Introduction to Deep Generative Models

Strasbourg
24.09.2020

Robert Maria

Richard Feynman: "*What I cannot create, I do not understand*"

Generative modeling: "*What I understand, I can* ***create***"

CS236 Stanford

“Generative models are one of the most promising approaches towards computers that understand our world.”

OpenAI blog

“The most interesting idea in the last 10 years in machine learning.”

Yann LeCunn

Overview

1) What are Deep Generative Models (DGM)

2) What are they good for

3) Some principles behind DGM

4) How to evaluate DGM

5) Challenges

What are Generative Models

A generative model simulates the data generative process

- simulators
- graphical models
- probabilistic programs
- ordinary / partial differential equations

What are Generative Models

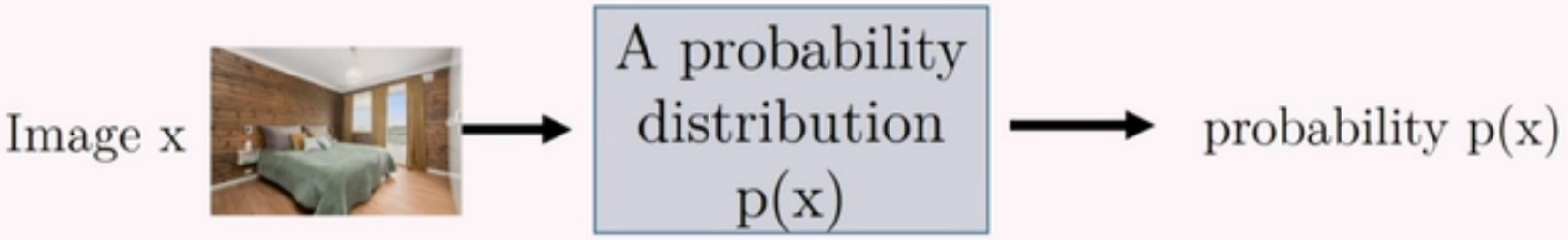
A generative model simulates the data generative process

- simulators
- graphical models
- probabilistic programs
- ordinary / partial differential equations

What are Statistical Generative Models

A statistical generative model is a **probability distribution** p(x)

- **Data:** samples (e.g., images of bedrooms)
- **Prior knowledge:** parametric form (e.g., Gaussian?), loss function (e.g., maximum likelihood?), optimization algorithm, etc.



It is generative because **sampling from p(x) generates new images**

1) What are Deep Generative Models

Don’t just extract patterns in data, but use those patterns to learn the underlying distribution of that data and use it to generate new data.

1) What are Deep Generative Models

Don’t just extract patterns in data, but use those patterns to learn the underlying distribution of that data and use it to generate new data.

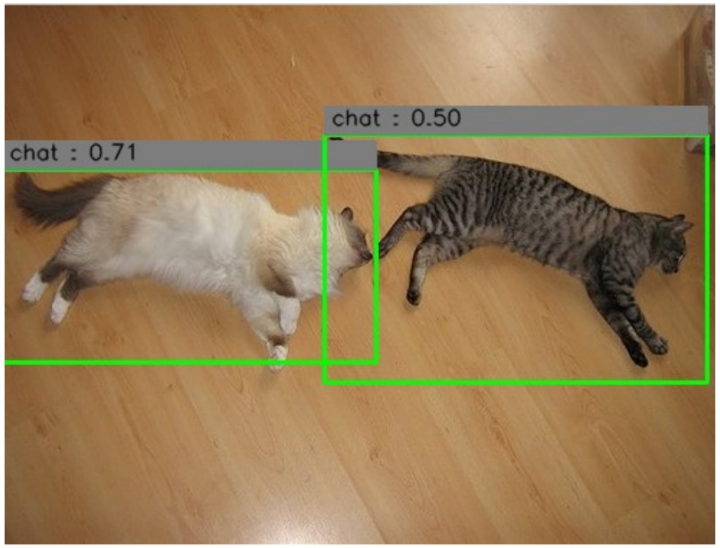
Why Deep Generative Models?

