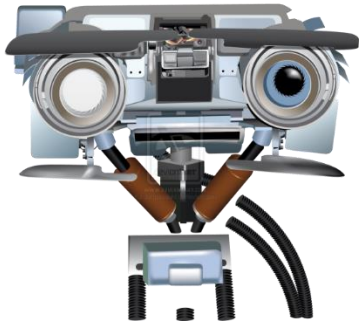
# Today: Outline

- Generative Adversarial Networks (GANs)
- Domain Adaptation/Generalization

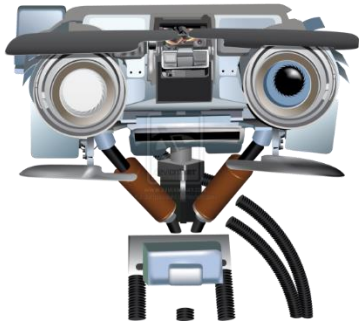
- Reminders:
  - Exam Jun 22 in class
    *(and ~12 hrs before for remote only students)*
  - Team Registration
  - Practice problems available
  - Thu is a free-choice lecture



# Generative Adversarial Networks (GANs)

Unsupervised Learning

Slides from lectures by Fei-Fei Li, Justin Johnson & Serena Yeung, Roger Grosse



# Supervised vs Unsupervised Learning Recap

Generative Adversarial Networks (GANs)

# 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.

Cat

Classification

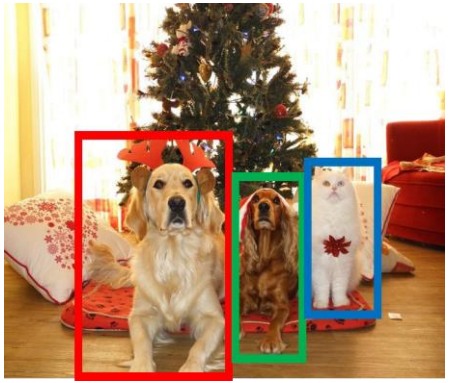
# Supervised vs Unsupervised Learning

**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

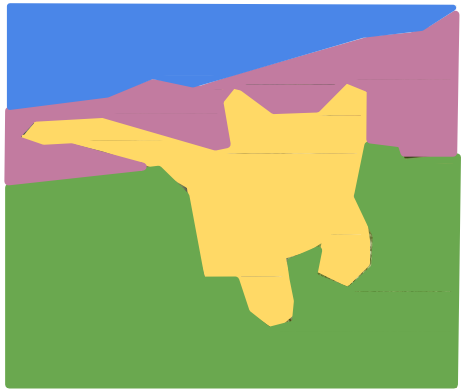
# Supervised vs Unsupervised Learning

**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.



GRASS, CAT,
TREE, SKY

Semantic Segmentation

# 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.

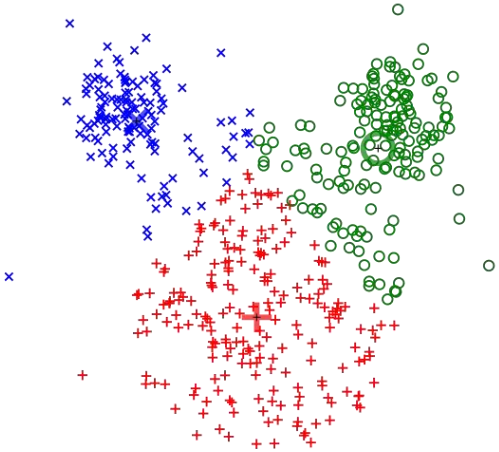
# Supervised vs Unsupervised Learning

**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

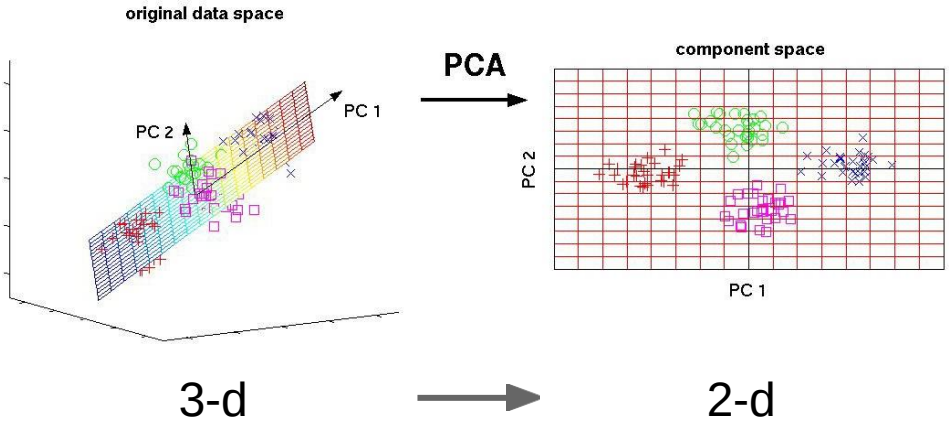
# Supervised vs Unsupervised Learning

**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.




Principal Component Analysis
(Dimensionality reduction)

# Supervised vs Unsupervised Learning

**Unsupervised Learning**

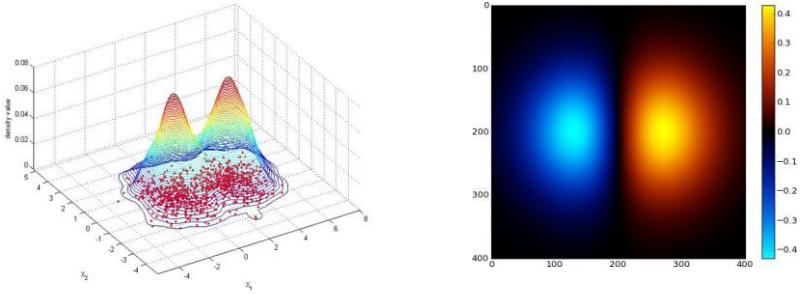
**Data**: x
Just data, no labels!

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

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

Figure copyright Ian Goodfellow, 2016. Reproduced with permission.

1-d density estimation

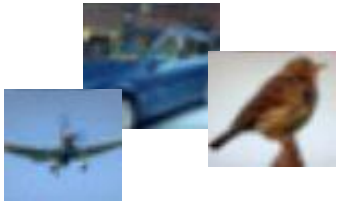


2-d density estimation

2-d density images left and right are CC0 public domain

# Generative Models

Given training data, generate new samples from same distribution



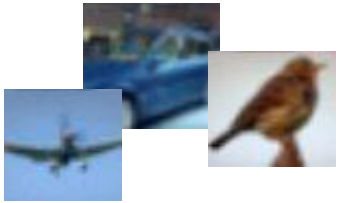
Training data ~ $p_{data}(x)$

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

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

# Generative Models

Given training data, generate new samples from same distribution



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 of time-series data can be used for simulation and planning (reinforcement learning applications!)
- Training generative models can also enable inference of latent representations that can be useful as general features

# Taxonomy of Generative Models

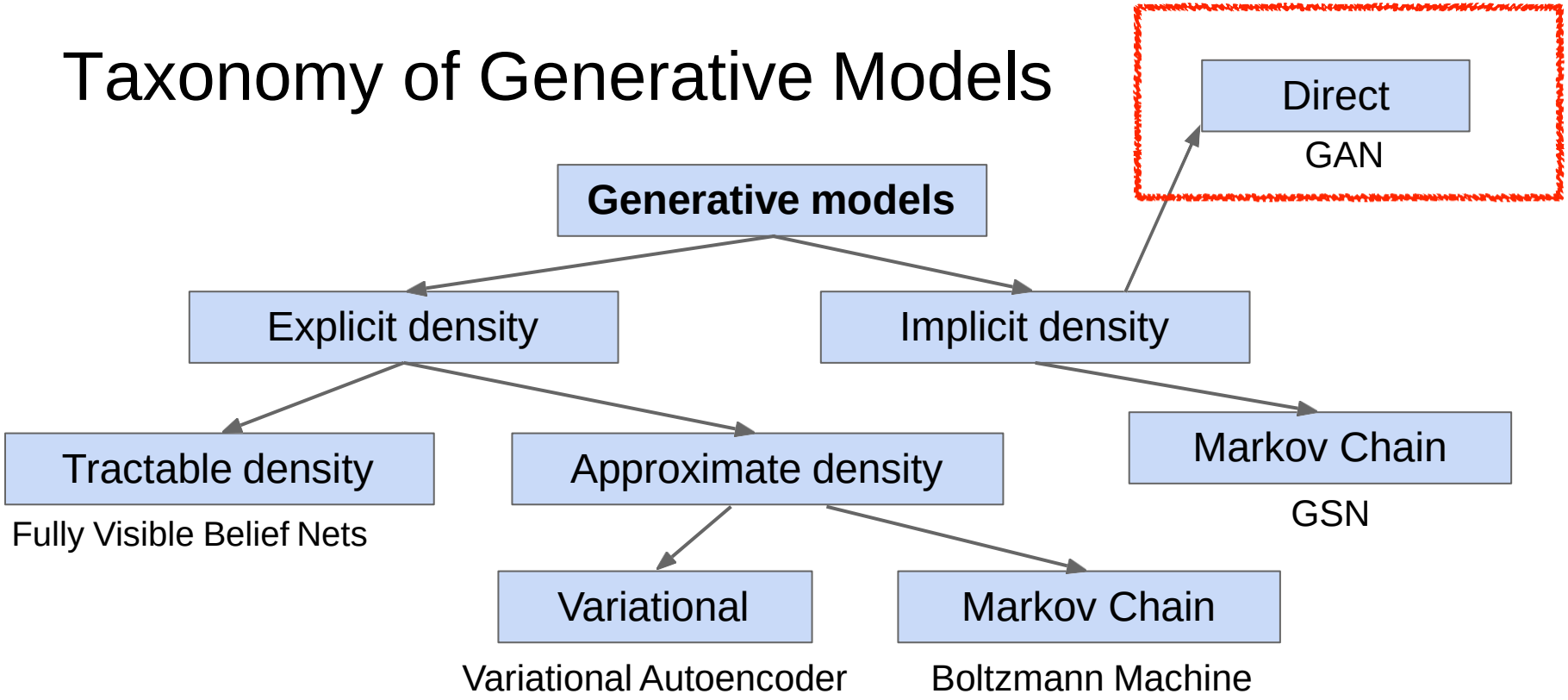
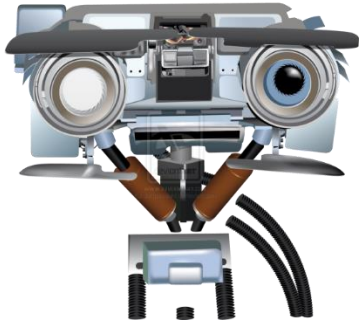



Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

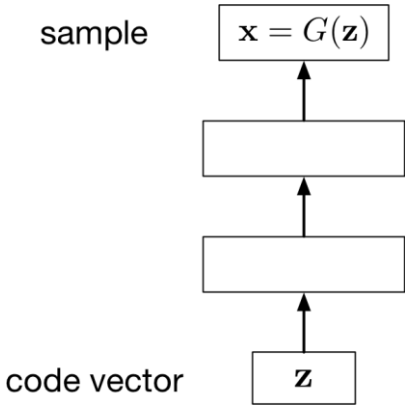


# Generative Adversarial Networks

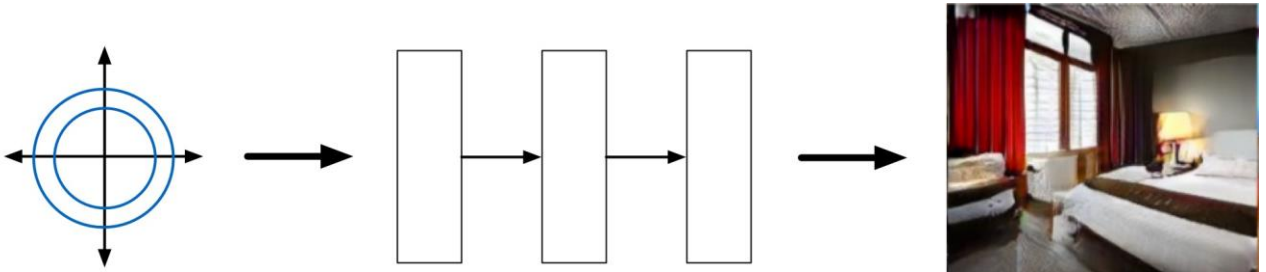
Generative Adversarial Networks (GANs)

# Implicit Generative Models

- Implicit generative models implicitly define a probability distribution
- Start by sampling the code vector **z** from a fixed, simple distribution (e.g. spherical Gaussian)
- The generator network computes a differentiable function $G$ mapping **z** to an **x** in data space



# Implicit Generative Models



Each dimension of the code vector is sampled independently from a simple distribution, e.g. Gaussian or uniform.

