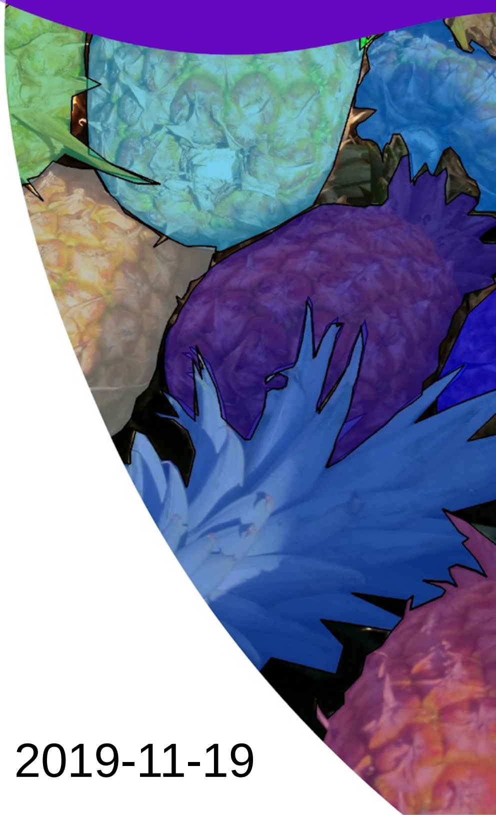
FACEBOOK AI

Energy-Based Self-Supervised Learning

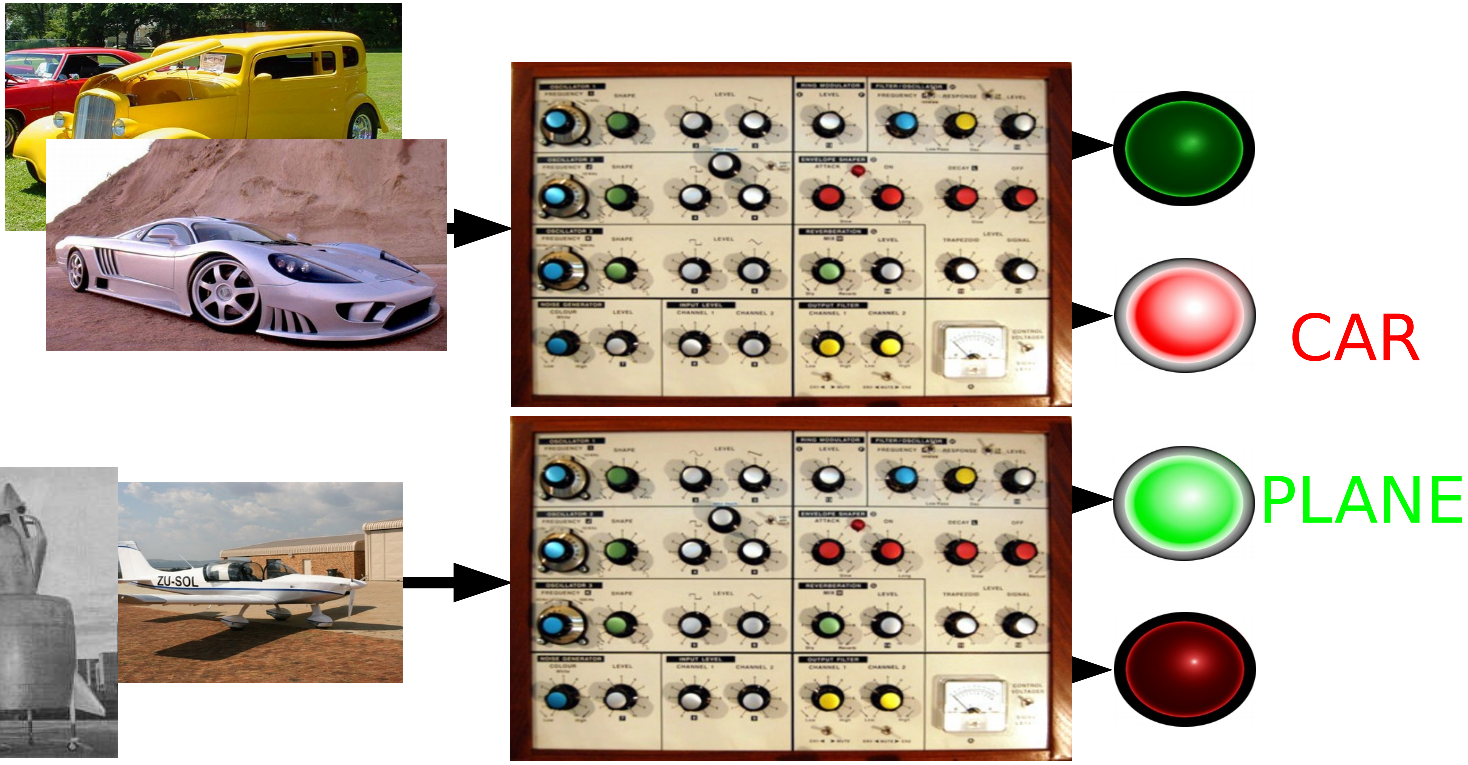
Yann LeCun
NYU - Courant Institute & Center for Data Science
Facebook AI Research
http://yann.lecun.com

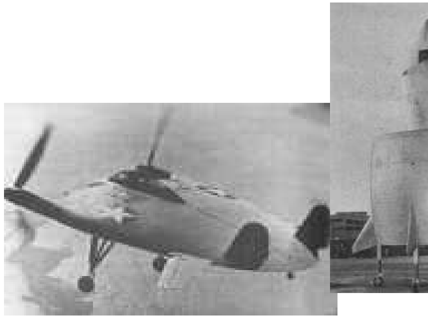


IPAM - 2019-11-19

Supervised Learning works but requires many labeled samples

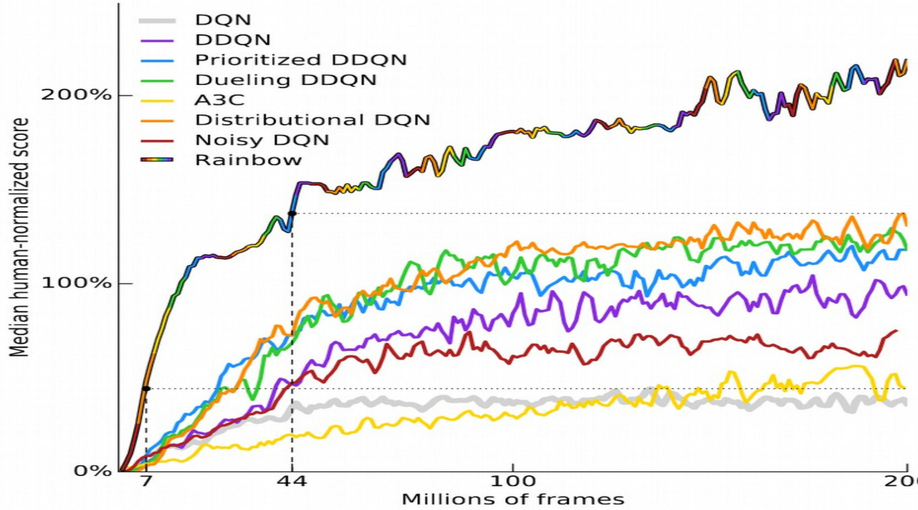
- **Training a machine by showing examples instead of programming it**
- **When the output is wrong, tweak the parameters of the machine**
- **Works well for:**
  - Speech→words
  - Image→categories
  - Portrait→ name
  - Photo→caption
  - Text→topic
  - ….

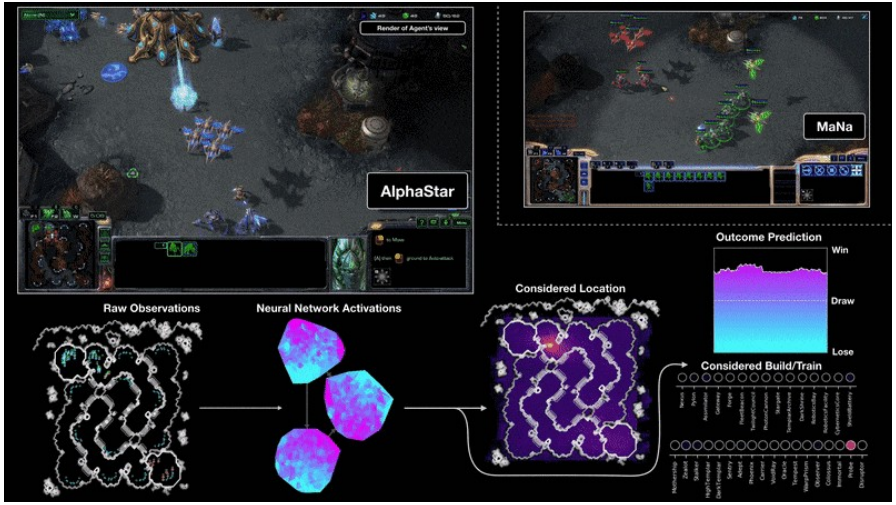




Reinforcement Learning: works great for games and simulations.

- **57 Atari games: takes 83 hours equivalent real-time (18 million frames) to reach a performance that humans reach in 15 minutes of play.**
  - [Hessel ArXiv:1710.02298]
- **Elf OpenGo v2: 20 million self-play games. (2000 GPU for 14 days)**
  - [Tian arXiv:1902.04522]
- **StarCraft: AlphaStar 200 years of equivalent real-time play**
  - [Vinyals blog post 2019]
- **OpenAI single-handed Rubik's cube**
  - 10,000 years of simulation





But RL Requires too many trials 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**

FACEBOOK AI

How do humans and animals learn so quickly?

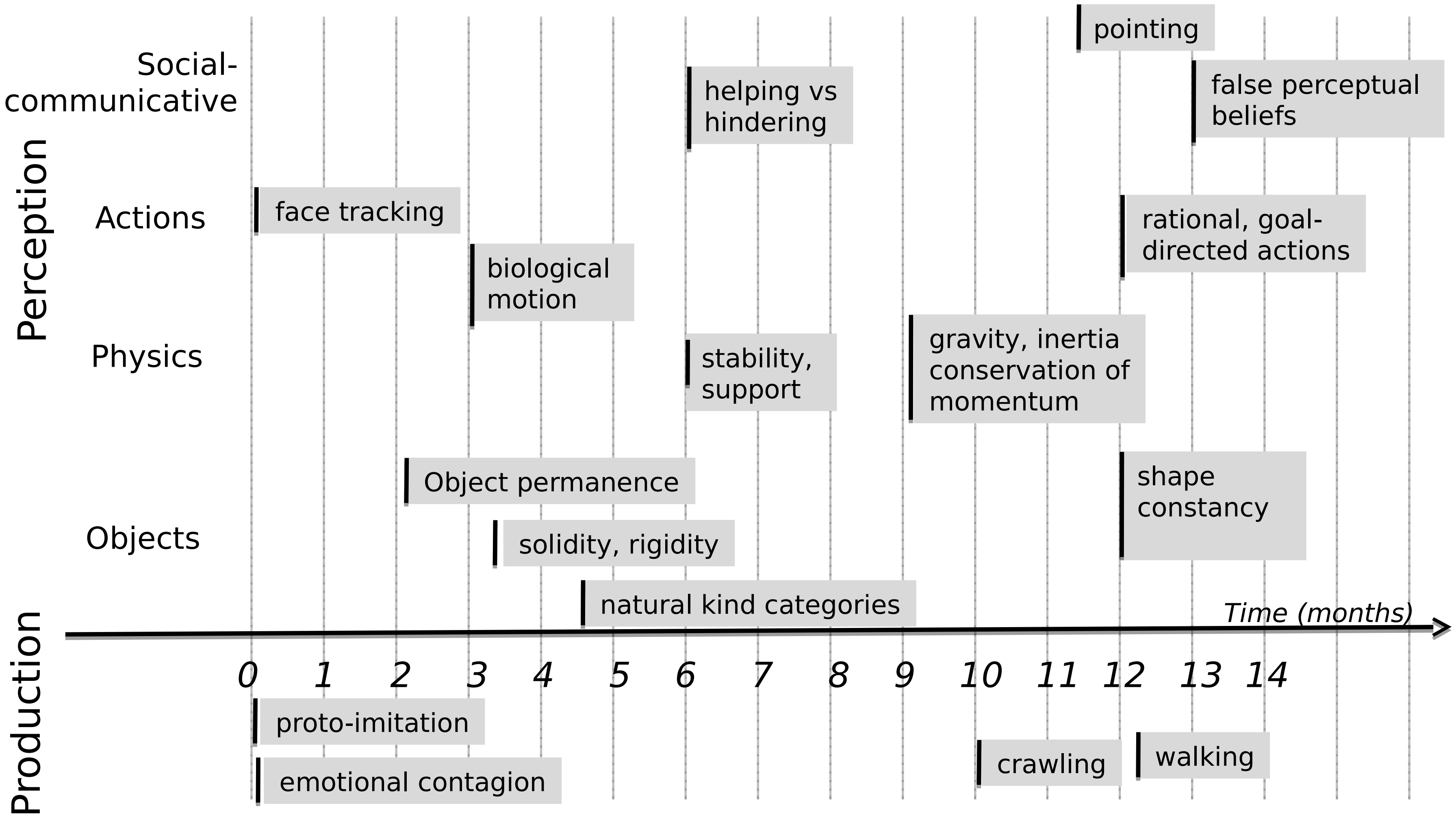
Not supervised.
Not Reinforced.

Babies learn how the world works by observation

- Largely by observation, with remarkably little interaction.

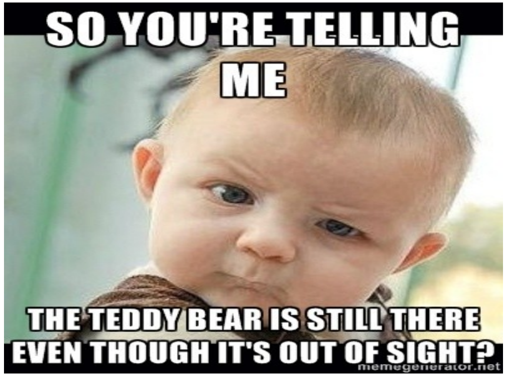
Photos courtesy of Emmanuel Dupoux

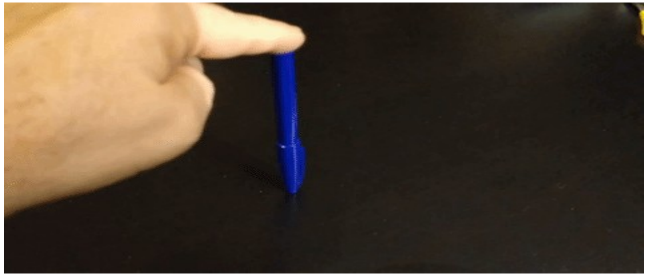
Early Conceptual Acquisition in Infants [from Emmanuel Dupoux]

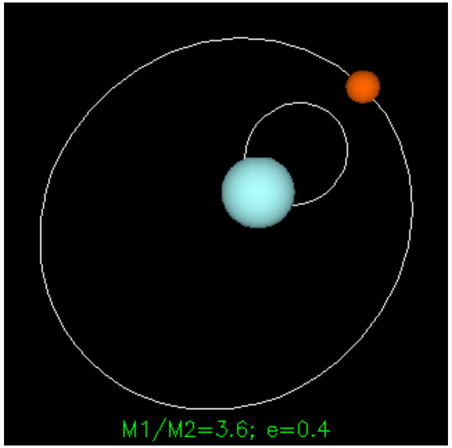


Prediction is the essence of Intelligence

- We learn models of the world by predicting







FACEBOOK AI

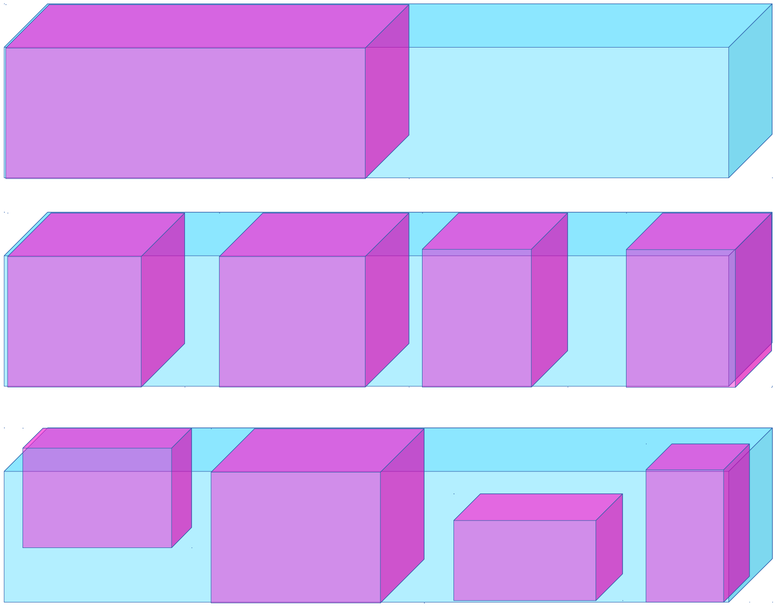
Self-Supervised Learning

Predict everything
from everything else

Self-Supervised Learning = Filling in the Blanks

- **Predict any part of the input from any other part.**
- **Predict the future from the past.**
- **Predict the masked from the visible.**
- **Predict the any occluded part from all available parts.**

time or space →



- **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**

Self-Supervised Learning: filling in the bl_nks

- **Natural Language Processing: works great!**

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

Decoder

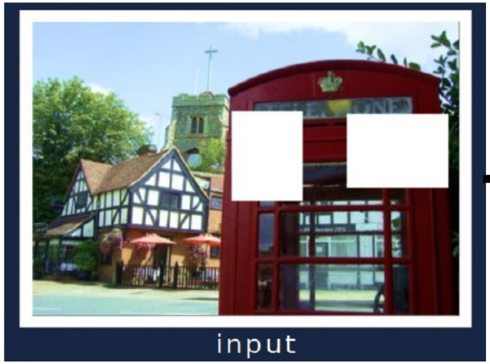
CODE

Encoder

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

- **Image Recognition / Understanding: works so-so**

[Pathak et al 2014]



Encoder → CODE → Decoder →

Learning Representations through Pretext SSL Tasks

- **Text / symbol sequences (discrete, works great!)**
  - Future word(s) prediction (NLM)
  - Masked words prediction (BERT et al.)
- **Image (continuous)**
  - Inpainting, colorization, super-resolution
- **Video (continuous)**
  - Future frame(s) prediction
  - Masked frames prediction
- **Signal / Audio (continuous)**
  - Restoration
  - Future prediction

Self-Supervised Learning works **very** well for text

- **Word2vec**
  - [Mikolov 2013]
- **FastText**
  - [Joulin 2016] (FAIR)
- **BERT**
  - Bidirectional Encoder Representations from Transformers
  - [Devlin 2018]
- **Cloze-Driven Auto-Encoder**
  - [Baevski 2019] (FAIR)
- **RoBERTa** [Ott 2019] (FAIR)

