Lecture 3.1
Generative Adversarial Networks

Dimitrios Papadopoulos

Associate Professor, DTU Compute

Last time

- Image classification, CNNs
- Loss functions and optimization
- Transfer Learning
- Saliency maps and Adversarial attacks

- Semantic Segmentation
- Unets, SAM

Today

- Generative Adversarial Networks
- Applications of GANs
- How to train a GAN
- Different GAN models
- Exercise 3

Part 3: GANs

Tuesday 13.6	09:00-11:00 Lecture - 2.1: Introduction to GANs 11:00-12:00 Exercise 2.1: GANs 13:00-17:00 Continue exercise	9-12 Dimitrios 13-15 Manxi 14-16 Paraskevas 15-17 Thanos
Wednesday 14.6	09:00-10:00 Lecture - 3.3: Manipulate image generation *10:00-11:00 Project 2 Poster session* - Give peer feedback to designated groups 11:00-17:00 Work on project	9-12 Dimitrios 13-15 Thanos 14-16 Paraskevas 15-17 Manxi
Thursday 15.6	09:00-17:00 Work on project **Project 3 deadline at midnight**	9-12 Dimitrios 13-15 Manxi 14-16 Paraskevas 15-17 Thanos

Discriminative vs Generative models

Data: x

Discriminative Model:
Learn a probability distribution p(y|x)

Generative Model:
Learn a probability distribution p(x)

Label: y

Cat

Generative models

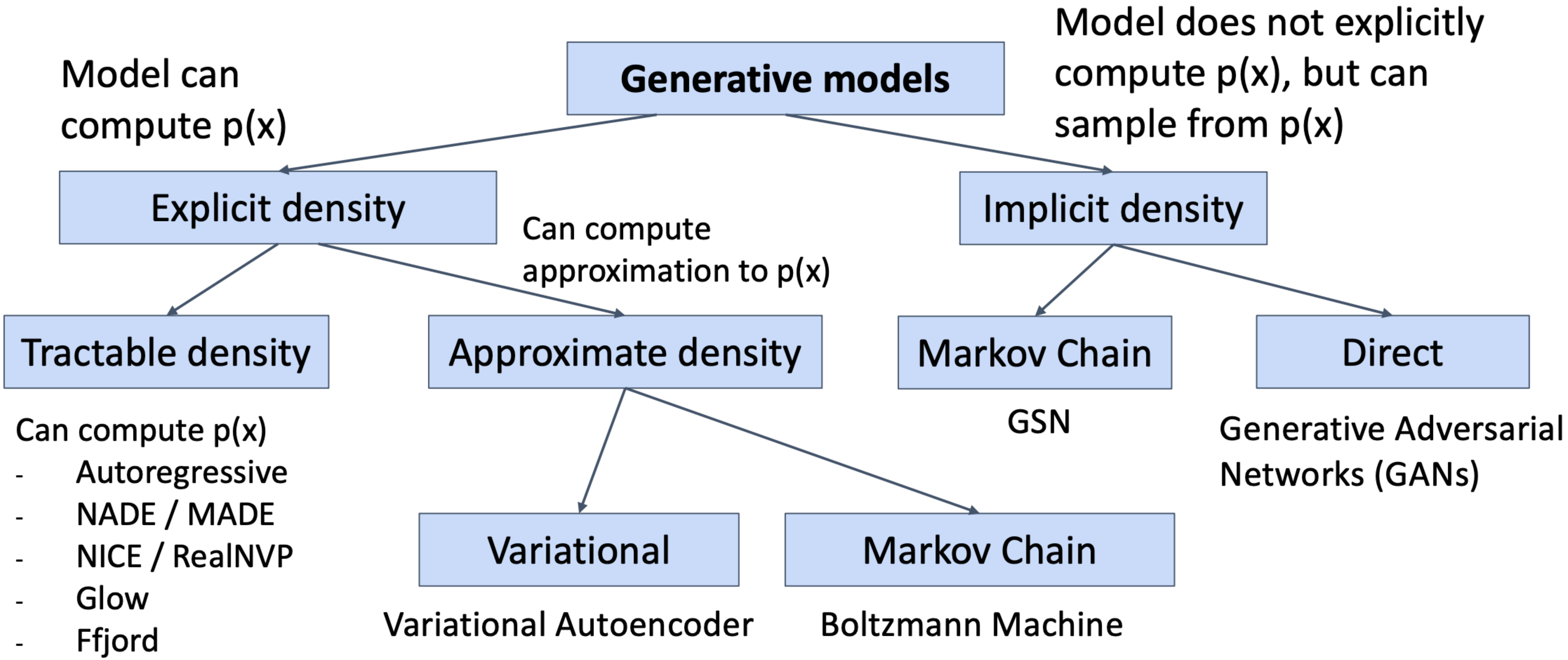


Figure adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Generative models

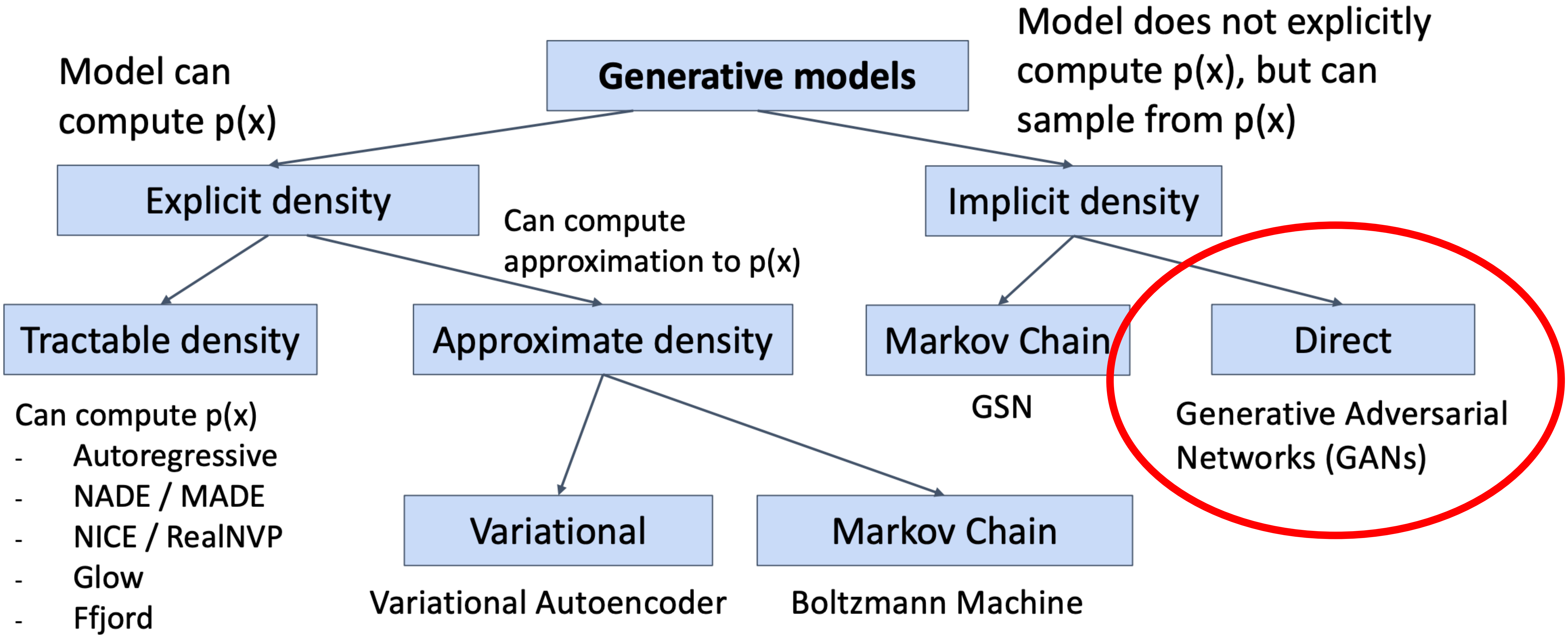


Figure adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Generative Adversarial Networks (GANs)

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie,[*] Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,[†] Aaron Courville, Yoshua Bengio[‡]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

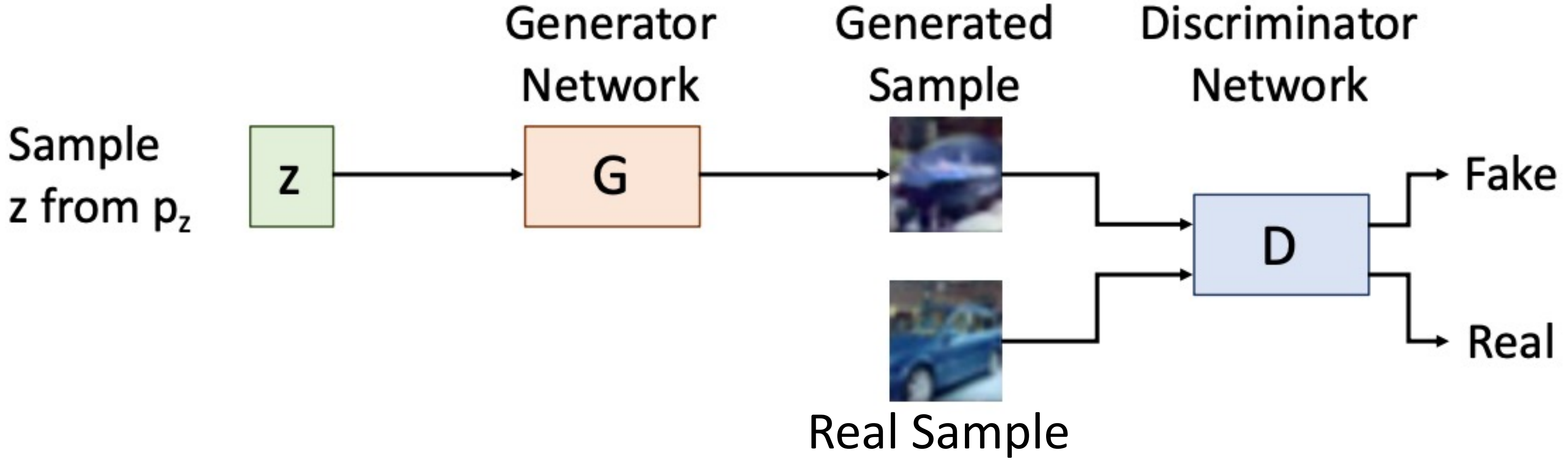
1 Introduction

The promise of deep learning is to discover rich, hierarchical models [2] that represent probability distributions over the kinds of data encountered in artificial intelligence applications, such as natural images, audio waveforms containing speech, and symbols in natural language corpora. So far, the most striking successes in deep learning have involved discriminative models, usually those that map a high-dimensional, rich sensory input to a class label [14, 22]. These striking successes have primarily been based on the backpropagation and dropout algorithms, using piecewise linear units [19, 9, 10] which have a particularly well-behaved gradient . Deep *generative* models have had less of an impact, due to the difficulty of approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies, and due to difficulty of leveraging the benefits of piecewise linear units in the generative context. We propose a new generative model estimation procedure that sidesteps these difficulties. [1]

