

Style Interleaved Learning for Generalizable Person Re-identification

Wentao Tan^{1*} Pengfei Wang^{1*} Changxing Ding^{1†} Mingming Gong² Kui Jia¹

¹ South China University of Technology, China

² University of Melbourne, Australia

{eewentaotan, eepengfei.wang}@mail.scut.edu.cn, chxding@scut.edu.cn

mingming.gong@unimelb.edu.au, kuijia@scut.edu.cn

Abstract

Domain generalization (DG) for person re-identification (ReID) is a challenging problem, as there is no access to target domain data permitted during the training process. Most existing DG ReID methods employ the same features for the updating of the feature extractor and classifier parameters. This common practice causes the model to overfit to existing feature styles in the source domain, resulting in sub-optimal generalization ability on target domains even if meta-learning is used. To solve this problem, we propose a novel **style interleaved learning framework**. Unlike conventional learning strategies, interleaved learning incorporates two forward propagations and one backward propagation for each iteration. We employ the features of interleaved styles to update the feature extractor and classifiers using different forward propagations, which helps the model avoid overfitting to certain domain styles. In order to fully explore the advantages of style interleaved learning, we further propose a novel **feature stylization** approach to diversify feature styles. This approach not only mixes the feature styles of multiple training samples, but also samples new and meaningful feature styles from batch-level style distribution. Extensive experimental results show that our model consistently outperforms state-of-the-art methods on large-scale benchmarks for DG ReID, yielding clear advantages in computational efficiency. Code is available at <https://github.com/WentaoTan/Interleaved-Learning>.

1. Introduction

The goal of person re-identification (ReID) is to identify images of the same person across multiple cameras. Due to its wide range of applications, such as seeking persons of interest (e.g. lost children), ReID research has experienced explosive growth in recent years [58, 24, 42, 52, 30, 51,

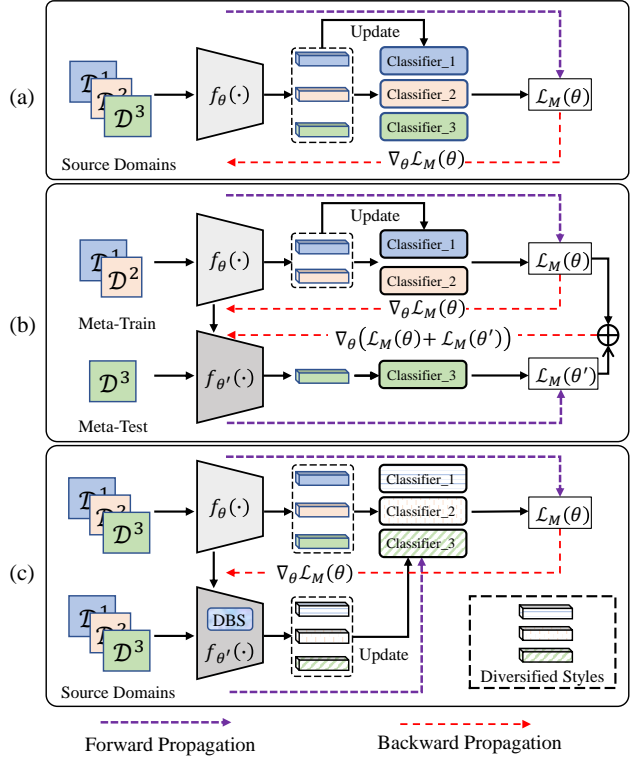


Figure 1. Differences between interleaved learning and existing learning schemes. We take the memory bank-based classifier as an example for illustration. (a) Conventional methods use the same features for the update of the feature extractor and multiple classifiers. (b) Meta-learning approaches divide multiple source domains into meta-train and meta-test domains, but still utilize the same features for the updating of the feature extractor and classifiers. (c) Interleaved learning uses features of different styles to update the feature extractor and classifiers. DBS is a powerful feature stylization approach, the details of which will be introduced in Section 3.2. Best viewed in color.

6, 54, 15, 47, 9, 45, 59, 57, 48]. Most existing approaches achieve remarkable performance when the training and testing data are drawn from the same domain. However, when

*Equal Contribution

†Corresponding Author

applying these ReID models to other domains (such as person images captured by a new camera system), ReID models generally exhibit clear performance drops due to domain gaps.

To alleviate these problems, domain generalization (DG) for person ReID has recently emerged as an important research topic [41, 14, 3, 4, 60]. DG ReID methods utilize labeled data from source domains to learn a generalizable model for unseen target domains. Compared with unsupervised domain adaptation (UDA) [37, 7, 55, 49], the DG task is more challenging, as it is unable to access any images in the target domain for model training. Moreover, unlike the traditional DG setting [35, 53], which assumes that both domains share the same classes, DG ReID is a more challenging open-set problem, in that there is no identity overlap between any two domains.

Most DG ReID methods [41, 14, 16] adopt one shared feature extractor and assign a separate classifier to each source domain. As shown in Fig. 1(a), the features of each domain extracted by the feature extractor are also used to update the parameters of the corresponding classifier. We contend that this common practice leads to sub-optimal generalization ability on unseen domains, since both the feature extractor (“player”) and classifiers (“referee”) are biased towards the same styles. Some methods adopt meta-learning [19], which divides multiple source domains into meta-train and meta-test domains to simulate real train-test domain shifts, but the above issue remains, as illustrated in Fig. 1(b). During the training process of meta-learning, the classifier for each domain is still updated according to the same features as those for loss computation.

To overcome the above limitations, we introduce a novel **style interleaved learning framework** for domain generalization. As shown in Fig. 1(c), this framework adopts features of interleaved styles to update parameters of the feature extractor and classifiers. Specifically, there are two forward propagations and one backward propagation for each iteration. In these two forward propagations, we use features of synthesized styles to update memory-based classifiers, and adopt features of the original styles for loss computation, which artificially causes domain shift between the feature extractor and classifiers. This results in style-robust gradients in the backward propagation, thereby promoting the generalization ability of the feature extractor. It is worth noting that the second forward propagation is very efficient, introducing negligible computational cost.

In our framework, it is necessary to synthesize new feature styles. Recent studies [12, 8] suggest that the ~~statistics of the channel-wise mean and standard deviation in the feature maps of one bottom CNN layer reflect style information.~~ Motivated by this observation, MixStyle [66] changes the feature styles by mixing the styles of two samples in a linear manner. However, the diversity of the styles pro-

duced may be insufficient, which constrains the power of interleaved learning. To address this problem, we propose a novel **feature stylization approach** that comprises two components. First, **Dir-MixStyle** mixes styles of more samples in a mini-batch. Second, **Batch-Style** samples new and meaningful styles from the batch-level style distributions; this non-linear operation introduces more diverse styles. By incorporating the new feature stylization approach into the interleaved learning framework, the representations produced by the feature extractor are more robust against domain shift.

