Supplementary Materials: Progressive Random Convolutions for Single Domain Generalization

Seokeon Choi Debasmit Das Sungha Choi Seunghan Yang Hyunsin Park Sungrack Yun
Qualcomm AI research†
{seokchoi,debadas,sunghac,seunghan,hyunsinp,sungrack}@qti.qualcomm.com

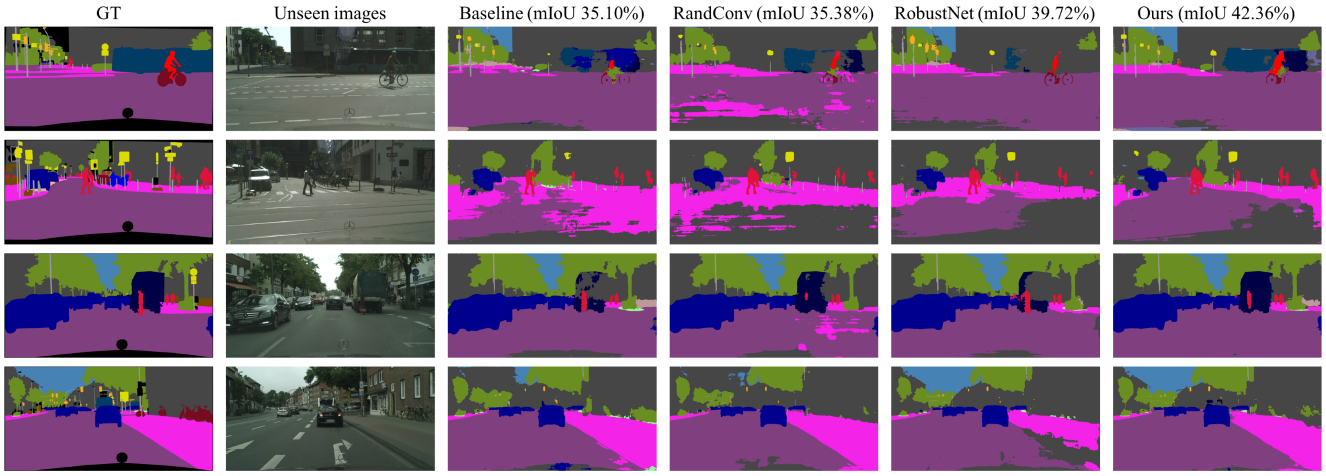

Figure 1. Segmentation results for unseen domain images. All models are trained on the GTAV [15] train set and validated on the Cityscapes [3] validation set. DeepLabV3+ is adopted as a baseline. Our method outperforms the baseline, RandConv [17], and RobustNet [2] methods.

1. Reproducibility

We have provided implementation details and pseudocode in the main paper for reproducibility. Note that all the experiments have been performed eight times and averaged.

2. Domain generalizable semantic segmentation

To show the applicability of Pro-RandConv, we conducted semantic segmentation experiments in addition to the object recognition experiments provided in the main paper. We use the experimental protocol used in RobustNet [2] for a fair comparison. We adopt a DeepLabV3+ [1] architecture with ResNet50 [8] as a baseline. We use the GTAV [15] dataset as the training domain and measure the generalization capability on the Cityscapes [3], BDD-100K [18], SYNTHIA [16], and Mapillary [11] datasets. Mean Intersection over Union (mIoU) is used to quantitatively evaluate semantic segmentation performance. We use a batch size of 8 for experiments on a single GPU, which is different from the experimental protocol in [2] using a batch size of 16 for GTAV. Except for the batch size, all environments are the same as the official experimental protocol, refer to [2] for more details. In our augmentation settings, we use all of the same hyperparameters for object recognition without additional tuning. We also randomly select only half of the images for each batch and perform augmentation.

