Mode-Guided Feature Augmentation for Domain Generalization

Muhammad Haris Khan[1]
muhammad.haris@mbzuai.ac.ae

Talha Zaidi[2]
tzaidi@ksu.edu

Salman Khan[1]
salman.khan@mbzuai.ac.ae

Fahad Shehbaz Khan[1]
fahad.khan@mbzuai.ac.ae

[1] Computer Vision Department
Mohamed Bin Zayed University of Artificial Intelligence
Abu Dhabi, UAE

[2] Computer Science Department
Kansas State University
Kansas, USA

Abstract

This paper tackles domain generalization (DG) problem, the task of utilizing only source domain(s) to learn a model that generalizes well to unseen domains. A key challenge faced by DG is often the limited diversity in available source domain(s) that restricts the network's ability in learning a generalized model. Existing DG approaches leveraging data augmentation to address this problem mostly rely on compute-intensive auxiliary networks coupled with various losses and also suffer from additional training overhead. To this end, we propose a simple and efficient DG approach to augment source domain(s). We hypothesize the existence of favourable correlation between the source and target domain's major modes of variation, and upon exploring those modes in the source domain we can realize meaningful alterations to background, appearance, pose and texture of object classes. Inspired by this, our new DG approach performs feature-space augmentation by identifying the dominant modes of change in the source domain and implicitly including the augmented versions along those directions to achieve a better generalization across domains. Our method shows competitive performance against the current state-of-the-art methods on three popular DG benchmarks. Further, encouraging results on challenging single-source setting validate strong domain generalization capabilities of our approach.

1 Introduction

Recently, deep neural network (DNN) based approaches have exhibited remarkable performance in various computer vision tasks. Majority of these successes belong to the closed-world supervised learning paradigm, which assumes that both training and testing examples are drawn from the same distribution. However, in realistic settings, this assumption is often violated and the trained model could exhibit poor performance. An active line of research, known as supervised (or unsupervised) domain adaptation [2, 5, 12, 19, 28, 29, 41], alleviates this performance degradation by utilizing labelled (or unlabelled) target domain examples.

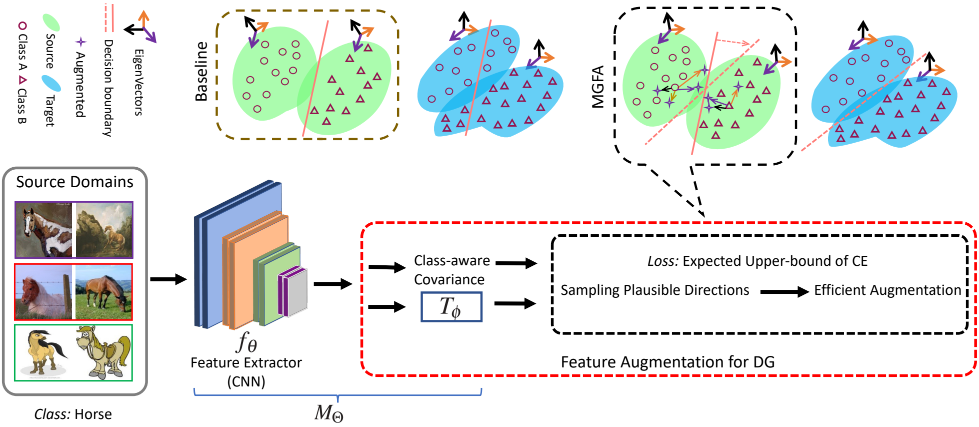

Figure 1: Top row: (baseline) There exist favourable similarity between the source and target domain's major modes of variation (shown as eigenvectors). (MGFA) Upon exploring those modes we can achieve meaningful alterations to background, appearance, and pose in object classes. With this motivation, our DG method (MGFA) performs feature-space augmentation by identifying these dominant modes in the source domain, and implicitly including the augmented versions along those directions to achieve a better generalization across domains. Bottom row: Overall architecture of MGFA. Fundamentally, it is an AGG [23] formulation, which aggregates data from all available source domains, and trains a deep CNN. We improve the diversity of source domains by a feature augmentation process for DG (red dotted box).

Despite being effective, these methods are restrictive as they demand pre-collecting and accessing the target domain data and require re-training for adaptation. In many real-world applications, the availability of target domain data is not guaranteed during training [26, 45], and it is required to generalize to new unseen domains. This is known as domain generalization (DG) problem [3, 13, 25, 26, 30, 33, 37, 48] whom setting is challenging, but rewarding due to minimal assumptions [16, 25].

