Machine Learning course,
advansed track

# Lecture 15: Generative models

**Radoslav Neychev**

MIPT
13.12.2019

# References

This lecture structure was inspired a lot by the series of Habr posts on Autoencoders and GANs in Keras

# Outline

- Generative models overview
- Simple Autoencoders (not generative models!)
- Variational Autoencoders (VAE)
  - Conditional VAE
- Generative Adversarial Networks
  - Conditional GAN

And a lot of gif images.

# Generative models taxonomy

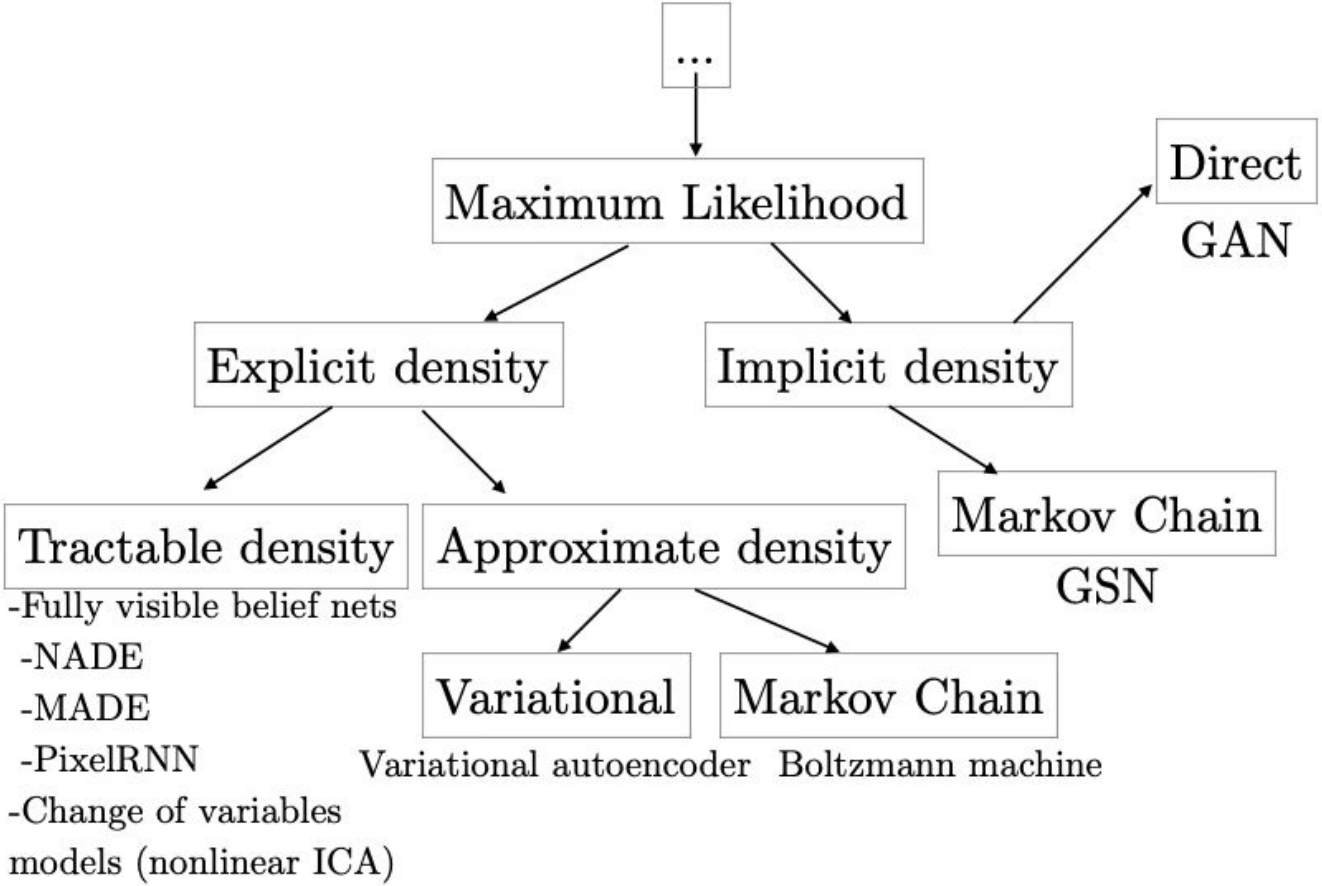



Image source: [Tutorial on Generative Adversarial Networks, Ian Goodfellow, NIPS, 2016]

# Generative models taxonomy

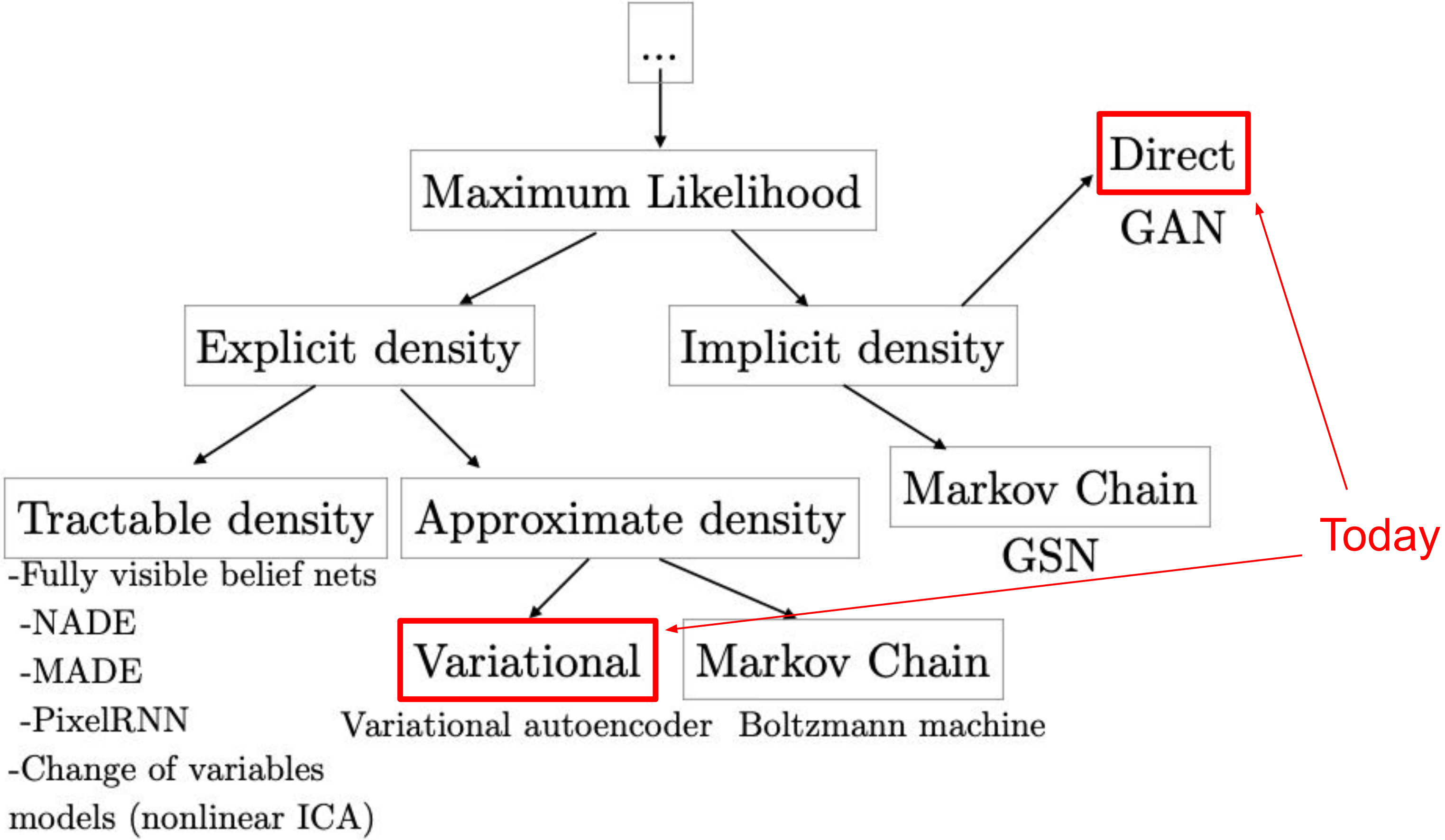



Image source: [Tutorial on Generative Adversarial Networks, Ian Goodfellow, NIPS, 2016]

# Autoencoders

Denote **z** as encoded with encoder E input **x**

$$\mathbf{z} = E(\mathbf{x}, \boldsymbol{\theta}_E)$$

Decoder D recovers **x** from latent representation

$$\hat{\mathbf{x}} = D(\mathbf{z}, \boldsymbol{\theta}_D)$$

Optimal parameters learned w.r.t. loss function L

$$[\boldsymbol{\theta}_E, \boldsymbol{\theta}_D] = \arg\min L(\hat{\mathbf{x}}, \mathbf{x})$$

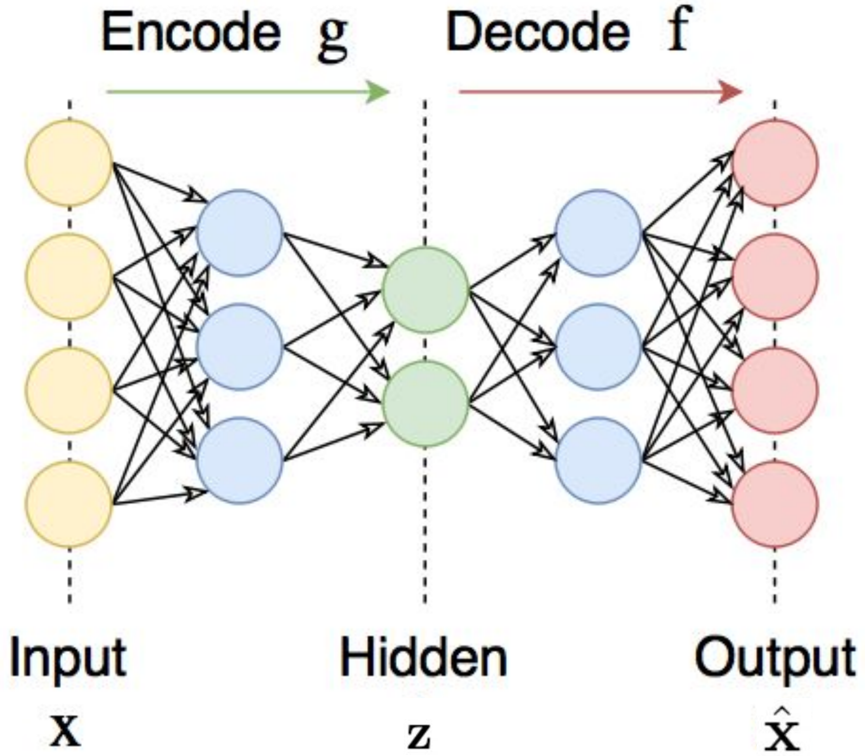



Image source: [Habr post on autoencoders and GANs]

