Domain Generalization by Learning and Removing Domain-specific Features – Appendix

Anonymous Author(s)
Affiliation
Address
email

A Implementation details

A.1 Data processing

The training data are processed with a fixed pipeline. The input images are resized and cropped to 224×224, then randomly transformed by horizontal flip, color jitter, and grayscale, and finally normalized with the standard ImageNet [9] channel statistics. This setting is widely used in previous works on domain generalization [1, 3, 4].

A.2 Training setting

The source data are split into a training and a validation set according to [1, 3, 4]. The Stochastic Gradient Descent is used as the optimizer with weight decay 10^{-4} and momentum 0.9. The learning rate is chosen from $\{10^{-3}, 10^{-4}\}$ according to the performance of the validation set. For the main results in the experiment section, we use three seeds $\{8, 9, 10\}$ to run the experiments and the final results are obtained by averaging these three runs. All the models are trained on NVIDIA V100 and P100.

A.3 Hyperparameters

Our framework has three hyperparameters including λ_1, λ_2 and λ_3. We set $\lambda_1 = 1$ for all the experiments. For λ_2 and λ_3, we follow the literature [3, 1, 2] and directly use the leave-one-domain-out cross-validation to select their values. We use grid search to find the values for λ_2 and λ_3. λ_2 is selected from $\{0.1, 0.01\}$ and λ_3 is selected from $\{1, 0.1, 0.01, 0.001, 0.0001\}$.

B License of existing assets

The existing datasets and codes used in this paper are publicly available. The licenses are listed as follows.

Datasets: Office-Home [14] is for non-profit academic research and education only. We cannot find the license for PACS [6] and VLCS [13].

Codes: AlexNet, ResNet18 and ResNet50 are pretrained with ImageNet in torchvision. They are under BSD 3-Clause License. U-net is under GNU General Public License v3.0.

C Domain-specific and domain-invariant features

To demonstrate the ability of our framework in capturing the domain-invariant features, we first compare our domain-specific classifier with the one proposed by Epi-FCR [7], and then visually illustrate the domain-specific and domain-invariant features learned by our framework.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

C.1 Comparison of domain-specific classifiers

The domain-specific classifier learns domain-specific features from a single source domain for classification. Epi-FCR [7] also introduces a domain-specific classifier that is trained by the classification loss of a source domain. However, this method cannot ensure that its domain-specific classifier would not use domain-invariant features for classification. Unlike Epi-FCR, besides minimizing the classification loss for each source domain, our domain-specific classifier also maximizes the classification uncertainty on the remaining source domains. The domain-specific classifier is therefore designed to only learn domain-specific features. In Table 1, we show the performance of using the domain-specific classifiers from Epi-FCR and our framework on PACS. The performance of Epi-FCR is obtained by training our encoder-decoder network and domain-invariant classifier with the domain-specific classifiers from Epi-FCR. We can see that the prediction performance using our domain-specific classifiers consistently outperforms that obtained by Epi-FCR, which shows that our domain-specific classifiers can better learn domain-specific features than the domain-specific classifiers from Epi-FCR.

Table 1: Prediction accuracy (%) on PACS with different domain-specific classifiers.

Method	**A**	**C**	**P**	**S**	Avg.
Epi-FCR [7]	64.32	71.97	88.03	71.38	73.92
LRDG (ours)	**72.01**	**73.12**	**89.50**	**74.86**	**77.37**

