Style Interleaved Learning for Generalizable Person Re-Identification

Wentao Tan, Student Member, IEEE, Changxing Ding, Member, IEEE, Pengfei Wang, Student Member, IEEE, Mingming Gong, Member, IEEE, and Kui Jia, Member, IEEE

Abstract—Domain generalization (DG) for person re-identification (ReID) is a challenging problem, as access to target domain data is not permitted during the training process. Most existing DG ReID methods update the feature extractor and classifier parameters based on the same features. 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. To solve this problem, we propose a novel style interleaved learning (IL) framework. Unlike conventional learning strategies, IL 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 to prevent the model from overfitting to certain domain styles. To generate interleaved feature styles, we further propose a new style stylization approach. It produces a wide range of meaningful styles that are both different and independent from the original styles in the source domain, which caters to the IL methodology. Extensive experimental results show that our model not only consistently outperforms state-of-the-art methods on large-scale benchmarks for DG ReID, but also has clear advantages in computational efficiency.

Index Terms—Domain generalization, interleaved learning, person re-identification.

I. 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, which include seeking persons of interest (e.g., lost children), ReID research has undergone explosive growth in recent years [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. Most existing approaches can achieve remarkable performance when the training and testing data are drawn from the same domain. However, when these ReID models are applied to other domains (such as person images captured by a new camera system), they often suffer from clear performance drops due to the presence of domain gaps. To alleviate these problems, domain generalization (DG) for person ReID has recently emerged as an important research topic [15], [16], [17], [18], [19], [20], [21], [22]. DG ReID methods utilize labeled data from source domains to learn a generalizable model for unseen target domains. Compared with unsupervised domain adaptation (UDA) [9], [23], [24], [25], [26], [27], the DG task is more challenging, as it cannot access any images in the target domain for model training purposes. Moreover, unlike the traditional DG setting [28], [29], [30], [31], [32], [33], [34], 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 [16], [20], [21], [22], [35], [36], [37] adopt a single 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 the model displaying sub-optimal generalization ability on unseen domains, since both the feature extractor (“player”) and classifiers (“referees”) are biased towards the same styles. We further provide an example in Section II that vividly illustrates how this common practice affects the model’s generalization ability. Some DG methods adopt meta-learning [17], [19], [21], [38], which involves dividing multiple source domains into meta-train and meta-test domains to simulate real train-test domain shifts; however, the above issue persists under these circumstances, as illustrated in Fig. 1(b). During the meta-learning training process, the classifier for each domain is still updated based on the same features as those used for loss computation.

To overcome the above limitations, we introduce a novel style interleaved learning (IL) framework for DG. As shown in Fig. 1(c), this framework adopts the features of interleaved styles to update the parameters of the feature extractor and classifiers. In more detail, there are two forward propagations and one backward propagation for each iteration. In the two
the training data. It then performs uniform sampling within the estimated intervals, which ensures that each style obtained is both meaningful and independent from the original styles in the training data. In the experimentation section, we demonstrate that ISG outperforms existing works [39, 40, 41] by large margins under the IL framework.

In summary, the main contributions of this article are three-folds:

1) We propose a novel IL framework for domain generalization, which is highly computationally efficient and can be readily applied to various model architectures.
2) We propose a new feature stylization approach, the Interleaved Style Generator. Compared with existing stylization techniques, ISG produces feature styles that are not only more diverse but also compatible with our IL framework.
3) We perform extensive experiments on multiple DG ReID benchmarks, which show that our approach consistently outperforms state-of-the-art methods by significant margins.

The remainder of this article is structured as follows. In Section II, related works on DG for person ReID and IL are briefly reviewed. Section III provides a description of the proposed IL framework and ISG. Section IV presents comprehensive experiments and an analysis of the results. Section V serves as our conclusion.

II. RELATED WORKS

We divide our review of the related literature into two parts: 1) Domain Generalization (DG), and 2) Interleaved Learning (IL).

A. Domain Generalization

The goal of DG methods is 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 [42, 43, 44, 45, 46, 47, 48], training strategy [49, 50, 51, 52], data augmentation [33, 39, 53, 54, 55, 56], and the causal mechanism [32, 57].

Existing works in the field of ReID largely seek to improve DG performance from three perspectives: network architecture, training strategy, and data augmentation. For the first category of methods, Xu et al. [20] designed a new network that learns both domain-specific and domain-invariant features. During testing, this network adaptively integrates the above features based on the correlation between the statistical information of the test sample and that of the source domains to produce more generalizable features. Jiao et al. [22] introduced a new normalization module that employs dynamic convolution to remove styles while maintaining discriminative patterns during feature normalization. Choi et al. [17] proposed 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 difference between domains and alleviate the loss of discriminative information.
With regard to training strategies, some works adopt meta-
learning [38]. These works divide the source domains into mul-
tiple meta-train datasets and one meta-test dataset, which mim-
ics the domain gap encountered during testing. Eliminating this
domain gap during training can improve generalization ability.
For instance, Zhang et al. [21] apply meta-learning to train a
dynamic neural network capable of learning domain-specific
features and embedding them into a shared feature space. As a
result, the network is better able to extract generalizable
features from an unknown target domain. Moreover, Zhao et al. [19]
improved the traditional meta-learning by enriching the
data distribution in the meta-test stage. These authors pro-
posed a meta batch normalization (MetaBN) module that can
inject domain information from the meta-train datasets into the
meta-test features. This process generates a wider range of fea-
ture styles and improves the model’s ability to generalize to new
domains.

