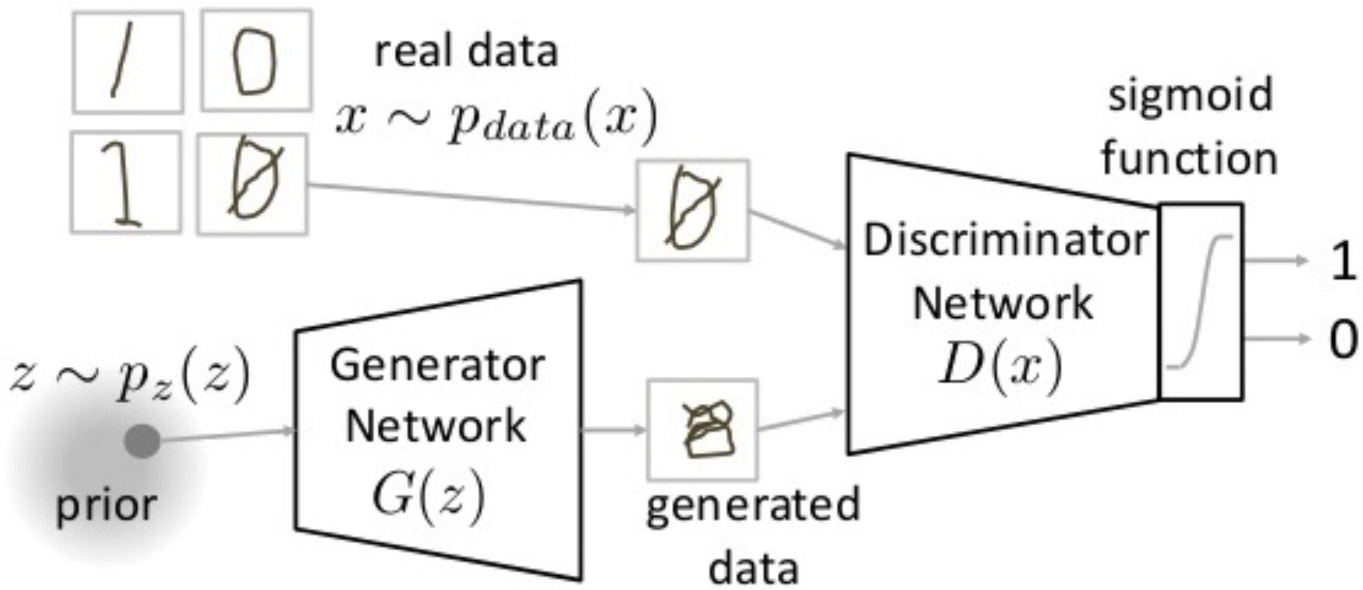
# Generative Adversial Networks




by aydin ahmadli

# ROADMAP

- Supervised vs Unsupervised Learning
- Why study Generative Modeling?
- How do generative models work?
- Generative adversarial network

