# Unsupervised Learning

## Lecture 10

Includes slides from Emily Denton,
Abhinav Gupta & others

# Overview

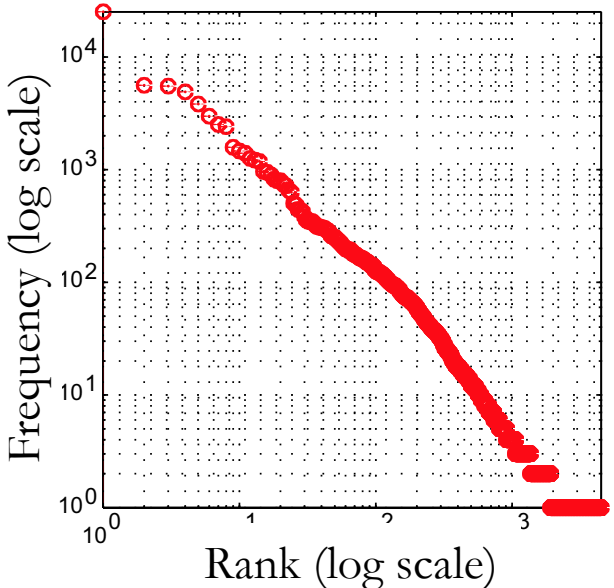
- Motivation
- Unsupervised Learning
  - Literature review
  - Generative models of video
    - [Stochastic Video Generation, Denton et al., ICML 2018]

- Self-supervised learning
  - Review of approaches from vision
  - [Unsupervised Learning by Predicting Noise, Bojanowski & Joulin, ICML 2017]

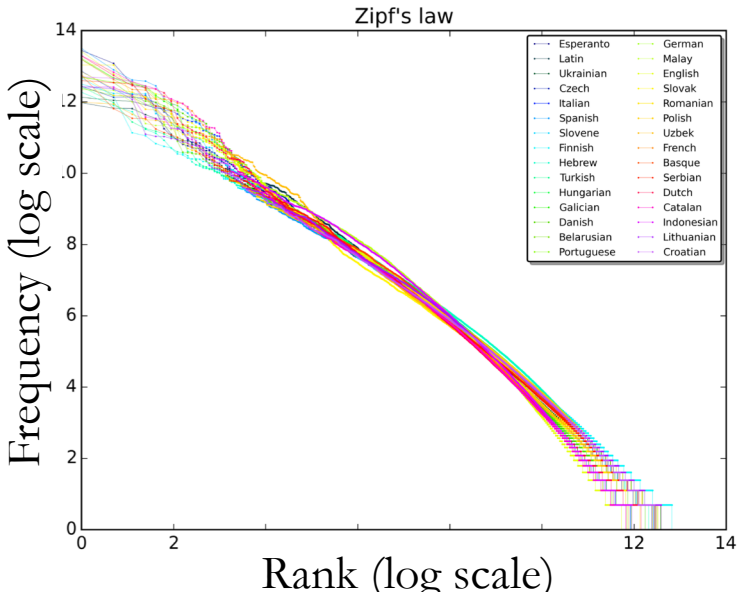
# Unsupervised Learning

- Learning without labels (or from just a few)
  - Need to capture structure inherent in data
- Practical importance:
  - Very uneven distribution of categories in real-world (Zipf's law)
  - Lots of rare categories with few examples

Objects in Vision Dataset (LabelMe)




Words in Wikipedia Corpus



# Arguments for Unsupervised Learning

- Want to be able to exploit unlabeled data
  - Vast amount of it often available
  - Essentially free
- Good regularizer for supervised learning
  - Helps generalization
  - Transfer learning
  - Zero/one/few - shot learning

# Unsupervised Learning

- Biological argument [from G. Hinton]:
  - Our brains have 10^15 connections
  - We live for 10^9 secs
  - Need 10^6 bits/sec
  - Insufficient information from occasional high level label
  - Only source with enough information is input itself

- Challenging problem: big focus on many DL groups

# Historical Note

- Deep Learning revival started in ~2006
  - Hinton & Salakhudinov Science paper on RBMs
- Unsupervised Learning was focus from 2006-2012
- In ~2012 great results in vision, speech with supervised methods appeared
  - Less interest in unsupervised learning

# Overview of Unsupervised Approaches

- Given just data {X}
  - Unlike supervised learning there are no **provided** labels {Y}

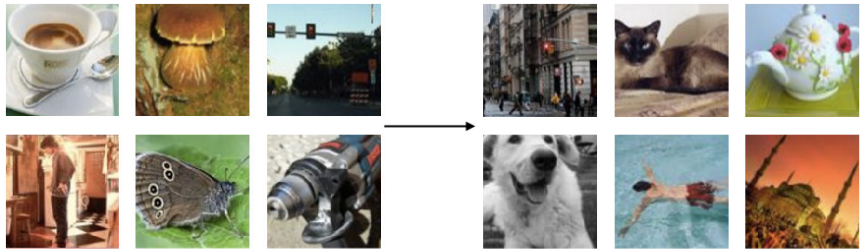
1. Density modeling, i.e. build model of p(X)
   - Enables sampling of new data
   - Evaluate probability of a data point
   - Can be conditional model, e.g. p(X_t | X_{t-1},…)
   - Requires (deep) generative architectures

# Density Modeling

- Have access to $x \sim p_{data}(x)$ through training set
- Want to learn a model $x \sim p_{model}(x)$
- Want $p_{model}$ to be similar to $p_{data}$:

Samples from true data distribution have high likelihood under $p_{model}$

Samples drawn from $p_{model}$ reflect structure of $p_{data}$



Training examples Model samples

# 2. "Self supervised" learning

- Find supervision signal y **within the input data**
- This signal is then used as a target:

$$y : \mathcal{X} \to \mathcal{Y}$$

$$x \mapsto y(x)$$

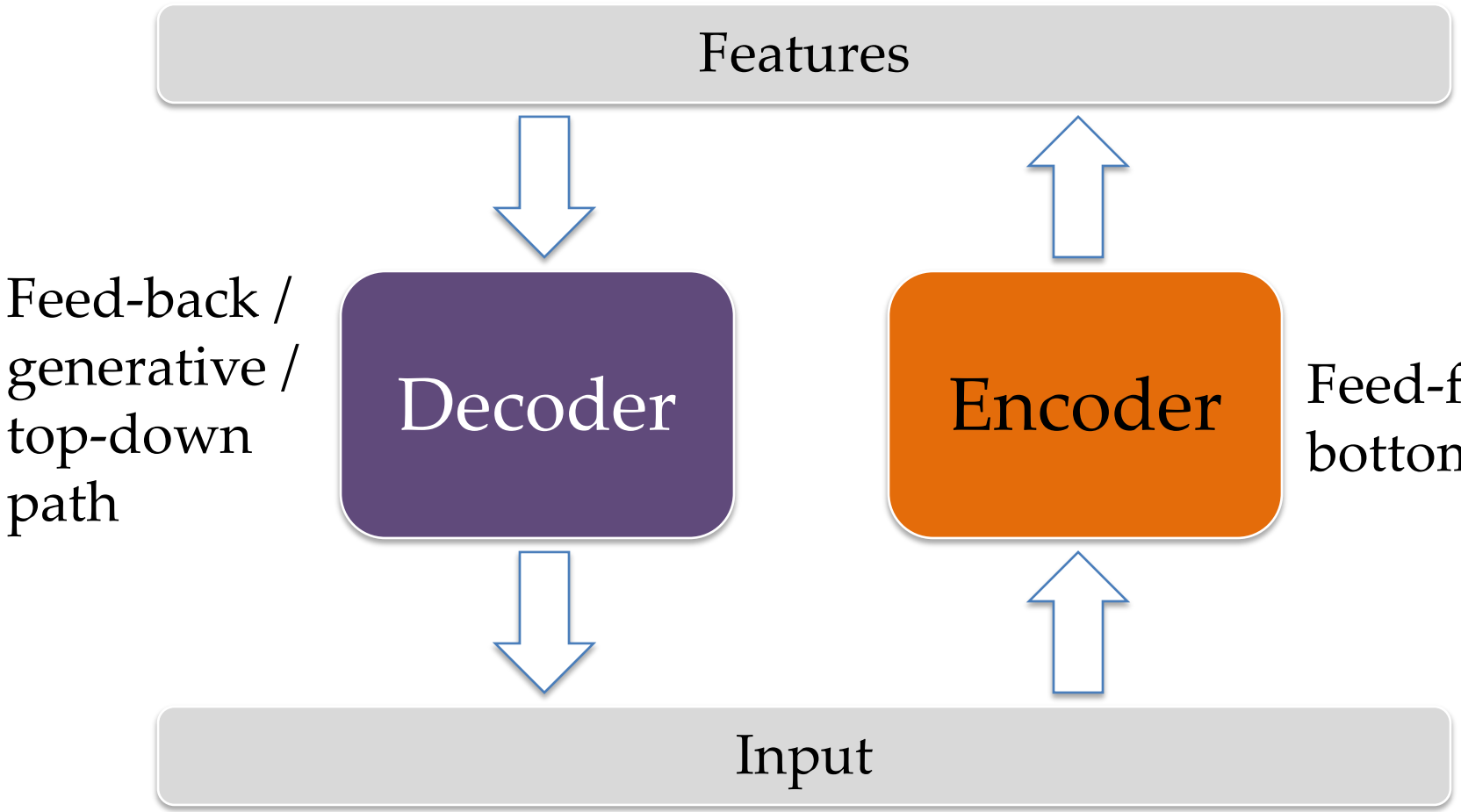
- Allows the use of standard supervised learning losses and architectures

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(f_\theta(x_i), y(x_i))$$

- Pre-training of representation for subsequent task
- Typically involves some insight into domain to pick y
- Inspired by word2vec (Mikolov et al. 2013)
  - E.g. The cat sat on the mat
  - X = {The, cat, NULL, on the mat}
  - Y = {sat}

# 1. Density Modeling of Natural Signals using Deep Learning

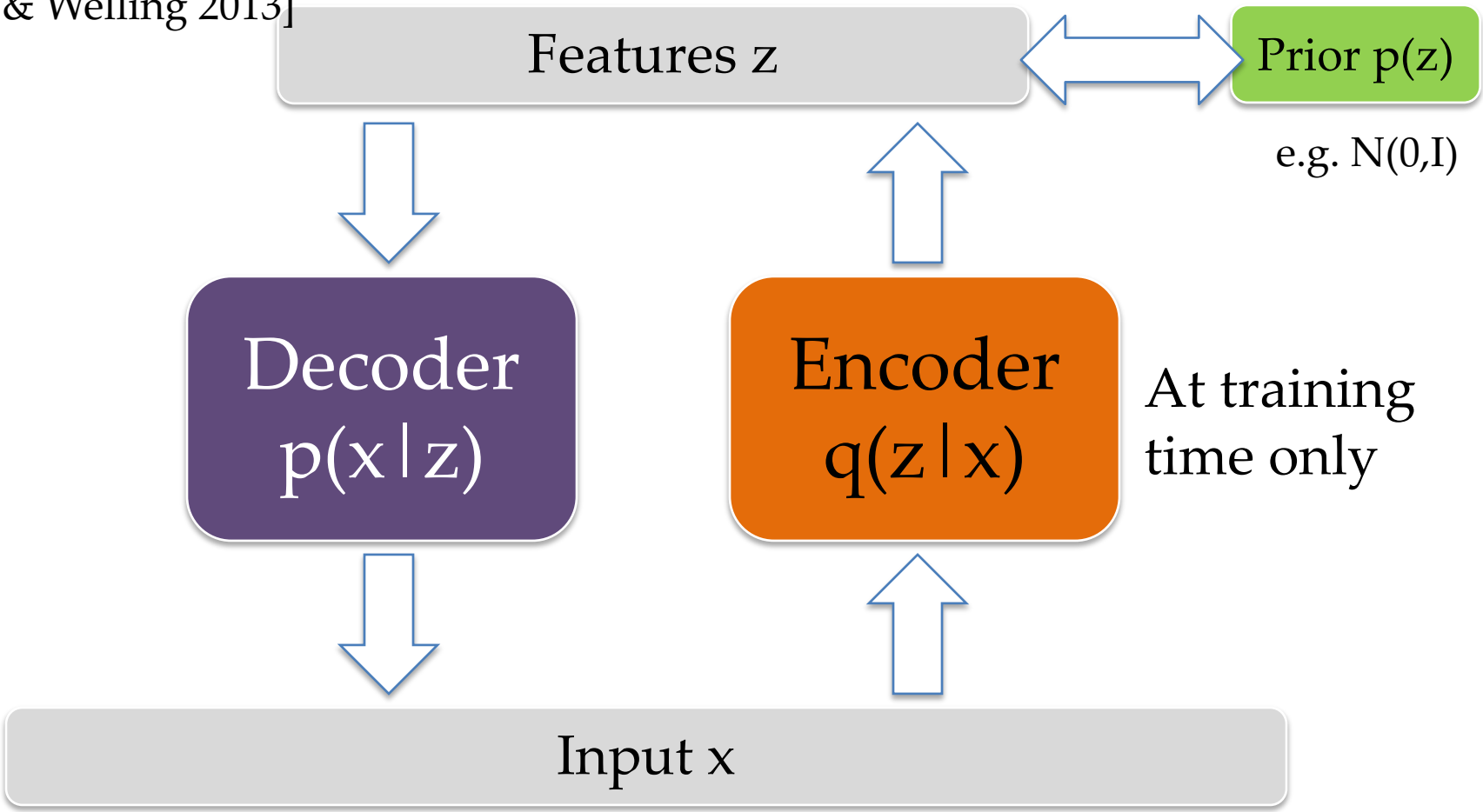
# Auto-Encoder




- Encoder/Decoder will be deep network
- Slightly different architectures for decoder (needs to output imag
- Architecture depends on application

# Variational Auto-Encoder

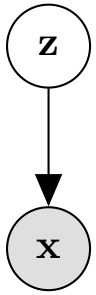
[Kingma & Welling 2013]




- Makes auto-encoder into a true generative model

$$\underbrace{\mathbb{E}_{q(z|x)} \log p(x|z)}_{\text{Reconstruction term}} - \underbrace{D_{KL}(q(z|x)||p(z))}_{\text{Prior term}}$$

# Directed graphical models




- We assume data is generated by:

$$z \sim p(z) \qquad x \sim p(x|z)$$

- $z$ is latent/hidden $x$ is observed (image)
- Use $\theta$ to denote parameters of the generative model

# Parameter estimation

- Given dataset $\{x_1, ..., x_n\}$, maximize likelihood of data under model:

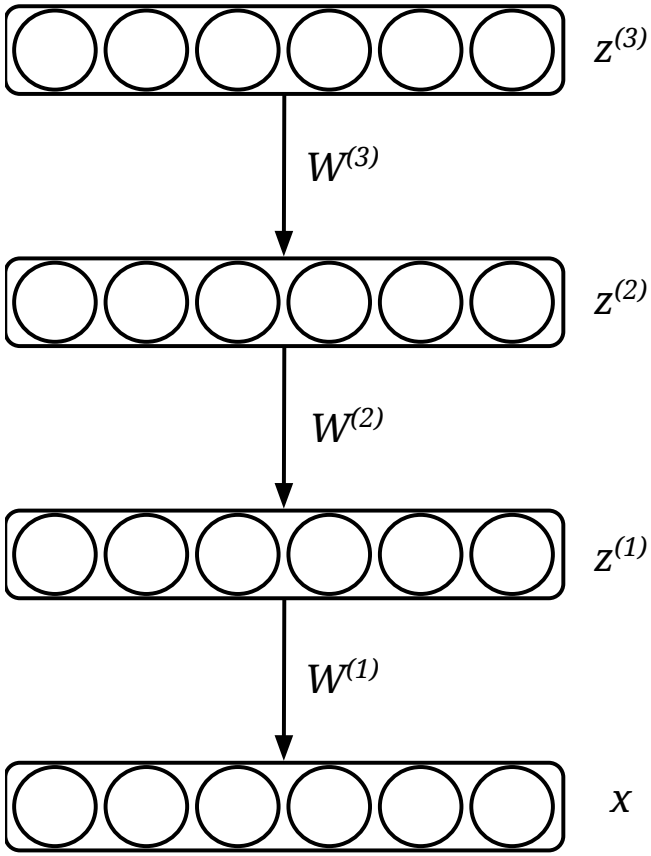
$$\max_\theta \sum_{i=1}^{n} \log p(x_i; \theta) = \max_\theta \sum_{i=1}^{n} \sum_z \log p(x_i, z; \theta)$$

- This quantity often intractable, difficult to optimize directly

- Can be optimized with iterative Expectation Maximization (EM) algorithm
  - Fix parameters and compute log likelihood wrt $p(z|x; \theta^t)$
  - Fix $z$ find parameters $\theta^{(t+1)}$ to maximize log likelihood

# Parameter estimation

- Standard EM requires access to posterior $p(z|x)$
- For the deep neural net models we care about this is infeasible
- Solution: introduce *variational* approximation $q(z;\phi)$ to $p(z|x)$
- Will give bound on log likelihood



# Bounding the marginal likelihood

Recall Jenson's inequality: When $f$ is concave, $f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$

$$
\begin{aligned}
\log p(x) &= \log \int_z p(x,z) \\
&= \log \int_z q(z) \frac{p(x,z)}{q(z)} \\
&\geq \int_z q(z) \log \frac{p(x,z)}{q(z)} = L(x;\theta,\phi) \qquad \text{(by Jensons inequality)} \\
&= \int_z q(z) \log p(x,z) - \int_z q(z) \log q(z) \\
&= \underbrace{\mathbb{E}_{q(z)}[\log p(x,z)]}_{\text{Expectation of joint distribution}} + \underbrace{\mathrm{H}(q(z))}_{\text{Entropy}}
\end{aligned}
$$

Bound is tight when variational approximation matches true posterior:

$$\log p(x) - L(x;\theta,\phi) = \log p(x) - \int_z q(z) \log \frac{p(x,z)}{q(z)}$$

Evidence Lower BOund (ELBO)

$$\begin{aligned}
&= \int_z q(z) \log p(x) - \int_z q(z) \log \frac{p(x,z)}{q(z)} \\
&= \int_z q(z) \log \frac{q(z)p(x)}{p(x,z)} \\
&= \int_z q(z) \log \frac{q(z)}{p(z|x)} \\
&= D_{KL}(q(z;\phi)||p(z|x))
\end{aligned}$$

# Learning directed graphical models

- Maximize bound on likelihood of data:

$$\max_{\theta} \sum_{i=1}^{N} \log p(x_i; \theta) \geq \max_{\theta, \phi_1, \ldots, \phi_N} \sum_{i=1}^{N} L(x_i; \theta, \phi_i)$$

- Historically, used different $\phi_i$ for every data point
  - But we'll move away from this soon..
- Can still use EM style algorithm to iteratively optimize
- For more info see Blei *et al.* (2003)

# New method of learning: approximate inference model

- Instead of having different variational parameters for each data point, fit a conditional parametric function
- The output of this function will be the parameters of the variational distribution $q(z|x)$
- Instead of q(z) we have $q_\phi(z|x)$

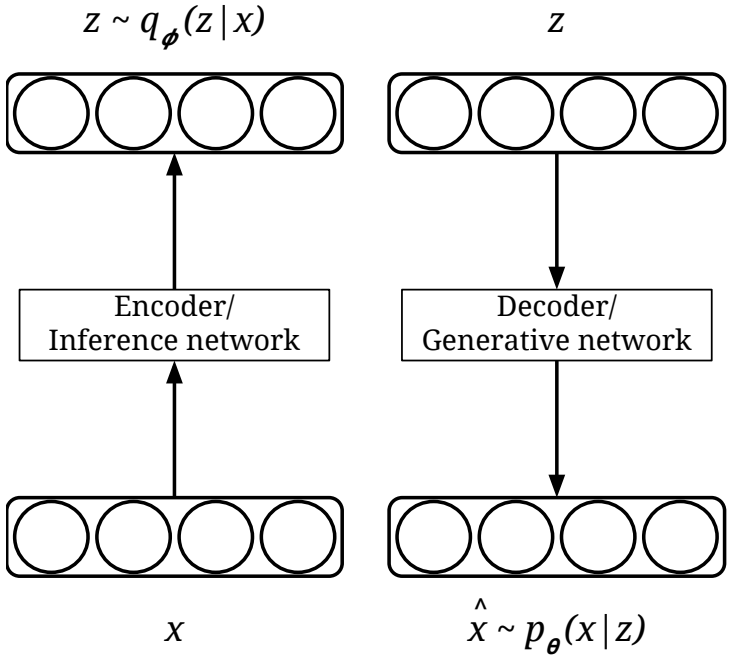
Evidence Lower BOund (ELBO)

- ELBO becomes:

$$L(x;\theta,\phi) = \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x,z)]}_{\text{Expectation of joint distribution}} + \underbrace{\mathrm{H}(q_\phi(z|x))}_{\text{Entropy}}$$

# Variational autoencoder

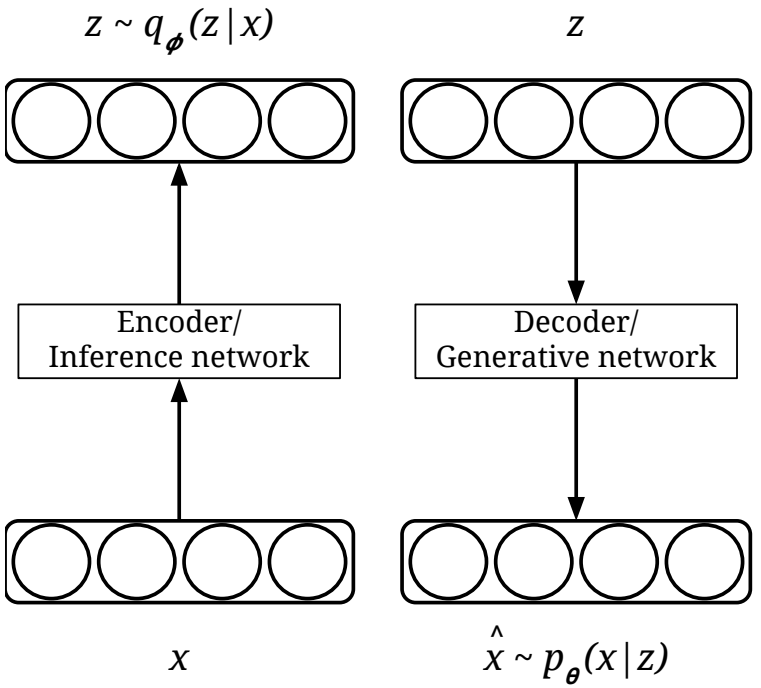
- *Encoder* network maps from image space to latent space
  - Outputs parameters of $q_\phi(z|x)$
- *Decoder* maps from latent space back into image space
  - Outputs parameters of $p_\theta(x|z)$




[Kingma & Welling (2013)]

# Example

- *Encoder* network outputs mean and variance of Normal distribution
  - $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$
- *Decoder* network outputs mean (and optionally variance) of Normal distribution
  - $p_\theta(x|z) = \mathcal{N}(\mu_\theta(z), \mathbf{I})$




[Kingma & Welling (2013)]

