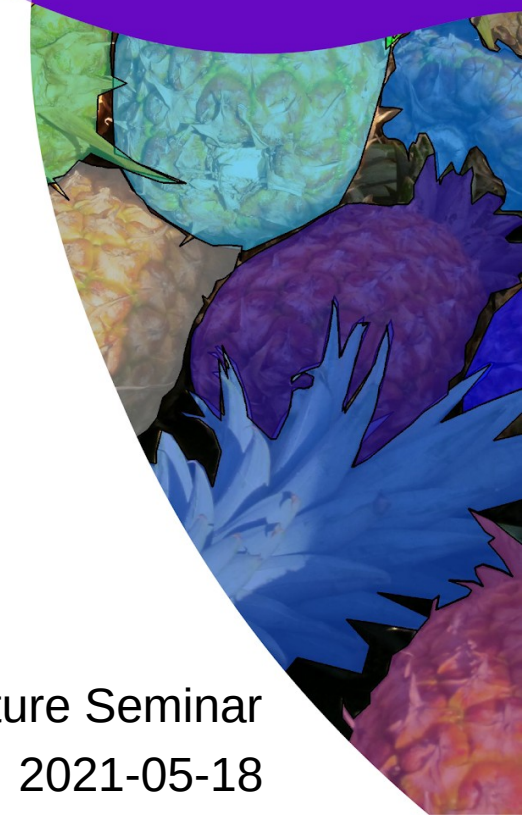
NEW YORK UNIVERSITY

# Energy-Based Learning

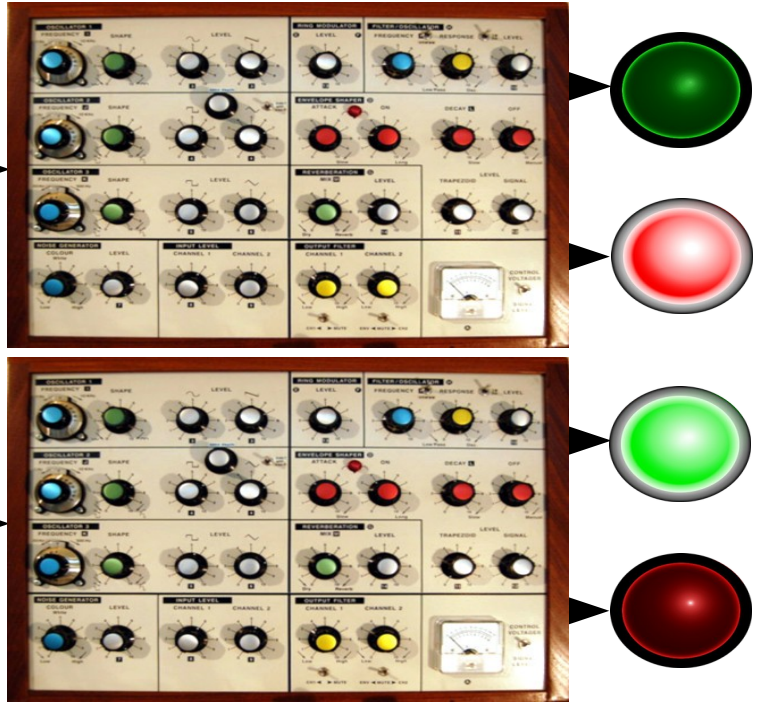


Yann LeCun
NYU - Courant Institute & Center for Data Science
Facebook AI Research
http://yann.lecun.com

Harvard Mathematical Picture Seminar
2021-05-18

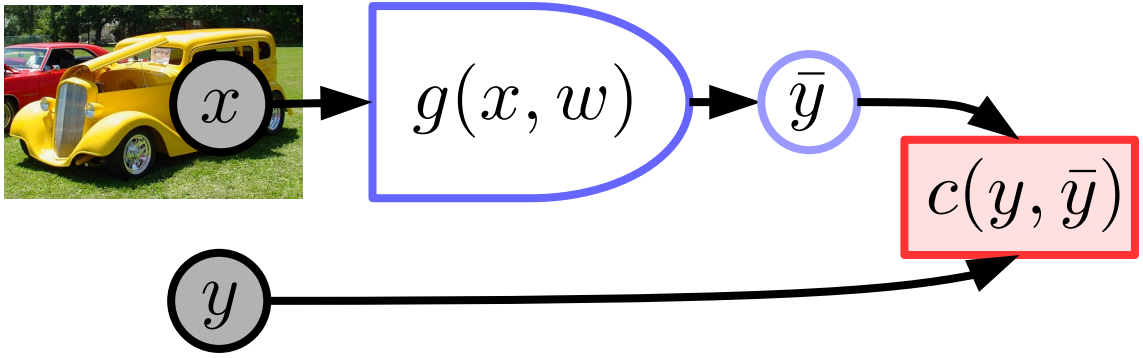
# Machine Learning, Supervised Learning

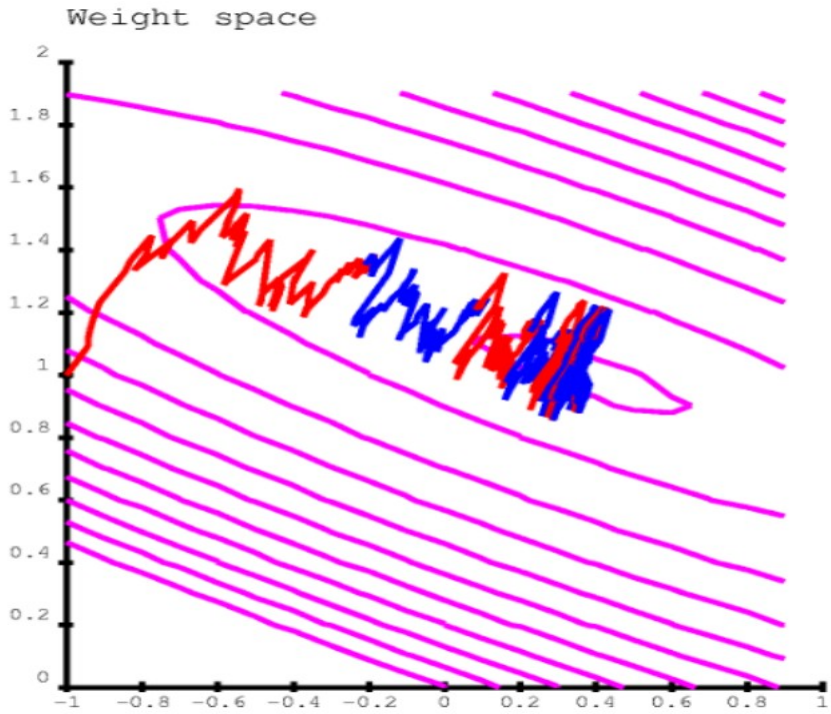
- **Training a machine by showing examples instead of programming it**
- **When the output is wrong, tweak the parameters of the machine**
- **Works well for:**
  - Speech → words
  - Image → categories
  - Portrait → name
  - Photo → caption
  - Text → topic
  - ….



# 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



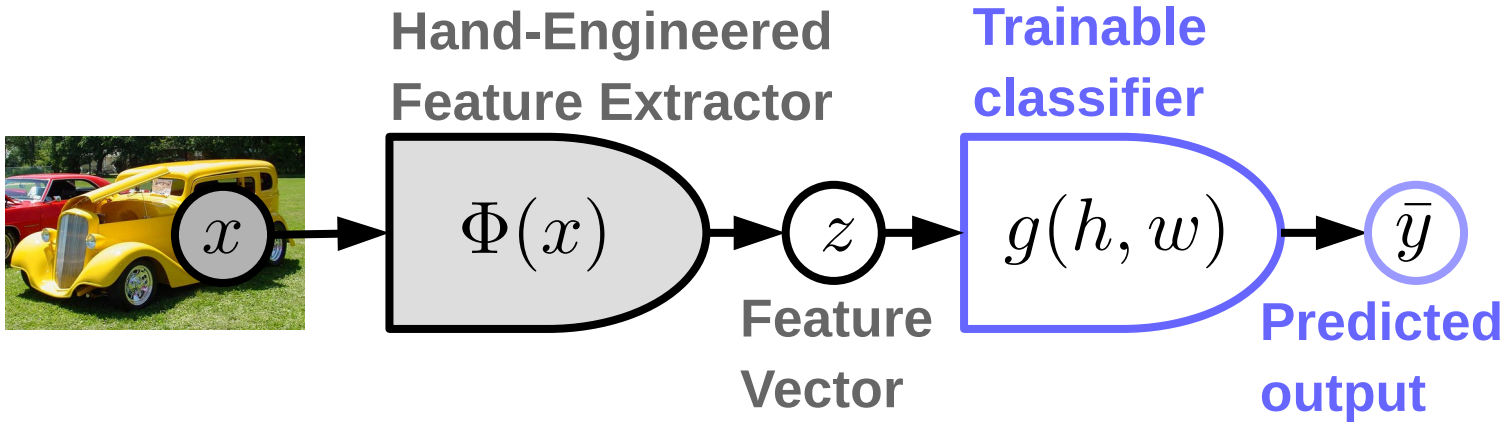




$$L(x,y,w) = c(y, g(x,w))$$

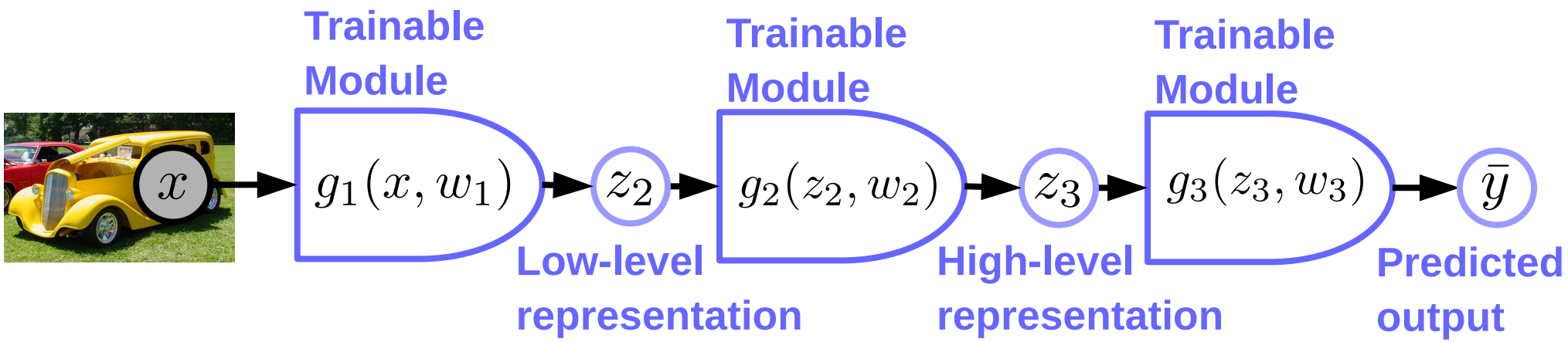
$$w \leftarrow w - \eta \frac{\partial L(x,y,w)}{w}$$

# Traditional Machine Learning → Deep Learning

- **Traditional Machine Learning & Pattern Recognition**




- **Deep Learning: learning hierarchical representations**

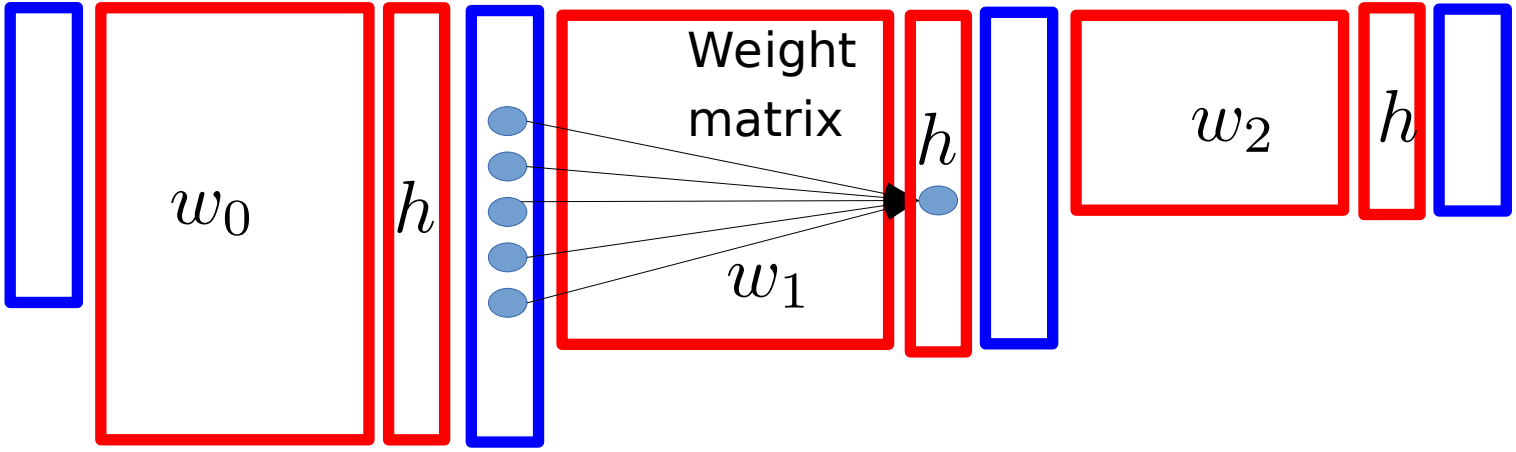


# (Deep) Multi-Layer Neural Nets

- 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

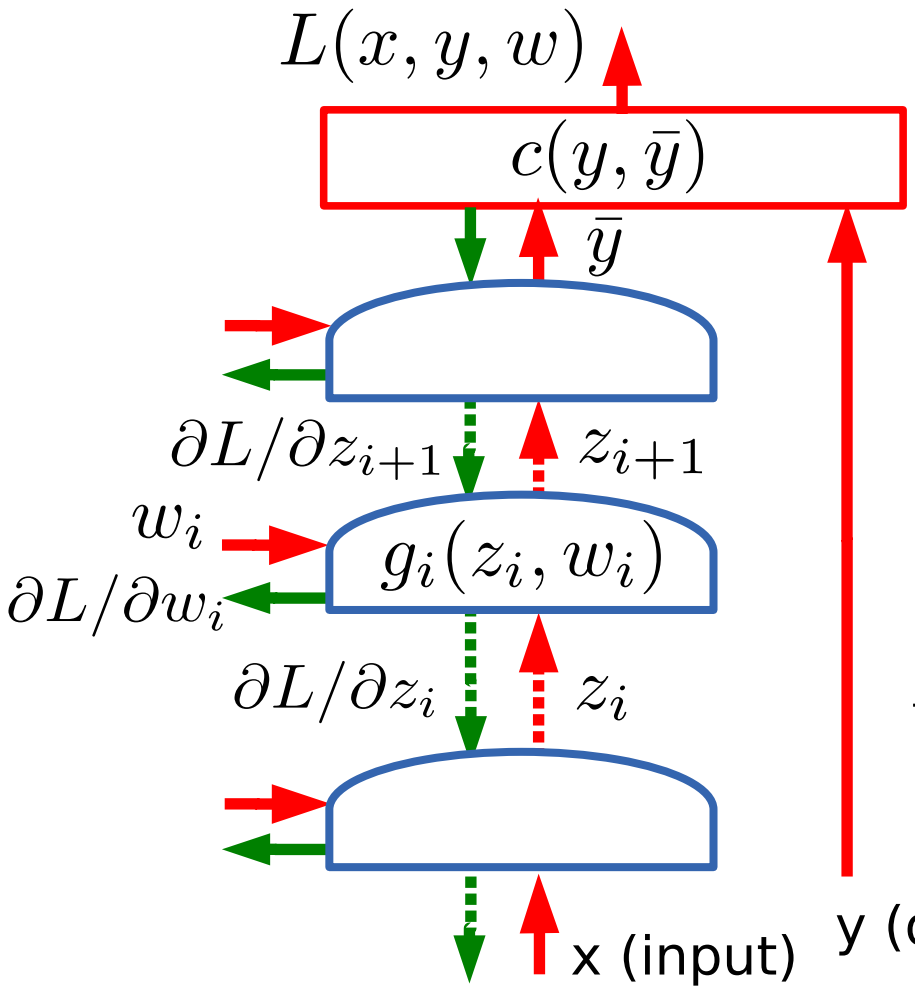
$$ReLU(x) = max(x, 0)$$

Ceci est une voiture




$$x \rightarrow w_0 x \rightarrow s_1 \rightarrow h(s_1) \rightarrow z_1 \rightarrow w_1 z_1 \rightarrow s_2 \rightarrow h(s_1) \rightarrow z_2 \rightarrow w_2 z_2 \rightarrow s_3$$

# Computing Gradients by Back-Propagation




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

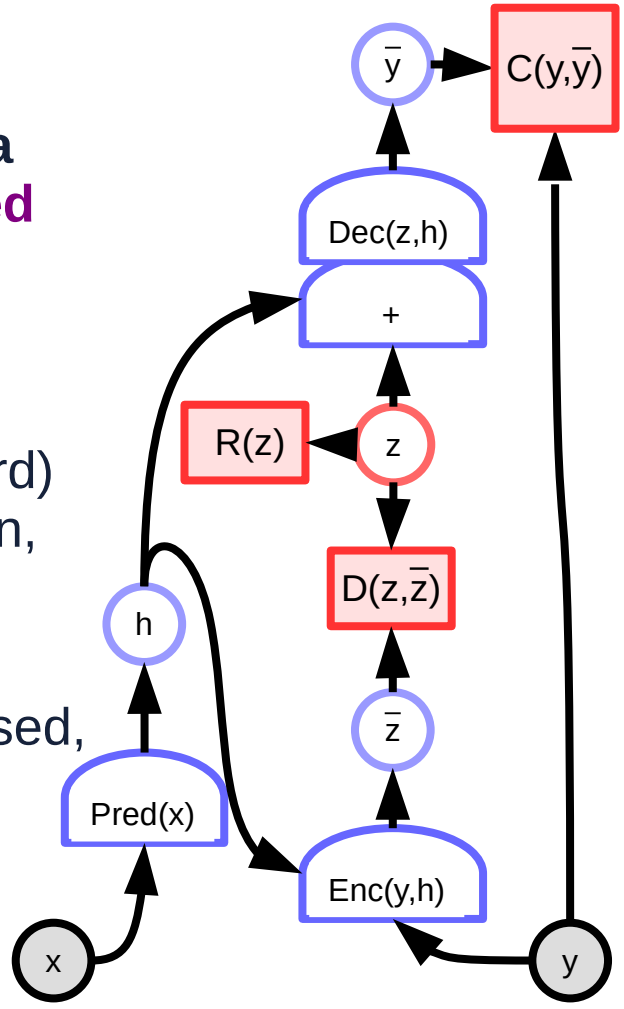
$$\frac{\partial L}{\partial z_i} = \frac{\partial L}{\partial z_{i+1}} \frac{\partial z_{i+1}}{\partial z_i} = \frac{\partial L}{\partial z_{i+1}} \frac{\partial g_i(z_i, w_i)}{\partial z_i}$$

- **Backprop for the weight gradients:**

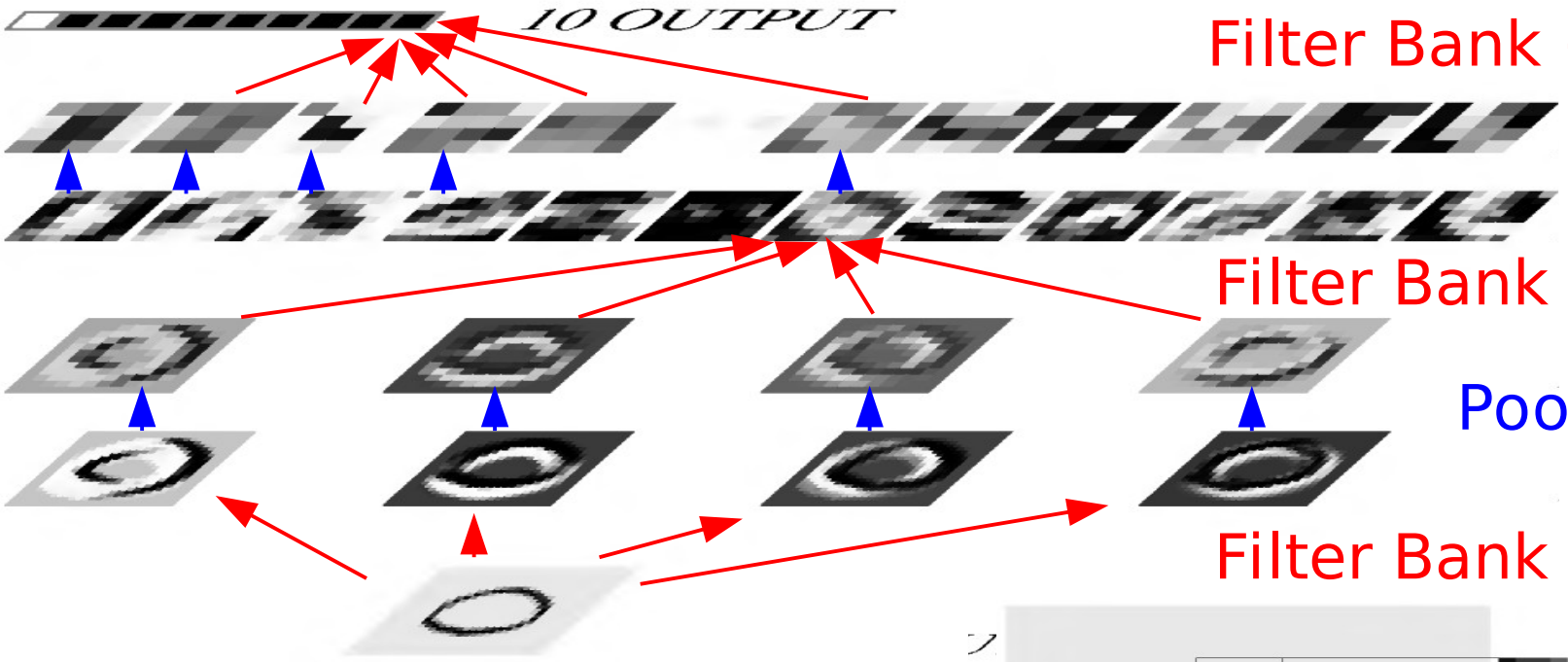
$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial z_{i+1}} \frac{\partial z_{i+1}}{\partial w_i} = \frac{\partial L}{\partial z_{i+1}} \frac{\partial g_i(z_i, w_i)}{\partial w_i}$$

# What is Deep Learning?

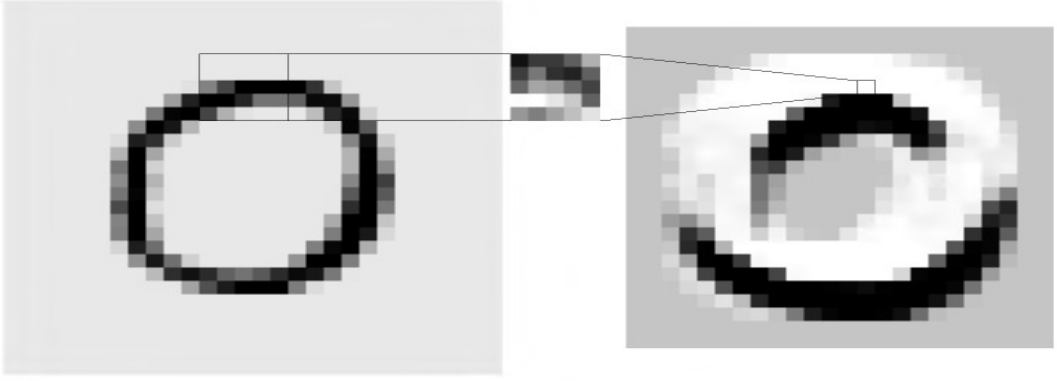
- **Definition: Deep Learning is building a system by assembling parameterized modules into a (possibly dynamic) computation graph, and training it to perform a task by optimizing the parameters using a gradient-based method.**
  - Graph can be defined dynamically by input-dependent programs: differentiable 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.

