Progressive Random Convolutions for Single Domain Generalization

Seokeon Choi[1], Debasmit Das[2], Sungha Choi[1], Seunghan Yang[1], Hyunsin Park[1], Sungrack Yun[1]

[1]Qualcomm Korea YH
[2]Qualcomm Technologies, Inc.

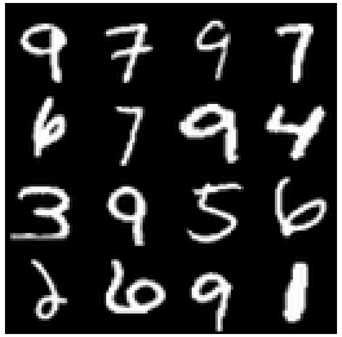
Presenter: Seokeon

Engineer, Senior

Overview

Progressive Random Convolution for Single Domain Generalization

- Deep neural networks often struggle to generalize to <u>out-of-distribution data.</u>
- We propose a <u>**simple and lightweight image augmentation technique**</u> based on <u>Progressive Random Convolutions.</u>

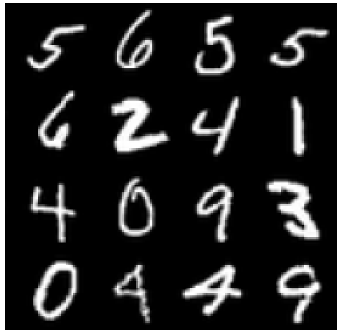

<Training set: X_{train}>

Deep Neural Network

Progressive Random Convolutions (**Pro-RandConv**)


<Augmented set: X_{aug}>

Out-of-distribution data


<Test set: A>


<Test set: B>


<Test set: C>

	Test set: A	Test set: B	Test set: C
Classification accuracy when learning with X_{train}	98.6%	32.5%	42.3%
Classification accuracy when learning with X_{train} & X_{aug}	98.6+0.7%	32.5+37.2%	42.3+37.5%

Motivation

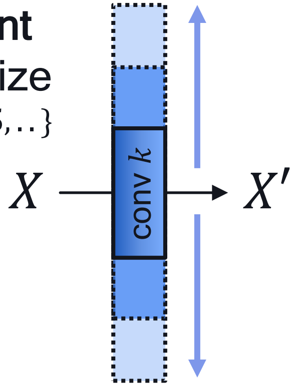
Progressive Random Convolution for Single Domain Generalization

Random Convolutions (ICLR'21)
- A single convolution layer (**randomly initialized**) ➔ Structural limitations

RandConv (ICLR'21)

Input image
$[C_{in} \times 32 \times 32]$

Different
kernel size
$k \in \{1, 3, 5, ..\}$



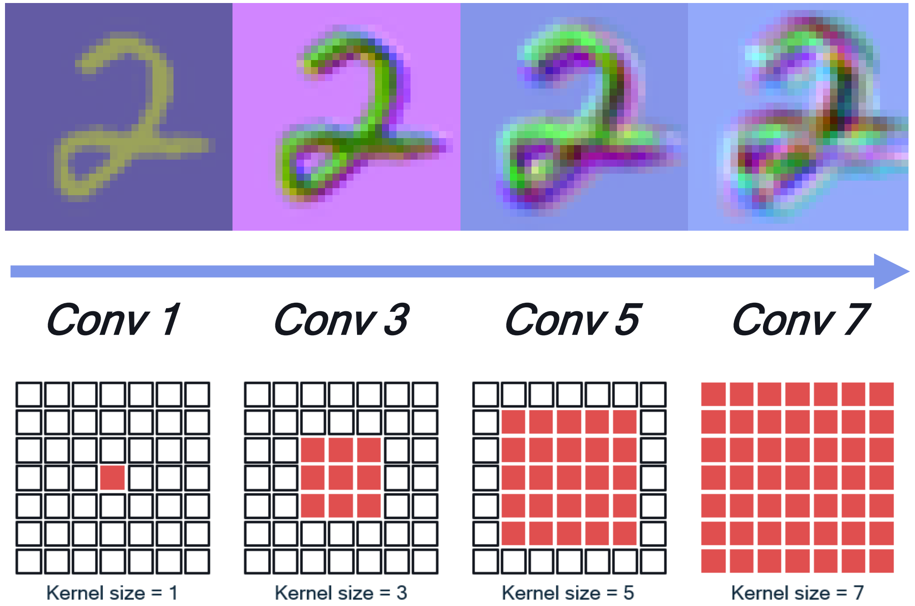
Conv k: $w \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$

