10707: Deep Learning

Russ Salakhutdinov

Machine Learning Department
rsalakhu@cs.cmu.edu

Generative Adversarial Networks

Statistical Generative Models

Data

+

Model family, loss function, optimization algorithm, etc.

Prior Knowledge

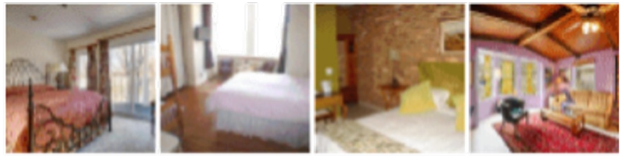
Learning

Image x

A probability distribution p(x)

probability p(x)

Sampling from p(x) **generates** new images:



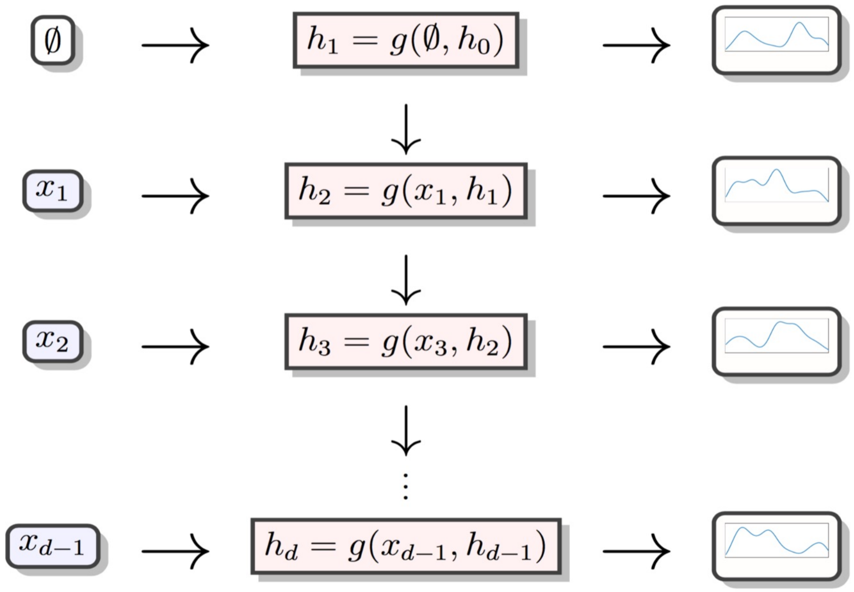
Grover and Ermon, DGM Tutorial

Fully Observed Models

- Density Estimation by Autoregression

$$p(x_1, \ldots, x_d) = \prod_{i=1}^{d} p(x_i | x_{i-1}, \ldots, x_1) \approx \prod_{i=1}^{d} p(x_i | g(x_{i-1}, \ldots, x_1))$$

Each conditional can be a deep neural network



$p(x_1)$

$p(x_2 | x_1)$

$p(x_3 | x_2, x_1)$

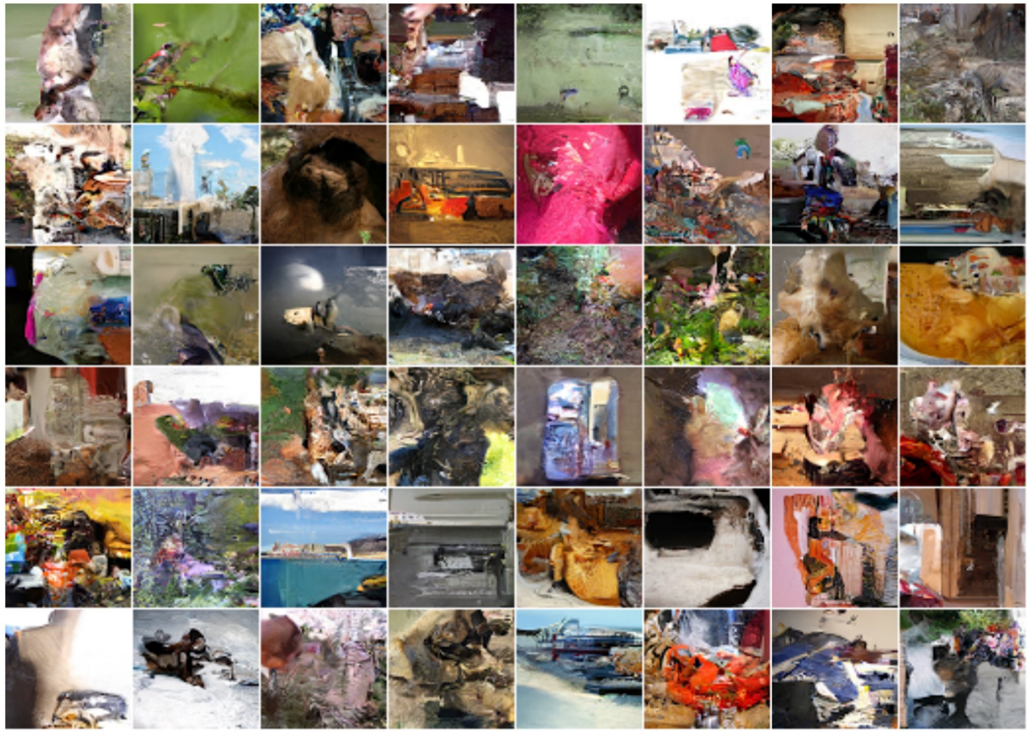
$p(x_d | x_d, \ldots, x_1)$

- Ordering of variables is crucial

NADE (Uria 2013), MADE (Germain 2017), MAF (Papamakarios 2017), PixelCNN (van den Oord, et al, 2016)

Fully Observed Models

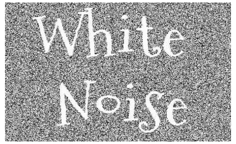
- Density Estimation by Autoregression

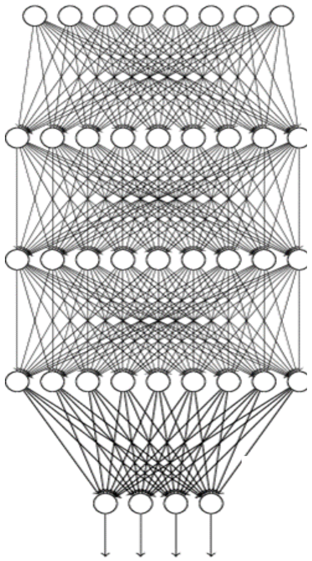


PixelCNN (van den Oord, et al, 2016)

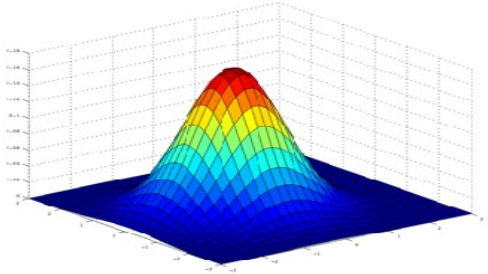
NADE (Uria 2013), MADE (Germain 2017), MAF (Papamakarios 2017), PixelCNN (van den Oord, et al, 2016)

Deep Directed Generative Models



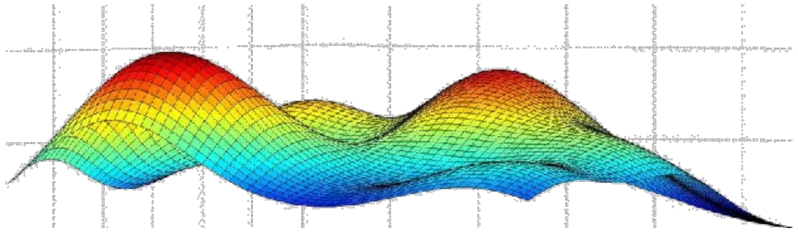


Code Z



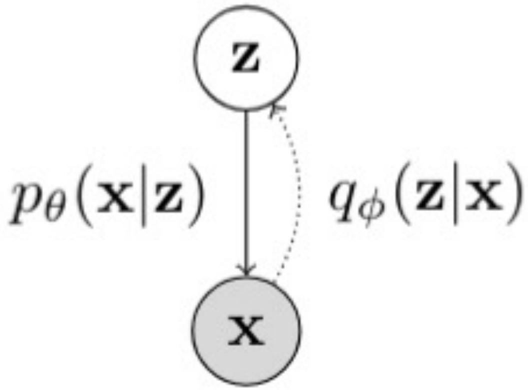
- Recognition
- Bottom-up
- Q(z|x)

- Generative
- Top-Down
- P(x|z)



D_{real}

- Latent Variable Models

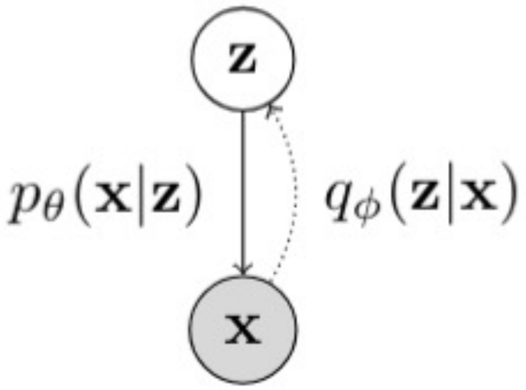


$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$

- Conditional distributions are parameterized by deep neural networks

Directed Deep Generative Models

- Directed Latent Variable Models with Inference Network



- Maximum log-likelihood objective

$$\max_\theta \sum_{\mathbf{x}\in\mathcal{D}} \log p_\theta(\mathbf{x})$$

- Marginal log-likelihood is intractable:

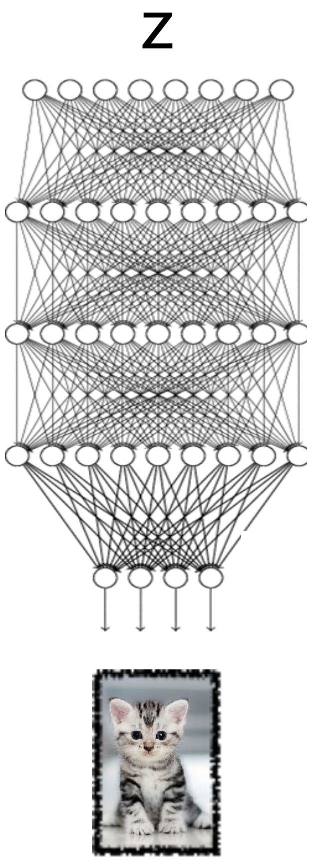
$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$

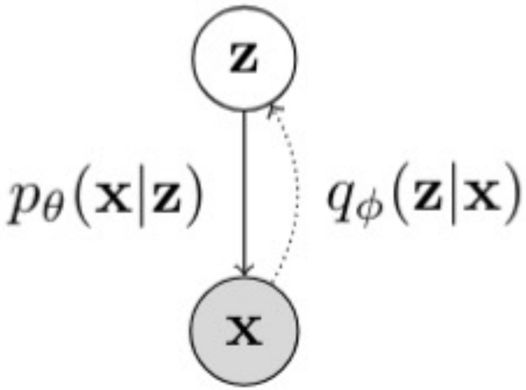
- Key idea: Approximate true posterior p(z|x) with a simple, tractable distribution q(z|x) (inference/recognition network).

