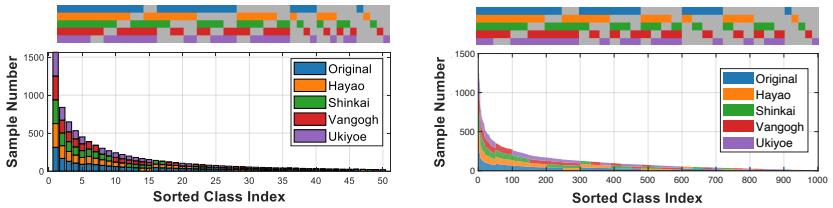
1 Experiment Settings

1.1 Datasets

AWA2-LTS and ImageNet-LTS: These two datasets were modified from the well-established **AWA2** [7] and **ImageNet-LT** [3] datasets. The former (50 categories) is a benchmark for zero-shot learning, whilst the latter (1000 categories) has been widely used for long-tailed visual recognition. To accommodate our practical needs for benchmarking **LT-DS**, the following steps were applied.

First of all, for **ImageNet-LT**, the training/validation/testing splits are defined in [3], where the training subset is long-tailed distributed whilst validation and testing ones are balanced sampled. Our further processing of each subset was based on the original splits. For **AWA2**, we randomly extracted 50 samples of each category for testing, and 30 samples for validation. Since the training data of **AWA2** is not long-tailed originally, we performed random sampling to convert the training subset of **AWA2** to a long-tailed version with the maximum number 1565, and the minimum number 20, following $n_c = \left\lfloor n_{max} \exp(-\frac{\sqrt{c-1}}{7} \times \log \frac{n_{min}}{n_{max}}) \right\rfloor$, where c refers the sorted class index, starting from 1 to the class number $C = 50$.

Subsequently, for the training data, each image was assigned a style randomly selected from the five ones (i.e., Original, Hayao, Shinkai, Vangogh, Ukiyoe). To simulate the realistic scenario where head classes are common across domains whilst non-head classes appear in only certain specific domains due to their low-frequency, we deliberately reduced the number of domain candidates for those non-head classes. The categorical distributions of the training subset in each domain are shown in Figs. 1(a) and 1(b).



(a) **AWA2-LTS** (b) **ImageNet-LTS**

Fig. 1. Categorical Distribution of Training Subset of Each Domain. Left: **AWA2-LTS**; Right: **ImageNet-LTS**. The bottom row shows the categorical distribution across domains and categories, whereas the top row presents the existence of classes in the training subset of each domain (grey indicates non-existence).

Regarding the validation subset, after assigning each image with a style with equal probability, we only kept those seen classes in the training subset of each individual domain. This ensures that the training label set and validation label

set of each domain are totally the same. Regarding the testing subset, each image was assigned with a style randomly picked from five styles, and all classes were kept, so that all the classes are seen in each domain.

In total, for **AWA2-LTS**, totally 50 classes and 5 domains exist, whereas 1000 classes and 5 domains for **ImageNet-LTS**. Detailed instructions for generating the proposed **LT-DS** datasets and the indexes of corresponding training/validation/testing splits for all images for benchmarking can be found at `https://github.com/guxiao0822/LT-DS/tree/main/dataset`.

1.2 Implementation Details

Architectures: For **AWA2-LTS** and **ImageNet-LTS**, we applied the ResNet-10 as the feature extractor f, and a fully connected (FC) layer as the classifier h. For d and e, they are composed of a FC , BatchNorm, and ReLU layer. The whole network was randomly initialized without applying pretrained weights, so as to avoid the overlap of classes between our proposed datasets and the original ImageNet.

Training Details: We utilized SGD for optimization. The random seed was set as 0 for reproduction purposes. The parameters used in this paper was listed as in Table 1. In addition, β_2 were decayed by 0.1 after 40 and 80 epochs.

Table 1. Hyperparameters (HP) in our experimental settings.

