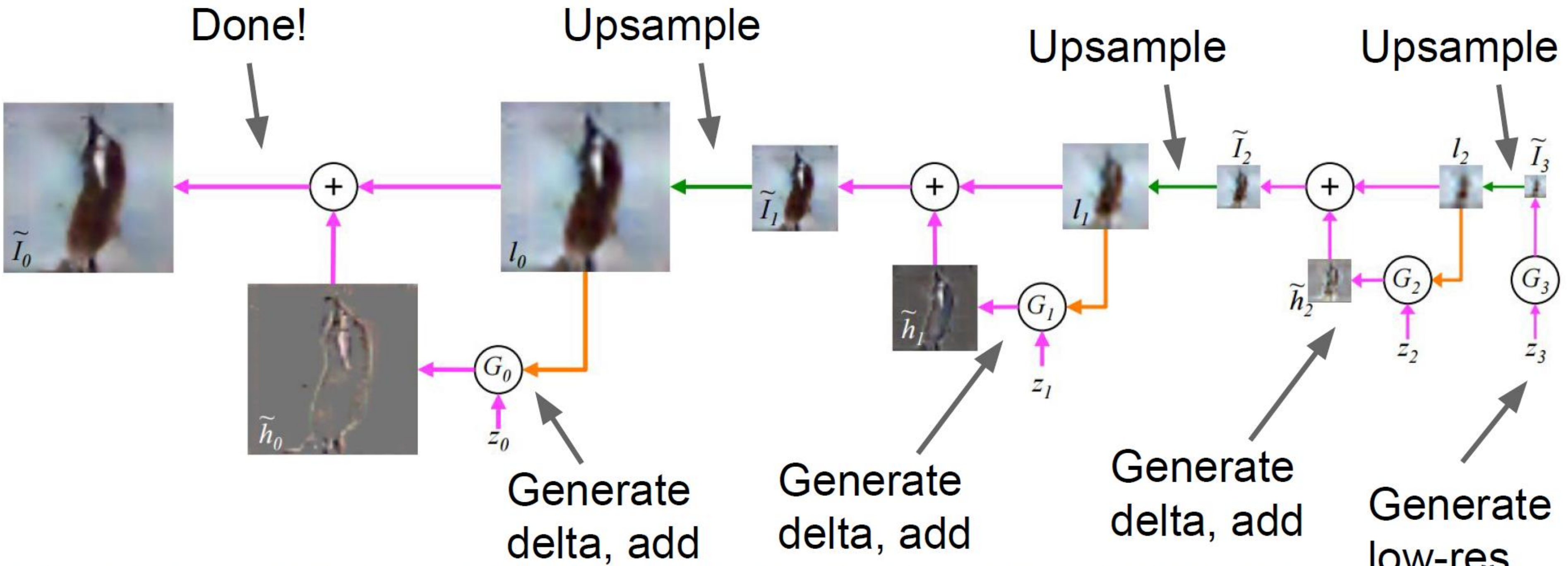
GAN Architectures and Conditional GANs

GAN Architectures

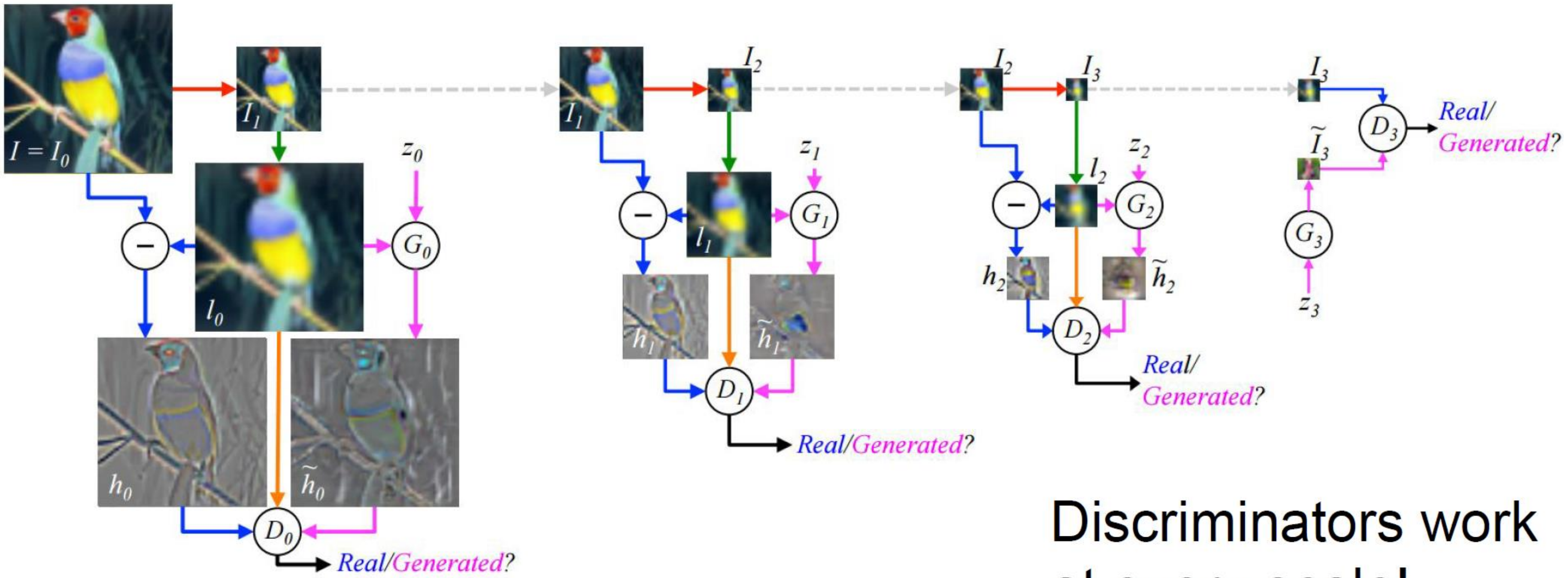
Multiscale GANs



Denton et al, "Deep generative image models using a Laplacian pyramid of adversarial networks", NIPS 2015

Credit: Li/Karpathy/Johnson

Multiscale GANs

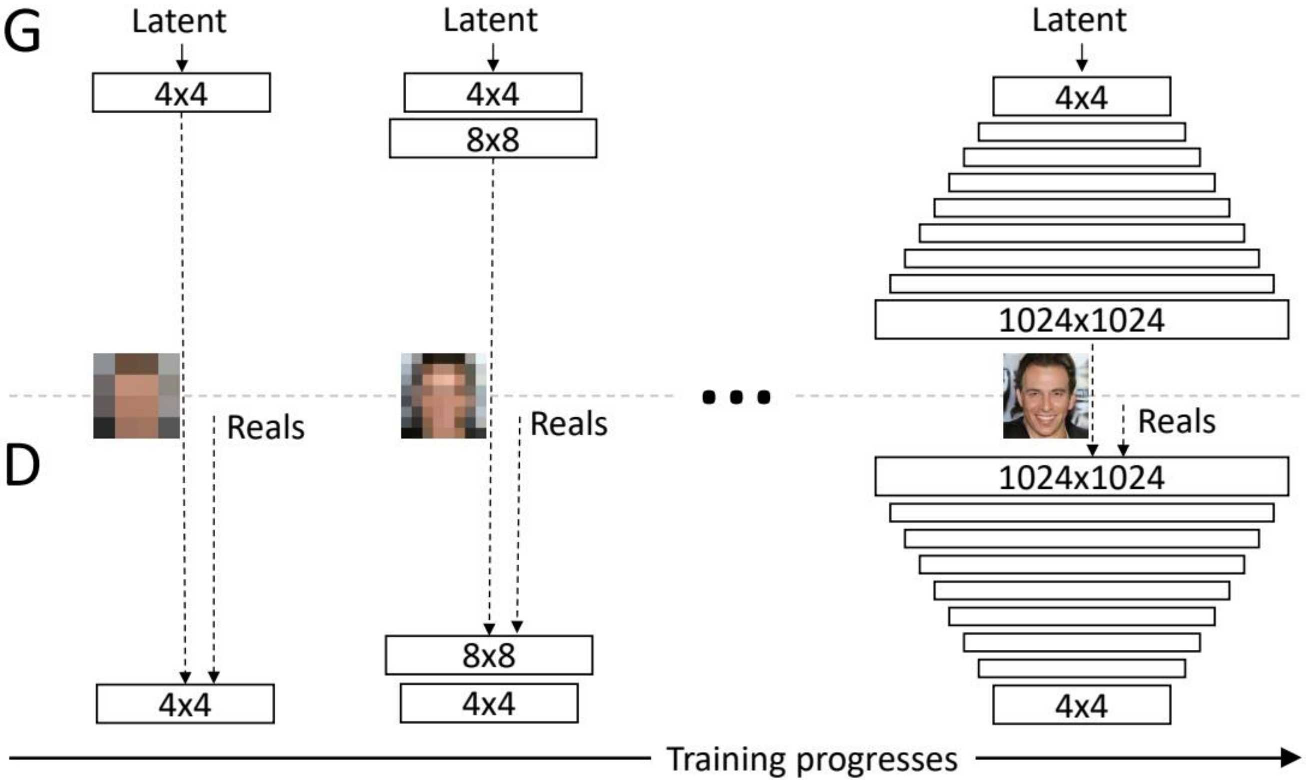


Denton et al, NIPS 2015

Discriminators work
at every scale!

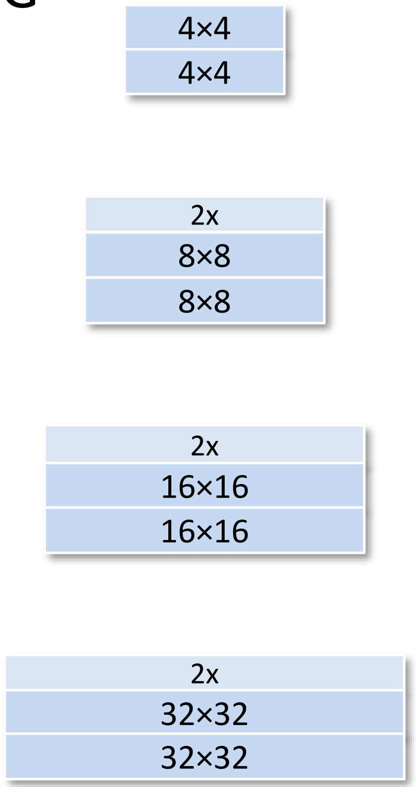
Credit: Li/Karpathy/Johnson

Progressive Growing GANs



https://github.com/tkarras/progressive_growing_of_gans [Karras et al. 17]

