STYLE BALANCING AND TEST-TIME STYLE SHIFTING FOR DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

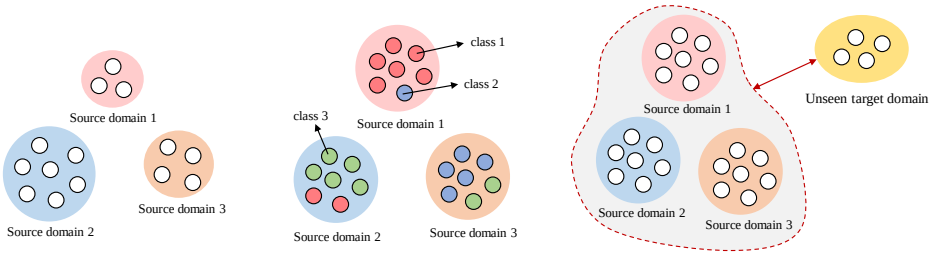
Given a training set that consists of multiple source domains, the goal of domain generalization (DG) is to train the model to have generalization capability on the unseen target domain. Although various solutions have been proposed, existing ideas suffer from severe cross-domain data/class imbalance issues that naturally arise in DG. Moreover, the performance of prior works are degraded in practice where the gap between the style statistics of source and target domains is large. In this paper, we propose a new strategy to handle these issues in DG. We first propose *style balancing*, which strategically balances the number of samples for each class across all source domains in the style-space, providing a great platform for the model to get exposed to various styles per classes during training. Based on the model trained with our style balancing, we also propose *test-time style shifting*, which shifts the style of the test sample (that has a large style gap with the source domains) to the nearest source domain that the model is already familiar with, to further improve the prediction performance. Our style balancing and test-time style shifting work in a highly complementary fashion, and can successfully work in conjunction with various other DG schemes. Experimental results on benchmark datasets show the improved performance of our scheme over existing methods.

1 INTRODUCTION

The huge success of deep convolutional neural networks (CNNs) relies on the assumption that the *domains* of the training data and the test data are the same. However, this assumption does not hold in practice. For example, in self-driving cars, although we may only have train images on sunny days and foggy days during training (source domains), we would have to make predictions for images on snowy days during testing (unseen target domain). As another example, we should sometimes make predictions on art painting images as a target domain, although we only have photograph and cartoon image datasets during training as source domains. Due to the practical significance of this problem setup, the field of domain generalization (DG) is receiving considerable attention nowadays.

Given a training set that consists of multiple (or a single) source domains, the goal of DG is to achieve generalization capability on the *unseen target domain*. Existing works tackle this problem via meta-learning (Li et al., 2019; 2018a; Zhao et al., 2021), data augmentation (Nam et al., 2021; Shankar et al., 2018; Yue et al., 2019; Zhou et al., 2020) or domain alignment (Li et al., 2018b;c;b; Erfani et al., 2016). Recently, motivated by the observations (Huang & Belongie, 2017) that the domain characteristic of data has a strong correlation with the feature statistics (or style statistics) of the early layers of CNNs, the authors of (Zhou et al., 2021; Li et al., 2022; Zhang et al., 2022) proposed to generate new style statistics during training via style augmentation. However, there are two critical issues that limit the performance of current DG approaches, as described in Fig. 1.

Issue 1: Cross-domain data/class imbalance issues. First, existing methods potentially suffer from cross-domain data/class imbalance issues that are unique to DG setups. When the number of training samples for a specific domain is limited in a *cross-domain data imbalance* setup shown in Fig. 1 (a), the domain diversity of the overall training set becomes limited, which results in reduced generalization performance. Similarly, when the train data of a specific class is concentrated in one domain in a *cross-domain class imbalance* setup of Fig. 1 (b), the domain diversity of this class becomes limited, degrading the model performance. To see how critical these issues are, we compare the average accuracy of the model on balanced PACS vs. imbalanced PACS in the following setting: given three



(a) Cross-domain data imbalance issue (b) Cross-domain class imbalance issue (c) Large style gap between source/target domains

Figure 1: DG-specific issues depicted in the style-space. Cross-domain data/class imbalance issues in Figs. 1(a) and 1(b) reduce the domain diversity; both issues arise during training and degrade the generalization capability of the model on the unseen domain. The issue in Fig. 1(c), which arises during testing, also degrades the performance since the trained model is not familiar with the unseen domain that has a large style gap with the source domains. For example, Sketch domain in Fig. 2 has a large style gap with other domains.

source domains, we removed 80% of train data of two source domains (except the largest one) to model the imbalanced dataset, and constructed a balanced dataset having the same number of train samples with the imbalanced one. By adopting the well-known style-augmentation strategy termed MixStyle (Zhou et al., 2021), in Table 1, the accuracy is degraded more than 6% due to the imbalance issue, confirming its significance. These cross-domain data/class imbalance issues have different characteristics compared to the traditional class imbalance problem in a single domain; when a specific domain lacks certain classes, it turns out in Section 5.3 that existing methods based on resampling or reweighting fail to handle these DG-specific imbalance issues, while our solution can effectively address them.

Table 1: Effect of imbalance issue in PACS. MixStyle is adopted as a baseline. The proposed style balancing (SB) reduces the gap with the balanced setting.

Methods	Accuracy
MixStyle (imbalanced dataset)	75.65
MixStyle + SB (imbalanced dataset)	78.45
MixStyle (balanced dataset)	81.88

Issue 2: Large style gap between source and target domains. Regarding the second issue, the target domain can have significantly different feature-level style statistics compared to that of source domains in practice, as shown in Fig. 1 (c). Especially in style-augmentation based DG strategies (e.g., MixStyle (Zhou et al., 2021) or EFDMix (Zhang et al., 2022)) where new style statistics are generated based on the source domains, this issue becomes critical. For example, as shown later in Section 5.1, the model performance on the Sketch domain in Fig. 2 (which has a large style gap with other source domains) is much lower compared to the performance on other domains, when style-augmentation is applied solely.

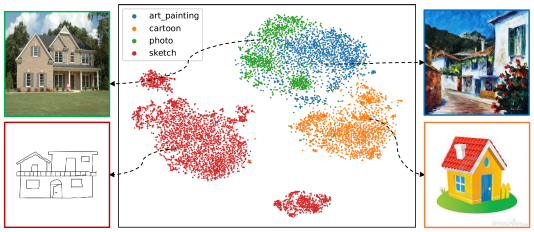


Figure 2: t-SNE of concatenated feature-level style statistics $\Phi = [\mu, \sigma]$ of each sample, obtained from the output of second residual block of ResNet-18. Samples are clustered based on domain characteristics. In this example, Sketch domain has a large style gap with other domains, resulting in low accuracy. Our test-time style shifting (TS) can boost up the model performance on the target domain that has a large style gap with the source domains.

Contributions. In this paper, we propose a new solution that tackles the above issues in DG. We first propose *style balancing*, which strategically balances the number of samples for each class across all source domains in the style-space, to handle the imbalance issues. We specifically choose the sample that has similar style statistics to other samples (and thus has a similar role compared to others) in the same domain, and convert the style of this sample to another domain. This strategy provides a great platform for the model to explore various domains per classes during training. By utilizing the trained model based on our style balancing, we also propose *test-time style shifting*, which shifts the style of the test sample (that has a large style gap with the source domains) to the nearest source domain that the model is already familiar with. This strategy enables the model to handle any target domains with arbitrary style statistics; by reducing the style gap between source and target, the performance can be significantly improved without additional model update at test-time. Experimental results on various DG benchmarks show the improved performance of our scheme over existing methods.

Our style balancing and test-time style shifting work in a highly complementary fashion; removing one of these components can significantly degrade the performance in practice having aforementioned issues at the same time. Our solution is compatible with not only the style-augmentation based DG schemes (e.g., MixStyle) that operate in the style space as ours, but also other DG ideas relying on domain alignment or meta-learning.

2 RELATED WORKS

DG with style augmentation. DG has been actively studied for the past few years using meta-learning (Li et al., 2019; Chen et al., 2022; Du et al., 2020; Li et al., 2018a; Zhao et al., 2021), data augmentation (Nam et al., 2021; Shankar et al., 2018; Yue et al., 2019; Zhou et al., 2020), domain alignment (Li et al., 2018b;c;b; Erfani et al., 2016) and so on. Motivated by the works (Huang & Belongie, 2017; Dumoulin et al., 2017) showing that style information is preserved at the early layers of CNNs, various style augmentation methods such as MixStyle (Zhou et al., 2021), DSU (Li et al., 2022), Style Neophile (Kang et al., 2022) and EFDMix (Zhang et al., 2022) have been recently proposed. As in our solution, style augmentation based DG schemes can be simply applied to any tasks/models and operate in the style space defined with style statistics. However, the performance of these methods are potentially limited in practice due to cross-domain imbalance and style gap issues. In Section 5, we show that our solution can successfully work in conjunction with recent style augmentation strategies (and also with other DG methods) to handle these issues.