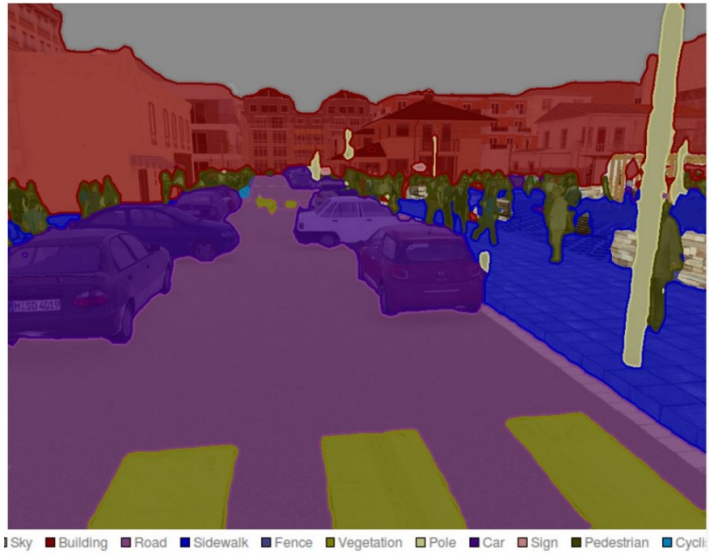
The idea behind the recent progress of generative modeling is to convert the generation problem to a prediction one and use deep learning algorithms to learn such a problem.

source

Discriminative vs Generative

Some tasks based on discriminative learning:
classification, detection, semantic segmentation, etc.





Supervised learning setup: we provide data x and label y

Discriminative vs Generative

Discriminative: classify bedroom vs. dining room

The input image X is always given. **Goal**: a good decision boundary, via **conditional distribution**

P(Y = Bedroom | X=  1

Discriminative vs Generative

Discriminative: classify bedroom vs. dining room

The input image X is always given. **Goal**: a good decision boundary, via **conditional distribution**

P(Y = Bedroom | X= 

Generative: generate X

Y=B , X=  Y=D , X= 

Y=B , X=  Y=D , X= 

... ...

Y=B , X=  Y=D , X= 

The input X is **not** given. Requires a model of the **joint distribution**

P(Y = Bedroom , X= 

Stanford CS 236

Discriminative vs Generative

Joint and conditional are related via **Bayes Rule**:

$$P(Y = \text{Bedroom} \mid X = \text{  }) = \frac{P(Y = \text{Bedroom}, X = \text{  })}{P(X = \text{  })}$$

