More Generative Models

CS280

Spring 2025

Angjoo Kanazawa

Mid-term Logistics

- March 19th Wed before spring break
  - Next Wednesday
- Written exam
- One page (8.5 x 11 in/A4) cheatsheat allowed (both sides)

Final project logistics

- Group of 3 is encouraged. Maximum 4, but more people = higher expectations.

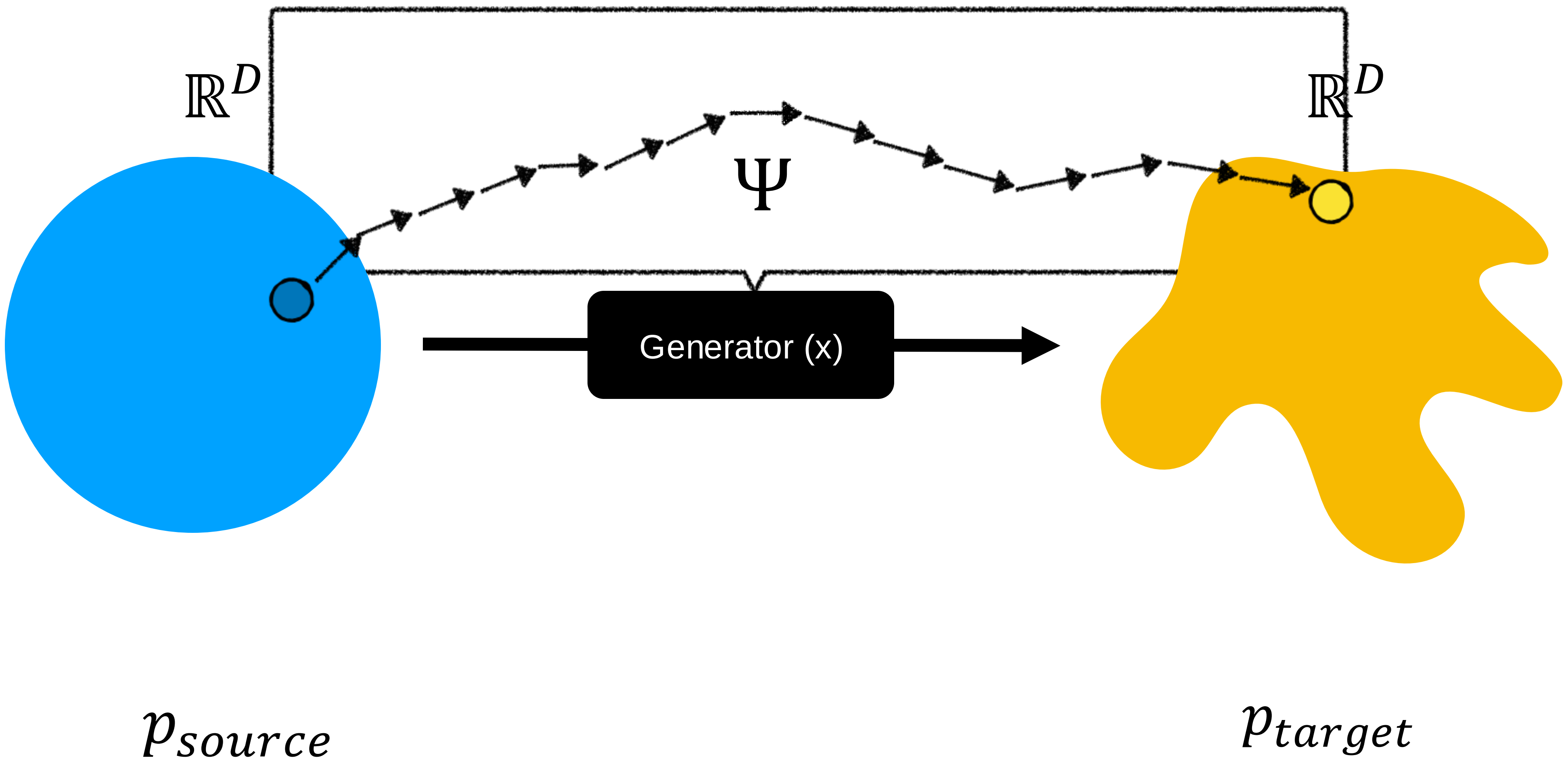
Deliverables:

- 1 page proposal due a week+ after Spring break
- Final project presentation during RRR week over the two dates 5/5 and 5/7
  - Presentation length: 3~5min
- Written report due 5/14

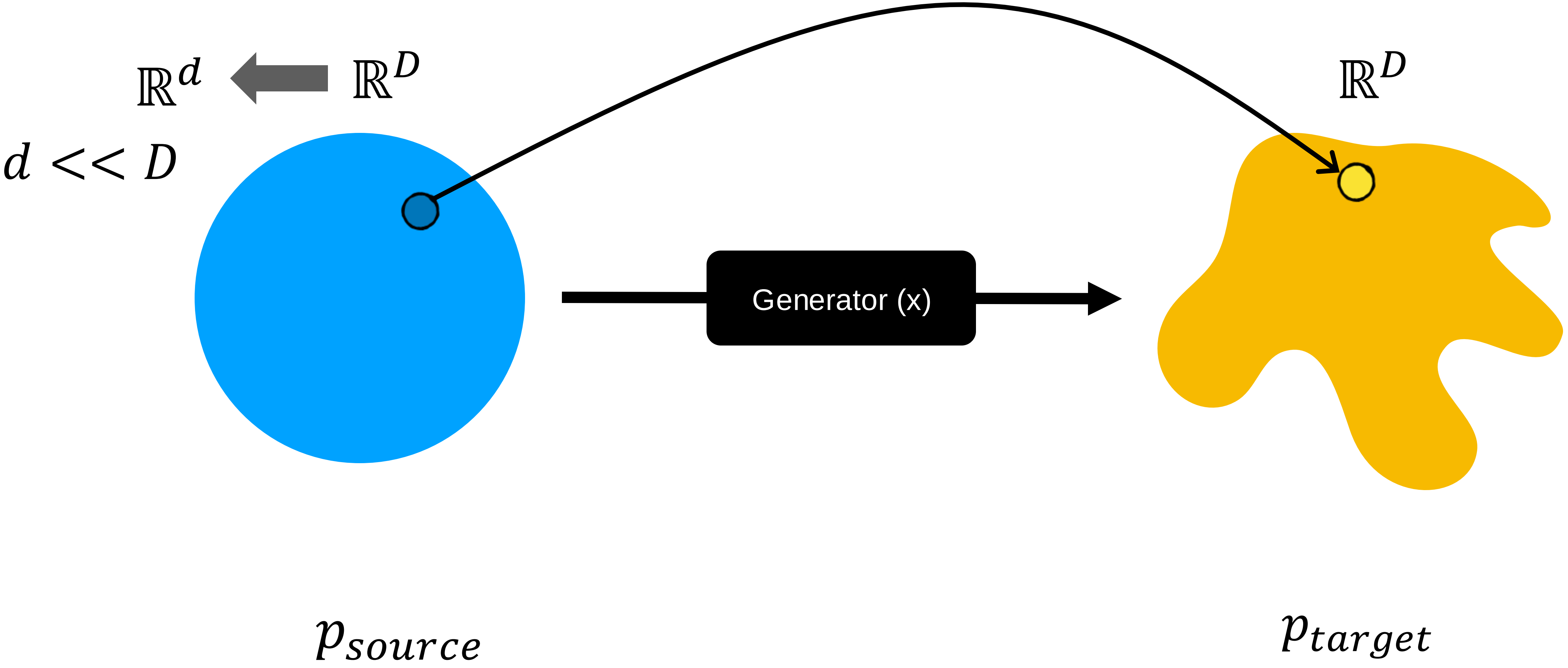
Generative Models

- Autoregressive models
- Flow based generative models
- **Variational autoencoders (VAEs)**
- **Generative adversarial networks (GANs)**

