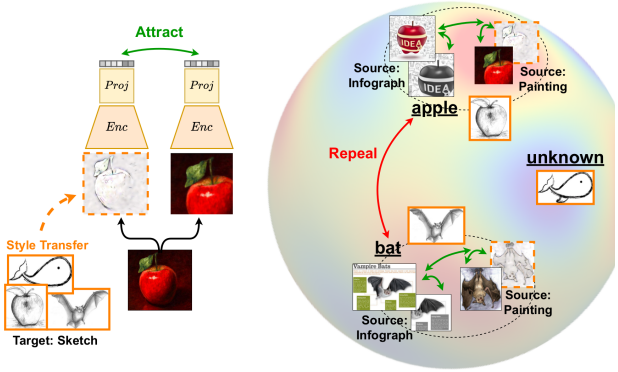

Figure 1: With HyMOS we exploit supervised contrastive learning to tackle all the challenges of multi-source Open-Set domain adaptation. We introduce style transfer in the double path contrastive logic to obtain a domain invariant representation. By balancing class and source domains in each training batch we obtain class-wise domain alignment. The learned embedding space naturally isolates target *unknown* samples in low-density regions, while the *known* samples lay close to the corresponding class cluster and can be easily involved in self-training for further adaptation.

*The authors equally contributed to this work.

978-1-6654-0915-5/21/$31.00 ©2021 IEEE
DOI 10.1109/WACV51458.2022.00110

ration between *known* and *unknown* categories within the unlabeled target.

The foundational problem that all the current open-world adaptive learning models try to solve is the limited generalization ability of the albeit powerful deep learning models. This can be at least partially explained considering two well known CNN shortcomings: (1) deep models yield features that describe mostly local rather than global statistics, which causes a bias on the image style of the training data [17]; (2) the cross-entropy loss, widely used for supervised learning, produces overconfident predictions thus biasing the model towards the labeled class set [18, 21]. Existing solutions adopt multi-stage learning procedures, combine several losses to compensate for the cross-entropy over-reliance on source supervision, and close the domain gap with adversarial techniques. The obtained approaches are difficult to train with several hyperparameters and manually set thresholds, or include complex models to generate samples of a synthetic *unknown* source class (see Table 1).

With our work, we propose a supervised model that avoids the drawbacks of the cross-entropy loss, while learning a style-invariant embedding space that naturally isolates the *unknown* categories. Specifically, we build over the very recent contrastive learning trend [6, 13, 18], where the encoder learns the invariance between two augmented views of one image (positive pair) while maximizing the distance among augmented versions of different instances (negative pair). We show how **a single supervised contrastive learning objective can tackle every challenge of multi-source Open-Set domain adaptation** (see Figure 1):

- source to source class-wise alignment comes by simply balancing data batches over classes and domains;
- source to target adaptation is obtained by first getting domain invariant features via the introduction of style transfer among the augmentations of contrastive learning. Then, a progressive and auto-regulated self-training procedure further improves the alignment between the target and the source classes clusters;
- the separation between *known* and *unknown* target data comes from a self-paced threshold based on the observed data distribution on the learned hyperspherical feature embedding. Indeed, the contrastive objective leads to compact and well separated *known* class clusters [54], leaving *unknown* samples isolated in low-density regions.

To highlight the important role of the **Hy**perspherical feature space for our **M**ulti-source **O**pen-**S**et approach, we dub it **HyMOS**. We present an extensive experimental analysis on three multi-source Open-Set datasets, showing how HyMOS outperforms current state-of-the-art methods. A thorough ablation study provides details on the internal functioning of the method. Further application to related challenging scenarios (multi-source closed set and multi-source universal) show promising results.

Method	No. of Losses	No. of HPs	Threshold
Inheritable [19]	4	2	not used - synthesize *unknown* target
ROS [2]	6	4	reject a fixed portion of Target
CMU [10]	$2 + \|\mathcal{C}_s\|$	3	validated
DANCE [41]	3	3	fixed value depending on $\|\mathcal{C}_s\|$
PGL [29]	3	4	reject a fixed portion of Target
MOSDANET [38]	$4 + \|\mathcal{S}\|$	2	validated
HyMOS	1	1	self-paced, updates online while training

Table 1: Comparison with existing open-set and universal domain adaptation approaches. HPs indicate the hyperparameters, $|\mathcal{C}_s|$ the number of source categories, $|\mathcal{S}|$ is the number of source domains. Note that synthesizing new samples is a time-consuming operation and any validation procedure requires at least a dedicated per-dataset tuning.

2. Related works

Domain Adaptation A model trained and tested on data sharing the same label set but drawn from two different marginal distributions will inevitably show low performance. *Closed-Set domain adaptation* addresses this problem by increasing the invariance of the learned features over source and target domains. Several approaches focus on minimizing statistical metrics that reflect the distribution discrepancy [60, 26, 20]. Others rely on adversarial learning [11, 27, 47]. Recent strategies also exploit batch and feature normalization [4, 22, 59] as well as self-supervision [3, 58] [1, 33] to learn robust cross-domain embeddings. A different stream of works investigates how to reduce the domain shift directly at pixel level via generative models which transfer the style of the source to the target and vice-versa [39, 43, 12, 28]. When dealing with multiple sources, the extra challenge is in aligning all the domains among each other while producing a high discriminative feature space [48] . Source weighting techniques exploit knowledge graphs and feature transferability measures evaluated once or multiple times over training [36, 61, 52].