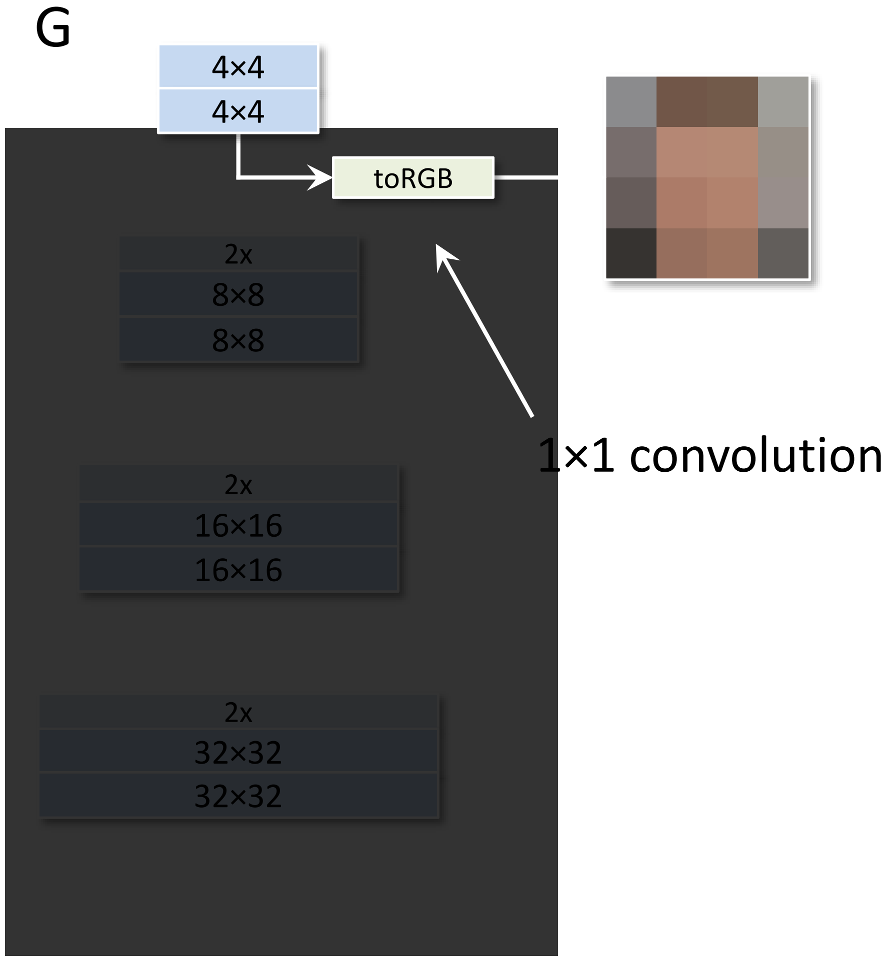
G



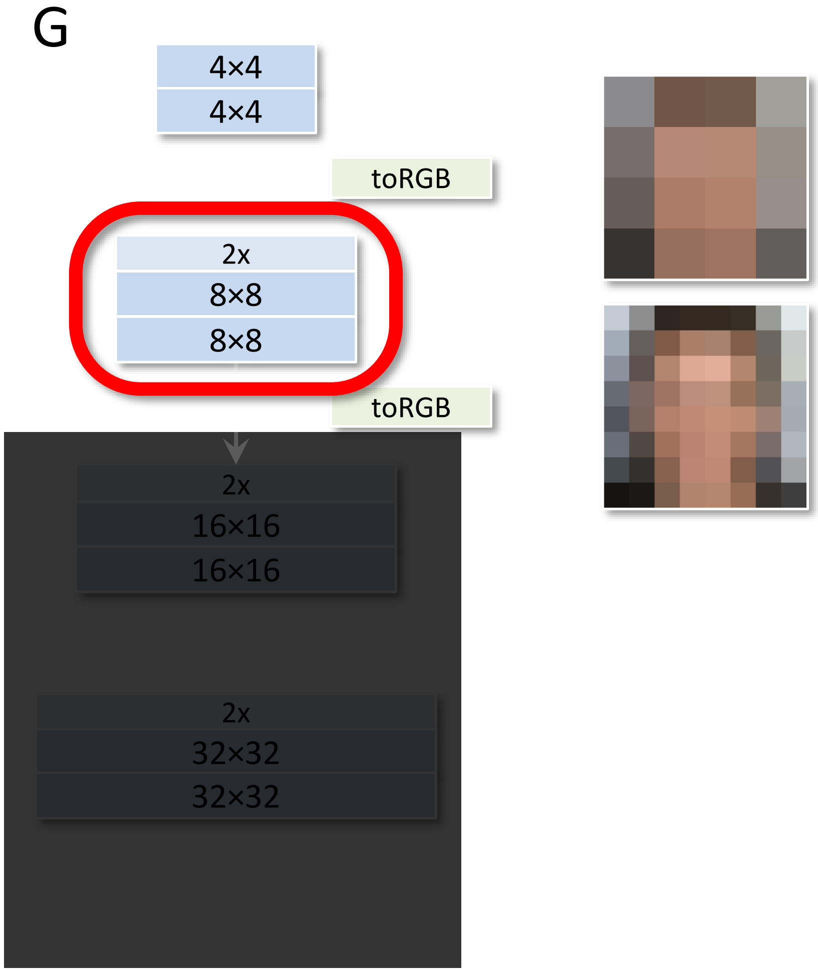
Replicated block

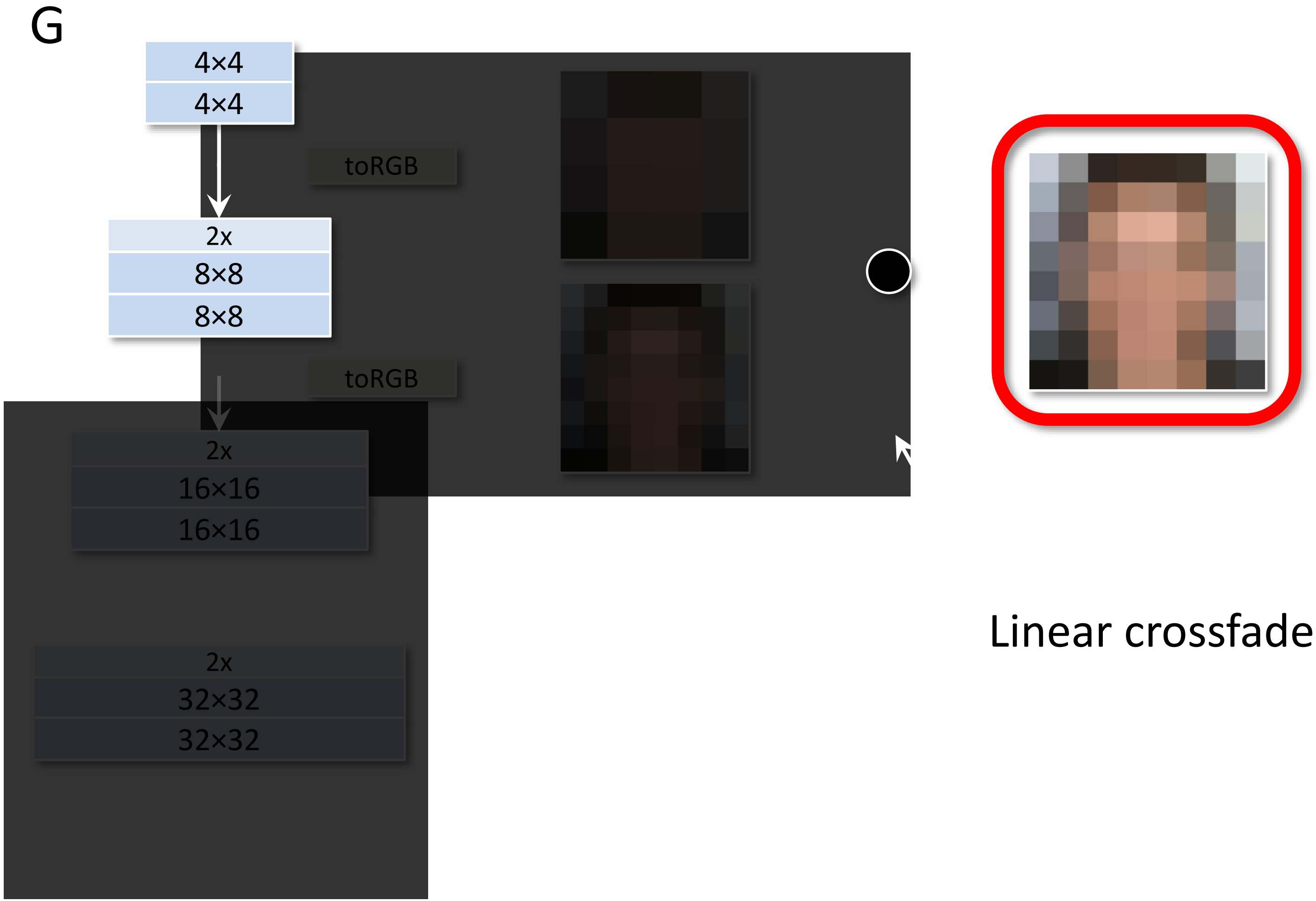
Nearest-neighbor upsampling

3×3 convolution

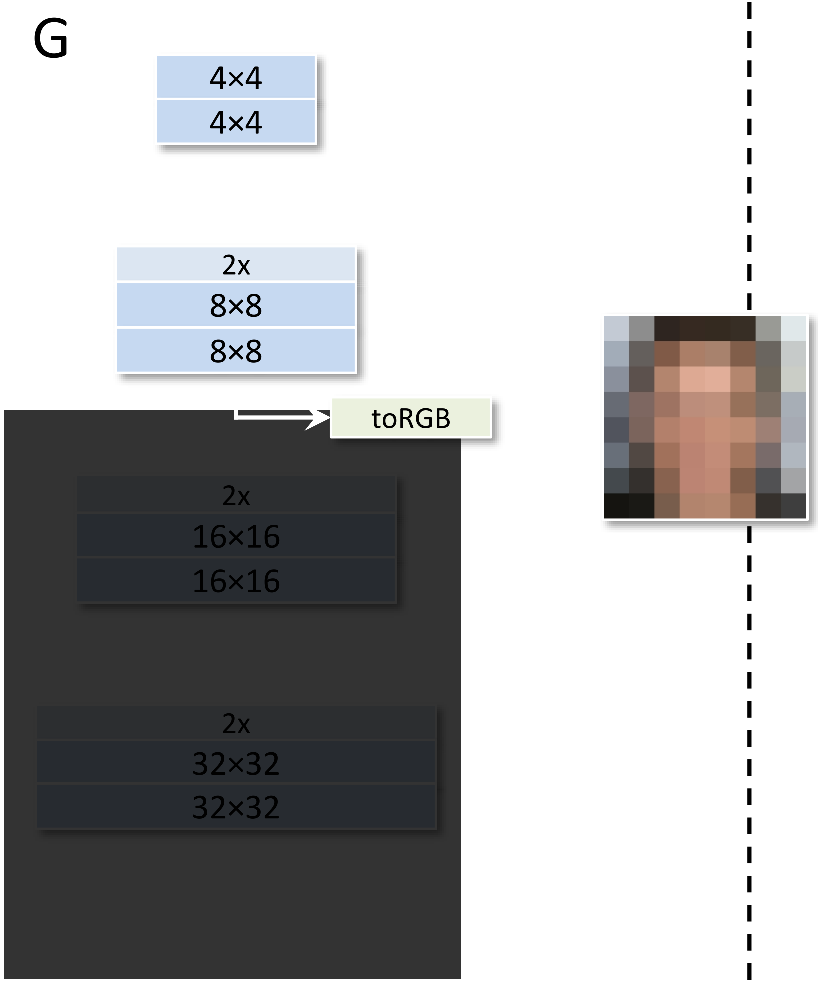


G
4×4
4×4
toRGB
2x
8×8
8×8
2x
16×16
16×16
2x
32×32
32×32
1×1 convolution


G
4×4
4×4
toRGB
2x
8×8
8×8
toRGB
2x
16×16
16×16
2x
32×32
32×32

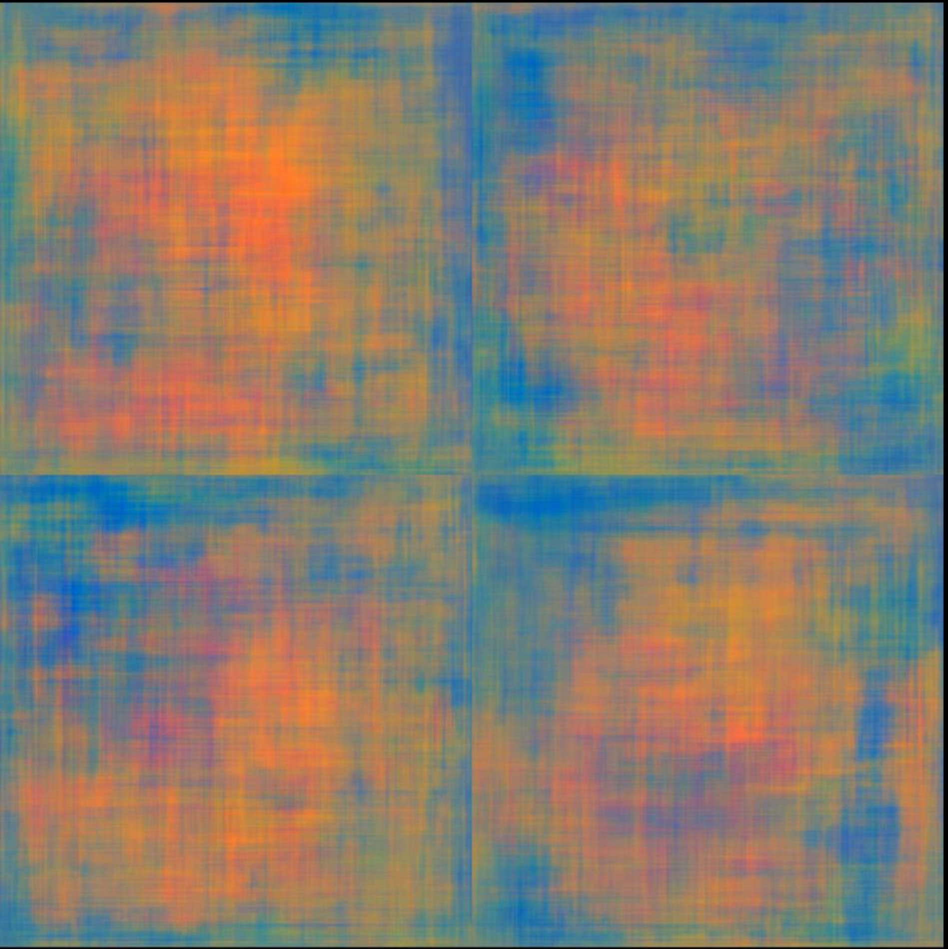


Linear crossfade

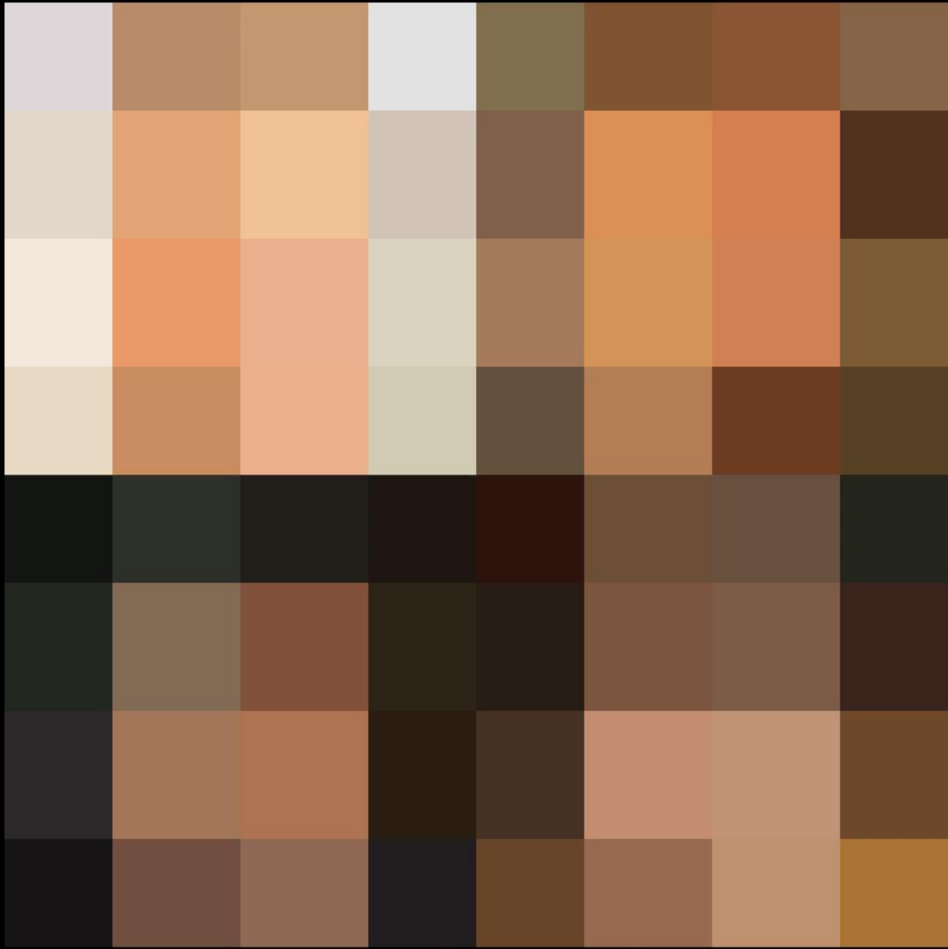

G
4×4
4×4
2x
8×8
8×8
toRGB
2x
16×16
16×16
2x
32×32
32×32


G
4×4
4×4
2x
8×8
8×8
toRGB
2x
16×16
16×16
2x
32×32
32×32
D
32×32
32×32
0.5x
16×16
16×16
0.5x
fromRGB
8×8
8×8
0.5x
4×4
4×4

5 min 00 sec

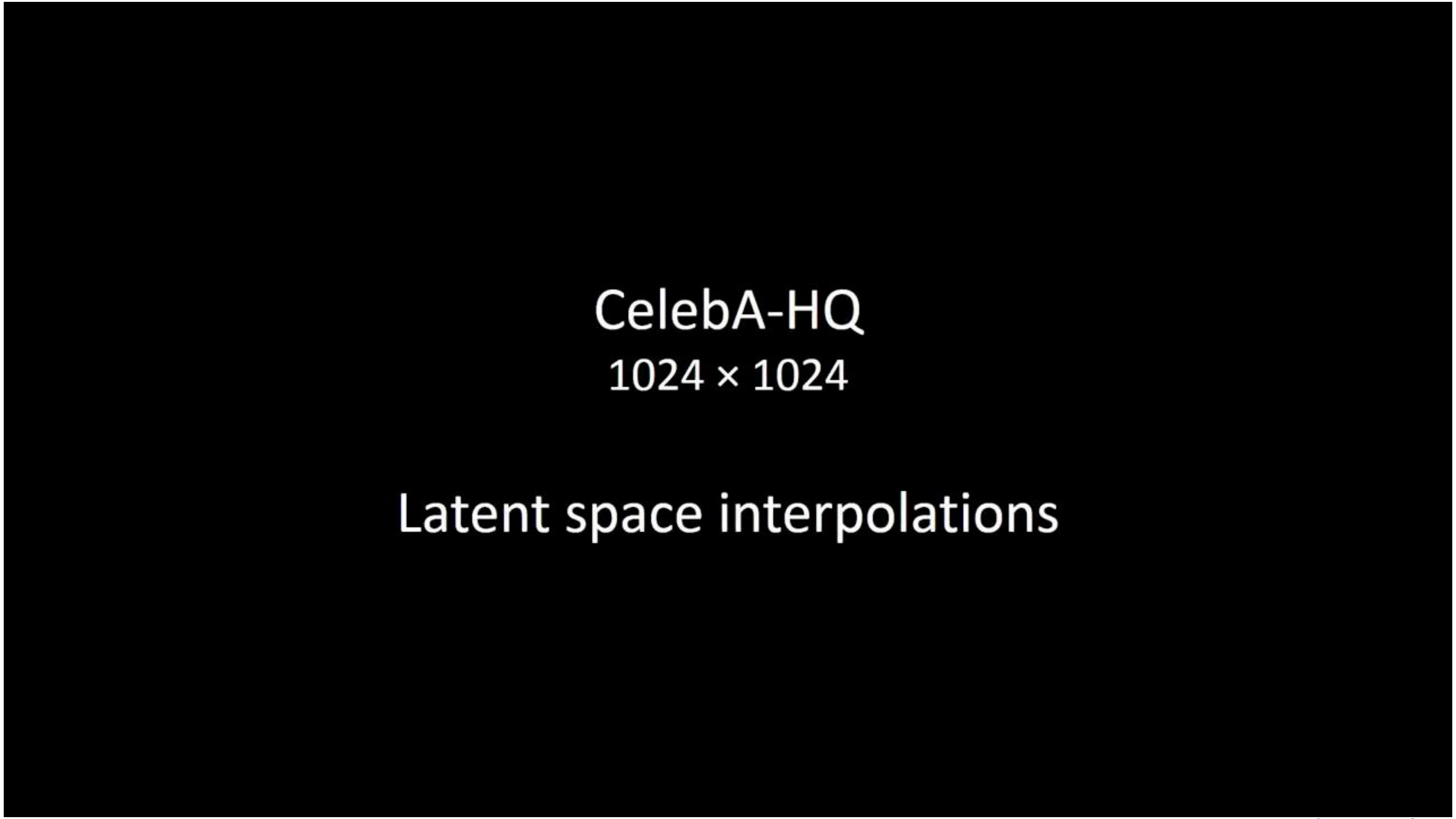


Fixed resolution



Progressive growing

Progressive Growing GANs



https://github.com/tkarras/progressive_growing_of_gans [Karras et al. 17]

Lots of GAN Variations

- Hundreds of GAN papers in the last two years
  - > Mostly with different losses
  - > Extremely hard to train and evaluate

Are GANs Created Equal? A Large-Scale Study

Mario Lucic* Karol Kurach* Marcin Michalski Sylvain Gelly Olivier Bousquet
Google Brain

Abstract

Generative adversarial networks (GAN) are a powerful subclass of generative models. Despite a very rich research

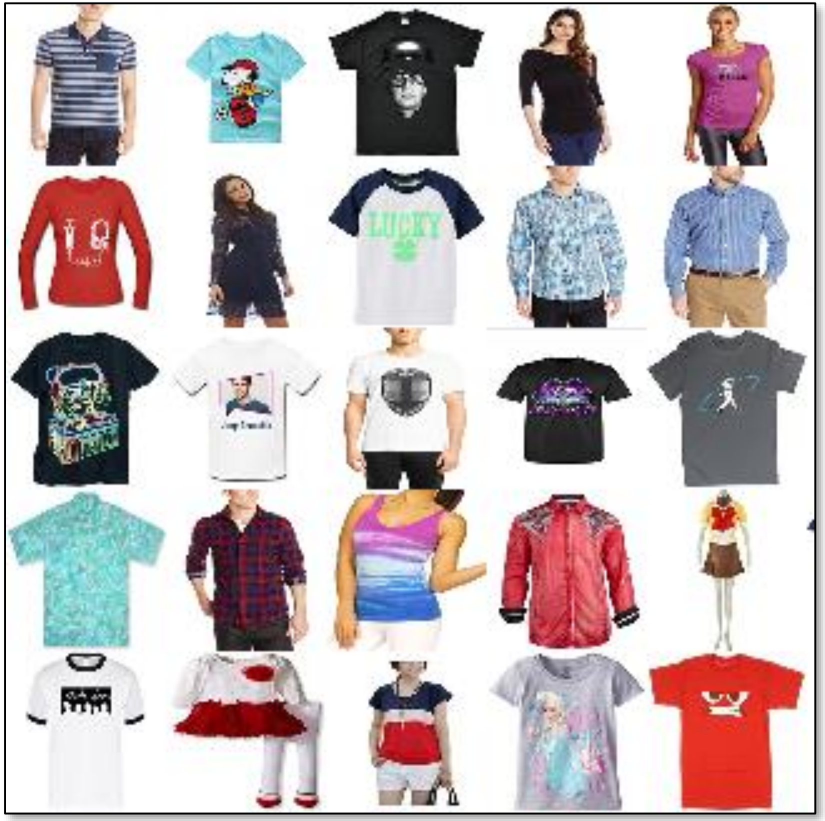
GAN algorithm(s) perform objectively better than the others. That's partially due to the lack of robust and consistent metric, as well as limited comparisons which put all algorithms on equal footage, including the computational