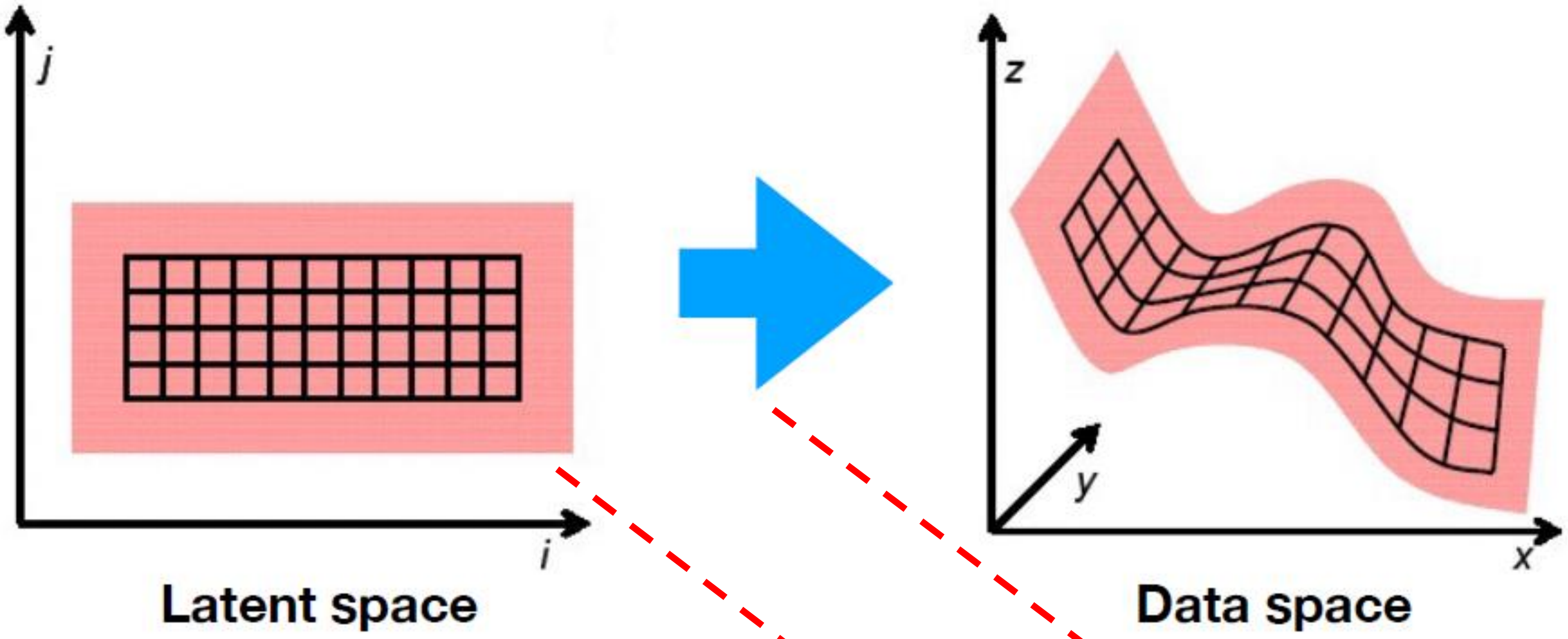
Flow based Generative Models



Latent Space Generative Models



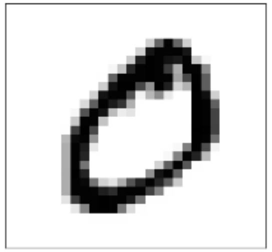
Latent space mapping approach

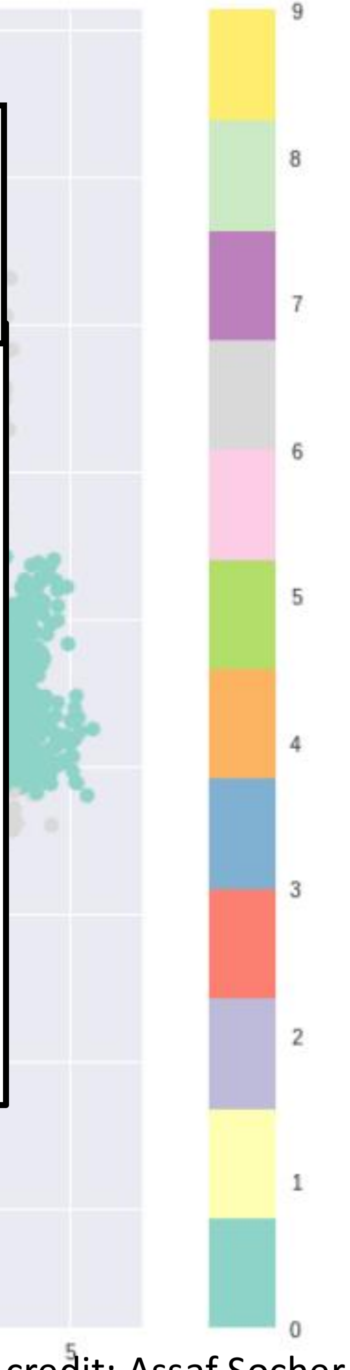


Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

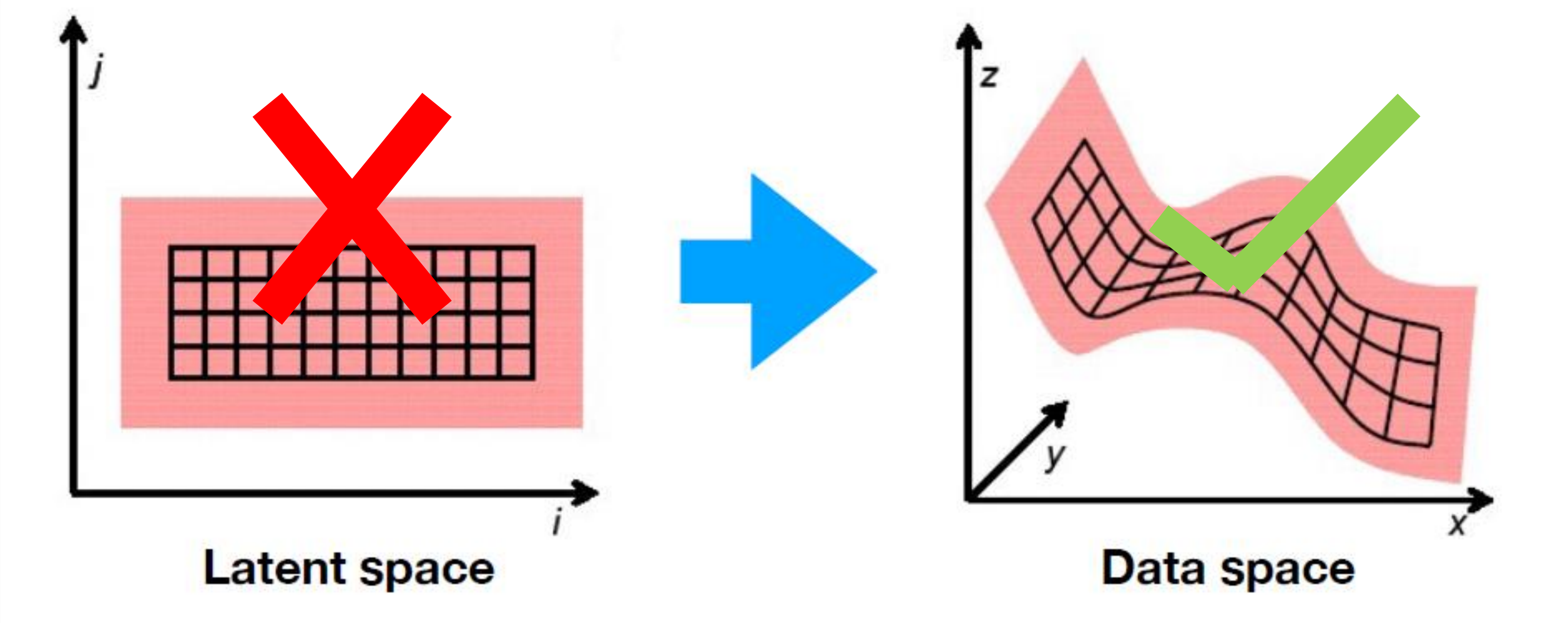
Slide credit: Assaf Socher

Autoencoders

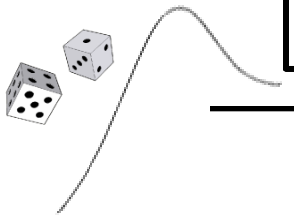


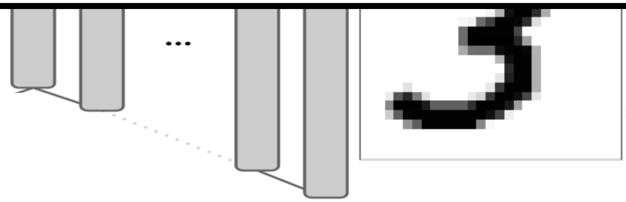


AE does not transform one *pre-determined* distribution to another!



- Gene





Slide credit: Assaf Socher

Variational Autoencoders

$$f(x) = z$$

- The Encoder maps x to z.
- But need to be able to sample from z! like p(z)
- So we need the encoder to learn p(z|x)
- But how to get p(z|x)….?
- What's wrong?
- So learn q(z|x) and make it close to p(z)
- Through ELBO (will do later):

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

$$DL(q(z|x)||p(z))$$

$$\max \log p(x) >= \mathbb{E}_{q(z|x)} \log p(x|z) - DL(q(z|x)||p(z))$$

- i.e. max p(x|z) while regularize q(z|x) to be close to p(z)

KL divergence on q(z|x) with p(x)

- Has a nice closed form if we use gaussians for q and p

$$q(z|x) = \mathcal{N}(\mu, \sigma^2) \qquad\qquad p(z) = \mathcal{N}(0, I)$$

$$D_{\mathrm{KL}}(q(z|x) \parallel p(z)) = \frac{1}{2} \sum_{i=1}^{n} \left[\sigma_i^2 + \mu_i^2 - 1 - \log \sigma_i^2\right]$$

Loss: $$\mathbb{E}_{q(z|x)} \log p(x|z) - \frac{1}{2} \sum_{i=1}^{n} \left[\sigma_i^2 + \mu_i^2 - 1 - \log \sigma_i^2\right]$$

Objective: $\mathbb{E}_{z\sim q(z|x)} \log p(x_i|z) - DL(q(z|x_i)||p(z))$

Another perspective

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x\in\mathcal{X}} P(x)\ \log\left(\frac{P(x)}{Q(x)}\right).$$

$$D_{KL}(q_\phi(z|x_i)||p(z)) = \mathbb{E}_{z\sim q_\phi(z|x_i)}\left[\log \frac{q_\phi(z|x_i)}{p(z)}\right]$$

$$= \underbrace{\mathbb{E}_{z\sim q_\phi(z|x_i)}[\log q_\phi(z|x_i)]}_{-\mathcal{H}(q_\phi(z|x_i))} - \mathbb{E}_{z\sim q_\phi(z|x_i)}[\log p(z)]$$

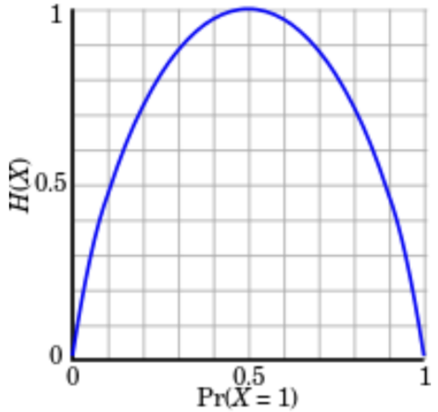
Entropy!

$$\mathcal{H}(p) = -E_{x\sim p(x)}[\log p(x)] = -\int_x p(x)\log p(x)dx$$

$$-D_{KL}(q_\phi(z|x_i)||p(z)) = \mathbb{E}_{z\sim q_\phi(z|x_i)}[\log p(z)] + \mathcal{H}(q_\phi(z|x_i))$$

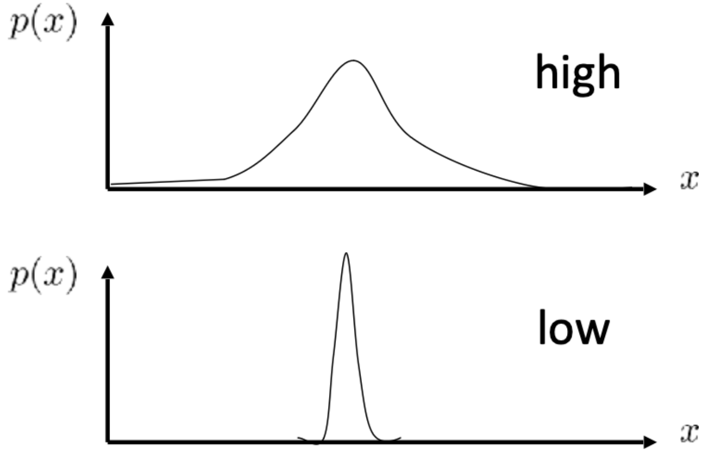
Re-written Objective: $\mathbb{E}_{z\sim q(z|x)} \log p(x_i|z) + \log p(z) + \mathcal{H}(q(z|x_i))$

A brief aside...



