MultiMatch: Multi-task Learning for Semi-supervised Domain Generalization

LEI QI, School of Computer Science and Engineering (Southeast University), and Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, Nanjing, China
HONGPENG YANG, School of Cyber Science and Engineering, Southeast University, Nanjing, China
YINGHUAN SHI, The State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
XIN GENG, School of Computer Science and Engineering (Southeast University), and Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, Nanjing, China

Domain generalization (DG) aims at learning a model on source domains to well generalize on the unseen target domain. Although it has achieved great success, most of the existing methods require the label information for all training samples in source domains, which is time-consuming and expensive in the real-world application. In this article, we resort to solving the semi-supervised domain generalization (SSDG) task, where there are a few label information in each source domain. To address the task, we first analyze the theory of multi-domain learning, which highlights that (1) mitigating the impact of domain gap and (2) exploiting all samples to train the model can effectively reduce the generalization error in each source domain so as to improve the quality of pseudo-labels. According to the analysis, we propose MultiMatch, i.e., extending FixMatch to the multi-task learning framework, producing the high-quality pseudo-label for SSDG. To be specific, we consider each training domain as a single task (i.e., local task) and combine all training domains together (i.e., global task) to train an extra task for the unseen test domain. In the multi-task framework, we utilize the independent batch normalization and classifier for each task, which can effectively alleviate the interference from different domains during pseudo-labeling. Also, most of the parameters in the framework are shared, which can be trained by all training samples sufficiently. Moreover, to further boost the pseudo-label accuracy and the model’s generalization, we fuse the predictions from the global task and local task during training and testing, respectively. A series of experiments validate the effectiveness of the proposed method, and it outperforms the existing semi-supervised methods and the SSDG method on several benchmark DG datasets.

CCS Concepts: • Computing methodologies → Semi-supervised learning settings;

Additional Key Words and Phrases: Multi-task learning, semi-supervised learning, domain generalization
1 INTRODUCTION

Deep learning has achieved a remarkable success in many application tasks [8, 15, 30, 40], such as computer vision and natural language processing. However, when there is the domain shift between the training set and the test set, the typical deep model cannot effectively work on the test dataset [47, 63]. Hence, it requires training a new model for a new scenario, which is infeasible in real-world applications. Recently, a new task named domain generalization (DG) is proposed [28, 31], where there are several available source domains during training, and the test set is unknown, thus the training and test samples are various from the data-distribution perspective. The goal of DG is to train a generalizable model for the unseen target domain using several source domains.

Several domain generalization methods have been developed to handle this issue [32, 37, 41, 51, 56, 58]. However, these methods need to label all data from source domains, which is expensive and time-consuming in real-world applications. In general, a few labels for a dataset are easy to obtain, thus the semi-supervised domain generalization (SSDG) task is recently proposed [64], where there are a few labeled data in each source domain while most of the samples do not have the label information. To tackle this task, StyleMatch [64] is proposed via extending the FixMatch [43] with a couple of new ingredients, which resorts to enhancing the diversity from the image level and the classifier level. However, this method does not handle the data-distribution discrepancy of different source domains, resulting in a negative impact on the accuracy of pseudo-labels during training.

Similarly, in the conventional semi-supervised learning (SSL), most existing methods assume that all training samples are sampled from the same data distribution [13, 21, 33, 44, 48, 57, 62]. Differently, in the semi-supervised domain generalization task, the training samples from different distributions, as illustrated in Figure 1. Therefore, if directly using these methods to deal with the SSDG task, then we cannot obtain the accurate pseudo-labels during the training course, because the domain shift in the training set is not beneficial to producing the accurate pseudo-label.

In this article, we mainly resort to obtaining accurate pseudo-labels so as to enhance the model’s discrimination and generalization in the unseen domain. We first analyze the generalization error on a domain using the theory of multi-domain learning [2]. Based on the upper bound of the generalization error, we can obtain a conclusion that (1) alleviating the interference of different domains and (2) using all samples to train the model can effectively reduce the upper bound of generalization error on each domain, which means that the accurate pseudo-labels can be generated.

Inspired by the theory of multi-domain learning, we extend the FixMatch [43] to a multi-task learning method, named MultiMatch, for semi-supervised domain generalization. Specifically, we build the independent local task for each domain to mitigate the interference from different domains during generating pseudo-labels. In addition, we also construct a jointly global task for predicting unseen target domains. Particularly, each independent local task is utilized to give the pseudo-label, while the jointly global task is trained using the pseudo-label. Furthermore, benefiting from the multi-task framework, we can fuse the predictions from the global and local tasks to further improve the accuracy of pseudo-labels and the generalization capability of the model in the

FixMatch is an excellent baseline in SSDG, which will be validated in the experimental section.
training and test stages, respectively. We conduct a series of experiments on several DG benchmark datasets. Experimental results demonstrate the effectiveness of the proposed method, and the proposed method can produce a performance improvement when compared with the semi-supervised learning methods and the semi-supervised DG method. Moreover, we also verify the efficacy of each module in our method. In conclusion, our main contributions can be summarized as:

- We analyze the semi-supervised domain generalization task based on the generalization error of multi-domain learning, which inspires us to design a method to obtain the high-quality pseudo-label during training.
- We propose a simple yet effective multi-task learning (i.e., MultiMatch) for semi-supervised domain generalization, which can effectively reduce the interference from different domains during pseudo-labeling. Also, most of the modules in the model are shared for all domains, which can be sufficiently trained by all samples. To further promote the accuracy of pseudo-labels and the capability of the model, we propose to merge the outputs of the local task and the global task together to yield a robust prediction in the training and test stages.
- We evaluate our approach on multiple standard benchmark datasets, and the results show that our approach outperforms the state-of-the-art accuracy. Moreover, the ablation study and further analysis are provided to validate the efficacy of our method.

The rest of this article is organized as follows. We review some related work in Section 2. The proposed method is introduced in Section 3. Experimental results and analysis are presented in Sections 4 and 5 is the conclusion.

2 RELATED WORK

In this section, we investigate the related work to our work, including the semi-supervised learning methods and the domain generalization methods. The detailed investigation is presented in the following part.