Grover and Ermon, DGM Tutorial

Variational Autoencoders (VAEs)

- Single stochastic (Gaussian) layer, followed by many deterministic layers





$$p(\mathbf{z}) = \mathcal{N}(0, I)$$

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}\big(\mu(\mathbf{z}, \theta), \Sigma(\mathbf{z}, \theta)\big)$$

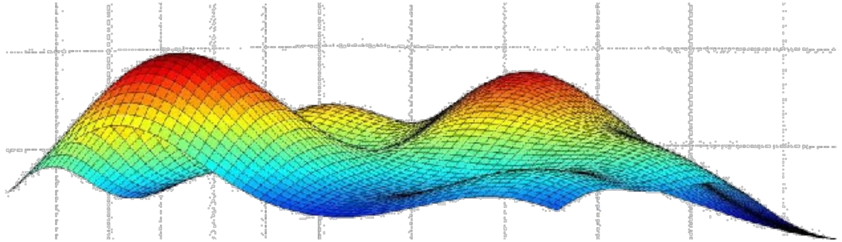
Deep neural network parameterized by θ.
(Can use different noise models)

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mu(\mathbf{x}, \phi), \Sigma(\mathbf{x}, \phi)\big)$$

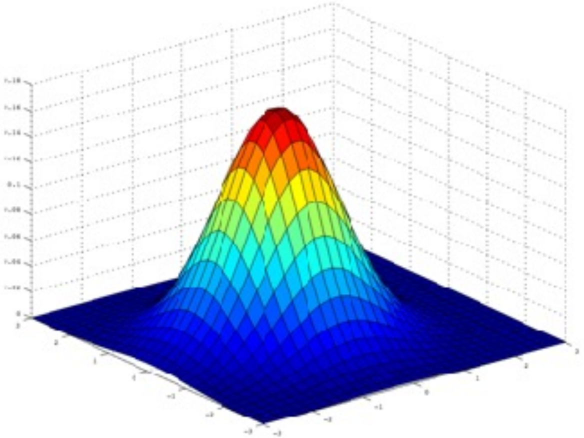
Deep neural network parameterized by ϕ.

Generative Adversarial Networks (GAN)

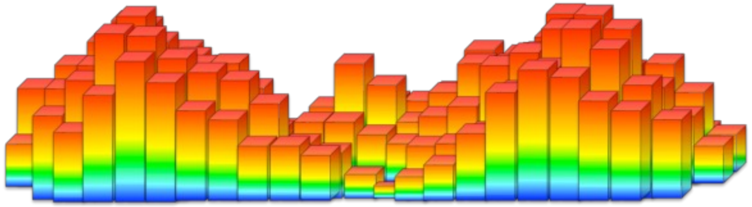
- Implicit generative model for an unknown target density $p(x)$
- Converts sample from a known noise density $p_{\mathcal{Z}}(z)$ to the target $p(x)$



Unknown target density $p(x)$ of data over domain $\mathcal{X}$, e.g. $\mathbb{R}^{32\times 32}$



Noise density $p_{\mathcal{Z}}(z)$ over space $\mathcal{Z}$



Distribution of generated samples should follow target density $p(x)$

Slide Credit: Manzil Zaheer]

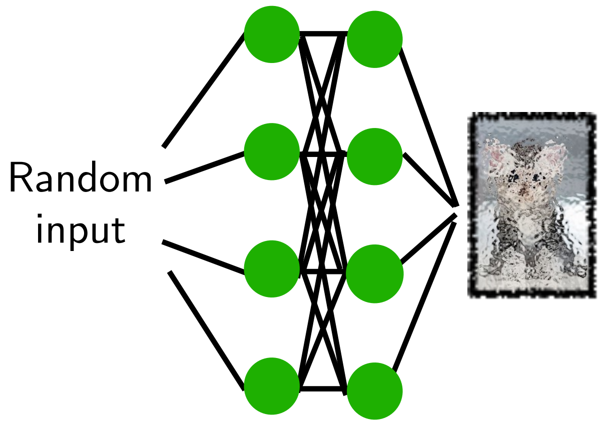
Goodfellow et al, 2014

GAN Formulation

- GAN consists of two components

Generator

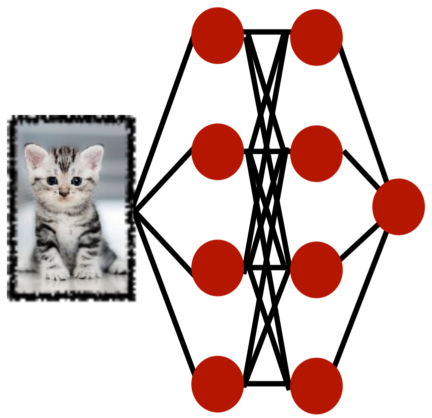
$G : \mathcal{Z} \to \mathcal{X}$



Goal: Produce samples indistinguishable from true data

Discriminator

$D : \mathcal{X} \to \mathbb{R}$



Goal: Distinguish true and generated data apart

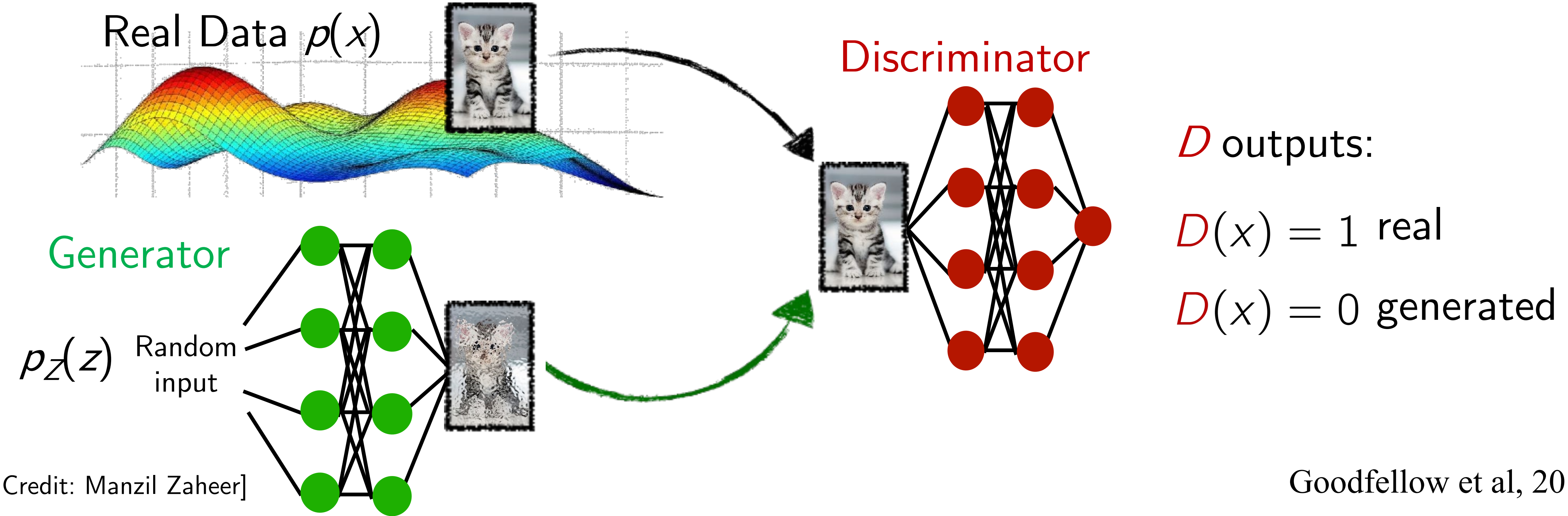
Slide Credit: Manzil Zaheer]

Goodfellow et al, 2014

GAN Formulation: Discriminator

- Discriminator's objective: Tell real and generated data apart like a classifier

$$\max_{D} \mathbb{E}_{x\sim p}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_Z}\left[\log\left(1 - D(G(z))\right)\right]$$



