Generalizing to Unseen Domains via Patch Mix

Juncheng Yang
Wuhan University
Zuchao Li
Wuhan University
Chao Li
JD Health International Inc.
Shuai Xie
JD Explore Academy
Wei Yu
Wuhan University
Shijun Li (shjli@whu.edu.cn)
Wuhan University

Research Article

Keywords: Domain Generalization, PatchMix, Domain Discriminator, Vision Transformer, Data Augmentation

Posted Date: August 7th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3221198/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Multimedia Systems on January 19th, 2024. See the published version at https://doi.org/10.1007/s00530-023-01213-8.

Generalizing to Unseen Domains via Patch Mix

Juncheng Yang[1,3], Zuchao Li[2*], Chao Li[4], Shuai Xie[5], Wei Yu[1], Shijun Li[1*]

[1]School of Computer Science, Wuhan University, Wuhan, 430072, Hubei, China.

[2]National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan, 430072, Hubei, China.

[3]School of Electronic Information Engineering, Henan Polytechnic Institute, Nanyang, 473000, Henan, China.

[4]JD Health International Inc., Beijing, China.

[5]JD Explore Academy, Beijing, China.

*Corresponding author(s). E-mail(s): shjli@whu.edu.cn;

Abstract

Domain Generalization (DG) aims to transfer knowledge learned from multiple source domains to unseen domains. One of the primary challenges hinders DG is the insufficient diversity of source domains, which hampers the model's ability to learn to generalize. Traditional data augmentation methods, which fuse content, style, labels, etc., unable to effectively learn the global features from the source domains. In this paper, we present an innovative approach to domain generalization learning technique, called PatchMix, by stitching the patches of different source domains together to build domain mixed samples. This approach helps the model to learn the common features of different source domains at every glimpse. Meanwhile, a domain discriminator is introduced to preserve the model's ability to distinguish the source domains, which is proved to be helpful for the model to generalize to unseen domains. To our best knowledge, we are the first to unveil the equation that elucidates the correlation between the number of patches and the number of source domains. Our method, PatchMix, outperforms the current state-of-the-art (SOTA) on four benchmark datasets.

Keywords: Domain Generalization, PatchMix, Domain Discriminator, Vision Transformer, Data Augmentation

1 Introduction

Deep learning has made great achievements in computer vision, natural language processing, machine learning, etc., which assumes that training and test data obey the principle of independent and identical distribution. In the real world, however, data is very diverse in background, shape, and color, making it difficult to meet this assumption. The phenomenon in which the training and test data have a significantly different distribution is known as the domain shift problem [1–3]. This problem results

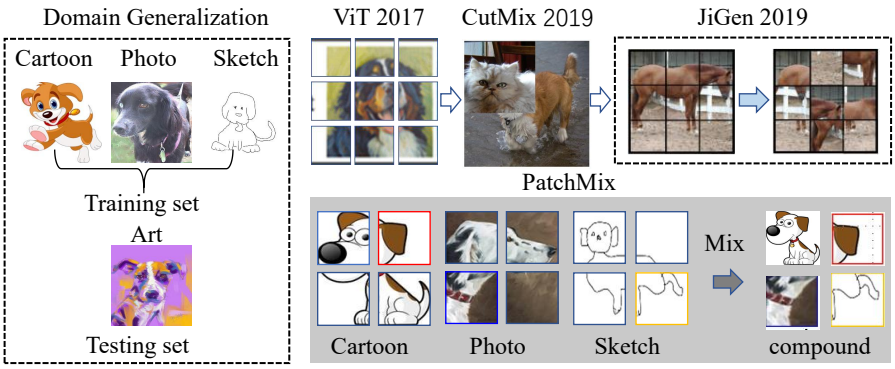

Fig. 1 Unlike Vision Transformer (ViT) [4], CutMix [5], and JiGen [6] methods, PatchMix reads images from different domains and generates new images to improve the generalization ability of the model.

in a well-trained model that can only achieve poor performance on the test set, either crashing or failing to converge. Even worse, the training data is not accessible, and we lack any information about the shape, distribution, or labels of the data before testing. This study is known as domain generalization (DG) [1, 7–10]. As shown on the left side of Figure 1, the model is trained on the data from Cartoon, Photo, and Sketch, then evaluate on the Art data. This is common in autonomous driving and medical diagnosis, thus attracting more and more attention.

Aligning the distribution of data through adversarial learning [11–14] is mainstream in the DG area. Besides, learning the data commonality of different domains through meta-learning [15–17], self-supervised learning [18, 19] and decoupling the domain-invariant and domain-first features in the data through feature disentanglement [20, 21] are also representative works. Recently, Huang et al. [22] proposed to enhance the data diversity using data augmentation such as changing the angle, cropping the raw data, or constructing generative adversarial networks [23–26]. Zhou et al. [27] edited an unseen style and then mixed the content and style of images to generate new images, which achieves a higher diversity of training data as well as better generalization.

In addition to the aforementioned DG methods, there are some data augmentation-based techniques to improve model generalization, shown on the right side of Figure 1. One such method is MixUp [28], proposed by Zhang et al., which mixes the contents and labels of two images in proportion to generate a new image. However, MixUp

requires prior knowledge about the linear interpolation ratio, which is not available in DG problems where the target domain is not visible during training. Additionally, MixUp is typically designed for blending two images and not work well for more than three images, making it unsuitable for DG tasks. Another method is CutMix [5], proposed by Yun et al., which randomly selects a rectangular patch from one image and fills the corresponding region of another image with the selected patch. However, CutMix is designed to handle the combination of two images, making it less suitable for handling multiple domain images in DG tasks. JiGen [6] is another method based on the idea of patch, which resembles a child's jigsaw puzzle. JiGen uses self-supervised signals to solve the problem of disrupted image patch order. However, JiGen mainly focuses on solving the internal order of a single image and is not suitable for handling multiple images. In contrast, our proposed PatchMix method aims to address the domain generalization problem by using a different data fusion strategy. It cuts the data from the same class but different domains into patches and assembles them into domain mixed images for training. By doing this, PatchMix enhances the data diversity and improves the generalization of the model across multiple domains.

Beyond data augmentation, as a simple and effective technique for DG, PatchMix leverages the idea of splitting images into patches from Vision Transformer (ViT) and goes beyond traditional data augmentation methods to address the domain generalization problem. It reads data from multiple domains simultaneously, divides the images into different patches, and then assembles them into new composite images in various combinations. The process is illustrated in the lower shaded part on the right side of Figure 1. For example, if we take a picture of a dog from the Cartoon, Photo, and Sketch datasets, PatchMix will split these images into different patches and then select one patch from each domain to stitch them together into a new composite image. By using PatchMix, the model can effectively learn common features across different domains, enabling it to generalize better to unseen domains.

Meanwhile, we introduce domain labels as a supervised signal to achieve better distinction among patches, which enables the model to adapt to the differences among various domains and effectively distinguish the common and specific features present in each domain. Furthermore, it facilitates learning the domain-invariant and domain-variant aspects across different domains, thus mitigating the risk of overfitting and ultimately enhancing the model's generalization capabilities. Additionally, through experiments conducted on various datasets, we have derived insights into the correlation between the number of domains and the number of in-sample patches. These findings serve as a valuable reference, aiding in the optimal selection of the number of patches to maximize the efficacy of domain generalization. In summary, our major contributions are as follows:

- We propose the PatchMix method, which cuts the data from the same class but different domains into patches and assembles them into domain mixed images for effective cross-domain training.
- In model training, we introduce the domain discriminator as a regularization term, which allows the model to learn the common and specific properties among different domains and improves the generalization of the model.

- Experiments on four datasets, VLCS, PACS, OfficeHome, and Domainnet generalize the relationship between the number of patches and the number of domains, and show improvements over SOTA methods.