This is fed to a (deterministic) generator network.

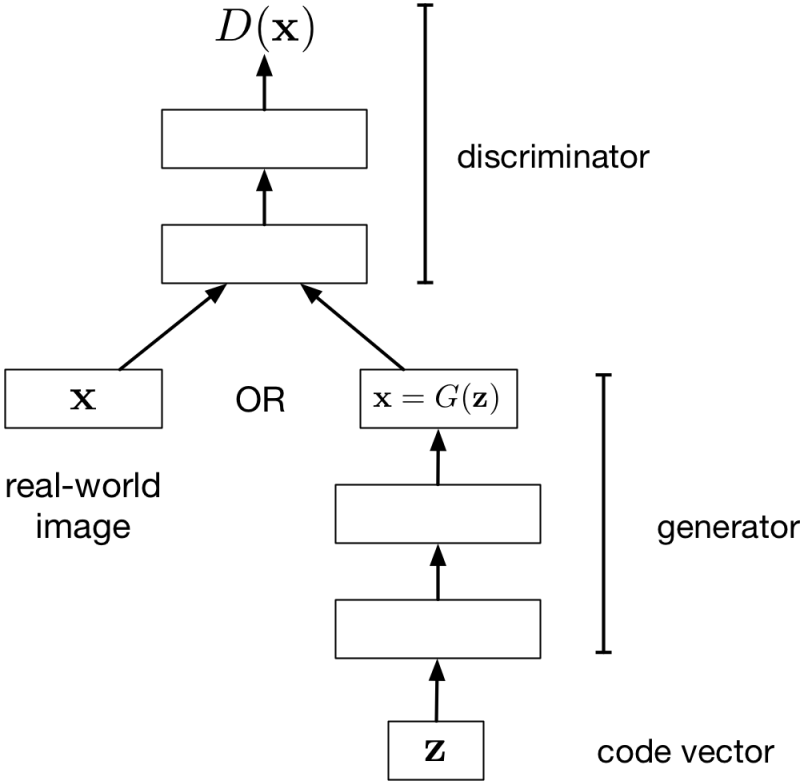
The network outputs an image.

This sort of architecture sounded preposterous to many of us, but amazingly, it works.

# Generative Adversarial Networks

- The advantage of implicit generative models: if you have some criterion for evaluating the quality of samples, then you can compute its gradient with respect to the network parameters, and update the network's parameters to make the sample a little better
- The idea behind Generative Adversarial Networks (GANs): train two different networks
  - The generator network tries to produce realistic-looking samples
  - The discriminator network tries to figure out whether an image came from the training set or the generator network
- The generator network tries to fool the discriminator network

# Generative Adversarial Networks



# Generative Adversarial Networks

- Let $D$ denote the discriminator's predicted probability of being data
- Discriminator's cost function: cross–entropy loss for task of classifying real vs. fake images

$$J_D = \mathrm{E}_{\mathbf{x}\sim D}[-\log D(\mathbf{x})] + \mathrm{E}_{\mathbf{z}}[-\log(1 - D(G(\mathbf{z})))]$$

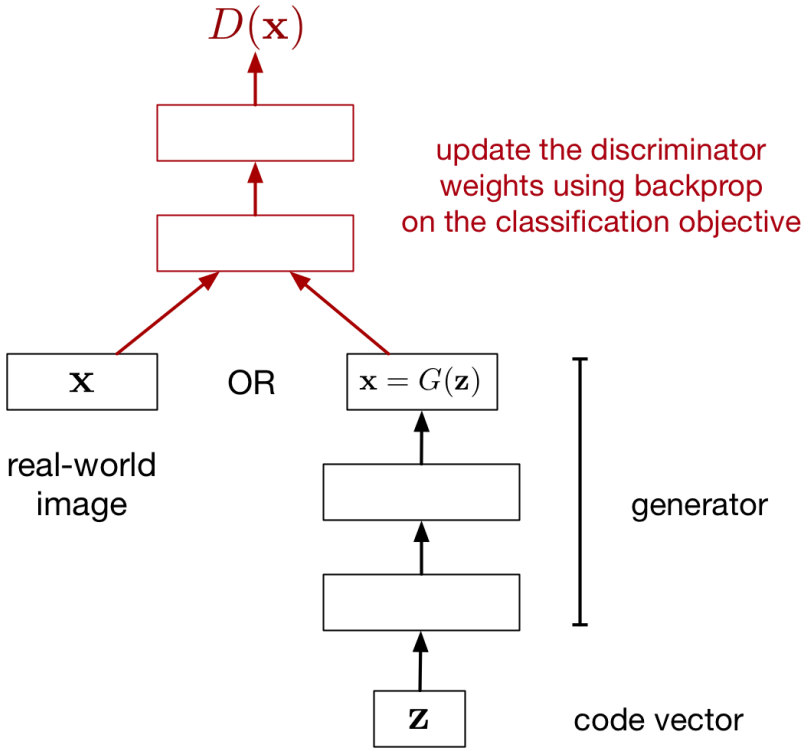
- One possible cost function for the generator: the opposite of the discriminator's

$$\begin{aligned} J_G &= -J_D \\ &= \text{const} + \mathrm{E}_{\mathbf{z}}[\log(1 - D(G(\mathbf{z})))] \end{aligned}$$

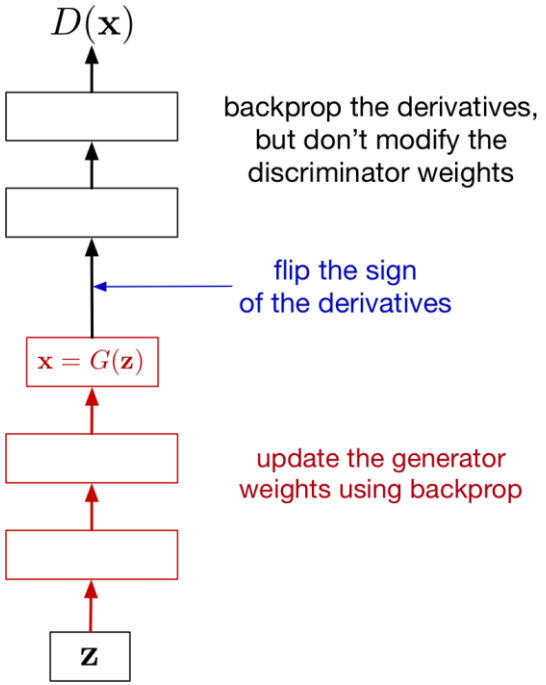
- This is called the minimax formulation, since the generator and discriminator are playing a zero–sum game against each other:

$$\max_G \min_D J_D$$

Updating the discriminator:

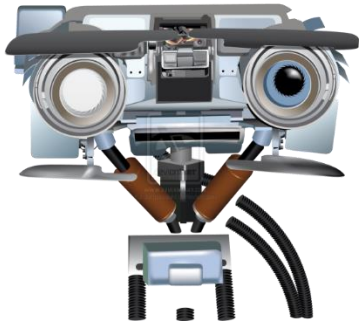


Updating the generator:



# Generative Adversarial Networks

- Since GANs were introduced in 2014, there have been hundreds of papers introducing various architectures and training methods.
- Most modern architectures are based on the Deep Convolutional GAN (DC-GAN), where the generator and discriminator are both conv nets.
- GAN Zoo: https://github.com/hindupuravinash/the-gan-zoo



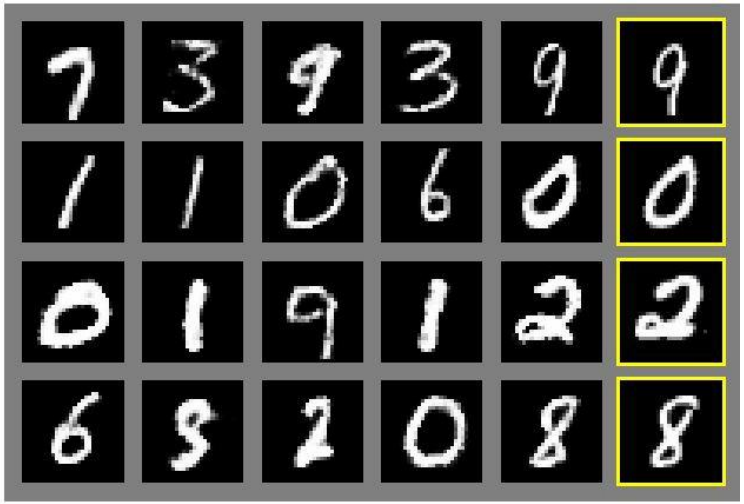
# GANs: Application to Image Generation

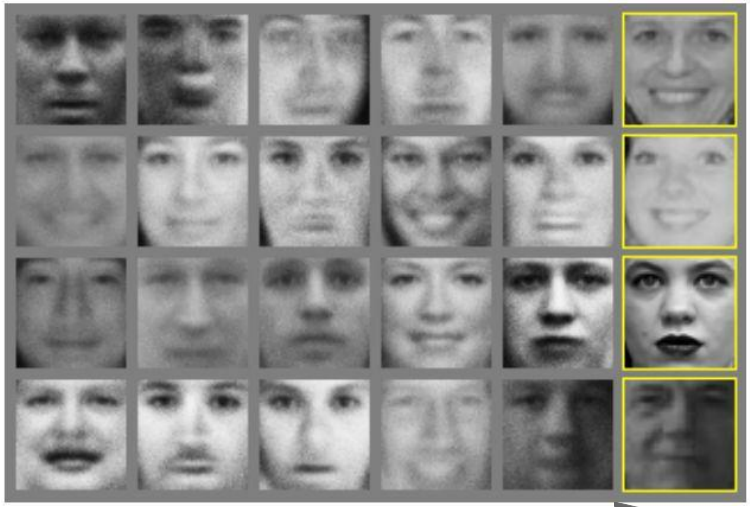
Generative Adversarial Networks (GANs)

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

# Generative Adversarial Nets

## Generated samples



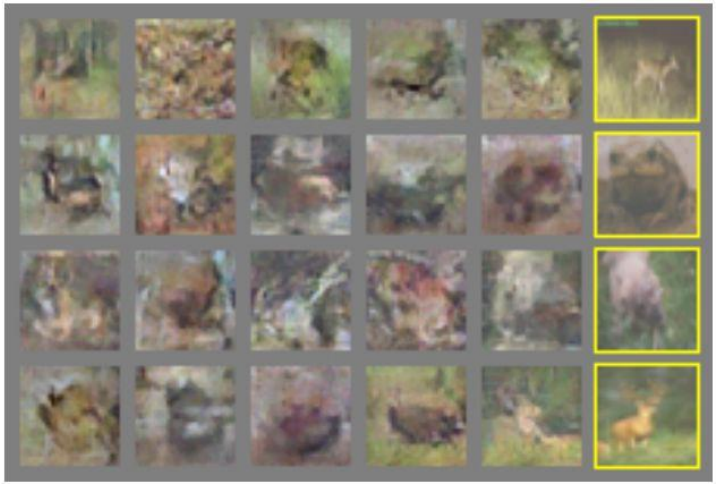


Nearest neighbor from training set

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

# Generative Adversarial Nets

Generated samples (CIFAR-10)



Nearest neighbor from training set

# GAN Samples

Celebrities:

Karras et al., 2017. Progressive growing of GANs for improved quality, stability, and variation

# GAN Samples

Bedrooms:

Karras et al., 2017. Progressive growing of GANs for improved quality, stability, and variation

# GAN Samples

Objects:

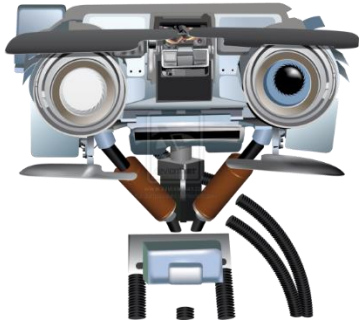

Karras et al., 2017. Progressive growing of GANs for improved quality, stability, and variation

# GAN Samples

GANs revolutionized generative modeling by producing crisp, high-resolution images.

The catch: we don't know how well they're modeling the distribution.

> Could they be memorizing training examples? (E.g., maybe they sometimes produce photos of real celebrities?)
>
> We have no way to tell if they are dropping important modes from the distribution.
>
> See Wu et al., "On the quantitative analysis of decoder-based generative models" for partial answers to these questions.