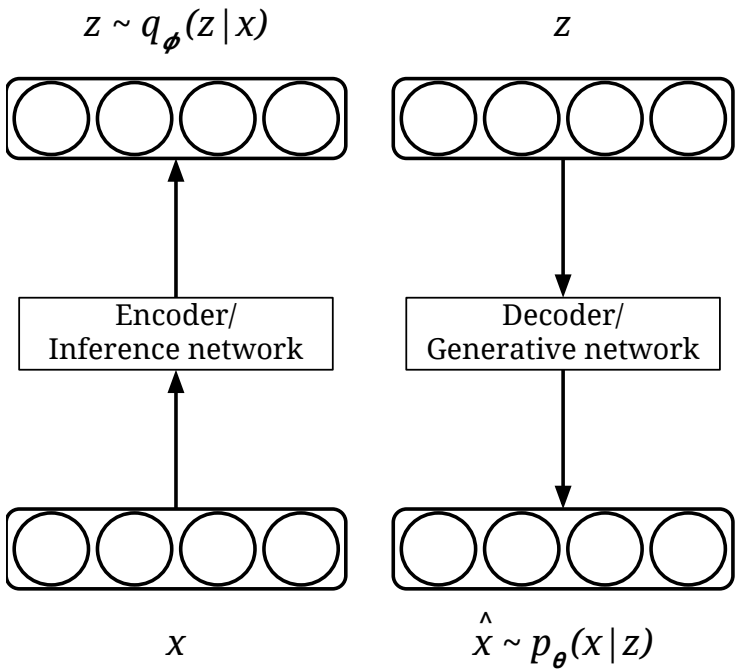
# Variational autoencoder

- Rearranging the ELBO:

$$
\begin{aligned}
L(x;\theta,\phi) &= \int_z q(z|x) \log \frac{p(x,z)}{q(z|x)} \\
&= \int_z q(z|x) \log \frac{p(x|z)p(z)}{q(z|x)} \\
&= \int_z q(z|x) \log p(x|z) + \int_z q(z|x) \log \frac{p(z)}{q(z|x)} \\
&= \mathbb{E}_{q(z|x)} \log p(x|z) - \mathbb{E}_{q(z|x)} \log \frac{q(z|x)}{p(z)} \\
&= \underbrace{\mathbb{E}_{q(z|x)} \log p(x|z)}_{\text{Reconstruction term}} - \underbrace{D_{KL}(q(z|x)||p(z))}_{\text{Prior term}}
\end{aligned}
$$

# Variational autoencoder

- Inference network outputs parameters of $q_\phi(z|x)$
- Generative network outputs parameters of $p_\theta(x|z)$
- Optimize $\theta$ and $\phi$ jointly by maximizing ELBO:




$$L(x;\theta,\phi) = \underbrace{\mathbb{E}_{q(z|x)} \log p(x|z)}_{\text{Reconstruction term}} - \underbrace{D_{KL}(q(z|x)||p(z))}_{\text{Prior term}}$$

# Stochastic gradient variation bayes (SGVB) estimator

- Reparameterization trick : re-parameterize $z \sim q_\phi(z|x)$ as

$$z = g_\phi(x, \epsilon) \text{ with } \epsilon \sim p(\epsilon)$$

- For example, with a Gaussian can write $z \sim \mathcal{N}(\mu, \sigma^2)$ as

$$z = \mu + \epsilon\sigma^2 \text{ with } \epsilon \sim \mathcal{N}(0, 1)$$

[Kingma & Welling (2013); Rezende *et al.* (2014)]

# Stochastic gradient variation bayes (SGVB) estimator

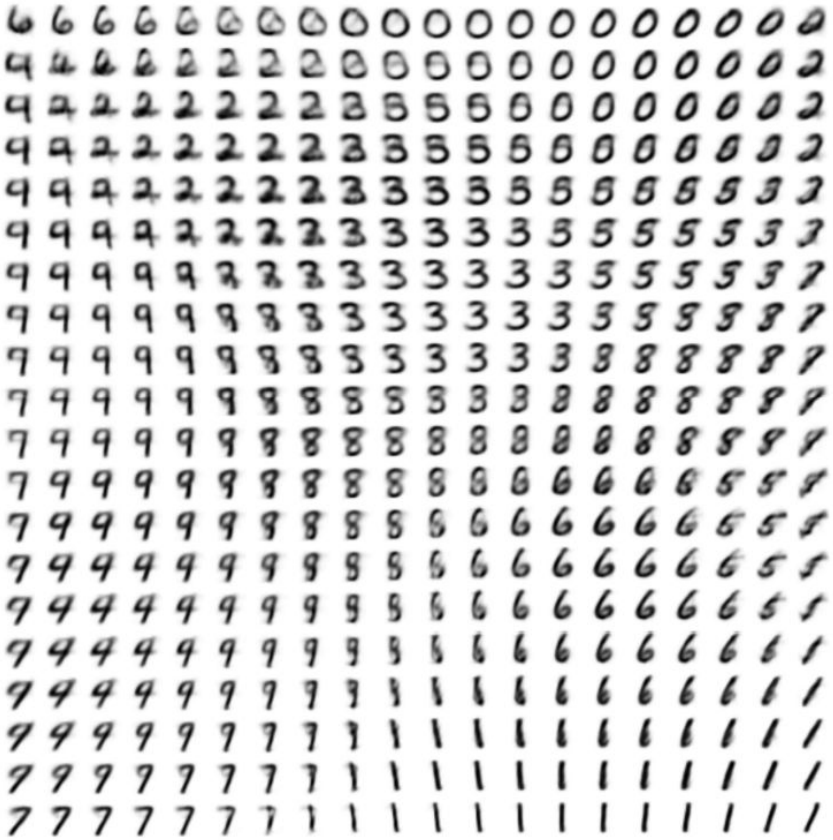
$$L(x;\theta,\phi) = \underbrace{\mathbb{E}_{q(z|x)} \log p(x|z)}_{\text{Reconstruction term}} - \underbrace{D_{KL}(q(z|x)||p(z))}_{\text{Prior term}}$$

- Using reparameterization trick we form Monte Carlo estimate of reconstruction term:

$$\mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z) = \mathbb{E}_{p(\epsilon)} \log p_\theta(x|g_\phi(x,\epsilon))$$
$$\simeq \frac{1}{L} \sum_{i=1}^{L} \log p_\theta(x|g_\phi(x,\epsilon)) \quad \text{where } \epsilon \sim p(\epsilon)$$

- KL divergence term can often be computed analytically (eg. Gaussian)

# VAE learned manifold



[Kingma & Welling (2013)]

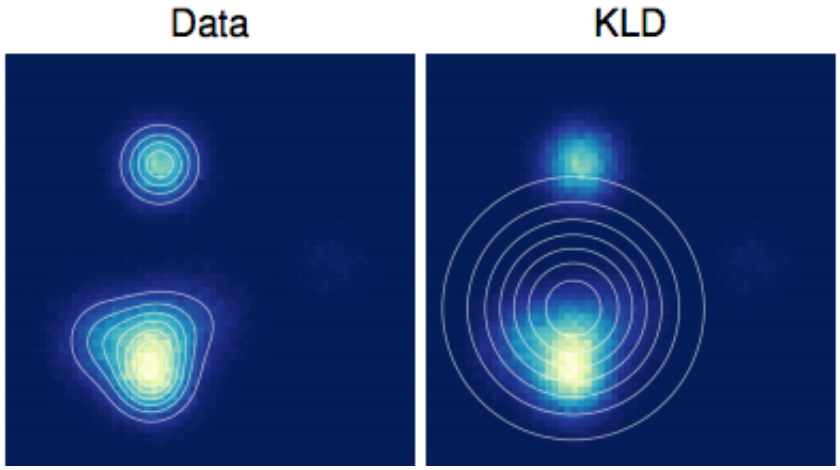
# VAE samples

(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

[Kingma & Welling (2013)]

# VAE tradeoffs

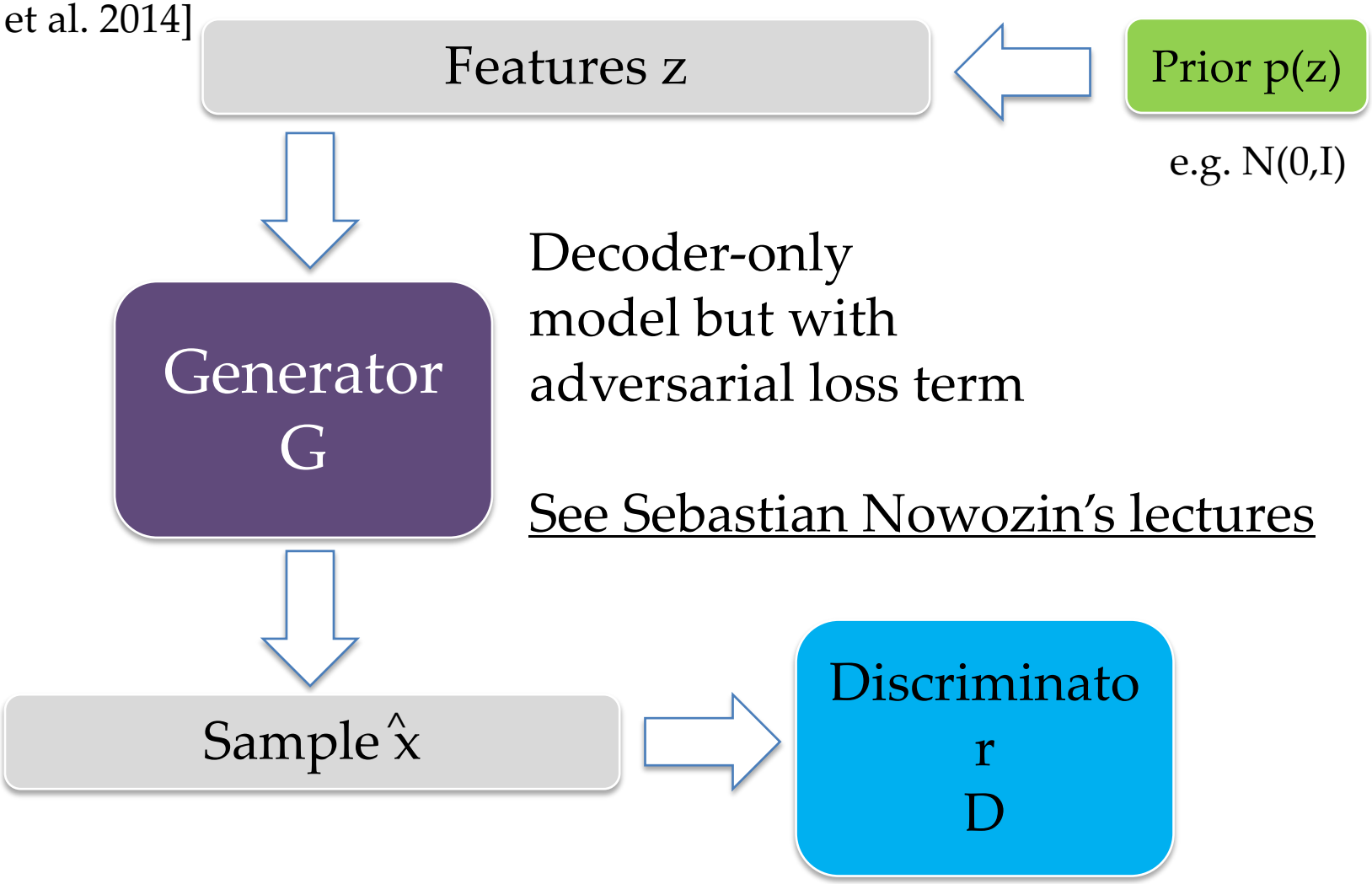
- Pros:
  - Theoretically pleasing
  - Optimizes bound on likelihood
  - Easy to implement
- Cons:
  - Samples tend to be blurry
    - Maximum likelihood minimizes $D_{KL}(p_{data}||p_{model})$




[Theis *et al.* (2016)]

# Generative Adversarial Networks

[Goodfellow et al. 2014]



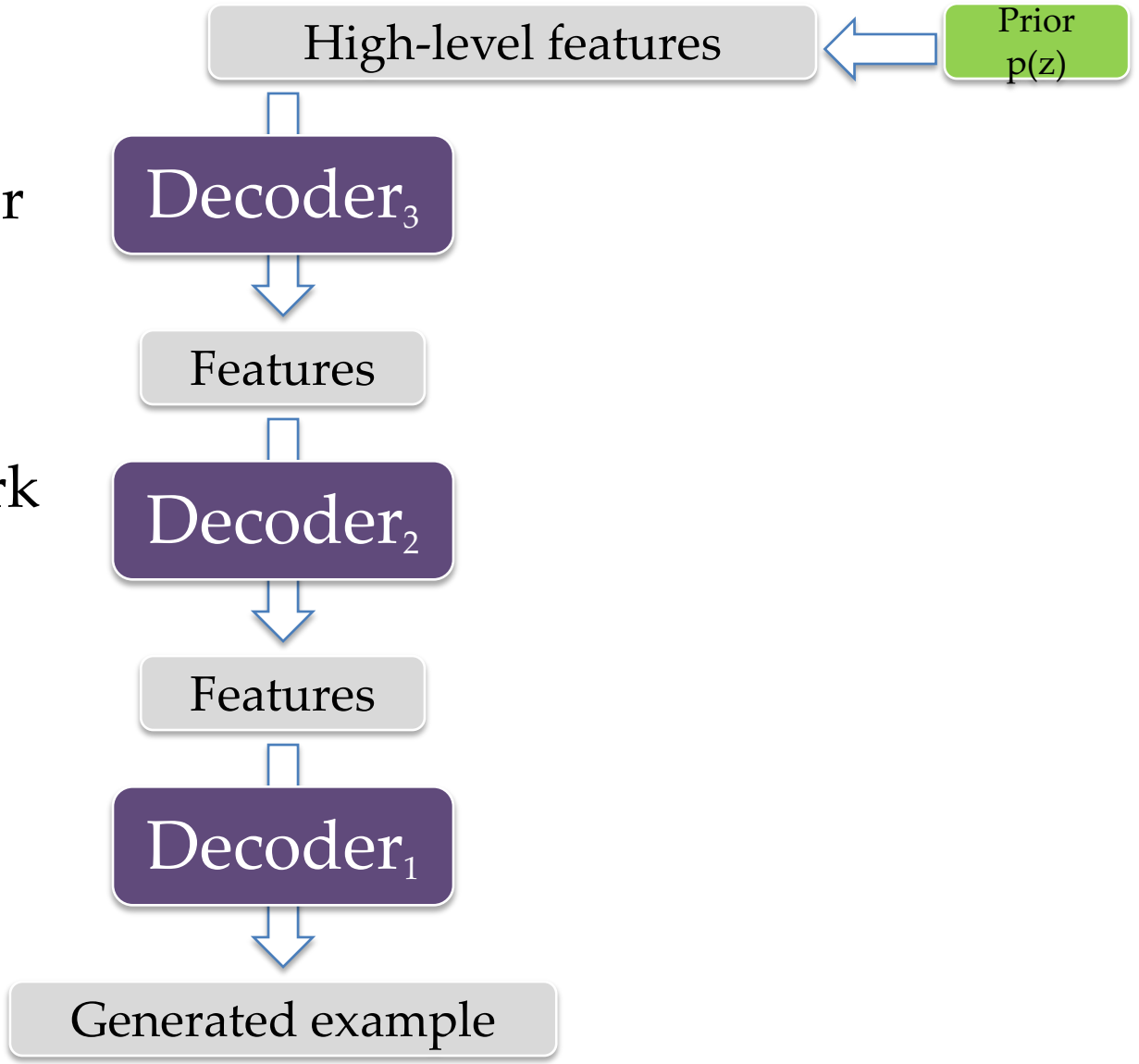

- Mini—max game between G and D

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

# GAN - Generating Samples

- Sample from prior p(z)

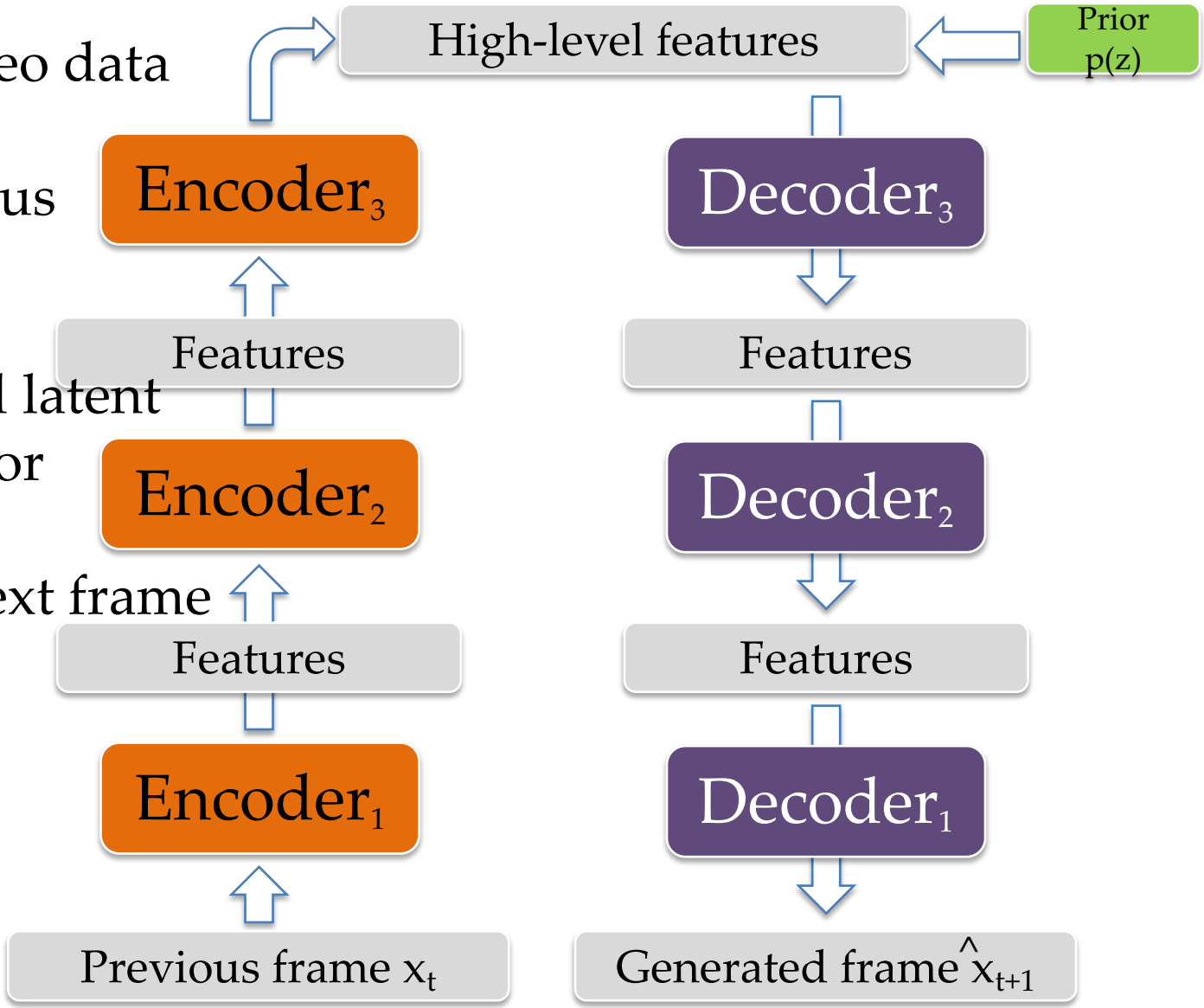
- Push through decoder network



# Conditional Generation

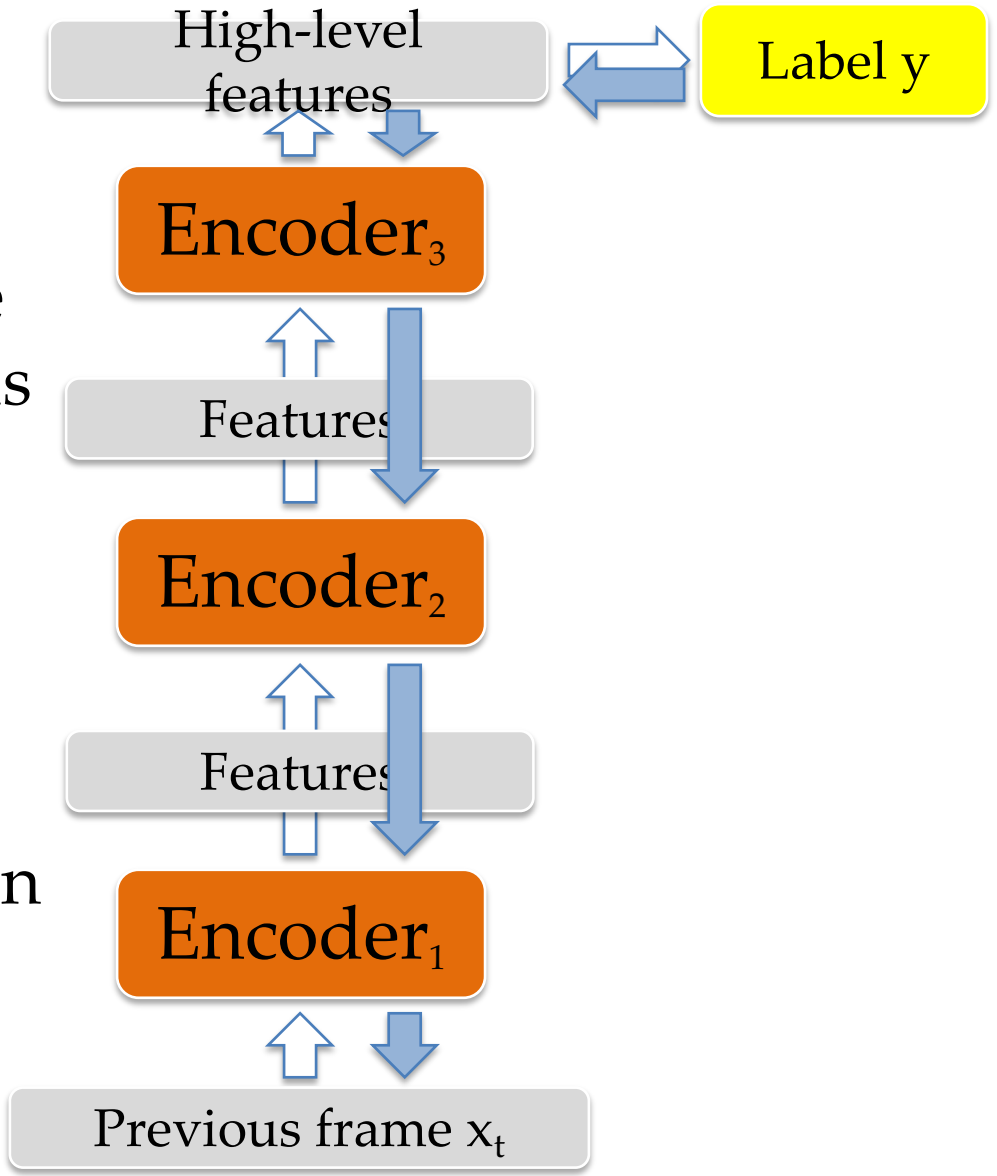
E.g. consider video data

- Encode previous frames(s)
- [Optional] add latent noise from prior
- Reconstruct next frame via decoder



# Supervised Fine-Tuning

- After unsupervised "pre-training", refine model with few labels on target task
- Unsupervised training phase learns "good" representation

