Generative Models
(This time without GANs)

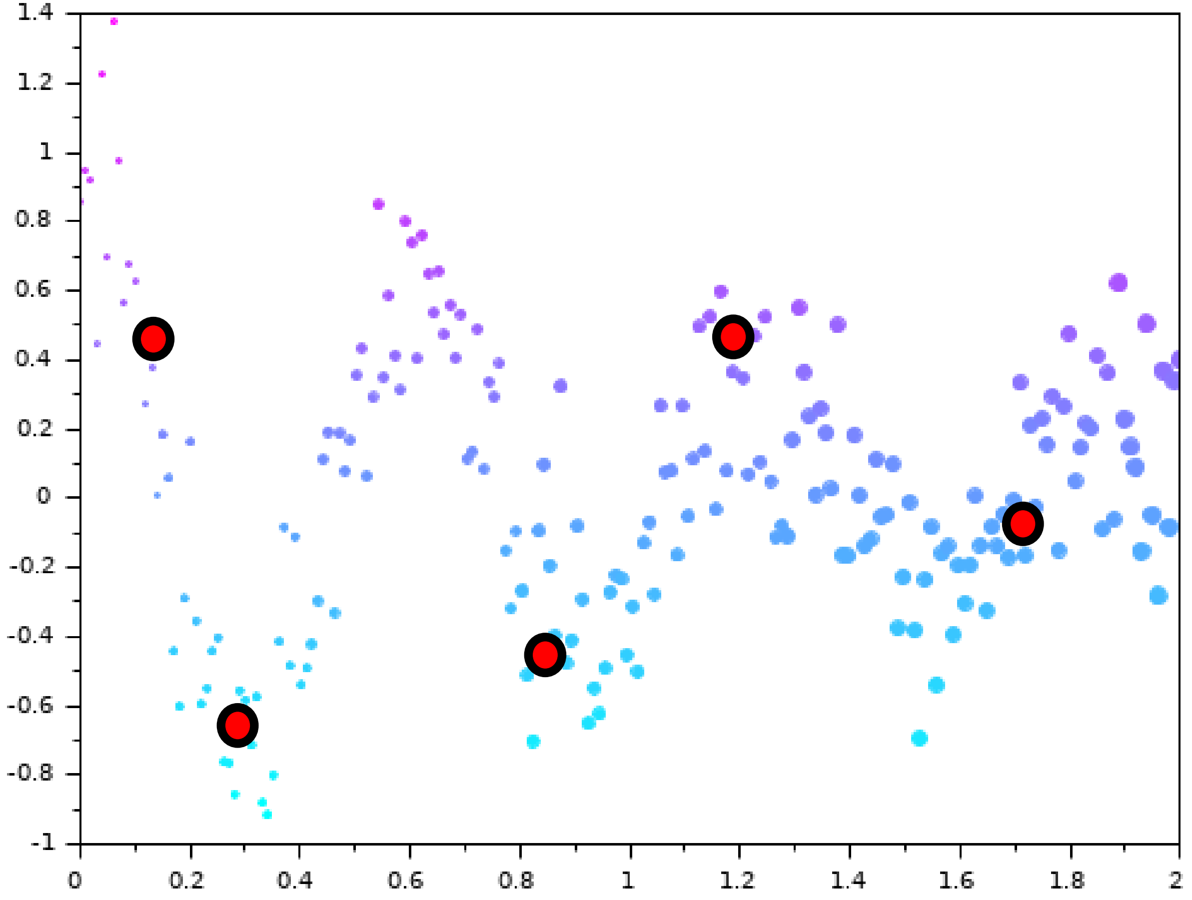
June 3rd, 2021

Slides & Figures credits:

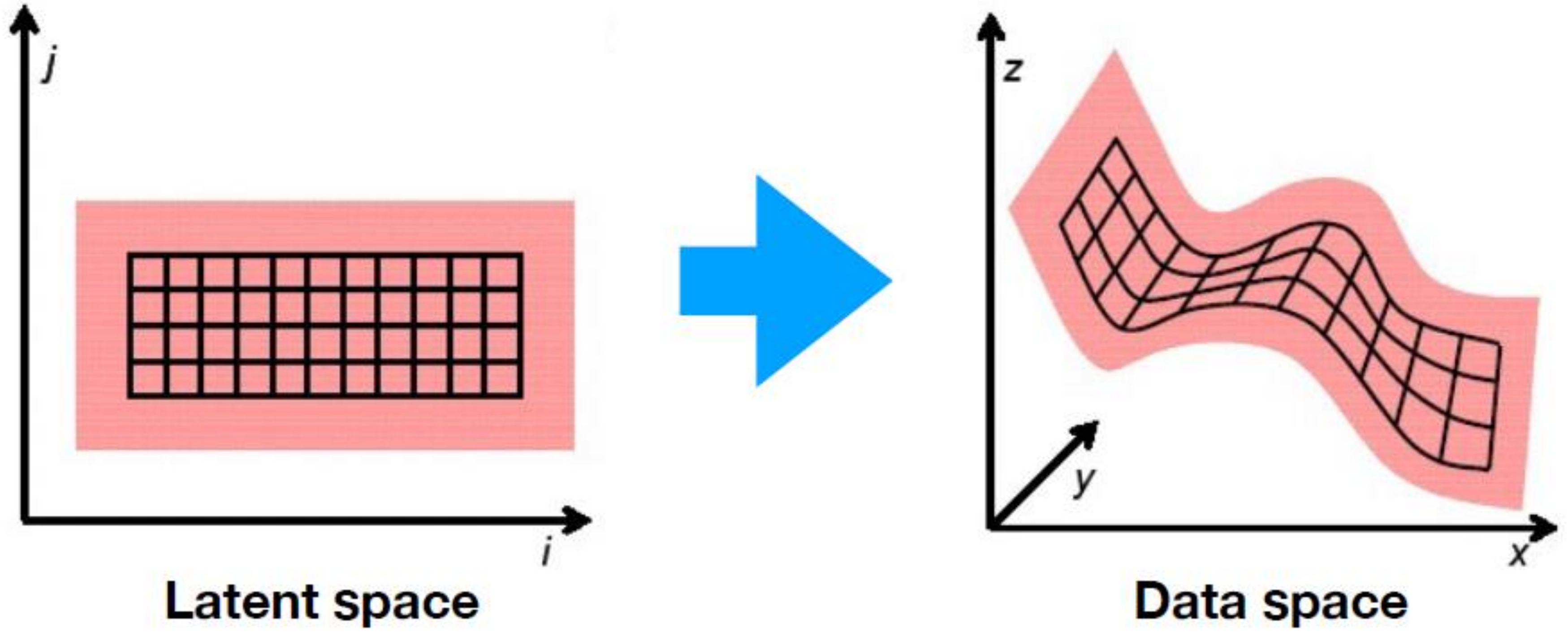
Assaf Shocher’s Slides, Introduction to Computer Vision 2020

Objective - Reminder

Goal: Find θ s.t. $\log\left(p_\theta\left(x^{(i)}\right)\right)$ is as high as possible

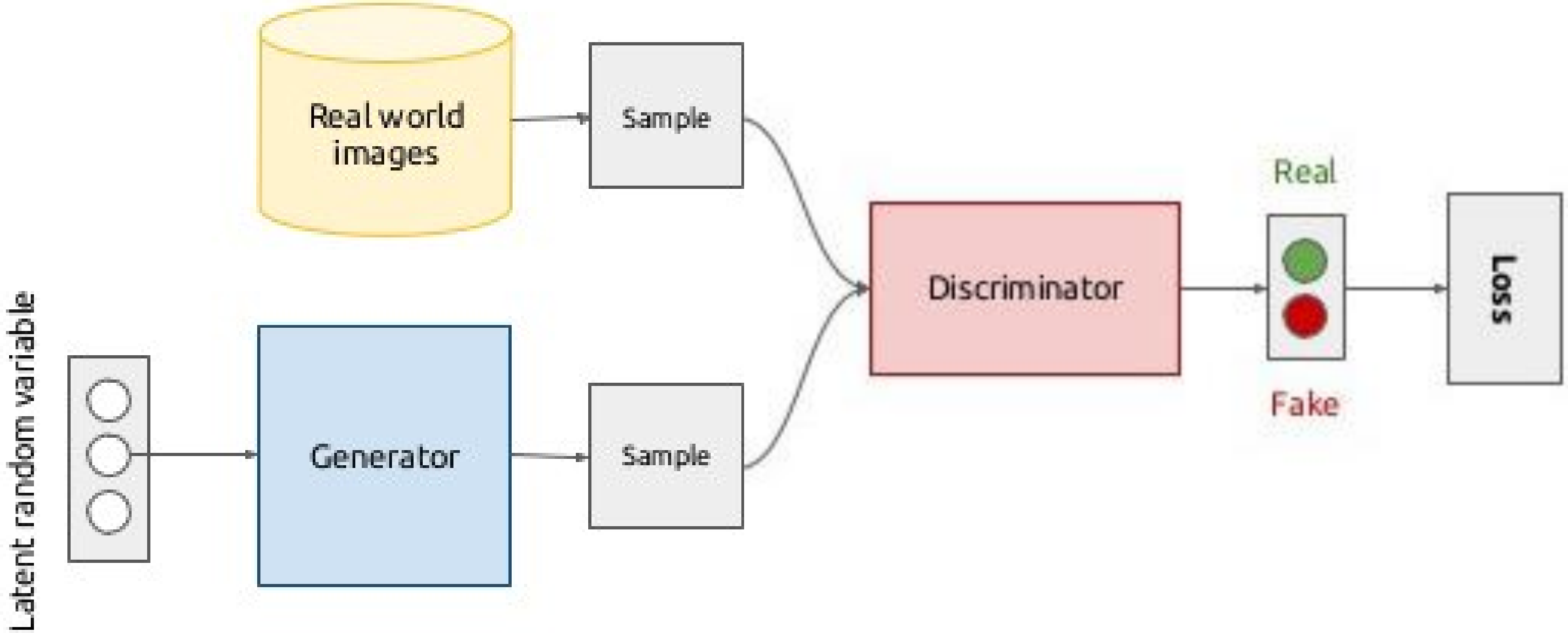


Latent Space Mapping Approach - Reminder



GANs- Reminder

[redacted]



$$\boxed{\mathcal{L}_{GAN} = \min_{G} \max_{D} \mathbb{E}_{x \sim p_{data}}[log\, D(x)] + \mathbb{E}_{z \sim p_z}[log(1 - D(G(z)))]}$$

Pros & Cons - GANs

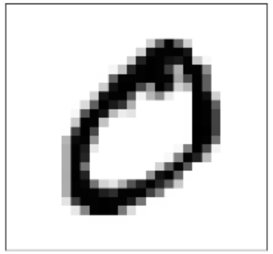
- Pros:
  - Very high qua
  - Fast inference

- Cons:
  - Difficult train
  - Evaluation is problematic
    - Using auxiliary methods: FID, IS
  - Mode collapse

Today – Generating Without GANs

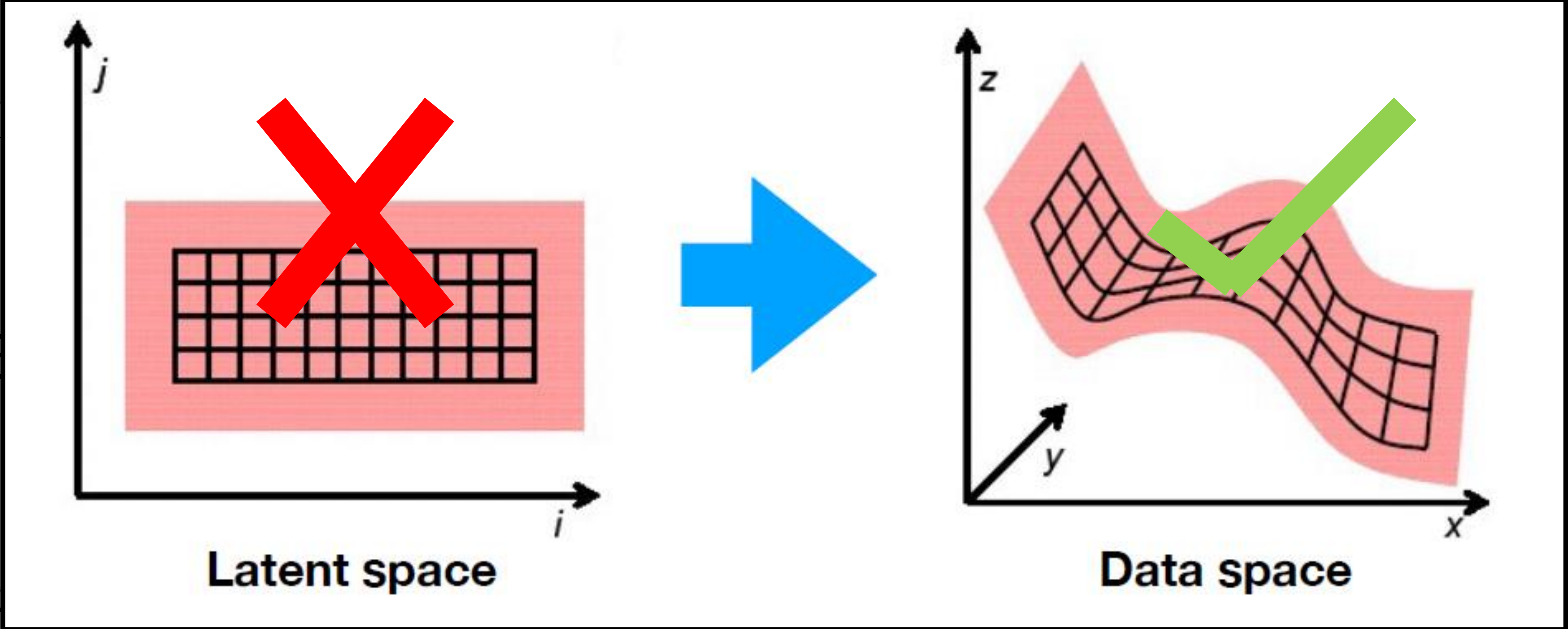
- Autoencoders
- Variational Autoencoders
- VQ-VAE + VA-VAE2
- IMLE