# Autoencoders

Denote **z** as encoded with encoder E input **x**

$$\mathbf{z} = E(\mathbf{x}, \boldsymbol{\theta}_E)$$

Decoder D recovers **x** from latent representation

$$\hat{\mathbf{x}} = D(\mathbf{z}, \boldsymbol{\theta}_D)$$

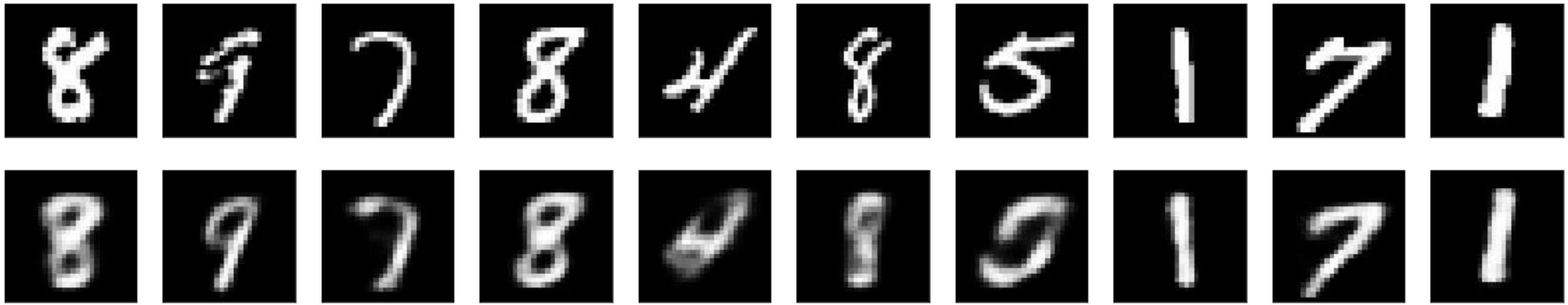
Simple example: PCA

Optimal parameters learned w.r.t. loss function L

$$[\boldsymbol{\theta}_E, \boldsymbol{\theta}_D] = \arg\min L(\hat{\mathbf{x}}, \mathbf{x})$$

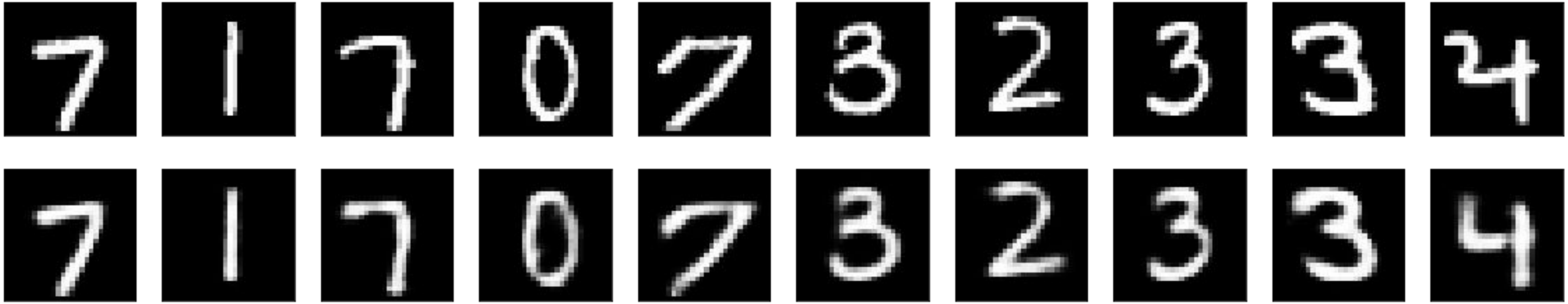
Image source: [Habr post on autoencoders and GANs]()

# PCA performance on MNIST



16 components

# Convolutional performance on MNIST

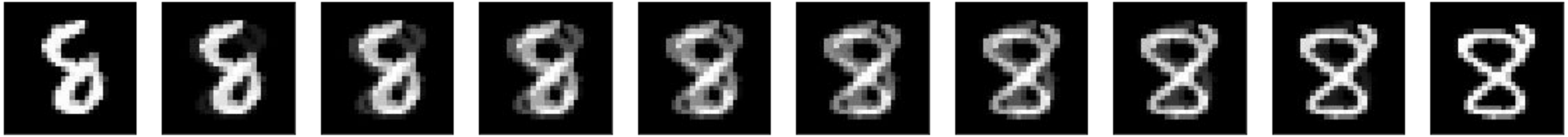


7 x 7 latent space

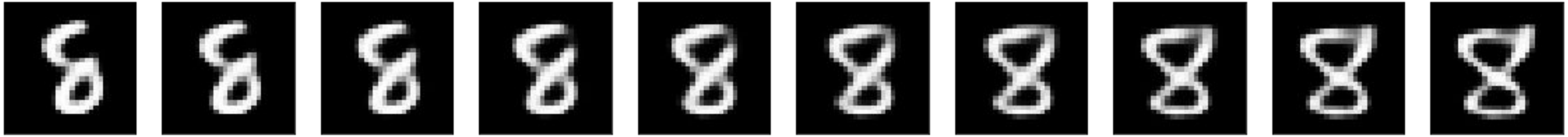
# Homotopy between samples

10 steps between samples

- In original feature space (28 x 28):



- In latent space (7 x 7):



# Latent space structure

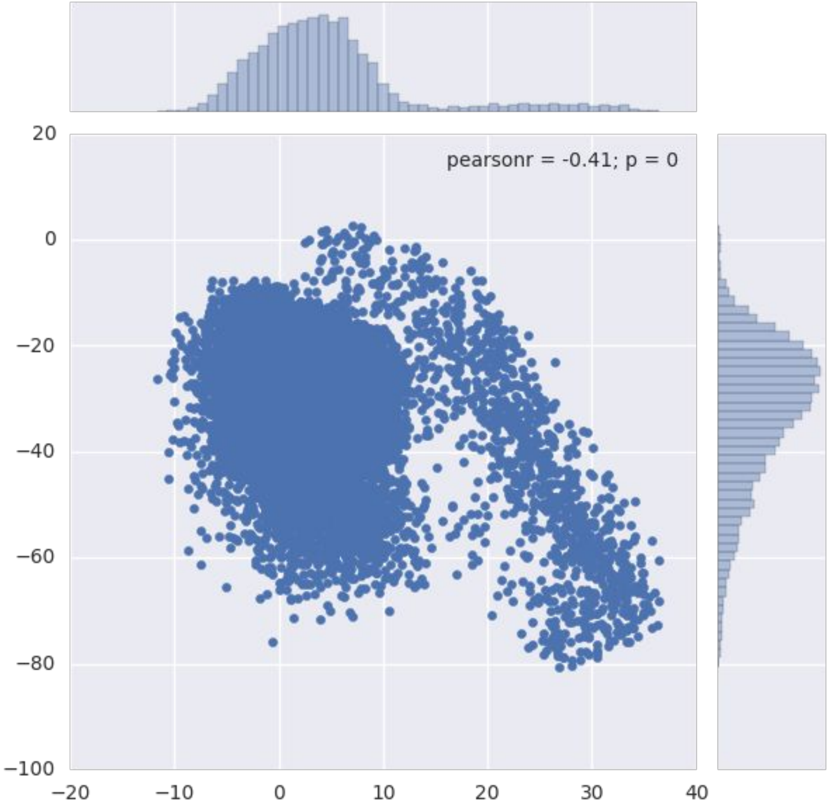



Image source: [Habr post on autoencoders and GANs]

# VAE intuition

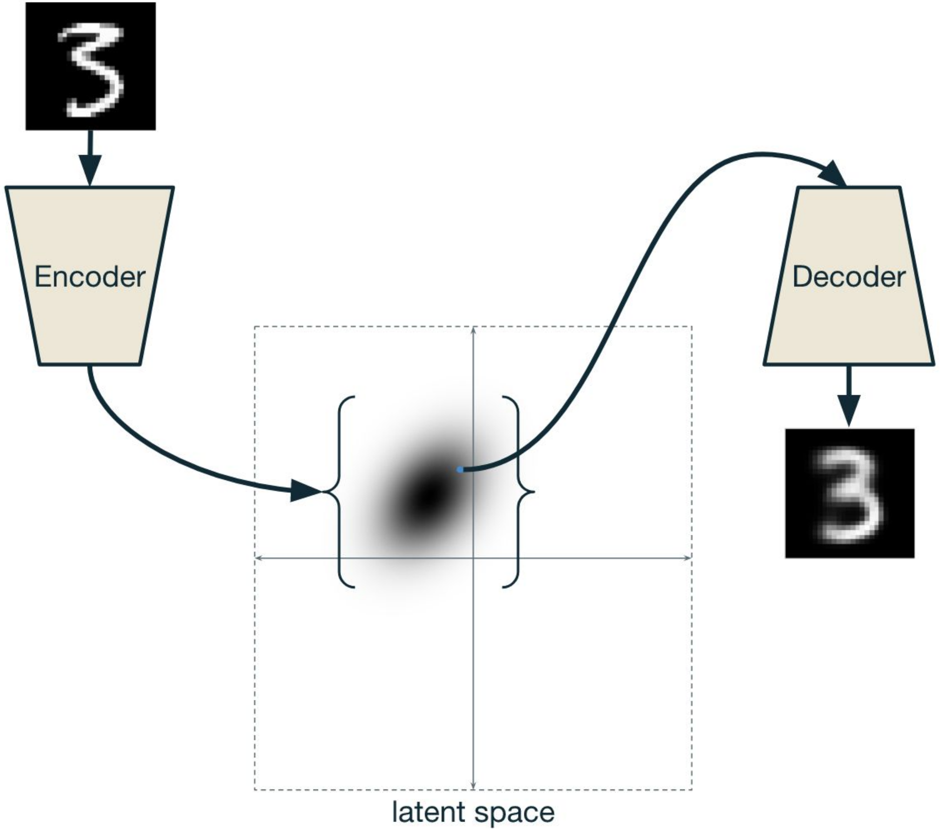



Image source: [Conditional Variational Autoencoders by Isaac Dykeman, 2016]

# KL divergence

Denote distributions $Q(z)$ and $P(z|X)$.

Kullback–Leibler divergence is defined as

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(z|X)\right]$$

# KL divergence

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(z|X)\right]$$

Applying the Bayes rule:

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(X|z) - \log P(z)\right] + \log P(X)$$

# KL divergence

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(z|X)\right]$$

Applying the Bayes rule:

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(X|z) - \log P(z)\right] + \log P(X)$$

$$\log P(X) - \mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z)\|P(z)\right]$$

# KL divergence

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(z|X)\right]$$

Applying the Bayes rule:

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(X|z) - \log P(z)\right] + \log P(X)$$

$$\log P(X) - \mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z)\|P(z)\right]$$

$$\boxed{\log P(X) - \mathcal{D}\left[Q(z|X)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z|X)\|P(z)\right]}$$

*This equation is the core of Variational Autoencoders*

# Structure of the latent space

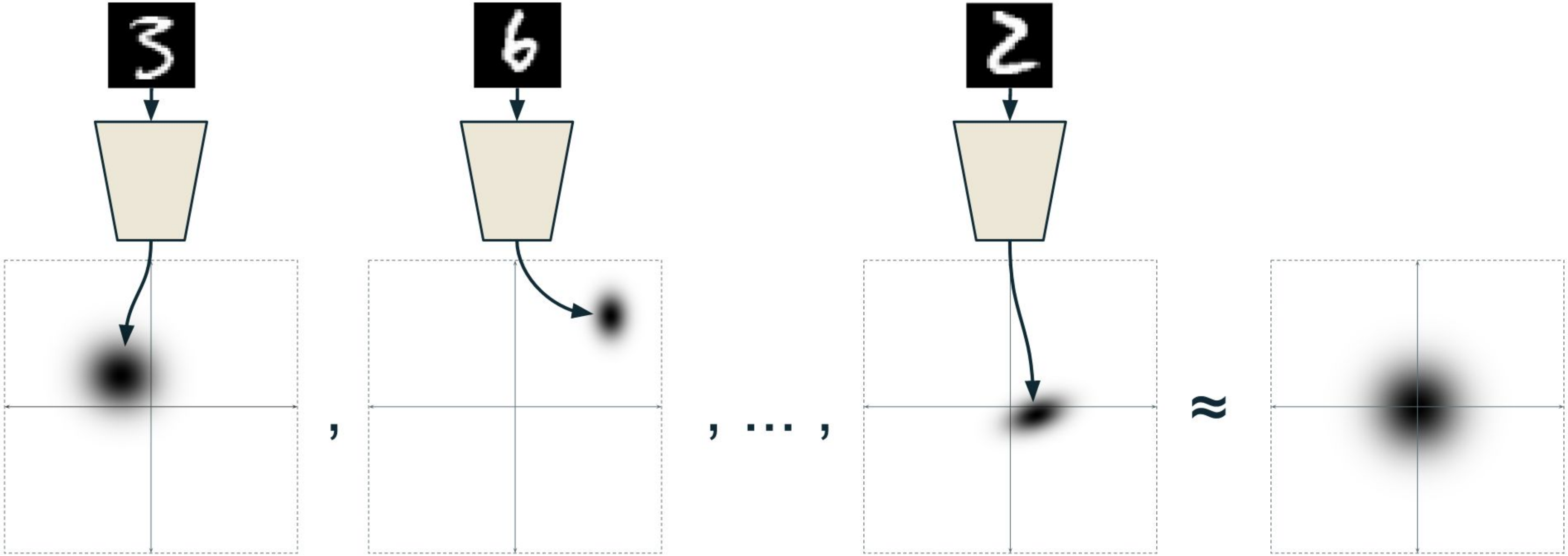



Image source: [Conditional Variational Autoencoders by Isaac Dykeman, 2016]