2 Related Work

Domain Generalization

The domain generalization (DG) methods can be categorized into three main approaches: data manipulation, representation learning, and strategy learning. The first approach is data manipulation: Yue et al. [29] proposed style randomization of real images with synthetic data from auxiliary datasets. Zhao et al. [26] utilized data generation methods to enhance model generalization. Some DG methods aim to learn domain-invariant representations by minimizing the domain discrepancy between available source data [12, 30, 31]. The objective is to learn features that are invariant to multiple source domains, thus generalizing them to unseen domains. The second is representation learning: Vapnik et al. [32] introduced Empirical Risk Minimization (ERM) as a classic approach. Muandet et al. [33] employed Maximum Mean Discrepancy (MMD) constraint. Ganin et al. [11] proposed Domain Adversarial Neural Network (DANN) to acquire domain-invariant feature representations. Li et al. [12] introduced a Conditional Invariant Adversarial Network (CDANN) for learning domain-invariant representations. Sun et al. [34] used Deep CORAL to align layer activation correlations in deep neural networks with nonlinear transformations. Arjovsky et al. [35] introduced Invariant Risk Minimization (IRM), enforcing uniformity of the optimal classifier in the representation space across all domains. The third is an analysis from learning strategies: Li et al. [16] proposed Meta-Learning for Domain Generalization (MLDG). Sagawa et al. [36] proposed GroupDRO, which requires explicit group annotation of samples for DG. Krueger et al. [37] reduced the variance of training domain risk extrapolation (VRex). Cha et al. [38] introduced the Stochastic Weighted Dense Averaging (SWAD) method for locating the minimum. Iwasawa et al. [39] focused on the Test-Time Template Adjuster (T3A) phase and calculated pseudo-prototype representations for each category, classifying samples based on their distance from the pseudo-prototypes. Carlucci et al. [6] improved model generalization by learning self-supervised signals solving jigsaw puzzles (JiGen) on the same images. Zhou et al. [27] proposed a probabilistic hybrid instance-level training sample-based approach. In this paper, the proposed method involves cropping and assembling images of the same category from different domains into patches for model training, leading to good results in domain generalization.

Vision Transformer

The Transformer [40] architecture was originally introduced in the field of natural language processing. However, its application was extended to computer vision tasks with the introduction of ViT [4]. ViT transformed images into sequences of patches and leveraged the benefits of global context modeling and large-scale pre-training data, rather than relying on image-specific inductive biases like Convolutional Neural Networks (CNNs). DeiT-Small [41] introduced a token-based distillation strategy that

allowed ViT to be trained on smaller datasets while still achieving better results than CNNs trained only on ImageNet data. CvT [42] was a hybrid model that combined the powerful feature extraction capabilities of CNNs and ViT, resulting in a more robust model compared to ViT alone. SDViT [43] proposed using self-distillation during training to reduce overfitting to the source domain. T2T-ViT [44] introduced a progressive tokenization approach, where adjacent tokens were recursively aggregated into a single token, leading to more effective representations. Swin Transformers [45] presented a novel approach that confined self-attention to non-overlapping local windows, establishing cross-window connections using shift windows. GE-ViTs [46] introduced a generalization-enhanced view of ViT, leveraging information theory and self-supervised learning to improve the model's generalization performance. All of these methods have achieved promising results in the domain generalization task.

Data Augmentation

Data augmentation is a common tool for deep learning and it is important to provide the performance of the model. Basic image processing such as rotation, flip, and crop are all direct operations on the image itself. Mixup [28] conducted convex combinations of sample pairs and their labels and established a linear relationship between data increments and supervised signals. CutMix [5] replaced deleted areas with patches from other images and could generate more natural images compared to Mixup. Fmix [47] utilized random masks of various shapes to enhance the performance of the model. AugMix [48] introduced multiple enhancement operations mixed into three enhancement chains, which are then combined using the convex combination principle. AutoAugment [49] employed an automatic search method to find effective data augmentation strategies. FastAutoAugment [50] proposed a fast augmentation algorithm for density matching, enabling the identification of suitable augmentation strategies through a more efficient search process. RandAugment [51] focused on saving computational resources by reducing the search space for data augmentation. In this paper, the patch method is enhanced, and the domain label is incorporated as a supervised signal during model training to further improve the generalization of the model. By leveraging both the improved patch method and domain label supervision, the proposed approach aims to achieve better performance in the domain generalization task.

3 The Proposed PatchMix Method

3.1 Preliminaries

Problem Settings

We assume a set of M available source domains as $\mathcal{S} = \{\mathcal{S}^m\}_{m=1}^{M}$, where $\mathcal{S}^m = \{(x_j^m, y_j^m)\}_{j=1}^{N}\}$ denotes a distribution over the input space $\mathcal{X}$ and label space $\mathcal{Y}$, N is the number of samples in $\mathcal{S}^m$. In addition, we assume a set of unseen target domains $\{\mathcal{T}\}_{t=1}^{T}$, where T represents the number of target domains, typically set to 1. DG aims to learn a proper mapping $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$, which is trained on the available $\mathcal{S}$ and generalized well on the unseen $\mathcal{T}$.

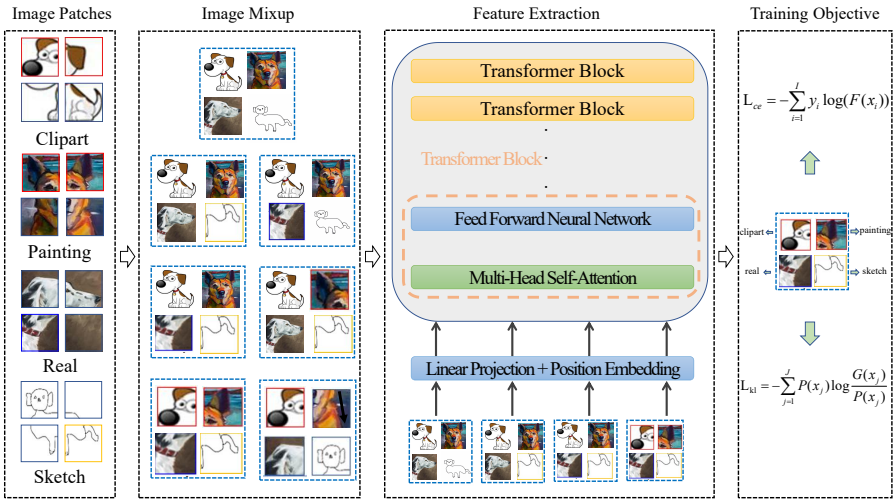

Fig. 2 Illustration of the proposed PatchMix. First, we cut the images of the multi-source domains S into small patches, including four domains above. Second, we sample one patch from each source domain S^m, and stitch them into domain mixed images, where each patch is in a fixed quadrant according to domain index m. For example, the Clipart domain is fixed on the top-left quadrant, while the Sketch domain is fixed on the bottom-right quadrant. Third, these domain mixed images are used to train a standard vision transformer. Finally, we use the standard image classification loss and the proposed domain classification loss to supervise the training process.

ViT Backbone

The ViT backbone first reshapes an image $x \in \mathcal{R}^{H\times W\times C}$ into a sequence of flattened 2D patches $\mathcal{R}^{G\times(P^2\cdot C)}$, where C is the number of channels, (H, W) represents the resolution of the original image, (P, P) denotes the resolution of each image patch, and $G = HW/P^2$ is the number of patches. The output of layer 0 a_0 can be expressed as:

$$a_0 = [x_{CLS}; x_p^1 U; x_p^2 U; ...; x_p^G U] + U_{pos}, \tag{1}$$

where x_{CLS} is a learnable embedded [CLS] token, $x_p \in \mathcal{R}^{G\times(P^2\cdot C)}$, $U \in \mathcal{R}^{(P^2\cdot C)\times H}$, $U_{pos} \in \mathcal{R}^{(G+1)\times H}$ means the standard learnable position embedding.

Then, the Self-Attention (SA) module projects these patches into three type vectors: queries $Q \in \mathcal{R}^{G\times d_k}$, keys $K \in \mathcal{R}^{G\times d_k}$ and values $V \in \mathcal{R}^{G\times d_v}$. d_k and d_v indicate their dimensions. SA module aims to emphasize the relationships among patches by computing the attention score as follows:

$$SA(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{2}$$

The Multi-Head Self-Attention (MSA) concatenates multiple scaled dot-product self attention modules, which is defined as:

$$MSA(Q, K, V) = Concat(head_1, ..., head_h)W^O, head_i = SA(QW_i^Q, KW_i^K, VW_i^V), \tag{3}$$

where h is the number of heads, QW_i^Q, KW_i^K and VW_i^V are three projection functions of $head_i$, W^O is the output projection function. The ViT block is composed of MSA and MLP modules, which apply Layer Normalization (LN) before each block and residual connectivity after each block, this can be defined as follows:

$$a_l^{'} = MSA(LN(a_{l-1})) + a_{l-1} \tag{4}$$

$$a_l = MLP(LN(a_l^{'})) + a_l^{'}. \tag{5}$$

where the $l \in \{1, \ldots, L\}$ means the number of block of ViT. In this paper, the value of L is set to 12 for both ViT and DeiT-Small. After several iterations of layers, the final prediction result can be represented as $LN(a_L^0)$.