2.1 Semi-Supervised Learning

The semi-supervised learning has achieved remarkable performance in the recent years [3, 4, 10, 11, 13, 21, 26, 33, 43, 44, 48, 49, 55, 57, 59, 62]. For example, Grandvalet et al. [13] propose to minimize entropy regularization, which enables us to incorporate unlabeled data in the standard supervised learning. Laine et al. [21] developed Temporal Ensembling, which maintains an exponential moving average of label predictions on each training example and penalizes predictions that are inconsistent with this target. However, because the targets change only once per epoch, Temporal Ensembling becomes unwieldy when learning large datasets. To overcome this issue,
Tarvainen et al. [44] designed Mean Teacher, which averages model weights instead of label predictions. In FixMatch [43], the method first generates pseudo-labels using the model’s predictions on weakly augmented unlabeled images and is then trained to predict the pseudo-label when fed a strongly augmented version of the same image. Furthermore, Zhang et al. [57] introduce a curriculum learning approach to leverage unlabeled data according to the model’s learning status. The core of the method is to flexibly adjust thresholds for different classes at each timestep to let pass informative unlabeled data and their pseudo-labels.

However, the success of the typical SSL largely depends on the assumption that the labeled and unlabeled data share an identical class distribution, which is hard to meet in the real-world application. The distribution mismatch between the labeled and unlabeled sets can cause severe bias in the pseudo-labels of SSL, resulting in significant performance degradation. To bridge this gap, Zhao et al. [61] put forward a new SSL learning framework, named Distribution Consistency SSL, which rectifies the pseudo-labels from a distribution perspective. Differently, Oh et al. [34] propose a general pseudo-labeling framework that class-adaptively blends the semantic pseudo-label from a similarity-based classifier to the linear one from the linear classifier.

Different from these existing semi-supervised learning settings, we aim to address the semi-supervised task with multiple domains in the training procedure, which results in the interference of different domains during pseudo-labeling because of the impact of domain shift.

2.2 Domain Generalization

Recently, some methods are also developed to address the domain generalization problem in some computer vision tasks [1, 5, 24, 27, 29, 32, 36, 41, 56, 58, 67], such as classification and semantic segmentation. Inspired by domain adaptation methods [52, 54], some works based on domain alignment [12, 25, 27, 31, 38, 39, 60] aim at mapping all samples from different domains into the same subspace to alleviate the difference of data distribution across different domains. For example, Muandet et al. [31] introduce a kernel-based optimization algorithm to learn the domain-invariant feature representation and enhance the discrimination capability of the feature representation. However, this method cannot guarantee the consistency of the conditional distribution; hence Zhao et al. [60] develop an entropy regularization term to measure the dependency between the learned feature and the class labels, which can effectively ensure the conditional invariance of learned feature, so that the classifier can also correctly classify the feature from different domains.

Besides, Gong et al. [12] exploit CycleGAN [68] to yield new styles of images that cannot be seen in the training set, which smoothly bridge the gap between source and target domains to boost the generalization of the model. Rahman et al. [38] also employ GAN to generate synthetic data and then mitigate domain discrepancy to achieve domain generalization. Differently, Li et al. [25] adopt an adversarial auto-encoder learning framework to learn a generalized latent feature representation in the hidden layer, and use Maximum Mean Discrepancy to align source domains, then they match the aligned distribution to an arbitrary prior distribution via adversarial feature learning. In this way, it can better generalize the feature of the hidden layer to other unknown domains. Rahman et al. [39] incorporate the correlation alignment module along with adversarial learning to help achieve a more domain-agnostic model because of the improved ability to more effectively reduce domain discrepancy. In addition to performing adversarial learning at the domain level to achieve domain alignment, Li et al. [27] also conduct domain adversarial tasks at the class level to align samples of each category from different domains.

Ideally, visual learning methods should be generalizable, for dealing with any unseen domain shift when deployed in a new target scenario, and they should be data efficient for reducing development costs by using as few labels as possible. However, the conventional DG methods, which are unable to handle unlabeled data, perform poorly with limited labels in the SSDG task.
Fig. 2. An illustration of the proposed MultiMatch for the semi-supervised domain generalization setting. In the method, training each domain is considered as the local task, and training all domains together is viewed as the global task. In the training course, we propose to employ the prediction fusion scheme to generate the final pseudo-label, which can be used to train the unlabeled samples. For example, we utilize the images based on weak augmentation to generate the pseudo-label. For an image from the $i$th domain, we can obtain two predictions from the \texttt{T}i-\texttt{BN} and \texttt{T}G-\texttt{BN} paths (i.e., the local and global tasks), and then we can fuse the two predictions to generate the final pseudo-label of the image by Function 1 (i.e., choose the highest confidence as pseudo-labels). For the image of the strong augmentation, we can obtain two predictions from the \texttt{T}i-\texttt{BN} and \texttt{T}G-\texttt{BN} paths and use the pseudo-label of the image to compute the loss and update the network.

To handle this task, Zhou et al. [64] propose StyleMatch, a simple approach that extends FixMatch with a couple of new ingredients tailored for SSDG: (1) stochastic modeling for reducing over-fitting in scarce labels and (2) multi-view consistency learning for enhancing the generalization capability of the model. However, StyleMatch cannot effectively address the interference of different domains during pseudo-labeling. In this article, we deal with the SSDG task from the multi-task learning perspective.

3 THE PROPOSED METHOD

In this article, our goal is to address the SSDG task. In this task, a few labeled samples on each domain are available, while most of the samples lack the label information, thus a key task is how to generate the accurate pseudo-label. Different from the conventional semi-supervised methods, the SSDG task contains multiple source domains with different distributions. We first analyze the generalization error of multi-domain learning, which shows the upper bound of the generalization error is related to the discrepancy of different domains and the number of training samples. Particularly, reducing the upper bound is equal to improving the accuracy of the pseudo-label. Based on the analysis, we develop a multi-task learning method (i.e., MultiMatch) to deal with the SSDG task, as illustrated in Figure 2. Besides, to further leverage the advantage of the multi-task learning framework, we fuse the predictions from different tasks to yield a better pseudo-label and more generalizable model in the training and test procedure. We will introduce our method in the following part.

3.1 Theoretical Insight

In the semi-supervised DG task, each domain contains labeled data and unlabeled data. To be specific, given $N$ domains in the training stage, we use $\mathcal{D}_i = \{\mathcal{D}_{li}, \mathcal{D}_{ui}\}$ to denote the $i$th domain,
where $D_i^l$ and $D_i^u$ represent the labeled and unlabeled samples in the $i$th domain, respectively. Since there is no label information for $D_i^u$, we aim to generate the high-quality pseudo-label in the training stage. Particularly, the SSDG task in the training stage can also be considered as multi-domain learning, and these unlabeled data can be viewed as the test data during pseudo-labeling. In the next part, we will introduce a theory of multi-domain learning to explore the semi-supervised DG task from the theoretical perspective.