Entropy:

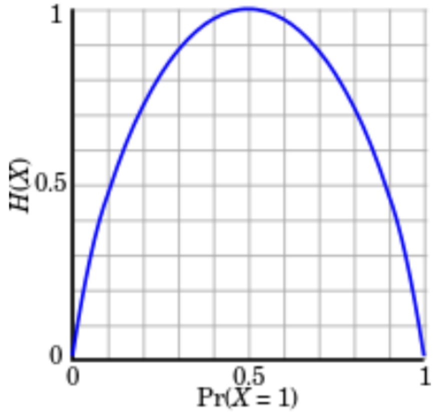
$$\mathcal{H}(p) = -E_{x \sim p(x)}[\log p(x)] = -\int_x p(x) \log p(x) dx$$



Intuition 1: how *random* is the random variable?

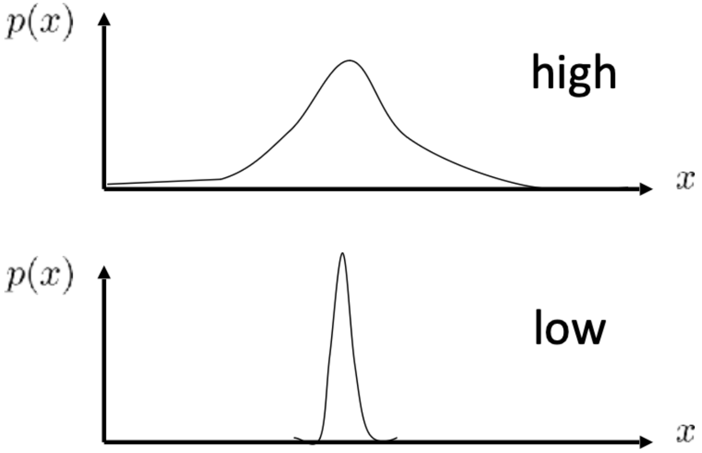
Intuition 2: how large is the log probability in expectation *under itself*

A brief aside...



Entropy:

$$\mathcal{H}(p) = -E_{x \sim p(x)}[\log p(x)] = -\int_x p(x) \log p(x) dx$$



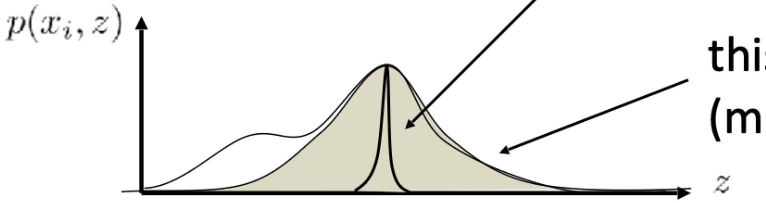
Intuition 1: how *random* is the random variable?

Intuition 2: how large is the log probability in expectation *under itself*

what do we expect this to do?

$$E_{z \sim q_i(z)}[\log p(x_i|z) + \log p(z)] + \mathcal{H}(q_i)$$

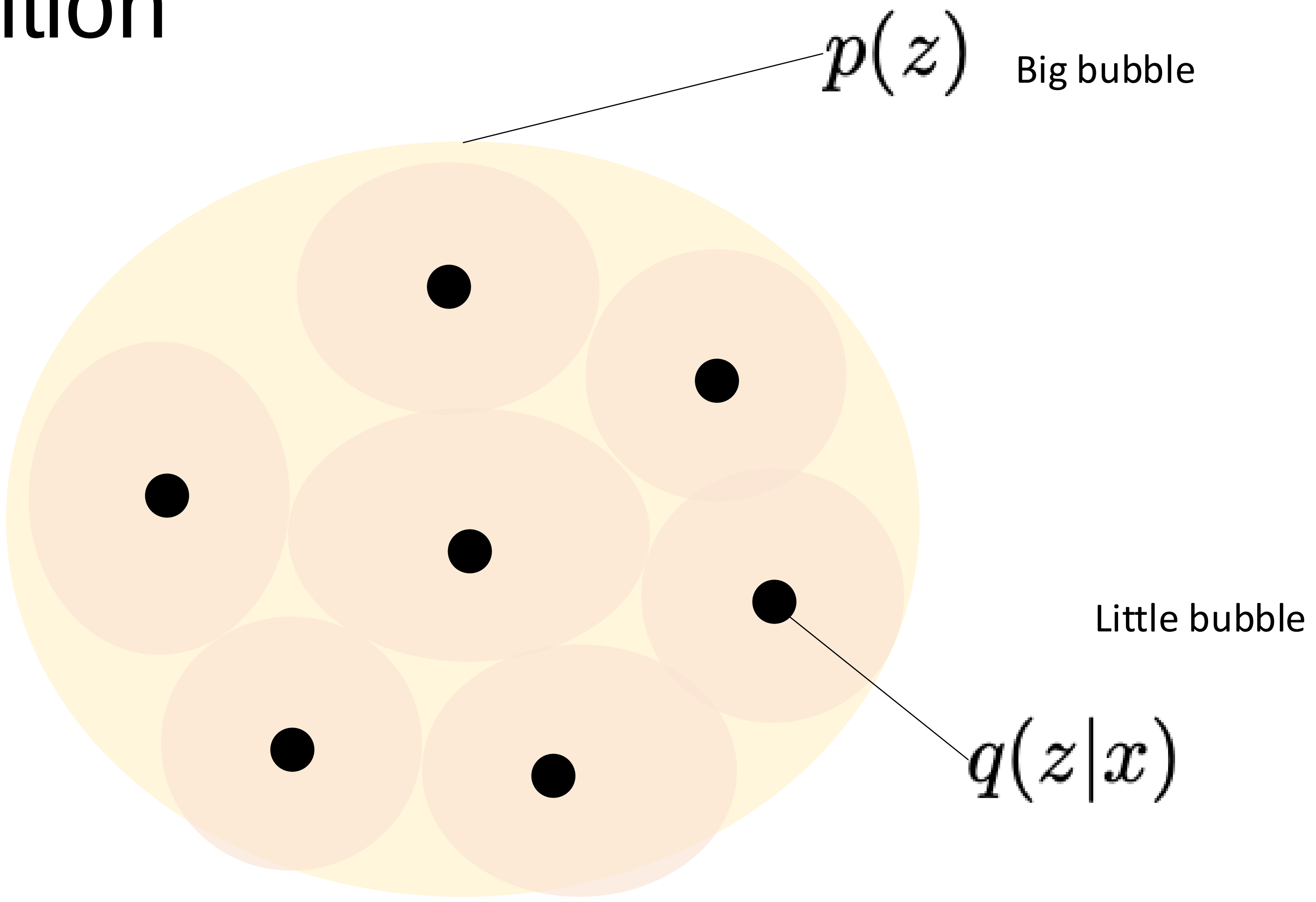
this maximizes the first part



this also maximizes the second part
(makes it as wide as possible)

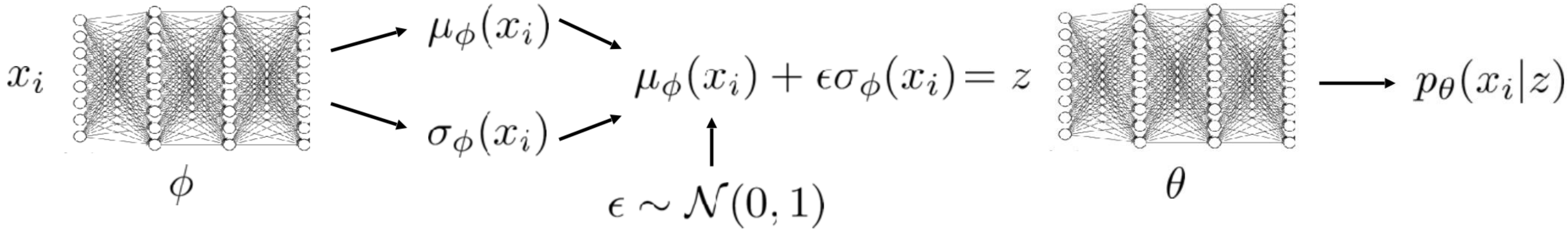
Slide credit: Chelsea Finn CS330

Another intuition



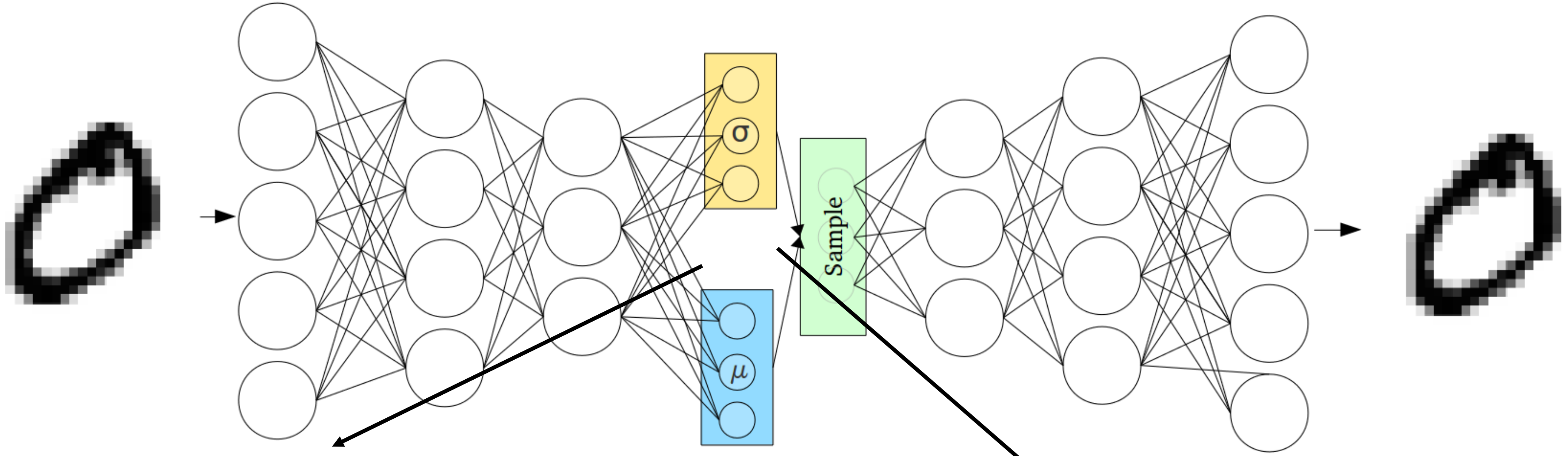
Nice converstaion with Yan LeCunn

How do you train through sampling?



