

# Lecture 15: Generative models

**Radoslav Neychev**

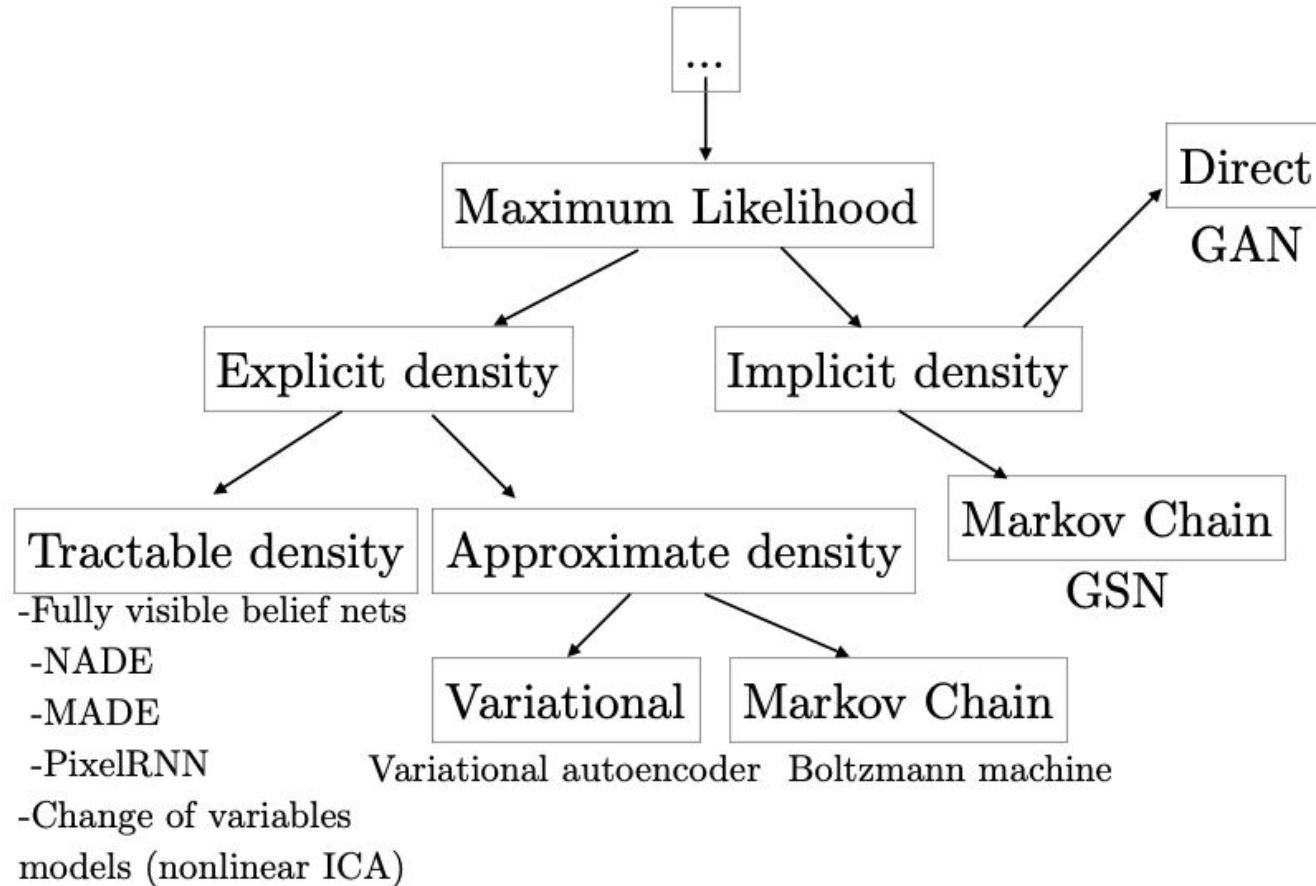
# References

This lecture structure was inspired a lot by the [series of Habr posts on Autoencoders and GANs in Keras](#)

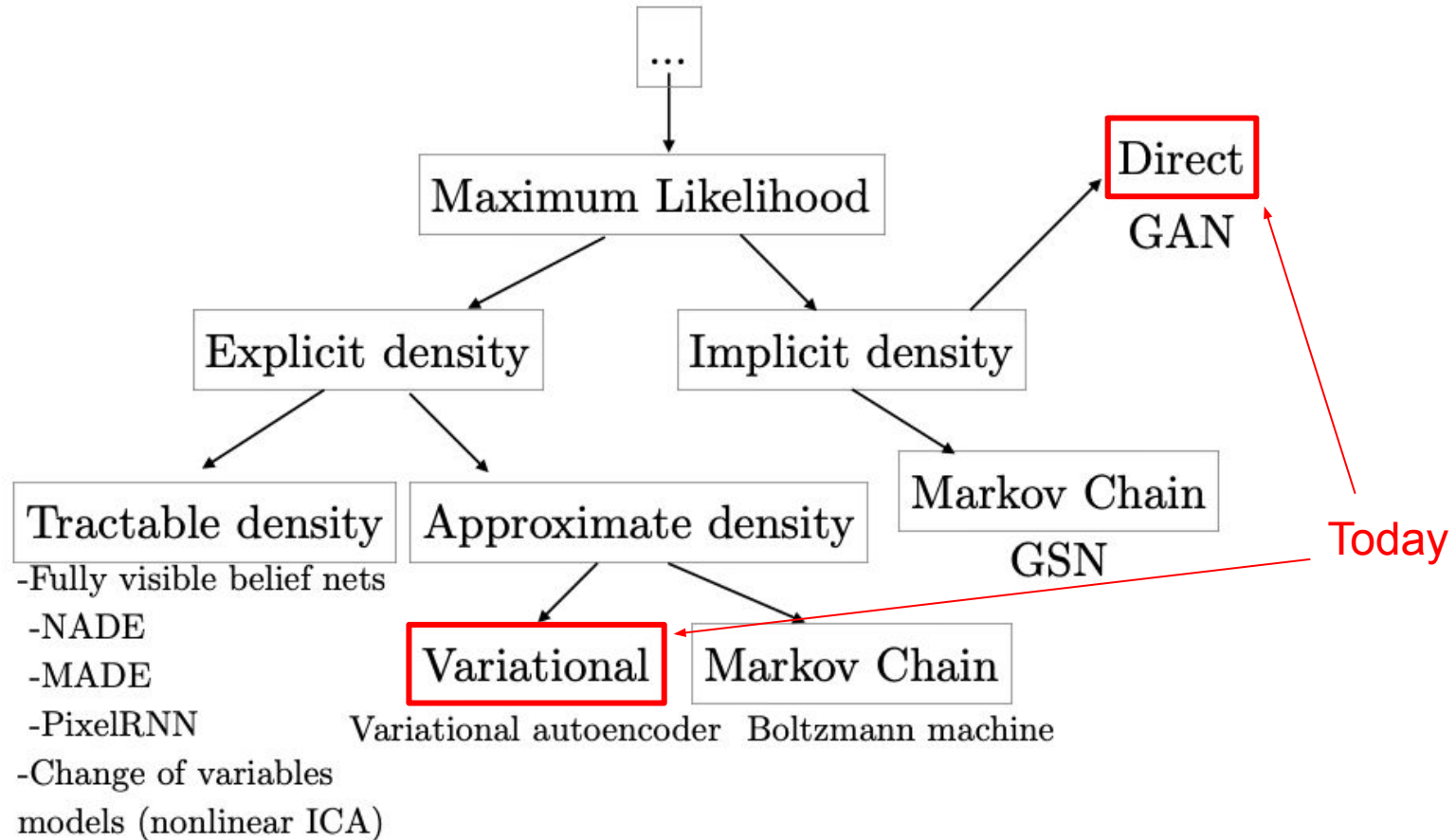
- Generative models overview
- Simple Autoencoders (not generative models!)
- Variational Autoencoders (VAE)
  - Conditional VAE
- Generative Adversarial Networks
  - Conditional GAN

And a lot of gif images.

# Generative models taxonomy



# Generative models taxonomy



# Autoencoders

Denote  $\mathbf{z}$  as encoded with encoder E input  $\mathbf{x}$

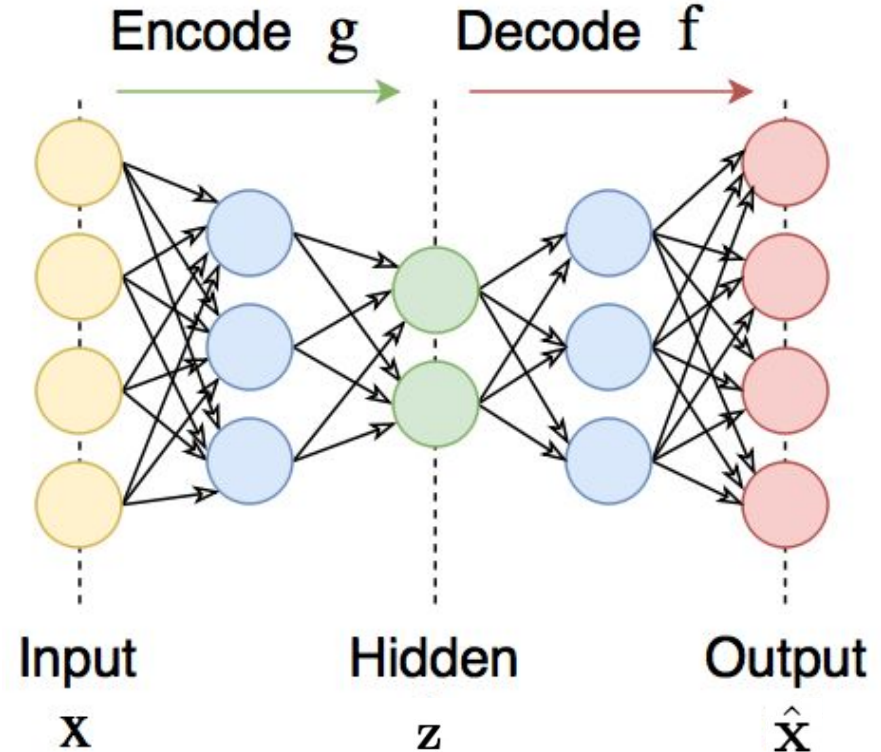
$$\mathbf{z} = E(\mathbf{x}, \boldsymbol{\theta}_E)$$