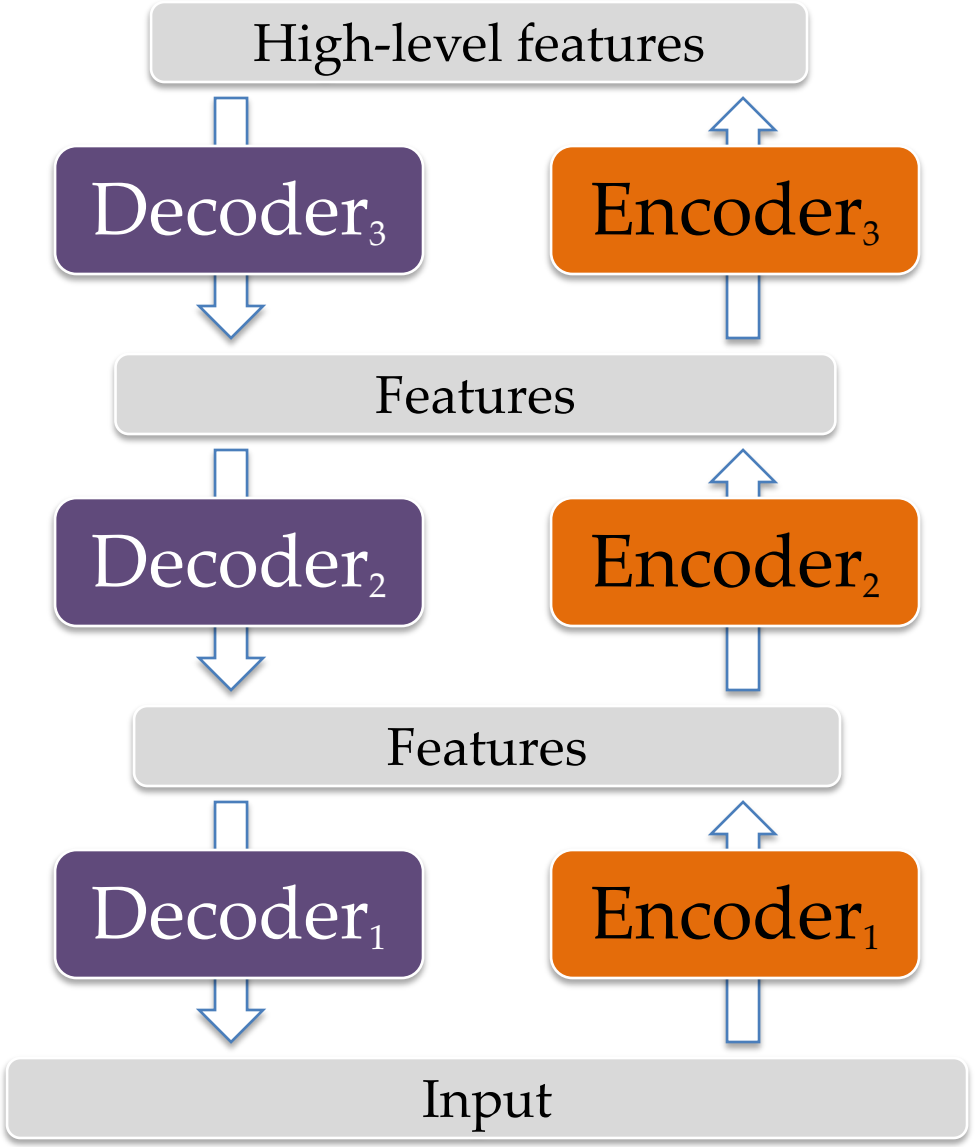


# Stacked Auto-Encoders

- Ladder Networks
  [Rasmus et al. 2015]
  - Reconstruction constraint at each layer
  - Trained end-to-end

- Can be trained layer-wise
  - Stacked RBMs

  [Hinton & Salakhutdinov 2006]



# Many Others Approaches

- Autoencoder (most unsupervised Deep Learning methods)
  - Restricted / Deep Boltzmann Machines
  - Denoising autoencoders
  - Predictive sparse decomposition
- Decoder-only
  - Sparse coding & hierarchical variants

# Autoregressive models

- Tractably model a joint distribution of the pixels in the image
- Learn to predict the next pixel given all the previously generated pixels
- Joint distribution of all pixels just product of conditionals:

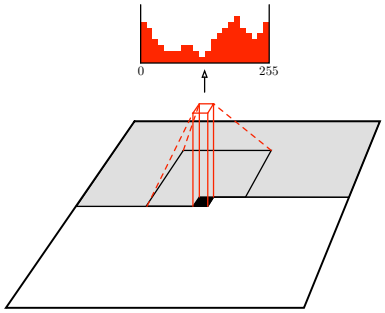
$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1})$$

# Pixel-CNN

[van den Oord et al., arXiv 1606.05328, 2016]

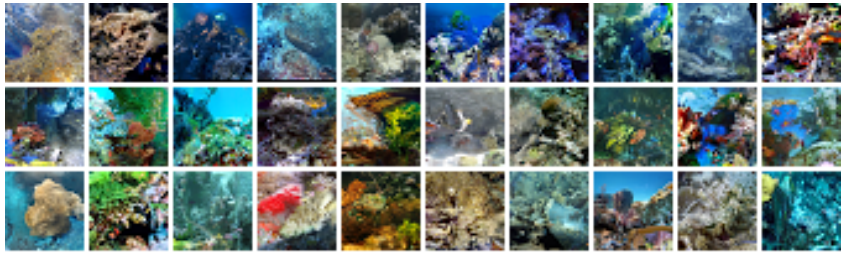
- Conditional generative model of images
- Generate each pixel, in raster-scan order
- Just predict distribution over a single pixel (can be multi-modal)
- See also Video Pixel Networks [Kalchbrenner et al., 2016],
- NADE [Larochelle & Murray 2011] & RIDE [Theis and Bethge, NIPS 2015].

$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1}).$$




| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

African elephant

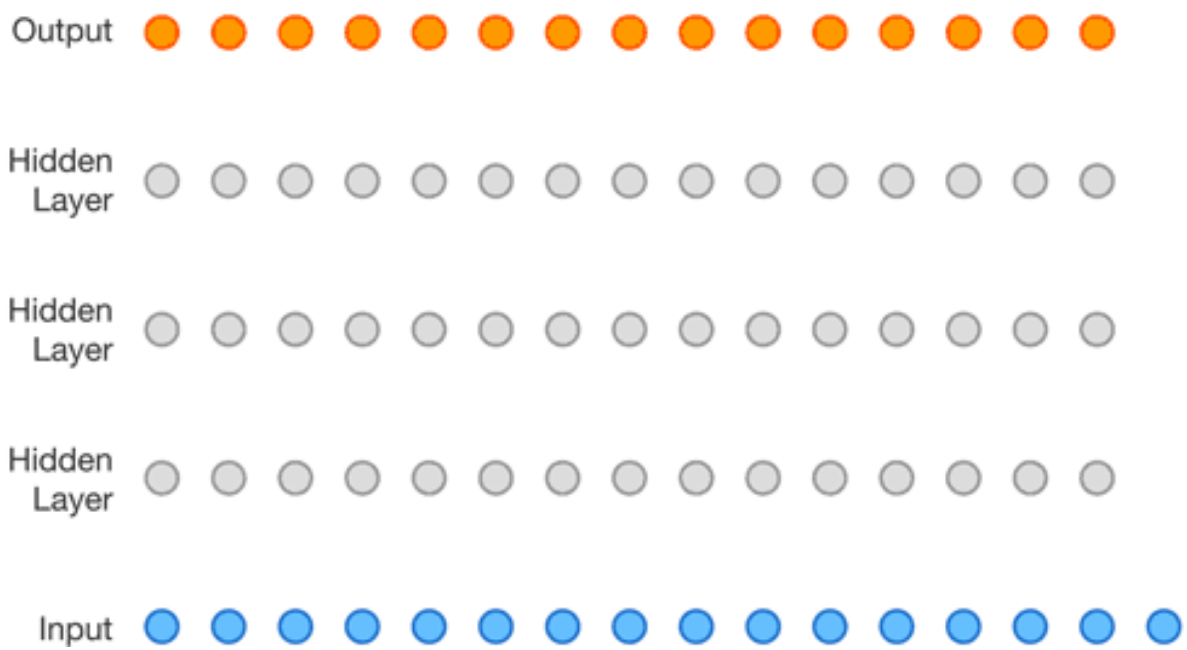


Coral Reef

# Wavenet

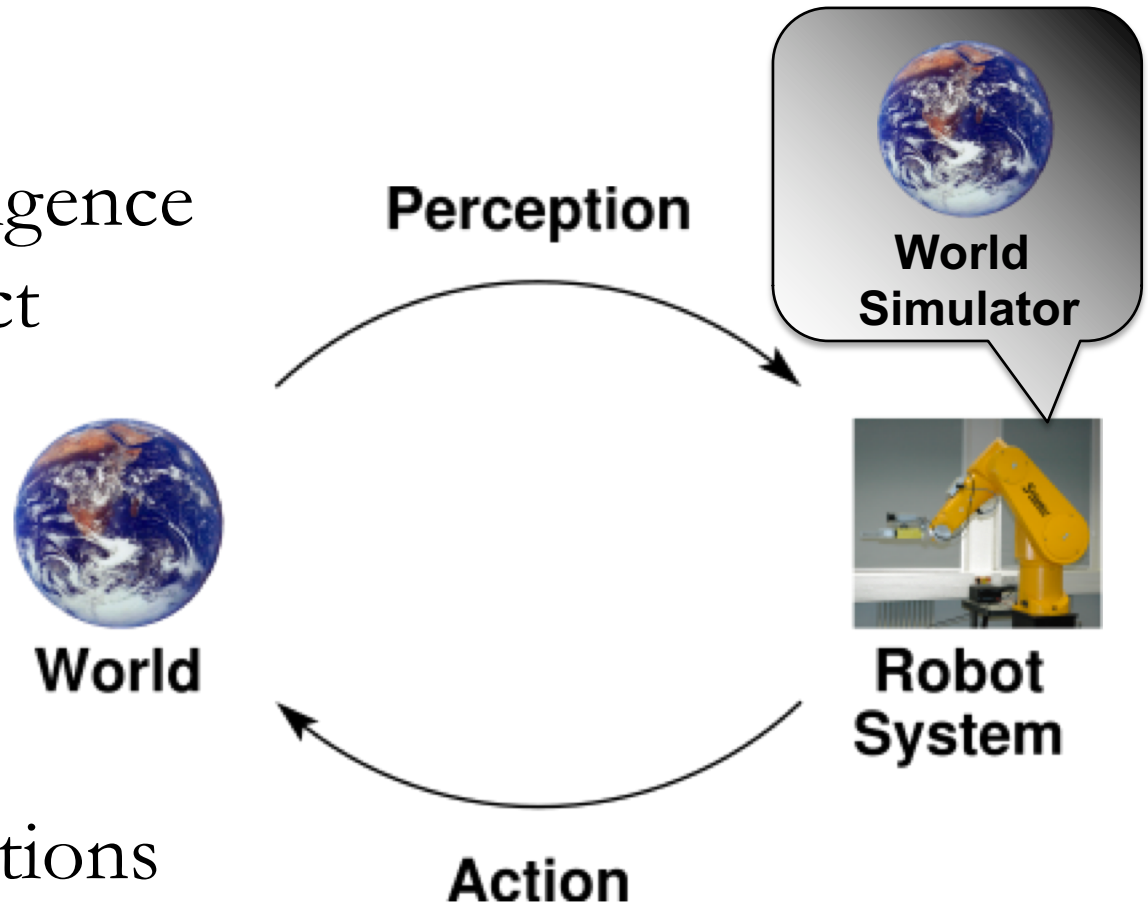
[van den Oord et al., arXiv 1609.03499, 2016]

- Generative model of raw speech waveform
- Condition on previous parts of waveform $p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$
- Dilated causal convolution layers
- Discrete output distribution (use softmax)



# An aside: Reasoning and Planning in World

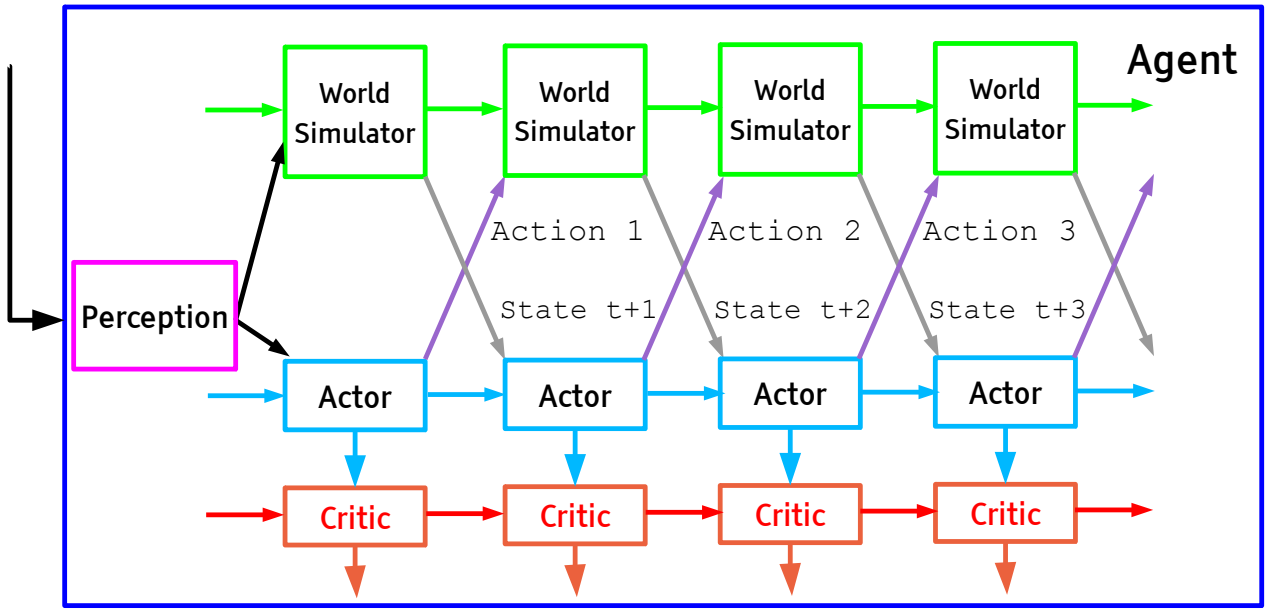
- Solving AI requires more than just perception
- Essence of intelligence is ability to predict
- To plan ahead need to simulate world & reason about possible actions




[Craik 1943; Dennett 1978; Sutton & Barto 1981; Sutton 1991]

# Planning in the World

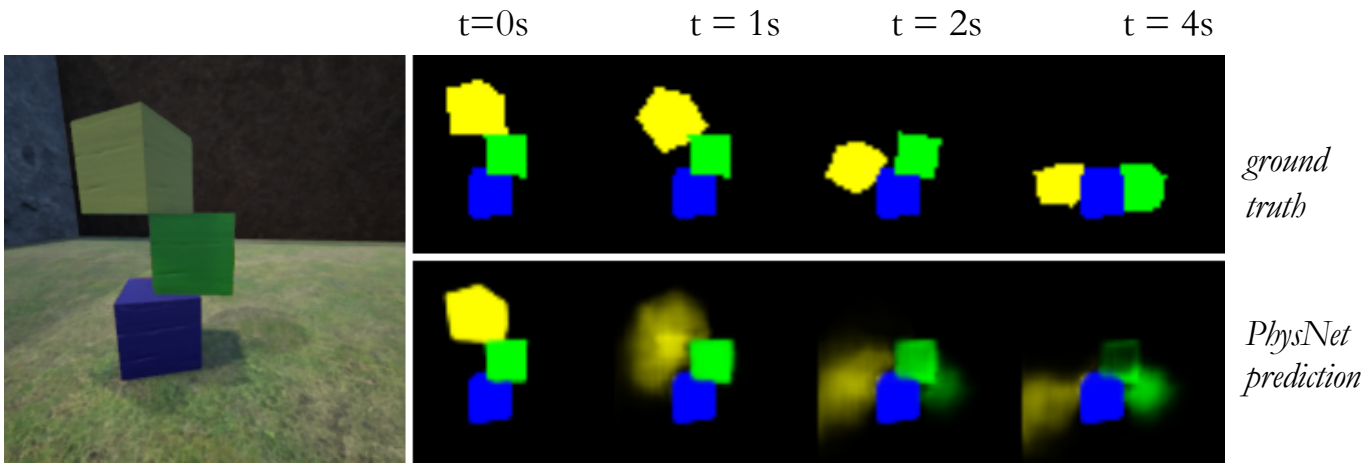
- Try out different action sequences in mind of robot/agent
- Need accurate world simulator




[Yann LeCun]

# Video Prediction

- Predict pixels of next frame, given previous ones
- Enables learning of world model/simulator
- Challenging due to inherent uncertainty in the dynamics of the world
  - Pixel wise loss functions can cause blurring due to multiple futures being accounted for




[Lerer, Gross, Fergus ICML 2016]

# Video Prediction

- Lots of prior work, e.g.:
  - LSTMs: Srivastava et al. (2015); Finn et al. (2016)
  - Discrete latent variables: Ranzato et al. (2014)
  - Optical flow: Xue et al. (2016); Walker et al. (2015)
  - Action-conditional: Chiappa et al. (2017) and Oh et al. (2015)


[Mathieu, Couprie, LeCun, ICLR 2016]

# Handling Uncertainty

- Video prediction is challenging due to inherent uncertainty in the dynamics of the world
- Pixel wise loss functions can cause blurring due to multiple futures being accounted for

- Two broad approaches:
  - GANs (Mathieu et al. 2015; Vondrick et al. 2016)
  - Latent variables (Henaff et al., 2017; Babaeizadeh et al. 2018; Denton & Fergus 2018)

# Stochastic Video Generation with a Learned Prior

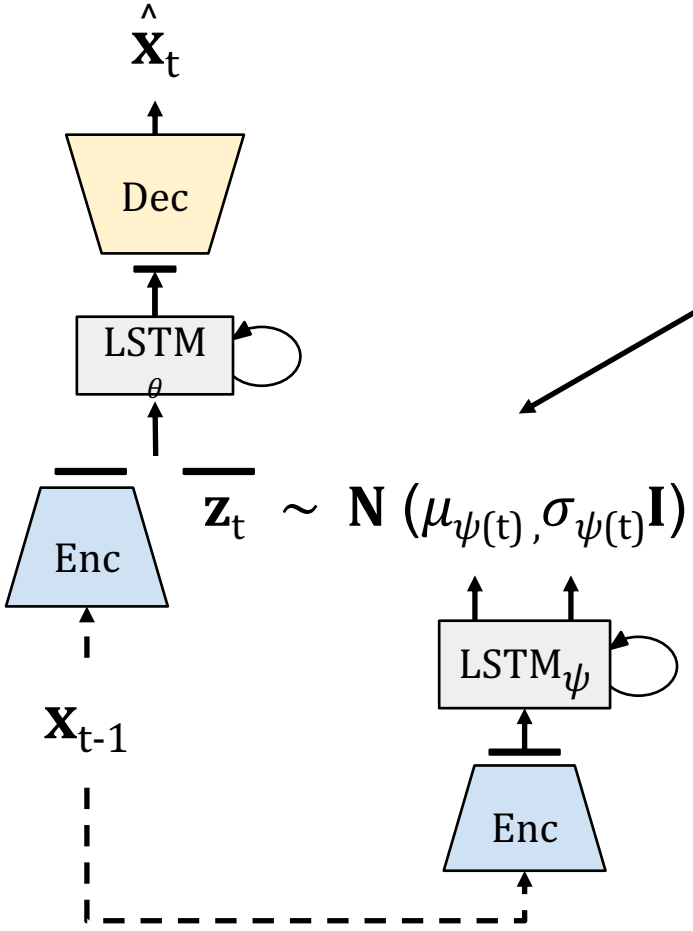
ICML 2018

Emily Denton[1] and Rob Fergus[12]

1


2

# Stochastic video generation, Denton & Fergus 2018

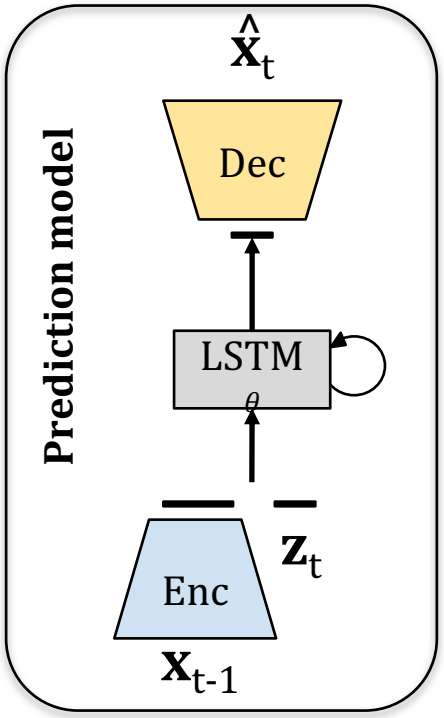



Learned prior dependent on all previous frames

- Combines a deterministic frame predictor with time-dependent stochastic latent variables
- Learned prior over the latent variables can can be interpreted as a predictive model of uncertainty

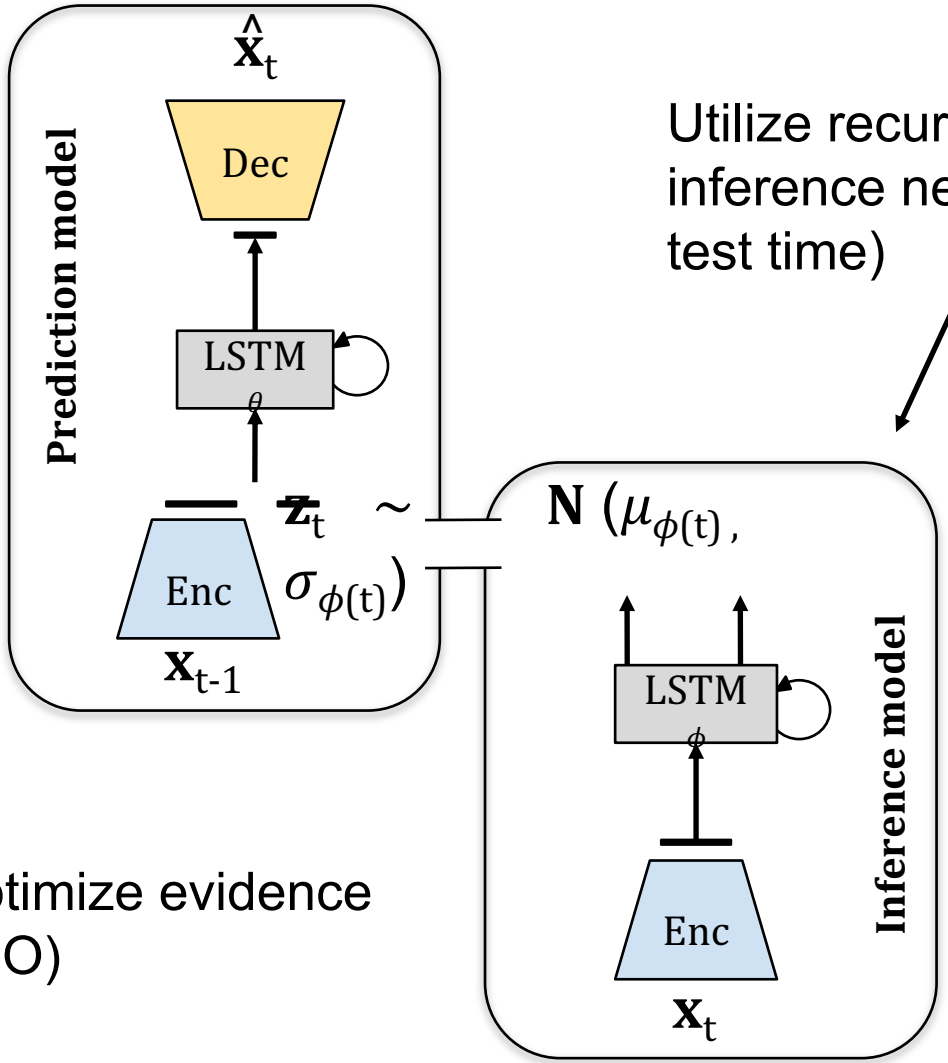
[Denton et al. ICML 2018]

