Harvard MBB
Distinguished Lecture II
2019-03-14

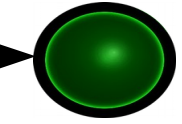
How Could Machines Learn Like Animals & Humans?

Yann LeCun
Facebook AI Research
New York University
http://yann.lecun.com

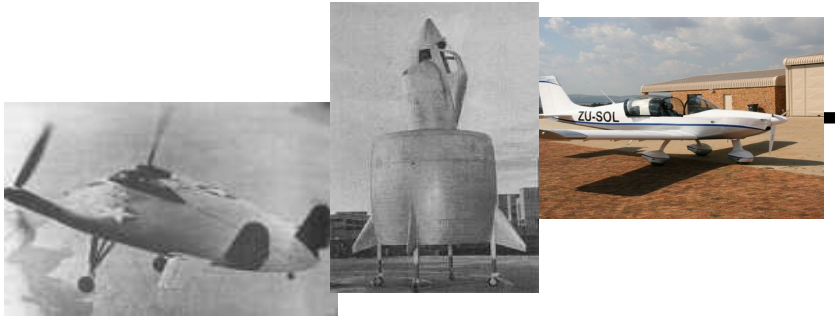
facebook
Artificial Intelligence Research

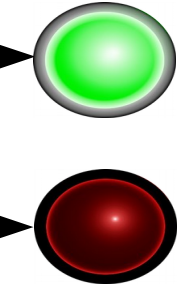
ML/AI today is mostly supervised learning

- **Training a machine by showing examples instead of programming it**
- **When the output is wrong, tweak the parameters of the machine**
- **Works well for:**
  - Speech→words
  - Image→categories
  - Portrait→ name
  - Photo→caption
  - Text→topic
  - ….



CAR



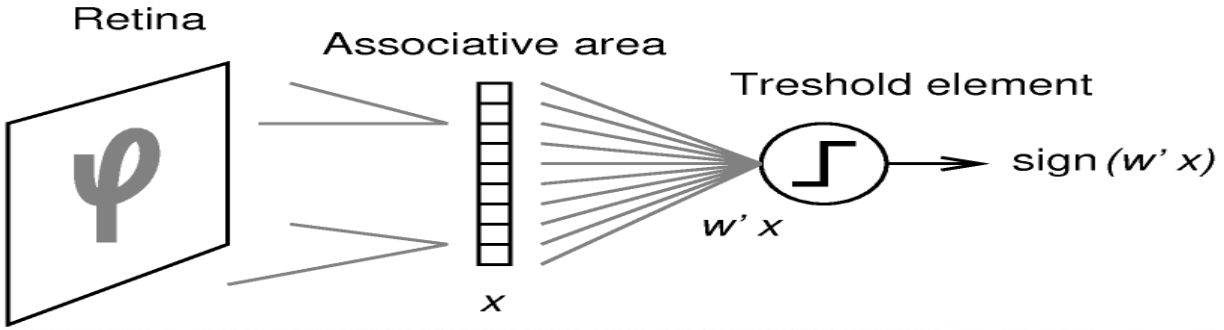


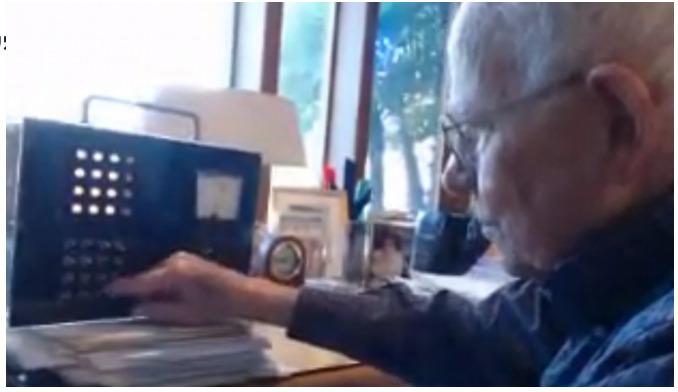
PLANE

The Idea goes back to the Perceptron & Adaline

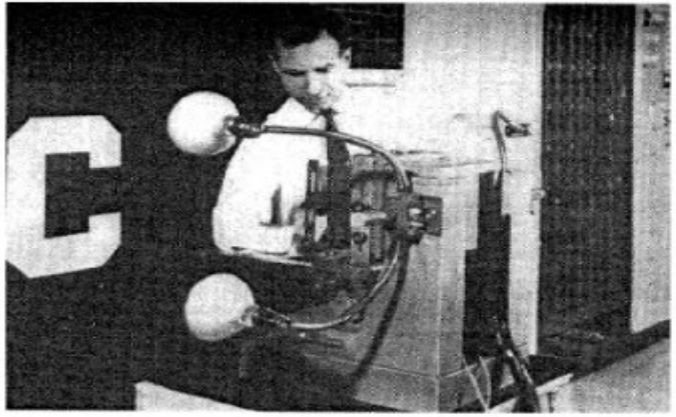
- **The McCulloch-Pitts Binar Neuron**
  - Perceptron: weights are motorized potentiometers
  - Adaline: Weights are electrochemical "memistors"

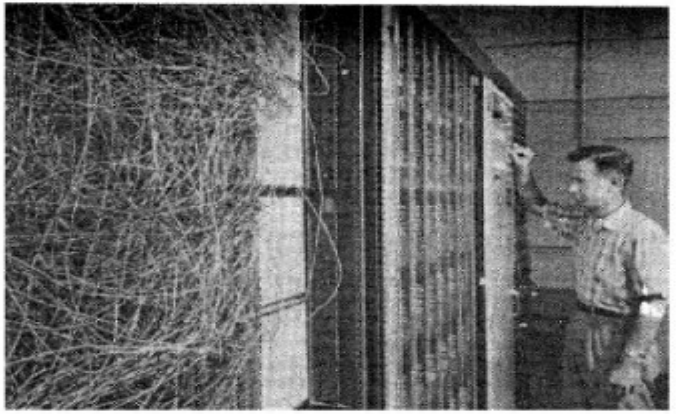
$$y = sign\left(\sum_{i=1}^{N} W_i X_i + b\right)$$

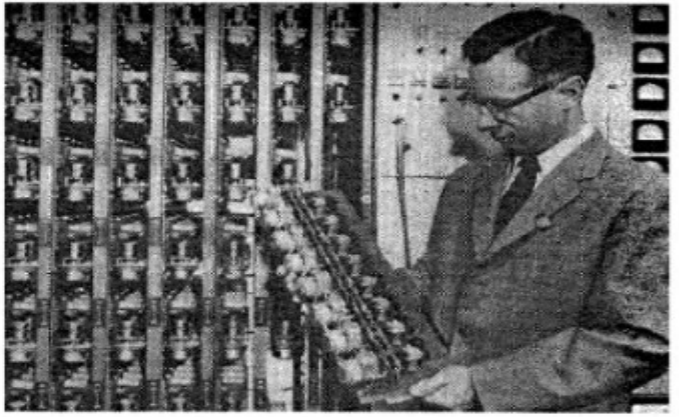




https://youtu.be/X1G2g3SiCwU

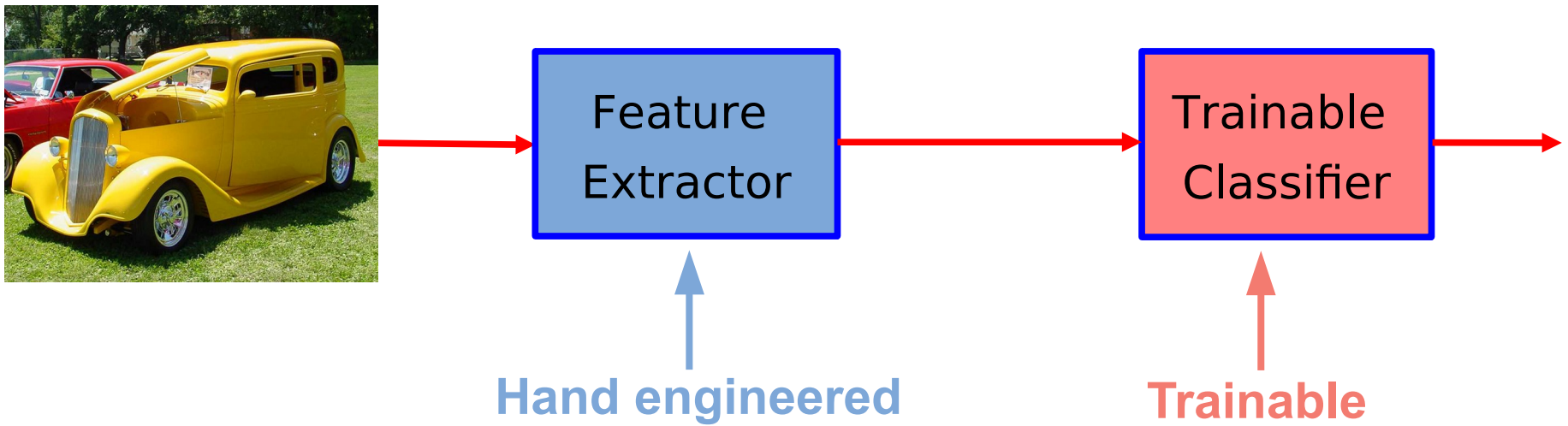






The Standard Paradigm of Pattern Recognition

- **...and "traditional" Machine Learning**



Multilayer Neural Nets and Deep Learning

Traditional Machine Learning

Feature Extractor

Trainable Classifier

Hand engineered

Trainable

Deep Learning

Trainable

Low-Level Features

Mid-Level Features

High-Level Features

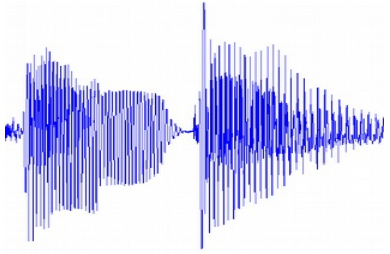
Trainable Classifier

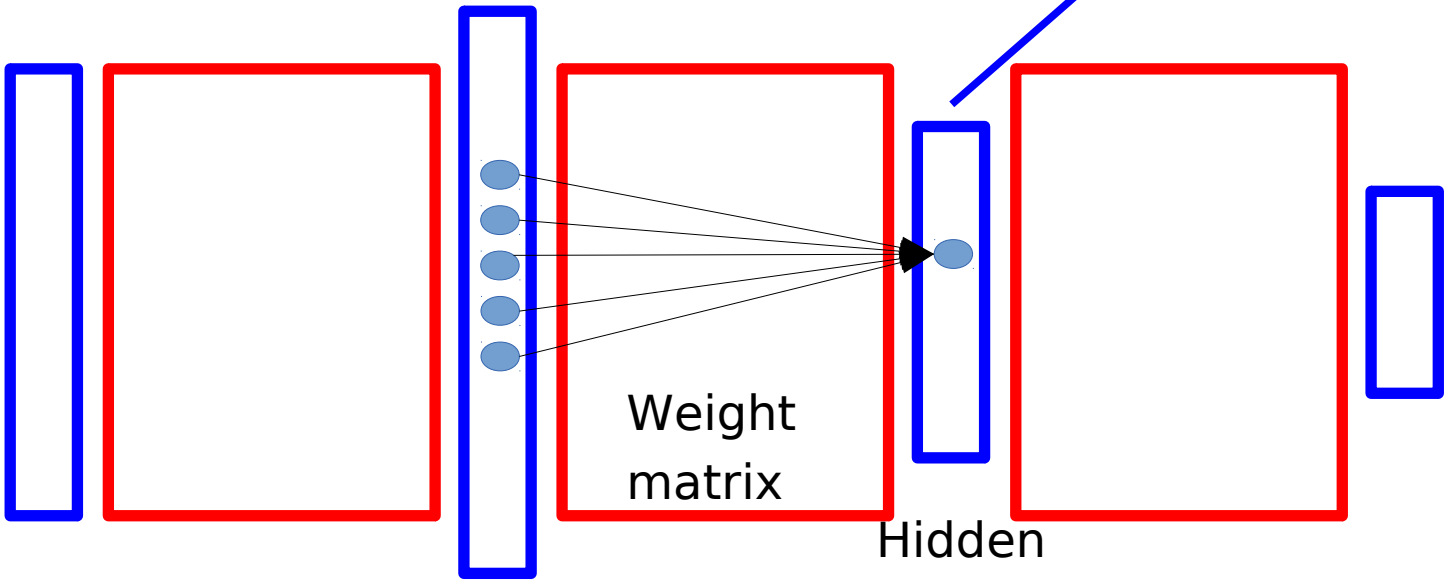
Multi-Layer Neural Nets

- Multiple Layers of simple units
- Each units computes a weighted sum of its inputs
- Weighted sum is passed through a non-linear function
- The learning algorithm changes the weights

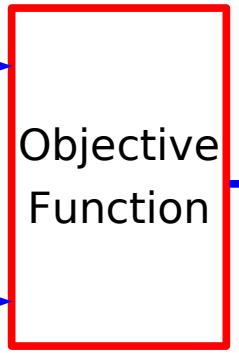
$$ReLU(x) = max(x, 0)$$

Ceci est une voiture

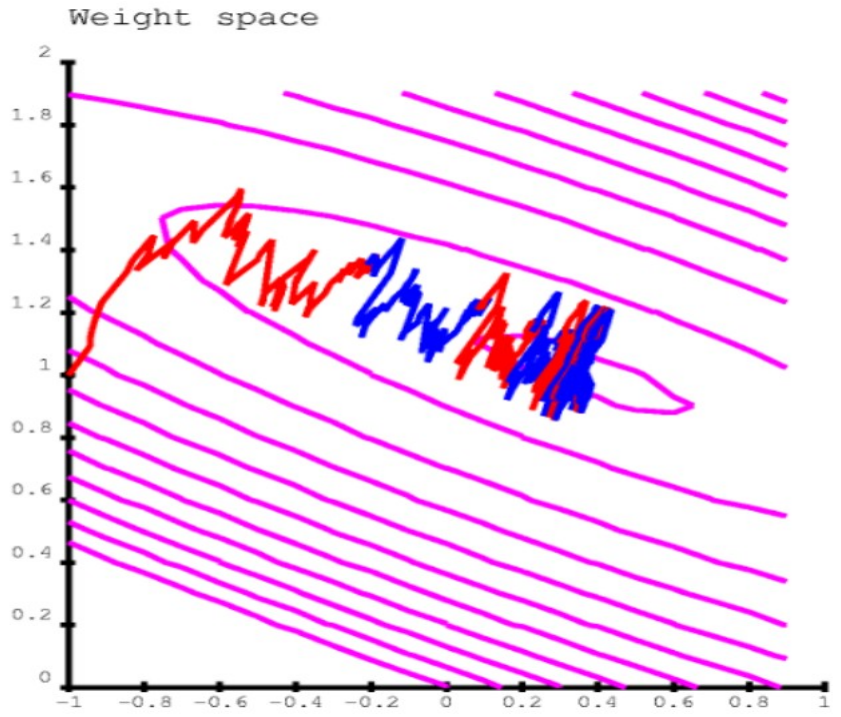




Supervised Machine Learning = Function Optimization



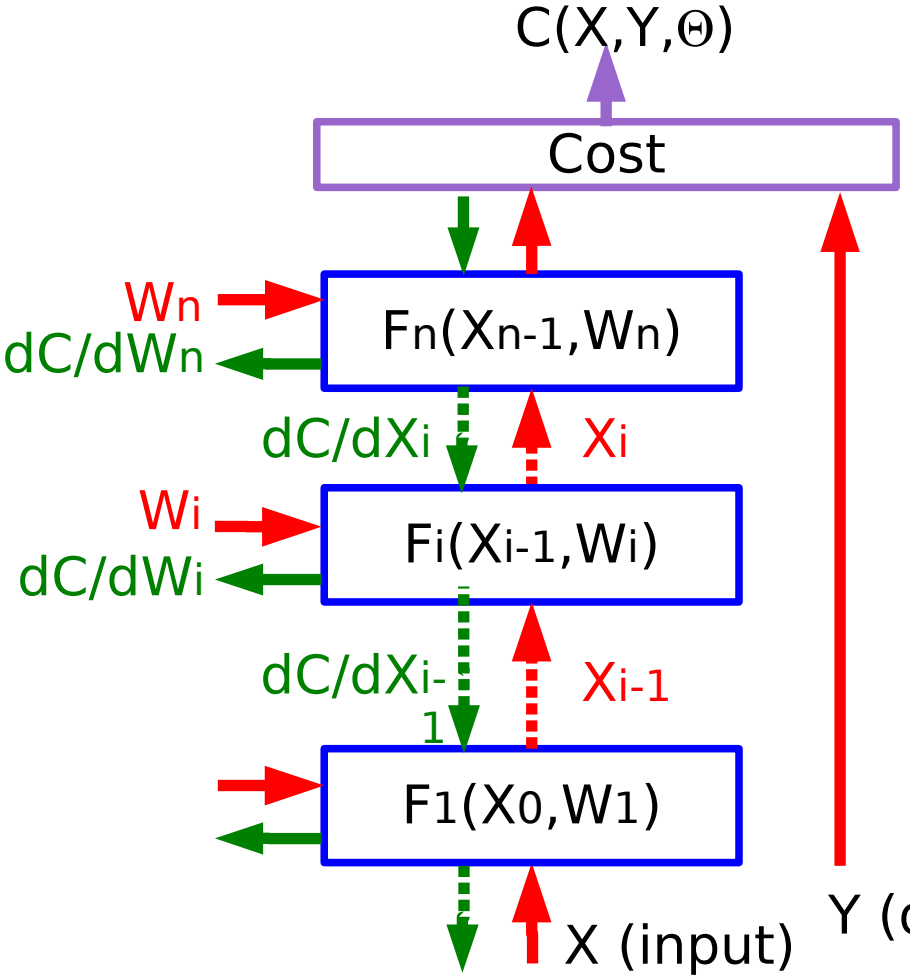
traffic light: -1



- **It's like walking in the mountains in a fog and following the direction of steepest descent to reach the village in the valley**
- **But each sample gives us a noisy estimate of the direction. So our path is a bit random.**
- **Stochastic Gradient Descent (SGD)**

$$W_i \leftarrow W_i - \eta \frac{\partial L(W, X)}{\partial W_i}$$

Computing Gradients by Back-Propagation



- **A practical Application of Chain Rule**

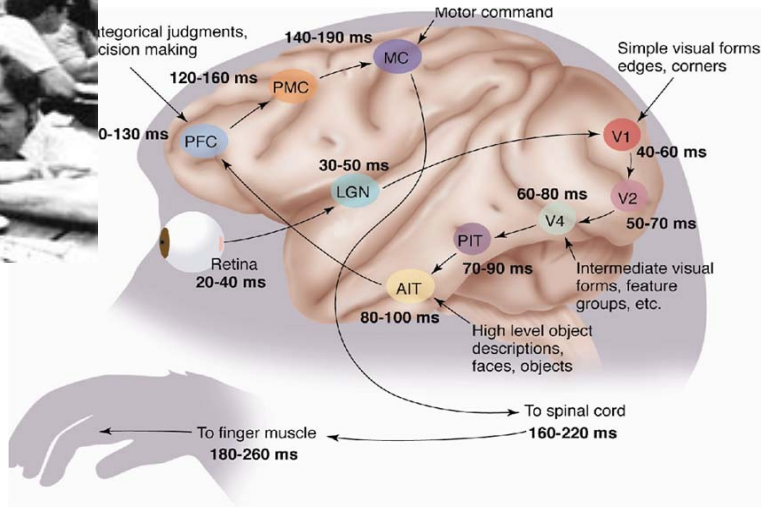
- **Backprop for the state gradients:**
- **$dC/dX_{i-1} = dC/dX_i \cdot dX_i/dX_{i-1}$**
- **$dC/dX_{i-1} = dC/dX_i \cdot dF_i(X_{i-1},W_i)/dX_{i-1}$**

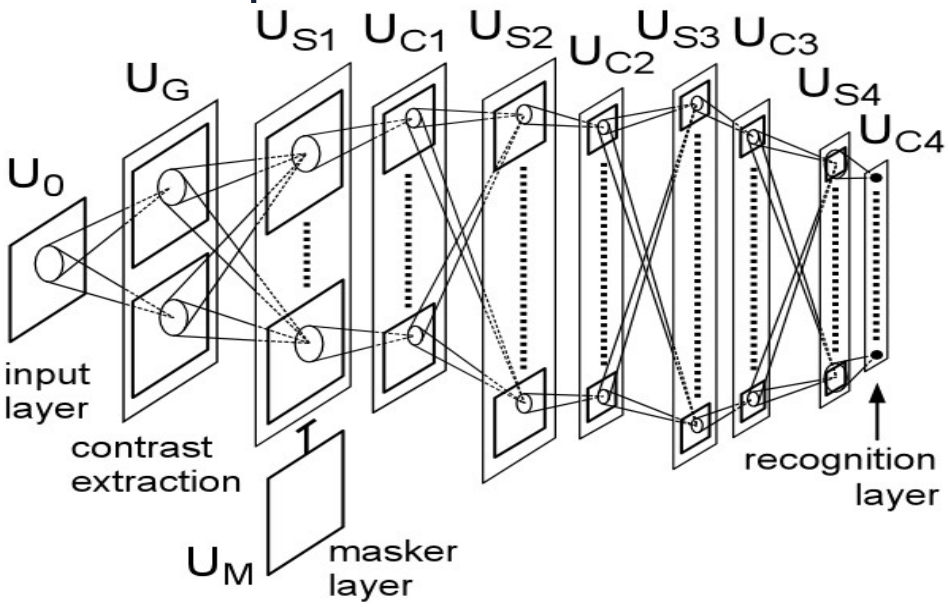
- **Backprop for the weight gradients:**
- **$dC/dW_i = dC/dX_i \cdot dX_i/dW_i$**
- **$dC/dW_i = dC/dX_i \cdot dF_i(X_{i-1},W_i)/dW_i$**

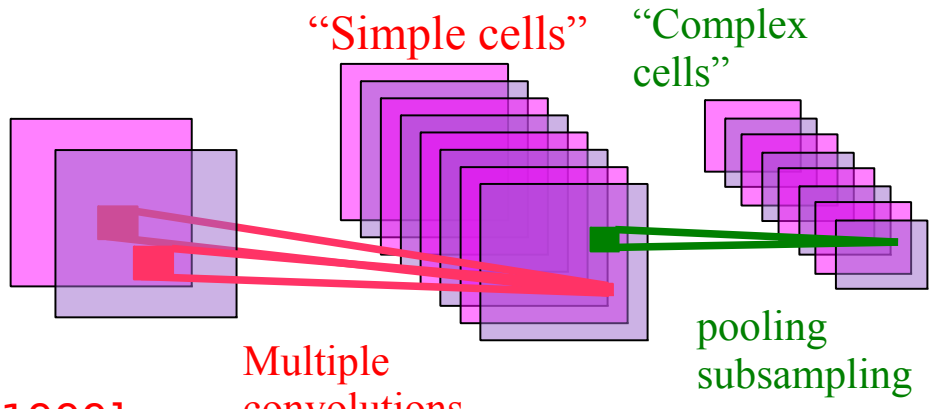
Hubel & Wiesel's Model of the Architecture of the Visual Cortex

- **[Hubel & Wiesel 1962]:**
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a

[Thorpe & Fabre-Thorpe 2001]

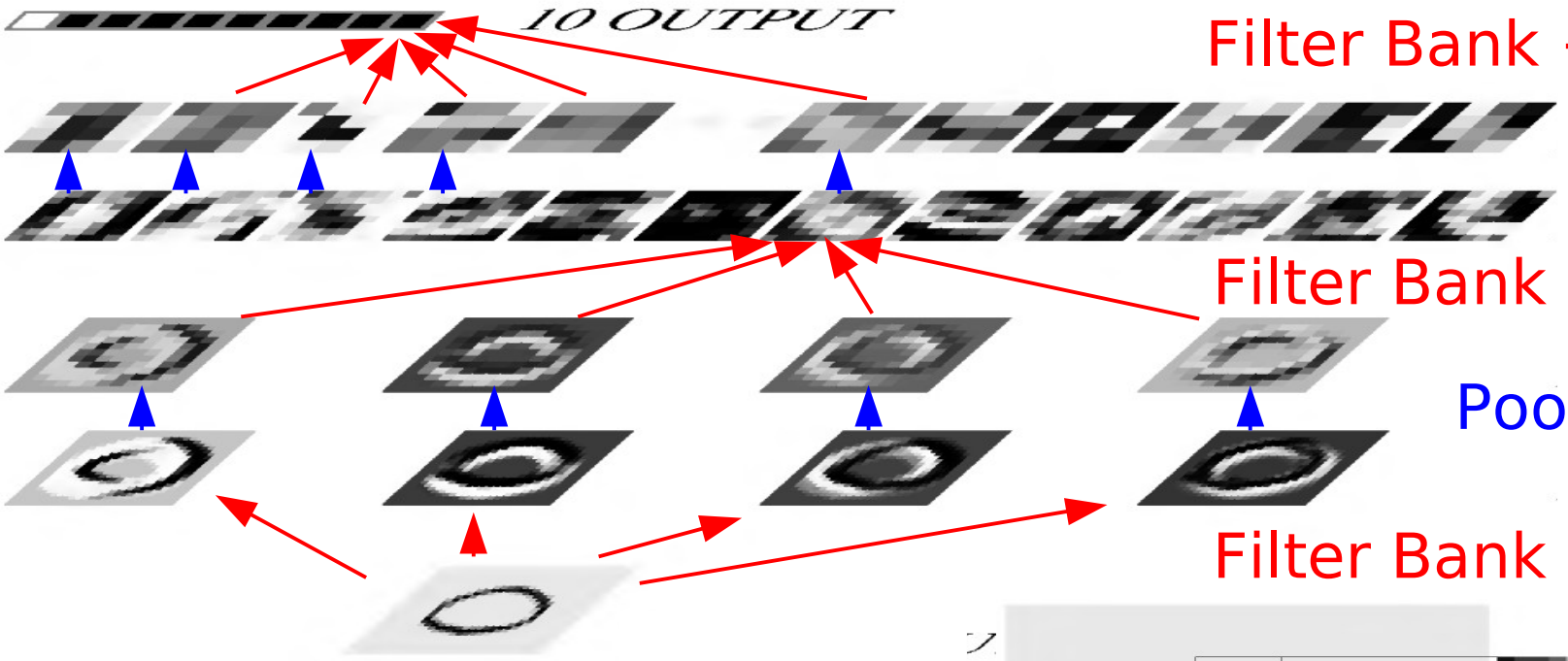






[Fukushima 1982][LeCun 1989, 1998],[Riesenhuber 1999].....

Convolutional Network Architecture [LeCun et al. NIPS 1989]



Filter Bank +non-linearity

Pooling

Filter Bank +non-linearity

Pooling

Filter Bank +non-linearity

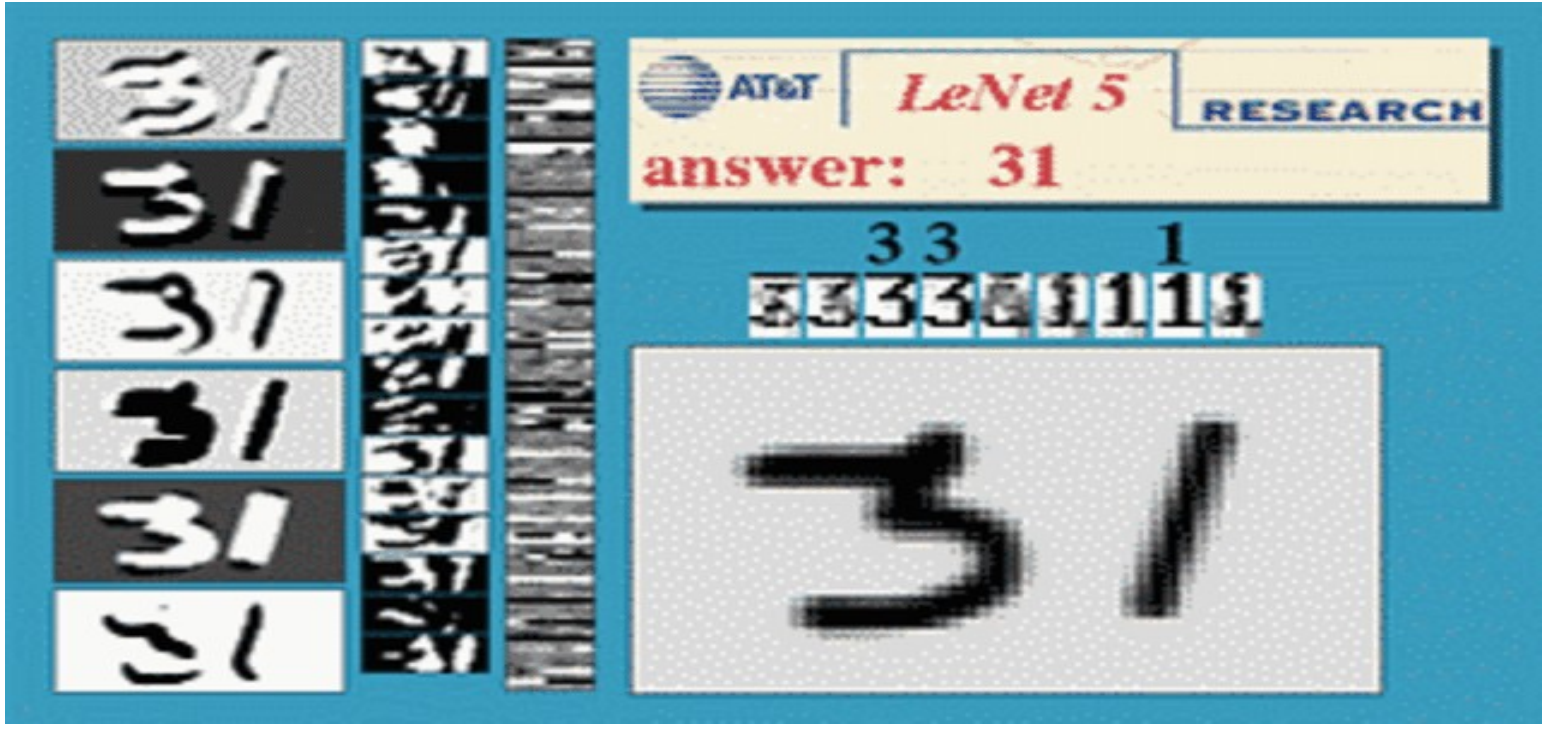
- **Inspired by [Hubel & Wiesel 1962] & [Fukushima 1982] (Neocognitron):**
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.

Convolutional Network (LeNet5, vintage 1990)

- **Filters-tanh → pooling → filters-tanh → pooling → filters-tanh**

ConvNets can recognize multiple objects

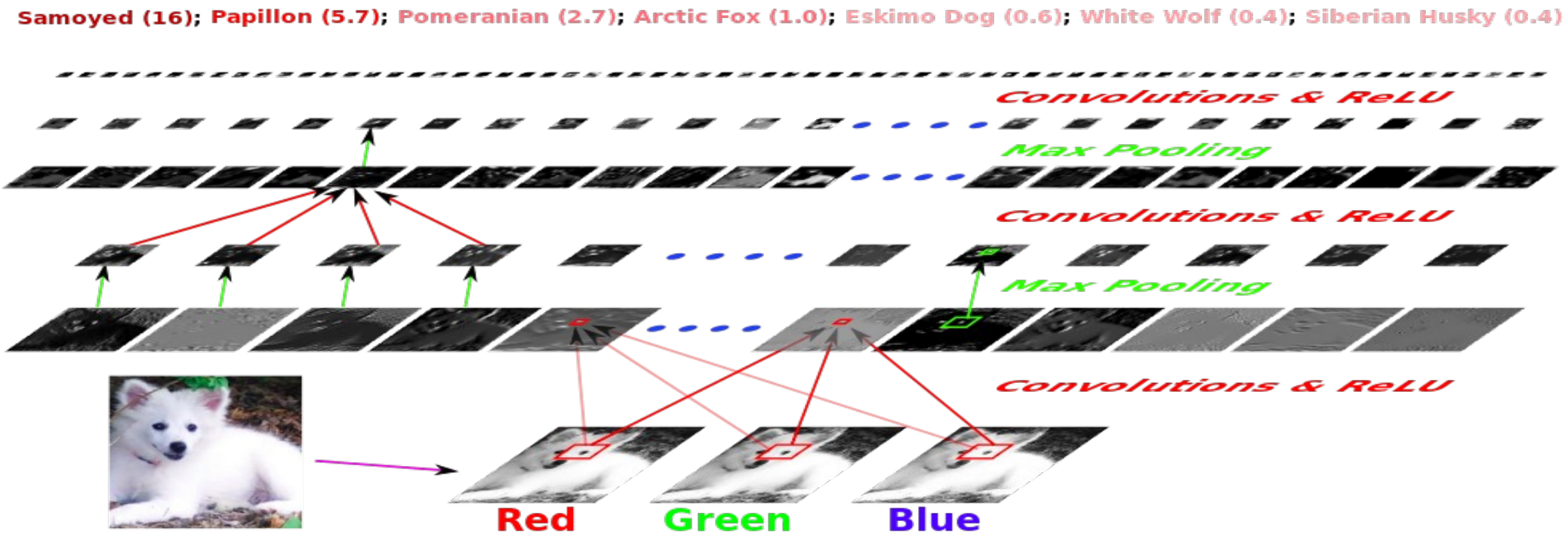
- **All layers are convolutional**
- **Networks performs simultaneous segmentation and recognition**



Semantic Segmentation with ConvNets (33 categories)

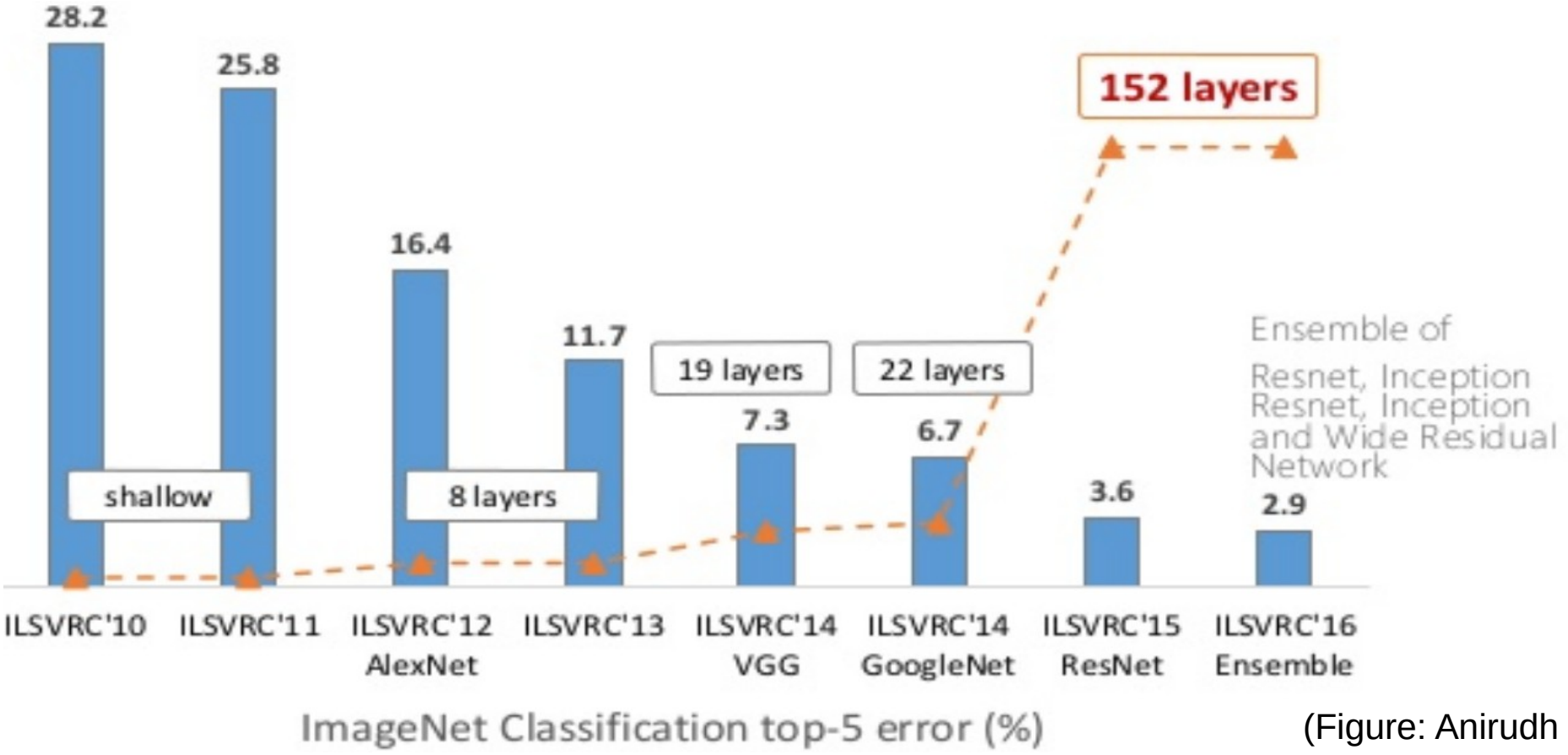
Deep ConvNets for Object Recognition (on GPU)

- AlexNet [Krizhevsky et al. NIPS 2012], OverFeat [Sermanet et al. 2013]
- 1 to 10 billion connections, 10 million to 1 billion parameters, 8 to 20 layers.



Error Rate on ImageNet

- **Depth inflation**

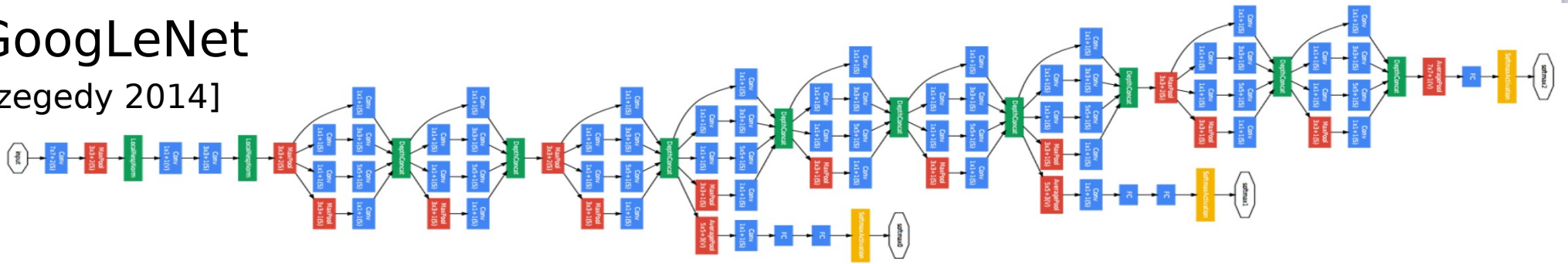


(Figure: Anirudh Koul)

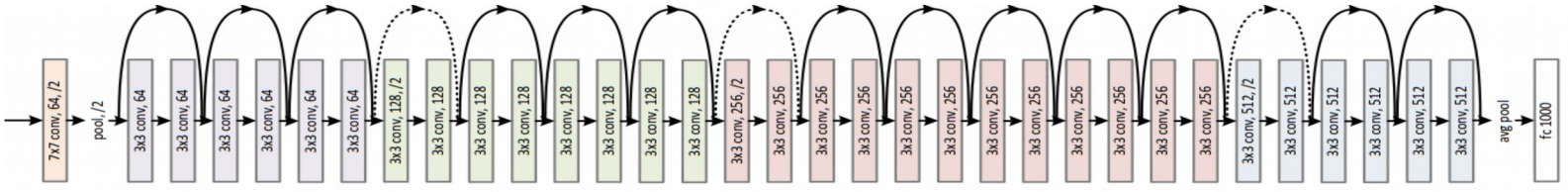
Deep ConvNets (depth inflation)

VGG
[Simonyan 2013]

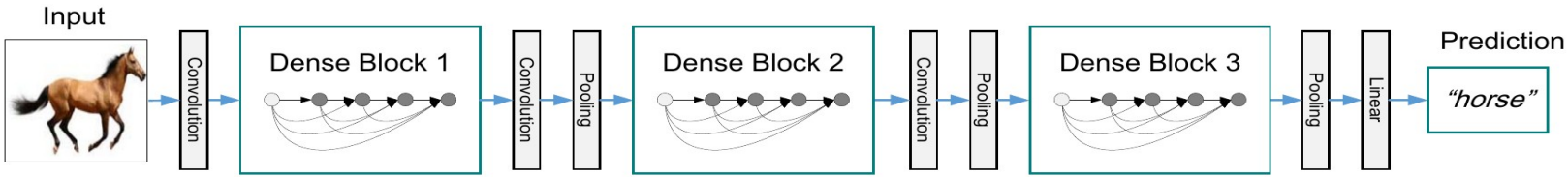
GoogLeNet
Szegedy 2014]



ResNet
[He et al. 2015]

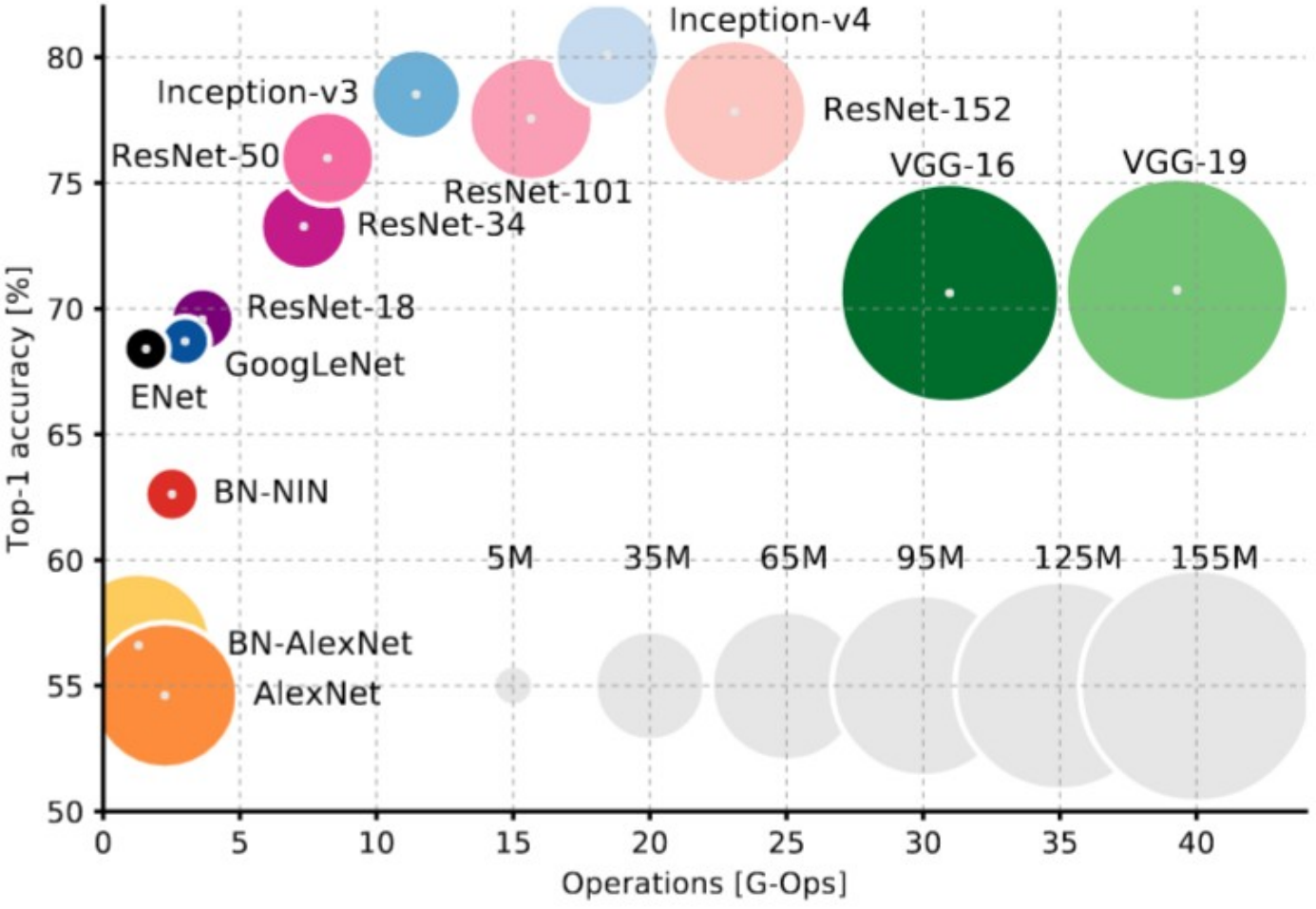


DenseNet
[Huang et al 2017]



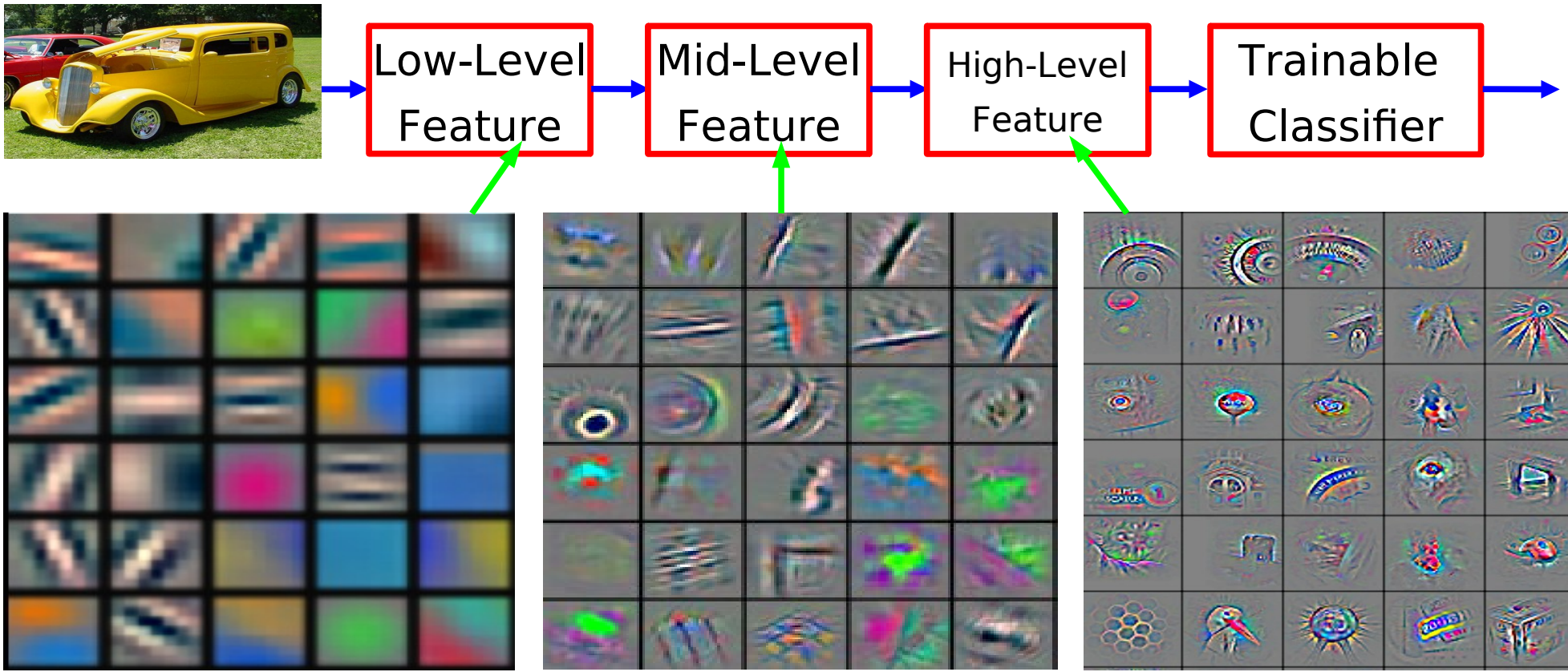
GOPS vs Accuracy on ImageNet vs #Parameters

- [Canziani 2016]
- ResNet50 and ResNet100 are used routinely in production.
- Each of the few billions photos uploaded on Facebook every day goes through a handful of ConvNets within 2 seconds.



