Stat 212b: Topics in Deep Learning
Lecture 6

Joan Bruna
UC Berkeley

Berkeley
UNIVERSITY OF CALIFORNIA

Review: Separable Scattering Operators

- Local averaging kernel: $x \star \phi_J$
  - locally translation invariant
  - stable to additive and geometric deformations
  - loss of high-frequency information.
- Recover lost information: $\mathcal{U}_J(x) = \{x \star \phi_J \, , \, |x \star \psi_\lambda|\}_{\lambda \in \Lambda_J} \, .$
  - Point-wise, non-expansive non-linearities: maintain stability.
  - Complex modulus maps energy towards low-frequencies.
- Cascade the "recovery" operator:

$$\mathcal{U}_J^2(x) = \{x \star \phi_J \, , \, |x \star \psi_\lambda| \star \phi_J , \, ||x \star \psi_\lambda| \star \psi_{\lambda'}|\}_{\lambda, \lambda' \in \Lambda_J} \, .$$

- Scattering coefficient along a path $p = (\lambda_1, \ldots, \lambda_m) \, :$

$$S_J[p]x(u) = |||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \ldots | \star \psi_{\lambda_m}| \star \phi_J(u) \, .$$

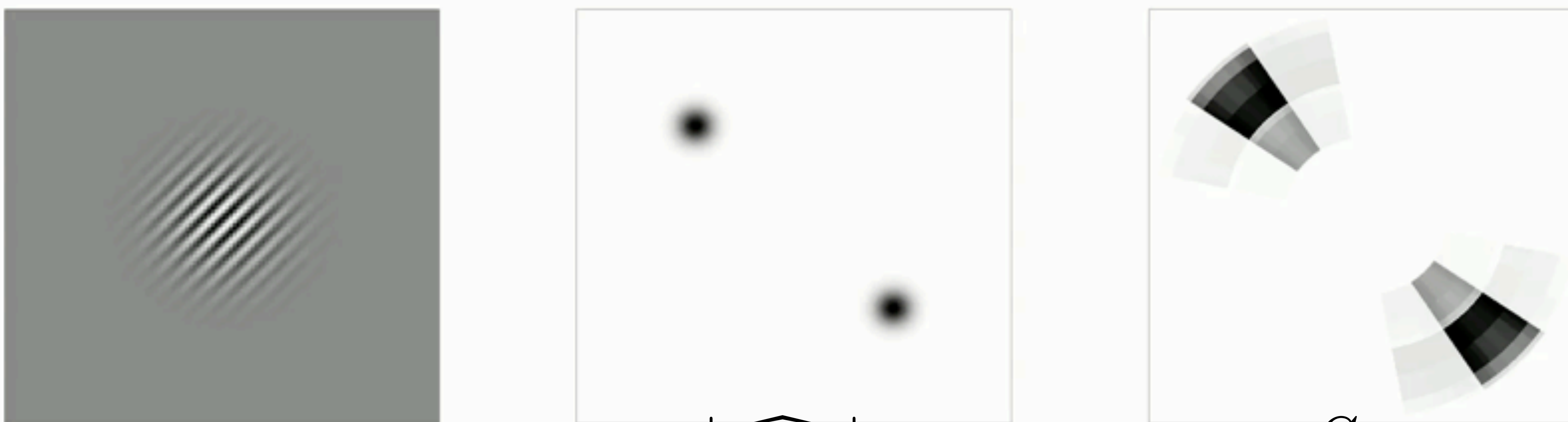
Review: Scattering Geometric Stability

- Geometric Stability:

$$\|S_J x\|^2 = \sum_{p \in \mathcal{P}_J} \|S_J[p]x\|^2$$

Theorem (Mallat'10): There exists C such that for all $x \in L^2(R^d)$ and all m, the m-th order scattering satisfies

$$\|S_J \varphi_\tau x - S_J x\| \leq C m \|x\| (2^{-J}|\tau|_\infty + \|\nabla \tau\|_\infty + \|H\tau\|_\infty) \; .$$



$\varphi_\tau x$ $\quad |\widehat{\varphi_\tau x}|$ $\quad S_J \varphi_\tau x$

Review: Limitations of Separable Scattering

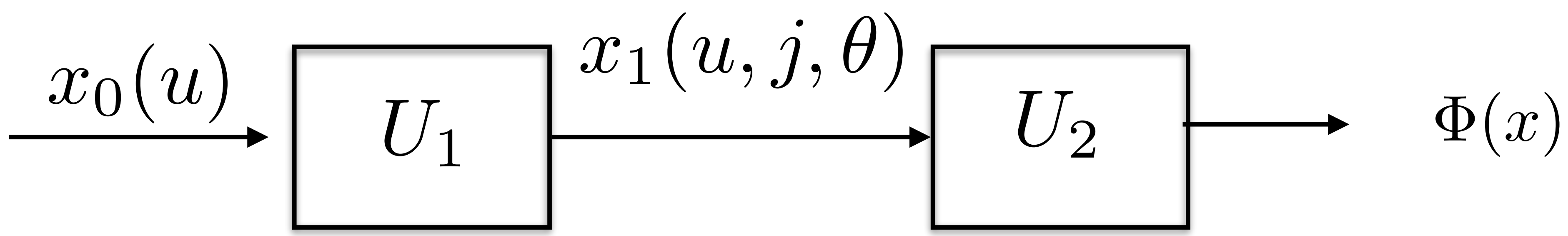
- No feature dimensionality reduction
  - The number of features increases exponentially with depth and polynomially with scale.

- We are indirectly assuming that each wavelet band is deformed independently
  - We cannot capture the *joint* deformation structure of feature maps
  - Loss of discriminability.

- The deformation model is rigid and non-adaptive
  - We cannot adapt to each class
  - Wavelets are hard to define *a priori* on high-dimensional domains.

Review: Joint Scattering

- We start by *lifting* the image with spatial wavelet convolutions: stable and covariant to roto-translations.

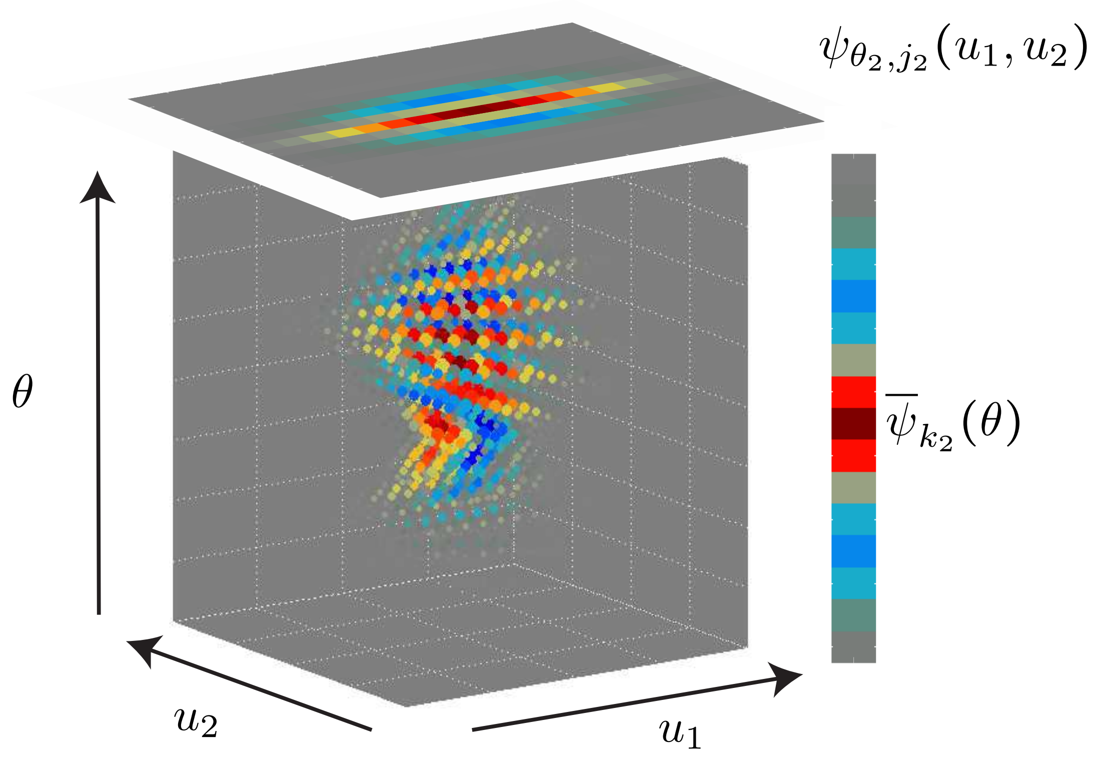


- We then adapt the second wavelet operator to its input joint variability structure.

- More discriminability.
- Requires defining wavelets on more complicated domains

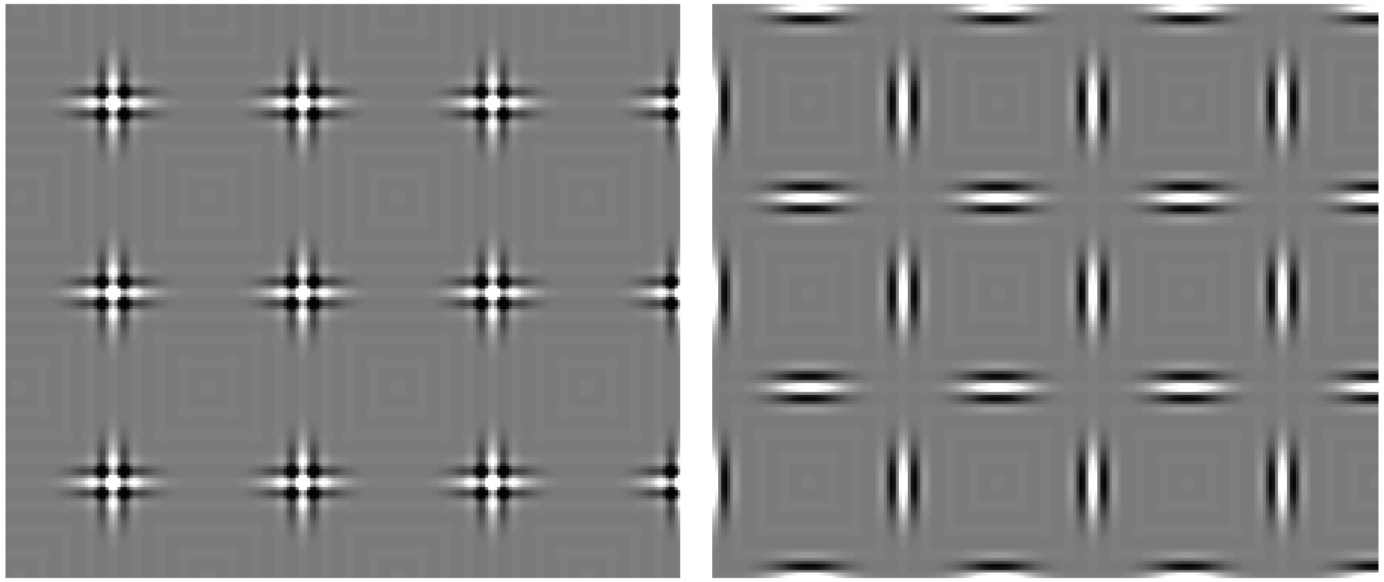
Example: Roto-Translation Scattering

- [Sifre and Mallat'13]



second layer wavelets constructed by a separable product on spatial and rotational wavelets

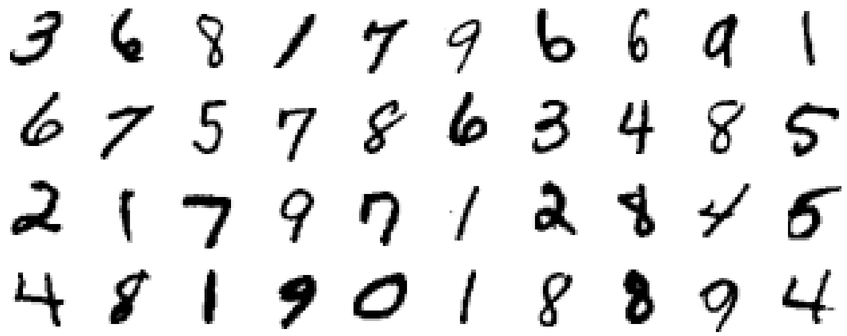
$$\Psi_\lambda(u,\theta) = \psi_{\lambda_1}(u)\psi_{\lambda_2}(\theta)$$



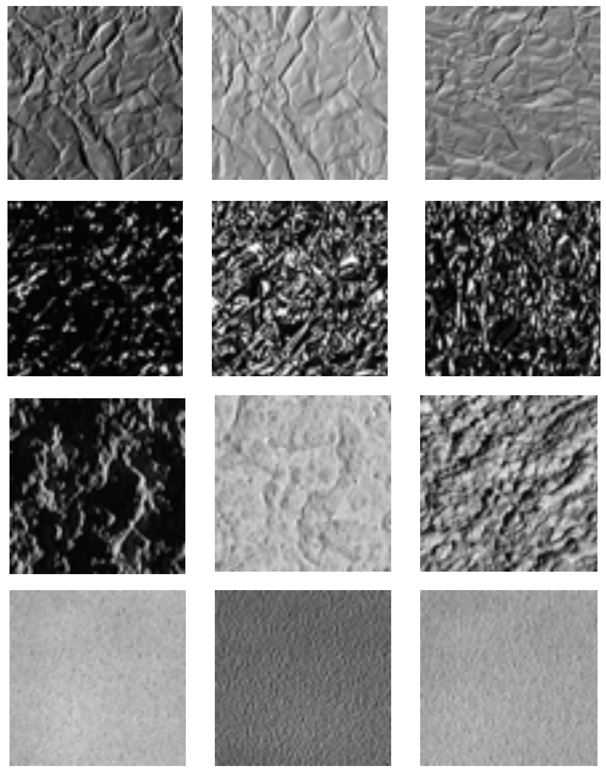
example of patterns that are discriminated by joint scattering but not with separable scattering.