HP	Description	AWA2-LTS	ImageNet-LTS
β_1	meta-train learning rate	0.2	0.2
β_2	final learning rate	0.1	0.1
k	top k similarity	5	5
λ	augmentation intensity	5	5
B	batch size of each domain	48	64
α	margin of contrastive loss	0.1	0.1
τ	temperature scaling constant	1/30	1/50
T_{max}	maximum training step (corresponding epoch number)	100	100
T_{Σ}	covariance tracking milestone step (corresponding epoch number)	40	40
w_1	weight of $\mathcal{L}_{Z2S}$	0.1	0.1
w_2	weight of $\mathcal{L}_{S2S}$	0.1	0.1
w_3	weight of $\mathcal{L}_{S2Z}$	0.1	0.1
w_4	weight of $\mathcal{L}_{Aug}$ and $\mathcal{L}_{MAug}$	0.1	0.1
w_{mte}	weight of $\mathcal{L}_{mte}$	0.3	0.3

For the compared methods, we used the same backbone for fair comparison, with the hyperparameter settings adopted in their original implementations.

2 Equation Proof

2.1 Distribution Calibrated Classification Loss

The distribution calibrated classification loss aims to calibrate the classification loss to a balanced category distribution, so that meta-train and meta-test both

aim to achieve ideal performance on balanced distributions.

$$\mathcal{L}_{dc}(\boldsymbol{x}_i, y_i, d_i; f, h) = -\log \frac{n_{y_i}^{d_i} \exp\left([h \circ f(\boldsymbol{x}_i)]_{y_i}\right)}{\sum_{c=1}^{C} n_c^{d_i} \exp([h \circ f(\boldsymbol{x}_i)]_c)}. \tag{1}$$

Without loss of generality and for simplicity, we denote the logits of class i as $\gamma_i = [h \circ f(\boldsymbol{x})]_i$. The probability of class ϕ_i after softmax normalization is $\frac{\exp(\gamma_i)}{\sum_c \exp(\gamma_c)}$.

Based on Bayesian theorem, the probability ϕ_i^d in domain d is formulated as below,

$$\phi_i^d = p^d(y = i|f(\boldsymbol{x})) = \frac{p^d(f(\boldsymbol{x})|y = i)p^d(y = i)}{p^d(f(\boldsymbol{x}))}. \tag{2}$$

For another domain d',

$$\phi_i^{d'} = \frac{p^{d'}(f(\boldsymbol{x})|y = i)p^{d'}(y = i)}{p^{d'}(f(\boldsymbol{x}))}. \tag{3}$$

Considering our visual-semantic mapping functional blocks, we assume that the term $\frac{p^d(f(\boldsymbol{x})|y=i)}{p^d(f(\boldsymbol{x}))}$ is identical across domains. Therefore,

$$\phi_i^{d'} = \phi_i^d \frac{p^{d'}(y = i)}{p^d(y = i)}, \tag{4}$$

$$= \frac{\exp(\gamma_i)}{\sum_c \exp(\gamma_c)} \frac{n_i^{d'} / \sum_c n_c^{d'}}{n_i^d / \sum_c n_c^d}, \tag{5}$$

$$= \frac{\sum_c n_c^d}{\sum_c n_c^{d'} \sum_c \exp(\gamma_c)} \frac{n_i^{d'} \exp(\gamma_i)}{n_i^d}. \tag{6}$$

Based on the property that the summed probability over all classes equals to one, we can derive

$$\sum_{i=1}^{C} \phi_i^{d'} = \frac{\sum_c n_c^d}{\sum_c n_c^{d'} \sum_c \exp(\gamma_c)} \sum_{i=1}^{C} \frac{n_i^{d'} \exp(\gamma_i)}{n_i^d} = 1. \tag{7}$$

Then, based on Equations (6) and (7), it can be derived that

$$\phi_i^{d'} = \frac{\frac{n_i^{d'}}{n_i^d} \exp(\gamma_i)}{\sum_c \frac{n_c^{d'}}{n_c^d} \exp(\gamma_c)}. \tag{8}$$

In our case, d' is recognized as a training domain with imbalanced distribution and d as a testing domain with balanced distribution. Since n_c^d are equal across classes, the probability of class i in d' can be rewritten as below,

$$\phi_i^{d'} = \frac{n_i^{d'} \exp(\gamma_i)}{\sum_c n_c^{d'} \exp(\gamma_c)}. \tag{9}$$

Till now, Equation (1) is sorted. It should be noted that we do not calibrate the classification loss from meta-train domain category distributions to meta-test distributions. This is because each individual domain exists unseen classes (impossible to be rebalanced), and however our final goal is to achieve good performance over all classes. Instead, for both meta-train and meta-test, we aim to calibrate the classification loss to a balanced distribution.