Class-imbalanced learning. Targeting class-imbalanced datasets, various over/down-sampling strategies (He et al., 2008; Pouyanfar et al., 2018) and loss function modification (e.g., reweighting) methods (Huang et al., 2016; Shu et al., 2019; Cui et al., 2019) have been proposed. While these works focus on class imbalance within a single domain, in a DG setup with multiple domains, cross-domain data/class imbalance issues make the problem more challenging. Especially when each domain lacks certain classes, these missing classes cannot be compensated via over/under-sampling or loss modification strategies. A recent work (Yang et al., 2022) focused on a similar multi-domain setup with imbalanced datasets, by defining a new loss function using the distance between representations. However, the loss function proposed in (Yang et al., 2022) does not capture the classes missing in each domain. Our style balancing module handles this issue by shifting the style statistics of the sample to another domain, compensating for the missing classes in each domain.

Test-time adaptation. Several test-time adaptation methods (Wang et al., 2020; Iwasawa & Matsuo, 2021; Pandey et al., 2021; Sun et al., 2020; Xiao et al., 2022) have been proposed to adapt the model to unseen target samples at test-time, where (Pandey et al., 2021; Iwasawa & Matsuo, 2021; Xiao et al., 2022) specifically focused on DG. In (Wang et al., 2020; Iwasawa & Matsuo, 2021; Sun et al., 2020), the authors proposed schemes to update model parameters during testing. Compared to these works, our test-time style shifting does not require further model update at test-time; we simply utilize adaptive instance-normalization (AdaIN) (Huang & Belongie, 2017) to shift the style of the test sample to the familiar source domain. Recently, the authors of (Xiao et al., 2022) proposed a method that does not require fine-tuning on target samples at test-time. However, this work requires additional networks and perform Monte Carlo sampling for variational inference, which increases the training costs. Notably, (Pandey et al., 2021) proposed to construct a source manifold at the output of the feature extractor, and projects the feature of the test samples to this source manifold while preserving class information. Orthogonal to this work focusing on the output of the feature extractor where the data are clustered according to classes (regardless of the domains), we focus on shifting the style statistics at earlier layers where the data are clustered according to the domains (regardless of the classes). Moreover, our test-time style shifting does not require additional changes in the model architecture or the objective function, making our scheme to be more compatible with any task/models. To the best of our knowledge, our test-time style shifting is the first work that shifts the feature-level *style statistics* of the target sample in the style space during testing.

We stress that our style balancing and test-time style shifting are orthogonal to the aforementioned works in that we only shift the style statistics in the style-space during training/testing. Previous works on domain generalization, class-imbalanced learning and test-time adaptation can work in conjunction with our scheme to improve the prediction performance further.

3 PROBLEM SETUP

3.1 BACKGROUNDS: FEATURE/STYLE AUGMENTATION IN DOMAIN GENERALIZATION

Let $x \in \mathbb{R}^{B\times C\times H\times W}$ be a batch of features at a specific layer, where B, C, H, W are the dimensions of batch, channel, height, width, respectively. We also let $\mu(x) \in \mathbb{R}^{B\times C}$ and $\sigma(x) \in \mathbb{R}^{B\times C}$ be the channel-wise mean and standard deviation of each instance (tensor) in a mini-batch, written as

$$\mu(x) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{b,c,h,w}, \quad \sigma^2(x) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(x_{b,c,h,w} - \mu(x))^2. \tag{1}$$

The values $\mu(x)$ and $\sigma(x)$ denote **instance-level feature statistics** of x. These values also denote **style statistics** since the instance-level feature statistics carry out style information in CNNs (Huang & Belongie, 2017). Now define new style statistics $\mu(y)$ and $\sigma(y)$ computed by feature y, corresponding to another batch of images. According to adaptive instance normalization (AdaIN) (Huang & Belongie, 2017), one can generate new features having content x and style y as follows:

$$\text{AdaIN}(x, y) = \sigma(y)\frac{x - \mu(x)}{\sigma(x)} + \mu(y), \tag{2}$$

which can be viewed as a style transfer performed in the feature space.

MixStyle and DSU. MixStyle (Zhou et al., 2021) and DSU (Li et al., 2022), which are motivated by AdaIN, specifically focus on constructing new style statistics to improve generalization as

$$\text{MixStyle}(x) = \gamma_{\text{mix}}\frac{x - \mu(x)}{\sigma(x)} + \beta_{\text{mix}}, \;\; \text{DSU}(x) = \gamma_{\text{dsu}}\frac{x - \mu(x)}{\sigma(x)} + \beta_{\text{dsu}} \tag{3}$$

where β and γ are the coefficients that determine the style of the image as in (2). MixStyle specifically mixes the style statistics as $\beta_{\text{mix}} = \lambda\mu(x) + (1-\lambda)\mu(y)$, $\gamma_{\text{mix}} = \lambda\sigma(x) + (1-\lambda)\sigma(y)$, where λ is the instance-wise weight with $0 < \lambda < 1$. On the other hand, DSU generates new styles by sampling β_{mix} and γ_{mix} from Gaussian distributions.

EFDMix. The authors of (Zhang et al., 2022) propose EFDM to replace AdaIN in (2). By redefining $x \in \mathbb{R}^{HW}$ on a specific sample and a channel, the elements of vector x are reordered in an ascending order as $[x_{\tau_1}, x_{\tau_2}, \ldots, x_{\tau_{HW}}]$, where $x_{\tau_i} \leq x_{\tau_j}$ holds for $i < j$ and $\{x_{\tau_i}\}_{i=1}^{HW}$ are the elements of vector x. The elements of y are similarly reordered as $[y_{\kappa_1}, y_{\kappa_2}, \ldots, y_{\kappa_{HW}}]$. Then, arbitrary style transfer can be performed as $\text{EFDM}(x, y)_{\tau_i} = y_{\kappa_i}$ to replace (2), where $\text{EFDM}(x)_{\tau_i}$ is the τ_i-th element of the output. Based on EFDM, the authors of (Zhang et al., 2022) also propose EFDMix, which replaces the concept of AdaIN in MixStyle to EFDM, in a channel-wise manner as follows:

$$\text{EFDMix}(x)_{\tau_i} = \lambda x_{\tau_i} + (1-\lambda)y_{\kappa_i}. \tag{4}$$

We note that MixStyle, DSU and EFDMix can be viewed as feature/style-augmentation schemes for DG, as they generate new styles at feature level during training.

3.2 Problem Formulation

Let N be the number of source domains and S_n be the set of train samples in source domain n, where $S = \cup_{n=1}^{N} S_n$ is the overall train set. Let $S_{n,k}$ be the set of train samples in domain n labeled as class k satisfying $S_n = \cup_{k=1}^{K} S_{n,k}$, where K is the number of classes. Given a sample $s \in S$, let $f(s) \in \mathbb{R}^{C\times H\times W}$ be the encoded features at a specific layer. We define $\mu(f(s)) \in \mathbb{R}^C$ and $\sigma(f(s)) \in \mathbb{R}^C$ as the channel-wise mean and standard deviation of $f(s)$, similar to (1). Related to the notations in Section 3.1, we have $x = [f(s_1), f(s_2), \ldots, f(s_B)]$, $\mu(x) = [\mu(f(s_1)), \mu(f(s_2)), \ldots, \mu(f(s_B))]$, $\sigma(x) = [\sigma(f(s_1)), \sigma(f(s_2)), \ldots, \sigma(f(s_B))]$ where B is the batch size. For any set $A \subseteq S$, we also define the mean of style statistics in set A as $\mu_A = \frac{1}{|A|}\sum_{s\in A}\mu(f(s))$ and $\sigma_A = \frac{1}{|A|}\sum_{s\in A}\sigma(f(s))$. Given a set A and corresponding μ_A, σ_A, the concatenation of these two are defined as

$$\Phi_A = [\mu_A, \sigma_A]. \tag{5}$$

Similarly, we define $\Phi(f(s)) = [\mu(f(s)), \sigma(f(s))]$ for any sample s. Using these notations, we can formally state the aforementioned issues in DG as follows.

Issue 1. Cross-domain data imbalance problem denotes where $|S_n|$ are different across all source domains $n \in \{1, 2, \ldots N\}$. Cross-domain class imbalance problem denotes where $|S_{n,k}|$ are different for all domains for a specific class k. Both issues limit the performance of existing DG methods.

