LEARNING TO LEARN DOMAIN-IN Variant PARAMETERS FOR DOMAIN GENERALIZATION

Feng Hou$^{1,2}$, Yao Zhang$^{1,2}$, Yang Liu$^{1,2}$, Jin Yuan$^3$, Cheng Zhong$^4$, Yang Zhang$^4$, Zhongchao Shi$^4$, Jianping Fan$^4$, Zhiqiang He$^{1,2,5}$

$^1$ Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
$^2$ University of Chinese Academy of Sciences, Beijing, China
$^3$ Southeast University, Nanjing, China
$^4$ AI Lab, Lenovo Research, Beijing, China
$^5$ Lenovo Ltd, Beijing, China

ABSTRACT

Due to domain shift, deep neural networks (DNNs) usually fail to generalize well on unknown test data in practice. Domain generalization (DG) aims to overcome this issue by capturing domain-invariant representations from source domains. Motivated by the insight that only partial parameters of DNNs are optimized to extract domain-invariant representations, we expect a general model that is capable of well perceiving and emphatically updating such domain-invariant parameters. In this paper, we propose two modules of Domain Decoupling and Combination (DDC) and Domain-invariance-guided Backpropagation (DIGB), which can encourage such general model to focus on the parameters that have a unified optimization direction between pairs of contrastive samples. Our extensive experiments on two benchmarks have demonstrated that our proposed method has achieved state-of-the-art performance with strong generalization capability.

Index Terms—Domain generalization, domain shift, domain-invariant parameters

1. INTRODUCTION

Deep neural networks (DNNs) have made advanced progress on computer vision over the past few years. Most algorithms strongly rely on the assumption that training data and test data are independent and identically distributed [1]. However, in many practical applications, domain shift often leads to performance degeneration of DNNs, which greatly hinder their further development. To deal with the issue of domain shift, Domain Generalization [2, 3, 4] is proposed to learn general representations without accessing the target domain. Existing methods have made tremendous progress on utilizing adversarial training [5, 6, 7], meta learning [8, 9, 10] and data augmentation [11, 4, 12, 13] to explore different types of invariances from the source domains. Such invariances commonly refers to domain-invariant information that is strongly related to task objectives.

Aside from the domain-invariant information, the other particular domain characteristics with different distributions also play an important role to tackle domain shift, namely domain-specific information. The major limitation of domain generalization is the restricted diversity of domain-specific information [4], some methods [11, 4, 12, 13] make efforts to increase this diversity by augmenting the source data. Although these methods have achieved a moderate improvement, only enriching the source data according to the existing distribution still suffers from the overfitting issue. As shown in Fig 1, the augmented and original samples share a high similarity in the feature space.

Inspired by the observation that partial parameters of a model dominate the domain-invariant representation [14], we propose to enable the model to adaptively perceive and update the domain-invariant parameters to relieve the overfitting issue. To achieve this, we design a framework that learns to learn domain-invariant parameters via paired contrastive samples. In particular, we first develop a Domain Decoupling and Combination (DDC) module, to decompose the features into domain-specific and domain-invariant ones via Fourier transform [15] and then randomly combine them to generate novel domain representations. The original and novel domain
The major challenge of domain generalization is to learn a model that generalizes well on unseen target domains. Notably, different domains share the same label space of $y$, but they contain pairs of data and label $\{(x_i^j, y_i^j)\}_{j=1}^{N_i}$, where $N_i$ is the number of images in the $i$-th domain. Notably, different domains share the same label space of $y_i \in \{1, \ldots, C\}$. The major challenge of domain generalization is to learn a domain-agnostic model $f_0 : X \rightarrow Y$ by only using the source domain-specific features in frequency space to extend the information, we investigate the different combinations of the domain-specific features in frequency space to extend the training samples.

In this section, we first describe the formulation of domain generalization for object recognition and its challenge, then present the proposed Domain Decoupling and Combination (DDC) and Domain-invariance-guided Backpropagation (DIGB) modules in details. The overview of our method is illustrated in Fig. 2.

## 2. METHOD

### 2.1. Preliminaries

In domain generalization, we denote the training set of multi-source domains as $D_s = \{D_1, D_2, \ldots, D_K\}$. Each domain $D_i$ contains pairs of data and label $\{(x_{i}^{j}, y_{i}^{j})\}_{j=1}^{N_i}$, where $N_i$ is the number of images in the $i$-th domain. Notably, different domains share the same label space of $y_i \in \{1, \ldots, C\}$. The major challenge of domain generalization is to learn a domain-agnostic model $f_0 : X \rightarrow Y$ by only using the source domains, so that it is able to generalize well on unseen target domains $D_t$.