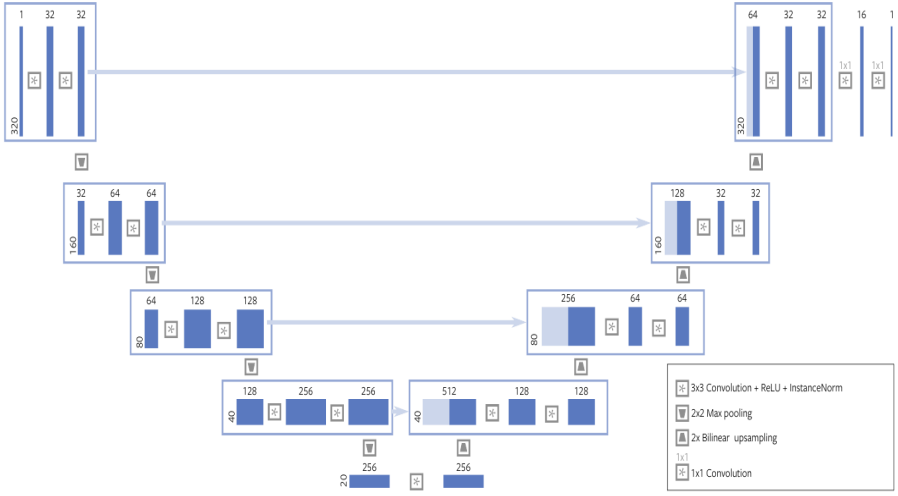


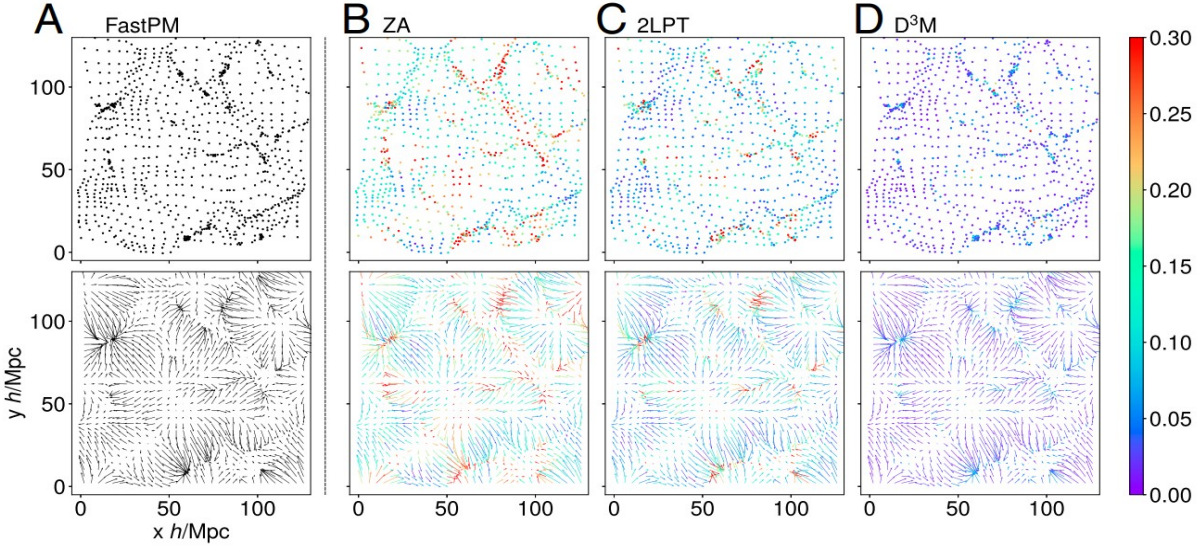
# ConvNets in Astrophysics [He et al. PNAS 07/2019]

## 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]

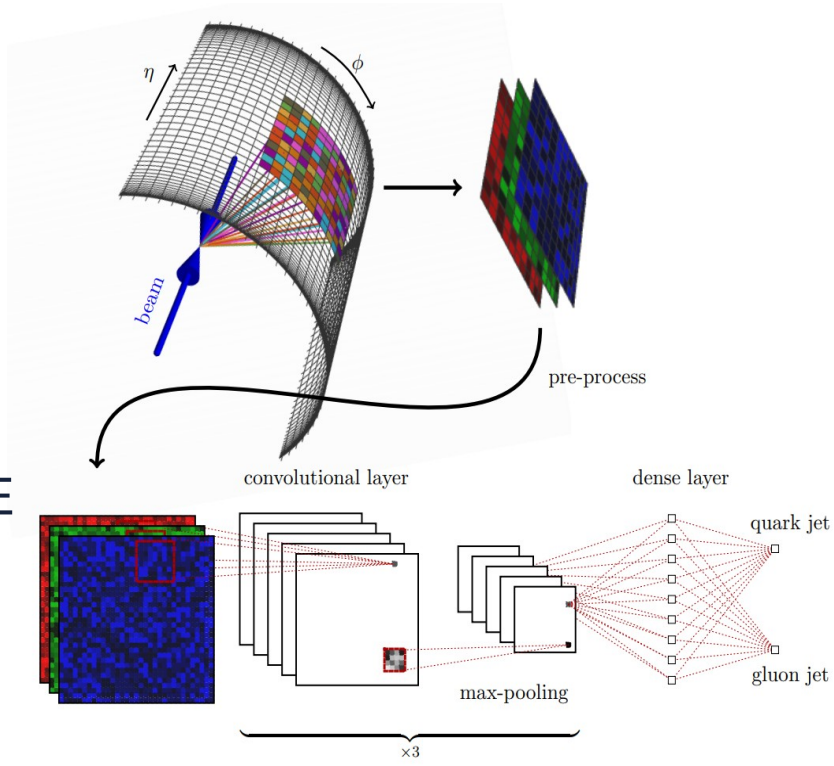
- 1. Train a coarse-grained 3D U-Net to approximate a fine-grained simulation on a small volume
- 2. Use it for a simulation on a large volume (the early universe)



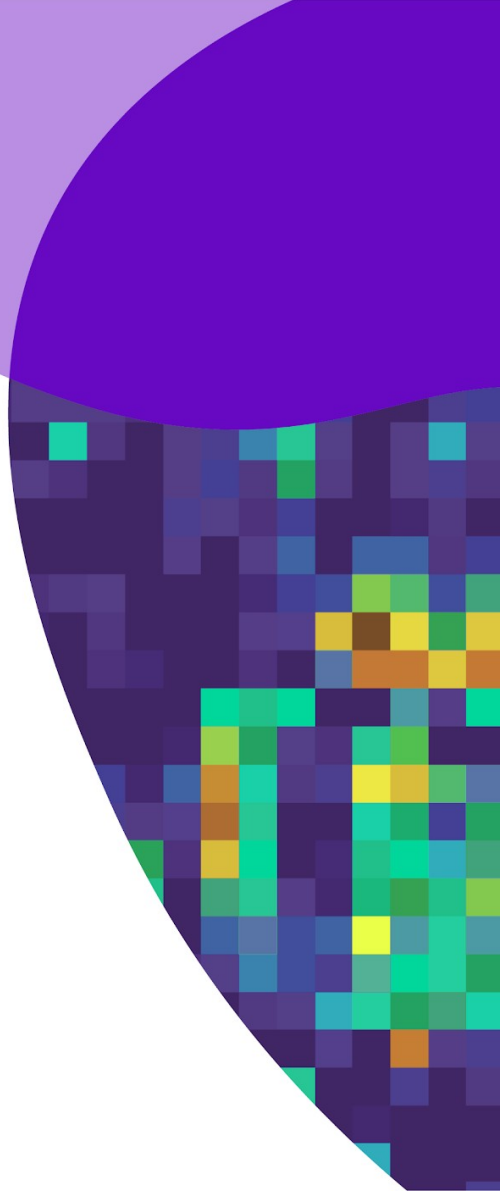
# ConvNets (and Deep Learning) in Physics

- **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**
  - Infering constants from observations
  - Statistical studies of galaxies,
  - Dark matter through gravitational lensing



FACEBOOK AI

# Backpropagation as Lagrangian Optimization



Optimization under constraints

# Reformulating Deep Learning

- **Loss**

$$L(x, y, w) = C(z_K, y) \text{ such that } z_{k+1} = g_k(z_k, w_k), \quad z_0 = x$$

- **Lagrangian for optimization under constraints**

$$L(x, y, z, \lambda, w) = C(z_K, y) + \sum_{k=0}^{K-1} \lambda_k^T (z_{k+1} - g_k(z_k, w_k))$$

- **Optimality conditions:**

$$\frac{\partial L(x, y, z, \lambda, w)}{\partial z_k} = 0, \quad \frac{\partial L(x, y, z, \lambda, w)}{\partial \lambda_k} = 0, \quad \frac{\partial L(x, y, z, \lambda, w)}{\partial w_k} = 0$$

[LeCun: “A theoretical framework for back-propagation”, 1988]

# Reformulating Deep Learning

$$\frac{\partial L(x,y,z,\lambda,w)}{\partial \lambda_k} = z_{k+1} - g_k(z_k, w_k) = 0 \Longrightarrow z_{k+1} = g_k(z_k, w_k)$$

$$\frac{\partial L(x,y,z,\lambda,w)}{\partial z_k} = \lambda_{k-1}^T - \lambda_k^T \frac{\partial g_{k-1}(z_{k-1}, w_{k-1})}{\partial z_k} = 0 \Longrightarrow$$

- **Backprop!**
  - Lambda is the gradient

$$\lambda_{k-1} = \frac{\partial g_{k-1}(z_{k-1}, w_{k-1})}{\partial z_k}^T \lambda_k$$

$$\frac{\partial L(x,y,z,\lambda,w)}{\partial w_k} = \lambda_{k+1}^T \frac{\partial g_k(z_k, w_k)}{\partial w_k}$$

# Three challenges for AI & Machine Learning

- **1. Learning with fewer labeled samples and/or fewer trials**
  - Supervised and reinforcement learning require too much samples/trails
  - Self-supervised learning / learning dependencies / to fill in the blanks
    - learning to represent the world in a non task-specific way

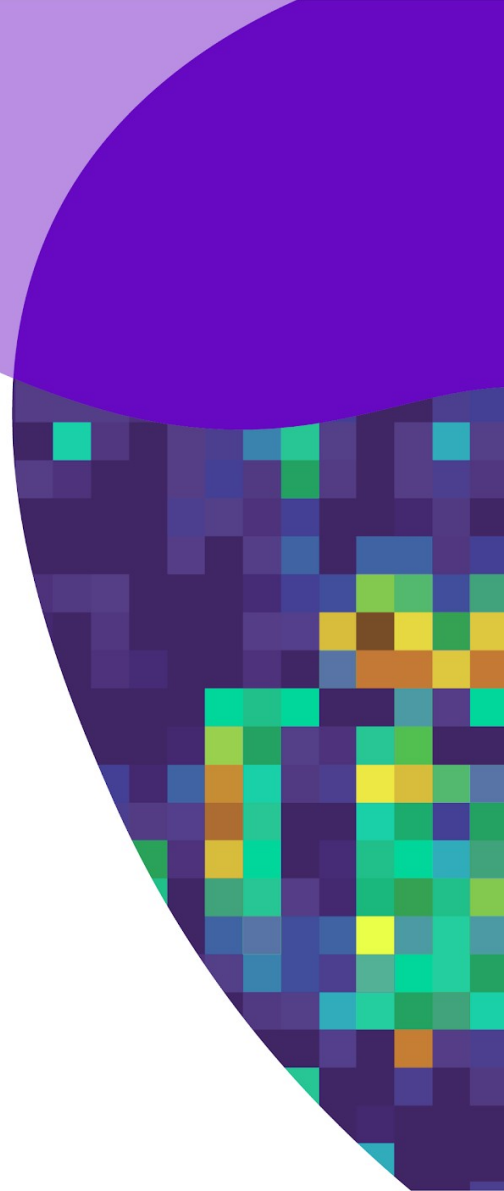
- **2. Learning to reason,** like Daniel Kahneman's "System 2"
  - Beyond feed-forward, System 1 subconscious computation.
  - Making reasoning compatible with learning.

- **3. Learning to plan complex action sequences**
  - Learning hierarchical representations of action plans

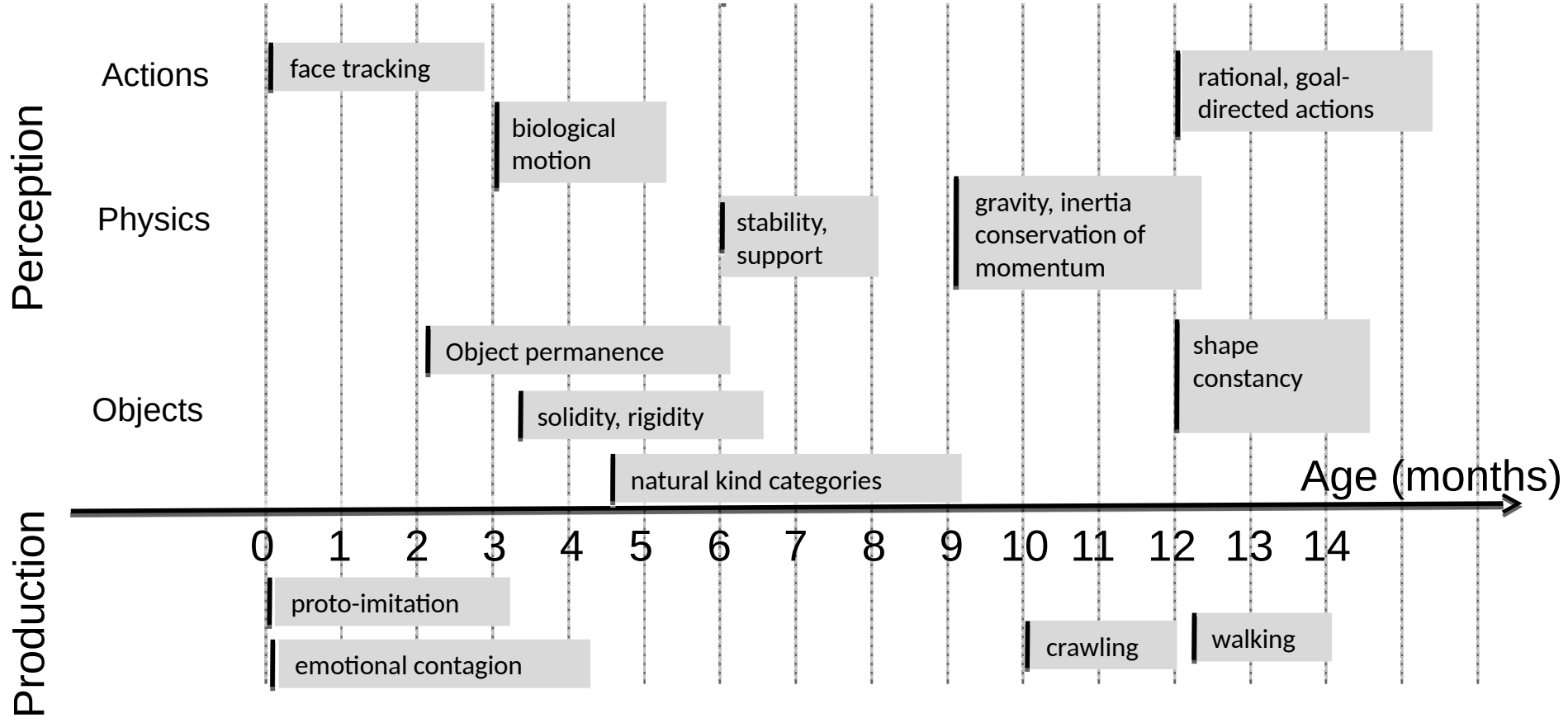
FACEBOOK AI

# How do humans and animals learn so quickly?

Not supervised.
Not Reinforced.



# When infants learn models of the world [after Emmanuel Dupoux]



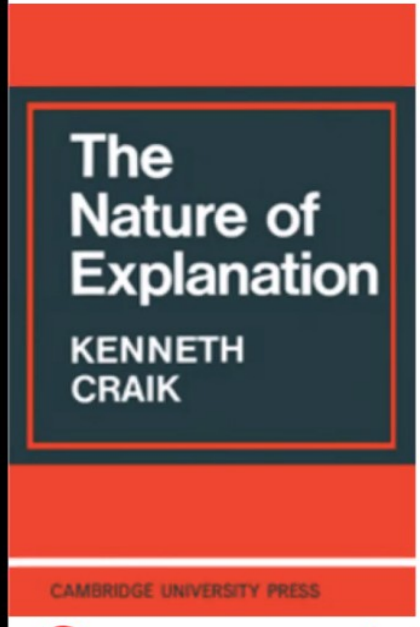
# How do Human and Animal Babies Learn?

- **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**

# Common sense is a collection of models of the world

## AI systems need to build "mental models"



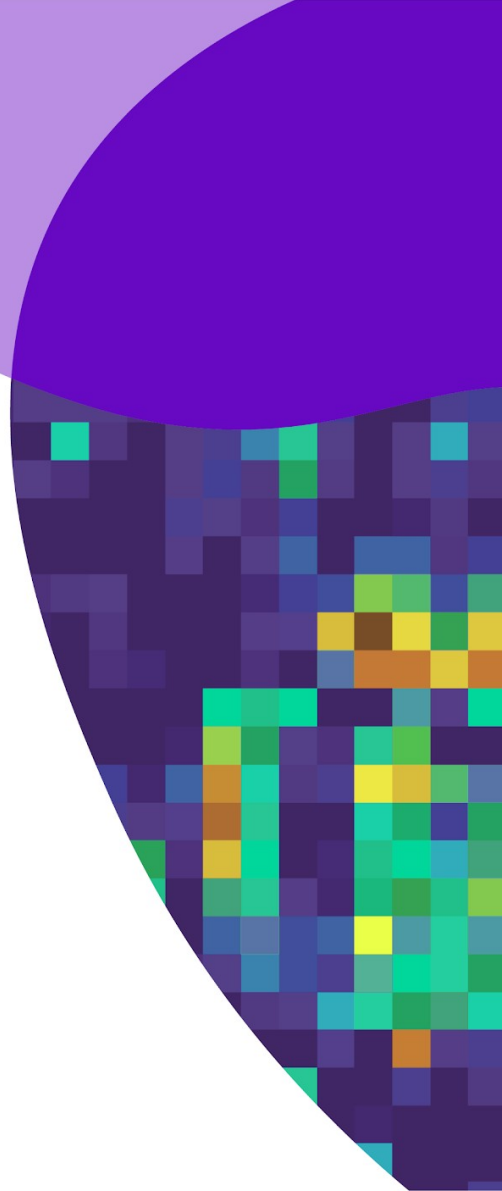

If the organism carries a `small-scale model' of external reality and of its own possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and the future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it (Craik, 1943,Ch. 5, p.61)

Commonsense is not just facts, it is a collection of models


Jitendra Malik

FACEBOOK AI

# Self-Supervised Learning

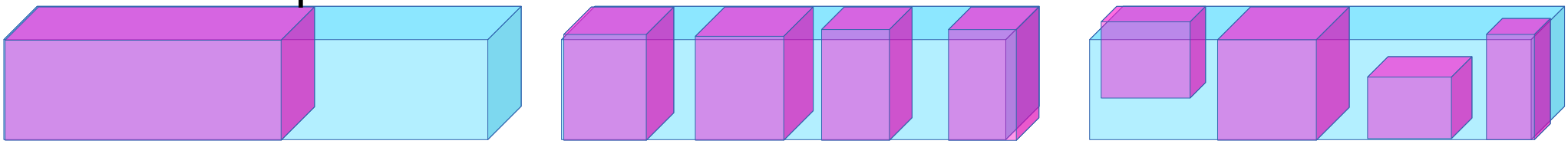


Capture dependencies.
Predict everything from everything else.

# Self-Supervised Learning = Learning to Fill in the Blanks

- **Reconstruct the input or Predict missing parts of the input.**

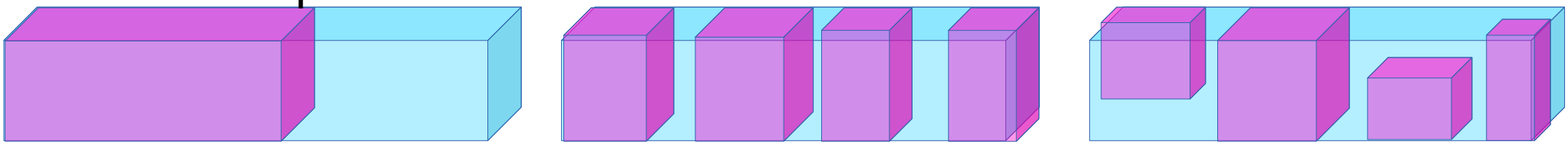
time or space →



# Self-Supervised Learning = Learning to Fill in the Blanks

- **Reconstruct the input or Predict missing parts of the input.**

time or space →



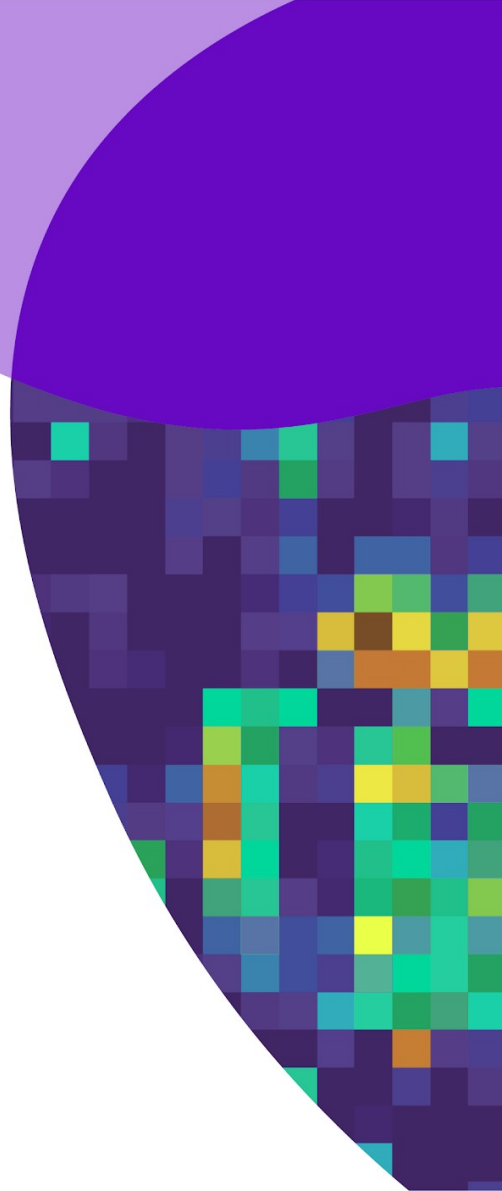
# Two Uses for Self-Supervised Learning

- **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?**

FACEBOOK AI

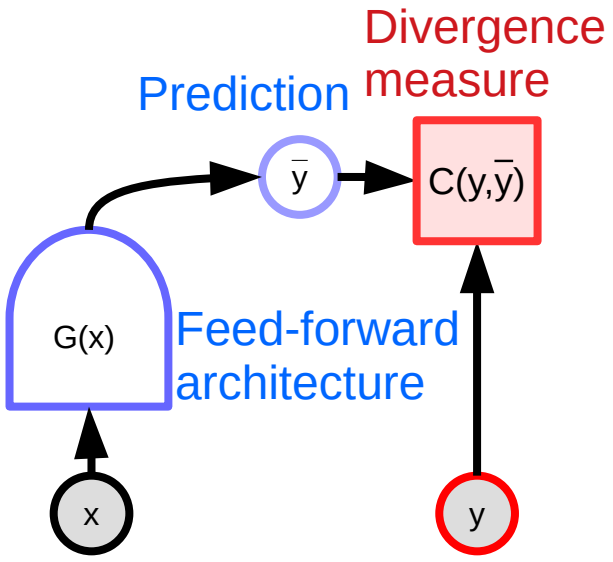
# Energy-Based Models

Capture dependencies through an energy function.

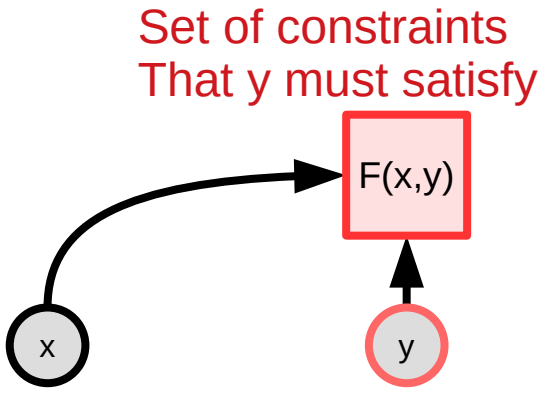


# Energy-Based Models

- **Feed-forward nets use a finite 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)

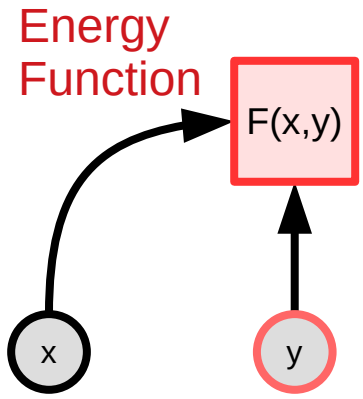



- **Inference through constraint satisfaction**
  - Finding an output that satisfies constraints: e.g a linguistically-correct translation or a transcription of speech into text.
  - Maximum likelihood inference in graphical models



# Energy-Based Models (EBM)

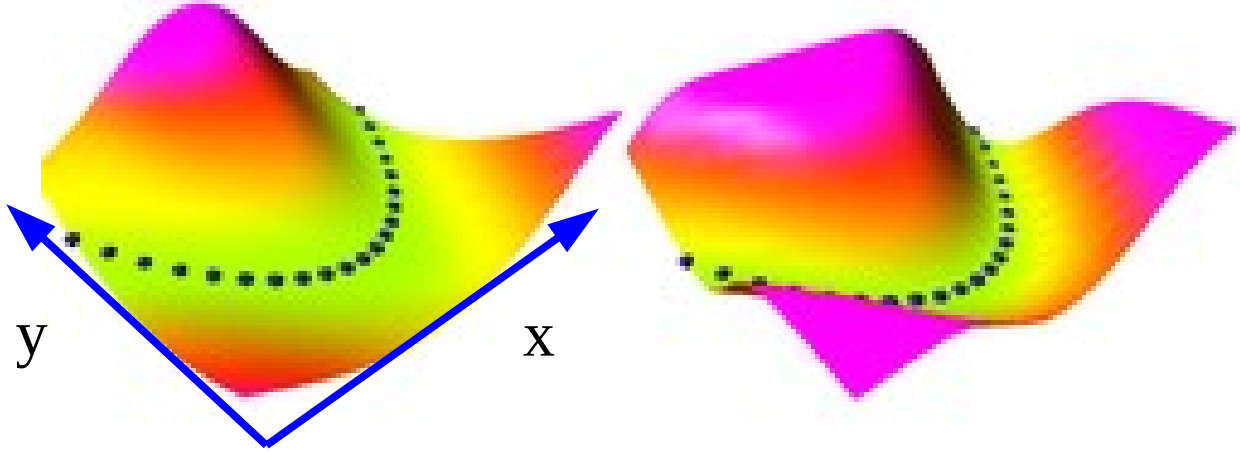
- **Energy function F(x,y) scalar-valued.**
  - Takes low values when y is compatible with x and higher values when y is less compatible with x
- **Inference: find values of y that make F(x,y) small.**
  - There may be multiple solutions




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

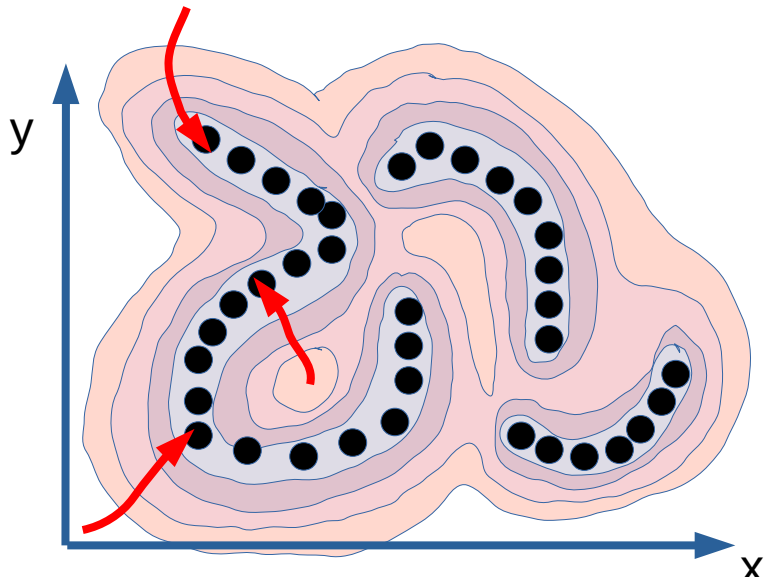
- **Note: the energy is used for inference, not for learning**

- **Example**
  - Blue dots are data points

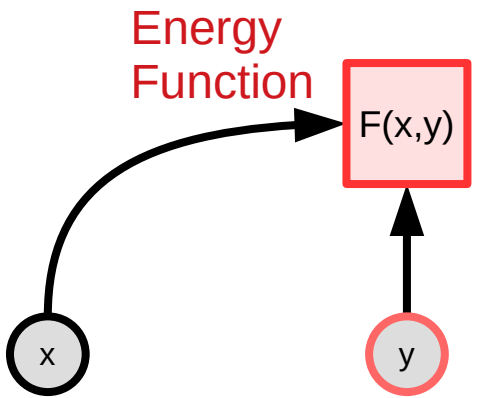


# Energy-Based Model: implicit function

- **Energy function that captures the x,y dependencies:**
  - Low energy near the data points. Higher energy everywhere else.
- **Inference**
  - Y discrete: exhaustive search, heuristic search, dynamic programming, belief propagation, simulated annealing, gradient-free optimization….
  - Y continuous/differentiable: gradient-based optimization….