Discriminative: X is always given, does not need to model $P(X = \text{  })$

Stanford CS 236

Types of Generative Models

Likelihood-based frameworks

- autoregressive models
- variational autoencoders
- flow models

Types of Generative Models

Likelihood-based frameworks

- autoregressive models
- variational autoencoders
- flow models

Generative Adversarial Networks

Types of Generative Models

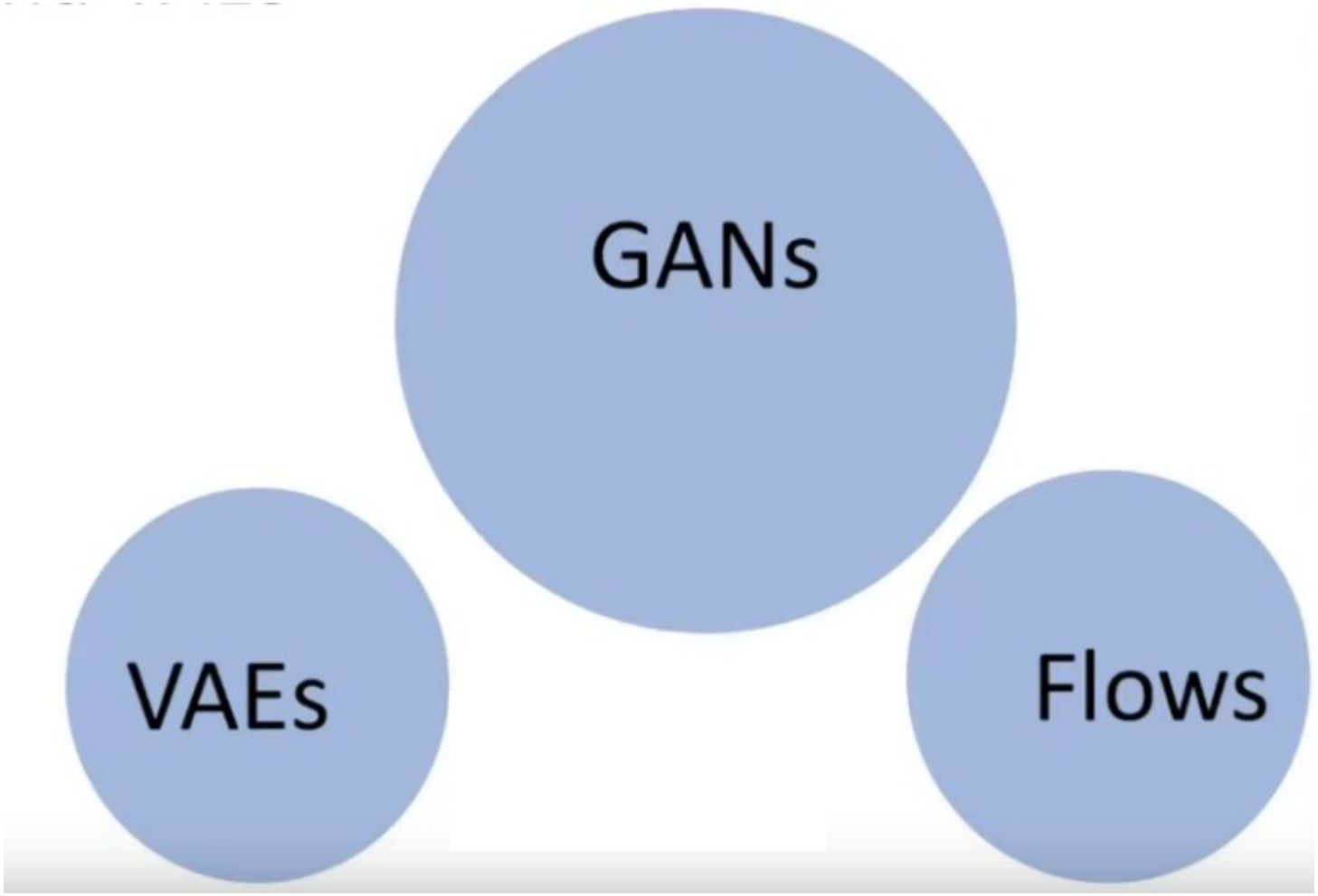
Likelihood-based frameworks

- autoregressive models
- variational autoencoders
- flow models

Generative Adversarial Networks

Score function (estimating gradients of the data distribution)

Most popular methods today



M Welling,
PTSGM@ECCV2020

Overview

1) What are Deep Generative Models (DGM)

2) What are they good for

3) Some principles behind DGM

4) How to evaluate DGM

5) Challenges

2) What are they good for

2) What are they good for

Generate images

2) What are they good for

Generate images

Generate audio

source

2) What are they good for

Generate images

Generate audio

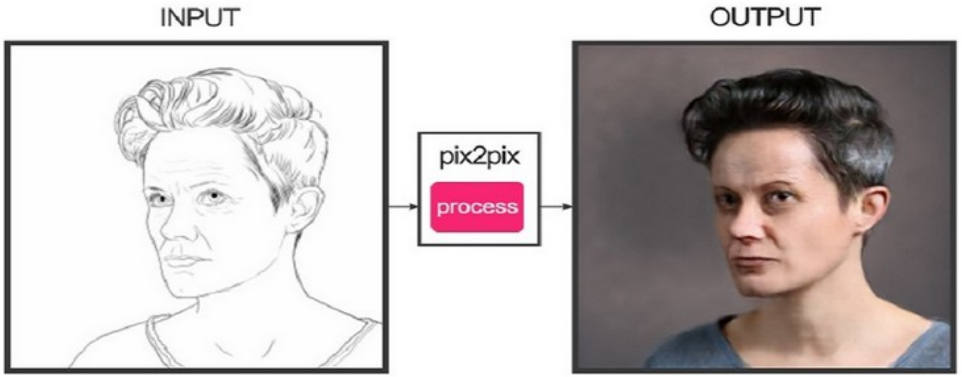

source

Generate text

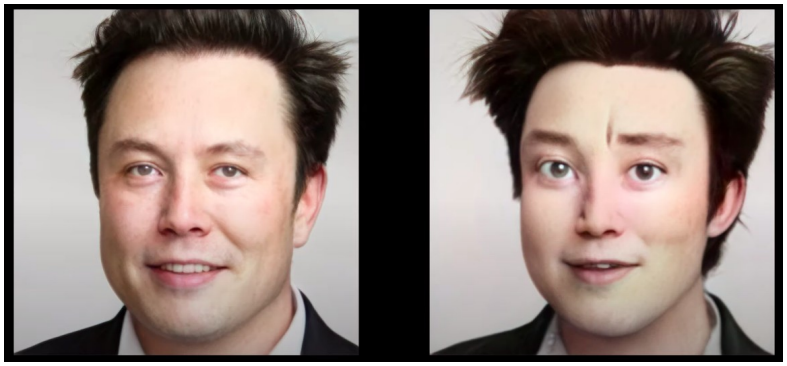
In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English. They also were found to have perfectly coiffed hair, and wore what appeared to be Dior makeup.