Multilayer Architectures == Compositional Structure of Data

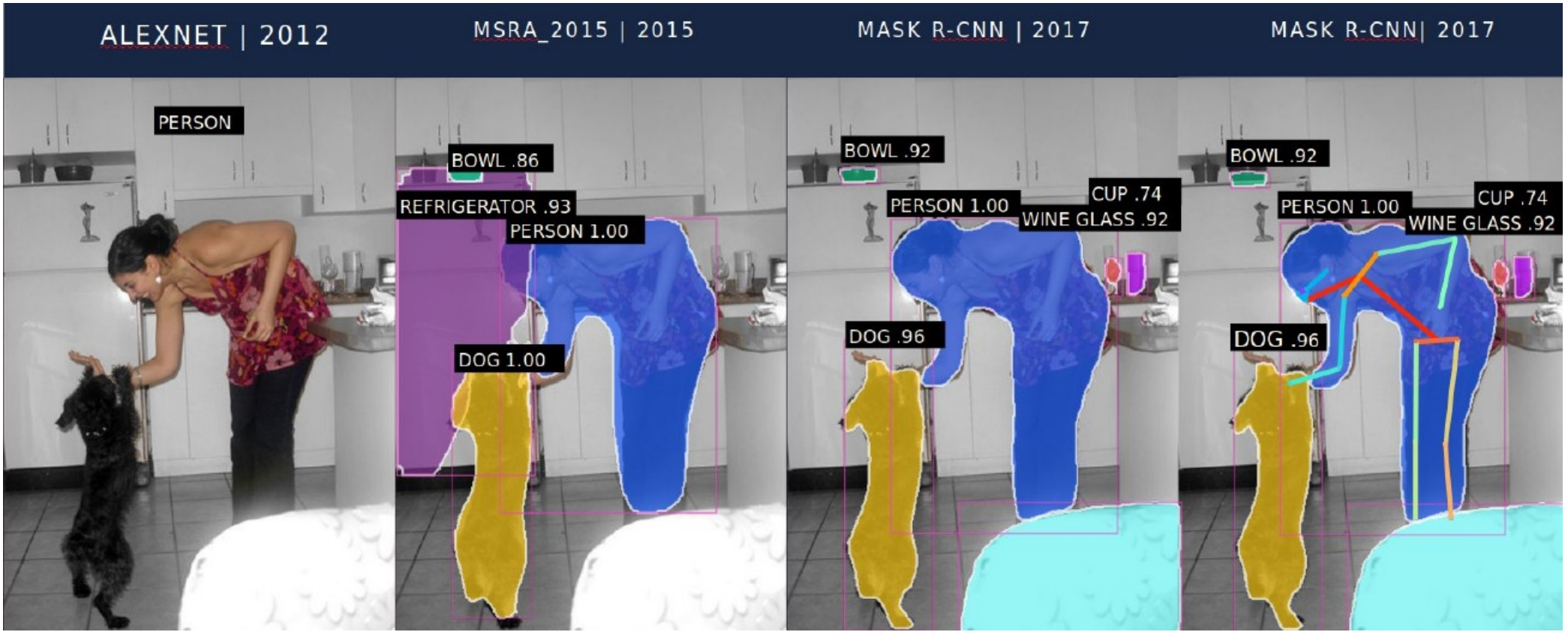
- **Natural is data is compositional => it is efficiently representable hierarchically**



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

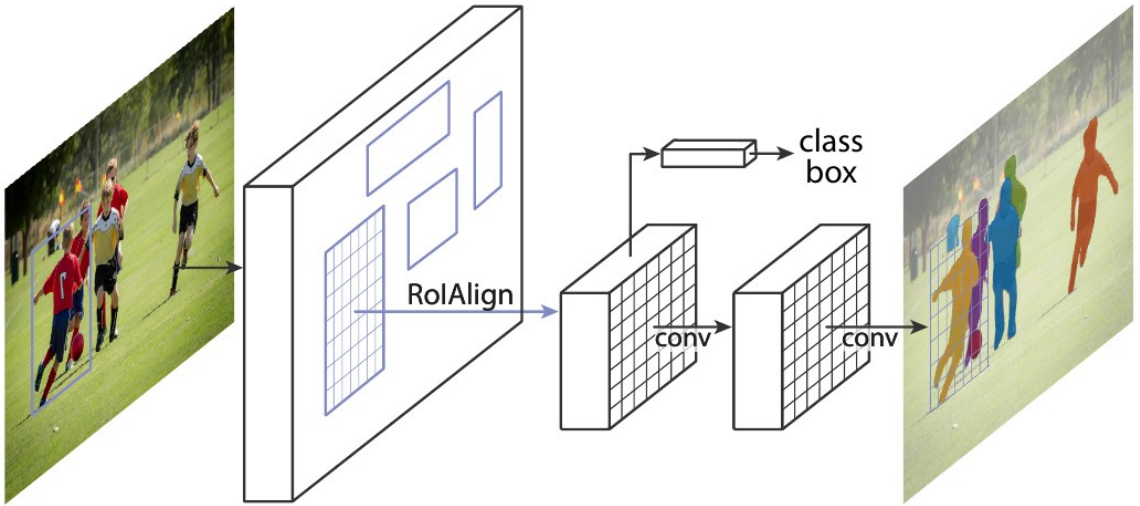
Progress in Computer Vision

- **[He 2017]**



Mask R-CNN: instance segmentation

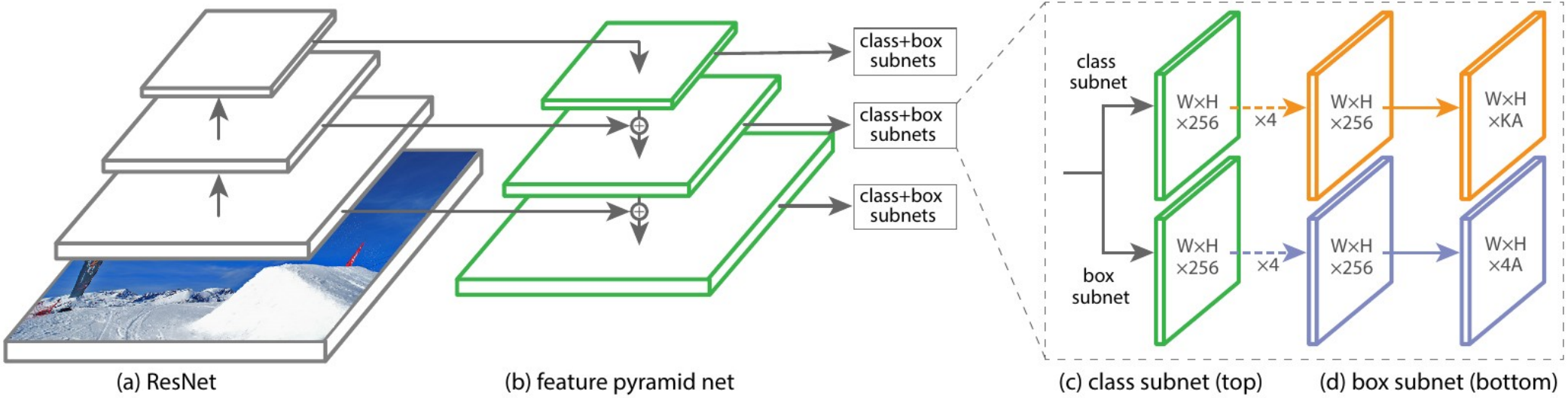
- [He, Gkioxari, Dollar, Girshick arXiv:1703.06870]
- ConvNet produces an object mask for each region of interest
- Combined ventral and dorsal pathways



	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [7]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [20] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [20] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

RetinaNet, feature pyramid network

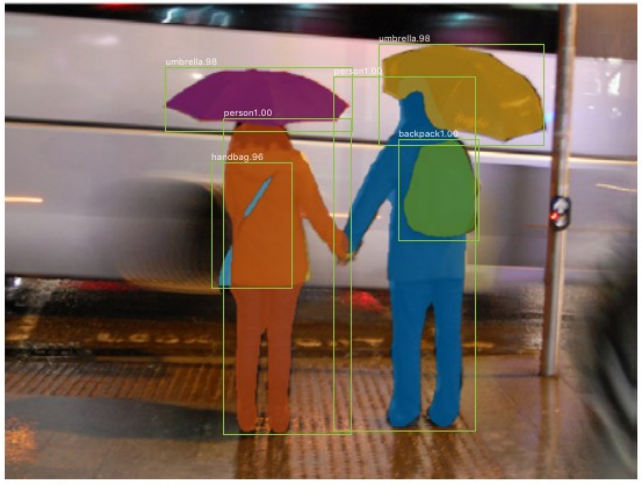
- **One-pass object detection**
- **[Lin et al. ArXiv:1708.02002]**

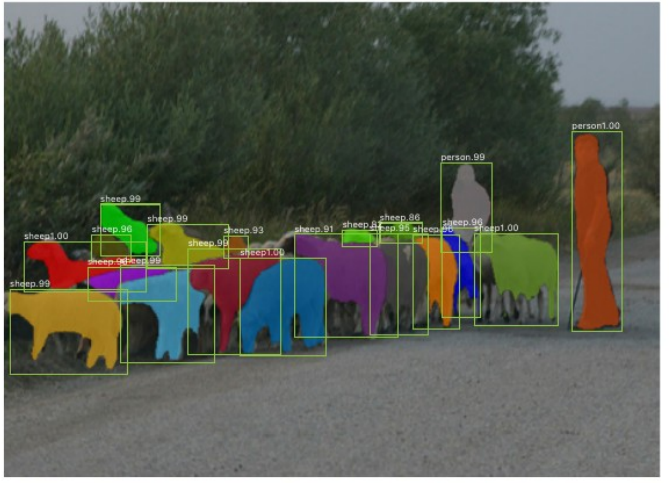


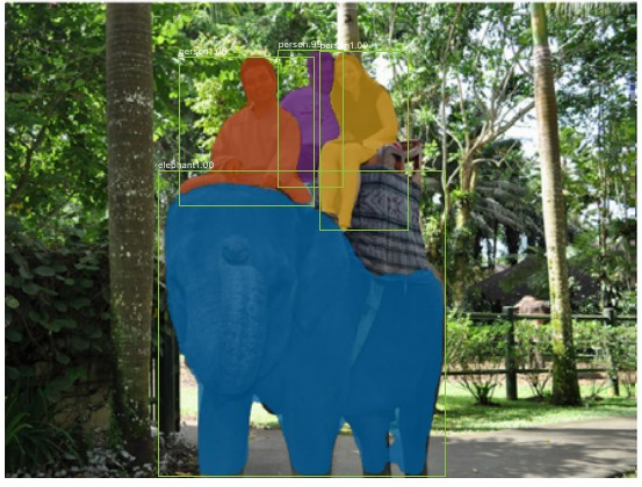
Mask-RCNN Results on COCO dataset

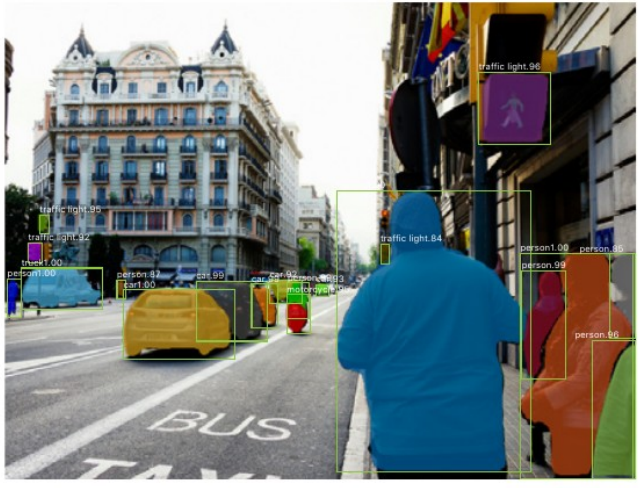
- **Individual objects are segmented.**

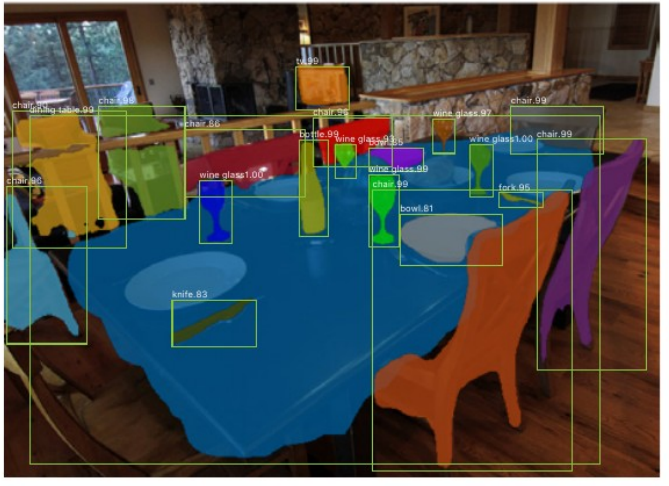
Mask R-CNN Results on COCO test set





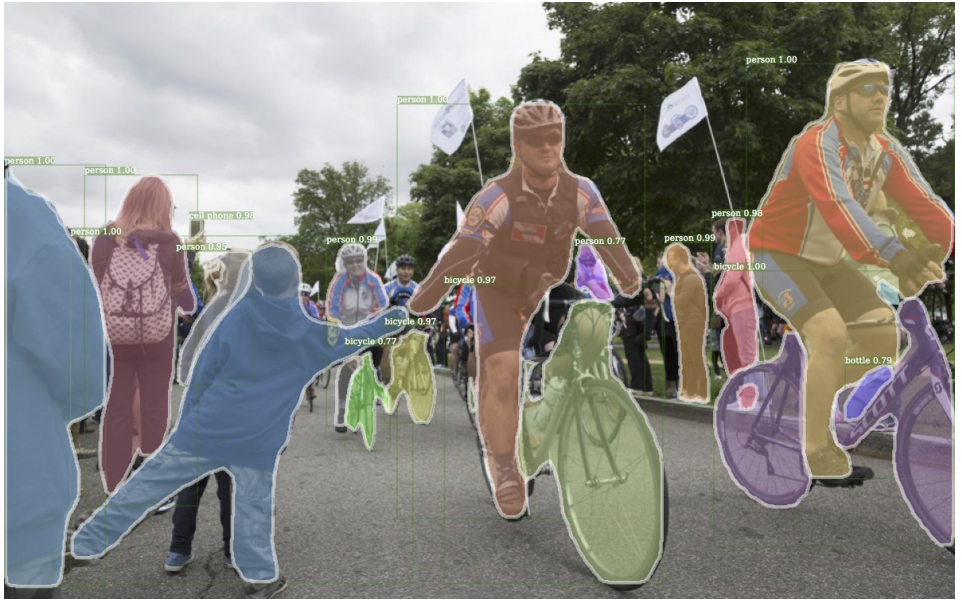


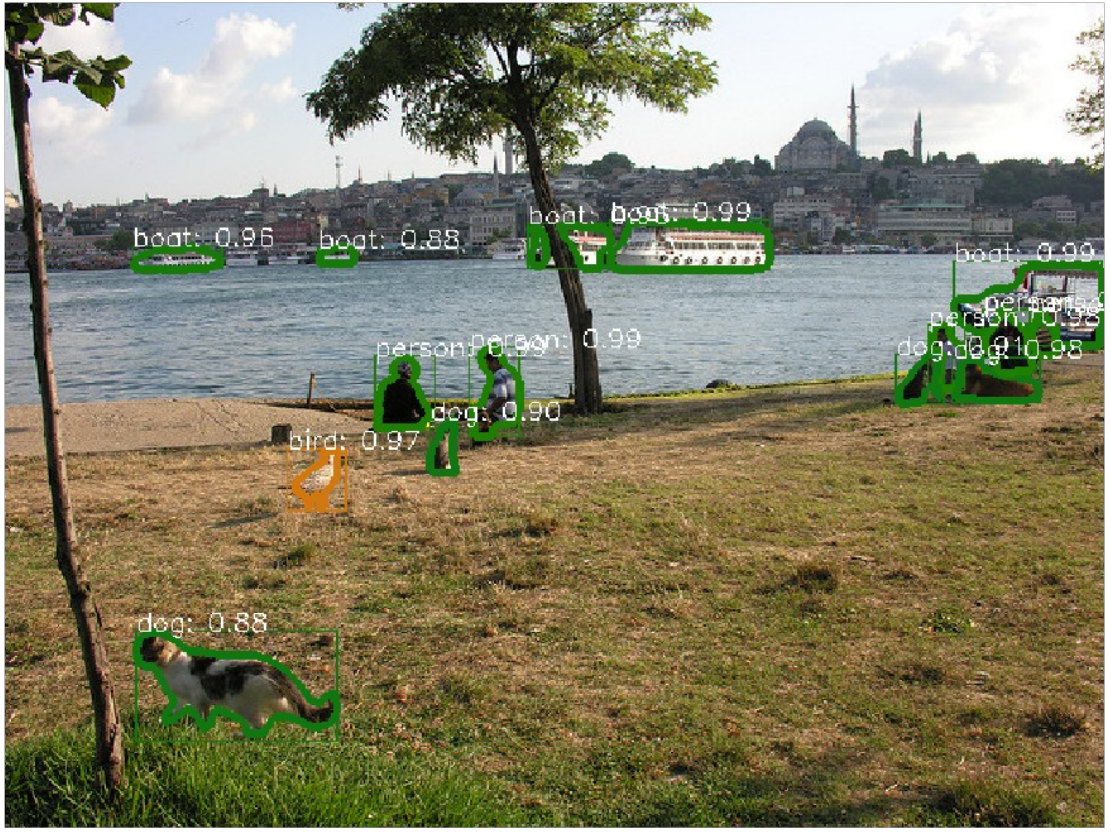




Detectron: open source vision in PyTorch

https://github.com/facebookresearch/maskrcnn-benchmark



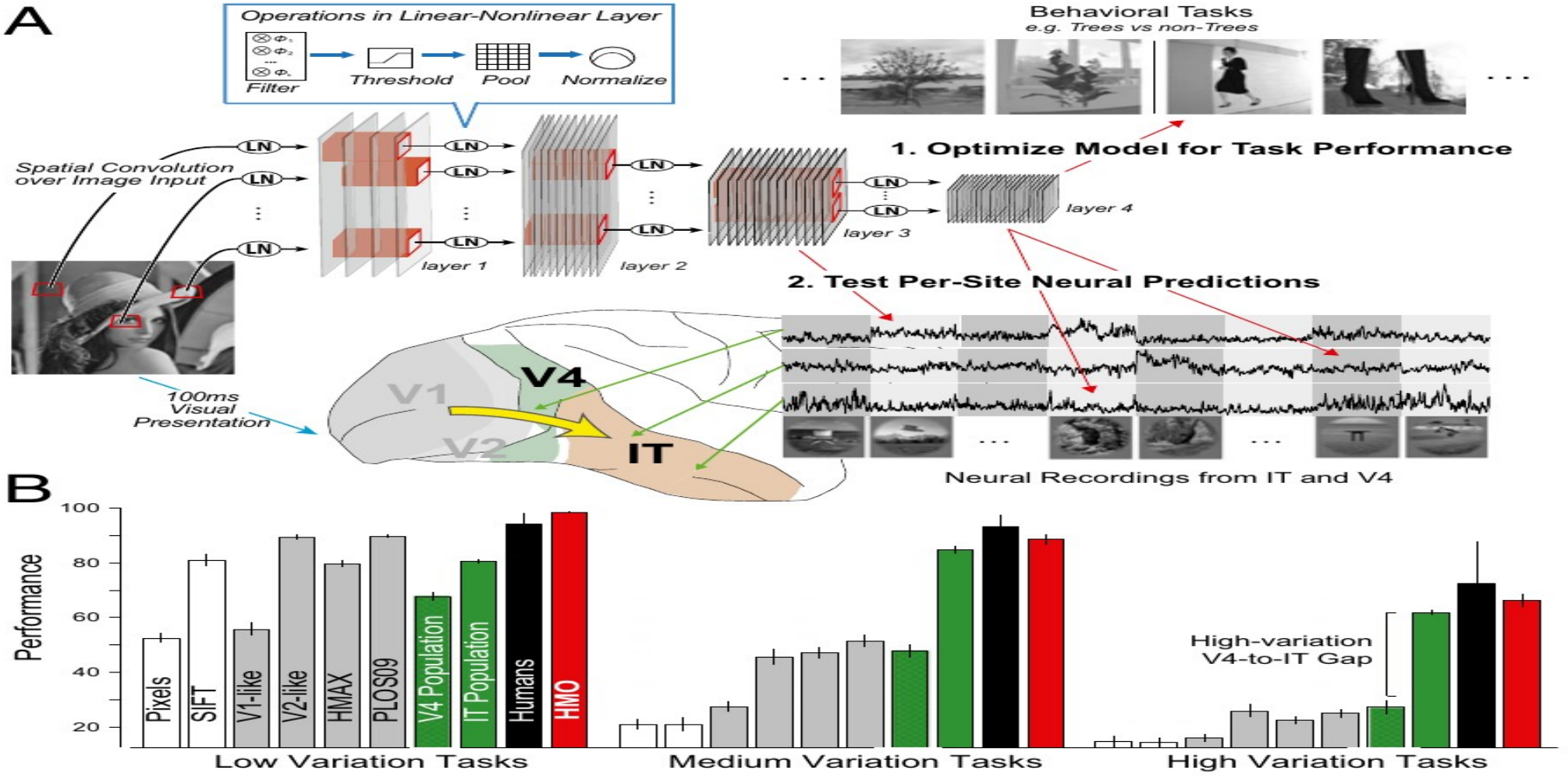


Biologically Correct ConvNets

Good models of the ventral pathway?

facebook
Artificial Intelligence Research

ConvNets & The Visual System [Yamins et al. PNAS 2014]



ConvNets as Models of the Visual System?

- **[Yamins & Di Carlo 2016]**

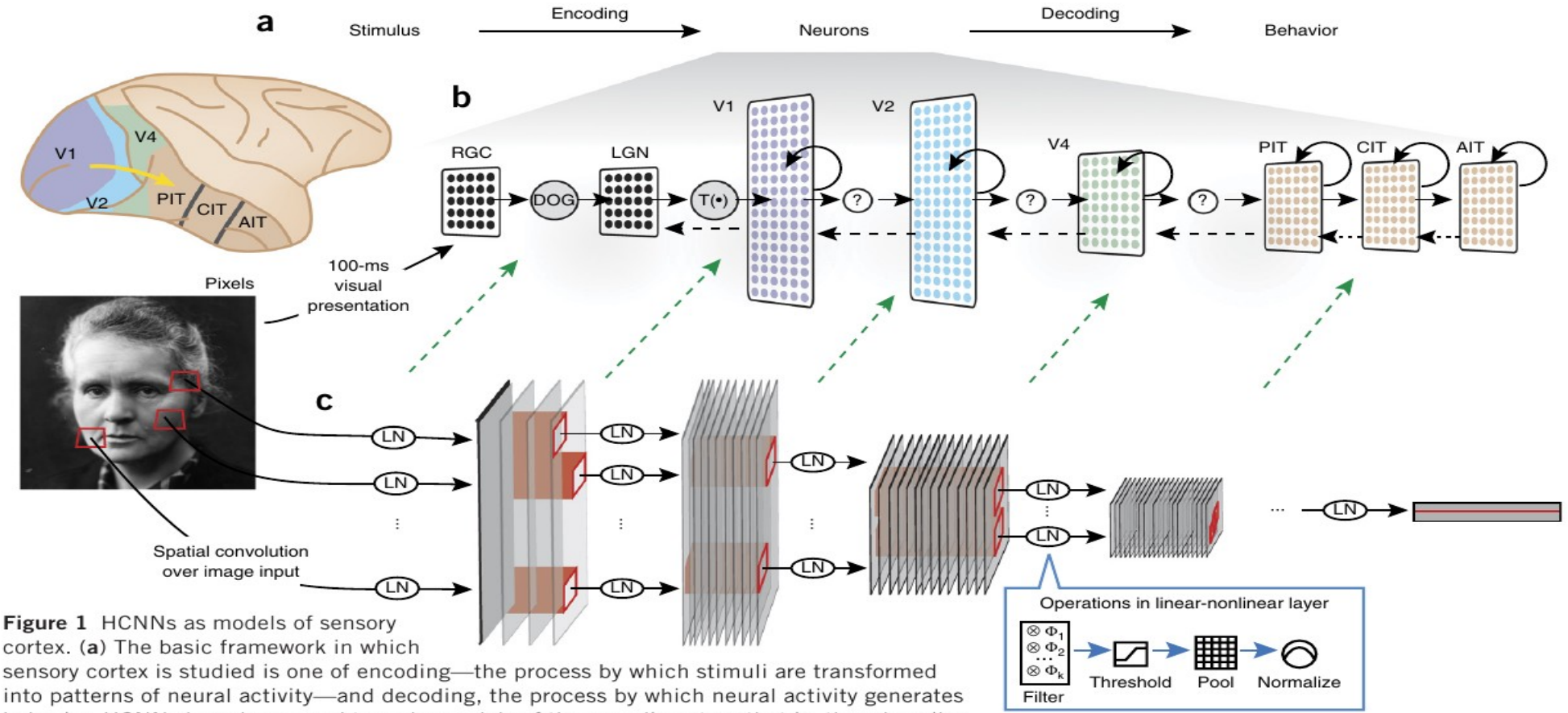
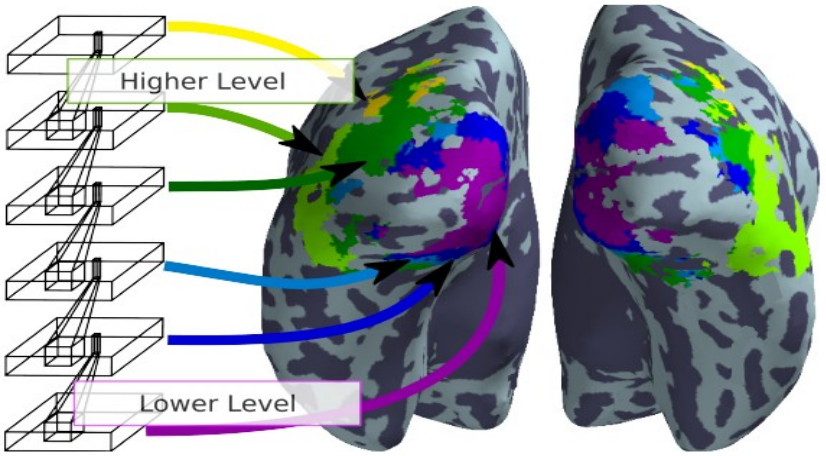


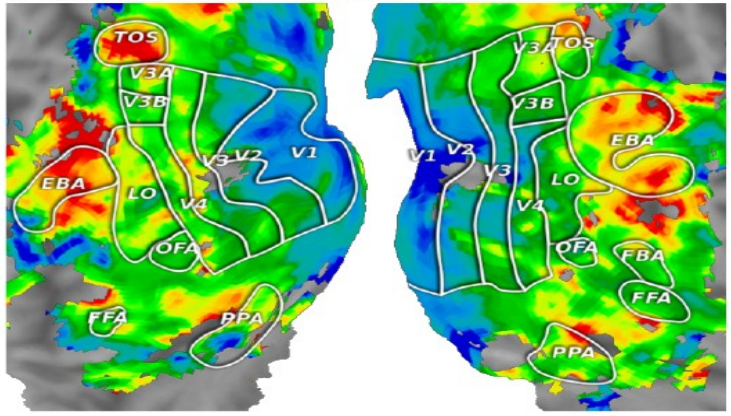
Figure 1 HCNNs as models of sensory cortex. (**a**) The basic framework in which sensory cortex is studied is one of encoding—the process by which stimuli are transformed into patterns of neural activity—and decoding, the process by which neural activity generates behavior. HCNNs have been used to make models of the encoding step; that is, they describe

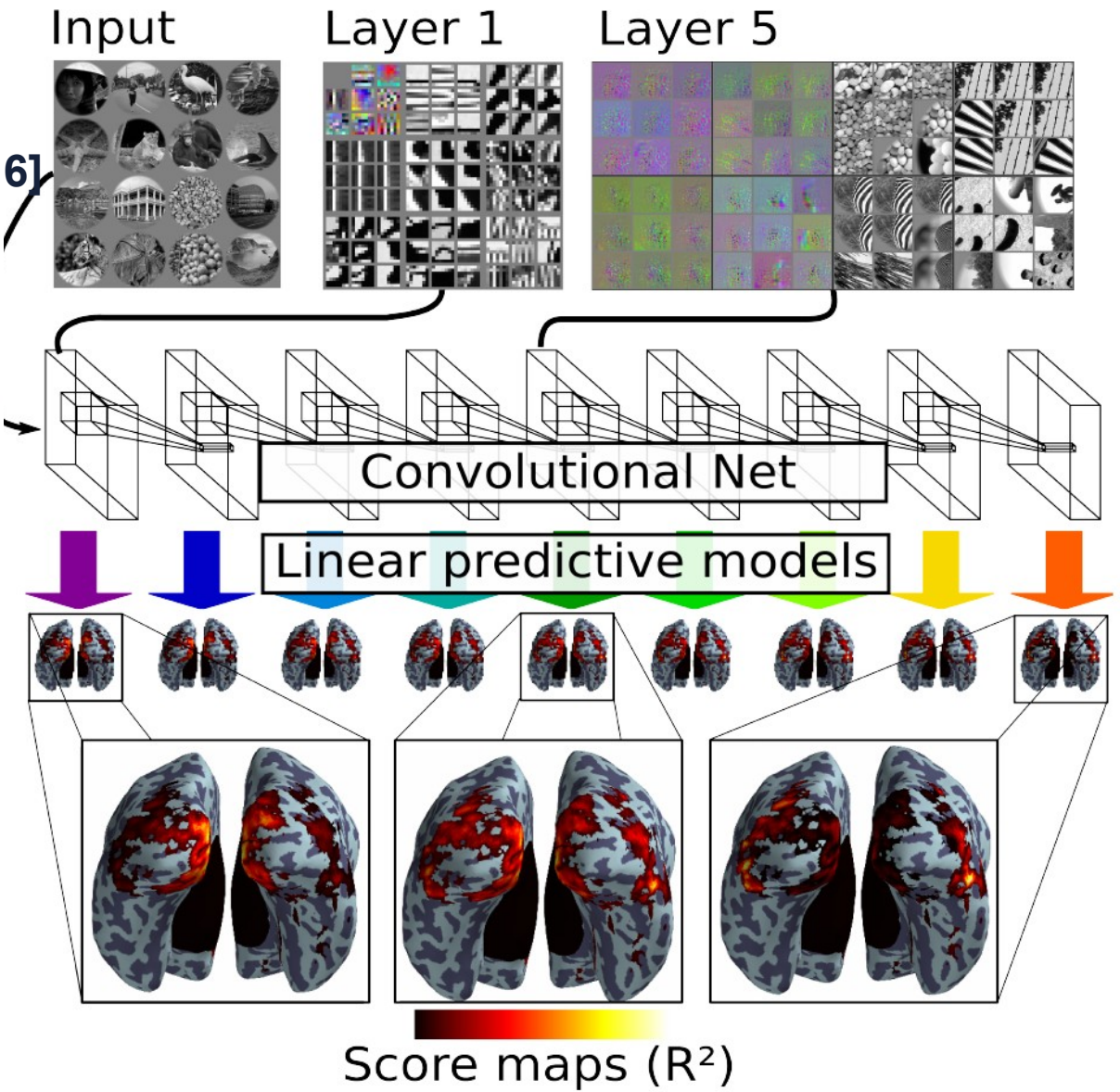
ConvNet models & fMRI

- **[Eickenberg et al. NeuroImage 2016]**



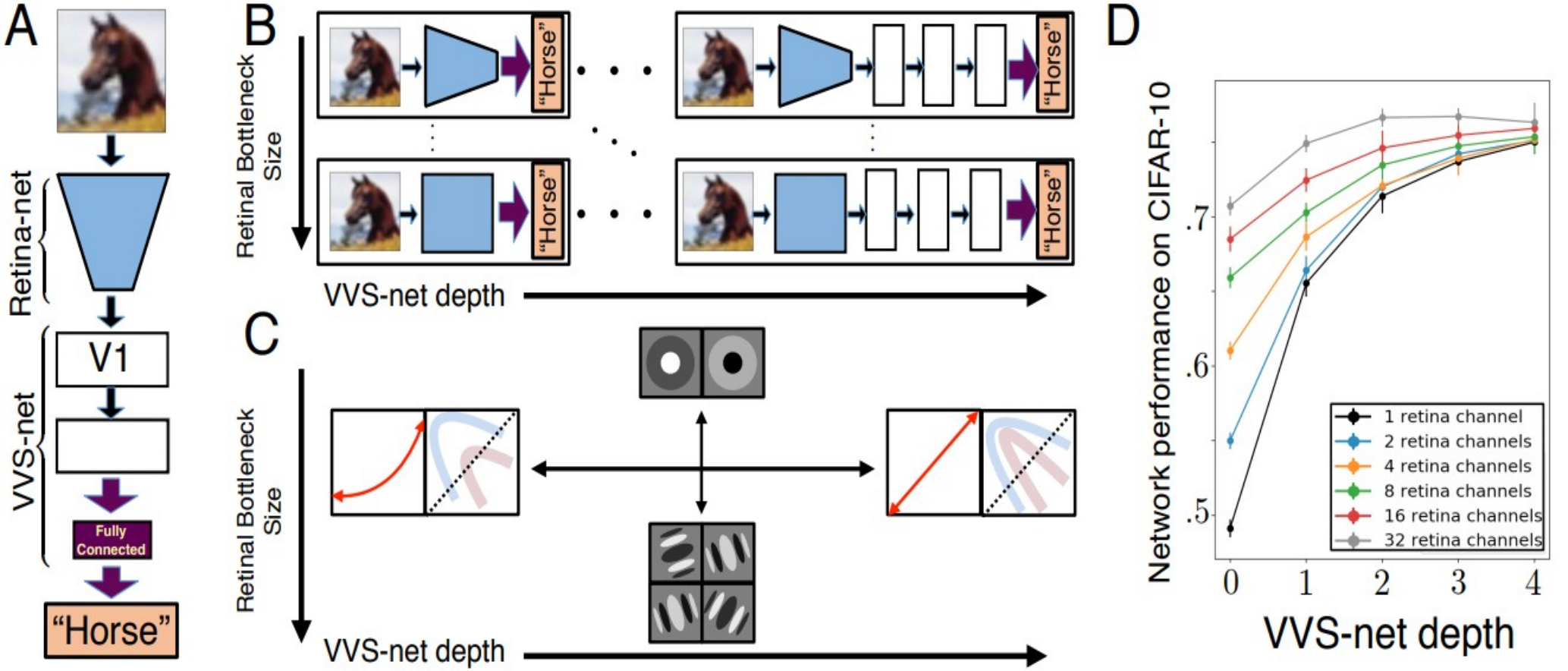
B Fingerprint summaries for Huth2012





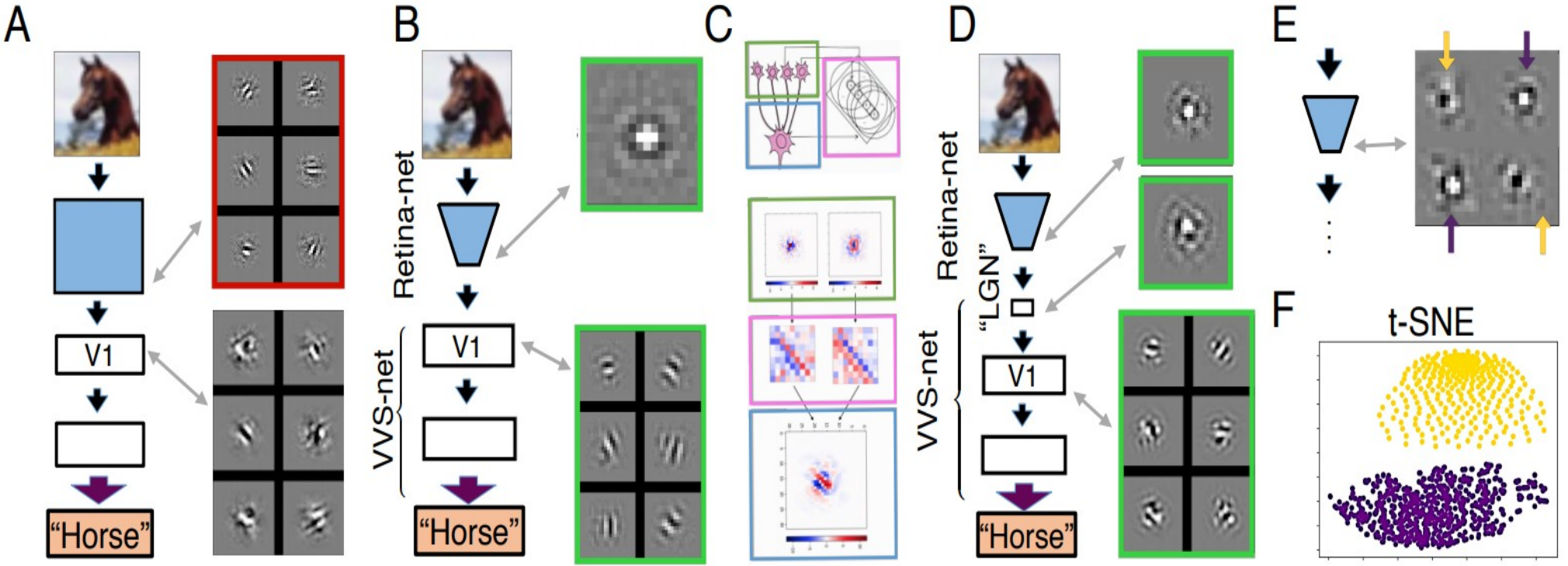
Anatomically Correct ConvNet?

- **[Lindsay et al. ICLR 2019]**



Anatomically Correct ConvNet?

- [Lindsay et al. ICLR 2019]
- Center-surround emerges from "Retina-net" only if there is a bottleneck (few feature maps).



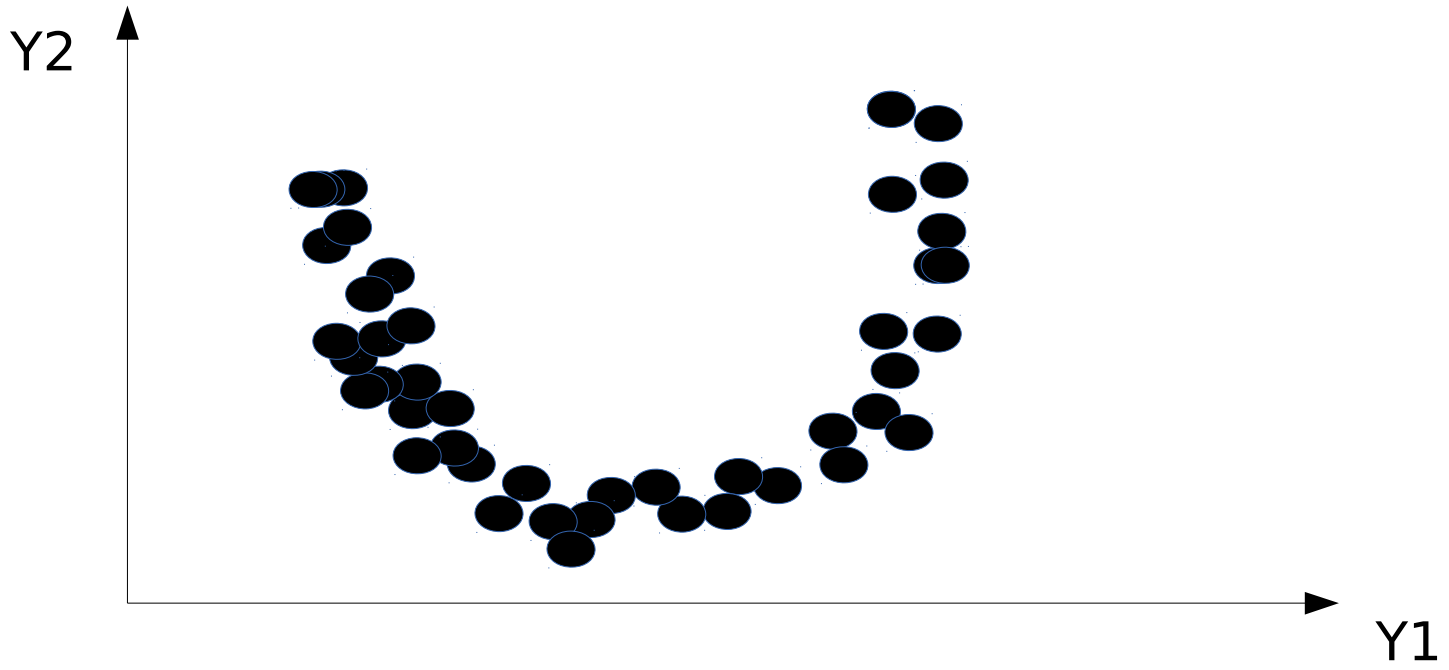
Unsupervised Feature Learning

Auto-Encoders and the likes

facebook
Artificial Intelligence Research

Energy-Based Unsupervised Learning

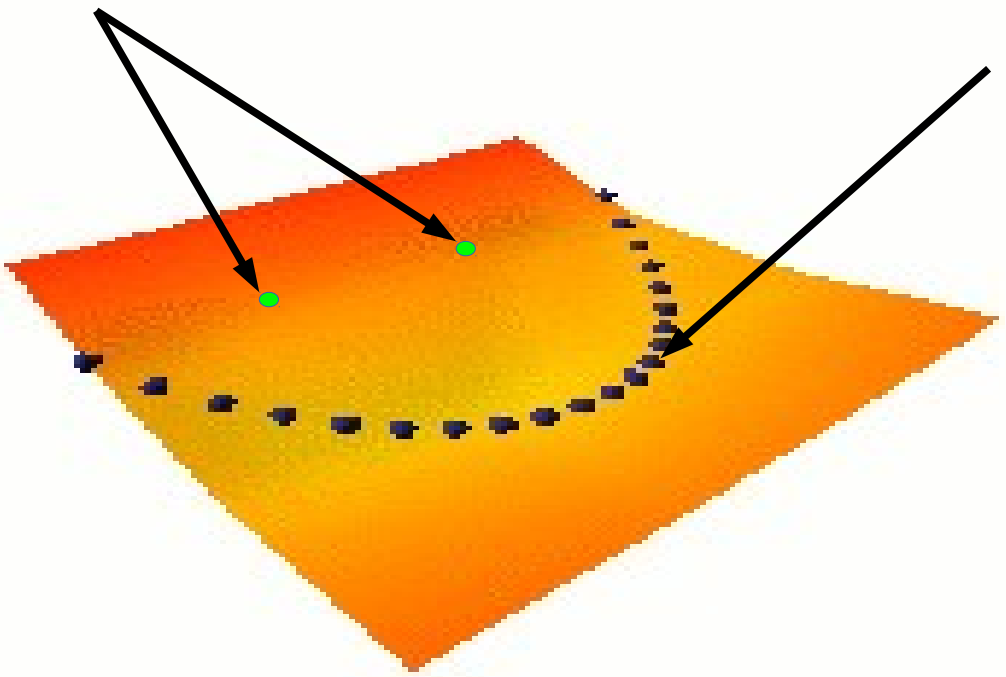
- **Learning an energy function (or contrast function) that takes**
  - Low values on the data manifold
  - Higher values everywhere else



Energy-Based Unsupervised Learning

- **Energy Function:** Takes low value on data manifold, higher values everywhere else
- Push down on the energy of desired outputs. Push up on everything else.
- But how do we choose where to push up?