### 2.2. Domain Decoupling and Combination

The Domain Decoupling and Combination (DDC) module is designed to generate pairs of contrastive samples with the same domain-invariant information but different domain-specific information based on the decomposition of frequency signal. Given a sample $z \in \mathbb{R}^{C \times H \times W}$, the formulation of its Fourier transformation [15] on each channel is defined as:

$$\mathcal{F}(z)(u, v) = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} z(h, w)e^{-j2\pi(\frac{h}{H}u + \frac{w}{W}v)}, \quad (1)$$

where $u$ and $v$ represent the coordinates in frequency space. Subsequently, the frequency signal $\mathcal{F}(z)$ is decomposed to amplitude spectrum $\mathcal{A} \in \mathbb{R}^{C \times H \times W}$ and phase spectrum $\mathcal{P} \in \mathbb{R}^{C \times H \times W}$. This is motivated by the finding that the amplitude component $\mathcal{A}$ retains rich domain-specific information (i.e., style), and the phase component $\mathcal{P}$ mainly captures domain-invariant information (i.e., content) [16]. Ideally, a controllable and efficient data generation can be achieved by exchanging the amplitude component $\mathcal{A}_s$ from other source data $z_s$ while preserving the phase component $\mathcal{P}_c$ of each instance $z_c$.

As shown in Fig. 2(b), we introduce a binary mask whose origin locates at the center: $\mathcal{M} = 1_{(h, w) \in [-\alpha H, \alpha H, -\alpha W, \alpha W]}$, where $\alpha$ is a hyper-parameter within the range of $[0, 1]$ to control the proportion of the frequency component that needs to be exchanged. In our implementation, we sample the controller $\alpha$ from a beta distribution to achieve a trade-off between high and low frequencies, which represent the details and the overview of images respectively [17]. Then a linearly interpolating mechanism is introduced to synthesize the novel style information:

$$\hat{\mathcal{A}}' = ((1 - \lambda)\mathcal{A}_c + \lambda\mathcal{A}_s) \ast \mathcal{M} + \mathcal{A}_c \ast (1 - \mathcal{M}), \quad (2)$$
where $\lambda$ is an instance-specific weight, following the random mix weight in MixStyle [4]. Finally, the novel domain feature $\hat{z}'$ is generated by combining different style information from arbitrary source domains with unaffected content information via inverse Fourier transformation: $\hat{z}' = F^{-1}(\hat{A}' \cdot P_c)$.

2.3. Domain-invariance-guided Backpropagation

The motivation behind Domain-invariance-guided Backpropagation (DIGB) is that the model optimization direction is sensitive to the distribution shift [18]. The main idea is to distinguish both domain-specific and domain-invariant parameters by comparing their evolution paths on different domain data. As aforementioned, the proposed DDC module generates adequate pairs of contrastive samples from different distributions that are then fed into a siamese network, and the corresponding predictions $\hat{y}$ and $\hat{y}'$ are supervised by the same label $y$. The individual classification error $\ell_\theta$ for the $j$-th data sample $(x_i^j, y_i^j)$ from domain $D_i \in D_s$ is calculated as:

$$\ell_\theta(f_\theta(x_i^j), y_i^j) = -y_i^j \log(\sigma(f_\theta(x_i^j; \theta))),$$

(3)

where $\sigma$ denotes the softmax function. Note that the classification error of the contrastive sample of $(x_i^j, y_i^j)$ is formulated similarly as Eq. 3 with its unique prediction as $f_\theta'(x_i^j)$. As for the backpropagation algorithm employing gradient descent, the gradient vector $g_l^i = \nabla_{\theta_l} \ell_\theta(f_\theta(x_i^j), y_i^j)$ for the $l$-th layer represents the optimization direction of search space $\theta_l$, where $l \in \{1, \ldots, L\}$ and $L$ denotes the total number of layers. Then the layer-wise sensitivity correlated to the domain shift can be viewed as the correlation between two optimization directions for the parameters of each layer, such as $g_l^i$ and $g_l'^i$. Following [18], the cosine similarity $s_l^i$ is utilized to represent the correlation between the two gradient vectors:

$$s_l^i = \frac{\hat{g}_l^i \cdot g_l'^i}{\|\hat{g}_l^i\| \|g_l'^i\|} \in \mathbb{R}.$$