source

2) What are they good for

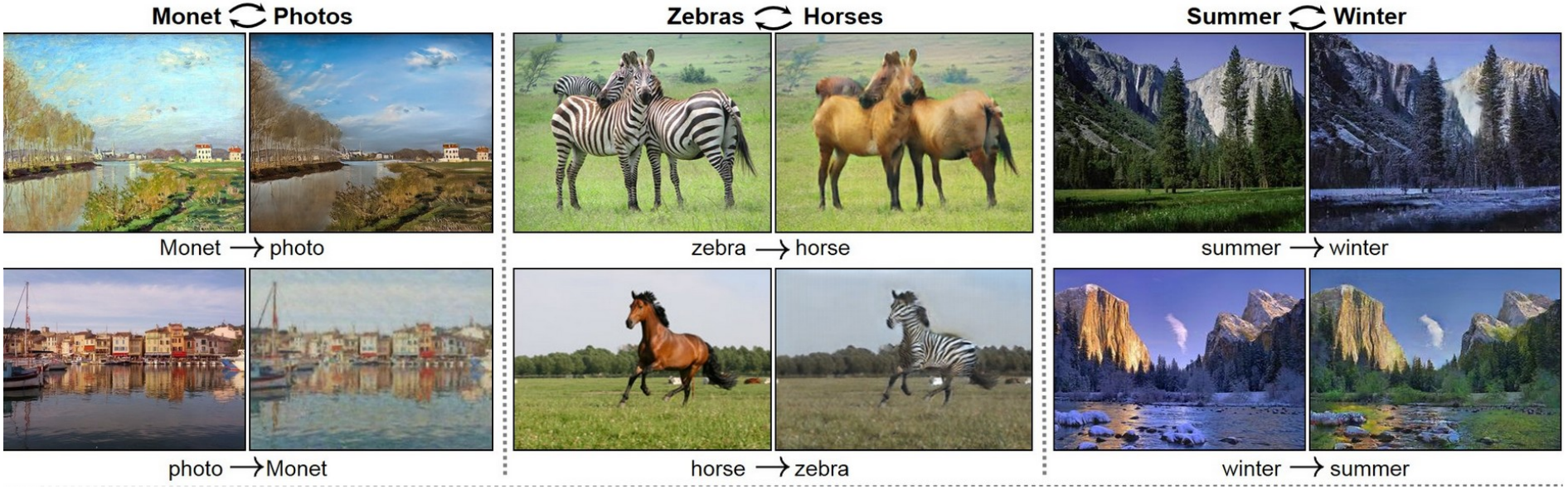


source



source

source



source

2) What are they good for

De-biasing datasets, generate synthetic data for skewed datasets

Homogeneous skin color, pose

VS

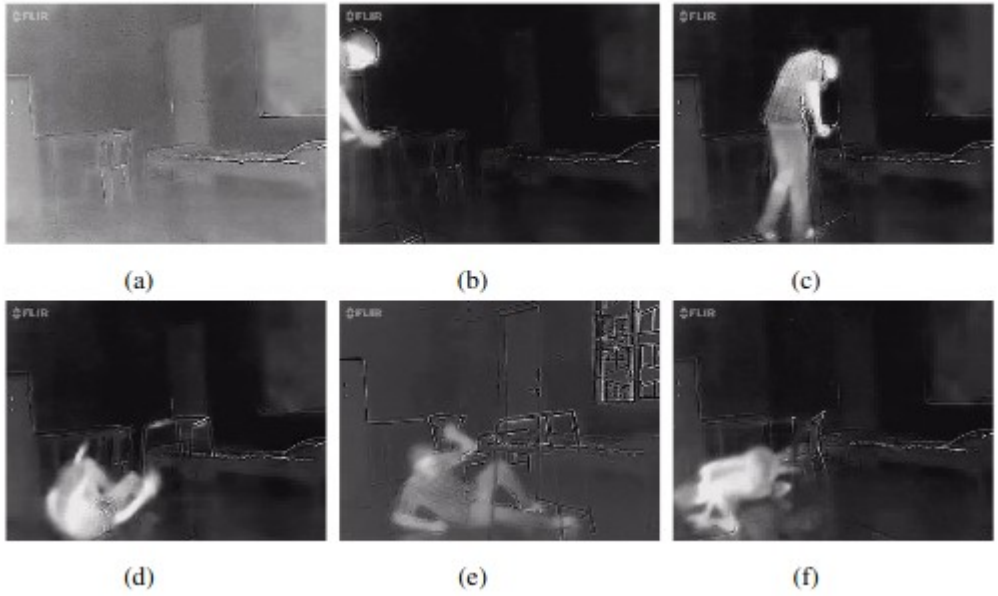
Diverse skin color, pose, illumination

MIT S191

If we manage to capture the underlying factors of disentanglement.