Conditional Generative Adversarial Networks (cGANs)

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

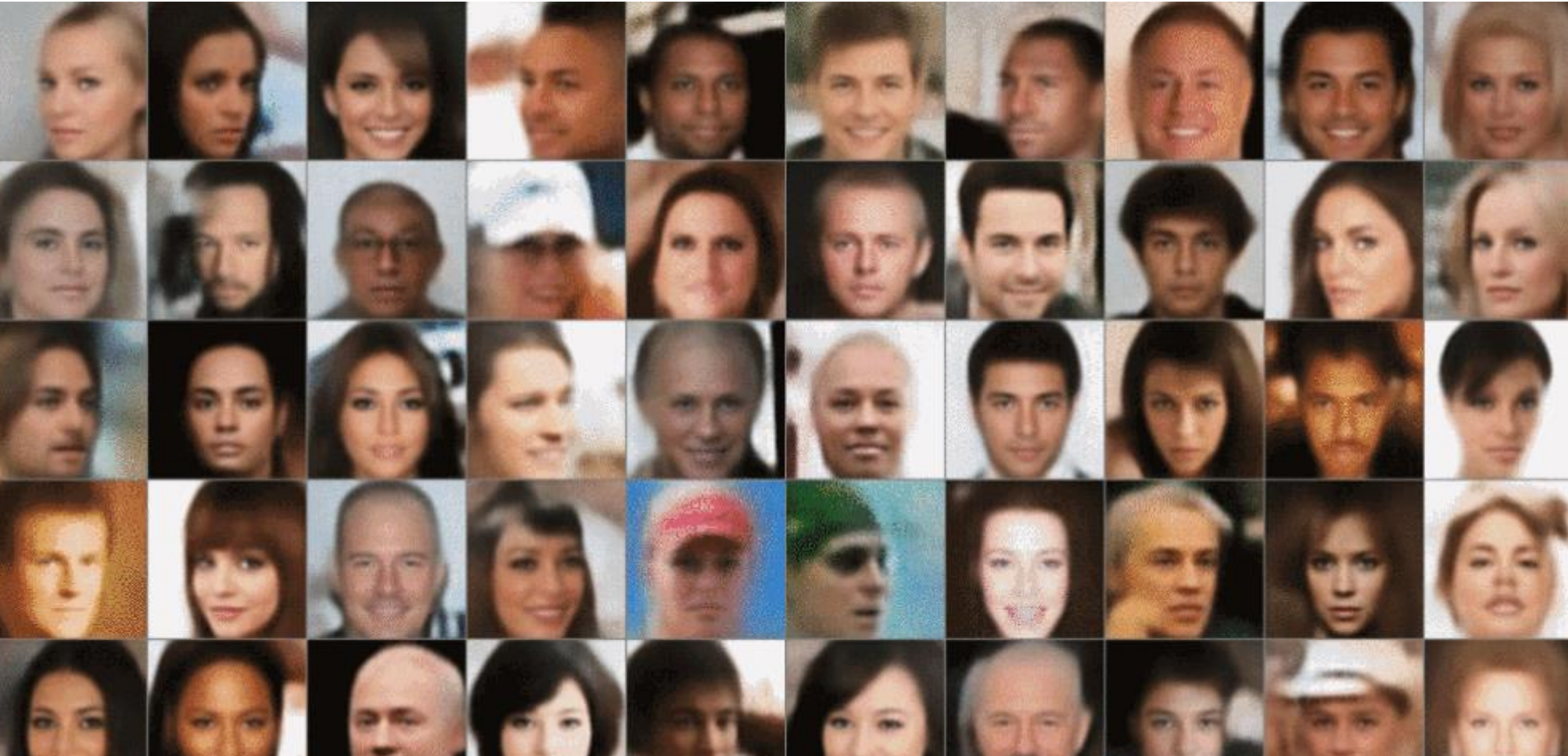
Sampled Data -> G(z)

[Radford et al. 15]

GAN Manifold

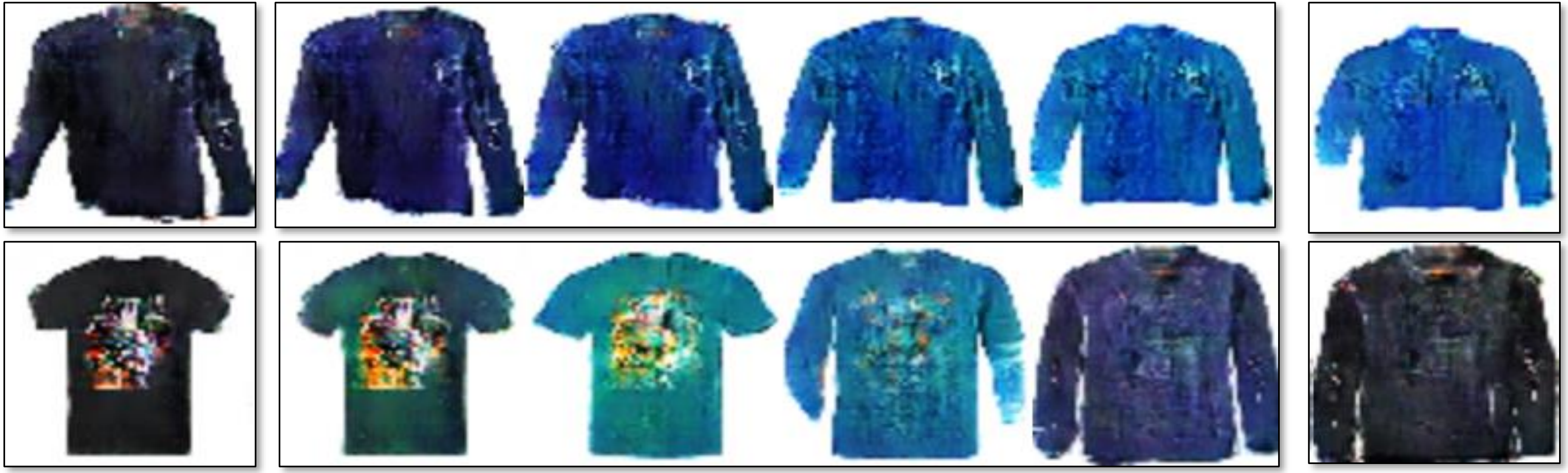
[Bojanowski et al 17]

GAN Manifold



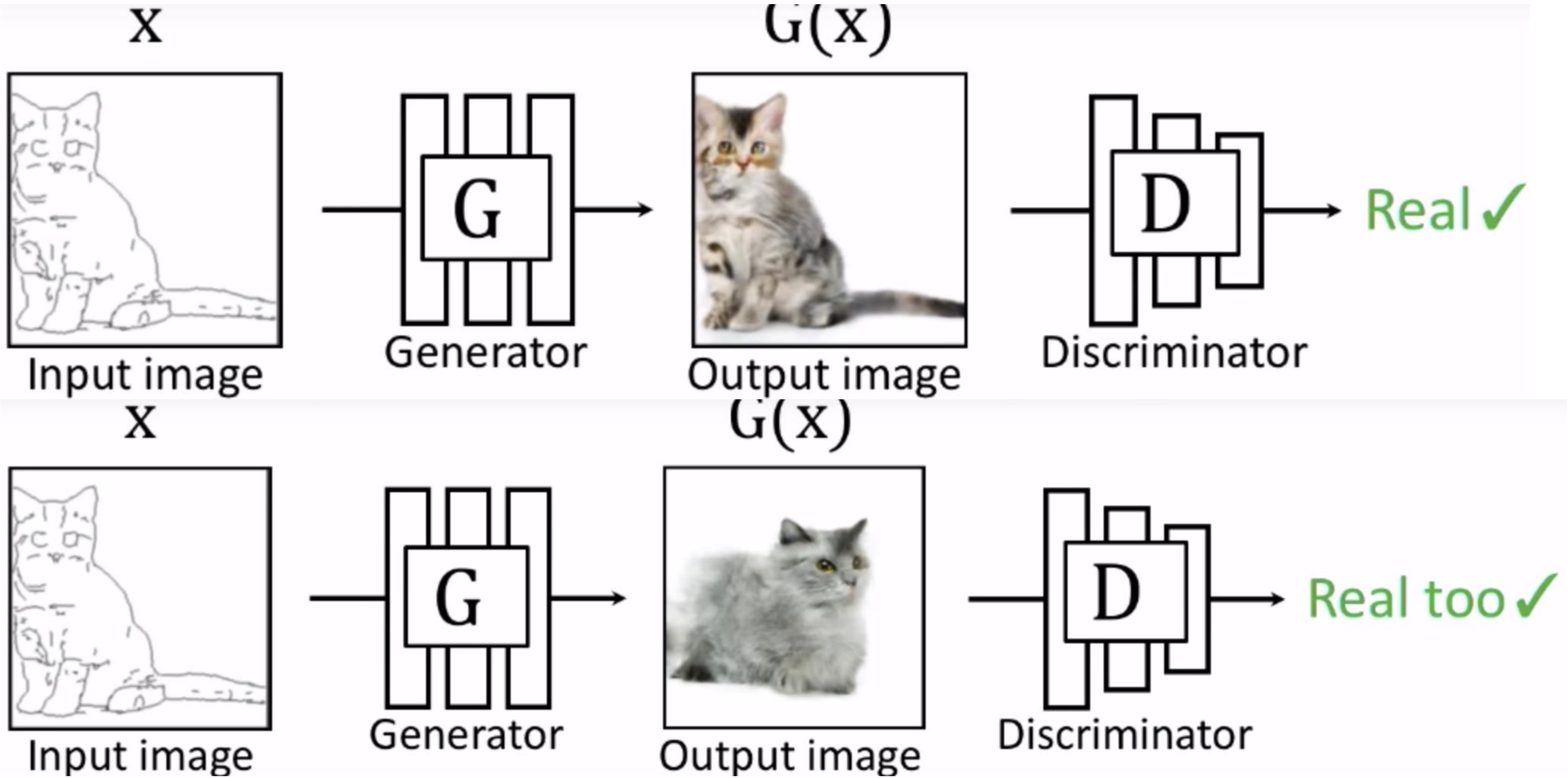
GAN Manifold

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

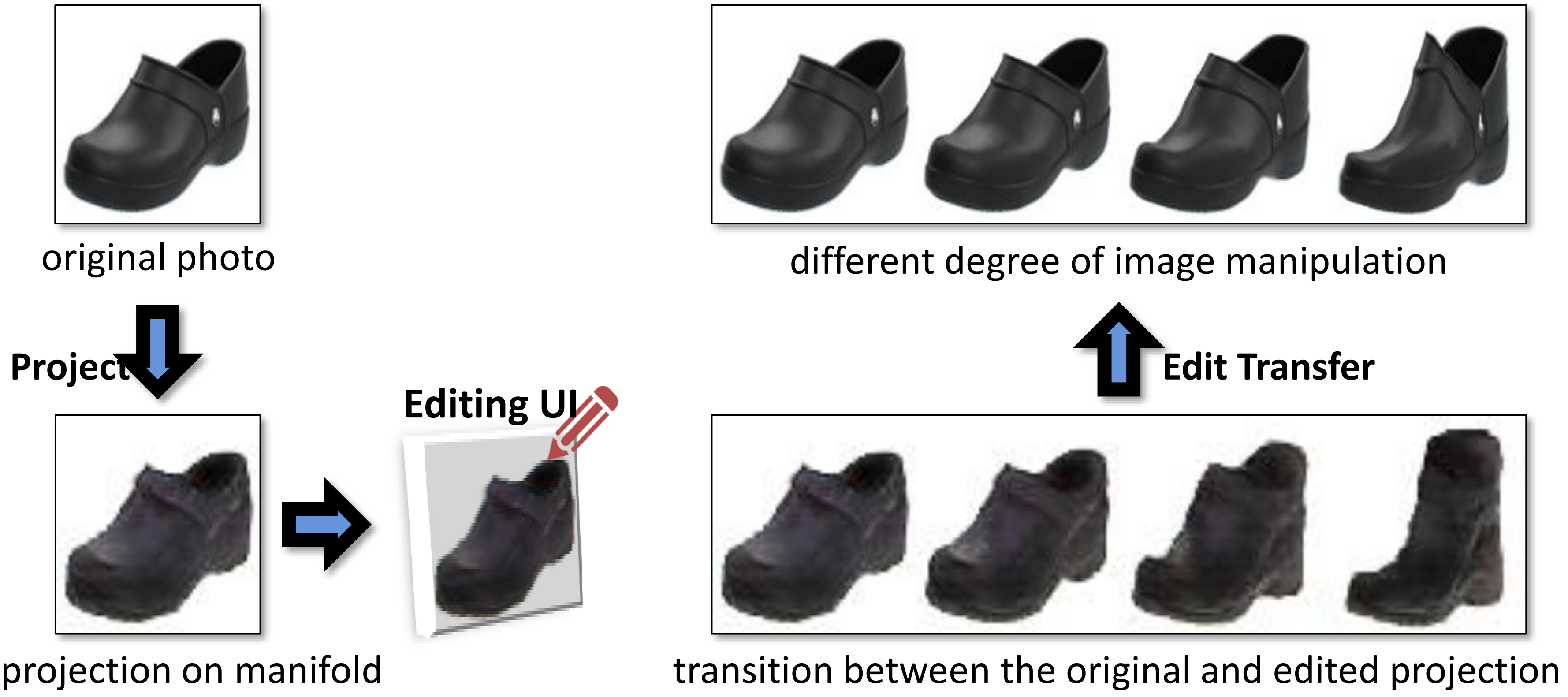


[Radford et al. 15]

Conditional GANs (cGANs)

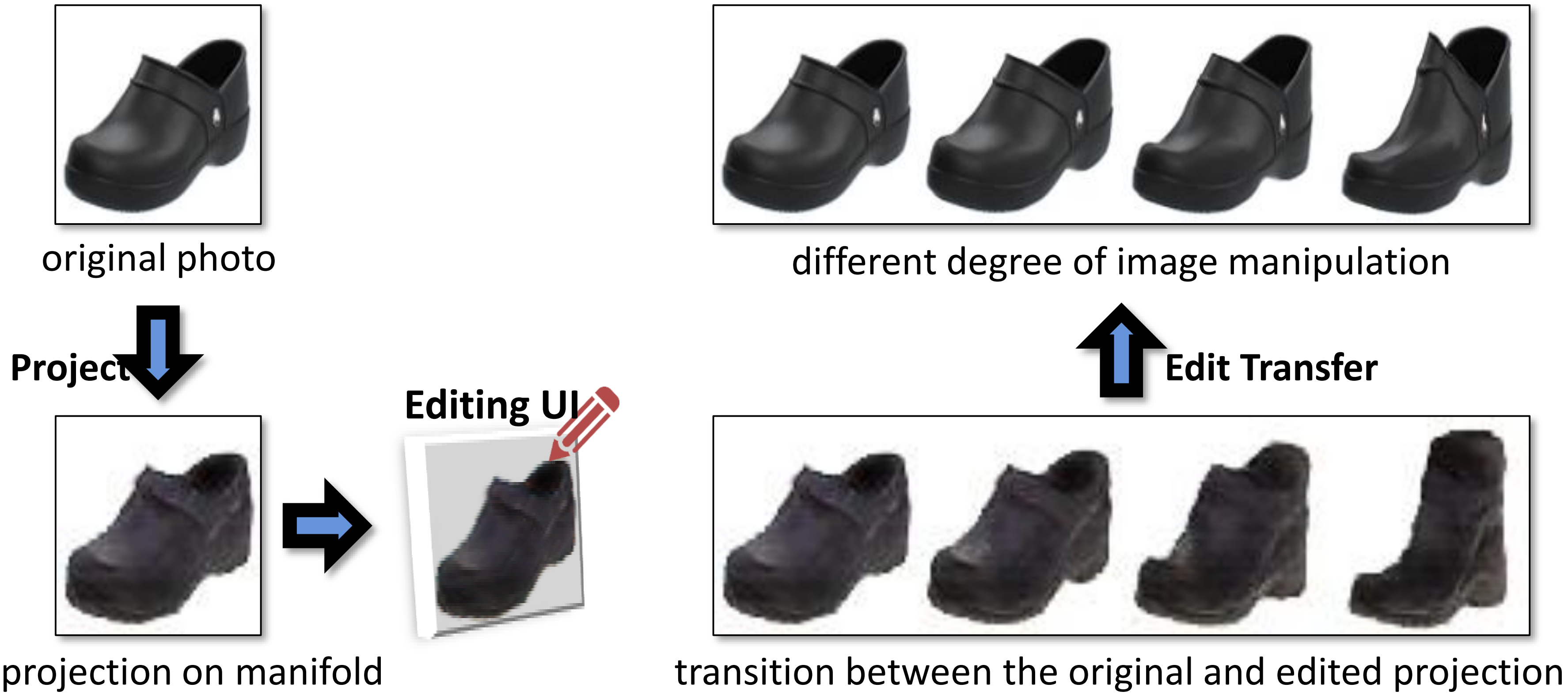


iGANs: Overview



Slide credit Zhu / [Zhu et al. 16]

iGANs: Overview



Slide credit Zhu / [Zhu et al. 16]