Considering that the target is unlabeled, being sure that its semantic content perfectly matches that of the source is unrealistic. *Open-Set domain adaptation* tackles target domains which include new unknown classes with respect to the source. After the definition of the problem in [35], a first group of works proposed various approaches to maximize the separation between known and unknown target samples while exploiting adversarial-based methods to align the known classes [42, 24, 9]. Most recently, [34] introduces a self-ensembling based method to minimize the model mismatch between the class assignment proposed by the source, and the inherent target cluster distribution. ROS [2] shows how to exploit the self-supervised rotation recognition task to deal with both these objectives. In [19] a model is directly trained on the source with an extra set of negative samples produced via the suppression of class-specific feature maps activations. PGL [29] exploits a graph neural network

with episodic training to suppress the underlying conditional shift, while adversarial learning reduces the marginal shift between the source and target distributions. The only published method dealing with multi-source Open-Set is MOSDANET [38] which adds a clustering objective over a standard supervised classification model to maximize the similarity among samples of the same class but different domains. Moreover, it exploits adversarial learning for domain adaptation: it has a tailored margin loss to penalize cases with a small difference in known and unknown prediction output, and finally it includes the potential target samples in the training procedure via pseudo-labeling.

The methods dealing with *universal domain adaptation* cover a wide range of scenarios with private classes in source and/or target, including the Open-Set. In DANCE [41] a neighborhood clustering technique is integrated with the standard cross-entropy loss to learn the structure of the target, while an entropy-based score is used to align or reject the target samples. CMU [10] exploits a multi-classifier ensemble together with an unknown scoring function that combines entropy, confidence, and consistency measures.

Contrastive Learning Lately, self-supervised learning methods have shown that, by relying only on unlabeled data, it is still possible to get classification performance similar to those of the supervised approaches [45, 16, 6, 13]. Contrastive learning builds over instance discrimination techniques [56] (treating every instance as a class of its own), and aims at maximizing the agreement among multiple augmentations of the same sample, while pushing different instances far apart. Several methods have implemented this strategy by imposing the described constraints on the learned normalized embedding space. They differ in how positive and negative data pairs are sampled and stored: among the most cited, SimCLR [6] adopts a large batch size, while MoCo [13] maintains a momentum encoder and a limited queue of previous samples. The effectiveness of the contrastive self-supervised learned embeddings is generally evaluated by using the pretext feature model as starting point for a downstream supervised task. However, more direct ways to incorporate supervision are currently attracting large attention [18, 55] and show how view invariance and semantic knowledge can be combined to tackle novelty detection [44], cross-domain generalization [63] or few-shot classification [30]. Those approaches extend deep learning large margin models, demonstrating to be more robust across domains [25, 50, 32]. Current research is investigating ways to improve negative sampling in contrastive learning [7], also proposing strategies to choose the best augmentation views [46, 37].

Learning on the Unit Hypersphere Fixed-norm representations have nice properties that support deep learning computational stability and their empirical success has been demonstrated over several tasks both within- and across-domains [57, 51, 59]. In particular, [31] shows how setting class prototypes a priori on the unit hypersphere allows to free the output dimensionality from a constrained number of classes. The uniform distribution of the data centers implies large margin separation among them and leaves space to include new categories while maintaining a highly discriminative embedding. A recent work has also highlighted how learning features uniformly distributed on the unit hypersphere with compact positive pairs is a crucial component of the success of contrastive learning [54].

3. Method

To tackle multi-source Open Set domain adaptation we focus on building a robust, highly structured feature space with domain-aligned, compact, and well-separated class clusters, keeping *unknown* target samples away from the centers. We obtain this effect by minimizing the supervised contrastive loss and paying attention to how data are fed to the model. In particular: (a) we design a domain and class balanced sampling strategy for mini-batch building, which allows to obtain a perfect class-wise alignment among the sources; (b) we add style transfer to the standard semantic-preserving transformations used to create sample pairs in contrastive learning, which provides a domain invariant feature embedding; (c) we refine source-target alignment by progressively including the target domain in the learning objective through self-training; (d) we tackle *known-unknown* separation in the target domain by learning a self-paced threshold based on data distribution. We use this threshold both at inference time and when selecting *known* target samples for self-training. In the following, after an introduction on the learning framework, we discuss all the listed points in detail. An overview of HyMOS is illustrated in Figure 2 and summarized in Algorithm 1 (see the supplementary material for the eval. procedure).

Problem Framework In multi-source Open-Set domain adaptation we are given L labeled source domains $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_L\}$, where $\mathcal{S}_i = \{\boldsymbol{x}_j^{s_i}, y_j^{s_i}\}_{j=1}^{N^{s_i}} \sim p_i$, and one unlabeled target domain $\mathcal{T} = \{\boldsymbol{x}_j^t\}_{j=1}^{N^t} \sim q$, all drawn from different data distributions $p_{i=1,\dots,L}, q$. The sources share the same label set $y^s \in \mathcal{C}_s$, and it holds $\mathcal{C}_s \subset \mathcal{C}_t$, thus the target covers $\mathcal{C}_{t\setminus s}$ additional classes which are considered *unknown*. Starting from this setup, the goal is to train a model on the source data, able to identify the label of each target sample, by either assigning it to one of the *known* $\mathcal{C}_s$ classes, or rejecting it as *unknown*. Given the different relatedness levels of the target with each of the available sources, reducing the domain shift while avoiding the risk of *negative-transfer* may be difficult, especially when the *openness* $\mathbb{O} = 1 - \frac{|\mathcal{C}_s|}{|\mathcal{C}_t|}$ increases.