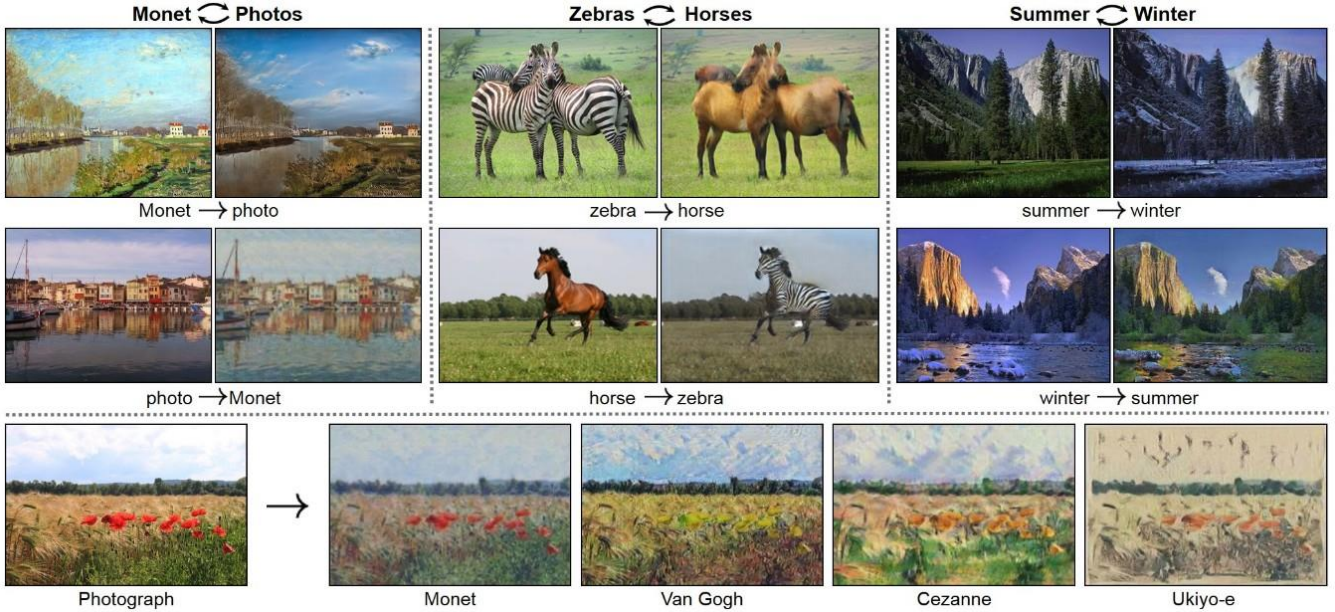


# Cycle GANs

Generative Adversarial Networks (GANs)

# CycleGAN

Style transfer problem: change the style of an image while preserving the content.

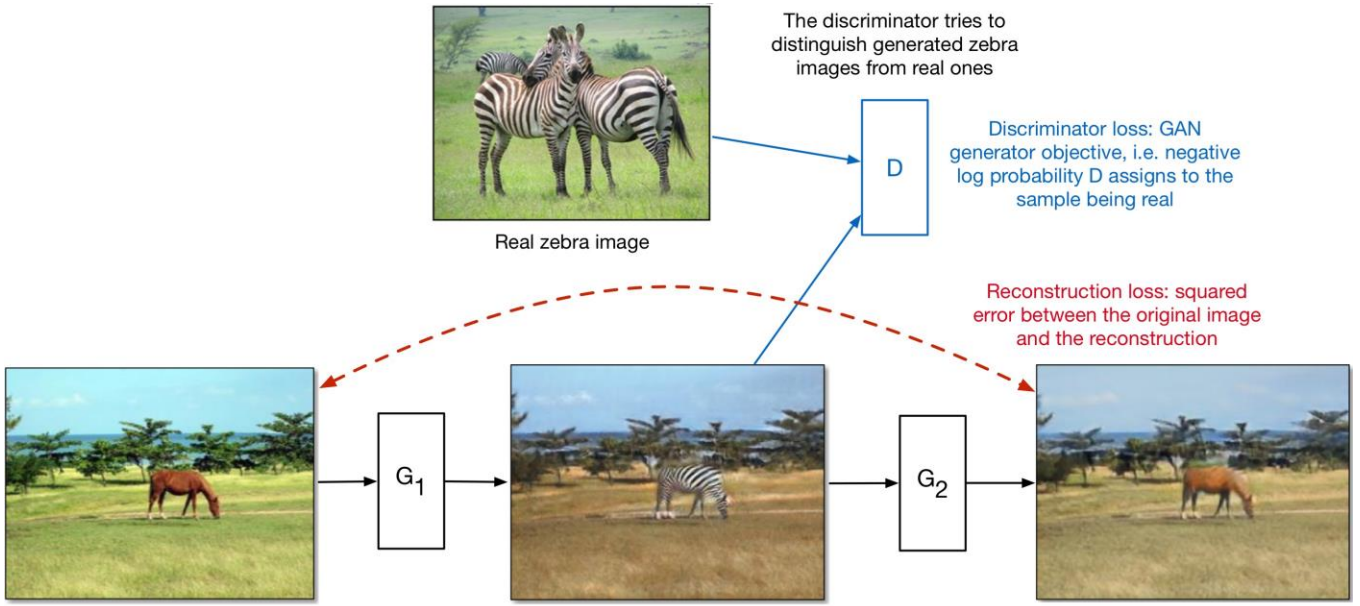



Data: Two unrelated collections of images, one for each style

# CycleGAN

- If we had paired data (same content in both styles), this would be a supervised learning problem. But this is hard to find.
- The CycleGAN architecture learns to do it from unpaired data.
  - Train two different generator nets to go from style 1 to style 2, and vice versa.
  - Make sure the generated samples of style 2 are indistinguishable from real images by a discriminator net.
  - Make sure the generators are cycle-consistent: mapping from style 1 to style 2 and back again should give you almost the original image.

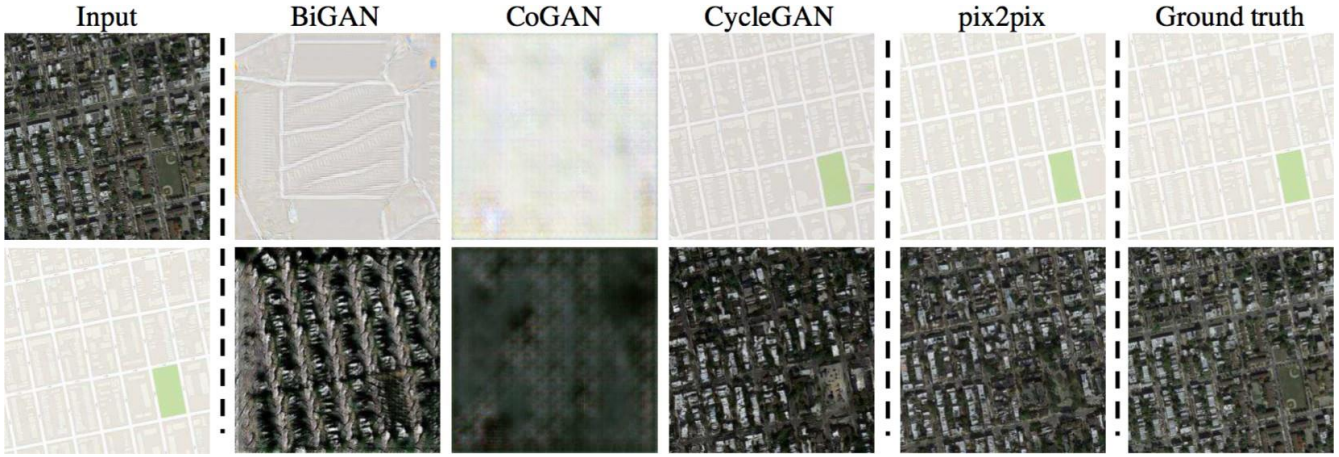
# CycleGAN




Total loss = discriminator loss + reconstruction loss

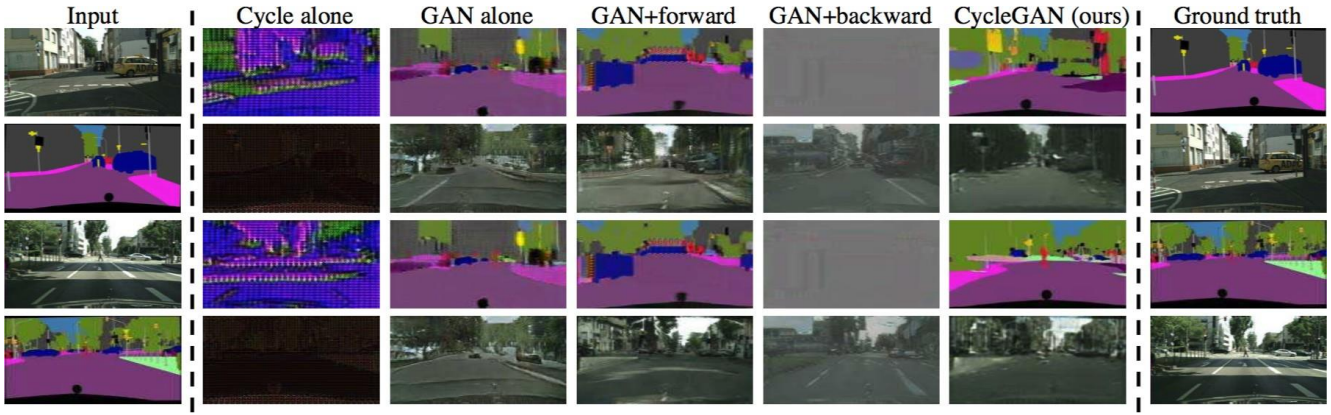
# CycleGAN

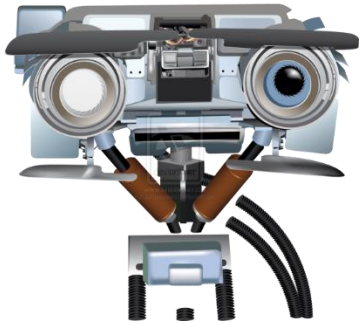
Style transfer between aerial photos and maps:



# CycleGAN

Style transfer between road scenes and semantic segmentations (labels of every pixel in an image by object category):

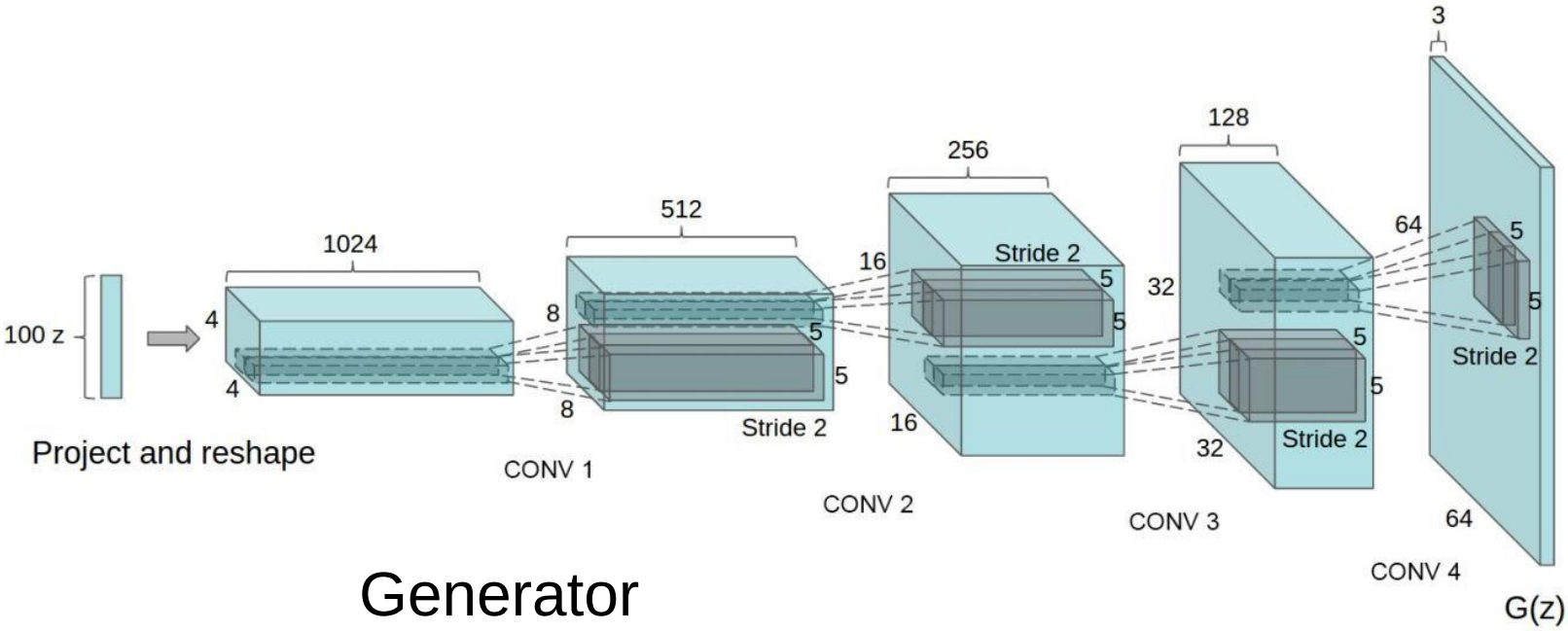




# GANs: Convolutional Architectures

Generative Adversarial Networks (GANs)

# Generative Adversarial Nets: Convolutional Architectures




Generator

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

# Generative Adversarial Nets: Convolutional Architectures

Samples from the model look amazing!

