Introduction to Generative Models

Agenda

Today: Basics, Jan 2nd: Advanced

1. Goal, motivation, Basic methods
   1. Parametric methods
   2. Autoregressive methods
   3. Latent space mapping
2. Variational Auto Encoder (VAE)
3. Generative Adversarial Networks (GAN)
   1. Introduction (basic setup, intuition)
   2. Evaluation
   3. Image to image (pix2pix, ~~CycleGAN~~)
   4. Problems and how to improve GAN performance (losses, tricks etc.)
   5. StyleGAN
   6. Extras (GAN Dissection, Single Image)

A brain riding a rocketship heading towards the moon.

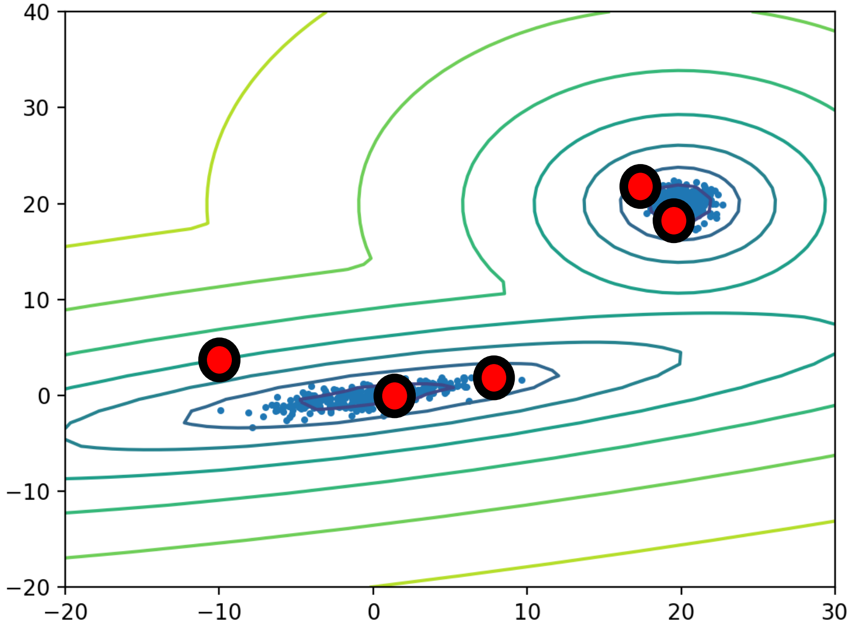
A dragon fruit wearing karate belt in the snow.

Generative methods:

- Parametric distribution estimation (e.g. GMM)
- Autoregressive models (e.g. RNN, Causal CNN, Diffusion)
- Latent space mapping (e.g. VAE, GAN)

Parametric Distribution Estimation Example: GMM

Step 1: observe a set of samples



Step 2: assume a GMM model

$$p(x|\theta) = \sum_i \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

Step 3: perform maximum likelihood learning

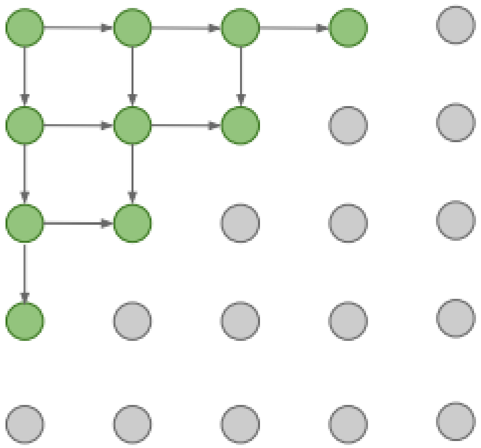
$$\max_\theta \sum_{x^{(j)} \in \mathrm{Dataset}} \log p(\theta|x^{(j)})$$

Step 4: Sample

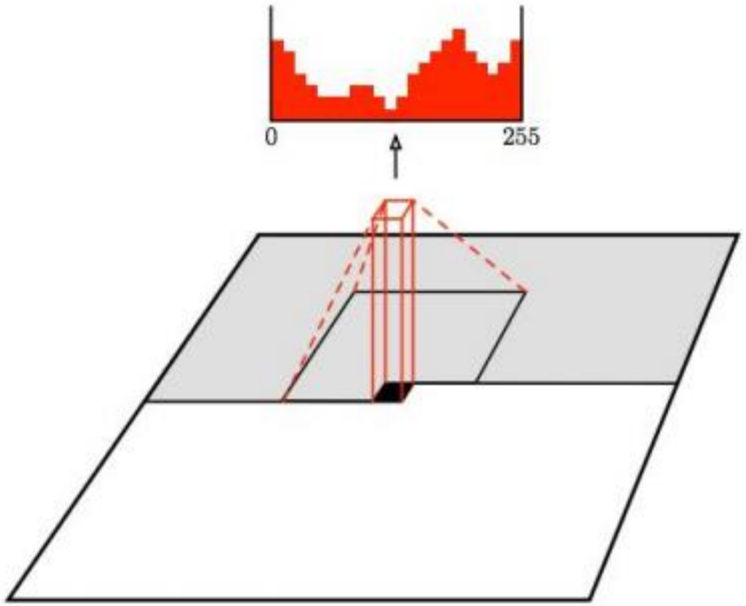
Elements in the slide taken from NVIDIA CVPR’18 GANs tutorial

Autoregressive image generation - Basic

PixelRNN



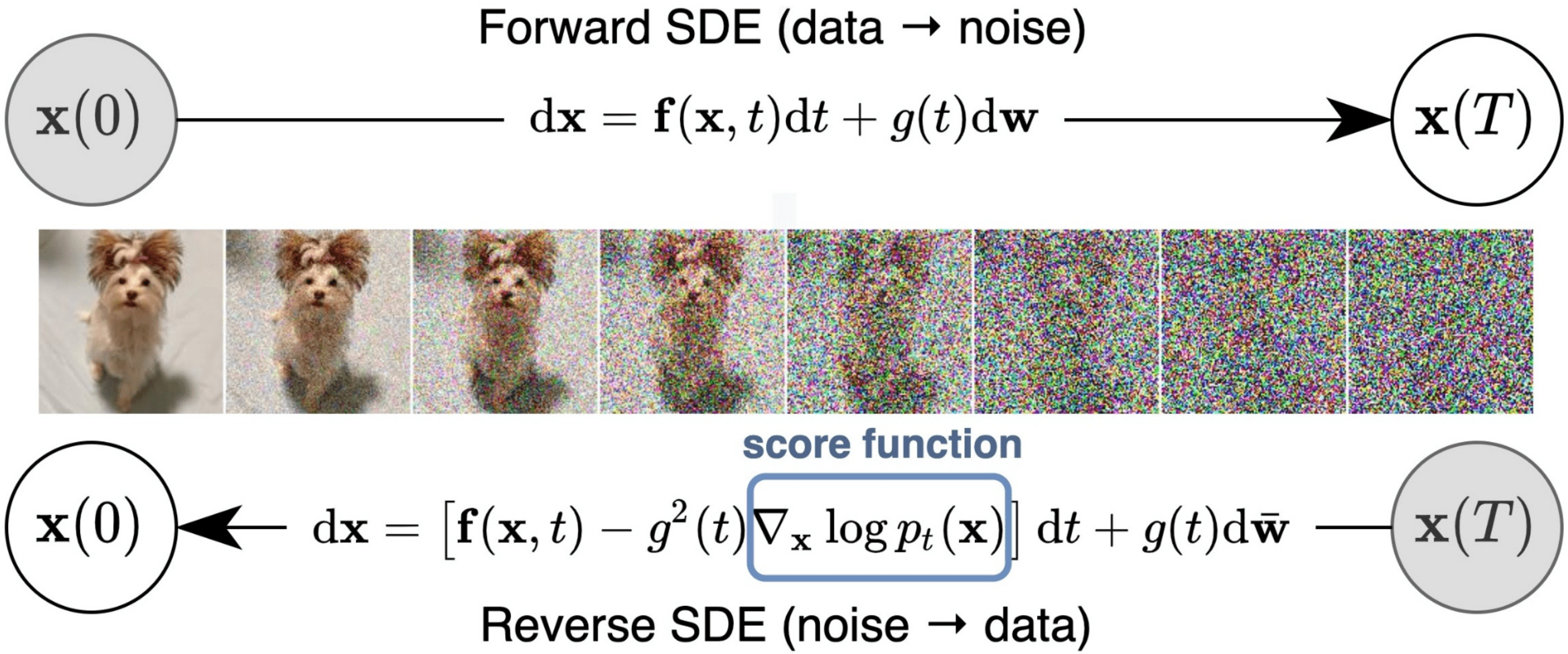
PixelCNN



Van der Oord 2016

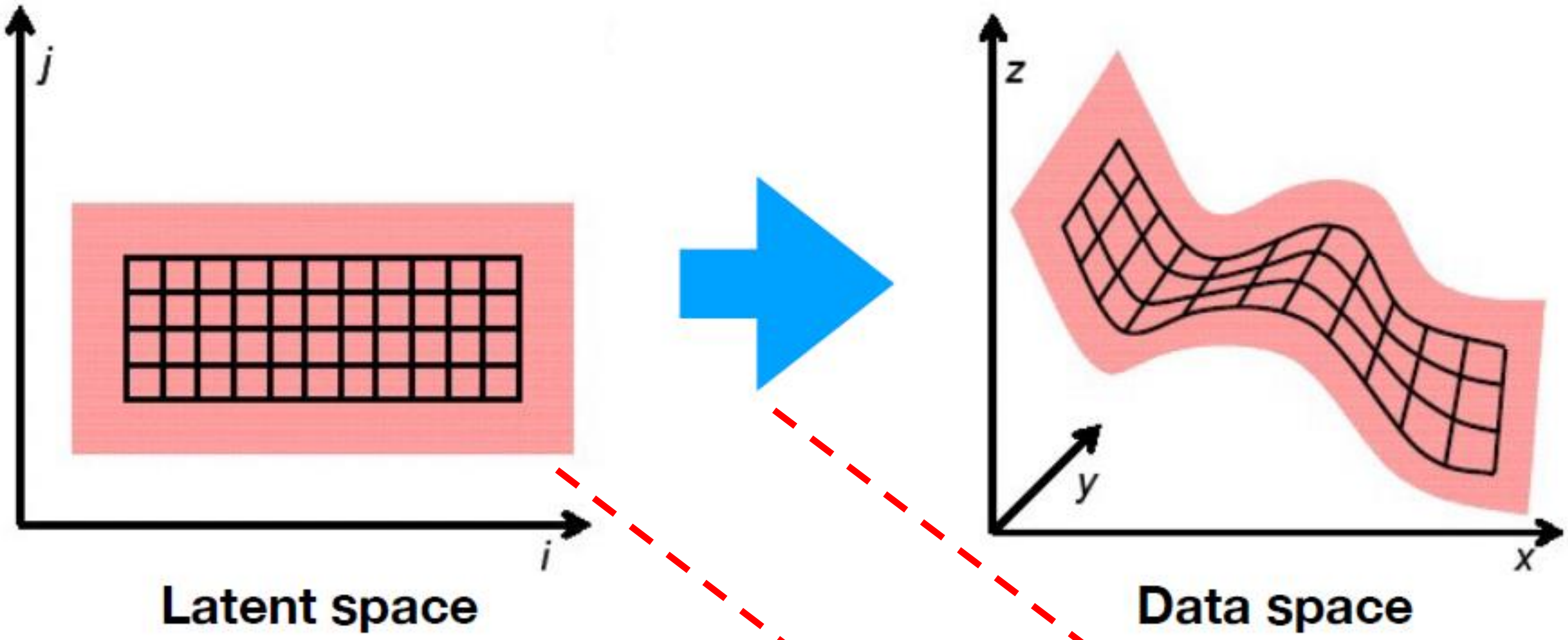
Autoregressive image generation - Recent

Diffusion models:



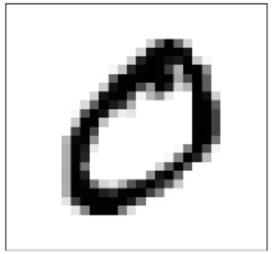
Wait for the advanced generative models class!