Issue 2. Let t be the test sample in the unseen target domain. In practice, the style gap between source and target domains could be large, i.e., $\|\Phi_{S_n} - \Phi(f(t))\|$ is large for all $n \in \{1, 2, \ldots, N\}$ (see Sketch domain in Fig. 1(c)). This issue can degrade the performance at testing since the trained model is not familiar with the new target domain that has a large gap with the source domains.

Goal. The goal of this paper is to tackle the above two issues for general DG approaches. Our specific goal is to improve the domain diversity of individual classes during training, while reducing the gap between style statistics of source and target domains at testing.

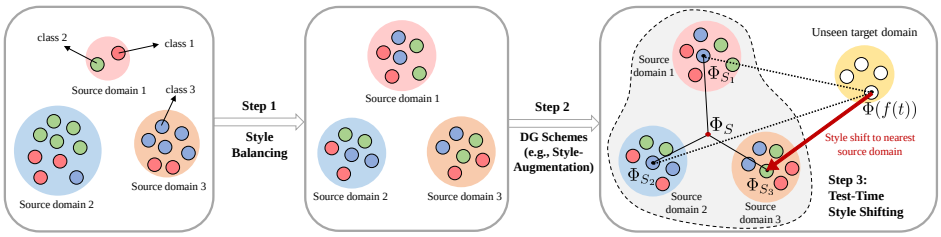


Figure 3: An overview of proposed idea. Given imbalanced feature statistics at a specific layer, style balancing is first performed to balance the style statistics across all domains. Then, a specific DG scheme (e.g., style-augmentation) can be adopted for training. At testing, the style of the test sample (far from the source domains) is shifted to the nearest source domain based on our test-time style shifting. The style balancing module and the test-time style shifting module can be flexibly applied at any layers of the backbone network.

4 Proposed Algorithm

Section 4.1 describes our *style balancing* to handle the cross-domain data/class imbalance issues. Based on the model obtained by our style balancing, in Section 4.2, we propose *test-time style shifting* to handle the issue on the large style gap between source and target domains. A high-level description of our idea is shown in Fig. 3. The pseudo codes are provided in Appendix A.11.

4.1 Style Balancing

We first describe style balancing which strategically shifts the style (i.e., style statistics) of each train sample to another source domain (that has insufficient amount of data samples for each class), to handle the cross-domain imbalance issues. Given a mini-batch, style balancing is applied to each class $k \in \{1, 2, \dots, K\}$ independently. Hence, we describe our scheme focusing on class k.

Step 1: Determining the number of samples to be shifted. To focus on class k, given a specific mini-batch, we define $\tilde{S}_{n,k}$ as the set of samples that belong to source domain n labeled as class k. We would like to balance the number of samples across all source domains $n \in \{1, 2, \dots, N\}$ so that each domain has average number of samples $Q_k := \frac{1}{N}\sum_{n=1}^{N} |\tilde{S}_{n,k}|$ for class k. If $|\tilde{S}_{n,k}| > Q_k$ holds, $|\tilde{S}_{n,k}| - Q_k$ samples in source domain n should shift their styles to other domains that have less than Q_k samples. Otherwise (i.e., $|\tilde{S}_{n,k}| < Q_k$), we similarly shift the styles of samples in other domains (that have more than Q_k samples) to domain n. Based on this, one can easily determine the number of samples to be shifted from domain n to another domain n' for all $n, n' \in \{1, 2, \dots, N\}$, in order to balance class k across all source domains.

Step 2: Sample selection. In this step, for domain n satisfying $|\tilde{S}_{n,k}| > Q_k$, we propose a strategy for selecting $|\tilde{S}_{n,k}| - Q_k$ samples to be shifted from domain n to other source domains.

Key insight: Our key insight is that samples having similar style statistics would provide similar effects on improving domain diversity, when existing DG schemes are applied. Based on this intuition, we propose to move the style of the sample that has very similar style statistics with other samples.

We first define the distance between the style statistics of any two samples $s_i, s_j \in \tilde{S}_{n,k}$ as

$$d_{i,j} = \|\Phi(f(s_i)) - \Phi(f(s_j))\|, \tag{6}$$

where $\|\cdot\|$ denotes the Euclidean distance. Now we choose two samples s_{i^*} and s_{j^*} from $\tilde{S}_{n,k}$ that satisfy $(i^*, j^*) = \text{argmin}_{(i,j)} d_{i,j}$; these two samples have the closest style statistics so that similar effect can be observed even when one of these samples are removed from source domain n. Among these two samples, we choose the sample that has a smaller minimum distance with other samples, and shift its style to another domain; we choose sample s_{i^*} if $\min\{d_{z,i^*}\}_{z=1,z\neq j^*}^{|\tilde{S}_{n,k}|} < \min\{d_{z,j^*}\}_{z=1,z\neq i^*}^{|\tilde{S}_{n,k}|}$ and choose sample s_{j^*}, otherwise. This process is repeated until $|\tilde{S}_{n,k}| - Q_k$ samples are selected from source domain n. We repeat this process for all source domains $n \in \{1, 2, \dots, N\}$.

Step 3: Balancing. Suppose sample s in domain n has to shift its style to domain n', according to Steps 1 and 2 above. We randomly select two samples $s'_1, s'_2 \in S_{n'}$ from domain n' and shift the style of s to s'_1, s'_2 by using EFDM, and apply EFDMix. Specifically, our style balancing (SB) performs

$$\text{SB}(f(s))_{\tau_i} = \lambda f(s'_1)_{\kappa_i} + (1-\lambda) f(s'_2)_{\eta_i} + f(s)_{\tau_i} - \langle f(s)_{\tau_i} \rangle, \tag{7}$$

where τ_i, κ_i, η_i are the indices of the i-th smallest elements of vectors $f(s)$, $f(s'_1)$, $f(s'_2)$, respectively. $\langle\cdot\rangle$ is the stop gradient operation; $\langle f(s)\rangle$ denotes the copy of $f(s)$ detached from computational graph. The term $f(s) - \langle f(s)\rangle$ is introduced to facilitate backpropagation of sample s as in (Zhang et al., 2022). This process eventually shifts the style of sample s in source domain n to source domain n'. As in MixStyle and EFDMix, λ is a mixing parameter which is sampled from the Beta distribution.

The above three steps are applied to the samples in each class $k \in \{1, 2, \ldots, K\}$ independently. By balancing the number of samples for each class across all source domains, our style balancing provides a great platform for the model to get exposed to various styles per classes during traininig.

4.2 Test-Time Style Shifting

In order to improve the model performance during testing, we propose test-time style shifting. If the test sample has a large style gap with the source domains, then the style of the test sample is shifted to the nearest source domain that the model is already familiar with. Otherwise, i.e., when the style gaps between the test sample and the sources are small, the test sample keeps its original style.

Let $t \in T$ be the test sample from an arbitrary unseen domain in test set T, where $f(t)$ is the encoded features of t at a specific layer. Recall that $\mu(f(t))$ and $\sigma(f(t))$ are the channel-wise mean and standard deviation of $f(t)$. Also recall that the mean of feature statistics in each source domain $n \in \{1, 2, \ldots, N\}$ are written as $\mu_{S_n} = \frac{1}{|S_n|}\sum_{s\in S_n}\mu(f(s))$ and $\sigma_{S_n} = \frac{1}{|S_n|}\sum_{s\in S_n}\sigma(f(s))$. We also define the mean feature statistics averaged over all source domains as $\mu_S = \frac{1}{N}\sum_{n=1}^{N}\mu_{S_n}$ and $\sigma_S = \frac{1}{N}\sum_{n=1}^{N}\sigma_{S_n}$. According to the definition in (5), we have $\Phi_{S_n} = [\mu_{S_n}, \sigma_{S_n}]$, $\Phi_S = [\mu_S, \sigma_S]$.

Based on these notations, at a specific layer, we generate new style statistics of sample t as

$$\Phi(f(t))_{\text{new}} \leftarrow \begin{cases} \Phi_{S_{n'}} & \text{if } \frac{1}{N}\sum_{n=1}^{N}\|\Phi(f(t)) - \Phi_{S_n}\| > \alpha\Big(\frac{1}{N}\sum_{n=1}^{N}\|\Phi_S - \Phi_{S_n}\|\Big), \\ \Phi(f(t)) & \text{otherwise,} \end{cases} \tag{8}$$

where $\Phi(f(t))_{\text{new}} = [\mu(f(t))_{\text{new}}, \sigma(f(t))_{\text{new}}]$, n' is the index of the closest source domain to the test sample t, i.e., $n' = \text{argmin}_n\|\Phi(f(t)) - \Phi_{S_n}\|$, and α is a hyperparameter greater than or equal to 0.