# VAE so far

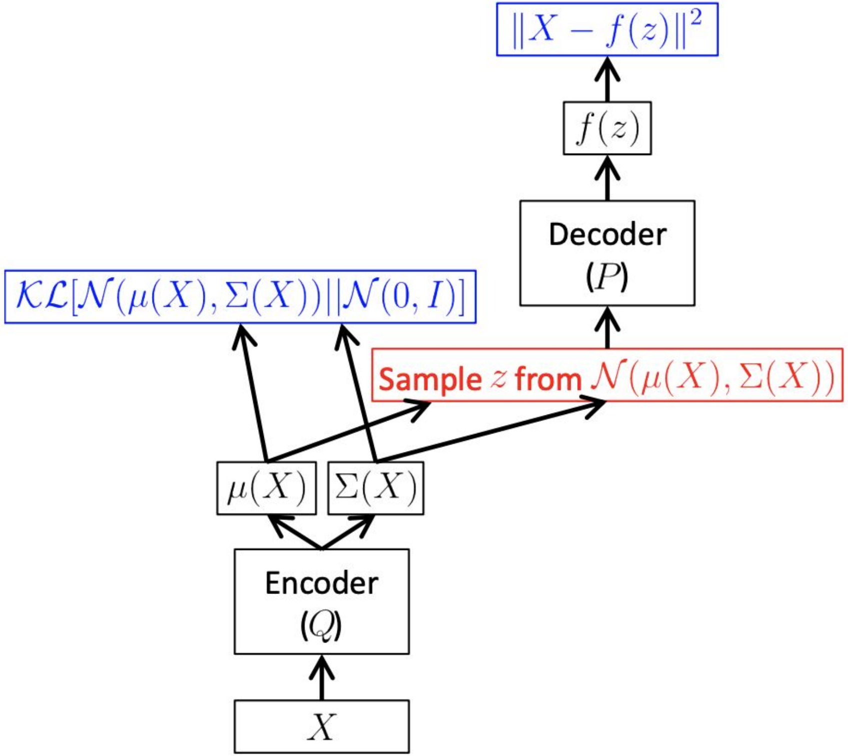



Image source: [Tutorial on Variational Autoencoders by Carl Doersch, 2016]

# VAE so far

$$\mathcal{D}[\mathcal{N}(\mu(X),\Sigma(X))\|\mathcal{N}(0,I)] = \frac{1}{2}\left(\mathrm{tr}\left(\Sigma(X)\right) + (\mu(X))^\top(\mu(X)) - k - \log\det\left(\Sigma(X)\right)\right)$$

*Try to derive it by yourself*

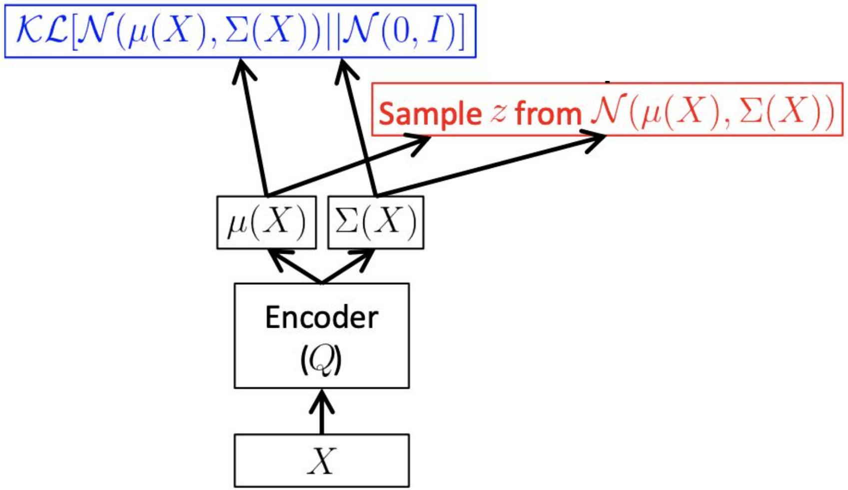





Image source: [Tutorial on Variational Autoencoders by Carl Doersch, 2016]

# VAE so far

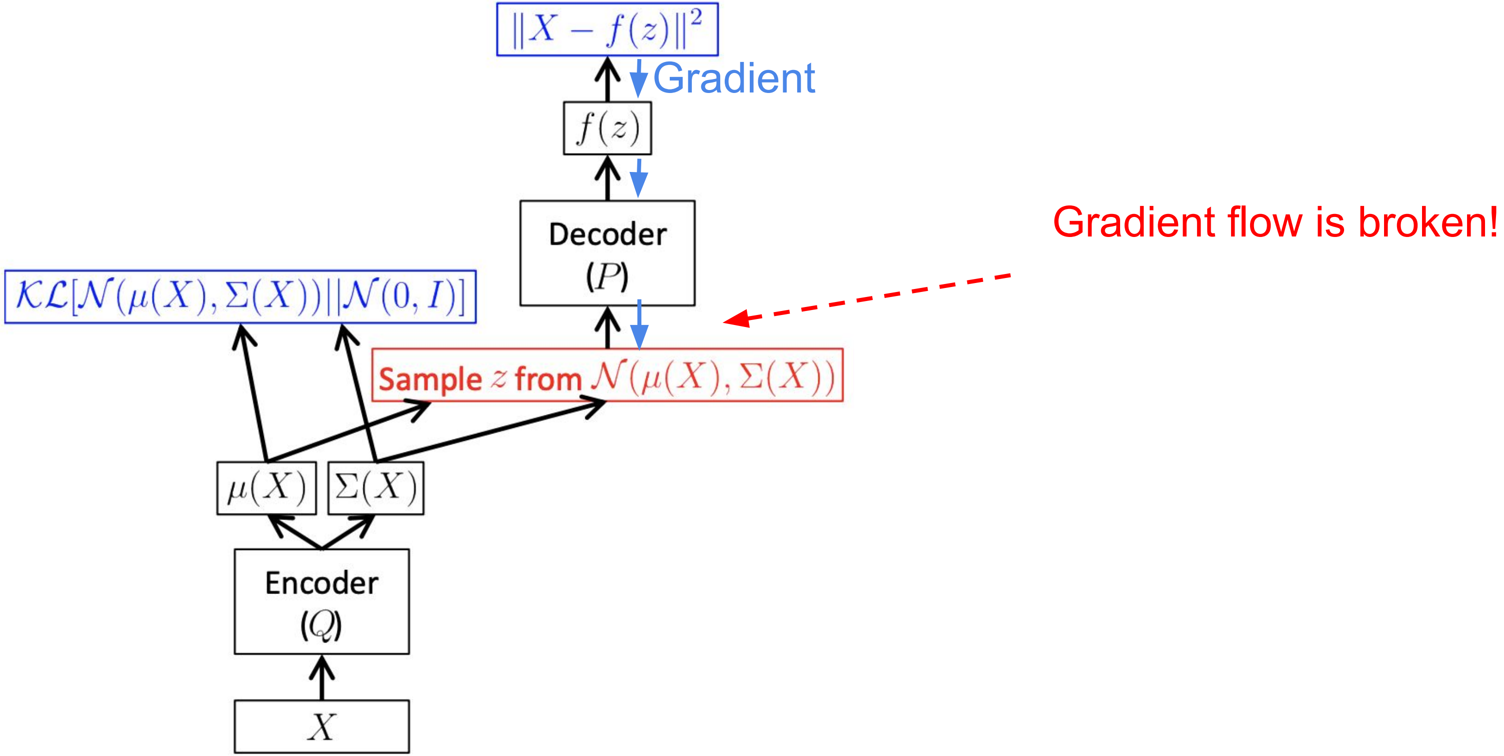
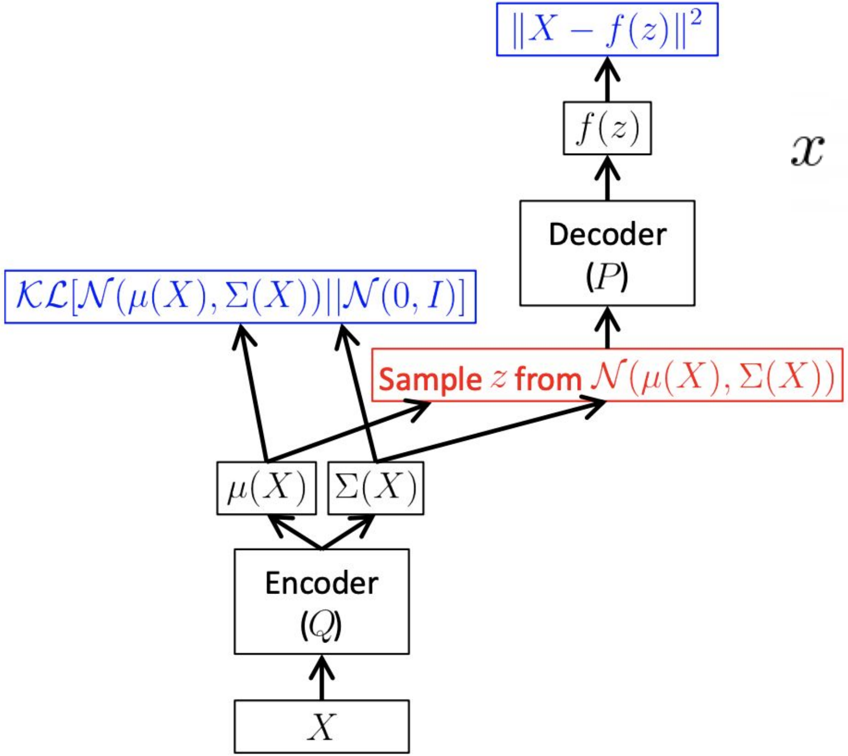



Image source: [Tutorial on Variational Autoencoders by Carl Doersch, 2016]