**Training at time *t*:**




[Denton et al. ICML 2018]

**Training at time *t*:**

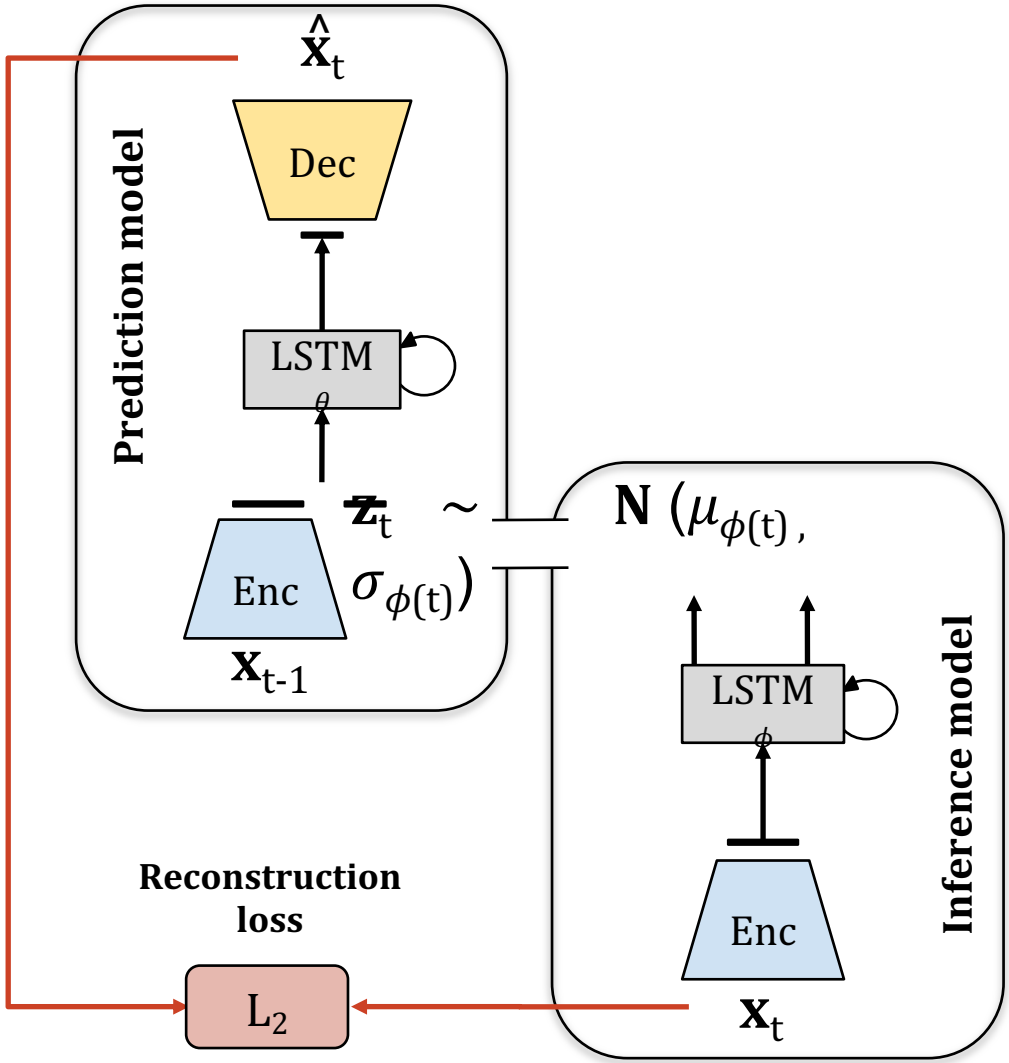



Utilize recurrent approximate inference network (not used at test time)

Train model by optimize evidence lower bound (ELBO)

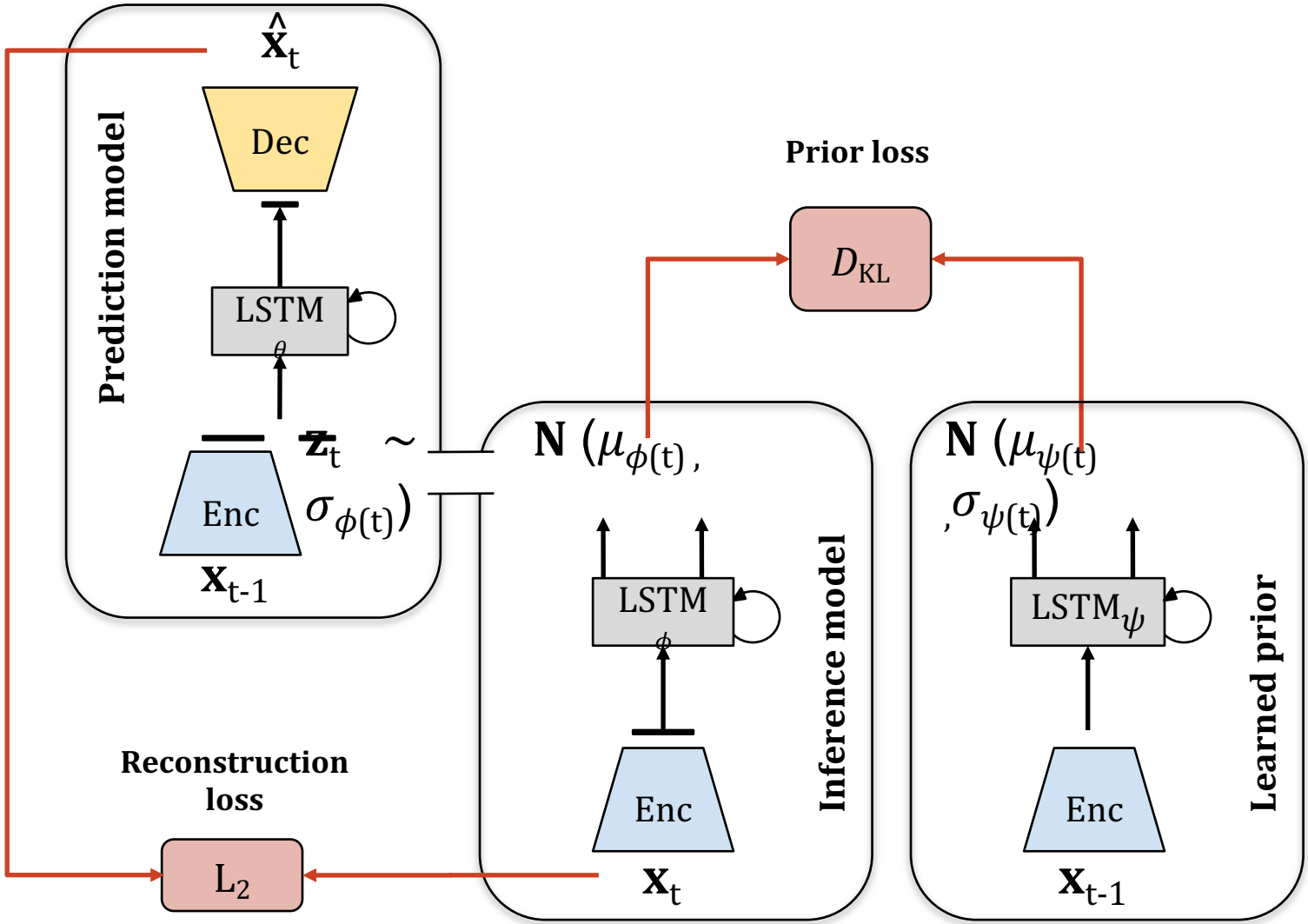
[Denton et al. ICML 2018]

**Training at time *t*:**



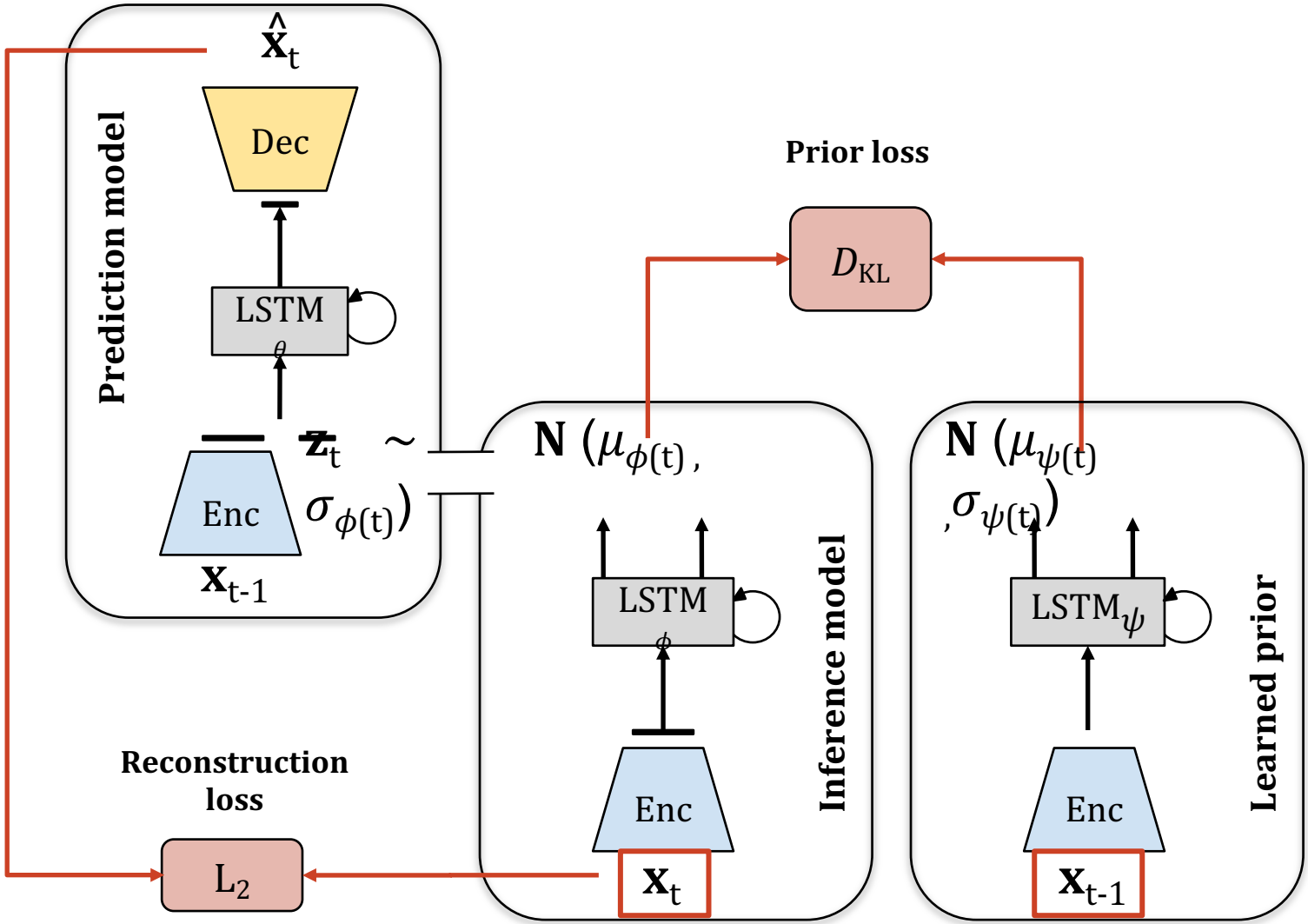

[Denton et al. ICML 2018]

# Training at time *t*:



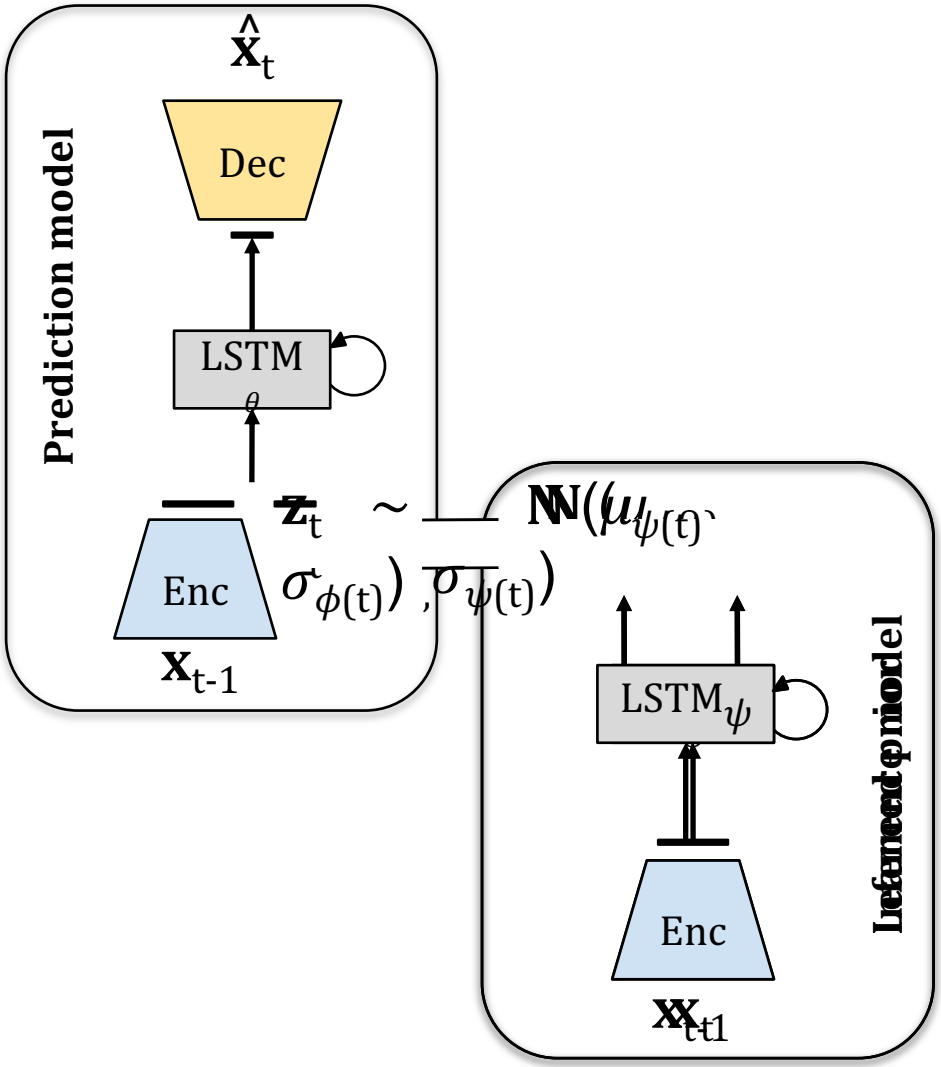

[Denton et al. ICML 2018]

# Training at time *t*:



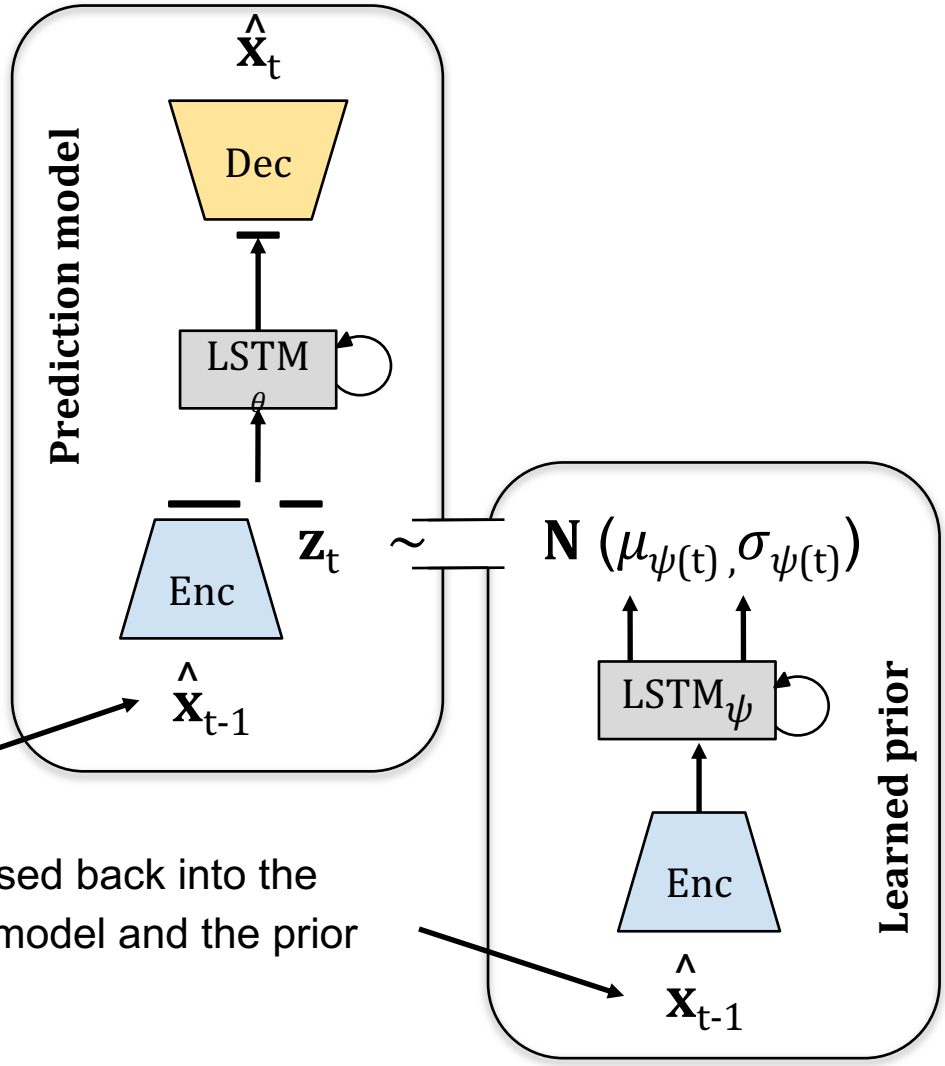

[Denton et al. ICML 2018]

**Generation at time *t*:**




[Denton et al. ICML 2018]

# Generation at time *t*:




Generated frames passed back into the input of the prediction model and the prior

[Denton et al. ICML 2018]

## Babaeizadeh et al. (2018)

***Inference***

Feed forward net encodes entire video sequence:

$$\boldsymbol{z}_t \sim q_\phi(\boldsymbol{z} \mid \boldsymbol{x}_{1:T})$$

## Denton et al. (2018)

***Inference***

Recurrent net produces different distribution for every *t*:

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \prod_t q_\phi^{(t)}(\mathbf{z}_t \mid \mathbf{x}_{1:t})$$

$$\mathbf{z}_t \sim q_\phi^{(t)}(\mathbf{z}_t \mid \mathbf{x}_{1:t})$$

[*Babaeizadeh et al. Stochastic variational video prediction. ICLR, 2018.*]

[Denton et al. ICML 2018]

## Babaeizadeh et al. (2018)

***Inference***

Feed forward net encodes entire video sequence:

$$\boldsymbol{z}_t \sim q_\phi(\boldsymbol{z} | \boldsymbol{x}_{1:T})$$

***Generation***

$$\boldsymbol{z}_t \sim \mathbf{N}(\mathbf{0}, \mathbf{I})$$

## Denton et al. (2018)

***Inference***

Recurrent net produces different distribution for every *t*:

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \prod_t q_\phi^{(t)}(\mathbf{z}_t \mid \mathbf{x}_{1:t})$$

$$\mathbf{z}_t \sim q_\phi^{(t)}(\mathbf{z}_t \mid \mathbf{x}_{1:t})$$

***Generation***

$$\mathbf{z}_t \sim \mathbf{N}(\mu_\psi(\mathbf{x}_{1:t-1}), \sigma_\psi(\mathbf{x}_{1:t-1})\mathbf{I})$$

[*Babaeizadeh et al. Stochastic variational video prediction. ICLR, 2017.*]

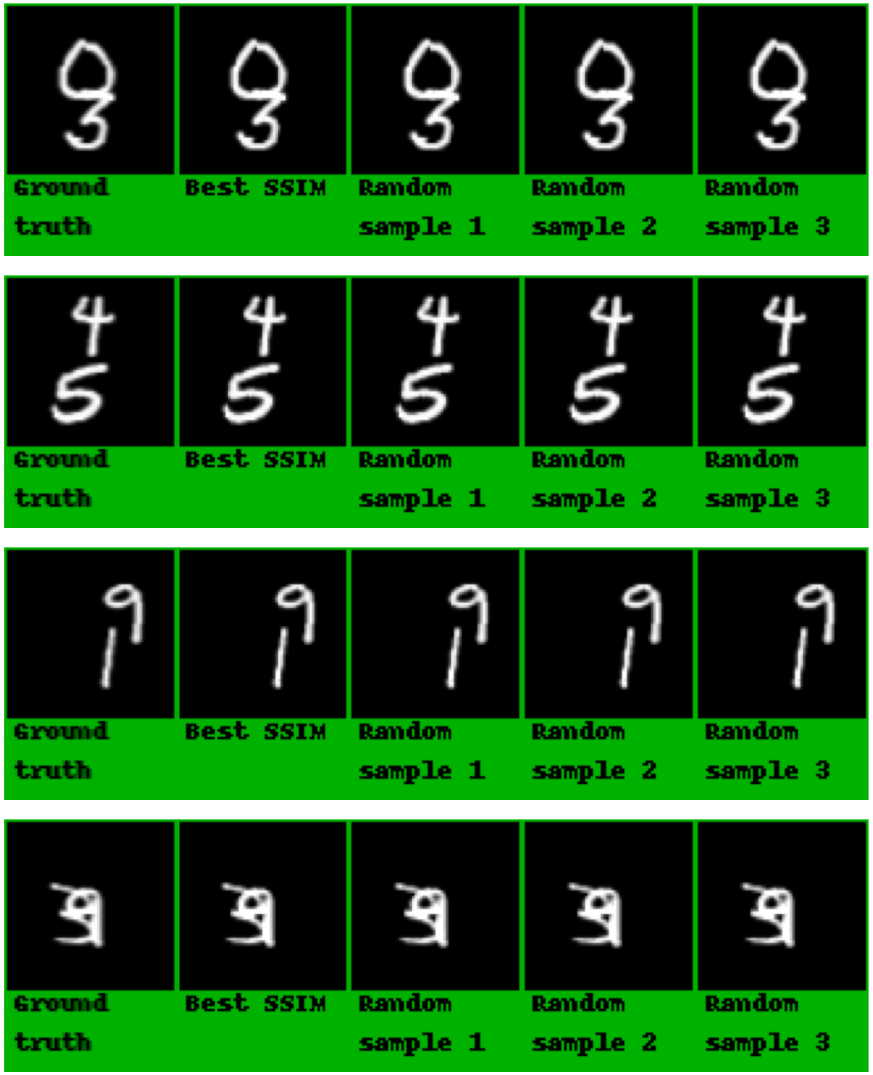
[Denton et al. ICML 2018]