Some methods address DG problem by increasing the diversity of source domains [4, 31, 37, 40, 42]. These methods aim to generate domain-guided novel examples and augment them with the source data. For instance, [48] learns a CNN generator using various losses and synthesizes novel examples [48]. However, the generator-based approaches are limited because it is a compute-intensive process since training and inferring generative models for augmentation are both non-trivial procedures. Some data augmentation approaches to DG [37, 40, 48] face additional training overhead (for the actual task) proportional to the number of augmented examples. Finally, some existing DG methods build on complex training schemes [1, 8, 25], few depend on non-trivial balancing of various constraints [8, 26], and quite a few require domain identifiers in addition to class labels [7, 27, 37, 48].

In this paper, we propose a new DG approach to augmenting source domains in a simple, effective and efficient manner. We hypothesize that there exist favourable correlation between the source and target domain's major modes of variation (Sec. 3.1). In terms of high-level intuition, our observation is also supported by the subspace alignment approaches [11, 14, 38, 39] in the domain adaptation (DA) literature. Fig. 1 (top row: baseline), visualizes this similarity in terms of orientation between their respective eigenvectors. We anticipate that upon exploring these major (principal) modes of variation in source domain(s), it is possible to achieve meaningful alterations to the background, appearance and texture of object classes without actually compromising the class information. To this end, we propose a domain generalization (DG) approach, dubbed as MGFA, that performs feature-space augmentation by identifying these dominant modes in the source domain(s), and implicitly including the augmented versions along those directions (Fig. 1 top row: MGFA) to achieve a better generalization across target domain.

Contributions. We identify exploitable correlation between the source and target domain's major modes of variation, and presume that their exploration allows meaningful changes to background, appearance, and texture of an object class which can be harnessed to achieve efficient augmentations. To actualize this, we propose a new domain generalization (DG) approach capable of increasing the diversity of source domain(s) by identifying the dominant modes of change in the source domain(s) and implicitly including the augmented versions along those directions. Our mode-guided augmentation process is simple and efficient, and as such imposes a minimal extra training overhead. Experimental results on three popular DG benchmarks: PACS [23], VLCS [10], and Digits-DG [12, 22, 22, 34] show the superiority of our approach against the existing state-of-the-art methods. Further, promising results in challenging single-source setting corroborate the strong DG capabilities of our approach.

2 Related Work

Some DG methods aim to learn a domain-invariant feature space assuming that there exists an underlying space shared by all source domains and the unseen target domain [3, 9, 13, 16, 26, 33]. Blanchard *et al.* [3] pulls all training data together to form one dataset, and learns a single SVM classifier. Muandet *et al.* [33] employed maximum mean discrepancy (MMD) constraint to minimise the discrepancy between all source domains. A few methods seek to improve model robustness using a low-rank parameterized CNN model [23], masking features via gradients [18], leveraging auxiliary tasks [6, 43], employing domain-specific masks [7], and incorporating domain-specific normalizations [36]. For instance, Chattopadhyay *et al.* [7] proposed domain-specific masks to balance between domain-invariant and domain-specific feature learning. Some DG methods leverage meta-learning framework to expose the model to domain shift during training. The source domains are divided into disjoint meta-train and meta-test sets, and a model is trained on the meta-train set such that it generalizes to the meta-test set [1, 8, 24, 25]. Li *et al.* [25] proposed to robustify a shared feature extractor and a classifier using domain-specific models in an episodic learning strategy. Meta-learning based DG methods are prone to overfitting since the available training data remains unchanged. Recently, Gulrajani *et al.* [16] showed that a carefully implemented empirical risk minimization (ERM) achieves state-of-the-art performance.

Data augmentation is an intuitive approach for improving the diversity of source domain(s). Cross-grad training [37] trains a label classifier and a domain classifier jointly by their respective perturbations. Likewise, Volpi *et al.* [42] imposed wasserstein constraint in the semantic space to generate adversarial samples from fictitious target distribution to be different at pixel level. Tobin *et al.* [40] generated a variety of samples from simulated environments via random renderings to localize the objects in real-world scenarios. Blanchard *et al.* [4] formulated a kernel-based approach which predicts classifiers from an augmented feature space. Recently, Zhou *et al.* [48] proposed to learn full CNN generator by employing various losses to generate new examples. These generated examples are then aggregated with the original source examples to train the actual task model. On the contrary, our mode-guided feature augmentation approach to DG identifies dominant modes of