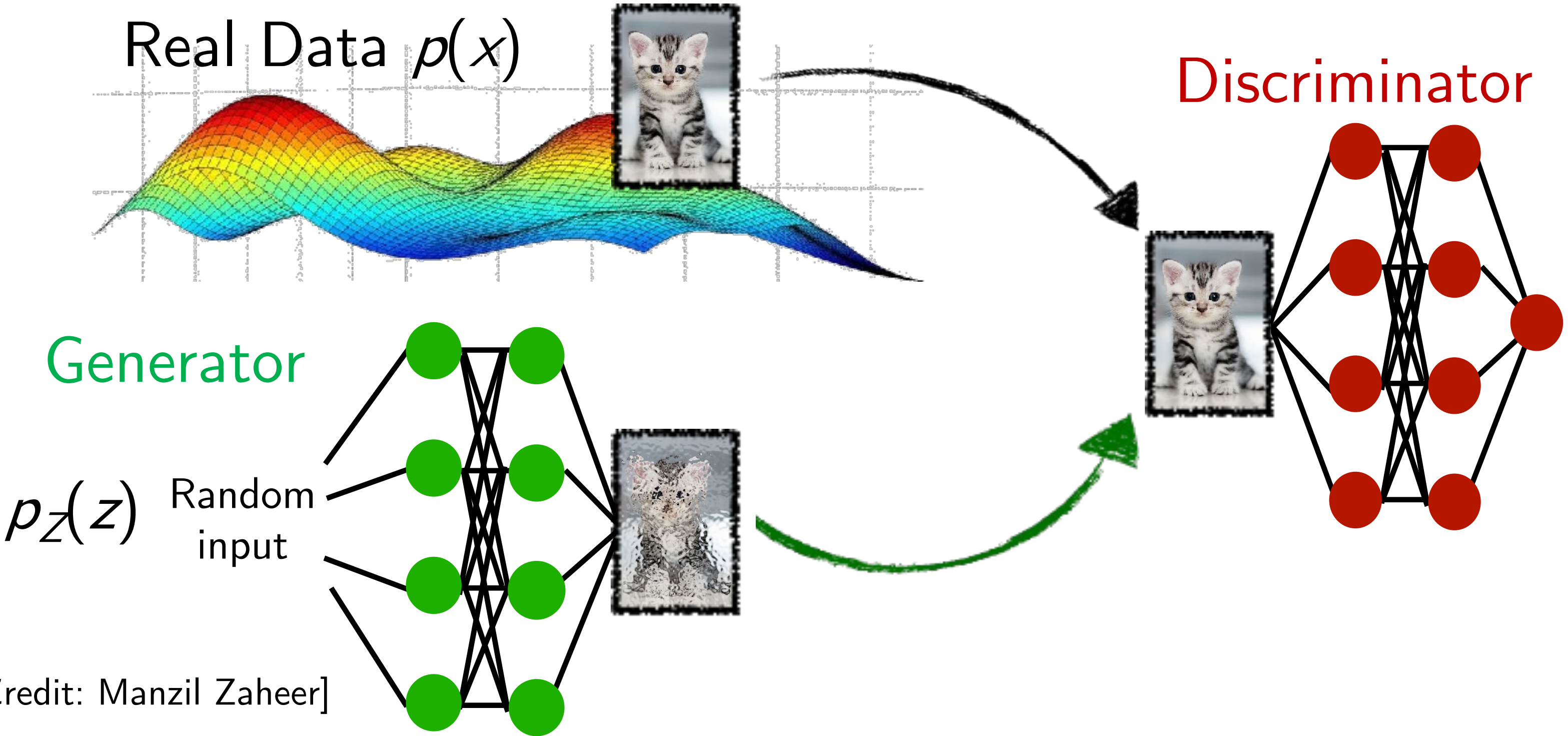
Slide Credit: Manzil Zaheer]

Goodfellow et al, 2014

GAN Formulation: Generator

- Generator's objective: Fool the best discriminator

$$\min_{G} \max_{D} \mathbb{E}_{x \sim p}\left[\log D(x)\right] + \mathbb{E}_{z \sim p_Z}\left[\log\left(1 - D(G(z))\right)\right]$$



D outputs:

$D(x) = 1$ real

$D(x) = 0$ generated

Slide Credit: Manzil Zaheer]

Goodfellow et al, 2014

GAN Formulation: Optimization

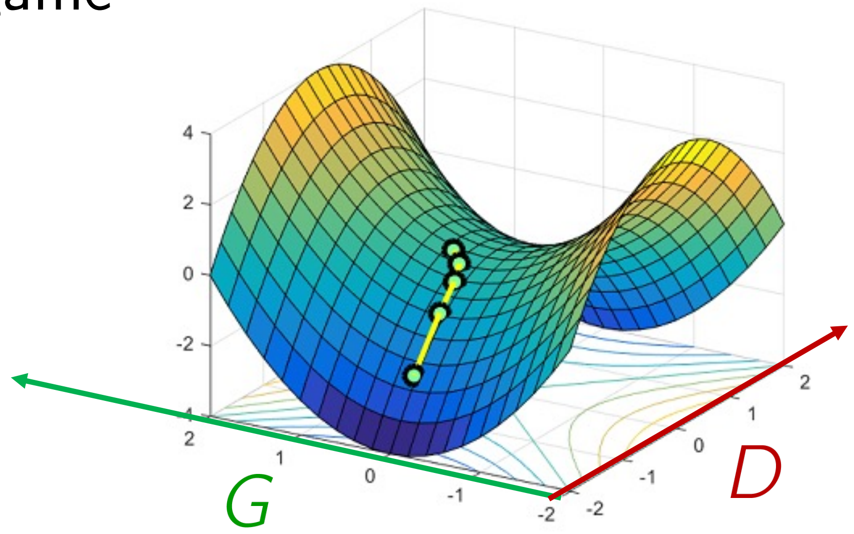
- Overall GAN optimization

$$\min_{G} \max_{D} V(G, D) = \mathbb{E}_{x \sim p}\left[\log D(x)\right] + \mathbb{E}_{z \sim p_Z}\left[\log\left(1 - D(G(z))\right)\right]$$

- The generator-discriminator are iteratively updated using SGD to find "equilibrium" of a "min-max objective" like a game

$$G \leftarrow G - \eta_G \nabla_G V(G, D)$$

$$D \leftarrow D - \eta_D \nabla_D V(G, D)$$



Slide Credit: Manzil Zaheer]

Distributional perspective - Discriminator

$$\min_G \max_D V(G, D) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]$$

- For a fixed generator, discriminator is maximizing negative cross entropy
- Optimal discriminator is given by:

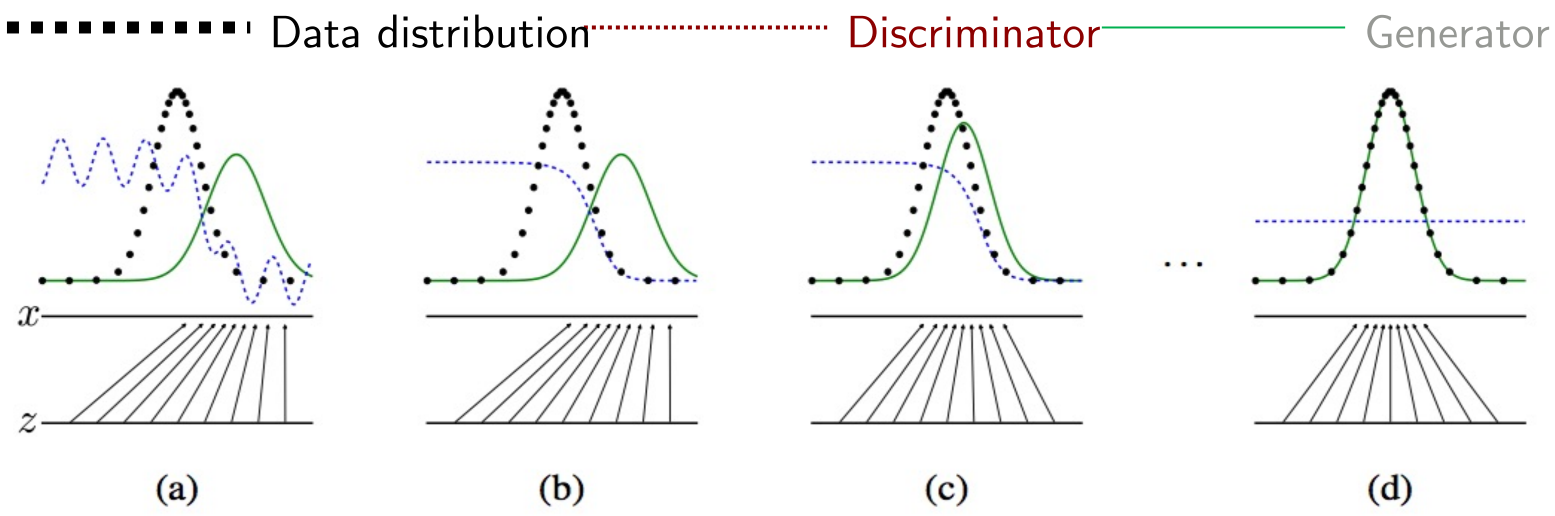
$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

Goodfellow et al, 2014

A minimax learning objective

- During learning, generator and discriminator are updated alternatively

$$\min_{\theta} \max_{\phi} V(G_\theta, D_\phi) = \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}}[\log D_\phi(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D_\phi(G_\theta(\mathbf{z})))]$$



Current state | Update discriminator | Update generator | Convergence

Goodfellow et al, 2014

Evaluation

- Likelihoods may not be defined or tractable
- Directed model permits ancestral sampling
  - For labelled datasets, metrics such as inception scores quantify sample diversity and quality using pretrained classifiers

Wu et al., 2017, Grover et al., 2018, Salimans et al., 2016, Heusel et al., 2018

Mode Collapse

- In practice, GANs suffer from mode collapse

Arjovsky et al., 2017

Wasserstein GAN

- WGAN optimization

$$\min_{G} \max_{D} W(G, D) = \mathbb{E}_{x \sim p}\left[D(x)\right] - \mathbb{E}_{z \sim p_Z}\left[D(G(z))\right]$$

- Difference in expected output on real vs. generated images
  - Generator attempts to drive objective ≈ 0
- More stable optimization

D outputs:

$D(x) = 1$ real

$D(x) = 0$ generated

Compare to training DBMs

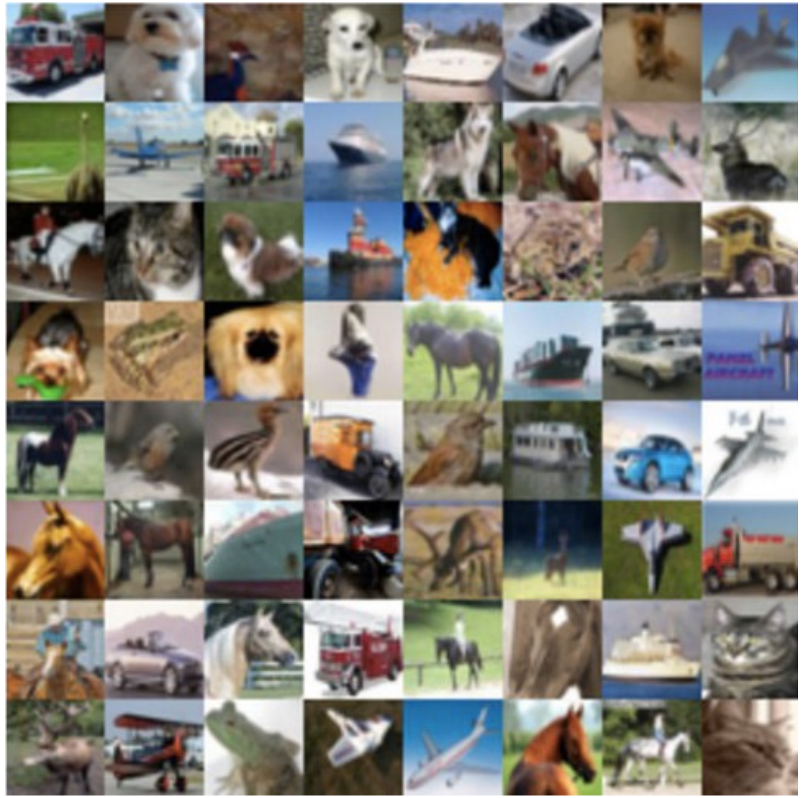
$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}] - \mathbb{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}]$$