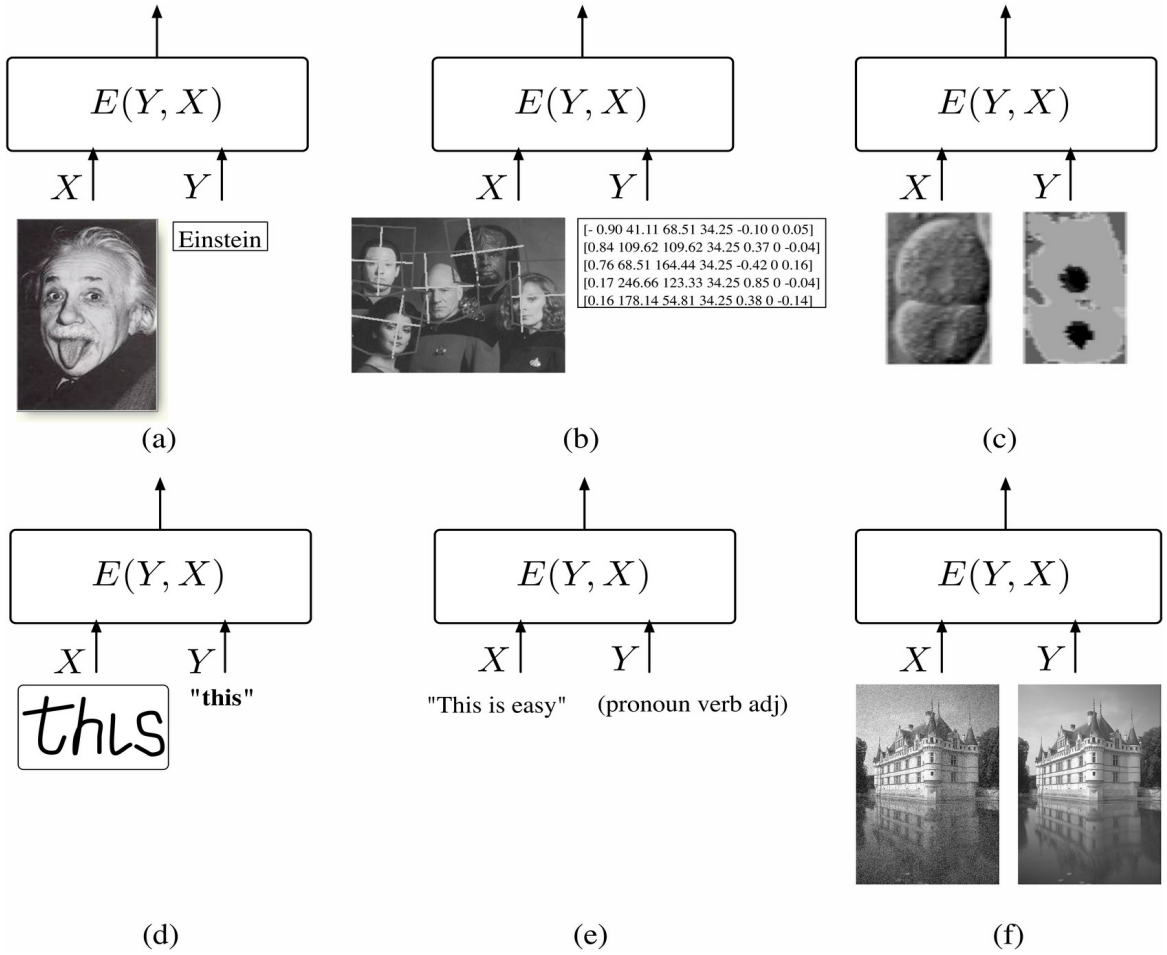



$$\check{y} = \text{argmin}_y F(x, y)$$



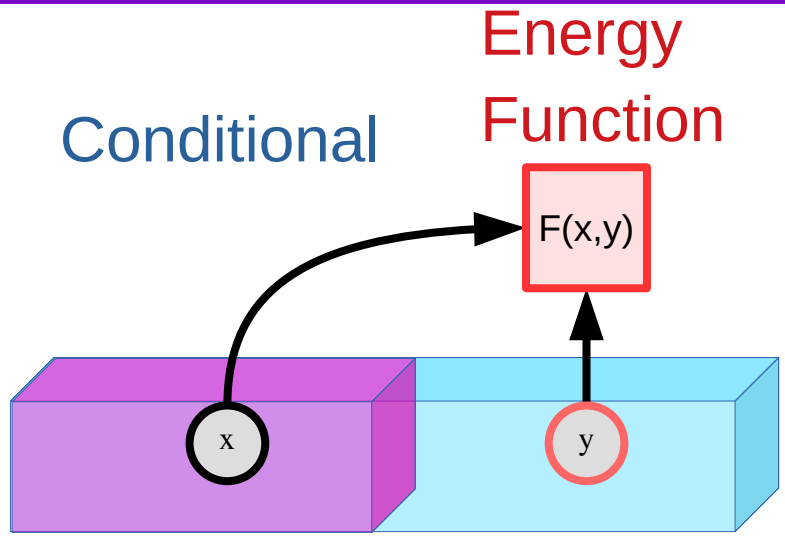
# When inference is hard

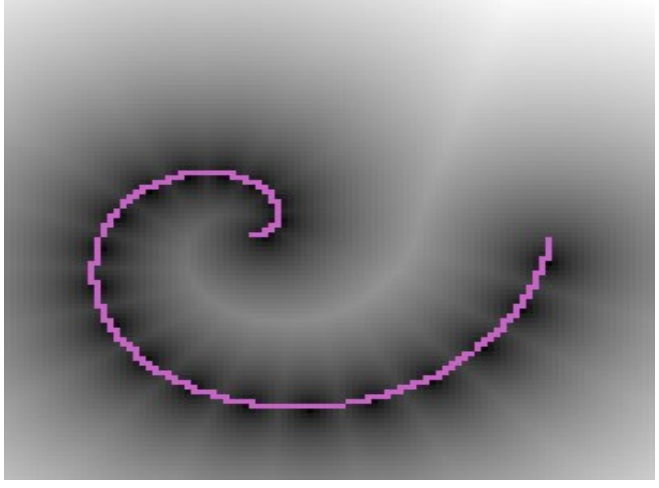
- **Cases where inference is hard:**
  - Output is a high-dimensional object with structure:
    - Sequence, image, video,…
  - Output has compositional structure:
    - Text, action sequence,…
  - Output results from a long chain of reasoning
    - That can be reduced to an optimization problem



# Conditional and Unconditional Energy-Based Models

- **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

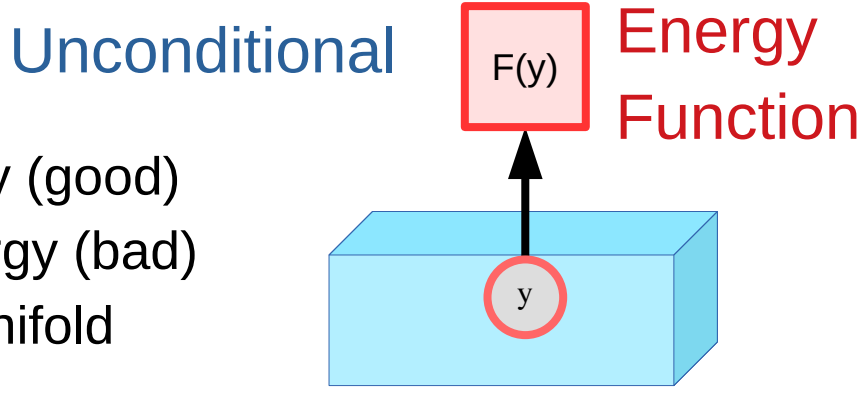



Dark = low energy (good)

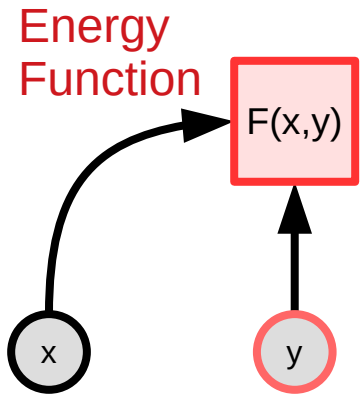
Bright = high energy (bad)

Purple = data manifold



# Energy-Based Models vs Probabilistic Models

- **Probabilistic models are a special case of EBM**
  - Energies are like un-normalized negative log probabilities
- **Why use EBM instead of probabilistic models?**
  - EBM gives more flexibility in the choice of the scoring function.
  - More flexibility in the choice of objective function for learning
- **From energy to probability: Gibbs-Boltzmann distribution**
  - Beta is a positive constant



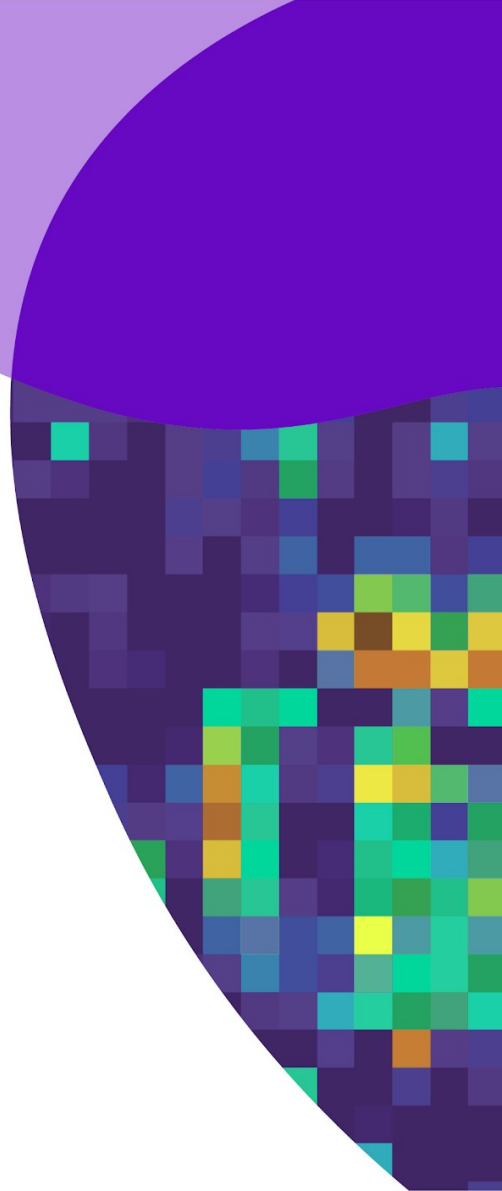

$$P(y|x) = \frac{e^{-\beta F(x,y)}}{\int_{y'} e^{-\beta F(x,y')}}$$

FACEBOOK AI

# EBM Architectures

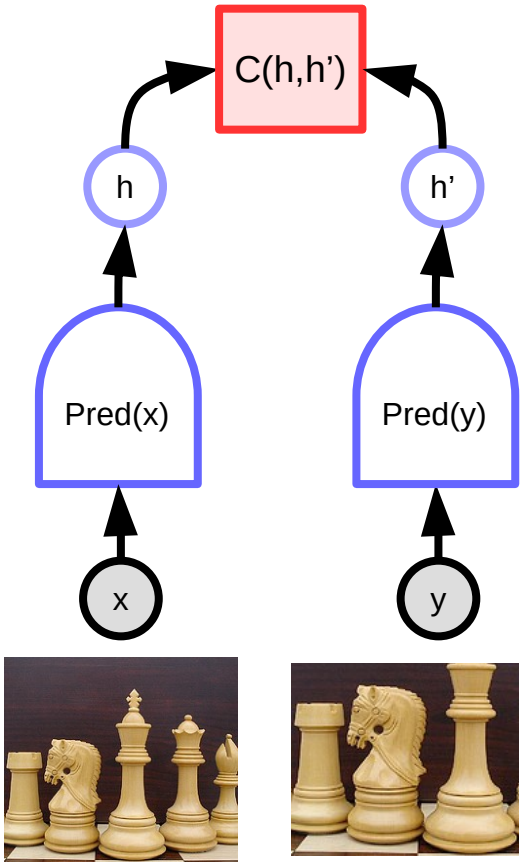
## for multimodal prediction

1. Joint embedding architectures
2. Latent variable models



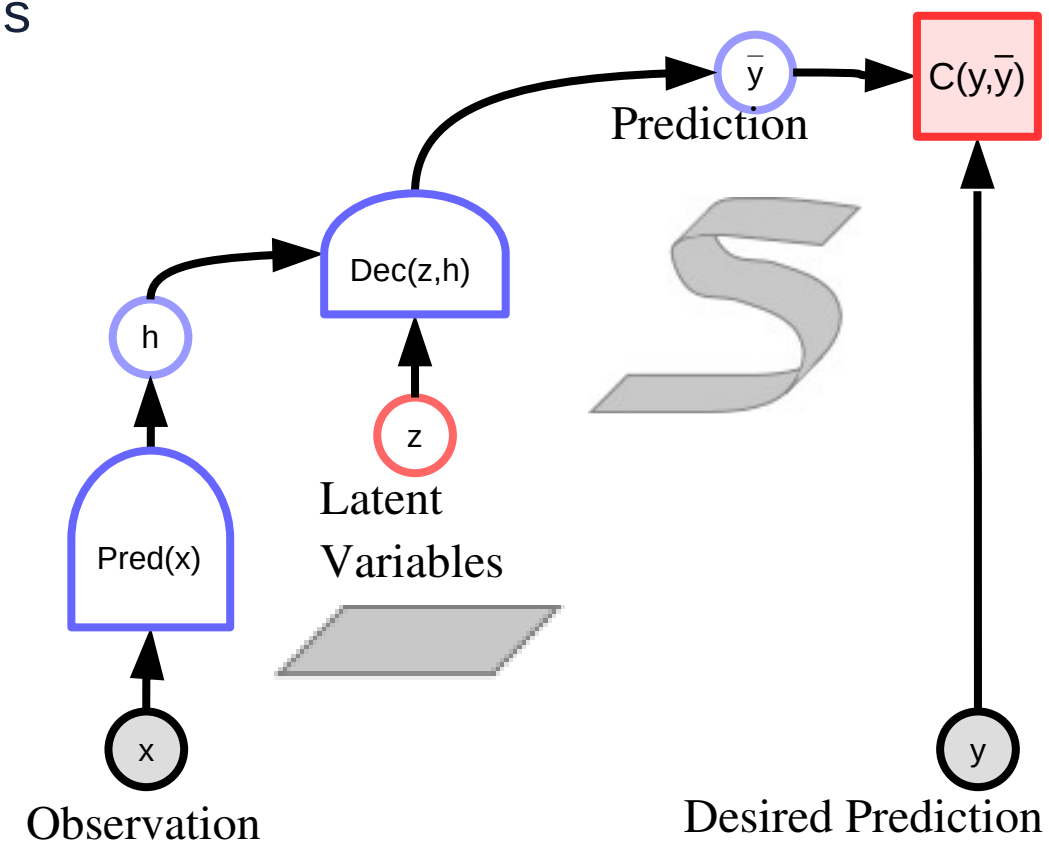
# Joint Embedding

- **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: <in a few slides>**
- **Many successful examples for image recognition:**
  - 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]
  - …..



# Latent-Variable Predictive/Generative EBM

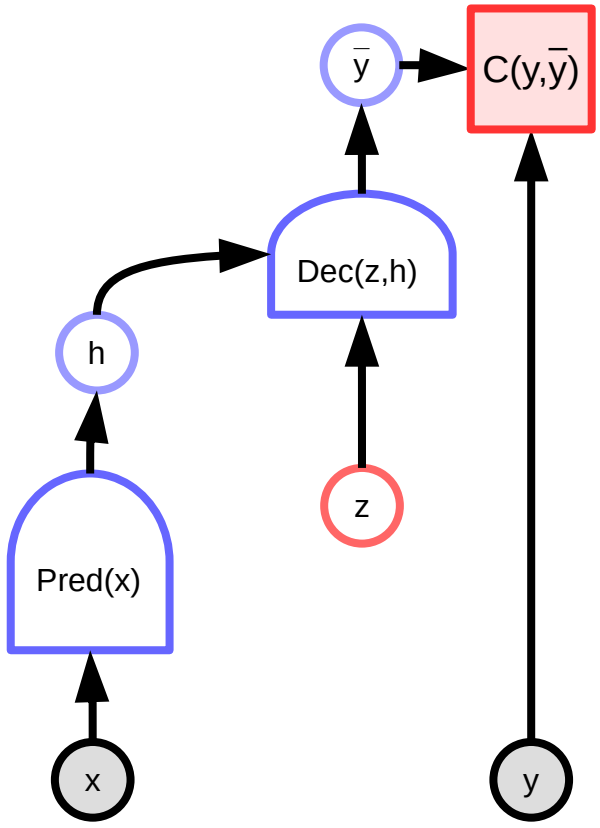
- **Latent variables:**
  - parameterize the set of predictions
- **Ideally, the latent variable represents independent explanatory factors of variation of the prediction.**
- **The information capacity of the latent variable must be minimized.**
  - Otherwise all the information for the prediction will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



# When inference involves latent variables

- **Latent variables are variables whose value is never given to us.**
  - Examples: to read a handwritten word, it helps to know where the characters are

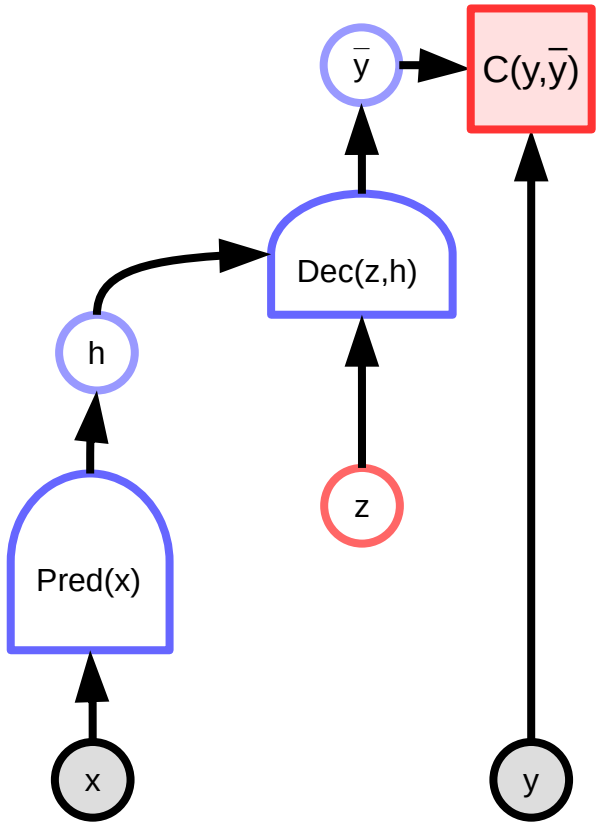
  - To recognize speech, it helps to know where the words and phonemes are
    - Youcantreadthisifyoudontunderstandenglish
    - Vousnepouvezpaslirececisivousneparlezpasfrançais



# When inference involves latent variables

- **Latent variables are variables whose value is never given to us.**
  - Examples: to read a handwritten word, it helps to know where the characters are

  - To recognize speech, it helps to know where the words and phonemes are
    - You can't read this if you don't understand english
    - Vous ne pouvez pas lire ceci si vous ne parlez pas français



# Latent-Variable EBM: inference

- **Simultaneous minimization with respect to y and z**

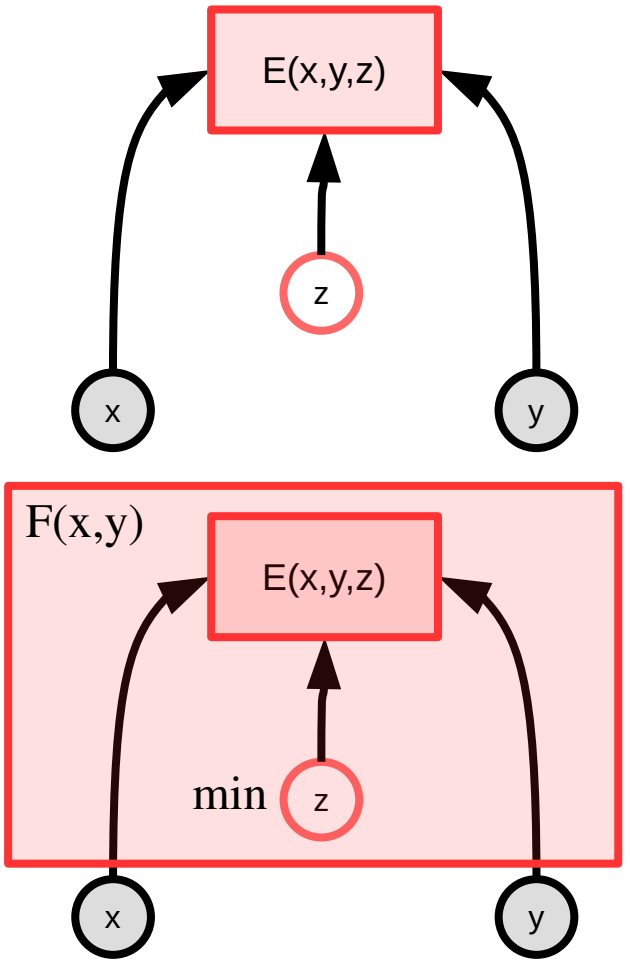
$$\check{y}, \check{z} = \text{argmin}_{y,z} E(x, y, z)$$

- **Redefinition of F(x,y)**

$$F_\infty(x, y) = \min_z E(x, y, z)$$

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

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



# Marginalizing over a latent variable

$$P(y|x) = \int_z P(y,z|x) \qquad P(y,z|x) = \frac{e^{-\beta E(x,y,z)}}{\int_y \int_z e^{-\beta E(x,y,z)}}$$

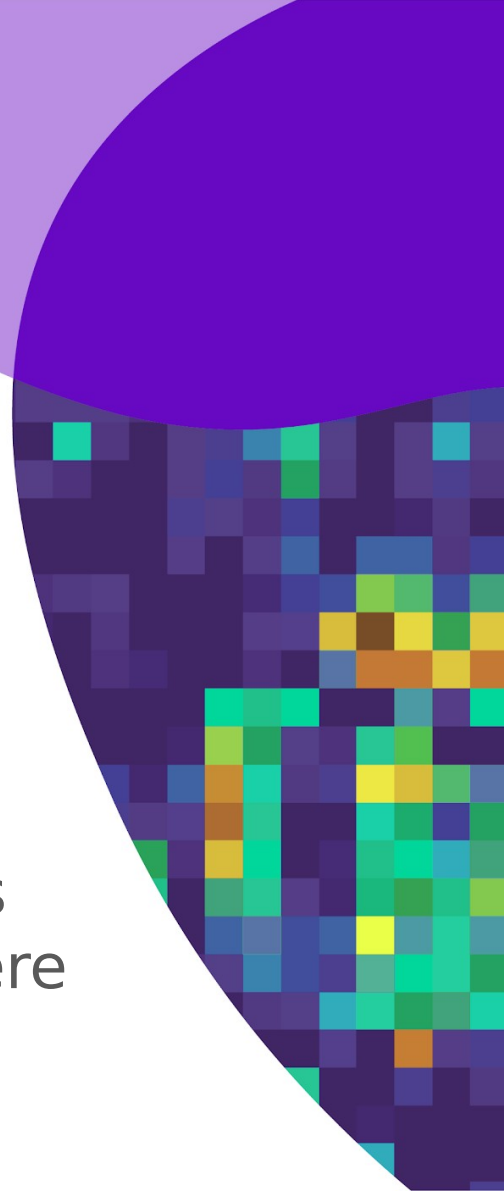
$$P(y|x) = \frac{\int_z e^{-\beta E(x,y,z)}}{\int_y \int_z e^{-\beta E(x,y,z)}} = \frac{e^{-\beta\left[-\frac{1}{\beta}\log \int_z e^{-\beta E(x,y,z)}\right]}}{\int_y e^{-\beta\left[-\frac{1}{\beta}\log \int_z e^{-\beta E(x,y,z)}\right]}} = \frac{e^{-\beta F_\beta(x,y)}}{\int_y e^{-\beta F_\beta(x,y)}}$$

- **With the following definition for the free energy F(x,y), we recover the usual Gibbs formula**

$$F_\beta(x,y) = -\frac{1}{\beta}\log \int_z e^{-\beta E(x,y,z)} \qquad P(y|x) = \frac{e^{-\beta F(x,y)}}{\int_{y'} e^{-\beta F(x,y')}}$$

FACEBOOK AI

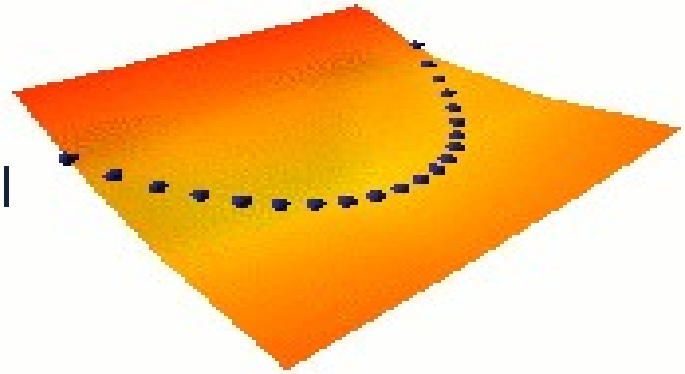
# Training EBMs

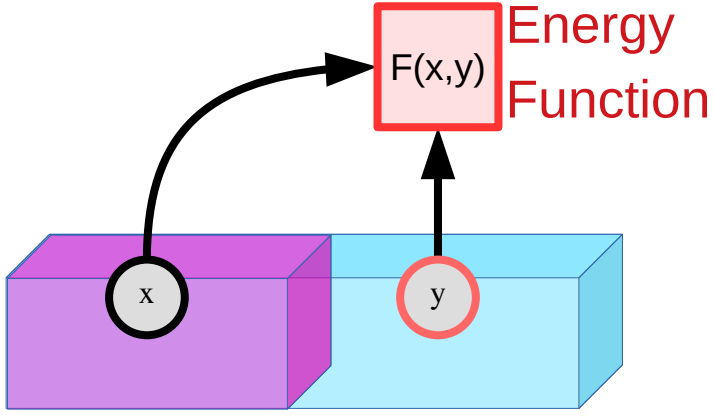


Push down on the energy of data points
Make sure the energy is higher elsewhere

# Training an Energy-Based Model

- **Parameterize F(x,y)**
- **Training sample: 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], ŷ)
  - **2. Regularized/Architectural Methods:** build F(x,y) so that the volume of low energy regions is limited or minimized through regularization

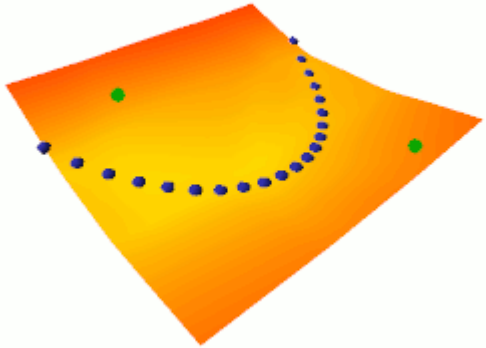




# Contrastive methods vs Regularized Methods

- **Contrastive methods: works with any architecture**
  - Expensive in high dimension
  - Example of contrastive loss: pick a ŷ to push up.

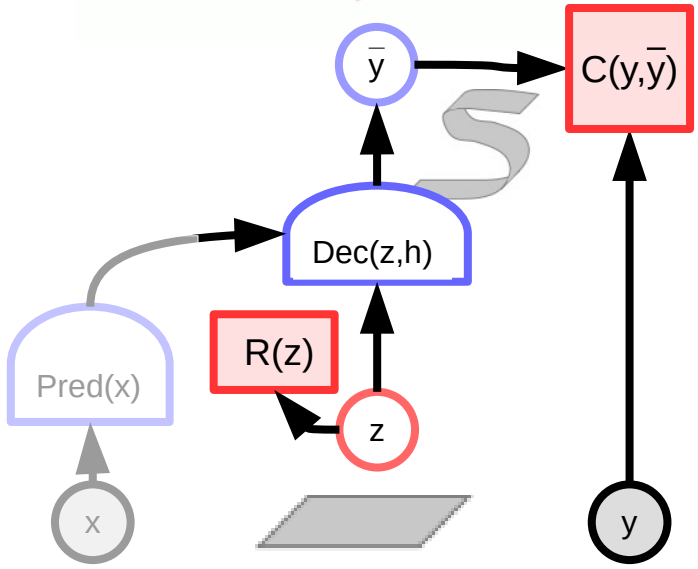
$$\mathcal{L}(x, y, \hat{y}, w) = [F_w(x, y) - F_w(x, \hat{y}) + m(y, \hat{y})]^+$$



- **Regularized methods: minimizing the volume of low-energy space**
  - E.g. by limiting the capacity of the latent

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

$$F_w(x, y) = \min_z [C(\mathrm{Dec}(\mathrm{Pred}(x), z), y) + R(z)]$$



# Contrastive Methods vs Regularized/Architectural Methods

- **Contrastive: [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 upper-bounded: PCA, K-means, Gaussian Mixture Model, Square ICA, normalizing flows…
  - A2: use a regularization term that minimizes the volume of space that has low energy: Sparse coding, sparse auto-encoder, LISTA, Discretized AE/VQVAE, Contracting AE, Saturating AE, Noisy AE, Variational AE, SwAV, BYOL, Barlow Twins.
  - A4: minimize the gradient and maximize the curvature around data points: score matching

# Max Likelihood is a (bad) Contrastive Method

- **Push down on data points,**
- **Push up on all points**
- **Max likelihood / probabilistic models**

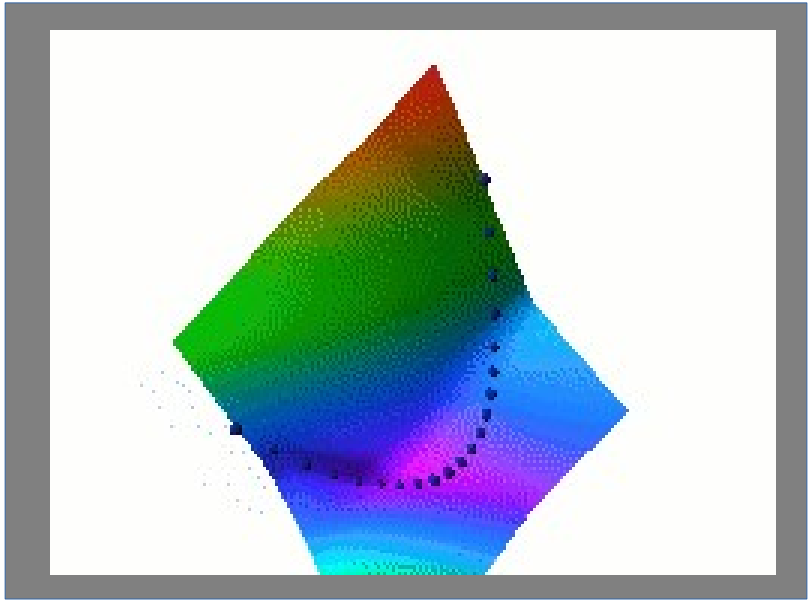
$$P_w(y|x) = \frac{e^{-\beta F_w(x,y)}}{\int_{y'} e^{-\beta F_w(x,y')}}$$

- Loss: $$\mathcal{L}(x,y,w) = F_w(x,y) + \frac{1}{\beta}\log\int_{y'} e^{-\beta F_w(x,y')}$$
- Gradient: $$\frac{\partial \mathcal{L}(x,y,w)}{\partial w} = \frac{\partial F_w(x,y)}{\partial w} - \int_{y'} P_w(y'|x)\frac{\partial F_w(x,y')}{\partial w}$$
- 2nd term is intractable: MC/MCMC/HMC/CD: $\hat{y}$ sampled from $P_w(y|x)$

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

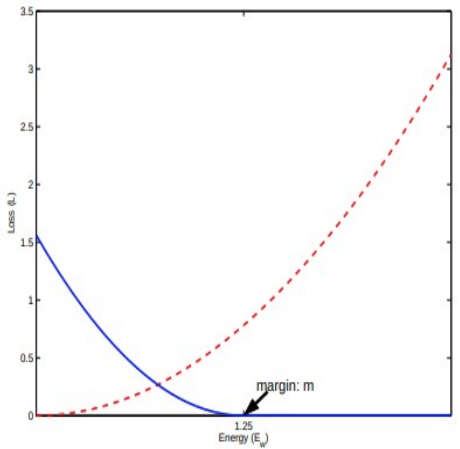
# Problem with Max Likelihood / Probabilistic Methods

- **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 Wassertstein GAN.
  - So that gradient-based inference works
  - Equivalent to a prior
  - But then, why use a probabilistic model?