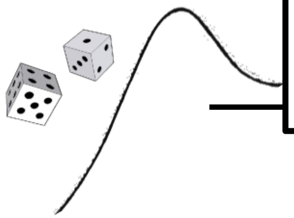
Autoencoders



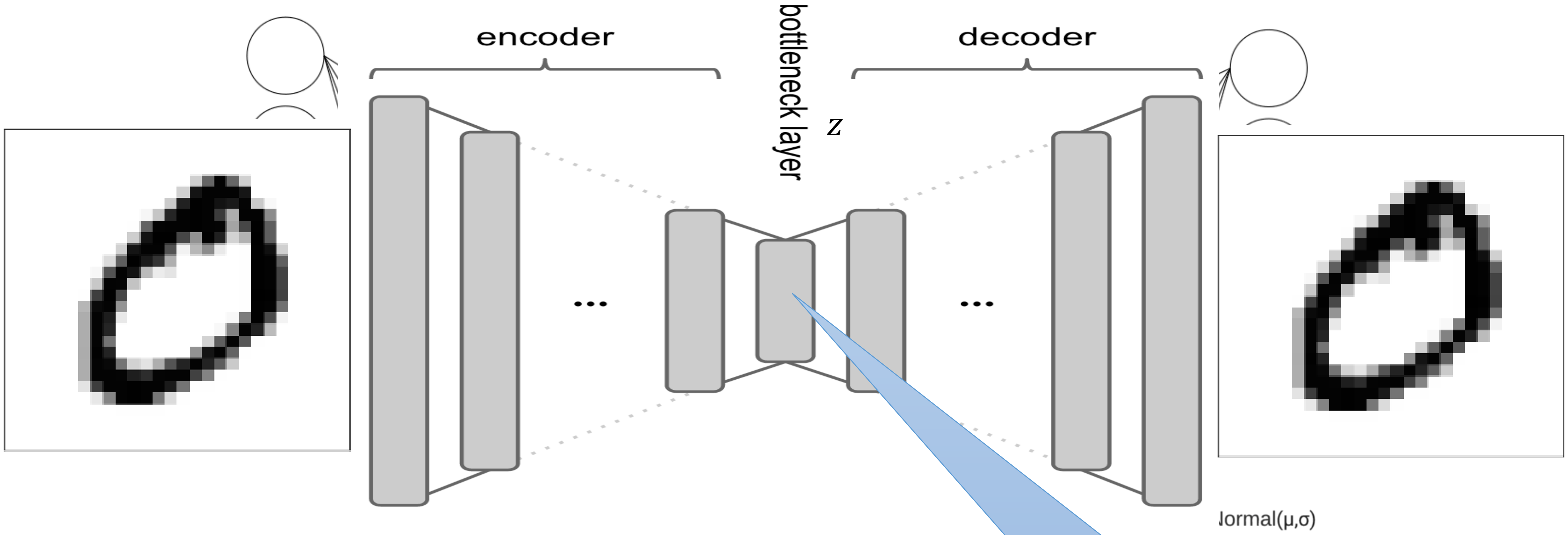
- Gene

AE does not transform one *pre-determined* distribution to another!





Variational Autoencoders (Kingma&Welling 14')

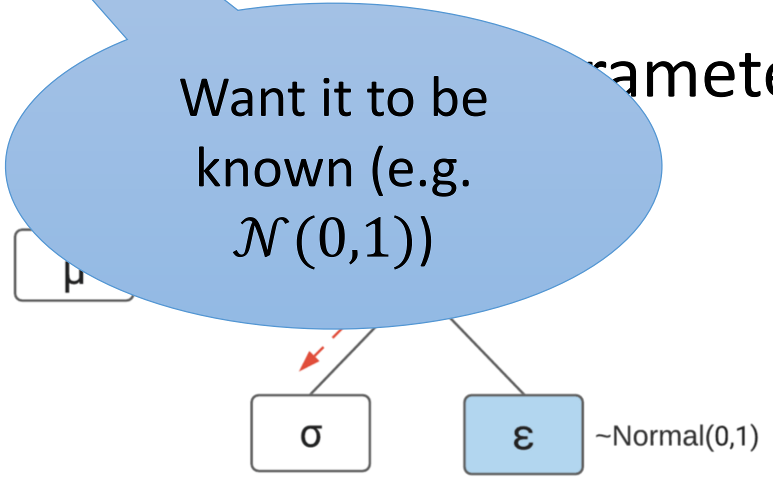


Encourage $p(z) \sim \mathcal{N}(0,1)$

by KL divergence:

$$\sum_{i=1}^{n} \sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1$$

rameterization

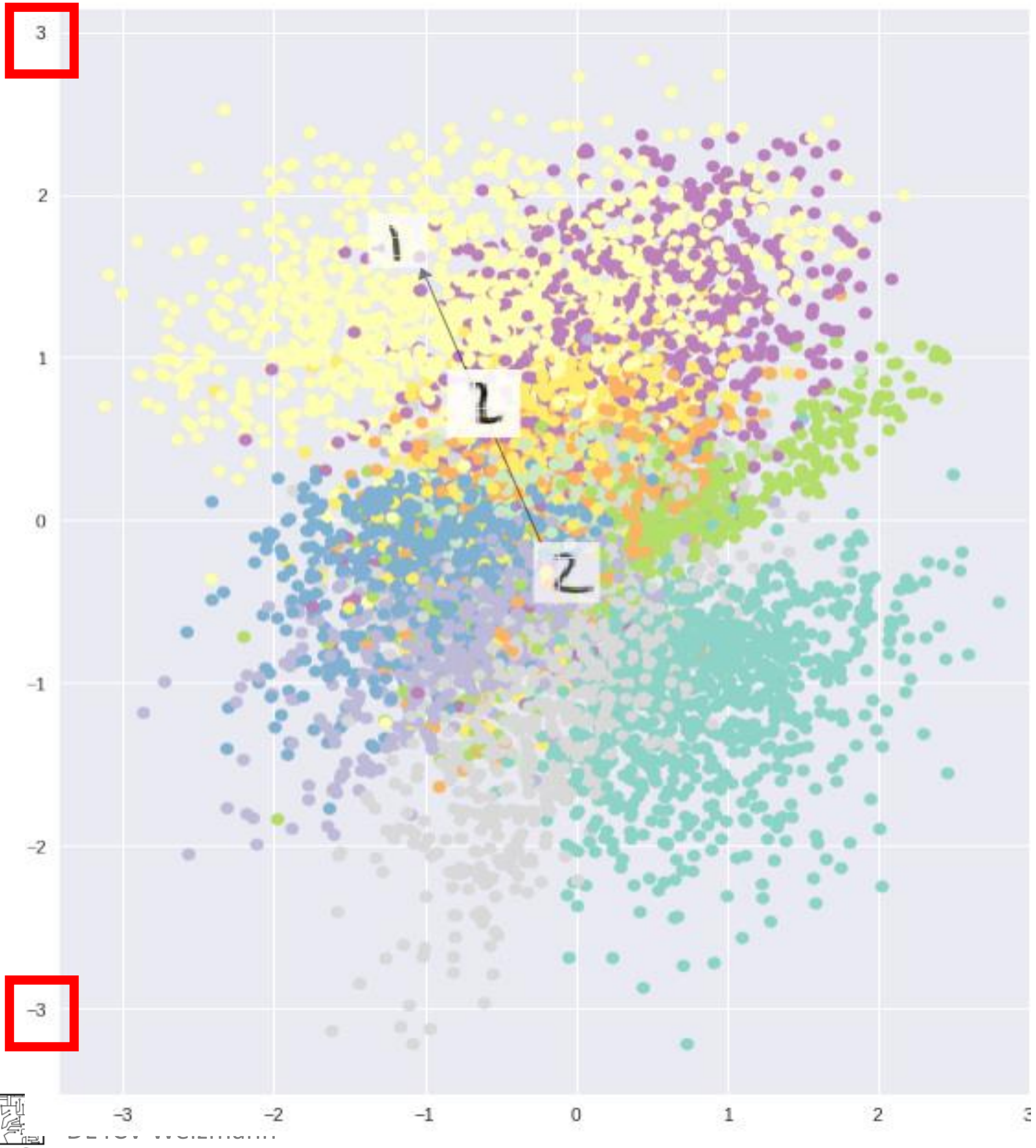


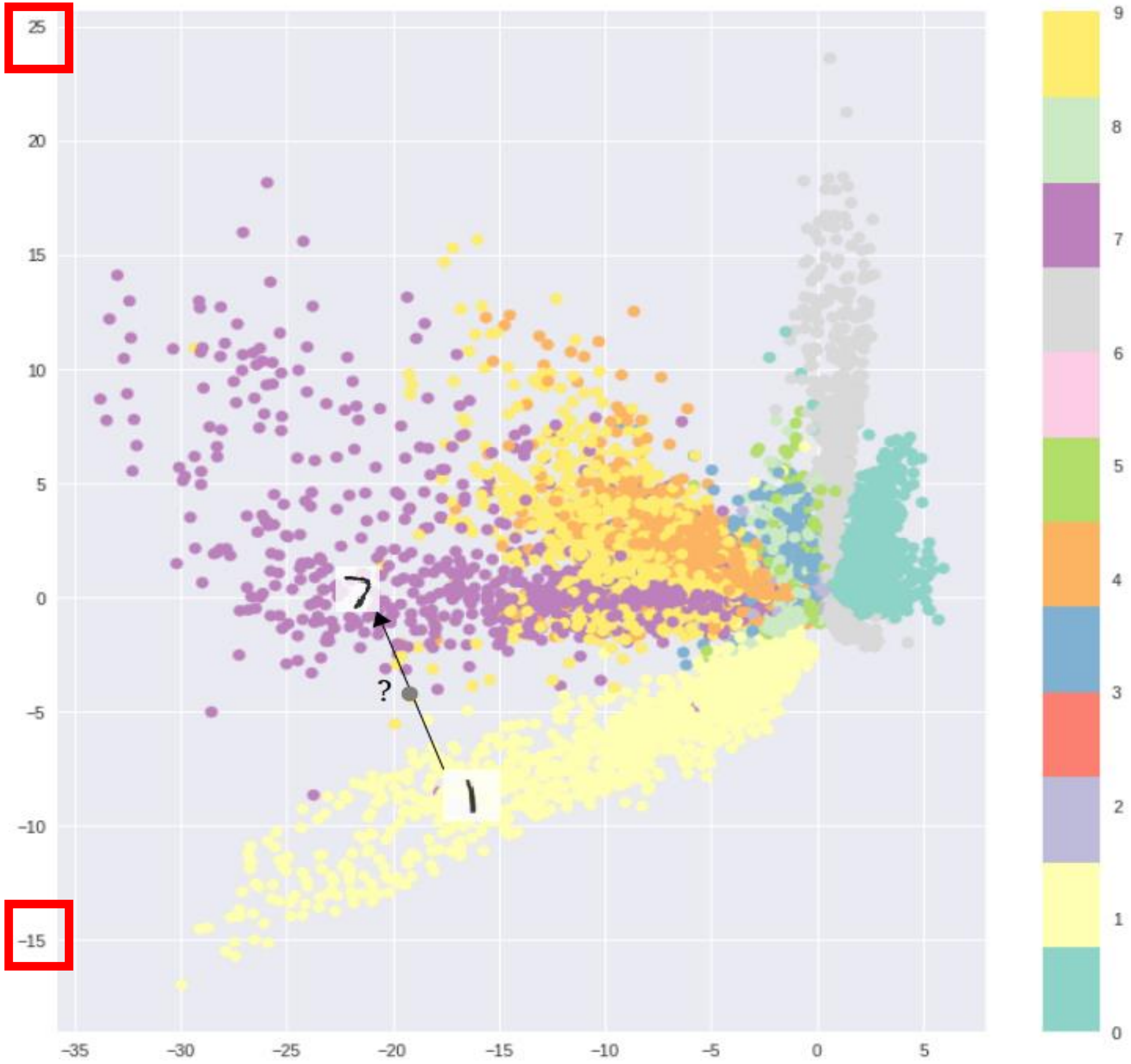
[redacted]

VAE

AE

Also check out the scale!