Introduced in 2014 by Ian Goodfellow

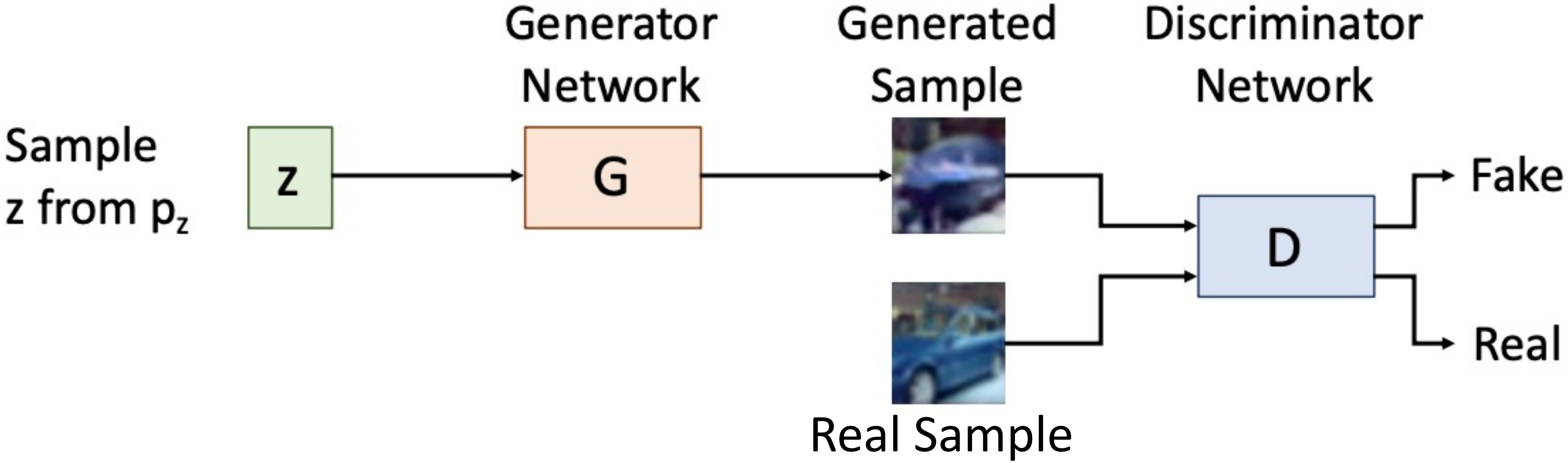
Generative Adversarial Networks (GANs)

Introduced in 2014 by Ian Goodfellow



Generative Adversarial Networks (GANs)

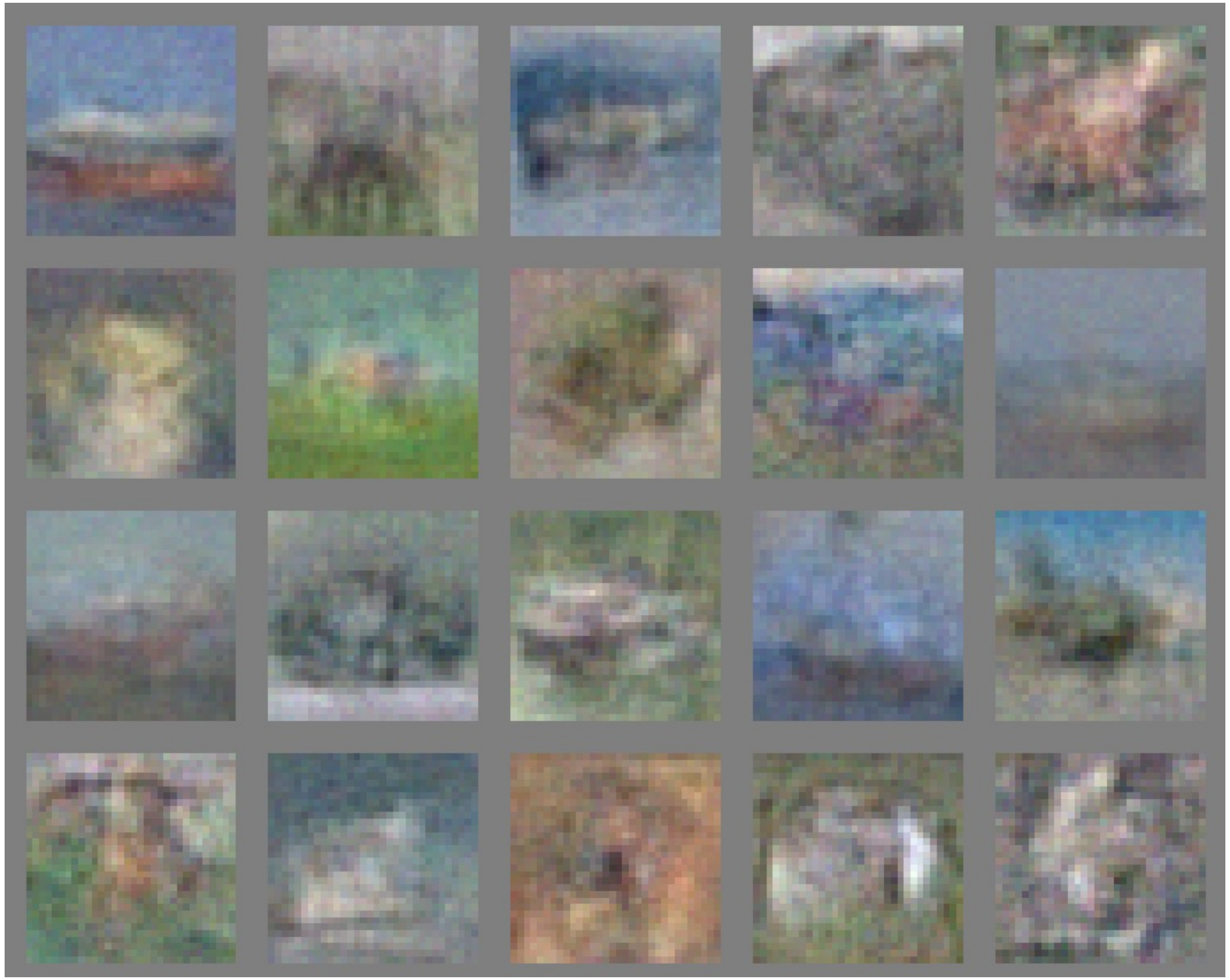
Introduced in 2014 by Ian Goodfellow



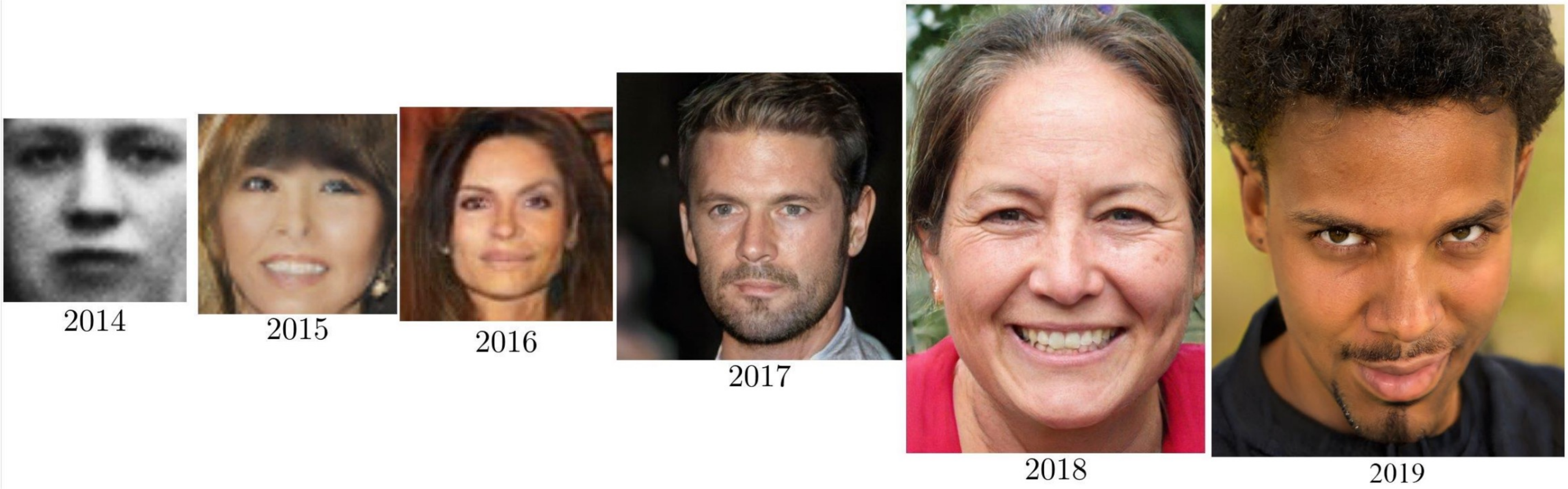
Generator learns a mapping from one probability distribution to another
(Commonly from a low dimensional Gaussian distribution to the distribution of images you train it on)

Generative Adversarial Networks (GANs)

Introduced in 2014 by Ian Goodfellow



GAN progress on face generation



Source: https://twitter.com/goodfellow_ian/status/1084973596236144640/
+StyleGANv2

Examples

BigGAN [2018]

Which face is real?

