Cross-Domain Feature Augmentation for Domain Generalization

Yingnan Liu[1,2] , **Yingtian Zou**[1,2] , **Rui Qiao**[1] , **Fusheng Liu**[2] , **Mong Li Lee**[1,2] and **Wynne Hsu**[1,2]

[1]School of Computing, National University of Singapore
[2]Institute of Data Science, National University of Singapore
{liu.yingnan, yingtian, rui.qiao, fusheng}@u.nus.edu, {leeml, whsu}@comp.nus.edu.sg

Abstract

Domain generalization aims to develop models that are robust to distribution shifts. Existing methods focus on learning invariance across domains to enhance model robustness, and data augmentation has been widely used to learn invariant predictors, with most methods performing augmentation in the input space. However, augmentation in the input space has limited diversity whereas in the feature space is more versatile and has shown promising results. Nonetheless, feature semantics is seldom considered and existing feature augmentation methods suffer from a limited variety of augmented features. We decompose features into class-generic, class-specific, domain-generic, and domain-specific components. We propose a cross-domain feature augmentation method named XDomainMix that enables us to increase sample diversity while emphasizing the learning of invariant representations to achieve domain generalization. Experiments on widely used benchmark datasets demonstrate that our proposed method is able to achieve state-of-the-art performance. Quantitative analysis indicates that our feature augmentation approach facilitates the learning of effective models that are invariant across different domains.

1 Introduction

Deep learning methods typically assume that training and testing data are independent and identically distributed. However, this assumption is often violated in the real world, leading to a decrease in model performance when faced with a different distribution [Torralba and Efros, 2011]. The field of domain generalization aims to mitigate this issue by learning a model from one or more distinct yet related training domains, with the goal of generalizing effectively to domains that have not been previously encountered. Studies suggest that the poor generalization on unseen distributions can be attributed to the failure of learning the *invariance* across different domains during the training phase [Muandet *et al.*, 2013; Li *et al.*, 2018]. To tackle this, research has focused on representation learning and data augmentation as key to learning invariance.
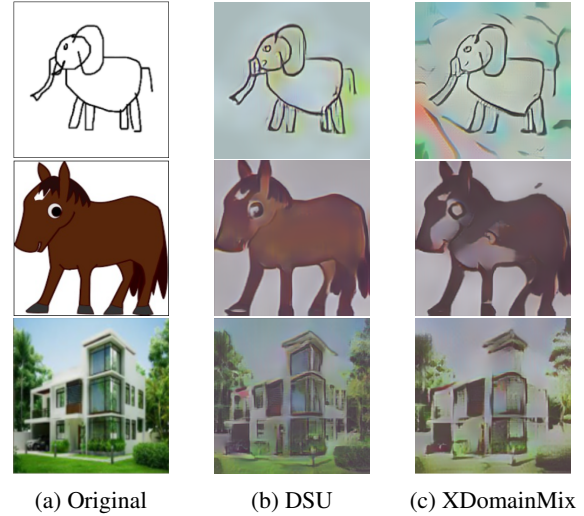

Figure 1: Samples of images reconstructed from features produced by DSU [Li *et al.*, 2022] and the proposed XDomainMix. The elephant reconstructed from XDomainMix's features shows a more complex background. The horse reconstructed from XDomainMix's features displays different characteristics such as a white belly. The top floor of the house generated by XDomainMix's features shows solid walls, instead of glass walls in the original. In contrast, images reconstructed from DSU's features have limited diversity and appear largely similar to the original images.

Invariant representation learning aims to align representation across domains [Shi *et al.*, 2022], and learn invariant causal predictors [Arjovsky *et al.*, 2019]. They usually impose regularization, which may result in a hard optimization problem [Yao *et al.*, 2022a]. In contrast, data augmentation techniques propose to generate additional samples for the learning of invariance, and avoid the complexities in the regularization approach [Mancini *et al.*, 2020]. Data augmentation can be generally classified into two types: input space and feature space augmentation. The former often encounters limitations due to a lack of diversity in the augmented data [Li *et al.*, 2021b], while the latter offers more versatility and has yielded promising outcomes [Zhou *et al.*, 2021].

Despite the versatility of feature space augmentation, existing methods such as MixStyle [Zhou *et al.*, 2021] and DSU [Li *et al.*, 2022] do not consider feature semantics during the

augmentation process. Instead, they alter feature statistics which often leads to a limited range of diversity. This lack of diversity in the generated features motivates us to decompose the features according to feature semantics. We build on prior research which suggests that the features learned for each class can be viewed as a combination of class-specific and class-generic components [Chu *et al.*, 2020]. The class-specific component carries information unique to a class, while the class-generic component carries information that is shared across classes. We observe that, even within the same class, features of samples from different domains can be distinguished, indicating that these features may contain domain-specific information. As such, we broaden our understanding of features to include domain-specific and domain-generic components.