In summary, the main contributions of this paper are three-fold:

- We propose a novel interleaved learning framework for domain generalization. Compared with methods that adopt non-standard model architectures or meta-learning schemes, our method is both more effective and more efficient.
- We propose a novel feature stylization approach, which produces more diverse and meaningful styles, rather than being limited to those produced by linear interpolations.
- We perform extensive experiments on multiple DG ReID benchmarks, our approach consistently outperforms state-of-the-art methods by significant margins.

2. Related Work

2.1. Domain Generalization

Domain generalization methods aim to learn a model from one or several related source domains in a way that enables the model to generalize well to *unseen* target domains. Existing DG methods handle domain shift from various perspectives, including domain alignment [33, 32, 21, 20, 26], meta learning scheme [19, 1, 27], data augmentation [44, 39, 66, 31], and disentangled representation learning [18, 17, 46].

In the field of ReID, existing works improve DG performance from three perspectives: network architecture, training strategy, and data augmentation. For the first category of methods, Dai *et al.* [4] designed a voting net, which adaptively integrates the output features from multiple expert networks based on the similarity between the target domain and each source domain, resulting in more generalizable features. Choi *et al.* [3] designed a batch-instance normalization (BIN) module that combines batch normalization (BN) and instance normalization (IN). With the help of learnable balancing parameters, BIN can both reduce the style variations between domains and alleviate the loss of discriminative information. With regard to training strategies, some works adopt meta-learning [19]. These works

divide the multiple source domains into multiple meta-train data sets and one meta-test data set, which mimics the domain gap encountered during testing. Eliminating this domain gap during training can improve generalization ability. For example, Zhao *et al.* [60] improved traditional meta learning by means of a meta batch normalization layer, which diversifies data distributions in the meta-test stage. Finally, another popular method is data augmentation based on style transfer. The purpose of style transfer is to change the style of an image while ensuring that its semantic content remains unchanged. Data augmentation diversifies the styles of the source domains and thus improves the generalizability of the trained model. For example, Zhou *et al.* [66] proposed MixStyle, which combines the styles of two samples in a linear manner. The newly synthesized styles play the role of new domains and thereby improve model generalization power.

2.2. Interleaved Learning

Interleaved learning was first introduced in the field of cognitive science and educational psychology [10, 36, 2]. In conventional learning, students are asked to do exercises to master a certain type of knowledge in a particular assignment, *e.g.*, a dozen problems that are all solved by using the Pythagorean theorem. This approach is referred to as “blocked learning” and means that students are aware of what kind of knowledge is required to solve each problem before reading the question. However, students that learn in this way may not perform well on a more comprehensive exam, in which different types of problems are mixed together. In other words, the students “overfit” to the same problem type. In interleaved learning, each assignment includes different types of problems that are arranged in an interleaved order. Interleaved practice requires students to choose a strategy based on the problem itself rather than relying on a fixed strategy. Studies in cognitive science [10, 36, 2] conclude that interleaving can effectively promote inductive learning.

Conventional DG ReID pipelines may result in overfitting to existing domain styles, similar to the example of overfitting to the same problem-solving strategy described above. To address this problem, we propose a novel interleaved learning framework for DG ReID. In our framework, we adopt features of different styles for classifier updating and loss computation, which enables the feature extractor to avoid overfitting to specific feature styles. Interleaved learning efficiently improves the model’s generalization ability on unseen domains, just as it can aid students to perform well when faced with various types of questions.

To the best of our knowledge, this is the first time that interleaved learning has been introduced to the field of ReID. Experimental results show that our framework significantly improves the DG ReID performance.

3. Methodology

An overview of our style interleaved learning framework is presented in Fig. 2. For DG ReID, we are provided with S source domains $\mathcal{D}_S = \{\mathcal{D}^s\}_{s=1}^S$, where $\mathcal{D}^s = (\mathbf{x}_k^s, y_k^s)_{k=1}^{N^s}$. N^s is the number of samples and S denotes the number of source domains in the training stage. The label spaces of the source domains are disjoint. The goal is to train a generalizable model using the source data. In the testing stage, the model is evaluated directly on the unseen target domain \mathcal{D}_T .

3.1. Style Interleaved Learning Framework

Our style interleaved learning framework (Fig. 2) includes a **CNN-based feature extractor** $f_\theta(\cdot)$ and maintains an individual **memory-based classifier** for each source domain. Unlike conventional learning, interleaved learning utilizes two forward propagations and one backward propagation for each iteration.

In the first forward propagation, we do not artificially change the feature styles. Feature vectors produced by $f_\theta(\cdot)$ are used for loss computation with class prototypes stored in memory banks. It is worth noting that the memory banks remain unchanged in this step. In the backward propagation, the model is optimized in the same way as conventional learning strategies. In the second forward propagation, we propose a novel **Dirichlet Batch feature Stylization (DBS)** module to generate the stylized image features that are utilized to update memory banks. For a source domain \mathcal{D}^s with K^s identities, its memory \mathcal{M}^s has K^s slots, where the i -th slot saves the prototype centroid \mathbf{c}_i^s of the i -th identity. No backward propagation is required after this forward propagation.

The First Forward Propagation. During each training iteration, for an image \mathbf{x}_i^s from \mathcal{D}^s , we forward it through the feature extractor and obtain the L2-normalized feature \mathbf{f}_i^s , *i.e.*, $\mathbf{f}_i^s = f_\theta(\mathbf{x}_i^s)$. We calculate the **memory-based identification loss** as follows:

$$\mathcal{L}_s = - \sum_{i=1}^{N^s} \log \frac{\exp(\langle \mathbf{f}_i^s, \mathbf{c}_+^s \rangle / \tau)}{\sum_{k=1}^{K^s} \exp(\langle \mathbf{f}_i^s, \mathbf{c}_k^s \rangle / \tau)}, \quad (1)$$

where \mathbf{c}_+^s stands for the positive class prototype corresponding to \mathbf{f}_i^s , τ is the temperature factor, and $\langle \cdot, \cdot \rangle$ indicates the computation of cosine similarity. The loss value is low when \mathbf{f}_i^s is similar to \mathbf{c}_+^s and dissimilar to all other class prototypes. It is worth noted that \mathbf{f}_i^s is not used to update the memory bank.

