Generative Adversarial Networks

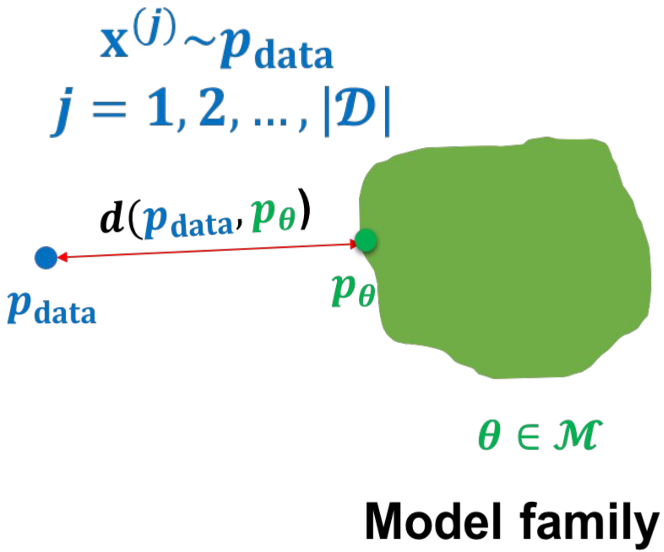
Harvard | Spring 2022 | Anna Trella
Original Material Provided by Bill Zhang

The Task

Given a dataset and an underlying distribution, approximate using a parameterized distribution

Key applications:

- **Density estimation**: What is the probability of a given data point occurring, i.e. **p(x)**?
- **Sampling**: How can we generate new data points from the distribution, i.e. **x’ ~ p(x)**?
- **Representation learning**: How can we learn meaningful representations of the data?



Credit: Stanford CS236 (Introduction)

Likelihood-Free Learning

We do not train using maximum likelihood.

$$\cancel{\theta^* = \arg\max_\theta \log \mathcal{L}(\theta)}$$

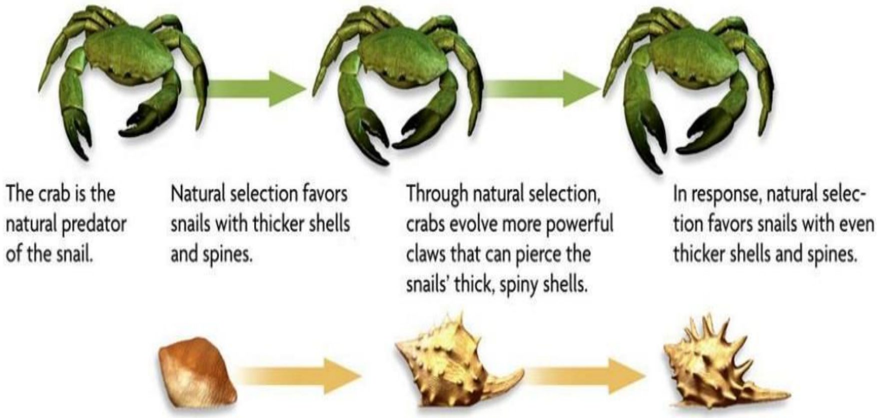
Why not? Doesn't the best generative model produces samples with high likelihood?

- High likelihood does not always correspond to good models (e.g. our model memorize the training set)

Credit: Stanford CS236 (Generative Adversarial Networks)

Likelihood-Free Learning

- We instead draw inspiration from nature



Credit: Stanford CS236 ([Generative Adversarial Networks])

Intuition: The Arms Race

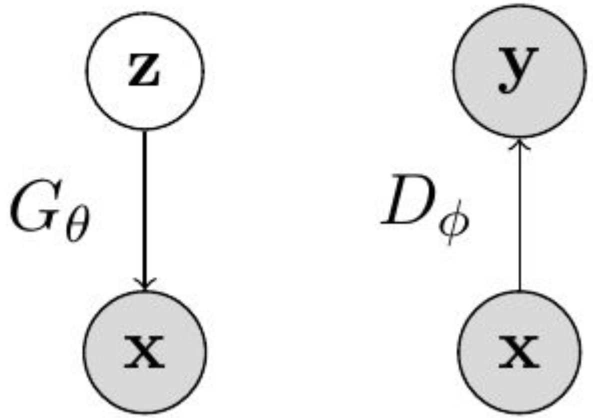
The two components of a GAN are the **generator** and **discriminator**

- The **generator** learns latent **z** from the data, then tries to produce fake data **x**
- The **discriminator** learns to tell apart the fake data **x** and real data

Both can be modeled with neural networks

This is a minimax game, and the hope is the generator will become very good at sampling

What are the pros and cons of GANs?



