CS 2770: Computer Vision

Generative Adversarial Networks

Prof. Adriana Kovashka
University of Pittsburgh

April 14, 2020

Plan for this lecture

- Generative models: What are they?
- Technique: Generative Adversarial Networks
- Applications
- Conditional GANs
- Cycle-consistency loss
- Dealing with sparse data, progressive training

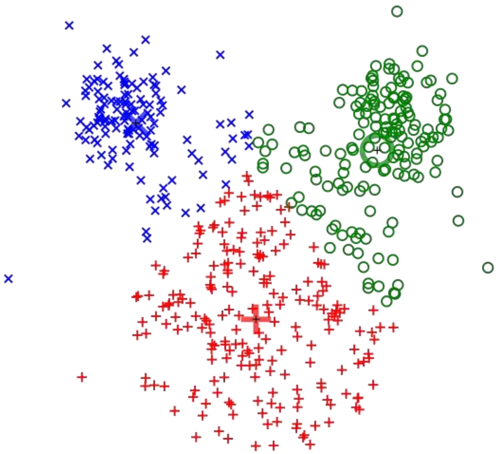
Supervised vs Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.



K-means clustering

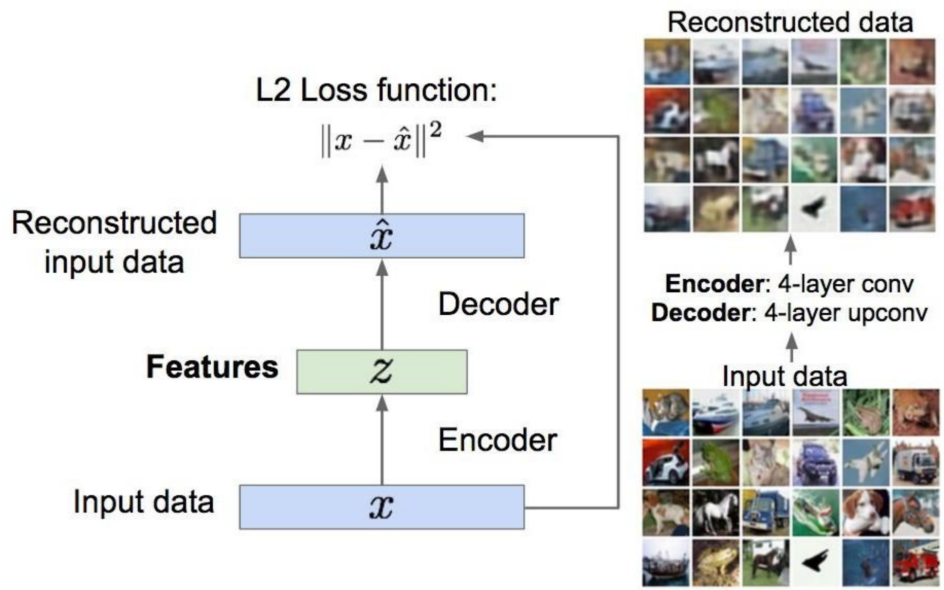
Supervised vs Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

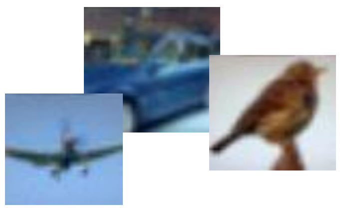
Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.



Autoencoders
(Feature learning)

Generative Models

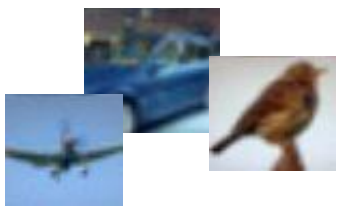


Training data ~ $p_{data}(x)$

Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Generative Models



Training data ~ $p_{data}(x)$

Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Addresses density estimation, a core problem in unsupervised learning

Several flavors:

- Explicit density estimation: explicitly define and solve for $p_{model}(x)$
- Implicit density estimation: learn model that can sample from $p_{model}(x)$ w/o explicitly defining it

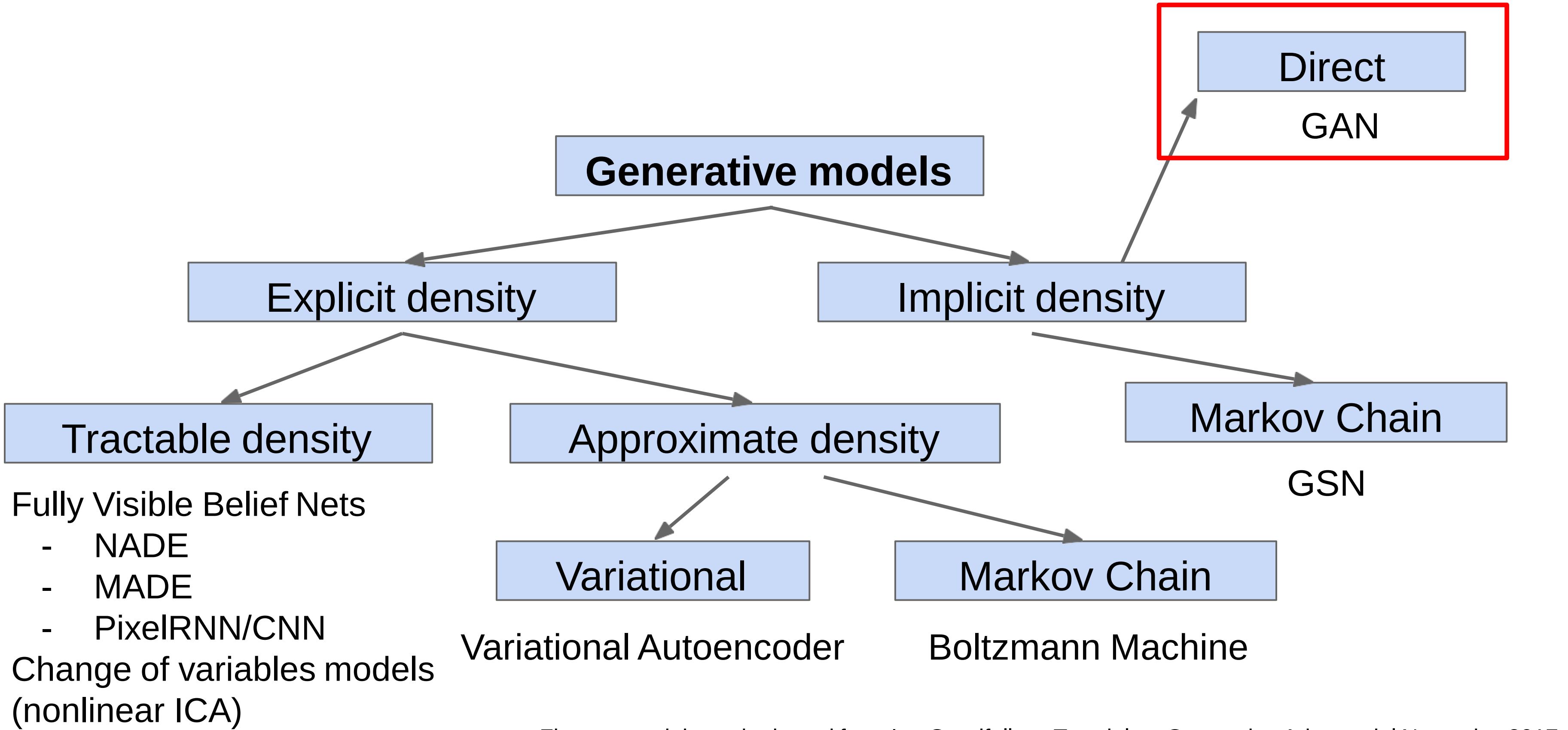
Why Generative Models?

- Realistic samples for artwork, super-resolution, colorization, etc.

- Generative models can be used to enhance training datasets with diverse synthetic data
- Generative models of time-series data can be used for simulation

Adapted from Serena Young

Taxonomy of Generative Models



Generative Adversarial Networks

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.

Q: What can we use to represent this complex transformation?

Generative Adversarial Networks

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

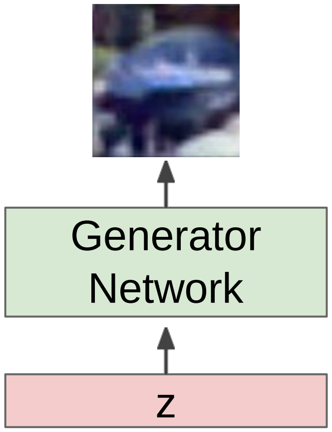
Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.

Q: What can we use to represent this complex transformation?

A: A neural network!

Output: Sample from training distribution



Input: Random noise

Training GANs: Two-player game

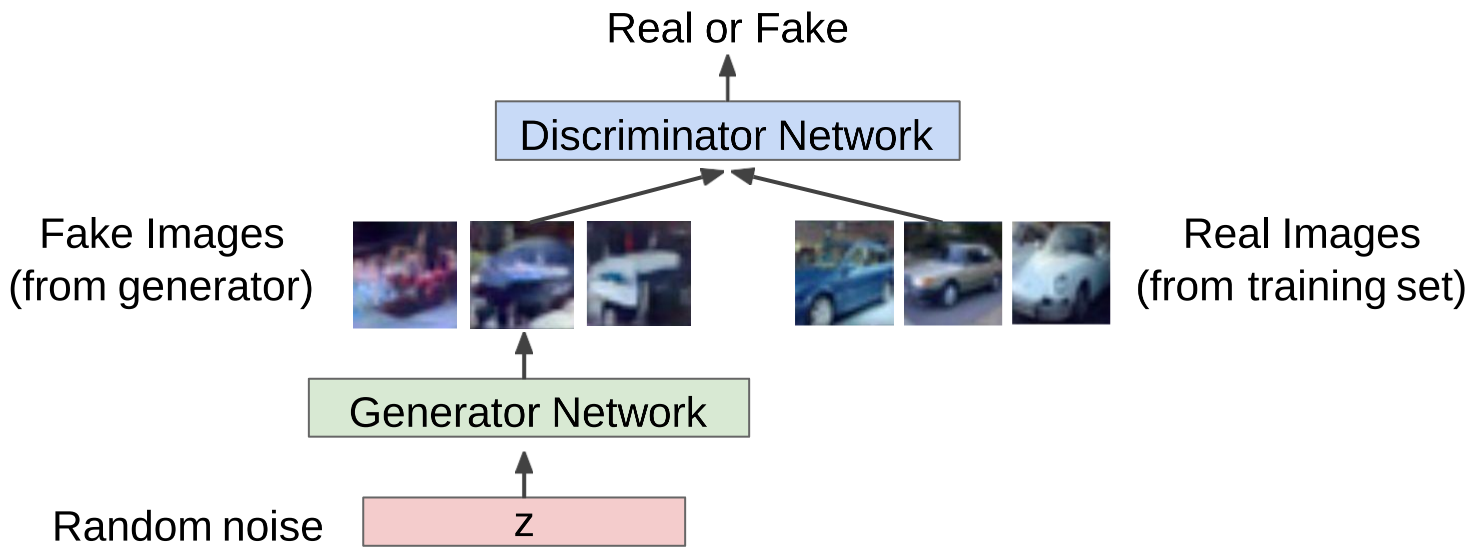
Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

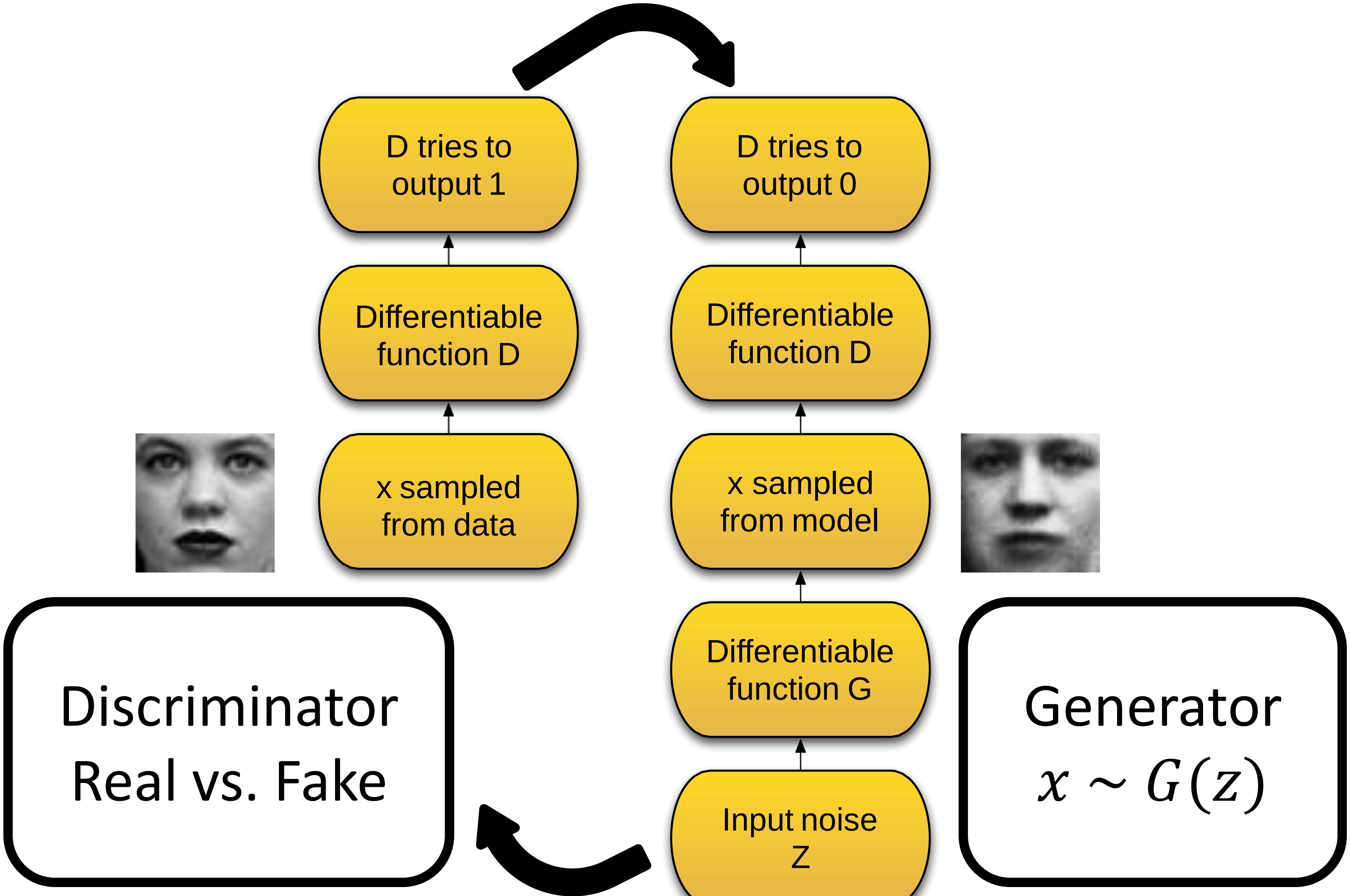
Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images



Adversarial Networks Framework



Ian Goodfellow

Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

Train jointly in **minimax game**

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

Train jointly in **minimax game**

Discriminator outputs likelihood in (0,1) of real image

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x\sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z\sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

Serena Young

Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

Train jointly in **minimax game**

Discriminator outputs likelihood in (0,1) of real image

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

- Discriminator (θ_d) wants to **maximize objective** such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)
- Generator (θ_g) wants to **minimize objective** such that D(G(z)) is close to 1 (discriminator is fooled into thinking generated G(z) is real)

Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Alternate between:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

2. **Gradient descent** on generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Alternate between:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

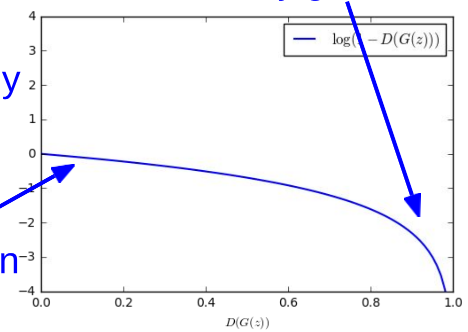
2. **Gradient descent** on generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

In practice, optimizing this generator objective does not work well!