Latent space mapping approach

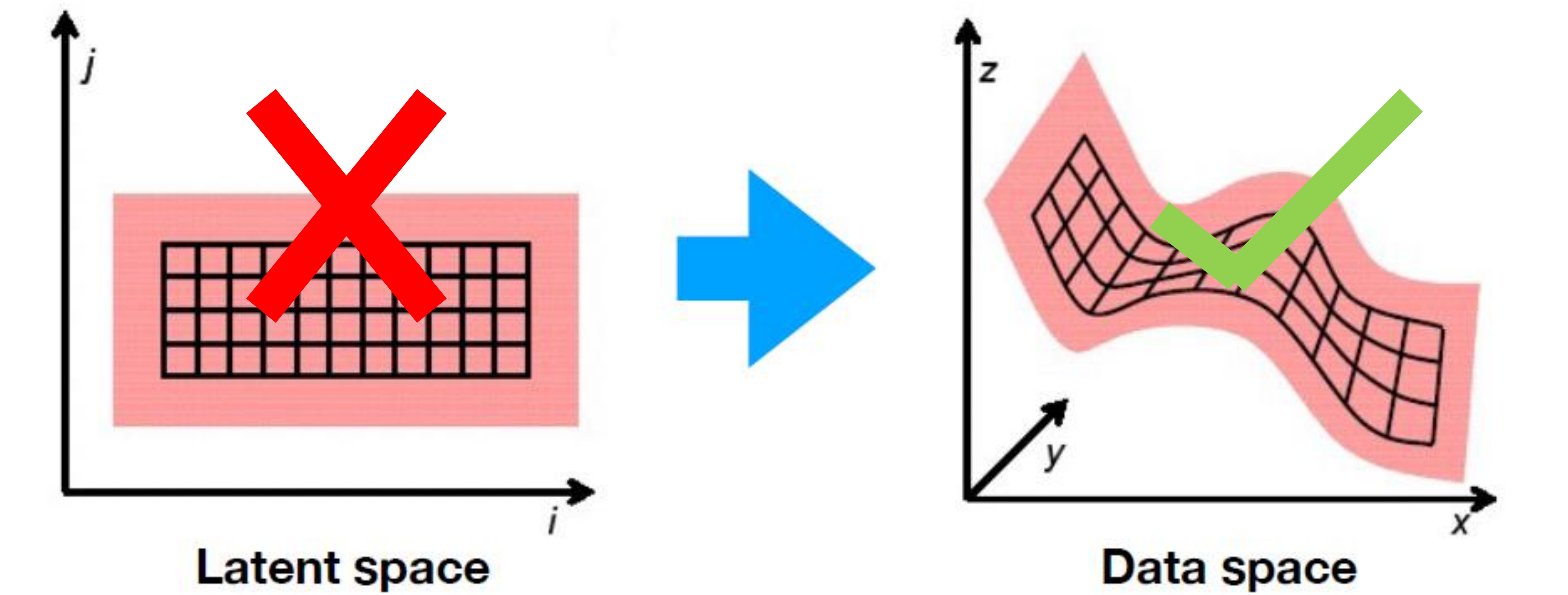


Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

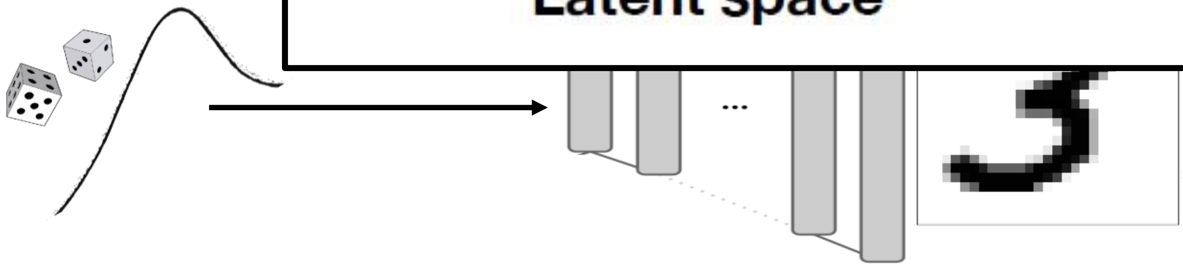
Autoencoders

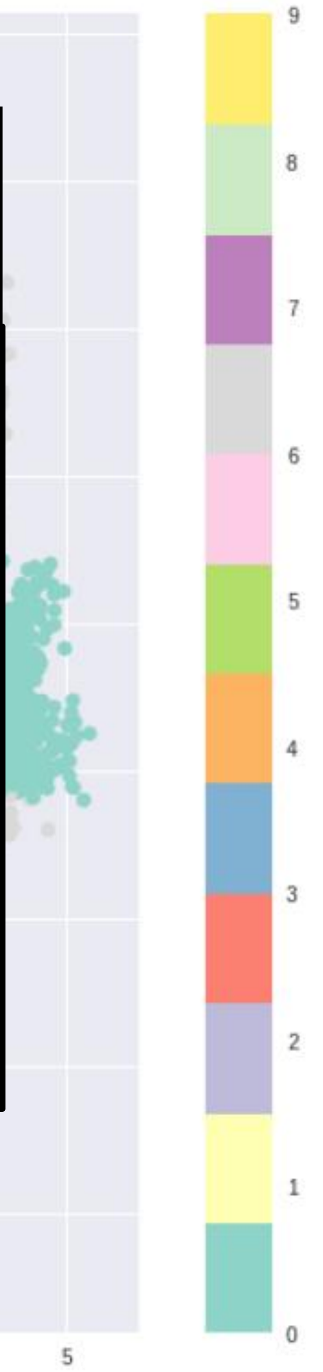


AE does not transform one *pre-determined* distribution to another!

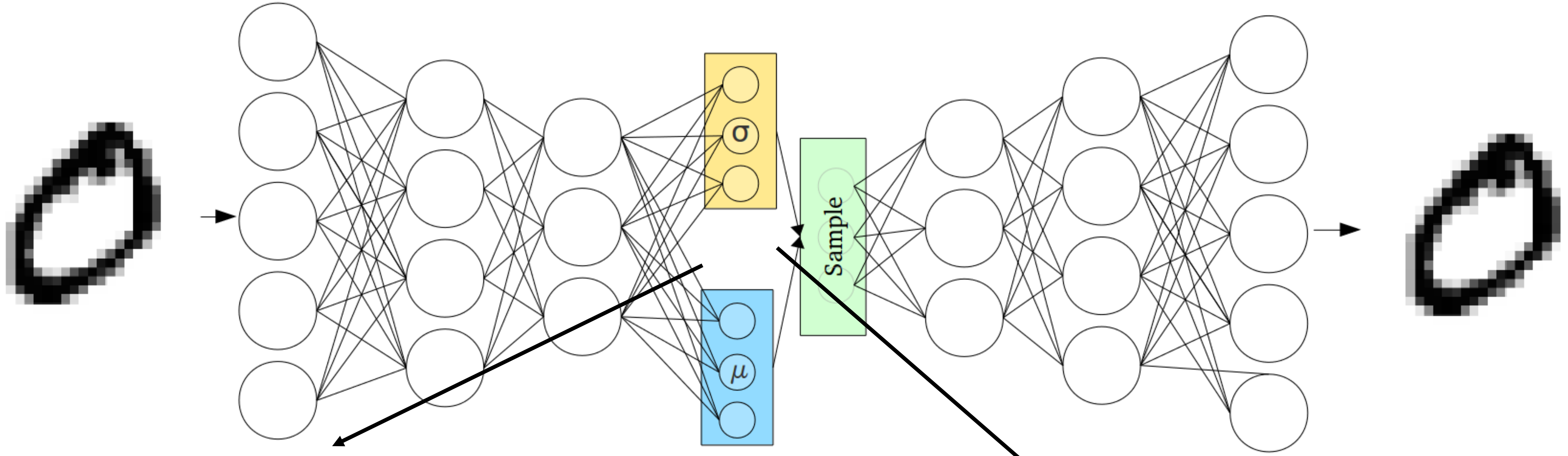


- Gene





Variational Autoencoders (Kingma&Welling 2014)

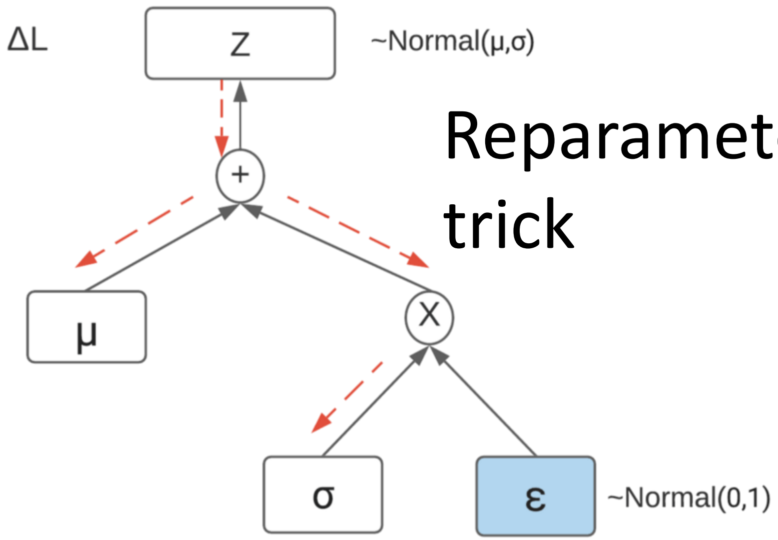


Regularization:

encourage $p(z) \sim \mathrm{N}(0,1)$

by KL divergence:

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

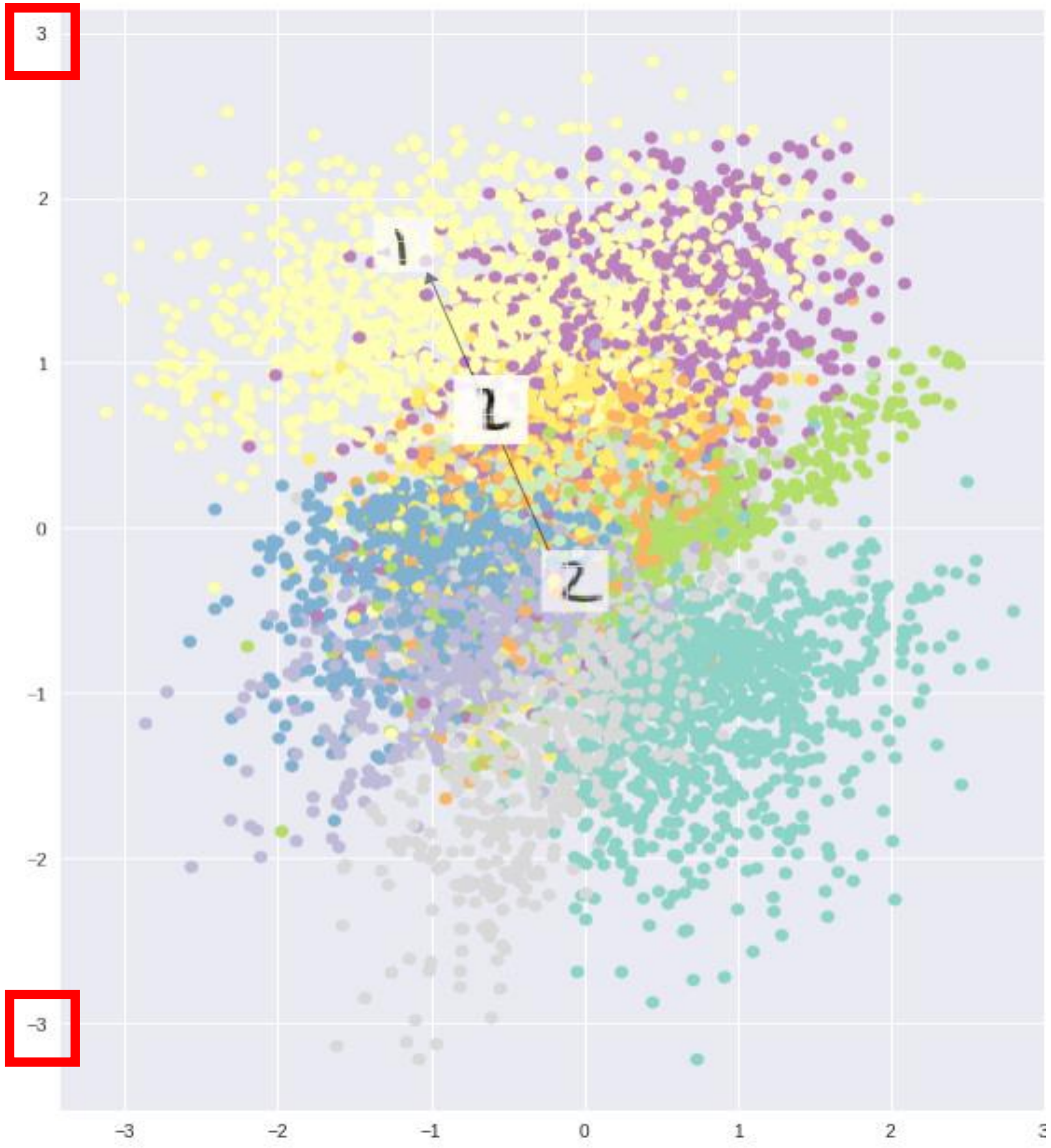


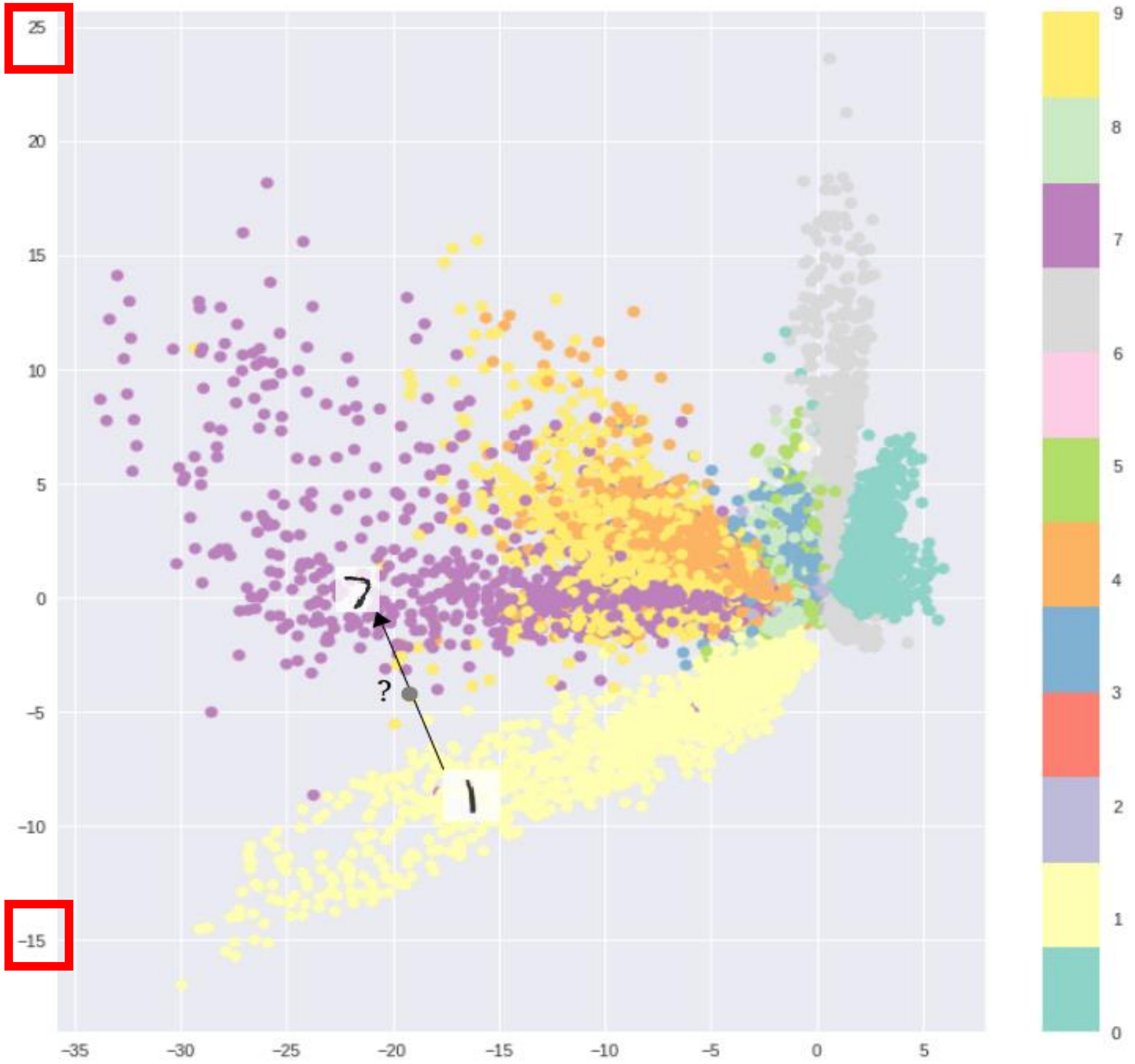
Reparameterization trick

VAE

AE

Also check out the scale!

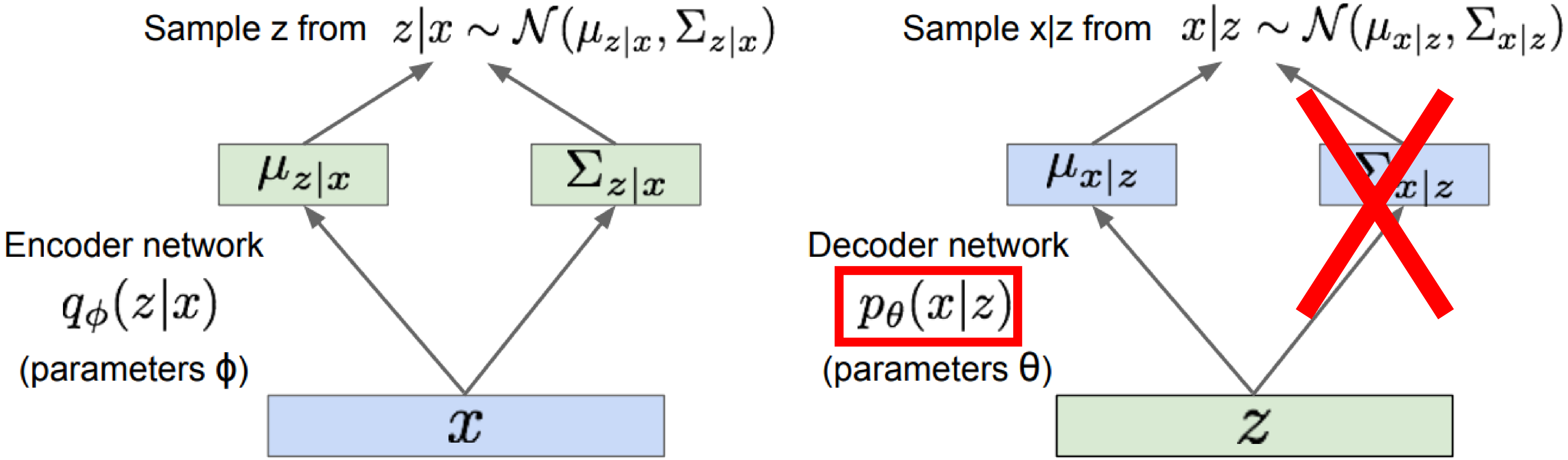




Probabilistic interpretation

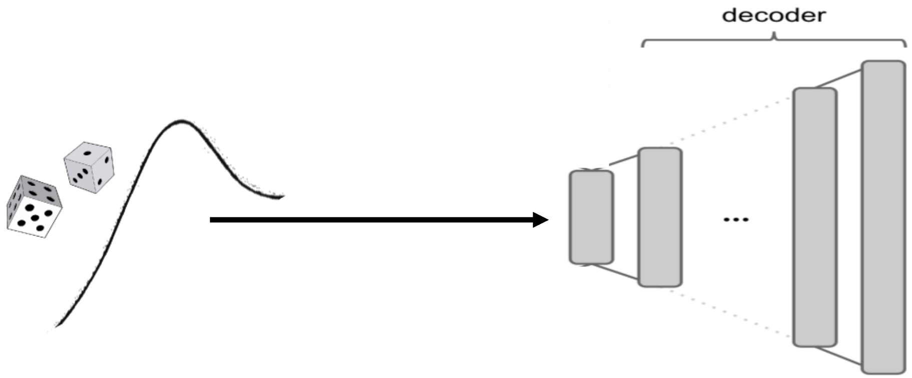
Data likelihood: $p_\theta(x) = \int \overset{N(0,1)}{p_\theta(z)} \boxed{p_\theta(x|z)} dz$

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

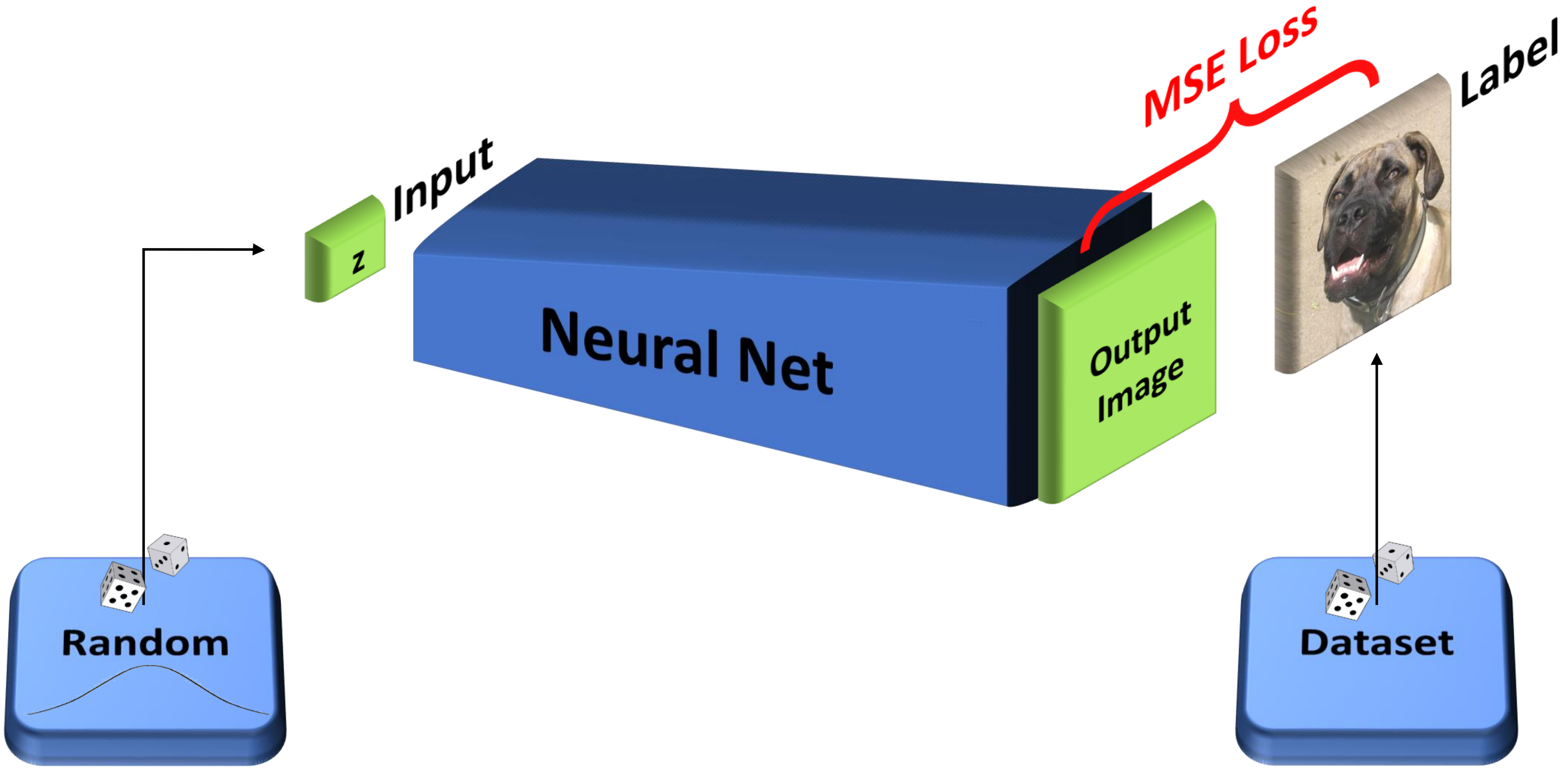


$$\log p_\theta(x^{(i)}) =$$

Generate data



How about this idea for a generative model?

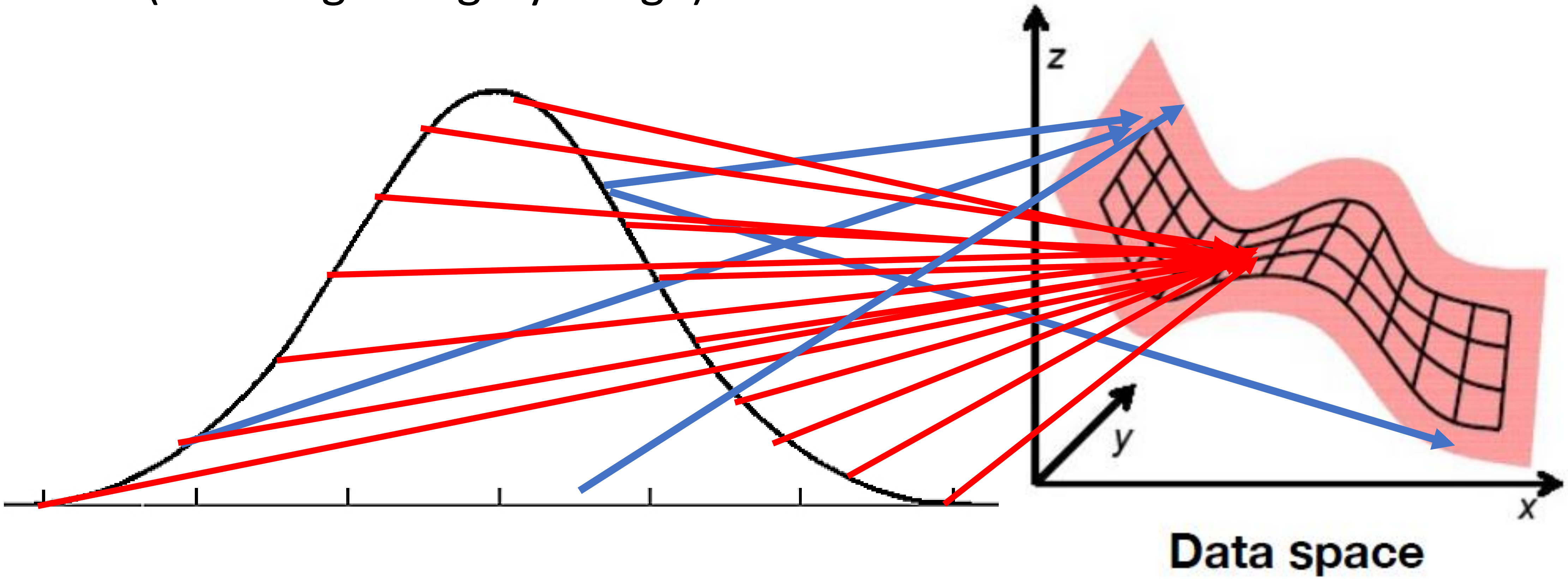


No good!

