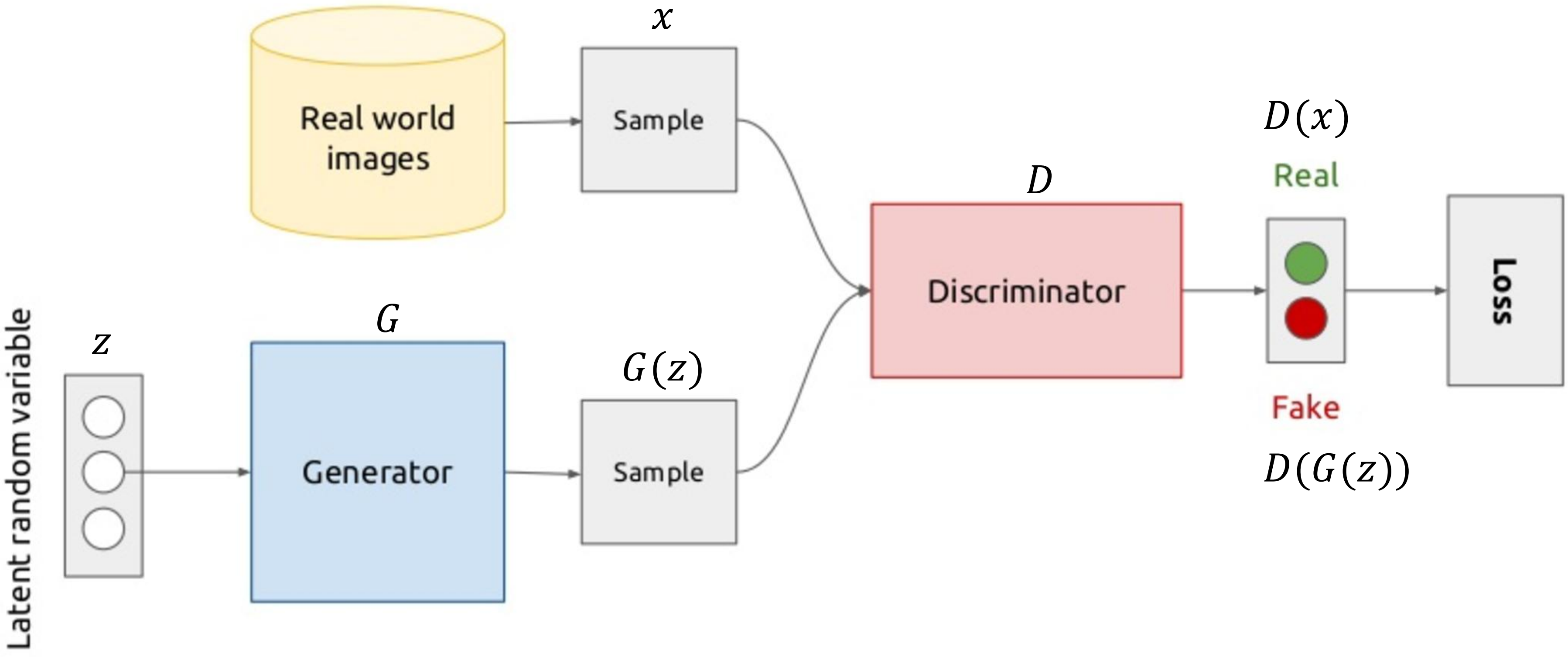
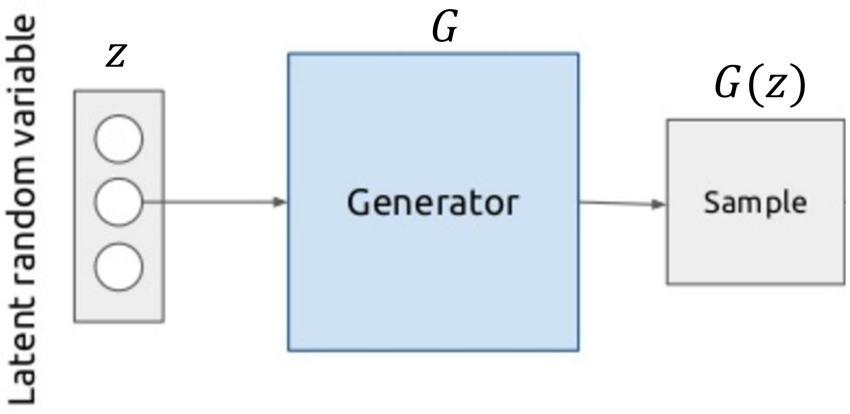
Conditional Generative Adversarial Networks (cGANs)

Generative Adversarial Networks (GANs)



Generative Adversarial Networks (GANs)

At test time: sample random variable -> obtain generated sample



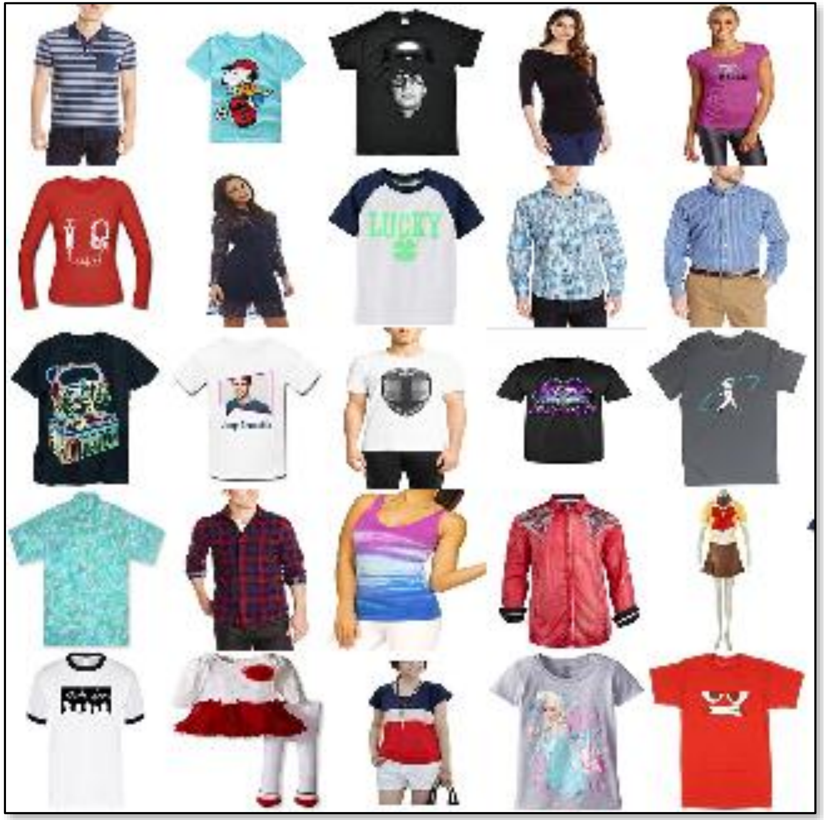
Conditional GANs (cGANs)

- Gain control of output

- Modeling (e.g., sketch-based modeling, etc.)
  - Add semantic meaning to latent space manifold

- Domain transfer
  - Labels on A -> transfer to B, train network on B, test on B
  - More later

GAN Manifold

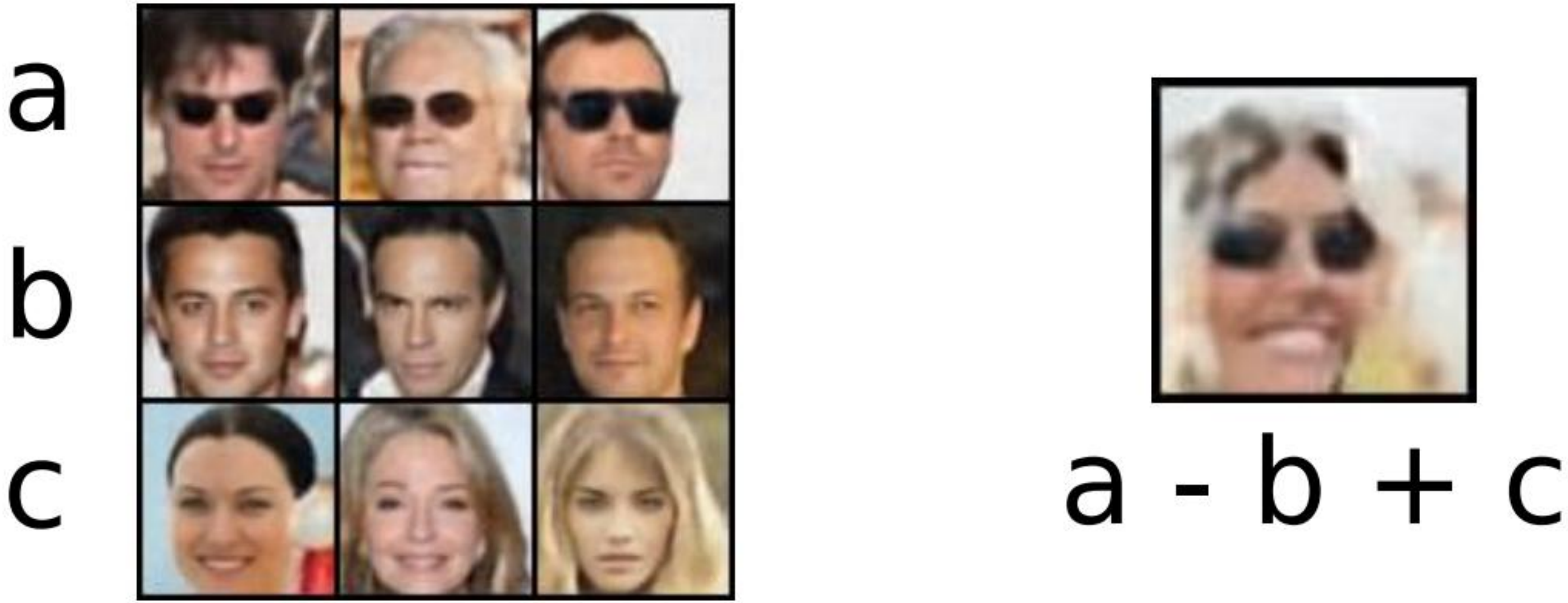


Train Data

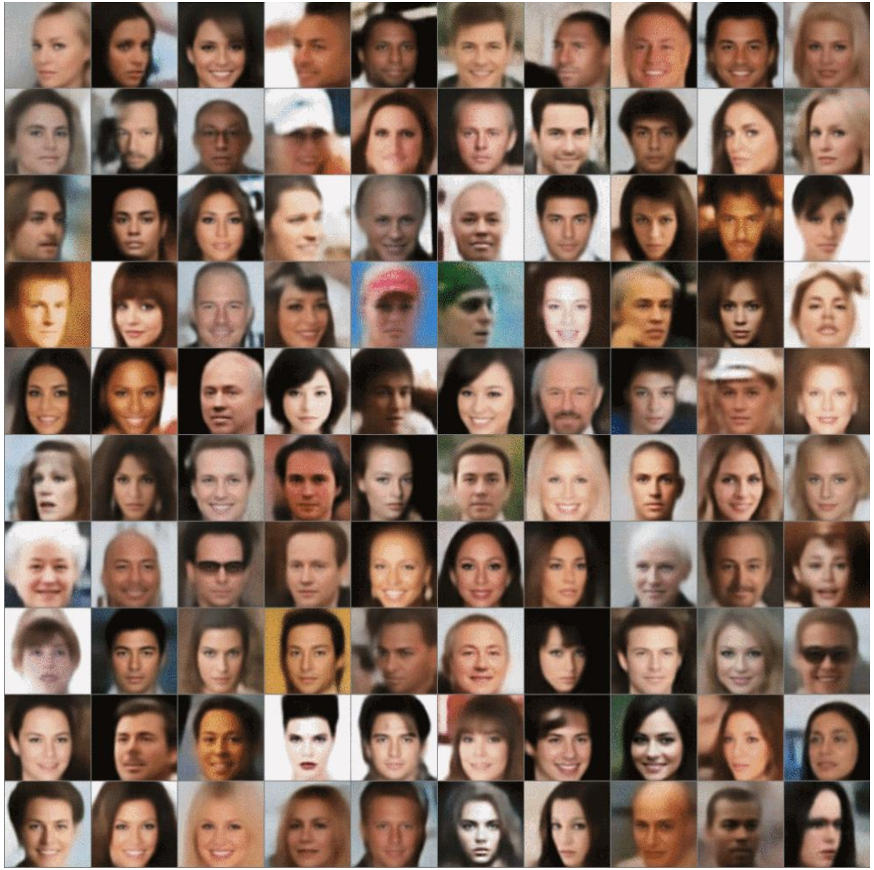
Sampled Data -> G(z)