When sample is likely fake, want to learn from it to improve generator. But gradient in this region is relatively flat!

Gradient signal dominated by region where sample is already good



Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Alternate between:

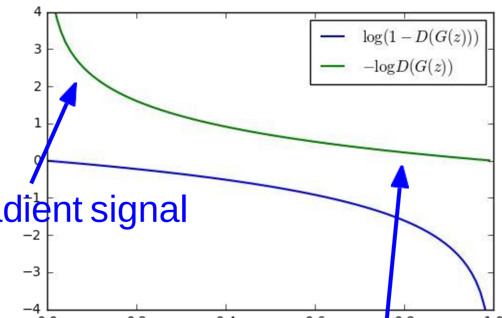
1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

2. **Instead: Gradient ascent** on generator, different objective

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

Instead of minimizing likelihood of discriminator being correct, now maximize likelihood of discriminator being wrong.
Same objective of fooling discriminator, but now higher gradient signal for bad samples => works much better! Standard in practice.



Serena Young

Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Putting it together: GAN training algorithm

for number of training iterations **do**
  for k steps **do**
   • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
   • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
   • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

  end for
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by ascending its stochastic gradient (improved objective):

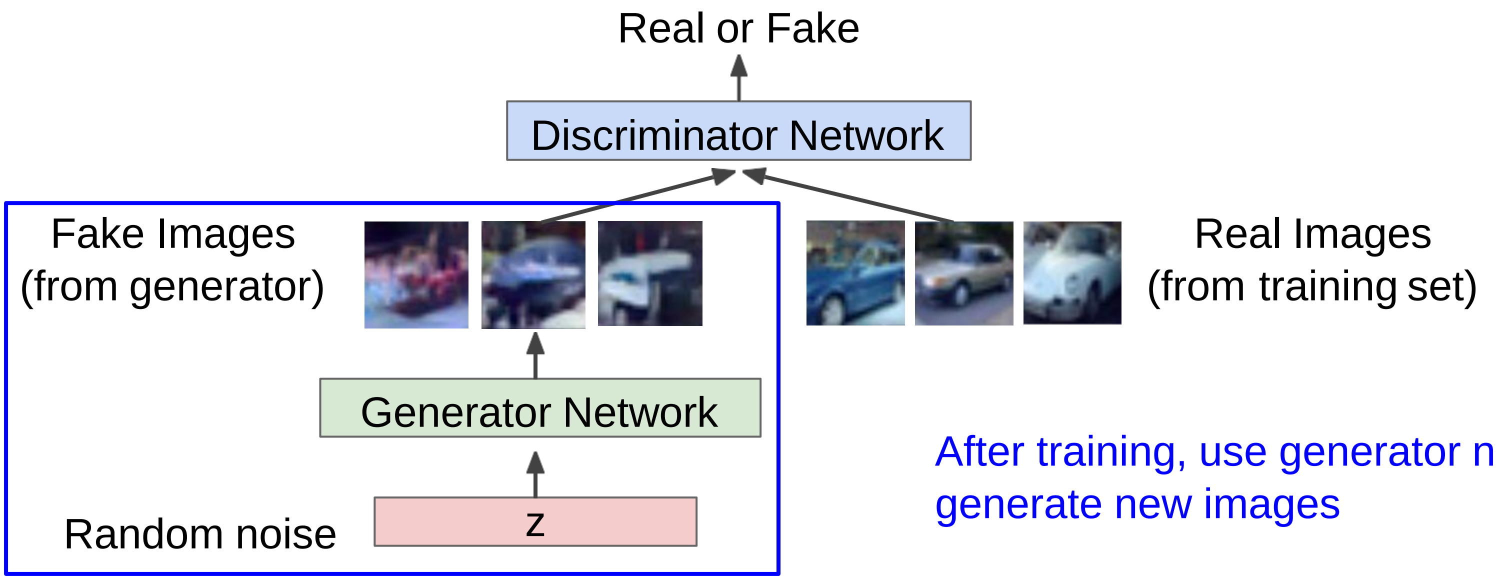
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images



After training, use generator network to generate new images

GAN training is challenging

- Vanishing gradient – when discriminator is very good
- Mode collapse – too little diversity in the samples generated
- Lack of convergence because hard to reach Nash equilibrium
- Loss metric doesn’t always correspond to image quality; Frechet Inception Distance (FID) is a decent choice

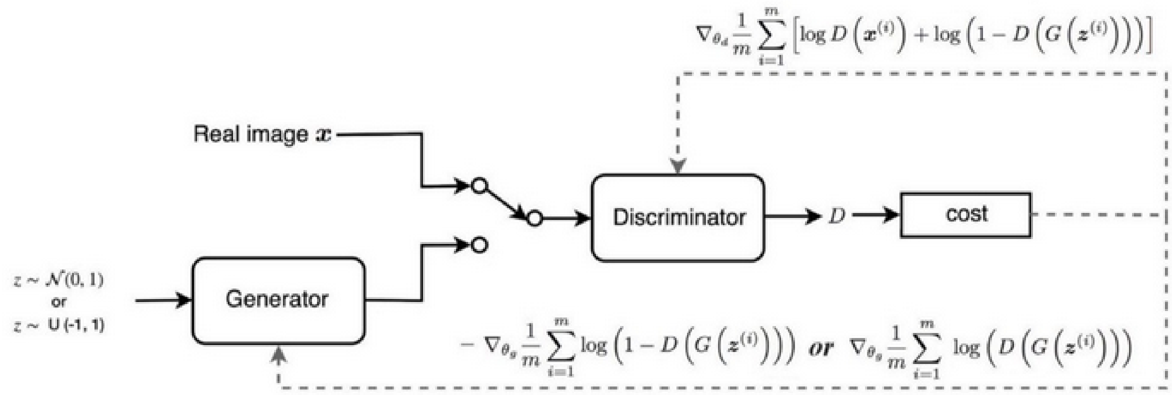
Alternative loss functions

Name	Paper Link	Value Function
GAN	Arxiv	$L_D^{GAN} = E\big[\log\big(D(x)\big)\big] + E\big[\log\big(1 - D(G(z))\big)\big]$ $L_G^{GAN} = E\big[\log\big(D(G(z))\big)\big]$
LSGAN	Arxiv	$L_D^{LSGAN} = E[(D(x) - 1)^2] + E[D(G(z))^2]$ $L_G^{LSGAN} = E[(D(G(z)) - 1)^2]$
WGAN	Arxiv	$L_D^{WGAN} = E[D(x)] - E[D(G(z))]$ $L_G^{WGAN} = E[D(G(z))]$ $W_D \leftarrow clip\_by\_value(W_D, -0.01, 0.01)$
WGAN_GP	Arxiv	$L_D^{WGAN\_GP} = L_D^{WGAN} + \lambda E[(\lvert\nabla D(\alpha x - (1 - \alpha G(z)))\rvert - 1)^2]$ $L_G^{WGAN\_GP} = L_G^{WGAN}$
DRAGAN	Arxiv	$L_D^{DRAGAN} = L_D^{GAN} + \lambda E\big[\big(\lvert\nabla D(\alpha x - (1 - \alpha x_p))\rvert - 1\big)^2\big]$ $L_G^{DRAGAN} = L_G^{GAN}$
CGAN	Arxiv	$L_D^{CGAN} = E\big[\log\big(D(x, c)\big)\big] + E\big[\log\big(1 - D(G(z), c)\big)\big]$ $L_G^{CGAN} = E\big[\log\big(D(G(z), c)\big)\big]$
infoGAN	Arxiv	$L_{D,Q}^{infoGAN} = L_D^{GAN} - \lambda L_I(c, c')$ $L_G^{infoGAN} = L_G^{GAN} - \lambda L_I(c, c')$
ACGAN	Arxiv	$L_{D,Q}^{ACGAN} = L_D^{GAN} + E[P(class = c \mid x)] + E[P(class = c \mid G(z))]$ $L_G^{ACGAN} = L_G^{GAN} + E[P(class = c \mid G(z))]$
EBGAN	Arxiv	$L_D^{EBGAN} = D_{AE}(x) + \max(0, m - D_{AE}(G(z)))$ $L_G^{EBGAN} = D_{AE}(G(z)) + \lambda \cdot PT$
BEGAN	Arxiv	$L_D^{BEGAN} = D_{AE}(x) - k_t D_{AE}(G(z))$ $L_G^{BEGAN} = D_{AE}(G(z))$ $k_{t+1} = k_t + \lambda(\gamma D_{AE}(x) - D_{AE}(G(z)))$

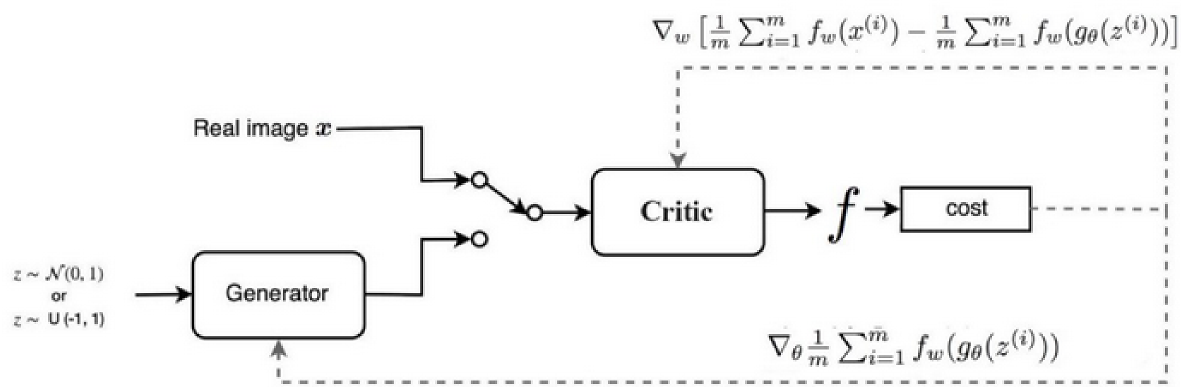
https://github.com/hwalsuklee/tensorflow-generative-model-collections

WGAN vs GAN

GAN:



WGAN



https://medium.com/@jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490

Tips and tricks

- Use batchnorm, ReLU
- Regularize norm of gradients
- Use one of the new loss functions
- Add noise to inputs or labels
- Append image similarity to avoid mode collapse
- Use labels when available (CGAN)
- …

https://github.com/soumith/talks/blob/master/2017-ICCV_Venice/How_To_Train_a_GAN.pdf
https://towardsdatascience.com/gan-ways-to-improve-gan-performance-acf37f9f59b

Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

Smiling woman Neutral woman Neutral man

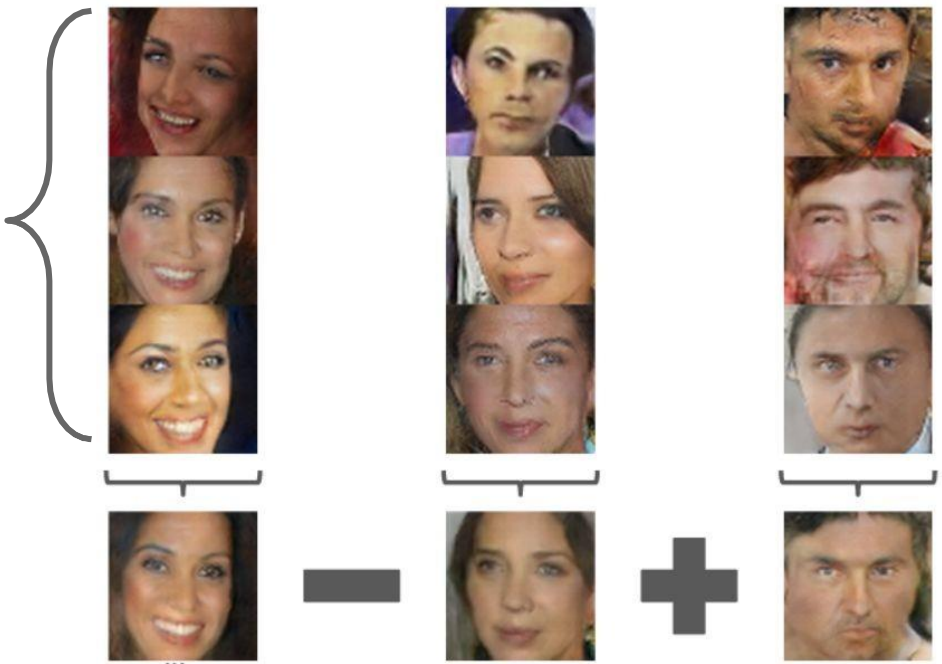
Samples from the model

Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

Smiling woman Neutral woman Neutral man

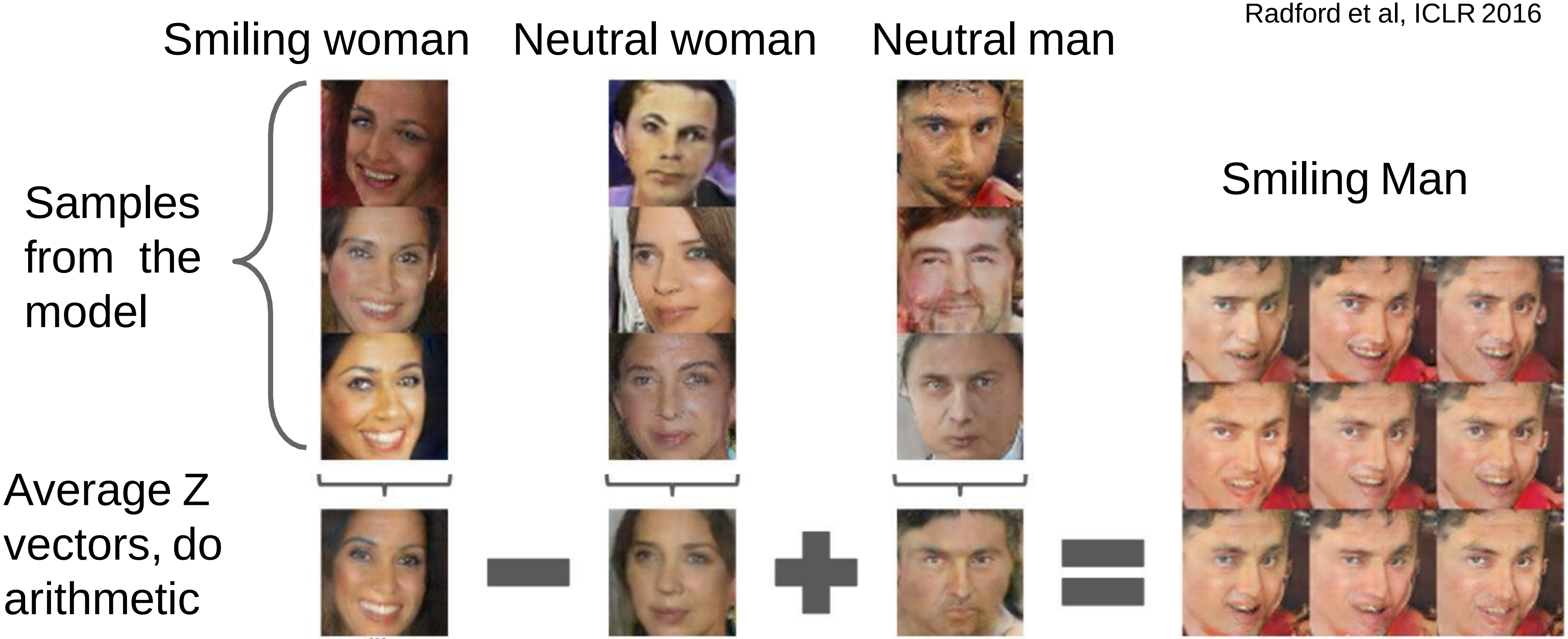
Samples from the model



Average Z vectors, do arithmetic

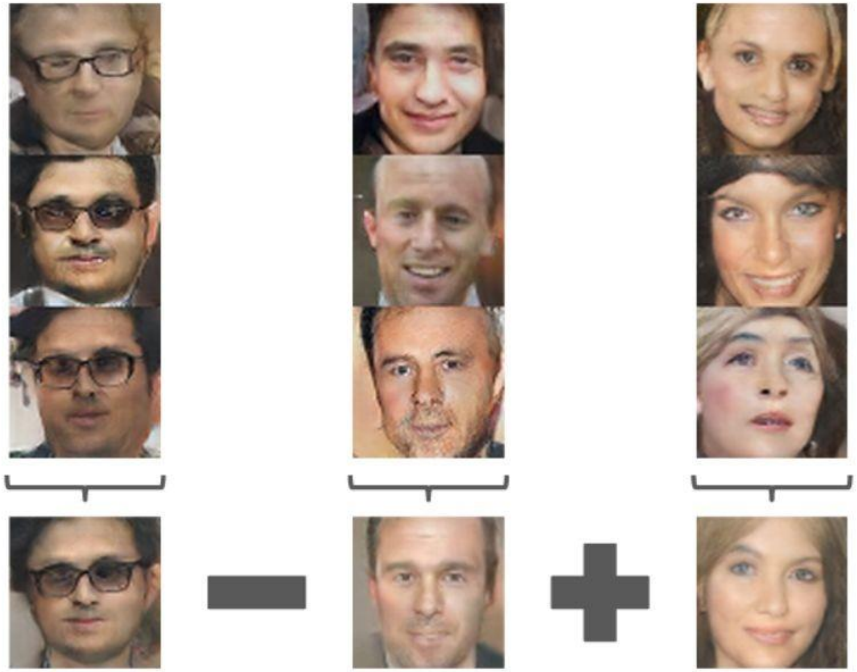
Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016



Generative Adversarial Nets: Interpretable Vector Math

Glasses man No glasses man No glasses woman



Radford et al,
ICLR 2016

Serena Young

Generative Adversarial Nets: Interpretable Vector Math

Radford et al,
ICLR 2016

What is in this image?

