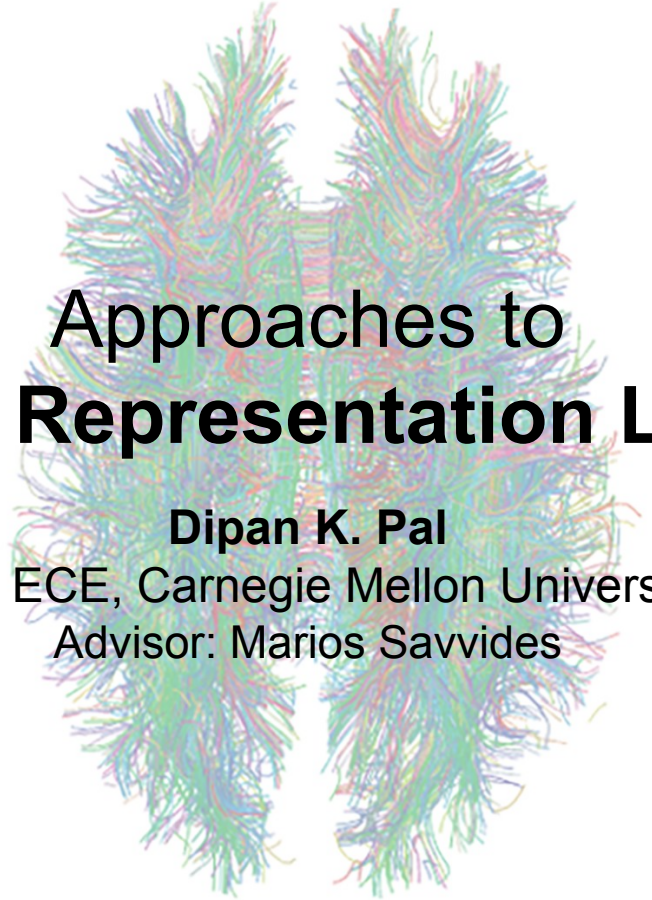
Electrical & Computer ENGINEERING



Approaches to **Invariant Representation Learning**

Dipan K. Pal
PhD ECE, Carnegie Mellon University
Advisor: Marios Savvides

The Problem of Representation Learning

{* EMERGENT TECH *}

Tesla death smash probe: Neither driver nor autopilot saw the truck

Report shows driver took a hands-off approach to driving

By Gareth Corfield 20 Jun 2017 at 21:58 143 SHARE ▼

"Neither Autopilot nor the driver noticed the white side of the tractor trailer against a brightly lit sky, so the brake was not applied. The high ride

Tesla tricked into speeding by researchers using electrical tape

BY KATE GIBSON
FEBRUARY 19, 2020 / 1:47 PM / MONEYWATCH

A Tesla vehicle was tricked into speeding by researchers who put electrical tape over a speed limit sign. The findings, released Wednesday by security firm McAfee, highlight possible difficulties with autonomous driving systems.

McAfee technicians fooled the car into reading the speed limit as 85 miles per hour by placing black tape across the middle of the first digit on a 35 mile per hour sign. That caused the vehicle's cruise control system to automatically accelerate, according to McAfee.

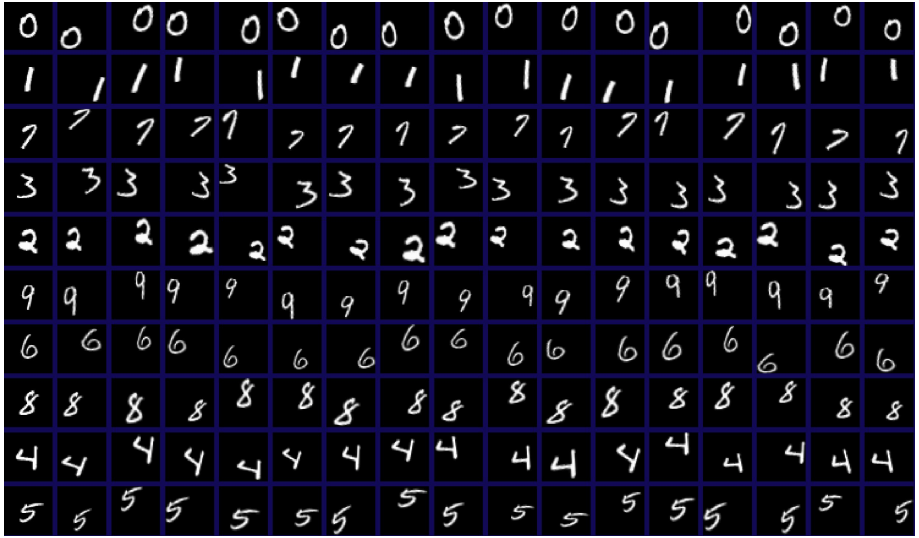
A Motivating Example

MNIST augmented with extreme rotation and translation

Consider MNIST augmented with **both** rotation and extreme translation.

To be invariant to such transformations, we would require

- **Knowledge** of the added transformations *apriori.*
- **A different architecture** invariant for each nuisance transformation.

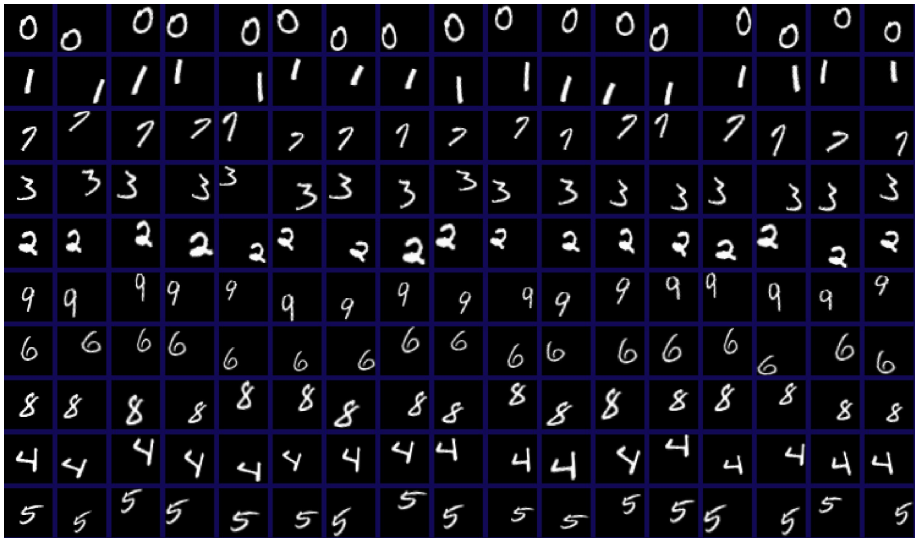


A Motivating Example

MNIST augmented with extreme rotation and translation

We design **network architectures** which in principle can address both issues simultaneously:

- ***No a priori knowledge*** of nuisance transformations is required. NPTNs and PRCNs can model complex transformations.
- ***No change in architecture required*.** The exact same network can adapt to different transformations.



The Outline

Chapter 1
The Invariance Problem

Chapter 2
Generalizing Convolution Neural Networks

Chapter 3
Non-Parametric Transformation Networks

Chapter 4
Permanent Random Connectome Networks

The Outline

Chapter 1
The Invariance Problem

Chapter 2
Generalizing Convolution Neural Networks

Chapter 3
Non-Parametric Transformation Networks

Chapter 4
Permanent Random Connectome Networks

Some Tools for Learning Representations

And one way to categorize them

Loss Functions

Self-Supervision

Data Augmentation

Architectures

Almost Every Tool for Learning Representations

Explicitly Encourages Invariant Representations

Except

Loss Functions

Cross Entropy
Triplet Loss ’15
Center Loss ’16
SphereFace ’17
L2 Constrained SM ’17
Copernican Loss ’17
Ring Loss ‘18
CosFace ’18
ArcFace ’18

Self-Supervision

Robust Features ’08
Data Init ’15
Jigsaw ’15 ‘16 ‘18
Split Brain ’16
Colorization ’16
Adversarial ’16
Context Encoders ’16
RotNet ’18
Contrastive Predictive ’19
Video Jigsaw ’19
Momentum Contrast ’19
Pretext Invariant RL ’19
SimCLR ’20

Data Augmentation

AutoAugment ’18
AdvProp ’19
PBAutoAugment ’19
RandAugment ’19
FastAutoAugment ’19
Adversarial AutoAugment ’20
MaxUp ’20

Architectures

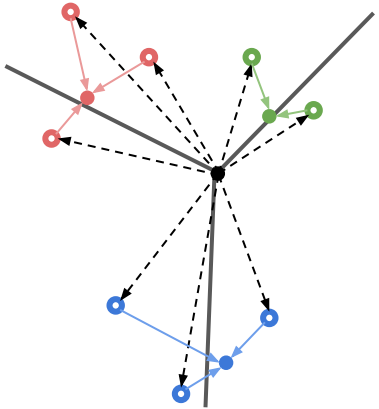
ConvNet ’98
AlexNet ’12
VGG ’14
GoogLeNet ’14
SPP ’14
ResNets ’15
DenseNets ’16
WideResNets ’16
PolyNet ’16
MobileNets ’17
DPN ’17
SENet ’18
GPipe ’18
MobileNetV2 ’19
EfficientNet ’19
Res2Net ’20

Loss Functions for Supervised Classification

Some of Our Work

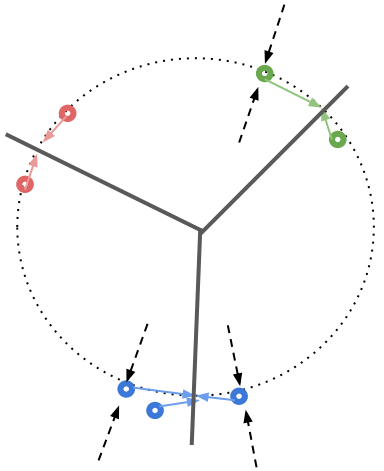
Loss Functions

Cross Entropy
Triplet Loss ’15
Center Loss ’16
SphereFace ’17
L2 Constrained SM ’17
Copernican Loss ’17
Ring Loss ‘18
CosFace ’18
ArcFace ’18



Copernican Loss (Our Work)

The first loss function with a **cosine distance metric**, effective for *both* object and face recognition



Ring Loss (Our Work)

SOTA results on Janus IJB-A and IJB-CS3 large scale face recognition benchmarks

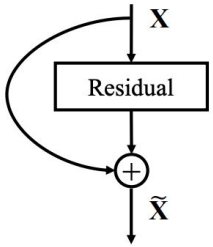
Dipan K. Pal and Marios Savvides, “**Copernican loss: Learning a Discriminative Cosine Embedding**”, *CMU Tech Report,* 2017
Zheng, Yutong, **Dipan K. Pal,** and Marios Savvides. "**Ring loss: Convex feature normalization for face recognition**." CVPR 2018 **(oral spotlight)**

Most Architectures for Representation Learning

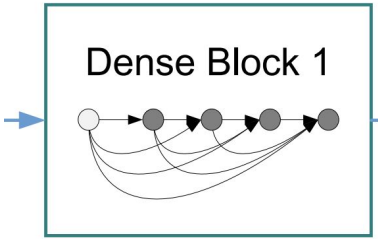
Does Not Address Invariance Explicitly

Architectures

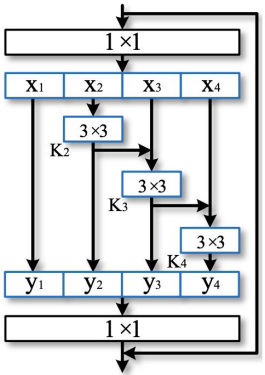
ConvNet '98
AlexNet '12
VGG '14
GoogLeNet '14
SPP '14
ResNets '15
DenseNets '16
WideResNets '16
PolyNet '16
MobileNets '17
Capsule Nets '17
DPN '17
SENet '18
GPipe '18
MobileNetV2 '19
EfficientNet '19
Res2Net '20



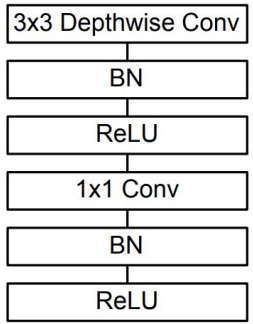
ResNet Module



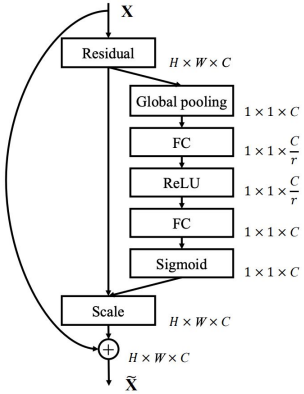
DenseNet



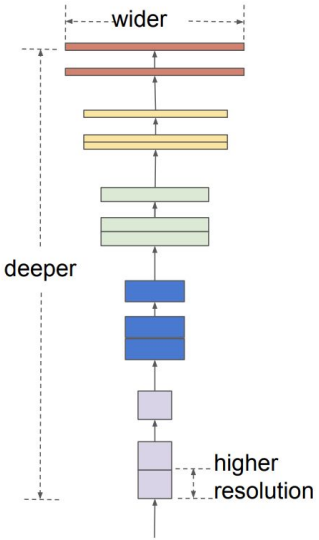
Res2Net



MobileNet



SE-ResNet Module
SE Net Block



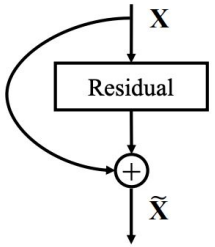
Efficient Net

Most Architectures for Representation Learning

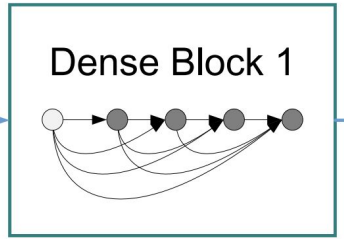
Boil Down to Only Two Components

The Convolution Layer

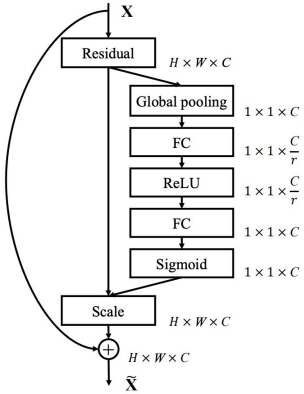
The Skip Connection



ResNet Module

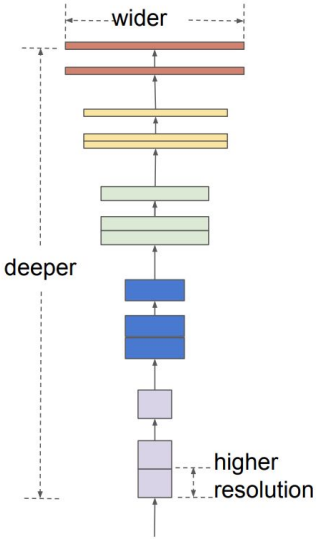


DenseNet

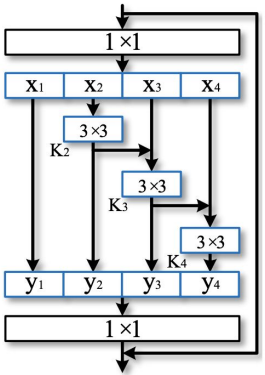


SE-ResNet Module

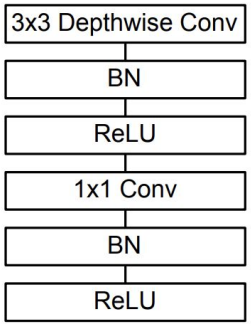
SE Net Block



Efficient Net



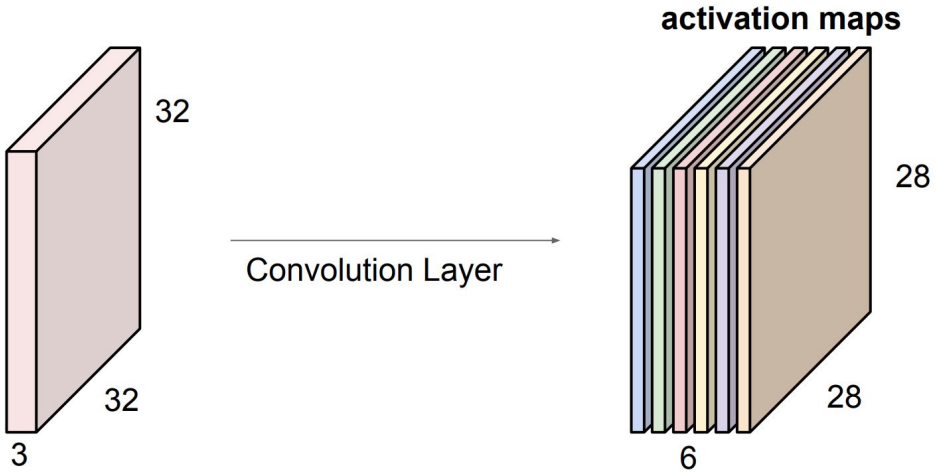
Res2Net



MobileNet

Architectures for Representation Learning

We Focus on the Convolution Layer

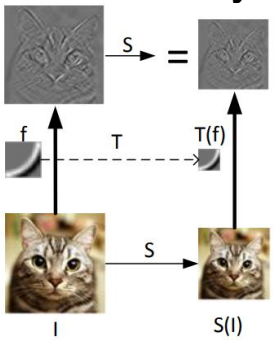


The Convolution Layer

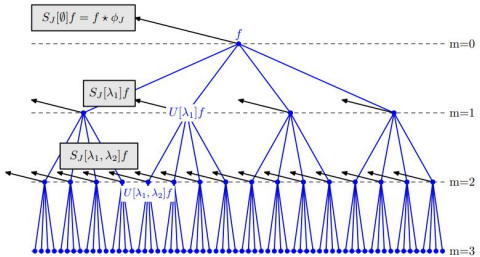
We redesign the core convolution layer itself

Few Prior Architectures do Encourage Invariance

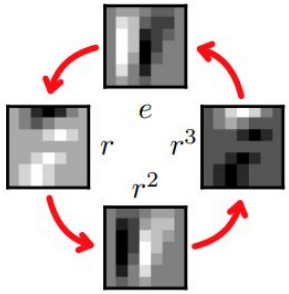
However, Only Parametric Invariance



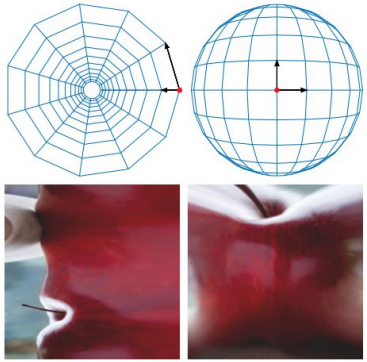
Scale Inv CNN ’14



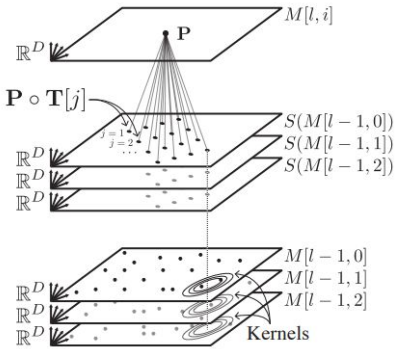
ScatNet ’15



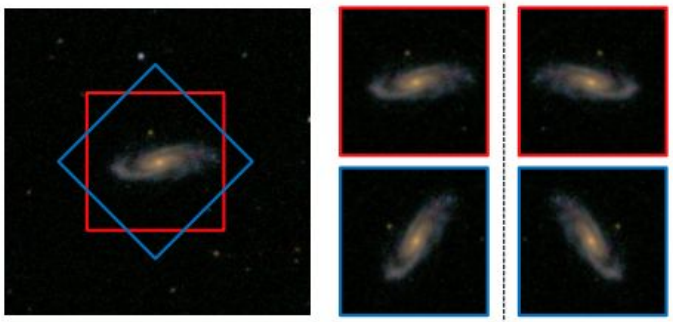
Group-CNN ’16



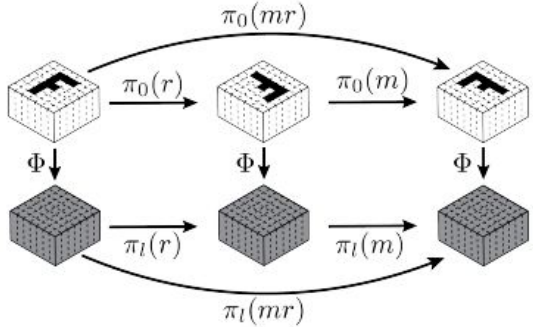
Warped CNN ’18



Deep Symmetry Nets ’14



Rotation Inv Net ’15



Sreeable CNN ’17

The Outline

Chapter 1
The Invariance Problem

Chapter 2
Generalizing Convolution Neural Networks

Chapter 3
Non-Parametric Transformation Networks

Chapter 4
Permanent Random Connectome Networks

The Chapter Summary

Chapter 1
The Invariance Problem

- Invariance to nuisance transformations is **still an unsolved problem.**
- **Most efforts** in the field of loss functions, self-supervision, data augmentation have **directly or indirectly focused on invariance**.
- Most studies on **architecture have not focused on invariance.**
- The few that have, only focused on **parametric invariances without learning them.**

We address this gap in architecture research.