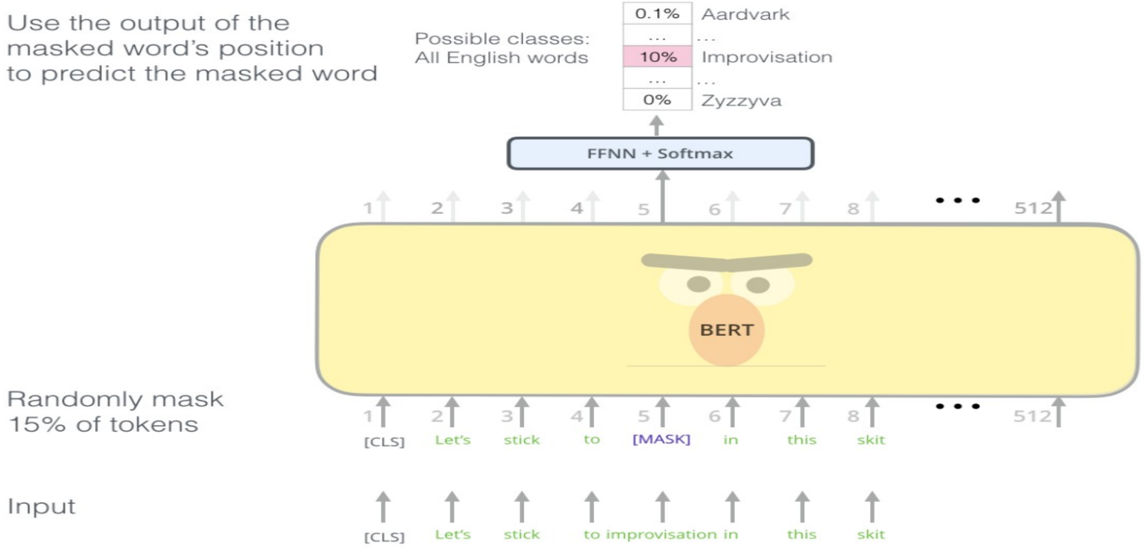
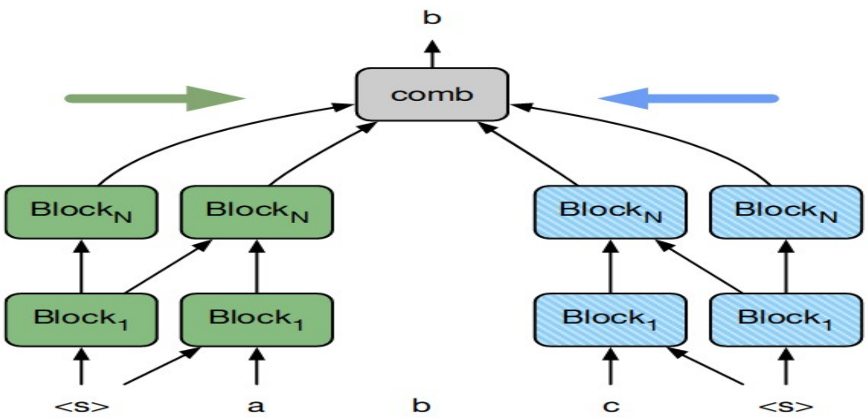


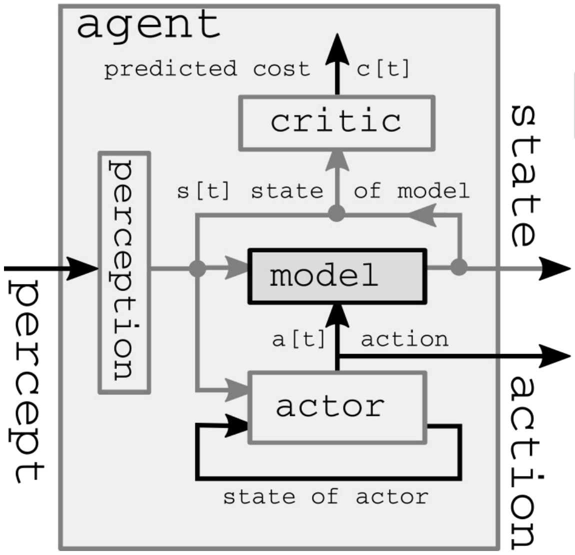
Figure credit: Jay Alammar http://jalammar.github.io/illustrated-bert/

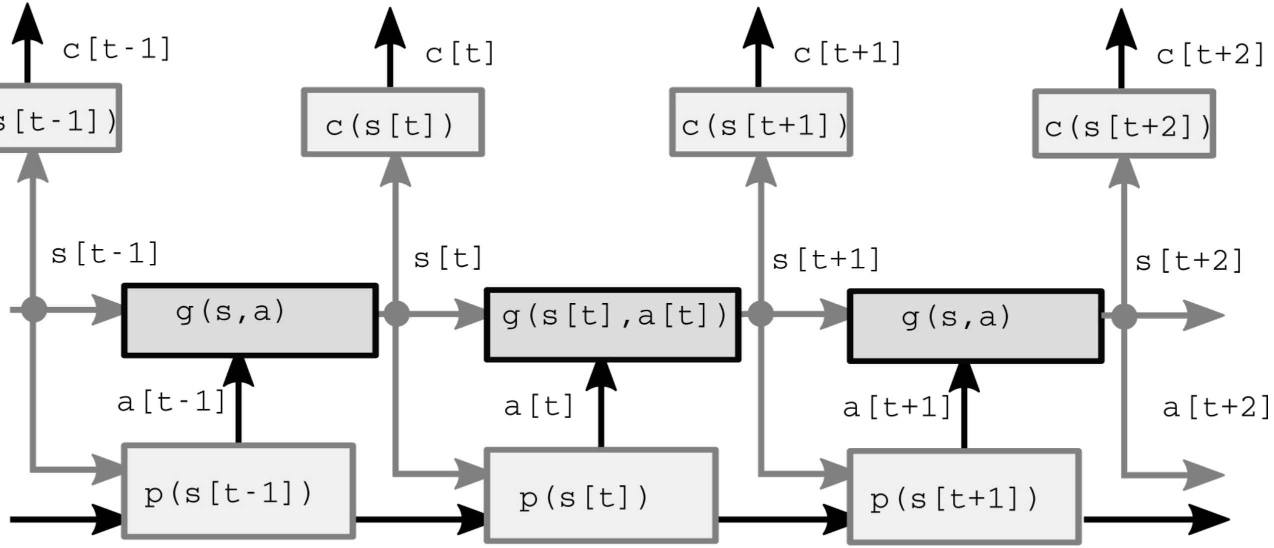


SSL works less well for images and video

Learning World Models for Autonomous AI Agents

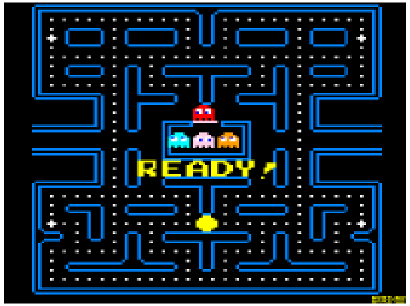
- **Learning forward models for control**
  - s[t+1] = g(s[t], a[t], z[t])
  - Model-predictive control, model-predictive policy learning, model-based RL
  - Robotics, games, dialog, HCI, etc





Three Types of Learning

- **Reinforcement Learning**
  - The machine predicts a scalar reward given once in a while.
  - **weak feedback**



- **Supervised Learning**
  - The machine predicts a category or a few numbers for each input
  - **medium feedback**

- **Self-supervised Learning**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **A lot of feedback**

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**

The Next AI Revolution

With thanks to Alyosha Efros and Gil Scott Heron

Get the T-shirt!

Jitendra Malik: "Labels are the opium of the machine learning researcher"

FACEBOOK AI

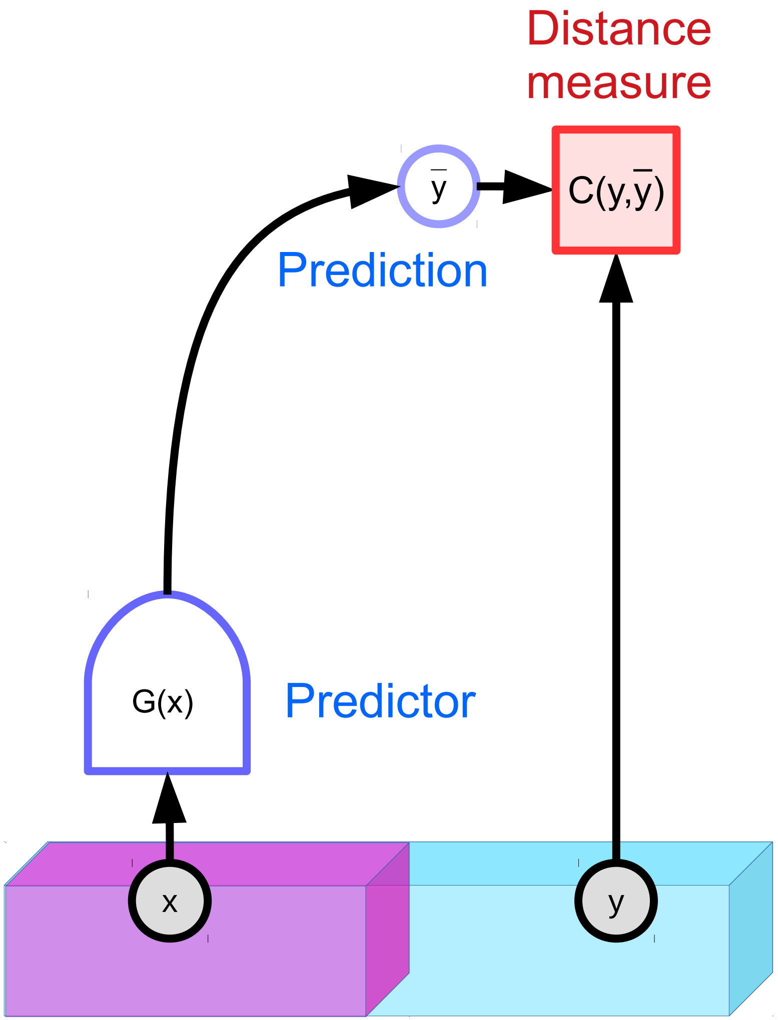
Energy-Based Models

Learning to deal with uncertainty while eschewing probabilities

Problem: uncertainty!

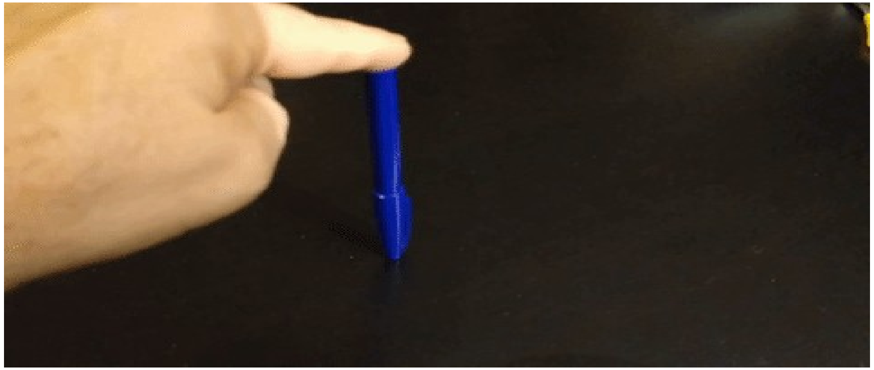
- There are **many** plausible words that complete a text.
- There are **infinitely many** plausible frames to complete a video.
- **Deterministic predictors don't work!**
- **How to deal with uncertainty in the prediction?**

$$E(x,y)=C(y,G(x))$$



The world is not entirely predictable / stochastic

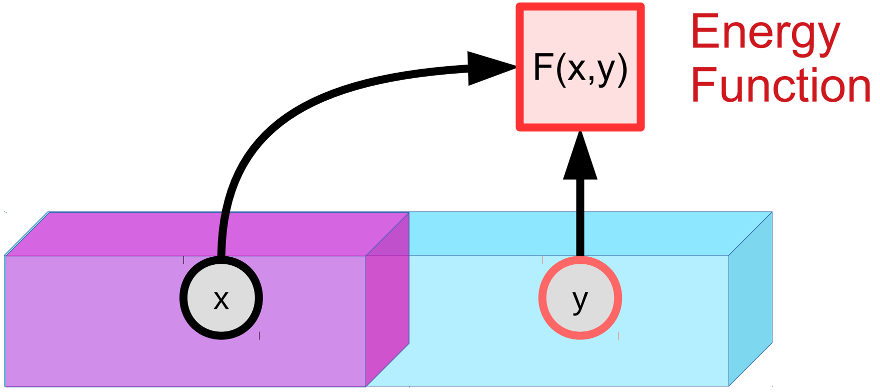
- **Video prediction:**
  - A deterministic predictor with L2 distance will predict the average of all plausible futures.
- **Blurry prediction!**



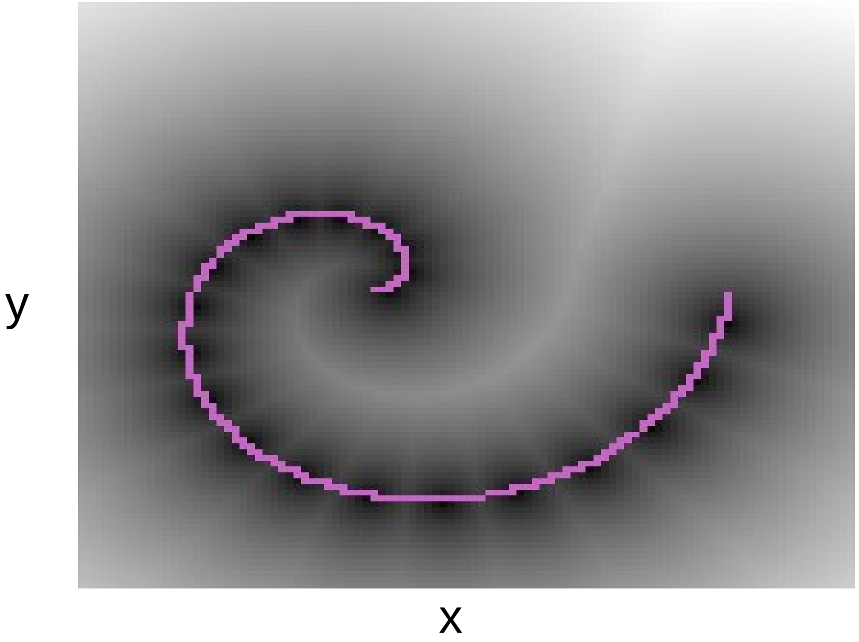
Energy-Based Model

- **Scalar-valued energy function: F(x,y)**
  - measures the compatibility between x and y
  - Low energy: y is good prediction from x
  - High energy: y is bad prediction from x

- Inference: $\check{y} = argmin_y F(x, y)$



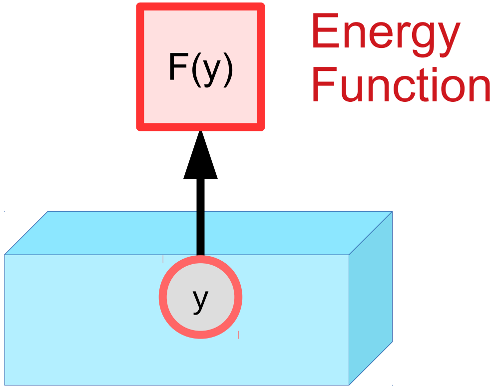
Dark = low energy (good)
Bright = high energy (bad)
Purple = data manifold



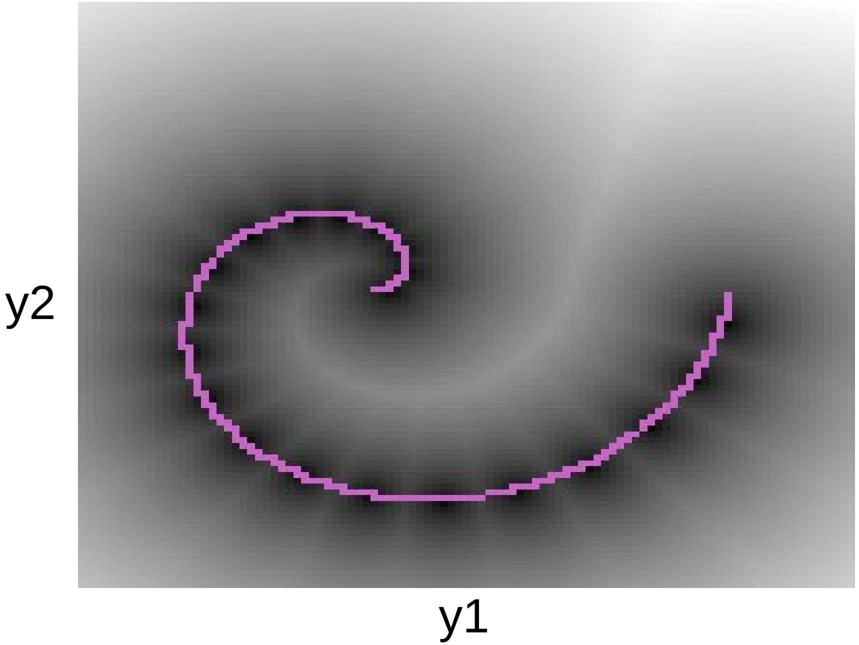
[Figure from M-A Ranzato's PhD thesis]

Energy-Based Model: unconditional version

- **Scalar-valued energy function: 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
  - Example: auto-encoders, generative models (energy = -log likelihood)

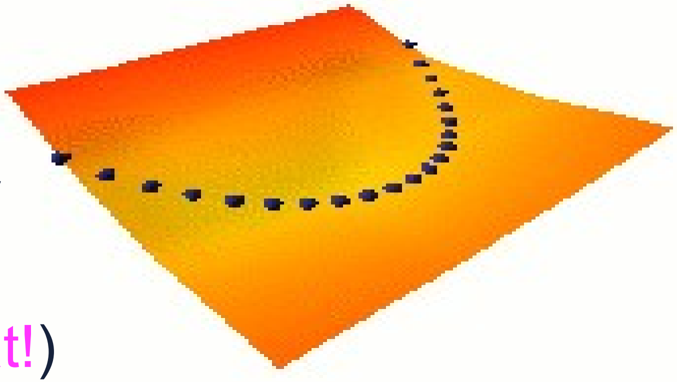


Dark = low energy (good)
Bright = high energy (bad)
Purple = data manifold

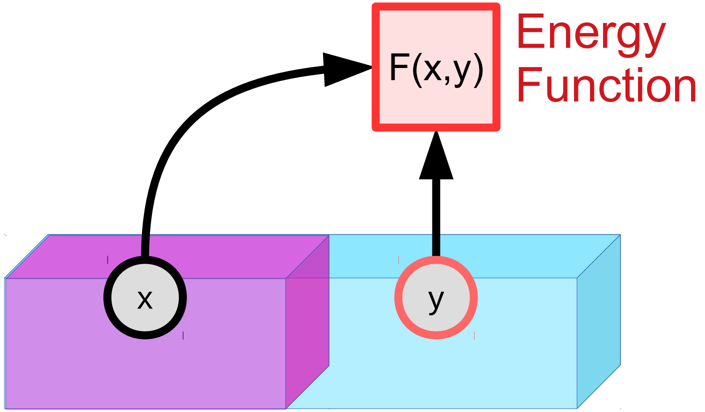


Training an Energy-Based Model

- **Parameterize F(x,y)**
- **Get training data (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]
  - F is smooth (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. **Architectural Methods**: build F(x,y) so that the volume of low energy regions is limited or minimized through regularization



Seven Strategies to Shape the Energy Function

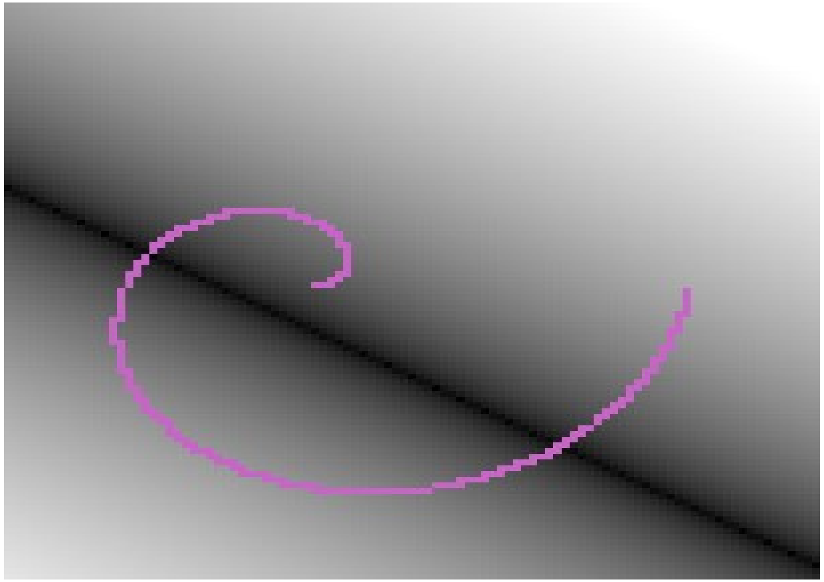
- **Contrastive: [they all are different ways to pick which points to push up]**
  - C1: push down of the energy of data points, push up everywhere else: Max likelihood (needs tractable partition function or variational approximation)
  - C2: push down of the energy of data points, push up on chosen locations: max likelihood with MC/MMC/HMC, Contrastive divergence, Metric learning, 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)
- **Architectural: [they all are different ways to limit the information capacity of the code]**
  - A1: build the machine so that the volume of low energy stuff is bounded: PCA, K-means, Gaussian Mixture Model, Square ICA…
  - A2: use a regularization term that measures the volume of space that has low energy: Sparse coding, sparse auto-encoder, LISTA, Variational auto-encoders
  - 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

Simple examples: PCA and K-means

- **Limit the capacity of z so that the volume of low energy stuff is bounded**
  - PCA, K-means, GMM, square ICA...

