Learning Non-Parametric Invariances from Data with **Permanent Random Connectomes**

Dipan K. Pal and Marios Savvides

Carnegie Mellon University

Motivation

To be invariant to nuisance transformations in data, we would require

- **Knowledge** of the added transformations apriori

- **A different network architecture,** invariant for each nuisance transformation (inductive bias)

Loose Biological Motivation:

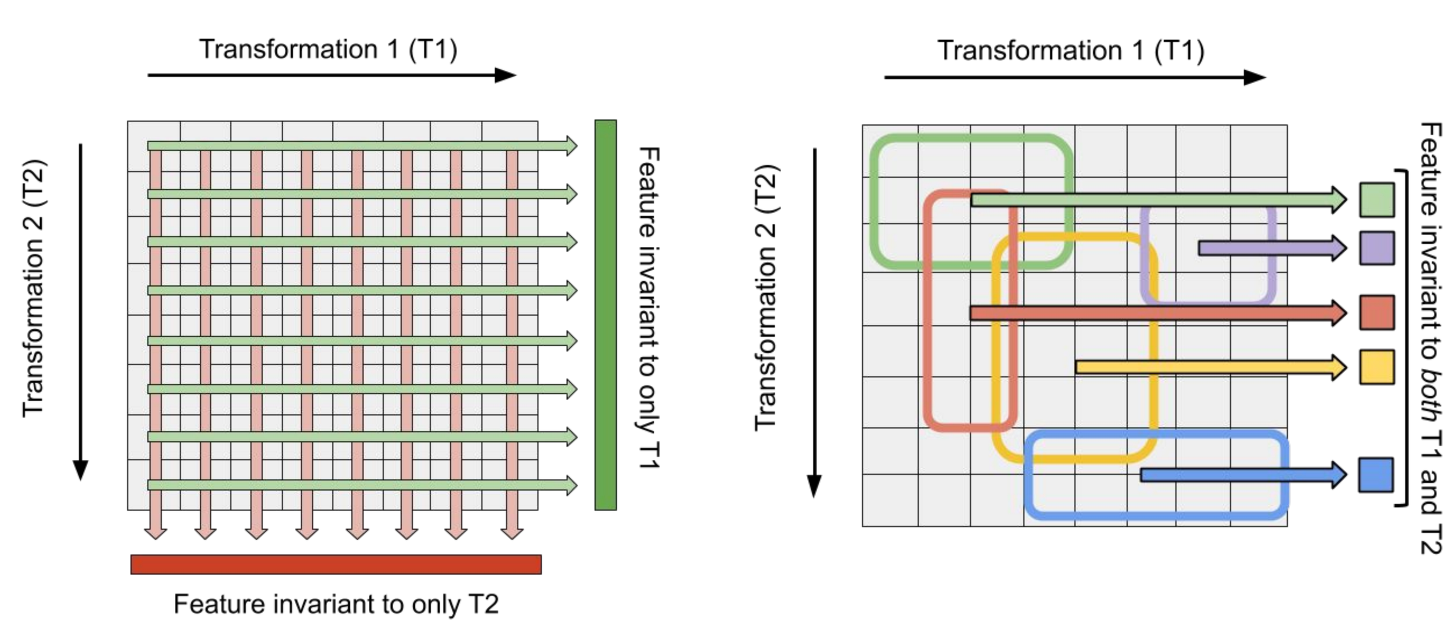
- Cortex lacks precise local pathways for backprop
- Unstructured local connections are common

PRC-NPTNs advantages:

