Generative Models

What we will learn today

- ❑ Autoencoder
- ❑ Variational Autoencoder
- ❑ Generative Adversarial Network

references

Supervised vs. Unsupervised Learning

Supervised Learning

Data: (x, y)

x is data, y is label

Data Labeling is needed

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

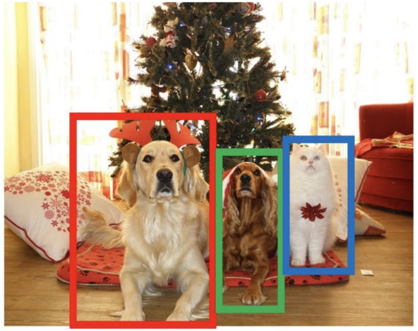
$$y = f(x)$$

Examples: classification, regression, object detection, semantic segmentation, image segmentation etc.

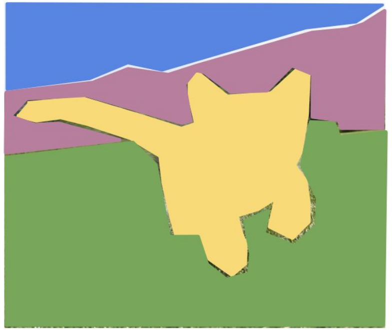
Classification: Input x

y = 8

Object Detection



Semantic Segmentation



Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Supervised vs. Unsupervised Learning

Supervised Learning

Data: (x,y)

x is data, y is label

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

y = $f(x)$

Examples: classification, regression, object detection, semantic segmentation, image segmentation etc.

Unsupervised Learning

Data: x

Just data, No labels!

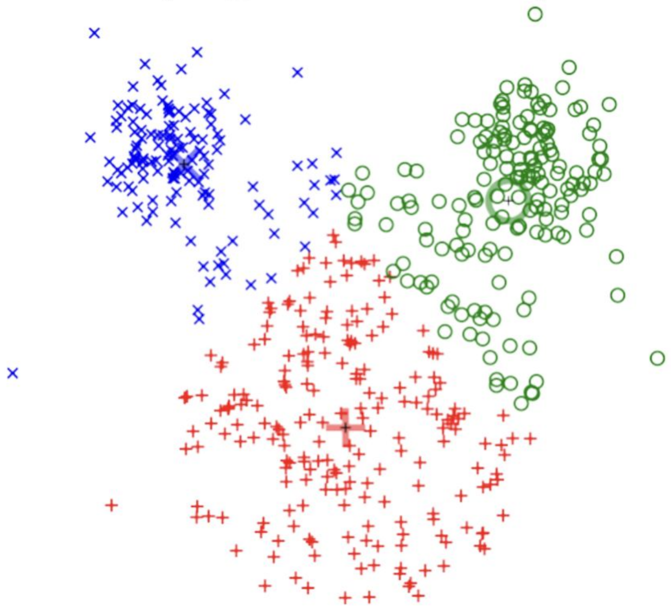
Goal: Learn some underlying hidden *structure* of the data

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

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Supervised vs. Unsupervised Learning

Clustering
(e.g. k-Means)



Unsupervised Learning

Data: x

Just data, No labels!

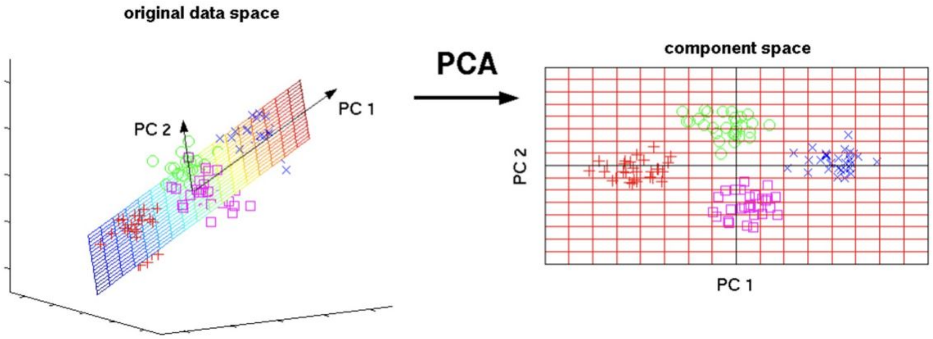
Goal: Learn some underlying hidden *structure* of the data

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

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Supervised vs. Unsupervised Learning

Dimensionality Reduction
(e.g. Principal Component Analysis- PCA)



3D → 2D

Unsupervised Learning

Data: x

Just data, No labels!

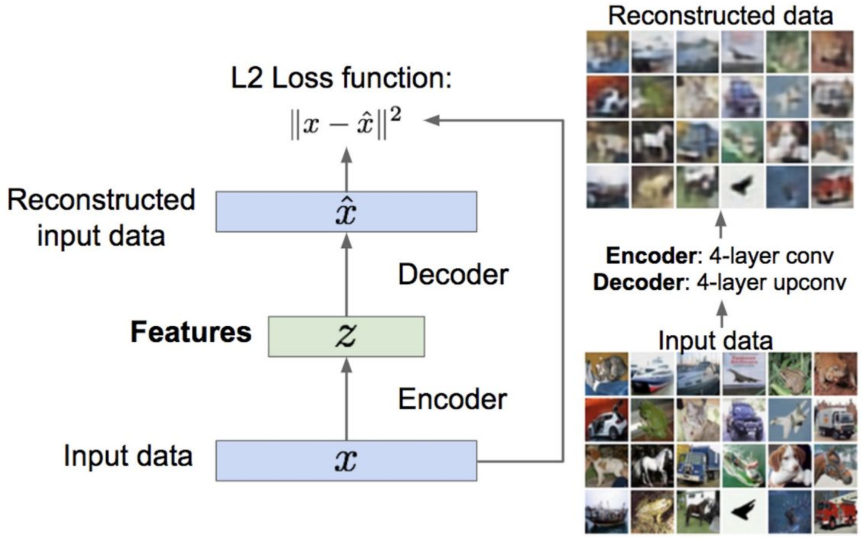
Goal: Learn some underlying hidden *structure* of the data

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

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Supervised vs. Unsupervised Learning

Feature Learning
(e.g. Autoencoders)



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.

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Discriminative vs. Generative Models

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

Generative Model: Learn a probability distribution $P(x)$

Conditional Generative Model: Learn a probability distribution $P(x|y)$

Data: x

Label: y = Cat

Probability Recap:

Density Function

$P(x)$ assigns a positive number to each possible x; higher numbers mean x is more likely

Density functions are **normalized**:

$$\int_x P(x)dx = 1$$

Different values of x **complete** for density

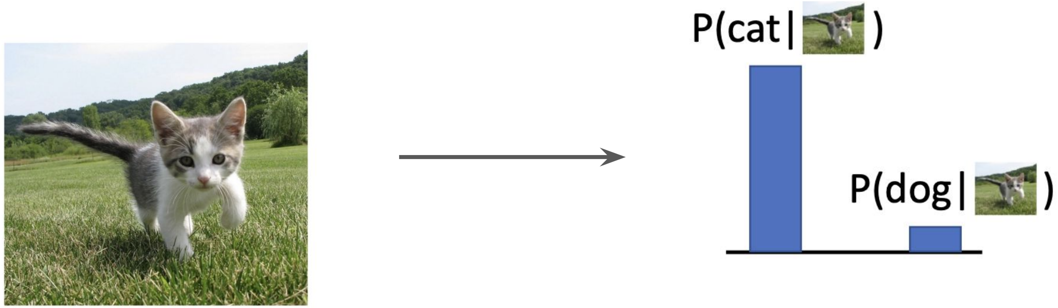
Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Discriminative vs. Generative Models

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

Generative Model: Learn a probability distribution $P(x)$

Conditional Generative Model: Learn a probability distribution $P(x|y)$



Density Function

$P(x)$ assigns a positive number to each possible x; higher numbers mean x is more likely

Density functions are **normalized**:

$$\int_x P(x)dx = 1$$

Different values of x **complete** for density

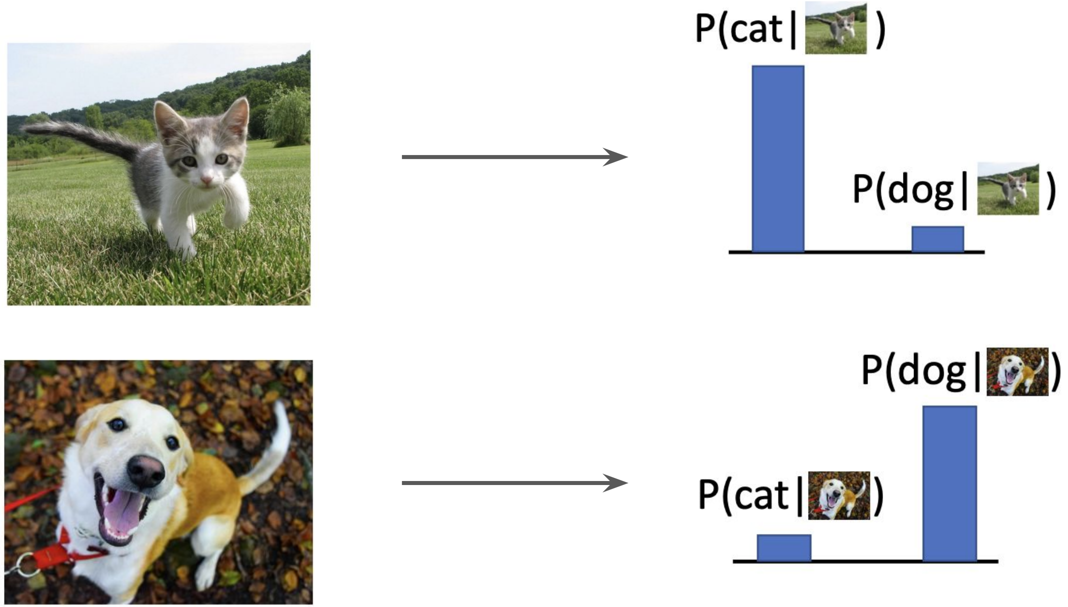
Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Discriminative vs. Generative Models

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

Generative Model: Learn a probability distribution $P(x)$

Conditional Generative Model: Learn a probability distribution $P(x|y)$

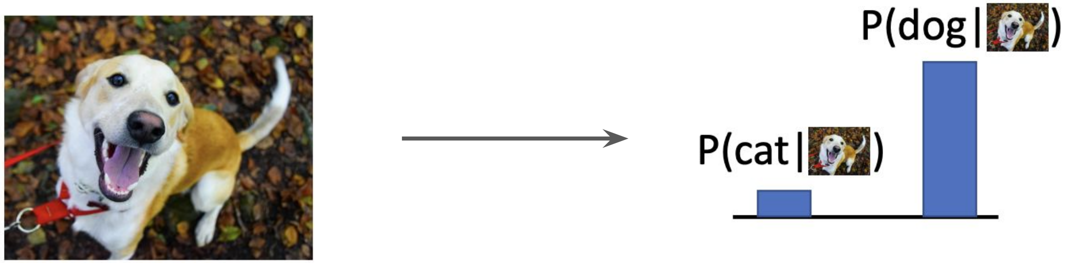


Discriminative model: the possible labels for each input "compete" for probability mass. But no competition between **images**

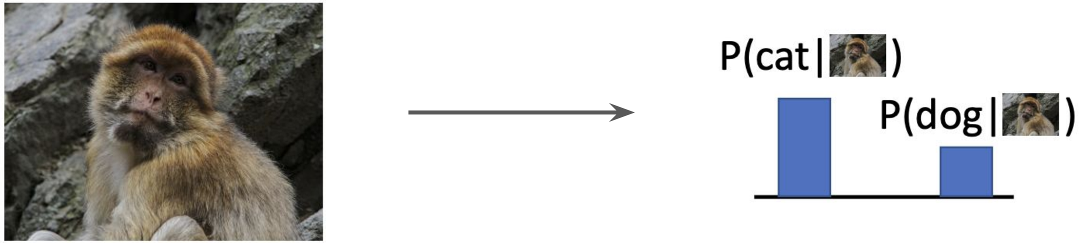
Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Discriminative vs. Generative Models

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



Generative Model: Learn a probability distribution $P(x)$



Conditional Generative Model: Learn a probability distribution $P(x|y)$

Discriminative model: No way for the model to handle unreasonable inputs; it must give label distributions for all images

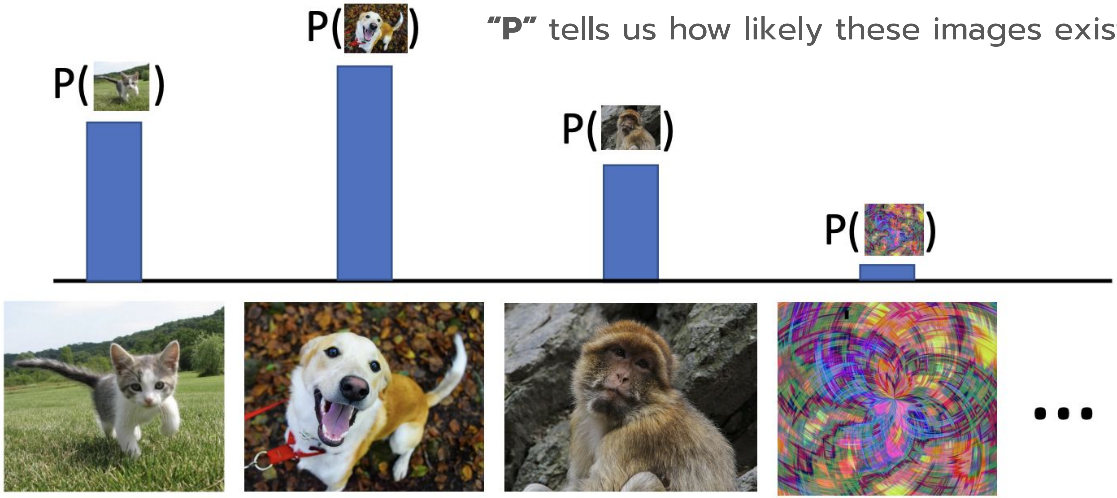
Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Discriminative vs. Generative Models

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

Generative Model: Learn a probability distribution $P(x)$

Conditional Generative Model: Learn a probability distribution $P(x|y)$



"P" tells us how likely these images exist

Generative model: All possible images compete with each other for probability mass

Requires deep image understanding! Is a dog more likely to sit or stand? How about 3-legged dog vs 3-armed monkey?

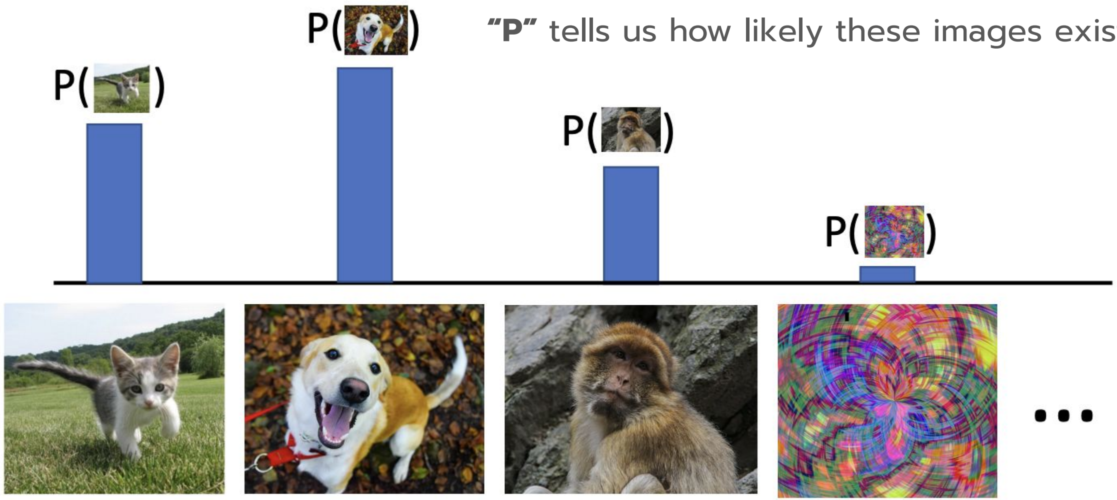
Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Discriminative vs. Generative Models

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

Generative Model: Learn a probability distribution $P(x)$

Conditional Generative Model: Learn a probability distribution $P(x|y)$



"P" tells us how likely these images exist

Generative model: All possible images compete with each other for probability mass

Model can "reject" unreasonable inputs by assigning them small values

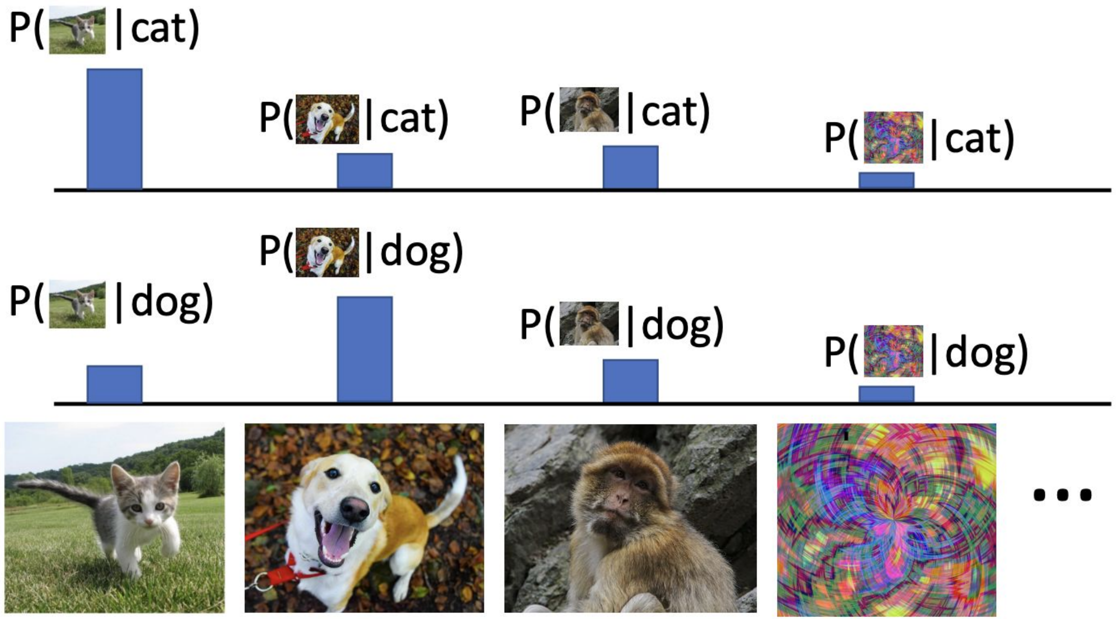
Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Discriminative vs. Generative Models

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

Generative Model: Learn a probability distribution $P(x)$

Conditional Generative Model: Learn a probability distribution $P(x|y)$



Conditional Generative Model: Each possible label induces a competition among all images

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Discriminative vs. Generative Models

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

Generative Model: Learn a probability distribution $P(x)$

Conditional Generative Model: Learn a probability distribution $P(x|y)$

These models may seem to be very distinct.