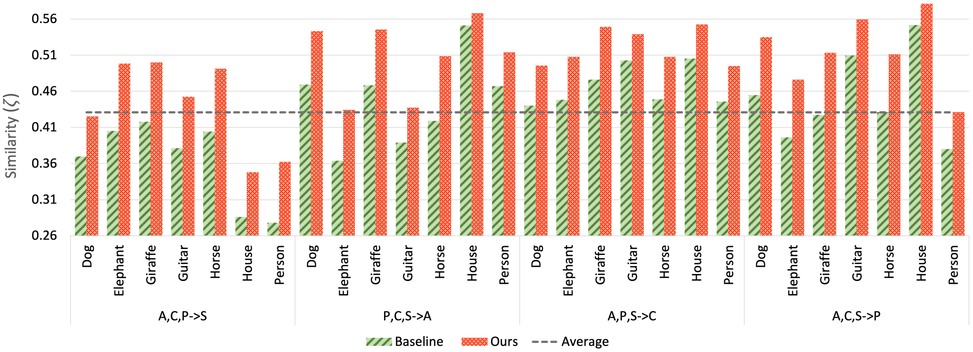

Figure 2: (Baseline) There exists reasonable similarity between the source domain(s) and target domain's principal modes of variation. Each bar denotes similarity between multi-source and target domain's subspaces for a certain semantic class. The average line is the mean of all similarity values in case of baseline. Our method further increases this similarity by performing implicit feature augmentation along these directions (in source data) in an efficient way. A,C,P, and S denote Arts, Cartoon, Photo, and Sketch domains, respectively, in PACS dataset [23]. Note, A,C,P→S denotes the domain shift scenario when source domains for training are Arts, Cartoon, and Photo and the target domain for testing is Sketch.

change in the source domain(s) and implicitly includes the augmented versions along those directions. CuMix [31] revisits the mixup technique [46], and performs image and feature interpolation by mixing randomly chosen intra-domain and inter-domain samples. Further, this mixing strategy gets increasingly complex during training in a curriculum fashion. In contrast, our method is not only fundamentally different but more effective and efficient as it enables sampling meaningful perturbed examples, guided by principal modes of variation, and allows implicitly augmenting these perturbed examples through a single loss function.

Compared to previous DG methods, our approach is simple and efficient as it avoids: training compute-intensive auxiliary networks, complex training procedures, non-trivial balancing of different losses, and paying additional training overhead proportional to the number of augmented versions. Further, it does not require domain identifiers, which might not be readily available.

3 Feature Augmentation for DG

3.1 Preliminaries

Problem Settings. We explore two DG settings: multi-source DG and single-source. In multi-source DG settings, we assume that we are provided with k source domains $D = \{D^1, ..., D^k\}$, where D^k is the k^{th} source domain comprising data-label pairs $(x_i^k \in \mathcal{X}, y_i^k \in \mathcal{Y})$. k and i denote domain index and example index, respectively. In single-source DG settings, the data is available from only one source domain. In this work, we consider the object recognition task, and the aim is to learn a mapping function $M_\Theta : \mathcal{X} \rightarrow \mathcal{Y}$ that accurately predicts examples from an unseen testing domain.

Motivation. We aim to estimate the characteristics of the unseen target domain by figuring out its commonalities with the source domain(s). Concretely, we identify if there exists any correlation between the source domain(s) and a target domain's principal modes of variation. Let us define two matrices, $A \in \mathbb{R}^{U \times R}$ and $B \in \mathbb{R}^{U \times R}$ representing two subspaces with vector columns $\{a_i\}_{i=1,R}$ and $\{b_i\}_{i=1,R}$ comprising set of principal modes selected after performing eigenvalue decomposition on source domain and target domain covariance matrices, respectively. Where U is the dimensionality of features from a CNN. We can compare these subspaces by defining the vector projection of each a_j onto the set of modes $\{b_i\}_{i=1,R}$, $p_j^{AB} = \sum_{i=1}^{R}(a_j.b_i)b_i$ [15]. The Gramian matrix $G(R \times R)$ of the set of vectors $\{p_i^{AB}\}_{i=1,R}$ can be computed as the matrix of inner products, whose entries are $G_{ij} = (p_i^{AB}.p_j^{AB})$. Now, the diagonalization of G, $L_G^T G L_G = \Lambda_G$, lets us utilize the eigenvalues of G, $\{\lambda_i\}_{i=1,R}$, as the measure of similarity between the two subspaces. The smallest angle between any pair of orthogonal modes of A and B is defined as $cos^{-1}\sqrt{\lambda_1}$, where λ_1 is the largest eigenvalue of G. Therefore, the sum of eigenvalues of G equals the sum of squares of the cosines of the angles between the two subspaces [21]. Since all the eigenvalues of G vary between 0 and 1, which correspond to critical angles between $0°$ and $90°$, the similarity measure between the two subspaces is defined as $\zeta^{AB} = \frac{\sum_{i=1}^{R}\lambda i}{R}$. If $\zeta^{AB} \approx 0$, the two subspaces are dissimialar, while $\zeta^{AB} \approx 1$ shows that the two subspaces share the same orientation. We use the above procedure for every pair of semantic class in a multi-source DG setting and plot the results in Fig. 2 both for baseline and MGFA. Our baseline collects the datasets of all available source domains, and trains a single deep CNN in an end-to-end manner [23]. We see the existence of favourable correlation between the source domain(s) and target domain's principal modes of variation and in comparison to baseline our mode-guided feature augmentations, described next in Sec. 3.2, further improves this correlation between source and target domains.