Arjovsky et al., 2017

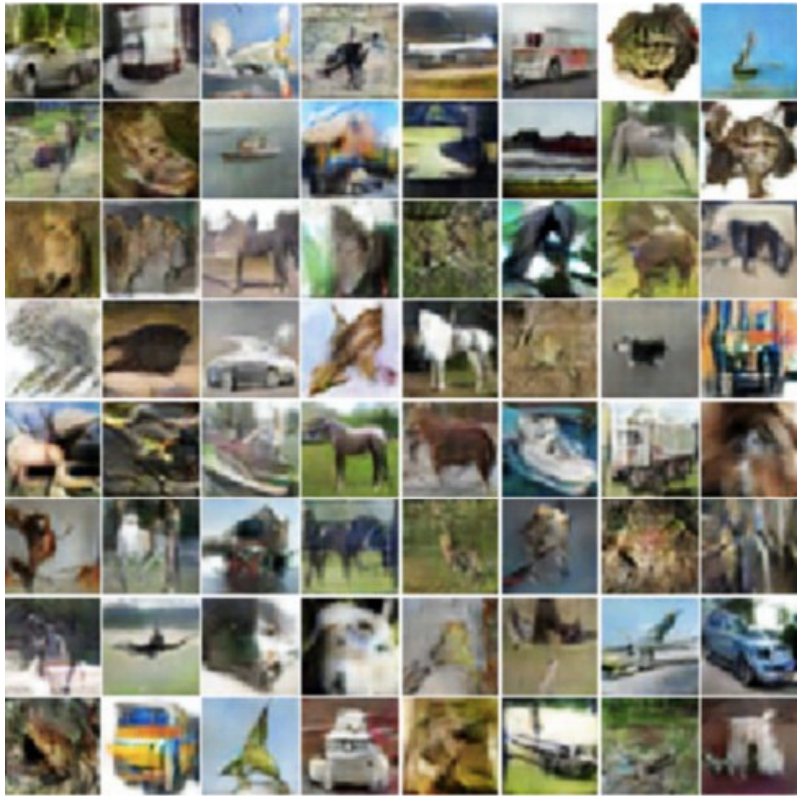
LSUN Bedroom: Samples

Radford et al., 2015

CIFAR Dataset



Training



Samples

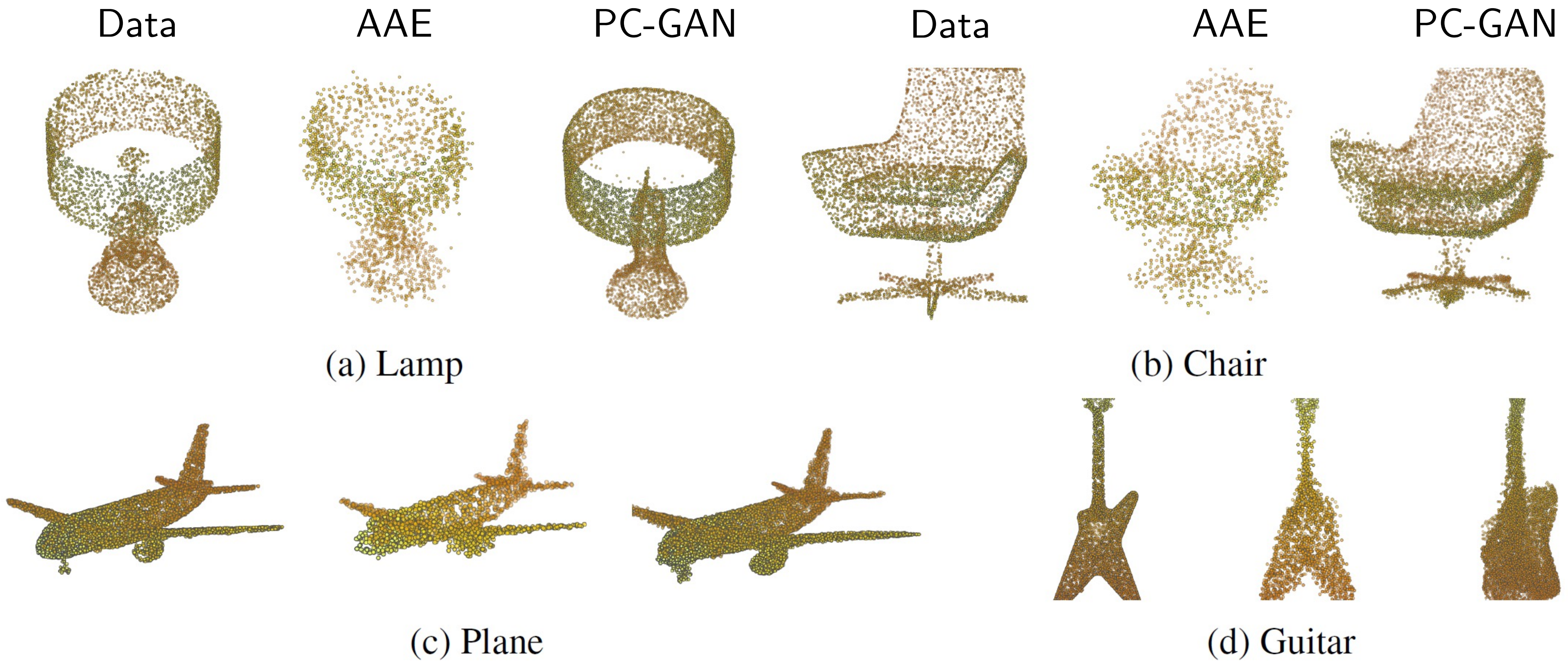
Salimans et. al., 2016

ImageNet: Cherry-Picked Samples

- Open Question: How can we quantitatively evaluate these models!

