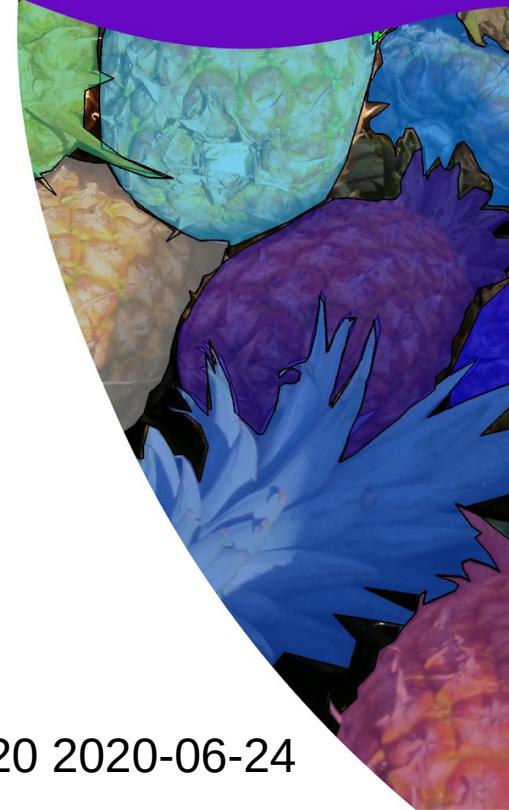




The Deep Learning - Applied Math Connection



Yann LeCun

NYU - Courant Institute & Center for Data Science

Facebook AI Research

<http://yann.lecun.com>

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



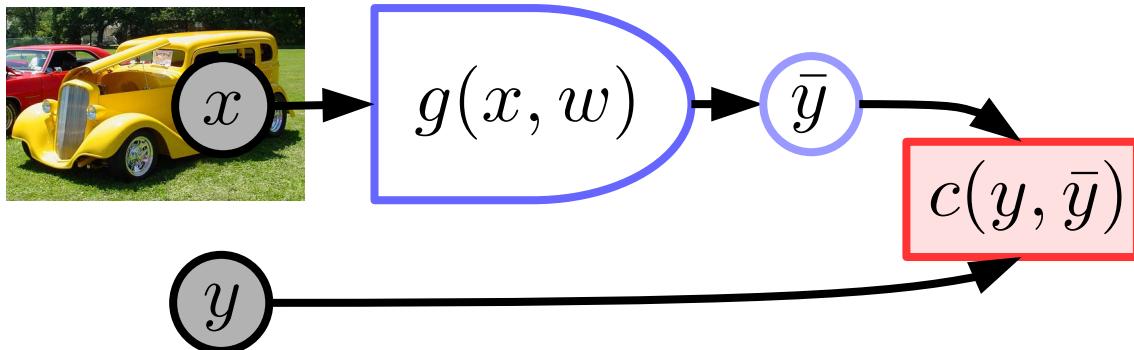
PLANE



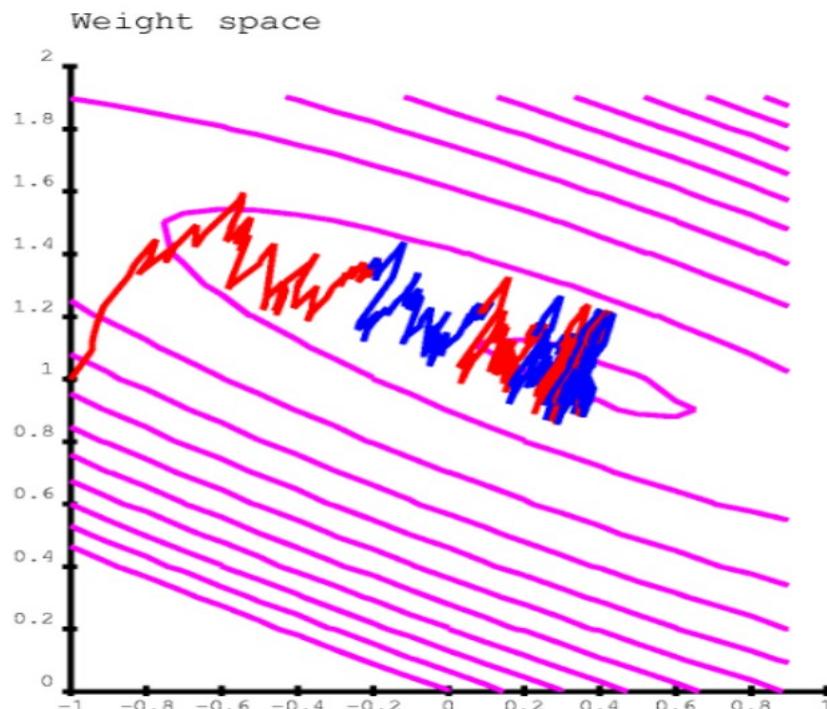
Supervised Machine Learning = Function Optimization

► Stochastic Gradient Descent (SGD):

- it's like walking down the mountain in a fog and following the direction of steepest descent to reach the valley
- But each sample gives us a noisy estimate of the direction. So our path is a bit random



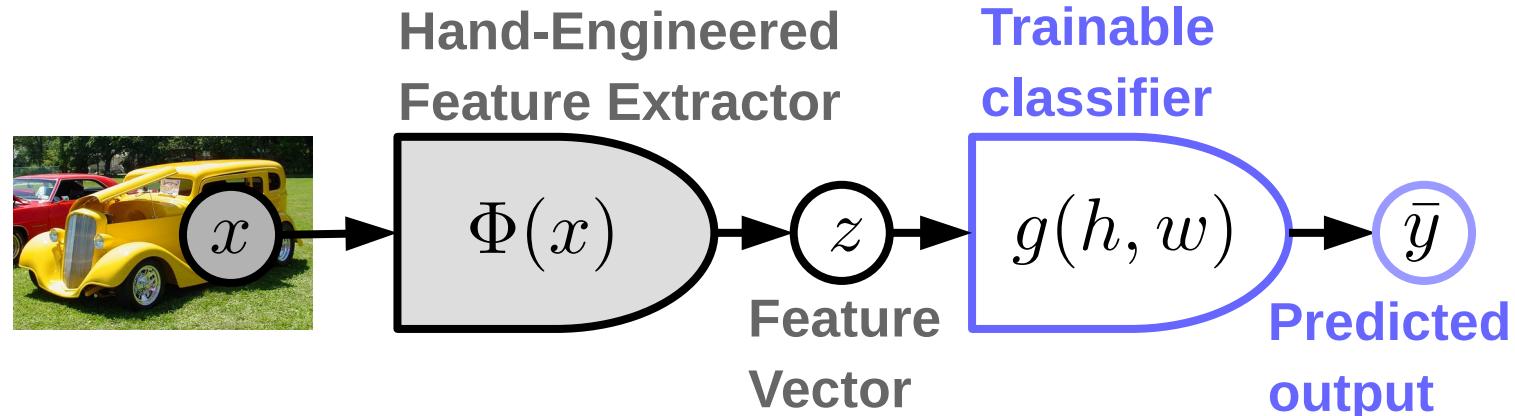
$$L(x, y, w) = c(y, g(x, w))$$



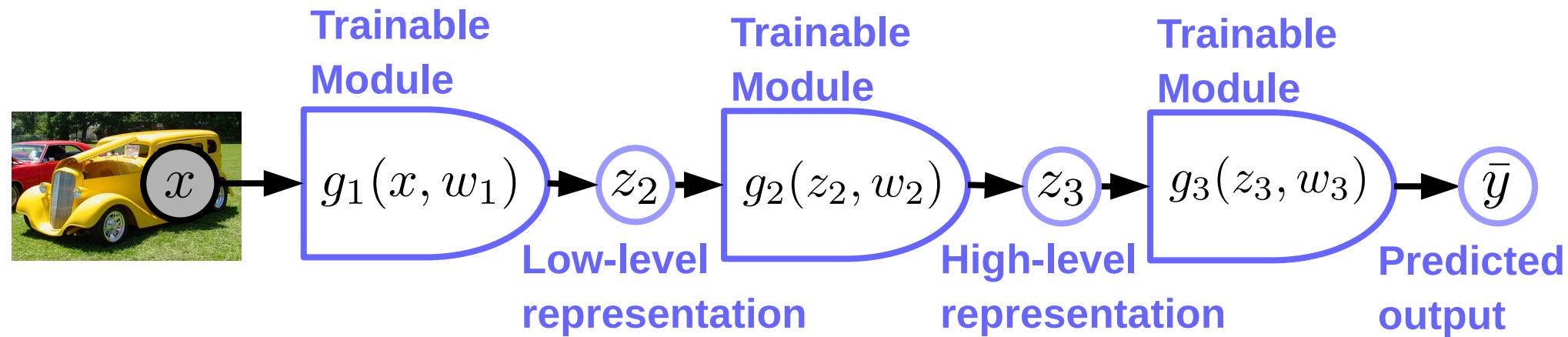
$$w \leftarrow w - \eta \frac{\partial L(x, y, w)}{w}$$

Traditional Machine Learning → Deep Learning

- ▶ Traditional Machine Learning & Pattern Recognition



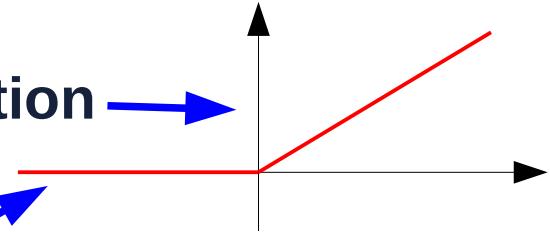
- ▶ Deep Learning: learning hierarchical representations



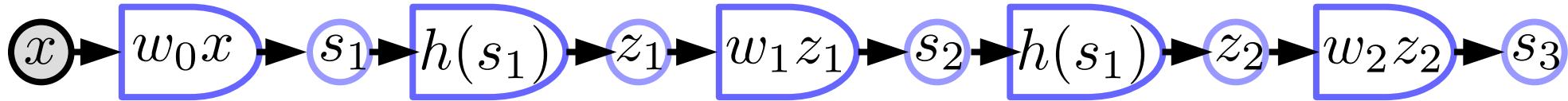
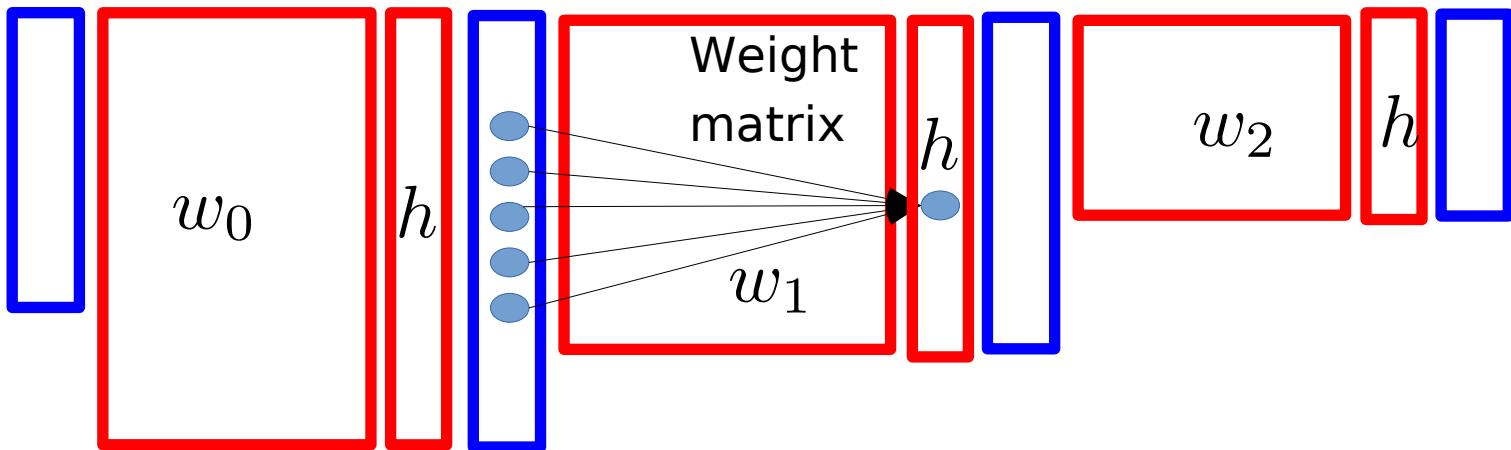
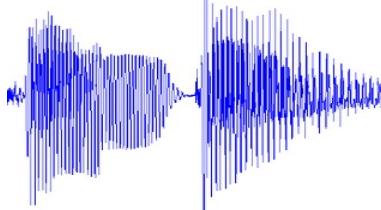
(Deep) Multi-Layer Neural Nets

- 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**

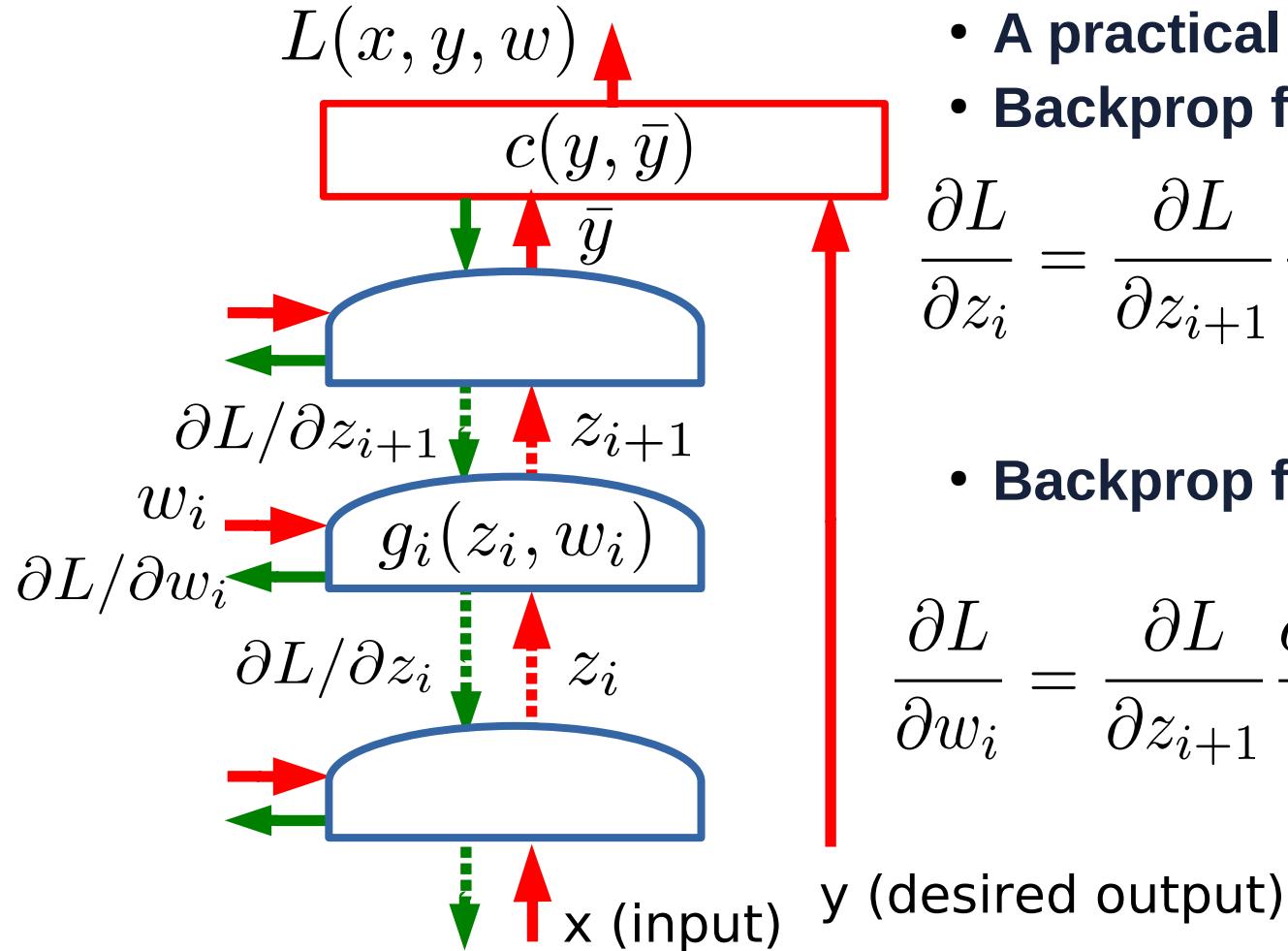
$$\text{ReLU}(x) = \max(x, 0)$$



Ceci est une voiture



Computing Gradients by Back-Propagation



- A practical Application of Chain Rule
- Backprop for the state gradients:

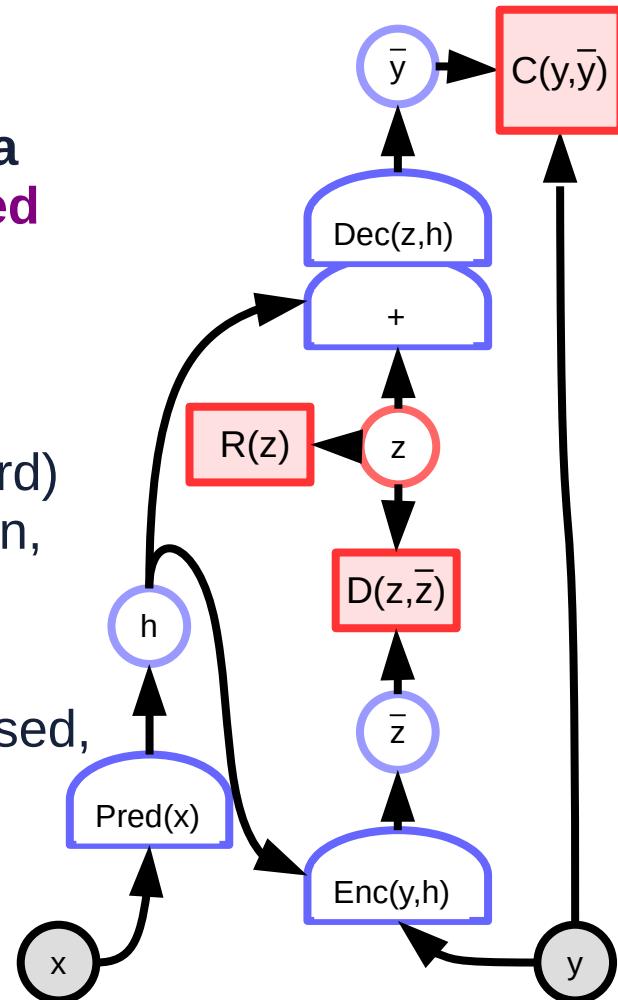
$$\frac{\partial L}{\partial z_i} = \frac{\partial L}{\partial z_{i+1}} \frac{\partial z_{i+1}}{\partial z_i} = \frac{\partial L}{\partial z_{i+1}} \frac{\partial g_i(z_i, w_i)}{\partial z_i}$$

- Backprop for the weight gradients:

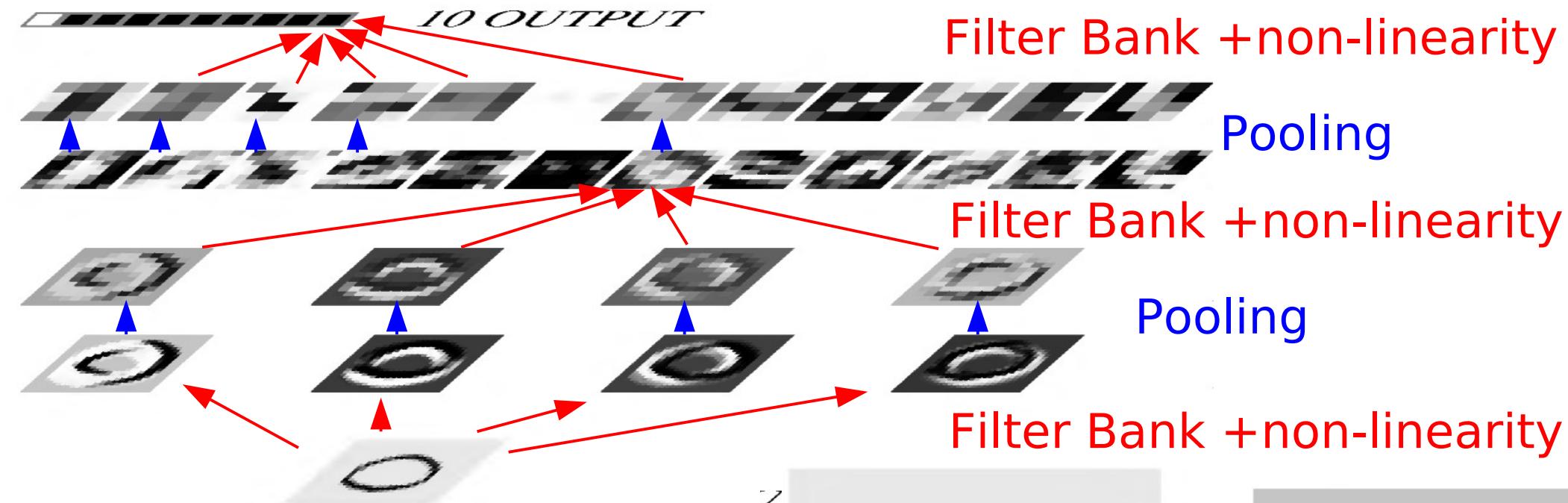
$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial z_{i+1}} \frac{\partial z_{i+1}}{\partial w_i} = \frac{\partial L}{\partial z_{i+1}} \frac{\partial g_i(z_i, w_i)}{\partial w_i}$$

What is Deep Learning?

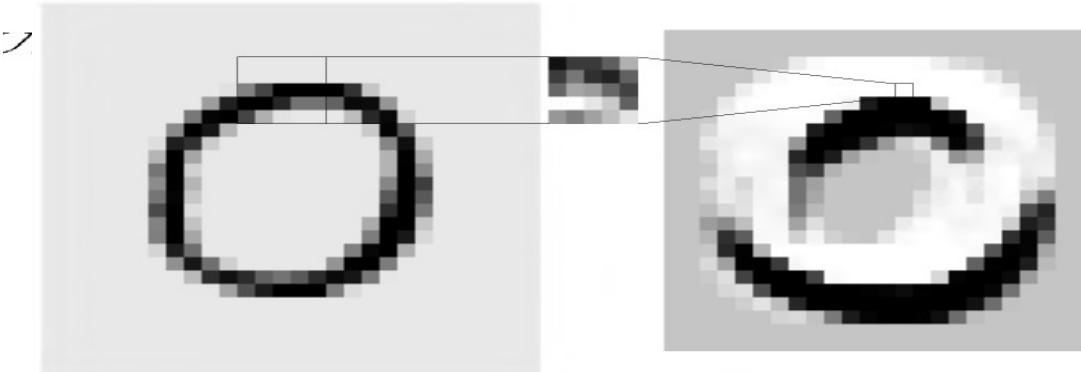
- ▶ **Definition:** Deep Learning is building a system by assembling parameterized **modules** into a (possibly dynamic) computation **graph**, and training it to perform a task by optimizing the parameters using a **gradient-based method**.
- ▶ Graph can be defined dynamically by input-dependent programs: **differentiable programming**
- ▶ Output may be computed through complex (non feed-forward) process, e.g. by **minimizing some energy function**: relaxation, constraint satisfaction, structured prediction,....
- ▶ Learning paradigms and objective functions are up to the designer: supervised, reinforced, self-supervised/unsupervised, classification, prediction, reconstruction,....
- ▶ **Note:** the limitations of Supervised Learning are sometimes mistakenly seen as intrinsic limitations of DL



Convolutional Network Architecture [LeCun et al. NIPS 1989]



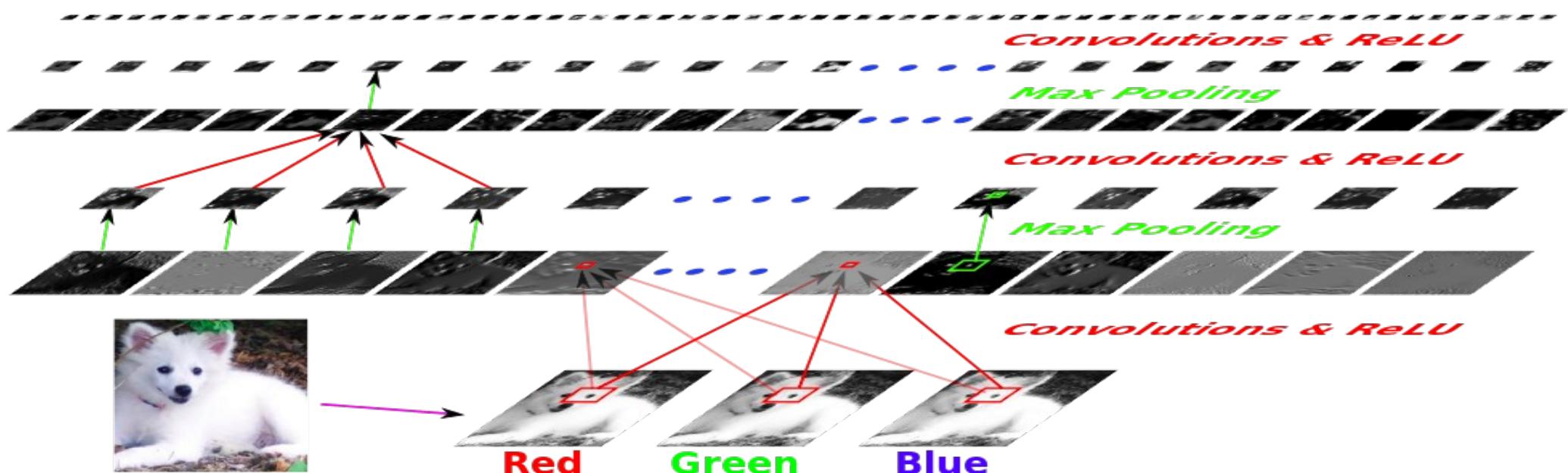
- Inspired by [Hubel & Wiesel 1962] & [Fukushima 1982] (Neocognitron):
 - ▶ **simple cells** detect local features
 - ▶ **complex cells** “pool” the outputs of simple cells within a retinotopic neighborhood.