The Backward Propagation. The total loss is a combination of identification losses on all source domains:

$$\mathcal{L}_M(\theta) = \frac{1}{S} \sum_{s=1}^S \mathcal{L}_s, \quad (2)$$

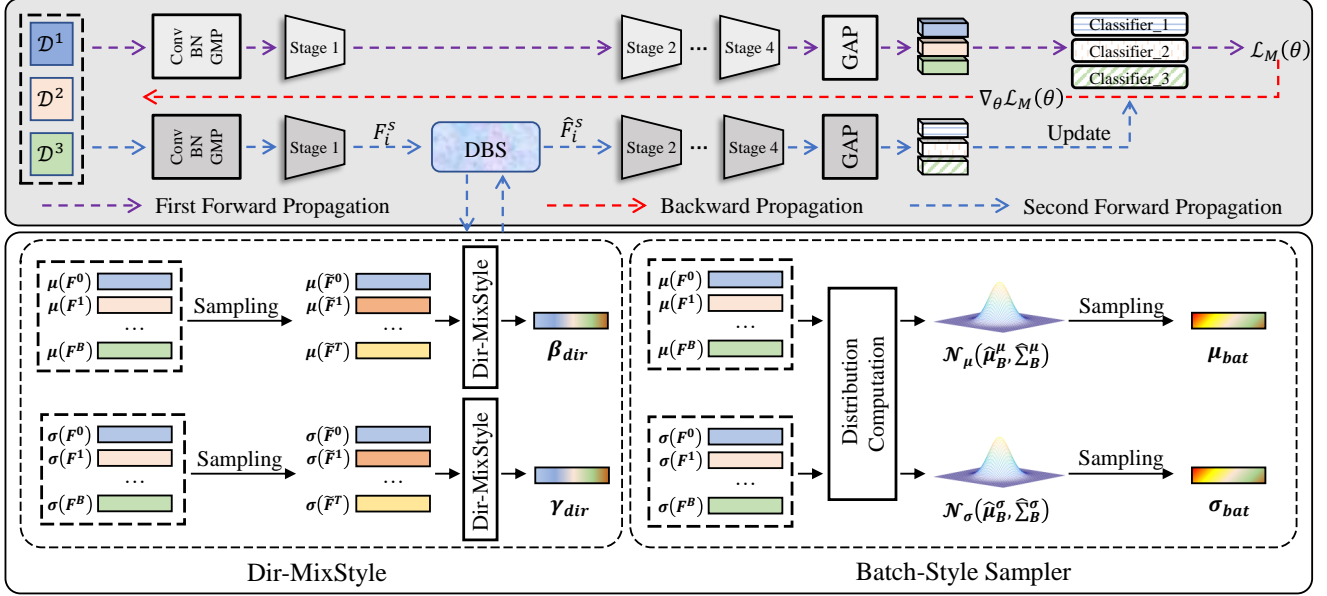


Figure 2. Illustration of the interleaved learning framework. We adopt the ResNet-50 model as backbone in this figure. Our framework incorporates two forward propagations and one backward propagation for each iteration. In the first forward propagation, we compute the loss $\mathcal{L}_M(\theta)$ according to features of original styles and the class prototypes stored in memory banks. In the backward propagation, parameters of the feature extractor are updated according to $\nabla_{\theta} \mathcal{L}_M(\theta)$. In the second forward propagation, we adopt the DBS module to generate stylized image features, which are used to update memory banks only. (Best viewed in color.)

where θ denotes the parameters of $f_{\theta}(\cdot)$ and is optimized via gradient descent:

$$\theta' \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}_M(\theta), \quad (3)$$

where α is the learning rate.

The Second Forward Propagation. The core concept of interleaved learning involves adopting features of different styles for memory updating and loss computation. The generated styles in the second forward pass should be as diverse as possible while still remaining the semantic content of the image. To achieve this goal, we propose a DBS module to transform the feature styles.

In more detail, we denote the feature maps of x_i^s output by a certain layer of $f_{\theta'}(\cdot)$ as $F_i^s \in \mathbb{R}^{C \times H \times W}$, where C , H , and W respectively denote the number of channels, height, and width, we transform the styles of F_i^s in the following way:

$$\hat{F}_i^s = \text{DBS}(F_i^s), \quad (4)$$

where $\text{DBS}(\cdot)$ is a feature stylization approach, the details of which will be introduced in Section 3.2.

We next forward \hat{F}_i^s through the remaining layers of $f_{\theta'}(\cdot)$ and obtain the L2-normalized feature vector \hat{f}_i^s . In each iteration, we adopt \hat{f}_i^s to update the corresponding class prototype c_+^s in the memory banks:

$$c_+^s \leftarrow \eta c_+^s + (1 - \eta) \hat{f}_i^s, \quad \hat{f}_i^s \in \mathcal{I}_+, \quad (5)$$

where $\eta \in [0, 1]$ is a momentum coefficient, while \mathcal{I}_+ denotes the set of samples belonging to the identity of x_i^s in

the batch. Our style interleaved learning framework repeats the above three steps until the end of training. It is worth noting that the second forward propagation is highly efficient and introduces only a small additional computational cost; more details will be provided in the section 4.4.

3.2. Feature Stylization Approach

3.2.1 Preliminaries

Recent studies on style transfer [12, 66] suggest that the style information can be revealed by the feature statistics of one CNN bottom layer for each image. It is therefore reasonable to change the feature style of an image by modifying its feature statistics.

Instance Normalization. IN has been adopted in many works on style transfer due to its ability to achieve style normalization. It is formulated as follows:

$$\text{IN}(\mathbf{F}) = \gamma \odot \frac{\mathbf{F} - \mu(\mathbf{F})}{\sigma(\mathbf{F})} + \beta, \quad (6)$$

where $\gamma, \beta \in \mathbb{R}^C$ denote channel-wise affine transformation parameters. \odot is Hadamard product, while $\mu(\mathbf{F}), \sigma(\mathbf{F}) \in \mathbb{R}^C$ store the means and standard deviations computed within each channel of \mathbf{F} . Specifically,

$$\mu_c(\mathbf{F}) = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W F_{chw}, \quad (7)$$

and

$$\sigma_c(\mathbf{F}) = \sqrt{\frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W (\mathbf{F}_{chw} - \mu_c(\mathbf{F}))^2}. \quad (8)$$

MixStyle. Inspired by AdaIN [12], which changes the style of one image by replacing γ and β with statistics of another image, MixStyle mixes the feature statistics of two images in a linear manner. Specifically,

$$\gamma_{MS} = \lambda \sigma(\mathbf{F}) + (1 - \lambda) \sigma(\mathbf{F}'), \quad (9)$$

$$\beta_{MS} = \lambda \mu(\mathbf{F}) + (1 - \lambda) \mu(\mathbf{F}'), \quad (10)$$

where λ is a weight randomly sampled from the beta distribution, while \mathbf{F} and \mathbf{F}' denote the feature maps of two instances. Finally, γ_{MS} and β_{MS} are applied to \mathbf{F} in order to change its style,

$$\text{MixStyle}(\mathbf{F}) = \gamma_{MS} \odot \frac{\mathbf{F} - \mu(\mathbf{F})}{\sigma(\mathbf{F})} + \beta_{MS}. \quad (11)$$

3.2.2 Dirichlet-Batch Feature Stylization

The approach discussed above generates new feature styles via linear interpolation between two samples, which restricts the diversity of the synthesized styles. To address this problem, we introduce a powerful stylization method named **Dirichlet-Batch Feature Stylization** (DBS). It comprises two components, which are named **Dir-MixStyle** and **Batch-Style Sampler**. The former fully exploits the potential of linear interpolation by mixing styles of multiple samples. The latter samples new styles from an estimated batch-level style distribution, providing a variety of styles not present in the source data.