Although, they are actually **not fully** distinct.

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

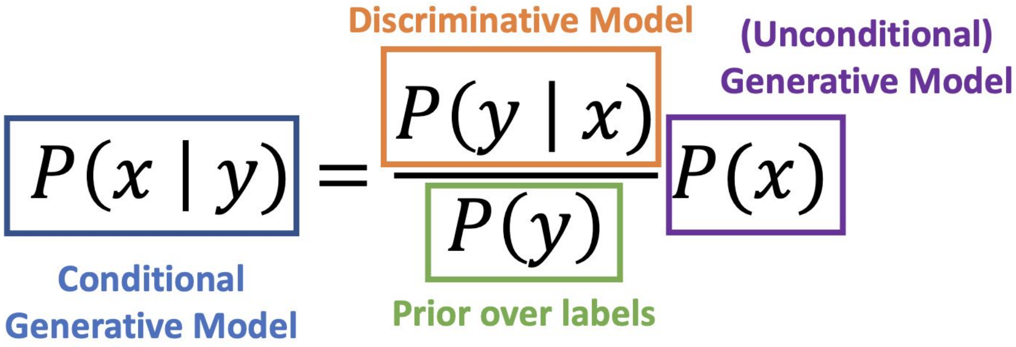
Discriminative vs. Generative Models

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

Generative Model: Learn a probability distribution $P(x)$

Conditional Generative Model: Learn a probability distribution $P(x|y)$

Recall **Bayes' Rule:**



$$P(x \mid y) = \frac{P(y \mid x)}{P(y)} P(x)$$

Discriminative Model

(Unconditional) Generative Model

Conditional Generative Model

Prior over labels

We can build a conditional generative model from other components!

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Discriminative vs. Generative Models

Discriminative Model: Learn a probability distribution $P(y|x)$ → Assign labels to data

Feature learning (with labels)

Generative Model: Learn a probability distribution $P(x)$ → Detect outliers

Feature learning (without labels)

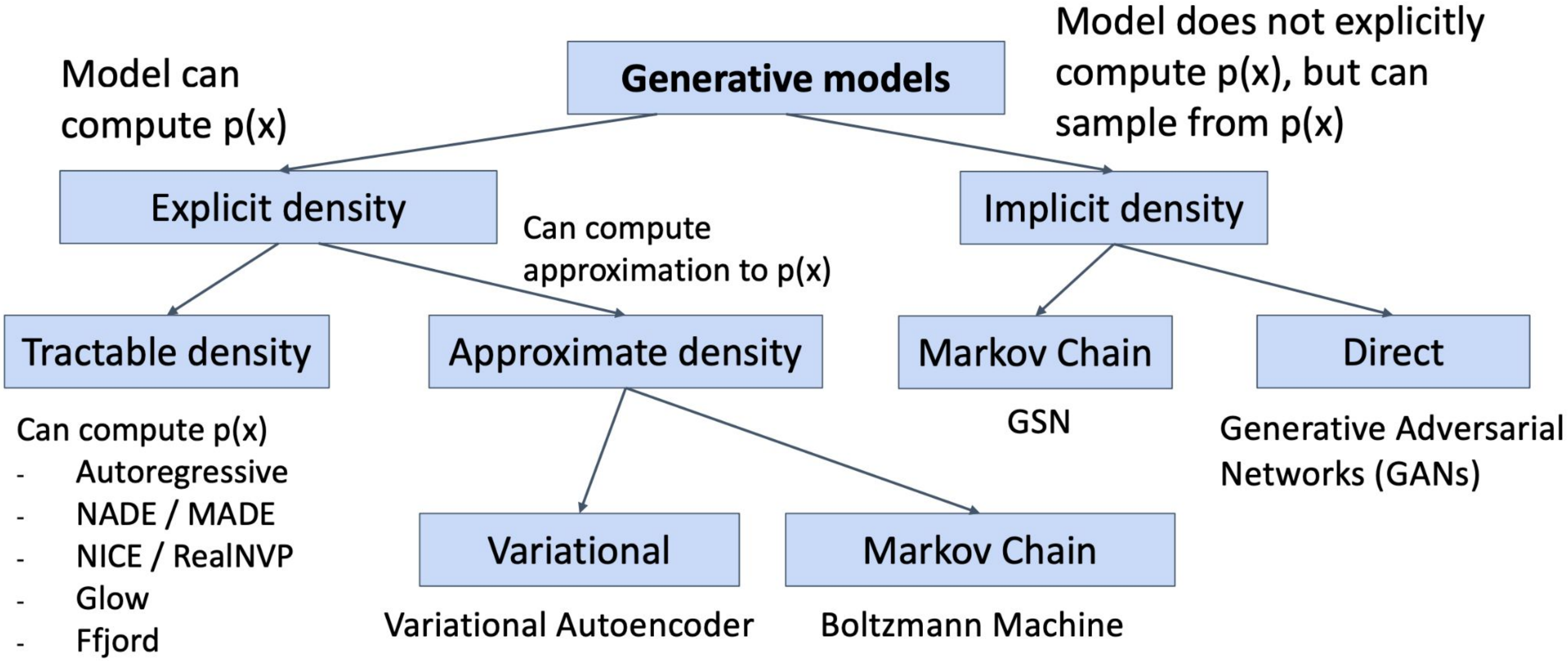
Sample to **generate** new data

Conditional Generative Model: Learn a probability distribution $P(x|y)$ → Assign labels, while rejecting outliers!

Generate new data conditioned on input labels

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Taxonomy of Generative Models



Model can compute p(x)

Generative models

Model does not explicitly compute p(x), but can sample from p(x)

Explicit density

Can compute approximation to p(x)

Implicit density

Tractable density

Approximate density

Markov Chain

Direct

Can compute p(x)

- Autoregressive
- NADE / MADE
- NICE / RealNVP
- Glow
- Ffjord

GSN

Generative Adversarial Networks (GANs)

Variational

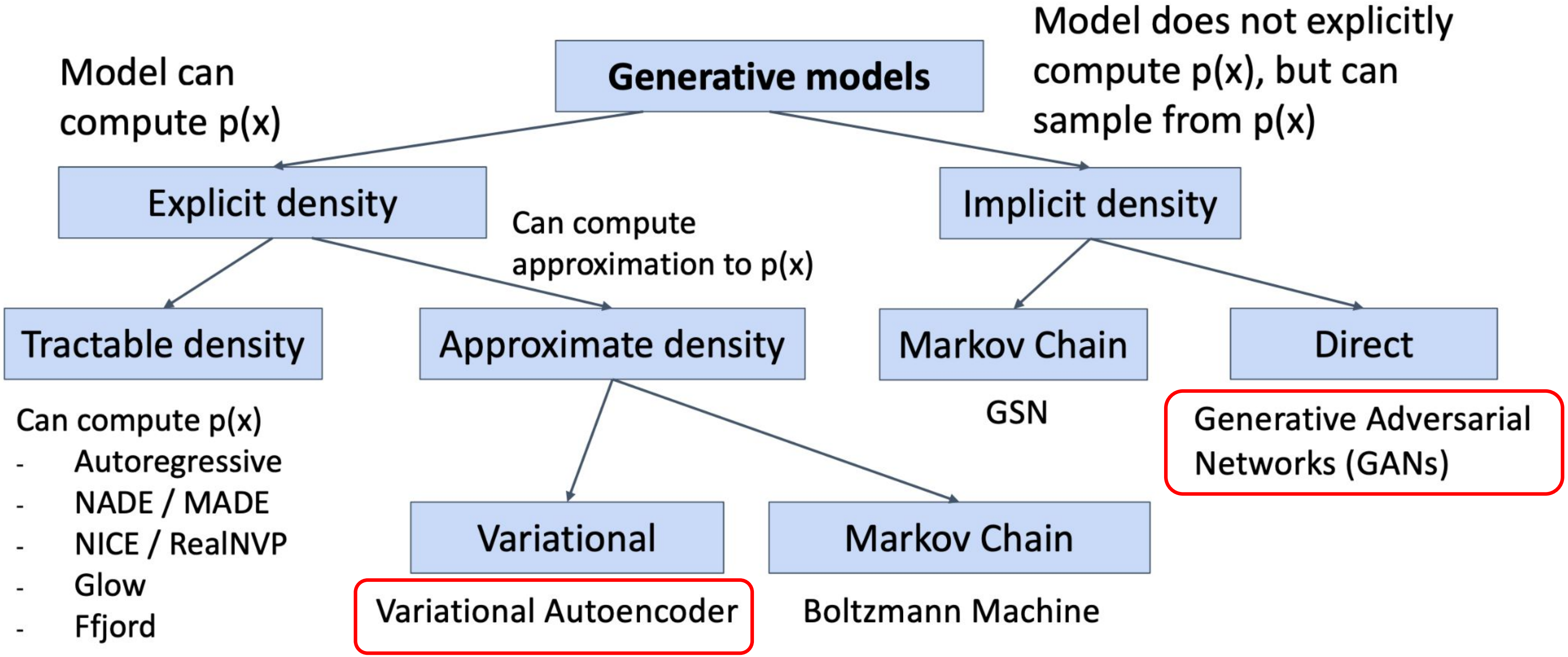
Markov Chain

Variational Autoencoder

Boltzmann Machine

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Taxonomy of Generative Models

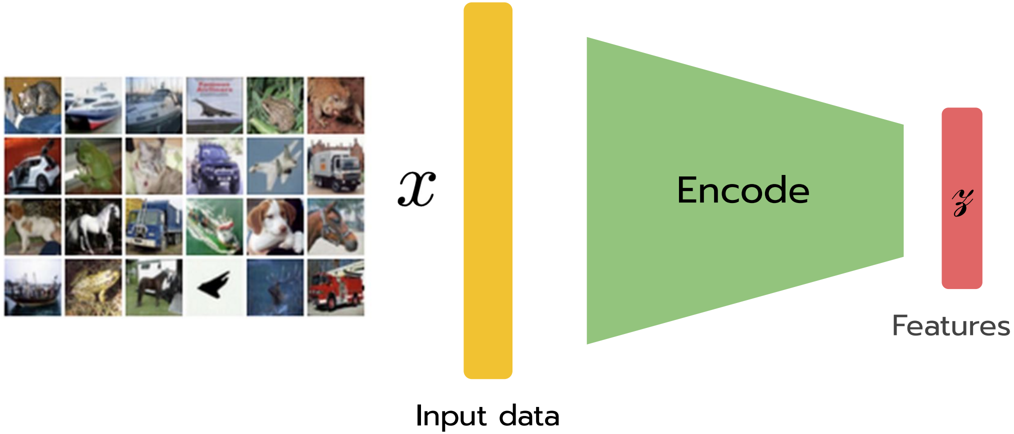


Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Variational AUTOENCODERS

Autoencoders (Regular, non-variational)

Unsupervised method for learning feature vectors from raw data x, without any labels.
"Autoencoding" = encoding itself



Features should extract useful information (maybe object identities, properties, scene type, etc) that we can use for downstream tasks

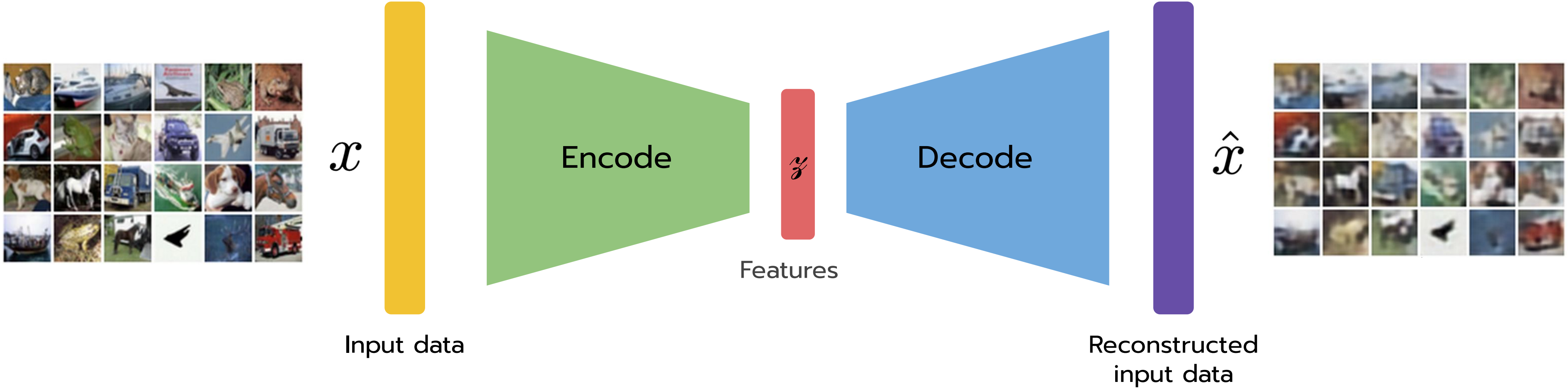
Problem: How can we **learn** this **feature transform** from raw data?

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Variational AUTOENCODERS

Autoencoders (Regular, non-variational)

Idea: Use the features to **reconstruct the input data** with a decoder
“Autoencoding” = encoding itself



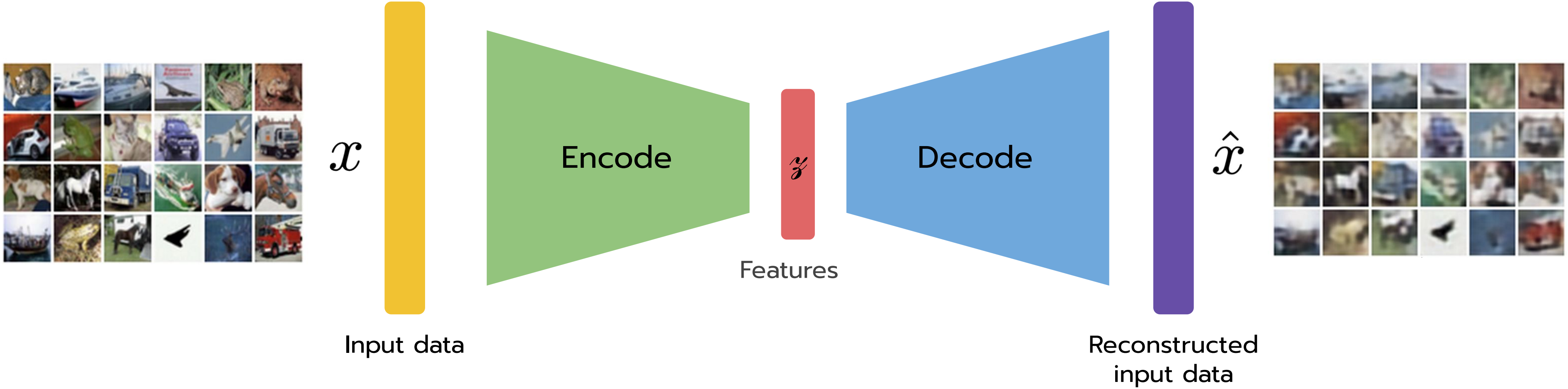
Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Variational AUTOENCODERS

Autoencoders (Regular, non-variational)

Idea: Use the features to **reconstruct the input data** with a decoder

“Autoencoding” = encoding itself



“Encoder” learns mapping from the data, $\mathscr{X}$, to a low-dimensional latent space, $\mathscr{z}$

“Decoder” learns mapping back from latent space, $\mathscr{z}$, to reconstructed observation, $\hat{x}$

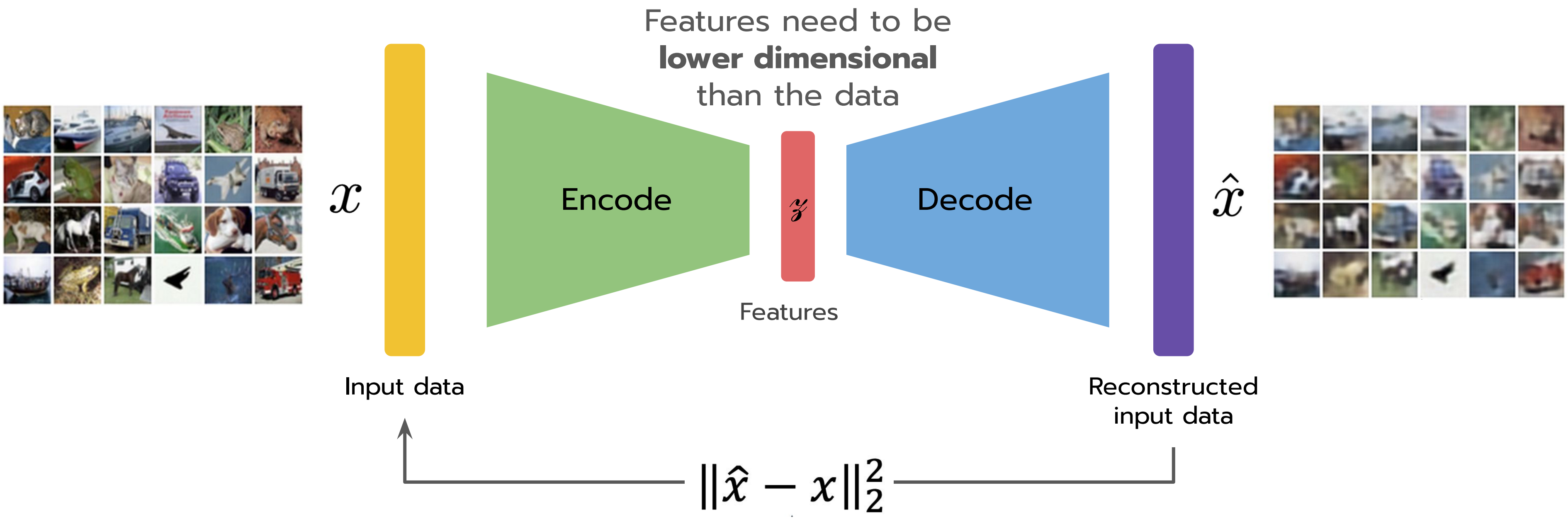
Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Variational AUTOENCODERS

Autoencoders (Regular, non-variational)

Loss: L2 distance between input and reconstructed data.