Randomly sampled from Gaussian distribution

$$w \sim N\left(0, \frac{1}{k^2 C_{in}}\right)$$

Augmented image
$[C_{out} \times 32 \times 32]$

Properties
- Similar global shapes
- Random local textures

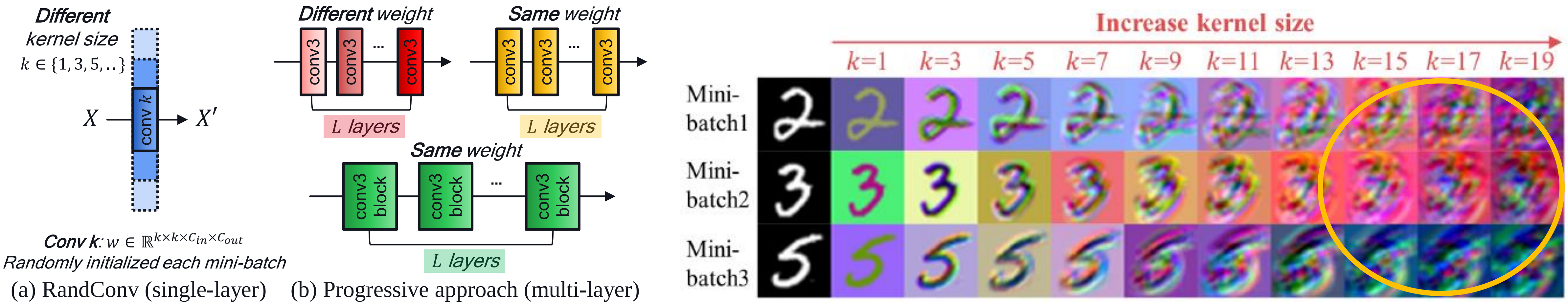


Xu, Zhenlin, et al. "Robust and Generalizable Visual Representation Learning via Random Convolutions." ICLR'21

Motivation

Progressive Random Convolution for Single Domain Generalization

Random Convolutions (ICLR'21)
- Structural limitations (Single convolution layer): the problems of limited diversity and semantic distortion



(a) RandConv (single-layer) (b) Progressive approach (multi-layer)

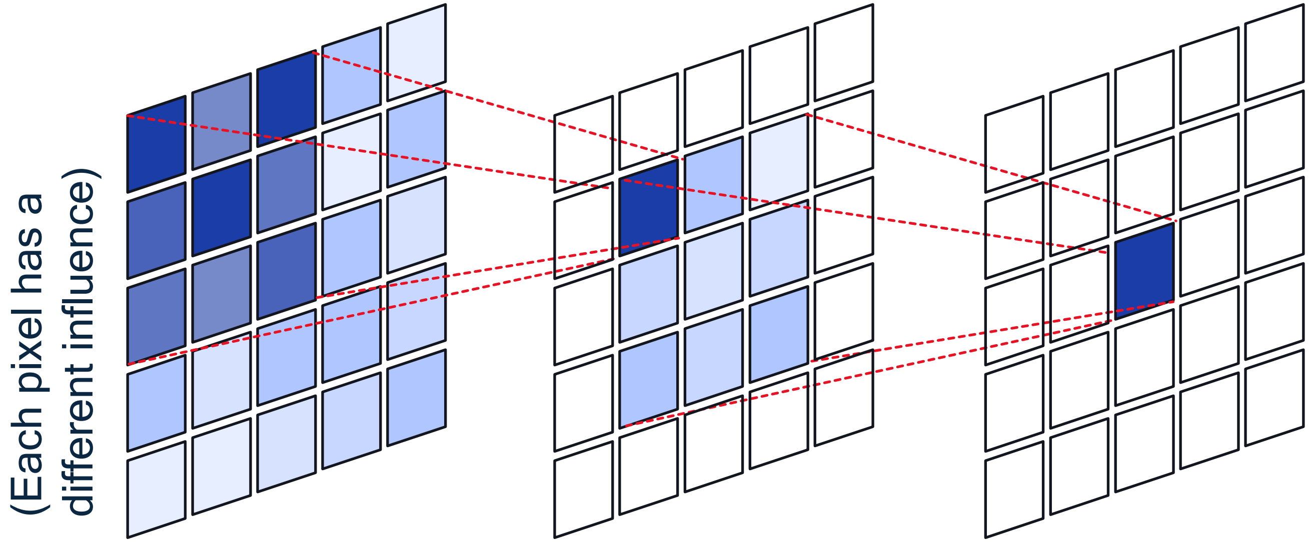
Limitations
- Artificial patterns
- Semantic distortion

Proposed method

Progressive Random Convolution for Single Domain Generalization

Main contribution 1: Progressive approach

Effective Receptive Fields (ERF): how much each input pixel can influence one output pixel

