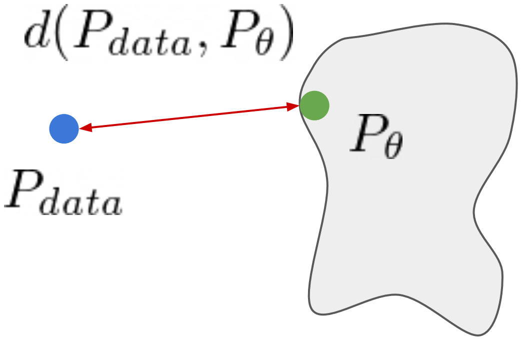
Deep Generative Models

Dr. Svetlin Penkov

Generative Modelling

$$x_i \sim P_{data}, i = 1, \ldots, N$$



$$\theta \in M$$

Find **Θ*** such that:

1. <u>Generation</u>: If $x_{new} \sim P_{\Theta^*}(x)$ then x_{new} should look like a car
2. <u>Density estimation</u>: $P_{\Theta^*}(x)$ should be high if x is a car, and low otherwise
3. <u>Representation learning</u>: Learn common attributes amongst x_i's

Deep Generative Models

- Encode distributions using deep neural networks...

Representing Distributions

- Continuous RVs e.g.

$$\mathcal{N}(\mu_\theta(x), \Sigma_\theta(x))$$

  where μ_θ and Σ_θ are neural networks parameterised by θ.
- Similarly, one can encode other continuous distributions (Gamma, Beta, etc…)

Representing Distributions

- Discrete RVs e.g.

$$Cat(\theta_1, \ldots, \theta_k)$$

$$\theta_i = \frac{e^{z_i}}{\sum_{j=0}^{K} e^{z_j}} = \frac{e^{f_i(x)}}{\sum_{j=0}^{K} e^{f_j(x)}}$$

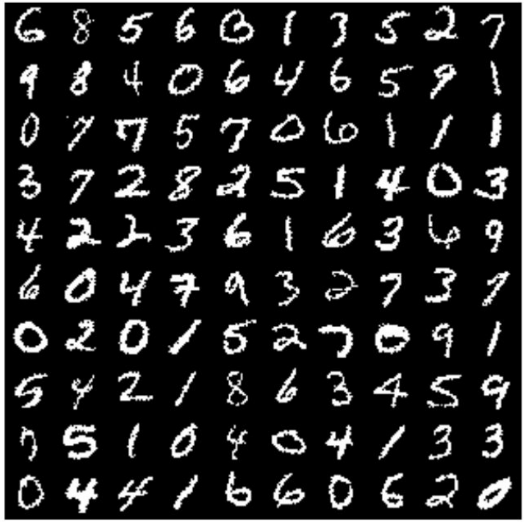
$$\theta = \mathrm{softmax}(\mathbf{f}_\phi(x))$$

  where $\mathbf{f}_\phi$ is a neural network parameterised by ϕ.

- Similarly, one can encode other discrete distributions (Bern, Mulinom, etc…)

Train on MNIST

Data



Generated Samples

- 28x28 binary images

$$x \in \{0, 1\}^{784}$$

- Factorise distribution

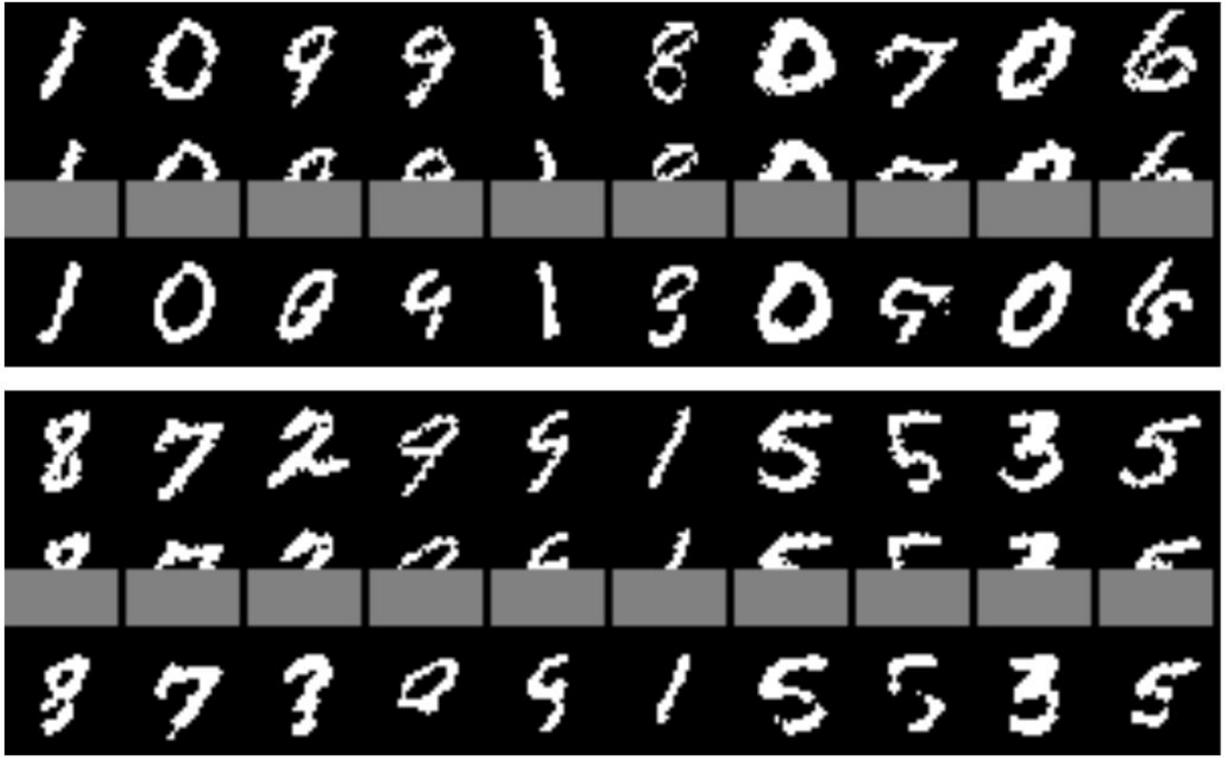
$$p(x|\alpha) = p(x_1|\alpha_1)p(x_2|x_1, \alpha_2) \ldots p(x_{784}|x_1, \ldots x_{783}, \alpha_{784})$$

where

$$p(x_i|x_{j<i}, \alpha_i) = \sigma(\alpha_{i,0} + \sum_{j=1}^{i} \alpha_{i,j} x_j)$$

G. Zhe et al., *Artificial Intelligence and Statistics*, 2015.

Missing Data Prediction



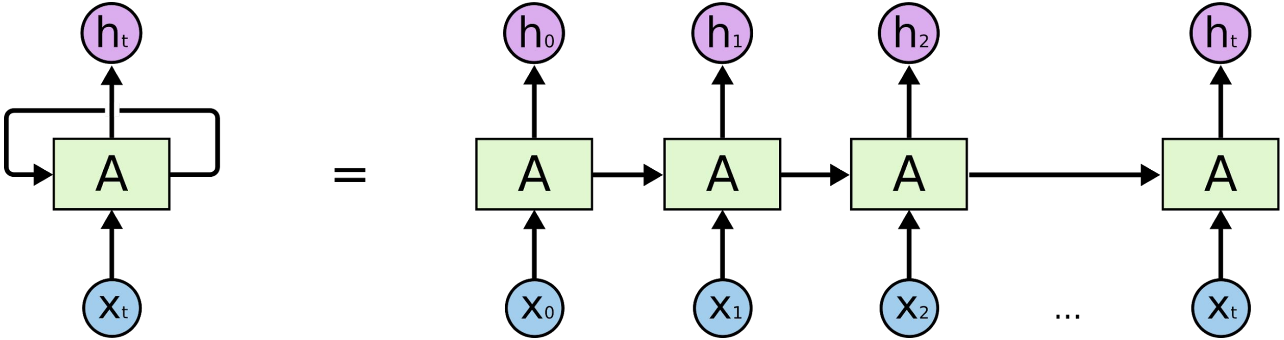
G. Zhe et al., *Artificial Intelligence and Statistics*, 2015.