PCA: z is low dimensional

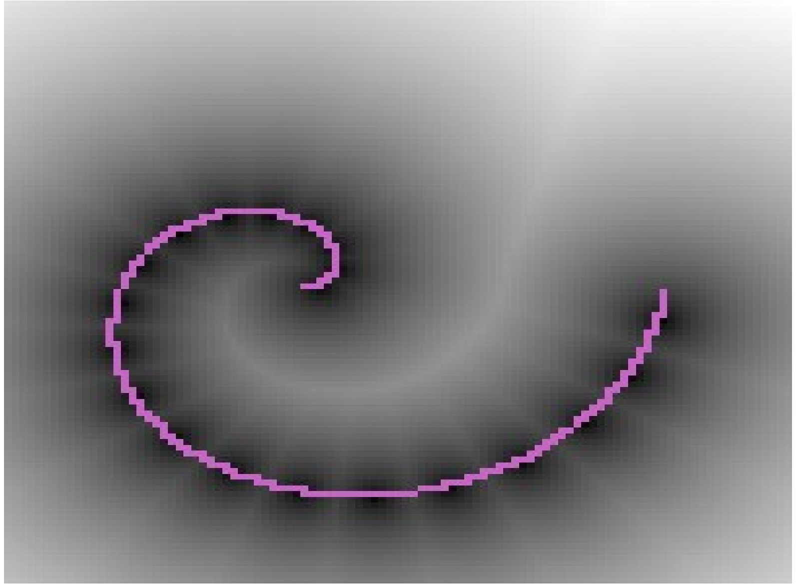
$$F(Y) = ||W^T WY - Y||^2$$



K-Means,

Z constrained to 1-of-K code

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

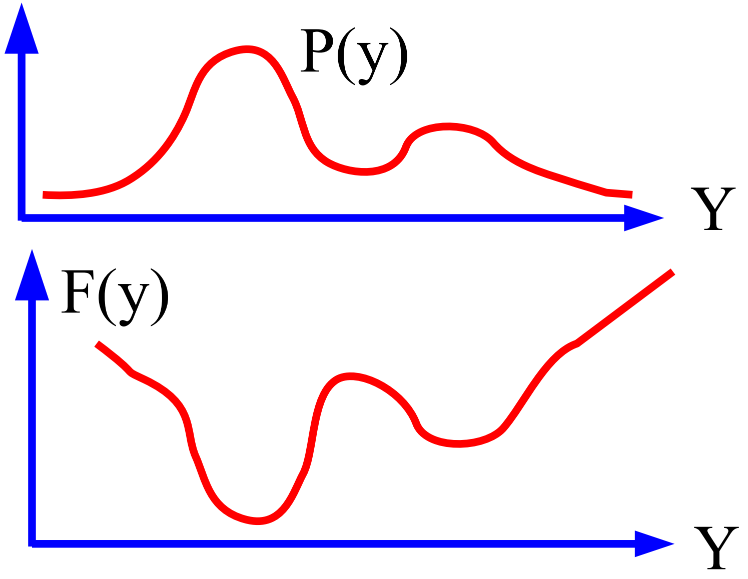


Familiar Example: Maximum Likelihood Learning

- **The energy can be interpreted as an unnormalized negative log density**
- **Gibbs distribution: Probability proportional to exp(-energy)**
  - Beta parameter is akin to an inverse temperature
- **Don't compute probabilities unless you absolutely have to**
  - Because the denominator is often intractable

$$P(y) = -\frac{\exp[-\beta F(y)]}{\int_{y'} \exp[-\beta F(y')]}$$

$$P(y|x) = -\frac{\exp[-\beta F(x,y)]}{\int_{y'} \exp[-\beta F(x,y')]}$$



push down of the energy of data points, push up everywhere else

- **Max likelihood (requires a tractable partition function)**

Maximizing P(Y|W) on training samples

make this big

$$P(Y|W) = \frac{e^{-\beta E(Y,W)}}{\int_y e^{-\beta E(y,W)}}$$

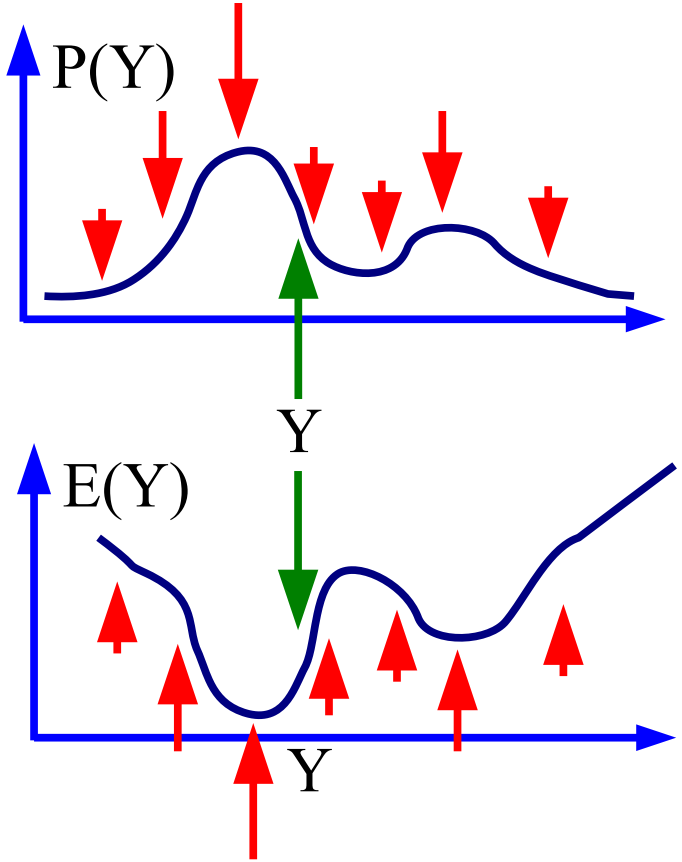
make this small

Minimizing -log P(Y,W) on training samples

$$L(Y,W) = E(Y,W) + \frac{1}{\beta} \log \int_y e^{-\beta E(y,W)}$$

make this small

make this big



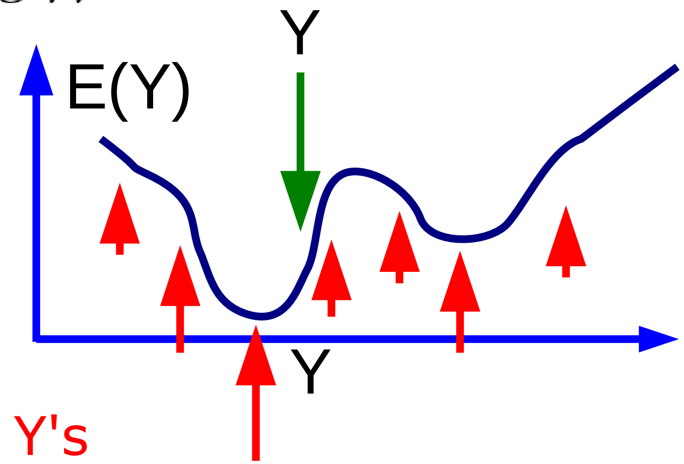
push down of the energy of data points, push up everywhere else

Gradient of the negative log-likelihood loss for one sample Y:

$$\frac{\partial L(Y,W)}{\partial W} = \frac{\partial E(Y,W)}{\partial W} - \int_y P(y|W)\frac{\partial E(y,W)}{\partial W}$$

Gradient descent:

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



Pushes down on the energy of the samples

Pulls up on the energy of low-energy Y's

$$W \leftarrow W - \eta \frac{\partial E(Y,W)}{\partial W} + \eta \int_y P(y|W)\frac{\partial E(y,W)}{\partial W}$$

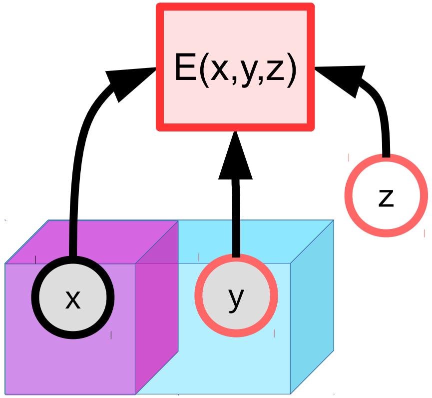
Latent-Variable EBM

- **Allowing multiple predictions through a latent variable**

- **Conditional:**

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

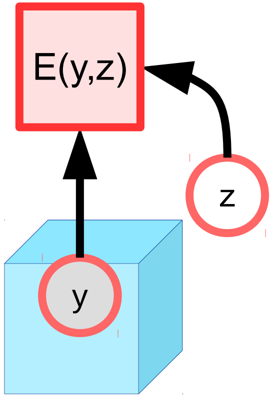
$$F(x,y)=-\frac{1}{\beta}\log\left[\int_z \exp(-\beta E(x,y,z))\right]$$



- **Unconditional**

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

$$F(y)=-\frac{1}{\beta}\log\left[\int_z \exp(-\beta E(y,z))\right]$$



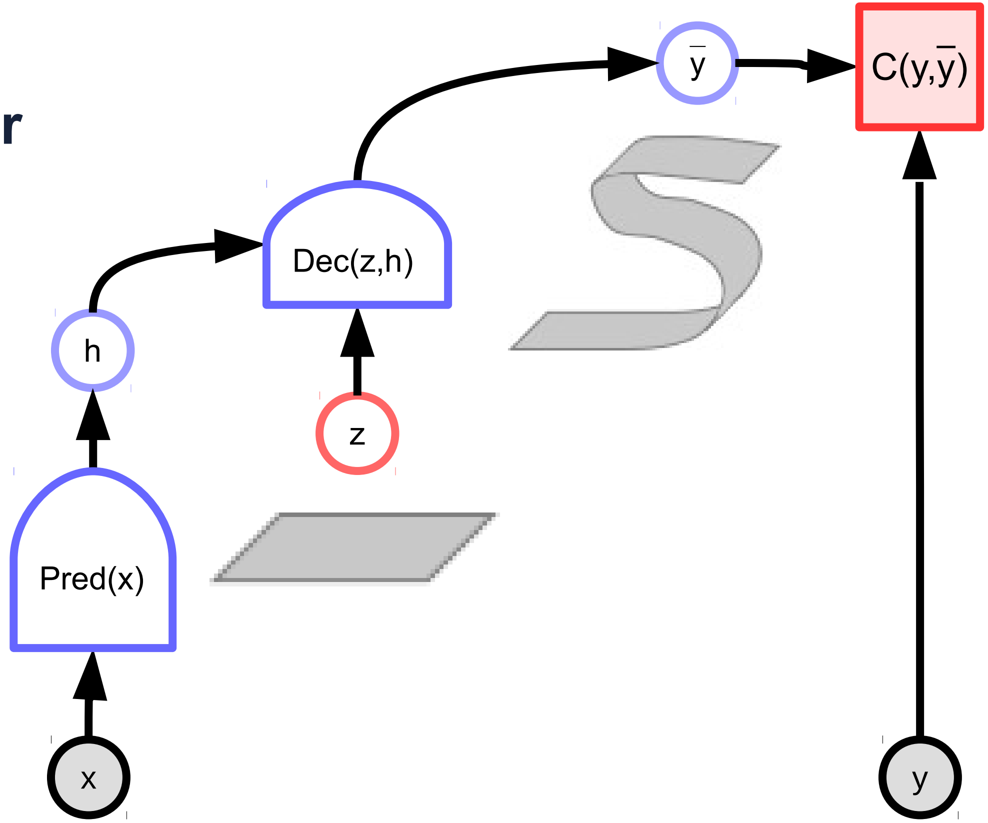
Latent-Variable EBM for multimodal prediction

- **Allowing multiple predictions through a latent variable**
- **As z varies over a set, y varies over the manifold of possible predictions**

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

- **Examples:**
  - K-means
  - Sparse modeling
  - GLO

  [Bojanowski arXiv:1707.05776]

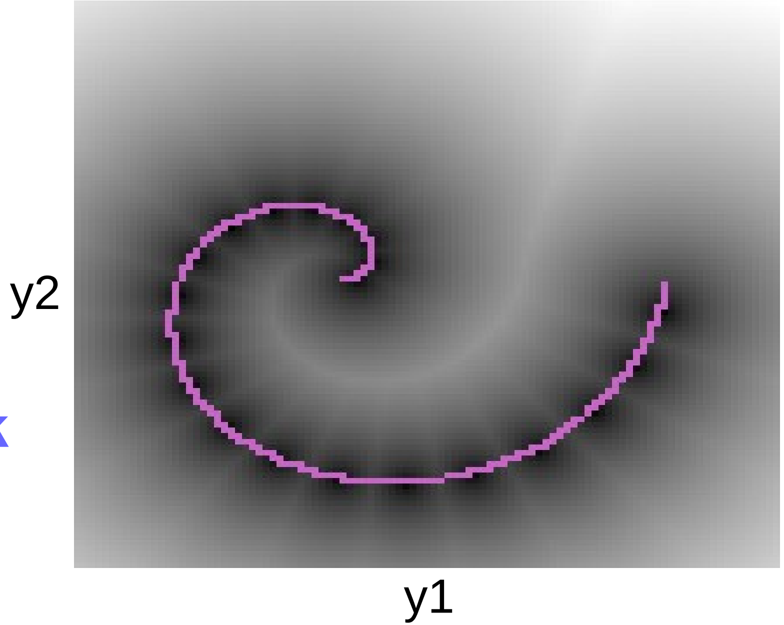


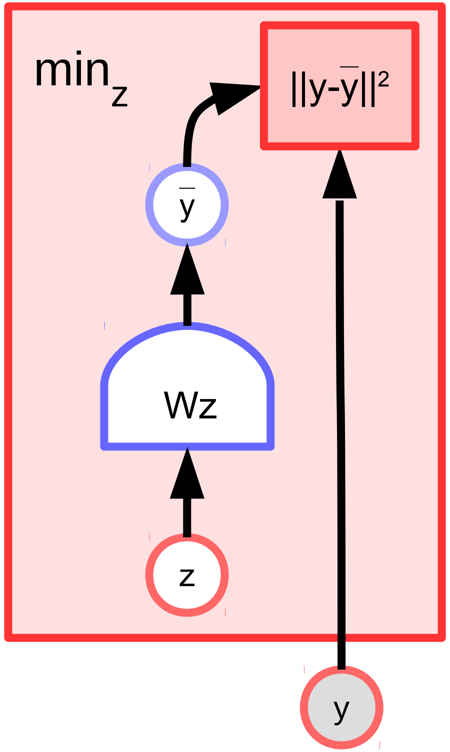
Latent-Variable EBM example: K-means

- **Decoder is linear, z is a 1-hot vector (discrete)**
- **Energy function:** $E(y,z) = ||y - Wz||^2 \quad z \in 1\,hot$
- **Inference by exhaustive search**

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

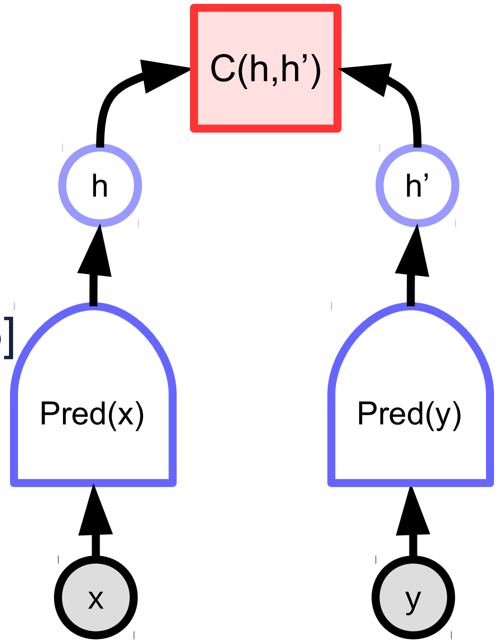
- **Volume of low-energy regions limited by number of prototypes k**





Contrastive Embedding

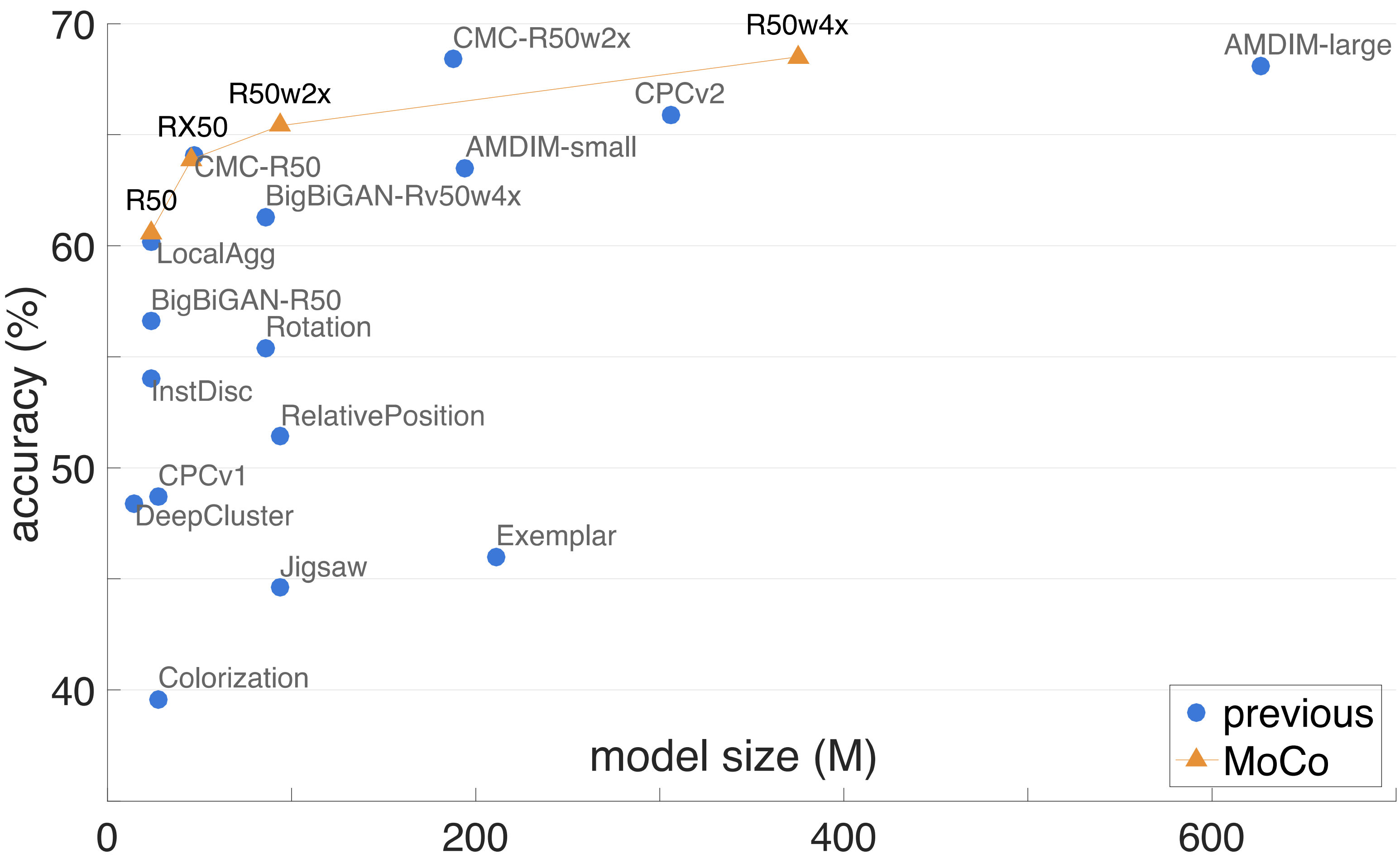
- **Distance measured in feature space**
- **Multiple “predictions” through feature invariance**
- **Siamese nets, metric learning** [YLC NIPS’93,CVPR’05,CVPR’06]
- **Advantage: no pixel-level reconstruction**
- **Difficulty: hard negative mining**
- **Successful examples for images:**
  - DeepFace [Taigman et al. CVPR’14]
  - PIRL [Misra et al. To appear]
  - MoCo [He et al. Arxiv:1911.05722]
- **Video / Audio**
  - Temporal proximity [Taylor CVPR’11]
  - Slow feature [Goroshin NIPS’15]