Classification with Scattering

- State-of-the art on pattern and texture recognition using separable scattering followed by SVM:
  - MNIST, USPS [Pami'13]



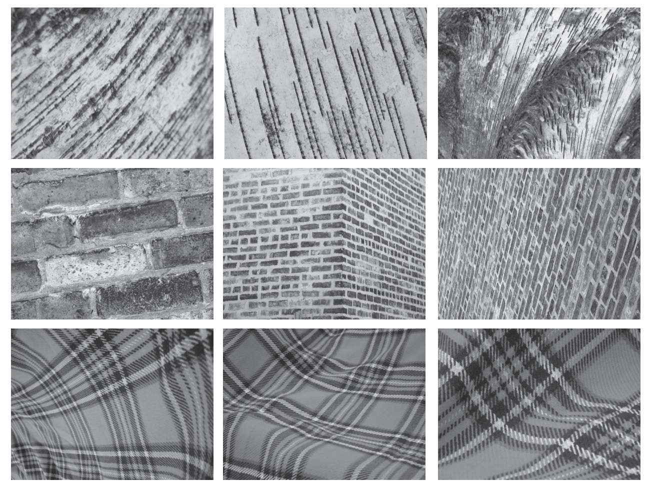
  - Texture (CUREt) [Pami'13]



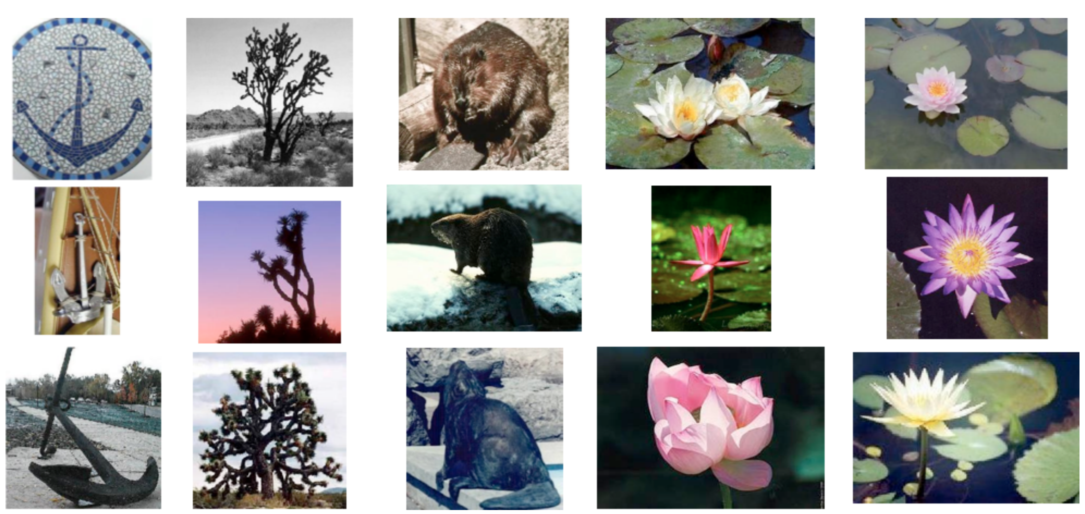
  - Music Genre Classification (GTZAN) [IEEE Acoustic '13]

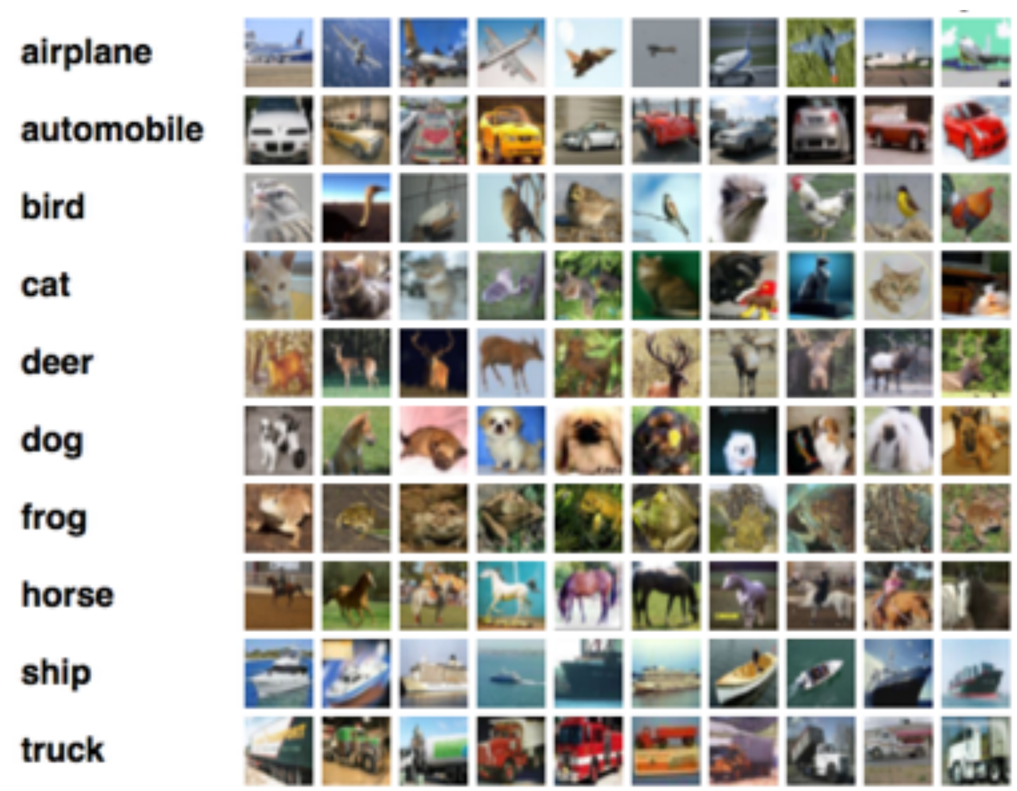
Classification with Scattering

- Joint Scattering Improves Performance:
  - More complicated Texture (KTH,UIUC,UMD) [Sifre&Mallat, CVPR'13]



  - Small-mid scale Object Recognition (Caltech, CIFAR) [Oyallon&Mallat, CVPR'15]
    - ~17% error on Cifar-10





Limitations of Joint Scattering

- Variability from physical world expressed in the language of transformation groups and deformations
  - However, there are not many possible groups: essentially the affine group and its subgroups.

Limitations of Joint Scattering

- Variability from physical world expressed in the language of transformation groups and deformations
  - However, there are not many possible groups: essentially the affine group and its subgroups.

- As a new wavelet layer is introduced, we create new coordinates, but we do not destroy existing coordinates
  - Hard to scale: dimensionality reduction is needed.
  - Wavelet design complicated beyond roto-translation groups.

Limitations of Joint Scattering

- Variability from physical world expressed in the language of transformation groups and deformations
  - However, there are not many possible groups: essentially the affine group and its subgroups.

- As a new wavelet layer is introduced, we create new coordinates, but we do not destroy existing coordinates
  - Hard to scale: dimensionality reduction is needed.
  - Wavelet design complicated beyond roto-translation groups.
- Beyond physics, many deformations are class-specific and not small.
  - Learning filters from data rather than designing them.

Objectives

- Convolutional Neural Networks
  - Review of supervised learning
  - Modular interpretation
  - Streamlining
  - Layer-wise vs Global model.

- Properties of CNN representations
  - Invariance and Covariance
  - Stability and Discriminability
  - Redundancy.
  - Transfer Learning
  - Weakly supervised learning.

From Scattering to CNNs

- Given $x(u, \lambda)$ and a group G acting on both u and λ, we defined wavelet convolutions over G as

$$x \star_G \psi_{\lambda'}(u, \lambda) = \int_v \int_\alpha \psi_\lambda(R_{-\alpha}(u - v))x(v, \alpha)dvd\alpha$$

From Scattering to CNNs

- Given $x(u, \lambda)$ and a group G acting on both u and λ, we defined wavelet convolutions over G as

$$x \star_G \psi_{\lambda'}(u, \lambda) = \int_v \int_\alpha \psi_\lambda(R_{-\alpha}(u - v))x(v, \alpha)dvd\alpha$$

- In discrete coordinates,

$$x \star_G \psi_{\lambda'}(u, \lambda) = \sum_v \sum_\alpha \overline{\psi}_{\lambda'}(u - v, \alpha, \lambda)x(v, \alpha)$$

- Which in general is a convolutional tensor.

Convolutional Neural Networks

- Let $x(u, \lambda)$ be signal, with $u \in \{1, \dots, N\} \times \{1, \dots, N\}$, $\lambda \in \Lambda$.
- Convolutional Tensor:

  Given $\Psi = \{\psi(v, \lambda, \lambda')\}$ with $v \in \{1, N\}^2$, $\lambda \in \Lambda$, $\lambda' \in \Lambda'$, the tensor convolution is

$$x * \Psi(u, \lambda') := \sum_v \sum_\lambda x(u - v, \lambda)\psi(v, \lambda, \lambda')$$

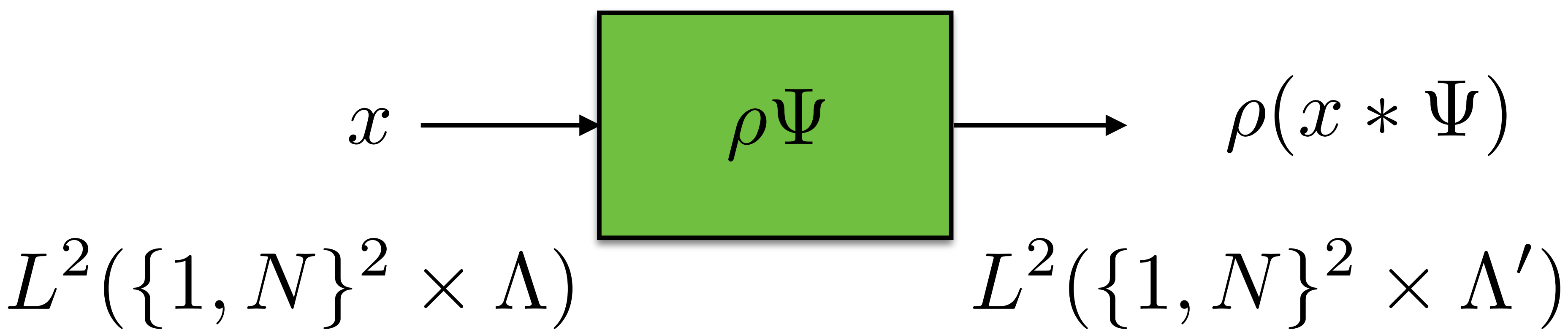
$$= \sum_\lambda (x(\cdot, \lambda) \star \psi(\cdot, \lambda, \lambda'))(u)$$

Convolutional Neural Networks

- Let $x(u,\lambda)$ be signal, with $u \in \{1,\dots,N\} \times \{1,\dots,N\}$, $\lambda \in \Lambda$.
- Convolutional Tensor:

  Given $\Psi = \{\psi(v,\lambda,\lambda')\}$ with $v \in \{1,N\}^2$,
  $\lambda \in \Lambda$, $\lambda' \in \Lambda'$, the tensor convolution is

$$x * \Psi(u,\lambda') := \sum_v \sum_\lambda x(u-v,\lambda)\psi(v,\lambda,\lambda')$$
$$= \sum_\lambda (x(\cdot,\lambda) \star \psi(\cdot,\lambda,\lambda'))(u)$$



(ρ point-wise non-linearity)

Convolutional Neural Networks

- Downsampling or Pooling operator:
  reduce spatial and/or feature resolution

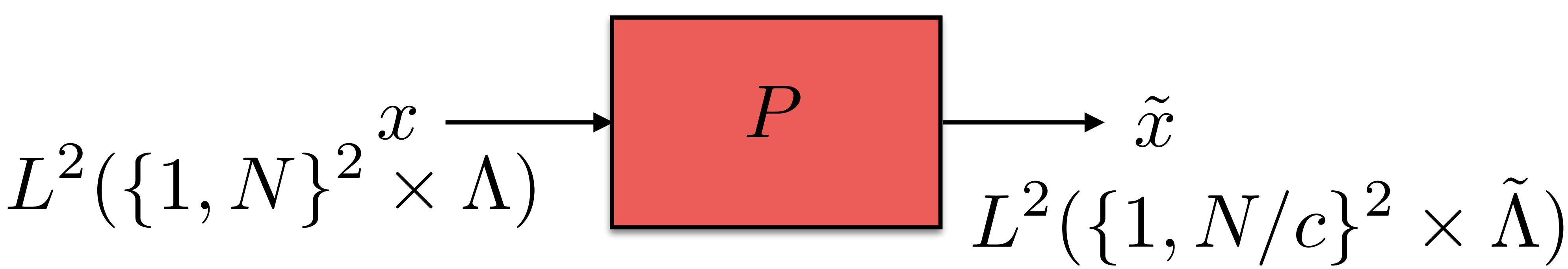
Convolutional Neural Networks

- Downsampling or Pooling operator:
  reduce spatial and/or feature resolution

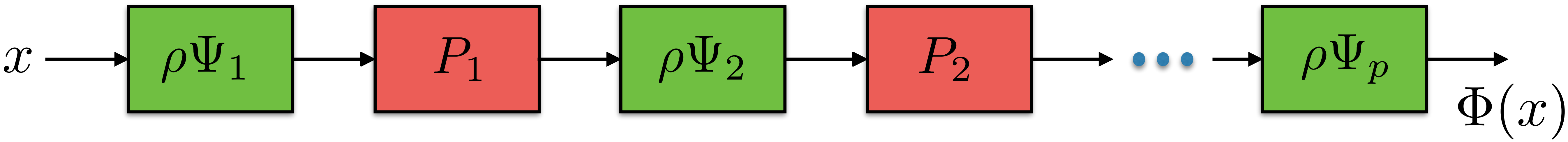
  - Non-adaptive and linear: ϕ_c: lowpass averaging kernel
    $\tilde{x}(\tilde{u}, \tilde{\lambda}) = \sum_v \sum_\lambda \phi_c(v, \lambda) x(c\tilde{u} - v, c_\lambda \tilde{\lambda} - \lambda)$