3.2 Operating in Feature-space

The deep feature space has been shown to encapsulate the human visual perception and provides enough structure to identify meaningful augmentation directions [35, 47]. As an example, we hypothesize that sampling along the right directions in feature space correspond to background and appearance changes without altering the underlying class. Further, exploration in the feature space is efficient compared to the input space as we will show in Sec. 3.2.2. The feature-space augmentation process is realized in two stages: 1) estimating the category-aware distribution in an online fashion in the feature space (Sec. 3.2.1), and 2) minimizing the upper bound of an expected classification loss function (Sec. 3.2.2). The first stage enables sampling meaningful augmented versions while the second stage allows efficiently augmenting these altered examples and avoiding training with large amounts of extra data. Fig. 1 (bottom row) displays an overall architecture of the proposed framework. Since our augmentation operates in the feature space, we decompose the model (M_Θ) into a feature extractor (f_θ) and a task network (T_ϕ), where Θ, θ, ϕ denote the parameters of the complete, feature, and task networks, respectively.

3.2.1 Searching Meaningful Augmentations

Random Sampling. It is possible to get augmented versions by randomly sampling in the deep feature space. This space is quite sparse and high-dimensional, for instance, ResNet-18 [17] produces a 512-dimensional feature vector on an image from the PACS dataset where regions corresponding to feasible inputs will most likely be sparsely distributed. Even if each dimension can take on 2 different values, we will still get 2^{512} feature values. Therefore, random sampling will mostly yield useless directions, and hence, uninformative augmented versions that can even confuse the model on real inputs.

Class-aware Sampling. Instead of random sampling, inspired by [44], we sample random vectors from a multivariate zero-mean normal distribution whose covariance is proportional to the intra-class covariance matrix of real class examples. This class-aware covariance matrix models the class-conditional distribution and can capture rich semantic information as it encodes the major modes of variations in a particular class. We note that in general, these modes of variations correspond to feasible changes in color, appearance, texture, pose, viewpoint and backgrounds (see Fig. 5). Let, the feature extractor f_θ generates a U dimensional feature vector u_i corresponding to an input example x_i i.e., $f(x_i,\theta) = u_i \in \mathbb{R}^U$. We drop the domain-label k from x_i as our method does not require domain identifiers. We generate augmented versions corresponding to u_i along the major class modes in two stages.

Stage-I. We setup a zero-mean multivariate normal distribution $\mathcal{N}(0,\Sigma_{y_i})$, where Σ_{y_i} is the class-conditional covariance matrix estimated from the deep features of all examples in class y_i. We compute these matrices online by taking into account the statistics of all mini-batches,

$$\mu_j^{(t)} = \frac{n_j^{(t-1)}\mu_j^{(t-1)} + m_j^{(t)}\acute{\mu}_j^{(t)}}{n_j^{(t-1)} + m_j^{(t)}}, \tag{1}$$

$$\Sigma_j^{(t)} = \frac{n_j^{(t-1)}\Sigma_j^{(t-1)} + m_j^{(t)}\acute{\Sigma}_j^{(t)}}{n_j^{(t-1)} + m_j^{(t)}} + \frac{n_j^{(t-1)}m_j^{(t)}(\mu_j^{(t-1)} - \acute{\mu}_j^{(t)})(\mu_j^{(t-1)} - \acute{\mu}_j^{(t)})^T}{(n_j^{(t-1)} + m_j^{(t)})^2} \tag{2}$$

where $\mu_j^{(t)}$ and $\Sigma_j^{(t)}$ are the mean values and covariance estimates of the features belonging to j^{th} class at t^{th} training iteration. $\acute{\mu}_j^{(t)}$ and $\acute{\Sigma}_j^{(t)}$ are the mean values and covariance estimates of the features in j^{th} class in t^{th} mini-batch. Further, $n_j^{(t)} = n_j^{(t-1)} + m_j^{(t)}$, where $n_j^{(t)}$ is the total number of training examples in the j^{th} class in all t mini-batches. $m_j^{(t)}$ denotes the number of training examples in the j^{th} class only in t^{th} mini-batch.

Stage-II. While the network is training, a covariance matrix is estimated for each class. To generate the augmented version $\hat{u}_i$, corresponding to u_i, u_i is translated along a random direction sampled from $\mathcal{N}(0,\gamma\Sigma_{y_i})$ as $\hat{u}_i \sim \mathcal{N}(u_i,\gamma\Sigma_{y_i})$, where γ is a non-negative scalar to regulate the strength of meaningful augmentation. Owing to the dynamic evolution of covariance matrices during training, the covariance estimation in the first few epochs might not be useful. To this end, the γ parameter is simply made a function of the current epoch t as, $\gamma = (\frac{t}{Q}) \times \gamma_0$, where Q is total number of training epochs.