PixelRNN

- Use an RNN to learn

$$p(x|\alpha) = p(x_1|\alpha_1)p(x_2|x_1, \alpha_2)\ldots p(x_{784}|x_1, \ldots x_{783}, \alpha_{784})$$

- Train RNN, such that $h_t = p(x_t|x_{1:t-1})$



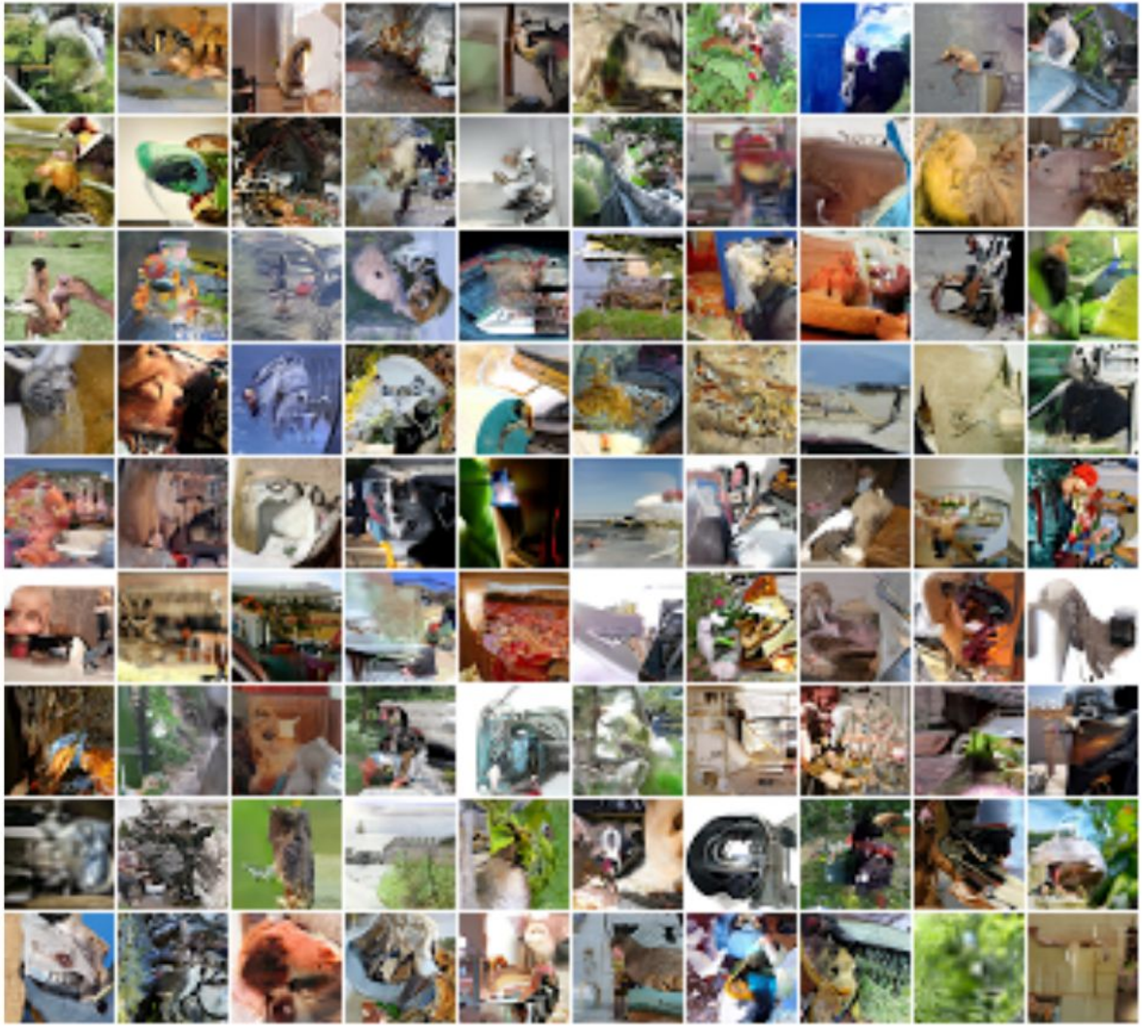
Oord et al., ICML, 2016

PixelRNN

- Deal with RGB images

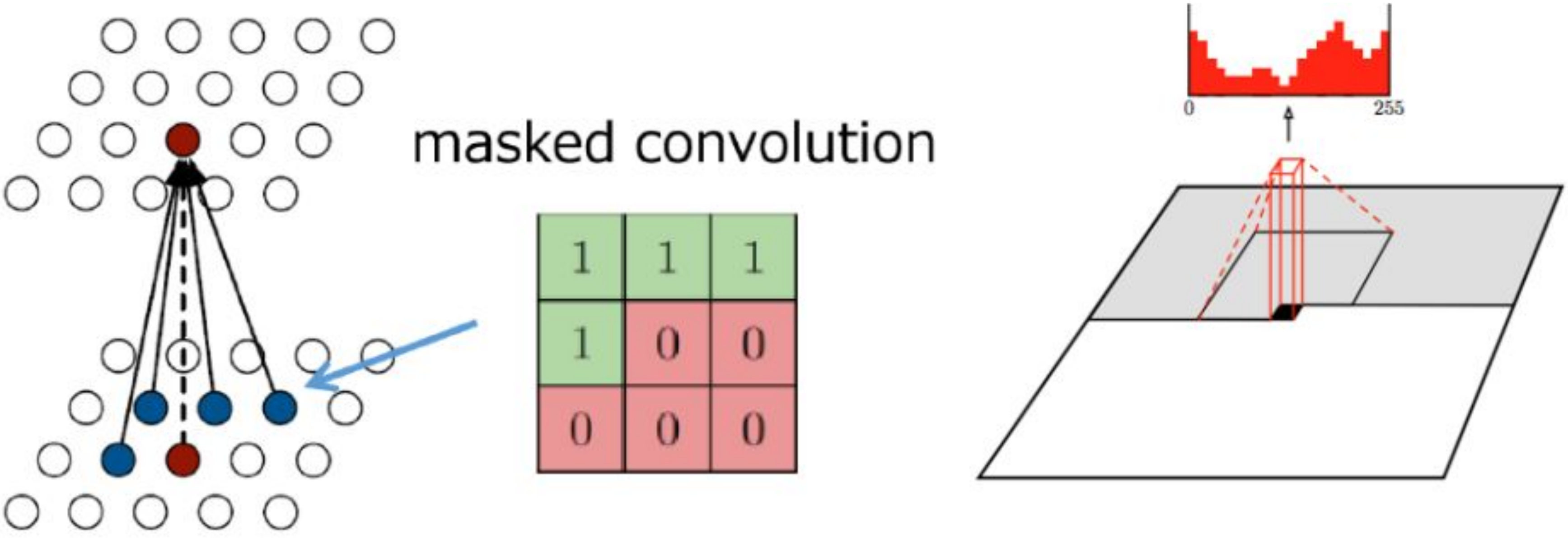
$$p(x_t|x_{1:t-1}) = p(x_t^R|x_{1:t-1})p(x_t^G|x_{1:t-1})p(x_t^B|x_{1:t-1})$$