3.2 PatchMix

Patch and Mixup

PatchMix is a novel approach that integrates the concepts of image patches and mixup. In traditional mixup, the images from different classes or domains are linearly combined to synthesize new samples, which encourages the model to learn from diverse sources. However, this method may not fully exploit the unique information present in individual images, especially easy to lose the domain characteristics. As shown in Figure 2, PatchMix addresses this limitation by taking advantage of both image patches and mixup. By combining patches from various domains, we create domain mixed samples that offer a more comprehensive view of the data distribution. This technique enhances the model's ability to capture the underlying features and enables more robust generalization across domains.

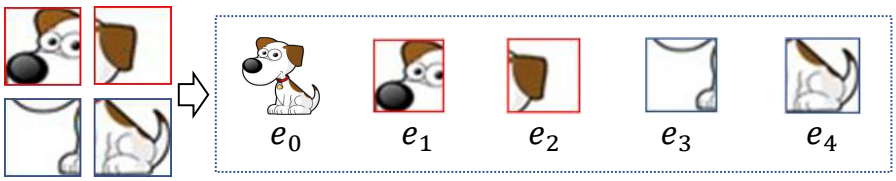

Fig. 3 An image can generate five type patches: e_0, e_1, e_2, e_3, and e_4 with two strategies.

Given an image, we apply two strategies to generate the image patches, as illustrated in Figure 3. The first strategy zooms out the original image to a fixed patch size, like e_0. The second strategy cuts a quarter of the original image as one patch, like e_1, e_2, e_3, and e_4. The domain mixed images are constructed by sampling patches from various source domains with these two strategies. This process is shown in Algorithm 1, where Bernoulli(0.5) is a Bernoulli distribution with probability 0.5, Uniform$(1, |S_i|)$ is a uniform distribution over the integers from 1 to the number of images in source domain $|S_i|$, the function sample(S_i) selects an image from the i-th domain, while the function sample_patch$(S_i[j])$ cuts a patch from the j-th image in $|S_i|$. After sampling Φ patches, we stitch them together to build a domain-mixed sample.

Algorithm 1 Sampling patches from source domains

for i=1 to Φ **do**
  $x \sim$ Bernoulli(0.5)
  if $x = 1$ **then**
    P_i = sample(S_i) ▷ sample the original image
  else
    $j \sim$ Uniform$(1, |S_i|)$
    P_i = sample_patch($S_i[j]$) ▷ sample one patch from the original image
  end if
end for

Table 1 The generation process of Domain Soft label.

Domain Soft Label Encodings				
Φ	S	Relationship between Φ and S	Encodings	Is valid
$\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$	$\{S_1\}$	$\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\} \in S_1$	$\{1, 0, 0\}$	*True*
$\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$	$\{S_1, S_2, S_3\}$	$\varphi_1 \in S_1, \{\varphi_2, \varphi_4\} \in S_2, \varphi_3 \in S_3$	$\{0.25, 0.5, 0.25\}$	*True*
$\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$	$\{S_1, S_2, S_3\}$	$\varphi_1 \in S_1, \varphi_2 \in S_2, \varphi_3 \in S_3, \varphi_4 \in \phi$	$\{0.333, 0.333, 0.333\}$	*True*
$\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$	$\{S_1, S_3\}$	$\varphi_1 \in S_1, \{\varphi_2, \varphi_4\} \in \phi, \varphi_3 \in S_3$	$\{0.5, 0, 0.5\}$	*True*
$\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$	$\{S_1\}$	$\varphi_1 \in S_1, \{\varphi_2, \varphi_3, \varphi_4\} \in \phi$	$\{1, 0, 0\}$	*True*
$\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$	$\{\}$	$\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\} \in \phi$	$\{0, 0, 0\}$	*False*

Training Objective

Given an input x_i, a standard cross-entropy loss is used to supervise the image classification process, which is defined as:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{I} y_i log(\mathcal{F}(x_i)), \tag{6}$$

where I is the number of training samples, $\mathcal{F}(x_i)$ and y_i are the prediction and ground truth, respectively.

Domain Soft Label

To enhance the learning of valuable features from diverse domains, we introduce a novel domain label supervision method that leverages both general and specific information from each domain patch. Our approach aims to train a versatile classifier capable of not only making accurate classifications but also determining the contribution of each source domain. Unlike previous methods that attempt to confuse multiple domains, we take inspiration from human cognitive processes, where individuals make inferences based on their retained domain knowledge. Under this perspective, we introduce the concept of imbalanced domain mixed sample to augment the training process, thus improving the model's ability to distinguish between different domains. To achieve this, we design a simple yet effective domain-imbalanced sampling strategy, where each patch can be sampled from any source domains or set to zero as ϕ. For instance,

considering a domain-mixed image with four patches $\varphi_1, \varphi_2, \varphi_3, \varphi_4$ from three source domains S_1, S_2, S_3, Table 1 illustrates the candidate domain soft labels. We use the Kullback-Leibler divergence as the domain classification loss, which can be defined as:

$$\mathcal{L}_{kl} = -\sum_{j=1}^{J} P(x_j) log \frac{G(x_j)}{P(x_j)}), \tag{7}$$

where $P(x_j)$ is the domain label prediction, $G(x_j)$ is the domain label ground truth, J is the number of source domains in this sample. Finally, the overall learning objective is formulated as:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{kl}, \tag{8}$$

where λ is a coefficient to control the proportion of two loss functions.

3.3 Analytical Evaluation

During the training process, PatchMix integrates information from patches in multiple source domains to enhance the diversity of training data. The method aims to find a model $f \in \mathcal{F}$, such that the loss function, defined as the expectation of the discrepancy between the hypothetical $f_\theta(x)$ and the actual sample y, is minimized. Formally, this loss function can be defined as:

$$\mathcal{L}(f) = E[\ell(f_\theta(x), y)] = \int \ell(f_\theta(x), y) d(p(x, y)), \tag{9}$$

where $\ell(\cdot)$ represents the loss function, $f_\theta(x)$ is the output of the model f on input x parameterized by θ, and (x, y) is a data sample drawn from the distribution p, the goal of the model is to find a $f^* = argmin\mathcal{L}(f)$. The approximation of $\mathcal{L}(f)$ is ERM can be defined as:

$$\mathcal{L}_{erm}(f) = \frac{1}{M} \sum_{i=1}^{M} \ell(f_\theta(x_i), y_i). \tag{10}$$

The empirical Rademacher complexity of the hypothesis class can be used to define $\mathcal{R}_M(\mathcal{F})$. This complexity measure is a tool for characterizing the capacity of the hypothesis class to fit the empirical distribution. It can be defined as follows:

$$\mathcal{R}_M(\mathcal{F}) = \mathbb{E}_\rho \left[\sup_{f \in \mathcal{F}} \frac{1}{M} \sum_{i=1}^{M} \rho_i \ell(f(x_i), y_i) \right], \tag{11}$$

where ρ_i are Rademacher random variables. By using $\mathcal{R}_M(\mathcal{F})$, we can establish bounds [52–55] on the generalization error of the learning algorithm. Note that $\Psi(\mathcal{S}) = \sup_{f \in \mathcal{F}} \frac{1}{M} \sum_{i=1}^{M} (\mathcal{L}(f) - \mathcal{L}_{erm}(f))$ satisfies the bounded differences property required by McDiarmid's inequality. This property means that if we construct $\mathcal{S}'$

by replacing any one of the (x, y) pairs in $\mathcal{S}$ with another random variable also drawn from p, then $|\Psi(\mathcal{S}) - \Psi(\mathcal{S}')| \leq \frac{1}{M}$. As a result, McDiarmid's inequality can be used to define the confidence interval as follows: with confidence at least $1 - \frac{\delta}{2}$, we have $\Psi(\mathcal{S}) \leq \mathbb{E}_{S_p}[\Psi(\mathcal{S})] + \sqrt{\frac{\ln(2/\delta)}{2M}}$. We proceed by bounding the expected value of $\mathbb{E}_{S_p}[\Psi(\mathcal{S})]$ as:

$$\mathbb{E}_{S_p}[\Psi(\mathcal{S})] = \mathbb{E}_{S_p}\left[\sup_{f\in\mathcal{F}}\left(\mathbb{E}_{S_q}[\mathcal{L}(f)] - \frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_{erm_i}(f)\right)\right] \tag{12}$$