3.2.2 Augmenting Meaningful Examples

Naive Approach. A straightforward approach to realize the data augmentation is to augment each u_i with V number of augmented versions, generating an aggregated feature set of size $V \times N$. So, the overall network M_Θ is trained by minimizing the Cross-Entropy (CE) loss:

$$\mathcal{L}_V(W,b,\Theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{V}\sum_{v=1}^{V} -\log\Big(\frac{e^{w_{y_i}^T u_i^v + b_{y_i}}}{\sum_{j=1}^{C} e^{w_j^T u_i^v + b_j}}\Big), \tag{3}$$

where $W = [w_1,...,w_C]^T \in \mathbb{R}^{C\times U}$ and $b = [b_1,...,b_C]^T \in \mathbb{R}^C$ denote the weight matrices and biases corresponding to the task network T_ϕ, respectively.

Efficient Approach. As expected, this straightforward approach will become compute-intensive bottleneck when V will grow large. To counter this, we assume that V approaches to infinity and resort to deriving the upper-bound of the loss function, which results in an efficient realization of the augmentation. Formally, when $V \to \infty$, we have the expectation of the CE loss under all possible augmented features:

$$\mathcal{L}_\infty(W,b,\Theta|\Sigma) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\hat{u}_i}[-\log(\frac{e^{w_{y_i}^T \hat{u}_i + b_{y_i}}}{\sum_{j=1}^{C} e^{w_j^T \hat{u}_i + b_j}})] \tag{4}$$

Assuming $\mathcal{L}_\infty$ can be computed efficiently, then we can directly minimize it without actually sampling the augmented features. Since Eq. 4 is difficult to compute in its exact form, we derive an upper bound for $\mathcal{L}_\infty$, as shown by the following proposition [44]:

Proposition 1 *Suppose that* $\hat{u}_i \sim \mathcal{N}(u_i, \gamma\Sigma_{y_i})$. *Then we have an upper bound of* $\mathcal{L}_\infty$, *given by:* $\mathcal{L}_\infty \leq \frac{1}{N}\sum_{i=1}^{N} -\log(\frac{e^{w_{y_i}^T u_i + b_{y_i}}}{\sum_{j=1}^{C} e^{w_j^T u_i + b_j + \frac{\gamma}{2} z_{jy_i}^T \Sigma_{y_i} z_{jy_i}}}) \triangleq \bar{\mathcal{L}}_\infty$,, *where* $z_{jy_i} = w_j - w_{y_i}$.

In essence, Proposition 1 is a surrogate loss for an efficient realization of the data augmentation. This avoids minimizing the exact loss function $\mathcal{L}_\infty$ and, instead, allows optimizing its upper bound $\bar{\mathcal{L}}_\infty$ in a more efficient manner. Finally, when $\gamma \to 0$, it amounts to no feature augmentation and $\mathcal{L}_\infty$ becomes the standard CE loss.

4 Experiments

Datasets. PACS [23] is a recent DG benchmark exhibiting severe domain discrepancy. It shares 7 object categories across 4 different domains: Photos (P), Arts (A), Cartoons (C) and Sketches (S). We follow the protocol in [23], including the train and validation splits, for fair comparison. **VLCS** [10] is a classic DG benchmark for the task of object recognition; it comprises five object categories from four different domains (PASCAL VOC 2007 (P), LabelMe (L), Caltech (C), and Sun (S) datasets). **Digits-DG** comprises four different digit datasets including MNIST (M) [22], MNIST-M (MM) [12], SVHN (S) [34] and SYN (Sy) [22], which differ drastically in font style, stroke color and background. We follow the protocol in [48], including the train and validation splits to draw fair comparison.

Network Architecture and Implementation Details. Our baseline model aggregates the datasets of all available source domains, and trains a single deep CNN in an end-to-end manner [23]. Our framework allows using different network architectures. In ResNet-18/50 [17], we take features after the avg.pooling layer to estimate the class-aware covariance matrices. In Alexnet [20], we use the FC7 layer output to approximate the same by their diagonals. We empirically set the hyper-parameter γ_0 from the set $\{0.5, 1.0, 1.5, 2.0\}$ according to the performance on the validation set. We implement our framework using PyTorch on a NVIDIA RTX 6000 GPU. For PACS and VLCS, we employ SGD optimizer and train for 100 epochs with a batch size of 128. The learning rate is set as 0.001 and decreased to 0.0001 after 80 epochs. We also adopt the same on-the-fly data augmentation as JiGen [6] to train the baseline and our method. For Digits-DG, following [48], we construct the CNN backbone with four 64-kernel 3×3 conv. layers and a softmax layer, and insert ReLU and 2×2 max-pooling after each conv. layer. It is trained using SGD with an initial learning rate of 0.05 and batch size of 126 for 100 epochs. The learning rate is decayed by 0.1 every 20 epochs. For model selection, we use exactly the same validation set in both PACS and VLCS datasets as used in [6, 18, 25] to achieve the model selection. Each training domain is split into training and validation subsets. Then, the validation subsets of each training domain are pooled to create an