Oord et al., ICML, 2016



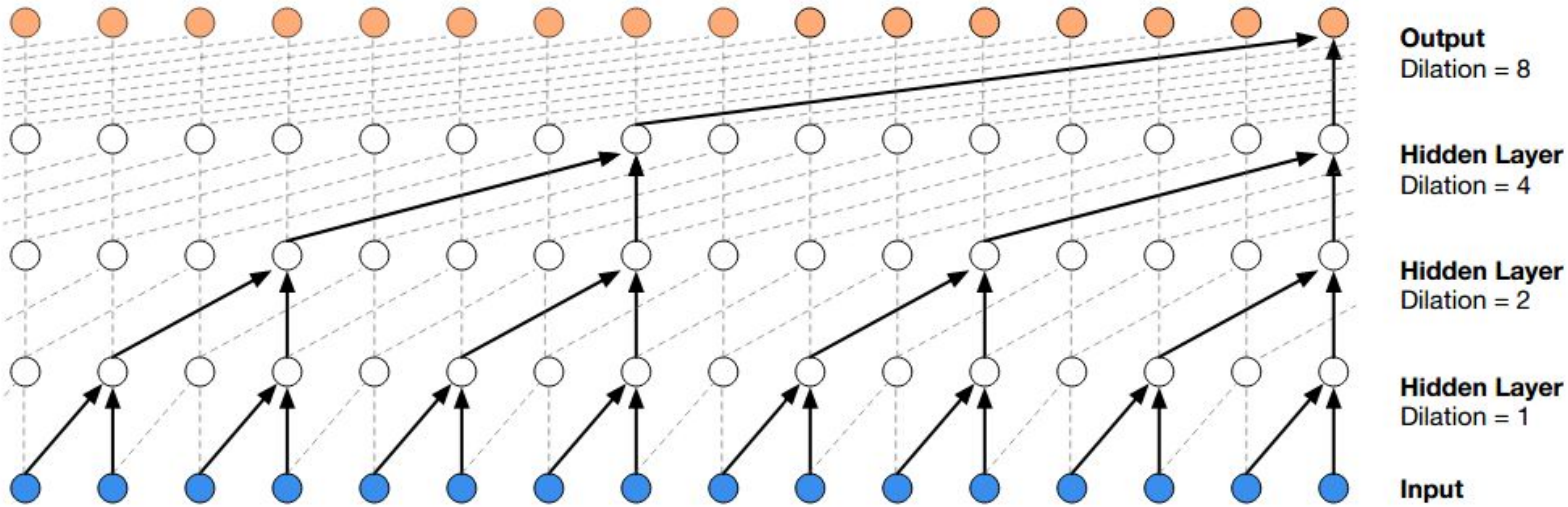
Oord et al., ICML, 2016

PixelCNN



Oord et al., ICML, 2016

WaveNet



https://deepmind.com/blog/article/wavenet-generative-model-raw-audio

Variational Autoencoder

- Find common features amongst x_i's
- Model the features as latent random variables
- Expected data likelihood per point x_i

$$l(x_i, \theta) = -\mathbb{E}_{p(z|x_i)}[\log p_\theta(x_i|z)]$$

Variational Autoencoder (VAE)

- Find common features amongst x_i's
- Model the features as latent random variables
- Expected data likelihood per point x_i

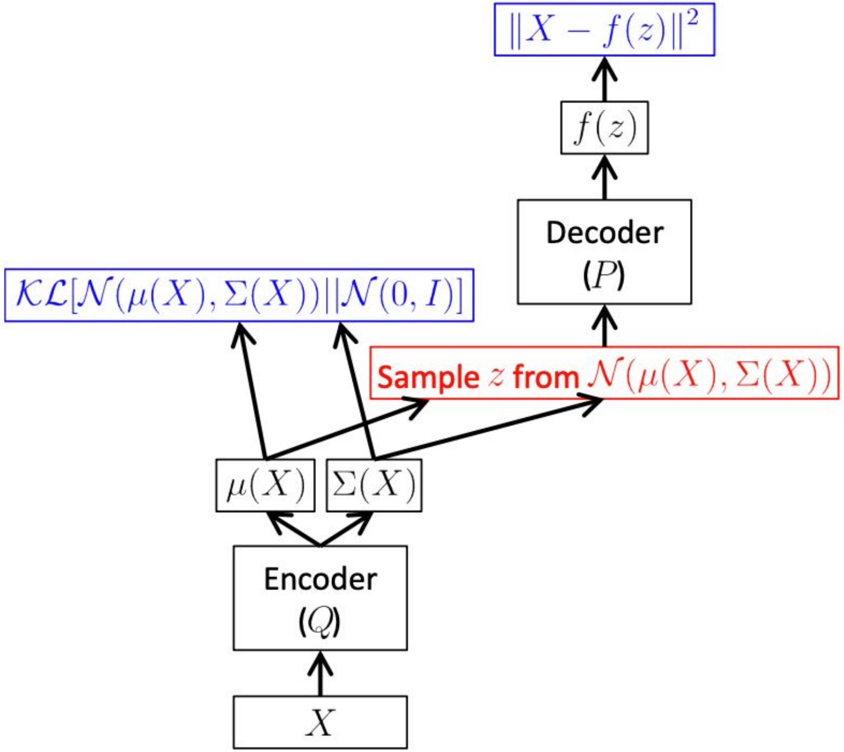
$$l(x_i, \theta) = -\mathbb{E}_{p(z|x_i)}[\log p_\theta(x_i|z)]$$

- Posterior is intractable so approximate

$$l(x_i, \theta, \phi) = \underbrace{-\mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)]}_{\text{Expected Reconstruction Error}} + \underbrace{\mathbb{KL}(q_\phi(z|x_i)||p(z))}_{\text{Regulariser}}$$