$$= \mathbb{E}_{S_p}\left[\sup_{f\in\mathcal{F}}\mathbb{E}_{S_q}\left(\frac{1}{M}\sum_{i=1}^{M}(\mathcal{L}_{q_i}(f) - \mathcal{L}_{erm_i}(f))\right)\right] \tag{13}$$

$$\leq \mathbb{E}_{S_p,S_q}\left[\sup_{f\in\mathcal{F}}\frac{1}{M}\sum_{i=1}^{M}\left(\mathcal{L}_{q_i}(f) - \mathcal{L}_{erm_i}(f)\right)\right] \tag{14}$$

$$= \mathbb{E}_{S_p,S_q}\mathbb{E}_{\rho}\left[\sup_{f\in\mathcal{F}}\frac{1}{M}\sum_{i=1}^{M}\rho_i\left(\mathcal{L}_{q_i}(f) - \mathcal{L}_{erm_i}(f)\right)\right] \tag{15}$$

$$\leq 2\mathbb{E}_{S_p}\mathbb{E}_{\rho}\left[\sup_{f\in\mathcal{F}}\frac{1}{M}\sum_{i=1}^{M}\rho_i\mathcal{L}_{erm_i}(f)\right] \tag{16}$$

$$\leq 2\mathcal{R}_M(\mathcal{F}) + 2\sqrt{\frac{ln(2/\delta)}{2M}}, \tag{17}$$

where the ρ is a vector of Radmacher random variables. Similarly, for $\Psi(\mathcal{H})$ with M domains and N samples in each domain can be defined as:

$$\Psi(\mathcal{H}) = 2\mathcal{R}_{MN}(\mathcal{F}) + 3\sqrt{\frac{ln(2/\delta)}{2MN}}. \tag{18}$$

To minimize the expected loss $\mathcal{L}(f)$, a combination of empirical risk minimization $\mathcal{L}_{erm}$, $\Psi(\mathcal{S})$ and $\Psi(\mathcal{H})$ can be utilized, the function $\mathcal{L}(f)$ can be defined as:

$$\mathcal{L}(f) \leq \mathcal{L}_{erm}(f) + 2\mathcal{R}_{MN}(F) + 2\mathcal{R}_M(F) + 3\sqrt{\frac{ln(2/\delta)}{2MN}} + 3\sqrt{\frac{ln(2/\delta)}{2M}}, \tag{19}$$

where M represents the number of domains, N denotes the number of training samples in each domain, and $\mathcal{R}(F)$ represents the empirical Rademacher complexity of the hypothesis class, it becomes evident that PatchMix effectively reduces the generalization gap. This reduction is achieved by enhancing the quantity and quality of training data through the process of patch transformation, as indicated in Eq. (19).

4 Experiment

In this section, we present an overview of the datasets used in our experiments and provide a detailed outline of the experimental setup. Subsequently, we conduct extensive

Table 2 PatchMix was compared to common DG methods on four datasets, VLCS, PACS, OfficeHome, and DomainNet, with the best results shown in bold.

Algorithm	Backbone	Params	VLCS	PACS	OfficeHome	DomainNet	Average
ERM [32]	ResNet-50 [56]	25.6M	77.4 ± 0.3	85.7 ± 0.5	67.5 ± 0.5	41.2 ± 0.2	68.0
CORAL [34]	ResNet-50 [56]	25.6M	77.7 ± 0.5	86.0 ± 0.2	68.6 ± 0.4	41.8 ± 0.2	68.5
DANN [11]	ResNet-50 [56]	25.6M	78.7 ± 0.3	84.6 ± 1.1	65.4 ± 0.6	38.4 ± 0.0	66.8
GroupDRO [36]	ResNet-50 [56]	25.6M	78.1 ± 0.2	86.8 ± 0.4	68.4 ± 0.1	39.6 ± 0.3	68.2
PatchMix	ResNet-50 [56]	25.6M	78.8 ± 0.3	86.1 ± 0.4	68.9 ± 0.2	42.1 ± 0.5	**69.0**
ERM [32]	ViT [4]	86.6M	78.5 ± 0.2	83.7 ± 0.6	78.6 ± 0.3	46.0 ± 0.1	71.7
CORAL [34]	ViT [4]	86.6M	78.6 ± 0.4	83.8 ± 0.6	78.7 ± 0.3	46.1 ± 0.0	71.8
DANN [11]	ViT [4]	86.6M	78.8 ± 0.3	84.1 ± 0.6	78.9 ± 0.7	46.2 ± 0.5	72.0
GroupDRO [36]	ViT [4]	86.6M	78.7 ± 0.4	83.6 ± 0.1	78.7 ± 0.3	46.1 ± 0.5	71.8
PatchMix	ViT [4]	86.6M	79.0 ± 0.2	84.5 ± 0.6	80.6 ± 0.3	46.6 ± 0.2	**72.7**
ERM [32]	DeiT-Small [41]	22.1M	78.3 ± 0.1	87.2 ± 0.6	71.4 ± 0.3	45.5 ± 0.2	70.6
CORAL [34]	DeiT-Small [41]	22.1M	78.5 ± 0.2	87.5 ± 0.8	72.3 ± 0.3	45.7 ± 0.1	71.0
DANN [11]	DeiT-Small [41]	22.1M	78.4 ± 0.2	87.2 ± 0.4	72.8 ± 0.3	45.1 ± 0.7	70.9
GroupDRO [36]	DeiT-Small [41]	22.1M	78.6 ± 0.4	87.6 ± 0.3	72.4 ± 0.6	45.6 ± 0.7	71.1
PatchMix	DeiT-Small [41]	22.1M	79.1 ± 0.3	88.2 ± 0.6	74.2 ± 0.2	46.1 ± 0.2	**71.9**

Table 3 In the VLCS dataset, the performance of PatchMix in three different backbones, ResNet-50, ViT and DeiT-Small.

			VLCS				
Model	Backbone	Params	Caltech101	LableMe	SUN09	VOC2007	Average
ERM [32]	ResNet-50 [56]	25.6M	97.6 ± 1.0	63.3 ± 0.9	72.2 ± 0.5	76.4 ± 1.5	77.4
PatchMix	ResNet-50 [56]	25.6M	97.9 ± 0.6	65.6 ± 0.8	74.2 ± 0.9	77.5 ± 0.4	**78.8**
ERM [32]	ViT [4]	86.6M	97.1 ± 0.4	64.9 ± 0.6	74.3 ± 0.7	77.8 ± 1.2	78.5
PatchMix	ViT [4]	86.6M	97.8 ± 0.4	65.3 ± 0.8	75.1 ± 0.3	77.6 ± 1.1	**79.0**
ERM [32]	DeiT-Small [41]	22.1M	96.7 ± 0.8	65.2 ± 1.0	73.9 ± 0.3	77.4 ± 0.6	78.3
PatchMix	DeiT-Small [41]	22.1M	97.4 ± 0.6	66.3 ± 1.0	74.7 ± 0.4	78.1 ± 1.2	**79.1**

experiments on four common datasets: VLCS, PACS, OfficeHome, and DomainNet. These datasets encompass a wide range of domains and classes, making them suitable for evaluating the performance of our method. After presenting the experimental results, we delve into a thorough analysis of each factor that could potentially influence the model's performance. Through the investigation of various aspects, our goal is to gain deeper insights into the effectiveness of our approach and its generalization capabilities.

4.1 Datasets and Setup

Our proposed method was comprehensively evaluated on four benchmark datasets commonly used in DG research: VLCS [58], PACS [59], Office-Home [60], and DomainNet [61]. The VLCS dataset comprises four domains: Caltech101 (1,415 images), LabelMe (2,656 images), SUN09 (3,282 images), and VOC2007 (3,376 images), collectively containing 10,729 images classified into 5 classes. The PACS dataset consists of four domains: Art Painting (2,048 images), Cartoon (2,344 images), Photo (1,670 images), and Sketch (3,929 images). It contains a total of 9,991 images classified into

Table 4 Comparison of PatchMix with different data augmentation methods and image stitching methods on the PACS dataset with ResNet-50, ViT, and DeiT-Small as the backbone.