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.

Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic Fox (1.0); Eskimo Dog (0.6); White Wolf (0.4); Siberian Husky (0.4)



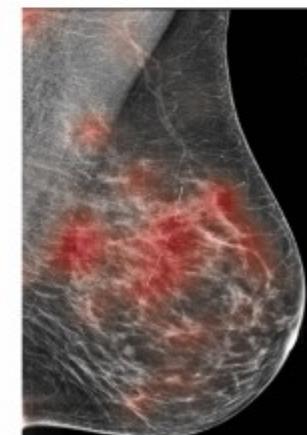
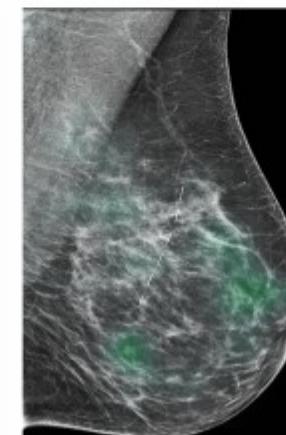
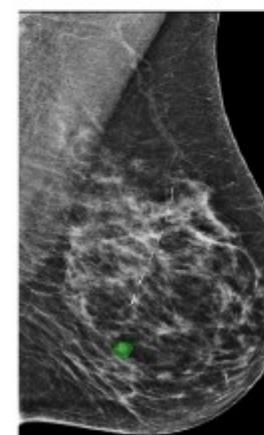
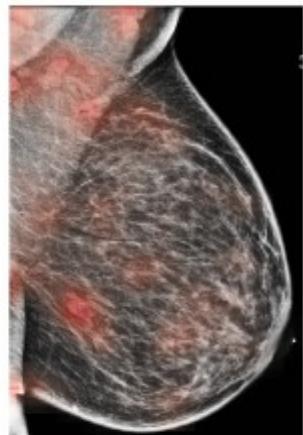
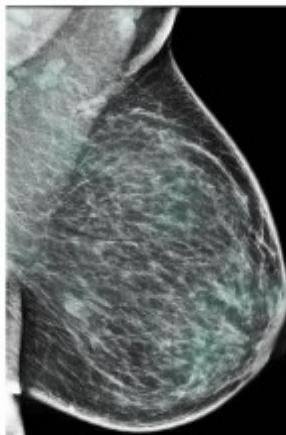
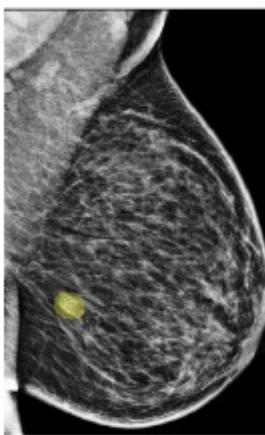
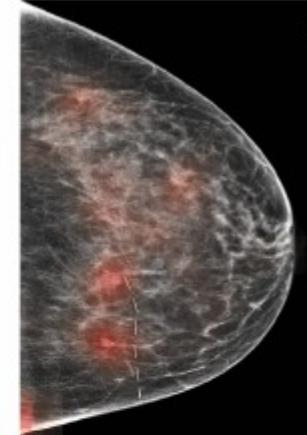
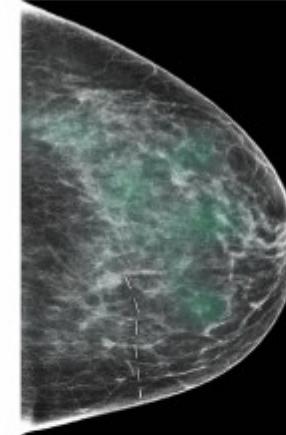
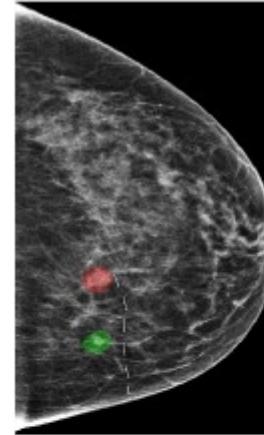
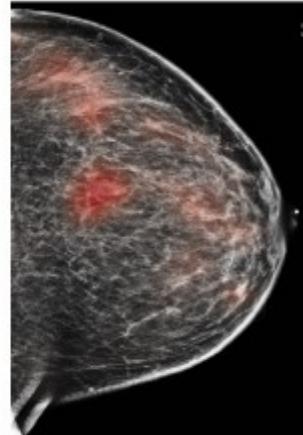
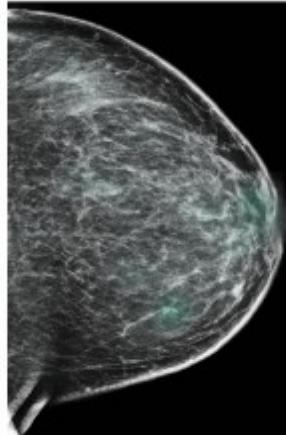
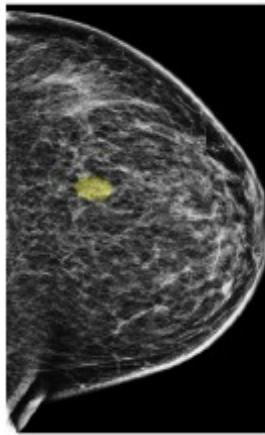
Detectron2 (FAIR) [Girshick 2019]

- ▶ Panoptic instance segmentation, (dense) body pose estimation
- ▶ Open source: <https://github.com/facebookresearch/detectron2>



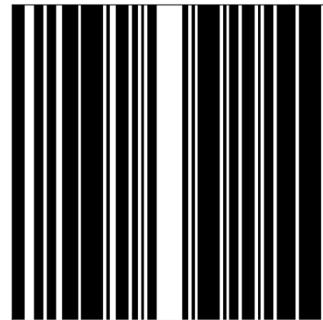
Breast Cancer Detection (NYU)

► [Wu et al. ArXiv:1903.08297] https://github.com/nyukat/breast_cancer_classifier

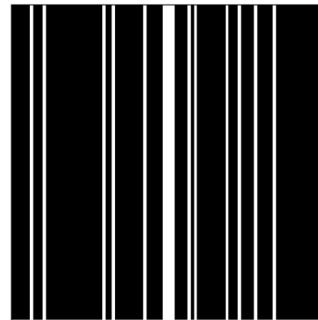


FastMRI (NYU+FAIR): 4x-8x speed up for MRI data acquisition

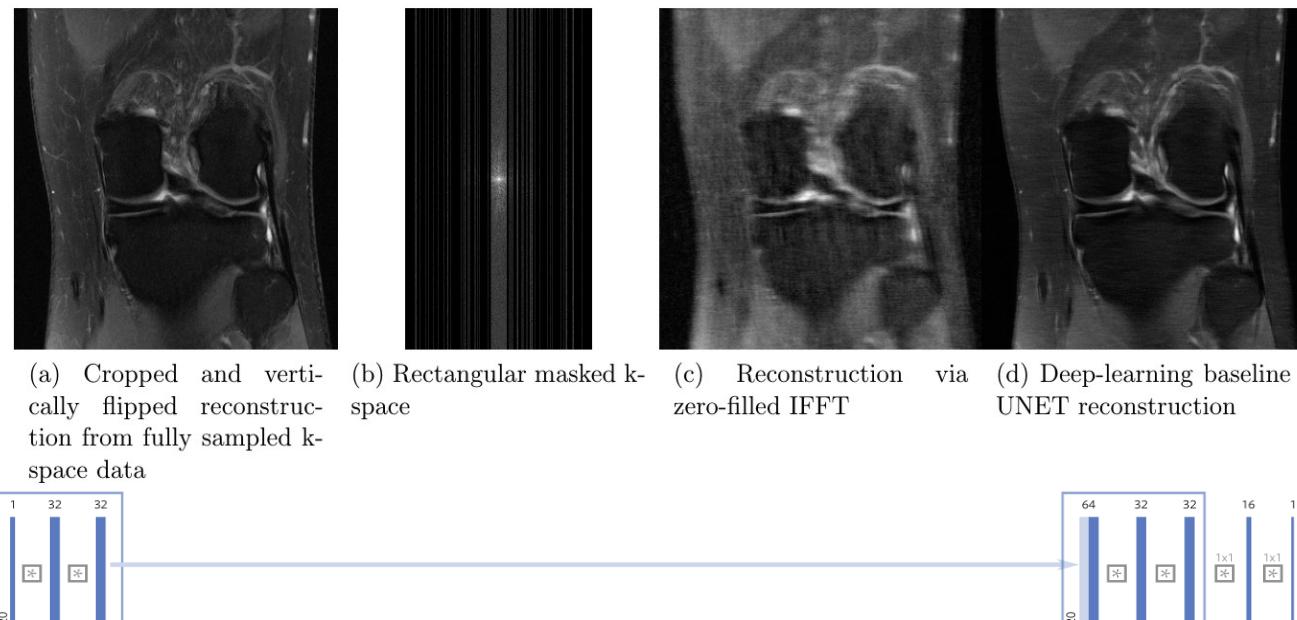
- ▶ MRI images subsampled (in k-space) by 4x and 8x
- ▶ [Zbontar et al. ArXiv:1811.08839]
- ▶ U-Net architecture
- ▶ 4-fold acceleration
- ▶ 8-fold acceleration
- ▶ K-space masks



(a) 4-fold acceleration



(b) 8-fold acceleration



(a) Cropped and vertically flipped reconstruction from fully sampled k-space data

(b) Rectangular masked k-space

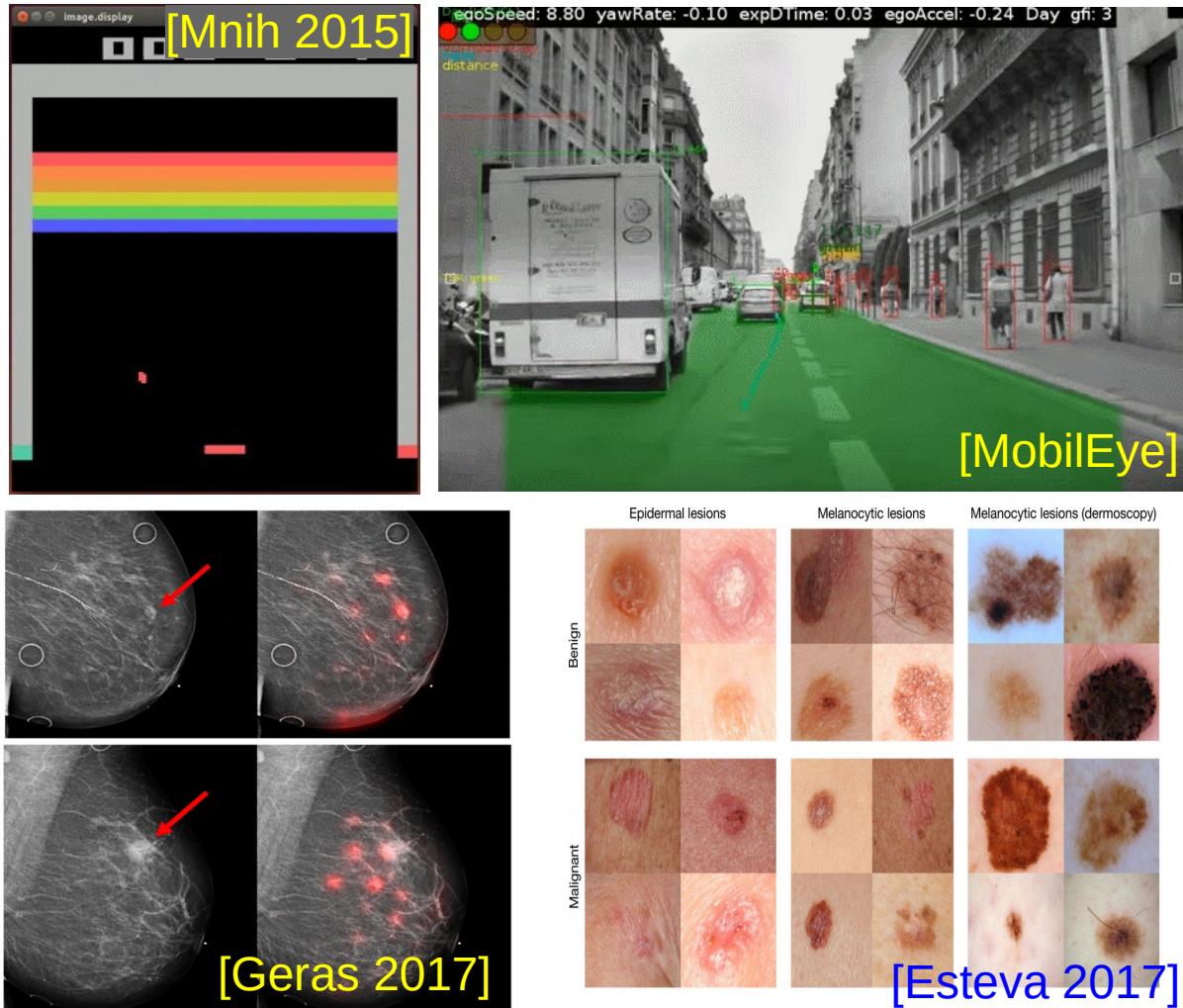
(c) Reconstruction via zero-filled IFFT

(d) Deep-learning baseline UNET reconstruction

$\downarrow \uparrow$	3x3 Convolution + ReLU + InstanceNorm
\square	2x2 Max pooling
$\square \uparrow$	2x2 Bilinear upsampling
$\downarrow \square$	1x1 Convolution

Applications of Deep Learning

- ▶ Medicine/medical image analysis
- ▶ Self-driving cars
- ▶ Accessibility
- ▶ Face recognition
- ▶ Language translation
- ▶ Virtual assistants*
- ▶ Content Understanding for:
 - ▶ Filtering
 - ▶ Selection/ranking
 - ▶ Search
 - ▶ Games
 - ▶ Security, anomaly detection
 - ▶ Diagnosis, prediction
 - ▶ Science: physics, biology.



Deep Learning Saves Lives

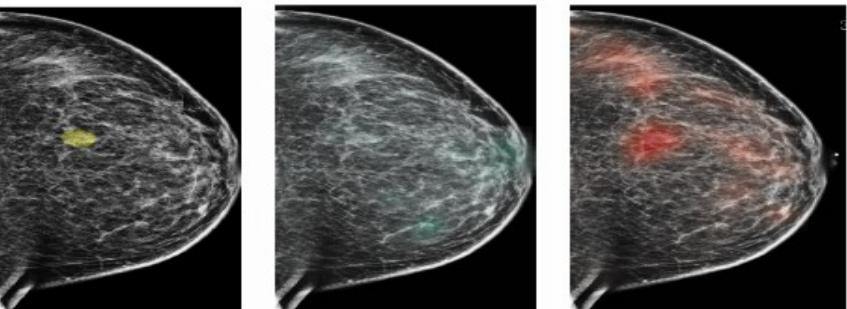
- ▶ **Automated Emergency Braking Systems**

- ▶ Reduce collisions by 40%
- ▶ Use Convolutional nets.



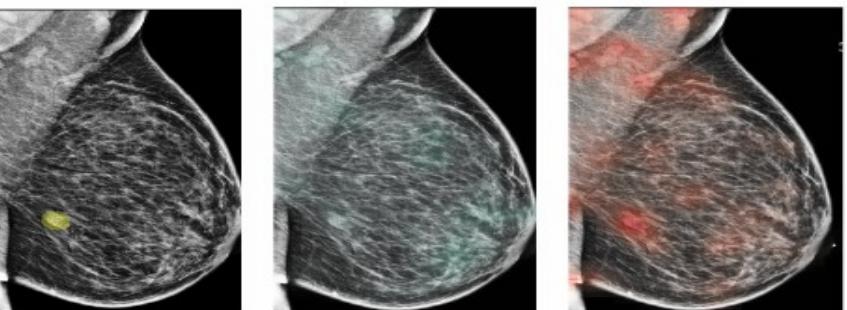
- ▶ **Tumor detection**

- ▶ [Wu et al. ArXiv:1903.08297]
https://github.com/nyukat/breast_cancer_classifier

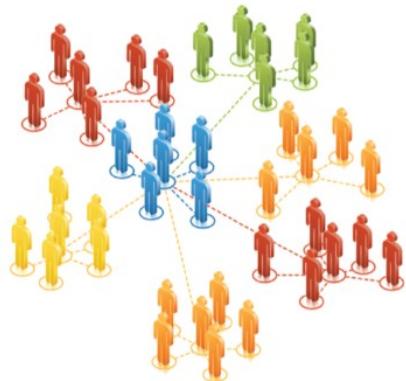


- ▶ **Content filtering.**

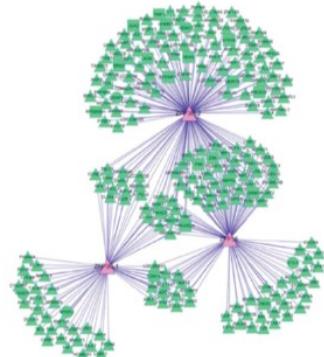
- ▶ Hate speech, calls to violence, weapon sales, terrorist propaganda....



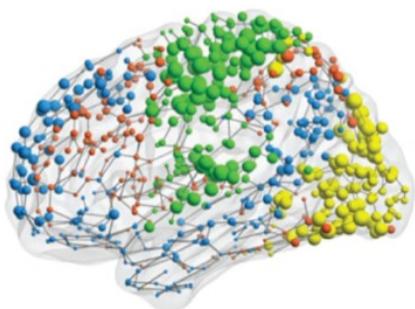
ConvNets on Graphs (fixed and data-dependent)



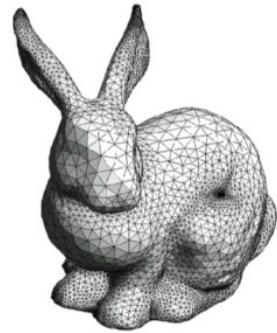
Social networks



Regulatory networks



Functional networks



3D shapes

- ▶ **Graphs can represent: Natural language, social networks, chemistry, physics, communication networks...**

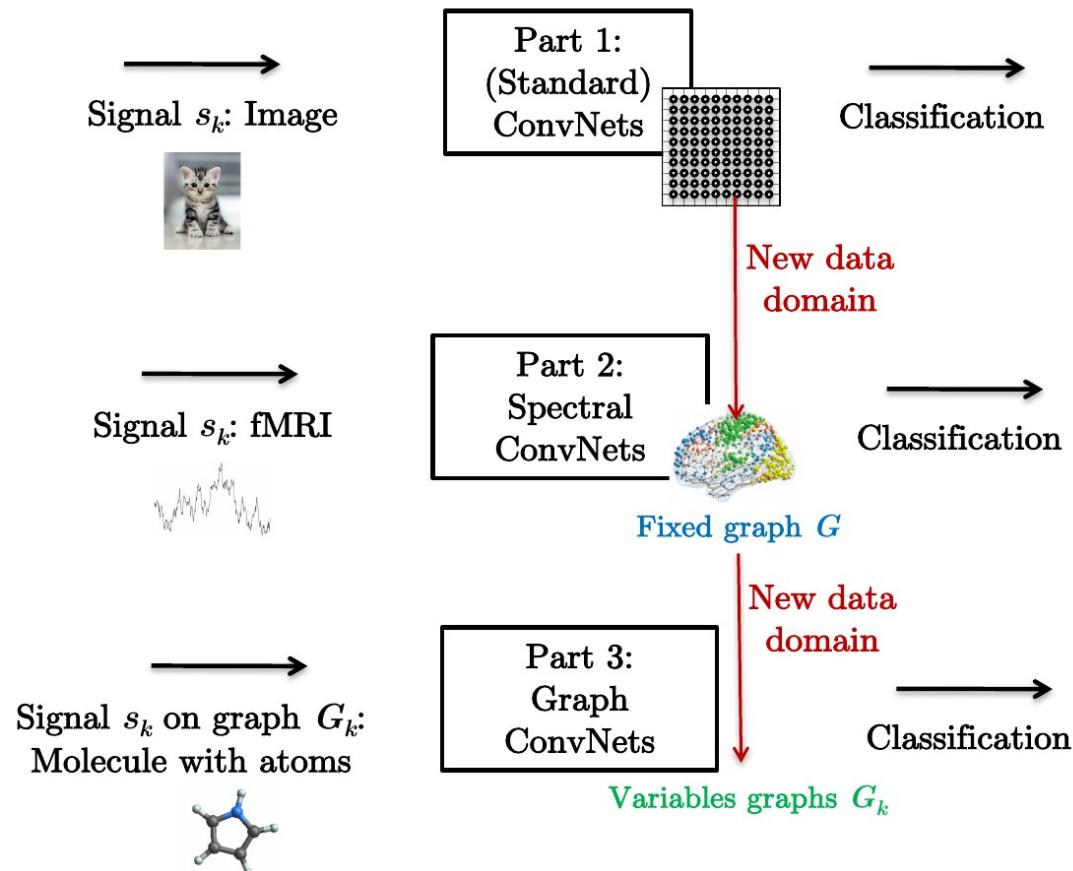
=

Graphs/
Networks

- ▶ **Review paper: “Geometric deep learning: going beyond euclidean data”, MM Bronstein, J Bruna, Y LeCun, A Szlam, P Vandergheynst, IEEE Signal Processing Magazine 34 (4), 18-42, 2017 [ArXiv:1611.08097]**

Spectral ConvNets / Graph ConvNets

- ▶ Regular grid graph
- ▶ Standard ConvNet
- ▶ Fixed irregular graph
- ▶ Spectral ConvNet
- ▶ Dynamic irregular graph
- ▶ Graph ConvNet



IPAM workshop:

<http://www.ipam.ucla.edu/programs/workshops/new-deep-learning-techniques/>

ConvNets (and Deep Learning) in Physics

► Approximate solutions of PDEs with a learned update

► Integration step of PDE solver: $Z(t+1) = Z(t) + dt*G(Z(t))$

where is $G()$ a translation-invariant local operator.