# Supervised vs Unsupervised Learning

## Supervised Learning

**Data**: (x,y)
x is data, y is label

**Goal**: Learn a *function* to map x->y

**Examples**: Classification, regression, object detection, semantic segmentation, image captioning, etc.

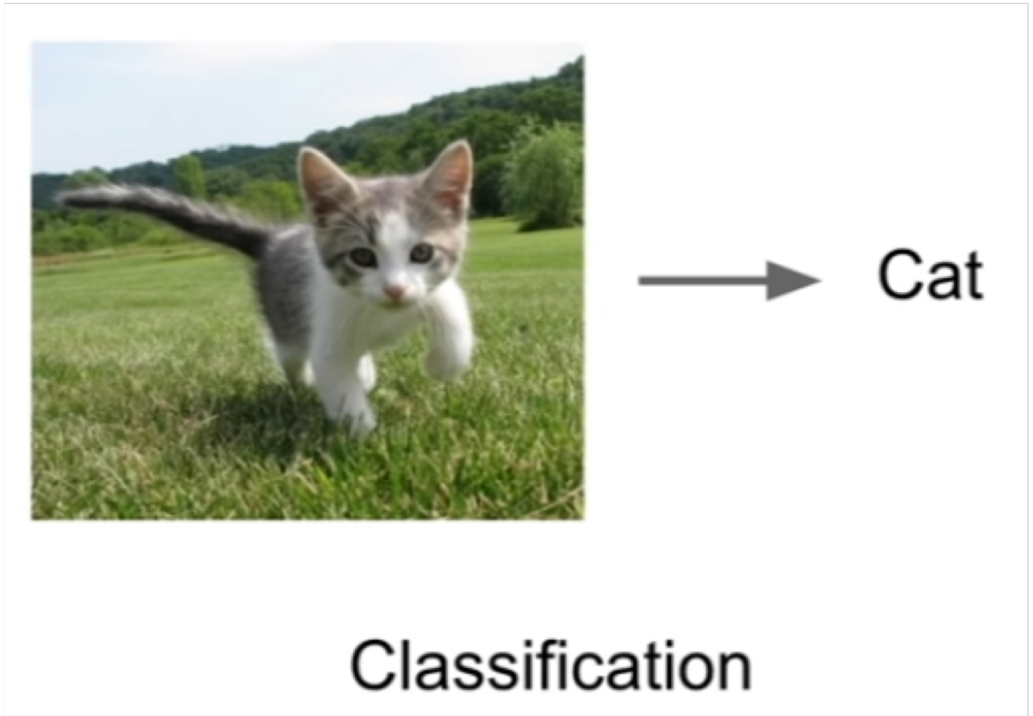
# Supervised vs Unsupervised Learning

**Supervised Learning**

**Data**: (x,y)
x is data, y is label

**Goal**: Learn a *function* to map x->y

**Examples**: Classification, regression, object detection, semantic segmentation, image captioning, etc.



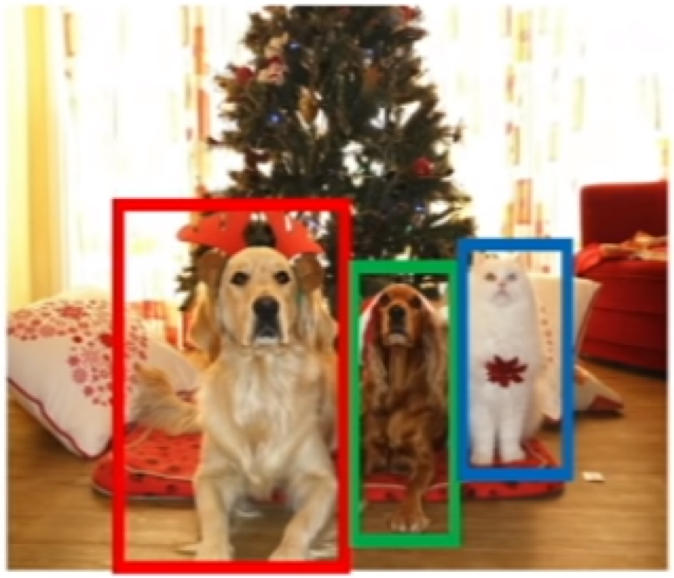
# Supervised vs Unsupervised Learning

## Supervised Learning

**Data**: (x,y)
x is data, y is label

**Goal**: Learn a *function* to map x->y

**Examples**: Classification, regression, object detection, semantic segmentation, image captioning, etc.



DOG, DOG, CAT

Object Detection

# Supervised vs Unsupervised Learning

## Supervised Learning

**Data**: (x,y)
x is data, y is label

**Goal**: Learn a *function* to map x->y

**Examples**: Classification, regression, object detection, semantic segmentation, image captioning, etc.


Semantic Segmentation

# Supervised vs Unsupervised Learning

## Supervised Learning

**Data**: (x,y)
x is data, y is label

**Goal**: Learn a *function* to map x->y

**Examples**: Classification, regression, object detection, semantic segmentation, image captioning, etc.

*A cat sitting on a suitcase on the floor*

Image captioning

# Supervised vs Unsupervised Learning

## Unsupervised Learning

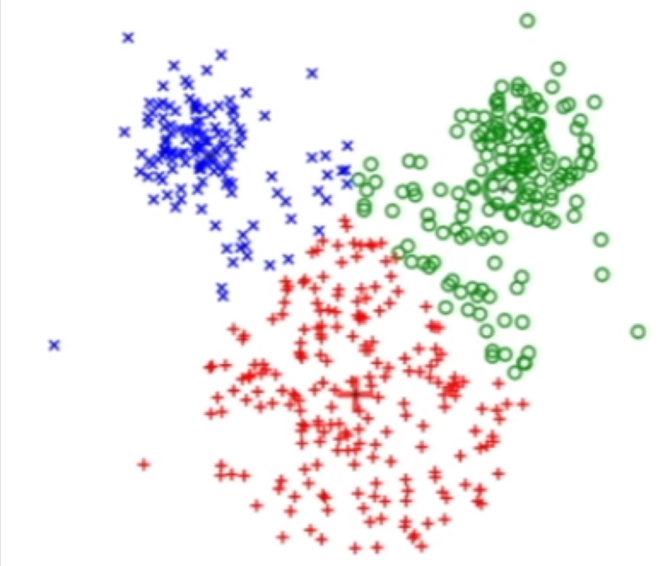
**Data**: x
Just data, no labels!

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

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

# Example 1: K-means clustering

**Goal** : to find groups within the data that are similar by some type of metric.

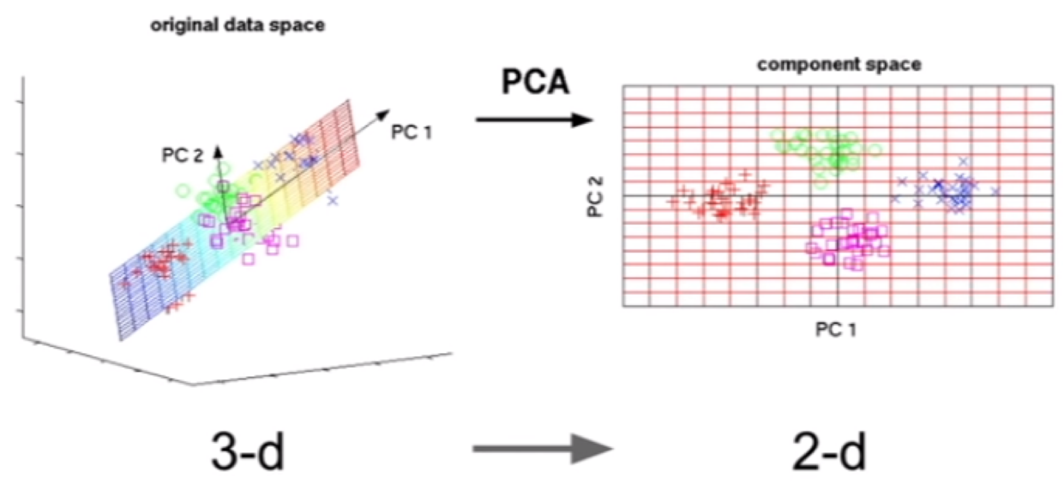


K-means clustering

# Example 2: Dimensionality reduction

**Goal**: to find axes along which our training data has the most variation.
**Underlying Structure**: axes

*In the right example, we start off with data in 3D and we are going to find two axes of variation and reduce our data projected down to 2D.*

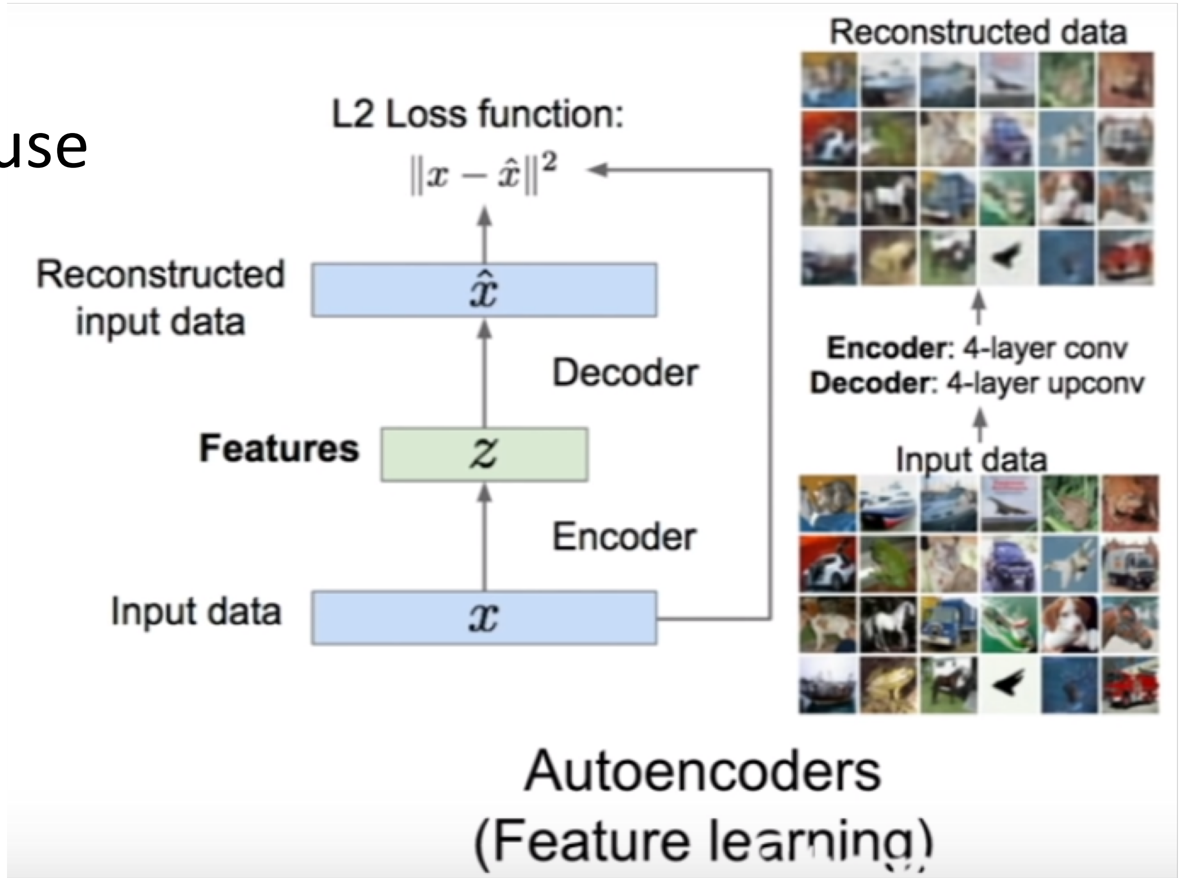



Principal Component Analysis
(Dimensionality reduction)