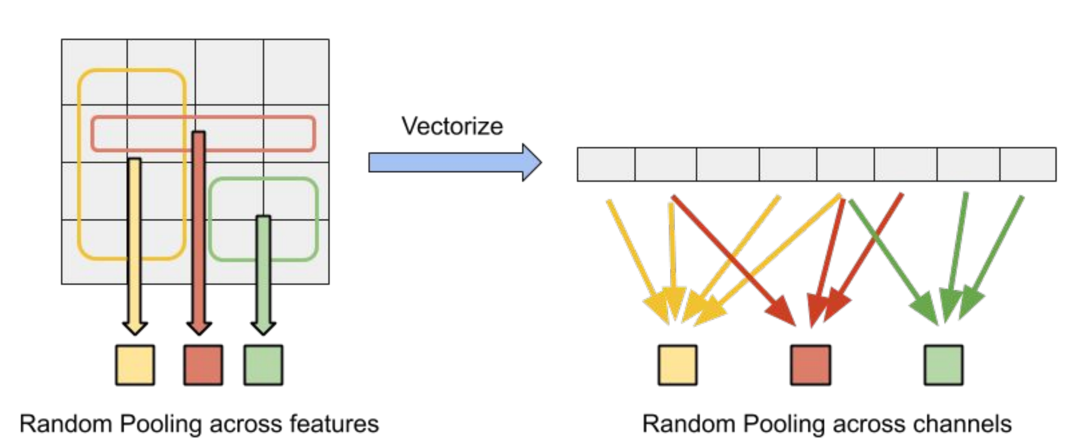
- ***No apriori knowledge*** of nuisance transformations is required.
- ***No change in architecture required.*** The exact same network can adapt to different transformations.
- ***Can be invariant towards combinations of transformations.*** A rich set of invariances can be invoked *explicitly* through the architecture.

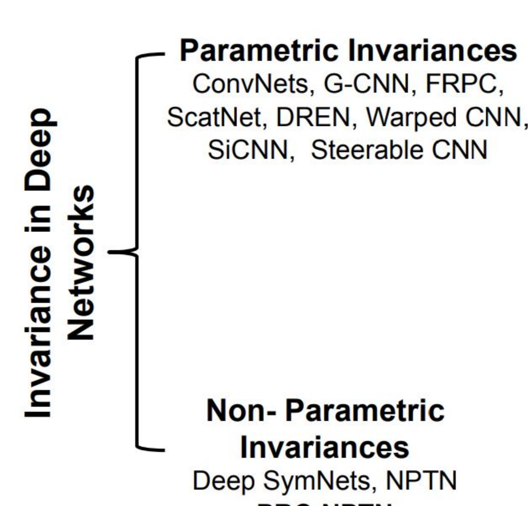
Do **permanent random connections** actually *improve* generalization in deep networks?