(Yeh et al., 2016)

Ian Goodfellow

Generative modeling reveals a face

(Yeh et al., 2016)

Artificial Fashion: vue.ai

Ian Goodfellow

Celebrities Who Never Existed

Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR **2018**

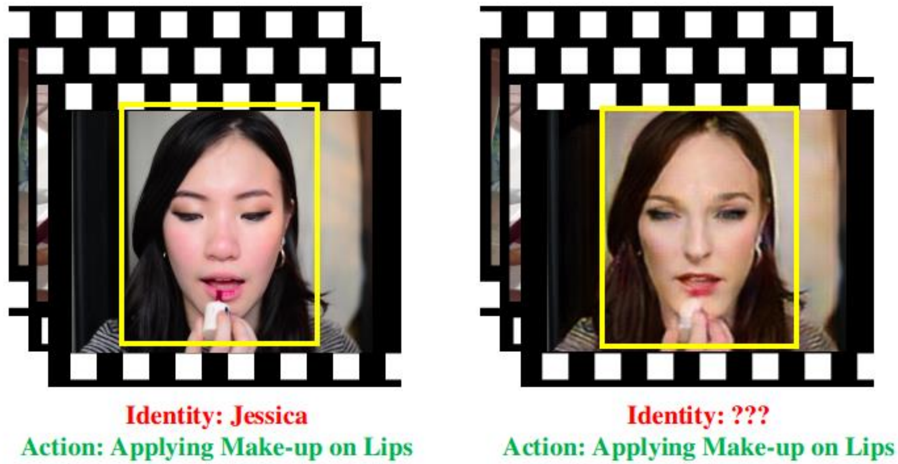
Creative Adversarial Networks

CAN: Top ranked by human subjects

(Elgammal et al., 2017)

Ian Goodfellow

GANs for Privacy (Action Detection)

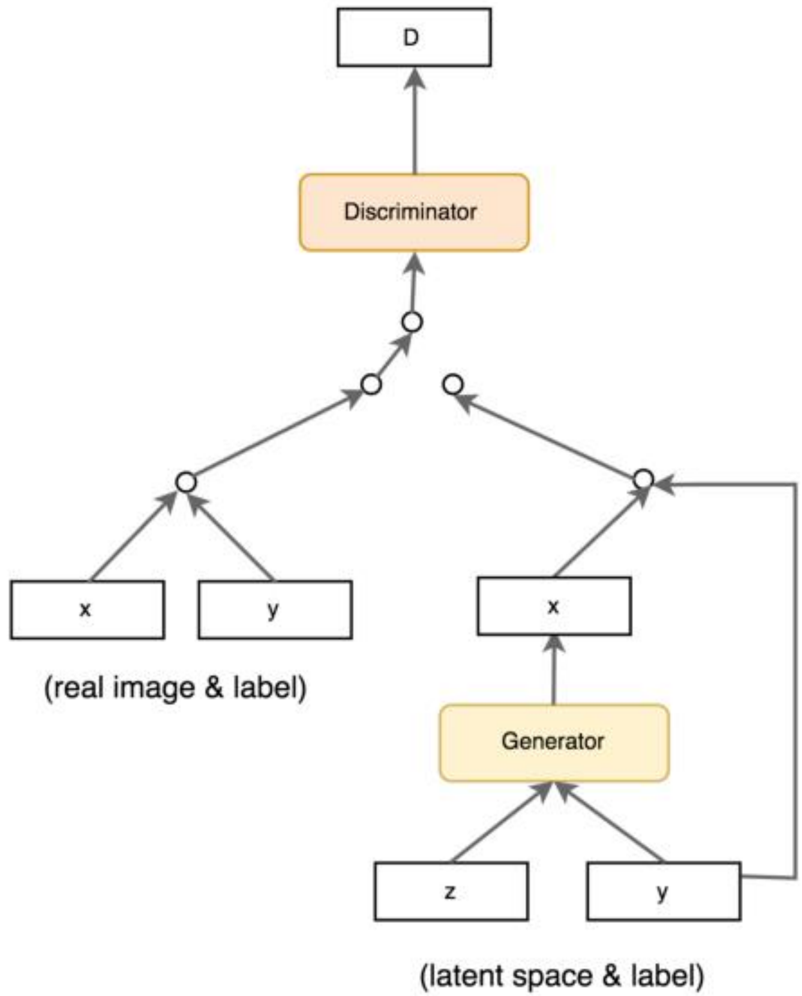


Ren et al., "Learning to Anonymize Faces for Privacy Preserving Action Detection", ECCV 2018

Plan for this lecture

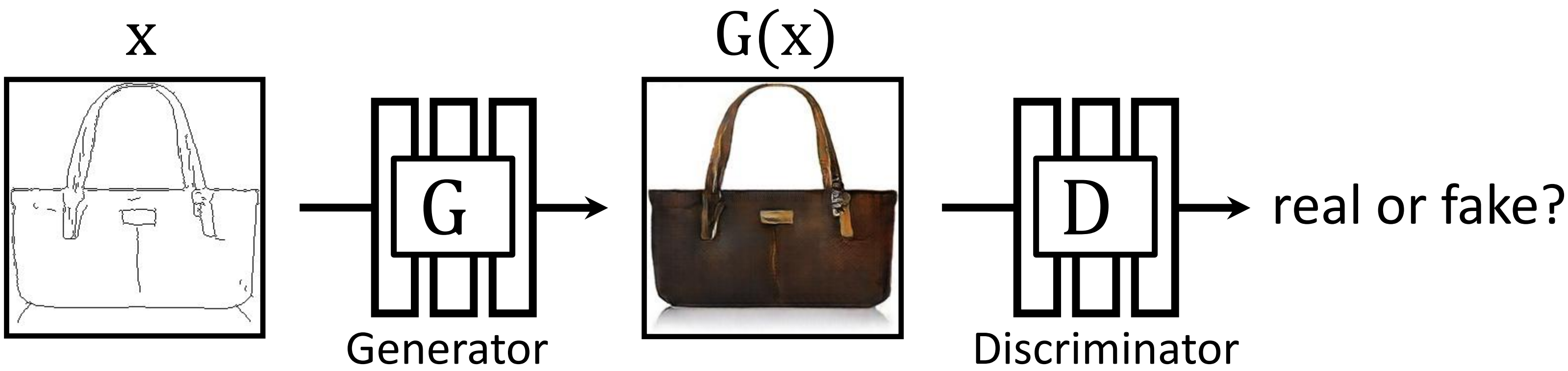
- Generative models: What are they?
- Technique: Generative Adversarial Networks
- Applications
- Conditional GANs
- Cycle-consistency loss
- Dealing with sparse data, progressive training

Conditional GANs



https://medium.com/@jonathan_hui/gan-cgan-infogan-using-labels-to-improve-gan-8ba4de5f9c3d

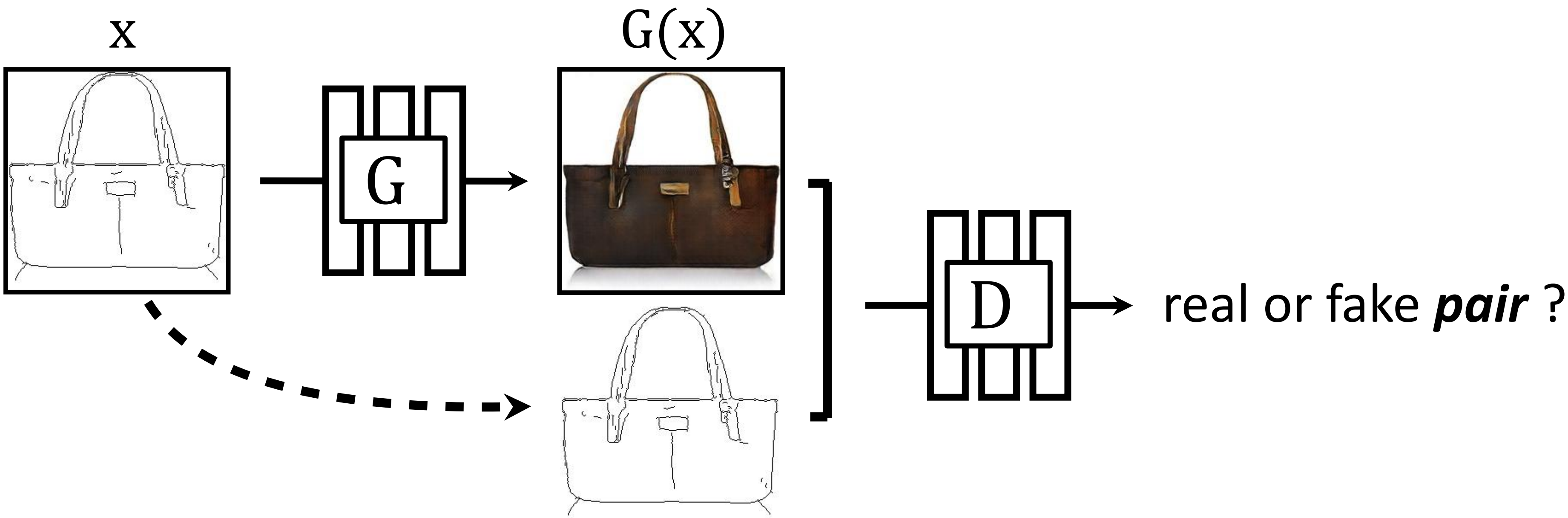
GANs



G: generate fake samples that can fool D
D: classify fake samples vs. real images

[Goodfellow et al. 2014]

Conditional GANs



Adapted from Jun-Yan Zhu

Edges → Images

Input Output Input Output Input Output

Edges from [Xie & Tu, 2015]

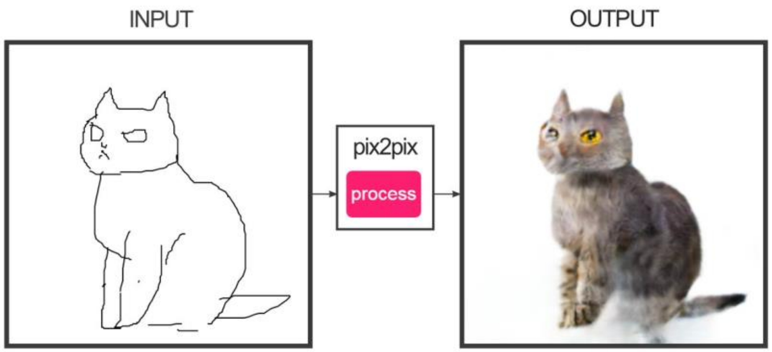
Pix2pix / CycleGAN

Sketches → Images

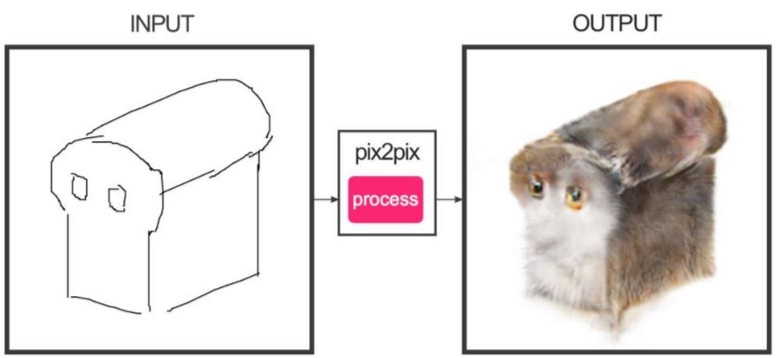
Trained on Edges → Images

Data from [Eitz, Hays, Alexa, 2012]

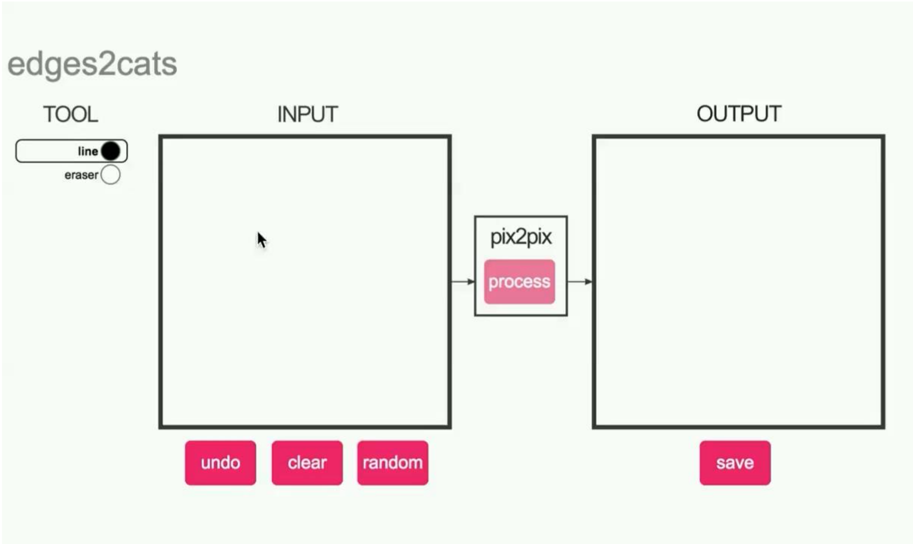
#edges2cats [Christopher Hesse]