# Contrastive Methods: many possible losses

- **Push down on data points, push up on 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]:
      $\mathcal{L}(x, y, \hat{y}, w) = \left([F_w(x, y)]^+\right)^2 + \left([m(y, \hat{y}) - F_w(x, \hat{y})]^+\right)^2$



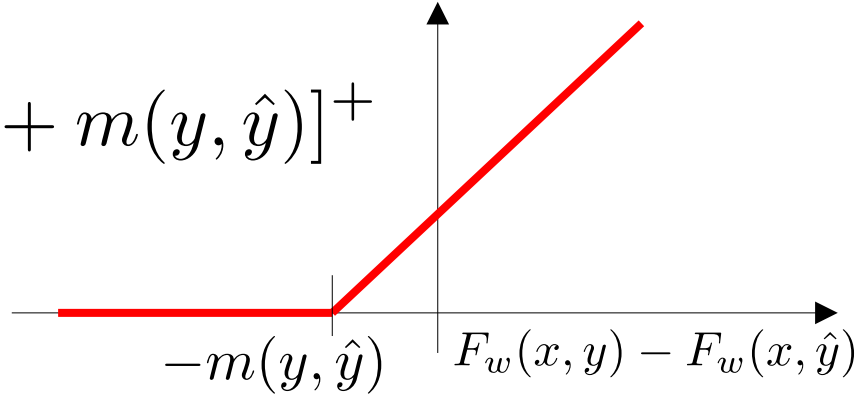
# General margin loss

- **Considers all possible outputs (or a well-chosen subset)**

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

- **Hinge loss that makes F(x,y) lower than F(x,y') by a quantity (margin) that depends on the distance between y and y'**
- **Example:**

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



# Plenty of Contrastive Loss Functions to Choose From

**Good and bad loss functions: good ones have non-zero margin**

| Loss | Formula | Margin |
|---|---|---|
| energy loss | $F(x,y)$ | 0 |
| perceptron | $F(x,y) - \min_{\check{y}\in} F(x,\check{y})$ | 0 |
| hinge | $\max(0, m + F(x,y) - F(x,\hat{y}))$ | $m$ |
| log | $\log\left(1 + e^{F(x,y)-F(x,\hat{y})}\right)$ | $\infty$ |
| LVQ2 | $\min(M, \max(0, F(x,y) - F(x,\hat{y}))$ | 0 |
| MCE | $\left(1 + e^{-(F(x,y)-F(x,\hat{y}))}\right)^{-1}$ | $\infty$ |
| square-square | $F(x,y)^2 - (\max(0, m(y,\hat{y}) - F(x,\hat{y})))^2$ | $m$ |
| square-exp | $F(x,y)^2 + \beta e^{-F(x,\hat{y})}$ | $\infty$ |
| NLL/MMI | $F(x,y) + \frac{1}{\beta}\log\int_{\hat{y}\in} e^{-\beta E(x,\hat{y})}$ | $\infty$ |
| MEE | $1 - e^{-\beta F(x,y)} / \int_{\hat{y}\in} e^{-\beta F(x,\hat{y})}$ | $\infty$ |

# 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:

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

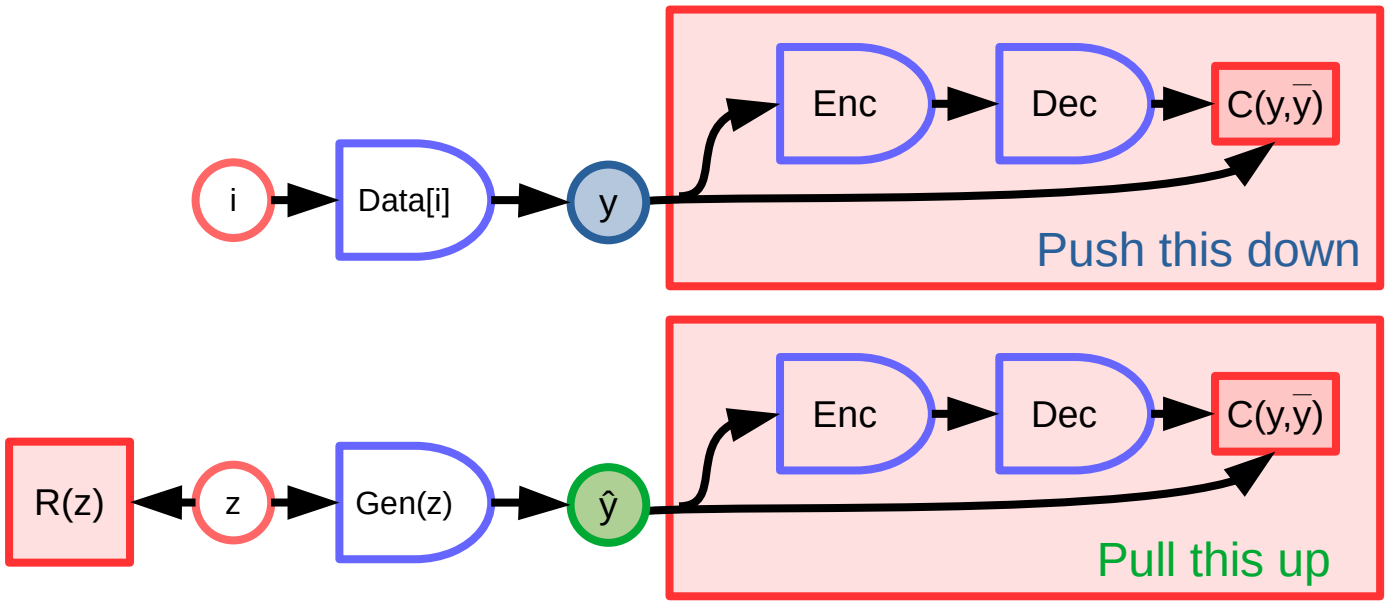
    - Where H must be an increasing fn of the data energies and decreasing fn of the contrastive point energies within the margin.
    - M is a margin matrix for all pairs of y and ŷ in the group.
  - Example: Neighborhood Component Analysis, Noise Contrastive Estimation, InfoNCE (implicit infinite margin)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

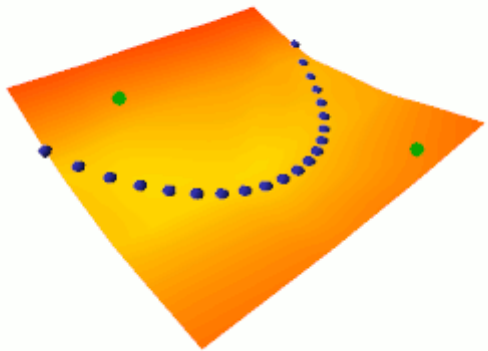
$$\mathcal{L}(x, y, \hat{y}_1, \dots \hat{y}_{p^-}, w) = -\log \frac{e^{-F_w(x,y)}}{e^{-F_w(x,y)} + \sum_{i=1}^{p^-} e^{-F_w(x,\hat{y}_i,w)}}$$

# GANs are secretly a contrastive method for EBM

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

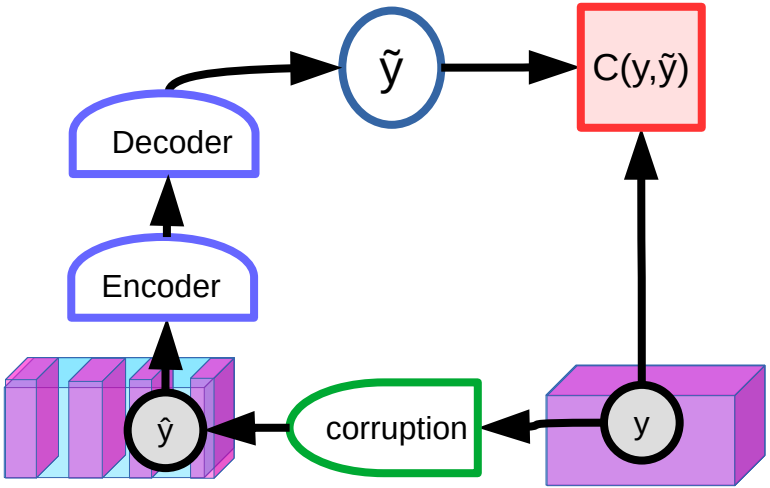
$$\mathcal{L}(x, y, \hat{y}, w) = H(F_w(x, y), F_w(x, \hat{y}), m(y, \hat{y}))$$



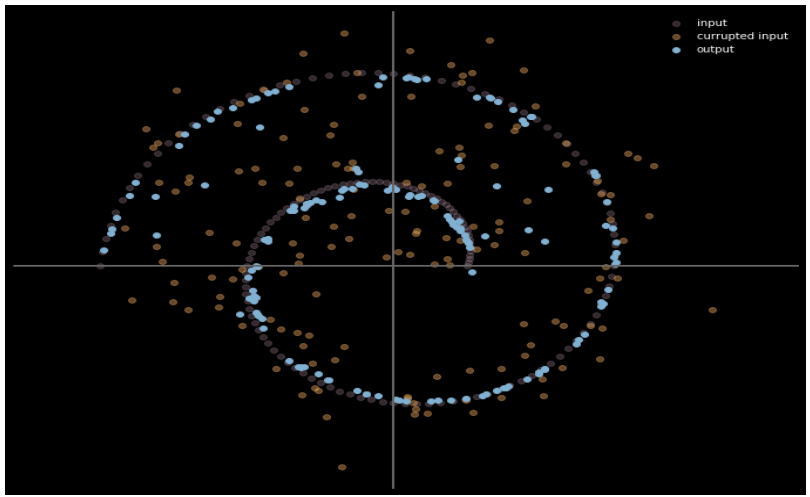
# Contrastive Methods in NLP / Denoising AE / Masked AE

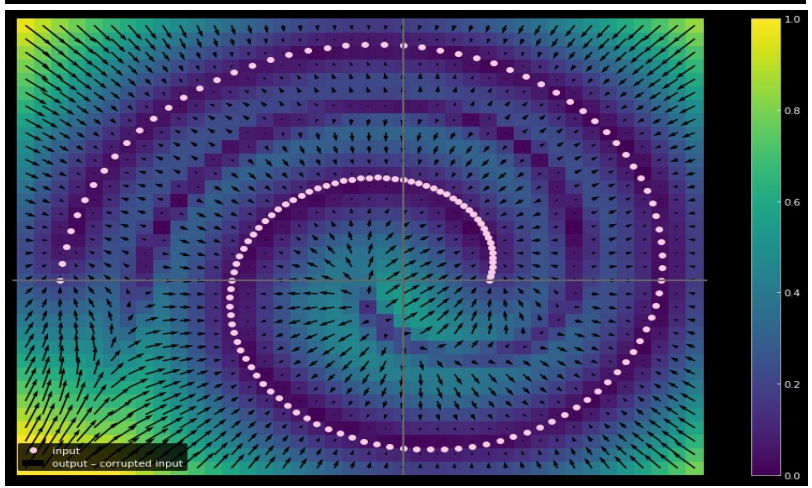
- **Denoising AE** [Vincent 2008]
- **Contrastive method for NLP**
  - [Collobert-Weston 2011]
- **Masked AE: Learning text representations**
  - BERT [Devlin 2018], RoBERTa [Ott 2019]




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

# Supervised Symbol Manipulation

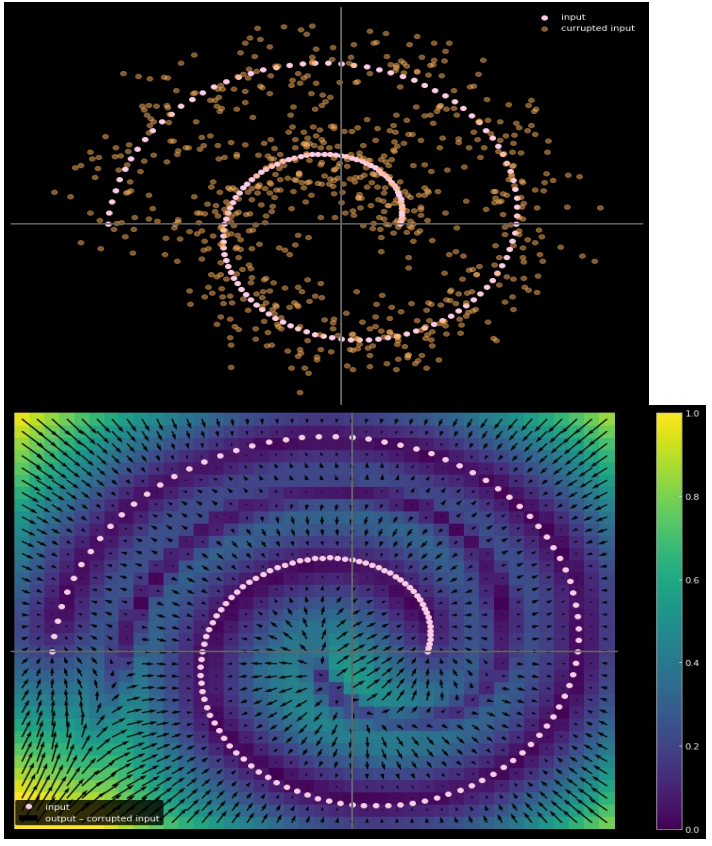
- **Solving integrals and differential equations symbolically with a transformer architecture**
  - [Lample & Charton arXiv:1912.01412]
  - Accuracy on various problems →

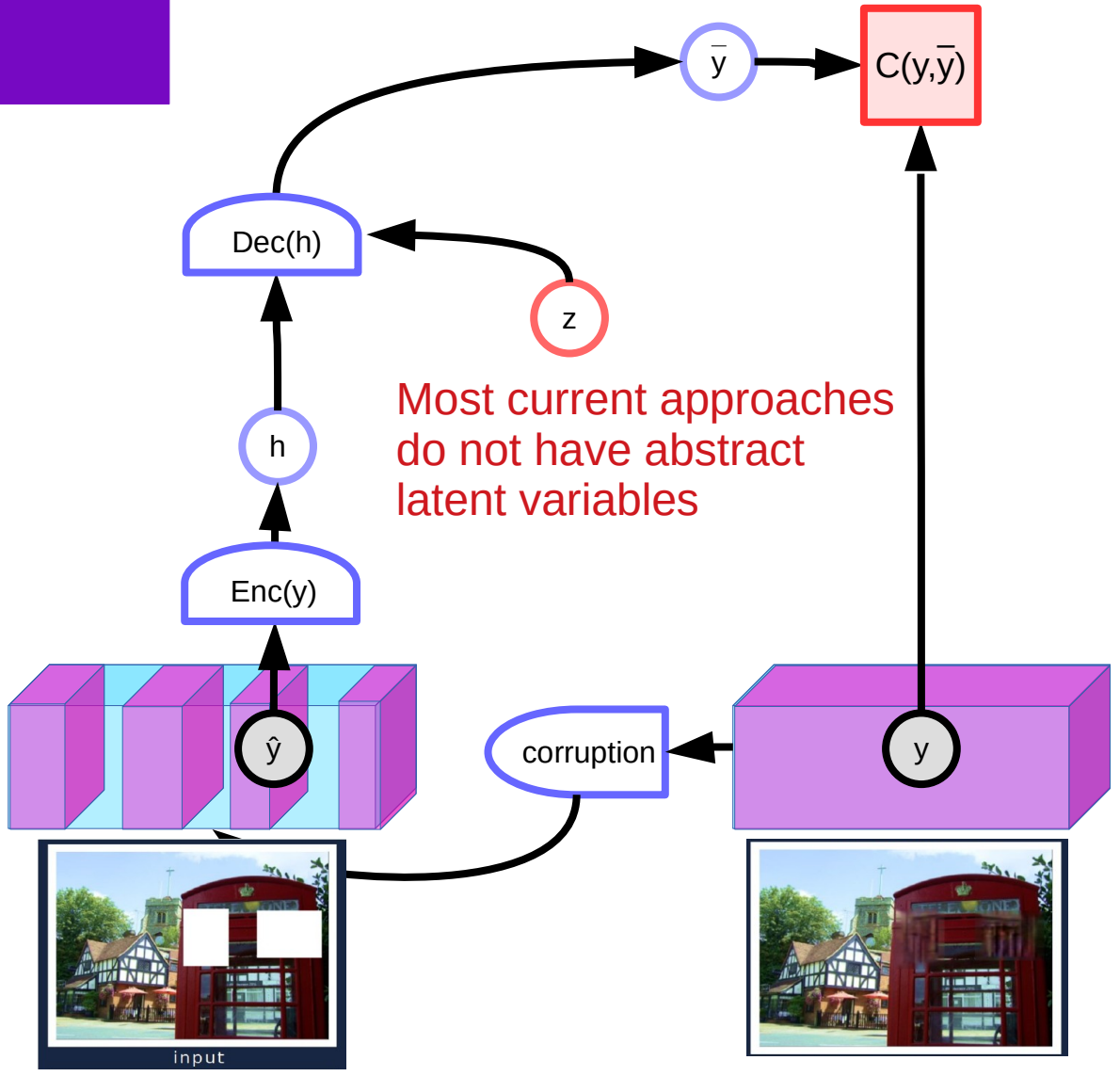
| | Integration (BWD) | ODE (order 1) | ODE (order 2) |
|---|---|---|---|
| Mathematica (30s) | 84.0 | 77.2 | 61.6 |
| Matlab | 65.2 | - | - |
| Maple | 67.4 | - | - |
| Beam size 1 | 98.4 | 81.2 | 40.8 |
| Beam size 10 | 99.6 | 94.0 | 73.2 |
| Beam size 50 | 99.6 | 97.0 | 81.0 |

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

# Denoising AE in continuous domains?

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



# Reconstructing Images is Hard

- **Representing uncertainty in the prediction**
  - Easy for discrete object, hard for continuous ones
- **Predicting missing words is easy**
  - Words are in a discrete set
  - We can represent distributions over discrete sets (softmax)
- **Predicting missing images is hard**
  - Images are high-dimensional and continuous
  - We don’t have good ways of representing distributions over images
- **Solution: no reconstruction, joint embedding**

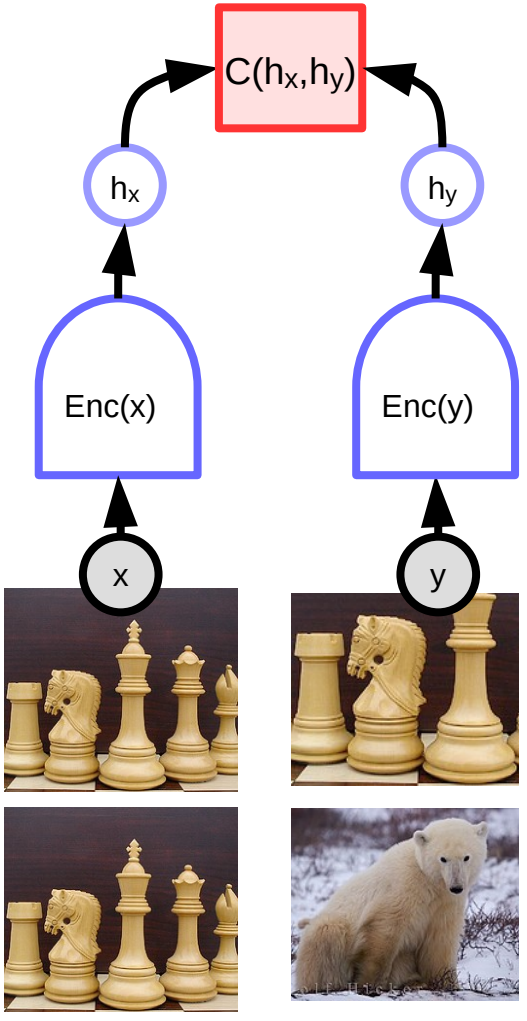
# Contrastive Joint Embedding

$$F(x, y) = C(\text{Enc}(x), \text{Enc}(y))$$

- **Siamese nets, metric learning**
- **Two identical networks with shared weights**
  - Signature verification: [Bromley NIPS'93],
  - Face verification [Chopra CVPR'05]
  - Face reco, DeepFace [Taigman et al. CVPR 2014]
  - Video feature learning [Taylor CVPR 2011]
  - Use square-square or square-exp loss

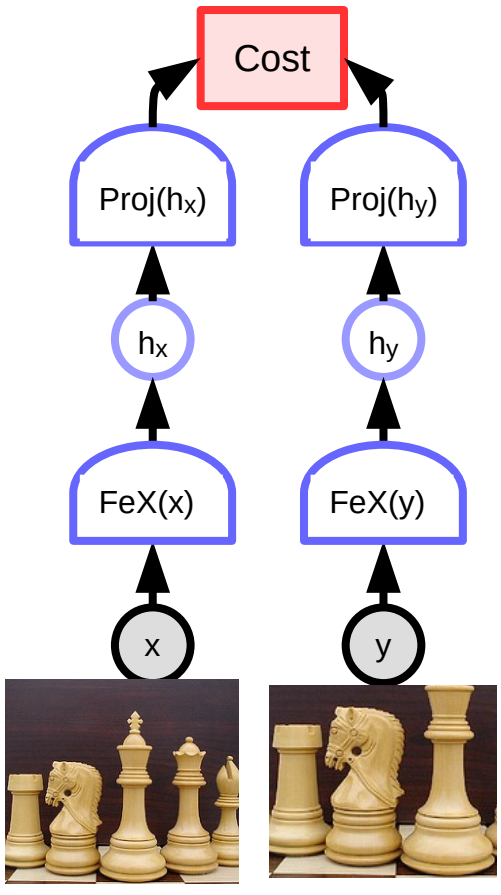
$$\mathcal{L}(x, y, \hat{y}, w) = \left([F_w(x, y)]^+\right)^2 + \left([m - F_w(x, \hat{y})]^+\right)^2$$

- **Advantages:**
  - no pixel-level reconstruction
  - Learns a similarity metric
  - Multimodality through encoder invariance

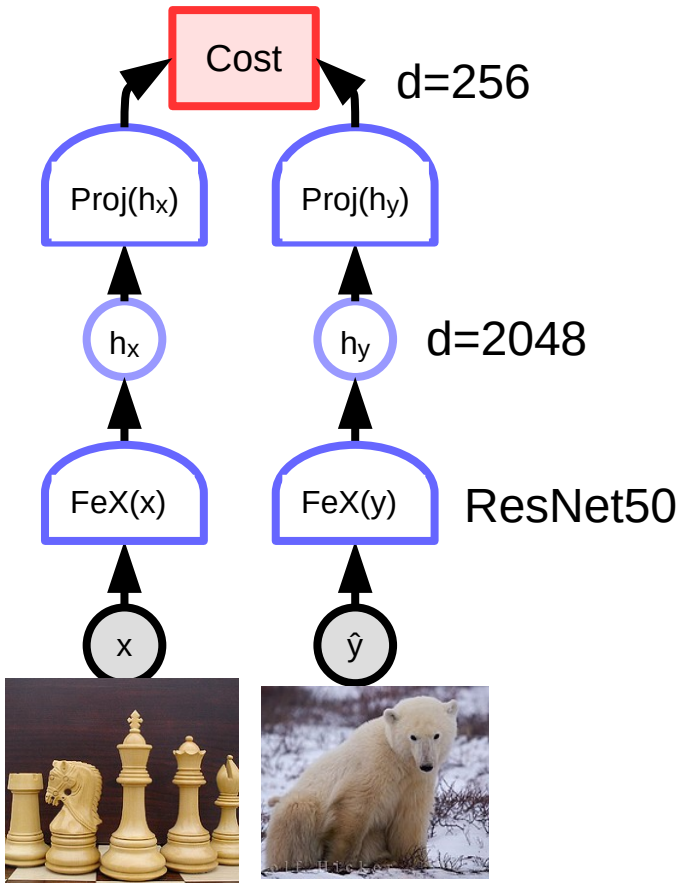


# Contrastive Joint Embedding

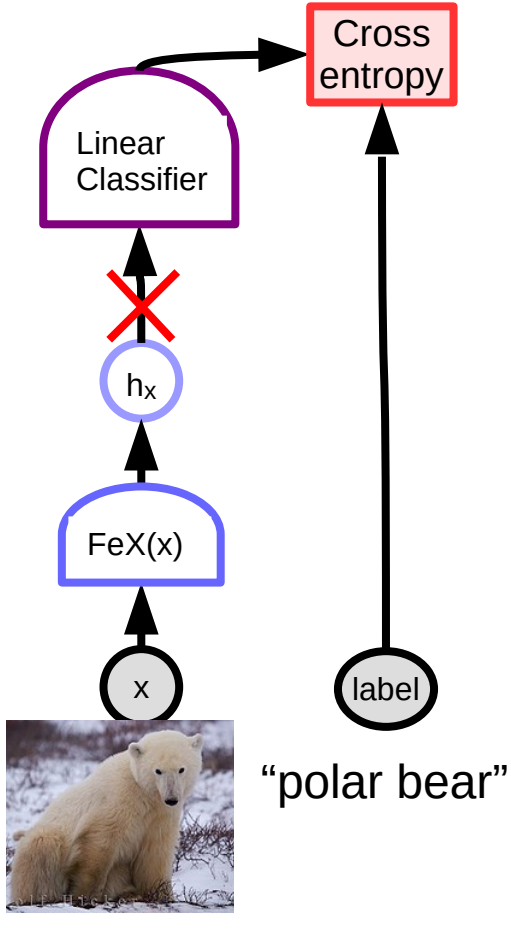
Make F(x,y) small




Make F(x,ŷ) large

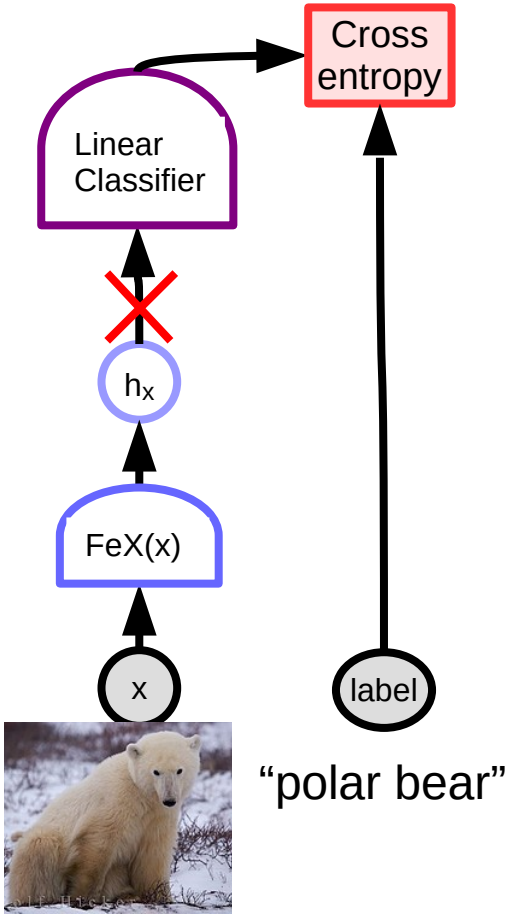



Training a supervised linear head



# Contrastive Joint Embedding

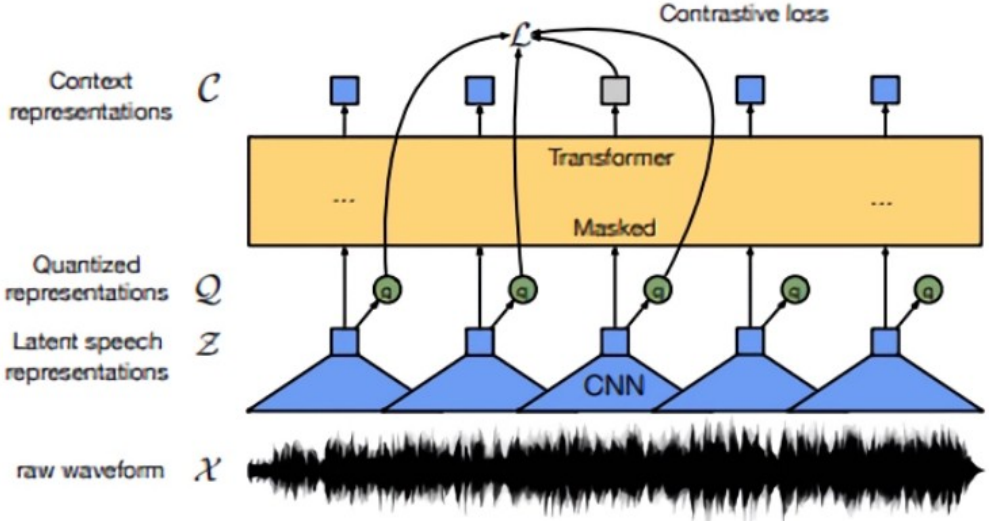
- **Issues:**
  - Hard negative mining
  - Expensive computationally
  - Only works for small dimension of embeddings (256)
- **Successful examples for image recognition:**
  - PIRL [Misra et al. Arxiv:1912.01991]
  - MoCo [He et al. Arxiv:1911.05722]
  - SimCLR [Chen et al. Arxiv:2002.05709]
- **Use InfoNCE group loss**