  - Non-adaptive and non-linear:
    $\tilde{x}(\tilde{u}, \tilde{\lambda}) = \max_{|v| \leq c, |\lambda| \leq c} x(c\tilde{u} - v, c\tilde{\lambda} - \lambda)$

  Adaptive and linear:
  $\tilde{x}(\tilde{u}, \tilde{\lambda}) = x * \Psi(c\tilde{u}, c\tilde{\lambda})$

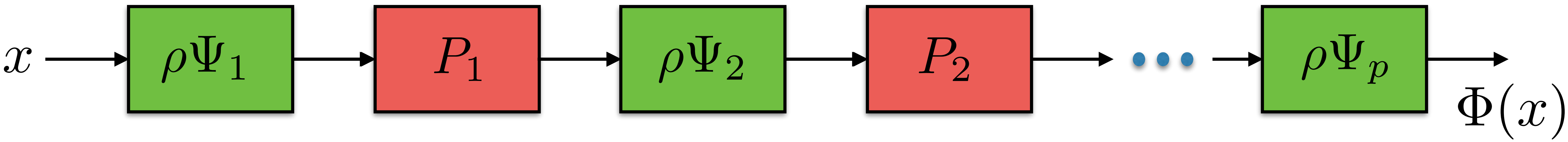


Convolutional Neural Networks



$$\Phi(x) = \rho(\rho(P_1(\rho(x * \Psi_1)) * \Psi_2)..)$$

Convolutional Neural Networks



$$\Phi(x) = \rho(\rho(P_1(\rho(x * \Psi_1)) * \Psi_2)..)$$

- Architectures vary in terms of
  - Number p of layers (from 2 to >100).
  - Size of the tensors (typically [3-7 x 3-7 x 16-256])
  - Presence/absence and type of pooling operator.
    - Recent models tend to avoid non-adaptive pooling.

CNNs for Classification

- When task is classification, $\Phi(x)$ estimates the class label of x , $y \in \{1, K\}$
- The conditional probability $p(y \mid x)$ is modeled with a multinomial distribution with parameters $\pi_k(\Phi(x))$, $k \leq K$.

CNNs for Classification

- When task is classification, $\Phi(x)$ estimates the class label of x , $y \in \{1, K\}$
- The conditional probability $p(y \mid x)$ is modeled with a multinomial distribution with parameters $\pi_k(\Phi(x))$, $k \leq K$.
- If the last layer has K feature maps, we parametrize using the *softmax* distribution:

$$p(y = k \mid x) = \frac{e^{\overline{\Phi_k(x)}}}{\sum_{j \leq K} e^{\overline{\Phi_j(x)}}} ,$$

$\overline{\Phi_j(x)}$: spatial average of output channel j

CNN for Classification

- We optimize the parameters of the model via Maximum Likelihood (multinomial logistic regression):

Given iid training data $(x_i, y_i)_i$, the negative joint log-likelihood is

$$\mathcal{E}(\Psi) = \sum_i \log p(y = y_i | x_i) = \sum_i \left(\overline{\Phi_{y_i}(x_i)} - \log \left(\sum_j e^{\overline{\Phi_j(x_i)}} \right) \right)$$

CNN for Classification

- We optimize the parameters of the model via Maximum Likelihood (multinomial logistic regression):

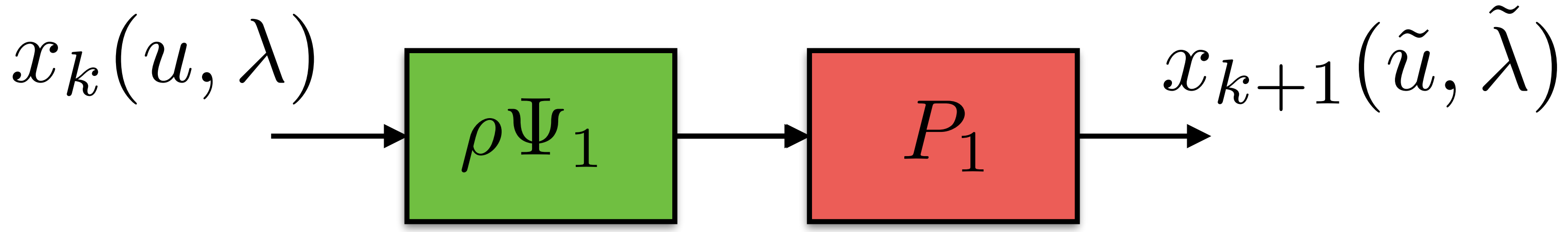
Given iid training data $(x_i, y_i)_i$, the negative joint log-likelihood is

$$\mathcal{E}(\Psi) = \sum_i \log p(y = y_i | x_i) = \sum_i \left(\overline{\Phi_{y_i}(x_i)} - \log \left(\sum_j e^{\overline{\Phi_j(x_i)}} \right) \right)$$

- Other parametrizations of the multinomial are possible
  - See for example http://arxiv.org/abs/1506.08230 , where a contrast-invariant loss replaces multinomial logistic regression.

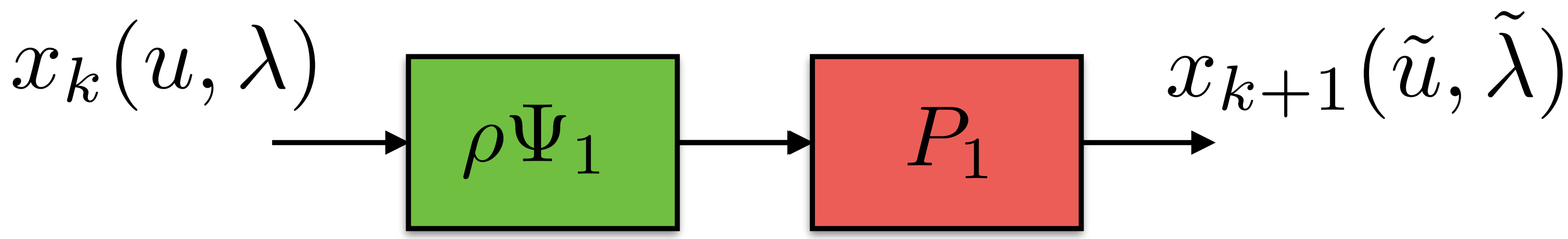
Geometric Interpretations

- We can start by analyzing a chunk of the form



Geometric Interpretations

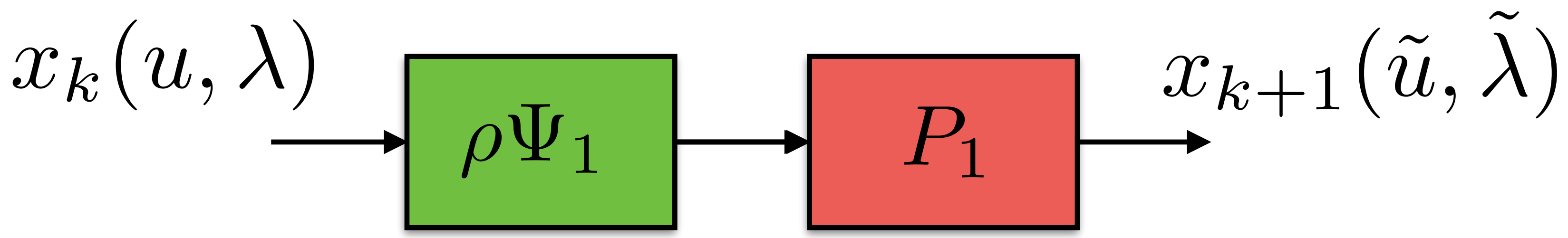
- We can start by analyzing a chunk of the form



- Let us assume that pooling is an average (non-adaptive).
- Consider a thresholding nonlinearity: $\rho(x) = \max(0, x - t)$
- And let us forget (for now) about the convolutional aspect.

Geometric Interpretations

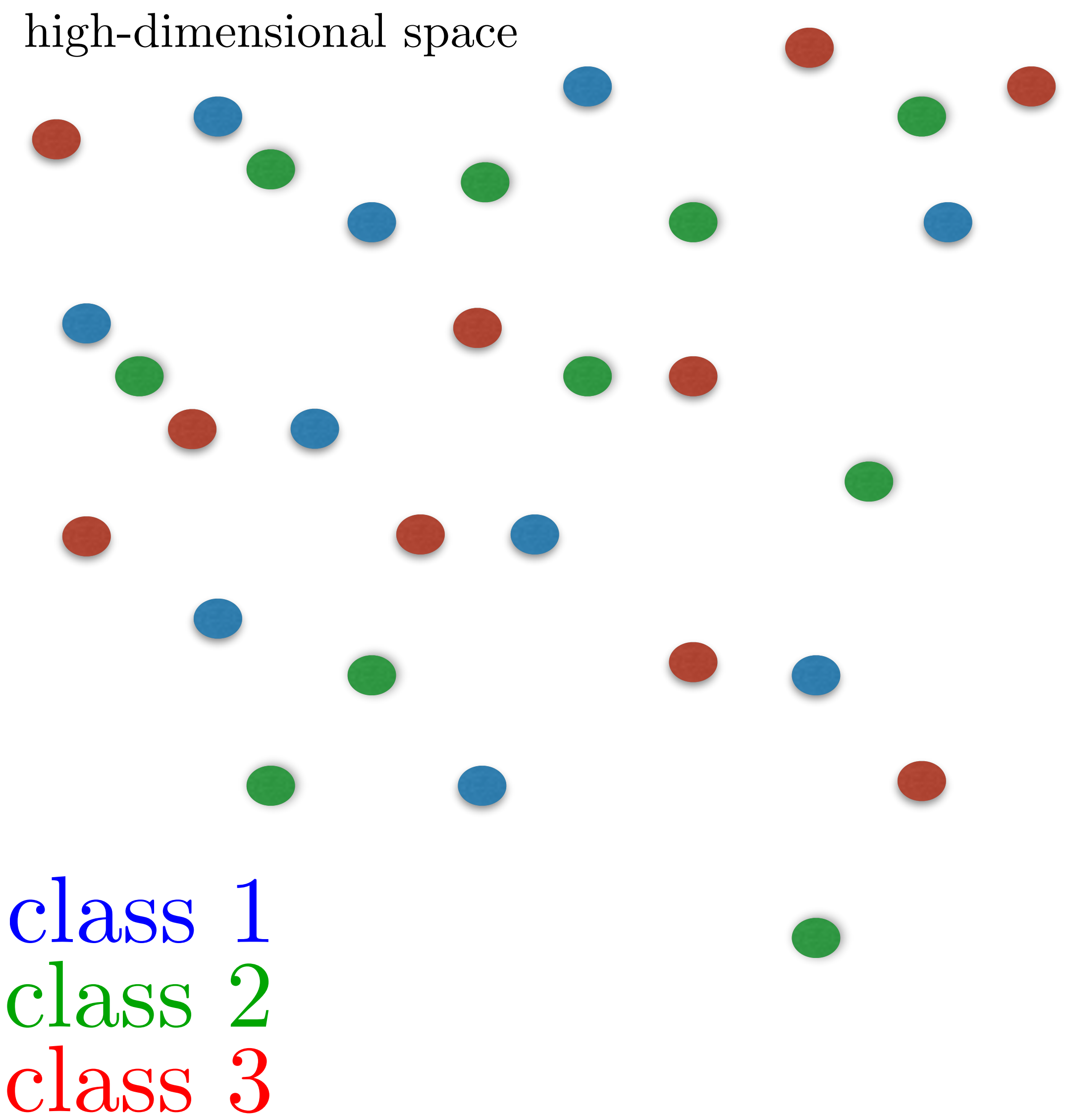
- We can start by analyzing a chunk of the form



- Let us assume that pooling is an average (non-adaptive).
- Consider a thresholding nonlinearity: $\rho(x) = \max(0, x - t)$
- And let us forget (for now) about the convolutional aspect.