@gods_tail



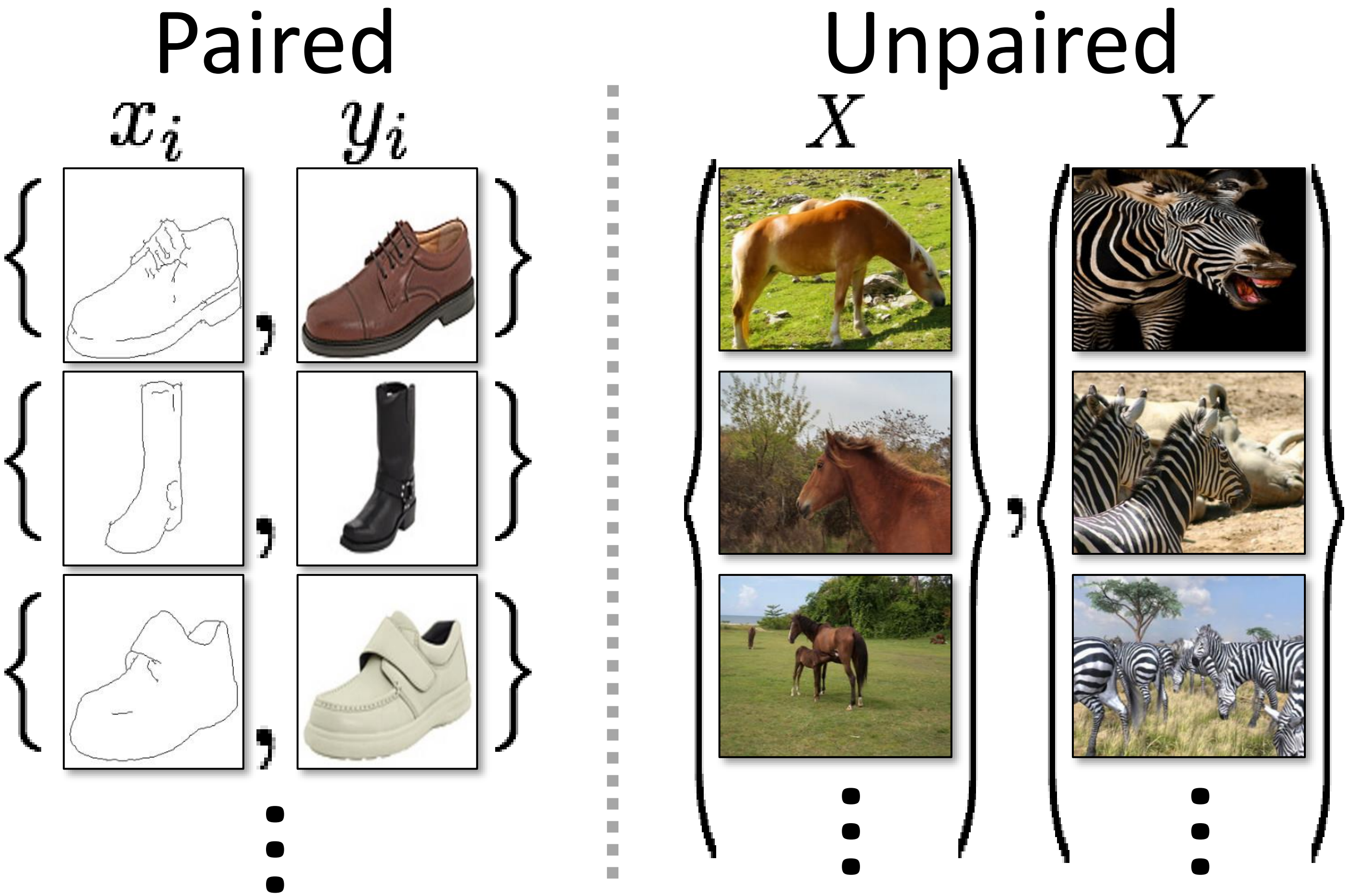
Ivy Tasi @ivymyt



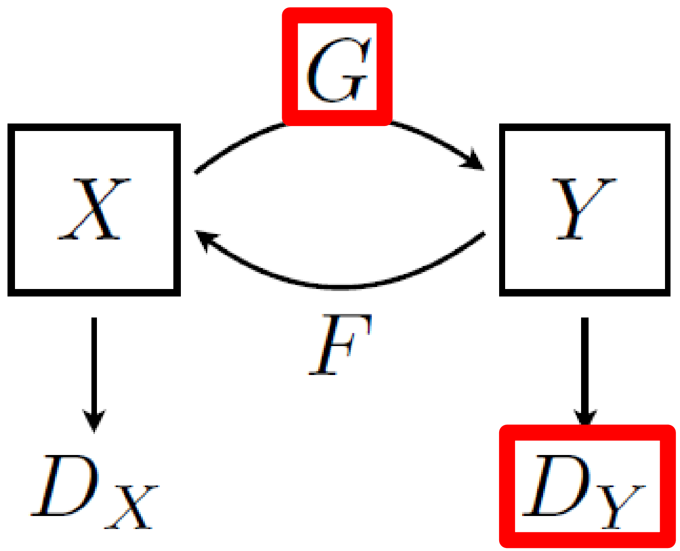
@matthematician

Vitaly Vidmirov @vvid

https://affinelayer.com/pixsrv/


Paired
Unpaired
x_i
y_i
X
Y

Cycle Consistency

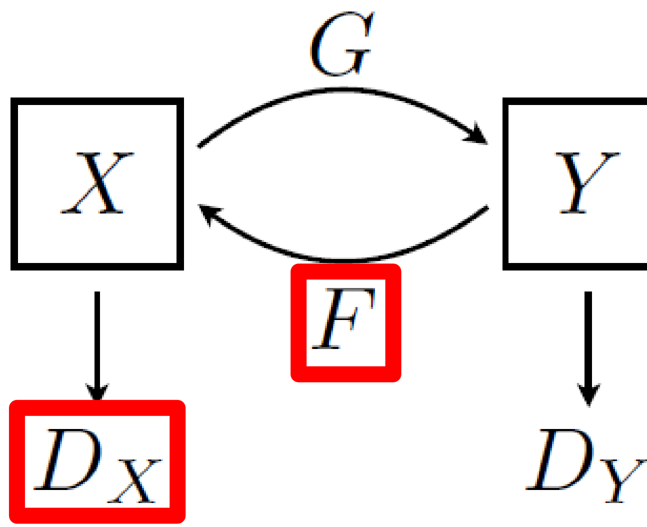


Discriminator $\mathrm{D_Y}$: $L_{GAN}(G(x), y)$

Real zebras vs. generated zebras

Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle Consistency

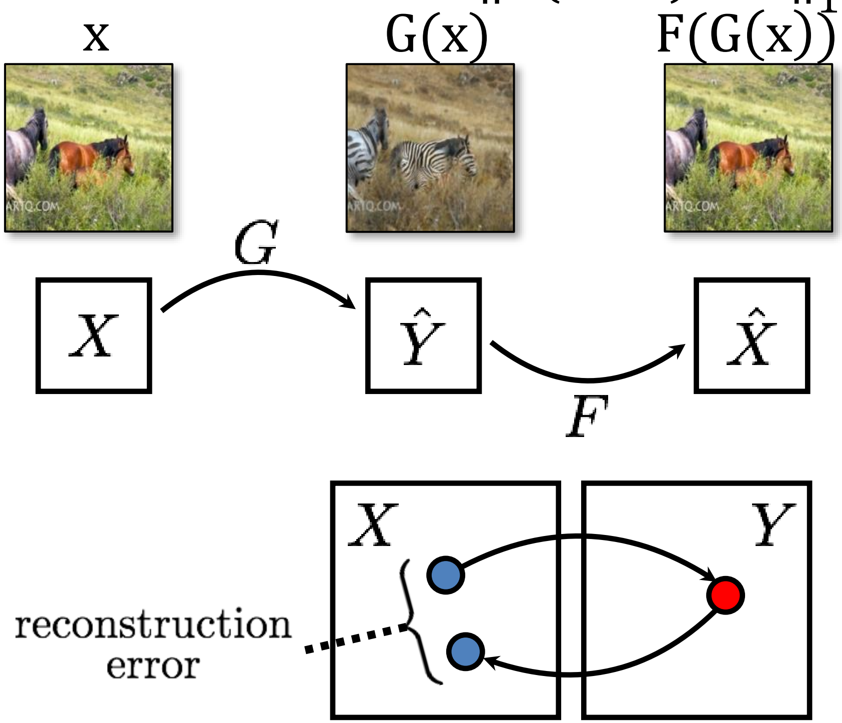


Discriminator $\mathrm{D_Y}$: $L_{GAN}(G(x), y)$
Real zebras vs. generated zebras
Discriminator $\mathrm{D_X}$: $L_{GAN}(F(y), x)$
Real horses vs. generated horses

Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle Consistency

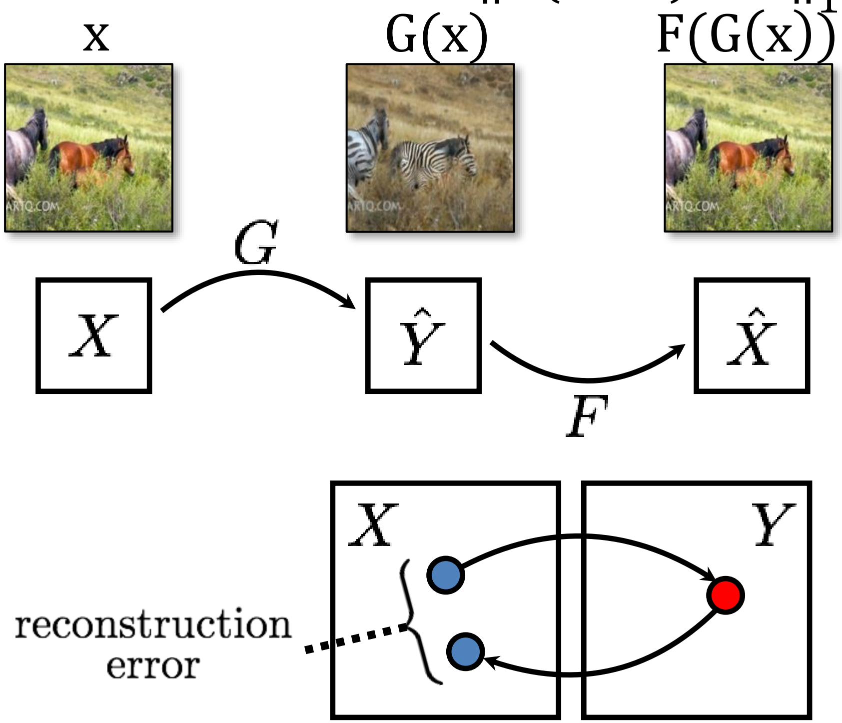
Forward cycle loss: $\left\|F(G(x)) - x\right\|_1$



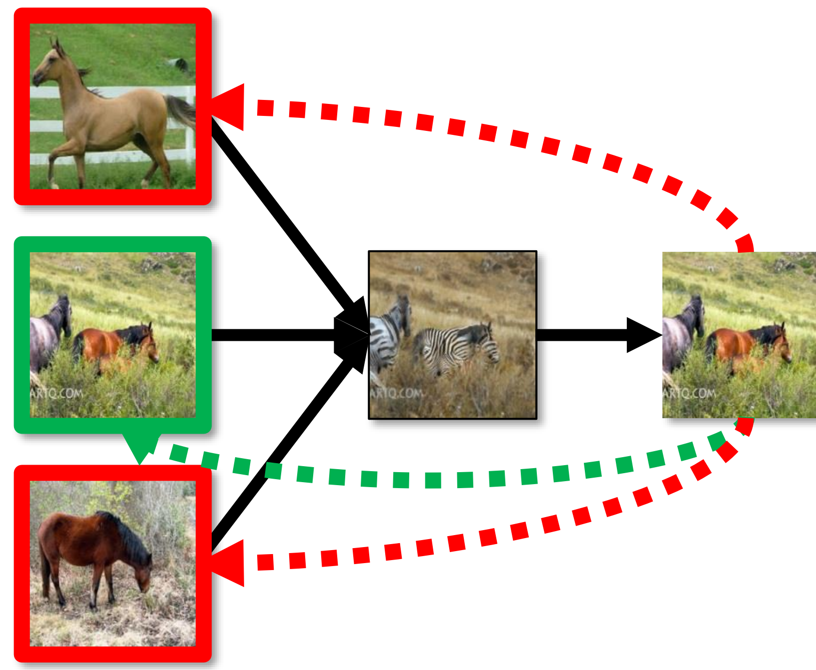
Zhu et al., "[Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks]", ICCV 2017

Cycle Consistency

Forward cycle loss: $\left\| F(G(x)) - x \right\|_1$



Small Large cycle loss



Helps cope with mode collapse

Zhu et al., "[Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks]", ICCV 2017

Training Details: Objective

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))],$$

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].$$

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F),$$

$$G^*, F^* = \arg \min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y).$$

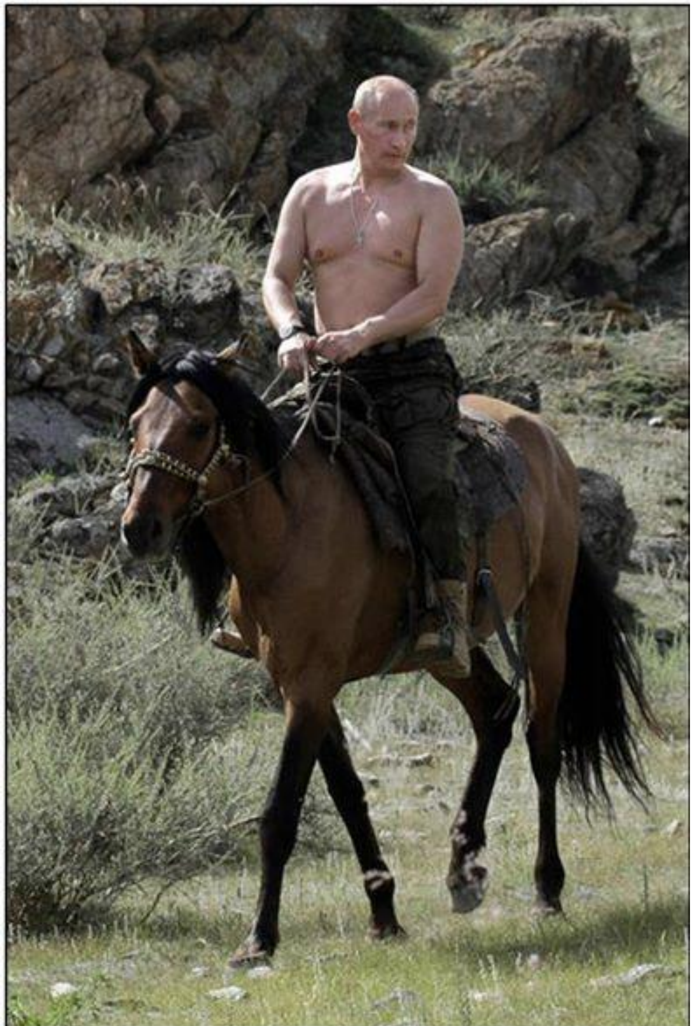
Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Pix2pix / CycleGAN