Does not use any labels! Just raw data!



Features need to be **lower dimensional** than the data

$$||\hat{x} - x||_2^2$$

Loss function

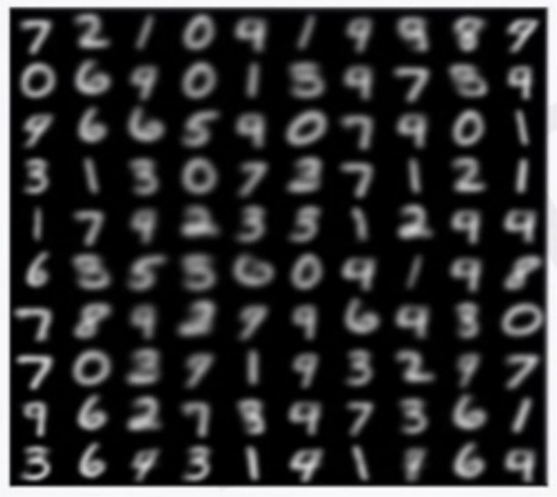
Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Variational AUTOENCODERS

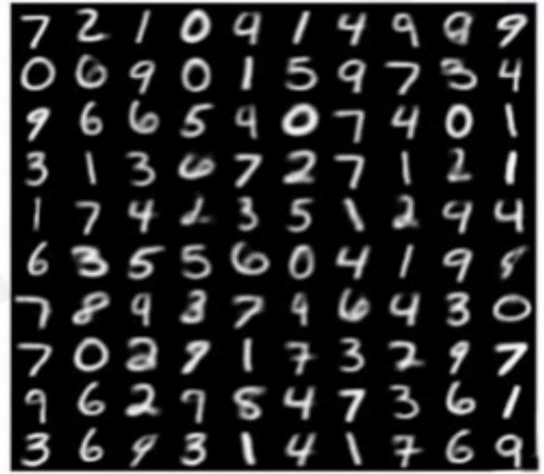
Autoencoders (Regular, non-variational)

Autoencoding is a form of compression! **Size of latent vector** affects the reconstructed data quality.

2D latent space



5D latent space



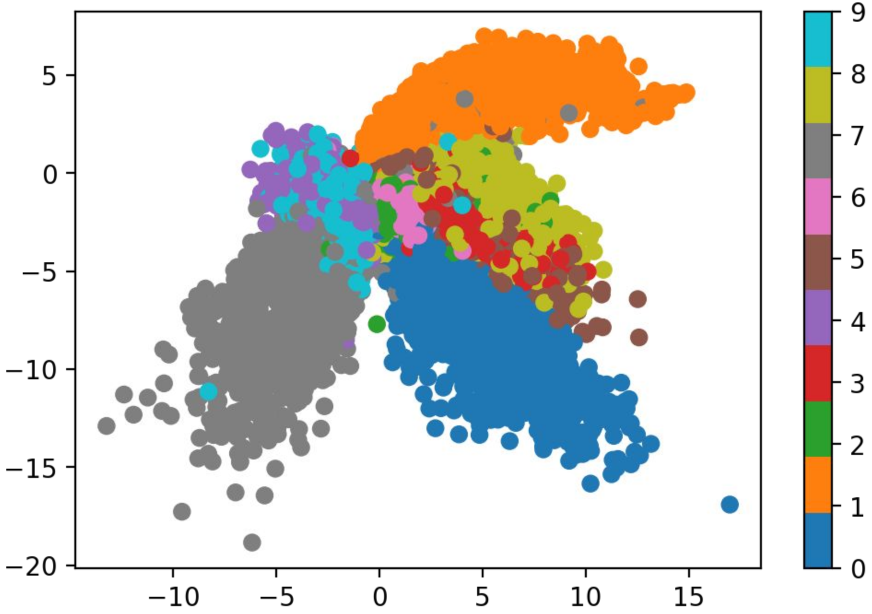
Ground truth

Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Variational AUTOENCODERS

Autoencoders (Regular, non-variational)

Latent Space of MNIST dataset

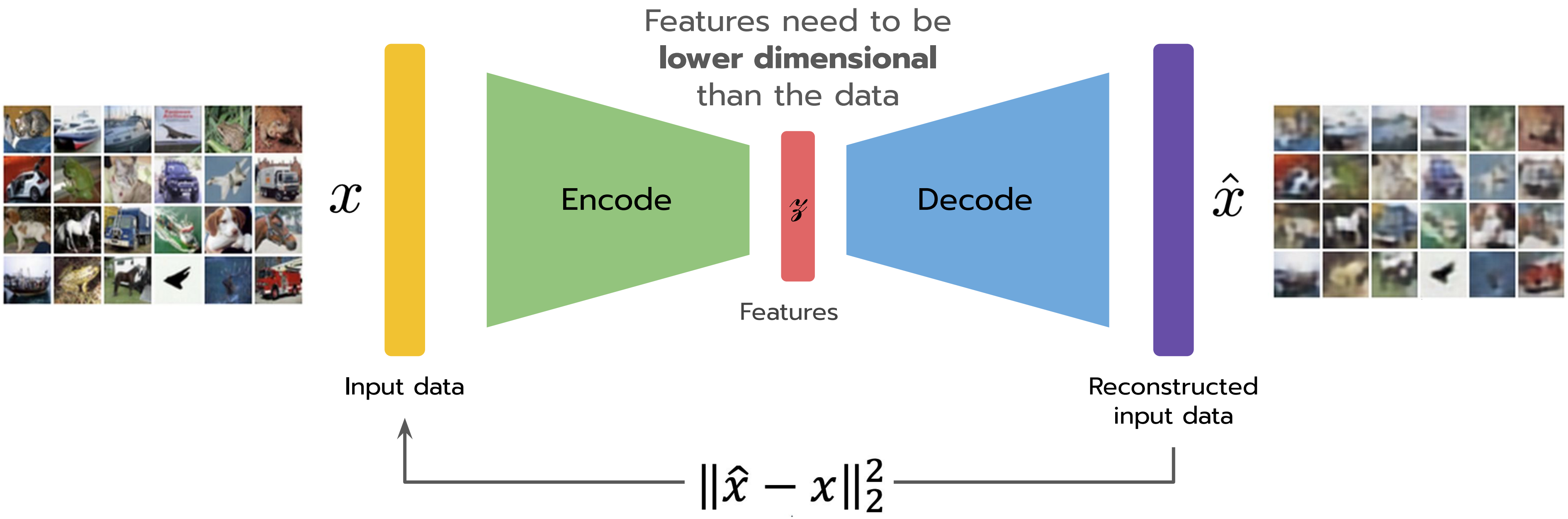


Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Variational AUTOENCODERS

Autoencoders (Regular, non-variational)

After training, **throw away decoder** and use encoder for a downstream task



Features need to be **lower dimensional** than the data

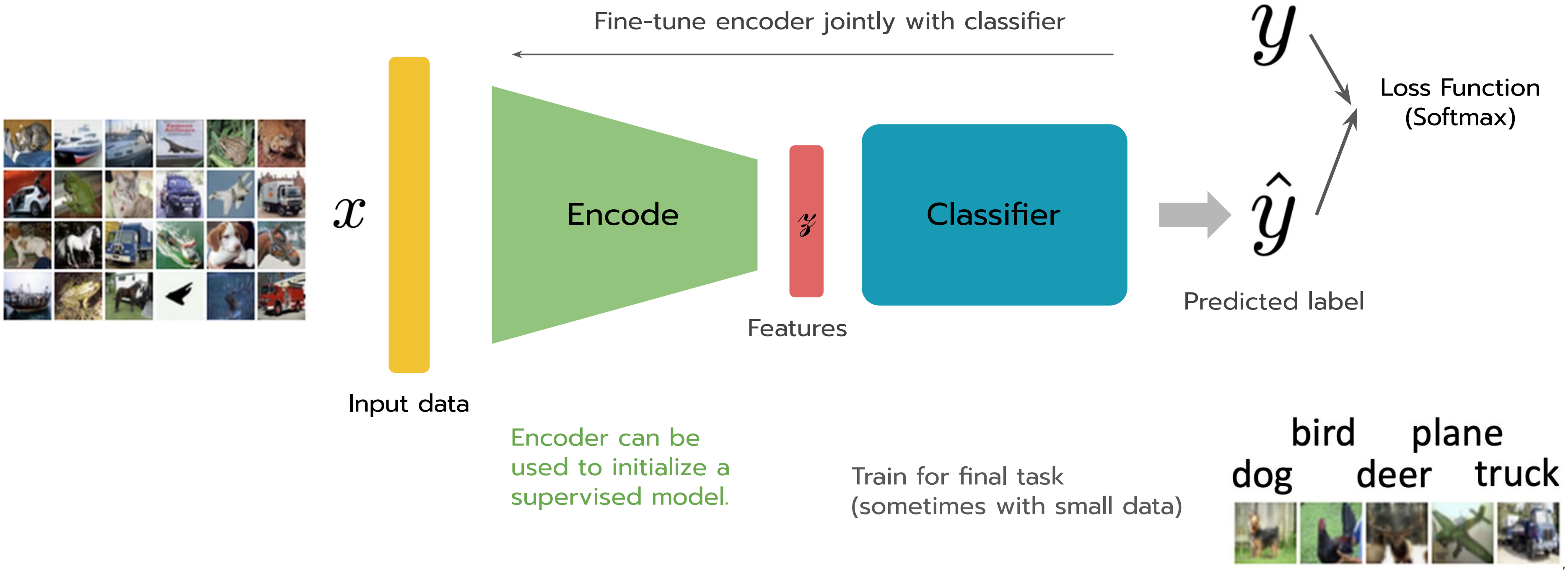
$$||\hat{x} - x||_2^2$$

Loss function

Variational AUTOENCODERS

Autoencoders (Regular, non-variational)

After training, **throw away decoder** and use encoder for a downstream task



Encoder can be used to initialize a supervised model.

Train for final task
(sometimes with small data)

Variational AUTOENCODERS

Applications

- Dimensionality Reduction and Feature Extraction
- Denoising Data
- Anomaly detections

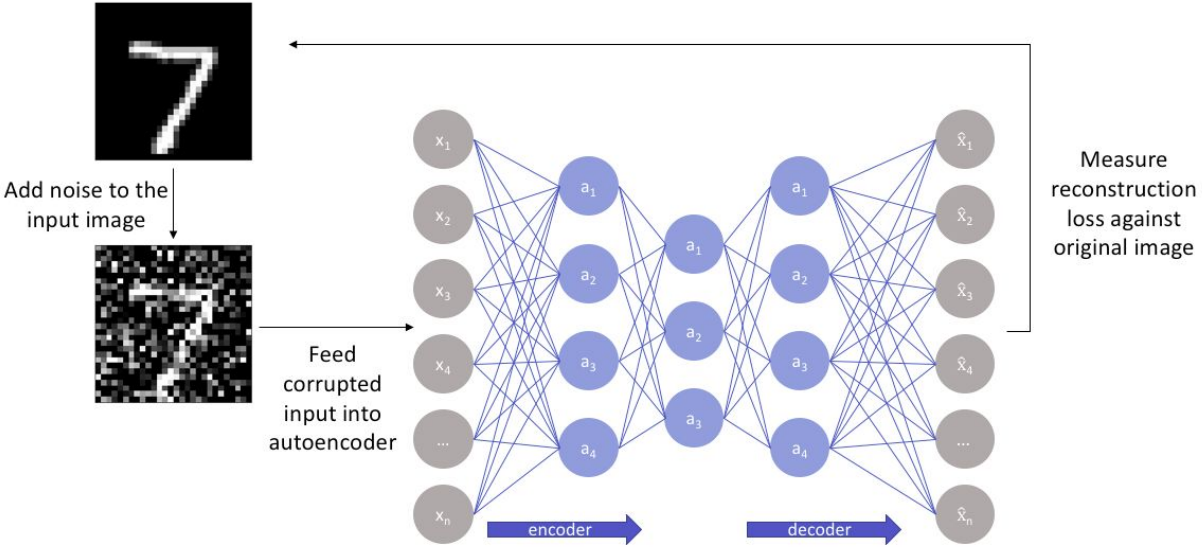


Image from: https://www.v7labs.com/blog/autoencoders-guide

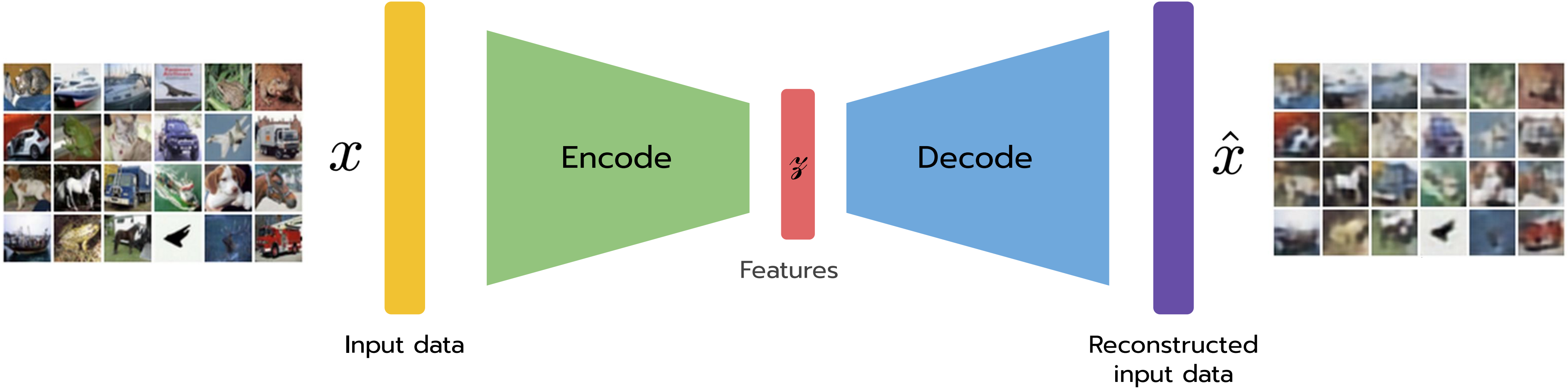
Variational AUTOENCODERS

Autoencoders (Regular, non-variational)

Autoencoders learn **latent features** for data without any labels!

Can use features to initialize a supervised model

Not probabilistic: No way to sample **new data** from learned model

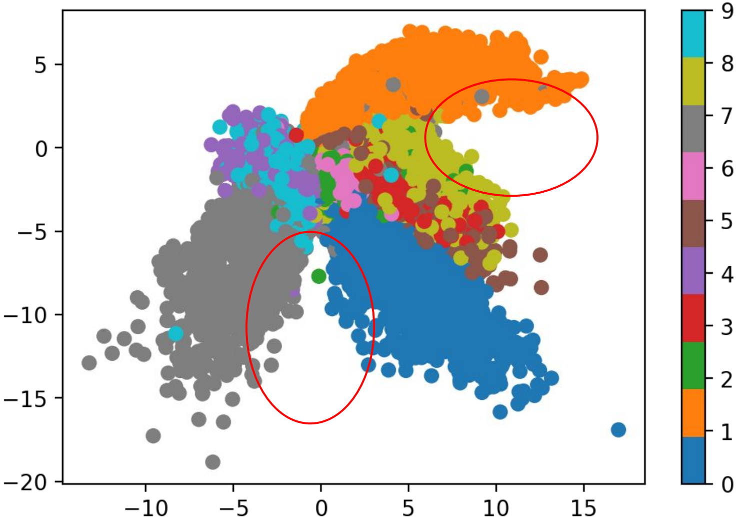


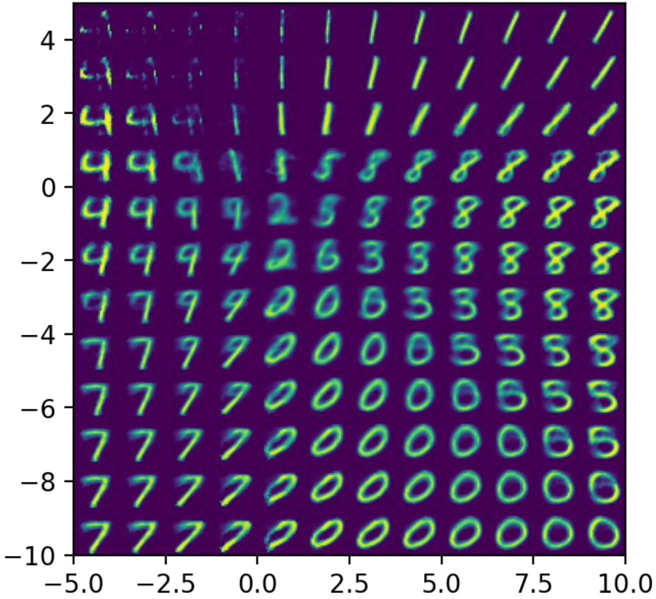
Variational AUTOENCODERS

Autoencoders (Regular, non-variational)

There are "gaps" in the latent space, where data is never mapped to.