Dir-MixStyle. We offer Dir-MixStyle to promote style diversity by mixing the style information of more samples. For one image with feature maps \mathbf{F} , we randomly sample T samples in the same batch, whose feature maps are denoted as $\tilde{\mathbf{F}}^t$ ($1 \leq t \leq T$). For the sake of unity, we represent \mathbf{F} as $\tilde{\mathbf{F}}^0$. Dir-MixStyle computes the mixed feature styles as follows:

$$\gamma_{dir} = \sum_{t=0}^T \lambda^t \sigma(\tilde{\mathbf{F}}^t), \quad (12)$$

$$\beta_{dir} = \sum_{t=0}^T \lambda^t \mu(\tilde{\mathbf{F}}^t), \quad (13)$$

where $\lambda = \{\lambda^t\}_{t=0}^T$ is sampled from the Dirichlet distribution [40]. Formally, $\lambda \sim \text{Dirichlet}(\varepsilon)$, where $\varepsilon \in (0, \infty)$ is a hyper-parameter.

Batch-Style Sampler. The operations in Dir-MixStyle are linear and therefore the produced styles are still limited. Batch-Style Sampler complements Dir-MixStyle by generating new styles in a nonlinear manner. More specifically,

we model the distributions of both style vectors in Eq. 7 and 8 using Gaussian distributions. For the sake of simplicity, we denote the mean and standard deviation vectors for the b -th instance in a batch as $\mu_b \in \mathbb{R}^C$ and $\sigma_b \in \mathbb{R}^C$ respectively. We calculate the distributions of both style vectors in a mini-batch as follows:

$$\hat{\mu}_B^\mu = \frac{1}{B} \sum_{b=1}^B \mu_b, \quad (14)$$

$$\hat{\Sigma}_B^\mu = \frac{1}{B} \sum_{b=1}^B (\mu_b - \hat{\mu}_B^\mu)(\mu_b - \hat{\mu}_B^\mu)^T, \quad (15)$$

$$\hat{\mu}_B^\sigma = \frac{1}{B} \sum_{b=1}^B \sigma_b, \quad (16)$$

$$\hat{\Sigma}_B^\sigma = \frac{1}{B} \sum_{b=1}^B (\sigma_b - \hat{\mu}_B^\sigma)(\sigma_b - \hat{\mu}_B^\sigma)^T, \quad (17)$$

where $\hat{\mu}_B^\sigma$ and $\hat{\Sigma}_B^\mu$ characterize the distribution of the style vector in Eq. 7, while $\hat{\mu}_B^\sigma$ and $\hat{\Sigma}_B^\sigma$ describe the distribution of the style vector in 8. B is the batchsize.

Now we obtain two multi-dimensional Gaussian distributions referred to as $\mathcal{N}_\mu(\hat{\mu}_B^\mu, \hat{\Sigma}_B^\mu)$ and $\mathcal{N}_\sigma(\hat{\mu}_B^\sigma, \hat{\Sigma}_B^\sigma)$. Next, we sample one pair of style vectors from the two distributions:

$$\mu_{\text{bat}} \sim \mathcal{N}_\mu(\hat{\mu}_B^\mu, \hat{\Sigma}_B^\mu), \quad (18)$$

$$\sigma_{\text{bat}} \sim \mathcal{N}_\sigma(\hat{\mu}_B^\sigma, \hat{\Sigma}_B^\sigma). \quad (19)$$

The fitting of the Gaussian distributions is nonlinear. In this way, we obtain new styles in a nonlinear fashion that do not exist in the original training data. We note that there is a concurrent method DSU [25] also samples new styles from Gaussian distributions. However, it adds perturbation to the original style of one sample so the generated style is still closely related to the original one. Perturbations are generated according to the ‘‘uncertainty’’ ($\hat{\Sigma}_B^\mu$ and $\hat{\Sigma}_B^\sigma$). In comparison, our Batch-Style Sampler not only calculates $\hat{\Sigma}_B^\mu$ and $\hat{\Sigma}_B^\sigma$, but also $\hat{\mu}_B^\mu$ and $\hat{\mu}_B^\sigma$. It adopts the four terms to construct two Gaussian distributions. This means the style generated by Batch-Style Sampler is more independent to the original one and therefore can be more diverse. We empirically show the superiority of our Batch-Style Sampler in the experimentation section.

Combination of Dir-MixStyle and Batch-Style Sampler. In Fig. 3, we visualize the styles produced by Dir-MixStyle and Batch-Style Sampler to analyze their respective advantages. The four images in Fig. 3 are plotted according to the same model and the same data. As shown in Fig. 3 (b), Dir-MixStyle produces new and meaningful styles that are quite close to those of the source data. In contrast, as shown in Fig. 3 (c), the Batch-Style Sampler is able to generate significantly more diverse styles than Dir-MixStyle. However, some sampled styles are far away from those of the source data, bringing in risks in authenticity.

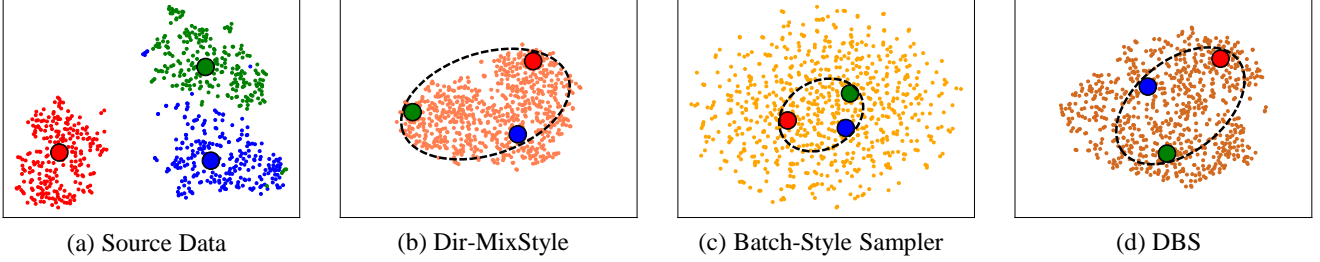


Figure 3. The 2-D t-SNE [43] visualization of feature styles. The four subfigures are plotted according to the same model and the same data. We concatenate the channel-wise mean and standard deviation of a feature map to represent its style. The red, green and blue circles represent the average style of all selected samples in one source domain, respectively. Best viewed in color.

Therefore, the two strategies have advantages in reliability and diversity, respectively. This motivates us to combine them as follows:

$$\gamma_{DBS} = z\gamma_{dir} + (1 - z)\sigma_{bat}, \quad (20)$$

$$\beta_{DBS} = z\beta_{dir} + (1 - z)\mu_{bat}, \quad (21)$$

where $z \in [0, 1]$ is a balancing parameter. Styles generated by DBS are illustrated in Fig. 3 (d). It is shown that the generated styles are more diverse than those produced by Dir-MixStyle and are more authentic than those produced by Batch-Style Sampler. Subsequent experiments in Table 4 also confirm this analysis.