Positive pair:
Make F small

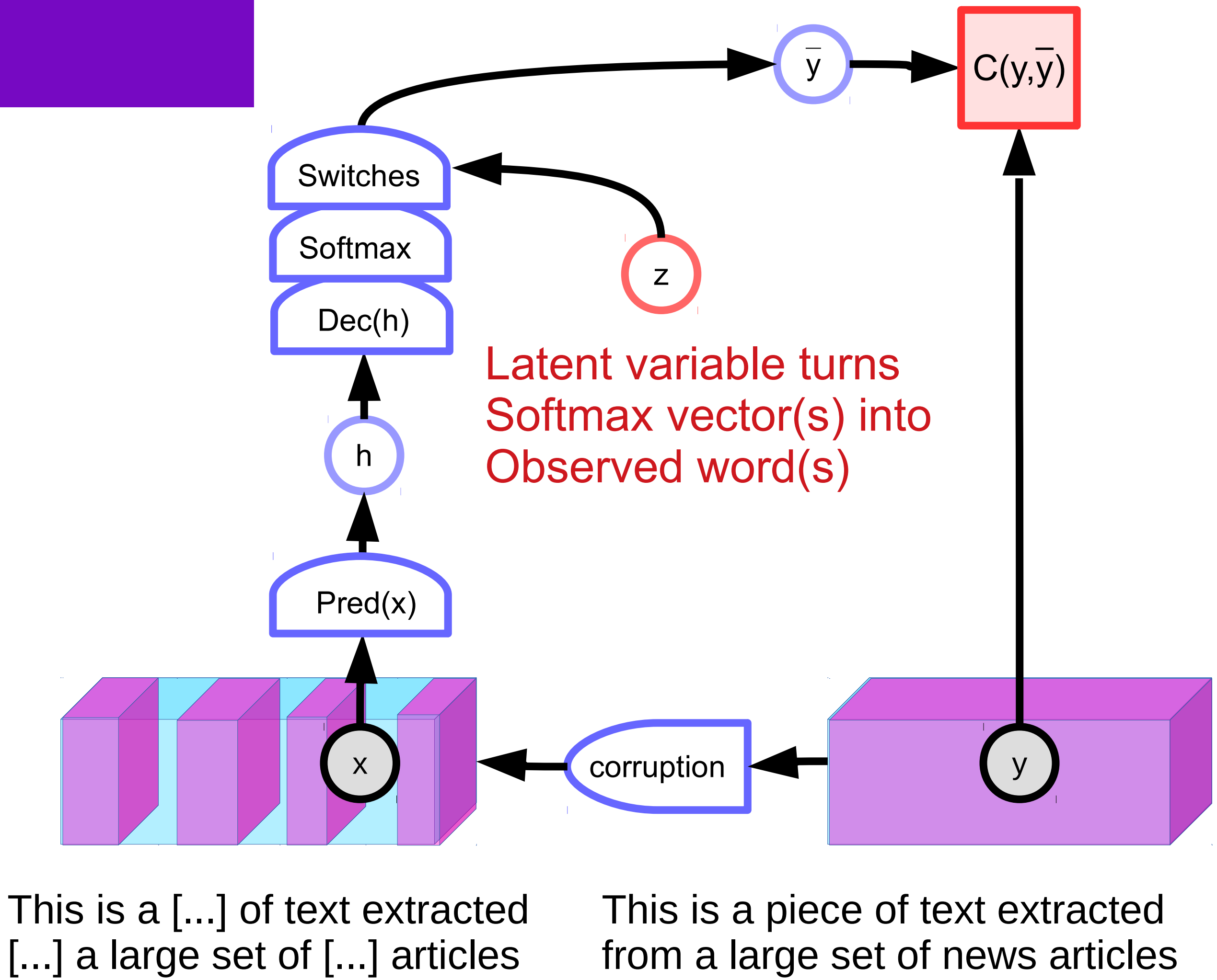

Negative pair:
Make F large


MoCo on ImageNet [He et al. Arxiv:1911.05722]



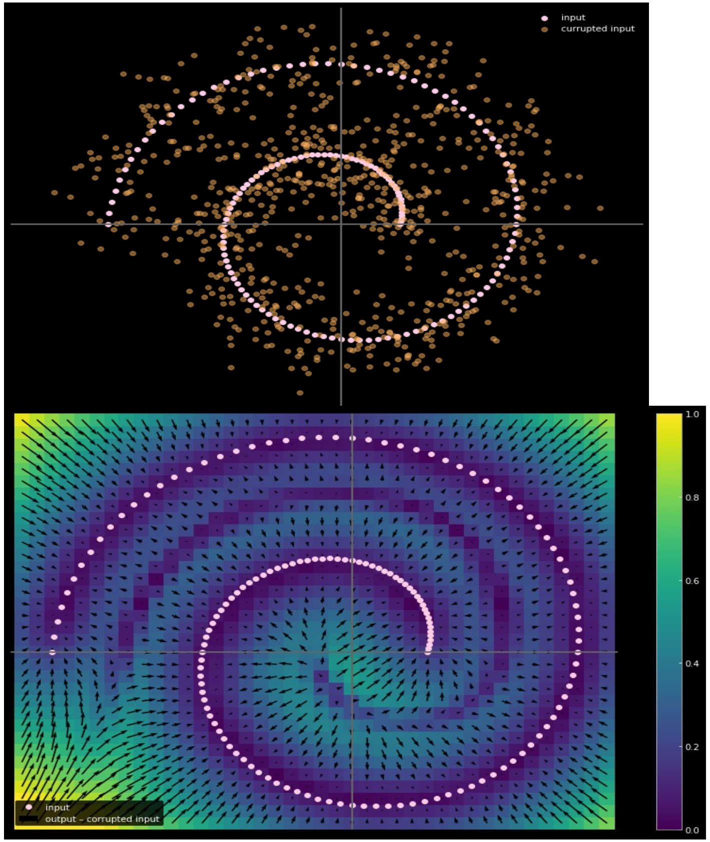
Denoising AE: discrete

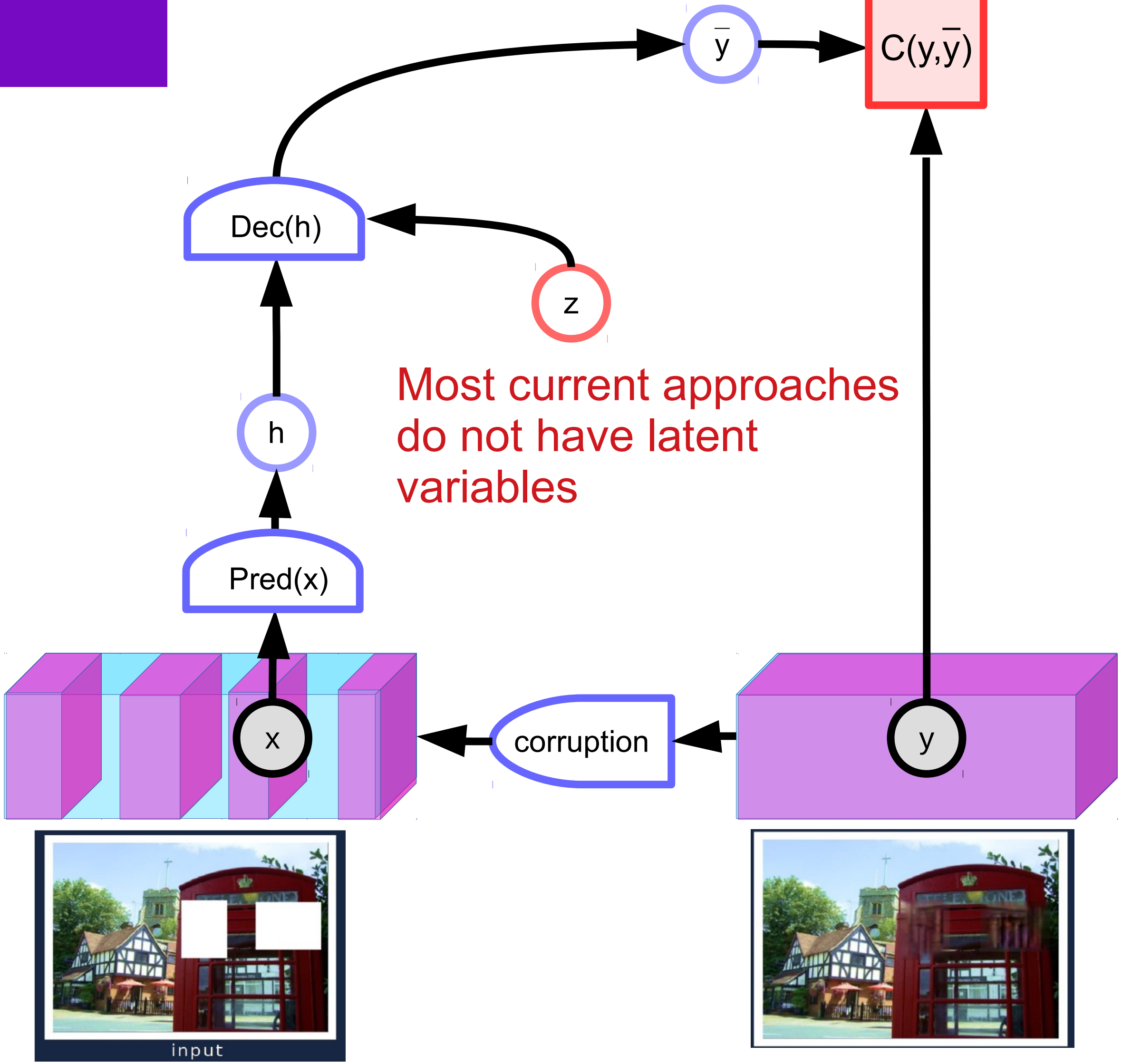
- **[Vincent et al. JMLR 2008]**
- **Masked Auto-Encoder**
  - [BERT et al.]
- **Issues:**
  - latent variables are in output space
  - No abstract LV to control the output
  - How to cover the space of corruptions?



Denoising AE: continuous

- **Image inpainting** [Pathak 17]
- Latent variables? GAN?





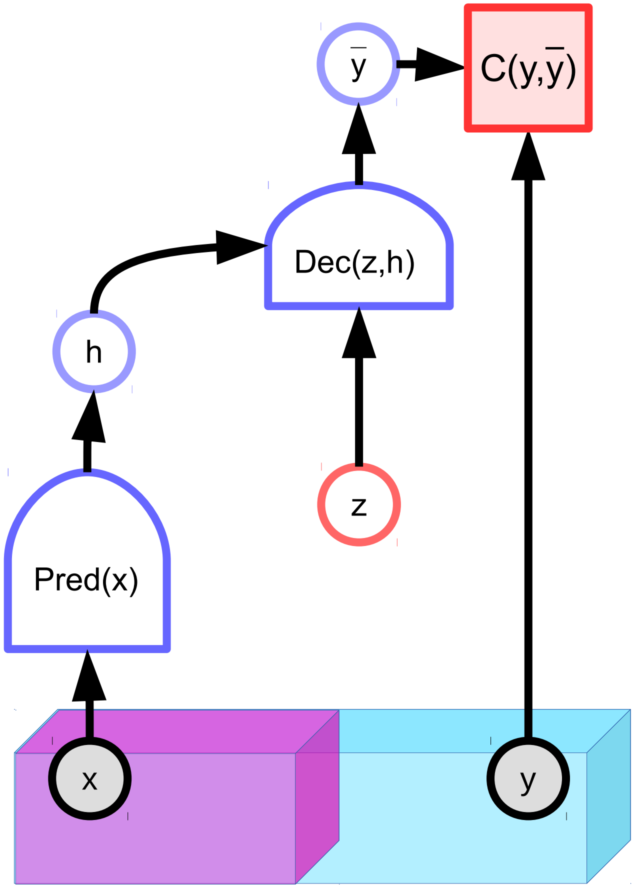
Most current approaches do not have latent variables

Prediction with Latent Variables

- **If the Latent has too much capacity...**
  - e.g. if it has the same dimension as y
- **… then the entire y space could be perfectly reconstructed**

$$E(x,y,z)=C(y,Dec(Pred(x),z))$$

- **For every y, there is always a z that will reconstruct it perfectly**
  - The energy function would be zero everywhere
  - This is no a good model….
- **Solution: limiting the information capacity of the latent variable z.**

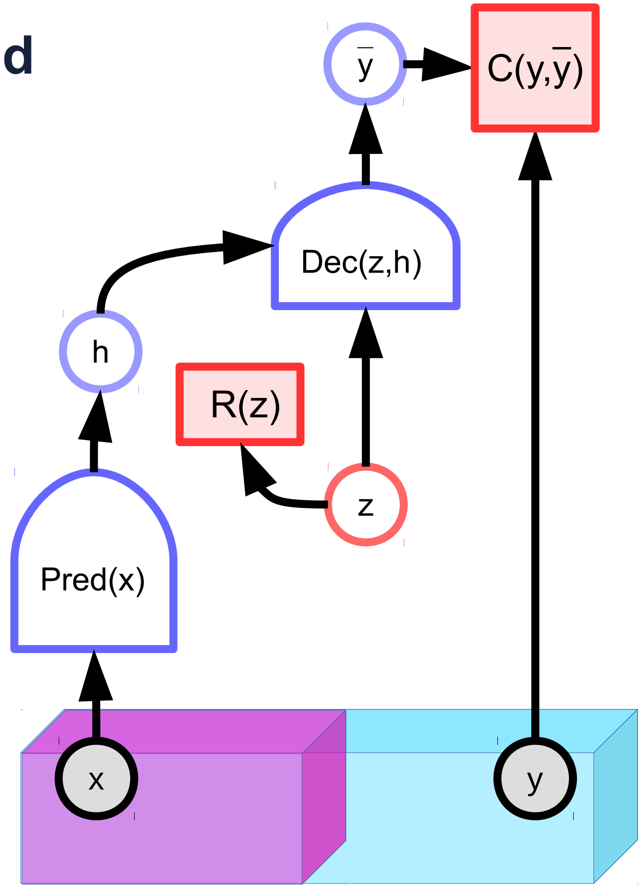


Regularized Latent Variable EBM

- **Regularizer R(z) limits the information capacity of z**
- **Without regularization, every y may be reconstructed exactly (flat energy surface)**

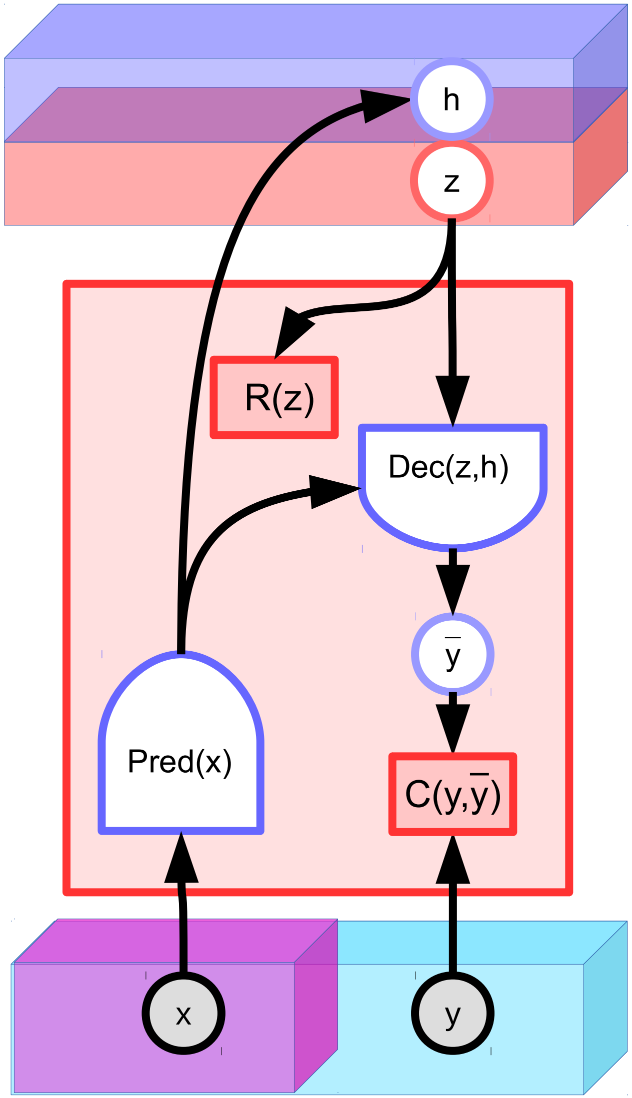
$$E(x,y,z)=C(y,Dec(Pred(x),z))+\lambda R(z)$$

- **Examples of R(z):**
  - Effective dimension
  - Quantization / discretization
  - L0 norm (# of non-0 components)
  - L1 norm with decoder normalization
  - Maximize lateral inhibition / competition
  - Add noise to z while limiting its L2 norm (VAE)
  - <your_information_throttling_method_goes_here>



Sequence → Abstract Features

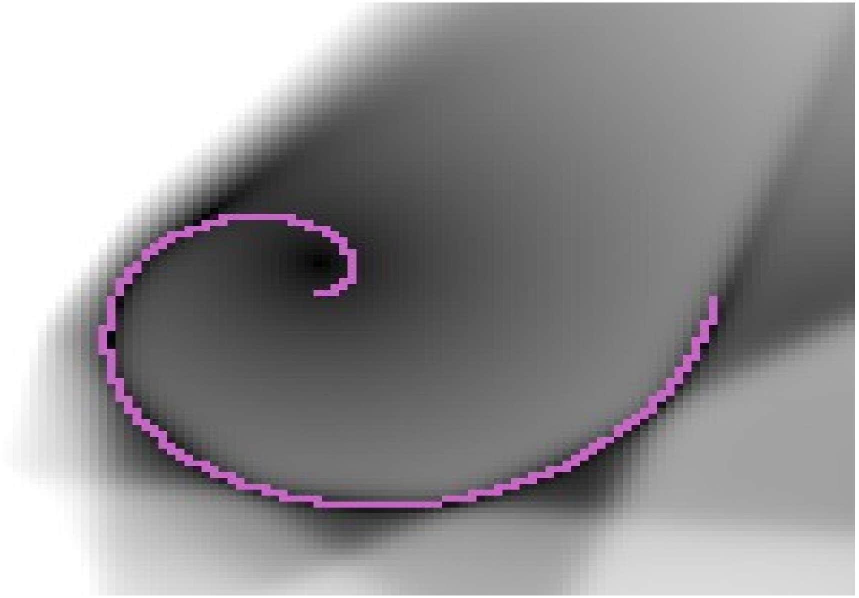
- **Regularized LV EBM is passed over a sequence (e.g. a video, audio, text)**
- **The sequence of corresponding h and z is collected**
  - It contains all the information about the input sequence
  - h contains the information in x that is useful to predict y
  - z contains the complementary information, not present in x or h.
- **Several such SSL modules can be stacked to learn hierarchical representations of sequences**