Multimodality not obtained!

In expectation: every noise is mapped to every instance

Best L2 solution: All noise is mapped to the mean
(For images: ~ grey image)



Generative Adversarial Networks

# of GAN related papers per year
(Salehi et al.)

Q: What makes a good counterfeiter?

Train G

Q: Who do you train first?

A: Alternate training! G,D,G,D….

Minimax game: Make the best cop do the worst mistake!

Update weights **Don't update weights**

Maxi

$$\boxed{\min_{G} \max_{D}\{\mathbb{E}_{x\sim p_{data}} \log(D(x)) + \mathbb{E}_{z\sim p_z} log(1 - D(G(z)))\}}$$

FAQ1: Why does it work?

- D learns probability! G trains to sample instance with high probability!
- Objective does not determine mapping directly- arrangement of latent space is learned!
- Theory: minimizes JS divergence between generated and real distributions.

FAQ2: Why alternating?

- Gradients are meaningless when game is unbalanced.

- Pre-train D? Negative examples?

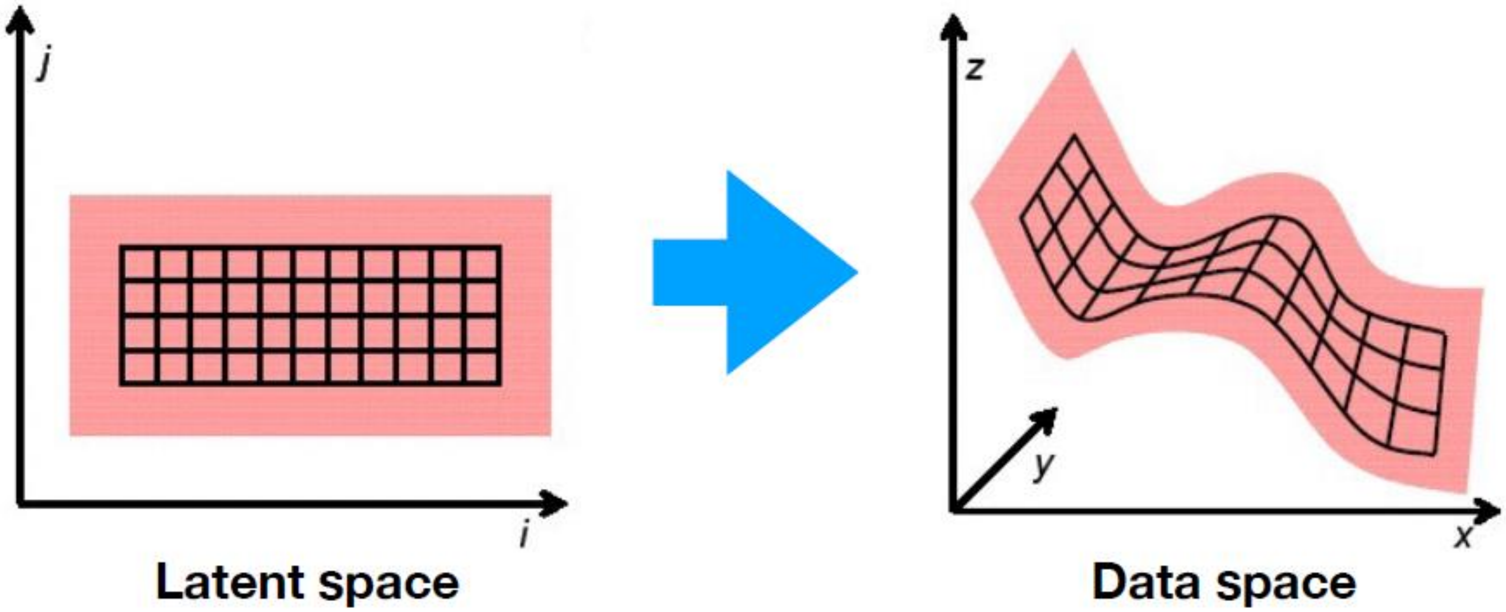
- Pre-train G? What loss?
  For G, D is a **learned loss function**

GANs, Goodfellow 2014

DCGAN Radford 2015

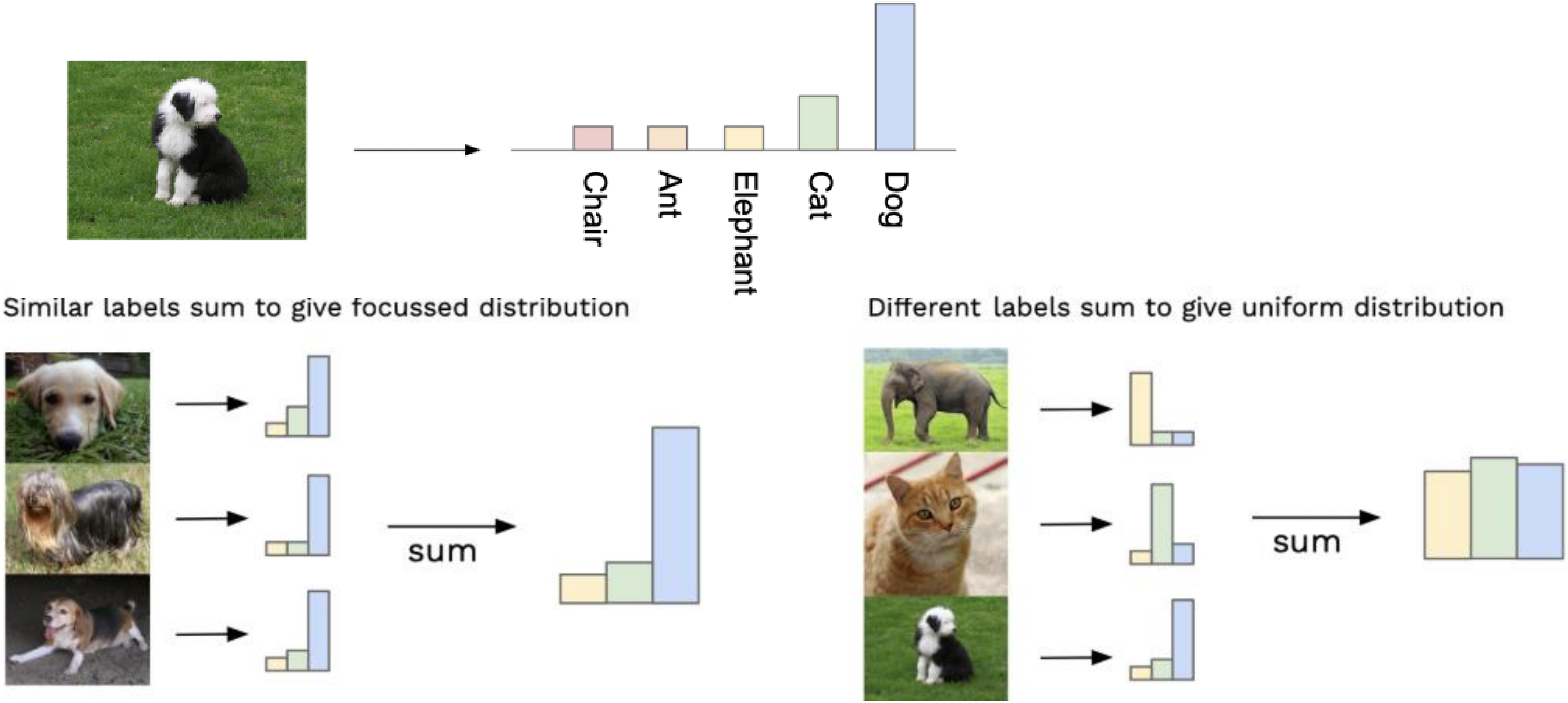
Latent space interpolation

Why does it work?



1. Every point is mapped to a valid example.
2. Network is continuous.

Evaluation metrics: Inception score

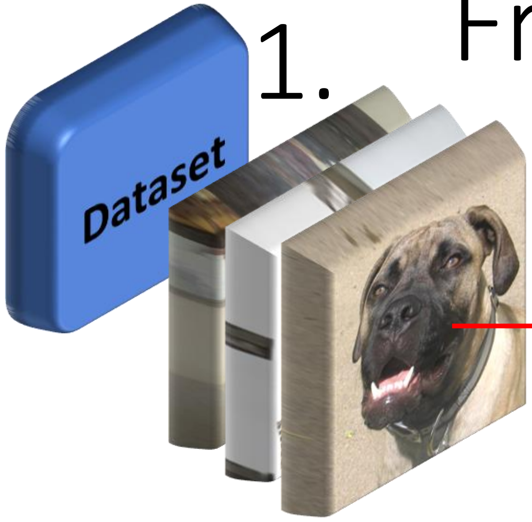


$$\mathbf{IS}(G) = \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}(p(y|\mathbf{x}) \parallel p(y)) \right)$$

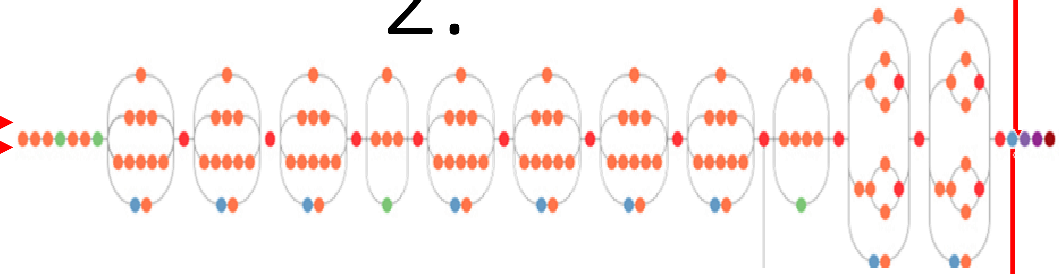
Fréchet Inception Distance (FID)

Depends on the number of samples!

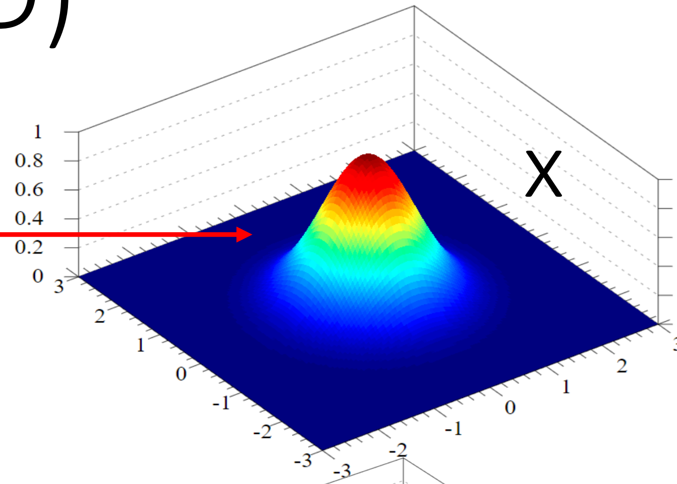
1.

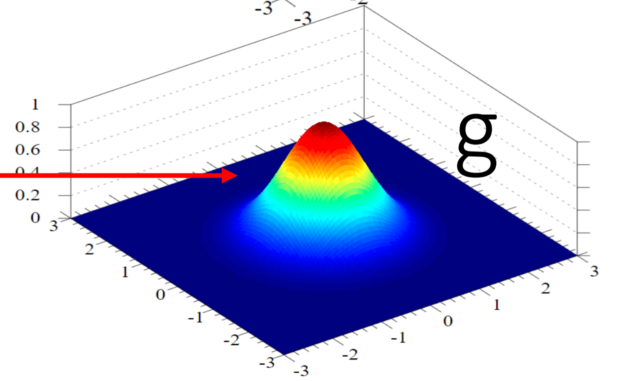


2.