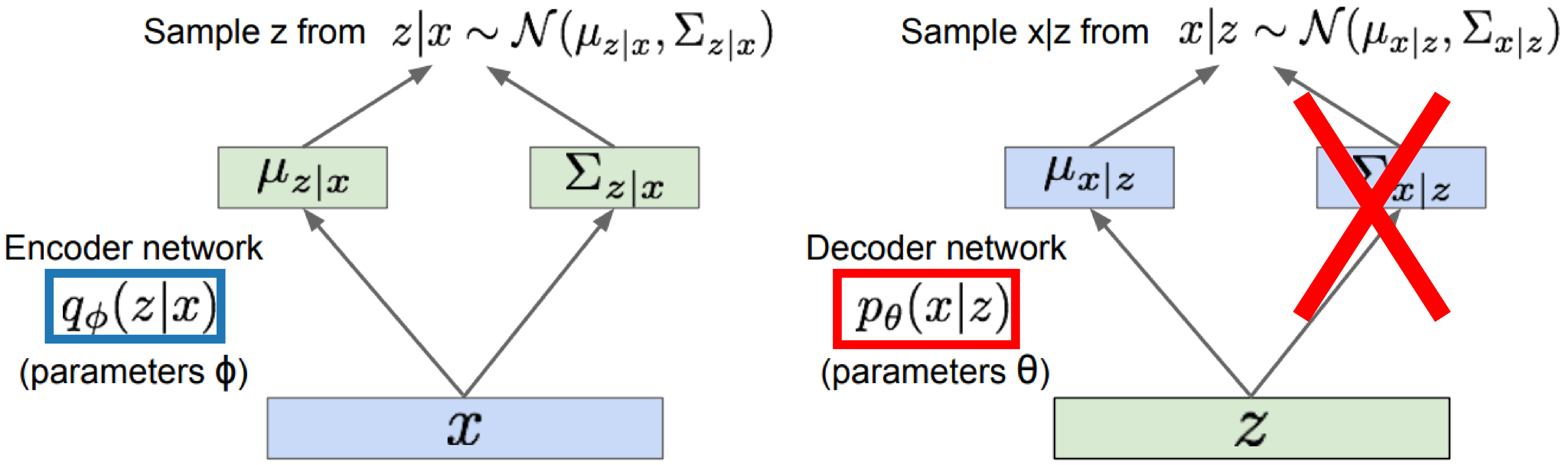


Probabilistic Interpretation

Goal: make $\log\left(p_\theta\left(x^{(i)}\right)\right)$ as high as possible

$$\mathcal{N}(0,1)$$

Using Bayes rule: $p_\theta(x) = \dfrac{\boxed{p_\theta(x|z)}\,p_\theta(z)}{\boxed{p_\theta(z|x)}}$



Probabilistic Interpretation

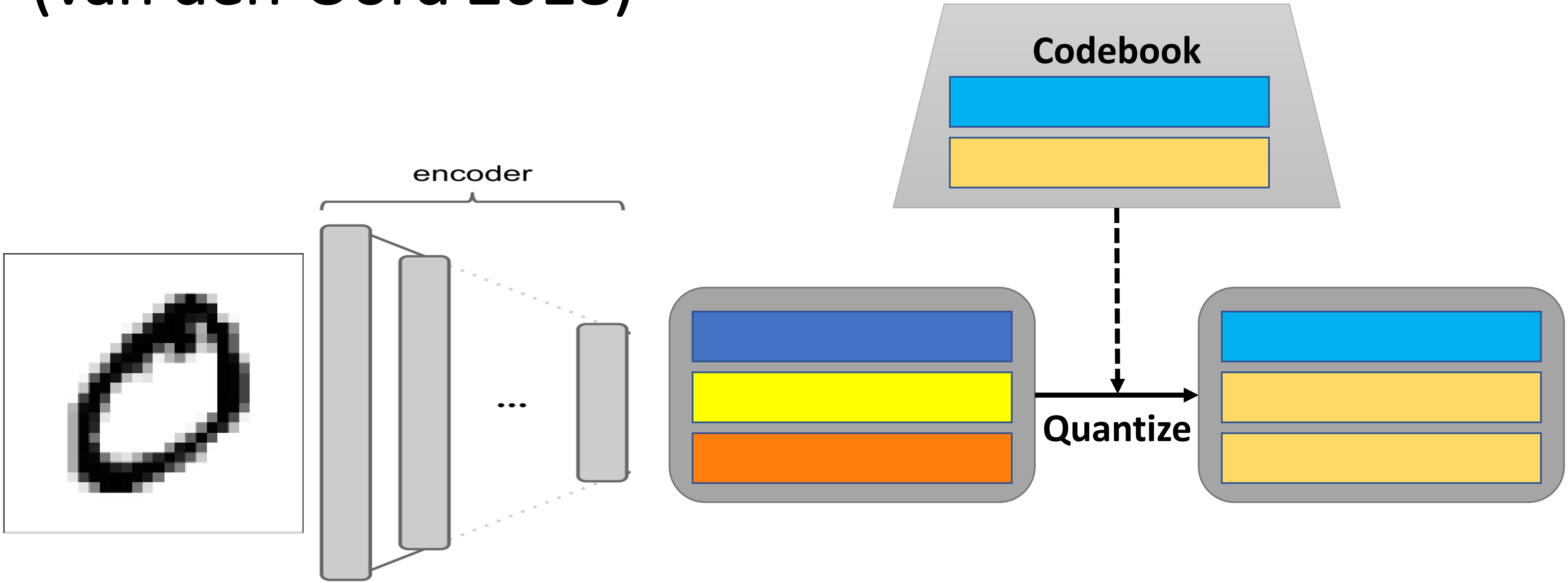
$$\log p_\theta(x^{(i)}) =$$

VAEs optimize a lower bound of the log likelihood

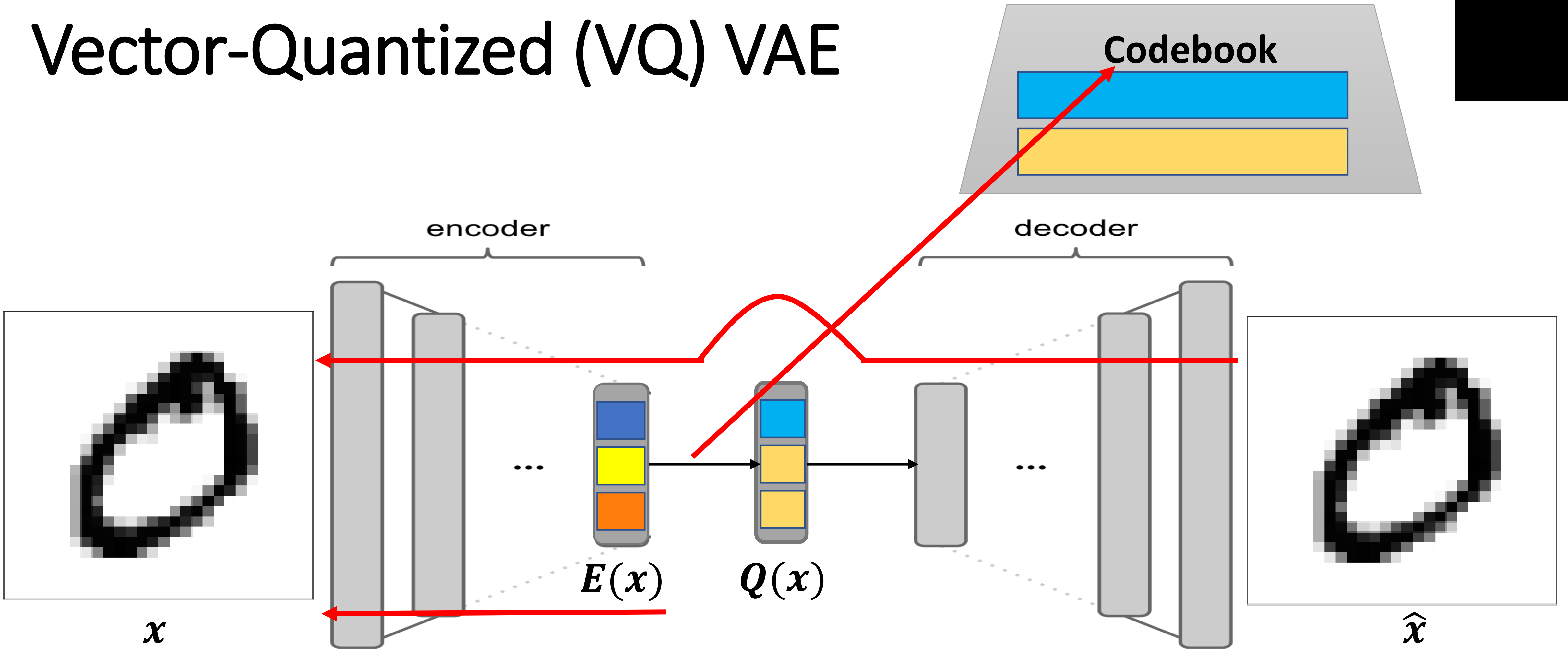
Slide credit: Stanford cs231n

Generate Data

Vector-Quantized (VQ) VAE (van den-Oord 2018)



Vector-Quantized (VQ) VAE

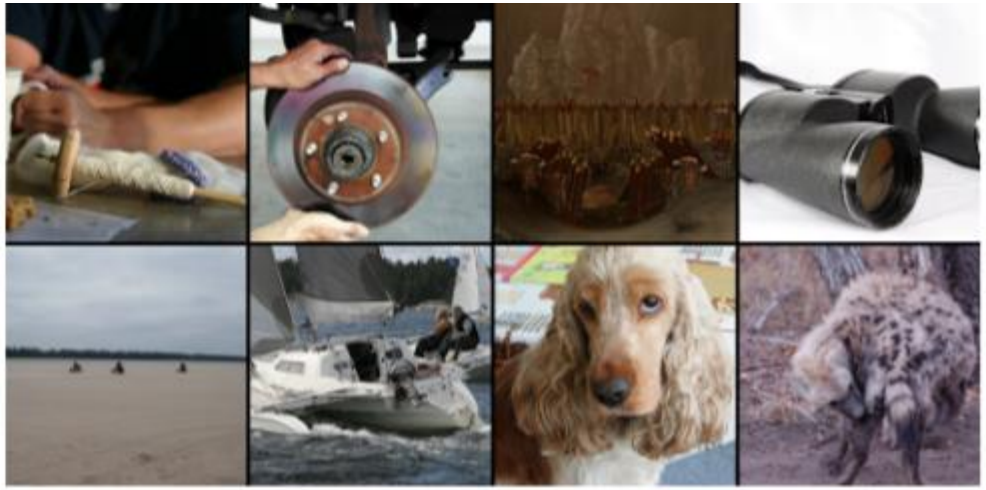


$$\mathcal{L}_{rec} = \|\hat{x} - x\|_2^2 \qquad \mathcal{L}_{commit} = \|E(x) - sg(Q(x))\|_2^2 \qquad \mathcal{L}_{codebook} = \|sg(E(x)) - Q(x)\|_2^2$$

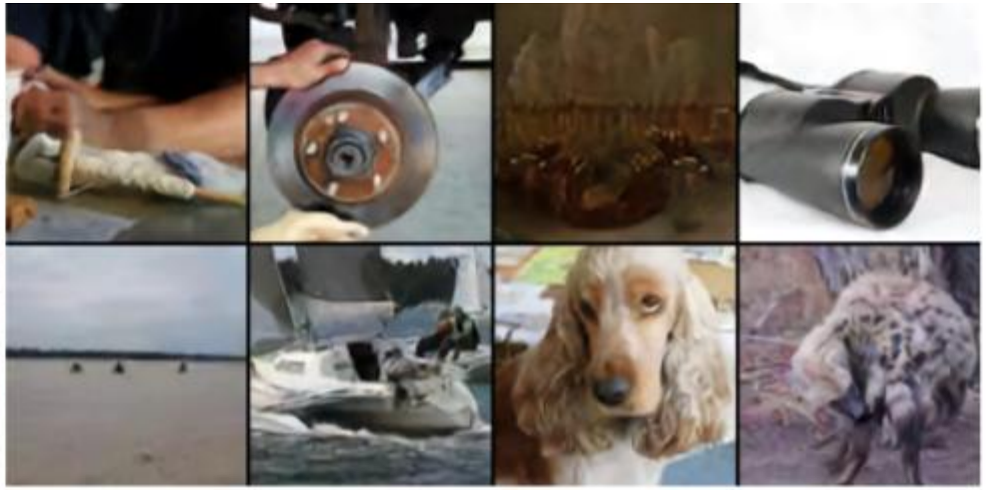
Quantization is non-differentiable!