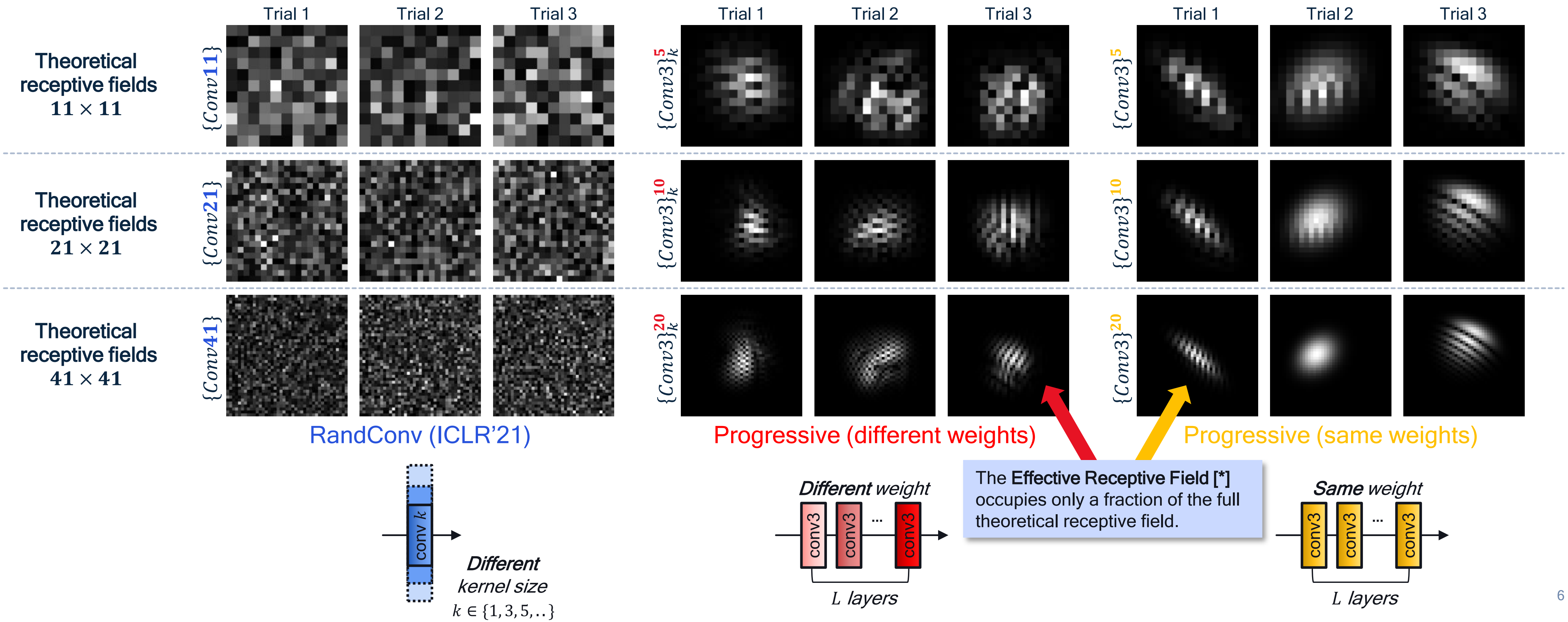


Luo, Wenjie, et al. "Understanding the effective receptive field in deep convolutional neural networks." NeurIPS’16

Proposed method

Progressive Random Convolution for Single Domain Generalization

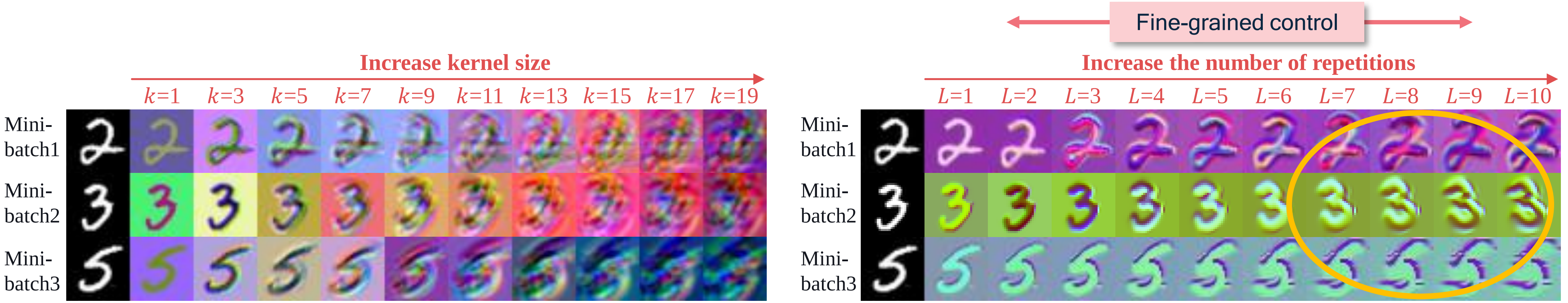
Main contribution 1: Progressive approach



Proposed method

Progressive Random Convolution for Single Domain Generalization

Main contribution 1: Progressive approach



(a) Examples of images augmented by RandConv

(b) Examples of images augmented by the proposed Pro-RandConv

RandConv (ICLR'21)

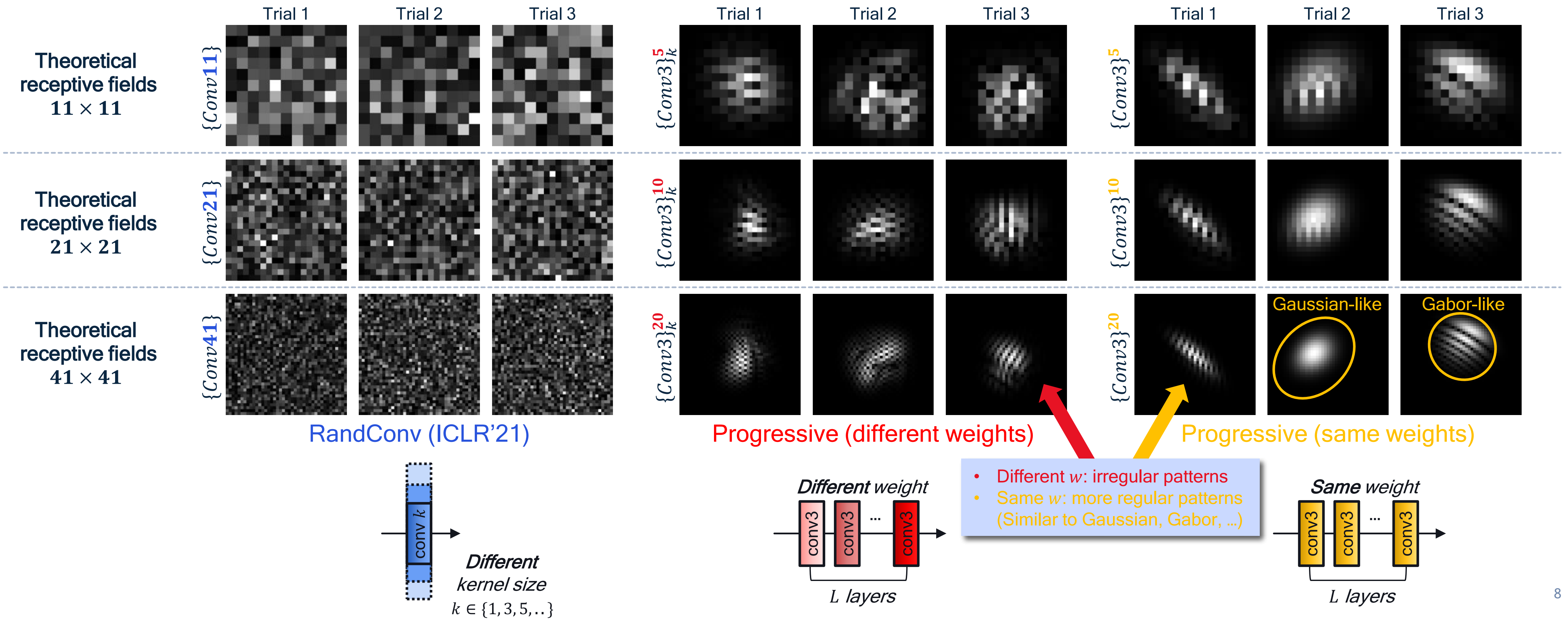
Pro-RandConv (Ours)