Pix2pix / CycleGAN

Pix2pix / CycleGAN

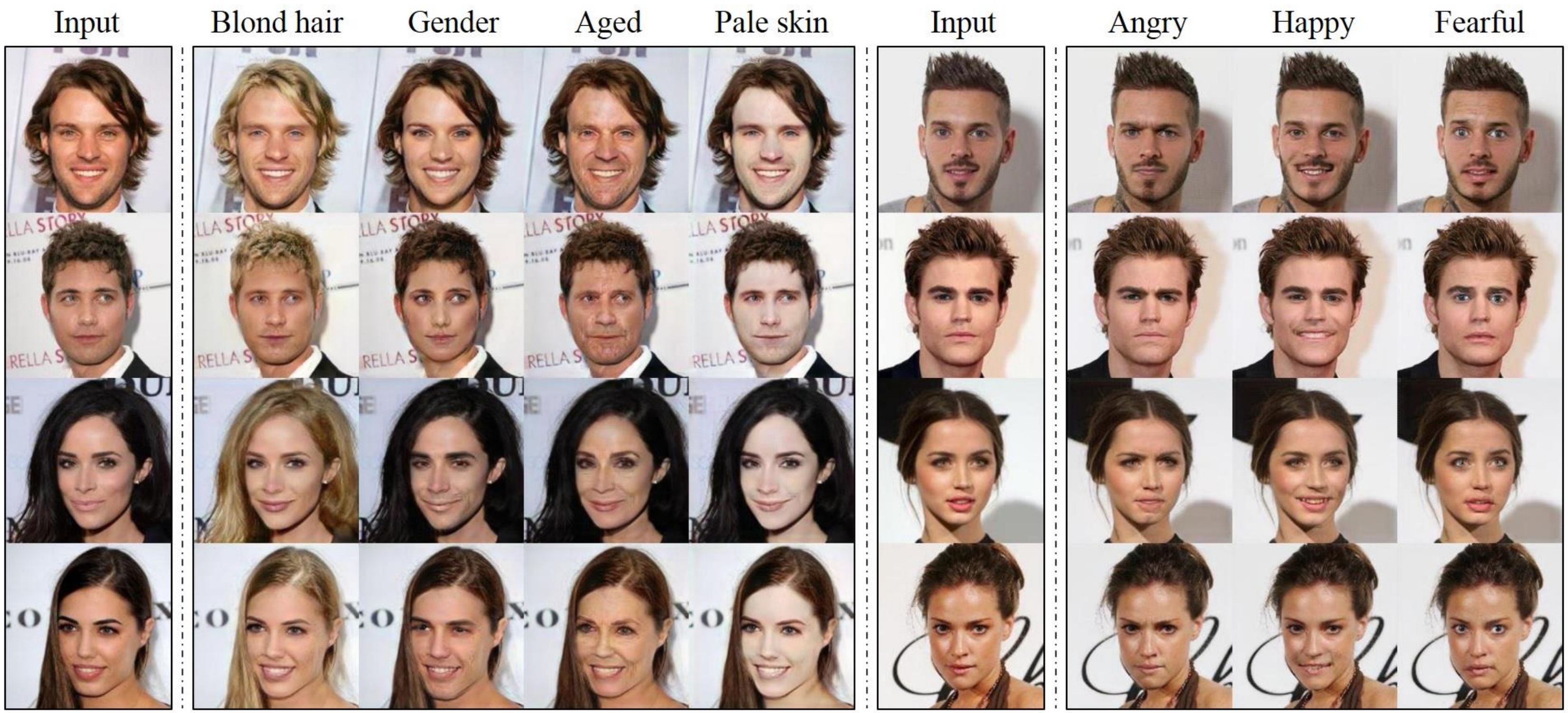
Pix2pix / CycleGAN



Pix2pix / CycleGAN

Pix2pix / CycleGAN

StarGAN



Choi et al., "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", CVPR 2018

Plan for this lecture

- Generative models: What are they?
- Technique: Generative Adversarial Networks
- Applications
- Conditional GANs
- Cycle-consistency loss
- Dealing with sparse data, progressive training

Generating with little data for ads

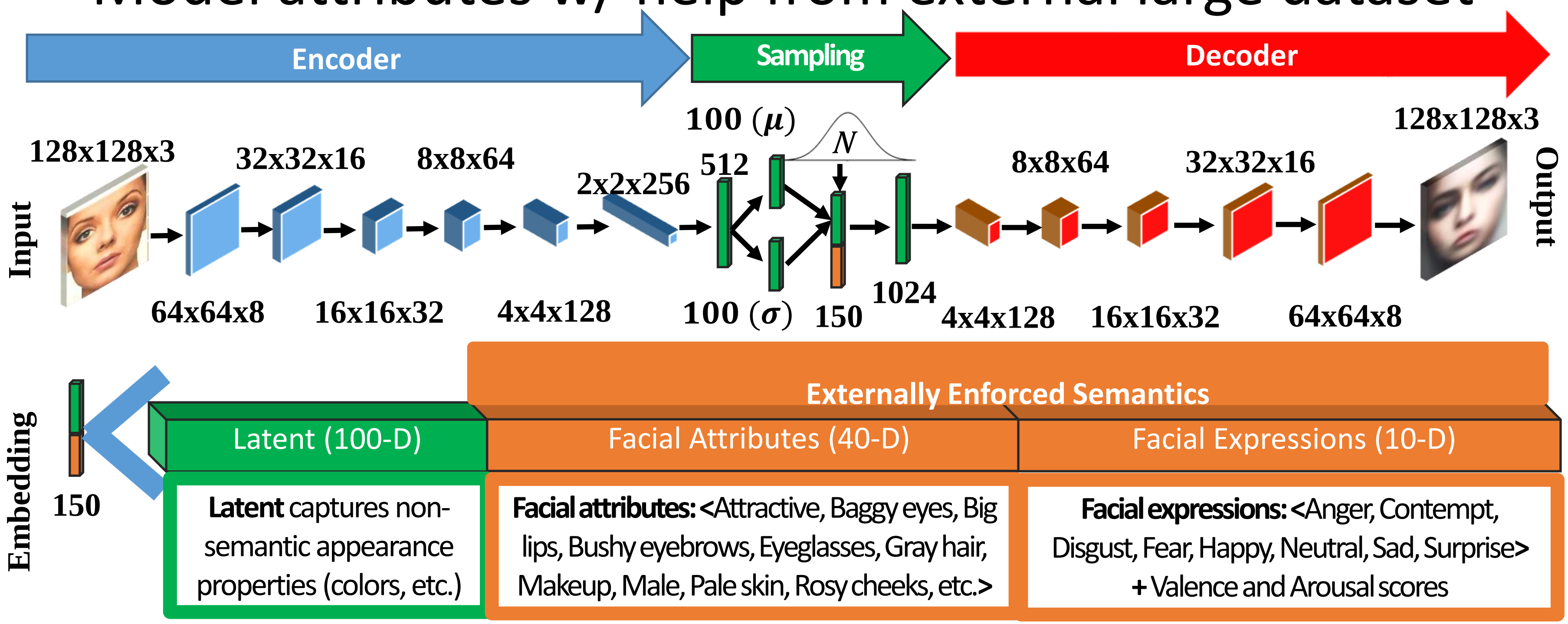
- Faces are persuasive and carry meaning/sentiment

- We learn to generate faces appropriate for each ad category
- Because our data is so diverse yet limited in count, standard approaches that directly model pixel distributions don’t work well

Thomas and Kovashka, BMVC 2018

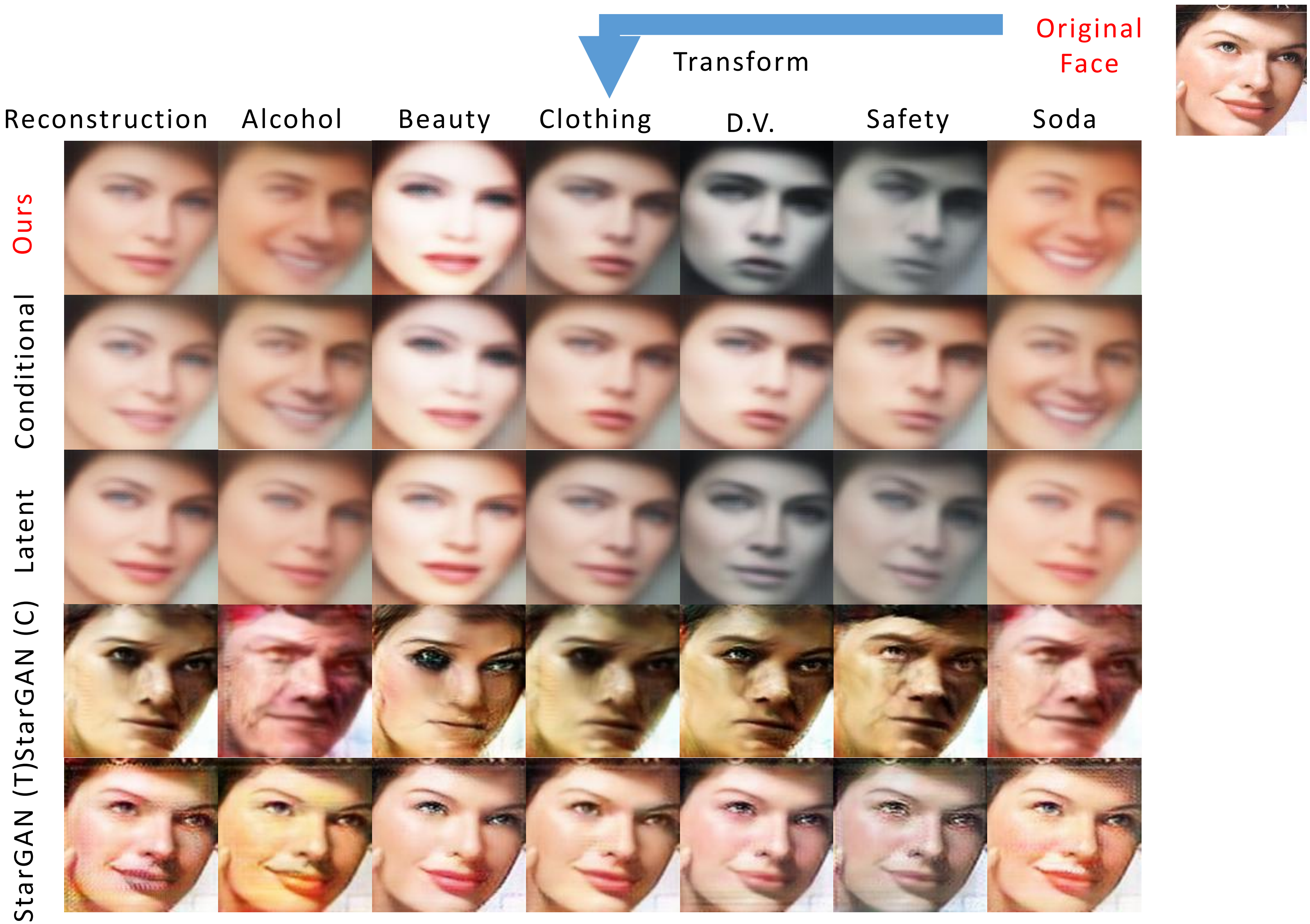
Generating with little data for ads

- Instead we model the distribution over *attributes* for each category (e.g. domestic violence ads contain “black eye”, beauty contains “red lips”)
- Generate an image with the attributes of an ad class
- Model attributes w/ help from external large dataset



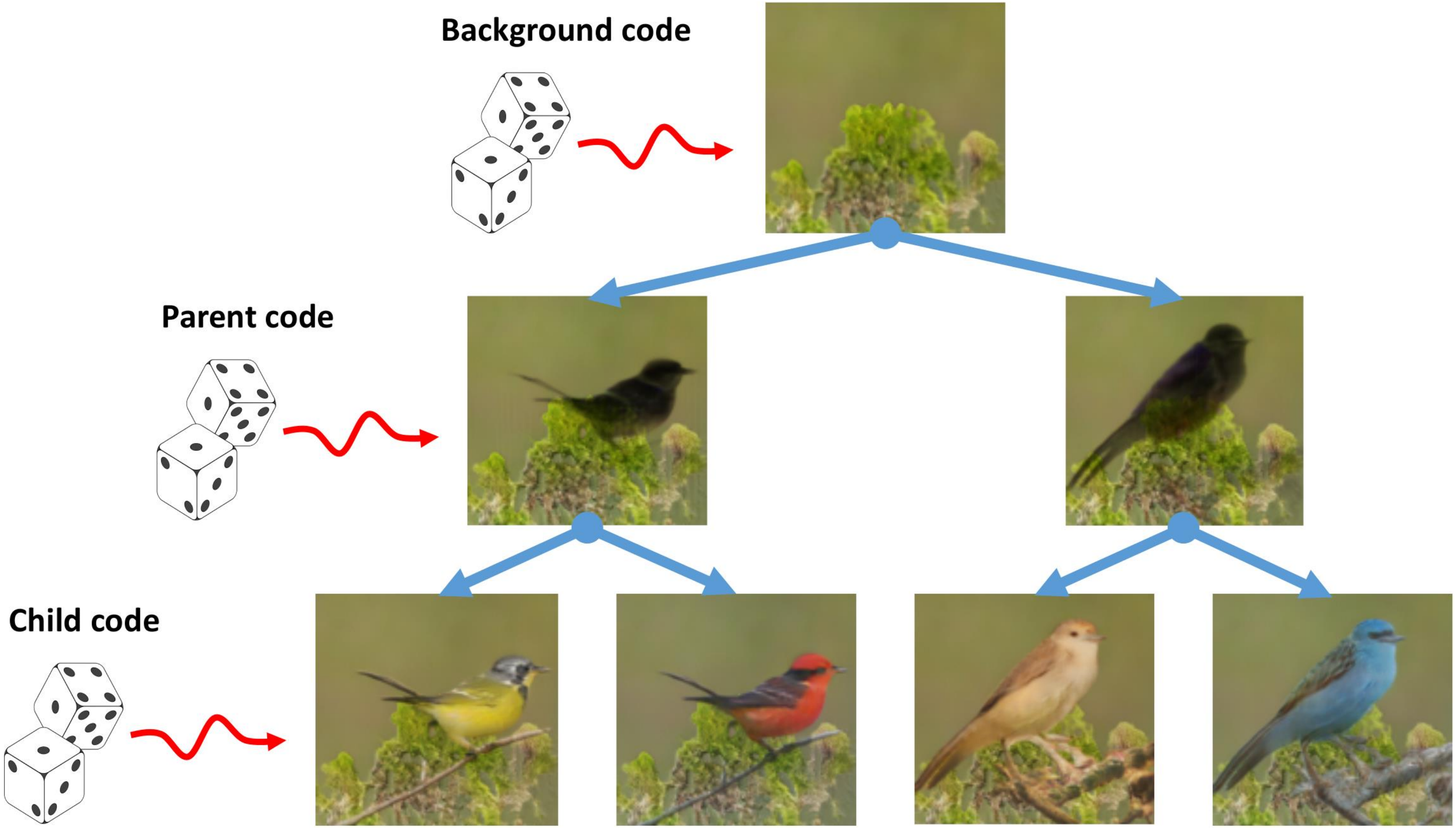
Thomas and Kovashka, BMVC 2018

Generating with little data for ads



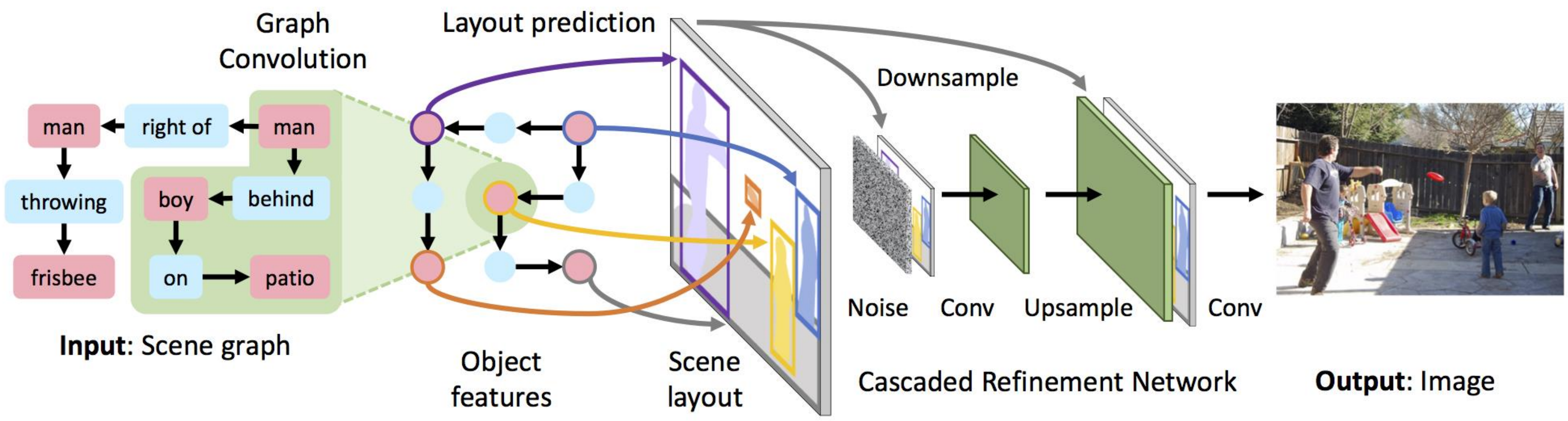
Thomas and Kovashka, BMVC 2018

Stagewise generation



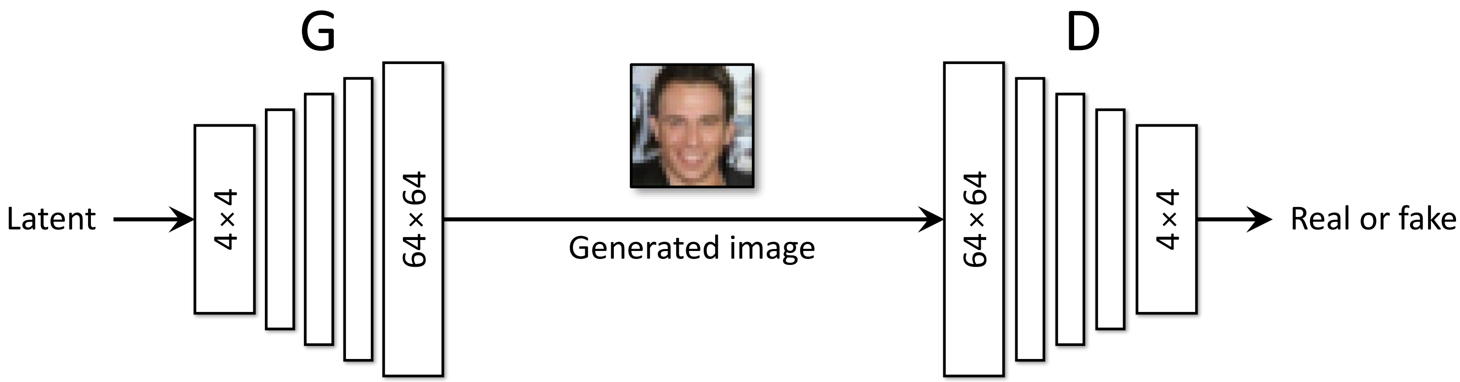
Singh et al., "FineGAN: Unsupervised Hierarchical Disentanglement for Fine-Grained Object Generation and Discovery", CVPR 2019