If we sample a latent vector from a **region** in the **latent space that was never seen** by the decoder during training, the output might not make any sense at all.



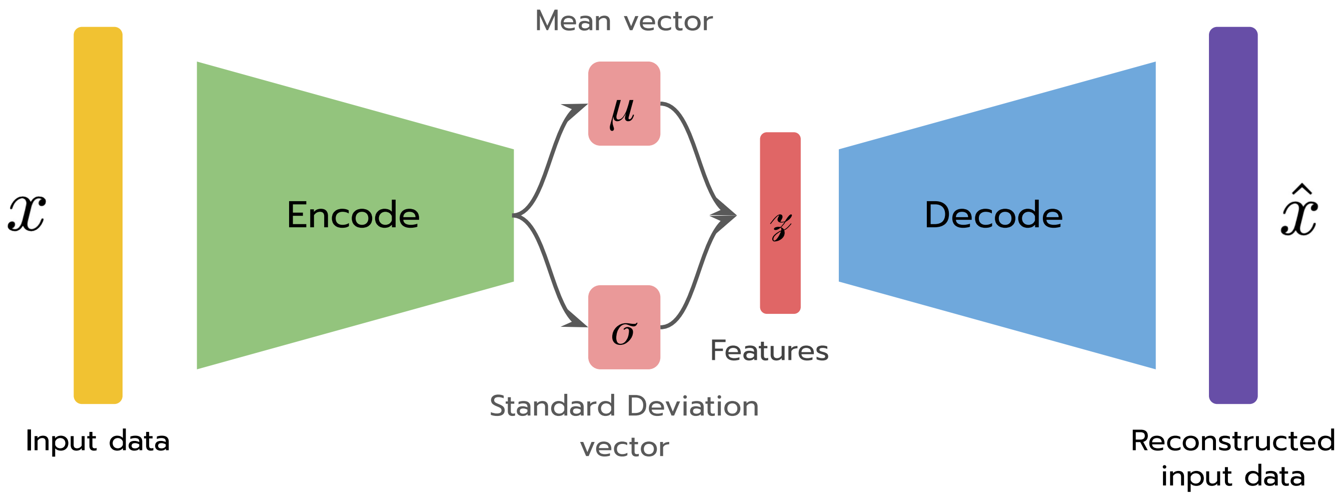


the latent space, z, can become **disjoint** and **non-continuous**.

Image from: https://avandekleut.github.io/vae/

Variational Autoencoder

Variational autoencoders are a probabilistic twist on autoencoders!
Sample from the mean and standard deviation to compute latent sample



Overview of VAE : Instead of learning an arbitrary function, the network learns the parameters of **probability distribution** modeling the data

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Variational Autoencoder

Intuition

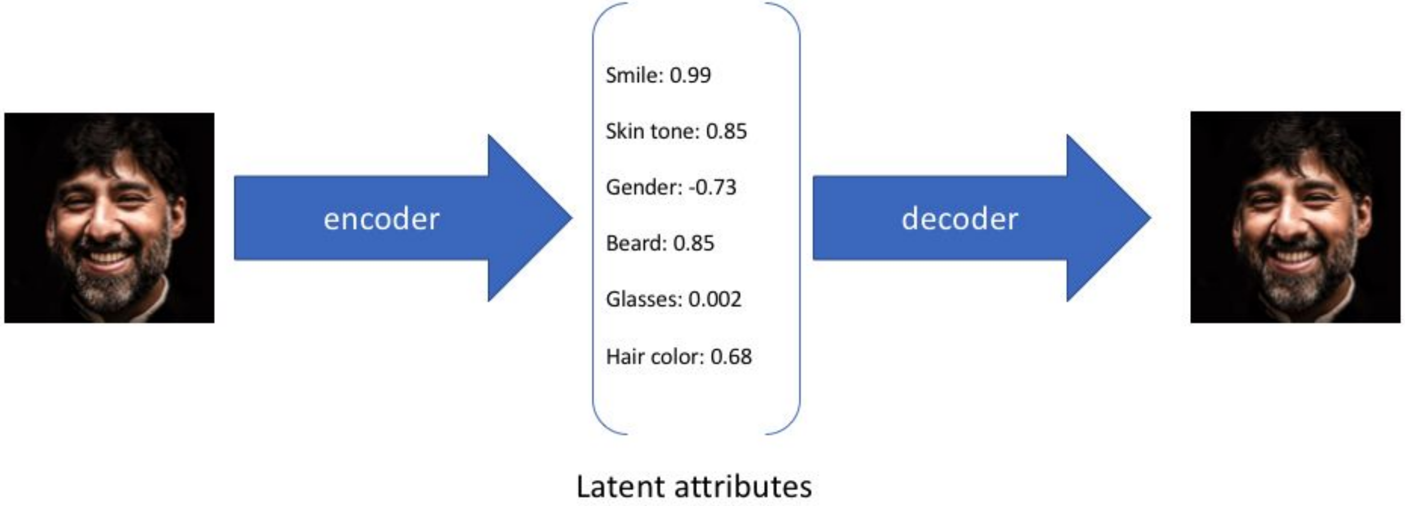


Image from: https://www.jeremyjordan.me/variational-autoencoders/

Variational Autoencoder

Intuition

Using a variational autoencoder, we can describe latent attributes in probabilistic terms.

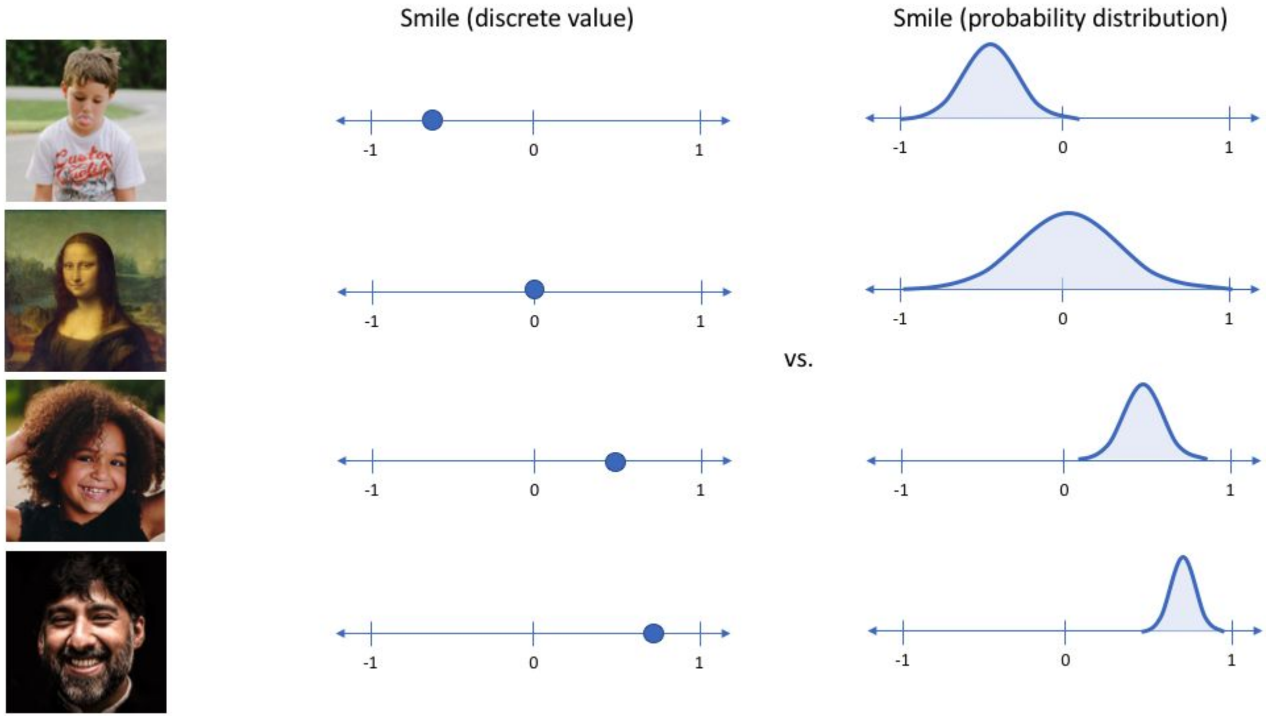
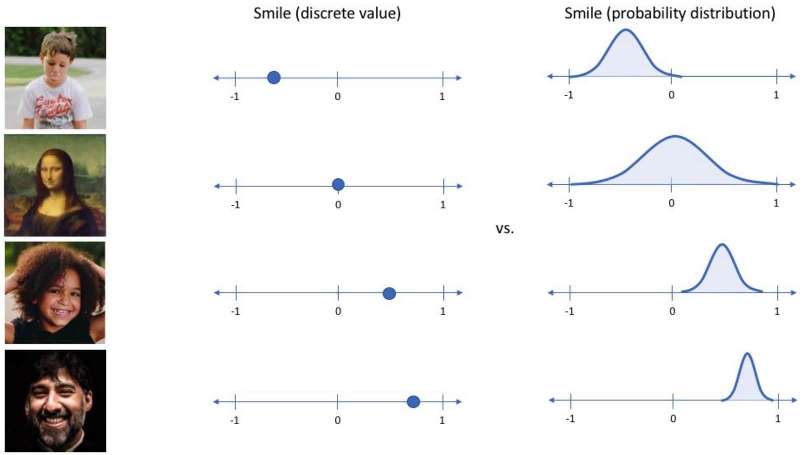
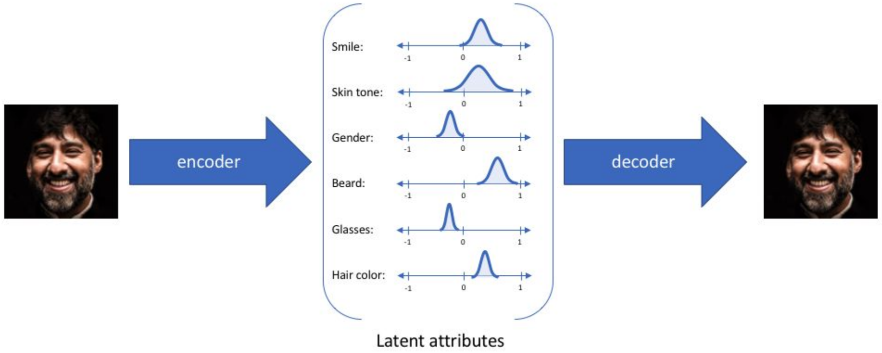


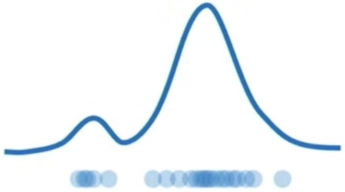
Image from: https://www.jeremyjordan.me/variational-autoencoders/

Variational Autoencoder

Intuition







After training with lots of samples, the probability distribution of each attribute is learned. So that we can represent each latent attribute as a probability distribution.

Image from: https://www.jeremyjordan.me/variational-autoencoders/

Variational Autoencoder

Intuition

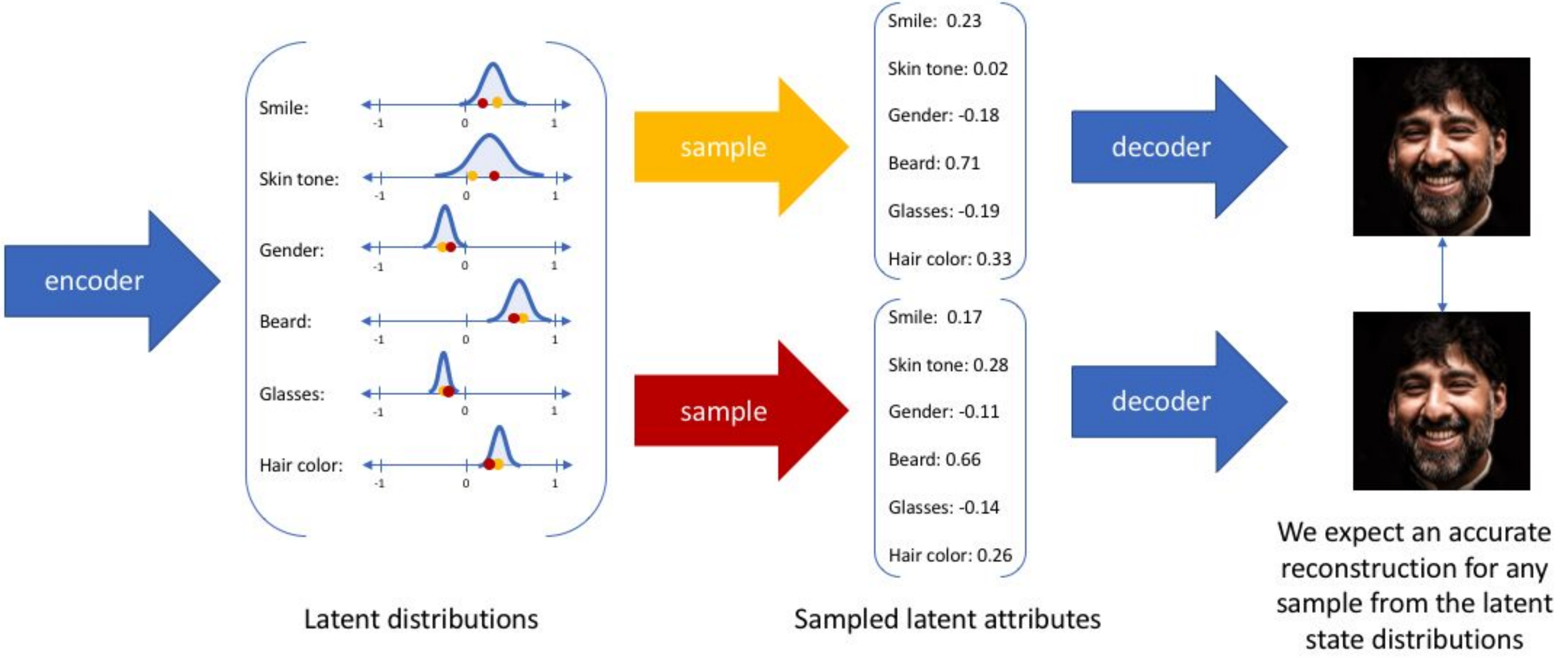
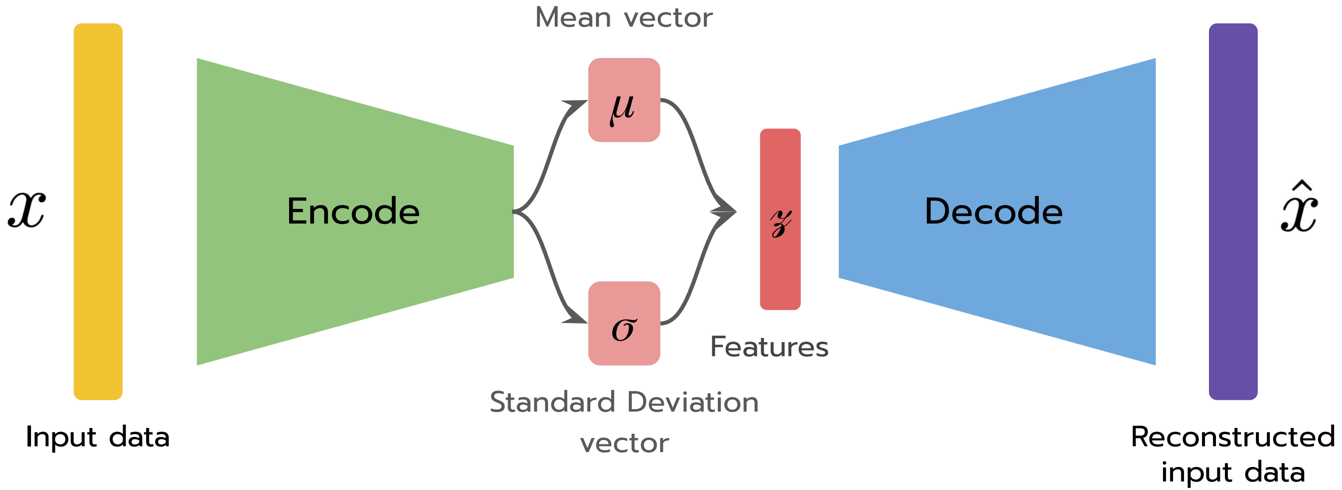


Image from: https://www.jeremyjordan.me/variational-autoencoders/

Variational Autoencoder

Probabilistic spin on autoencoders: we want to do

1. **Learn latent features** z from raw data
2. Sample from the model to **generate new data**



Cherdsak Kingkan

Variational Autoencoder

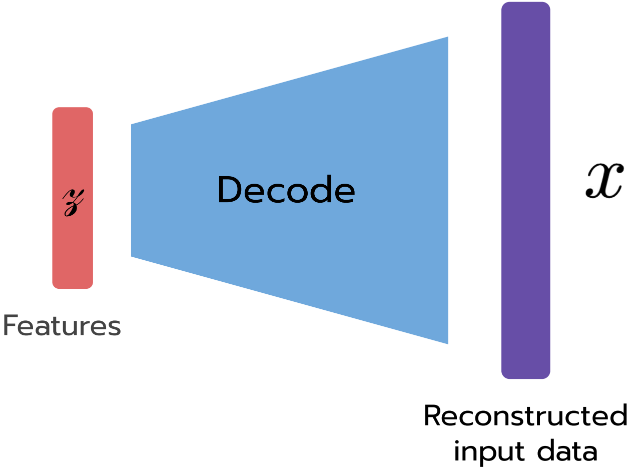
Probabilistic spin on autoencoders: we want to do

1. **Learn latent features** z from raw data
2. Sample from the model to **generate new data**

After training, we want to **sample** a hidden variable z which **generate** an observation x

We can only see x, but we want to infer the attributes of z, i.e., we want to compute

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$



Variational Autoencoder

Probabilistic spin on autoencoders: we want to do

1. **Learn latent features** z from raw data
2. Sample from the model to **generate new data**

After training, we want to **sample** a hidden variable z which **generate** an observation x