Alleviate semantic distortion issues

Proposed method

Progressive Random Convolution for Single Domain Generalization

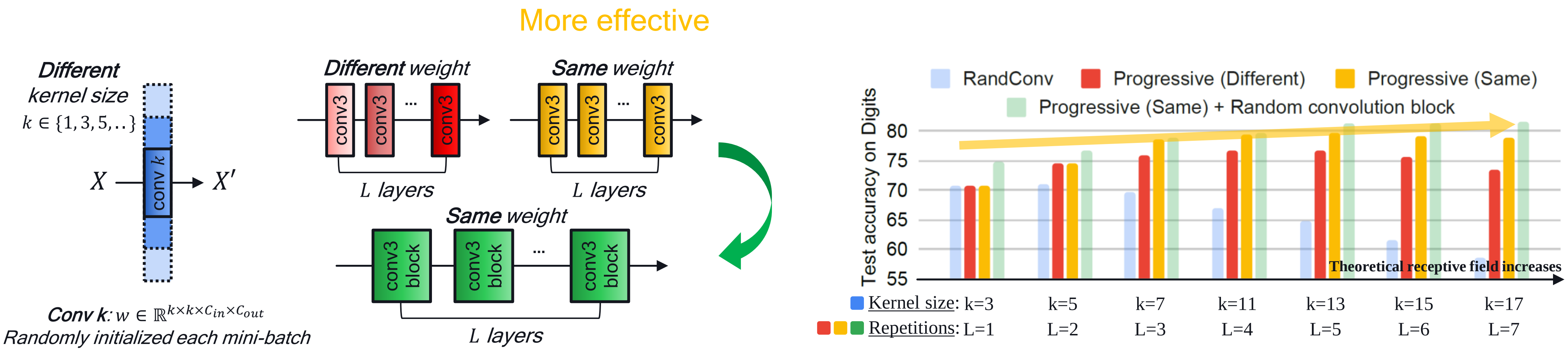
Main contribution 1: Progressive approach



Proposed method

Progressive Random Convolution for Single Domain Generalization

Main contribution 1: Progressive approach
- RandConv (ICLR'21) < Progressive approach with different weights < Progressive approach with the same weights (better)

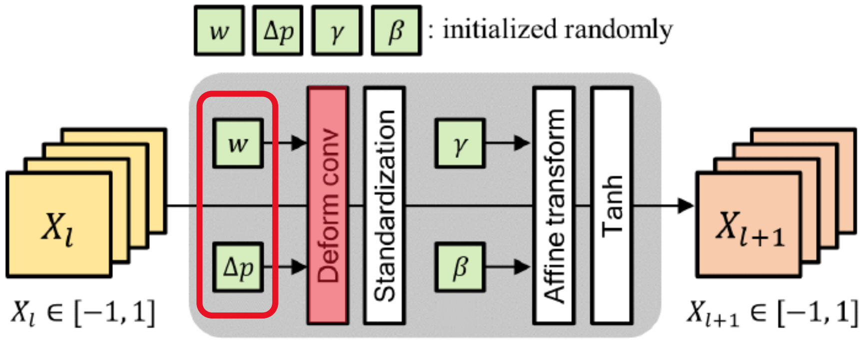


(a) RandConv (single-layer) (b) Progressive approach (multi-layer) (c) Performance with changing kernel size (k) or number of repetitions (L)

Proposed method

Progressive Random Convolution for Single Domain Generalization

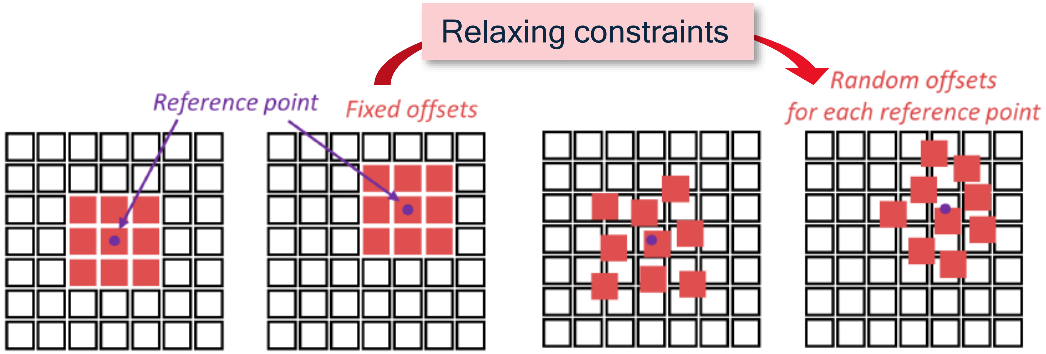
Main contribution 2: Random convolution blocks (Advanced design)
- Texture diversification by random deformable convolution: a generalized version of random convolutions