Slide adapted from: Chelsea Finn CS330

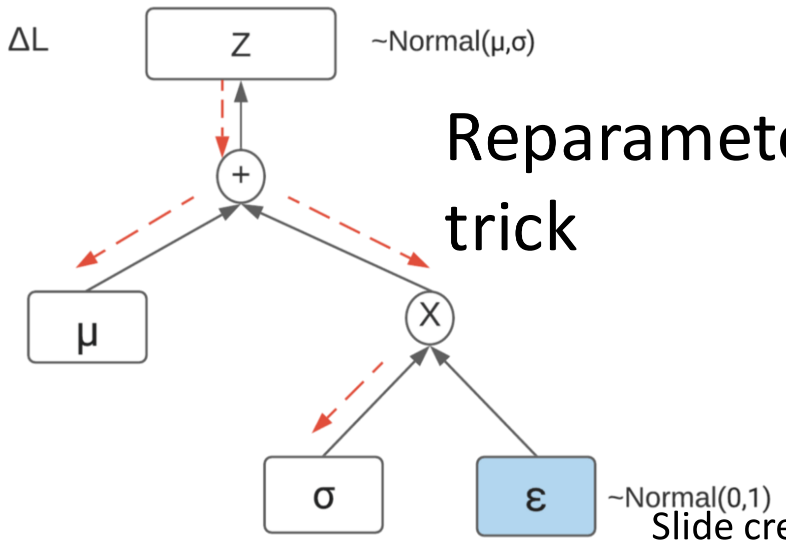
Variational Autoencoders (Kingma&Welling 2014)



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

by KL divergence:

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

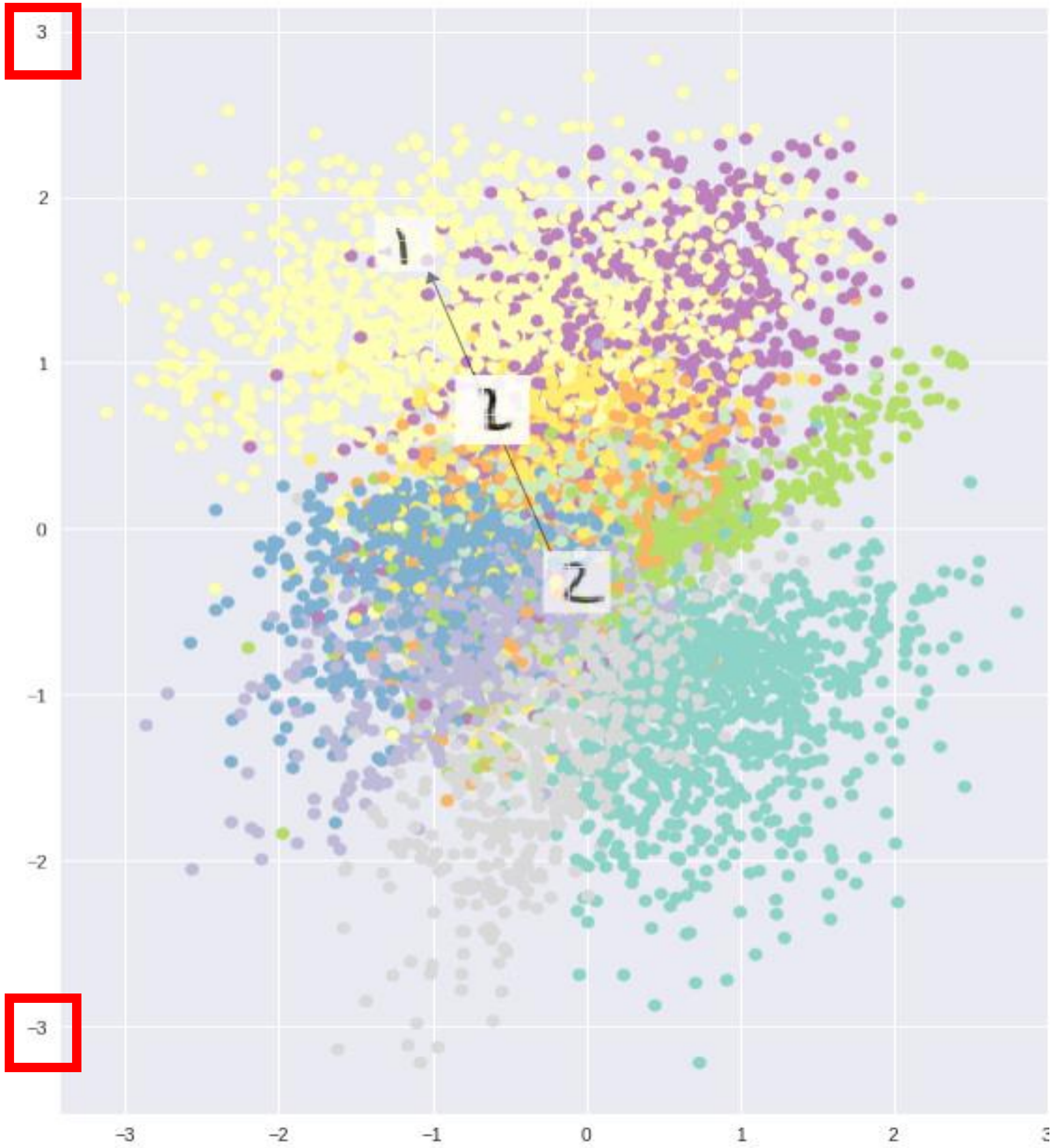


Reparameterization trick

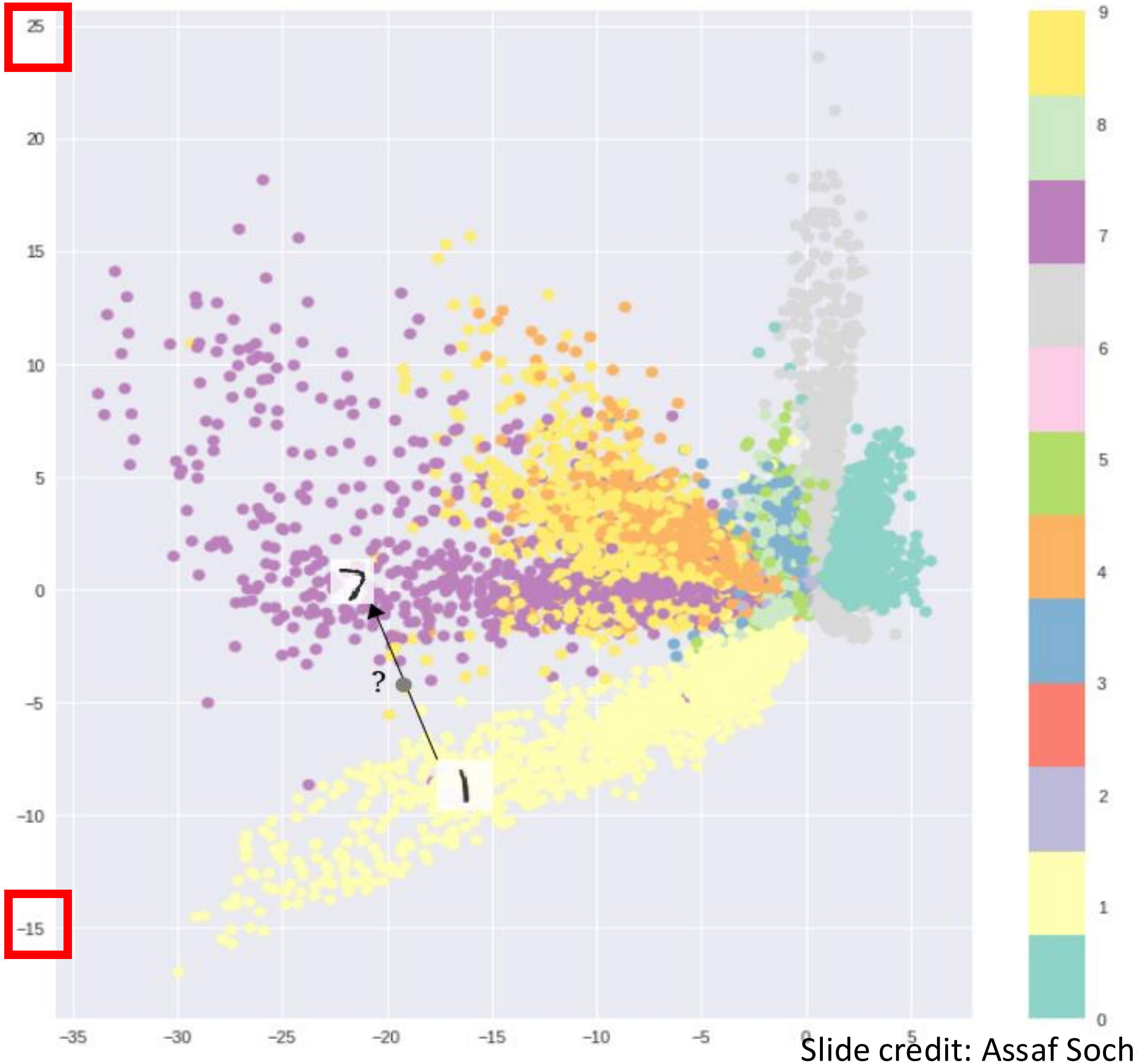
Slide credit: Assaf Socher

Also check out the scale!