Contrastive Learning Formulation In self-supervised contrastive learning [6, 13], two transformed views of every

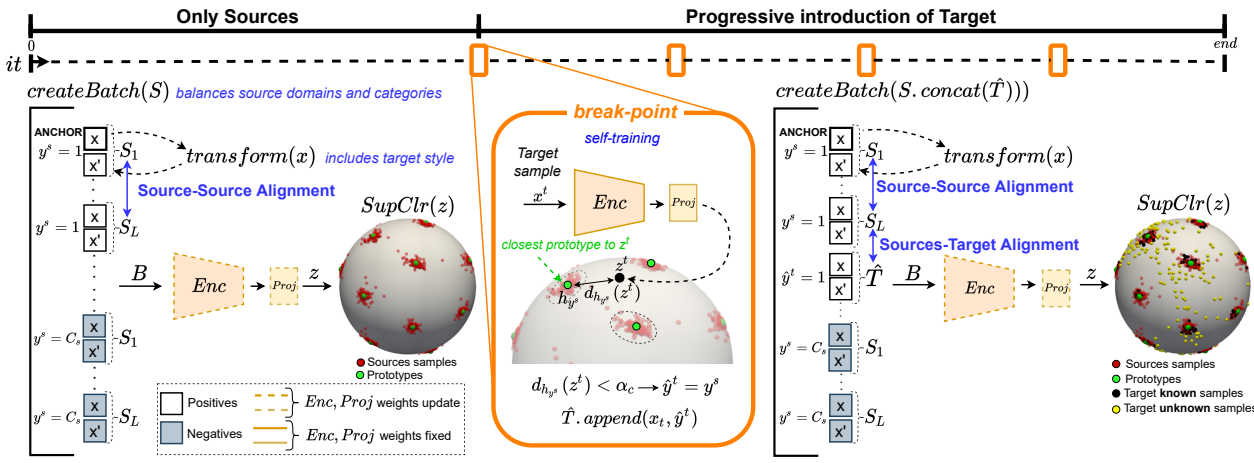

Figure 2: Schematic illustration of HyMOS (best viewed in color). We use the same notation adopted in Algorithm 1, please refer to it to follow the flow of the method.

Algorithm 1 HyMOS training procedure

Input: $\{\boldsymbol{x}^s, y^s\} \in \mathcal{S}$, $\boldsymbol{x}^t \in \mathcal{T}$, α_m, AdaIN model
Output: Enc and $Proj$

procedure TRANSFORM($\boldsymbol{x}$)
  $styleAugment = random(True, False)$
  $\boldsymbol{x}' = randomCrop(\boldsymbol{x})$
  if $styleAugment$ **then**
    return $styleTransf(\boldsymbol{x}')$ ▷ target style
  else
    return $grayScale(jitter((\boldsymbol{x}')))$

procedure CREATEBATCH(D) ▷ D: set of domains
  $batch = [\,]$ ▷ balance domains and categories
  for each y^s **in** $\{1, ..., |\mathcal{C}_s|\}$ **do**
    for each D_i **with** i **in** $\{1, ..., |D|\}$ **do**
      $\boldsymbol{x}'_{(y^s,D_i)} = transform(\boldsymbol{x}_{(y^s,D_i)})$
      $batch.append(\boldsymbol{x}_{(y^s,D_i)}, \boldsymbol{x}'_{(y^s,D_i)})$
  return $batch$ ▷ $len(batch) = |\mathcal{C}_s| \times |D| \times 2$

procedure MAIN()
  $\hat{T} = [\,]$
  for it **in** $range(0, end)$ **do**
    if it **in** $break\text{-}points$ **then**
      $\hat{T} = [\,]$
      $\alpha \leftarrow$ (Eq. 4) ; $\alpha_c = \alpha_m \cdot \alpha$
      for x_t **in** $\mathcal{T}$ **do**
        $\boldsymbol{z}^t = Proj(Enc(\boldsymbol{x}^t))$
        $h_{y^s} \leftarrow$ `closest prototype to` $\boldsymbol{z}^t$
        if $d_{h_{y^s}}(\boldsymbol{z}^t) < \alpha_c$ **then** ▷ self-training
          $\hat{y}^t = y^s$; $\hat{T}.append(\boldsymbol{x}^t, \hat{y}^t)$
    $B = createBatch(\mathcal{S}.concat(\hat{T}))$
    $\boldsymbol{z} = Proj(Enc(B))$
    $loss = SupClr(\boldsymbol{z})$ (Eq. 1)
    Update $Enc, Proj \leftarrow \nabla loss$

input image are propagated through a CNN network. The views are obtained via standard augmentation strategies as grayscale, random crop, and color jittering. For each sample $\{\boldsymbol{x}^s_k, y^s_k\}$ in the double batch $B = \{k = 1, \ldots, 2K\}$ the features obtained via the encoder $Enc(\boldsymbol{x}^s_k)$ enter the final contrastive head that further projects them to a normalized embedding, producing $\boldsymbol{z}^s_k = Proj(Enc(\boldsymbol{x}^s_k))$. On the obtained hyperspherical space the samples are compared among each other: the similarity between two augmented views of the same instance is maximized, while the similarity of two different instances is minimized.