Finally, another popular method is data augmentation, which
diversifies the styles of the source data and thereby improves
the model’s generalization ability. Most methods in this cate-
gory directly change the feature statistics of one training im-
age. For example, Nuriel et al. [41] proposed pAdaIN, which
swaps feature statistics between samples in one batch. Zhou et
al. [39] introduced MixStyle, an approach that combines the
statistics of two samples in a linear manner. Li et al. [40] de-
dsigned DSU, which imposes disturbance on the original feature
statistics. There are also methods that implement style transfer
in the frequency domain. For example, Xu et al. [31] proposed
the FACT method, which fuses or exchanges the low-frequency
portions of the amplitude spectra of two images to achieve style
transfer.

Our proposed IL approach belongs to the second category
of methods in that it focuses on the training strategy. Unlike
meta-learning, which imitates the domain gap by partitioning
multiple source domains, IL introduces a domain gap between
the feature extractor and classifier by synthesizing the features
of interleaved styles. In the experimentation section, we demon-
strate that IL is both more powerful and more efficient.

B. Interleaved Learning

IL was first introduced in the fields of cognitive science and
educational psychology [58], [59], [60]. In conventional learn-
ing, students are asked to complete exercises in a particular as-
signment in order to master a certain type of knowledge (for
example, a dozen problems that can all be solved by using the
Pythagorean theorem). This approach, which is referred to as
“blocked learning”, results in students becoming aware of what
kind of knowledge is required to solve each problem before they
read the question. However, students that learn in this way may
not perform well on a more comprehensive exam in which dif-
ferent types of problems are mixed together; in other words, the
students “overfit” to the same problem type. In IL, each assign-
ment 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 [58], [59],
[60] have concluded that interleaving can effectively promote
inductive learning.

Similar to the example of overfitting to the same problem-
solving strategy described above, conventional ReID pipelines
may result in overfitting to existing domain styles. To address
this problem, we propose a novel IL framework for DG ReID.
Our framework adopts the features of interleaved styles for clas-
sifier updating and loss computation, which prevents the feature
extractor from overfitting to specific feature styles. Just as IL can
help students to perform well when faced with various types of
questions, it can also be used to efficiently improves the model’s
generalization ability on unseen domains.

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

III. METHODOLOGY

An overview of our style interleaved learning framework is
presented in Fig. 2. For DG ReID, we are provided with S source
domains \( D_{source} = \{D^s\}_{s=1}^S \); here, \( D^s = \{x^s_k, y^s_k\}_{k=1}^{N^s} \). \( N^s \) is
the number of samples, while \( S \) denotes the number of source
domains. 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 directly evaluated on the unseen
target domain \( D_{target} \).

A. Style Interleaved Learning Framework

Our style interleaved learning framework (Fig. 2) includes a
CNN-based feature extractor \( f_0(\cdot) \) and maintains an individual
memory-based classifier for each source domain. Unlike con-
ventional learning, IL utilizes two forward propagations and one
backward propagation for each iteration.

In the first forward propagation, we do not artificially change
the feature styles. Instead, feature vectors produced by \( f_0(\cdot) \)
are used to perform loss computations with class prototypes
stored in memory banks (it should be noted that the memory
banks remain unchanged in this step). In the backward propa-
gation, moreover, the model is optimized in the same way as
classical inductive learning. In the second forward propa-
gation, we introduce our proposed ISG, an effective method
of generating stylized image features that are utilized to update
the memory banks. For a source domain \( D^s \) with \( K^s \) iden-
tities, its memory \( \mathcal{M}^s \) has \( K^s \) slots, where the \( i \)-th slot saves
the prototype centroid \( c_i^s \) of the \( i \)-th identity. After this sec-
tioned method, no further backward propagation is
required.

1) The first forward propagation: During each training it-
eration, for an image \( x^s_i \) from \( D^s \), we forward it through the
feature extractor and obtain the L2-normalized feature \( f_i^s \), i.e.,
\( f_i^s = f_0(x^s_i) \). We calculate the memory-based identification loss
as follows:

\[
L_s = \frac{1}{N^s} \sum_{i=1}^{N^s} \log \frac{\exp((f_i^s, c_i^s)/\tau)}{\sum_{k=1}^{K^s} \exp((f_i^s, c_k^s)/\tau)},
\]
where \( a_x \) stands for the positive class prototype corresponding to \( f_{\gamma} \), \( \tau \) is the temperature factor, and \( \langle \cdot, \cdot \rangle \) denotes the computation of cosine similarity. The loss value is low when \( f_{\gamma} \) is similar to \( c_x \) and dissimilar to all other class prototypes. It is worth noting that \( f_{\gamma} \) is not used to update the memory bank.