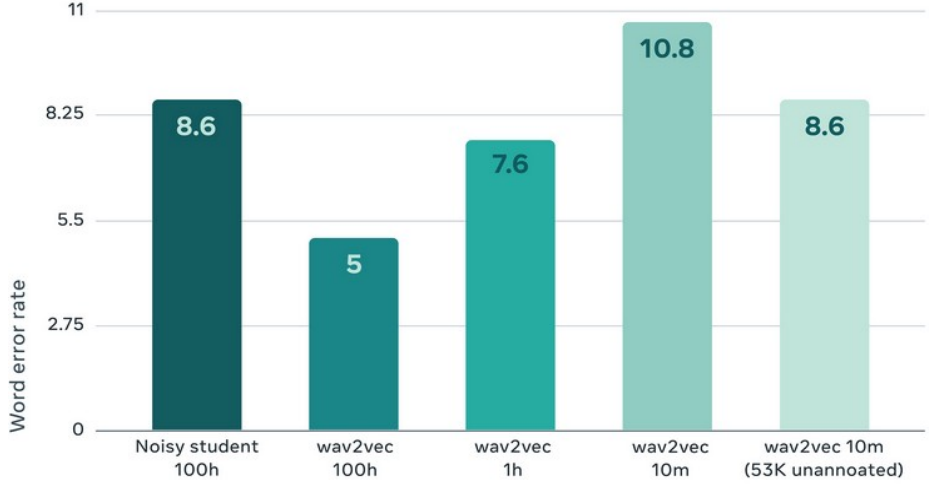



$$\mathcal{L}(x, y, \hat{y_1}, \ldots \hat{y}_q, w) = F_w(x, y) + \log \left[ e^{-F_w(x,y)} + \sum_{i=1}^{q} e^{-F_w(x,\hat{y}_i,w)} \right]$$

# Wav2Vec 2.0: SSL for speech recognition

- **Pre-train on 960h of unlabeled speech,**
- **then train with 10 minutes, 1h or 100h of labeled speech**
- **Results on LibriSpeech**
  - Wav2vec on 10 minutes = Same WER as previous SOTA on 100h
  - Papers: [Baevski et al. NeurIPS 2020] [Xu et al. ArXiv:2010.11430]
  - Code: Github: PyTorch/fairseq

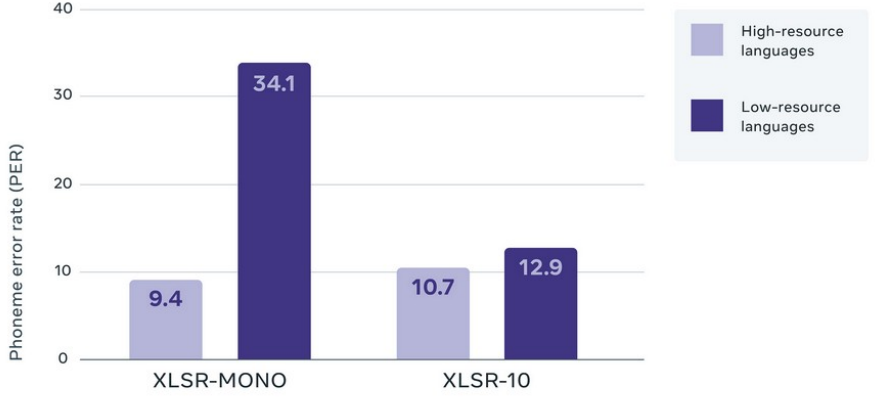


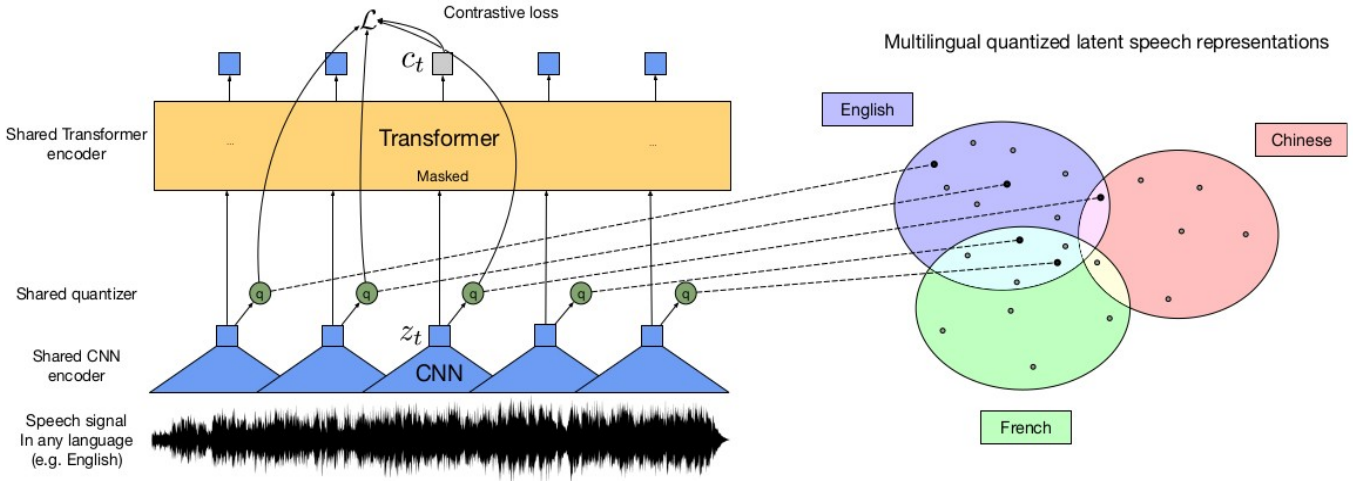

WER for Noisy Student self-training with 100 hours of labeled data. Wav2vec 2.0 with 100 hours, 1 hour, and only 10 minutes of labeled data. All models use the remainder of the LibriSpeech corpus (total 960 hours) as unannotated data, except for the last result, which uses 53K hours from LibriVox.

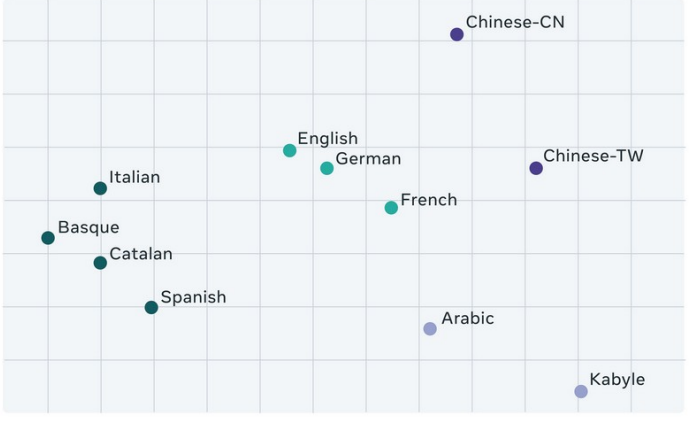
# XLSR: multilingual speech recognition

- **Multilingual self-supervised ASR**
  - [Conneau arXiv:2006.13979]
  - Raw audio → ConvNet → Transformer
  - CommonVoice: 72% reduction of PER
  - BABEL: 16% reduction of WER




Results on the Common Voice benchmark in terms of phoneme error rate (PER), comparing training on each language individually (XLSR-Mono) with training on all 10 languages simultaneously (XLSR-10).

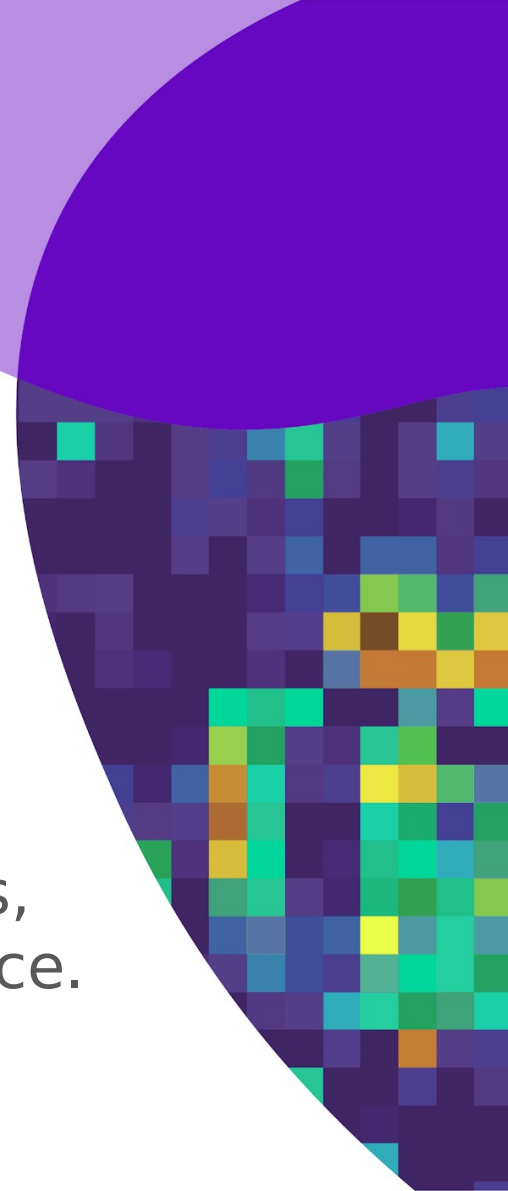



Visualization of how the learned units are used across languages. Graph shows a 2D PCA plot of how units are used in each language. Languages closer to each other, like English and German or Basque and Catalan, tend to use similar units.

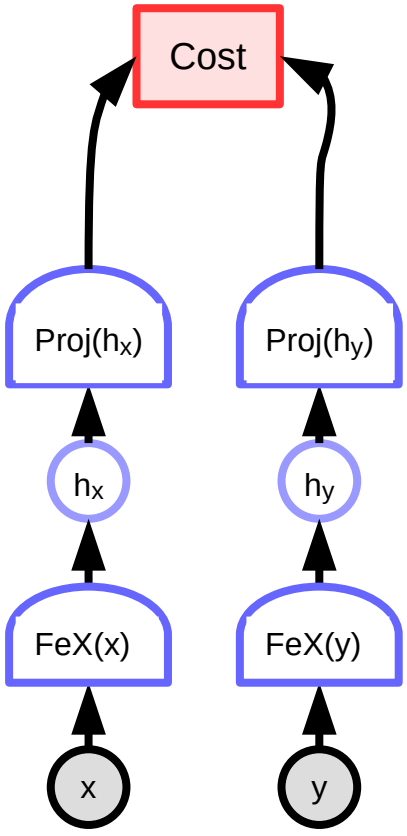
FACEBOOK AI

# Non-Contrastive EBM Training

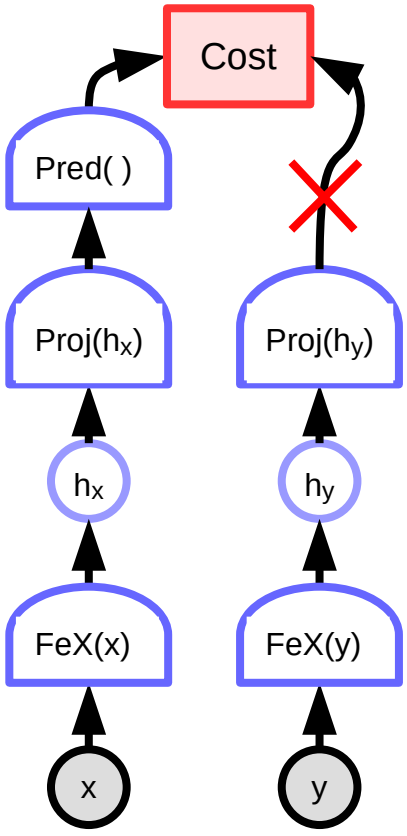


Push down on the energy of data points,
Minimize the volume of low-energy space.

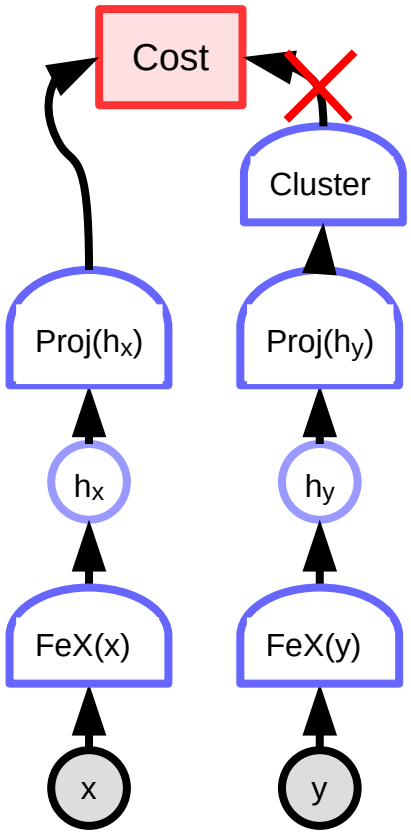
# Joint Embedding Architectures & Methods to prevent collapse.



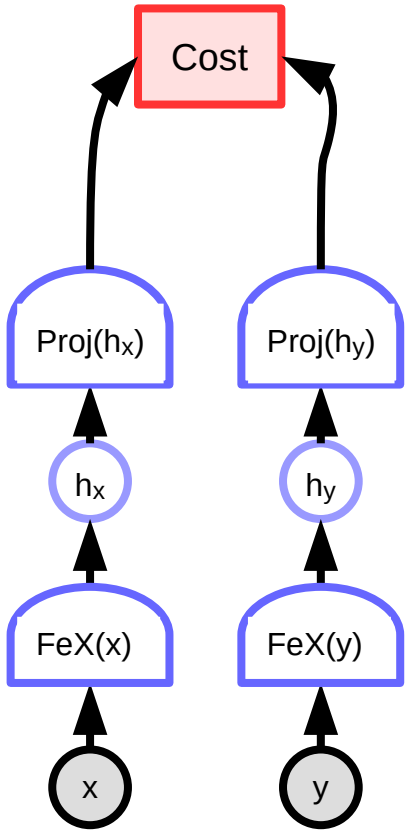

**Contrastive**
DrLIM, PIRL,
MoCo,SimCLR,




**Distillation**
BYOL, SimSiam



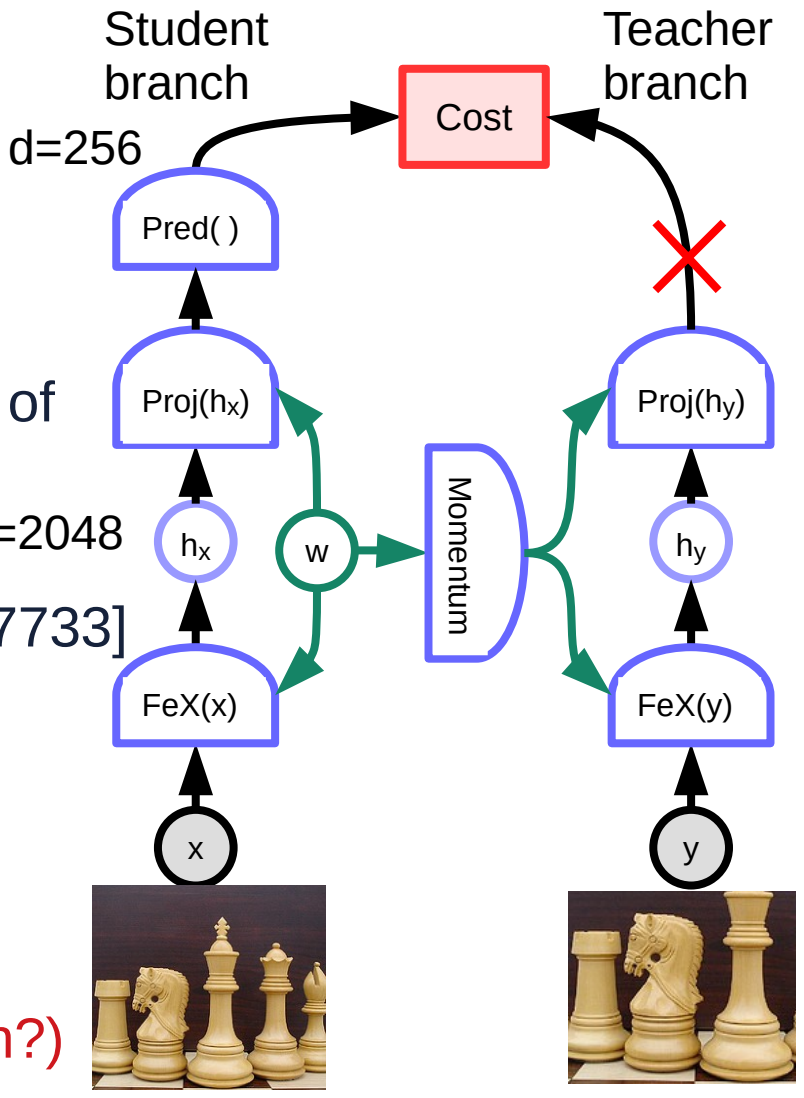

**Quantization**
DeepCluster, SwAV




**Information Max**
Barlow Twins

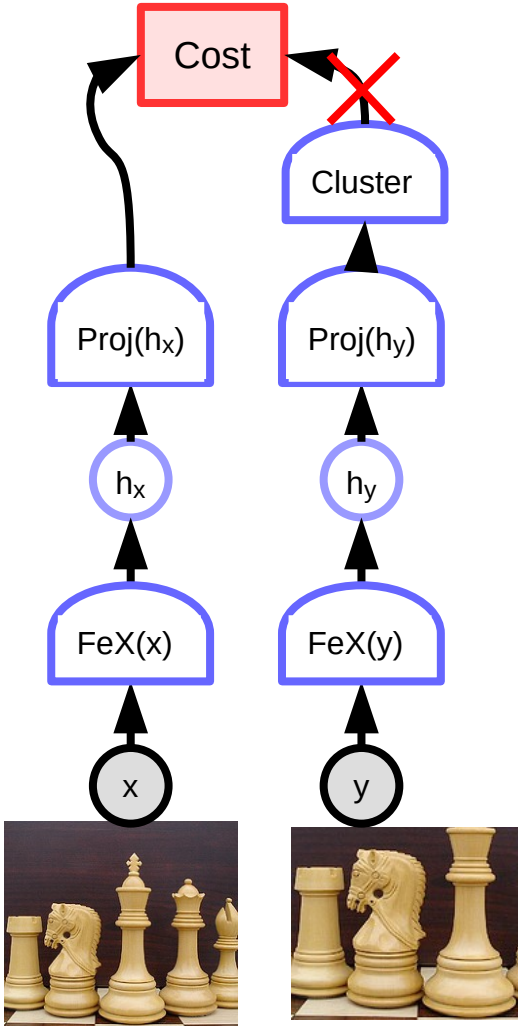
# Distillation Methods

- **Modified Siamese nets**
  - Predictor head eliminates variation of representations due to distortions
  - BYOL: Teacher branch uses a moving-average of the parameters of the student branch
- **Examples:**
  - Bootstrap Your Own Latents [Grill arXiv:2006.07733]
  - SimSiam [Chen & He arXiv:2011.10566]
- **Advantages**
  - No negative samples
- **Issues**
  - Not clear why they don't collapse (normalization?)



# Quantization Methods

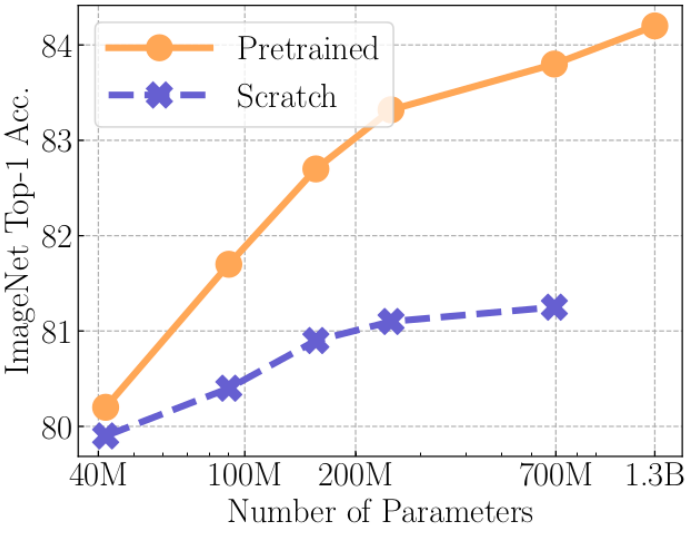
- **K-means clustering on embedding vectors**
  - Ensuring that all clusters are populated
    - Sinkhorn-Knapp procedure (information maximization)
  - Cluster centers used as targets for student branch
- **Examples**
  - DeepCluster [Caron arXiv:1807.05520]
  - SwAV [Caron arXiv:2006.09882]
- **Advantage:**
  - Works really well!
  - Uses large distortions (multicrop)
  - Scales to very large datasets

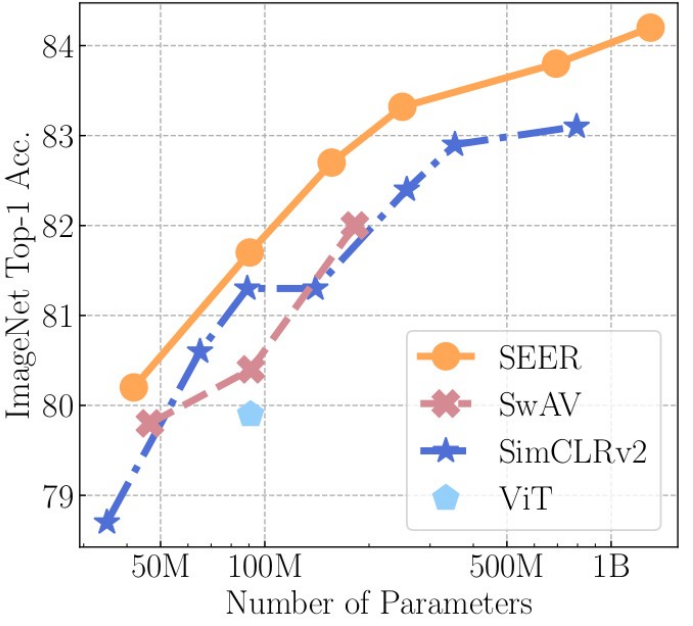


# SEER [Goyal et al. ArXiv:2103.01988]

- **SwAV training on 1 billion random IG images**
- **RegNet architecture**
- **Fine-tuned on various datasets**
  - ImgNet full: 84.% top-1 correct
  - ImgNet 10%: 77.9%,  ImgNet 1%: 60.5%
  - Inaturalist: 50.8%, Places205: 62.7%
  - Pascal VOC2007: 92.6%
- **Code: https://vissl.ai/**

| Method | Data | #images | Arch. | #param. | Top-1 |
|---|---|---|---|---|---|
| DeeperCluster [6] | YFCC100M | 96M | VGG16 | 138M | 74.9 |
| ViT [14] | JFT | 300M | ViT-B/16 | 91M | 79.9 |
| SwAV [7] | IG | 1B | RX101-32x16d | 182M | 82.0 |
| SimCLRv2 [9] | ImageNet | 1.2M | RN152w3+SK | 795M | 83.1 |
| SEER | IG | 1B | RG128 | 693M | 83.8 |
| SEER | IG | 1B | RG256 | 1.3B | **84.2** |



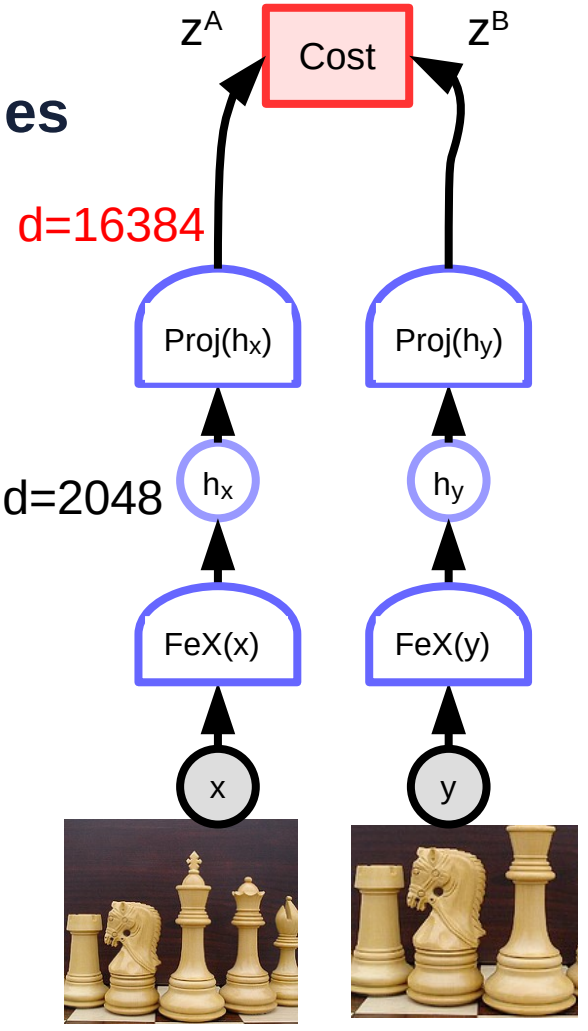



# Information Maximization

- **Minimizes redundancy between embedding variables**
  - Maximizes information content of embedding vectors
- **Example: Barlow Twins**
  - [Zbontar et al. ArXiv:2103.03230]
  - Maximizes normalized correlation between the same variable in the two branches over a batch.
  - Minimizes normalized correlation between different variables in the two branches
  - Centered vectors $z^A$,$z^B$