Credit: Stanford CS236 (Generative Adversarial Networks), Goodfellow et al. 2014

GAN Objective

- The **generator** minimizes a two-sample test objective $p_{data} = p_\theta$
- The **discriminator** maximizes the objective $p_{\text{data}} \neq p_\theta$

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

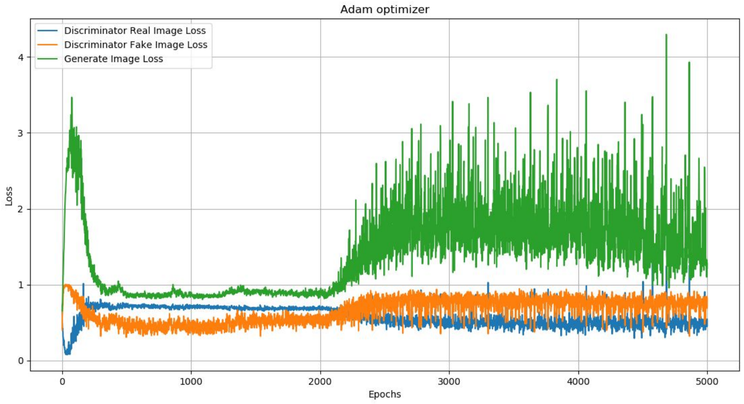
A good discriminator is correctly performing binary classification:
Given a fixed generate G, the discriminator assigns probability 1 to true data points and probability 0 to generated samples

Credit: Stanford CS236 (Generative Adversarial Networks), Goodfellow et al. 2014

Generative Adversarial Networks

Pros:

- Likelihood-free learning is more flexible
- Can perform very, very well empirically
- Lots of research being done



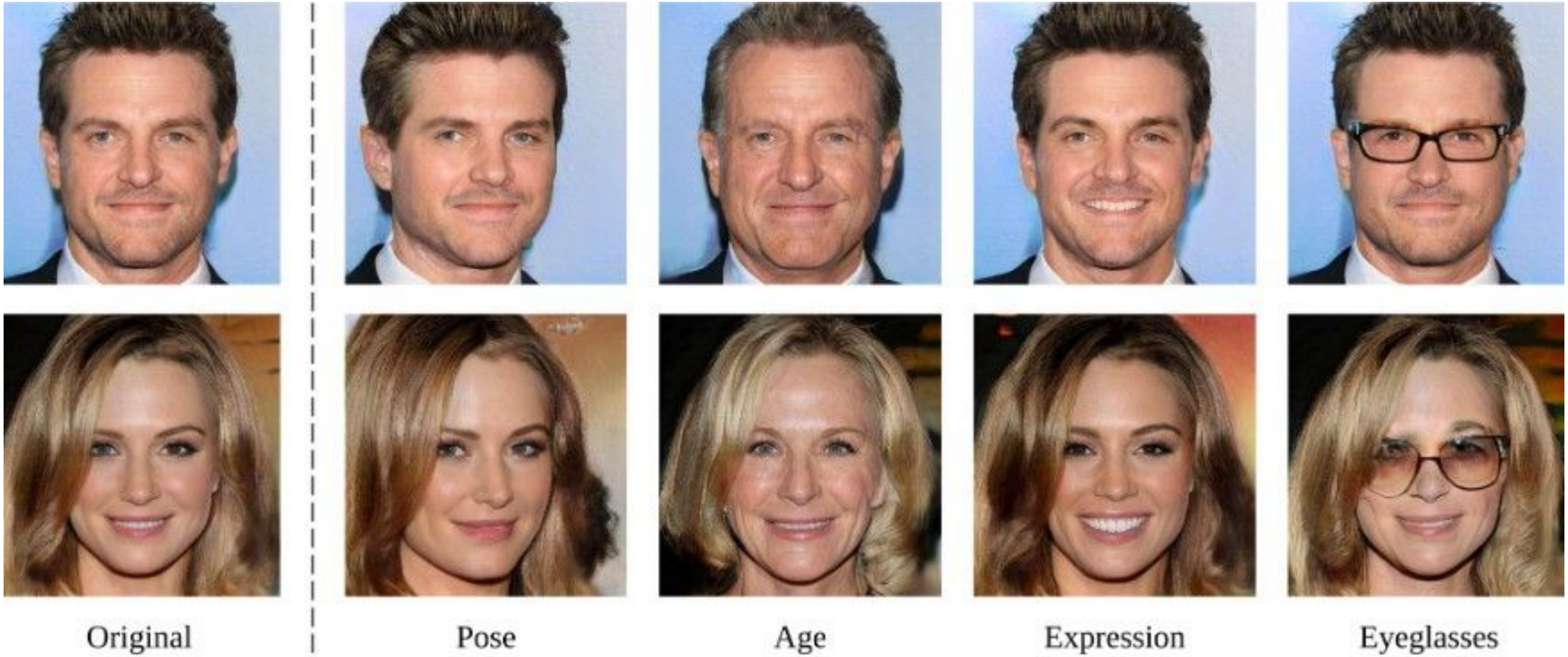
Cons:

- No theoretical guarantees
- The optimization can be unstable
- Potential for mode collapse (generator samples from narrow region)
- Many tricks required to improve performance

Credit: Stanford CS236 (Generative Adversarial Networks)

GAN Output Examples

For Faces



Generate Realistic Photos

Credit: Jason Brownlee [Article Link](#)
Example of Realistic Synthetic Photographs Generated with BigGANTaken from Large Scale GAN Training for High Fidelity Natural Image Synthesis, 2018.

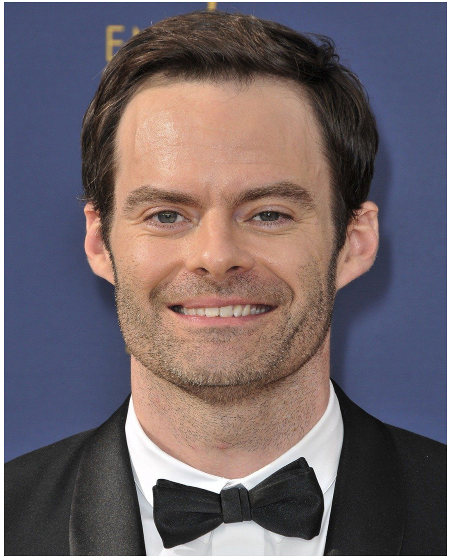
Generate Anime Characters

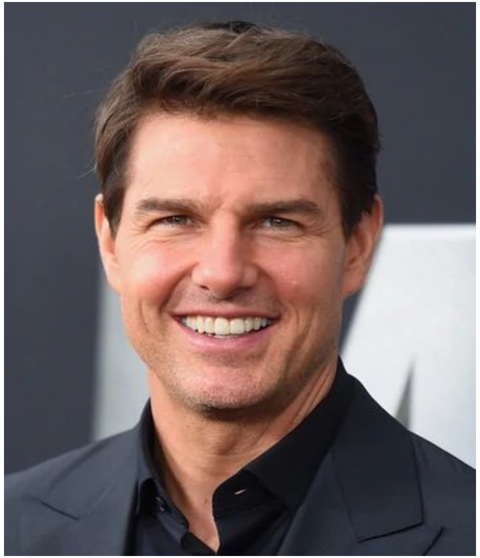
Credit: Jason Brownlee [Article Link]
Example of GAN-Generated Anime Character Faces.Taken from Towards the Automatic Anime Characters Creation with Generative Adversarial Networks, 2017.

DeepFakes

Fake videos which use GANs to imitate real people

- They can be very compelling!