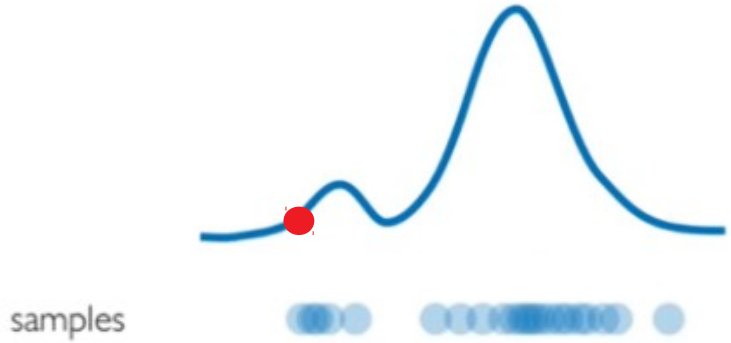
2) What are they good for

Anomaly detection



source

Density estimation



Detect falls as anomalies to the activities of daily living.

Overview

1) What are Deep Generative Models (DGM)

2) What are they good for

3) Some principles behind DGM

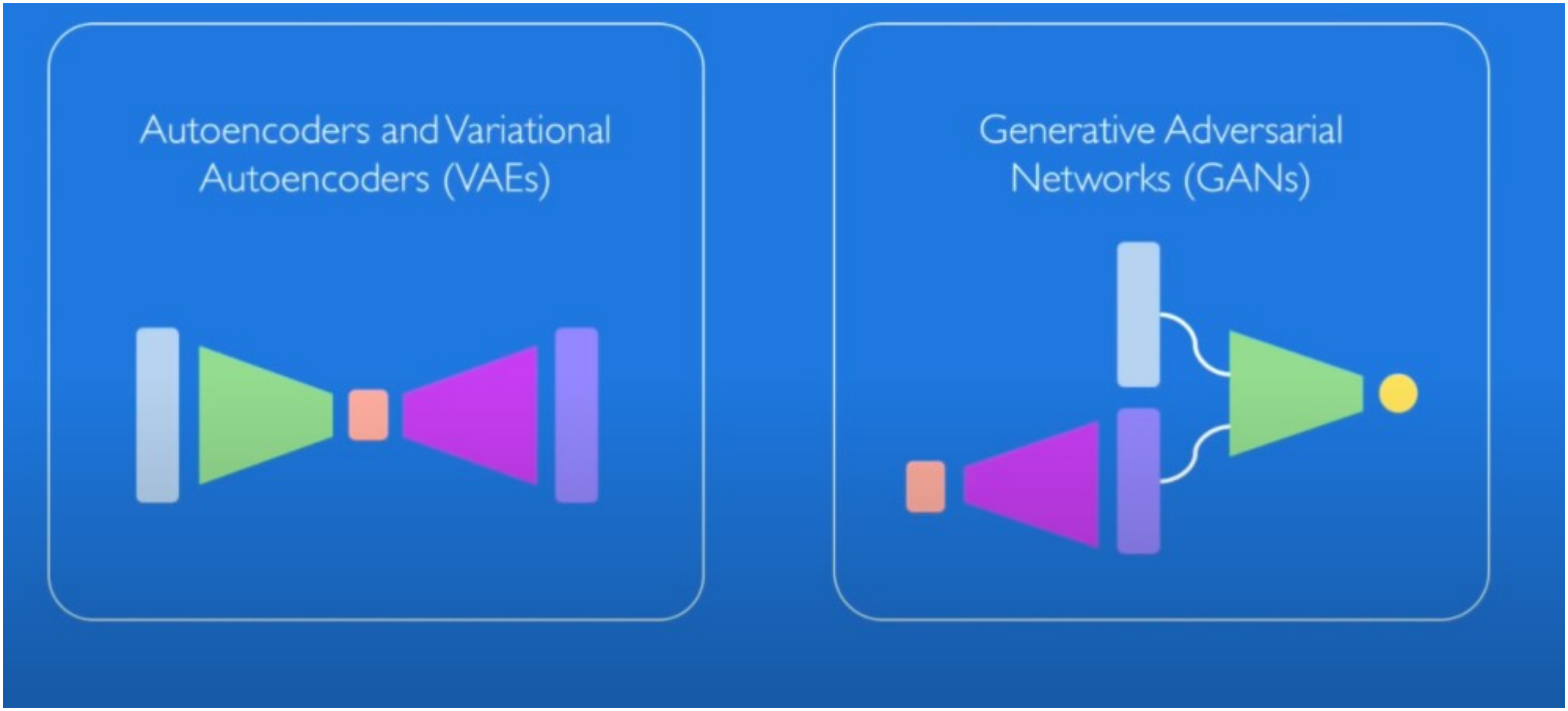
4) How to evaluate DGM

5) Challenges

What is a latent variable?

MIT 6.S191

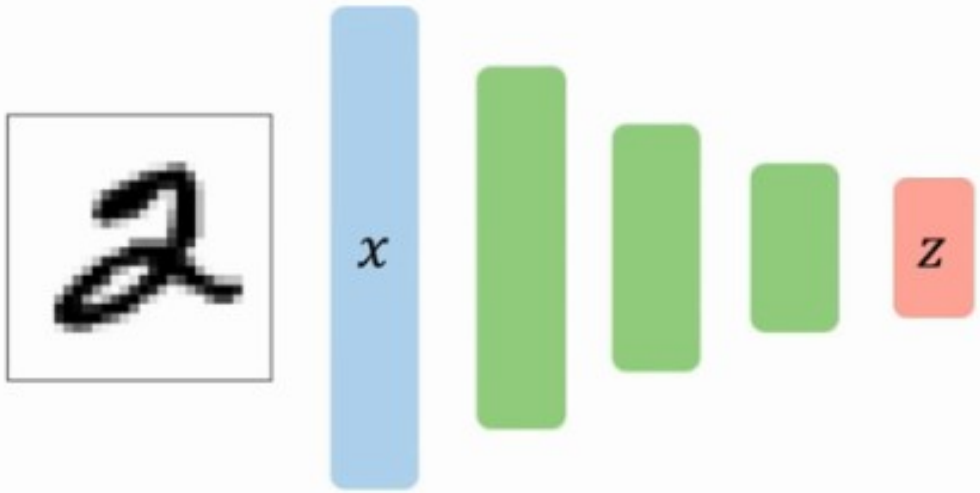