Here, we consider hypotheses $h \in \mathcal{H}$ (i.e., prediction function), and give a vector $\alpha = (\alpha_1, \ldots, \alpha_N)$ of domain weights with $\sum_{j=1}^N \alpha_j = 1$. In addition, we assume that the learner receives a total of $m$ labeled training samples, with $m_j = \beta_j m$ from the $j$th domain $D_j$. We define the empirical $\alpha$-weighted error of function $h$ as

$$\hat{\epsilon}_\alpha(h) = \sum_{j=1}^N \alpha_j \hat{\epsilon}_j(h) = \sum_{j=1}^N \frac{\alpha_j}{m_j} \sum_{x \in S_j} |h(x) - f_j(x)|,$$

(1)

where $f_j(x)$ is a labeling function for the $j$th domain (i.e., the mapping from a sample to its ground truth).

**Theorem 1** ([2]). Let $\mathcal{H}$ be a hypothesis space of VC dimension $d$. For each domain $j \in \{1, \ldots, N\}$, let $S_j$ be a labeled sample of size $\beta jm$ generated by drawing $\beta_j m$ points from $D_j$ and labeling them according to $f_j(\cdot)$. If $\hat{h} \in \mathcal{H}$ is the empirical minimizer of $\hat{\epsilon}_\alpha(h)$ for a fixed weight vector $\alpha$ on these samples and $\epsilon^*_T = \min_{h \in \mathcal{H}} \epsilon_T(h)$ is the target error minimizer, then the upper bound of the generalization error on the target domain $\mathcal{D}_T$, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, can be written as

$$\epsilon_T(\hat{h}) \leq \epsilon_T(\epsilon^*_T) + 2 \sqrt{\sum_{j=1}^N \frac{\alpha_j^2}{\beta_j} \frac{d \log(2m) + \log(\delta)}{2m}} + \sum_{j=1}^N \alpha_j (2\lambda_j + d_{\mathcal{H}^\Delta \mathcal{H}}(\mathcal{D}_j, \mathcal{D}_T)),$$

(2)

where

$$\lambda_j = \min_{h \in \mathcal{H}} \{\epsilon_T(h) + \epsilon_j(h)\}.$$

The detailed demonstration process can be found in Reference [2].

In Equation (2), the third term indicates the distribution discrepancy between $\mathcal{D}_j$ and $\mathcal{D}_T$. It is worth noting that, in the semi-supervised DG setting, all $\{\alpha_i\}_{i=1}^N$ and $\{\beta_i\}_{i=1}^N$ are $\frac{1}{N}$. Therefore, the Equation (2) can be rewritten as

$$\epsilon_T(\hat{h}) \leq \epsilon_T(\epsilon^*_T) + \sum_{j=1}^N \frac{2\lambda_j}{N} + 2 \sqrt{\frac{d \log(2m) + \log(\delta)}{2m}} + \frac{1}{N} d_{\mathcal{H}^\Delta \mathcal{H}}(\mathcal{D}_j, \mathcal{D}_T).$$

(4)

According to Equation (4), the upper bound of the generalization error on the target domain is mainly decided by four terms, where the first two terms can be considered as the constant. There are two observations in the third term and the fourth term. (1) In the third term, when $m$ is a larger value, the total value is smaller, which indicates that we should use the available samples to train the model. In other words, most modules in the designed model should be shared for all domains.
or tasks. (2) The last term is the distribution discrepancy between the source-target domain and the other domains. If we can alleviate the discrepancy, then the generalization error on the source-target domain can also be reduced.

Particularly, since the semi-supervised DG task consists of multiple domains, it can be also viewed as the multi-domain learning paradigm in the training stage. Besides, the unlabeled training data can be also considered as the test data during pseudo-labeling. Therefore, it means that if we can effectively reduce the upper bound of the generalization error in Equation (4), then it will result in generating a high-quality pseudo-label for each domain in the training stage.

3.2 Multi-task Learning Framework

FixMatch is a significant simplification of existing SSL methods [43]. It first generates pseudo-labels using the model’s predictions on weakly augmented unlabeled images. For a given image, the pseudo-label is only retained if the model produces a high-confidence prediction. The model is then trained to predict the pseudo-label when fed a strongly augmented version of the same image. Despite its simplicity, FixMatch achieves state-of-the-art performance across a variety of standard semi-supervised learning benchmarks. In the SSDG task, FixMatch also achieves better performance than other conventional semi-supervised methods [48, 57], which is verified in our experiment. In this article, we aim to extend FixMatch to a multi-task framework.

Based on the upper bound of the generalization error in Equation (4), the proposed method needs to satisfy two requirements: (1) most of modules in the model are shared for all domains, which can be sufficiently trained by all samples, and (2) the model can reduce the interference of domain gap between different domains. Therefore, we propose a multi-task learning framework, named MultiMatch, to address the SSDG task.

In this part, we will describe our multi-task learning framework. Since there are multiple different domains in the SSDG task, we consider training each domain as an independent task (i.e., the local task for each domain), which can effectively reduce the interference between different domains during pseudo-labeling. Besides, considering the SSDG task also needs to exploit the trained model on the unseen domain, we add a global task where we employ pseudo-labels from the local task to train the model. We assume that there are $N$ source domains in the training stage, thus we will build $N + 1$ tasks in our framework. To be specific, we will employ the independent batch normalization (BN) [6] and classifier for each task in our method, and other modules are shared for all tasks. The BN can be formulated by

$$BN(f_d) = \gamma \frac{f_d - M_d(\phi)}{M_{\sigma}(\phi)} + \beta, \ d \in \phi,$$

where $\phi$ is the domain set and $M_d(\phi)$ and $M_{\sigma}(\phi)$ represent the statistics for the $\phi$. When $\phi$ merely includes a domain, the sample will be normalized by the statistics from the own domain. Differently, when $\phi$ includes all domains, each sample will be normalized by the shared statistics from all domains. Since there exists a domain gap across different domains, the shared statistics will bring the noise error, as shown in the last term of Equation (4), resulting in larger generalization error.

Remark. In our multi-task learning framework, using the independent BN can effectively mitigate the interference of different domains, as shown in Equation (4). In addition, in our method, most of the modules are shared for all domains, which can sufficiently exploit all samples to reduce the third item in Equation (4). Hence, our multi-task learning framework can obtain a small generalization error on each domain so as to generate an accurate pseudo-label for each domain in the training stage. Last, the multi-task framework with multiple classifiers can produce the ensemble

---

2When the model generates the pseudo-label for a source domain, this source domain is named the source-target domain.
prediction for pseudo-label and test evaluation in the training and test procedure. We will give the prediction fusion scheme in the next part.