We introduce a method called XDomainMix that changes domain-specific components of a feature while preserving class-specific components. With this, the model is able to learn features that are not tied to specific domains, allowing it to make predictions based on features that are invariant across domains. Figure 1 shows samples of original images and reconstructed images based on existing feature augmentation technique (DSU) and the proposed XDomainMix. We visualize the augmented features using a pre-trained autoencoder [Huang and Belongie, 2017]. From the reconstructed images, we see that DSU's augmented features remain largely the same as that of the original image feature. On the other hand, the images reconstructed from the features obtained using XDomainMix have richer variety while preserving the salient features of the class.

Results of experiments on benchmark datasets show the superiority of XDomainMix for domain generalization. We quantitatively measure the invariance of learned representation and prediction to show that the models trained with XDomainMix's features are more invariant across domains compared to state-of-the-art feature augmentation methods. Our measurement of the divergence between original features and augmented features shows that XDomainMix results in more diverse augmentation.

2 Related Work

To learn invariance, existing domain generalization approaches can be categorized into representation learning methods and data augmentation methods. Works on learning invariant representation employ regularizers to align representations or gradients [Sun and Saenko, 2016; Li *et al.*, 2020; Kim *et al.*, 2021; Mahajan *et al.*, 2021; Shi *et al.*, 2022; Rame *et al.*, 2022; Yao *et al.*, 2022b] across different domains, enforce the optimal classifier on top of the representation space to be the same across all domains [Arjovsky *et al.*, 2019; Ahuja *et al.*, 2021], or uses distributionally robust optimization [Sagawa *et al.*, 2020]. However, the use of regularization terms during learning of invariant representation could make the learning process more complex and potentially limit the model's expressive power.

Another approach is to employ data augmentation to learn invariance. Existing work that operates in the input space includes network-learned transformation [Zhou *et al.*, 2020;

Li *et al.*, 2021a], adversarial data augmentation [Volpi *et al.*, 2018; Shankar *et al.*, 2018], mixup [Mancini *et al.*, 2020; Yao *et al.*, 2022a], and Fourier-based transformation [Xu *et al.*, 2021]. Each of these techniques manipulates the input data in different ways to create variations that help the model learn invariant features. However, the range of transformations that can be applied in the input space is often limited.

On the other hand, feature augmentation can offer more flexibility and potential for learning more effective invariant representations. Prior work has generated diverse distributions in the feature space by changing feature statistics [Zhou *et al.*, 2021; Jeon *et al.*, 2021; Wang *et al.*, 2022; Li *et al.*, 2022; Fan *et al.*, 2023], adding noise [Li *et al.*, 2021b], or mixing up features from different domains [Mancini *et al.*, 2020; Qiao and Peng, 2021]. For example, MixStyle [Zhou *et al.*, 2021] synthesizes new domains by mixing the feature statistics of two features. DSU [Li *et al.*, 2022] extends the idea by modeling feature statistics as a probability distribution and using new feature statistics drawn from the distribution to augment features. In addition to generating diverse distributions, RSC [Huang *et al.*, 2020] adopts a different approach by discarding the most activated features instead of generating diverse data. This encourages the model to use less-activated features that might be associated with labels relevant to data outside the domain.

In contrast to existing methods, our work carefully considers feature semantics by leveraging class-label information and domain information to augment features. This increases intra-class variability and helps the model to learn a broader understanding of each class, thus improving its ability to handle new, unseen data.

3 Proposed Method

We consider the problem where we have a set of source domains $\mathcal{D}_S = \{S_1, \cdots, S_N\}$, $N > 1$ and a set of unseen domains $\mathcal{D}_U$. Each source domain $S_i = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{n_i}$ has a joint distribution on the input x and the label y. Domains in $\mathcal{D}_U$ have distinct joint distributions from those of the domains in $\mathcal{D}_S$. We assume that all domains in D_S and D_U share the same label space but the class distribution across domains need not be the same. The goal is to learn a mapping $g : x \rightarrow y$ using the source domains in $\mathcal{D}_S$ such that the error is minimized when g is applied to samples in $\mathcal{D}_U$.

In deep learning, g is typically realized as a composition of two functions: a feature extractor $f : x \rightarrow Z$ that maps input x to Z in the latent feature space, followed by a classifier $c : Z \rightarrow y$ that maps Z to the output label y. Ideally, f should extract features that are domain invariant yet retain class-specific information. The features of a given input can be decomposed into two components: class-specific and class-generic. The class-specific component consists of feature semantics that are strongly correlated with class labels, making them more important in discriminating a target class from other classes.