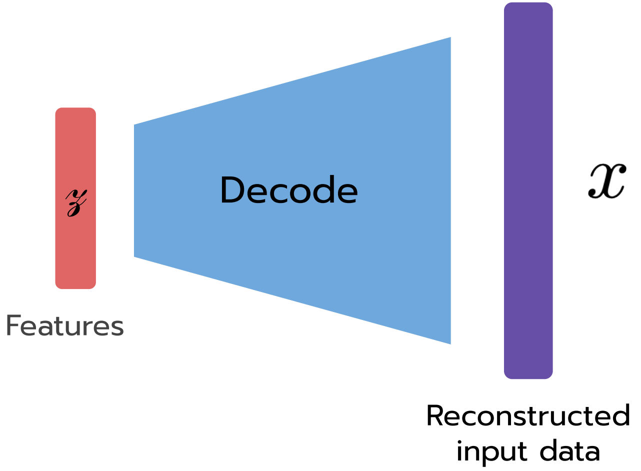
We can only see x, but we want to infer the attributes of z, i.e., we want to compute

compute with **Decoder network**

we assumed Gaussian prior

$$p(z|x) = \frac{\boxed{p(x|z)}\boxed{p(z)}}{\boxed{p(x)}}$$

Very difficult to compute (technically possible, but practical impossible)



Variational Autoencoder

Probabilistic spin on autoencoders: we want to do

1. **Learn latent features** z from raw data
2. Sample from the model to **generate new data**

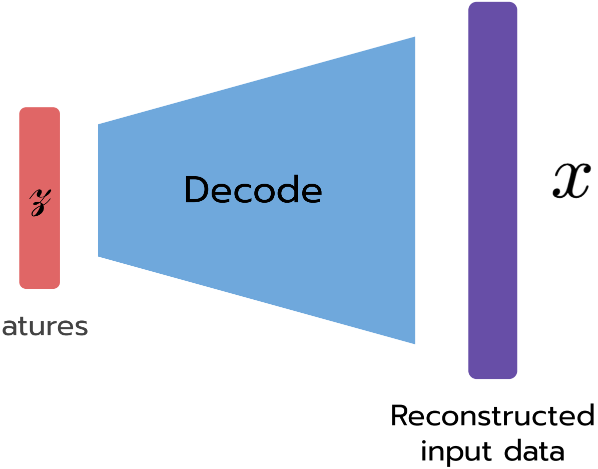
After training, we want to **sample** a hidden variable z which **generate** an observation x

We can only see x, but we want to infer the attributes of z, i.e., we want to compute

compute with **Decoder network**

we assumed Gaussian prior

$$p(z|x) = \frac{\boxed{p(x|z)}\boxed{p(z)}}{\boxed{p(x)}}$$



$$p(x) = \int p(x|z)p(z)dz$$

Impossible to integrate over all z (intractable)

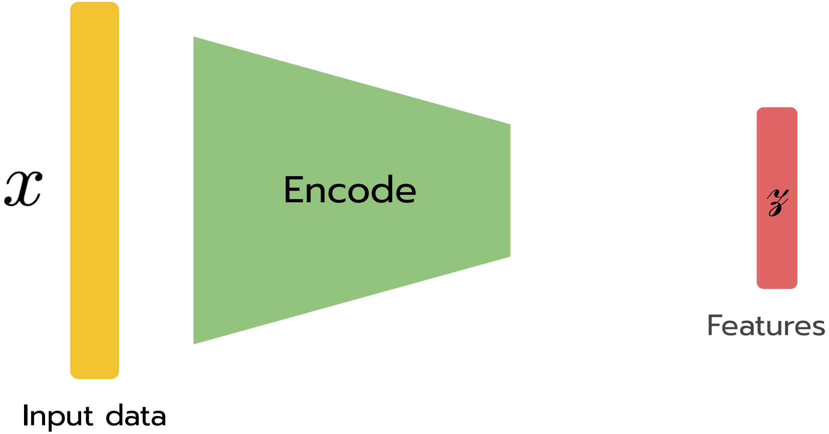
Variational Autoencoder

Probabilistic spin on autoencoders: we want to do
1. **Learn latent features** z from raw data
2. Sample from the model to **generate new data**

We can approximate $p(z|x)$ by another distribution $q(z|x)$, i.e.

Compute with Encoder Network

$$\boxed{q(z|x)} \approx p(z|x)$$



the KL divergence is a measure of difference between two probability distributions. Thus, if we wanted to ensure that $q(z|x)$ was similar to $p(z|x)$, we could minimize the KL divergence between the two distributions.

$$minD_{KL}(q(z|x)||p(z|x))$$

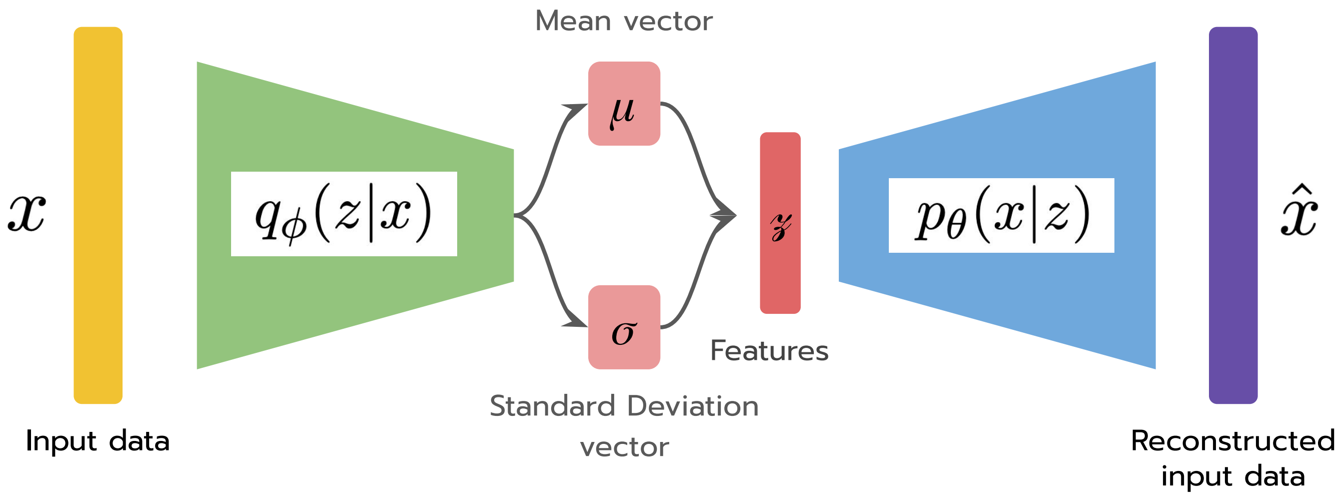
Variational Autoencoder

Common choice of prior - Gaussian Distribution

$$p(z) = \mathcal{N}(\mu, \sigma)$$

Probabilistic spin on autoencoders: we want to do

1. **Learn latent features** z from raw data
2. Sample from the model to **generate new data**



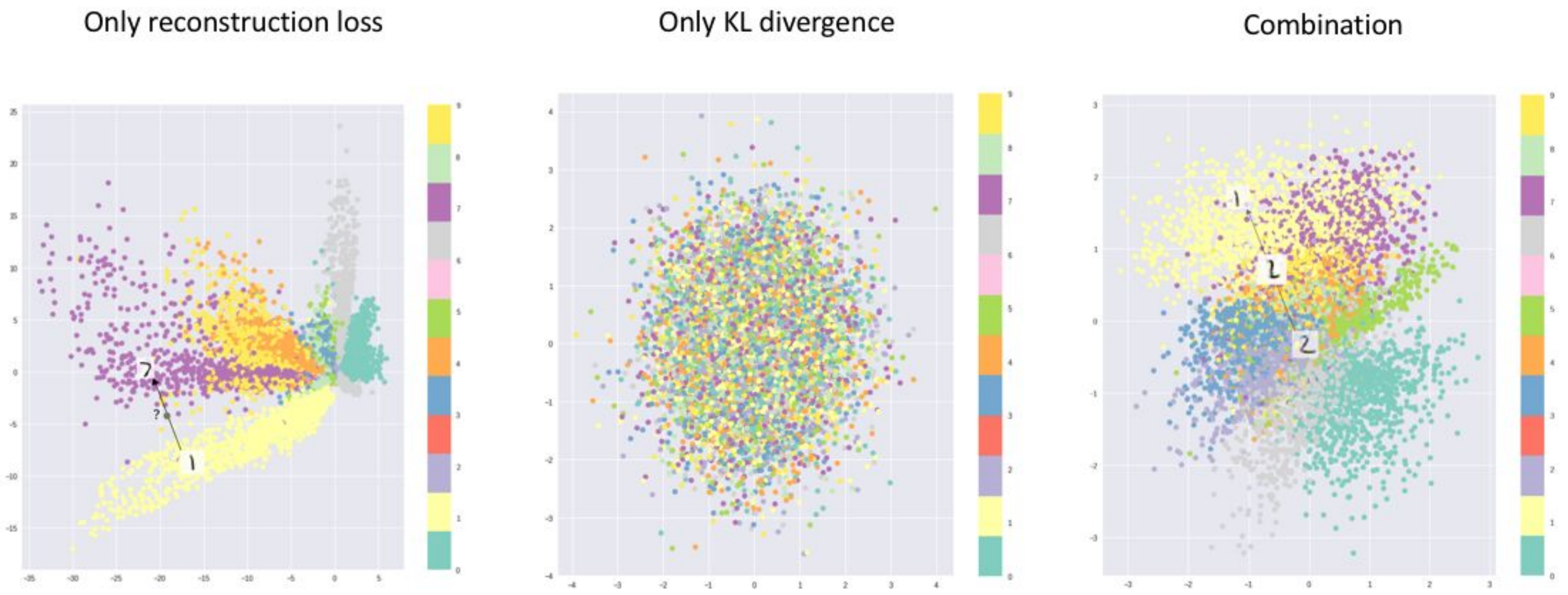
The reconstruction likelihood

$$\boxed{E_{q(z|x)} \log p(x|z)} - \boxed{D_{KL}(q(z|x)||p(z))}$$

Regularization Loss: ensures that our **learned distribution** is similar to the true **prior distribution**

Variational Autoencoder

Why regularization term is needed



Only reconstruction loss

Only KL divergence

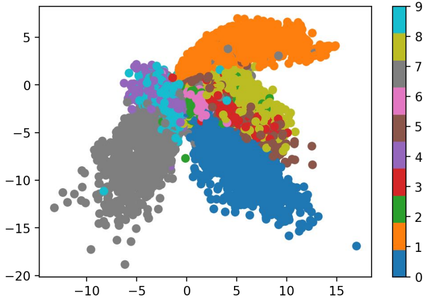
Combination

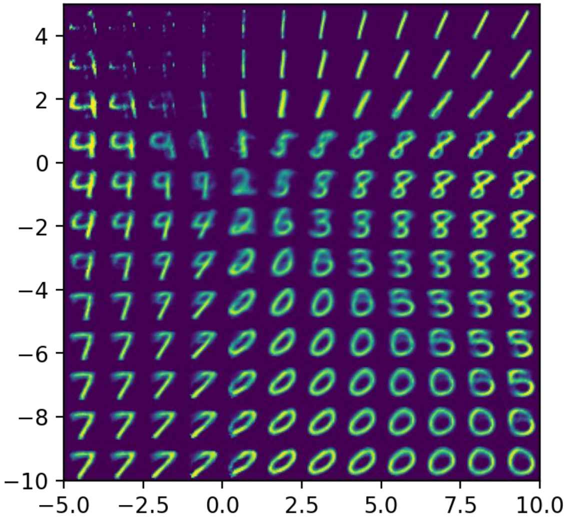
Image from: https://www.jeremyjordan.me/variational-autoencoders/

Variational Autoencoder

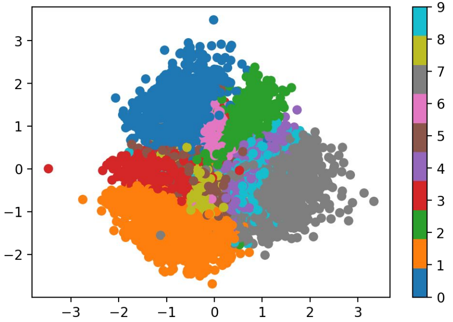
Why regularization term is needed

Autoencoder





Variational Autoencoder



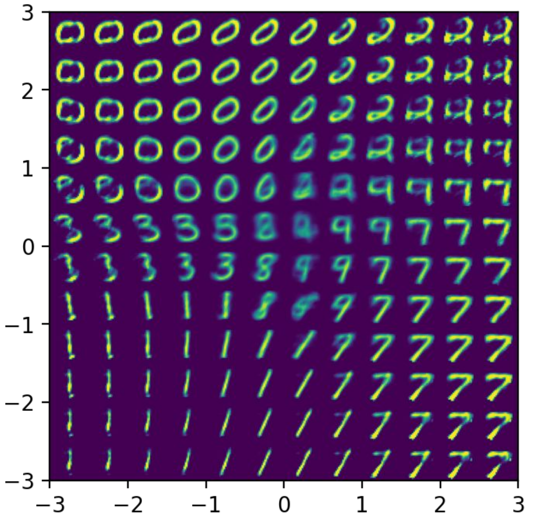
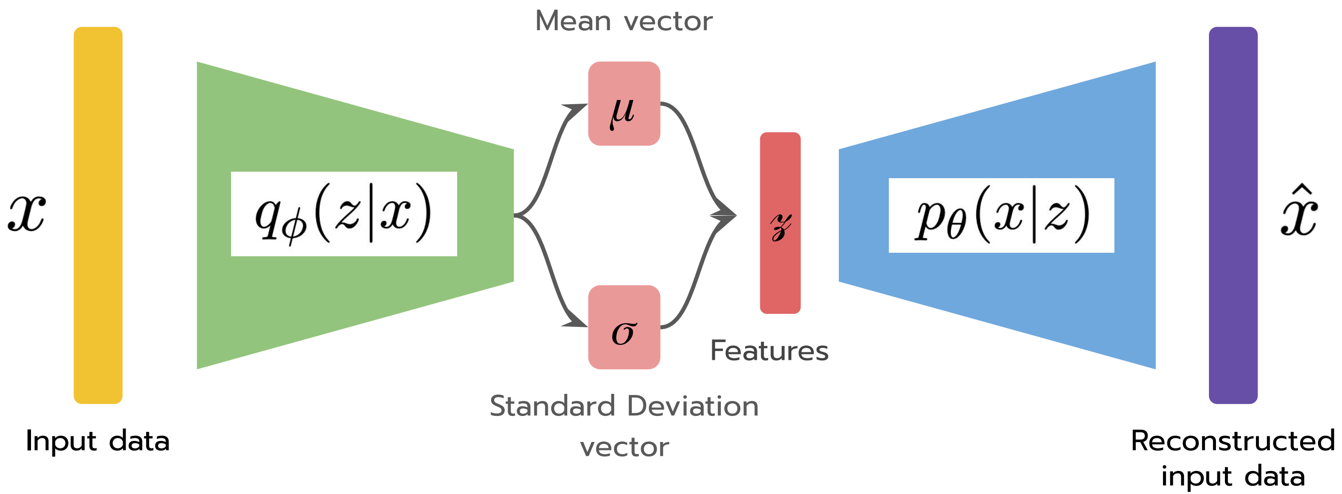


Image from: https://www.jeremyjordan.me/variational-autoencoders/

Variational Autoencoder

How to perform backpropagation when training training the model



x

Input data

$q_\phi(z|x)$

Mean vector

μ

σ

Standard Deviation vector

z

Features

$p_\theta(x|z)$

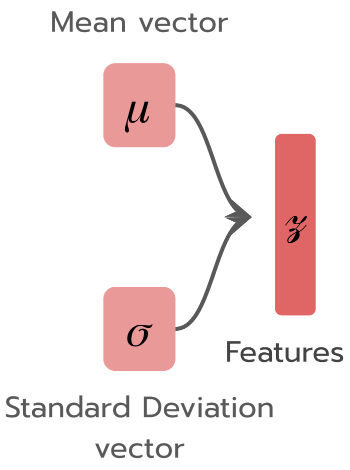
$\hat{x}$

Reconstructed input data

$$E_{q(z|x)} \log p(x|z) - D_{KL}(q(z|x)||p(z))$$

Variational Autoencoder

Reparameterization trick



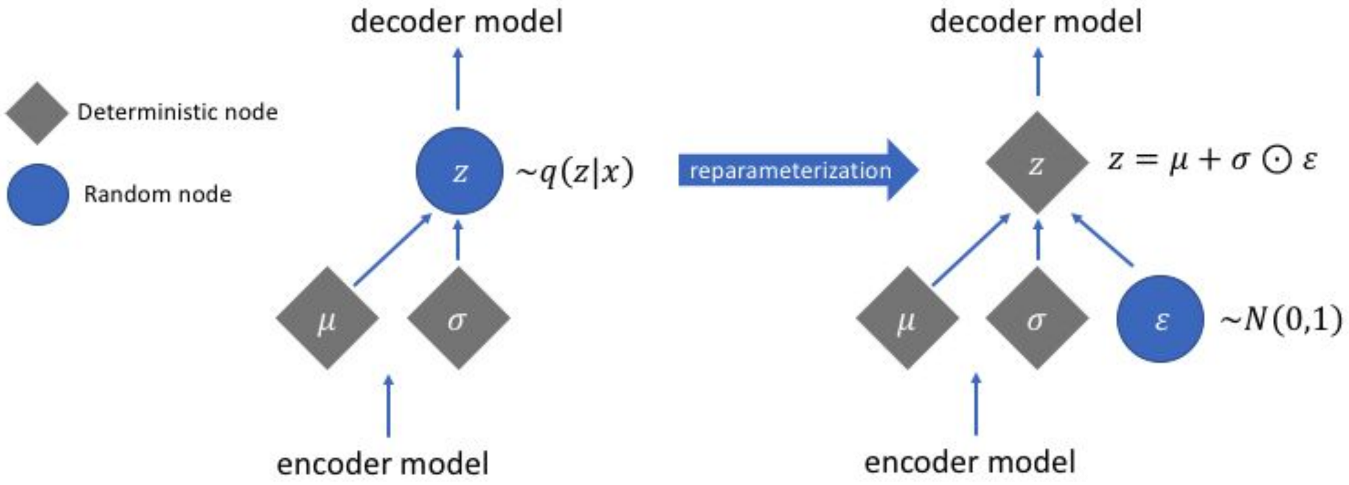
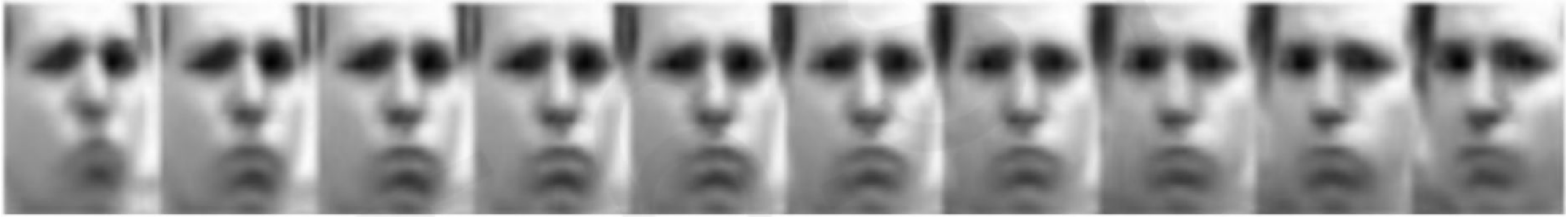


Image from: https://www.jeremyjordan.me/variational-autoencoders/

Variational Autoencoder

Latent perturbation

Slowly increase or decrease a **single latent variable**. Keep all other variables fixed.