3.3 Prediction Fusion

For an image $x_i$ from the $i$th domain, we can generate its output $Y_i = [y_1; \cdots; y_{N+1}] \in \mathbb{R}^{C \times (N+1)}$, where $C$ is the number of the classes. In the training stage, due to the interference of different domains with each other, we employ the output of the $i$th task as the main prediction. To further guarantee the reliability of pseudo-labels in the training course, we combine it with the output of the global task to generate the final pseudo-labels. The motivation of fusing the global prediction is that the local and global predictions can be viewed as the private and shared information for all images, and thus they are complementary. In this article, we fuse them together with the Function 1 (i.e., choose the highest confidence as pseudo-labels) to get the accurate pseudo-label. For an image $x_i$ from the $i$th domain, the pseudo-label is produced by

$$\hat{y} = \text{PredictLabel}(y_i; y_{N+1}),$$

where PredictLabel$(\cdot)$ is defined in Function 1. In the test stage, we use the output of the global task as the main prediction, because the test domain is unseen (i.e., we do not know which training domain is similar to the test domain). Besides, for a test image, we also fuse the output of the most similar task to yield the final prediction result as

$$y_{max} = \text{SelectTask}(y_1; \cdots; y_N);$$

$$\tilde{y} = \text{PredictLabel} \left( \frac{y_{max} + y_{N+1}}{2} \right),$$

where SelectTask$(\cdot)$ is defined in Function 2 and returns the prediction of the most similar task (i.e., the column has the maximum value in a matrix) for a test image in the test procedure.

**Function 1: PredictLabel**

1. **Input:** A matrix $Y \in \mathbb{R}^{c \times n}$.
2. **Output:** A vector $y \in \mathbb{R}^{c \times 1}$.
3. Find the maximum value of each line in $Y$ to generate a vector $y \in \mathbb{R}^{c \times 1}$.
4. Find the position (e.g., $p$) of the maximum value of the vector $y$.
5. Setting the position ($p$) of $y$ as 1 (i.e., $y(p)=1$), and other positions as 0.
6. Return $y$.

**Function 2: SelectTask**

1. **Input:** A matrix $Y \in \mathbb{R}^{c \times n}$.
2. **Output:** A vector $y \in \mathbb{R}^{c \times 1}$.
3. Find the column’s position (e.g., $p$) of the maximum value of the vector $Y$.
4. Select the $p$th column from Y as $y$ (i.e., $y = Y(:,p)$).
5. Return $y$.

3.4 Training Process

In our method, we merely use the cross-entropy loss to train our model as in FixMatch [43]. In the training course, we randomly select the same number of labeled and unlabeled images from each domain to form a batch. It is worth noting that each image passes the domain-specific BN and classifier, and all images are required to pass the global task. Simultaneously, we fuse the
Algorithm 1: The training procedure of the proposed MultiMatch

1: **Input:** All training domains \( \{D^l_i, D^u_i \}_{i=1}^N \).
2: **Output:** The trained parameters of the model \( \theta \).
3: **Initialization:** Initialize the parameters \( \theta \).
4: **for** \( \text{iter} \leq \text{MaxIter} \) **do**
5: Generating a batch of images \( \{B^l, B^u\} \).
6: Using the labeled samples \( B^l \) to train the model, each image passes the domain-specific BN and classifier, and all images pass the global BN and classifier.
7: Conducting weak and strong augmentations for \( B^u \) to generate \( B^u_w \) and \( B^u_s \).
8: Based on \( B^u_w \) to yield the pseudo-label \( Y_u \) by Equation (6).
9: Using \( B^u_s \) and \( Y_u \) to train the model, and the forward path of all images is consistent with \( B^l \).
10: **end for**

predictions of the domain-specific task and the global task together to generate accurate pseudo-labels. The overall training process is shown in Algorithm 1.

4 EXPERIMENTS

In this section, we first introduce the experimental datasets and settings in Section 4.1. Then, we compare the proposed MultiMatch with the state-of-the-art SSL methods and the SSDG method in Section 4.2, respectively. To validate the effectiveness of various components in our MultiMatch, we conduct ablation studies in Section 4.3. Last, we further analyze the property of the proposed method in Section 4.4.

4.1 Datasets and Experimental Settings

4.1.1 Datasets. In this article, we conduct the experiments to validate the effectiveness of our method on three benchmark DG datasets as follows:

- **PACS** [23] consists of four different domains: Photo, Art, Cartoon, and Sketch. It contains 9,991 images with seven object categories in total, including Photo (1,670 images), Art (2,048 images), Cartoon (2,344 images), and Sketch (3,929 images).
- **Office-Home** [46] contains 15,588 images of 65 categories of office and home objects. It has four different domains, namely Art (2,427 images), Clipart (4,365 images), Product (4,439 images), and Real World (4,357 images), which is originally introduced for UDA but is also applicable in the DG setting.
- **miniDomainNet** [66] takes a subset of DomainNet [35] and utilizes a smaller image size (96 × 96). miniDomainNet includes four domains and 126 classes. As a result, miniDomainNet contains 18,703 images of Clipart, 31,202 images of Painting, 65,609 images of Real, and 24,492 images of Sketch.

We show some examples from PACS and Office-home in Figure 3. As seen, there is an obvious difference among different domains. Besides, we also visualize the features of four categories on PACS by t-SNE [45], as illustrated in Figure 4. In this figure, different colors are different domains. We observe that different domains appear in different spaces, which validates that there exists a domain shift in the training set.

4.1.2 Implementation Details. Following the common practice [17], ResNet18 [15] pre-trained on ImageNet [7] is employed as the CNN backbone (for all models compared in this article). We randomly sample 64 images from each source domain to construct a minibatch for labeled and
unlabeled data, respectively. The initial learning rate is set as 0.003 for the pre-trained backbone. In the experiment, to enrich the diversity of the augmentation, we integrate the AdaIN augmentation [16] into the strong augmentation scheme. Particularly, the augmentation scheme is utilized in all experiments, including the baseline (i.e., FixMatch), and thus it is a fair comparison in the experiment.

4.2 Comparison with State-of-the-art Methods

We conduct the experiment to compare our method with some semi-supervised learning methods (i.e., MeanTeacher [44], EntMin [13], DebiasPL [48], FlexMatch [57], and FixMatch [43]) and the semi-supervised DG method (i.e., StyleMatch [64]). In this experiment, we run these methods under two different settings (i.e., 10 labels per class and 5 labels per class) on three benchmark datasets. The experimental results are reported in Tables 1 and 2. As seen in this table, among these typical semi-supervised methods, FixMatch can achieve the best performance. Compared with FixMatch, our method can further improve performance. For example, on PACS our method outperforms FixMatch by +3.74% (81.57 vs. 77.83) and +5.09% (80.04 vs. 74.95) under the “10 labels per class” case and the “5 labels per class” case, respectively. Besides, on the large-scale dataset (i.e., miniDomainNet), our method can also achieve an obvious improvement, which is attributed to the fact that our method reduces the interference of different domains while guaranteeing that all training samples are utilized to train the model.