[Radford et al. 15] Deep Convolutional Generative Adversarial Networks

GAN Manifold

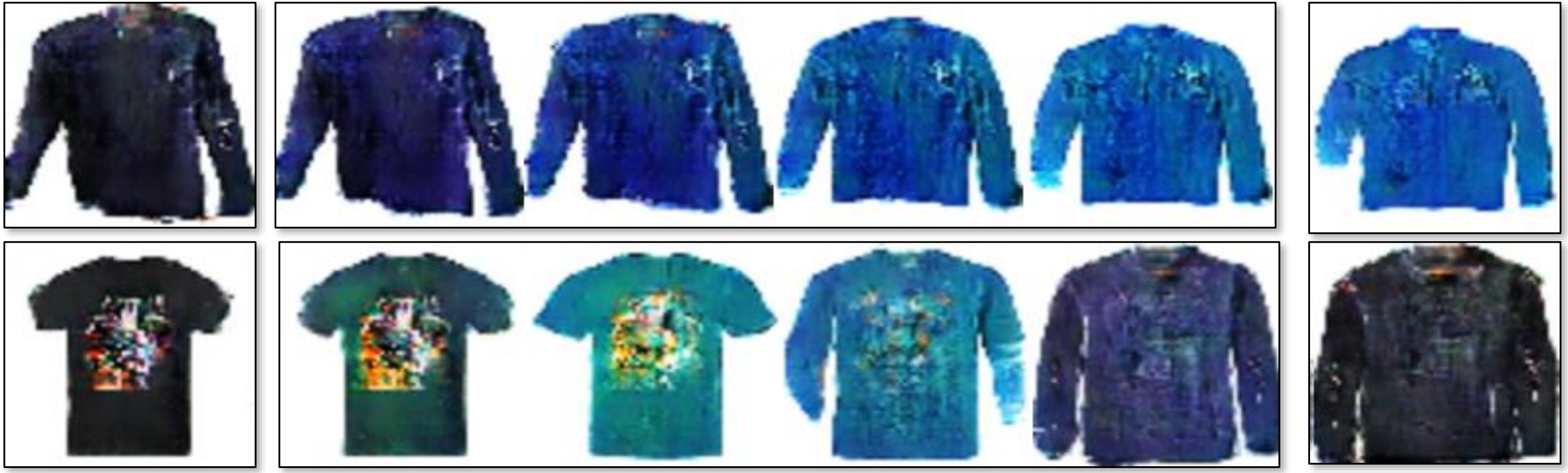


GAN Manifold



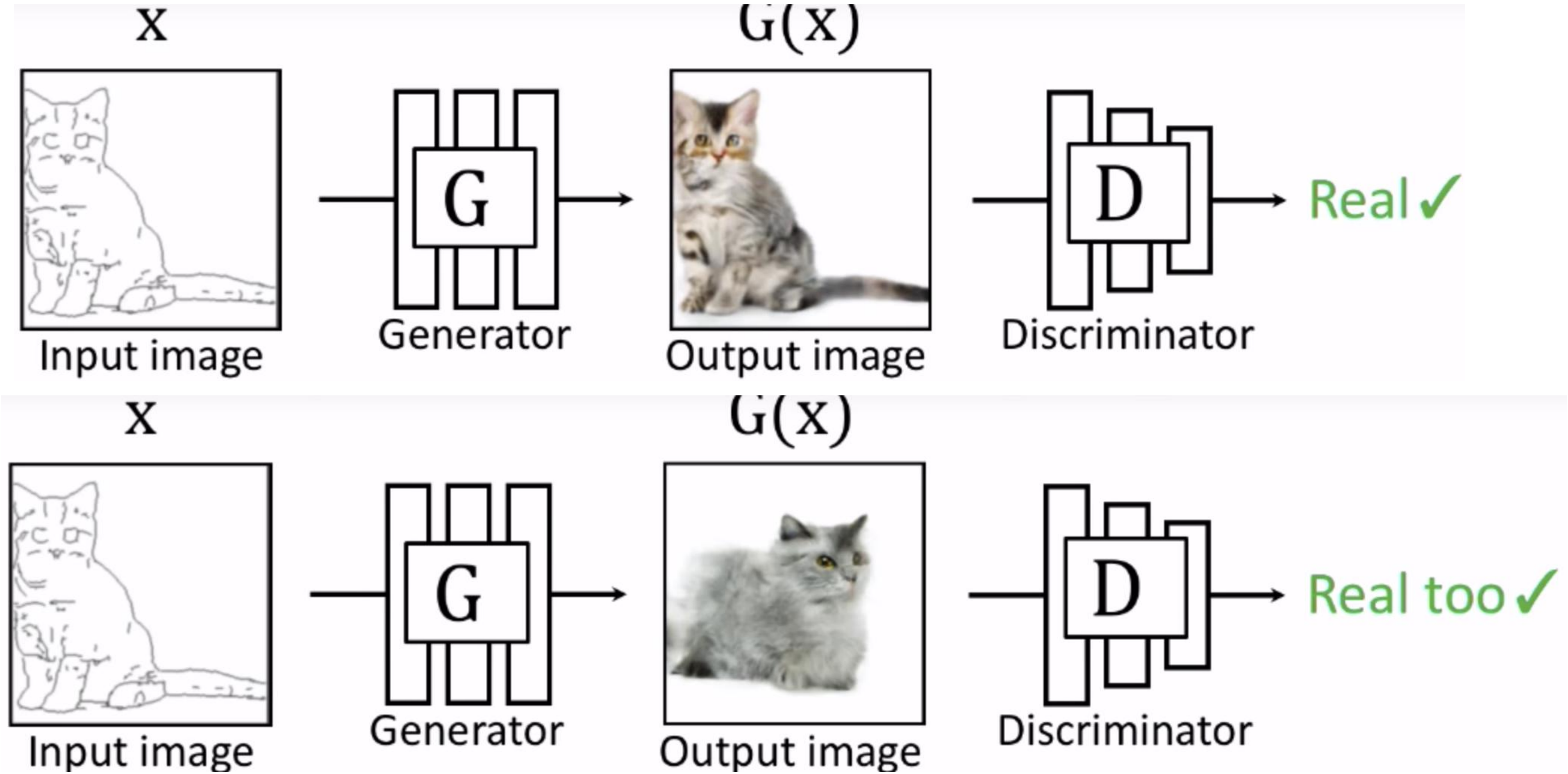
GAN Manifold

$G(z_0)$ Linear interpolation in z space: $G(z_0 + t \cdot (z_1 - z_0))$ $G(z_1)$

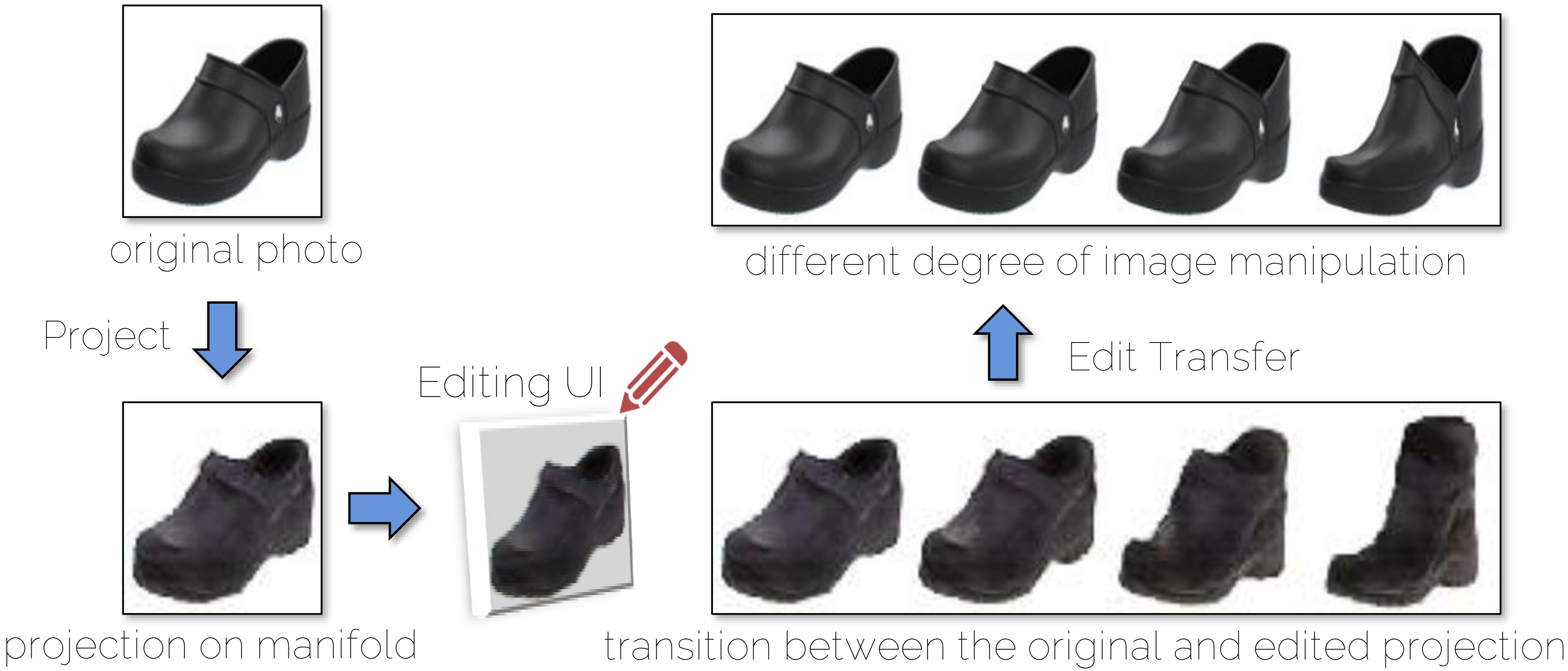


[Radford et al. 15] Deep Convolutional Generative Adversarial Networks

Conditional GANs (cGANs)

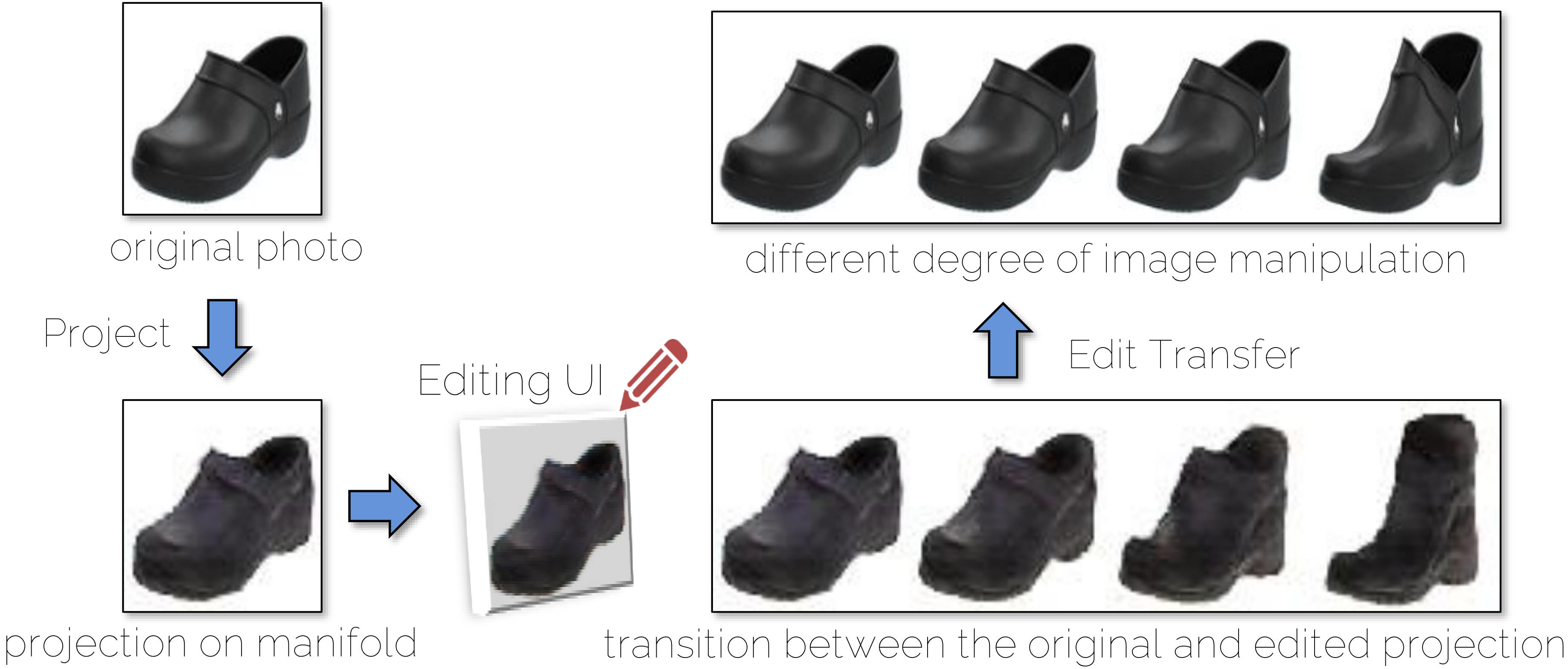


iGANs: Overview



[Zhu et al. 16] Generative Visual Manipulation on the Natural Image Manifold

iGANs: Overview



[Zhu et al. 16] Generative Visual Manipulation on the Natural Image Manifold

iGANs: Projecting an Image onto the Manifold

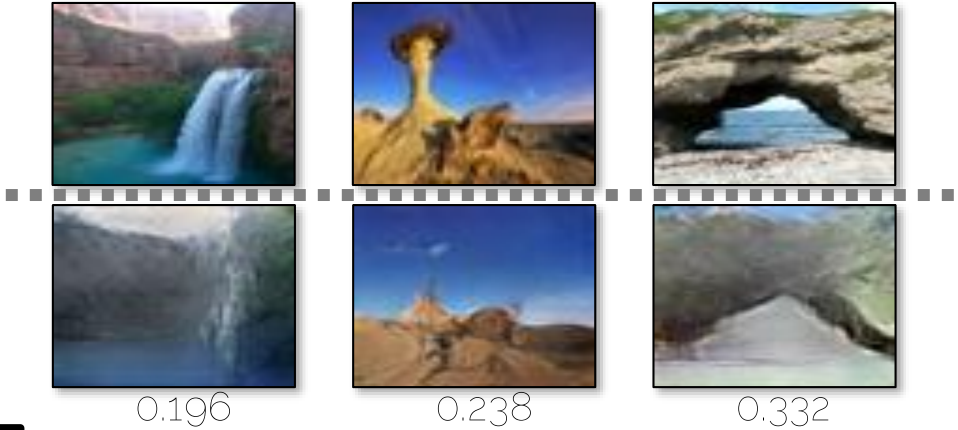
Input: real image x^R
Output: latent vector z

Optimization

$$z^* = \arg\min \mathcal{L}(G(z), x^R)$$

Reconstruction loss L

Generative model $G(z)$



iGANs: Projecting an Image onto the Manifold

Input: real image $\boldsymbol{x}^R$
Output: latent vector z

Optimization

$$z^* = \arg\min \mathcal{L}(G(z), x^R)$$

Inverting Network $\mathbf{z} = P(x)$

$$\theta_P^* = \arg\min_{\theta_P} \sum_{x_n^R} \mathcal{L}(\underbrace{G(P(x^R; \theta_P))}_{\text{Auto-encoder}}, x^R)$$

with a fixed decoder $\mathbf{G}$

iGANs: Projecting an Image onto the Manifold

Input: real image x^R
Output: latent vector z

Optimization

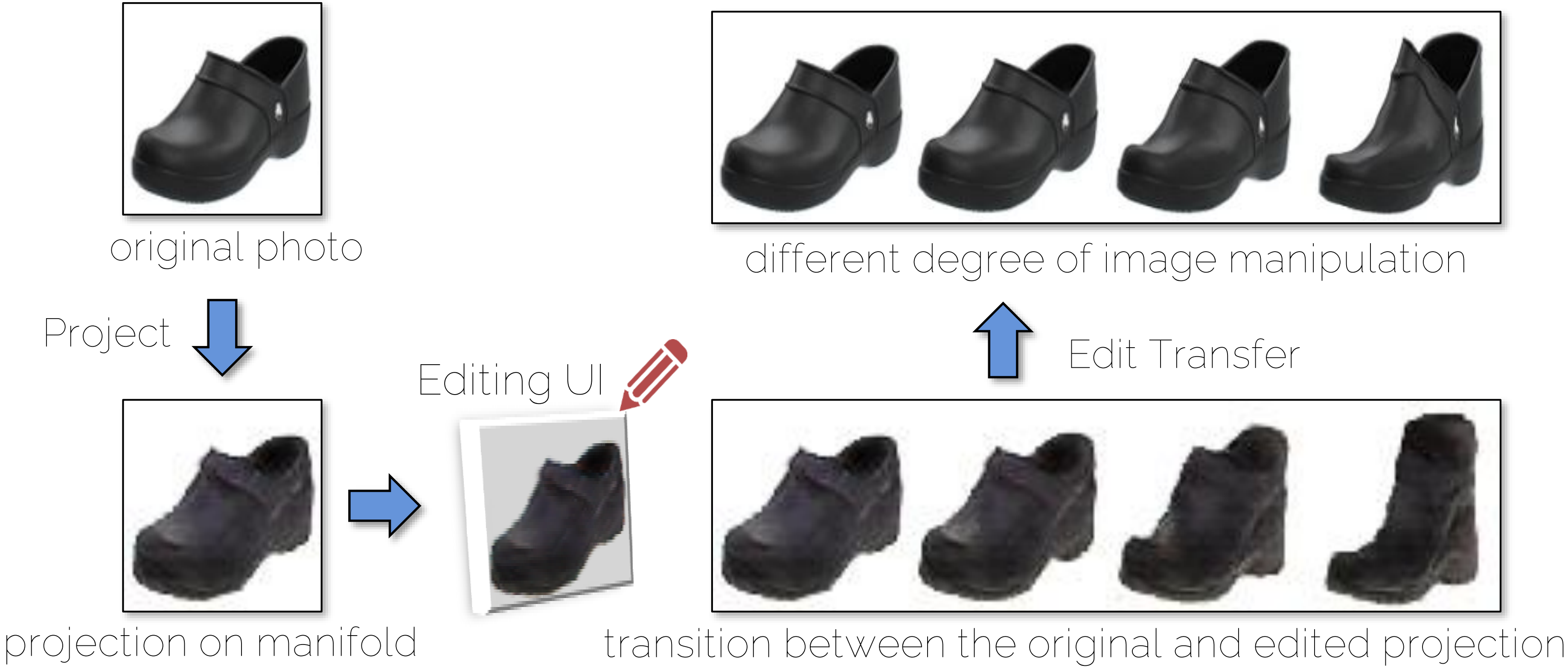
$$z^* = \arg\min \mathcal{L}(G(z), x^R)$$

Inverting Network $\mathbf{z} = P(x)$

$$\theta_P^* = \arg\min_{\theta_P} \sum_{x_n^R} \mathcal{L}(\underbrace{G(P(x^R; \theta_P))}, x^R)$$

Hybrid Method
Use the network as initialization for the optimization problem

iGANs: Overview



iGANs: Manipulating the Latent Vector

constraint violation loss L_g

user guidance image

Objective: $$z^* = \arg\min_{z \in \mathbb{Z}} \Big\{ \underbrace{\sum_g \big(\mathcal{L}_g(G(z), v_g)}_{\text{data term}} + \underbrace{\lambda_s \cdot \|z - z_0\|_2^2}_{\text{manifold smoothness}} \Big\}.$$