VQ-VAE Reconstructions

Real

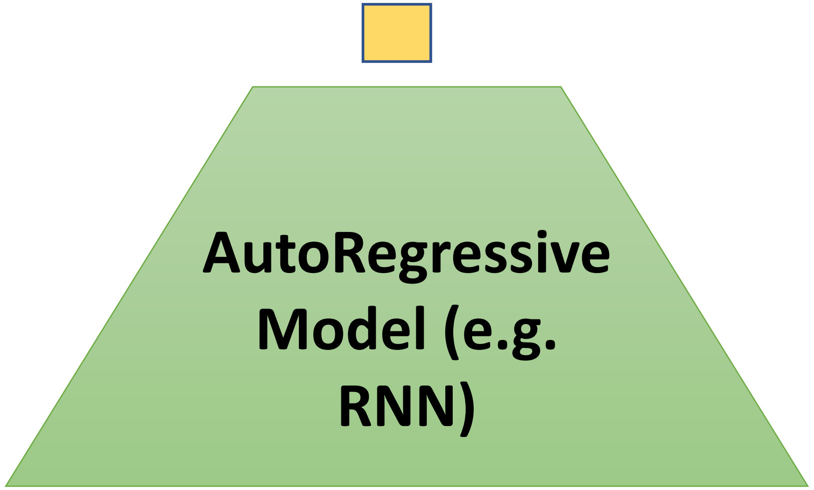


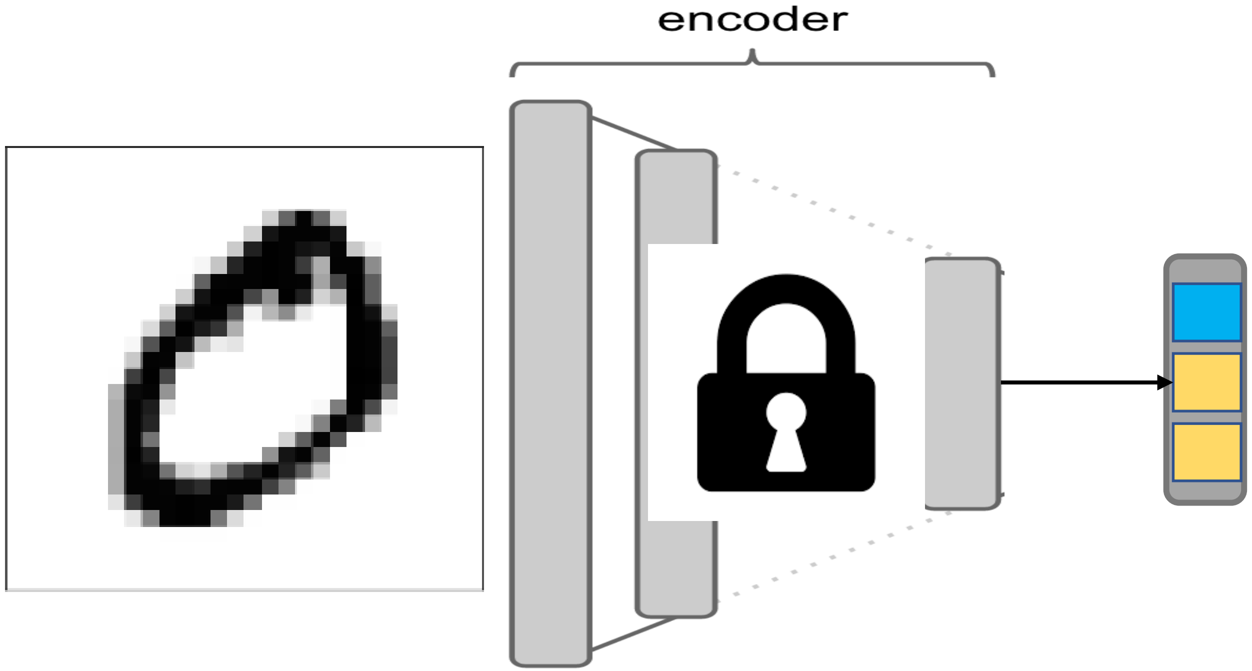
Reconstructed



Sampling New Instances

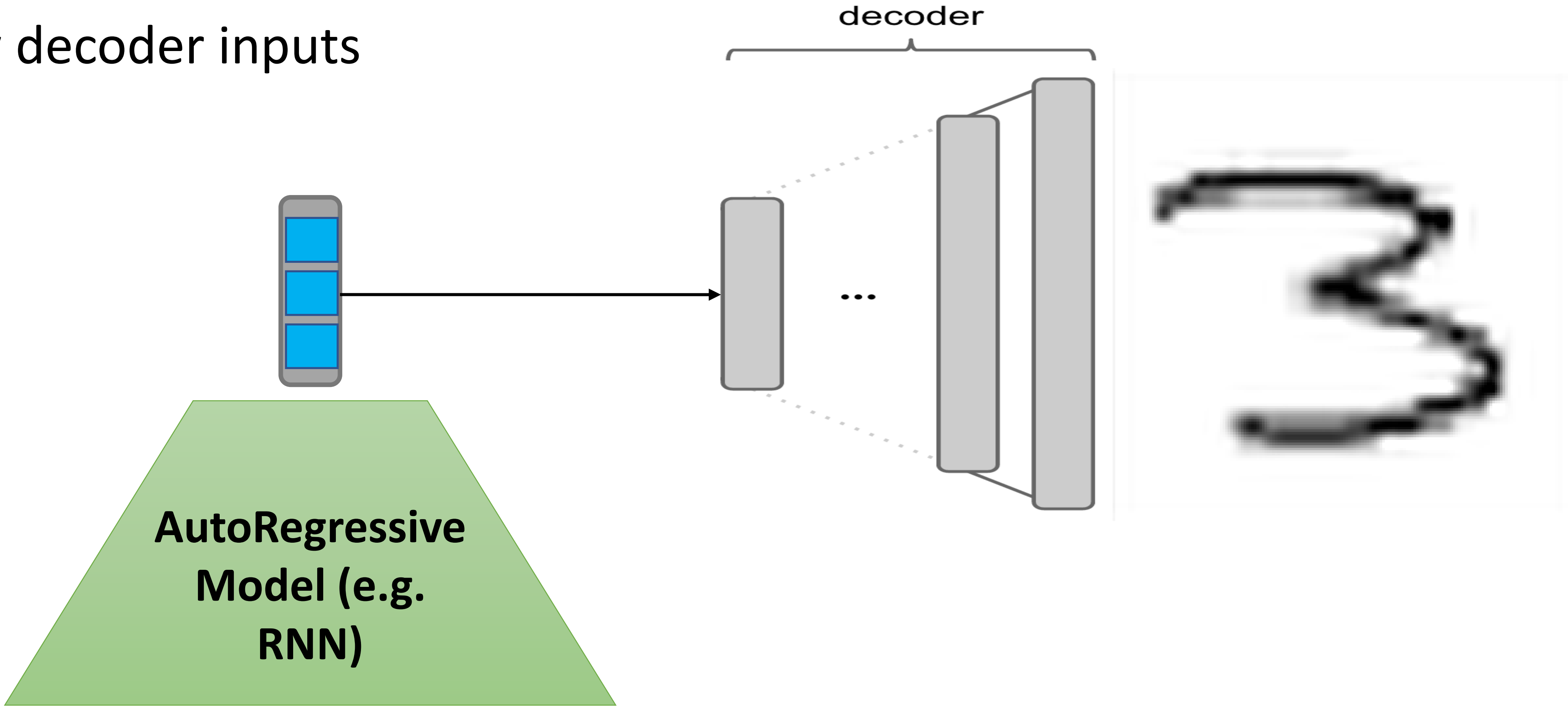
Training an autoregressive model to find $p(z)$





Sampling New Instances

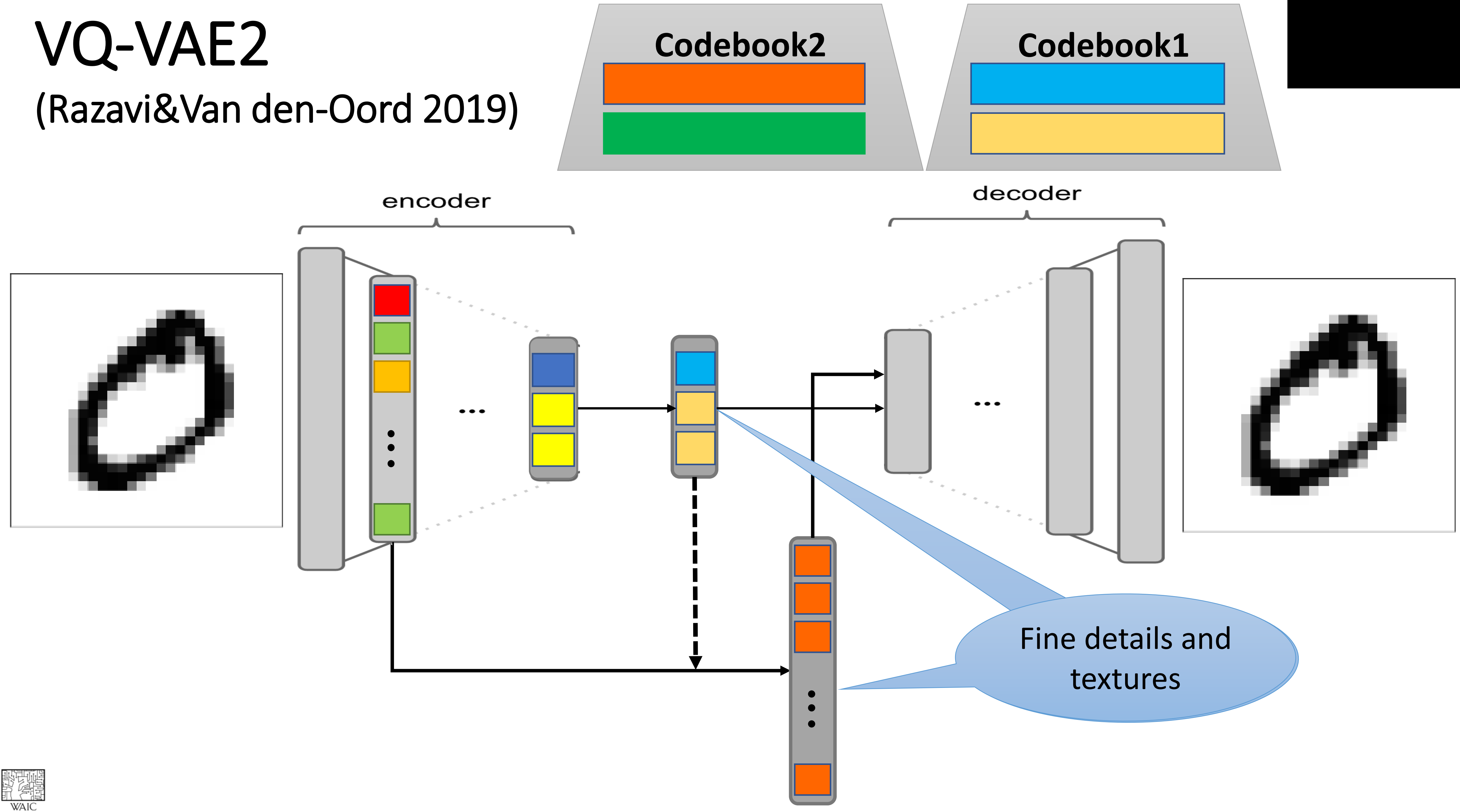
Once trained, use it to
generate new decoder inputs



Sampling New Instances - Results

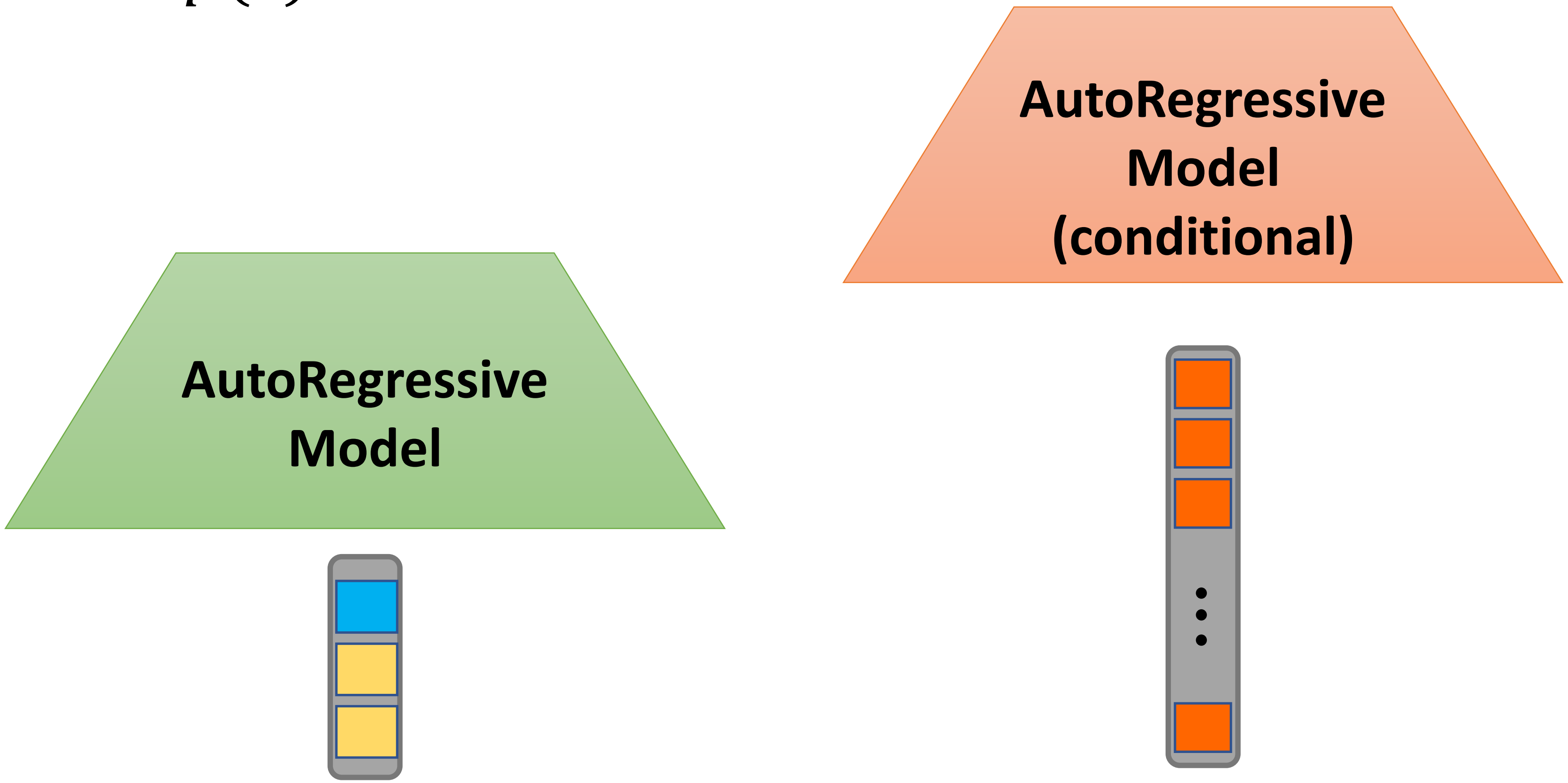
VQ-VAE2

(Razavi&Van den-Oord 2019)