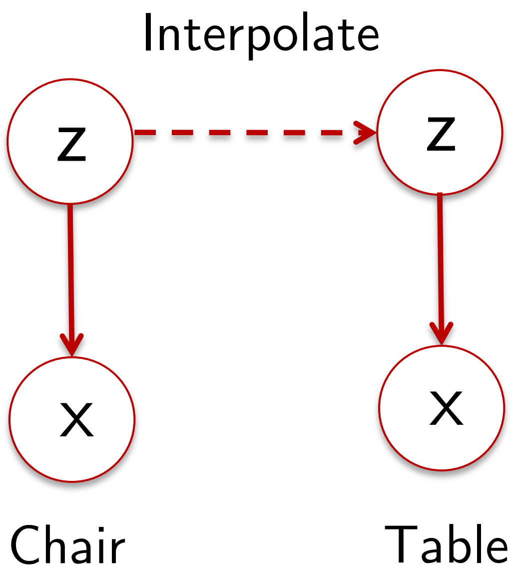
Slide Credit: Ian Goodfellow

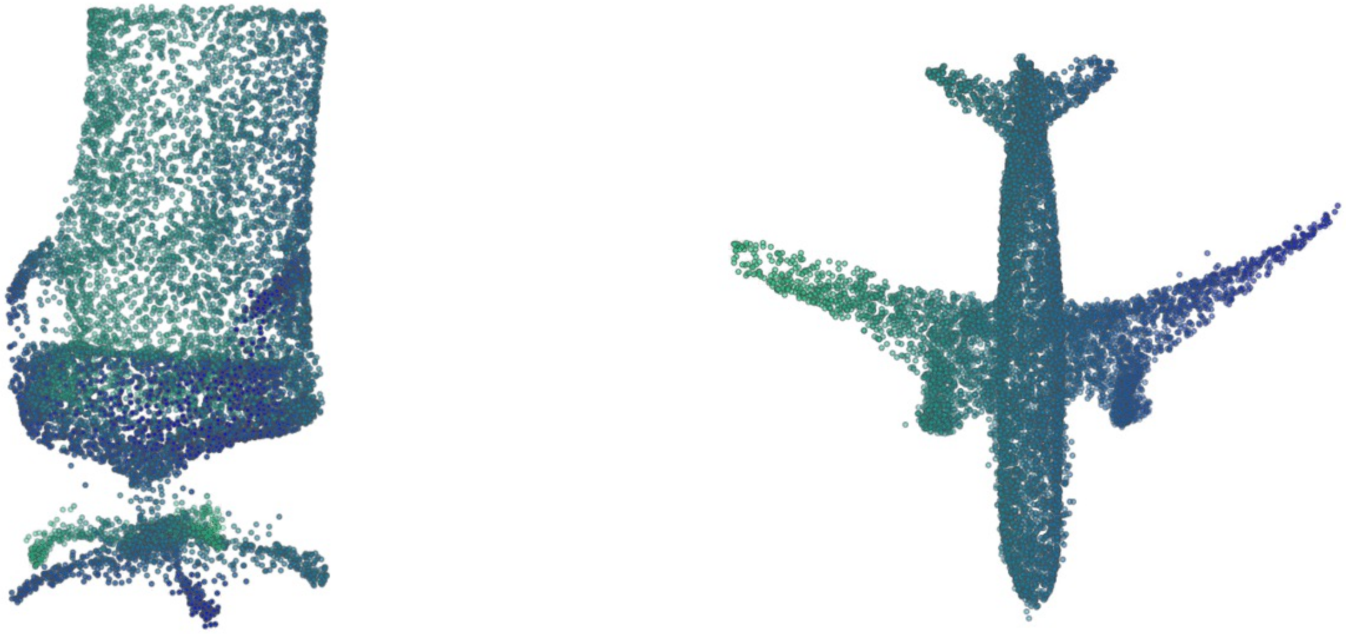
Modelling Point Cloud Data



Zaheer et al. Point Cloud GAN 2018

Interpolation in Latent Space

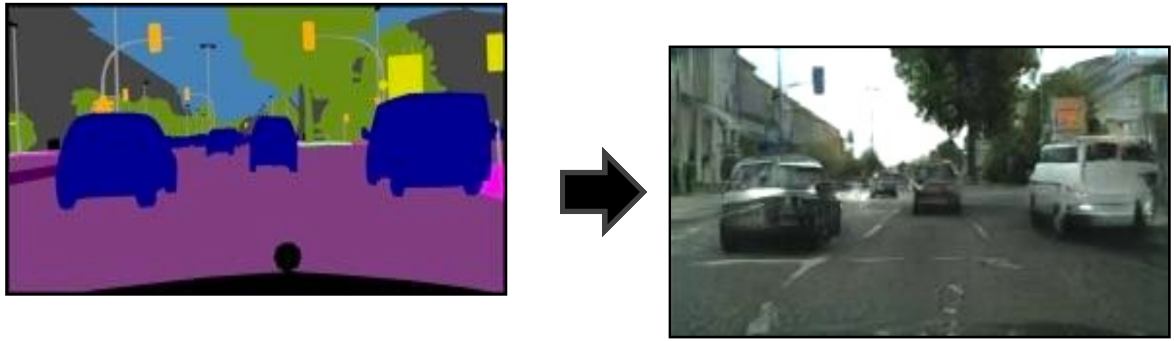




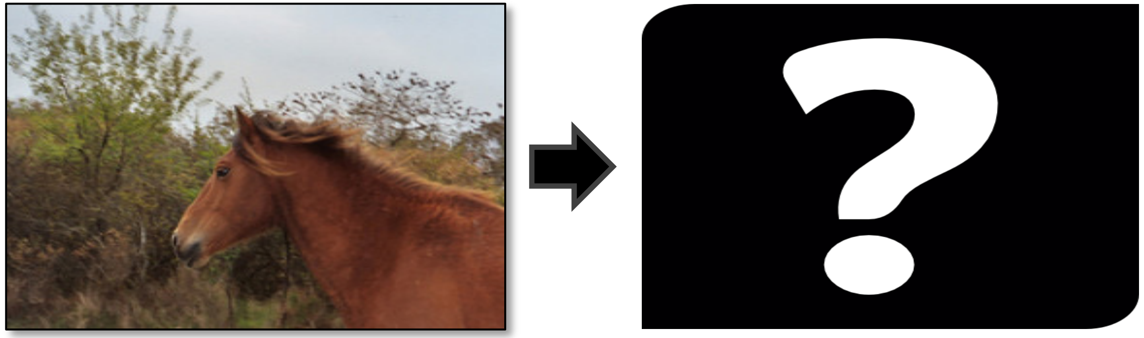
Zaheer et al. Point Cloud GAN 2018

Cycle GAN

Paired



Label photo: per-pixel labeling



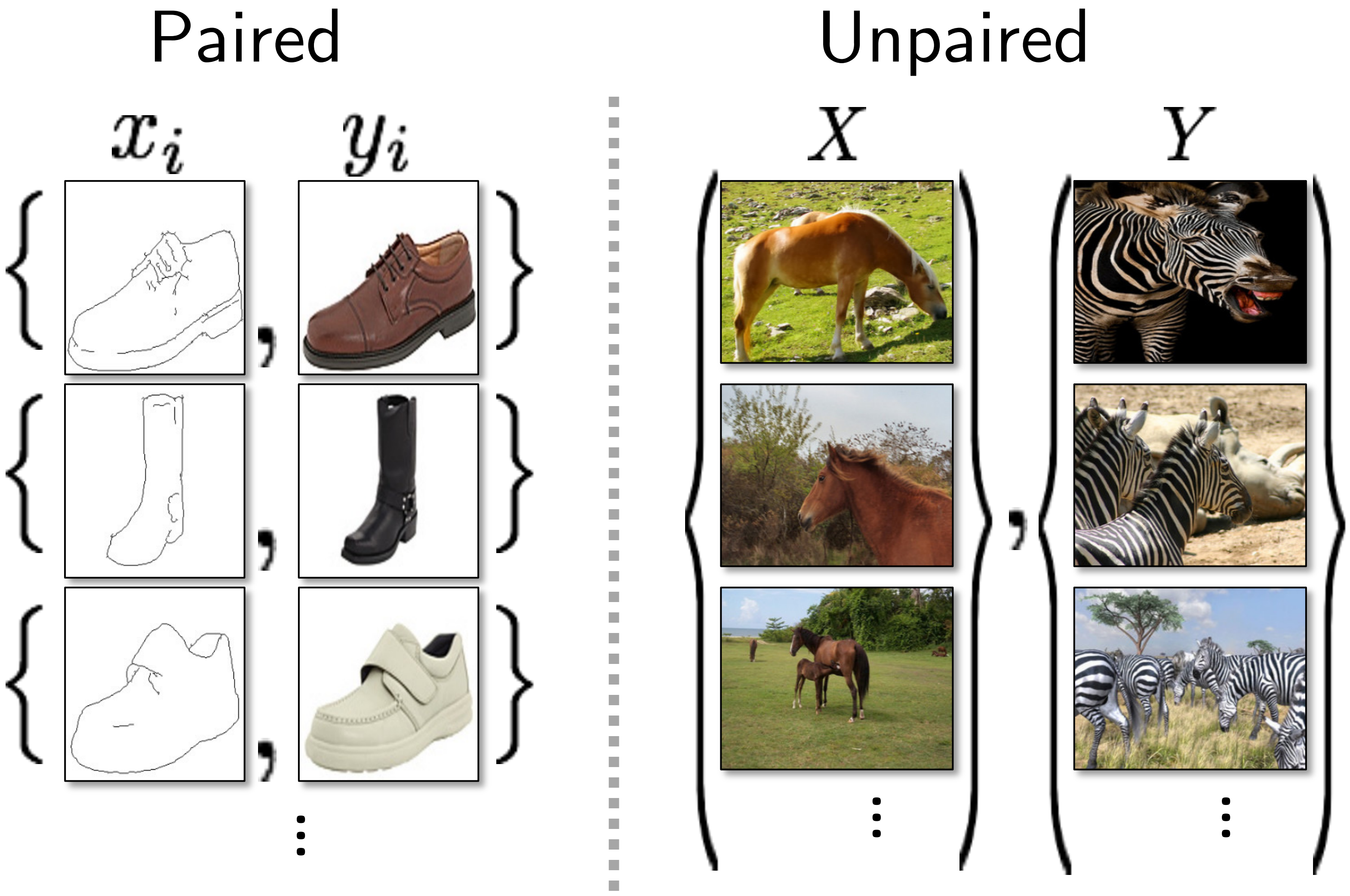
Horse zebra: how to get zebras?

- Expensive to collect pairs.
- Impossible in many scenarios.

[Zhu*, Park*, Isola, and Efros, ICCV 2017]

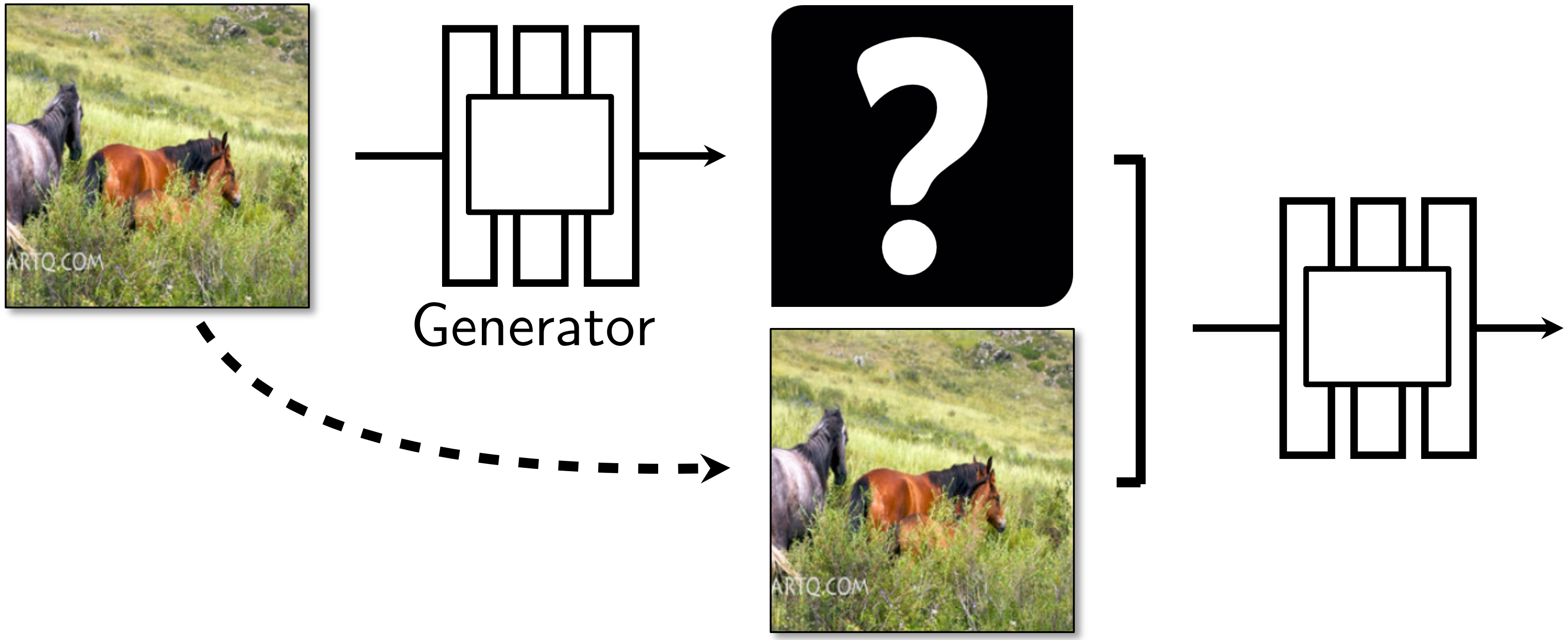
Slide credit: Jun-Yan Zhu

Cycle GAN



Slide credit: Jun-Yan Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]



No input-output pairs!