Stagewise generation



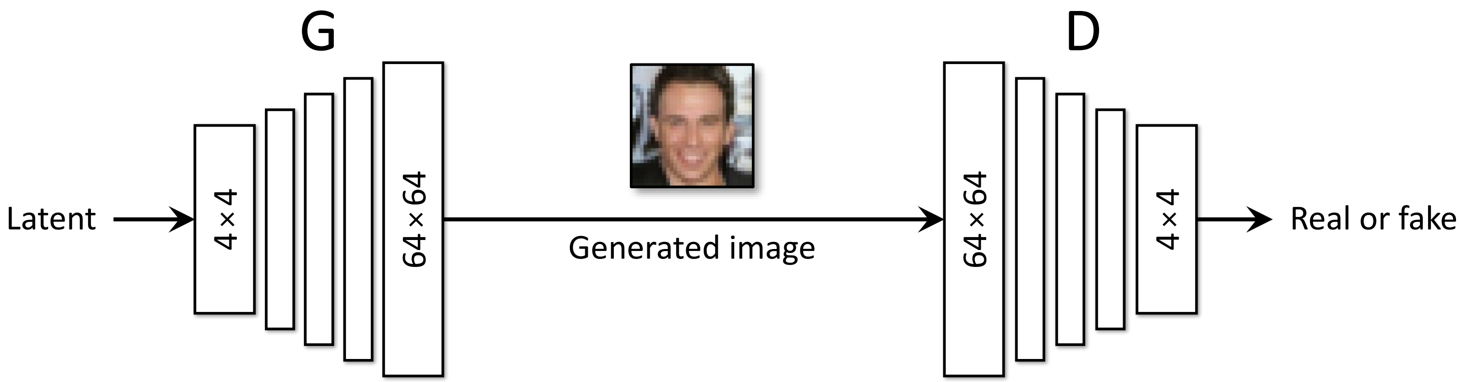
Johnson et al., "Image Generation from Scene Graphs", CVPR 2018

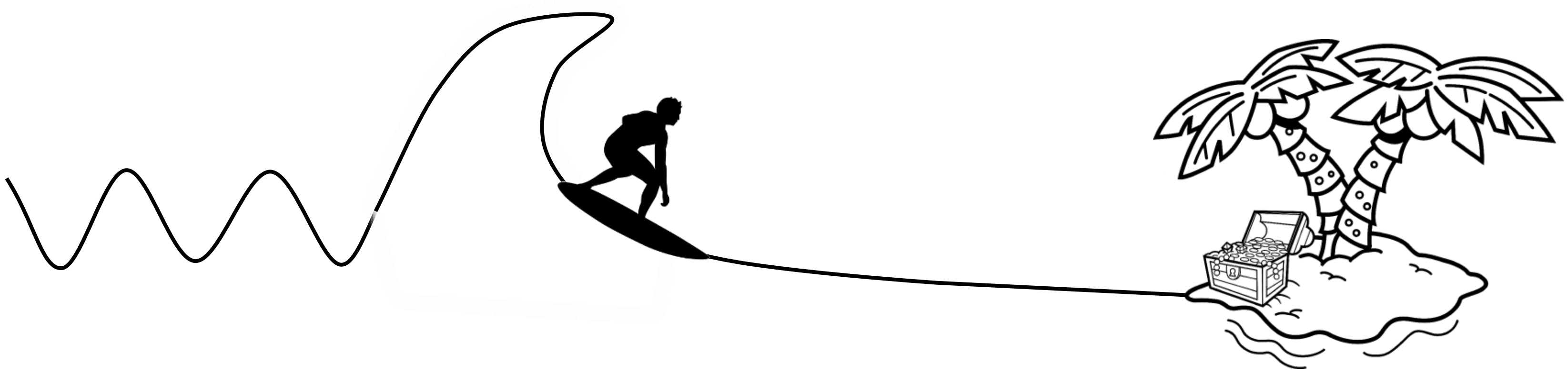
Progressive generation



Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018

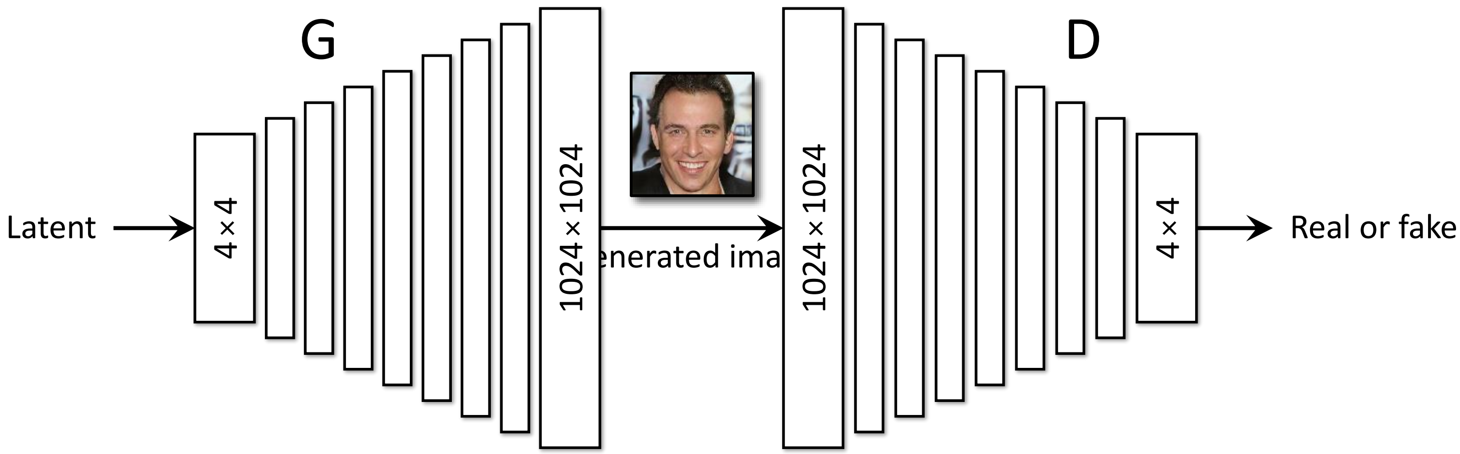
Progressive generation

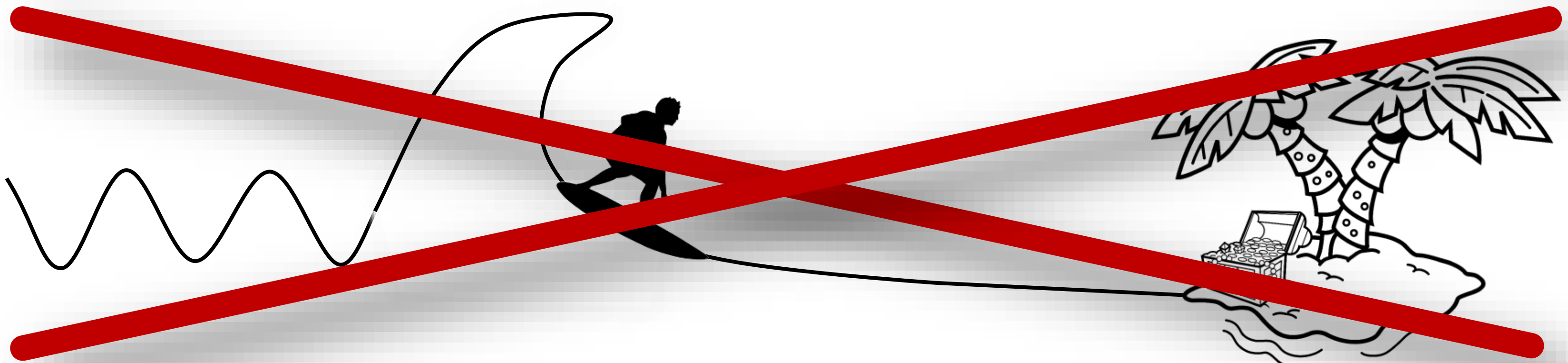




Karras et al., "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR 2018

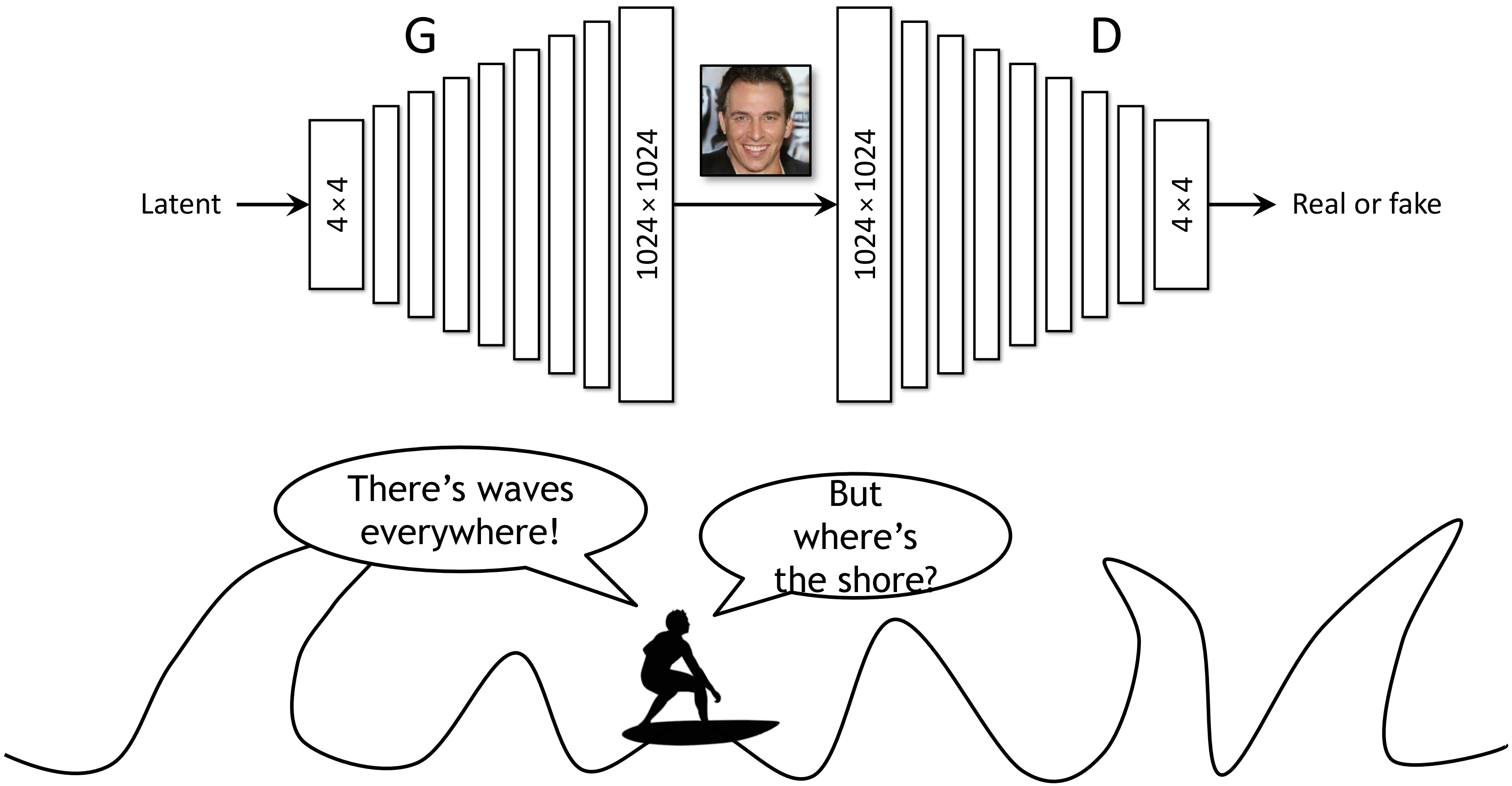
Progressive generation





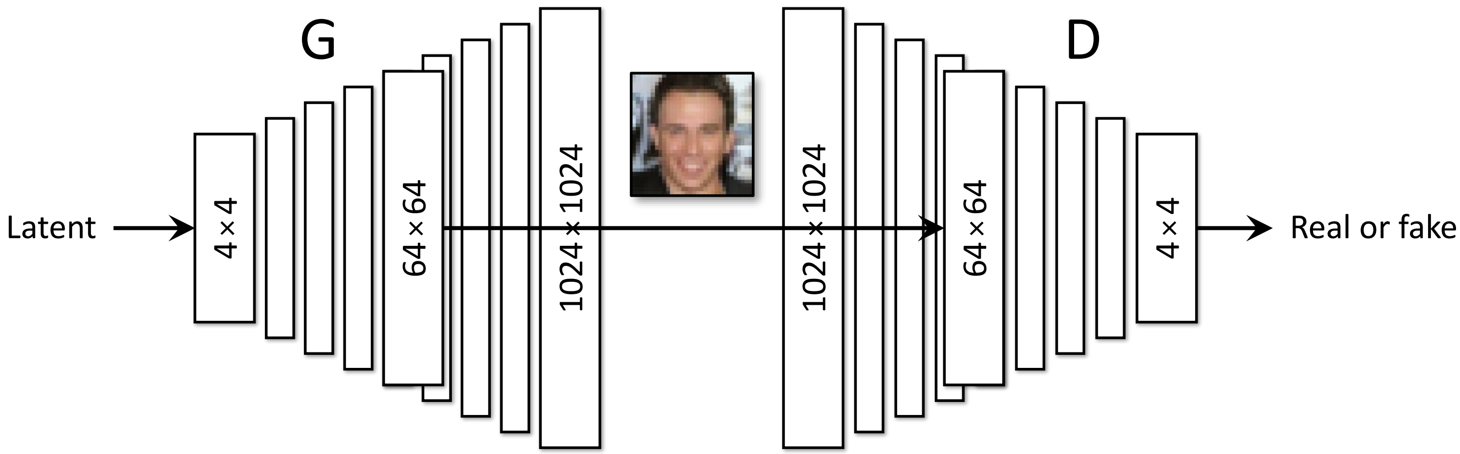
Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018