# Example 3: Feature learning

**In this case**, our *loss* is trying to reconstruct the input data and use it to learn features.

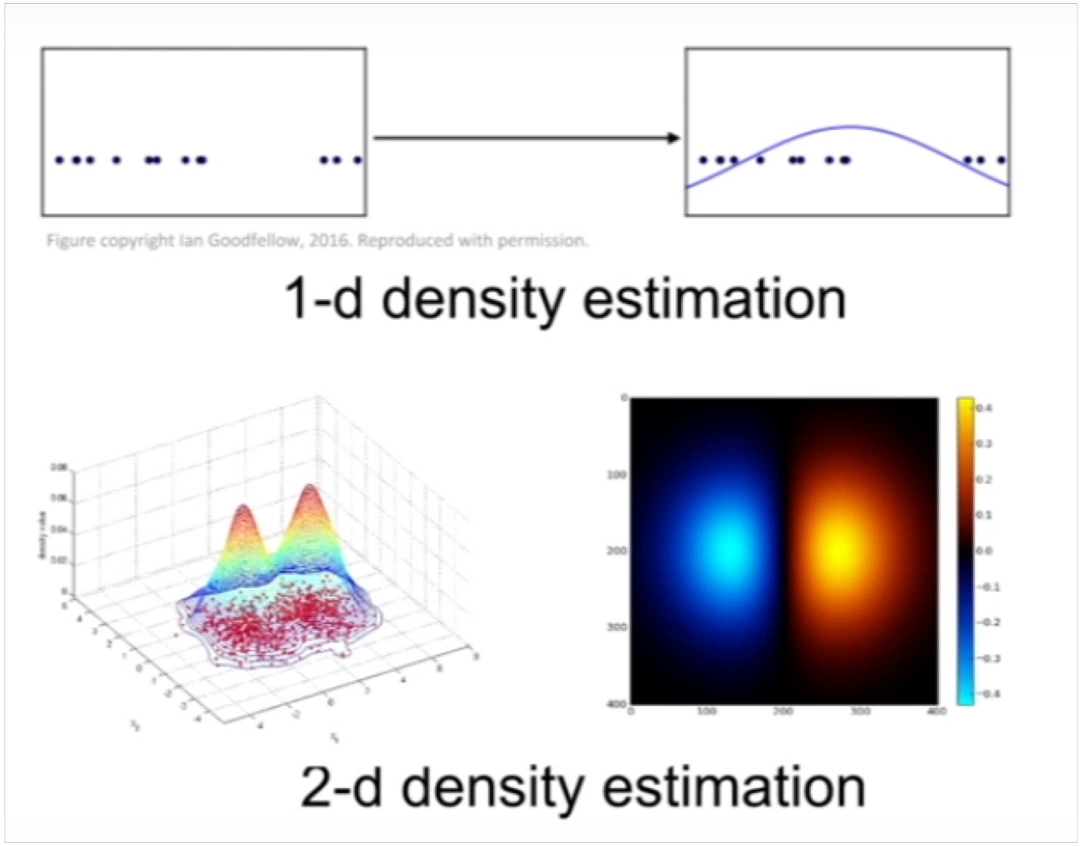
**Advantage**: We are learning a feature representation without using any *additional external labels*



# Example 4: Density estimation

**Goal**: to estimate ***underlying distribution*** of our data.

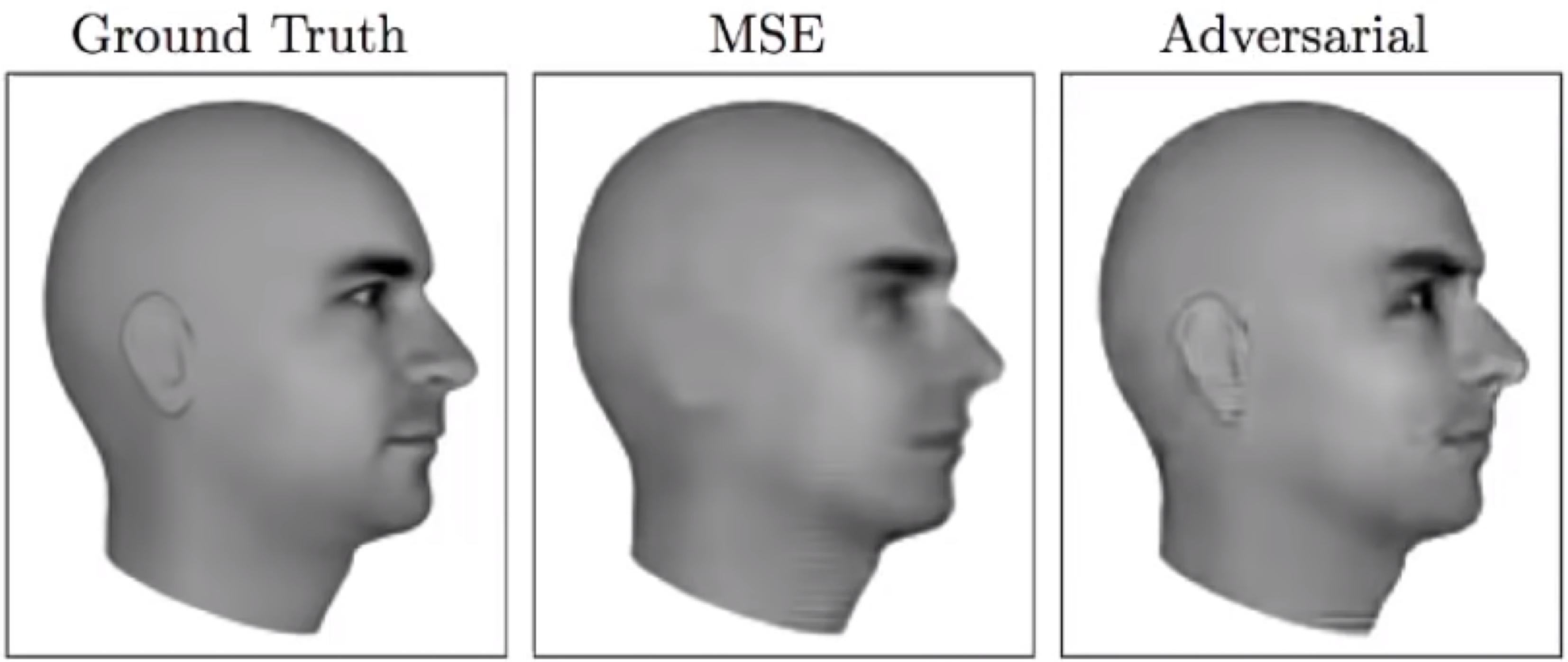
**In the right example**, in top case, we have points in 1D. And we fit *Gaussian* into this density. In bottom case, we have data in 2D and we fit the model such that density is higher where there is more points concentrated