Slide credit: Jun-Yan Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]

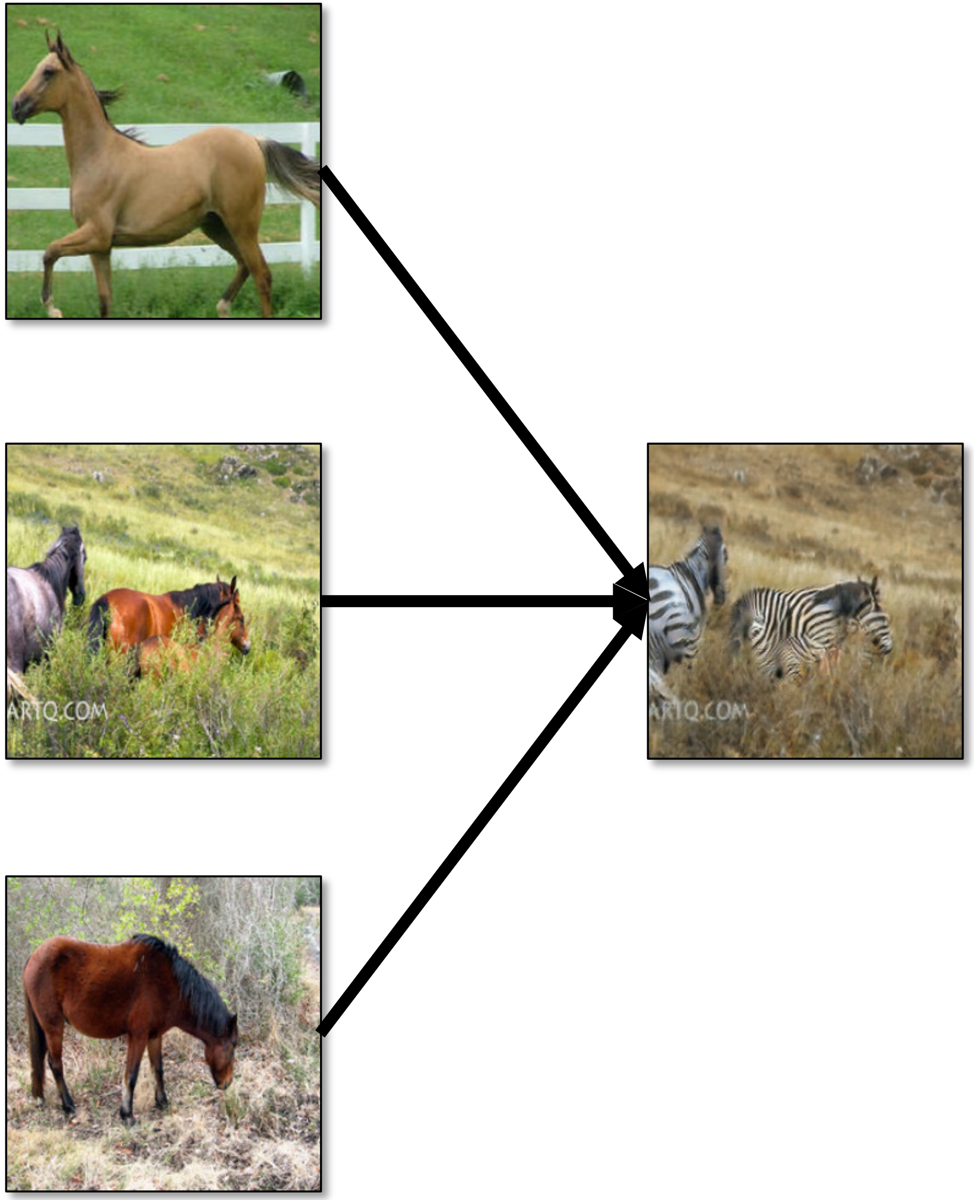
Slide credit: Jun-Yan Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]

GANs doesn't force output to correspond to input

Slide credit: Jun-Yan Zhu

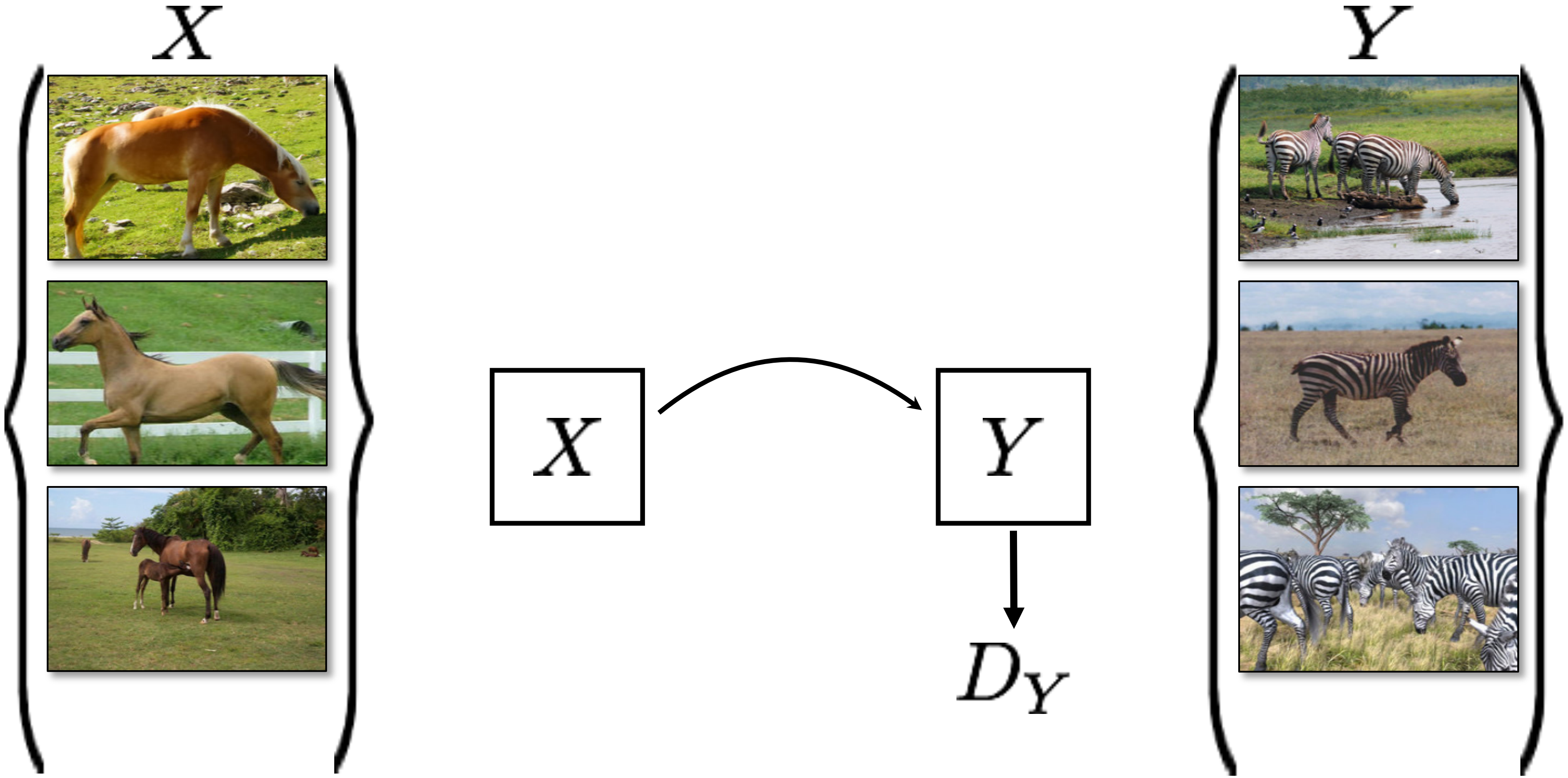
[Zhu*, Park*, Isola, and Efros, ICCV 2017]



mode collapse

Slide credit: Jun-Yan Zhu

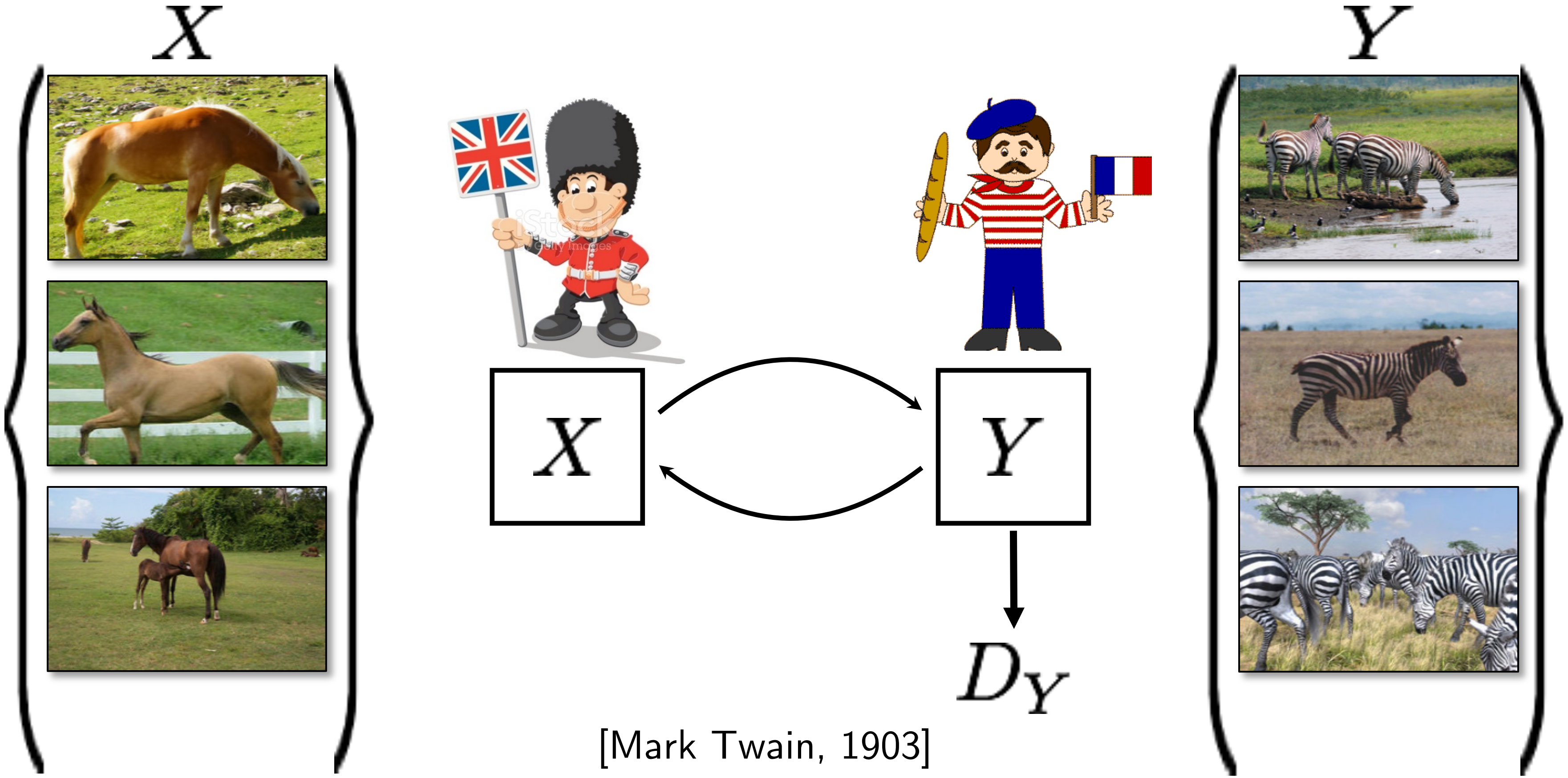
Cycle Consistent Adversarial Networks



Slide credit: Jun-Yan Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]