The Outline

Chapter 1
The Invariance Problem

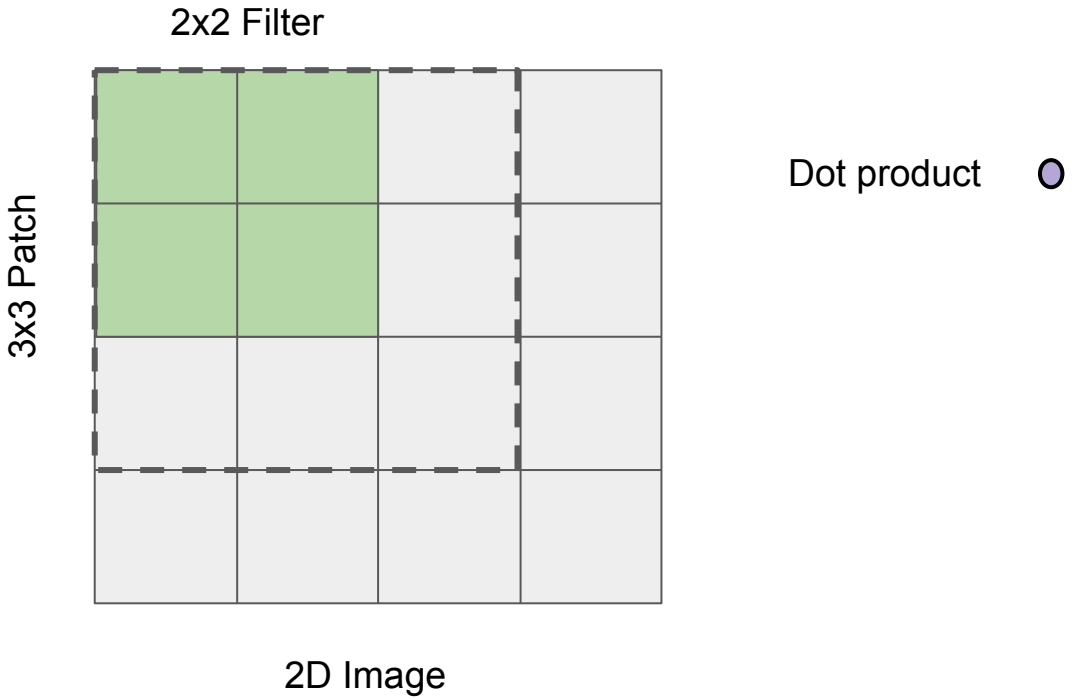
Chapter 2
Generalizing Convolution Neural Networks

Chapter 3
Non-Parametric Transformation Networks

Chapter 4
Permanent Random Connectome Networks

Digging Deeper into ConvNets

An alternate view of the convolution + pooling operation



Digging Deeper into ConvNets

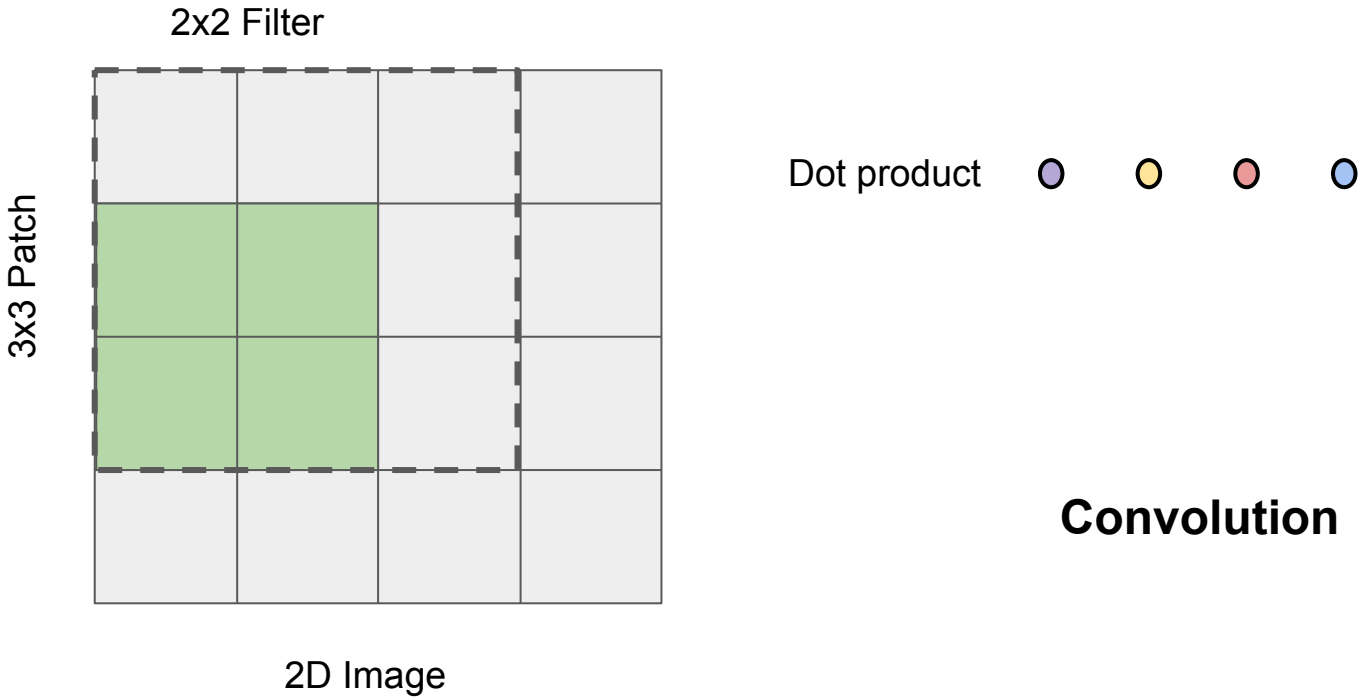
An alternate view of the convolution + pooling operation

Digging Deeper into ConvNets

An alternate view of the convolution + pooling operation

Digging Deeper into ConvNets

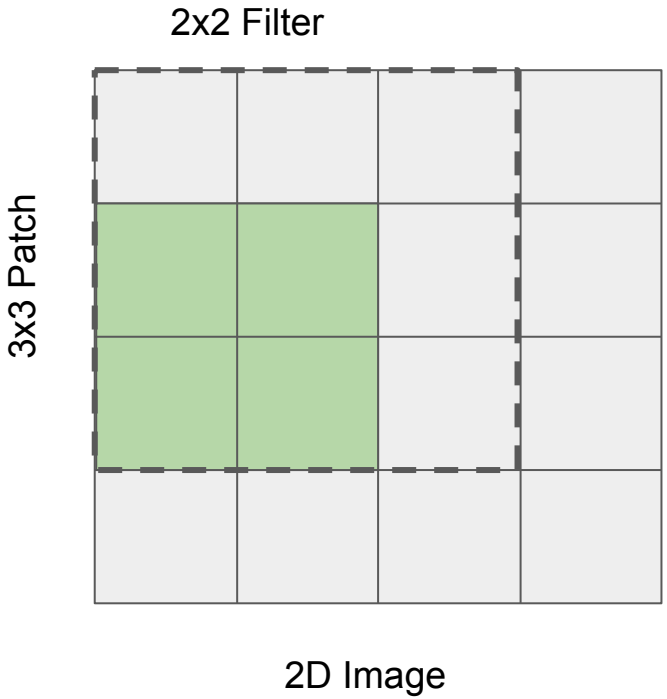
An alternate view of the convolution + pooling operation

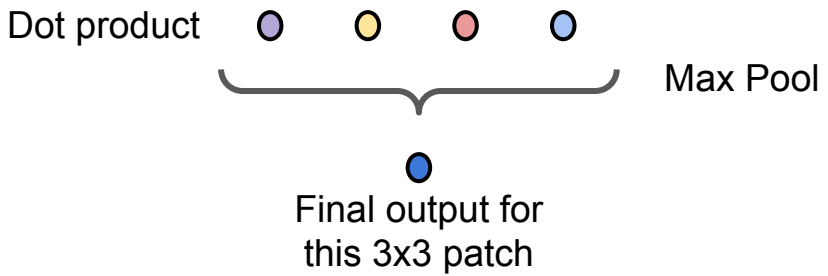


Convolution

Digging Deeper into ConvNets

An alternate view of the convolution + pooling operation





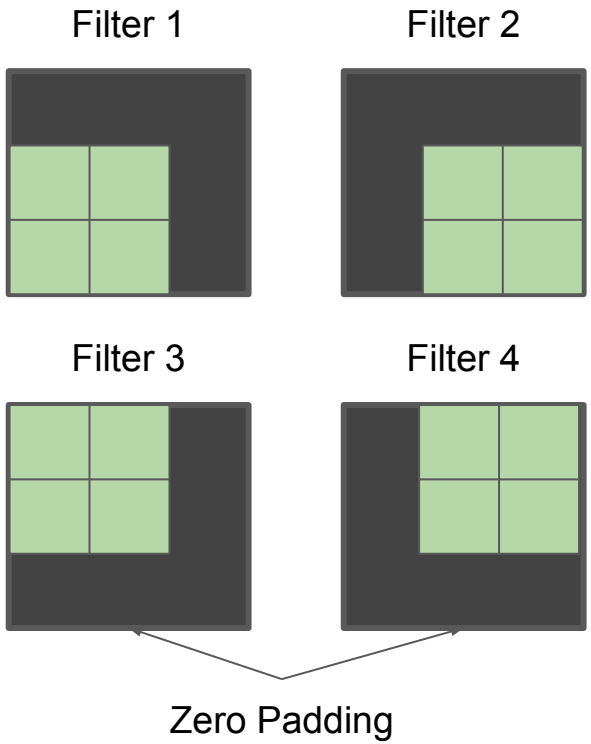
Convolution

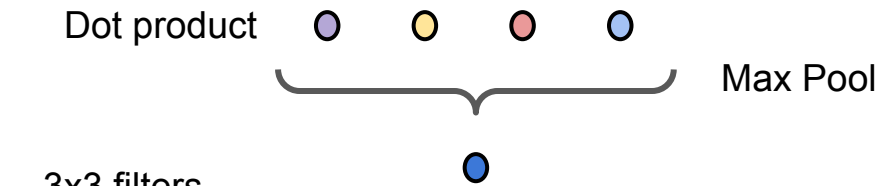
+

Pooling

Digging Deeper into ConvNets

An alternate view of the convolution + pooling operation





3x3 filters

2x2 non-zeros

Final output for this 3x3 patch

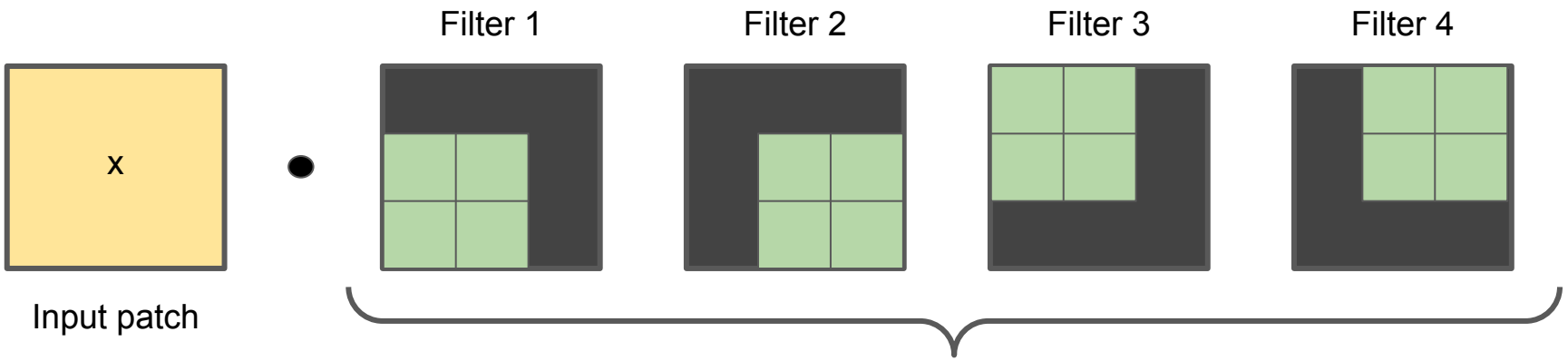
Convolution

+

Pooling

Digging Deeper into ConvNets

Translation between filters results in translation invariance



Filters are **translated** versions of each other.

Pooling over their dot products results in a **translation** invariant feature of x.

A Group Theoretic Approach to Invariance

Unitary groups are useful for invoking invariance

We can model a transformation as an element of a set of transformations forming a group.

A group is any set of operators that follows 4 axioms.

Consider a square matrix operator U, it is unitary iff for any vectors x and y

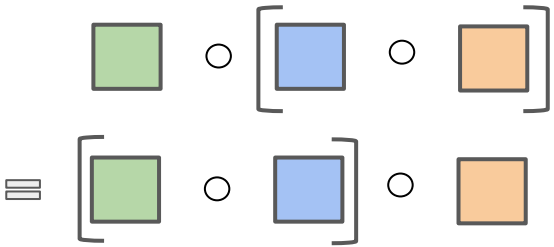
$$\langle Ux, Uy \rangle = \langle x, y \rangle$$

i.e. the matrix preserves dot products in its range
A *group* of *unitary* elements is a **unitary group**.

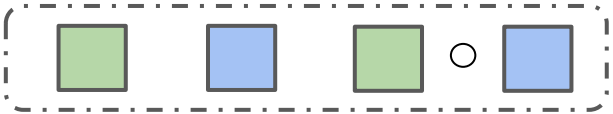
1) *Identity*

2) *Inverse*

3) *Associativity*

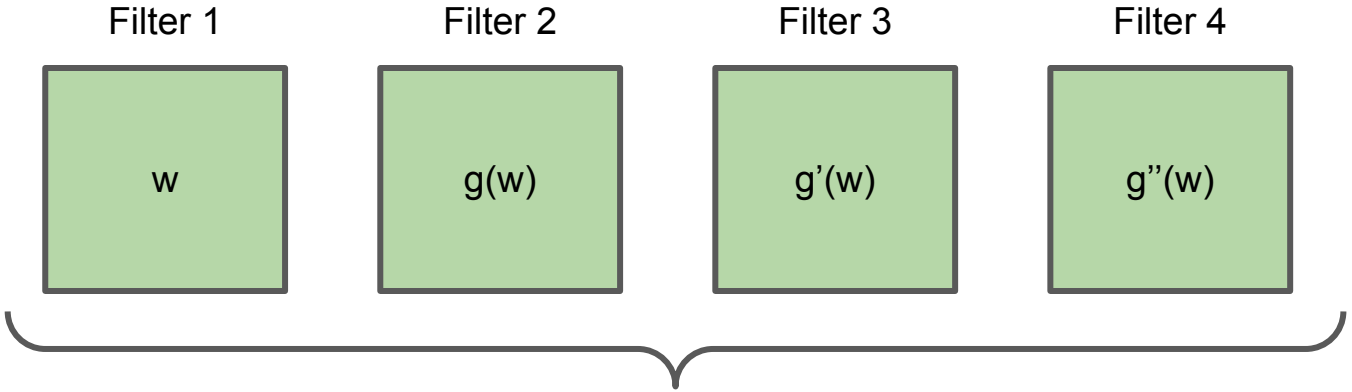


4) *Closure*



Towards More General Invariances

More complex transformation between filters results in general invariances



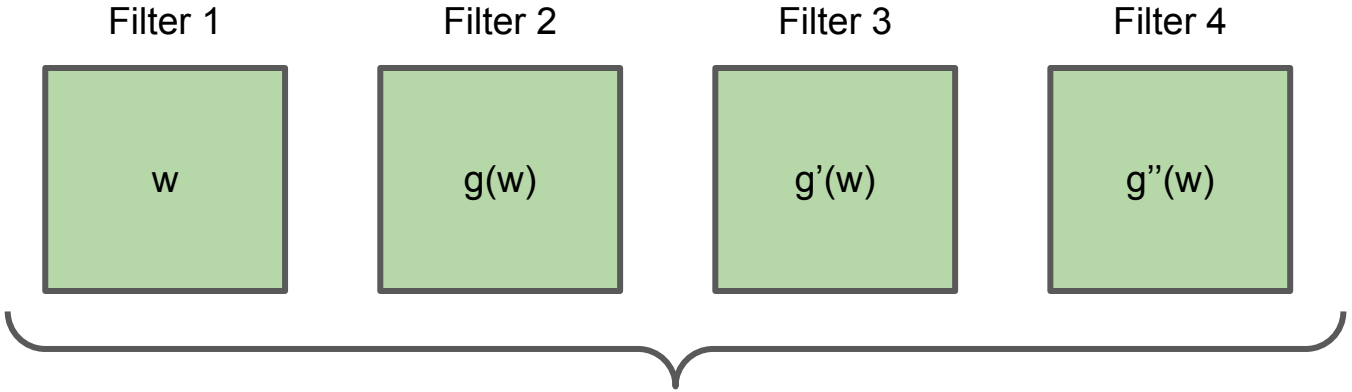
Lemma. If transformations g, g', g'' belong to a unitary group G, and x, w are any vectors and $T(x) = \max\langle x, g(w)\rangle$ for $\forall\ g \in G$, then

$T(x) = T(g(x))$

i.e. for any unitary group of transformations G, the max of the dot product with all elements of the group will be **invariant to the transformations in the group.**

Towards More General Invariances

More complex transformation between filters results in general invariances



Lemma. If transformations g, g', g'' belong to a unitary group G, and x, w are any vectors and $T(x) = \max\langle x, g(w)\rangle$ for $\forall\ g \in G$, then

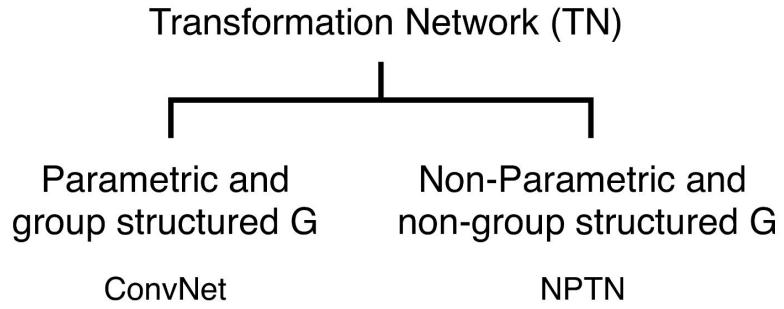
$T(x) = T(g(x))$

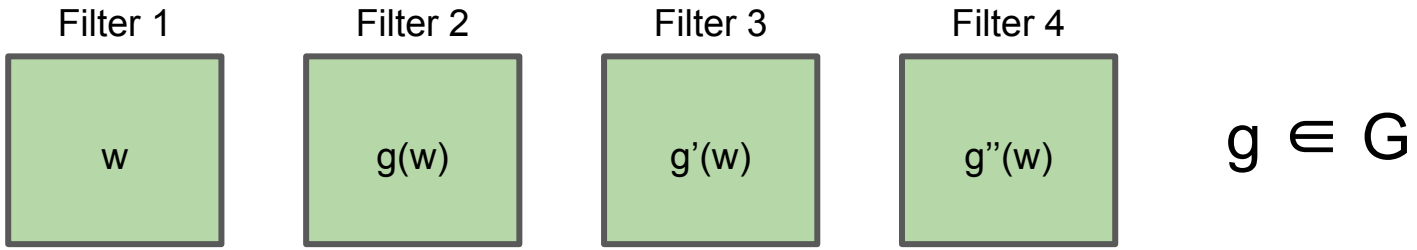
If G is allowed to be more general (not limited to being just translation), more **general invariances can be learnt.**

The Transformation Network Paradigm

Towards Generalizing ConvNets

The **Transformation Network (TNs)** framework is a generalization of convolutional architectures which models *networks which can be invariant to transformations in a set G.*





The Outline

Chapter 1
The Invariance Problem

Chapter 2
Generalizing Convolution Neural Networks

Chapter 3
Non-Parametric Transformation Networks

Chapter 4
Permanent Random Connectome Networks

The Chapter Summary

Chapter 2
Generalizing Convolution Neural Networks

- Convolutions are motivated for two distinct reasons: **weight sharing** and **translation invariance**.
- Convolutions can be seen as **dot-product with translated filters.**
- **Unitary groups** allow us to model perfect invariance.
- The proposed Transformation Network framework **allows a direct generalization** to non-parametric groups.