			PACS				
Model	Backbone	Params	Art	Cartoon	Photo	Sketch	Average
ERM [32]	ResNet-50 [56]	25.6M	88.1 ± 0.1	77.9 ± 1.3	97.8 ± 0.0	79.1 ± 0.9	85.7
JiGen [6]	ResNet-50 [56]	25.6M	85.6 ± 0.2	78.1 ± 0.8	96.8 ± 0.3	77.6 ± 0.8	84.5
CutMix [5]	ResNet-50 [56]	25.6M	85.7 ± 0.3	76.9 ± 0.7	96.2 ± 0.2	77.2 ± 1.1	84.0
CutOut [57]	ResNet-50 [56]	25.6M	84.4 ± 0.2	77.1 ± 0.6	96.2 ± 0.6	76.9 ± 0.7	83.7
MixUp [28]	ResNet-50 [56]	25.6M	86.5 ± 0.3	76.6 ± 1.5	97.9 ± 0.2	76.5 ± 1.2	84.4
MixStyle [27]	ResNet-50 [56]	25.6M	86.9 ± 0.3	77.8 ± 0.3	98.1 ± 0.2	77.1 ± 0.5	85.0
PatchMix	ResNet-50 [56]	25.6M	87.5 ± 0.1	78.9 ± 0.3	98.3 ± 0.6	79.8 ± 0.8	**86.1**
ERM [32]	ViT [4]	88.6M	89.1 ± 0.2	83.4 ± 0.8	99.5 ± 0.1	62.8 ± 1.1	83.7
JiGen [6]	ViT [4]	88.6M	90.7 ± 0.6	83.3 ± 0.3	99.0 ± 0.4	60.1 ± 0.9	83.3
CutMix [5]	ViT [4]	88.6M	89.1 ± 0.1	81.5 ± 0.4	99.2 ± 0.6	57.9 ± 0.8	81.9
CutOut [57]	ViT [4]	88.6M	91.8 ± 0.4	81.7 ± 0.3	98.8 ± 0.5	62.7 ± 0.9	83.8
MixUp [28]	ViT [4]	88.6M	90.5 ± 0.2	81.5 ± 0.3	99.8 ± 0.1	64.1 ± 0.8	84.0
MixStyle [27]	ViT [4]	88.6M	90.4 ± 0.1	81.8 ± 0.3	98.1 ± 0.6	65.1 ± 0.8	83.9
PatchMix	ViT [4]	88.6M	89.7 ± 0.4	80.9 ± 0.6	99.8 ± 0.1	67.4 ± 1.3	**84.5**
ERM [32]	DeiT-Small [41]	22.1M	89.3 ± 0.2	82.4 ± 0.3	98.9 ± 0.6	78.0 ± 1.2	87.2
JiGen [6]	DeiT-Small [41]	22.1M	88.9 ± 0.3	82.9 ± 0.3	98.6 ± 0.4	77.9 ± 0.6	87.1
CutMix [5]	DeiT-Small [41]	22.1M	88.4 ± 0.3	82.6 ± 0.3	98.9 ± 0.5	77.5 ± 1.6	86.9
CutOut [57]	DeiT-Small [41]	22.1M	89.9 ± 0.6	83.4 ± 0.3	99.0 ± 0.4	78.4 ± 0.8	87.7
MixUp [28]	DeiT-Small [41]	22.1M	86.3 ± 0.3	83.6 ± 0.3	98.6 ± 0.2	76.8 ± 1.1	86.3
MixStyle [27]	DeiT-Small [41]	22.1M	88.5 ± 0.2	83.1 ± 0.3	98.5 ± 0.4	77.8 ± 0.9	87.0
PatchMix	DeiT-Small [41]	22.1M	90.6 ± 0.6	83.4 ± 0.2	98.9 ± 0.3	79.8 ± 1.6	**88.2**

Table 5 Comparison of PatchMix results with ERM on ResNet-50, ViT and DeiT-Small backbones on the OfficeHome dataset.

			OfficeHome				
Model	Backbone	Params	Art	Clipart	Product	Real World	Average
ERM [32]	ResNet-50 [56]	25.6M	62.7 ± 1.1	53.4 ± 0.6	76.5 ± 0.4	77.3 ± 0.3	67.5
PatchMix	ResNet-50 [56]	25.6M	63.5 ± 1.0	54.7 ± 0.4	78.3 ± 0.1	79.1 ± 0.6	**68.9**
ERM [32]	ViT [4]	88.6M	78.3 ± 0.8	61.0 ± 0.5	86.5 ± 0.4	88.5 ± 0.2	78.6
PatchMix	ViT [4]	88.6M	80.1 ± 0.7	63.5 ± 0.2	88.7 ± 0.5	90.0 ± 0.7	**80.6**
ERM [32]	DeiT-Small [41]	22.1M	67.6 ± 0.7	57.0 ± 0.4	79.4 ± 0.3	81.6 ± 0.6	71.4
PatchMix	DeiT-Small [41]	22.1M	71.9 ± 0.4	58.9 ± 0.2	81.9 ± 0.6	84.3 ± 0.3	**74.3**

7 classes. The Office-Home dataset contains 15,588 images of 65 classes for object recognition in office and home environments. It is organized into four domains: Art (2,427 images), Clipart (4,365 images), Product (4,439 images), and Real World (4,357 images). The DomainNet dataset is more extensive, with six domains: Clipart (48,129 images), Infograph (51,605 images), Painting (72,266 images), Quickdraw (172,500 images), Real (172,947 images), and Sketch (69,128 images). It comprises 345 classes and a total of 586,575 images.

Table 6 Comparison of PatchMix results with ERM on ResNet-50, ViT and DeiT-Small backbones on the DomainNet dataset.

			DomainNet						
Model	Backbone	Params	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Average
ERM [32]	ResNet-50 [56]	25.6M	58.4 ± 0.3	19.2 ± 0.4	46.3 ± 0.5	12.8 ± 0.0	60.6 ± 0.5	49.7 ± 0.8	41.2
PatchMix	ResNet-50 [56]	25.6M	59.3 ± 0.6	19.6 ± 0.3	46.8 ± 0.7	13.2 ± 0.1	62.2 ± 0.3	51.6 ± 0.4	**42.1**
ERM [32]	ViT [4]	88.6M	60.9 ± 0.3	27.2 ± 0.1	52.4 ± 0.4	15.8 ± 0.2	65.7 ± 0.1	53.9 ± 0.6	46.0
PatchMix	ViT [4]	88.6M	61.9 ± 0.4	27.0 ± 0.2	53.6 ± 0.6	16.1 ± 0.3	66.2 ± 0.3	54.6 ± 0.5	**46.6**
ERM [32]	DeiT-Small [41]	22.1M	62.9 ± 0.2	23.3 ± 0.1	53.1 ± 0.4	15.7 ± 0.6	65.7 ± 0.3	52.4 ± 0.2	45.5
PatchMix	DeiT-Small [41]	22.1M	62.7 ± 0.2	25.4 ± 0.6	53.6 ± 0.3	15.4 ± 0.4	66.1 ± 0.1	53.2 ± 0.5	**46.1**

4.2 Implementation Details

To ensure fair comparisons, we adopt the training and evaluation protocol from DomainBed [8]. Specifically, we select one domain as the test domain and use the remaining domains as source domains to train the model. We use Top-1 classification accuracy as the performance metric and average all results over three runs with different random seeds. For all ViT-based methods, including our proposed approach, we use AdamW [62] as the optimizer and the default hyperparameters of ERM from DomainBed. These hyperparameters include a weight decay of 5e-04, a learning rate of 1e-05, and a batch size of 32. To ensure the generalizability of our proposed approach, we report results using three different backbones: ResNet-50 [56] (25.6 million parameters), ViT(vit_base_patch16_224) [63] (88.6 million parameters) and DeiT-Small [64] (22.1 million parameters). By using multiple backbones, we demonstrate the effectiveness of our approach across different model architectures.

4.3 Comparison with the SOTA

As evident from Table 2, across the four datasets (VLCS, PACS, OfficeHome, and DomainNet) and utilizing ResNet-50, ViT, and DeiT-Small as backbone models, PatchMix achieves results of 69.0%, 72.7%, and 71.9%, respectively. Notably, these results are 1%, 1%, and 1.3% higher than those obtained by ERM.

VLCS

As shown in Table 3, our approach yielded favorable outcomes in the Caltech101, LabelMe, SUN09, and VOC2007 domains of the VLCS dataset. Specifically, when ResNet-50 serves as the backbone, PatchMix achieves a 1.4% improvement over ERM. Similarly, with ViT as the backbone, PatchMix outperforms ERM by 1.5%, and with DeiT-Small as the backbone, PatchMix surpasses ERM by 0.8%.