VAE



AE

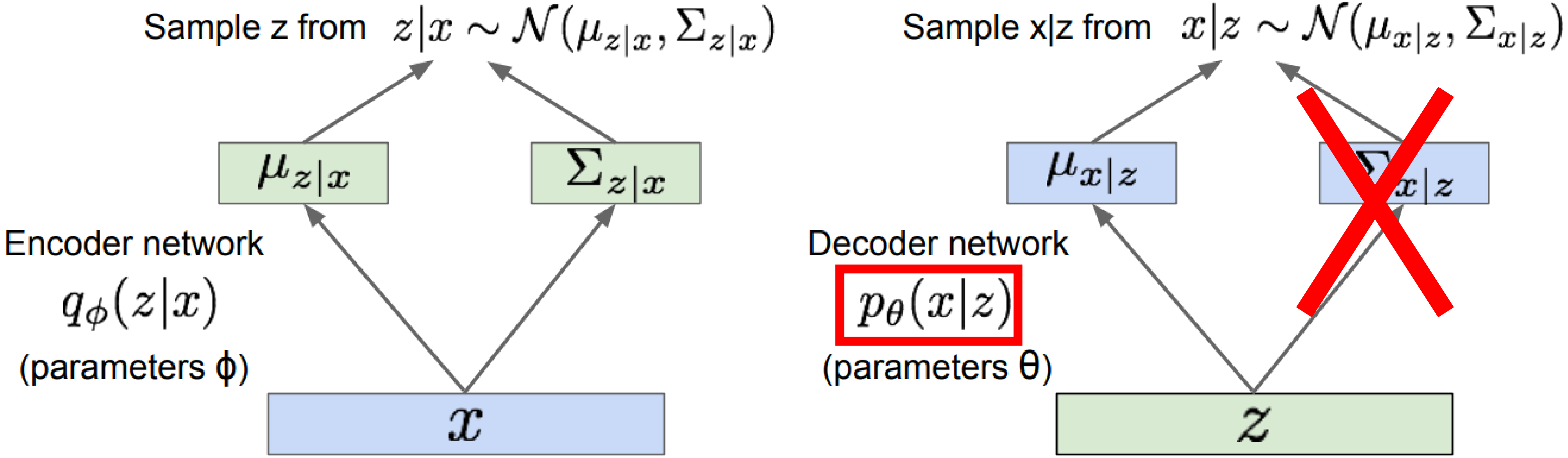


Slide credit: Assaf Socher

Probabilistic interpretation

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

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



Slide credit: Assaf Socher

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

Slide credit: Stanford cs231n

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad (\text{Multiply by constant}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Logarithms}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))
\end{aligned}
$$

Decoder network gives $p_\theta(x|z)$, can compute estimate of this term through sampling. (Sampling differentiable through reparam. trick, see paper.)

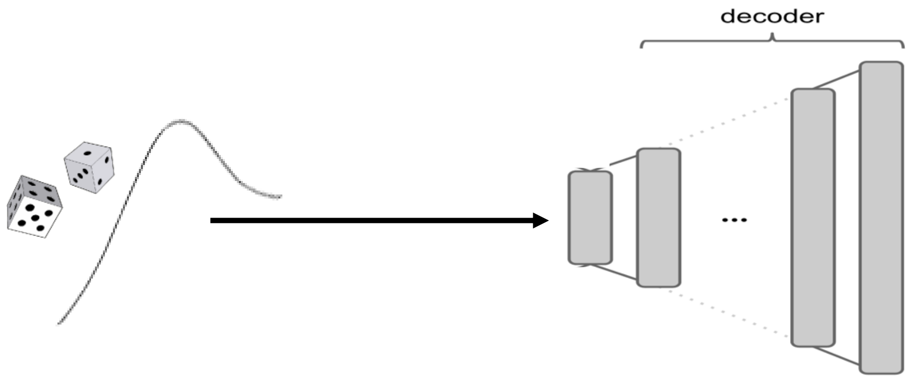
This KL term (between Gaussians for encoder and z prior) has nice closed-form solution!

$p_\theta(z|x)$ intractable (saw earlier), can't compute this KL term :(But we know KL divergence always >= 0.

Slide credit: Stanford cs231n

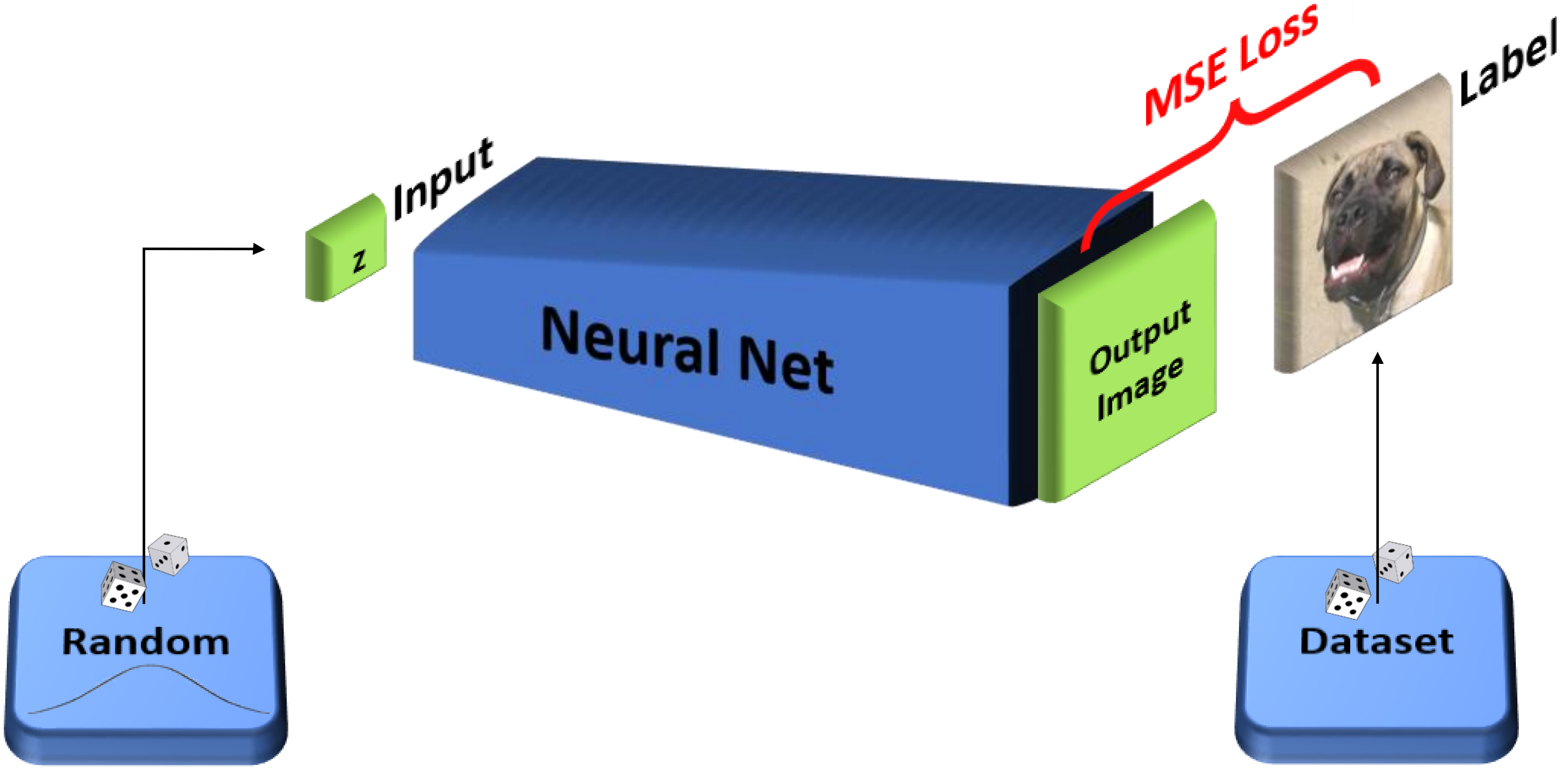
Slide credit: Assaf Socher

Generate data



Slide credit: Assaf Socher

How about this idea for a generative model?

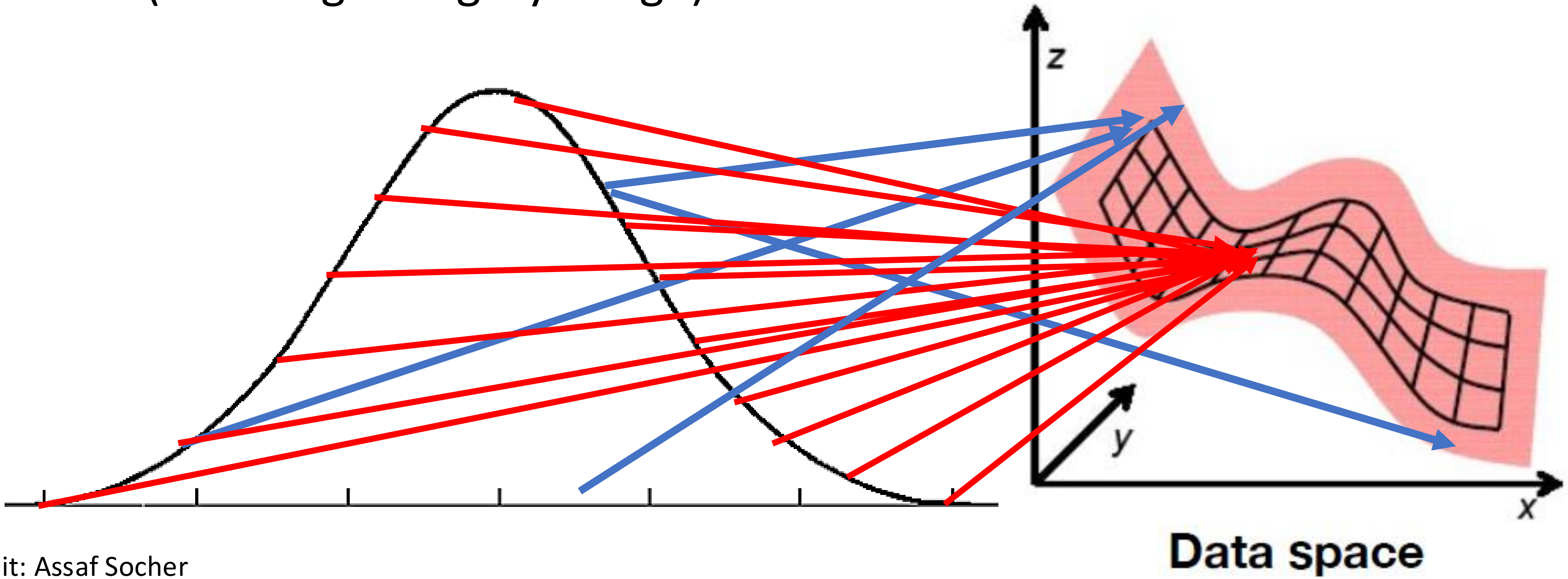


Slide credit: Assaf Socher

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)



Slide credit: Assaf Socher

Generative Adversarial Networks

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

Slide credit: Assaf Socher

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

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

Slide credit: Assaf Socher

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.

Slide credit: Assaf Socher

Slide credit: Assaf Socher

FAQ2: Why alternating?

- Gradients are meaningless whe
  game is unbalanced.

- Pre-train D? Negative examples?