When the image labels are available the sample comparison can be performed both instance-wise, as in the self-supervised case, and class-wise [18]: every samples of the same class y^s_k are considered as positives, while the negative pairs are composed by samples of different classes. We indicate with $\nu(k) = B \setminus \{k\}$ the double batch without the *anchor* sample of index k, and the positive pairs are $\pi(k) = \{k' \in \nu(k) : y^s_{k'} = y^s_k\}$. Thus, the supervised contrastive loss is [18]:

$$\mathcal{L}_{SupClr} = \sum_{k=1}^{2K} \frac{-1}{|\pi(k)|} \sum_{k' \in \pi(k)} \log \frac{\exp(\sigma(\boldsymbol{z}^s_k, \boldsymbol{z}^s_{k'})/\tau)}{\sum_{n \in \nu(k)} \exp(\sigma(\boldsymbol{z}^s_k, \boldsymbol{z}^s_n)/\tau)}, \tag{1}$$

where $\tau \in \mathbb{R}^+$ is the temperature, and $\sigma(\cdot, \cdot)$ is the cosine similarity.

(a) HyMOS Source-Source Class-Wise Domain Alignment The supervised contrastive loss aims at learning compact class clusters with large margins. This ability can be exploited to perform source-source class-wise domain alignment by composing each training mini-batch with samples coming from different visual domains. We evenly divide each batch to cover all the $|\mathcal{C}_s|$ classes, and for each class, we select an equal number of samples from all the L

source domains. The loss in Eq. (1) does the rest, providing an embedding space where samples of the same class are concentrated in the same region regardless of the domain, while different classes are far apart from each other.

(b) HyMOS Source-Target Style Invariance The image transformations used in contrastive learning are meant to let the model focus on core semantic information while making it invariant to the irrelevant cues that they introduce. When dealing with data from different domains we desire a representation able to neglect major differences in visual appearance that go beyond mild grayscale or color jittering. This calls for dedicated semantic-preserving image transformations. We propose an augmentation based on style transfer as it is perfectly suitable for this goal: it does not affect the image content while changing significantly the global image texture. In particular, we use the AdaIN [15] model trained jointly on source and target data in order to transfer the style from target images into source ones. As this augmentation is applied randomly, the loss function will explicitly compare original source images with target-like ones and learn to ignore the style difference.

We highlight that our approach to obtain style invariance is safe from *negative-transfer*. This is one of the main issues in Open-Set domain adaptation due to the risk of aligning *unknown* target categories to the *known* ones of the source. All existing methods [2, 10, 41, 62, 24] try to mitigate this problem by avoiding the inclusion of unknown samples or down-weighting them in the adaptation process. Thus, they are forced to identify the unknown samples before learning the domain invariant model by designing complex strategies. With style transfer, instead, we learn a domain agnostic representation since the beginning of the training process: it disregards the semantic content of the target so we can draw the style also from samples belonging to *unknown* categories without incurring in *negative-transfer*.

(c) HyMOS Adaptation Refinement via Self-Training In order to get a perfect source-target alignment, it would be enough to include the target data as an additional source domain while training the supervised contrastive model with the strategies described above. Of course this is not possible as class labels are not available for target samples. However, once the model trained on source data and including target style invariance is robust enough, one could use it to produce pseudo-labels for target data by simply exploiting its predictions. We choose exactly this approach: after an initial source-only learning episode, we start progressively integrating the target samples in our learning objective, by passing through evaluation steps that we call *self-training break-points*, which allow us to select confident *known* target samples. Through this iterative technique, we propagate label knowledge from source to target data, improving the compactness of our class clusters while progressively leaving *unknown* target data in low-density regions of the hyperspherical feature space.

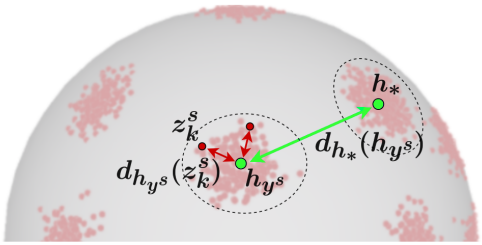

Figure 3: Illustration of the distances used to set the class prediction and the self-training procedure.

(d) HyMOS Known-Unknown Separation and Classification on the Hypersphere The obtained embedding, with well clustered *known* categories separated by large margins and *unknown* samples in isolated areas, provides the ideal condition to perform distance-based classification. Differently from previous literature [18, 44] where the contrastive models were used only as pretext tasks and the projection head was later dropped in favor of a standard cross-entropy loss, we propose to stay on the hypersphere while delivering the final predictions. We define the prototype of each source class y^s by computing the corresponding feature average $h_{y^s} = \frac{1}{N_{y^s}} \sum_{k \in y^s} \boldsymbol{z}_k^s$, re-projected on the unit hypersphere. For any target sample $\boldsymbol{z}^t$ we measure the cosine similarity to each source class prototype and we rescale it in $[0, 1]$ to define the distance $d_{h_{y^s}}(\boldsymbol{z}^t) = \{1 - \sigma_{[0,1]}(\boldsymbol{z}^t, h_{y^s})\}$ for $y^s \in \{1, \dots, |\mathcal{C}_s|\}$, which is used as confidence measure for label assignment.