In addition, StyleMatch is developed to address the semi-supervised domain generalization task. Compared with it, our method has better experimental results on all settings and datasets, except for the “5 labels per class” case on PACS. For example, on miniDomainNet, our method increases StyleMatch by +3.66% (58.79 vs. 55.13) and +3.57% (54.61 vs. 51.04) under the “10 labels per class” case and the “5 labels per class” case, respectively. This confirms the effectiveness of our method when compared with the SOTA method, thanking the advantage of the multi-task learning framework in the semi-supervised domain generalization task. The effectiveness of each module in our proposed MultiMatch will be verified in the next ablation study.
4.3 Ablation Studies

In the experiment, we first validate the effectiveness of the multi-task learning framework, and then analyze the efficacy of the fusion prediction scheme in the training stage and the test stage, respectively. The experimental results are shown in Table 3, where “MTL+TRAIN-local+TEST-global” denotes that the pseudo-labels are generated by the domain-specific path (i.e., local task), and the final prediction during test is based on the global task. In other words, “MTL+TRAIN-local+TEST-global” means that we do not utilize the fusion prediction scheme in both the training and test stages. “MTL+TRAIN-global-local+TEST-global-local” indicates that we employ the fusion prediction scheme in both the training and test stages.

As seen in Table 3, “MTL+TRAIN-local+TEST-global” outperforms “Baseline” on both PACS and Office-Home, which confirms the effectiveness of the multi-task learning framework in the semi-supervised domain generalization task. For example, the multi-task learning framework can bring an obvious improvement by +3.37% (61.23 vs. 57.86) on Office-Home. In addition, the
Table 1. Comparison between Our Method and Different Semi-supervised (DG) Methods under Different Numbers of Labeled Samples on PACA and Office-Home

| Method            | PACS               | Office-Home         |
|-------------------|--------------------|---------------------|
|                   | P  | A  | C  | S  | Avg | A  | C  | P  | R  | Avg |
| MeanTeacher [44]  | 85.95 | 62.41 | 57.94 | 47.66 | 63.49 | 49.92 | 43.42 | 64.61 | 68.79 | 56.69 |
| EntMin [13]       | 89.39 | 72.77 | 70.55 | 54.38 | 71.77 | 51.92 | 44.92 | 66.85 | 70.52 | 58.55 |
| FlexMatch [57]    | 66.04 | 48.44 | 60.79 | 53.23 | 57.13 | 25.63 | 28.25 | 37.98 | 36.15 | 32.00 |
| DebiasPL [48]     | 93.23 | 74.60 | 67.23 | 63.49 | 74.64 | 44.71 | 38.62 | 64.42 | 68.39 | 54.04 |
| FixMatch [43]     | 87.40 | 76.85 | 69.40 | 77.68 | 77.83 | 49.90 | 50.98 | 63.79 | 66.75 | 57.86 |
| StyleMatch [64]   | 90.04 | 79.43 | 73.75 | 78.40 | 80.41 | 52.82 | 51.60 | 65.31 | 68.61 | 59.59 |
| MultiMatch (ours) | **93.25** | **83.30** | 73.00 | 76.74 | **81.57** | **56.32** | **52.76** | **68.94** | **72.28** | **62.57** |

5 labels per class

| Method            | PACS               | Office-Home         |
|-------------------|--------------------|---------------------|
|                   |                  |                    |
| MeanTeacher [44]  | 73.54 | 56.00 | 52.64 | 36.97 | 54.79 | 44.65 | 39.15 | 59.18 | 62.98 | 51.49 |
| EntMin [13]       | 79.99 | 67.01 | 65.67 | 47.96 | 65.16 | 48.11 | 41.72 | 62.41 | 63.97 | 54.05 |
| FlexMatch [57]    | 61.80 | 23.83 | 37.28 | 48.09 | 42.75 | 19.04 | 26.16 | 30.81 | 23.00 | 24.75 |
| DebiasPL [48]     | 89.52 | 65.52 | 54.09 | 29.22 | 59.59 | 41.41 | 33.45 | 58.77 | 60.59 | 48.56 |
| FixMatch [43]     | 86.15 | 74.34 | 69.08 | 73.11 | 75.67 | 46.60 | 48.74 | 59.65 | 63.21 | 54.55 |
| StyleMatch [64]   | 89.25 | 78.54 | 74.44 | 79.06 | **80.32** | 51.53 | 50.00 | 60.88 | 64.47 | 56.72 |
| MultiMatch (ours) | **90.66** | **82.43** | 72.43 | 74.64 | 80.04 | **53.71** | **50.21** | **63.99** | **67.81** | **58.93** |

Note that "P," "A," "C," and "S" denote different domains on PACS. "Avg" is the average result of all domains. The bold is the best result.

Table 2. Comparison between Our Method and Different Semi-supervised (DG) Methods under Different Numbers of Labeled Samples on miniDomainNet

| Method            | miniDomainNet      |
|-------------------|-------------------|
|                   | C        | P        | R        | S        | Avg      |
|                   | 10 labels per class |
| MeanTeacher [44]  | 46.75    | 49.48    | 60.98    | 37.05    | 48.56    |
| EntMin [13]       | 46.96    | 49.59    | 61.10    | 37.26    | 48.73    |
| FlexMatch [57]    | 38.97    | 32.44    | 39.34    | 11.82    | 30.64    |
| DebiasPL [48]     | 54.18    | 54.27    | 57.53    | 47.93    | 53.48    |
| FixMatch [43]     | 51.22    | 54.99    | 59.53    | 52.45    | 54.55    |
| StyleMatch [64]   | 52.76    | 56.15    | 58.72    | 52.89    | 55.13    |
| MultiMatch (ours) | **56.13** | **59.15** | **63.68** | **56.18** | **58.79** |

5 labels per class

| Method            | miniDomainNet      |
|-------------------|-------------------|
|                   | C        | P        | R        | S        | Avg      |
| MeanTeacher [44]  | 39.01    | 42.92    | 54.40    | 31.13    | 41.87    |
| EntMin [13]       | 39.39    | 43.35    | 54.80    | 31.72    | 42.32    |
| FlexMatch [57]    | 13.80    | 20.52    | 17.88    | 20.40    | 18.15    |
| DebiasPL [48]     | 50.87    | 52.11    | 51.86    | 44.61    | 49.86    |
| FixMatch [43]     | 47.23    | 52.88    | 55.99    | **52.88** | 52.24    |
| StyleMatch [64]   | 46.30    | 52.63    | 54.65    | 50.59    | 51.04    |
| MultiMatch (ours) | **50.99** | **56.03** | **58.53** | **52.88** | **54.61** |

