AutoEncoders

first meet with basic priciple

Weiwen Chen

2019.5.9


大哥抽中华

VAE

Variational AutoEncoder

Some background first: Autoencoders

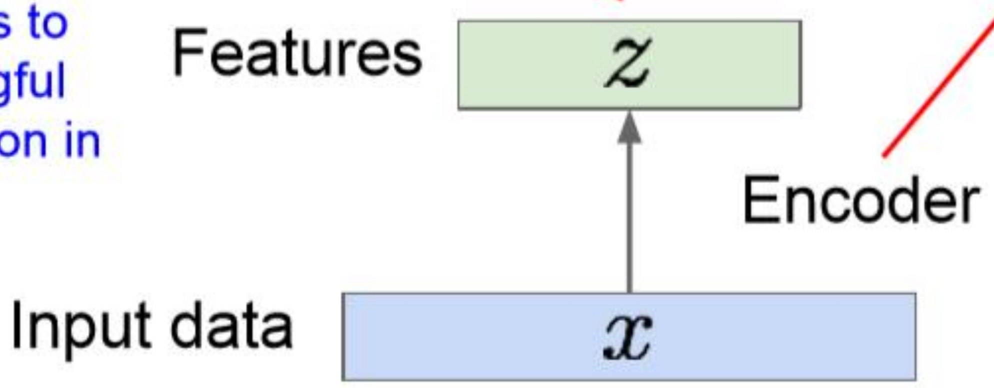
Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

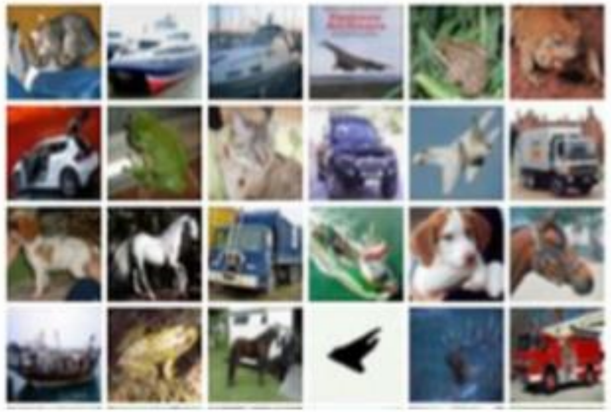
z usually smaller than **x**
(dimensionality reduction)

Q: Why dimensionality reduction?

A: Want features to capture meaningful factors of variation in data

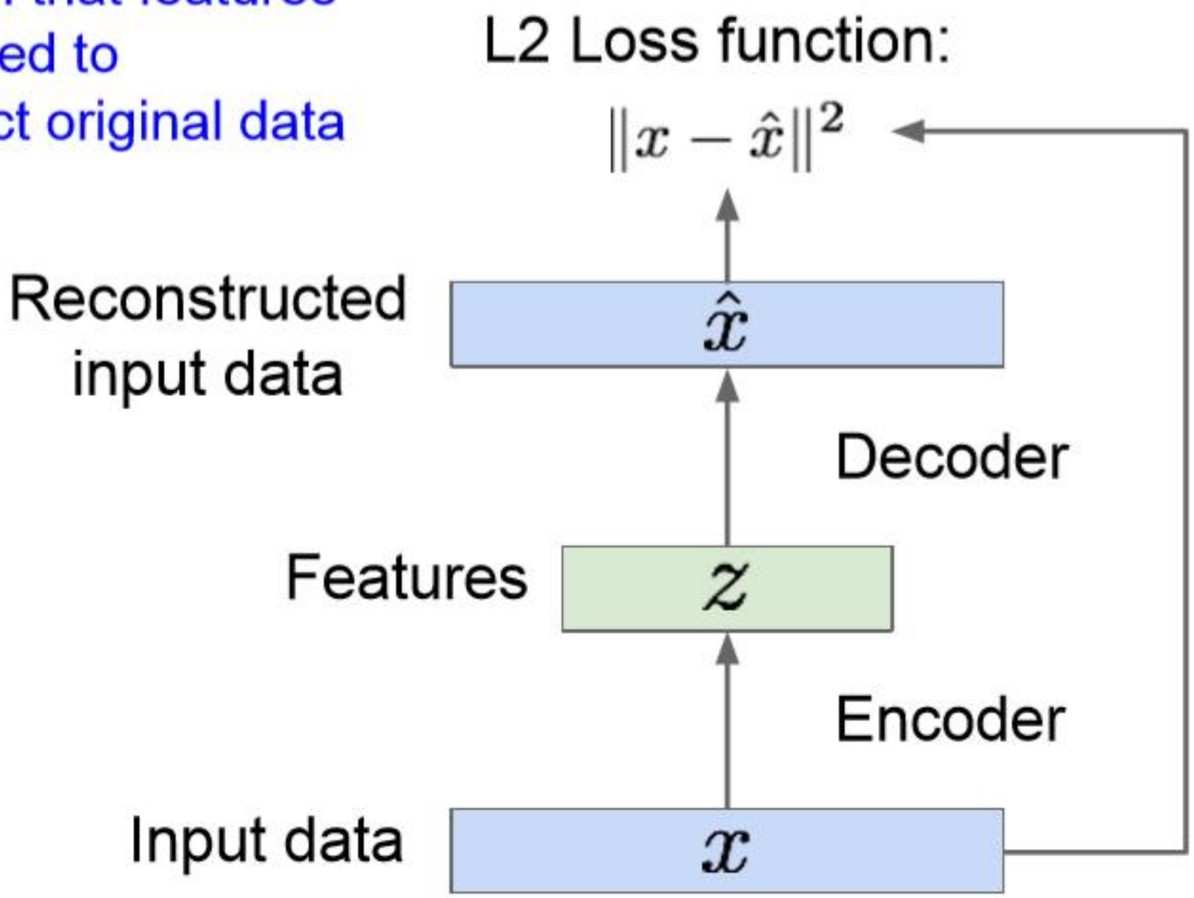
Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN

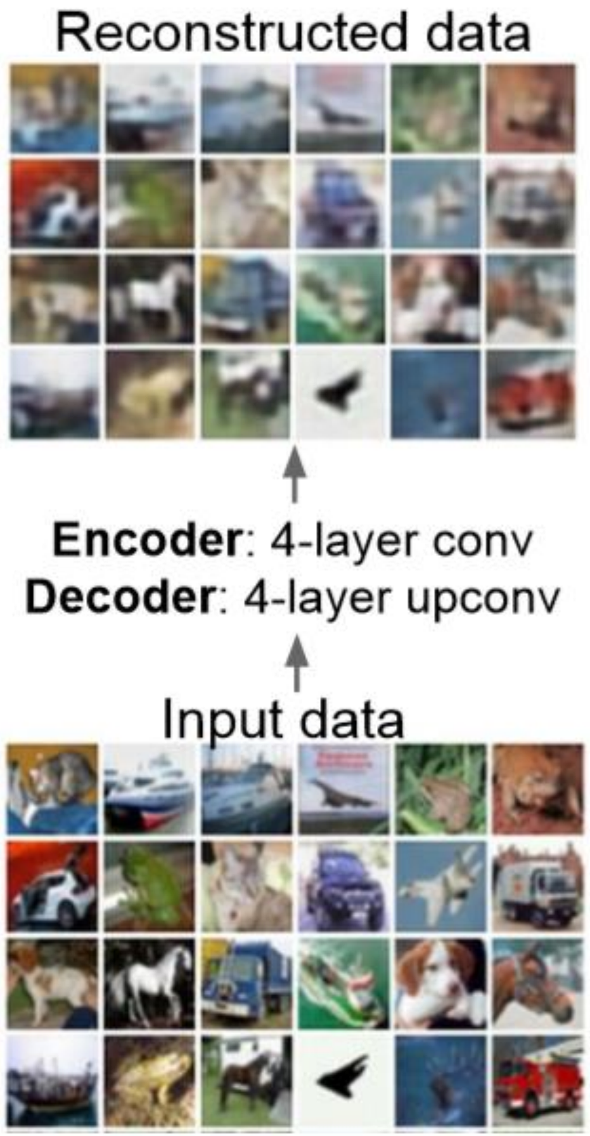




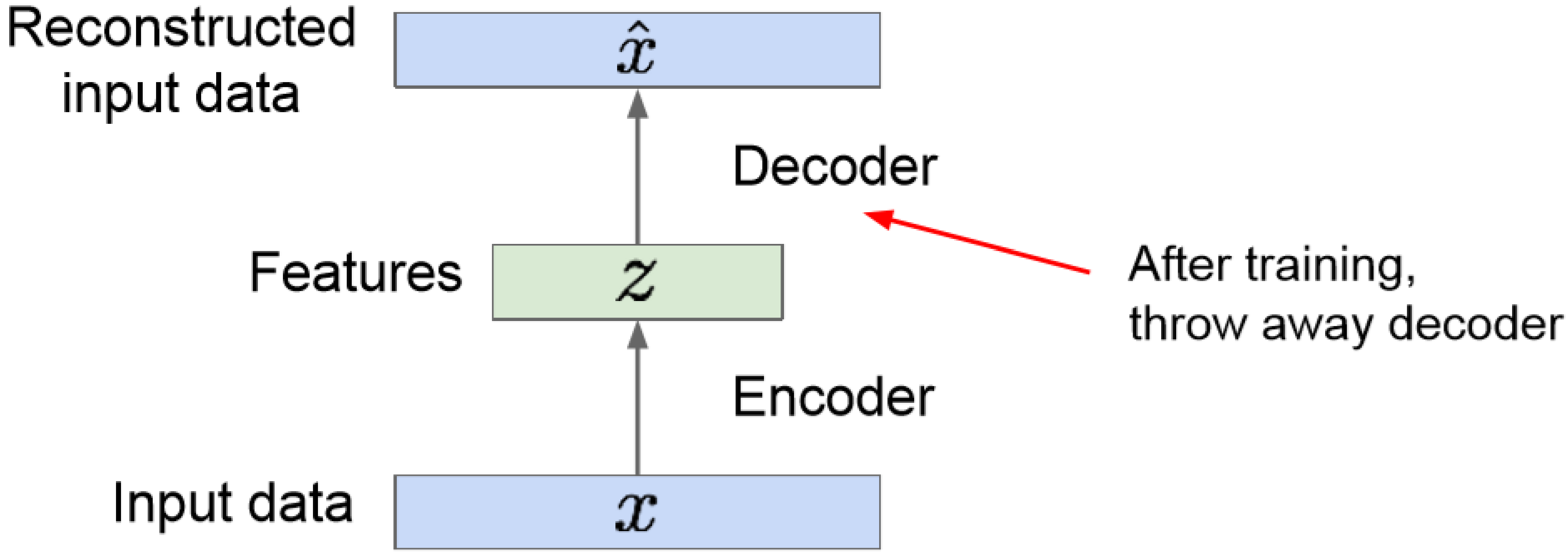
Some background first: Autoencoders

Train such that features can be used to reconstruct original data

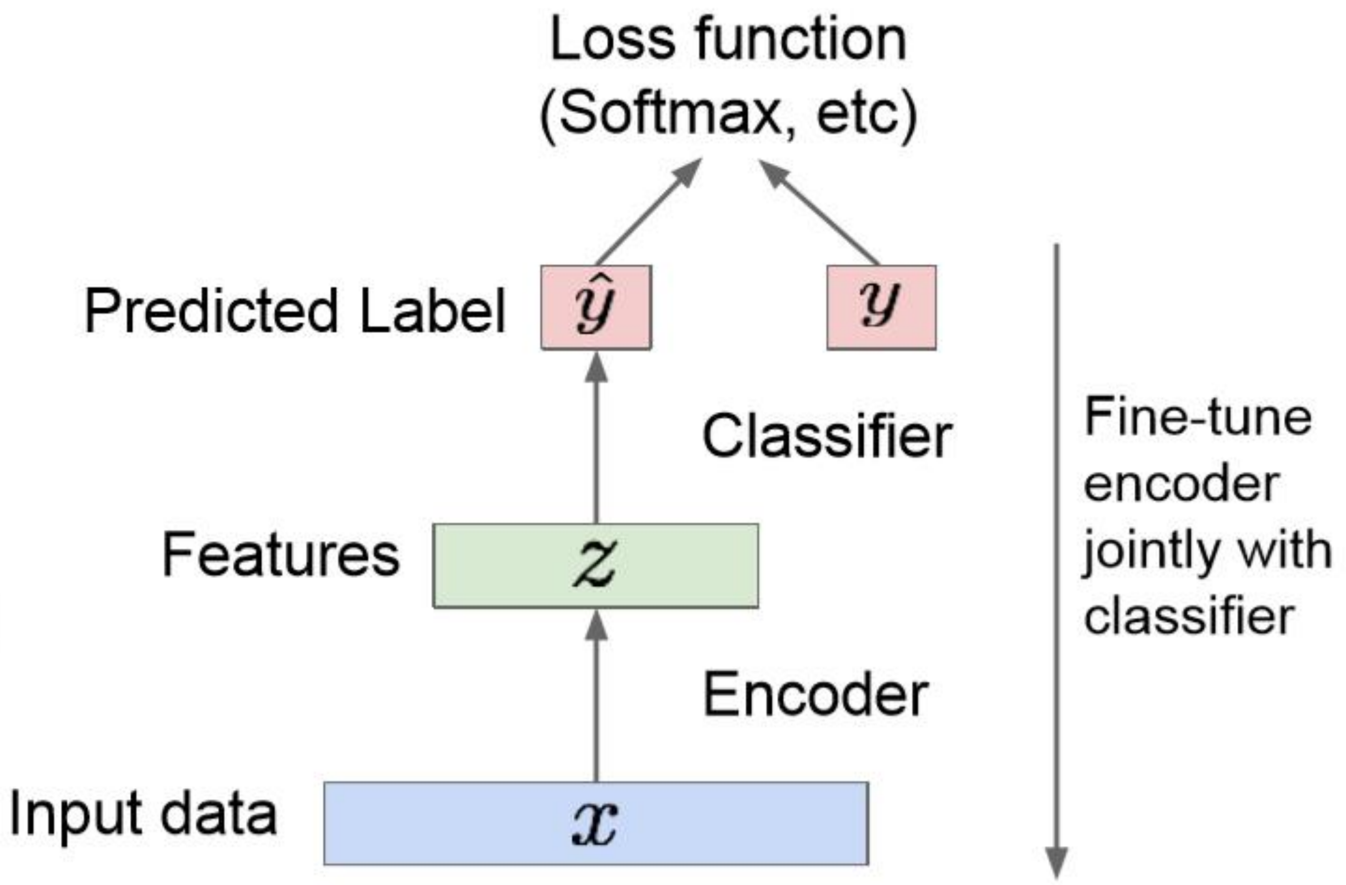





哒哒~


Reconstructed input data
$\hat{x}$
Decoder
Features
z
After training,
throw away decoder
Encoder
Input data
x

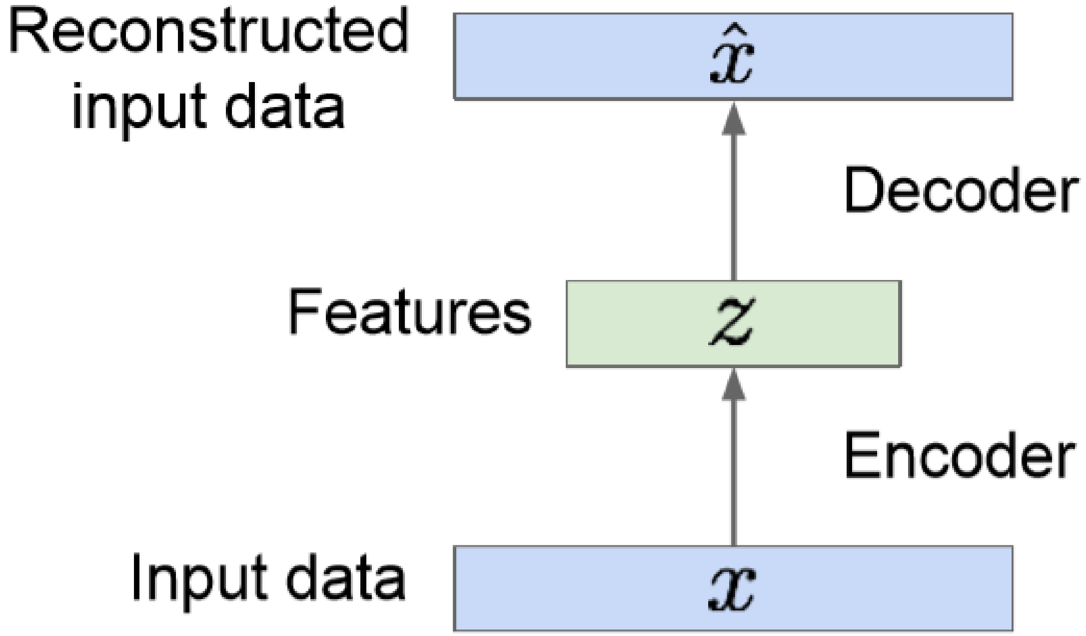
Some background first: Autoencoders



Encoder can be used to initialize a **supervised** model

Train for final task (sometimes with small data)

Some background first: Autoencoders



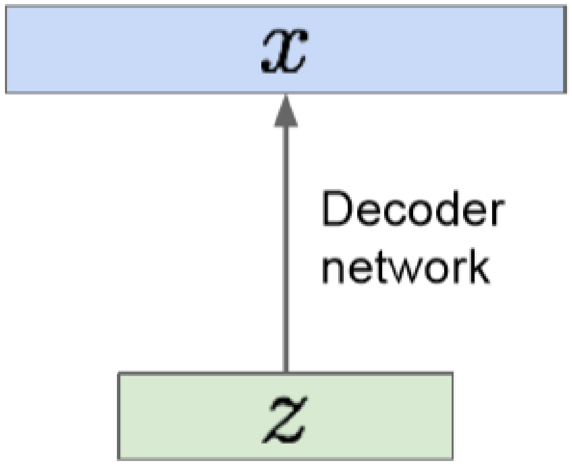
Autoencoders can reconstruct data, and can learn features to initialize a supervised model

Features capture factors of variation in training data. Can we generate new images from an autoencoder?

Variational Autoencoders

Sample from
true conditional
$p_{\theta^*}(x \mid z^{(i)})$

Sample from
true prior
$p_{\theta^*}(z)$



We want to estimate the true parameters θ^* of this generative model.

How to train the model?

Remember strategy for training generative models from FVBNs. Learn model parameters to maximize likelihood of training data

$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$

Q: What is the problem with this?

Intractable!

Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Variational Autoencoders: Intractability

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Posterior density also intractable: $p_\theta(z|x) = p_\theta(x|z) p_\theta(z) / p_\theta(x)$

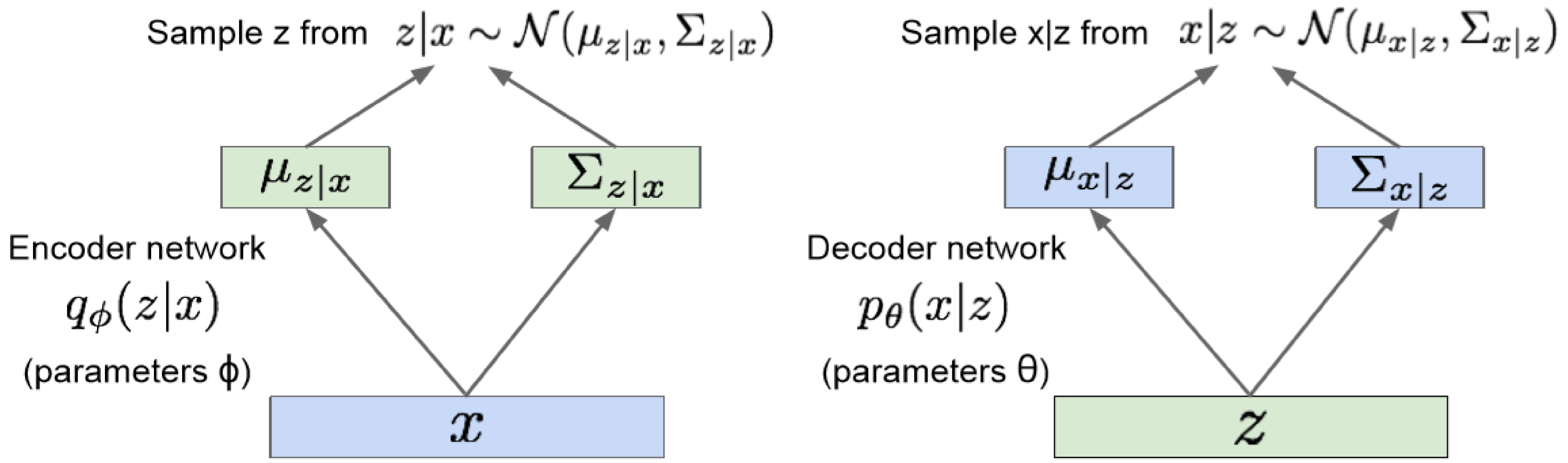
Solution: In addition to decoder network modeling $p_\theta(x|z)$, define additional encoder network $q_\phi(z|x)$ that approximates $p_\theta(z|x)$

Will see that this allows us to derive a lower bound on the data likelihood that is tractable, which we can optimize

Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Variational Autoencoders

Since we're modeling probabilistic generation of data, encoder and decoder networks are probabilistic



Encoder and decoder networks also called "recognition"/"inference" and "generation" networks

Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z)\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Bayes' Rule)}\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \qquad \text{(Multiply by constant)}\\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Logarithms)}\\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))
\end{aligned}
$$

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Bayes' Rule})$$

Make approximate posterior distribution close to prior

Reconstruct the input data

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad (\text{Multiply by constant})$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Logarithms})$$

$$= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}$$

$$\boxed{\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)},\theta,\phi)}$$

Variational lower bound ("ELBO")

$$\boxed{\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)},\theta,\phi)}$$

Training: Maximize lower bound

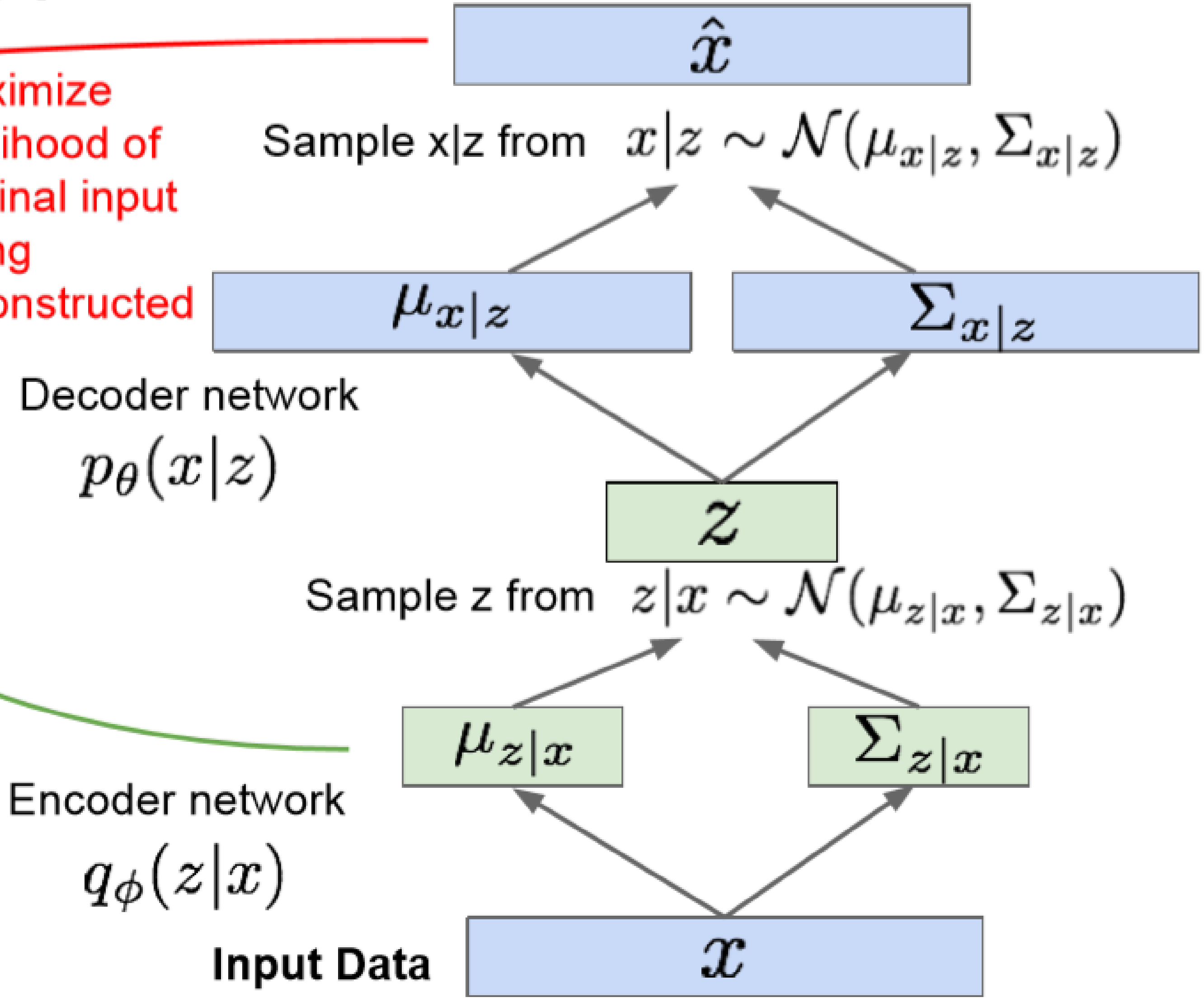
Variational Autoencoders

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z \left[\log p_\theta(x^{(i)} \mid z) \right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