Implausible futures (high energy)



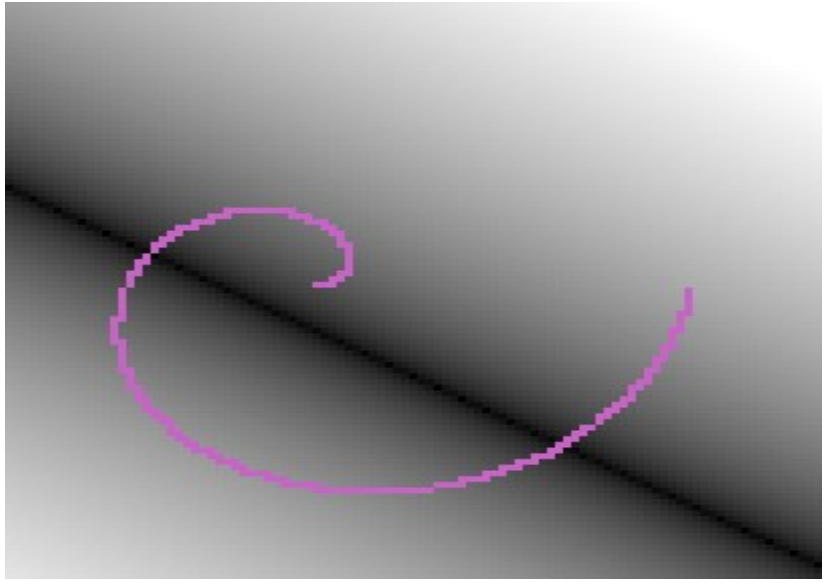
Plausible futures (low energy)

Energy surface for PCA and K-means

- **1. build the machine so that the volume of low energy stuff is constant**
  - PCA, K-means, GMM, square ICA...

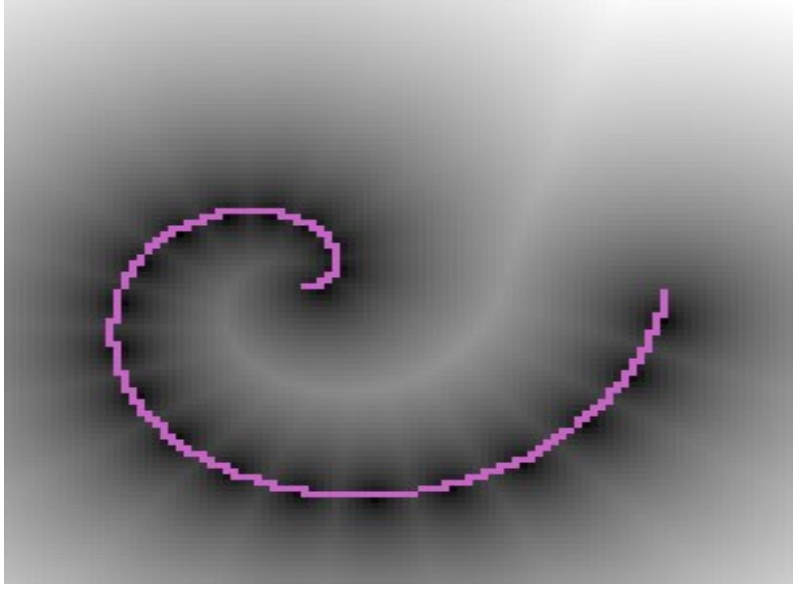
PCA

$$E(Y) = ||W^T WY - Y||^2$$



K-Means,
Z constrained to 1-of-K code

$$E(Y) = min_z \sum_i ||Y - W_i Z_i||^2$$



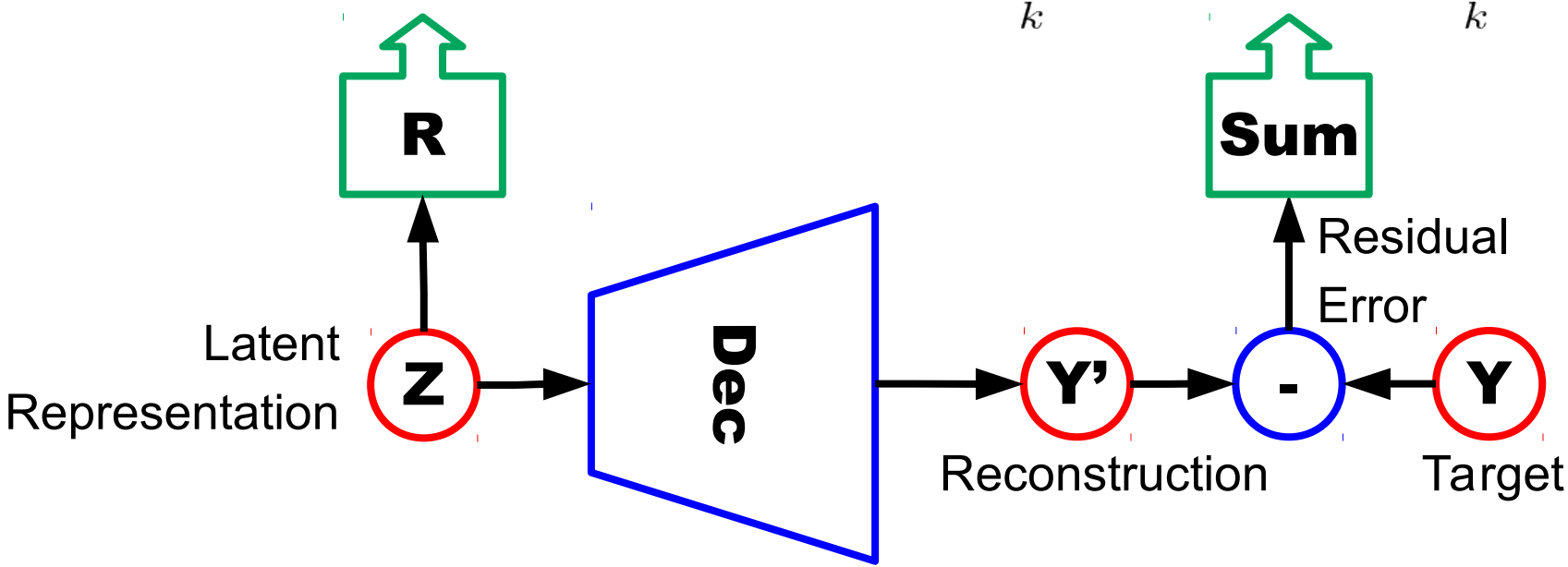
Seven Strategies to Shape the Energy Function

- **1. build the machine so that the volume of low energy stuff is constant**
  - PCA, K-means, GMM, square ICA
- **2. push down of the energy of data points, push up everywhere else**
  - Max likelihood (needs tractable partition function or variational approximation)
- **3. push down of the energy of data points, push up on chosen locations**
  - Contrastive divergence, Ratio Matching, Noise Contrastive Estimation, Min Probability Flow
- **4. minimize the gradient and maximize the curvature around data points**
  - score matching
- **5. train a dynamical system so that the dynamics goes to the manifold**
  - denoising auto-encoder
- **6. use a regularizer that limits the volume of space that has low energy**
  - Sparse coding, sparse auto-encoder, PSD
- **7. if E(Y) = ||Y - G(Y)||^2, make G(Y) as "constant" as possible.**
  - Contracting auto-encoder, saturating auto-encoder

The “Decoder with Regularized Latent Variable” Model

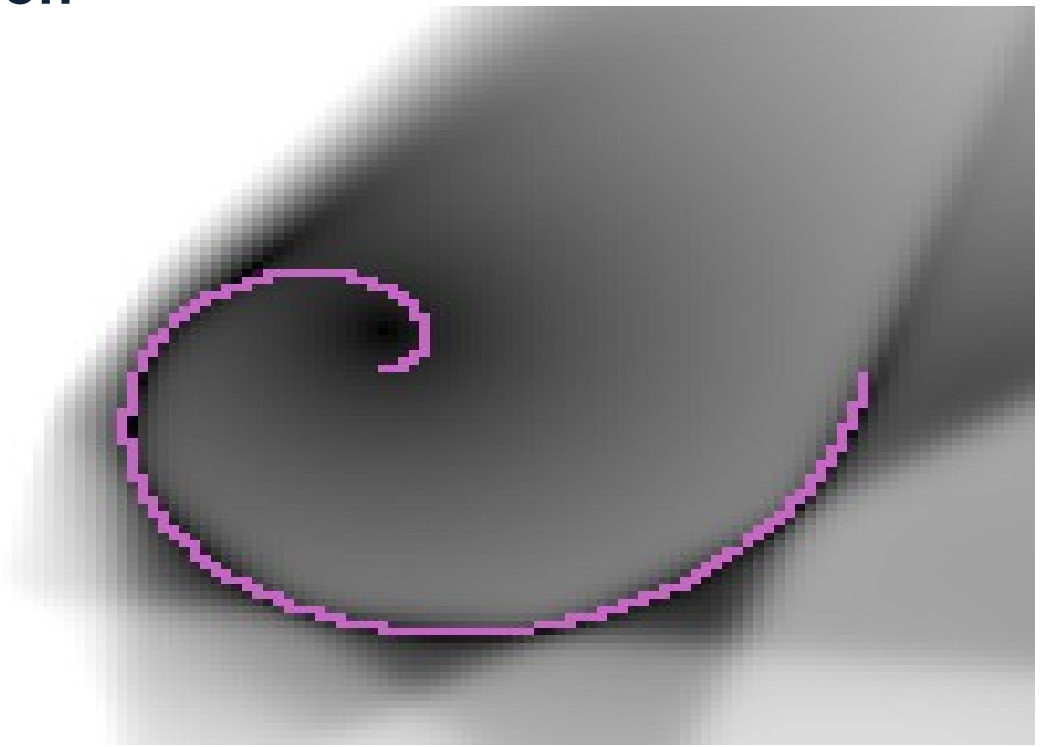
- **Y’ = Dec(Z) Z* = argmin || Y – Dec(Z) || + R(Z)**
- **Linear decoder: K-Means, basis pursuit, K-SVD, sparse coding,….**
- **Sparse modeling: [Olshausen Field 1997]**

$$E(Y,Z) = ||Y - \sum_k W_k Z_k||^2 + \alpha \sum_k |Z_k|$$



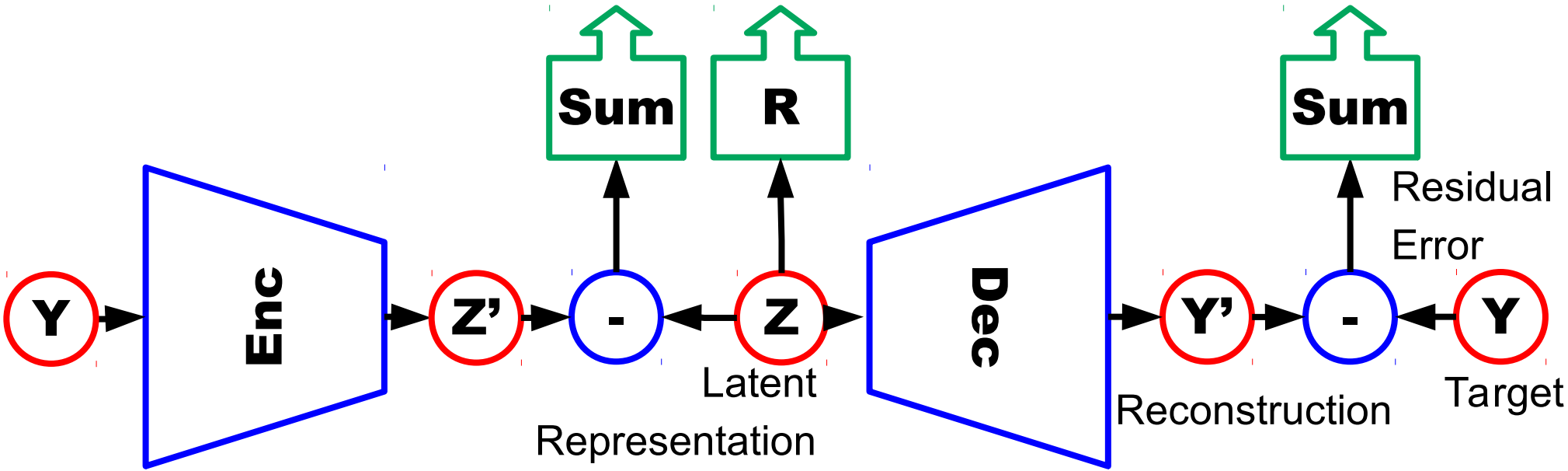
Energy Surface for Sparse Modeling

- **Sparse coding, sparse auto-encoder, Predictive Sparse Decomposition**



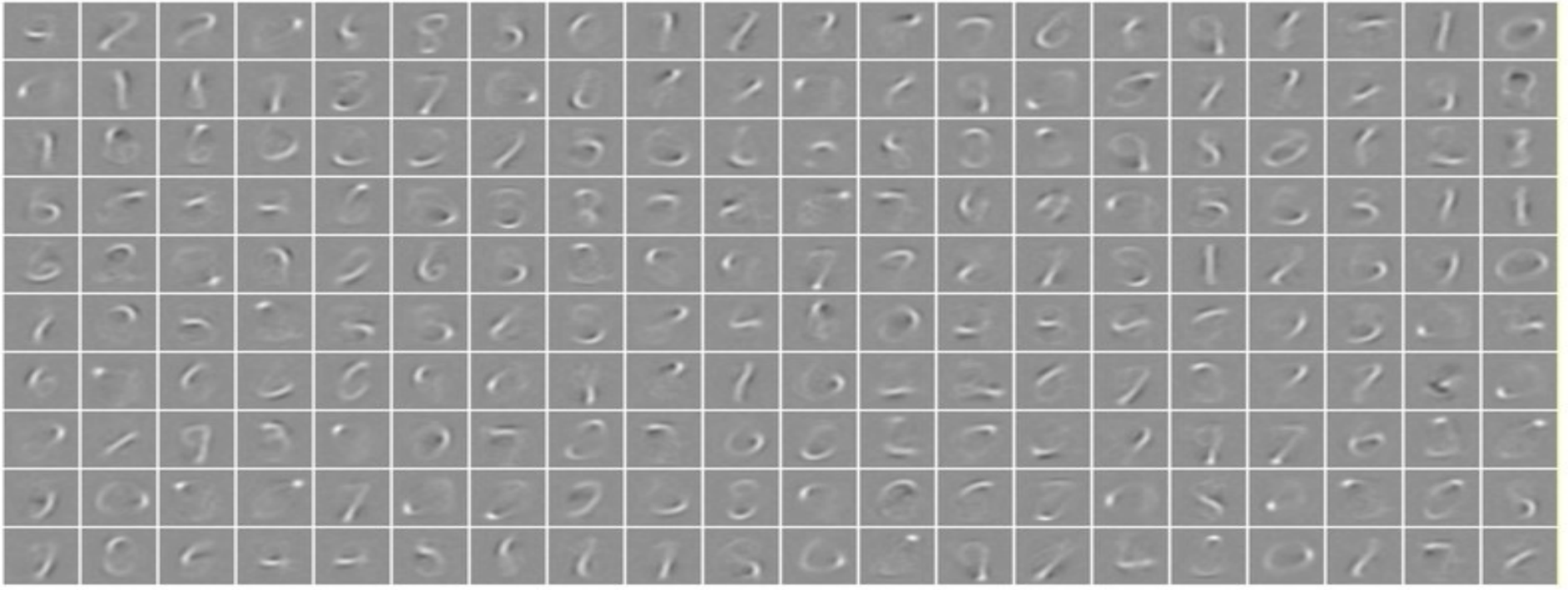
The “Encoder-Decoder with latent vars” Model

- **Z* = argmin || Y – Dec(Z) || + R(Z) + || Z – Enc(Y) ||**
- **Linear decoder: Predictive Sparse Decomposition**
  - [Kavuckcuoglu 2008 (arxiv:1010.3467)] [Kavukcuoglu CVPR 2009]



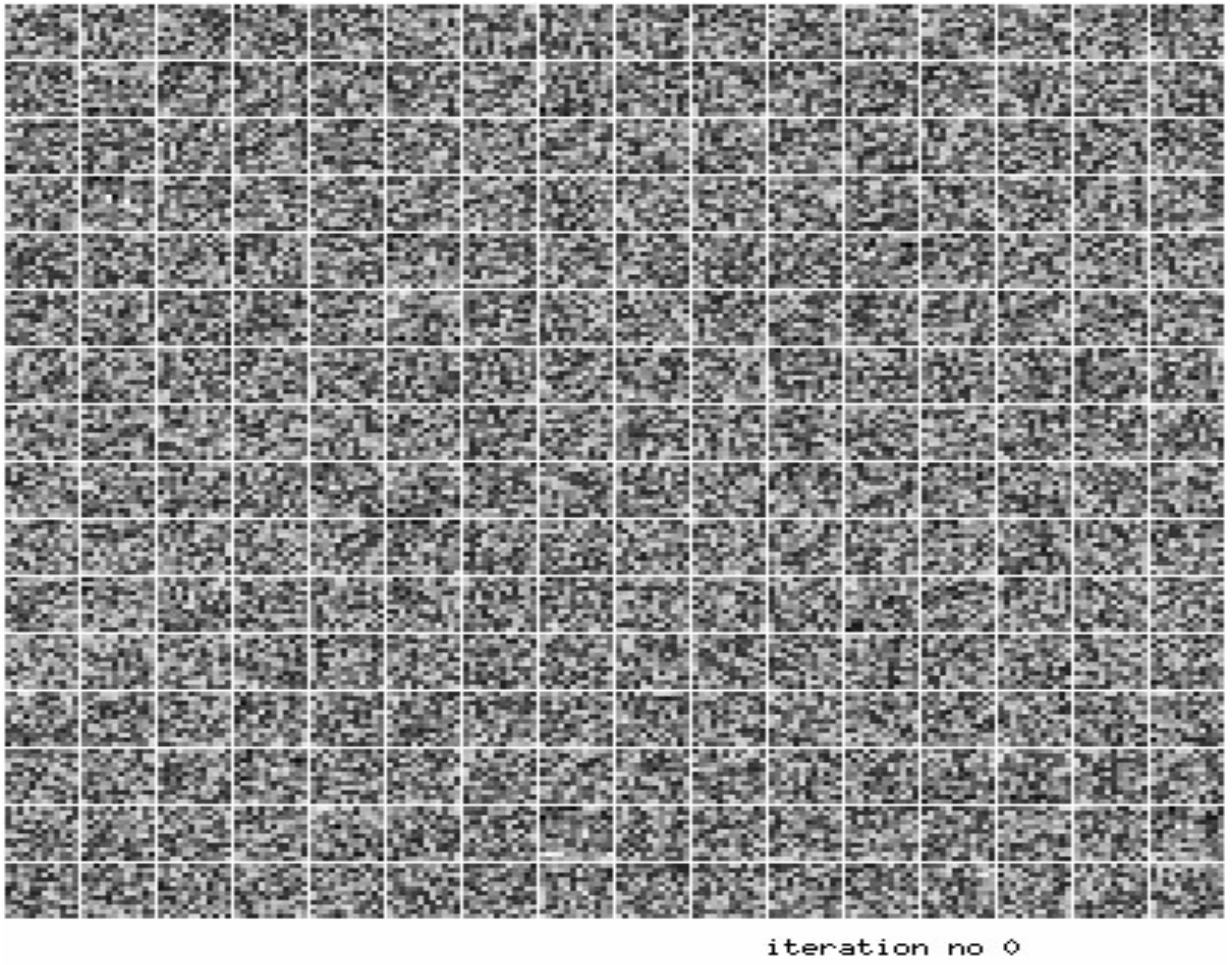
PSD: Basis Functions on MNIST

- **Basis functions (and encoder matrix) are digit parts**

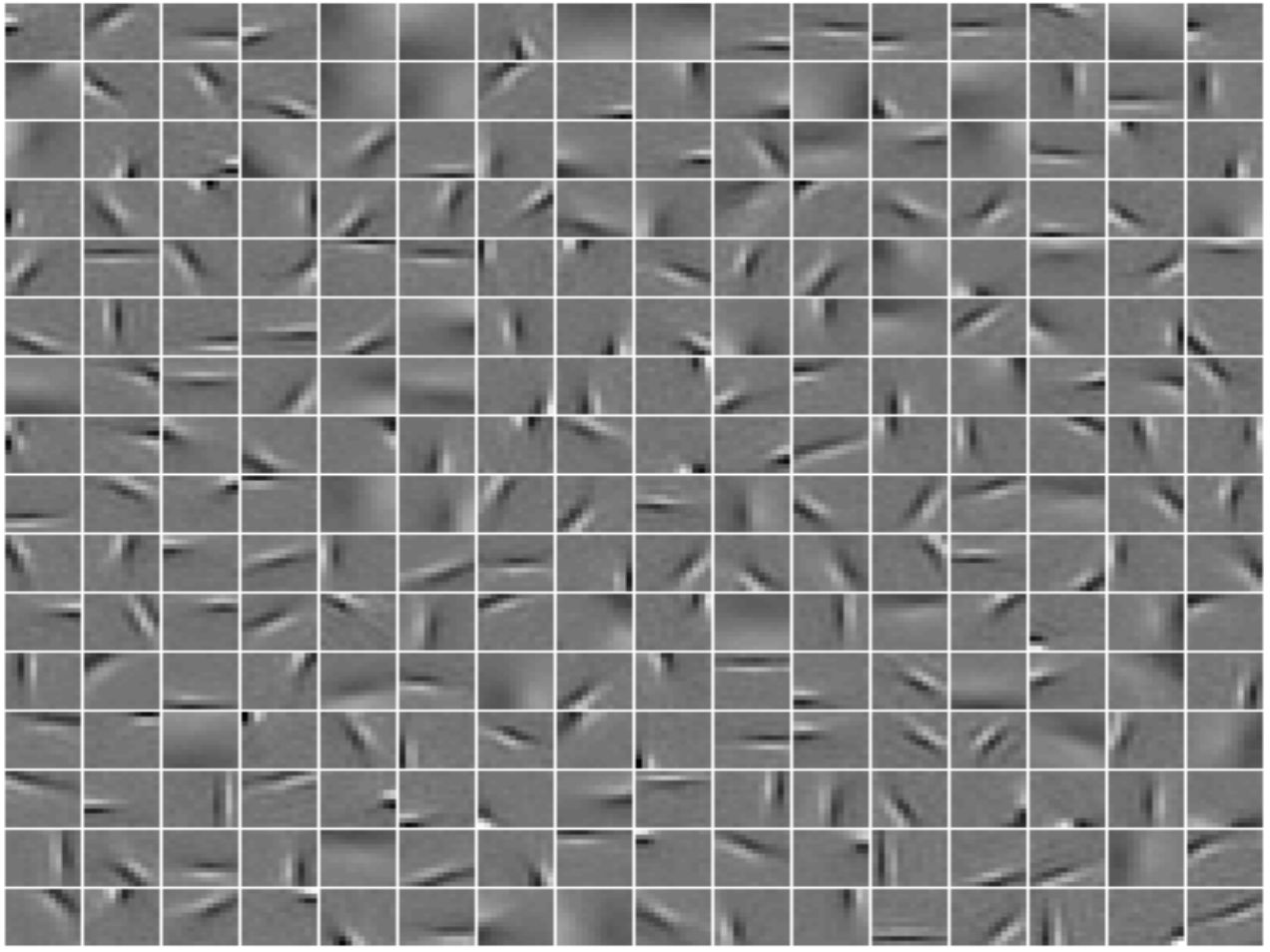


Predictive Sparse Decomposition (PSD): Training

- **Training on natural images patches.**
  - 12X12
  - 256 basis functions

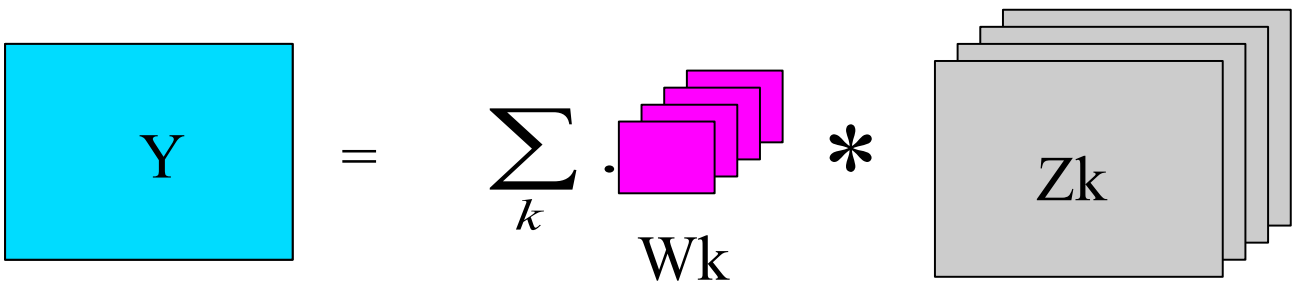


Learned Features on natural patches: V1-like receptive fields



Convolutional Sparse Coding

- **Replace the dot products with dictionary element by convolutions.**
  - Input Y is a full image
  - Each code component Zk is a feature map (an image)
  - Each dictionary element is a convolution kernel
- **Regular sparse coding** $E(Y,Z) = ||Y - \sum_k W_k Z_k||^2 + \alpha \sum_k |Z_k|$
- **Convolutional S.C.** $E(Y,Z) = ||Y - \sum_k W_k * Z_k||^2 + \alpha \sum_k |Z_k|$

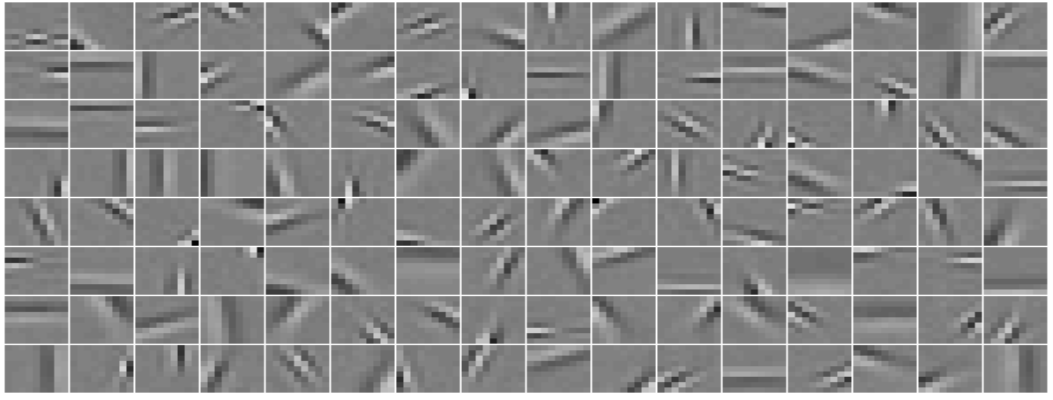


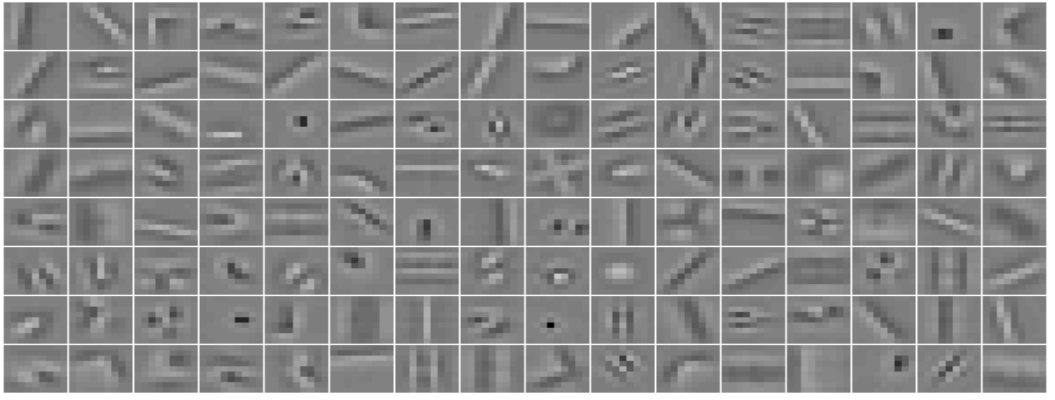
Also used in “deconvolutional networks” [Zeiler, Taylor, Fergus CVPR 2010]

Convolutional PSD: Encoder with a soft sh() Function

- **Convolutional Formulation**
  - Extend sparse coding from ***PATCH*** to ***IMAGE***

$$\mathcal{L}(x, z, \mathcal{D}) = \frac{1}{2}||x - \sum_{k=1}^{K} \mathcal{D}_k * z_k||_2^2 + \sum_{k=1}^{K} ||z_k - f(W^k * x)||_2^2 + |z|_1$$

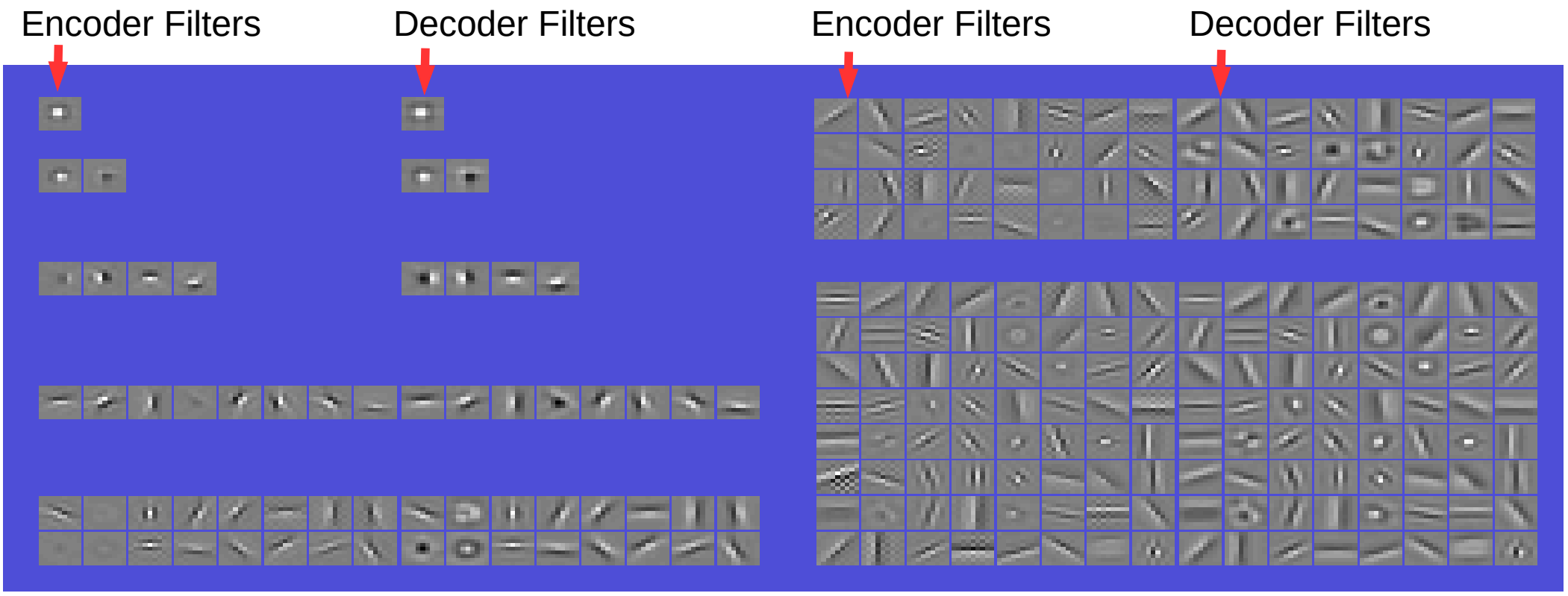




- ***PATCH*** based learning
- ***CONVOLUTIONAL*** learning

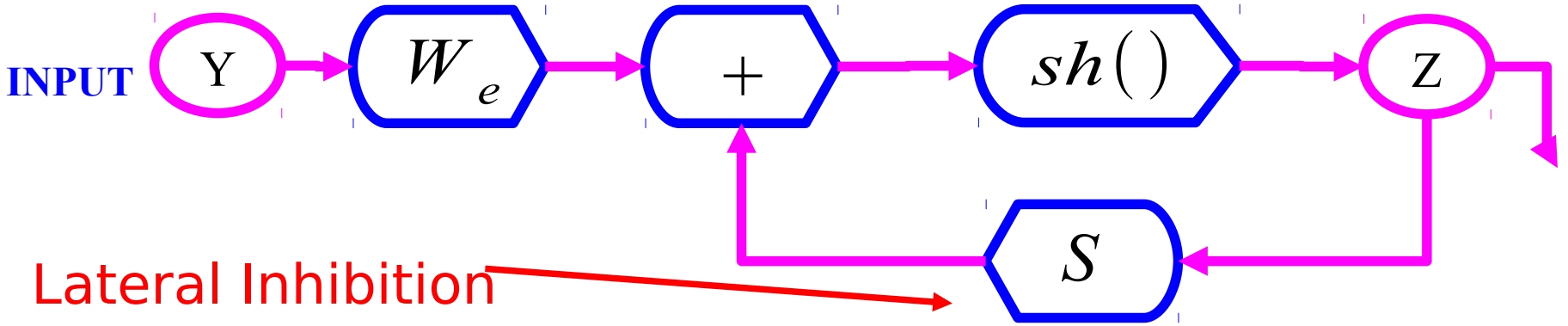
Convolutional Sparse Auto-Encoder on Natural Images

- **Filters and Basis Functions obtained**
  - with 1, 2, 4, 8, 16, 32, and 64 filters.



Better Idea: Give the "right" structure to the encoder

- **ISTA/FISTA: iterative algorithm that converges to optimal sparse code**



$$Z(t+1) = \text{Shrinkage}_{\lambda/L}\left[Z(t) - \frac{1}{L}W_d^T(W_d Z(t) - Y)\right]$$

- **ISTA/FISTA reparameterized:**

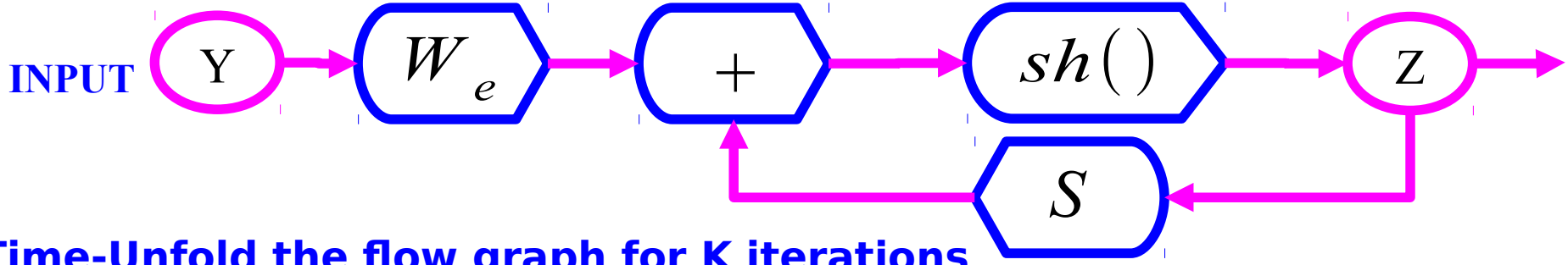
$$Z(t+1) = \text{Shrinkage}_{\lambda/L}\left[W_e^T Y + S Z(t)\right]; \quad W_e = \frac{1}{L}W_d; \quad S = I - \frac{1}{L}W_d^T W_d$$

- **LISTA (Learned ISTA): learn the We and S matrices to get fast solutions**

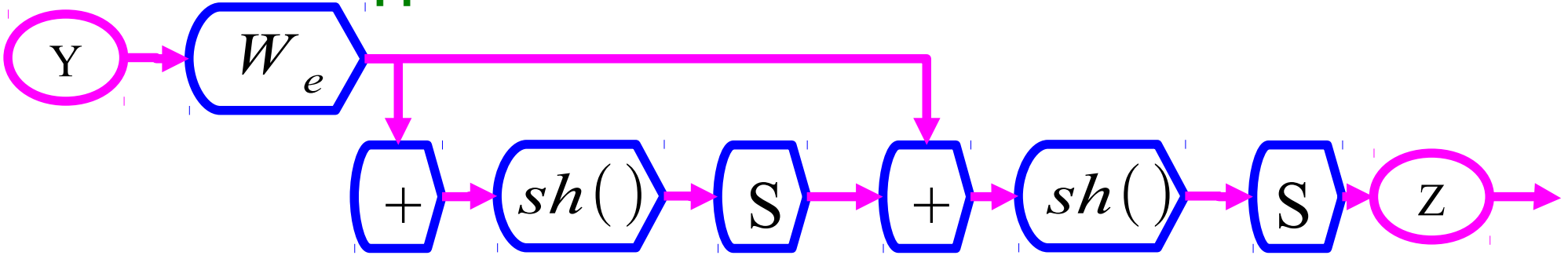
[Gregor & LeCun, ICML 2010], [Bronstein et al. ICML 2012], [Rolfe & LeCun ICLR 2013]

LISTA: Train We and S matrices to give a good approximation quickly

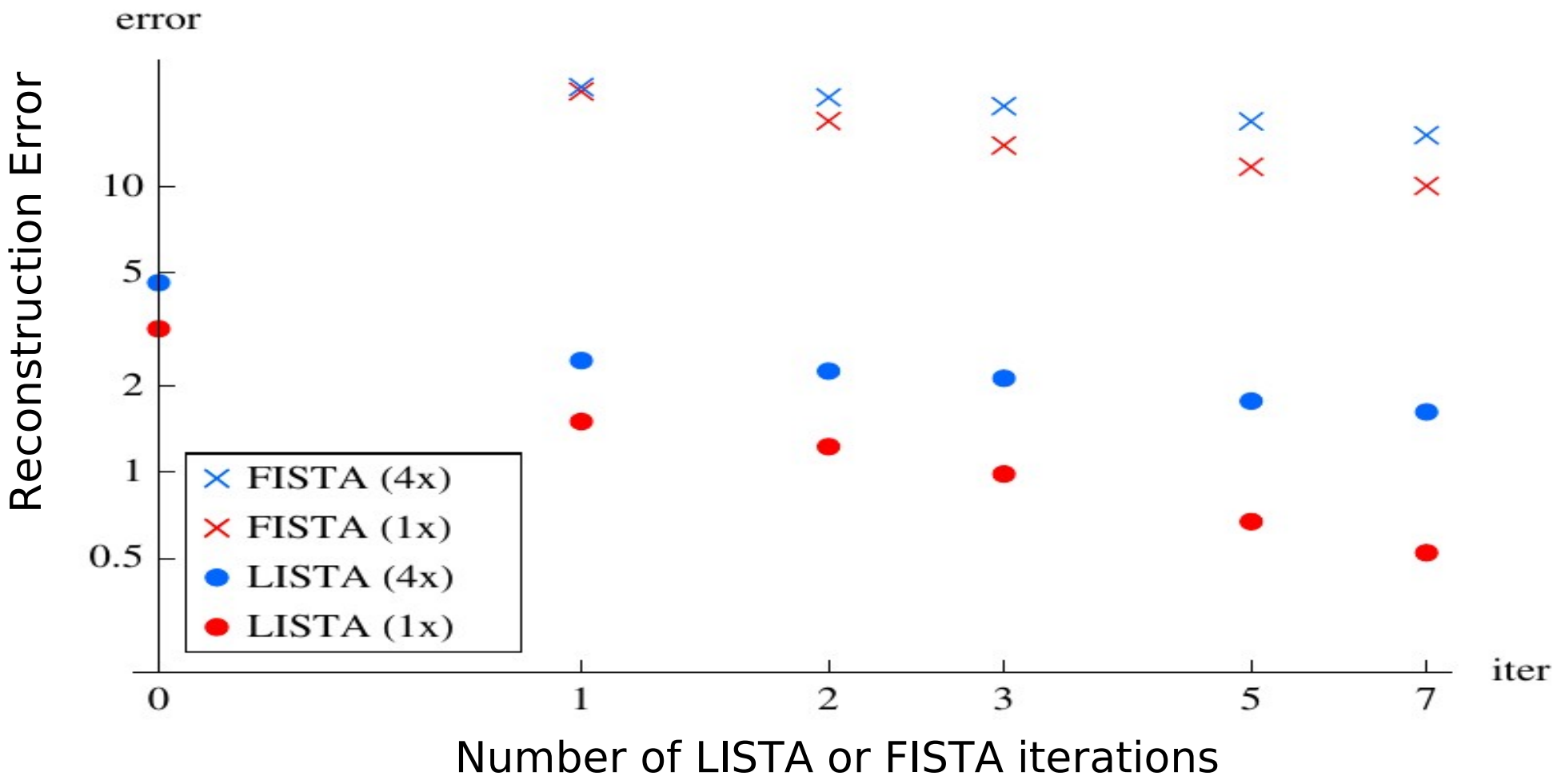
- **Think of the FISTA flow graph as a recurrent neural net where We and S are trainable parameters**