The Outline

Chapter 1
The Invariance Problem

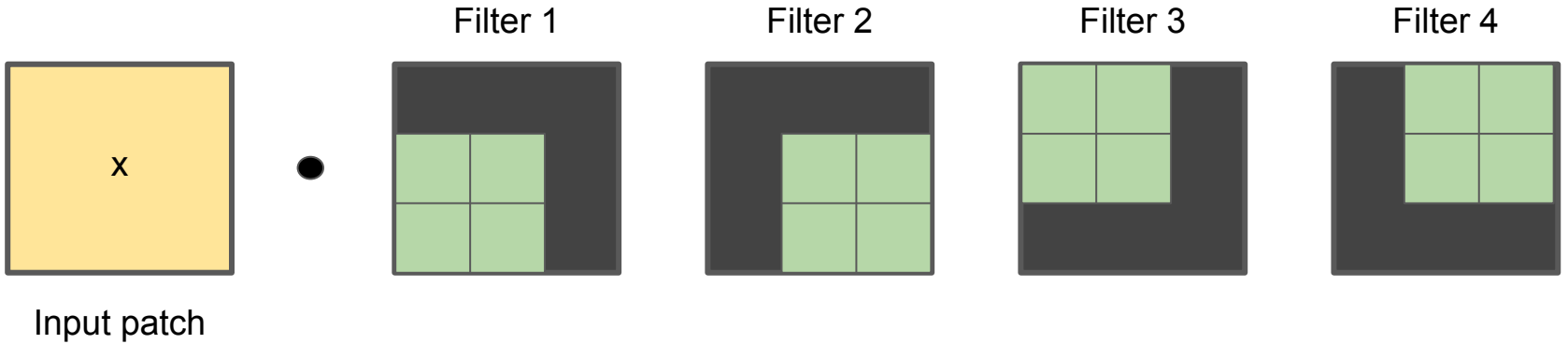
Chapter 2
Generalizing Convolution Neural Networks

Chapter 3
Non-Parametric Transformation Networks

Chapter 4
Permanent Random Connectome Networks

Non-parametric Transformation Networks

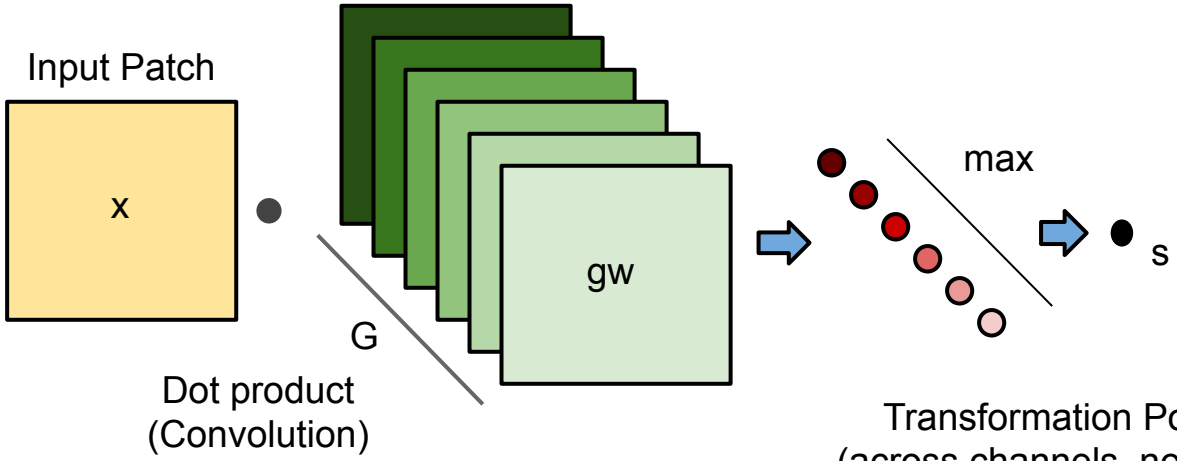
Single Channel Architecture



Non-parametric Transformation Networks

Single Channel Architecture

Dot product with multiple filters is computed. These filters **model the transformation** to be invariant towards.
Eg. translated filters -> translation invariance



Dot product with a filter is a **convolution** when *spatial weight sharing* is enforced.

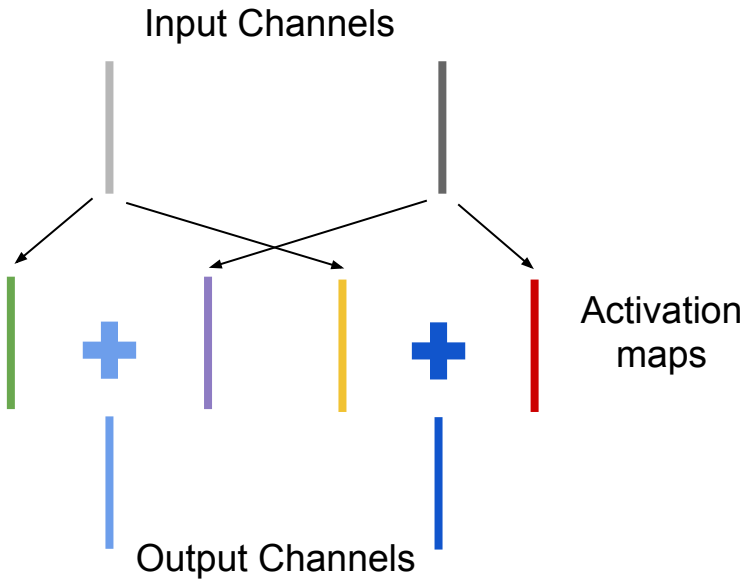
Non-parametric Transformation Networks

The Vanilla ConvNet Architecture

Consider a **vanilla ConvNet** with 2 input channels and 2 output channels.

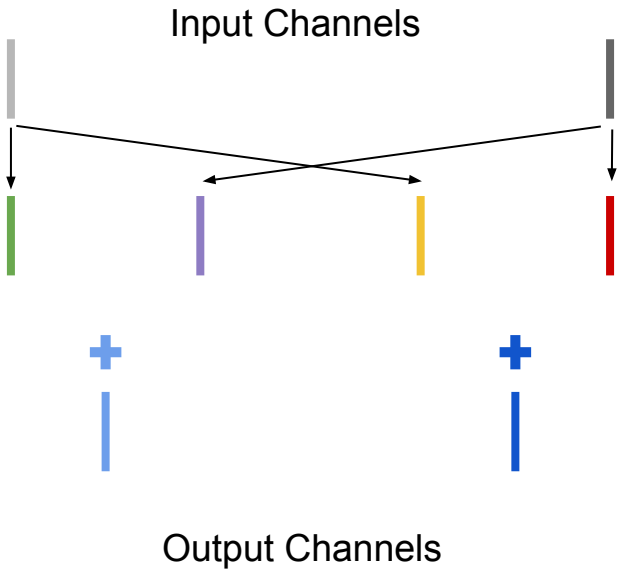
2 inputs and 2 outputs imply 4 2D filters.

Arrows indicate convolution with a single filter/kernel.



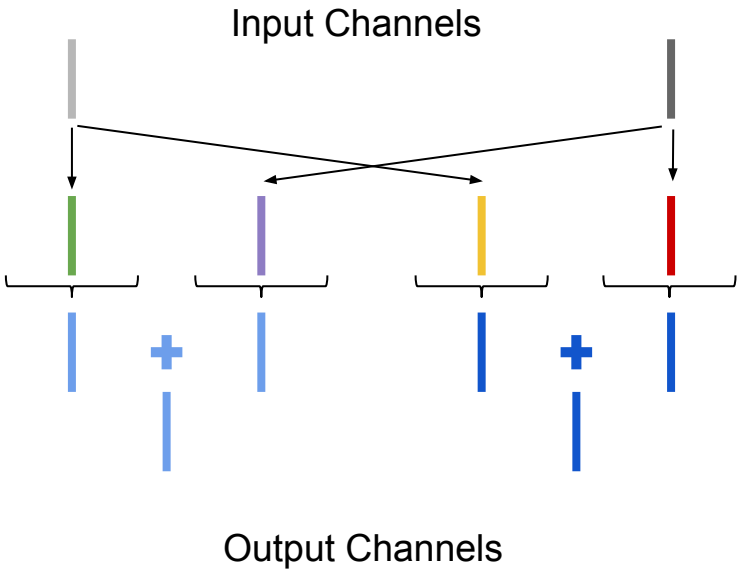
Non-parametric Transformation Networks

Towards a Multi-Channel Architecture



Non-parametric Transformation Networks

Towards a Multi-Channel Architecture

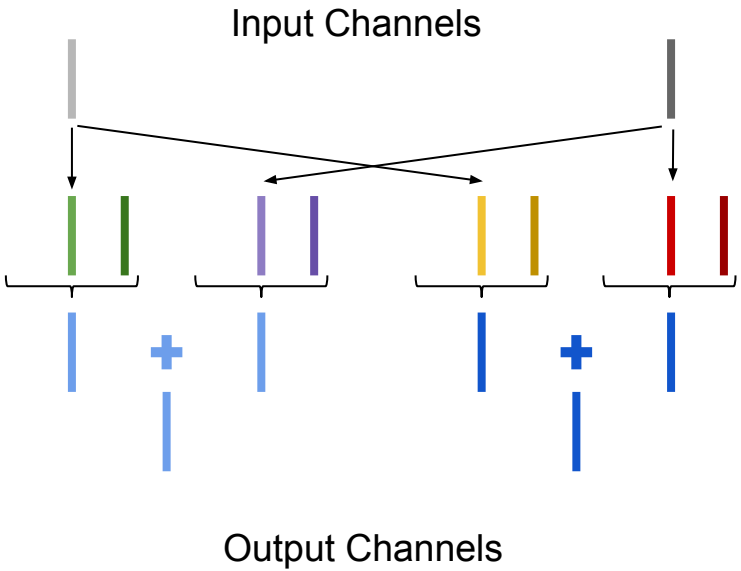


$|G| = 1$

The number of filters we pool over.

Non-parametric Transformation Networks

Towards a Multi-Channel Architecture

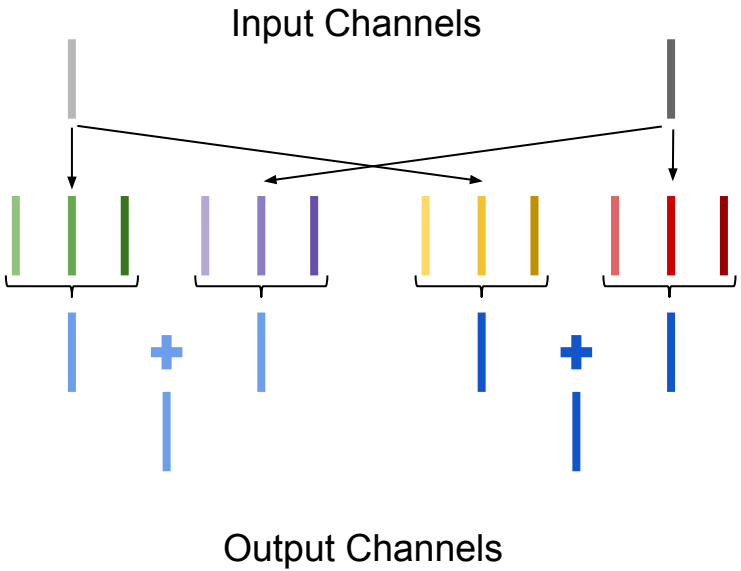


$|G| = 2$

The number of filters we pool over.

Non-parametric Transformation Networks

Towards a Multi-Channel Architecture

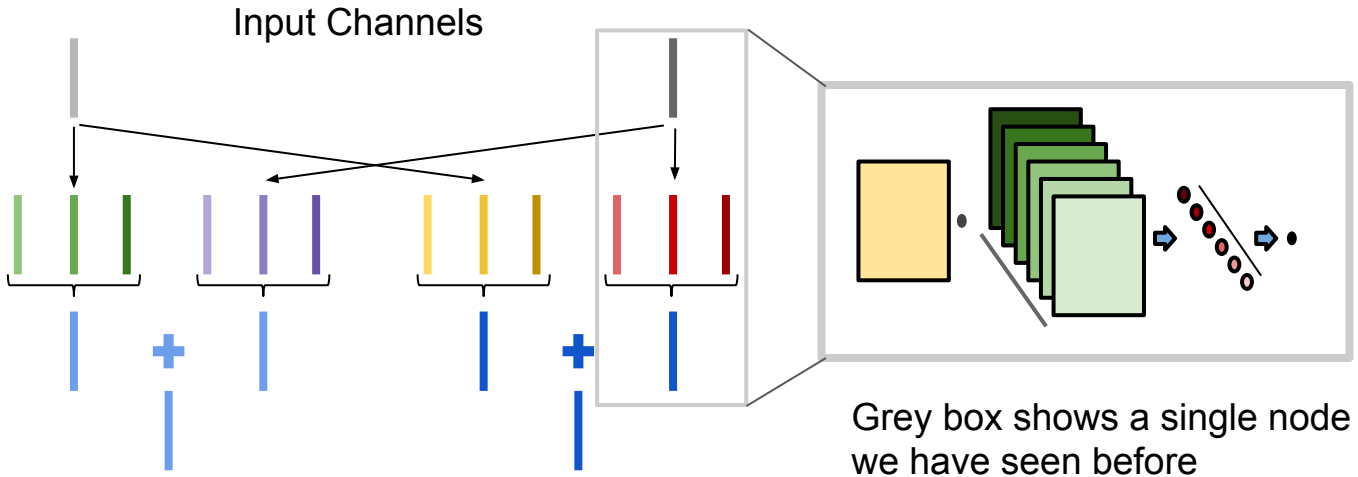


$|G| = 3$

The number of filters we pool over.

Non-parametric Transformation Networks

Towards a Multi-Channel Architecture

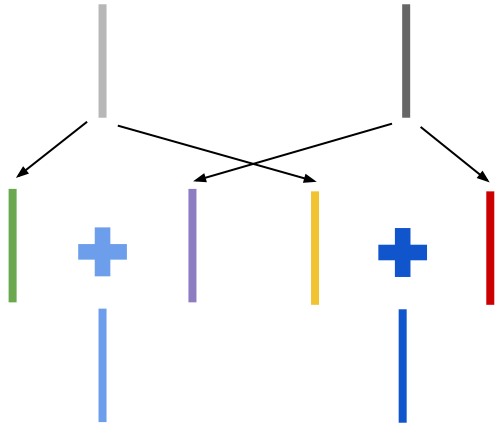


NPTN with |G|=3

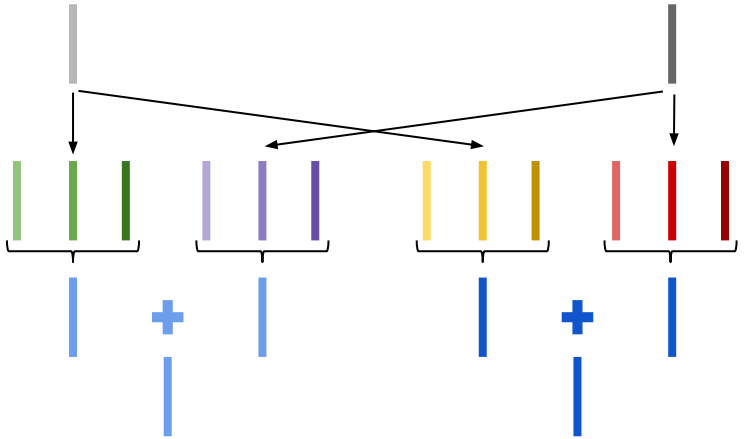
Grey box shows a single node that we have seen before

Non-parametric Transformation Networks

Towards a Multi-Channel Architecture



Vanilla CNN

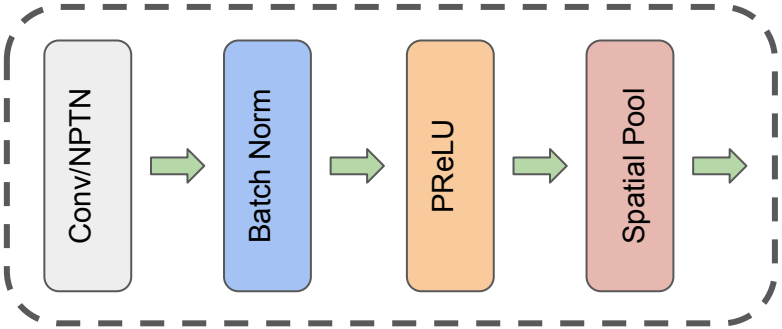


NPTN

Experiments: CIFAR 10

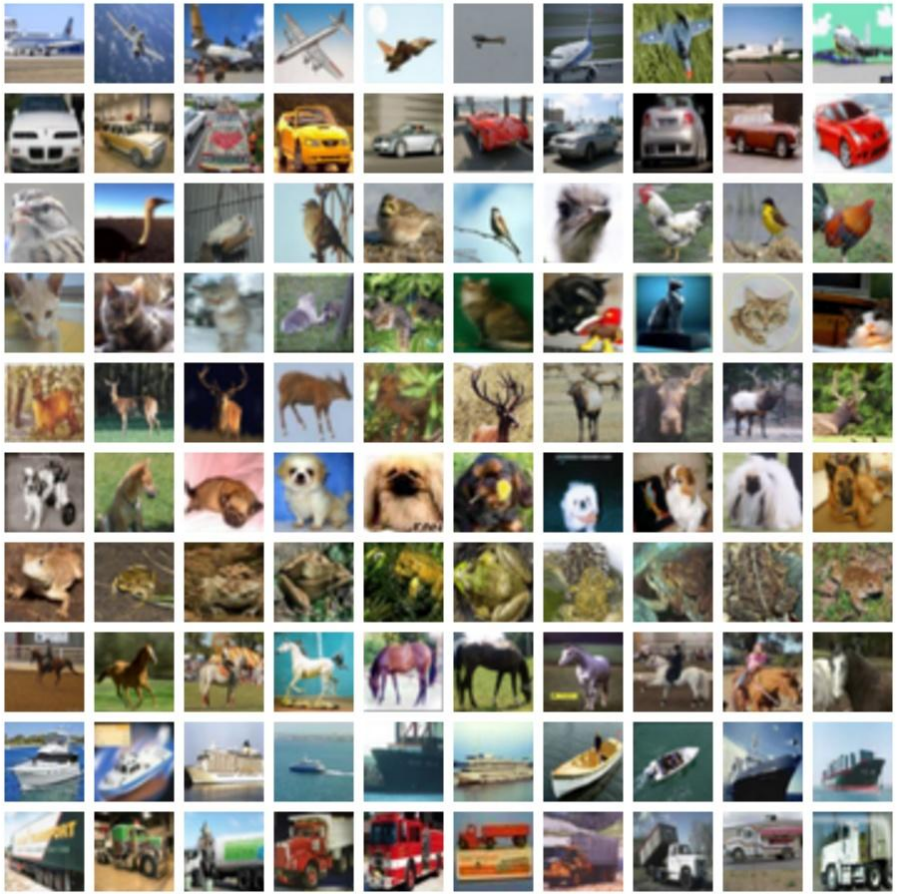
Experimental Setup: The Dataset

2 layered network followed by FC, each layer is



Conv baseline and NPTN had **same number of convolution filters.**

Width of NPTN layers was lowered while |G| was increased.



Experiments: CIFAR 10

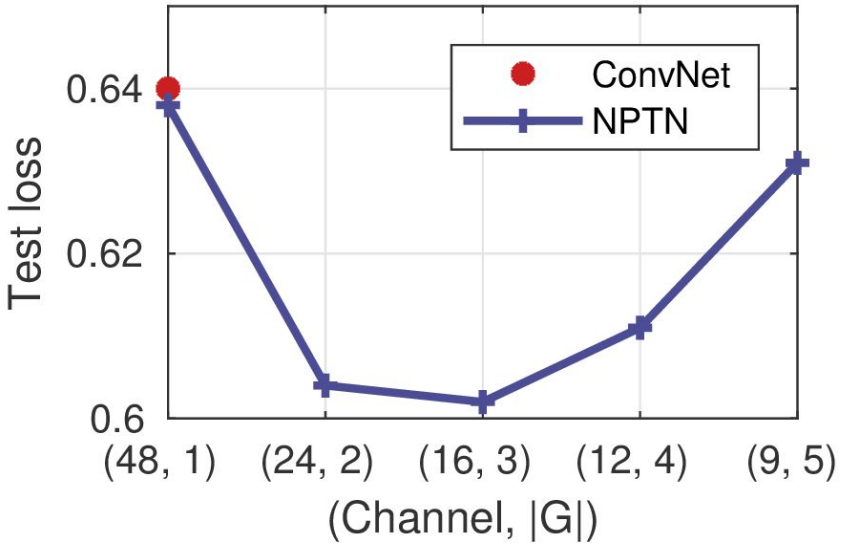
Results: Higher |G| provides better generalization

All networks have the **same number of convolution filters.**

ConvNet has slightly larger number of parameter in the FC layer.

As |G| increased, performance generally **increased.**

Recall that |G| is the number of filters to pool within each NPTN channel.