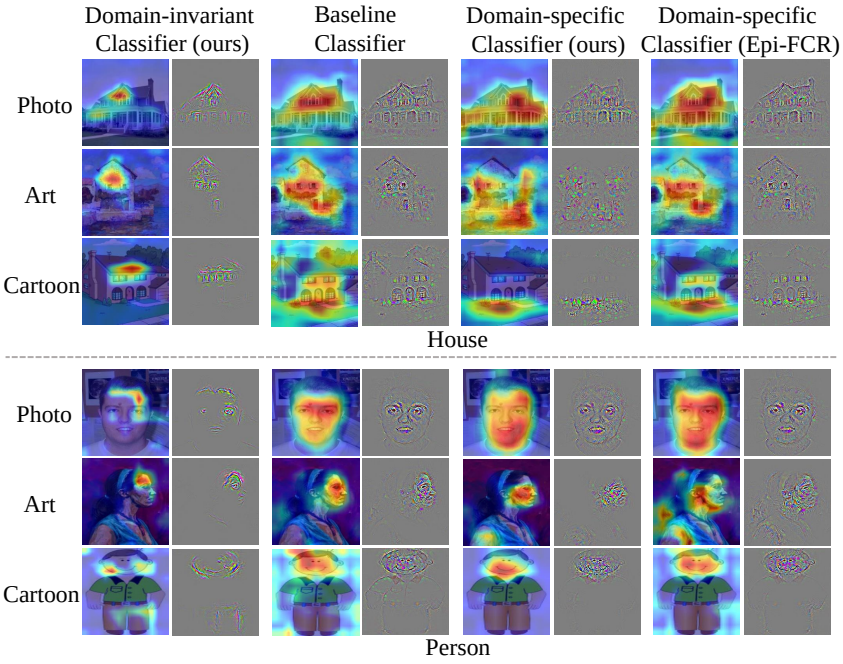

Figure 1: Grad-CAM and Guided Grad-CAM for House and Person on source domains with different methods. For each category (House or Person), three images from Photo (1st row), Art (2nd row), and Cartoon (3rd row) are shown with different methods. For each method, the left and right images are the visualization results of Grad-CAM and Guided Grad-CAM. *Domain-invariant Classifier (Ours)* is the domain-invariant classifier proposed in this paper. *Baseline Classifier* is the classifier obtained from the baseline method. *Domain-specific Classifier (Ours)* is the domain-specific classifier proposed in this paper. *Domain-specific Classifier (Epi-FCR)* is the domain-specific classifier used by Epi-FCR.

C.2 Visualization of domain-specific and domain-invariant features

To intuitively illustrate the domain-specific features and domain-invariant features, we show the Grad-CAM and Guided Grad-CAM [10] visualization of the images from House and Person categories in Fig. 1 and Fig. 2. Grad-CAM is a visualization technique that locates the important regions in the image for prediction. Guided Grad-CAM combines Grad-CAM with Guided backpropagation [12] to obtain a high-resolution gradient visualization. For this experiment, the source domains are Photo, Art painting, and Cartoon, and the target domain is Sketch.

In Fig. 1, we show the Grad-CAM and Guided Grad-CAM visualization of the images from the source domains obtained from four models including our domain-invariant classifier, the baseline classifier, our domain-specific classifier, and the domain-specific classifier from EPI-FCR. The baseline classifier is the baseline model that is trained by minimizing the cross entropy loss on all source domains. We compare these classifiers to show that they recognize different features for inference. Our domain-invariant classifier focuses on the features of triangular roofs or the top of the windows to recognize houses and locates the features from hairlines or head shapes to recognize a person. These features exist in all source domains, which can be treated as domain-invariant features. The baseline classifier focuses on the doors, windows, and backgrounds of houses and the whole face of a person. It captures both domain-specific and domain-invariant features. For example, backgrounds (e.g. grassland, trees, and flowers) and face details do not always exist in all source domains while head shapes belong to all source domains. The domain-specific classifier from Epi-FCR uses features that are similar to the baseline classifier and tends to use the domain-invariant features for classification. Our domain-specific classifier uses different features (e.g. grassland, trees, and flowers for House, and the lower faces for Person) that belong to the specific domains. It can better capture the domain-specific features than that from Epi-FCR.