Furthermore, features can also be decomposed into domain-specific and domain-generic components. This is because samples from different domains, even if they belong to the same class, possess unique feature characteristics to their
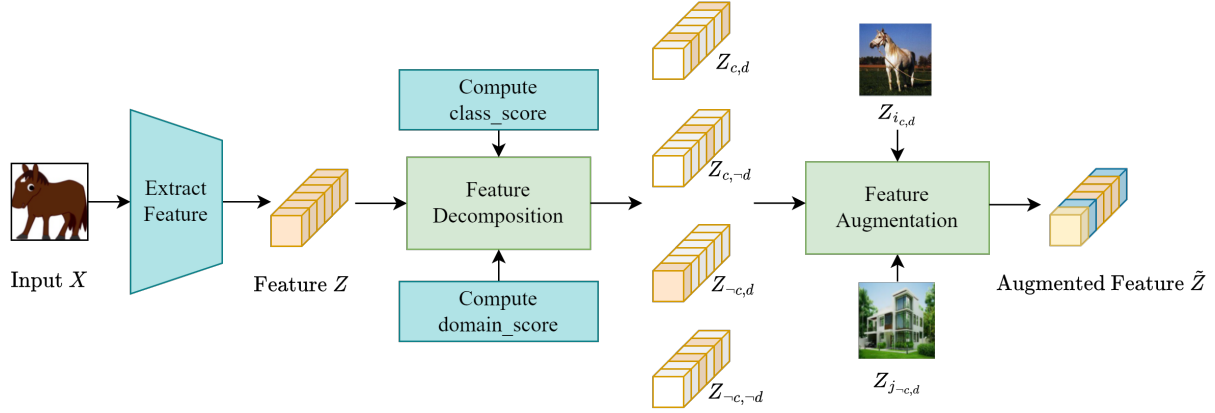
Figure 2: Overview of XDomainMix. To perform augmentation, the feature of an input is decomposed into four components based on the semantics' correlation with class and domain. Afterward, features of other two samples from different domains, one from the same class and one from a different class are used to augment features by changing domain-specific feature components.

respective domains. We extend these notions to decompose the features extracted by f into four distinct components: class-specific domain-specific ($Z_{c,d}$), class-specific domain-generic ($Z_{c,\neg d}$), class-generic domain-specific ($Z_{\neg c,d}$), and class-generic domain-generic ($Z_{\neg c,\neg d}$) component.

We determine whether an extracted feature contains information that is specific to a class or a domain by its importance to the prediction of the class and domain respectively. In other words, if a feature is important to the prediction of a specific class and a specific domain, it is considered a class-specific domain-specific component. If a feature is crucial in class prediction but not domain prediction, it falls into the class-specific domain-generic category. Similarly, features that are important for domain but not class predictions are categorized as class-generic domain-specific, and those not significant to either are labeled as class-generic domain-generic.

Our proposed feature augmentation strategy, XDomain-Mix, performs augmentation in the feature space by modifying the domain-specific component of features in a way that preserves class-related information. To discourage the use of domain-specific features for class prediction and encourage the exploitation of less activated features, class-specific domain-specific component is discarded with some probability during training. Details of feature decomposition and augmentation are described in the following subsections. Figure 2 shows an overview of our proposed method.

3.1 Feature Decomposition

Suppose the feature extractor f extracts $Z = f(x) \in \mathbb{R}^K$. Let z^k be the kth dimension of Z. We determine whether z^k is class-specific or class-generic via the class importance score, which is computed as the product of feature value and the derivative of class classifier c's predicted logit v_c of x's ground truth class [Selvaraju et al., 2017; Chu et al., 2020] as they show how much z^k contributes to v_c:

$$\text{class\_score}(z^k) = \frac{\partial v_c}{\partial z^k} z^k \qquad (1)$$

To determine if z^k is domain-specific, we employ a domain classifier d that has an identical architecture as the class classifier c. d is trained to predict domain labels of features extracted by the feature extractor. Similar to Equation 1, domain importance score is computed using the derivative of d's predicted logit v_d of x's ground truth domain:

$$\text{domain\_score}(z^k) = \frac{\partial v_d}{\partial z^k} z^k \qquad (2)$$

Let τ_c and τ_d be predefined thresholds for filtering feature dimensions that are class-specific and domain-specific respectively. We obtain a class-specific mask $M_c \in \mathbb{R}^K$ and a domain-specific mask $M_d \in \mathbb{R}^K$ on Z for $\{z^k\}_{k=1}^K$ where their respective k^{th} entries are given as follows:

$$M_c[k] = \begin{cases} 1 & \text{if class\_score}(z^k) > \tau_c \\ 0 & \text{otherwise} \end{cases},$$
$$M_d[k] = \begin{cases} 1 & \text{if domain\_score}(z^k) > \tau_d \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Complementary class-generic mask and domain-generic mask are obtained by $\mathbb{1} - M_c$ and $\mathbb{1} - M_d$ where $\mathbb{1}$ is the tensor of values 1 and of the same size as Z. Class-specific domain-specific ($Z_{c,d}$), class-specific domain-generic ($Z_{c,\neg d}$), class-generic domain-specific ($Z_{\neg c,d}$), and class-generic domain-generic ($Z_{\neg c,\neg d}$) feature components are obtained by

$$\begin{aligned} Z_{c,d} &= M_c \odot M_d \odot Z \\ Z_{c,\neg d} &= M_c \odot (\mathbb{1} - M_d) \odot Z \\ Z_{\neg c,d} &= (\mathbb{1} - M_c) \odot M_d \odot Z \\ Z_{\neg c,\neg d} &= (\mathbb{1} - M_c) \odot (\mathbb{1} - M_d) \odot Z \end{aligned} \qquad (4)$$

where $\odot$ is element-wise multiplication. Note that $Z_{c,d} + Z_{c,\neg d} + Z_{\neg c,d} + Z_{\neg c,\neg d} = Z$.

