Supplementary Material

1 Evaluation on Digits-DG

Implementation. We exploit a compact convolutional network as our backbone on Digits-DG dataset, following [10]. Images are resized to 32×32, and we split the data into 80% for training and 20% for validating. The models are trained on the training set of source domains for fair comparison. Models are trained from scratch, and the training details are the same as those of PACS except for the initial learning rate and batch size. We initialize the learning rate using 0.05 and set batch size to 128.

Results. From the results in Table 1, we can see that our AugLearn and AugLearn-F again can both improve ERM baseline, while AugLearn-F enables a larger boost 1.5% compared with 0.4% of AugLearn. Unsurprisingly, our AugLearn and AugLearn-F variants are still complementary with MixStyle on this benchmark, producing 0.7% and 2.1% accuracy gains and leading to the new state of the art results on this benchmark. Overall, these results demonstrate the good generality of our proposed methods.

Method	MNIST	MNIST-M	SVHN	SYN	Average
ERM	95.8	59.8	62.8	79.4	74.5
CCSA	95.2	58.2	65.5	79.1	74.5
MMD-AAE	96.5	58.4	65.0	78.4	74.6
CrossGrad	96.7	61.1	65.3	80.2	75.8
JiGen	96.5	61.4	63.7	74.0	73.9
DDAIG	96.6	64.1	68.6	81.0	77.6
L2A-OT	96.7	63.9	68.6	**83.2**	78.1
MixStyle	96.5	63.5	64.7	81.2	76.5
ERM+AugLearn	96.1	62.3	63.2	78.1	74.9 (+0.4)
ERM+AugLearn-F	96.3	63.9	64.3	79.3	76.0 (+1.5)
MixStyle+AugLearn	96.7	65.1	66.8	80.2	77.2 (+0.7)
MixStyle+AugLearn-F	**96.9**	**65.6**	**69.2**	82.8	**78.6** (+2.1)

Table 1: Leave-one-domain-out generalization results on Digits-DG.

2 Evaluations on DomainNet

Implementation. We exploit ResNet18 as the backbone and resize the image to 32×32. We train the model on the training set and test the trained model on the testing set following [2]. Other training settings are the same as those of PACS dataset.

Results. From the results in Table 2, we observe that AugLearn(-F) outperforms ERM method with a clear improvement margin, 1.28%(1.56%) accuracy on DomainNet dataset. Notably, AugLearn(-F) outperforms ERM on the most challenging held-out quickdraw domain by 2.19%(3.55%).

Method	clipart	infograph	painting	quickdraw	real	sketch	Average
ERM	38.76	18.22	30.66	16.46	32.98	29.67	27.79
ERM+AugLearn	39.21	20.17	31.84	18.65	33.69	30.84	29.07
ERM+AugLearn-F	39.86	20.76	30.97	20.01	32.86	31.64	29.35

Table 2: Leave-one-domain-out generalization results on DomainNet.

3 Comparison between DG and DA methods

Domain Generalization assumes no access to target domain data during model training, while Domain Adapation exploits target distributions for model training. DG and DA are rarely compared directly. Now, we compare our method with several DA methods such as MCD [5], M^{3}SDA[7] and CMSS [9] on PACS dataset in Table3. Those DA models perform better than AugLearn(-F), which is expected, as they get access to the target domain data during training.

Method	Art	Cartoon	Photo	Sketch	Average
ERM	78.5	75.2	96.2	67.9	79.5
MCD	88.7	88.9	96.4	73.9	87.0
M^{3}SDA	89.3	88.9	97.3	76.7	88.3
CMSS	88.6	90.4	96.9	82.0	89.5
ERM+AugLearn	82.9	78.8	94.5	80.1	84.1
ERM+AugLearn-F	81.9	79.2	95.3	80.7	84.3

Table 3: Comparison between DG and DA methods on PACS.

4 AugLearn vs. Strong Augmentations

In order to verify the efficacy of our proposed AugLearn(-F), we also compare our methods with several standard strong augmentation methods, such as Cutout [2], CutMix [8] and DropBlock [4], on PACS. From the results in Figure 1, we observe that our proposed AugLearn(-F) method outperforms all strong augmentation methods on all domains with a large margin except the photo domain. It is worth noting that the AugLearn(-F) method achieves much better performance than strong augmentation methods on the held-out sketch domain. This means our AugLearn method is capable of generating augmented images to make the model focus on abstract shapes. This suggests that optimizing the augmentation module explicitly with domain shift is beneficial to learn the underlying abstract information which is generalizable to unseen domains.