2) The backward propagation: The total loss is a combination of identification losses on all source domains, which is used to optimize the model via gradient descent:

\[
\mathcal{L}_M(\theta) = \frac{1}{S} \sum_{s=1}^{S} \mathcal{L}_s, \tag{2}
\]

\[
\theta' \leftarrow \theta - \alpha \nabla \mathcal{L}_M(\theta), \tag{3}
\]

where \( \theta \) denotes the parameters of \( f(\cdot) \), while \( \alpha \) is the learning rate.

3) The second forward propagation: The core concept of IL involves adopting the features of different styles for memory updating and loss computation. The styles generated in the second forward pass should be interleaved with the sample styles from the first forward pass while still ensuring that the semantic content of the image is retained. To achieve this goal, we introduce an ISG module to transform the feature styles.

In more detail, we denote the feature maps of \( x^i_s \) output by a certain layer of \( f(\cdot) \) as \( F^i_s \in \mathbb{R}^{C \times H \times W} \), where \( C, H, \) and \( W \) denote the number of channels, the height, and the width, respectively. We transform the styles of \( F^i_s \) in the following way:

\[
\hat{F}^i_s = \text{ISG} (F^i_s), \tag{4}
\]

where ISG(\cdot) is a feature stylization approach, the details of which will be introduced in Section III-B.

We next forward \( \hat{F}^i_s \) through the remaining layers of \( f(\cdot) \) 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 \( c^i_x \) in the memory banks:

\[
c^i_x \leftarrow \eta c^i_x + (1 - \eta) \hat{f}^i_s, \quad \hat{f}^i_s \in \mathcal{I}_+, \tag{5}
\]

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 training is complete. It should be noted that the second forward propagation is highly efficient and introduces only a small additional computational cost; more details will be provided in Section IV-D.

B. Interleaved Style Generator

1) Preliminaries: Recent studies on style transfer [39], [40], [61] suggest that the style information for each image can be revealed by the feature statistics of one CNN bottom layer. Specifically, for a feature map \( F \in \mathbb{R}^{C \times H \times W} \), its style can be represented via \( \mu(F), \sigma(F) \in \mathbb{R}^C \), which respectively store the means and standard deviations computed within each channel of \( F \):

\[
\mu_x(F) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} F_{chw}, \tag{6}
\]

\[
\sigma_x(F) = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (F_{chw} - \mu_x(F))^2}. \tag{7}
\]

It is therefore reasonable to change the feature style of an image by modifying its feature statistics. The obtained new feature map can be represented as follows:

\[
\tilde{F} = \gamma \odot F - \mu_x(F) + \beta, \tag{8}
\]

where \( \gamma \) and \( \beta \) denote the channel-wise affine transformation parameters, while \( \odot \) is the Hadamard product. Various approaches have been proposed to obtain reasonable values of \( \gamma \) and \( \beta \). As illustrated in Fig. 3, pAdaIN [41] swaps the feature statistics of two images in one batch, with the newly introduced values defined as the \( \gamma \) and \( \beta \) for each image. MixStyle [39] mixes the feature statistics of two images in a linear manner to obtain \( \gamma \) and \( \beta \). DSU [40] imposes disturbances on the original feature statistics of each image to obtain the affine transformation parameters.

2) Proposed approach: The IL framework favors interleaved styles. However, the feature styles synthesized by most existing approaches [39], [40] remain closely related to those of the training images, which deviates from the spirit of IL. In the following, we propose an Interleaved Style Generator, which generates styles that are both meaningful and independent from the original styles in each batch.
Fig. 3. Comparisons between ISG and pAdaIN [41], MixStyle [39], and DSU [40]. We assume there are three source domains in this figure. (a) pAdaIN swaps feature styles between two samples. (b) MixStyle linearly mixes the styles of two samples. (c) DSU imposes disturbance on the original style of each sample. (d) ISG uniformly samples a style within the estimated style space. Best viewed in color.

Similar to [40], we model the distributions of both style vectors in (6) and (7) 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 first calculate the distributions of both style vectors in a mini-batch as follows:

\[
\hat{\mu}_\mu = \frac{1}{B} \sum_{b=1}^{B} \mu_b, \quad \hat{\mu}_\sigma = \frac{1}{B} \sum_{b=1}^{B} \sigma_b, \quad (9)
\]
\[
\hat{\sigma}_\mu^2 = \frac{1}{B} \sum_{b=1}^{B} (\mu_b - \hat{\mu}_\mu) \odot (\mu_b - \hat{\mu}_\mu), \quad (10a)
\]
\[
\hat{\sigma}_\sigma^2 = \frac{1}{B} \sum_{b=1}^{B} (\sigma_b - \hat{\mu}_\sigma) \odot (\sigma_b - \hat{\mu}_\sigma), \quad (10b)
\]

where \( \hat{\mu}_\mu \) and \( \hat{\sigma}_\mu \) characterize the distribution of the style vector in (6), while \( \hat{\mu}_\sigma \) and \( \hat{\sigma}_\sigma \) describe the distribution of the style vector in (7) and \( B \) is the batch size.