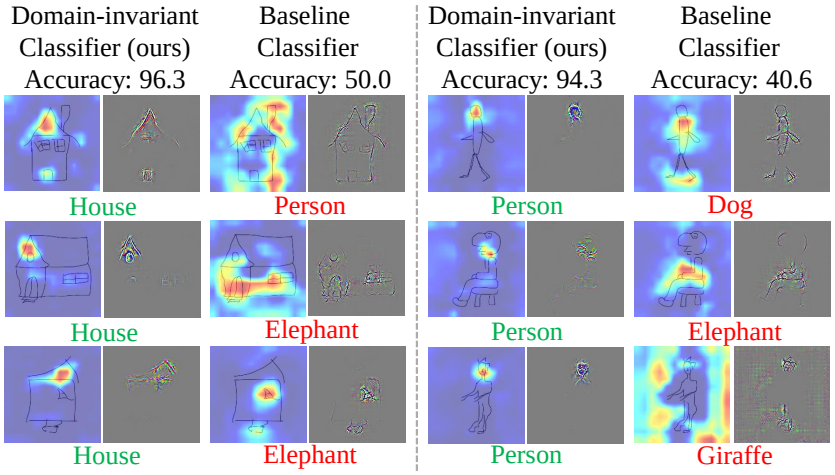

Figure 2: Grad-CAM and Guided Grad-CAM for House and Person on the target domain (Sketch) with different methods. For each category (House or Person), three images from Sketch are shown with different methods. For each method, the left and right images are the visualization results of Grad-CAM and Guided Grad-CAM. *Domain-invariant Classifier (Ours)* is the domain-invariant classifier proposed in this paper. *Baseline Classifier* is the classifier obtained from the baseline method. The images' classes predicted by each classifier are below each image. The prediction accuracy (%) is also shown.

Fig. 2 shows the Grad-CAM and Guided Grad-CAM visualization of the images from the target domain Sketch. We compare our domain-invariant classifier and the baseline classifier. The prediction accuracy of each classifier is also shown. Our domain-invariant classier can capture triangular roofs to recognize houses and head shapes to recognize persons, but the baseline classifier hardly extracts useful features to correctly identify objects. As illustrated in the figure, the baseline classifier categorizes the houses as Person and Elephant and classifies the persons as Dog, Elephant, and Giraffe. With our domain-invariant classier, the classification accuracy for House is increased by about 46% and the classification accuracy for Person is improved by almost 54%. This demonstrates

the advantage of our framework for learning domain-invariant features compared with the baseline classifier.

D Loss functions

In this experiment, we compare the performance of different loss functions for the uncertainty loss L_U and the reconstruction loss L_R. For the L_U, we evaluate two losses including the entropy loss that measures the entropy of the posterior probability of classification and the least likely loss [8] that aims to predict the least likely class. For the L_R, we evaluate three losses including l_1, l_2 and perceptual loss [5]. The l_1 or l_2 loss measures pixel-wise similarity while the perceptual loss measures semantic similarity between images. Johnson et al. [5] proposed two perceptual loss functions: feature reconstruction loss and style reconstruction loss. We only use the feature reconstruction loss because we aim to reconstruct the semantic features instead of the style of the images. To compute the perceptual loss, we use the VGG [11] pre-trained by ImageNet as the loss network [5]. Since the domain of ImageNet is different from the source domains, we first fine-tune the loss network with the source domains and further use it to compute the feature reconstruction loss.

Table 2: Prediction accuracy (%) on PACS with loss functions L_U and L_R. *EL*: entropy loss; *LLL*: least likely loss; l_2: l_2 reconstruction loss; l_1: l_1 reconstruction loss; *PL*: perceptual loss with the loss network pre-trained by ImageNet; *PL (src)*: perceptual loss with the loss network fine-tuned by the source domains.

L_U,L_R	**A**	**C**	**P**	**S**	Avg.
EL, l_2	**72.01**	73.12	**89.50**	**74.86**	**77.37**
EL, l_1	67.44	**74.11**	88.93	74.65	76.28
EL, PL	70.12	71.43	88.01	73.49	75.76
EL, PL (src)	68.36	72.13	88.33	74.58	75.85
LLL, l_2	68.37	71.24	87.76	73.31	75.17

In Table 2, we show the prediction accuracy of these loss functions on PACS with AlexNet backbone. As shown in the table, the entropy loss consistently achieves better performance than the least likely loss. The performance of the l_1 and l_2 reconstruction loss is comparable, but the latter has better average accuracy. The performance of the perceptual loss is worse than that of the l_2 reconstruction loss. Even though the loss network is fine-tuned by the source domains, the performance of the perceptual loss shows no improvement. Overall, we use the entropy loss and the l_2 reconstruction loss as the default loss functions.

References

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1006–1016, 2018.

[2] 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, 2021.

[3] 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.

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

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[6] 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.

[7] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1446–1455, 2019.

[8] Matthias Minderer, Olivier Bachem, Neil Houlsby, and Michael Tschannen. Automatic shortcut removal for self-supervised representation learning. *International Conference on Machine Learning*, 2020.

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[10] 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.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[12] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *ICLR (workshop track)*, 2015.

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

[14] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.