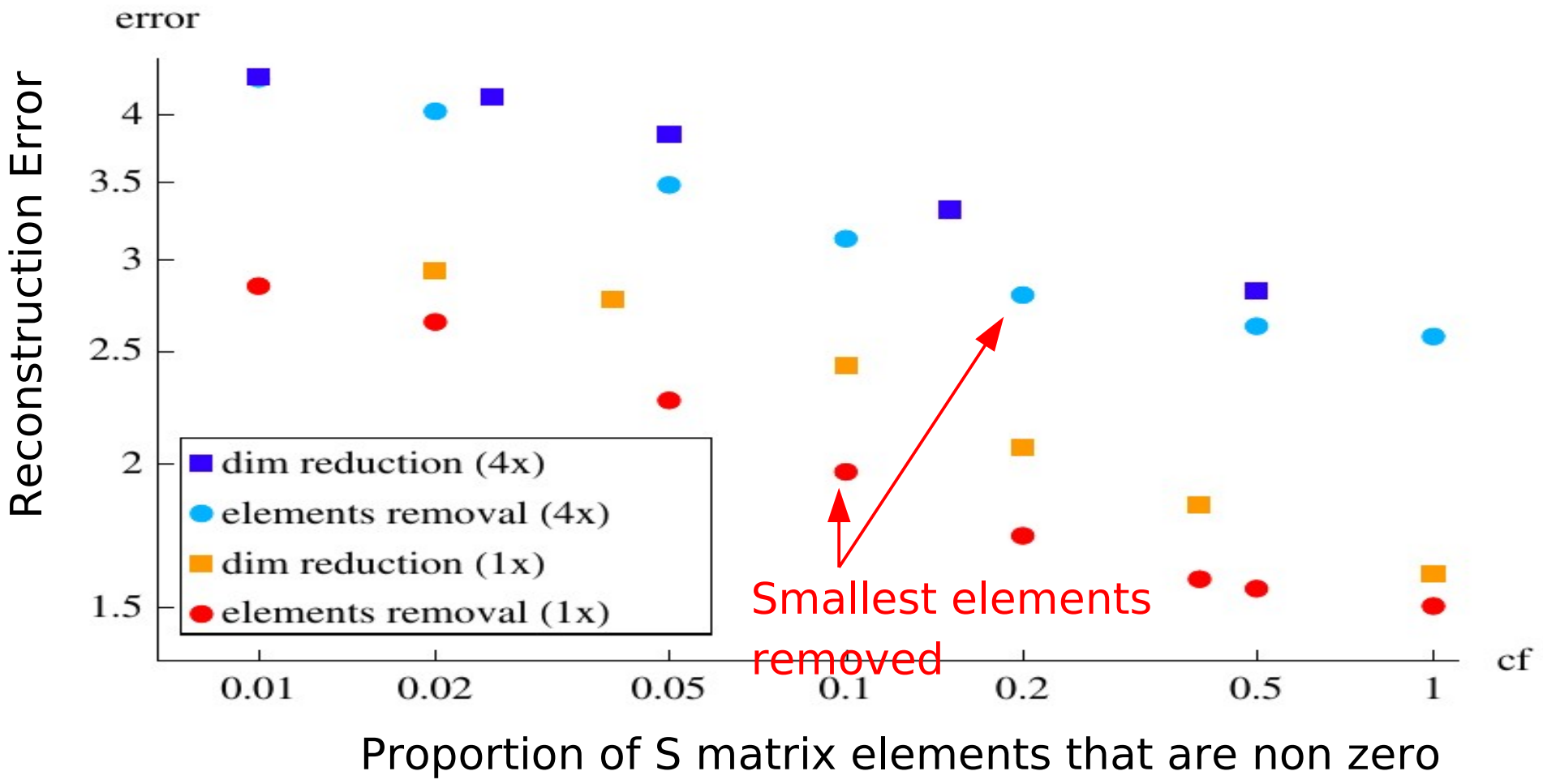
- **Time-Unfold the flow graph for K iterations**
- **Learn the We and S matrices with "backprop-through-time"**
- **Get the best approximate solution within K iterations**



Learning ISTA (LISTA) vs ISTA/FISTA

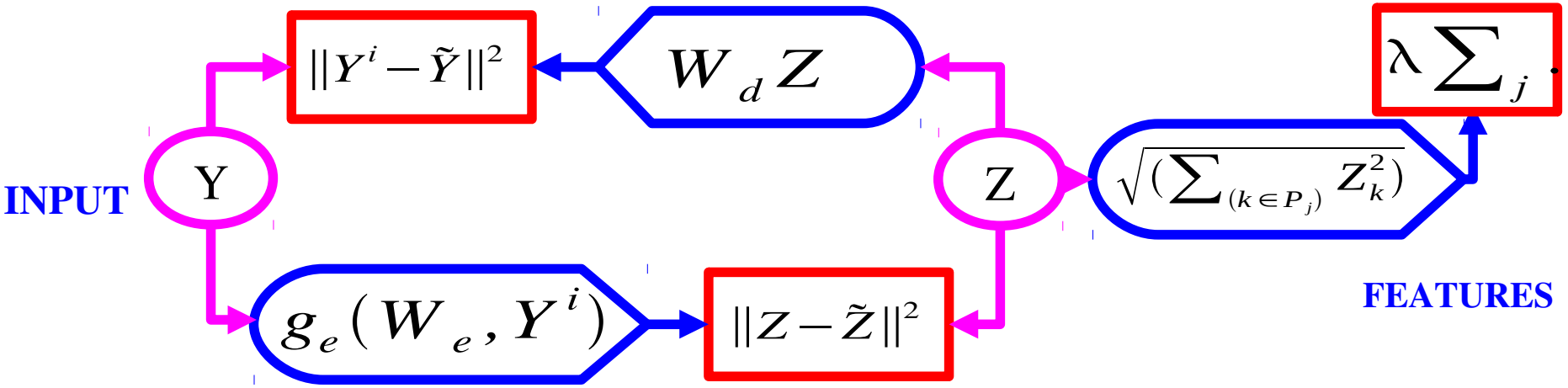


LISTA with partial mutual inhibition matrix



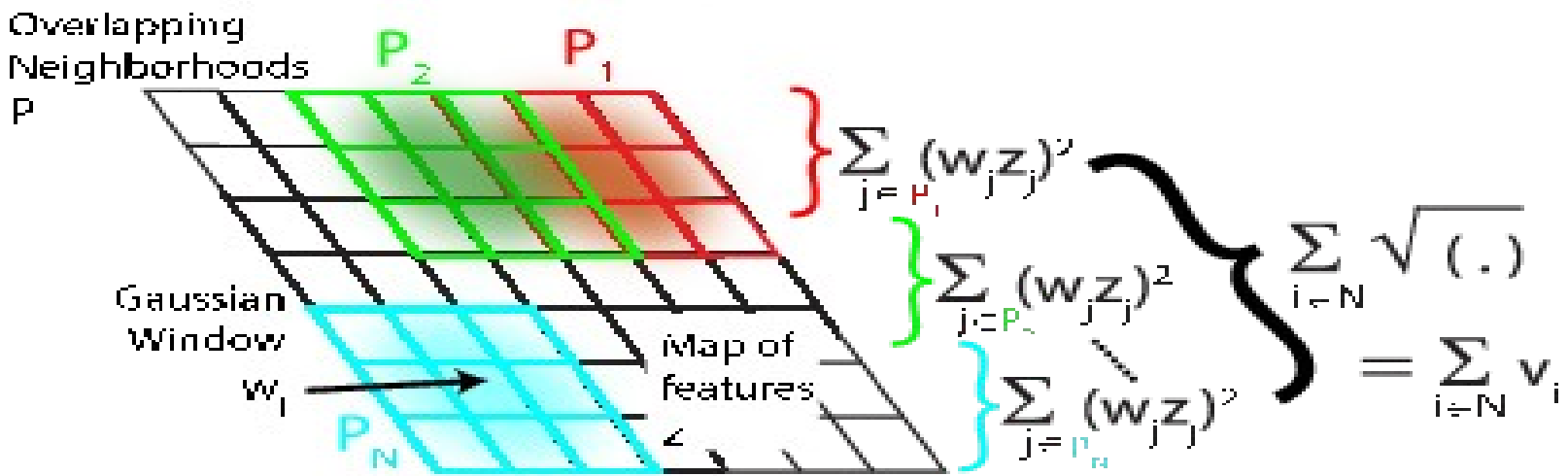
Learning Invariant Features [Kavukcuoglu et al. CVPR 2009]

- **Unsupervised PSD ignores the spatial pooling step.**
- **Could we devise a similar method that learns the pooling layer as well?**
- **Idea [Hyvarinen & Hoyer 2001]: group sparsity on pools of features**
  - Minimum number of pools must be non-zero
  - Number of features that are on within a pool doesn't matter
  - Polls tend to regroup similar features



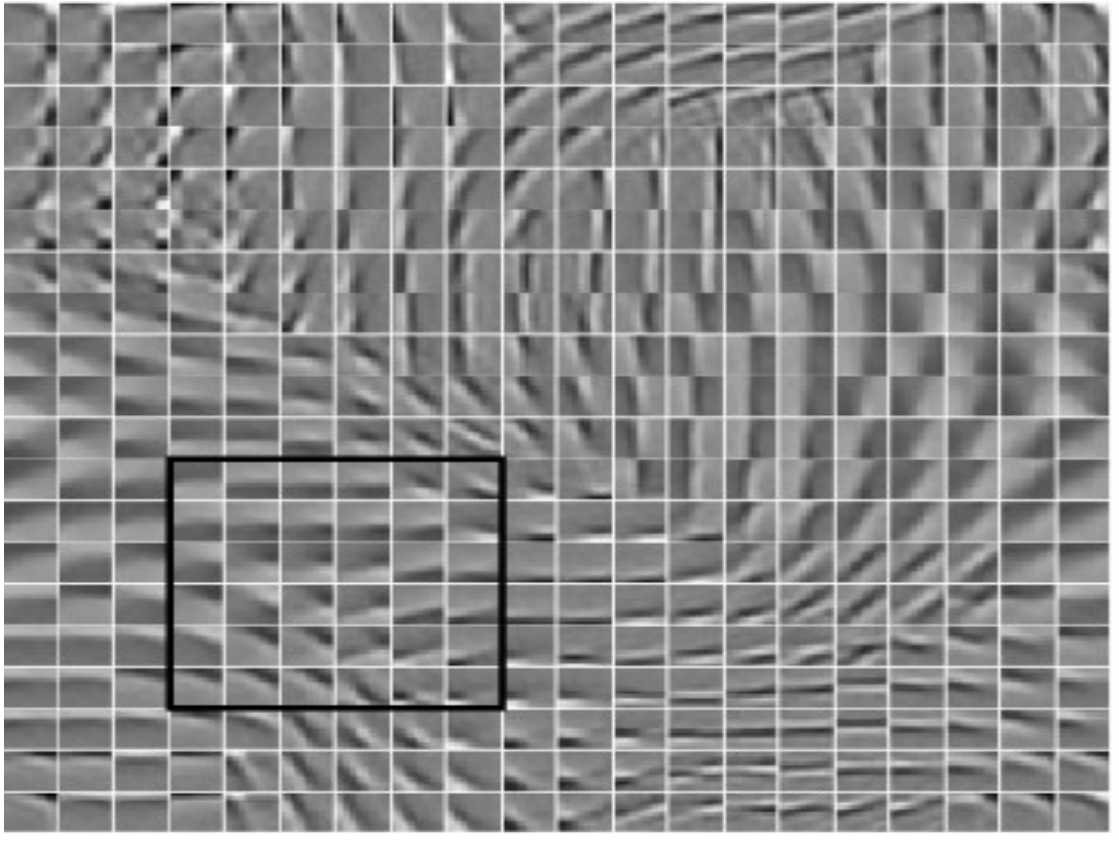
Why just pool over space? Why not over orientation?

- **Using an idea from Hyvarinen: topographic square pooling (subspace ICA)**
  - 1. Apply filters on a patch (with suitable non-linearity)
  - 2. Arrange filter outputs on a 2D plane
  - 3. square filter outputs
  - 4. minimize sqrt of sum of blocks of squared filter outputs



Why just pool over space? Why not over orientation?

- **The filters arrange themselves spontaneously so that similar filters enter the same pool.**
- **The pooling units can be seen as complex cells**
- **They are invariant to local transformations of the input**
  - For some it's translations, for others rotations, or other transformations.



Pinwheels!

- **Does that look pinwheely to you?**

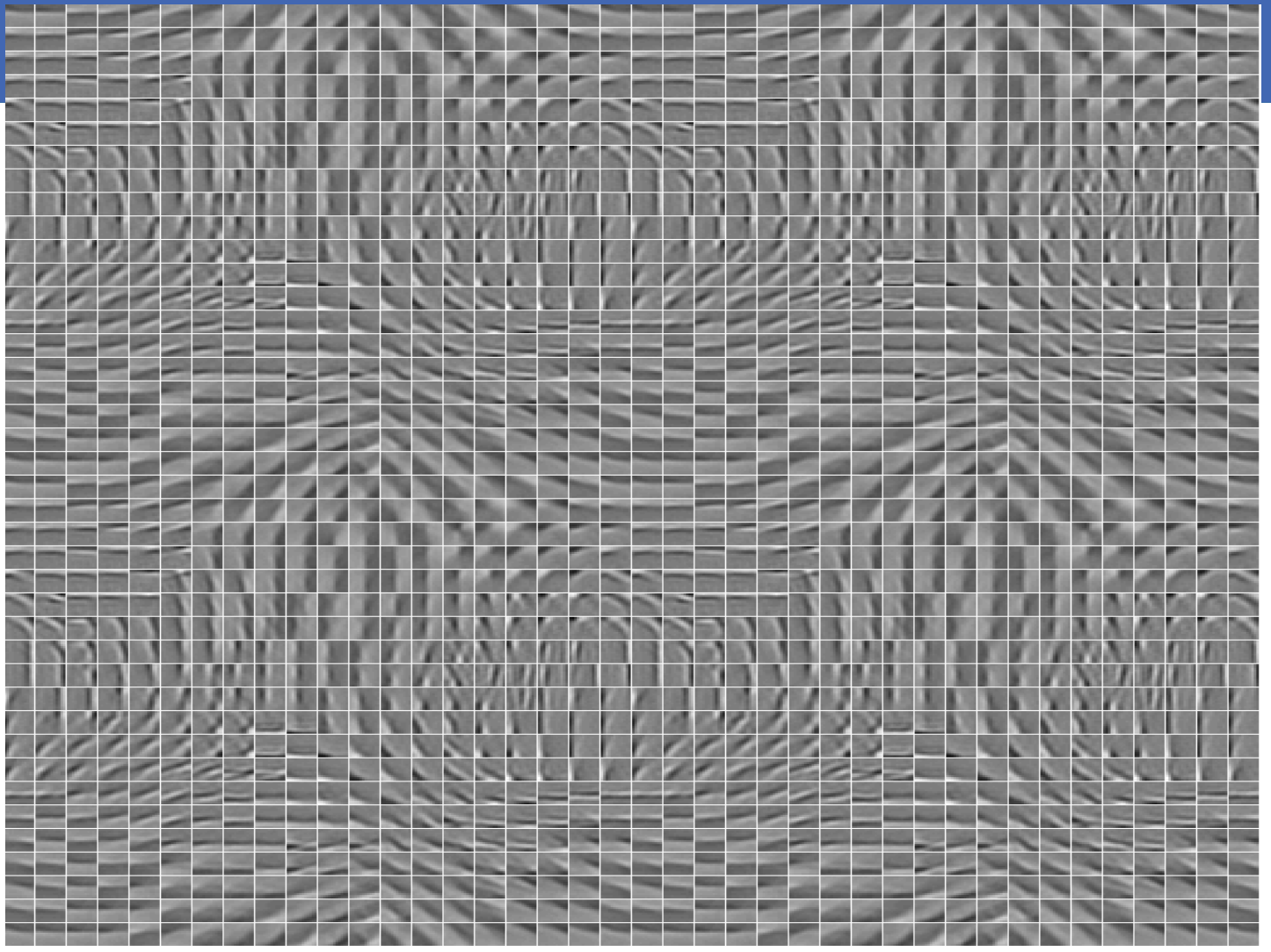


Image-level training, local filters but no weight sharing!

- **Training on 115x115 images. Kernels are 15x15 (not shared across space!)**
- **[Gregor & LeCun arXiv:1006.044]**
  - "Emergence of Complex-Like Cells in a Temporal Product Network with Local Receptive Fields"

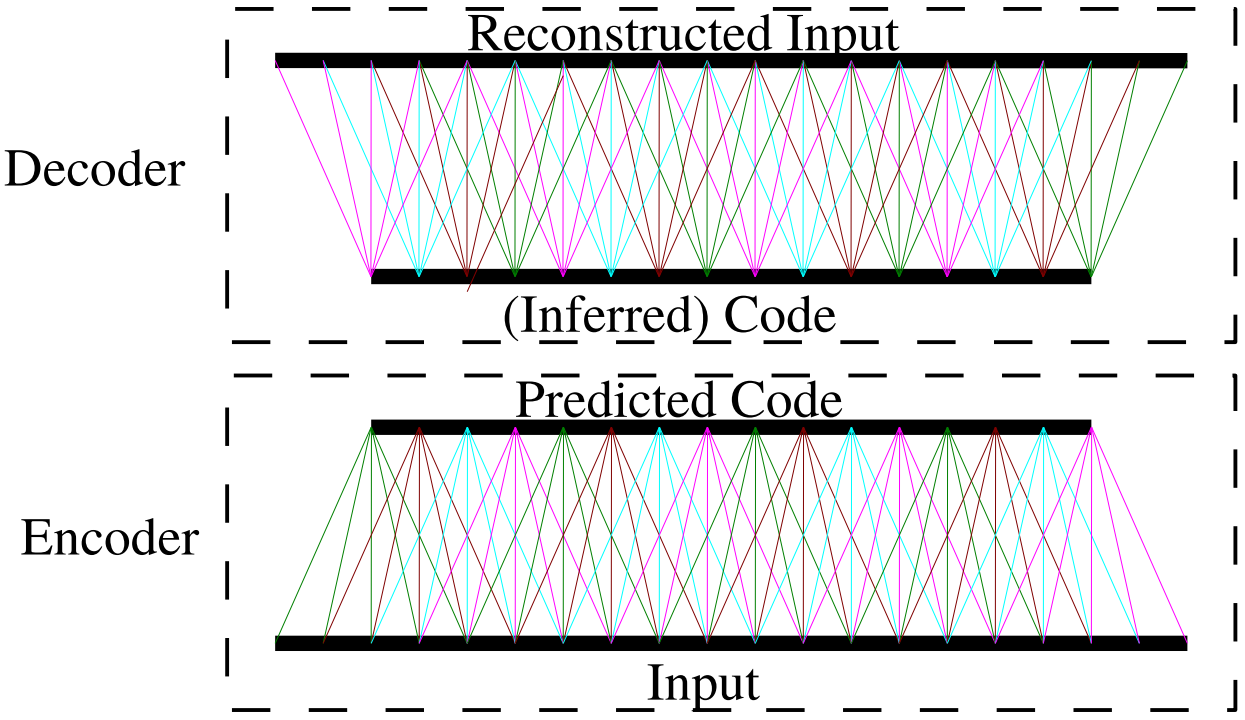


Image-level training, local filters but no weight sharing!

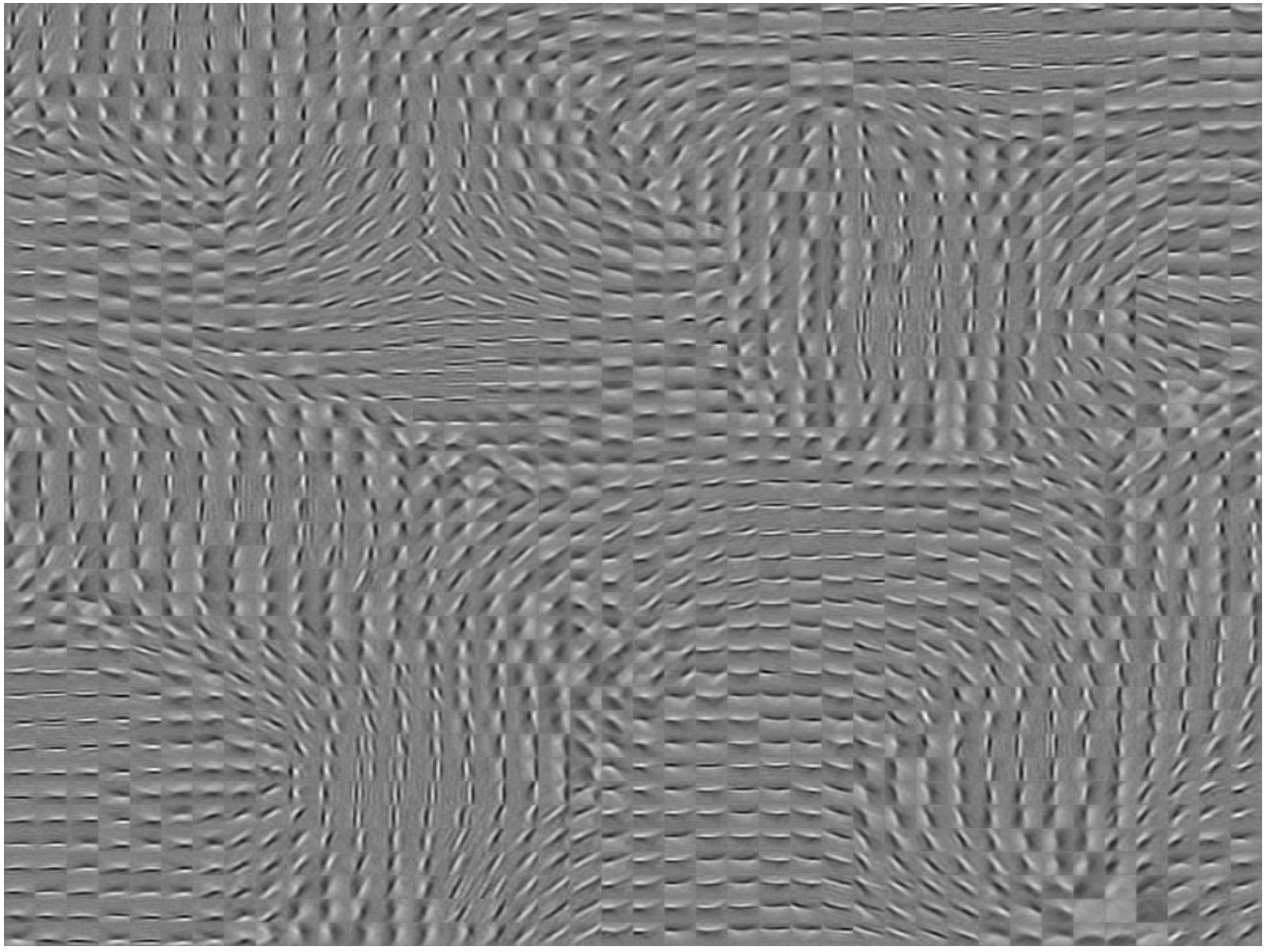
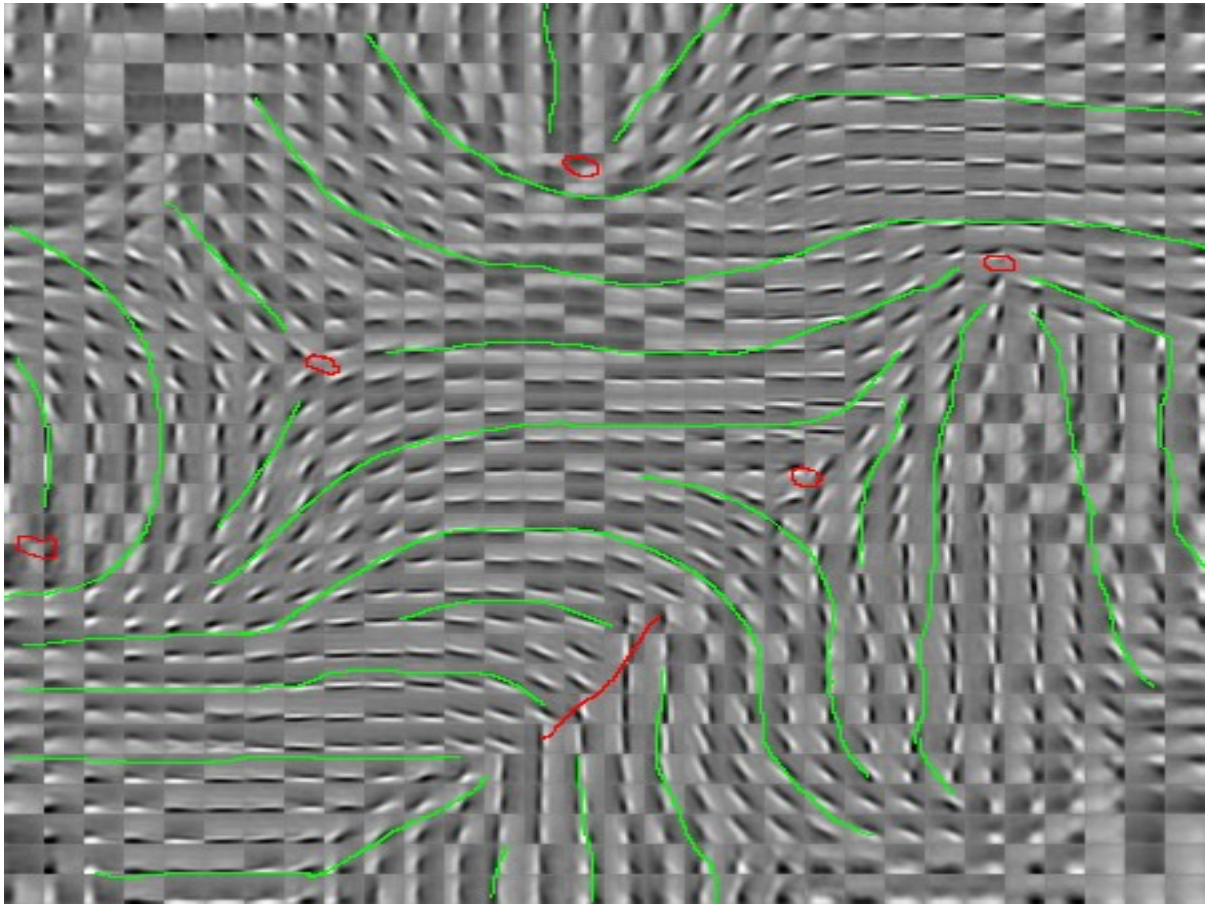


Image-level training, local filters but no weight sharing!

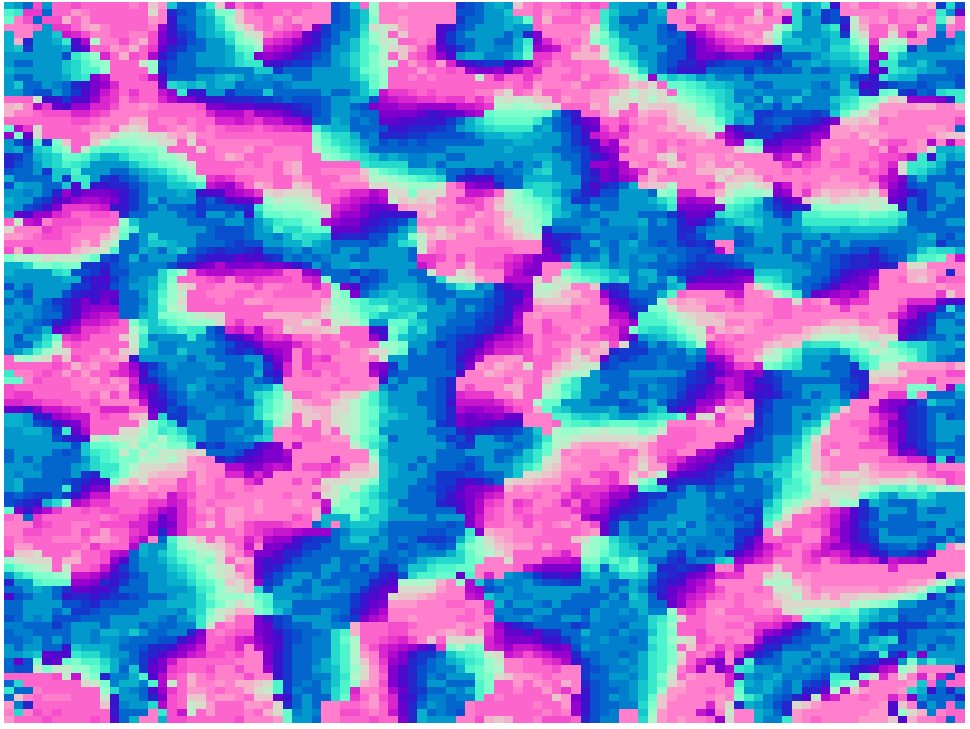
- **Training on 115x115 images. Kernels are 15x15 (not shared across space!)**



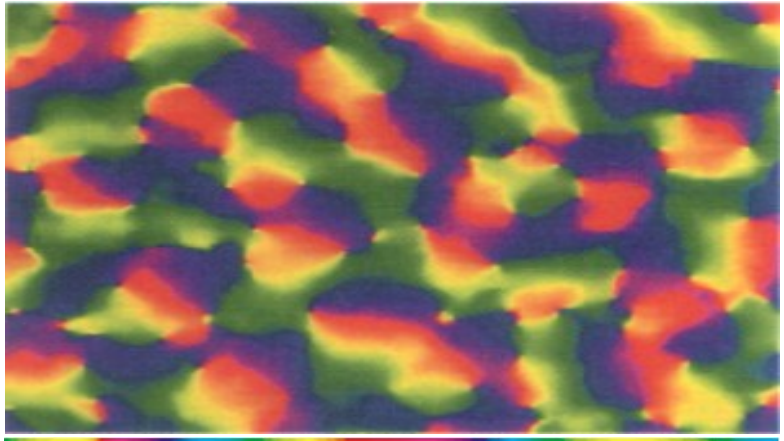
Orientation Selectivity Map

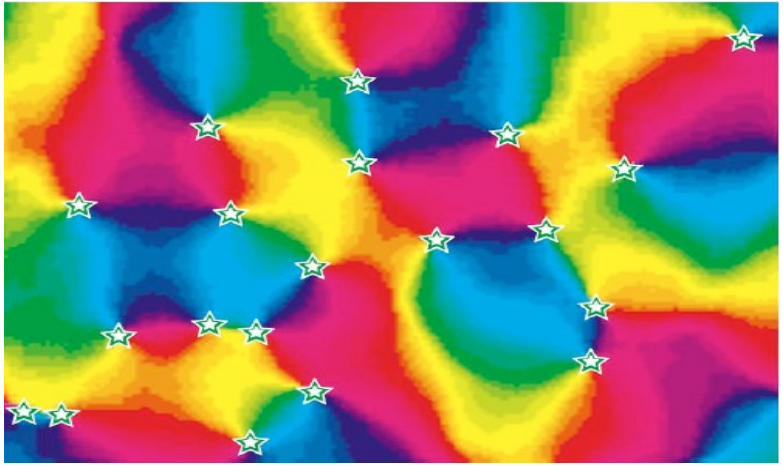
119x119 Image Input, 100x100 Code

20x20 Receptive field size, sigma=5



K Obermayer and GG Blasdel, Journal of Neuroscience, Vol 13, 4114-4129 (**Monkey**)

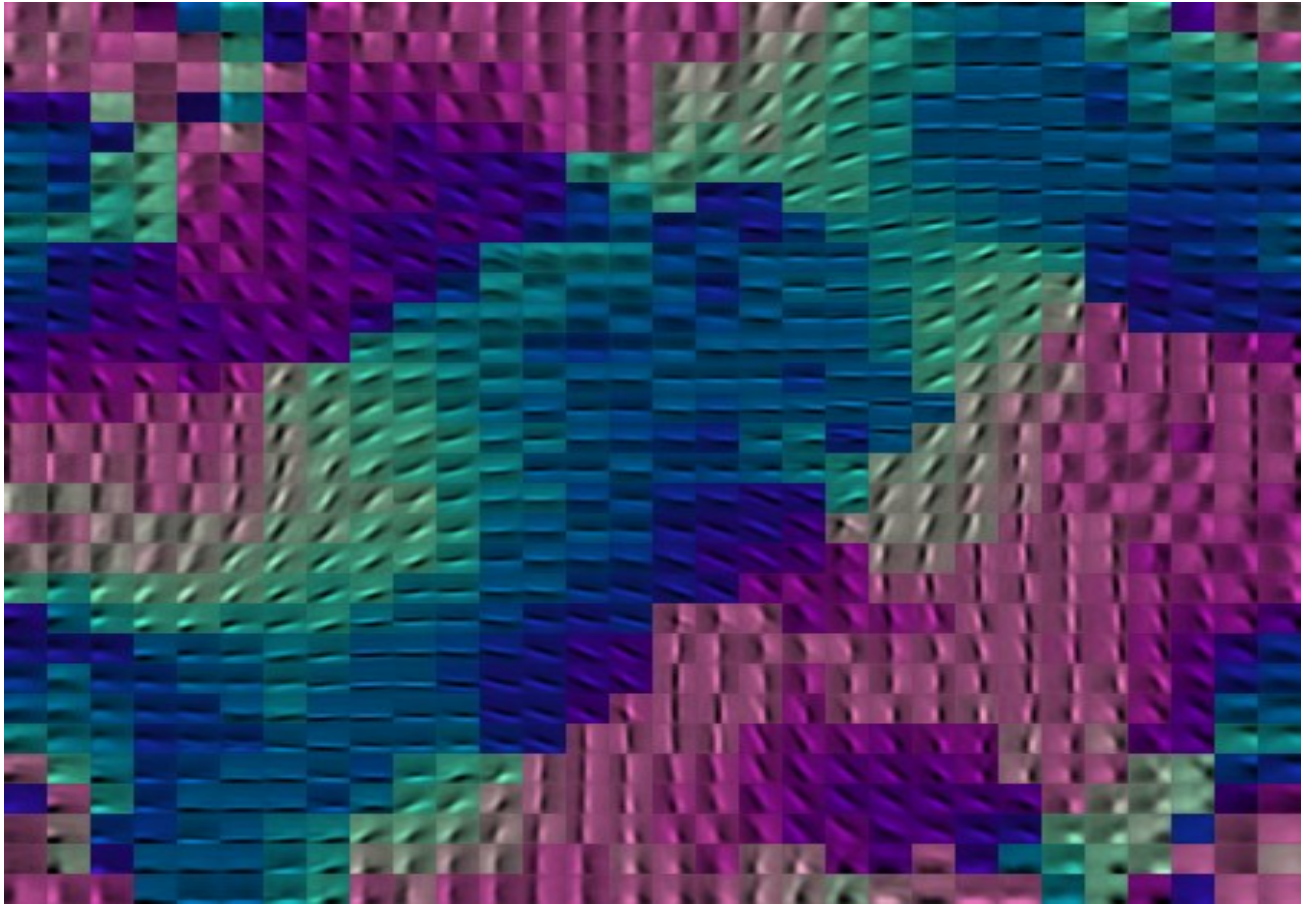




Michael C. Crair, et. al. The Journal of Neurophysiology Vol. 77 No. 6 June 1997, pp. 3381-3385 (**Cat**)

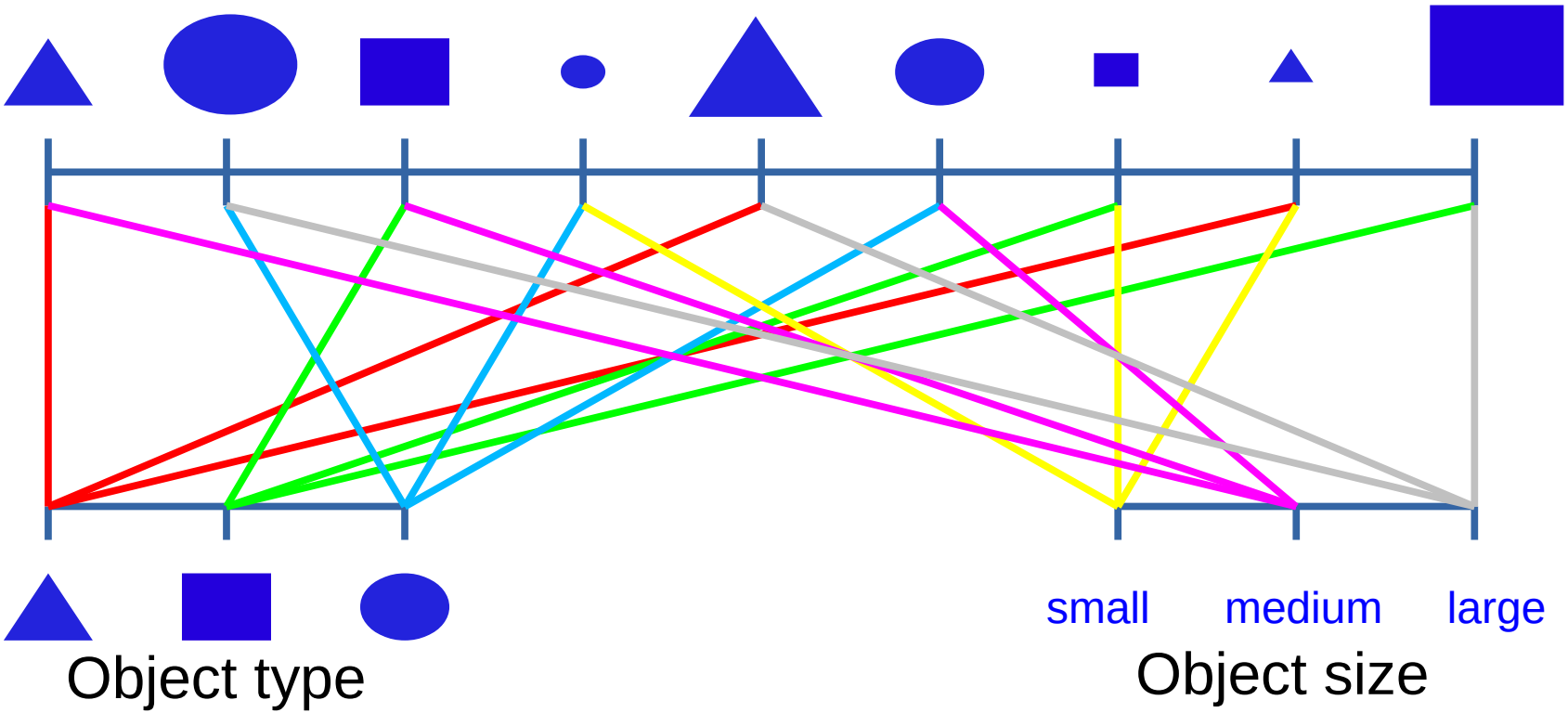
Same Method, withTraining at the Image Level (vs patch)

- **Color indicates orientation (by fitting Gabors)**

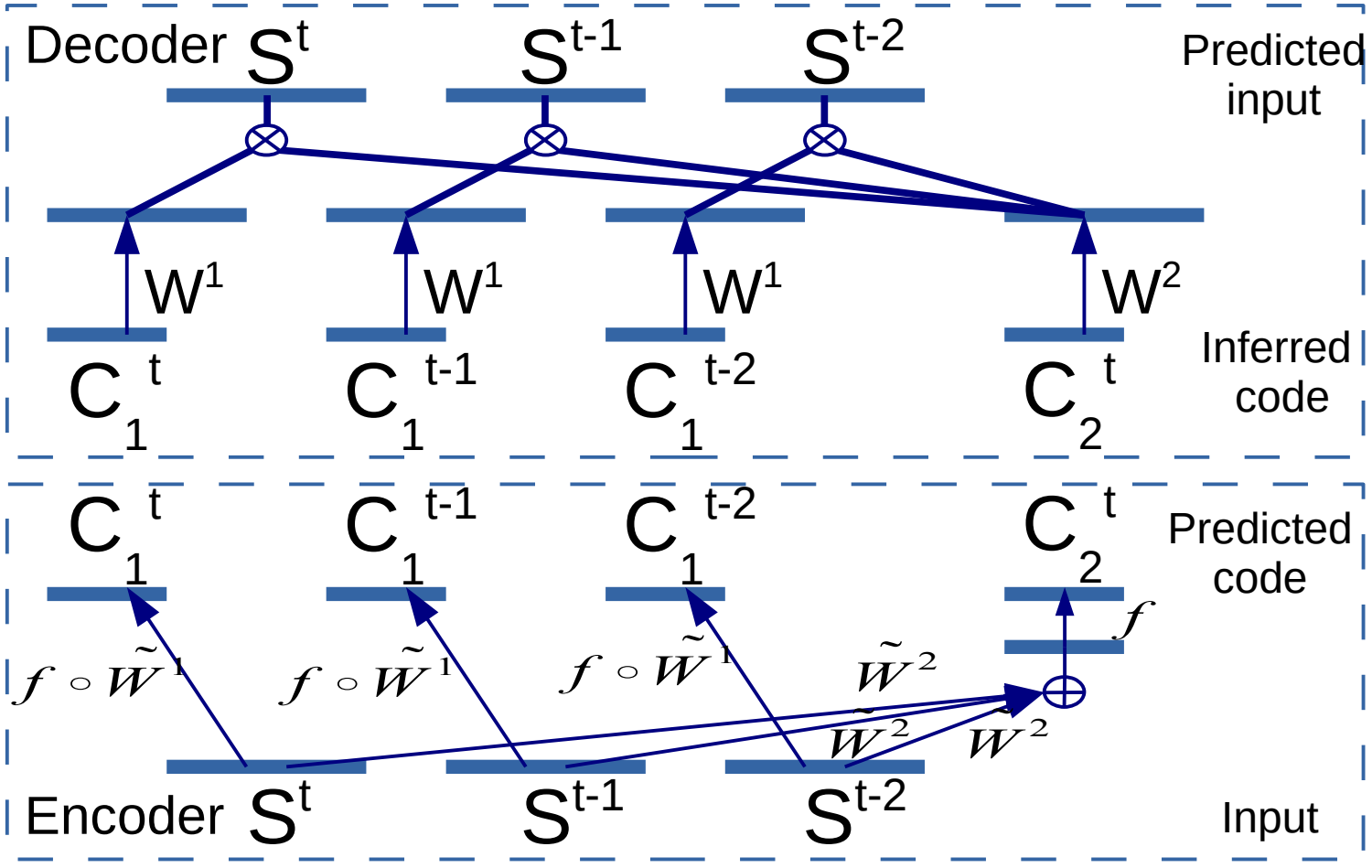


Invariant Features through Temporal Constancy

- **[Gregor & LeCun arXiv:1105.5307]**
  - "Efficient Learning of Sparse Invariant Representations"
- Object is cross-product of object type and instantiation parameters
  - What-where product network: [Hinton 1981] [von der Malsburg 1985]

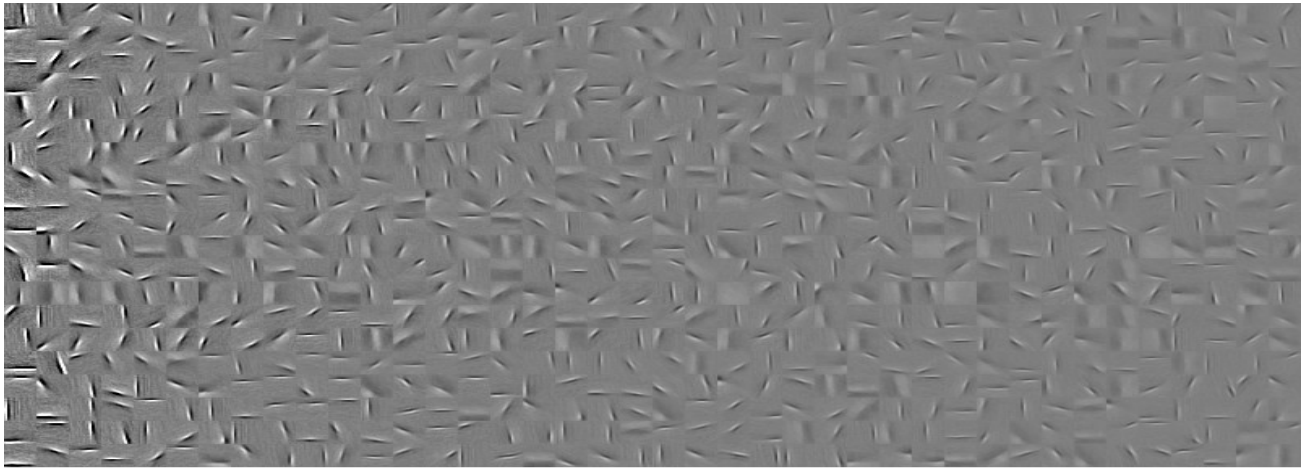


Invariant Features through Temporal Constancy

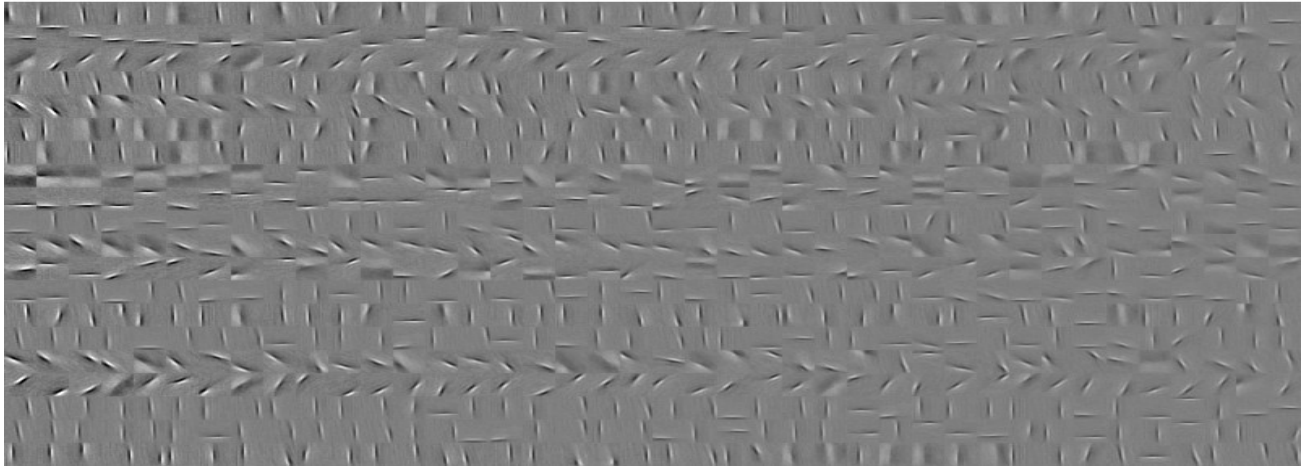


Invariant Features through Temporal Constancy

C1
(where)