Finally, we modify the style of \mathbf{F} by replacing the scaling and shifting parameters in Eq. 6 with γ_{DBS} and β_{DBS} to achieve meaningful domain style transfer, as follows:

$$DBS(\mathbf{F}) = \gamma_{DBS} \odot \frac{\mathbf{F} - \mu(\mathbf{F})}{\sigma(\mathbf{F})} + \beta_{DBS}. \quad (22)$$

We plug in the DBS module after one bottom CNN layer, *e.g.*, the first stage of the ResNet-50 model. Since DBS is parameter-free and is only used in the second forward propagation, the computational cost it introduces is very small. During inference, we remove DBS from the extractor.

4. Experiments

4.1. Datasets and Settings

Datasets. We conduct extensive experiments on public ReID datasets, namely Market1501 [61], DukeMTMC-ReID [62], CUHK03 [23] and MSMT17 [50]. For simplicity, we denote them as M, D, C3, and MS respectively. The details of the datasets are presented in Table 1. The same as [4, 3, 56], all images in each source dataset are used for training regardless of the train or test splits in its own protocol. We adopt the mean average precision (mAP) and Rank-1 accuracy as the evaluation metrics.

Settings. To facilitate comprehensive comparisons with existing works [4, 3, 16], we adopt two popular evaluation protocols.

Table 1. Dataset details. We use all images in each source dataset for training regardless of the train or test splits.

Dataset	Train		Probe		Gallery	
	IDs	Images	IDs	Images	IDs	Images
Market1501 [61]	751	12936	750	3368	751	15913
DukeMTMC [62]	702	16522	702	2228	1110	17661
CUHK03 [23]	767	7365	700	1400	700	5332
MSMT17 [50]	1041	32621	3060	11659	3060	82161

Protocol-1. This is the leave-one-out setting for M, D, C3, and MS. This setting selects one dataset from the four for testing and uses the remaining datasets for training.

Protocol-2. This protocol includes the M and D datasets. They take turns being used as the source domain and target domain, respectively.

4.2. Implementation Details

We use the ResNet-50 model [11] pretrained on ImageNet [5] as the feature extractor. Following [29, 4, 60], we set the stride of the last residual layer as 1. We sample 64 images from each source domain, including 16 identities and 4 images per identity; as a result, our batch size is $64 \times S$. For data augmentation, we perform random cropping and random flipping. For the memory, η and τ are set to 0.2 and 0.05 according to [60]. For the Dir-MixStyle module, T and ε are set to 2 and 0.1, respectively. For the Batch-Style sampler, the balancing parameter z is set to 0.9. We optimize the model using the Adam optimizer and train the model for 70 epochs. The learning rate is initialized as 3.5×10^{-4} and then divided by 10 at the 30-th and 50-th epochs. We use the warmup strategy [29] in the first 10 epochs. All experiments are conducted with PyTorch.

4.3. Comparisons with State-of-the-Art Methods

Protocol-1. To facilitate fair comparison, we adopt the same training data as [4] to train M³L [60] and obtain better results than those reported in the original paper. The comparisons in Table 2 show that our method consistently outperforms state-of-the-art methods by notable margins. In particular, our method outperforms those based on meta learning, *e.g.*, RaMoE [4], M³L [60], and MetaBIN [3]. It is worth noting that the computational cost of meta learning-based approaches is far higher than ours, as will be further

Table 2. Comparisons with state-of-the-art methods on multi-source DG ReID benchmarks under Protocol-1. † indicates evaluation results according to the code released by the authors.

Method	Backbone	D+C3+MS→M		M+C3+MS→D		M+D+C3→MS		M+D+MS→C3		Average	
		mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1
QACnv [28]	ResNet50	35.6	65.7	47.1	66.1	7.5	24.3	21.0	23.5	27.8	44.9
CBN [67]	ResNet50	47.3	74.7	50.1	70.0	15.4	37.0	25.7	25.2	34.6	51.7
SNR [16]	SNR	48.5	75.2	48.3	66.7	13.8	35.1	29.0	29.1	34.9	51.5
M ³ L [60]	ResNet50	48.1	74.5	50.5	69.4	12.9	33.0	29.9	30.7	35.4	51.9
M ³ L [60]	ResNet50-IBN	50.2	75.9	51.1	69.2	14.7	36.9	32.1	33.1	37.0	53.8
OSNet [64]	OSNet	44.2	72.5	47.0	65.2	12.6	33.2	23.3	23.9	31.8	48.7
OSNet-IBN [64]	OSNet-IBN	44.9	73.0	45.7	64.6	16.2	39.8	25.4	25.7	33.0	50.8
OSNet-AIN [64]	OSNet-AIN	45.8	73.3	47.2	65.6	16.2	40.2	27.1	27.4	34.1	51.6
MECL [56]	ResNet50	56.5	80.0	53.4	70.0	13.3	32.7	31.5	32.1	38.7	53.7
RaMoE [4]	ResNet50	56.5	82.0	56.9	73.6	13.5	34.1	35.5	36.6	40.6	56.6
M ³ L [†] [60]	ResNet50	59.6	81.5	54.5	71.8	16.0	36.9	35.2	36.4	41.3	56.7
MetaBIN [†] [3]	ResNet50-BIN	61.2	83.2	54.9	71.3	17.0	40.8	37.5	38.1	42.7	58.4
IL	ResNet50	65.1	85.4	57.2	75.8	20.5	46.0	39.0	40.3	45.5	61.9

Table 3. Comparisons with state-of-the-art methods on single-source DG ReID benchmarks under Protocol-2.

Method	Backbone	M→D		D→M	
		mAP	Rank-1	mAP	Rank-1
IBN-Net [34]	IBN-Net	24.3	43.7	23.5	50.7
OSNet [64]	OSNet	25.9	44.7	24.0	52.2
OSNet-IBN [64]	OSNet-IBN	27.6	47.9	27.4	57.8
CrossGrad [38]	ResNet50	27.1	48.5	26.3	56.7
QACnv [28]	ResNet50	28.7	48.8	27.2	58.6
L2A-OT [65]	ResNet50	29.2	50.1	30.2	63.8
OSNet-AIN [64]	OSNet-AIN	30.5	52.4	30.6	61.0
SNR [16]	SNR	33.6	55.1	33.9	66.7
MetaBIN [3]	ResNet50-BIN	33.1	55.2	35.9	69.2
Baseline	ResNet50	29.2	48.0	30.8	59.2
IL	ResNet50	34.4	55.6	40.8	70.2

explained in Section 4.4.

The interleaved and meta learning strategies solve the DG ReID problem from different perspectives. Specifically, in interleaved learning, the styles of features used for classifier updating change continuously and are different from those used for loss computation. This prevents the feature extractor from overfitting to specific feature styles, such as those contained in the source domain data. In comparison, meta learning divides the source domains into meta-train and meta-test domains to simulate the domain shift that will be encountered during the testing stage. Unfortunately, the classifier for each domain is still updated according to the same features as those for loss computation, which affects the generalization power of the ReID model.