# Why study Generative Models?

- Simulate possible futures for planning. (Reinforcement Learning)
- Missing data
  - Semi-supervised learning
- Multi-modal outputs
- Realistic generation tasks

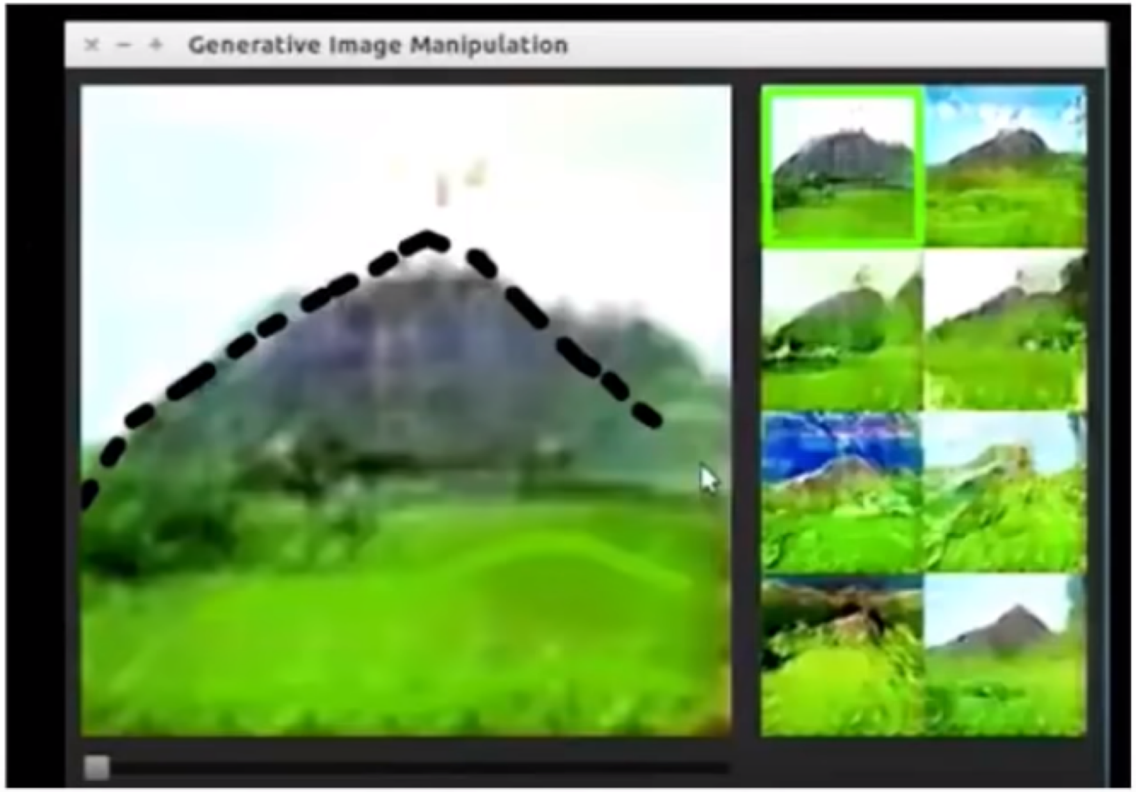
# Next Video Frame Prediction



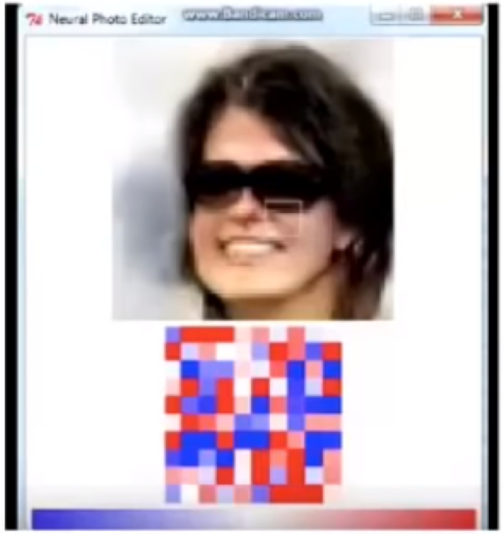
# Single Image Super-Resolution

| original | bicubic (21.59dB/0.6423) | SRResNet (23.44dB/0.7777) | SRGAN (20.34dB/0.6562) |
|---|---|---|---|