Radford et al, ICLR 2016

# Generative Adversarial Nets: Interpretable Vector Math

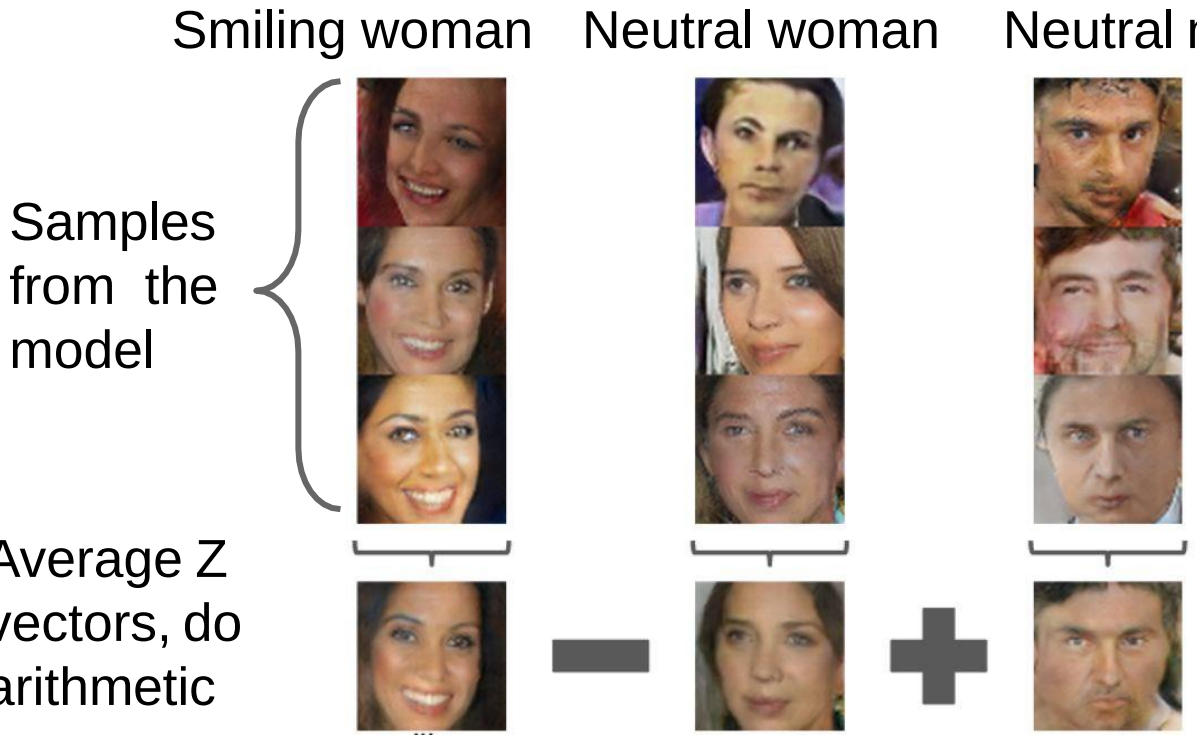
Radford et al, ICLR 2016

Smiling woman | Neutral woman | Neutral man

Samples from the model

# Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016



# Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

Smiling woman | Neutral woman | Neutral man

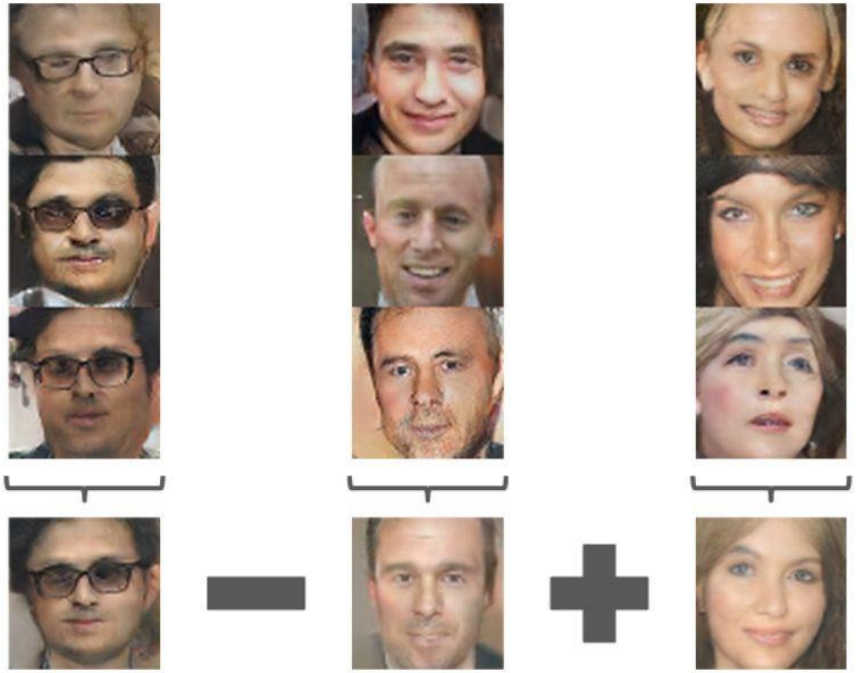
Samples from the model

Average Z vectors, do arithmetic

Smiling Man

# Generative Adversarial Nets: Interpretable Vector Math

Glasses man — No glasses man + No glasses woman



Radford et al, ICLR 2016

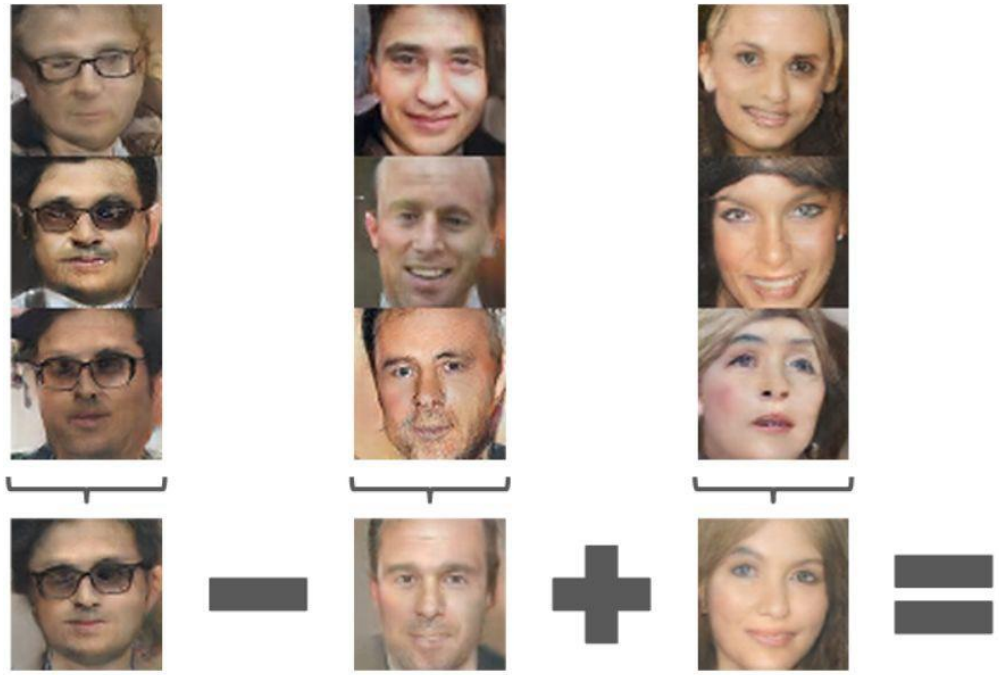
# Generative Adversarial Nets: Interpretable Vector Math

Glasses man

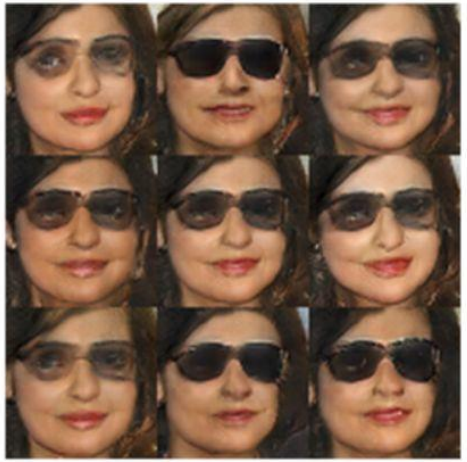
No glasses man

No glasses woman

Radford et al, ICLR 2016

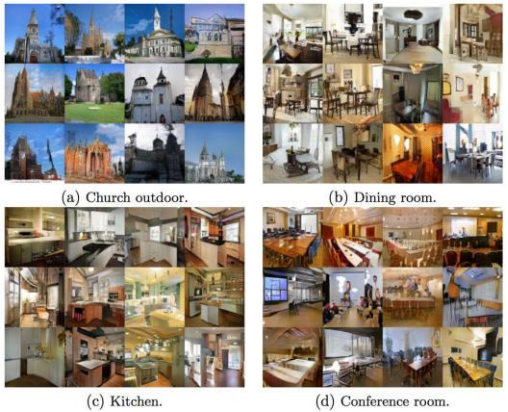


Woman with glasses



# 2017: Year of the GAN

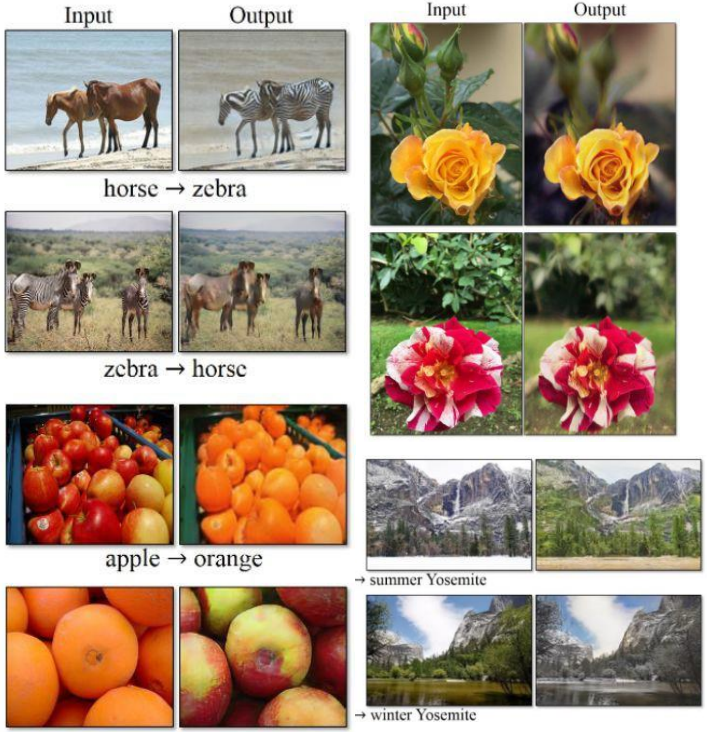
## Better training and generation




LSGAN. Mao et al. 2017.

BEGAN. Bertholet et al. 2017.

## Source->Target domain transfer



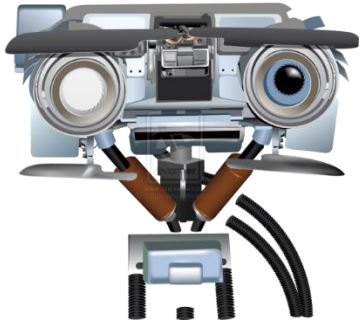

CycleGAN. Zhu et al. 2017.

## Text -> Image Synthesis


Reed et al. 2017.