Now based on $\mu(f(t))_{\text{new}}$ and $\sigma(f(t))_{\text{new}}$, following the process of AdaIN in (2), our test-time style shifting (TS) shifts the style of sample t while preserving its content at the corresponding layer as

$$\text{TS}(f(t)) = \sigma(f(t))_{\text{new}}\frac{f(t) - \mu(f(t))}{\sigma(f(t))} + \mu(f(t))_{\text{new}}. \tag{9}$$

Intuitions. In (8), if there is a large gap between style statistics of source domains and the test sample, we shift the style statistics of the test sample to the *nearest source domain*. This enables predictions on the domain that the model is already familiar with. Otherwise, i.e., when the gap between the style statistics of source domains and the test sample is acceptable, the model is likely to be well-trained on the style of the test sample; thus, we let the test sample t to keep its current style statistics. A comprehensive study on our test-time style shifting is provided in Appendix A.3.

4.3 Overall Procedure and Discussions

The overall procedure of our algorithm is shown in Fig. 3. Given imbalanced style statistics, we first perform style balancing. Then, we can apply any DG methods for training (e.g., style augmentation). When training is finished, we apply our test-time style shifting and make a prediction.

Where to apply SB and TS. Our style balancing (SB) and test-time style shifting (TS) can be flexibly applied at any layer of the backbone. During training, we only have SB module, which is discarded when training is finished. During testing, TS module is applied at a predetermined layer. Various ablations and complexity analysis of our SB and TS modules are provided in Section 5 and Appendix.

Compatibility with various DG methods. The simplest way to combine our work with others is to apply style augmentation (e.g., MixStyle) after SB, which also work in the style space as our scheme. Our method can also work in conjunction with other DG strategies due to the high flexibility of SB and TS modules. For example, the proposed SB module can be applied at the inner optimization process of meta-learning DG approach (Li et al., 2018a) to handle the imbalance issues in the meta-train source domains. As another example, our SB can be applied at the feature learning network of

Table 2: Effect of style balancing (SB) and test-time style shifting (TS) on **original PACS** using ResNet-18. We reproduced the results of MixStyle, DSU, EFDMix while other values are from original papers (denoted with *).

Methods	Reference	Art	Cartoon	Photo	Sketch	Average
L2A-OT* (Zhou et al., 2020)	ECCV'20	83.3	78.2	96.2	73.6	82.8
pAdaIN* (Nuriel et al., 2021)	CVPR'21	81.74	76.91	96.29	75.13	82.51
SagNet* (Nam et al., 2021)	CVPR'21	83.58	77.66	95.47	76.3	83.25
Baseline (ResNet-18)	-	73.97	74.71	96.07	65.71	77.62
Baseline + SB	Ours	80.55	77.16	96.39	71.68	81.44
Baseline + TS	Ours	73.89	75.14	95.87	72.00	79.23
Baseline + SB + TS	Ours	80.60	77.58	96.35	74.37	**82.22**
MixStyle (Zhou et al., 2021)	ICLR'21	82.54	79.42	95.88	74.06	82.98
MixStyle + SB	Ours	83.48	79.07	96.15	73.74	83.11
MixStyle + TS	Ours	82.59	79.99	95.88	78.66	84.28
MixStyle + SB + TS	Ours	83.62	80.07	96.15	78.66	**84.63**
DSU (Li et al., 2022)	ICLR'22	81.78	78.66	95.91	76.75	83.27
DSU + SB	Ours	80.98	79.61	95.95	78.66	83.80
DSU + TS	Ours	81.12	80.31	95.82	79.19	84.11
DSU + SB + TS	Ours	80.73	80.69	95.83	79.47	**84.18**
EFDMix (Zhang et al., 2022)	CVPR'22	83.40	79.87	96.43	74.49	83.55
EFDMix + SB	Ours	83.32	79.47	96.59	74.42	83.45
EFDMix+ TS	Ours	83.41	81.41	96.25	78.40	**84.87**
EFDMix + SB + TS	Ours	83.33	80.56	96.55	78.61	84.77

conditional invariant deep DG method (Li et al., 2018c). For all methods, we can apply our TS at a specific layer of the network during testing. In Section 5, we show via experiments that our SB and TS are compatible not only with style augmentation based schemes but also with other DG methods relying on meta-learning or domain alignment, and improve the model performance.

Hyperparameters. In our SB module, note that the mixing parameter λ in (7) is sampled from Beta distribution as $\lambda \sim Beta(\tau, \tau)$. This parameter also appears in MixStyle (Zhou et al., 2021) and EFDMix (Zhang et al., 2022), and we set $\tau = 0.1$ for all experiments as in these prior works. Compared to existing style augmentation methods, our scheme requires an additional hyperparameter α that appears in (8) of our TS module, which is set to 3 for all classification results. A more detailed discussion regarding α is provided in Appendix A.5.

Our SB and TS work in a highly complementary fashion to handle the two issues at the same time. In the next section, we show that (i) adopting these two components can significantly boost up the performance of recent DG strategies, and (ii) removing one of these components can degrade the performance in practical scenarios having both issues at the same time.

5 Experimental Results

5.1 Generalization on Multi-Domain Classification

Experimental setup. Targeting multi-domain classification, we perform experiments using PACS (Li et al., 2017) with 4 domains (Art, Cartoon, Photo, Sketch) and VLCS (Fang et al., 2013) with 4 domains (Caltech, LabelMe, Pascal, Sun), which are the commonly adopted benchmarks for DG. We consider the leave-one-domain-out setting where the model is trained on three domains and tested on the remaining one domain. Following the setups in (Zhou et al., 2021; Li et al., 2022; Zhang et al., 2022), we adopt ResNet-18 pre-trained on ImageNet as a backbone. For PACS, the proposed SB module is probabilistically operated once at first or second or third residual blocks during training, while the TS module is operated at the second residual block during testing. Other implementation details and ablations on SB/TS locations are provided in Appendix. We consider not only the original PACS and VLCS datasets but also the imbalanced/reduced version of each dataset. To model the cross-domain data imbalance scenario, we keep the training data of the largest source domain while removing 80% of the training data of the remaining two domains. When constructing the cross-domain class-imbalanced dataset, among 7 classes in PACS, we select 3 classes from the first source domain, other 2 classes from the second source domain, and the remaining 2 classes from the last source domain. In VLCS, among 5 classes, we select 2, 2, 1 classes from each source domain to construct the imbalanced dataset. The class imbalanced dataset could be also constructed in different settings, e.g., in a long-tailed imbalance setting (Cao et al., 2019). The corresponding results are reported in Appendix A.10. The performance is obtained by averaging the results over 3 independent trials. More details on our experimental setup are provided in Appendix.

Table 3: Effect of style balancing (SB) and test-time style shifting (TS) on **imbalanced PACS**. Compared to the result in Table 2, the role of SB becomes more significant in severely imbalanced scenarios.

Methods	Reference	Cross-domain data imbalance					Cross-domain class imbalance				
		Art	Cartoon	Photo	Sketch	Avg.	Art	Cartoon	Photo	Sketch	Avg.
MixStyle	ICLR'21	71.73	73.80	90.60	66.48	75.65	39.91	54.08	56.45	44.82	48.82
MixStyle + SB	Ours	76.53	75.61	93.33	68.34	78.45	44.49	55.57	56.28	44.93	50.32
MixStyle + TS	Ours	72.04	74.01	90.60	75.12	77.94	39.98	54.01	56.45	44.44	48.74
MixStyle + SB + TS	Ours	76.97	76.62	93.29	75.88	**80.69**	44.50	55.84	56.28	46.68	**50.83**
DSU	ICLR'22	75.76	75.26	91.90	72.45	78.84	29.61	45.24	46.90	39.37	40.28
DSU + SB	Ours	76.04	76.15	92.87	73.47	79.64	45.09	53.93	60.25	47.74	51.75
DSU + TS	Ours	75.49	76.69	91.92	76.36	80.12	29.78	44.54	46.90	36.65	39.47
DSU + SB + TS	Ours	75.93	77.39	92.85	75.90	**80.52**	45.03	54.42	60.24	49.20	**52.22**
EFDMix	CVPR'22	75.33	75.67	90.59	71.07	78.16	44.68	54.87	58.15	44.64	50.59
EFDMix + SB	Ours	77.91	76.38	92.79	70.99	79.52	46.63	54.84	57.89	44.47	50.96
EFDMix+ TS	Ours	75.39	75.92	90.56	74.97	79.21	44.56	55.05	58.15	45.96	50.93
EFDMix + SB + TS	Ours	77.90	76.54	92.71	76.37	**80.88**	46.03	55.29	57.87	49.99	**52.30**

Table 4: Performance on **imbalanced VLCS**.