- Cool website
  - http://www.whichfaceisreal.com

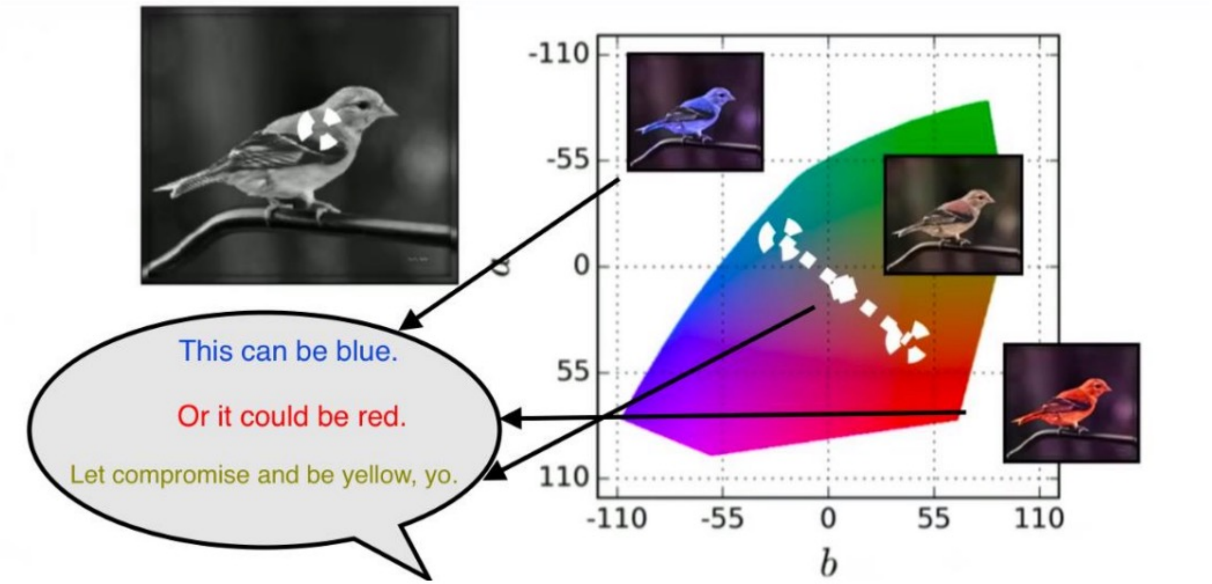
- Others
  - https://thisrentaldoesnotexist.com/
  - https://thiscatdoesnotexist.com/
  - https://thishorsedoesnotexist.com/
  - https://www.thiswaifudoesnotexist.net/

Why?

- Data without labels is abundant
- Being able to learn the distribution of your data is useful
- Many applications

Why do we need a discriminator?

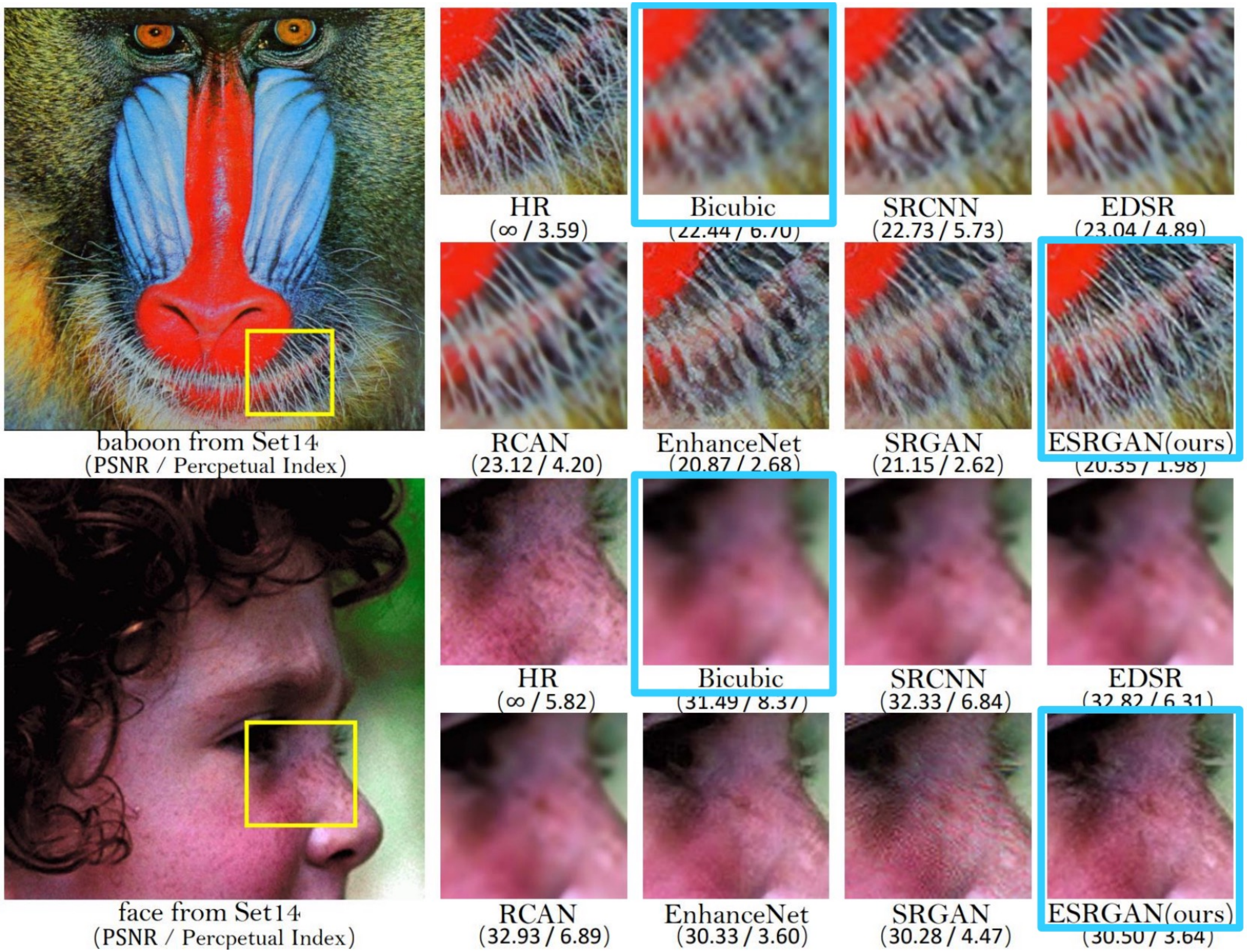
- The network should output an image
- Use a traditional loss?
  - L2 loss (mean squared error) gives blurry images
  - L1 loss (mean absolute error) gives sharper images
  - Both are very sensitive to pixel changes that don't mean anything perceptually



Applications

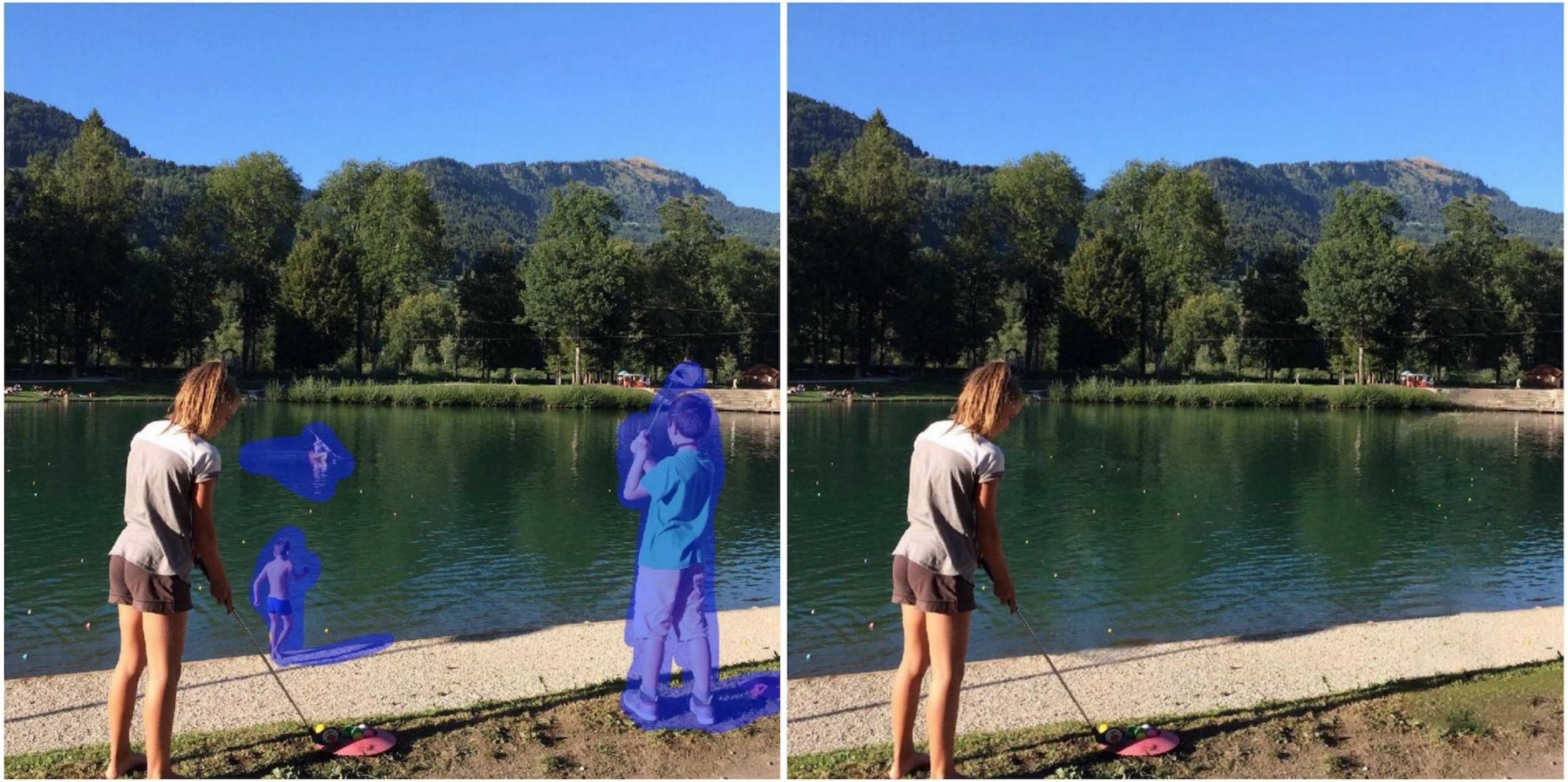
- Super-resolution
- Colorization
- In-painting
- Domain transfer
- Generating additional training data