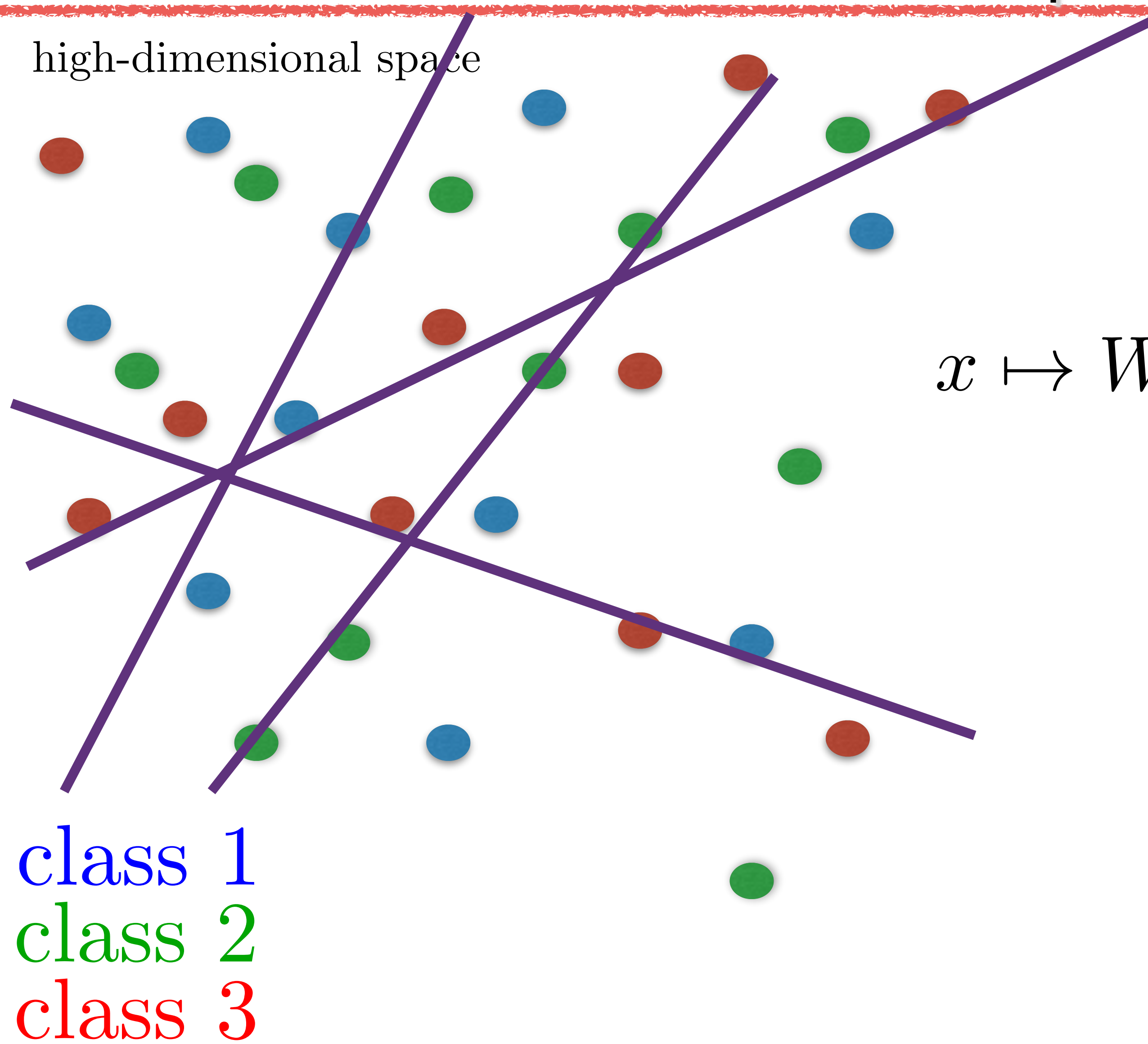
- What is the role of this operator? Intuition?

Geometric Interpretations



- Intraclass variability is highly nonlinear.
- But we are attempting to progressively linearize it by cascading instances of the previous operator.

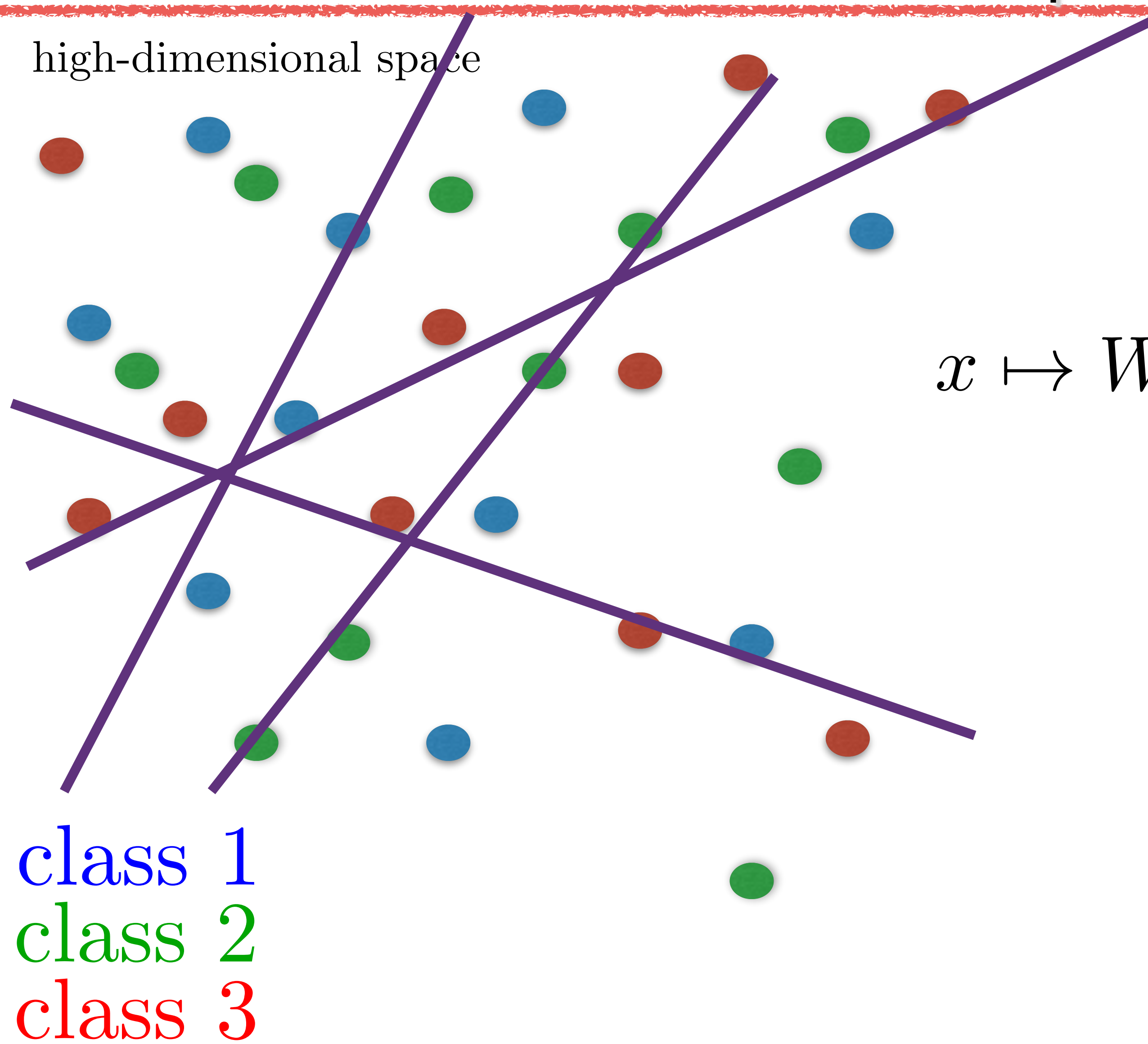
Geometric Interpretations



$x \mapsto W^T x$ with a redundant (fat) matrix

- 1: "trap" intraclass variability within low-dimensional affine subspaces appropriately chosen.

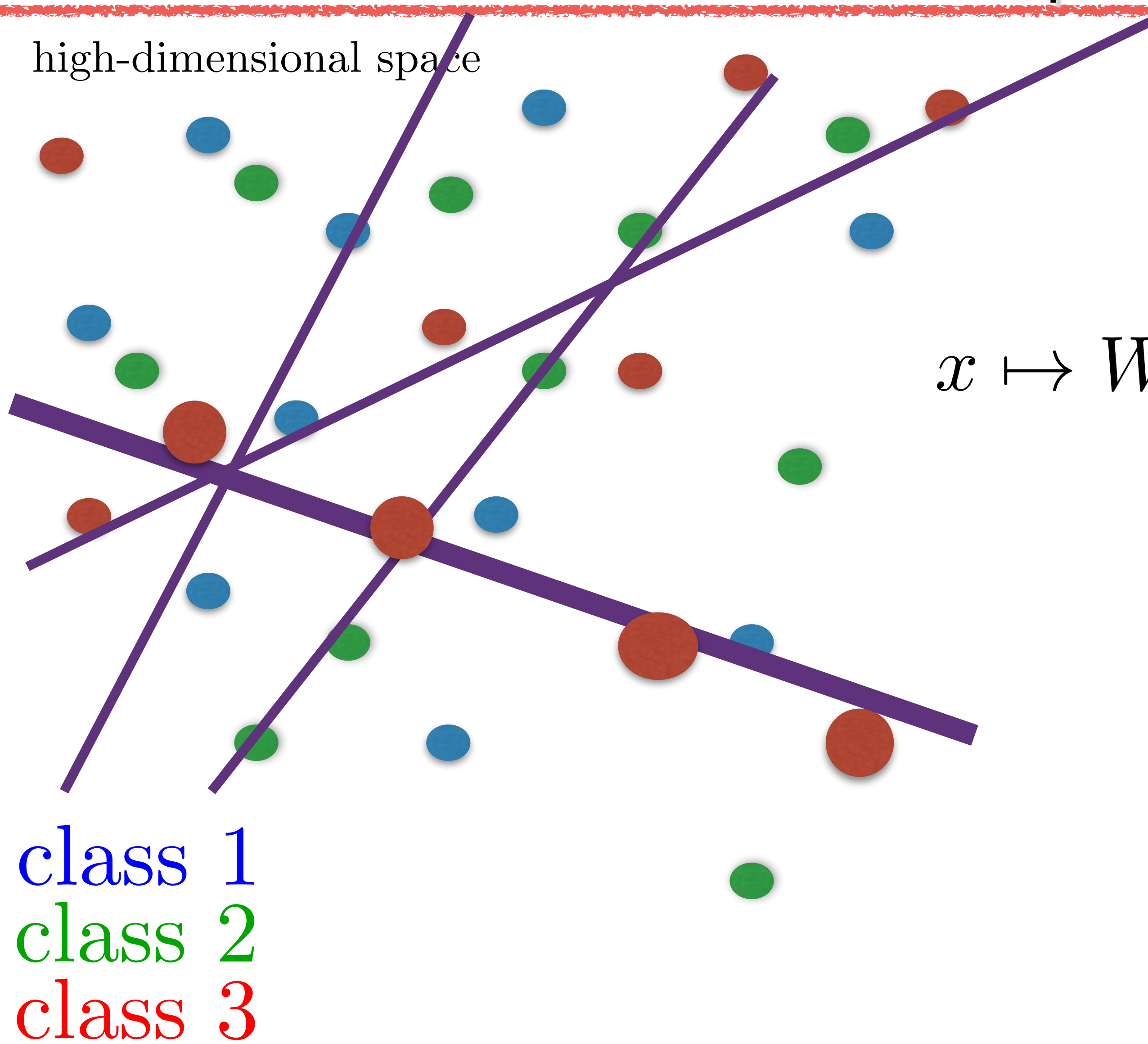
Geometric Interpretations



$x \mapsto W^T x$ with a redundant (fat) matrix

- 1: "trap" intraclass variability within low-dimensional affine subspaces appropriately chosen.
  - In this example we are not sharing models, but later we will see that *parallel* models are key for generalization.

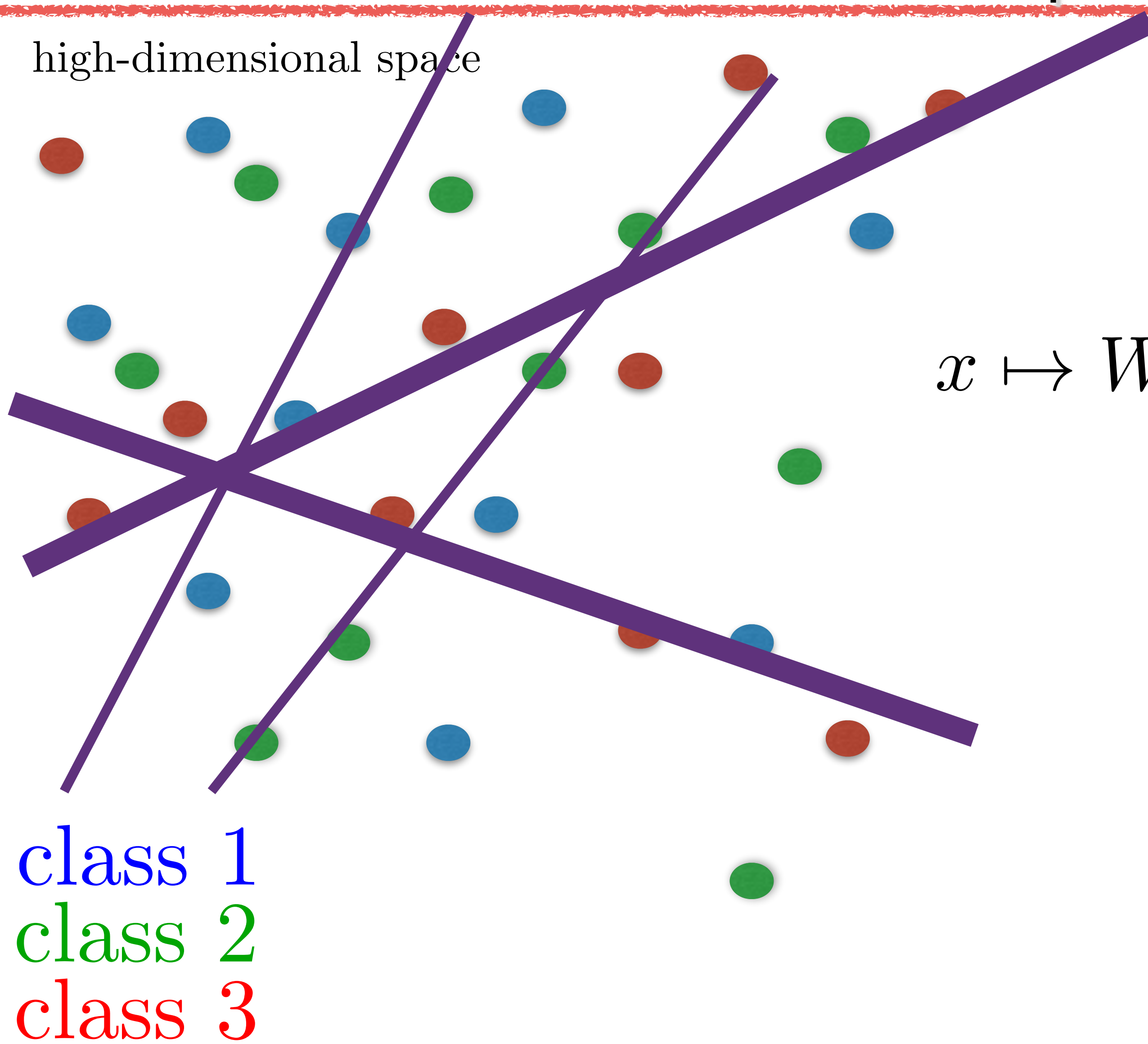
Geometric Interpretations



$x \mapsto W^T x$ with a redundant (fat) matrix

- 2. detect distance to each affine model with a thresholding
  - Thresholding operates along 1-dimensional subspaces (complex modulus instead looks at 2-dimensional)

Geometric Interpretations



$x \mapsto W^T x$ with a redundant (fat) matrix

- 3: "stitch" different linear pieces together by pooling the output of the two subspace detectors.
  - Can be done by smoothing or by computing any statistic (max-pooling)

Geometric Interpretation

- But in high-dimensional image recognition, this operator alone is not sufficient: there are exponentially many linear pieces required: curse of dimensionality.