Cycle Consistent Adversarial Networks

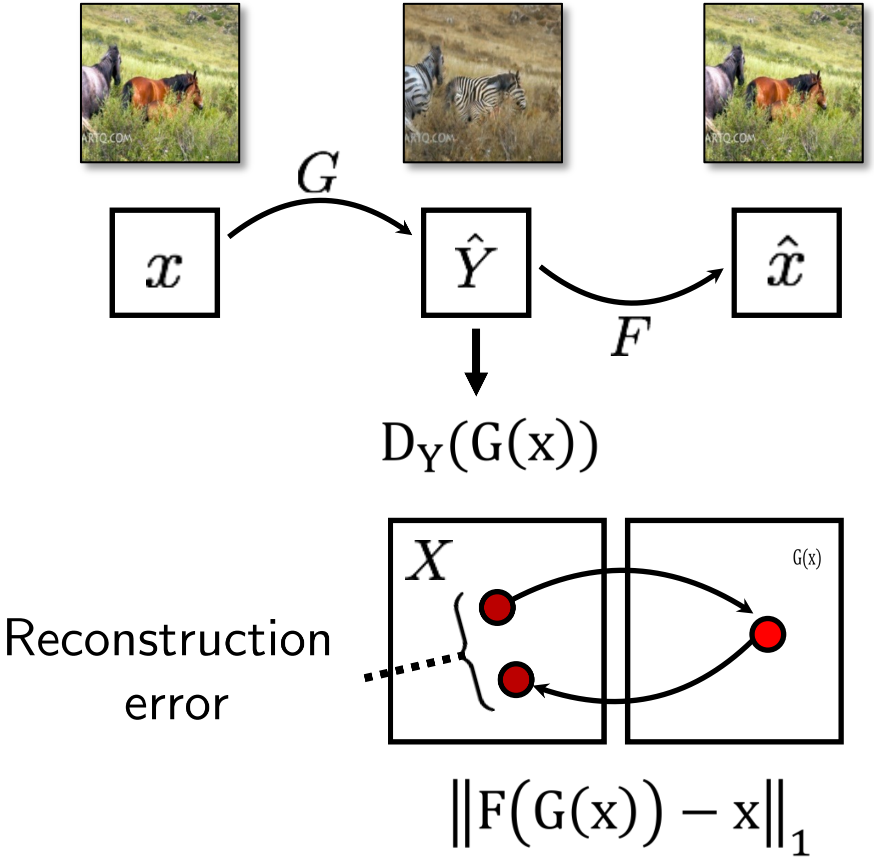


[Mark Twain, 1903]

Slide credit: Jun-Yan Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]

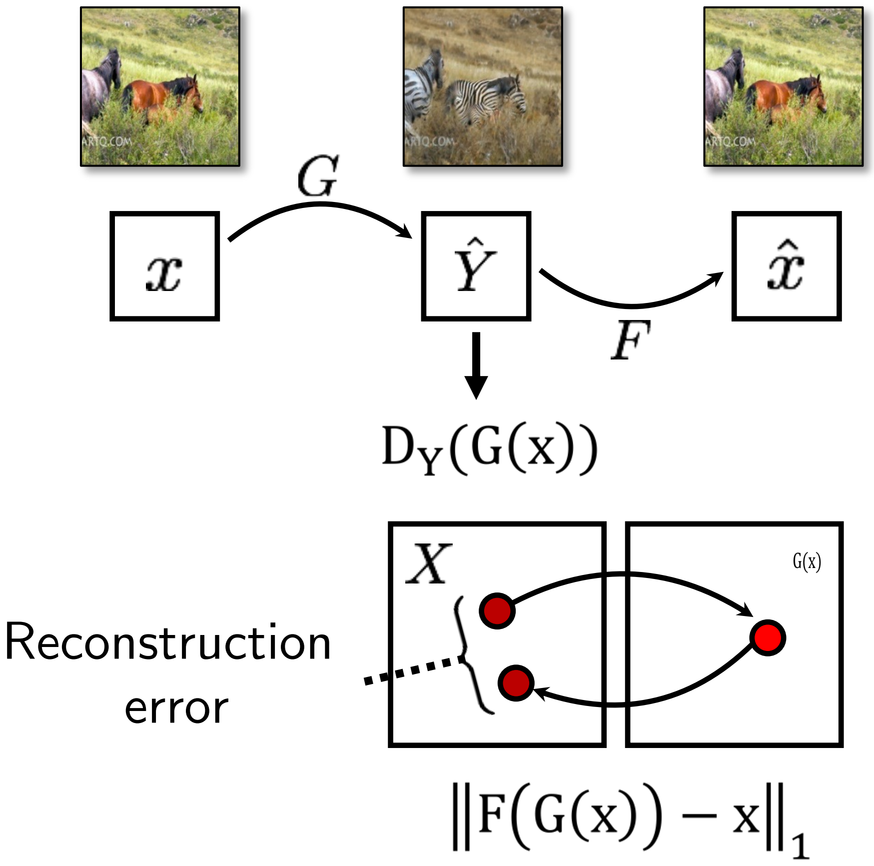
Cycle Consistency Loss



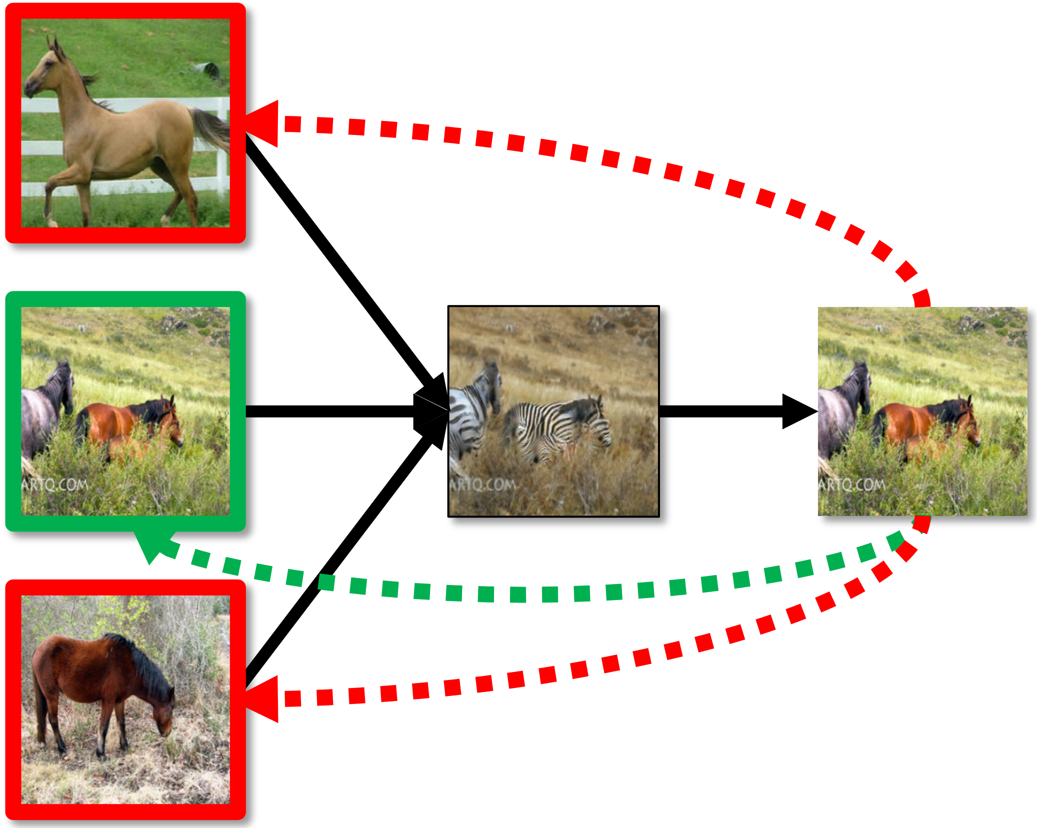
Slide credit: Jun-Yan Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]