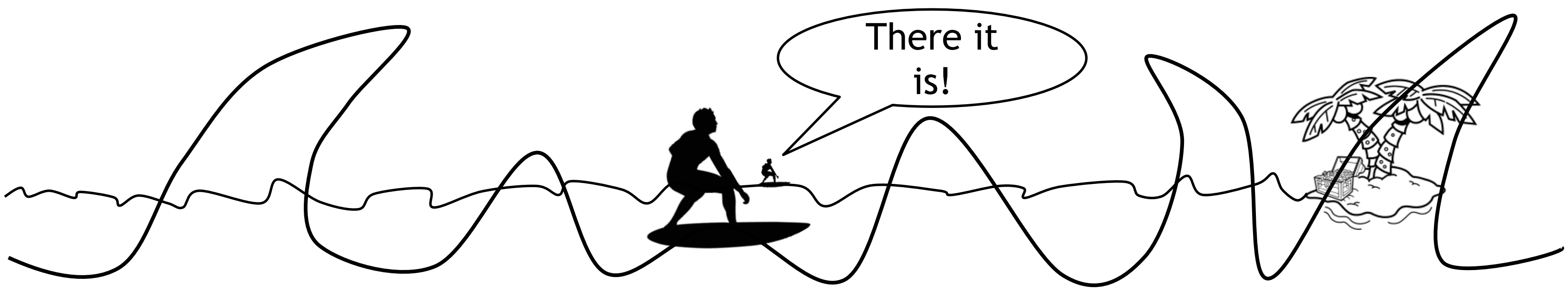
Progressive generation



Karras et al., "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR 2018

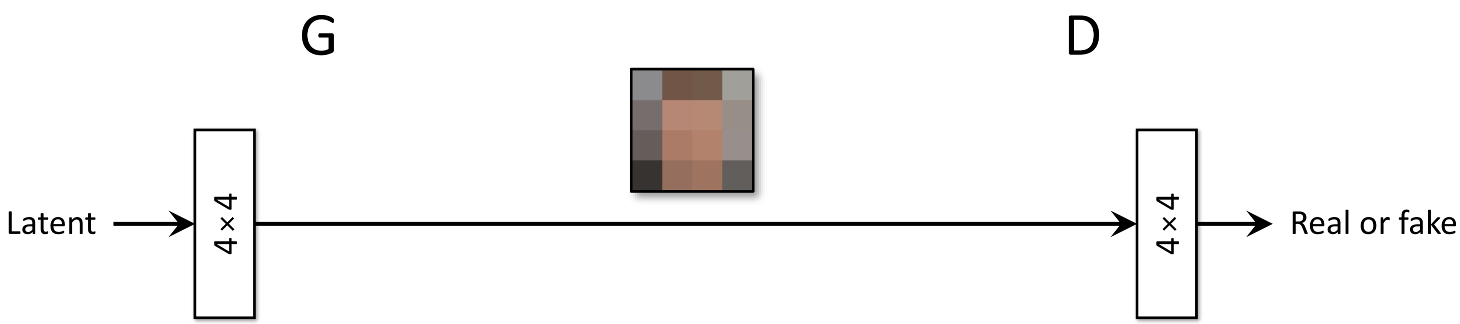
Progressive generation

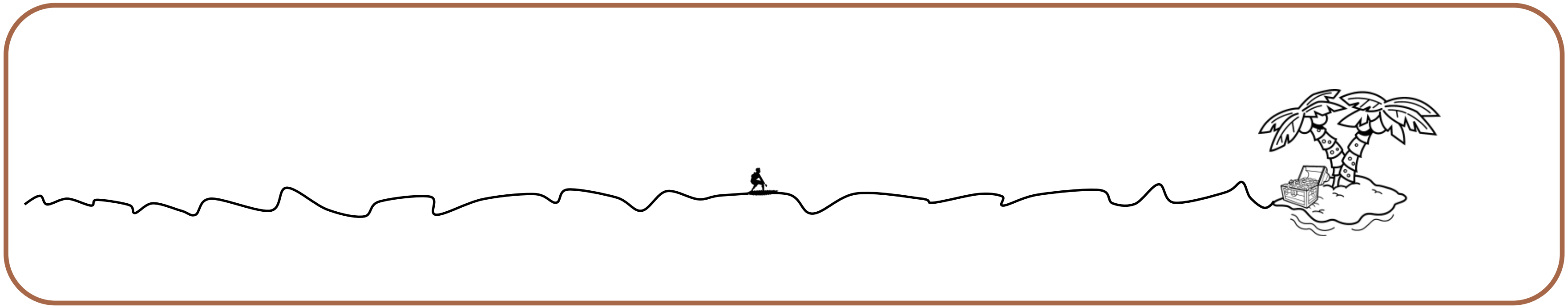




Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018

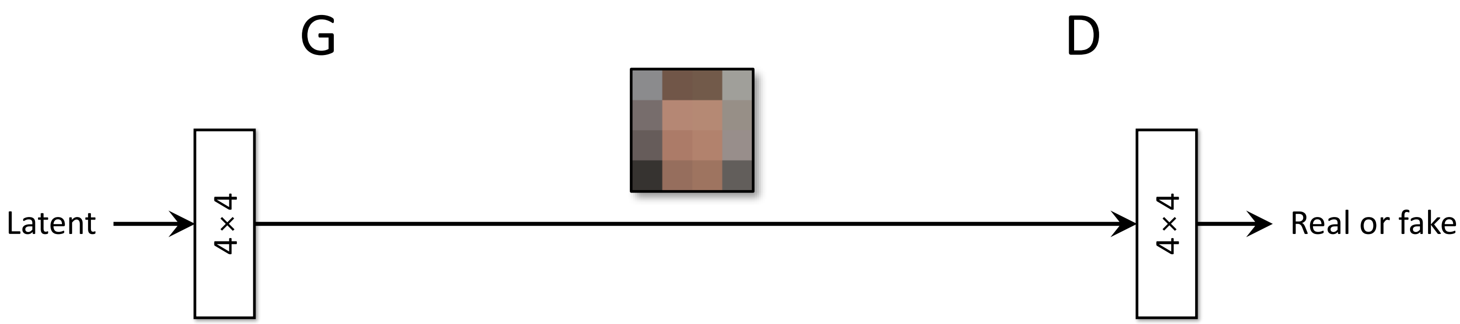
Progressive generation

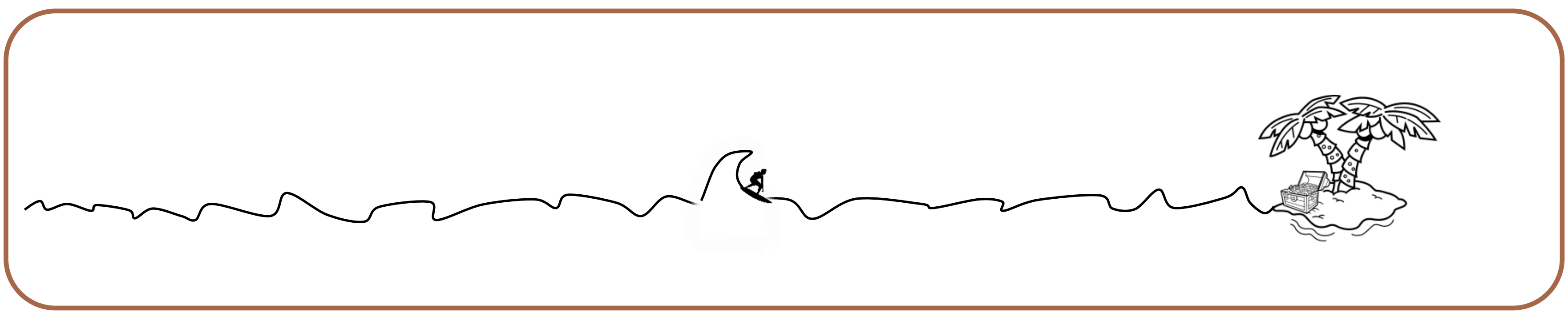




Karras et al., "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR 2018

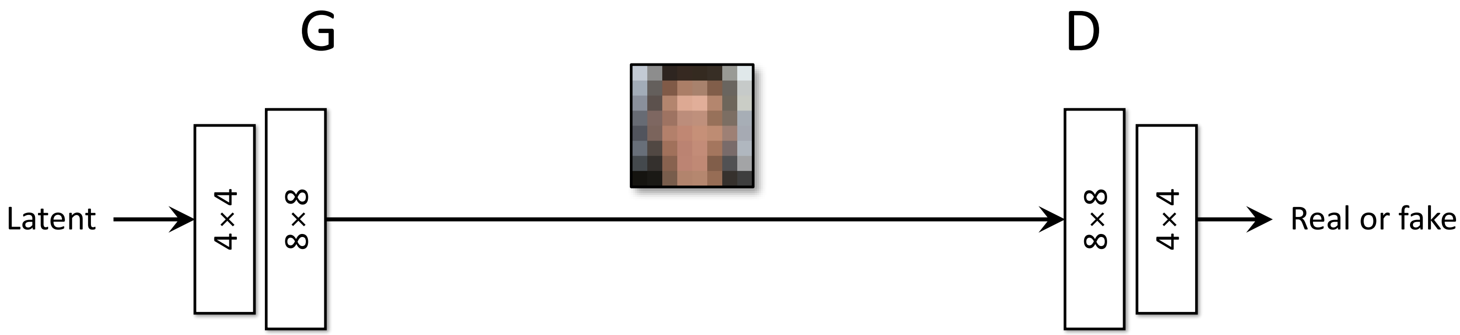
Progressive generation

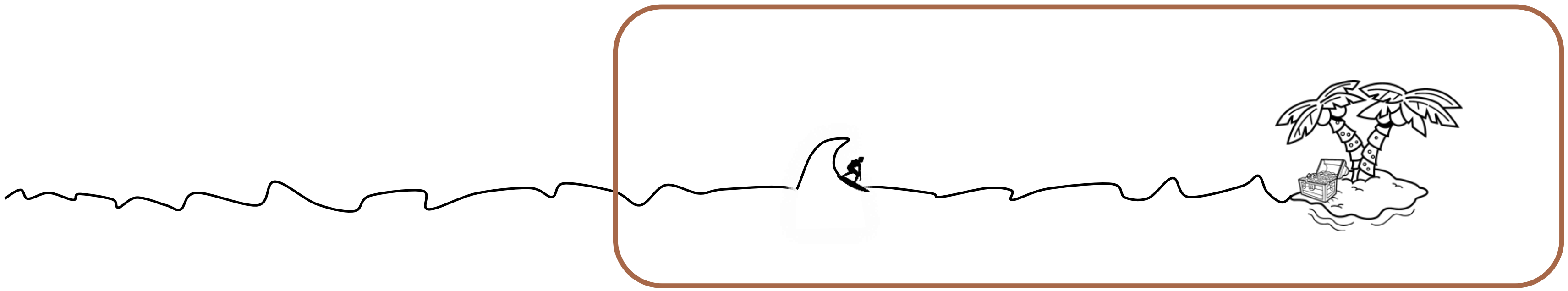




Karras et al., "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR 2018

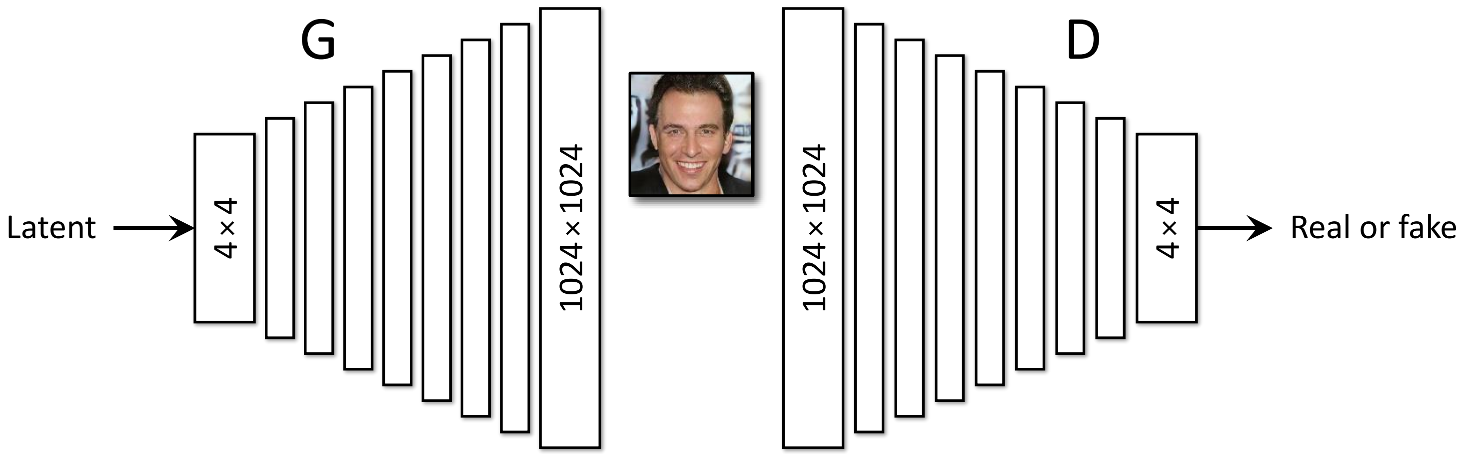
Progressive generation

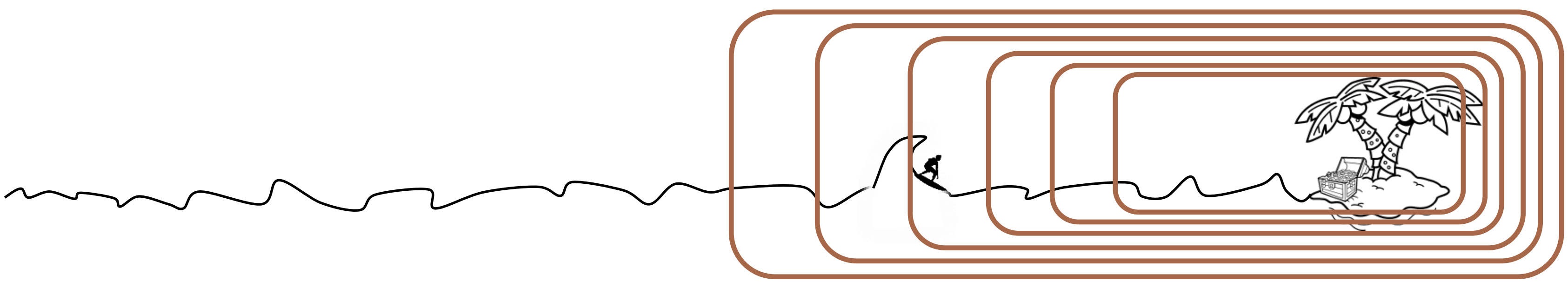




Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018

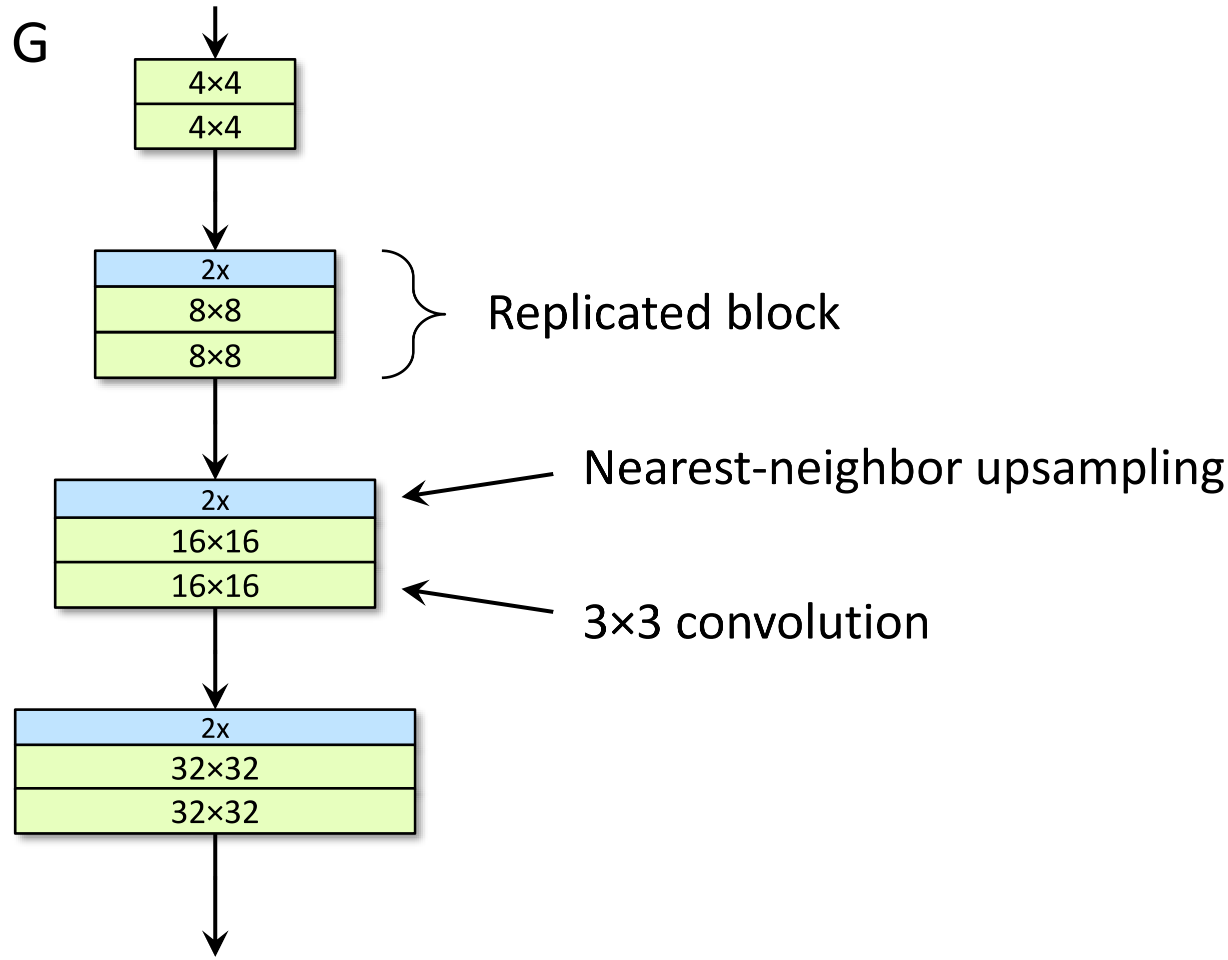
Progressive generation





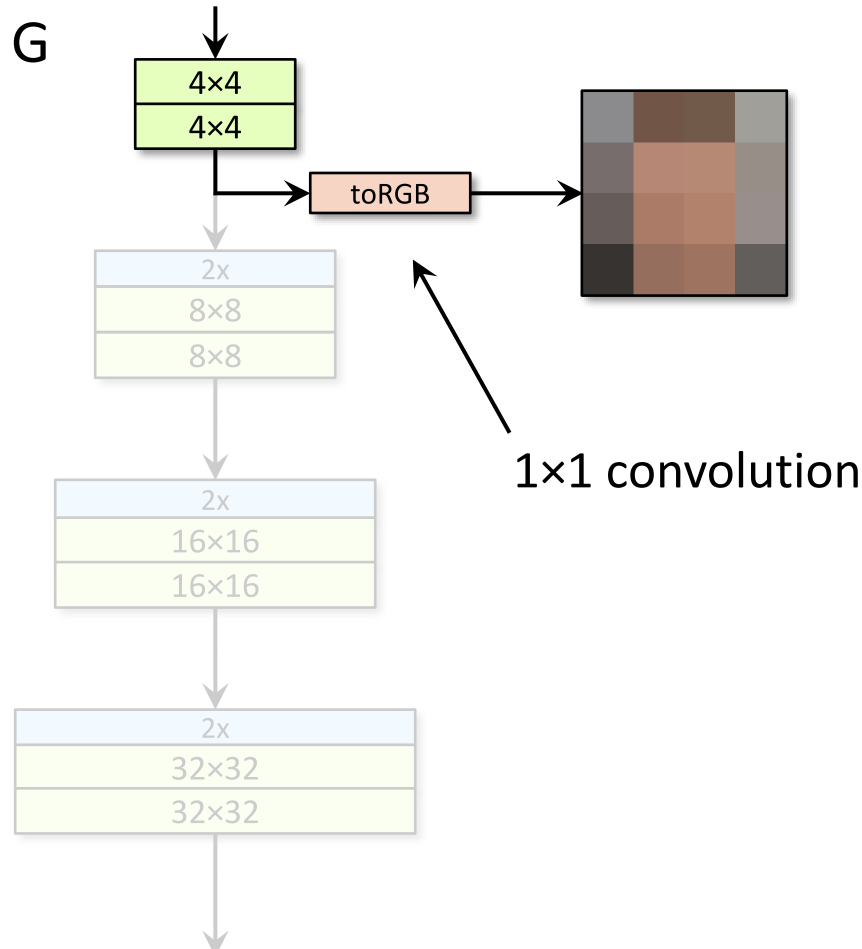
Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018

Progressive generation