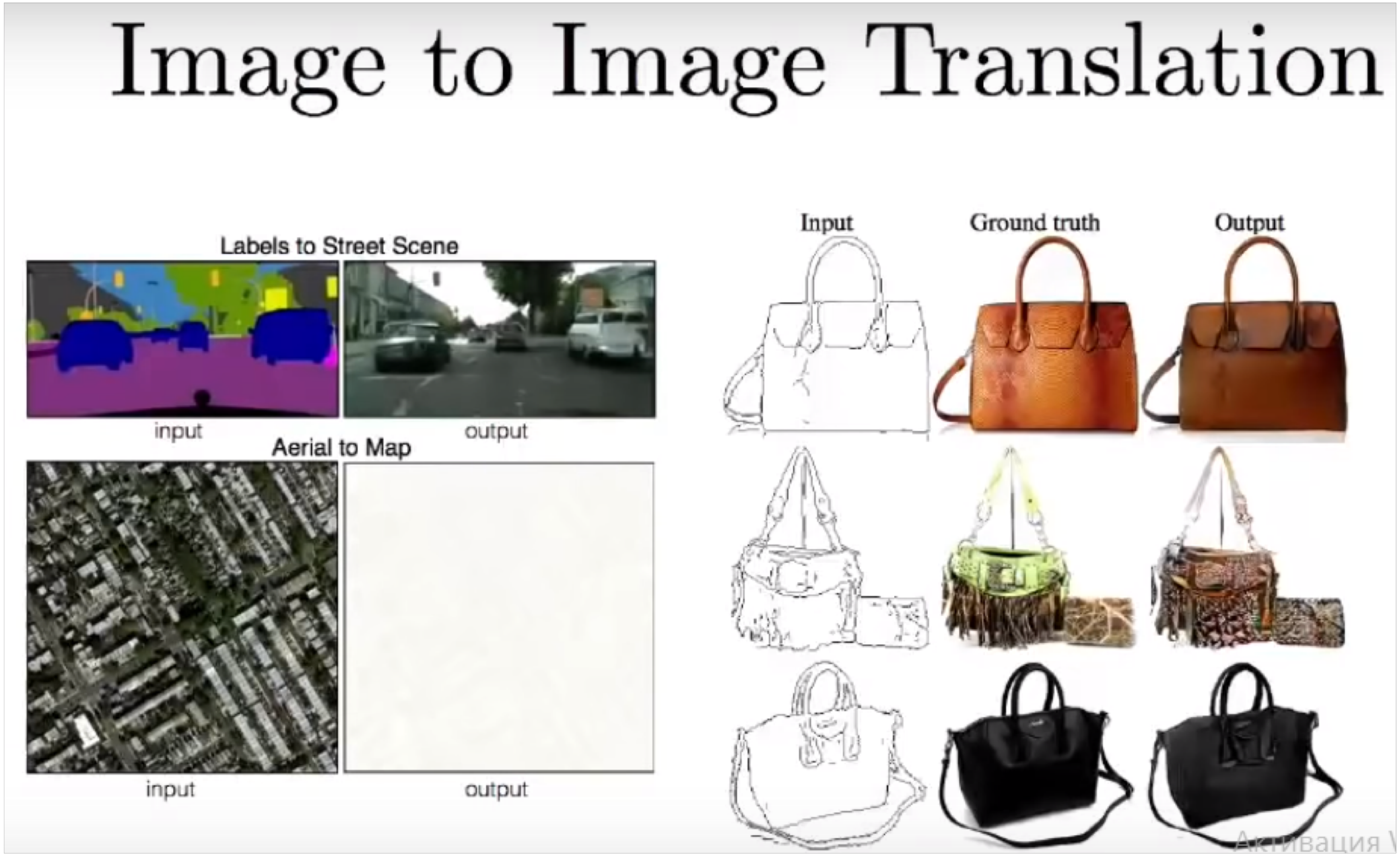
# iGAN



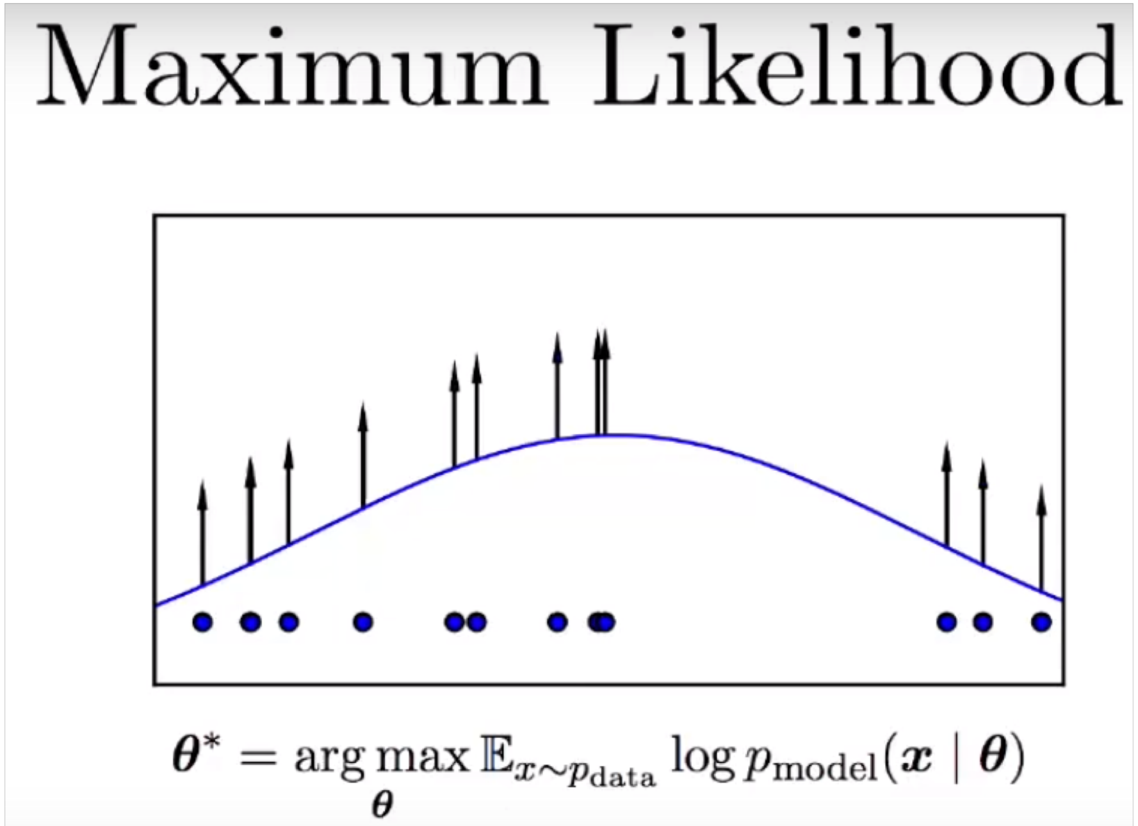
# Introspective Adversarial Networks




youtube

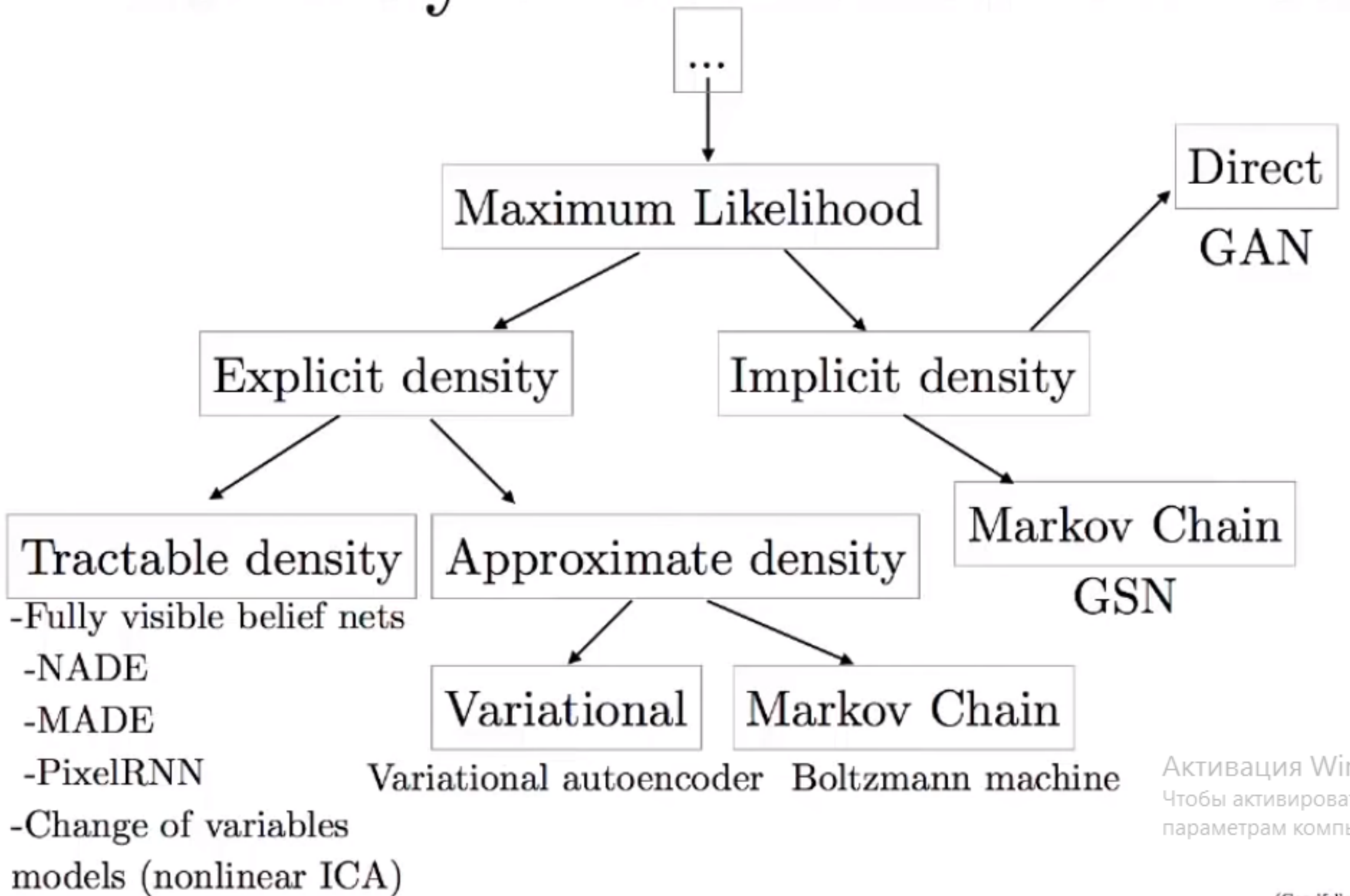



https://www.youtube.com/watch?v=EYjdLppmERE




Its easiest to compare many different models if we describe all of them as performing *Maximum Likelihood*