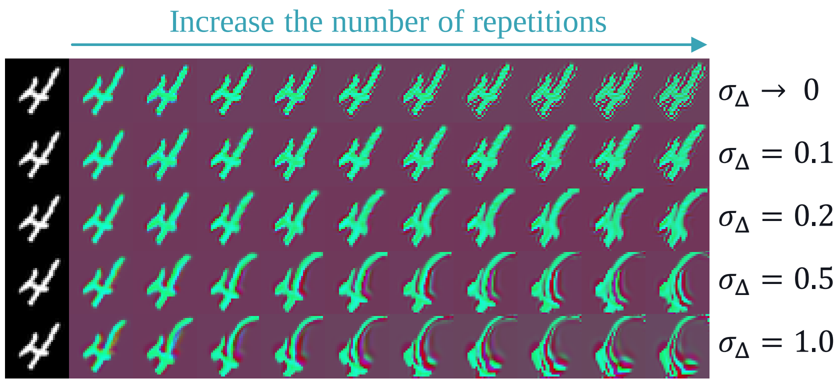
Random convolution block

w: Convolution weights ➔ $w \sim N(0, \sigma_w^2)$

Δp: Deformable offsets ➔ $\Delta p \sim N(0, \sigma_\Delta^2)$



(a) Basic random convolution (b) Random deformable convolution

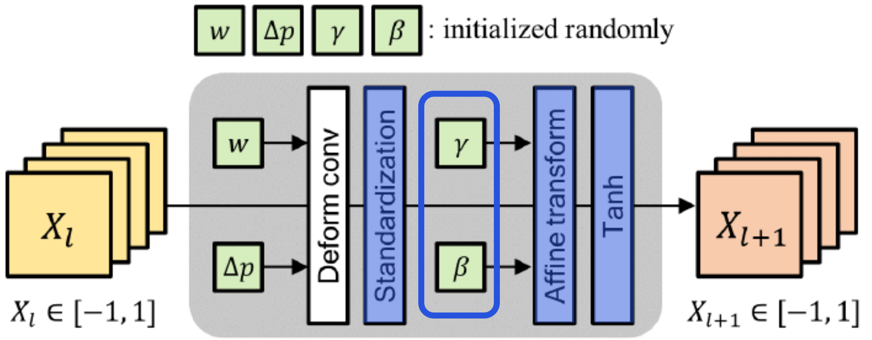


Distortion scale of deformable offsets

Proposed method

Progressive Random Convolution for Single Domain Generalization

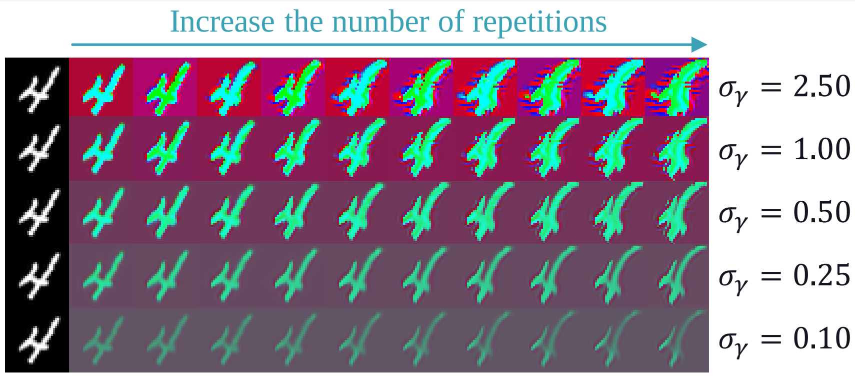
Main contribution 2: Random convolution blocks (Advanced design)
- Contrast diversification by random style transfer (AdaIN): the role of random gamma correction



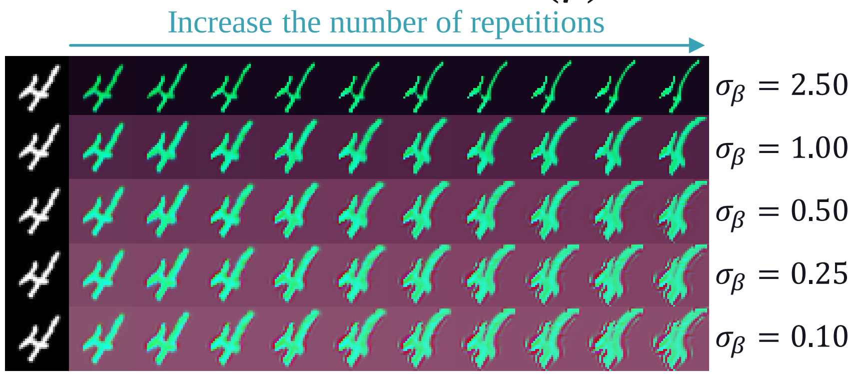
Random convolution block

γ: Affine transformation ➔ $\gamma \sim N(0, \sigma_\gamma^2)$

β: Affine transformation ➔ $\beta \sim N(0, \sigma_\beta^2)$



Affine transformation (γ)