Geometric Interpretation

- But in high-dimensional image recognition, this operator alone is not sufficient: there are exponentially many linear pieces required: curse of dimensionality.

- Intra-class variability model (i.e. deformation model):

$$f\left(\{\varphi_{\tau, f(x)} x\}\right) \approx f(x)$$

  - Besides small geometric deformations, we must include clutter and large class-specific variability (for example, chair styles).
  - It is a high-dimensional variability model

Geometric Interpretation

- Adjoint deformation operator:

The adjoint φ^* of a linear operator φ is such that

$$\forall\ x, w\ ,\ \langle \varphi x, w \rangle = \langle x, \varphi^* w \rangle$$

(in finite dimension, it is just the transpose of a matrix)

$$\left(\langle Ax, w \rangle = w^T (Ax) = x^T (A^T w) = \langle x, A^T w \rangle\right)$$

Geometric Interpretation

- Adjoint deformation operator:

The adjoint φ^* of a linear operator φ is such that

$$\forall\ x, w\ ,\ \langle \varphi x, w \rangle = \langle x, \varphi^* w \rangle$$

(in finite dimension, it is just the transpose of a matrix)

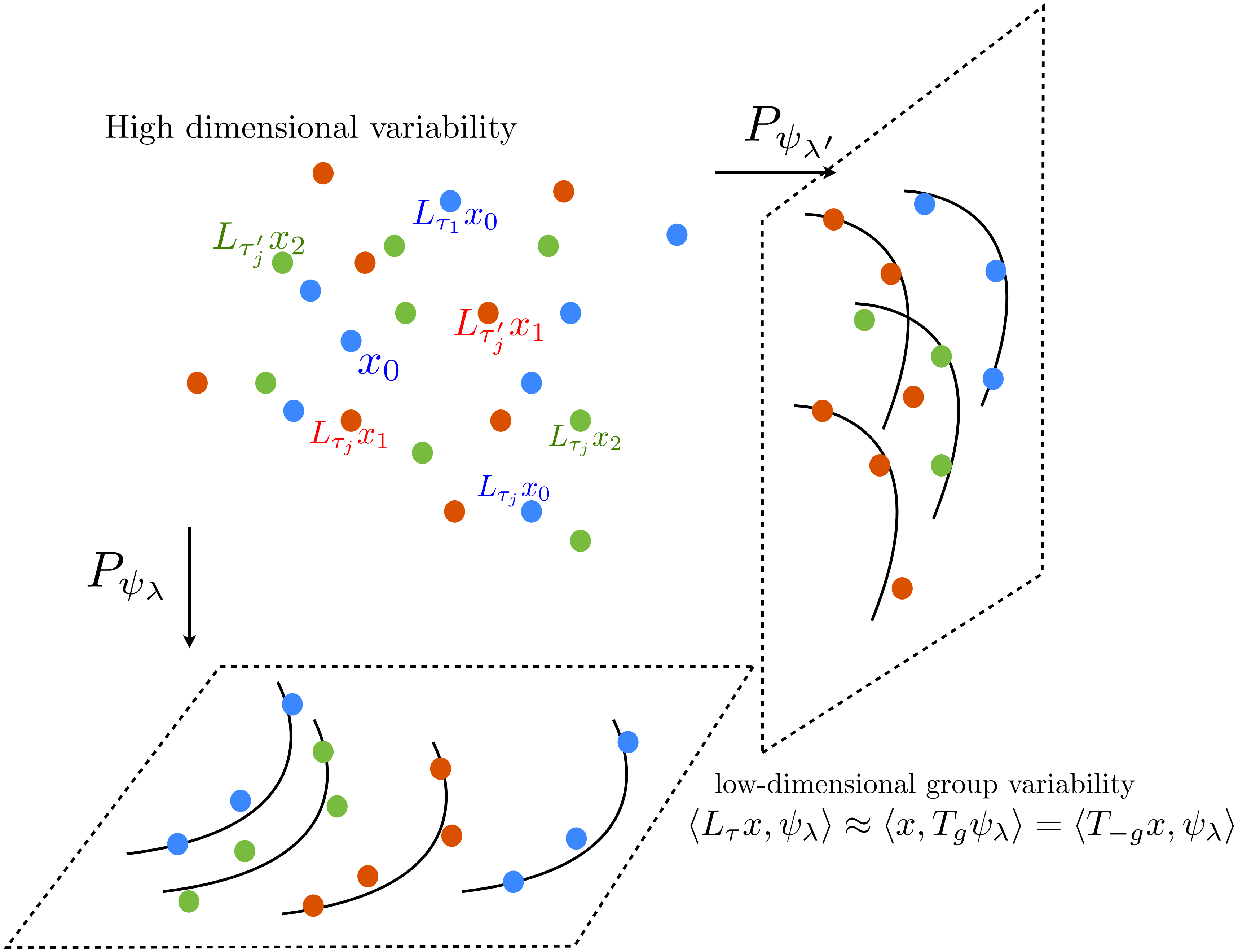
$$(\langle Ax, w \rangle = w^T(Ax) = x^T(A^T w) = \langle x, A^T w \rangle)$$

- Our linear measurements W interact with deformations as $\langle \varphi_\tau x, w_k \rangle = \langle x, \varphi_\tau^* w_k \rangle$
  - We want measurements that factorize variability.
  - If w_k are localized, they factorize deformations in local neighborhoods: each measure "sees" approximately a translation

$$\langle x, \varphi_\tau^* w_k \rangle = \langle x, T_v w_k \rangle + \epsilon$$

T_v: translation

Geometric Interpretation



Geometric Interpretation

- The measurements are shared for every input:
  - Factors need to be useful across different inputs.
  - At the same time, measurements need to capture joint dependencies in order to preserve discriminability.

- However, large variability might be class-specific, object-specific:
  - We will see that thresholding and sparsity inducing filters create specialized invariants.

Streamlining CNNs

- Previous CNN models also contained *local contrast normalization* layers:

$$\tilde{x}(u,\lambda) = \frac{x(u,\lambda)}{S(u,\lambda)} \ , \ S(u,\lambda) = \epsilon + \left(\sum_{|v|\leq C, |\lambda'|\leq C'|} |x(u+v,\lambda+\lambda')|^q \right)^{1/q}$$

Streamlining CNNs

- Previous CNN models also contained *local contrast normalization* layers:

$$\tilde{x}(u,\lambda) = \frac{x(u,\lambda)}{S(u,\lambda)} \;,\; S(u,\lambda) = \epsilon + \left(\sum_{|v|\leq C, |\lambda'|\leq C'|} |x(u+v, \lambda+\lambda')|^q \right)^{1/q}$$

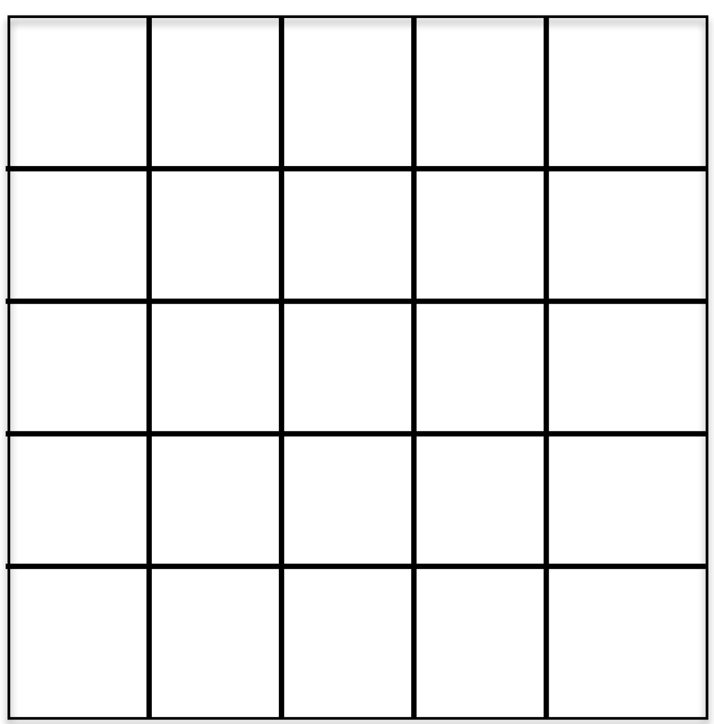
- Provides invariance to amplitude changes.
- Can improve gradient flow towards initial layers.
- However, modern CNNs do not use it: contrast invariance is low-dimensional, it can be learnt by the classifier
- And there are other optimization improvements that attenuate the "vanishing gradient" problem.

Streamlining CNNs

- An important parameter is the spatial kernel size: how to choose it?

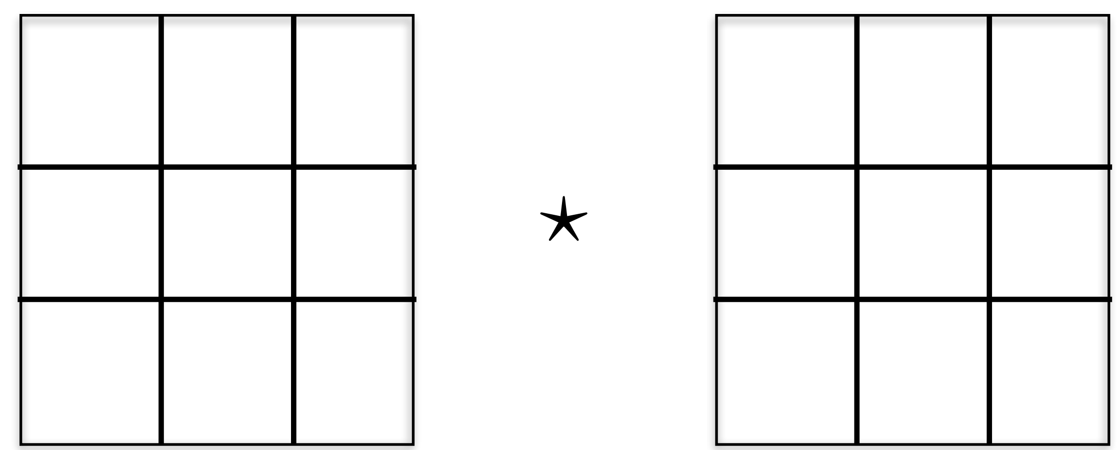
Streamlining CNNs

- An important parameter is the spatial kernel size: how to choose it?
- Previous CNNs explored the parameter space: typically kernel sizes < 10.