PACS	P	A	C	S	Avg.
MLDG[22]	88.00	66.23	66.88	58.96	70.01
Epi-FCR[25]	86.1	64.7	72.3	65.0	72.0
Meta-Reg[1]	91.07	69.82	70.35	59.26	72.67
JiGen[6]	89.00	67.63	71.71	65.18	73.38
MASF[8]	90.68	70.35	72.46	67.33	75.21
RSC[18]	91.25	68.39	69.23	67.09	73.99
DMG[7]	87.31	64.65	69.88	71.42	73.32
AGG(Baseline)	89.62	66.12	69.40	61.13	71.56
Ours	90.31	69.13	70.36	72.05	75.46

PACS	P	A	C	S	Avg.
Meta-Reg[1]	95.50	83.70	77.20	70.30	81.70
CrossGrad[37]	96.0	79.8	76.8	70.2	80.7
MASF[8]	94.99	80.29	77.17	71.69	81.04
JiGen[6]	96.03	79.42	75.25	71.35	80.51
Epi-FCR[25]	93.9	82.1	77.0	73.0	81.5
DMG[7]	93.35	76.90	80.38	75.21	81.46
RSC[18]	94.63	81.34	75.02	75.01	81.50
CuMix[31]	95.1	82.3	76.5	72.6	81.6
AGG(Baseline)	96.00	78.67	73.93	70.59	79.79
Ours	95.40	81.70	77.61	76.02	82.68

PACS	Meta-Reg[1]	MASF[8]	RSC[18]	DMG[7]	AGG(Baseline)	Ours
P	97.60	95.01	96.82	94.49	97.80	97.86
A	87.20	82.89	87.10	82.57	85.49	86.40
C	79.20	80.49	79.67	78.11	75.56	79.45
S	70.30	72.29	79.85	78.32	75.36	78.72
Avg.	83.60	82.67	85.86	83.37	83.55	85.60

Table 1: Domain generalization results on PACS dataset with recognition accuracy (%) using AlexNet (top Left), ResNet-18 (top right) and ResNet-50 (bottom) backbones. Numbers in red and blue correspond to the best and the second-best performance, respectively.

overall validation set. Finally, the model maximizing the accuracy on the overall validation set is chosen. This is also called as model selection in DG using training domain validation set [16]. Similarly, for Digits-DG dataset, we use the same validation set as adopted in [48] to achieve the model selection.

Comparison with state-of-the-art. We show the effectiveness of our method (MGFA) on PACS with three backbone networks: AlexNet, ResNet-18, and ResNet-50 (Table 1). With AlexNet, MGFA provides a significant gain of 3.9% in terms of overall(average) accuracy over the baseline and outperforms all prior approaches, including DMG[7] and RSC[18]. Finally, MGFA achieves the best accuracy of 72.05% in the most severe domain shift of P,A,C → S shift. With ResNet-18, MGFA delivers the best overall(average) accuracy of 82.68%, and also surpasses all previous methods in most severe domain shift i.e. P,A,C → S. Finally, with ResNet-50, MGFA provides a gain of 2.05% in overall(average) accuracy over baseline, and achieves the best accuracy of 97.86% accuracy in A,C,S→P shift.

Table 2 reports results on VLCS using AlexNet architecture and Digits-DG datasets. In VLCS, MGFA provides an improvement of 1.62% over the baseline (AGG) and obtains a superior performance of 74.47% than the prior methods in overall(average) accuracy. Moreover, MGFA demonstrates the best accuracy for L,C,S → P and P,L,S → C shifts. In Digits-DG, L2A-OT[48] is the best performing method in overall accuracy, however, it relies on compute-intensive CNN generator (50M params.) with several losses and further incurs at least 100% extra computational (training) overhead over the baseline. In contrast, our method builds on a rather simple and efficient augmentation process and records the highest accuracy of 69.35% on SVHN as the target domain. It does not depend on any auxiliary network, and incurs only 0.2% additional computational (training) overhead over the baseline.

Single Source Domain Generalization. To further validate the effectiveness of our method, we perform DG experiments in single source settings. In this setting, we train a model on a single source domain and then test it on all other domains. This is a much harder DG setting than multi-source DG due to two obvious reasons: 1) given k source domains, the available training data is $1/k$ times, compared to k times in multi-source settings, 2) given $k+1$ domains in total, the testing data is k times, compared to $1/k$ times in multi-source settings. Table 3 reports single-source DG results on PACS, VLCS and Digits-DG datasets. In PACS, MGFA provides improved results over the baseline in all (four) shifts (around 1.2% gain)