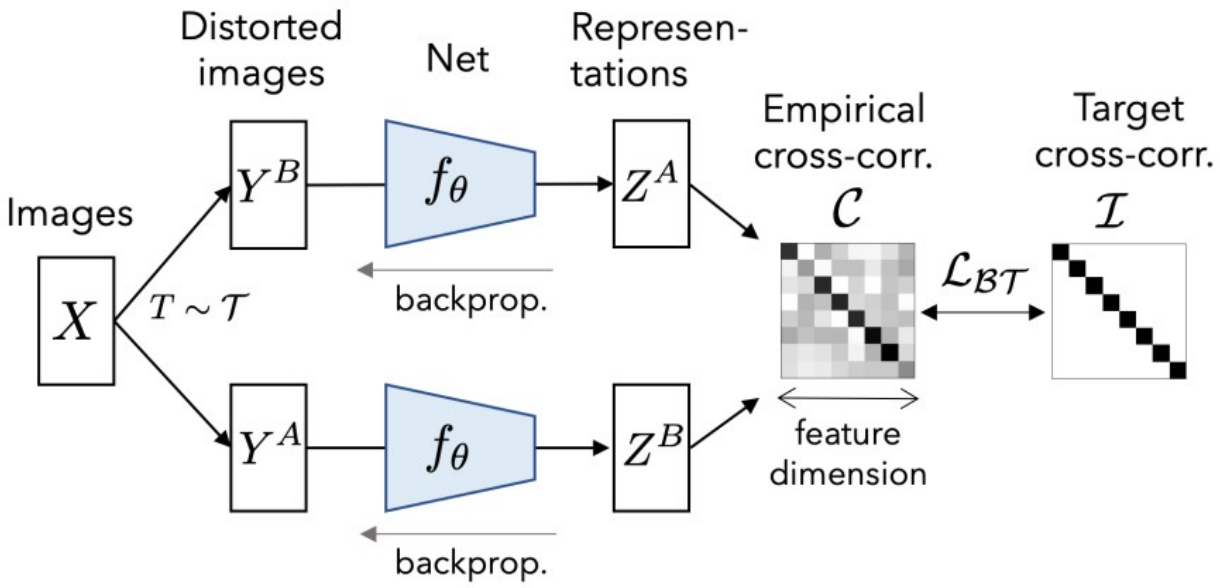
$$\mathcal{C}_{ij} \triangleq \frac{\sum_b z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_b \left(z^A_{b,i}\right)^2} \sqrt{\sum_b \left(z^B_{b,j}\right)^2}}$$

$$\mathcal{L}_{\mathcal{BT}} \triangleq \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}}$$



# Barlow Twins [Zbontar et al. ArXiv:2103.03230]

- Accuracy on ImageNet with linear classifier head
- Best accuracy for d=16,384 and batch-size=1024

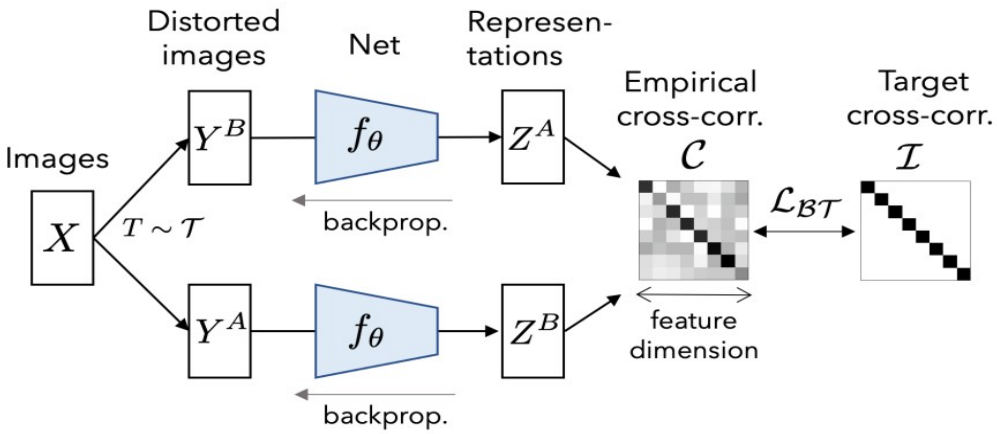



| Method | Top-1 | Top-5 |
|---|---|---|
| Supervised | 76.5 | |
| MoCo | 60.6 | |
| PIRL | 63.6 | - |
| SimCLR | 69.3 | 89.0 |
| MoCo v2 | 71.1 | 90.1 |
| SimSiam | 71.3 | - |
| SwAV | 71.8 | - |
| BYOL | 74.3 | 91.6 |
| SwAV (w/ multi-crop) | 75.3 | - |
| Barlow Twins (ours) | 73.2 | 91.0 |

**Updated version: "VICReg" [Bardes et al. ArXiv:2105.04906]**

# Barlow Twins [Zbontar et al. ArXiv:2103.03230]

- Accuracy on ImageNet with linear classifier head

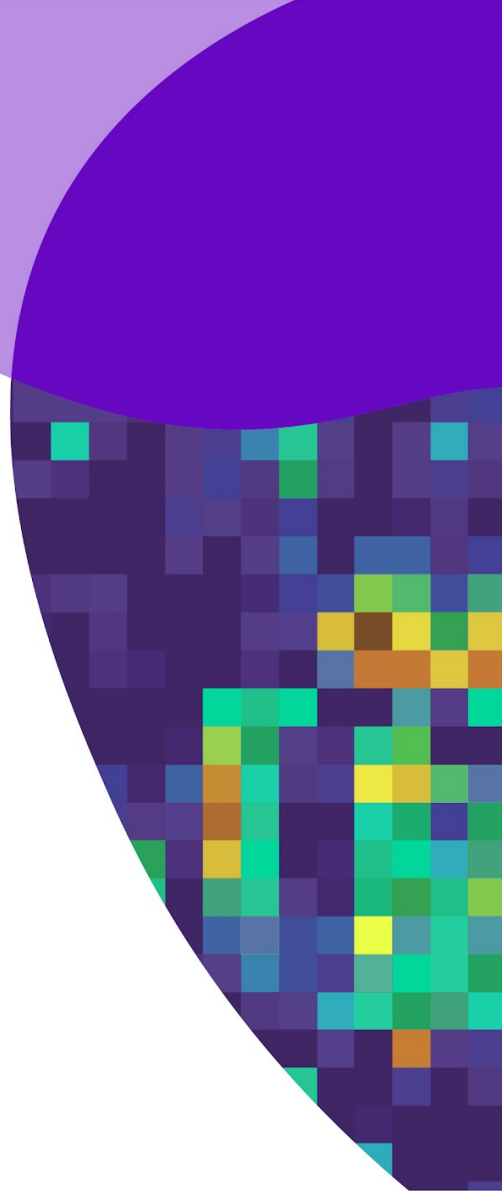



| Method | Top-1 | | Top-5 | |
|---|---|---|---|---|
| | 1% | 10% | 1% | 10% |
| Supervised | 25.4 | 56.4 | 48.4 | 80.4 |
| PIRL | - | - | 57.2 | 83.8 |
| SIMCLR | 48.3 | 65.6 | 75.5 | 87.8 |
| BYOL | 53.2 | 68.8 | 78.4 | 89.0 |
| SWAV (w/ multi-crop) | 53.9 | **70.2** | 78.5 | **89.9** |
| BARLOW TWINS (ours) | **55.0** | 69.7 | **79.2** | 89.3 |

| Method | VOC07+12 det | | | COCO det | | | COCO instance seg | | |
|---|---|---|---|---|---|---|---|---|---|
| | $AP_{all}$ | $AP_{50}$ | $AP_{75}$ | $AP^{bb}$ | $AP^{bb}_{50}$ | $AP^{bb}_{75}$ | $AP^{mk}$ | $AP^{mk}_{50}$ | $AP^{mk}_{75}$ |
| Sup. | 53.5 | 81.3 | 58.8 | 38.2 | 58.2 | 41.2 | 33.3 | 54.7 | 35.2 |
| MoCo-v2 | **57.4** | 82.5 | **64.0** | **39.3** | 58.9 | **42.5** | **34.4** | 55.8 | 36.5 |
| SwAV | 56.1 | **82.6** | 62.7 | 38.4 | 58.6 | 41.3 | 33.8 | 55.2 | 35.9 |
| SimSiam | 57 | 82.4 | 63.7 | 39.2 | **59.3** | 42.1 | **34.4** | **56.0** | **36.7** |
| BT (ours) | 56.8 | **82.6** | 63.4 | 39.2 | 59.0 | **42.5** | 34.3 | **56.0** | 36.5 |

FACEBOOK AI

# Generative Latent-Variable Architectures 1. Architectural methods



Construct the architecture so that the volume of low-energy space is fixed or limited.

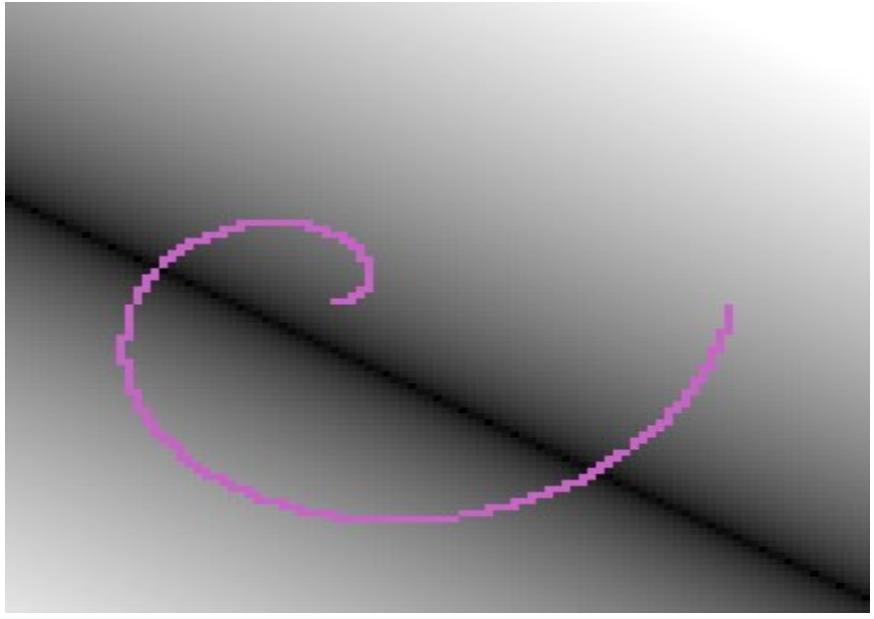
# Architectural Methods

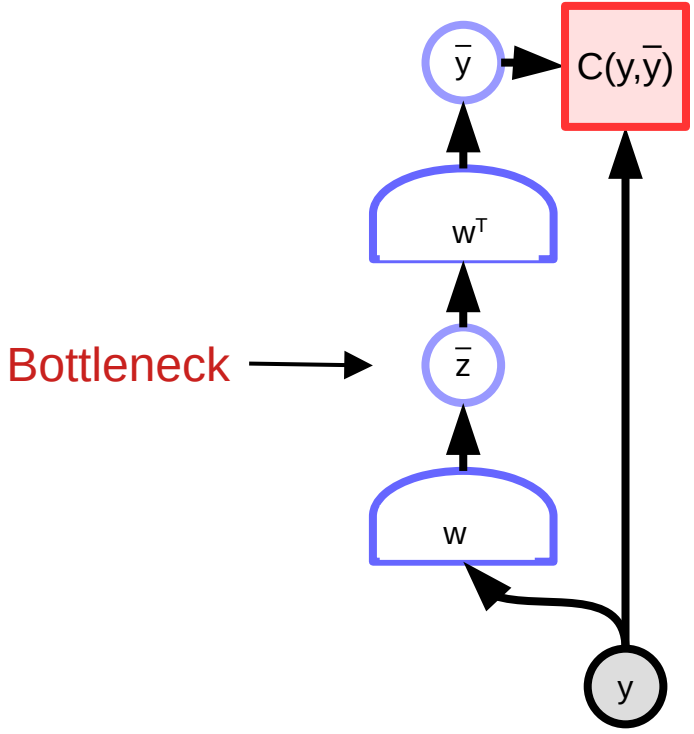
- **Architectural:** **[Different ways to limit the information capacity of the latent representation]**
  - A1: build the machine so that the volume of low energy space is bounded:
  - PCA: volume is dimension of principal subspace
  - K-means: volume limited by number of prototypes
  - Analytically normalized probabilistic models: volume is fixed

    Gaussian, and Gaussian Mixture Model (model is normalized)
  - Square Independent Component Analysis
  - Latent variable models with fixed latent distribution: volume is less than the "volume" (entropy) of the latent variable prior distribution.
    - Normalizing flows

# Principal Component Analysis

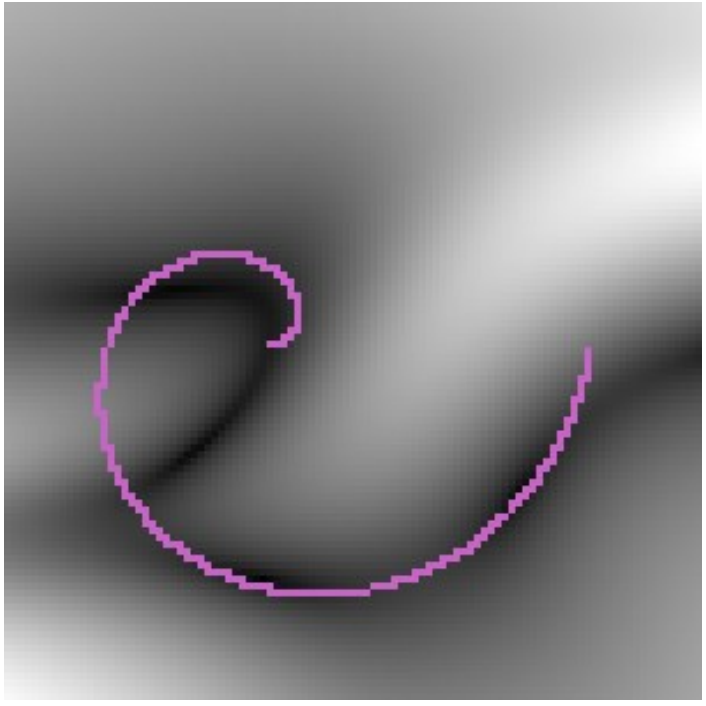
- **PCA is a 2-layer linear auto-encoder with a bottleneck.**
  - Energy: $F_w(y) = ||y - Dec(Enc(y))||^2 = ||y - w^T w||^2$
  - Loss $L(y, w) = F_w(y)$

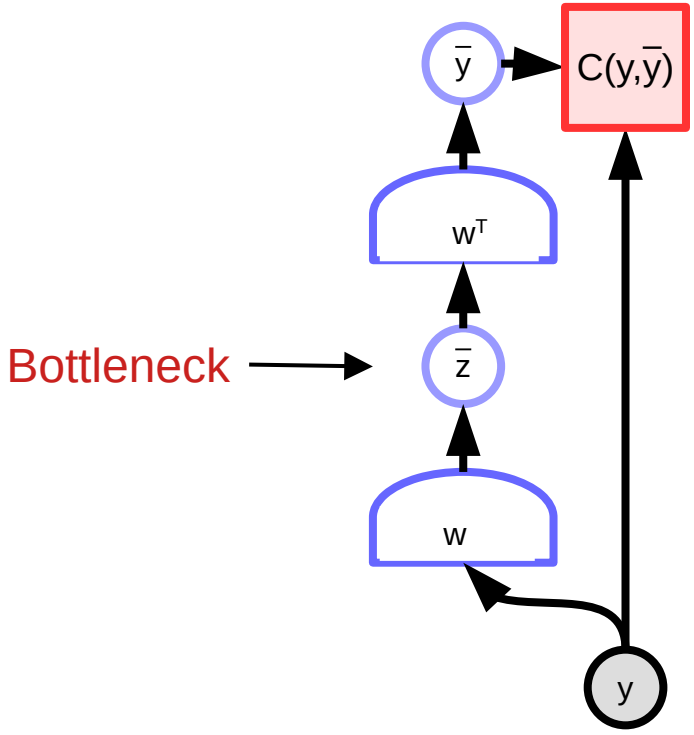




# Auto-Encoder with Bottleneck

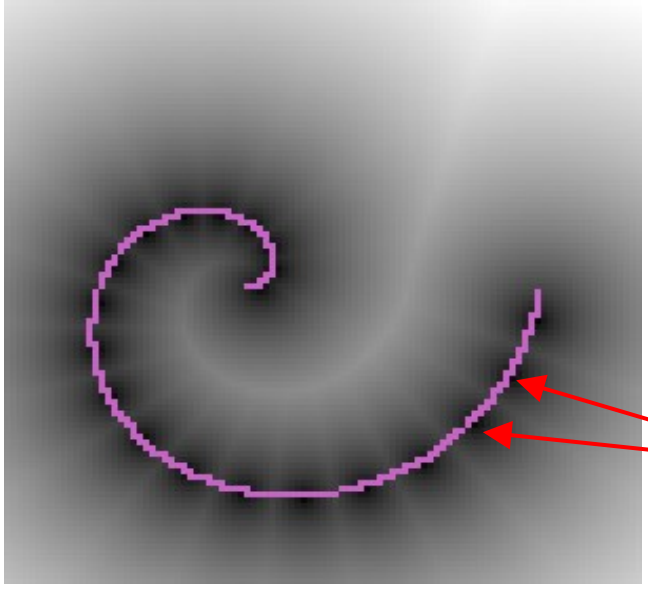
- **non-linear auto-encoder with a bottleneck.**
  - Energy: $F_w(y) = ||y - Dec(Enc(y))||^2$
  - Loss: $L(y, w) = F_w(y)$



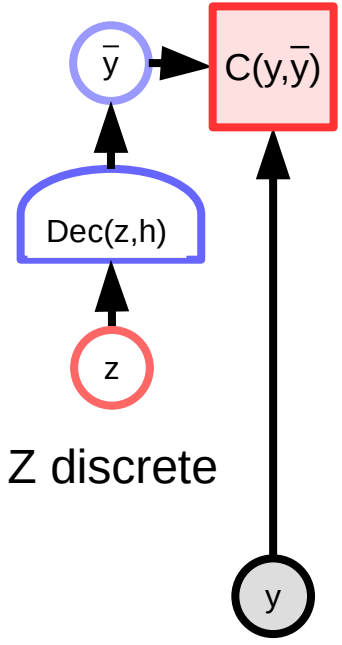


# K-Means

- **Discrete latent-variable model with linear decoder**
  - Energy: $E(y,z) = ||y - Dec(z)||^2 = ||y - wz||^2$
  - Free Energy $F(y) = \min_{z \in \mathcal{Z}} E(y,z)$
  - Loss: $L(y,w) = F_w(y)$

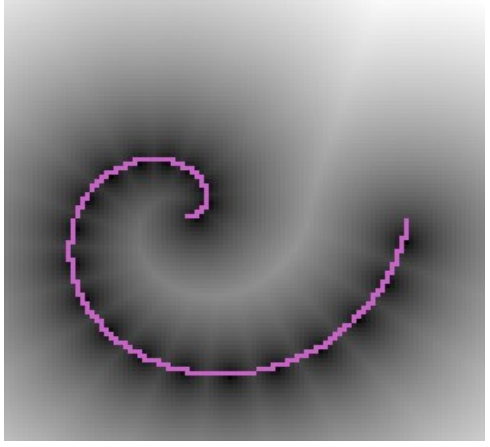


- Latent vector z is constrained to be a 1-hot vector: [0,0,..,01,0,..,0]
- 1 component selects a column of w
- F(y)=0 iff y is equal to a column of w.

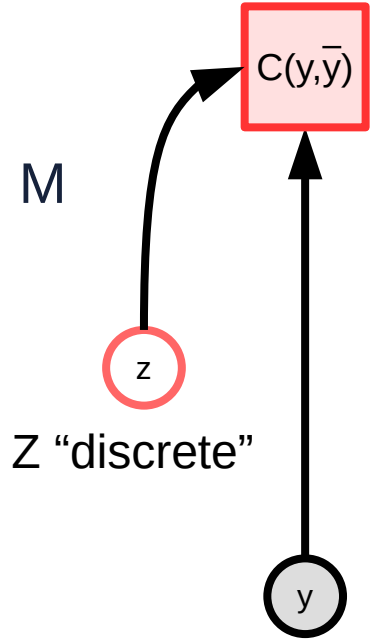


# Gaussian Mixture Model

- **Similar to K-means with soft marginalization over latent.**
  - Energy: $E(y,z) = (y - wz)^T (Mz)(y - wz)$ $\quad (Mz)_{ij} = \sum_k M_{ijk} z_k$
  - Free Energy $\quad F(y) = -\frac{1}{\beta} \log \sum_{z \in \mathcal{Z}} e^{\beta E(y,z)}$
  - Loss: $L(y,w) = F_w(y)$ with normalization constraint on M

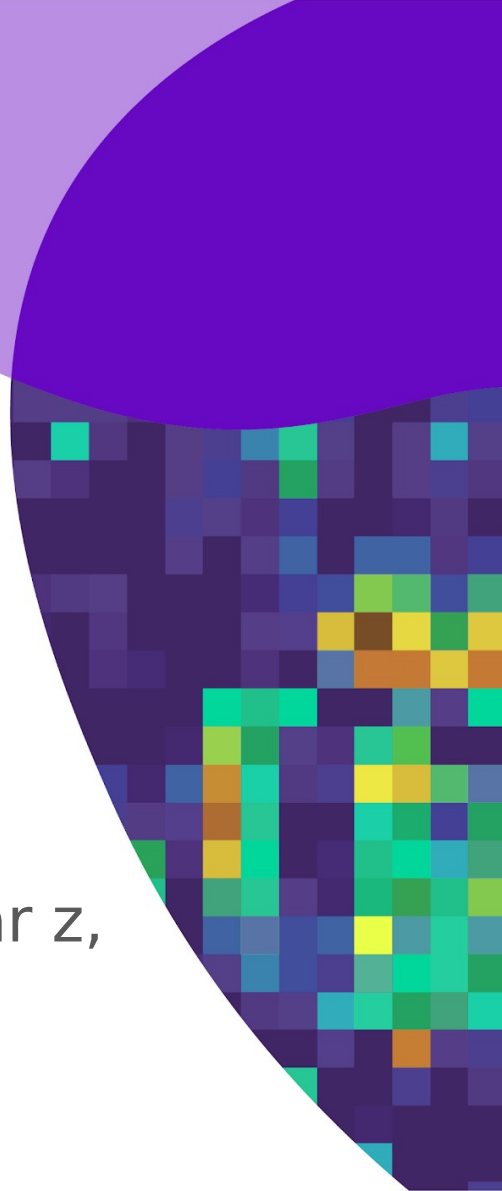


- Latent vector z is constrained to be a 1-hot vector: [0,0,..,01,0,..,0]
- But marginalization makes it “soft”



FACEBOOK AI

# Generative Latent-Variable Architectures
# 2. Regularized Latent-Var Methods



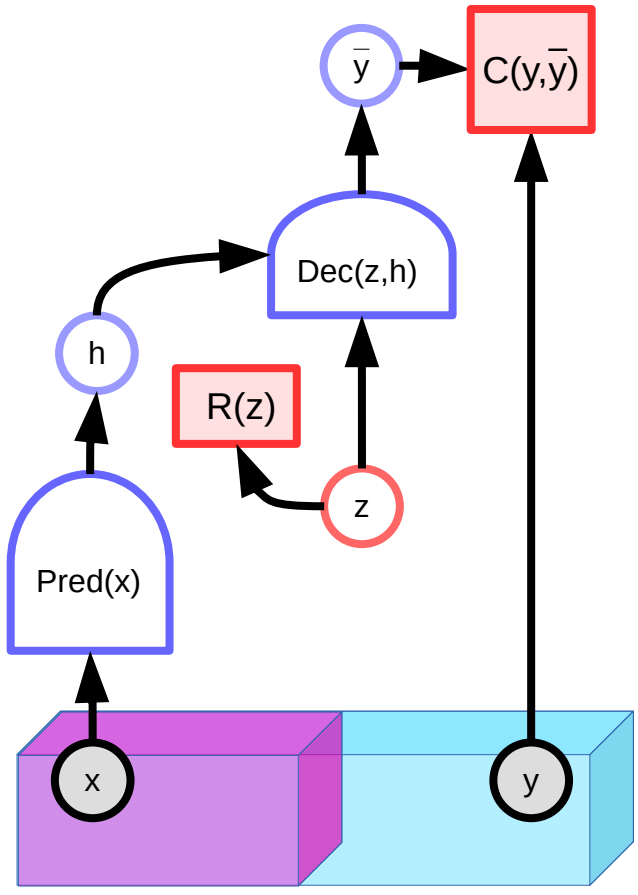
1 predict y,
2 parameterize plausible y through a latent var z,
3 limit/regularize the information capacity of z

# 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**

$$E(x, y, z) = C(y, \mathrm{Dec}(\mathrm{Pred}(x), z))$$

- **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: regularizer R(z) to limit the information capacity of the latent variable z.**



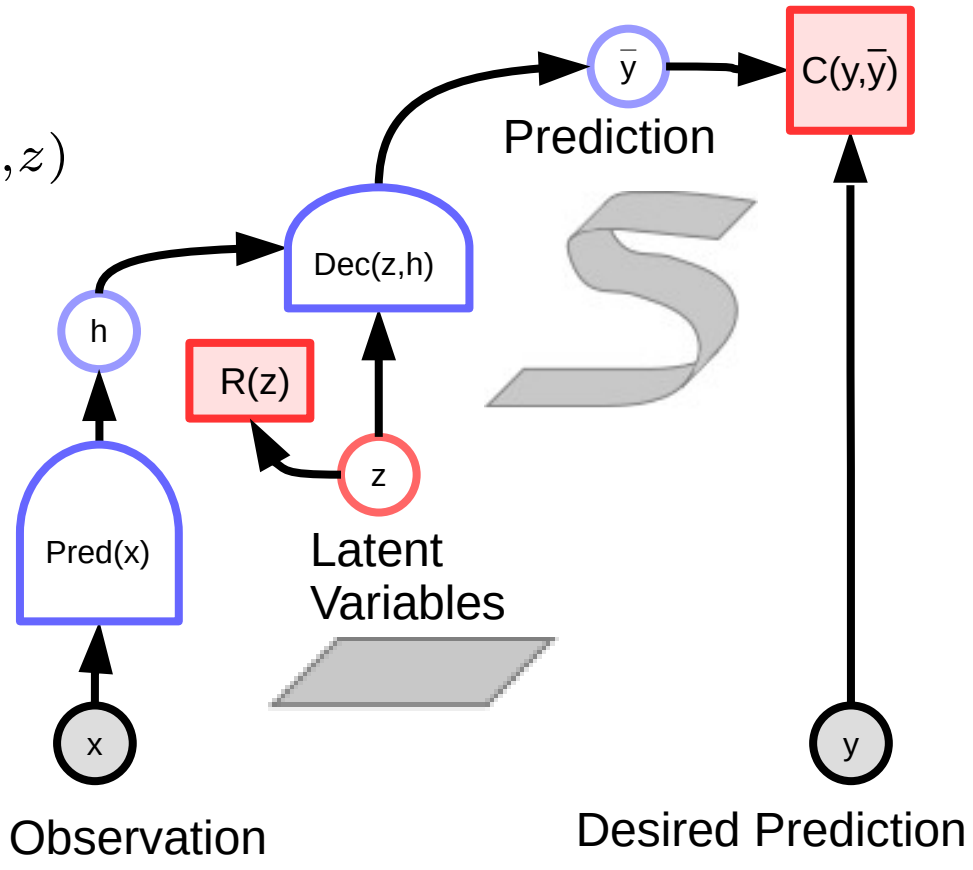
# Architecture for Multimodal Output: latent variable EBM

- **Latent variables: parameterize the set of predictions**

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

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

- **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.