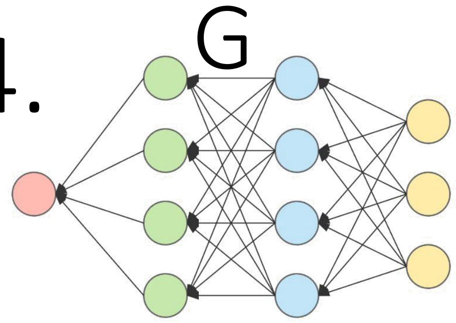


3.





4.



5. $\texttt{FID}(x,g) = ||\mu_x - \mu_g||_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}})$

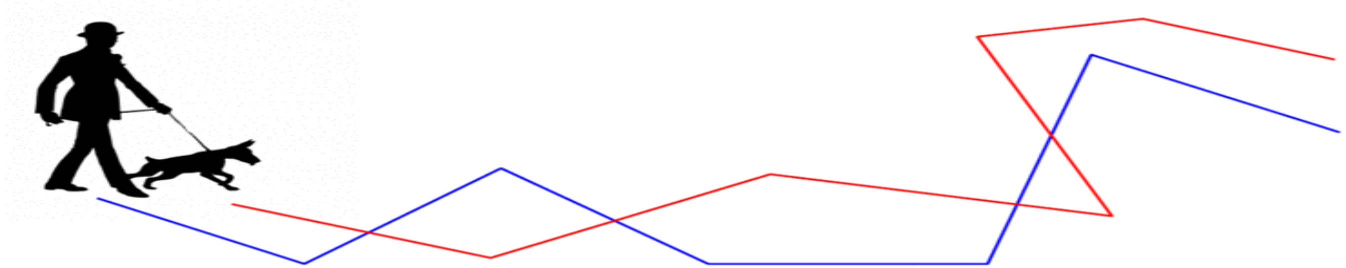
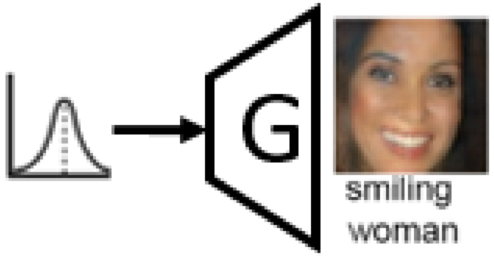
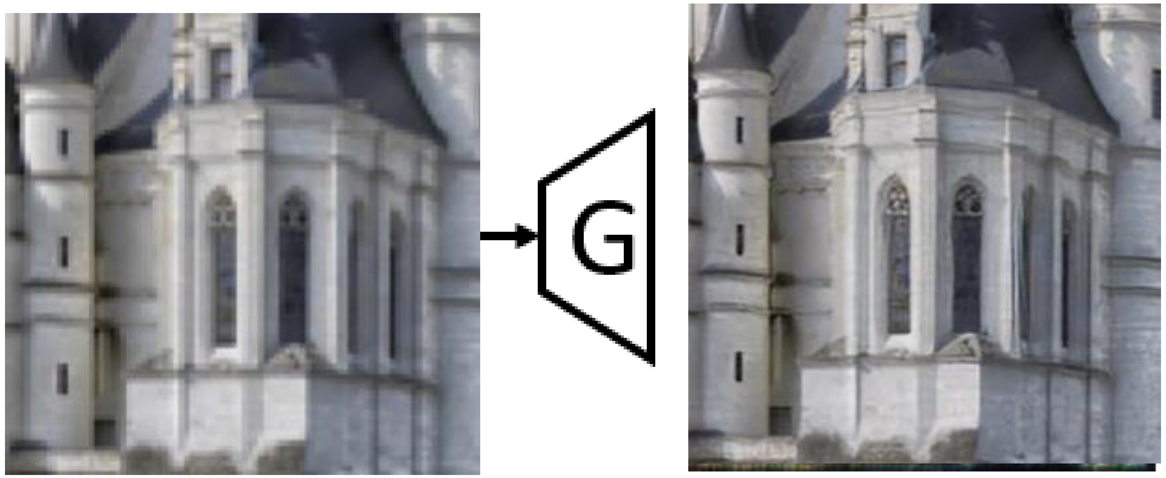


Image to Image translation

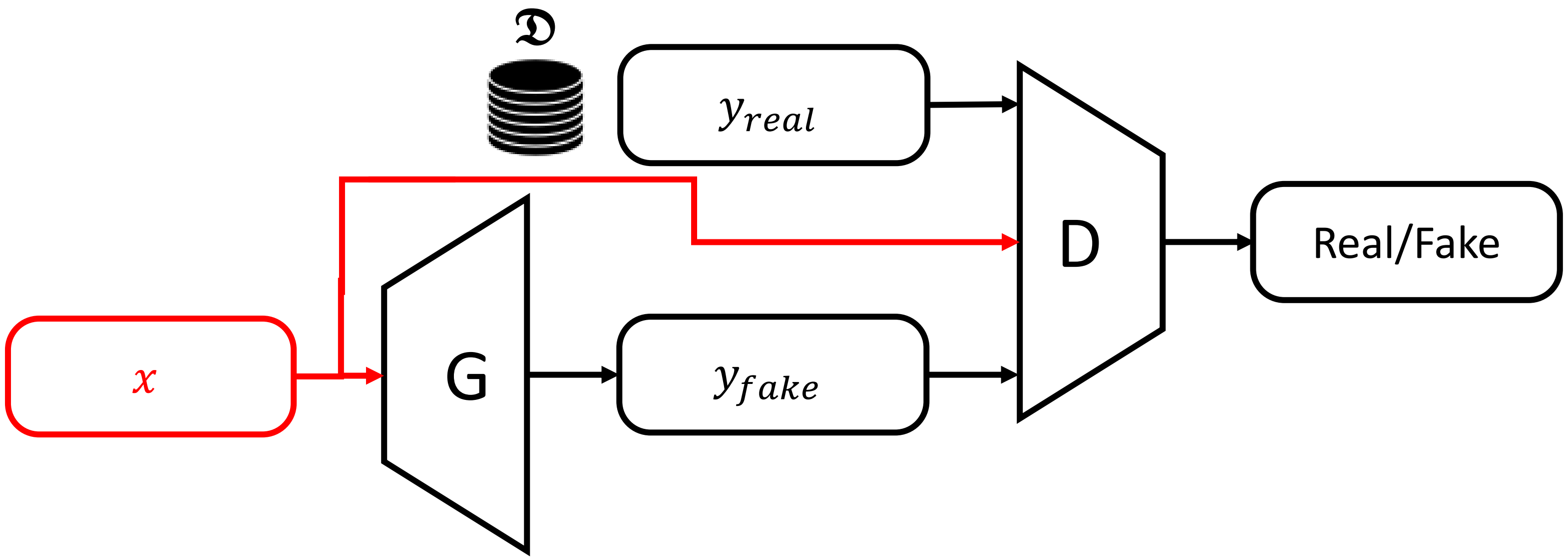




Isola et al. Nov2016

Slide by Sefi Bell-Kligler & Akhiad Bercovich

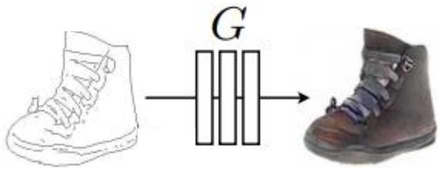
Conditional GAN



$$\mathcal{L}_{C-GAN} = \min_G \max_D \mathbb{E}[\log D(y, x)] + \mathbb{E}[\log(1 - D(G(x), x))]$$

Slide by Sefi Bell-Kligler & Akhiad Bercovich

Pix2Pix



$$\mathcal{L}_{C-GAN} = \min_{G}\max_{D} \mathbb{E}[\log D(y, x)] + \mathbb{E}[\log(1 - D(G(x), x))]$$

$$\mathcal{L}_{L1} = \|y - G(x, z)\|_1$$

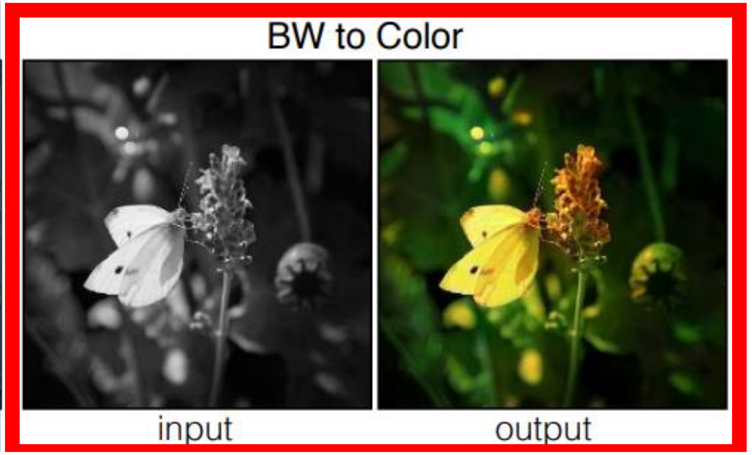
$$Objective = \mathcal{L}_{C-GAN} + \lambda \cdot \mathcal{L}_{L1}$$

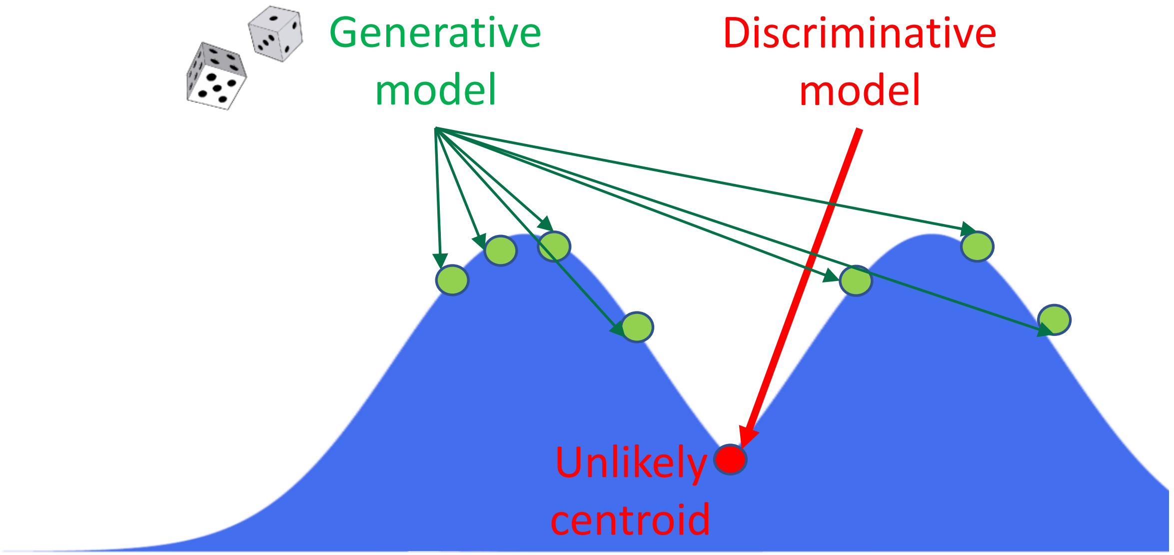
Isola et al. Nov2016

Slide by Sefi Bell-Kligler & Akhiad Bercovich

Generative VS Discriminative

What would happen if we train regular supervised mapping?