Super resolution example



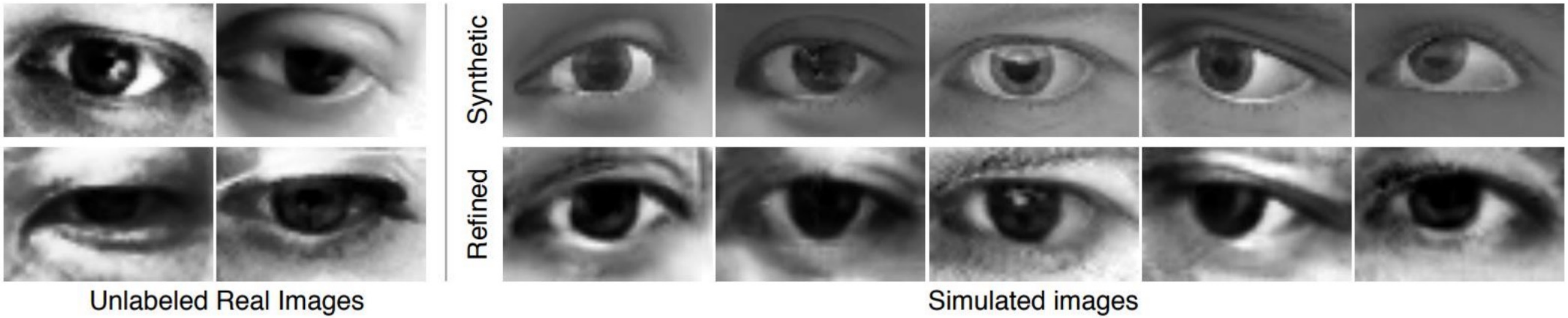
Inpainting

- Zeng, Yu, et al. "High-resolution image inpainting with iterative confidence feedback and guided upsampling." *European Conference on Computer Vision*. Springer, Cham, 2020.



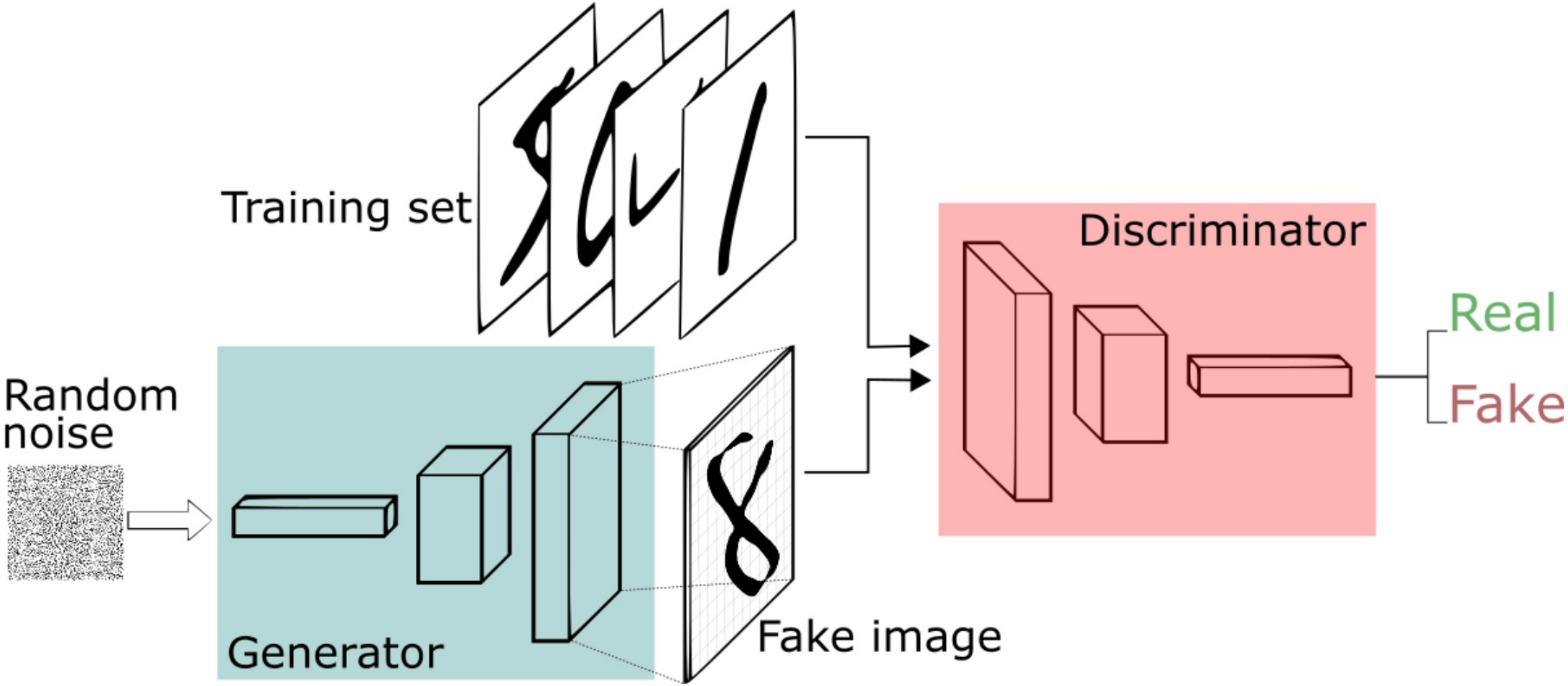
Generating more training data

- Making rendered images look like real images
  - But because they are rendered, we have ground truth labels



[2016] Ashish Shrivastava et al. Learning from Simulated and Unsupervised Images through Adversarial Training
https://arxiv.org/abs/1612.07828

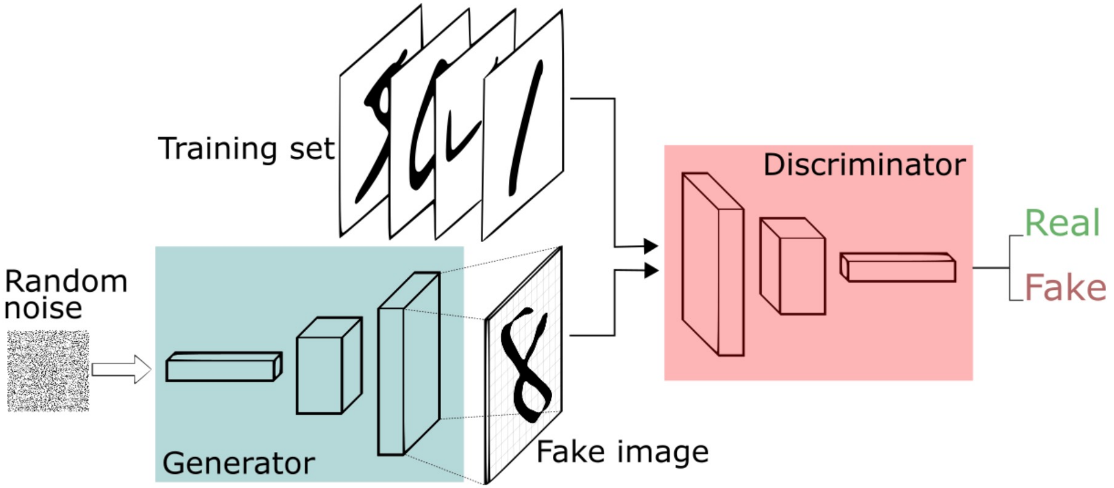
GANs



Training set
Discriminator
Real
Fake
Random noise
Generator
Fake image

How?

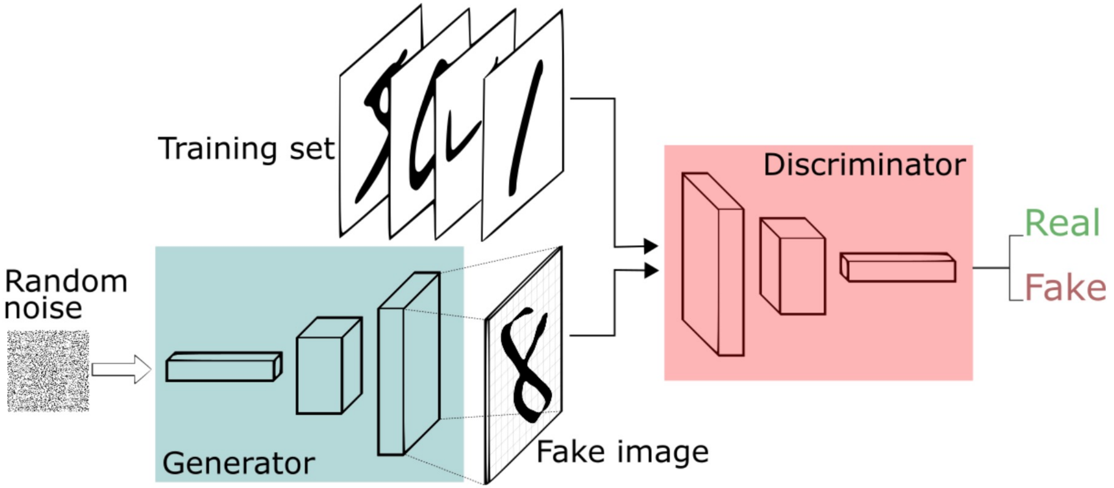
- Fully connected
- Many different losses possible
- Train generator and discriminator in an alternating fashion
  - Train discriminator for k iterations (can be k=2) (or k=1 and higher LR for D)
  - Then train generator once
  - Repeat
- Adam $\beta_1 = 0.5$, learning rate = 0.0002
  - Default parameter of $\beta_1 = 0.9$ doesn't work well (sometimes)
- Shorthand:
  - G: Generator
  - D: Discriminator