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

GANs, Goodfellow 2014

Slide credit: Assaf Socher

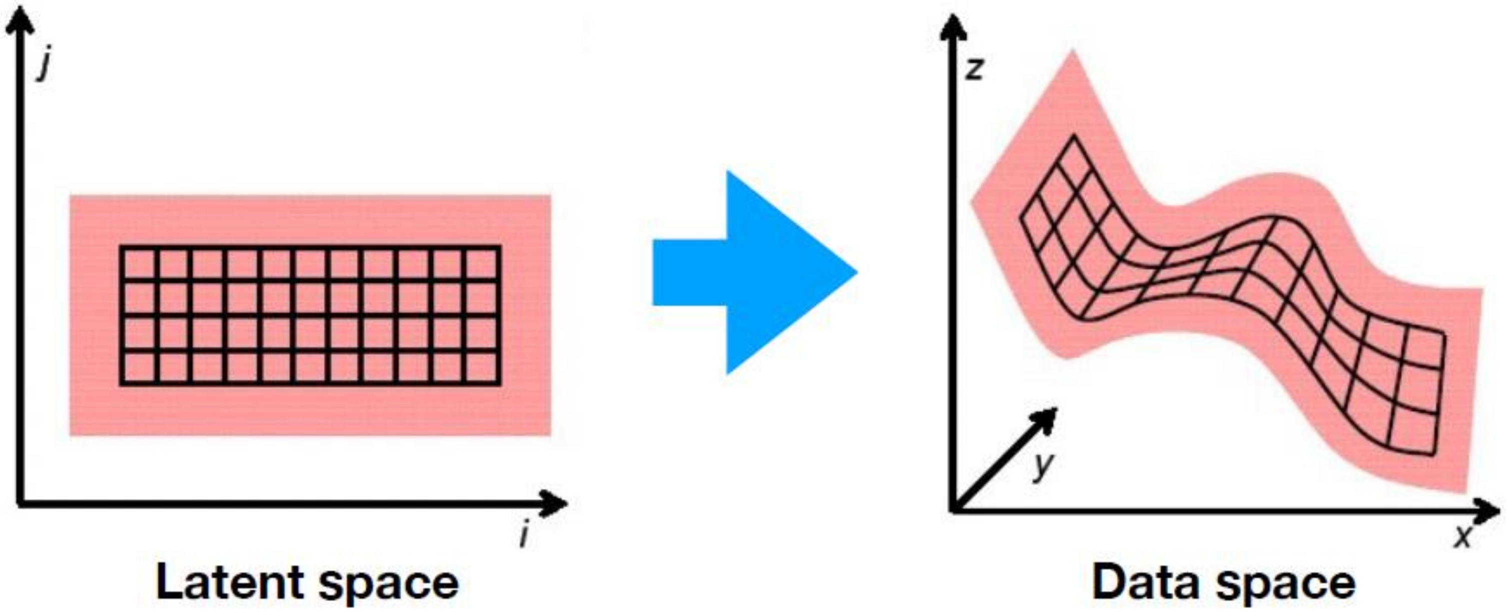
DCGAN Radford 2015

Slide credit: Assaf Socher

Latent space interpolation

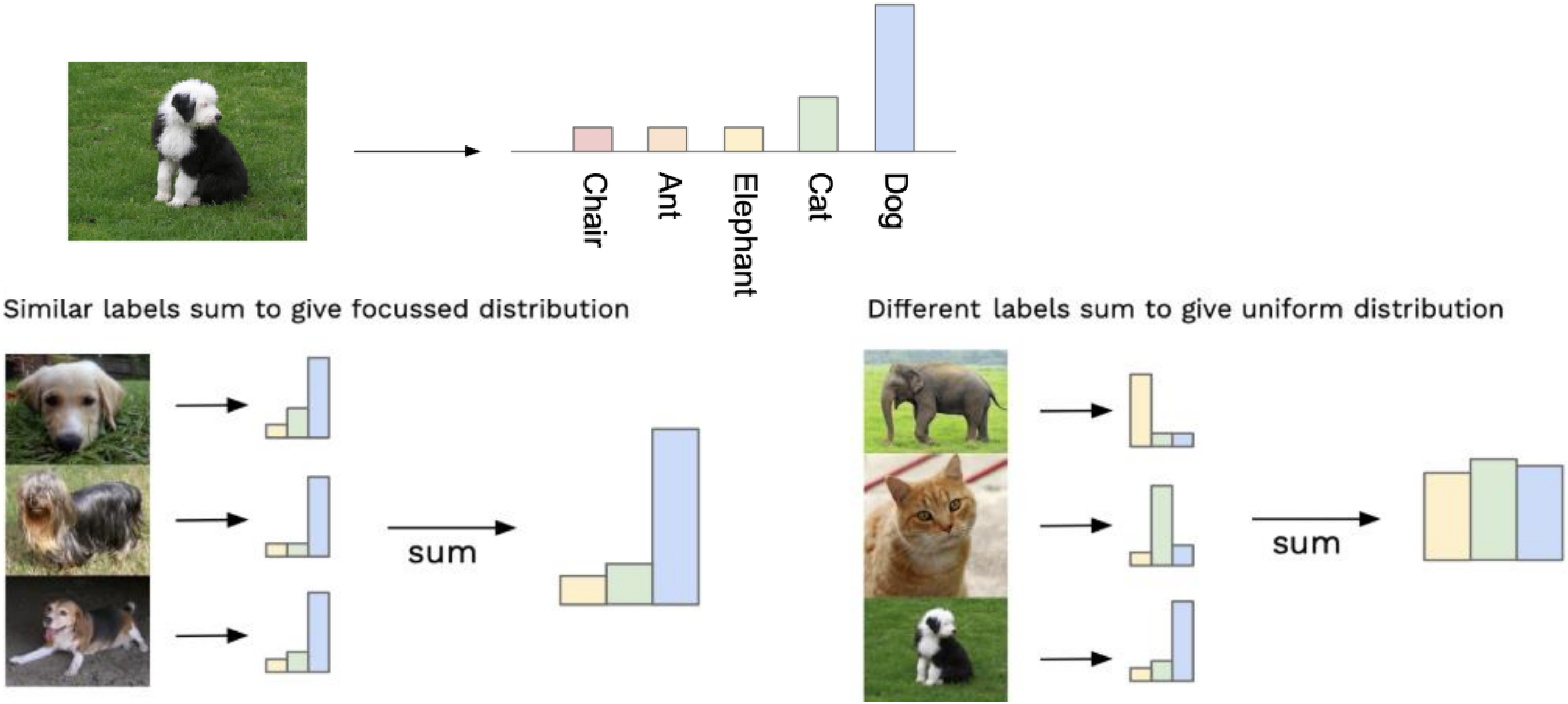
Slide credit: Assaf Socher

Why does it work?



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

Evaluation metrics: Inception score

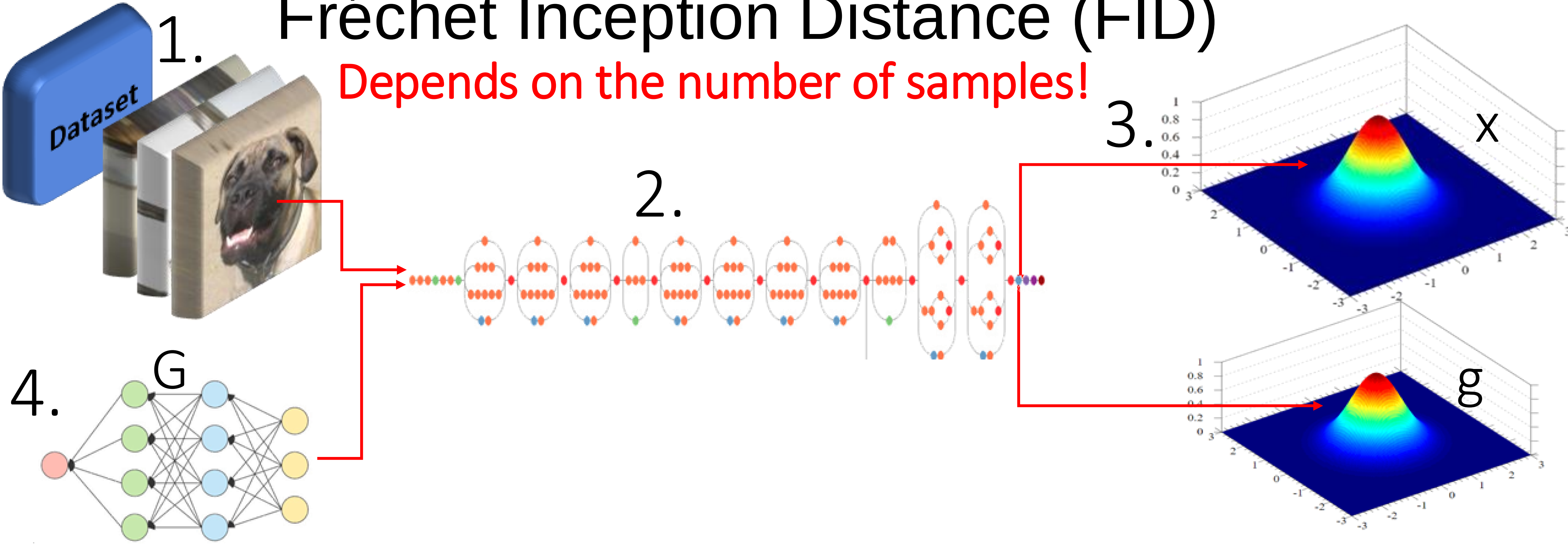


$$\mathrm{IS}(G) = \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}\left(p(y|\mathbf{x}) \parallel p(y) \right) \right)$$

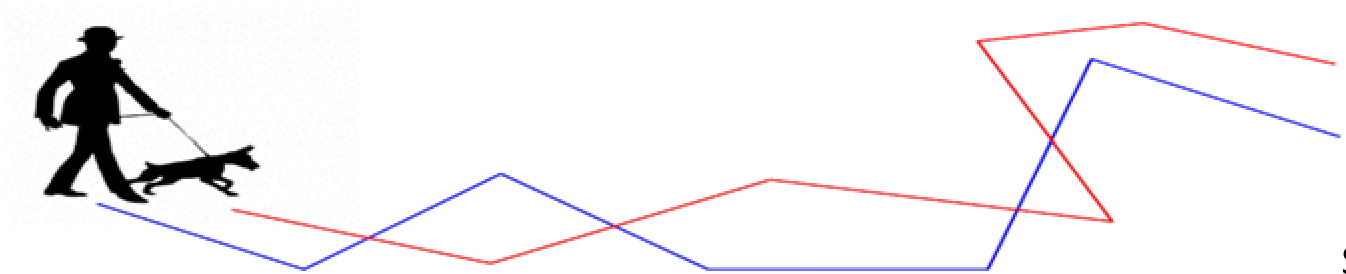
Slide credit: Assaf Socher

Fréchet Inception Distance (FID)

Depends on the number of samples!