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 x^R

Output: latent vector z

Optimization

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

Inverting Network $z = P(x)$

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

with a fixed decoder G

Slide credit Zhu / [Zhu et al. 16]

iGANs: Projecting an Image onto the Manifold

Input: real image x^R

Output: latent vector z

Optimization

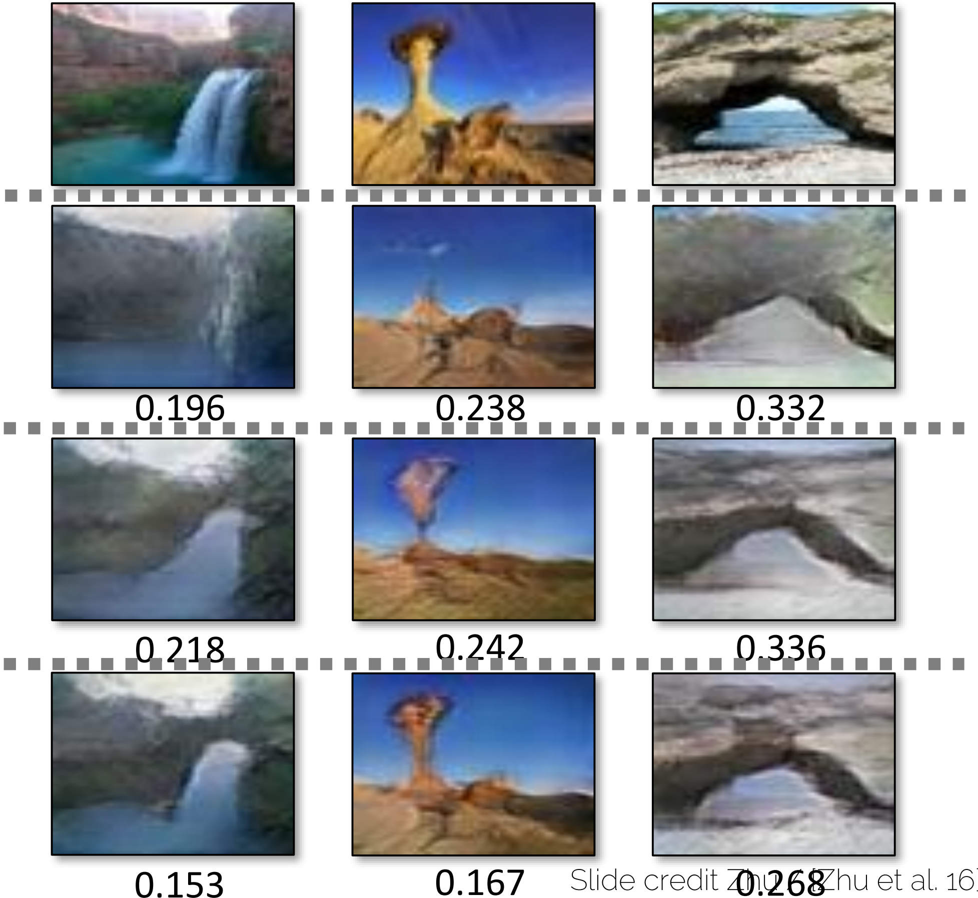
$$z^* = \arg\min \mathcal{L}(G(z), x^R)$$

Inverting Network $z = P(x)$

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

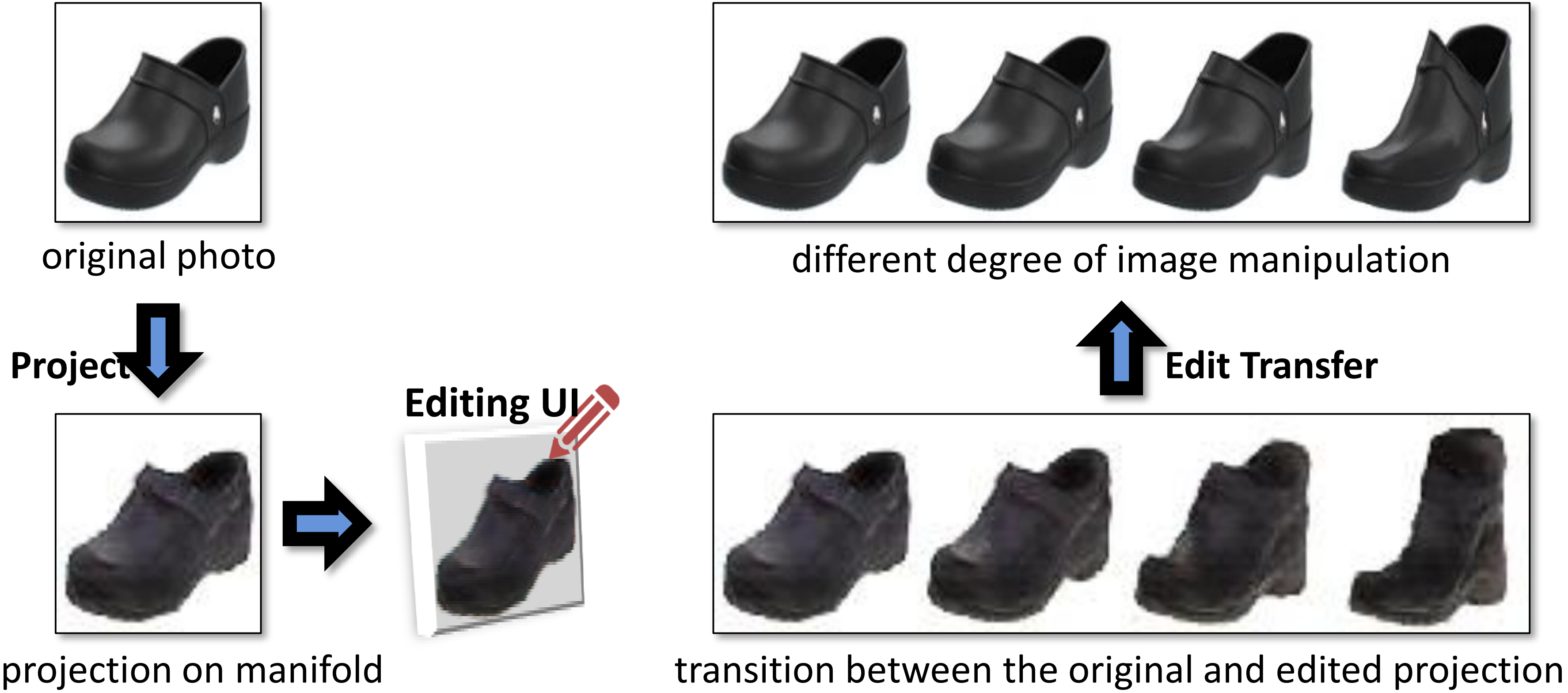
Hybrid Method

Use the network as initialization for the optimization problem



Slide credit Zhu [Zhu et al. 16]

iGANs: Overview



Slide credit Zhu / [Zhu et al. 16]

iGANs: Manipulating the Latent Vector

constraint violation loss L_g

user guidance image

Objective:

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

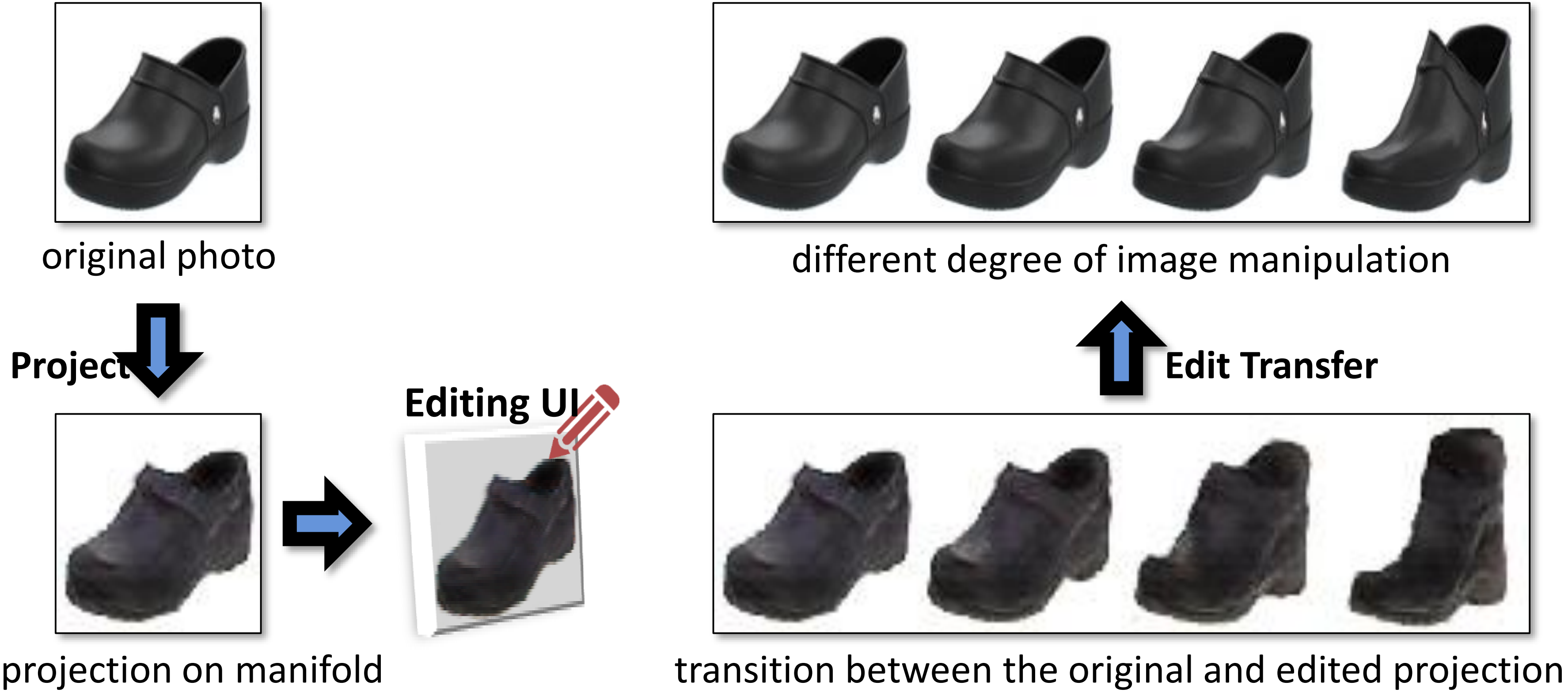
Guidance v_g

$G(z)$

z_0

Slide credit Zhu / [Zhu et al. 16]

iGANs: Overview

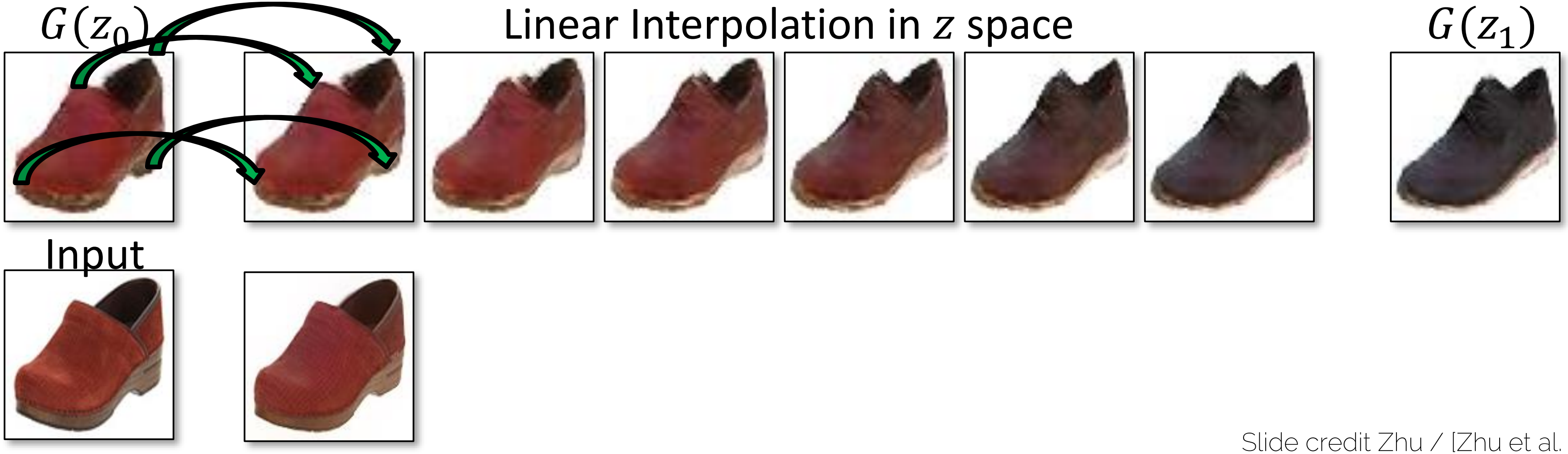


Slide credit Zhu / [Zhu et al. 16]

iGANs: Edit Transfer

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

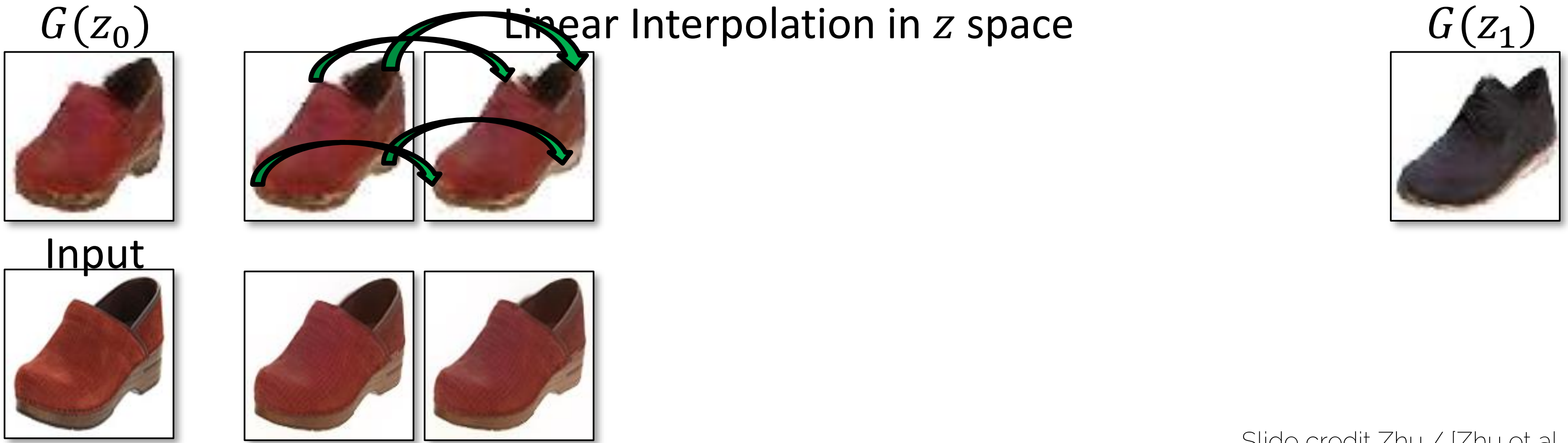
$$\iint \underbrace{\|I(x,y,t) - \boxed{A} \cdot I(x+\boxed{u}, y+\boxed{v}, t+1)\|^2}_{\text{data term}} + \underbrace{\sigma_s(\|\nabla u\|^2 + \|\nabla v\|^2)}_{\text{spatial reg}} + \underbrace{\sigma_c\|\nabla A\|^2}_{\text{color reg}} dxdy$$



iGANs: Edit Transfer

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

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

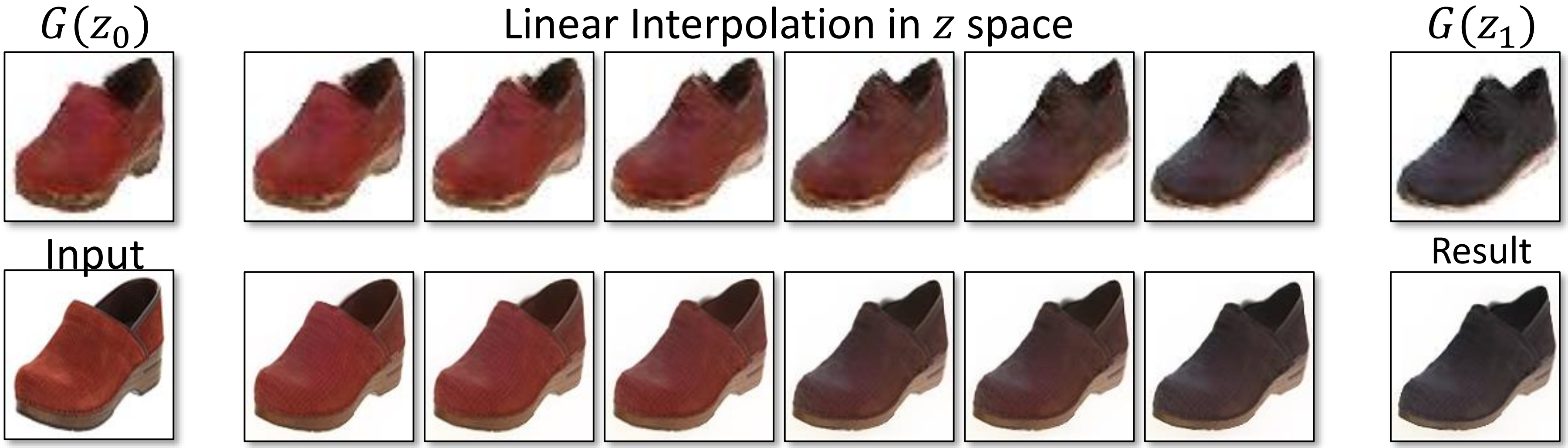


Slide credit Zhu / [Zhu et al. 16]