Maximize likelihood of original input being reconstructed

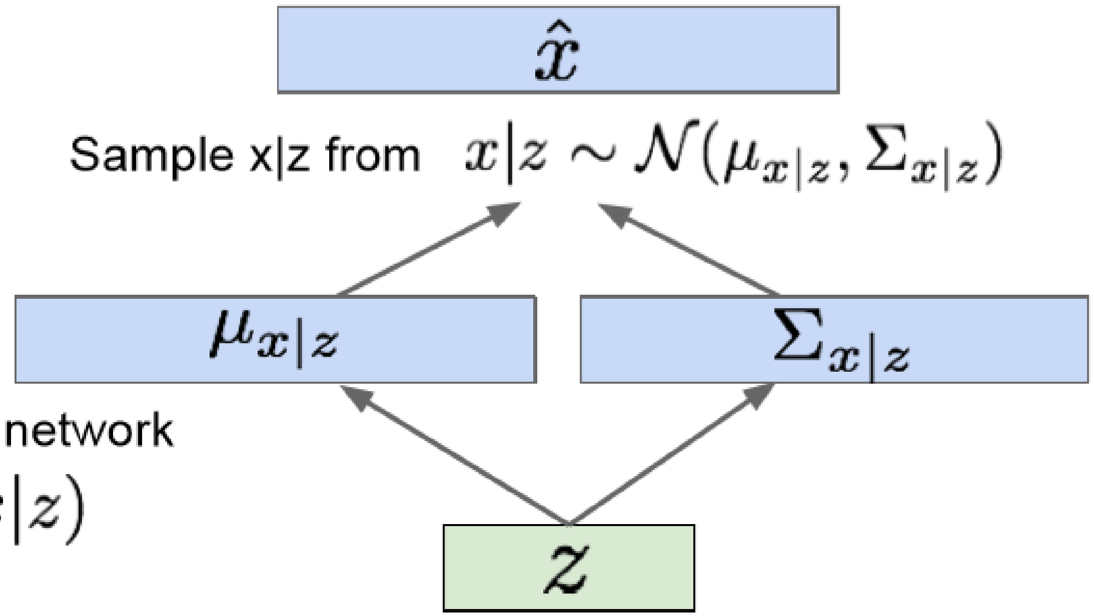
Make approximate posterior distribution close to prior



For every minibatch of input data: compute this forward pass, and then backprop!

Variational Autoencoders: Generating Data!

Use decoder network. Now sample z from prior!

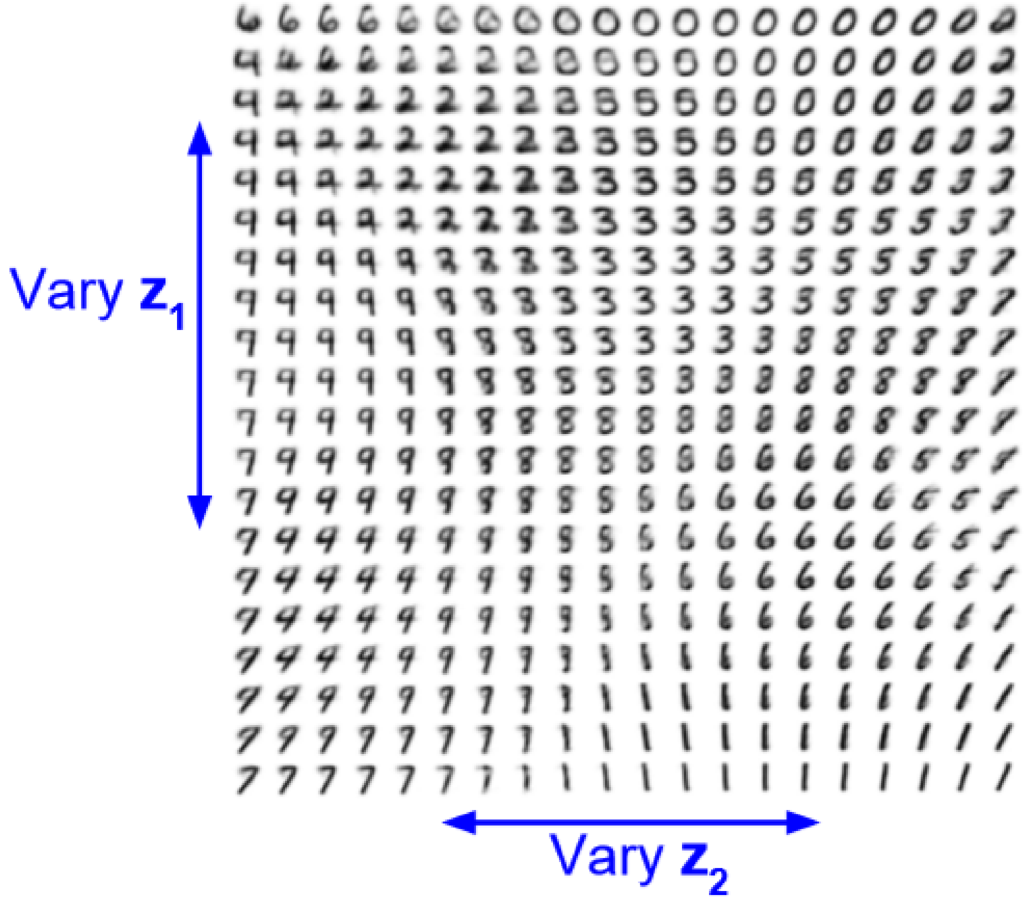


Decoder network

$p_\theta(x|z)$

Sample z from $z \sim \mathcal{N}(0, I)$

Data manifold for 2-d **z**



Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Variational Autoencoders: Generating Data!