# Stochastic Moving MNIST

- Stochastic variant of the Moving MNIST dataset (*Srivastava et al.*, 2015)

**Green: Ground truth input**
**Red: generated frames**

- Model conditioned on 5 frames and trained to predict next 10 frames
- Best SSIM chosen from 100 samples




[Denton et al. ICML 2018]

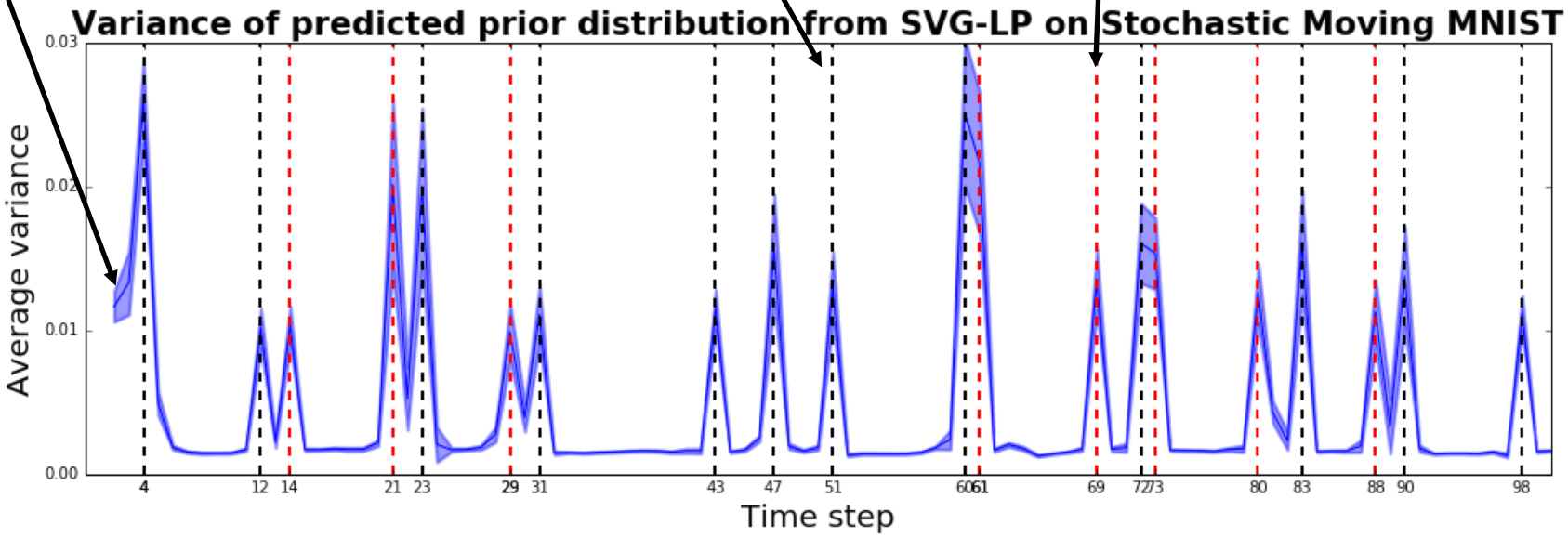
# Learned prior can be interpreted as a model of uncertainty

Prior predicts *low variance distribution for deterministic parts of the video*, *high variance distribution as points of uncertainty*

Predicted variance from our models learned prior

**Black dashed lines:**
digit 1 collides with wall

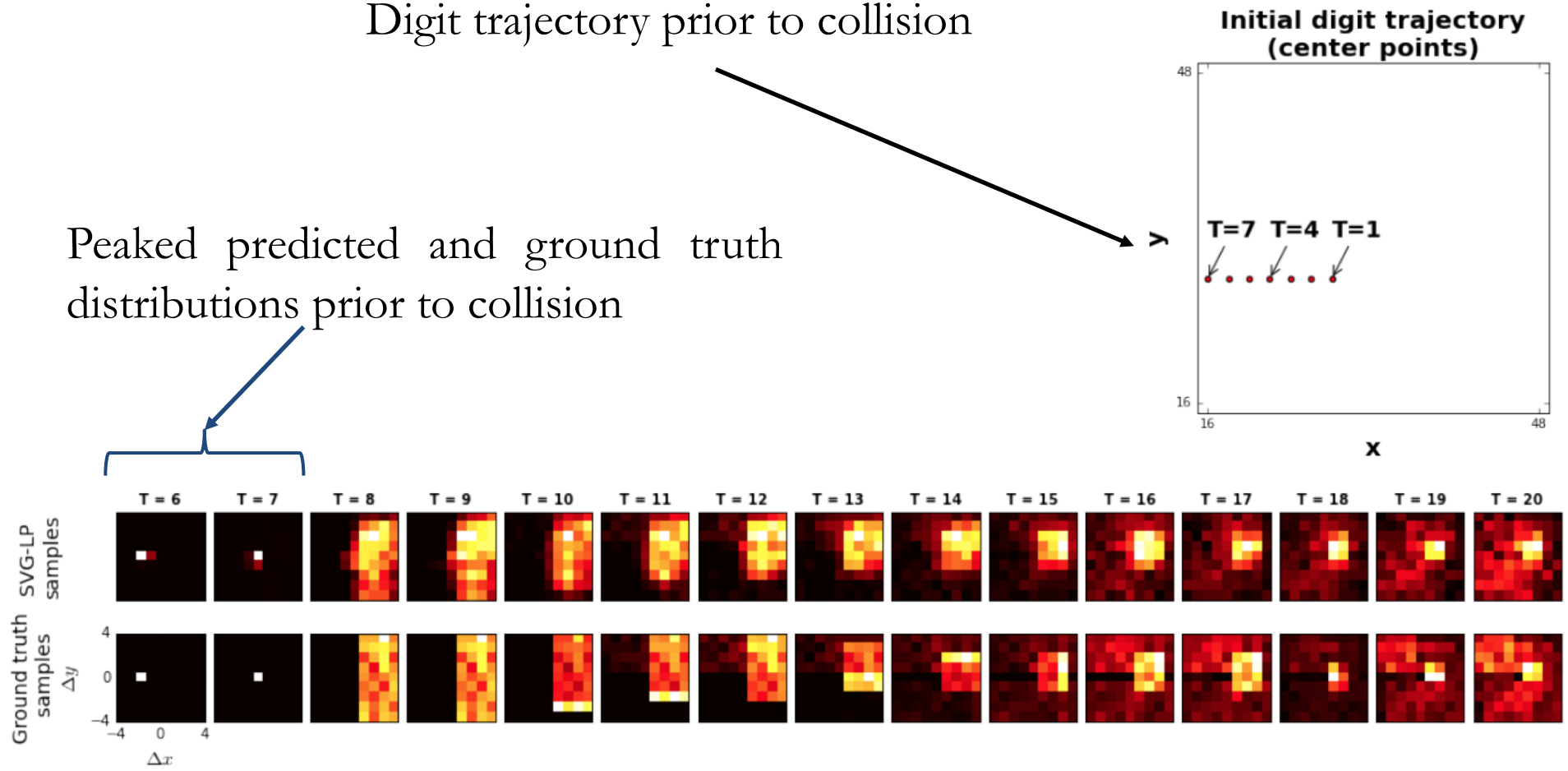
**Red dashed lines:**
digit 2 collides with wall




[Denton et al. ICML 2018]

Digit trajectory prior to collision

Peaked predicted and ground truth distributions prior to collision




[Denton et al. ICML 2018]

# BAIR robot push dataset (Ebert et al., 2017)

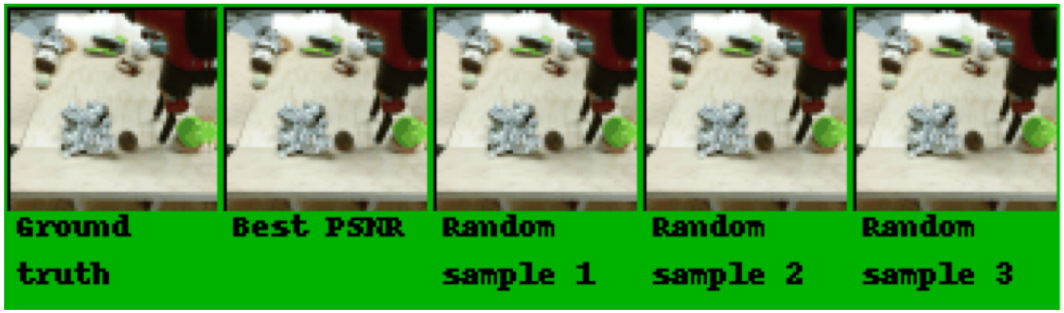
- Sawyer robotic arm pushing a variety of objects around a table top
- 30 frames in sequence, 64x64 resolution
- Movements of the arm are highly stochastic

*[Ebert et al. Self-supervised visual planning with temporal skip connections. CoRL, 2017.]*

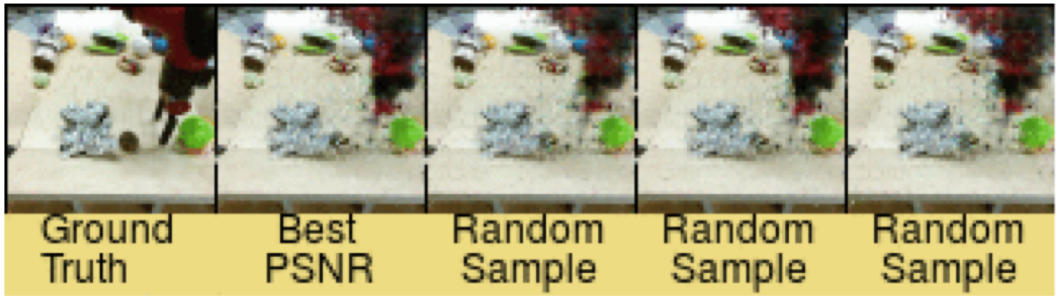
[Denton et al. ICML 2018]

# BAIR robot push dataset

**SVG-LP (ours)**




**Babaeizadeh et al. (2018)**

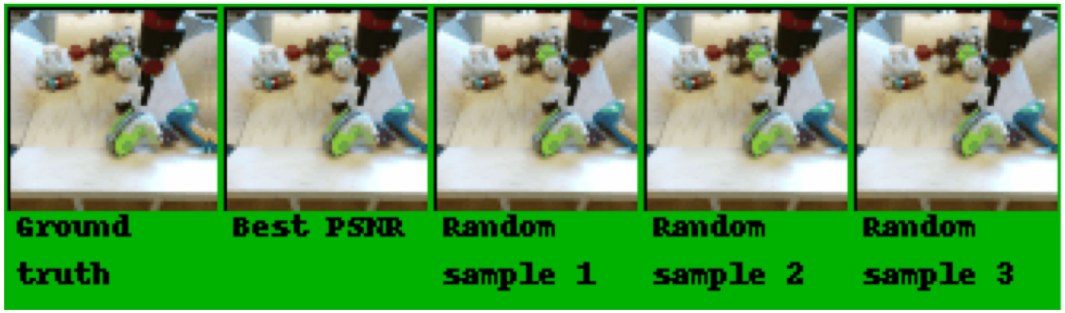



[*Babaeizadeh et al. Stochastic variational video prediction. ICLR, 2018.*]

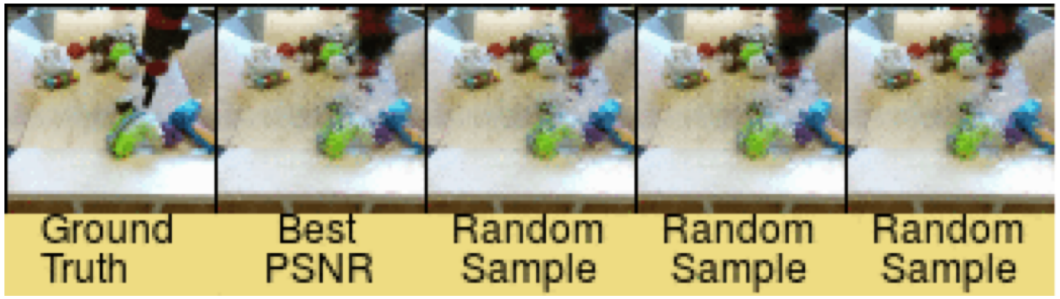
[Denton et al. ICML 2018]

# BAIR robot push dataset

**SVG-LP (ours)**




**Babaeizadeh et al. (2018)**

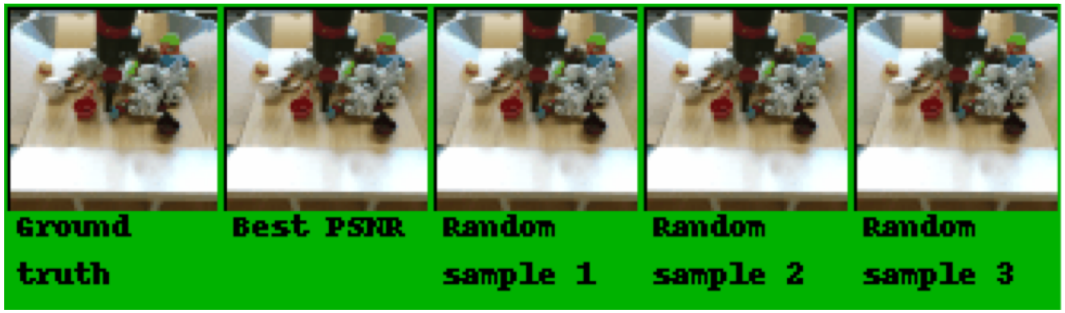



[*Babaeizadeh et al. Stochastic variational video prediction. ICLR, 2018.*]

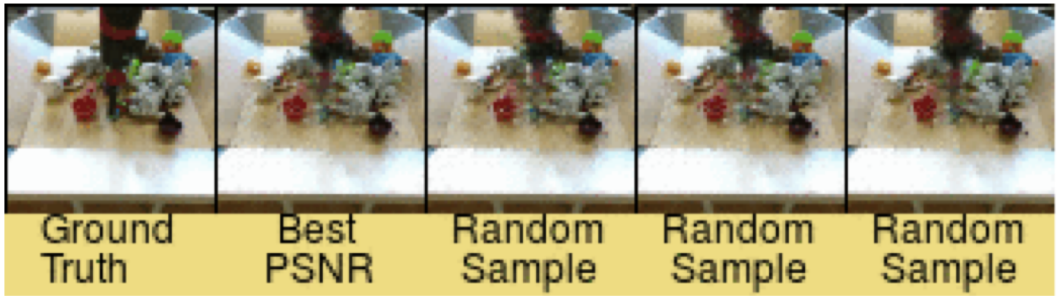
[Denton et al. ICML 2018]

# BAIR robot push dataset

**SVG-LP (ours)**




**Babaeizadeh et al. (2018)**




[*Babaeizadeh et al. Stochastic variational video prediction. ICLR, 2018.*]

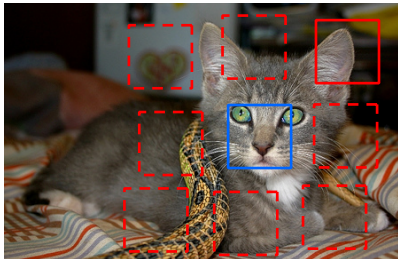
[Denton et al. ICML 2018]

# 2. Self-supervised Learning

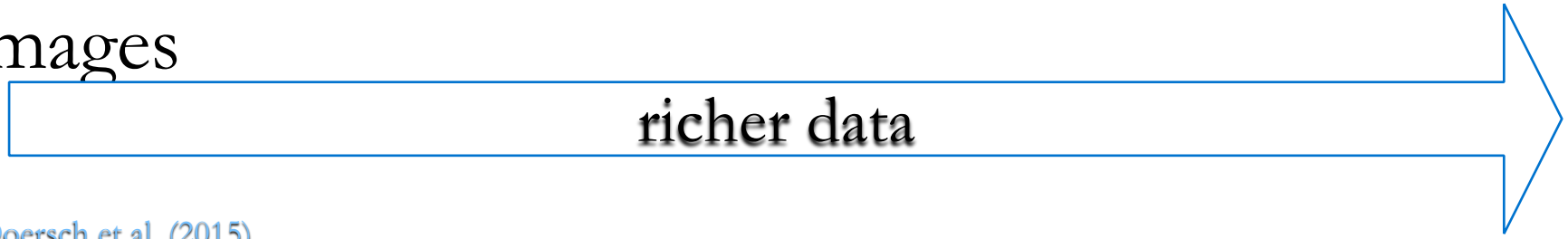
# Unsupervised learning as a pre-training step

- Target task is high-level understanding of signal
  - E.g. Classification, detection
- Unsupervised learning to pre-train models
  - Then fine-tune with labels on target task
- Some success in NLP
  - Word2vec for word embeddings
  - Language modeling for machine translation
- No equivalent success in computer vision or other domains
- But there are a lot of attempts!

# Range of self-supervised systems

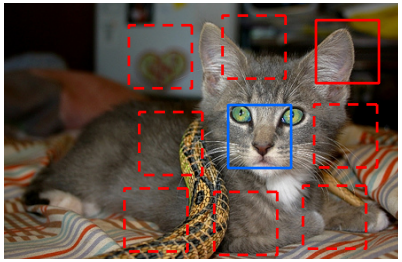


images




- Doersch et al. (2015)
- Zhang et al. (2016)
- Zhang et al. (2017)
- Noroozi et al. (2016)
- Pathak et al. (2016)

# Range of self-supervised systems



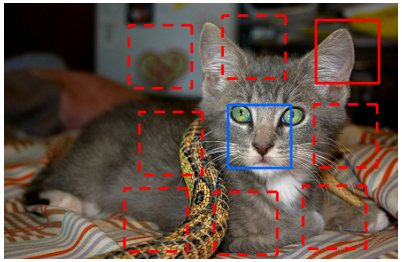
images videos

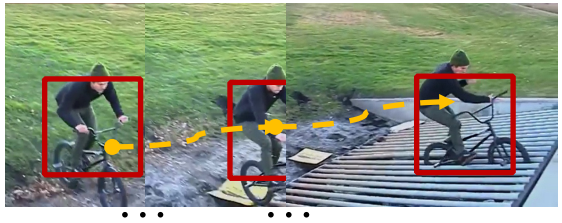
richer data

- Doersch et al. (2015)
- Zhang et al. (2016)
- Zhang et al. (2017)
- Noroozi et al. (2016)
- Pathak et al. (2016)

- Wang et al. (2015)
- Misra et al. (2016)
- Pathak et al. (2017)

# Range of self-supervised systems





images videos sound & depth

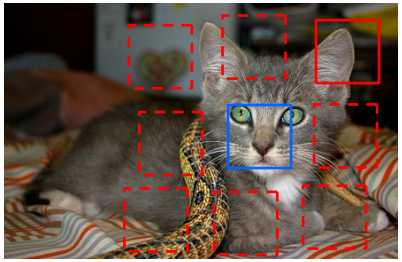
**richer data**

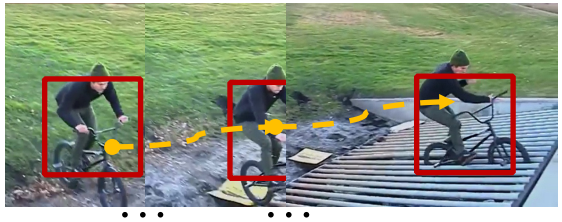
- Doersch et al. (2015)
- Zhang et al. (2016)
- Zhang et al. (2017)
- Noroozi et al. (2016)
- Pathak et al. (2016)

- Wang et al. (2015)
- Misra et al. (2016)
- Pathak et al. (2017)

- Owens et al. (2016)
- Zhang et al. (2017)
- Bansal et al. (2016)

# Range of self-supervised systems





images videos sound & depth actions

**richer data**

- Doersch et al. (2015)
- Zhang et al. (2016)
- Zhang et al. (2017)
- Norouzi et al. (2016)
- Pathak et al. (2016)

- Wang et al. (2015)
- Misra et al. (2016)
- Pathak et al. (2017)

- Owens et al. (2016)
- Zhang et al. (2017)
- Bansal et al. (2016)

- Agarwal et al. (2015)
- Jayaraman et al. (2015)
- Pinto et al. (2016)
- Agarwal et al. (2016)
- Pinto et al. (2017)
- Pinto et al. (2016)

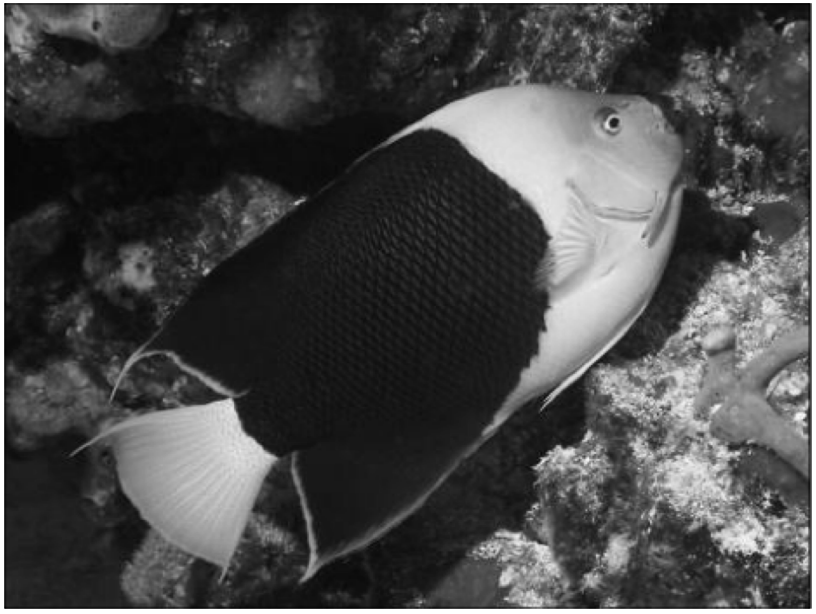
# Image colorization

Colorful Image Colorization

Richard Zhang, Phillip Isola, Alexei (Alyosha) Efros

slides from Zhang

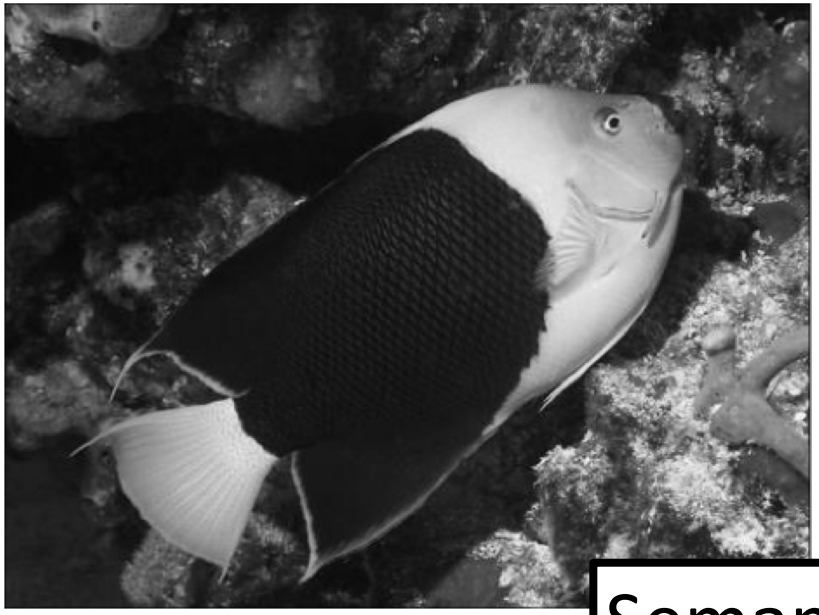
**http://richzhang.github.io/colorization/**