Training GANs is hard

- Stability

- Mode col

Training
GANs
Be like:

- GANs can over-train

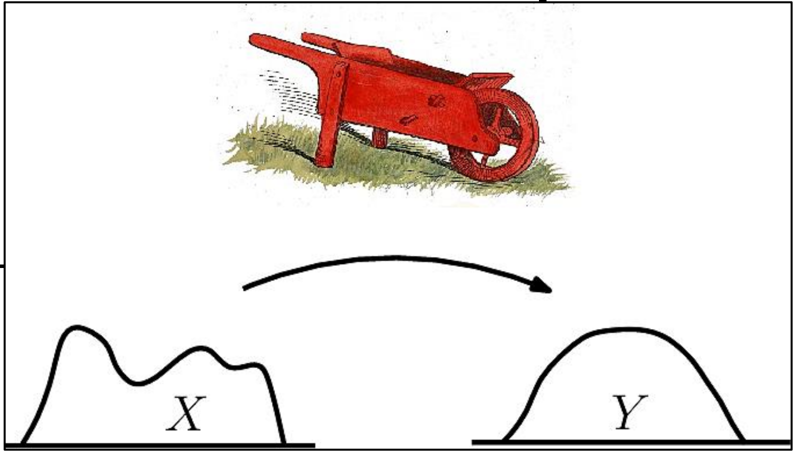
- Lo

Types of GAN losses - Wasserstein GAN

Discriminator

GAN $$\max_D E_{x\sim p_X}[\log D(x)] + E_{z\sim p_Z}[\log(1 - D(G(z))]$$

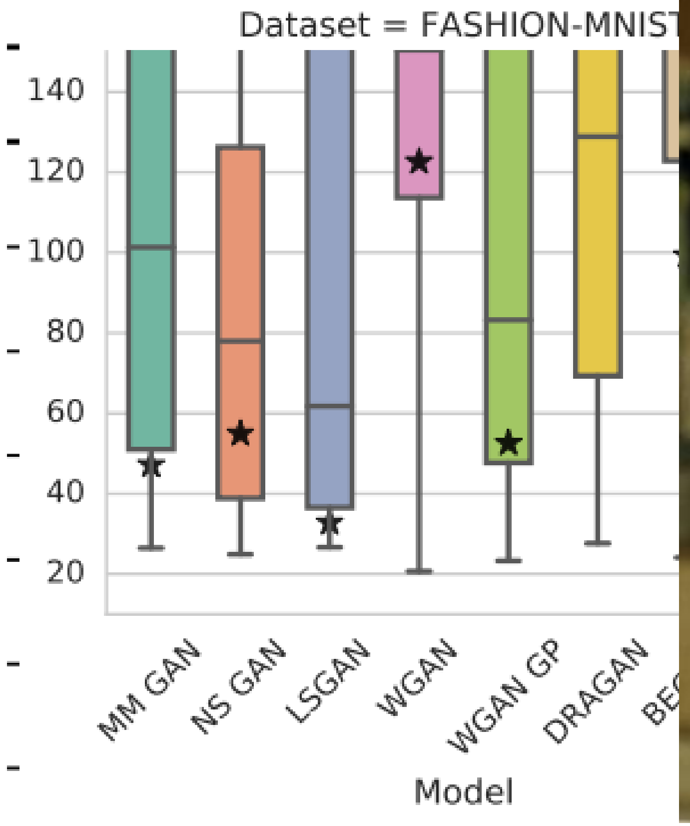
WGAN $$\max_D E_{x\sim p_X}[D(x)] - E_{z\sim p_Z}[D(G(z))]$$

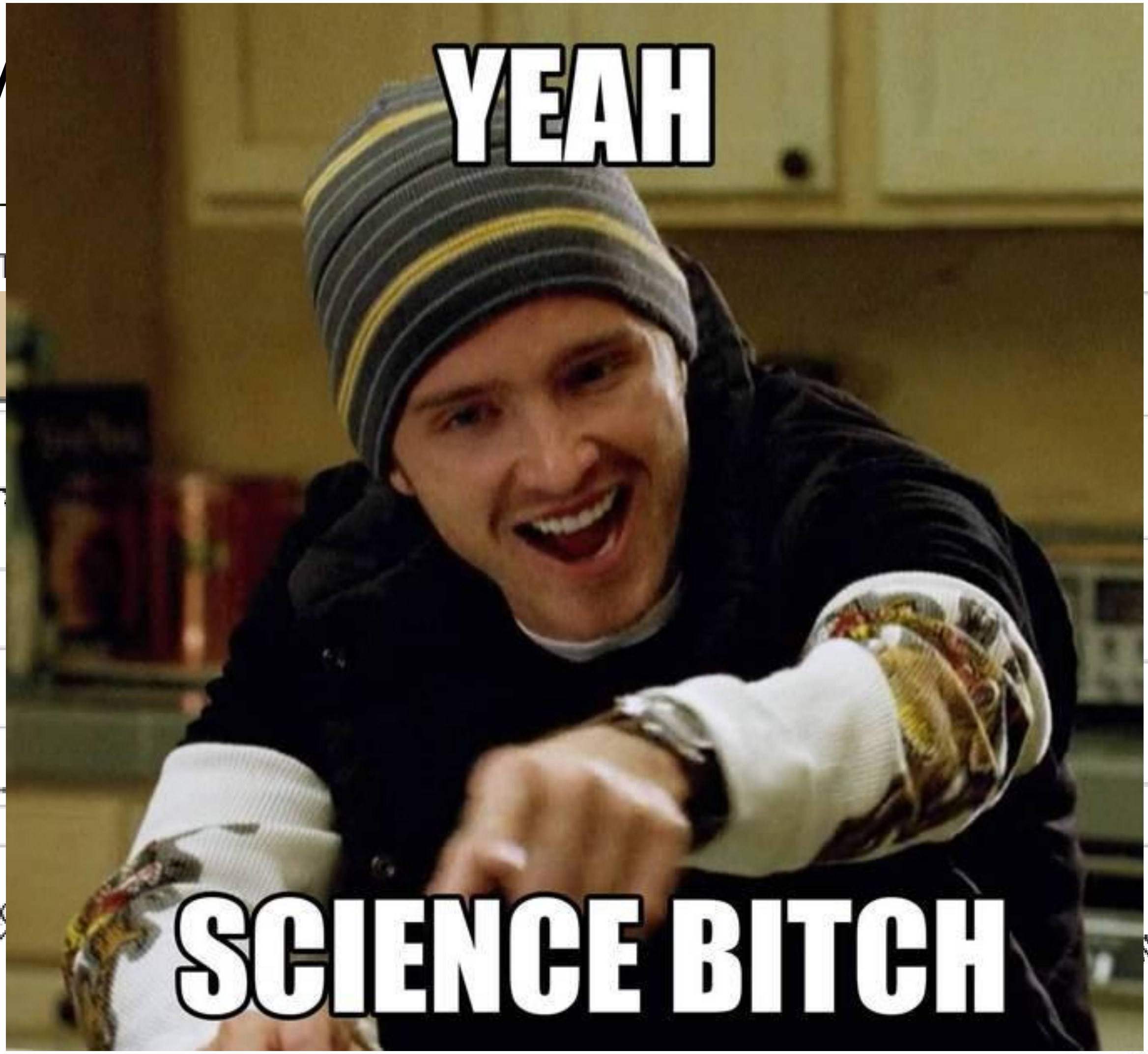


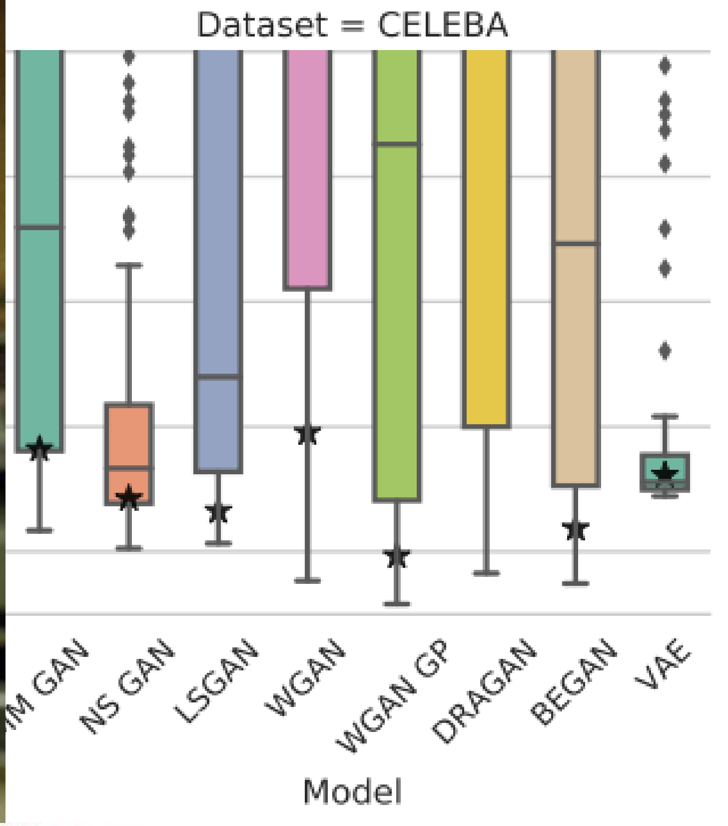
WGAN: minimize earth mover distance between p_X and $p_{G(Z)}$

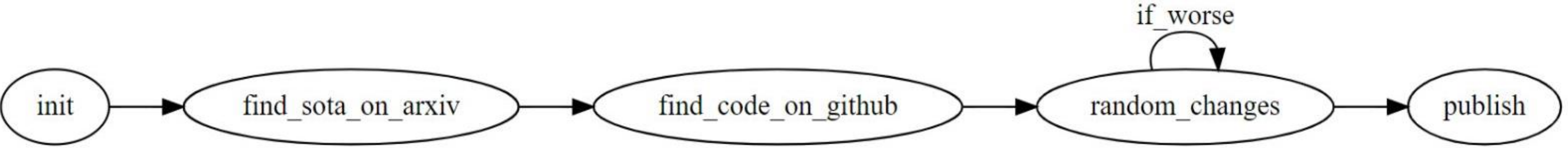
$$EM(p_X, p_{G(Z)}) = \inf_{\gamma \in \prod(p_X, p_{G(Z)})} E_{(x,y)\sim\gamma}[||x - y||]$$