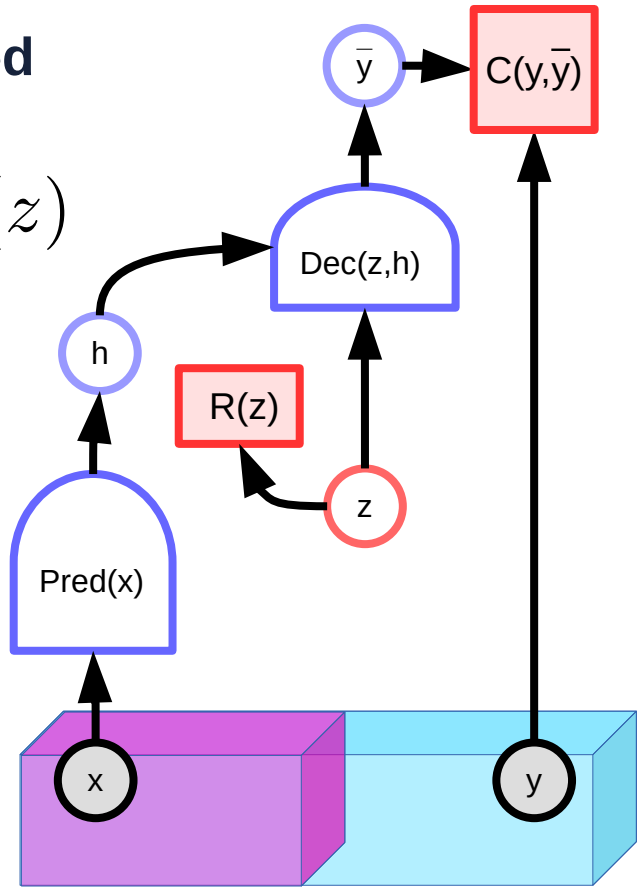


# Regularized Latent Variable EBM

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

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

- **Examples of R(z):**
  - Effective dimension / spectrum of covariance
  - 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>

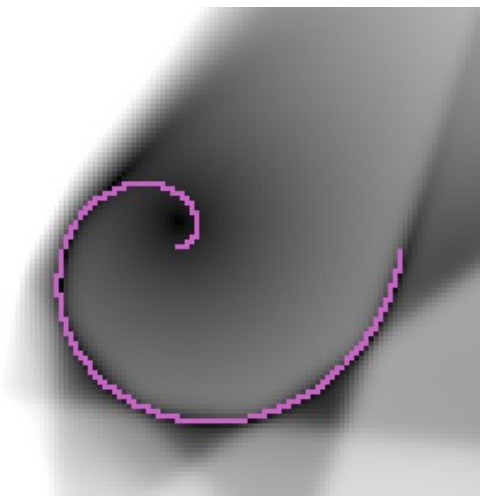


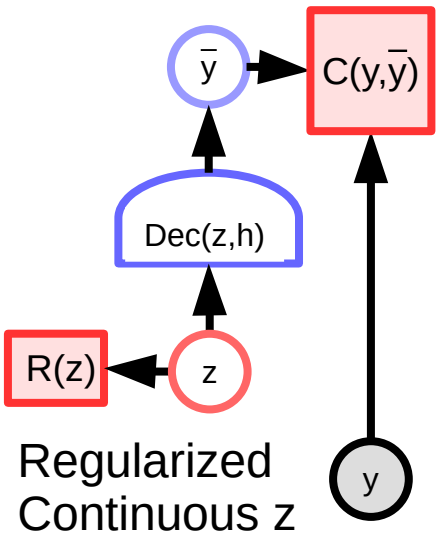
# Regularized Latent Variable Methods: Sparse Coding

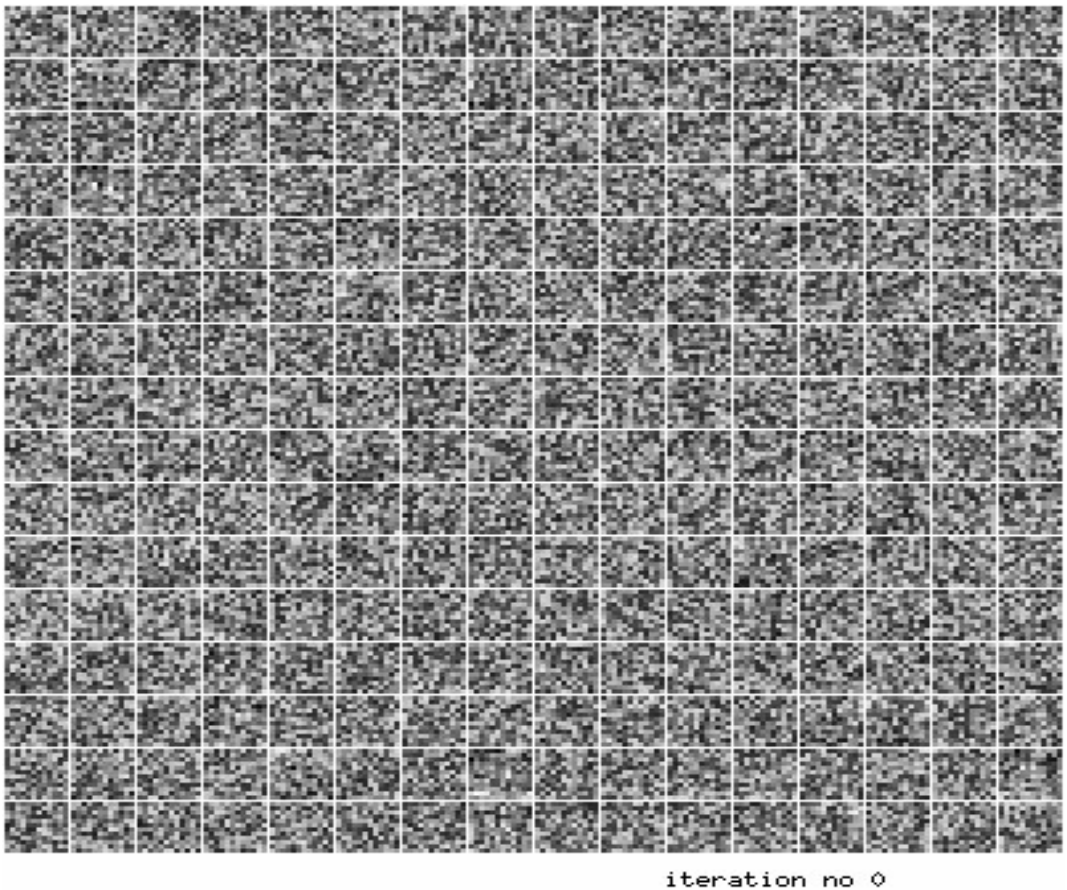
- **A2:** **regularize the volume of the low energy regions**

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

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

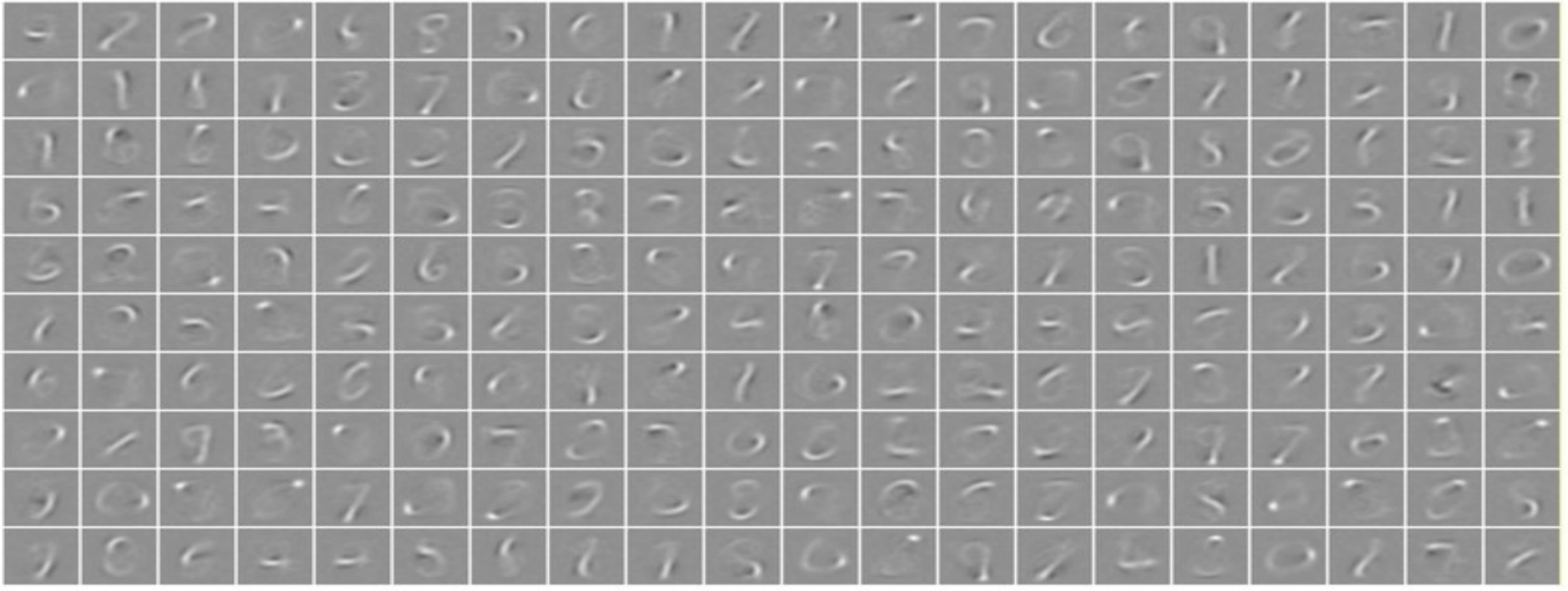




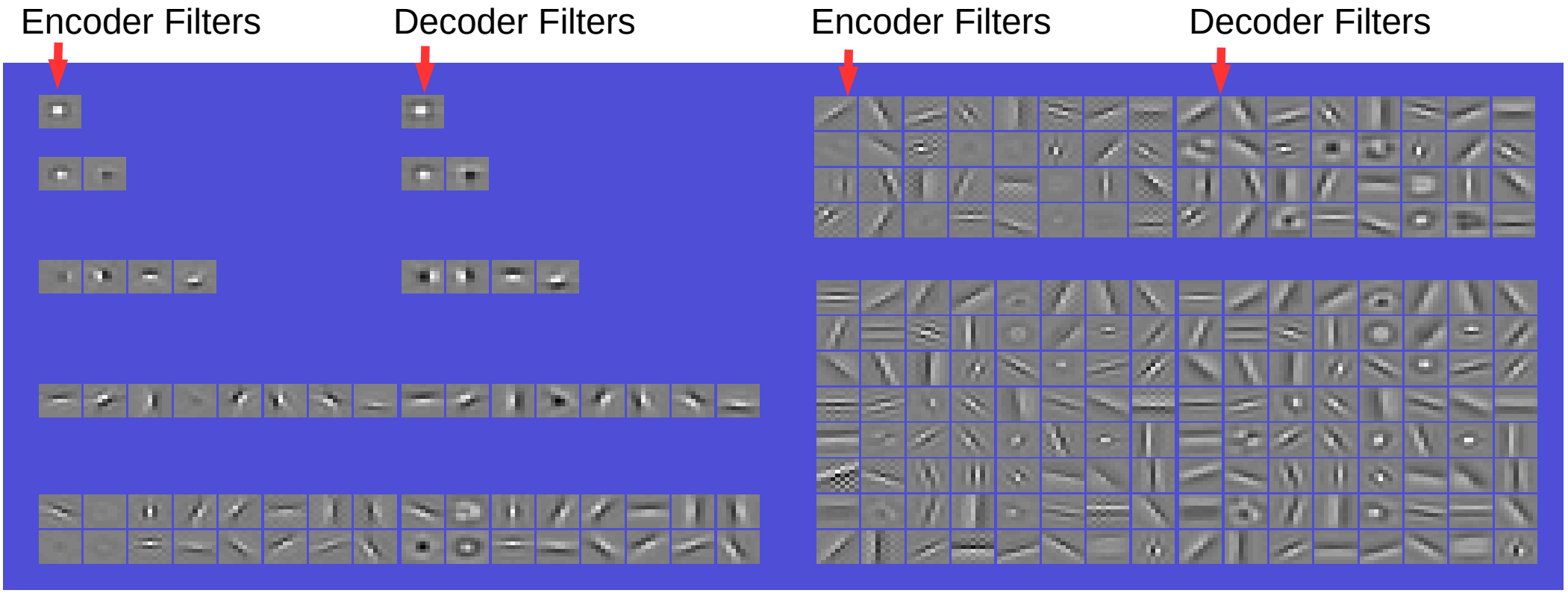
# Sparse Modeling on handwritten digits (MNIST)

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



# Convolutional Sparse Auto-Encoder on Natural Images

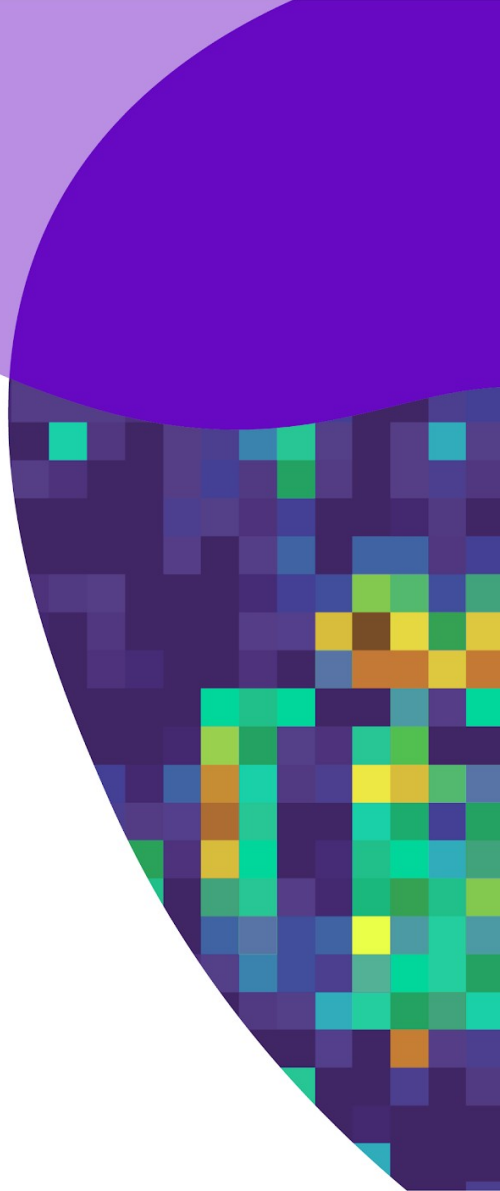
- **Filters and Basis Functions obtained. Linear decoder (conv)**
  - with 1, 2, 4, 8, 16, 32, and 64 filters [Kavukcuoglu NIPS 2010]



FACEBOOK AI

# Amortized Inference: Learning to predict the latent variable

Regularized auto-encoders:
Sparse AE
Variational AE



# Amortized Inference

- **Training an encoder to give an approximate solution to the inference optimization problem**

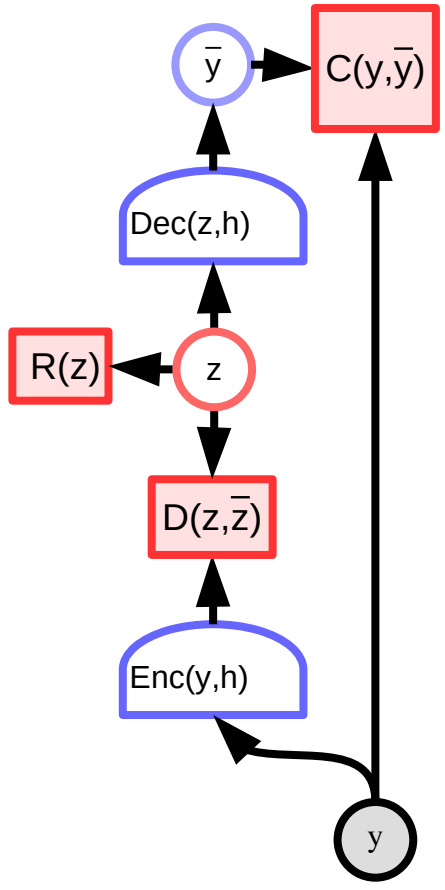
- **Regularized Auto-Encoder, Sparse AE, LISTA**

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

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

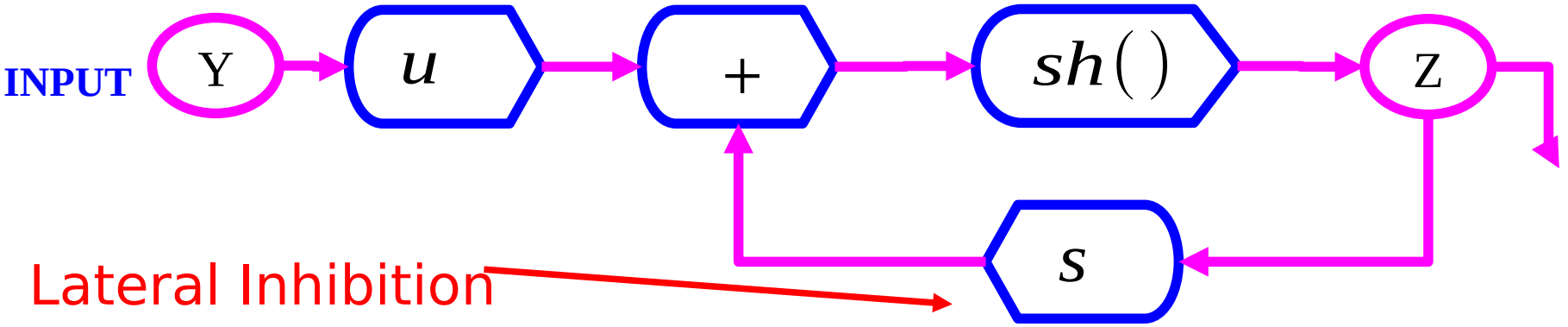
- **Variational AE**
  - Approximated by sampling and variational approximation

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



# Giving the "right" structure to the encoder

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




$$z(t+1) = \text{Shrink}_{\alpha\eta}\left[z(t) - \eta w^t(wz(t) - y)\right]$$

- **ISTA/FastISTA reparameterized:**

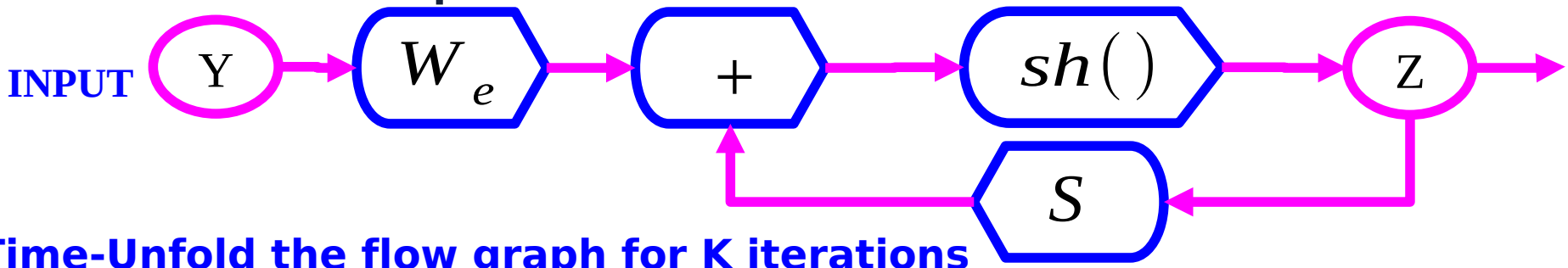
$$z(t+1) = \text{Shrink}_{\alpha\eta}\left[sz(t) + uy\right]; \quad u = \eta w; \quad s = I - \eta w^T w$$

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

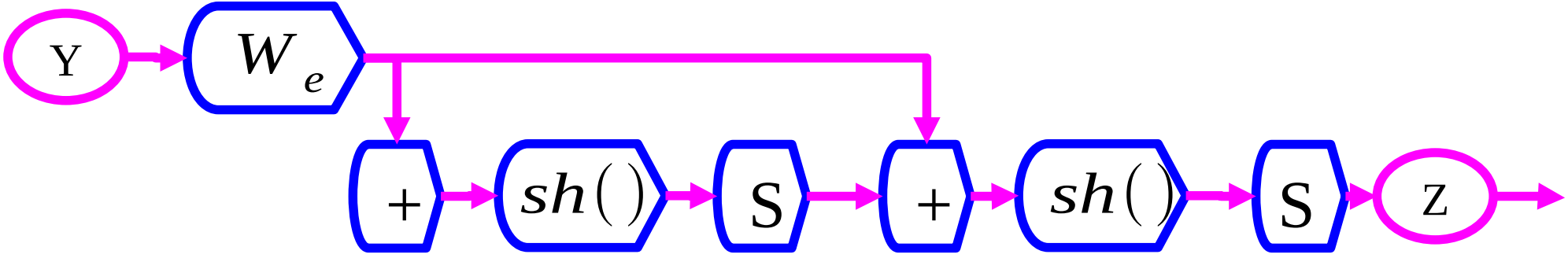
[Gregor & LeCun, ICML 2010], [Bronstein et al. ICML 2012], [Rolfe & LeCun ICLR 2013]

# 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**



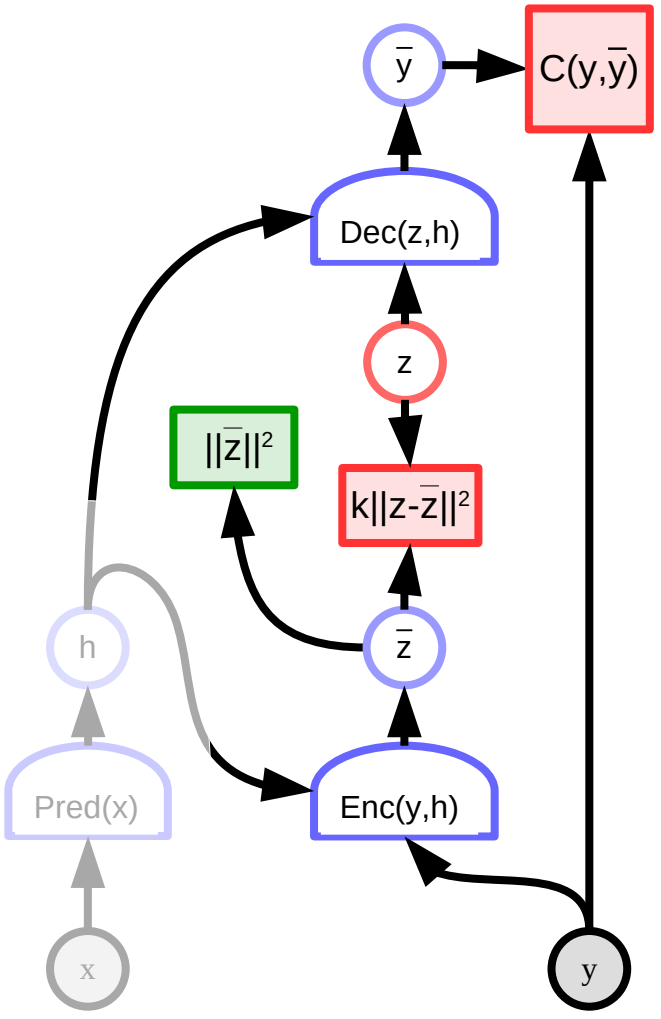

- **Time-Unfold the flow graph for K iterations**
- **Learn the We and S matrices with "backprop-through-time"**
- **Get the best approximate solution within K iterations**



# 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.**

$$E(y,z) = C(y, Dec(z)) + k(z - Enc(y))^T M(z - Enc(y)) + \gamma ||Enc(y)||^2$$



# Variational Auto-Encoder

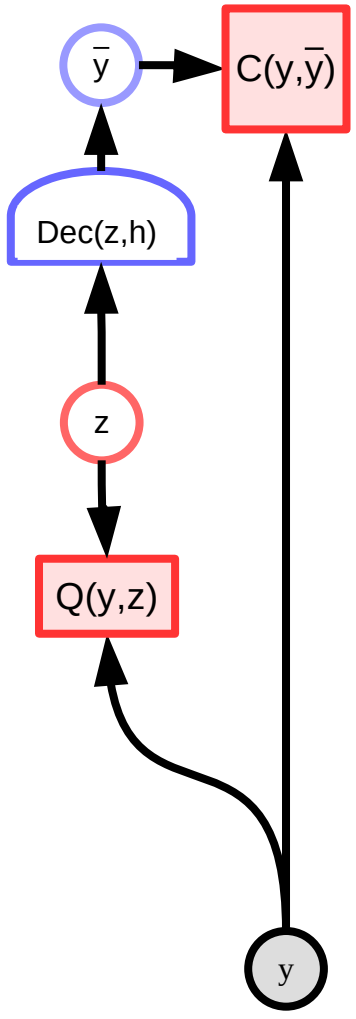
- **Variational approximation of marginalization over z**

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

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

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

$$q(z|y) = \frac{e^{-\beta Q(y,z)}}{\int_{z'} e^{-\beta Q(y,z')}}$$



# Variational Auto-Encoder

- **Variational approximation of marginalization over z**

$$F(y) = -\frac{1}{\beta} \log \int_z q(z|y) \frac{e^{-\beta E(y,z)}}{q(z|y)}$$

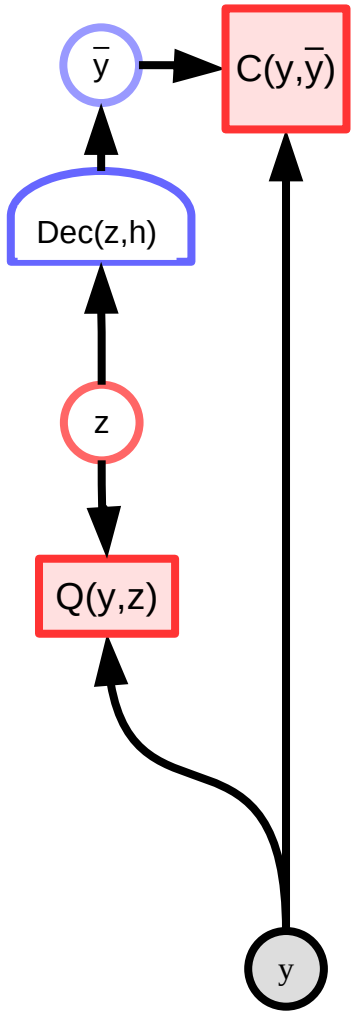
**Jensen's inequality: log(average()) < average(log())**

$$F(y) \leq \tilde{F}(y) = \int_z q(z|y) \left[ -\frac{1}{\beta} \log \frac{e^{-\beta E(y,z)}}{q(z|y)} \right]$$