Experiments: Transformed MNIST

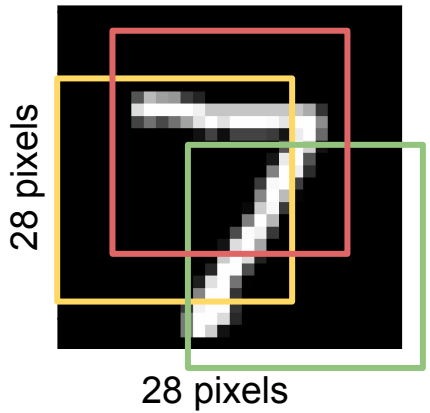
Experimental Setup: Evaluation Protocol

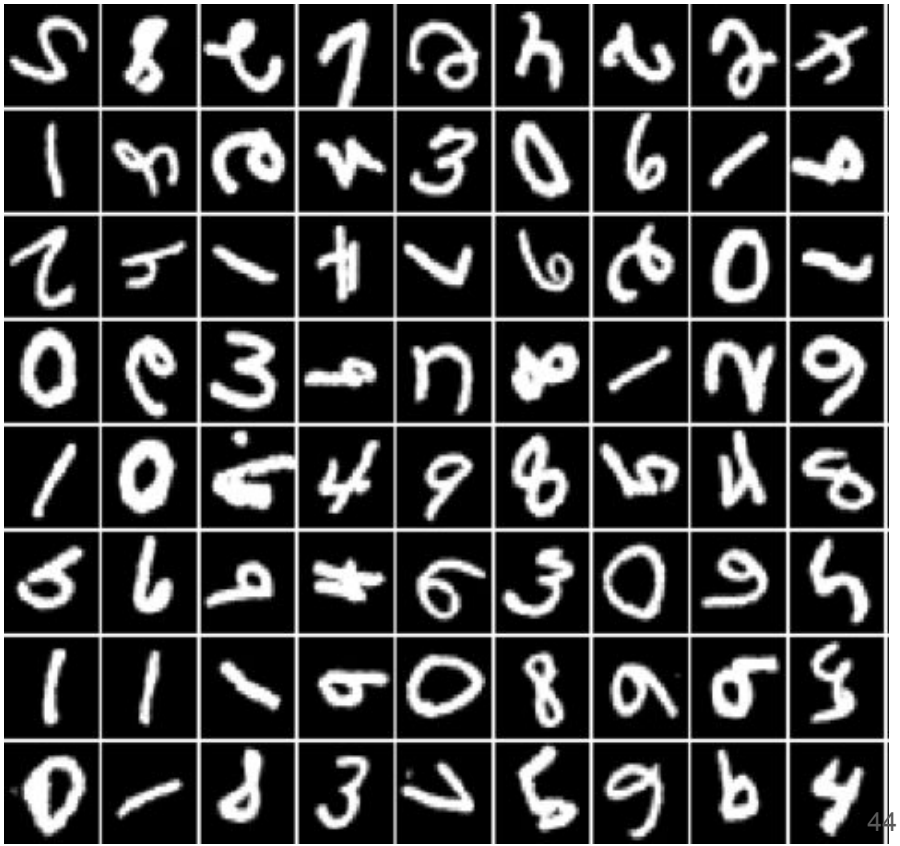
During training *and* testing, each digit was randomly **rotated** or **translated** with increasing range.

Rotation: 0, 30, 60, 90 degrees

Translation: 0, 4, 8, 12 pixels

Image padded, translated/rotated and cropped

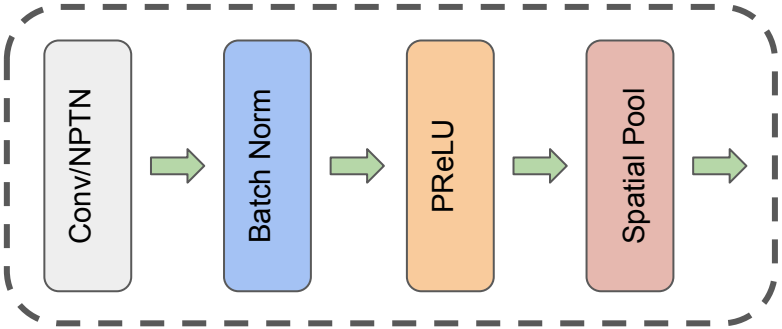




Experiments: Transformed MNIST

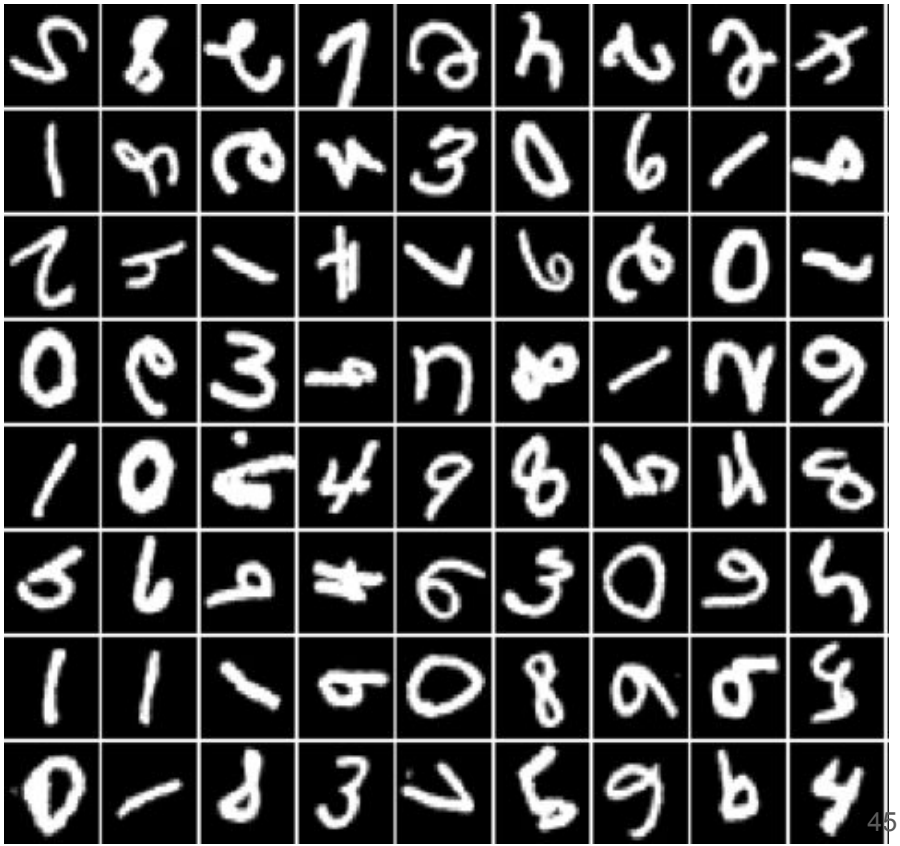
Experimental Setup: The Networks

2 layered network followed by FC, each layer is



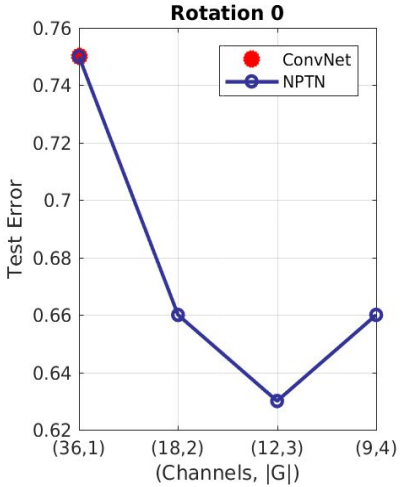
Conv baseline and NPTN had **same number of convolution filters.**

Width of NPTN layers was lowered while |G| was increased.

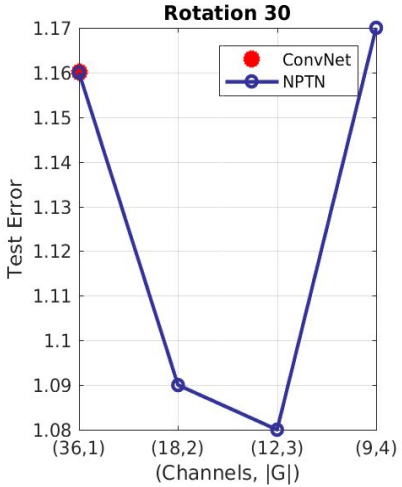


Experiments: Transformed MNIST

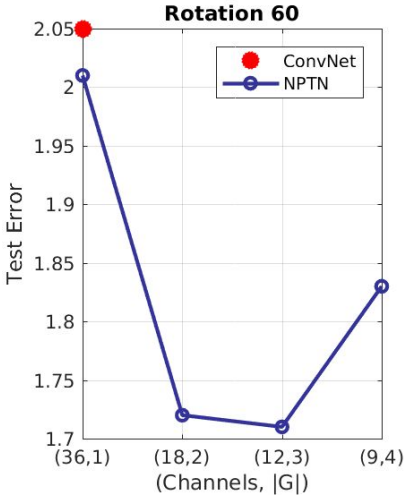
Results: NPTNs can solve the motivating MNIST example



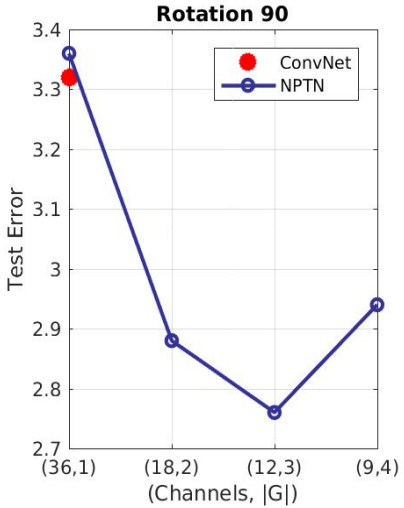
Δ=0.13



Δ=0.08



Δ=0.35



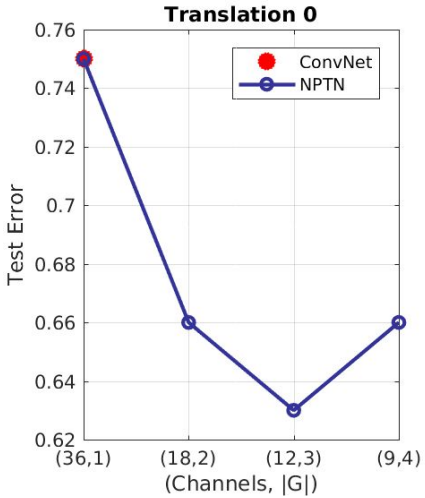
Δ=0.60

Test error generally **decreases with |G|**. Differences in loss is larger for larger transformations.

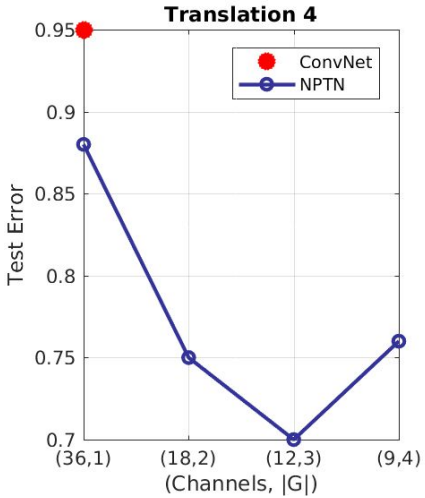
Recall that |G| is the number of filters to pool within each NPTN channel.

Experiments: Transformed MNIST

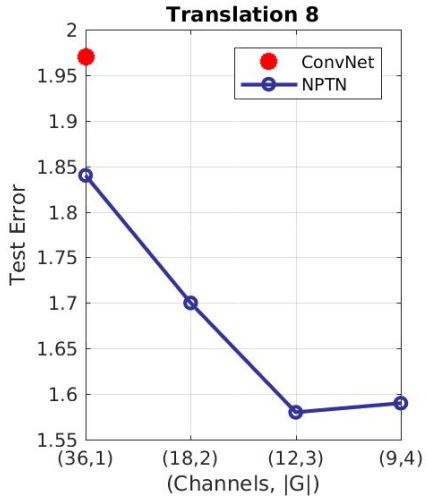
Results: NPTNs can solve the motivating MNIST example



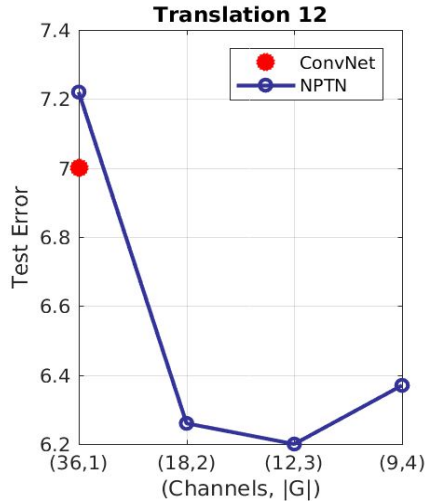
Δ=0.13



Δ=0.25



Δ=0.39



Δ=0.80

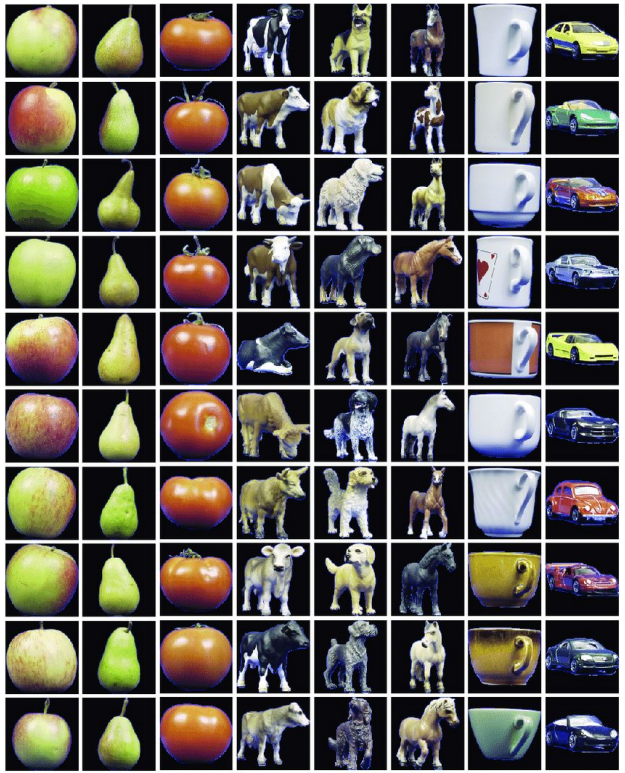
Test error generally **decreases with |G|**. Differences in loss is larger for larger transformations.

Recall that |G| is the number of filters to pool within each NPTN channel.

Experiments: ETH-80

Experimental Setup: The Dataset

8 image categories
10 instances of object/category
41 images per instance
2,300 training images, **980** test images
50 x 50 images
Primary nuisance transformation: **3D pose**



B. Leibe and, B.Schiele. Analyzing Appearance and Contour Based Methods for Object Categorization. CVPR, vol.2, pp. 409-415, 2003

Experiments: ETH-80

Results: NPTNs generalize better

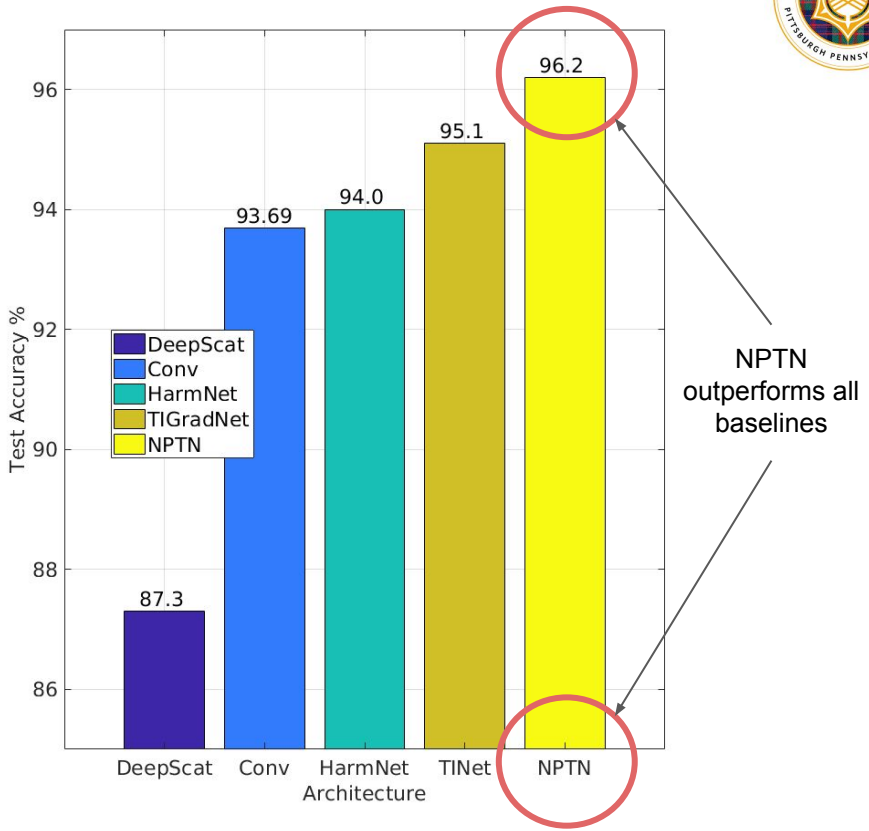
Each model had a similar architecture with **1.4 million parameters.**

NPTN architecture formed by replacing Conv layers with NPTN layers.

Same number of filters b/w Conv and NPTN.

NPTN outperforms other invariance motivated architectures while being simpler.

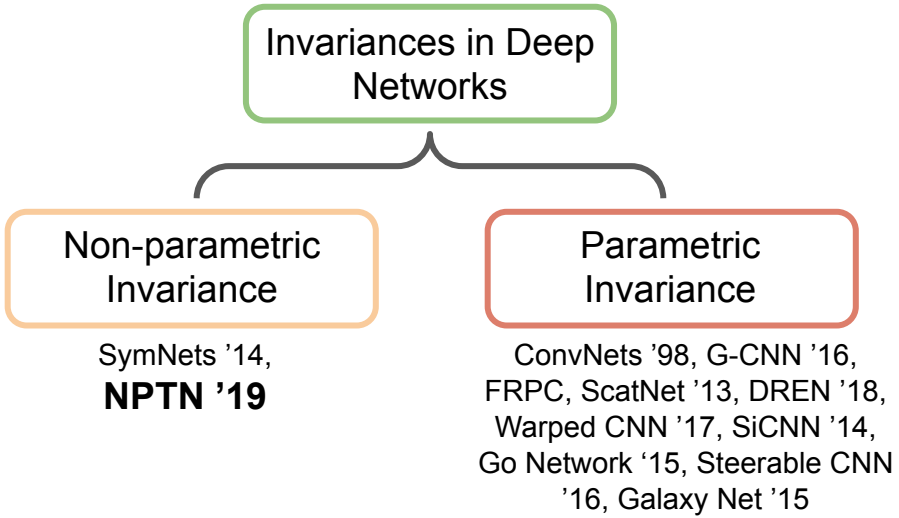
STN achieves 45.1%. All results averaged over 10 runs.



Khasanova, Renata, and Pascal Frossard. "Graph-based isometry invariant representation learning." In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1847-1856. JMLR. org, 2017.

NPTNs

A New Dimension in Network Design



Generalizes better with same number of parameters.

Designed to learn invariances from data

Provides a new dimension of network design

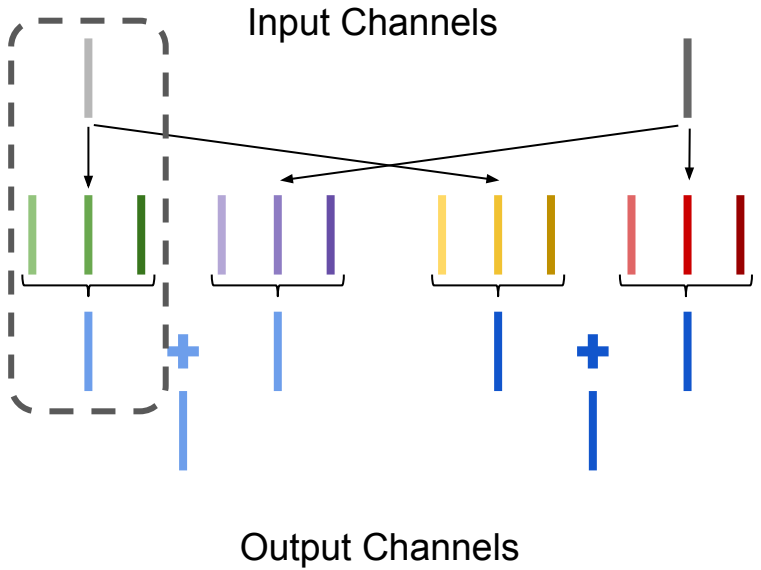
Pal, Dipan K., and Marios Savvides. "**Non-Parametric Transformation Networks for Learning General Invariances from Data**." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4667-4674. **2019**.