GAN training

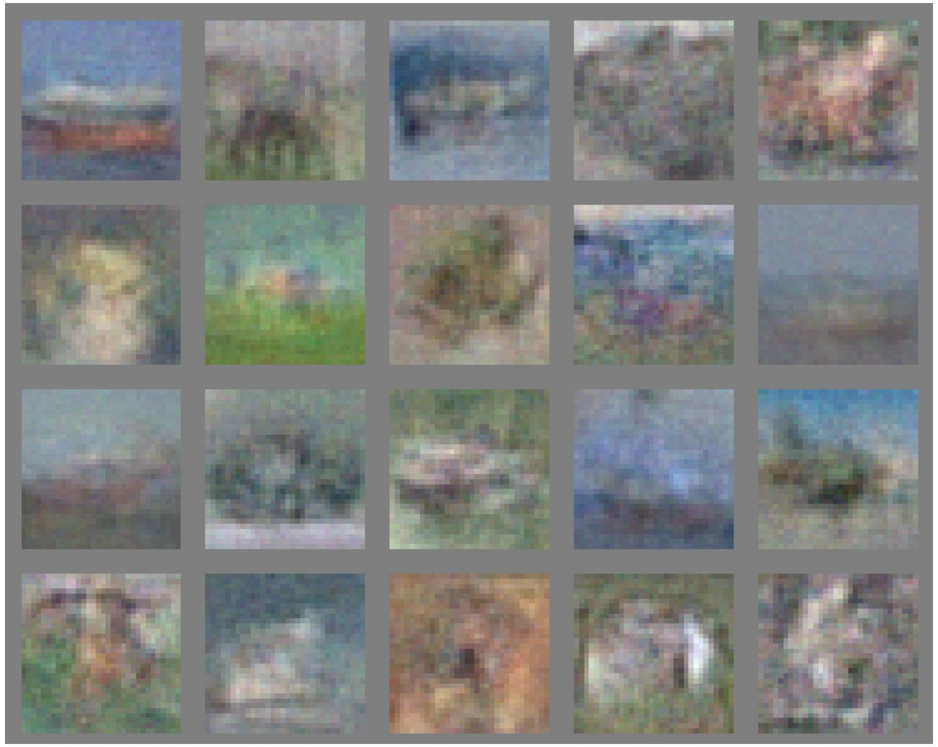
- Training GANs is still very hard
  - Many problems exist
  - Non-convergence
    - The models never converge and worse they become unstable.
  - Mode collapse
    - The generator produces a single or limited modes.
      - i.e. the images are not as diverse as the true data.

- Many tricks exist



Original GAN – Goodfellow 2014

- Original GAN paper
- Uses only fully connected layers
  - Limited to generating small images
- Discriminator
  - Binary cross entropy loss



DCGAN, 2015

- How to do GANs with convolutional layers?
  - Replace any pooling layers with strided convolutions (discriminator) and transposed-strided convolutions (generator)
  - Use batchnorm in both the generator and the discriminator
  - Use LeakyReLU activation in the discriminator for all layers
  - Use ReLU activation in generator for all layers except for the output, which uses Tanh
    - Later people recommend using LeakyReLU in both G and D

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." (2015).

DCGAN example architecture

- [filter h, filter w, stride]

(a) SNDCGAN discriminator

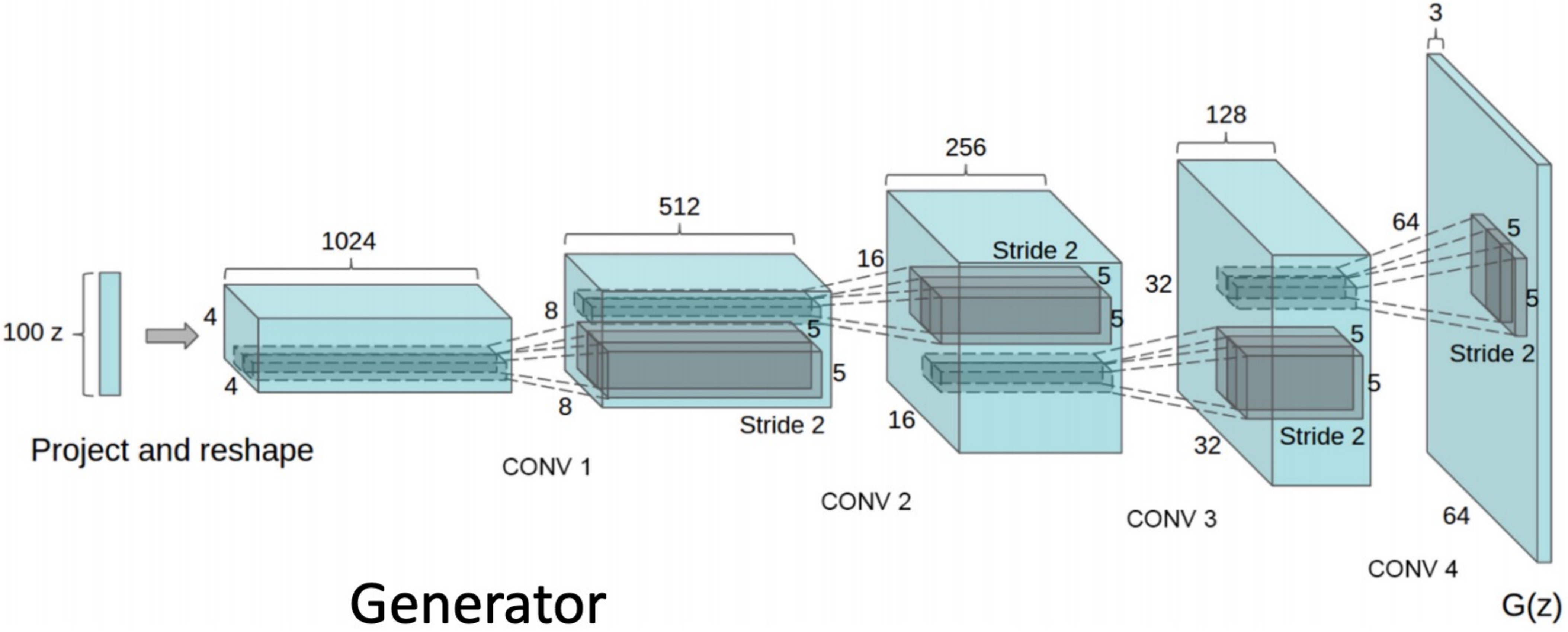
LAYER	KERNEL	OUTPUT
Conv, lReLU	$[3, 3, 1]$	$h \times w \times 64$
Conv, lReLU	$[4, 4, 2]$	$h/2 \times w/2 \times 128$
Conv, lReLU	$[3, 3, 1]$	$h/2 \times w/2 \times 128$
Conv, lReLU	$[4, 4, 2]$	$h/4 \times w/4 \times 256$
Conv, lReLU	$[3, 3, 1]$	$h/4 \times w/4 \times 256$
Conv, lReLU	$[4, 4, 2]$	$h/8 \times w/8 \times 512$
Conv, lReLU	$[3, 3, 1]$	$h/8 \times w/8 \times 512$
Linear	-	1

(b) SNDCGAN generator

LAYER	KERNEL	OUTPUT
z	-	128
Linear, BN, ReLU	-	$h/8 \times w/8 \times 512$
Deconv, BN, ReLU	$[4, 4, 2]$	$h/4 \times w/4 \times 256$
Deconv, BN, ReLU	$[4, 4, 2]$	$h/2 \times w/2 \times 128$
Deconv, BN, ReLU	$[4, 4, 2]$	$h \times w \times 64$
Deconv, Tanh	$[3, 3, 1]$	$h \times w \times 3$

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." (2015).

DCGAN example architecture



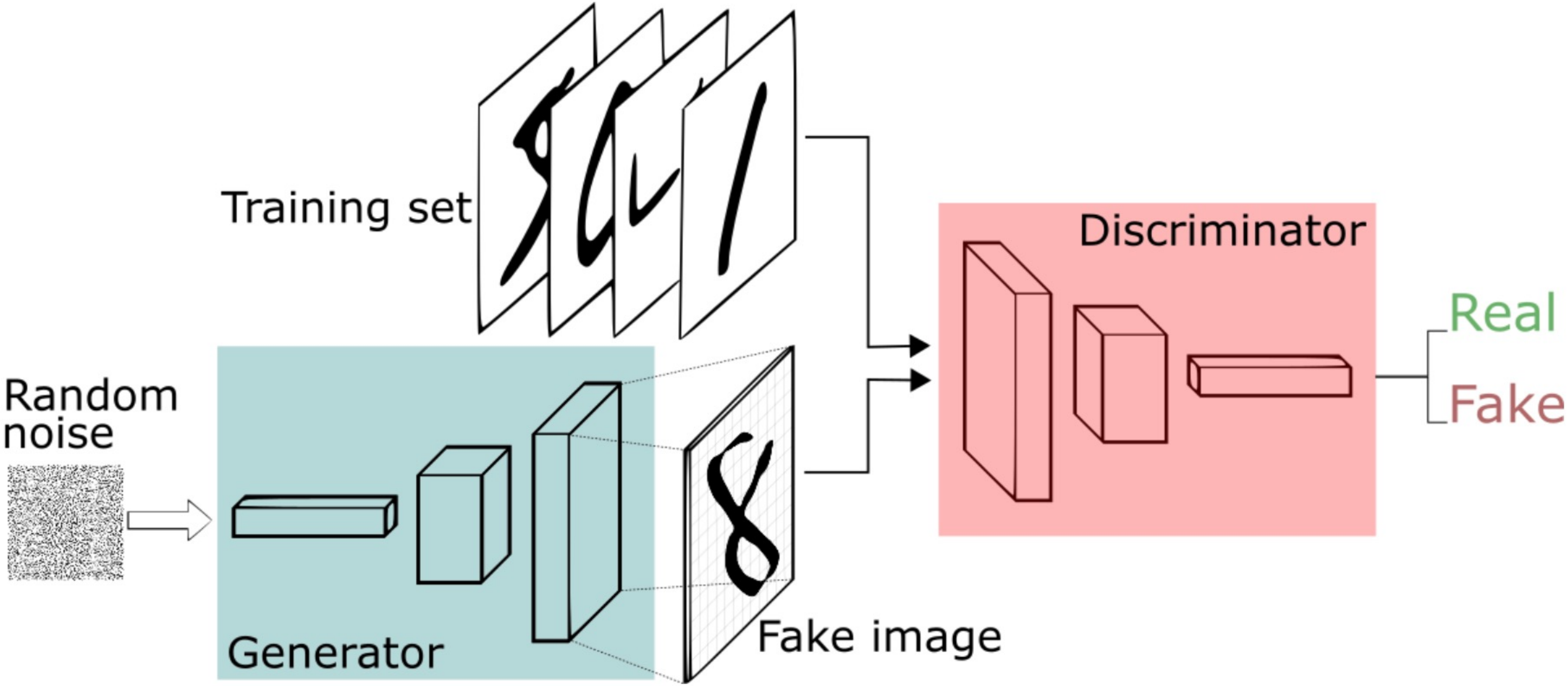
Radford et al, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", ICLR 2016

DC-GAN

Samples from the model look much better!