$\mathcal{F}$

Grayscale image: *L* channel $\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

Color information: *ab* cha $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$



$\mathcal{F}$

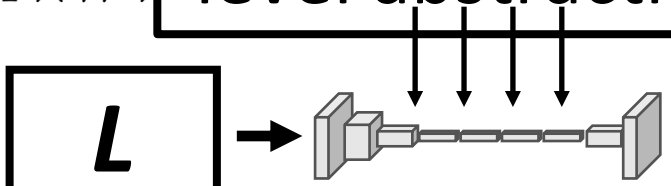
Grayscale image

$\mathbf{X} \in \mathbb{R}^{H \times W \times$

Semantics? Higher-level abstraction?

oncatenate (L,ab)

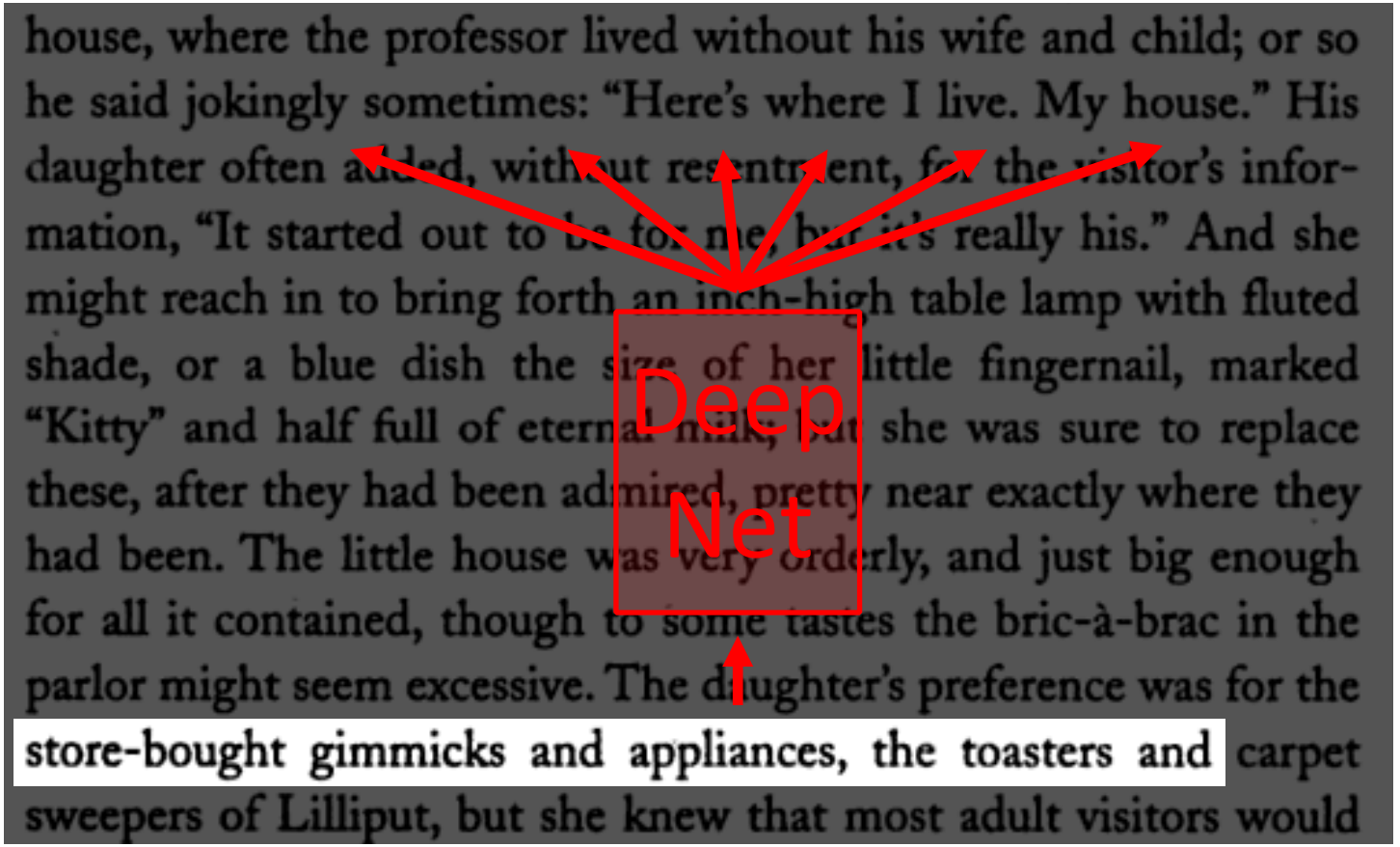
$(\mathbf{X}, \hat{\mathbf{Y}})$




"Free" supervisory signal

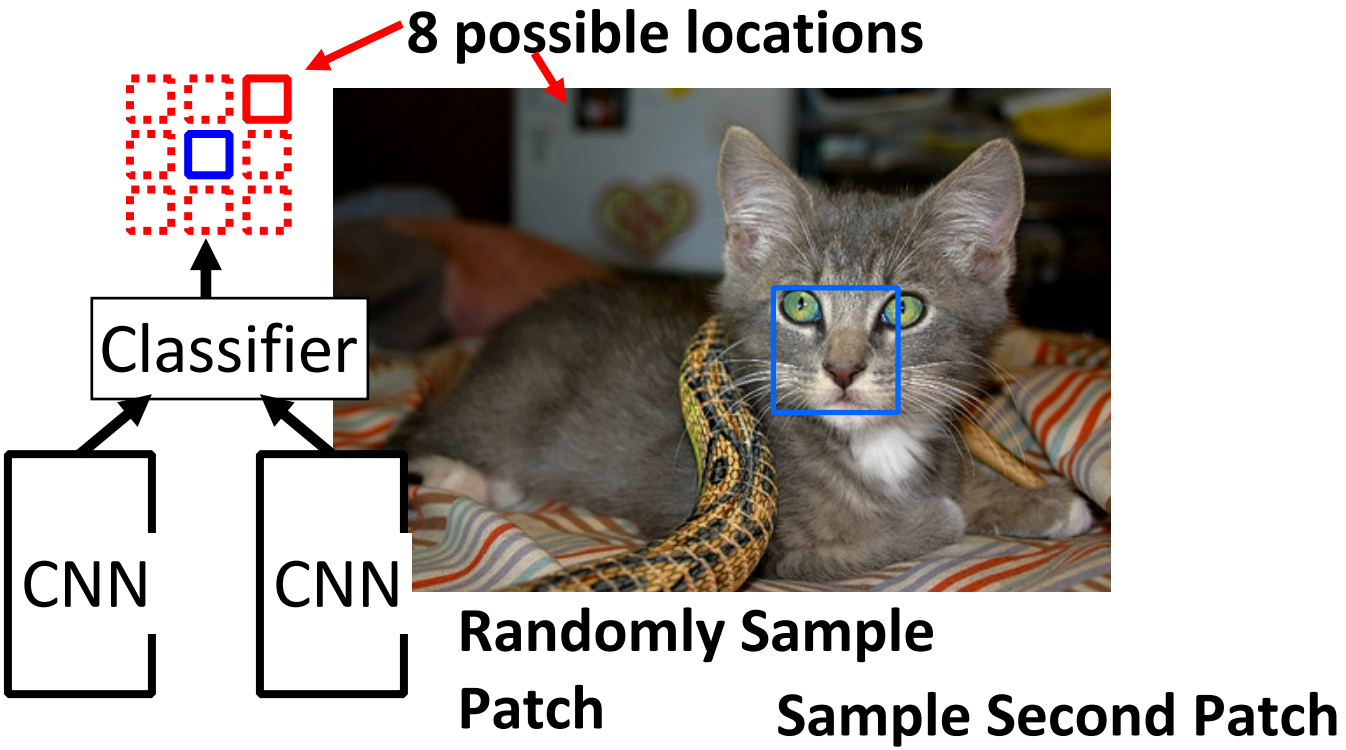
# Context as supervision

Collobert & Weston 2008; Mikolov et al. 2013



Unsupervised Visual Representation Learning by Context Prediction
[Doersch et al. ICCV 2015]

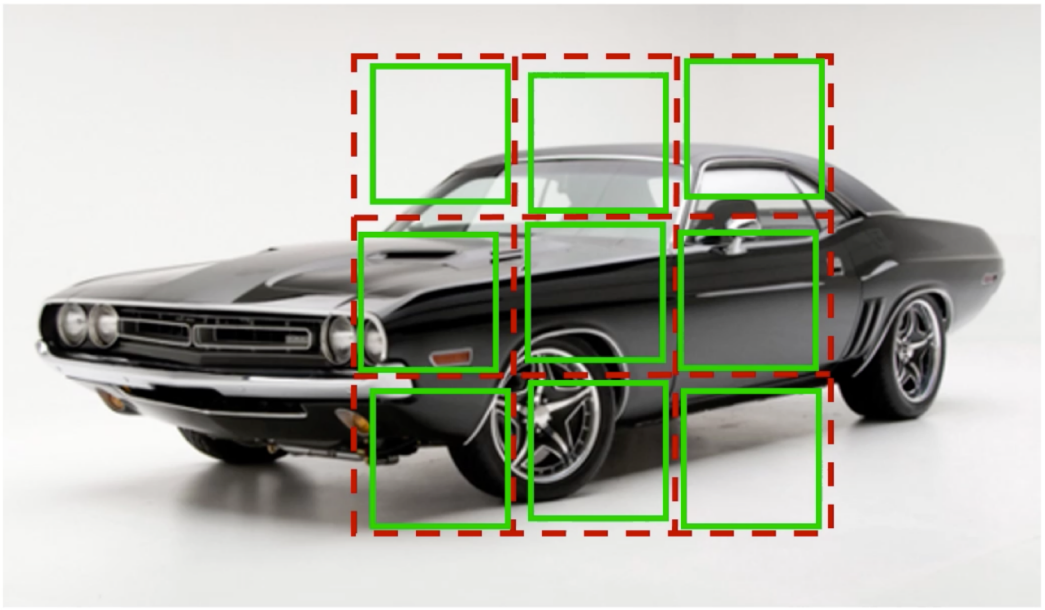
# Relative Position Task



# Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles, Noorozi et al. (2016)

What do we learn when we solve a Jigsaw puzzle?

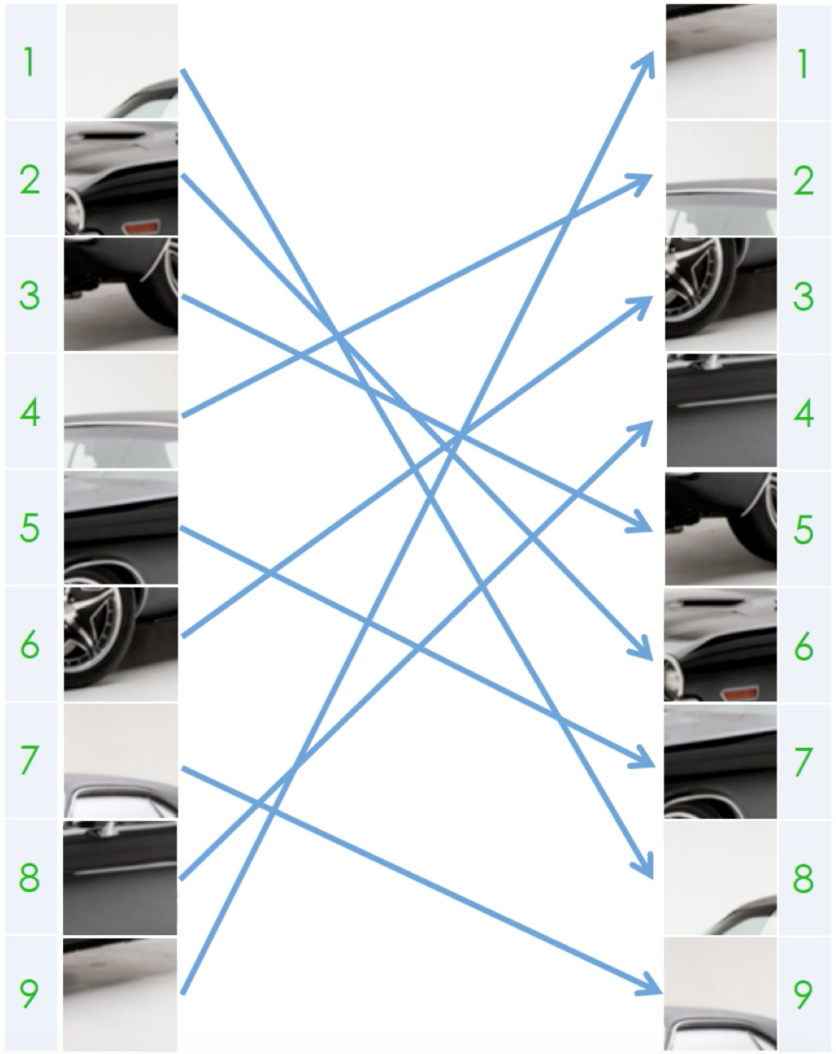
[Noorozi et al. (2016)]



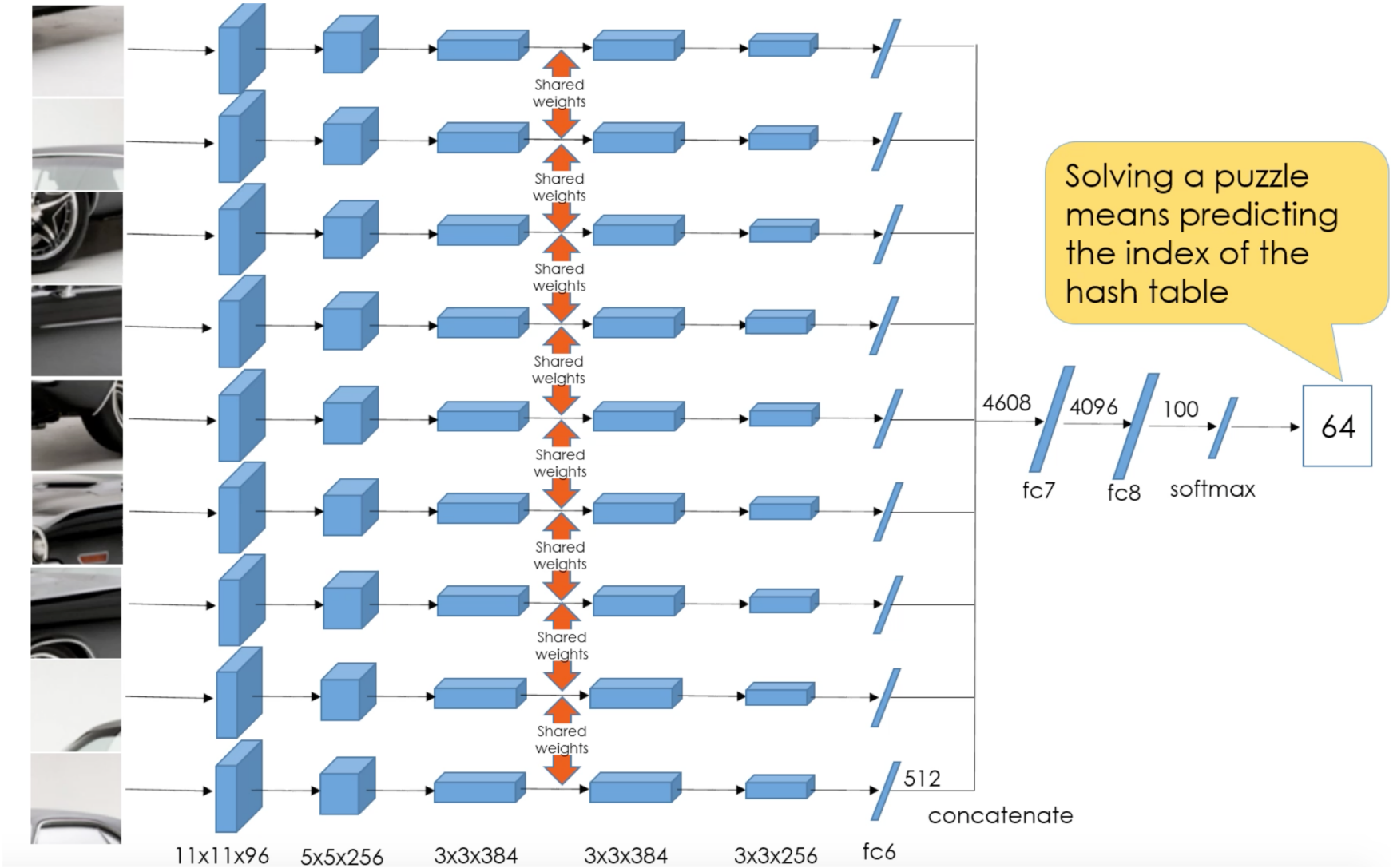
Hash Set

| index | table |
|---|---|
| | |
| 64 | 9,4,6,8,3,2,5,1,7 |
| | |
| | |

Reorder patches according to the selected hash table




[Noorozi et al. (2016)]



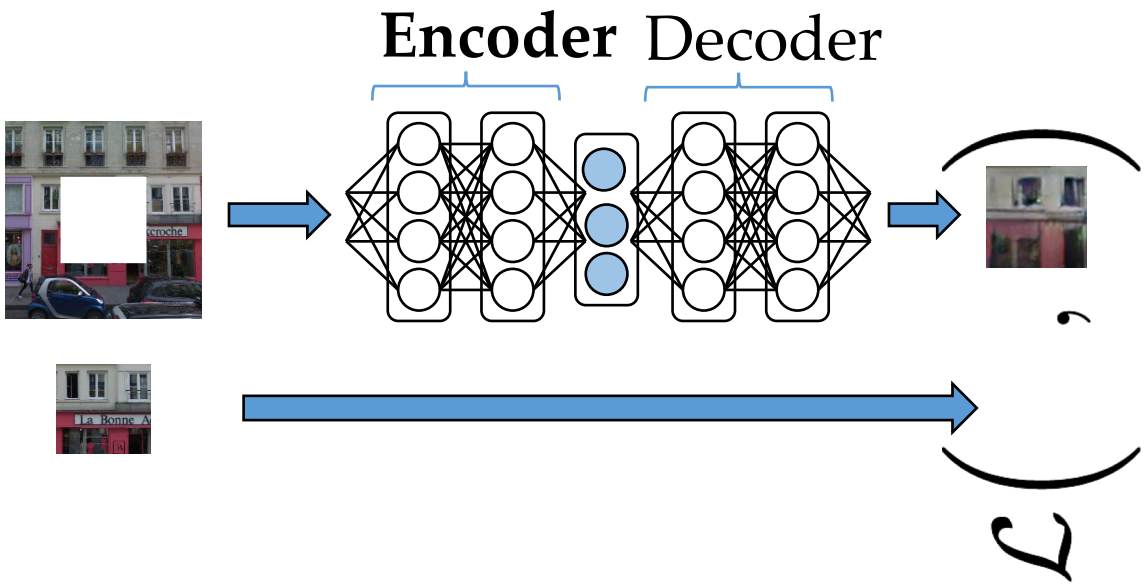

[Noorozi et al. (2016)]

# Feature Learning by Inpainting

[Context Encoders: Feature Learning by Inpainting, Pathak et al. (2016)]

**[Pathak et al. (2016)]**

# Context Encoders




- Encoder can be substituted with any network architecture like AlexNet etc.
- Decoder is a set of UpConv/deconv/frac-strided-conv layers

**[Pathak et al. (2016)]**

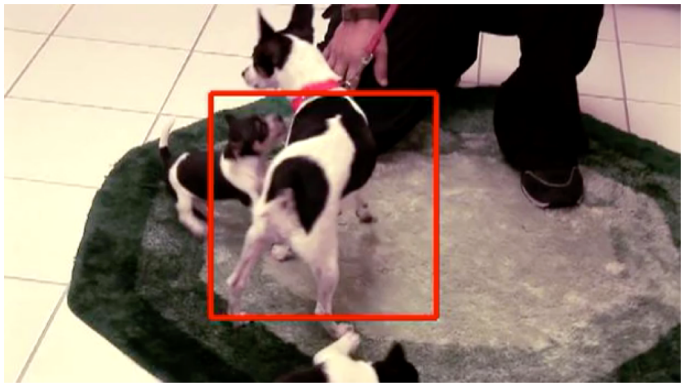
# Combined L2 + GAN loss

Input Image L2 Loss Adversarial Loss Joint Loss

[Pathak et al. (2016)]

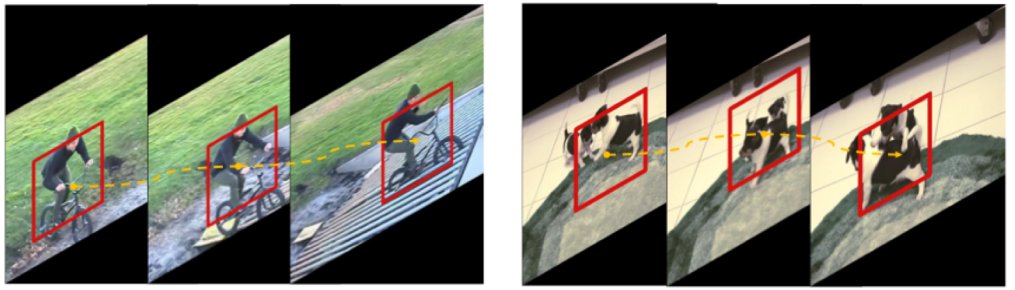
# Unsupervised Learning of Visual Representations using Videos, Wang & Gupta 2015

Idea: Object Tracking in Videos

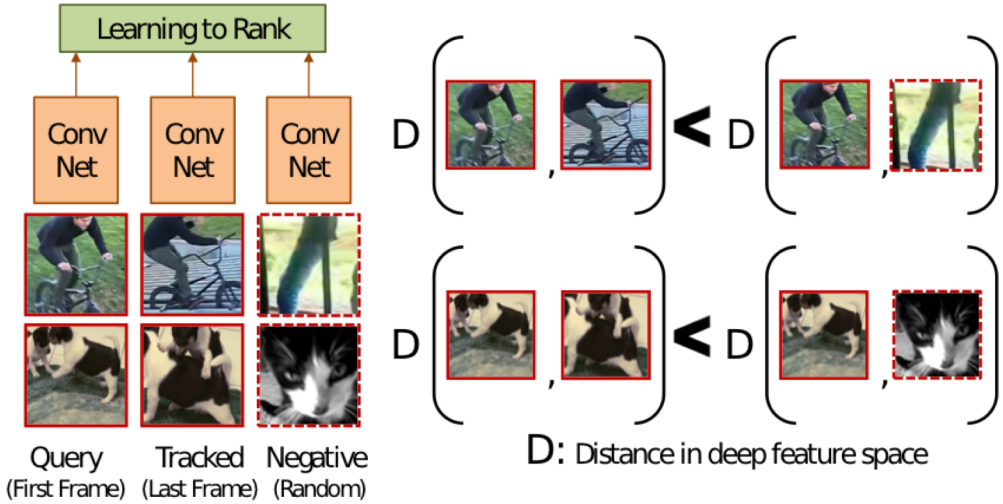


[Wang & Gupta 2015]