(4)

Here, we intentionally use a continuous interval to distinguish domain-specific and domain-invariant model parameters rather than discrete values. Specifically, the calculated cosine similarities are processed with a min-max normalization and formed into an enhancement vector before sending to the conventional optimizer (i.e., SGD): $\hat{g}_l = (g_l + g_l') \cdot w_l$. Furthermore, to avoid unstable measurement, the similarity vector $w$ is updated via exponential moving average: $w = \beta \cdot w + (1 - \beta) \cdot \hat{w}$, where $\hat{w}$ is the estimation from current mini-batch, and $\beta$ is the momentum hyper-parameter with default 0.999.

3. EXPERIMENTS

In this section, we evaluate our methods on two popular domain generalization benchmarks, and analyze the effectiveness from a wide range of aspects.

3.1. Datasets and Implementation Details

PACS [21] consists of 9,991 images from 4 domains with different image styles and 7 classes. Office-Home [22] is a larger and more challenging dataset with 4 domains. Each domain involves 65 categories of objects found in office and home.

We adopt ResNet-18 and ResNet-50 pre-trained on ImageNet as our backbone following previous work [4, 11]. We train the network with Stochastic Gradient Descent (SGD) optimizer and set the batch size to 64. The initial learning rate is 0.002 with cosine decay rule for 50 epochs. We use the common leave-one-domain-out evaluation to demonstrate the effectiveness of our method. Specifically, we select three domains as the source for training, and the remaining one as the target. All reported results are the average of three runs with different random seeds following previous work [4].

3.2. Comparison with State-of-the-art Methods

PACS: The results are shown in Table 1. Compared with the baseline of vanilla ResNet-18, our method achieves significant improvements of 10.1% on art painting, 5.7% on cartoon, and 14.4% on sketch. Eventually, the average accuracy exceeds the baseline of 7.7% which demonstrates the effectiveness of our method. When it comes to the widely used methods, our method beats most of them by a large margin, including MixStyle [4], DSU [12]. Significantly, our method also surpasses the latest state-of-the-art (SOTA) TAF-Cal [13] with 0.4% on ResNet-18, and MVDG [10] with 0.6% on ResNet-50.

| Dataset | Backbone | PACS | ResNet50 | Office-Home |
|---------|----------|------|----------|-------------|
| Target Domain | A | C | P | S | Avg | A | C | P | R | Avg |
| Baseline | 77.0 | 75.5 | 98.6 | 69.2 | 78.5 | 86.2 | 78.7 | 97.7 | 70.6 | 83.3 | 58.9 | 49.4 | 74.3 | 56.2 | 46.7 |
| MixStyle [4] | 84.1 | 78.8 | 96.1 | 75.9 | 85.7 | 85.7 | 58.7 | 74.2 | 75.8 | 84.3 | 60.2 | 54.8 | 74.1 | 75.1 | 66.1 |
| DSU [12] | 85.6 | 79.6 | 95.6 | 77.6 | 84.1 | 85.1 | 85.4 | 81.8 | 94.5 | 84.5 | 80.2 | 60.3 | 54.9 | 74.5 | 76.6 | 66.6 |
| FACT [11] | 85.4 | 78.4 | 95.2 | 79.2 | 84.6 | 85.9 | 91.6 | 98.4 | 84.5 | 88.2 | 60.3 | 54.9 | 74.5 | 76.6 | 66.6 |
| RSC [19] | 83.4 | 80.3 | 96.0 | 80.9 | 85.2 | 85.7 | 87.9 | 92.2 | 97.9 | 83.4 | 85.9 | 60.2 | 53.5 | 74.4 | 76.6 | 66.2 |
| FSDCL [20] | 85.3 | 81.3 | 95.6 | 81.2 | 85.9 | 85.5 | 83.8 | 94.6 | 82.9 | 88.0 | 60.3 | 54.3 | 75.1 | 77.5 | 66.8 |
| MVDG [10] | 85.6 | 80.0 | 95.5 | 85.4 | 86.6 | 89.3 | 93.4 | 94.7 | 84.6 | 89.3 | 60.3 | 54.3 | 75.1 | 77.5 | 66.8 |
| TAF-Cal [13] | 85.7 | 82.6 | 96.1 | 82.6 | 88.4 | 90.5 | 84.9 | 97.8 | 83.8 | 89.3 | 61.5 | 58.9 | 74.9 | 77.0 | 67.1 |