Tabel 1 shows a comparison of generalization performance in semantic segmentation. To prove the superiority of Pro-RandConv, we compare the performance not only with RandConv [17] but also with RobustNet [2], a domain generalization method for semantic segmentation. Besides that, we compare the performance with various competitors (*e.g.* Switchable Whitening (SW) [13], IBN-Net [12], and IterNorm [9]) provided by [2]. Our method outperforms all of the competitors including RandConv and RobustNet by a big margin. In particular, we note that our method shows a great performance improvement on real-world datasets (*i.e.* Cityscapes, BDD-100K, and Mapillary). We also provide experimental results with various versions to observe the importance of each component. All components except the

† Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

Table 1. Performance comparison of mIoU (higher is better) (%). The models are trained on the train set of GTAV (G), and evaluated on Cityscapes (C), BDD-100K (B), SYNTHIA (S), and Mapillary (M) validation sets. ResNet-50 is used with an output stride of 16. DeepLabV3+ is adopted as a baseline. * denotes reproduced results. Ours-A is a version with only a progressive approach, Ours-B is a version with a progressive approach and contrast diversification, Ours-C is a version with a progressive approach, contrast diversification, and Gaussian smoothing, and Ours-D is a version with all components applied.

Methods	C	B	S	M	Avg.
Baseline [2]	28.95	25.14	26.23	28.18	27.13
SW [13]	29.91	27.48	27.61	29.71	28.68
IterNorm [9]	31.81	32.70	27.07	33.88	31.37
IBN-Net [12]	33.85	32.30	27.90	37.75	32.95
RobustNet [2]	36.58	35.20	**28.30**	40.33	35.11
Baseline*	35.10	27.18	26.71	30.63	29.91
RandConv* [17]	35.38	30.92	24.45	32.43	30.80
RobustNet* [2]	39.72	35.61	26.87	39.50	35.43
Ours-A	39.53	34.14	26.30	36.74	34.18
Ours-B	41.60	34.95	26.18	41.31	36.01
Ours-C	**42.36**	**37.03**	25.52	**41.63**	**36.64**
Ours-D	40.48	36.68	26.82	40.76	36.19

deformable offset improve the generalization performance, which can be interpreted as the geometrical change of the object shape from the deformable offset causing a negative effect on the pixel-level classification. We expect to get better generalization performance if we change the ground truth to accommodate geometric changes. Figure 1 describes the semantic segmentation results on Cityscapes. Ours-C version of the model with removed deformable offsets is used for visualization.

3. Strategies for selecting images to augment

In the main paper, we provided a performance on a basic learning strategy using both original images $\mathbf{X}_0$ and augmented images $\mathbf{X}_L$. Table 2 shows various data fraction methods to effectively use augmented data for training. RandConv [17] applied augmentation with half probability for every mini-batch. That is, sometimes the original images are used and at other times the augmented images are used for training. We call this a batch-level image augmentation strategy. We first compare these batch-level augmentation strategies using only the original images, using only the augmented images, and using both sets with half probability. Using only original images significantly degrades out-of-domain performance. On the other hand, using only augmented images degrades in-domain performance, especially in PACS. Therefore, it is important to properly combine the two types of images to balance in-domain and out-of-domain performance.

Next, we provide experiments on an instance-level augmentation strategy to learn both original and augmented images within a mini-batch. $P_r(\mathbf{X}_0, \mathbf{X}_L)$ indicates this strategy, where r is the data fraction of the original images. The larger r, the higher the proportion of the original images in the mini-batch. Generally, a high value of r tends to improve in-domain performance and decrease out-of-domain performance. The most appropriate solution is to set r to a value of 0.5 or to sample r from $U(0, 1)$. In particular, the random sampling strategy achieves satisfactory values for both in-domain performance and out-of-domain performance, and obtains comparable performance to the basic strategy using both original and augmented images. It is noteworthy that RandConv degrades the in-domain performance on PACS from 95.37% to 92.75% compared to the baseline, whereas our method improves both in-domain and out-of-domain performance.