w of size $2L+1$

$\sim (2L+1)^2$ parameters



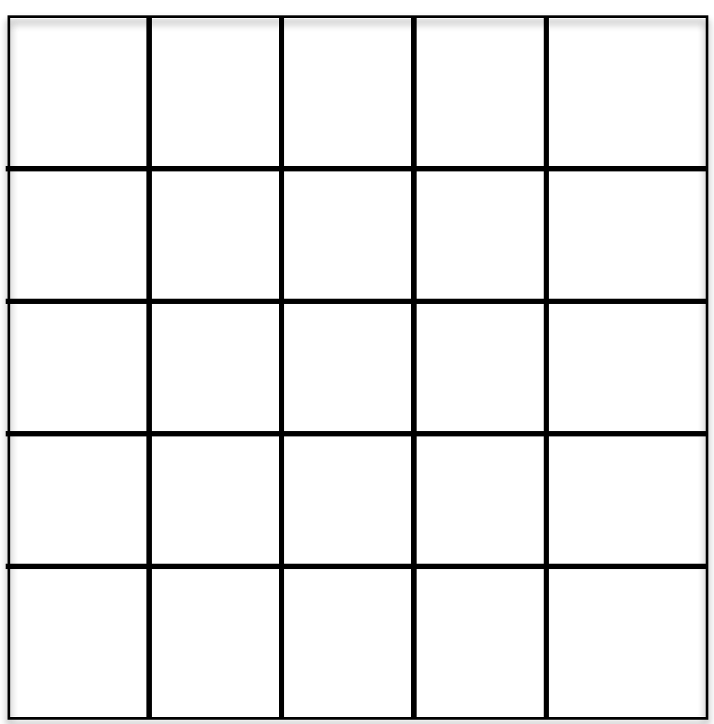
$h_1,\ h_2$ of size $L+1$ each

Then $h_1 \star h_2$ is of size $2L+1$

$\sim 2(L+1)^2$ parameters

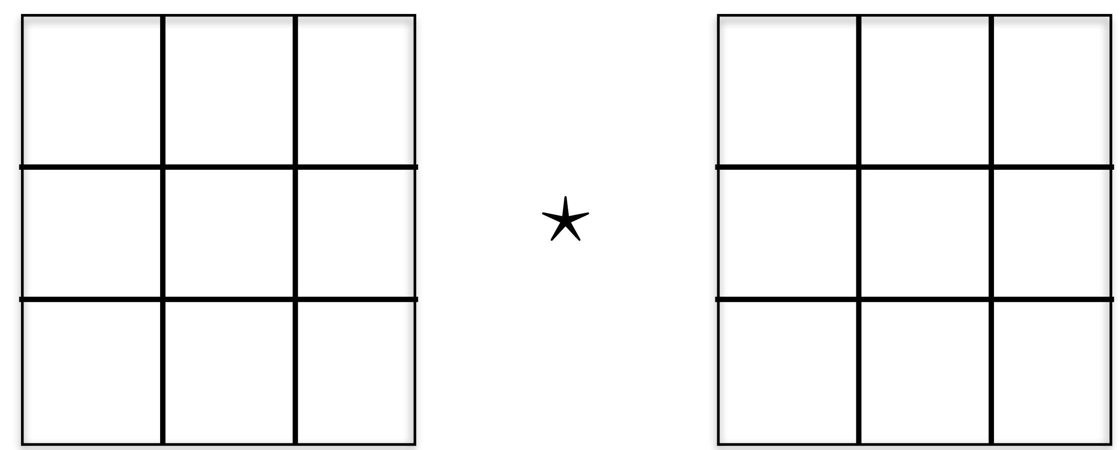
Streamlining CNNs

- Modern CNNs prefer to replace larger spatial kernels by a cascade of small (3x3, or even 1x3, 3x1) kernels.
- It sacrifices frequency resolution in favor of smaller parameter size.



w of size $2L+1$

$\sim (2L+1)^2$ parameters



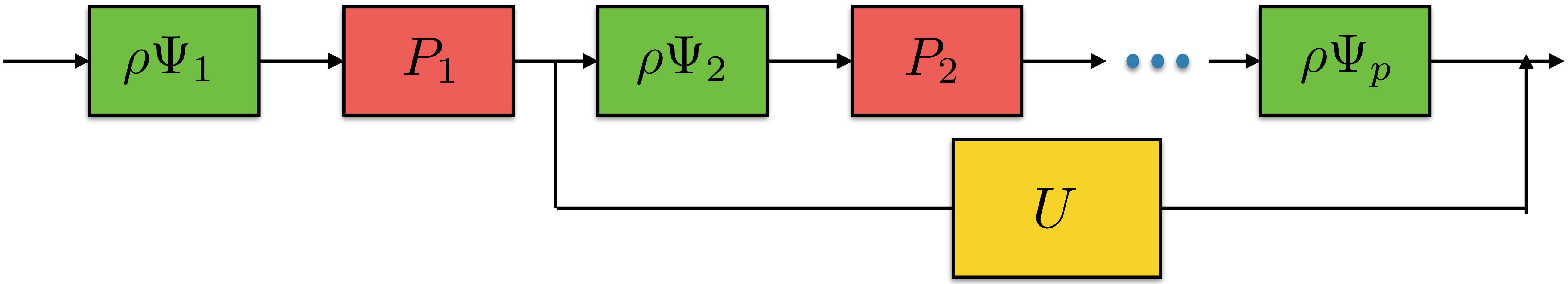
$h_1,\ h_2$ of size $L+1$ each

Then $h_1 \star h_2$ is of size $2L+1$

$\sim 2(L+1)^2$ parameters

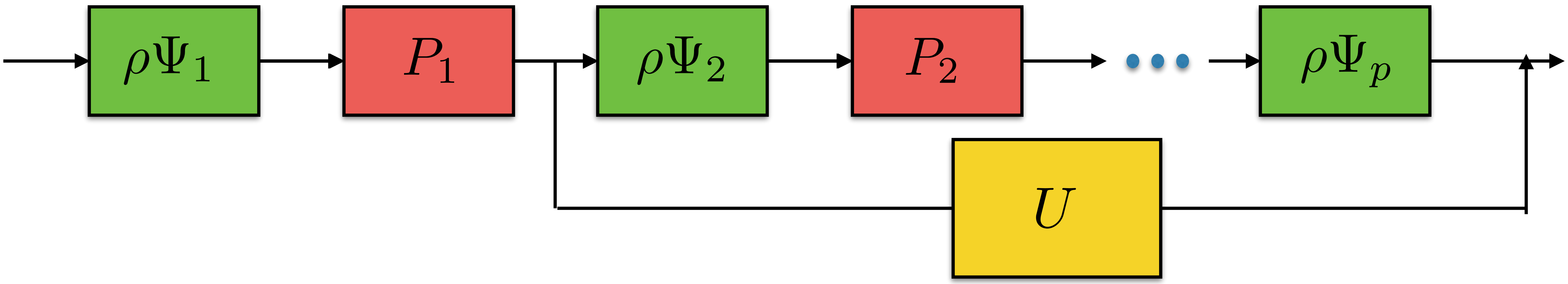
Streamlining CNNs

- Another recent trend is to use *"skip-connections"*:



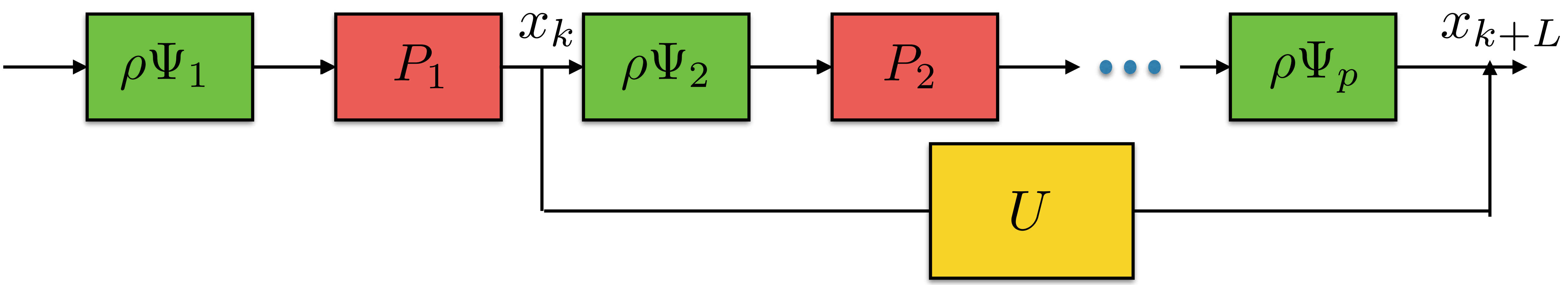
Streamlining CNNs

- Another recent trend is to use *"skip-connections"*:



- The operator U is as simple as a linear projection or even the identity (if there are no downsampling layers in between)
  - Deep Residual Learning (K. He et al '15)
  - Highway Networks (Srivastava et al '15) use slightly more complicated U modules with "gating".

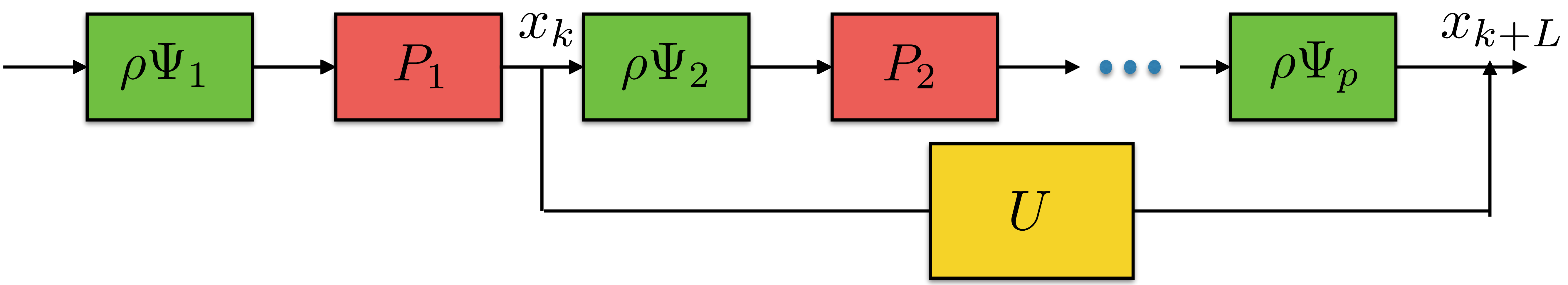
Streamlining CNNs



$$x_{k+L} = x_k + \Phi_k(x_k)$$

- Each subnetwork Φ_k is thus learning a *residual* representation

Streamlining CNNs

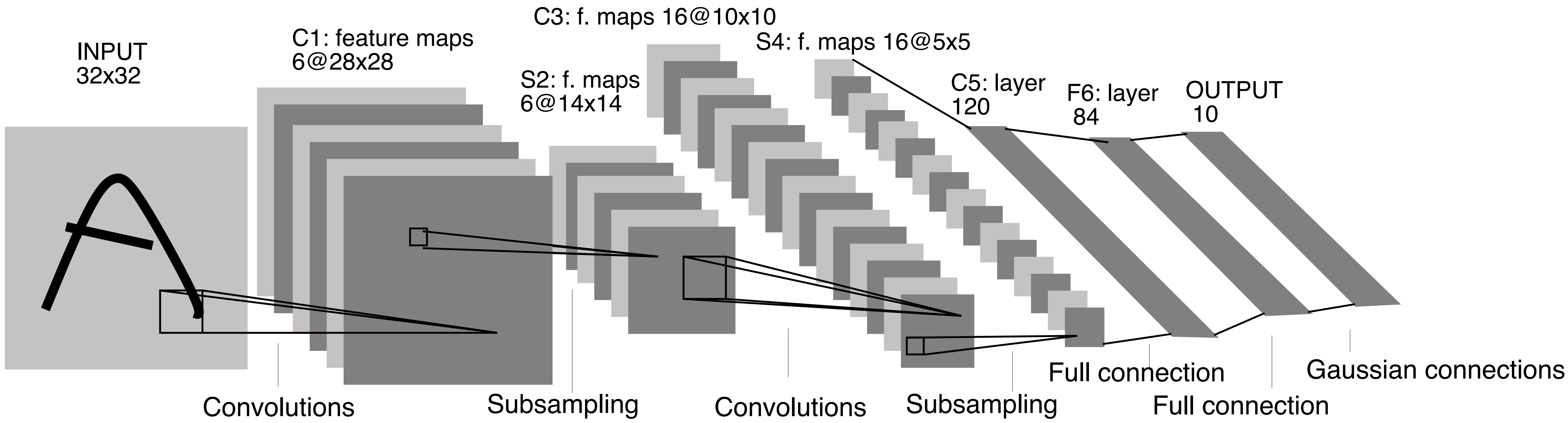


$$x_{k+L} = x_k + \Phi_k(x_k)$$

- Each subnetwork Φ_k is thus learning a *residual* representation
- This allows for training much deeper networks effectively
  - We will come back to this phenomena later.
  - The subnetworks can concentrate on low-dimensional projections without loss of discriminability.