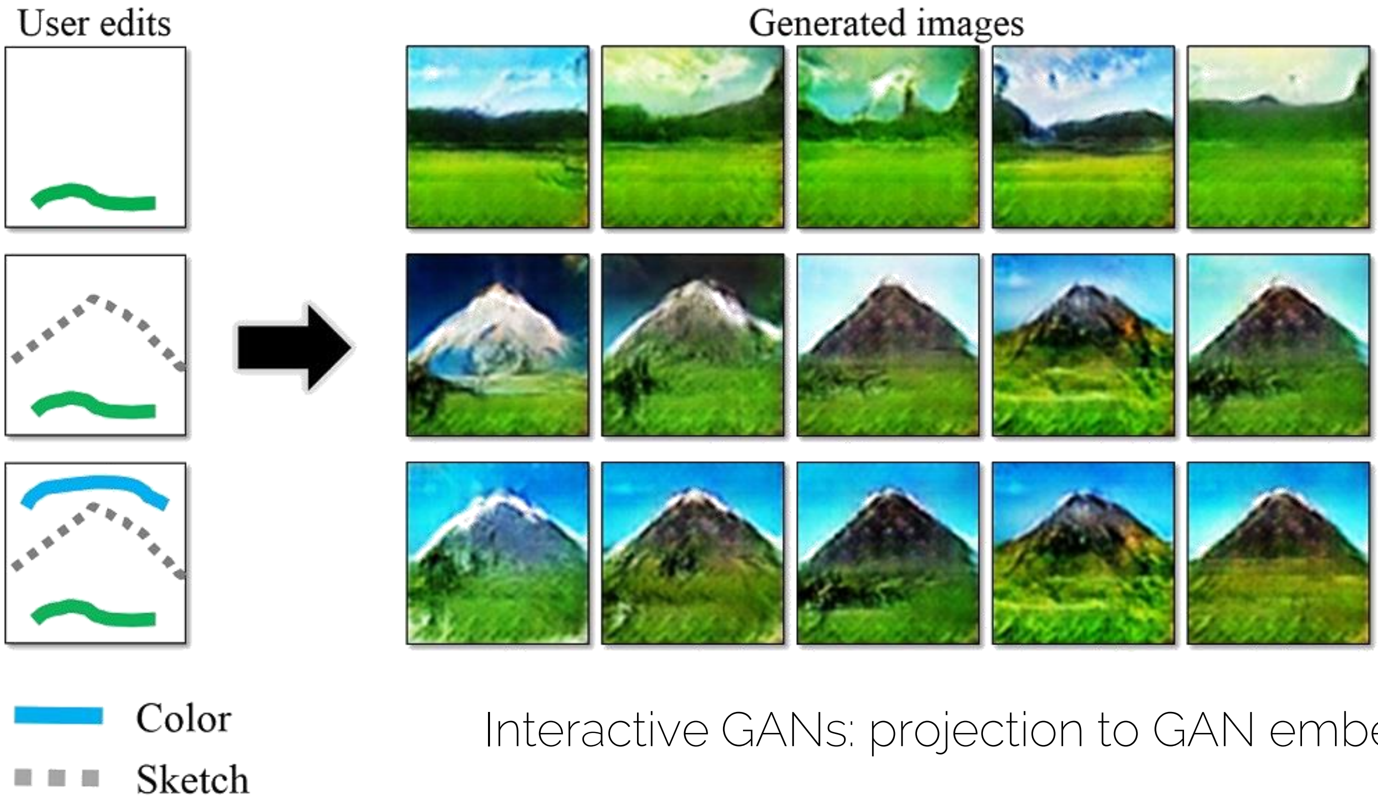
iGANs: Edit Transfer

Motion (u, v)+ Color ($A_{3\times4}$): 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



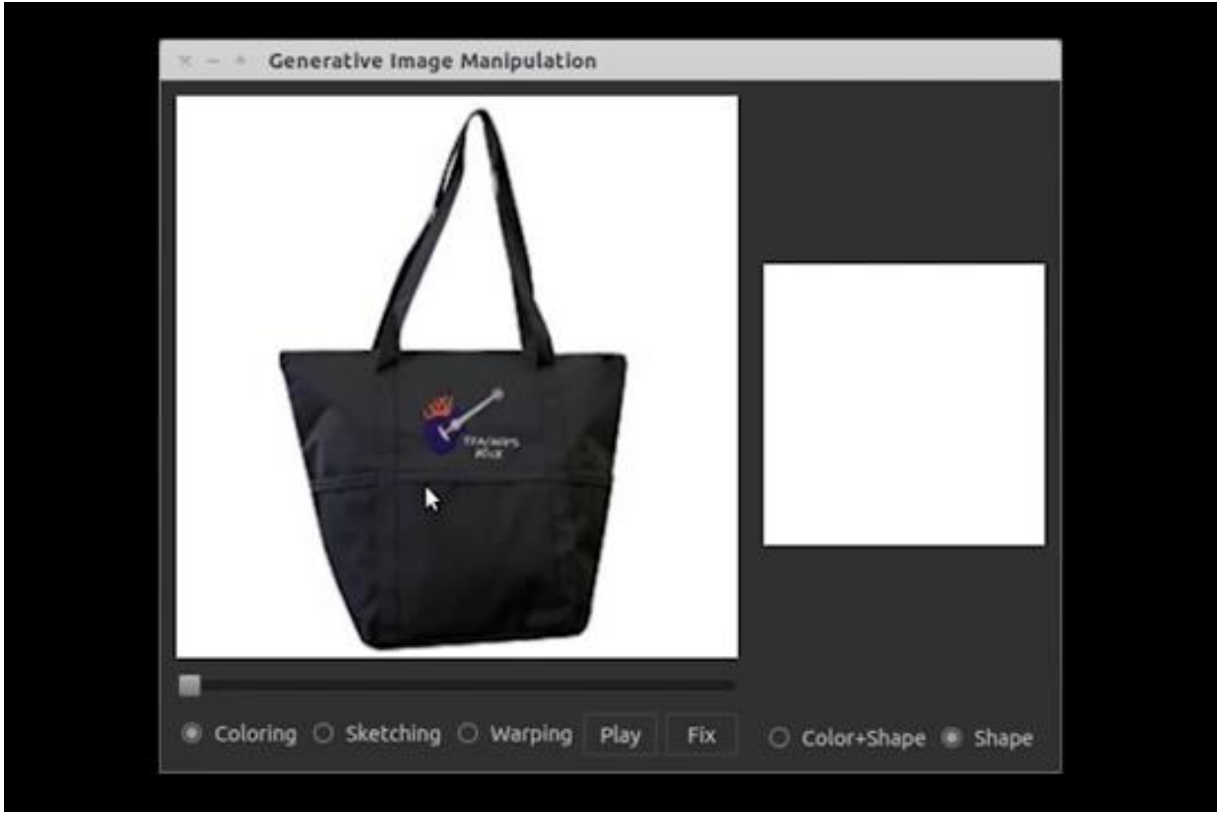
Interactive GANs: projection to GAN embedding

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.]

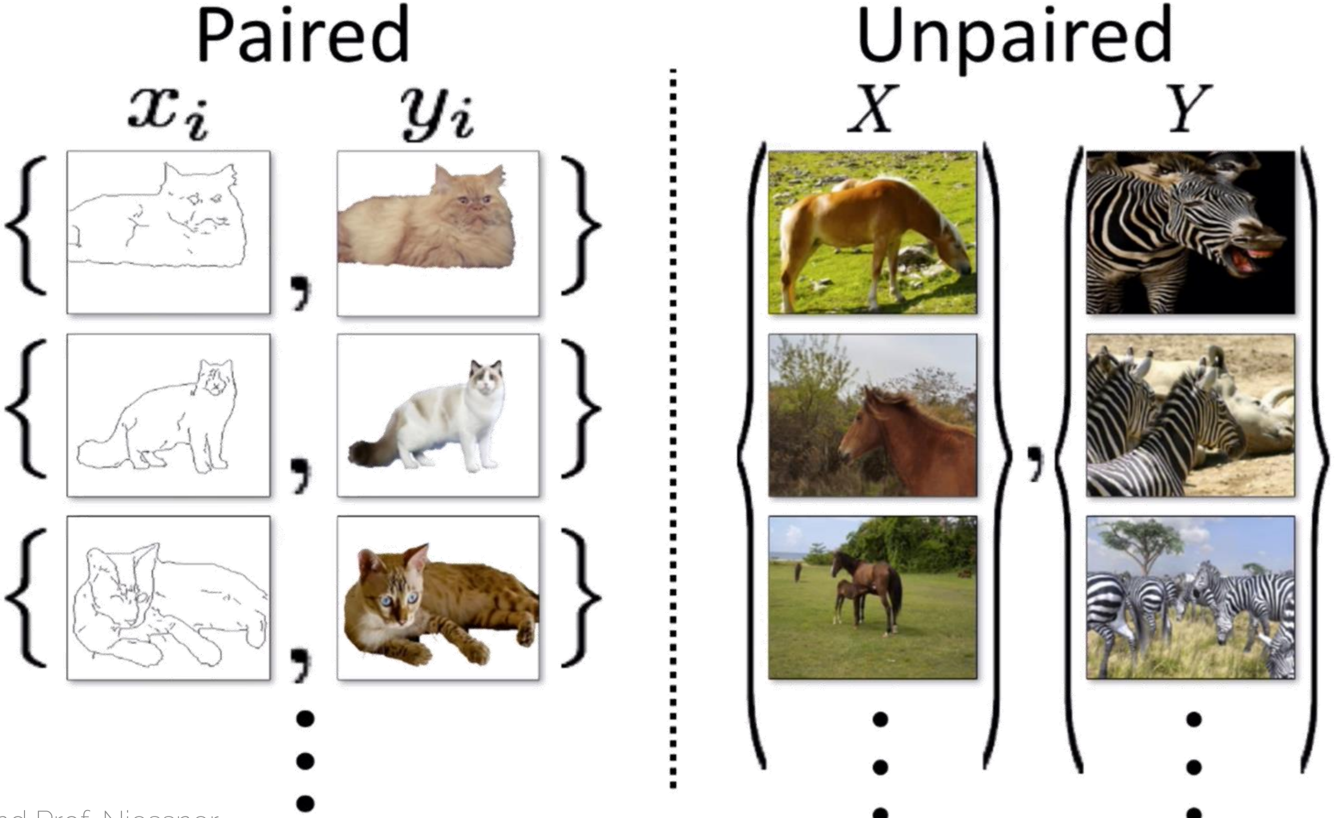
Mapping in Latent Space is Difficult!

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

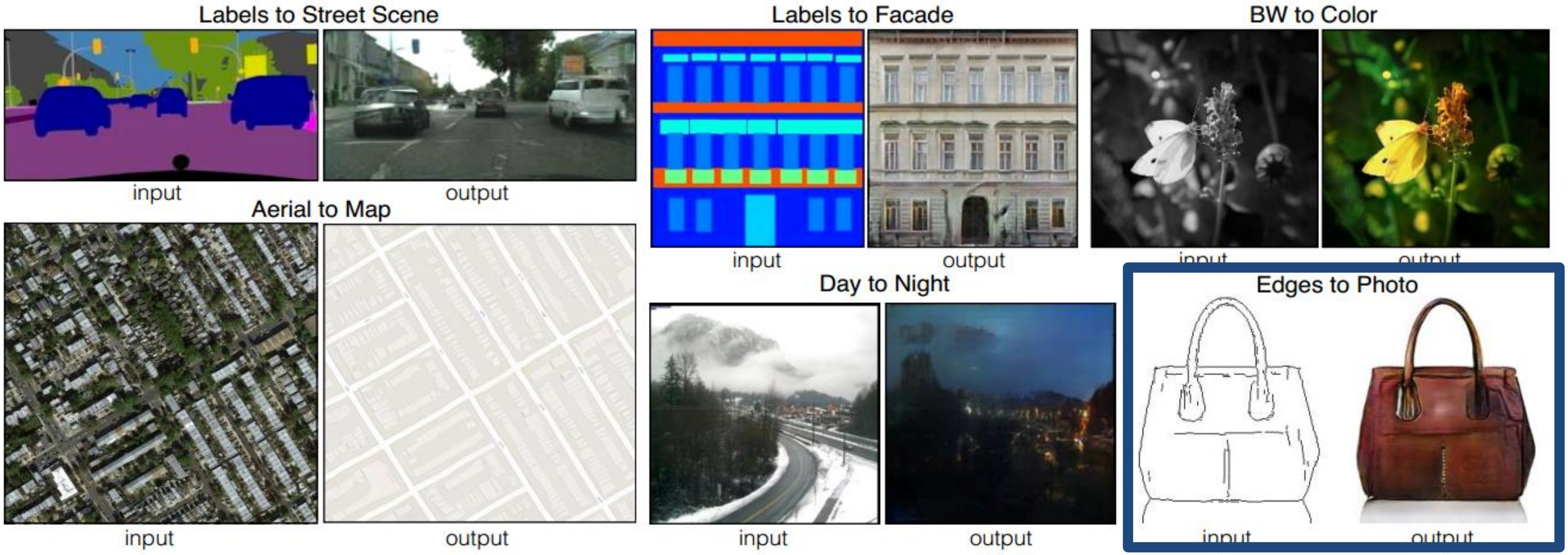
(a) Azimuth (pose) (b) Presence or absence of glasses

InfoGAN [Chen et al. 16]

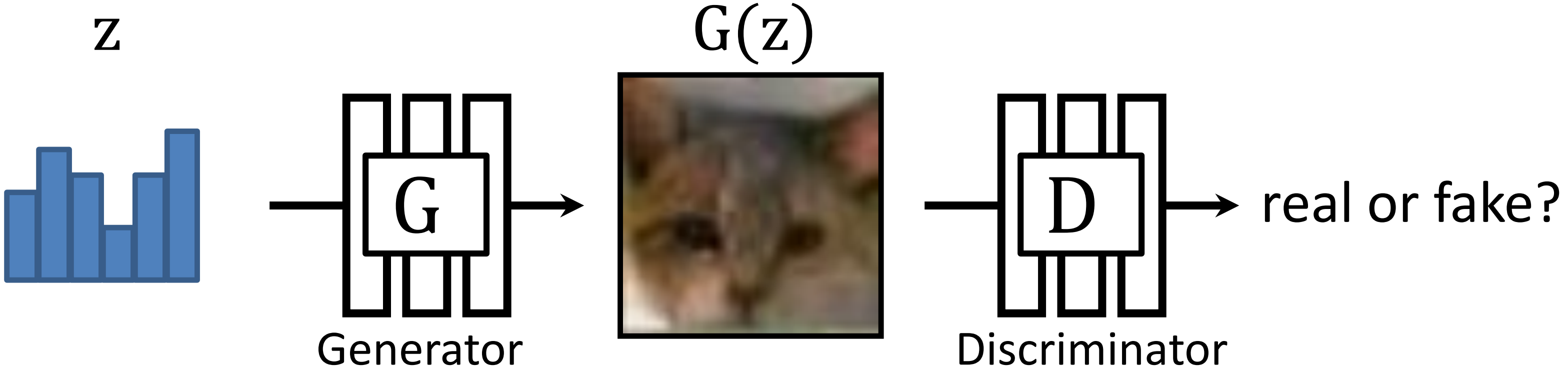
Paired vs Unpaired Setting



pix2pix: Image-to-Image Translation



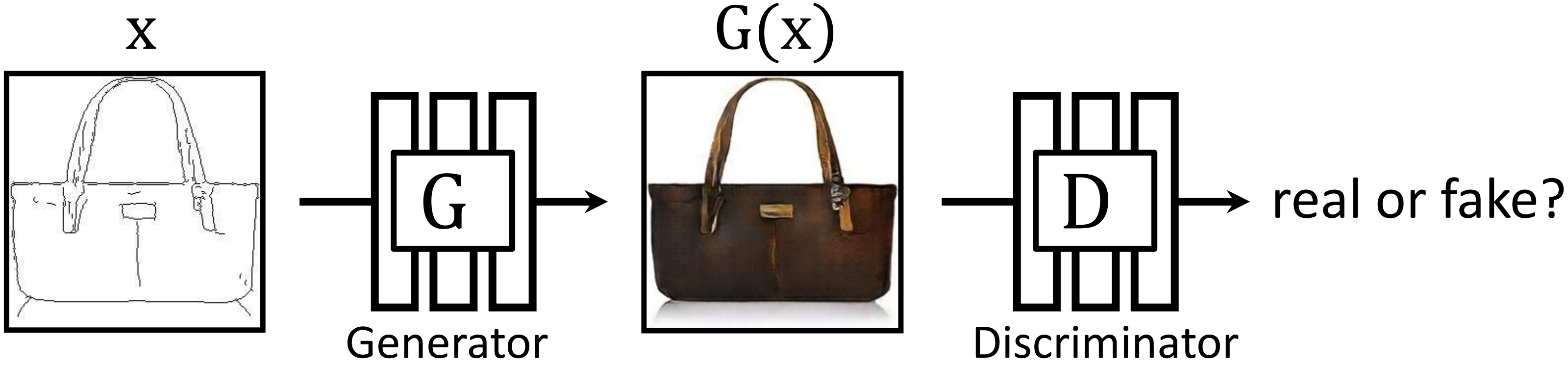
slides credit: Isola / Zhu



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

slides credit: Isola / Zhu

[Goodfellow et al. 2014]