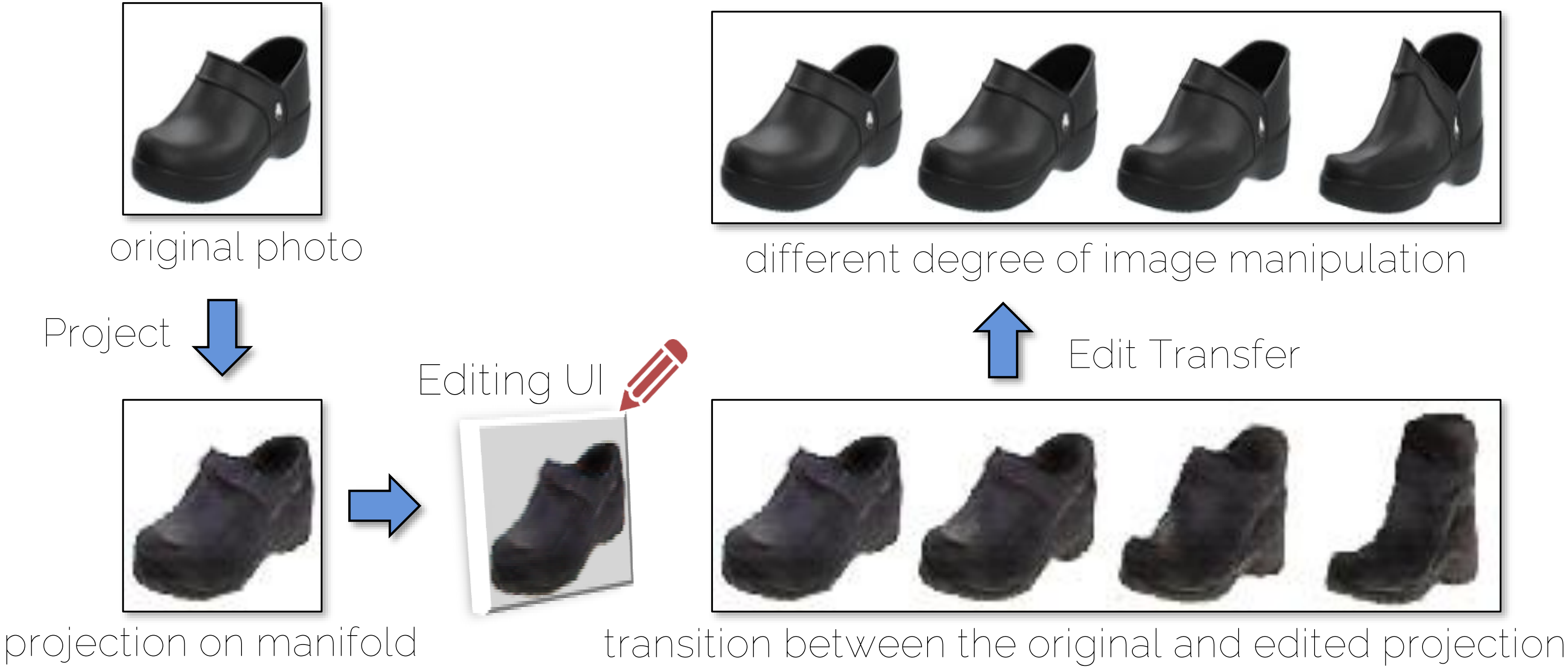
Guidance v_g

$G(z)$

z_0

[Zhu et al. 16] Generative Visual Manipulation on the Natural Image Manifold

iGANs: Overview



iGANs: Edit Transfer

Motion (u, v)+ Color ($\boldsymbol{A}_{3\times4}$): estimate per-pixel geometric and color variation

$$\iint \underbrace{\|I(x,y,t)-\boxed{A}\cdot I(x+\boxed{u},y+\boxed{v},t+1)\|^2}_{\text{data term}}+\underbrace{\sigma_s(\|\nabla u\|^2+\|\nabla v\|^2)}_{\text{spatial reg}}+\underbrace{\sigma_c\|\nabla A\|^2}_{\text{color reg}}dxdy$$

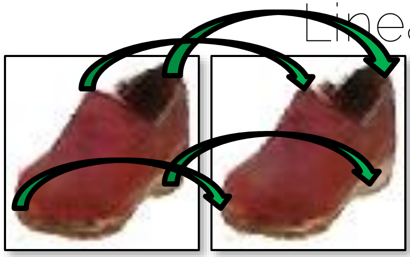
iGANs: Edit Transfer

Motion (u, v)+ Color ($\boldsymbol{A_{3\times 4}}$): estimate per-pixel geometric and color variation

$$\iint \underbrace{\|I(x,y,t) - \boxed{A}\cdot I(x+\boxed{u}, y+\boxed{v}, t+1)\|^2}_{\text{data term}} + \underbrace{\sigma_s(\|\nabla u\|^2 + \|\nabla v\|^2)}_{\text{spatial reg}} + \underbrace{\sigma_c\|\nabla A\|^2}_{\text{color reg}} dxdy$$

$G(z_0)$

Linear Interpolation in $\boldsymbol{z}$ space



$G(z_1)$

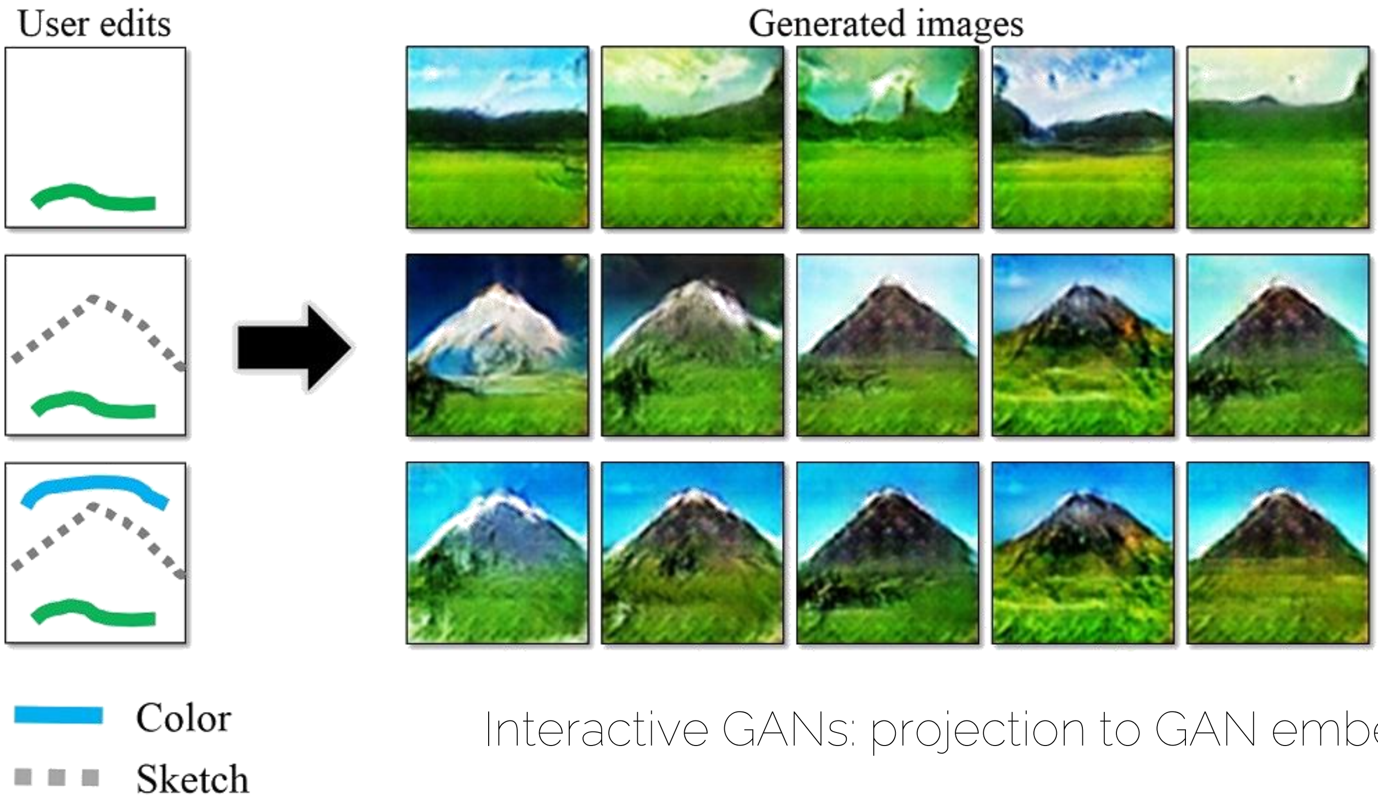
Input

iGANs: Edit Transfer

Motion (u, v)+ Color ($\boldsymbol{A}_{3\times4}$): estimate per-pixel geometric and color variation

$$\iint \underbrace{\|I(x,y,t) - \boxed{A} \cdot I(x+\boxed{u}, y+\boxed{v}, t+1)\|^2}_{\text{data term}} + \underbrace{\sigma_s(\|\nabla u\|^2 + \|\nabla v\|^2)}_{\text{spatial reg}} + \underbrace{\sigma_c\|\nabla A\|^2}_{\text{color reg}} dxdy$$

cGANs: Interactive GANs



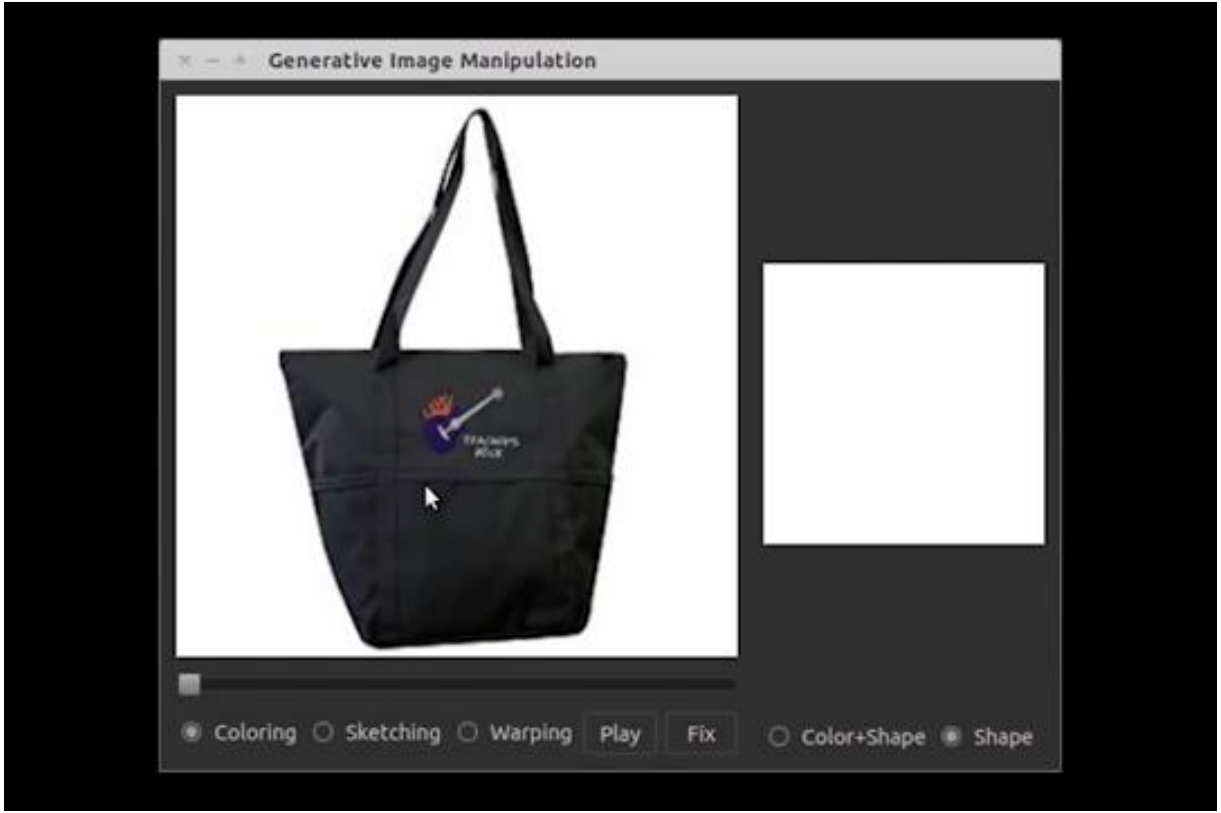
Interactive GANs: projection to GAN embedding

https://github.com/junyanz/iGAN [Zhu et al. 16.]

cGANs: Interactive GANs

https://github.com/junyanz/iGAN [Zhu et al. 16.]

cGANs: Interactive GANs



https://github.com/junyanz/iGAN [Zhu et al. 16.]

Mapping in Latent Space is Difficult!

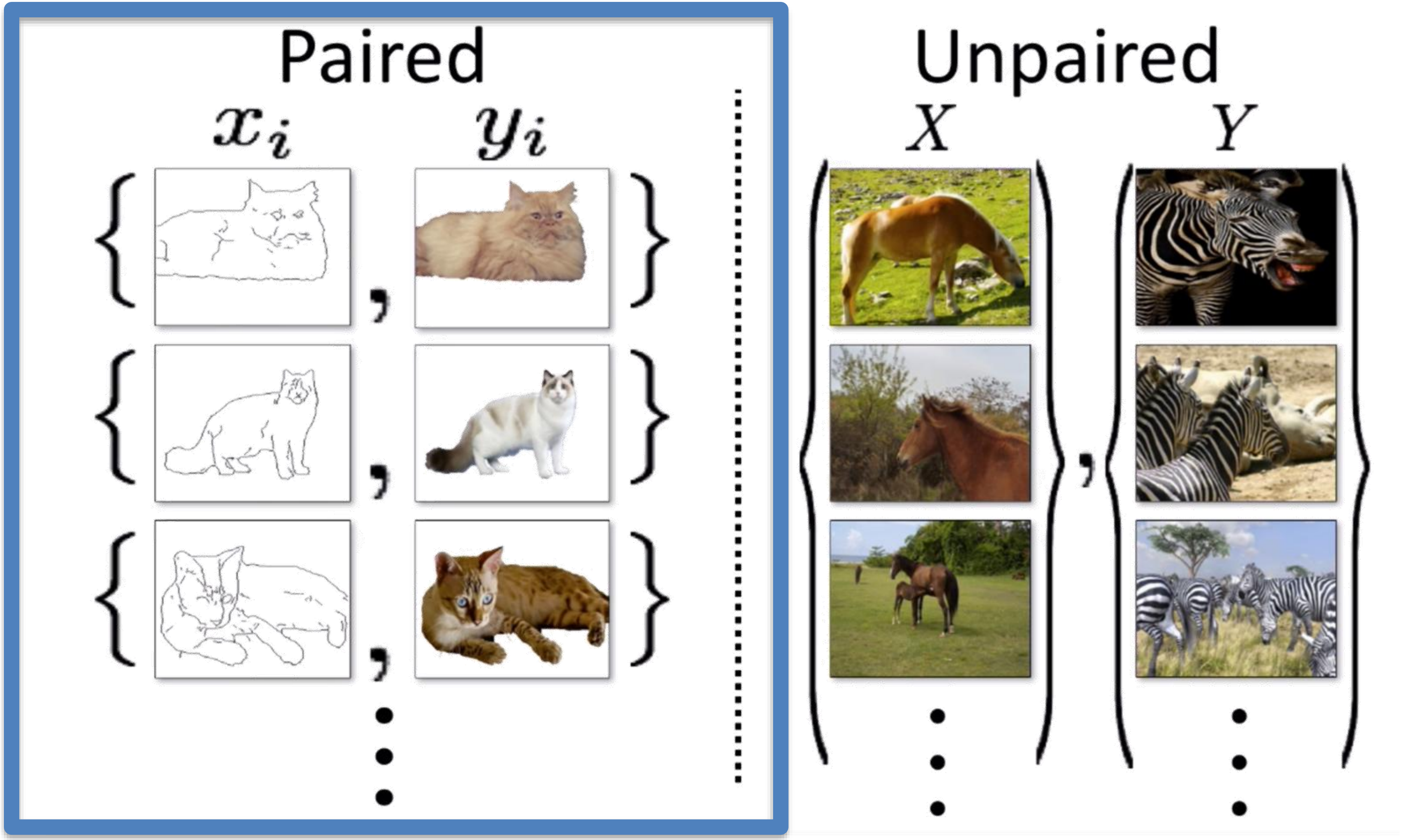
- Semantics are missing
- In most cases, no labels available
- Ideally, need some unsupervised disentangled rep.