Methods	Reference	Caltech	LabelMe	Pascal	Sun	Average
MixStyle (Zhou et al., 2021)	ICLR'21	68.87	53.32	55.12	39.09	54.10
MixStyle + SB	Ours	69.97	53.87	55.51	38.51	54.47
MixStyle + TS	Ours	73.51	53.20	55.15	38.98	**55.21**
MixStyle + SB + TS	Ours	73.27	53.78	55.02	38.58	55.16
DSU (Li et al., 2022)	ICLR'22	63.07	54.13	56.01	39.90	53.28
DSU + SB	Ours	74.02	53.40	55.91	40.22	55.89
DSU + TS	Ours	65.99	53.90	55.93	40.02	53.96
DSU + SB + TS	Ours	75.99	53.50	55.46	40.28	**56.31**

Baselines. First, we consider the state-of-the-art style augmentation schemes, MixStyle (Zhou et al., 2021), DSU (Li et al., 2022), EFDMix (Zhang et al., 2022), that also work in the style space as ours. Built upon each method, we apply our SB and TS to validate the effectiveness of the proposed ideas. For a fair comparison, all hyperparemters are set to be same as in the original setup of each baseline. We also apply our methods to the pure baseline without any DG algorithm. We compare the performance with other recent works on DG: L2A-OT (Zhou et al., 2020), pAdaIN (Nuriel et al., 2021), SagNet (Nam et al., 2021). Finally, to confirm the compatibility with other DG methods, we also apply our SB/TS to MLDG (Li et al., 2018a) and CDANN (Li et al., 2018c) in Section 5.3.

Result 1: Original dataset. We first observe Table 2, which shows the results on original PACS. Both SB and TS play important roles in all baselines. The performance gain of SB is noticeable since PACS is already slightly imbalanced across domains. The performance gain of TS is especially large in Sketch, since the Sketch domain has a large style gap with other source domains (see Fig. 2). The overall results show that our scheme significantly boosts up the performance of recent style-augmentation methods. Our scheme also outperforms other recent methods for DG.

Result 2: Cross-domain data imbalanced dataset. In Table 3, we consider two different imbalanced versions of PACS. Since less training data is used in Table 3, the performance is generally degraded compared to the results in Table 2. We first observe the left part of Table 3, the data imbalance case. The advantage of SB is significant compared to the case in Table 2; the major performance gains of Art, Cartoon, Photo come from SB, showing the effectiveness of SB to improve the domain diversity during training. On the other hand, the main performance gain of Sketch comes from TB as in Table 2; again, this is because Sketch has a significant style gap with other three source domains as shown in Fig. 2. The overall results confirm the advantage of both SB and TS.

Result 3: Cross-domain class imbalanced dataset. In cross-domain class imbalance scenario (right part of Table 3), different from the trends in original dataset and cross-domain data imbalanced dataset, directly applying TB (without SB) does not improve the performance in general (even in Sketch). This is because the model trained without SB lack generalization capability in this scenario, indicating the importance of SB. The performance gain of SB is especially large when combined with DSU; compared to MixStyle or EFDMix, in DSU, each class tends to get exposed to only a limited styles and thus show limited performance. A meaningful gain is obtained when TS is applied after SB, again confirming the effectiveness of SB. Table 4 shows the performance on cross-domain class imbalanced VLCS. Although the performance gain is smaller compared to PACS due to the small style gaps of source and target domains, the trend is consistent with the results in PACS.

Table 5: Performance on **person re-ID task**, using Market1501 and GRID datasets.

Methods	Reference	Market → GRID				GRID → Market			
		mAP	R1	R5	R10	mAP	R1	R5	R10
MixStyle (Zhou et al., 2021)	ICLR'21	35.30	26.67	**44.53**	53.07	5.25	16.40	30.05	37.05
MixStyle + SB + TS	Ours	**36.30**	**28.27**	42.93	**55.47**	**5.70**	**17.75**	**31.90**	**39.65**
DSU (Li et al., 2022)	ICLR'22	38.57	30.40	46.40	53.07	4.45	14.90	27.65	34.60
DSU + SB + TS	Ours	**40.10**	**30.67**	**48.00**	**58.13**	**5.25**	**16.70**	**31.60**	**38.85**

5.2 Generalization on Instance Retrieval

We also consider a different task, known as multi-domain instance retrieval. We consider person re-identification (re-ID), where the goal is to match the same person using various camera views. This setup can be viewed as a multi-domain image matching problem by regarding different camera views as distinct domains. As in the setup of (Zhang et al., 2022), we adopt Market1501 (Zheng et al., 2015) and GRID (Loy et al., 2009) datasets, and train the model in one dataset and test on the other one. We train OSNet (Zhou et al., 2019) which was specifically designed for person re-ID. Other details are provided in Appendix. Table 5 shows the corresponding results, indicating that our idea is powerful even in multi-domain image matching problem.

5.3 Further Experiments and Discussions

Compatibility with other DG methods. Since our SB and TS modules are applicable to any CNN-based feature extractor, our scheme can also work effectively with other DG strategies based on meta-learning and domain alignment. Table 6 shows the results of MLDG (Li et al., 2018a) (meta-learning based method) and CDANN (Li et al., 2018c) (domain alignment based method) combined with our scheme on cross-domain class imbalanced PACS. We consider the DomainBed setup for experiments. It can be seen that our scheme improves the performance of both methods, confirming that both SB and TS can work in conjunction with various DG methods to mitigate the cross-domain imbalance and the style gap issues.

Table 6: Compatibility of our methods with other DG strategies on imbalanced PACS.

Methods	Accuracy
MLDG	40.26
MLDG + SB	53.28
MLDG + SB + TS	**53.54**
CDANN	25.61
CDANN + SB	46.21
CDANN + SB + TS	**46.22**

Comparison with existing class imbalance methods. In Table 7, we compare SB with existing class imbalanced learning methods in a cross-domain class imbalance scenario, under the same setup in Table 3. We consider the following baselines: undersampling majority classes, oversampling minority classes, reweighting the objective function based on the *effective number* (Cui et al., 2019). We also compare our method with the recent work (Yang et al., 2022) focusing on a multi-domain setup with imbalanced datasets, in Appendix. It can be seen that existing methods generally fail to handle the imbalance issues since the missing classes of each domain cannot be compensated via over/under-sampling or reweighting in this cross-domain class imbalance setup. Our SB effectively alleviates this issue, significantly improving the model performance.

Table 7: Comparison with existing class imbalance methods on imbalanced PACS.

Methods	Accuracy
DSU (Li et al., 2022)	40.28
DSU + SB	**51.75**
DSU Undersampling	40.37
DSU Undersampling + SB	**47.74**
DSU Oversampling	43.91
DSU Oversampling + SB	**54.01**
DSU Reweighting	41.57
DSU Reweighting + SB	**52.57**

Additional experimental results. Other results including results in a DomainBed setup, results without domain labels, results on long-tailed imbalance settings, results on other datasets are shown in Appendix. We also perform comprehensive studies on our SB and TS modules, in Appendix.

6 Conclusion

We proposed style balancing and test-time style shifting, new strategies that can handle the current issues in domain generalization. Style balancing provides a platform for the model to get exposed to various styles per classes during training, while test-time style shifting enables the model to make predictions on the familiar style regardless of the target domain. Our solution provides a new guideline for domain generalization in practice, where handling the imbalance issues and reducing the gap between the source and target domains are of paramount importance.

REPRODUCIBILITY STATEMENT

The detailed experimental setups of our experiments are described in Section 5.1 of the main manuscript and Appendix A.12. Our code is provided in Supplementary Material.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *arXiv preprint arXiv:1711.07910*, 2017.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34, 2021.

Chaoqi Chen, Jiongcheng Li, Xiaoguang Han, Xiaoqing Liu, and Yizhou Yu. Compound domain generalization via meta-knowledge encoding. *arXiv preprint arXiv:2203.13006*, 2022.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees GM Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. In *European Conference on Computer Vision*, pp. 200–216. Springer, 2020.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. 2017.

Sarah Erfani, Mahsa Baktashmotlagh, Masud Moshtaghi, Xuan Nguyen, Christopher Leckie, James Bailey, and Rao Kotagiri. Robust domain generalisation by enforcing distribution invariance. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pp. 1455–1461. AAAI Press, 2016.

Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1657–1664, 2013.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pp. 1322–1328. IEEE, 2008.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5375–5384, 2016.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*, pp. 124–140. Springer, 2020.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34, 2021.

Juwon Kang, Sohyun Lee, Namyup Kim, and Suha Kwak. Style neophile: Constantly seeking novel styles for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7130–7140, 2022.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018a.

Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1446–1455, 2019.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018b.

Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and Lingyu Duan. Uncertainty modeling for out-of-distribution generalization. In *International Conference on Learning Representations*, 2022.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 624–639, 2018c.

Chen Change Loy, Tao Xiang, and Shaogang Gong. Multi-camera activity correlation analysis. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1988–1995. IEEE, 2009.

Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8690–8699, 2021.

Oren Nuriel, Sagie Benaim, and Lior Wolf. Permuted adain: reducing the bias towards global statistics in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9482–9491, 2021.

Prashant Pandey, Mrigank Raman, Sumanth Varambally, and Prathosh Ap. Generalization on unseen domains via inference-time label-preserving target projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12924–12933, 2021.

Samira Pouyanfar, Yudong Tao, Anup Mohan, Haiman Tian, Ahmed S Kaseb, Kent Gauen, Ryan Dailey, Sarah Aghajanzadeh, Yung-Hsiang Lu, Shu-Ching Chen, et al. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pp. 112–117. IEEE, 2018.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*, 2018.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pp. 9229–9248. PMLR, 2020.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2020.

Zehao Xiao, Xiantong Zhen, Ling Shao, and Cees GM Snoek. Learning to generalize across domains on single test samples. *arXiv preprint arXiv:2202.08045*, 2022.

Yuzhe Yang, Hao Wang, and Dina Katabi. On multi-domain long-tailed recognition, generalization and beyond. *arXiv preprint arXiv:2203.09513*, 2022.

Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2100–2110, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Marvin Mengxin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group shift. 2020.

Yabin Zhang, Minghan Li, Ruihuang Li, Kui Jia, and Lei Zhang. Exact feature distribution matching for arbitrary style transfer and domain generalization. *arXiv preprint arXiv:2203.07740*, 2022.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6277–6286, 2021.

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pp. 1116–1124, 2015.

Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3702–3712, 2019.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European conference on computer vision*, pp. 561–578. Springer, 2020.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021.

A APPENDIX

A.1 COMPARISON WITH OTHER STATE-OF-THE-ARTS IN DOMAINBED SETUP

Following the DomainBed setup (Gulrajani & Lopez-Paz, 2021), in Table 8, we compare our approach with other state-of-the-arts using ResNet-50. Training-domain validation strategy is used for selecting the model in DomainBed setup. It can be seen that the proposed scheme combined with MixStyle achieves the best performance with average accuracy of 86.6%. We also combine our scheme with one of the state-of-the-art benchmarks, termed SWAD (Cha et al., 2021). It is shown that our scheme can further improve the performance of the existing method. The overall results in Table 8 show that our style balancing (SB) and test-time style shifting (TS) can be easily combined with other state-of-the-arts to achieve the best performance.

Table 8: Performance in DomainBed setup.

Methods	Art	Cartoon	Photo	Sketch	Average
ERM (Vapnik, 1999)	84.7	80.8	97.2	79.3	85.5
IRM (Arjovsky et al., 2019)	84.8	76.4	96.7	76.1	83.5
GroupDRO (Sagawa et al., 2019)	83.5	79.1	96.7	78.3	84.4
Mixup (Zhang et al., 2017)	86.1	78.9	97.6	75.8	84.6
MLDG (Li et al., 2018a)	85.5	80.1	97.4	76.6	84.9
CORAL (Sun & Saenko, 2016)	88.3	80.0	97.5	78.8	86.2
MMD (Li et al., 2018c)	86.1	79.4	96.6	76.5	84.6
DANN (Ganin et al., 2016)	86.4	77.4	97.3	73.5	83.6
CDANN (Li et al., 2018c)	84.6	75.5	96.8	73.5	82.6
MTL (Blanchard et al., 2017)	87.5	77.1	96.4	77.3	84.6
SagNet (Nam et al., 2021)	87.4	80.7	97.1	80.0	86.3
ARM (Zhang et al., 2020)	86.8	76.8	97.4	79.3	85.1
VREx (Krueger et al., 2021)	86.0	79.1	96.9	77.7	84.9
RSC (Huang et al., 2020)	85.4	79.7	97.6	78.2	85.2
EFDMix (Zhang et al., 2022),	86.7	80.3	96.3	80.8	86.0
MixStyle (Zhou et al., 2021)	85.6	80.6	95.5	81.6	85.8
MixStyle + SB (ours)	87.8	82.1	95.6	81.0	**86.6**
Combination with SWAD					
SWAD (Cha et al., 2021)	89.3	83.4	**97.3**	82.5	88.1
SWAD + MixStyle	90.3	84.4	97.2	85.0	89.2
SWAD + MixStyle + SB + TS (ours)	**90.8**	**84.5**	97.1	**85.4**	**89.4**

A.2 ABLATION STUDIES ON STYLE BALANCING

We first provide ablations studies on our style balancing (SB) module.

Effect of proposed sample selection method. In Step 2 of our style balancing procedure, we proposed to move the style of the sample that has very similar statistics with other samples. To validate the effectiveness of this idea, here we provide results with random sample selection; for domain n satisfying $|\tilde{S}_{n,k}| > Q_k$, we randomly select $|\tilde{S}_{n,k}| - Q_k$ samples to shift their styles to other domains. Tables 9 and 10 show the results in domain imbalance scenario and cross-domain class imbalance scenario, respectively. The setup is exactly the same as in the main manuscript. The results of both Tables 9 and 10 confirm the effectiveness of our sample selection strategy in SB compared to the random sampling strategy. The gain is especially large in cross-domain class imbalance scenario.

Where to shift samples. Suppose sample s shifts its style from domain n to domain n' in Step 3 of our style balancing procedure. In the main manuscript, we randomly selected two samples from domain n' and mixed the feature statistics of these samples via EFDMix (Zhang et al., 2022), and shifted the style statistics of sample s to this mixed style. Here we consider another baseline for ablation study: we randomly sample the new style statistics from Gaussian distribution with mean and variance computed from the samples in domain n', and shift the style of sample s via AdaIN

Table 9: Effect of the proposed sample selection method in style balancing (SB) in **domain imbalanced PACS**.

Methods	Art	Cartoon	Photo	Sketch	Average
MixStyle + random sampling based SB	76.25	75.49	93.05	68.00	78.20
MixStyle + proposed SB	76.53	75.61	93.33	68.34	**78.45**
MixStyle + random sampling based SB + TS	76.58	76.17	93.05	75.91	80.43
MixStyle + proposed SB + TS	76.97	76.62	93.29	75.88	**80.69**

Table 10: Effect of the proposed sample selection method in style balancing (SB) in **cross-domain class imbalanced PACS**.

Methods	Art	Cartoon	Photo	Sketch	Average
MixStyle + random sampling based SB	43.37	54.02	54.91	45.74	49.51
MixStyle + proposed SB	44.49	55.57	56.28	44.93	**50.32**
MixStyle + random sampling based SB + TS	43.43	54.32	54.91	44.60	49.32
MixStyle + proposed SB + TS	44.50	55.84	56.28	46.68	**50.83**

(Huang & Belongie, 2017) to this new style. Table 11 compares our method with this SB variant, confirming the advantage of the proposed strategy.

Table 11: Comparison with another style balancing (SB) variant in original PACS.

Methods	Art	Cartoon	Photo	Sketch	Average
MixStyle + SB variant	82.83	79.24	95.69	73.18	82.73
MixStyle + proposed SB	83.48	79.07	96.15	73.74	**83.11**
MixStyle + SB variant + TS	82.91	79.95	95.69	79.07	84.40
MixStyle + proposed SB + TS	83.62	80.07	96.15	78.66	**84.63**

A.3 Ablation Studies on Test-Time Style Shifting

In this section, we provide ablation studies on our test-time style shifting (TS) module.

Variants of test-time style shifting. We investigate the performance of other possible variants of TS. We consider two additional strategies for TS: first, instead of only shifting the style of the test samples that have large style gaps with the source domains (as in the main manuscript), we consider a scheme that *shifts the styles of all samples* to the nearest source domain (TS variant 1). We also consider a scheme that shifts the style of the sample to the nearest sample among randomly selected 100 samples, based on the condition in equation (8) of the main manuscript (TS variant 2). Table 12 compares the results of variants of TS. It can be first seen that TS variant 2 has lower performance compared to others, indicating that shifting the style to the nearest sample is less effective compared to the scheme that shifts the style to the nearest center of the source domain. Shifting all the samples (TS variant 1) can improve the performance on Cartoon or Sketch domains, but suffers from performance degradation on Art or Photo; this indicates that it is better to keep the sample's original style when the gap with the source domain is small. In general, TS variant 1 achieves similar or lower performance compared to our TS strategy.