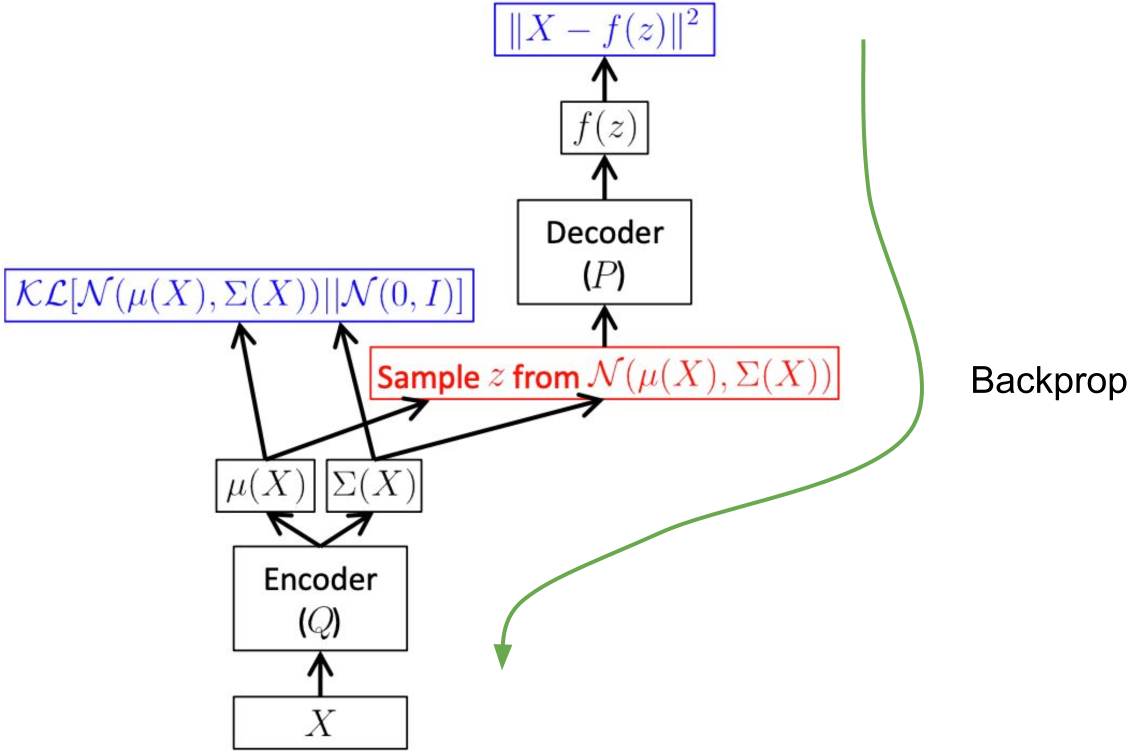
Kingma et al, arXiv:1312.6114, 2014

VAE Computational Graph



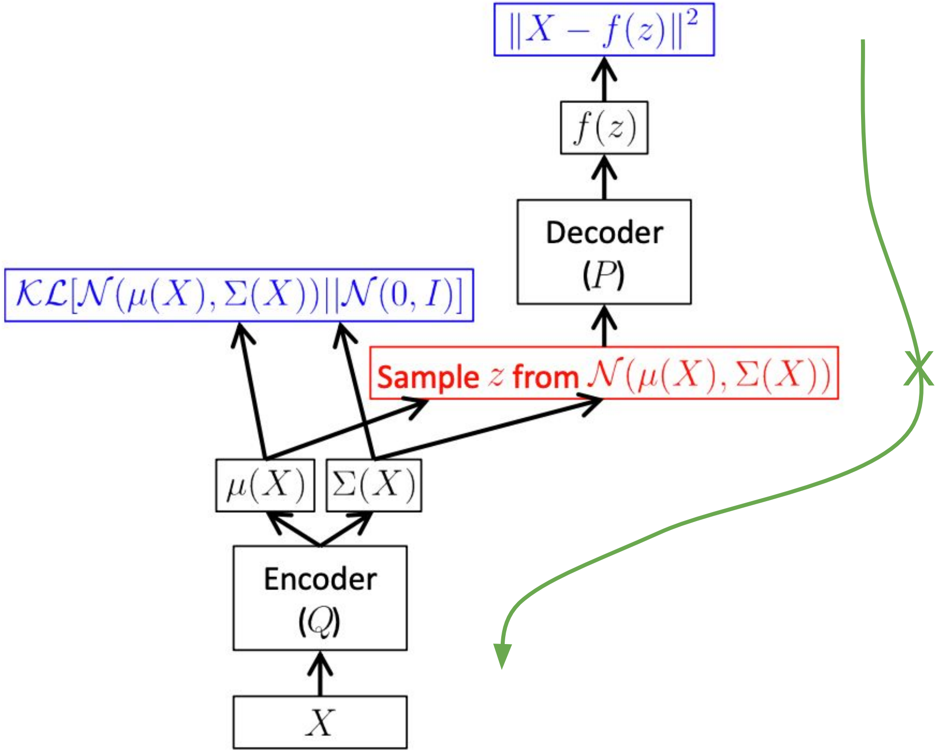
Kingma et al, arXiv:1312.6114, 2014

VAE Computational Graph



Kingma et al, arXiv:1312.6114, 2014

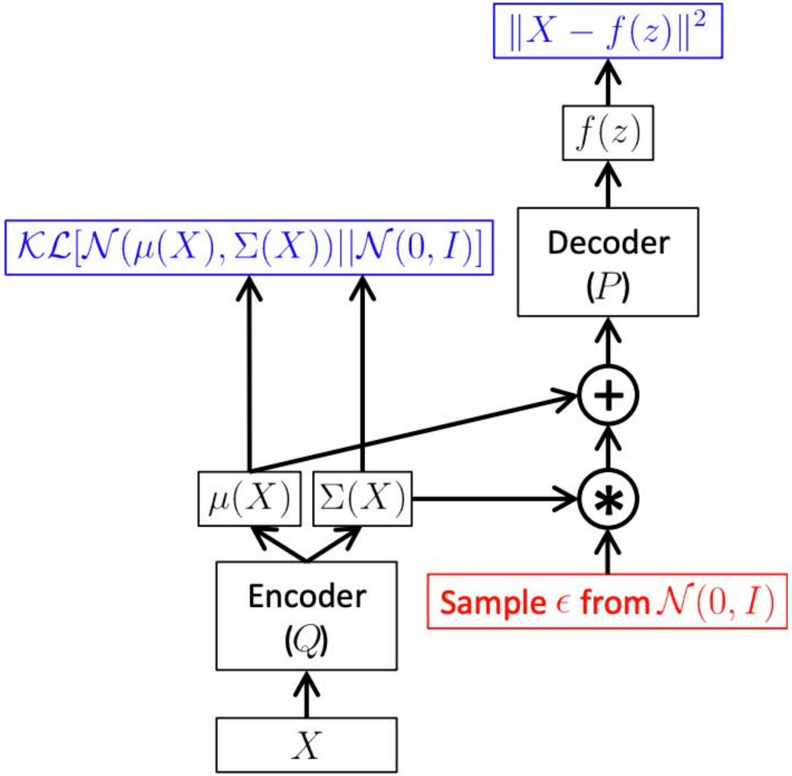
VAE Computational Graph



Sampling is a non-differentiable operation

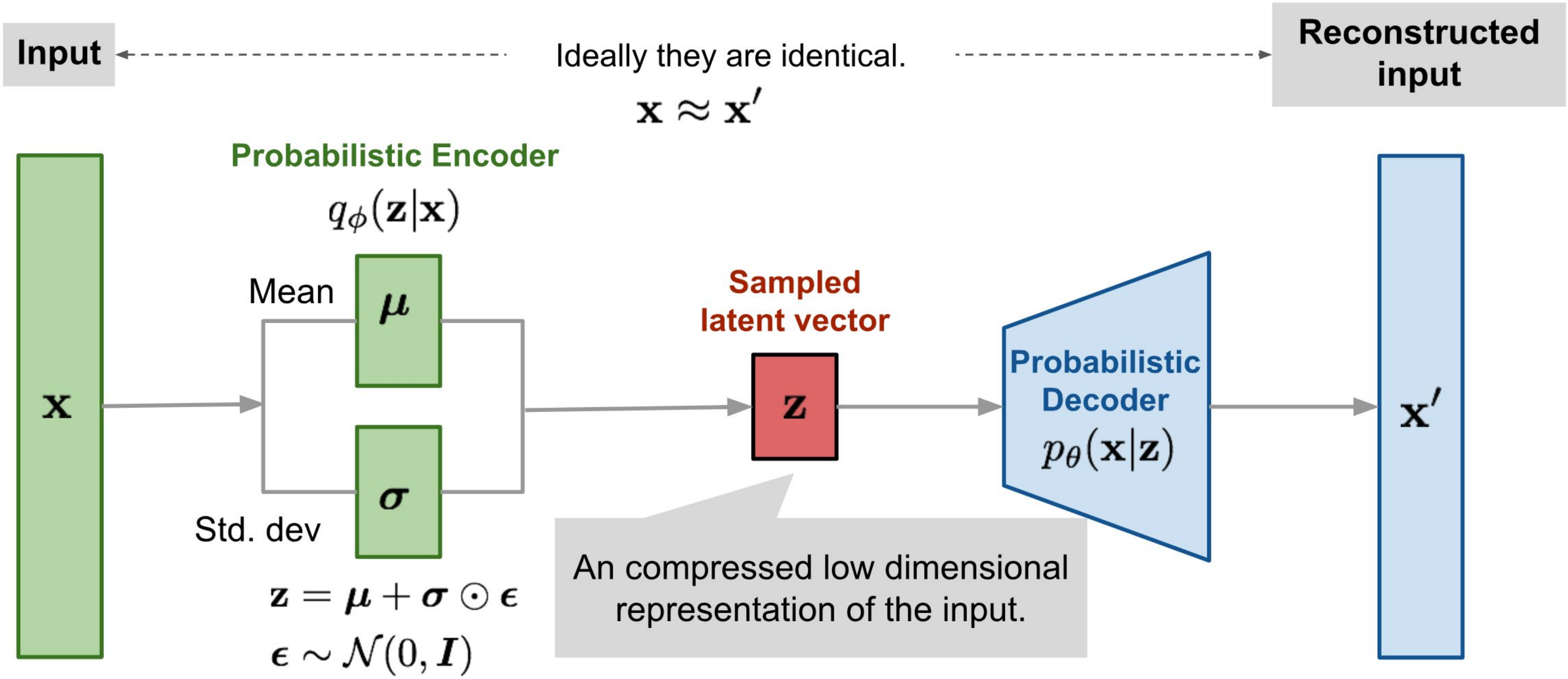
Kingma et al, arXiv:1312.6114, 2014

VAE Reparametrisation Trick



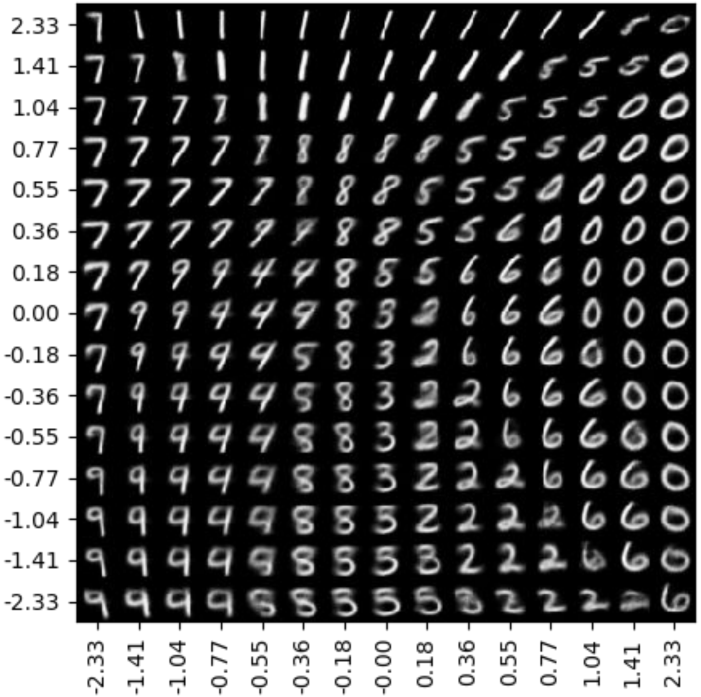