Latent variable models



MIT 6.S191

Autoencoders

Unsupervised approach for learning a **lower-dimensional** feature representation from unlabeled training data

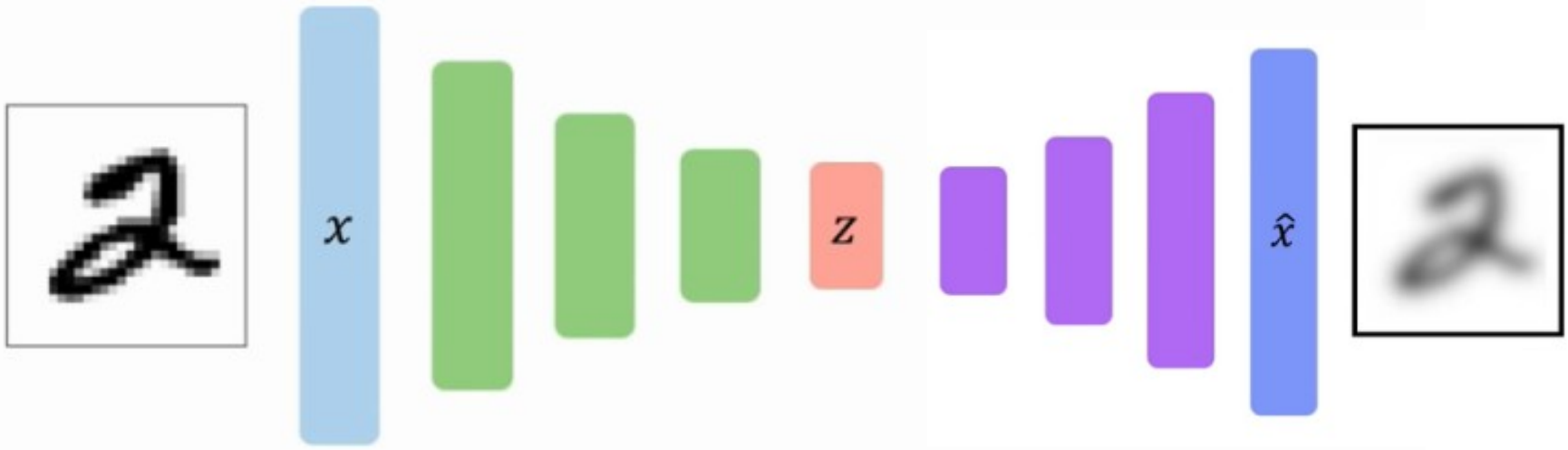


S191

"Encoder" learns mapping from the data, x, to a low-dimensional latent space, z

Autoencoders

Unsupervised approach for learning a **lower-dimensional** feature representation from unlabeled training data