Example: $G(Z(t)) = V*f(W*Z(t))$ conv->transfer_func->conv

► High energy Physics

► Lattice QCD

► Fluid Dynamics

► Prediction of aero/hydro-dynamical properties of solids

► Shape refinement by gradient descent

► Cosmology / Astrophysics

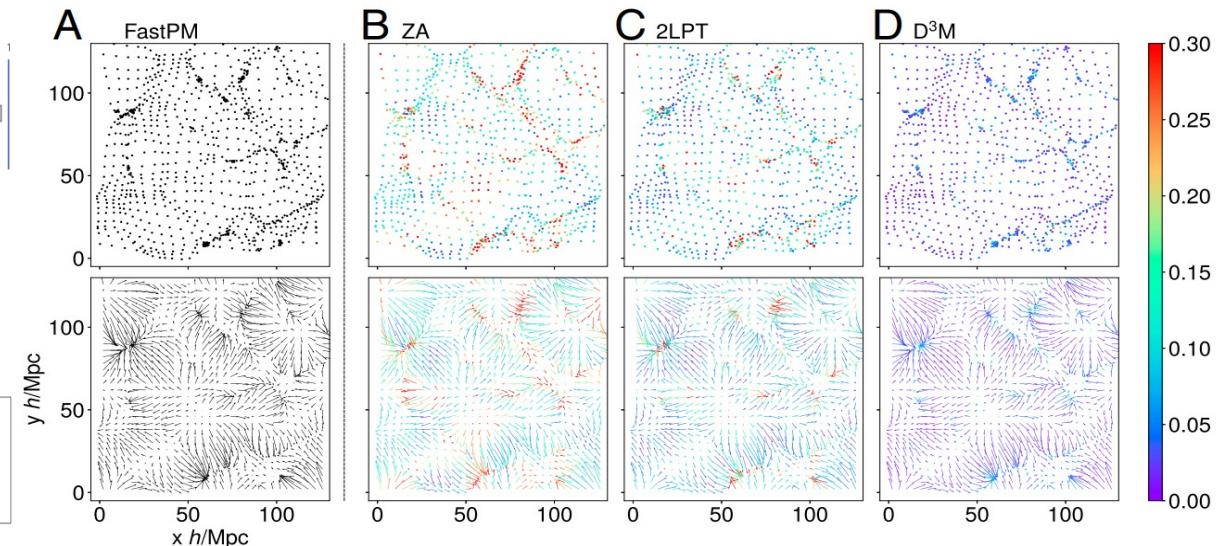
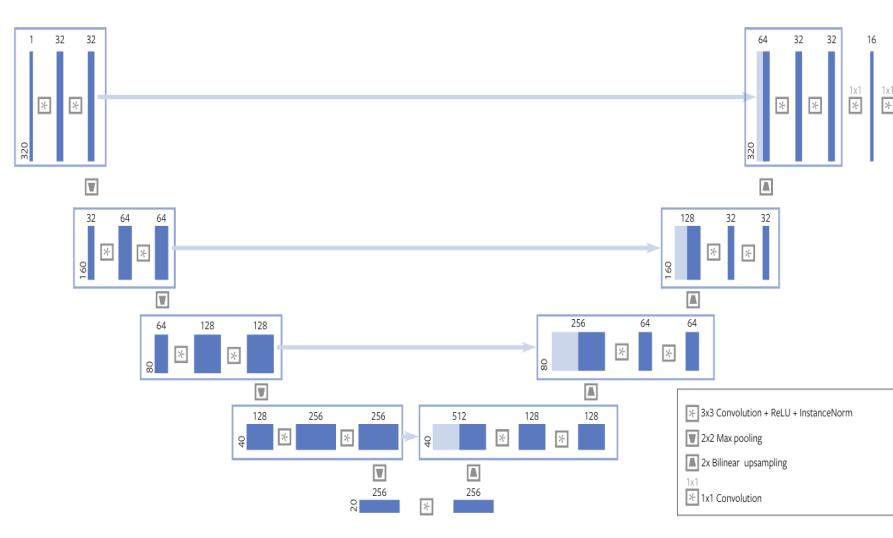
► Large-scale simulation of the early universe

ConvNets in Astrophysics [He et al. PNAS 07/2019]

Learning to predict the cosmological structure formation

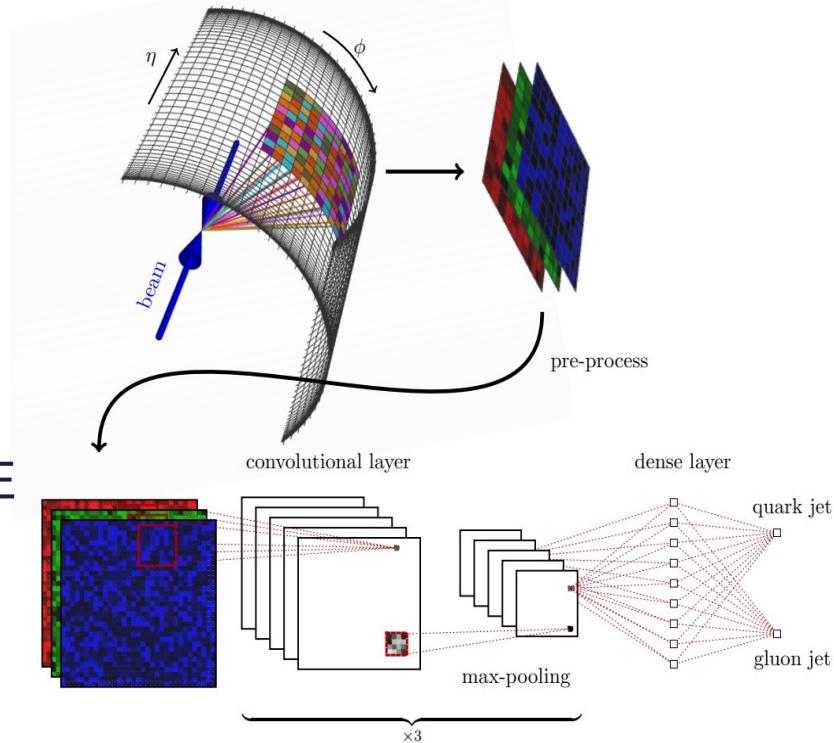
Siyu He^{a,b,c,1}, Yin Li^{d,e,f}, Yu Feng^{d,e}, Shirley Ho^{a,b,c,d,e,1}, Siamak Ravanbakhsh^g, Wei Chen^c, and Barnabás Póczos^h

- ▶ 1. Train a coarse-grained 3D U-Net to approximate a fine-grained simulation on a small volume
- ▶ 2. Use it for a simulation on a large volume (the early universe)



ConvNets (and Deep Learning) in Physics

- ▶ **Material Science / Molecular dynamics**
 - ▶ Protein structure/function prediction
 - ▶ Prediction of material properties
- ▶ **High energy Physics**
 - ▶ Jet filtering / analysis
 - ▶ “Deep learning in color: towards automated quark/gluon jet discrimination”, P Komiske, E Metodiev, M Schwartz, arXiv:1612.01551
- ▶ **Cosmology / Astrophysics**
 - ▶ Inferring constants from observations
 - ▶ Statistical studies of galaxies,
 - ▶ Dark matter through gravitational lensing



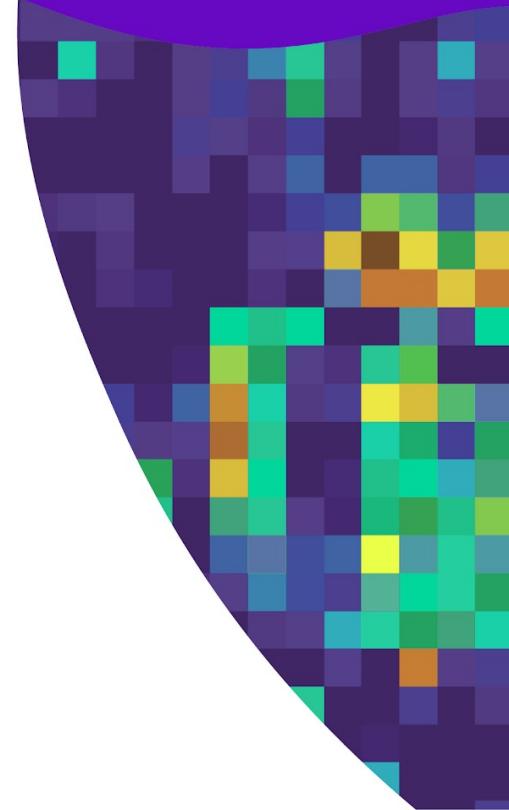
Applications of ConvNets

- ▶ **Self-driving cars, visual perception**
- ▶ **Medical signal and image analysis**
 - ▶ Radiology, dermatology, EEG/seizure prediction....
- ▶ **Bioinformatics/genomics**
- ▶ **Speech recognition**
- ▶ **Language translation**
- ▶ **Image restoration/manipulation/style transfer**
- ▶ **Robotics, manipulation**
- ▶ **Physics**
 - ▶ High-energy physics, astrophysics
- ▶ **New applications appear every day**
 - ▶ E.g. environmental protection,....

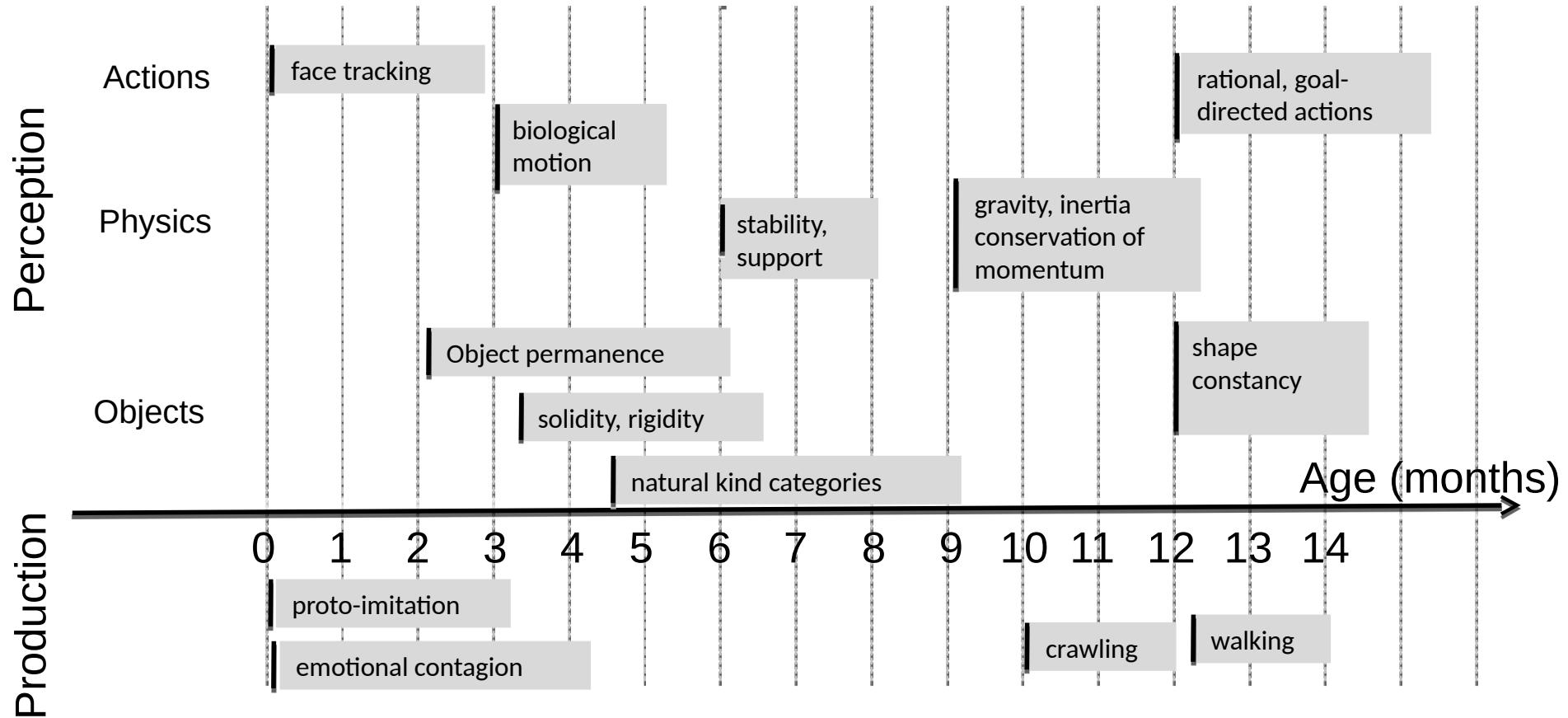


How do humans and animals learn so quickly?

Barely supervised.
Rarely Reinforced.



Early Concept Acquisition in Infants [from Emmanuel Dupoux]



How do Human and Animal Babies Learn?

- ▶ How do they learn how the world works?
- ▶ Largely by **observation**, with remarkably little interaction (initially).
- ▶ They accumulate enormous amounts of **background knowledge**
 - ▶ About the structure of the world, like intuitive physics.
- ▶ Perhaps **common sense** emerges from this knowledge?



Photos courtesy of
Emmanuel Dupoux

The Future of Representation Learning, ML, CV, NLP, AI



With thanks to
Alyosha Efros,
Gil Scott Heron, &
Eugène Delacroix



Get the T-shirt!

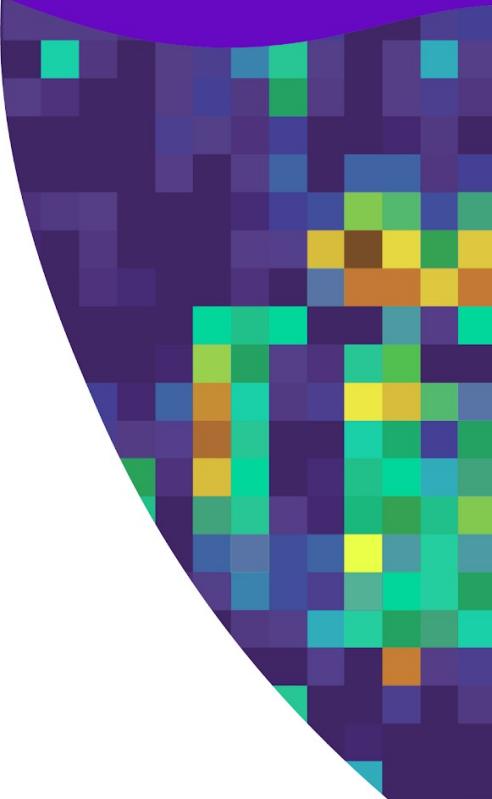
Jitendra Malik: “Labels are the opium of the machine learning researcher”

Three challenges for Deep Learning

- ▶ **Deep Supervised Learning works well for perception**
 - ▶ When labeled data is abundant. But there is rarely enough data.
- ▶ **Deep Reinforcement Learning works well for action generation**
 - ▶ When trials are cheap, e.g. in simulation. Too slow for the real world.
- ▶ **Three problems the community must solve:**
 - ▶ **1. Learning with fewer labeled samples and/or fewer trials**
 - ▶ Self-supervised learning / learning dependencies / to fill in the blanks
 - ▶ learning to represent the world in a non task-specific way
 - ▶ **2. Learning to reason**, beyond “system 1” feed-forward computation.
 - ▶ Making reasoning compatible with gradient-based learning.
 - ▶ **3. Learning to plan complex action sequences**
 - ▶ Learning hierarchical representations of action plans

Self-Supervised Learning

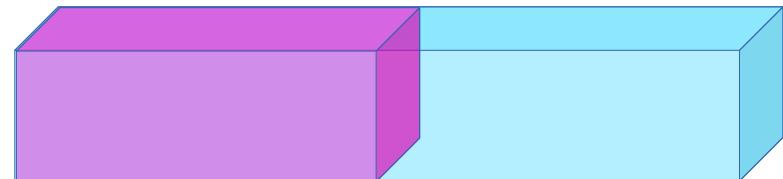
Capture dependencies.
Predict everything from everything else.



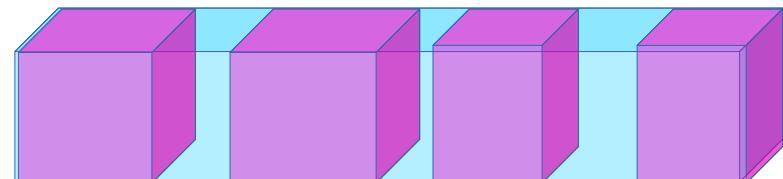
Self-Supervised Learning = Filling in the Blanks

- ▶ Predict any part of the input from any other part.

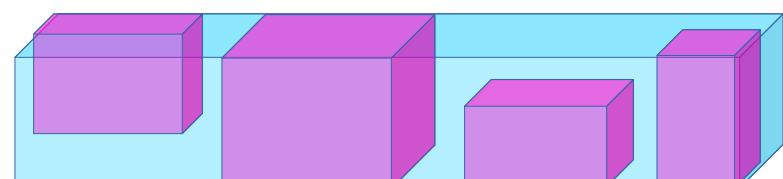
time or space →



- ▶ Predict the **future** from the **past**.



- ▶ Predict the **invisible** from the **visible**.



- ▶ Predict any **occluded, masked, or corrupted part from all available parts**.

- ▶ Pretend there is a part of the input you don't know and predict that.
- ▶ Reconstruction = SSL when any part could be known or unknown

Two Uses for Self-Supervised Learning

- ▶ **1. Learning hierarchical representations of the world**
 - ▶ SSL pre-training precedes a supervised or RL phase
- ▶ **2. Learning predictive (forward) models of the world**
 - ▶ Learning models for Model-Predictive Control, policy learning for control, or model-based RL.
- ▶ **Question:** how to represent uncertainty/multi-modality in the prediction?

Inference & Multimodal Predictions through Constraint Relaxation

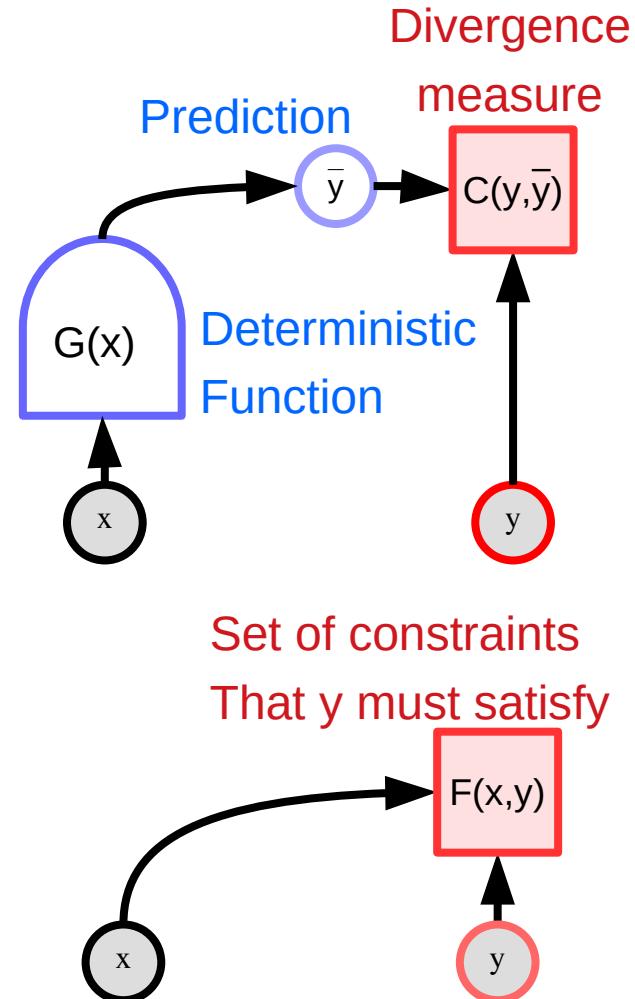
- ▶ Feed-forward nets use a fixed number of steps to produce a single output.

- ▶ What if...

- ▶ The problem requires a complex computation to produce its output? (complex inference)
- ▶ There are multiple possible outputs for a single input? (e.g. predicting future video frames)

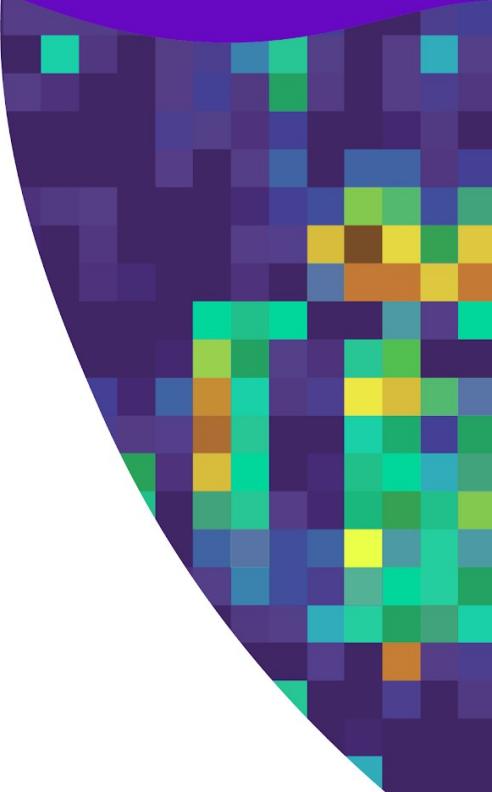
- ▶ Inference through constraint relaxation

- ▶ Finding an output that satisfies constraints: e.g. a linguistically correct translation or speech transcription.
- ▶ Maximum likelihood inference in graphical models



Energy-Based Models

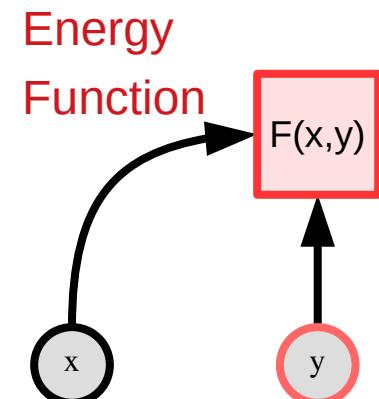
Capture dependencies through
an energy function.



Energy-Based Models (EBM)

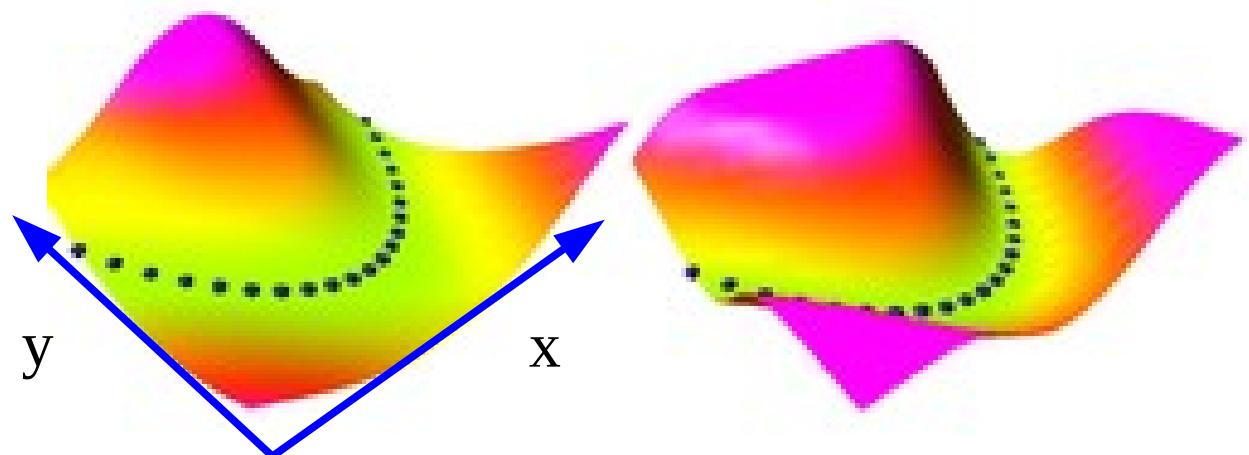
- ▶ **Energy function $F(x,y)$ scalar-valued.**
- ▶ Takes low values when y is compatible with x and higher values when y is less compatible with x
- ▶ **Inference:** find values of y that make $F(x,y)$ small.
- ▶ There may be multiple solutions

$$\check{y} = \operatorname{argmin}_y F(x, y)$$



- ▶ Note: the energy is used for inference, not for learning

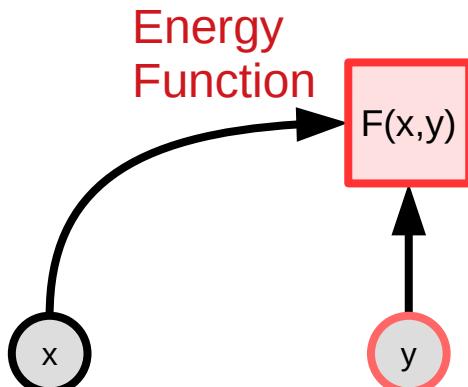
- ▶ Example
- ▶ Blue dots are data points