Decoder D recovers  $\mathbf{x}$  from latent representation

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

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

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



# Autoencoders

Denote  $\mathbf{z}$  as encoded with encoder E input  $\mathbf{x}$

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

Decoder D recovers  $\mathbf{x}$  from latent representation

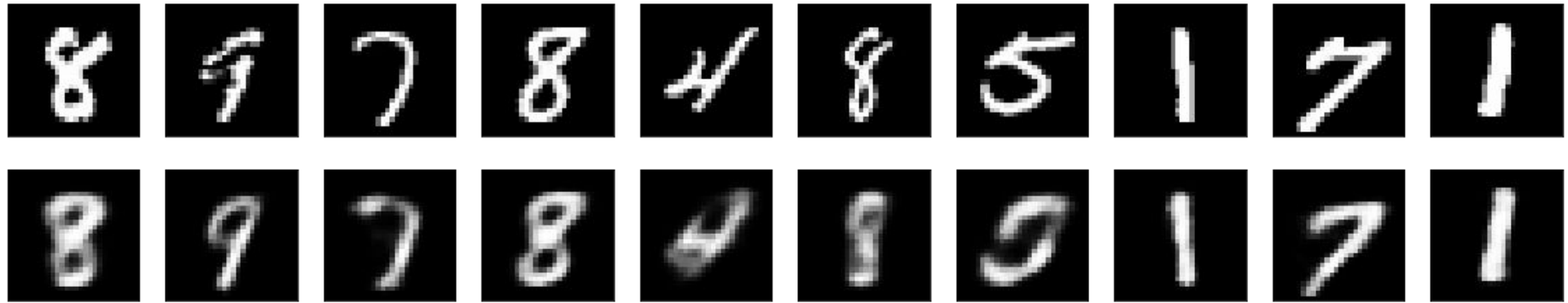
$$\hat{\mathbf{x}} = D(\mathbf{z}, \boldsymbol{\theta}_D)$$

Simple example: PCA

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

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

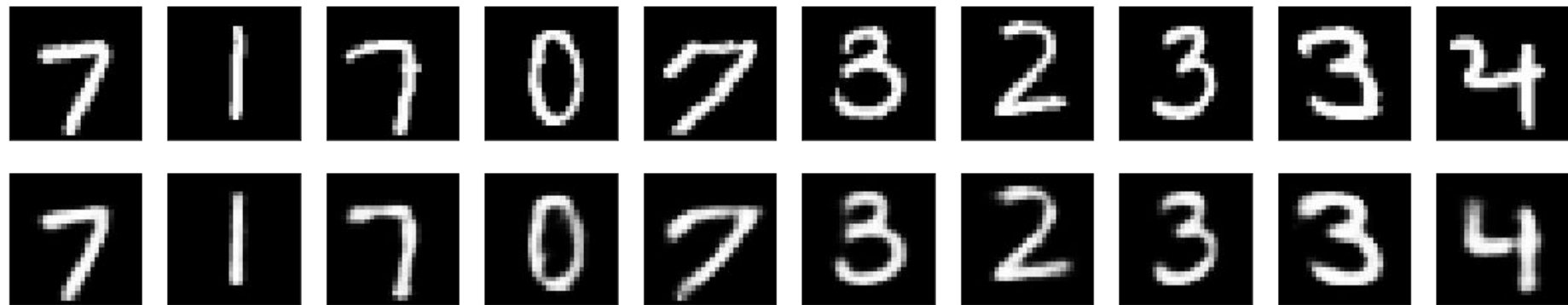
# PCA performance on MNIST



16 components



# Convolutional performance on MNIST

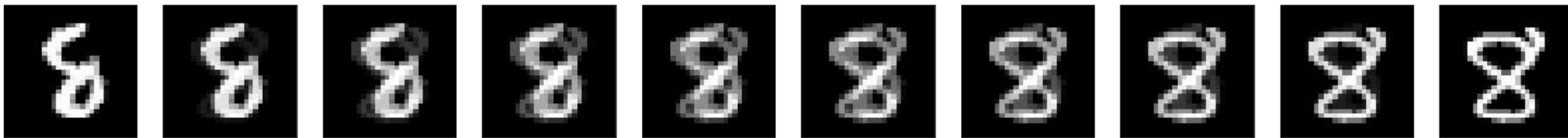


7 x 7 latent space

# Homotopy between samples

10 steps between samples

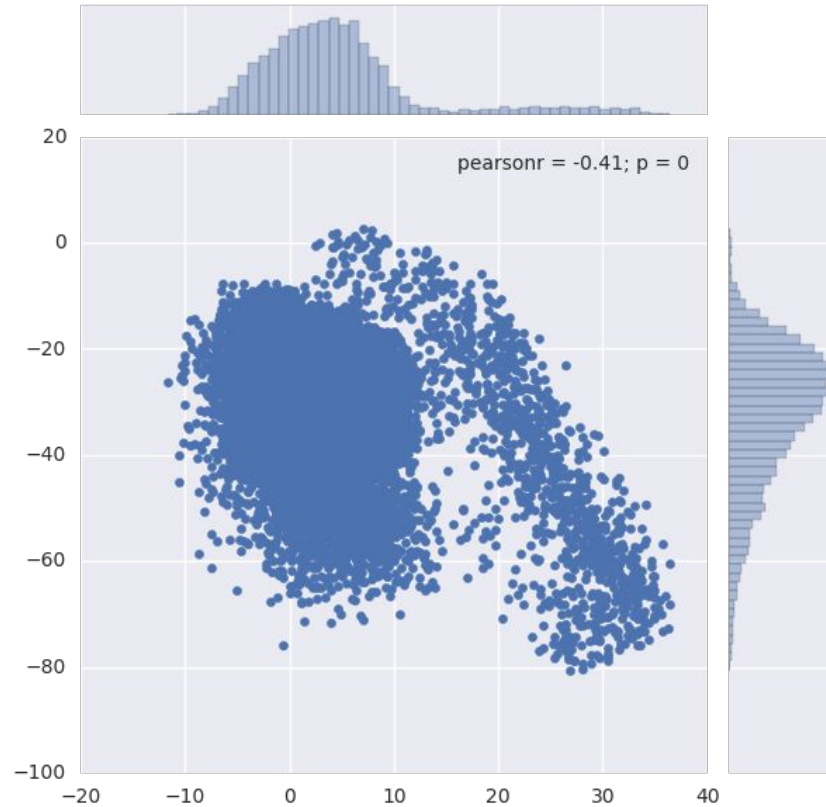
- In original feature space (28 x 28):



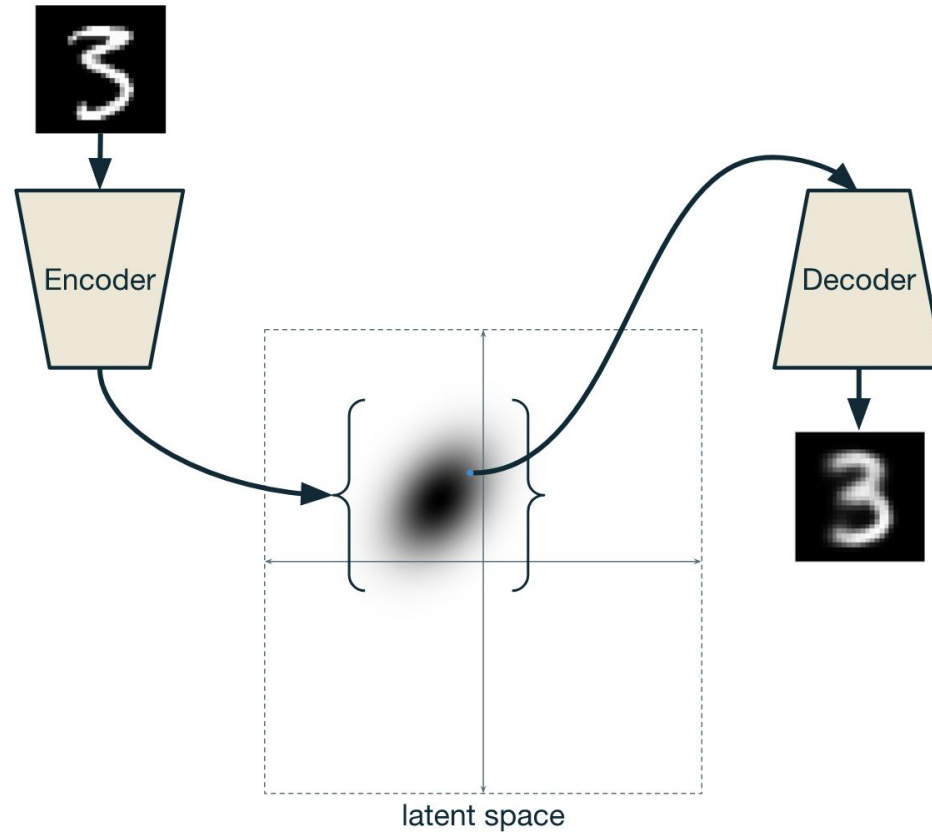
- In latent space (7 x 7):



# Latent space structure



# VAE intuition



# KL divergence

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

Kullback–Leibler divergence is defined as

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

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

Applying the Bayes rule:

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

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

Applying the Bayes rule:

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

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

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

Applying the Bayes rule:

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

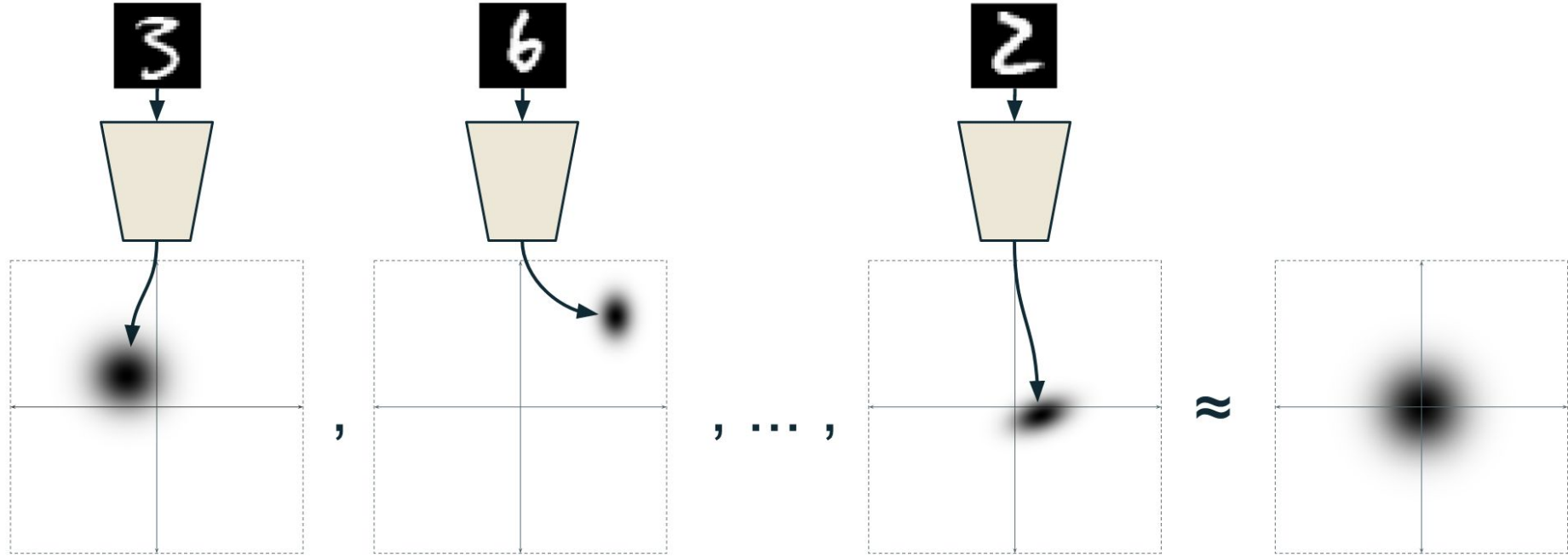
$$\log P(X) - \mathcal{D} [Q(z) \| P(z|X)] = E_{z \sim Q} [\log P(X|z)] - \mathcal{D} [Q(z) \| P(z)]$$