Radford et al, ICLR 2016

GAN training and loss



minimax GAN (vanilla GAN)

- GANs are a two player game
  - Which game do they play? Minimax!
  - G tries to minimize
  - D tries to maximize
- Loss:

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

For G

Rather than training G to minimize $\log(1 - D(G(z)))$ we can train G to maximize $\log D(G(z))$.

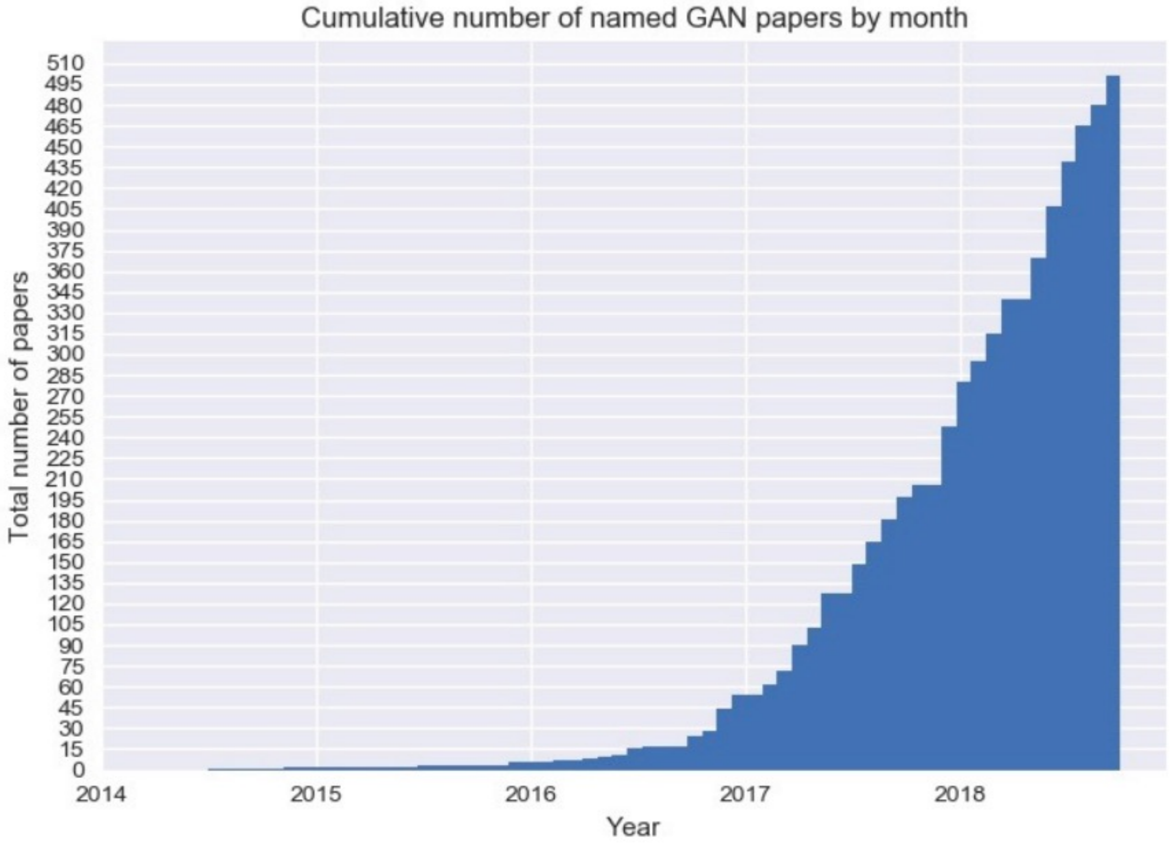
Minimizes the Jensen-Shannon divergence between p_d and p_g.

minimax GAN

- How to translate to code?
  - Expectation can be intractable, and requires solving complicated integrals
- Estimate the expectations as the mean of the current minibatch
  - The current minibatch is a sample from the full distributions
  - An average is an estimate of the expected value

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

2017 to present: Explosion of GANs



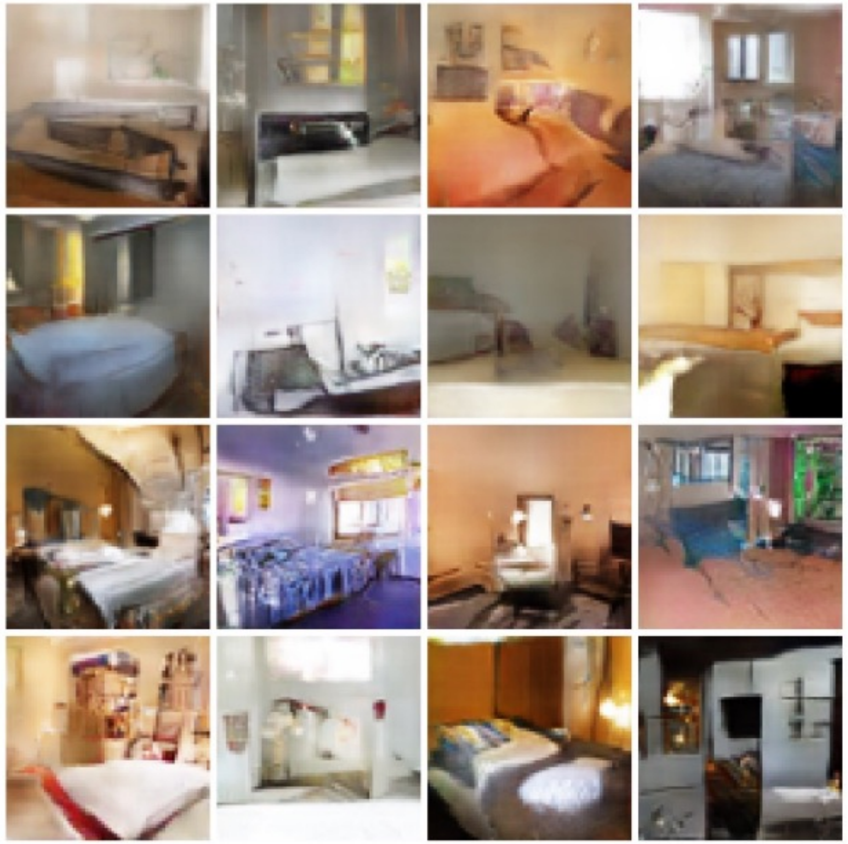
https://github.com/hindupuravinash/the-gan-zoo

- 3D-ED-GAN - Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (github)
- 3D-IWGAN - Improved Adversarial Systems for 3D Object Generation and Reconstruction (github)
- 3D-PhysNet - 3D-PhysNet: Learning the Intuitive Physics of Non-Rigid Object Deformations
- 3D-RecGAN - 3D Object Reconstruction from a Single Depth View with Adversarial Learning (github)
- ABC-GAN - ABC-GAN: Adaptive Blur and Control for improved training stability of Generative Adversarial Networks (github)
- ABC-GAN - GANs for LIFE: Generative Adversarial Networks for Likelihood Free Inference
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- ACGAN - Coverless Information Hiding Based on Generative adversarial networks
- acGAN - On-line Adaptative Curriculum Learning for GANs
- ACtuAL - ACtuAL: Actor-Critic Under Adversarial Learning
- AdaGAN - AdaGAN: Boosting Generative Models
- Adaptive GAN - Customizing an Adversarial Example Generator with Class-Conditional GANs
- AdvEntuRe - AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples
- AdvGAN - Generating adversarial examples with adversarial networks
- AE-GAN - AE-GAN: adversarial eliminating with GAN
- AE-OT - Latent Space Optimal Transport for Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AF-DCGAN - AF-DCGAN: Amplitude Feature Deep Convolutional GAN for Fingerprint Construction in Indoor Localization System
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AIM - Generating Informative and Diverse Conversational Responses via Adversarial Information Maximization
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference (github)
- AlignGAN - AlignGAN: Learning to Align Cross-Domain Images with Conditional Generative Adversarial Networks
- AlphaGAN - AlphaGAN: Generative adversarial networks for natural image matting
- AM-GAN - Activation Maximization Generative Adversarial Nets
- AmbientGAN - AmbientGAN: Generative models from lossy measurements (github)
- AMC-GAN - Video Prediction with Appearance and Motion Conditions
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- APD - Adversarial Distillation of Bayesian Neural Network Posteriors
- APE-GAN - APE-GAN: Adversarial Perturbation Elimination with GAN
- ARAE - Adversarially Regularized Autoencoders for Generating Discrete Structures (github)
- ARDA - Adversarial Representation Learning for Domain Adaptation
- ARIGAN - ARIGAN: Synthetic Arabidopsis Plants using Generative Adversarial Network
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- ASDL-GAN - Automatic Steganographic Distortion Learning Using a Generative Adversarial Network
- ATA-GAN - Attention-Aware Generative Adversarial Networks (ATA-GANs)
- Attention-GAN - Attention-GAN for Object Transfiguration in Wild Images
- AttGAN - Arbitrary Facial Attribute Editing: Only Change What You Want (github)
- AttnGAN - AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks (github)
- AVID - AVID: Adversarial Visual Irregularity Detection
- B-DCGAN - B-DCGAN:Evaluation of Binarized DCGAN for FPGA
- b-GAN - Generative Adversarial Nets from a Density Ratio Estimation Perspective
- BAGAN - BAGAN: Data Augmentation with Balancing GAN
- Bayesian GAN - Deep and Hierarchical Implicit Models
- Bayesian GAN - Bayesian GAN (github)
- BCGAN - Bayesian Conditional Generative Adverserial Networks
- BCGAN - Bidirectional Conditional Generative Adversarial networks
- BEAM - Boltzmann Encoded Adversarial Machines
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BEGAN-CS - Escaping from Collapsing Modes in a Constrained Space
- Bellman GAN - Distributional Multivariate Policy Evaluation and Exploration with the Bellman