Bold is the best result.
fusion prediction scheme is also effective in both the training and test stages. As seen in Table 3, “MTL+TRAIN-global-local+TEST-global” outperforms “MTL+TRAIN-local+TEST-global,” and “MTL+TRAIN-global-local+TEST-global-local” outperforms “MTL+TRAIN-local+TEST-global-local,” which indicates the effectiveness of the fusion prediction scheme in the training stage. Meanwhile, “MTL+TRAIN-local+TEST-global-local” outperforms “MTL+TRAIN-global-local+TEST-global,” and “MTL+TRAIN-global-local+TEST-global-local” outperforms “MTL+TRAIN-global-local+TEST-global,” which confirms the effectiveness of the fusion prediction scheme in the test stage. In our method, we use the fusion manner in Section 3.3. We will also investigate some other fusion manners in further analysis.

### 4.4 Further Analysis

**Evaluation on different fusion manners.** In this part, we evaluate different fusion manners in our framework on PACS. Experimental results are listed in Table 4. Each fusion manner is described in the following formulas. “TEST-avg-all” is the mean of the outputs from all tasks in the testing stage. “TEST-max” denotes the maximum output among all tasks in the testing stage. “TRAIN-avg” means using the mean of outputs from the own task and the global task during pseudo-labeling. As observed in Table 4, “TRAIN-max+TEST-avg” (i.e., Equations (6) and (8)) can obtain a slight improvement when compared with other fusion schemes. Besides, compared with
the “MTL+TRAIN-local+TEST-global” (i.e., without using the fusion scheme in our method), all schemes in Table 4 outperform it, which indicates that the prediction’s ensemble is significant for our method during training and testing.

**TEST-avg-all:**

\[
\text{Class} = \text{MaxClass}\left(\frac{y_1 + \cdots + y_{N+1}}{N+1}\right),
\]

**TEST-max:**

\[
\text{Class} = \text{MaxClass}\left([y_{\text{max}}; y_{N+1}]\right),
\]

where \( y_{\text{max}} \) is defined in Equation (7).

**TRAIN-avg:**

\[
\text{Class} = \text{MaxClass}\left(\frac{y_i + y_{N+1}}{2}\right),
\]

where \( y_i \) is the prediction of an image from the \( i \)th domain.

The accuracy of pseudo-labels. In this experiment, we use **Precision**, **Recall**, and **macro-f1** to evaluate the accuracy of pseudo-labels. Here we give the detailed definition as follows:

\[
\text{Precision} = \frac{N_c}{N} \times 100, \quad (9)
\]

where \( N \) and \( N_c \) denote the number of the total samples and the correctly predicted samples. In addition, we define \( TP \) as true positive, \( TN \) as true negative, \( FP \) as false positive, as \( FN \) as false negative. To be specific, for the \( i \)th class, \( TP_i \) is the number of the true \( i \)th class among the predicted \( i \)th class; \( TN_i \) is the number of the true non \( i \)th class among the predicted non \( i \)th class; \( FP_i \) is the number of the false \( i \)th class among the predicted \( i \)th class; \( FN_i \) is the number of the false non \( i \)th class among the predicted non \( i \)th class.

For the \( i \)th class, we have

\[
\text{Recall}_i = \frac{TP_i}{TP_i + FN_i}, \quad \text{Precision}_i = \frac{TP_i}{TP_i + FP_i}, \quad F_i = 2\frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}. \quad (10)
\]

Hence, we can obtain

\[
\text{Recall} = \frac{100}{C} \sum_{c=1}^{C} \text{Recall}_i, \quad \text{macro-f1} = \frac{100}{C} \sum_{c=1}^{C} F_i, \quad (11)
\]

where \( C \) is the number of classes. Table 5 shows the experimental results, where “MTL+TRAIN-local” represents the accuracy of the pseudo-labels from the domain-specific classifier in our method, and “MTL+TRAIN-global-local” is the accuracy of the pseudo-labels generated by fusing the domain-specific classifier and the global classifier in our method. As observed in Table 5, “MTL+TRAIN-local” can improve the macro-f1 of FixMatch by +1.37 (90.63 vs. 89.26) and +1.27 (70.41 vs. 69.14) on PACS and Office-Home, respectively. This validates that using the independent task for each domain can indeed alleviate the interference of different domains so as to improve the pseudo-labels of unlabeled data. Besides, “MTL+TRAIN-global-local” outperforms all other methods, e.g., “MTL+TRAIN-global-local” increases the macro-f1 of StyleMatch by +1.35 (92.70 vs. 91.35) and +2.39 (73.07 vs. 70.68) on on PACS and Office-Home, respectively. Therefore, this confirms that our method can achieve better accuracy of pseudo-labels to enhance the generalization capability of the model. Furthermore, we also display the accuracy of pseudo-labels of different methods at different epochs on PACS and Office-Home in Figure 5, respectively. As seen, our method can give more accurate pseudo-labels at each epoch when compared with FixMatch and StyleMatch.
Table 5. The Accuracy of Pseudo-labels of Different Methods on PACS and Office-Home under the Case with 10 Labels Per Class

| Dataset      | Method                        | Precision | Recall | macro-f1 |
|--------------|-------------------------------|-----------|--------|----------|
| PACS         | FixMatch                      | 88.83     | 89.52  | 89.26    |
|              | StyleMatch                    | 90.84     | 91.52  | 91.35    |
|              | MTL+TRAIN-local               | 90.17     | 90.86  | 90.63    |
|              | MTL+TRAIN-global-local        | 92.23     | 92.82  | 92.70    |
| Office-Home  | FixMatch                      | 71.05     | 69.97  | 69.14    |
|              | StyleMatch                    | 72.67     | 71.41  | 70.68    |
|              | MTL+TRAIN-local               | 72.22     | 71.25  | 70.41    |
|              | MTL+TRAIN-global-local        | 74.86     | 73.76  | 73.07    |

Fig. 5. The precision of pseudo-labels of different methods at different epochs on PACS and Office-Home, respectively.

Fig. 6. Comparison between our MultiMatch and StyleMatch under different numbers of the labeled data on PACS. “ALL” denotes that all training samples have label information during training.