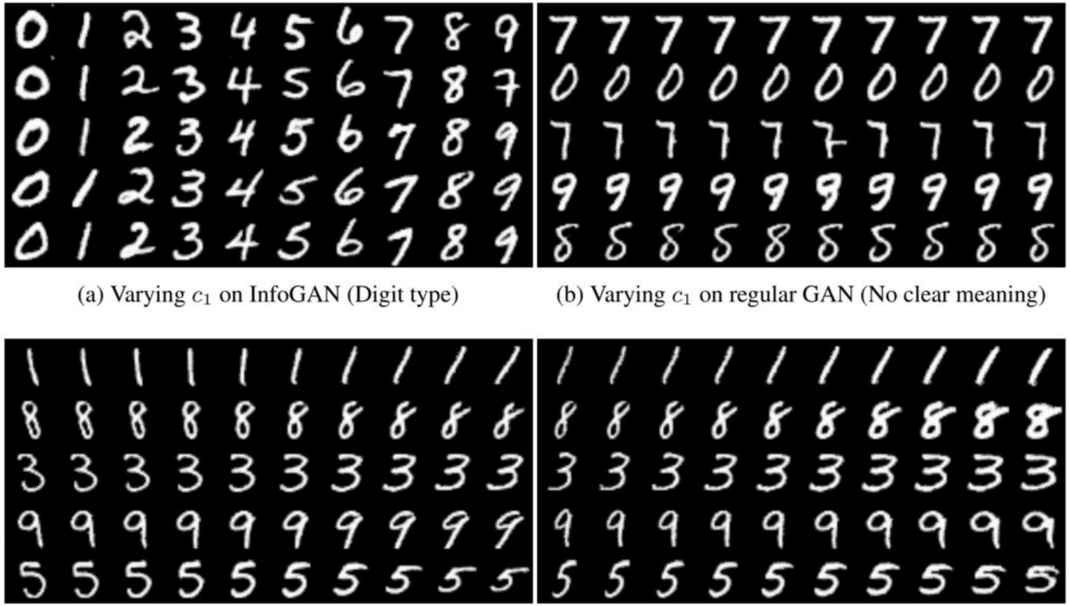


Modified GANs

InfoGAN

Structure the latent space of the generator so dimensions have semantic meaning

- Task: identify digits
- Two additional variables that represent angle and thickness of digit stroke



(a) Varying c_1 on InfoGAN (Digit type)

(b) Varying c_1 on regular GAN (No clear meaning)

(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Credit: [Chen et al. 2016]

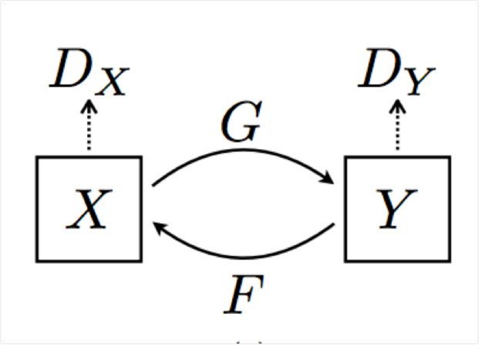
CycleGAN

An extension of the GAN that allows for translation across domains X and Y

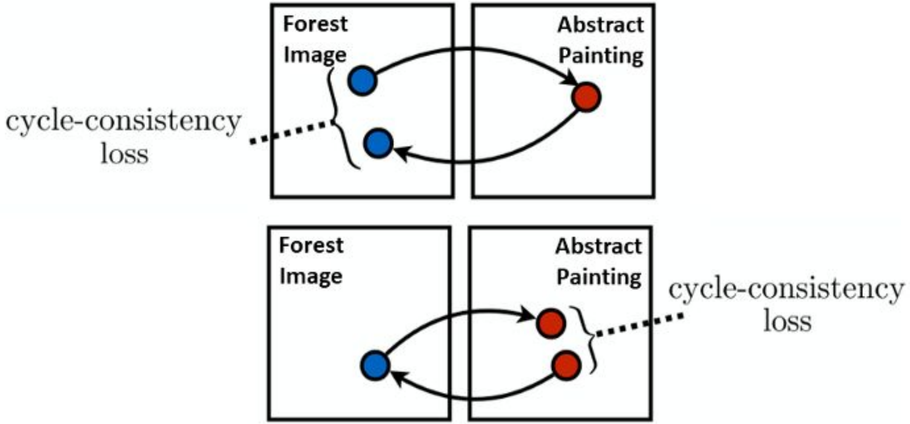
- Two generators G: X → Y, F: Y → X
- Two discriminators D_X and D_Y
- D_X compares F(Y) and true data X
- D_Y compares F(G) and true data Y