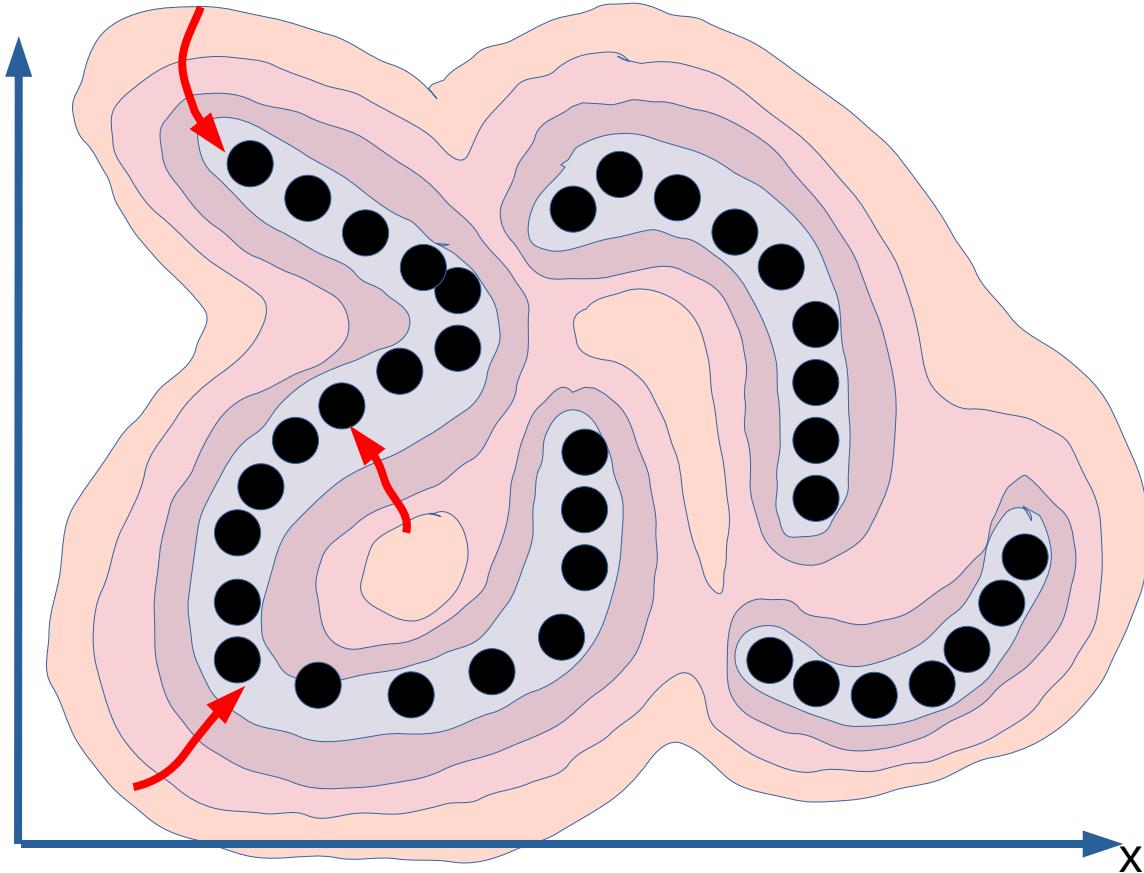
Energy-Based Model: gradient-based inference

- ▶ If y is continuous
 - ▶ We can use a gradient-based method for inference.

$$\check{y} = \operatorname{argmin}_y F(x, y)$$

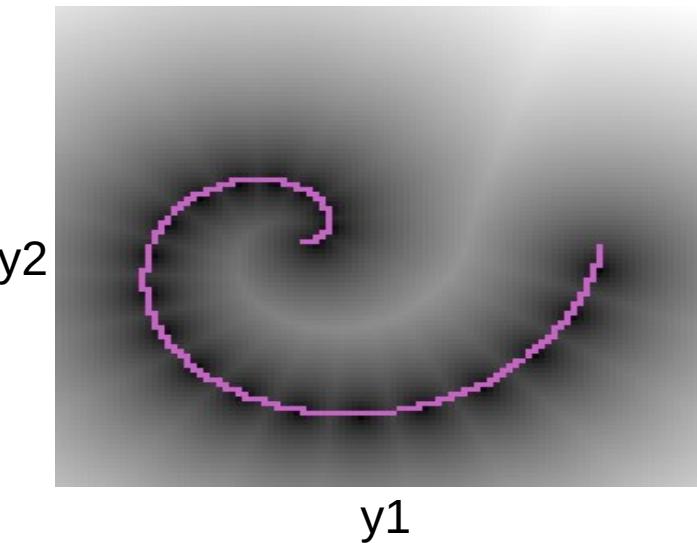


y

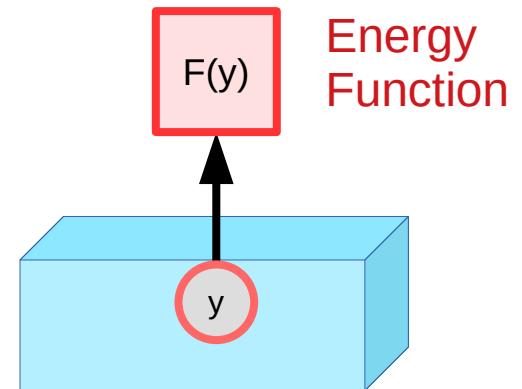
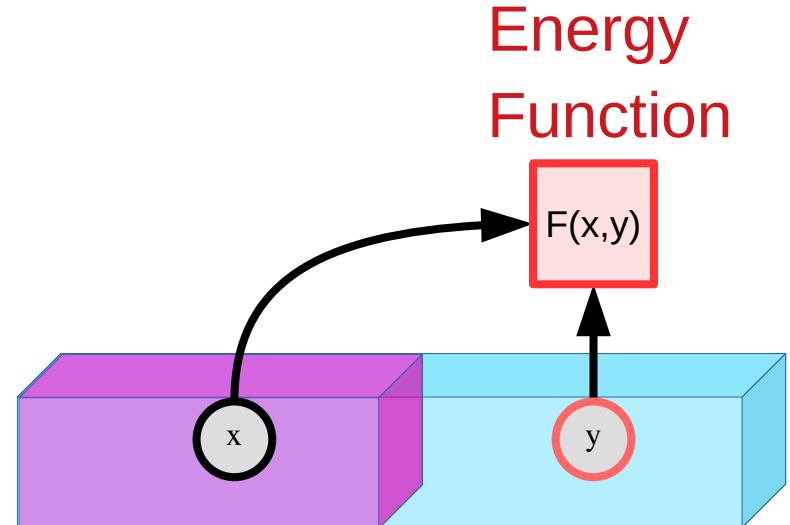


Energy-Based Model: unconditional version

- ▶ Conditional EBM: $F(x,y)$
- ▶ Unconditional EBM: $F(y)$
- ▶ measures the compatibility between the components of y
- ▶ If we don't know in advance which part of y is known and which part is unknown

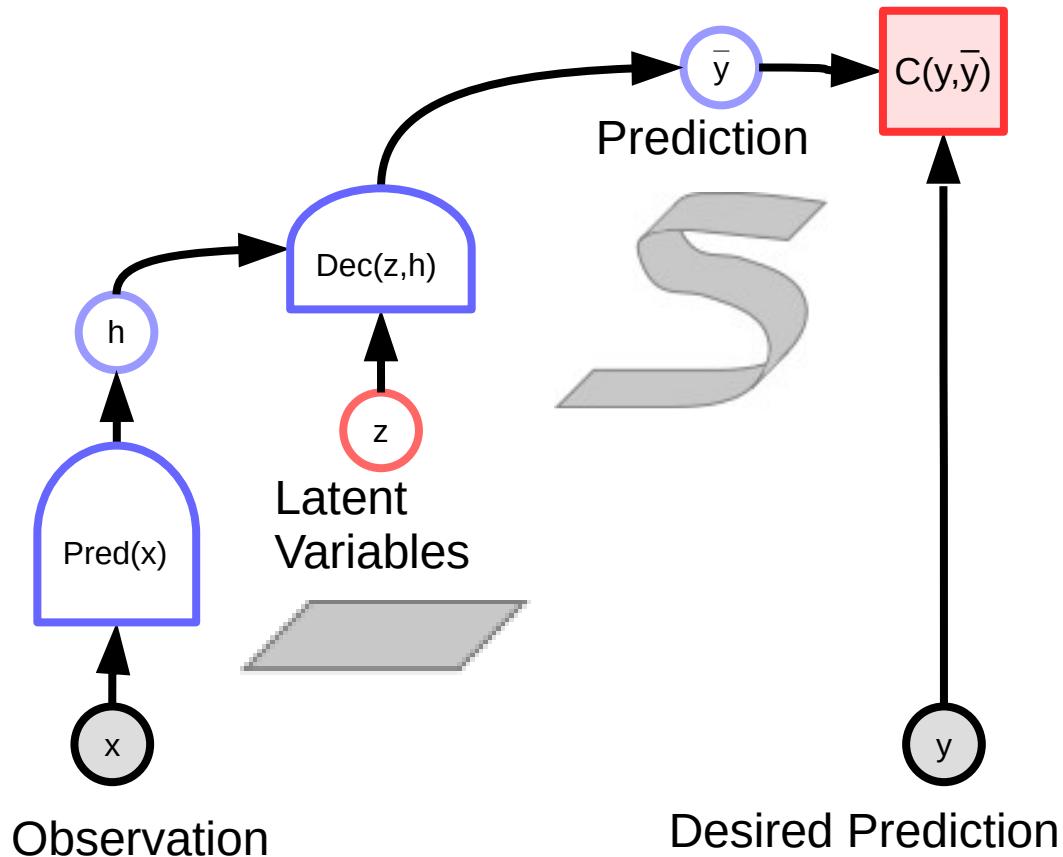


Dark = low energy (good)
 Bright = high energy (bad)
Purple = data manifold



Architecture for Multimodal Output: latent variable EBM

- ▶ **Latent variables:**
 - ▶ parameterize the set of predictions
- ▶ **Ideally, the latent variable represents independent explanatory factors of variation of the prediction.**
- ▶ **The information capacity of the latent variable must be minimized.**
 - ▶ Otherwise all the information for the prediction will go into it.



Latent-Variable EBM: inference

- ▶ Simultaneous minimization with respect to y and z

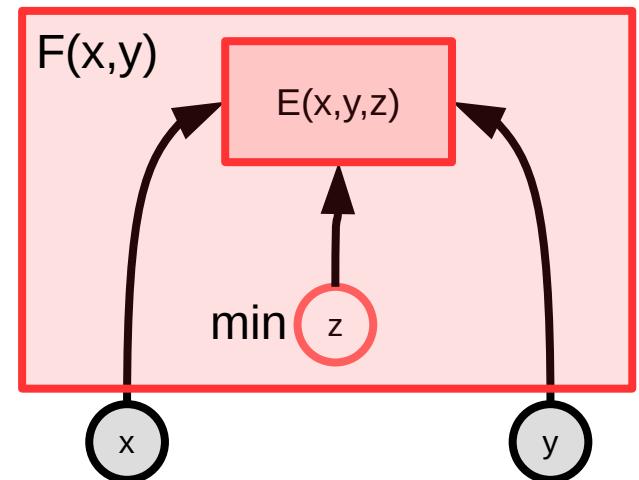
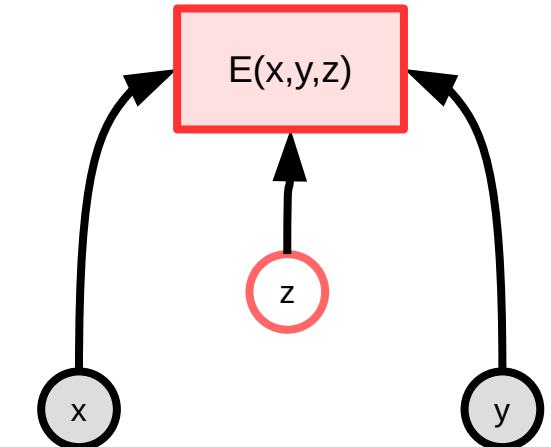
$$\check{y}, \check{z} = \operatorname{argmin}_{y,z} E(x, y, z)$$

- ▶ Redefinition of $F(x,y)$

$$F_\infty(x, y) = \operatorname{argmin}_z E(x, y, z)$$

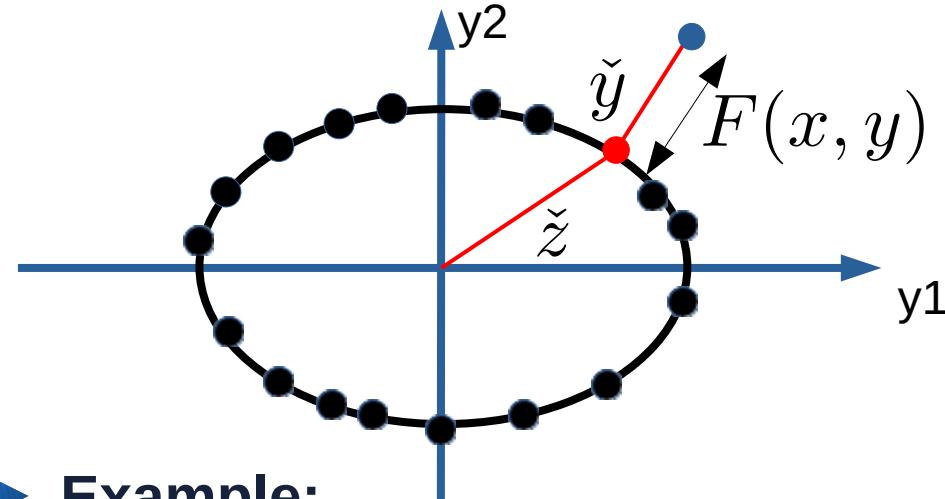
$$F_\beta(x, y) = -\frac{1}{\beta} \log \int_z e^{-\beta E(x, y, z)}$$

$$\check{y} = \operatorname{argmin}_y F(x, y)$$



Inference with Latent Variable EBMs

- ▶ The latent variable **parameterizes** the data manifold(s).
- ▶ The energy computes a **distance** to the data manifold(s).
- ▶ The gradient of the energy points to the closest point on the data manifold(s).

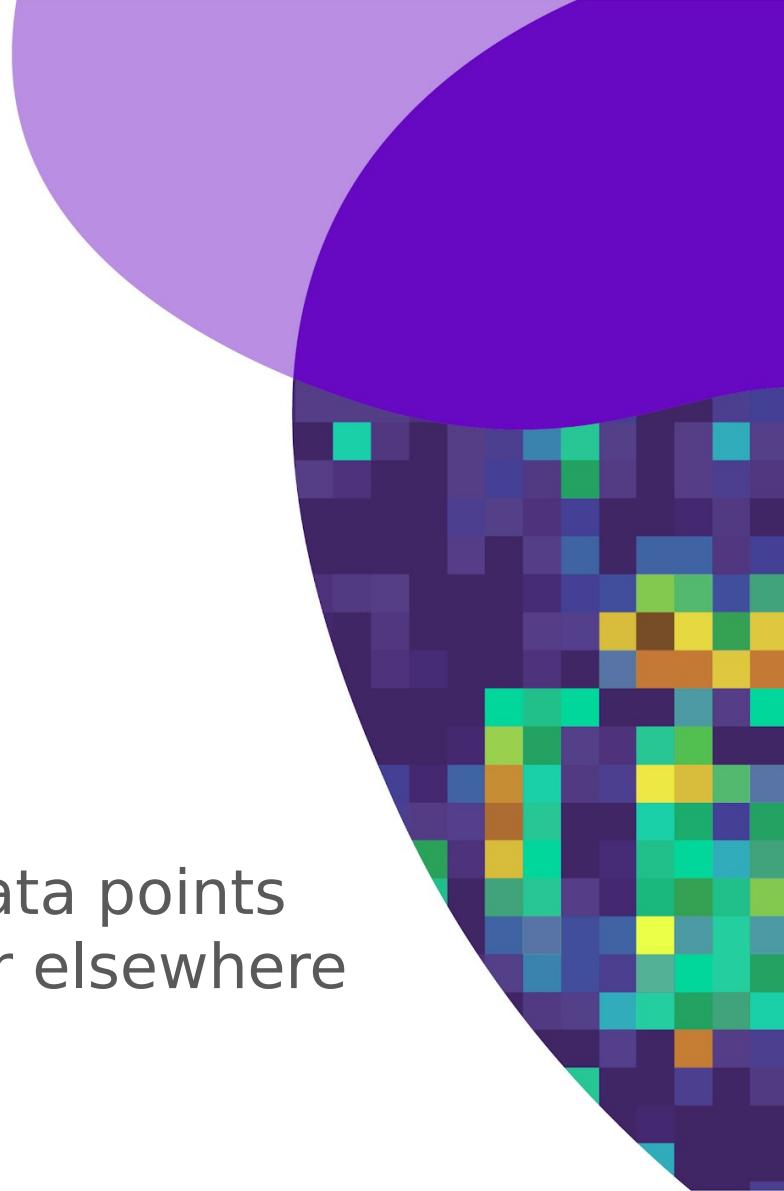


- ▶ Example:
 - ▶ Data manifold = ellipse
 - ▶ Latent variable = angle
 - ▶ Energy = distance to ellipse

$$E(y, z) = (y_1 - r_1 \sin(z))^2 + (y_2 - r_2 \cos(z))^2$$

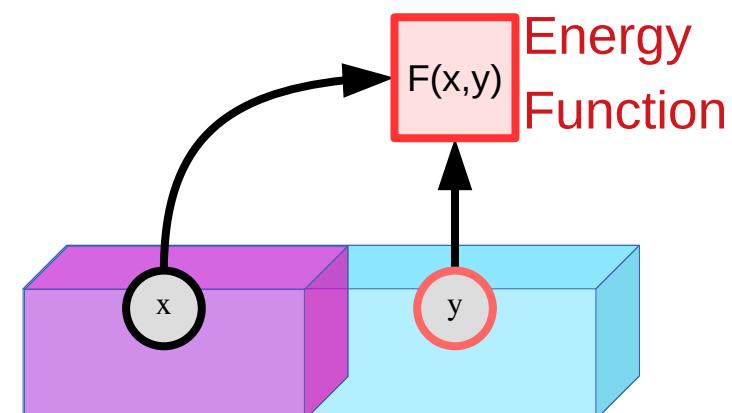
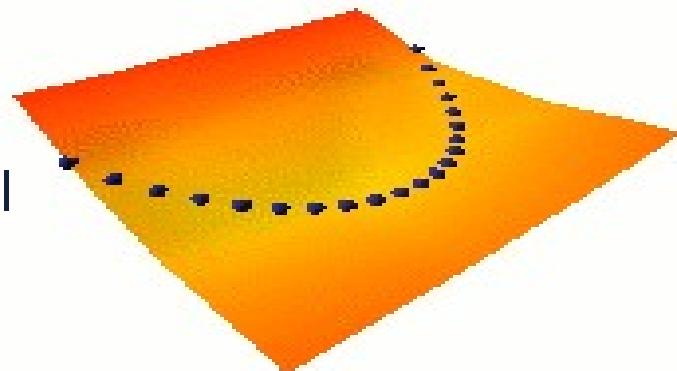
Training EBMs

Push down on the energy of data points
Make sure the energy is higher elsewhere



Training an Energy-Based Model

- ▶ Parameterize $F(x,y)$
- ▶ Training samples: $x[i], y[i]$
- ▶ Shape $F(x,y)$ so that:
 - ▶ $F(x[i], y[i])$ is strictly smaller than $F(x[i], y)$ for all y different from $y[i]$
 - ▶ Keep F smooth
 - ▶ Max-likelihood probabilistic methods break that!
- ▶ Two classes of learning methods:
 - ▶ 1. **Contrastive methods**: push down on $F(x[i], y[i])$, push up on other points $F(x[i], y')$
 - ▶ 2. **Regularized/Architectural Methods**: build $F(x,y)$ so that the volume of low energy regions is limited or minimized through regularization



Contrastive Methods vs Regularized/Architectural Methods

- ▶ **Contrastive:** [they all are different ways to pick which points to push up]
 - ▶ C1: push down of the energy of data points, push up everywhere else: Max likelihood (needs tractable partition function or variational approximation)
 - ▶ C2: push down of the energy of data points, push up on chosen locations: max likelihood with MC/MMC/HMC, Contrastive divergence, Metric learning/Siamese nets, Ratio Matching, Noise Contrastive Estimation, Min Probability Flow, adversarial generator/GANs
 - ▶ C3: train a function that maps points off the data manifold to points on the data manifold: denoising auto-encoder, masked auto-encoder (e.g. BERT)
- ▶ **Regularized/Architectural:** [Different ways to limit the information capacity of the latent representation]
 - ▶ A1: build the machine so that the volume of low energy space is bounded: PCA, K-means, Gaussian Mixture Model, Square ICA, normalizing flows...
 - ▶ A2: use a regularization term that measures the volume of space that has low energy: Sparse coding, sparse auto-encoder, LISTA, Variational Auto-Encoders, discretization/VQ/VQVAE.
 - ▶ A3: $F(x,y) = C(y, G(x,y))$, make $G(x,y)$ as "constant" as possible with respect to y : Contracting auto-encoder, saturating auto-encoder
 - ▶ A4: minimize the gradient and maximize the curvature around data points: score matching

Contrastive Methods: Max likelihood / Probabilistic Methods

- ▶ **Push down on data points, push up of other points**

- ▶ well chosen contrastive points

- ▶ **Max likelihood / probabilistic models**

$$P_w(y|x) = \frac{e^{-\beta F_w(x,y')}}{\int_{y'} e^{-\beta F_w(x,y')}}$$

- ▶ Loss: $\mathcal{L}(x, y, w) = F_w(x, y) + \frac{1}{\beta} \log \int_{y'} e^{-\beta F_w(x, y')}$

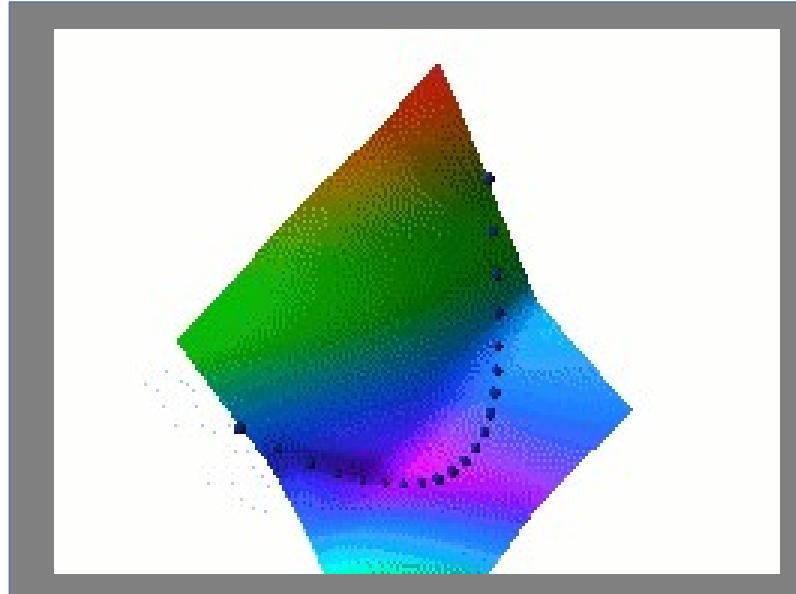
- ▶ Gradient: $\frac{\partial \mathcal{L}(x, y, w)}{\partial w} = \frac{\partial F_w(x, y)}{\partial w} - \int_{y'} P_w(y'|x) \frac{\partial F_w(x, y')}{\partial w}$

- ▶ MC/MCMC/HMC/CD: \hat{y} sampled from $P_w(y|x)$

$$\frac{\partial \mathcal{L}(x, y, w)}{\partial w} = \frac{\partial F_w(x, y)}{\partial w} - \frac{\partial F_w(x, \hat{y})}{\partial w}$$

Problem with Max Likelihood / Probabilistic Methods

- ▶ It wants to make the difference between the energy on the data manifold and the energy just outside of it infinitely large!
- ▶ It wants to make the data manifold an infinitely deep and infinitely narrow canyon.
- ▶ The loss must be regularized to keep the energy smooth
 - ▶ e.g. à la Wasserstein GAN.
 - ▶ So that gradient-based inference works
 - ▶ Equivalent to a prior
 - ▶ But then, why use a probabilistic model?



