Single Domain Generalization via Normalised Cross-correlation Based Convolutions

WeiQin Chuah[1*] Ruwan Tennakoon[1*] Reza Hoseinnezhad[1] David Suter[2] Alireza Bab-Hadiashar[1]
RMIT University, Australia[1] Edith Cowan University (ECU), Australia[2]
{wei.qin.chuah,ruwan.tennakoon,rezah,abh}@rmit.edu.au, d.suter@ecu.edu.au

In this supplementary document, we provide detailed information regarding the implementation of our approach for different benchmarks in Section 1. Additionally, in Section 2.5, we present a comprehensive comparison of domain generalization performance on the Camelyon-17-Wilds benchmark, evaluating multiple models including Convolution Neural Networks (CNN), CNN trained with Random Convolution, our XCNorm, R-XCNorm, and R-XCNorm trained with Random Convolution.

1. Experimental and Implementation Details

We built our proposed XCNorm & R-XCNorm networks by replacing the convolution and dense layers in the corresponding CNN architectures (LeNet for Digits, WideResNet for CIFAR-10-C, and ResNet-18 for Camelyon-17 and DomainNet) with either XCNorm or R-XCNorm layers that have the same number of channels and kernel size. Sharpening and NBAM were applied to all layers except the last layer. The last layer consists of XCNorm or R-XCNorm with Sharpening and Gradient scaling. The remaining hyper-parameters including learning rate, and number of iterations were tuned using hold-out validation data from the source domain. The architecture of our proposed XCNorm& R-XCNorm networks was based on their respective convolution neural network counterpart mentioned above. To facilitate fairer comparisons, we replaced convolution and dense layers in CNN with either XCNormor R-XCNormlayers with the same number of channels and kernel size. The activation functions (e.g. ReLU) and normalization layer (e.g. Batch Normalization) between layers were removed from the original network architecture.

1.1. Digits-DG

Experimental Setup: In line with previous works on S-DG [3–6], we adopted the same experimental setup. Specifically, we resized all training images to 32×32 and converted them from grayscale to RGB format by duplicating the channels. We also adopted the LeNet architecture as the backbone network and trained it with a batch size of 32.

1.2. CIFAR-10-C

Experimental Setup: Our experiments were conducted using the CIFAR-10 training set, comprising $50,000$ images for model training. The trained models were evaluated on the corrupted test set, which consisted of $10,000$ images. We utilized the widely adopted WideResNet (16-4) architecture as our backbone network, following prior works. Model optimization was performed for 100 epochs using the Adam optimizer with a batch size of 128. The initial learning rate was set to $1e^{-3}$, and a cosine annealing scheduler was employed.

1.3. Camelyon-17

Experimental Setup: In line with the WILDS benchmark [1], we consider each medical center as an individual domain. Our experimental procedure involves training models on a single domain and evaluating their performance on the remaining domains. The source domain data is divided into training and validation sets using an 80/20 splitting ratio. To evaluate domain generalization performance, we utilize the AUROC metric. Following the approach outlined in [8], we employ the ResNet-18 architecture as our network backbone and set the batch size to 128. All images are resized to 96×96 and data augmentation techniques, such as random horizontal flipping and random rotation, are applied during training. The models are trained for 10 epochs using the Adam optimizer, with an initial learning rate of $1e^{-3}$ and decayed using a cosine annealing scheduler.

1.4. DomainNet

Experimental Configuration: Our experimental methodology entails training models on the *real* domain and subsequently gauging their performance across the remaining domains. For the source domain data, an $80/20$ split ratio is adopted to segregate it into training and validation

[*]WeiQin Chuah and Ruwan Tennakoon assert equal contributions.

Domain	None	Batch	Instance	XCNorm	R-XCNorm
MNIST-M	57.25	34.00	55.25	66.67	71.00
SVHN	31.17	11.25	34.08	59.92	65.42
SYN	39.50	25.92	41.00	67.33	71.25
USPS	77.33	74.19	76.23	85.24	89.04

Table 1. Out-of-domain performance (accuracy %) comparison between standard normalization methods and our proposed methods. The *None* model denotes a model without any applied normalization.

Dataset	Baseline	XCNorm	R-XCNorm
MNIST	98.56	98.79	99.17
CIFAR	91.40	90.87	91.23

Table 2. In-domain performance (accuracy %) comparison between baseline model and our proposed XCNorm and its robust variant R-XCNorm.

subsets. In assessing single domain generalization performance, we employ the top-1 accuracy metric. We selected the ResNet-18 architecture as our network backbone and configured the batch size at 64. Image dimensions are uniformly resized to 224×224, and solely random horizontal flipping is applied for data augmentation during the training phase. Model training spans 100 epochs, facilitated by the Adam optimizer with an initial learning rate of $1e^{-3}$. Notably, all models are trained from scratch without reliance on pre-trained ImageNet weights. Additionally, to circumvent models overfitting to classes with a sparse image presence, training is confined to the top 200 classes boasting at least 500 images.