Subsequently, we represent the range of meaningful feature styles as \([\hat{\mu}_\mu - \rho \cdot \hat{\sigma}_\mu, \hat{\mu}_\mu + \rho \cdot \hat{\sigma}_\mu]\) and \([\hat{\mu}_\sigma - \rho \cdot \hat{\sigma}_\sigma, \hat{\mu}_\sigma + \rho \cdot \hat{\sigma}_\sigma]\), where \( \rho \) is a hyper-parameter. We then obtain new feature styles via uniform sampling within the above intervals, as follows:

\[
\beta_{\text{ISG}} \sim U_\mu \left( \mu_{\text{in}} - \rho \cdot \sigma_{\text{in}}, \mu_{\text{in}} + \rho \cdot \sigma_{\text{in}} \right), \quad (11)
\]
\[
\gamma_{\text{ISG}} \sim U_\sigma \left( \sigma_{\text{in}} - \rho \cdot \sigma_{\text{in}}, \sigma_{\text{in}} + \rho \cdot \sigma_{\text{in}} \right). \quad (12)
\]

Finally, we modify the style of \( F \) by replacing the \( \gamma \) and \( \beta \) in (8) with \( \gamma_{\text{ISG}} \) and \( \beta_{\text{ISG}} \) to achieve meaningful style transfer, as follows:

\[
\text{ISG}(F) = \gamma_{\text{ISG}} \odot \frac{F - \mu(F)}{\sigma(F)} + \beta_{\text{ISG}}. \quad (13)
\]

The use of uniform sampling has two key benefits. First, the obtained \( \beta_{\text{ISG}} \) and \( \gamma_{\text{ISG}} \) are independent from the original styles \( \mu(F) \) and \( \sigma(F) \). Therefore, ISG is highly suitable for use with IL, which requires features of interleaved styles for the updating of the feature extractor and classifiers. Second, as revealed in Fig. 3, the obtained styles are more diverse than those produced by existing works such as [39], [40], [41].

Algorithm 1: Pseudo-code for Interleaved Style Generator

```plaintext
if not in training mode:
    return x
if random probability > p:
    return x
B,C = x.size(0), x.size(1)
# compute instance mean & standard deviation
mu = x.mean(dim=[2, 3], keepdim=True)
var = x.var(dim=[2, 3], keepdim=True)
sig = (var + eps).sqrt()
# normalize input
x_normed = (x - mu) / sig
# compute the parameter of two intervals
mu_ns = sig.mean(0).squeeze()
mu_std = mu.std(0).squeeze()
# perform uniform sampling
Uniform_mu = Uniform(low=mu_ns - rho*mu_std, high=mu_ns + rho*mu_std, Unifrom_sig = Uniform(low=mu_std - rho*mu_std, high=mu_std + rho*mu_std)
mu_isg = Uniform(mu_sample([B, 1]).reshape([B, C, 1]))
mu_isg = Uniform_sample([B, C, 1]).reshape([B, C, 1])
# denormalize input using the sampled statistics
return x_normed + sig_isg * mu_isg
```

We insert the ISG module after one bottom CNN layer (e.g., at the first stage of the ResNet50 model). Since ISG is parameter-free and is used only in the second forward propagation, the computational cost it introduces is very small. During inference, we remove ISG from the feature extractor. A pseudo-code for ISG is presented in Algorithm 1.

IV. EXPERIMENTS

A. Datasets and Settings

Datasets: We conduct extensive experiments on several public ReID datasets, namely Market1501 [62], DukeMTMC-ReID [63], CUHK03 [64], MSMT17 [65] and CUHK-SYSU [66]. It is worth noting that DukeMTMC-reID has been
widely adopted in existing DG ReID works [17], [18], [19], [22], [35], [67], [68]. In particular, to the best of our knowledge, there is no alternative setting for the single-source DG ReID task to Protocol-3 introduced below. Therefore, this database is still adopted in this work. For simplicity, we abbreviate the names of these datasets as M, D, C3, MS and CS, respectively. Adopting the same approach as [17], [18], [35], all images in each source dataset are used for training, regardless of the train or test splits provided in the individual protocols. We adopt mean average precision (mAP) and Rank-1 accuracy as the evaluation metrics.

Settings: To facilitate comprehensive comparisons with existing works [17], [18], [35], we adopt three popular evaluation protocols.

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 is the leave-one-out setting for M, C3, and MS. This setting selects one dataset from the three for testing and uses the remaining datasets plus CS for training.

Protocol-3: This protocol includes the M and D datasets, which take turns being used as the source domain and target domain.