Kingma et al, arXiv:1312.6114, 2014

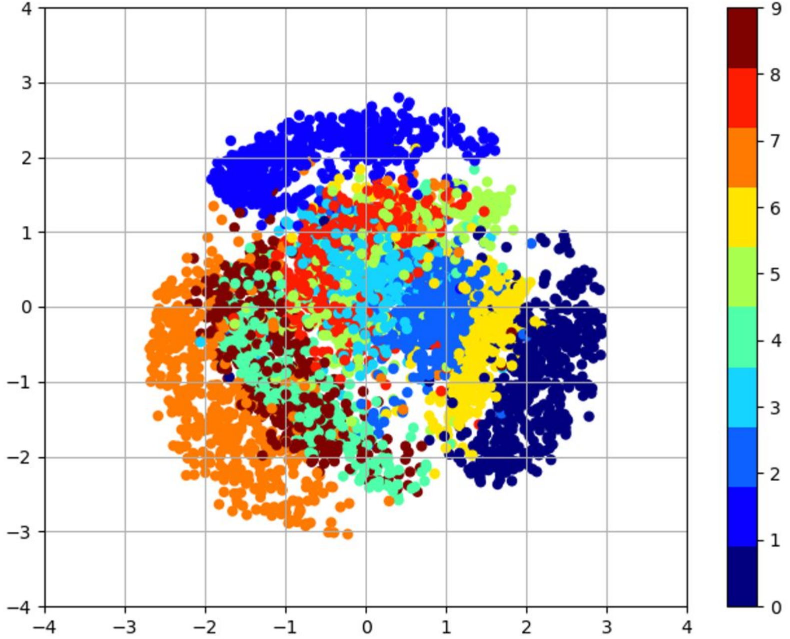
VAE



https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

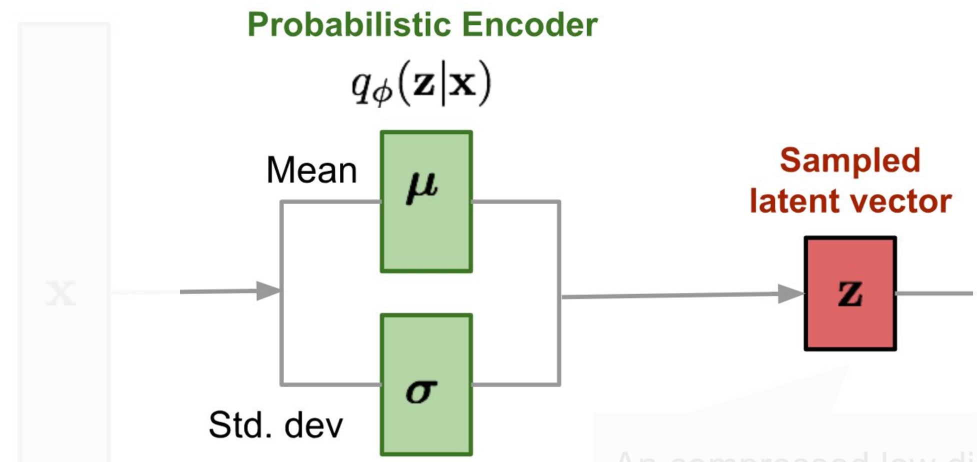
Latent Space Manifolds





https://github.com/ChengBinJin/VAE-Tensorflow
http://louistiao.me/listings/vae/.ipynb_checkpoints/convolutional_variational_autoencoder_mnist-checkpoint.ipynb.html

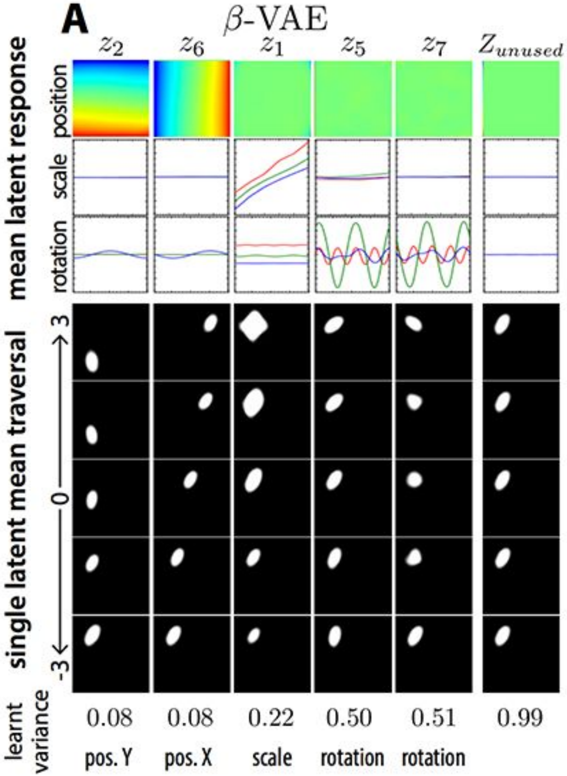
Representation Learning



- Maximise independence
- Minimise mutual information
- Use structured proposal *q*
- .
- .