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

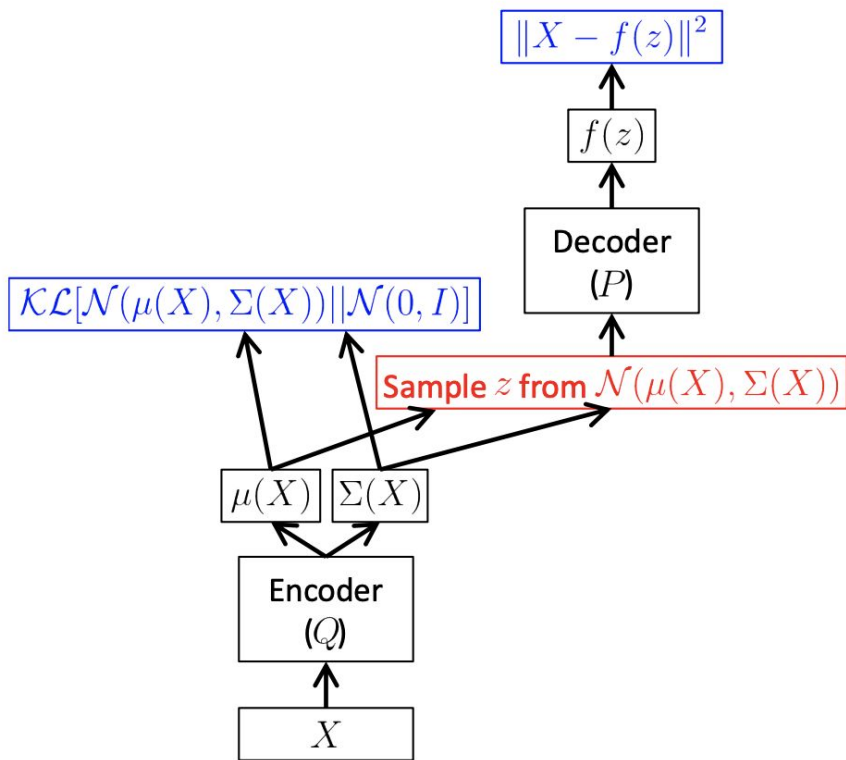
*This equation is the core of Variational Autoencoders*



# Structure of the latent space



# VAE so far



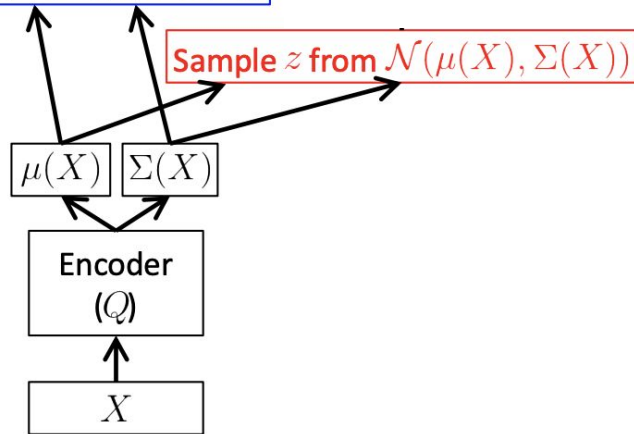
# VAE so far

$$\mathcal{D}[\mathcal{N}(\mu(X), \Sigma(X)) \| \mathcal{N}(0, I)] =$$

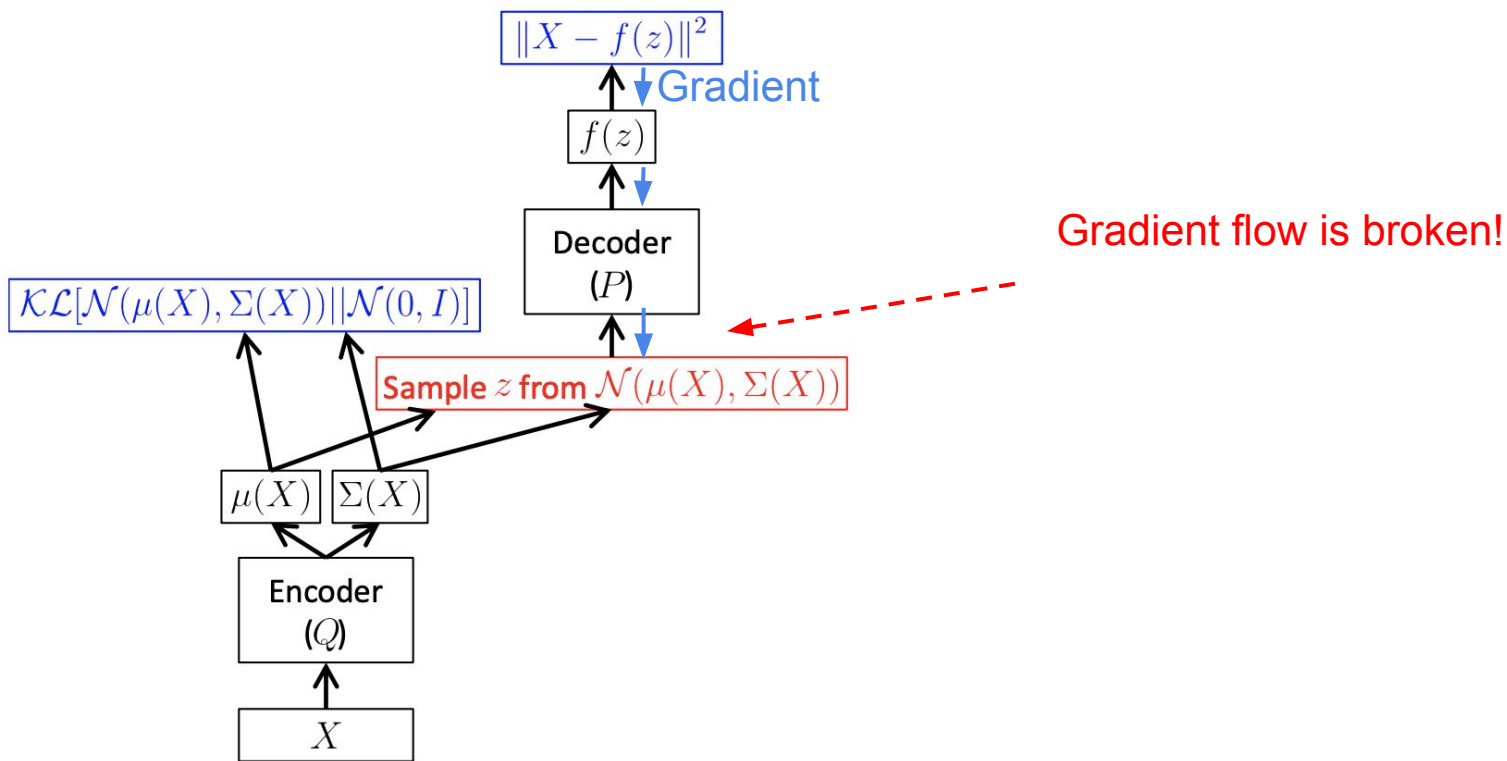
$$\frac{1}{2} \left( \text{tr}(\Sigma(X)) + (\mu(X))^{\top} (\mu(X)) - k - \log \det(\Sigma(X)) \right)$$