Table 2. Strategies for selecting training images in the single domain generalization setting on Digits and PACS in terms of accuracy (%). LeNet and ResNet18 are used for training on Digits and PACS, respectively. RC* denotes the reproduced results of RandConv. $\mathbf{X}_0$ and $\mathbf{X}_L$ indicate original images and augmented images passing through L-layers. L is sampled as $L \sim U(1, L_{max} = 10)$ for each mini-batch, respectively. $P_r(\mathbf{X}_0, \mathbf{X}_L)$ means an instance-level augmentation strategy, where r is the data fraction of the original images. The larger r, the higher the proportion of the original images in the mini-batch.

Methods	Selection strategies	Digits		PACS	
		In-domain	Out-of-domain	In-domain	Out-of-domain
RC* [17]	$\mathbf{X}_0$ or $\mathbf{X}_1$	98.90	74.84	92.75	67.50
Ours (batch)	only $\mathbf{X}_0$ (baseline)	98.64	52.00	95.37	63.15
	only $\mathbf{X}_L$	99.25	80.99	94.66	68.10
	$\mathbf{X}_0$ or $\mathbf{X}_L$	99.25	81.08	95.59	67.65
Ours (instance)	$P_{r=0.25}(\mathbf{X}_0,\mathbf{X}_L)$	99.28	81.20	95.18	68.43
	$P_{r=0.50}(\mathbf{X}_0,\mathbf{X}_L)$	**99.31**	81.13	95.65	68.20
	$P_{r=0.75}(\mathbf{X}_0,\mathbf{X}_L)$	99.25	80.22	95.73	67.26
	$P_{r\sim U(0,1)}(\mathbf{X}_0,\mathbf{X}_L)$	99.27	80.66	**96.00**	**69.11**
Ours ($\mathbf{X}_0$ and $\mathbf{X}_L$)		99.29	**81.35**	95.51	68.88

4. Component analysis

Table 3 shows a detailed performance comparison for each component of Pro-RandConv. First, we analyze whether we can improve performance by adding our components to the single-layer approach used in RandConv [17]. Gaussian smoothing of convolution weights does not have a significant effect in a single-layer approach, whereas contrast diversification and deformable offsets help to improve performance. However, it does not contribute to a significant performance improvement, because of the limitation of style diversity and the problem of excessive semantic distortion in the single-layer approach. In addition, the method of variously adjusting the variance of the Gaussian distribution without fixing the convolution weight to He-initialization [7] shows some performance improvement on Digits.

Second, we analyze the influence of components in detail under our progressive approach. The key to the progressive approach is to initialize one layer and keep the remaining

Table 3. Performance analysis for detailed components in terms of accuracy (%). LeNet and ResNet18 are used for training on Digits and PACS, respectively. SDG and MDG indicate single domain generalization and multi domain generalization settings, respectively. *Single* denotes the single-layer approach used by RandConv. *Multi* (D/S) represents our progressive approach, where D means to initialize all layers differently, and S means to initialize one layer and use it equally for all layers.

Model	Conv. smooth	Contrast	Offsets	Digits	PACS	
				SDG	SDG	MDG
Baseline	-	-	-	52.00	63.13	81.45
Single	-	-	-	74.84	67.50	82.43
	✓	-	-	74.34	67.95	82.53
	-	✓	-	77.14	67.81	83.16
	-	-	✓	75.73	67.24	82.38
	$w \sim N(0, \sigma_w), \sigma_w \sim U(\epsilon, 1)$			76.59	67.46	82.53
	$w \sim N(0, \sigma_w), \sigma_w \sim U(\epsilon, 2)$			75.80	67.99	82.50
Multi (D)	-	-	-	74.72	66.47	82.49
Multi (S)	-	-	-	78.26	67.89	83.72
	-	✓	-	77.09	68.73	84.17
	-	-	✓	77.41	68.25	84.04
	-	✓	✓	77.08	**69.01**	84.24
	✓	-	-	80.03	68.3	83.79
	✓	✓	-	80.02	68.55	84.22
	✓	-	✓	81.06	67.98	83.77
	✓	✓	✓	**81.35**	68.88	**84.29**