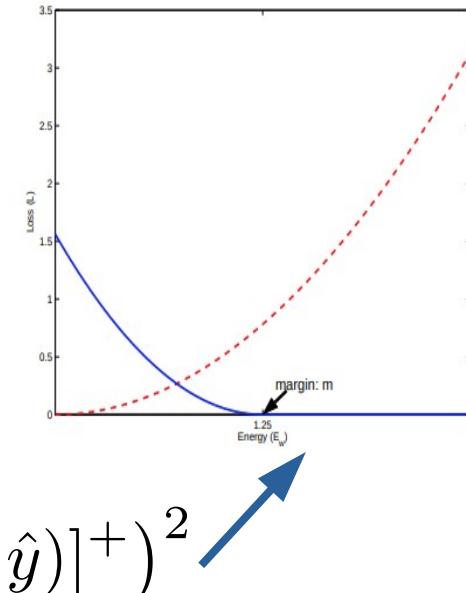
Contrastive Methods: other losses

- ▶ Push down on data points, push up of other points
- ▶ well chosen contrastive points
- ▶ General margin loss: $\mathcal{L}(x, y, \hat{y}, w) = H(F_w(x, y), F_w(x, \hat{y}), m(y, \hat{y}))$
 - ▶ Where $H(F^+, F^-, m)$ is a strictly increasing function of F^+ and a strictly decreasing function of F^- , at least whenever $F^- - F^+ < m$.
- ▶ Examples:
 - ▶ Simple [Bromley 1993]:

$$\mathcal{L}(x, y, \hat{y}, w) = [F_w(x, y)]^+ + [m(y, \hat{y}) - F_w(x, \hat{y})]^+$$
 - ▶ Hinge pair loss [Altun 2003], Ranking loss [Weston 2010]:

$$\mathcal{L}(x, y, \hat{y}, w) = [F_w(x, y) - F_w(x, \hat{y}) + m(y, \hat{y})]^+$$
 - ▶ Square-Square: [Chopra CVPR 2005] [Hadsell CVPR 2006]:

$$\mathcal{L}(x, y, \hat{y}, w) = ([F_w(x, y)]^+)^2 + ([m(y, \hat{y}) - F_w(x, \hat{y})]^+)^2$$



Contrastive Methods: group losses

- ▶ Push down on a group of data points, push up on a group of contrastive points
- ▶ General group loss on p^+ data points and p^- contrastive points:

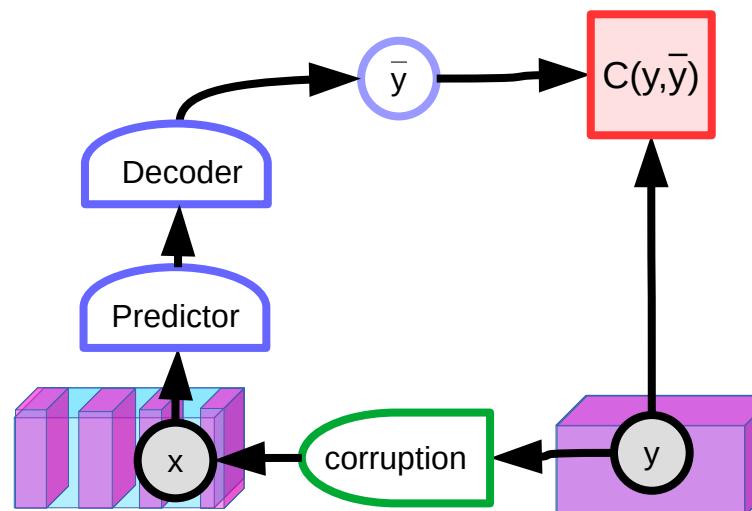
$$\mathcal{L}(x_1 \dots x_{p^+}, y_1 \dots y_{p^+}, \hat{y}_1 \dots \hat{y}_{p^-}, w) = H\left(E(x_1, y_1), \dots E(x_{p^+}, y_{p^+}), E(x_1, \hat{y}_1), \dots E(x_{p^+}, \hat{y}_{p^+}), M(Y_{1\dots p^+}, \hat{Y}_{1\dots p^-})\right)$$

- ▶ Where H must be an increasing fn of the data energies and decreasing fn of the contrastive point energies within the margin.
- ▶ M is a margin matrix for all pairs of y and \hat{y} in the group.
- ▶ **Example:** Neighborhood Component Analysis, Noise Contrastive Estimation (implicit infinite margin) [Goldberger 2005] [Gutmann 2010]...[Misra 2019] [Chen 2020]

$$\mathcal{L}(x, y, \hat{y}_1, \dots, \hat{y}_{p^-}, w) = \frac{e^{-E_w(x, y)}}{e^{-E_w(x, y)} + \sum_{i=1}^{p^-} e^{-E_w(x, \hat{y}_i, w)}}$$

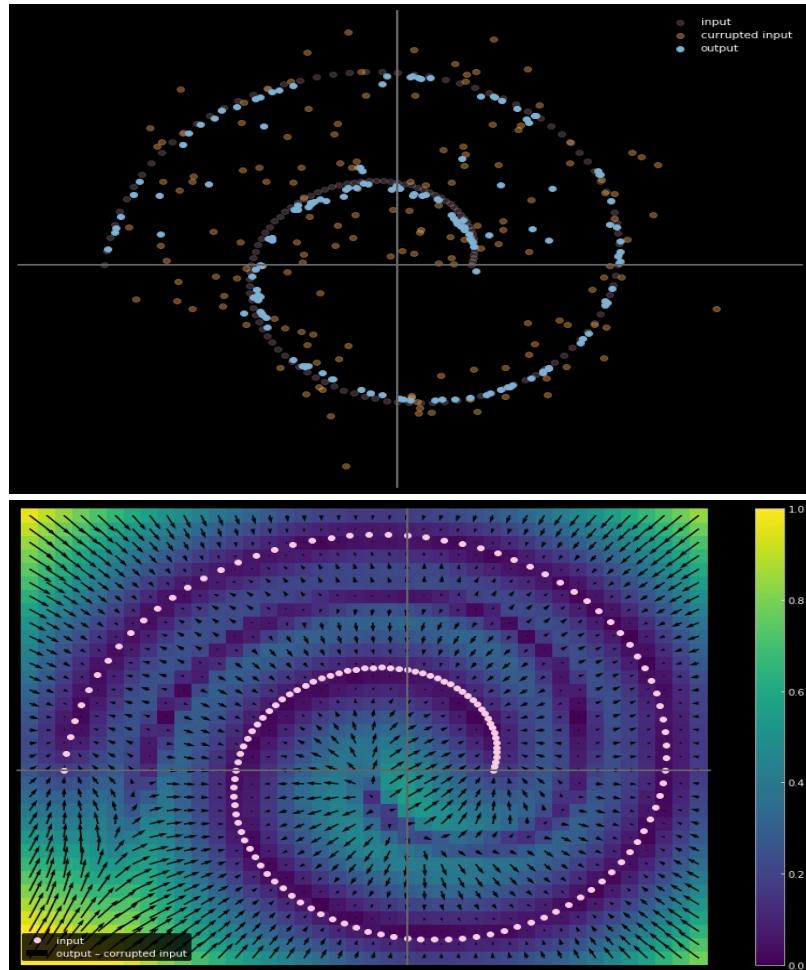
Contrastive Method: Denoising AE / Masked AE

- ▶ **Denoising AE [Vincent 2008] / Masked AE**
- ▶ Learning text representations
- ▶ [Collobert-Weston 2011], BERT, RoBERTa...



This is a [...] of text extracted
[...] a large set of [...] articles

This is a piece of text extracted
from a large set of news articles



Figures: Alfredo Canziani

The Blender Chatbot

- ▶ From Facebook
- ▶ A large-scale language model
- ▶ Pre-trained in a self-supervised manner
- ▶ Trained on live dialogs
- ▶ Open source

If you could have dinner with one person, dead or alive, who would that be?

That's a tough one. I think I'd have to go with Steve Jobs. I'd love to pick his brain.

Ah, interesting choice. What do you know about Steve Jobs?

He was one of the most influential people in the history of technology. He was a visionary.

What questions would you ask him?

I'd want to know how he came up with the idea for the iPod. It was so revolutionary at the time.

Totally...the iPod changed the way everyone listens to music.

It really did. It's crazy to think that it was released in 2001. That was a long time ago.

Omg wow, that makes me feel old!

I know, right? Time flies by so fast. I wish I could go back to those simpler times.

Me too, friend, me too. Speaking of music, what's your favorite album to listen to lately?

I've been listening to a lot of Led Zeppelin lately. They're my favorite band.

Supervised Symbol Manipulation

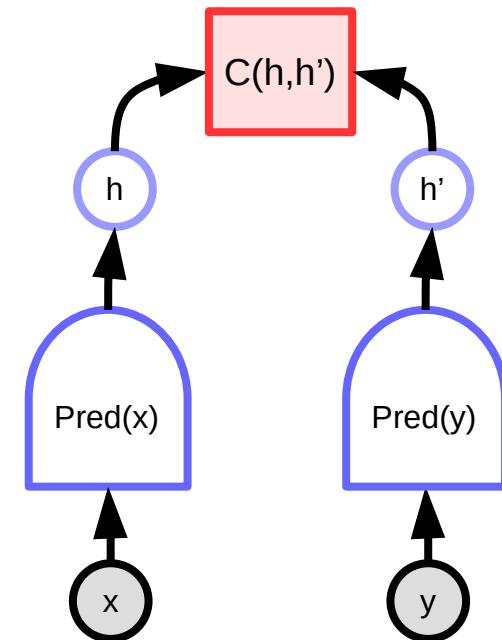
- ▶ Solving integrals and differential equations symbolically with a transformer architecture
- ▶ [Lample & Charton
arXiv:1912.01412]
- ▶ Accuracy on various problems →

	Integration (BWD)	ODE (order 1)	ODE (order 2)
Mathematica (30s)	84.0	77.2	61.6
Matlab	65.2	-	-
Maple	67.4	-	-
Beam size 1	98.4	81.2	40.8
Beam size 10	99.6	94.0	73.2
Beam size 50	99.6	97.0	81.0

Equation	Solution
$y' = \frac{16x^3 - 42x^2 + 2x}{(-16x^8 + 112x^7 - 204x^6 + 28x^5 - x^4 + 1)^{1/2}}$	$y = \sin^{-1}(4x^4 - 14x^3 + x^2)$
$3xy \cos(x) - \sqrt{9x^2 \sin(x)^2 + 1}y' + 3y \sin(x) = 0$	$y = c \exp(\sinh^{-1}(3x \sin(x)))$
$4x^4yy'' - 8x^4y'^2 - 8x^3yy' - 3x^3y'' - 8x^2y^2 - 6x^2y' - 3x^2y'' - 9xy' - 3y = 0$	$y = \frac{c_1 + 3x + 3 \log(x)}{x(c_2 + 4x)}$

Contrastive Embedding

- ▶ Distance measured in feature space
- ▶ Multiple “predictions” through feature invariance
- ▶ Siamese nets, metric learning
 - ▶ [Bromley NIPS'93] [Chopra CVPR'05] [Hadsell CVPR'06]
- ▶ Advantage: no pixel-level reconstruction
- ▶ Difficulty: hard negative mining
- ▶ Successful examples for images:
 - ▶ DeepFace [Taigman et al. CVPR 2014]
 - ▶ PIRL [Misra et al. Arxiv:1912.01991]
 - ▶ MoCo [He et al. Arxiv:1911.05722]
 - ▶ SimCLR [Chen et al. Arxiv:2002.05709]
- ▶ Video / Audio
- ▶ Temporal proximity [Taylor CVPR 2011]



Positive pair:
Make F small



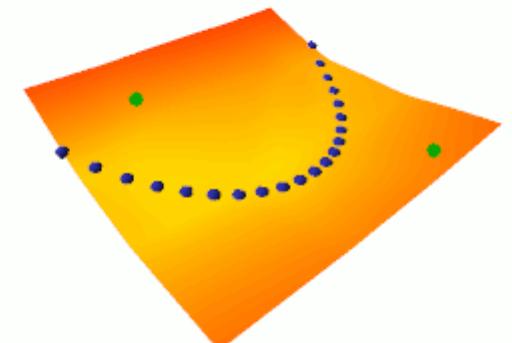
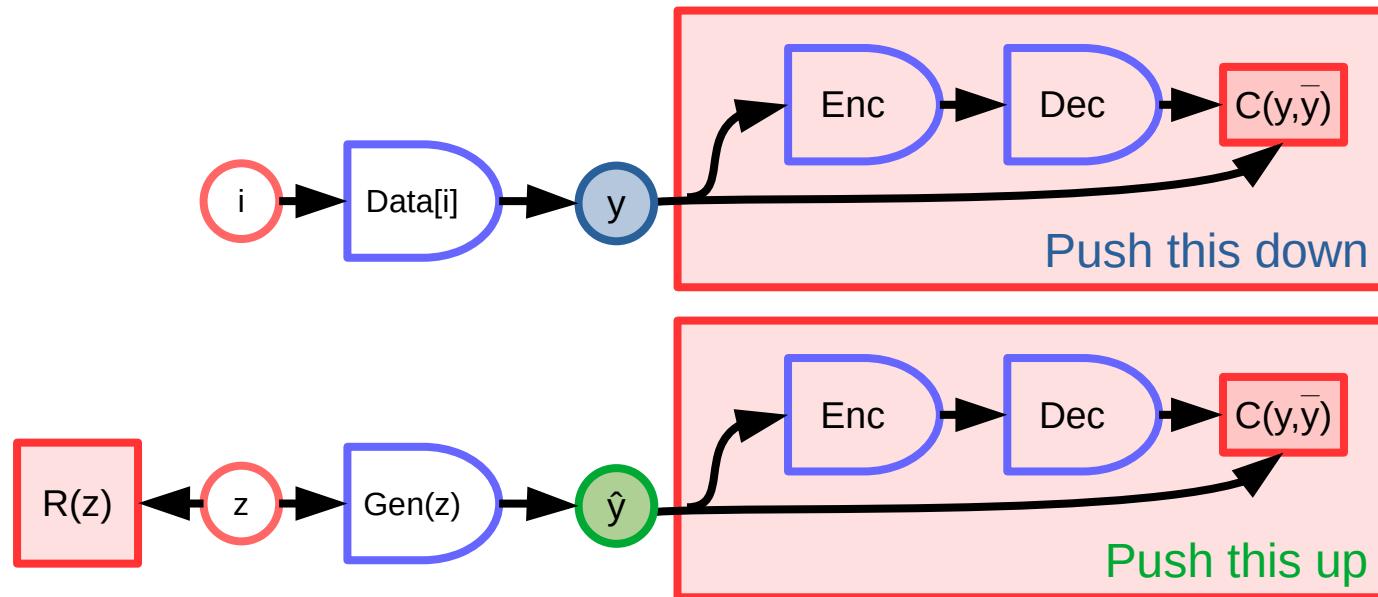
Negative pair:
Make F large



GANs: training a network to generate contrastive samples

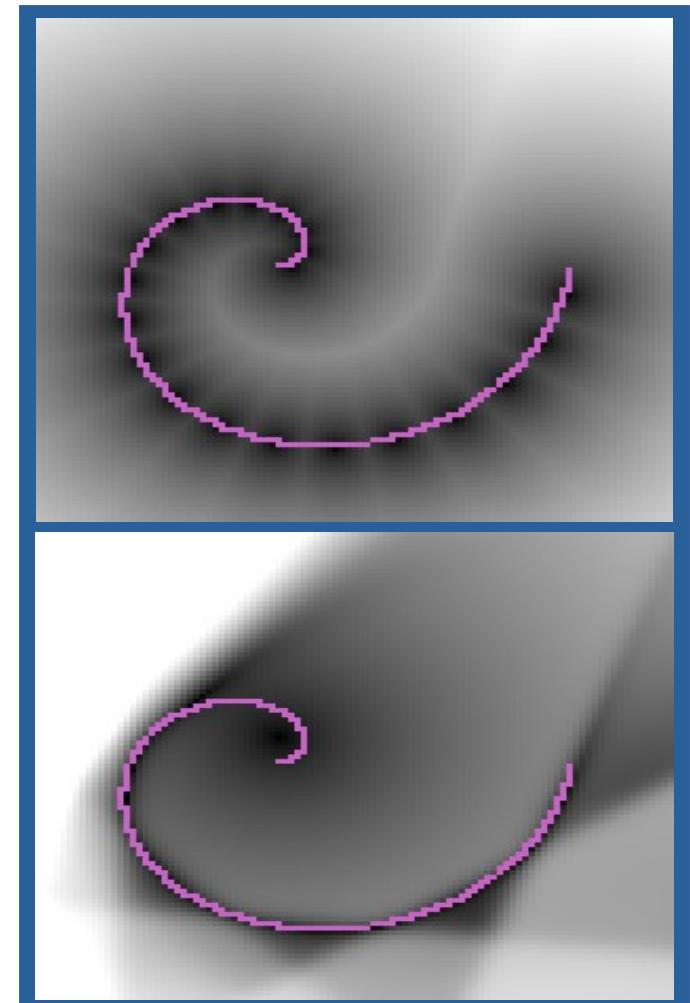
- ▶ **Energy-Based GAN** [Zhao 2016], **Wasserstein GAN** [Arjovsky 2017],...
- ▶ GANs generate nice images
- ▶ But learning representations of image has not been successful.

$$\mathcal{L}(x, y, \hat{y}, w) = H(F_w(x, y), F_w(x, \hat{y}), m(y, \hat{y}))$$



Architectural Methods & Regularized Methods

- ▶ Different ways to limit the information capacity of the representation
- ▶ A1: build the machine so that the volume of the low energy regions is bounded:
 - ▶ K-means, Gaussian Mixture Model, PCA, Bottleneck AE, Discretized AE (VQVAE),...
- ▶ A2: regularize the volume of the low energy regions:
 - ▶ Sparse coding, Sparse Auto-Encoder, LISTA, Variational Auto-Encoder.



Architectural Methods

- ▶ A1: build the machine so that the volume of the low energy regions is bounded:
- ▶ K-means, Gaussian Mixture Model, PCA, Bottleneck AE, Discretized AE (VQVAE),...

PCA

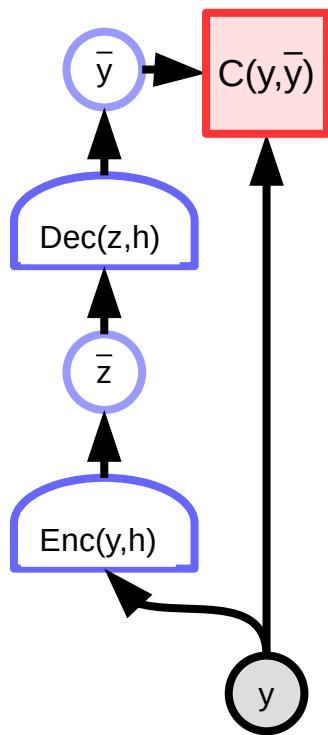
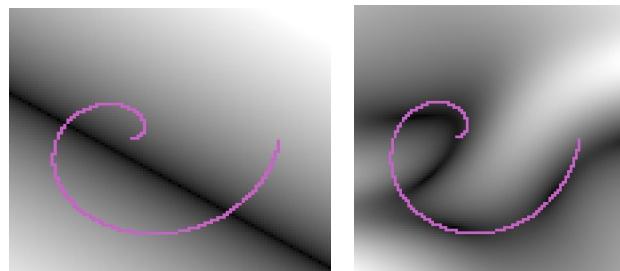
$$F(y) = \|y - w^t w y\|^2$$

Bottleneck AE

$$F(y) = C(y, \bar{y})$$

$$\bar{y} = \text{Dec}(\bar{z})$$

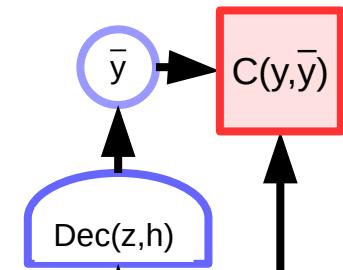
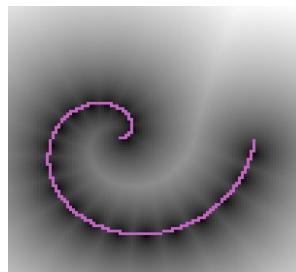
$$\bar{z} = \text{Enc}(y)$$



▶ K-means: z is a k-dim, 1-hot vector

$$E(y, z) = \|y - wz\|^2$$

$$F(y) = \min_z \|y - wz\|^2$$



Z discrete

▶ Gaussian Mixture

$$F(y) = -\log \sum_k \frac{e^{u_k}}{\sum_q e^{u_q}} e^{-\|y - w_k z_k\|^2}$$

y

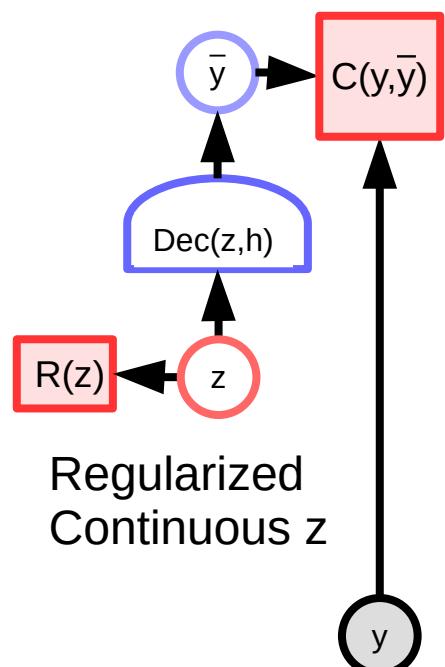
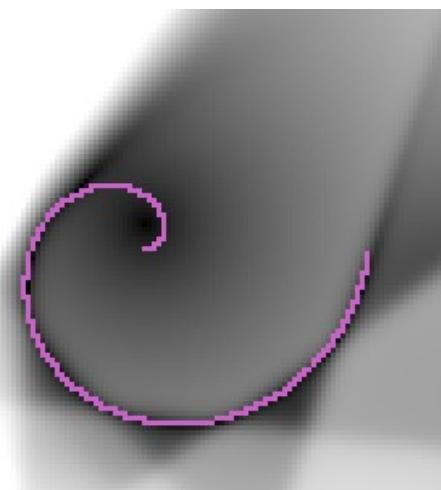
Regularized Latent Variable Methods

- A2: regularize the volume of the low energy regions

- Sparse coding

$$E(y, z) = \|y - wz\|^2 + \lambda|z|_{L1|}$$

$$F(y) = \min_z E(y, z)$$



- Regularized Auto-Encoder, Sparse AE, LISTA

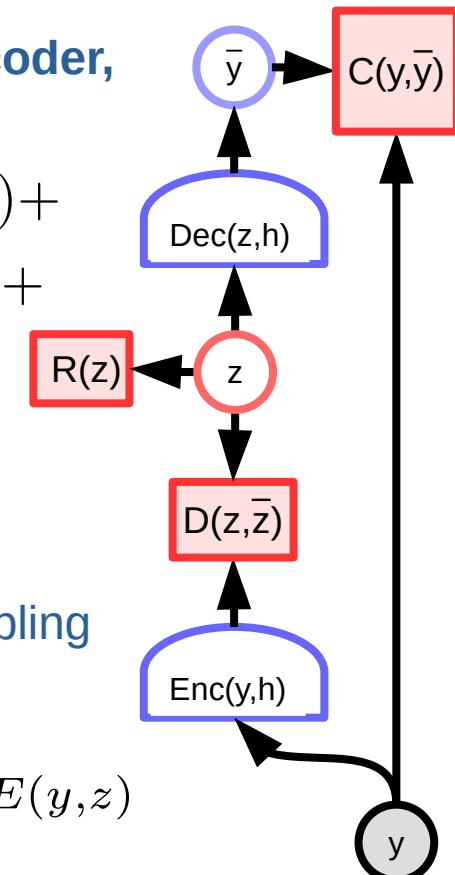
$$E(y, z) = C(y, \text{Dec}((z)) + D(z, \text{Enc}(y)) + \lambda R(z)$$

$$F(y) = \min_z E(y, z)$$

- Variational AE

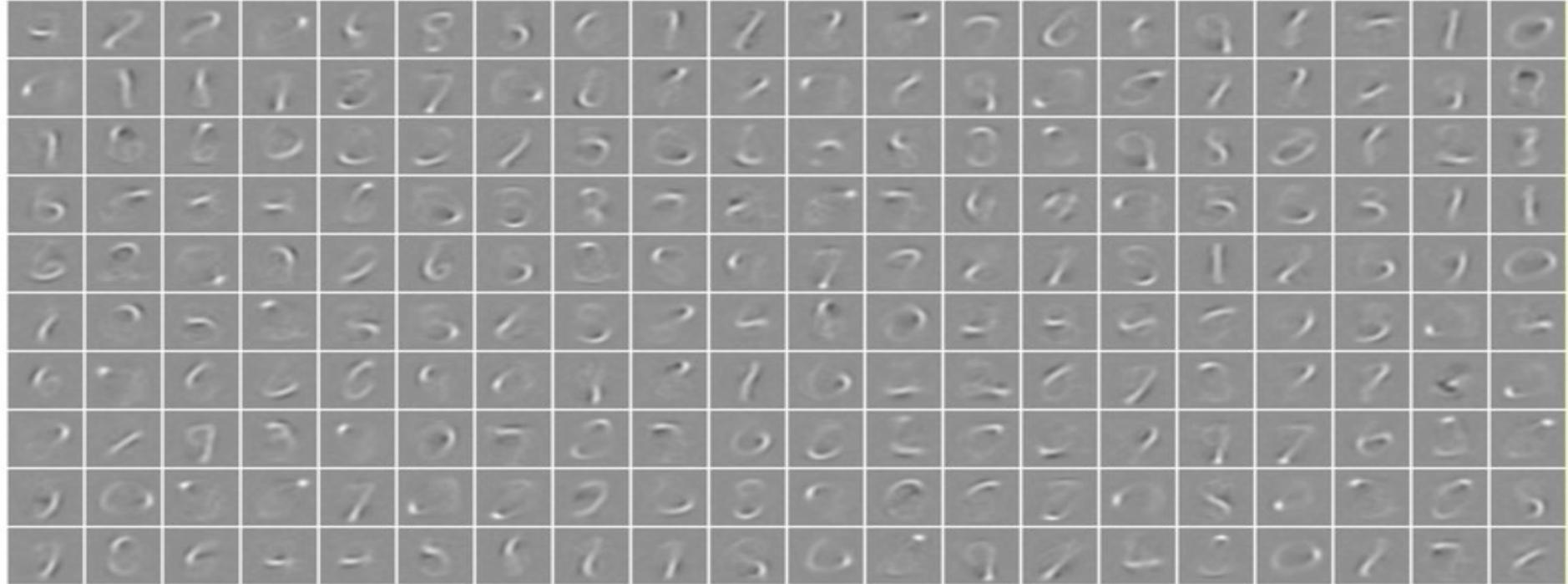
- Approximated by sampling and variational approximation