PACS

To better showcase the exceptional performance of PatchMix, this paper conducts comparison experiments using ResNet-50, ViT, and DeiT-Small as backbones, along with three common data augmentation methods, namely CutMix, CutOut, and MixUp. Additionally, two other methods, JiGen (jigsaw puzzle method) and MixStyle (fusion style), are included for a total of five methods. As demonstrated in Table 4, our approach yields significant improvements across all domains of PACS. Specifically, in the comparison experiment with ResNet-50 as the backbone, PatchMix achieves an

average accuracy of 86.1% over the four PACS domains, surpassing ERM by 0.4%. With ViT as the backbone, PatchMix outperforms ERM by 0.8% and MixUp methods by 0.5%. Furthermore, in the experiment using DeiT-Small as the backbone, PatchMix exhibits an improvement of 1.0% over ERM methods.

OfficeHome

Additionally, we conducted experiments on the OfficeHome dataset, and the results presented in Table 5 showcase the exceptional performance of our method across all four domains. Specifically, PatchMix outperforms ERM by 68.9%, 80.6%, and 74.3% in the models utilizing ResNet-50, ViT, and DeiT-Small as the backbone, respectively. These outstanding results confirm the robust generalizability of our method across all the aforementioned domains.

DomainNet

As presented in Table 6, our method demonstrates impressive performance on the DomainNet dataset across the six domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. In comparison to ERM, PatchMix exhibits substantial superiority, outperforming ERM by 42.1%, 46.6%, and 46.1% when tested with ResNet-50, ViT, and DeiT-Small as the backbone, respectively.

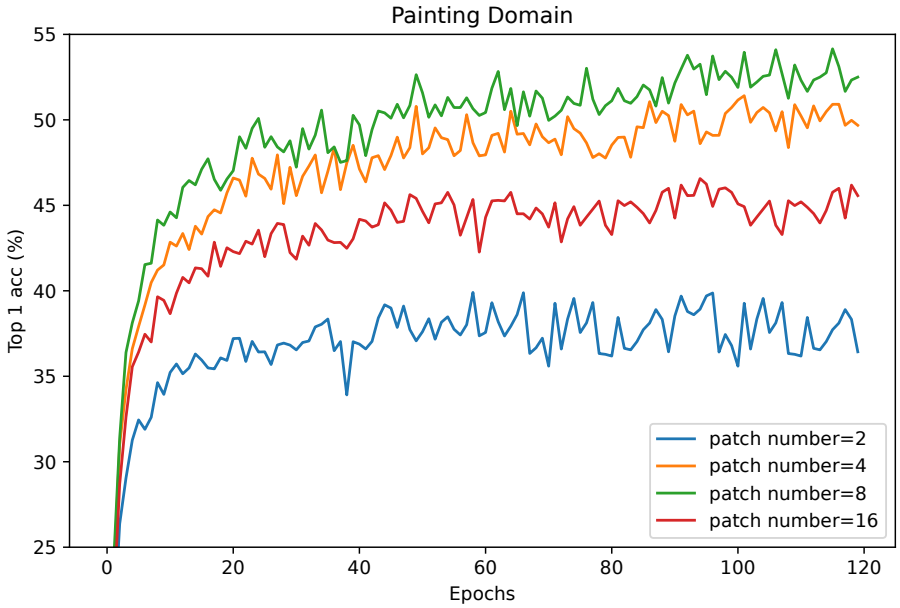

Fig. 4 Impact of a different number of patches on system performance in Painting Domain of DomainNet dataset.

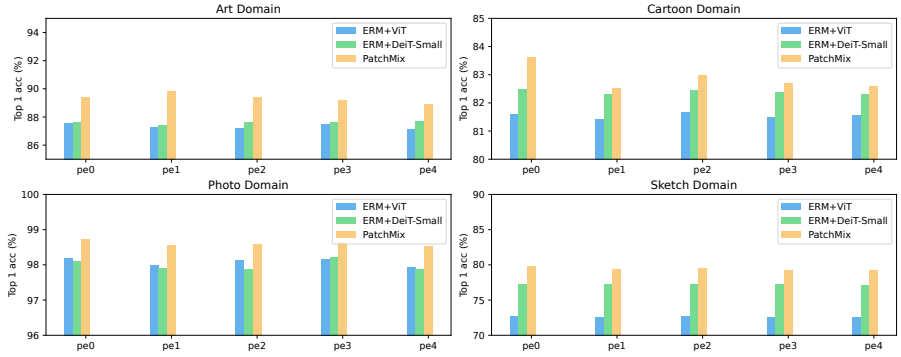

Fig. 5 Comparison of pe_0 to pe_4 pairs of PatchMix performance on 4 domains of PACS dataset.

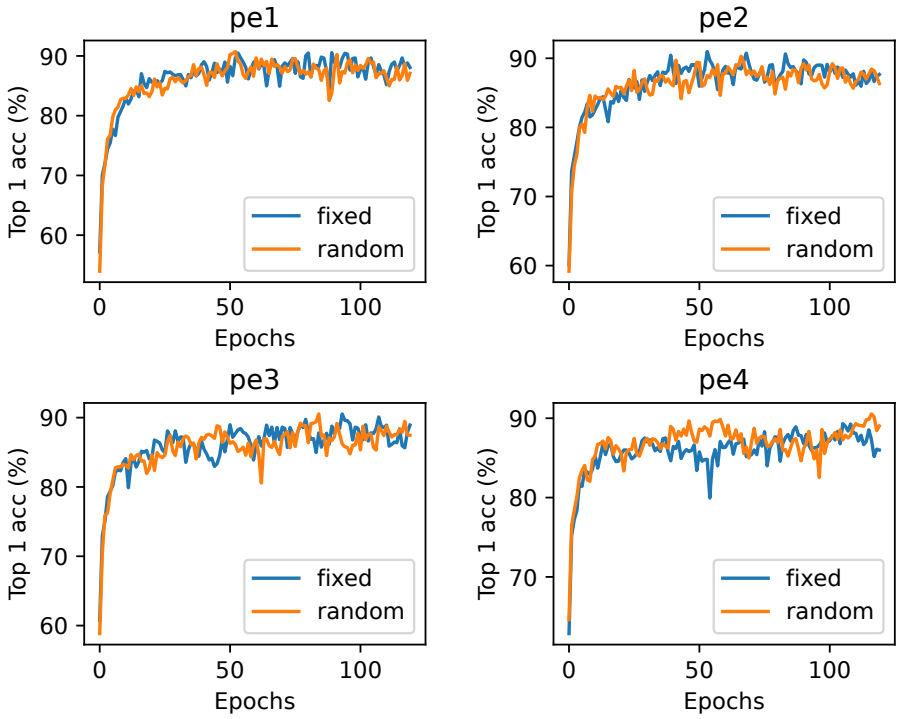

Fig. 6 Comparing the effects of patch transformations in fixed position and random position.

4.4 Ablation Study and Analysis

Patch Number Analysis

This paper achieves favorable results by dividing images into 4 patches, as evident from Table 2. To provide a clearer understanding of the impact of the number of patches on accuracy, we conducted experiments on the dataset comprising six domains of DomainNet: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. In these experiments, we varied the number of patches per image, exploring values of 4, 8, and

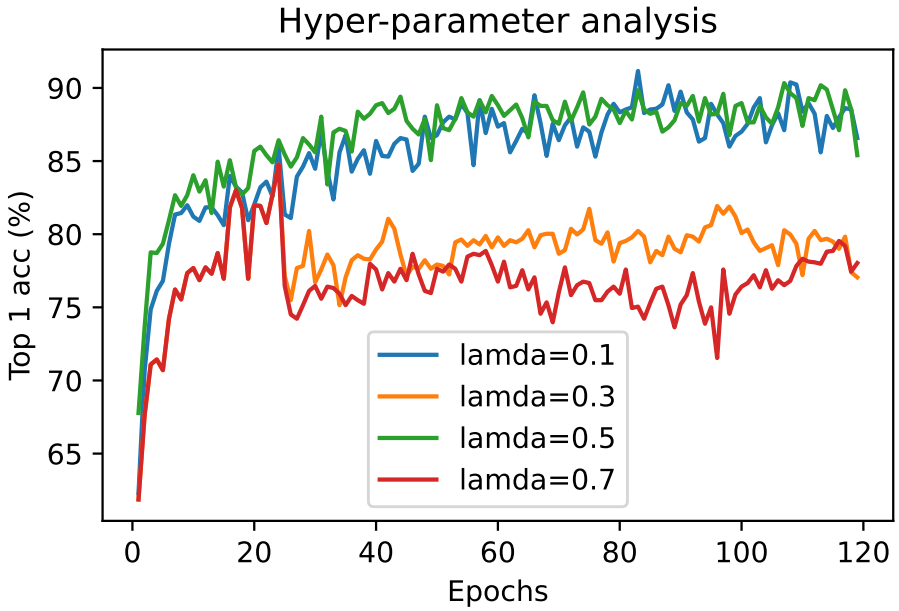

Fig. 7 Effect of different Hyper-parameter values λ on model performance.