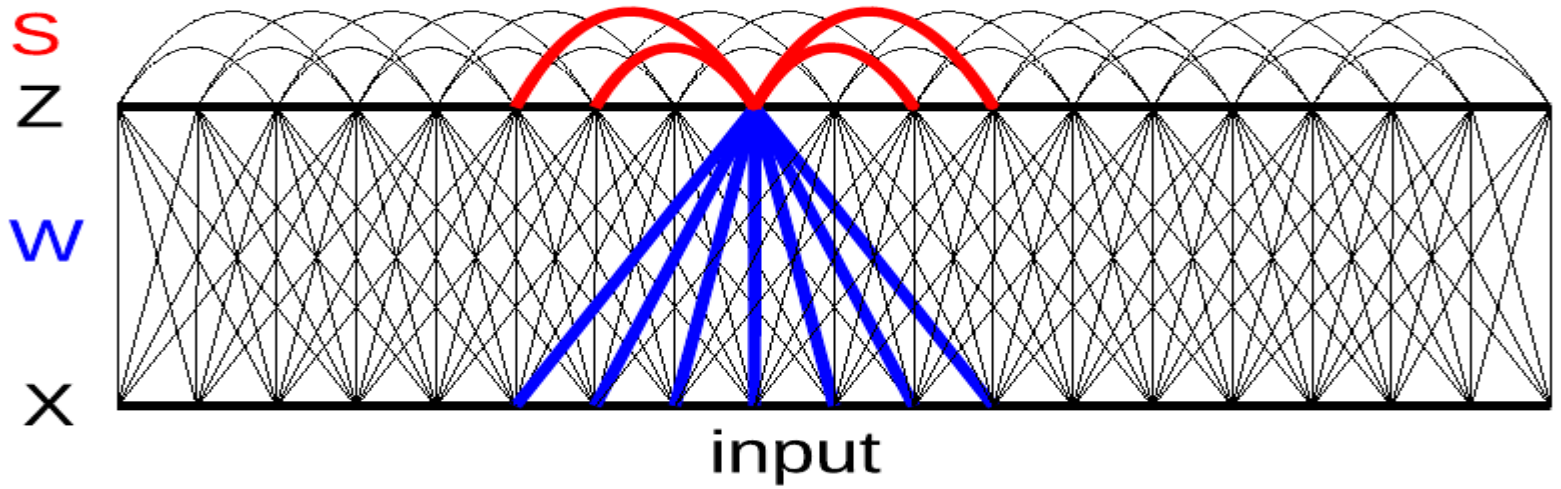
C2
(what)



Invariant Features through Lateral Inhibition

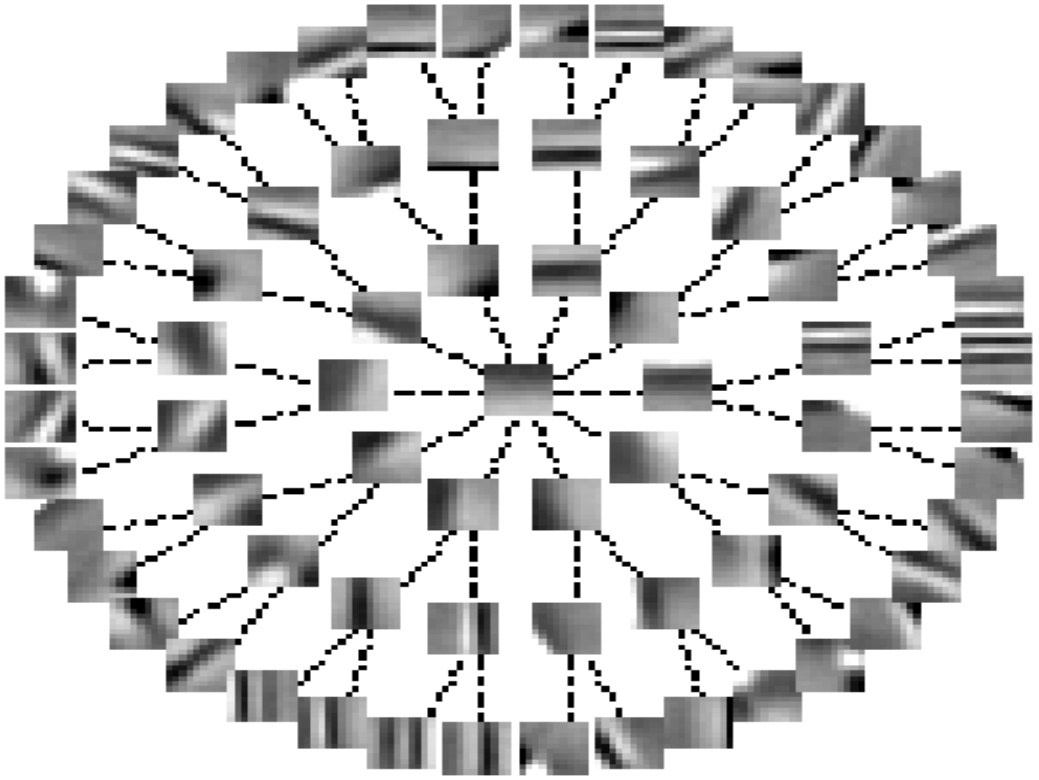
- **[Gregor, Szlam, LeCun NIPS 2011]**
- **Replace the L1 sparsity term by a lateral inhibition matrix**

$$\min_{W,Z} \sum_{x \in X} ||Wz - x||^2 + |z|^T S|z|$$



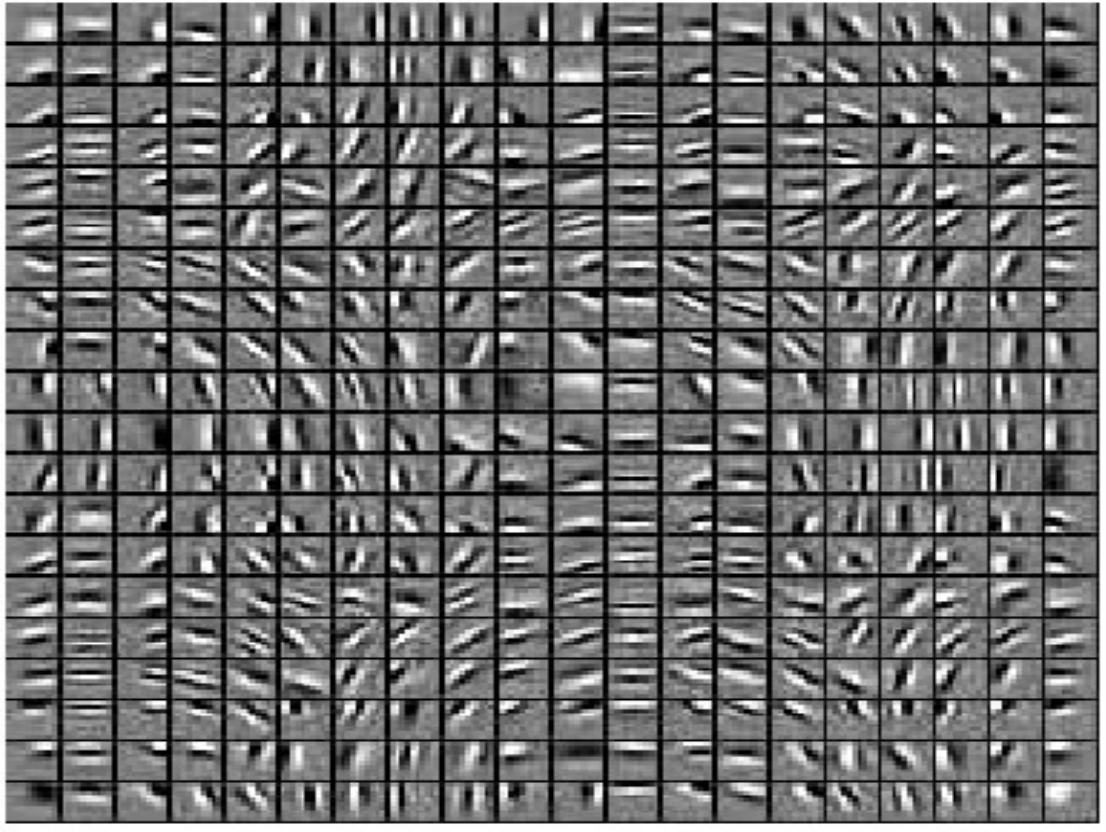
Invariant Features Lateral Inhibition

- **Zeros in S matrix have tree structure**



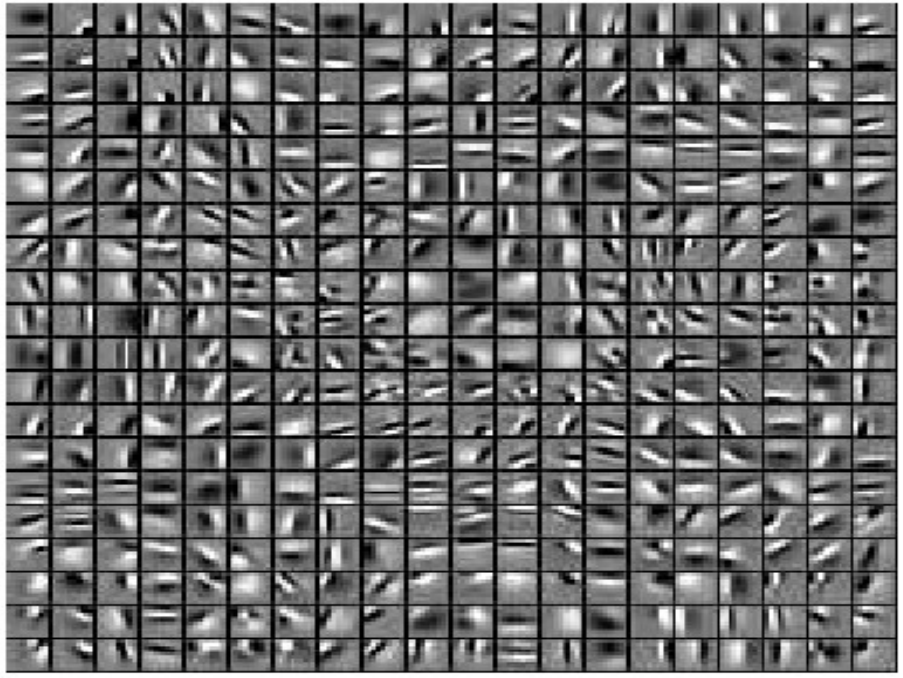
Invariant Features Lateral Inhibition

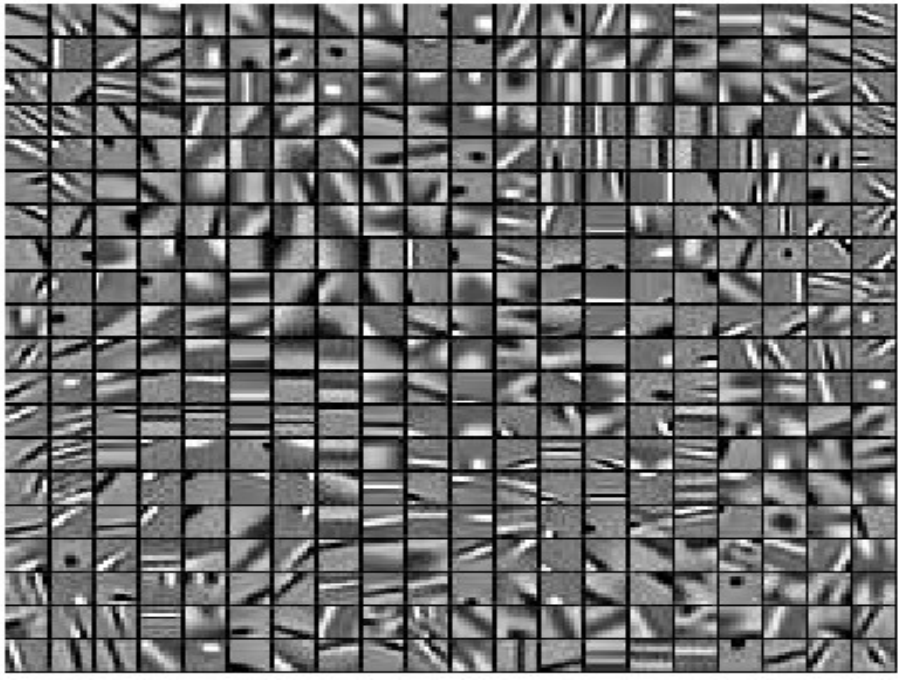
- **Non-zero values in S form a ring in a 2D topology**
  - Input patches are high-pass filtered



Invariant Features Lateral Inhibition

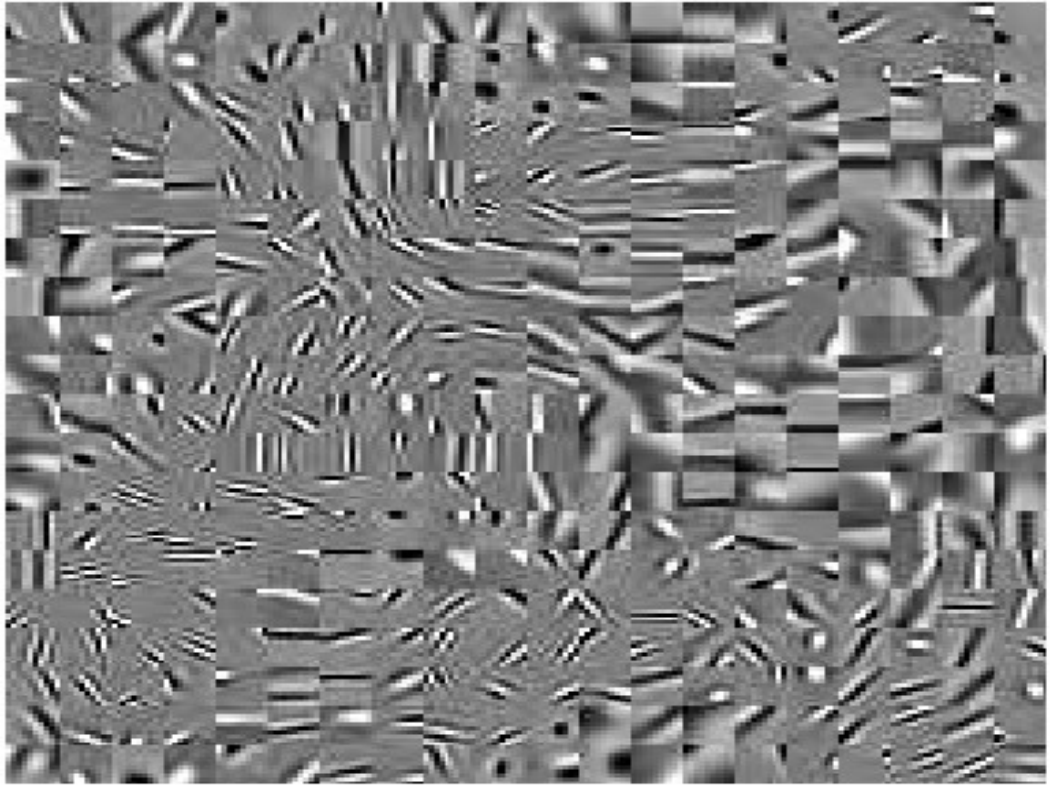
- **Non-zero values in S form a ring in a 2D topology**
  - Left: non high-pass filtering of input
  - Right: patch-level mean removal





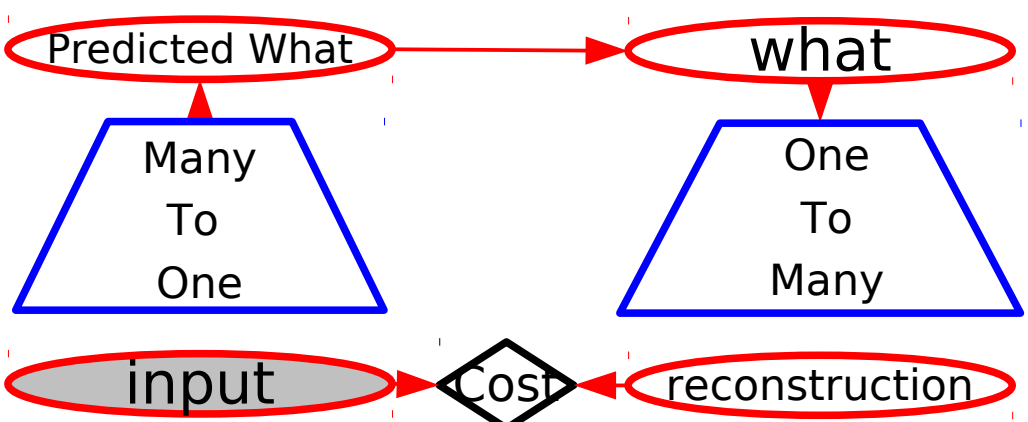
Invariant Features Short-Range Lateral Excitation + L1

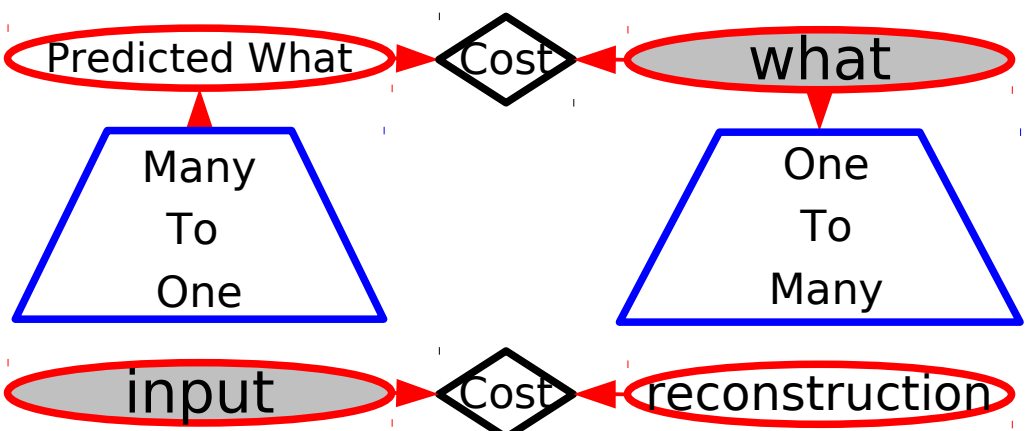
Orientation selectivity with multiple scales



Supervised and Unsupervised in One Learning Rule?

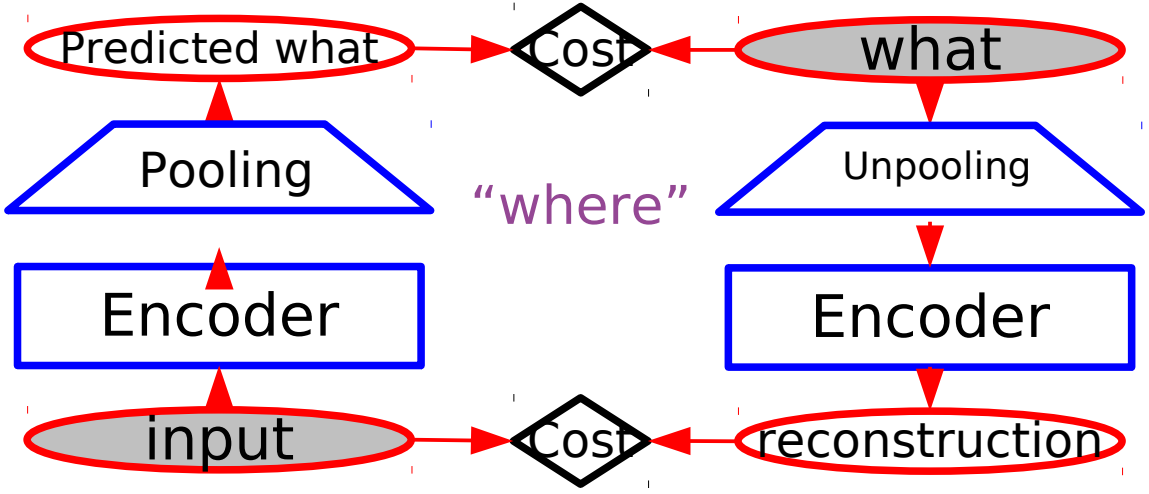
- **Boltzmann Machines have all the right properties [Hinton 1893] [OK, OK 1983 ;-]**
  - Sup & unsup, generative & discriminative in one simple/local learning rule
  - Feedback circuit reconstructs and propagates virtual hidden targets
  - But they don't really work (or at least they don't scale).
- **Problem: the feedforward path eliminates information**
- **If the feedforward path is invariant, then the reconstruction path is a one-to-many mapping**
  - Usual solution: sampling. But I'm allergic.





Supervised and Unsupervised in One Learning Rule?

- **Idea: keep the complementary information around**
  - So that the generative path is a function
- **What-Where Auto-Encoder**
  - [Ranzato 2007], [Gregor 2011], Hinton's Capsules
  - Very old ideas by Hinton & Zemel, von der Malsburg and others about separating identity from instantiation parameters.

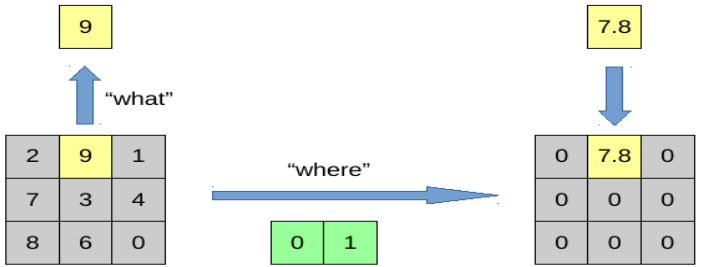


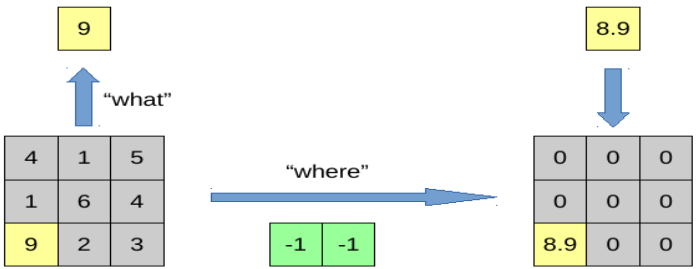
Computing the "where": Phase Pooling

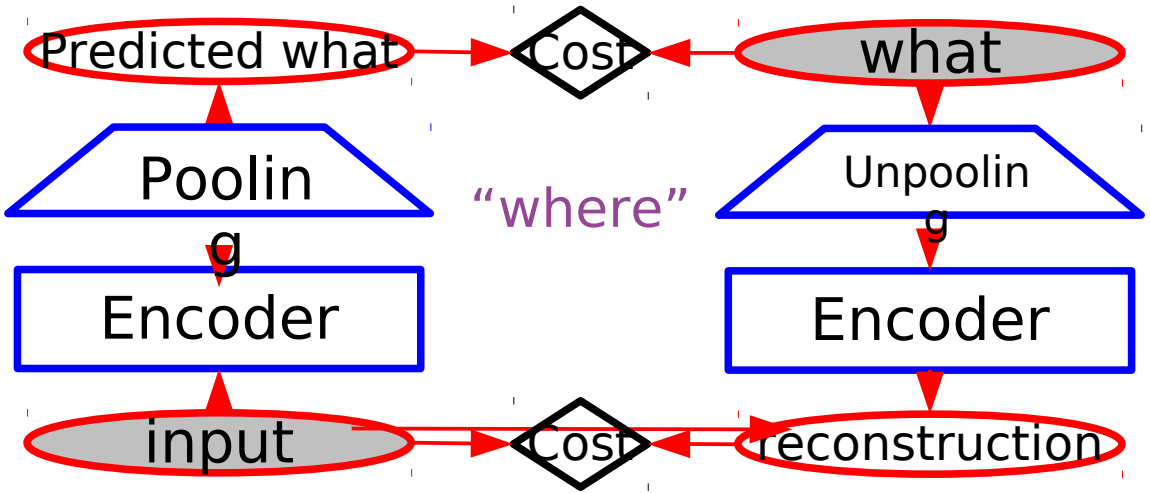
A funny kind of pooling/unpooling

$$m_k = \sum_{N_k} z(f,x,y) \frac{e^{\beta z(f,x,y)}}{\sum_{N_k} e^{\beta z(f',x',y')}} \approx \max_{N_k} z(f,x,y)$$

$$\mathbf{p}_k = \sum_{N_k} \begin{bmatrix} f \\ x \\ y \end{bmatrix} \frac{e^{\beta z(f,x,y)}}{\sum_{N_k} e^{\beta z(f',x',y')}} \approx \arg\max_{N_k} z(f,x,y)$$

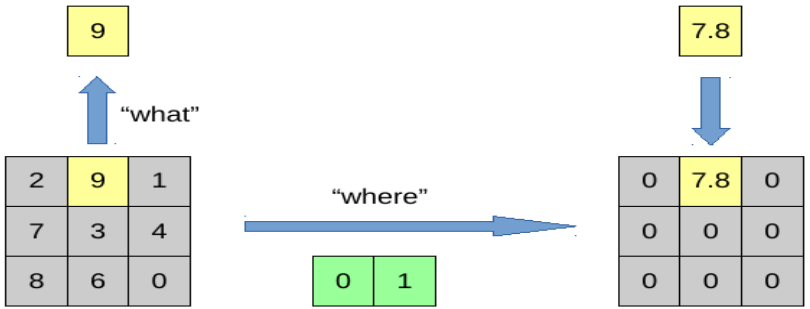




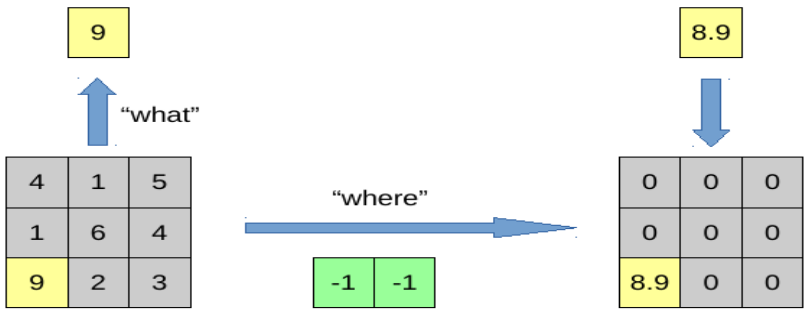


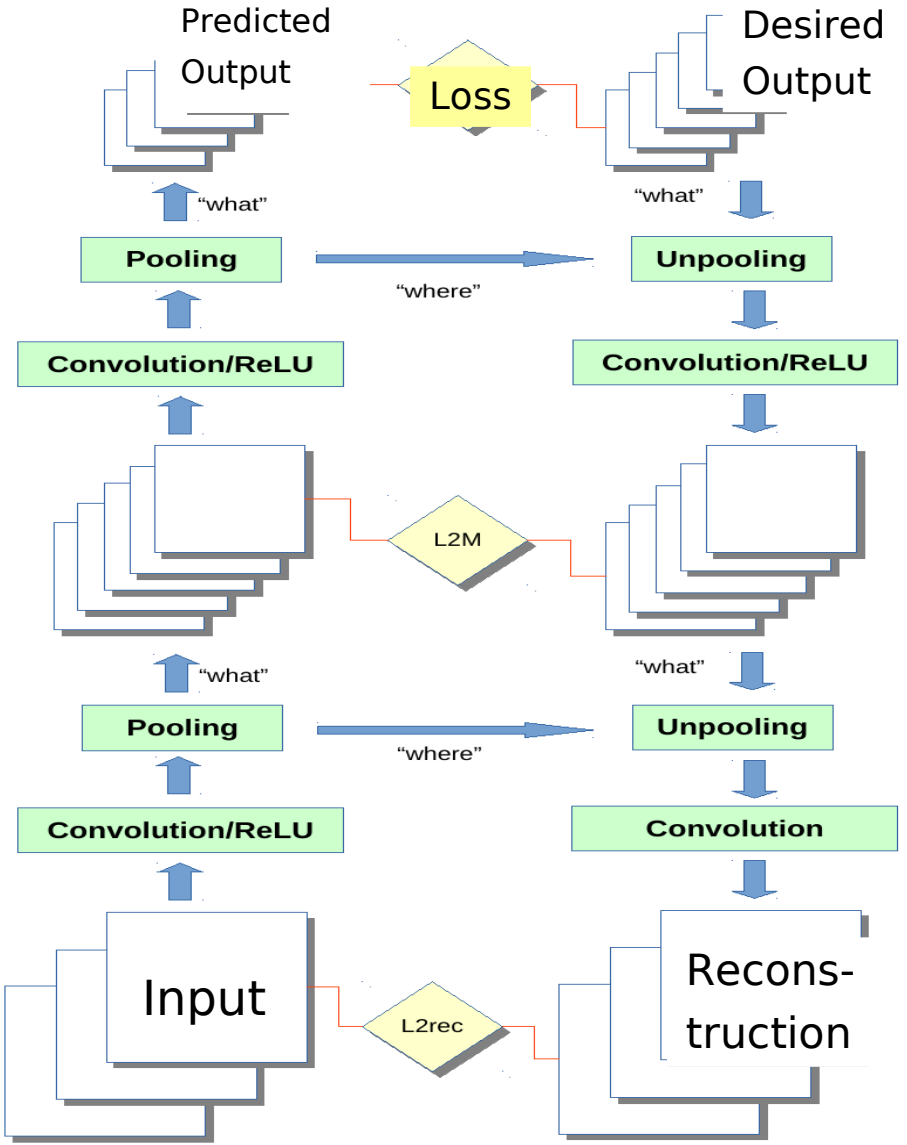
Stacked What-Where Auto-Encoder (SWWAE)

- [Zhao, Mathieu, LeCun arXiv:1506.023
- **Stacked What-Where Auto-Encoder**



- **A bit like a ConvNet paired with a DeConvNet**





SWWAE: Reconstructions with Unpooling

- **Network: MNNIST → [Conv 5x5] → 16 fmaps → Conv 3x3 → 32 fmaps → Pooling PxP**
- **Trained unsupervised. Hard max-pooling at test time.**

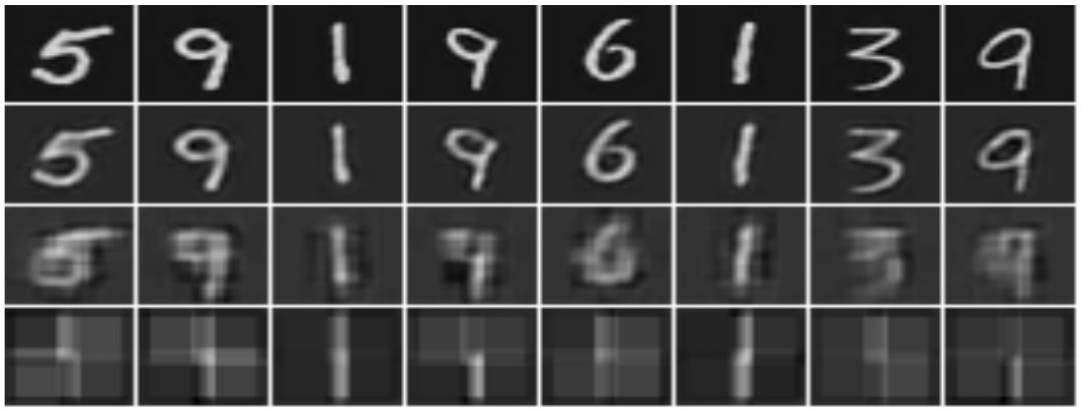
Upsampling

P=2

P=4

P=8

P=16



Unpooling

P=2

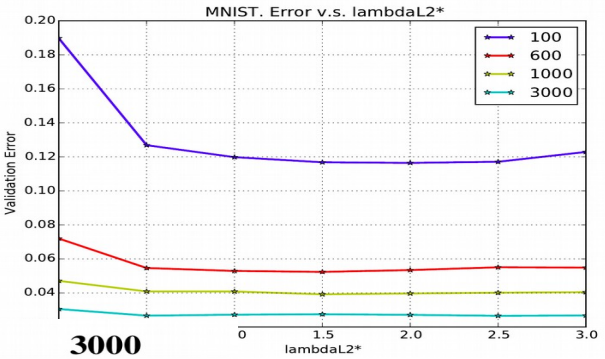
P=4

P=8

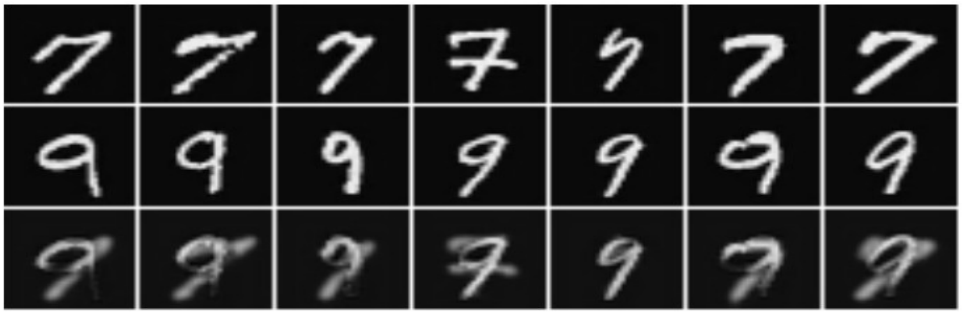
P=16

MNIST: Recognition & Generation

model / N	100	600	1000	3000
SWWAE	[1]11.65 ± 0.20	[1]5.24 ± 0.06	[1]3.92 ± 0.09	[2]2.66 ± 0.07
dp	21.11 ± 0.65	7.11 ± 0.27	5.34 ± 0.39	3.23 ± 0.30
dp-fc	16.09 ± 0.96	6.14 ± 0.36	4.368 ± 0.50	2.98 ± 0.08
unsup-sfx	17.81 ± 0.06	8.41 ± 0.08	6.40 ± 0.06	4.76 ± 0.03
unsup-pretr	-	9.80 ± 0.06	6.135 ± 0.03	4.41 ± 3.11
noL2M	[2]13.48 ± 2.35	[2]5.69 ± 0.33	[2]3.97 ± 0.37	[1]2.52 ± 0.06

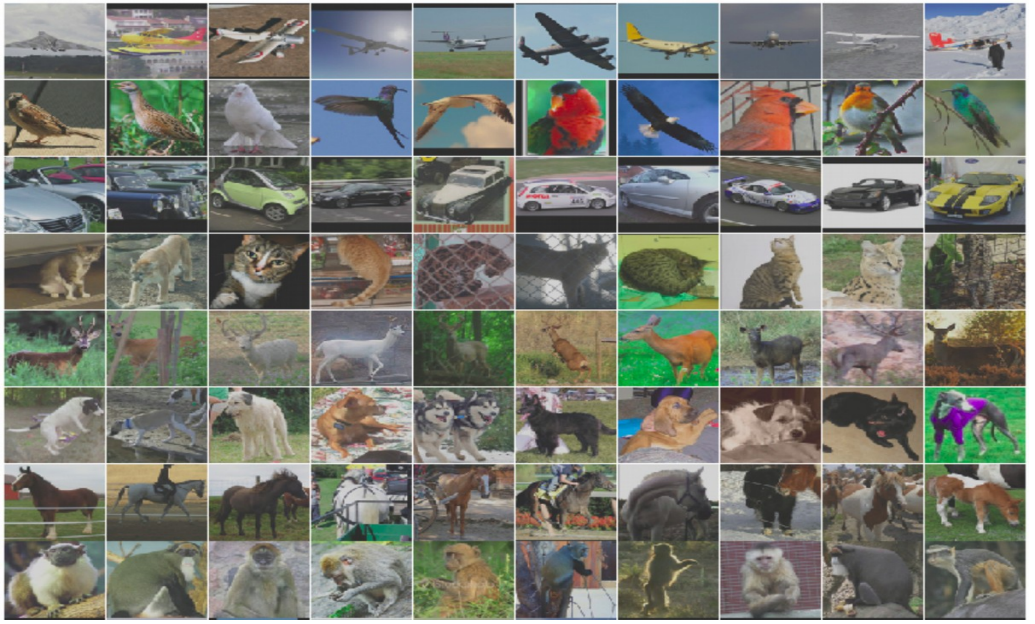


model / N	100	600	1000	3000
Convnet (LeCun et al. (1998))	22.98	7.86	6.45	3.35
TSVM (Vapnik & Vapnik (1998))	16.81	6.16	5.38	3.45
CAE (Rifai et al. (2011b))	13.47	6.3	4.77	3.22
MTC (Rifai et al. (2011a))	12.03	5.13	3.64	2.57
PL-DAE (Lee (2013))	10.49	5.03	3.46	2.69
WTA-AE (Makhzani & Frey (2014))	-	**2.37**	1.92	-
M1+M2 (Kingma et al. (2014))	3.33 ± 0.14	2.59 ± 0.05	2.40 ± 0.02	2.18 ± 0.04
LadderNetwork (Rasmus et al. (2015a))	**1.13 ± 0.04**	-	**1.00 ± 0.06**	-
SWWAE without dropout	9.17 ± 0.11	4.16 ± 0.11	3.39 ± 0.01	2.50 ± 0.01
SWWAE with dropout	8.71 ± 0.34	3.31 ± 0.40	2.83 ± 0.10	**2.10 ± 0.22**



STL-10: Recognition

- 10 classes: airplane, bird, cat, car, deer, dog, horse, monkey, ship, truck
- 96x96 color images (from ImageNet)
- 10 predefined folds with 1000 training samples each (100 per class)
- 5000 total training samples
- 100K unlabeled samples (other categories)
- 8000 test samples (800 per class)
  [Coates et al. 2011]
- (64)3c-4p-(64)3c-3p-(128)3c-(128)3c-2p-(256)3c-(256)3c-(256)3c-(512)3c-(512)3c-(512)3c-2p-10fc

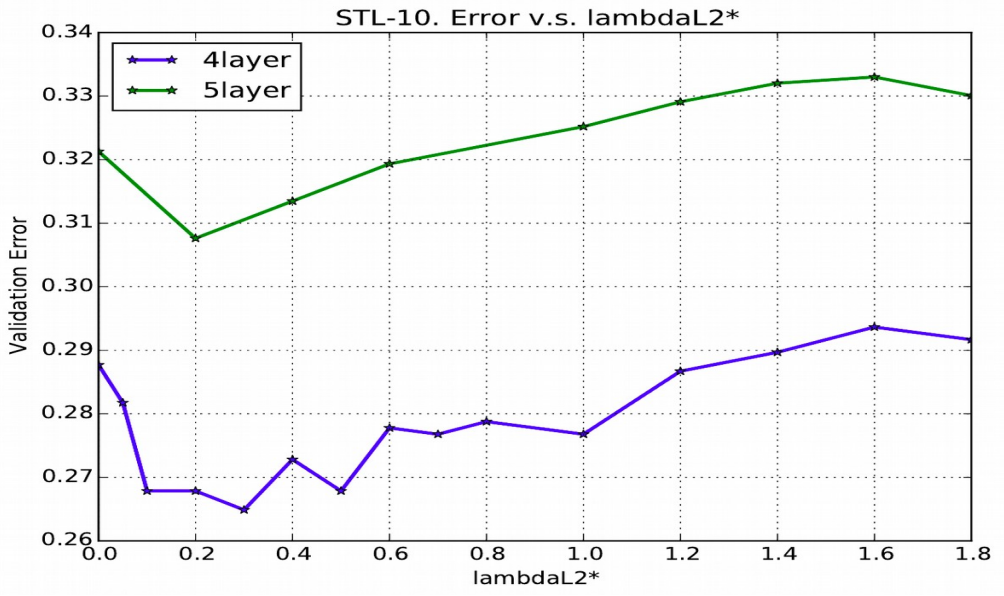


model	accuracy
Multi-task Bayesian Optimization (Swersky et al. (2013))	70.1%
Zero-bias Convnets + ADCU (Paine et al. (2014))	70.2%
Exemplar Convnets (Dosovitskiy et al. (2014))	**75.4%**
SWWAE	74.33%
Convnet of same configuration	57.45%

STL-10: Recognition

- 10 classes: airplane, bird, cat, car, deer, dog, horse, monkey, ship, truck
- 96x96 color images (from ImageNet)
- 10 predefined folds with 1000 training samples each (100 per class)
- 5000 total training samples
- 100K unlabeled samples (other categories)
- 8000 test samples (800 per class)

[Coates et al, 2011]



model	accuracy(val)
SWWAE-4layer	[1]73.51%
SWWAE-5layer	69.24%
noL2M-5layer	68.26%
noL2M-4layer	70.93%
dp-4layer	70.54%
dp-fc-4layer	71.43%

model	accuracy
Convolutional Kernel Networks [17]	62.32%
HMP [1]	64.5%
NOMP [16]	67.9%
Multi-task Bayesian Optimization [27]	70.1%
Zero-bias Convnets + ADCU [20]	70.2%
Exemplar Convnets [2]	72.8%
SWWAE-4layer	[1]74.80%

CIFAR-10, CIFAR-100, Unsup on 80M Tiny Images

- **60,000 labeled samples**
- **80 Million unlabeled samples**
- **(128)3c-(256)3c-2p-(256)3c-(512)3c-2p-(512)3c-(512)3c-2p-(512)3c-(512)3c-2p-128fc-10fc**

model	CIFAR-10	CIFAR-100
All-Convnet (Springenberg et al. (2014))	**92.75%**	66.29%
Highway Network (Srivastava et al. (2015))	92.40%	67.76%
Deeply-supervised nets (Lee et al. (2014))	92.03%	65.43%
Fractional Max-pooling with large augmentation (Graham (2014))	95.50%	68.55%
SWWAE ($\lambda_{L2rec} = 1, \lambda_{L2M} = 0.2$)	92.23%	**69.12%**
Convnet of same configuration	91.33%	67.50%

Unsupervised Learning Through Temporal Prediction

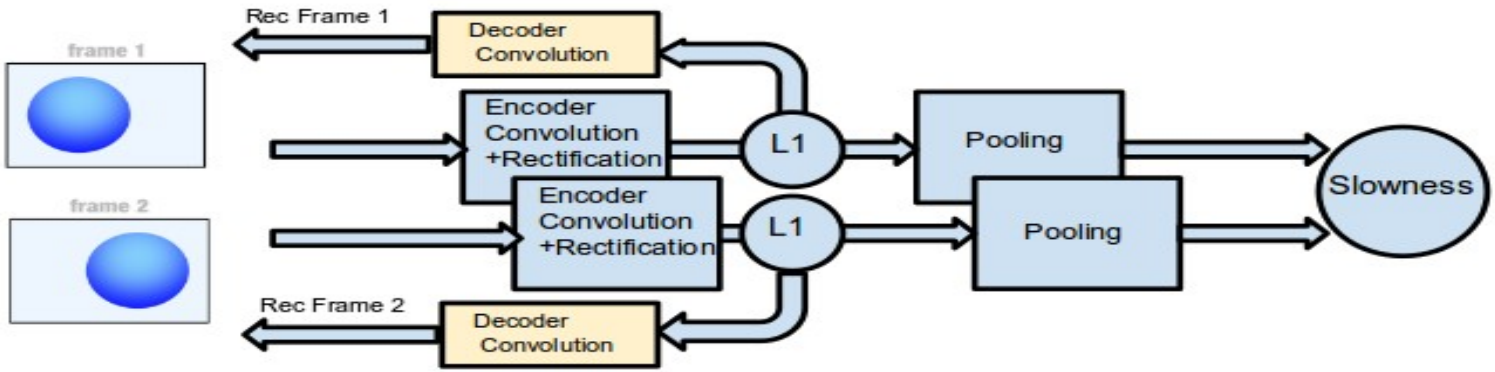
Goroshin & al

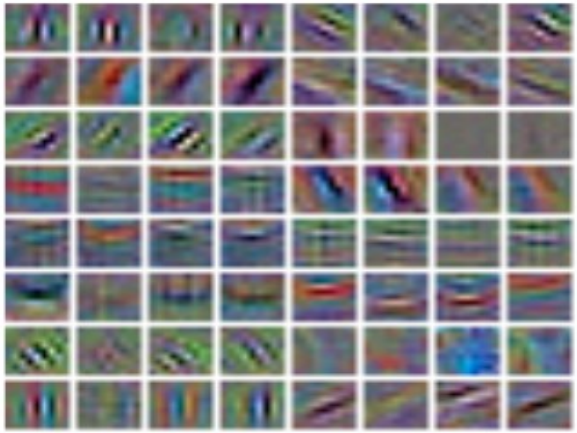
“Unsupervised Feature Learning from Temporal Data”