NPTNs

Some Limitations

Activation maps are pooled over that originate only from the **same channel.**

This limits the complexity of the transformations that can be pooled over.



NPTNs

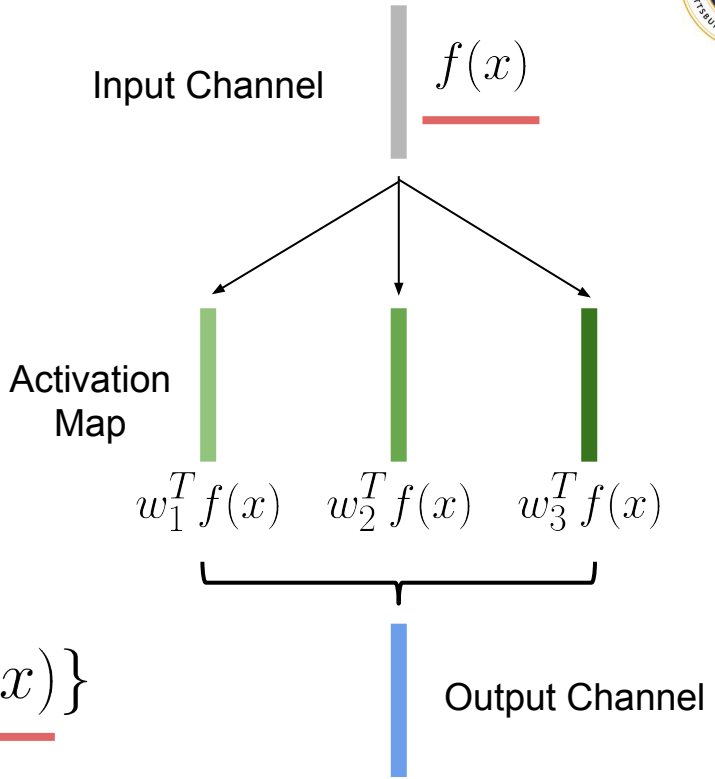
Some Limitations

Activation maps are pooled over that originate only from the **same channel.**

This limits the complexity of the transformations that can be pooled over.

$$max\{w_1^T f(x), w_2^T f(x), w_3^T f(x)\}$$

$$= max\{w^T f(x), g w^T f(x), g' w^T f(x)\}$$



Further, filter computation is not reused. Wasted if output does not win.

The Outline

Chapter 1
The Invariance Problem

Chapter 2
Generalizing Convolution Neural Networks

Chapter 3
Non-Parametric Transformation Networks

Chapter 4
Permanent Random Connectome Networks

The Chapter Summary

- **NPTNs can learn non-parametric transformations** from data itself and provide benefits to supervised learning.
- Provide **alternate dimension for network architecture** development.
- Only **linear invariance** within a layer and **computation is not reused.**

Chapter 3
Non-Parametric
Transformation Networks

The Outline

Chapter 1
The Invariance Problem

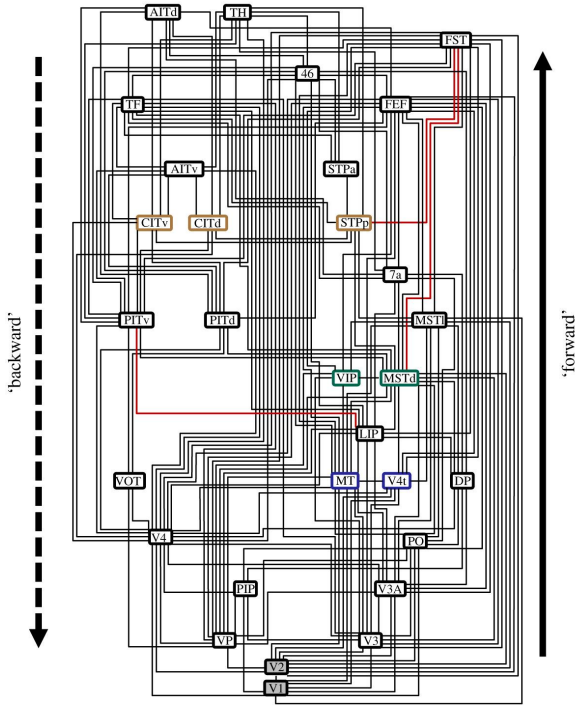
Chapter 2
Generalizing Convolution Neural Networks

Chapter 3
Non-Parametric Transformation Networks

Chapter 4
Permanent Random Connectome Networks

Connectomes in the Brain

Complex yet Structured



Connectomes in the Brain

Existence of Locally Random Connectomes

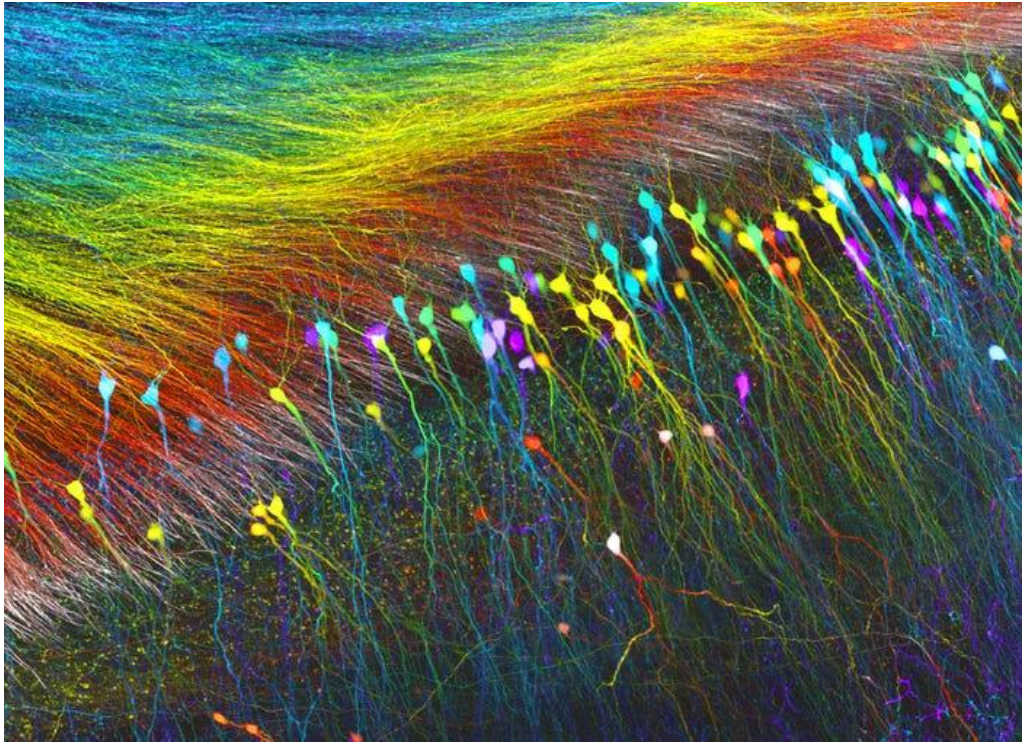
Evidence has shown the *lack of precise pathways* for transport of gradients (Grossberg ‘87, Mazzoni ‘91, Xie ‘03).

Connectomes in the Brain

Existence of Locally Random Connectomes

Evidence has shown the *lack of precise pathways* for transport of gradients (Grossberg ‘87, Mazzoni ‘91, Xie ‘03).

Permanent random local unstructured connections are common in the cortex (Corey ’12, Schottdorf ’15).



Connectomes in the Brain

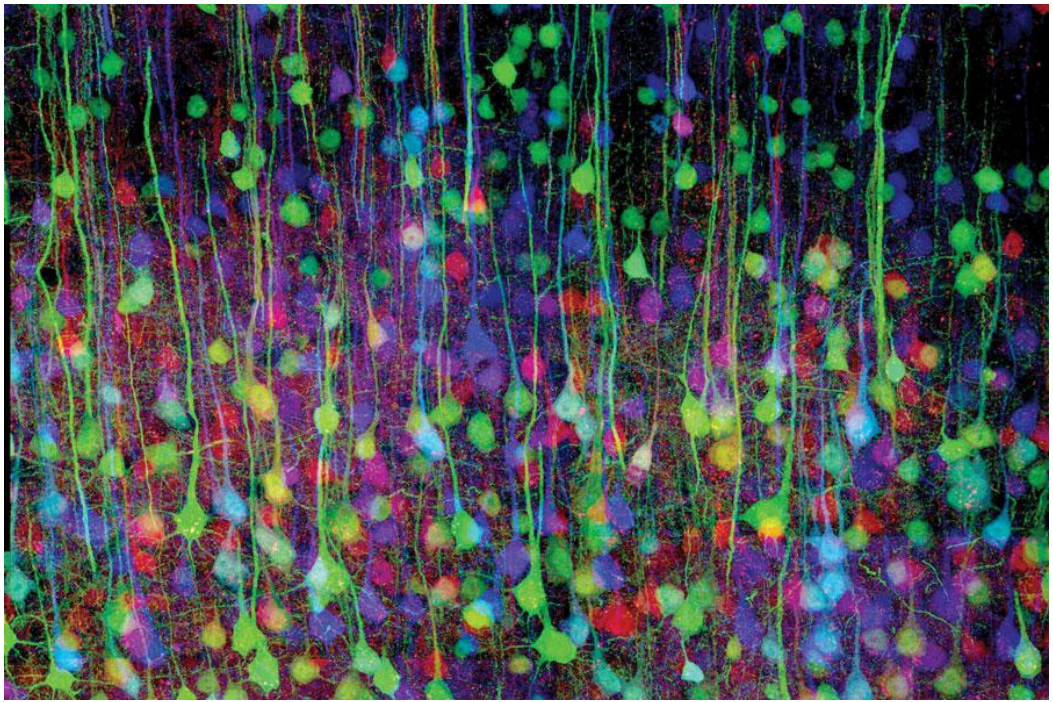
Existence of Locally Random Connectomes

Evidence has shown the *lack of precise pathways* for transport of gradients (Grossberg ‘87, Mazzoni ‘91, Xie ‘03).

Permanent random local unstructured connections are common in the cortex (Corey ’12, Schottdorf ’15).

Orientation selectivity can arise in the visual cortex even through local permanent random connections (Hansel ’12).

Intriguing Question: *How would permanent random connections be motivated or used in networks architectures?*



Invariance to Multiple Transformations

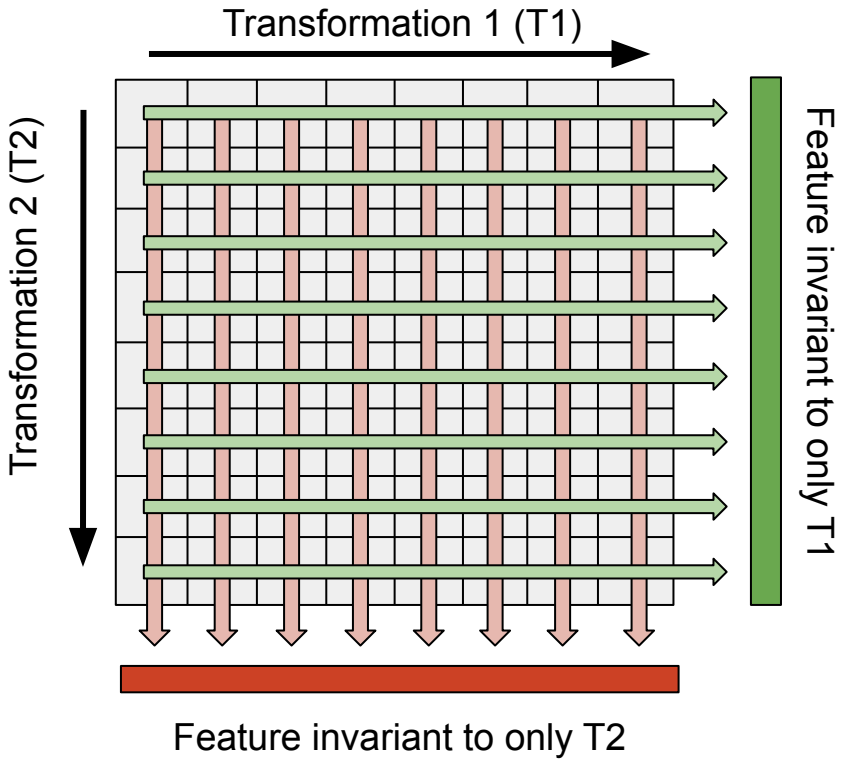
One type of Invariance per Representation

Consider a set of transformed templates with two transformations.

Each pooling operation in a representation invariant to the corresponding transformation.

Potential problem: Feature representation scales *linearly* with number of transformations.

Desired: A **single** representation that is even partially invariant to multiple transformations.



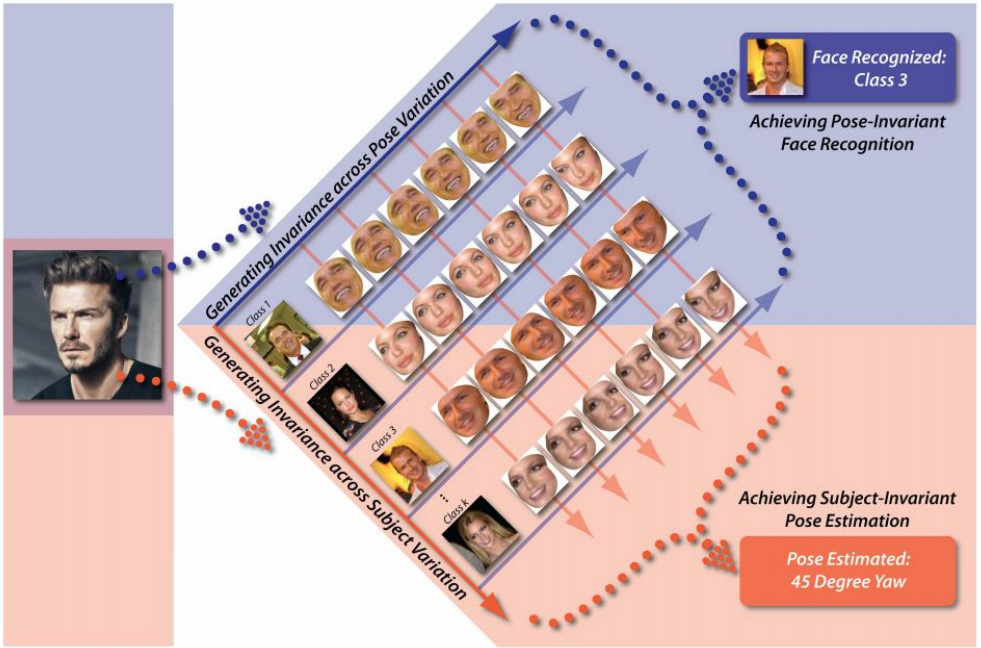
Discriminative Invariant Kernel Features

Face Recognition and Pose Estimation within a Single Framework

Framework extracted two features with **complimentary invariances.**

Achieved **state-of-the-art results on LFW** (prior to deep learning), using raw pixels.

Theoretical result: DIKF filters form a group of transformed filters under a group in the kernel space, hence group invariance holds exactly.



Pal, Dipan K., Felix Juefei-Xu, and Marios Savvides. "**Discriminative invariant kernel features: a bells-and-whistles-free approach to unsupervised face recognition and pose estimation.**" In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5590-5599. **2016**. **(spotlight presentation)**

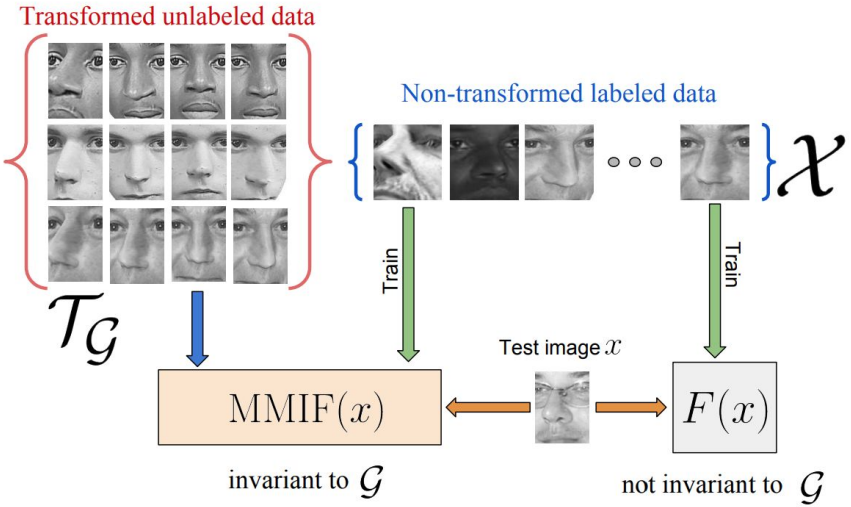
Max-Margin Invariant Features

Invariant Kernel Feature Extraction using SVMs

Semi-supervised learning

Learn transformation invariance from **unlabelled data**

First invariance theoretical guarantees on SVMs



Theorem 3.1. *(MMIF is invariant to learnt transformations)* $\text{MMIF}(x') = \text{MMIF}(gx') \;\; \forall x' \forall g \in \mathcal{G}$
where $\mathcal{G}$ *is observed only through* $\mathcal{T}_{\mathcal{G}} = \{gt_i \mid \forall g \in \mathcal{G}\}_{i=\{1,\dots M\}}$.

Pal, Dipan K., Ashwin Kannan, Gautam Arakalgud, and Marios Savvides. "Max-margin Invariant Features from Transformed Unlabelled Data." In *Advances in Neural Information Processing Systems*, pp. 1439-1447. 2017.

Invariance to Multiple Transformations

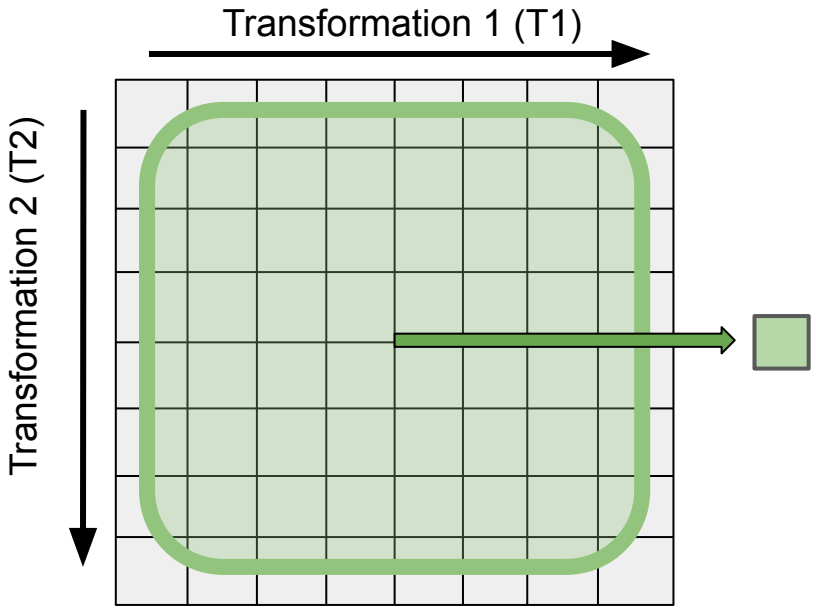
Pool Simultaneously to Learn Multiple Invariances?

A potential solution: Pool over multiple transformations simultaneously?

This operation leads to a 1D feature representation that is invariant but **loses discriminability.**

The representation is essentially ***useless***.

How would one preserve discriminability while ensuring invariance?



Invariance to Multiple Transformations

The Need for Pooling over Limited Range of Transformations

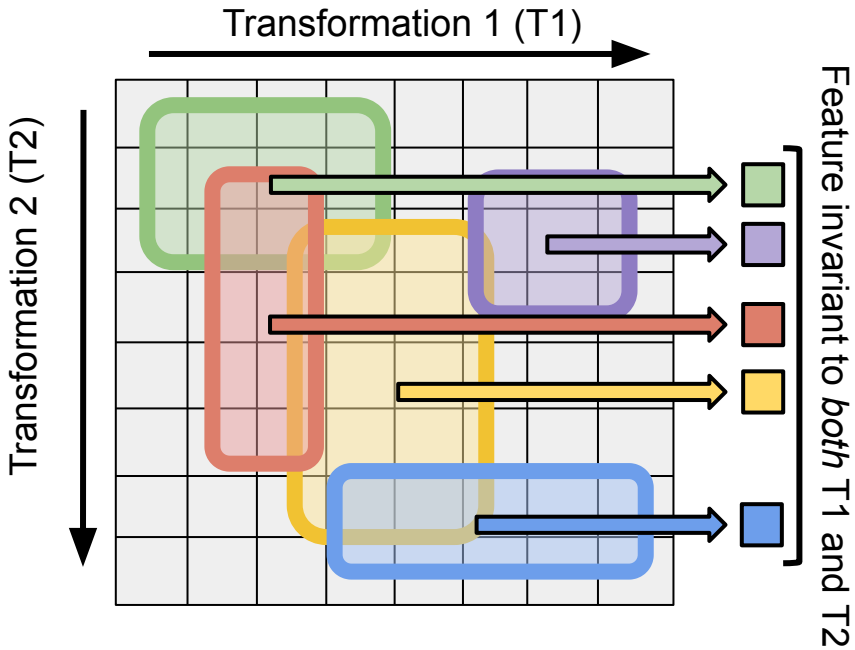
A potential solution: Pool over multiple transformations simultaneously *but with limited range, with multiple such supports.*

Each dimension pools over only *a limited support.*

Collectively, all dimensions:

- Provide **invariance to overlapping ranges** of transformations.
- Preserve **discriminability**.