$$F(y) = -\log \int_z e^{-E(y, z)}$$



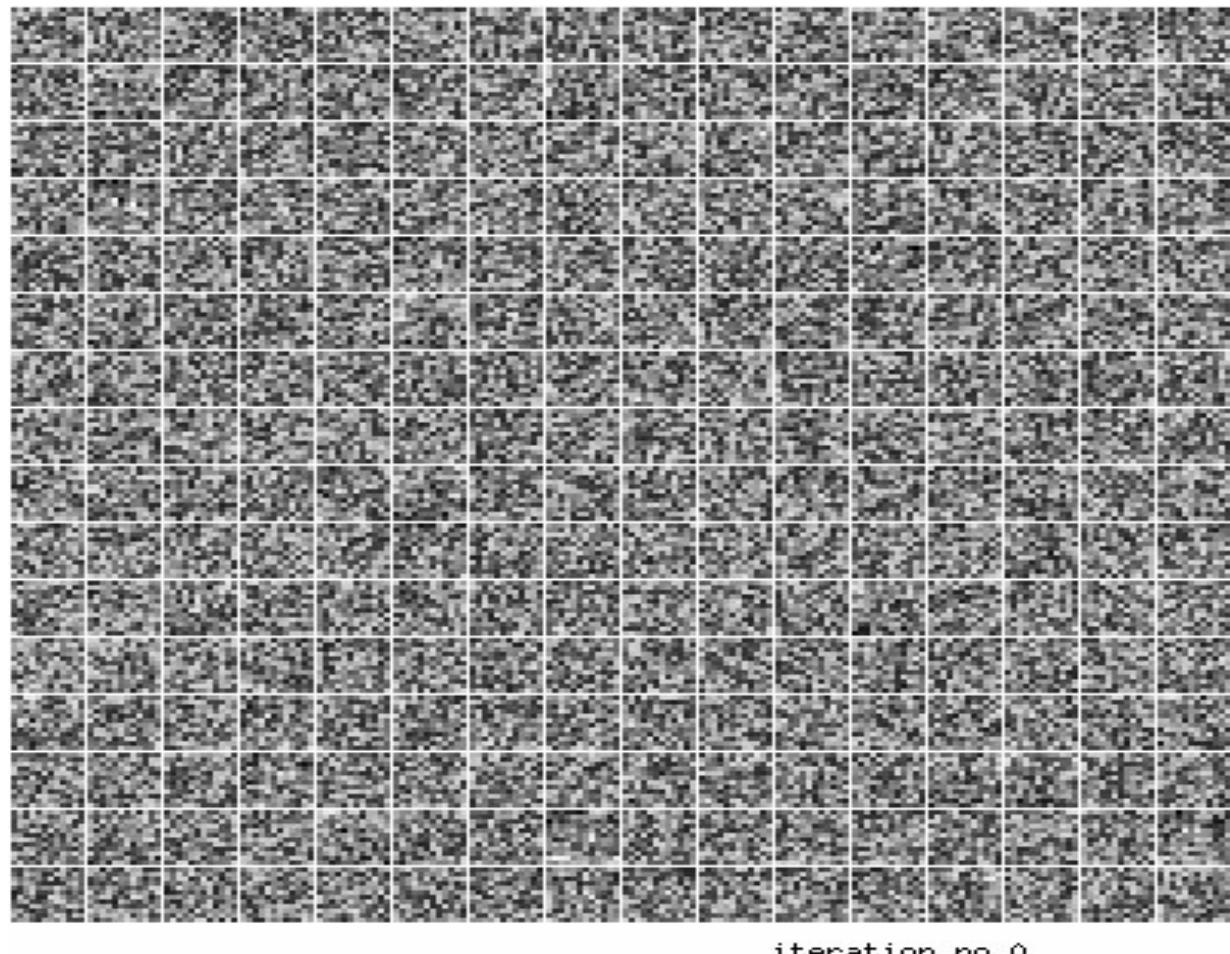
Sparse Modeling on handwritten digits (MNIST)

- Basis functions (columns of decoder matrix) are digit parts
- All digits are a linear combination of a small number of these

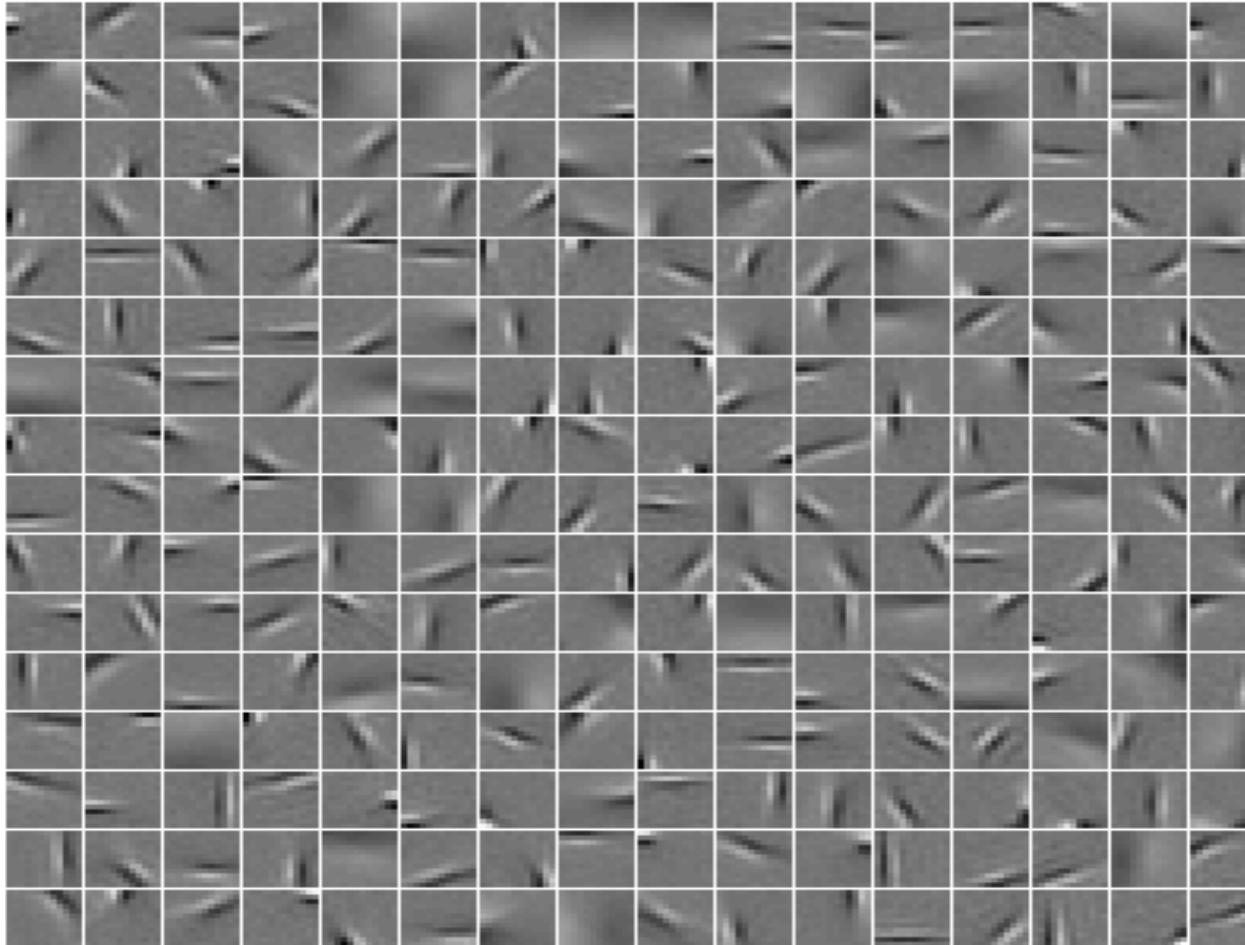


Predictive Sparse Decomposition (PSD): Training

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

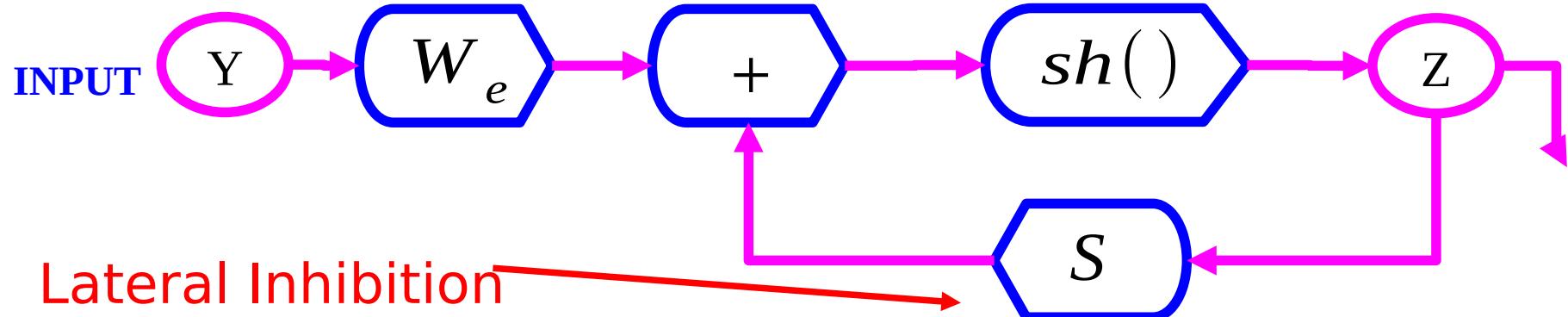


Learned Features on natural patches: V1-like receptive fields



Giving the “right” structure to the encoder

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



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

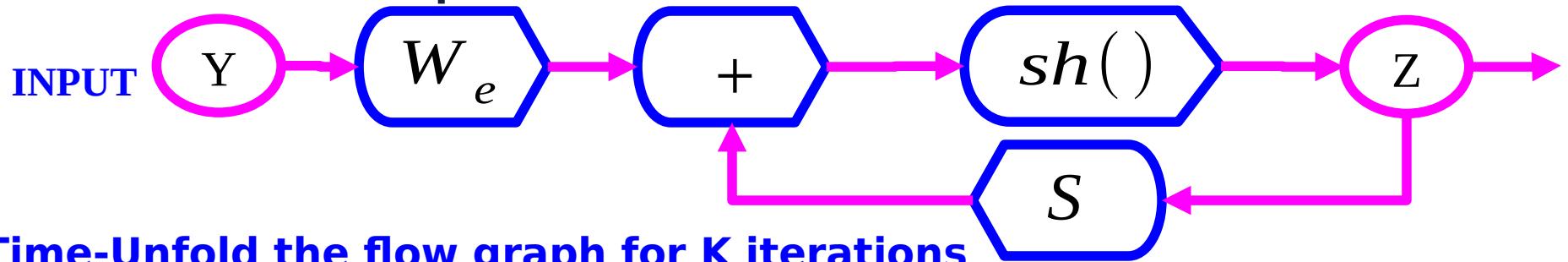
- ISTA/FastISTA reparameterized:

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

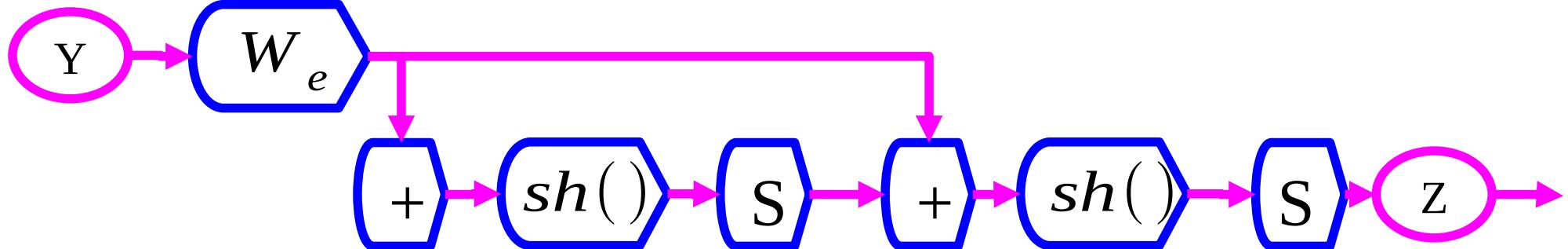
- LISTA (Learned ISTA): learn the W_e and S matrices to get fast solutions**

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

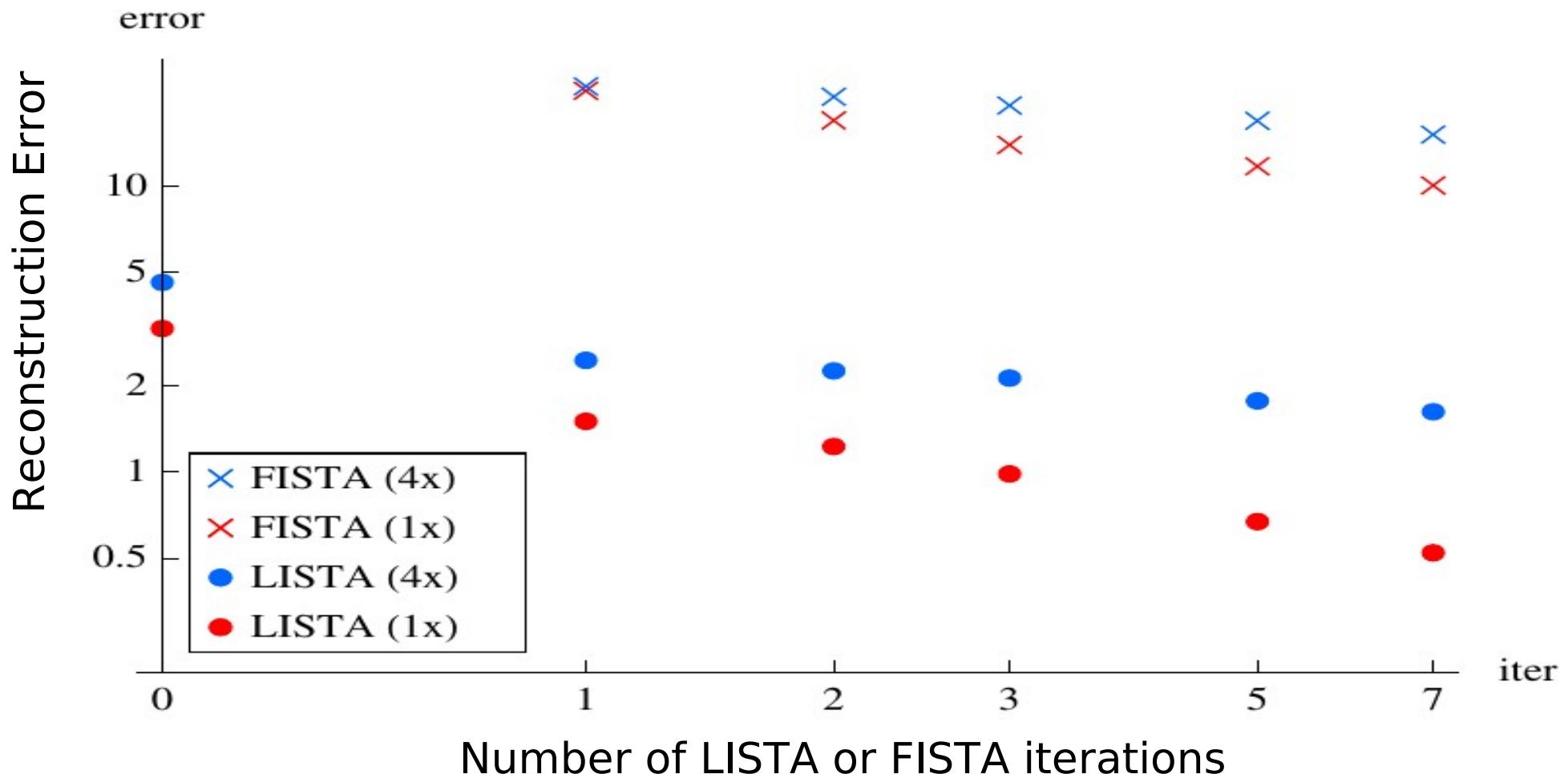
- Think of the FISTA flow graph as a recurrent neural net where We and S are trainable parameters



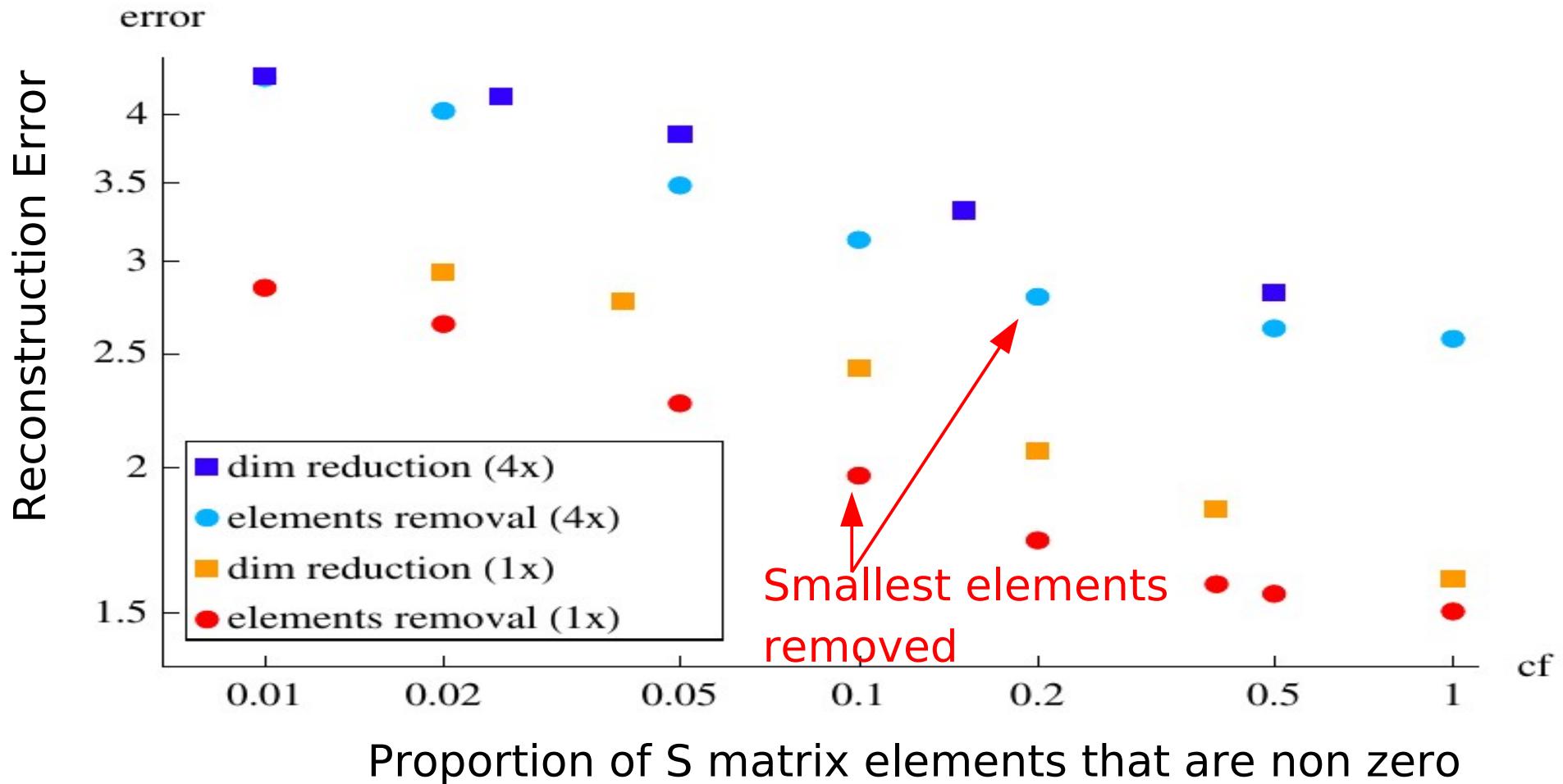
- Time-Unfold the flow graph for K iterations
- Learn the We and S matrices with “backprop-through-time”
- Get the best approximate solution within K iterations



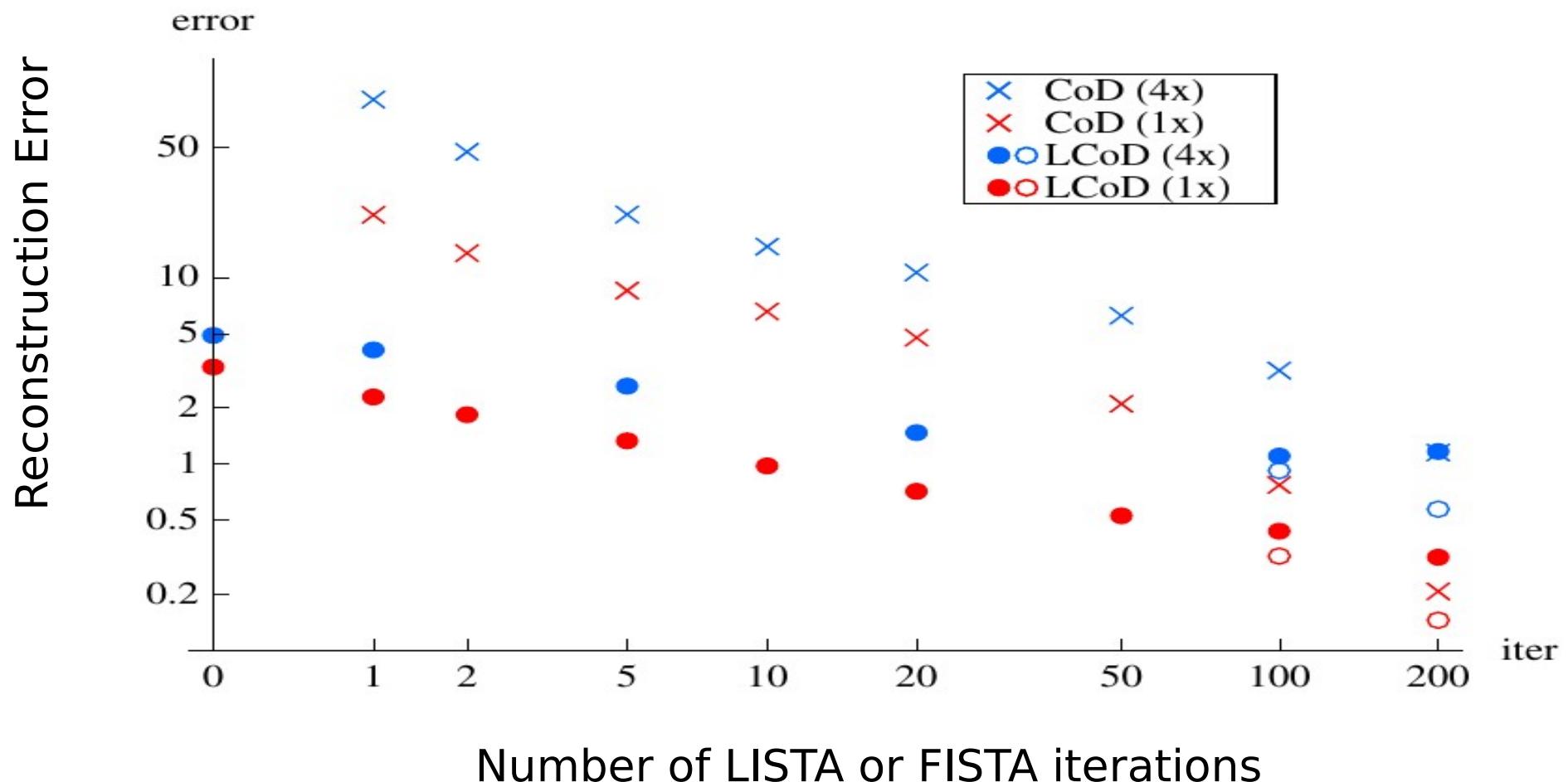
Learning ISTA (LISTA) vs ISTA/FISTA



LISTA with partial mutual inhibition matrix



Learning Coordinate Descent (LcoD): faster than LISTA



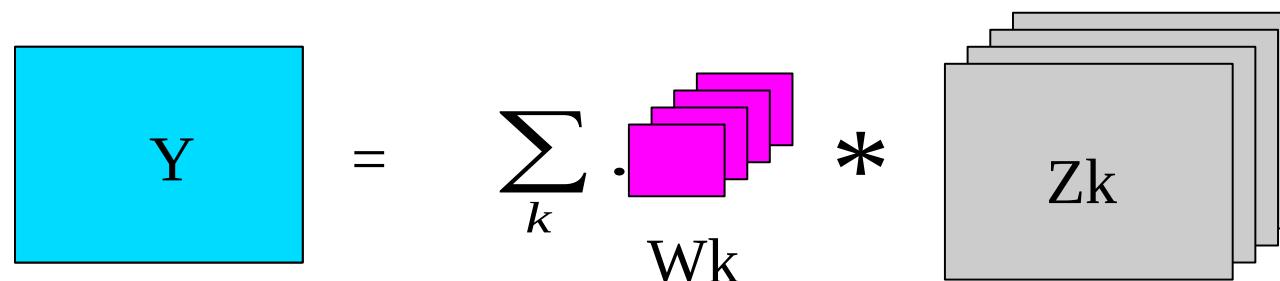
Convolutional Sparse Coding

Replace the dot products with dictionary element by convolutions.