To decide whether a sample belongs to a *known* category we need a threshold on the distance from the *known* class prototypes. How to define such a threshold is a widely discussed problem in the Open Set literature, with many methods choosing values a priori and keeping them fixed while training [10, 41]. Instead, we propose to directly extract it from the observed data distribution, obtaining a value that changes online during the learning process. Specifically we introduce two metrics to evaluate it: the *class sparsity*:

$$\theta = \frac{1}{|C_s|} \sum_{y^s \in C_s} d_{h_*}(h_{y^s}) \,, \tag{2}$$

where h_* is the closest prototype to each h_{y^s}, and the *class compactness*:

$$\phi = \frac{1}{|C_s|} \sum_{y^s \in C_s} \left\{ \frac{1}{N_{y^s}} \sum_{k \in y^s} d_{h_{y^s}}(z_k^s) \right\} . \tag{3}$$

In words, the former collects the prototype-to-prototype minimal distances and provides a measure of inter-class separation, while the latter evaluates whether the samples of each class are tight around the corresponding prototype (see Figure 3). A dataset with a large number of categories, each with small intra-class variability, results in a feature

scenario with high compactness but low sparsity, for which a low threshold is needed. On the other extreme, a dataset with a limited number of categories showing large intra-class variability corresponds to low compactness and high sparsity condition for which we can allow a higher threshold. We compute our threshold by:

$$\alpha = \phi \cdot \left[log\left(\frac{\theta}{2\phi}\right) + 1 \right], \tag{4}$$

where $\theta/2\phi$ estimates the average ratio between the distance of two adjacent prototypes and the radii of the respective clusters. The use of the threshold at inference time is straightforward:

$$\hat{y}^t = \begin{cases} \arg\min_{y^s}(d_{h_{y^s}}(\boldsymbol{z}^t)) & \text{if } \min_{y^s}(d_{h_{y^s}}(\boldsymbol{z}^t)) < \alpha \\ \texttt{unknown} & \text{if } \min_{y^s}(d_{h_{y^s}}(\boldsymbol{z}^t)) \geq \alpha\,. \end{cases} \tag{5}$$

We exploit this threshold also for the self-training breakpoints described before. Only in this phase we are particularly cautious , thus we include a multiplier α_m that allows us to keep a more conservative threshold: $\alpha_c = \alpha_m \cdot \alpha$. This multiplier can be kept fixed to 0.5 and it is the only hyperparameter of HyMOS.

4. Experiments

We implemented HyMOS with a ResNet-50 [14] backbone and two fully connected layers which define the contrastive head. All the implementation details as well as the Pytorch code are provided in the supplementary material.

Datasets We evaluate our approach on three image classification benchmarks, following the same setting used in [38], with one domain considered in turn as target. **Office31** [40] comprises three domains: Webcam (W), Dslr (D) and Amazon (A) each containing 31 object categories. We set as known the first 20 classes in alphabetic order, while the remaining 11 are unknown. **Office-Home** [49] is made by four domains: Art (Ar), Clipart (Cl), Product (Pr), RealWorld (Rw) with 65 classes. The first 45 categories in alphabetic order are known, and the remaining 20 are unknown. **DomainNet** [36] is a more challenging testbed than the previous ones. It contains six domains and 345 classes. As in [38], we consider Infograph (I), Painting (P), Sketch (S) and Clipart (C), selecting randomly 50 samples per class or using all the images in case of lower cardinality. The first 100 classes in alphabetic order are known, while the remaining 245 are unknown.

Results We compare HyMOS with several state-of-the-art baselines proposed for single-source Open-Set (Inheritable [19], ROS [2], PGL [29]), multi-source Open-Set (MOSDANET [38]) and universal domain adaptation (CMU [10], DANCE [41]). We use the code provided by the authors[1], and for all the methods that do not specify how to manage multiple sources, we apply the *Source Combine* strategy [36] that considers the union of all the source data in a single domain. We adopt the HOS metric, defined in [2, 10] for a fair evaluation of Open-Set methods: it is the harmonic mean between the average class accuracy over the known classes *OS** and the accuracy over the unknown class *UNK*: $HOS = 2\frac{\text{OS}^* \times \text{UNK}}{\text{OS}^* + \text{UNK}}$.

Table 2 collects the obtained results showing how HyMOS outperforms all the baselines. The gain of HyMOS with respect to the best competitor ROS, varies from 1.9% points on OfficeHome, up to 10.8% on DomainNet. Besides being simpler than the reference approaches, HyMOS shows to be robust to the significantly different scenarios covered by the three datasets in terms of number of shared and private classes, as well as nature and extent of the domain gaps. These peculiarities make HyMOS the most suitable approach in a variety of real-world applications.

We also benchmark HyMOS with the best competitor ROS in terms of the $AUROC$ (Area under the Receiver Operating Characteristic curve) metric, which has the advantage of being threshold-independent. In our case, the *normality score* used to evaluate whether a sample is *known* or *unknown* is its distance from the nearest source class prototype, while ROS exploits a combination of entropy and probability output of an auxiliary rotation recognition classifier. Even in this case, HyMOS outperforms ROS, reflecting what is already observed in terms of HOS. This also confirms how well *known* and *unknown* samples are separated in the learned hyperspherical embedding space.

Analysis on the Threshold For HyMOS we designed a self-paced procedure that learns the dynamic threshold α from the data distribution. Figure 4 (left) provides an overview of α at different training iterations : for Office31 and Office-Home the threshold decreases over time while for DomainNet it increases. These variations evidence how the data clusters move: as the training proceeds they become more compact and the reciprocal distance increases towards a more uniform class distribution on the hypersphere. For DomainNet the second event occurs faster than the first: this trend is correlated with the class cardinality which is higher with respect to that of other datasets. In all the cases, the threshold converges to a stable value.