(a) Azimuth (pose)

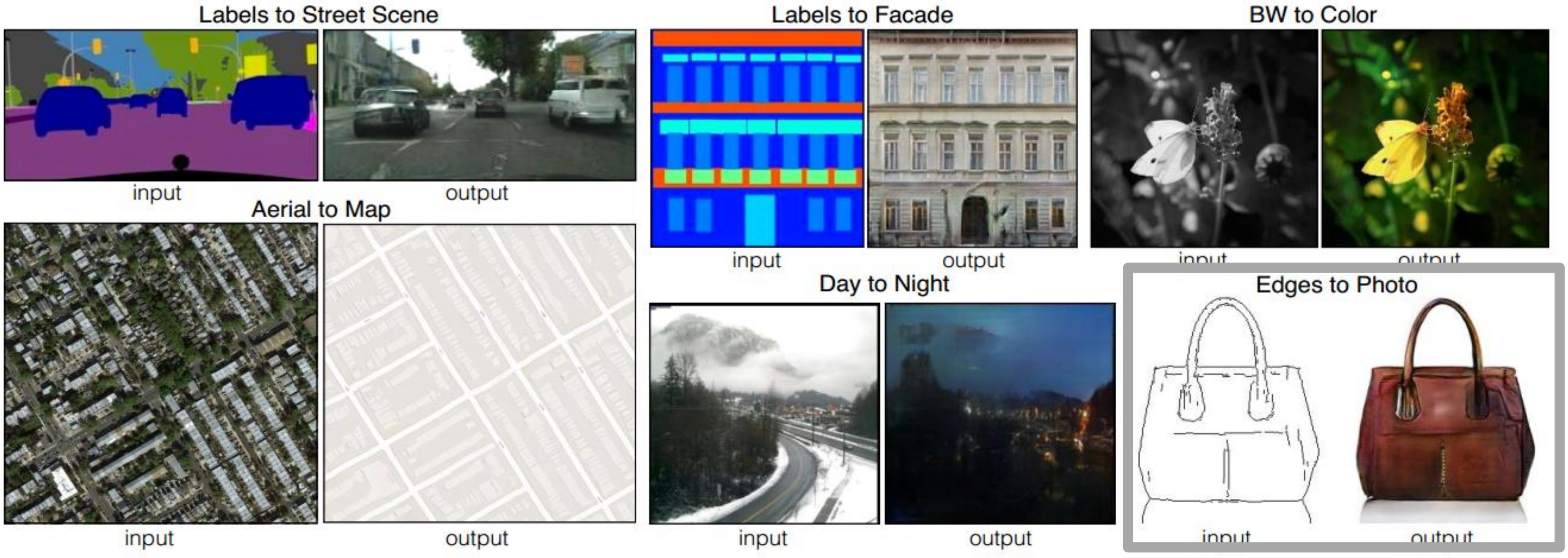
(b) Presence or absence of glasses

InfoGAN [Chen et al. 16]

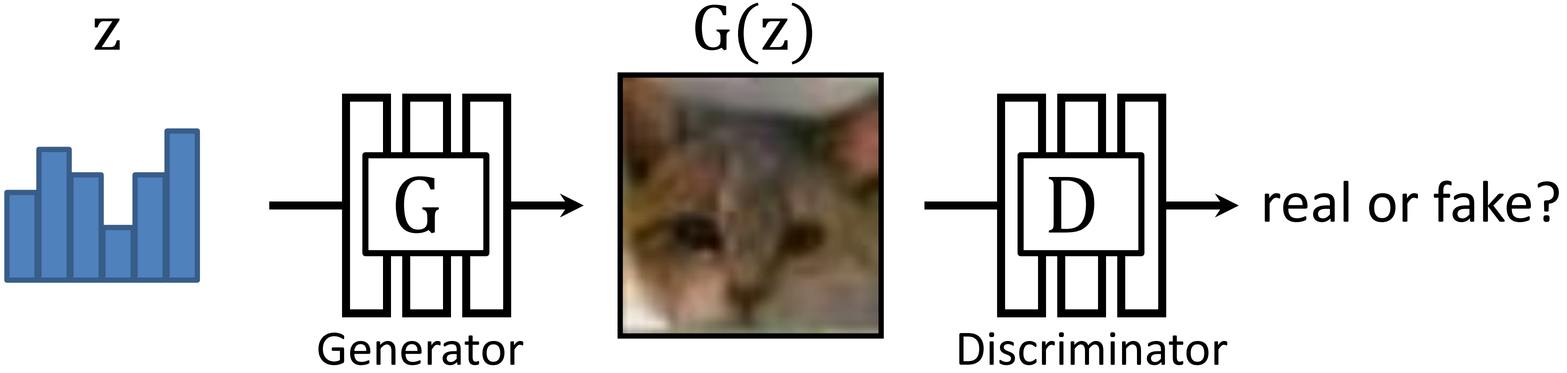
Paired vs Unpaired Setting



pix2pix: Image-to-Image Translation

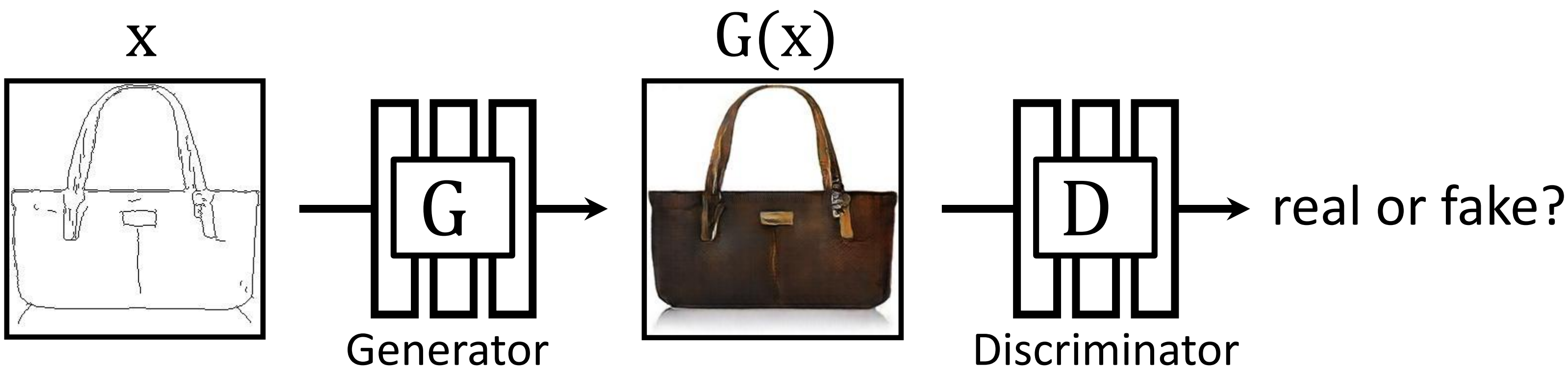


[Isola et al. 16] Image-to-Image Translation with Conditional Adversarial Networks

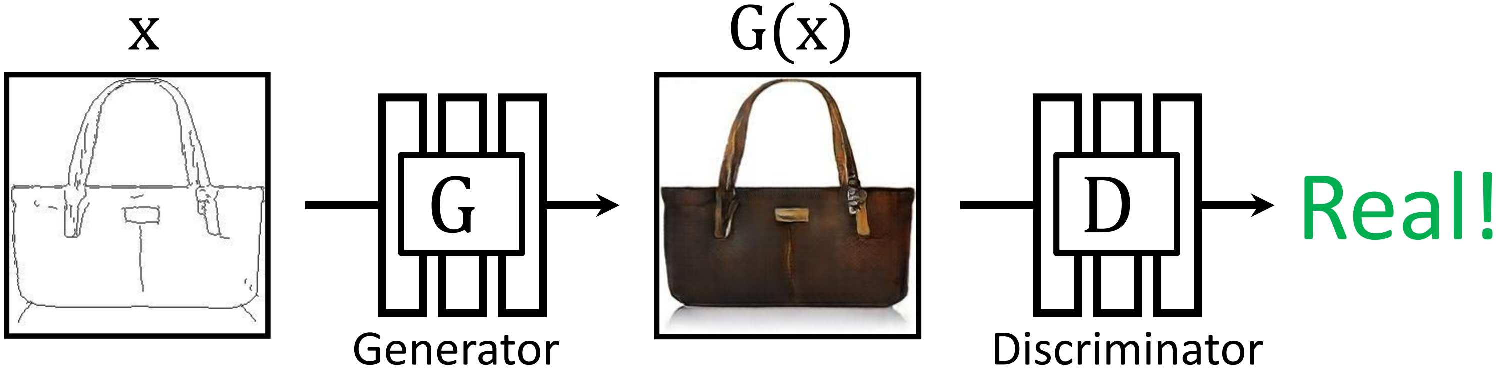


$$\min_{G} \max_{D} \mathbb{E}_{z,x}[\log D(G(z)) + \log(1 - D(x))]$$

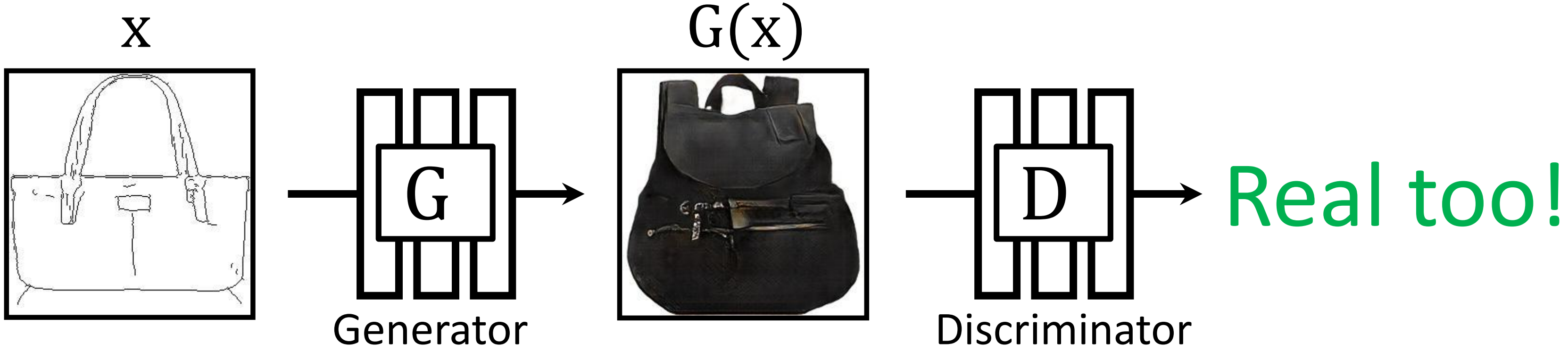
[Goodfellow et al. 2014] Generative Adversial Networks



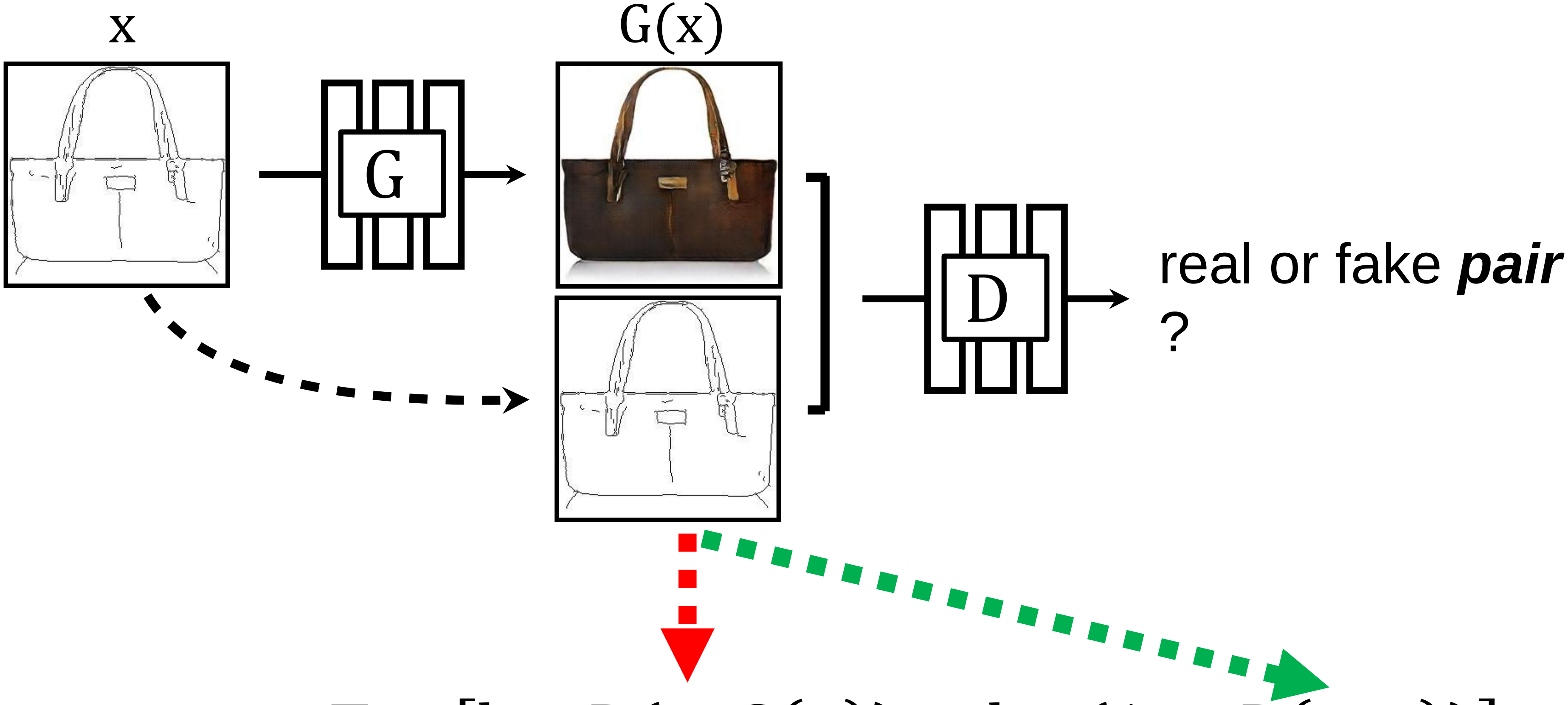
$$\min_G \max_D \mathbb{E}_{x,y}[\log D(G(x)) + \log(1 - D(y))]$$