Protocol-2. As shown in Table 3, our method still outperforms state-of-the-art methods under Protocol-2. Some recent works adopt non-standard model architectures for DG ReID. For example, both SNR [16] and MetaBIN [3] attempt to eliminate style discrepancies between instances by inserting IN layers into the backbone model. Our method does not change the model structure during inference and achieves better performance. Moreover, our style interleaved learning framework significantly improves the performance relative to the baseline. The above comparisons

justify the effectiveness of the interleaved learning strategy.

4.4. Ablation Study

We first conduct ablation study under Protocol-1 to verify the effectiveness of each component in our interleaved learning framework. Experimental results are tabulated in Table 4. In this table, we adopt two baselines that differ only in terms of their data sampling strategies (*i.e.*, balanced sampling and unbalanced sampling). For the first strategy, we randomly sample 64 images of 16 identities from each source domain. For the second strategy, we simply sample 64 images randomly from each source domain. Since the number of images for each identity is different, the second strategy is an unbalanced strategy. Recent works [4, 60] mainly adopt the first strategy. We conduct experiments on both baselines and prove that our interleaving learning framework consistently achieves superior performance.

Interleaved learning framework. It is evident that interleaved learning significantly promotes the generalization ability of both baselines. This is because providing the classifiers with richer and interleaved feature styles can help the model avoid overfitting to certain styles, such as those contained in the source domains. Specifically, interleaved feature styles introduce domain shift between the feature extractor and classifiers. Eliminating this domain shift improves the generalization ability of the feature extractor. Finally, our framework outperforms the strong baseline by 5.8% and 6.8% in terms of mAP and 4.2% and 12.6% in terms of Rank-1 in experiments of D+C3+MS→M and M+D+C3→MS, respectively.

DBS feature stylization. We draw two conclusions from Table 4: 1) Dir-MixStyle and Batch-Style Sampler improve the performance of both baselines. Their applications to the forward propagation of baselines provide features of diverse styles, which can be regarded as a data augmentation method. 2) In the interleaved learning framework, the model equipped with DBS outperforms models that adopt Dir-MixStyle or Batch-Style Sampler alone. This is because

Table 4. Ablation study on each key component. “Dir” is short for Dir-MixStyle. “Bat” stands for Batch-Style Sampler. A tick to “Dir” or “Bat” indicates that feature stylization is employed, while a tick to “IL” means that the interleaved learning scheme has been adopted.

Dir	Bat	IL	D+C3+MS→M		M+D+C3→MS	
			mAP	Rank-1	mAP	Rank-1
			59.3	81.2	13.7	33.4
✓			61.3	82.5	15.8	37.7
	✓		60.0	81.7	14.6	35.9
✓	✓		61.6	83.5	16.4	38.1
✓		✓	63.4	84.0	19.1	42.8
	✓	✓	63.1	84.2	18.9	43.1
✓	✓	✓	65.1	85.4	20.5	46.0

Dir	Bat	IL	D+C3+MS→M		M+D+C3→MS	
			mAP	Rank-1	mAP	Rank-1
			53.9	77.6	10.4	26.6
✓			55.6	78.7	12.5	31.4
	✓		54.9	78.6	11.3	28.3
✓	✓		56.2	79.9	12.8	31.9
✓		✓	59.6	82.3	15.9	38.0
	✓	✓	58.4	81.8	15.2	36.8
✓	✓	✓	61.0	83.5	17.1	39.9

DBS combines the respective strengths of Dir-MixStyle or Batch-Style Sampler, thus achieving better generalization ability.

Comparison with different stylization method.

Table 5 shows the performance of different feature-level stylization methods. All of them are tested under our interleaved learning framework. It can be seen that Dir-MixStyle consistently outperforms MixStyle. This indicates that mixing styles of more samples improves the diversity of the generated styles. Moreover, our Batch-Style Sampler approach outperforms two recently proposed stylization methods, namely, SFA-S [22] and FSDCL [13]. SFA-S [22] perturbs the original style information with random noise. In comparison, the new style information provided by Batch-Style Sampler may be more meaningful, as it is sampled from batch-level style distributions.

Comparison with GAN-based method. In this experiment, we compare the performance of DBS with one representative Generative Adversarial Network (GAN)-based image stylization method named CamStyle [63]. As CamStyle only provides synthesized images on the Market1501 and DukeMTMC datasets, we perform comparisons on D→M and M→D tasks. Specifically, we remove the DBS module and feed the style-transferred images by CamStyle to the second forward pass of IL. All the other settings remain the same. As shown in Table 6, DBS consistently outperforms the GAN-based method. This is because DBS produces more diverse and meaningful styles while the styles

Table 5. Performance comparisons between different feature stylization methods under the interleaved learning framework.

Method	D+C3+MS→M		M+D+C3→MS	
	mAP	Rank-1	mAP	Rank-1
MixStyle [66]	62.7	83.2	18.3	42.1
Dir-MixStyle ($T=2$)	63.4	84.0	19.1	42.8
Dir-MixStyle ($T=3$)	63.1	84.4	19.6	43.6
Dir-MixStyle ($T=4$)	63.4	84.3	19.6	43.7
DSU [25]	61.7	82.7	15.0	35.4
Bat	63.1	84.2	18.9	43.1
Dir+Noise [22]	63.3	84.1	19.4	43.8
Dir+FSDCL [13]	64.1	84.3	19.8	44.0
Dir+Bat (DBS)	65.1	85.4	20.5	46.0

Table 6. Performance comparisons between DBS and GAN under the interleaved learning framework.

Method	M→D		D→M	
	mAP	Rank-1	mAP	Rank-1
Baseline	29.2	48.0	30.8	59.2
GAN	33.0	52.3	32.8	62.4
DBS	34.4	55.6	40.8	70.2

of synthesized images by a single GAN model are usually limited. Moreover, compared with GAN-based methods, DBS also has clear advantages in terms of time and space complexities.

Interleaved learning VS. data augmentation. Data augmentation is a common strategy to prevent overfitting. As illustrated in Fig. 4, when DBS is employed for data augmentation, model generalization ability improves and the best activation probabilities P_{Aug} is 0.5. However, simply using DBS as a data augmentation strategy is not the best choice, as the feature extractor and classifiers are still optimized according to the same features. In comparison, embedding DBS into the IL framework presents superior performance. Moreover, as shown in Fig. 4, model generalization ability consistently improves with increasing activation probability of DBS in the second forward pass of IL. The above results justify that it is beneficial to employ features of interleaved styles for loss computation and classifier updating.

Ablation study on the balancing parameter z . In DBS, we use z to weight styles from Dir-MixStyle and Batch-Style Sampler. As illustrated in Fig. 5, the optimal value of z is 0.9. When z is smaller than 0.9, impact of Batch-Style Sampler increases, reducing reliability of the produced DBS styles. In contrast, when z is larger than 0.9, DBS styles are reliable but lack diversity, resulting in sub-optimal generalization ability.