3.2 Cross Domain Feature Augmentation

To achieve domain invariance and reduce reliance on domain-specific information presented in training domains during

prediction, we manipulate domain-specific feature components to enhance diversity from a domain perspective. Further, the augmentation should increase feature diversity while preserving class semantics using existing data. This is achieved by mixing the class-specific domain-specific feature component of a sample with the class-specific domain-specific feature component of a same-class sample from other domains. For class-generic domain-specific feature component, it is mixed with the class-generic domain-specific feature component of a different-class sample from other domains, introducing further diversity.

Specifically, for the feature Z extracted from input x, we randomly sample two inputs x_i and x_j whose domains are different from x. Further, x_i has the same class label as x while x_j is from a different class. Let Z_i be the feature extracted from input x_i and Z_j be the feature extracted from x_j. Then we have

$$\tilde{Z}_{c,d} = \lambda_1 Z_{c,d} + (1 - \lambda_1)Z_{ic,d},$$
$$\tilde{Z}_{\neg c,d} = \lambda_2 Z_{\neg c,d} + (1 - \lambda_2)Z_{j \neg c,d} \tag{5}$$

where λ_1 and λ_2 are the mixup ratios independently sampled from a uniform distribution $U(0,1)$.

A new feature $\tilde{Z}$ with the same class label as Z is generated by replacing the domain-specific component in Z as follows:

$$\tilde{Z} = \tilde{Z}_{c,d} + \tilde{Z}_{\neg c,d} + Z_{c,\neg d} + Z_{\neg c, \neg d} \tag{6}$$

To further encourage the model to focus on domain-invariant features and exploit the less activated feature during class prediction, we discard the class-specific domain-specific feature component with some probability p_{discard} as follows:

$$\tilde{Z} = \begin{cases} \tilde{Z}_{\neg c,d} + Z_{c,\neg d} + Z_{\neg c,\neg d} & \text{if } p \leq p_{\text{discard}} \\ \tilde{Z}_{c,d} + \tilde{Z}_{\neg c,d} + Z_{c,\neg d} + Z_{\neg c,\neg d} & \text{otherwise} \end{cases} \tag{7}$$

where p is randomly sampled from a uniform distribution $U(0,1)$.

3.3 Training Procedure

Prior research has shown that empirical risk minimization (ERM) [Vapnik, 1999] is a competitive baseline [Gulrajani and Lopez-Paz, 2021; Wiles *et al.*, 2022]. The objective function of ERM is given by

$$\mathcal{L}_{erm} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(\hat{y}_j^{(i)}, y_j^{(i)}) \tag{8}$$

where ℓ is the loss function to measure the error between the predicted class $\hat{y}_j^{(i)}$ and the ground truth $y_j^{(i)}$. N is the number of training domains and n_i is the number of training samples in domain i.

We train the model in two phases. During the warm-up phase, the feature extractor f and class classifier c are trained on the original dataset for class label prediction following $\mathcal{L}_{erm}$. The domain classifier d is trained using Z, the features extracted by f, to predict domain labels. The objective function of d is given by

$$\mathcal{L}_d = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(d(Z_j^{(i)}), i) \tag{9}$$

where ℓ is the loss function to measure the error between the predicted domain $d(Z_j^{(i)})$ and the ground truth i.

When the warm-up phase is completed, we use Equation 4 to decompose the features obtained from f. Augmented features are then derived using Equation 7. Both feature extractor f and class classifier c are trained using the original and augmented features with the following objective function:

$$\mathcal{L}_{aug} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2n_i}\sum_{j=1}^{n_i}\left[\ell(c(Z_j^{(i)}), y_j^{(i)}) + \ell(c(\tilde{Z}_j^{(i)}), y_j^{(i)})\right] \tag{10}$$

where $\tilde{Z}_j^{(i)}$ is the augmented feature derived from $Z_j^{(i)}$. $c(Z_j^{(i)})$ is the predicted class given $Z_j^{(i)}$, and $c(\tilde{Z}_j^{(i)})$ is the predicted class given $\tilde{Z}_j^{(i)}$. $y_j^{(i)}$ is the ground truth class.

We also train the domain classifier d using $\mathcal{L}_d$. Note that the domain classifier d is not trained using this set of augmented features, as the augmented features do not have assigned domain labels since they need not follow the distribution of existing domains.

4 Performance Study

We implement our proposed solution using PyTorch 1.12.0 and perform a series of experiments on NVIDIA Tesla V100 GPU to evaluate the effectiveness of the proposed XDomainMix. The following benchmark datasets are used:

- Camelyon17 [Bandi *et al.*, 2018] from Wilds [Koh *et al.*, 2021]. This dataset contains 455,954 tumor and normal tissue slide images from 5 hospitals (domains). Distribution shift arises from variations in patient population, slide staining, and image acquisition.

- FMoW [Christie *et al.*, 2018] from Wilds. This dataset contains 141,696 satellite images from 62 land use categories across 16 years from 5 regions (domains).