MIT 6.S191

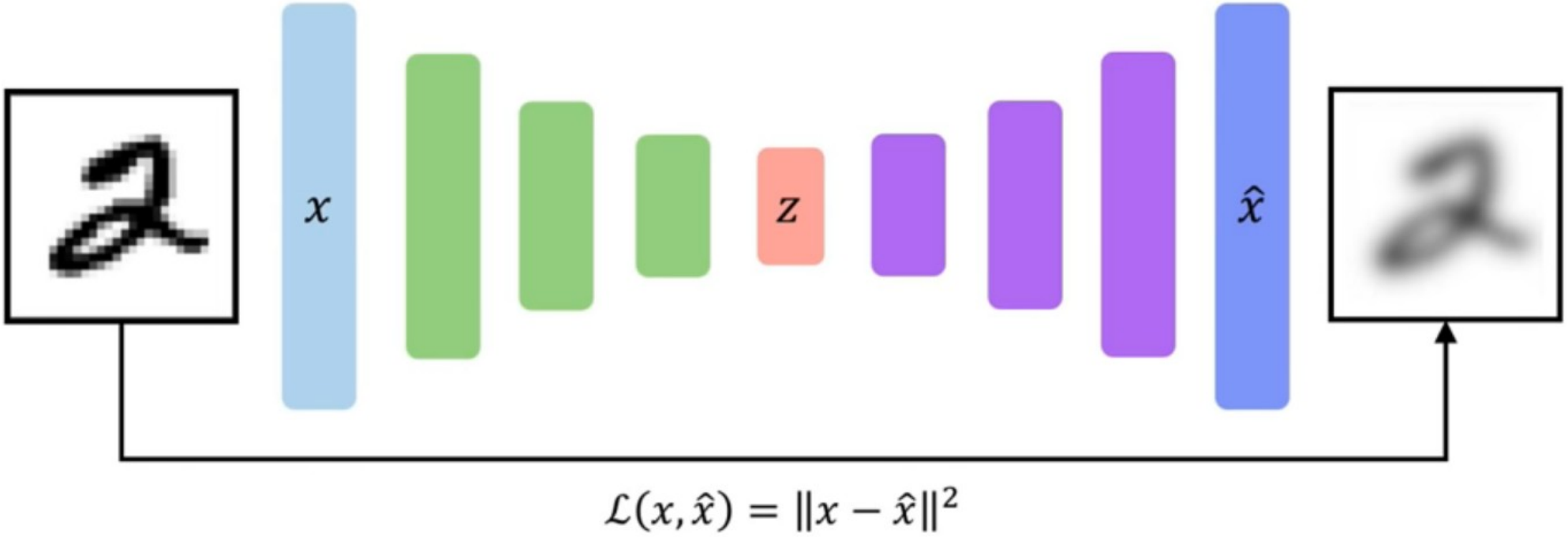
"Encoder" learns mapping from the data, x, to a low-dimensional latent space, z

"Decoder" learns mapping back from latent, z, to a reconstructed observation, $\hat{x}$

Autoencoders

How can we learn this latent space?

Train the model to use these features to **reconstruct the original data**



$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2$$

MIT 6.S191

Autoencoders

Autoencoding is a form of compression!
Smaller latent space will force a larger training bottleneck