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



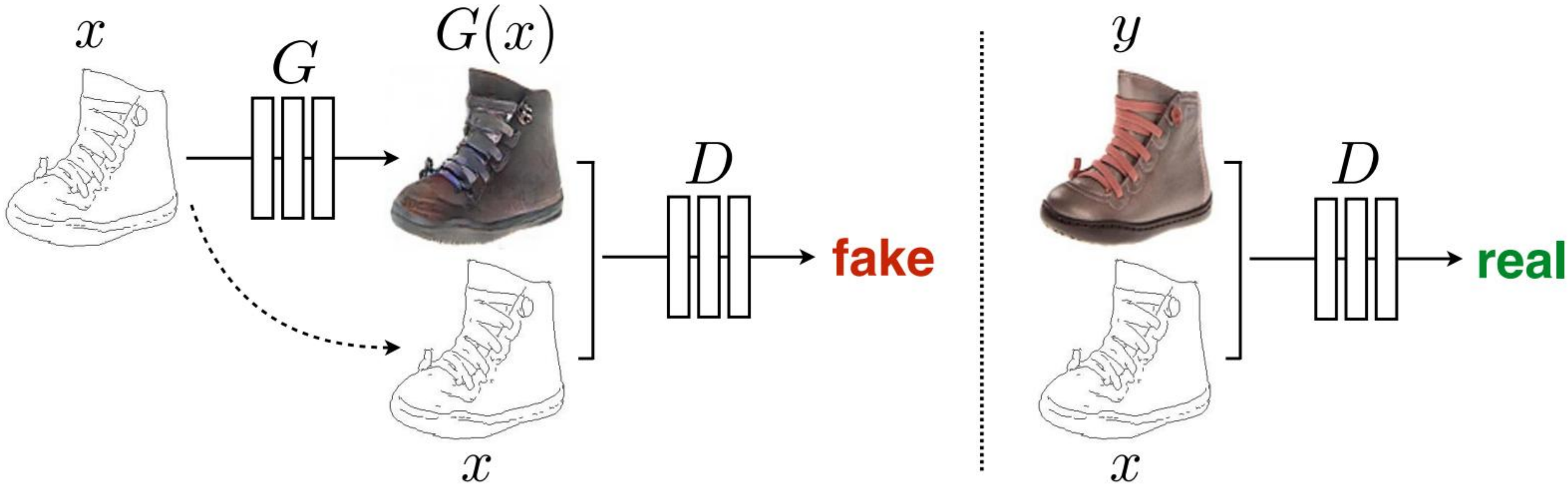
$$\min_G \max_D \mathbb{E}_{x,y}[\log D(G(x)) + \log(1 - D(y))]$$



$$\min_{G}\max_{D}\mathbb{E}_{x,y}[\log \underbrace{D(x,G(x))}_{\text{fake pair}} + \log(1-\underbrace{D(x,y)}_{\text{real pair}})]$$

match joint distribution $p(G(x),y) \sim p(x,y)$

Pix2Pix



Pix2Pix: Paired Setting

- Great when we have 'free' training data
- Often called self-supervised
- Think about these settings ☺

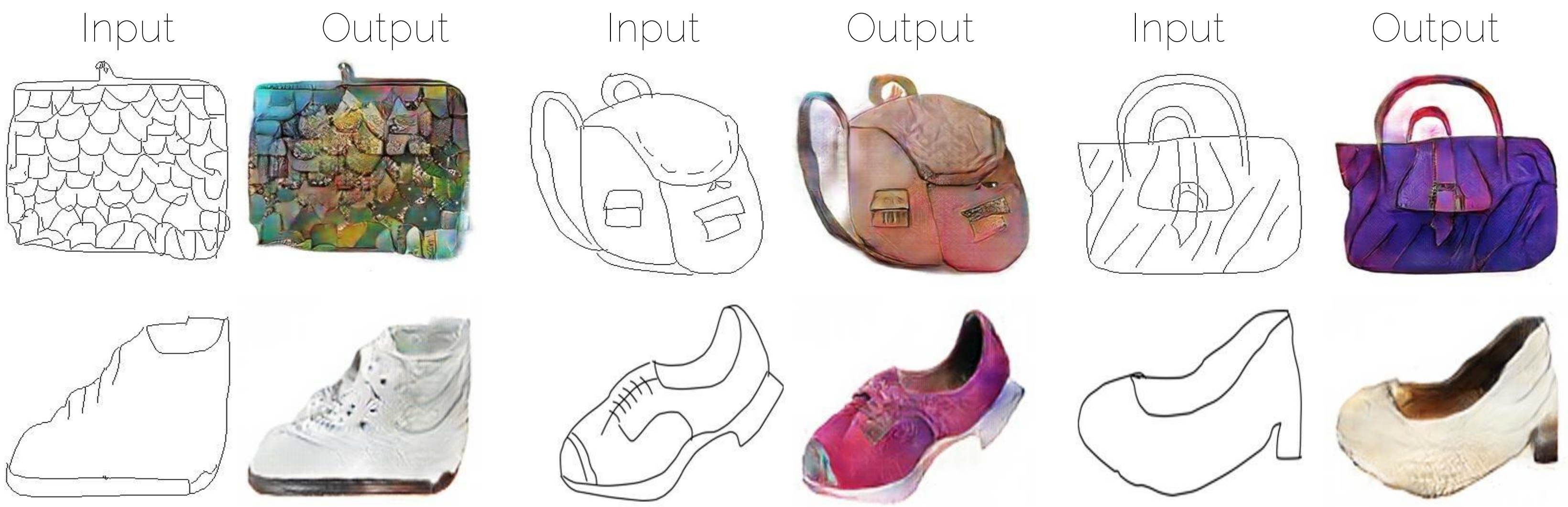
Pix2Pix - Examples

Edges → Images

[Xie & Tu 15] Holistically-Nested Edge Detection

Pix2Pix - Examples

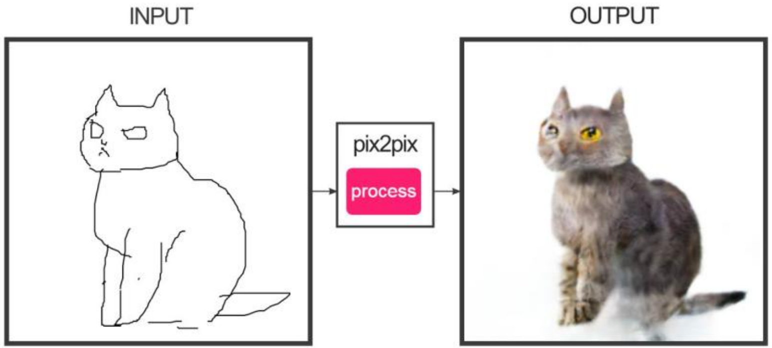
Sketches → Images

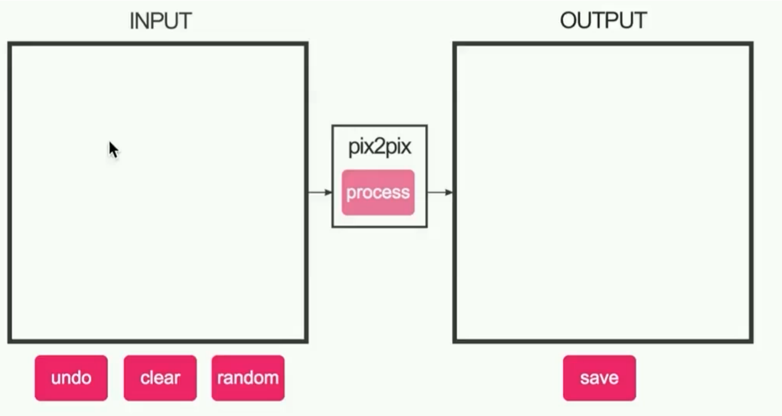


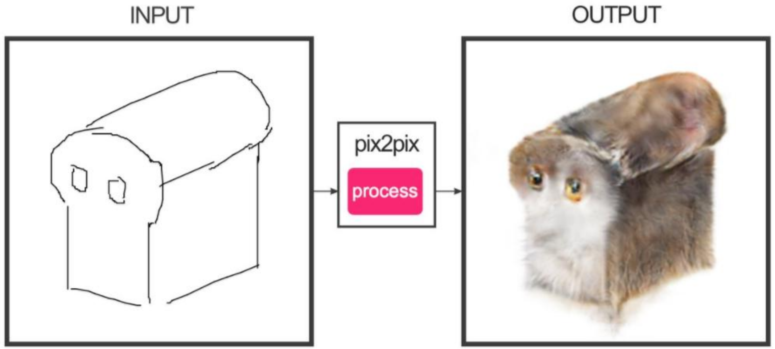
Trained on Edges → Images

How Do Humans Sketch Objects [Eitz, Hays, Alexa, 2012]

Pix2Pix - Examples







Vitaly Vidmirov @vvid

#edges2cats by Christopher Hesse https://affinelayer.com/pixsrv/

Pix2Pix - Examples

Data from [maps.google.com]

Pix2Pix - Examples

BW → Color

Input | Output | Input | Output | Input | Output

Data from [Russakovsky et al. 2015 - ImageNet]

Ideas behind Pix2Pix

- $L = L_{GAN} + \lambda L_1$ (makes it more constraint)
- Unet / skip connections for preserving structure
- Noise only through dropout
  - cGANs tend to learn to ignore the random vector z
  - Still want probabilistic model

Ideas behind Pix2Pix

- L1 or L2 loss for low frequency details

- GAN discriminator for high frequency details

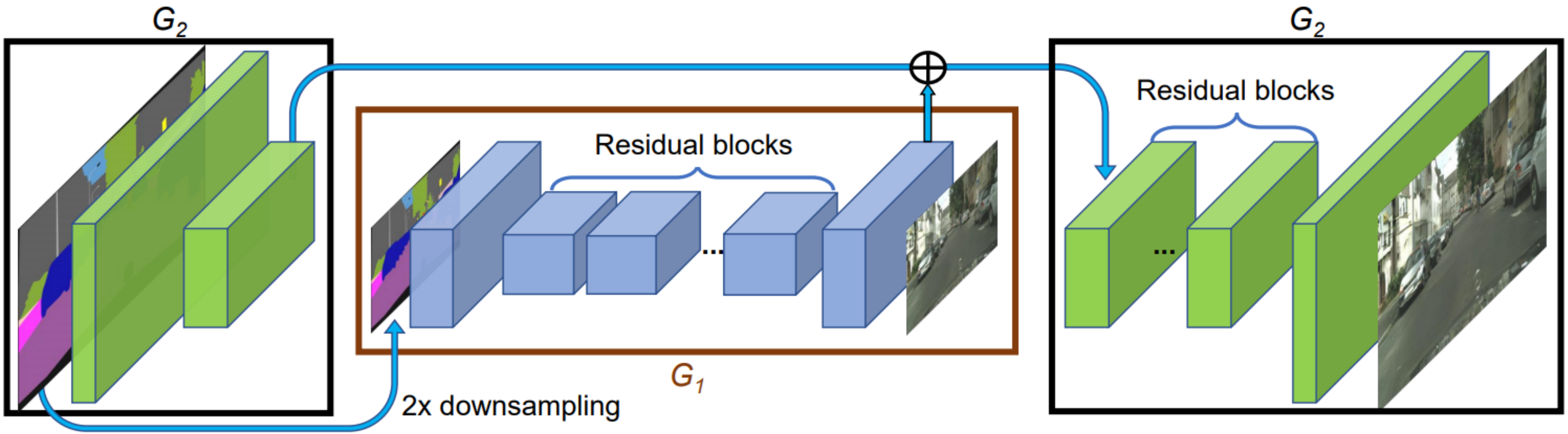
-> PatchGAN
  - GAN discriminator applied only to local patches
  - It's fully-convolutional; i.e., can run on arbitrary image sizes

Pix2PixHD

- Expand the pix2pix idea to multi-scale
- Coarse-to-fine generator + discriminator
- G's and D's are the same but since they operate on different resolutions, they have effectively a larger receptive field

[Wang et al. 18] pix2pixHD

Pix2PixHD



[Wang et al. 18] pix2pixHD

Pix2PixHD

- Use of multi-scale discriminators

- $\min_{G} \max_{D_1, D_2, D_3} \sum_{k=1,2,3} L_{GAN}(G, D_k)$

- Can make various combinations of stacking discriminator and generator
  - E.g., have a single G and downsample generated and real images – or have intermediate real images (cf. ProGAN)

Pix2PixHD

[Wang et al. 18] pix2pixHD

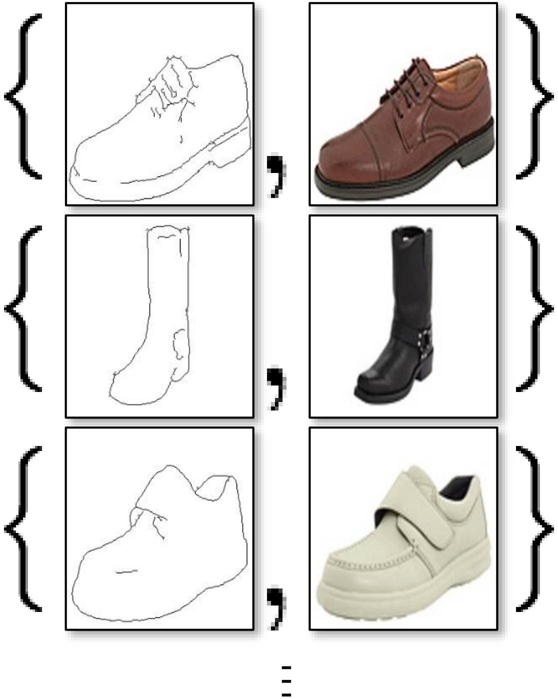
Pix2PixHD

[Wang et al. 18] pix2pixHD

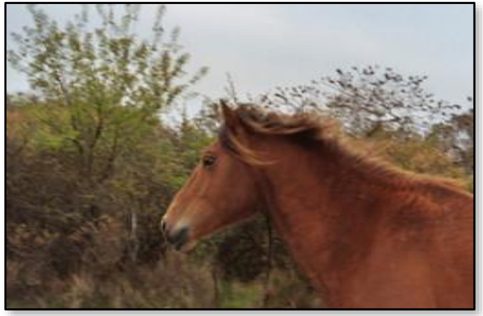
Pix2PixHD (Interactive Results)

Paired

$$\left\{ x_i, y_i \right\}$$



Label ↔ photo: per-pixel labeling

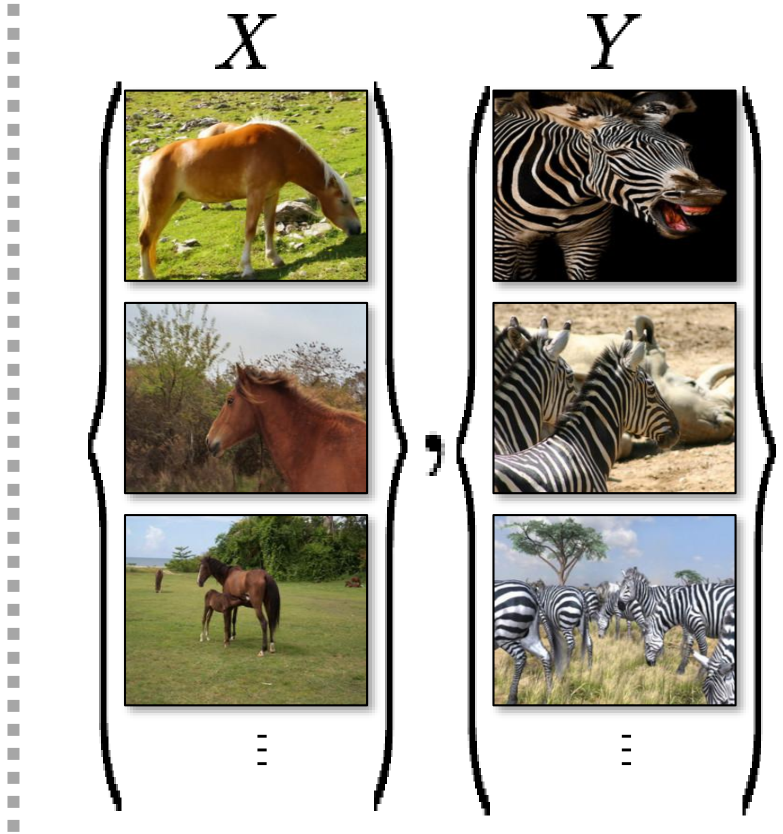


Horse ↔ zebra: how to get zebras?