β-VAE

$$l(x_i, \theta, \phi) = -\mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)] + \beta \mathbb{KL}(q_\phi(z|x_i)||p(z))$$

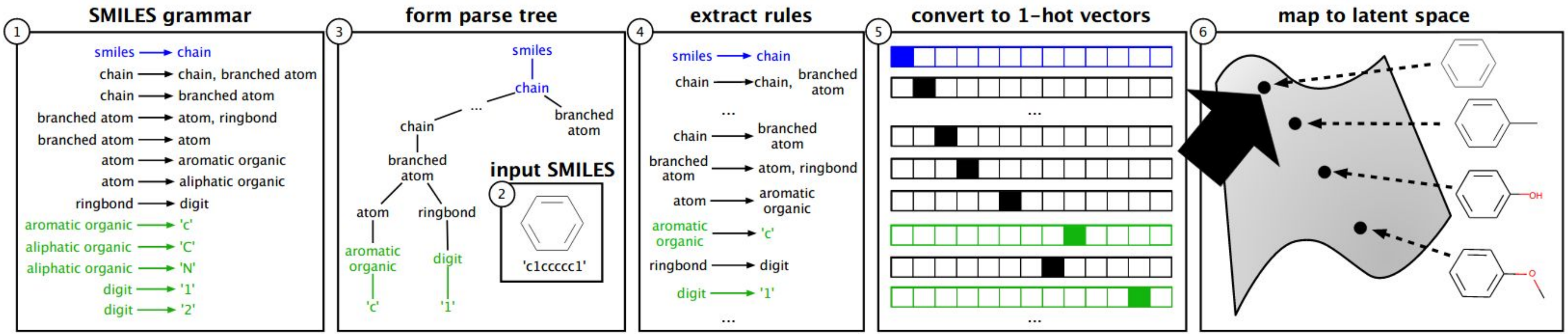


Higgins et al., *ICLR,* 2017

Structured data

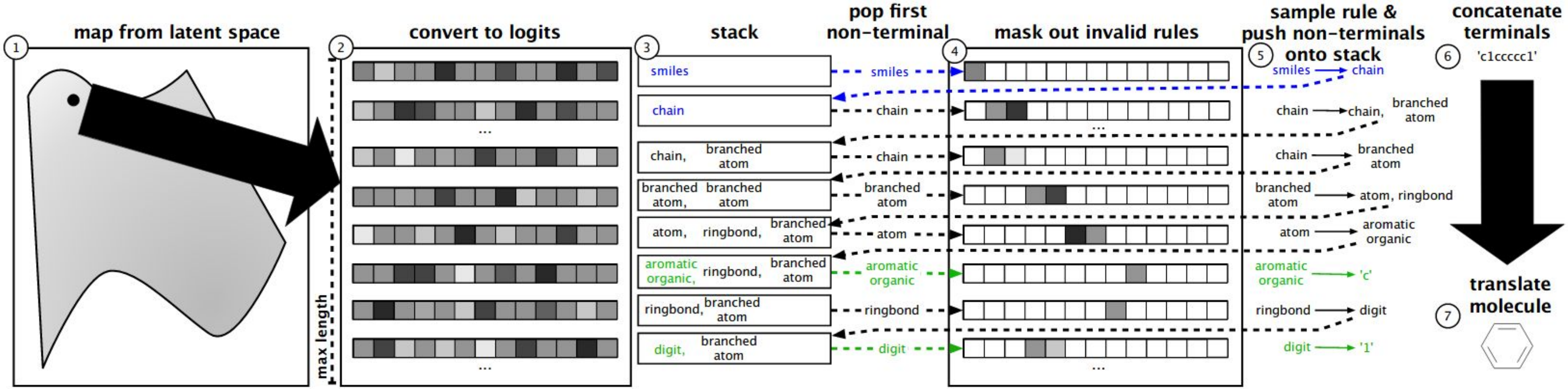
- Lots of redundancy in images
- What if we want to generate well formed strings
  - e.g. maths expressions, programs, DNA sequences, chemical molecule descriptions
- Even a single error could make the sample useless