## Many GAN applications

Pix2pix. Isola 2017. Many examples at https://phillipi.github.io/pix2pix/



# Domain Adaptation

# Has deep learning solved vision?

pedestrian detection FAIL

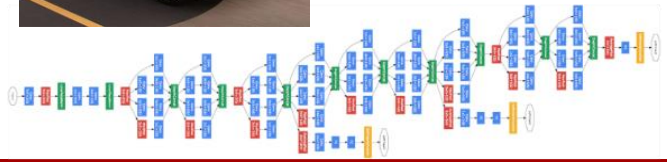
https://www.youtube.com/watch?v=w2pwxv8rFkU

# “What you saw is not what you get”

What your net is trained on

What it’s asked to label

**“Dataset Bias”**
**“Domain Shift”**



# Problem: Domain Shift

Input Image


True Segmentation

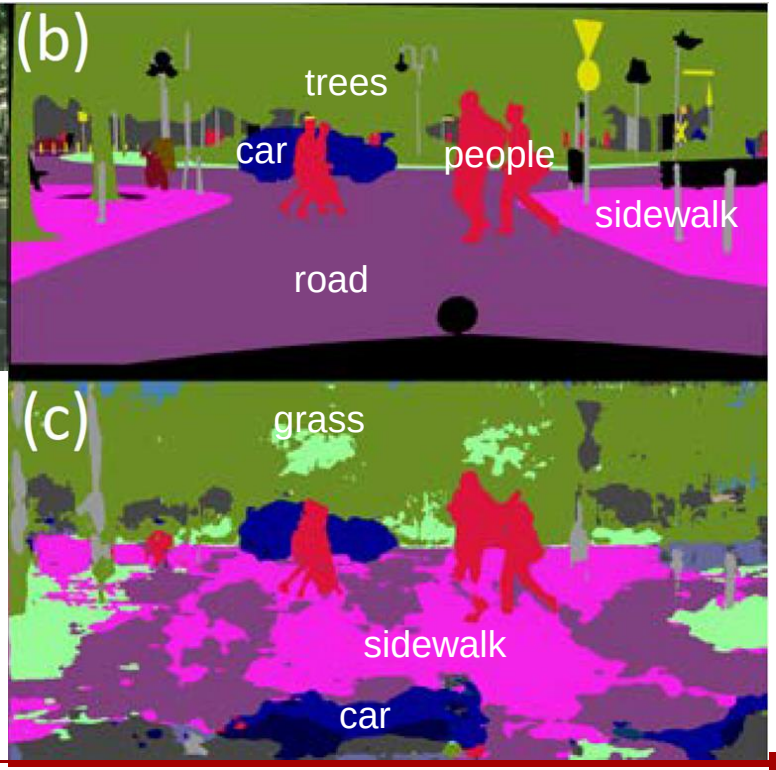
Model Output

# Solution: Domain Adaptation

Input Image

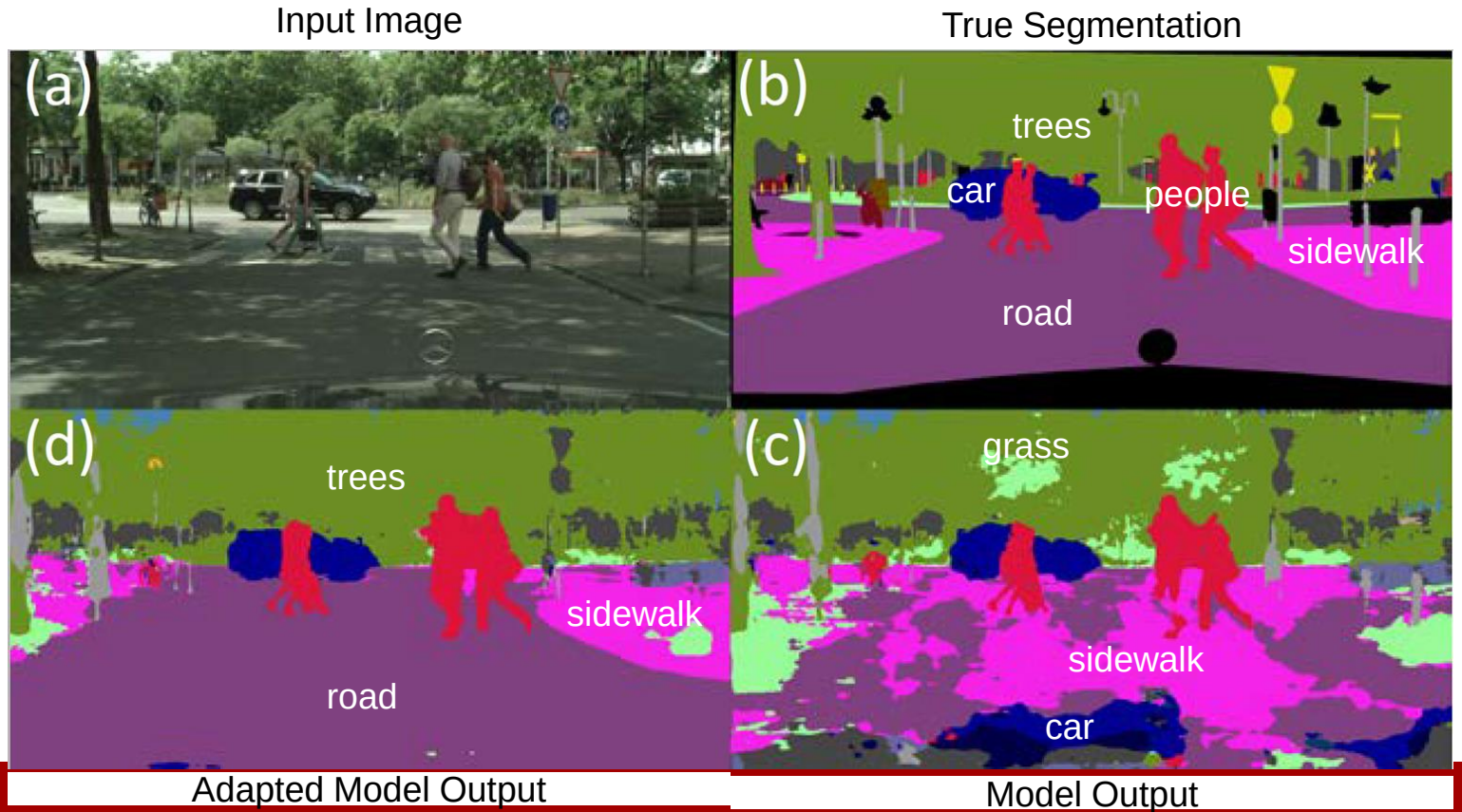

Adapted Model Output

True Segmentation



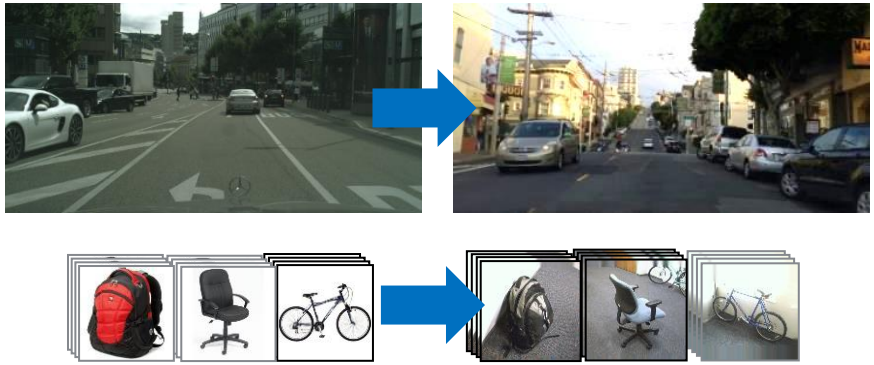

Model Output

# Solution: Domain Adaptation

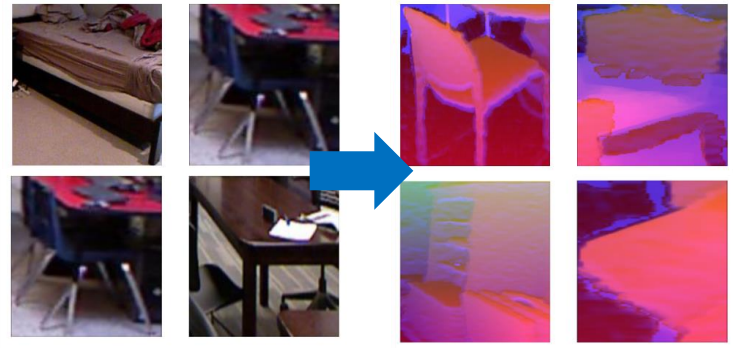


# Applications of Domain Adaptation

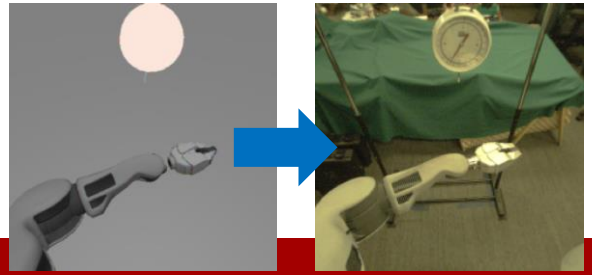
**From dataset to dataset**



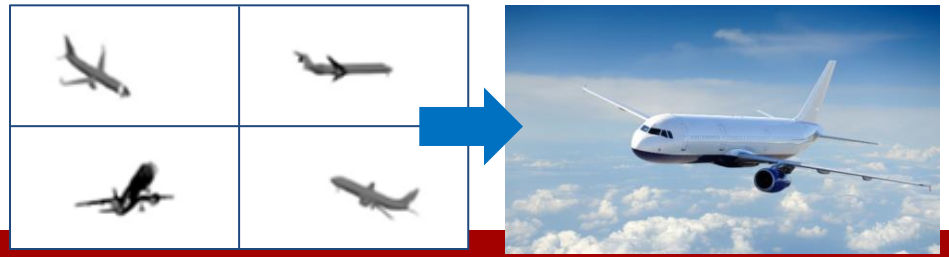
**From RGB to depth**



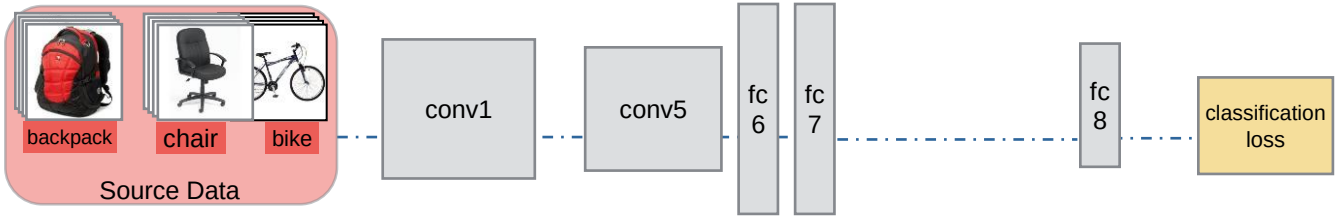
**From simulated to real control**



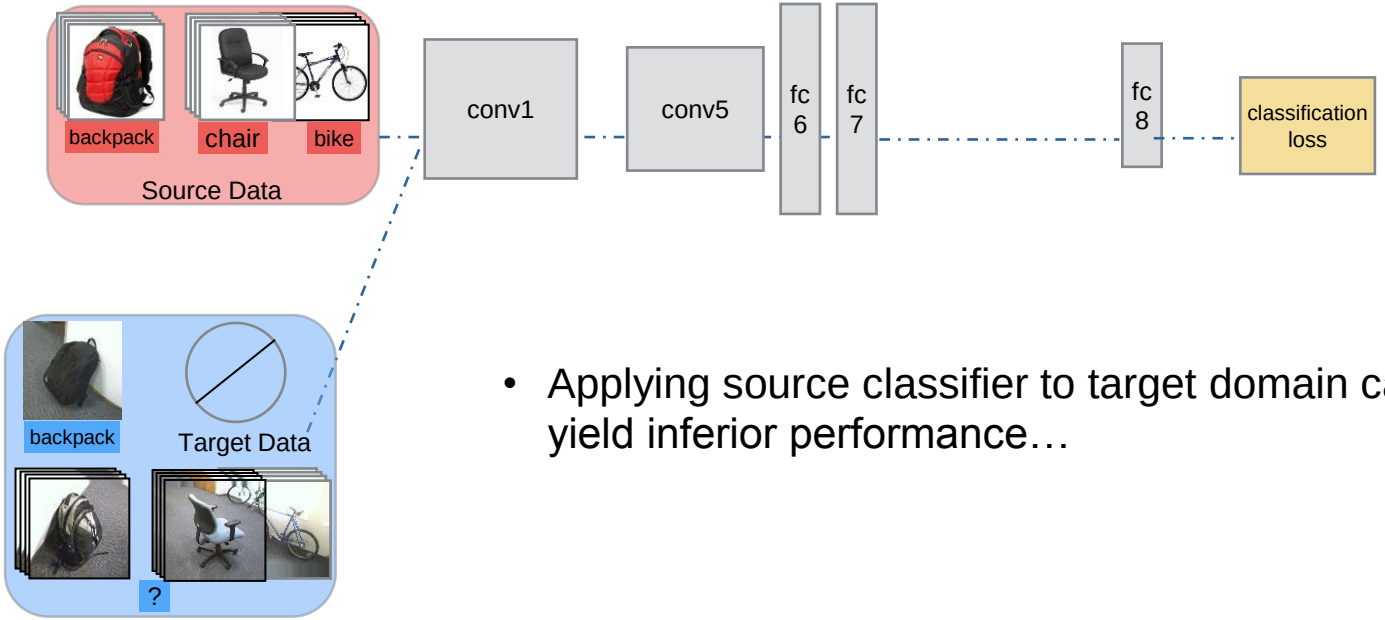
**From CAD models to real images**



# How to adapt a deep network?

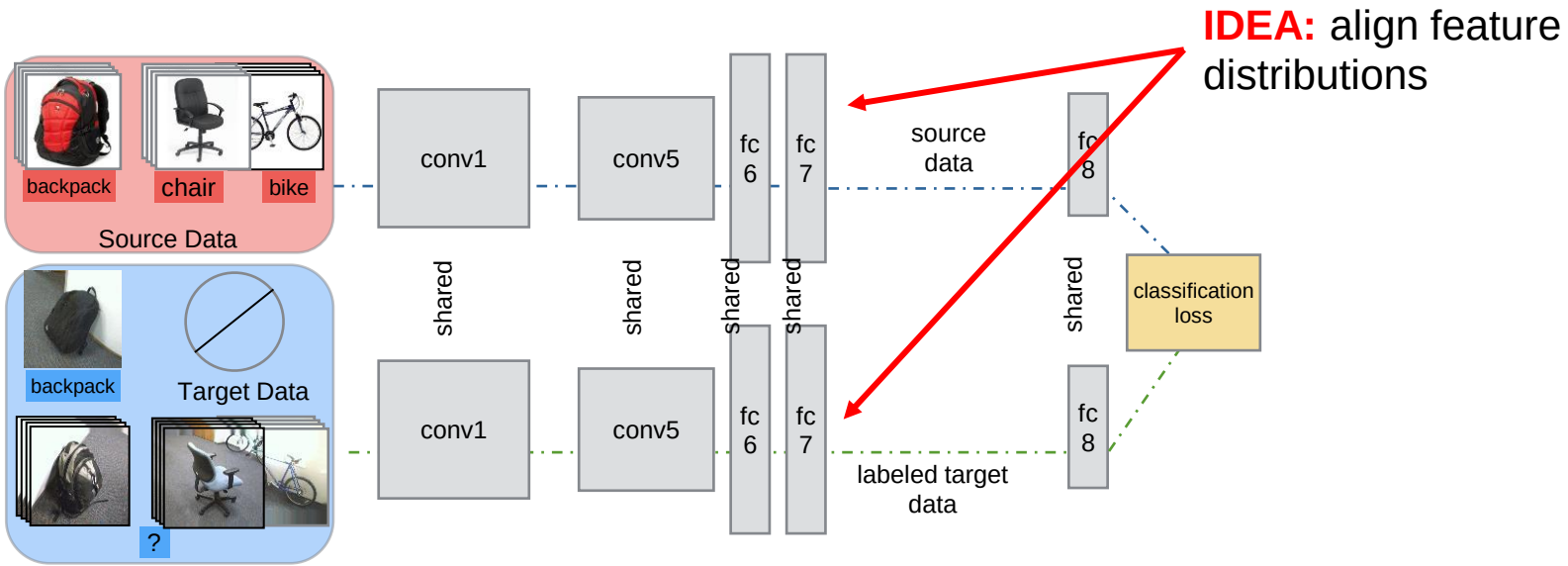