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

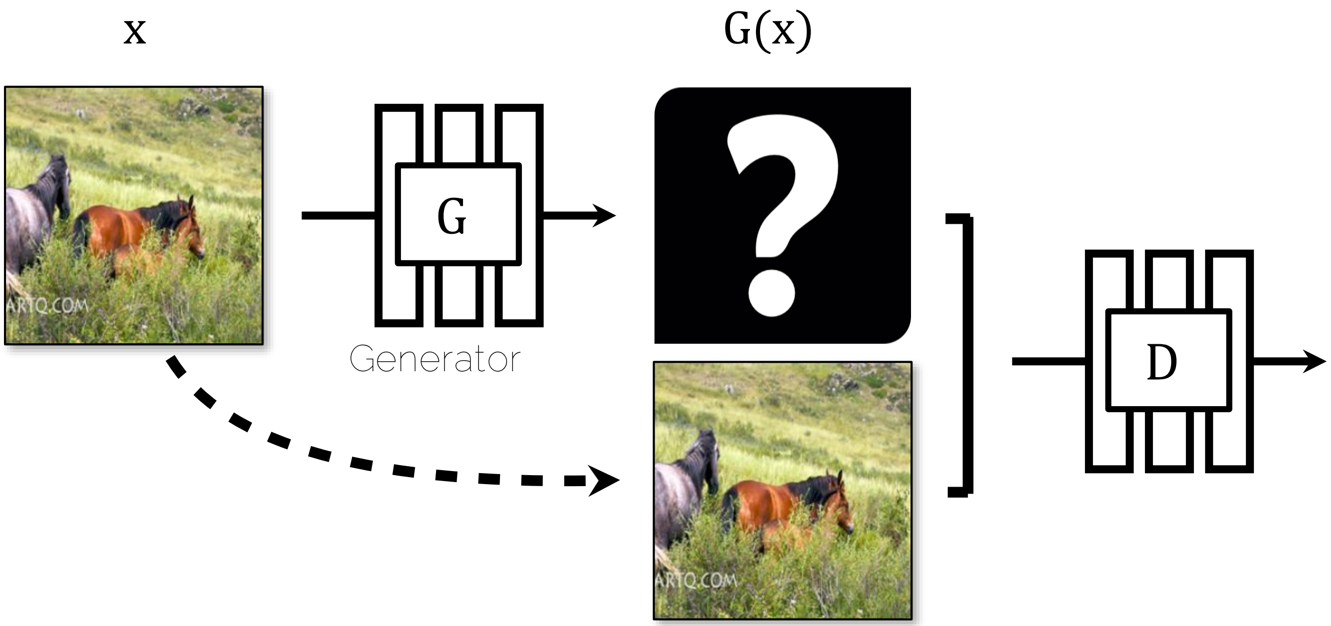
Paired

Unpaired



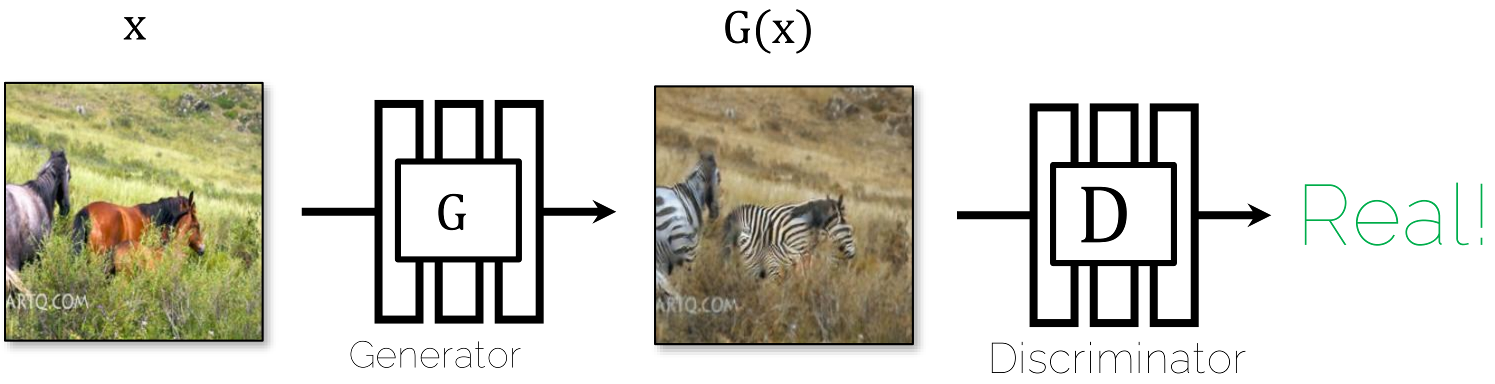
Cycle-Consistent Adversarial Networks

Cycle-Consistent Adversarial Networks

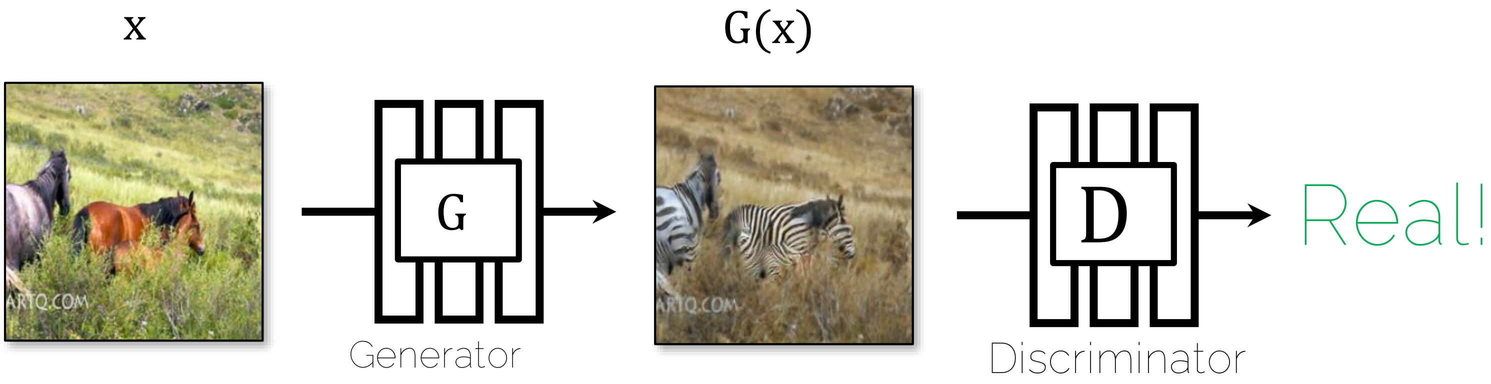


No input-output pairs!

Cycle-Consistent Adversarial Networks

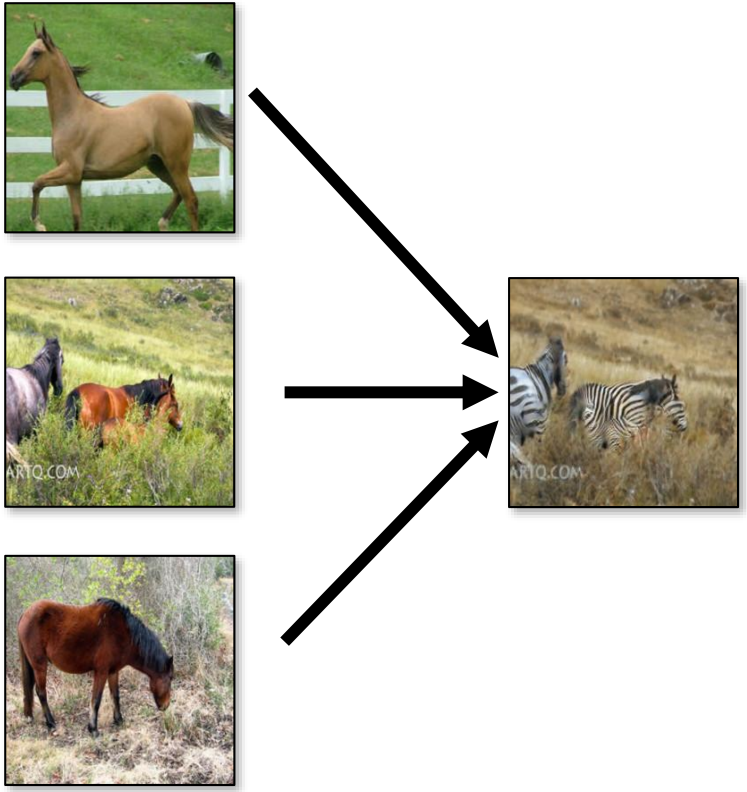


Cycle-Consistent Adversarial Networks



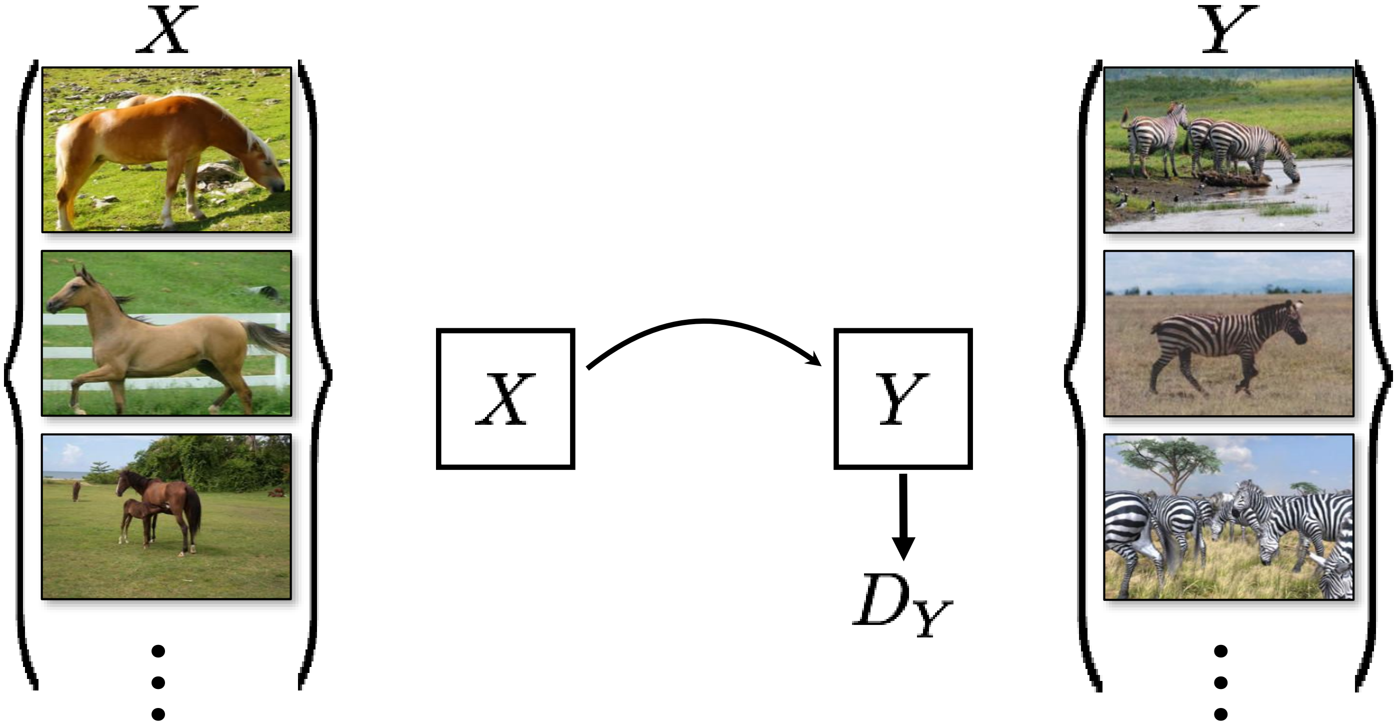
GANs doesn’t force output to correspond to input

Cycle-Consistent Adversarial Networks



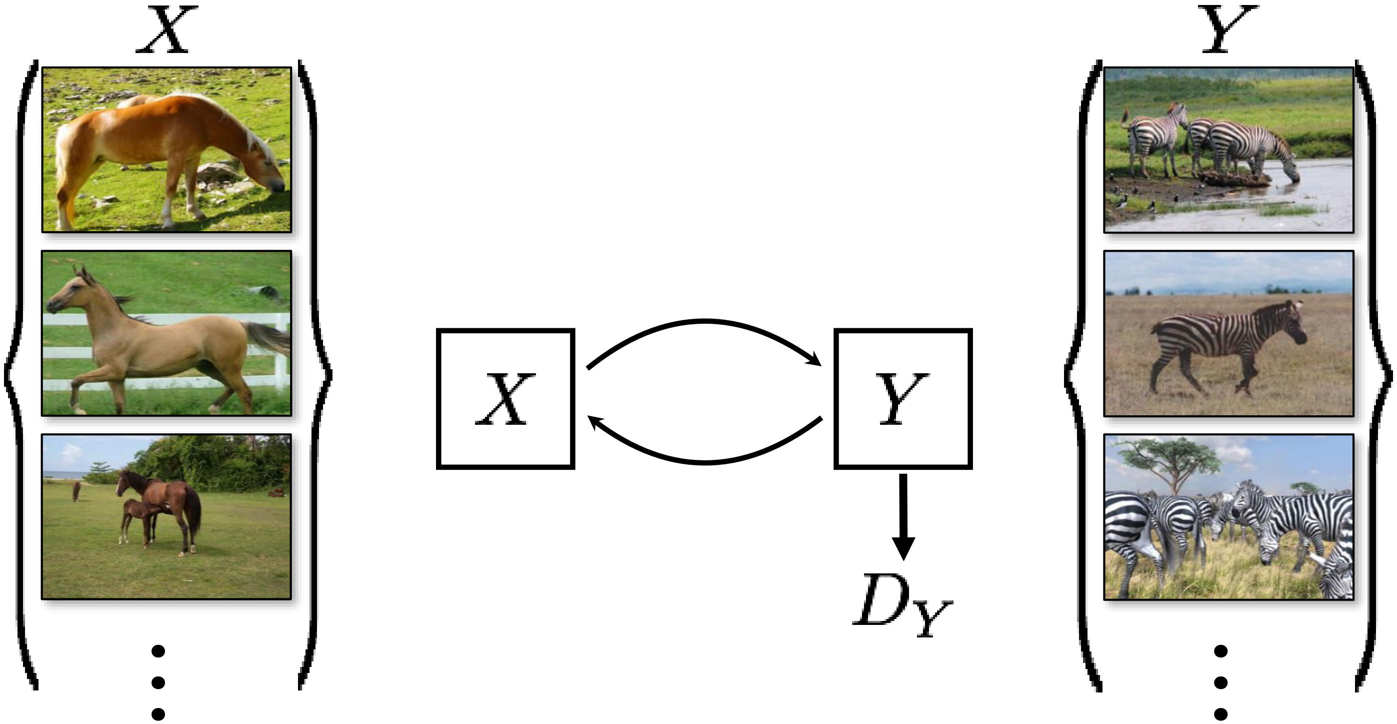
mode collapse!