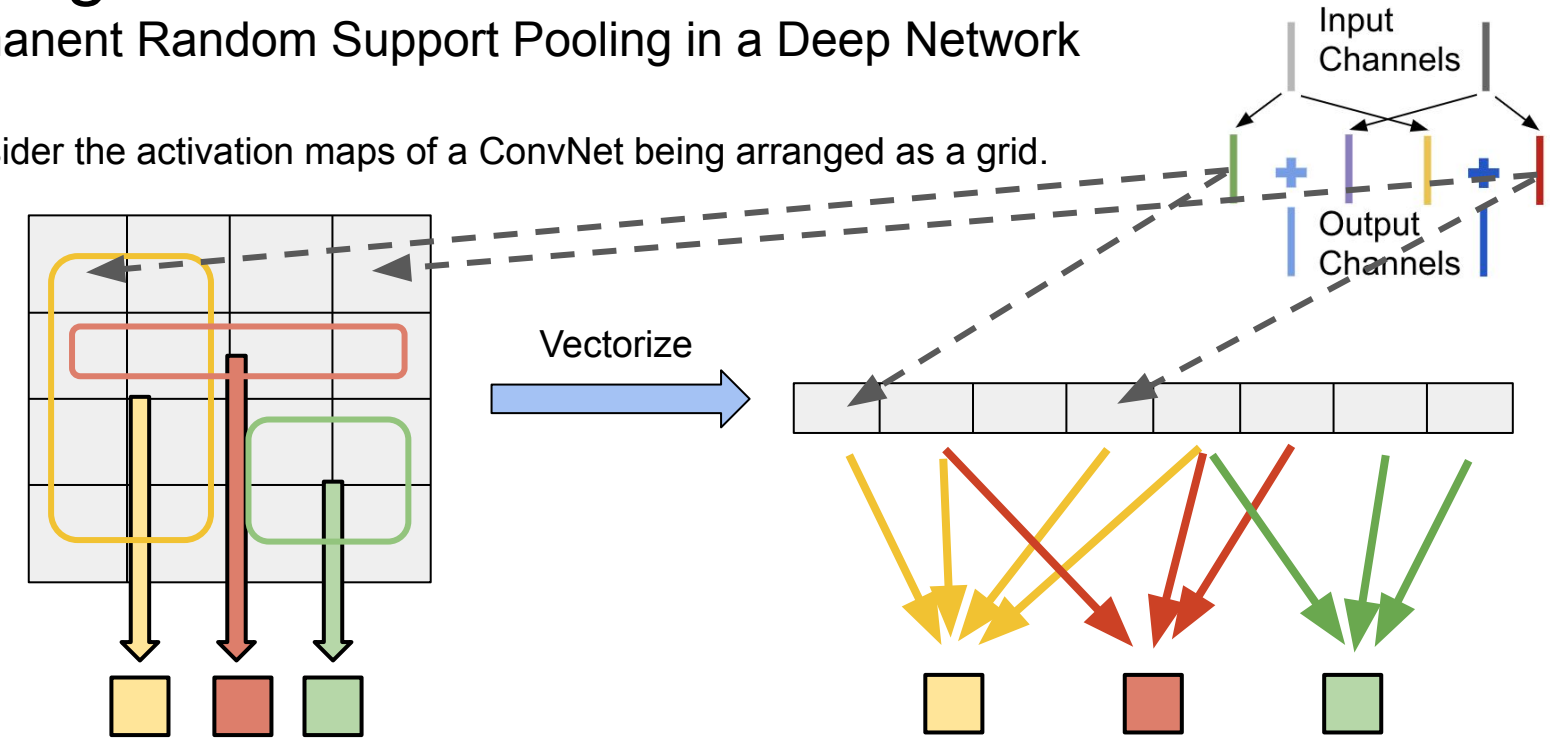
How to choose such supports?
An efficient solution: **Randomly at initialization!**



Emergence of Permanent Random Connectomes

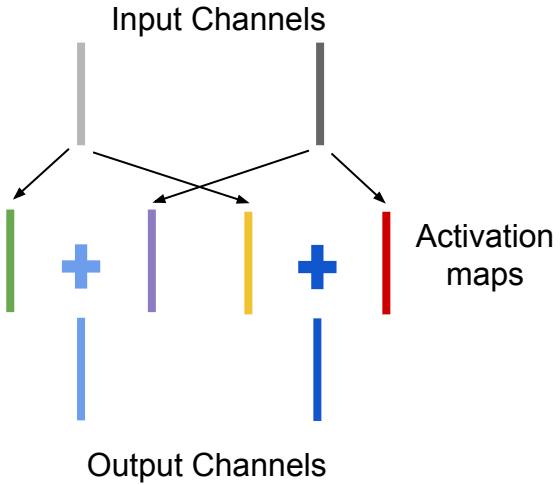
Permanent Random Support Pooling in a Deep Network

Consider the activation maps of a ConvNet being arranged as a grid.

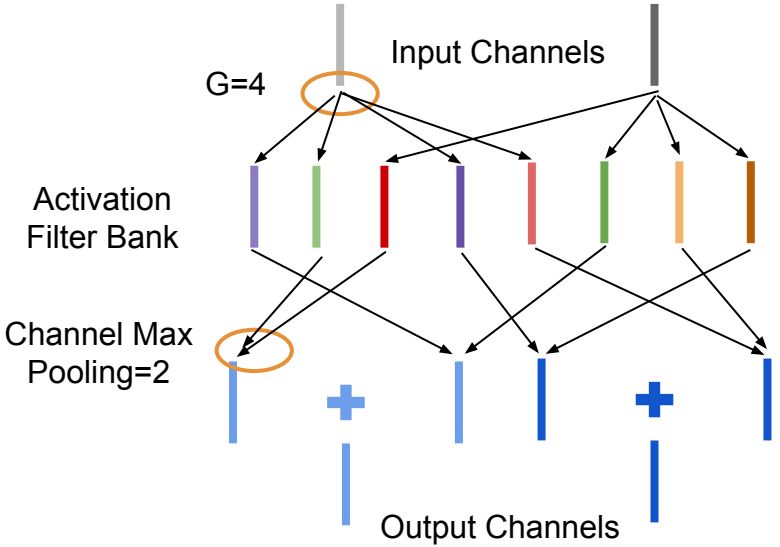


Permanent Random Connectome - NPTNs

A New Dimension in Network Design



Vanilla Conv layer



PRC-NPTN layer

Permanent Random Connectome - NPTNs

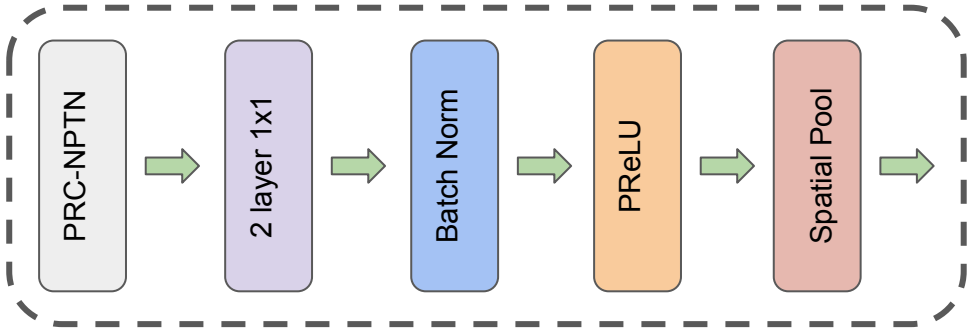
A New Dimension in Network Design

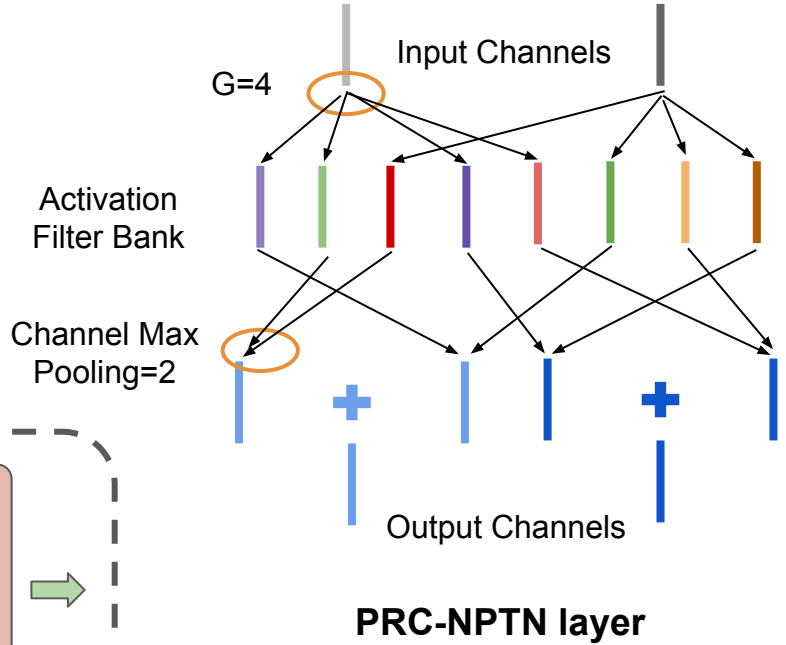
PRC-NPTNs are the first architecture to employ **permanent** and **random** connectomes.

Parameters: |G|, Channel Max Pool (CMP)

Typically used along with a 1x1 network downstream.

Random connections **do not ever change** once initialized, even through *training* and *testing*.





PRC-NPTN layer

Permanent Random Connectome - NPTNs

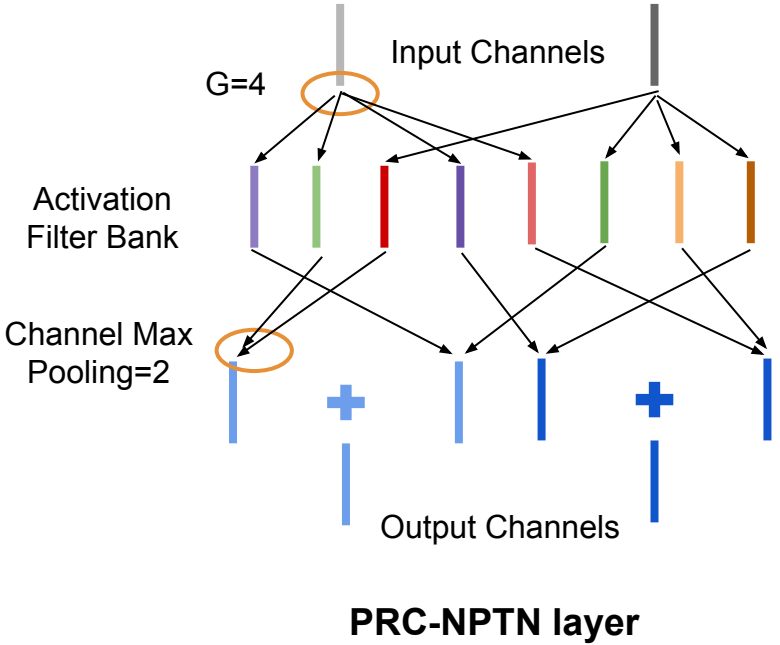
Provable Invariance Without Assuming Group Structure

Previous result required an assumption that the **transformations are group structured**, which is limiting.

We present a more general result to show invariance **without** assuming any group based structure.

Lemma: *For any two vectors x and w, if g is a **unitary transformation** that is sampled from some distribution such that T(x) = ⟨x, g(w)⟩follows a uniform distribution, then*

$$\text{Var}(\ \mathbf{max}(T(x))\ ) \leq \text{Var}(T(x))$$



PRC-NPTN layer

Experiments: Individually Transformed MNIST

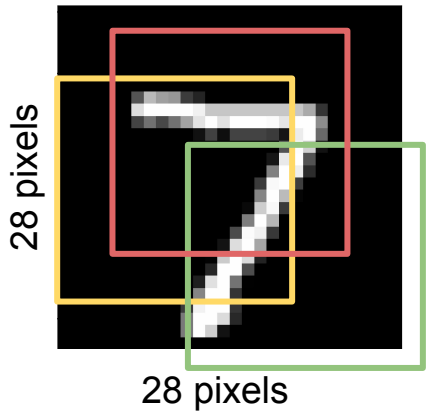
Experimental Setup: The Protocol

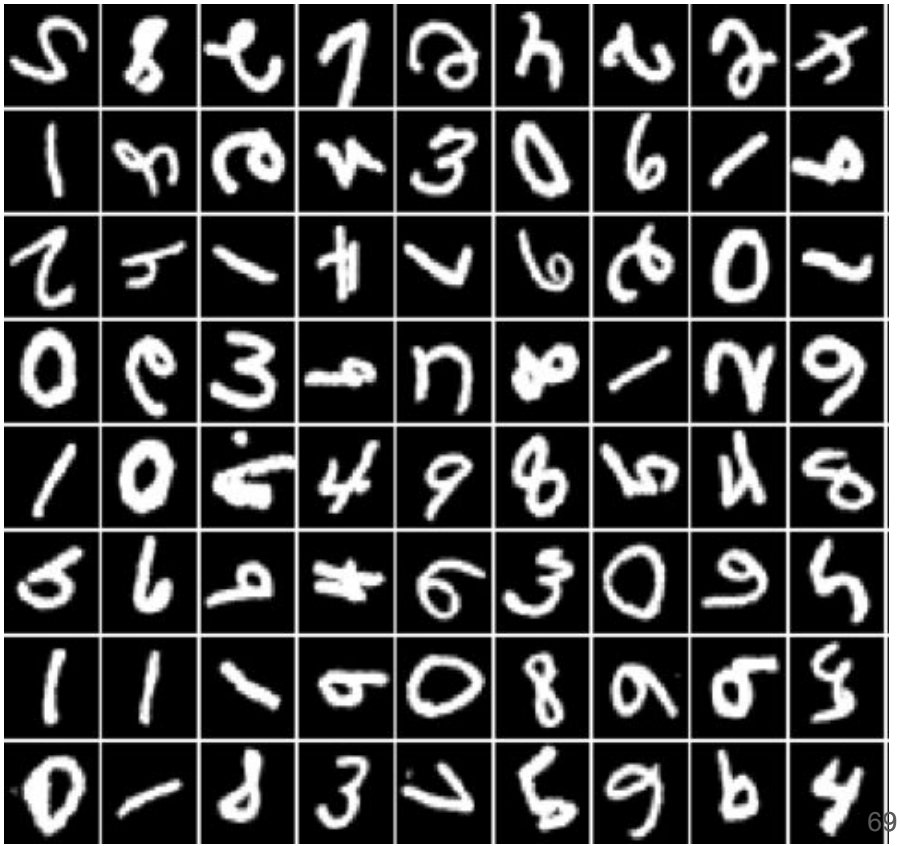
During training *and* testing, each digit was randomly **rotated** or **translated** with increasing range.

Rotation: 0, 30, 60, 90 degrees

Translation: 0, 4, 8, 12 pixels

Image padded, translated/rotated and cropped

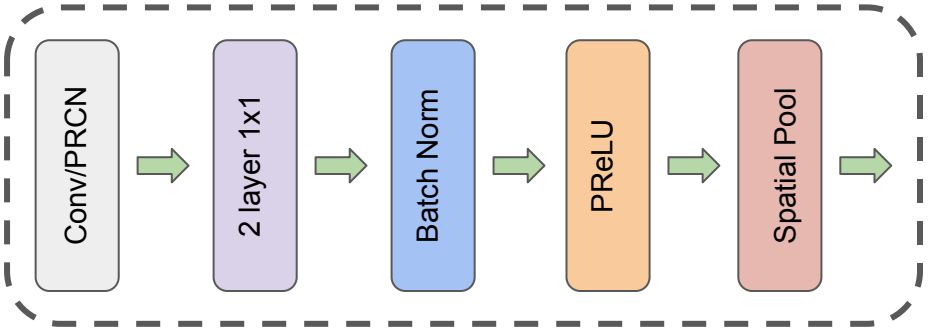




Experiments: Individually Transformed MNIST

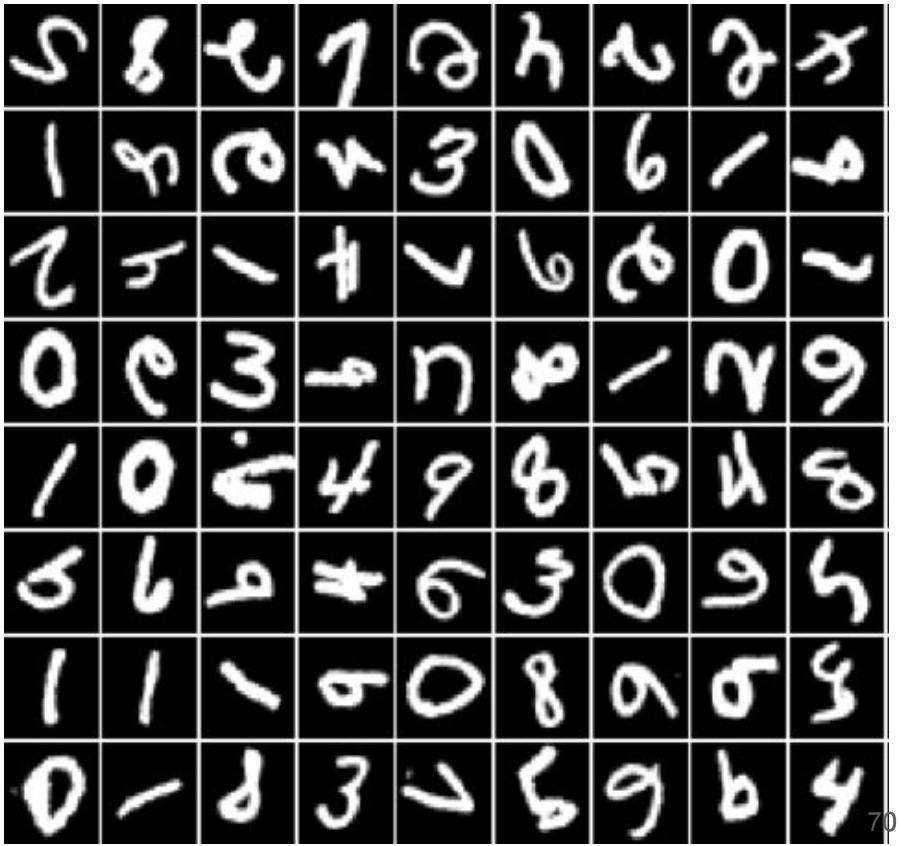
Experimental Setup: The Networks

2 layered network followed by FC, each layer is



Conv baseline, NPTN and PRC-NPTN had **same number of convolution filters.**

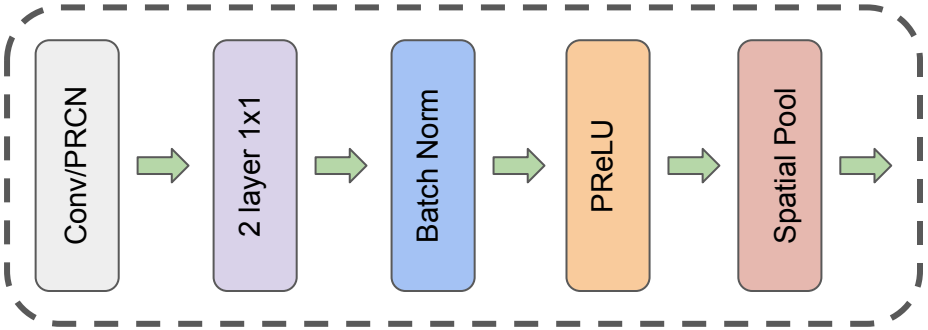
Width of PRC-NPTN layers was lowered while |G| was increased.