Grammer VAE (GVAE) - Encoder



Kusner et al., ICML, 2018

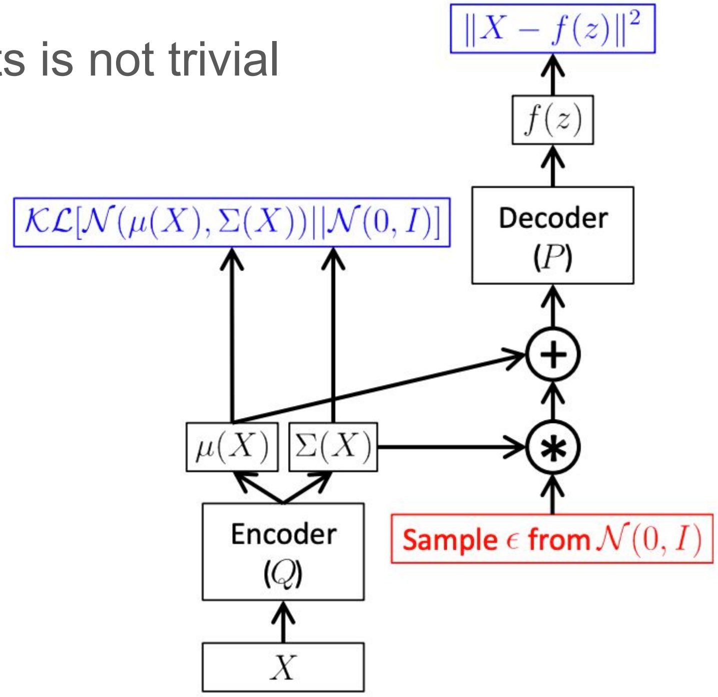
Grammer VAE (GVAE) - Decoder



Kusner et al., ICML, 2018

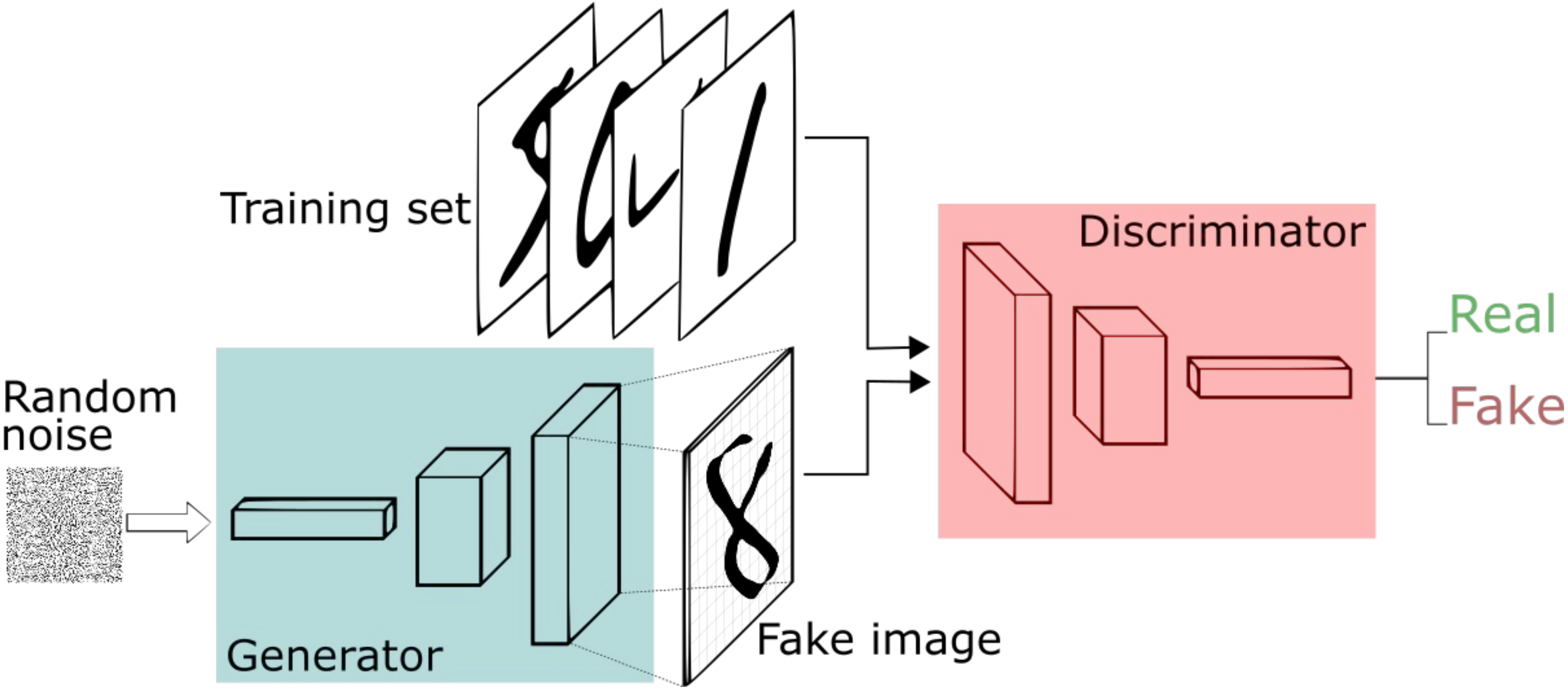
Distance Metric

- Sampling more complex images tends to give blurry results
- Defining a distance metric for complex objects is not trivial
- Can we avoid / learn the distance metric?



Generative Adversarial Networks (GANs)

Goodfellow, NeurIPS, 2014



https://sthalles.github.io/assets/dcgan/GANs.png

Generative Adversarial Networks (GANs)

- 2-player game objective function (i.e. How well are fake samples detected?)

$$\min_{\theta} \max_{\phi} V(G_\theta, D_\phi) = \underbrace{E_{\mathbf{x} \sim p_{\text{data}}}[\log D_\phi(\mathbf{x})]}_{\text{Does D output 1 when data is real?}} + \underbrace{E_{\mathbf{z} \sim p(\mathbf{z})}[\log(1 - D_\phi(G_\theta(\mathbf{z})))]}_{\text{Does D output 0 when data is from generator?}}$$

- G wants to deceive D (decrease objective)
- D wants to detect generated samples (maximise objective)
- Tricky to train (mode collapse)

4.5 Years of Progress on Faces

2014

2015

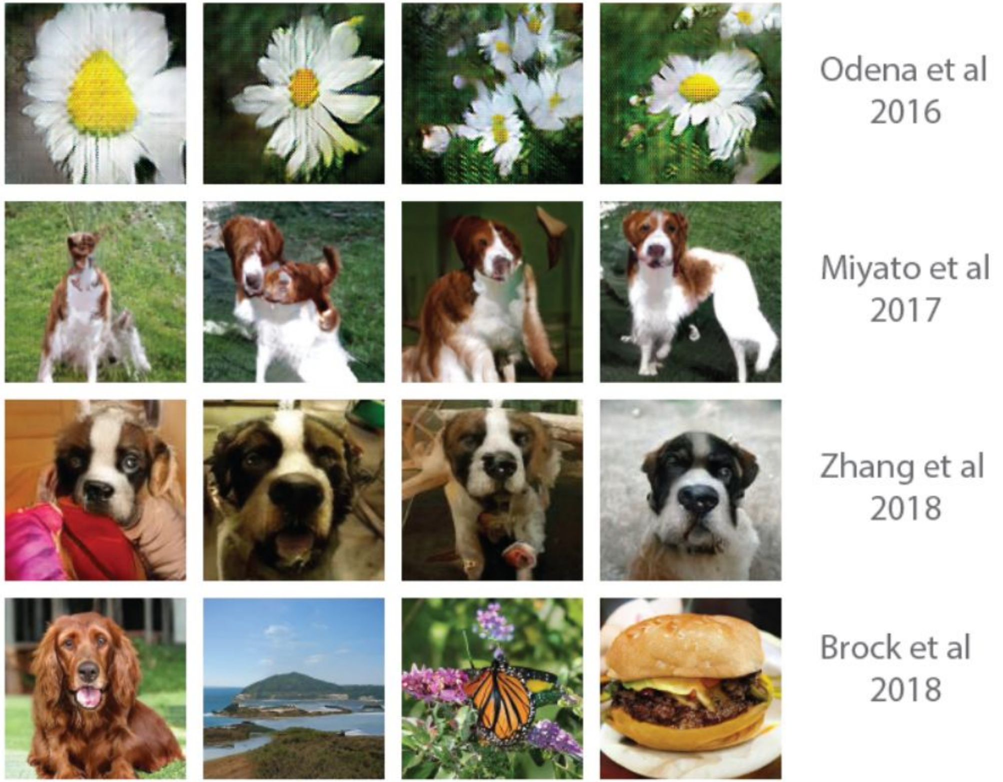
2016

2017

2018

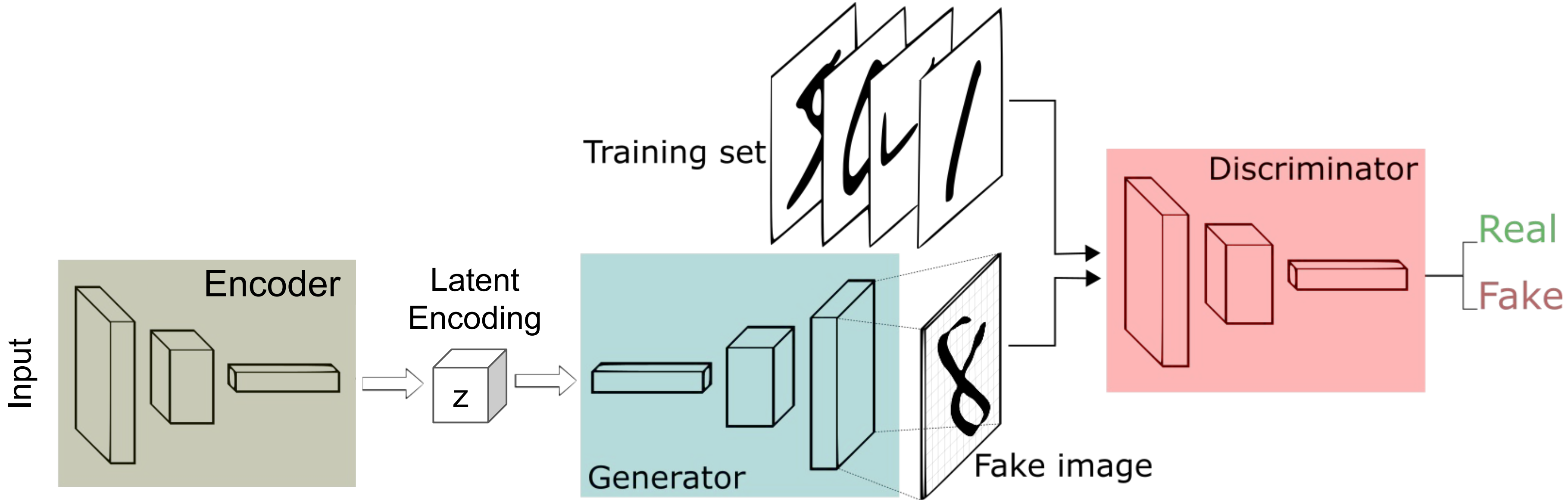
https://www.iangoodfellow.com/slides/2019-05-07.pdf