# Reparametrization trick




$$x \sim \mathcal{N}(\mu, \sigma^2), \quad z = \frac{x - \mu}{\sigma} \sim \mathcal{N}(0, 1)$$

$$\Rightarrow x = \sigma(z + \mu) \sim \mathcal{N}(\mu, \sigma^2)$$

Image source: Tutorial on Variational Autoencoders by Carl Doersch, 2016

# Reparametrization trick

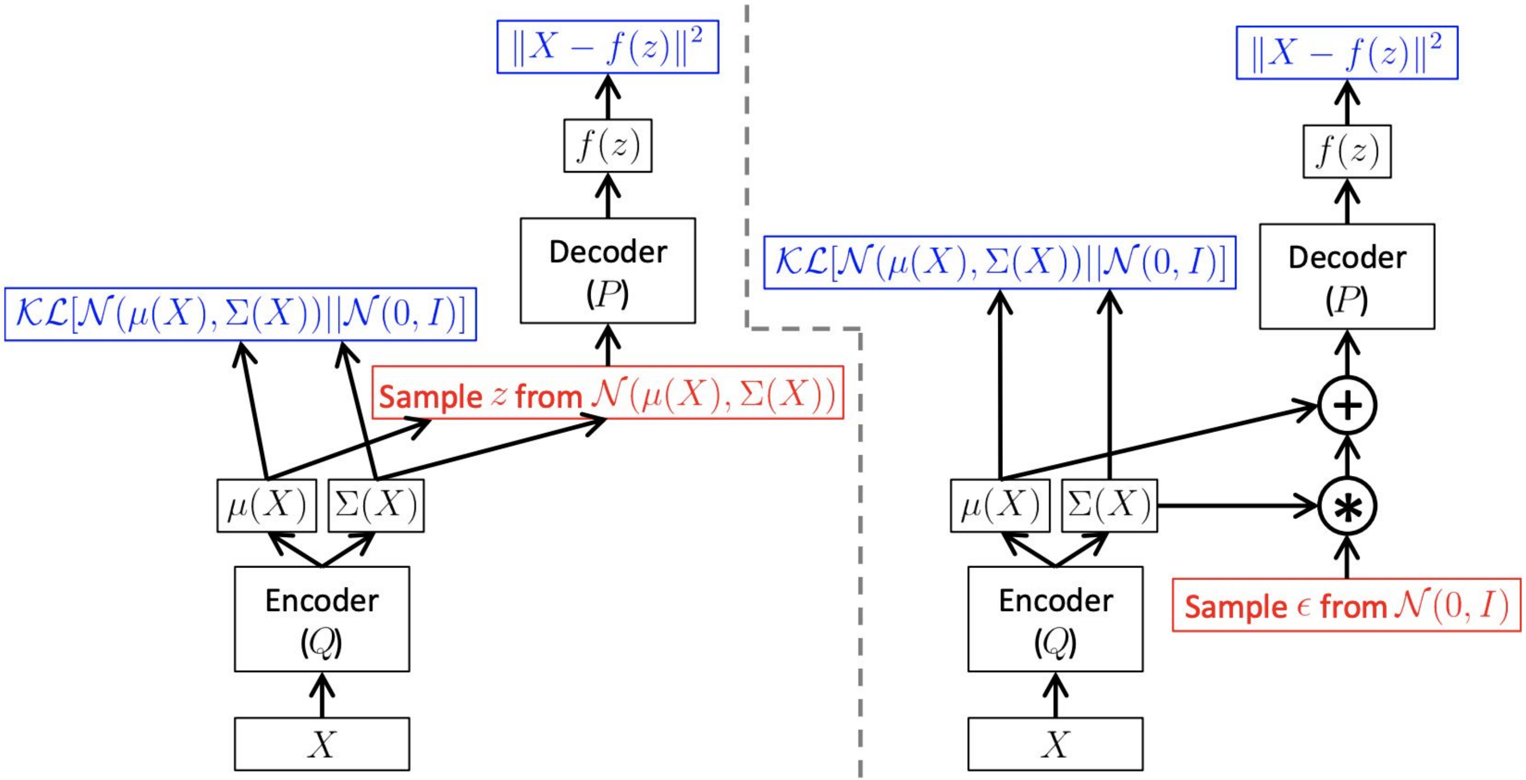
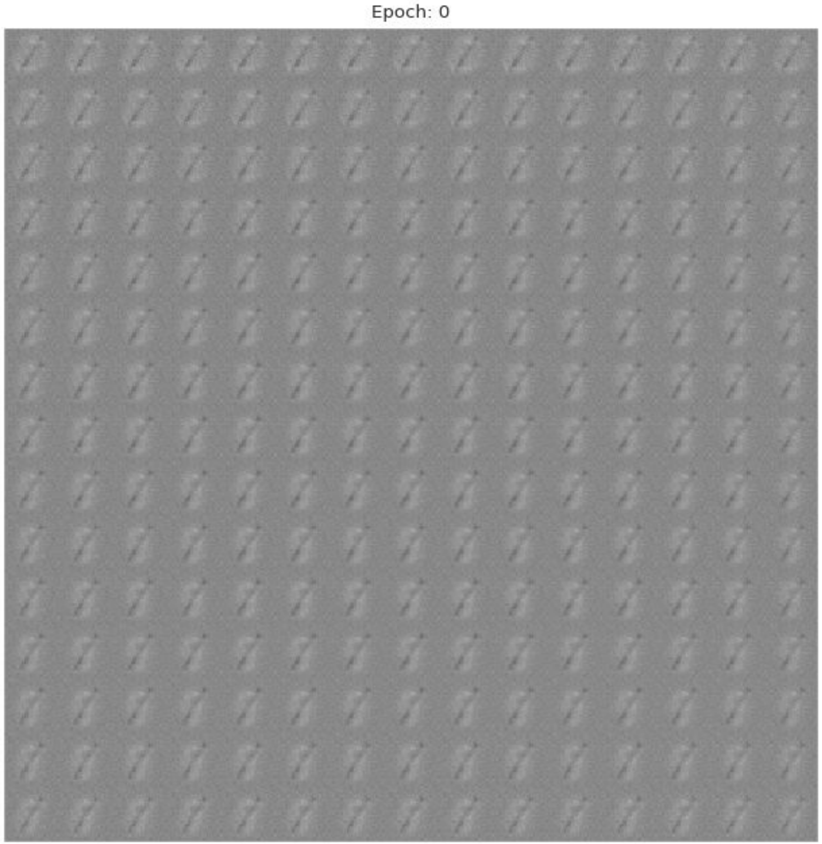





Image source: Tutorial on Variational Autoencoders by Carl Doersch, 2016

# VAE manifold



# Structure of the latent space

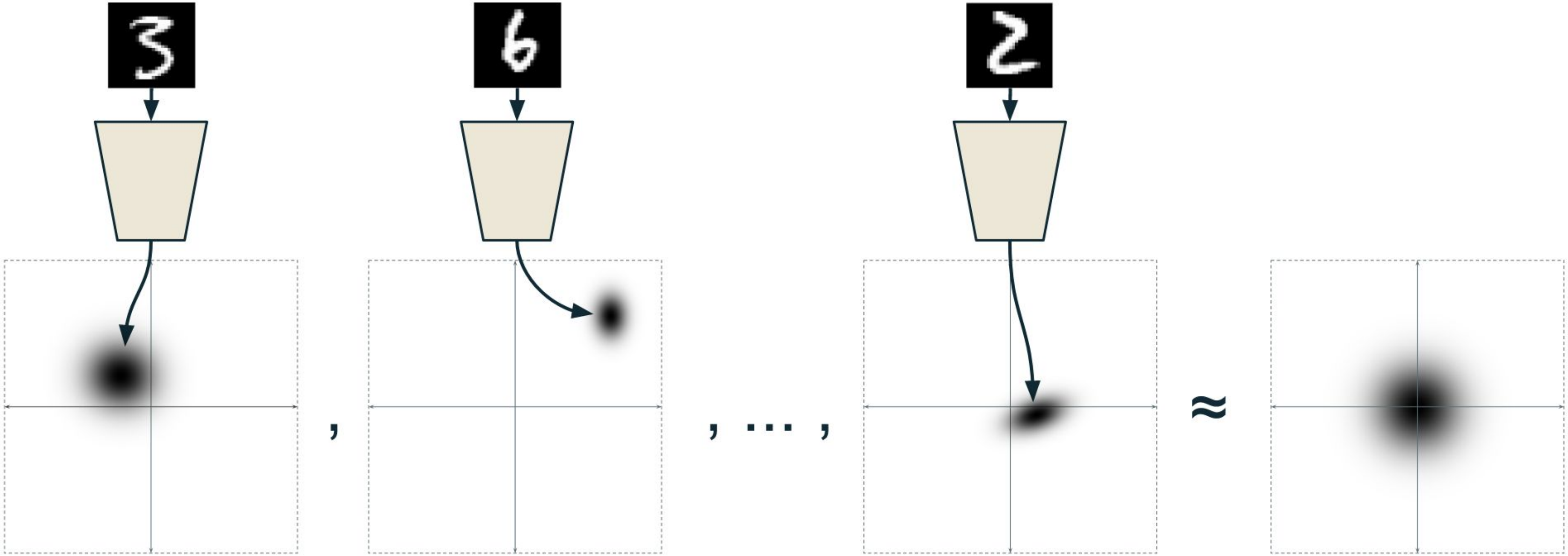
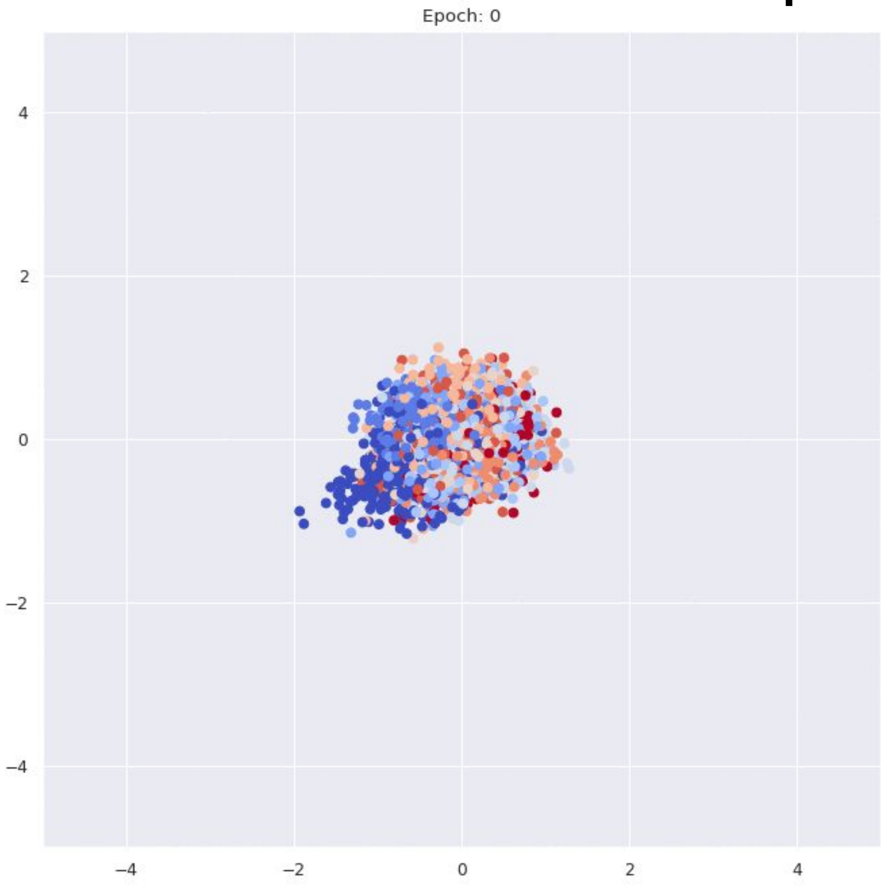




Image source: [Conditional Variational Autoencoders by Isaac Dykeman, 2016]