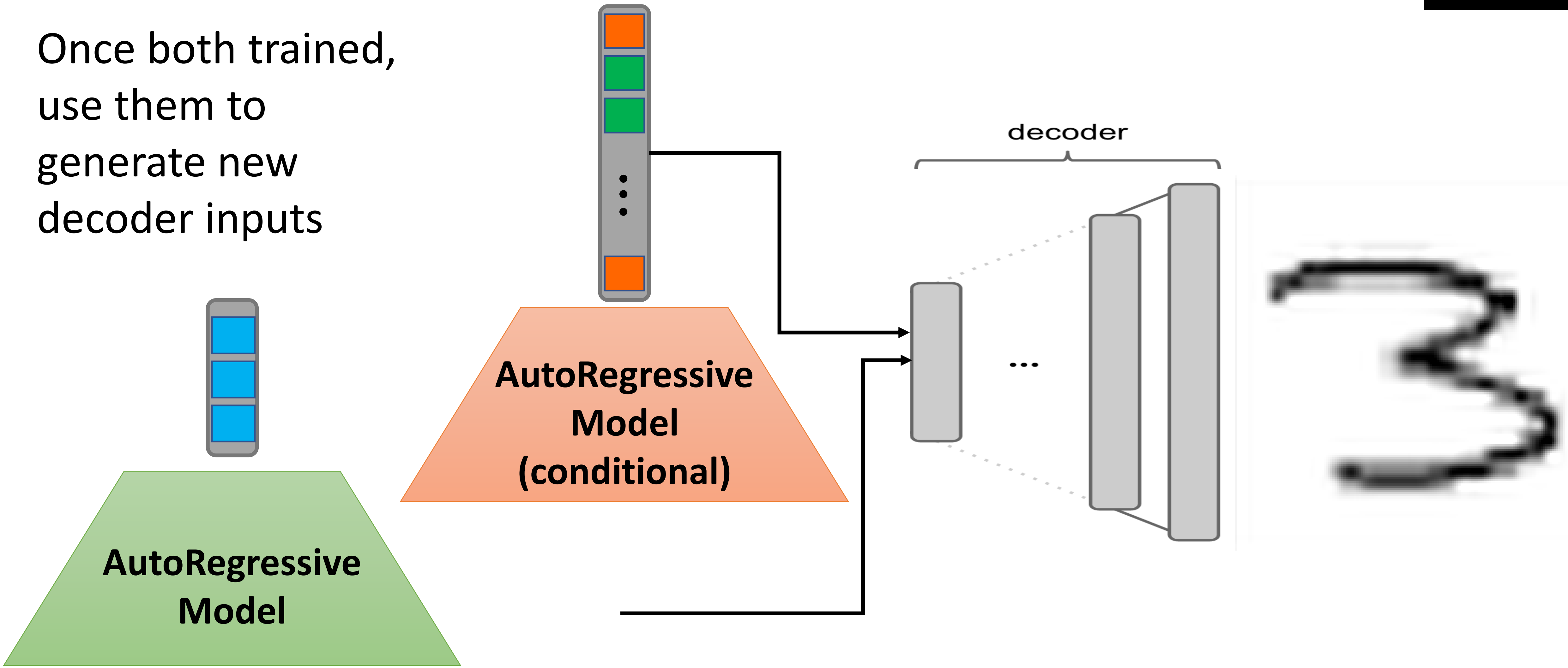
VQ-VAE2 – Sampling New Instances

Training two autoregressive models to find $p(z)$



VQ-VAE2 – Generation

Once both trained, use them to generate new decoder inputs



VQ-VAE2 – Results

WAIC

VQ-VAE2 – Results

(~)VQ-VAE + Transformers: DALL-E

Ramesh et al., 2021

Teapot in the shape of a rubik's cube

Soap-dispenser in the shape of a doughnut

Store front with 'pytorch'

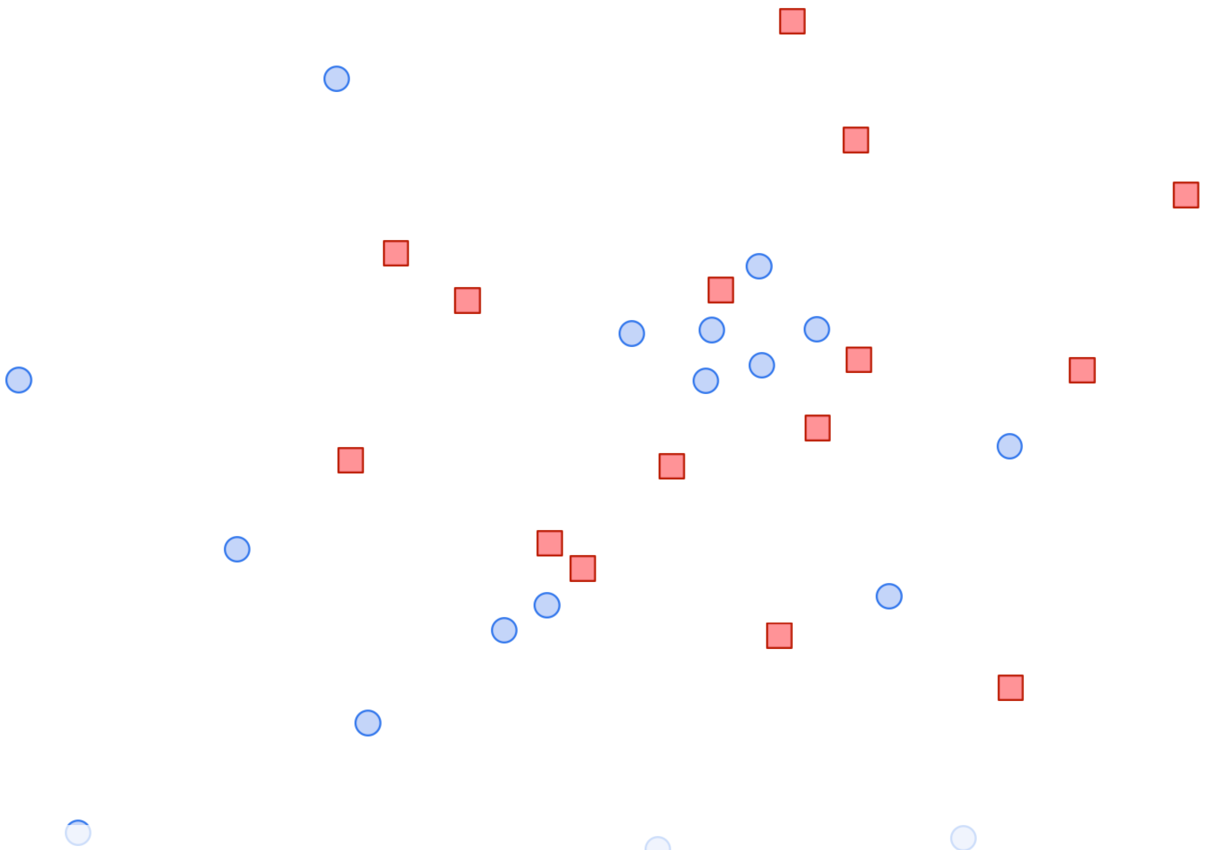
WAIC

Implicit Maximum Likelihood Estimation (IMLE)

Li&Malik 2018

- GAN: All **outputs** should fool the dis
- Prone to mode collapse

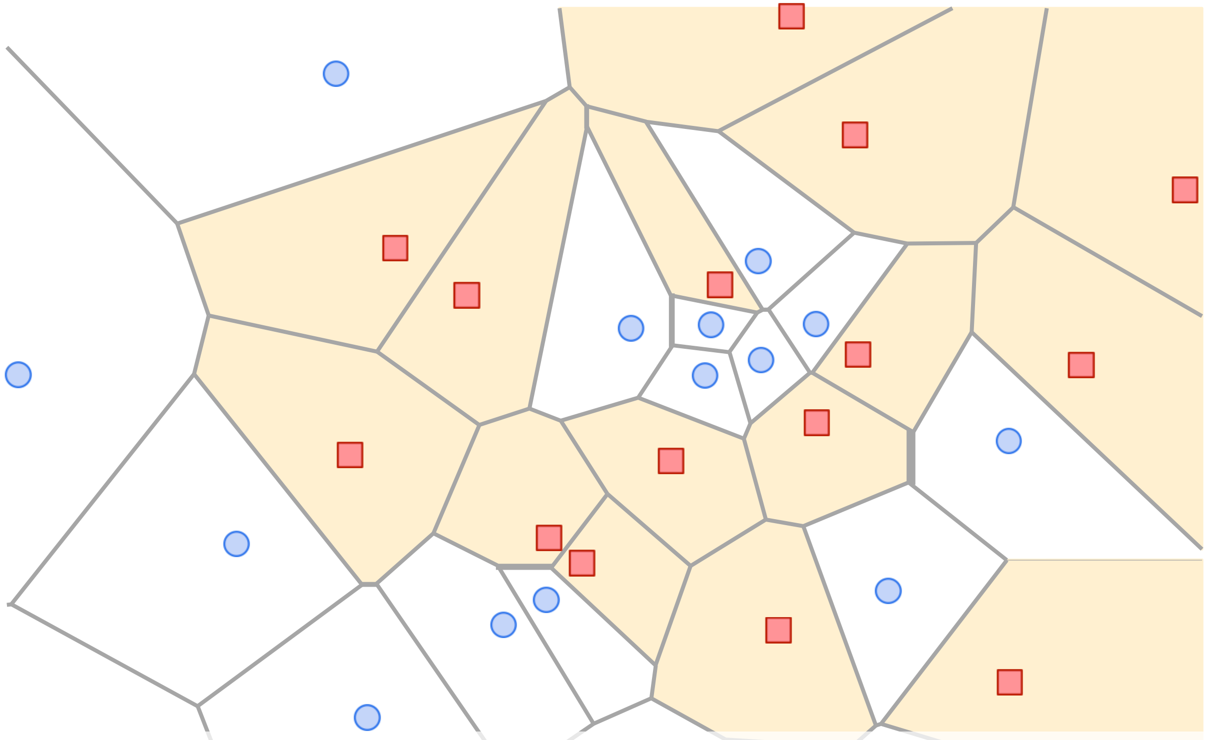
Generative Adversarial Nets



Squares are data examples; circles are samples.

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

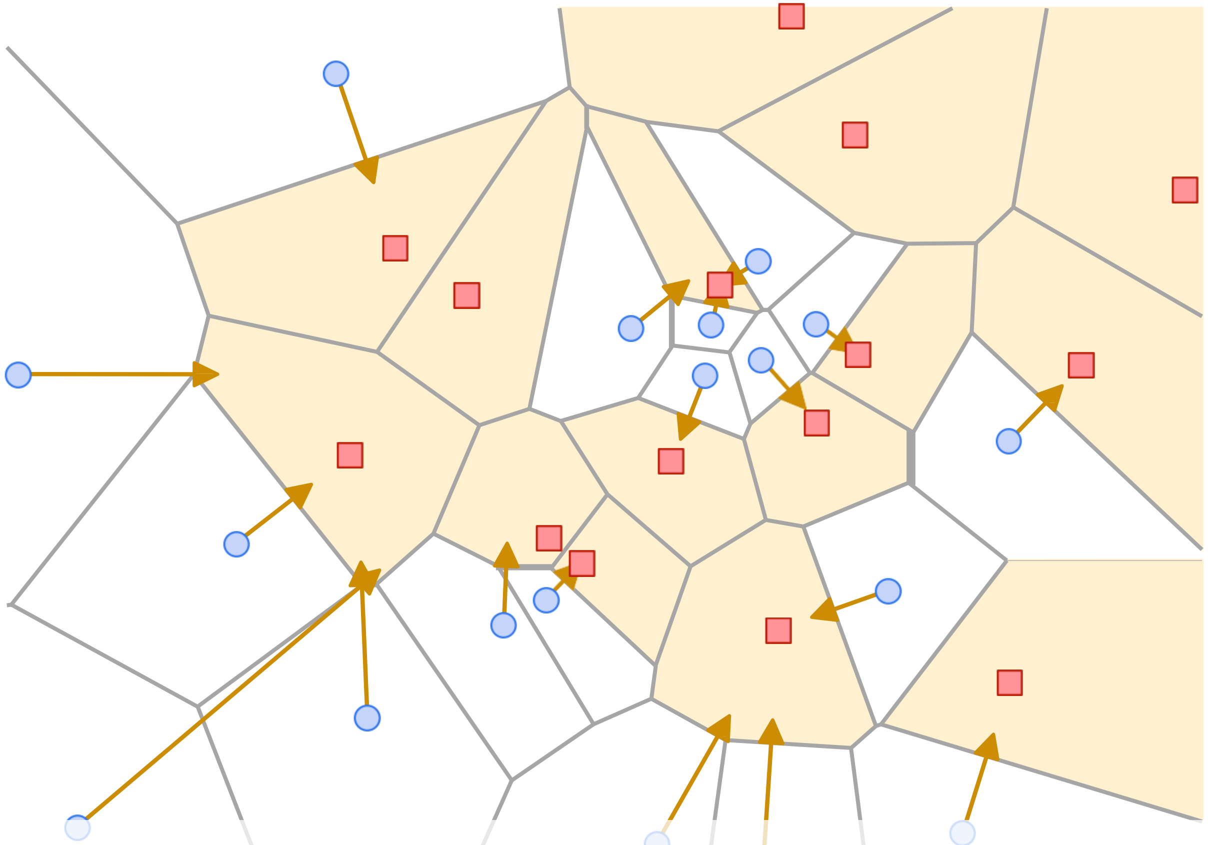
Generative Adversarial Nets



After training discriminator:
highlighted regions are classified as real.