- PACS [Li *et al.*, 2017]. This dataset contains 9,991 images of 7 objects in 4 visual styles (domains): art painting, cartoon, photo, and sketch.

- TerraIncognita [Beery *et al.*, 2018]. The dataset contains 24,788 images from 10 categories of wild animals taken from 4 different locations (domains).

- DomainNet [Peng *et al.*, 2019]. This dataset contains 586,575 images from 365 classes in 6 visual styles (domains): clipart, infograph, painting, quickdraw, real, and sketch.

The class importance thresholds τ_c and domain importance thresholds τ_d in Equation 3 are set as follows: τ_c is set to be the 50%-quantile of the class importance scores of $\{z^k\}$ in a feature Z so that 50% dimensions are considered class-specific, while the remaining 50% are class-generic. τ_d controls the strength of the augmentation as it determines the identification of domain-specific feature components. We employ a cyclic changing scheme for τ_d to let the model learn gradually from weak augmentation to strong augmentation and give the domain classifier more time to adapt to a more domain-invariant feature extractor. The value is initially set

Method	Camelyon17	FMoW	PACS	TerraIncognita	DomainNet
ERM	70.3±6.4	32.3±1.3	85.5±0.2	46.1±1.8	43.8±0.1
GroupDRO	68.4±7.3	30.8±0.8	84.4±0.8	43.2±1.1	33.3±0.2
RSC	77.0±4.9^	32.6±0.5^	85.2±0.9	46.6±1.0	38.9±0.5
MixStyle	62.6±6.3^	32.9±0.5^	85.2±0.3	44.0±0.7	34.0±0.1
DSU	69.6±6.3^	32.5±0.6^	85.5±0.6^	41.5±0.9^	42.6±0.2^
LISA	77.1±6.5	35.5±0.7	83.1±0.2^	47.2±1.1^	42.3±0.3^
Fish	74.7±7.1	34.6±0.2	85.5±0.3	45.1±1.3	42.7±0.2
XDomainMix	**80.9±3.2**	**35.9±0.8**	**86.4±0.4**	**48.2±1.3**	**44.0±0.2**

Table 1: Domain generalization performance of XDomainMix compared with state-of-the-art methods performance published in [Yao *et al.*, 2022a; Gulrajani and Lopez-Paz, 2021; Cha *et al.*, 2021]. Results with ^ are produced by us.

to be 90%-quantile of domain importance scores of $\{z^k\}$ in a feature Z. As the training proceeds, τ_d is decreased by 10% quantile for every n step until it reaches the 50%-quantile, where it also remains for n steps. The same cycle is repeated where τ_d is set to be 90%-quantile again. Input x_i, x_j used in augmentation (Equation 5) are samples from the same training batch. p_{discard} is set to 0.2.

For Camelyon17 and FMoW datasets, we follow the setup in LISA [Yao *et al.*, 2022a]. Non-pretrained DenseNet-121 is used for Canmelyon17 and pretrained DenseNet-121 is used for FMoW. We use the same partitioning in Wilds [Koh *et al.*, 2021] to obtain the training, validation, and test domains. The batch size is set to 32, and the model is trained for 2 epochs for Camelyon17 and 5 epochs for FMoW. The learning rate and weight decay are set to 1e-4 and 0. The warm-up phase is set to 4000 steps. We tune the step n in $\{100, 500\}$ for changing τ_d. The best model is selected based on its performance in the validation domain.

For PACS, TerraIncognita and DomainNet datasets, we follow the setup in DomainBed [Gulrajani and Lopez-Paz, 2021], and use a pre-trained ResNet-50. Each domain in the dataset is used as a test domain in turn, with the remaining domains serving as training domains. The batch size is set to 32 (24 for DomainNet), and the model is trained for 5000 steps (15000 steps for DomainNet). We tune the learning rate in $\{2e\text{-}5, 3e\text{-}5, 4e\text{-}5, 5e\text{-}5, 6e\text{-}5\}$ and weight decay in (1e-6, 1e-2) using the DomainBed framework. The warm-up phase is set to 3000 steps and n is set to 100 steps. The best model is selected based on its performance on the validations split of the training domains.

4.1 Domain Generalization Performance

We compare our proposed XDomainMix with ERM [Vapnik, 1999] and the following state-of-the-art methods:

- GroupDRO [Sagawa *et al.*, 2020] minimizes worst-case loss for distributionally robust optimization.
- RSC [Huang *et al.*, 2020] discards features that have higher activation to activate the remaining features appear to be applicable to out-of-domain data.
- MixStyle [Zhou *et al.*, 2021] synthesizes new domains by mixing feature statistics of two features.
- DSU [Li *et al.*, 2022] synthesizes new domains by re-normalizing feature statistics of features with the ones drawn from a probability distribution.

- LISA [Yao *et al.*, 2022a] selectively mixes up samples to learn an invariant predictor.
- Fish [Shi *et al.*, 2022] aligns gradients across domains by maximizing the gradient inner product.

Classification accuracy, which is the ratio of the number of correct predictions to the total number of samples is reported. Following the instruction of datasets, average accuracy on the test domain over 10 runs is reported for the Camelyon17 dataset; worst-group accuracy on the test domain over 3 runs is reported for the FMoW dataset. For PACS, TerraIncognita and DomainNet datasets, the averaged accuracy of the test domains over 3 runs is reported.