**Test on different numbers of the labeled data.** We also conduct the comparison between our proposed MultiMatch and StyleMatch under different numbers of the labeled data on PACS, as shown in Figure 6. According to this figure, except for the “5 labels per class” case, our MultiMatch obviously outperforms StyleMatch in all other cases. For example, when we use 160 labels per class, our MultiMatch improves the performance by +1.1% (83.43 vs. 82.33) when compared with StyleMatch. Besides, given the label information for all training samples, our method can also obtain better results than StyleMatch.

**Evaluation on the independent BN and the independent classifier.** In this part, we validate the effectiveness of the independent BN and the independent classifier in our method. Experimental results are listed in Table 6, where “w/ SBN” and “w/ SC” indicate using the shared BN and the
Table 6. Evaluation on the Independent BN and the Independent Classifier on PACS and Office-Home, Respectively

| Module      | PACS          |          | Office-Home |          |
|-------------|---------------|----------|-------------|----------|
|             | P  | A  | C  | S  | Avg | P  | A  | C  | P  | R  | Avg |
| w/ SBN      | 91.62 | 82.09 | 73.10 | 73.70 | 80.13 | 55.30 | 51.65 | 68.01 | 71.56 | 61.63 |
| w/ SC       | 91.07 | 78.10 | 71.12 | 76.12 | 79.10 | 54.68 | 52.28 | 67.38 | 71.22 | 61.39 |
| Ours        | 93.25 | 83.30 | 73.00 | 76.74 | 81.57 | 56.32 | 52.76 | 68.94 | 72.28 | 62.57 |

Fig. 7. Experimental results of the typical DG methods under the labeled samples in SSDG on PACS and Office-Home, respectively. Note that “L” denotes only using the labeled samples, and “L+U” uses the labeled and unlabeled samples together.

Effectiveness of the unlabeled data in SSDG. To validate the effectiveness of the unlabeled data in SSDG, we train the supervised DG methods using the labeled samples, including ResNet18, CrossGrad [42], DDAIG [65], and RSC [17]. Experimental results are shown in Figure 7. As observed in this figure, using the unlabeled data can obtain better performance than these conventional DG methods with the labeled data, which validates that the unlabeled samples are very meaningful in the SSDG task. Furthermore, our MultiMatch can effectively mine the information from these unlabeled data, which has been reported in Table 1.

Test on the supervised case. In the experiment, we train our model in the supervised setting, and compare it with some supervised DG methods, as reported in Tables 7 and 8. MLDG [22], MASF [9], and MetaReg [1] are meta-learning-based methods. FACT [53], RSC [17], and FS-DCL [19] are augmentation-based methods. In addition, VDN [50] and SNR [20] aim at learning domain-invariant features, and DAEL [66] is an ensemble learning-based method. Compared with these supervised methods, our MultiMatch is also competitive in the supervised case.
Table 7. Experimental Results of Different Methods under the Supervised Setting on PACS

| Method     | P   | A   | C   | S   | Avg  |
|------------|-----|-----|-----|-----|------|
| MLDG [22]  | 94.30 | 79.50 | 77.30 | 71.50 | 80.65 |
| MASF [9]   | 94.99 | 80.29 | 77.17 | 71.69 | 81.04 |
| MetaReg [1] | 95.50 | 83.70 | 77.20 | 70.30 | 81.68 |
| SNR [20]   | 94.50 | 80.3 | 78.20 | 74.10 | 81.80 |
| VDN [50]   | 94.00 | 82.60 | 78.50 | 82.70 | 84.45 |
| FACT [53]  | 95.15 | 85.37 | 78.38 | 79.15 | 84.51 |
| MultiMatch (ours) | 95.79 | 84.44 | 75.83 | 82.82 | 84.72 |

The **bold** and gray are the best result and the second-best result, respectively.

Table 8. Experimental Results of Different Methods under the Supervised Setting on Office-Home

| Method     | A   | C   | P   | R   | Avg  |
|------------|-----|-----|-----|-----|------|
| RSC [17]   | 58.42 | 47.90 | 71.63 | 74.54 | 63.12 |
| DAEL [66]  | 59.40 | 55.10 | 74.00 | 75.70 | 66.05 |
| SNR [20]   | 61.20 | 53.70 | 74.20 | 71.50 | 66.10 |
| FSDCL [19] | 60.24 | 53.54 | 74.36 | 76.66 | 66.20 |
| FACT [53]  | 60.34 | 54.85 | 74.48 | 76.55 | **66.56** |
| MultiMatch (ours) | 59.71 | 56.25 | 74.28 | 75.70 | 66.49 |

Table 9. Comparison between Our Method and L2G [18] on PACS and Office-Home under 10 Labels per Class

| PACS          | Method | P   | A   | C   | S   | Avg  |
|---------------|--------|-----|-----|-----|-----|------|
| L2G [18]      | **93.40** | 83.24 | 69.18 | 75.70 | 80.38 |
| Ours          | 93.25  | **83.30** | **73.00** | **76.74** | **81.57** |

| Office-Home   | Method | A   | C   | P   | R   | Avg  |
|---------------|--------|-----|-----|-----|-----|------|
| L2G [18]      | 55.92  | 51.58 | 68.12 | 71.62 | 61.81 |
| Ours          | **56.32** | 52.76 | **68.94** | **72.28** | **62.57** |

Comparison with “local-to-global” method. In our task, we achieve the scheme in Reference [18] that uses each local branch prediction to distill global branch prediction. Experimental results are shown in Table 9. As seen in this table, our method can obtain the better performance than the L2G [18]. The advantage of our method is fusing the prediction of global task based on our “PredictLabel” scheme in Function 1 can further improve the accuracy of the pseudo-label, which has been validated in Table 5 of our article.

5 CONCLUSION

In this article, we aim to tackle the SSDG task. Different from the typical semi-supervised task, the challenge of SSDG is that there exist multiple different domains with latent distribution discrepancy. To address this issue, we first explore the theory of multi-domain learning to generate more accurate pseudo-labels for unlabeled samples. Then, we propose to utilize a multi-task...
learning framework to mitigate the impact of the domain discrepancy and sufficiently exploit all training samples, which can effectively enhance the model’s generalization. We conduct the experiment on multiple benchmark datasets, which verifies the effectiveness of the proposed method.

REFERENCES

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. 2018. MetaReg: Towards domain generalization using meta-regularization. In Advances in Neural Information Processing Systems (NeurIPS’18). 1006–1016.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. Mach. Learn. 79, 1 (2010), 151–175.

[3] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. 2020. ReMixMatch: Semi-supervised learning with distribution matching and augmentation anchoring. In International Conference on Learning Representations (ICLR’20).

[4] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. MixMatch: A holistic approach to semi-supervised learning. In Advances in Neural Information Processing Systems (NeurIPS’19). 5050–5060.