Cycle Consistency Loss



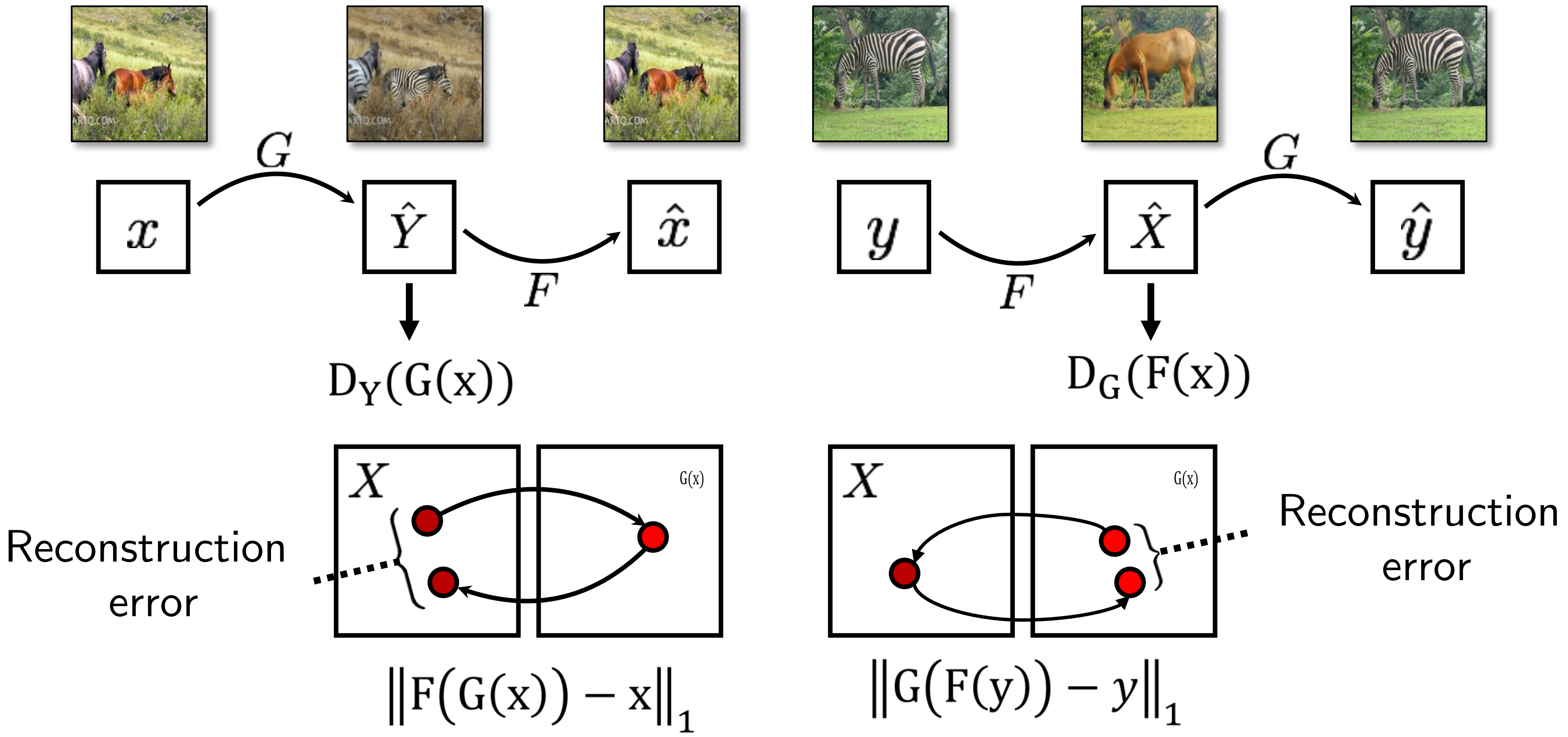
Small cycle loss
Large cycle loss



Slide credit: Jun-Yan Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]

Cycle Consistency Loss



Slide credit: Jun-Yan Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]

Collection Style Transfer

Ukiyo-e Cezanne

Van Gogh Monet


Input
Monet
Van Gogh
Cezanne
Ukiyo-e

Conditional Generation

- Conditional generative model P(zebra images| horse images)

- Style Transfer

Input Image

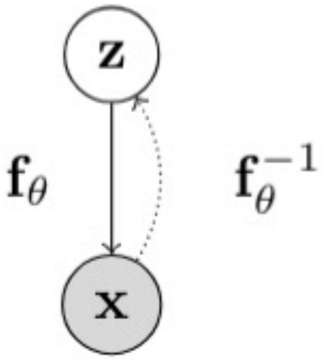
Monet

Van Gogh

Zhou el al., Cycle GAN 2017

Normalizing Flows

- Directed Latent Variable Invertible models



- The mapping between x and z is deterministic and invertible:

$$\begin{aligned} \mathbf{x} &= \mathbf{f}_\theta(\mathbf{z}) \\ \mathbf{z} &= \mathbf{f}_\theta^{-1}(\mathbf{x}) \end{aligned}$$

- Use change-of-variables to relate densities between z and x

$$p_X(\mathbf{x};\theta) = p_Z(\mathbf{z}) \left| \det \frac{\partial \mathbf{f}_\theta^{-1}(\mathbf{x})}{\partial X} \right|_{X=\mathbf{x}}$$

Grover and Ermon DGM Tutorial, NICE (Dinh et al. 2014), Real NVP (Dinh et al. 2016)

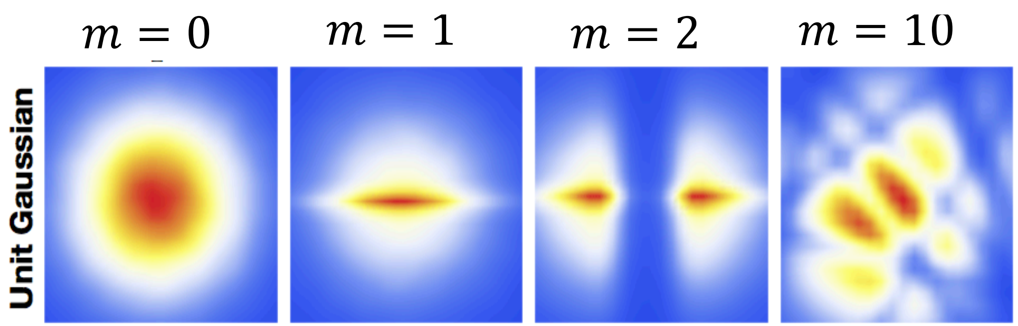
Normalizing Flows

- Invertible transformations can be composed:

$$\mathbf{x} = \mathbf{f}_\theta^M \circ \cdots \circ \mathbf{f}_\theta^1(\mathbf{z}^0); \qquad p_X(\mathbf{x};\theta) = p_{Z^0}(\mathbf{z}^0) \prod_{m=1}^{M} \left| \det \frac{\partial (\mathbf{f}_\theta^m)^{-1}}{\partial Z^m} \right|_{Z^m = \mathbf{z}^m}$$

- Planar Flows

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u} g(\mathbf{w}^\top \mathbf{z} + b)$$



Rezendre and Mohamed, 2016

Normalizing Flows

- Maximum log-likelihood objective

$$\max_{\theta} \log p_X(\mathcal{D};\theta) = \sum_{\mathbf{x}\in\mathcal{D}} \left(\log p_Z(\mathbf{z}) - \log \left| \det \frac{\partial (\mathbf{f}_\theta)^{-1}}{\partial X} \right|_{X=\mathbf{x}} \right)$$

- Exact log-likelihood evaluation via inverse transformations
- Sampling from the model

$$\mathbf{z} \sim p_Z(\mathbf{z}), \quad \mathbf{x} = \mathbf{f}_\theta(\mathbf{z})$$

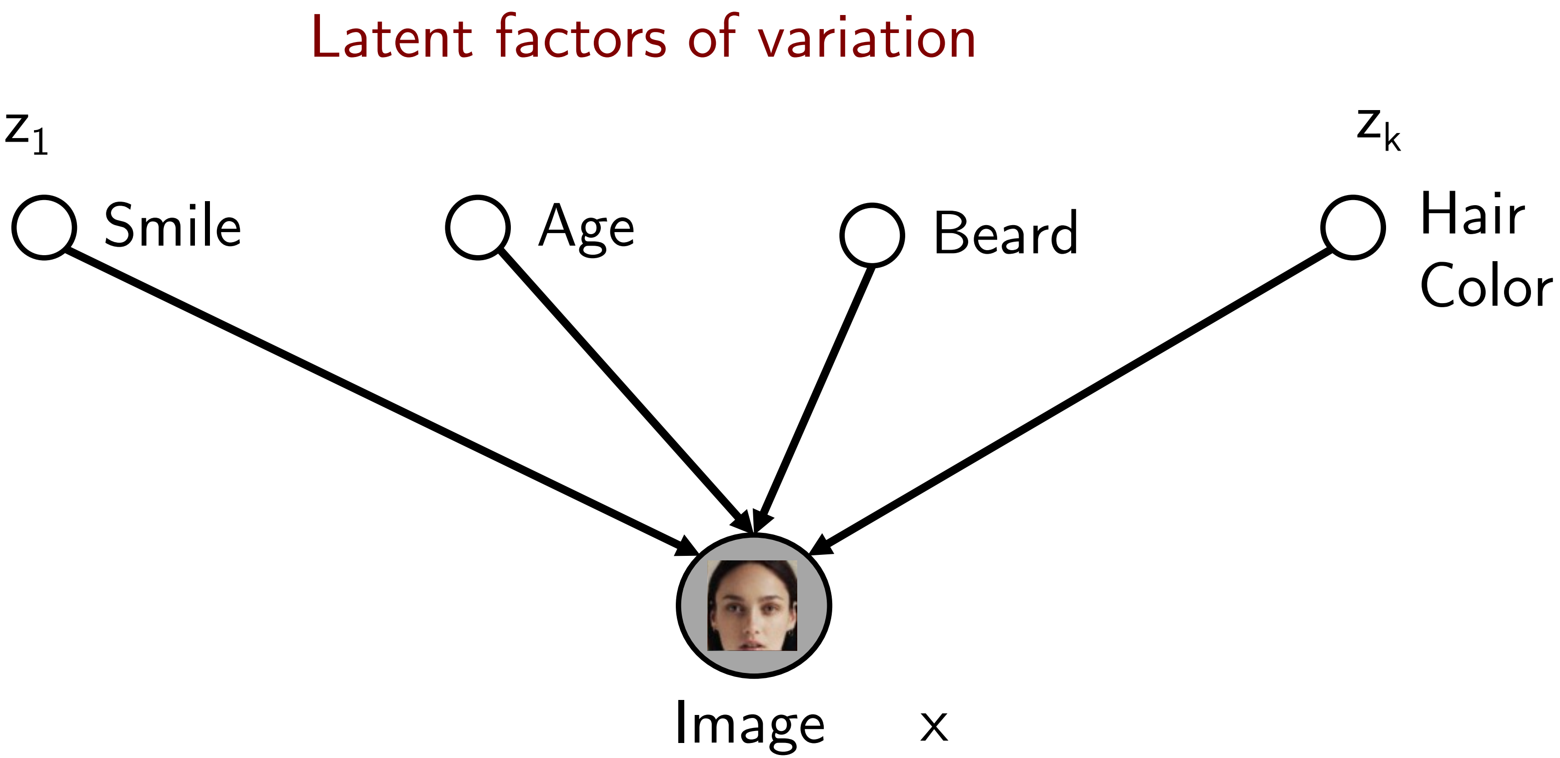
- Inference over the latent representations:

$$\mathbf{z} \quad = \quad \mathbf{f}_\theta^{-1}(\mathbf{x})$$

Rezendre and Mohamed, 2016, Grover and Ermon DGM Tutorial

Example: GLOW

- Generative Flow with Invertible 1x1 Convolutions
  https://blog.openai.com/glow/



Kingma, Dhariwal, 2018

Example: GLOW

Input

Smile

Add Beard

Increase Age

https://blog.openai.com/glow/

Flow Models

- Simple prior that allows for sampling and tractable likelihood evaluation e.g., isotropic Gaussian

- Invertible transformations with tractable evaluation:
  - Likelihood evaluation requires efficient evaluation of inverse
  - Sampling requires efficient evaluation of inverse

- Tractable evaluation of determinants of Jacobian for large models
  - Computing determinants for a large matrix is prohibitive
  - Key idea: Determinant of triangular matrices is the product of the diagonal entries, i.e., an operation