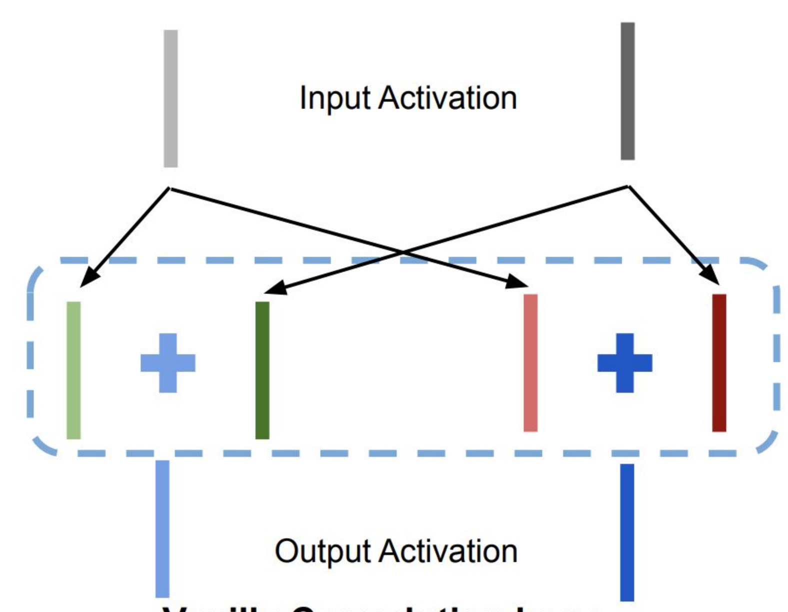
Permanent Random Connectome Networks



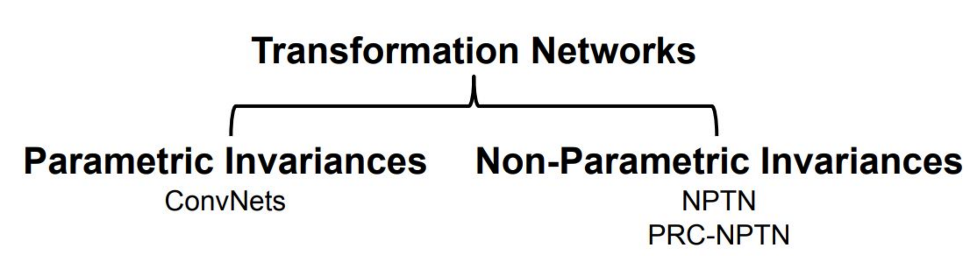
Top left: Invariances can invoked for individual transformations by pooling over each. **Top right:** Invariances to multiple transformations through permanent random support pooling. **Bottom left:** Permanent random support when vectorized leads to permanent random channel pooling. **Bottom right:** Most architectures offer only parametric invariance

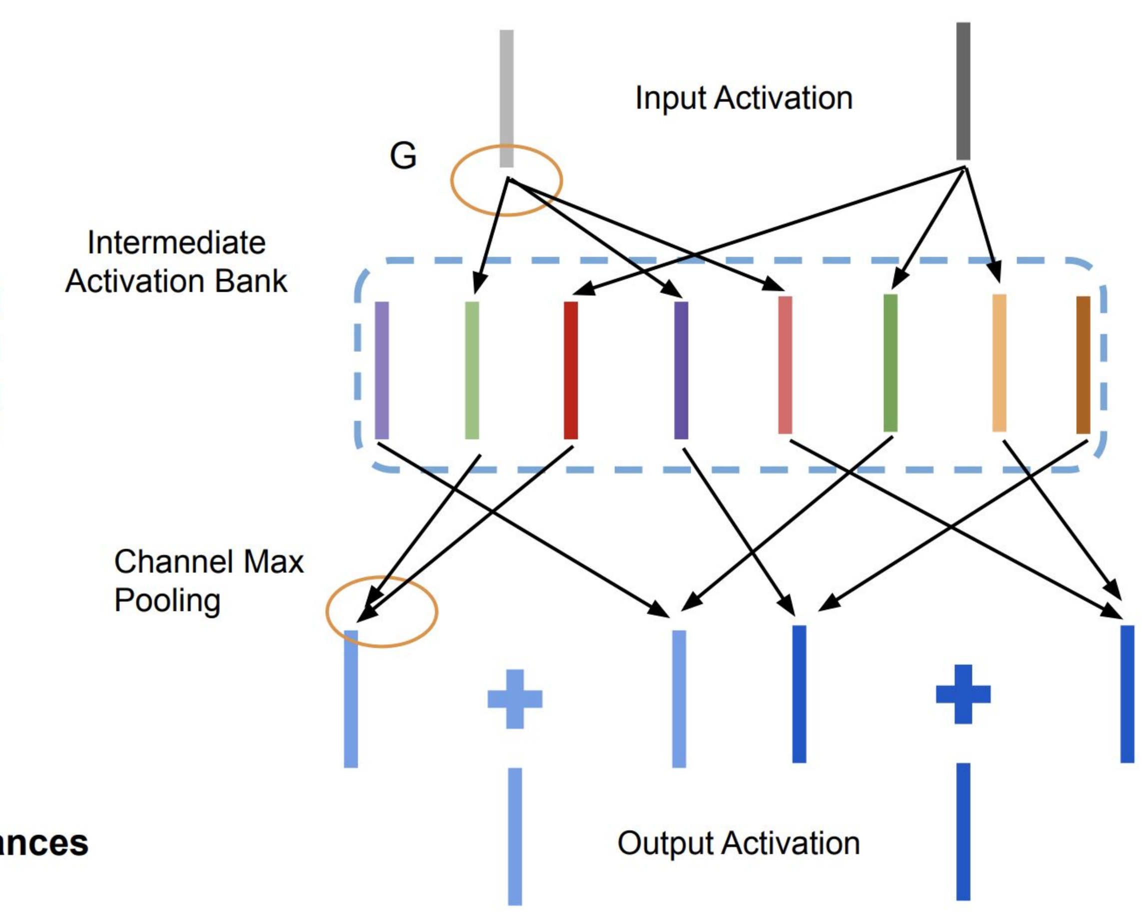






Vanilla Convolution layer





PRC-NPTN layer

Left bottom: Transformation Networks were introduced as a general framework for modelling feedforward convolutional networks.