[5] Fabio Maria Carlucci, Antonio D’Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. 2019. Domain generalization by solving jigsaw puzzles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’19). 2229–2238.

[6] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. 2019. Domain-specific batch normalization for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’19). 7354–7362.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’09). 248–255.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the National Conference on Artificial Intelligence (NAACL-HLT).

[9] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. 2019. Domain generalization via model-agnostic learning of semantic features. In Advances in Neural Information Processing Systems (NeurIPS’19). 6447–6458.

[10] Sichao Fu, Weifeng Liu, Weili Guan, Yicong Zhou, Dapeng Tao, and Changsheng Xu. 2021. Dynamic graph learning convolutional networks for semi-supervised classification. ACM Trans. Multimedia Comput. Commun. Appl. 17, 1s (2021), 1–13.

[11] Chengyue Gong, Dilin Wang, and Qiang Liu. 2021. AlphaMatch: Improving consistency for semi-supervised learning with alpha-divergence. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’21). 13683–13692.

[12] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. 2019. DLOW: Domain flow for adaptation and generalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’19). 2477–2486.

[13] Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. In Advances in Neural Information Processing Systems (NeurIPS’04). 529–536.

[14] Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, and Jerome H. Friedman. 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Vol. 2.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’16). 770–778.

[16] Xun Huang and Serge J. Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the International Conference on Computer Vision (ICCV’17). 1510–1519.

[17] Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. 2020. Self-challenging improves cross-domain generalization. In Proceedings of the European Conference on Computer Vision (ECCV’20). 124–140.

[18] Takashi Isobe, Xu Jia, Shuaijun Chen, Jianzhong He, Yongjie Shi, Jianzhuang Liu, Huchuan Lu, and Shengjin Wang. 2021. Multi-target domain adaptation with collaborative consistency learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’21). 8187–8196.

[19] Seokkyu Jeon, Kibeom Hong, Pilhyeon Lee, Jewook Lee, and Hyeran Byun. 2021. Feature stylization and domain-aware contrastive learning for domain generalization. In Proceedings of the ACM International Conference on Multimedia (MM’21). 22–31.

[20] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. 2022. Style normalization and restitution for domain generalization and adaptation. IEEE Trans. Multimedia 24 (2022), 3636–3651.
[21] Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. In Proceedings of the International Conference on Learning Representations (ICLR’17).

[22] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. 2018. Learning to generalize: Meta-learning for domain generalization. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI’18). 3490–3497.

[23] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. 2017. Deeper, broader and artier domain generalization. In Proceedings of the International Conference on Computer Vision (ICCV’17). 5543–5551.

[24] Da Li, Jianguo Zhang, Tongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. 2019. Episodic training for domain generalization. In Proceedings of the International Conference on Computer Vision (ICCV’19). 1446–1455.

[25] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. 2018. Domain generalization with adversarial feature learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’18). 5400–5409.

[26] Junnan Li, Caiming Xiong, and Steven C.H. Hoi. 2021. CoMatch: Semi-supervised learning with contrastive graph regularization. In Proceedings of the International Conference on Computer Vision (ICCV’21). 9475–9484.

[27] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. 2018. Deep domain generalization via conditional invariant adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV’18). 647–663.

[28] Yajing Liu, Zhiwei Xiong, Ya Li, Yuning Lu, Xinmei Tian, and Zheng-Jun Zha. 2023. Category-stitch learning for union domain generalization. ACM Trans. Multimedia Comput. Commun. Appl. 19, 1 (2023), 1–19.

[29] Yajing Liu, Zhiwei Xiong, Ya Li, Xinmei Tian, and Zheng-Jun Zha. 2021. Domain generalization via encoding and resampling in a unified latent space. IEEE Trans. Multimedia (2021).

[30] Hao Luo, Wei Jiang, Youzhi Gu, Fuxi Liu, Xingyu Liao, Shenqi Lai, and Jianyang Gu. 2020. A strong baseline and batch normalization neck for deep person re-identification. IEEE Trans. Multimedia 22, 10 (2020), 2597–2609.

[31] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In Proceedings of the International Conference on Machine Learning (ICML’13). 10–18.

[32] Hyeonseob Nam, Hyunjae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. 2021. Reducing domain gap by reducing style bias. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’21). 8690–8699.

[33] Islam Nassar, Samitha Herath, Ehsan Abbasnejad, Wray Buntine, and Gholamreza Haffari. 2021. All labels are not created equal: Enhancing semi-supervision via label grouping and co-training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’21). 7241–7250.

[34] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. 2022. DASO: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’22). 9786–9796.

[35] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In Proceedings of the International Conference on Computer Vision (ICCV’19). 1406–1415.

[36] Lei Qi, Lei Wang, Jing Huo, Yinhuan Shi, and Yang Gao. 2021. Reducing domain gap by reducing style bias. In Proceedings of the International Conference on Computer Vision (ICCV’19). 100 (2020), 107124.

[37] Lei Qi, Lei Wang, Yinhuan Shi, and Xin Geng. 2022. A novel mix-normalization method for generalizable multi-source person re-identification. IEEE Trans. Multimedia (2022).

[38] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. 2019. Multi-component image translation for deep domain generalization. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV’19). 579–588.

[39] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. 2020. Correlation-aware adversarial domain adaptation and generalization. Pattern Recogn. 100 (2020), 107124.

[40] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 6 (2017), 1137–1149.

[41] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. 2020. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In Advances in Neural Information Processing Systems (NeurIPS’20).
[44] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Advances in Neural Information Processing Systems (NeurIPS’17). 1195–1204.

[45] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 11 (2008), 2579–2605.

[46] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep hashing network for unsupervised domain adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’17). 5018–5027.

[47] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. 2021. Generalizing to unseen domains: A survey on domain generalization. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI’21). 4627–4635.

[48] Xudong Wang, Zhirong Wu, Long Lian, and Stella X. Yu. 2022. Debiasied learning from naturally imbalanced pseudo-labels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’22). 14647–14657.

[49] Yue Wang, Lei Qi, Yinghuan Shi, and Yang Gao. 2022. Feature-based style randomization for domain generalization. ACM Trans. Multimedia Comput. Commun. Appl. 18, 4 (2022), 99:1–99:16.
MultiMatch: Multi-task Learning for Semi-Supervised Domain Generalization

[67] Ziqi Zhou, Lei Qi, Xin Yang, Dong Ni, and Yinghuan Shi. 2022. Generalizable cross-modality medical image segmentation via style augmentation and dual normalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’22). 20856–20865.

[68] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the International Conference on Computer Vision (ICCV’17). 2242–2251.

Received 29 November 2022; revised 15 September 2023; accepted 15 February 2024