Experiments: Individually Transformed MNIST

Experimental Setup: The Networks

Each exp results averaged over **10 runs.**



Additional baselines:

- Conv with **two 1x1 layers**
- Conv with **512 channel width**

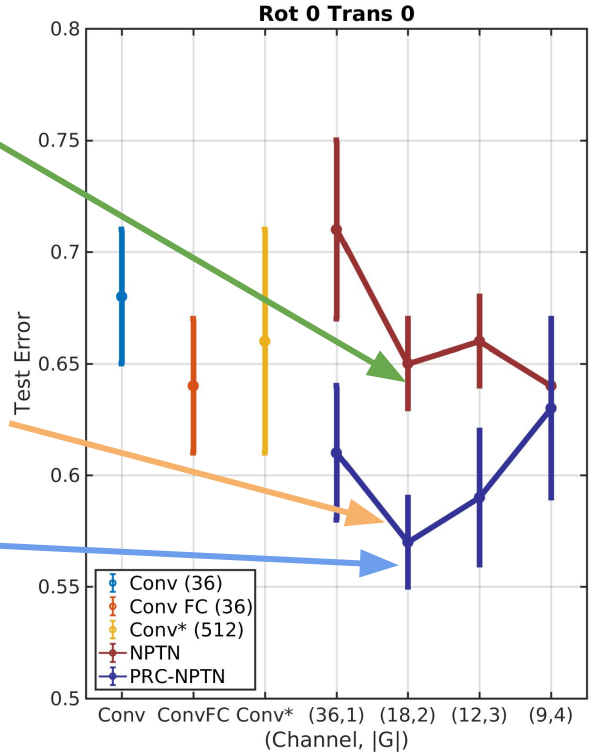
Experiments: Individually Transformed MNIST

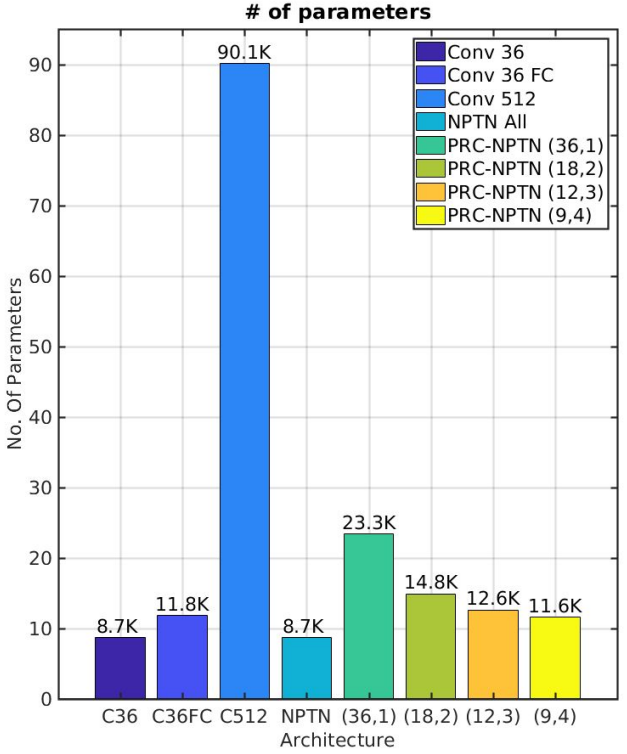
Results: Baseline, No Added Transformations

NPTN outperforms Conv (36) baseline.

PRC-NPTN significantly outperforms other baselines.

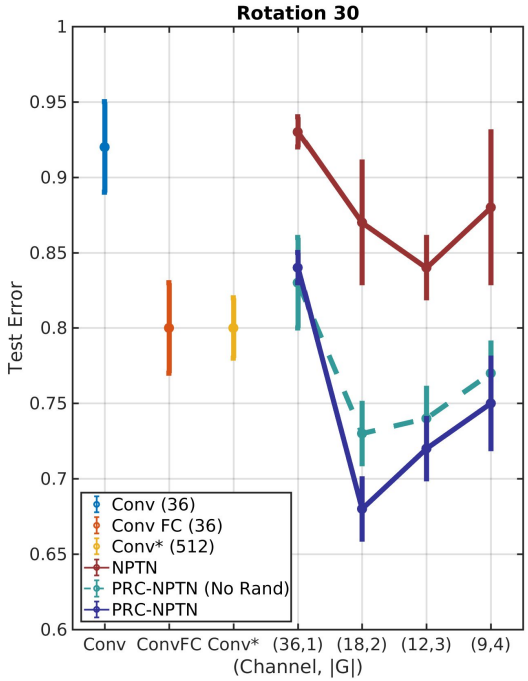
Small variance despite random connectomes

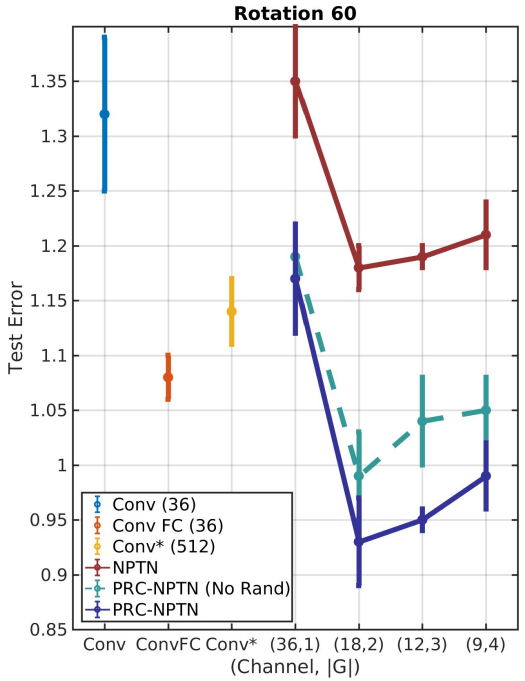


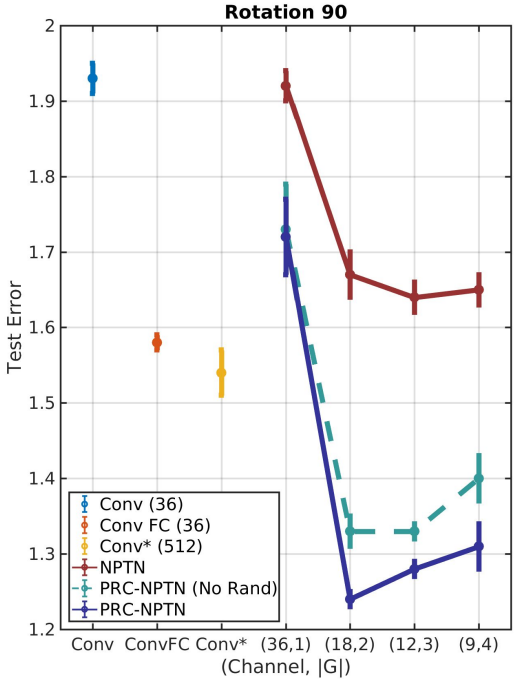


Experiments: Individually Transformed MNIST

Results: Added Individual Rotations

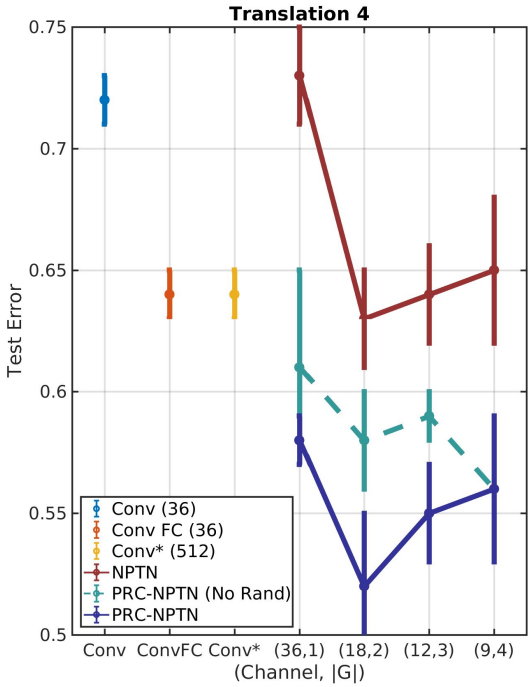


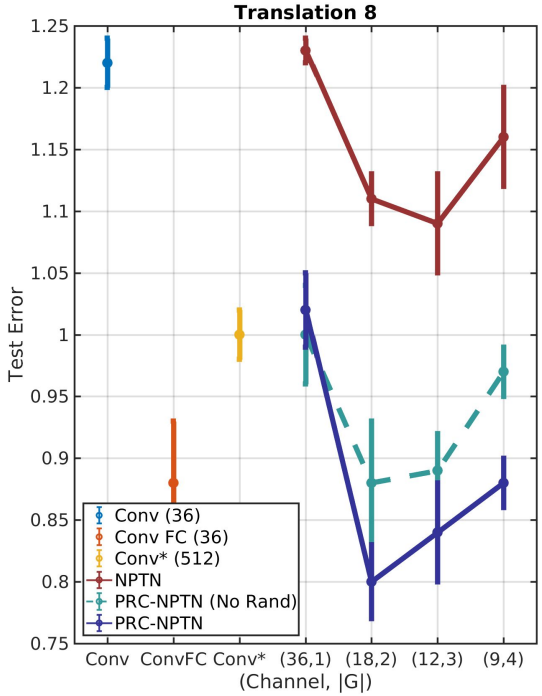


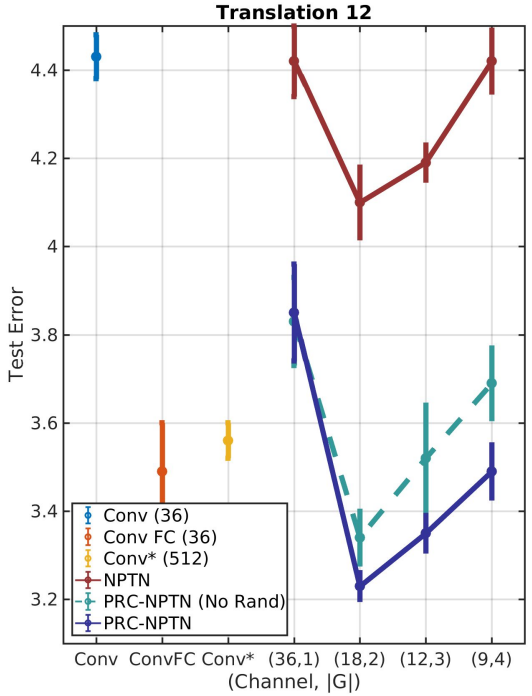


Experiments: Individually Transformed MNIST

Results: Added Individual Translations







Experiments: Simultaneous Transformed MNIST

Experimental Setup: The Protocol

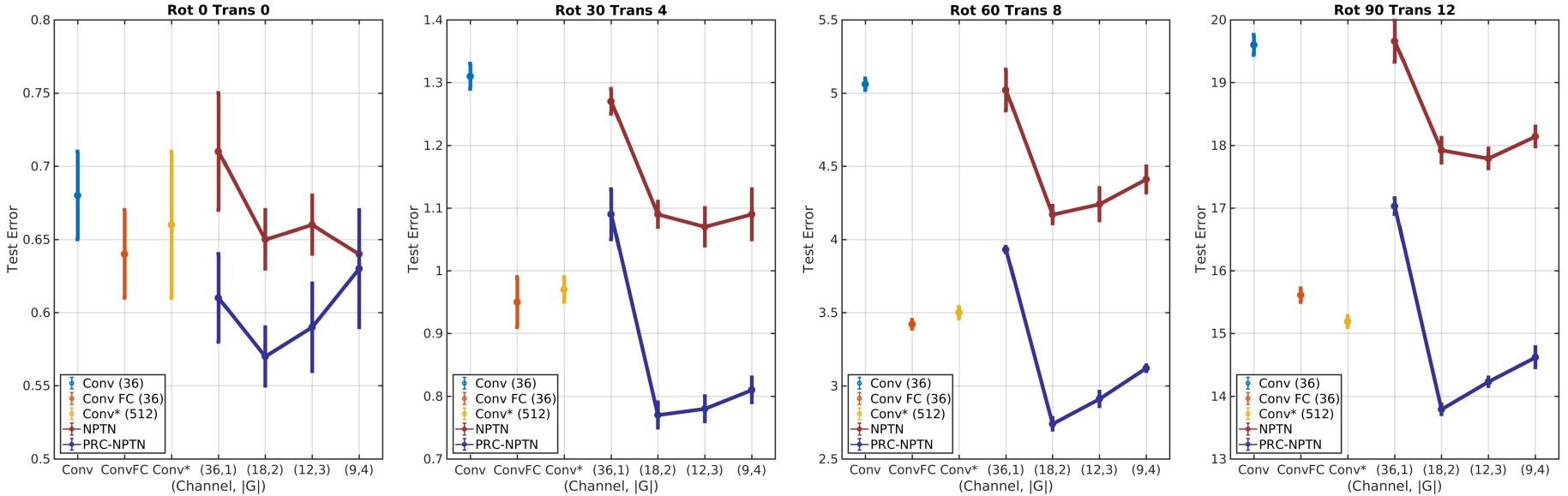
During training *and* testing, each digit was randomly **rotated AND translated** simultaneously with increasing range.

Rotation and Translation: 0° 0 pixels, 30° 4 pixels, 60° 8 pixels, 90° 12 pixels.

Complexity of the problem is exponentially higher than individual transformations.

Experiments: Simultaneous Transformed MNIST

Results: PRC-NPTNs Better Handle Increasing Complexity



Exponentially increasing complexity due to multiple transformations acting simultaneously

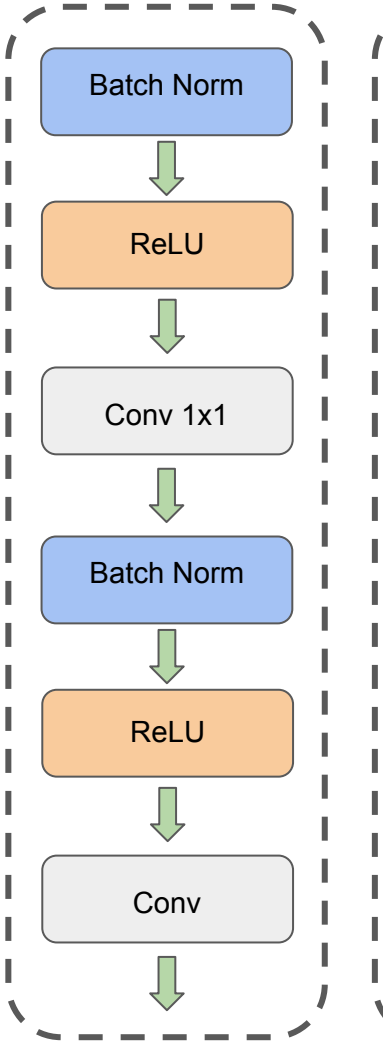
Experiments: CIFAR 10

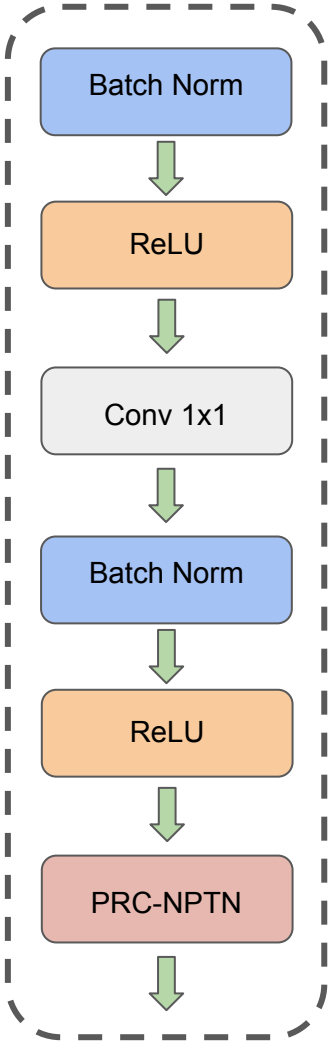
Experimental Setup: The Networks

Modified DenseNets to use PRC-NPTNs as the primary convolution layer.

Vanilla DenseNets were 12 layered with each layer as shown.

Vanilla DenseNet
x12
Growth Rate = 12





DensePRC-NPTN
x12
Growth Rate = 12

Conv baseline and PRC-NPTN had **same number of convolution filters and parameters.**

|G| = 12, CMP was varied from 1 through 4.

Results averaged over 5 runs.

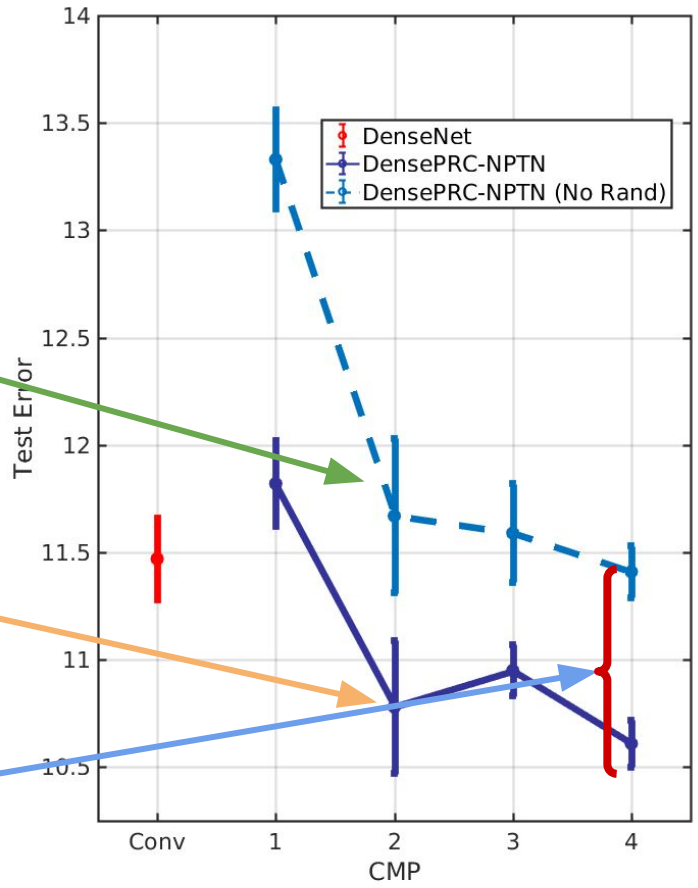
Experiments: CIFAR 10

Results: PRC-NPTN improve DenseNets

Increase in CMP generally increases performance

DensePRC-NPTN significantly outperforms vanilla DenseNets

7.5% reduction in error with **random** connectomes

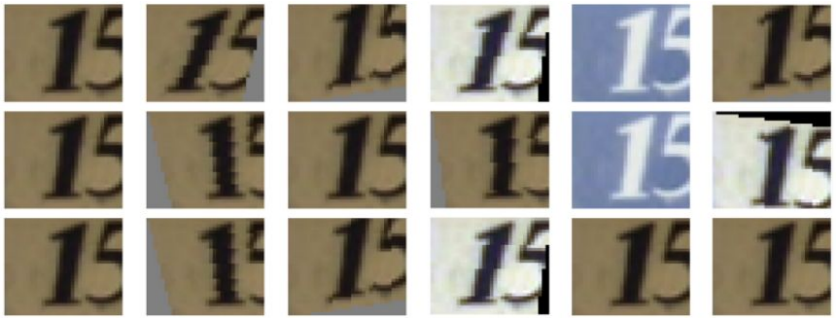


Experiments: CIFAR 10

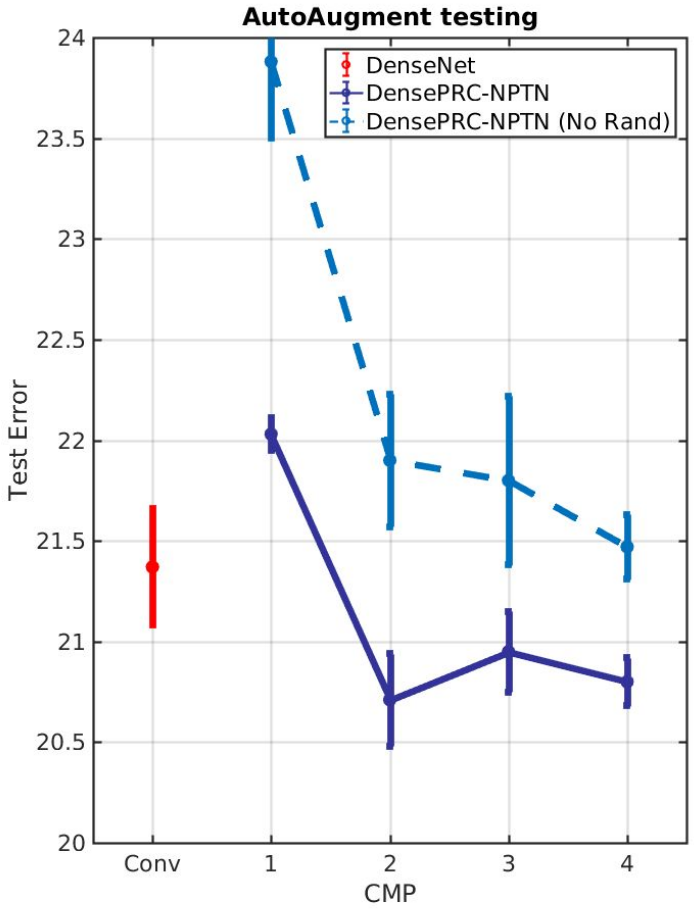
Results: AutoAugment Testing

AutoAugment randomly adds 16 image level degradations.