*Try to derive it by yourself*

$$\mathcal{KL}[\mathcal{N}(\mu(X), \Sigma(X)) \| \mathcal{N}(0, I)]$$

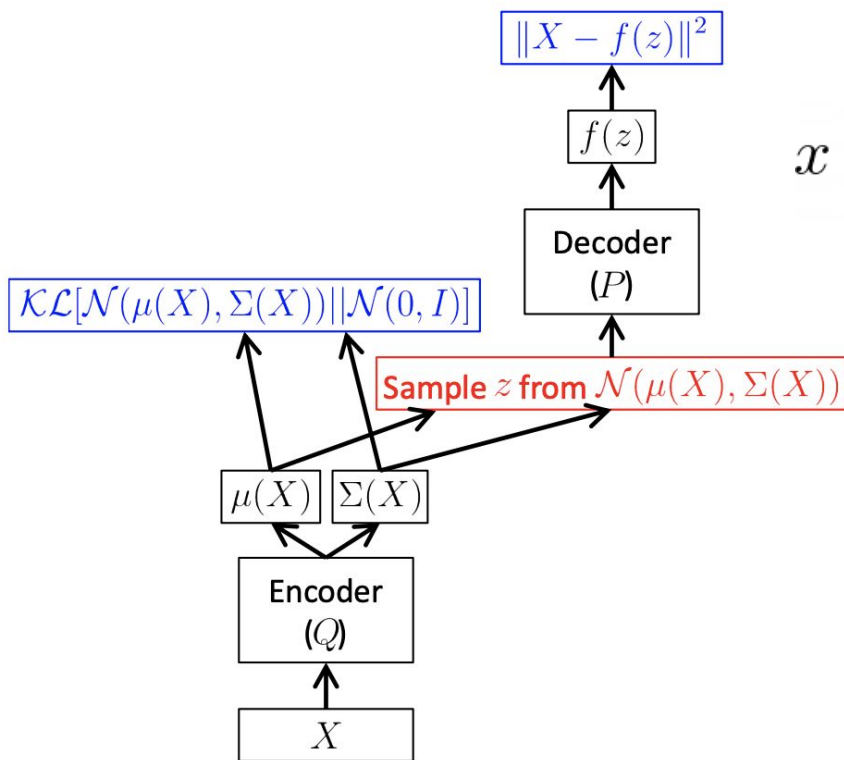


# VAE so far

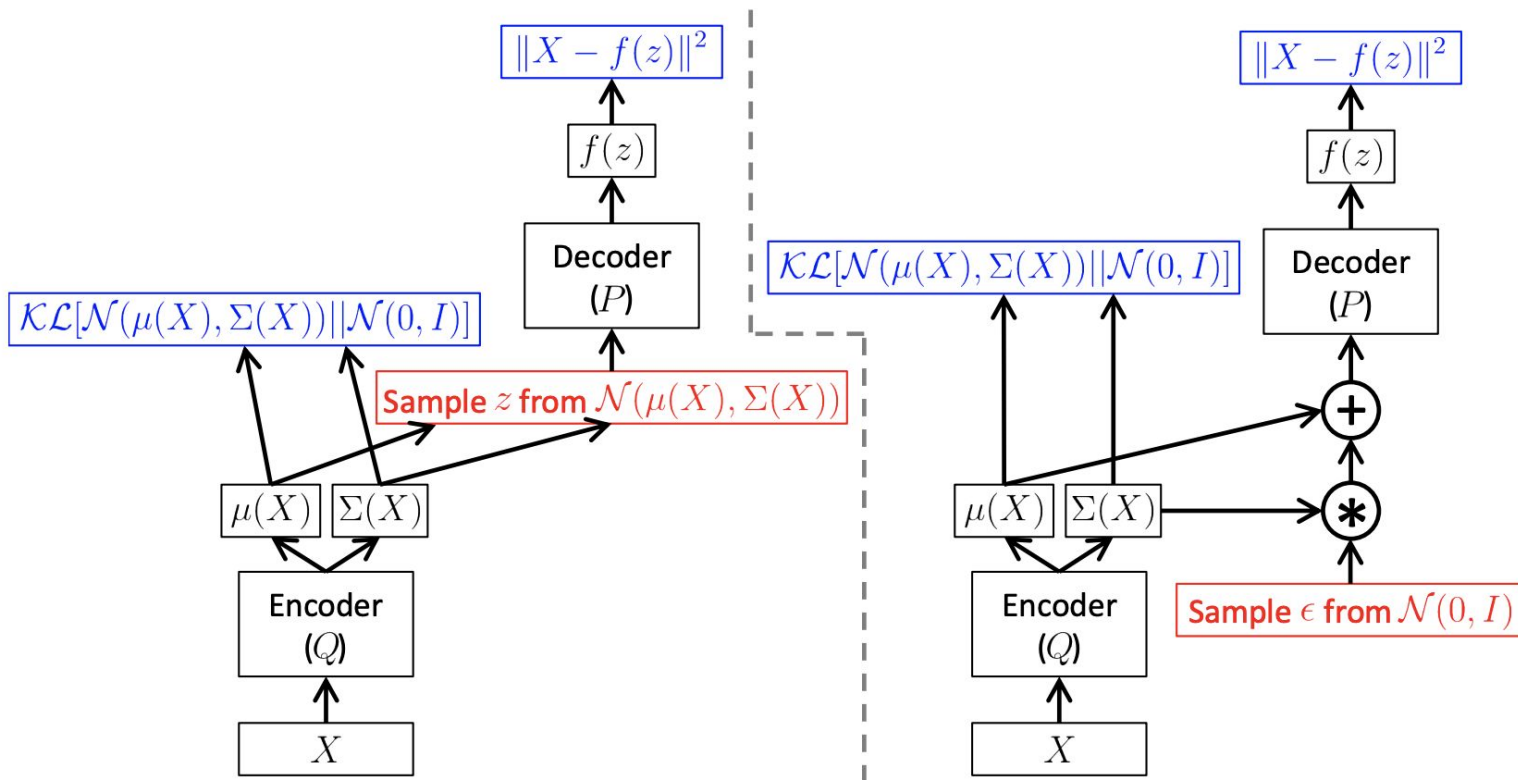


# Reparametrization trick

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

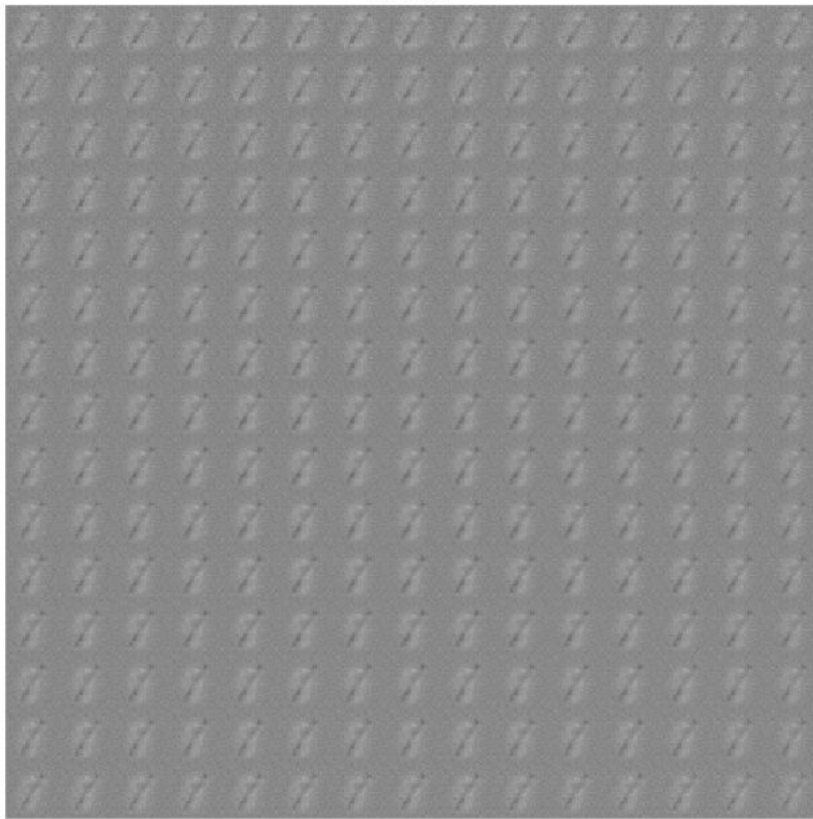


# Reparametrization trick

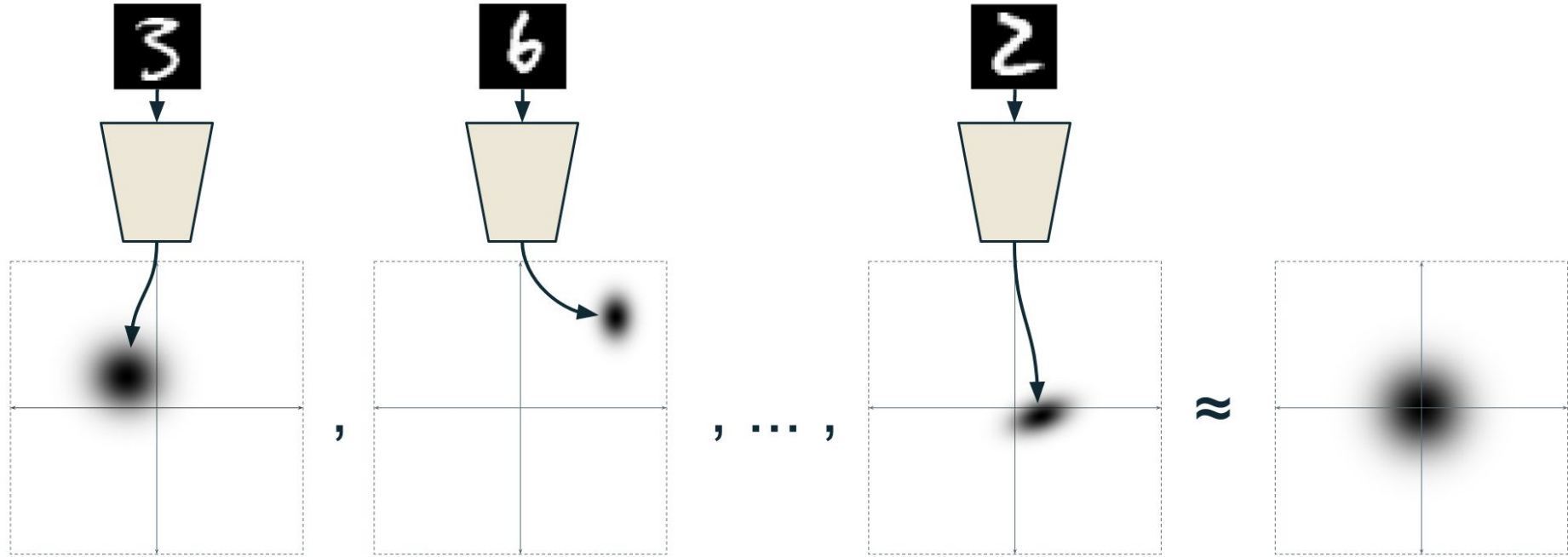