32x32 CIFAR-10

Labeled Faces in the Wild

β-VAE

β-VAE

- VAE loss $$L = E_{q(z|x)}[logp(x|z)] - KL[q(z|x)||p(z)]$$

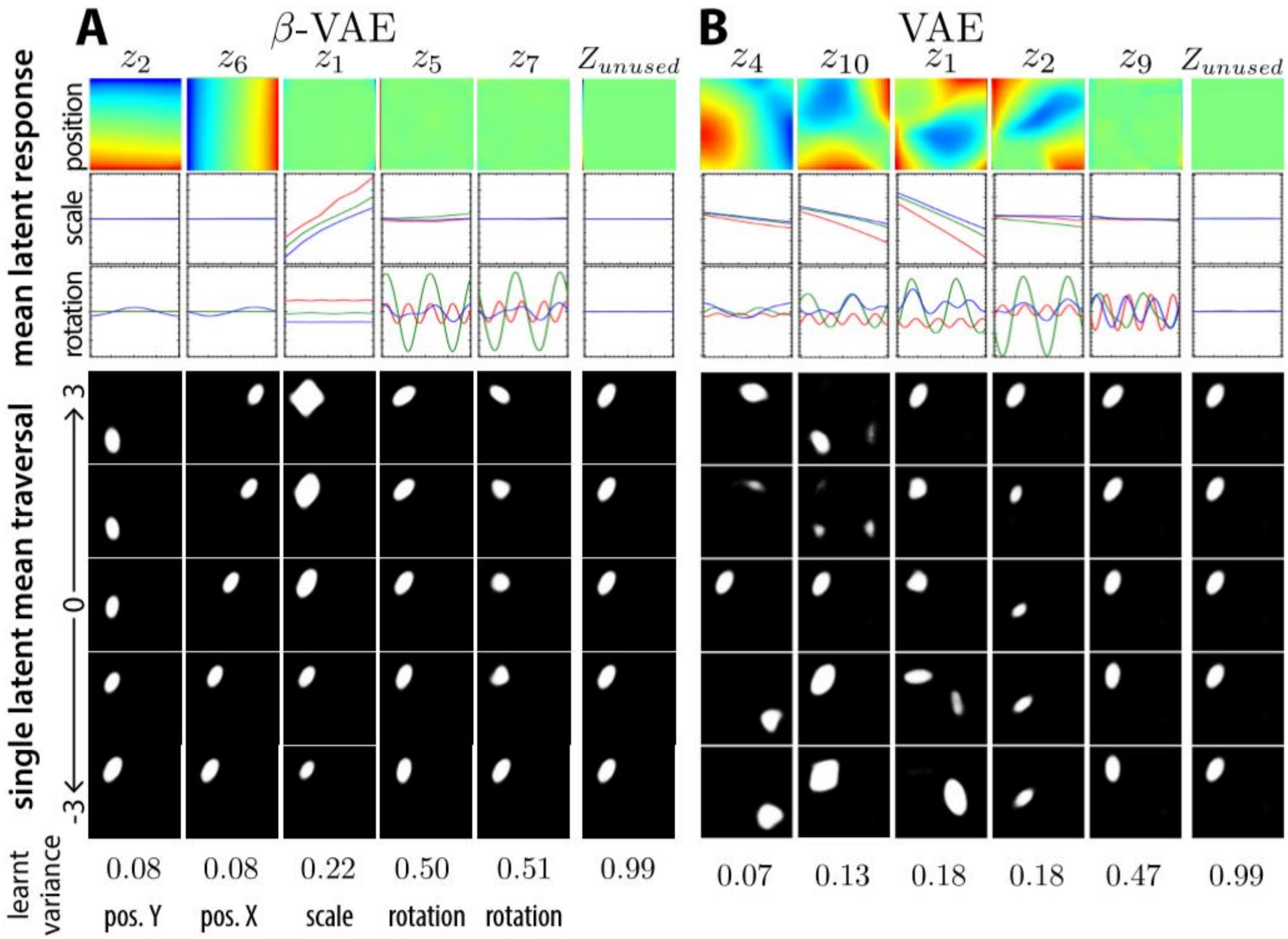
- β-VAE Loss $$L = E_{q(z|x)}[logp(x|z)] - \beta * KL[q(z|x)||p(z)]$$

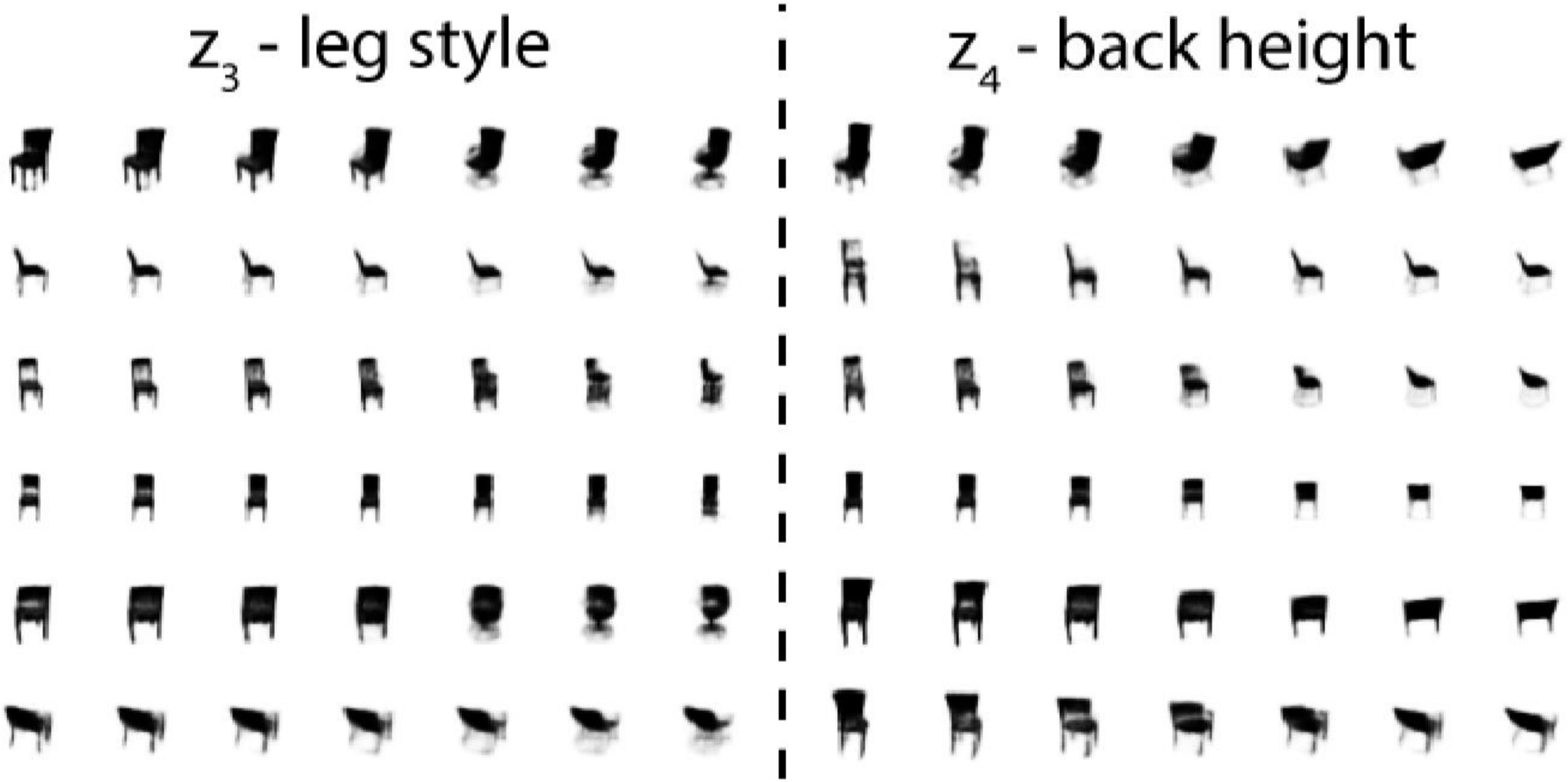
- stronger constraint on the latent bottleneck
- limits the representation capacity of z.