Added to *train and test* to increase nuisance transformations.



Cubuk, Ekin D., Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. "AutoAugment: Learning augmentation strategies from data." In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 113-123. 2019.

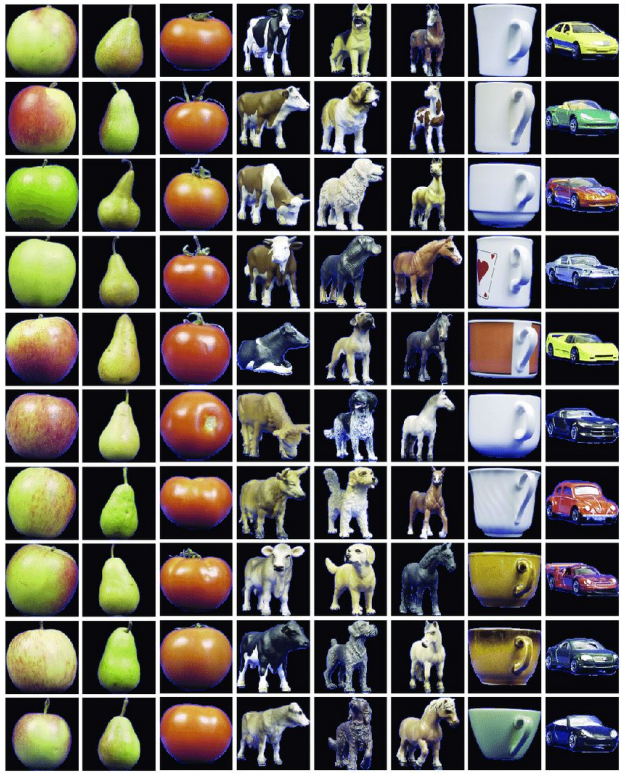


Experiments: ETH-80

Experimental Setup: The Dataset

8 image categories
10 instances of object/category
41 images per instance
2,300 training images, **980** test images
50 x 50 images
Primary nuisance transformation: **3D pose**

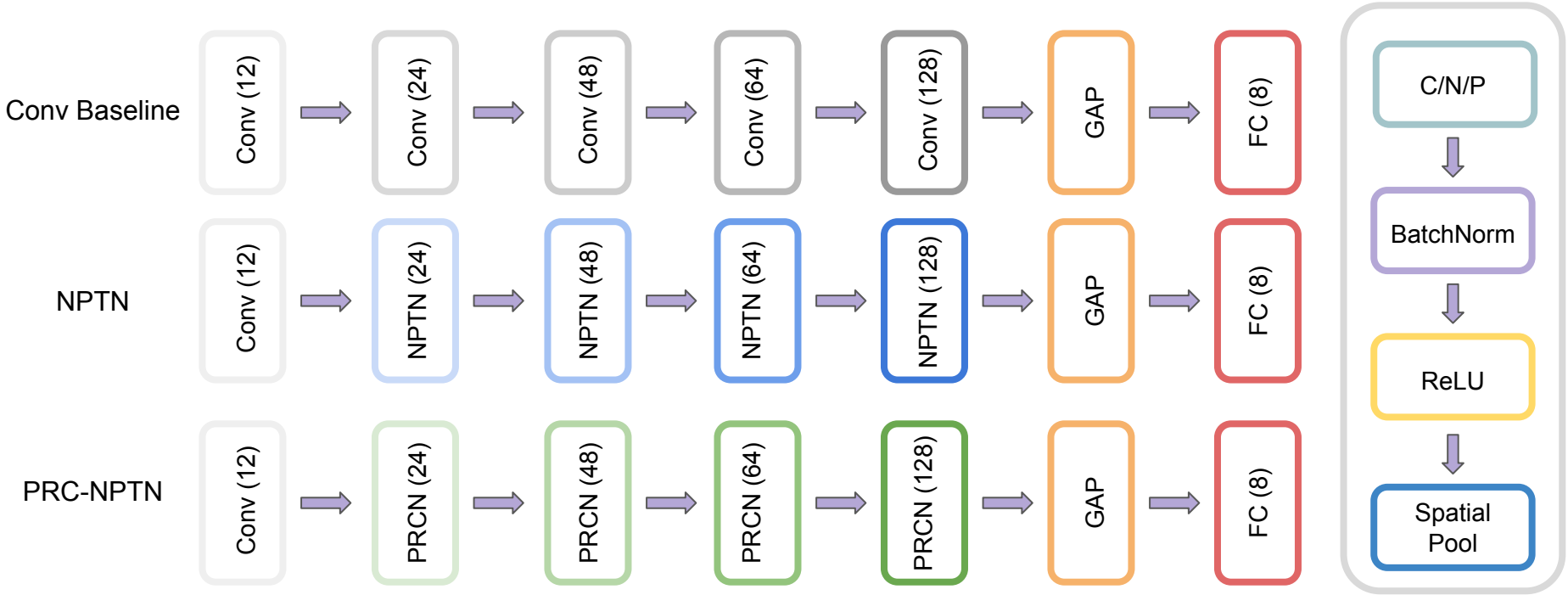
Results averaged over 10 runs with Adam at 0.01 LR for 150 epochs with a LR drop at 100 by a factor of 10.



B. Leibe and, B.Schiele. Analyzing Appearance and Contour Based Methods for Object Categorization. CVPR, vol.2, pp. 409-415, 2003

Experiments: ETH-80

Experimental Setup: The Networks



More modern architectures with a single small FC layer and larger width. Conv 1x1 also tested as baseline.

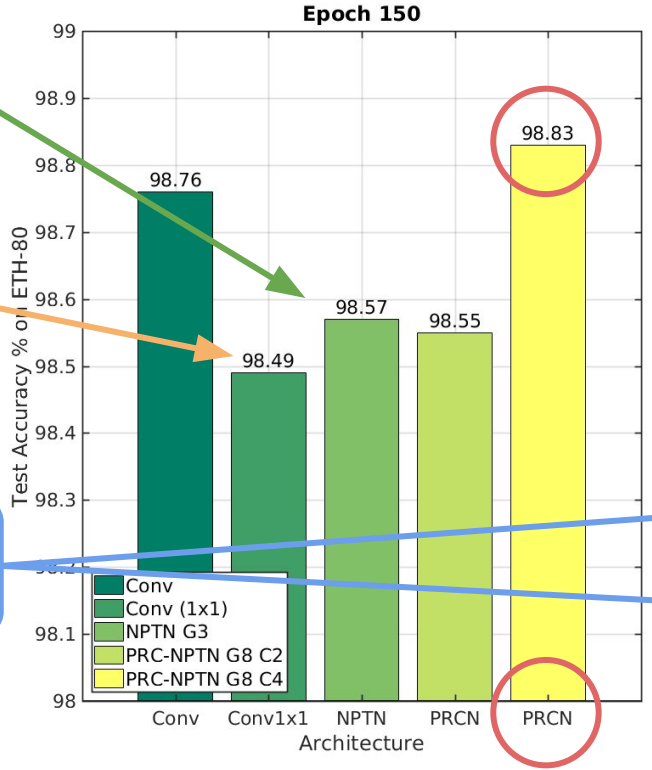
Experiments: ETH-80

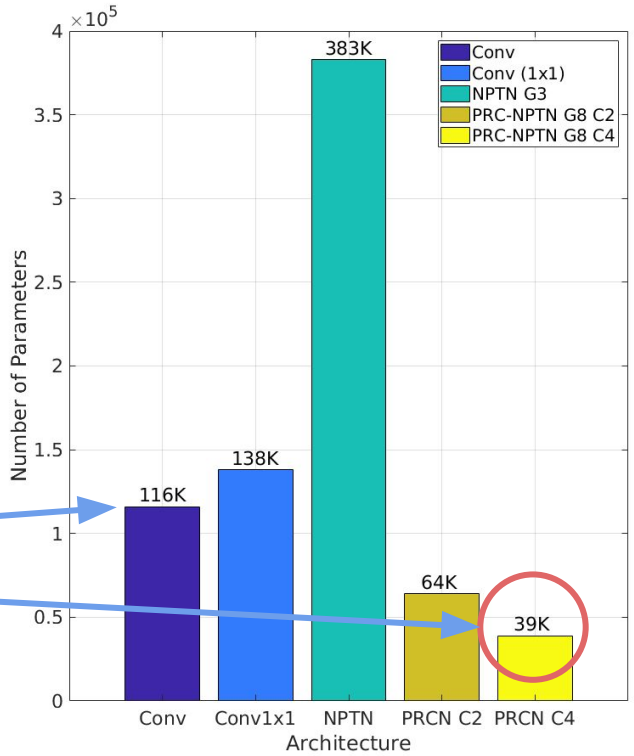
Results: PRC-NPTNs Outperform Baselines with Fewer Parameters

Large NPTN fails to outperform

Naively adding 1x1 conv layers hurt generalization

Almost 3X reduction in parameters





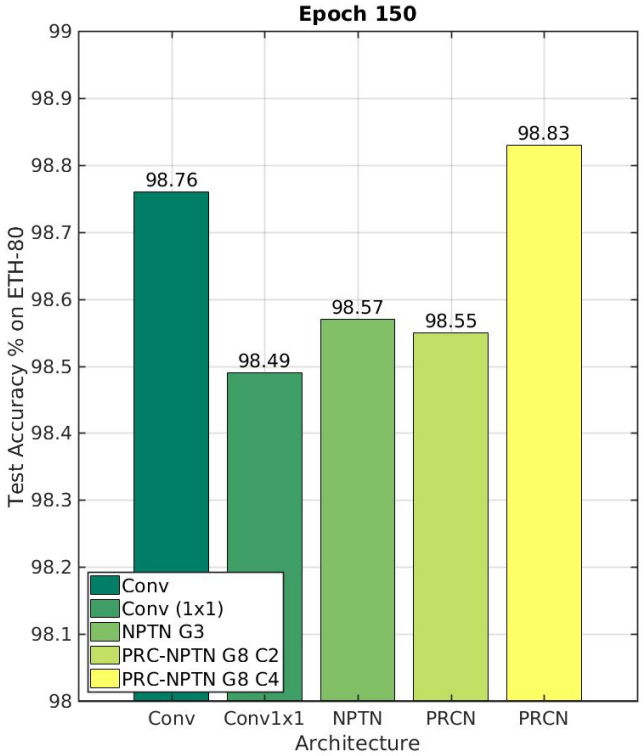
Experiments: ETH-80

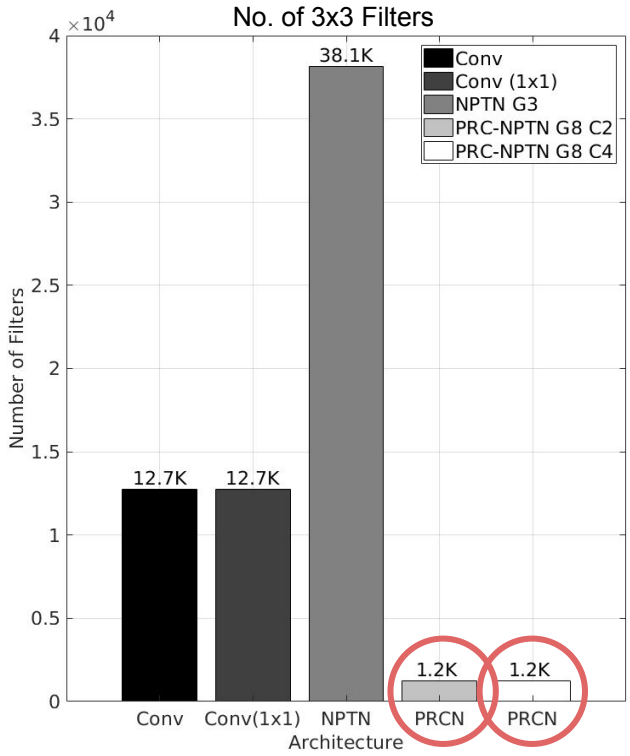
Results: PRC-NPTNs Perform with Far Fewer Filters

10X fewer filters than ConvNets

31X fewer filters than NPTNs

Efficient filter use from scratch, without pruning

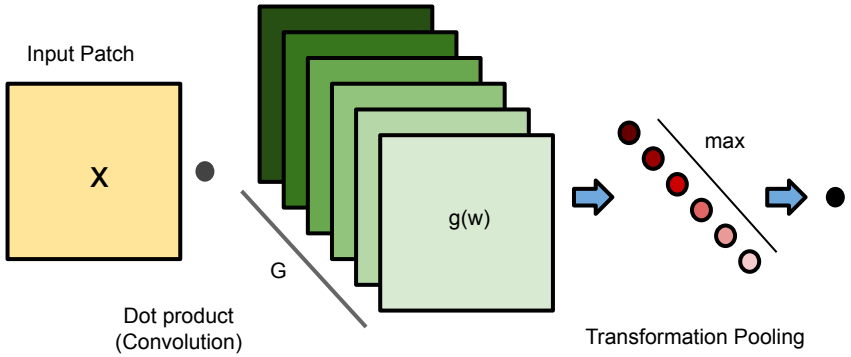




In Summary

Novel Dimensions in Network Architecture Design

We introduced the **Transformation Network framework** to develop network architectures capable of non-parametric invariances.



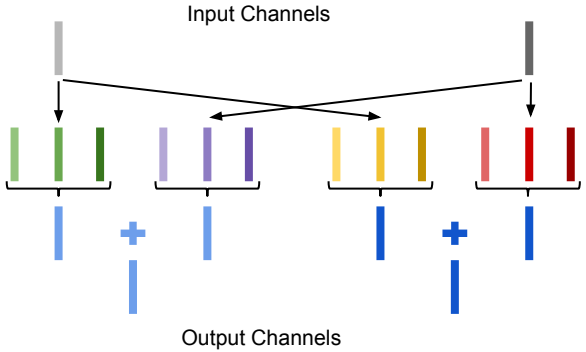
Transformation Network framework

In Summary

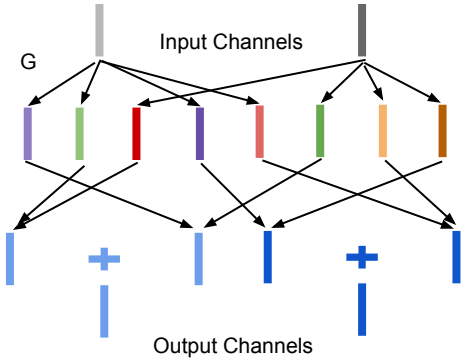
Novel Dimensions in Network Architecture Design

We introduced the **Transformation Network framework** to develop network architectures capable of non-parametric invariances.

Developed **NPTNs** and **PRC-NPTNs** that learn transformation invariance towards seen input from data itself.



NPTN



PRC-NPTN

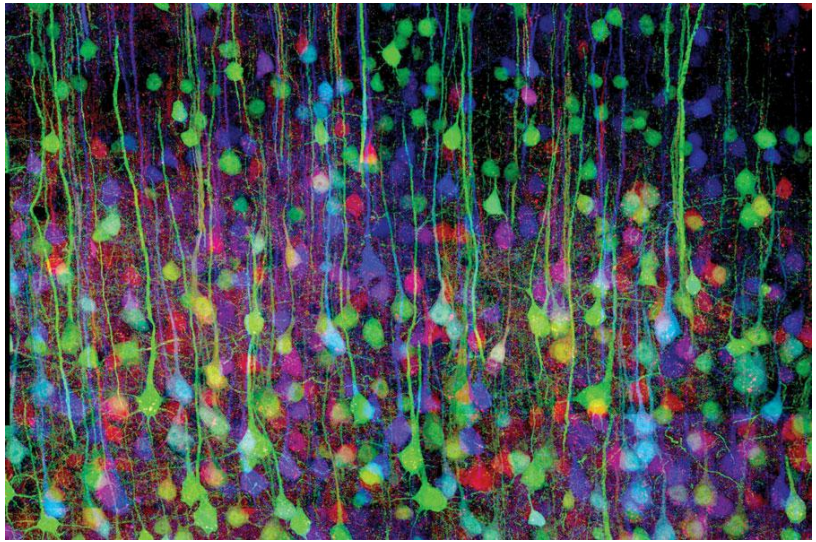
In Summary

Novel Dimensions in Network Architecture Design

We introduced the **Transformation Network framework** to develop network architectures capable of non-parametric invariances.

Developed **NPTNs** and **PRC-NPTNs** that learn transformation invariance towards seen input from data itself.

PRC-NPTNs are loosely **biologically inspired** and are the first architecture to incorporate **completely random and permanent connectomes.**



In Summary

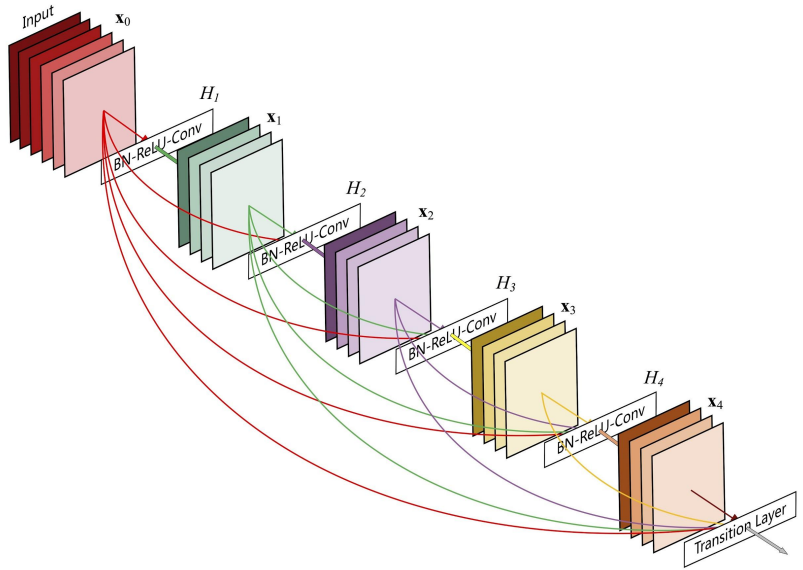
Novel Dimensions in Network Architecture Design

We introduced the **Transformation Network framework** to develop network architectures capable of non-parametric invariances.

Developed **NPTNs** and **PRC-NPTNs** that learn transformation invariance towards seen input from data itself.

PRC-NPTNs are loosely **biologically inspired** and are the first architecture to incorporate **completely random and permanent connectomes.**

Provide **alternate dimensions of improvement** to skip connections.



In Summary

Novel Dimensions in Network Architecture Design

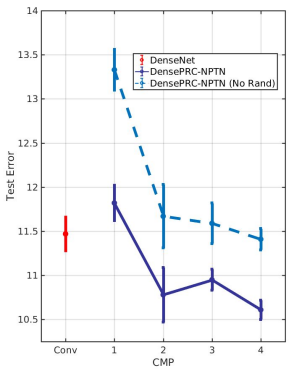
We introduced the **Transformation Network framework** to develop network architectures capable of non-parametric invariances.

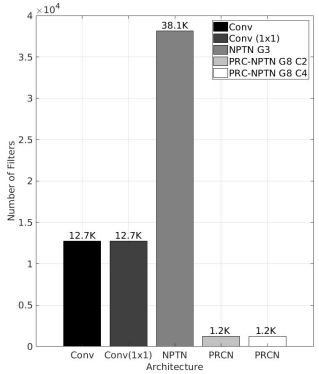
Developed **NPTNs** and **PRC-NPTNs** that learn transformation invariance towards seen input from data itself.

PRC-NPTNs are loosely **biologically inspired** and are the first architecture to incorporate **completely random and permanent connectomes.**

Provide **alternate dimensions of improvement** to skip connections.

PRC-NPTNs **consistently generalize better** than ConvNets **without** the use of skip connections, while **converging faster** and being significantly more **efficient with filter** usage.





Pal, Dipan K., and Marios Savvides. "Learning Non-Parametric Invariances from Data with Permanent Random Connectomes" In *Advances in Neural Information Processing Systems,* workshop on Shared Visual Representations in Human and Machine Intelligence *(**SVRHM**)*, 2019, *British Machine Vision Conference (**BMVC**) 2020*

Questions? and Thank you!

We introduced the **Transformation Network framework** to develop network architectures capable of non-parametric invariances.

Developed **NPTNs** and **PRC-NPTNs** that learn transformation invariance towards seen input from data itself.

PRC-NPTNs are loosely **biologically inspired** and are the first architecture to incorporate **completely random and permanent connectomes.**

Provide **alternate dimensions of improvement** to skip connections.

PRC-NPTNs **consistently generalize better** than ConvNets **without** the use of skip connections, while **converging faster** and being significantly more **efficient with filter** usage.