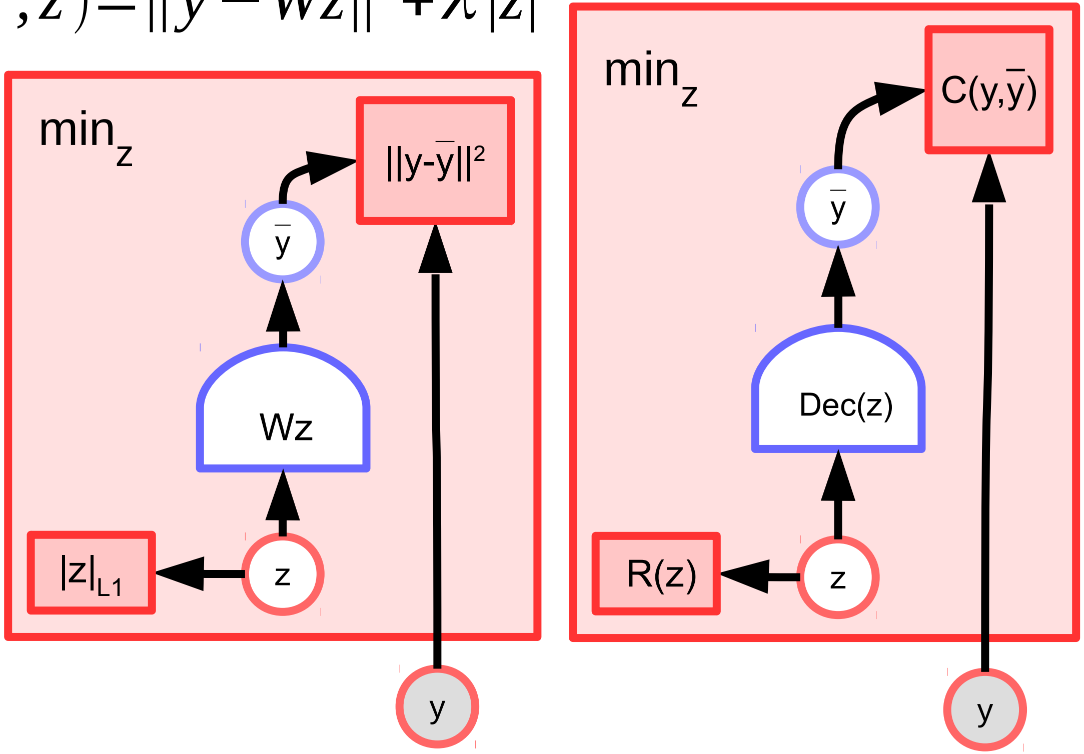


Unconditional Regularized Latent Variable EBM

- **Unconditional form. Reconstruction. No x, no predictor.**
- **Example: sparse modeling**
  - Linear decoder
  - L1 regularizer on Z

$$E(y,z) = \|y - Wz\|^2 + \lambda |z|$$



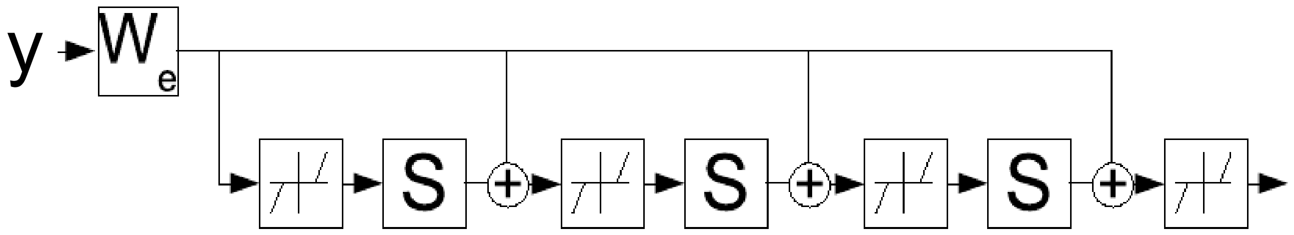


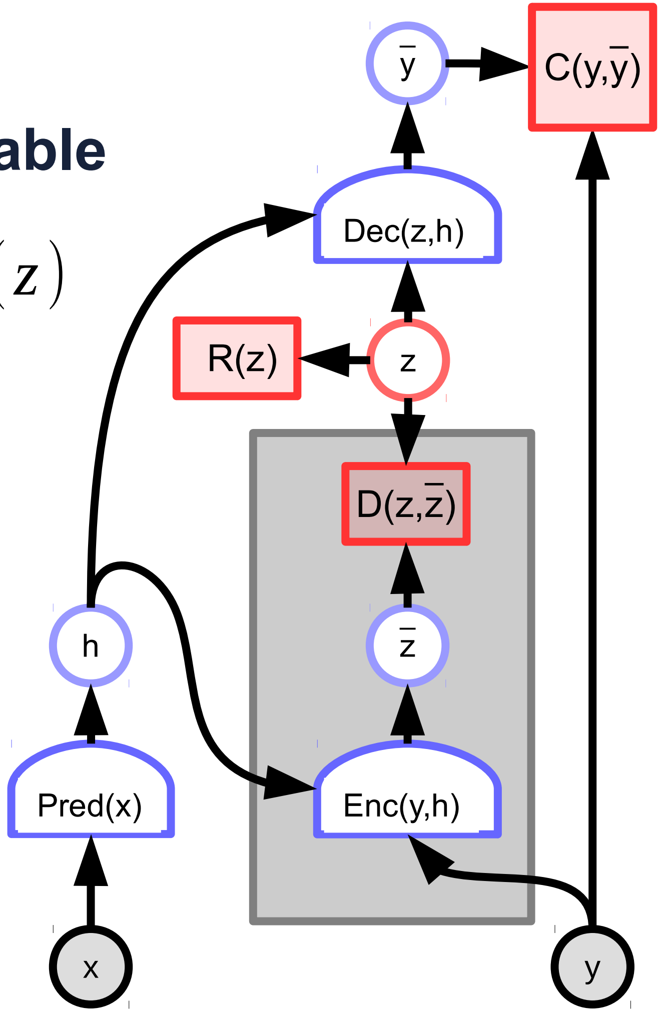
LatVar inference is expensive!

- **Let's train an encoder to predict the latent variable**

$$E(x,y,z) = C(y, Dec(z,h)) + D(z, Enc(x,y)) + \lambda R(z)$$

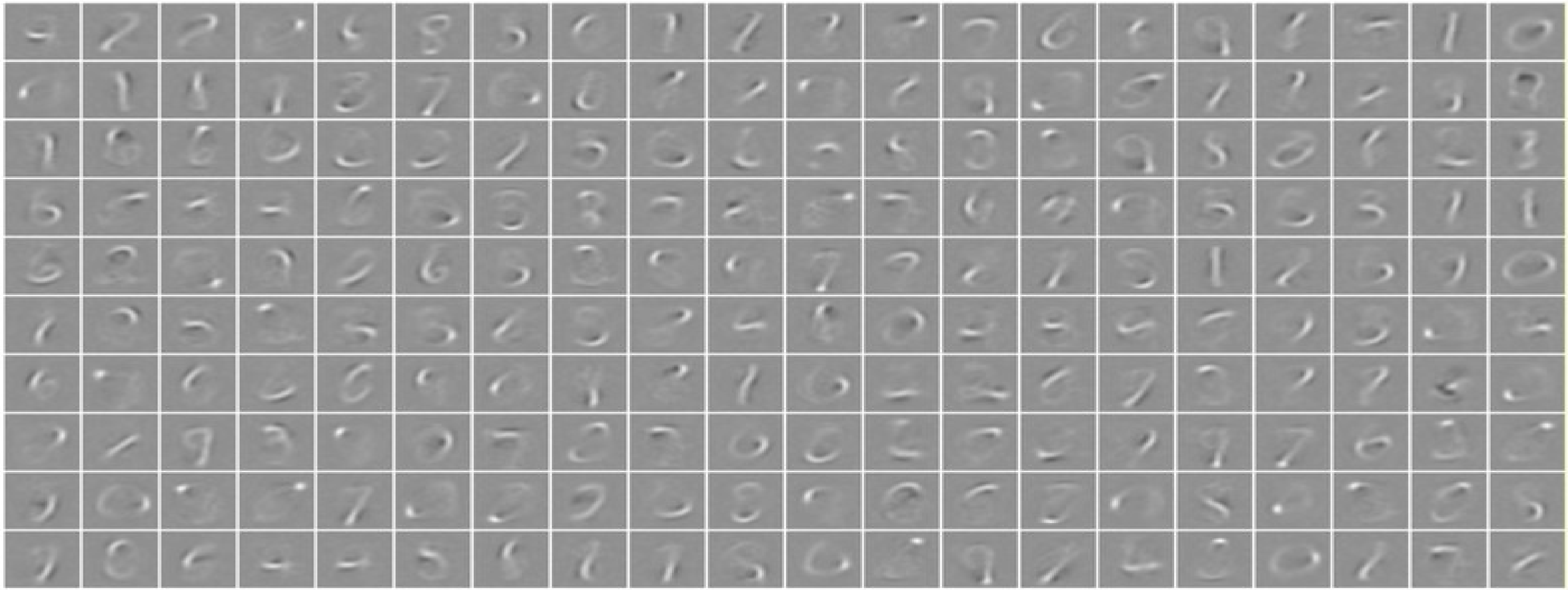
- **Predictive Sparse Modeling**
  - R(z) = L1 norm of z
  - Dec(z,h) gain must be bounded (clipped weights)
  - Sparse Auto-Encoder
  - LISTA [Gregor ICML 2010]





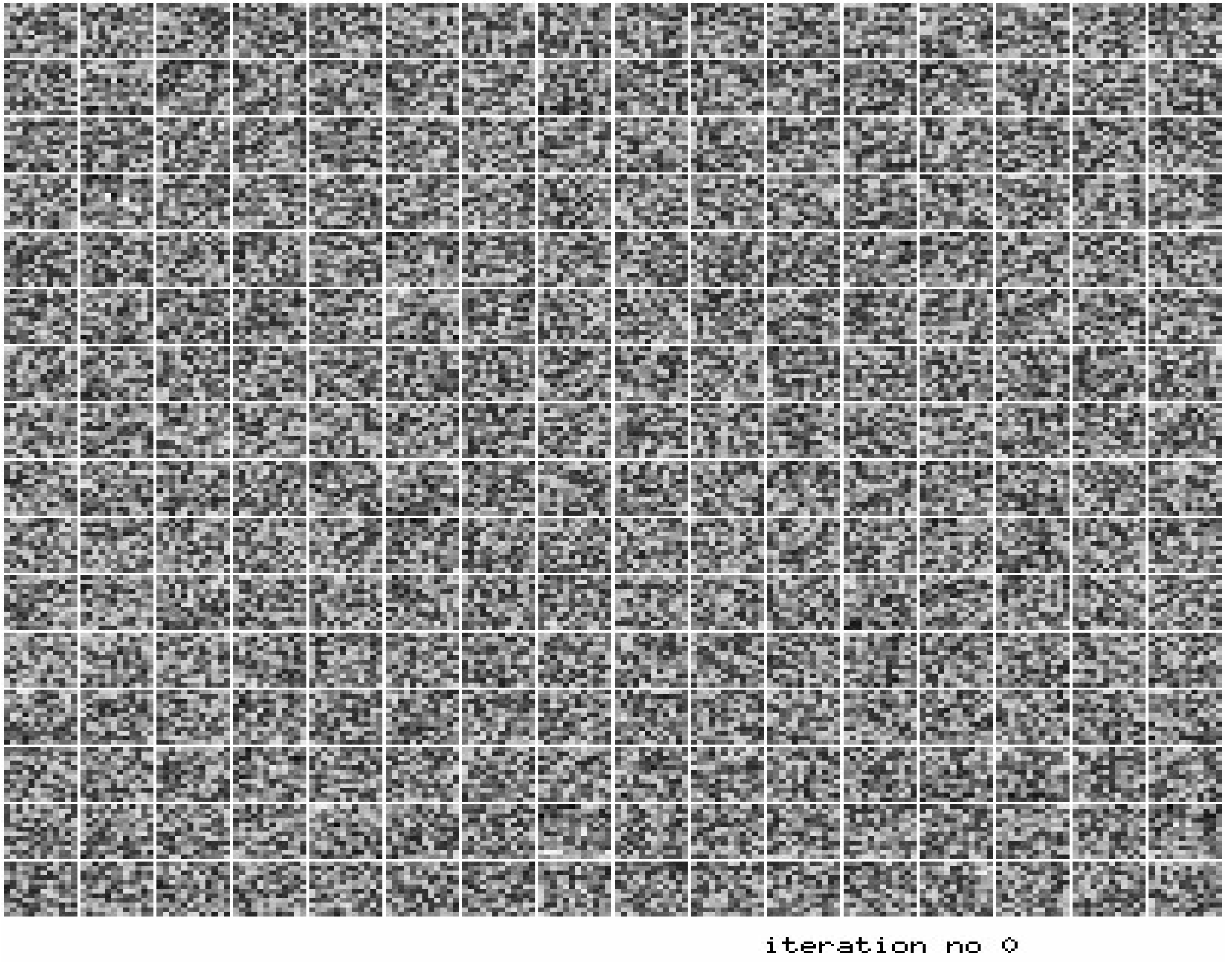
Sparse AE on handwritten digits (MNIST)

- **256 basis functionsBasis functions (columns of decoder matrix) are digit parts**
- **All digits are a linear combination of a small number of these**

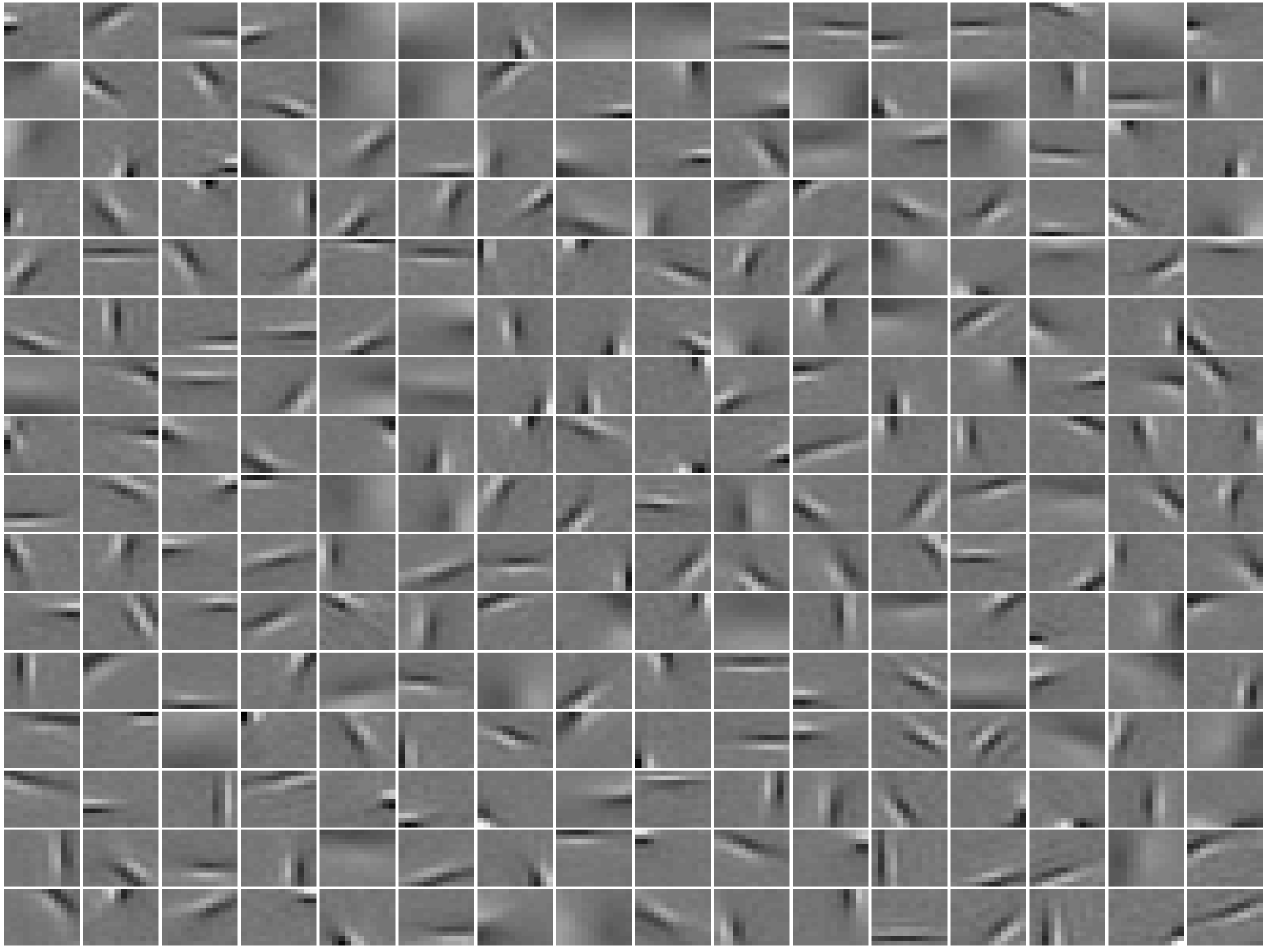


Predictive Sparse Decomposition (PSD): Training

- **Training on natural images patches.**
  - 12X12
  - 256 basis functions
  - [Ranzato 2007]

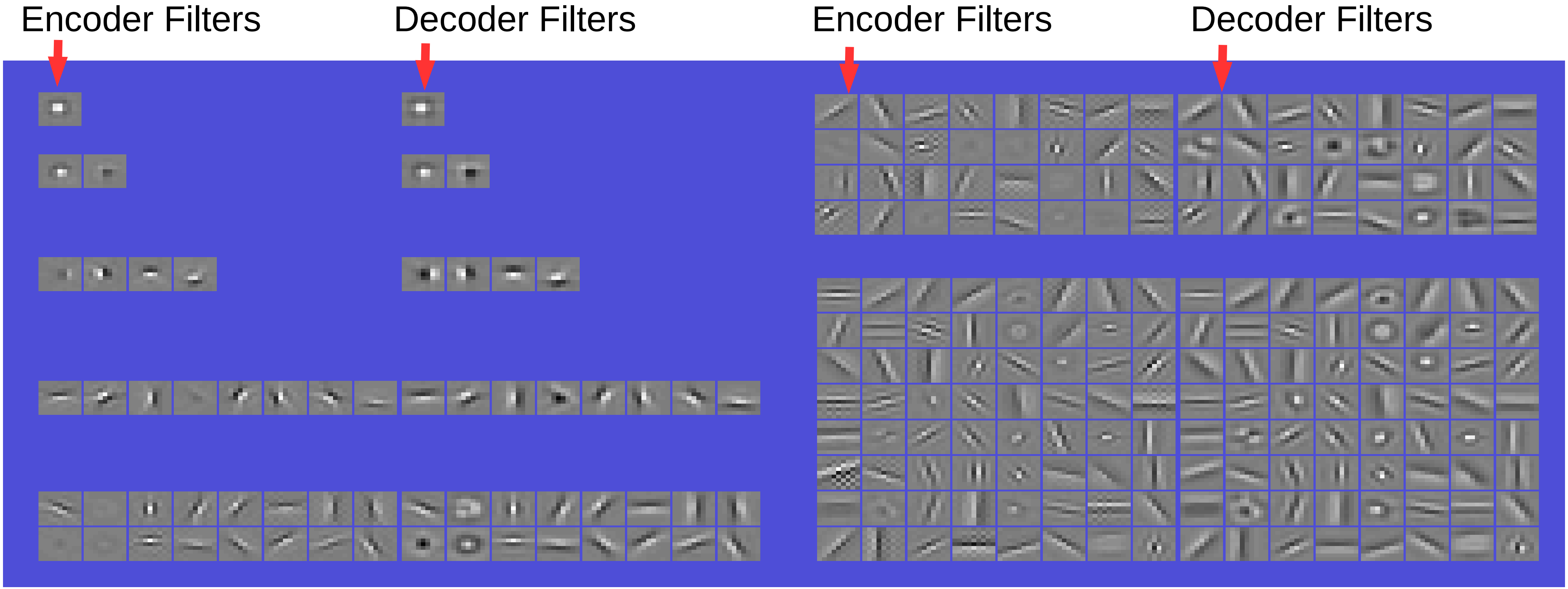


Learned Features: V1-like receptive fields



Convolutional Sparse Auto-Encoder on Natural Images

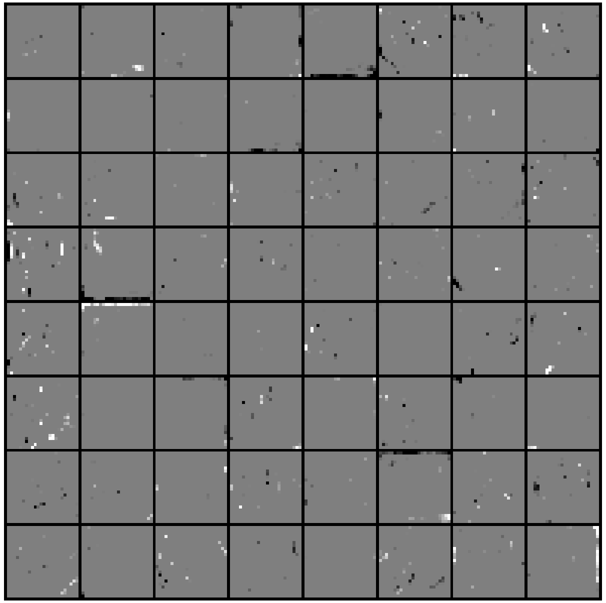
- **Filters and Basis Functions obtained. Linear decoder (conv)**
  - with 1, 2, 4, 8, 16, 32, and 64 filters [Kavukcuoglu NIPS 2010]