layers with the same parameters, which leads to a significant performance improvement. Next, we compare the performance with and without Gaussian smoothing of the convolution layer. In Digits, since the size of the object is relatively small, the multi-layer structure of the 3×3 convolution layer has excessive diversity. Thus, increasing the contrast and texture diversity without Gaussian smoothing has the effect of inducing semantic distortion. In other words, it is more effective to secure the contrast and texture diversity while controlling the deformation scale of texture with Gaussian smoothing. Conversely, in PACS, since the resolution of the image is large, the multi-layer structure of the 3×3 convolution layer is inefficient in diversity. Therefore, even if Gaussian smoothing is not applied, the generalization capability can be improved by contrast diversification and deformable offsets.

5. Additional performance analysis

5.1. Comparison with traditional augmentation

In this section, we compare the traditional augmentation methods with our Pro-RandConv. Table 4 and Table 5 provide performance comparisons on Digits and PACS, respectively. In both datasets, *color jitter* and *grayscale* are more effective than *perspective* and *rotate* in terms of improving generalization ability. Also, AutoAugment [4] and RandAugment [5], which apply various augmentation types simultaneously, enhance domain generalization capability more than single augmentation methods. Furthermore, the proposed Pro-RandConv outperforms all these augmentation methods with a simple random network structure. Thanks to its effective generalization capability, we argue that the proposed Pro-RandConv could be a strong baseline for various tasks.

Table 4. Performance comparison with traditional augmentation techniques in the single domain generalization setting on Digits in terms of accuracy (%). Each column title indicates the target domain. LeNet is used for training. * denotes reproduced results.

Methods	SVHN	MNIST-M	SYN	USPS	Avg.
Baseline	32.52	54.92	42.34	78.21	52.00
Color jitter*	36.04	57.56	43.94	77.76	53.83
Grayscale*	32.92	55.44	42.38	78.22	52.24
Pespective*	33.63	43.86	40.92	69.12	46.88
Rotate*	31.99	54.86	38.22	69.54	48.65
AutoAugment [4]	45.23	60.53	64.52	80.62	62.72
RandAugment [5]	54.77	74.05	59.60	77.33	66.44
Ours	**69.67**	**82.30**	**79.77**	**93.67**	**81.35**

Table 5. Performance comparison with traditional augmentation techniques in the single domain generalization setting on PACS in terms of accuracy (%). Each column title indicates the source domain. ResNet18 is used for training. * denotes reproduced results.

Methods	Art	Cartoon	Photo	Sketch	Avg.
Baseline	74.64	73.36	56.31	48.27	63.15
Color jitter*	75.94	76.56	59.27	50.24	65.50
Grayscale*	74.29	75.75	58.96	47.67	64.17
Pespective*	72.29	70.17	59.99	43.79	61.31
Rotate*	73.47	71.06	56.95	46.61	62.02
AutoAugment* [4]	76.48	77.09	60.99	52.46	66.76
RandAugment* [5]	76.76	78.00	62.09	56.40	68.31
Ours	**76.98**	**78.54**	**62.89**	**57.11**	**68.88**

Table 6. Performance comparison on Digits in detail for a fair comparison (%). In MNIST-M, two different kinds of sets (A/B) are utilized. LeNet is used for training. Ours^{-T} and Ours^{-C} indicate disabling texture diversification and contrast diversification, respectively. RC denotes the official results of RandConv.