# VAE latent space distribution



# Conditional VAE intuition

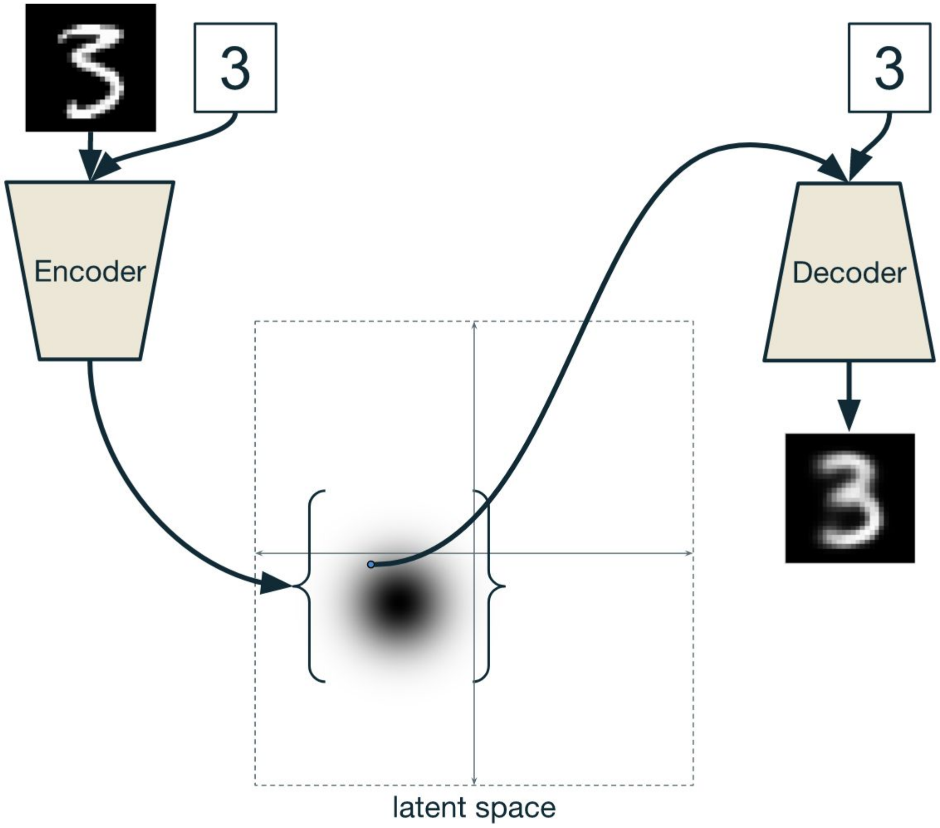





Image source: [Conditional Variational Autoencoders by Isaac Dykeman, 2016]

# Conditional VAE

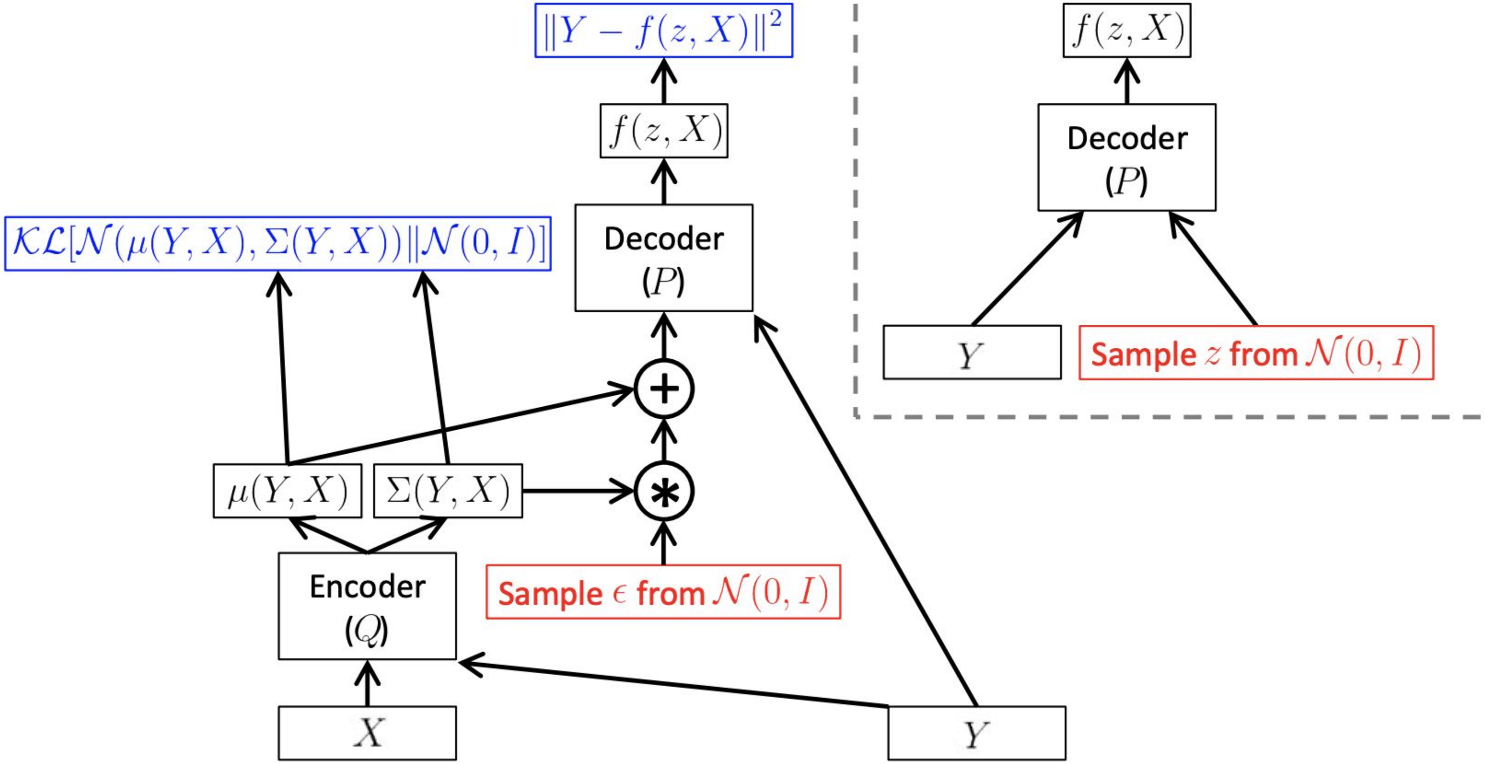
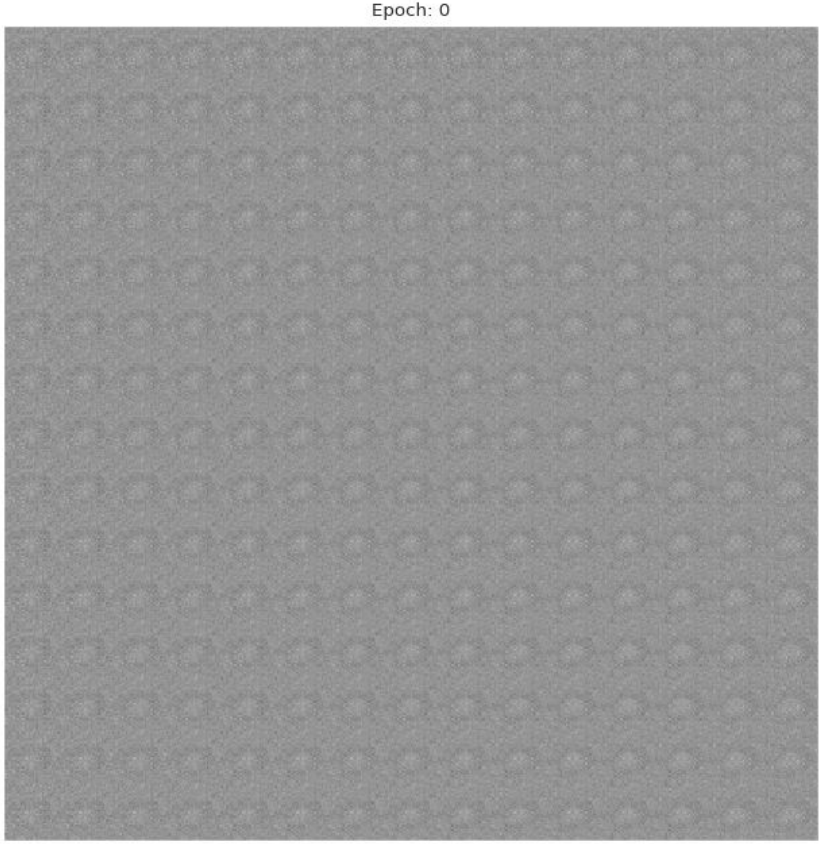
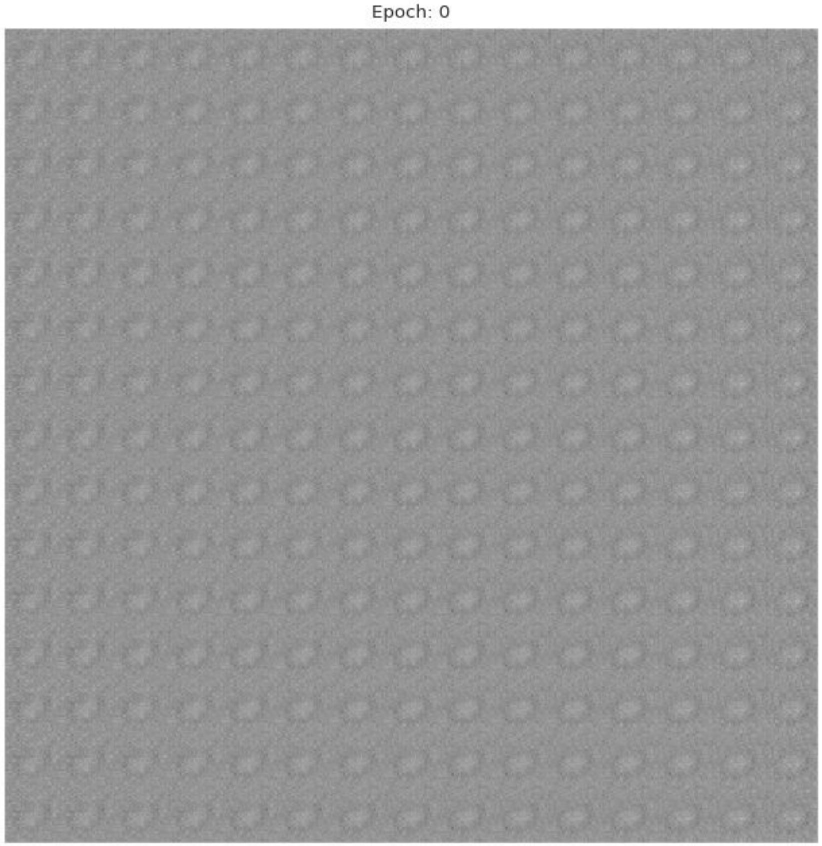



Image source: [Tutorial on Variational Autoencoders by Carl Doersch, 2016]