1.5. Random Convolution

We adopt the same implementation of Random Convolution from the original work by Xu *et al.* [7]. Random Convolution is defined as follows:

$$\texttt{RandConv}(I, k, \alpha) = \alpha I + (1-\alpha)(I * \Theta) \quad (1)$$

where $I \in \mathbb{R}^{3,H,W}$ represents the input image with spatial dimensions H and W, $k \in \mathbb{R}$ denotes the kernel size, and α is the mixing weight uniformly sampled from the range $[0, 1]$. The convolution weights Θ are randomly sampled from a Gaussian distribution $\mathcal{N}(0, \frac{1}{3k^2})$. We sample k uniformly from a pool $\mathcal{K} \in [1, 3, ..., n]$, where n represents the maximum kernel size.

For the Digits-DG and CIFAR-10-C experiments, we set n to 5 to preserve finer shapes in low-resolution images, such as 32×32. In the case of the Camelyon-17 dataset, we set n to 7.

Dataset	BatchSize	ImageSize	Baseline	XCNorm	R-XCNorm
MNIST	32	32×32	7.62	11.4	12.7
CIFAR-10	128	32×32	31.4	71.2	124
Camelyon-17	128	96×96	27.8	69.8	111

Table 3. Training speed (second per epoch) comparison for all selected datasets and benchmark with different batch sizes and image sizes.

Method	Clipart	Inforgraph	Painting	Quickdraw	Sketch	Average
Baseline	26.73	6.24	22.86	5.46	18.70	16.00
XCNorm	27.07	6.05	23.30	5.24	19.38	16.21
R-XCNorm	28.81	7.19	25.61	5.83	19.70	17.43

Table 4. Comparison of model performance on the DomainNet dataset for single domain generalization. All methods were trained on the real domain and the remaining domains are set as the out-of-domain test sets.

2. Additional Experimental Results

2.1. Performance Comparison: XCNorm vs. Standard Normalization Techniques

In Table 1, we present a comprehensive comparison of single domain generalization performance between XCNorm, its robust variant (R-XCNorm), and widely used standard normalization techniques including Batch Normalization (BN) and Instance Normalization (IN). This evaluation is conducted on the Digits-DG benchmark dataset.

As demonstrated in Table 1, it becomes evident that BN significantly compromises OOD performance, whereas IN showcases minimal influence in comparison to the baseline (absence of any normalization). In striking contrast, our proposed approach attains notably superior performance results.

2.2. In-domain Performance Comparison

As demonstrated in Table 2, a notable similarity in performance is evident across the MNIST and CIFAR-10 datasets among the baseline model, our method, and its robust variant. Notably, our method enhances single-domain generalization performance without compromising within-domain efficacy.

2.3. Training Time Comparison

In Table 3, a comparison of the training time per epoch is presented, encompassing the baseline model, our proposed XCNorm, and its robust iteration R-XCNorm. This analysis is conducted across selected datasets, namely MNIST, CIFAR-10, and Camelyon-17. As evident in Table 3, both XCNorm and R-XCNorm necessitate considerably greater computational resources compared to the baseline model. Nevertheless, this increase of computational time is judiciously balanced by the substantial improvement in out-of-domain (OOD) performance, rendering this trade-off no-

Figure 1. Visual comparison of the ambulance class across Real (top), Clipart (middle) and Infograph (bottom) domains.

tably advantageous.

2.4. Comparisons on DomainNet

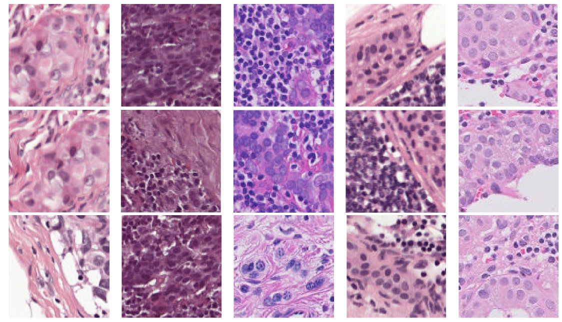
In Table 4, we compare the single domain generalization performance between the baseline model, our proposed XCNorm and its robust variant R-XCNorm on the DomainNet benchmark [2]. The implementation details of this experiment are included in Section 1.4 of this supplementary document.