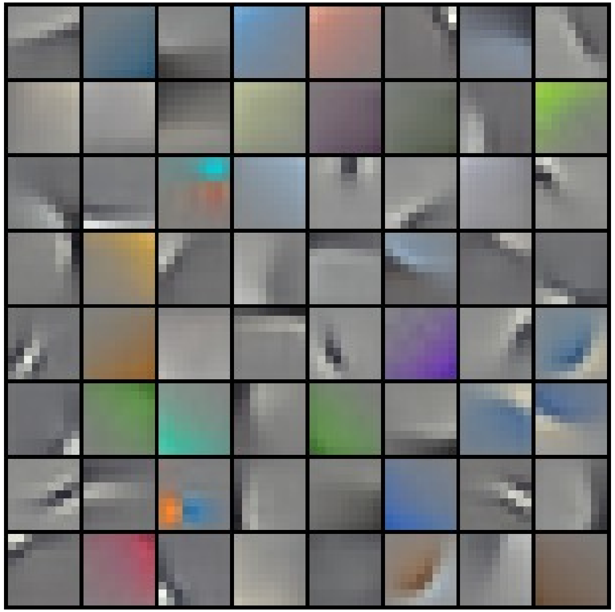
Convolutional Sparse Auto-Encoder on Natural Images

- Trained on CIFAR 10 (32x32 color images)
- Architecture: Linear decoder, LISTA recurrent encoder
- Pytorch implementation (talk to Jure Zbontar)

sparse codes (z) from encoder

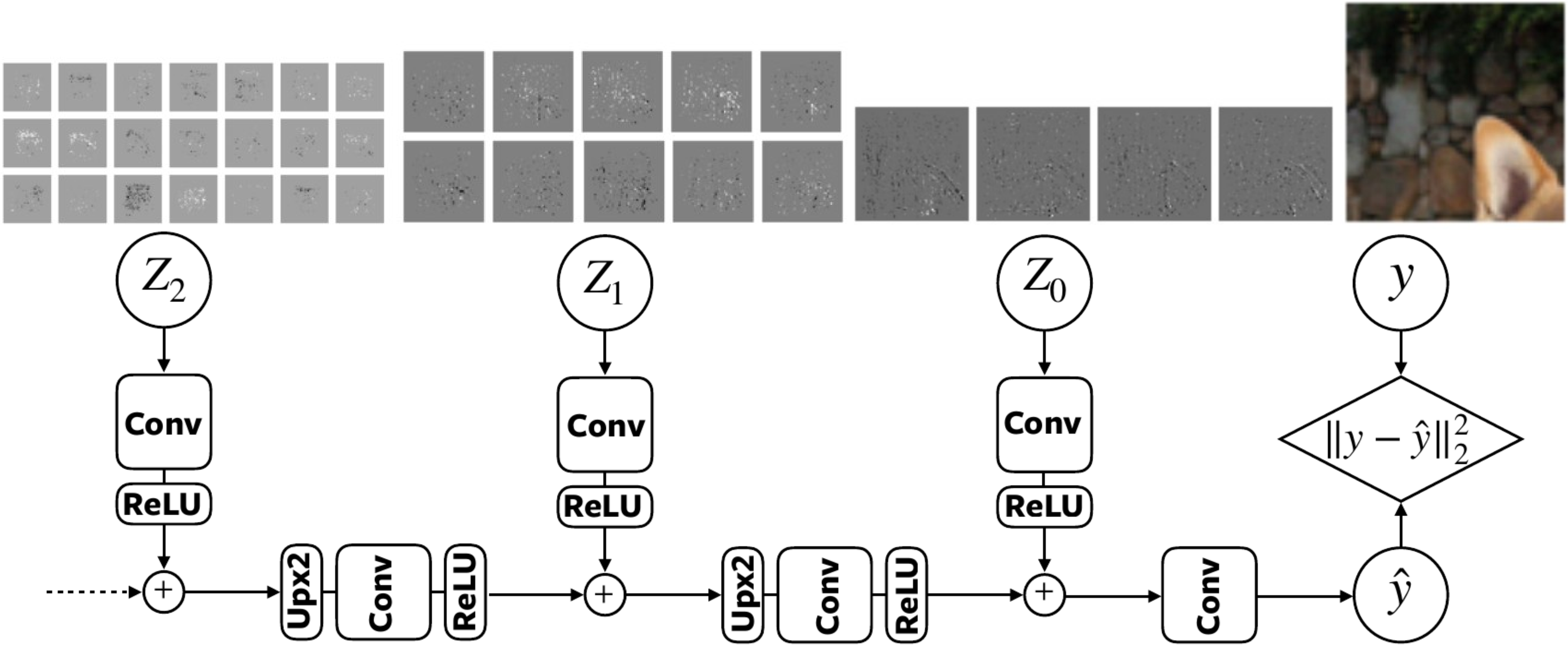


9x9 decoder kernels



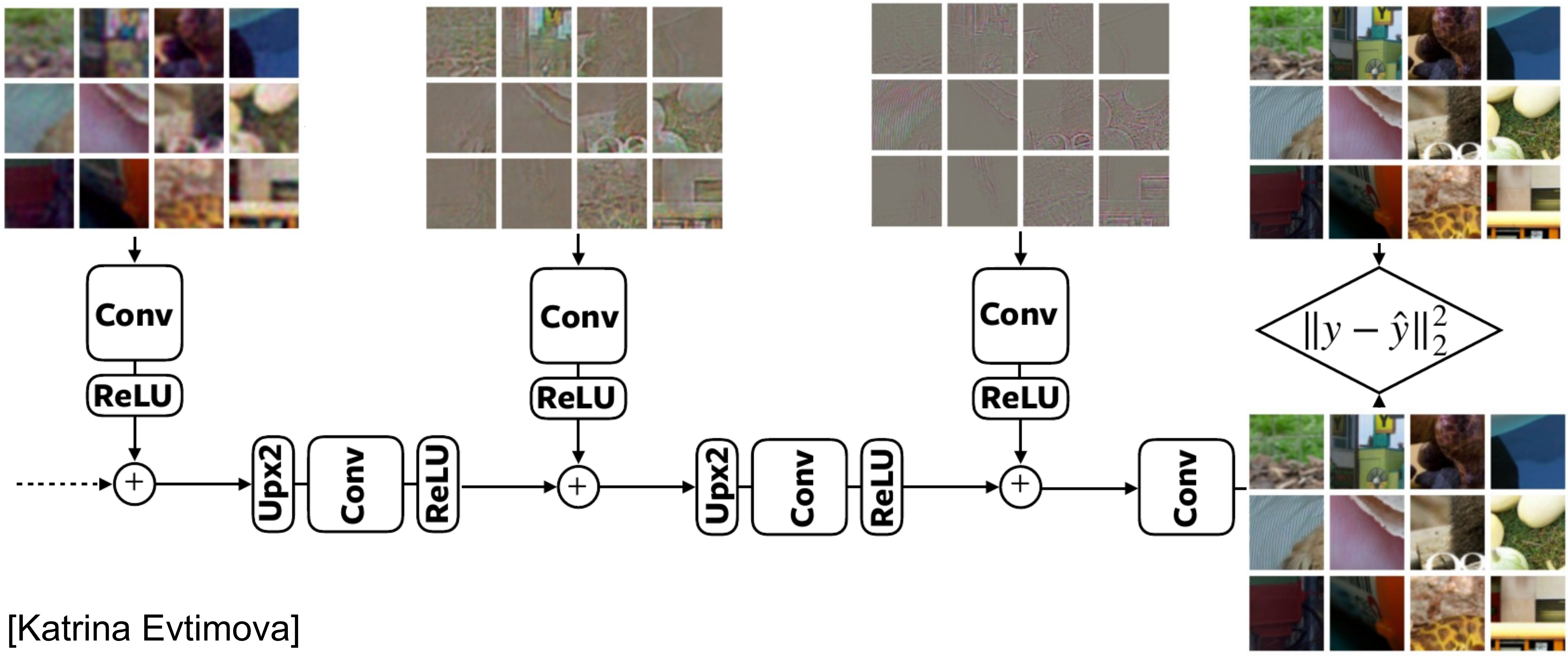
Multilayer Convolutional Sparse Modeling

- **Learning hierarchical representations**



Multilayer Convolutional Sparse Modeling

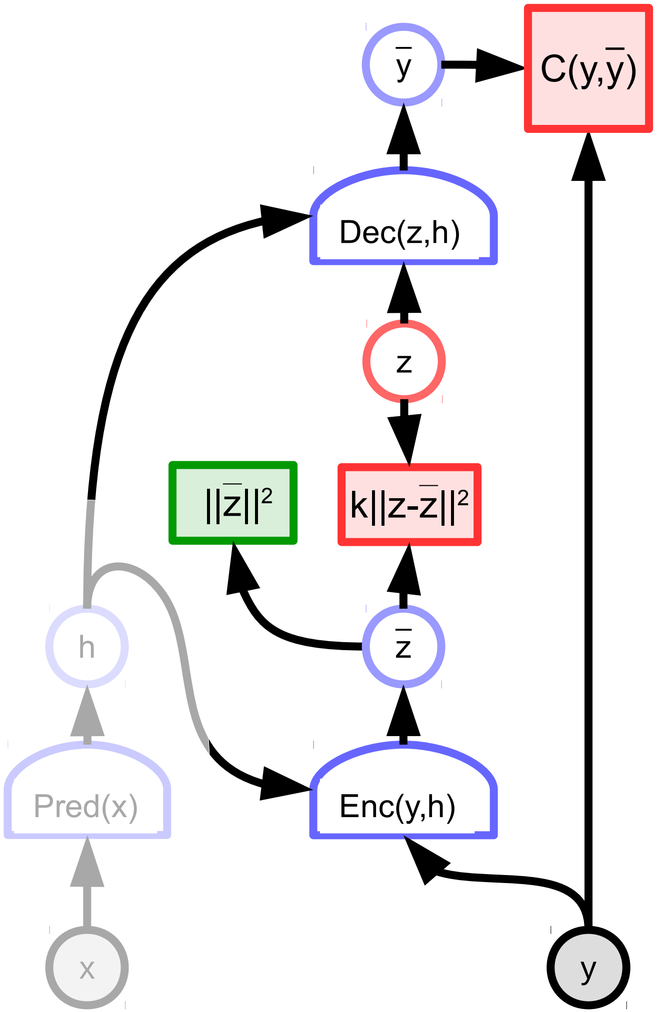
- Reconstructions from Z2, Z1, Z0 and all of (Z2,Z1,Z0)



[Katrina Evtimova]

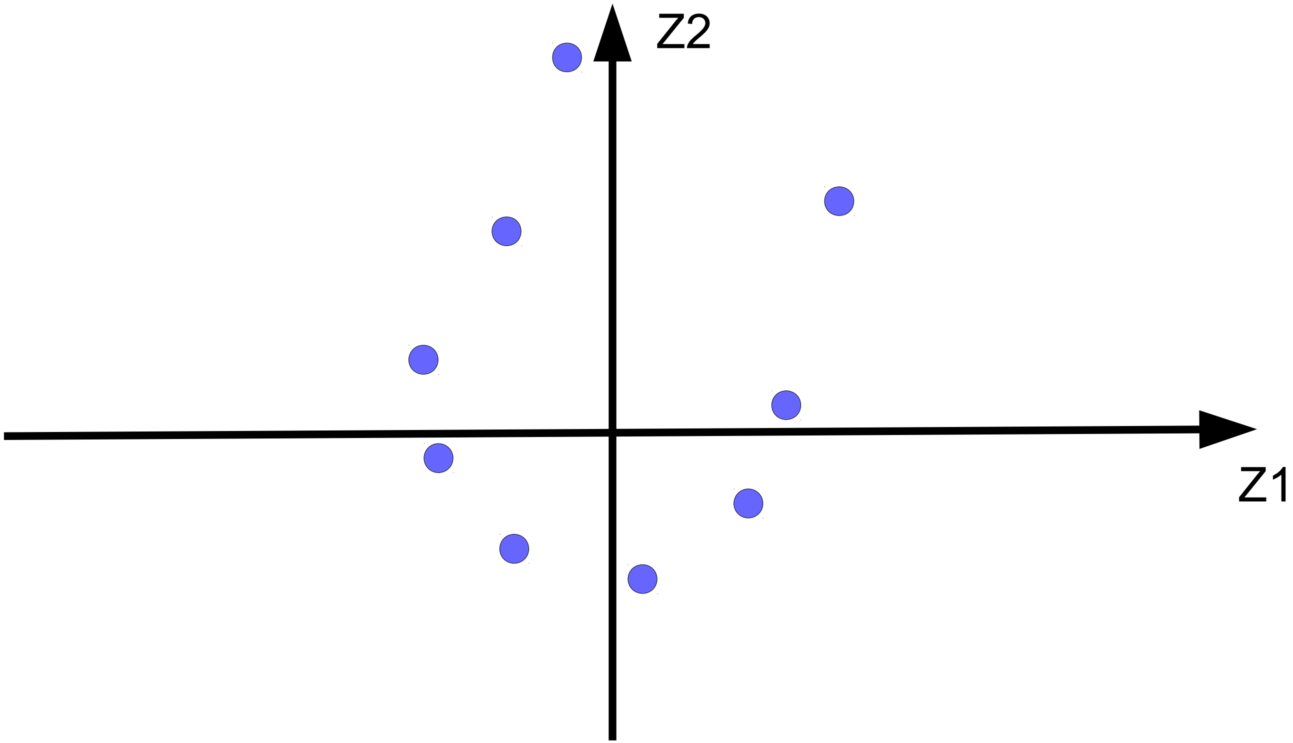
Variational Auto-Encoder

- **Limiting the information capacity of the code by adding Gaussian noise**
- **The energy term $k||z-\bar{z}||^2$ is seen as the log of a prior from which to sample z**
- **The encoder output is regularized to have a mean and a variance close to zero.**



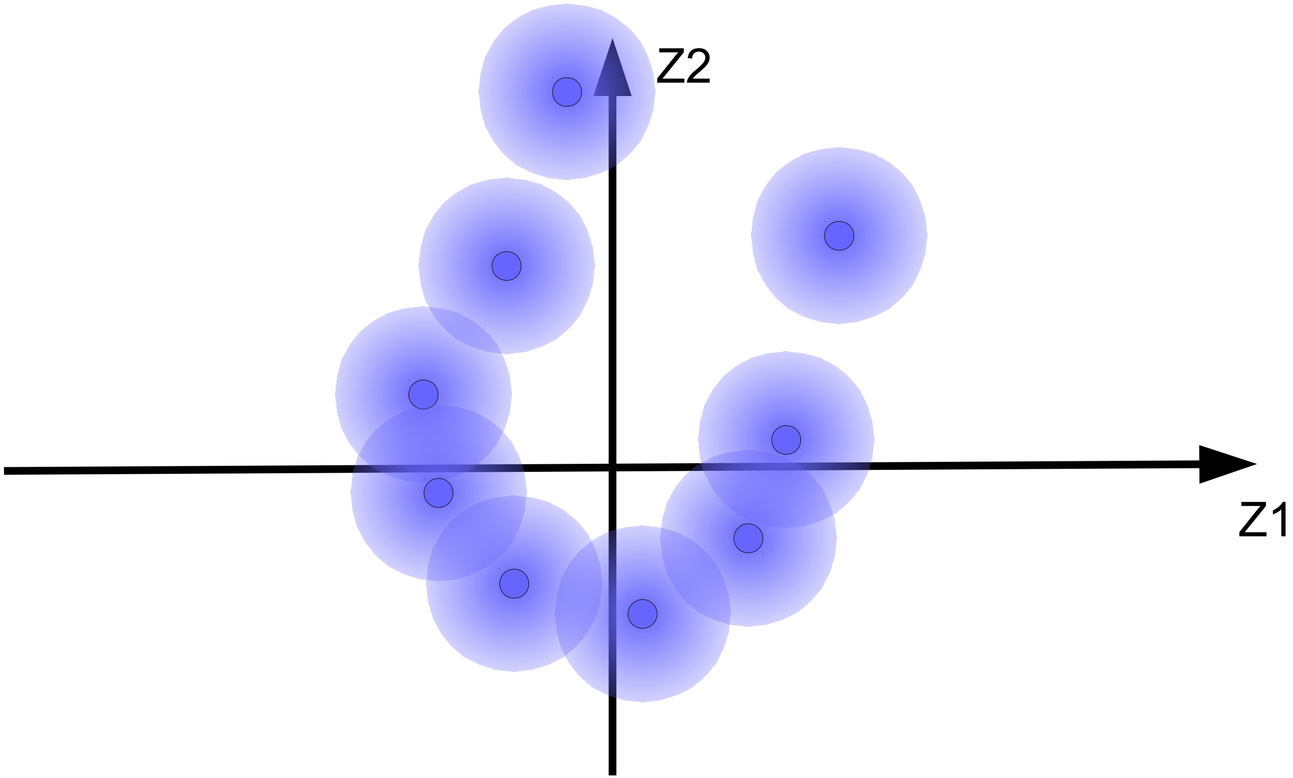
Variational Auto-Encoder

- **Code vectors for training samples**



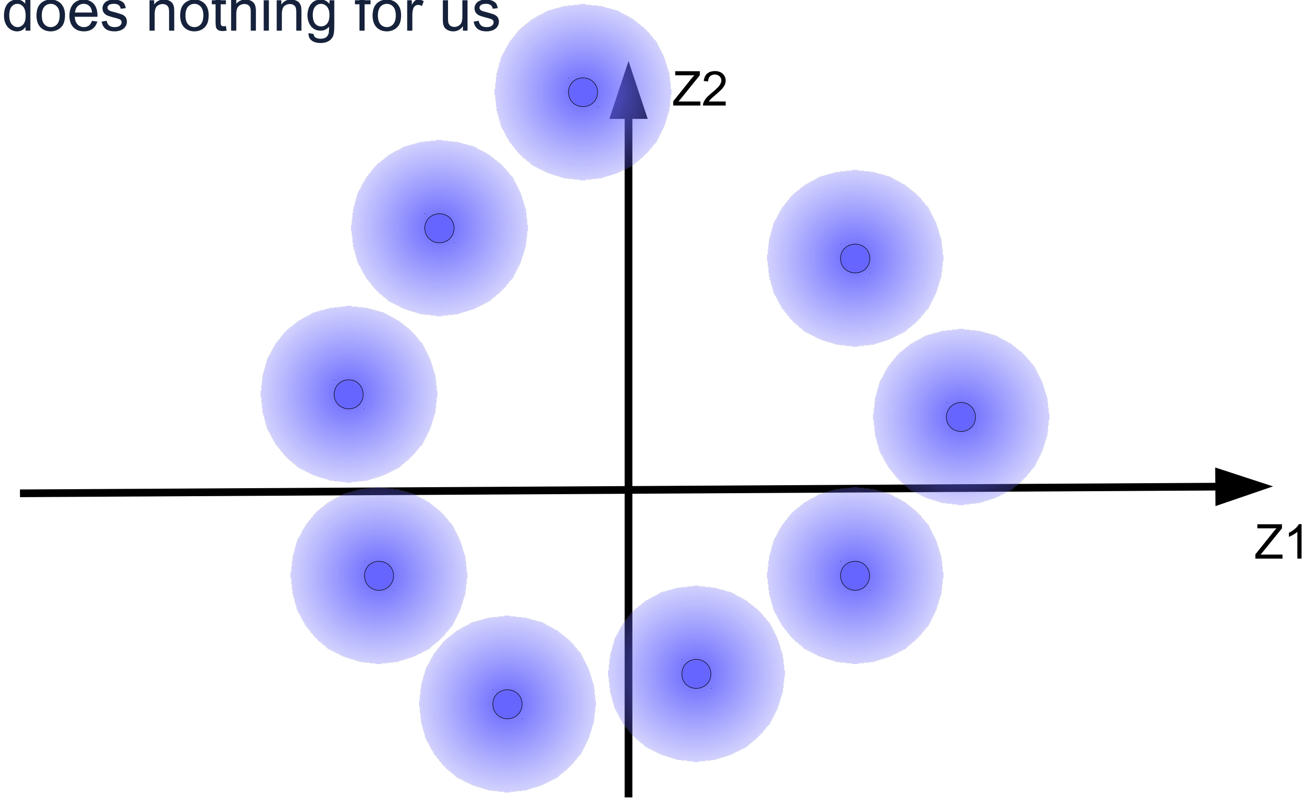
Variational Auto-Encoder

- **Code vectors for training sample with Gaussian noise**
  - Some fuzzy balls overlap, causing bad reconstructions