ICCV 2015 & arXiv:1504.02518

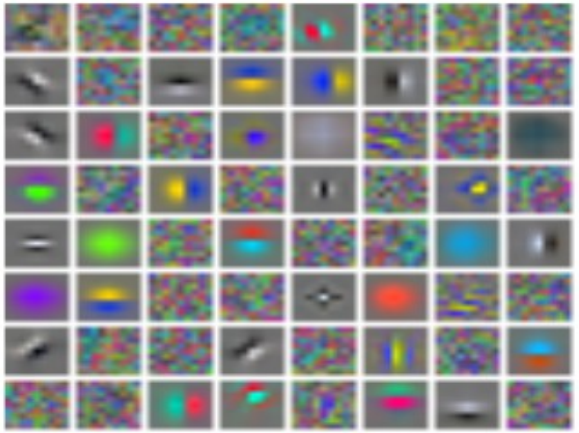
Goroshin, Mathieu, LeCun:

Sparse Auto-Encoder with "Slow Feature" Penalty

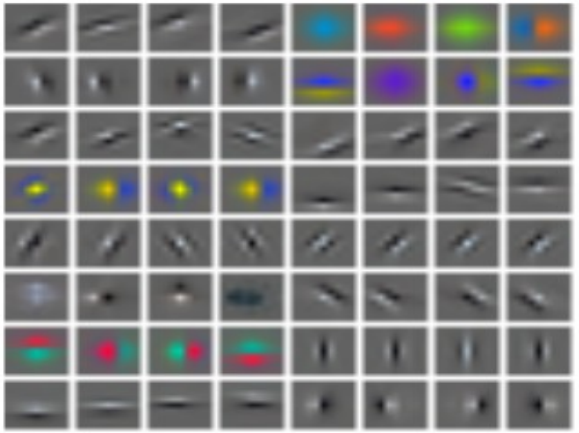




Supervised filters CIFAR10



sparse conv. auto-encoder



slow & sparse conv. auto-encoder
Trained on YouTube videos

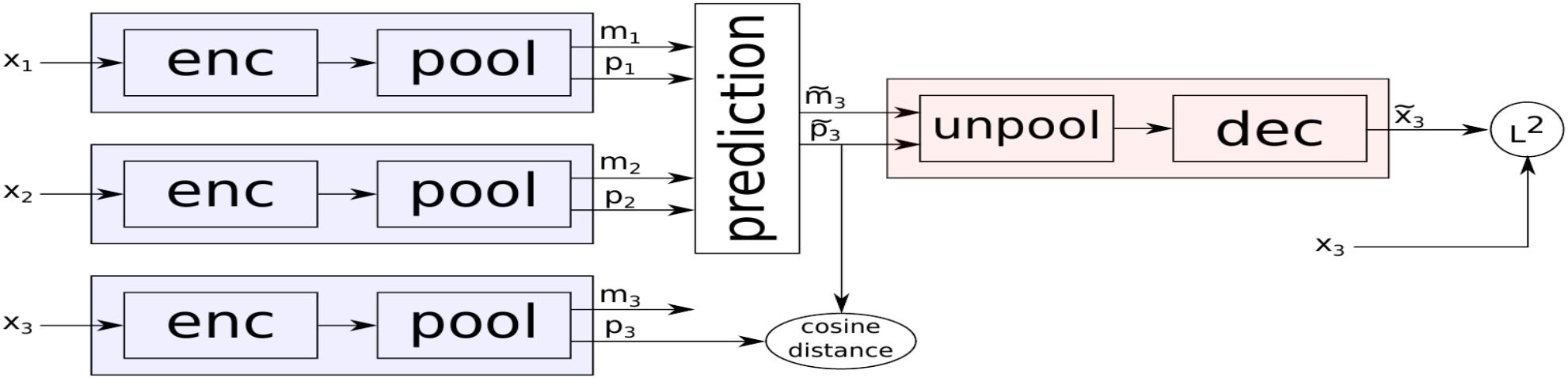
[Goroshin et al. ICCV 2015, Arxiv:1412.6056]

Unsupervised Learning by Prediction and Linearization

[Goroshin, Mathieu, LeCun NIPS 2015, arXiv:1506.03011]

- Trained on 3 successive frames of video
- Maximize the colinearity between successive changes of feature vectors
- So that "natural" changes stay in linear manifold in feature space.

$$L = \frac{1}{2}\|G_W(\mathbf{a}\begin{bmatrix} z^t & z^{t-1}\end{bmatrix}^T) - x^{t+1}\|_2^2 - \lambda \frac{(z^t - z^{t-1})^T(z^{t+1} - z^t)}{\|z^t - z^{t-1}\|\,\|z^{t+1} - z^t\|}$$



Unsupervised Learning by Prediction and Linearization

Y. LeCun

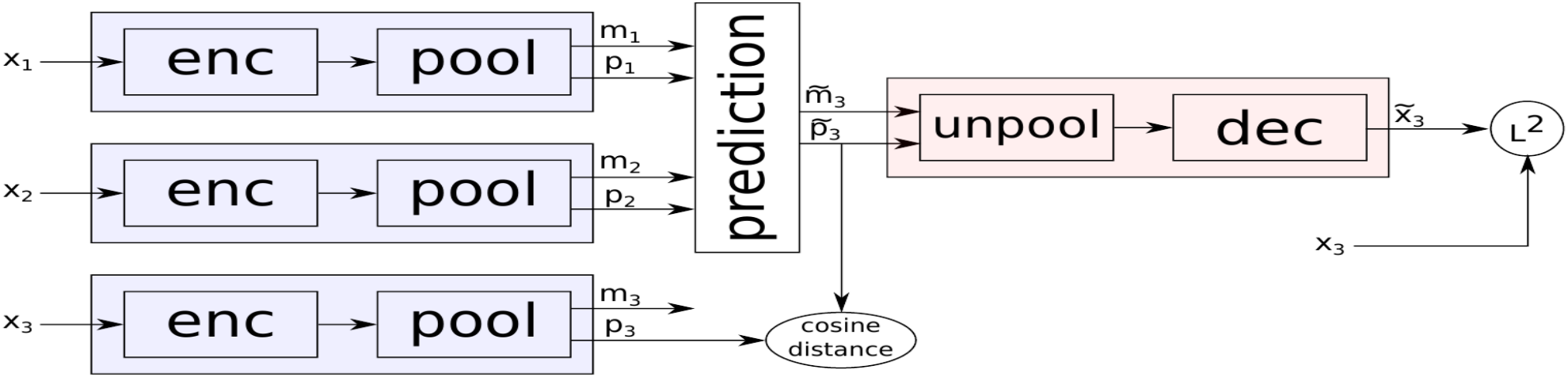
$$L = \frac{1}{2}\|G_W(\mathbf{a}\,[z^t \quad z^{t-1}]^T) - x^{t+1}\|_2^2 - \lambda \frac{(z^t - z^{t-1})^T(z^{t+1} - z^t)}{\|z^t - z^{t-1}\|\,\|z^{t+1} - z^t\|}$$

- **Magnitude**
  - (soft max)

$$m_k = \sum_{N_k} z(f,x,y)\frac{e^{\beta z(f,x,y)}}{\sum_{N_k} e^{\beta z(f',x',y')}} \approx \max_{N_k} z(f,x,y)$$

- **Phase**
  - (soft argmax)

$$\mathbf{p}_k = \sum_{N_k} \begin{bmatrix} f \\ x \\ y \end{bmatrix} \frac{e^{\beta z(f,x,y)}}{\sum_{N_k} e^{\beta z(f',x',y')}} \approx \arg\max_{N_k} z(f,x,y)$$

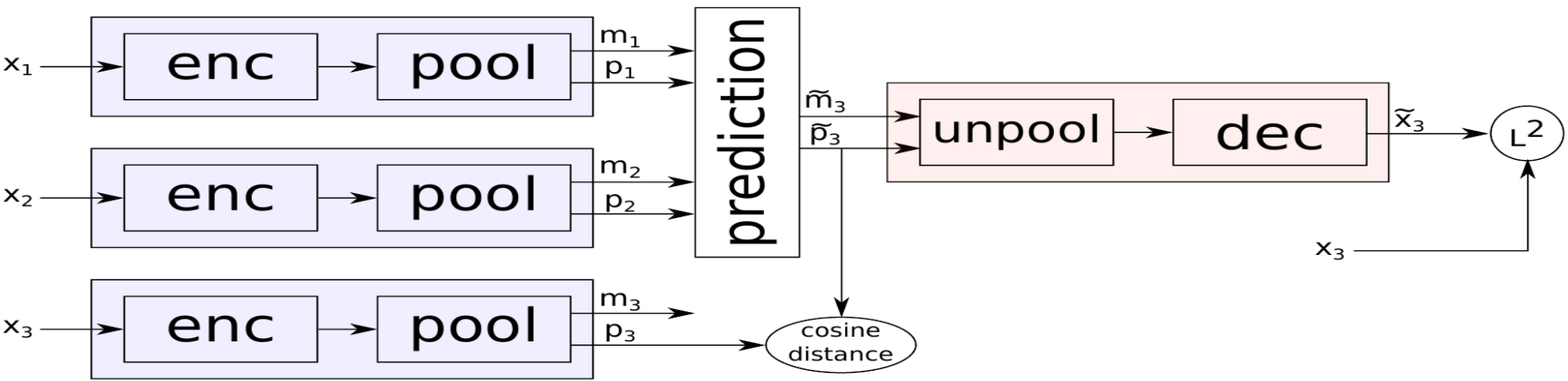


Unsupervised Learning by Prediction and Linearization

[Goroshin, Mathieu, LeCun NIPS 2015, arXiv:1506.03011]

- **Version 2:**
  - Make sure the “what”/magnitude changes slowly
  - Make sure the “where”/phase changes linearly

$$m^{t+1} = \frac{m^t + m^{t-1}}{2}$$
$$\mathbf{p}^{t+1} = 2\mathbf{p}^t - \mathbf{p}^{t-1}$$



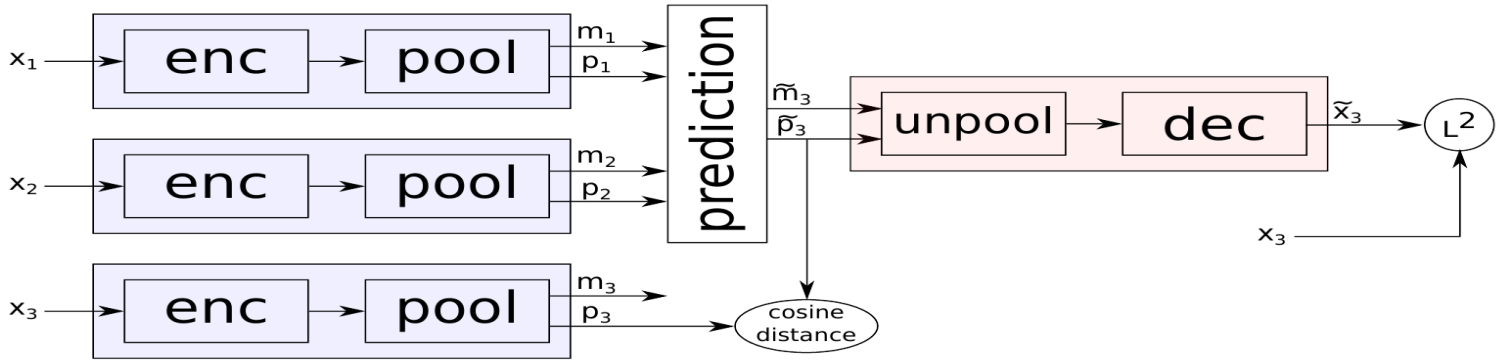
Unsupervised Learning by Prediction and Linearization

[Goroshin, Mathieu, LeCun arXiv:1506.03011]

- **Version 3: the world is unpredictable**
  - Add latent variable to compensate for that.
  - Minimize over them during training

$$\hat{z}_\delta^{t+1} = z^t + (W_1 \delta) \odot \mathbf{a} \begin{bmatrix} z^t & z^{t-1} \end{bmatrix}^T$$

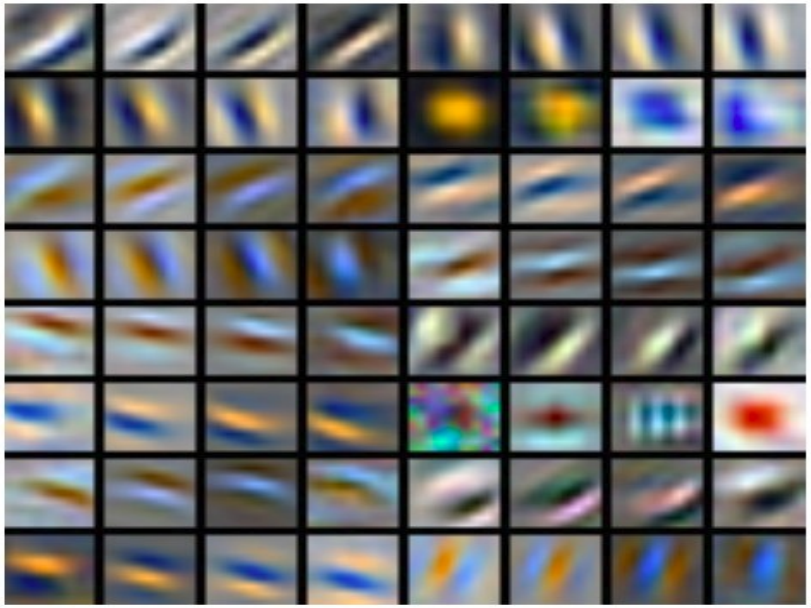
$$L = \min_\delta \| G_W(\hat{z}_\delta^{t+1}) - x^{t+1} \|_2^2 - \lambda \frac{(z^t - z^{t-1})^T (z^{t+1} - z^t)}{\| z^t - z^{t-1} \| \| z^{t+1} - z^t \|}$$



Architectures

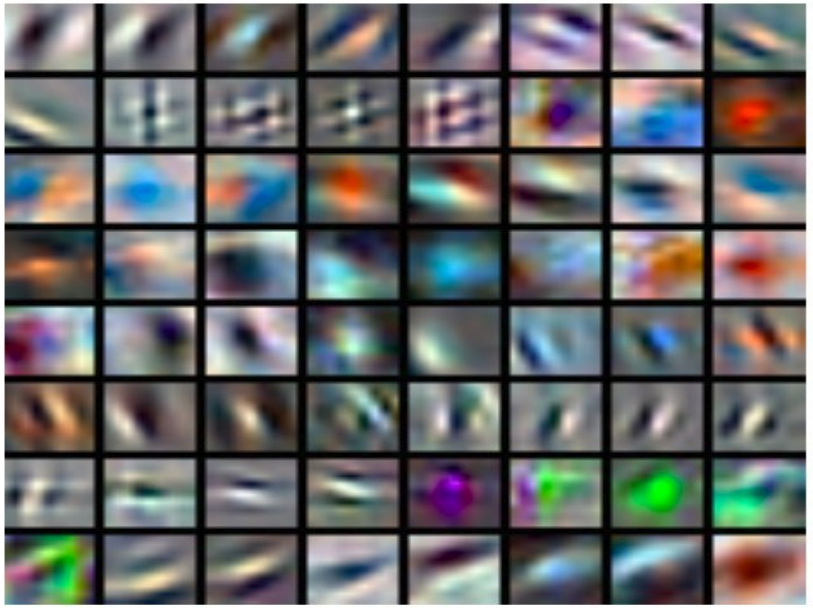
	Encoder	Prediction	Decoder
Shallow Architecture 1	Conv+ReLU $64 \times 9 \times 9$ Phase Pool 4	Average Mag. Linear Extrap. Phase	Conv $64 \times 9 \times 9$
Shallow Architecture 2	Conv+ReLU $64 \times 9 \times 9$ Phase Pool 4 stride 2	Average Mag. Linear Extrap. Phase	Conv $64 \times 9 \times 9$
Deep Architecture 1	Conv+ReLU $16 \times 9 \times 9$ Conv+ReLU $32 \times 9 \times 9$ FC+ReLU 8192×4096	None	FC+ReLU $\mathbf{8192} \times 8192$ Reshape $32 \times 16 \times 16$ SpatialPadding 8×8 Conv+ReLU $16 \times 9 \times 9$ SpatialPadding 8×8 Conv $1 \times 9 \times 9$
Deep Architecture 2	Conv+ReLU $16 \times 9 \times 9$ Conv+ReLU $32 \times 9 \times 9$ FC+ReLU 8192×4096	Linear Extrapolation	FC+ReLU 4096×8192 Reshape $32 \times 16 \times 16$ SpatialPadding 8×8 Conv+ReLU $16 \times 9 \times 9$ SpatialPadding 8×8 Conv $1 \times 9 \times 9$
Deep Architecture 3	Conv+ReLU $16 \times 9 \times 9$ Conv+ReLU $32 \times 9 \times 9$ FC+ReLU 8192×4096 Reshape $64 \times 8 \times 8$ Phase Pool 8×8	Average Mag. Linear Extrap. Phase	Unpool 8×8 FC+ReLU 4096×8192 Reshape $32 \times 16 \times 16$ SpatialPadding 8×8 Conv+ReLU $16 \times 9 \times 9$ SpatialPadding 8×8 Conv $1 \times 9 \times 9$

Filters: pooling over groups of 4 filters.



(a) Shallow Architecture 1

Non-overlapping pooling
4x4x4 (feat,x,y)

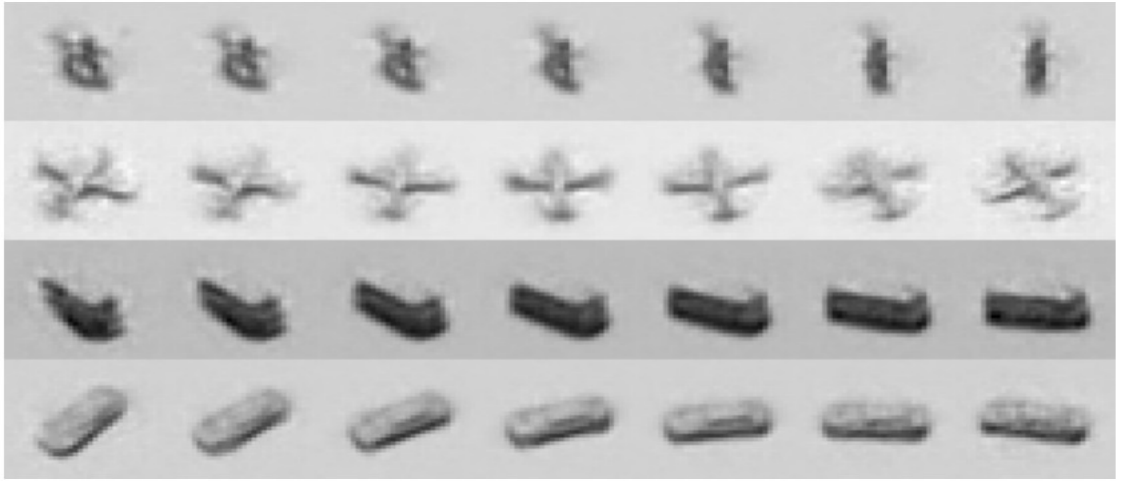


(b) Shallow Architecture 2

Overlapping pooling
4x4x4 (feat,x,y)
Stride 2 over features

Image interpolation in feature space: deterministic world

- **No phase pooling**
- **No curvature regularization**



- **With phase pooling**
- **With curvature regularization**

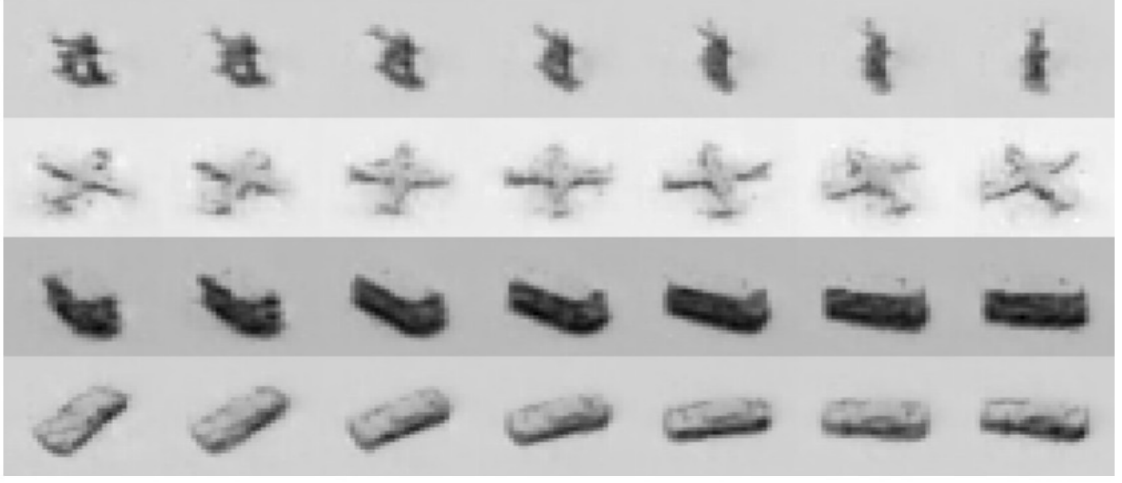


Image interpolation in feature space: deterministic world

- **Image interpolation with the basic model (siamese net)**

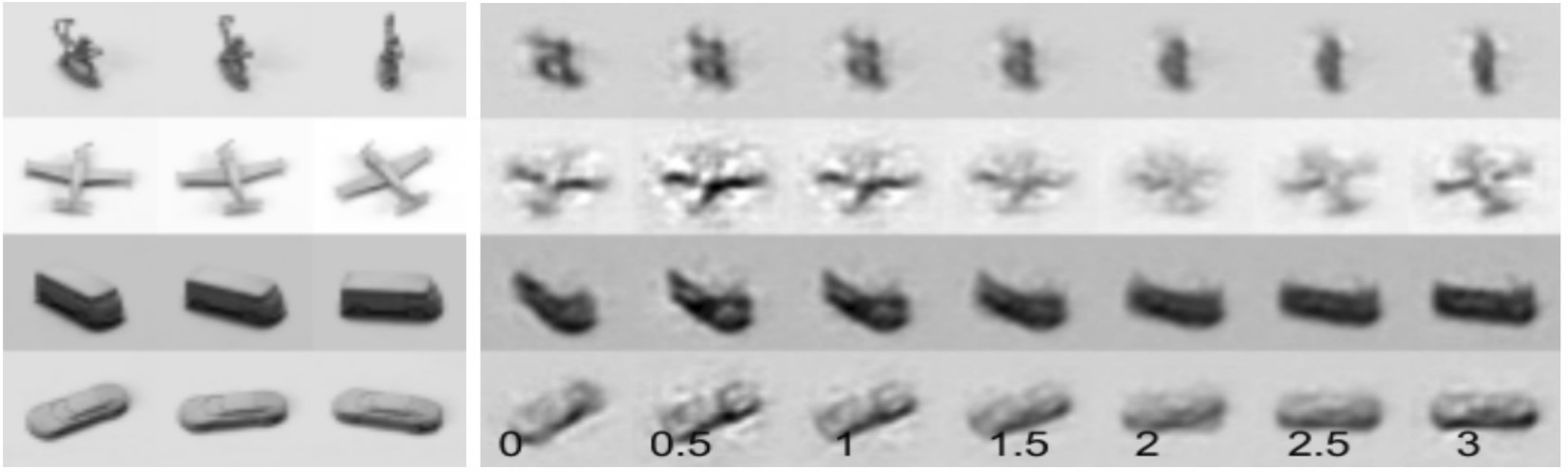
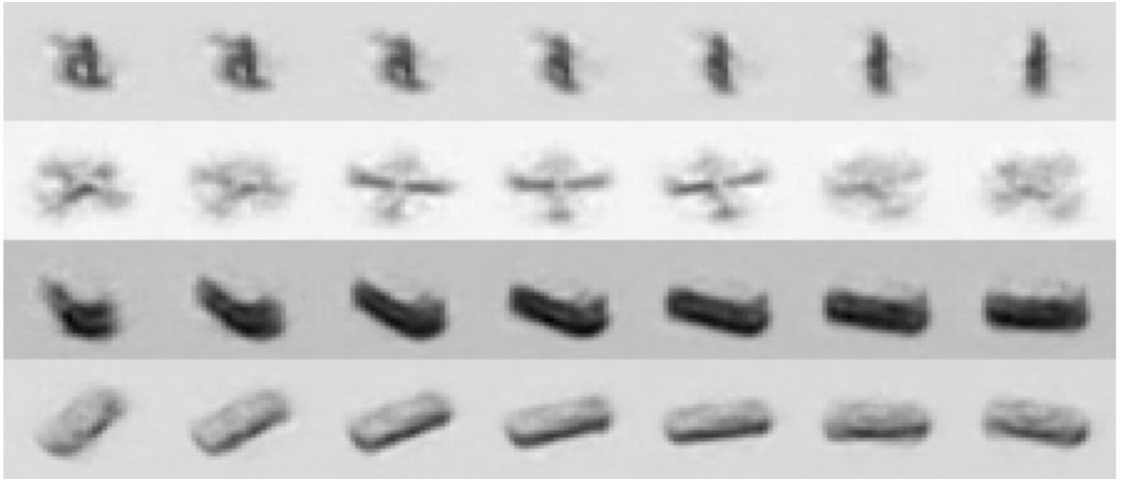


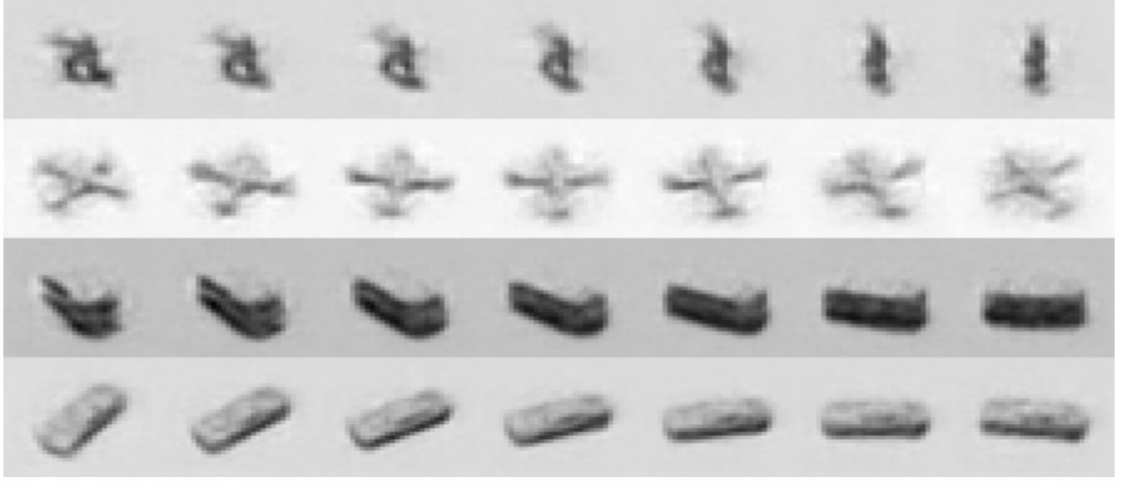
Image interpolation in feature space: unpredictable world

Y. LeCun

- **Phase interpolation**
- **No latent variable**



- **Phase interpolation**
- **With latent variable**



Is (Deep) Reinforcement Learning how Humans and Animals Learn?

Works great for games and virtual environments.

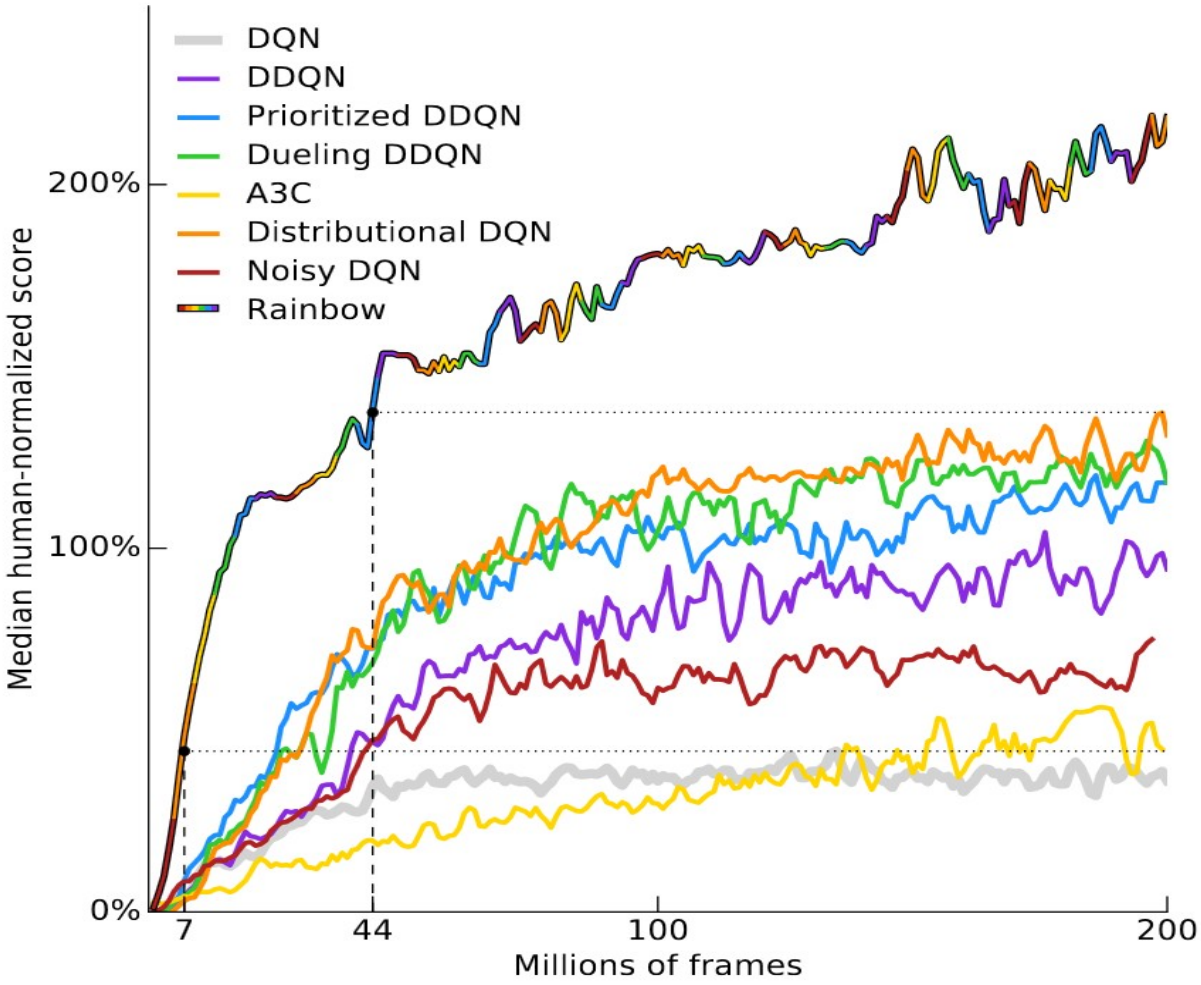
Too slow for the real-world.

Reinforcement Learning works fine for games

- **RL works well for games**
  - Playing Atari games [Mnih 2013], Go [Silver 2016, Tian 2018], Doom [Tian 2017], StarCraft...
  - RL requires too many trials.
  - 100 hours to reach the performance that a human can reach in 15 minutes on Atari games [Hessel ArXiv:1710.02298]
  - RL often doesn't really work in the real world
- **FAIR open Source go player: OpenGo**
  https://github.com/pytorch/elf

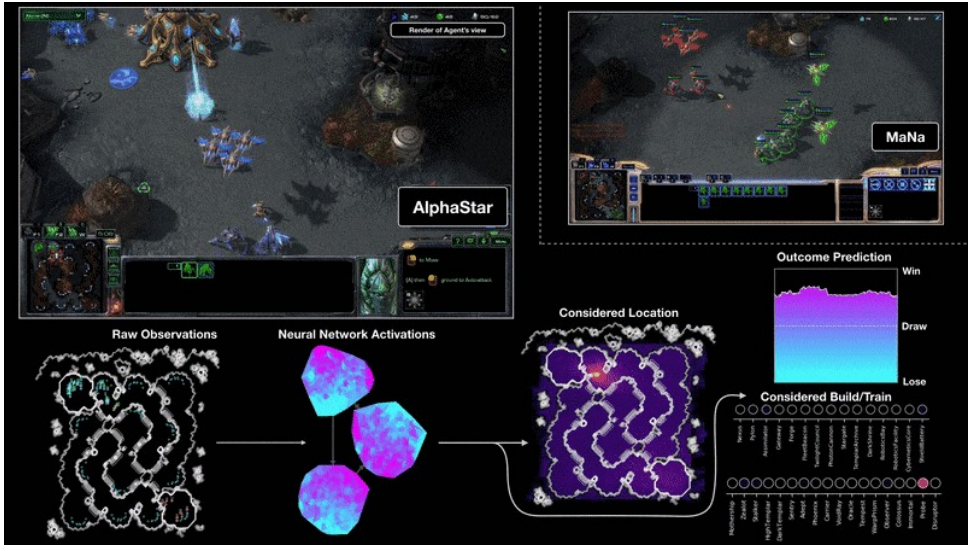
Pure RL requires many, many trials to learn a task

- [Hessel ArXiv:1710.02298]
- Median performance on 57 Atari games relative to human performance (100%=human)
- Most methods require over 50 million frames to match human performance (230 hours of play)
- The best method (combination) takes 18 million frames (83 hours).



AlphaStar: DeepMind's StarCarft II bot [Vinyals et al. 2019]

- **Large ConvNet+TransformerNet+LSTM**
  - Looks at the map and produces an action
  - Phase 1: imitation learning on 500,000 recorded games
  - Phase 2: reinforcement learning: equivalent to 200 years of play!



Pure RL is hard to use in the real world

- **Pure RL requires too many trials to learn anything**
  - it's OK in a game
  - it's not OK in the real world
- **RL works in simple virtual world that you can run faster than real-time on many machines in parallel.**

- **Anything you do in the real world can kill you**
- **You can't run the real world faster than real time**

What are we missing to get to “real” AI?

- **What we can have**
  - Safer cars, autonomous cars
  - Better medical image analysis
  - Personalized medicine
  - Adequate language translation
  - Useful but stupid chatbots
  - Information search, retrieval, filtering
  - Numerous applications in energy, finance, manufacturing, environmental protection, commerce, law, artistic creation, games,…..
- **What we cannot have (yet)**
  - Machines with common sense
  - Intelligent personal assistants
  - “Smart” chatbots”
  - Household robots
  - Agile and dexterous robots
  - Artificial General Intelligence (AGI)

How do Humans and Animal Learn?

So quickly

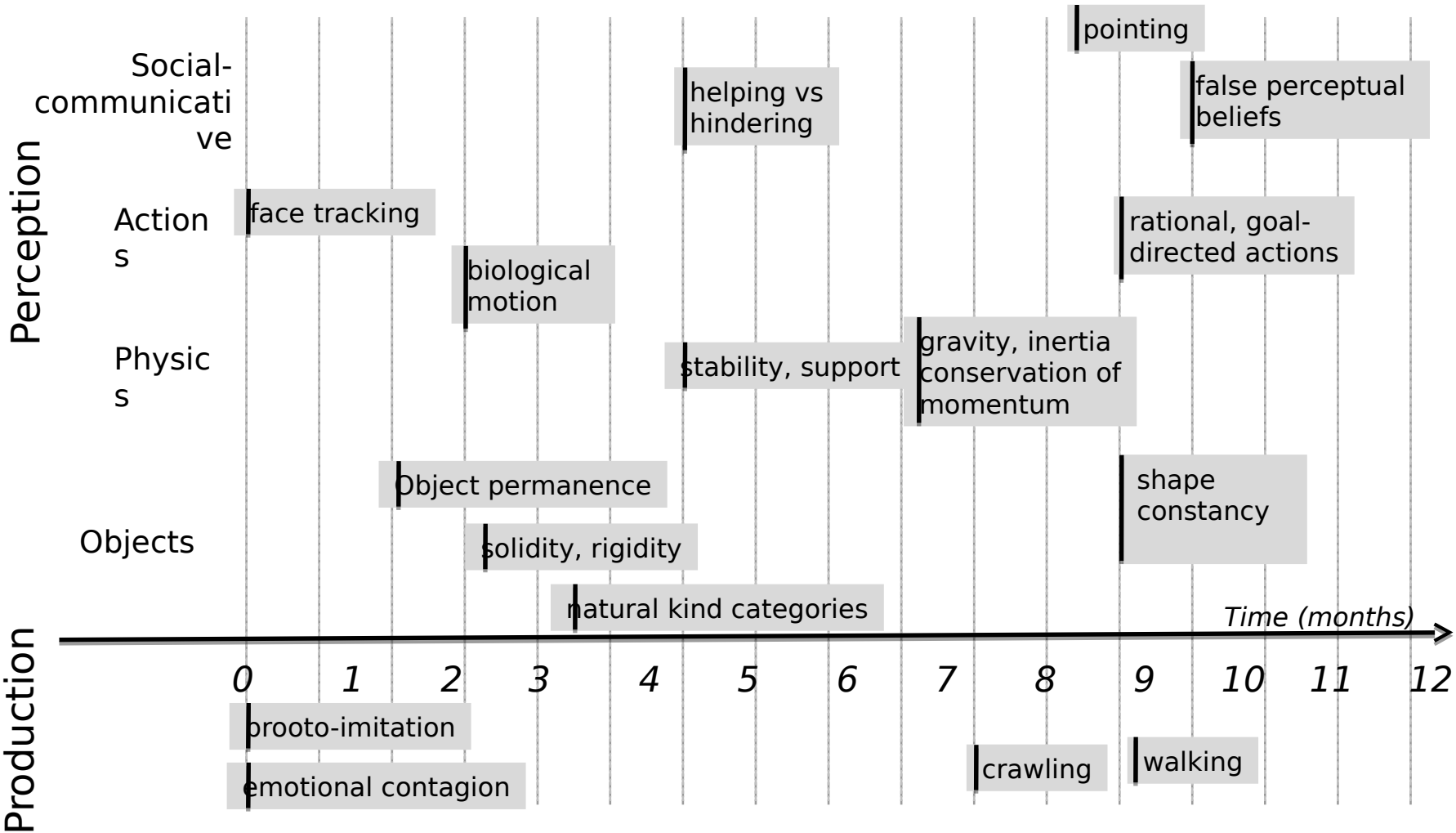
facebook
Artificial Intelligence Research

Babies learn how the world works by observation

- **Largely by observation, with remarkably little interaction.**

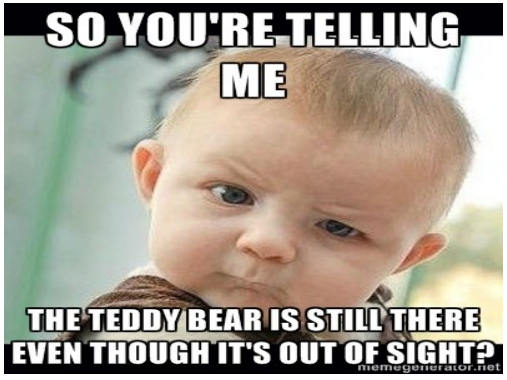
Photos courtesy of Emmanuel Dupoux

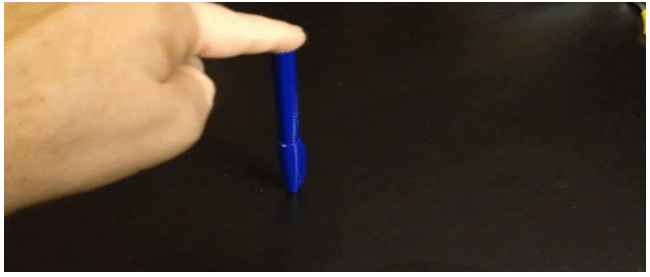
Early Conceptual Acquisition in Infants [from Emmanuel Dupoux]

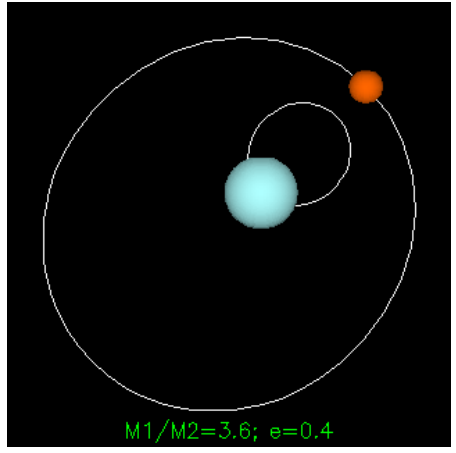


Prediction is the essence of Intelligence

- **We learn models of the world by predicting**







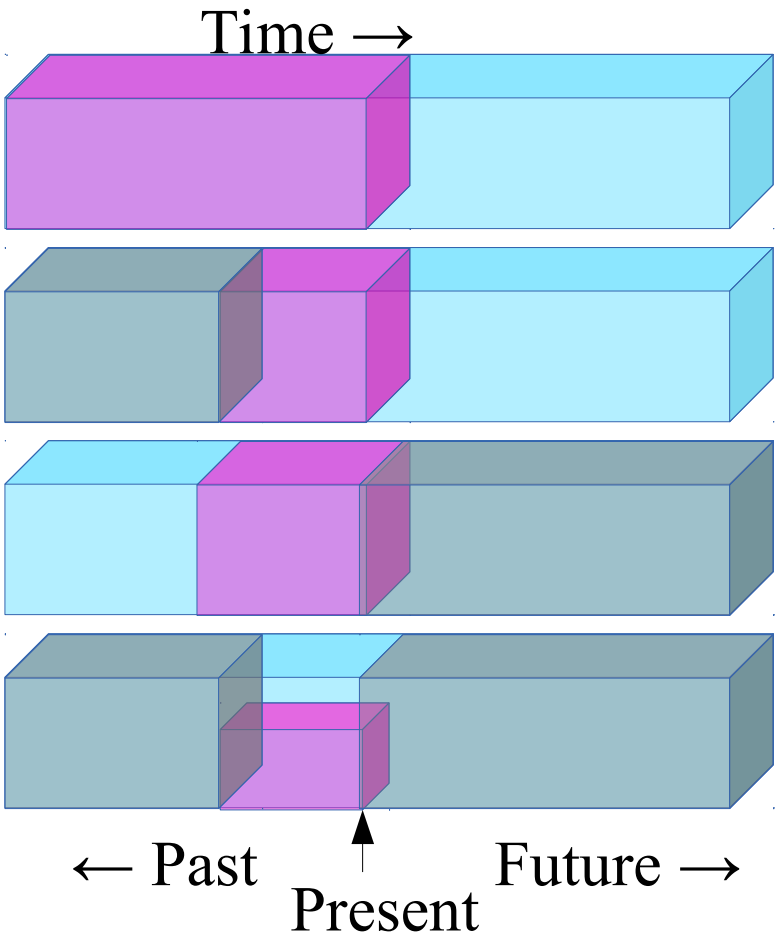
The Salvation: Self-Supervised Learning?