The order of forward and backward propagations. A variant of interleaved learning involves the second forward propagation being moved to the place between the first for-

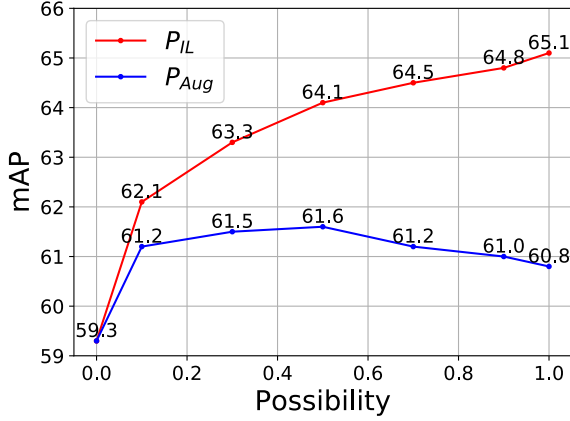


Figure 4. Performance comparison between interleaved learning and data augmentation. P_{IL} and P_{Aug} stand for the probability that DBS being activated under the IL framework and the common data augmentation setting, respectively. Experiments are conducted under the D+C3+MS→M setting.

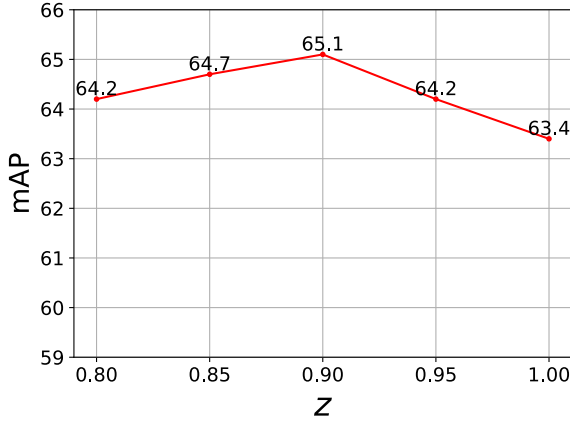


Figure 5. Ablation study on the balancing parameter z that weights styles from Dir-MixStyle and Batch-Style Sampler. The optimal value of z is 0.9. Experiments are conducted under the D+C3+MS→M setting.

ward propagation and the backward propagation. We compare the performance of these two schemes in Table 7 and find that our proposed scheme achieves better performance. This may be because after the backward propagation, the updated feature extractor $f_{\theta'}$ produces more discriminative features, promoting the quality of the prototypes stored in the memory banks.

The position to apply DBS. We place DBS to different stages of the ResNet-50 model and compare their performance in Table 8. It is shown that the best performance is achieved when DBS is placed after the first stage of ResNet-50. When it is placed after stage4, the performance de-

Table 7. Ablation study on the order of forward and backward propagations. “F” and “B” stand for the forward and backward propagations, respectively.

Variant	D+C3+MS→M		M+D+C3→MS	
	mAP	Rank-1	mAP	Rank-1
Baseline	59.3	81.2	13.7	33.4
FFB	62.5	84.7	20.1	45.5
FBF	65.1	85.4	20.5	46.0

Table 8. Ablation study on the position to apply DBS.

Stage	D+C3+MS→M		M+D+C3→MS	
	mAP	Rank-1	mAP	Rank-1
baseline	59.3	81.2	13.7	33.4
after stage1	65.1	85.4	20.5	46.0
after stage2	63.6	83.8	18.6	41.7
after stage3	63.9	84.4	18.6	41.6
after stage4	30.9	55.3	4.8	13.6

Table 9. Comparisons of model complexities. Experiments are conducted on the D+C3+MS→M setting.

Method	Train	Inference	Params
RaMoE [4]	0.989s/iter	0.94ms/img	31.5M
M ³ L [60]	1.719s/iter	0.45ms/img	23.5M
MetaBIN [3]	0.948s/iter	0.72ms/img	38.7M
IL w/o DBS	0.277s/iter	0.45ms/img	23.5M
IL	0.282s/iter	0.45ms/img	23.5M

grades dramatically; this is because features produced by stage4 contain rich semantic information. In comparison, the bottom CNN layers, *e.g.*, layers in stage 1, contain more style information, as also verified in [16, 66].

Comparisons of Model Complexity. In this experiment, we demonstrate that interleaved learning not only achieves superior performance in terms of ReID accuracy, but also is advantageous in terms of time and space complexities. To facilitate fair comparison, we adopt the same batch size and the same Titan V GPU for all methods in Table 9. It is shown that the computational cost of interleaved learning is significantly lower than that of meta learning-based methods in the training stage. Specifically, the time cost introduced by the second forward propagation step in each iteration of interleaved learning is found to be negligible. By contrast, meta learning requires two backward propagations, resulting in a high computational cost. During testing, we remove the DBS module from the feature extractor; therefore, it is used as a single standard backbone model, and the test speed is very fast.

5. Conclusions

In this paper, we propose a novel style interleaved learning framework for domain generalizable person ReID (DG ReID). This learning strategy adopts features of different

styles for classifier updating and loss computation, preventing the feature extractor from overfitting to existing feature styles contained in the source domains. We further introduce a novel feature stylization approach to produce more diverse and meaningful styles. Extensive experiments demonstrate that our approach consistently outperforms state-of-the-art methods by notable margins. Although the obtained results are promising, the generalization ability remains unsatisfying on large datasets such as MSMT17. This finding motivates us to develop more powerful DG ReID methods in the future.