Some Famous CNNs

- "LeNet" for handwritten digit recognition:

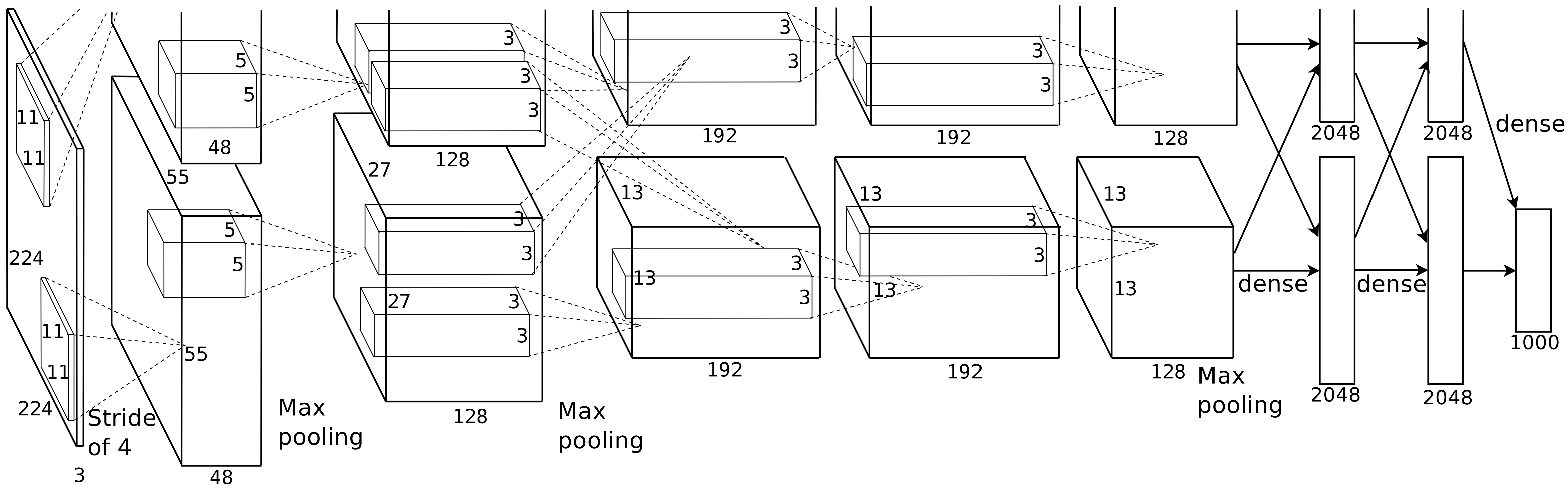


[LeCun, Bottou, Bengio & Hafner '98]

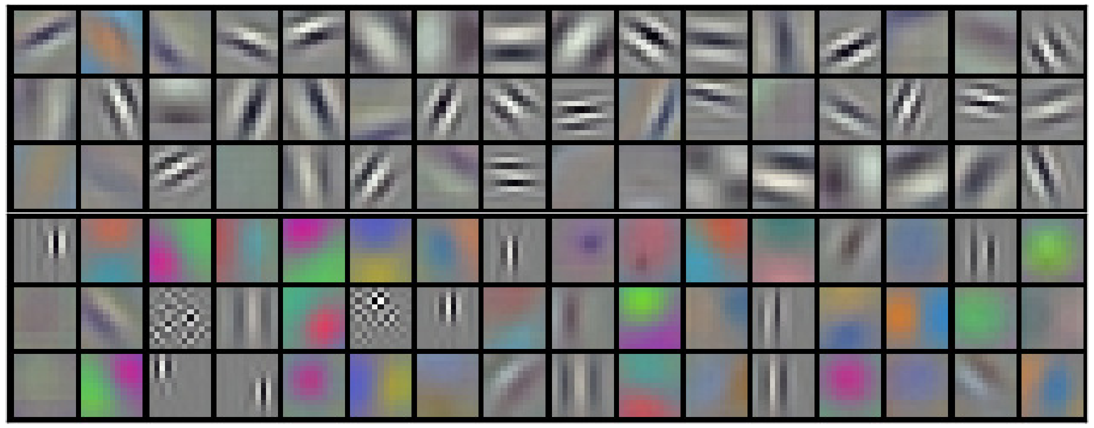
- Uses sigmoidal non-linearities
- 5 layer network with no explicit pooling (trainable).

Some Famous CNNs

- AlexNet [Krizhevsky et al, '12]:

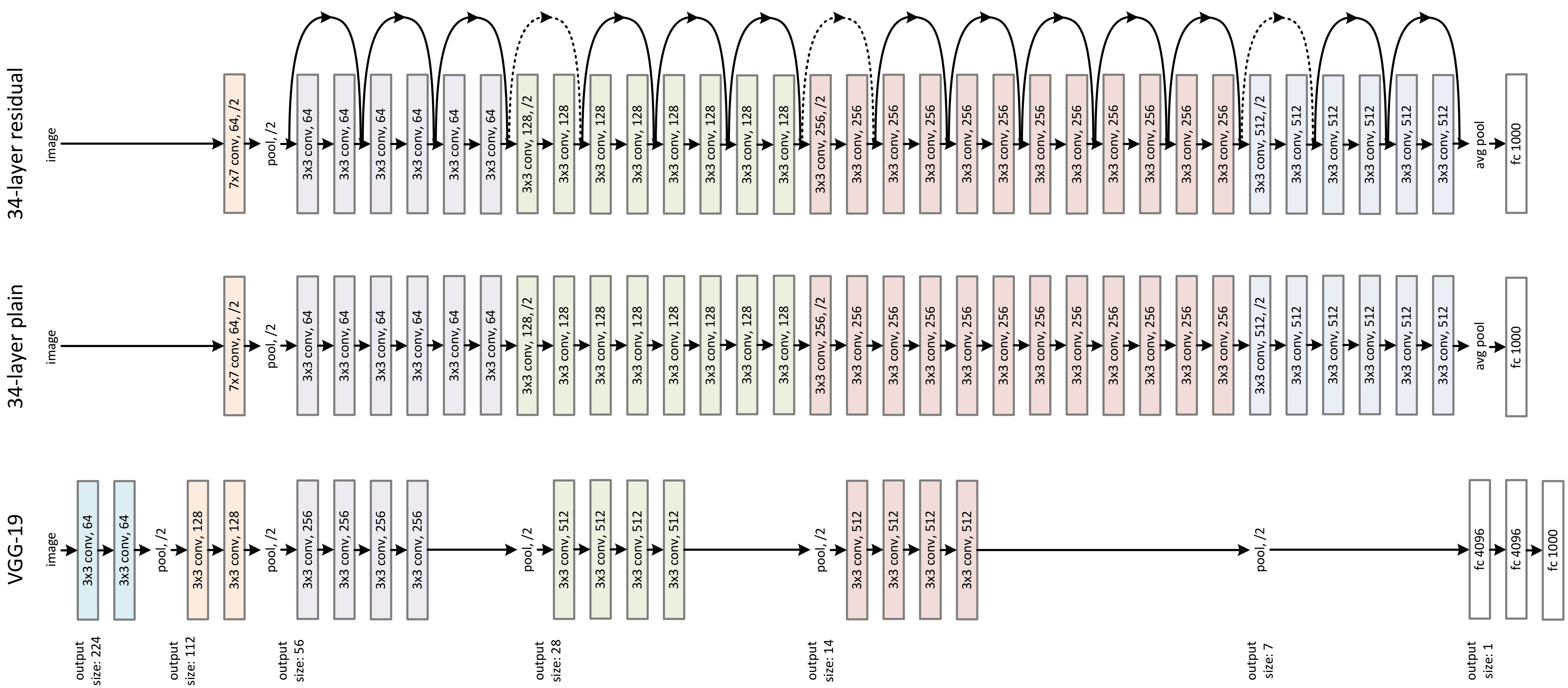


- 5 convolutional layers and 2 "fully connected" layers.
- Employs local normalization.
- Trained on Imagenet with Dropout.



Some Famous CNNs

- *ResNet* [He et al, '15]:



- Trained with linear skip connections.

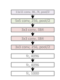
Some Famous CNNs

- "Revolution of Depth" (from Kaiming slides)

Research

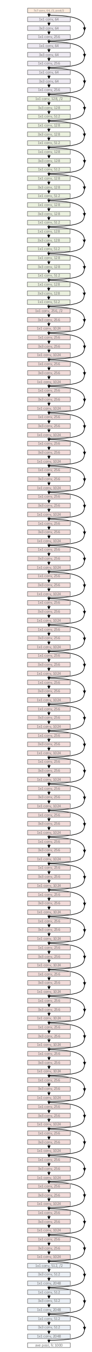
Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)



ICCV15

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.

Properties of learnt CNN representations

Invariance and Covariance

- Do CNNs effectively linearize variability from common transformation groups as a byproduct of supervised training?

Invariance and Covariance

- Do CNNs effectively linearize variability from common transformation groups as a byproduct of supervised training?

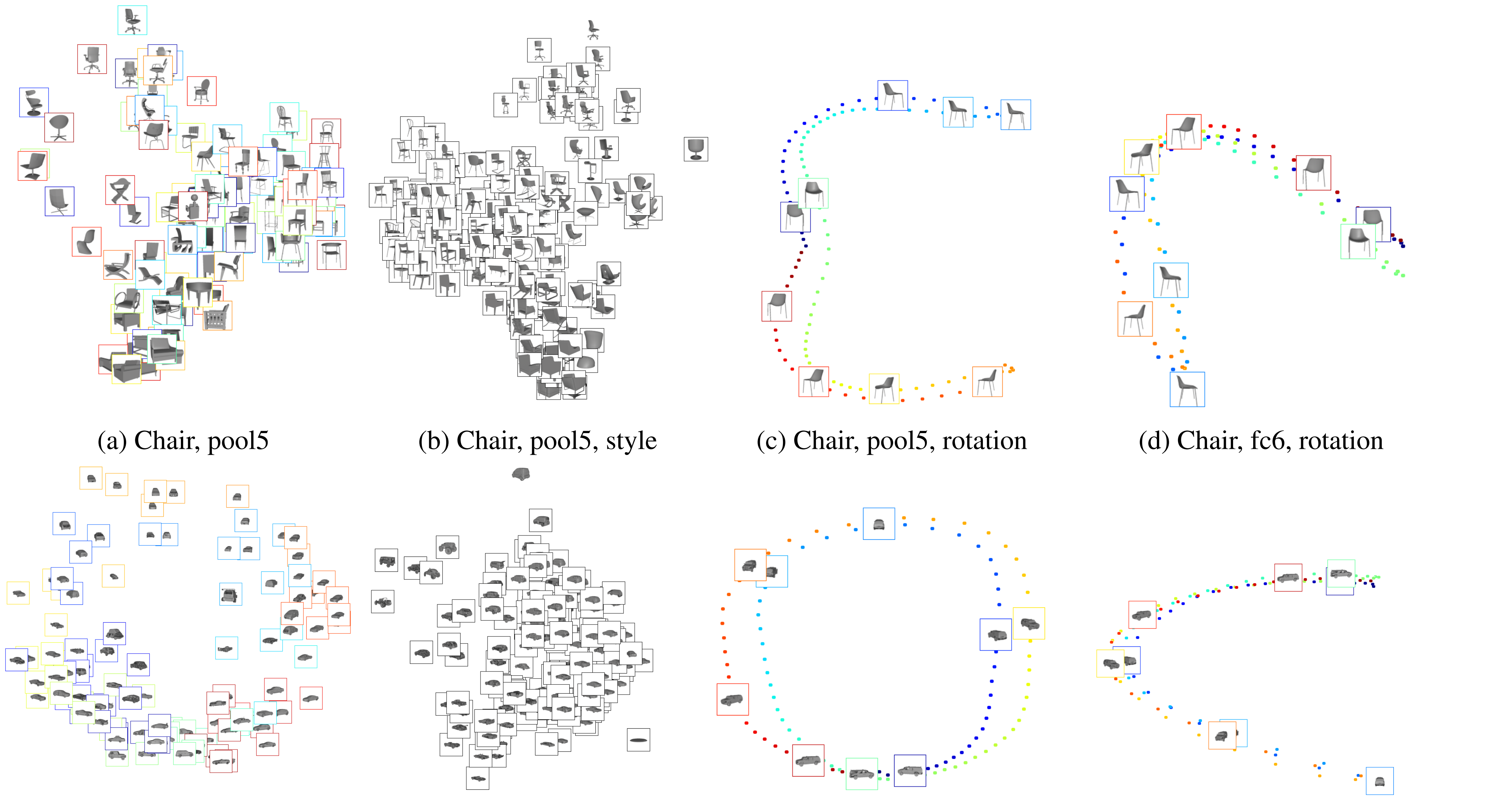
  - [Aubry & Rusell '15] studied this question empirically:

    For each layer k, consider $\Phi_k(x) = x_k(u, \lambda_k)$

    Given a transformation $\varphi(\theta)$ parametrized by θ, perform PCA on $\{\Phi_k(\varphi(\theta)x)\}_{x,\theta}$

Invariance and Covariance

- Principal components corresponding to different factors at different layers:

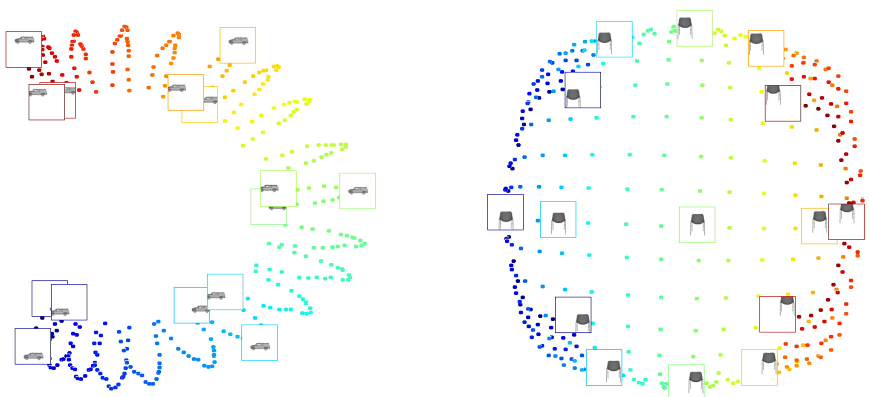


(a) Chair, pool5 (b) Chair, pool5, style (c) Chair, pool5, rotation (d) Chair, fc6, rotation

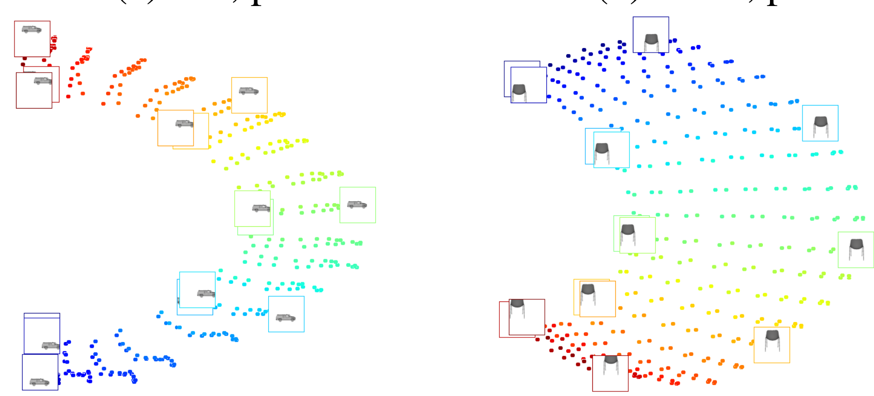
(e) Car, pool5 (f) Car, pool5, style (g) Car, pool5, rotation (h) Car, fc6, rotation