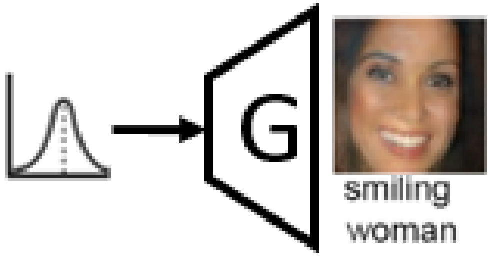


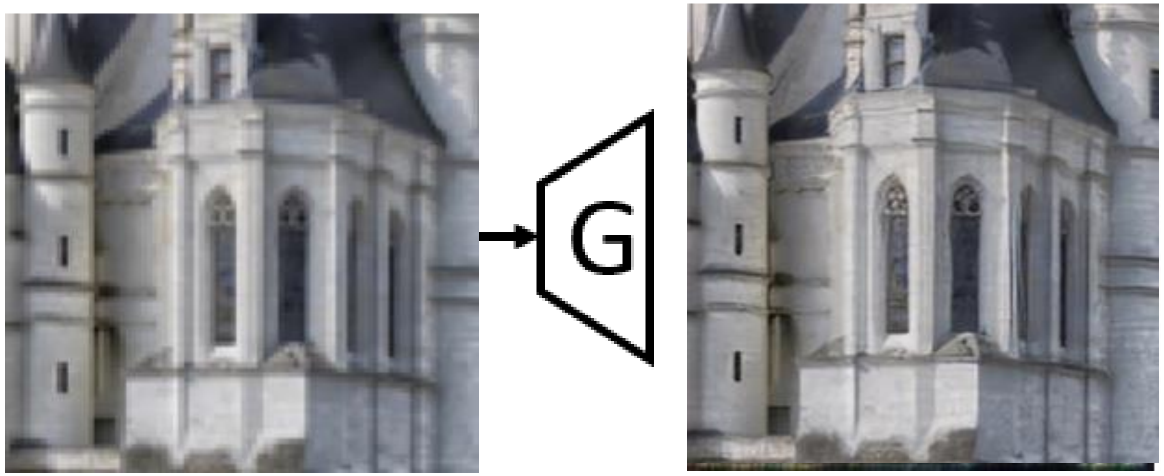
5. $$\mathtt{FID}(x,g) = ||\mu_x - \mu_g||_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}})$$



Slide credit: Assaf Socher

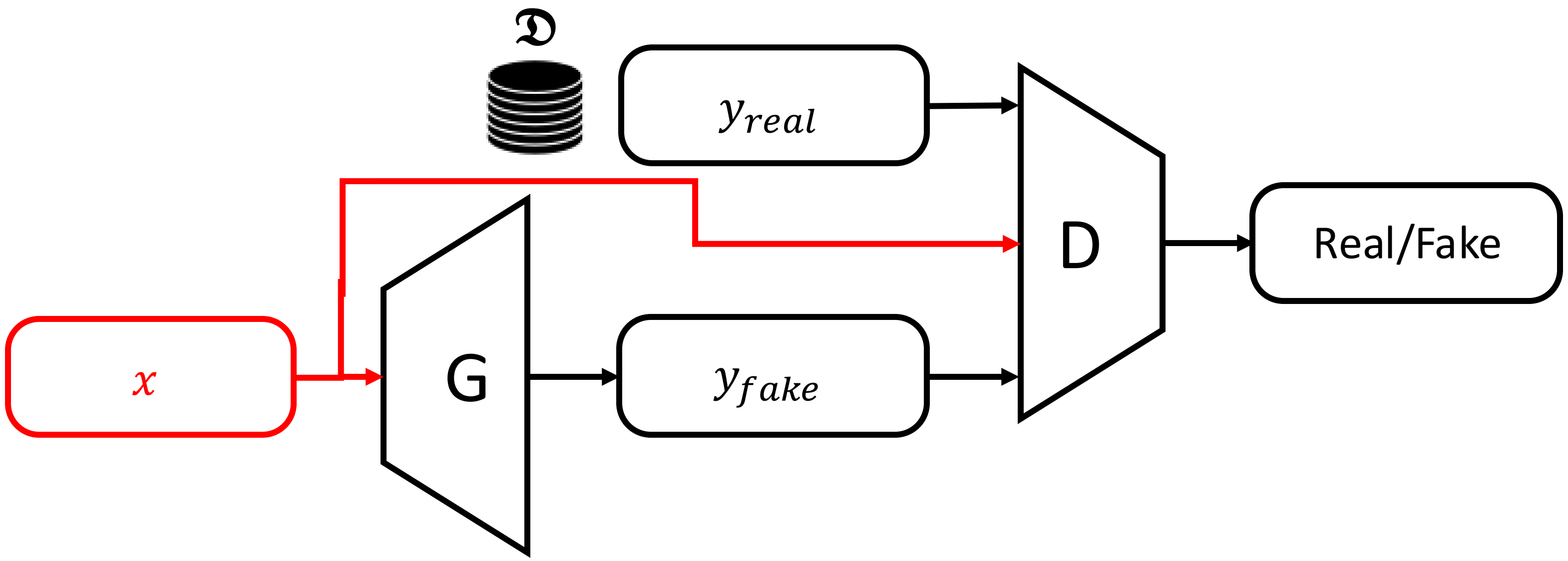
Image to Image translation





Isola et al. Nov2016

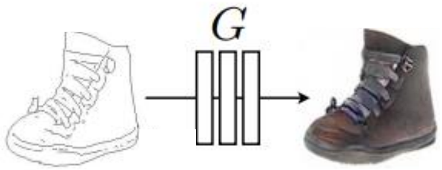
Conditional GAN



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

Slide by Sefi Bell-Kligler & Akhiad Bercovich

Pix2Pix



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

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

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

Slide by Sefi Bell-Kligler & Akhiad Bercovich

Isola et al. Nov2016

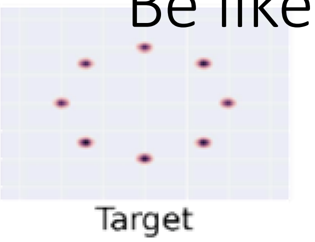
Slide credit: Assaf Socher

Training GANs is hard

- Stability

- Mode co

Training
GANs
Be like:

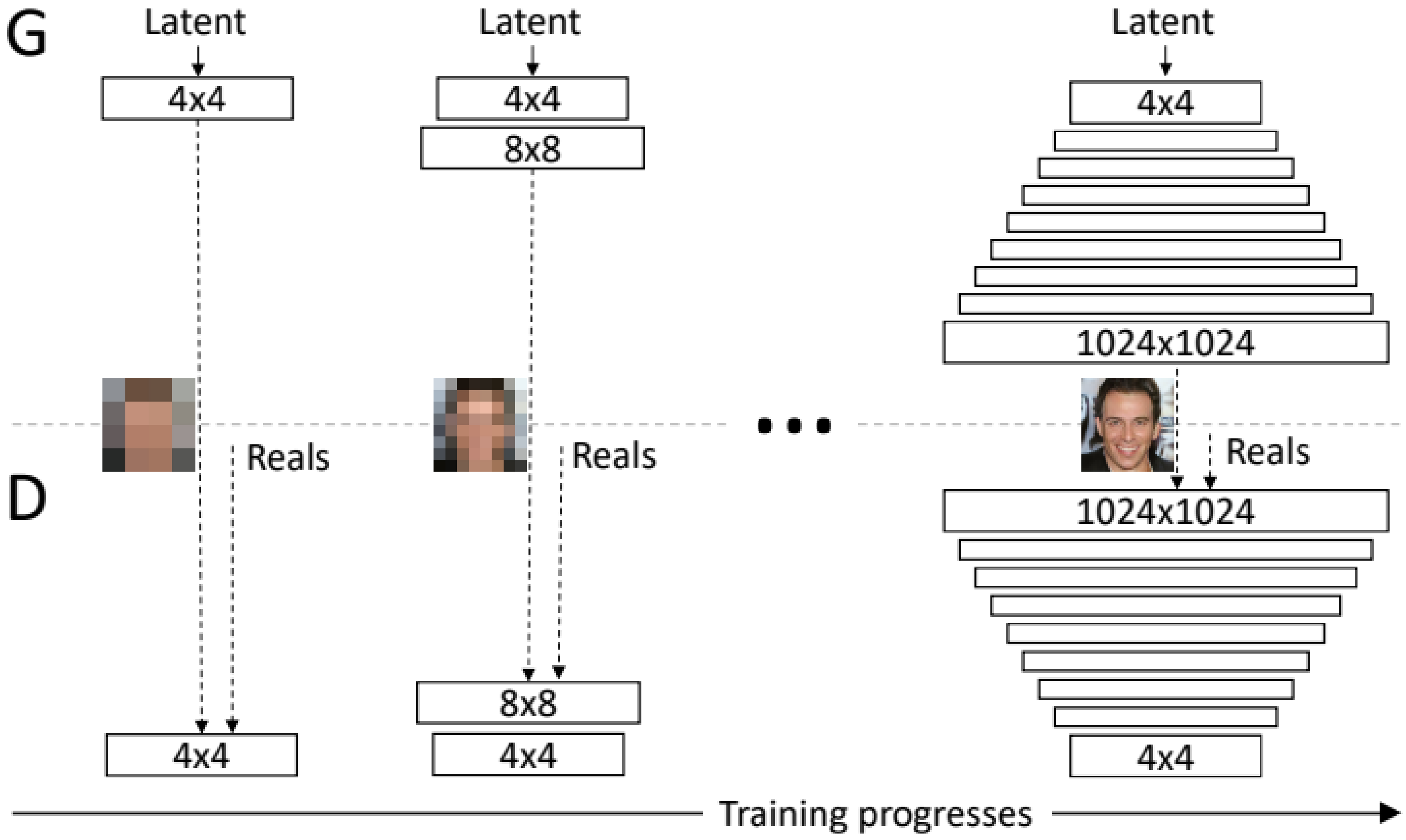


- GANs can over-train

- Lo

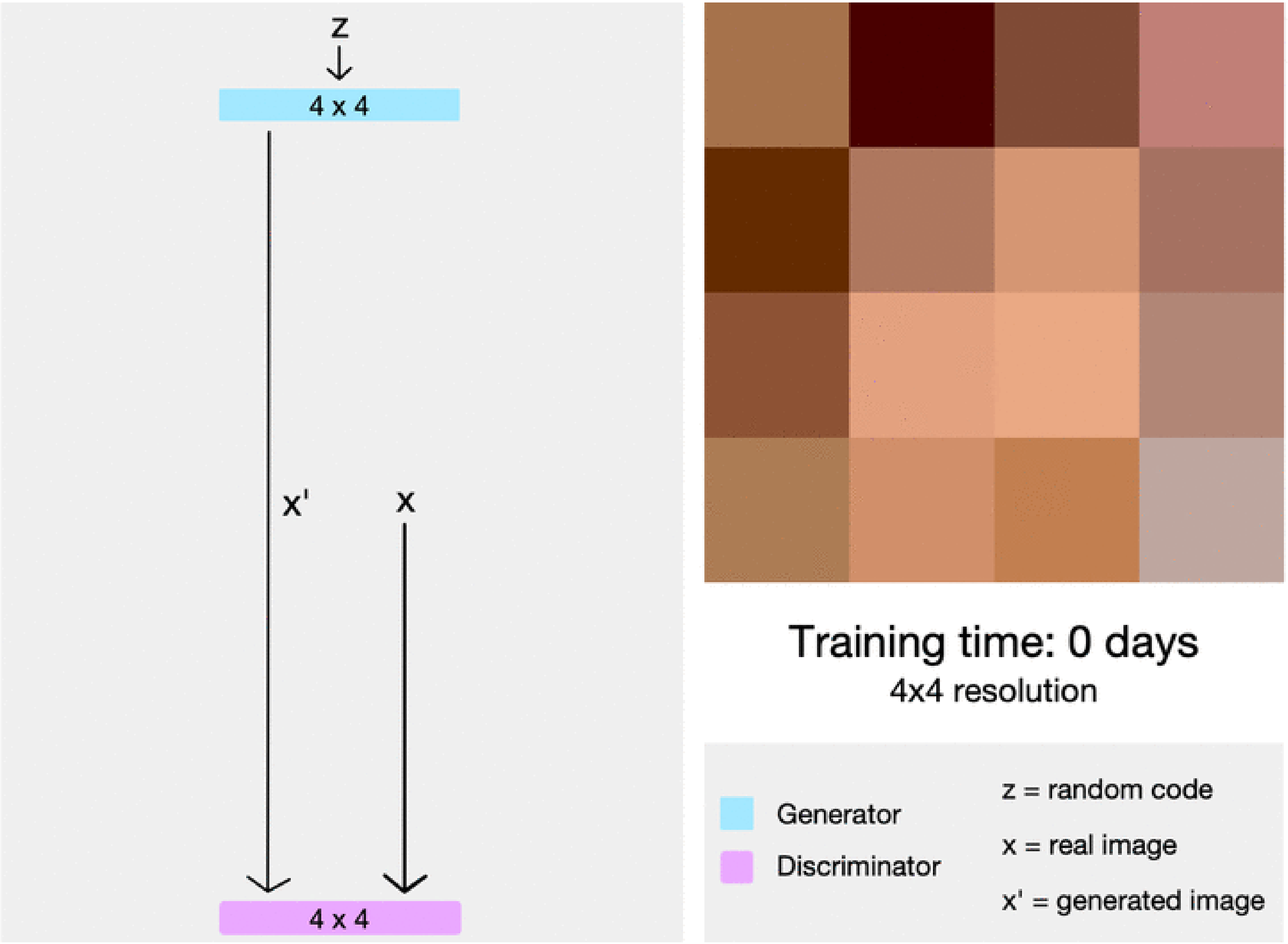
Slide credit: Assaf Socher

Progressive Grow



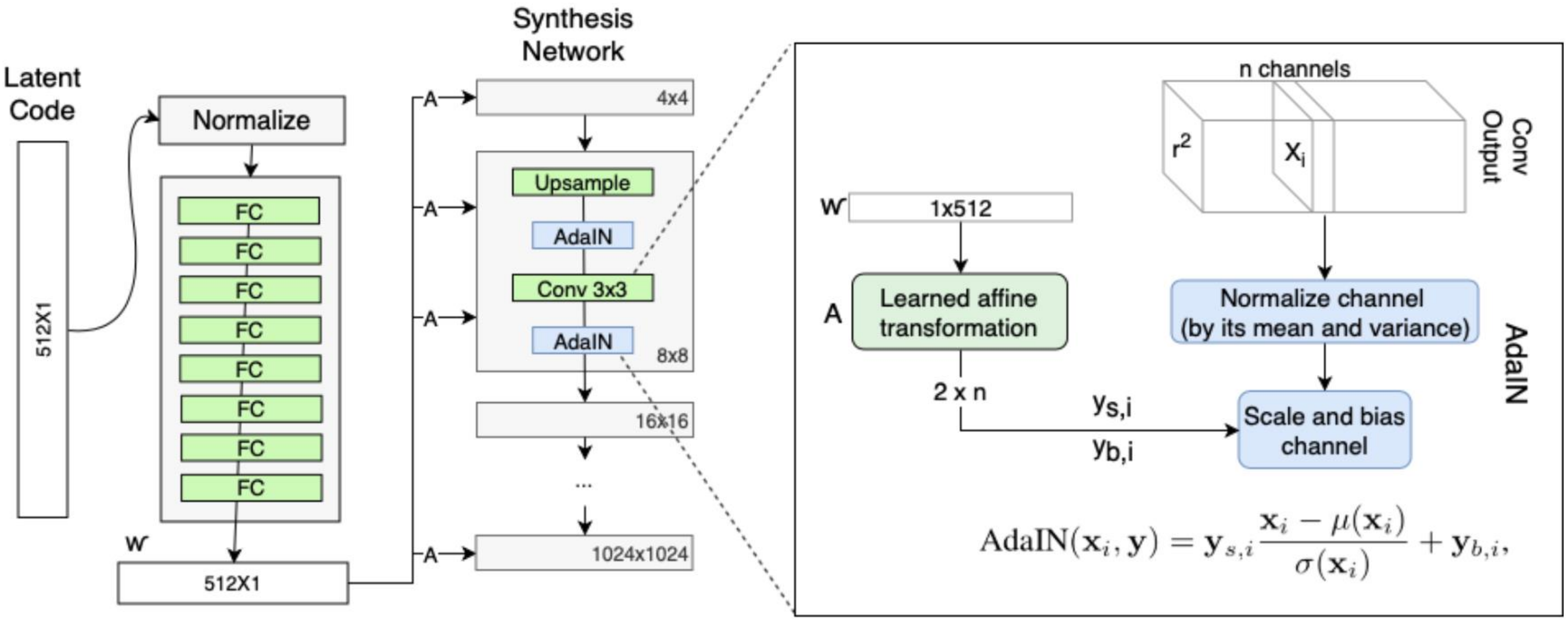
Progressive Growing of GAN, Karras et al., Feb2018

Slide by Sefi Bell-Kligler & Akhiad Bercovich



Progressive Growing of GAN, Karras et al., Feb2018

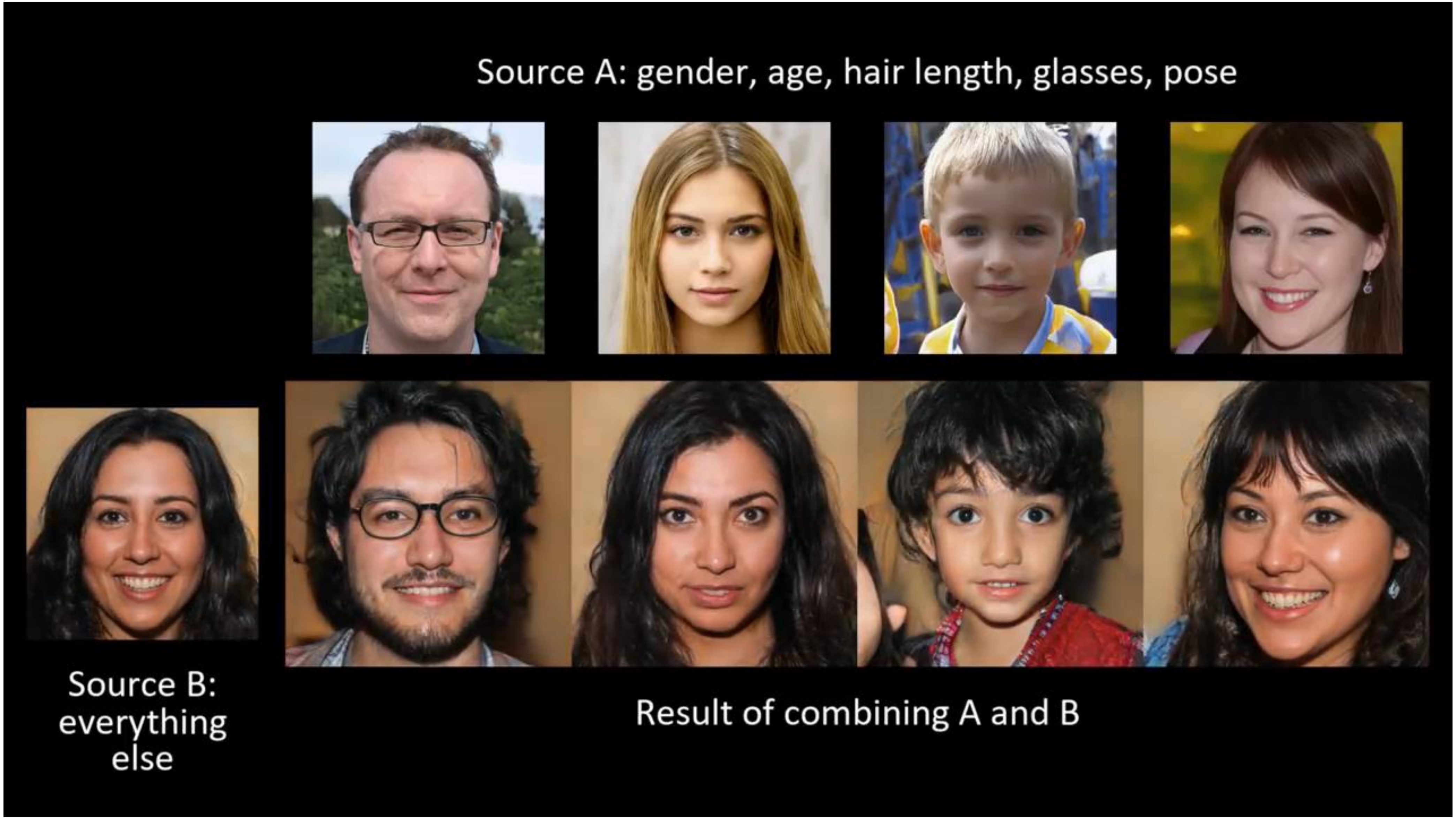
Style Modules (AdaIN)



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