# cVAE manifold



# cVAE manifold



# Transferring style with cVAE

Seed label is 1

# Once again

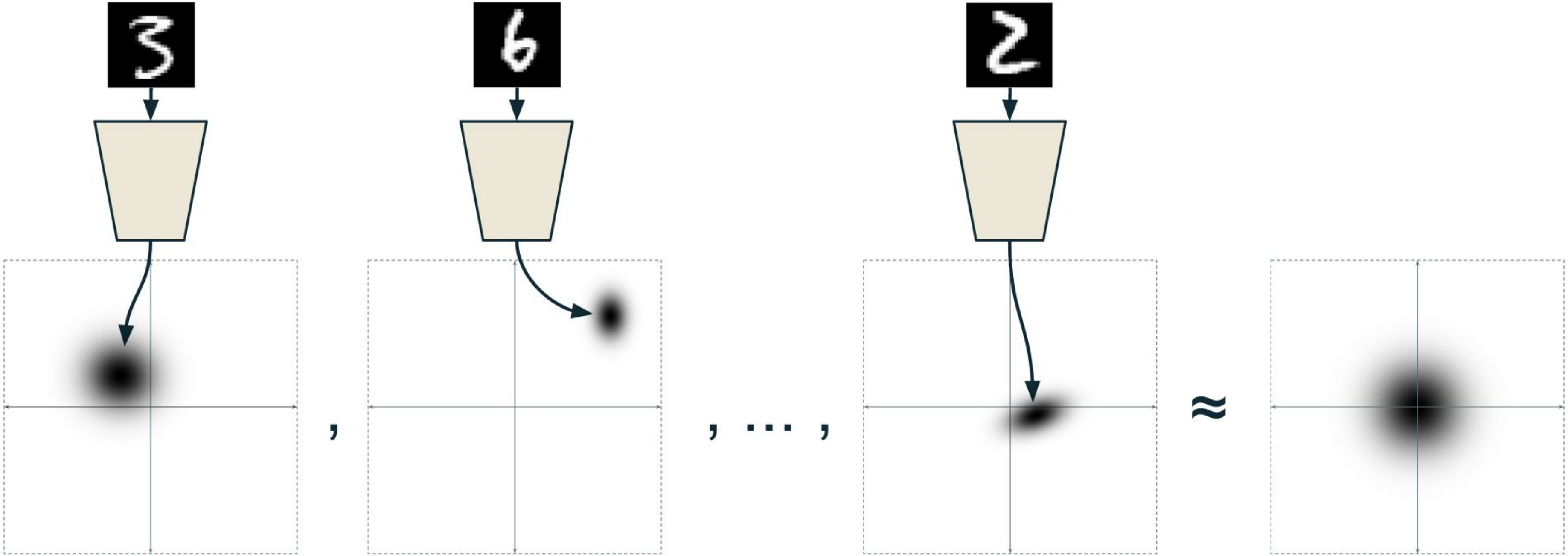
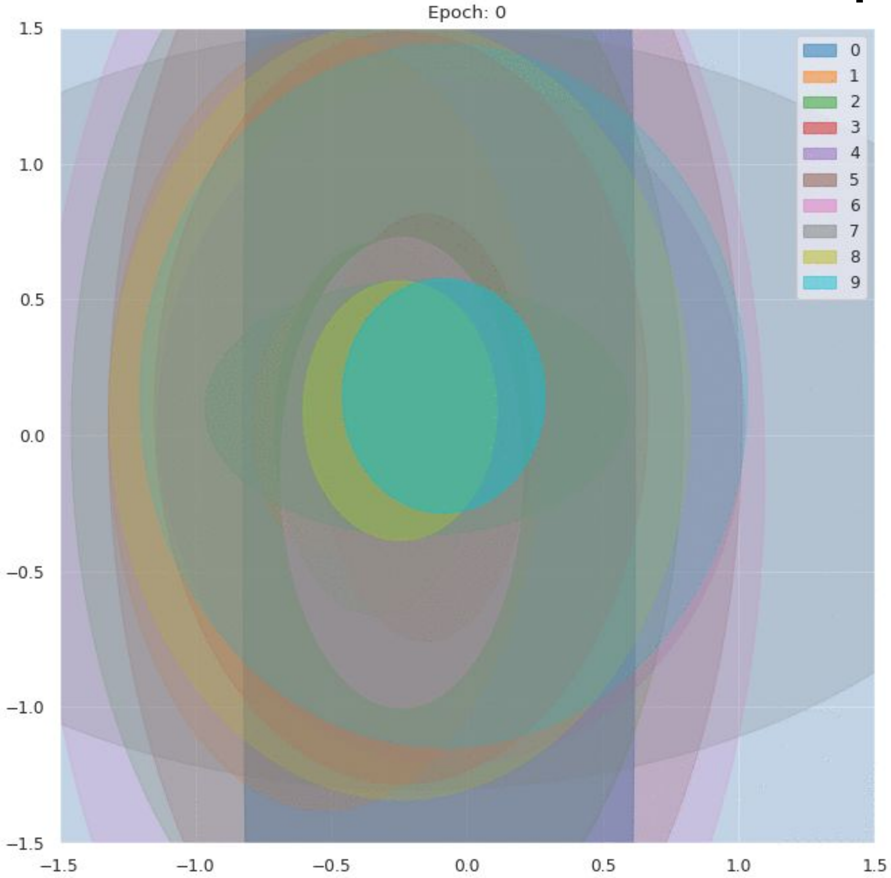



Image source: [Conditional Variational Autoencoders by Isaac Dykeman, 2016]

# cVAE latent space distribution



# GAN

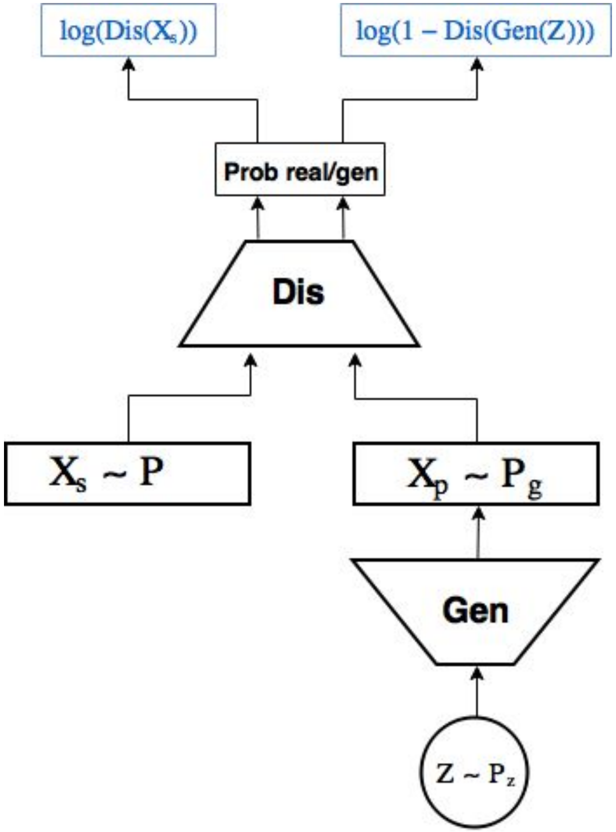
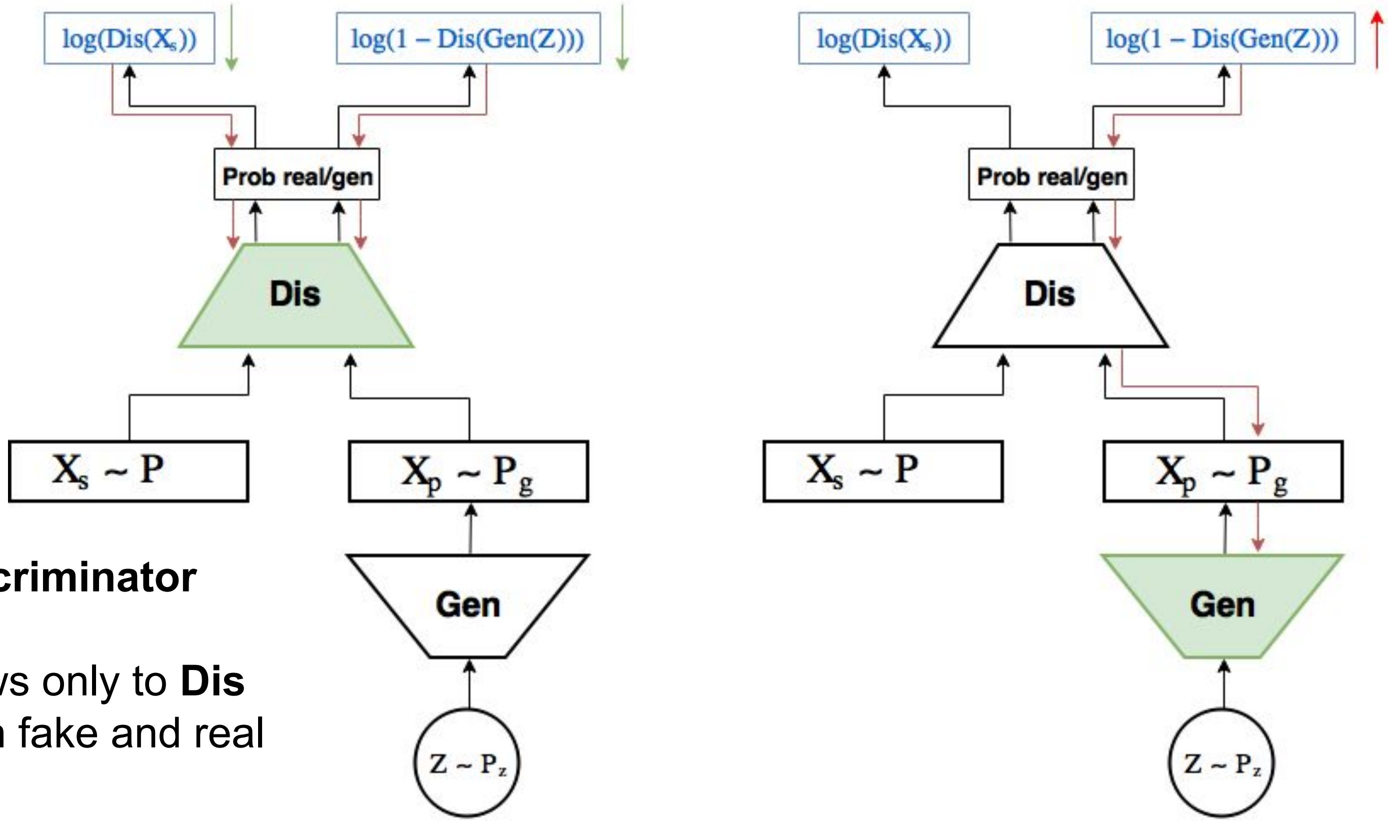



Image source: [Habr post on autoencoders and GANs]()