Table 1. Performance comparison with other methods for domain generalization.
Effectiveness Analysis: We also apply our DIGB module to another data augmentation methods [4, 12] that employing different domain decomposition methods. The results in Table 2(b) show that our module cooperates smoothly with the new domain decomposition method and achieves promising performance improvement (especially on sketch that has an extensive domain shift), which indicates that strengthening domain-invariant parameters is universal and effective enough to cope with many existing augmentation methods.

Robust Analysis: We perform robust evaluation on a new benchmark for robustness in real-world Out-of-Distribution Shifts [23]. In Table 2(c), our method achieves satisfactory improvement with the average accuracy of six real-world domain shifts compared to the baseline and MixStyle [4]. It indicates that our method is robust to varied domain shifts, not only the limited domain gap dominated by style information.

Domain Invariance Analysis: In Table 2(c), our method achieves satisfactory improvement with the average accuracy of six real-world domain shifts compared to the baseline and MixStyle [4]. It indicates that our method is robust to varied domain shifts, not only the limited domain gap dominated by style information.

Table 2. Ablation study and further analysis. Note that * represents the reproduced results.

| Method | A | C | P | S | Avg. | Method | A | C | P | S | Avg. |
|--------|---|---|---|---|------|--------|---|---|---|---|------|
| Baseline | 87.0 | 75.9 | 96.0 | 69.2 | 79.5 | MixStyle* [11] | 82.5 | 79.4 | 96.0 | 71.1 | 82.2 |
| +DDC | 83.4 | 78.6 | 95.2 | 77.4 | 83.7 | MixStyle-DIGB | 83.1 | 80.5 | 96.4 | 75.4 | 83.9 |
| +DIGB | 85.1 | 80.6 | 96.2 | 81.9 | 86.0 | DSC [12] | 81.5 | 80.0 | 96.0 | 76.6 | 83.5 |
| +MVP | 87.1 | 81.6 | 96.5 | 83.6 | 87.2 | DSC+DIGB | 83.7 | 81.8 | 96.2 | 77.5 | 84.8 |

(a) Ablation study

(b) Applying DIGB to other methods

Table 2. Ablation study and further analysis. Note that * represents the reproduced results.

| Methods | A | C | P | S | Avg. |
|---------|---|---|---|---|------|
| Baseline | 91.9 | 84.4 | 90.5 | 88.4 | 82.2 |
| MixStyle [4] | 91.3 | 83.4 | 88.4 | 87.8 | 82.2 |
| Ours | 92.2 | 85.2 | 90.5 | 89.5 | 85.9 |

(c) Results comparison on ROBIN dataset

3.3. Ablation Study and Further Analysis

In this section, we analyze the effectiveness of each proposed component based on PACS benchmark. For all experiments, we use the ImageNet pre-trained ResNet-18 as backbone.

Ablation Study: In order to investigate the role of each component, we conduct an ablation study on PACS dataset. As shown in Table 2(a), by applying the samples generated from DDC as augmentations the model achieves obvious improvements over the baseline, demonstrating that increasing the diversity of source domain is an effective way, which agrees with recent works [4, 12]. Furthermore, the DIGB can further enhance the performance by 2.3%. It reveals that exploring and emphasizing domain-invariant parameters empower our model to better generalize on novel domains. Eventually, with the help of Multi-view prediction (MVP) [10], we achieve a new SOTA with an averaged performance of 87.2%.

Effectiveness Analysis: We also apply our DIGB module to another data augmentation methods [4, 12] that employing different domain decomposition methods. The results in Table 2(b) show that our module cooperates smoothly with the new domain decomposition method and achieves promising performance improvement (especially on sketch that has an extensive domain shift), which indicates that strengthening domain-invariant parameters is universal and effective enough to cope with many existing augmentation methods.

Office-Home: As shown in Table 1 that our method performs the best results on three domains (Art, Product, and Real-world), and achieves SOTA average accuracy of 67.6%. Overall comparisons verify the effectiveness of DDC and DIGB.