Loss is informed by **cycle consistency**

- F(G(X)) = X, reconstruct the original



Source: Zhu et al., 2016



Credit: Stanford CS236 (Generative Adversarial Networks), Zhu et al. 2016, Zack Monge (Towards Data Science)

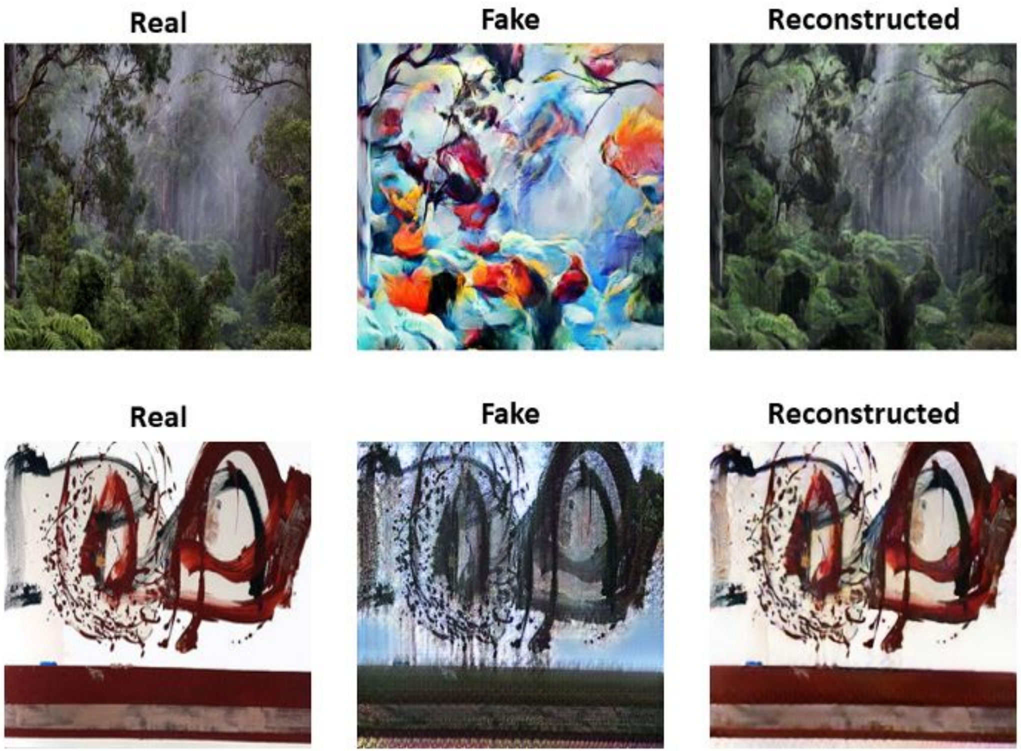
CycleGAN Applications

- X = forest photographs
- Y = abstract paintings

Goal is to train good:

- G: X → Y, photos to artwork
- F: Y → X, artwork to photos

Notice the structure of the image remains consistent while the style changes

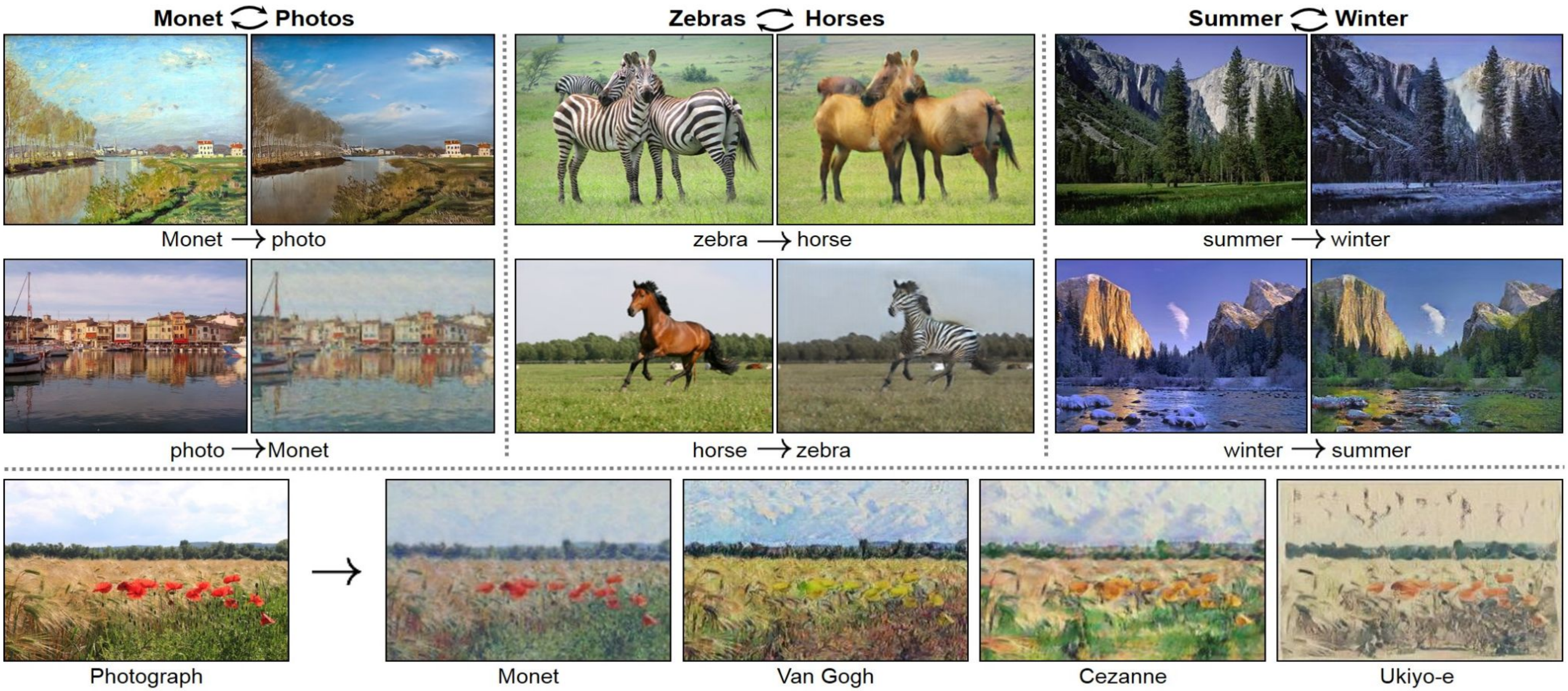


Credit: Zack Monge ([Towards Data Science])

Finished Product

Credit: Zack Monge (Towards Data Science)