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

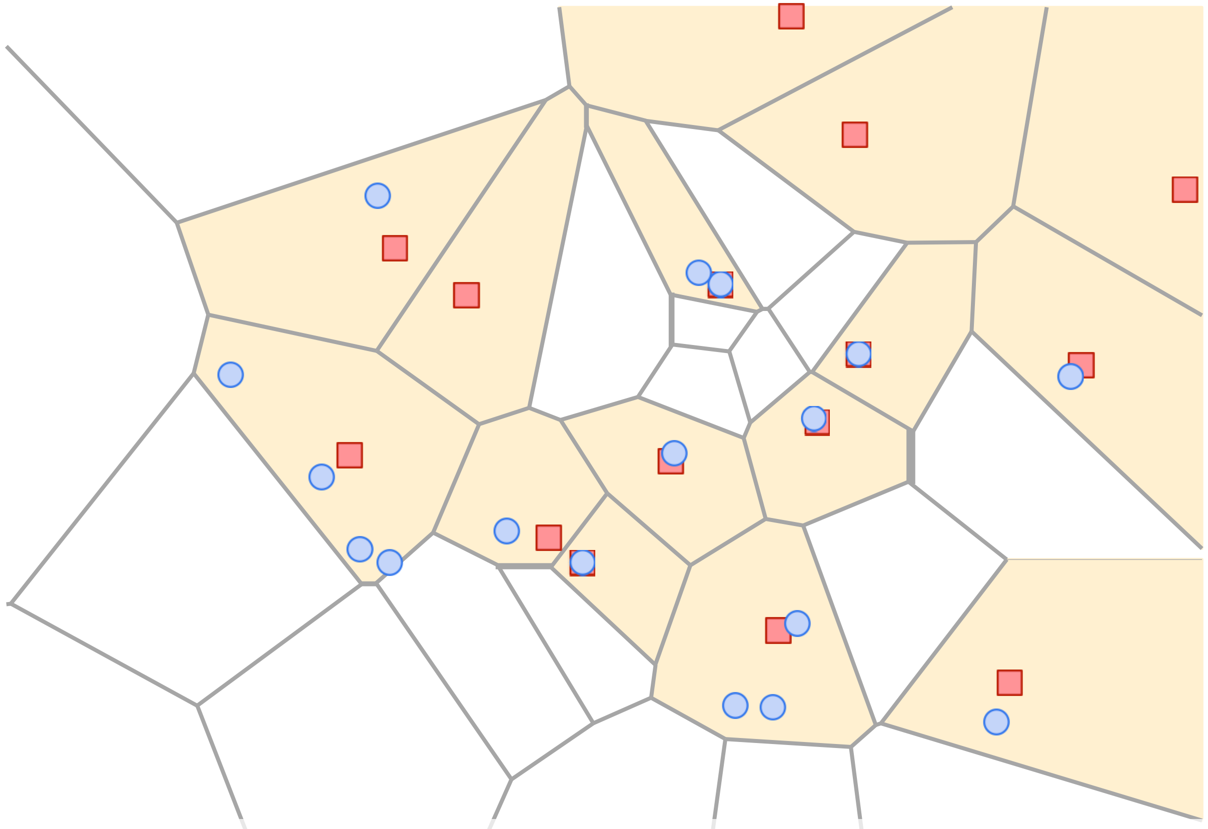
Generative Adversarial Nets



Training generator

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

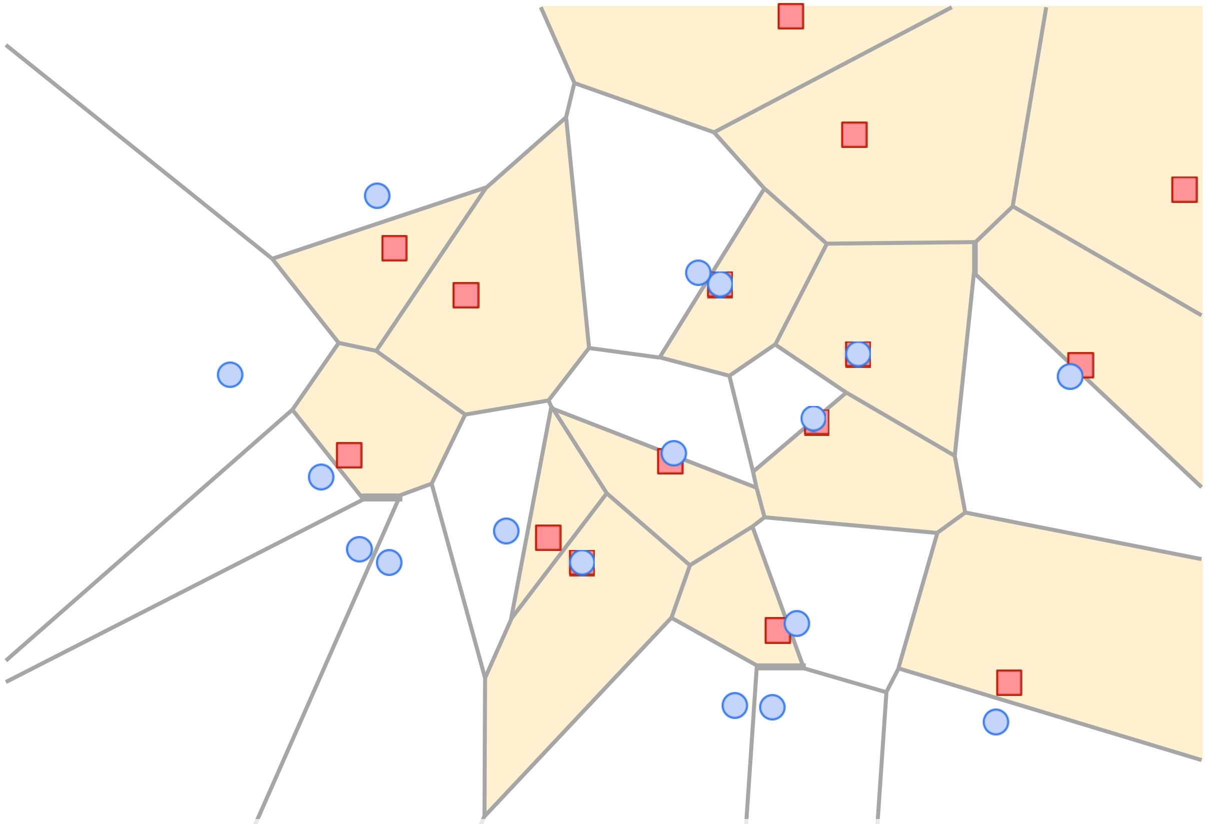
Generative Adversarial Nets



After training generator

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

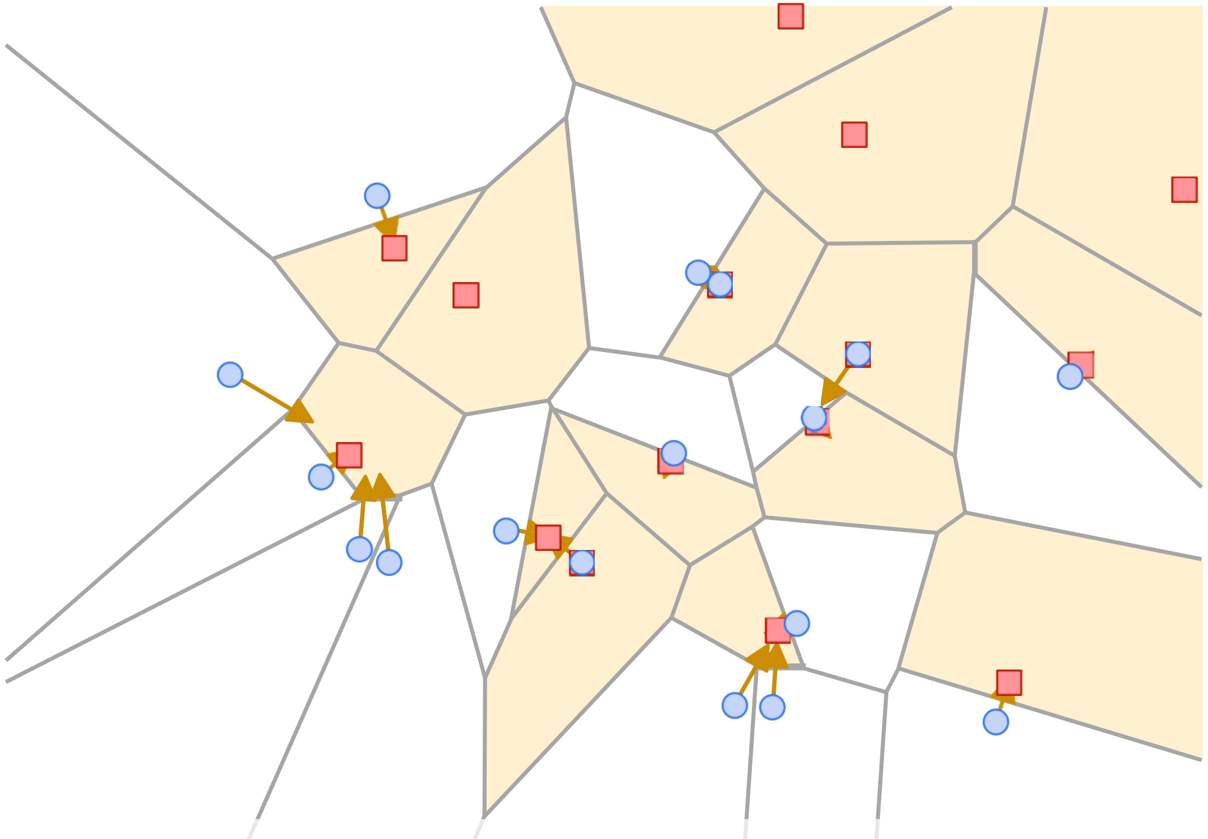
Generative Adversarial Nets



After training discriminator

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

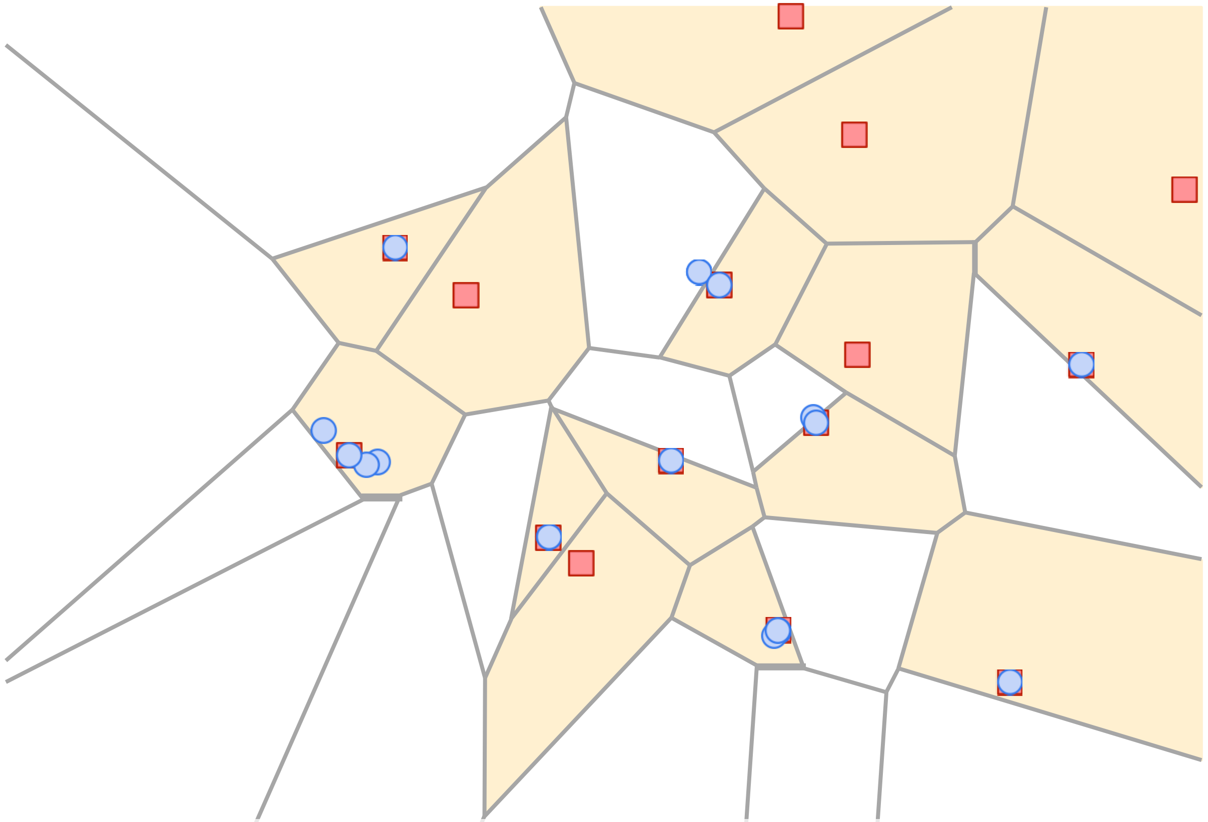
Generative Adversarial Nets



Training generator

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

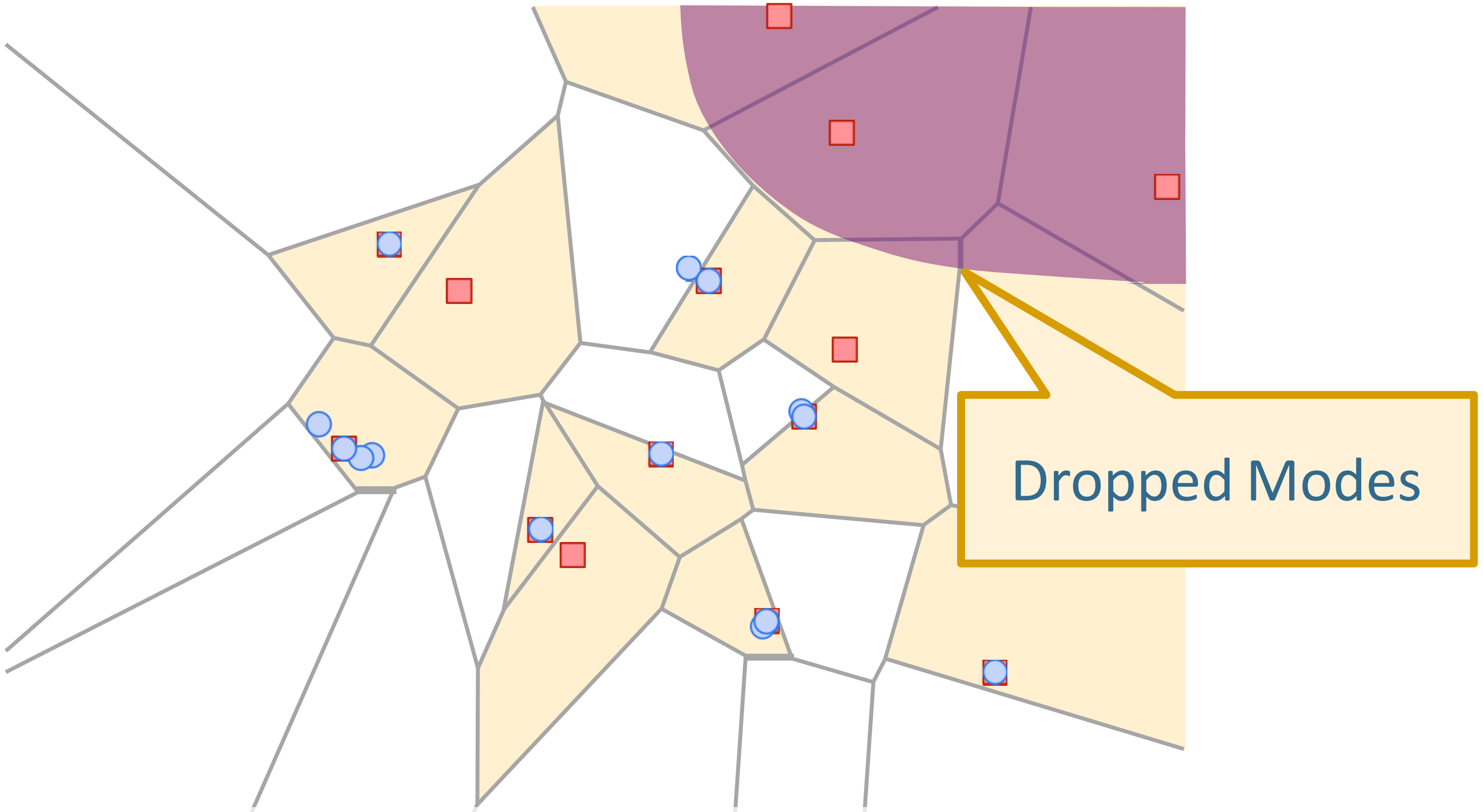
Generative Adversarial Nets



After training generator