- ▶ Input Y is a full image
- ▶ Each code component Z_k is a feature map (an image)
- ▶ Each dictionary element is a convolution kernel

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

Convolutional S.C. $E(Y, Z) = \sum_k |Y - \sum_k W_k * Z_k|^2 + \alpha \sum_k |Z_k|$

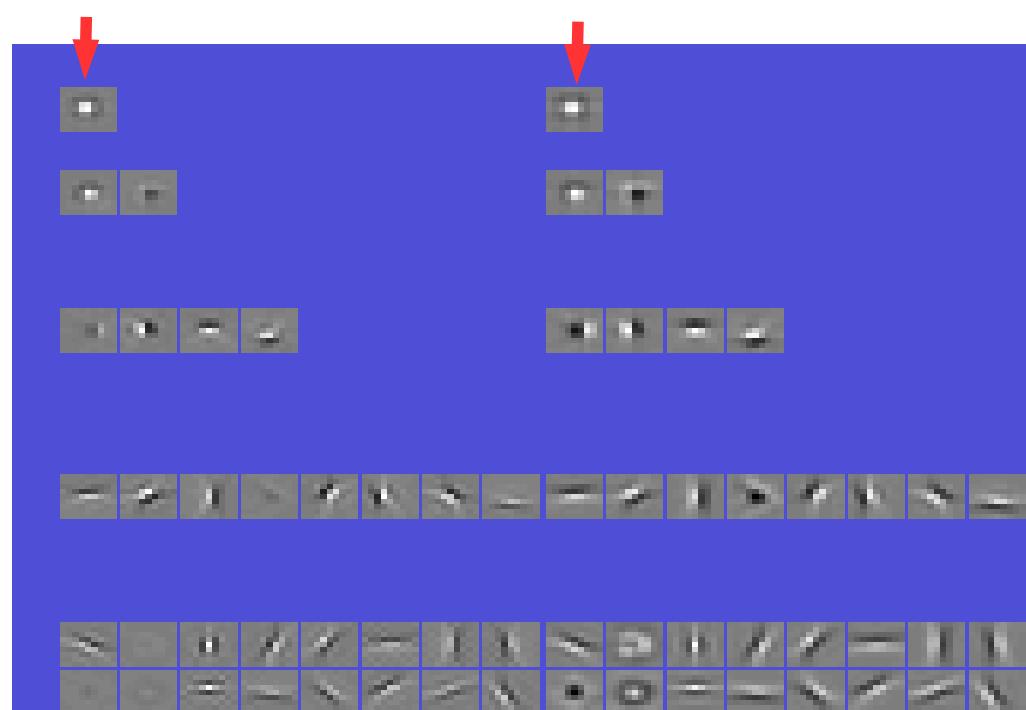


Also used in “deconvolutional networks” [Zeiler, Taylor, Fergus CVPR 2010]

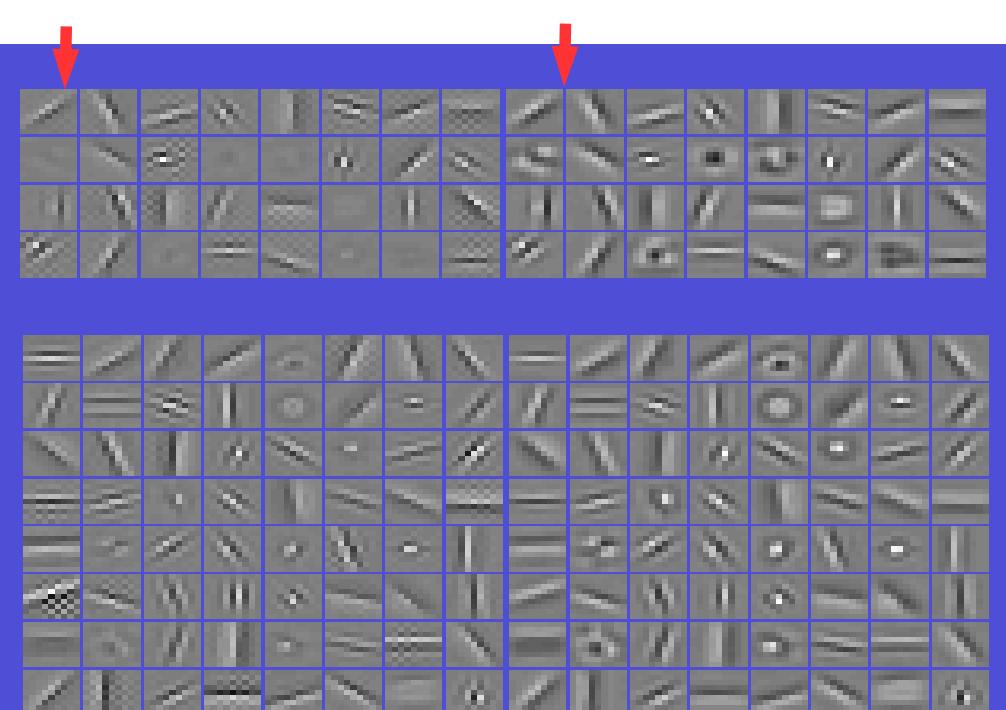
Convolutional Sparse Auto-Encoder on Natural Images

- ▶ Encoder filters and decoder filters. Decoder is linear (convolutional)
- ▶ with 1, 2, 4, 8, 16, 32, and 64 filters [Kavukcuoglu NIPS 2010]

Encoder Filters Decoder Filters Encoder Filters Decoder Filters

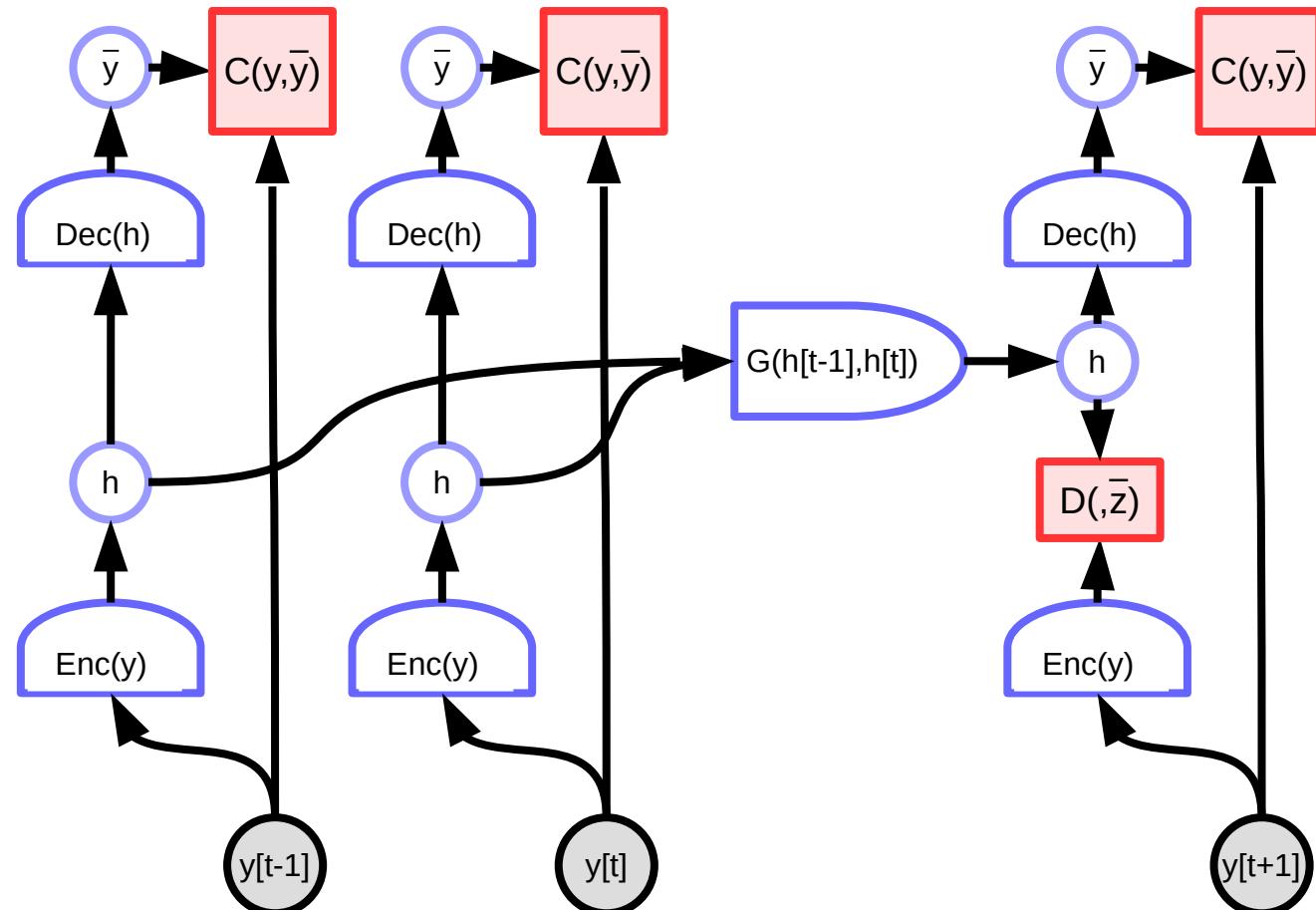


Encoder Filters Decoder Filters

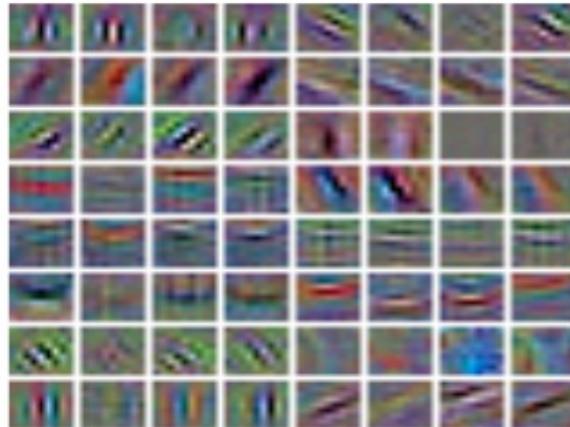


Temporal Regularization Methods

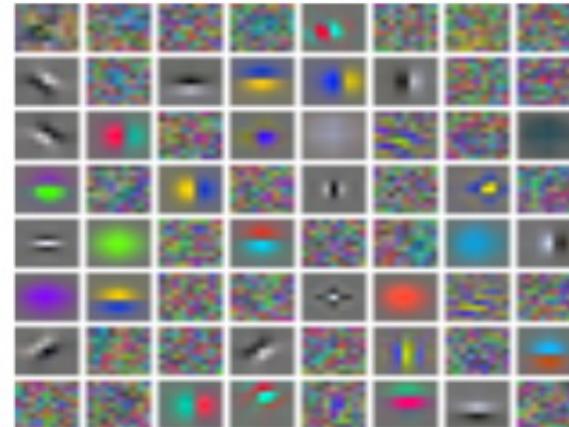
- ▶ Favors “flatness” and predictability of the representation.
 - ▶ Temporal invariance [Goroshin ICCV’15]
 - ▶ Linear predictability [Goroshin NIPS’16]
 - ▶ Minimal curvature [O. Hénaff 2019]
- ▶ Temporal proximity is an instance of similarity graph.
- ▶ Decoder alleviates need for contrastive samples



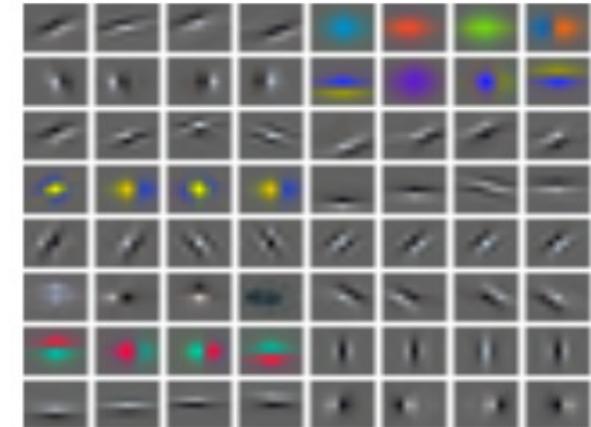
Sparse Auto-Encoder with “Slow Feature” Penalty



▶ Supervised filters CIFAR10

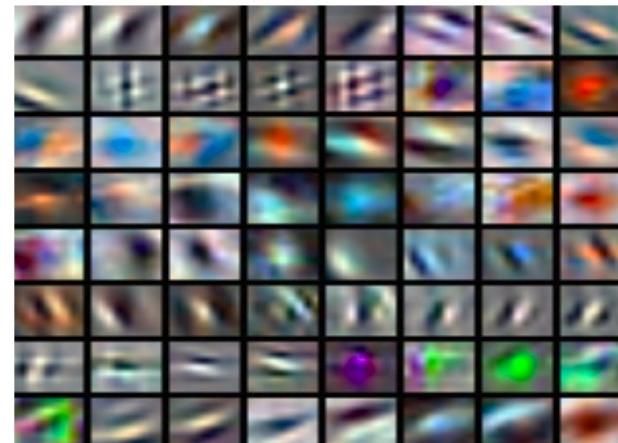
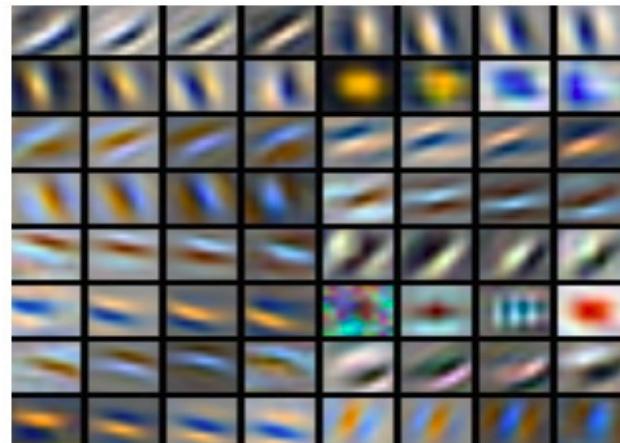


sparse conv. auto-encoder



slow & sparse convolutional AE
trained on YouTube videos

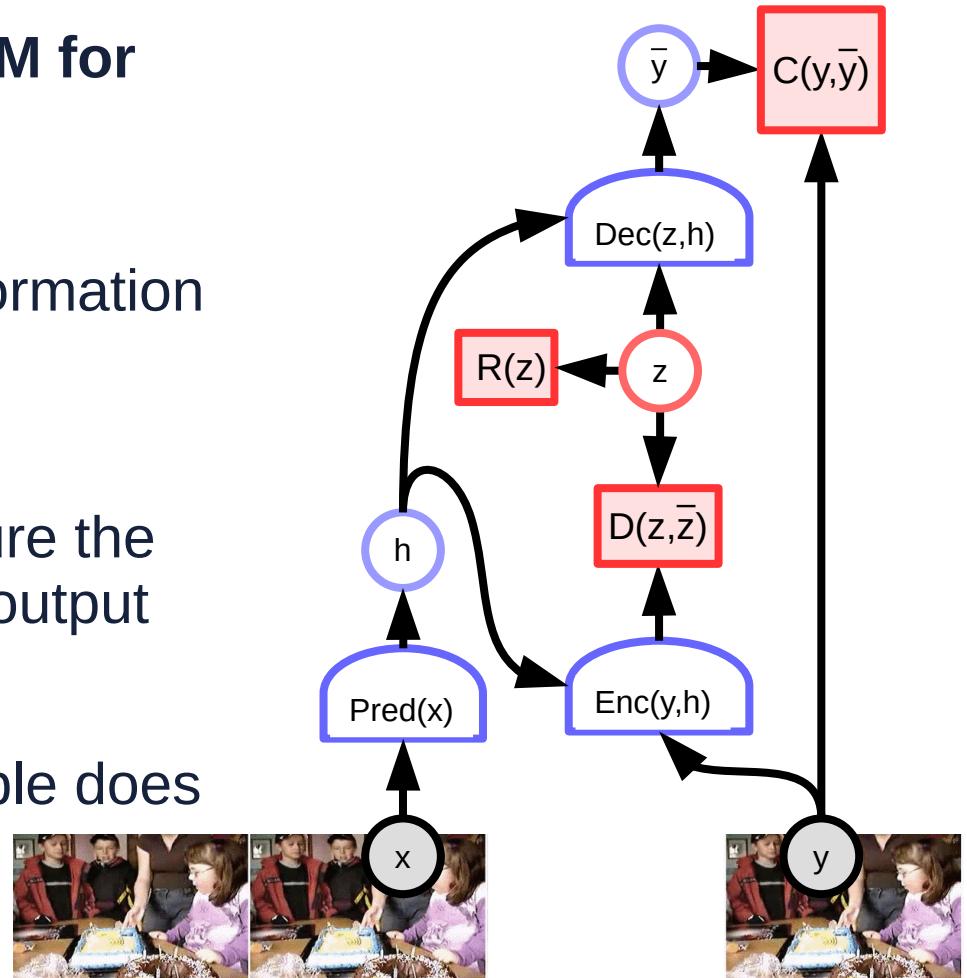
- ▶ Representation is pooled over non-overlapping groups of 4 features



- ▶ Representation is pooled over overlapping groups of 4 features

Conditional Regularized AE

- ▶ Regularized Latent Variable EBM for video Prediction
- ▶ Predictor captures the useful information from the past in h
- ▶ Regularized latent variable capture the unpredictable information in the output
- ▶ Regularizer ensures latent variable does not capture all the information.

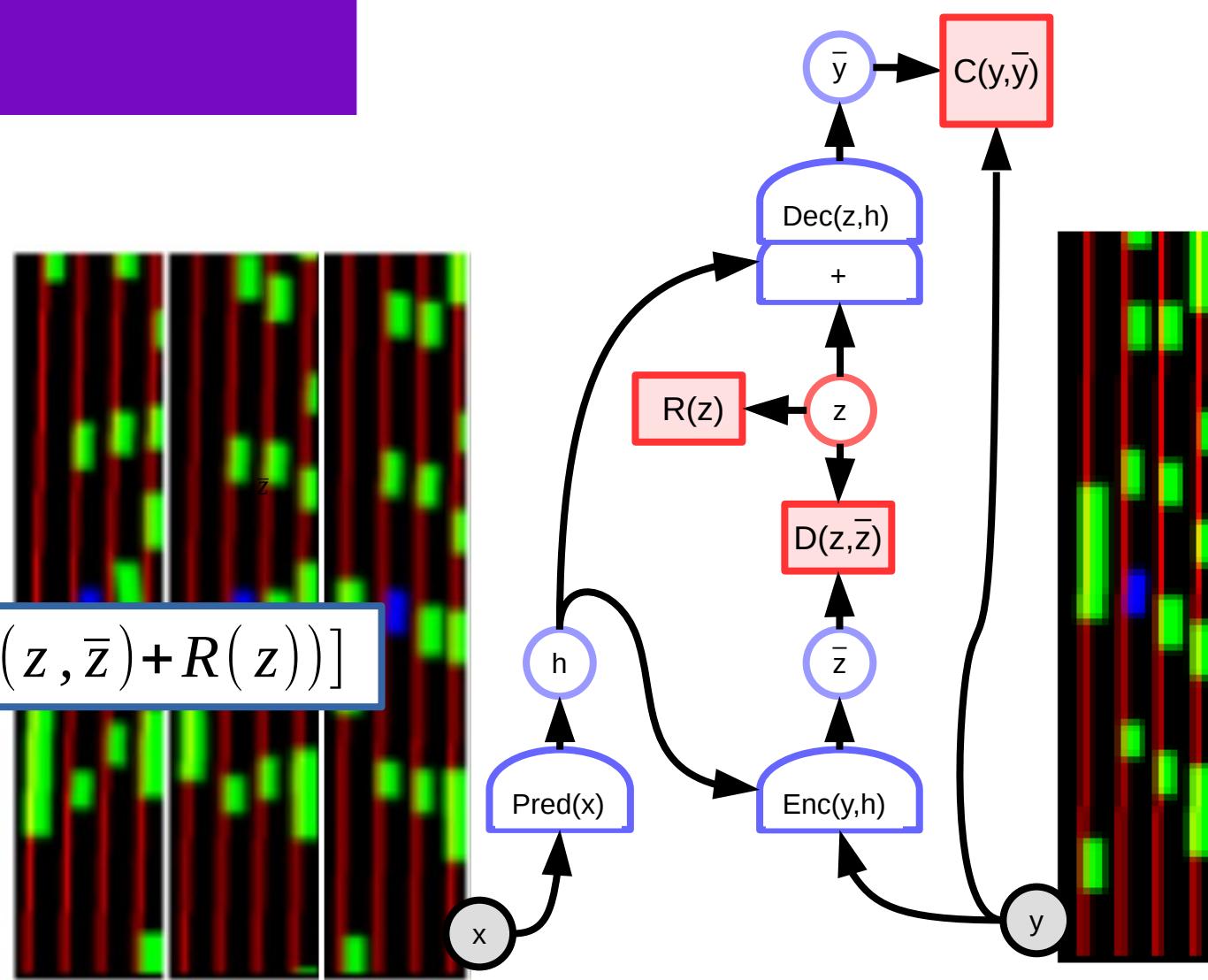


VAE + Drop Out

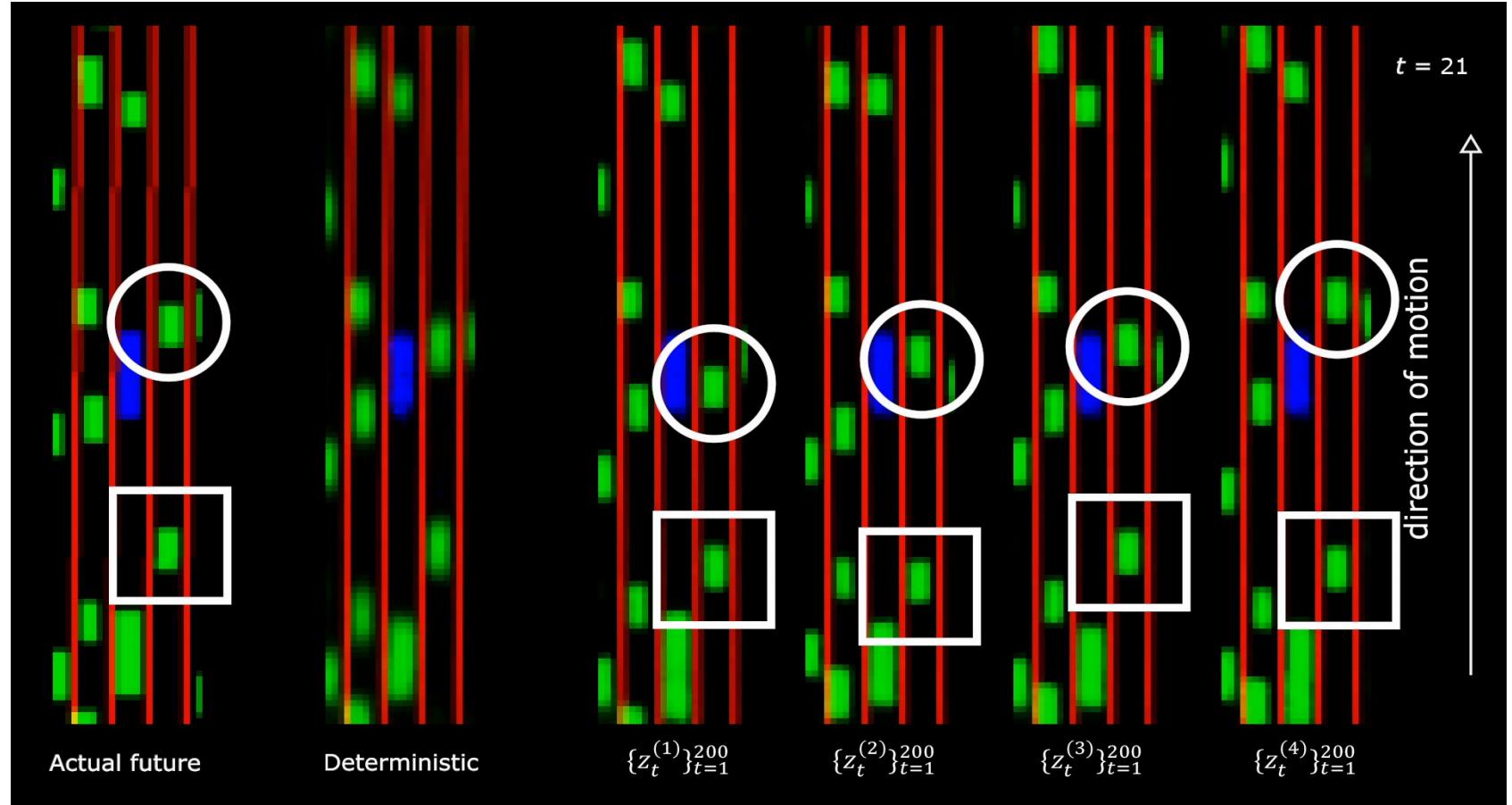
- ▶ **Training:**
 - ▶ Observe frames
 - ▶ Compute h
 - ▶ Predict \bar{z} from encoder
 - ▶ Sample z , with:

