Supplementary Material for: Improving Generalization with Domain Convex Game

Fangrui Lv[1] **Jian Liang** **Shuang Li**[1,*] **Jinming Zhang**[1] **Di Liu**

[1] Beijing Institute of Technology, China

[1] {fangruilv,shuangli,jinming-zhang}@bit.edu.cn {liangjianzb12,liudi010}@gmail.com

A. Social Impact

Our work focuses on domain generalization and attempts to make each training domain contribute to model generalization, which validates and further enhances the effectiveness of domain augmentation strand. This method produces a positive impact on the society and community, saves the cost and time of data annotation, boosts the reusability of knowledge across domains, and greatly improves the efficiency. Nevertheless, this work suffers from some negative influences, which is worthy of further research and exploration. Specifically, more jobs of classification or target detection for rare or variable conditions may be cancelled. Moerover, we should be cautious about the result of the failure of the system, which could render people believe that classification was unbiased. Still, it might be not, which might be misleading, e.g., when using the system in a highly variable unseen target domain.

B. Algorithm of DCG

In this work, we propose a Domain Convex Game (DCG) framework to guarantee and further enhance the validity of domain augmentation approaches by casting DG as a convex game between domains. Here, we summarize the training process of DCG based on the discussions in main body as Algorithm 1.

Algorithm 1 The Algorithm of Domain Convex Game.

```
Input: $P+Q$ diversified source domains $\mathcal{D}_s \cup \mathcal{D}_s^{aug}$; Hyper-parameters: $\omega, k$.
1: randomly initialize model parameters $\boldsymbol{\theta}$.
2: for iter in iterations do
3:     Randomly sample a mini-batch of $\mathcal{D}_s$ as $B$ and a mini-batch of $\mathcal{D}_s^{aug}$ as $B^{aug}$.
4:     Split: $\tilde{\mathcal{D}}_s$ and $\tilde{\mathcal{D}}_t \leftarrow B$, Pick out: $\tilde{\mathcal{D}}_s^{aug}$ from $B^{aug}$.
5:     Construct coalitions $S, T$ by randomly sampling from $\tilde{\mathcal{D}}_s \cup \tilde{\mathcal{D}}_s^{aug}$; construct coalitions $S \cup T, S \cap T$.
6:     Calculate supermodularity regularization loss $\mathcal{L}_{sm}$.
7:     Pick out low-quality samples $\mathcal{D}_{del}$ with the top-$k$ score.
8:     Calculate supervision loss $\mathcal{L}_{sup}$.
9:     Update $\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{sup} + \omega\mathcal{L}_{sm}$.
10: end for
```

C. Experimental Details

For all benchmarks, we conduct the commonly used leave-one-domain-out experiments [8], where we choose one domain as the unseen target domain for evaluation, and train the model on all remaining domains. We adopt the standard augmentation protocol as in [2], all images are resized to 224 × 224, following with random resized cropping, horizontal flipping and color jittering. And the Fourier domain augmentation strategy utilized to diversify source domains closely follows the implementations in [14]. The network backbone is set to ResNet-18 or ResNet-50 pre-trained on ImageNet [4] following other related works. We train the network using mini-batch SGD with batch size 16, momentum 0.9 and weight decay 5e-4 for 50 epochs. The initial learning rate is 0.001 and decayed by 0.1 at 80% of the total epochs. The meta step size α is set to be the same as the learning rate. For the hyper-parameters, i.e., the weight of regularization loss ω and the number of discarded bad samples in each iteration k, their values are selected on validation data following standard practice, where we use 90% of available data as training data and 10% as validation data. Specifically, we set $\omega = 0.1$ and $k = 5$ for all experiments. Our framework is implemented with PyTorch on NVIDIA GeForce RTX 3090 GPUs. All results are reported based on the average accuracy over three independent runs for a fair comparison.

D. Additional Results

D.1. Time cost analysis

We conduct experiments to study the efficiency of our method in the training and inference stages respectively, and the results are shown in Table 2. For the training stage, the time cost of DCG is indeed relatively high, which is due to

Methods	Art	Cartoon	Photo	Sketch	Avg.
			ResNet18		
MLDG [9]	78.70	73.30	94.00	65.10	80.70
L2A-OT [19]	83.30	78.20	96.20	73.60	82.80
RSC [5]	83.43	80.31	95.99	80.85	85.15
DSU [10]	83.60	79.60	95.80	77.60	84.10
DeepAll [18]	77.63±0.84	76.77±0.33	95.85±0.20	69.50±1.26	79.94
MASF [3]	80.29±0.18	77.17±0.08	94.99±0.09	71.69±0.22	81.04
DDAIG [18]	84.20±0.30	78.10±0.60	95.30±0.40	74.70±0.80	83.10
MixStyle [20]	84.10±0.40	78.80±0.40	96.10±0.30	75.90±0.90	83.70
FACT [14]	85.37±0.29	78.38±0.29	95.15±0.26	79.15±0.69	84.51
STNP [6]	84.41±0.62	79.25±0.98	94.93±0.07	**83.27±2.03**	85.47
DCG (*ours*)	**85.94±0.21**	**80.76±0.36**	**96.41±0.17**	82.08±0.44	**86.30**
			ResNet50		
RSC [5]	87.89	82.16	97.92	83.35	87.83
PCL [15]	90.20	83.90	**98.10**	82.60	88.70
DeepAll [18]	84.94±0.66	76.98±1.13	97.64±0.10	76.75±0.41	84.08
FACT [14]	89.63±0.51	81.77±0.19	96.75±0.10	84.46±0.78	88.15
DDG [17]	88.90±0.60	85.00±1.90	97.20±1.20	84.30±0.70	88.90
STNP [6]	**90.35±0.62**	84.20±1.43	96.73±0.46	85.18±0.46	89.11
DCG (*ours*)	90.24±0.48	**85.12±0.79**	97.76±0.13	**86.31±0.64**	**89.84**

Table 1. Leave-one-domain-out results on PACS.

the use of meta learning when constructing the regularization term and the backpropagation when calculating the score for sample filter. For substitute, we may edit the backpropagation path that computes gradients of inputs only on a smaller subnetwork to reduce time cost. Besides, we can see that for the inference stage, our DCG method is as efficient as other methods and does not incur additional time costs. Note that this work is an innovative effort to study the relation between model generalization and domain diversity, which is in a preliminary stage. And we will further explore more efficient techniques in future research.