B. Implementation Details

We use the ResNet50 model [70] pretrained on ImageNet [71] as the feature extractor. Following [6], [18], [19], 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, $\eta$ and $\tau$ are set to 0.2 and 0.05, in line with [19]. For the Batch-Style sampler, $\rho$ is set to 3. We optimize the model using the Adam optimizer and train the model for 70 epochs. The learning rate is initialized as $3.5 \times 10^{-4}$ and then divided by 10 at the 30-th and 50-th epochs. We use the warmup strategy [6] in the first 10 epochs. Finally, we adopt the same ResNet50 model optimized according to the conventional training strategy (Fig. 1(a)) as the baseline. The augmentation setting (Aug) mentioned in this section incorporates one forward pass and one backward pass. It employs one feature stylization method in the forward pass and adopts the output feature for loss computation and classifier updating simultaneously. In comparison, our IL framework consists of two forward passes and one backward pass. In the first forward pass, we do not use any feature stylization methods and do not update the memory-based classifiers. In the second forward pass, we employ the feature stylization method and use the stylised features to update the classifiers.

C. Comparisons With State-of-the-art Methods

Protocol-1: To facilitate fair comparison, we adopt the same training data as in [18] to [17] and obtain better results than those reported in the original article. The comparisons in Table I show that our method consistently outperforms state-of-the-art methods by notable margins. In particular, our method outperforms comparison methods based on meta-learning (e.g., RaMoE [18], M3L [19], and MetaBIN [17]).

The interleaved and meta-learning strategies solve the DG ReID problem from different perspectives. Specifically, in interleaved learning, the styles of the features used for classifier updating are different from those used for loss computation; this prevents the feature extractor from overfitting to the feature styles 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. However, the classifier for each domain is still updated according to the same features as those used for loss computation, which has a risk of overfitting to source domains.
TABLE II
COMPARISONS WITH STATE-OF-THE-ART METHODS ON MULTI-SOURCE DG ReID BENCHMARKS UNDER PROTOCOL-2

| Method | Backbone | M+MS+CS→C3 mAP | Rank-1 | M+CS+3→MS mAP | Rank-1 | MS+CS+3→M mAP | Rank-1 | Average mAP | Rank-1 |
|--------|----------|----------------|--------|----------------|--------|----------------|--------|-------------|--------|
| SNR [35] | SNR50 | 17.5 | 17.1 | 7.7 | 22.0 | 52.4 | 77.8 | 25.9 | 39.0 |
| QAConv [68] | QAConv50 | 32.9 | 33.3 | 17.6 | 46.6 | 66.5 | 85.0 | 39.0 | 55.0 |
| M$^2$L [19] | ResNet50 | 32.3 | 33.8 | 16.2 | 36.9 | 61.2 | 81.2 | 36.6 | 50.6 |
| M$^2$L [19] | ResNet50-IBN | 35.7 | 36.5 | 17.4 | 38.6 | 62.4 | 82.7 | 38.5 | 52.6 |
| MetaBIN [17] | ResNet50-BIN | 43.0 | 43.1 | 18.8 | 41.2 | 67.2 | 84.5 | 43.0 | 56.3 |
| ACL [21] | ACL | 49.4 | 50.1 | 21.7 | 47.3 | 76.8 | 90.6 | 49.3 | 62.7 |
| META [20] | META | 47.1 | 46.2 | 24.4 | 52.1 | 76.5 | 90.5 | 49.3 | 62.9 |
| Baseline | ResNet50 | 35.6 | 36.1 | 17.6 | 38.0 | 66.4 | 84.6 | 39.9 | 52.9 |
| IL | ResNet50 | 41.0 | 41.8 | 23.8 | 51.2 | 72.0 | 88.5 | 45.6 | 60.5 |
| Baseline | ACL | 41.9 | 42.3 | 21.4 | 44.1 | 72.4 | 87.9 | 45.2 | 58.1 |
| IL | ACL | 47.6 | 48.3 | 26.8 | 54.8 | 78.2 | 90.7 | 50.9 | 64.6 |
| IL | META | 48.9 | 48.8 | 26.9 | 54.8 | 78.9 | 91.2 | 51.6 | 64.9 |

Protocol-2: As shown in Table II, IL achieves superior performance with the ResNet50 backbone. Moreover, IL can be readily applied to stronger backbones such as ACL [21] or META [20]. For example, we simply replace the ResNet50 with an ACL backbone [21]. It should be noted that the training strategies adopted in [21] (e.g., meta-learning and cluster loss) are not employed in our experiments. Instead, we adopt the cross-entropy loss function $L$.

TABLE III
COMPARISONS WITH STATE-OF-THE-ART METHODS ON SINGLE-SOURCE DG ReID BENCHMARKS UNDER PROTOCOL-3

| Method | Backbone | M→D mAP | Rank-1 | D→M mAP | Rank-1 |
|--------|----------|----------|--------|----------|--------|
| IBNet [72] | IBNet | 24.3 | 43.7 | 23.5 | 50.7 |
| OSNet [67] | OSNet | 25.9 | 44.7 | 24.0 | 52.2 |
| OSNet-IBN [67] | OSNet-IBN | 27.6 | 47.9 | 27.4 | 57.8 |
| CrossGrad [73] | ResNet50 | 27.1 | 48.5 | 26.3 | 56.7 |
| QACov [68] | QACov50 | 28.7 | 48.8 | 27.2 | 58.6 |
| L2A-OT [74] | ResNet50 | 29.2 | 50.1 | 30.2 | 63.8 |
| OSNet-AIDN [67] | OSNet-AIDN | 30.5 | 52.4 | 30.6 | 61.0 |
| SNR [35] | SNR50 | 33.6 | 55.1 | 33.9 | 66.7 |
| MetaBIN [17] | ResNet50-BIN | 33.1 | 55.2 | 33.9 | 69.2 |
| DTIN-Net [22] | ResNet50-DTIN | 36.1 | 57.0 | 37.4 | 69.8 |
| Baseline | ResNet50 | 31.4 | 50.1 | 31.2 | 59.6 |
| IL | ResNet50 | 33.0 | 57.4 | 39.9 | 69.8 |