- **Variational free energy:**

$$\tilde{F}(y) = \int_z q(z|y) E(y,z) + \frac{1}{\beta} \int_z q(z|y) \log(q(z|y))$$

$$F = \langle E \rangle - TS$$



# Variational Auto-Encoder

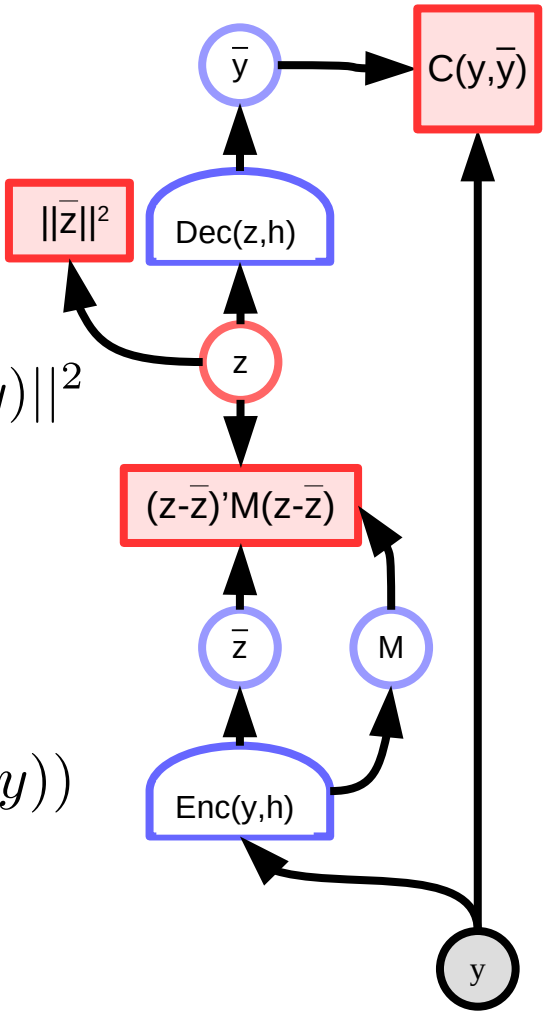
- **Which q(z|y) ?**
- **If Q(y,z) is quadratic, q(z|y) is Gaussian.**

$$Q_{w_e}(y,z) = (z - \mathrm{Enc}_{w_e}(y))^T M (z - \mathrm{Enc}_{w_e}(y)) + \gamma ||\mathrm{Enc}_{w_e}(y)||^2$$

$$q_{w_e}(z|y) = \frac{e^{-\beta Q_{w_e}(y,z)}}{\int_{z'} e^{-\beta Q_{w_e}(y,z')}}$$

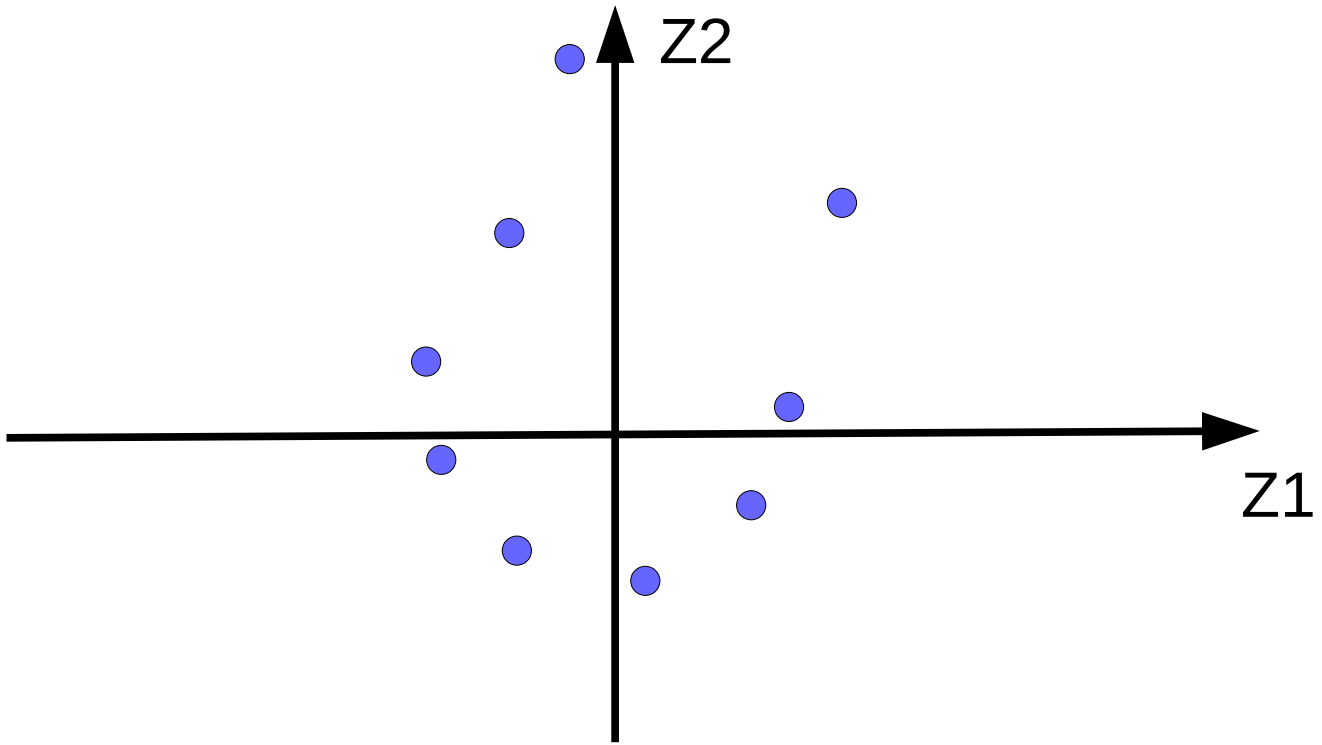
$$\tilde{F}_w(y) = \int_z q_{w_e}(z|y) E_{w_d}(y,z) + \frac{1}{\beta} \int_z q_{w_e}(z|y) \log(q_{w_e}(z|y))$$

- **Loss** $\mathcal{L}(y,w) = \tilde{F}_w(y)$



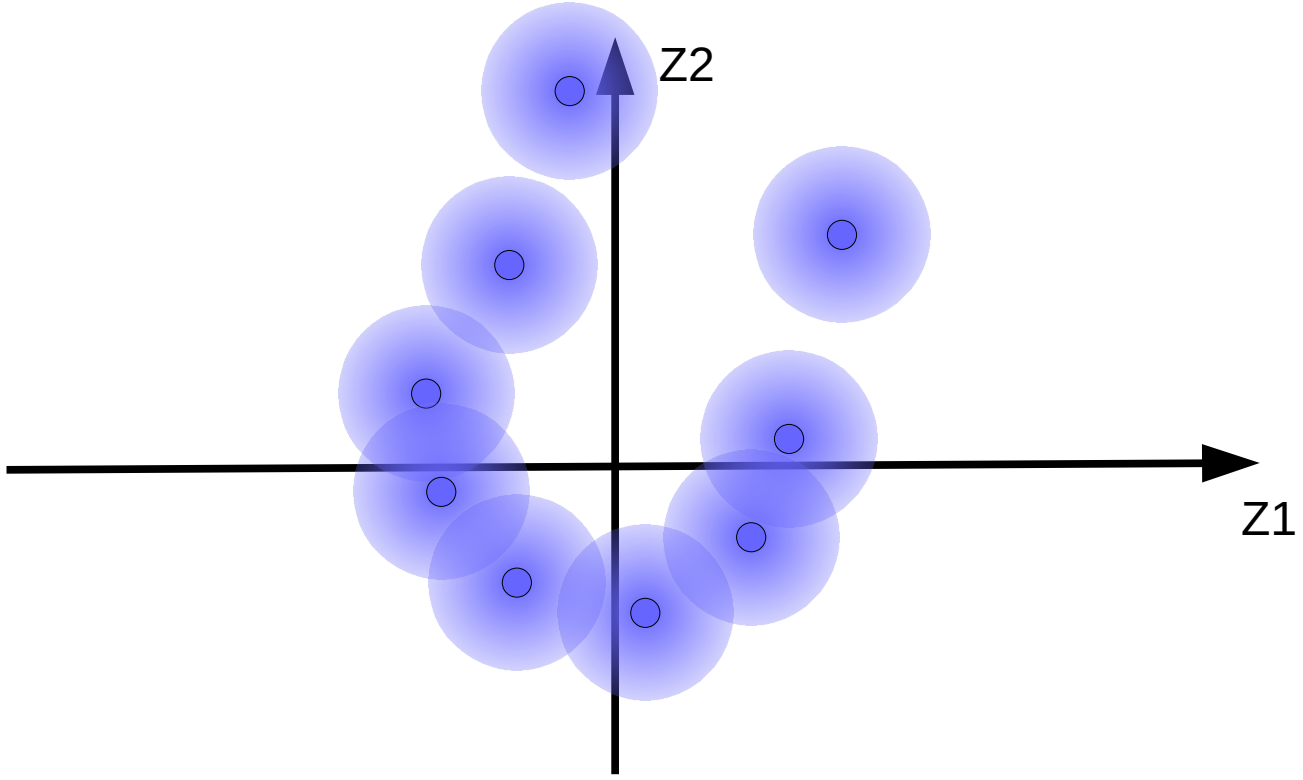
# Variational Auto-Encoder

- **Code vectors for training samples**



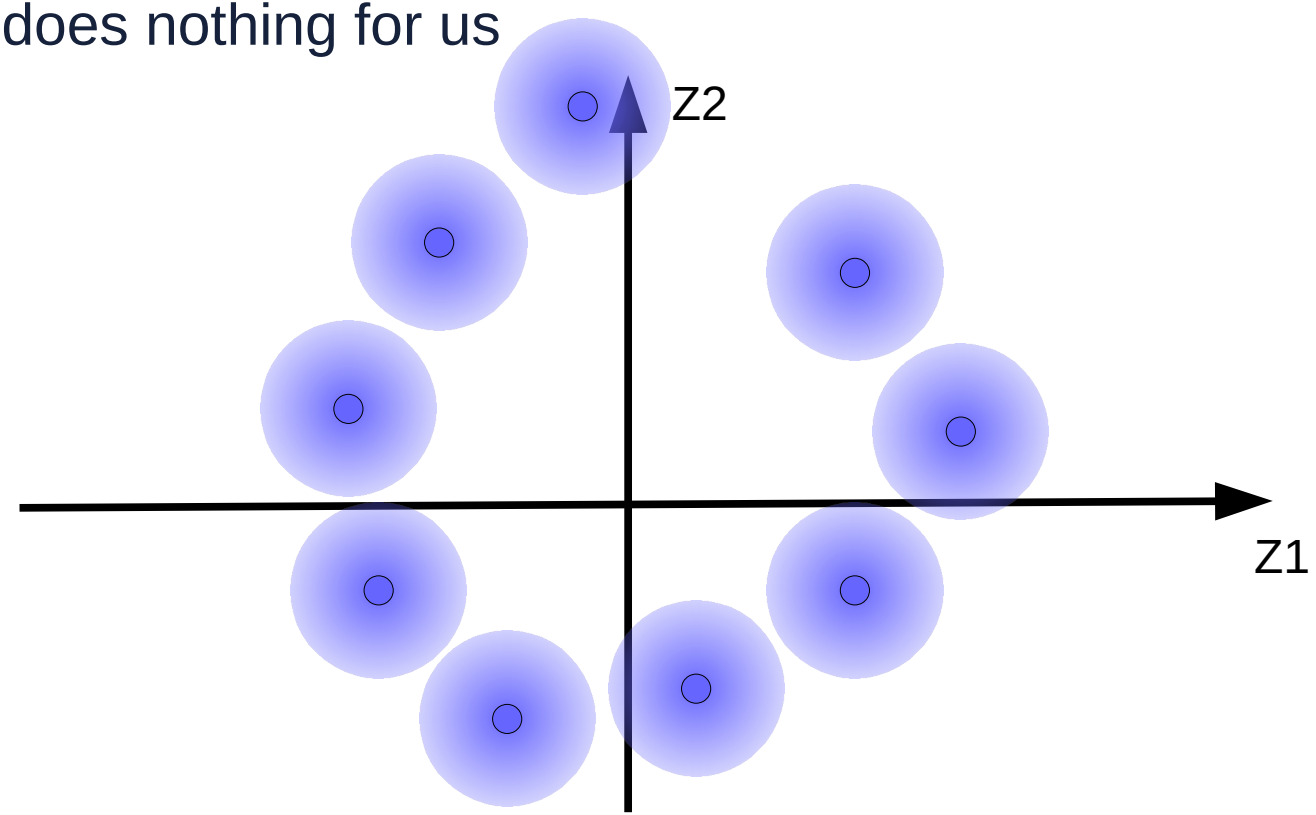
# Variational Auto-Encoder

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



# Variational Auto-Encoder

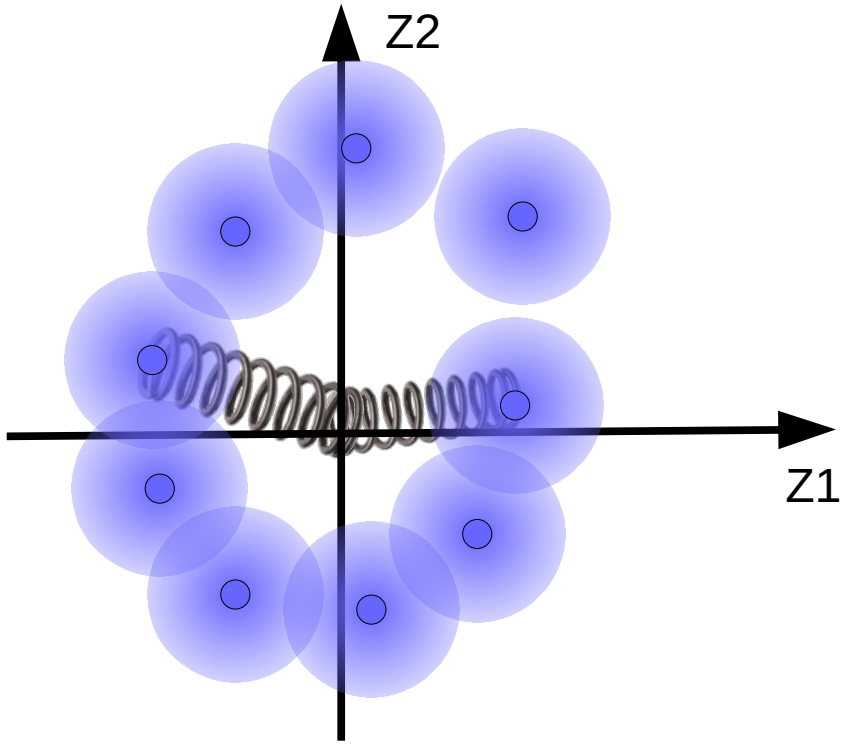
- **The code vectors want to move away from each other to minimize reconstruction error**
  - But that does nothing for us



# 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

Talia Konkle

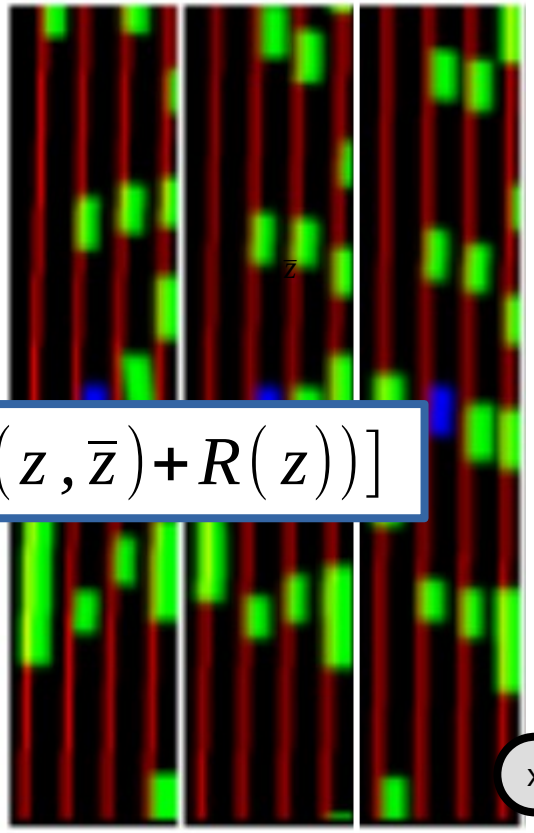


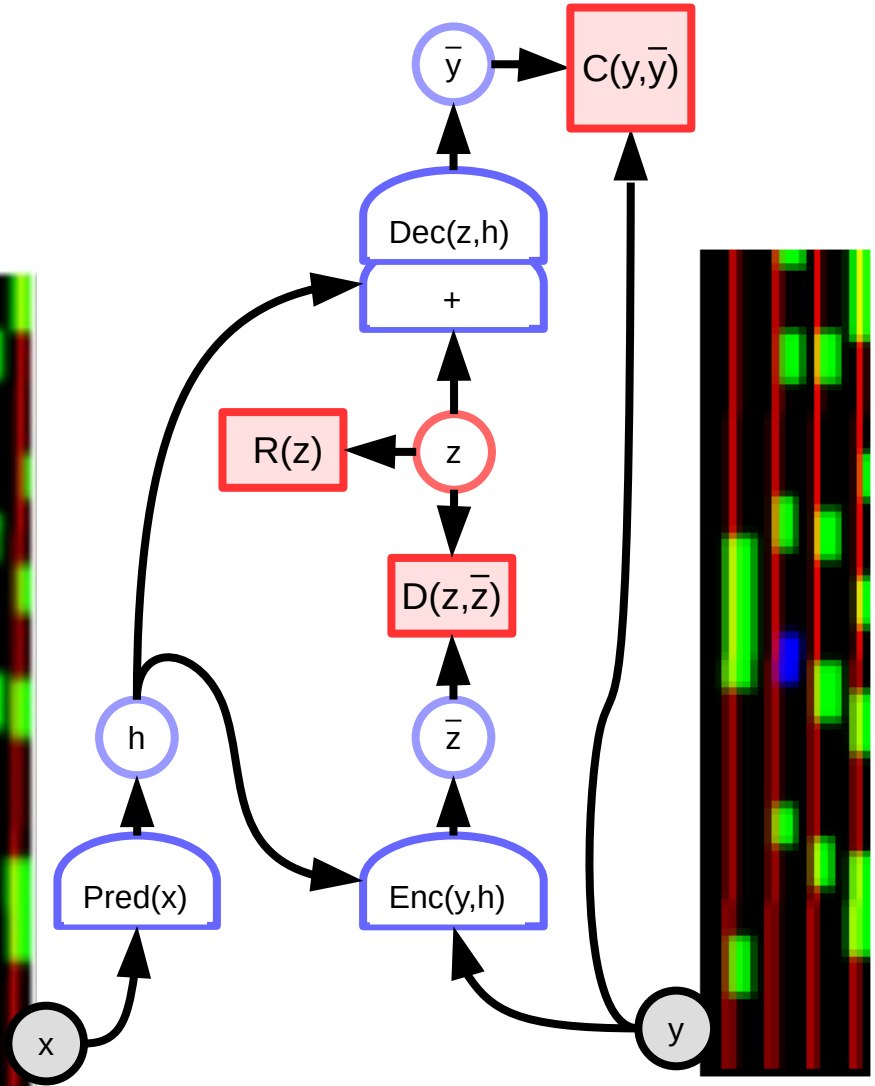
# Conditional VAE + Drop Out

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

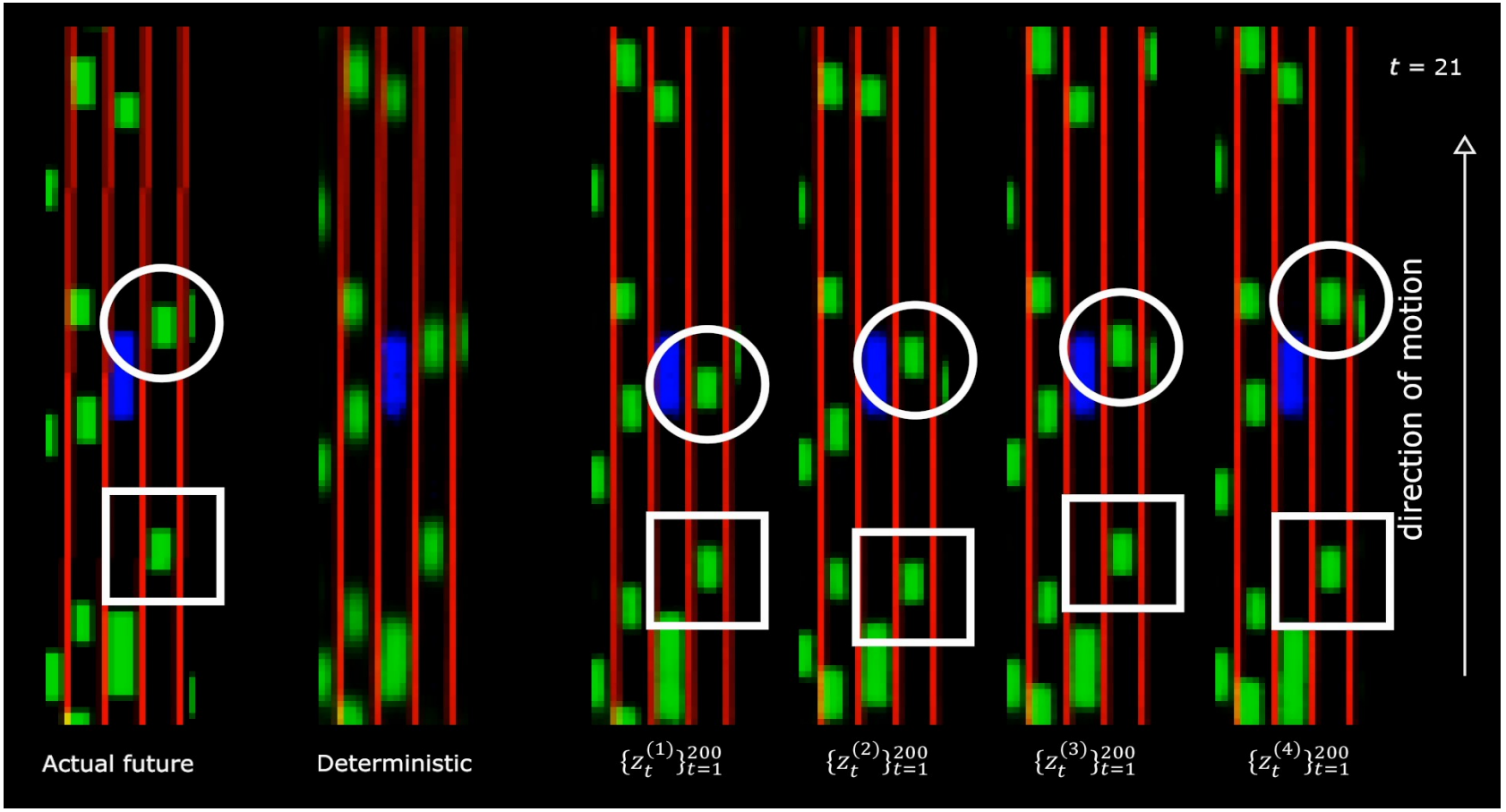
$$P(z/\bar{z}) \propto \exp[-\beta(D(z,\bar{z})+R(z))]$$

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



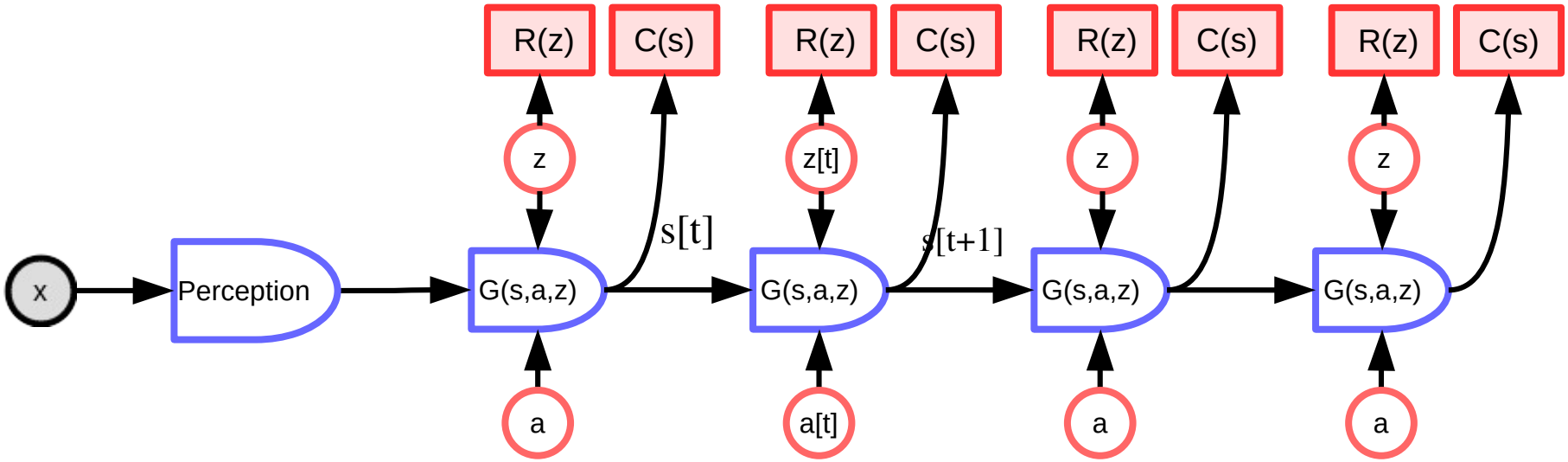



# Actual, Deterministic, VAE+Dropout Predictor/encoder



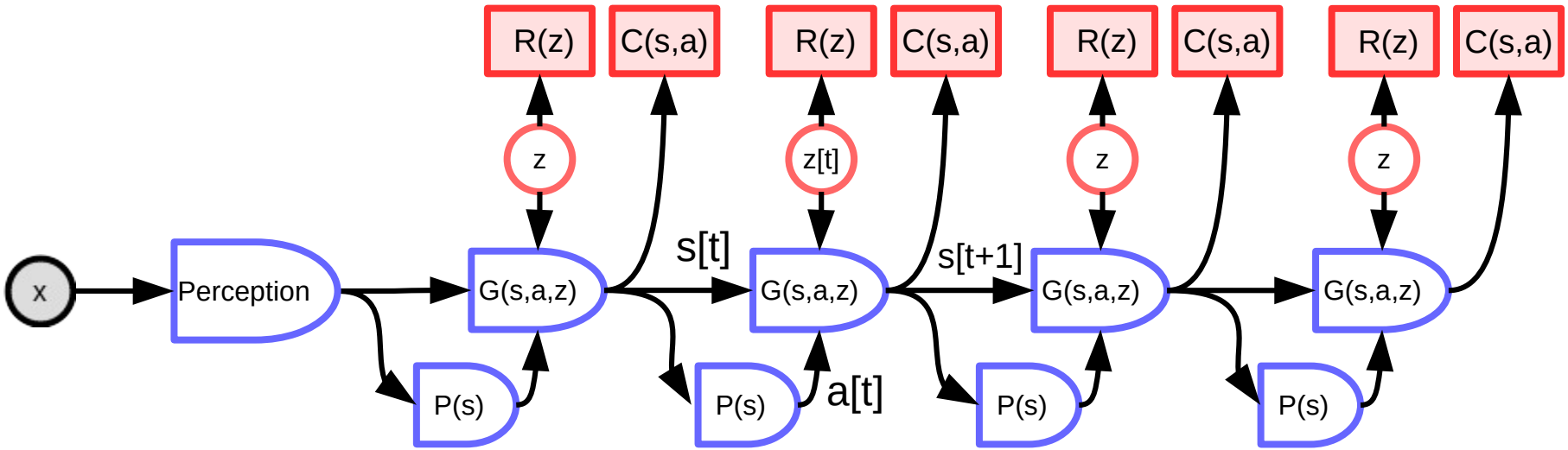
# Forward Model for Model-Predictive Control

- **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]…,a[T]) = argmin $\Sigma_t$ C(s[t])**
  **through backprop (== Kelley-Bryson adjoint state method)**



# 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)**



# Conclusion

- **SSL is the future of representation learning**
  - Contrastive reconstruction-based SSL works very well in NLP
  - Contrastive prediction-based SSL works well in speech
  - Reconstruction/prediction does not work well for images
  - Joint embedding methods work better for images.
  - Contrastive joint embedding is too expensive
  - The most promising methods are non contrastive:
    - Distillation (BYOL, SimSiam), Clustering (SwAV), infomax (Barlow Twins)
- **SSL can be used to learn world models under uncertainty**
  - Regularized latent variable models (e.g VAE).

# 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**

# World models for autonomous AI systems

- **World Model: predicts future states**
- **Critic: predicts expected objective**
- **Cost: computes objective**
- **Perception: estimates world state**
- **Actor: computes action**

- **We only have one world model engine!**

- **Configurator: configures the world model engine for the situation at hand.**