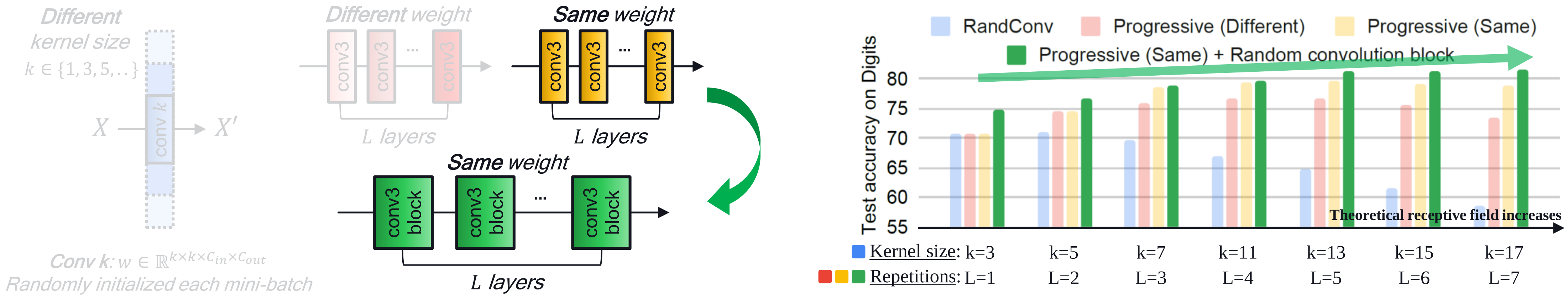
Affine transformation (β)

Proposed method

Progressive Random Convolution for Single Domain Generalization

Main contribution 2: Random convolution blocks (Advanced design)

- RandConv (ICLR'21) < Basic design (Different weights) < Basic design (Same weights) < Advanced design (Same weights)



(a) RandConv (single-layer) (b) Progressive approach (multi-layer) (c) Performance with changing kernel size (k) or number of repetitions (L)

Algorithm and training pipeline

Progressive Random Convolution for Single Domain Generalization

Algorithm 1 Pro-RandConv

Input: Source domain $\mathcal{S} = \{\mathbf{x}_n, y_n\}_{n=1}^{N_S}$

Output: Trained network $f_\phi(\cdot)$

Initialize network parameters ϕ

for $t = 1$ **to** T_{max} **do**

  Initialize a random convolution block $\mathcal{G}$:

  $w \sim N(0, \sigma_w^2)$ // Convolution weights

  $\Delta p \sim N(0, \sigma_\Delta^2)$ // Deformable offsets

  $\gamma \sim N(0, \sigma_\gamma^2)$ // Affine transformation (gamma)

  $\beta \sim N(0, \sigma_\beta^2)$ // Affine transformation (beta)

  Progressive augmentation:

  $\mathbf{X} \sim \mathcal{S}$ // Sample a mini-batch

  $\mathbf{X}_0 \leftarrow \mathbf{X}$ // Set an initial value

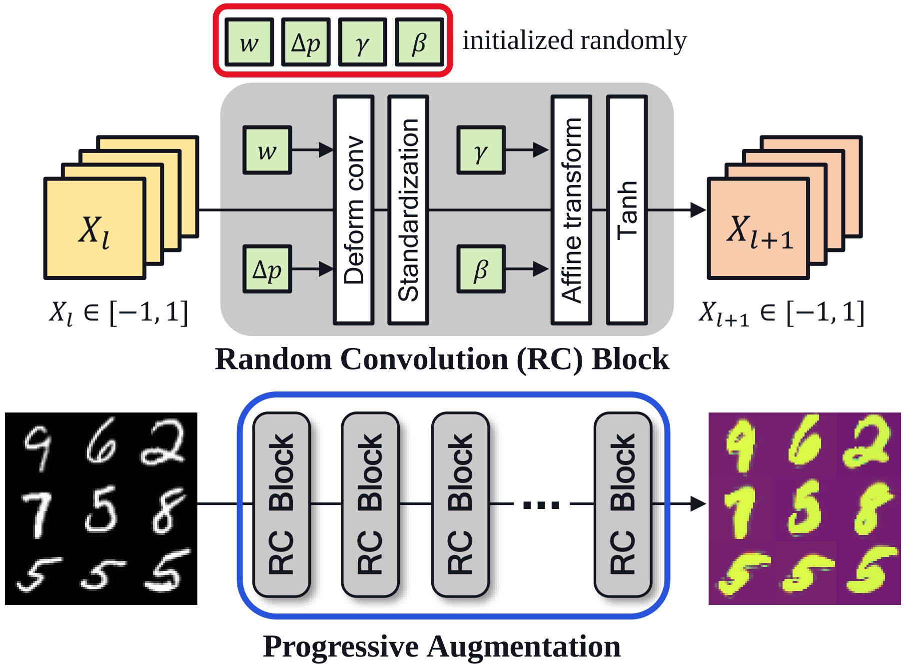
  $\boxed{L \sim U(\{1, 2, ..., L_{max}\})}$ // Repetition numbers

  for $l = 1$ **to** L **do**

    $\mathbf{X}_l = \mathcal{G}(\mathbf{X}_{l-1})$ // Apply Pro-RandConv

  Training a network:

  $\phi \leftarrow \phi - \alpha \nabla_\phi \mathcal{L}_{\text{task}}(\mathbf{X}_0, \mathbf{X}_L; \phi)$ // Network update