D. Ablation Study
To verify the effectiveness of each component in our IL framework, we conduct an ablation study under Protocol-1.

Interleaved learning framework: As is evident from Table IV, IL achieves significantly better performance than the baseline. This is because interleaved feature styles introduce a domain shift between the feature extractor and classifiers. Eliminating this domain shift improves the generalization ability of the feature extractor. Moreover, as shown in Table IV and Fig. 4, IL outperforms the common data augmentation strategy, which involves utilizing ISG to diversify feature styles for both the feature extractor and classifiers. When ISG is employed for data augmentation, the best activation probability is 0.5. In comparison, when it is used in the second forward pass of IL, the model generalization ability consistently improves as the activation probability increases. The above experimental results demonstrate the superiority of the IL framework.

Comparison with existing stylization methods: We conduct comparisons with two representative stylization methods, named MixStyle [39] and DSU [40], and present the results in Table V. We apply each of these methods to both the data augmentation and IL strategies.

TABLE IV
ABLATION STUDY ON EACH KEY COMPONENT

| ISG | IL | D+C→MS→M mAP | Rank-1 | D+D+C→MS mAP | Rank-1 |
|-----|----|---------------|--------|---------------|--------|
| ✔ | ✔ | 59.3 | 81.2 | 14.7 | 35.2 |
| ✔ | ✔ | 62.3 | 83.3 | 15.3 | 36.9 |
| ✔ | ✔ | 65.8 | 86.2 | 20.2 | 45.7 |
A tick next to "ISG" only indicates that ISG is utilized for data augmentation in the conventional learning strategy. A tick next to "IL" means that the IL scheme is adopted.

TABLE V
ABLATION STUDY ON EACH KEY COMPONENT

| MixStyle | DSU |
|---|---|
| M+MS+CS→C3 mAP | Rank-1 | M+CS+3→MS mAP | Rank-1 |
| SNR [35] | MetaBIN [17] | 35.6 | 50.1 | 33.9 | 66.7 |
| ResNet50-BIN | 31.4 | 50.1 | 31.2 | 59.6 |
| Baseline | ResNet50 | 33.0 | 57.4 | 39.9 | 69.8 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 4. Performance comparison between IL and data augmentation. $P_{IL}$ and $P_{Aug}$ denote the probability of ISG being activated under the IL framework and the common data augmentation setting, respectively. Experiments are conducted under the D+C3+MS → M setting.

TABLE V
COMPARISONS BETWEEN ISG AND EXISTING FEATURE STYLIZATION METHODS. EXPERIMENTS ARE CONDUCTED UNDER THE D+C3+MS → M SETTING

| Method    | MixStyle | DSU | ISG       |
|-----------|----------|-----|-----------|
|           | mAP      | Rank-1 | mAP      | Rank-1 | mAP      | Rank-1 |
| Baseline  | 59.3     | 81.2  | 59.3     | 81.2   | 59.3     | 81.2   |
| Augment   | 60.9     | 82.3  | 61.3     | 82.5   | 62.3     | 83.3   |
| IL        | 62.7     | 83.2  | 62.1     | 83.1   | 65.8     | 86.2   |

From these results, we can draw two conclusions: 1) Feature stylization methods perform better when applied to the interleaved learning framework. This suggests that updating the feature extractors and classifiers according to the features of interleaved styles is indeed beneficial. 2) ISG performs significantly better than MixStyle and DSU in the IL setting. Combined with the analysis in Fig. 3, we can safely conclude that synthesizing feature styles that are independent of the original styles is an essential element of successful IL.

Ablation study on the value of $\rho$: We employ $\rho$ to control the intervals at which ISG performs sampling. When the value of $\rho$ is too small, meaningful styles may be excluded, resulting in sub-optimal generalization capability. In contrast, meaningless styles can be introduced when the value of $\rho$ is too large. As illustrated in Fig. 5, the optimal value of $\rho$ is 3.

Ablation study on the sampling strategy: We test the performance of two different sampling strategies for ISG, namely Gaussian sampling and uniform sampling. The results are presented in Table VI. As can be seen from the results, uniform sampling is superior to Gaussian sampling. This is because styles that are sampled with equal probability can be more diverse (as shown in Fig. 6) and are less likely to be correlated with the original styles in the training images. Therefore, uniform sampling is more suitable for application in an IL context.