- BGAN - Binary Generative Adversarial Networks for Image Retrieval (github)
- Bi-GAN - Autonomously and Simultaneously Refining Deep Neural Network Parameters by a Bi-Generative Adversarial Network Aided Genetic Algorithm
- BicycleGAN - Toward Multimodal Image-to-Image Translation (github)
- BiGAN - Adversarial Feature Learning
- BinGAN - BinGAN: Learning Compact Binary Descriptors with a Regularized GAN
- BourGAN - BourGAN: Generative Networks with Metric Embeddings
- BranchGAN - Branched Generative Adversarial Networks for Multi-Scale Image Manifold Learning
- BRE - Improving GAN Training via Binarized Representation Entropy (BRE) Regularization (github)
- BridgeGAN - Generative Adversarial Frontal View to Bird View Synthesis
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- BubGAN - BubGAN: Bubble Generative Adversarial Networks for Synthesizing Realistic Bubbly Flow Images
- BWGAN - Banach Wasserstein GAN
- C-GAN - Face Aging with Contextual Generative Adversarial Nets
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training (github)
- CA-GAN - Composition-aided Sketch-realistic Portrait Generation
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks (github)
- CAN - CAN: Creative Adversarial Networks, Generating Art by Learning About Styles and Deviating from Style Norms
- CapsGAN - CapsGAN: Using Dynamic Routing for Generative Adversarial Networks
- CapsuleGAN - CapsuleGAN: Generative Adversarial Capsule Network
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CatGAN - CatGAN: Coupled Adversarial Transfer for Domain Generation
- CausalGAN - CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training
- CC-GAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks (github)
- cd-GAN - Conditional Image-to-Image Translation
- CDcGAN - Simultaneously Color-Depth Super-Resolution with Conditional Generative Adversarial Network
- CE-GAN - Deep Learning for Imbalance Data Classification using Class Expert Generative Adversarial Network
- CFG-GAN - Composite Functional Gradient Learning of Generative Adversarial Models
- CGAN - Conditional Generative Adversarial Nets
- CGAN - Controllable Generative Adversarial Network
- Chekhov GAN - An Online Learning Approach to Generative Adversarial Networks
- ciGAN - Conditional Infilling GANs for Data Augmentation in Mammogram Classification
- CinCGAN - Unsupervised Image Super-Resolution using Cycle-in-Cycle Generative Adversarial Networks
- CipherGAN - Unsupervised Cipher Cracking Using Discrete GANs
- ClusterGAN - ClusterGAN : Latent Space Clustering in Generative Adversarial Networks
- CM-GAN - CM-GANs: Cross-modal Generative Adversarial Networks for Common Representation Learning
- CoAtt-GAN - Are You Talking to Me? Reasoned Visual Dialog Generation through Adversarial Learning
- CoGAN - Coupled Generative Adversarial Networks
- ComboGAN - ComboGAN: Unrestrained Scalability for Image Domain Translation (github)
- ConceptGAN - Learning Compositional Visual Concepts with Mutual Consistency
- Conditional cycleGAN - Conditional CycleGAN for Attribute Guided Face Image Generation
- constrast-GAN - Generative Semantic Manipulation with Contrasting GAN
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- CorrGAN - Correlated discrete data generation using adversarial training
- Coulomb GAN - Coulomb GANs: Provably Optimal Nash Equilibria via Potential Fields
- Cover-GAN - Generative Steganography with Kerckhoffs' Principle based on Generative Adversarial Networks
- cowboy - Defending Against Adversarial Attacks by Leveraging an Entire GAN
- CR-GAN - CR-GAN: Learning Complete Representations for Multi-view Generation
- Cramèr GAN - The Cramer Distance as a Solution to Biased Wasserstein Gradients
- Cross-GAN - Crossing Generative Adversarial Networks for Cross-View Person Re-identification
- crVAE-GAN - Channel-Recurrent Variational Autoencoders
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CSG - Speech-Driven Expressive Talking Lips with Conditional Sequential Generative Adversarial Networks
- CT-GAN - CT-GAN: Conditional Transformation Generative Adversarial Network for Image Attribute Modification
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

GAN Improvements: Improved Loss Functions

Wasserstein GAN (WGAN)



Arjovsky, Chintala, and Bouttou, “Wasserstein GAN”, 2017

WGAN with Gradient Penalty (WGAN-GP)

Gulrajani et al, “Improved Training of Wasserstein GANs”, NeurIPS 2017

WGAN

Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." *arXiv preprint arXiv:1701.07875* (2017).

- Wasserstein GAN
  - D only outputs a number
  - Discriminator is now a "critic" (i.e. cannot classify real or fake)
  - Optimize approximation of Wasserstein-1 distance
    - Tries to push the means of the two distributions as far apart as possible

$$\max_{w \in \mathcal{W}} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z))]$$

  - Discriminator must be Lipschitz continuous
    - Otherwise it can push them arbitrarily far apart without becoming more discriminative
  - Introduce weight clipping in discriminator to enforce Lipschitz continuous
    - "Weight clipping is a clearly terrible way to enforce a Lipschitz constraint"

      -quote from the paper

LSGAN

Mao et al. *Least Squares Generative Adversarial Networks*

- Uses a least square loss instead
- Simple to implement

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[(D(\boldsymbol{x}) - b)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z})) - a)^2\right]$$

$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z})) - c)^2\right],$$

$a = -1, b = 1$ and $c = 0$ minimizes Pearson χ^2 distance between $p_d + p_g$ and $2p_g$.
$a = 0, b = 1$ and $c = 1$ is also a good choice.

Important note

- You should have no activation function on the last layer of your discriminator when using WGAN or LSGAN
  - The discriminator should be able to output any value
- But you also shouldn’t have it on your vanilla GAN
  - As you should use LogSigmoid to avoid numerical problems

Data augmentation

Zhao Zhengli et al. "Image Augmentations for GAN Training." *arXiv preprint arXiv: 2006.02595* (2020).

- Data augmentation is helpful during GAN training, if done to real AND fake images
  - Need differentiable augmentations
  - https://github.com/mit-han-lab/data-efficient-gans

Original Image ZoomOut ZoomIn Translation TranslationX TranslationY

FID - Frechet Inception Distance

Heusel, Martin, et al. "Gans trained by a two time-scale update rule converge to a local nash equilibrium." *Advances in neural information processing systems*. 2017.

- A metric attempting to quantitatively evaluate the quality of generated images
- Idea
  - Use features learned for classification, and see how similar they are for real and generated images
- How to:
  - Take the 2048 dimensional output of the global pooling layer in Inception v3
  - Compute the mean and covariance matrix of the features for real and fake images
  - Calculate the Frechet distance between the real and generated images

$$\|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\big(\boldsymbol{C} + \boldsymbol{C}_w - 2\big(\boldsymbol{C}\boldsymbol{C}_w\big)^{1/2}\big)$$

GAN Improvements: Higher Resolution

512 x 384 cars

1024 x 1024 faces

Karras et al, "A Style-Based Generator Architecture for Generative Adversarial Networks", CVPR 2019

Images are licensed under CC BY-NC 4.0

Conditional GANs

Recall: Conditional Generative Models learn p(x|y) instead of p(x)
Make generator and discriminator both take label y as an additional input!

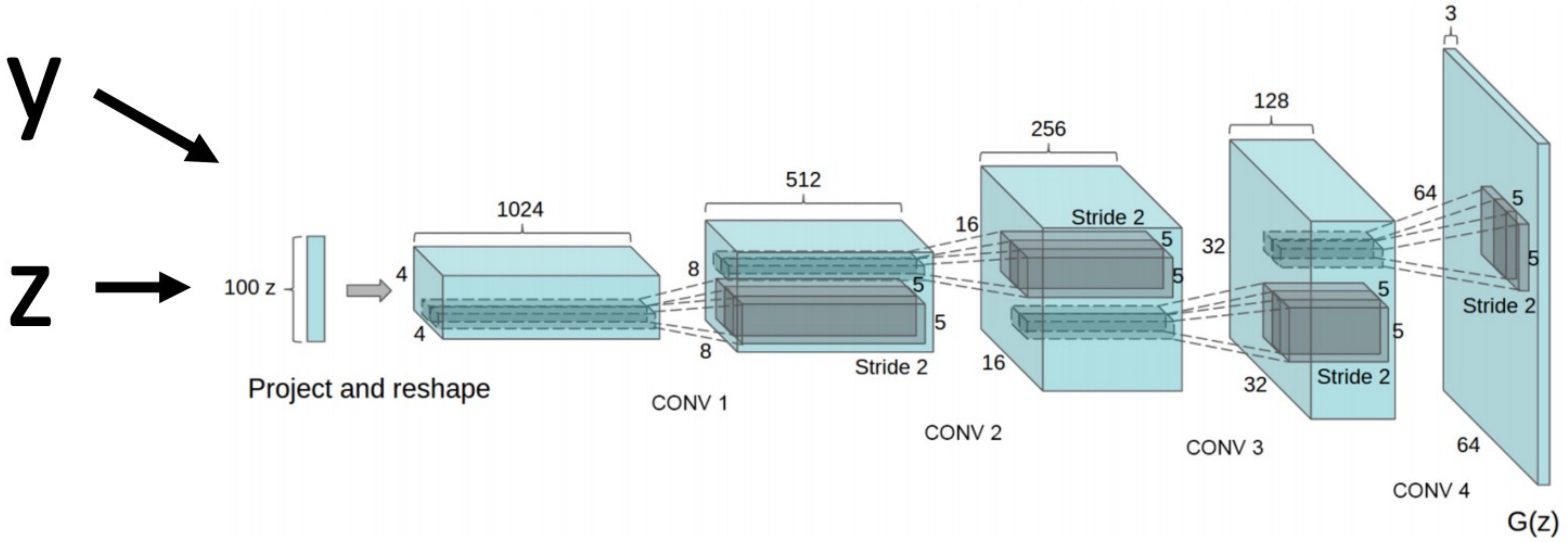


Figure credit: Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

Conditional GANs: Spectral Normalization

Welsh springer spaniel

Fire truck

Daisy

Miyato et al, “Spectral Normalization for Generative Adversarial Networks”, ICLR 2018