Head pose

Different dimension of z encodes **different interpretable latent features**

Image from: https://www.jeremyjordan.me/variational-autoencoders/

Variational Autoencoder

Latent perturbation

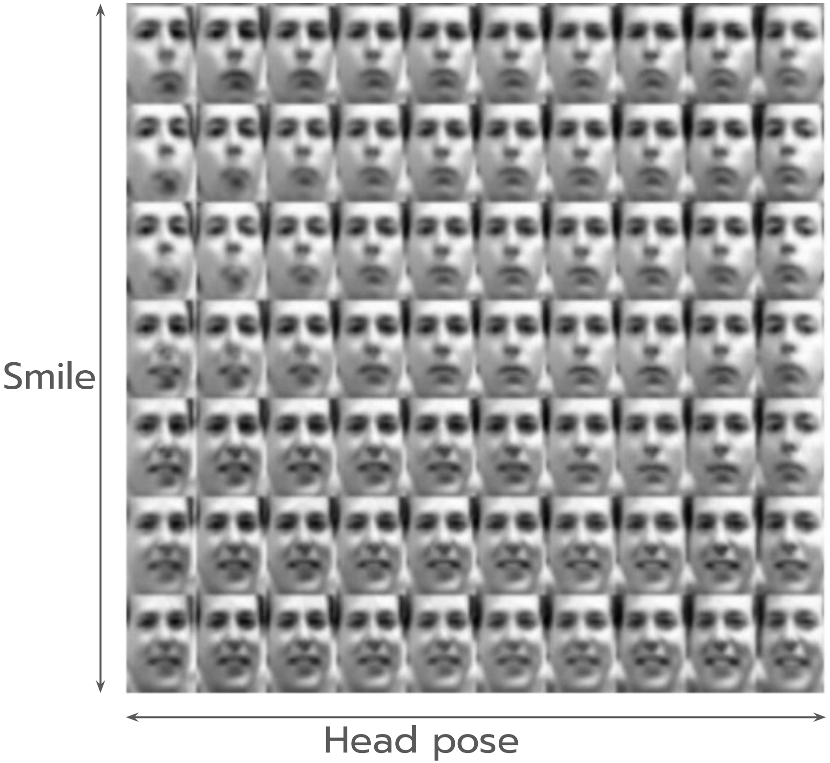


Image from: https://www.jeremyjordan.me/variational-autoencoders/

Ideally, we want latent variables that are uncorrelated with each other.

Enforce diagonal prior on the latent variables to encourage independence

Disentanglement

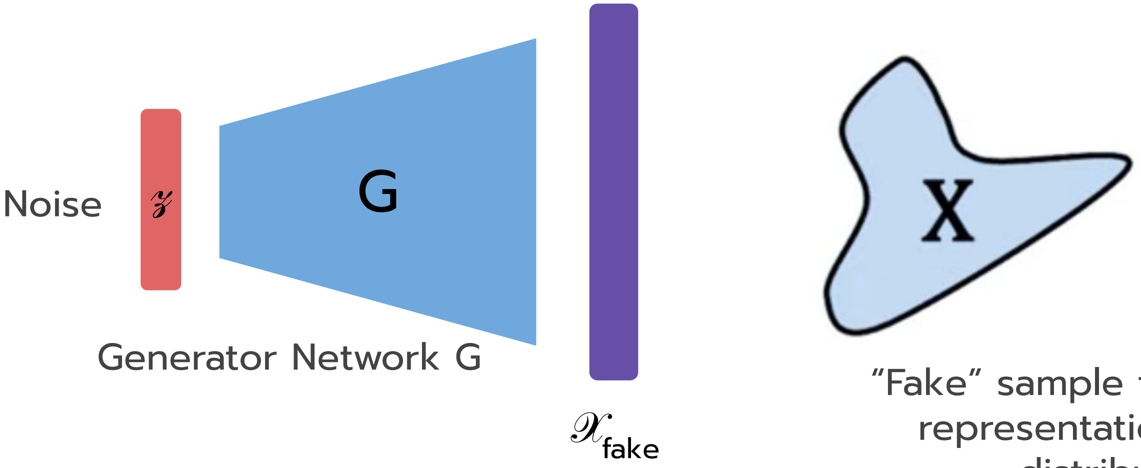
Generative Adversarial Network (GAN)

What if we just want to sample

Idea: do not explicitly model density, and instead just sample to generate new instances.

Problem: want to sample from complex distribution - cannot do this directly

Solution: sample from something simple (e.g., noise), learn a transformation to the data distribution.



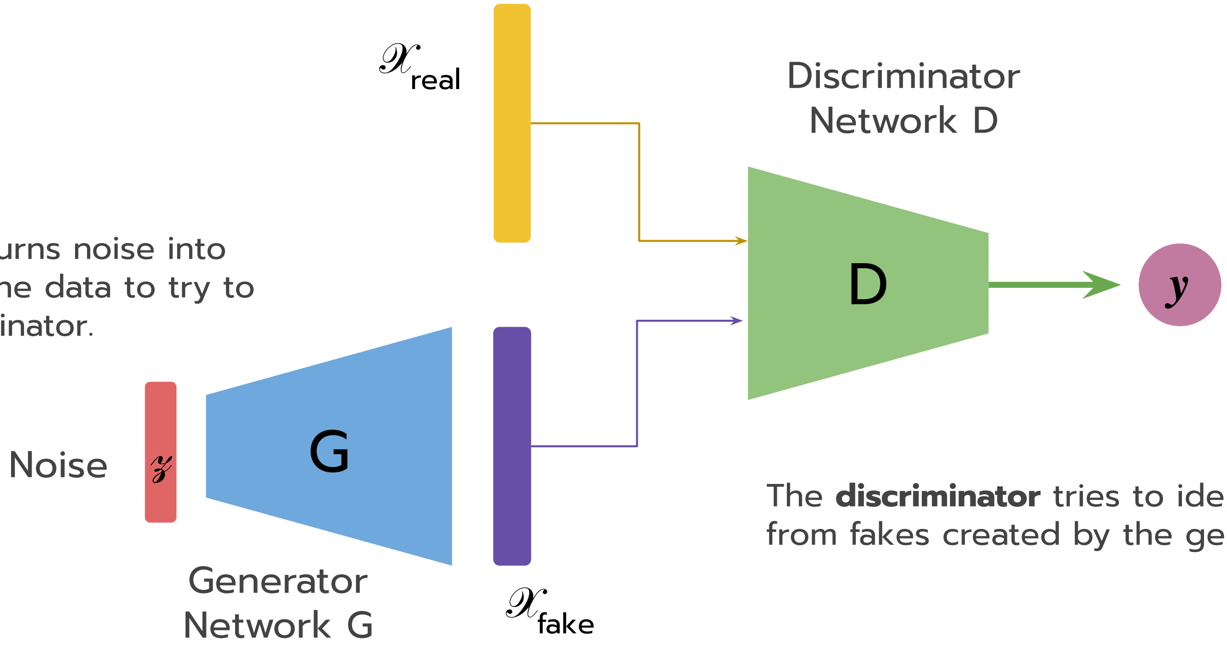
“Fake” sample from learned representation of data distribution

Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Networks (GANs)

GANs are a way to make a generative model by having two neural networks compete with each other

The **generator** turns noise into an imitation of the data to try to trick the discriminator.



The **discriminator** tries to identify real data from fakes created by the generator.

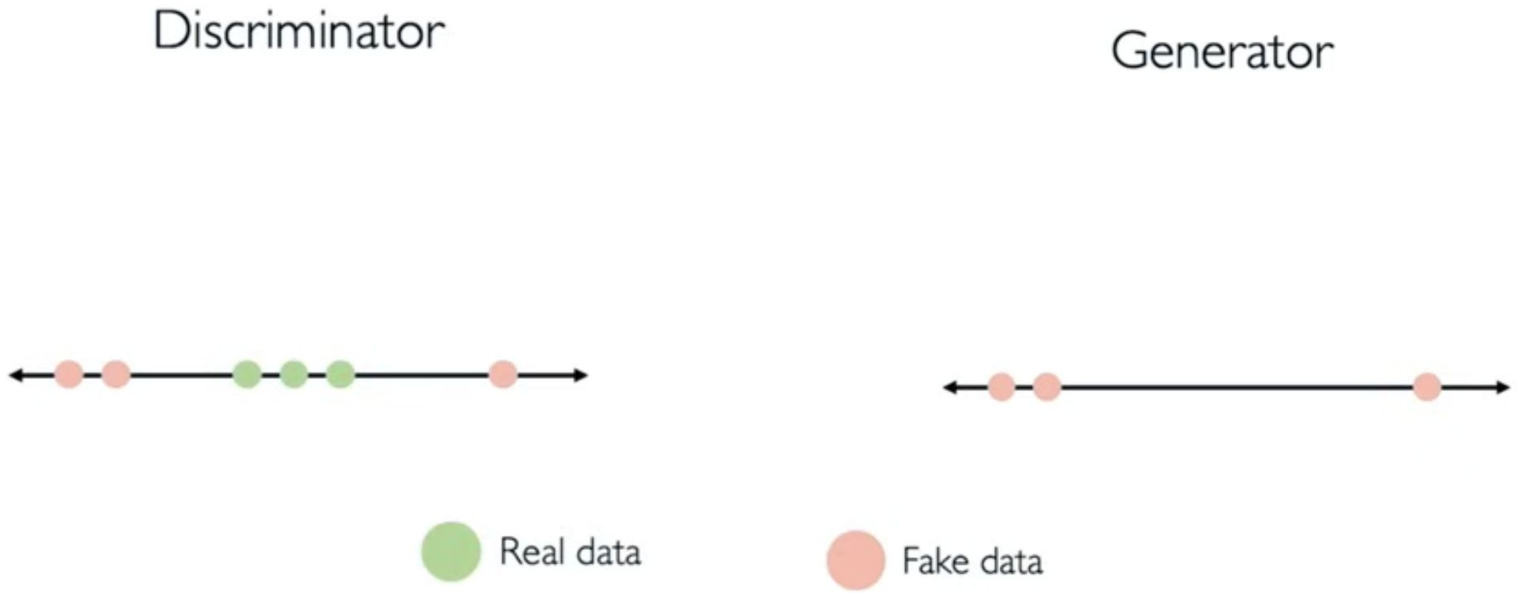
Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Network (GAN)

Intuition behind GANs

Discriminator looks at both real and fake data created by the generator

Generator starts from noise to try to create an imitation of the data



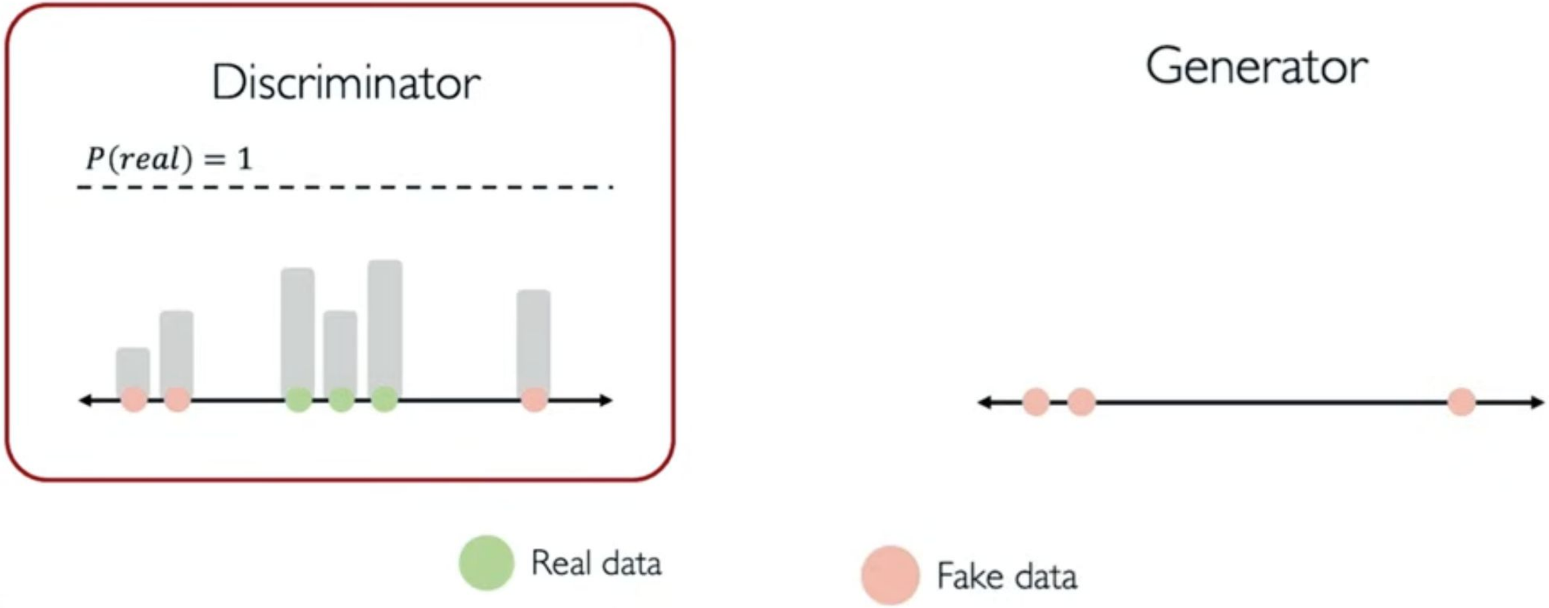
Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Network (GAN)

Intuition behind GANs

Discriminator tries to predict what's real and what's fake

Generator starts from noise to try to create an imitation of the data



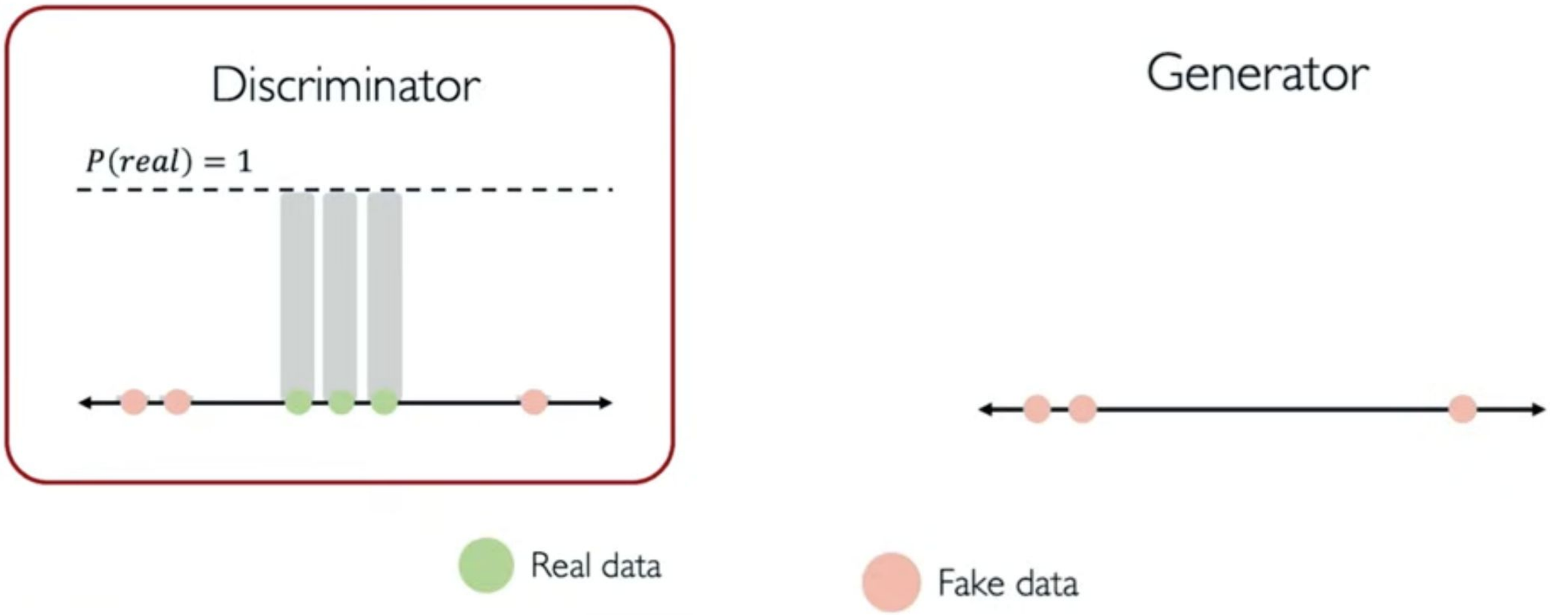
Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Network (GAN)