From these results, we can draw two conclusions: 1) Feature stylization methods perform better when applied to the interleaved learning framework. This suggests that updating the feature extractors and classifiers according to the features of interleaved styles is indeed beneficial. 2) ISG performs significantly better than MixStyle and DSU in the IL setting. Combined with the analysis in Fig. 3, we can safely conclude that synthesizing feature styles that are independent of the original styles is an essential element of successful IL.

Ablation study on the value of $\rho$: We employ $\rho$ to control the intervals at which ISG performs sampling. When the value of $\rho$ is too small, meaningful styles may be excluded, resulting in sub-optimal generalization capability. In contrast, meaningless styles can be introduced when the value of $\rho$ is too large. As illustrated in Fig. 5, the optimal value of $\rho$ is 3.

Ablation study on the sampling strategy: We test the performance of two different sampling strategies for ISG, namely Gaussian sampling and uniform sampling. The results are presented in Table VI. As can be seen from the results, uniform sampling is superior to Gaussian sampling. This is because styles that are sampled with equal probability can be more diverse (as shown in Fig. 6) and are less likely to be correlated with the original styles in the training images. Therefore, uniform sampling is more suitable for application in an IL context.

The order of forward and backward propagations: A variant of IL involves the second forward propagation being moved to the position between the first forward propagation and the backward propagation. We compare the performance of these two schemes in Table VII and find that our proposed scheme achieves better performance. This may be because the updated feature extractor $f_{\theta}'$ produces more discriminative features after the backward propagation has been performed, which improves the quality of the prototypes stored in the memory banks.

From these results, we can draw two conclusions: 1) Feature stylization methods perform better when applied to the interleaved learning framework. This suggests that updating the feature extractors and classifiers according to the features of interleaved styles is indeed beneficial. 2) ISG performs significantly better than MixStyle and DSU in the IL setting. Combined with the analysis in Fig. 3, we can safely conclude that synthesizing feature styles that are independent of the original styles is an essential element of successful IL.

Ablation study on the value of $\rho$: We employ $\rho$ to control the intervals at which ISG performs sampling. When the value of $\rho$ is too small, meaningful styles may be excluded, resulting in sub-optimal generalization capability. In contrast, meaningless styles can be introduced when the value of $\rho$ is too large. As illustrated in Fig. 5, the optimal value of $\rho$ is 3.

Ablation study on the sampling strategy: We test the performance of two different sampling strategies for ISG, namely Gaussian sampling and uniform sampling. The results are presented in Table VI. As can be seen from the results, uniform sampling is superior to Gaussian sampling. This is because styles that are sampled with equal probability can be more diverse (as shown in Fig. 6) and are less likely to be correlated with the original styles in the training images. Therefore, uniform sampling is more suitable for application in an IL context.

The order of forward and backward propagations: A variant of IL involves the second forward propagation being moved to the position between the first forward propagation and the backward propagation. We compare the performance of these two schemes in Table VII and find that our proposed scheme achieves better performance. This may be because the updated feature extractor $f_{\theta}'$ produces more discriminative features after the backward propagation has been performed, which improves the quality of the prototypes stored in the memory banks.
The position in which ISG is applied: We place ISG in different stages of the ResNet50 model and compare their performance in Table VIII. As the table shows, the best performance is achieved when ISG is placed after the first stage of ResNet50. Placing ISG behind more stages does not further improve performance. This may be because, when the hyper-parameter $\rho$ is properly set, it is possible to produce sufficiently diverse styles using only one ISG module. When the ISG module is placed after stage4, the performance degrades dramatically; this is because the features produced by stage4 contain rich semantic information. In comparison, the bottom CNN layers (e.g., layers in stage1) contain more style information, as is also verified in [35, 39].

Comparisons of model complexity: In this experiment, we demonstrate that IL not only achieves superior performance in terms of ReID accuracy, but is also advantageous in terms of time and space complexities. To facilitate a fair comparison, we utilize the same batch size and the same TITAN Xp GPU for all methods in Table IX. The results show that the computational cost of interleaved learning is significantly lower than that of meta-learning-based methods in the training stage. This is because meta-learning requires two backward propagations, resulting in a high computational cost. Moreover, the time cost introduced by the second forward pass in each iteration is found to be very small. It is also worth noting that the ISG module itself introduces a near-negligible cost of 0.005 s/iter. During testing, we remove the ISG module from the feature extractor; therefore, it is used as a single standard backbone model, and the test speed is very fast.

E. Qualitative Analysis

Comparisons in training loss and inference mAP curves: This experiment compares the average training loss and inference mAP curves between the IL framework and the baseline. The comparisons are illustrated in Fig. 7. It can be observed that the training loss of the IL framework decreases slower than that of the baseline. However, the inference mAPs of the IL framework are substantially higher. This is because the feature extractor and classifiers are updated with the features of interleaved styles in the IL framework, which reduces the model’s risk of overfitting on the source domain data and promotes the model’s ability to generalize to the target domain data.