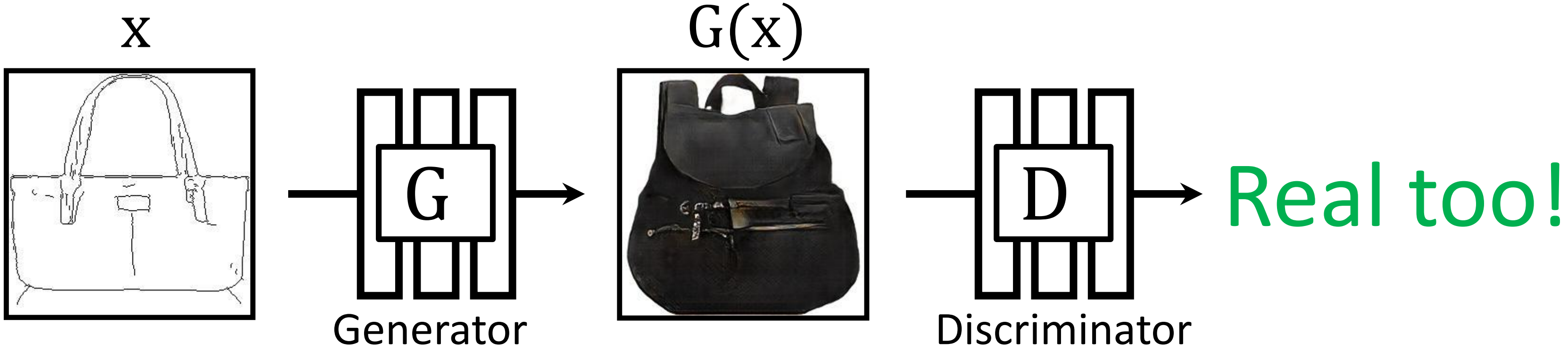


$$\min_{G} \max_{D} \mathbb{E}_{x,y}[\log D(G(x)) + \log(1 - D(y))]$$

slides credit: Isola / Zhu

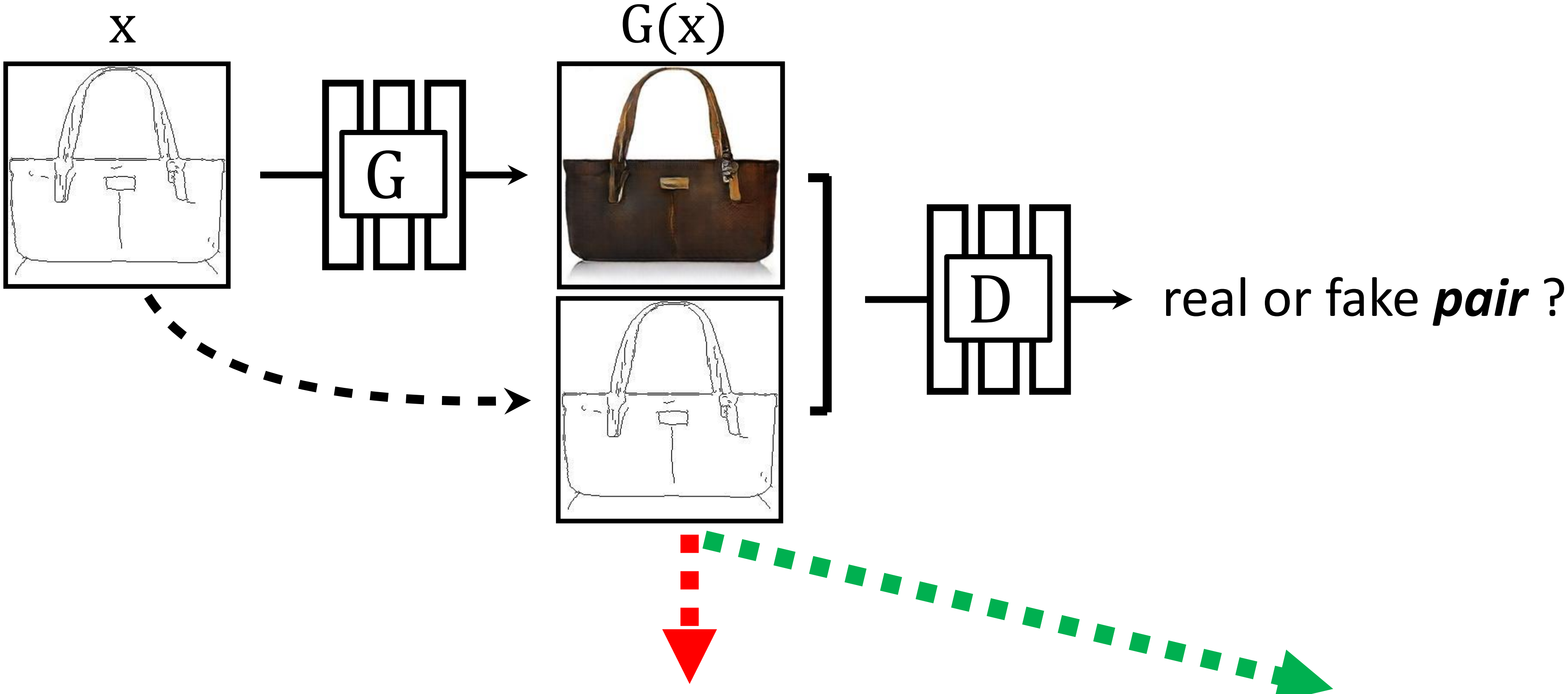
$$\min_G \max_D \mathbb{E}_{x,y}[\log D(G(x)) + \log(1 - D(y))]$$

slides credit: Isola / Zhu



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

slides credit: Isola / Zhu



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

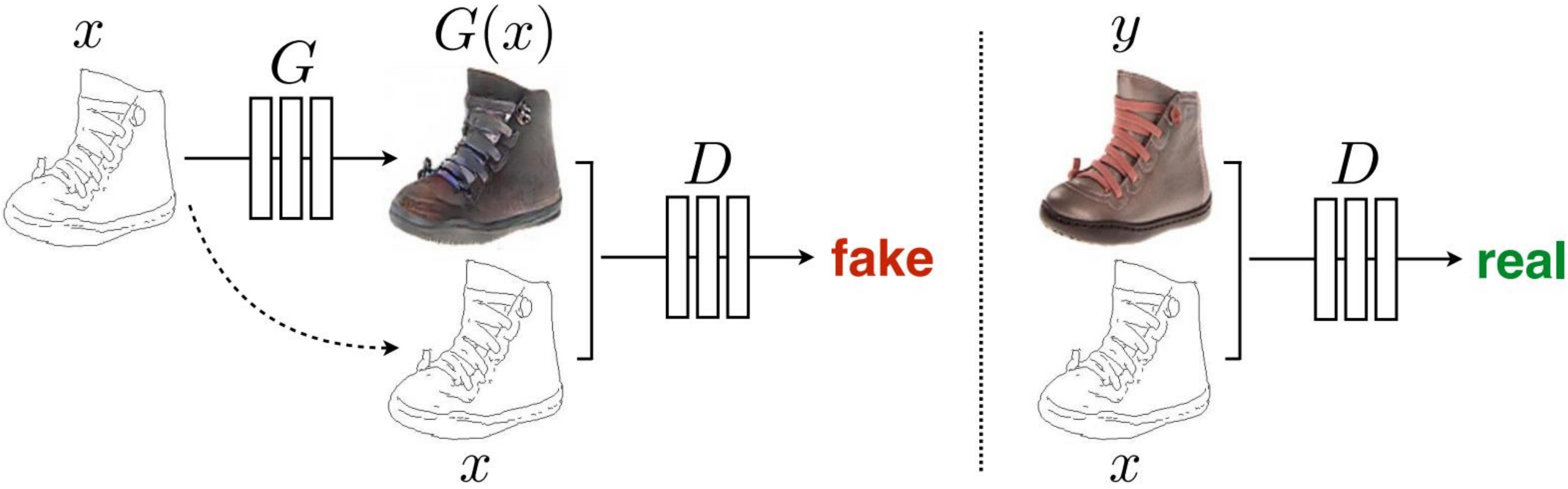
fake pair

real pair

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

slides credit: Isola / Zhu

pix2pix



Edges → Images

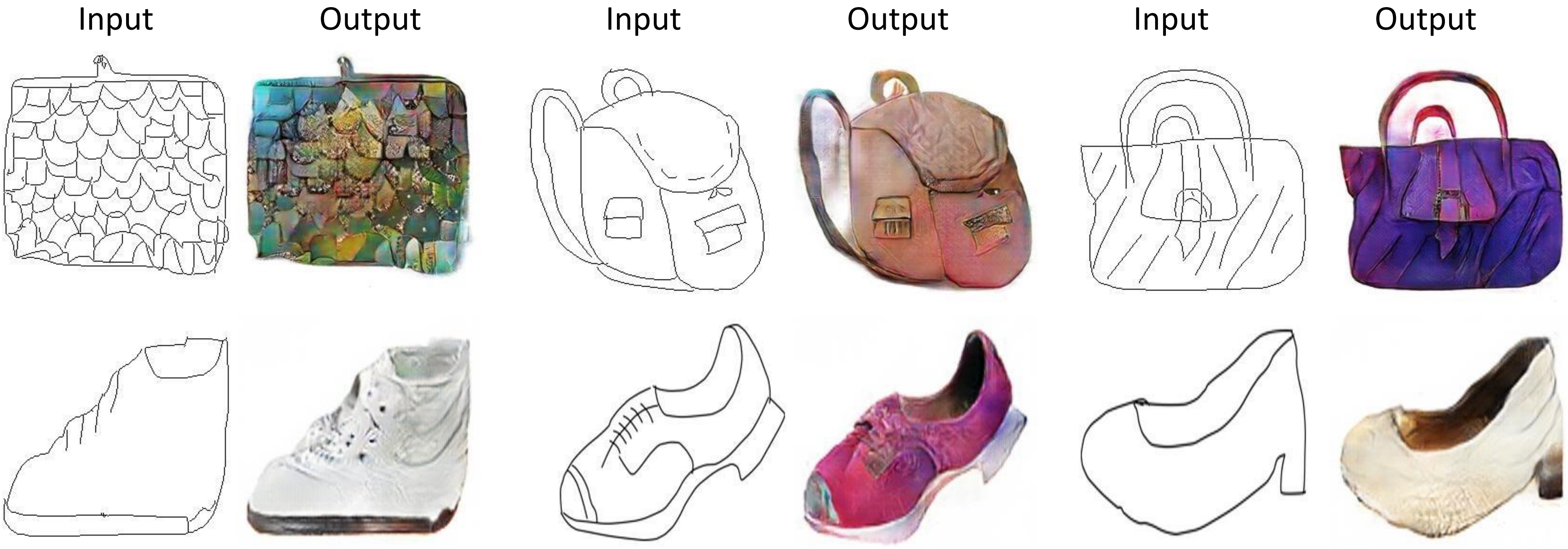
Edges from [Xie & Tu, 2015]

slides credit: Isola / Zhu

pix2pix: Paired Setting

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

Sketches → Images

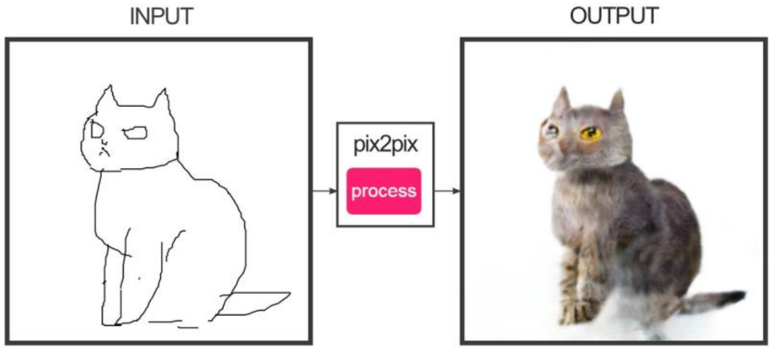


Trained on Edges → Images

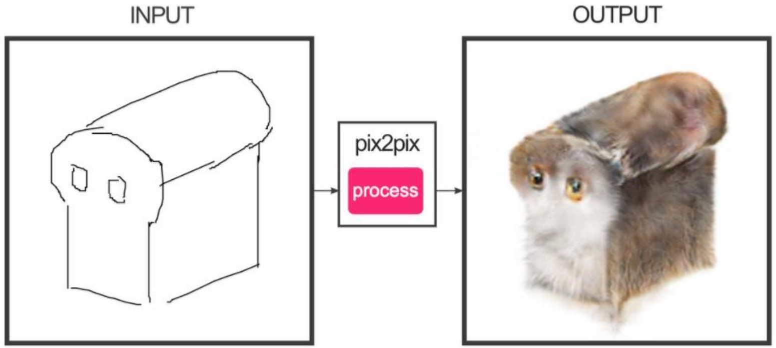
Data from [Eitz, Hays, Alexa, 2012]

slides credit: Isola / Zhu

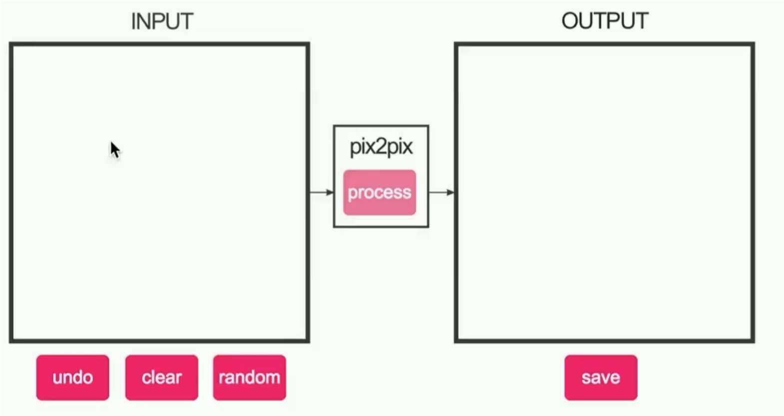
#edges2cats [Christopher Hesse]



@gods_tail



Ivy Tasi @ivymyt



@matthematician

Vitaly Vidmirov @vvid

https://affinelayer.com/pixsrv/

slides credit: Isola / Zhu

Input | Output | Groundtruth

Data from
[maps.google.com]

slides credit: Isola / Zhu

BW → Color

Input Output Input Output Input Output

Data from [Russakovsky et al. 2015]

slides credit: Isola / Zhu

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

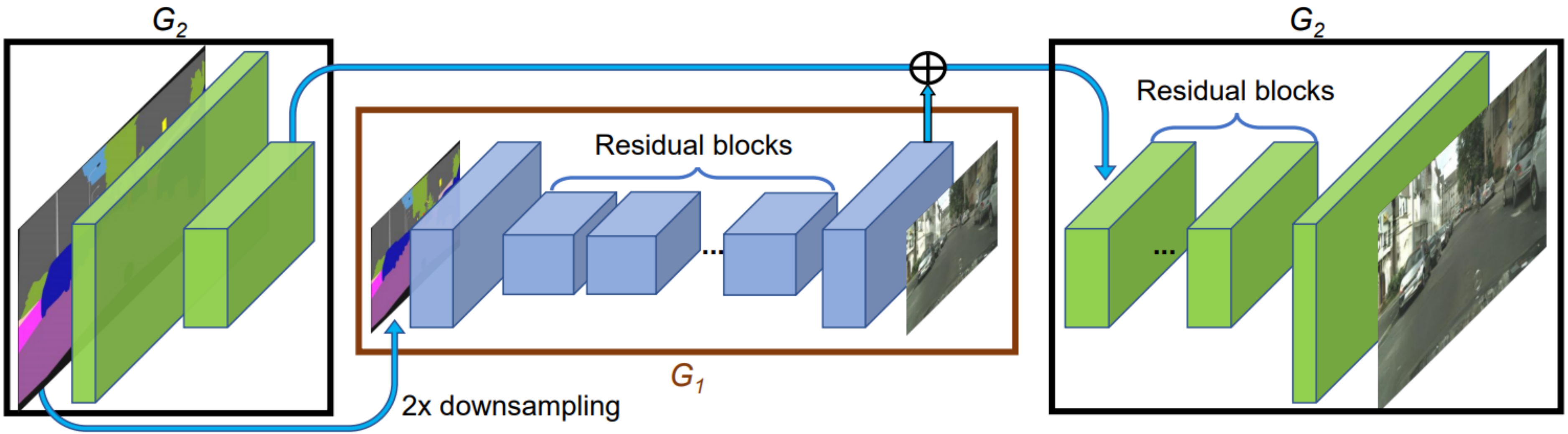
  – GAN discriminator applied only to local patches
  – It's fully-convolutional; i.e., can run on arbitrary image sizes