# Training GAN

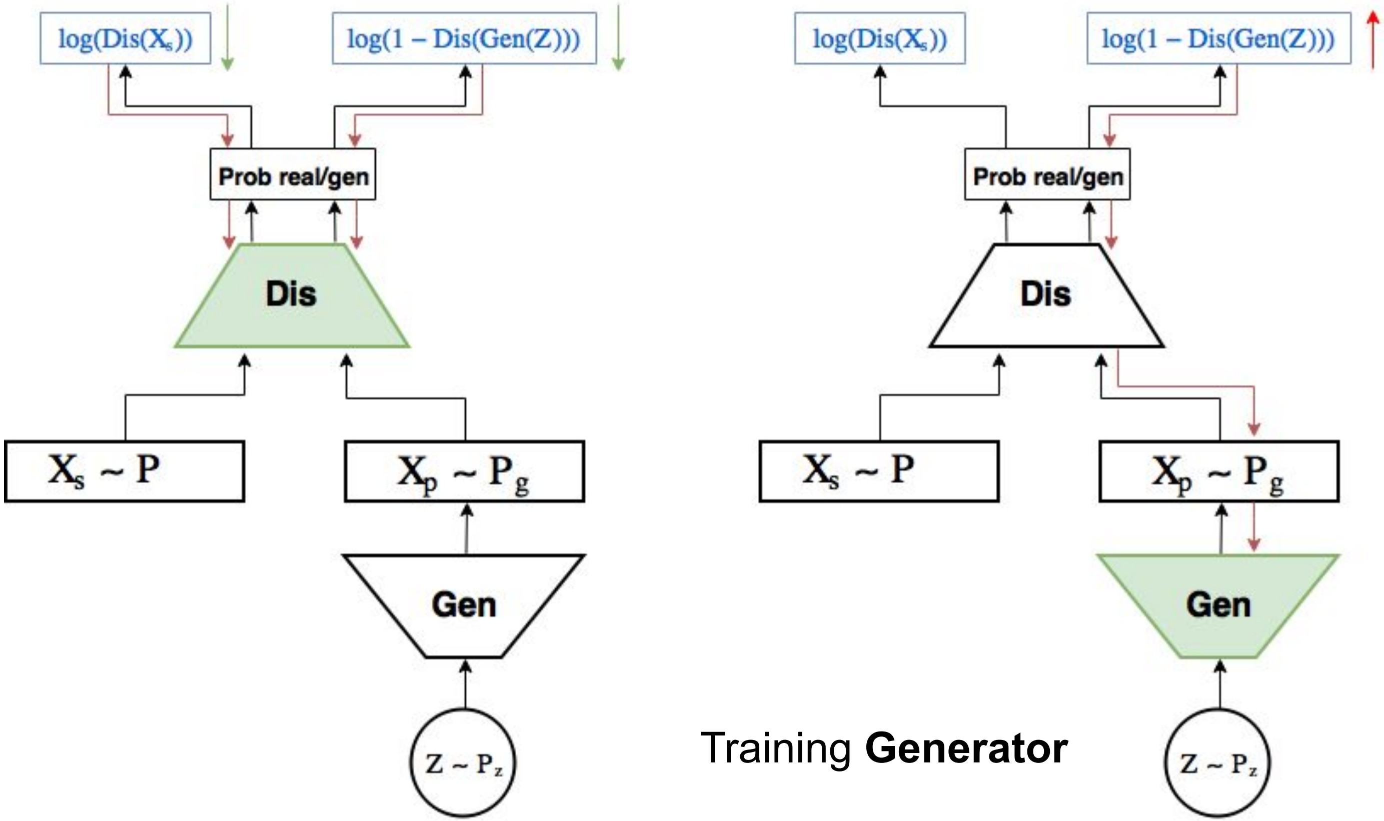



Training **Discriminator**

Gradient flows only to **Dis** to distinguish fake and real examples

Image source: [Habr post on autoencoders and GANs]

# Training GAN




Training **Generator**

Gradient flows to **Gen** with **Dis** weights freezed to fool the Discriminator

Image source: Habr post on autoencoders and GANs

# Optimization process in GAN

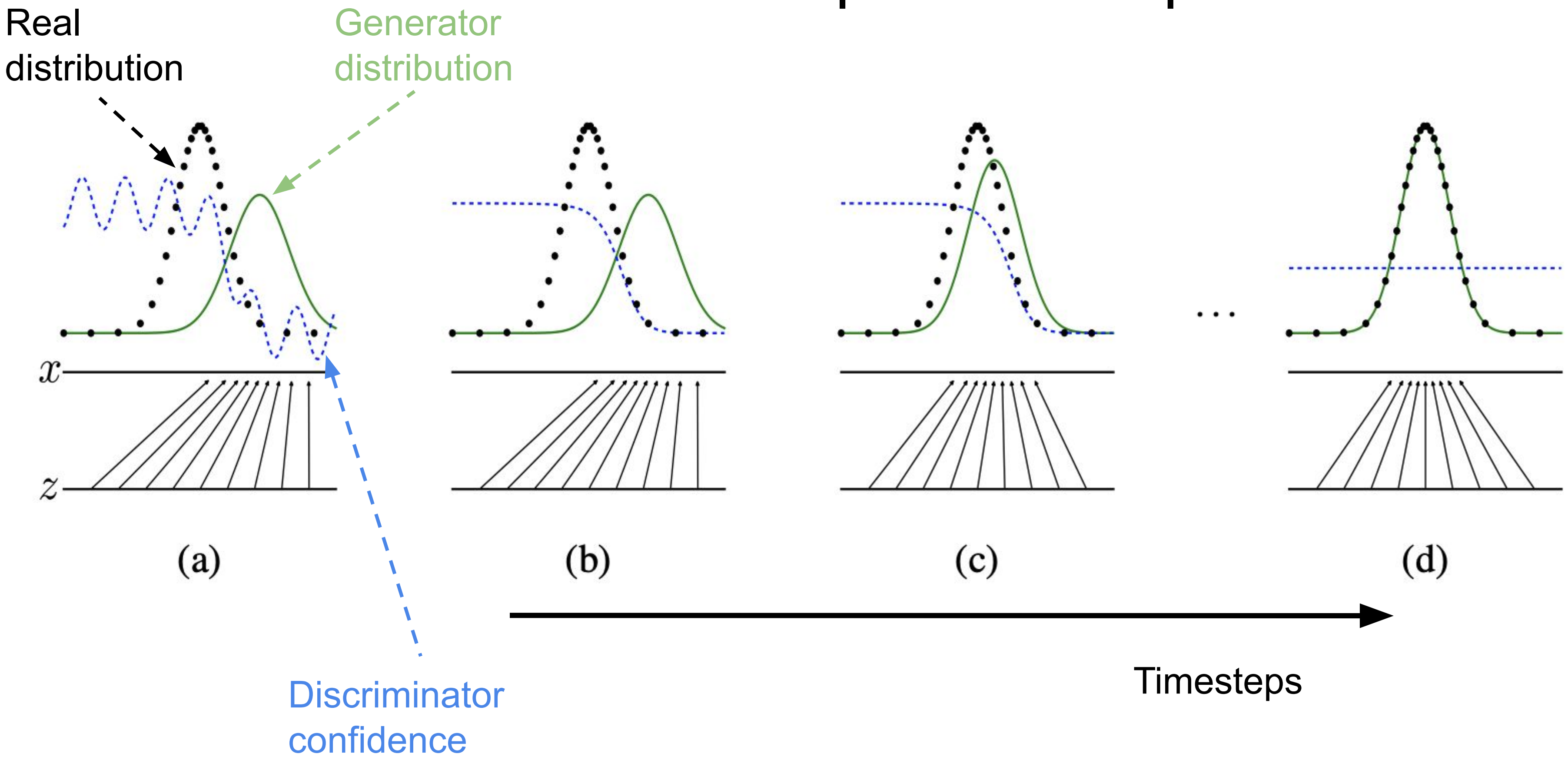



Image source: [Generative Adversarial Nets, Ian J. Goodfellow et al, 2014]

# GAN manifold

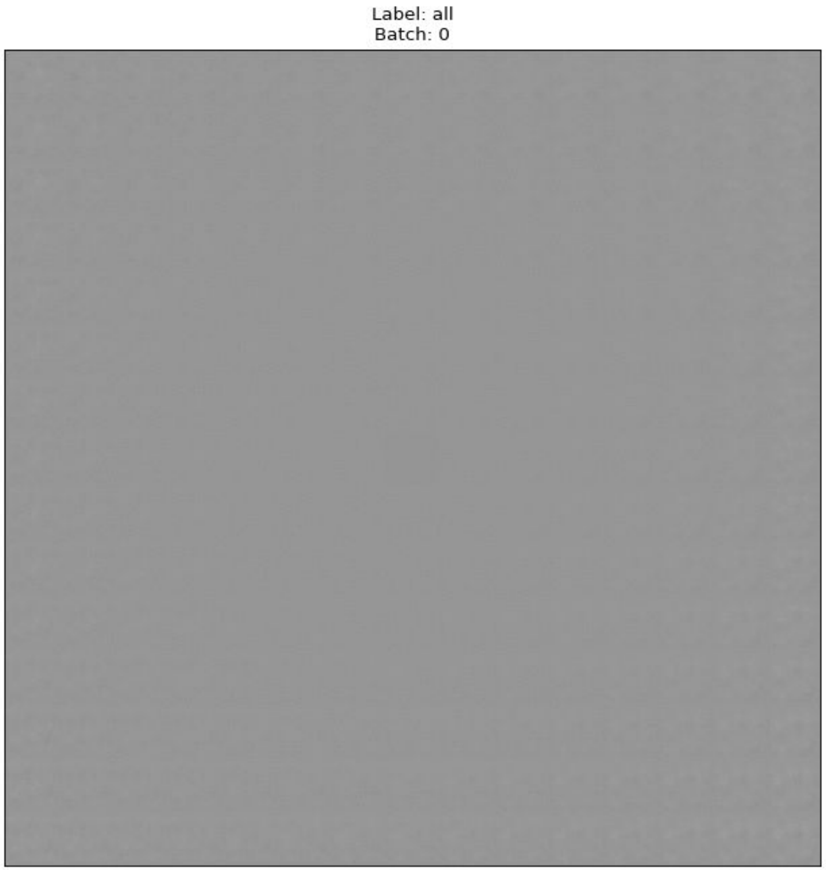



Image source: [Habr post on autoencoders and GANs]

# Conditional GAN

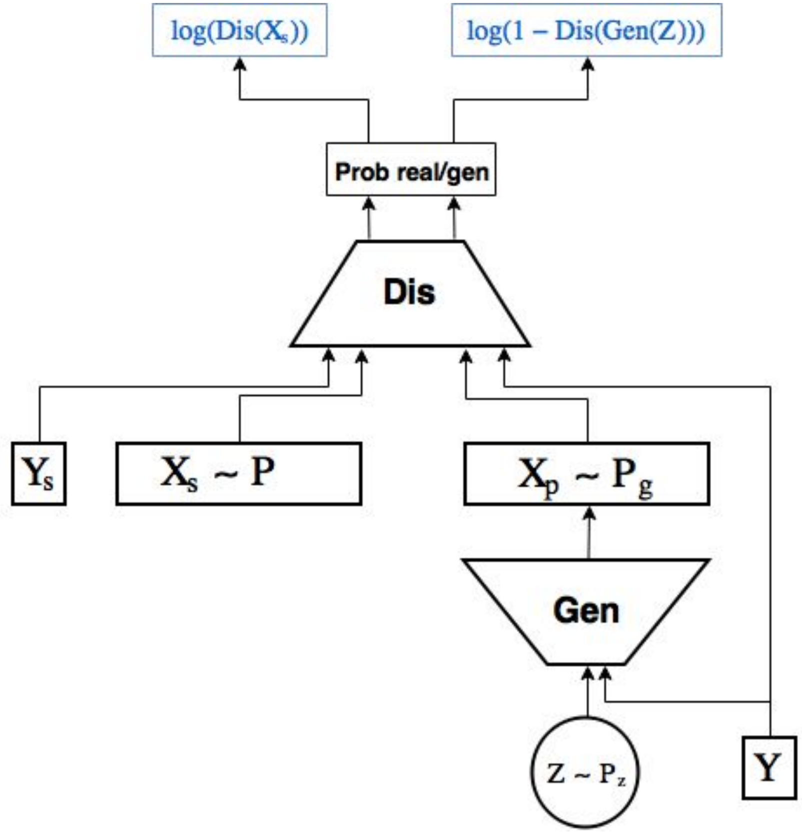
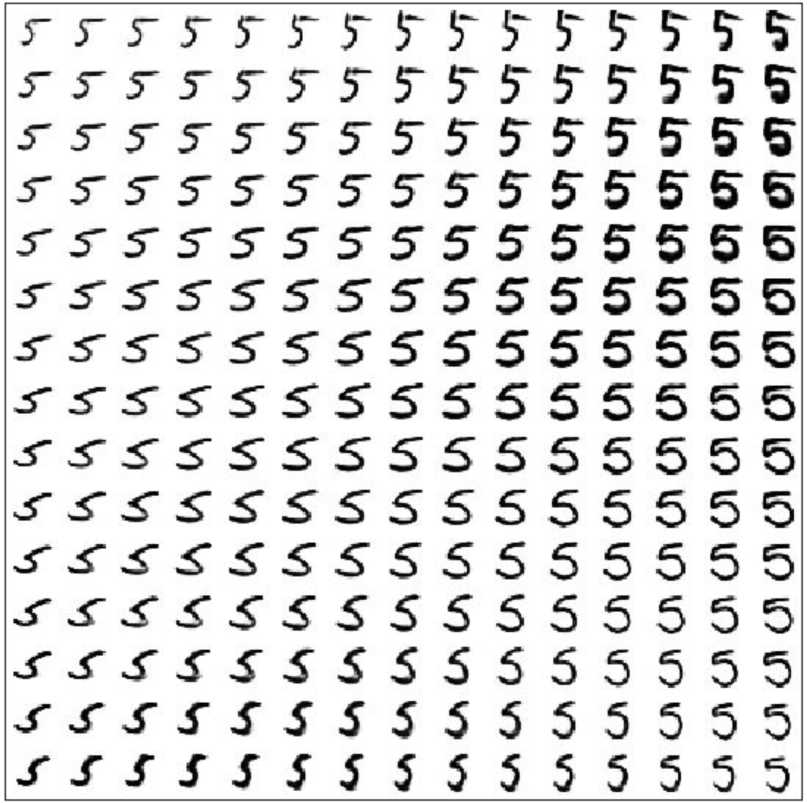