2 Years of Progress on ImageNet

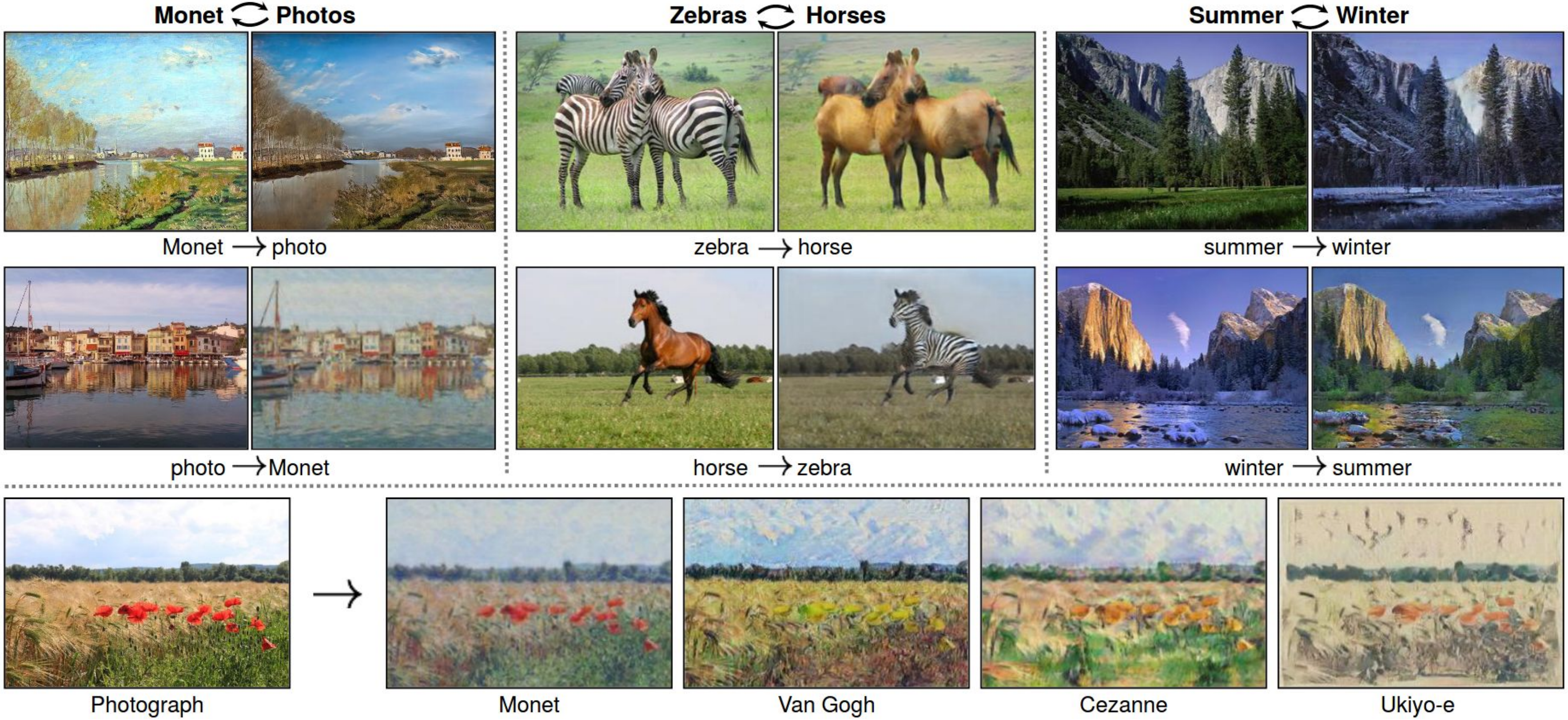


https://www.iangoodfellow.com/slides/2019-05-07.pdf

Conditional GANs

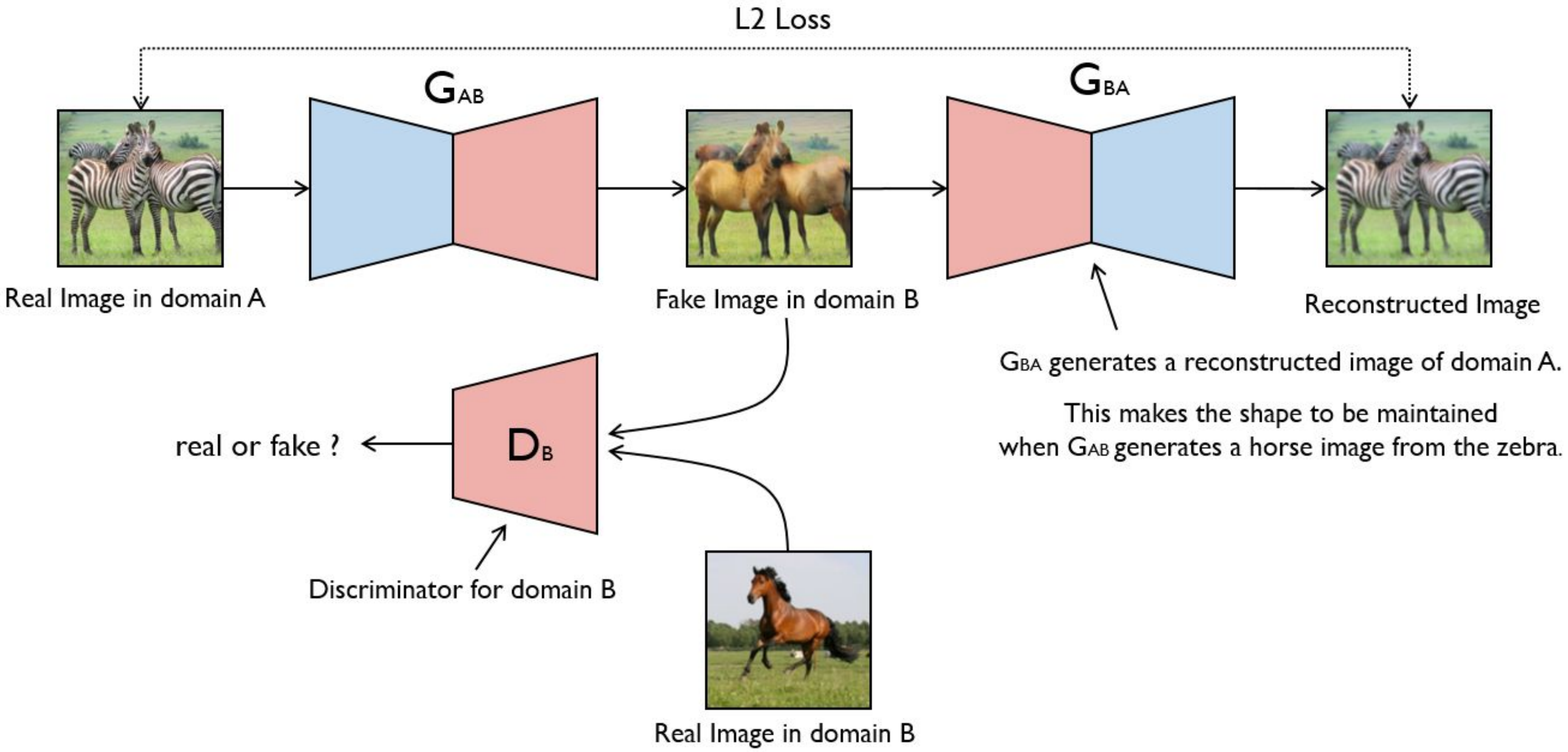


CycleGAN



Zhu et al, *ICCV*, 2017

CycleGAN



https://modelzoo.co/model/mnist-svhn-transfer

Everybody Dance Now

Chan et al., ICCV, 2019