# VAE manifold

Epoch: 0

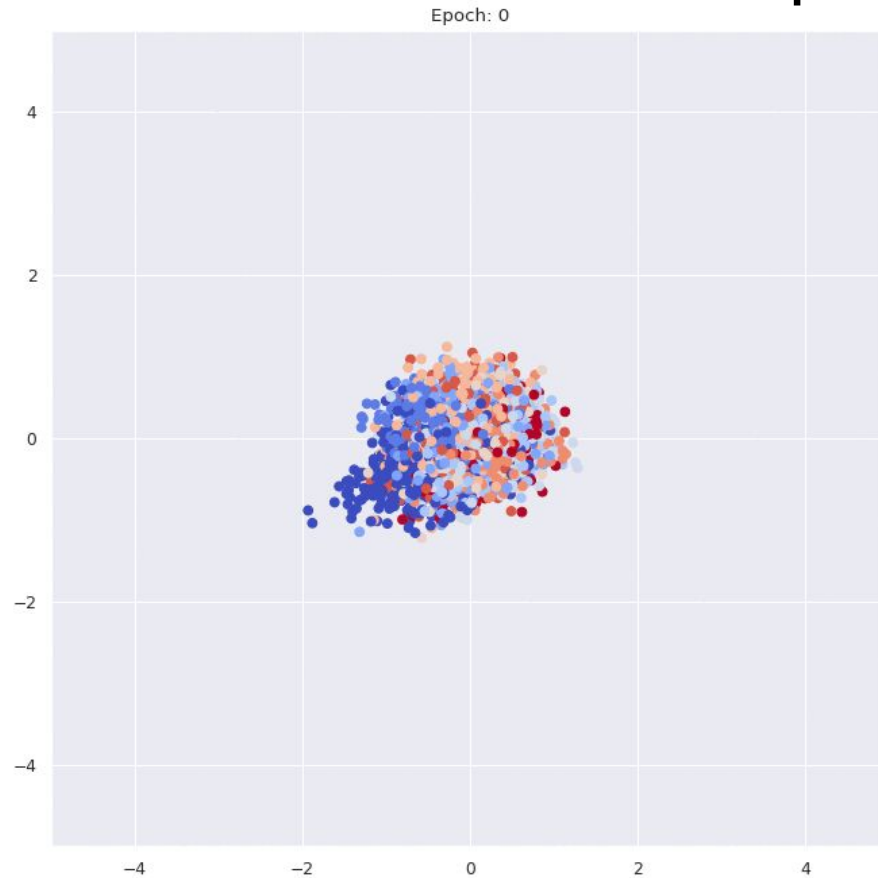


# Structure of the latent space

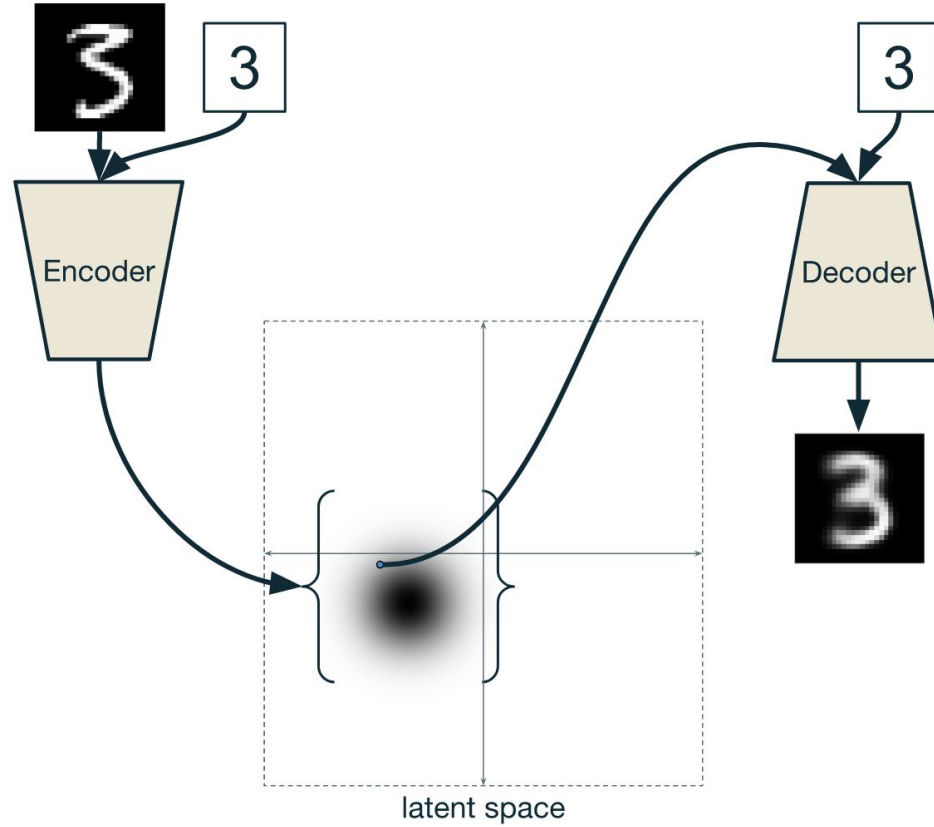




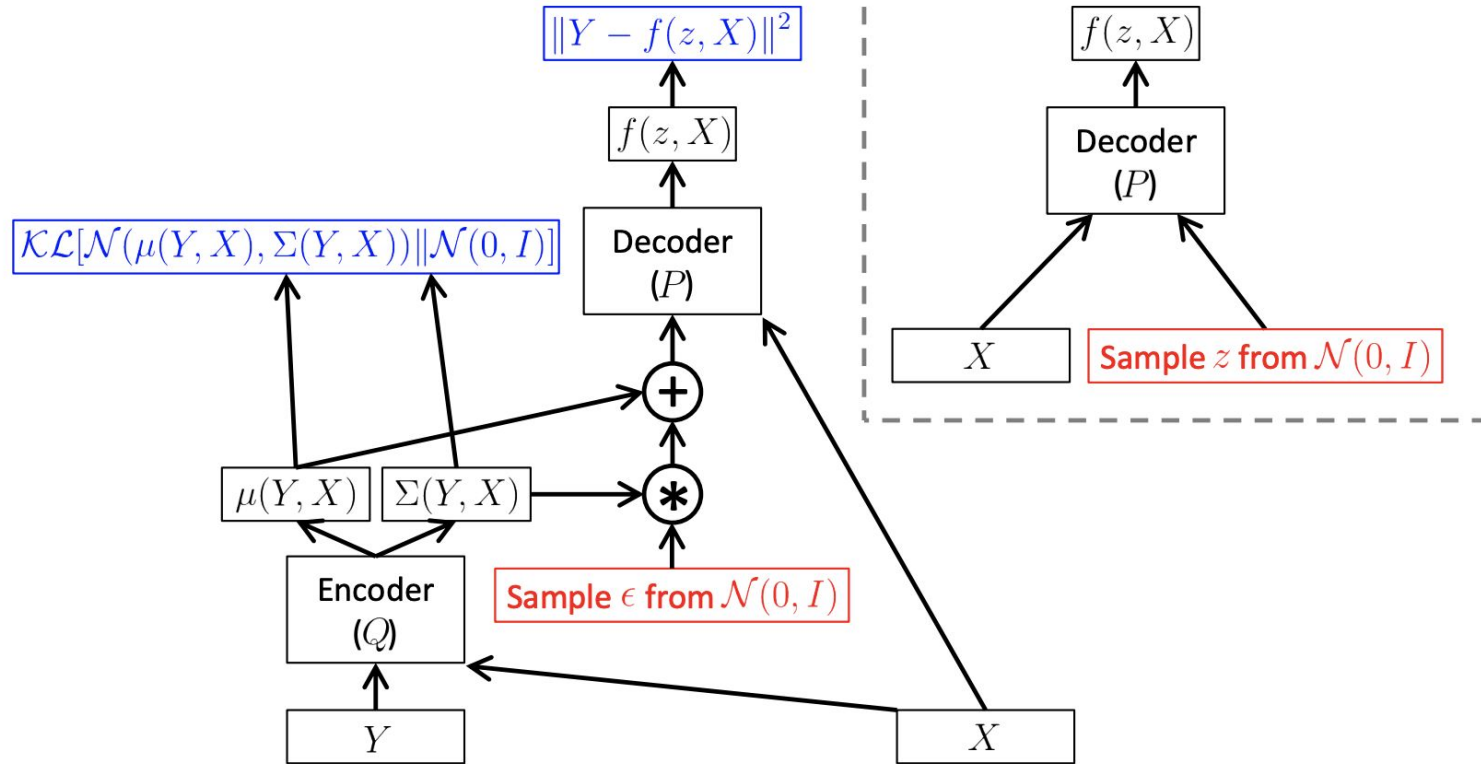
# VAE latent space distribution



# Conditional VAE intuition

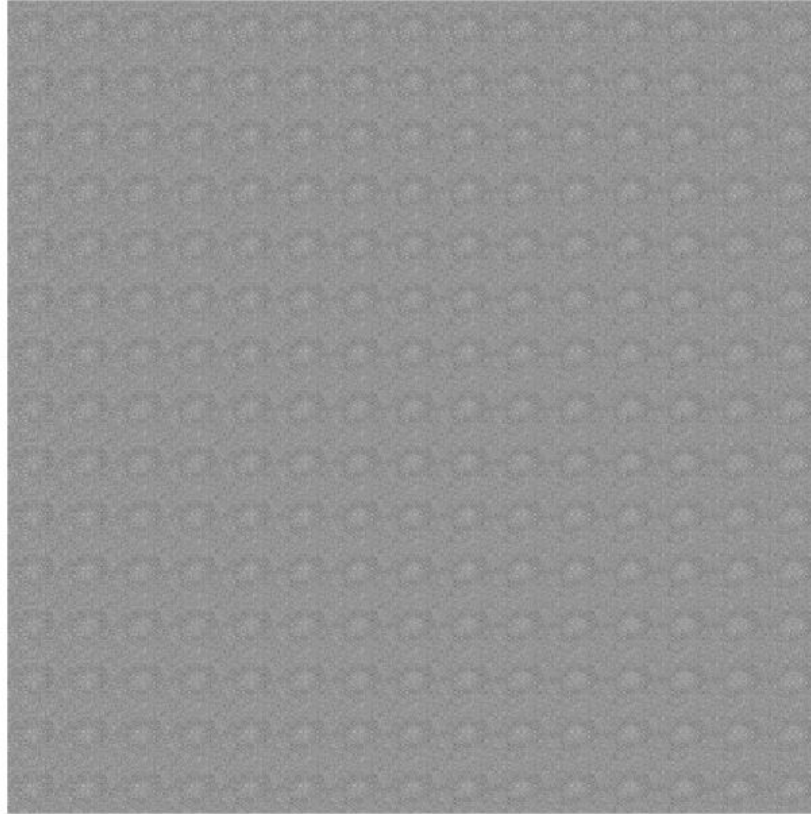


# Conditional VAE



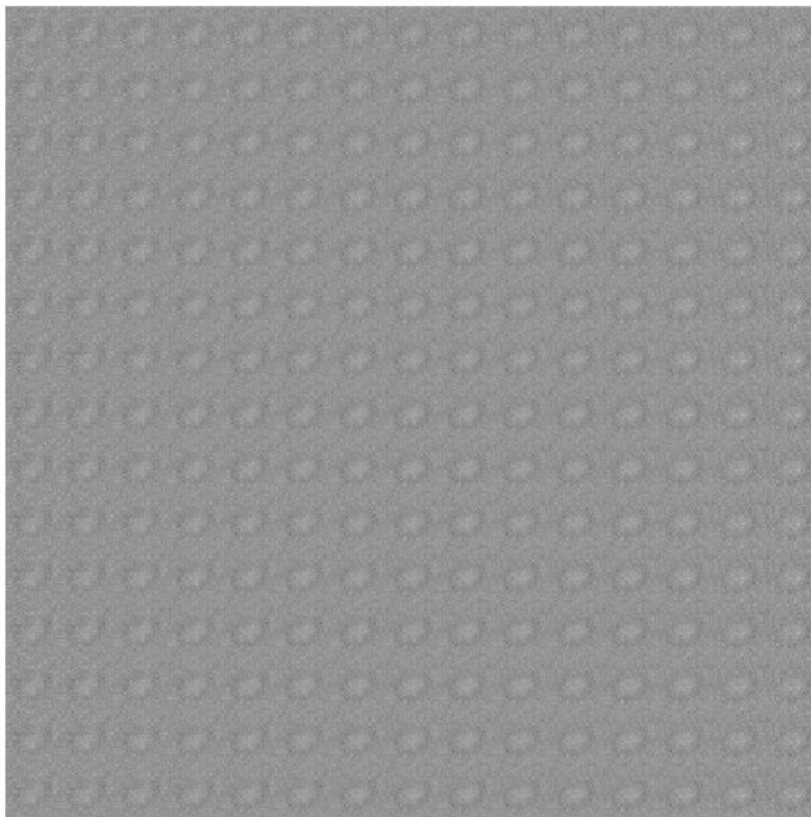
# cVAE manifold

Epoch: 0



# cVAE manifold

Epoch: 0