2.2 Augmentation Loss

Below are the surrogate loss for implicitly augmenting the feature diversity.

Denote the distribution of class c to be a multivariate Gaussian distribution, where $f(\boldsymbol{x})$ of class c obeys $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$. Consider a feature $f(\boldsymbol{x}_i)$ sampled along a direction from $\mathcal{N}(\boldsymbol{\mu}_c, \lambda\boldsymbol{\Sigma}_c)$, where λ indicates the augmentation intensity. Then the upper bound of classification loss can be viewed as below,

$$\mathbb{E}_{f(\boldsymbol{x}_i)}\Big[-\log \frac{\exp(w_{y_i}^\intercal f(\boldsymbol{x}_i) + b_{y_i})}{\sum_{c=1}^{C} \exp(w_c^\intercal f(\boldsymbol{x}_i) + b_c)}\Big] \tag{10}$$

$$= \mathbb{E}_{f(\boldsymbol{x}_i)}\Big[\log \sum_{c=1}^{C} \exp((w_c^\intercal - w_{y_i}^\intercal) f(\boldsymbol{x}_i) + (b_c - b_{y_i}))\Big] \tag{11}$$

$$\leqslant \log\Big[\sum_{c=1}^{C} \mathbb{E}_{f(\boldsymbol{x}_i)} \exp((w_c^\intercal - w_{y_i}^\intercal) f(\boldsymbol{x}_i) + (b_c - b_{y_i})\Big] \tag{12}$$

$$= \log\Big[\sum_{c=1}^{C} \exp((w_c^\intercal - w_{y_i}^\intercal)\boldsymbol{\mu}_{y_i} + (b_c - b_{y_i}) + \frac{\lambda}{2}(w_c^\intercal - w_{y_i}^\intercal)\boldsymbol{\Sigma}_{y_i}(w_c - w_{y_i}))\Big], \tag{13}$$

where Equation (12) is derived based on the convex property of log, whilst Equation (13) is based on the property that $\mathbb{E}[\exp(tX)] = \exp(t\mu + \frac{1}{2}\sigma^2 t^2), X \sim \mathcal{N}(\mu, \sigma^2)$. It should be noted that the bias $[b_1, b_2, ..., b_C]^T$ was not considered in the main text for simplicity, yet was taken into account during our practical implementations.

Online Estimation of Visual Feature Prototype and Feature Covariances. The online estimation of visual feature prototype is formulated in Equation (14). For each batch from $\{\boldsymbol{x}_i, y_i\}_{i=1}^{B}$ from domain n, if there are samples from class c, then its $\mathbf{v}_c^n$ is updated as below,

$$\mathbf{v}_c^n|_{new} = 0.5 \times \frac{1}{|\Lambda_c|} \sum_{y_i=c} f(\boldsymbol{x}_i) + 0.5 \times \mathbf{v}_c^n|_{old}, \tag{14}$$

where $|\Lambda_c|$ denotes the sample number of class c from current batch.

The covariance $\mathbf{\Sigma}$ is online estimated in the same manner as [6].

3 Supplementary Results and Discussions

3.1 Further Discussions on Ablation Studies

Here we added more discussions in terms of our ablation studies. The three core modules in our meta-learning framework were derived from Equation 1 in the main paper. We perform ablation studies to show the effectiveness of each module design, as well as their complementary benefits to each other. Below we give discussions based on the results in Tables 2,4 of the main paper.

(1) We proposed distribution-calibrated loss $\mathcal{L}_{dc}$ align the classification loss to a canonical balanced distribution, aiming to handle P(Y) shifts across domains. It outperforms BSCE, as shown in Table 2-BSCE.

(2) In addition, $\mathcal{L}_{dc}$ can unify both losses on $\mathcal{D}_{mtr}$ and $\mathcal{D}_{mte}$ to the same balanced distribution, showing better results when applying cross entropy instead in the meta learning setting (Table 4-d vs Table 4-c).

(3) Our Visual-Semantic mapping aims to learn domain-aligned unbiased representation by bidirectional Visual-Semantic mapping (Table 4-e,f,g) and cross prototype alignment. To validate the effectiveness of performing cross prototype alignment, in Table 4-k, we performed ablation study with a uni-domain prototype for alignment, and performance decrease can be noted. Since P(Y) of each domain is different, it is more flexible to build domain-specific prototype for cross-domain alignment, and it simultaneously mitigates the memory bottleneck issue, compared to directly sampling intra-class inter-domain samples for alignment. Moreover, our meta-learning setting can align the feature from $\mathcal{D}_{mte}$ to the prototype of $\mathcal{D}_{mtr}$, which further helps feature alignment across domains.