Experimental results: Digit recognition

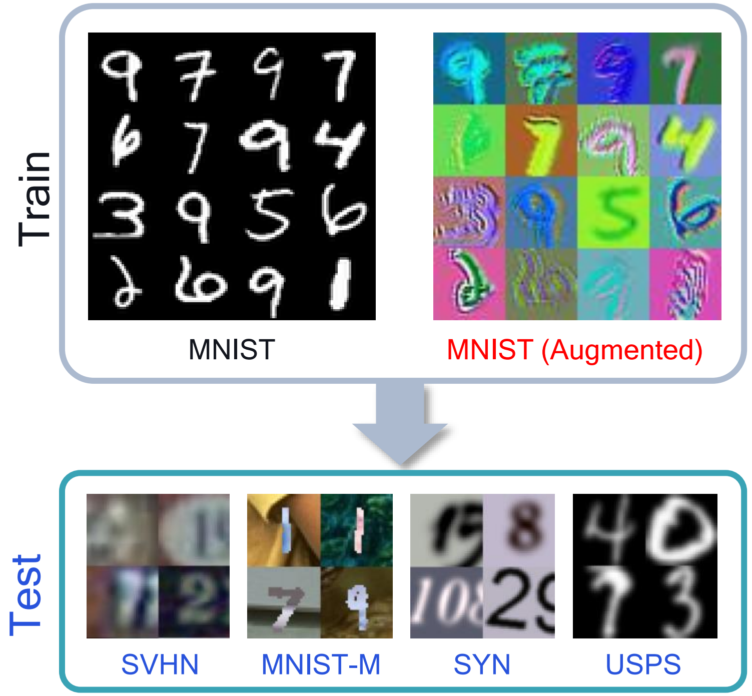
Progressive Random Convolution for Single Domain Generalization

- Dataset: Digits (MNIST ➔ SVHN, MNIST-M, SYN, USPS) [Model: LeNet]

Category	Paper	Methods	MNIST → SVHN	MNIST → MNIST-M	MNIST → SYN	MNIST → USPS	Average	Gap
Baseline	-	Baseline (ERM)	32.52	54.92	42.34	78.21	52.00	-29.35
Basic Data Augmentation	-	**Color jitter***	**36.04**	**57.56**	**43.94**	77.76	**53.83**	**-27.52**
	-	Grayscale*	32.92	55.44	42.38	**78.22**	52.24	-29.11
	-	Pespective*	33.63	43.86	40.92	69.12	46.88	-34.47
	-	Rotate*	31.99	54.86	38.22	69.54	48.65	-32.70
Automated Data Augmentation	CVPR'19	AutoAugment	45.23	60.53	**64.52**	**80.62**	62.72	-18.63
	CVPRW'20	**RandAugment**	**54.77**	**74.05**	59.60	77.33	**66.44**	**-14.91**
Adversarial Data Augmentation or Learnable Generator	NeurIPS'18	ADA	35.51	60.41	45.32	77.26	54.62	-26.73
	CVPR'20	M-ADA	42.55	67.94	48.95	78.53	59.49	-21.86
	NeurIPS'20	ME-ADA	42.56	63.27	50.39	81.04	59.32	-22.03
	ICCV'21	L2D	**62.86**	**87.30**	63.72	83.97	74.46	-6.89
	CVPR'21	**PDEN**	62.21	82.20	**69.39**	**85.26**	**74.77**	**-6.58**
Domain Generalization	ICCV'17	CCSA	25.89	49.29	37.31	83.72	49.05	-32.30
	CVPR'19	d-SNE	26.22	50.98	37.83	**93.16**	52.05	-29.30
	CVPR'19	JiGen	33.80	57.80	43.79	77.15	53.14	-28.21
	CVPR'22	**MetaCNN**	**66.50**	**88.27**	**70.66**	89.64	**78.76**	**-2.59**
Image Randomization (Non-trainable)	ICLR'21	RandConv*	61.66	**84.53**	67.87	85.31	74.84	-6.51
	Ours	Progressive (Diff)	60.73	78.47	71.46	88.20	74.72	-6.63
	Ours	Progressive (Same)	65.67	76.26	77.13	**93.98**	78.26	-3.09
	Ours	**Pro-RandConv**	**69.67**	82.30	**79.77**	93.67	**81.35**	

* denote reproduced results

<Digit recognition>



Averaged accuracy on test domains

52.0% ➔ 81.4%

Experimental results: Object recognition

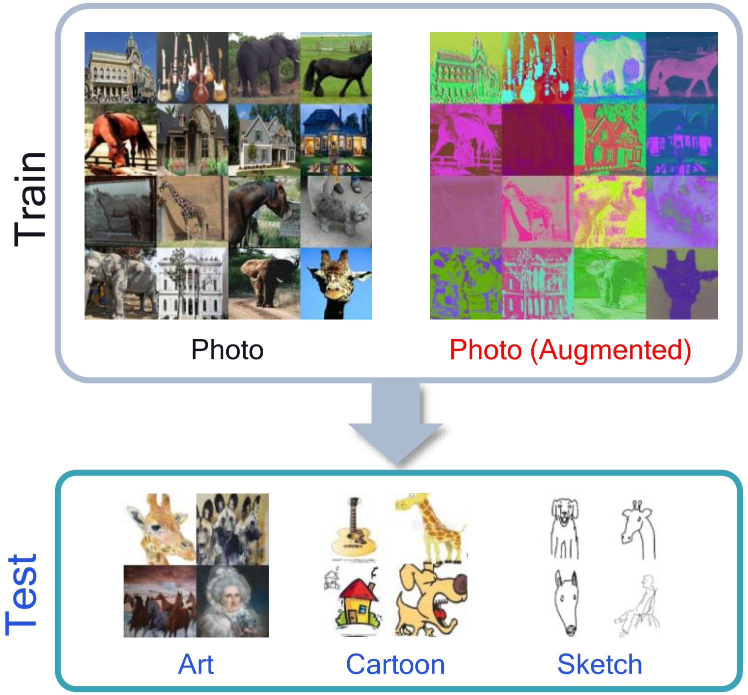
Progressive Random Convolution for Single Domain Generalization