Right. Each input channel is convolved with a number of filters (parameterized by G). Each of the resultant activation maps is connected to a one of the channel max pooling units randomly (initialized once, fixed during training and testing). Therefore, each channel pooling unit pools over a fixed random support.

Experimental Validation

Rotation	0°	***	90°	***
ConvNet (36)	$0.70_{\pm 0.03}$	-	$1.93_{\pm 0.02}$	-
ConvNet (36) FC	$0.66_{\pm 0.05}$	-	$1.58_{\pm 0.01}$	-
ConvNet (512)	$0.65_{\pm 0.04}$	-	$1.54_{\pm 0.03}$	-
NPTN (12,3)	$0.68_{\pm 0.06}$	-	$1.64_{\pm 0.02}$	-
PRCN (36,1)	$0.62_{\pm 0.08}$	$0.62_{\pm 0.06}$	$1.72_{\pm 0.05}$	$1.73_{\pm 0.06}$
PRCN (18,2)	$0.61_{\pm 0.02}$	$0.57_{\pm 0.02}$	$\mathbf{1.24_{\pm 0.01}}$	$1.33_{\pm 0.02}$
PRCN (12,3)	$\mathbf{0.58_{\pm 0.03}}$	$0.62_{\pm 0.04}$	$1.28_{\pm 0.01}$	$1.33_{\pm 0.01}$
PRCN (9,4)	$0.63_{\pm 0.02}$	$0.62_{\pm 0.04}$	$1.31_{\pm 0.03}$	$1.40_{\pm 0.03}$
Translations	**0 pixels**	***	**12 pixels**	***
ConvNet (36)	$0.69_{\pm 0.04}$	-	$4.43_{\pm 0.05}$	-
ConvNet (36) FC	$0.60_{\pm 0.02}$	-	$3.49_{\pm 0.11}$	
ConvNet (512)	$0.63_{\pm 0.02}$	-	$3.56_{\pm 0.04}$	-
NPTN (12,3)	$0.66_{\pm 0.02}$	-	$4.19_{\pm 0.04}$	-
PRC-NPTN (36,1)	$0.65_{\pm 0.02}$	$0.65_{\pm 0.05}$	$3.85_{\pm 0.11}$	$3.83_{\pm 0.10}$
PRC-NPTN (18,2)	$\mathbf{0.59_{\pm 0.07}}$	$0.59_{\pm 0.03}$	$\mathbf{3.23_{\pm 0.03}}$	$3.34_{\pm 0.06}$
PRC-NPTN (12,3)	$0.63_{\pm 0.02}$	$0.66_{\pm 0.08}$	$3.35_{\pm 0.04}$	$3.52_{\pm 0.12}$
PRC-NPTN (9,4)	$0.65_{\pm 0.02}$	$0.69_{\pm 0.03}$	$3.49_{\pm 0.46}$	$3.69_{\pm 0.08}$

EXP A (Above): Individual Transformations. Test error statistics with mean and standard deviation on MNIST with progressively extreme transformations with a) random rotations and b) random pixel shifts. *** indicates ablation runs without any randomization i.e. without any random connectomes (applicable only to PRC-NPTNs). Permanent and random channel pooling provides generalization benefits.