Types of G



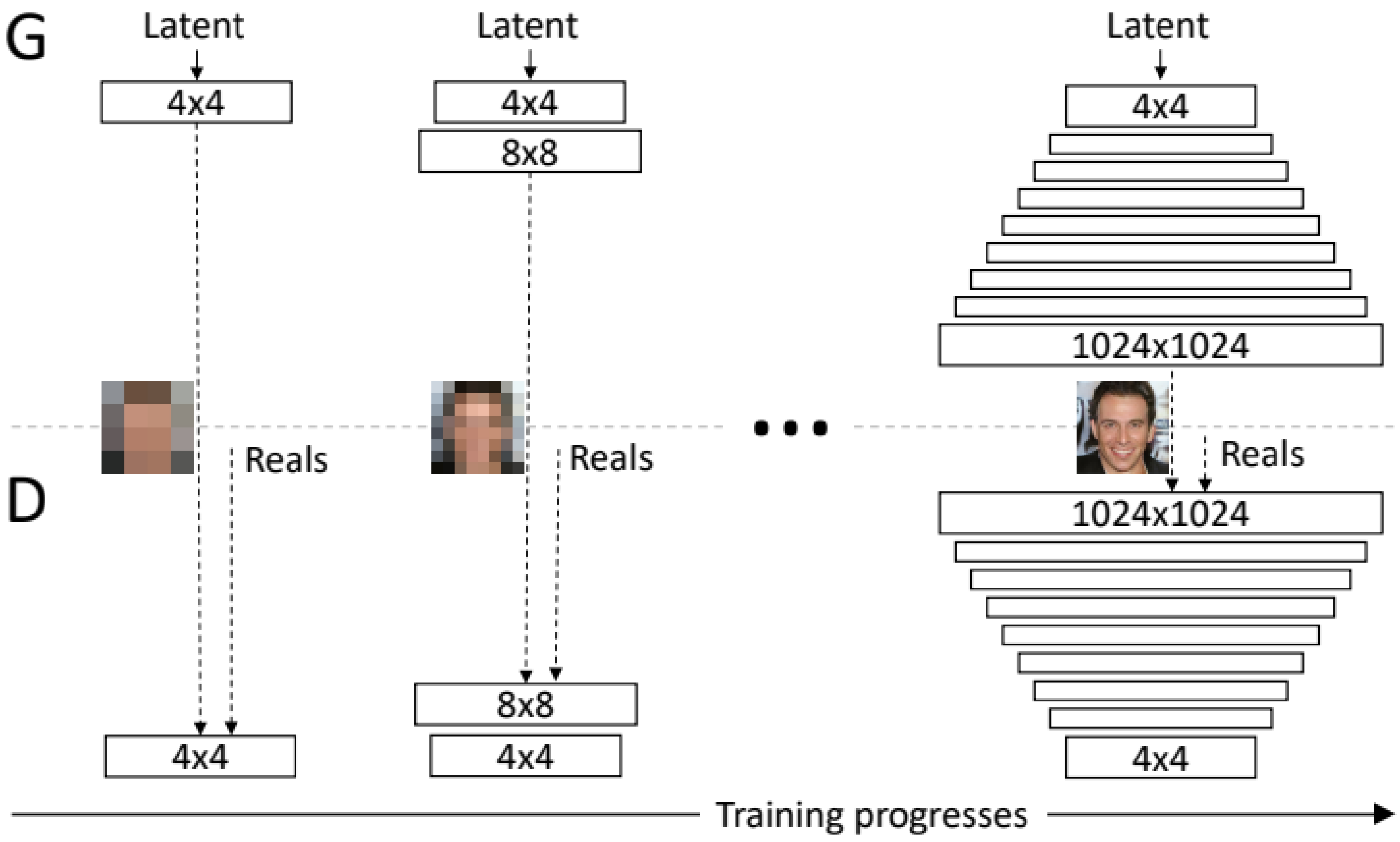






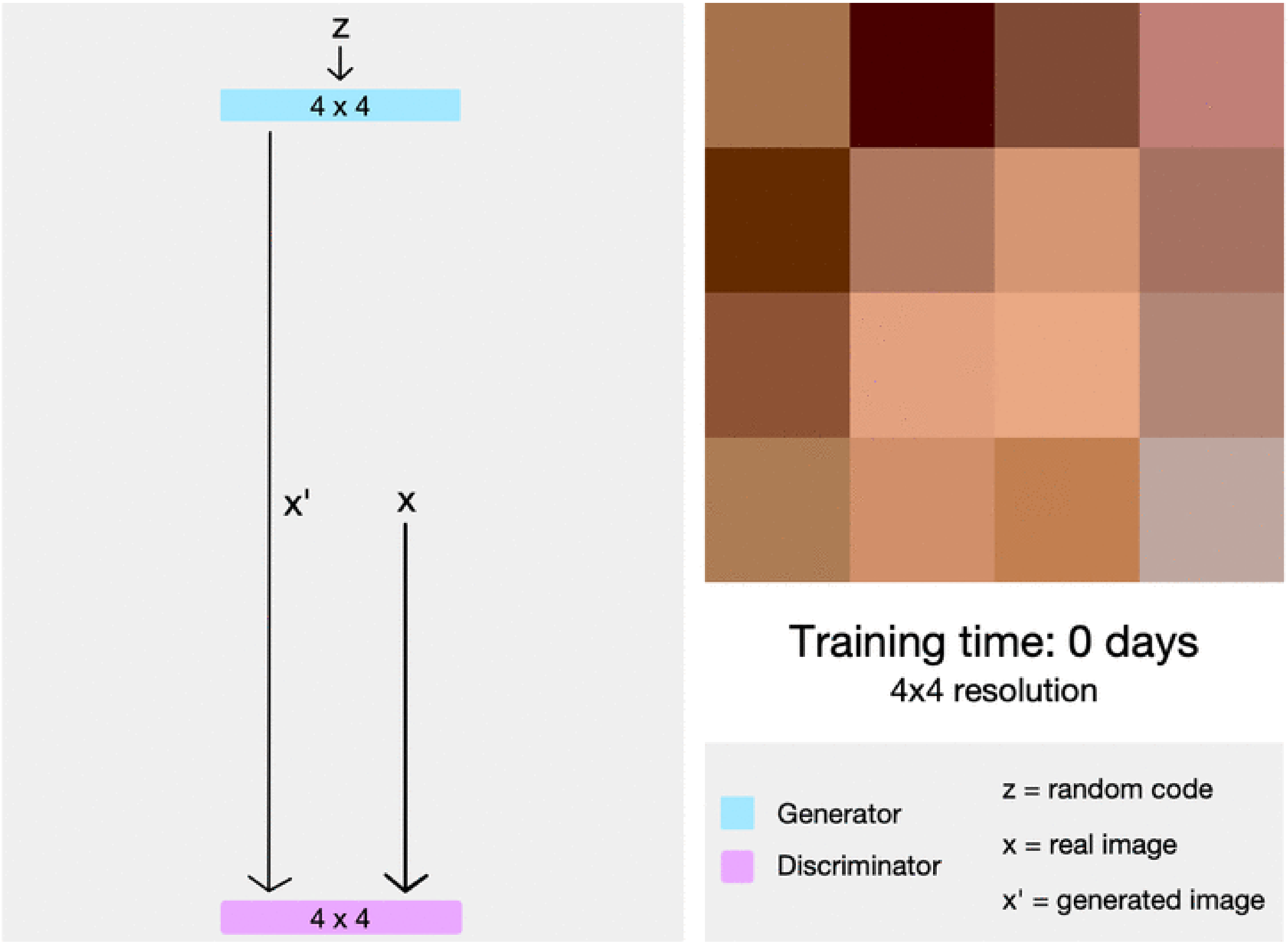
Mark Saroufim

Progressive Grow



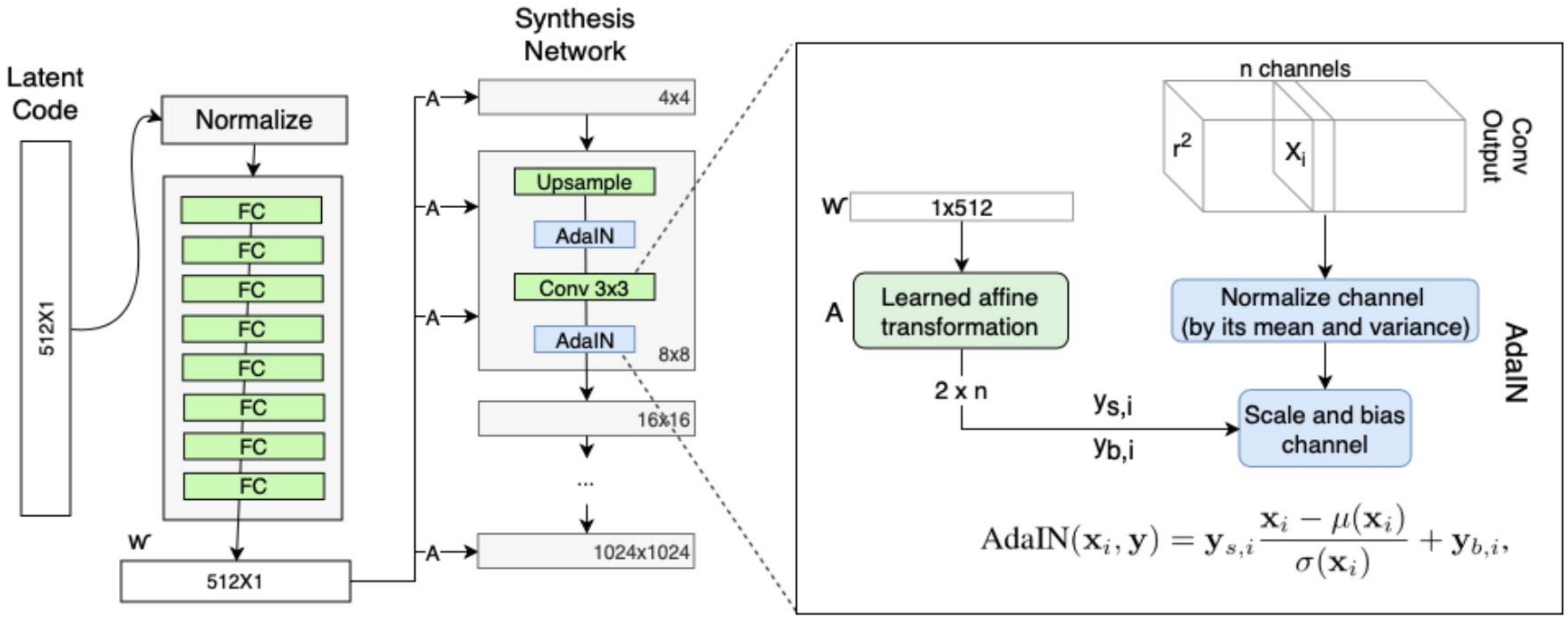
Progressive Growing of GAN, Karras et al., Feb2018

Slide by Sefi Bell-Kligler & Akhiad Bercovich



Progressive Growing of GAN, Karras et al., Feb2018

Style Modules (AdaIN)

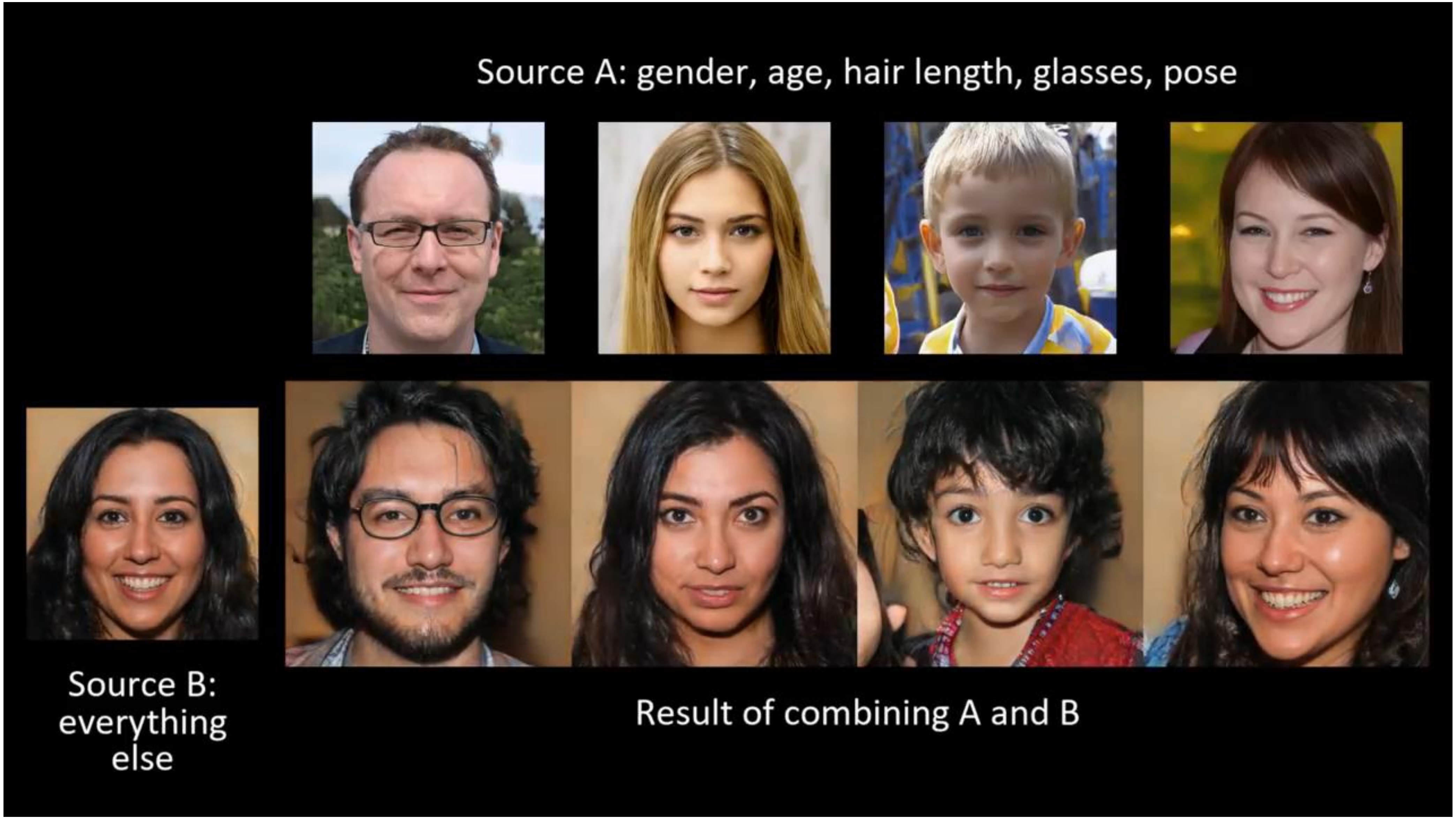


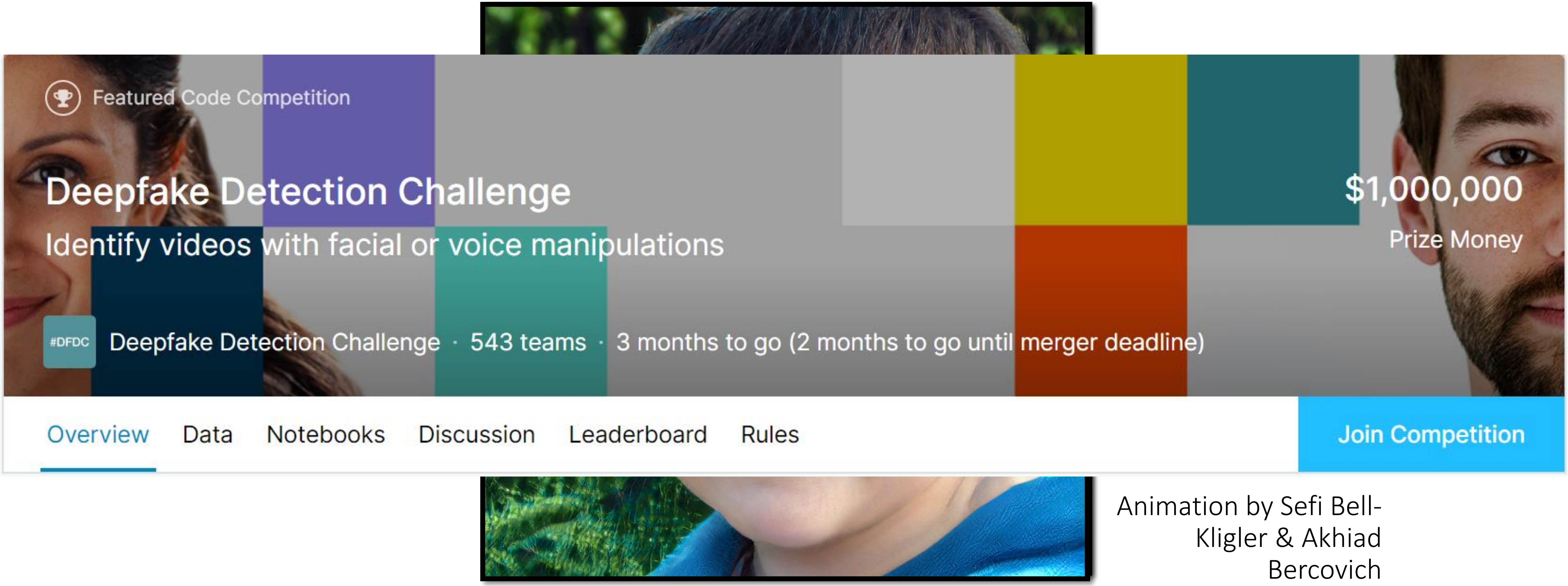
$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i}\frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i},$$