16. The comparison revealed that cutting images into 8 patches yielded better results compared to cutting them into 4 or 16 patches, as shown in Figure 4. Consequently, based on our analysis of the VLCS, PACS, OfficeHome, and DomainNet datasets, we conclude that the relationship between the number of patches and the number of domains can be defined as follows:

$$\Phi = \min\{2^m, 2^m \geq M, 1 \leq m \leq M\}, \tag{20}$$

where M is the total number of domains, and Φ is the number of patchs. Based on Equation (20), we set the value of the patch number to 4 for the four domains of VLCS, PACS, and OfficeHome. In contrast, for the six domains of DomainNet, we set the value of the patch number to 8.

Patch e_0 Effectiveness Analysis

To assess the impact of using an image-complete micrograph e_0 as a patch on the performance of PatchMix, we denote $\{pe_i, 0 \leq i \leq n\}$ as the count of each image containing e_0. As shown in Figure 5, we compare the performance values (accuracy) of the Art, Cartoon, Photo, and Sketch domains in PACS for ERM + ViT, ERM + DeiT-Small, and PatchMix from pe_0 to pe_4 respectively. It is evident that at pe_0, all four domains exhibit the highest performance, as the stitched graph at this stage closely resembles the original graph. The performance of pe_3 is relatively closer to pe_0, while the performance of pe_4 is comparatively lower. This trend can be attributed to the fact that as the e_0 number increases, the difference between the stitched image and the test image becomes relatively large, leading to a decrease in recognition accuracy.

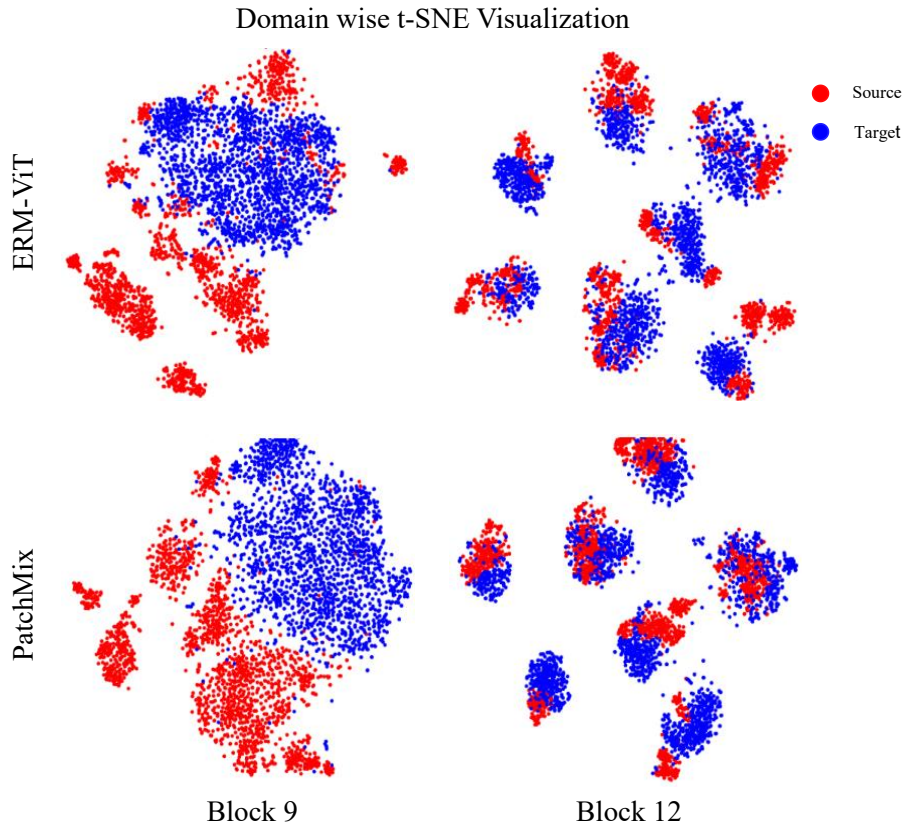

Fig. 8 Domain-wise t-SNE visualization of features from different blocks (9 & 12) in ERM-ViT and PatchMix which approach for Art domain in PACS dataset.

Table 7 Analyze the impact of different components of PatchMix on the OfficeHome dataset.

Domain Soft Label	Domain-mixed Sample	Acc
		71.4
	✓	73.5
✓	✓	74.2

To further validate the effectiveness of PatchMix, we conduct additional experiments on the PACS datasets. In this set of experiments, we directly take the 224 × 224 images from each of the three domains and stitch them together to create a larger 672 × 224 image. Subsequently, we modify the DeiT-Small network to accept input dimensions of [672, 224], but unfortunately, the model fails to converge during pre-training. we observe that although the model exhibits stronger oscillations, it fails to achieve convergence, with the accuracy only reaching 23.73% after 60 epochs, which is a substantial gap from the maximum accuracy of 88.2%.

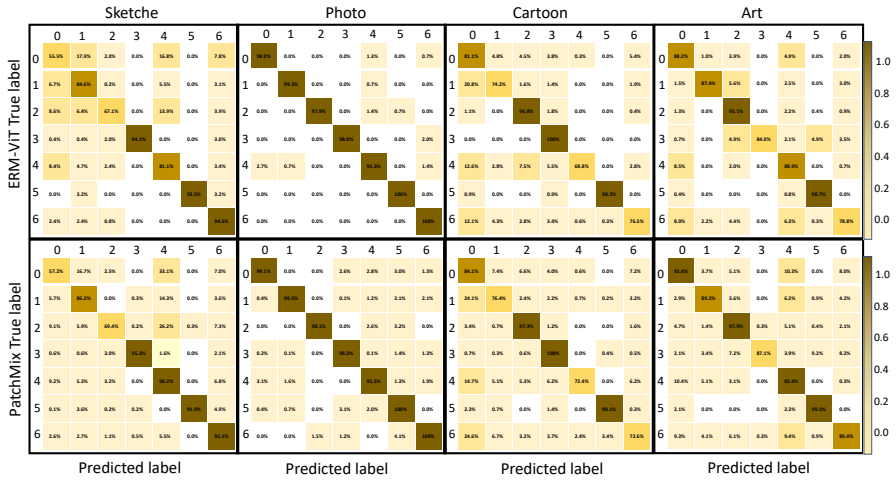

Fig. 9 Below are the confusion matrices for both the baseline method and PatchMix on the PACS dataset. The labels for each class are represented as follows: '0' for Dog, '1' for Elephant, '2' for Giraffe, '3' for Guitar, '4' for Horse, '5' for House, and '6' for Person.

Patch Position Analysis

To investigate the potential impact of patch positioning on accuracy, we conduct tests using the Art domain from the PACS dataset. Specifically, we evaluate the performance of pe_1 to pe_4 at both fixed and random locations. The results, depicted in Figure 6, demonstrate that the model exhibits similar convergence and accuracy for the same patch settings, regardless of whether they are positioned at fixed or random locations. This finding verifies that the position of the image within the patch does not significantly influence the model's performance.

Hyper-parameter Analysis

The overall loss of the model consists of two components: $\mathcal{L}_{ce}$ and $\mathcal{L}_{kl}$. To determine a suitable hyperparameter λ. we conducted experiments on the PACS dataset, and the results are presented in Figure 7. Remarkably, when setting the hyperparameter λ to 0.1, the model shows significantly improved convergence compared to other values.

Domain Soft Label Analysis

To evaluate the influence of domain soft labels on the model, we performed experiments on the OfficeHome dataset, using DeiT-Small as the backbone architecture, as indicated in Table 7. The results are summarized below: baseline (ERM) 71.4%, domain-mixed sample only 73.5%, domain soft labels + domain-mixed sample up to 74.2%. The results clearly demonstrate that domain soft labels continue to be beneficial in enhancing the model's performance, as they lead to an improvement of up to 74.2% compared to the domain-mixed sample only accuracy of 73.5%.

Feature Visualizations

Figure 8 illustrates the t-SNE display results for domain-wise comparisons across various blocks. Notably, our proposed approach, PatchMix, enables a significant increase in the overlap between the features of the source and target domains.