Cycle-Consistent Adversarial Networks



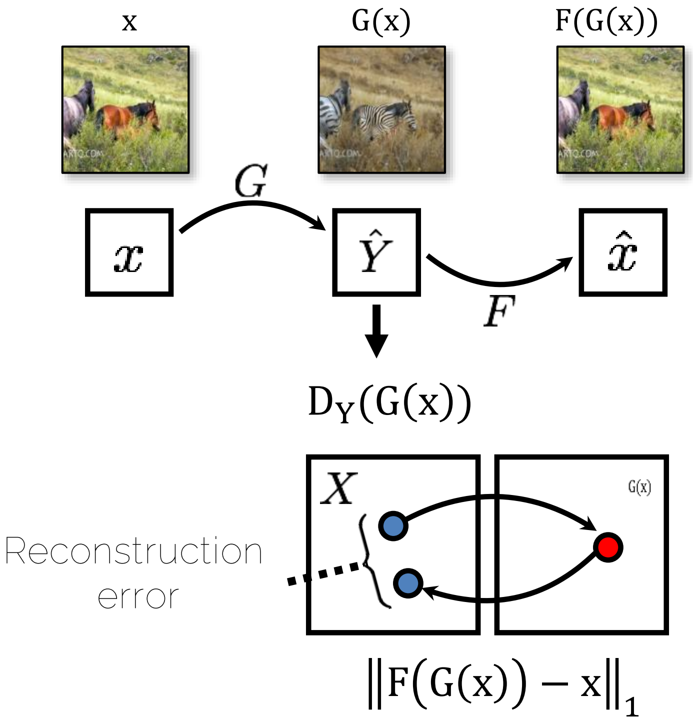
[Zhu et al. 17] Cycle-Consistent Adversarial Networks

Cycle-Consistent Adversarial Networks



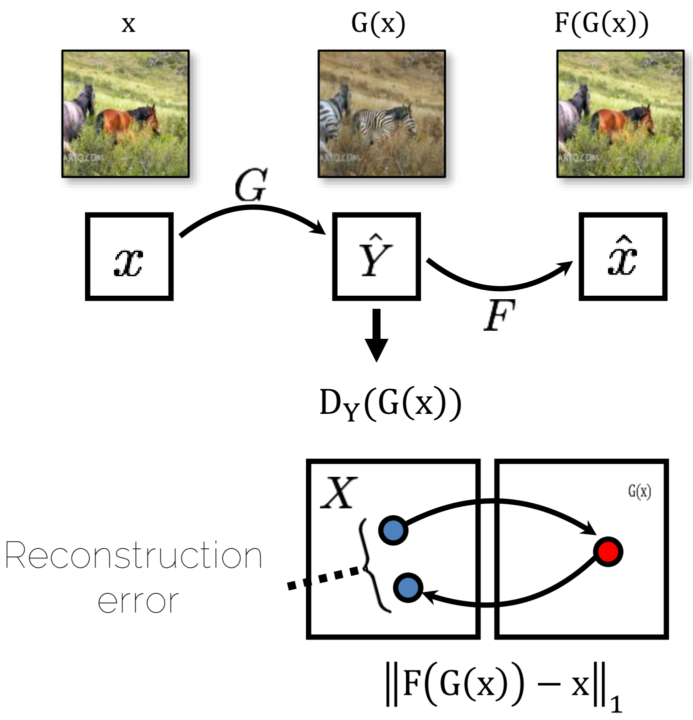
[Zhu et al. 17] Cycle-Consistent Adversarial Networks

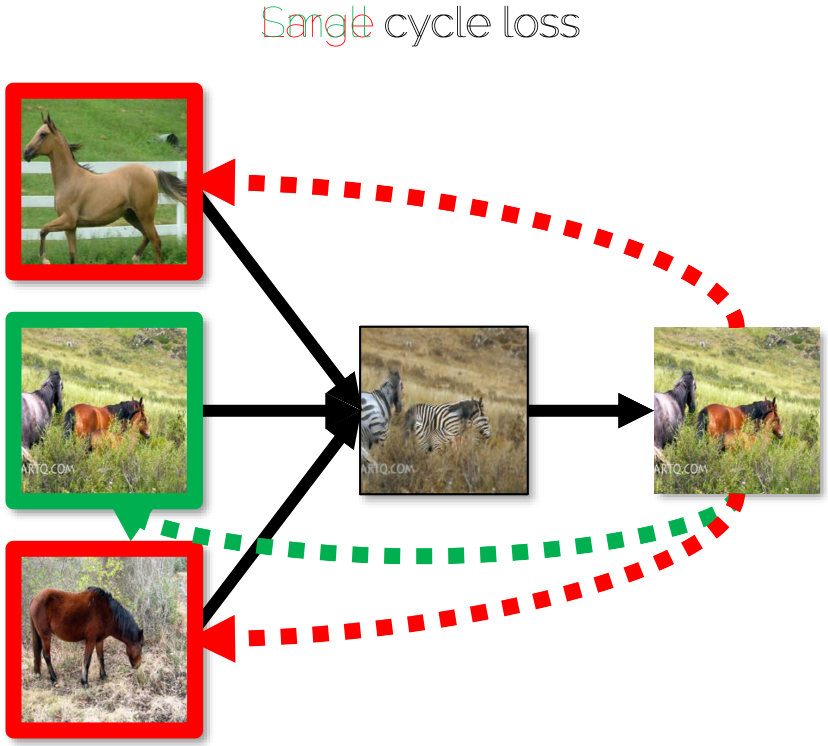
Cycle Consistency Loss



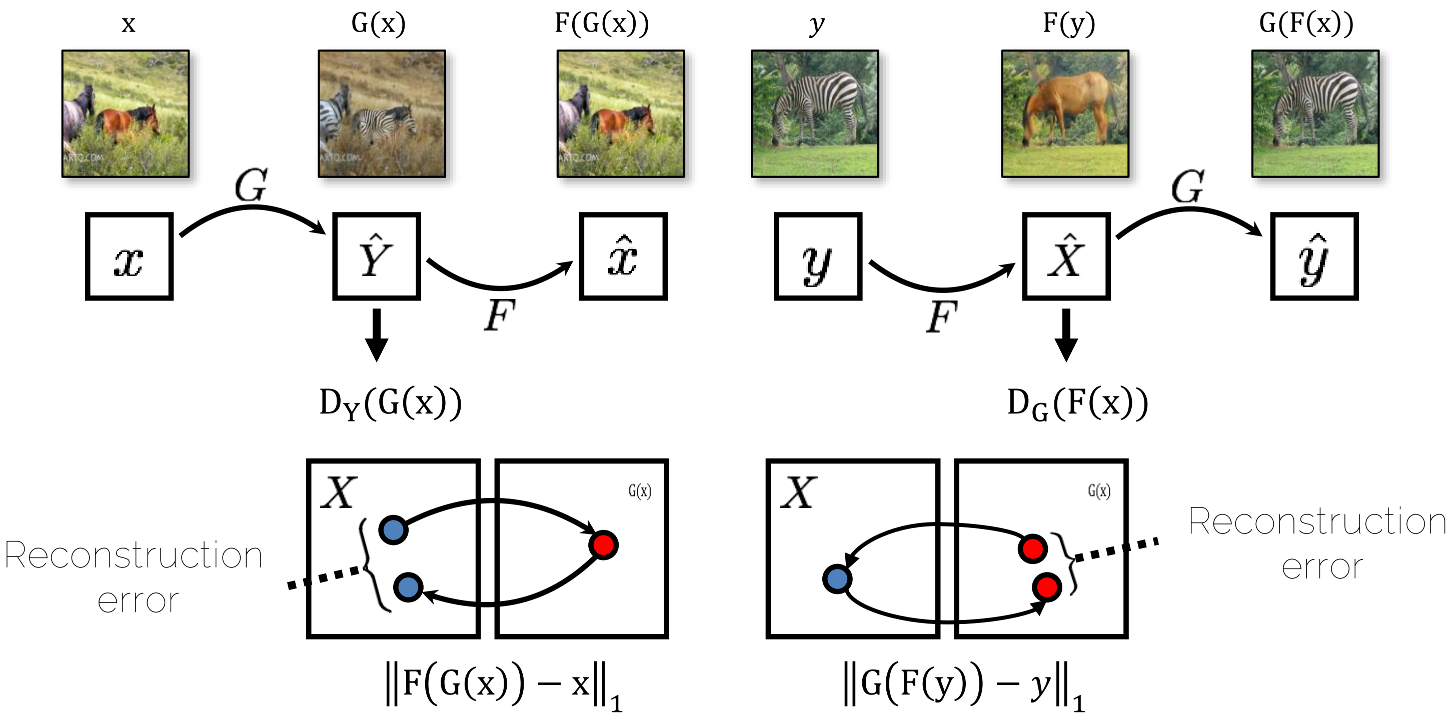
[Zhu et al. 17] Cycle-Consistent Adversarial Networks

Cycle Consistency Loss



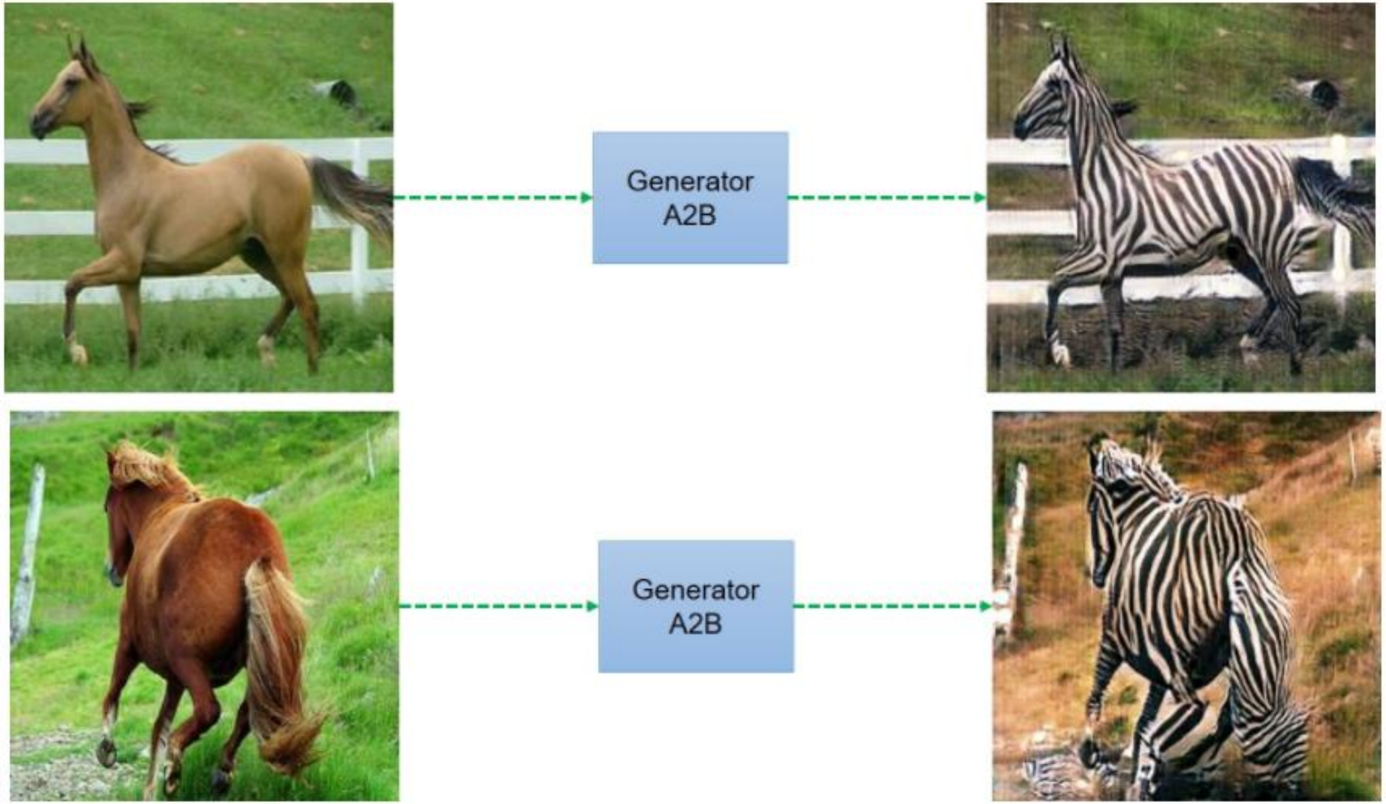


Cycle Consistency Loss



[Zhu et al. 17] Cycle-Consistent Adversarial Networks

Cycle GAN - Overview



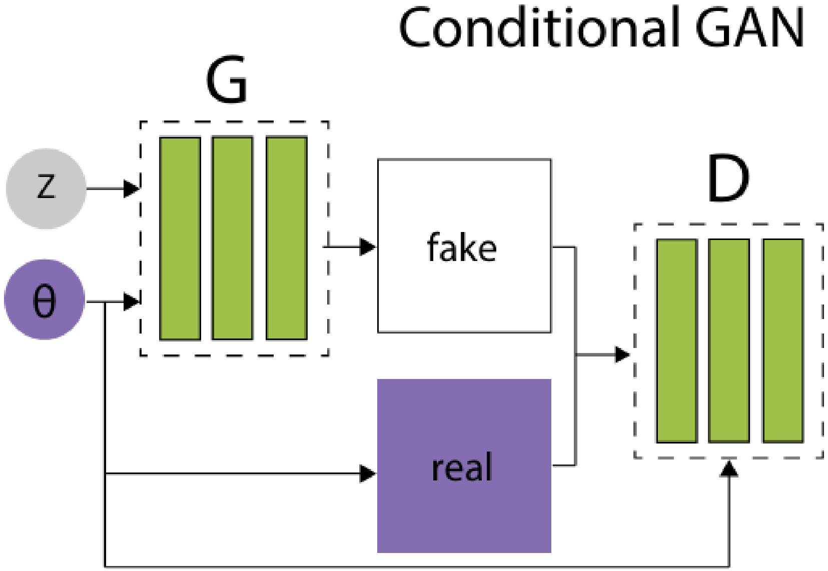
https://junyanz.github.io/CycleGAN/ [Zhu et al. 17.]