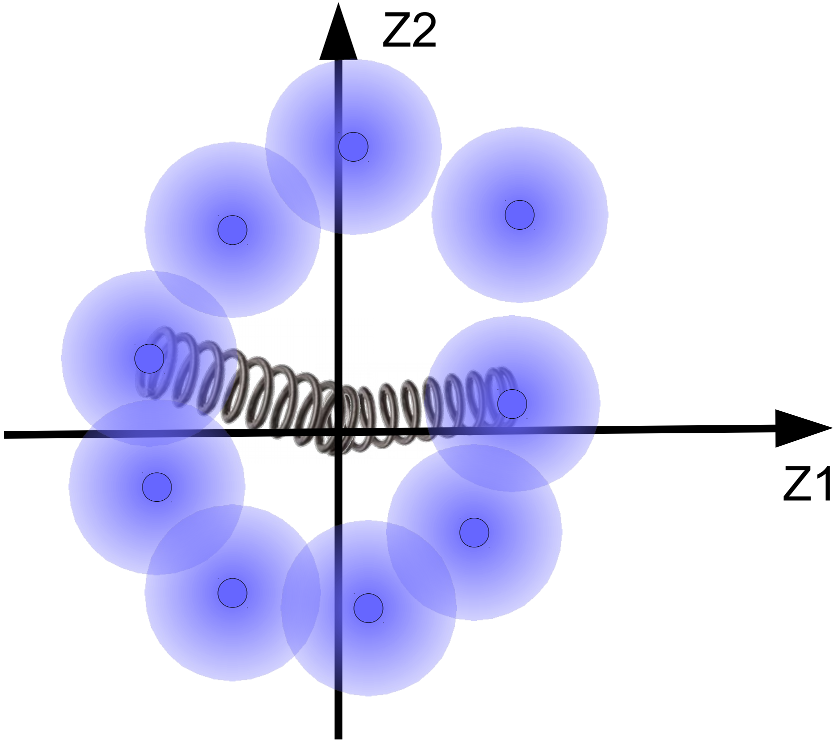
Variational Auto-Encoder

- **The code vectors want to move away from each other to minimize reconstruction error**
  - But that does nothing for us



Variational Auto-Encoder

- **Attach the balls to the center with a sping, so they don't fly away**
  - Minimize the square distances of the balls to the origin
- **Center the balls around the origin**
  - Make the center of mass zero
- **Make the sizes of the balls close to 1 in each dimension**
  - Through a so-called KL term



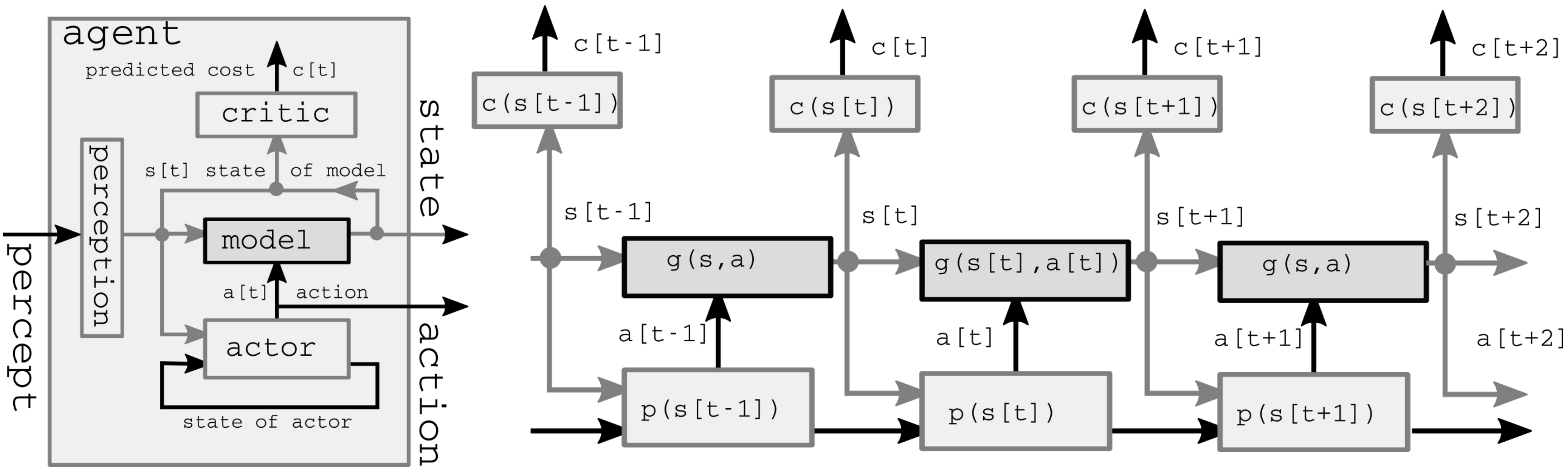
FACEBOOK AI

Learning a Forward Model for Autonomous Driving

Learning to predict what others around you will do

A Forward Model of the World

- **Learning forward models for control**
  - s[t+1] = g(s[t], a[t], z[t])
  - Classical optimal control: find a sequence of action that minimize the cost, according to the predictions of the forward model

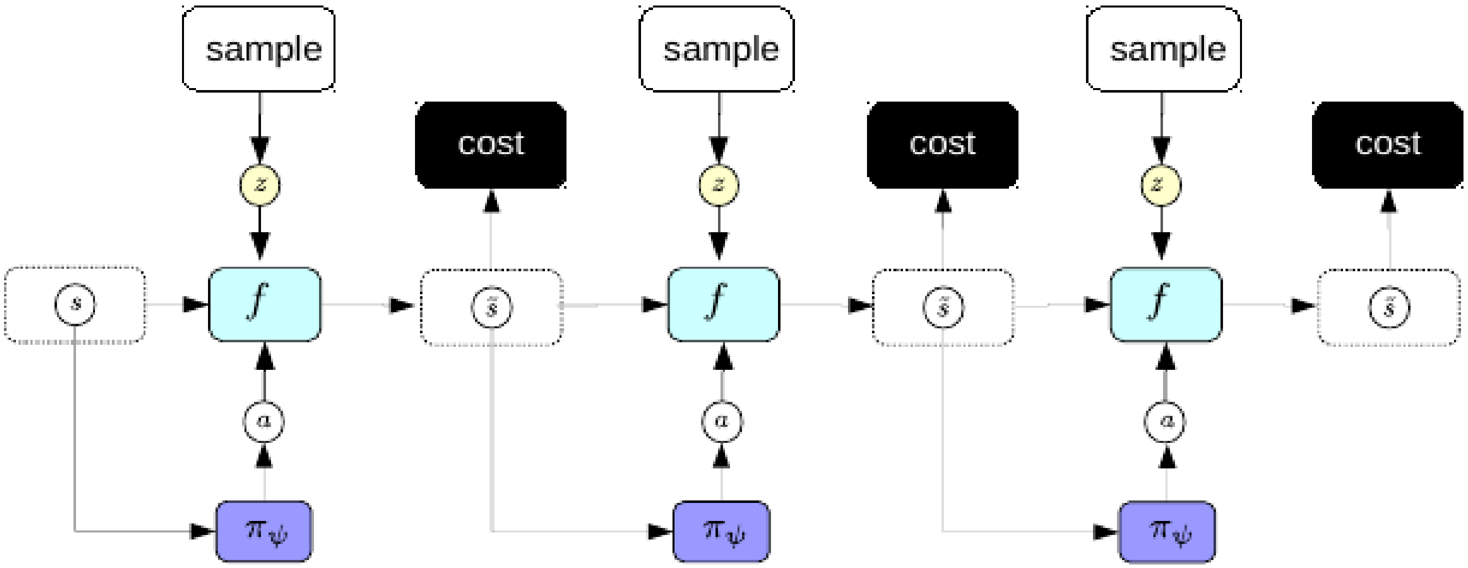


Planning/learning using a self-supervised predictive world model

- **Feed initial state**
- **Run the forward model**
- **Backpropagate gradient of cost**
- **Act**
  - (model-predictive control)

  or

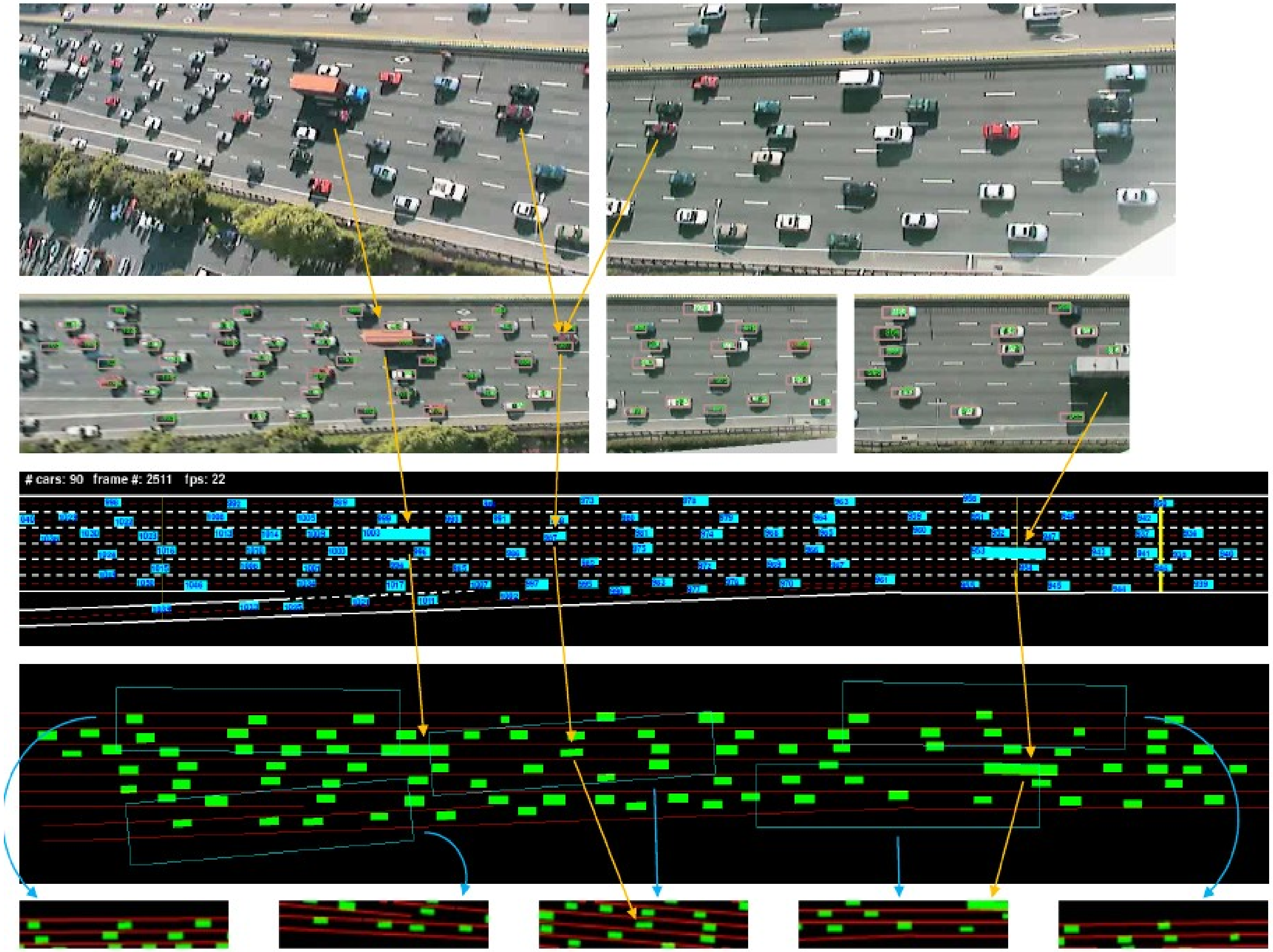
- **Use the gradient t train a policy network.**
- **Iterate**



Stochastic policy network (optimized)

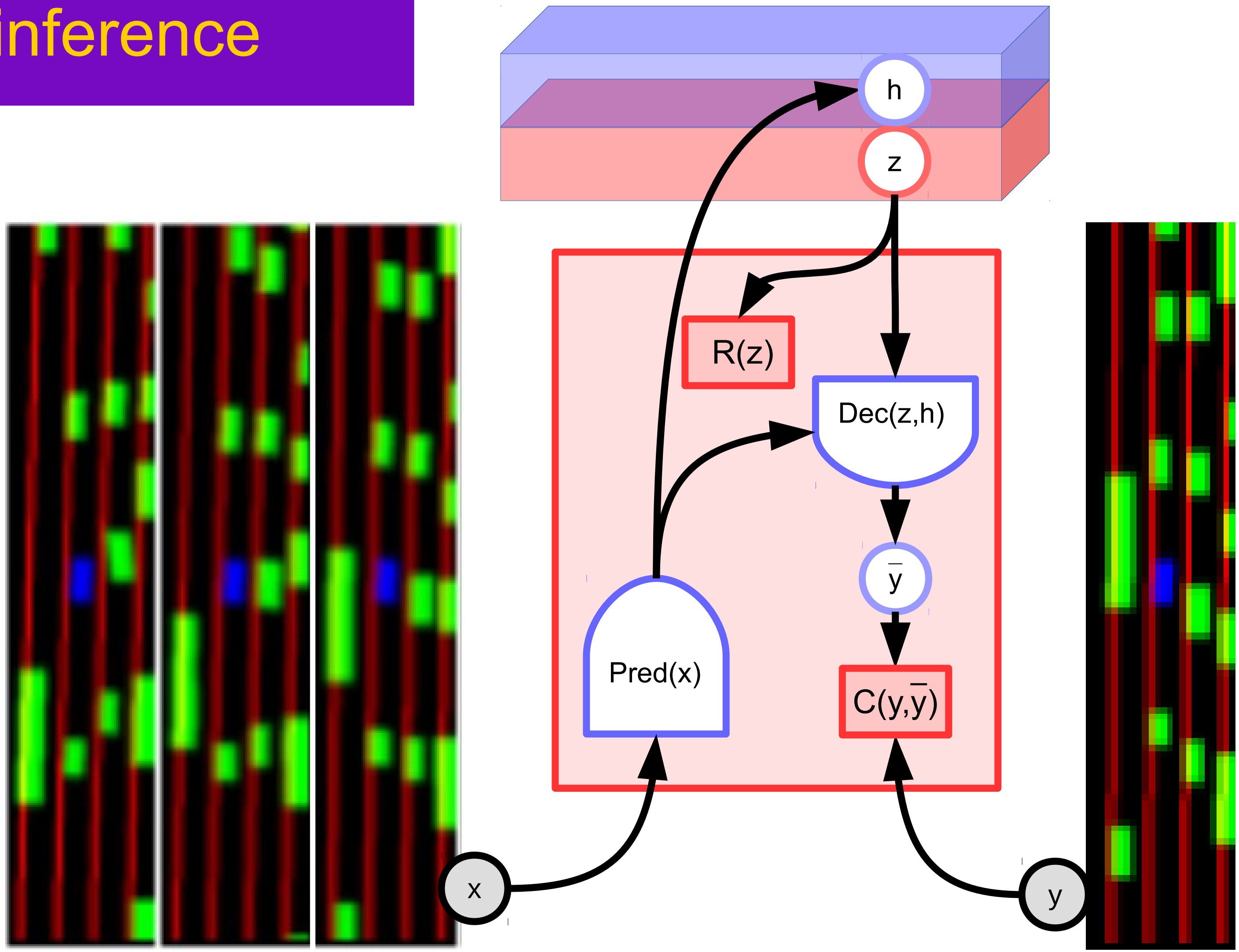
Using Forward Models to Plan (and to learn to drive)

- **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



Video Prediction: inference

- **After training:**
  - Observe frames
  - Compute h
  - Sample z
  - Predict next frame