# How to adapt a deep network?

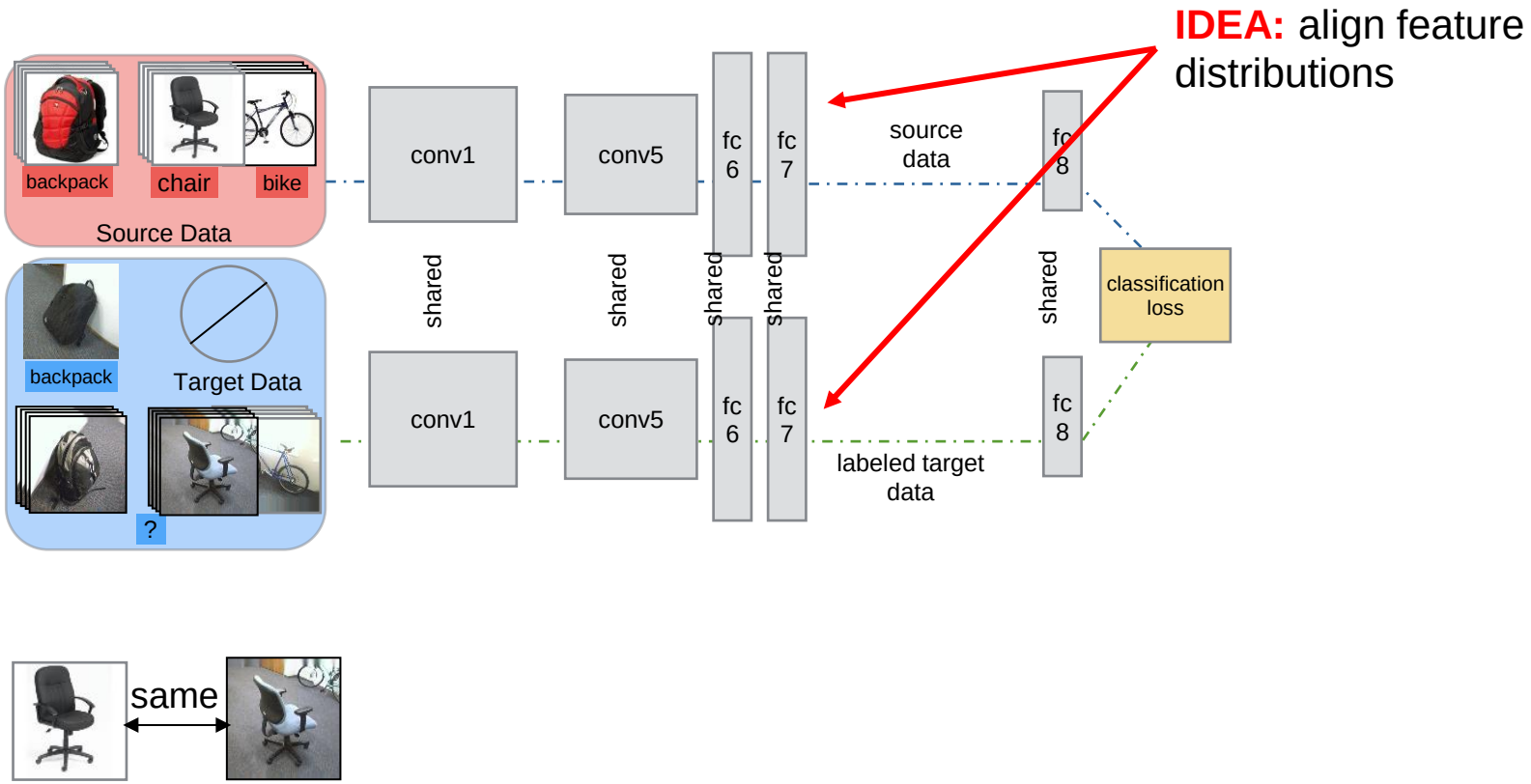



- Applying source classifier to target domain can yield inferior performance…

# How to adapt a deep network?

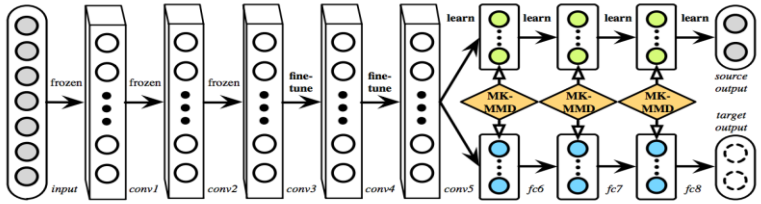


# How to adapt a deep network?

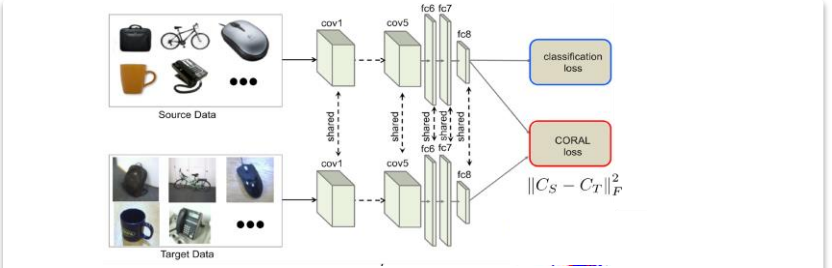


# Solution: align deep feature distributions

- by minimizing **distance** between distributions, e.g.

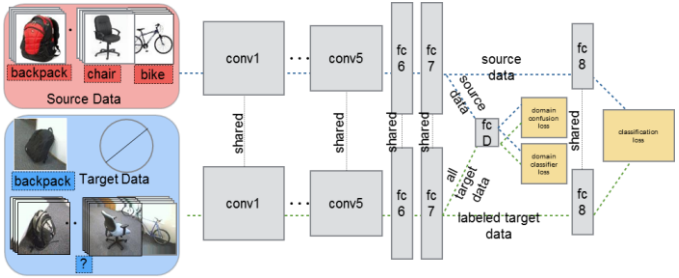



Maximum Mean Discrepancy M. Long, et al. ICML 2015

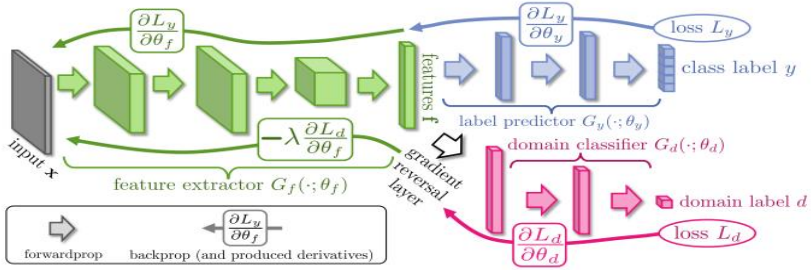



CORrelation ALignment Sun and Saenko, AAAI 2016

- …or by **adversarial** domain alignment, e.g.




Domain Confusion E. Tzeng et al. ICCV 2015



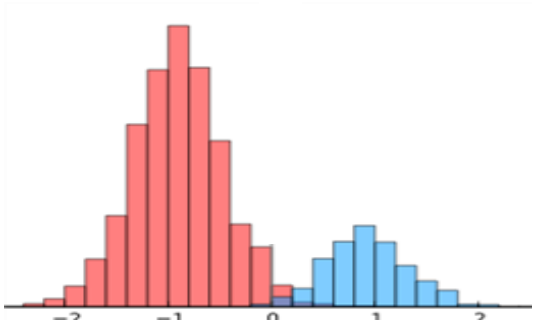

Reverse Gradient Y. Ganin and V. Lempitsky ICML 2015

# Adversarial Feature Alignment

# Adversarial networks

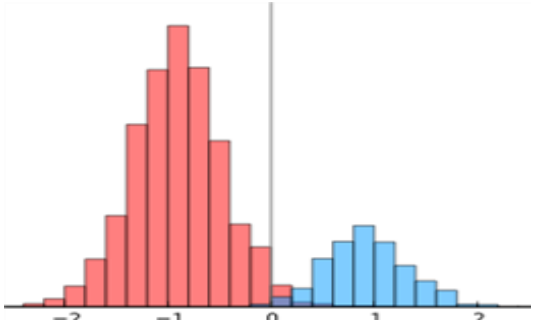
*Encoder*
*P*

*Reference*
*Q*



*P*

*Q*



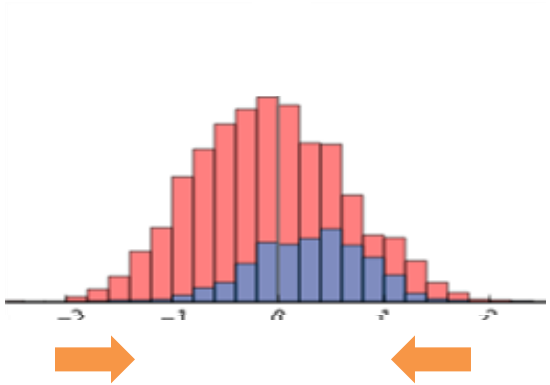
**Encoder**
**Generates features** such that their distribution P matches reference distribution Q