Methods	Training	Inference
DEEPALL [14]	168 s	5 s
FACT [14]	186 s	5 s
MLDG [9]	275 s	5 s
DCG w/o Filter.	349 s	5 s
DCG	467 s	5 s

Table 2. Running Time per Epoch.

D.2. Experimental Results with Error Bars

For the sake of objective, we run all the experiments multiple times with random seed. We report the average results in the main body of paper for elegant, and show the complete results with error bars in the form of mean±std below (Table. 1, 3, 4).

Methods	Art	Clipart	Product	Real	Avg.
MLDG [9]	52.88	45.72	69.90	72.68	60.30
SagNet [11]	60.20	45.38	70.42	73.38	62.34
RSC [5]	58.42	47.90	71.63	74.54	63.12
L2A-OT [19]	60.60	50.10	74.80	**77.00**	65.60
DSU [10]	60.20	54.80	74.10	75.10	66.10
DeepAll [18]	57.88±0.20	52.72±0.50	73.50±0.30	74.80±0.10	64.72
DDAIG [18]	59.20±0.10	52.30±0.30	74.60±0.30	76.00±0.10	65.50
MixStyle [20]	58.70±0.30	53.40±0.20	74.20±0.10	75.90±0.10	65.50
FACT [14]	60.34±0.11	54.85±0.37	74.48±0.13	76.55±0.10	66.56
STNP [6]	59.55±0.21	55.01±0.29	73.57±0.28	75.52±0.21	65.89
DCG (*ours*)	**60.67±0.14**	**55.46±0.32**	**75.26±0.18**	76.82±0.09	**67.05**

Table 3. Leave-one-domain-out results on Office-Home.

Methods	Clipart	Painting	Real	Sketch	Avg.
DeepAll [18]	65.30	58.40	64.70	59.00	61.86
ERM [13]	65.50 ± 0.3	57.10 ± 0.5	62.30 ± 0.2	57.10 ± 0.1	60.50
MLDG [9]	65.70 ± 0.2	57.00 ± 0.2	63.70 ± 0.3	58.10 ± 0.1	61.12
Mixup [16]	67.10 ± 0.2	59.10 ± 0.5	64.30 ± 0.3	59.20 ± 0.3	62.42
MMD [7]	65.00 ± 0.5	58.00 ± 0.2	63.80 ± 0.2	58.40 ± 0.7	61.30
SagNet [11]	65.00 ± 0.4	58.10 ± 0.2	64.20 ± 0.3	58.10 ± 0.4	61.35
CORAL [12]	66.50 ± 0.2	59.50 ± 0.4	66.00 ± 0.6	59.50 ± 0.1	62.87
MTL [1]	65.30 ± 0.5	59.00 ± 0.4	65.60 ± 0.4	58.50 ± 0.2	62.10
DCG (*ours*)	**69.38±0.19**	**61.79±0.22**	**66.34±0.27**	**63.21±0.09**	**65.18**

Table 4. Leave-one-domain-out results on Mini-DomainNet.

References

[1] Gilles Blanchard, Aniket Anand Deshmukh, Ürün Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *J. Mach. Learn. Res.*, pages 2:1–2:55, 2021. 3

[2] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, pages 2229–2238, 2019. 1

[3] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, pages 6447–6458, 2019. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[5] Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, pages 124–140, 2020. 2, 3

[6] Juwon Kang, Sohyun Lee, Namyup Kim, and Suha Kwak. Style neophile: Constantly seeking novel styles for domain generalization. In *CVPR*, pages 7130–7140, June 2022. 2, 3

[7] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, pages 10285–10295, 2019. 3

[8] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, pages 5543–5551, 2017. 1

[9] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, pages 3490–3497, 2018. 2, 3

[10] Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and Lingyu Duan. Uncertainty modeling for out-of-distribution generalization. In *ICLR*, 2022. 2, 3

[11] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap via style-agnostic networks. *CoRR*, abs/1910.11645, 2019. 3

[12] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019. 3

[13] Vladimir Vapnik. An overview of statistical learning theory. *IEEE Trans. Neural Networks*, 10(5):988–999, 1999. 3

[14] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *CVPR*, pages 14383–14392, 2021. 1, 2, 3

[15] 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 *CVPR*, pages 7087–7097, 2022. 2

[16] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 3

[17] Hanlin Zhang, Yi-Fan Zhang, Weiyang Liu, Adrian Weller, Bernhard Schölkopf, and Eric P. Xing. Towards principled disentanglement for domain generalization. In *CVPR*, pages 8014–8024, 2022. 2

[18] Kaiyang Zhou, Yongxin Yang, Timothy M. Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, pages 13025–13032, 2020. 2, 3

[19] Kaiyang Zhou, Yongxin Yang, Timothy M. Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, pages 561–578, 2020. 2, 3

[20] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021. 2, 3