- Dataset: PACS (4 domains: 1 domain for training, 3 domains for test) [Model: ResNet18]

Category	Paper	Methods	Art (A) → CPS	Cartoon (C) → APS	Photo (P) → ACS	Sketch (S) → ACP	Average	Gap
Baseline	-	Baseline (ERM)	74.64	73.36	56.31	48.27	63.15	-5.73
Basic Data Augmentation	-	**Color jitter***	**75.94**	**76.56**	59.27	**50.24**	**65.50**	**-3.38**
	-	Grayscale*	74.29	75.75	58.96	47.67	64.17	-4.71
	-	Pespective*	72.29	70.17	**59.99**	43.79	61.31	-7.57
	-	Rotate*	73.47	71.06	56.95	46.61	62.02	-6.86
Automated Data Augmentation	CVPR'19	AutoAugment*	76.48	77.09	60.99	52.46	66.76	-2.12
	CVPRW'20	**RandAugment***	**76.76**	**78.00**	**62.09**	**56.40**	**68.31**	**-0.57**
Adversarial Data Augmentation or Learnable Generator	NeurIPS'18	ADA	72.43	71.97	44.63	45.73	58.70	-10.18
	CVPR'21	SagNet	73.20	75.67	48.53	50.07	61.90	-6.98
	CVPR'22	GeoTexAug	72.07	**78.70**	49.07	**59.97**	65.00	-3.88
	ICCV'21	**L2D**	**76.91**	77.88	**52.29**	53.66	**65.18**	**-3.70**
Image Randomization (Non-trainable)	ICLR'21	RandConv*	76.93	76.47	62.46	54.13	67.50	-1.38
	Ours	Progressive (Diff)	75.46	75.39	60.02	55.02	66.47	-2.41
	Ours	Progressive (Same)	76.81	78.27	62.38	56.08	68.39	-0.49
	Ours	**Pro-RandConv**	**76.98**	**78.54**	**62.89**	**57.11**	**68.88**	

* denote reproduced results

<Object recognition>



Averaged accuracy on test domains

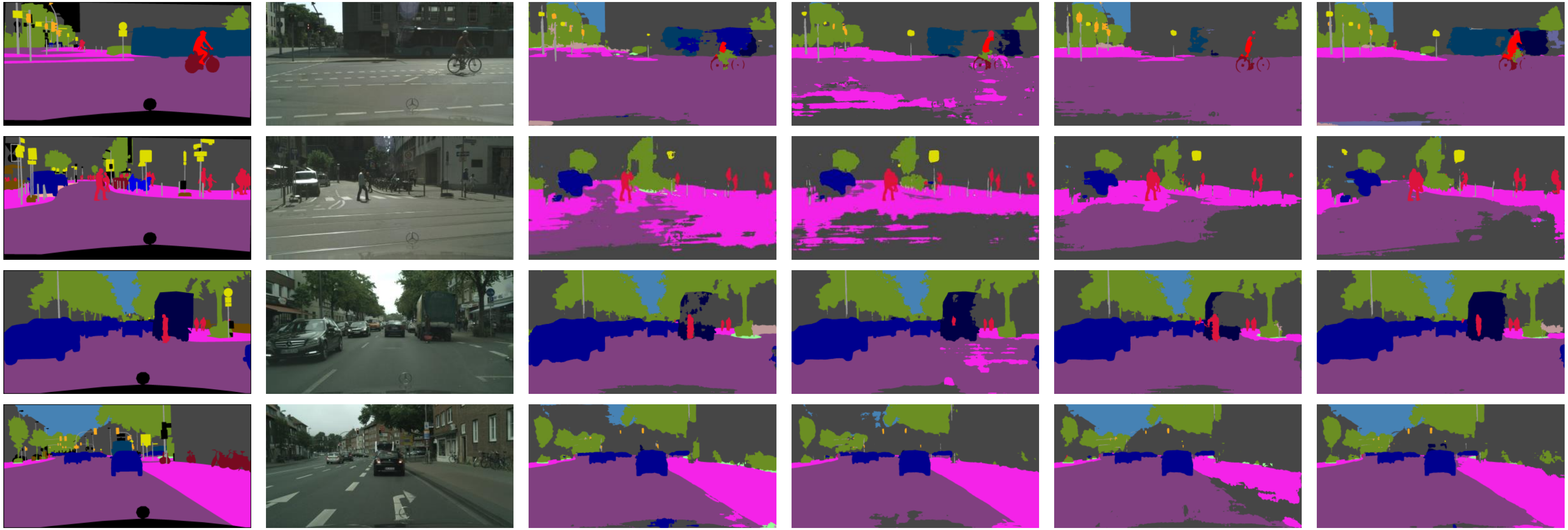
56.3% ➔ 62.9%

Experimental results: Semantic segmentation

Progressive Random Convolution for Single Domain Generalization

- Dataset: GTAV ➔ Cityscapes [Model: DeepLabV3+]

GT [Cityscapes]	Unseen images [Cityscapes]	Baseline [DeepLabV3] (mIoU 35.10%)	RandConv [ICLR'21] (mIoU 35.38%)	RobustNet [CVPR'21] (mIoU 39.72%)	Ours [CVPR'23] **(mIoU 42.36%)**



Thank you

Qualcomm

Follow us on:

For more information, visit us at:

qualcomm.com & qualcomm.com/blog