VLCS	P	L	C	S	Avg.
MLDG[22]	67.7	61.3	94.4	65.9	72.3
Epi-FCR[25]	67.1	64.3	94.1	65.9	72.9
JiGen[6]	70.62	60.90	96.93	64.30	73.19
MASF[8]	69.14	64.90	94.78	67.64	74.11
RSC[18]	72.58	61.06	97.16	65.27	74.01
AGG(Baseline)	70.80	59.40	96.91	63.58	72.67
Ours	73.53	61.85	98.26	64.26	74.47

Digits-DG	M	MM	S	Sy	Avg.
CCSA[32]	95.2	58.2	65.5	79.1	74.5
MMD-AAE[26]	96.5	58.4	65.0	78.4	74.6
CrossGrad[37]	96.7	61.1	65.3	80.2	75.8
JiGen[6]	96.5	61.4	63.7	74.0	73.9
L2A-OT[48]	96.7	63.9	68.6	83.2	78.1
AGG(Baseline)	94.0	58.5	66.7	74.6	73.4
Ours	95.71	60.66	69.35	74.38	75.02

Table 2: Domain generalization results on VLCS dataset using AlexNet [17] architecture (Left) and on Digits-DG dataset (Right).

PACS	A,C,S	P,C,S	P,A,S	P,A,C	Avg.
Vanilla	37.94	61.67	73.96	23.7	49.31
JiGen[6]	35.43	60.20	70.57	41.64	51.96
RSC[18]	33.65	63.69	73.90	36.43	51.91
Ours	39.30	63.21	76.31	29.96	52.19

VLCS	L,C,S	V,C,S	V,L,S	V,L,C	Avg.
Vanilla	69.94	52.14	47.11	61.10	57.57
JiGen[6]	68.54	49.10	48.44	61.28	56.84
RSC[18]	70.20	59.12	45.25	61.93	59.12
Ours	70.52	57.95	48.97	62.88	60.08

Digits-DG	M,MM,S	Sy,MM,S	Sy,M,MM	Sy,M,MM	Avg.
Vanilla	60.30	32.13	50.94	50.3	48.41
Ours	66.69	36.40	51.16	51.69	51.48

Table 3: Single-source DG results on PACS using ResNet-18, VLCS using AlexNet, and Digits-DG.

and overall accuracy (3.07% gain). Further, it shows improved performance than JiGen[6] and RSC[18] in overall accuracy and P→A,C,S and C→P,A,S shifts. P→A,C,S translates to training on Photo domain and then testing on Arts, Cartoon, and Sketch domains. Finally, we observe a similar trend in VLCS and Digits-DG datasets. Based on these results, we can conclude that MGFA is capable of generalizing when the training data is from a single domain and scarce and upon testing there are multiple domain shifts.

Analysis of Our Method. We first observe that class-aware sampling provides a significant gain of 2.04% over random sampling in overall accuracy (Table 4). This reflects the importance of exploring meaningful, plausible directions in the feature space that are related and shared between the different domains. After replacing γ with a fixed γ_0, we observe that it is beneficial to keep the value of γ relatively small in the first few epochs and its contribution should increase as the training evolves (Table 4). This is because the covariance estimation may not be expressive in the early epochs. We show both theoretically[1] and empirically that MGFA incurs little extra computational overhead, up to two and three orders of magnitude smaller than the computation cost of the single-source baseline with ResNet-18 and ResNet-50 networks, during the training process (Table 5). Fig. 3 displays the variation in performance of our method when taking different values of γ_0. We see that the overall accuracy of the method is mostly robust in the range $0.25 \leq \gamma_0 \leq 1.5$, however, it starts to drop after $\gamma_0 > 1.5$. Fig. 4 shows the class-wise recognition accuracy (%) in target domains by MGFA as a function of available training images from the source classes. The average class-wise accuracy for the classes where the number of training images is around 600 is 84.04%, and the same accuracy for the classes where training images are over 900 is 84.26%. Our method retains performance even with relatively low number of class-wise examples for estimating class-aware covariance matrix. Fig. 5 visualizes augmented images (obtained via reconstruction) for nine different classes. We see that MGFA can alter the background, pose, appearance, and color while mostly retaining the class-level semantics.

Limitations. Failures can be expected while producing meaningful augmentations if an im-

[1] See supplementary material for a description on how the theoretical complexity is obtained.

age contains significant differences (e.g., in object's appearance) from most of the images in its class (Fig. 5). For instance, an image of a monkey Fig. 5 shows only his face, while majority of others in the class show full body. This could be because meaningful directions for such under-represented images are not well captured in the class-aware covariance matrix.

Method/Domain	P	A	C	S	Avg.
AGG(Baseline)	96.00	78.67	73.93	70.59	79.79
Random Sampling	95.86	78.61	73.80	74.30	80.64
Fixed γ_0	94.37	81.73	73.97	75.84	81.47
Ours	95.40	81.70	77.61	76.02	82.68

Table 4: Performance comparison when replacing class-aware covariance sampling with random sampling and γ with a fixed γ_0.