Location of TS module. In the main manuscript, our TS module is applied at the output of the 2^{nd} residual block of ResNet-18, when training with PACS dataset. In Table 13, we applied the proposed TS module at different residual blocks. It is observed that applying TS module after the 1^{st} block or the 2^{nd} block or the 3^{rd} block improves the performance. However, operating our TS module after the 4^{th} residual block significantly degrades the performance, which is straightforward since data are clustered according to the classes (regardless of the domains) at later layers.

Table 12: Comparison with other test-time style shifting (TS) variants in original PACS.

Methods	Art	Cartoon	Photo	Sketch	Average
MixStyle + SB + TS variant 1 (shift all samples)	82.71	81.66	95.55	78.81	**84.68**
MixStyle + SB + TS variant 2 (shift to the nearest sample)	83.60	79.57	96.15	77.25	84.14
MixStyle + SB + proposed TS	83.62	80.07	96.15	78.66	**84.63**
DSU + SB + TS variant 1 (shift all samples)	79.75	80.18	94.80	79.49	83.55
DSU + SB + TS variant 2 (shift to the nearest sample)	80.58	80.14	95.83	77.92	83.62
DSU + SB + proposed TS	80.73	80.69	95.83	79.47	**84.18**

Table 13: Effect of location of test-time style shifting (TS) module in original PACS.

Methods	Art	Cartoon	Photo	Sketch	Average
MixStyle + SB	83.48	79.07	96.15	73.74	83.11
MixStyle + SB + TS (output of 1^{st} residual block)	83.50	79.11	96.15	75.67	83.61
MixStyle + SB + TS (output of 2^{nd} residual block)	83.62	80.07	96.15	78.66	**84.63**
MixStyle + SB + TS (output of 3^{rd} residual block)	83.66	79.80	96.09	77.85	84.35
MixStyle + SB + TS (output of 4^{th} residual block)	18.51	25.60	18.84	17.89	20.21

A.4 Results without Domain Labels

Throughout the main manuscript, we described our algorithm using domain labels. In Table 14, we show the performance of our scheme without any domain labels. Here, we provide pseudo domain labels using k-means clustering, where k is set to be 3. We apply our SB and TS by utilizing the clustered domains with pseudo labels. We let $\alpha = 2$ throughout all experiments in Table 14. Experimental results show that both SB and TS are effective even without any domain labels. The performance of TS *without* domain labels is sometimes even better compared to the case *with* domain labels. This indicates that it is more important to consider how the train samples are clustered in the style space, rather than the original domain label, during the TS process.

Table 14: Performance without domain label on original PACS.

Methods	Reference	Art	Cartoon	Photo	Sketch	Average
MixStyle (Zhou et al., 2021)	ICLR'21	82.65	78.84	96.09	72.23	82.45
MixStyle + SB	Ours	83.72	79.34	96.43	73.22	83.18
MixStyle + TS	Ours	83.10	80.99	96.15	78.11	84.59
MixStyle + SB + TS	Ours	83.61	81.79	96.31	79.03	**85.19**
DSU (Li et al., 2022)	ICLR'22	81.78	78.66	95.91	76.75	83.27
DSU + SB	Ours	81.92	79.14	95.95	78.54	83.89
DSU + TS	Ours	80.16	79.37	94.91	78.97	83.35
DSU + SB + TS	Ours	81.59	80.01	95.19	79.16	**83.99**
EFDMix (Zhang et al., 2022)	CVPR'22	83.35	79.91	96.67	74.52	83.61
EFDMix + SB	Ours	83.38	80.22	96.81	75.13	83.89
EFDMix+ TS	Ours	83.43	81.25	96.26	78.92	84.96
EFDMix + SB + TS	Ours	83.80	81.57	96.49	79.05	**85.23**

A.5 Effect of α

Recall that α is a hyperparameter that appears in equation (8) of the main manuscript. In the main manuscript, we set $\alpha = 3$ for all experiments for PACS and VLCS. However, this value may not be the optimal value for each domain/setup. In Table 15, we provide results on various α values. When α is large ($\alpha = 5$), most of the test samples do not shift their styles; this reduces to the scheme with only SB. When $\alpha = 0$, all the test samples move their styles to the nearest source domain, which can degrade the performance of specific domains (Art and Photo) but improves the performance of Cartoon and Sketch. One can also select the α value by considering the extended validation set at feature-level; one can additionally generate new styles that have large style gaps with the current

source domains, so that the extended set contains both samples that have small/large style gaps with the source domains. Nevertheless, whatever α we choose, we have additional performance improvement (or at least the same performance) compared to the case with no TS, confirming the advantage of our TS module.

Table 15: Performance with varying α on original PACS: *whatever α we choose, an additional performance gain can be obtained compared to no TS.*

Methods	Reference	Art	Cartoon	Photo	Sketch	Average
MixStyle (Zhou et al., 2021)	ICLR'21	82.54	79.42	95.88	74.06	82.98
MixStyle + SB	Ours	83.48	79.07	**96.15**	73.74	83.11
MixStyle + SB + TS ($\alpha = 0$)	Ours	82.71	81.66	95.55	**78.81**	84.68
MixStyle + SB + TS ($\alpha = 2$)	Ours	83.31	**81.81**	96.01	**78.81**	**84.99**
MixStyle + SB + TS ($\alpha = 3$)	Ours	**83.62**	80.07	**96.15**	78.66	84.63
MixStyle + SB + TS ($\alpha = 4$)	Ours	83.48	79.10	**96.15**	73.81	83.13
MixStyle + SB + TS ($\alpha = 5$)	Ours	83.48	79.07	**96.15**	73.74	83.11

A.6 Experiments on Office-Home Dataset

In addition to the results on PACS, VLCS, Market1501 and GRID in the main manuscript, in Table 16, we provide additional results on Office-Home dataset (Venkateswara et al., 2017) with 4 domains and 65 classes. We can observe a performance gain via SB even in the original Office-Home dataset. The performance gain of TS is marginal since the style gaps between domains are relatively small in Office-Home. Nevertheless, existing schemes can still benefit from the proposed SB and TS modules.

Table 16: Performance on original Office-Home dataset.

Methods	Reference	Art	Clipart	Product	Real world	Average
MixStyle (Zhou et al., 2021)	ICLR'21	57.99	53.04	73.64	74.98	64.91
MixStyle + SB	Ours	58.29	53.20	74.01	75.29	65.20
MixStyle + SB + TS	Ours	58.27	53.41	74.05	75.33	**65.27**

A.7 Comparison with BoDA

In this subsection, we compare our SB with BoDA (Yang et al., 2022), which is a recent work that reduces the domain representation gap for each class in a multi-domain imbalanced setup. Specifically, in BoDA, a new loss function is proposed to reduce the distance between a sample and the means of samples in different domains (that the sample does not belong to) for each class, in the representation space. For a fair comparison, we implemented BoDA in our experimental setup. We set the hyperparameters in BoDA to the default values in the original paper (Yang et al., 2022) which we found to achieve the best performance. In this experiment, we construct a cross-domain class imbalanced PACS where each domain has 5 classes out of a total of 7 (the missing classes of each domain are different between domains). As shown in Table 17, our SB with Baseline and Mixstyle shows better performance compared to the BoDA. This is because when a specific domain does not have certain classes, BoDA is not able to consider the mean of that domain for the missing class in computing the loss function of BoDA, reducing the generalization capability of the class. On the other hand, our SB can effectively improve the generalization performance by compensating for the missing class of each domain. Moreover, we stress that in the setup of Table 7, the loss function of BoDA cannot be defined due to lack of classes in most of the domains, making BoDA not applicable.

Table 17: Comparison with BoDA (Yang et al., 2022) in **cross-domain class imbalanced PACS** on ResNet-18.

Methods	Art	Cartoon	Photo	Sketch	Average
BoDA	53.56	63.27	94.97	60.92	68.18
Baseline (ResNet-18) + proposed SB	65.04	64.63	95.63	67.85	73.29
MixStyle + proposed SB	66.26	64.46	94.97	71.44	**74.28**

A.8 ADDITIONAL EXPERIMENTS USING RESNET-50

In Table 18, we show the results using ResNet-50. Other setups are exactly the same as in the main manuscript with ResNet-18. The results are consistent with all previous results, confirming the strong advantages of our SB and TS modules.