Our XCNorm demonstrates a slight performance enhancement compared to the baseline across the clipart, painting, and sketch domains. However, our method suffers from a minor performance dip in other domains, namely infograph and quickdraw. We attribute this decline to the additional information present in infographics, such as text and various objects, as well as the substantial contextual divergence between real and quickdraw domains (see Figure 1 for a visual comparison example). In contrast, our R-XCNorm consistently amplifies single domain generalization performance over the baseline across all test domains, surpassing the baseline by approximately 9%, as shown in Table 4.

2.5. Comprehensive Results on Camelyon-17

In Table 5, we present a comprehensive comparison of domain generalization performance on the Camelyon-17 benchmark using various models, including Convolutional Neural Network (CNN), CNN with Random Convolution (RC), our proposed XCNorm, its robust variant (R-XCNorm), and the robust variant trained with Random Convolution.

The CNN model demonstrates poor generalization when trained on a single source domain, particularly in domains 2 and 3. We attribute this to significant variations in the staining agent's color across different hospitals, as illustrated in Figure 2. However, by applying the Random Convolution augmentation, we observe a notable improvement in performance from 61.9% to 83.7%, as shown in Table 5.



(a) (b) (c) (d) (e)

Figure 2. Examples of histopathology data collected from five distinct medical centers, each representing a unique domain within the Camelyon-17 benchmark. The domains are labelled as (a)–(e), corresponding to domains 1–5, respectively. These examples showcase the variations in the staining agent's color and tissue characteristics across different domains, highlighting the challenges of domain generalization in histopathology image classification.

Significantly, our proposed XCNorm and R-XCNorm consistently outperform both the CNN model and the CNN model trained with Random Convolution, without relying on data augmentation techniques. Furthermore, when combined with augmentation methods such as Random Convolution, our approach (R-XCNorm + RC) achieves a robust model with impressive domain generalization performance.

References

[1] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021. 1

[2] 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 International Conference on Computer Vision*, pages 1406–1415, 2019. 3

[3] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565, 2020. 1

[4] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018. 1

[5] Chaoqun Wan, Xu Shen, Yonggang Zhang, Zhiheng Yin, Xinmei Tian, Feng Gao, Jianqiang Huang, and Xian-Sheng Hua. Meta convolutional neural networks for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4682–4691, 2022. 1

[6] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 834–843, 2021. 1

[7] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *International Conference on Learning Representations*, 2021. 2

[8] Nanyang Ye, Kaican Li, Haoyue Bai, Runpeng Yu, Lanqing Hong, Fengwei Zhou, Zhenguo Li, and Jun Zhu. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7947–7958, 2022. 1

Train Domain	Test Domain 1	2	3	4	5	AVG OOD	Overall OOD
Convolution							
1	-	94.2	69.2	71.6	75.5	77.6	61.9
2	55.8	-	83.6	38.4	36.8	53.7	
3	60.3	49.2	-	34.8	59.1	50.8	
4	83.0	74.2	26.3	-	64.8	62.1	
5	86.1	59.1	60.5	55.4	-	65.3	
Convolution + Random Convolution							
1	-	87.1	95.6	89.9	90.2	90.7	83.7
2	82.4	-	67.3	74.7	68.8	73.3	
3	96.9	81.7	-	86.3	97.3	89.7	
4	89.1	86.3	80.2	-	89.8	86.4	
5	89.5	71.8	80.5	71.2	-	78.2	
XCNorm							
1	-	91.4	90.9	94.9	93.0	92.5	89.3
2	88.9	-	72.9	89.1	82.3	83.3	
3	93.7	89.4	-	93.6	93.1	92.5	
4	92.7	92.5	88.9	-	68.3	85.6	
5	93.3	91.1	93.7	92.7	-	92.7	
R-XCNorm							
1	-	93.1	92.5	95.7	94.5	94.0	90.8
2	87.0	-	80.3	90.5	81.7	84.9	
3	94.6	89.5	-	93.8	94.9	93.2	
4	93.7	94.8	88.5	-	73.3	87.6	
5	95.2	92.3	96.4	93.5	-	94.4	
R-XCNorm + Random Convolution							
1	-	94.6	94.4	97.1	94.1	95.0	93.3
2	90.6	-	84.6	93.7	87.3	89.0	
3	96.3	93.3	-	97.0	96.9	95.9	
4	94.1	93.6	92.6	-	84.8	91.3	
5	95.8	93.9	95.7	96.5	-	95.5	

Table 5. A comprehensive comparison of model performance on the Camelyon-17 dataset for single domain generalization. The included models are Convolution Neural Network (CNN), CNN trained with Random Convolution, our XCNorm, R-XCNorm, and R-XCNorm trained with Random Convolution.