With massive amounts of data and very large networks

facebook
Artificial Intelligence Research

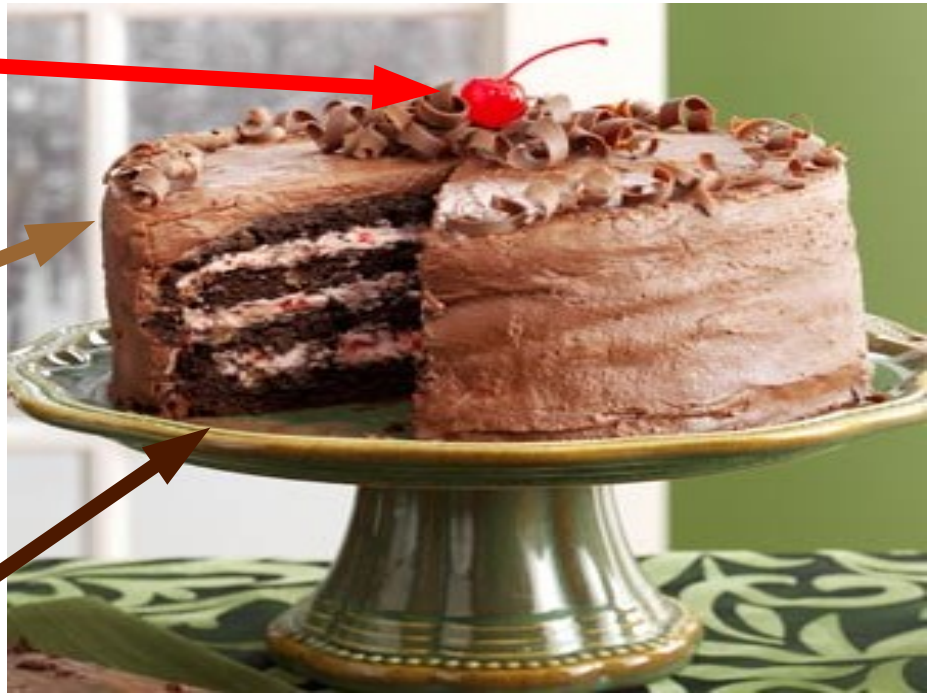
Self-Supervised Learning

- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the future from the recent past.
- Predict the past from the present.
- Predict the top from the bottom.
- Predict the occluded from the visible
- Pretend there is a part of the input you don't know and predict that.



How Much Information is the Machine Given during Learning?

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**



Self-Supervised Learning: Filling in the Blanks

Self-Supervised Learning works well for text

- **Word2vec**
  - [Mikolov 2013]
- **FastText**
  - [Joulin 2016]
- **BERT**
  - Bidirectional Encoder Representations from Transformers
  - [Devlin 2018]

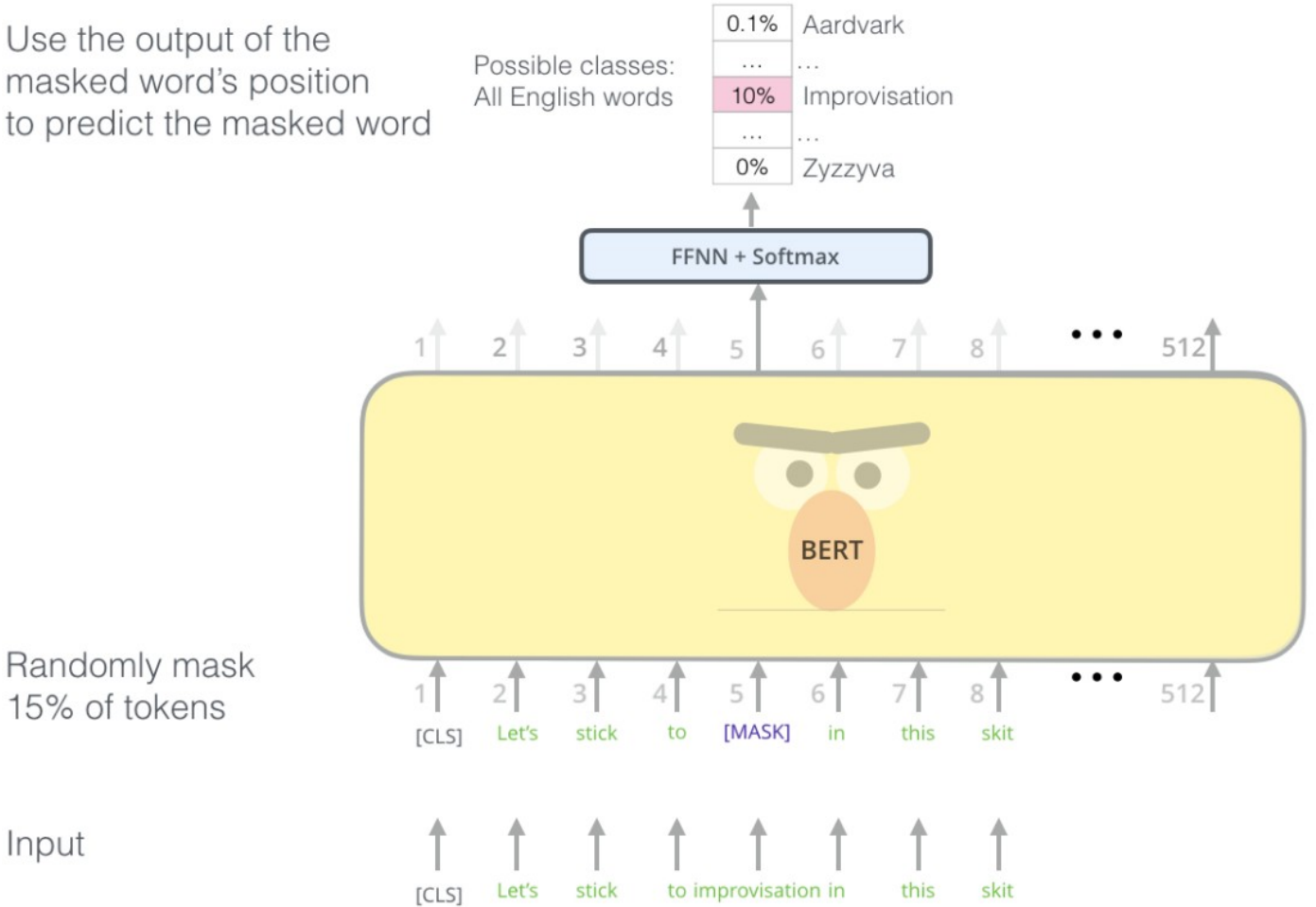
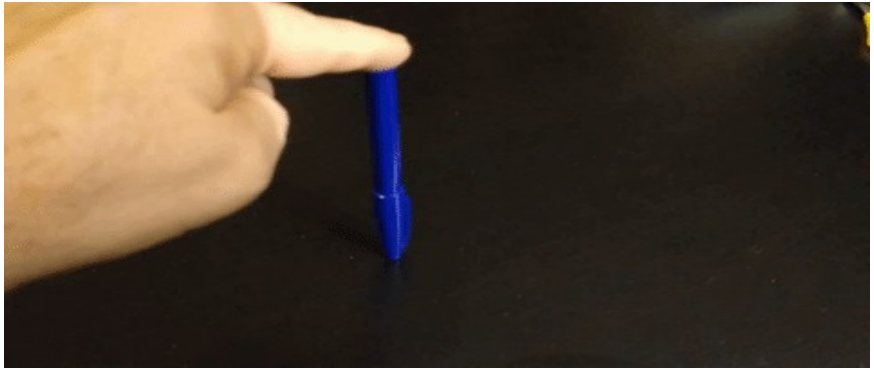


Figure credit: Jay Alammar http://jalammar.github.io/illustrated-bert/

But it doesn't really work for high-dim continuous signals

- **Video prediction:**
  - Multiple futures are possible.
  - Training a system to make a single prediction results in "blurry" results
  - the average of all the possible futures



The Next AI Revolution

With thanks
To
Alyosha Efros

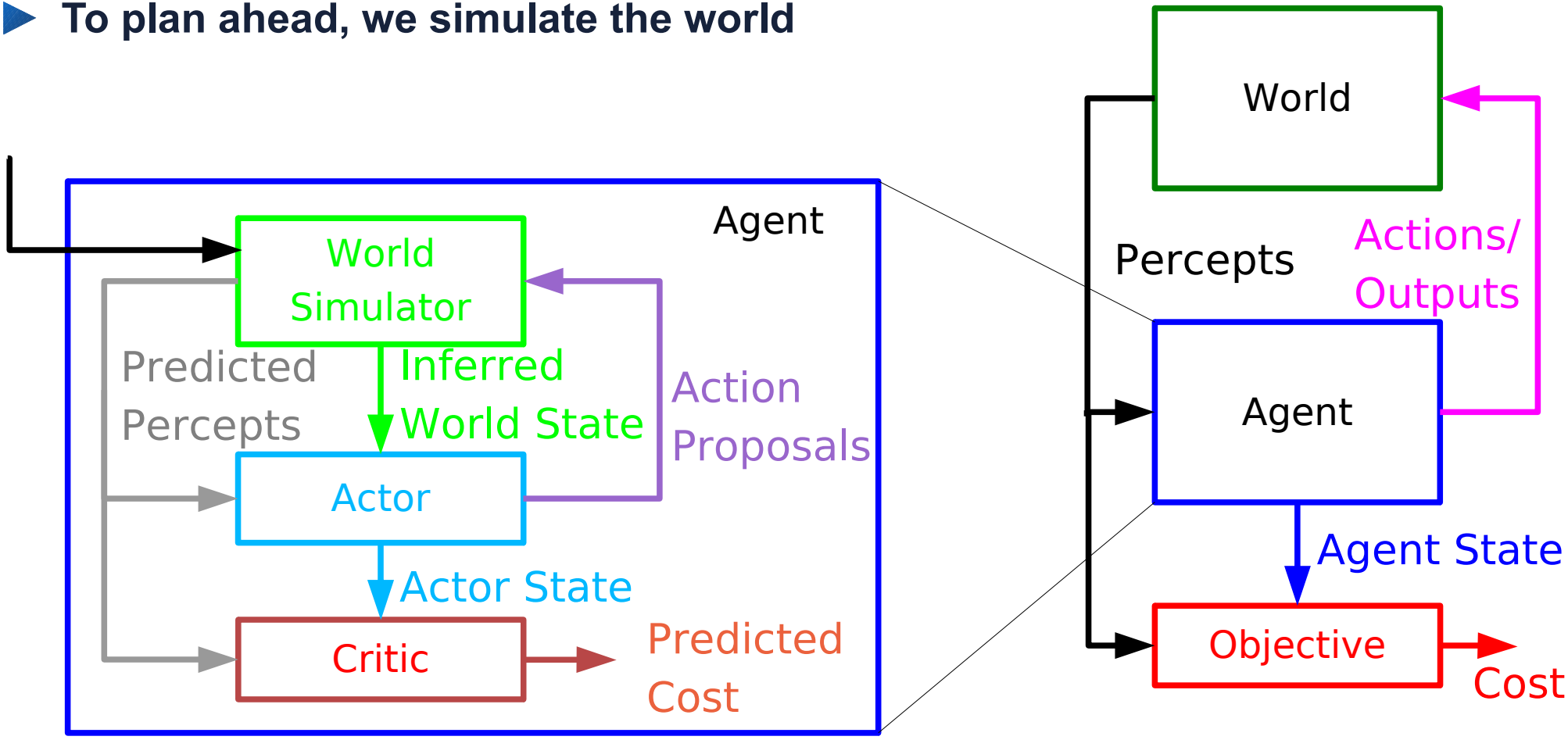
Learning Predictive Models of the World

Learning to predict, reason, and plan, Learning Common Sense.

facebook
Artificial Intelligence Research

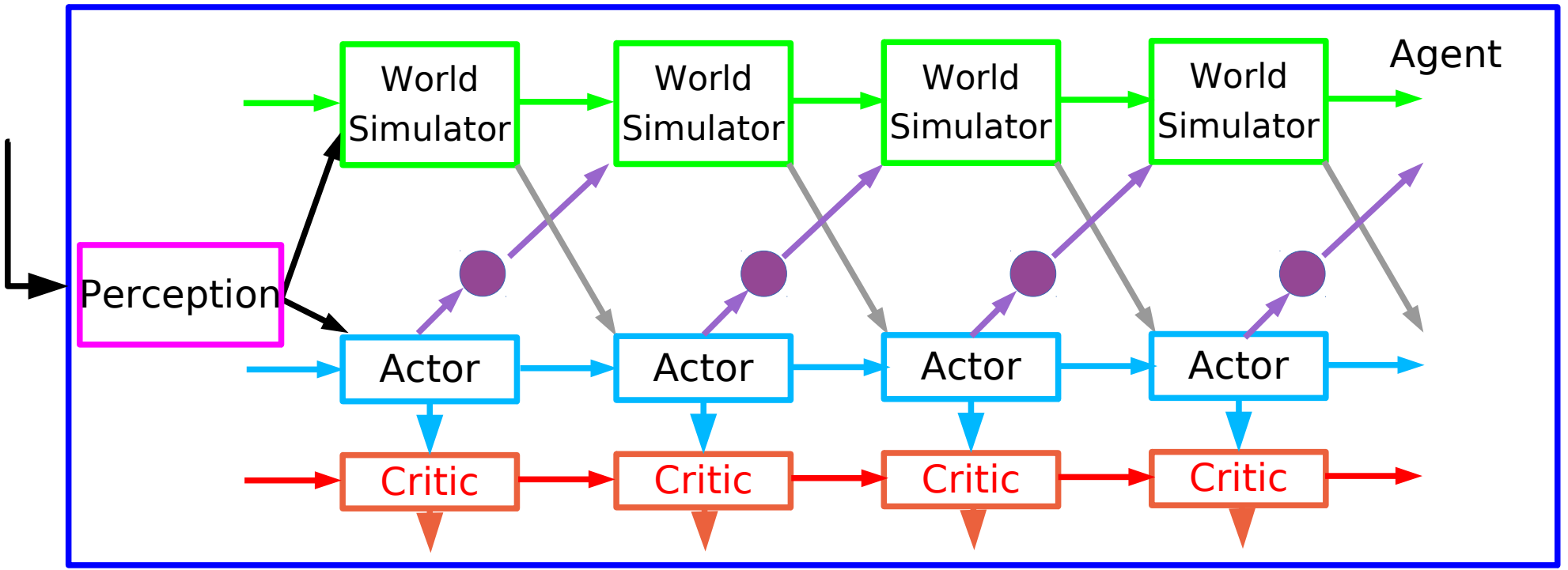
Planning Requires Prediction

- **To plan ahead, we simulate the world**



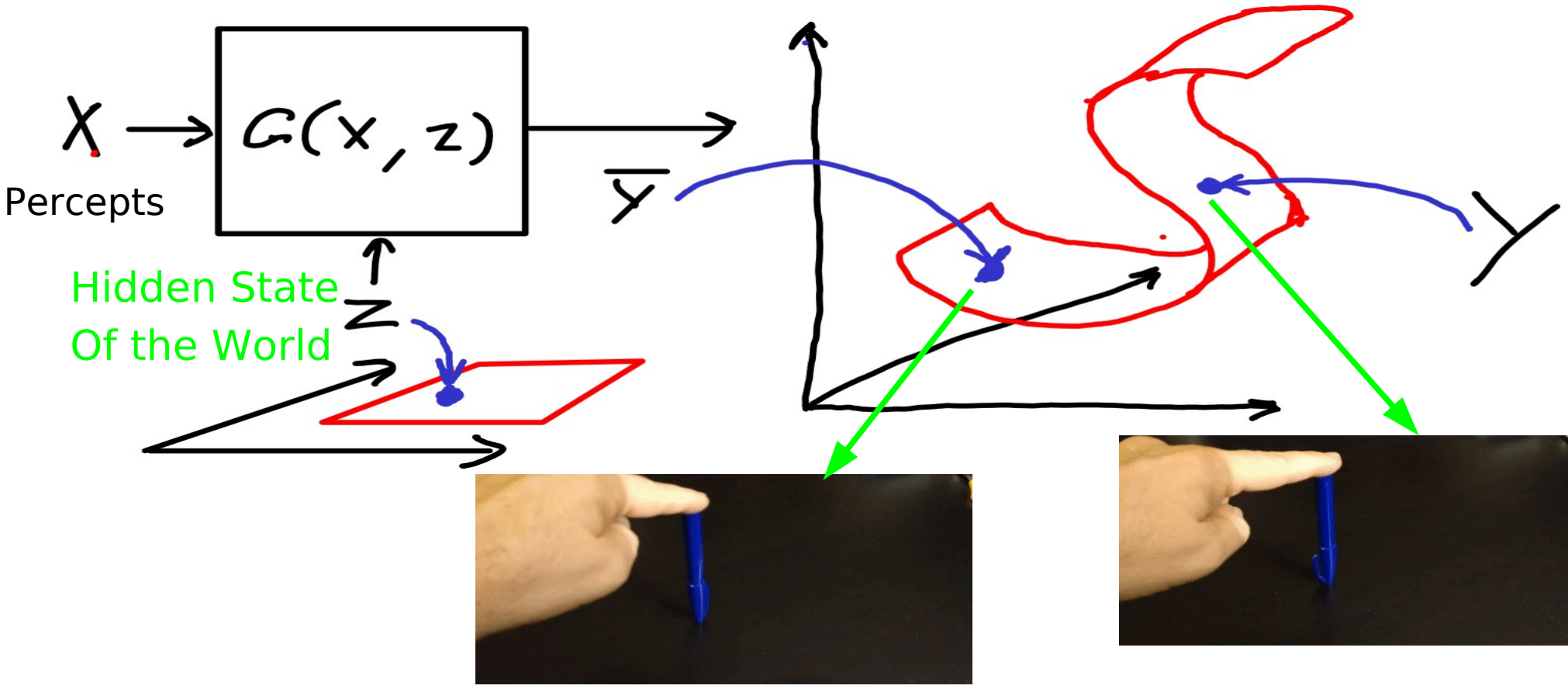
Training the Actor with Optimized Action Sequences

- **1. Find action sequence through optimization**
- **2. Use sequence as target to train the actor**
  - Over time we get a compact policy that requires no run-time optimization



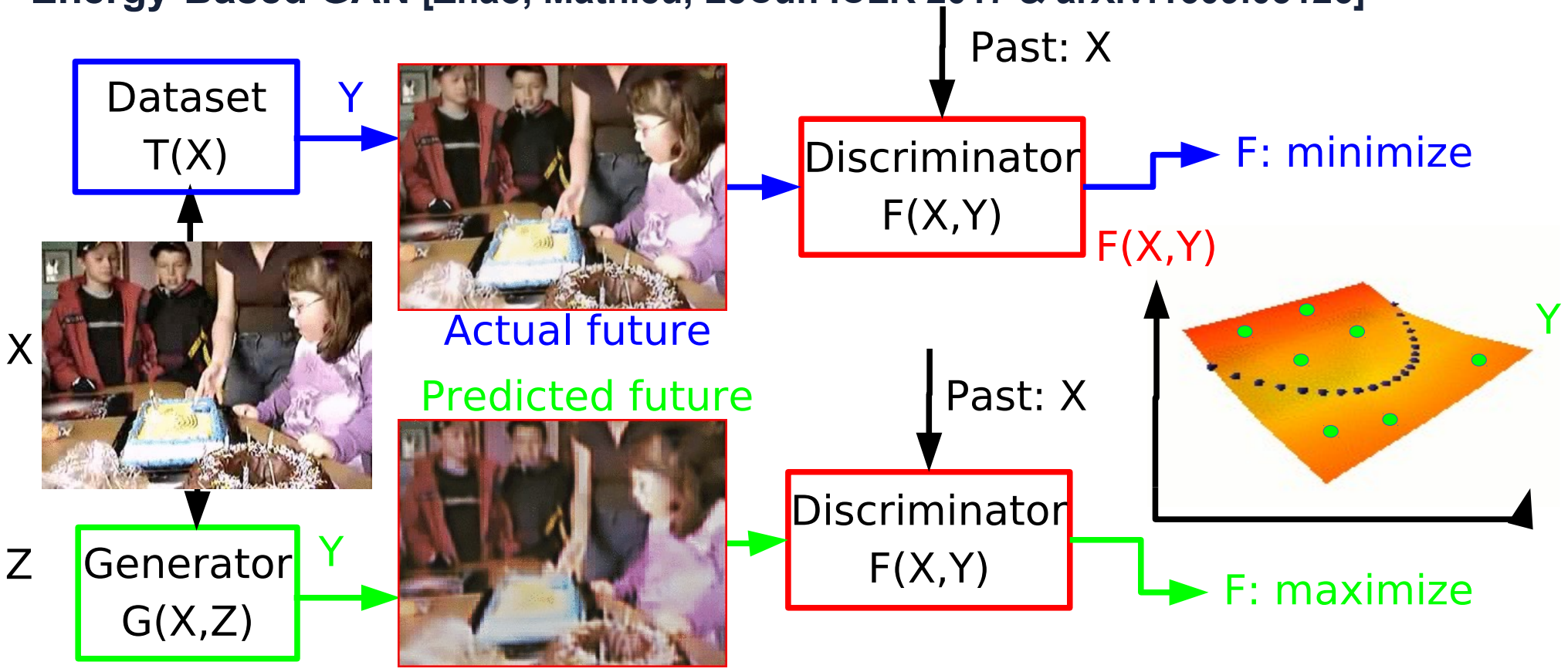
The Hard Part: Prediction Under Uncertainty

- **Invariant prediction: The training samples are merely representatives of a whole set of possible outputs (e.g. a manifold of outputs).**



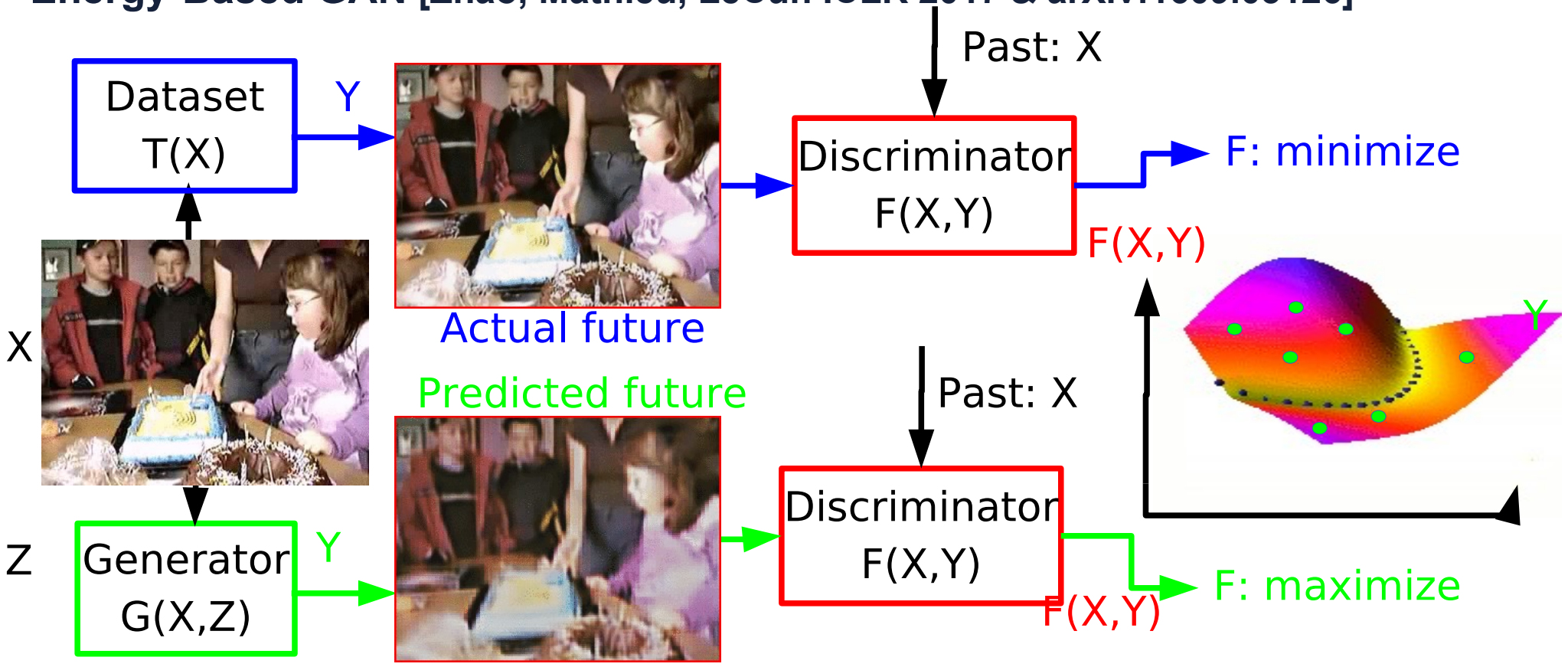
Adversarial Training: the key to prediction under uncertainty?

- **Generative Adversarial Networks (GAN) [Goodfellow et al. NIPS 2014],**
- **Energy-Based GAN [Zhao, Mathieu, LeCun ICLR 2017 & arXiv:1609.03126]**



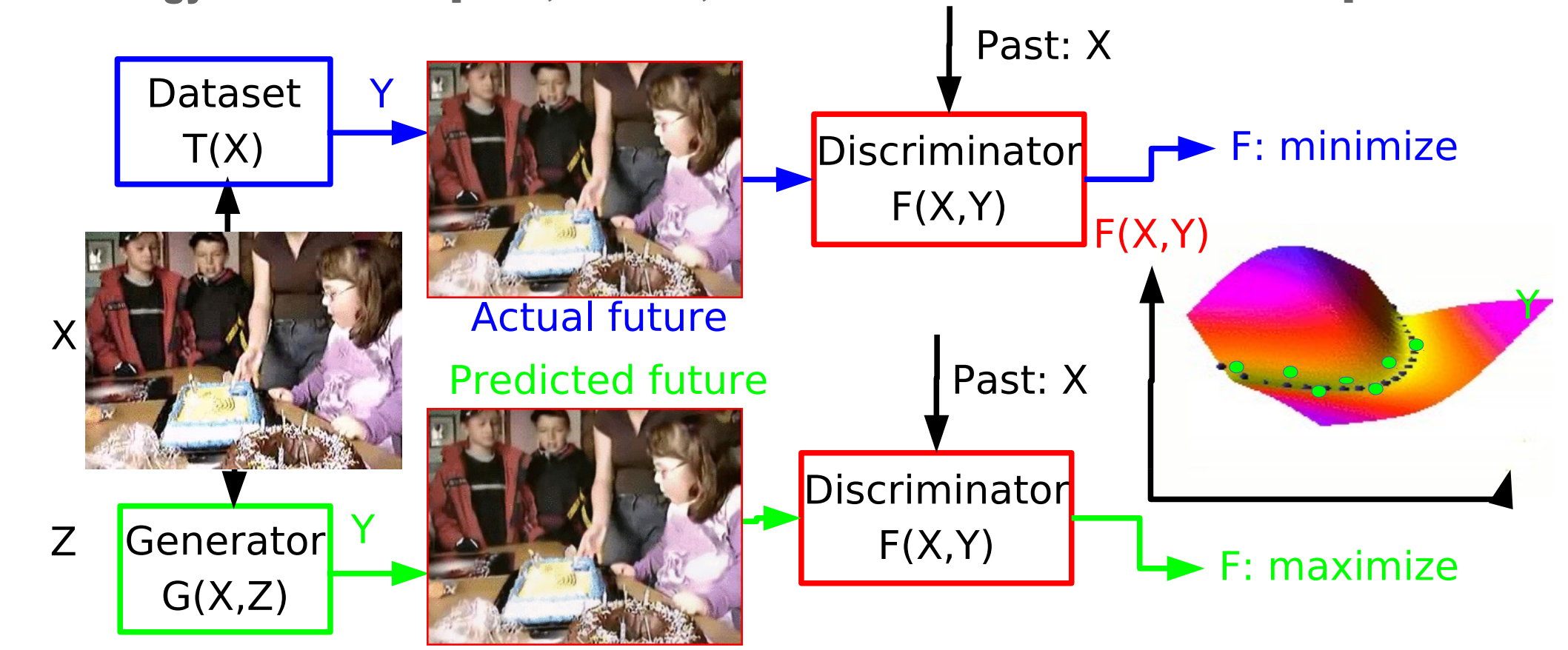
Adversarial Training: the key to prediction under uncertainty?

- **Generative Adversarial Networks (GAN) [Goodfellow et al. NIPS 2014],**
- **Energy-Based GAN [Zhao, Mathieu, LeCun ICLR 2017 & arXiv:1609.03126]**



Adversarial Training: the key to prediction under uncertainty?

- **Generative Adversarial Networks (GAN) [Goodfellow et al. NIPS 2014],**
- **Energy-Based GAN [Zhao, Mathieu, LeCun ICLR 2017 & arXiv:1609.03126]**



Faces “invented” by a GAN (Generative Adversarial Network)

- **Random vector → Generator Network → output image** [Goodfellow NIPS 2014]
  [Karras et al. ICLR 2018] (from NVIDIA)

Generative Adversarial Networks for Creation

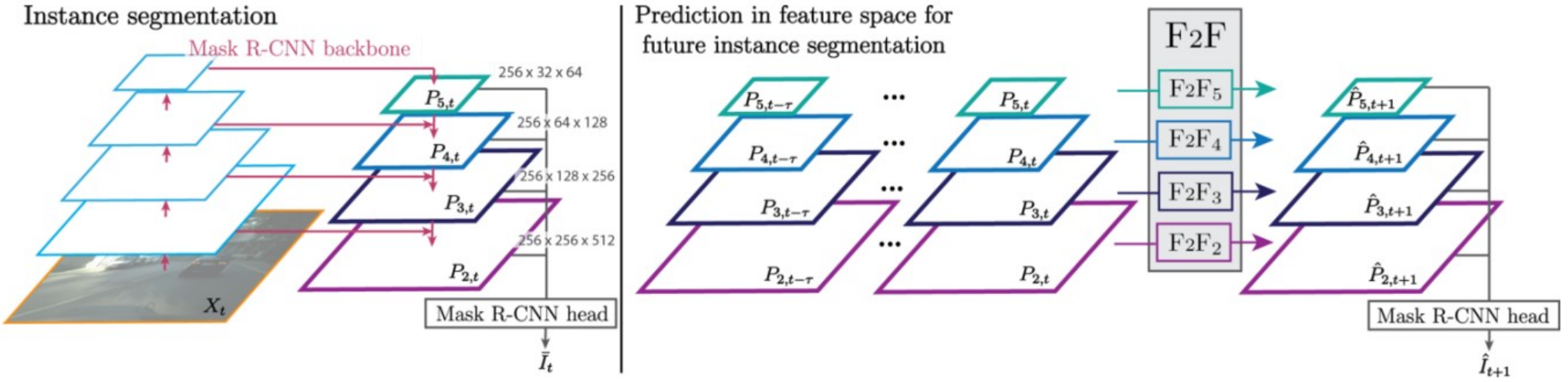
- **[Sbai 2017]**

Self-supervised Adversarial Learning for Video Prediction

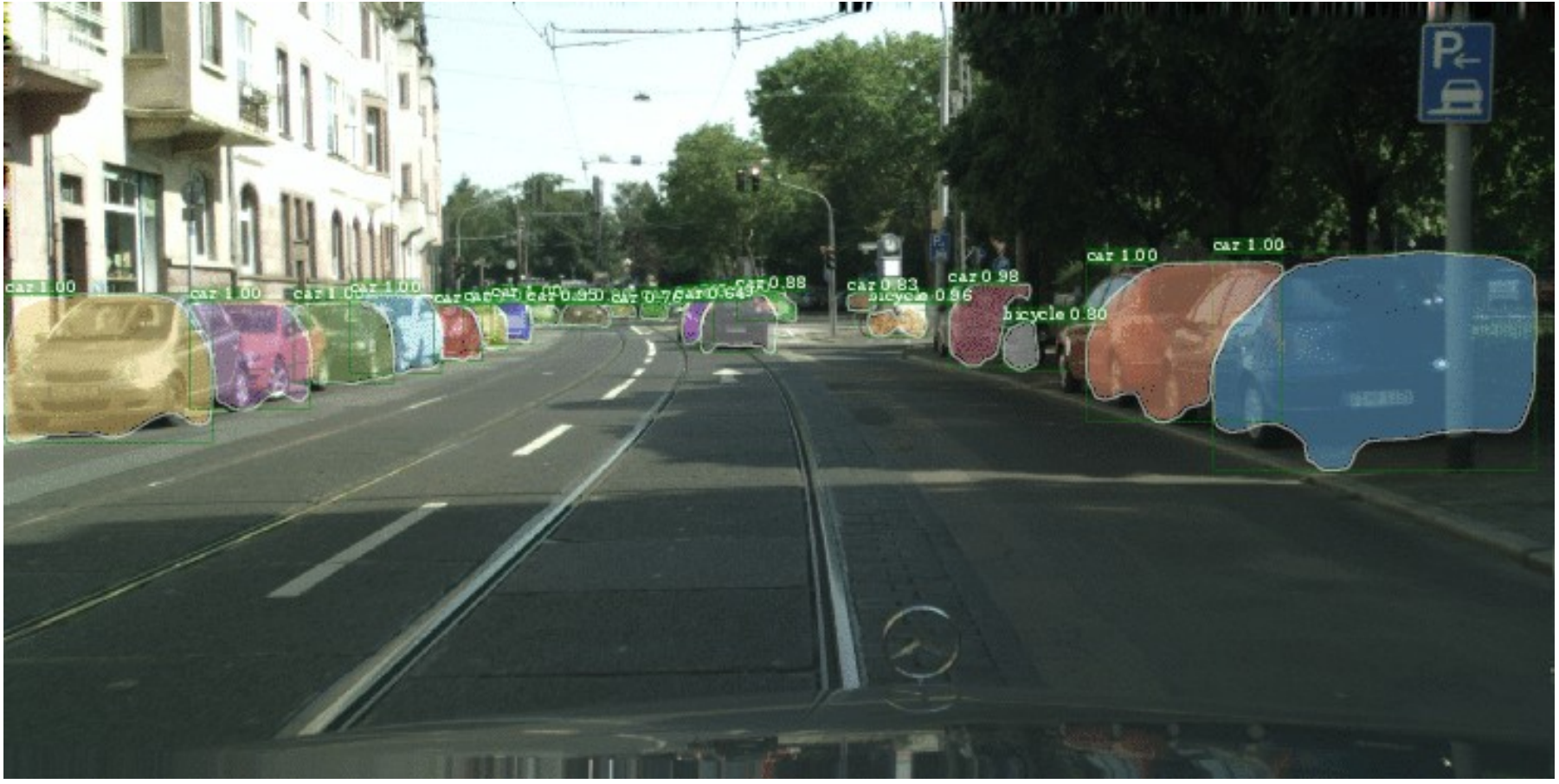
- **Our brains are "prediction machines"**
- **Can we train machines to predict the future?**
- **Some success with "adversarial training"**
  - [Mathieu, Couprie, LeCun arXiv:1511:05440]
- **But we are far from a complete solution.**

Predicting Instance Segmentation Maps

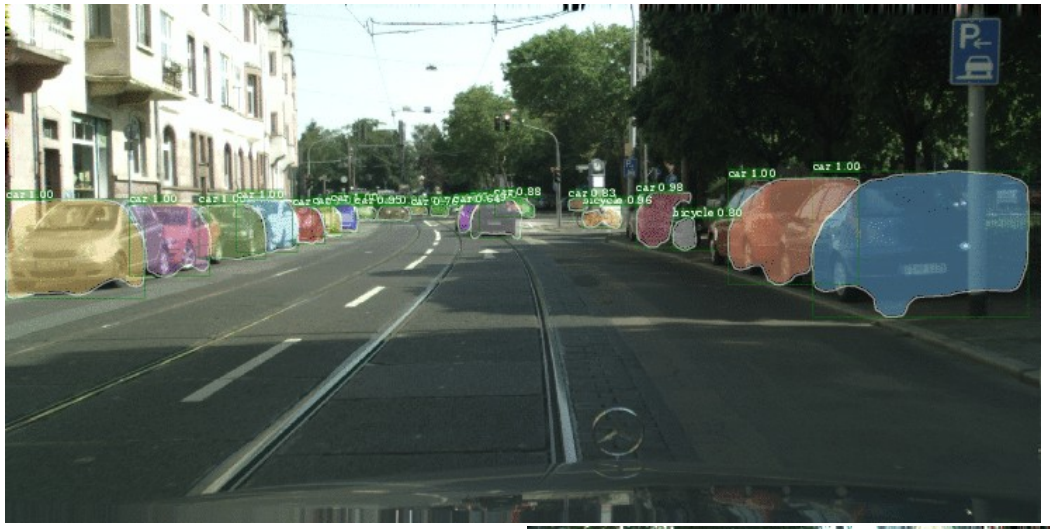
- [Luc, Couprie, LeCun, Verbeek ECCV 2018]
- Mask R-CNN Feature Pyramid Network backbone
- Trained for instance segmentation on COCO
- Separate predictors for each feature level

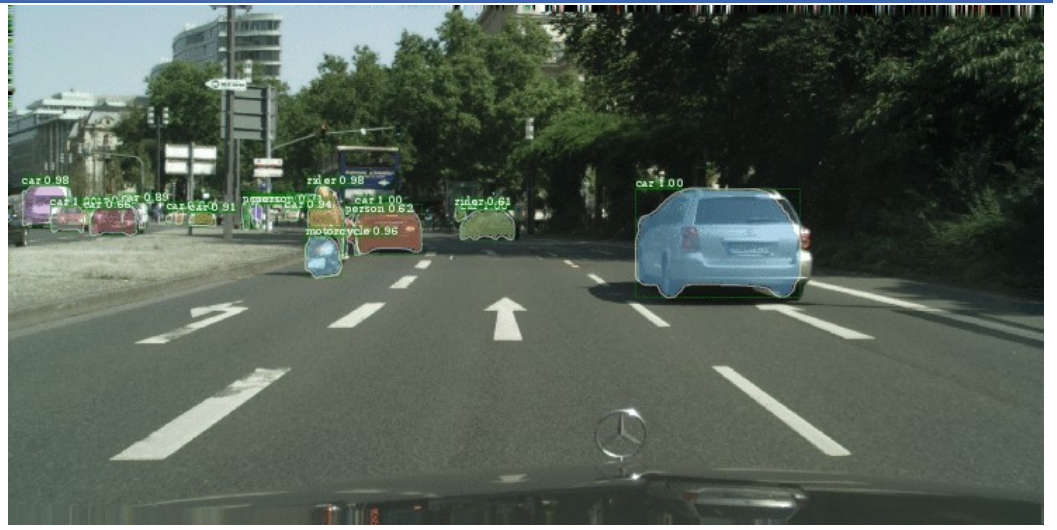


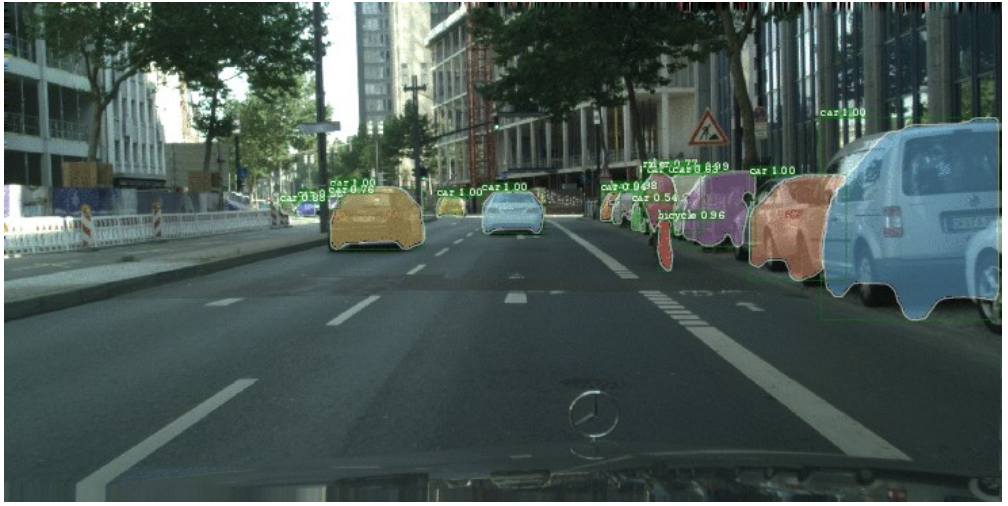
Predictions



Long-term predictions (10 frames, 1.8 seconds)







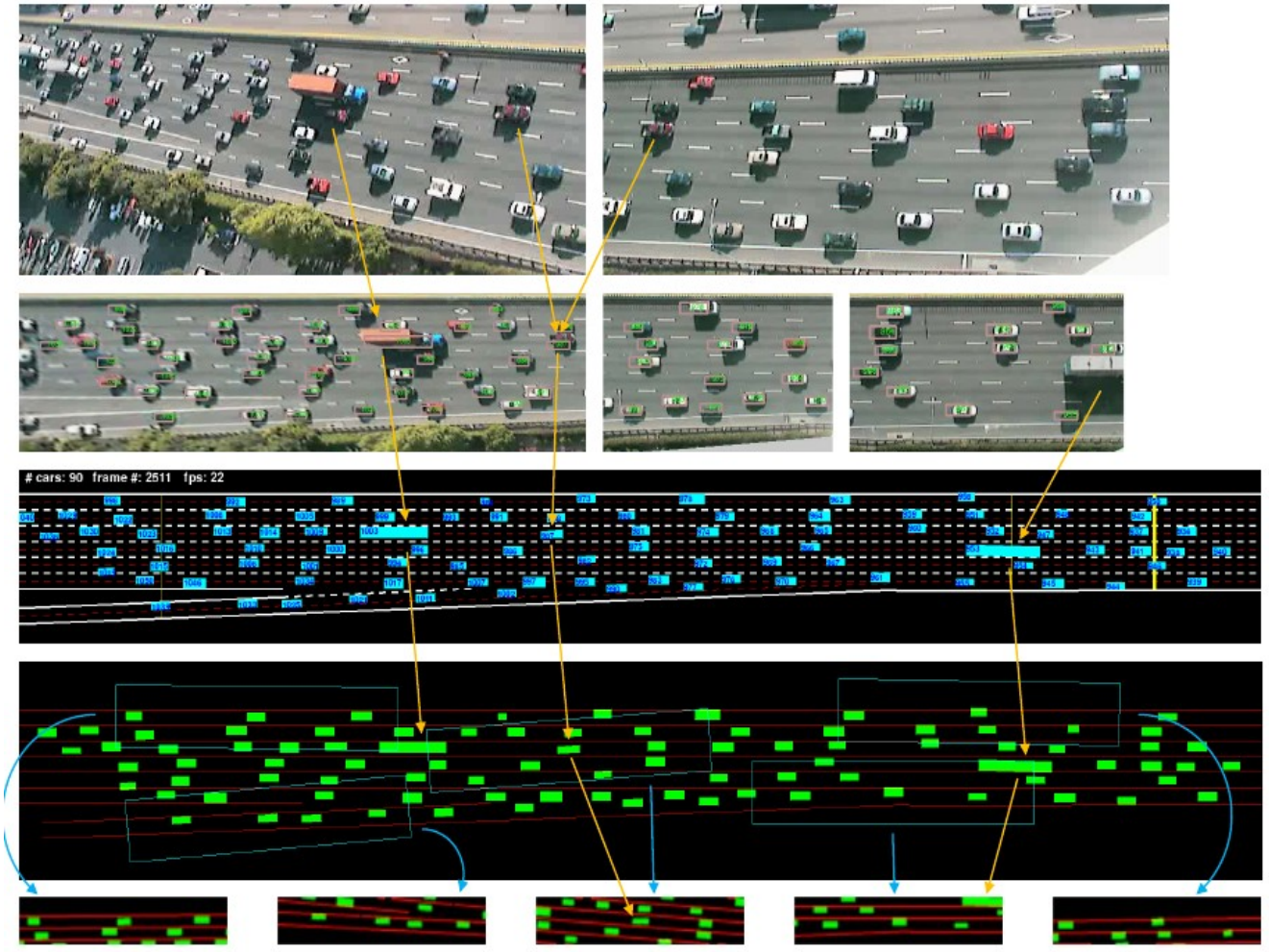
Self-Supervised Forward Models For Control

Learning motor skills with no interaction with the real world
[Henaff, Canziani, LeCun ICLR 2019]
[Henaff, Zhao, LeCun ArXiv:1711.04994]
[Henaff, Whitney, LeCun Arxiv:1705.07177]

facebook
Artificial Intelligence Research

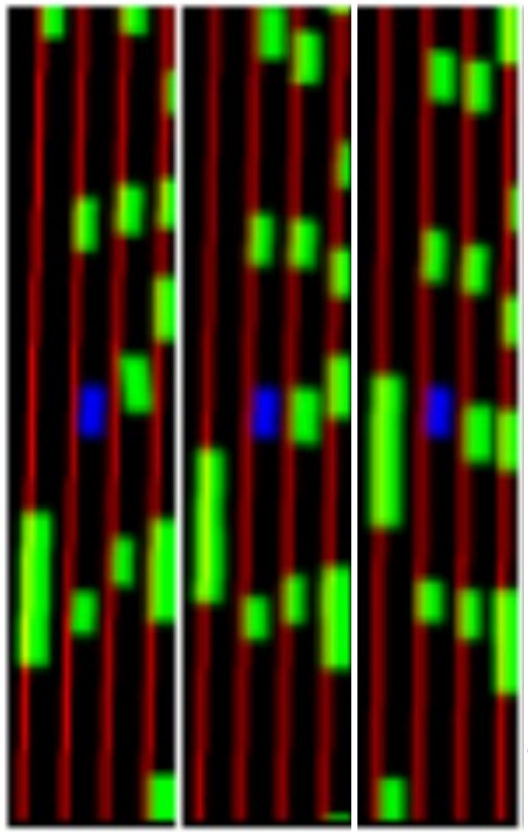
Using Forward Models to Plan (and to learn to drive)