β-VAE

- encourages more efficient latent encoding
- further encourages the disentanglement

- create a trade-off between reconstruction quality and the extent of disentanglement.


A
β-VAE
B
VAE
z_2 z_6 z_1 z_5 z_7 Z_{unused}
z_4 z_{10} z_1 z_2 z_9 Z_{unused}
mean latent response
position
scale
rotation
single latent mean traversal
3
0
-3
learnt variance
0.08 0.08 0.22 0.50 0.51 0.99
0.07 0.13 0.18 0.18 0.47 0.99
pos. Y
pos. X
scale
rotation
rotation


z_3 - leg style
z_4 - back height

Deep Spatial Autoencoder

Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, Pieter Abbeel 2016

Loss Func

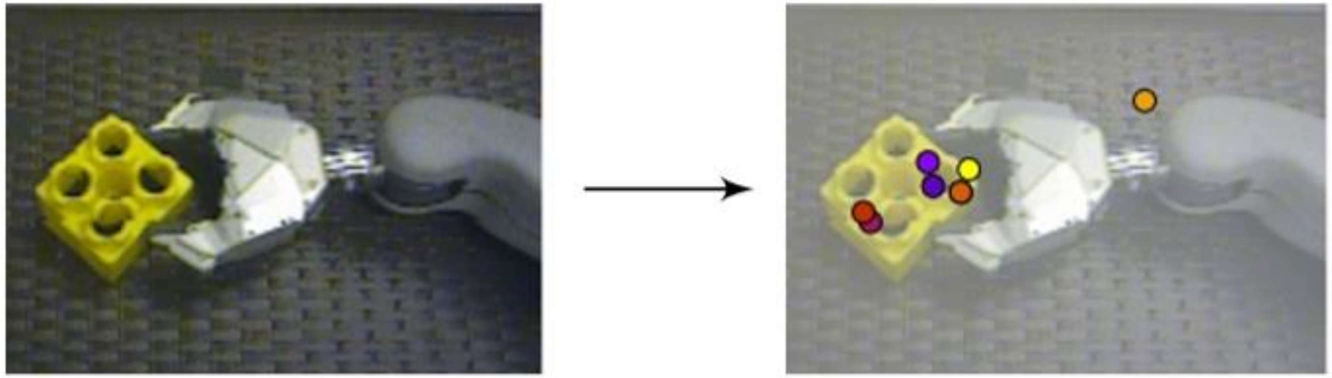
[illegible] [illegible], $g_{\text{slow}}(\mathbf{f}_t) = ||(\mathbf{f}_{t+1} - \mathbf{f}_t) - (\mathbf{f}_t - \mathbf{f}_{t-1})||_2^2$, to encourage the feature points to slowly change velocity. As a result, the overall autoencoder objective becomes:

$$\mathcal{L}_{\text{DSAE}} = \sum_{t,k} ||I_{\text{downsamp},k,t} - h_{\text{dec}}(\mathbf{f}_{k,t})||_2^2 + g_{\text{slow}}(\mathbf{f}_{k,t})$$

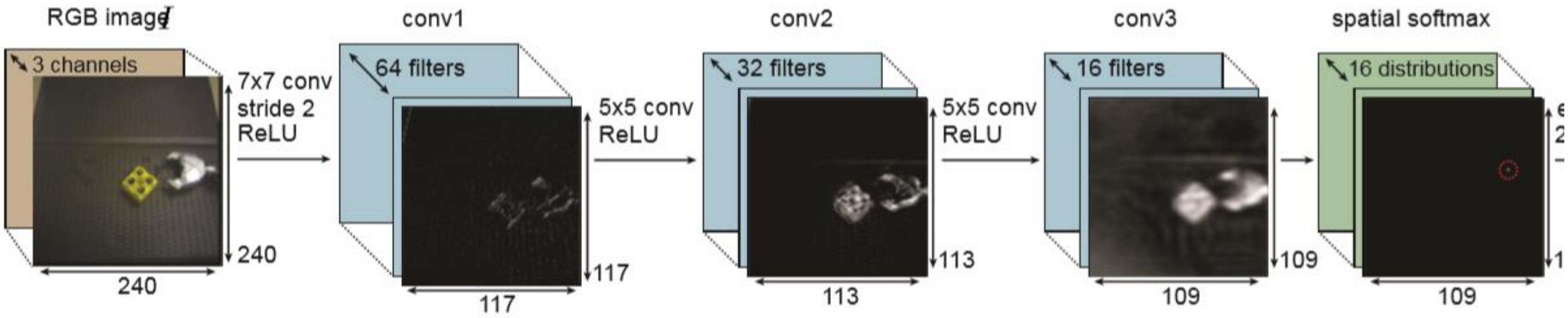
$\|x - \hat{x}\|^2$ $\qquad\qquad$ a

Framework

Train controller using $\tilde{\mathbf{x}}_t$; collect image data → Learn visual features → Train final controller with new state $[\tilde{\mathbf{x}}_t; \mathbf{f}_t]$