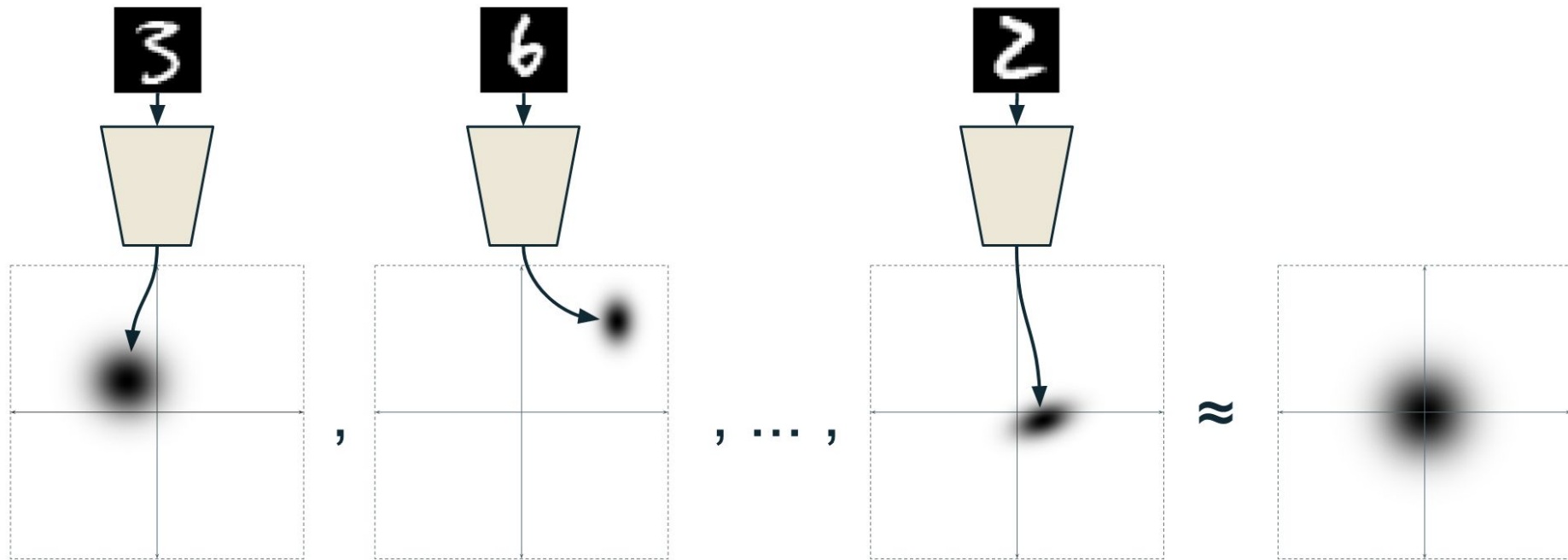


# Transferring style with cVAE

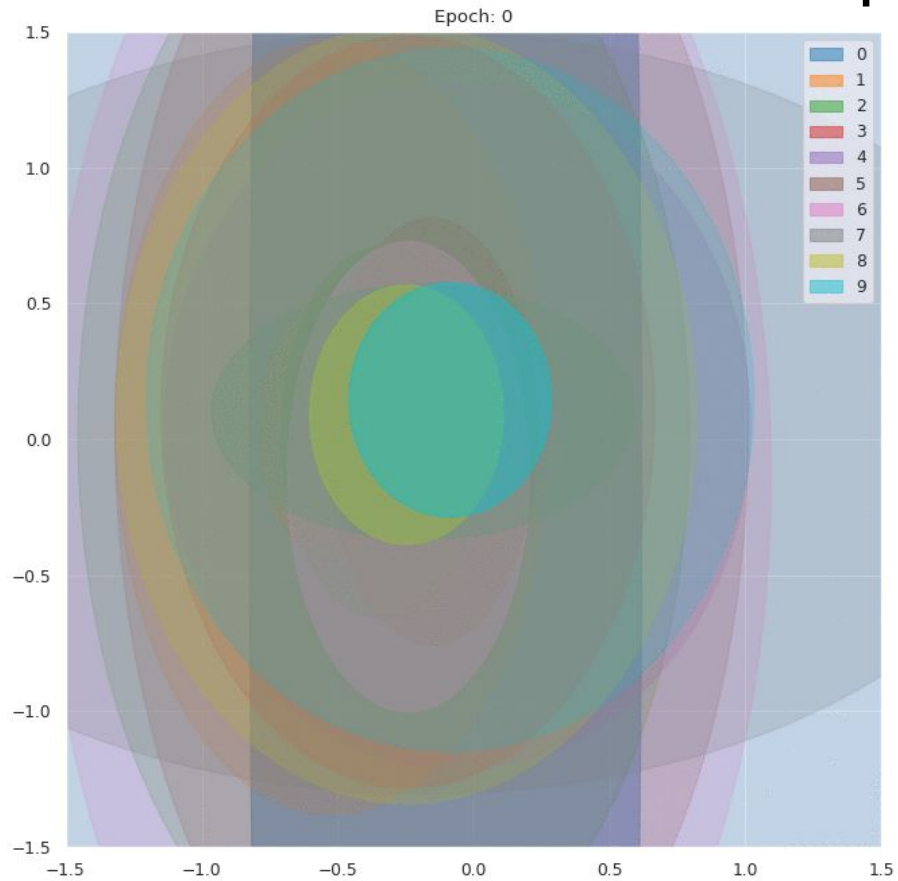


Seed label is 1

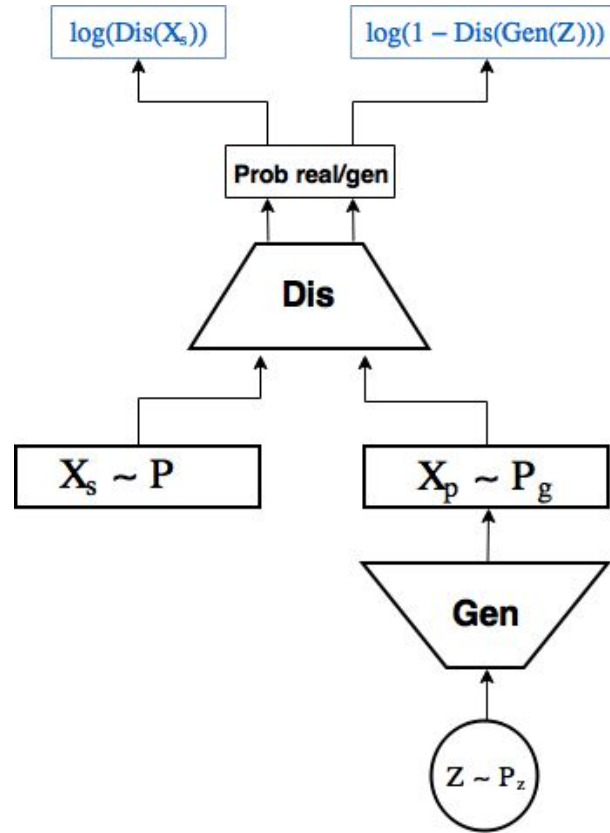
# Once again



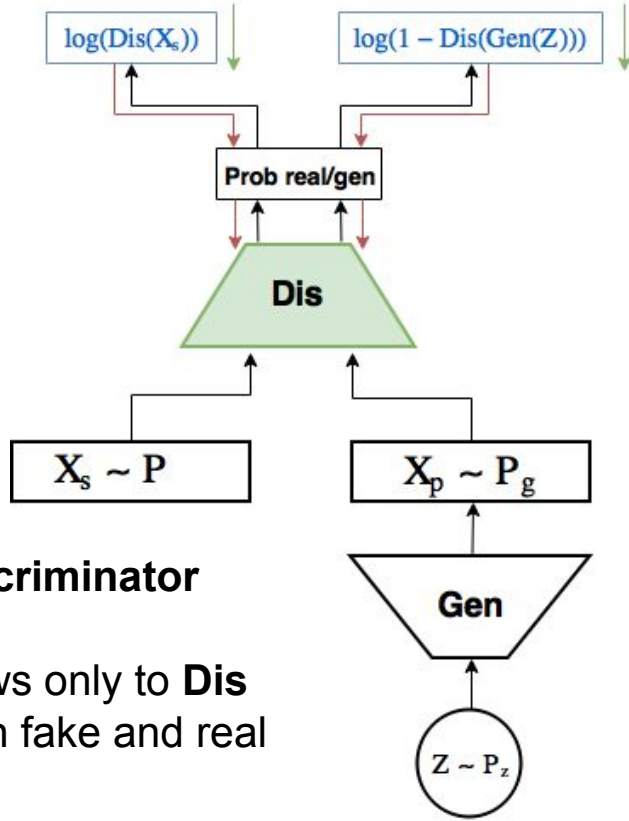
# cVAE latent space distribution





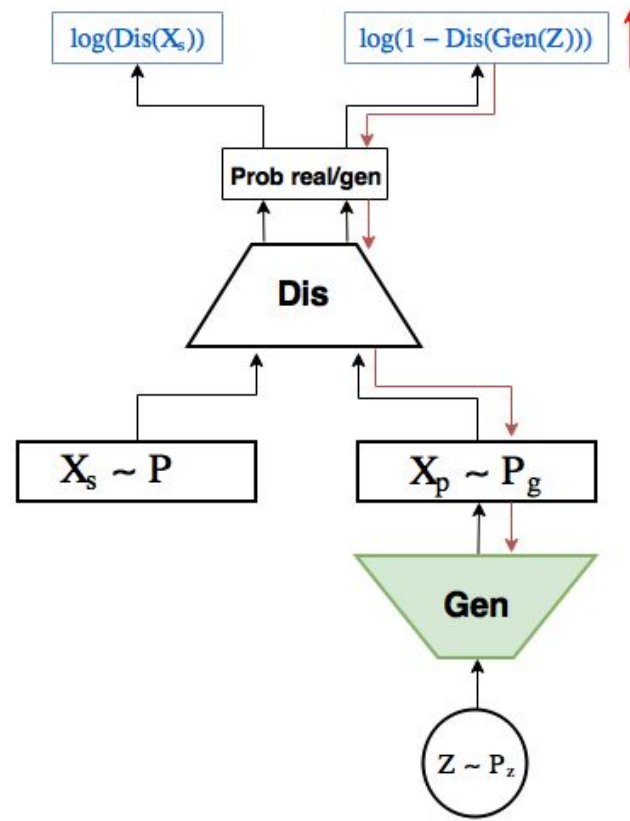


# Training GAN

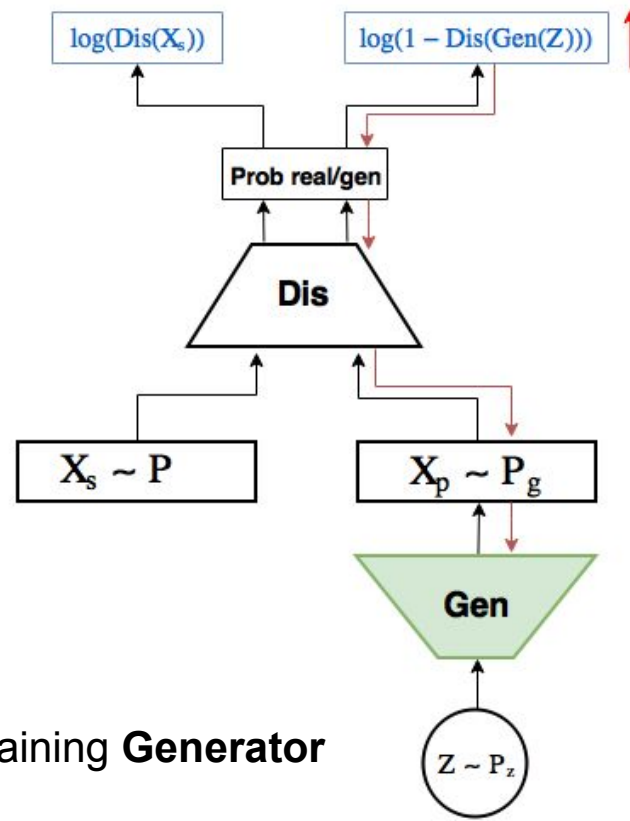
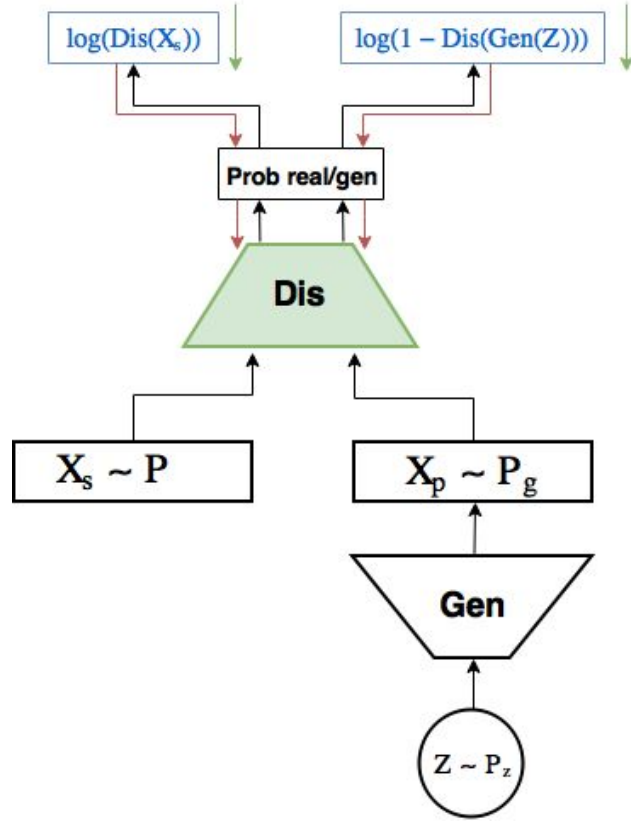