# Approach



(a) Unsupervised Tracking in Videos




(b) Siamese-triplet Network

(c) Ranking Objective

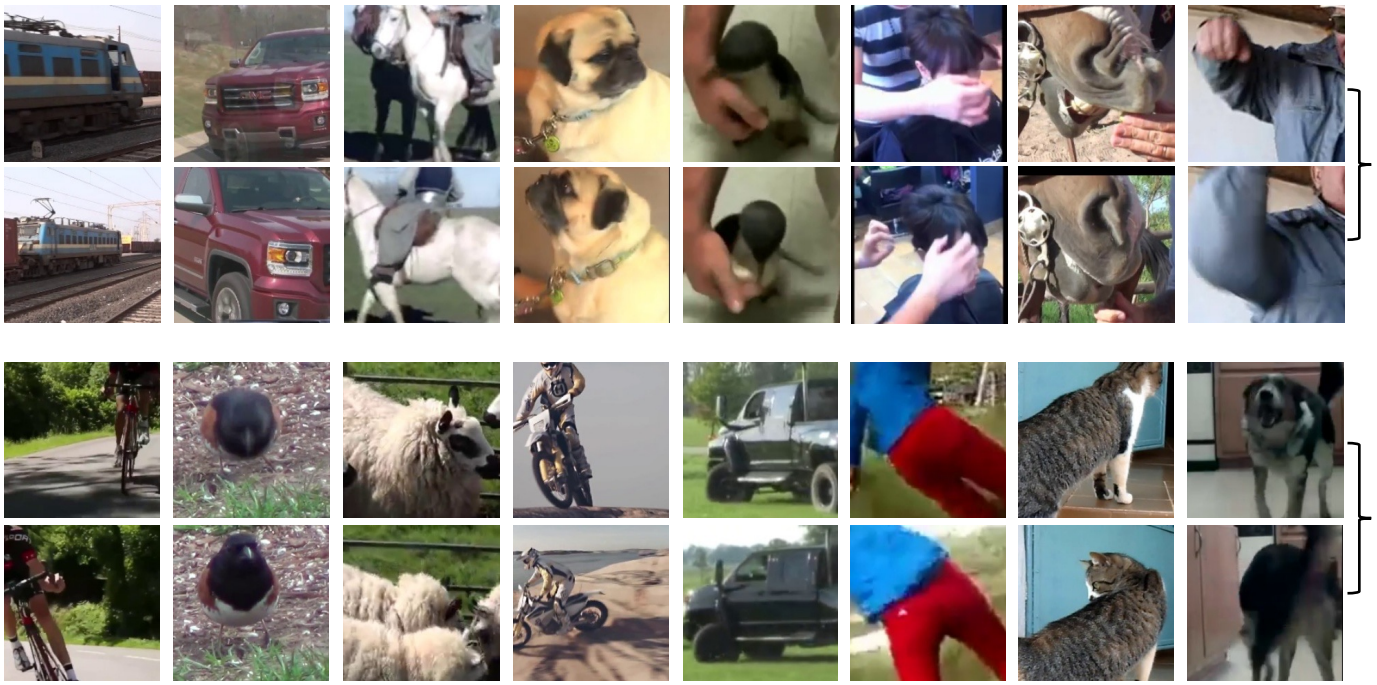
- Use object tracking in videos
- Classify if patches belong to the same track or not

[Wang & Gupta 2015]

# Patch Mining In Videos

- Track 8M patches in 100K videos from YouTube.

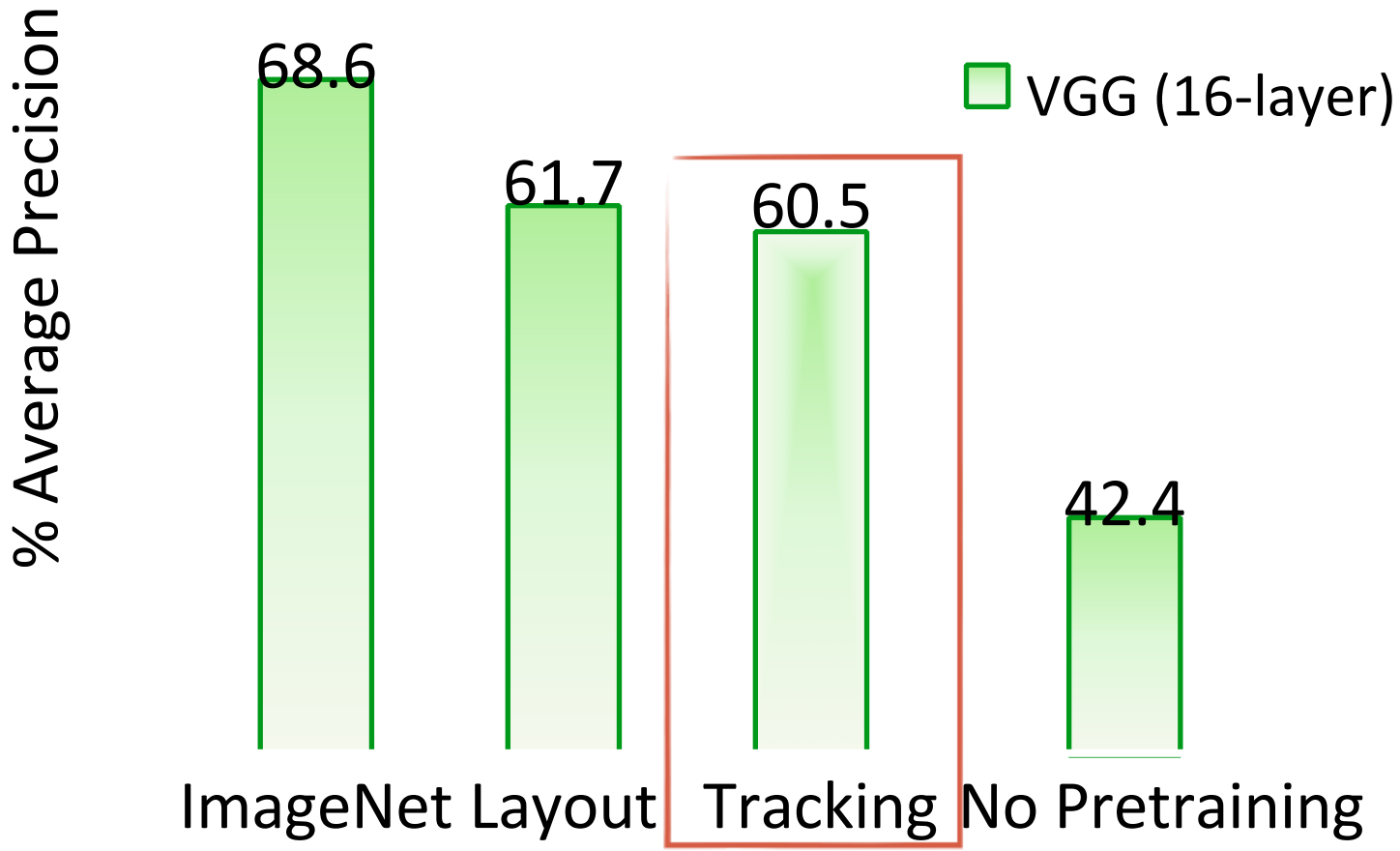
- Use off-the-shelf tracking algorithms with no learning.




[Wang & Gupta 2015]

# VOC 2007 Detection Performance

(pretraining for R-CNN)




[Wang & Gupta 2015]

# Leveraging Temporal Video Structure

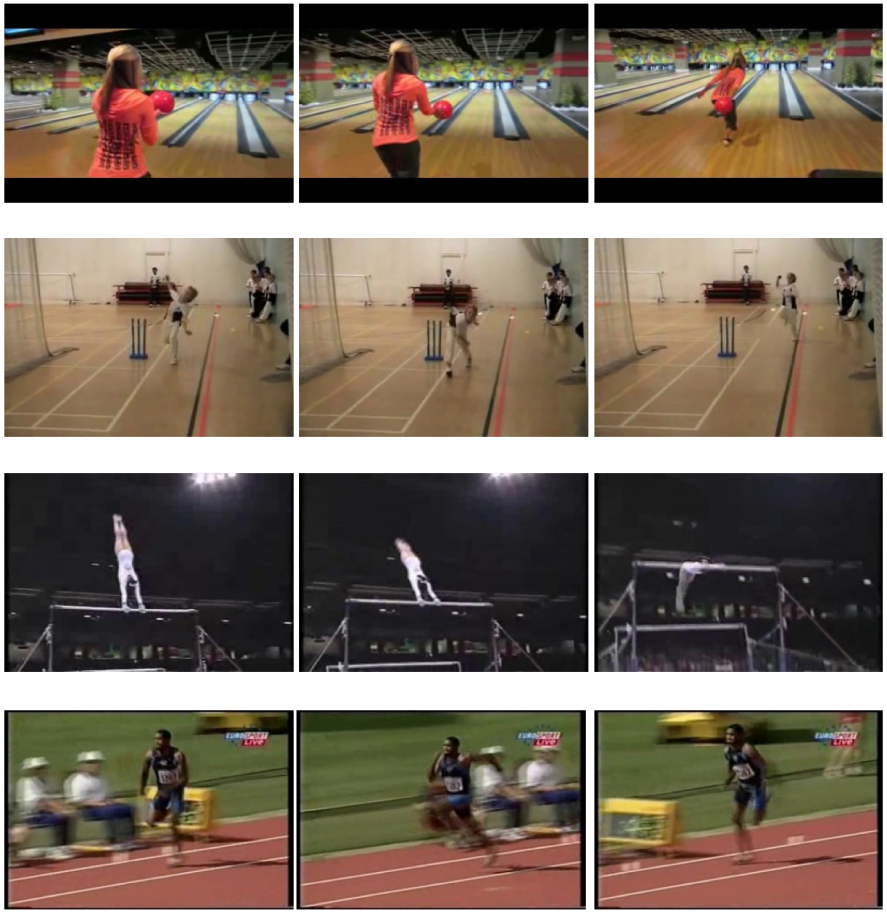
[Shuffle and learn: unsupervised learning using temporal order verification, Misra et al. ECCV 2016]

- Videos have temporal structure

- Can we use this to learn an image representation?

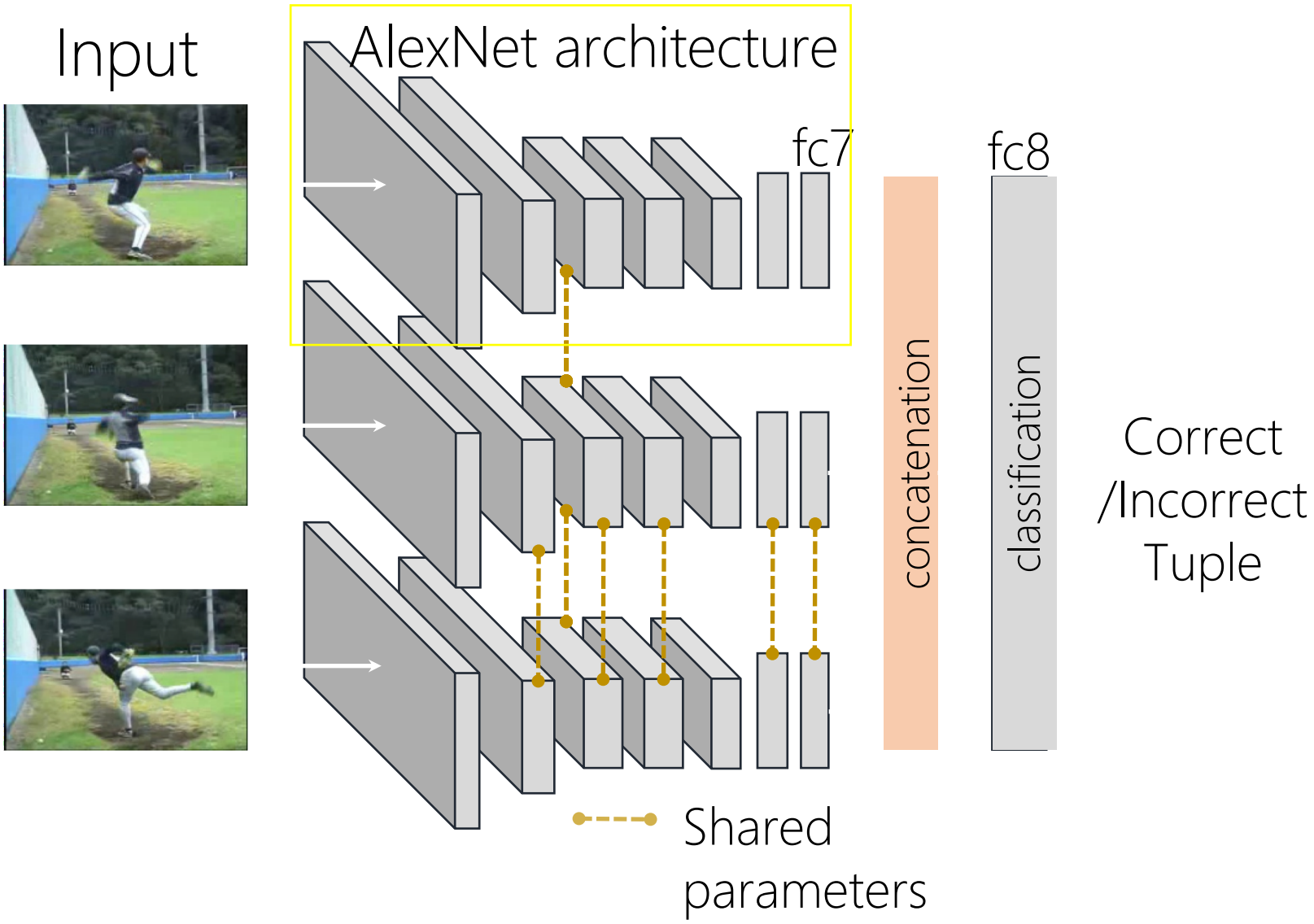



Positive



Negative

[Misra et al. ECCV 2016]




[Misra et al. ECCV 2016]

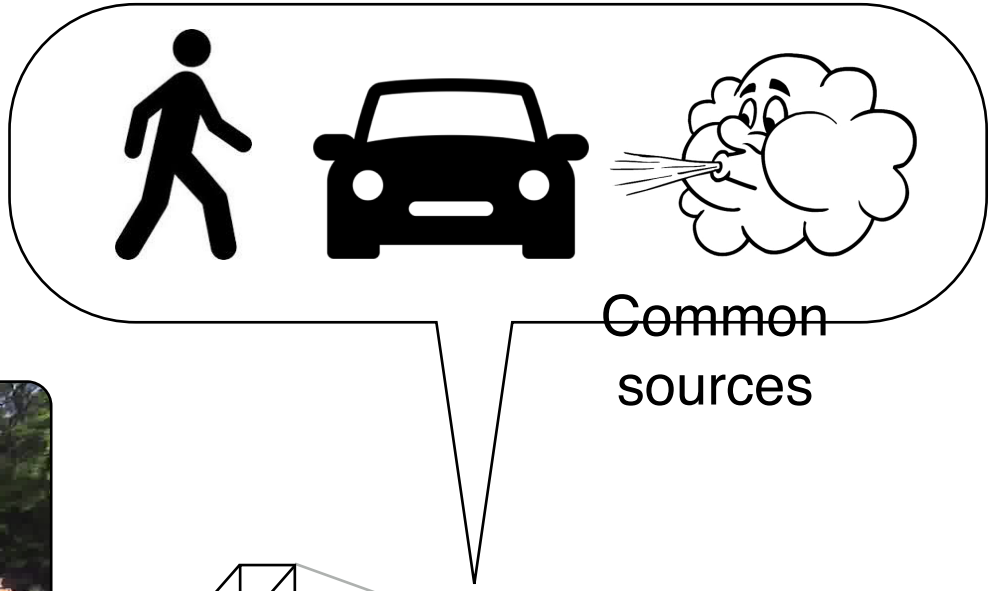
# Results: Finetune on Action Recognition

| Dataset | Initialization | Mean Classification Accuracy |
|---|---|---|
| UCF101 | Random | 38.6 |
| | Ours | 50.2 |
| | ImageNet pre-trained | **67.1** |
| HMDB51 | Random | 13.3 |
| | Ours | 18.1 |
| | UCF101 pre-trained | 15.2 |
| | ImageNet pre-trained | **28.5** |

[Misra et al. ECCV 2016]

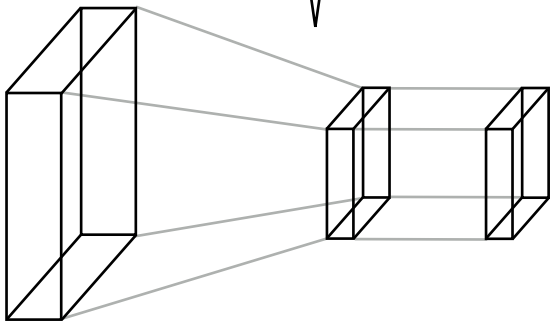
# Visual + Audio

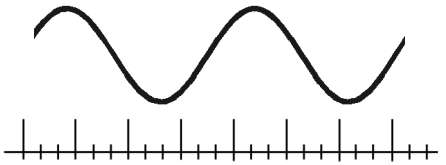
[Ambient Sound Provides Supervision for Visual Learning,
[Owens et al. (2016)]