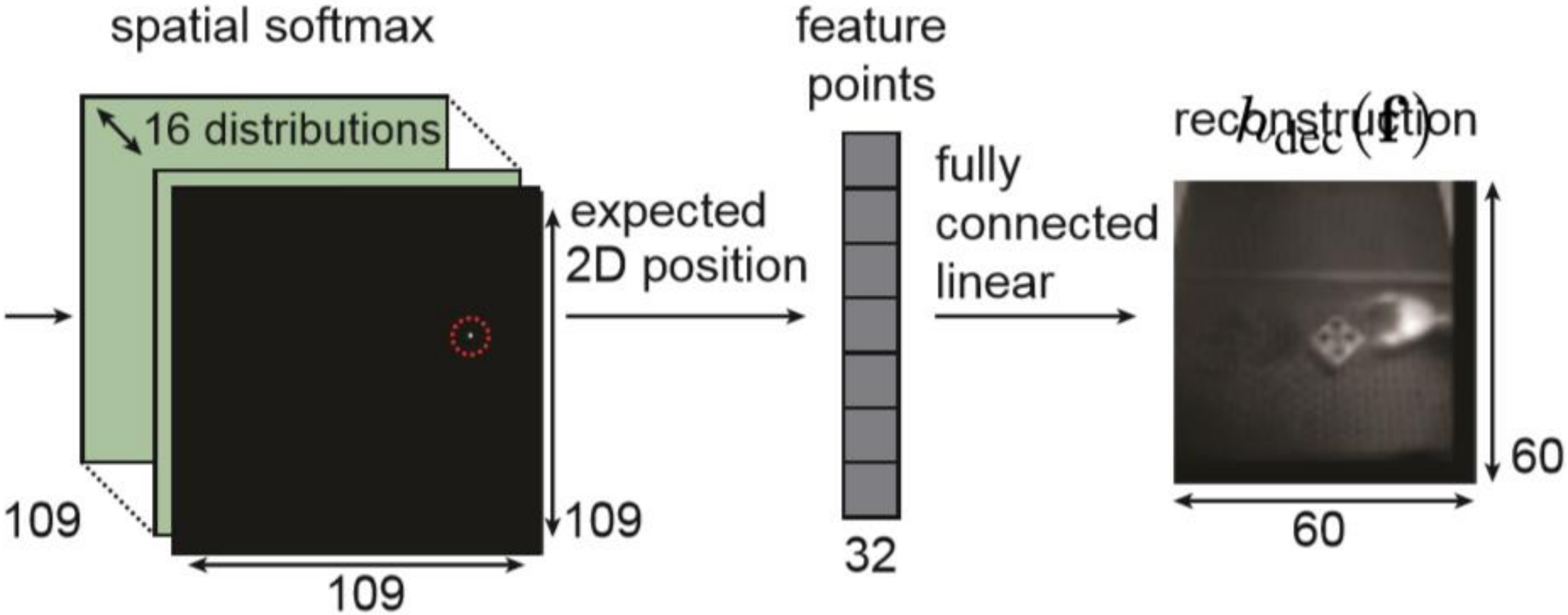


Normal CNN



$$\text{softmax} \quad s_{cij} = e^{a_{cij}/\alpha} / \sum_{i'j'} e^{a_{ci'j'}/\alpha},$$

Feature Points

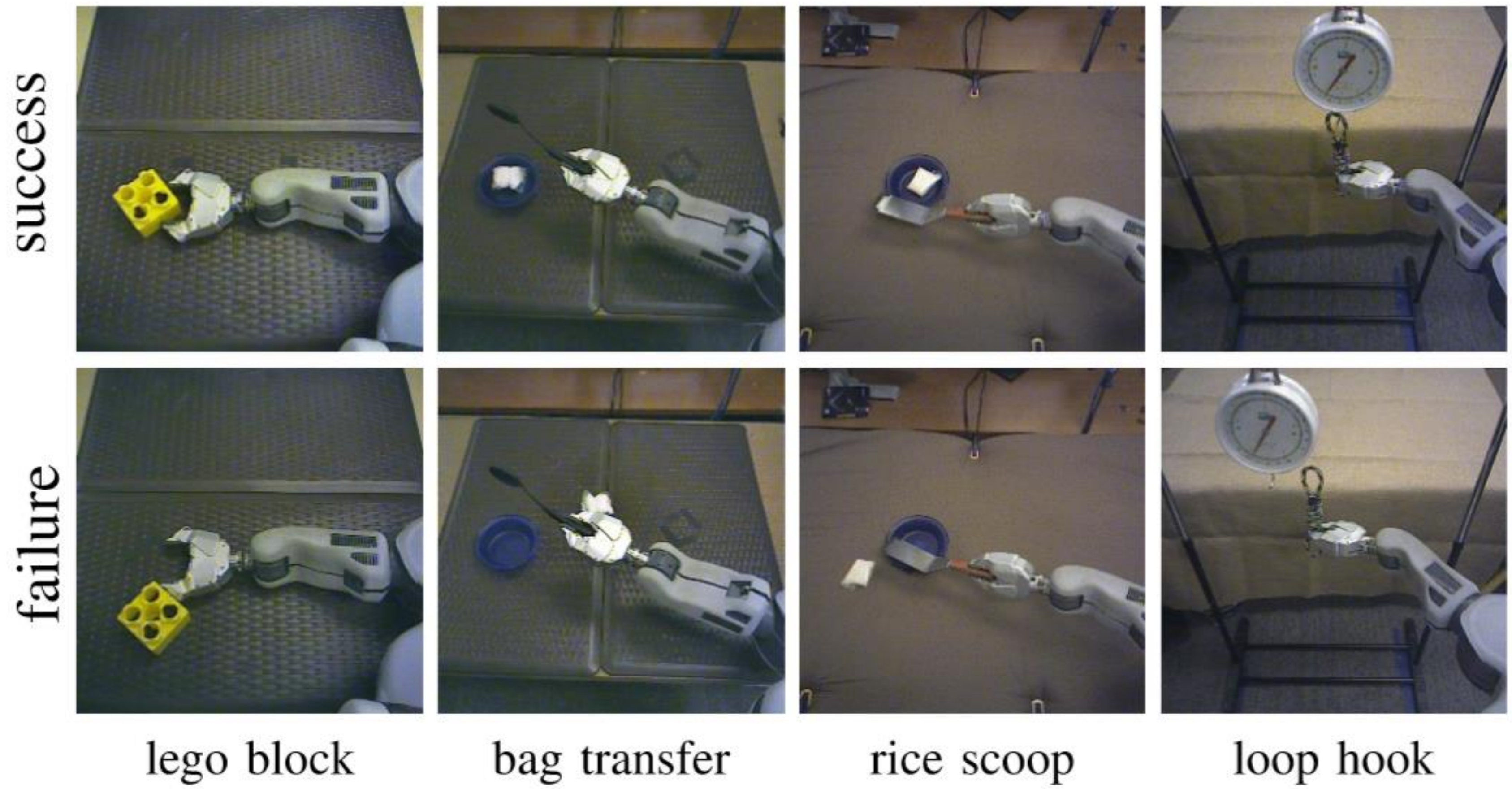


$$\mathbf{f}_c = \left(\sum_i i * s_{cij}, \sum_j j * s_{cij}\right)$$

State

- State =
  - EE-pos + Feature points

Tasks


success
failure
lego block
bag transfer
rice scoop
loop hook

Thanks