Comparisons between ISG and MixStyle: We visualize the styles produced by ISG and compare them with those generated by MixStyle [39] in Fig. 8. As the figure shows, ISG creates more diverse feature styles. More specifically, the styles generated by MixStyle are quite similar to the original styles; in comparison, the styles generated by ISG are scattered uniformly within the estimated intervals (11) and (12) and are therefore more diverse.

Comparisons between ISG and DSU: We visualize the styles generated by ISG and DSU [40] in Fig. 9. It can be observed
that the styles generated by DSU are correlated with the original style, while ISG creates styles that are more independent from the original style. The main reason is that DSU conducts Gaussian sampling centered on each original sample, while ISG conducts uniform sampling within much wider intervals centered on the mean style of all samples in the batch.

MixStyle and DSU are successful feature stylization methods for data augmentation purposes. However, their synthesized feature styles lack independence from the original styles. In comparison, our proposed ISG method exhibits good compatibility with the IL framework, where independence plays a vital role. Indeed, the experimental results in Table V clearly show that ISG outperforms these two methods by significant margins.

V. CONCLUSION

In this article, we propose a novel style interleaved learning framework for domain generalizable person ReID (DG ReID). This learning strategy adopts the features of different styles for classifier updating and loss computation, which prevents the feature extractor from overfitting to the existing feature styles contained in the source domains. We further introduce a new feature stylization approach, ISG, which can produce more diverse styles that are independent from the original styles in a batch. Extensive experiments demonstrate that our approach consistently outperforms the state-of-the-art methods by significant margins. Although IL and ISG are simple and efficient techniques, they still have some limitations: (1) The diversity of styles generated by ISG can vary depending on the source styles and the hyperparameter $\rho$. (2) Despite IL achieving state-of-the-art performance, it falls short in terms of generalization on specific datasets, such as the extensive MSMT17 dataset, when compared to supervised approaches. This observation motivates us to develop more powerful DG ReID methods in the future.

APPENDIX

This appendix provides more details about the feature stylization methods compared in this article.
This method mixes the statistics of two feature maps in a linear manner, specifically:

$$pAdaN(F) = \sigma(F') \odot \frac{F - \mu(F)}{\sigma(F)} + \mu(F'),$$

where $F$ and $F'$ are feature maps of two different samples in a batch. $pAdaN$ only employs styles present in the source data; therefore, the styles it employs lack diversity compared with those created by ISG.

MixStyle [39]: This method mixes the statistics of two feature maps in a linear manner. Specifically,

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

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

where $\lambda$ is a weight that is randomly sampled from the beta distribution. Finally, $\gamma_{MS}$ and $\beta_{MS}$ are applied to $F$ to change its style,

$$MixStyle(F) = \gamma_{MS} \odot \frac{F - \mu(F)}{\sigma(F)} + \beta_{MS}.$$
Wentao Tan (Student Member, IEEE) received the B.S. degree from the South China University of Technology, Guangzhou, China, in 2020. He is currently working toward the Ph.D. degree with the School of Future Technology. His research interests include person re-identification and domain generalization.

Changing Ding (Member, IEEE) received the Ph.D. degree from the University of Technology Sydney, Ultimo NSW, Australia, in 2016. He is currently a Professor with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China. He has authored or coauthored more than 30 articles in prominent journals and conferences, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the IEEE TRANSACTIONS ON IMAGE PROCESSING (T-IP), International Journal of Computer Vision (IJCV), CVPR, ECCV, and MICCAI. His research interests include human behavior analysis and biometrics.
Pengfei Wang (Student Member, IEEE) received the B.S. and M.S. degrees from the South China University of Technology, Guangzhou, China, in 2019 and 2022, respectively. He is currently working toward the Ph.D. degree with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. His research interests include deep learning and computer vision.

Mingming Gong (Member, IEEE) is currently a Senior Lecturer in data science with the School of Mathematics and Statistics, University of Melbourne, Parkville, VIC, Australia, and an affiliated Associate Professor of machine learning with the Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE. He has authored or coauthored more than 80 research papers in top venues such as ICML, NeurIPS, ICLR, CVPR, TPAMI, IJCV, TIP, TNNLS and was a meta-reviewer for NeurIPS, ICML, ICLR, IJCAI, AAAI, TMLR, NN. His research interests include causal learning and inference, deep learning, and computer vision. He was the recipient of the Australasian Artificial Intelligence Emerging Research Contribution Award and the Australian Research Council Early Career Researcher Award.

Kui Jia (Member, IEEE) received the Ph.D. degree in computer science from the Queen Mary University of London, London, U.K. He was with the Shenzhen Institute of Advanced Technology of the Chinese Academy of Sciences, Shenzhen, Chinese University of Hong Kong, Hong Kong, the Advanced Digital Science Center, University of Illinois Urbana-Champaign, Champaign, IL, USA, and the University of Macau, Macau, China. He is currently a Professor with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China, and is the Director of Geometric Perception and Intelligence Research Lab. His research interests include theoretical deep learning and its applications in vision and robotic problems, including deep learning of 3D data and deep transfer learning. He is an Associate Editor for TMLR and TIP.