**Adversary**
**Tries to discriminate** between samples from P and samples from Q

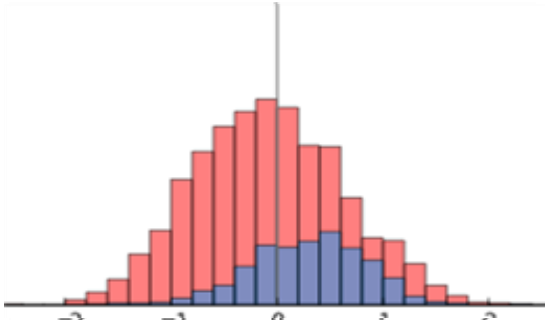
# Adversarial networks

*Encoder* *P*

*Reference* *Q*



*P* *Q*



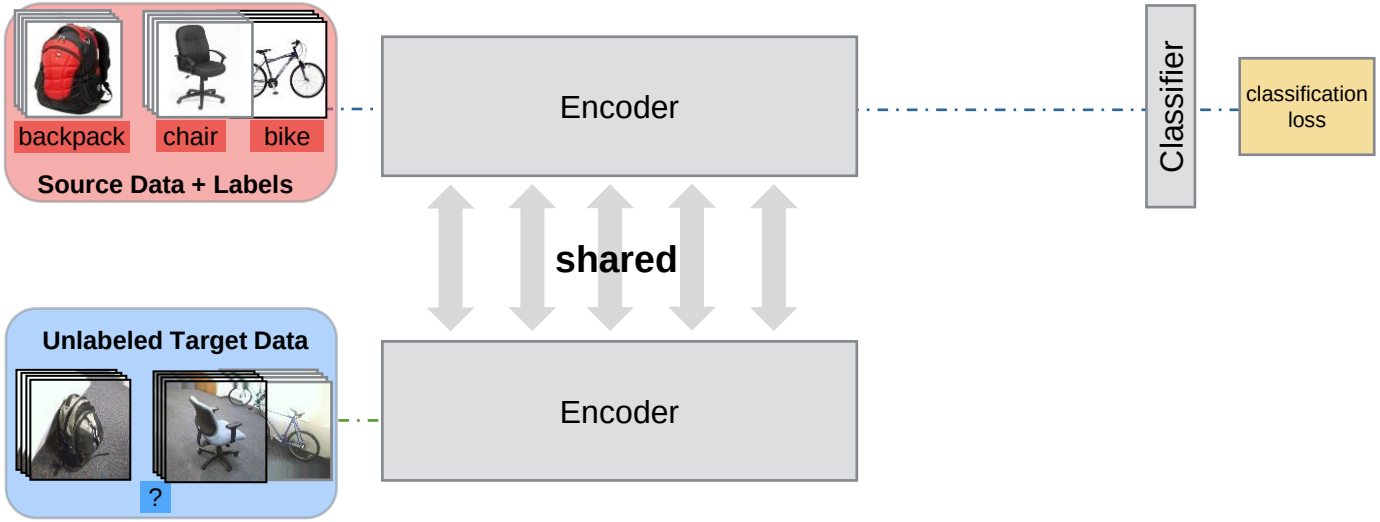
**Encoder**
Generates features such that their distribution P matches reference distribution Q

***fools adversary***

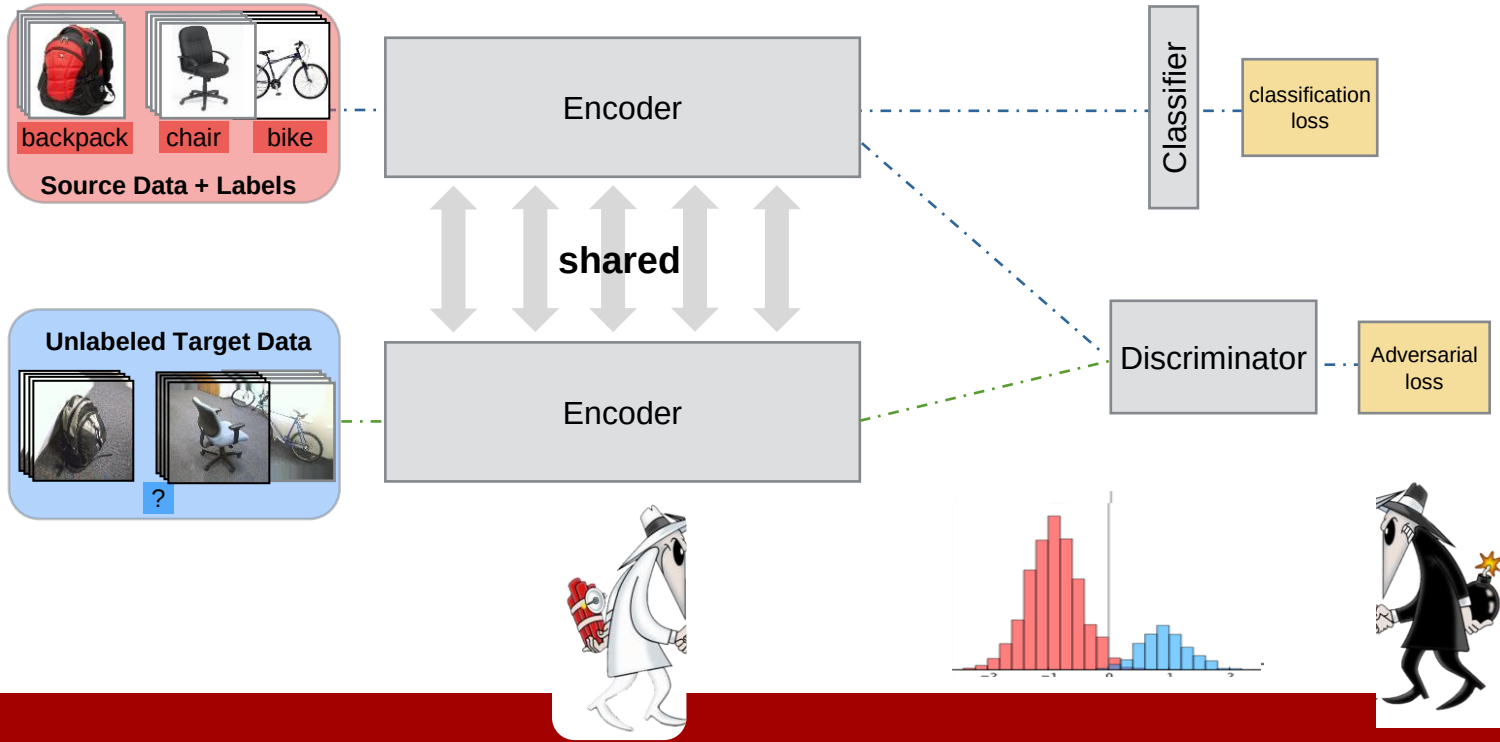
**Adversary**
Tries to discriminate between samples from P and samples from Q