Training **Discriminator**

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



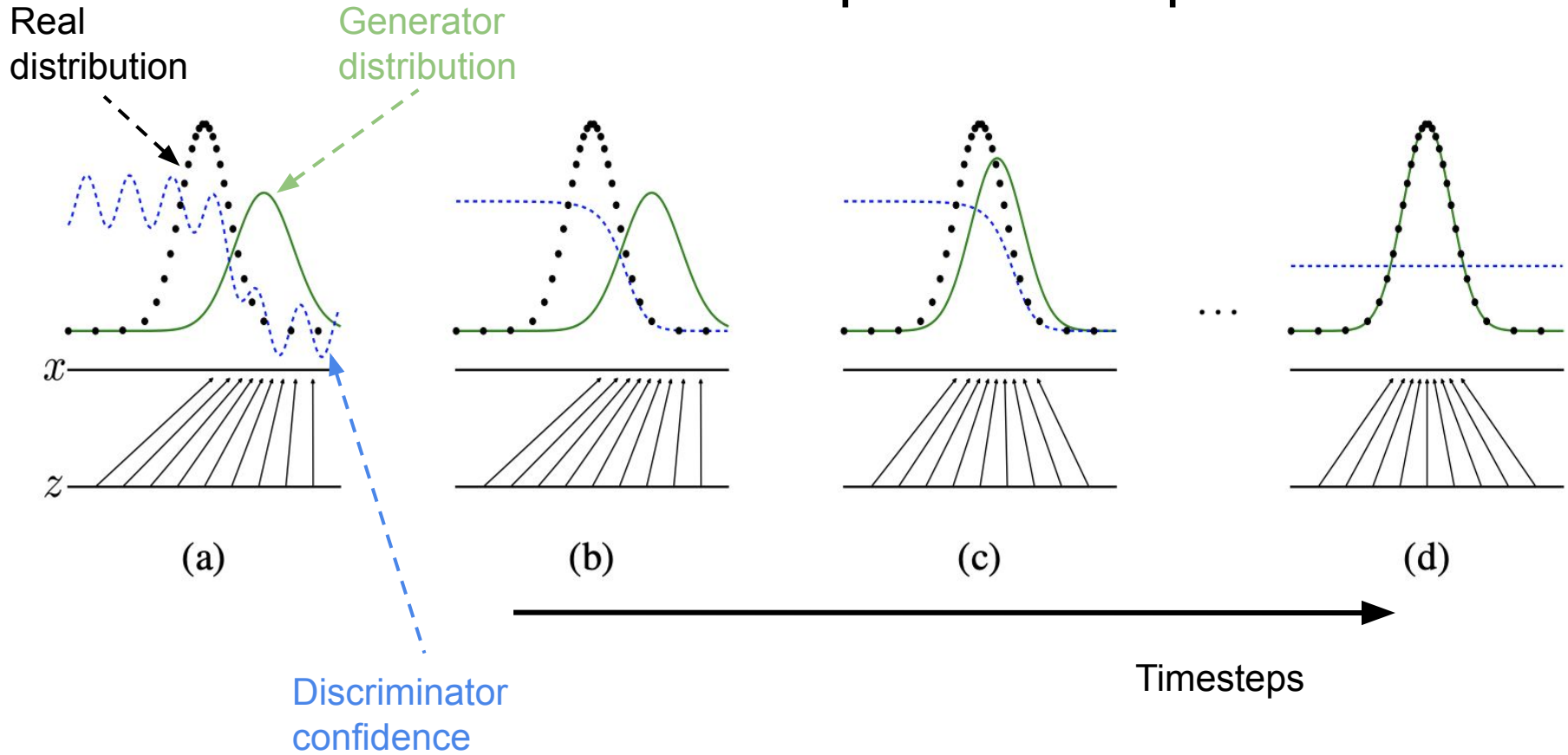
# Training GAN



Training **Generator**

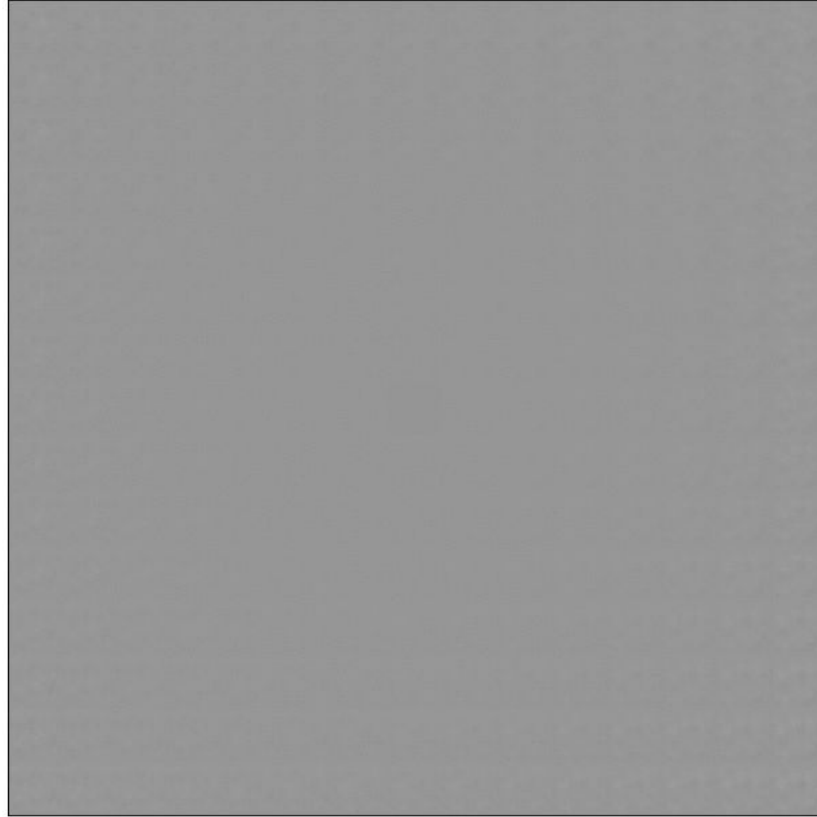
Gradient flows to **Gen** with **Dis** weights freezed to fool the Discriminator