The α_m multiplier used at training time to compute a conservative threshold is the only hyperparameter of HyMOS: by modifying it one could decide to favor recognition of *known* classes at the expense of a lower *unknown* recognition during training. The results in Table 3 show that α_m=0.5 is a safe choice regardless of the dataset. Moreover, by tuning this multiplier, the HOS performance of HyMOS remains always competitive with ROS, and can even

[1]For all the baseline methods the implementations are publicly available, with the only exception of MOSDANET [38] for which we obtained the code via private communications with the authors.

	Method		DomainNet			Office31				Office-Home				
			→ S	→ C	**Avg.**	→ W	→ D	→ A	**Avg.**	→ Rw	→ Cl	→ Ar	→ Pr	**Avg.**
HOS	Source Combine	Inheritable [19]	34.8	44.0	39.4	76.6	79.5	70.0	75.4	63.2	52.6	48.7	60.7	56.3
		ROS [2]	44.5	52.4	48.5	81.8	80.1	64.7	75.5	**73.0**	57.3	61.6	69.1	65.3
		CMU [10]	38.1	35.5	36.8	61.4	64.0	56.4	60.6	70.8	50.0	58.1	69.3	62.1
		DANCE [41]	30.0	37.6	33.8	38.5	59.7	58.0	52.0	12.4	16.1	18.6	22.9	17.5
		PGL [29]	18.5	19.4	19.0	43.3	37.7	35.6	38.9	40.0	31.5	31.8	42.2	36.4
	Multi-Source	MOSDANET [38]	40.0	39.3	39.6	60.5	71.5	**73.9**	68.6	65.0	51.1	54.3	65.9	59.1
		HyMOS	**57.5**	**61.0**	**59.3**	**90.2**	**89.9**	60.8	**80.3**	71.0	**64.6**	**62.2**	**71.1**	**67.2**
AUROC	Source Combine	ROS [2]	63.9	68.0	66.0	93.9	95.2	**73.5**	87.5	80.8	69.6	73.7	79.4	75.9
	Multi-Source	**HyMOS**	**71.9**	**75.8**	**73.9**	**96.9**	**96.1**	71.0	**88.0**	**81.1**	**76.4**	**75.3**	**79.6**	**78.1**

Table 2: Results averaged over three runs for each method on the DomainNet, Office31, and Office-Home datasets.

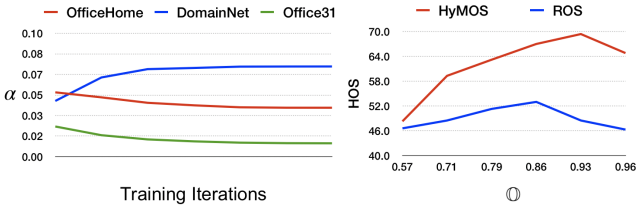

Figure 4: Left: analysis on the dynamic threshold α at different training iterations. Right: performance of HyMOS and ROS [2] at different openness ($\mathbb{O}$) levels.

Method		DomainNet	Office31	Office-Home
HyMOS	$\alpha_m = 0.3$	55.1	79.2	65.8
	$\alpha_m = 0.5$	59.3	**80.3**	**67.2**
	$\alpha_m = 0.7$	60.8	78.2	66.8
	$\alpha_m = 1.0$	**61.4**	74.1	65.8
ROS [2]		48.5	75.7	65.3

Table 3: Average performance (HOS) when changing the train-time multiplier α_m to the self-paced threshold α.

increase as in the case of DomainNet for α_m=1.

Increasing the Openness Level In real-world conditions, it is difficult to have direct control over the number of *unknown* classes in the unlabeled target, and it is natural to expect more *unknown* categories than *known* ones. To study how HyMOS reacts at different openness levels, we consider the DomainNet dataset and exploit its large class cardinality. The plot in Figure 4 (right) shows the HOS accuracy of HyMOS and how it outperforms its best competitor ROS at different openness values $\mathbb{O} \in \{0.5, 1\}$.

5. Ablation Analysis

We designed HyMOS to be straightforward while keeping in mind all the challenges of multi-source Open-Set domain adaptation. In the following we focus on each of them, providing a detailed ablation that sheds light on the inner functioning of our method. The results are in Table 4.

Source-Source Alignment Reducing the domain shift among the available sources improves model generalization. This aspect is largely discussed in multi-source Closed-Set domain adaptation literature [61, 52]. A dedicated source alignment component is also included in the only existing multi-source Open-Set method MOSDANET.

Method	Office-Home				
	→ Rw	→ Cl	→ Ar	→ Pr	**Avg.**
HyMOS	71.0	64.6	62.2	71.1	**67.2**
w/o Source Balance	69.2	58.4	60.6	70.2	64.6
Style Tr. Target Known (Oracle)	70.7	63.7	62.5	71.2	67.0
w/o Style Transfer	69.5	56.4	60.0	68.3	63.6
w/o Self-Training	72.2	55.0	58.6	71.5	64.3
Improved Cross-Entropy	61.5	61.2	58.1	57.1	59.5
ROS [2]	73.0	57.3	61.6	69.1	65.3
+ Source Balance	75.2	55.5	62.6	66.9	65.0
+ Style Transfer	62.6	46.3	52.0	60.1	55.2
+ Self-Training	69.6	59.1	61.5	60.5	62.7
+ S. Balance, Style Tr., Self-Train.	62.0	40.4	52.2	62.4	54.3

Table 4: Ablation Study, HOS results.