Table 1 shows the results. Our method consistently achieves the highest average accuracy across all the datasets, outperforming SOTA methods. This result suggests that XDomainMix is able to train models with good domain generalization ability.

4.2 Model Invariance

One advantage of XDomainMix is that is able to learn invariance across training domains. We quantify the invariance in terms of representation invariance and predictions invariance. Representation invariance refers to the disparity between representations of the same class across different domains. The distance between second-order statistics (covariances) [Sun and Saenko, 2016] can be used to measure representation invariance. Prediction invariance considers the variation in predictions across different domains. We employ risk variance [Yao *et al.*, 2022a] which measures how similar the model performs across domains.

Let $\{Z_j^{(i)}|y_j^{(i)} = y_c\}$ be the set of representations of class label y_c from domain i. We use $C_{y_c}^{(i)}$ to denote the covariance matrix of the representations. Given the set of class labels $\mathcal{Y}$ and the set of training domains $\mathcal{D}_S$, the measurement result is given by $\frac{1}{|\mathcal{Y}||\mathcal{D}_S|}\sum_{y_c\in\mathcal{Y}}\sum_{i,i'\in\mathcal{D}_S}||C_{y_c}^{(i)}-C_{y_c}^{(i')}||_F^2$. $||\cdot||_F^2$ denotes the squared matrix Frobenius norm. A smaller distance suggests that same-class representations across domains are more similar.

Let R^i be the loss in predicting the class labels of inputs from domain i. The risk variance is given by the variance among training domains, $\text{Var}\{R^1, R^2, ..., R^{|\mathcal{D}_S|}\}$. Lower risk variance suggests a more consistent model performance across domains.

Method	Camelyon17	FMoW	PACS	TerraIncognita	DomainNet
ERM	0.47±0.18	0.35±0.09	0.69±0.12	0.50±0.09	3.60±0.38
GroupDRO	0.21±0.02	0.40±0.04	0.77±0.16	0.43±0.05	2.13±0.09
RSC	0.32±0.17	0.55±0.14	42.3±12.8	29.9±3.02	17.3±1.43
MixStyle	0.28±0.26	0.36±0.05	0.69±0.03	0.38±0.09	3.39±0.17
DSU	**0.07±0.02**	0.32±0.02	0.33±0.04	9.18±1.84	4.61±0.24
LISA	0.19±0.05	0.29±0.05	0.04±0.00	0.13±0.01	0.72± 0.06
Fish	3.95±3.28	0.47±0.02	0.64±0.34	0.34±0.04	2.60±0.26
XDomainMix	0.19±0.07	**0.28±0.01**	**0.02±0.00**	**0.04±0.00**	**0.11±0.02**

(a) Representation invariance measured by distance of covariance matrix of same class representations across domains.

Method	Camelyon17	FMoW	PACS	TerraIncognita	DomainNet
ERM	1.55±0.27	139±50.0	9.89±2.09	12.7±2.6	553±26.5
GroupDRO	0.93±0.11	161±171	398±65.1	603±22.3	668±17.9
RSC	1.90±0.50	181±70.8	6.34±0.91	10.3±3.6	631±7.06
MixStyle	1.67±0.91	110±88.9	6.51±1.20	9.10±0.56	563±4.68
DSU	3.85±1.30	237±136	16.2±5.24	10.6±1.5	567±21.1
LISA	1.81±1.14	111±15.2	3.02±0.47	9.39±0.51	520±11.9
Fish	5.68±1.81	251±45.3	9.08±5.38	9.37±1.59	567±13.2
XDomainMix	**0.90±0.28**	**109±11.7**	**2.10±0.21**	**8.22±1.05**	**504±15.8**

(b) Prediction invariance measured by variance of risk across domains. The results are reported in the unit of 1e-3.

Table 2: Results of model invariance.

Method	Camelyon17	FMoW	PACS	TerraIncognita	DomainNet
MixStyle	6.65±0.18	8.58±0.50	3.11±0.34	3.20±0.26	2.81±0.11
DSU	26.77±2.53	20.93±1.87	6.97±0.03	10.85±0.32	5.75±0.01
XDomainMix	**38.82±0.28**	**34.06±0.11**	**14.82±0.11**	**14.36±0.12**	**10.27±0.02**

Table 3: Divergence of the augmented feature and original feature measured by MMD in the unit of 1e-2.

Table 2 shows the results. We see that our method has the smallest covariance distance in the FMoW, PACS, TerraIncognita, and DomainNet dataset, and the second-smallest in the Camelyon17 dataset. The results indicate that the representations of the same class learned by our method have the least divergence across domains. Additionally, XDomainMix has the lowest risk variance, suggesting that it is able to maintain consistent performance in predictions across domains. Overall, the results demonstrate that our approach is able to learn invariance at both the representation level and the prediction level.