# Taxonomy of Generative Models





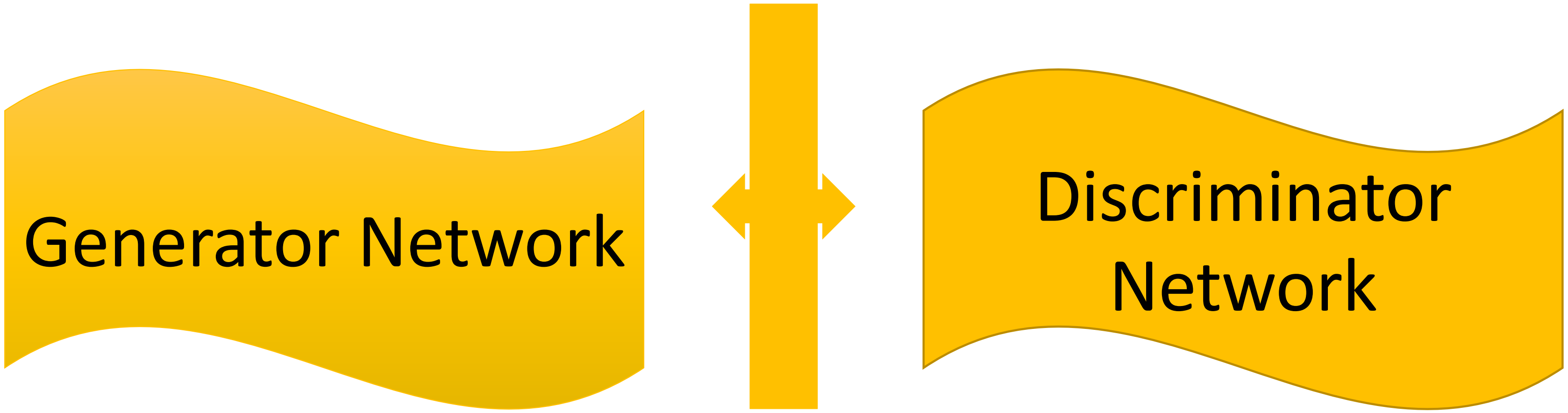

(Goodfellow 201

# Generative Adversarial Networks

# Advantages of GANs

1. They use a latent code that describes everything that is generated later. They have this property in common with other models like Variational Autoencoders and Boltzmann Machines . It is advantage they have over fully visible belief networks.
2. They are asymptotically consistent. So, if we are able to find the equilibrium point of the game defining generative adversarial network, we are guaranteed that we have actually recovered the true distribution that generates the data. For example, if we have infinite data, we eventually recover the correct distribution.
3. There are no Markov Chains needed neither to train Generative Adversarial Network nor to draw samples from it which is an important requirement.
4. They are often regarded as producing the best samples compared to other models