References

- [1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, 2018. 2
- [2] Shana K Carpenter and Frank E Mueller. The effects of interleaving versus blocking on foreign language pronunciation learning. *Memory & cognition*, 41(5), 2013. 3
- [3] Seokeon Choi, Taekyung Kim, Minki Jeong, Hyoungeob Park, and Changick Kim. Meta batch-instance normalization for generalizable person re-identification. In *ICCV*, 2021. 2, 6, 7, 9
- [4] Yongxing Dai, Xiaotong Li, Jun Liu, Zekun Tong, and Ling-Yu Duan. Generalizable person re-identification with relevance-aware mixture of experts. In *ICCV*, 2021. 2, 6, 7, 9
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *ICCV*, 2009. 6
- [6] Changxing Ding, Kan Wang, Pengfei Wang, and Dacheng Tao. Multi-task learning with coarse priors for robust part-aware person re-identification. *IEEE TPAMI*, 2020. 1
- [7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 2
- [8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *ICCV*, 2016. 2
- [9] Xun Gong, Zu Yao, Xin Li, Yueqiao Fan, Bin Luo, Jianfeng Fan, and Boji Lao. Lag-net: Multi-granularity network for person re-identification via local attention system. *IEEE TMM*, 2021. 1
- [10] DF Halpern, A Graesser, and M Hakei. Learning principles to guide pedagogy and the design of learning environments. In *Washington, DC: Association of Psychological Science Taskforce on Lifelong Learning at Work and at Home*, 25. 3
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *ICCV*, 2016. 6
- [12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2, 4, 5
- [13] Seogkyu Jeon, Kibeom Hong, Pilhyeon Lee, Jewook Lee, and Hyeran Byun. Feature stylization and domain-aware contrastive learning for domain generalization. In *ACM MM*, 2021. 8
- [14] Jieru Jia, Qiuqi Ruan, and Timothy M Hospedales. Frustratingly easy person re-identification: Generalizing person re-id in practice. *arXiv preprint arXiv:1905.03422*, 2019. 2
- [15] Bo Jiang, Xixi Wang, Aihua Zheng, Jin Tang, and Bin Luo. Ph-gcn: Person retrieval with part-based hierarchical graph convolutional network. *IEEE TMM*, 2021. 1
- [16] Xin Jin, Cuiling Lan, Wenjun Zeng, Zhibo Chen, and Li Zhang. Style normalization and restitution for generalizable person re-identification. In *ICCV*, 2020. 2, 6, 7, 9
- [17] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012. 2
- [18] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. 2
- [19] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018. 2
- [20] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *ICCV*, 2018. 2
- [21] Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex C Kot. Domain generalization for medical imaging classification with linear-dependency regularization. *arXiv preprint arXiv:2009.12829*, 2020. 2
- [22] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *ICCV*, 2021. 8
- [23] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *ICCV*, 2014. 6
- [24] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *CVPR*, 2018. 1
- [25] Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and Ling-Yu Duan. Uncertainty modeling for out-of-distribution generalization. *arXiv preprint arXiv:2202.03958*, 2022. 5, 8
- [26] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, 2018. 2
- [27] Yiyi Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *ICML*, 2019. 2

- [28] Shengcai Liao and Ling Shao. Interpretable and generalizable person re-identification with query-adaptive convolution and temporal lifting. In *ECCV*, 2020. 7
- [29] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *CVPR*, 2019. 6
- [30] Hao Luo, Wei Jiang, Youzhi Gu, Fuxu Liu, Xingyu Liao, Shenqi Lai, and Jianyang Gu. A strong baseline and batch normalization neck for deep person re-identification. *IEEE TMM*, 2019. 1
- [31] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *ECCV*, 2020. 2
- [32] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, 2017. 2
- [33] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013. 2
- [34] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018. 7
- [35] Prashant Pandey, Mrigank Raman, Sumanth Varambally, and Prathosh AP. Generalization on unseen domains via inference-time label-preserving target projections. In *CVPR*, 2021. 2
- [36] Harold Pashler, Patrice M Bain, Brian A Bottge, Arthur Graesser, Kenneth Koedinger, Mark McDaniel, and Janet Metcalfe. Organizing instruction and study to improve student learning. ies practice guide. ncer 2007-2004. *National Center for Education Research*, 2007. 3
- [37] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 2
- [38] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018. 7
- [39] Yichun Shi, Xiang Yu, Kihyuk Sohn, Manmohan Chandraker, and Anil K Jain. Towards universal representation learning for deep face recognition. In *ICCV*, 2020. 2
- [40] Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *ICCV*, 2021. 5
- [41] Jifei Song, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Generalizable person re-identification by domain-invariant mapping network. In *ICCV*, 2019. 2
- [42] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling. In *ECCV*, 2018. 1
- [43] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 6
- [44] Riccardo Volpi and Vittorio Murino. Addressing model vulnerability to distributional shifts over image transformation sets. In *ICCV*, 2019. 2
- [45] Chaoqun Wan, Yue Wu, Xinmei Tian, Jianqiang Huang, and Xian-Sheng Hua. Concentrated local part discovery with fine-grained part representation for person re-identification. *IEEE TMM*, 2019. 1
- [46] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *ICCV*, 2020. 2
- [47] Kan Wang, Pengfei Wang, Changxing Ding, and Dacheng Tao. Batch coherence-driven network for part-aware person re-identification. *IEEE TIP*, 2021. 1
- [48] Pengfei Wang, Changxing Ding, Zhiyin Shao, Zhibin Hong, Shengli Zhang, and Dacheng Tao. Quality-aware part models for occluded person re-identification. *IEEE TMM*, 2022. 1
- [49] Pengfei Wang, Changxing Ding, Wentao Tan, Mingming Gong, Kui Jia, and Dacheng Tao. Uncertainty-aware clustering for unsupervised domain adaptive object re-identification. *IEEE TMM*, 2022. 2
- [50] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *ICCV*, 2018. 6
- [51] Longhui Wei, Shiliang Zhang, Hantao Yao, Wen Gao, and Qi Tian. Glad: Global–local-alignment descriptor for scalable person re-identification. *IEEE TMM*, 2018. 1
- [52] Jing Xu, Rui Zhao, Feng Zhu, Huaming Wang, and Wanli Ouyang. Attention-aware compositional network for person re-identification. In *CVPR*, 2018. 1
- [53] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *CVPR*, 2021. 2
- [54] Cheng Yan, Guansong Pang, Xiao Bai, Changhong Liu, Ning Xin, Lin Gu, and Jun Zhou. Beyond triplet loss: person re-identification with fine-grained difference-aware pairwise loss. *IEEE TMM*, 2021. 1

- [55] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *CVPR*, 2019. 2
- [56] Shijie Yu, Feng Zhu, Dapeng Chen, Rui Zhao, Haobin Chen, Shixiang Tang, Jinguo Zhu, and Yu Qiao. Multiple domain experts collaborative learning: Multi-source domain generalization for person re-identification. *arXiv preprint arXiv:2105.12355*, 2021. 6, 7
- [57] Peng Zhang, Jingsong Xu, Qiang Wu, Yan Huang, and Xianye Ben. Learning spatial-temporal representations over walking tracklet for long-term person re-identification in the wild. *IEEE TMM*, 2020. 1
- [58] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Densely semantically aligned person re-identification. In *CVPR*, 2019. 1
- [59] Cairong Zhao, Xinbi Lv, Zhang Zhang, Wangmeng Zuo, Jun Wu, and Duoqian Miao. Deep fusion feature representation learning with hard mining center-triplet loss for person re-identification. *IEEE TMM*, 2020. 1
- [60] Yuyang Zhao, Zhun Zhong, Fengxiang Yang, Zhiming Luo, Yaojin Lin, Shaozi Li, and Nicu Sebe. Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In *ICCV*, 2021. 2, 3, 6, 7, 9
- [61] Liang Zheng, Liye Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015. 6
- [62] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*, 2017. 6
- [63] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camera style adaptation for person re-identification. In *CVPR*, 2018. 8
- [64] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Learning generalisable omni-scale representations for person re-identification. *IEEE TPAMI*, 2021. 7
- [65] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020. 7
- [66] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021. 2, 3, 4, 8, 9
- [67] Zijie Zhuang, Longhui Wei, Lingxi Xie, Tianyu Zhang, Hengheng Zhang, Haozhe Wu, Haizhou Ai, and Qi Tian. Rethinking the distribution gap of person re-identification with camera-based batch normalization. In *ECCV*, 2020. 7