HyMOS obtains cross-source adaptation by combining the supervised contrastive learning loss with an accurately designed batch sampling strategy: each training mini-batch contains one sample for each class and for each domain. The supervised contrastive loss provides a strong class-wise alignment by pulling together samples of the same class and pushing away samples of different classes regardless of the domain. HyMOS shows a gain in performance of 2.6% over its version without this balancing (row *w/o Source Balance*).

Source-Target Adaptation In HyMOS, both the style transfer augmentation and the auto-regulated self-training procedure contribute to aligning source and target without incurring the risk of *negative-transfer*. By adding target style transfer as one of the source augmentations we push the model to focus on domain agnostic visual characteristics without involving semantic content from the target. To evaluate the effect of this addition we present two ablation cases. We compare our method with an Oracle version where the target style is extracted only from *known* categories (line *Style Tr. Target Known (Oracle)*), and we conclude that HyMOS is not harmed when using the whole target for this adaptation step. Moreover, we deactivate style transfer (row *w/o Style Transfer*) causing a performance drop of 3.6%, which shows its important role in HyMOS.

Multi-Source Closed-Set								Multi-Source Universal			
Method	→ clp	→ inf	→ pnt	→ qdr	→ rel	→ skt	**Avg.**	Method	→ S	→ C	**Avg.**
Source Only [23]	52.1	23.4	47.7	13.0	60.7	46.5	40.6	CMU [10]	38.9	31.2	35.1
LtC-MSDA [53]	63.1	28.7	56.1	16.3	66.1	53.8	47.4	DANCE [41]	44.5	49.9	47.2
DRT [23]	71.0	31.6	**61.0**	12.3	71.4	60.7	51.3	ROS [2]	39.7	46.0	42.9
HyMOS	**71.5**	**41.8**	60.8	**34.5**	**74.2**	**66.6**	**58.2**	**HyMOS**	**54.6**	**57.1**	**55.9**

Table 5: Multi-Source Closed-Set (Accuracy) and Universal Domain Adaptation (HOS) performance on DomainNet.

Finally, a strong feature-level class-wise source-target alignment is obtained thanks to the self-training procedure, which selects confident target known samples (closest to the source class prototypes) and includes them in the learning objective. The gain of HyMOS with respect to its version without this strategy is 2.9% (row *w/o Self-Training*).

Comparison with an Improved Cross-Entropy Baseline Source balance, style transfer, and self-training appear as simple strategies that can be combined with any supervised learning model to improve its effectiveness in the multi-source Open-Set scenario. Still, we state that leveraging on supervised contrastive learning and its related hyperspherical embedding is crucial for the task at hand. To support our claim we substitute the contrastive loss of HyMOS with the standard cross-entropy loss. The row *Improved cross-entropy* reports the obtained results, showing that this baseline approach is significantly worse than HyMOS.

Comparison with an improved version of ROS [2] We also enriched our best competitor ROS with source balancing, style transfer, and self-training.

In the bottom part of Table 4, the + *Source Balance* row indicates that organizing the training data batches so that they contain a balanced set of categories and source domains does not provide an improvement with respect to the standard version of ROS. The source-to-source alignment visible for HyMOS does not appear here: indeed the cross-entropy loss does not induce the same inherent clustering and adaptation effect that can be obtained via contrastive learning. The row + *Style Transfer* shows a low performance for ROS when using this augmentation. By checking the predictions we observe a slight advantage in the recognition accuracy of the *known* classes, but a significant drop in the *unknown* accuracy which causes a decrease in the overall result. We also followed [38] to extend ROS with self-training. The corresponding row + *Self-Training* shows again a drop in performance: this procedure tends to propagate the recognition errors due to the cross-entropy overconfidence. Indeed self-training may induce a dangerous model drift, but recent literature has shown that its effectiveness and safe nature hold when the sample selection is performed with a self-pacing strategy based on the distribution of the unlabeled samples [5], exactly as in HyMOS.

Finally, when applying all the strategies at once, the results are similar to those obtained with style transfer alone. This last technique clearly steered the whole method towards a low performance.

6. Extension to Closed-Set and Universal

HyMOS can be easily extended to the simpler multi-source Closed-Set domain adaptation setting (perfect overlap between sources and target classes) and to the more challenging multi-source Universal domain adaptation case (both sources and target have their own private categories). We consider the DomainNet dataset and run an evaluation on those two scenarios, following [23] for Closed-Set and [10] for Universal. In the latter, sources and target share the first 150 classes in alphabetic order, the next 50 categories are sources private classes and the rest are target private classes. For Closed-Set we use as reference LtC-MSDA [53] and DRT [23] which leverage respectively on a graph connecting domain prototypes, and on a dynamic transfer that updates the model parameters on a per-sample basis. Table 5 collects the results and show how HyMOS gets promising performance with respect to several state-of-the-art methods in the two scenarios.

7. Conclusions

In this paper we introduced HyMOS, a straightforward approach for multi-source Open-Set domain adaptation. It exploits contrastive learning and the inherent properties of its hyperspherical feature space to avoid the limitations of the existing competing methods. HyMOS includes a tailored data balancing to enforce cross-source alignment and introduces style transfer among the instance transformations for source-target adaptation, keeping away from the risk of negative transfer. Finally, a self-training strategy refines the model without the need for manually set thresholds. Through extensive experiments, we demonstrated state-of-the-art results on three benchmarks and we delved into the details of the methods with several quantitative evaluations which shed light on its internal functioning. The application to the multi-source closed-set and universal scenario confirm the effectiveness of HyMOS, identifying it a good starting approach towards life-long learning for real-world applications.