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



[Wang et al. 18]

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)

[Wang et al. 18]

Pix2PixHD

Pix2PixHD

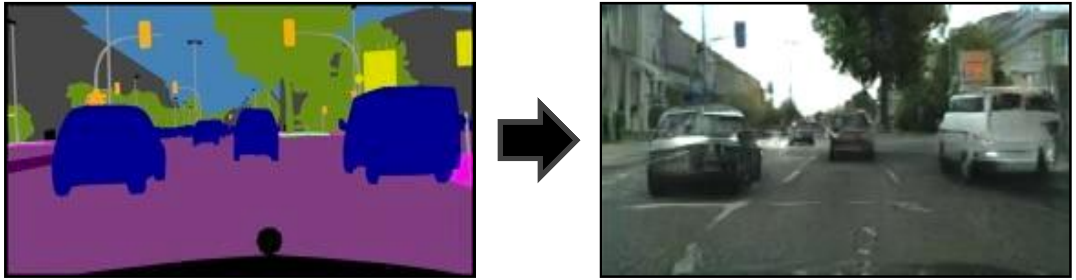
[Wang et al. 18]

Pix2PixHD (Interactive Results)

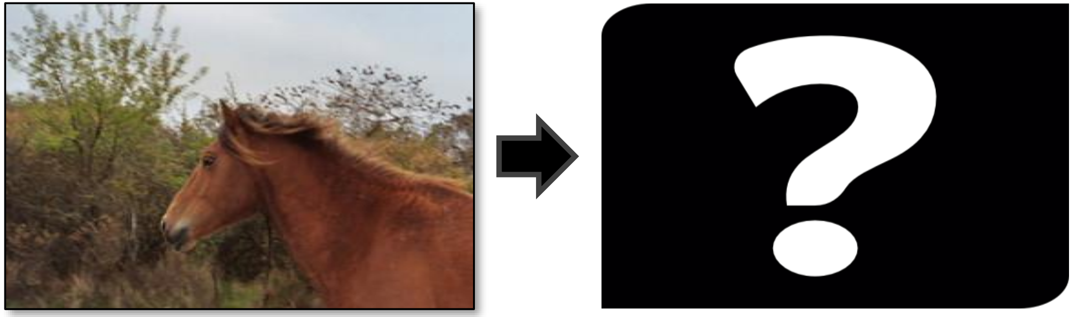
[Wang et al. 18]

Paired

x_i y_i



Label ↔ photo: per-pixel labeling

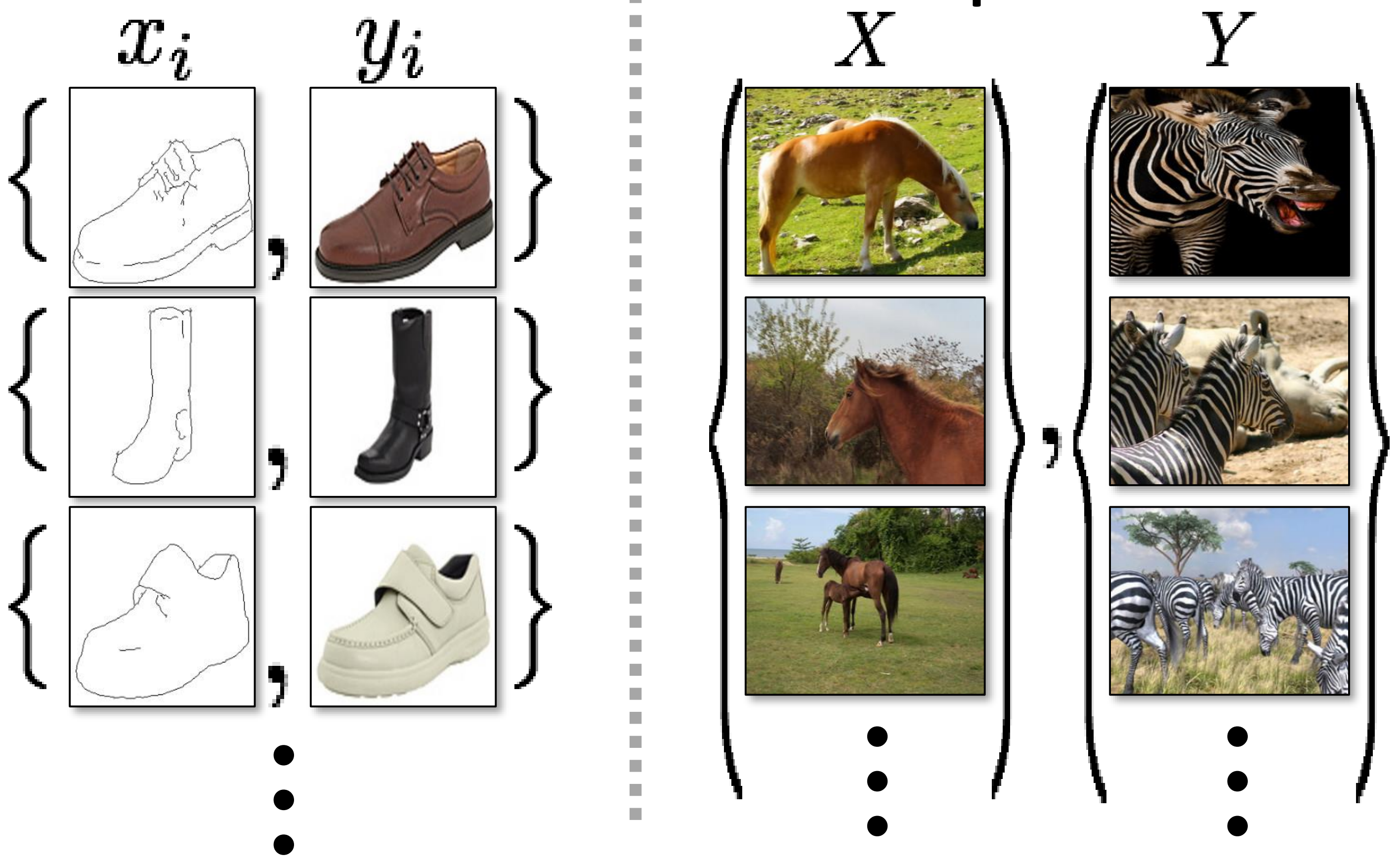


Horse ↔ zebra: how to get zebras?

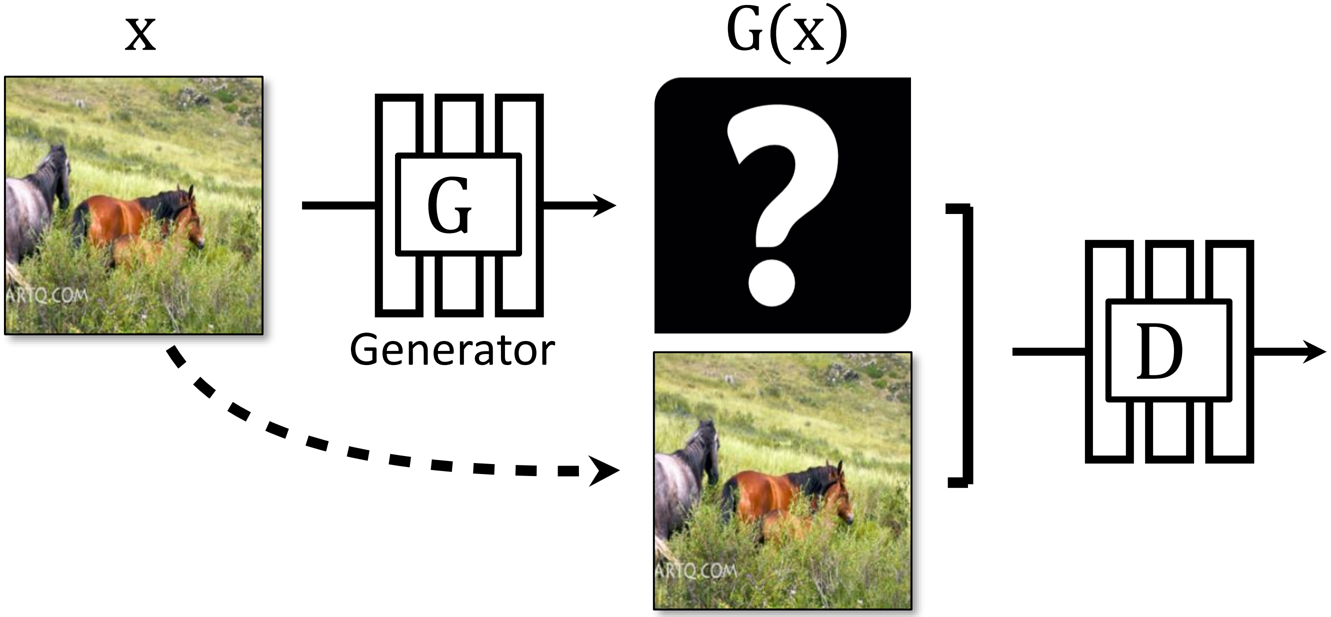
- Expensive to collect pairs.
- Impossible in many scenarios.

slides credit: Isola / Zhu

Paired | Unpaired

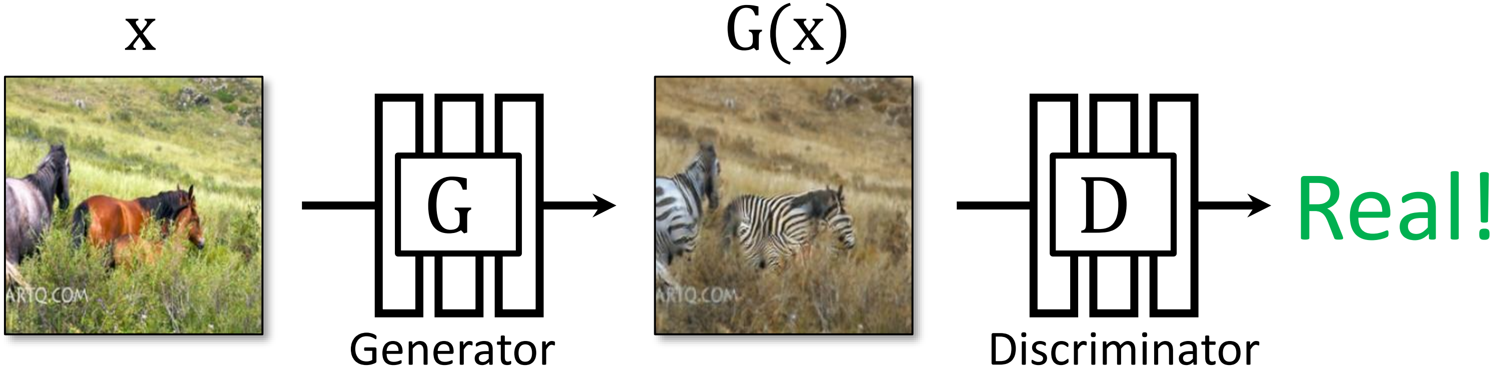


slides credit: Isola / Zhu

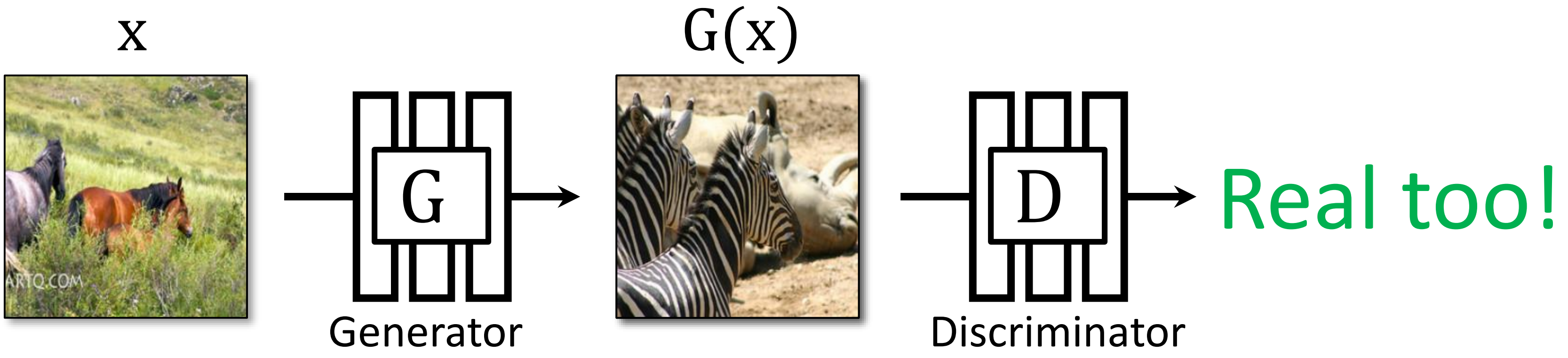


No input-output pairs!

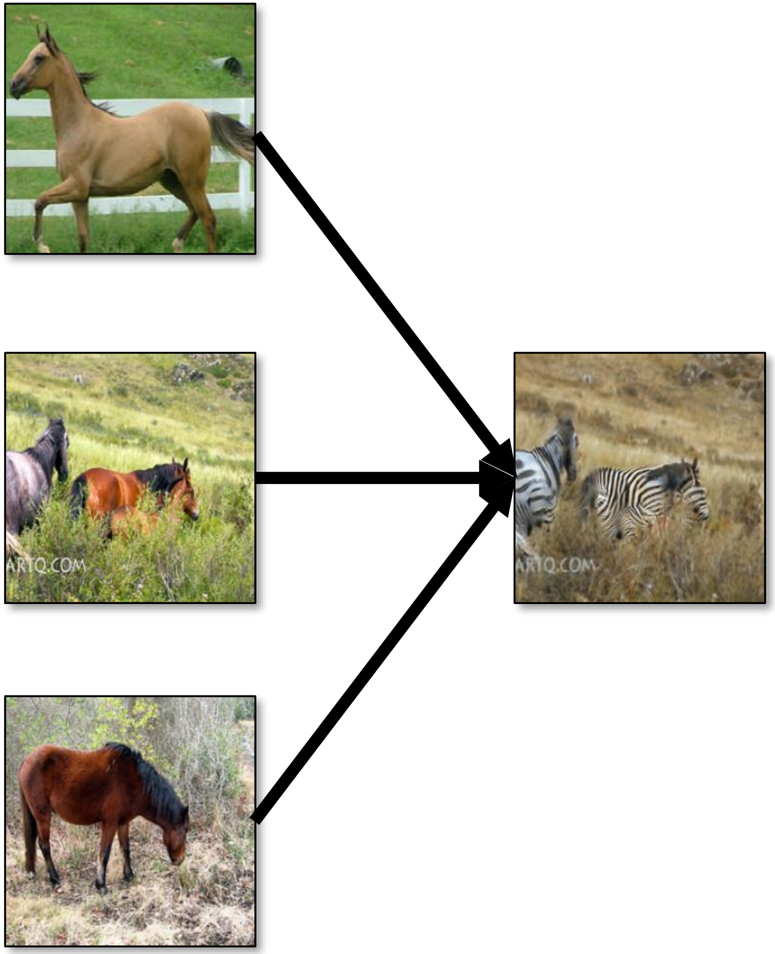
slides credit: Isola / Zhu



slides credit: Isola / Zhu



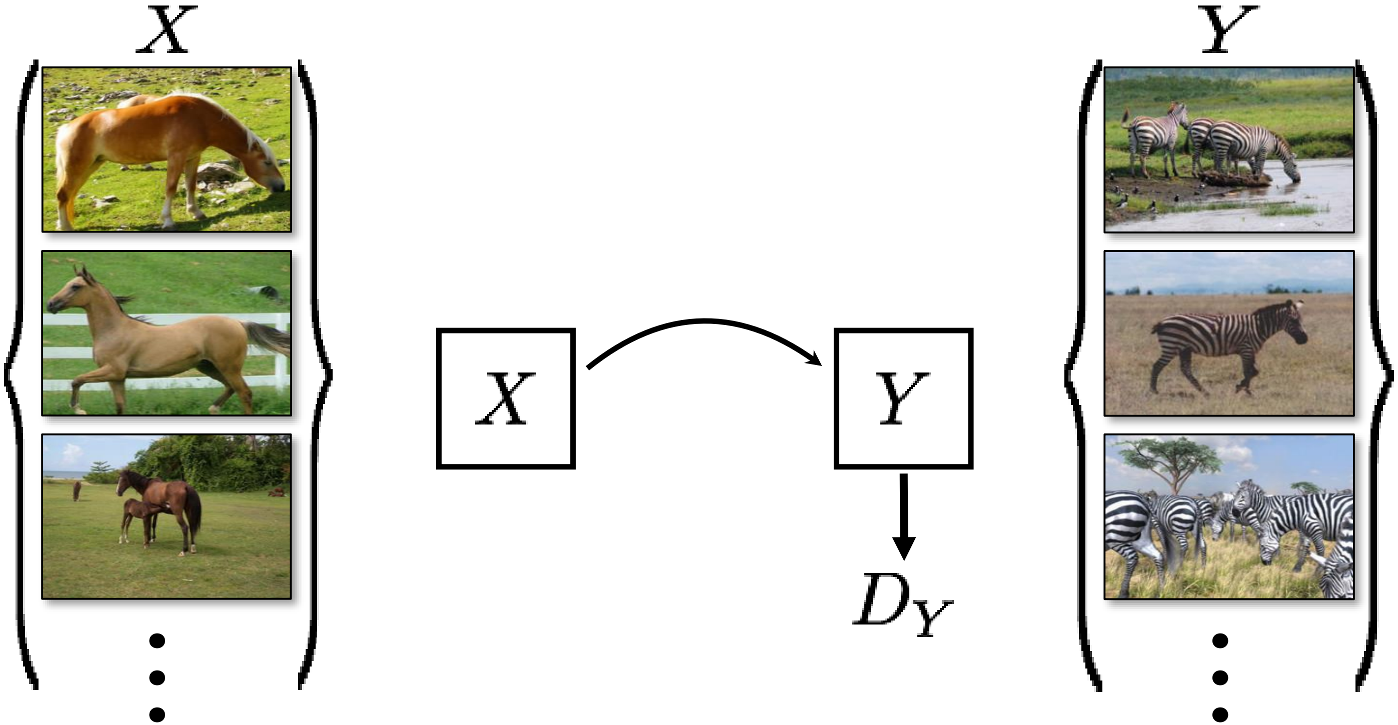
GANs doesn’t force output to correspond to input



mode collapse!