Table 18: Performance comparison using ResNet-50 on original PACS.

Methods	Reference	Art	Cartoon	Photo	Sketch	Average
MixStyle (Zhou et al., 2021)	ICLR'21	89.42	81.94	97.82	76.04	86.31
MixStyle + SB	Ours	89.782	81.71	97.80	75.93	86.31
MixStyle + SB + TS	Ours	89.92	81.77	97.80	80.20	**87.42**
DSU (Li et al., 2022)	ICLR'22	88.52	82.32	97.17	76.42	86.11
DSU + SB	Ours	88.05	82.90	97.62	80.08	87.16
DSU + SB + TS	Ours	88.11	82.94	97.59	82.04	**87.67**
EFDMix (Zhang et al., 2022)	CVPR'22	89.68	82.10	97.84	78.37	87.00
EFDMix + SB	Ours	90.08	81.75	97.72	78.17	86.93
EFDMix + SB + TS	Ours	90.1 4	81.80	97.66	81.16	**87.69**

A.9 ADDITIONAL EXPERIMENTS FOR INSTANCE RETRIEVAL

In this section, we provide the full version of Table 4 in the main manuscript. Table 19 shows the corresponding result, confirming the effectiveness of the proposed style balancing and test-time style shifting strategies for instance retrieval, especially when they are used together.

Table 19: Performance on person re-ID task, using Market1501 and GRID datasets.

Methods	Reference	Market → GRID				GRID → Market			
		mAP	R1	R5	R10	mAP	R1	R5	R10
MixStyle	ICLR'21	35.30	26.67	**44.53**	53.07	5.25	16.40	30.05	37.05
MixStyle + SB	Ours	35.73	27.73	42.93	52.00	**5.70**	17.70	**31.90**	**39.65**
MixStyle + TS	Ours	34.83	25.60	43.73	50.67	5.25	16.40	30.05	37.10
MixStyle + SB + TS	Ours	**36.30**	**28.27**	42.93	**55.47**	**5.70**	**17.75**	**31.90**	**39.65**
DSU	ICLR'22	38.57	30.40	46.40	53.07	4.45	14.90	27.65	34.60
DSU + SB	Ours	**41.47**	**33.33**	**48.80**	54.93	**5.25**	**16.75**	**31.65**	**38.85**
DSU + TS	Ours	37.27	28.00	46.13	55.73	4.40	14.75	27.35	34.60
DSU + SB + TS	Ours	40.10	30.67	48.00	**58.13**	**5.25**	16.70	31.60	**38.85**

A.10 ADDITIONAL EXPERIMENTS IN LONG-TAILED IMBALANCE SETTING

We have performed additional experiments on the long-tailed imbalance setting where the results are provided in Table 20. The imbalance ratio, which represents the ratio between sample sizes of the most frequent and least frequent class, is set to 64. The results are consistent with the ones in the main manuscript.

The results are consistent with the ones in our original manuscript, confirming the effectiveness of our algorithm in various imbalance scenarios including the setup in (Cao et al., 2019). These results are also provided in Table 2 of Appendix.

A.11 ALGORITHM IN PSEUDO CODE

Algorithm 1 shows the sample selection process in style balancing. The process for test-time style shifting is provided in Algorithm 2.

Table 20: Experimental results on long-tailed imbalance setting.

Methods	Reference	Art	Cartoon	Photo	Sketch	Average
MixStyle	ICLR'21	73.49	76.75	86.17	62.73	74.79
MixStyle + SB	Ours	76.46	75.30	88.20	61.81	75.44
MixStyle + TS	Ours	73.68	76.75	86.17	69.04	77.53
MixStyle + SB + TS	Ours	77.25	75.64	88.20	69.04	**77.53**
DSU	ICLR'22	75.47	76.01	89.31	60.81	75.40
DSU + SB	Ours	73.66	76.17	90.93	67.51	77.07
DSU + TS	Ours	74.54	76.43	89.16	66.55	76.67
DSU + SB + TS	Ours	73.27	76.09	90.78	68.92	**77.26**
.						

Algorithm 1 Sample Selection Process in Style Balancing (SB)

Input: $\tilde{S}_{n,k}$ (samples in domain n with class k, in a mini-batch) satisfying $|\tilde{S}_{n,k}| > Q_k$
Output: $Z_{n,k}$, which contains $|\tilde{S}_{n,k}| - Q_k$ samples (with class k) to be shifted from domain n to other source domains

$Z_{n,k} = \emptyset$, $E = 0$
while $E < |\tilde{S}_{n,k}| - Q_k$ **do**
  for all $s_i, s_j \in \tilde{S}_{n,k}$ $(i \neq j)$ **do**
    Compute $d_{i,j} = \|\Phi(f(s_i)) - \Phi(f(s_j))\|$
  end for
  Choose two samples $(i^*, j^*) = \text{argmin}_{(i,j)} d_{i,j}$.
  if $\min\{d_{z,i^*}\}_{z=1,z\neq j^*}^{|\tilde{S}_{n,k}|} < \min\{d_{z,j^*}\}_{z=1,z\neq i^*}^{|\tilde{S}_{n,k}|}$ **then**
    $Z_{n,k} \leftarrow Z_{n,k} \cup \{s_{i^*}\}$
  else
    $Z_{n,k} \leftarrow Z_{n,k} \cup \{s_{j^*}\}$
  end if
  $E \leftarrow E + 1$
end while

A.12 Other Implementation Details

Our work is built upon the official setup of EFDMix (Zhang et al., 2022). Different from the original setting of EFDMix, for image classification tasks, we trained the model for 150 epochs with a mini-batch size of 128. We also randomly sampled the data from all source domains in each mini-batch. Other setups are exactly the same as in MixStyle (Zhou et al., 2021), DSU (Li et al., 2022) and EFDMix (Zhang et al., 2022) when implementing each module; each module is activated with probability 0.5. Following the original setups, Mixstyle and EFDM are inserted after the 1st, 2nd and 3rd residual blocks for PACS. For other datasets, Mixstyle and EFDM are inserted after the 1st and 2nd residual blocks. DSU is inserted after 1st convolutional layer, max pooling, 1,2,3,4-th residual blocks. Here, our SB module is operated at the moment where MixStyle, DSU, EFDMix are first activated. The TS module is operated at first residual blocks during testing for VLCS, Office-Home and person re-ID task. We set $\alpha = 3$ for all experiments on image classification tasks, while $\alpha = 5$ is utilized for person re-ID task.

A.13 Complexities

Style balancing. Recall that B, N, K are the batch size, number of source domains, total number of classes, respectively. To compute the time complexity, suppose that there are $\frac{B}{NK}$ samples in a mini-batch corresponding to each domain n with class label k. Then, the addition complexity required for our style balancing in Step 2 becomes $\mathcal{O}((\frac{B}{NK})^2 \times N \times K) = \mathcal{O}(\frac{B^2}{NK})$, which is the additional cost for achieving an improved domain diversity.

Test-time style shifting. Once the style statistics of train samples are obtained, only the style gaps between the test sample and the center of N source domains are required for test-time style shifting; this makes the additional complexity negligible.

Algorithm 2 Test-Time Style Shifting (TS)

Input: Test sample t and the corresponding feature $f(t)$ at a specific layer (where TS module is operated), Φ_S and Φ_{S_n} for all source domains $n \in \{1, 2, \ldots, N\}$, and α.
Output: New feature TS$(f(t))$.

for each test sample t **do**
  Compute $\frac{1}{N}\sum_{n=1}^{N} \|\Phi(f(t)) - \Phi_{S_n}\|$
  if $\frac{1}{N}\sum_{n=1}^{N} \|\Phi(f(t)) - \Phi_{S_n}\| > \alpha\Big(\frac{1}{N}\sum_{n=1}^{N} \|\Phi_S - \Phi_{S_n}\|\Big)$ **then**
    $\Phi(f(t))_{\text{new}} = \Phi_{S_{n'}}$, where $n' = \text{argmin}_n \|\Phi(f(t)) - \Phi_{S_n}\|$ // style shift to the nearest source domain
  else
    $\Phi(f(t))_{\text{new}} = \Phi(f(t))$ // keep the original style
  end if
  From $\Phi(f(t))_{\text{new}} = [\mu(f(t))_{\text{new}}, \sigma(f(t))_{\text{new}}]$,
  compute $\text{TS}(f(t)) = \sigma(f(t))_{\text{new}} \frac{f(t) - \mu(f(t))}{\sigma(f(t))} + \mu(f(t))_{\text{new}}$
end for