Intuition behind GANs

Discriminator tries to predict what's real and what's fake

Generator starts from noise to try to create an imitation of the data



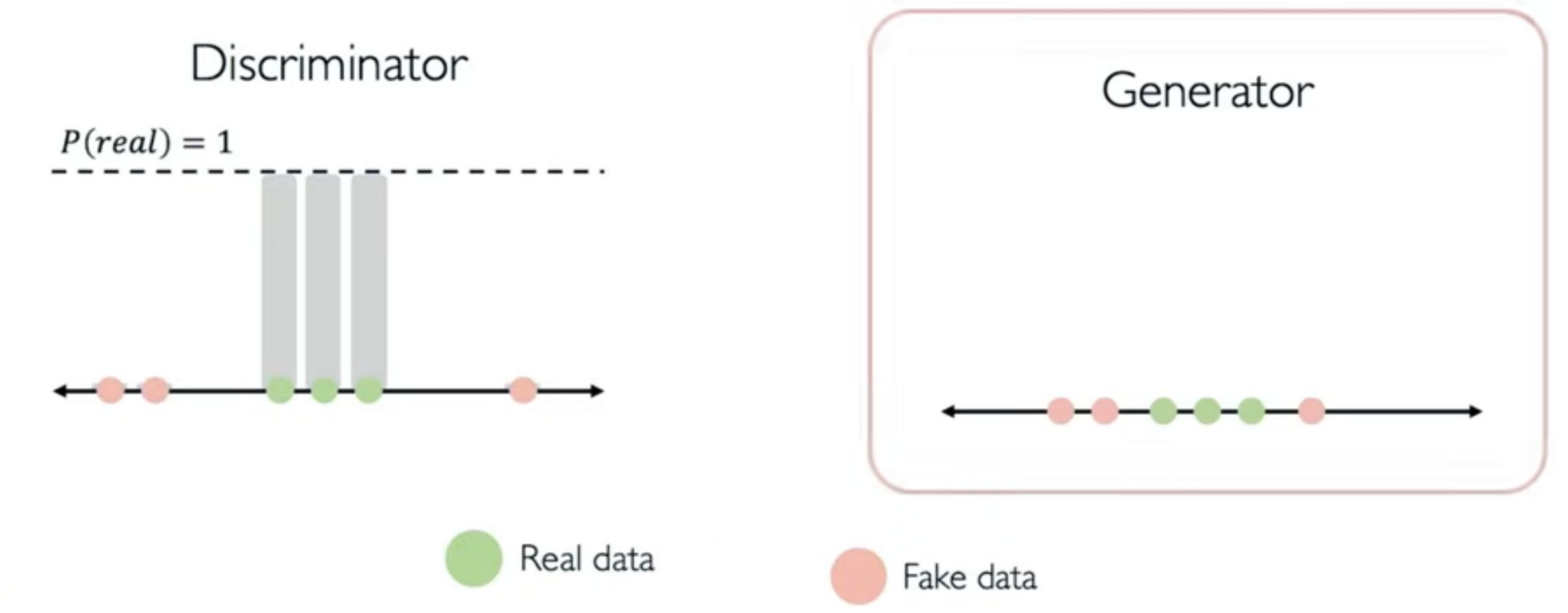
Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Network (GAN)

Intuition behind GANs

Discriminator tries to predict what's real and what's fake

Generator tries to improve its imitation of the data



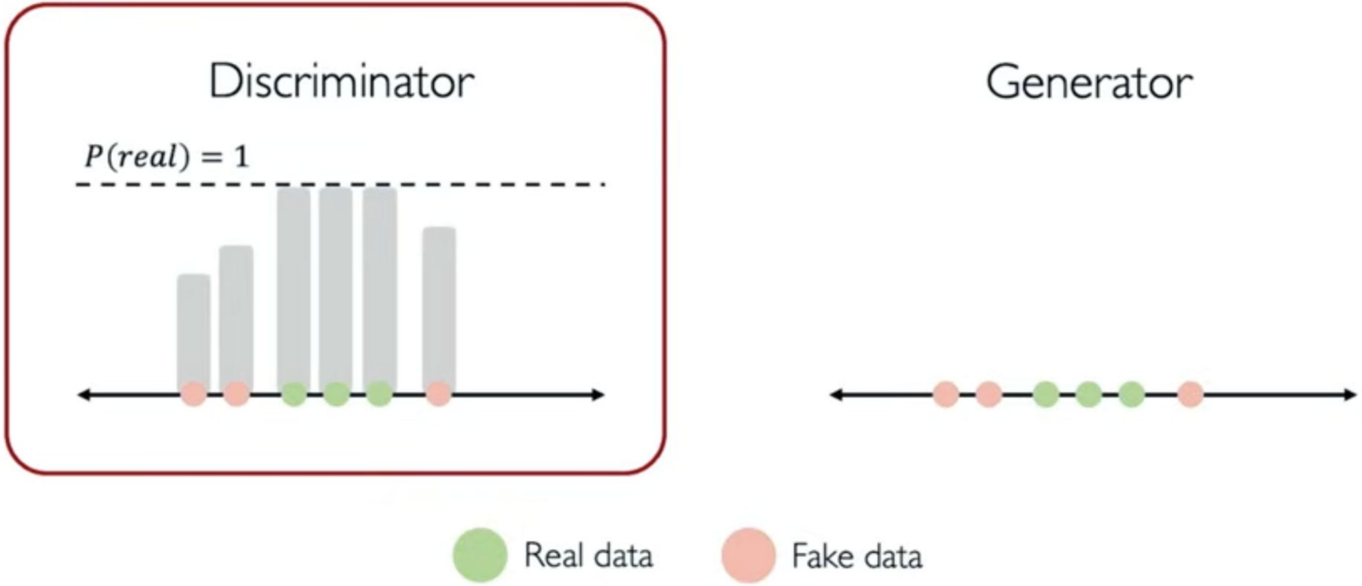
Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Network (GAN)

Intuition behind GANs

Discriminator tries to predict what's real and what's fake

Generator tries to improve its imitation of the data



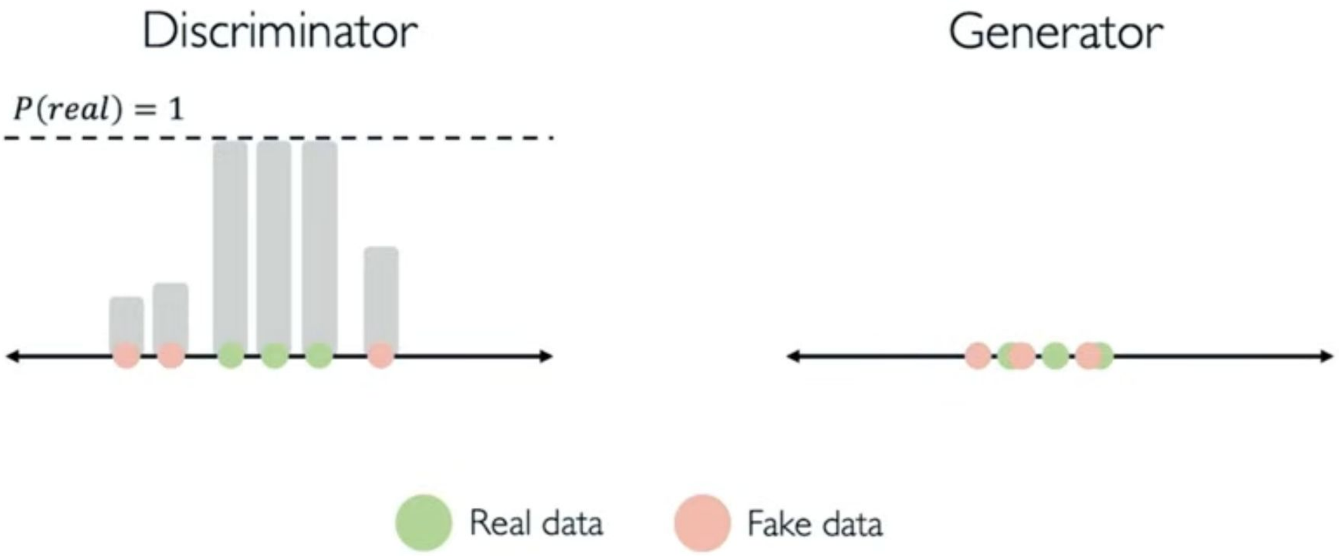
Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Network (GAN)

Intuition behind GANs

Discriminator tries to identify real data from fakes created by the generator.

Generator tries to create imitations of data to trick the discriminator.



Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

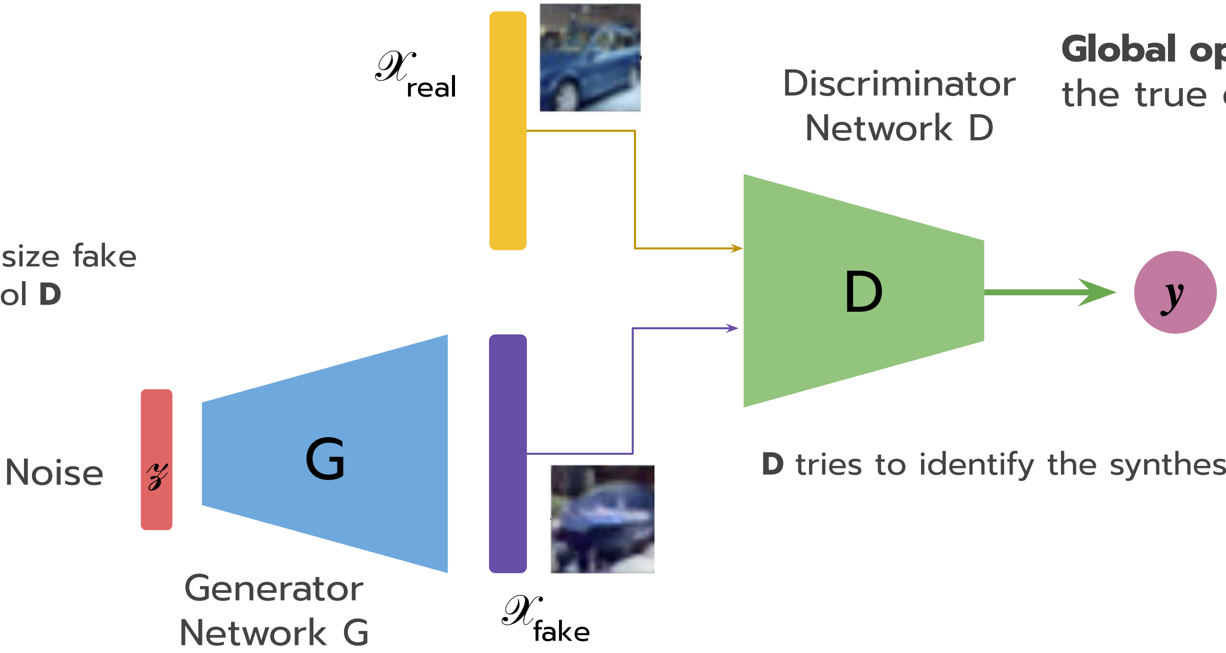
Generative Adversarial Networks (GANs)

Training GANs

Training: adversarial objective for **D** and **G**

Global optimum: G reproduces the true data distribution.

G tries to synthesize fake instances that fool **D**



D tries to identify the synthesized instances

Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Networks (GANs)

Training GANs

Jointly train generator G and discriminator D with a **minimax game**

$$\min_{G} \max_{D} \left(\overbrace{E_{x\sim p_{data}}[\log D(x)]}^{\text{Discriminator wants } D(x) = 1 \text{ for real data}} + E_{z\sim p(z)} \left[\underbrace{\overbrace{\log\left(1 - D(G(z))\right)}^{\text{Discriminator wants } D(x) = 0 \text{ for fake data}}}_{\text{Generator wants } D(x) = 1 \text{ for fake data}} \right] \right)$$

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Generative Adversarial Networks (GANs)

Training GANs

Jointly train generator G and discriminator D with a **minimax game**

Training G and D using alternating gradient updates

$$\min_{G}\max_{D}\left(E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p(z)}\left[\log\left(1 - D(G(z))\right)\right]\right)$$

$$= \min_{G}\max_{D} V(G, D)$$

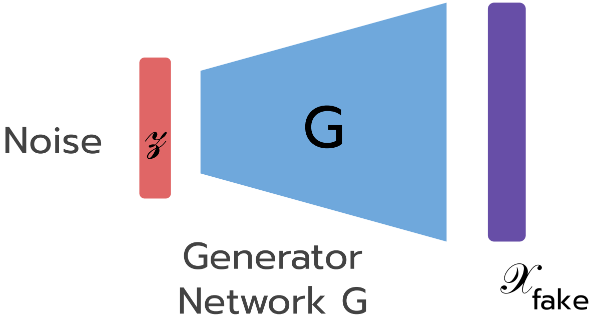
```
For t in 1, ... T:
    1. (Update D) $D = D + \alpha_D \frac{\partial V}{\partial D}$
    2. (Update G) $G = G - \alpha_G \frac{\partial V}{\partial G}$
```

Slide adopted from: Justin Johnson, EECS 498-007 / 598-005: Deep Learning for Computer Vision, University of Michigan, https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/

Generative Adversarial Network (GAN)

Generating new data with GANs

After training, use generator network to create **new data** that's never been seen before.



Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

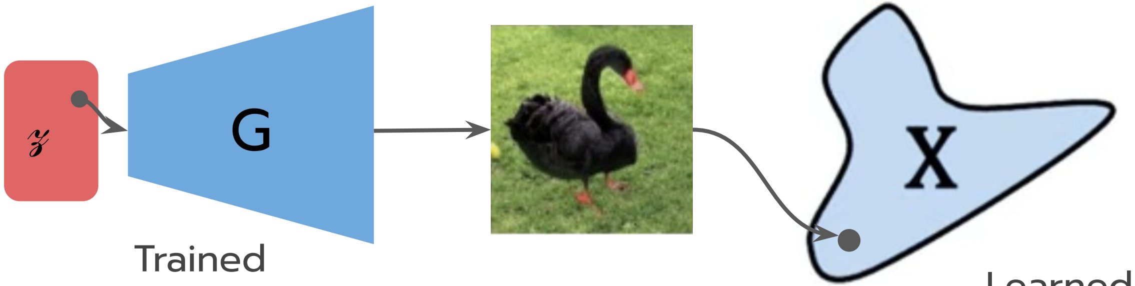
Generative Adversarial Network (GAN)

GANs are distribution transformers

After training, use generator network to create **new data** that's never been seen before.

Gaussian Noise

$z \sim N(0,1)$



Trained Generator

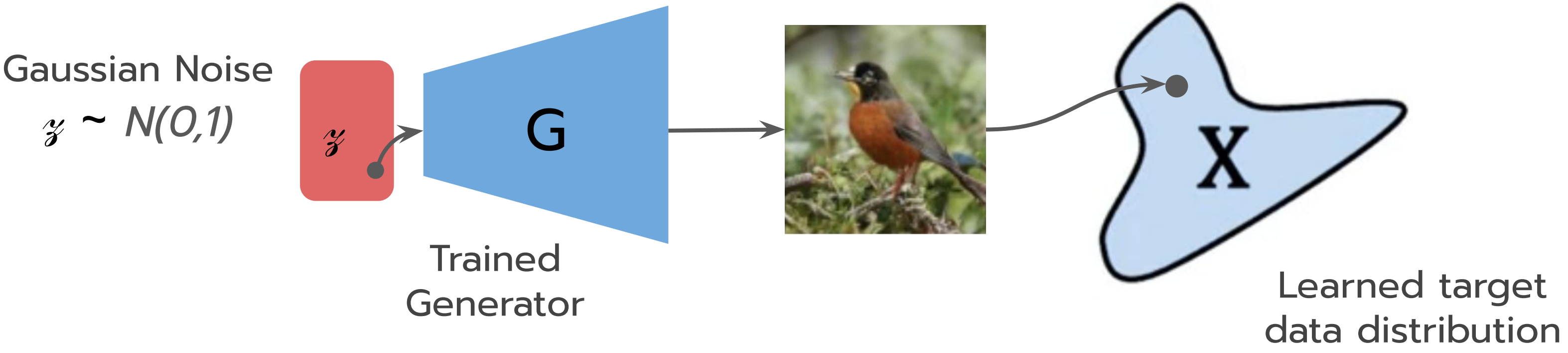
Learned target data distribution

Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Network (GAN)

GANs are distribution transformers

After training, use generator network to create **new data** that's never been seen before.

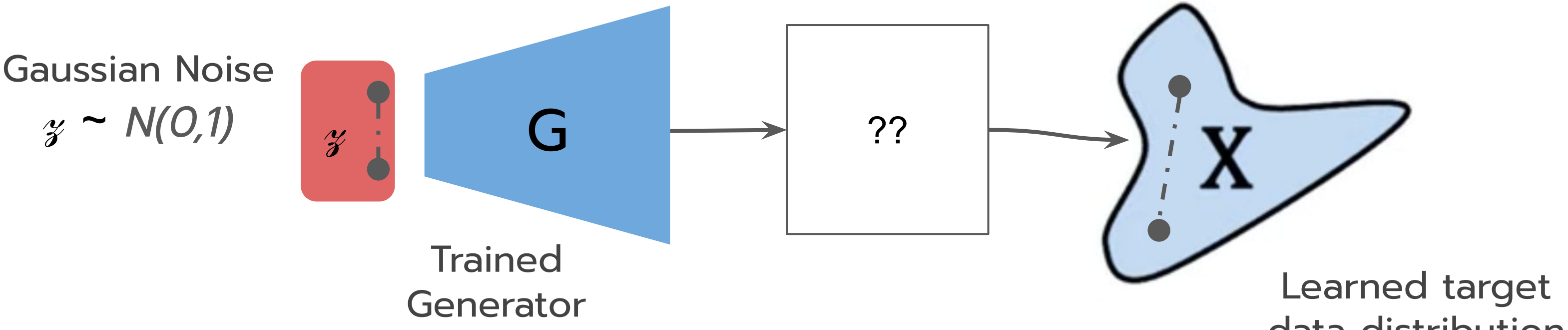


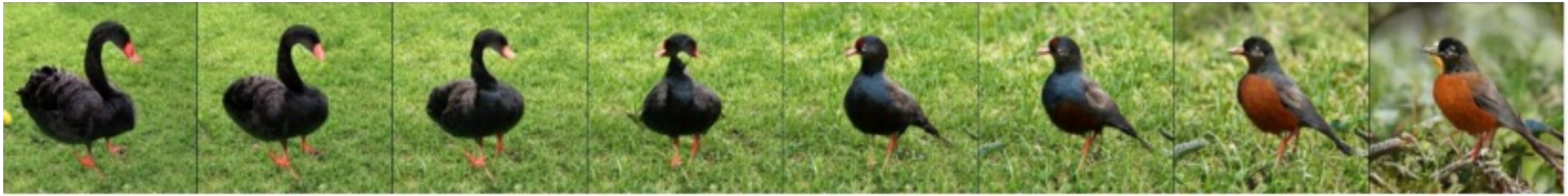
Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Network (GAN)

GANs are distribution transformers

After training, use generator network to create **new data** that's never been seen before.