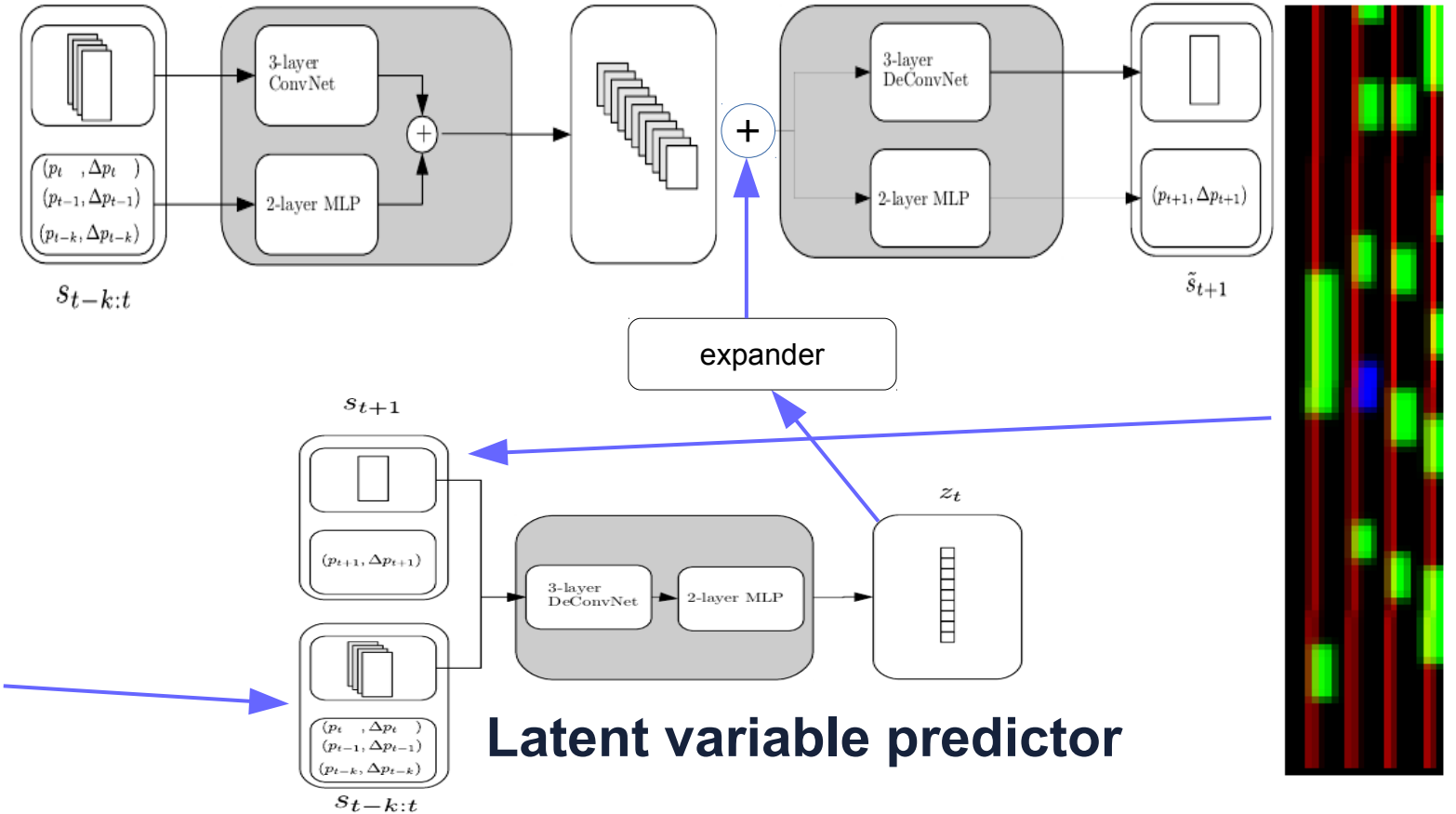
- **Overhead camera on highway.**
  - Vehicles are tracked
- **A "state" is a pixel representation of a rectangular window centered around each car.**
- **Forward model is trained to predict how every car moves relative to the central car.**
  - steering and acceleration are computed



Forward Model Architecture

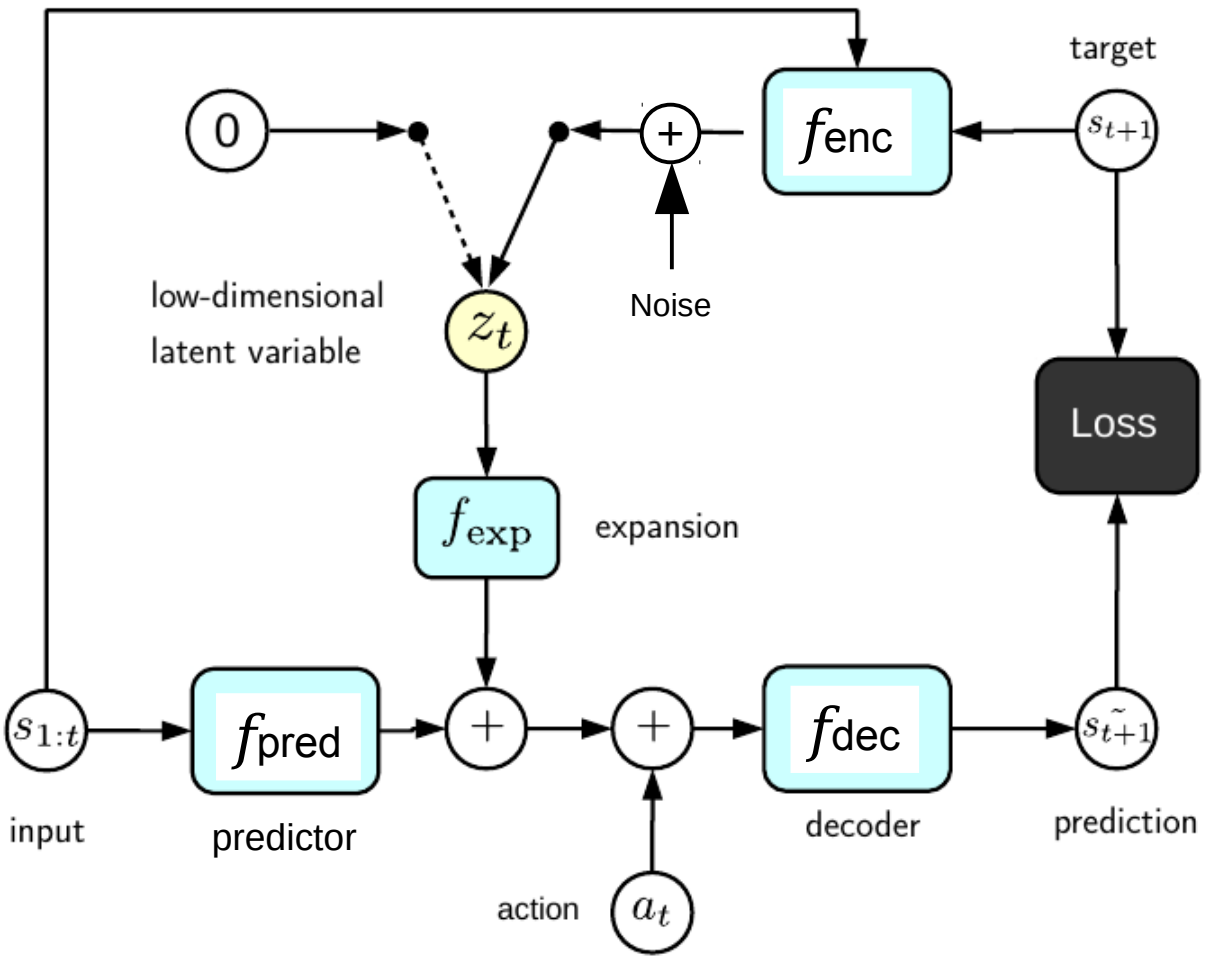
- **Architecture:**



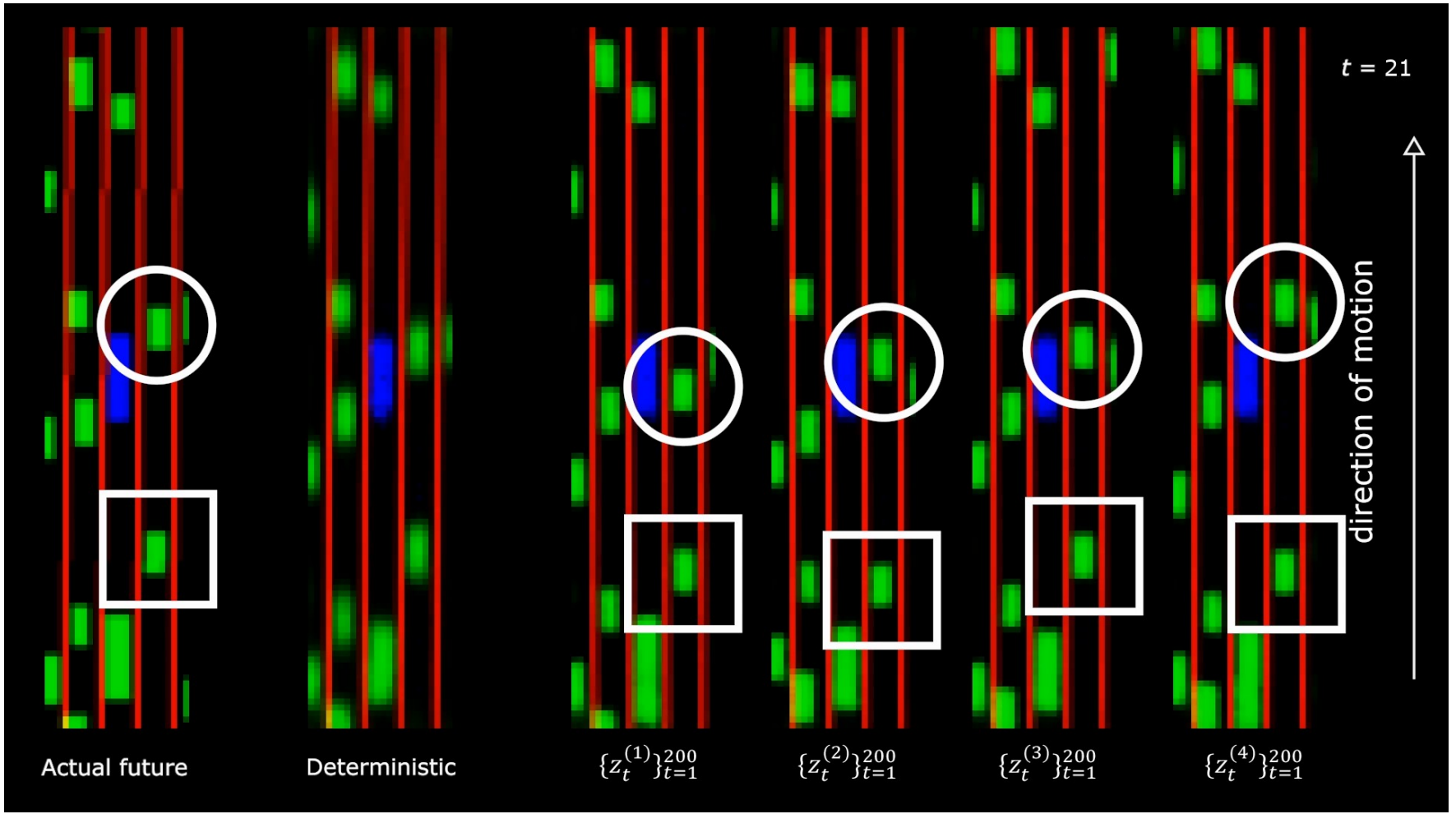


Stochastic Forward Modeling: regularized latent variable model

- **Latent variable is predicted from the target**
- **The latent variable is set to zero half the time during training (drop out) and corrupted with noise**
- **The model predicts as much as it can without the latent var.**
- **The latent var corrects the residual error.**

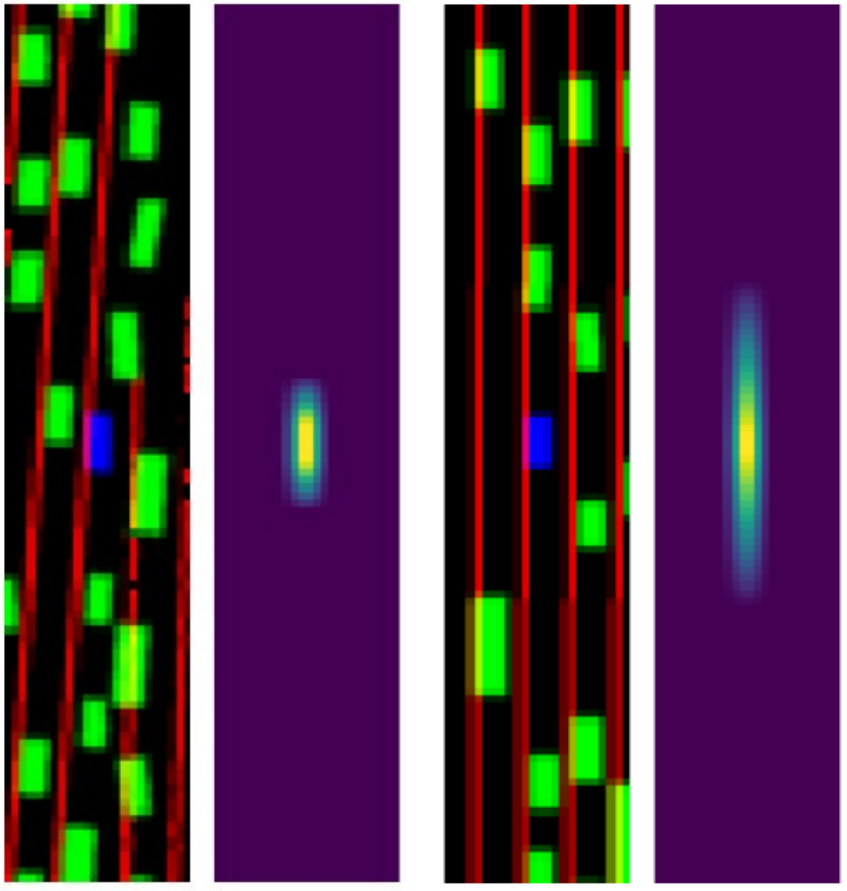


Actual, Deterministic, VAE+Dropout Predictor/encoder



Cost optimized for Planning & Policy Learning

- **Differentiable cost function**
  - Increases as car deviates from lane
  - Increases as car gets too close to other cars nearby in a speed-dependent way
- **Uncertainty cost:**
  - Increases when the costs from multiple predictions (obtained through sampling of drop-out) have high variance.
  - Prevents the system from exploring unknown/unpredictable configurations that may have low cost.

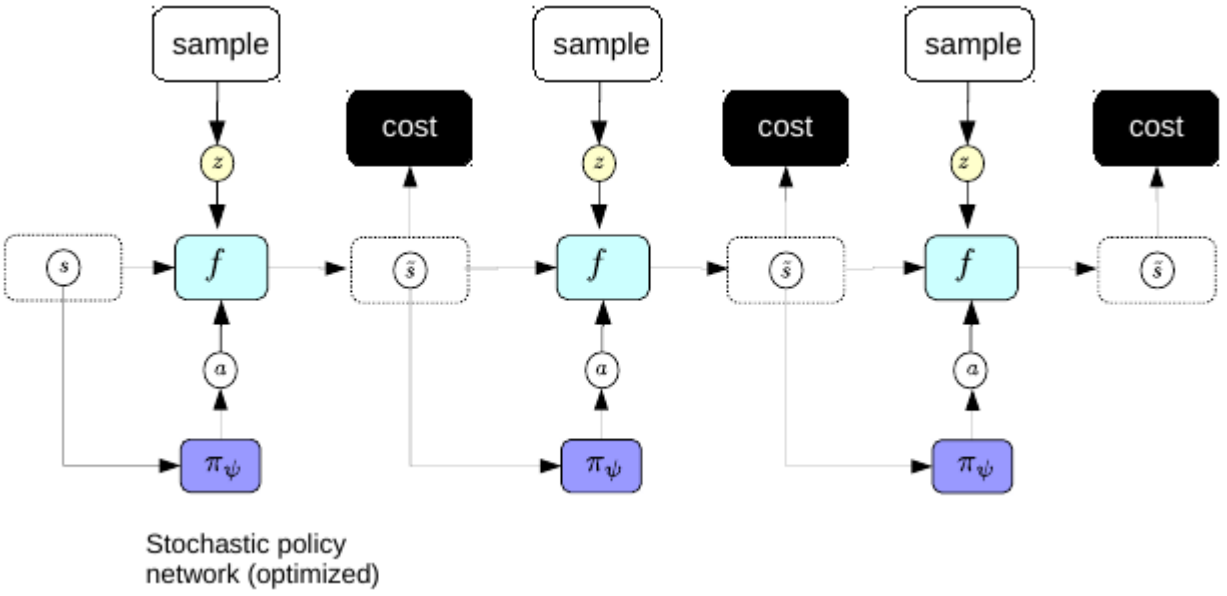


(a) 19.8 km/h (b) 50.3 km/h

Learning to Drive by Simulating it in your Head

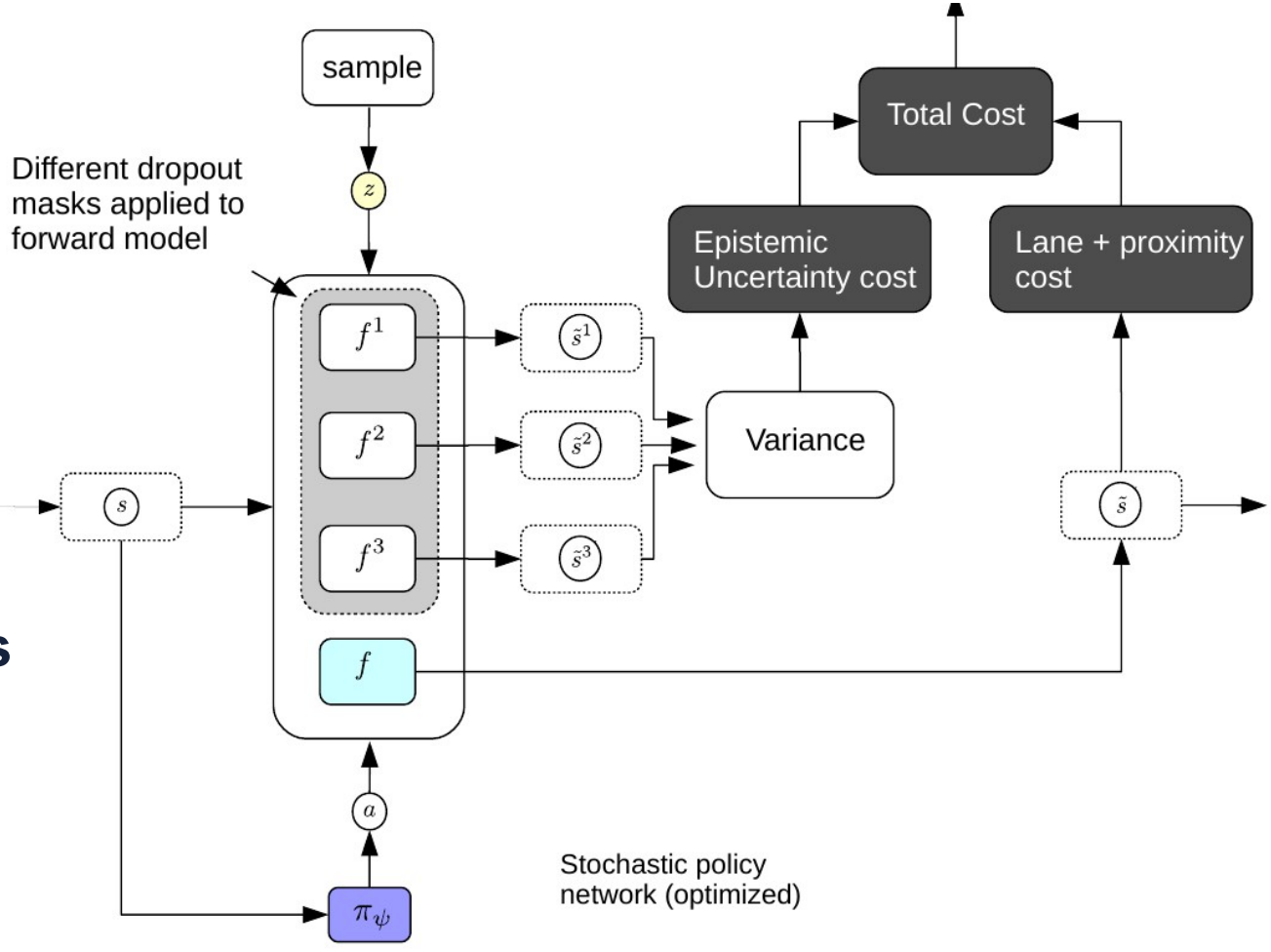
- Feed initial state
- Sample latent variable sequences of length 20
- Run the forward model with these sequences
- Backpropagate gradient of cost to train a policy network.
- Iterate

- No need for planning at run time.



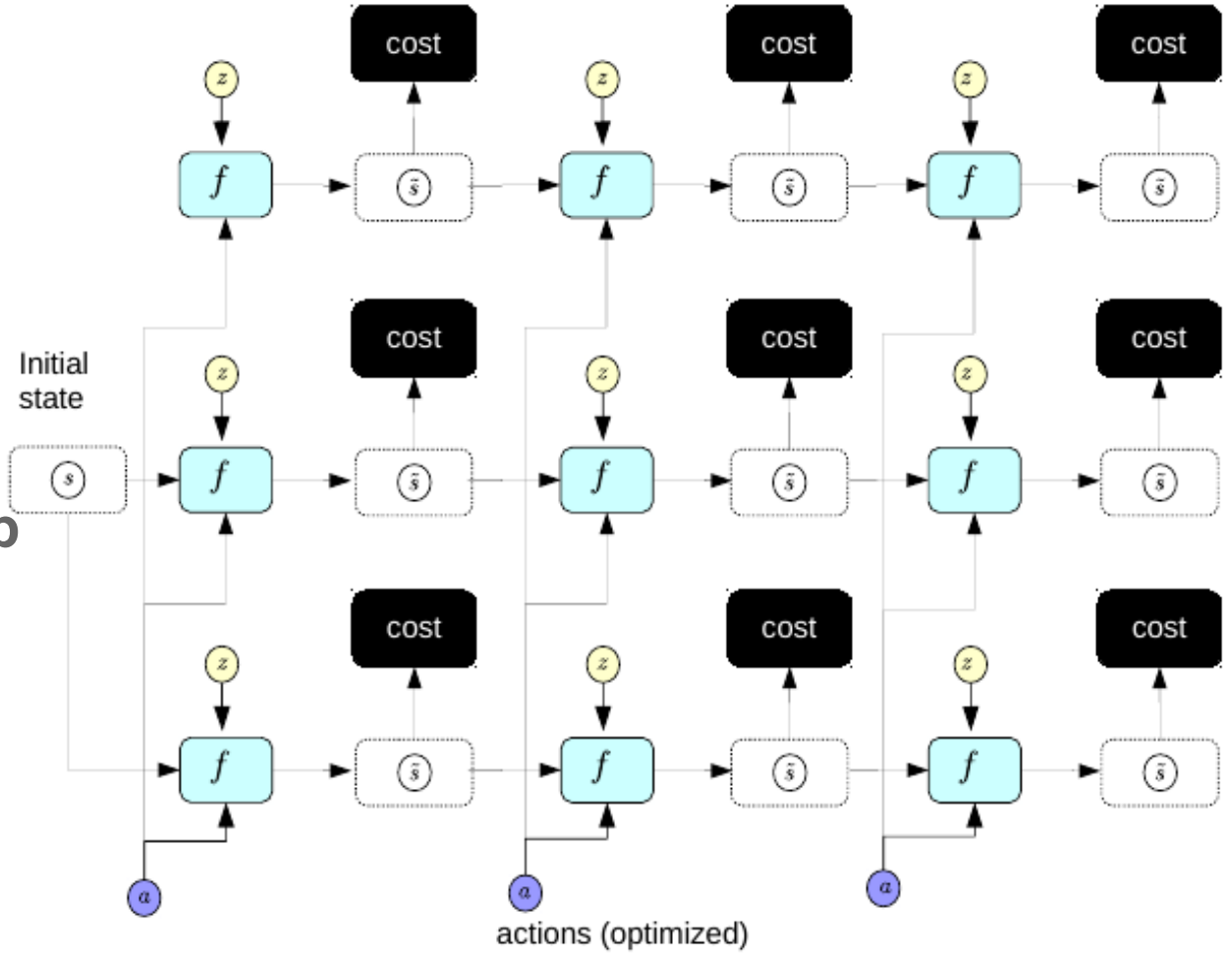
Adding an Uncertainty Cost (doesn't work without it)

- **Estimates epistemic uncertainty**
- **Samples multiple drop-puts in forward model**
- **Computes variance of predictions (differentiably)**
- **Train the policy network to minimize the lane&proximity cost plus the uncertainty cost.**
- **Avoids unpredictable outcomes**

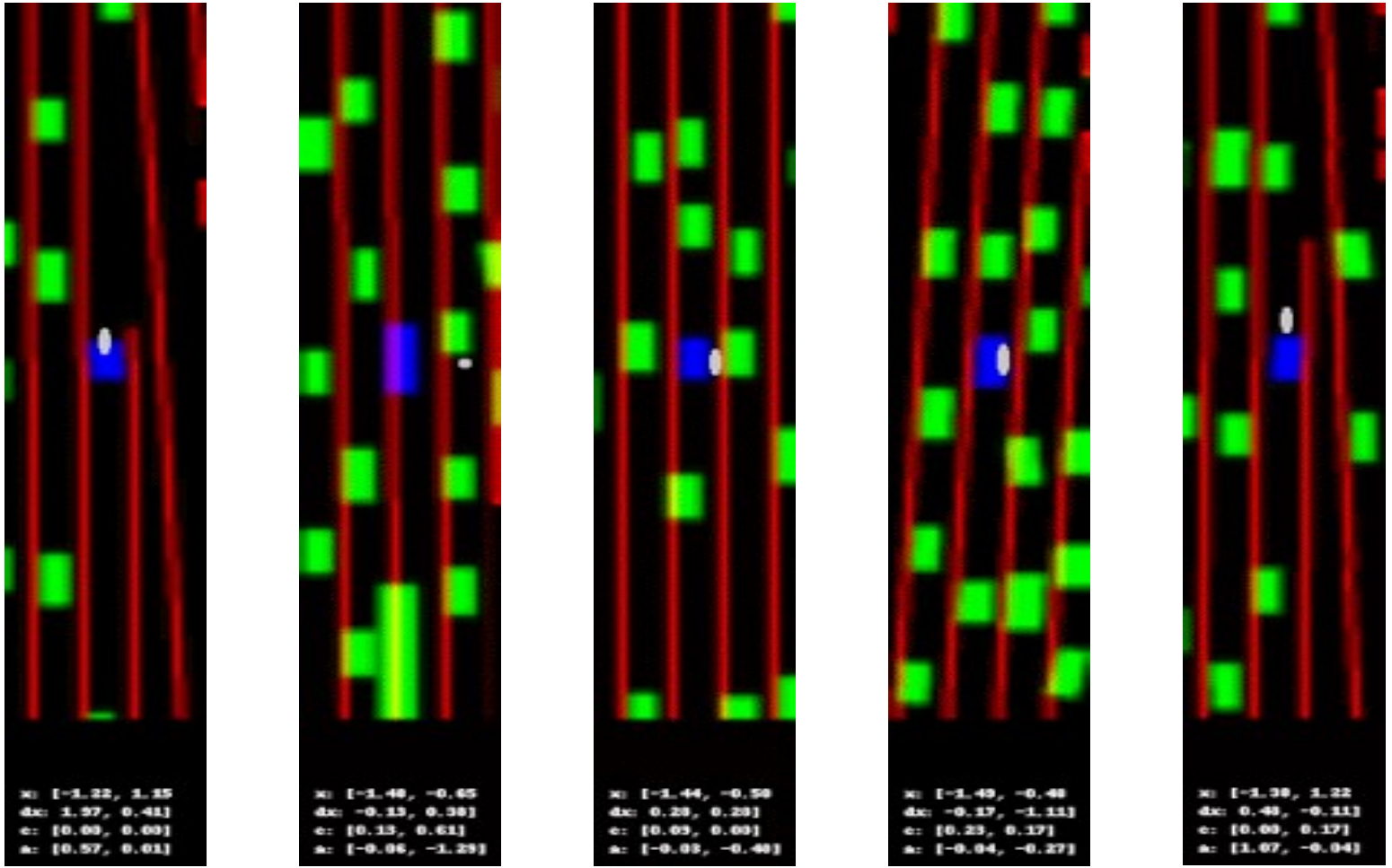


Approach 1: Planning with Stochastic Model Predictive Control

- Feed initial state
- Sample 10 latent variable sequences of length 20
- Run the forward model with these sequences
- Optimize the action sequence to minimize the average cost (by backprop and gradient descent)
- Take the first action
- Iterate
  - receding horizon planning

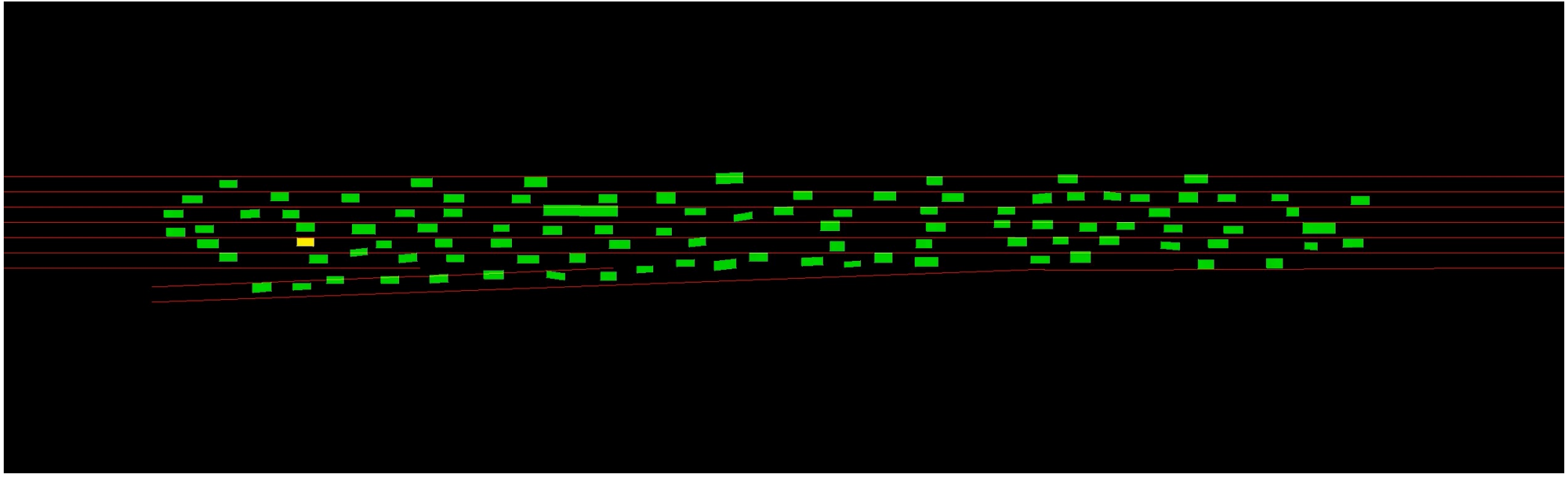


Driving an Invisible Car in “Real” Traffic



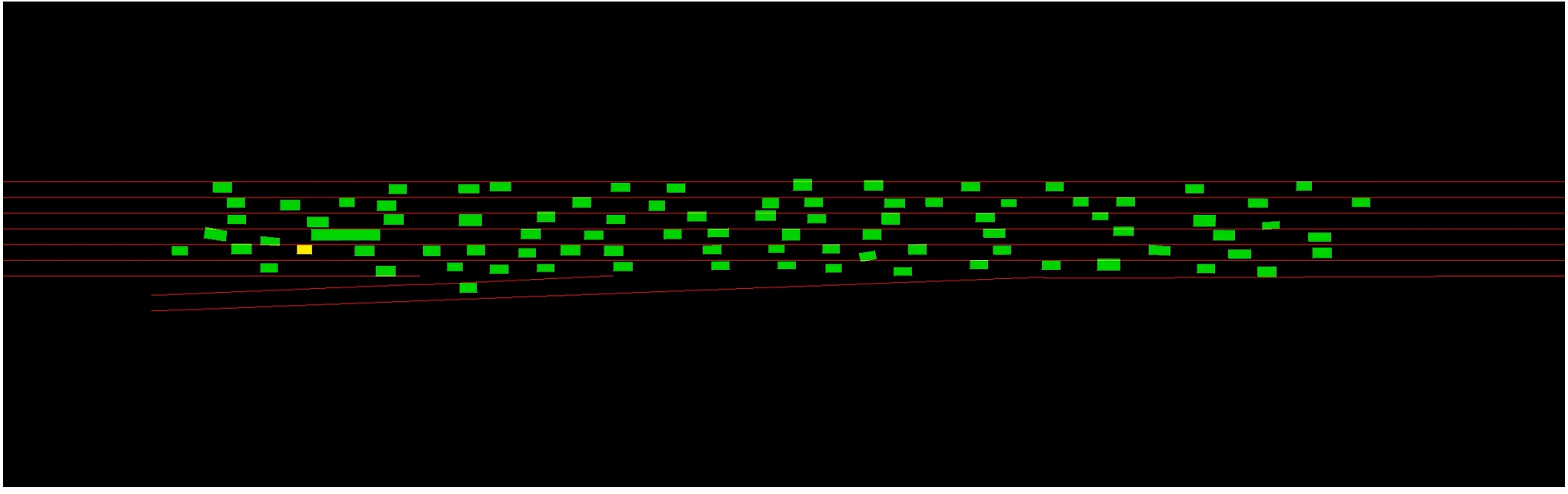
Driving!

- Yellow: real car
- Blue: bot-driven car



Driving!

- Yellow: real car
- Blue: bot-driven car



Lessons learned

- ***1 Model-free Reinforcement Learning is too slow in the real world***
  - *Requires too many "blind" interactions"*
- ***2: Regularized latent-variable energy-based models are a good way to learn features in an unsupervised fashion.***
- ***3: More generally, Self-Supervised learning is the future of DL***
  - *Networks will be much larger than today*
  - *We have unlimited amounts of data to train them*
  - *They will have sparse activation*
  - *Can electronic hardware take advantage of sparse activations?*
- ***4: Learning Models of the world accelerate learning of motor tasks***
- ***Prediction is the essence of intelligence***

When will the "True AI" revolution occur?

- **We won't have household robots and good digital friends (or assistants) until machines acquire common sense.**
- **This won't happen until we get machines to learn predictive world models**
- **Discovering the principles of it may take 2, 5, 10 or 20 years.**
- **Developing practical technology from it may take another 10 years**
  - The emergence of "true AI" will not be a singular event as in Hollywood movies.
- **We work on the assumption that there is "simple" principle (and a few algorithms) for AI, as there is for flight (aerodynamics) or engines (thermodynamics).**

What will super-intelligent AGI be like?

- **Will the "singularity" happen?**
  - No. Nothing is exponential forever
- **Future AI systems will have emotions and moral values**
  - How to align AI values with human values?

- **Will it take our jobs?**
  - No. But our jobs will change. Human experience will have high value.
  - it will empower humanity by amplifying our intelligence

- **Will it want to take over the world?**
  - No, the desire to dominate is not correlated with intelligence but with testosterone

Authentic human experience > material goods

- **Material goods:**
  - BlueRay player: $47
  - Handmade ceramic bowl: $750
- **Mozart's opera Die Zauberflöte**
  - Downloadable recording: $7
  - Ticket at the NYC Met: up to $807
- **Bright future for jazz musicians and artisans?**

Samsung

Samsung Smart Curved Design Blu-Ray Disc 1080p Player With Wired Ethernet Content Streaming Manufacturer Refurbished

19 customer reviews | 8 answered questions

Price: $46.88 & FREE Shipping

Mozart: Die Zauberflöte

Wiener Philharmoniker

January 1, 2012

19 customer reviews

Start your 30-day free trial of Unlimited to Prime pricing.

See all 50 formats and editions

Streaming Unlimited	MP3 $6.99	Audio CD $8.98

CENTER ORCHESTRA Row B-EE Email delivery by: 09/26/17	QTY 5	$751.00 ea.	BUY
ORCH Row B-O Email delivery by: 09/26/17	QTY 4	$772.00 ea.	BUY
CENTER ORCHESTRA Row A-DD Email delivery by: 09/26/17	QTY 5	$786.00 ea.	BUY
ORCH Row A-N Email delivery by: 09/26/17	QTY 4	$807.00 ea.	BUY

Scallop Bowl

$750.00 USD

Dimensions: 14"w x 7"h

H3-40

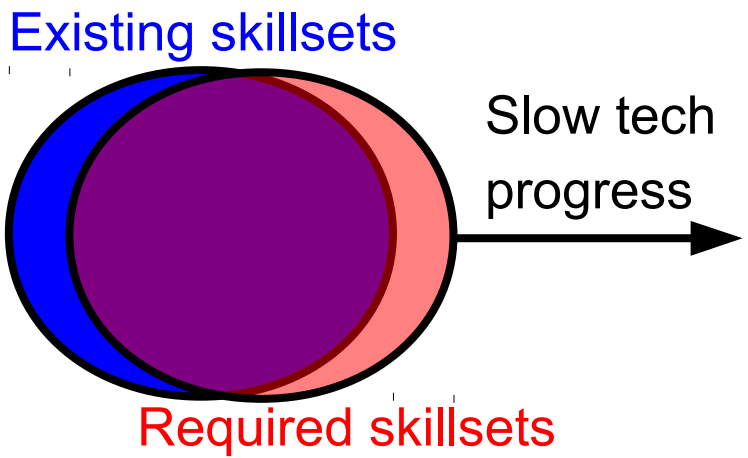
ADD TO CART

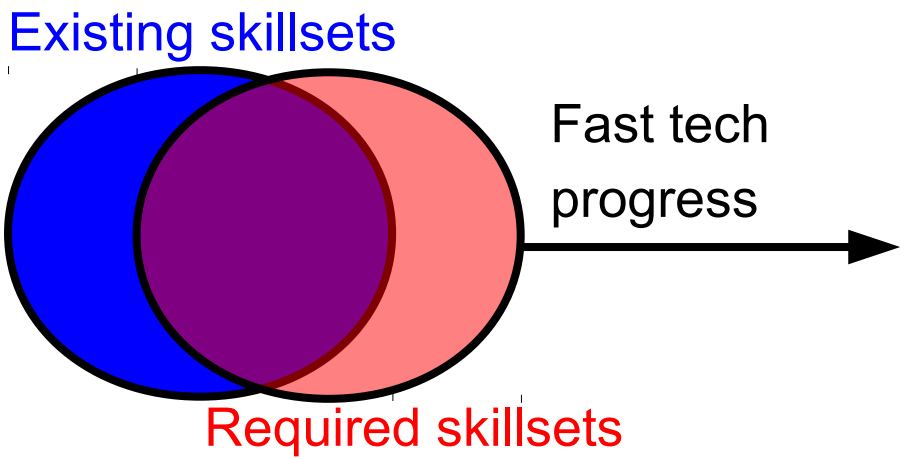
ADD TO REGISTRY

Tweet Like 0

AI is a "General Purpose Technology" (GPT)

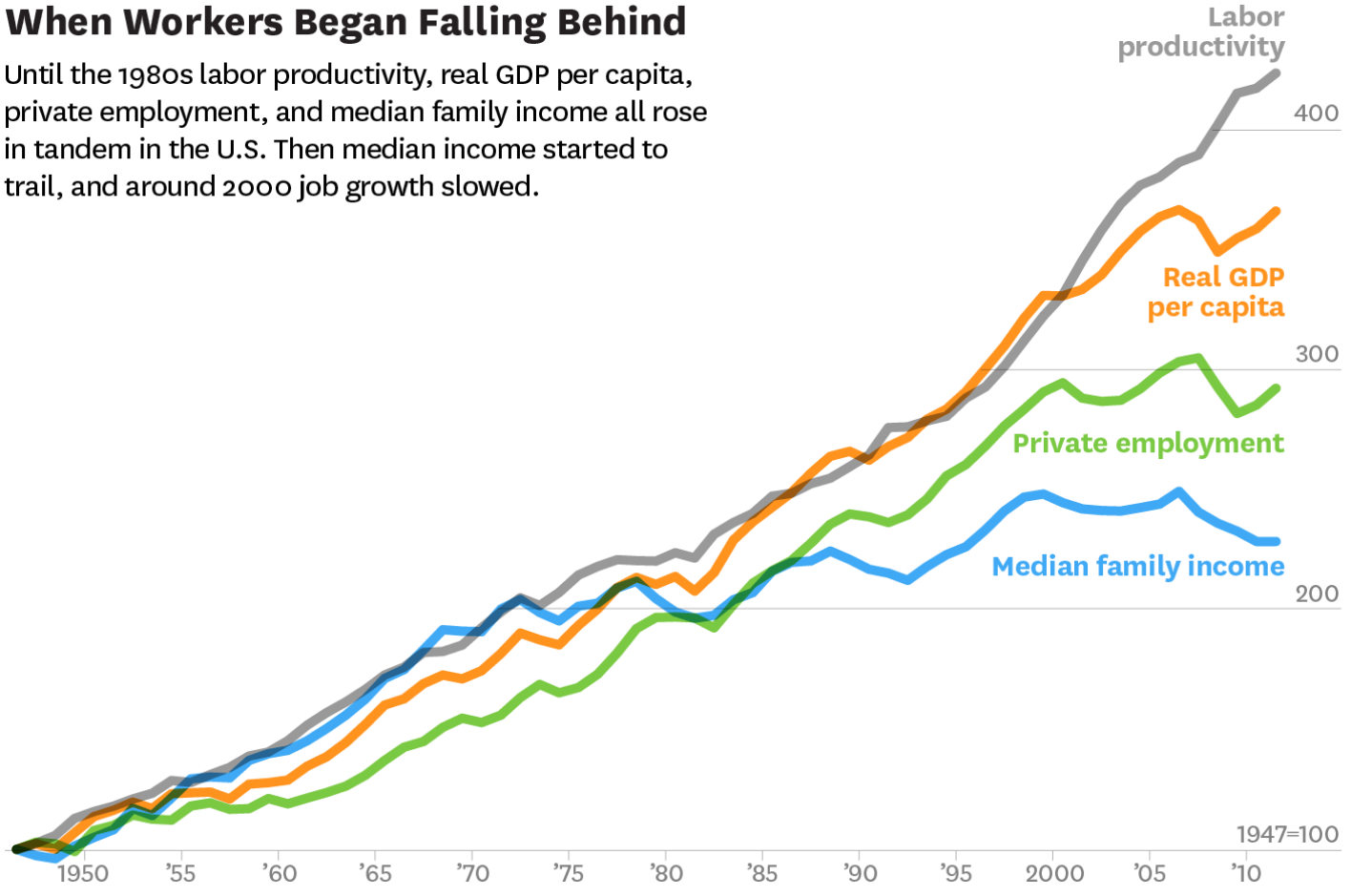
- **GPT: steam engine, electricity, computer...**
  - [Bresnahan & Trajtenberg 1995] "GPTs 'Engines of growth'?". J. Econometrics.
- **AI will affect many sector of the economy**
- **But it will take 10 or 20 years before we see the effect on productivity**
- **AI/automation → job displacement → technological unemployment**
- **Technology deployment is limited by how fast workers can train for it**





Does technological progress cause income inequality?

- **Erik Brynjolfsson & Andrew McAfee (MIT).**
- **Perhaps, but the fix is progressive taxation.**



Thank you

facebook
Artificial Intelligence Research