4.3 Diversity of Augmented Features

To show that XDomainMix can generate more diverse features, we measure the distance between the original and augmented features using maximum mean discrepancy (MMD). A higher MMD suggests that the distance between the original and augmented features is further. The same set of original features is used to ensure fairness and comparability of the measurement result. Average and standard deviation over three runs are reported. Table 3 shows the results. Features augmented by XDomainMix consistently have the highest MMD compared to MixStyle and DSU which are two SOTA feature augmentation methods. This suggests that the features augmented by XDomainMix exhibit the most deviation from the original features, leading to a more varied augmentation. Visualization of sample images reconstructed from aug-

mented features are given in Supplementary.

4.4 Experiments on the Identified Features

In this set of experiments, we demonstrate that XDomainMix is able to identify features that are important for class and domain prediction. We evaluate the model performance for class or domain prediction after eliminating those features with the highest importance score computed in Equations 1 and 2. A decrease in accuracy suggests that the features that have been removed are important for the predictions.

For comparison, we implement two alternative selection strategies: a random method that arbitrarily selects features to remove, and a gradient norm approach, where features are chosen for removal based on the magnitude of the gradient in the importance score computation. Samples in the validation set of PACS dataset are used in this experiment.

Figure 3 shows the results. Our method shows the largest drop in both class prediction and domain prediction accuracies compared to random removal and gradient norm methods. This indicates that XDomainMix is able to identify features that are specific to the domain and class effectively.

To visualize the extracted features, we map the high dimensional feature vectors obtained by f to a lower dimensional space, This transformation is carried out using two linear layers, as described in [Zhang *et al.*, 2021]. Figure 4(a) provides a visualization for the model that is trained on the PACS dataset, with Art as the unseen domain. We see that
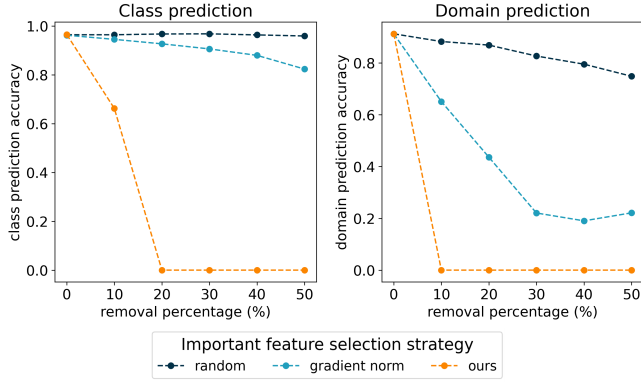
Figure 3: Prediction accuracy after removing x% of features with the highest importance scores.

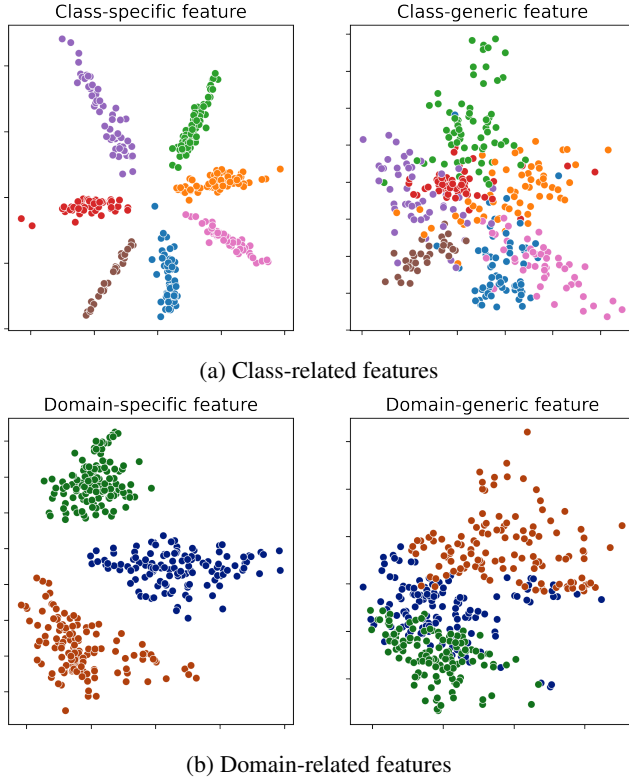(a) Class-related features

(b) Domain-related features

Figure 4: Visualization of features from different classes/domains, indicated by the different colors.

the features identified as class-specific are well separated by class. This is in contrast to the features that are generic across classes, which are not as clearly delineated.

Similarly, we also visualize the extracted domain-specific and domain-invariant features. As shown in Figure 4(b), the domain-specific features are noticeably better separated compared to the domain-invariant features.

4.5 Ablation Study

To understand the contribution of each component in the augmentation, we perform ablation studies on Camelyon17 and

FMoW datasets. Table 4 shows the result. Compared to the baseline, mixing class-specific domain-specific feature components ($Z_{c,d}$) only, or mixing class-generic domain-specific feature components ($Z_{\neg c,d}$) only in the augmentation can improve the performance. This suggests that by manipulating domain-specific feature components, models that are better at domain generalization can be learned. Mixing $Z_{\neg c,d}$ leads to greater improvement, indicating that enriching diversity by content from other classes is more helpful than simply intra-class augmentation.