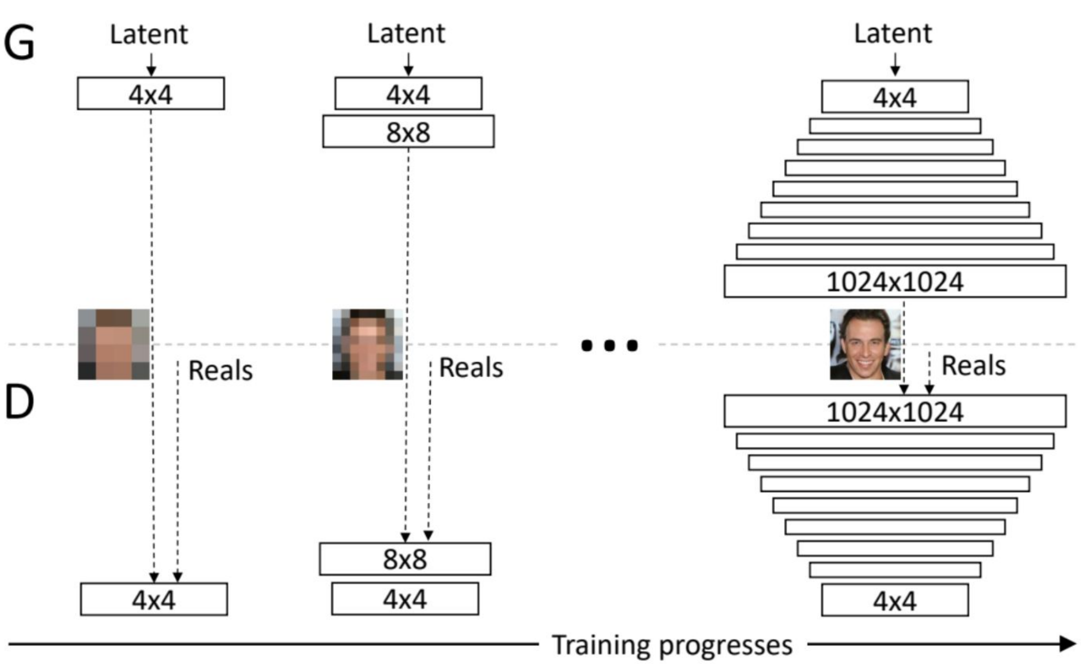


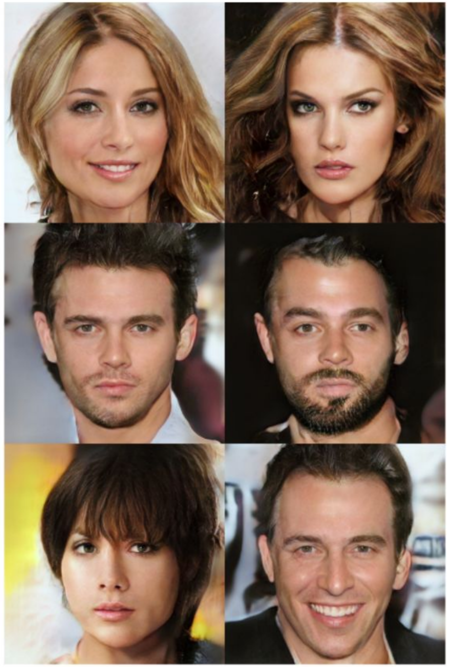


Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Network (GAN)

Applications - Progressively Growing GAN

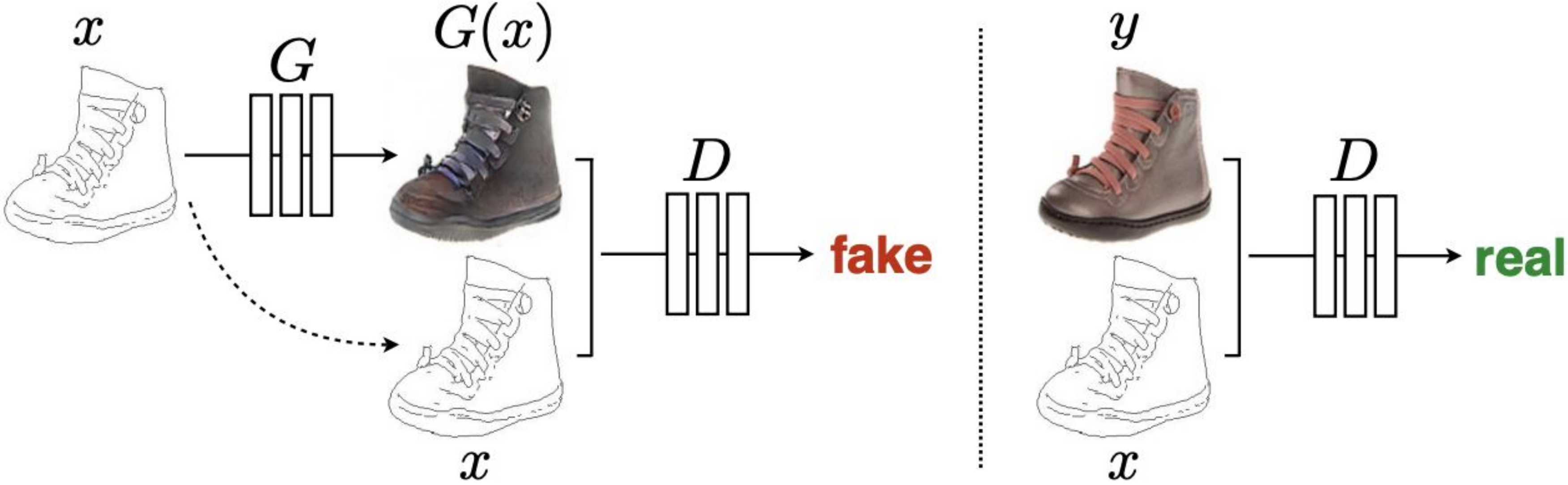




Slide adopted from: A. Amini, et al., MIT 6.S191 Introduction to Deep Learning, MIT, http://introtodeeplearning.com/

Generative Adversarial Network (GAN)

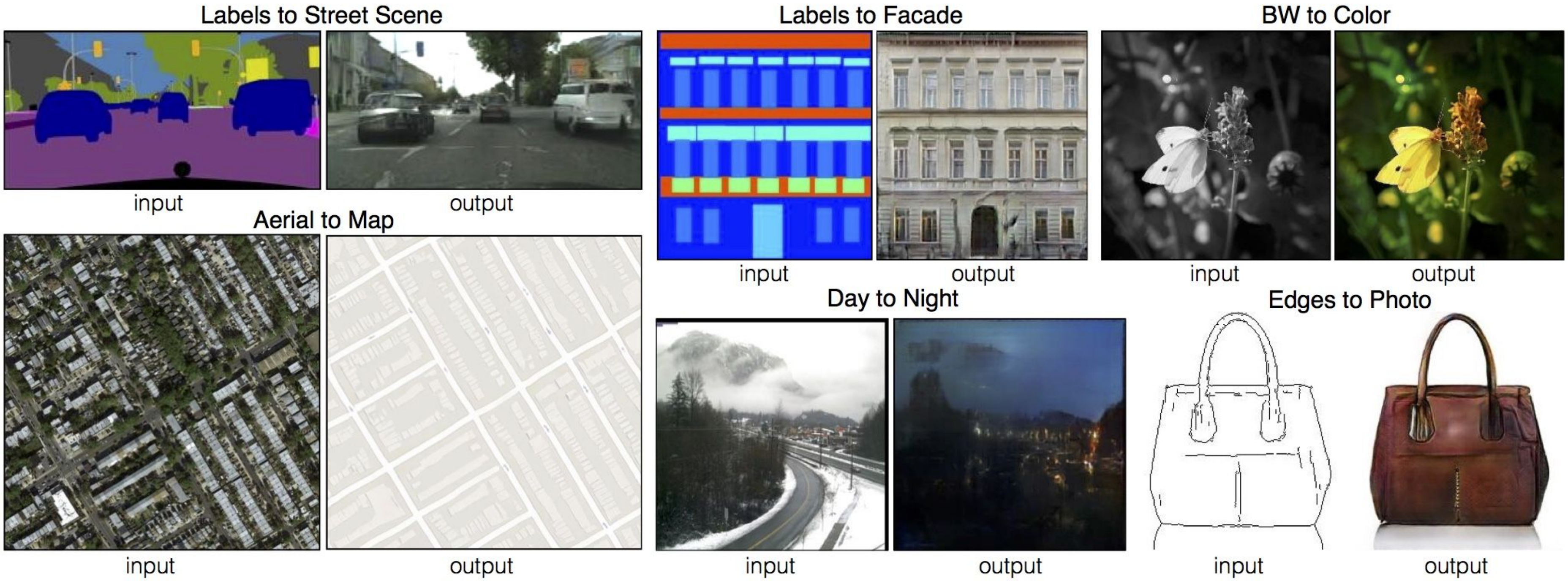
Applications - Conditional GAN



Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros
Image-to-Image Translation with Conditional Adversarial Networks, CVPR, 2017

Generative Adversarial Network (GAN)

Applications - Conditional GAN

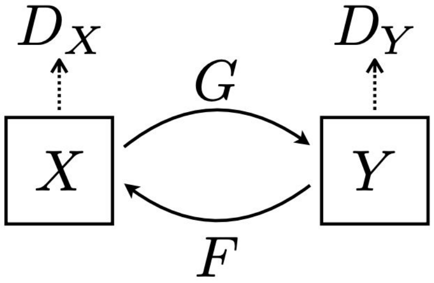


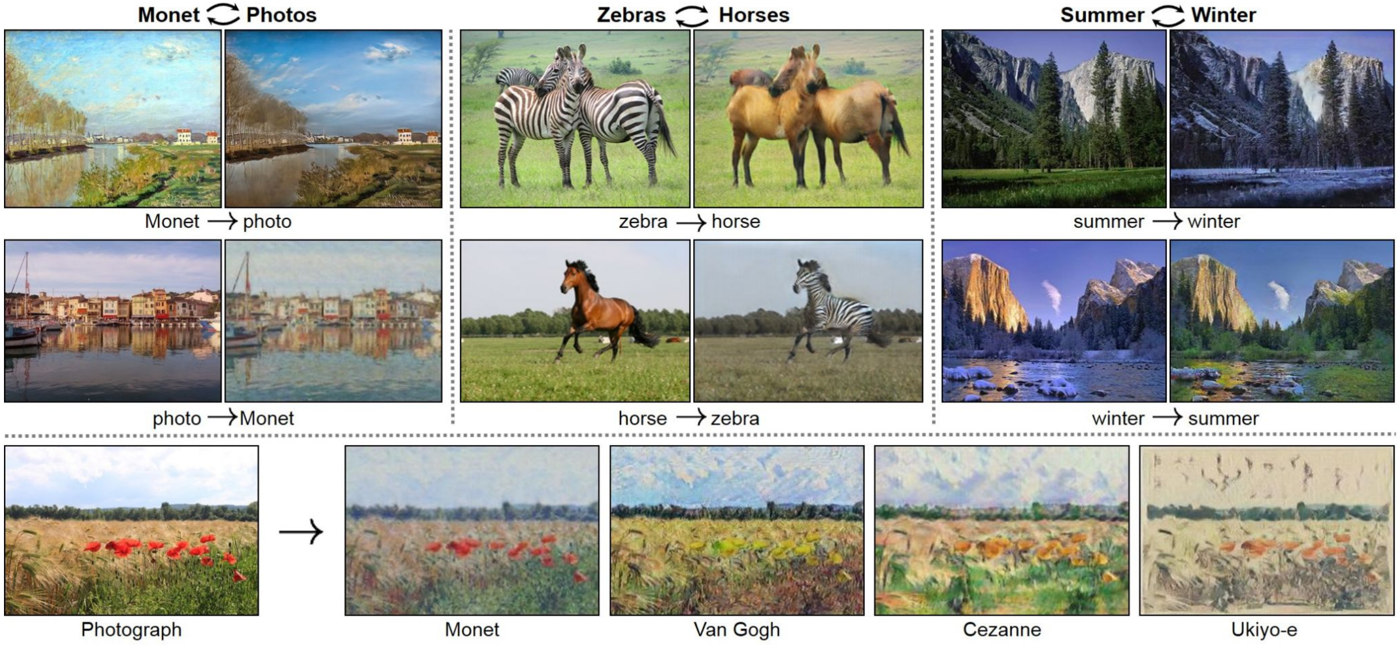
Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros
Image-to-Image Translation with Conditional Adversarial Networks, CVPR, 2017

Generative Adversarial Network (GAN)

Applications - Cycle GAN

Domain transformation: CycleGAN learns transformations across domains with unpaired data.




Monet ⇄ Photos

Zebras ⇄ Horses

Summer ⇄ Winter

Monet → photo

zebra → horse

summer → winter

photo → Monet

horse → zebra

winter → summer

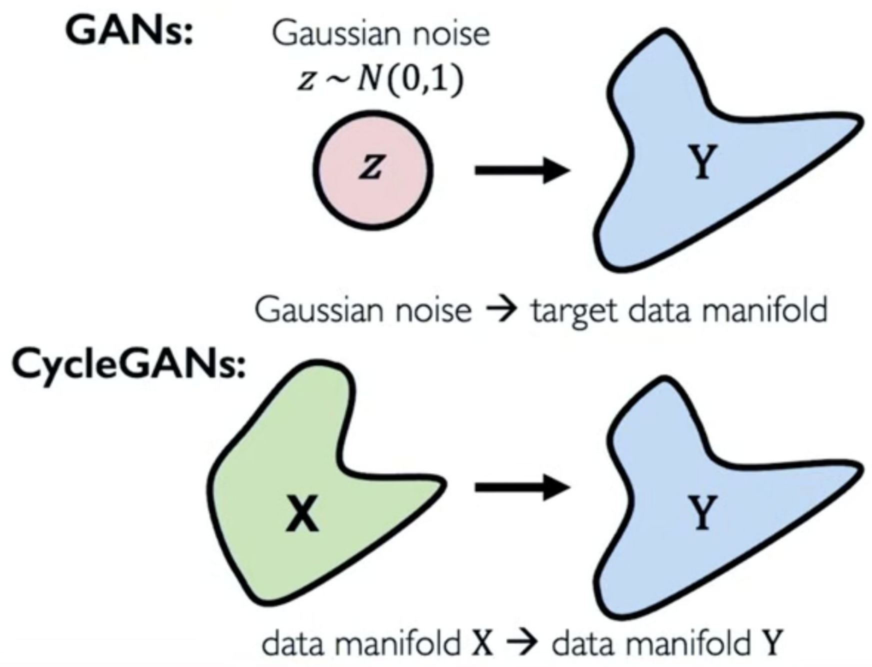
Photograph → Monet, Van Gogh, Cezanne, Ukiyo-e

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros
Image-to-Image Translation with Conditional Adversarial Networks, CVPR, 2017

Generative Adversarial Network (GAN)

Applications - Cycle GAN

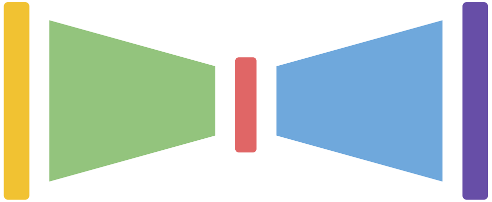
Distribution transformations.


GANs:
Gaussian noise
$z \sim N(0,1)$
z
Y
Gaussian noise → target data manifold
CycleGANs:
X
Y
data manifold X → data manifold Y

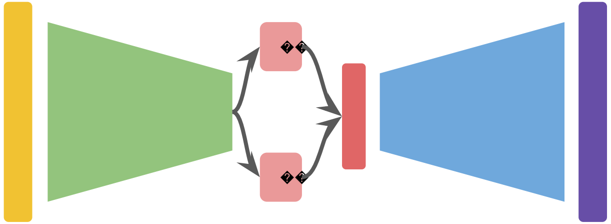
Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros
Image-to-Image Translation with Conditional Adversarial Networks, CVPR, 2017

Summarize

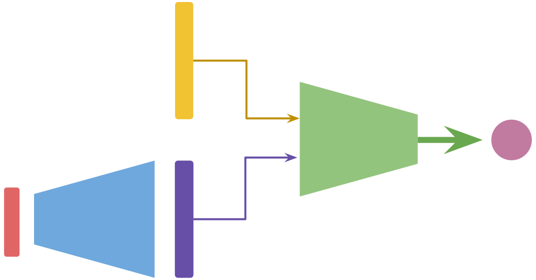
Autoencoder (AE)



Variational Autoencoder (VAE)



Generative Adversarial Network (GAN)



Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros
Image-to-Image Translation with Conditional Adversarial Networks, CVPR, 2017

Announcement

November 2, 2024

- No lecture

BUT

- There is a Lab session
  - k-Means
  - Mean Shift
  - UNet
  - AE/VAE
  - GAN