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

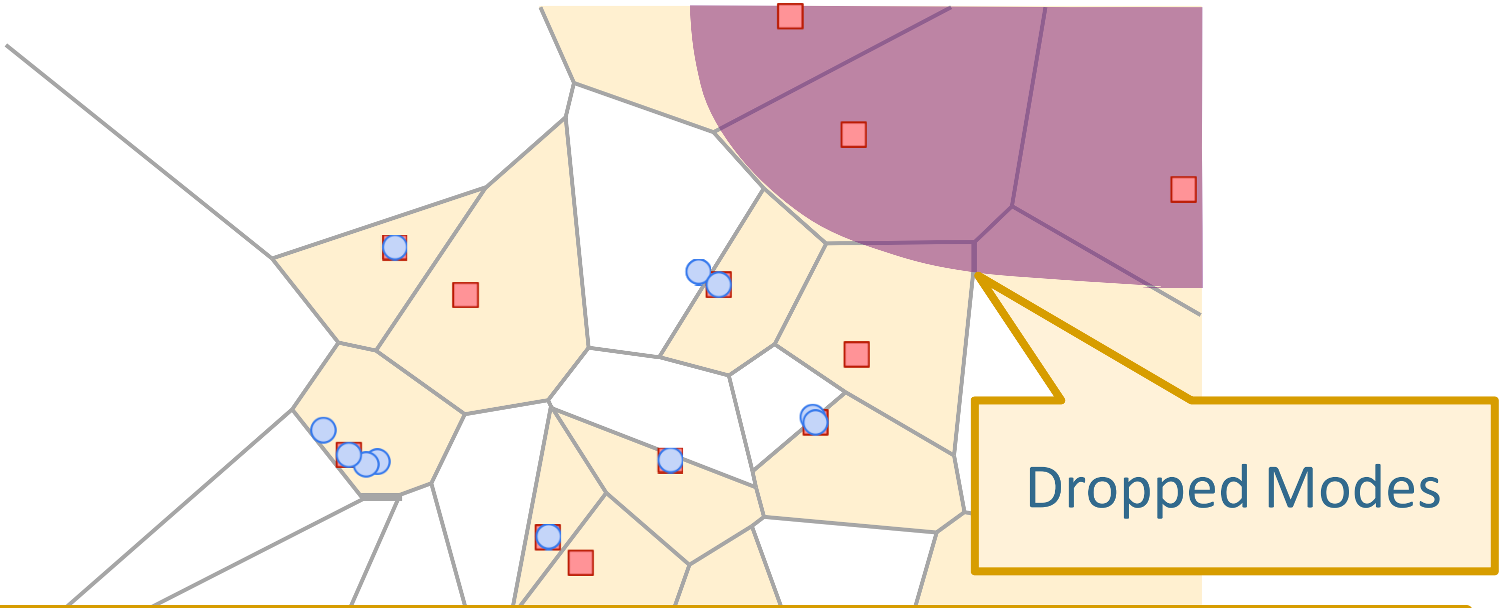
Generative Adversarial Nets



After training generator

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

Generative Adversarial Nets

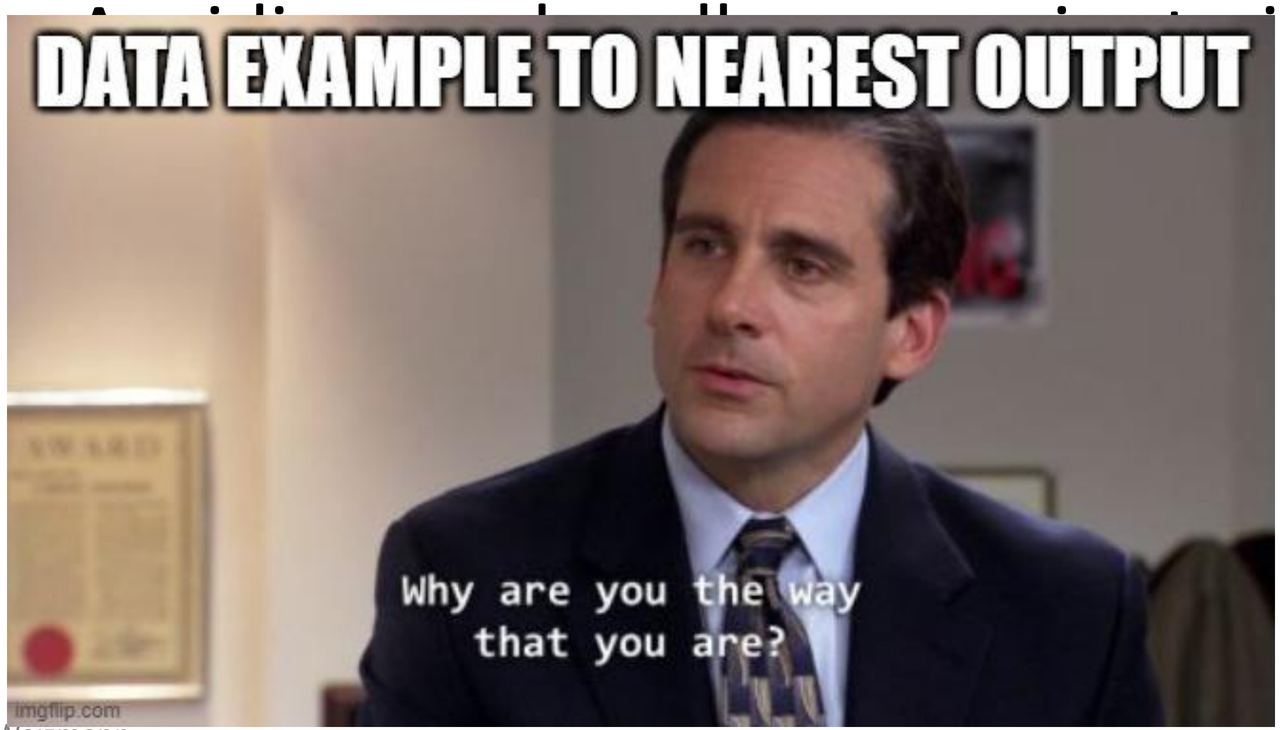


GANs only care about making each *sample* similar to some *data example*; it does not care about whether each *data example* is similar to some *sample*.

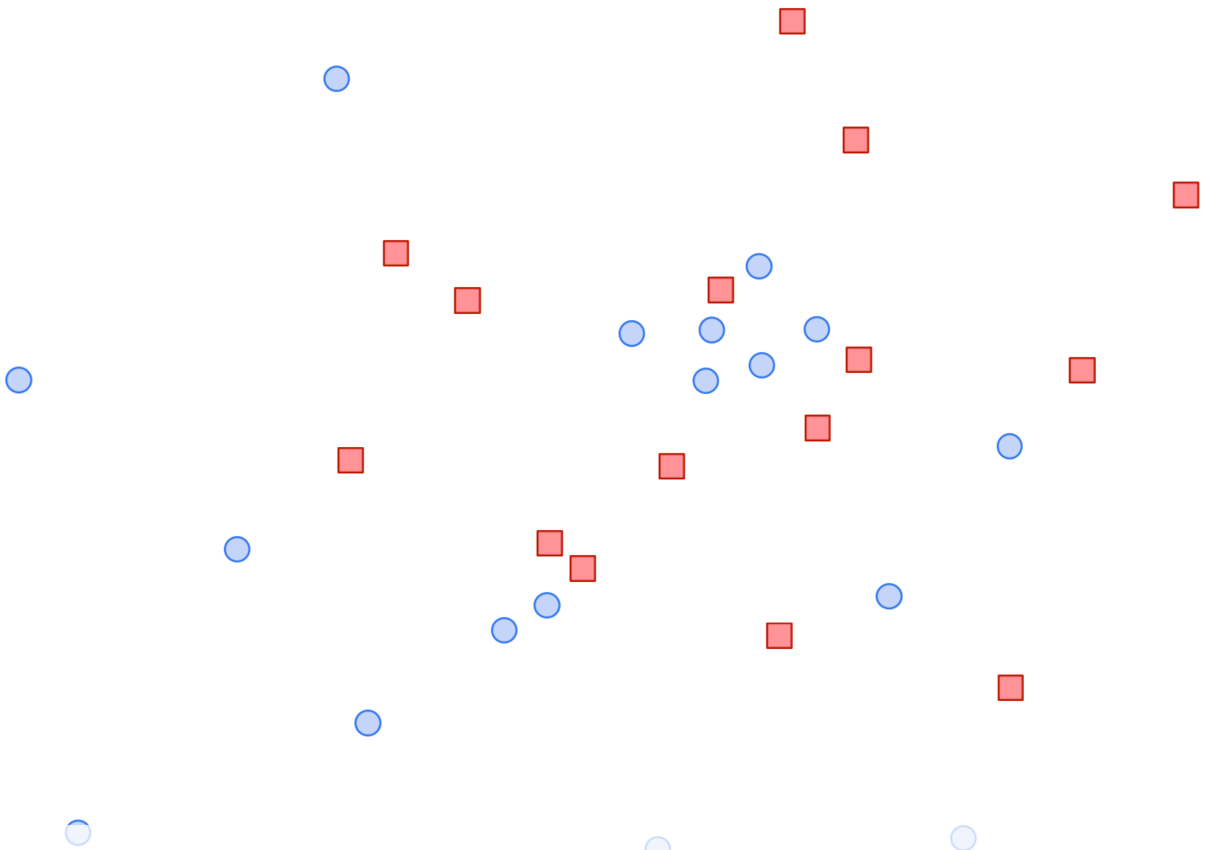
K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

Implicit Maximum Likelihood Estimation (IMLE)
Li&Malik 2018

- GAN: All **outputs** should fool the discriminator
    - Close to **some** data point!
- IMLE: All **data points** should have close outputs
- ining