Augmenting both $Z_{c,d}$ and $Z_{\neg c,d}$ does not consistently lead to performance improvement, possibly due to dataset-specific characteristics. Probabilistically discarding $Z_{c,d}$ seems to encourage the model to use less domain-specific information and exploit less activated features in prediction, which improves the domain generalization performance.

mix $Z_{c,d}$	mix $Z_{\neg c,d}$	discard $Z_{c,d}$	Camelyon17	FMoW
			70.3±6.4	32.3±1.3
✓			78.3±5.5	32.9±2.2
	✓		79.1±6.0	33.6±1.1
✓	✓		79.6±7.0	31.9±0.4
✓	✓	✓	**80.9±3.2**	**35.9±0.8**

Table 4: Ablation study.

5 Conclusion and Future Work

In this work, we have developed a feature augmentation method to address the domain generalization problem. Our approach aims to enhance data diversity within the feature space for learning models that are invariant across domains by mixing domain-specific components of features from different domains while retaining class-related information. We have also probabilistically discarded domain-specific features to discourage the model from using such features for their predictions, thereby achieving good domain generalization performance. Our experiments on multiple datasets demonstrate the effectiveness of the proposed method.

While our method presents a promising approach to solving the domain generalization problem, there are several limitations. Our method needs more than one training domain to perform cross-domain feature augmentation. Our method assumes that the datasets across different domains share the same label space and similar class distributions, and its performance may be affected if this is not the case. Our method is mainly empirically validated, and a theoretical analysis or guarantee of its performance is still lacking. Further research is needed to provide a deeper theoretical understanding of the proposed method and its performance bounds.

Acknowledgements

References

[Ahuja *et al.*, 2021] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:3438–3450, 2021.

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

[Bandi *et al.*, 2018] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 2018.

[Beery *et al.*, 2018] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[Cha *et al.*, 2021] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

[Christie *et al.*, 2018] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[Chu *et al.*, 2020] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *European Conference on Computer Vision*, pages 694–710. Springer, 2020.

[Fan *et al.*, 2023] Qi Fan, Mattia Segu, Yu-Wing Tai, Fisher Yu, Chi-Keung Tang, Bernt Schiele, and Dengxin Dai. Towards robust object detection invariant to real-world domain shifts. In *The Eleventh International Conference on Learning Representations*, 2023.

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

[Huang and Belongie, 2017] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

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

[Jeon *et al.*, 2021] Seogkyu Jeon, Kibeom Hong, Pilhyeon Lee, Jewook Lee, and Hyeran Byun. Feature stylization and domain-aware contrastive learning for domain generalization. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 22–31, 2021.

[Kim *et al.*, 2021] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9619–9628, 2021.

[Koh *et al.*, 2021] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

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

[Li *et al.*, 2018] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, number 1, 2018.

[Li *et al.*, 2020] Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex Kot. Domain generalization for medical imaging classification with linear-dependency regularization. *Advances in neural information processing systems*, 33:3118–3129, 2020.

[Li *et al.*, 2021a] Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 224–233, 2021.

[Li *et al.*, 2021b] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8886–8895, 2021.

[Li *et al.*, 2022] Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and LINGYU DUAN. Uncertainty modeling for out-of-distribution generalization. In *International Conference on Learning Representations*, 2022.

[Mahajan *et al.*, 2021] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR, 2021.

[Mancini *et al.*, 2020] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *European Conference on Computer Vision*, pages 466–483. Springer, 2020.

[Muandet *et al.*, 2013] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

[Peng *et al.*, 2019] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[Qiao and Peng, 2021] Fengchun Qiao and Xi Peng. Uncertainty-guided model generalization to unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6790–6800, 2021.

[Rame *et al.*, 2022] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. PMLR, 2022.

[Sagawa *et al.*, 2020] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020.

[Selvaraju *et al.*, 2017] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[Shankar *et al.*, 2018] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018.

[Shi *et al.*, 2022] Yuge Shi, Jeffrey Seely, Philip Torr, Siddharth N, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, 2022.

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

[Torralba and Efros, 2011] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[Vapnik, 1999] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[Volpi *et al.*, 2018] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

[Wang *et al.*, 2022] Yue Wang, Lei Qi, Yinghuan Shi, and Yang Gao. Feature-based style randomization for domain generalization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[Wiles *et al.*, 2022] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dj Dvijotham, and Ali Taylan Cemgil. A fine-grained analysis on distribution shift. In *International Conference on Learning Representations*, 2022.

[Xu *et al.*, 2021] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14383–14392, 2021.

[Yao *et al.*, 2022a] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 25407–25437. PMLR, 17–23 Jul 2022.

[Yao *et al.*, 2022b] Xufeng Yao, Yang Bai, Xinyun Zhang, Yuechen Zhang, Qi Sun, Ran Chen, Ruiyu Li, and Bei Yu. Pcl: Proxy-based contrastive learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7097–7107, 2022.

[Zhang *et al.*, 2021] Yifan Zhang, Bryan Hooi, Dapeng Hu, Jian Liang, and Jiashi Feng. Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

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

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