(4) The feature learned by Visual-Semantic mapping is also important for the augmentation module, otherwise domain shifts may dominate intra-class variances, as demonstrated in Table 4-h.

(5) In the augmentation module, we adopted the weighted term $\boldsymbol{n_k}$ in Equation 7. This leads to the situation where "header" classes would contribute more to updating the covariance matrix of tail classes, whereas "tailer" less to head classes. We performed the additional experiments by removing $\boldsymbol{n_k}$, with results presented in Table 4-l.

(6) The proposed meta-learning framework integrating the three modules is effective for **LT-DS** (Table 4-i vs Table 4-j).

3.2 Selection of Different Embeddings

We utilized typical embeddings as they are available along with previous open-source works on these datasets. Here, we added additional experiments comparing BERT, CLIP, and GloVe on **AWA2-LTS**. Results in Table 2 show that the embedding types do not affect the performance much, and our method consistently outperforms Agg w. Embs .

Table 2. Results of different embedding types based on Agg and Ours.

Methods	Acc-U	Acc	H
Agg	26.6	31.8	37.8
Agg w. BERT	28.2	33.1	39.1
Agg w. CLIP	27.6	32.8	39.2
Agg w. GloVe	27.0	32.4	38.2
Ours w. BERT	35.7	**42.0**	**44.6**
Ours w. CLIP	**35.8**	41.4	43.6
Ours w. GloVe	35.4	41.3	44.5

Table 3. Performance changes with different imbalance ratios.

Ratio	**Methods**	Acc-U	Acc	H
78	Agg	26.6	31.8	37.8
	Ours	**35.7**	**42.0**	**44.6**
50	Agg	21.7	27.6	32.8
	Ours	**32.0**	**37.6**	**41.0**
10	Agg	13.9	18.2	20.7
	Ours	**24.0**	**23.1**	**32.1**

3.3 Sample Complexity

Here, we added experiments with **AWA2-LTS** by changing the imbalance ratio of the training set (change head sample number), while kept using the same test set. Results in Table 3 show that ours outperforms Agg counterparts by a large margin in all settings.

3.4 PACS-ODG

Our targeted problem **LT-DS** has a similar setting to open domain generalization (**ODG**) proposed in [5]. We also evaluated our proposed framework on the open domain generalization task on the **PACS-ODG** dataset introduced in [5]. We followed the settings of [5]. The original class number and domain number of **PACS** [1] is 7 and 4, respectively. Each domain has its predefined training/validation/testing splits [1]. In the settings of open domain generalization, only part of the label set was selected in each individual training domain, and the trained model is tested on the held-out testing domain consisting of all classes. Following [5], three domains are used for training and validation, and the held-out domain is for testing. In line with [5], we reported the metrics **Acc-U** and **H-U** on the held-out domain. In addition, during testing, we also validated on the testing data of all the domains, where the domain-average accuracy of all non-open classes **Acc** were reported. The detailed split settings are listed in Table 4, and the domain order of each leave-one-domain-out loop is CPS-A, PAC-S, ACS-P, SPA-C.

Under the same experimental settings, we compared our results of ***Acc-U*** and ***H-U*** with the results of Agg, Epi-FCR [2], CuMix [4], DAML [4] reported in [5]. We leveraged the source-code of DAML [4] and reported its ***Acc*** result[1]. We also applied state-of-the-art domain generalization algorithm Mixstyle [8] for comparison. We used the ResNet-18 pretrained from ImageNet as f following previous works [4,5]. We did not apply Semantic-Similarity Augmentation module for **PACS-ODG** due to the limited class number and the non-existing long-tailed issue of this dataset.

[1]similar ***Acc-U*** and ***H-U*** results can be achieved as [5]

Table 4. Settings of Open Domain Generalization of **PACS-ODG**.

Domain	Training	Testing
Domain 1	0,1,3	0,1,2,3,4,5
Domain 2	0,2,4	0,1,2,3,4,5
Domain 3	1,2,5	0,1,2,3,4,5
Domain 4	-	0,1,2,3,4,5,6

Table 5. Results on **PACS-ODG** dataset.