5 Optimizing the hyperparameters of simple augmentations.

Table 4 shows the results of using augmentations, i.e. color jitter and cutmix, with a fixed hyperparameter and our optimized value. We can see our AugLearn module improves their

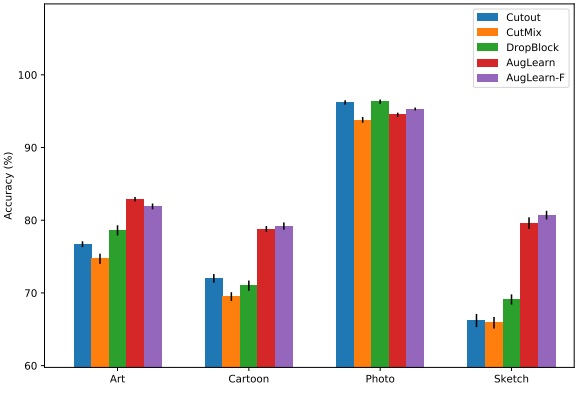

Figure 1: Comparison with standard strong augmentation methods on PACS.

performance from 77.83% and 75.98% to 79.19% and 77.03% on PACS respectively. This further shows that using a hand-crafted augmentation is not optimal for DG and demonstrates the necessity of our proposed meta-learning based augmentation module.

Method	Art	Cartoon	Photo	Sketch	Average
ERM	78.5	75.2	96.2	67.9	79.5
ColorJitter scale (fixed)	74.90	74.02	95.39	66.99	77.83
ColorJitter scale (learned)	76.35	76.28	95.68	68.44	79.19
Cutmix alpha (fixed)	74.72	69.48	93.81	65.89	75.98
Cutmix alpha (learned)	76.51	68.96	95.32	67.32	77.03
ERM+AugLearn	82.9	78.8	94.5	80.1	84.1
ERM+AugLearn-F	81.9	79.2	95.3	80.7	84.3

Table 4: Results of optimizing the hyperparameters of simple augmentations.

6 Applicability to Other Base DG Methods

Our proposed AugLearn is model agnostic and applicable to any base DG methods. Beside MixStyle, we also apply our AugLearn on top of other state of the art DG methods, such as JiGen [0] and CrossGrad [6], on Digits-DG. From the results in Table 5, we can see that incorporating our AugLearn improves both JiGen and CrossGrad noticeably with 1.4% and 1.7% accuracy improvement respectively, resulting in the state of the art performance on Digits-DG. These results further verify the generality and applicability of our AugLearn.

7 Training Cost

In our AugLearn, we meta optimize the augmentation module every 30 steps of updating the classification model. We calculate the training cost per step of our AugLearn is 0.049s, whereas MixStyle takes 0.03s per step. Apparently, we can see that our method is compara-

Method	AugLearn	MNIST	MNIST-M	SVHN	SYN	Avg.
JiGen	✗	97.6	59.1	66.0	87.8	77.6
	✓	98.1	61.3	69.2	87.5	79.0
CrossGrad	✗	95.7	60.3	63.5	80.1	74.9
	✓	96.3	63.4	66.1	80.6	76.6

Table 5: Applicability to other DG methods. The results of JiGen and CrossGrad are based on our run.

ble to the state of art methods in terms of the training cost, but trains a stronger model. This experiment is done on PACS using GPU Nvidia RTX 2080Ti.

References

[1] F. M. Carlucci, A. D'Innocente, S. Bucci, and T. Caputo, B.and Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[2] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[3] G. Ghiasi, T. Lin, and Q. V. Le. Dropblock: A regularization method for convolutional networks. *arXiv preprint arXiv:1810.12890*, 2018.

[4] 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/CVF international conference on computer vision*, pages 1406–1415, 2019.

[5] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

[6] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi. Generalizing across domains via cross-gradient training. In *Proceedings of the ICLR*, 2018.

[7] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. In *European Conference on Computer Vision*, pages 608–624. Springer, 2020.

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

[9] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33:16096–16107, 2020.

[10] K. Zhou, Y. Yang, T. M. Hospedales, and T. Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, pages 13025–13032, 2020.