The generator's Adaptive Instance Normalization (AdaIN)

StyleGAN, Karras et al. NVIDIA 2019

Results


Source A: gender, age, hair length, glasses, pose
Source B: everything else
Result of combining A and B

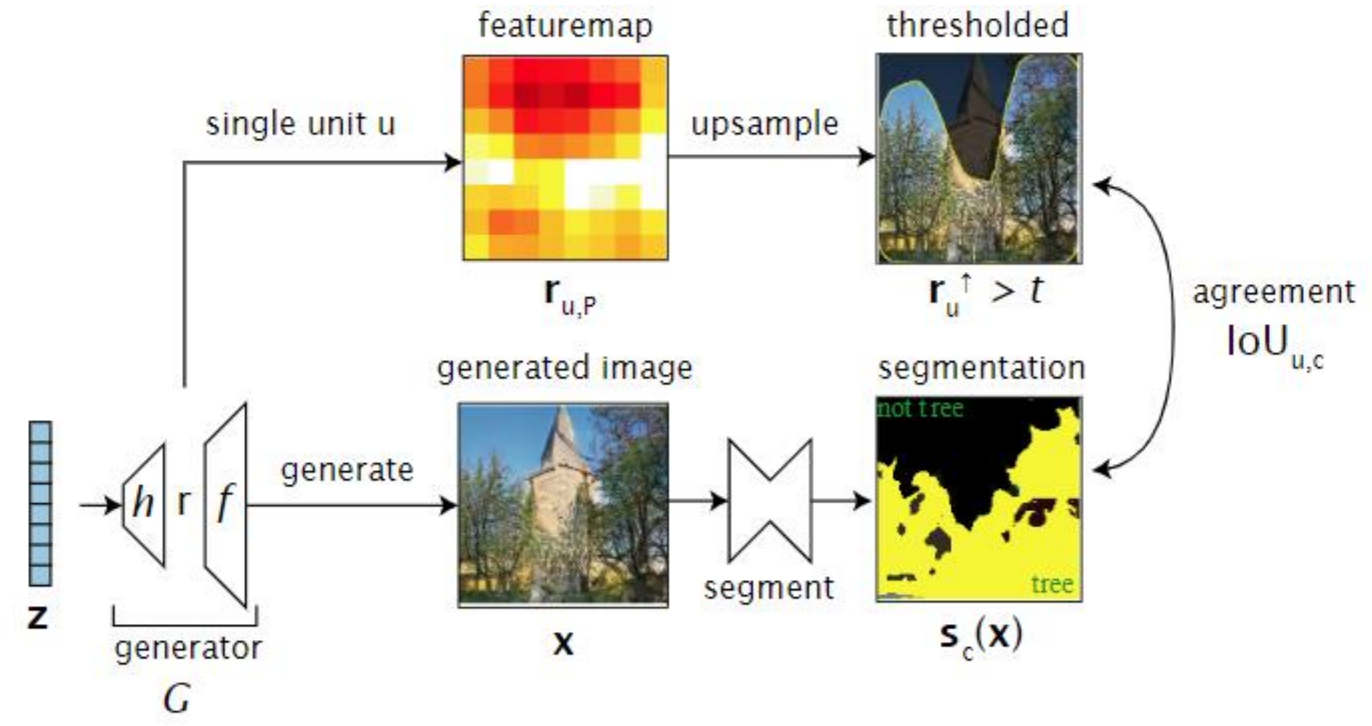


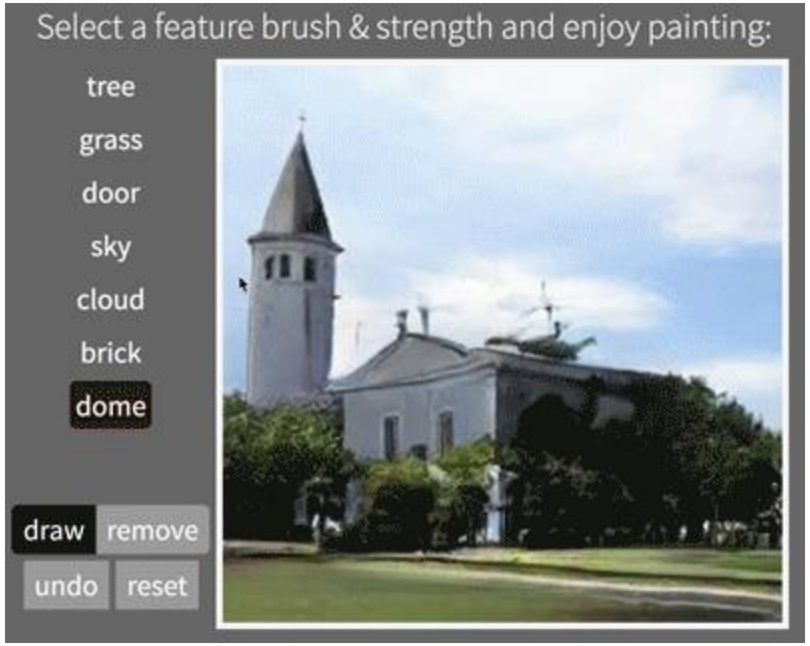
Animation by Sefi Bell-Kligler & Akhiad Bercovich

https://thispersondoesnotexist.com/

https://whichfaceisreal.com/ https://thisxdoesnotexist.com/

GAN Dissection





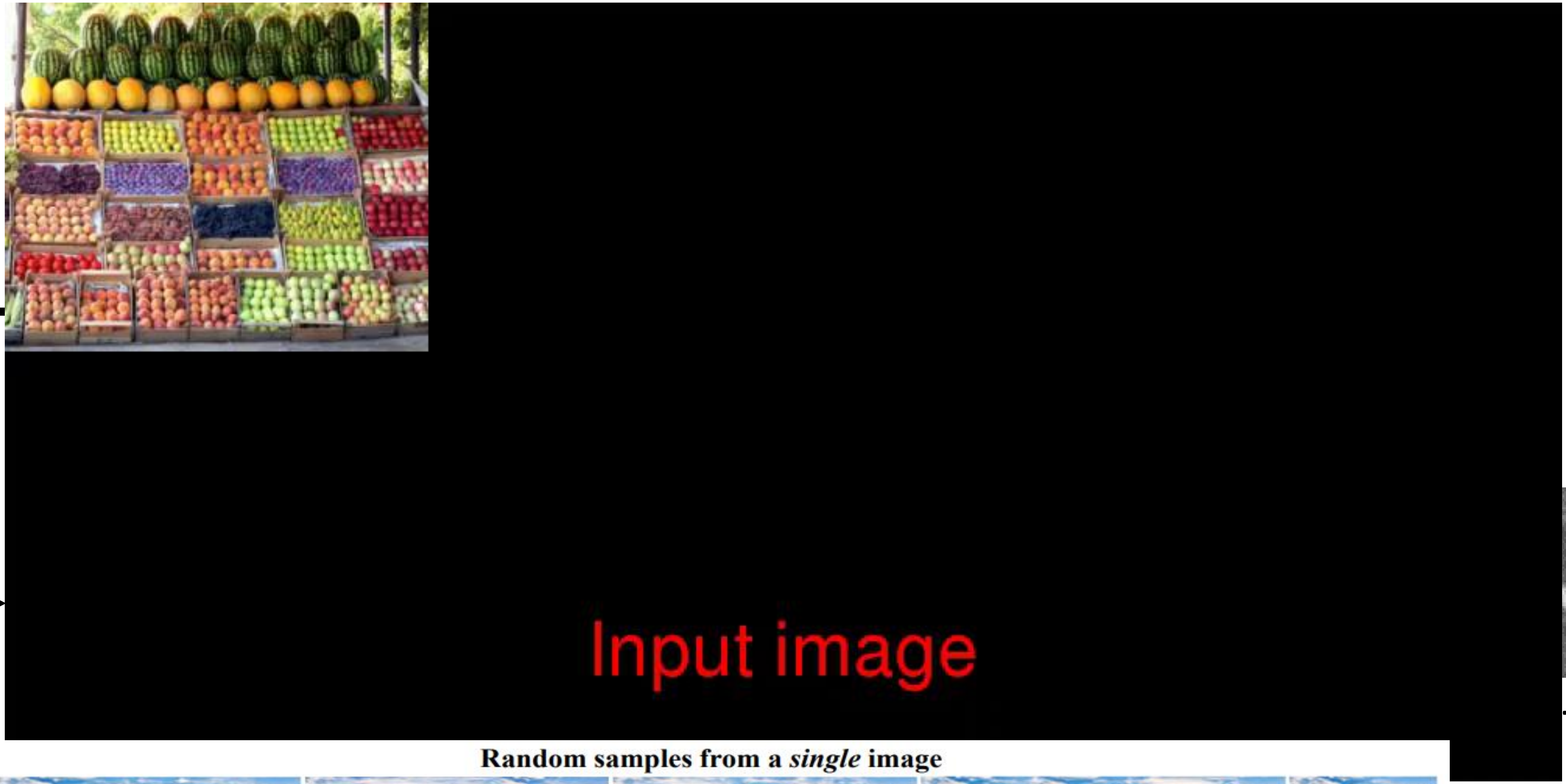
David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, Antonio Torralba

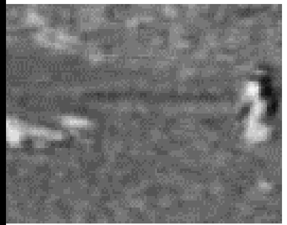
http://gandissect.res.ibm.com/ganpaint.html?project=churchoutdoor&layer=layer4

Training a GAN on a single image

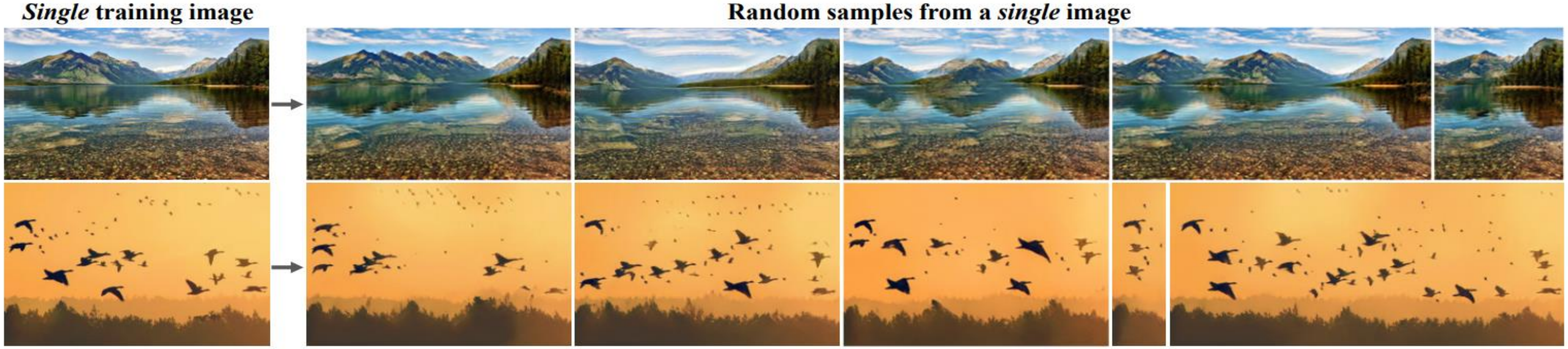
InGAN (Shocher, Bagon, Isola, Irani)

Input/noise





True/False map



SinGAN (Rott-Shaham, Dekel, Michaeli)

Thanks

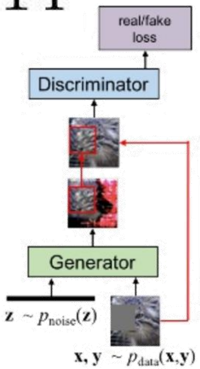
2014

2015

2016

2017

2018



Benny GANs