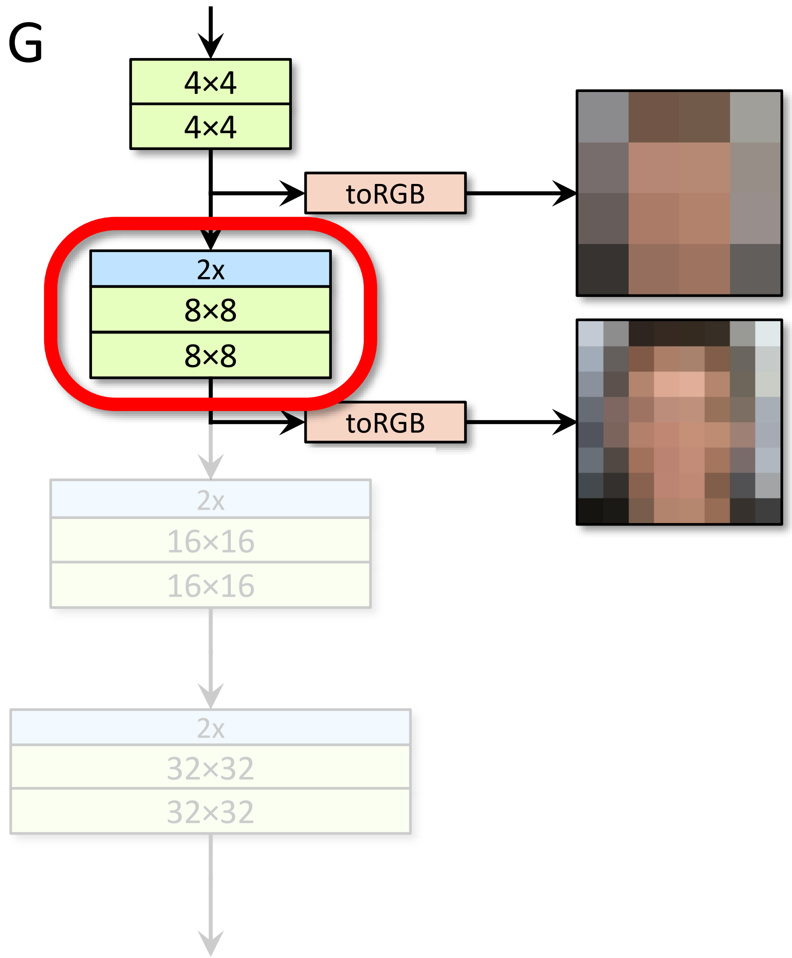
Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018

Progressive generation



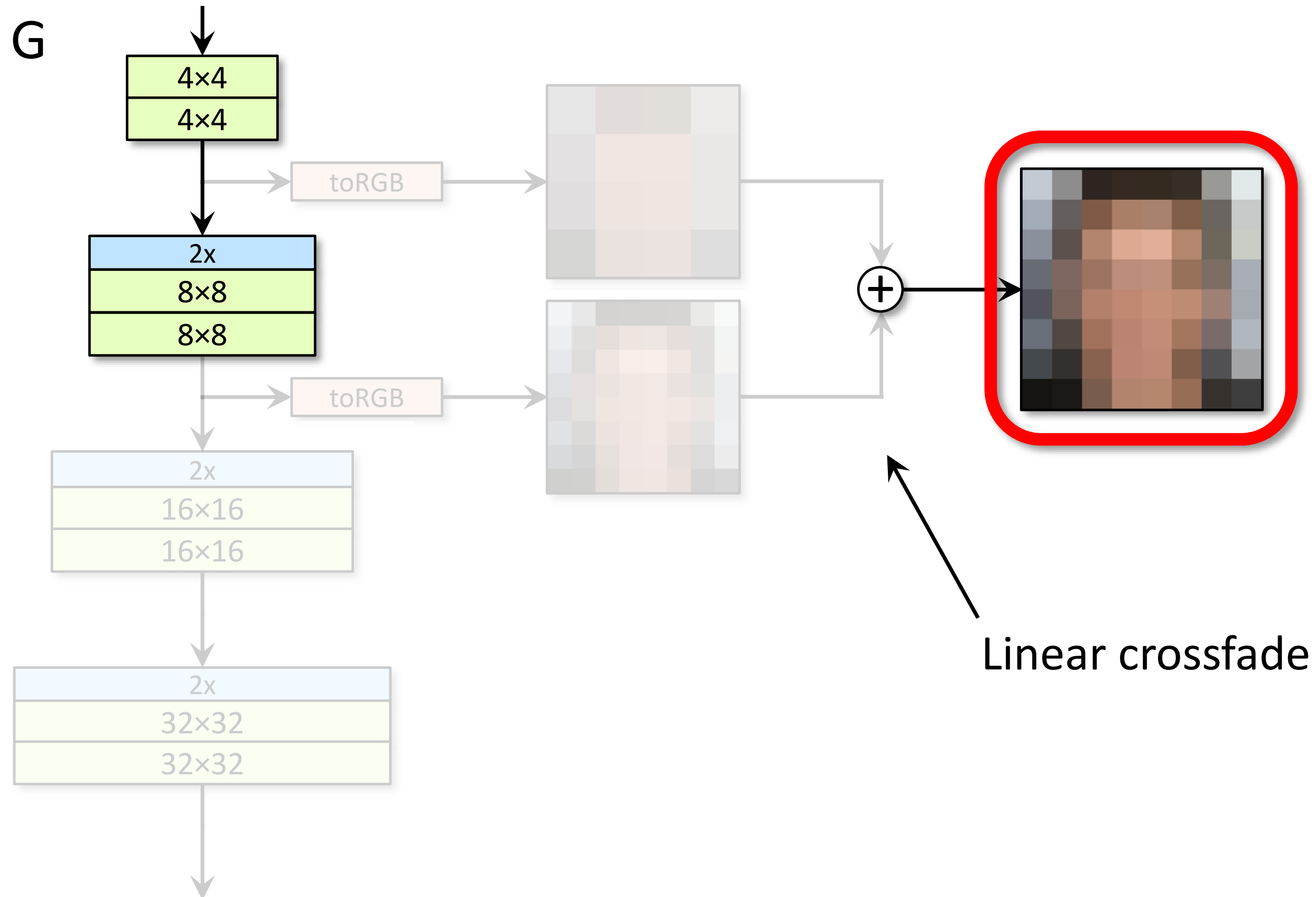
Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018

Progressive generation



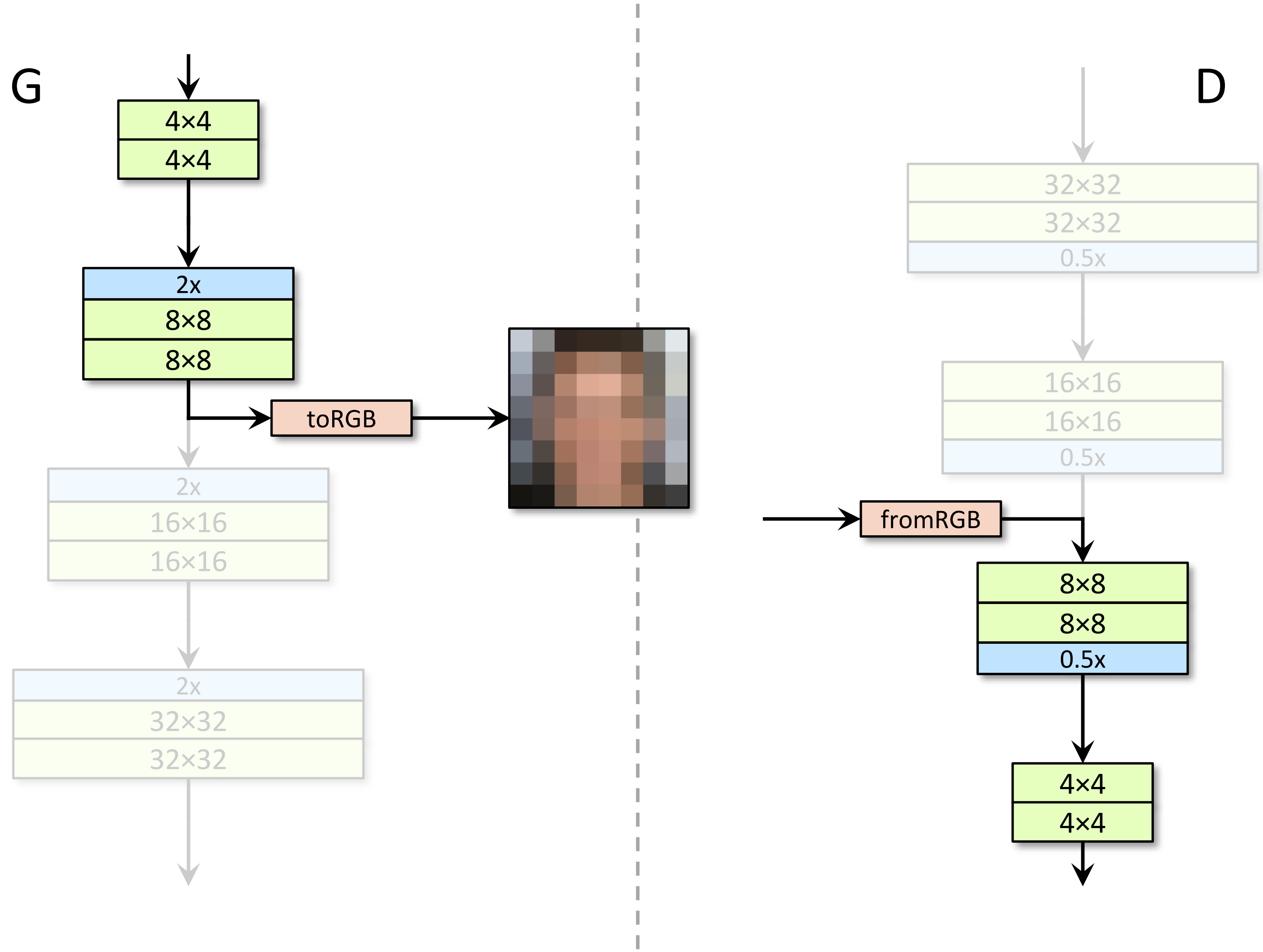
Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018

Progressive generation



Karras et al., "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR 2018

Progressive generation



Karras et al., "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR 2018

What’s next algorithmically?

And what are some social implications?

“Deepfakes”

https://www.technologyreview.com/s/611726/the-defense-department-has-produced-the-first-tools-for-catching-deepfakes/
https://www.niemanlab.org/2018/11/how-the-wall-street-journal-is-preparing-its-journalists-to-detect-deepfakes/

You can be anyone you want…

Karras et al., "A Style-Based Generator Architecture for Generative Adversarial Networks", https://arxiv.org/pdf/1812.04948.pdf