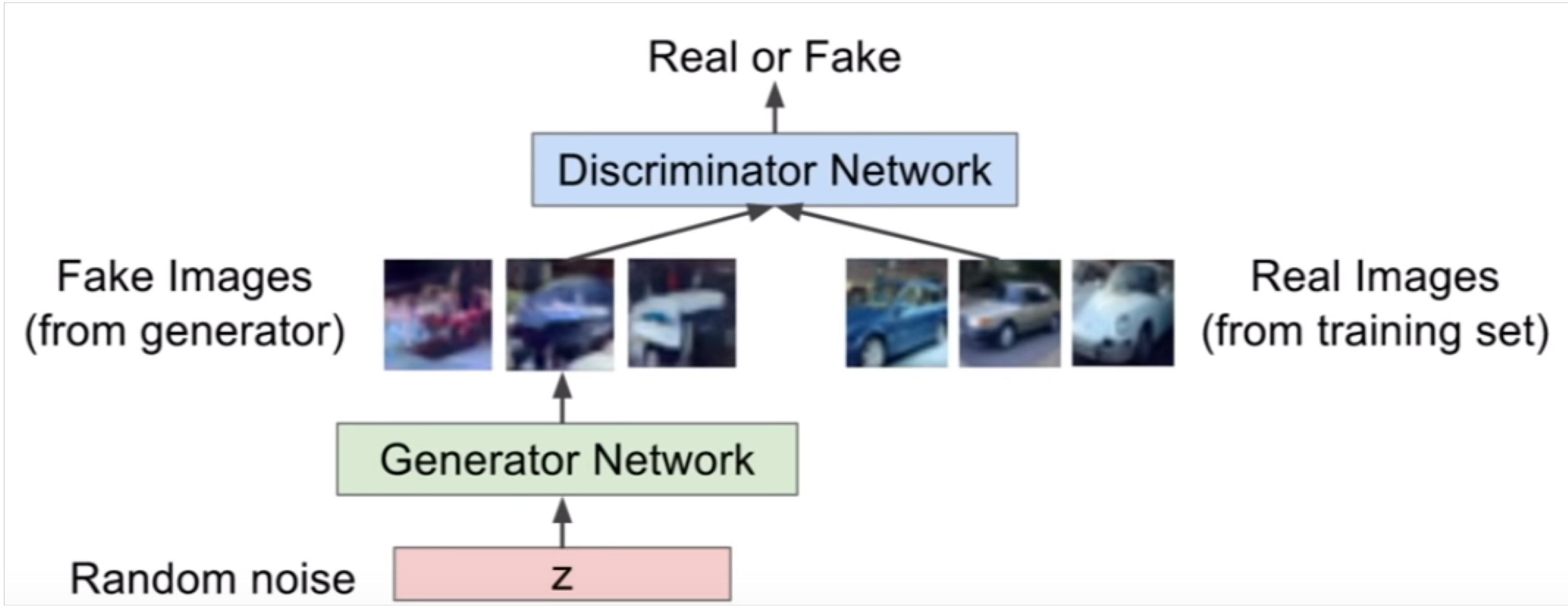
# Training GANs: Two-player game




Try to fool the discriminator by generating real-looking images

Try to distinguish between real and fake images

# Two-player Game



# Adversarial Nets Framework

x sampled from data

Differentiable function D

D(x) tries to be near 1

Input noise z

Differentiable function G

x sampled from model

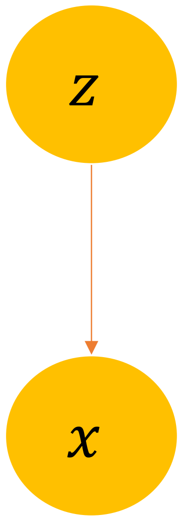
D

D tries to make D(G(z)) near 0, G tries to make D(G(z)) near 1,

# Generator Network

$$x = G(z; \theta^{(G)})$$

1. G must be differentiable
2. No invertibility required
3. Trainable for any size of z

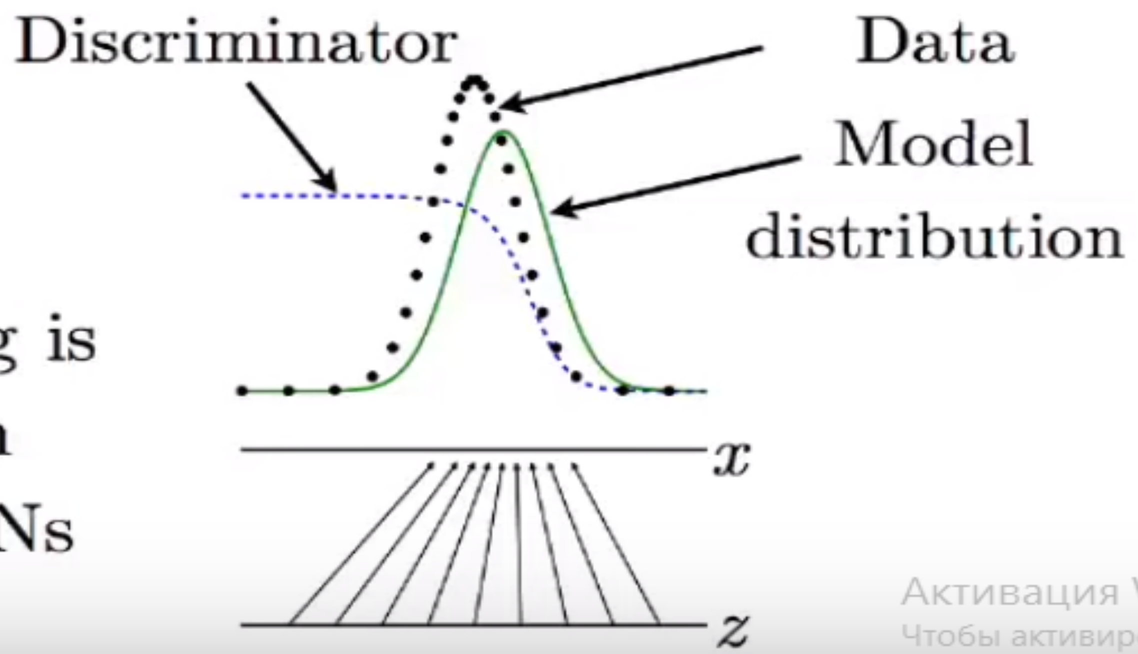


# Discriminator Strategy

Optimal $D(\boldsymbol{x})$ for any $p_{\text{data}}(\boldsymbol{x})$ and $p_{\text{model}}(\boldsymbol{x})$ is always

$$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)}$$

Estimating this ratio using supervised learning is the key approximation mechanism used by GANs



## Minimax Game

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

-Equilibrium is a saddle point of the discriminator loss

-Generator minimizes the log-probability of the discriminator being correct

# Minimax Game

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

- Discriminator ($\theta_d$) wants to **maximize objective** such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)
- Generator ($\theta_g$) 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 Procedure

- Use **Stochastic Gradient Descent** - optimization algorithm of choice on two minibatches simultaneously.
  - A minibatch of training examples
  - A minibatch of generated samples
- Optional: run ***k*** steps of one player for every step of the other player.

# 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**

# Generative Adversarial Nets: Convolutional Architectures

Generator is an upsampling network with fractionally-strided convolutions
Discriminator is a convolutional network

Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

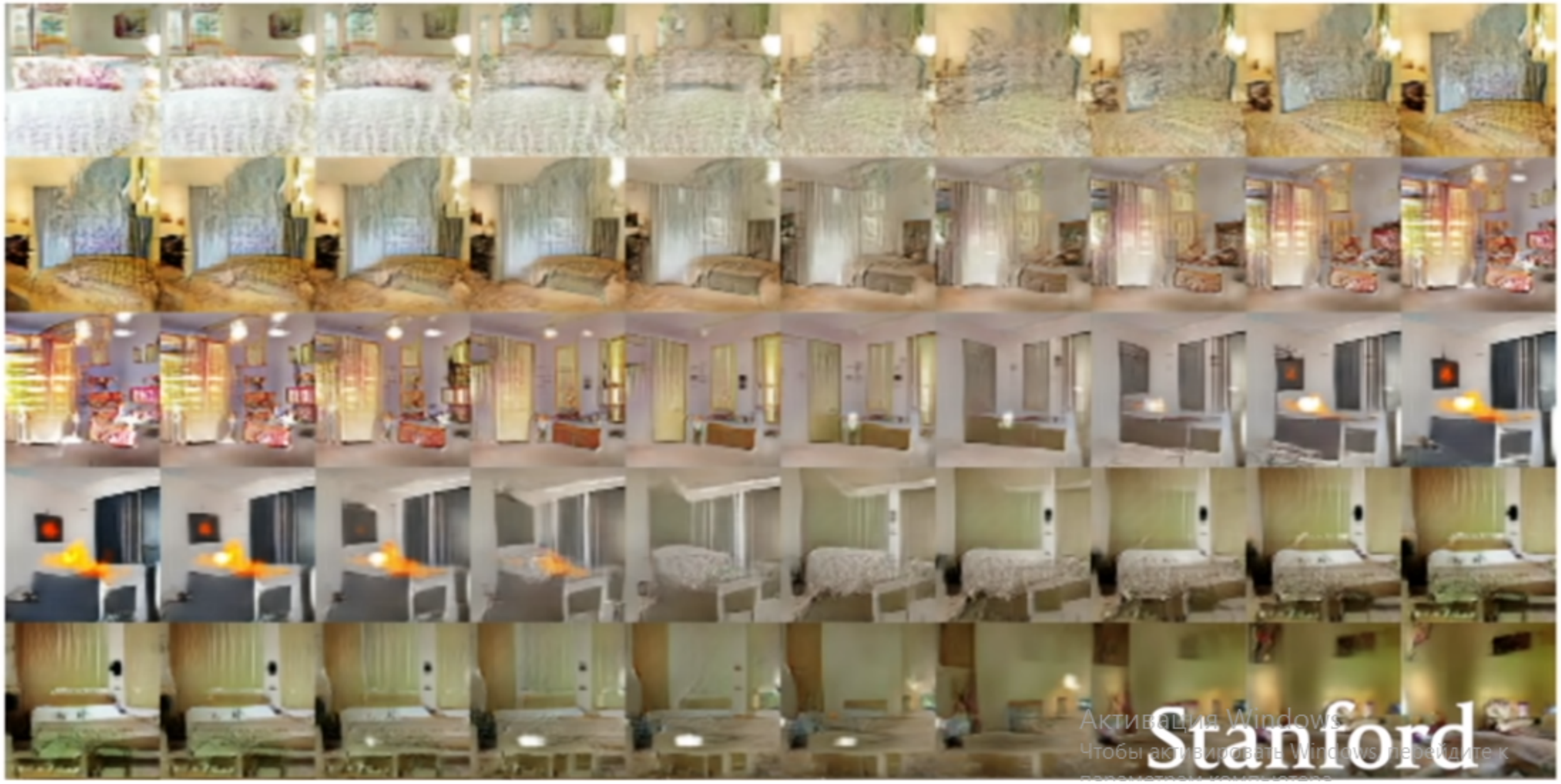
# Generative Adversarial Nets: Convolutional Architectures

Samples from the model look amazing!