Acknowledgements. This work was partially supported by the ERC project RoboExNovo. Computational resources were provided by IIT (HPC infrastructure). We also thank Biplab Banerjee for the discussions on MOSDANET [38].

References

[1] Silvia Bucci, Antonio D'Innocente, and Tatiana Tommasi. Tackling Partial Domain Adaptation with Self-supervision. In *ICIAP*, 2019.

[2] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation. In *ECCV*, 2020.

[3] Fabio M. Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019.

[4] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulo. Multidial: Domain alignment layers for (multisource) unsupervised domain adaptation. *IEEE TPAMI*, 2020.

[5] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *AAAI*, 2021.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[7] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *NeurIPS*, 2020.

[8] Gabriela Csurka. *Domain Adaptation in Computer Vision Applications*. Springer, 2017.

[9] Qianyu Feng, Guoliang Kang, Hehe Fan, and Yi Yang. Attract or distract: Exploit the margin of open set. In *ICCV*, 2019.

[10] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *ECCV*, 2020.

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[12] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *CVPR*, 2019.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[16] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020.

[17] Simon Jenni, Hailin Jin, and Paolo Favaro. Steering self-supervised feature learning beyond local pixel statistics. In *CVPR*, 2020.

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

[19] Jogendra Nath Kundu, Naveen Venkat, Ambareesh Revanur, R Venkatesh Babu, et al. Towards inheritable models for open-set domain adaptation. In *CVPR*, 2020.

[20] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, 2019.

[21] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2017.

[22] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. In *ICLR*, 2017.

[23] Yunsheng Li, Lu Yuan, Yinpeng Chen, Pei Wang, and Nuno Vasconcelos. Dynamic transfer for multi-source domain adaptation. In *CVPR*, 2021.

[24] Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *CVPR*, 2019.

[25] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

[26] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[27] Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018.

[28] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Adversarial style mining for one-shot unsupervised domain adaptation. In *NeurIPS*, 2020.

[29] Yadan Luo, Zijian Wang, Zi Huang, and Mahsa Baktashmotlagh. Progressive graph learning for open-set domain adaptation. In *ICML*, 2020.

[30] Orchid Majumder, Avinash Ravichandran, Subhransu Maji, Marzia Polito, Rahul Bhotika, and Stefano Soatto. Revisiting contrastive learning for few-shot classification. *arXiv preprint arXiv:2101.11058*, 2021.

[31] Pascal Mettes, Elise van der Pol, and Cees Snoek. Hyperspherical prototype networks. In *NeurIPS*, 2019.

[32] Benjamin J. Meyer and Tom Drummond. The importance of metric learning for robotic vision: Open set recognition and active learning. In *ICRA*, 2019.

[33] Yu Mitsuzumi, Go Irie, Daiki Ikami, and Takashi Shibata. Generalized Domain Adaptation. In *CVPR*, 2021.

[34] Yingwei Pan, Ting Yao, Yehao Li, Chong-Wah Ngo, and Tao Mei. Exploring category-agnostic clusters for open-set domain adaptation. In *CVPR*, 2020.

[35] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *ICCV*, 2017.

[36] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

[37] Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. In *NeurIPS*, 2020.

[38] Sayan Rakshit, Dipesh Tamboli, Pragati Shuddhodhan Meshram, Biplab Banerjee, Gemma Roig, and Subhasis Chaudhuri. Multi-source open-set deep adversarial domain adaptation. In *ECCV*, 2020.

[39] Paolo Russo, Fabio Maria Carlucci, Tatiana Tommasi, and Barbara Caputo. From source to target and back: symmetric bi-directional adaptive gan. In *CVPR*, 2018.

[40] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[41] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self-supervision. In *NeurIPS*, 2020.

[42] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *ECCV*, 2018.

[43] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, 2018.

[44] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *NeurIPS*, 2020.

[45] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020.

[46] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *NeurIPS*, 2020.

[47] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[48] Naveen Venkat, Jogendra Nath Kundu, Durgesh Singh, Ambareesh Revanur, and Venkatesh Babu R. Your classifier can secretly suffice multi-source domain adaptation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *NeurIPS*, 2020.

[49] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

[50] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[51] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *ACM Multimedia*, 2017.

[52] Hang Wang, Minghao Xu, Bingbing Ni, and Wenjun Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *ECCV*, 2020.

[53] Hang Wang, Minghao Xu, Bingbing Ni, and Wenjun Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *ECCV*, 2020.

[54] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

[55] Longhui Wei, Lingxi Xie, Jianzhong He, Jianlong Chang, Xiaopeng Zhang, Wengang Zhou, Houqiang Li, and Qi Tian. Can semantic labels assist self-supervised visual representation learning? *arXiv preprint arXiv:2011.08621*, 2020.

[56] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[57] Jiacheng Xu and Greg Durrett. Spherical latent spaces for stable variational autoencoders. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *EMNLP*, 2018.

[58] Jiaolong Xu, Liang Xiao, and Antonio Lopez. Self-supervised domain adaptation for computer vision tasks. *IEEE ACCESS*, 7:156694–156706, 2019.

[59] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, 2019.

[60] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *CVPR*, 2017.

[61] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. In *ECCV*, 2020.

[62] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *CVPR*, 2019.

[63] Yifan Zhang, Bryan Hooi, Dapeng Hu, Jian Liang, and Jiashi Feng. Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning. *arXiv preprint arXiv:2102.06605*, 2021.