	Art			Sketch			Photo			Cartoon			Avg		
Method	Acc-U	Acc	H-U	Acc-U	Acc	H-U	Acc-U	Acc	H-U	Acc-U	Acc	H-U	Acc-U	Acc	H-U
Agg	51.4	-	38.8	49.8	-	47.1	53.2	-	44.2	66.4	-	49.0	55.2	-	44.8
Epi-FCR[2]	54.2	-	41.2	46.4	-	46.1	70.0	-	48.4	72.0	-	58.2	60.6	-	48.5
CuMix[4]	53.9	-	38.7	37.7	-	28.7	65.7	-	49.3	74.2	-	47.5	57.9	-	41.1
DAML[5]	54.1	60.6	43.0	58.5	75.5	**56.7**	75.7	68.3	53.2	73.7	76.9	54.5	65.5	70.3	51.9
MixStyle[8]	56.0	57.8	47.0	51.7	75.0	44.9	58.7	57.0	33.0	**75.8**	75.5	**63.6**	60.5	66.3	47.1
Ours	**58.4**	**66.3**	**47.8**	**60.4**	**80.1**	49.1	**78.4**	**77.2**	**71.2**	71.3	**77.8**	59.6	**67.1**	**75.3**	**56.9**

As shown in Table 5, overall our method show favorable performance compared to other methods. Actually, although the authors of [5] did not consider the imbalance issue in their work, this indeed exists, since the label set is only partial in each set, leading to the "infinite" imbalance ratio. On the other hand, those classes common in all domains would contribute to more samples, thus leading to an overall imbalanced distribution. This imbalance problem, inherent in **ODG**, was however overlooked in existing studies.

3.5 Additional Qualitative Results

In this section, more qualitative results on **AWA2-LTS** are shown.

Inter-Domain Discrepancy: We present the inter-domain discrepancies on the testing subset (all five domains) during the training procedure. The distance is calculated by Fréchet distance. It can be observed in Fig. 2 that the inter-domain distances were largely decreased and maintained in a small scale by our proposed method, whilst the discrepancies under the Agg baseline are becoming much larger after training. This emphasizes the overfitting on seen domains by the conventional Agg method.

Covariance Similarity Before and After Semantic-Similarity Guided Update: Fig. 3 shows the inter-class similarity of the semantic embeddings, as well as of the original and updated covariance matrix. For visualization purposes, Fig. 3(a) shows the cosine similarity of inter-class semantic embeddings, whereas Figs. 3(b) and 3(c) calculates the pairwise distances between class i and j by $d(i,j) = \exp(-\left\|\mathbf{\Sigma}_i - \mathbf{\Sigma}_j\right\|_2)$. Guided by the similarity derived from semantic

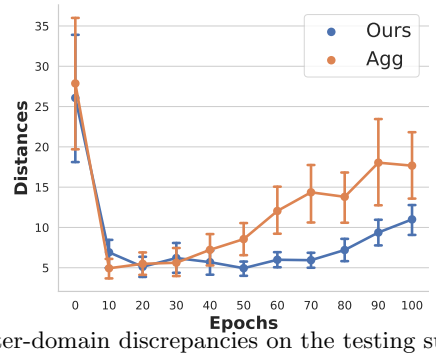
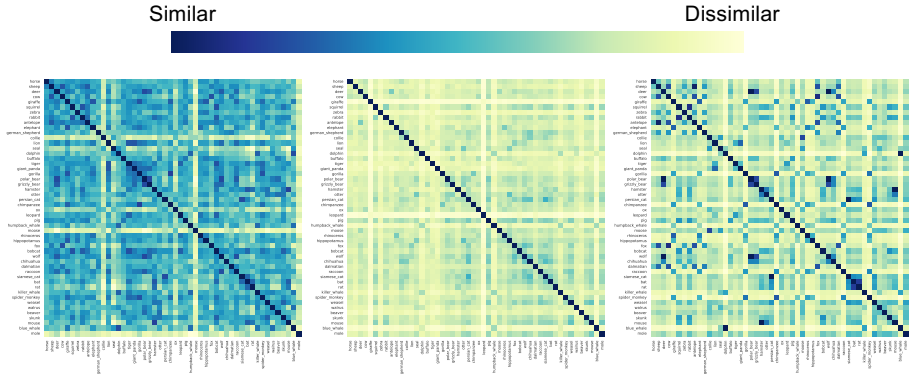


Fig. 2. Changes of inter-domain discrepancies on the testing subset (all five domains) during training. The inter-domain discrepancies of Agg start to increase significantly at a very early stage, whereas the discrepancies of our method remain more stable.

embeddings (as in Fig. 3(a), the original covariance matrix was updated by the top k most similar classes weighted by their class numbers. With the weighted update strategy, the covariance matrices of head classes are not affected too much, whereas those tail covariance matrices can be much influenced by those similar head classes for better modelling.