$$P(z|\bar{z}) \propto \exp[-\beta(D(z, \bar{z}) + R(z))]$$

- ▶ Half the time, set $z=0$
- ▶ Predict next frame
- ▶ backprop

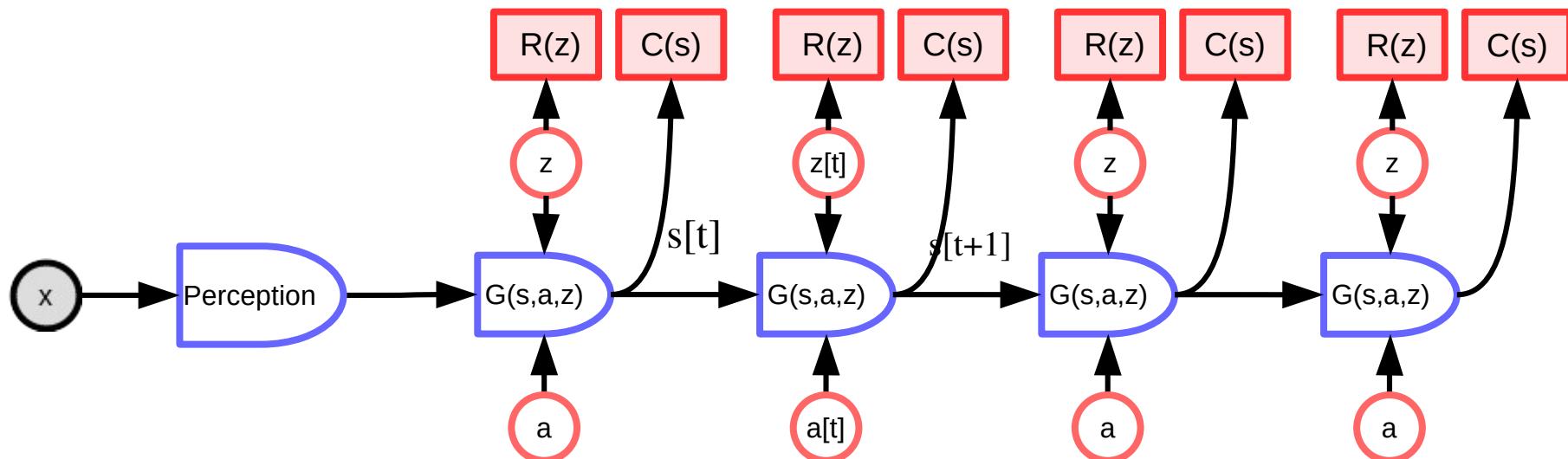


Actual, Deterministic, VAE+Dropout Predictor/encoder



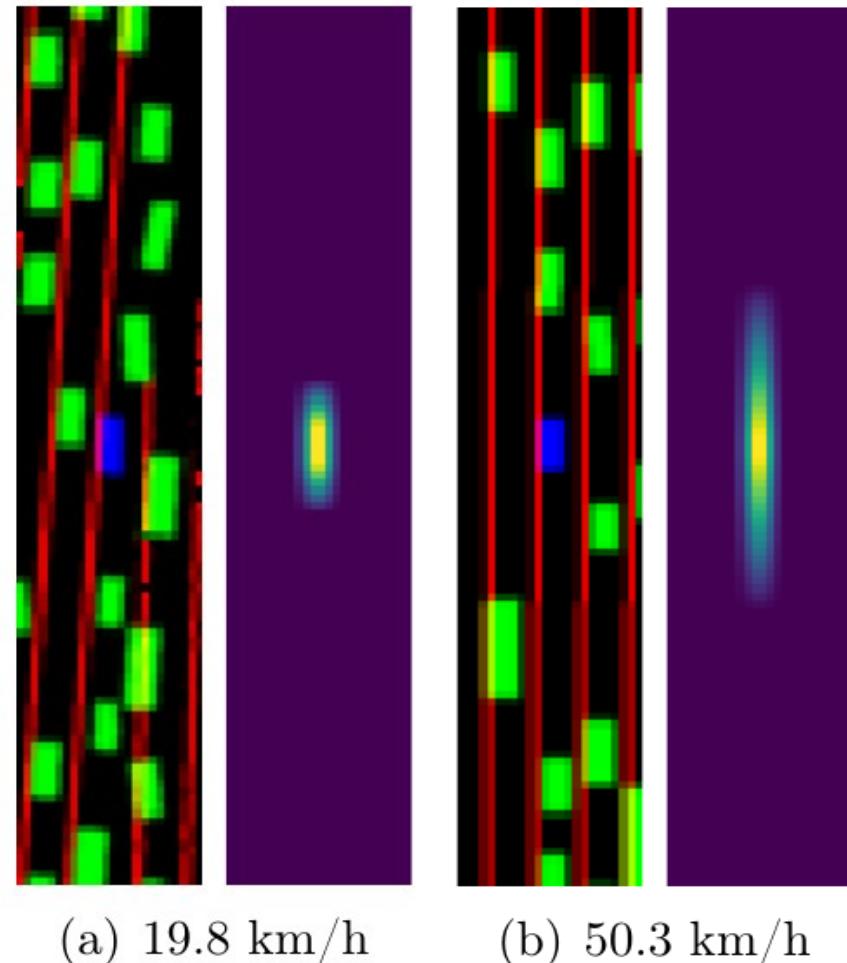
Forward Model for Model-Predictive Control

- ▶ Forward model: $s[t+1] = G(s[t], a[t], z[t])$
- ▶ Cost/Energy: $f[t] = C(s[t])$
- ▶ Latent variable z sampled from $q(z)$ proportional to $\exp(-R(z))$
- ▶ Optimize $(a[1], a[2], \dots, a[T]) = \operatorname{argmin} \sum_t C(s[t])$
through backprop (== Kelley-Bryson adjoint state method)



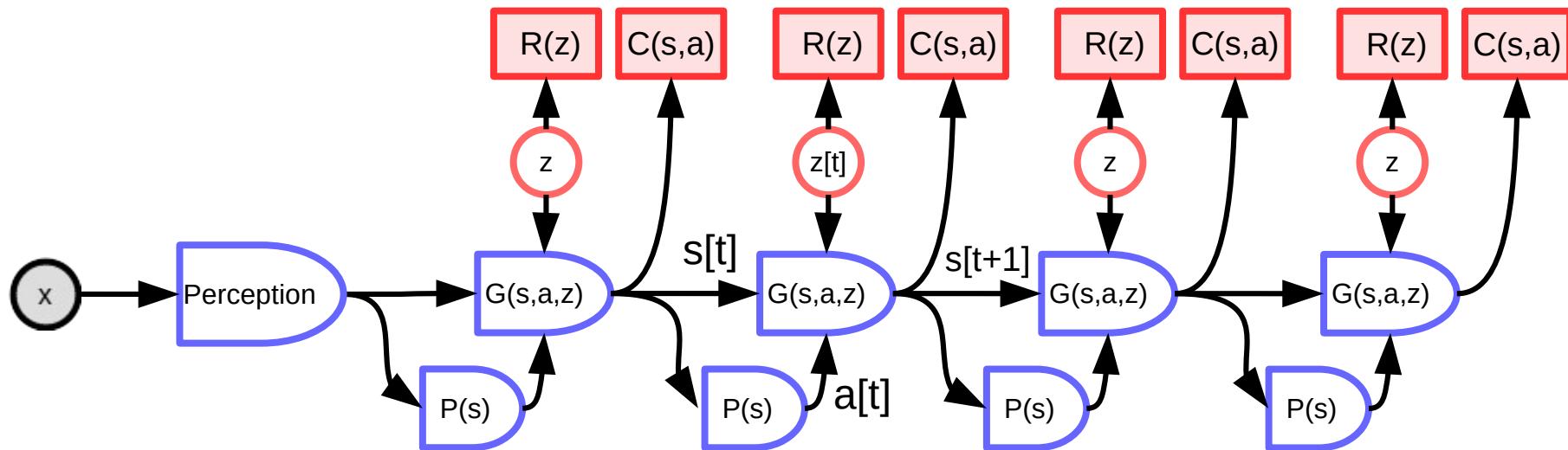
Cost optimized for Planning & Policy Learning

- ▶ **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.



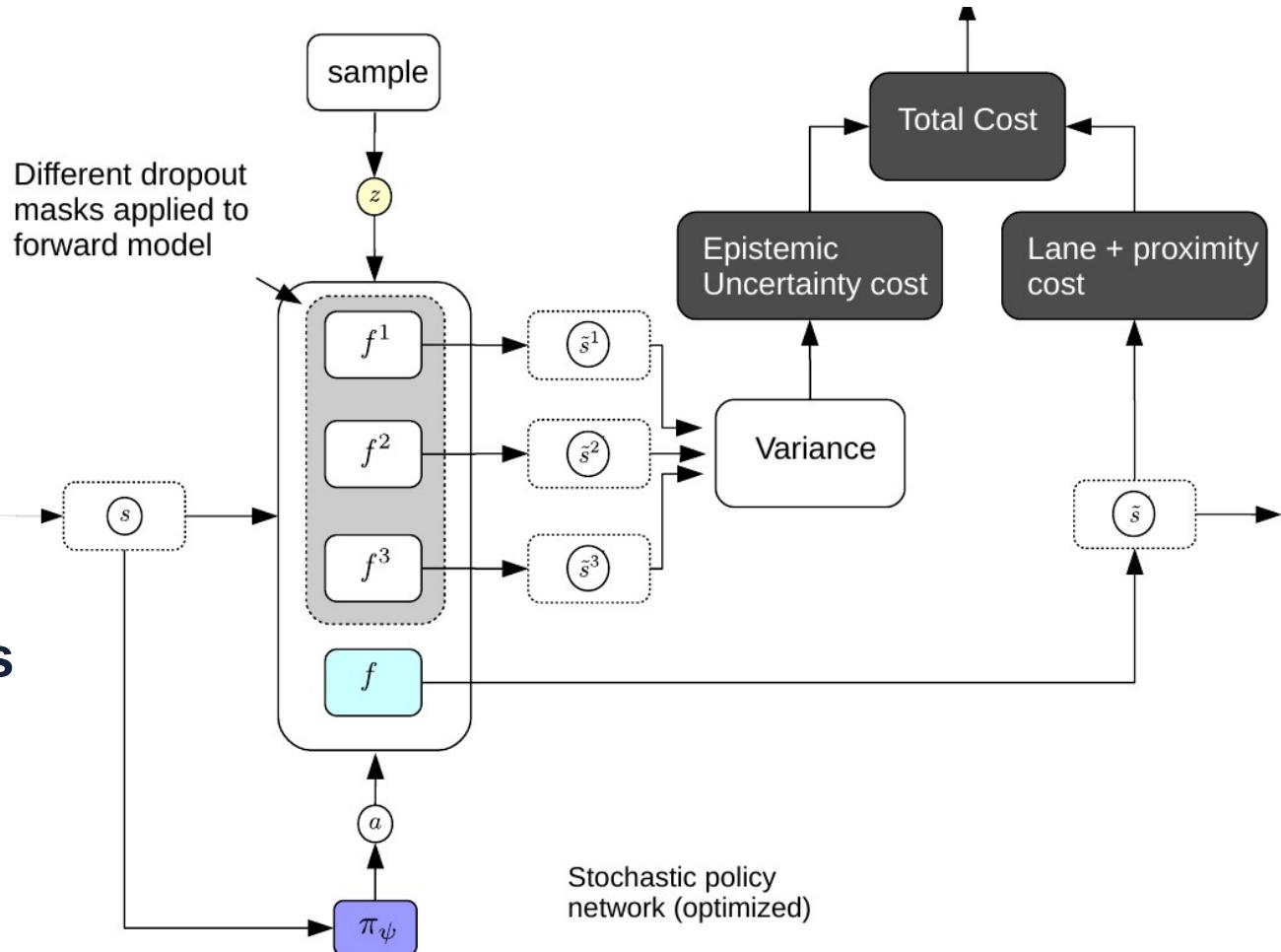
Forward Model for Gradient-Based Policy Learning

- ▶ Forward model: $s[t+1] = G(s[t], a[t], z[t])$
- ▶ Cost/Energy: $f[t] = C(s[t], a[t])$
- ▶ Latent variable z sampled from $q(z)$ proportional to $\exp(-R(z))$
- ▶ Policy: $a[t] = P(s[t])$
- ▶ Learn P through backprop (== Kelley-Bryson adjoint state method)



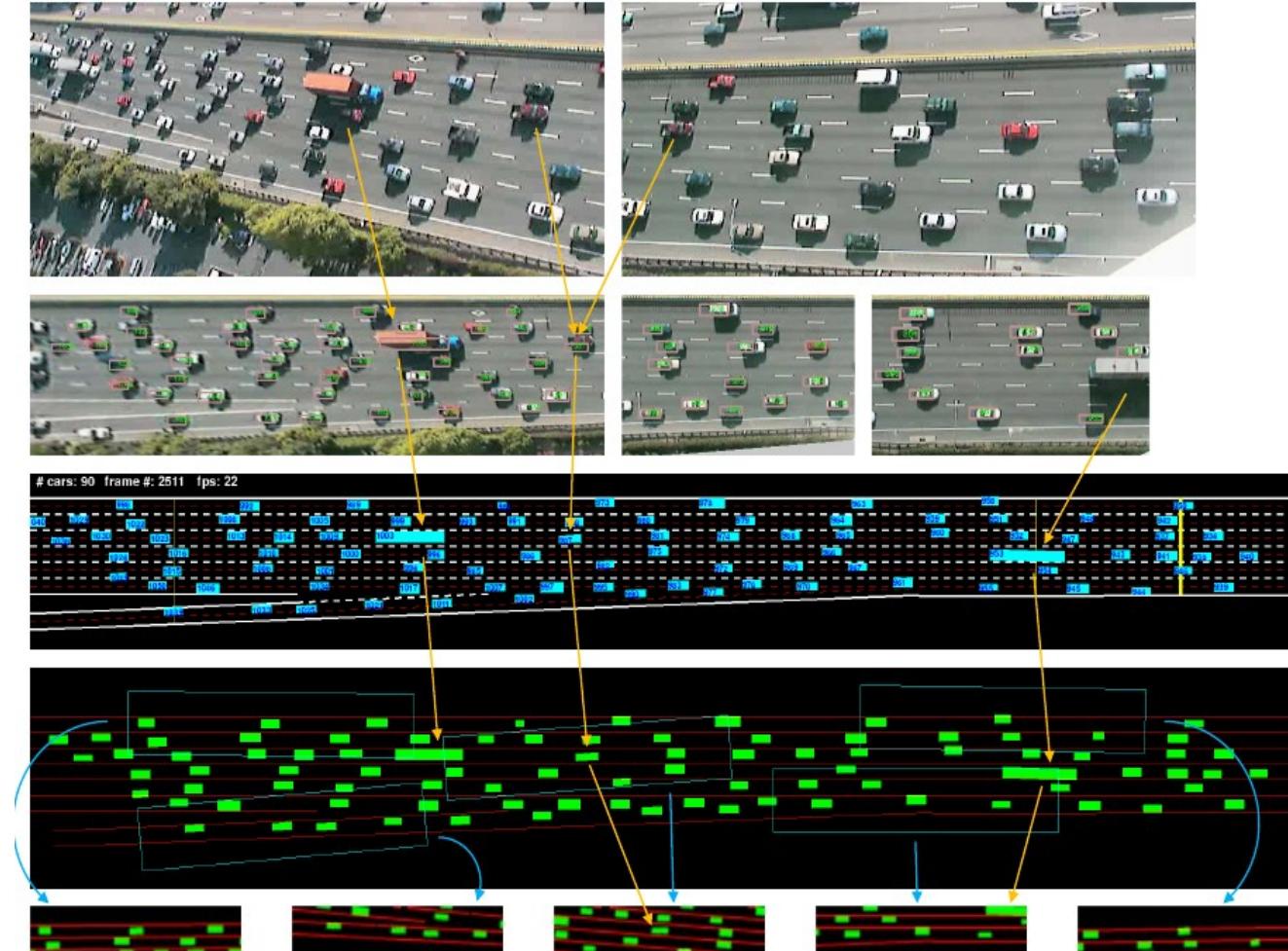
Adding an Uncertainty Cost (doesn't work without it)

- ▶ Estimates epistemic uncertainty
- ▶ Samples multiple dropouts 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

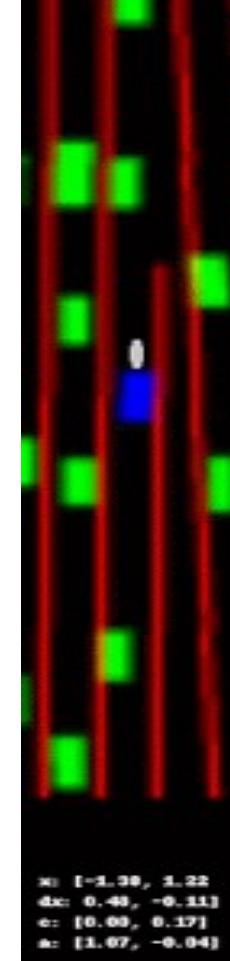
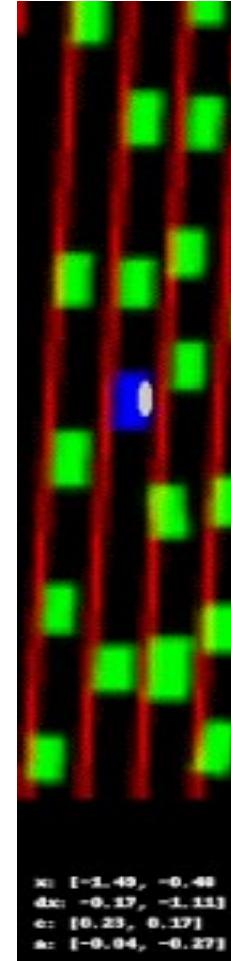
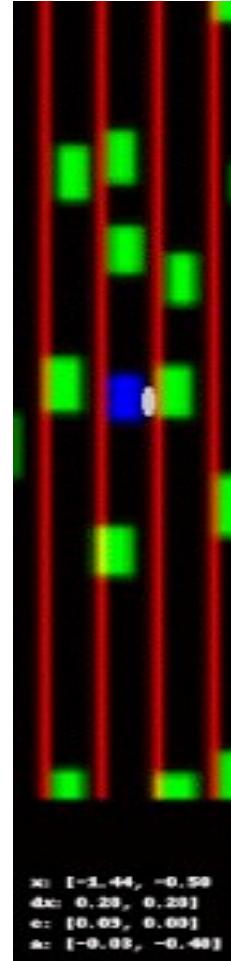
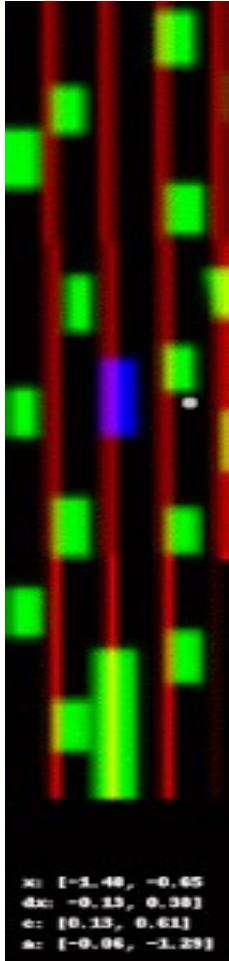
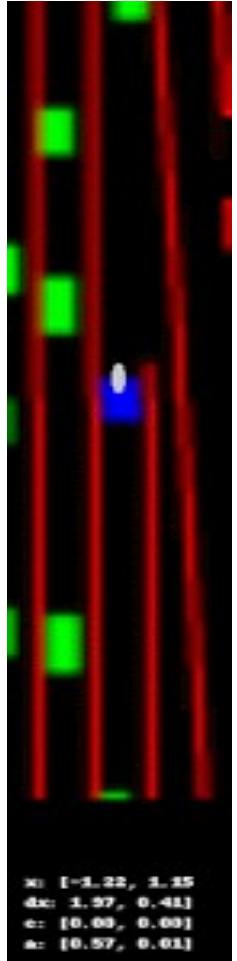


Using Forward Models to Plan (and to learn to drive)

- ▶ 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



Driving an Invisible Car in “Real” Traffic



Conclusions / Conjectures / Open Questions

- ▶ **Self-Supervised Learning: (SSL) is learning dependencies.**
- ▶ **Reasoning: is energy minimization**
- ▶ Vectors, not symbols; Latent variables to represent unknown states.

- ▶ **The main obstacle: representing uncertainty**
 - ▶ many y are compatible with an x
 - ▶ Predicting a point is insufficient. Predicting a distribution is often intractable.
- ▶ **Proposal: Energy-Based Models.**
 - ▶ Weaker than probabilistic models, but more flexible and tractable.
 - ▶ Two flavors: contrastive methods & regularized latent-variable methods.

- ▶ **My money is on Regularized Latent-Variable Energy-Based Models**

Conclusions / Conjectures / Open Questions

- ▶ Could Energy-Based SSL be a basis for **common sense**?
- ▶ Animals and humans learn (largely) self-supervised by observation
- ▶ Is the accumulation of knowledge about how the world works the basis of common sense?
- ▶ **Learning hierarchical representations of action plans.**
- ▶ We don't know how to do it.

- ▶ Oh, I almost forgot:
 - ▶ Scaling up SL or RL will **not** take us to Human-Level AI
 - ▶ There is no such thing as **AGI**. Intelligence is always specialized.
 - ▶ We should talk about rat-level, cat-level, or **human-level AI (HLAI)**.

DL: Engineering Science or Natural Science?

- ▶ **Engineering science: inventing new artifacts**
 - ▶ Telescope, steam engine, electromagnet, airplane, fertilizer, radio....
 - ▶ Methods: creation, intuition, tinkering, exploration, experimentation, happenstance....
 - ▶ guided by theoretical, conceptual, intuitive understanding.
- ▶ **Natural Science: discover, study and explain phenomena**
 - ▶ Optics, thermodynamics, electromagnetics, aerodynamics, chemistry, electronics,...
 - ▶ Methods: reproducible experiments in controlled conditions, mathematics, statistics, systematic experiments
 - ▶ guided by theoretical, conceptual, intuitive understanding.

Theory often Follows Invention

- ▶ **Telescope [1608]**
- ▶ **Steam engine [1695-1715]**
- ▶ **Electromagnetism [1820]**
- ▶ **Sailboat [???**
- ▶ **Airplane [1885-1905]**
- ▶ **Compounds [???**
- ▶ **Feedback amplifier [1927]**
- ▶ **Computer [1941-1945]**
- ▶ **Teletype [1906]**
- ▶ **Optics [1650-1700]**
- ▶ **Thermodynamics [1824-....]**
- ▶ **Electrodynamics [1821]**
- ▶ **Aerodynamics [1757]**
- ▶ **Wing theory [1907]**
- ▶ **Chemistry [1760s]**
- ▶ **Electronics [....]**
- ▶ **Computer Science [1950-1960]**
- ▶ **Information Theory [1948]**

Conclusions

- ▶ Do theory for understanding phenomena on interesting artifacts
- ▶ Don't get hypnotized by cute theory
- ▶ Don't get attracted by theoretical lampposts when the key is obviously elsewhere
- ▶ People ignore empirical results that don't fit their mental model
- ▶ Empiricism works. But extreme empiricism is inefficient
- ▶ Theory guides exploratory empiricism

Thank You!