Confusion Matrices

To offer a more intuitive comparison between the classification results and the ground truth values, we employed a confusion matrix to assess the accuracy of our classification approach. Notably, when tested on the PACS dataset, our method exhibited a reduction in false positives, signifying its superior performance compared to the baseline, which utilized ViT. Figure 9 presents the confusion matrix, further highlighting PatchMix's efficacy in generating fewer false positives.

5 Conclusion

This paper proposes PatchMix, a novel domain generalization technique that uses patch and mixup to enable the neural network to learn global data characteristics effectively. The method also introduces a domain discriminator as a regularization term to improve model generalization. Experiments on four benchmark datasets show that PatchMix outperforms current SOTA methods and presents a new equation for the relationship between the number of patches and the number of domains. Overall, PatchMix offers a simple and effective solution to domain generalization in machine learning.

6 Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (No.2042023kf1033).

References

[1] Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Domain generalization with mixstyle. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021

[2] Moreno-Torres, J.G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern recognition **45**(1), 521–530 (2012)

[3] Wu, K., Li, L., Han, Y.: Weighted progressive alignment for multi-source domain adaptation. Multimedia Systems **29**(1), 117–128 (2023)

[4] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In:

9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021

[5] Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6023–6032 (2019)

[6] Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., Tommasi, T.: Domain generalization by solving jigsaw puzzles. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2229–2238 (2019)

[7] Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W., Yu, P.: Generalizing to unseen domains: A survey on domain generalization. IEEE Transactions on Knowledge and Data Engineering (2022)

[8] Gulrajani, I., Lopez-Paz, D.: In search of lost domain generalization. arXiv preprint arXiv:2007.01434 (2020)

[9] Zunino, A., Bargal, S.A., Volpi, R., Sameki, M., Zhang, J., Sclaroff, S., Murino, V., Saenko, K.: Explainable deep classification models for domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3233–3242 (2021)

[10] Chen, Y., Wang, Y., Pan, Y., Yao, T., Tian, X., Mei, T.: A style and semantic memory mechanism for domain generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9164–9173 (2021)

[11] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research **17**(1), 2096–2030 (2016)

[12] Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., Tao, D.: Deep domain generalization via conditional invariant adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 624–639 (2018)

[13] Gong, R., Li, W., Chen, Y., Gool, L.V.: Dlow: Domain flow for adaptation and generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2477–2486 (2019)

[14] Sicilia, A., Zhao, X., Hwang, S.J.: Domain adversarial neural networks for domain generalization: When it works and how to improve. arXiv preprint arXiv:2102.03924 (2021)

[15] Balaji, Y., Sankaranarayanan, S., Chellappa, R.: Metareg: Towards domain generalization using meta-regularization. Advances in neural information processing systems **31** (2018)

[16] Li, D., Yang, Y., Song, Y.-Z., Hospedales, T.: Learning to generalize: Meta-learning for domain generalization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

[17] Chen, K., Zhuang, D., Chang, J.M.: Discriminative adversarial domain generalization with meta-learning based cross-domain validation. Neurocomputing **467**, 418–426 (2022)

[18] Jeon, S., Hong, K., Lee, P., Lee, J., Byun, H.: Feature stylization and domain-aware contrastive learning for domain generalization. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 22–31 (2021)

[19] Kim, D., Yoo, Y., Park, S., Kim, J., Lee, J.: Selfreg: Self-supervised contrastive regularization for domain generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9619–9628 (2021)

[20] Peng, X., Huang, Z., Sun, X., Saenko, K.: Domain agnostic learning with disentangled representations. In: International Conference on Machine Learning, pp. 5102–5112 (2019). PMLR

[21] Zhang, H., Zhang, Y.-F., Liu, W., Weller, A., Schölkopf, B., Xing, E.P.: Towards principled disentanglement for domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8024–8034 (2022)

[22] Huang, J., Guan, D., Xiao, A., Lu, S.: Fsdr: Frequency space domain randomization for domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6891–6902 (2021)

[23] Qiao, F., Zhao, L., Peng, X.: Learning to learn single domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12556–12565 (2020)

[24] Liu, A.H., Liu, Y.-C., Yeh, Y.-Y., Wang, Y.-C.F.: A unified feature disentangler for multi-domain image translation and manipulation. Advances in neural information processing systems **31** (2018)

[25] Zhou, K., Yang, Y., Hospedales, T., Xiang, T.: Learning to generate novel domains for domain generalization. In: European Conference on Computer Vision, pp. 561–578 (2020). Springer

[26] Zhao, Y., Zhong, Z., Yang, F., Luo, Z., Lin, Y., Li, S., Sebe, N.: 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)

[27] Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Mixstyle neural networks for domain

generalization and adaptation. arXiv preprint arXiv:2107.02053 (2021)

[28] Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)

[29] Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A., Keutzer, K., Gong, B.: 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)

[30] Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5715–5725 (2017)

[31] Ghifary, M., Balduzzi, D., Kleijn, W.B., Zhang, M.: Scatter component analysis: A unified framework for domain adaptation and domain generalization. IEEE transactions on pattern analysis and machine intelligence **39**(7), 1414–1430 (2016)

[32] Vapnik, V.: The Nature of Statistical Learning Theory. Springer

[33] Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: International Conference on Machine Learning, pp. 10–18 (2013)

[34] Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14, pp. 443–450 (2016). Springer

[35] Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019)

[36] Sagawa, S., Koh, P.W., Hashimoto, T.B., Liang, P.: Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. arXiv preprint arXiv:1911.08731 (2019)

[37] Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., Courville, A.: Out-of-distribution generalization via risk extrapolation (rex). In: International Conference on Machine Learning, pp. 5815–5826 (2021). PMLR

[38] Cha, J., Chun, S., Lee, K., Cho, H.-C., Park, S., Lee, Y., Park, S.: Swad: Domain generalization by seeking flat minima. Advances in Neural Information Processing Systems **34** (2021)

[39] Iwasawa, Y., Matsuo, Y.: Test-time classifier adjustment module for model-agnostic domain generalization. Advances in Neural Information Processing Systems **34**, 2427–2440 (2021)

[40] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. (2017)

[41] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event

[42] Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 22–31 (2021)

[43] Sultana, M., Naseer, M., Khan, M.H., Khan, S., Khan, F.S.: Self-distilled vision transformer for domain generalization. In: Proceedings of the Asian Conference on Computer Vision, pp. 3068–3085 (2022)

[44] Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. arXiv preprint arXiv:2101.11986 (2021)

[45] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)

[46] Zhang, C., Zhang, M., Zhang, S., Jin, D., Zhou, Q., Cai, Z., Zhao, H., Yi, S., Liu, X., Liu, Z.: Delving deep into the generalization of vision transformers under distribution shifts. arXiv preprint arXiv:2106.07617 (2021)

[47] Harris, E., Marcu, A., Painter, M., Niranjan, M., Prügel-Bennett, A., Hare, J.: Fmix: Enhancing mixed sample data augmentation. arXiv preprint arXiv:2002.12047 (2020)

[48] Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781 (2019)

[49] Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501 (2018)

[50] Lim, S., Kim, I., Kim, T., Kim, C., Kim, S.: Fast autoaugment. Advances in Neural Information Processing Systems **32** (2019)

[51] Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 702–703 (2020)

[52] Ye, H., Xie, C., Cai, T., Li, R., Li, Z., Wang, L.: Towards a theoretical framework of out-of-distribution generalization. Advances in Neural Information Processing Systems **34**, 23519–23531 (2021)

[53] Li, D., Gouk, H., Hospedales, T.: Finding lost dg: Explaining domain generalization via model complexity. arXiv preprint arXiv:2202.00563 (2022)

[54] Deshmukh, A.A., Lei, Y., Sharma, S., Dogan, U., Cutler, J.W., Scott, C.: A generalization error bound for multi-class domain generalization. arXiv preprint arXiv:1905.10392 (2019)

[55] Zhang, Y., Liu, T., Long, M., Jordan, M.: Bridging theory and algorithm for domain adaptation. In: International Conference on Machine Learning, pp. 7404–7413 (2019). PMLR

[56] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[57] DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)

[58] Fang, C., Xu, Y., Rockmore, D.N.: 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)

[59] Li, D., Yang, Y., Song, Y.-Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5542–5550 (2017)

[60] Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5018–5027 (2017)

[61] Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1406–1415 (2019)

[62] Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam (2017)

[63] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is

worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

[64] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention, pp. 10347–10357 (2021)