slides credit: Isola / Zhu

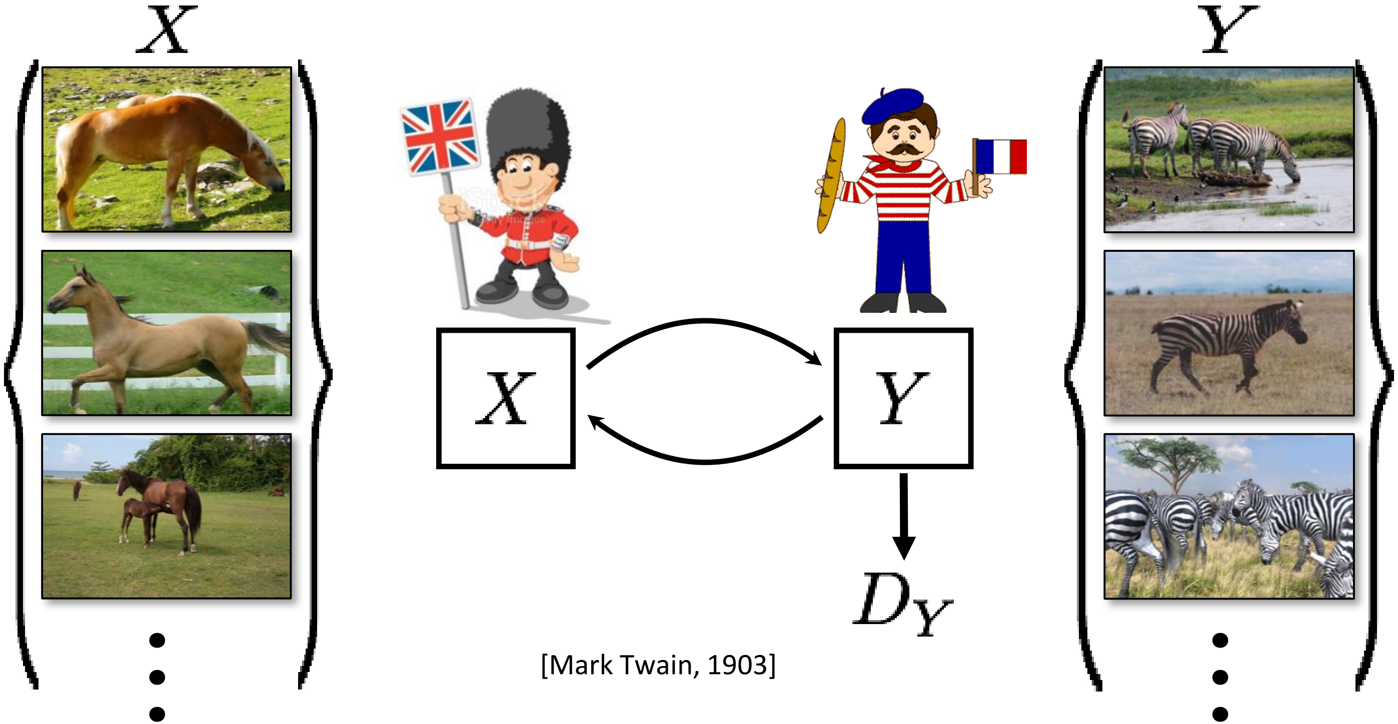
Cycle-Consistent Adversarial Networks



slides credit: Isola / Zhu

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

Cycle-Consistent Adversarial Networks

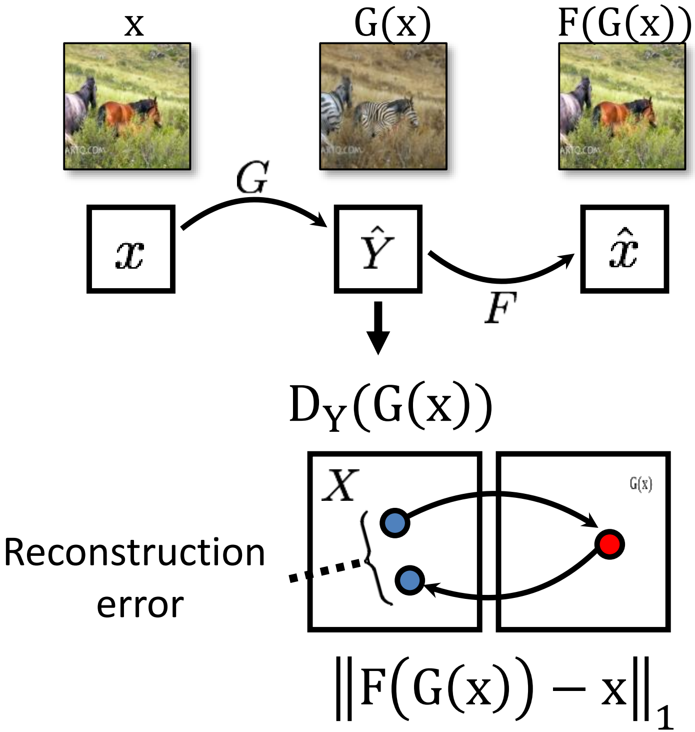


[Mark Twain, 1903]

slides credit: Isola / Zhu

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

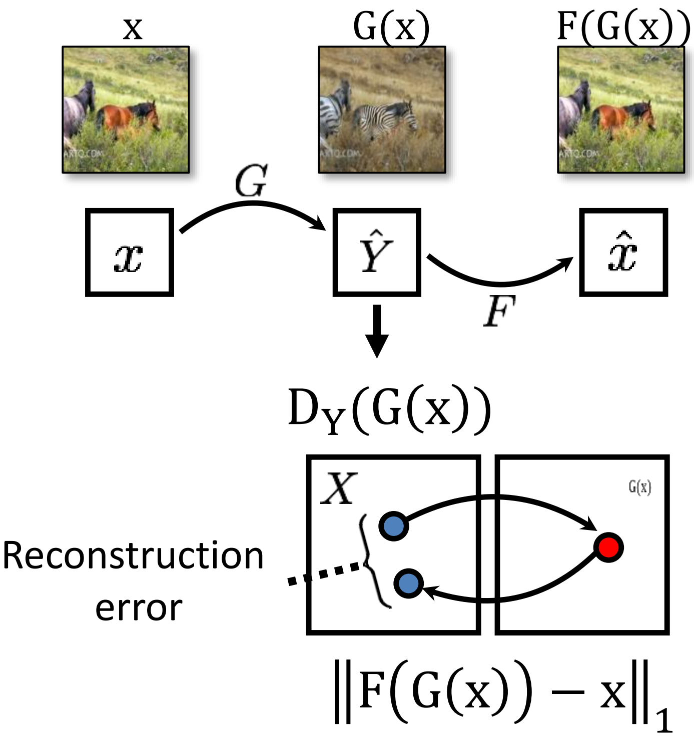
Cycle Consistency Loss

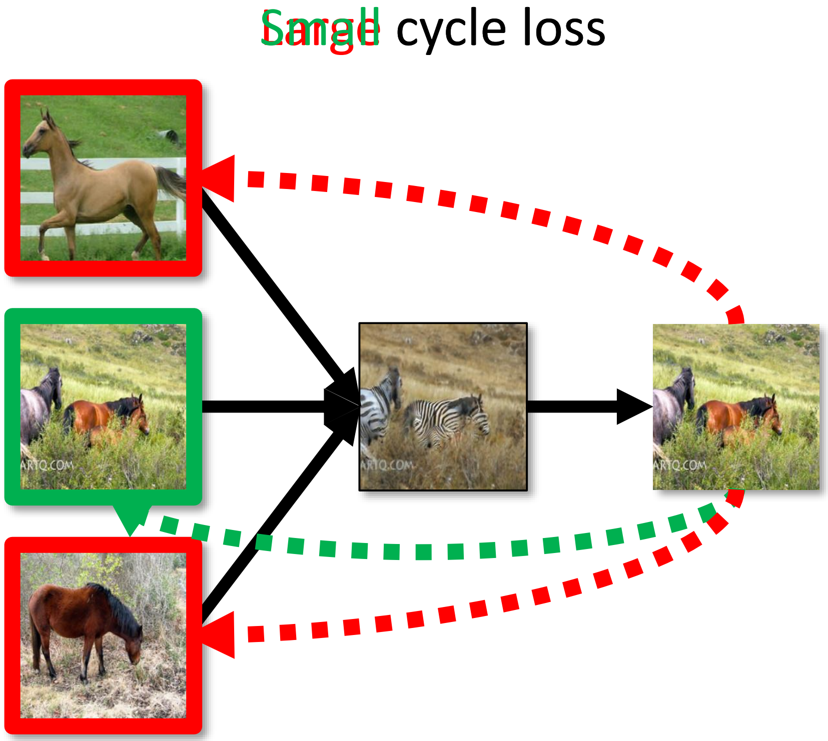


slides credit: Isola / Zhu

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

Cycle Consistency Loss

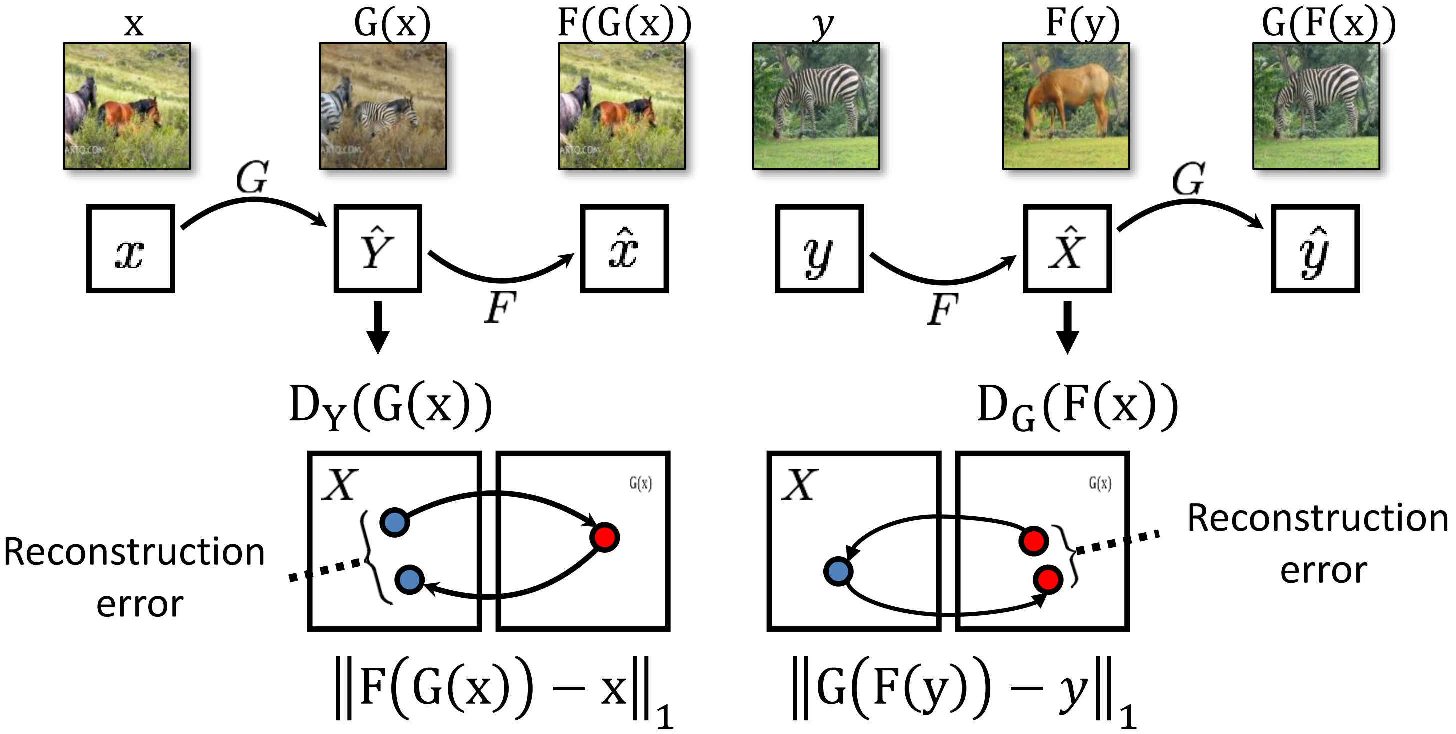




slides credit: Isola / Zhu

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

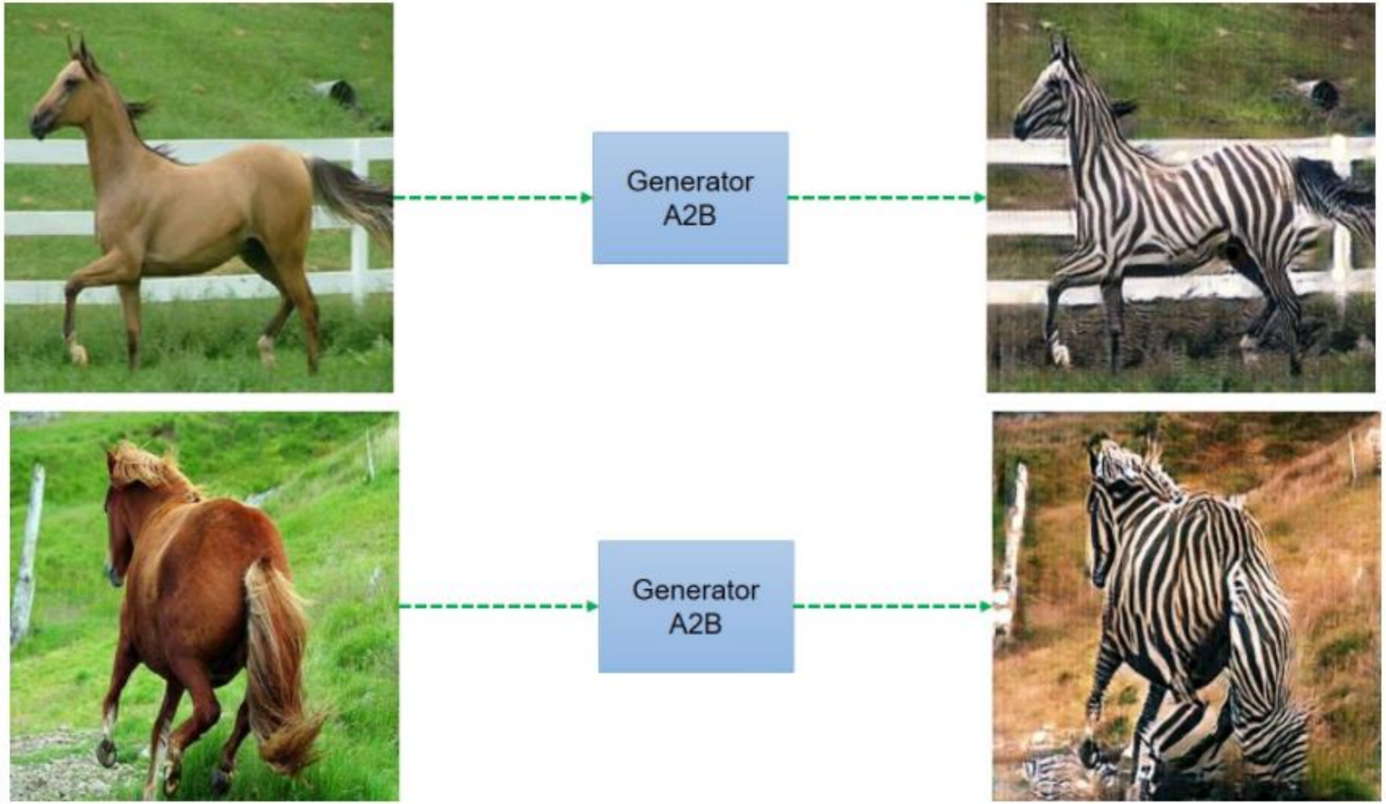
Cycle Consistency Loss



slides credit: Isola / Zhu

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

Cycle GAN - Overview



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

Monet's paintings → photos

slides credit: Isola / Zhu

slides credit: Isola / Zhu

slides credit: Isola / Zhu

Next Lectures

- Next Lectures:
  - Videos
  - Neural Rendering
  - 3D Deep Learning

- Keep working on the projects!

See you next week ☺