[Aubry & Rusell '15]

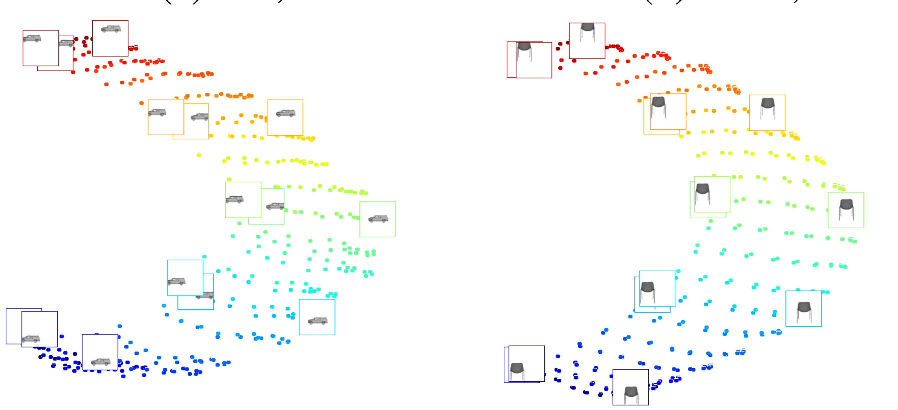
Invariance and Covariance



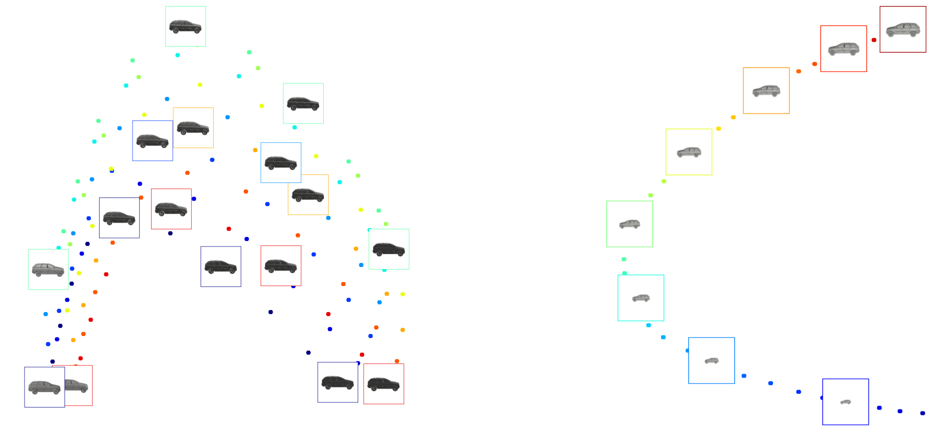
(a) Car, pool5 (b) Chair, pool5



(c) Car, fc6 (d) Chair, fc6



(e) Car, fc7 (f) Chair, fc7



(a) Lighting (b) Scale

(c) Object color (d) Background color

		pool5	fc6	fc7
Viewpoint	Places	26.8 % 8.5	21.4 % 7.0	17.8 % 5.9
	AlexNet	26.4 % 8.3	19.4 % 7.2	15.6 % 6.0
	VGG	21.2 % 10.0	16.4 % 7.7	12.3 % 6.2
Style	Places	26.8 % 136.3	39.1 % 105.5	49.4 % 54.6
	AlexNet	28.2 % 121.1	40.3 % 125.5	49.4 % 96.7
	VGG	26.4 % 181.9	44.3 % 136.3	56.2 % 94.2
Δ^L	Places	46.8 %	39.5 %	32.9 %
	AlexNet	45.0 %	40.3 %	35.0 %
	VGG	52.4 %	39.3 %	31.5 %

[Aubry & Rusell '15]

Invariance and Covariance

- Besides viewpoint and illumination, another major source of variability is clutter:



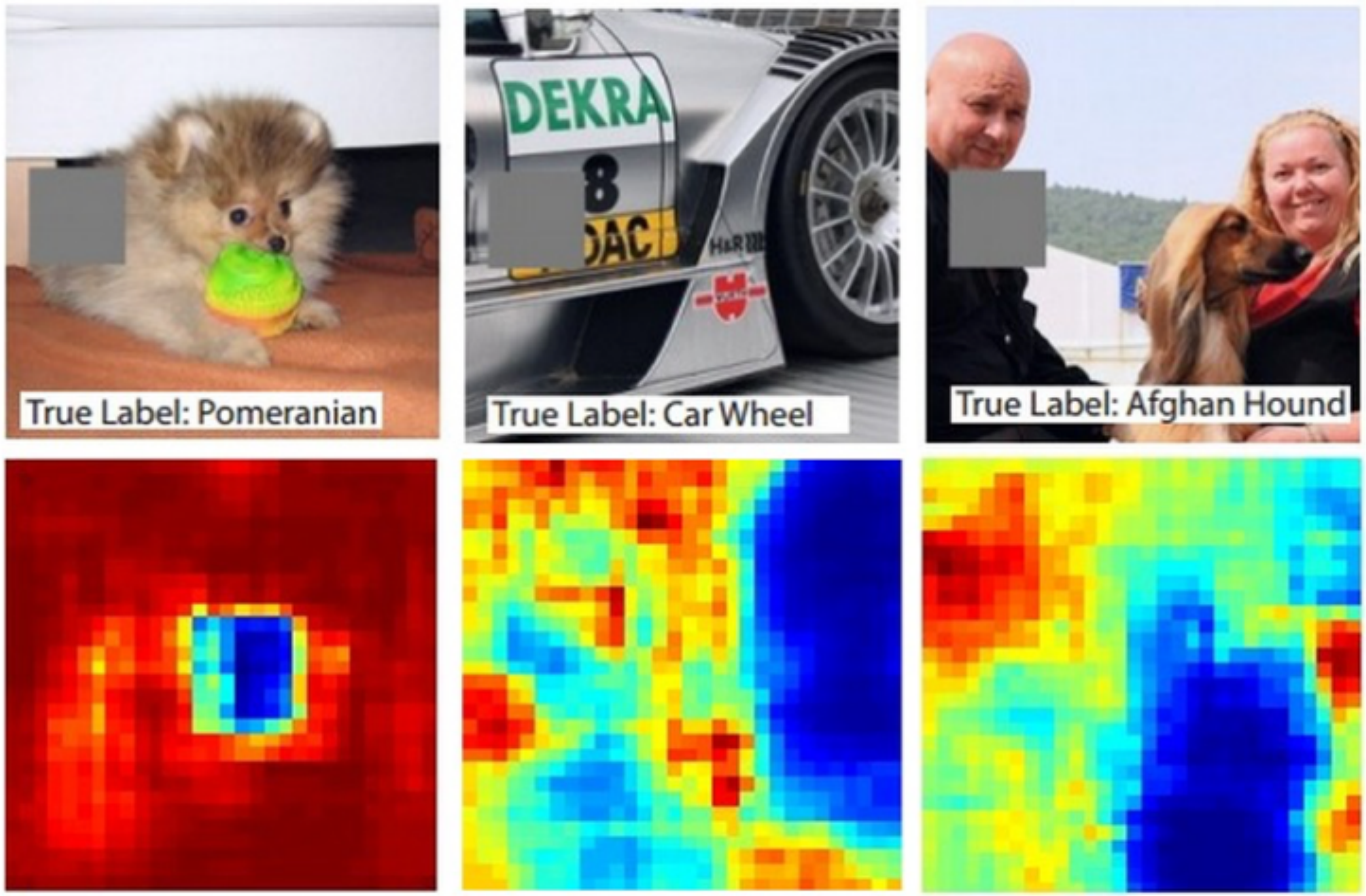

Clutter Robustness

- Clutter: High-dimensional variability
  - The model needs to *detect* a particular object and discard most of the signal energy.
  - The object of interest is localized at a certain scale.
  - Thresholding is an efficient operator to perform detection.

- Are CNNs robust to clutter?

Clutter

- [Zeiler and Fergus, '14]



  - Detection probability as a function of occluding square
  - The network effectively captures

(Un)Stability

- The weakest form of stability is additive:

$$\|\Phi(x+w) - \Phi(x)\| \leq \|w\|$$

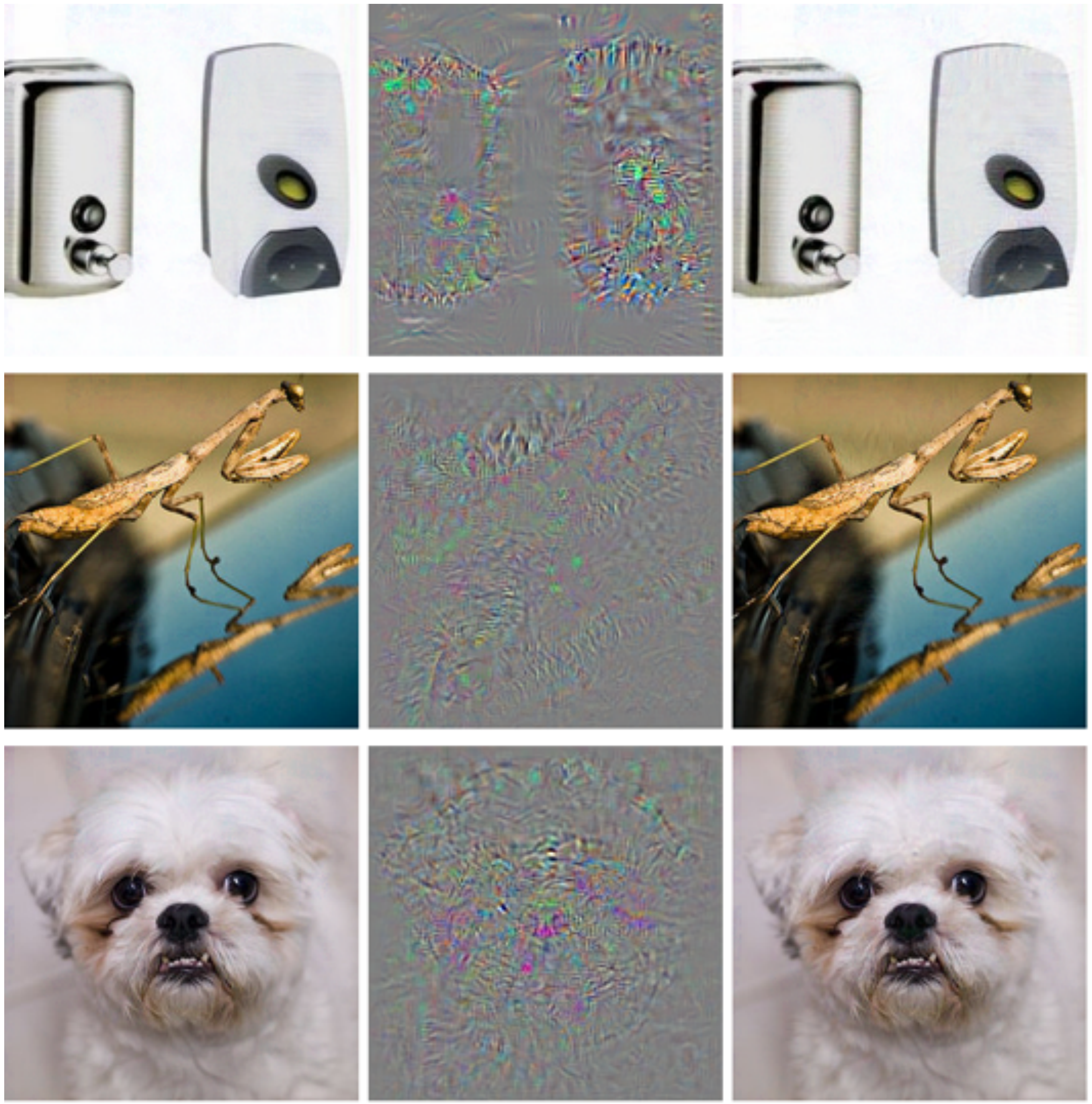
  - We saw that this can be enforced by having convolution tensors with operator norm $\|\Psi_k\| \leq 1$.

- Do CNNs possess this form of stability?
- Does it matter?

Instabilities of Deep Networks

[Szegedy *et al*, ICLR'14]

x $\tilde{x}$



correctly classified

classified as ostrich

Alex Krizhevsky's Imagenet
8 layer Deep ConvNet

$$\|x - \tilde{x}\| < 0.01\|x\|$$

Instabilities of Deep Networks

[joint work with Szegedy *et al*, *ICLR'14*]

- Additive Stability is not enforced.

$$\|\Phi_i(x) - \Phi_i(x')\| \leq \|W_i(x - x')\| \leq \|W_i\| \, \|x - x'\|$$

Layer	Size	$\|W_i\|$
Conv. 1	$3 \times 11 \times 11 \times 96$	2.75
Conv. 2	$96 \times 5 \times 5 \times 256$	10
Conv. 3	$256 \times 3 \times 3 \times 384$	7
Conv. 4	$384 \times 3 \times 3 \times 384$	7.5
Conv. 5	$384 \times 3 \times 3 \times 256$	11
FC. 1	9216×4096	3.12
FC. 2	4096×4096	4
FC. 3	4096×1000	4

(Un)Stability

- These *adversarial* examples are found by explicitly fooling the network:

$$\min \|x - \tilde{x}\|^2 \quad s.t. \quad p(y \mid \Phi(\tilde{x})) \perp p(y \mid \Phi(x))$$

- They are robust to different parametrization of $\Phi(x)$ and to different hyperparameters.

- However, these examples do not occur in practice.