4. CONCLUSION

In this paper, a novel framework is developed to identify and strengthen the domain-invariant parameters to relieve the overfitting on limited source data for domain generalization. Our extensive experiments and analysis have demonstrated the effectiveness of our proposed method.
5. REFERENCES

[1] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, et al., “Domain generalization: A survey,” arXiv preprint arXiv:2103.02503, 2021. 1

[2] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, et al., “Generalizing across domains via cross-gradient training,” arXiv preprint arXiv:1804.10745, 2018. 1

[3] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang, “Learning to generate novel domains for domain generalization,” in ECCV. Springer, 2020, pp. 561–578. 1

[4] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang, “Domain generalization with mixstyle,” arXiv preprint arXiv:2104.02008, 2021. 1, 3, 4

[5] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, et al., “Domain generalization via conditional invariant representations,” in AAAI, 2018, vol. 32. 1

[6] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, et al., “Deep domain generalization via conditional invariant adversarial networks,” in ECCV, 2018, pp. 624–639. 1

[7] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang, “Deep domain-adversarial image generation for domain generalisation,” in AAAI, 2020, vol. 34, pp. 13025–13032. 1

[8] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales, “Learning to generalize: Meta-learning for domain generalization,” in AAAI, 2018. 1

[9] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa, “Metareg: Towards domain generalization using meta-regularization,” NIPS, vol. 31, pp. 998–1008, 2018. 1

[10] Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao, “More is better: A novel multi-view framework for domain generalization,” arXiv preprint arXiv:2112.12329, 2021. 1, 3, 4

[11] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, et al., “A fourier-based framework for domain generalization,” in CVPR, 2021. 1, 3

[12] Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and Ling-Yu Du, “Uncertainty modeling for out-of-distribution generalization,” arXiv preprint arXiv:2202.03958, 2022. 1, 3, 4

[13] Xingchen Zhao, Chang Liu, Anthony Sicilia, Seong Jae Hwang, and Yun Fu, “Test-time fourier style calibration for domain generalization,” arXiv preprint arXiv:2205.06427, 2022. 1, 3

[14] Fan Wang, Zhongyi Han, Yongshun Gong, and Yilong Yin, “Exploring domain-invariant parameters for source free domain adaptation,” in CVPR, 2022, pp. 7151–7160. 1

[15] Henri J Nussbaumer, “The fast fourier transform,” in Fast Fourier Transform and Convolution Algorithms, pp. 80–111. Springer, 1981. 1, 2

[16] Alan V Oppenheim and Jae S Lim, “The importance of phase in signals,” Proceedings of the IEEE, vol. 69, no. 5, pp. 529–541, 1981. 2

[17] Zhiyu Jin, Xuli Shen, Bin Li, and Xiangyang Xue, “Style spectroscope: Improve interpretability and controllability through fourier analysis,” arXiv preprint arXiv:2208.06140, 2022. 2

[18] Sungha Choi, Seunghan Yang, Seokeon Choi, and Sunggrack Yun, “Improving test-time adaptation via shift-agnostic weight regularization and nearest source prototypes,” arXiv preprint arXiv:2207.11707, 2022. 3

[19] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang, “Self-challenging improves cross-domain generalization,” in ECCV. Springer, 2020, pp. 124–140. 3

[20] Seogkyu Jeon, Kibeom Hong, Pilhyeon Lee, Jeewook Lee, and Hyeran Byun, “Feature stylization and domain-aware contrastive learning for domain generalization,” in ACM MM, 2021, pp. 22–31. 3

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales, “Deeper, broader and artier domain generalization,” in ICCV, 2017, pp. 5542–5550. 3

[22] Chelsea Finn, Pieter Abbeel, and Sergey Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in ICML. PMLR, 2017, pp. 1126–1135. 3

[23] Bingchen Zhao, Shaozuo Yu, Wufei Ma, Mingxin Yu, Shenxiao Mei, Shengxiao Mei, Angtian Wang, Ju He, Alan Yuille, and Adam Kortylewski, “Robin: A benchmark for robustness to individual nuisances in real-world out-of-distribution shifts,” arXiv preprint arXiv:2111.14341, 2021. 4

[24] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Hancheol Cho, Seunghyun Park, Yunsung Lee, and Sungae Park, “Swad: Domain generalization by seeking flat minima,” NIPS, vol. 34, pp. 22405–22418, 2021. 4

[25] Laurens Van der Maaten and Geoffrey Hinton, “Visualizing data using t-sne,” Journal of machine learning research, vol. 9, no. 11, 2008. 4