Network/Dataset	PACS	VLCS	Digits-DG
ResNet-18	0.14/1.04	0.1/0.94	0.2/0.39
ResNet-50	0.04/1.93	0.03/0.58	0.07/0.50

Table 5: Extra computational overhead (theoretical/empirical in rel. %) during training by MGFA over AGG (baseline).

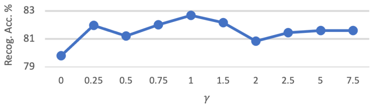

Figure 3: Impact of γ on the overall(average) accuracy of our method.

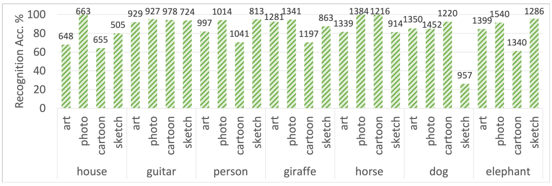

Figure 4: Class-wise accuracy (%) in target domains by MGFA as a function of training images from the source classes. The number on each bar is # training images from the source classes.

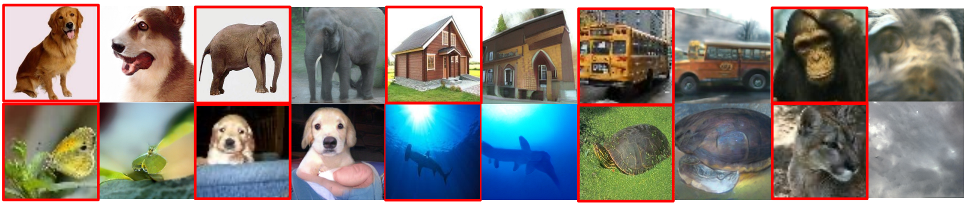

Figure 5: Visualizing augmented images that are obtained via reconstruction for nine different classes. Input image is in red box, and next one is an augmented version.

5 Conclusion and Future Work

We presented a new DG approach that performs feature augmentation by identifying the dominant modes of change in source domain(s) and then implicitly including the augmented versions along those directions. Further, our augmentation process is efficient bearing little extra training overhead. Experimental results on three benchmarks show that our method delivers favourable performance against the SOTA DG methods. Moreover, encouraging performance in challenging single source DG further validates the effectiveness of our method.

In future, we will investigate how the method's performance can be further improved in Digits-DG dataset. Also, we would like to introduce some explicit constraint to have more control on the transformations in the synthesized samples, especially when an image is under-represented in a training class. Finally, we will explore extending our loss formulation to unsupervised DA and source-free DA settings.

References

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pages 998–1008, 2018.

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[3] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24:2178–2186, 2011.

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

[5] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in neural information processing systems*, pages 343–351, 2016.

[6] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[7] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. 2020.

[8] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pages 6450–6461, 2019.

[9] Antonio D'Innocente and Barbara Caputo. Domain generalization with domain-specific aggregation modules. In *German Conference on Pattern Recognition*, pages 187–198. Springer, 2018.

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

[11] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

[12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 1180–1189. JMLR. org, 2015.

[13] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015.

[14] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012.

[15] Marcos Grosso, Adrian Kalstein, Gustavo Parisi, Adrian E Roitberg, and Sebastian Fernandez-Alberti. On the analysis and comparison of conformer-specific essential dynamics upon ligand binding to a protein. *The Journal of Chemical Physics*, 142(24): 06B619_1, 2015.

[16] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. 2020.

[19] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[21] WJ Krzanowski. Between-groups comparison of principal components. *Journal of the American Statistical Association*, 74(367):703–707, 1979.

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] 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.

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

[25] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. *In ICCV*, 2019.

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

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

[28] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[29] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[30] Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Best sources forward: domain generalization through source-specific nets. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1353–1357. IEEE, 2018.

[31] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 466–483. Springer, 2020.

[32] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5715–5725, 2017.

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

[34] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[35] Ekta Prashnani, Hong Cai, Yasamin Mostofi, and Pradeep Sen. Pieapp: Perceptual image-error assessment through pairwise preference. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[36] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. 2020.

[37] 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.

[38] Ashish Shrivastava, Sumit Shekhar, and Vishal M Patel. Unsupervised domain adaptation using parallel transport on grassmann manifold. In *IEEE winter conference on applications of computer vision*, pages 277–284. IEEE, 2014.

[39] Kowshik Thopalli, Rushil Anirudh, Jayaraman J Thiagarajan, and Pavan Turaga. Multiple subspace alignment improves domain adaptation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3552–3556. IEEE, 2019.

[40] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017.

[41] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[42] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems*, pages 5334–5344, 2018.

[43] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. 2020.

[44] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. Implicit semantic data augmentation for deep networks. In *Advances in Neural Information Processing Systems*, pages 12635–12644, 2019.

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

[46] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *In International Conference on Learning Representations (ICLR)*, 2018.

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[48] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. 2020.