Image source: [Habr post on autoencoders and GANs]

# cGAN manifolds



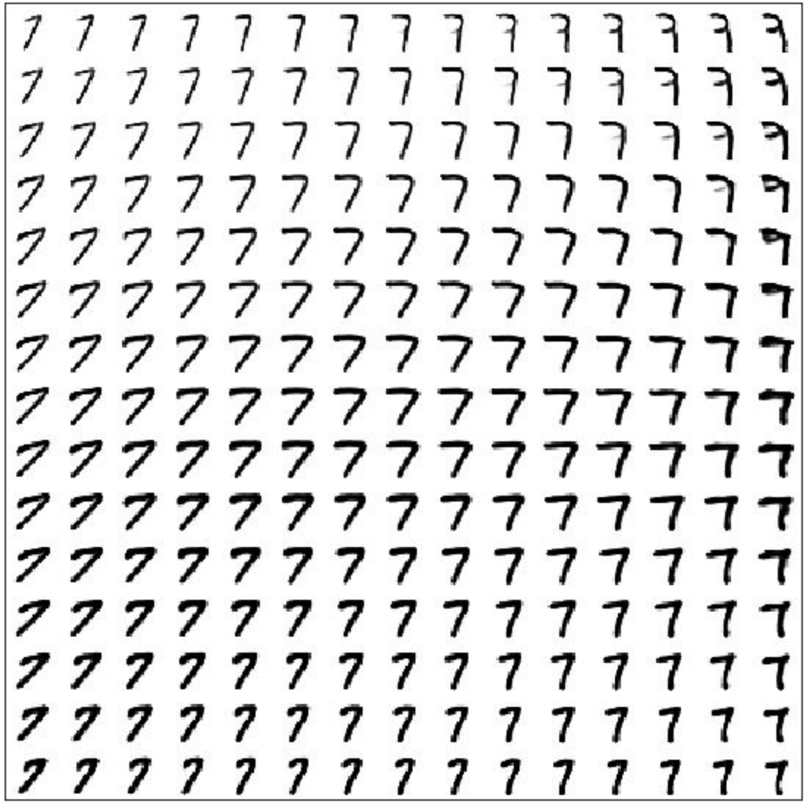


Image source: [Habr post on autoencoders and GANs]

# Some more combinations


Image source: Guiness World Records blog, original video PIKOTARO - PPAP (Pen Pineapple Apple Pen)

# Simple GAN

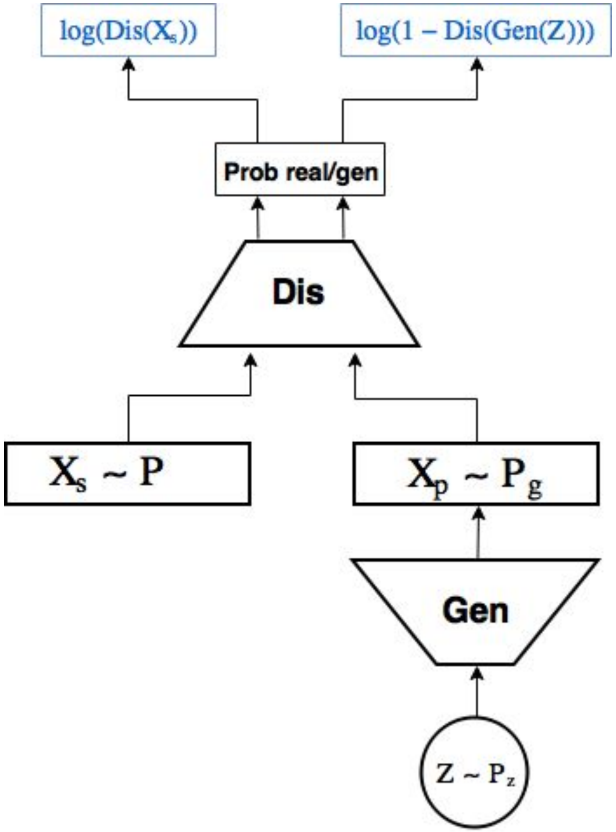
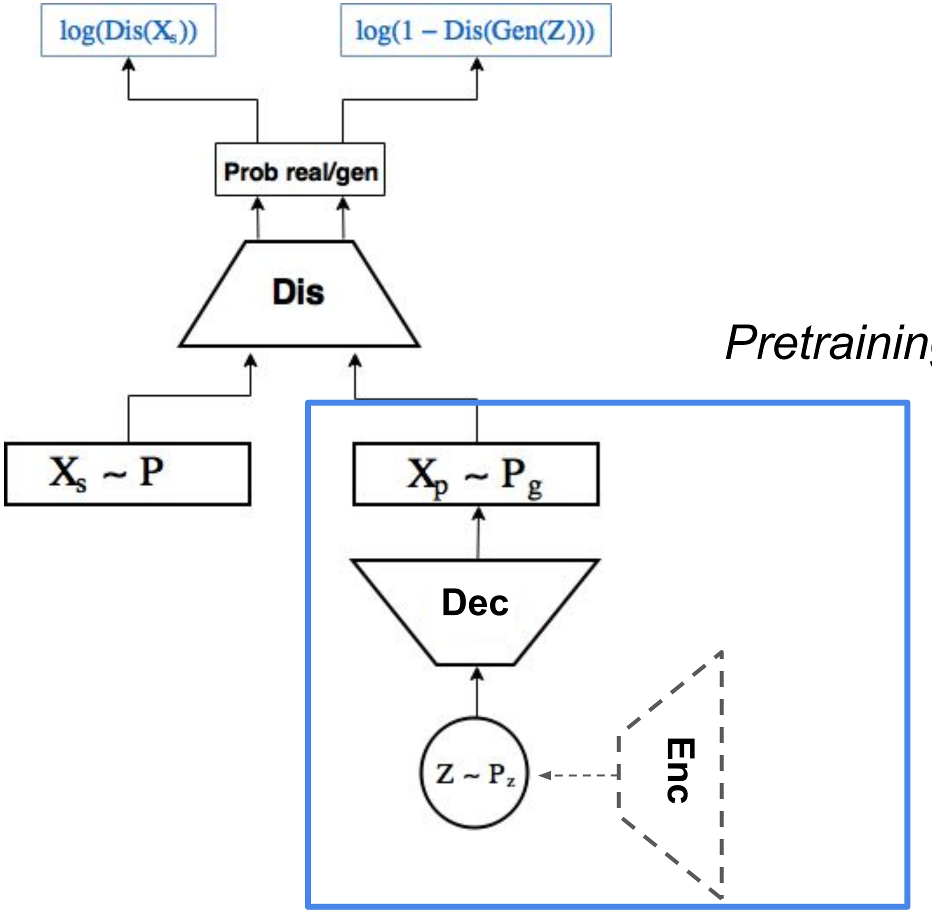



Image source: [Habr post on autoencoders and GANs]()

# VAE/GAN




Partial image source: [Habr post on autoencoders and GANs]

# VAE/GAN original illustration

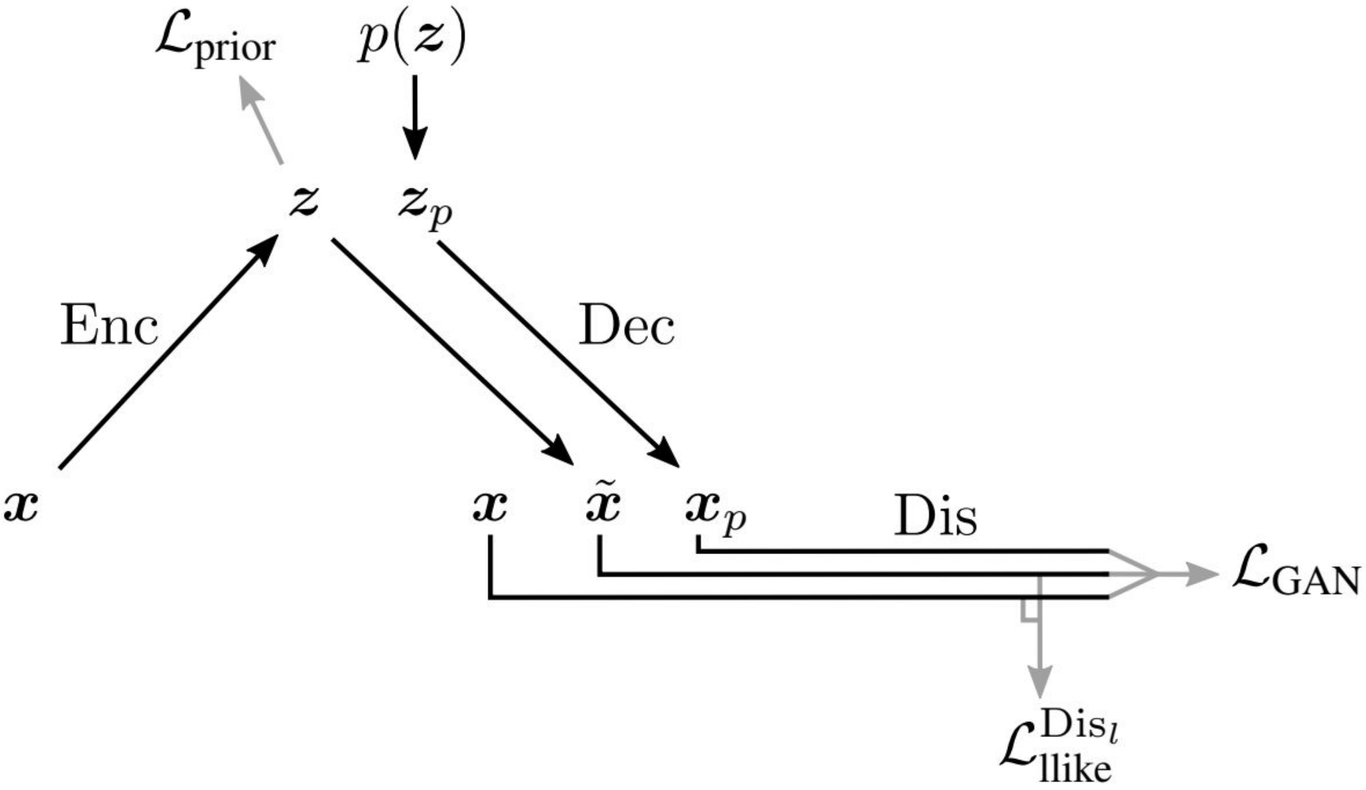



Image source: [Autoencoding beyond pixels using a learned similarity metric, Anders Boesen Lindbo Larsen et al, 2016]

# Q & A  and farewell

Time to write some code!