***tries harder***

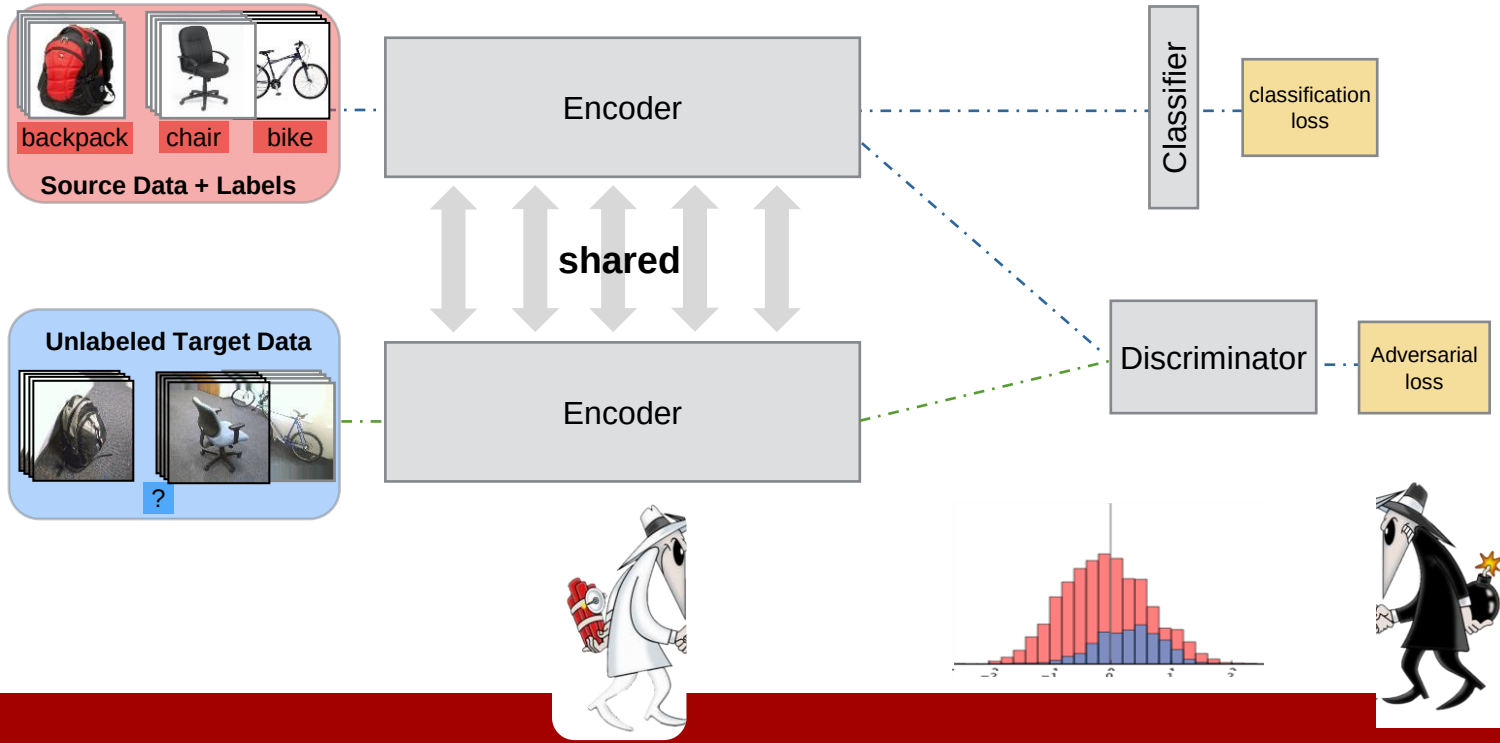
# Adversarial domain adaptation



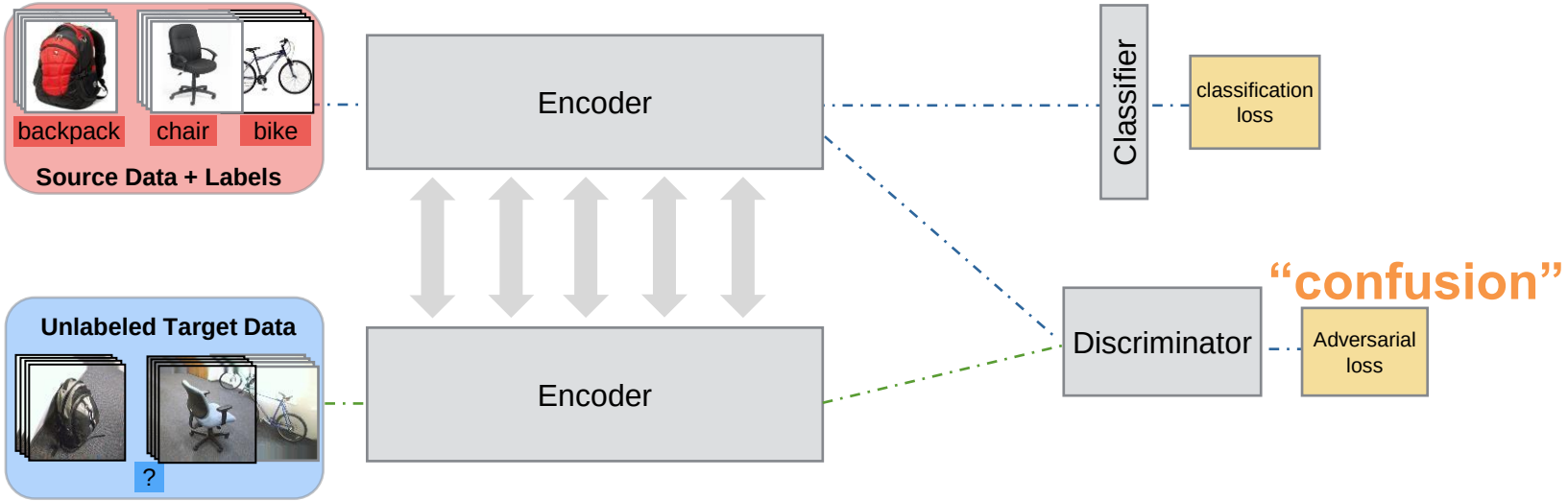
# Adversarial domain adaptation



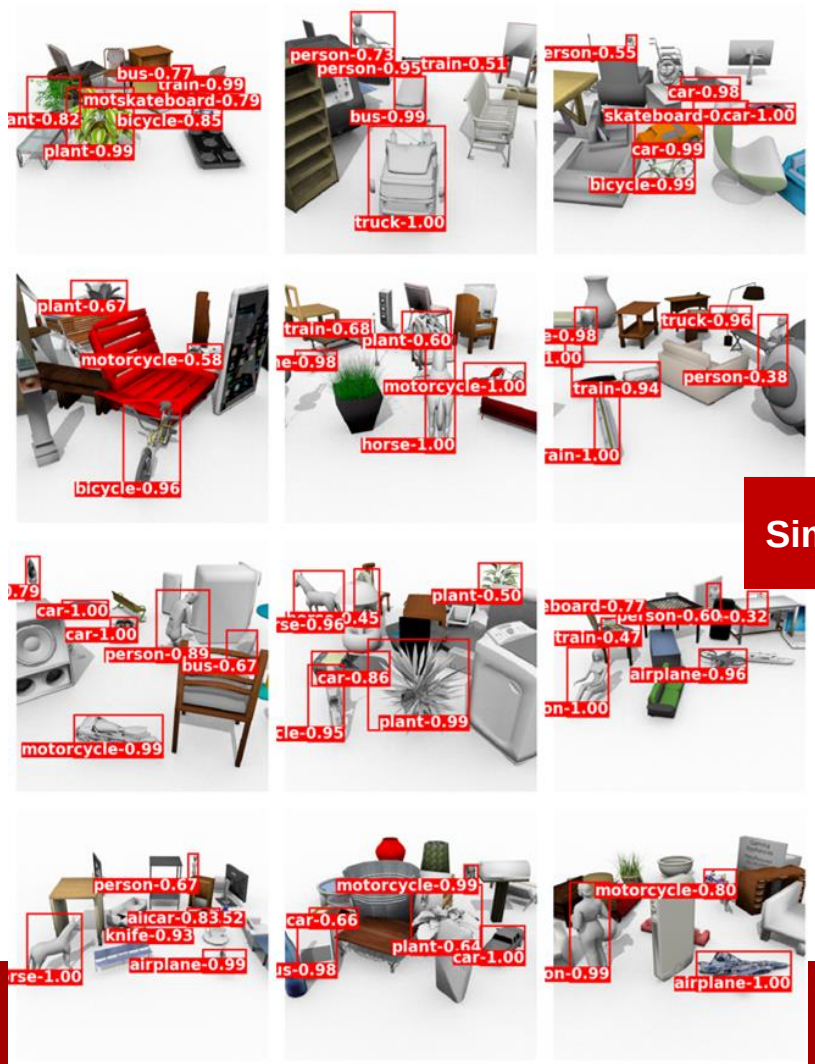
# Adversarial domain adaptation

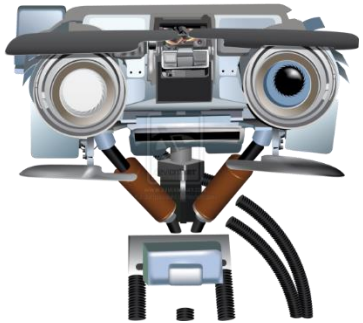


# Design choices in adversarial adaptation




[13] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks, NIPS 2016


bus-0.77
train-0.99
motskateboard-0.79
bicycle-0.85
plant-0.99
person-0.73
person-0.95
train-0.51
bus-0.99
truck-1.00
car-0.98
skateboard-0
car-1.00
car-0.99
bicycle-0.99
plant-0.67
motorcycle-0.58
bicycle-0.96
train-0.68
plant-0.60
motorcycle-1.00
horse-1.00
truck-0.96
person-0.38
train-0.94
car-1.00
car-1.00
person-0.89
bus-0.67
motorcycle-0.99
plant-0.50
car-0.86
plant-0.99
airplane-0.96
person-0.67
knife-0.93
airplane-0.99
motorcycle-0.99
car-0.66
plant-0.64
car-1.00
motorcycle-0.80
airplane-1.00
Sim 2 Real
person
person
plant
person
motorcycle
aeroplane
horse
horse
horse
person
horse
skateboard
person
truck
plant
train
aeroplane

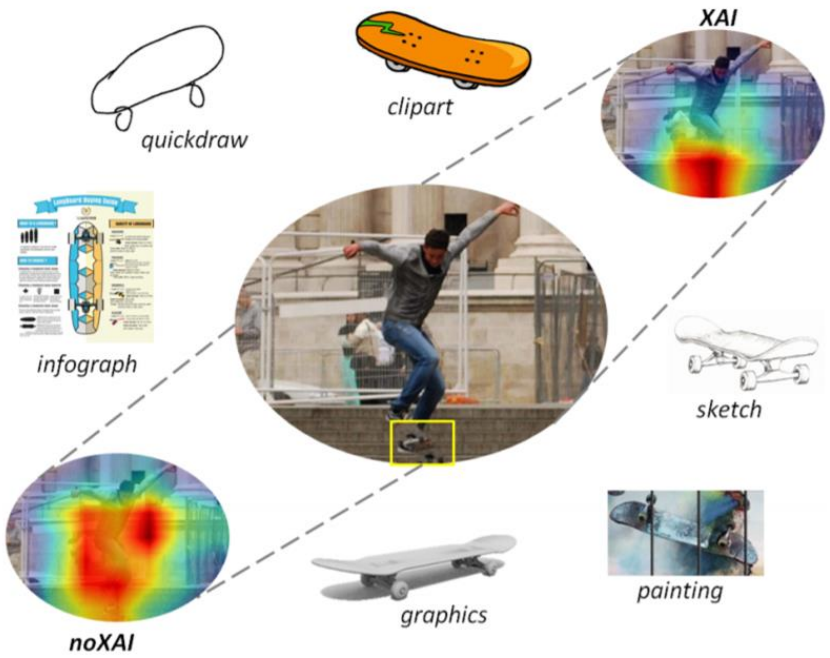


# Explainability and Domain Generalization

# Explainable AI (XAI) for Domain Generalization

- Training a deep neural network model to enforce explainability:
  *e.g. focusing on the skateboard region (red is most salient, and blue is least salient) for the ground-truth class skateboard in the central training image.*

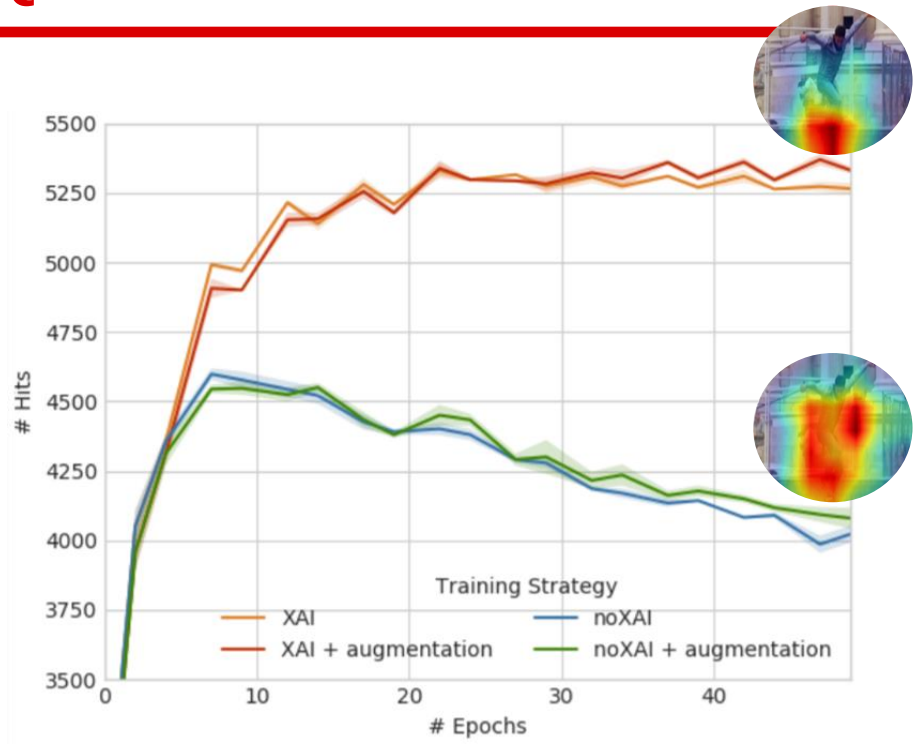
- This enables improved generalization to other domains where the background is not necessarily class-informative.



# Explainability Results: Quantitative *[Automated]*

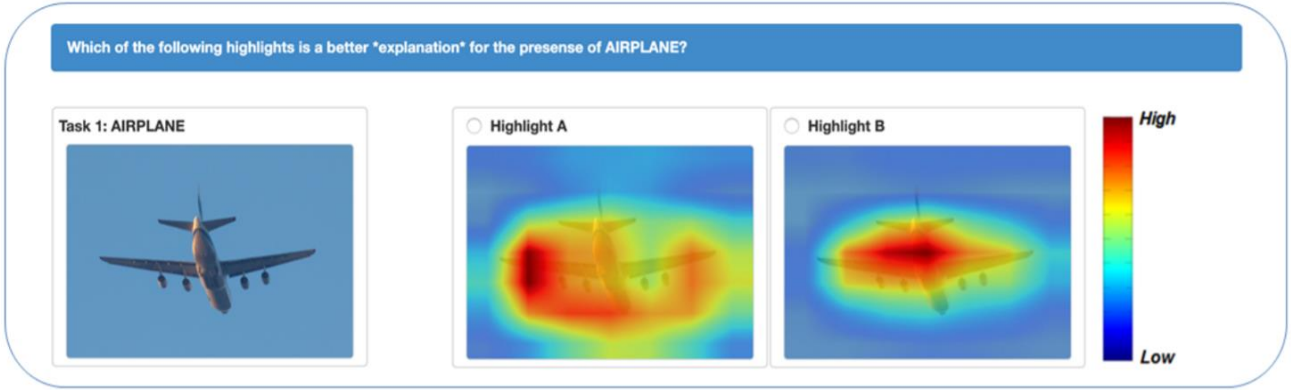
- The number of unseen MSCOCO images, among the 16K validation set, where the model is able to provide an accurate explanation for, among the correctly classified ones during training.

- We can see that the noXAI model fits the dataset bias at training time, while the XAI model improves its explainability over time for validation data.



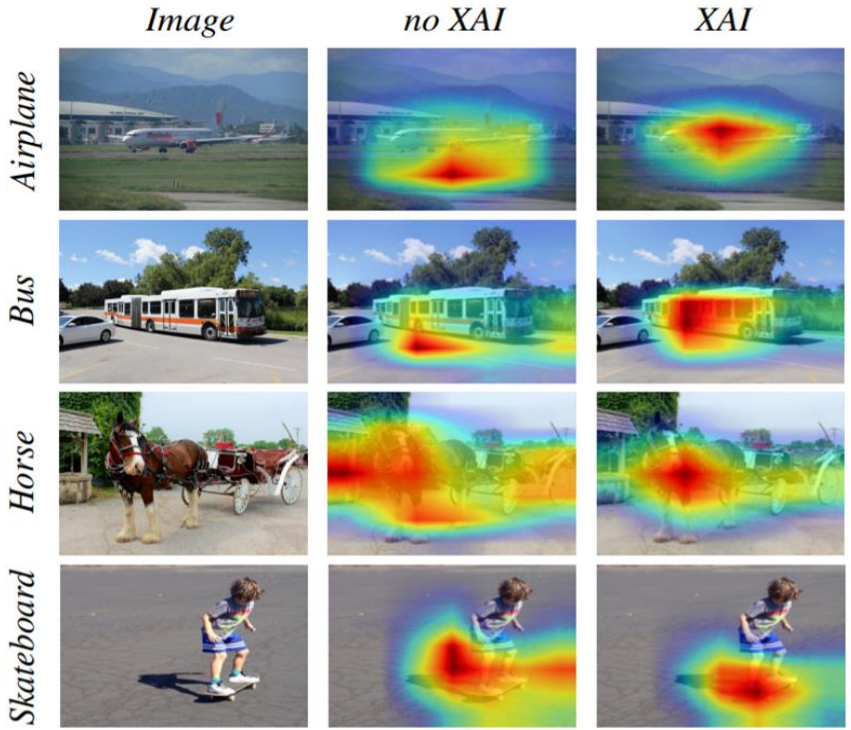
# Explainability Results: Quantitative *[Human Judgment]*

- The interface asks the users to select the evidence (“highlight”) they think is a better explanation for the presence of an object.
- 80% of the images with a winner choice favored the XAI explanation over the noXAI explanation.



# Explainability Results: Qualitative

- The XAI model, based on human spatial annotations, provides feedback that enables saliency to be better localized over the objects corresponding to the ground-truth class compared to the noXAI vanilla training of a deep model, for unseen validation data.



# Domain Adaptation and Generalization

- In *domain adaptation* one needs to know a priori the target distribution, which may not be available in practice.
- In standard *domain generalization* techniques, one needs several source domains for training, both of which may not be available in practice.
- A more generic formulation is *single-source domain generalization*, where one would like to avoid learning dataset bias for better generalization, but only has access to a single source distribution.

# Single-Source Domain Generalization Results

- Domain generalization on six *unseen* target domains from the Syn2Real and DomainNet datasets.

- Training has been conducted on a single source: the MSCOCO dataset, and no data from any of the target domains is used for training.