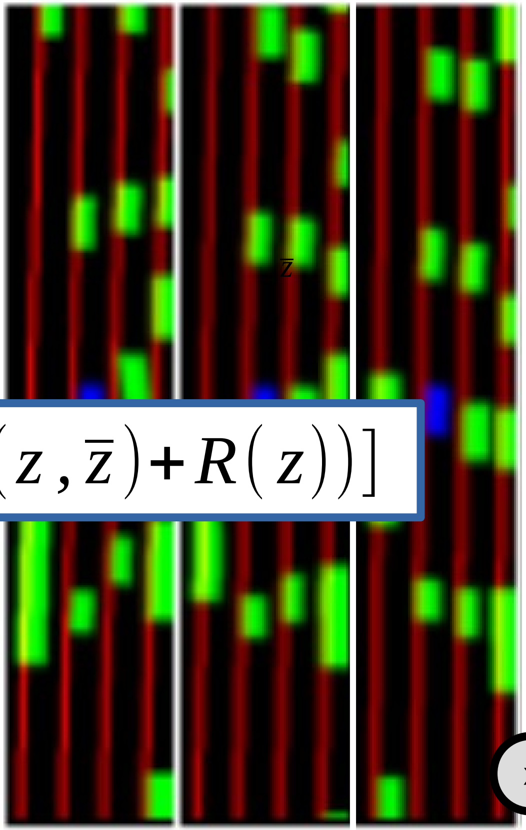


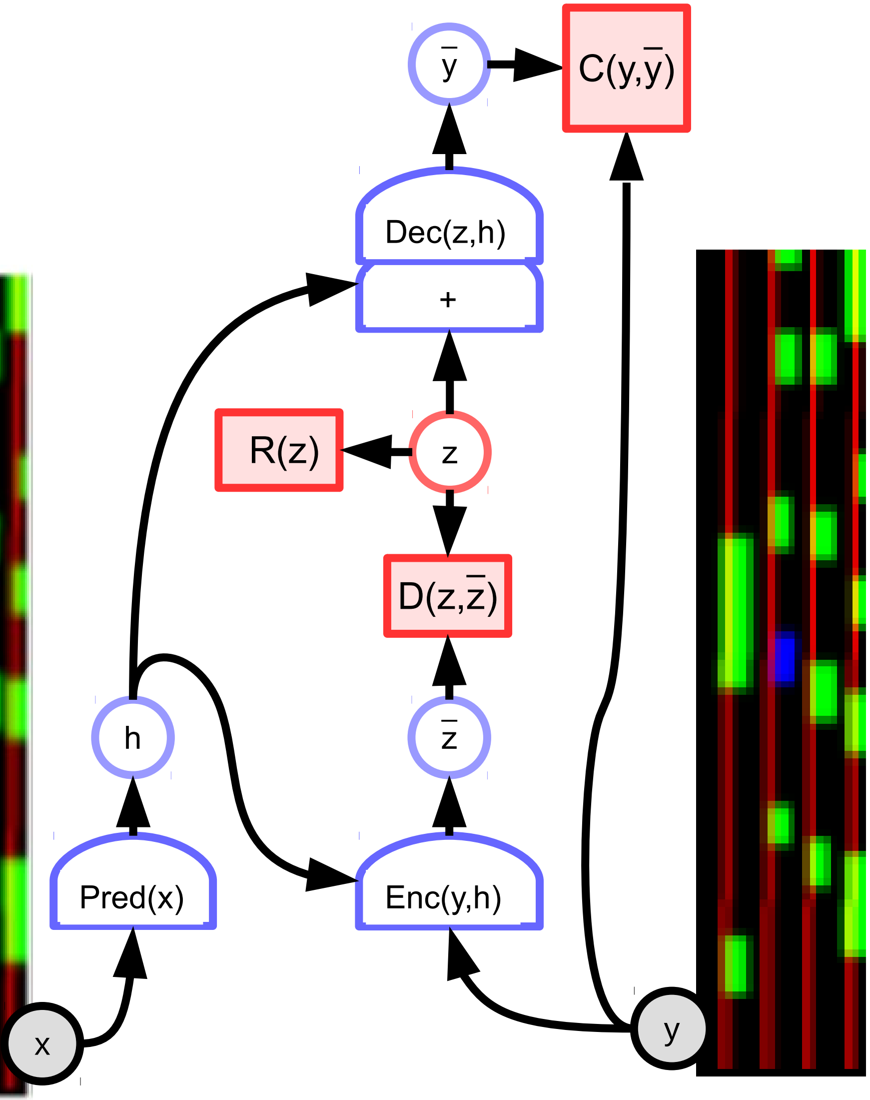
Video Prediction: training

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

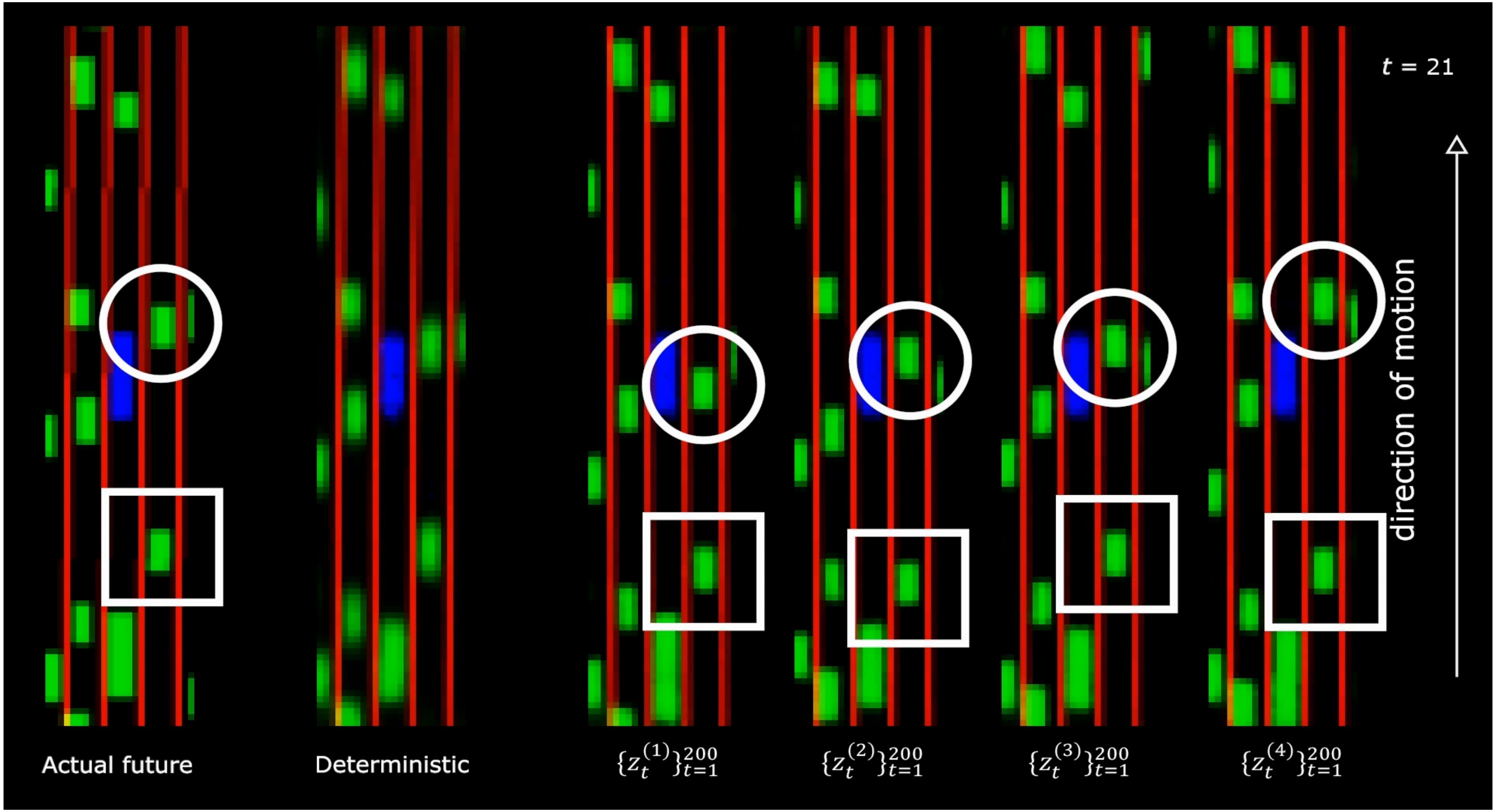
$$\boxed{P(z/\bar{z}) \propto \exp[-\beta(D(z,\bar{z})+R(z))]}$$

  - Predict next frame
  - backprop



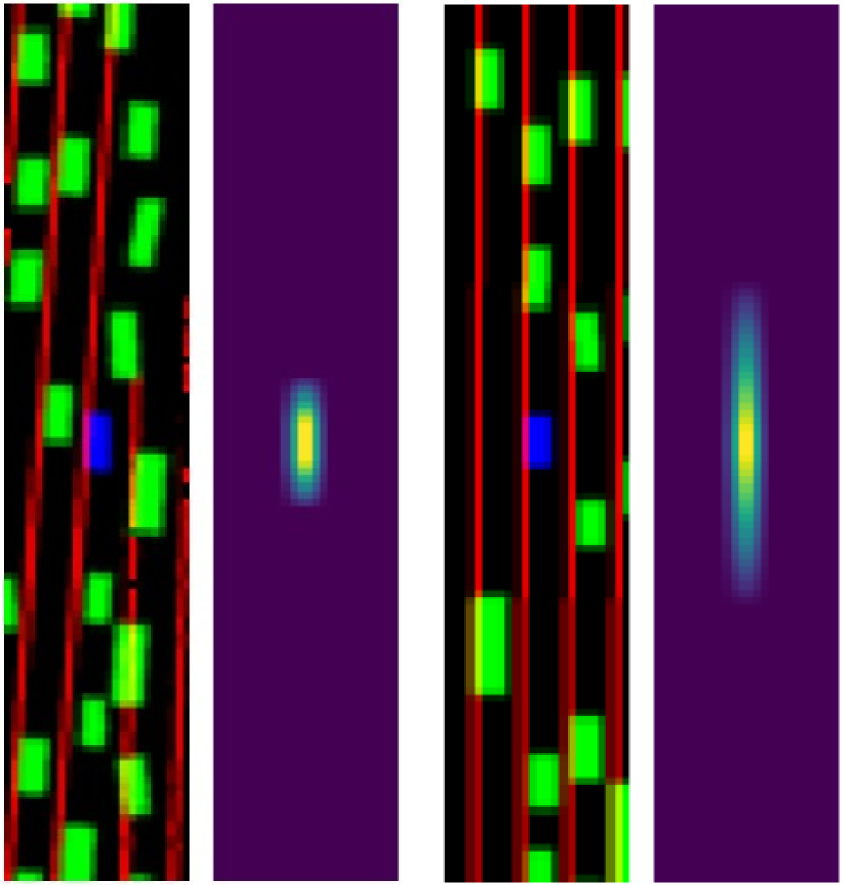


Actual, Deterministic, VAE+Dropout Predictor/encoder



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.

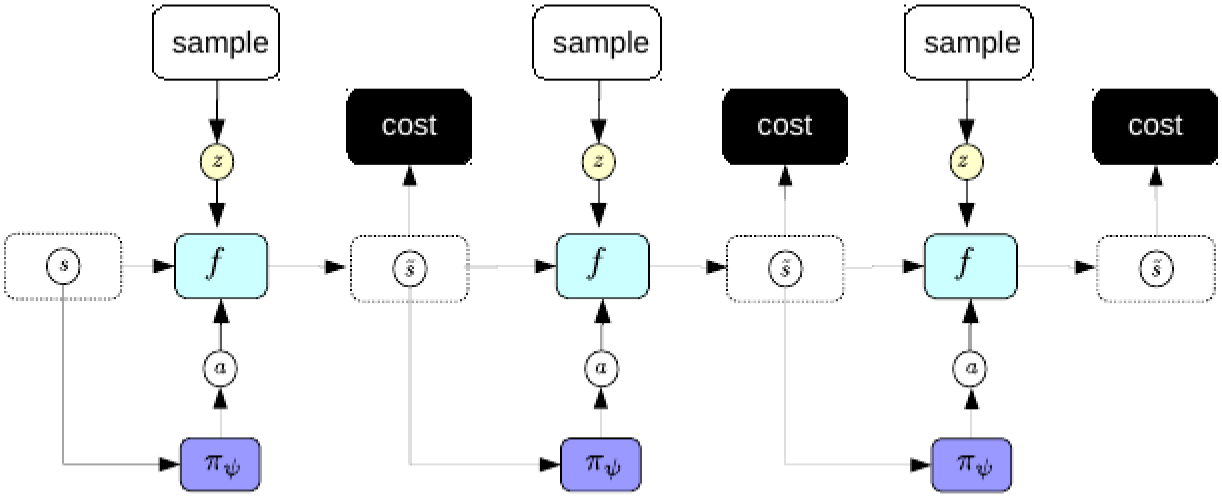


(a) 19.8 km/h (b) 50.3 km/h

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**

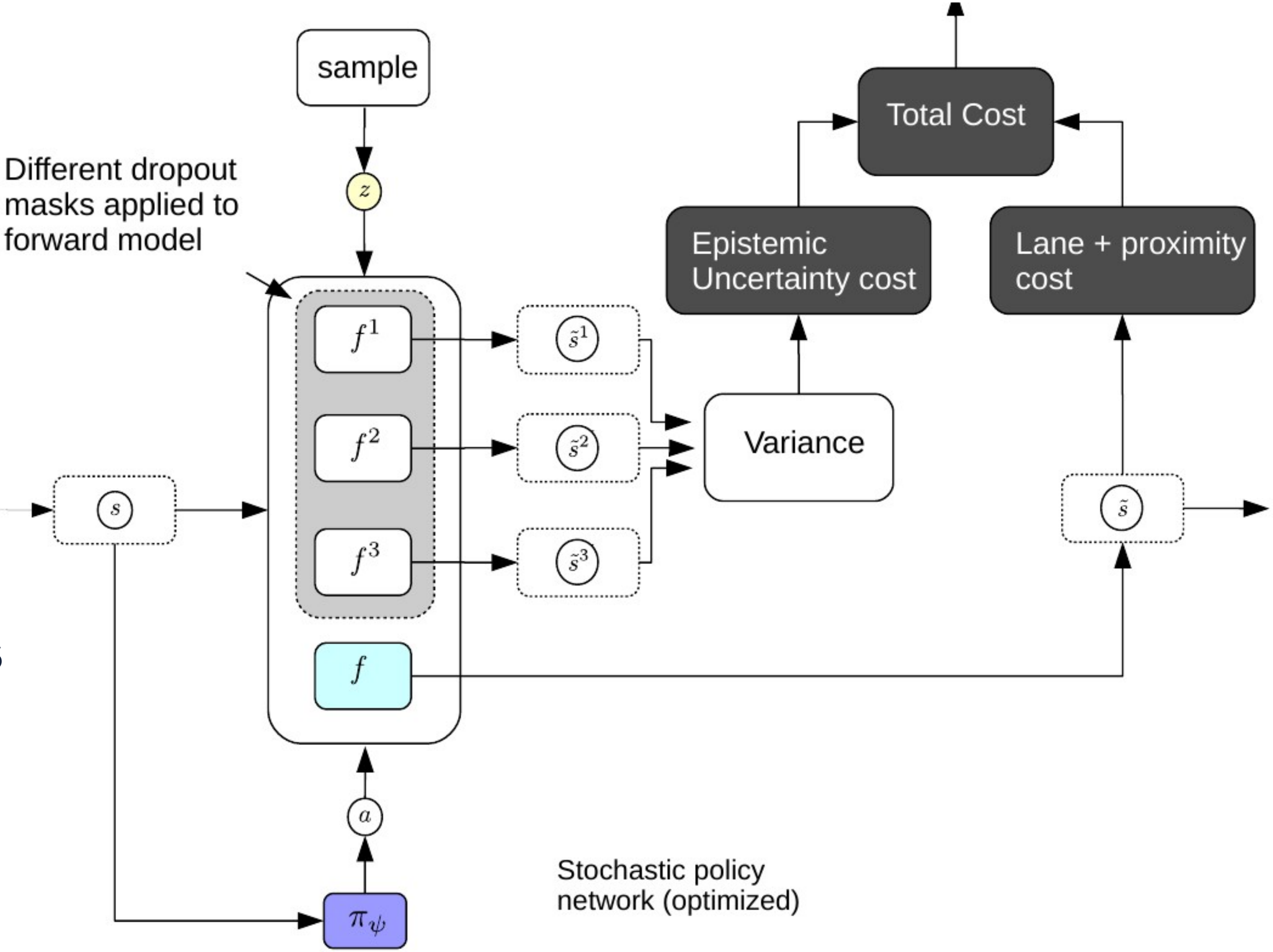
- **No need for planning at run time.**



Stochastic policy network (optimized)

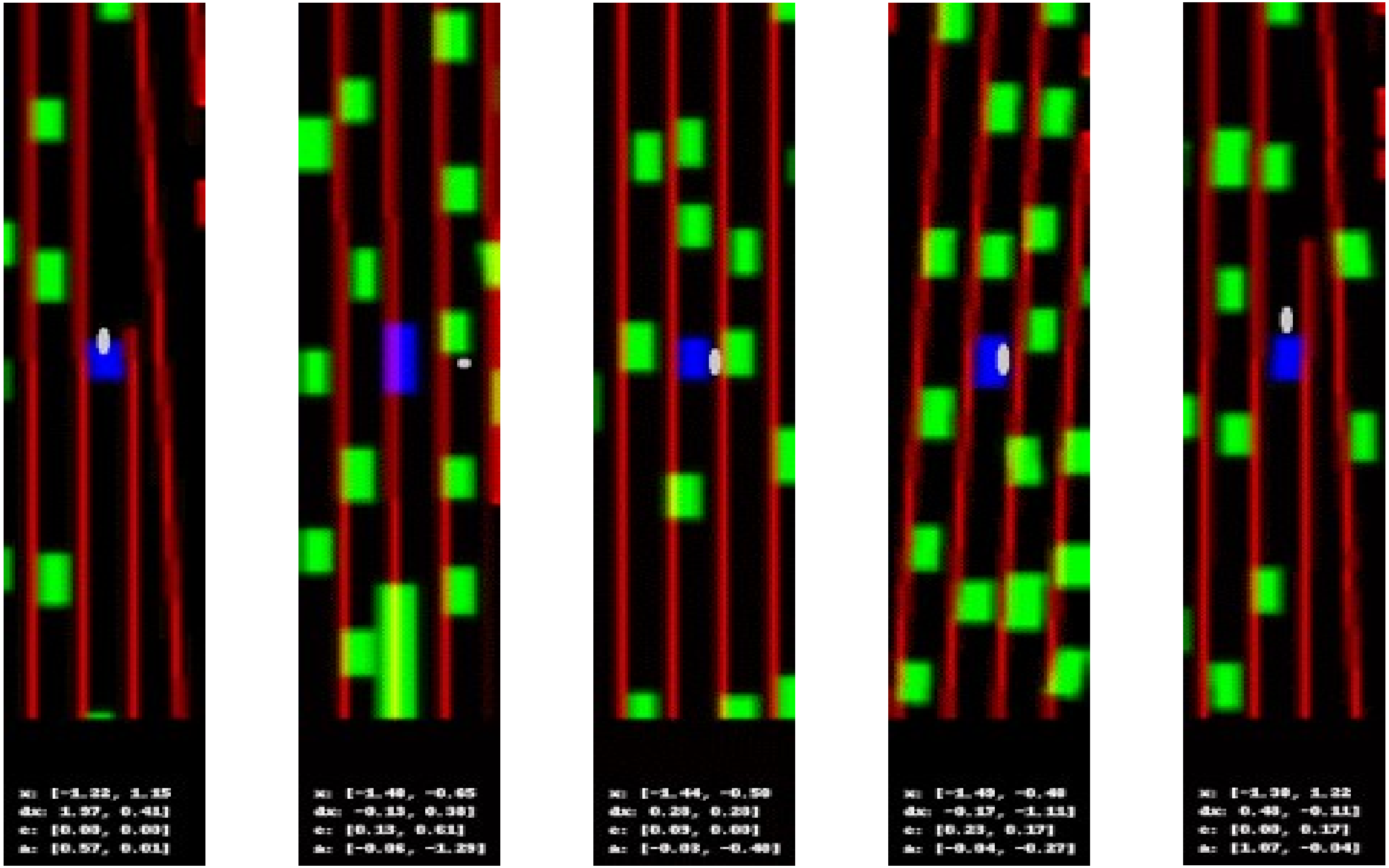
Adding an Uncertainty Cost (doesn't work without it)

- **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**



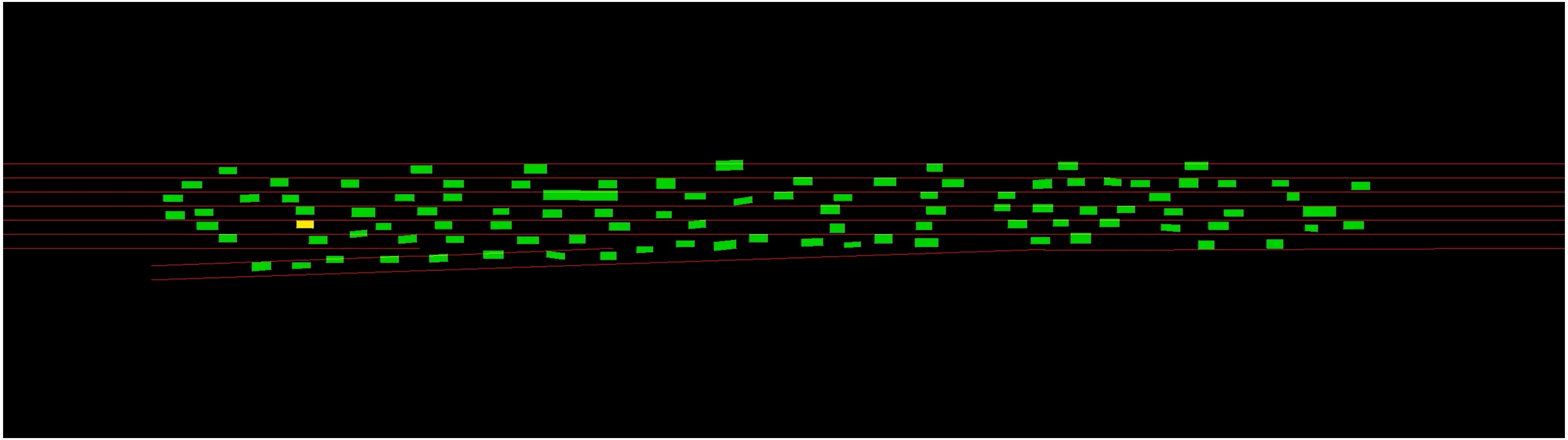
Y. LeCun

Driving an Invisible Car in “Real” Traffic



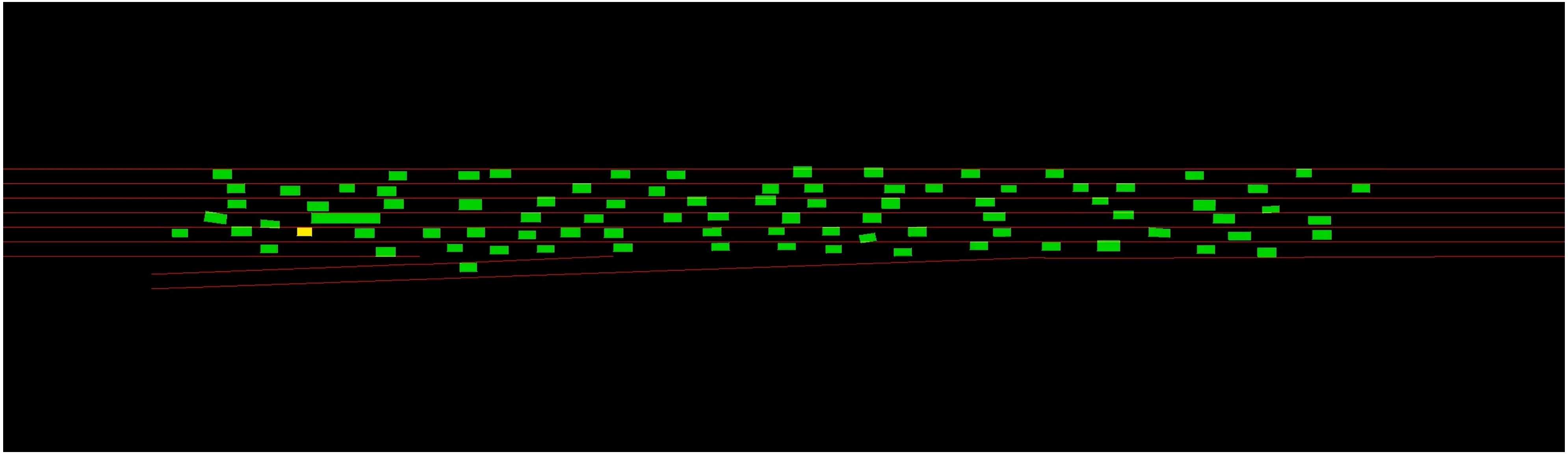
Driving!

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



Driving!

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



Take-Home Messages

- **SSL is the future**
  - Hierarchical feature learning for low-resource tasks
  - Hierarchical feature learning for **massive** networks
  - Learning Forward Models for Model-Based Control/RL
- **My money is on:**
  - Energy-Based Approaches
  - Latent-variable models to handle multimodality
  - Regularized Latent Variable models
  - Sparse Latent Variable Models
  - Latent Variable Prediction through a Trainable Encoder

FACEBOOK AI

Thank You!