Methods	SVHN	MNIST-M (A/B)	SYN	USPS	Average (A/B)
RC [17]	57.52	- / 87.76	62.88	83.36	- / 72.88
Ours^{-T}	62.76	74.52 / 81.91	78.07	93.01	77.09 / 78.94
Ours^{-C}	**70.35**	**82.98 / 88.34**	77.40	93.52	81.06 / 82.40
Ours	69.67	82.30 / 87.72	**79.77**	**93.67**	**81.35 / 82.72**

5.2. Fair comparison on MNIST-M

We confirmed that RandConv uses the test set of MNIST-M [6] differently from the existing methods (*e.g.* PDEN [10], M-ADA [14], and ME-ADA [19]). Existing methods use MNIST-M consisting of 9,001 images, which we refer to as set A. RandConv uses MNIST-M which consists of 10,000 images, which we refer to as set B. For a fair comparison, we compare the performance of both MNIST-M sets. Table 6 shows that performance comparison on two sets of MNIST-M. We emphasize that our Pro-RandConv method has higher generalization capability than RandConv [17] in all domains including MNIST-M.

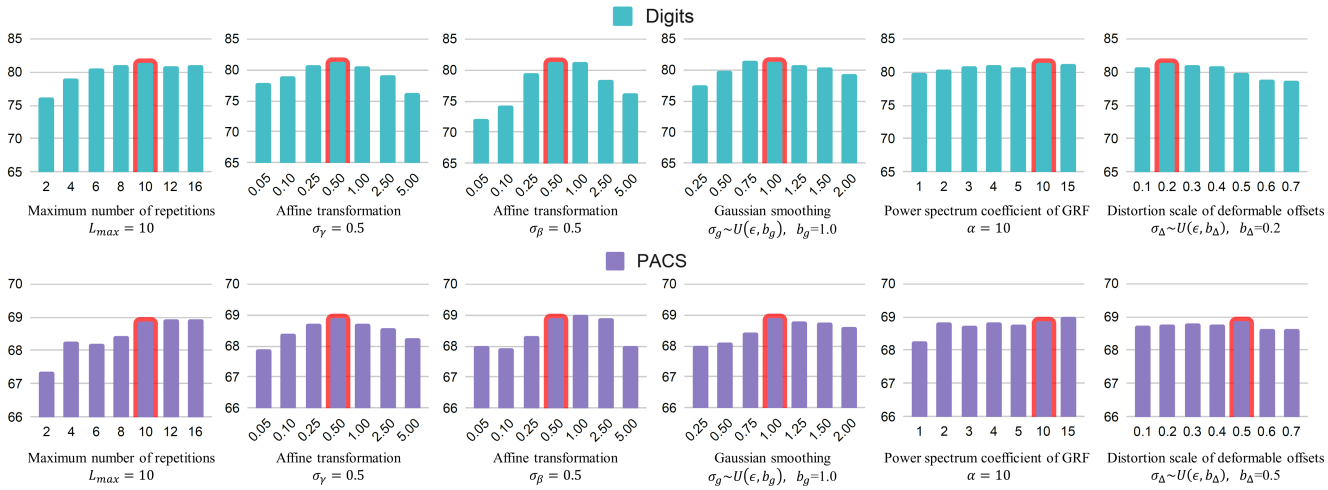

Figure 2. Analysis of hyperparameter selection in the single domain generalization setting on Digits and PACS.

6. Hyperparameter selection

6.1. Hyperparameters of the progressive approach

The core idea of this paper is a progressive method that initializes a random convolution layer once and then stacks it multiple times with the same structure. Eventually, from a hyperparameter selection perspective, RandConv's traditional approach of choosing the kernel size changes to choosing the number of repetitions of the convolution layers. For example, RandConv generates random-style images based on a kernel size randomly selected from $\{1, 3, 5, 7\}$ for each mini-batch. In a similar way, we choose a different number of repetitions with uniform sampling from 1 to L_{max} for each mini-batch. Figure 1(c) and 2(a) in the main paper show that as the kernel size increases, images augmented by RandConv easily lose their semantics and eventually the performance degrades rapidly. The progressive approach, on the other hand, is less sensitive to increasing L_{max}, since the performance does not degrade significantly as the receptive field increases, as shown in Fig 2. However, the computational cost increases proportionally to the number of repetitions, so we chose a reasonable value of 10 to account for the tradeoff.