# Optimization process in GAN

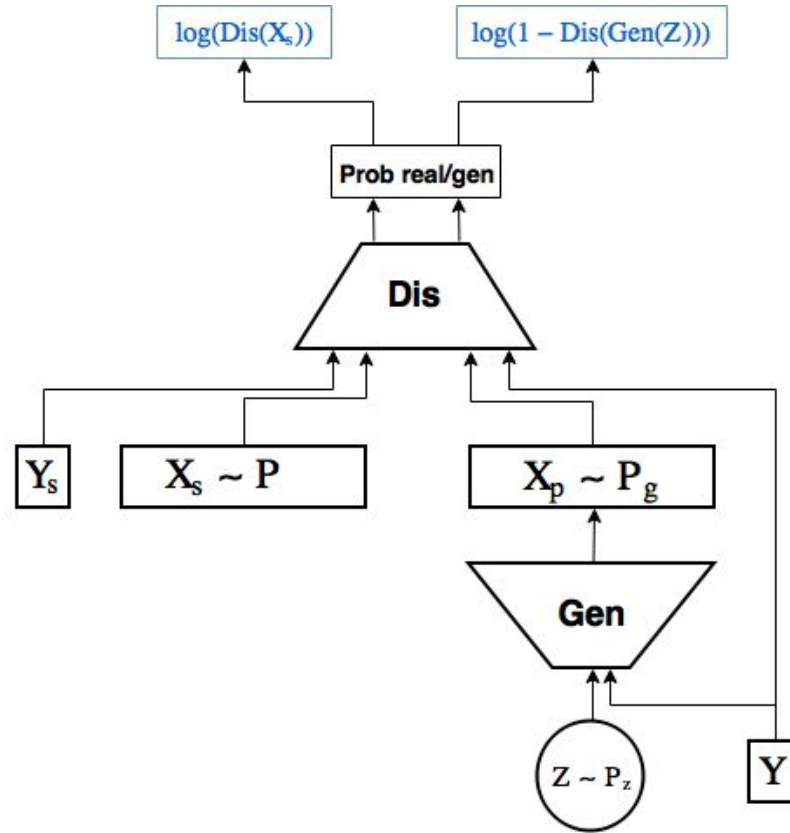


# GAN manifold

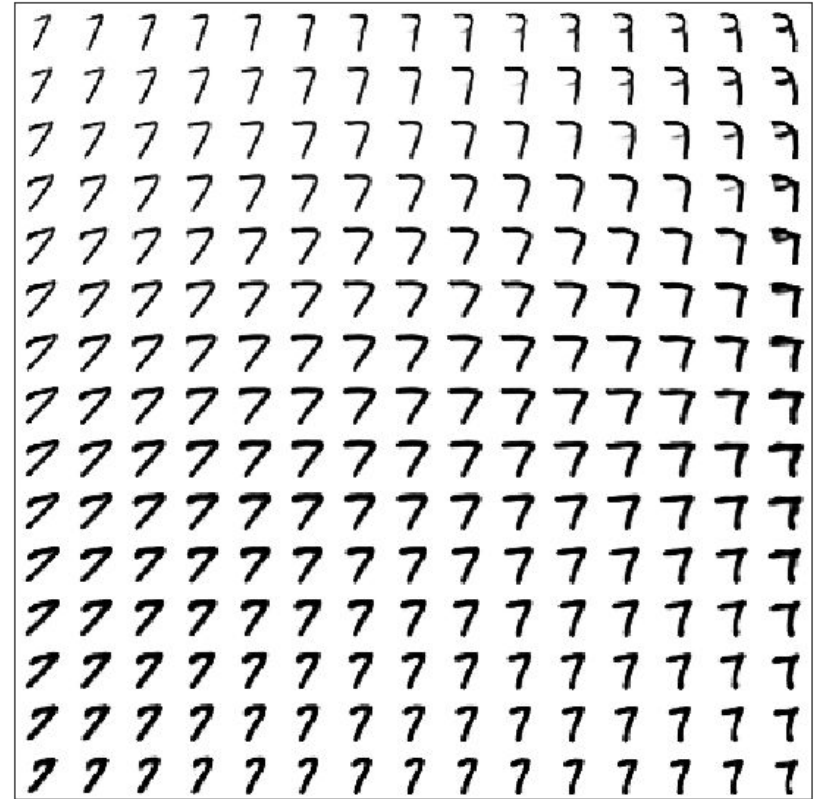
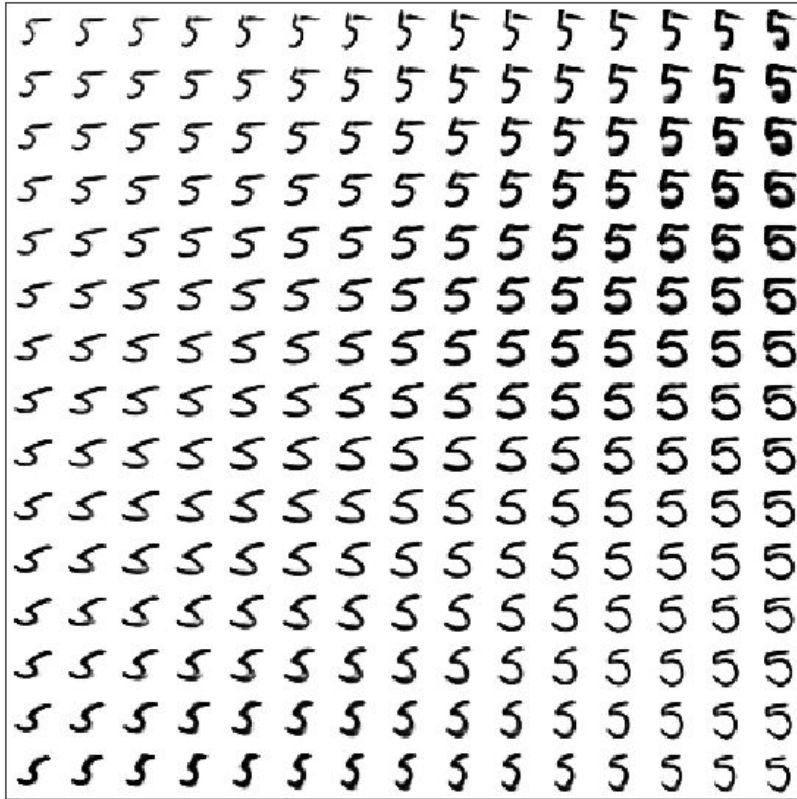
Label: all  
Batch: 0



# Conditional GAN



# cGAN manifolds



# Some more combinations



VAE

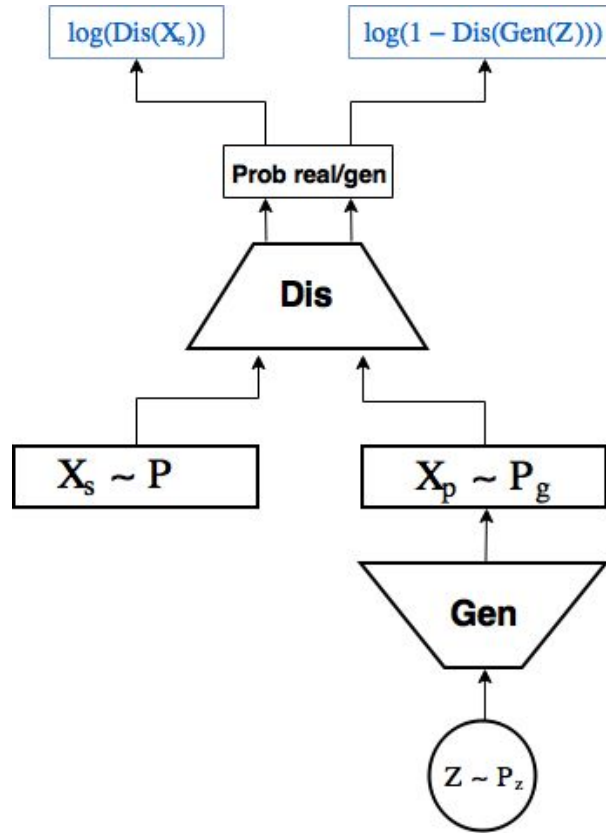
learning latent  
distribution

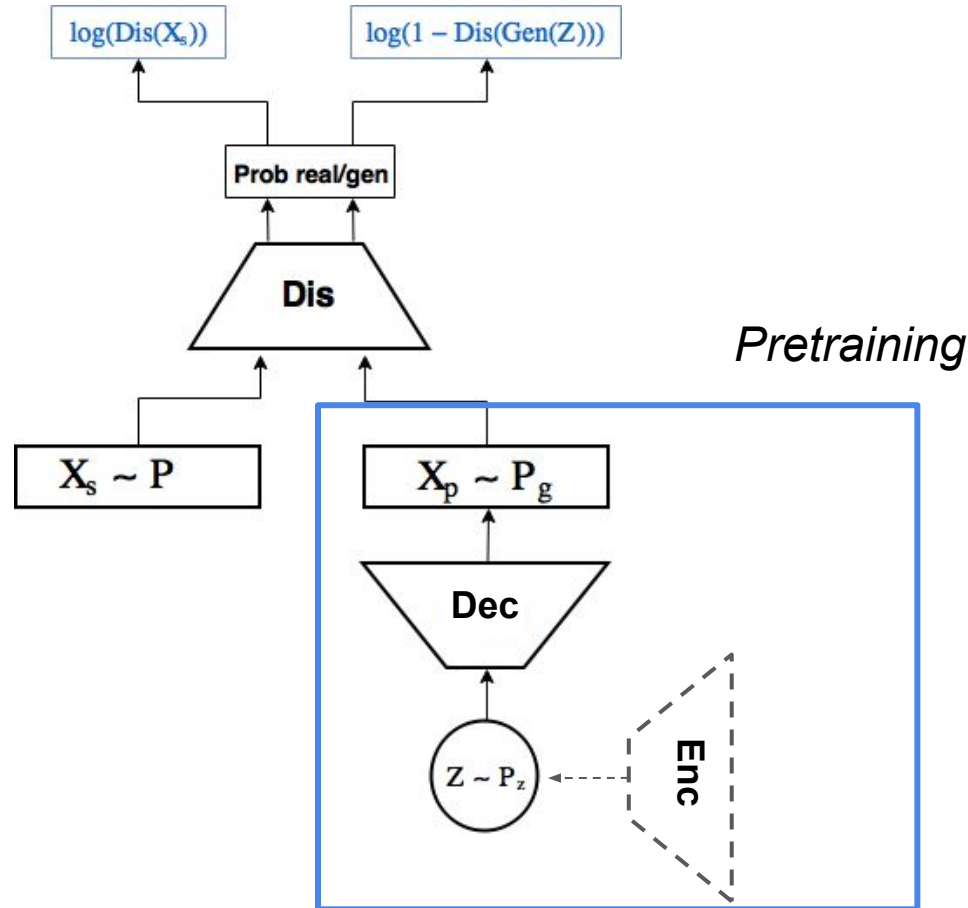
GAN

looking for good  
latent distribution

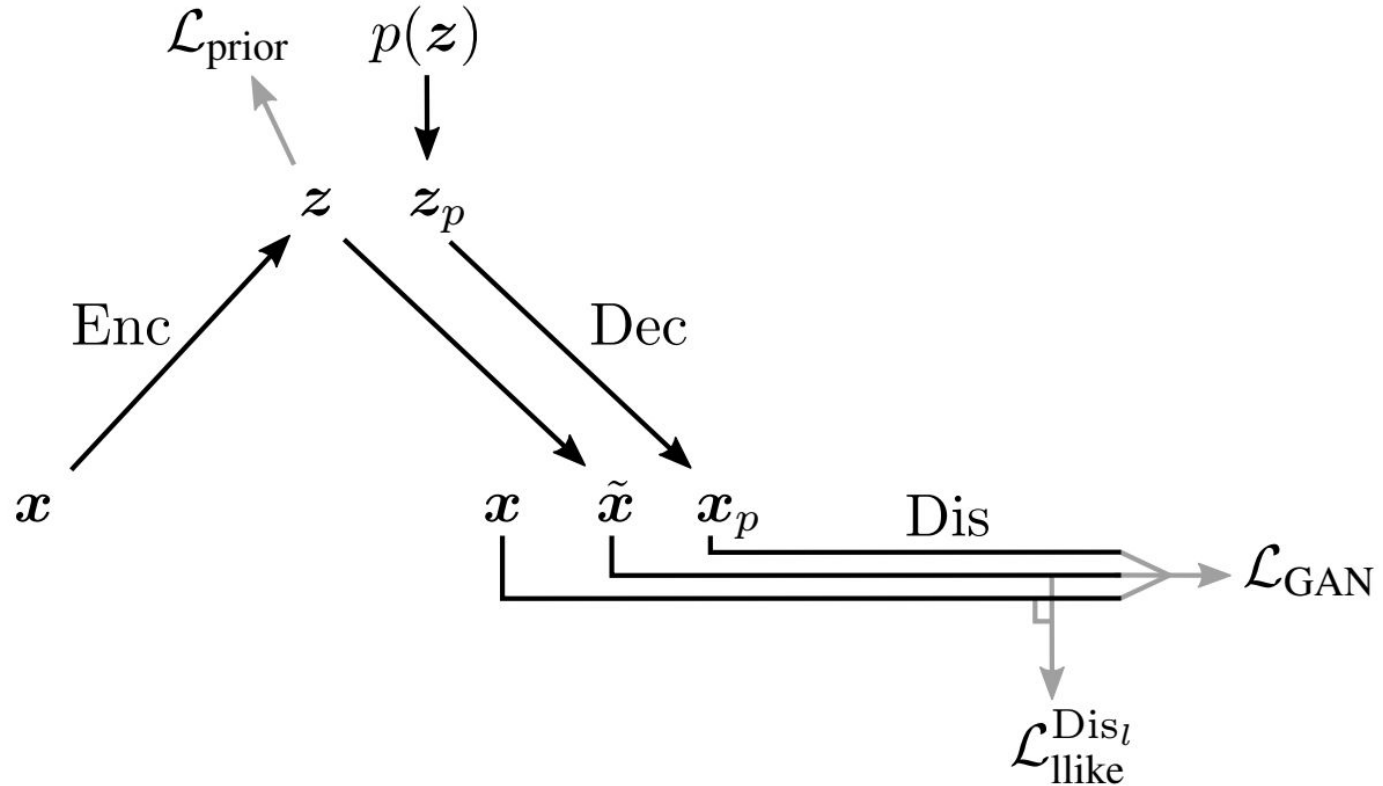


# Simple GAN





# VAE/GAN original illustration



# Q & A and farewell

Time to write some code!