Rot/Trans	0° 0	15° 2	30° 4	45° 6	60° 8	75° 10	90° 12
ConvNet (36)	$0.68_{\pm 0.03}$	$0.72_{\pm 0.02}$	$1.31_{\pm 0.02}$	$2.32_{\pm 0.04}$	$5.06_{\pm 0.04}$	$10.90_{\pm 0.08}$	$19.60_{\pm 0.16}$
ConvNet (36) FC	$0.64_{\pm 0.03}$	$0.66_{\pm 0.01}$	$0.95_{\pm 0.04}$	$1.50_{\pm 0.02}$	$3.42_{\pm 0.03}$	$8.14_{\pm 0.11}$	$15.61_{\pm 0.11}$
ConvNet (512)	$0.66_{\pm 0.05}$	$0.65_{\pm 0.02}$	$0.97_{\pm 0.02}$	$1.60_{\pm 0.04}$	$3.50_{\pm 0.04}$	$7.90_{\pm 0.06}$	$15.19_{\pm 0.09}$
NPTN (12,3)	$0.66_{\pm 0.02}$	$0.69_{\pm 0.03}$	$1.07_{\pm 0.03}$	$1.85_{\pm 0.02}$	$4.24_{\pm 0.11}$	$9.58_{\pm 0.06}$	$17.79_{\pm 0.16}$
PRC-NPTN (36,1)	$0.61_{\pm 0.03}$	$0.70_{\pm 0.01}$	$1.09_{\pm 0.04}$	$1.80_{\pm 0.02}$	$3.93_{\pm 0.02}$	$9.09_{\pm 0.11}$	$17.03_{\pm 0.13}$
PRC-NPTN (18,2)	$\mathbf{0.57_{\pm 0.02}}$	$\mathbf{0.58_{\pm 0.01}}$	$\mathbf{0.77_{\pm 0.02}}$	$\mathbf{1.21_{\pm 0.07}}$	$\mathbf{2.74_{\pm 0.04}}$	$\mathbf{6.78_{\pm 0.12}}$	$\mathbf{13.79_{\pm 0.08}}$
PRC-NPTN (12,3)	$0.59_{\pm 0.03}$	$0.58_{\pm 0.01}$	$0.78_{\pm 0.02}$	$1.26_{\pm 0.02}$	$2.91_{\pm 0.05}$	$7.13_{\pm 0.09}$	$14.23_{\pm 0.07}$
PRC-NPTN (9,4)	$0.63_{\pm 0.04}$	$0.59_{\pm 0.02}$	$0.81_{\pm 0.02}$	$1.35_{\pm 0.02}$	$3.12_{\pm 0.02}$	$7.26_{\pm 0.02}$	$14.62_{\pm 0.16}$

EXP B (Above): Simultaneous Transformations. PRC-NPTNs can achieve better generalization to rotation and extreme translation *simultaneously* due to permanent random support pooling. For PRC-NPTN and NPTN the brackets indicate the number of channels in the layer 1 and G.

EXP C (Bottom): DenseNets with Permanent Random Connectomes. PRC-NPTNs can be applied as a drop in replacement to conv layers. When applied to DenseNets, they provide clear generalization benefits. Permanent random connectomes seem to significantly help in generalizing DenseNets.

Method	CIFAR10	(w/o Random)	CIFAR 10++	(w/o Random)
DenseNet-Conv	$11.47_{\pm 0.19}$	-	$21.37_{\pm 0.29}$	-
DensePRC-NPTN (CMP=1)	$11.82_{\pm 0.20}$	$13.33_{\pm 0.23}$	$22.03_{\pm 0.08}$	$23.88_{\pm 0.38}$
DensePRC-NPTN (CMP=2)	$10.78_{\pm 0.31}$	$11.67_{\pm 0.36}$	$20.71_{\pm 0.23}$	$21.90_{\pm 0.33}$
DensePRC-NPTN (CMP=3)	$10.95_{\pm 0.12}$	$11.59_{\pm 0.23}$	$20.95_{\pm 0.20}$	$21.80_{\pm 0.42}$
DensePRC-NPTN (CMP=4)	$\mathbf{10.61_{\pm 0.11}}$	$11.41_{\pm 0.12}$	$\mathbf{20.80_{\pm 0.12}}$	$21.47_{\pm 0.16}$