128x128 images on ImageNet

Conditional GANs: BigGAN

Brock et al, "Large Scale GAN Training for High Fidelity Natural Image Synthesis", ICLR 2019

512x512 images on ImageNet

Conditioning on more than labels! Text to Image

This bird is red and brown in color, with a stubby beak

The bird is short and stubby with yellow on its body

A bird with a medium orange bill white body gray wings and webbed feet

This small black bird has a short, slightly curved bill and long legs

A picture of a very clean living room

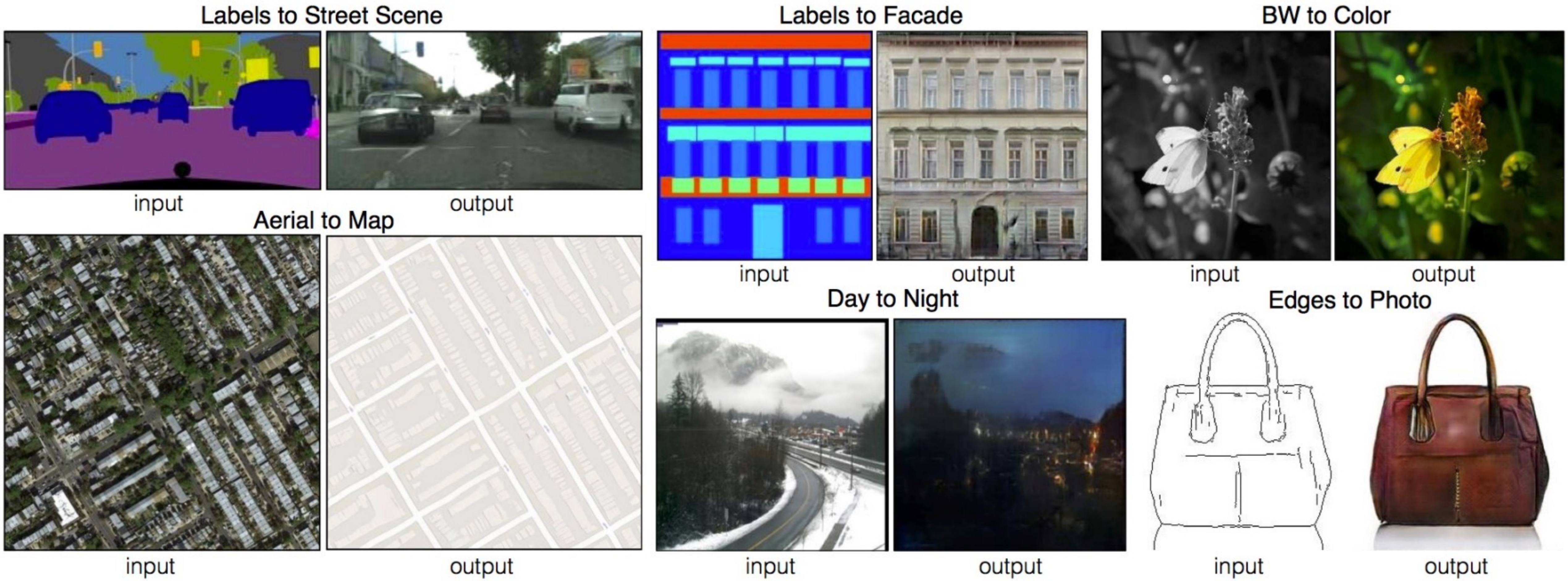
A group of people on skis stand in the snow

Eggs fruit candy nuts and meat served on white dish

A street sign on a stoplight pole in the middle of a day

Zhang et al, “StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks.”, TPAMI 2018
Zhang et al, “StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks.”, ICCV 2017
Reed et al, “Generative Adversarial Text-to-Image Synthesis”, ICML 2016

Image-to-Image Translation: Pix2Pix



Isola et al, "Image-to-Image Translation with Conditional Adversarial Nets", CVPR 2017

Unpaired Image-to-Image Translation: CycleGAN

Input Video: Horse

Output Video: Zebra

https://www.youtube.com/watch?v=9reHvktowLY

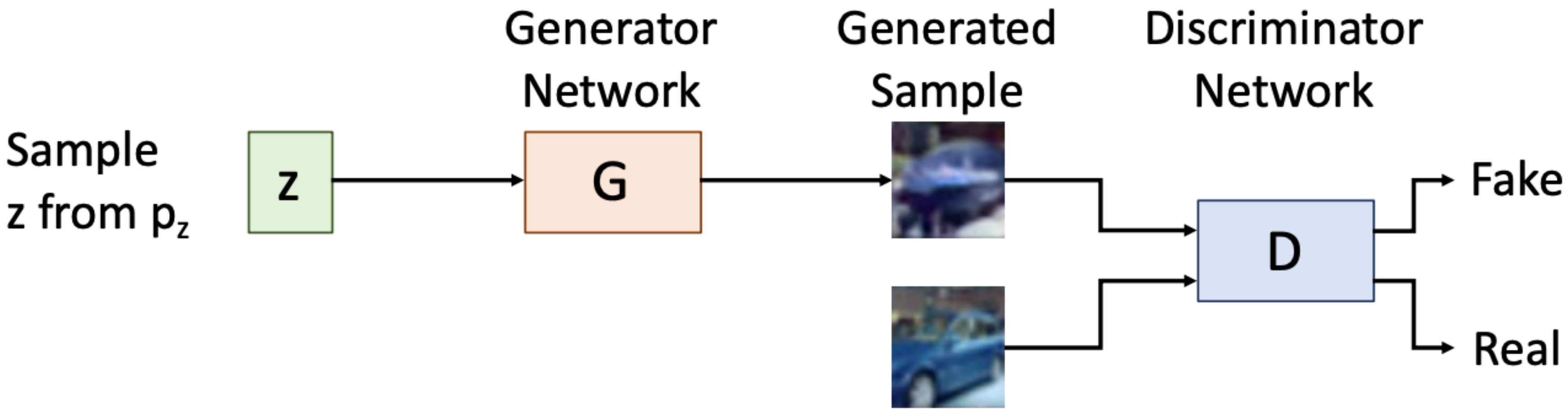
Zhu et al, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", ICCV 2017

GAN Summary

Jointly train two networks:

Discriminator: Classify data as real or fake

Generator: Generate data that fools the discriminator



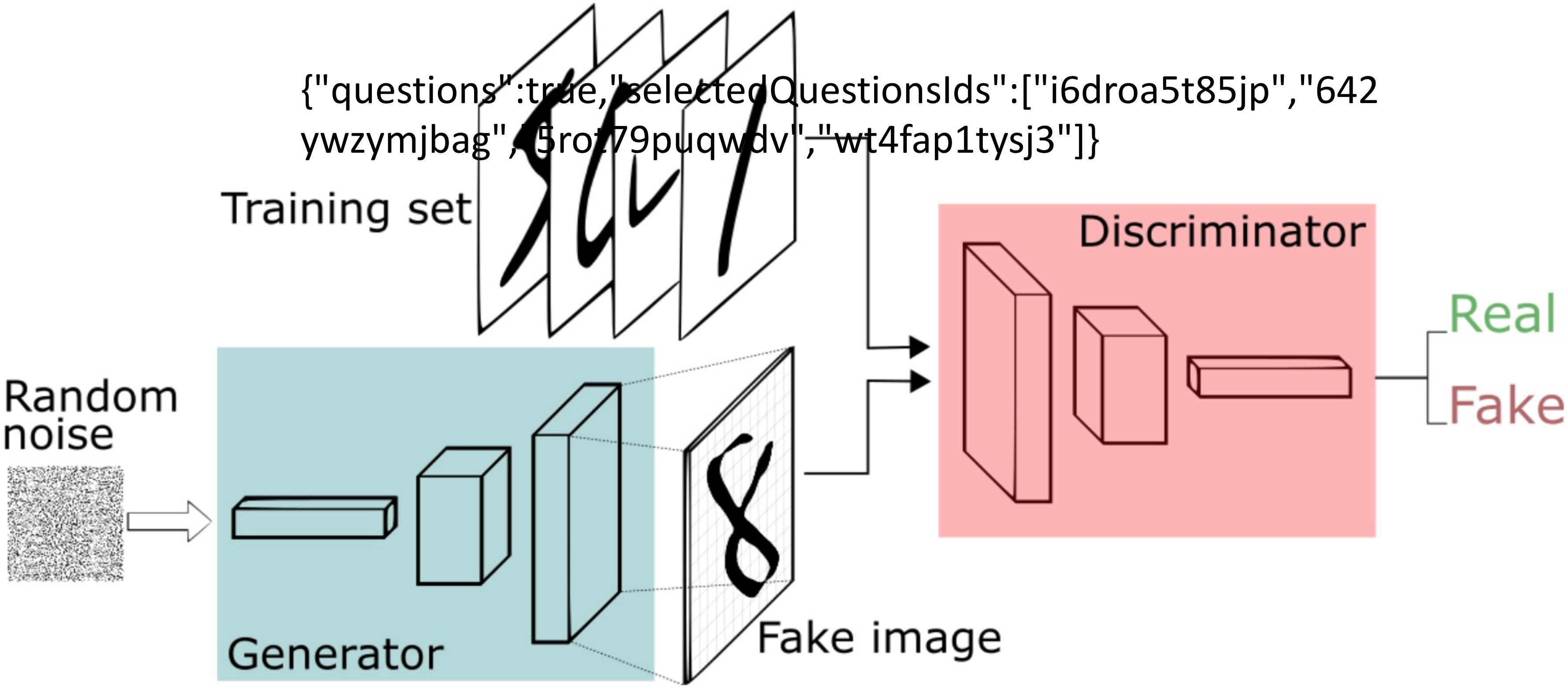
Under some assumptions, generator converges to true data distribution

Many applications! Very active area of research!

Exercise 3

Generate MNIST digits using a vanilla GAN

{"questions":true,"selectedQuestionsIds":["i6droa5t85jp","642ywzymjbag","5rot79puqwdv","wt4fap1tysj3"]}



Feedback

Join at **menti.com** use code **3886 2376**

- First half of the course?
- Today's lecture?