(a) Semantic Similarity (b) Pairwise Distance of Original Covariances (c) Pairwise Distance of Updated Covariances

Fig. 3. Pairwise Similarity of Semantic Embeddings, Original and Updated Covariance Matrices. Based on the inter-class semantic similarity guided from (a), the covariance matrices, especially of those tail classes, are updated based on the statistics from top k most similar classes.

Visualizations of Top-5 Retrieval In Fig. 4, we presented a few examples of top 5 most similar samples in the semantic embedding space when holding Original domain out. We selected one instance from one head, middle, and tail

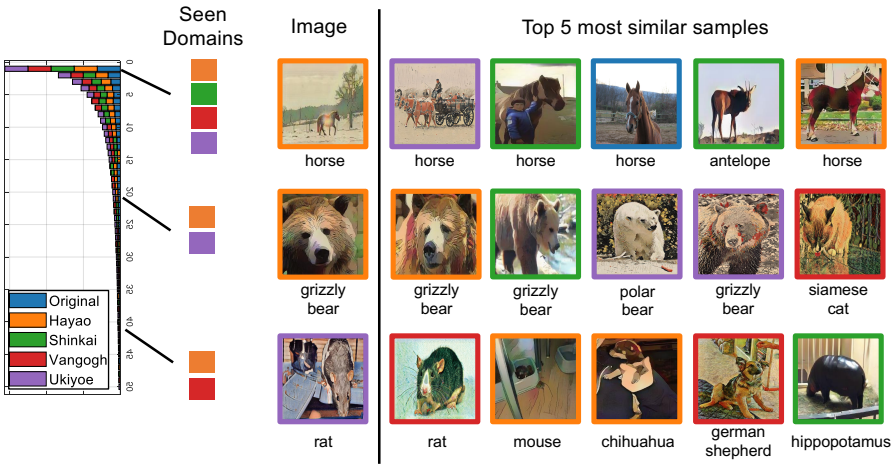


Fig. 4. Representative examples of top 5 most similar samples in the semantic embedding space. "Seen domains" mean the domains in the training subset of which the current category is available.

class, separately. The seen domains of each class during training are visualized on the left side. It can be observed from Fig. 4 that the similarity is mostly based on its semantic meaning rather than the domain styles. For example, in the Row 1 of Fig. 4, the most similar three samples of the Hayao *horse* are from other different domains. Similarly, in the Row 3, even *rat* of Ukiyoe is not seen during training, the most similar counterpart is from the same class yet a different domain. We also noticed some fine-grained variances between some similar classes based on this visualization, i.e., *grizzly bear* vs *polar bear*, *rat* vs *mouse*. The incapability of distinguishing them may be due to the limited effectiveness of the backbone. One possible direction of future work is to explore more effective and deeper backbones to enable better recognition performance to distinguish such small differences.

References

1. 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. pp. 5542–5550 (2017)
2. Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.Z., Hospedales, T.M.: Episodic training for domain generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1446–1455 (2019)
3. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2537–2546 (2019)
4. Mancini, M., Akata, Z., Ricci, E., Caputo, B.: Towards recognizing unseen categories in unseen domains. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16. pp. 466–483. Springer (2020)
5. Shu, Y., Cao, Z., Wang, C., Wang, J., Long, M.: Open domain generalization with domain-augmented meta-learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9624–9633 (2021)
6. Wang, Y., Pan, X., Song, S., Zhang, H., Huang, G., Wu, C.: Implicit semantic data augmentation for deep networks. Advances in Neural Information Processing Systems **32**, 12635–12644 (2019)
7. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. IEEE transactions on pattern analysis and machine intelligence **41**(9), 2251–2265 (2018)
8. Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Mixstyle neural networks for domain generalization and adaptation. arXiv preprint arXiv:2107.02053 (2021)