Common sources

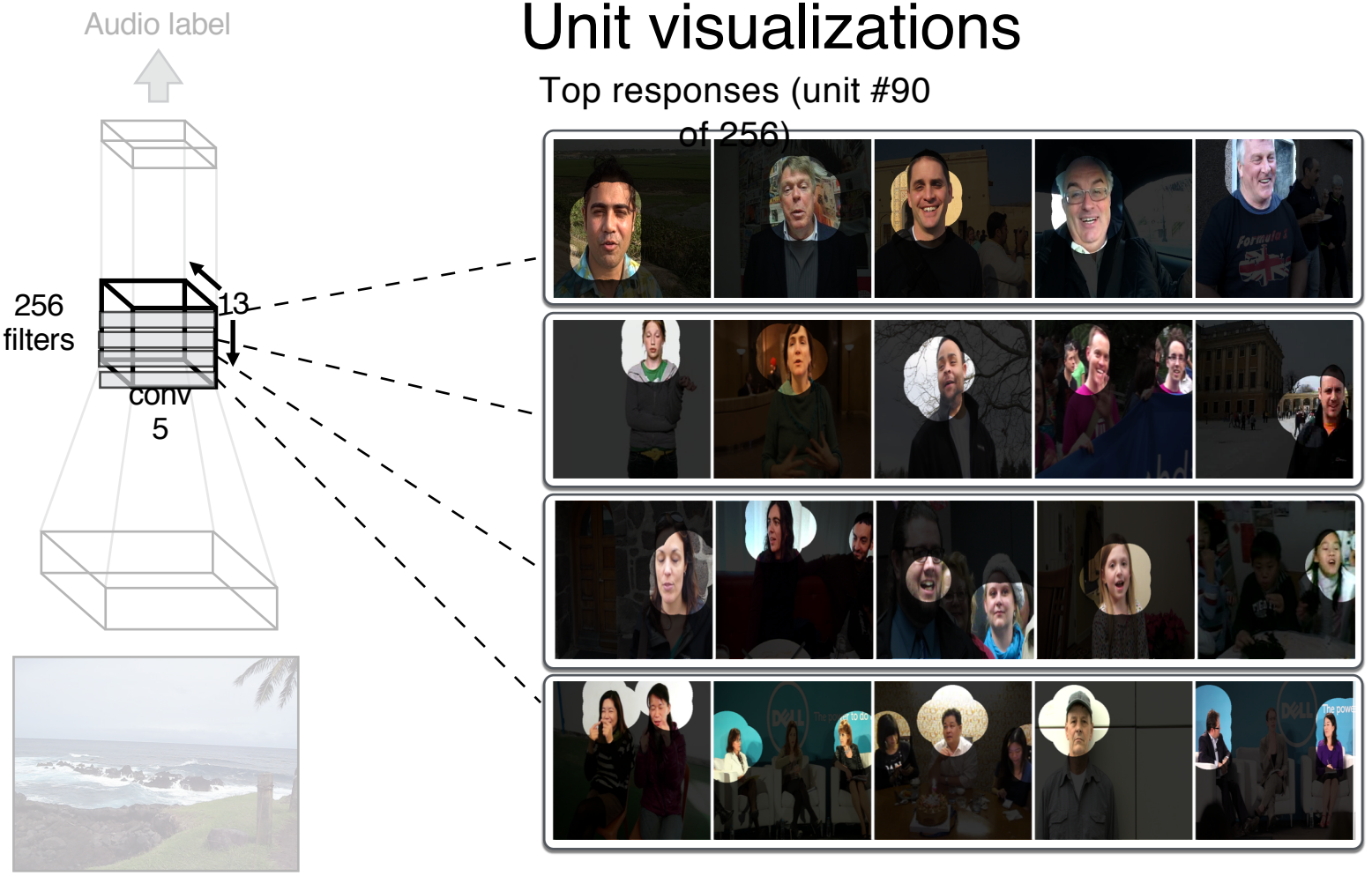
Image



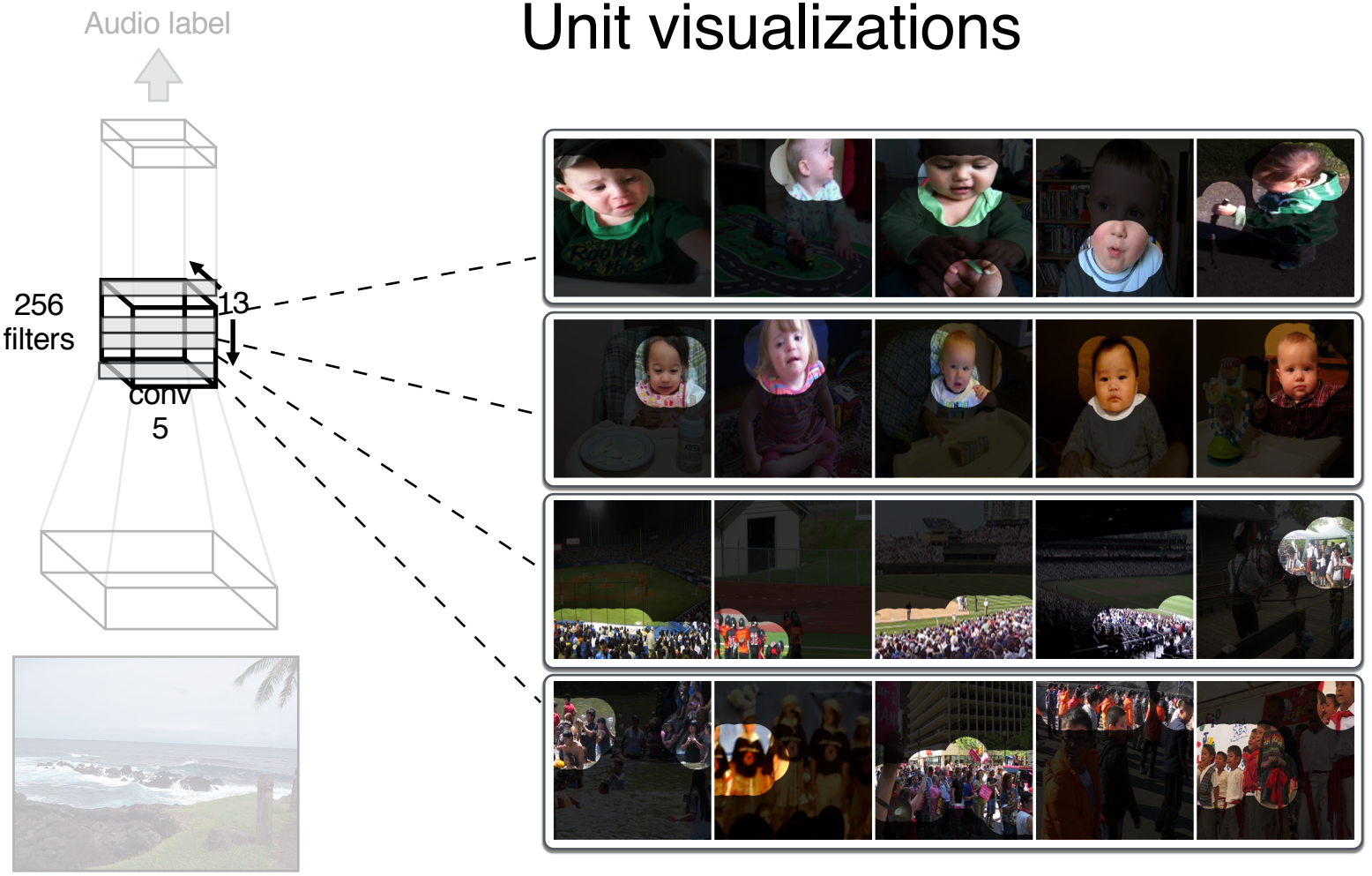


Audio

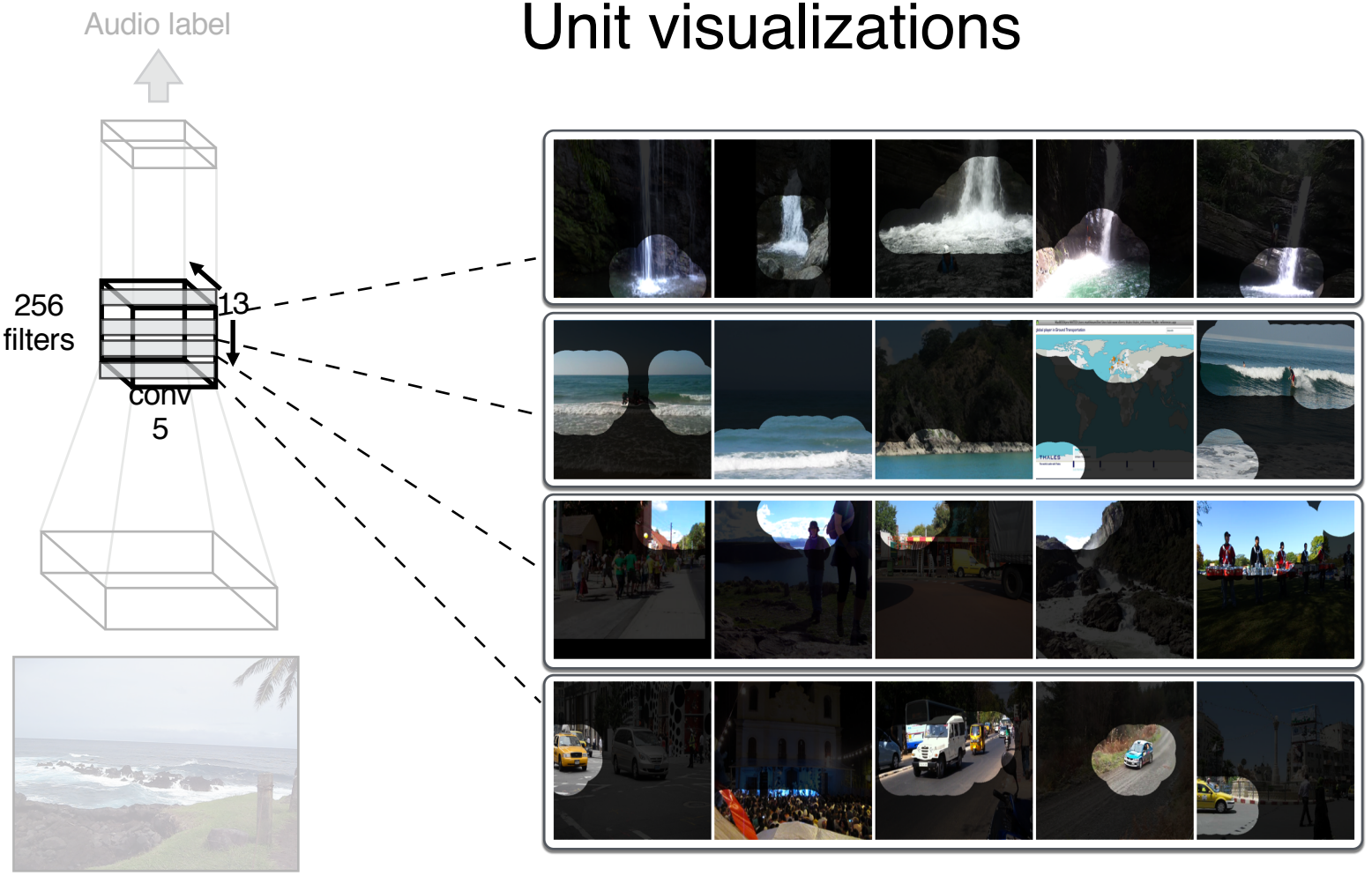
# Unit visualizations



# Unit visualizations



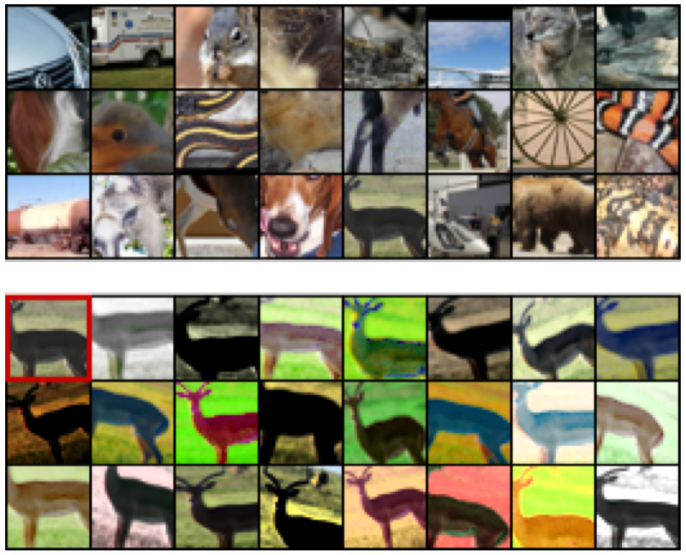
# Unit visualizations



# Main issue with all these methods

- All these models rely on expert knowledge
- Need to define y(x) for each new domain
- Not clear how to select a y(x) that is a good target to learn all-purpose features

# [Dosovitskiy et al. ICLR 2014]



- 1 class = single image + its transformations
- Learn to classify each "class"
- Domain knowledge about appropriate transformations
- does not scale

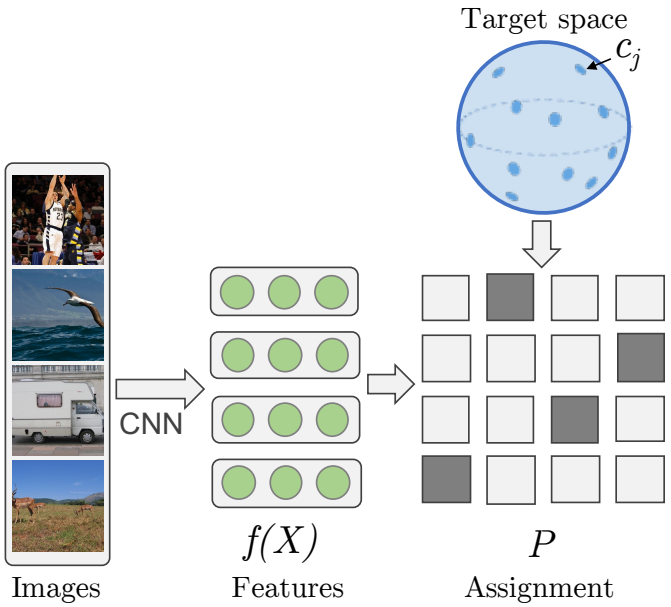
# Unsupervised Learning by Predicting Noise

**Piotr Bojanowski, Armand Joulin**

ICML 2017

# Unsupervised Learning by Predicting Noise

[Bojanowski & Joulin, ICML 2017]




- Inspired by Dosovitskiy et al.
- Learn mapping from images to a sphere
- Fix targets on sphere
- Simultaneously:
  - Learn the mapping
  - Optimize the assignment between images and targets

# Deep Discriminative Clustering

- We are given a set of n images

$$\{x_1, \ldots, x_n\}$$

- We want to learn a visual features f without using labels

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \min_{y_i} \ell(f_\theta(x_i), y_i)$$

$$\min_{\theta} \min_{Y} \frac{1}{2n} \|f_\theta(X) - Y\|_F^2$$

- We use the L2 loss

[Bojanowski & Joulin, ICML 2017]

# Label Collapse Problem

- Optimization over Y would lead to a collapse
- Repulsive costs are tricky to use
- Can impose constraints on Y but hard to optimize

[Bojanowski & Joulin, ICML 2017]

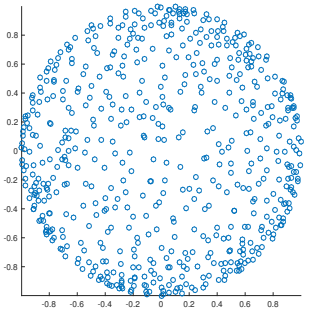
# Fixing the Target Representation

- Instead, we fix the target representation
- Allow a reassignment between targets and images

$$Y = PC \qquad \mathcal{P} = \{P \in \{0,1\}^{n \times k} \mid P\mathbf{1} = \mathbf{1}, P^\top \mathbf{1} = \mathbf{1}\}$$

- Targets C are **uniformly sampled on the sphere**

$$\min_\theta \min_{P \in \mathcal{P}} \frac{1}{2n} \|f_\theta(X) - PC\|_F^2$$



- Final objective function

[Bojanowski & Joulin, ICML 2017]

# Optimization

- We minimize our cost function in an on-line fashion
- We use the following algorithm:

**Require:** $T$ batches of images, $\lambda_0 > 0$
  **for** $t = \{1, \ldots, T\}$ **do**
    Obtain batch $b$ and representations $r$
    Compute $f_\theta(X_b)$
    Compute $P^*$ by minimizing w.r.t. $P$
    Compute $\nabla_\theta L(\theta)$ using $P^*$
    Update $\theta \leftarrow \theta - \lambda_t \nabla_\theta L(\theta)$
  **end for**

[Bojanowski & Joulin, ICML 2017]

# Optimizing the Permutation Matrix

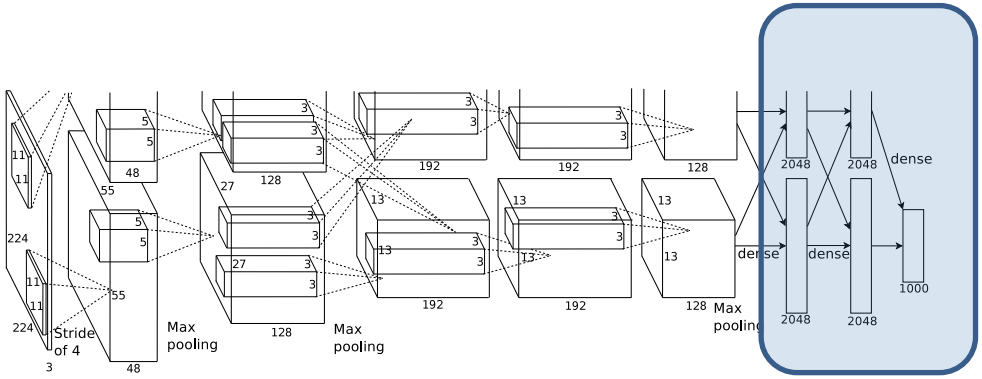
- At theta fixed, the permutation is obtained by solving

$$\max_{P \in \mathcal{P}} \operatorname{Tr}\left(P C f_\theta(X)^\top\right).$$

- Which is a linear program on the set of permutation matrices $\mathcal{O}(nb^2)$
- We can use the Hungarian algorithm

[Bojanowski & Joulin, ICML 2017]

# Experimental Setup




- AlexNet architecture
- Learn unsupervised features on ImageNet training set
- Retrain a classifier on top for a target transfer task, i.e. PASCAL VOC Classification / Detection

[Bojanowski & Joulin, ICML 2017]

# Baselines

- Self supervised models
  - Wang & Gupta – Temporal coherence in videos
  - Doersch et al. – Predict context patches
  - Zhang et al. – Predict color
  - Norouzi & Favaro – Solve jigsaw puzzles

- Unsupervised model
  - GAN
  - Auto-encoder
  - BI-GAN (Donahue et al.)

[Bojanowski & Joulin, ICML 2017]

# Pascal VOC - results

| | Classification | | Detection |
|---|---|---|---|
| Trained layers | fc6-8 | all | all |
| ImageNet labels | 78.9 | 79.9 | 56.8 |
| Agrawal et al. | 31.0 | 54.2 | 43.9 |
| Pathak et al. | 34.6 | 56.5 | 44.5 |
| Wang & Gupta | 55.6 | 63.1 | 47.4 |
| Doersch et al. | 55.1 | 65.3 | 51.1 |
| Zhang et al. | 61.5 | 65.6 | 46.9 |
| Autoencoder | 16.0 | 53.8 | 41.9 |
| GAN | 40.5 | 56.4 | - |
| BiGAN | 52.3 | 60.1 | 46.9 |
| NAT | 56.7 | 65.3 | 49.4 |

- compare favorably to SOTA
- Poor performance of AE / GAN

[Bojanowski & Joulin, ICML 2017]

# Nearest Neighbor Queries

[Bojanowski & Joulin, ICML 2017]

# Bojanowski & Joulin Summary

- Simple unsupervised approach
- No domain expert knowledge
- Scales to **very large** datasets
- Close to supervised pipeline
- SOTA performance (at the time) amongst unsupervised methods

# Deep Clustering for Unsupervised Learning of Visual Features

**Mathilde Caron, Piotr Bojanowski, Armand Joulin, Matthijs Douze**

ECCV 2018

# Motivation

- **Simpler** unsupervised approach?
- **Closer** to supervised pipeline?

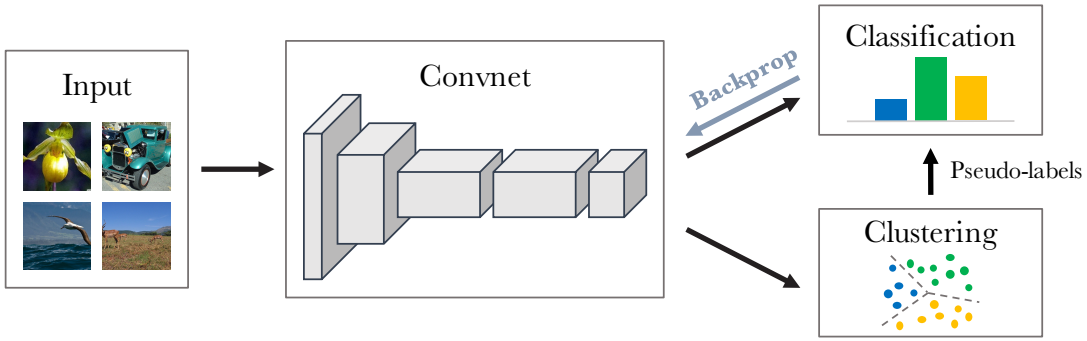
→clustering methods are simpler and closer

... but they suffer from the cluster collapsing problem

# Is the cluster collapsing a problem?

- Clustering almost never used with neural networks
- Optimal solution:
  - 1 cluster with every data point and trivial features
- This problem arises in discriminative clustering too
- Their solution: avoid non-empty clusters.
- if we enforce non-empty clusters, are other optimal trivial solutions?

# Simple model to test this idea




- Alternate between:
  - clustering features (k-means)
  - feature learning (classification)
- (Vaguely) conceptually similar to the E-M procedure

# How do we avoid the collapse of clusters?

- 3 simple tricks:
- **Reassign empty clusters** during k-means
- Using a **discriminative loss** during classification (logistic regression)
- **Class uniform sampling** during feature learning to avoid one cluster to dominate the others.

...and a lot of standard small “deep learning” tricks

# Alternate optimization

- Alternates between k-means:

$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^{N} \min_{y_n \in \{0,1\}^k} \|f_\theta(x_n) - C y_n\|_2^2 \quad \text{such that} \quad y_n^\top 1_k = 1$$

- And learning the parameters of the network:

$$\min_{\theta, W} \frac{1}{N} \sum_{n=1}^{N} \ell\left(g_W\left(f_\theta(x_n)\right), y_n\right)$$

# Transfer learning

| | Classification | | Detection | | Segmentation | |
|---|---|---|---|---|---|---|
| Method | FC6-8 | ALL | FC6-8 | ALL | FC6-8 | ALL |
| ImageNet labels | 78.9 | 79.9 | – | 56.8 | – | 48.0 |
| Random-rgb | 33.2 | 57.0 | 22.2 | 44.5 | 15.2 | 30.1 |
| Random-sobel | 29.0 | 61.9 | 18.9 | 47.9 | 13.0 | 32.0 |
| Pathak *et al.* [38] | 34.6 | 56.5 | – | 44.5 | – | 29.7 |
| Donahue *et al.* [20]* | 52.3 | 60.1 | – | 46.9 | – | 35.2 |
| Pathak *et al.* [27] | – | 61.0 | – | 52.2 | – | – |
| Owens *et al.* [44]* | 52.3 | 61.3 | – | – | – | – |
| Wang and Gupta [29]* | 55.6 | 63.1 | 32.8† | 47.2 | 26.0† | 35.4† |
| Doersch *et al.* [25]* | 55.1 | 65.3 | – | 51.1 | – | – |
| Bojanowski and Joulin [19]* | 56.7 | 65.3 | 33.7† | 49.4 | 26.7† | 37.1† |
| Zhang *et al.* [28]* | 61.5 | 65.9 | 43.4† | 46.9 | 35.8† | 35.6 |
| Zhang *et al.* [43]* | 63.0 | 67.1 | – | 46.7 | – | 36.0 |
| Noroozi and Favaro [26] | – | 67.6 | – | 53.2 | – | 37.6 |
| Noroozi *et al.* [45] | – | 67.7 | – | 51.4 | – | 36.6 |
| DeepCluster | **70.9** | **72.9** | **48.7** | **55.0** | **43.4** | **44.7** |

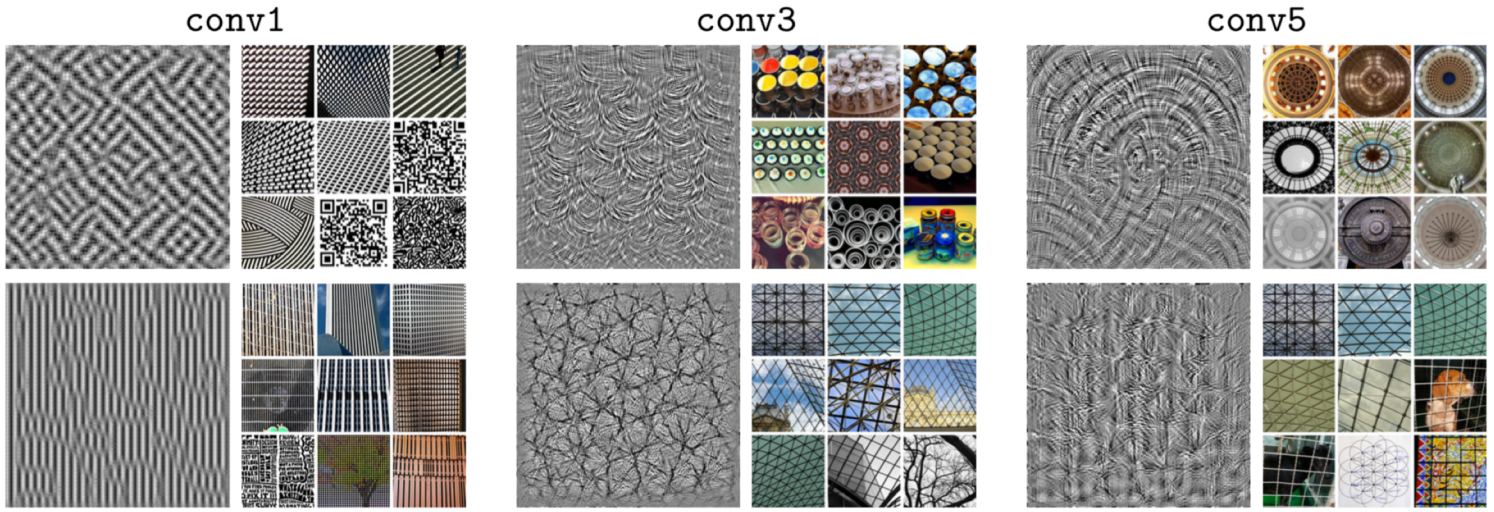
Transfer on Pascal VOC 2007 (higher the better)

# Transfer learning

| | | Classification | | Detection | | Segmentation | |
|---|---|---|---|---|---|---|---|
| Method | Training set | FC6-8 | ALL | FC6-8 | ALL | FC6-8 | ALL |
| Best competitor | ImageNet | 63.0 | 67.7 | 43.4† | 53.2 | 35.8† | 37.7 |
| DeepCluster | ImageNet | 70.9 | 72.9 | 48.7 | 55.0 | 43.4 | 44.7 |
| DeepCluster | YFCC100M | 67.3 | 69.3 | 45.6 | 53.0 | 39.2 | 42.2 |

Works on random internet images

# Reasonable features




Looking at the activation at different layers

# Limitation and future work

- Proof of concept that clustering works with deep learning
- Ad-hoc model with no theoretical guarantees
- Does not scale well:
  - converge in 12days on high-end GPU on Imagenet
  - Requires a k-means on the full dataset
- Future work:
  - More principled models
  - Looking at other “traditional” unsupervised learning approaches

# Summary

- Power of DL comes from ability to learn good representations
- Wide range of Unsupervised / Self-Supervised methods that devise "free" supervisory signals which can be used to learn representations via DL
- Unsolved problem:
  - Should be domain agnostic
  - Should be (nearly) as good as supervised methods