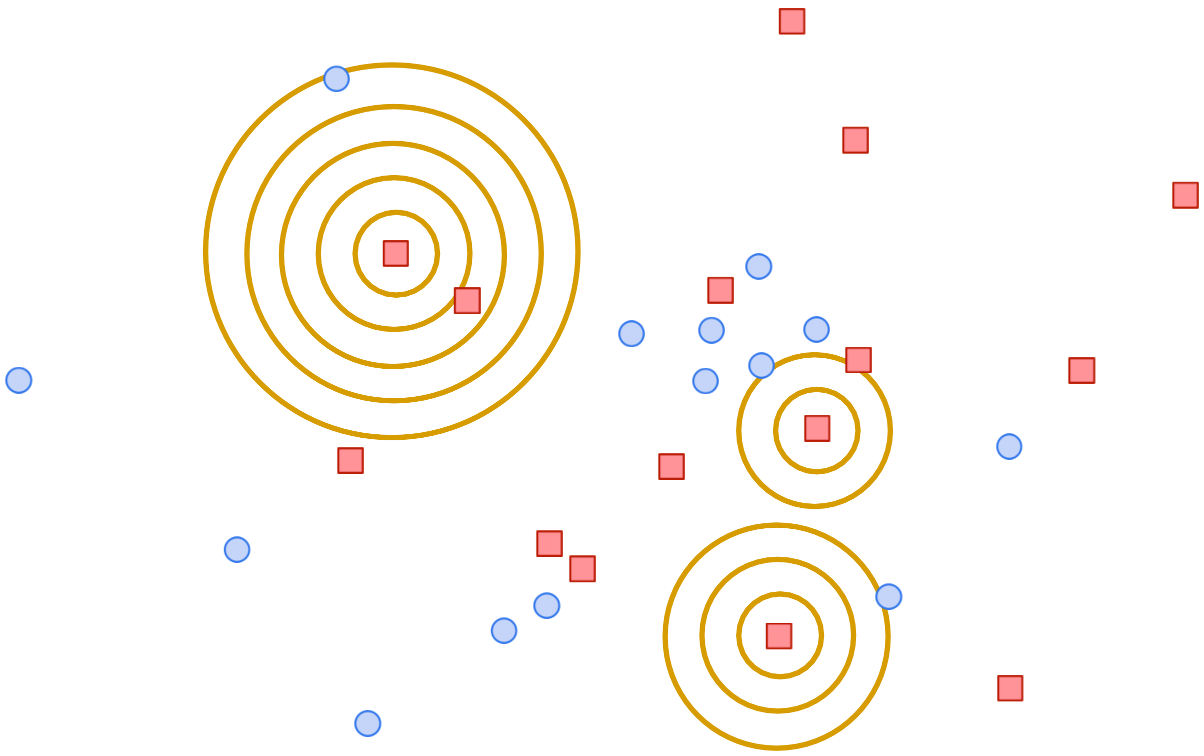
IMLE - Method



Squares are data examples; circles are samples.

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

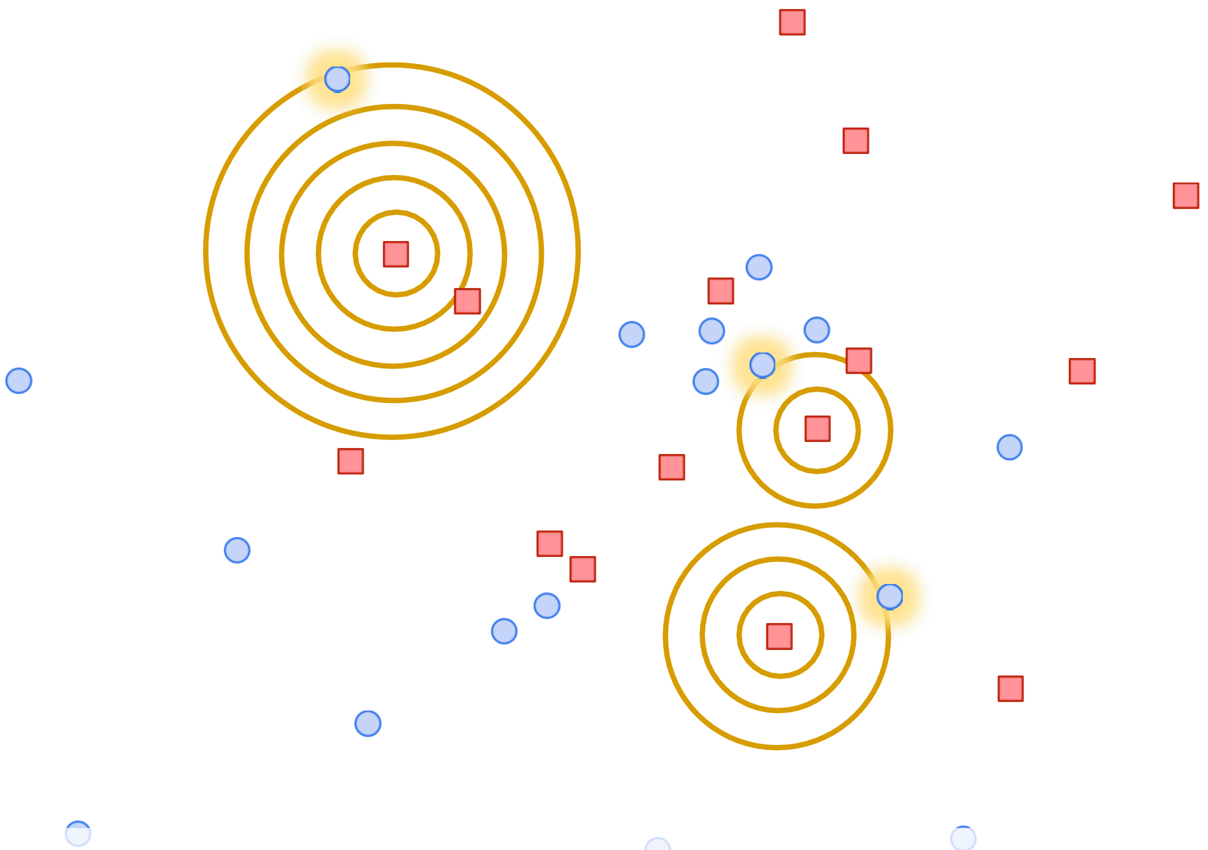
IMLE - Method



Find the nearest sample to each data example.

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

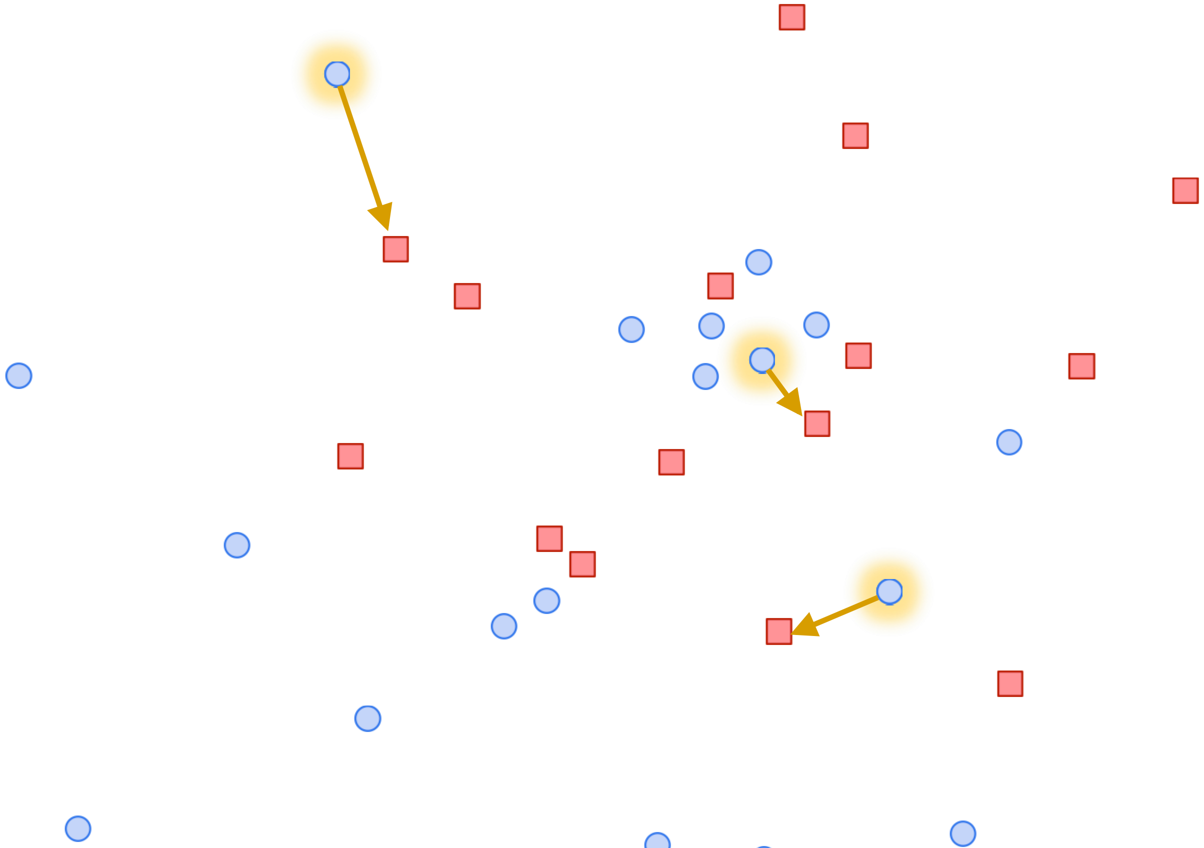
IMLE - Method



Nearest samples are found.

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

IMLE - Method



Pull sample towards data example.

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv: 1809.09087*, 2018

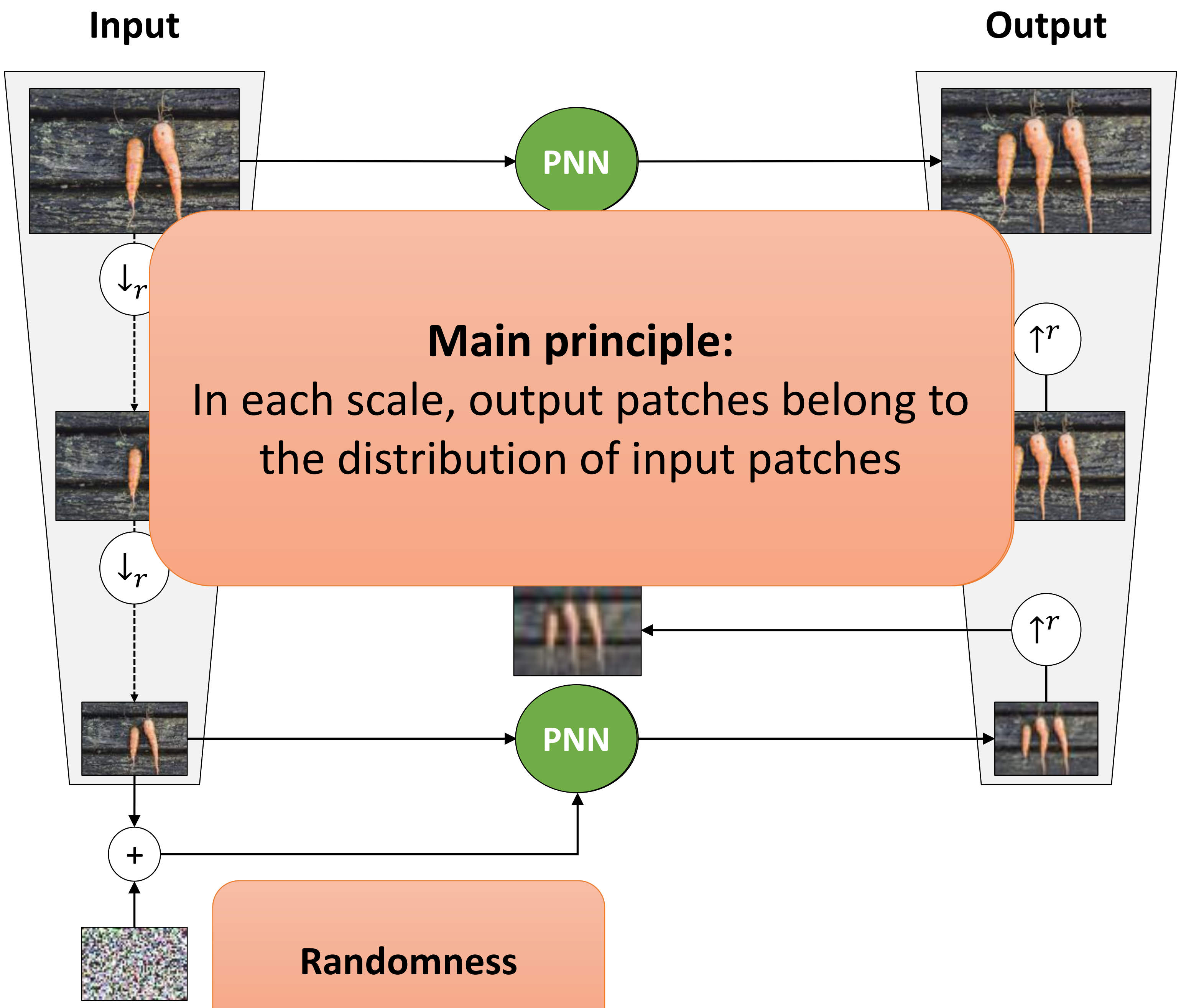
IMLE - Results

Summary

Alternatives to GANs (overcoming some of the problems):

- Autoencoders
- Variational Autoencoders
- VQ-VAE
- IMLE

GPNN



Main principle:
In each scale, output patches belong to the distribution of input patches

Randomness

Diverse images generated from a single image

Source Image | **Ours** | SinGAN







QUESTIONS?
imgflip.com