Monet's paintings → photos



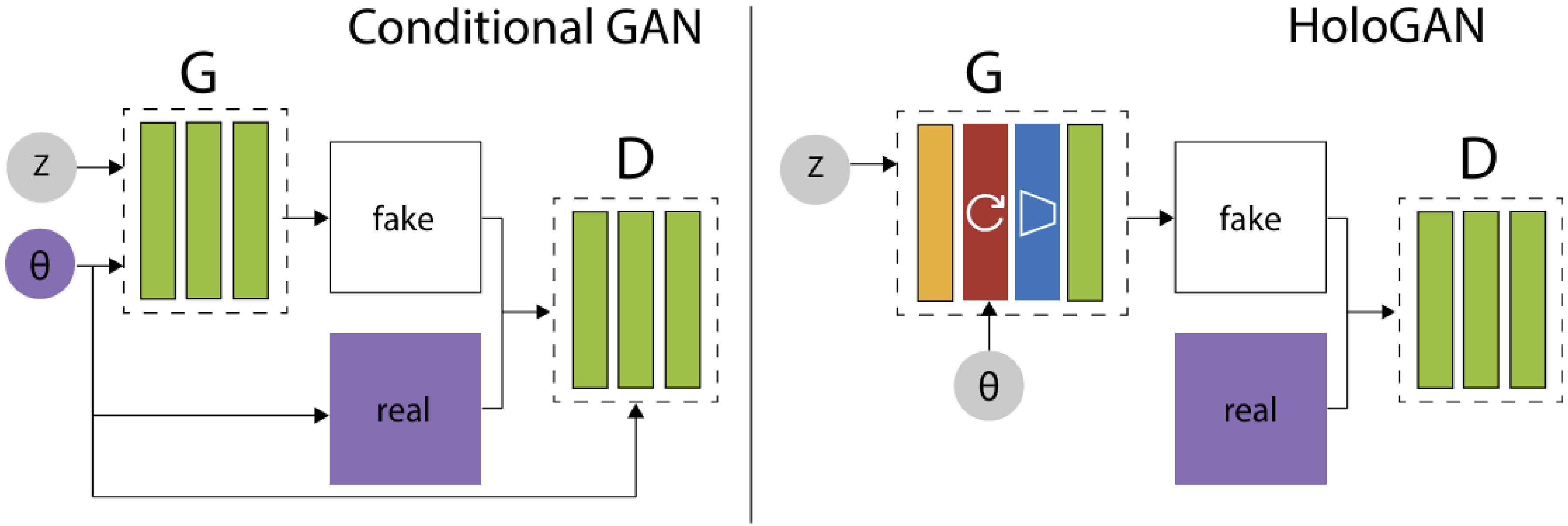

3D Aware GANs

HoloGAN



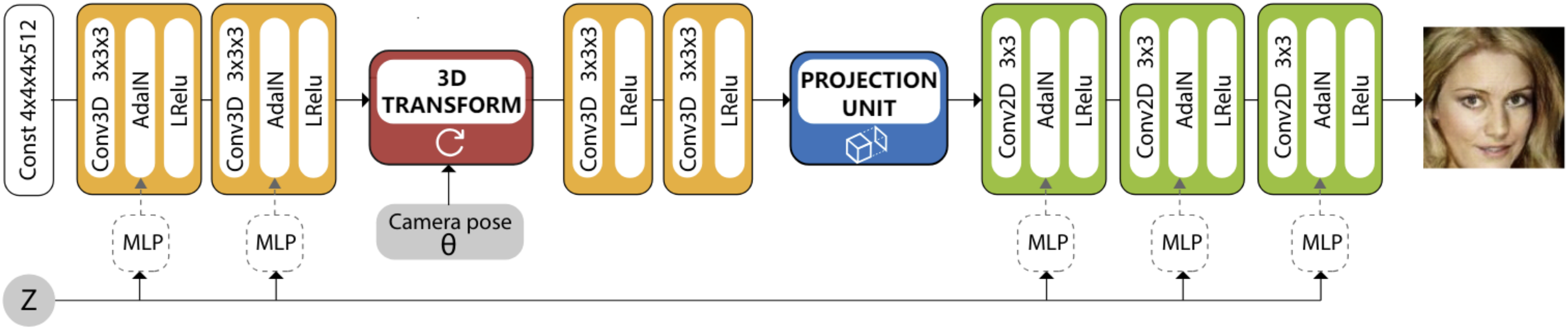
Nguyen-Phuoc et al. 19, HoloGAN

HoloGAN



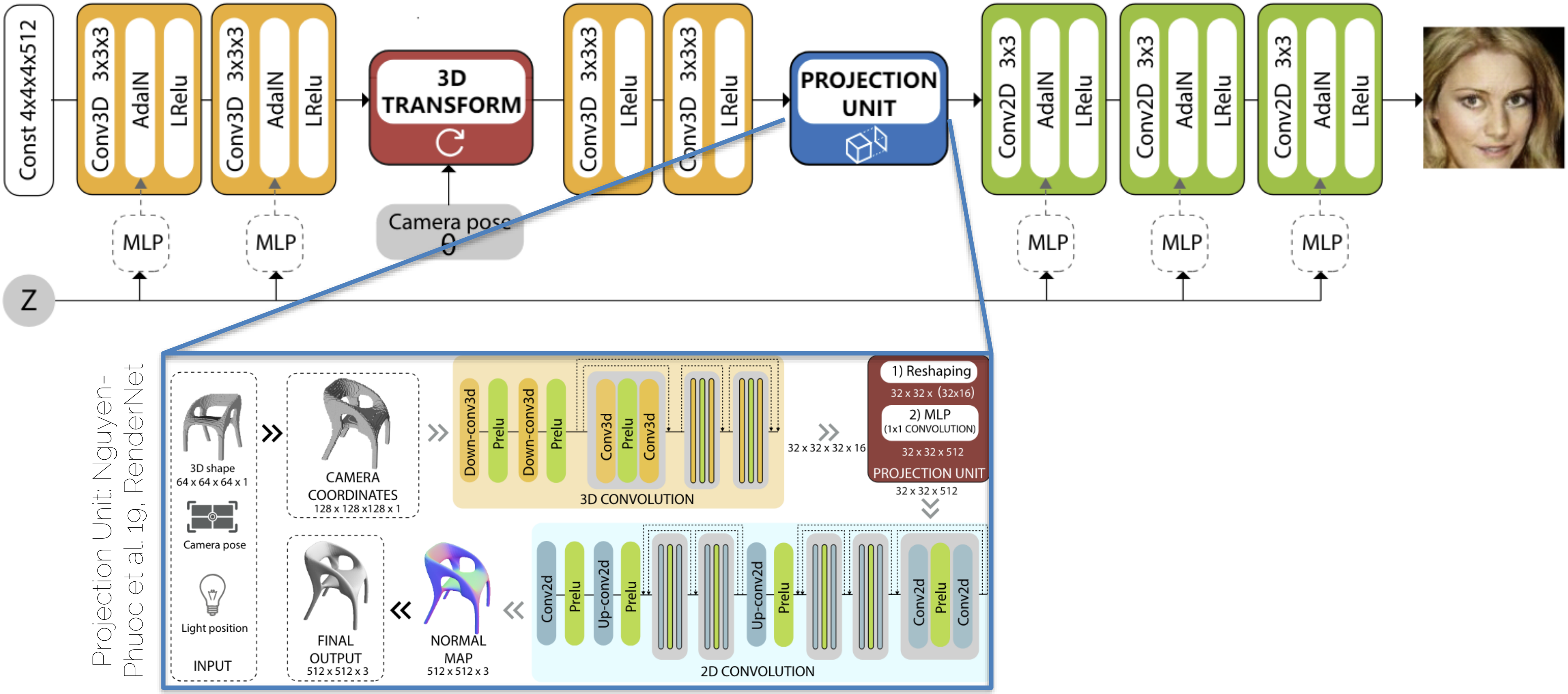
HoloGAN

HoloGAN Generator



HoloGAN

HoloGAN Generator

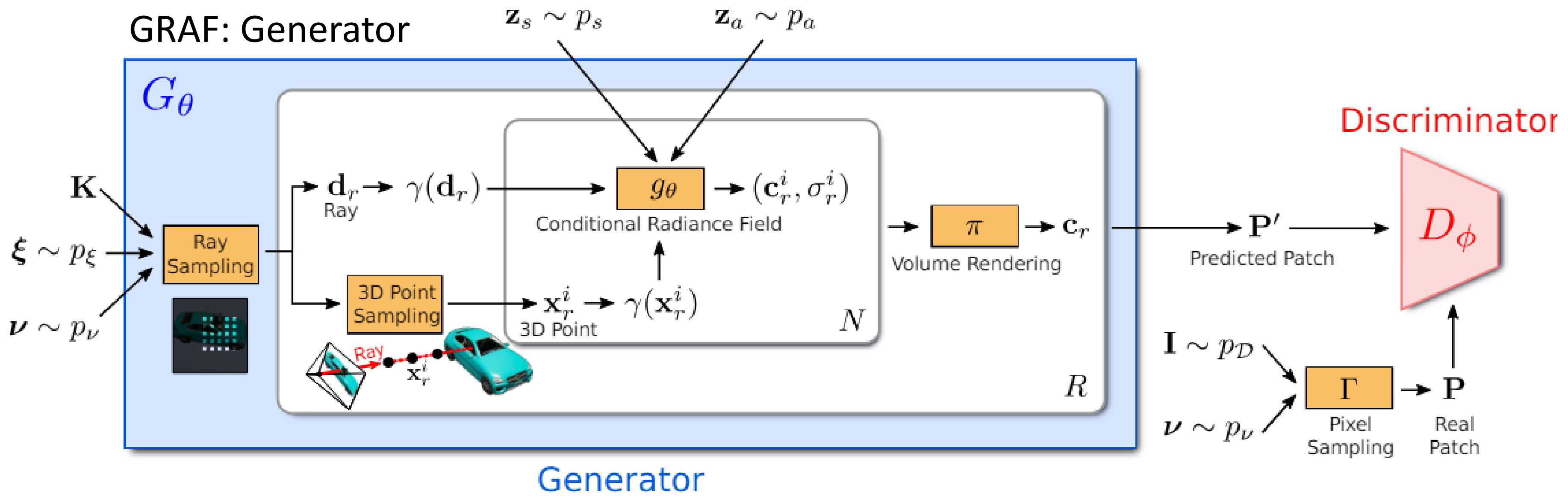


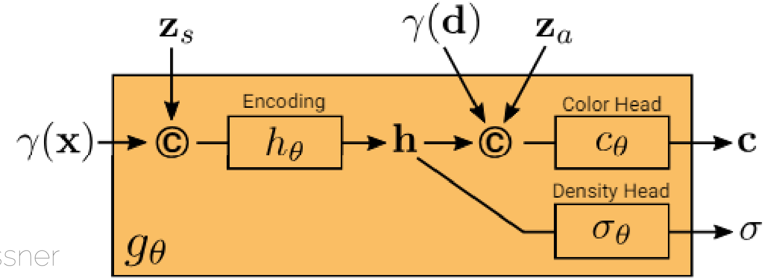
HoloGAN

Nguyen-Phuoc et al. 19, HoloGAN

GRAF: Generative Radiance Field

GRAF: Generator





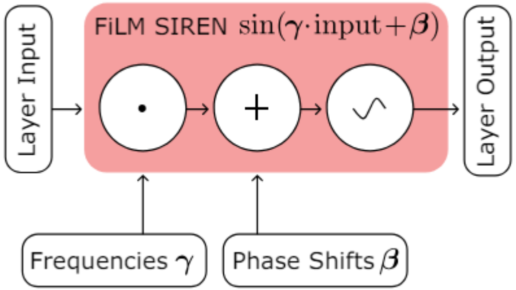
GRAF: Discriminator: 2D Conv Patch D

GRAF: Generative Radiance Field

Schwarz et al. 21: GRAF

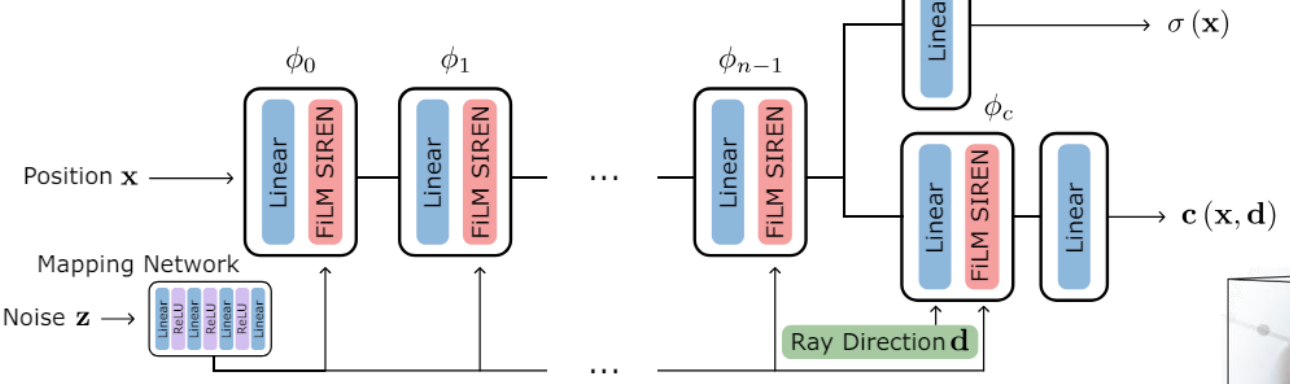
Pi-GAN

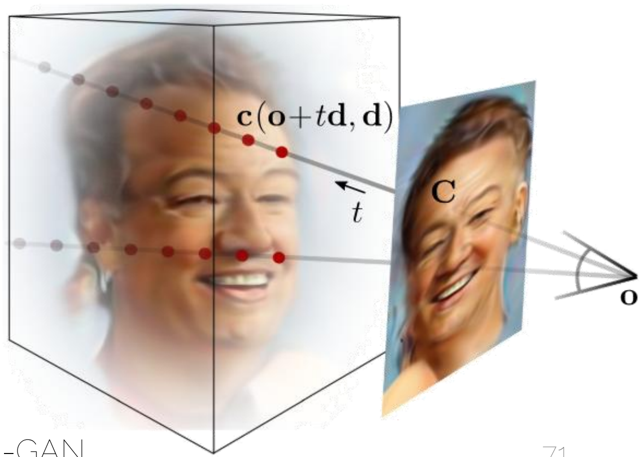
(b)



(a)

Pi-GAN Generator



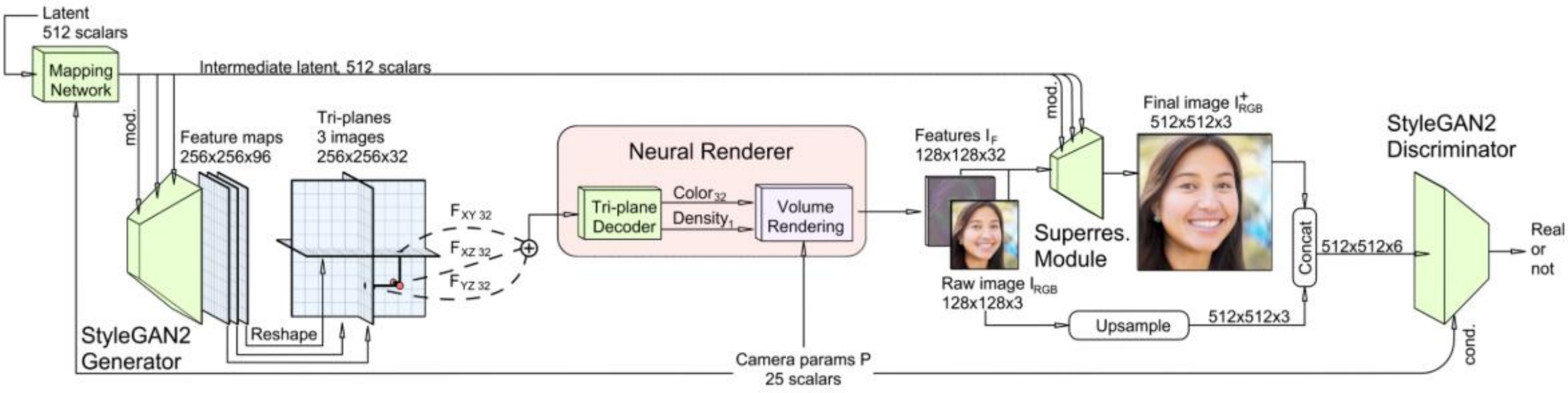


Pi-GAN Discriminator:
Progressive GAN (only grows resolution)

Pi-GAN

Chan et al. 21: Pi-GAN

EG3D: Efficient Geometry-aware 3D Generative Adversarial Network

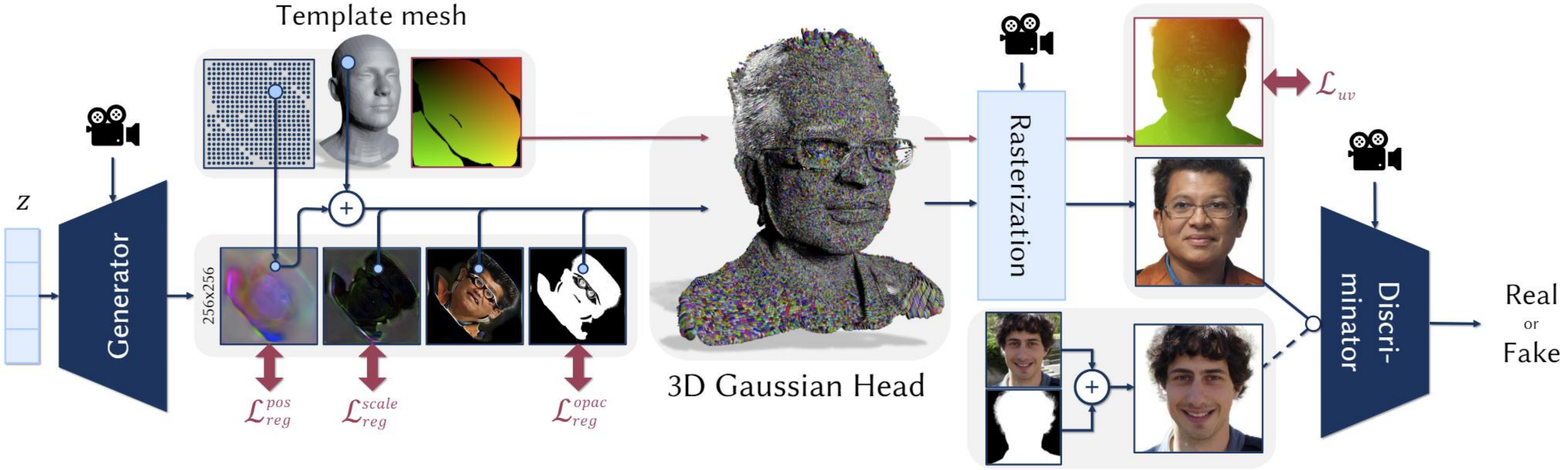


Chan et al. 22: EG3D

EG3D: Efficient Geometry-aware 3D Generative Adversarial Network

Chan et al. 22: EG3D

GGHead: NeRF -> 3DGS



Kirschstein et al. 24: GGHead

Reading Homework

- [Zhu et al. 2016] Generative Visual Manipulation on the Natural Image Manifold
  - https://arxiv.org/abs/1609.03552
- [Isola, et al. 2017] Image-to-image translation with conditional adversarial networks
  - https://phillipi.github.io/pix2pix/
- [Zhu et al. 2017] Unpaired image-to-image translation using cycle-consistent adversarial networks
  - https://arxiv.org/abs/1703.10593

Thanks for watching!