Radford et al, ICLR 2016

# Generative Adversarial Nets: Convolutional Architectures

Interpolating between random points in laten space




Radford et al, ICLR 2016

# 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

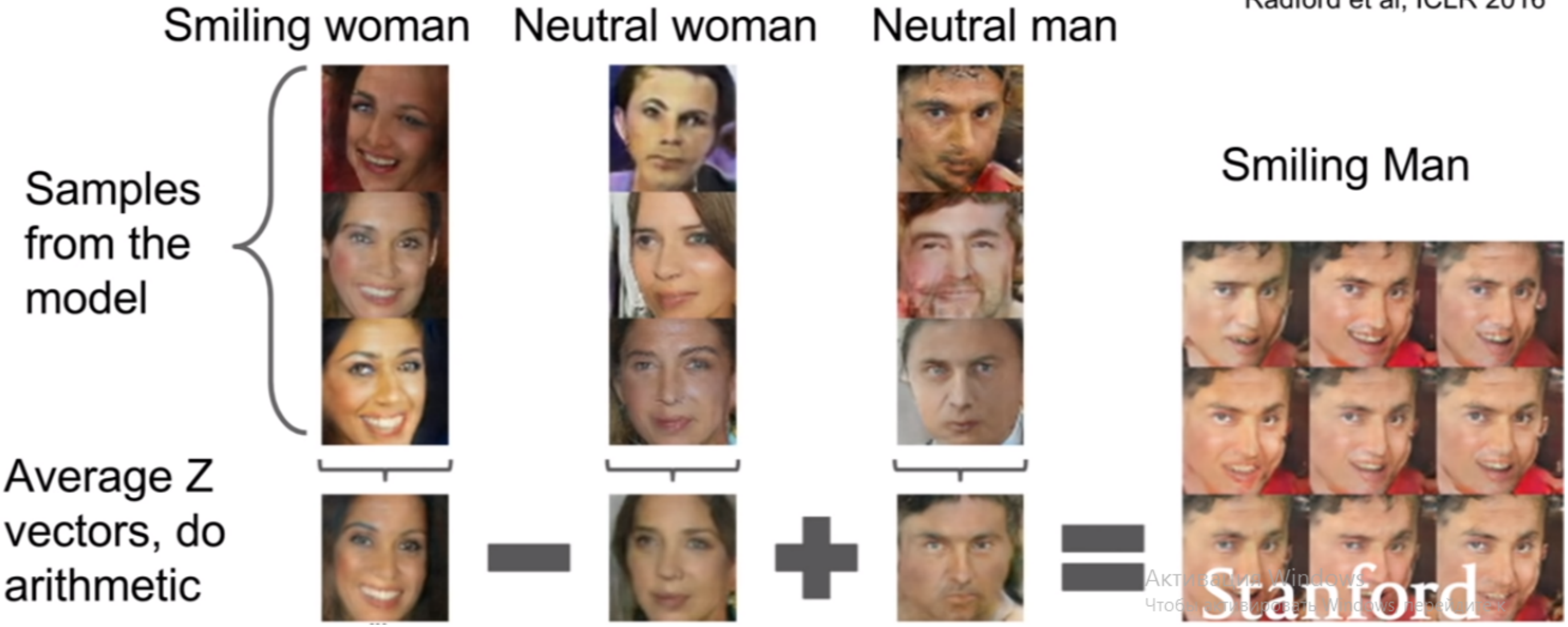
Smiling woman Neutral woman Neutral man

Samples from the model

Average Z vectors, do arithmetic

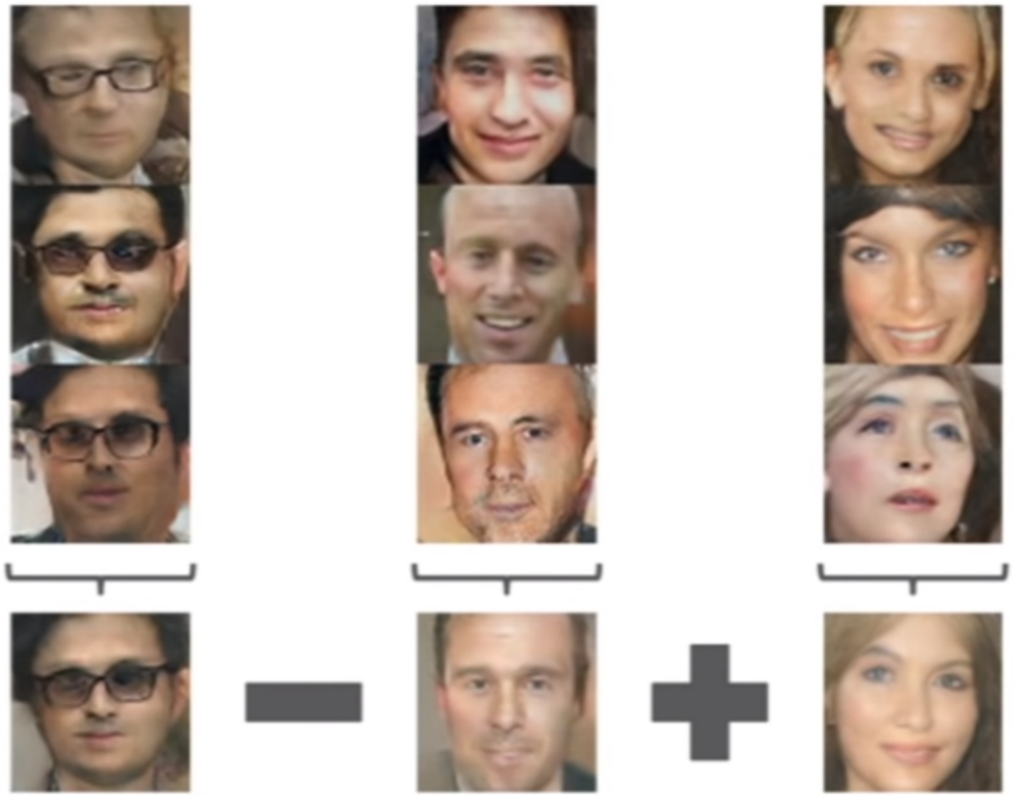
# Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016



# Generative Adversarial Nets: Interpretable Vector Math

Glasses man    No glasses man    No glasses woman



Radford et al,
ICLR 2016

# Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

Glasses man    No glasses man    No glasses woman    Woman with glasses

# References:


1. https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

2. https://towardsdatascience.com/generative-adversarial-networks-gans-a-beginners-guide-5b38eceece24

3. https://www.kdnuggets.com/2018/10/generative-adversarial-networks-paper-reading-road-map.html

4. https://openreview.net/forum?id=Byxz4n09tQ