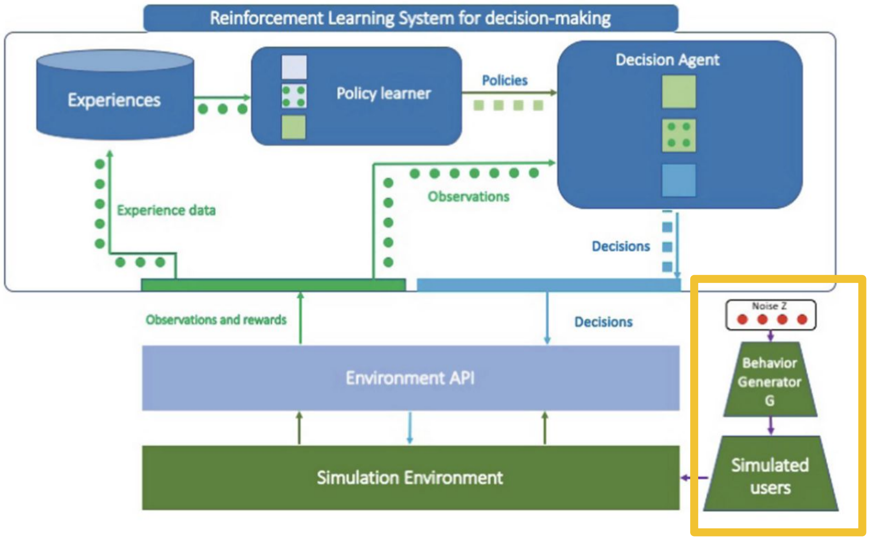
Image Translation



Credit: [Zhu et al. 2016]()

Data Privacy

- Idea: If we can generate data close to the original dataset, we can provide the generated data set for auditing or reproducibility and ensures user privacy.

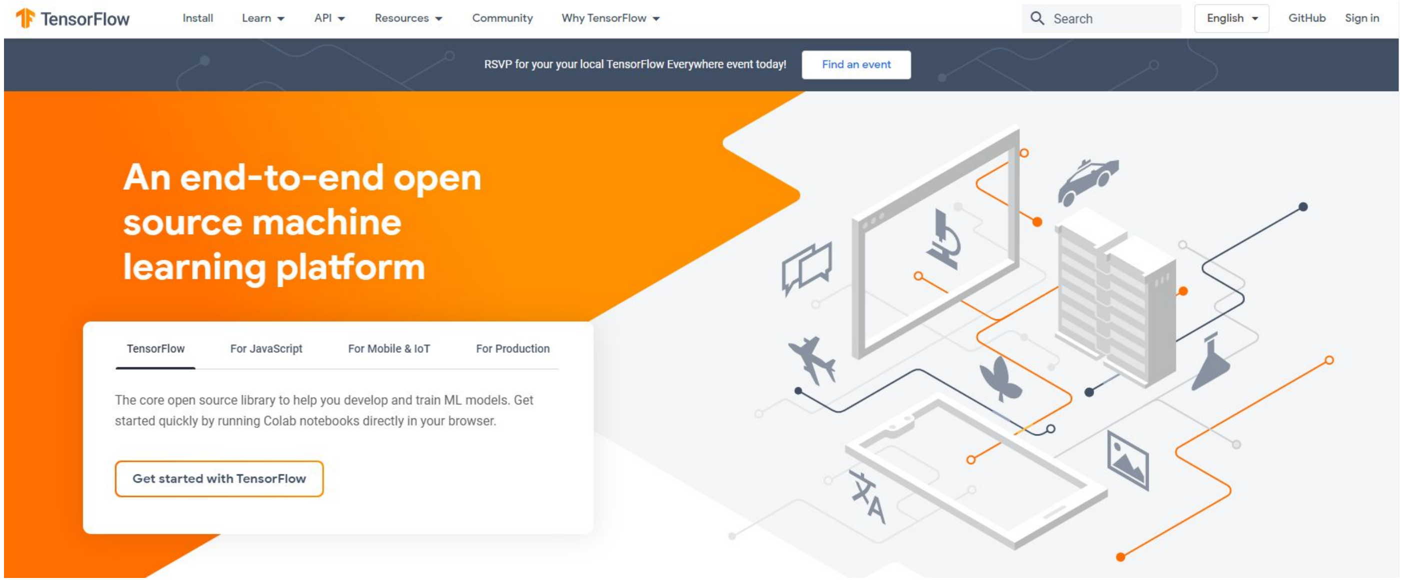


The GAN generates sensory data for simulated users.

Credit: [(Hassouni, 2018)]

Hands-On Practice

- Variational autoencoder tutorial
- Deep convolutional GAN tutorial, CycleGAN tutorial



Recap

Generative adversarial networks are very cool!

- Likelihood-free learning, but can be difficult to converge
- Extensions like InfoGAN and CycleGAN have many applications

As our computing power has improved, deep generative models are becoming a more influential part of our lives

- Detecting deepfakes?
- The value of GAN-generated artwork
- Image translation
- The value of GAN-generated data for privacy