6.2. Hyperparameters of convolution blocks

We further provide a performance comparison for all hyperparameters in the random convolution block, as shown in Fig. 2. We first analyze the hyperparameters for contrast diversification. We chose σ_γ and σ_β to be 0.5, as they show the highest performance on both Digits and PACS datasets. This means that the affine transformation parameters, γ and β, are sampled from $N(0, 0.5^2)$. Figure 7(a) and (b) in the main paper show that γ and β can cause false distortion or saturation if they are smaller or larger than 0.5, so we recommend keeping them at 0.5 regardless of the dataset.

Next, we analyze the hyperparameters for the convolution weights. The convolution weights are initialized by [7] as in RandConv (i.e., $\sigma_w = 1/\sqrt{k^2 C_{in}} = 1/\sqrt{3^3}$). We further apply Gaussian smoothing to this kernel. For Gaussian smoothing $g[i_m, j_m] = \exp(-\frac{i_m^2 + j_m^2}{2\sigma_g^2})$, the smoothing scale is sampled from $\sigma_g \sim U(\epsilon, b_g)$, where ϵ indicates a small value. This means that σ_g is randomly sampled for each mini-batch, so the smoothing effect is different each time. This technique can be used to mitigate the problem of severely distorted object semantics when the random offset of the deformation convolution is too irregular and large in scale. We chose b_g to be 1.0 because it performs best on both Digits and PACS datasets. As with the hyperparameter selection for contrast diversification, we set the same value for all datasets.

Finally, we introduce hyperparameters for deformable convolution that further enhance texture diversity. The tensor for deformable offsets consists of $(2k^2, H, W)$, where k is the kernel size of the convolution layer, and H and W are the height and weight of an image, respectively. That is, there are a total of $2k^2$ offsets per pixel in the image of $H \times W$, where 2 means the values of Δi_m and Δj_m. To induce natural geometric variation, we consider spatial correlation by generating a total of $2k^2$ Gaussian Random Fields (GRF) with a size of $H \times W$. We refer to this code[1] for the GRF implementation, where spatial correlation can be controlled by varying the coefficient α of the power spectrum. As shown in Fig. 7(f) of the main paper, the larger the coefficient α, the higher the spatial correlation. We scaled the Gaussian random field (GRF) by choosing a coefficient of 10 for the power spectrum. Another hyperparameter is the distortion scale σ_Δ of the deformable offset. In particular,

[1] https://github.com/bsciolla/gaussian-random-fields

geometric information such as rotation is an important attribute for digits recognition, so severe deformation impairs class-specific semantic information. Figure 7(e) in the main paper shows that the shape of the object becomes unrecognizable as the scale increases. This hyperparameter is also related to the size of the image, so we choose different hyperparameters according to image size. For Digits, a small scale of 0.2 is used, while for PACS, OfficeHome, and VLCS, a scale of 0.5 is used. As with the other hyperparameters, uniform sampling is performed as $U(\epsilon, b_\Delta)$ to make it less sensitive to hyperparameter selection.

References

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1

[2] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11580–11590, 2021. 1, 2

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1

[4] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 3

[6] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 3

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 2, 4

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[9] Lei Huang, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Iterative normalization: Beyond standardization towards efficient whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4874–4883, 2019. 1, 2

[10] Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 224–233, 2021. 3

[11] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017. 1

[12] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 464–479, 2018. 1, 2

[13] Xingang Pan, Xiaohang Zhan, Jianping Shi, Xiaoou Tang, and Ping Luo. Switchable whitening for deep representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1863–1871, 2019. 1, 2

[14] 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. 3

[15] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 1

[16] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 1

[17] 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. 1, 2, 3

[18] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 1

[19] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3