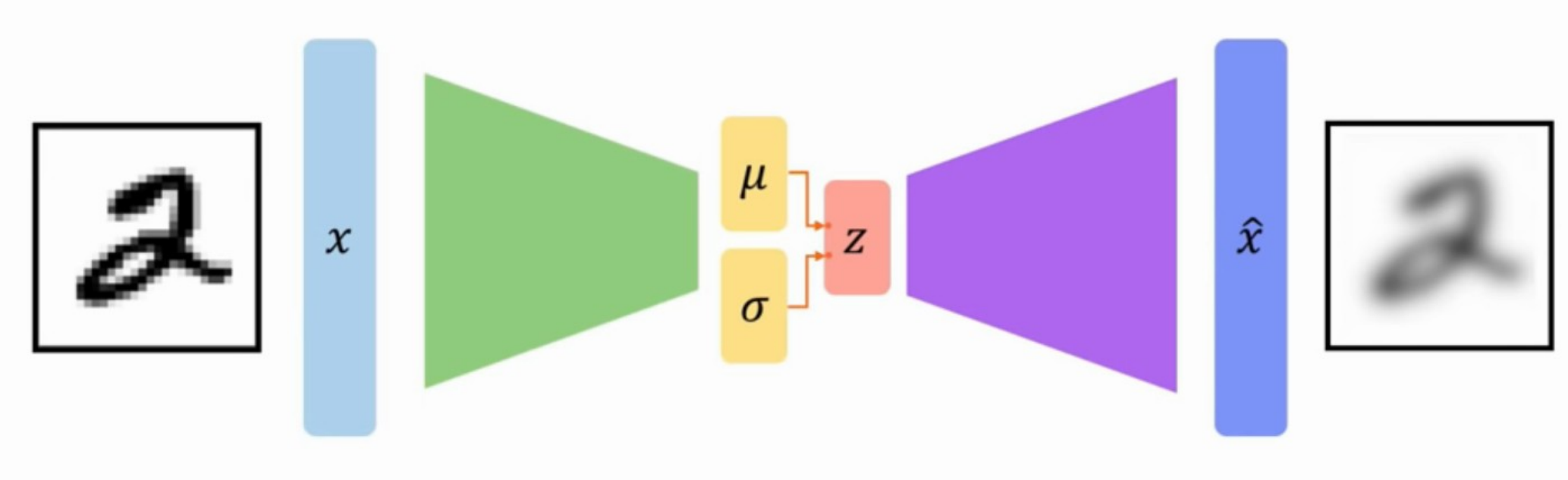
2D latent space

5D latent space

Ground Truth

MIT 6.S191

Variational Autoencoders

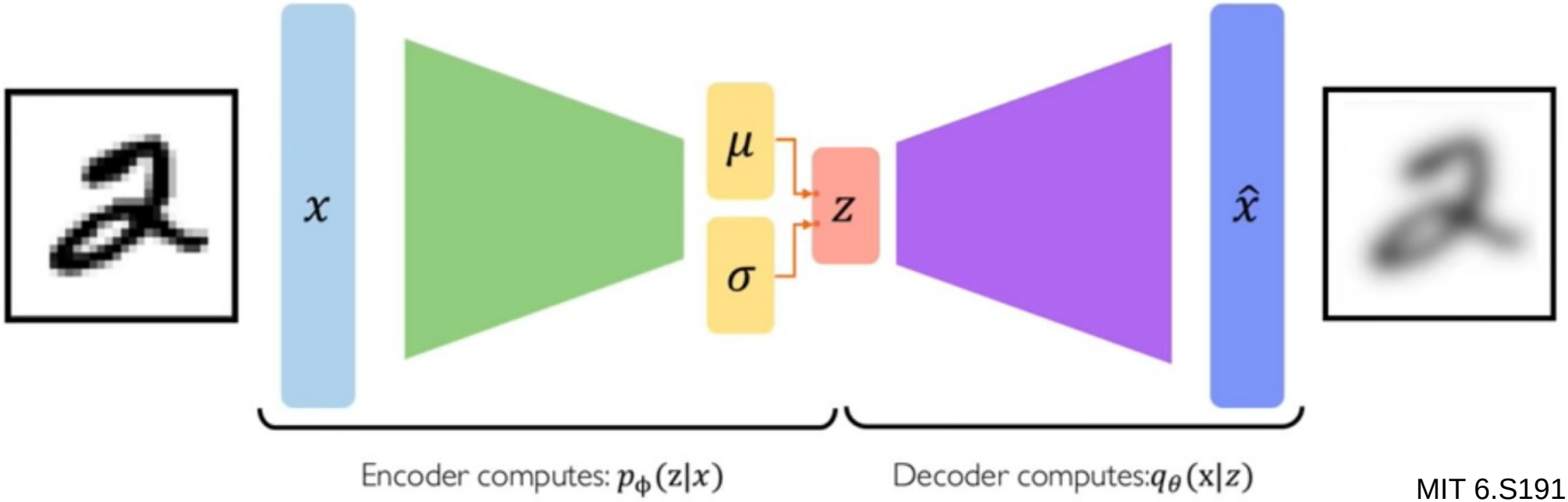


MIT 6.S191

Variational autoencoders are a probabilistic twist on autoencoders!

Sample from the mean and standard dev. to compute latent sample

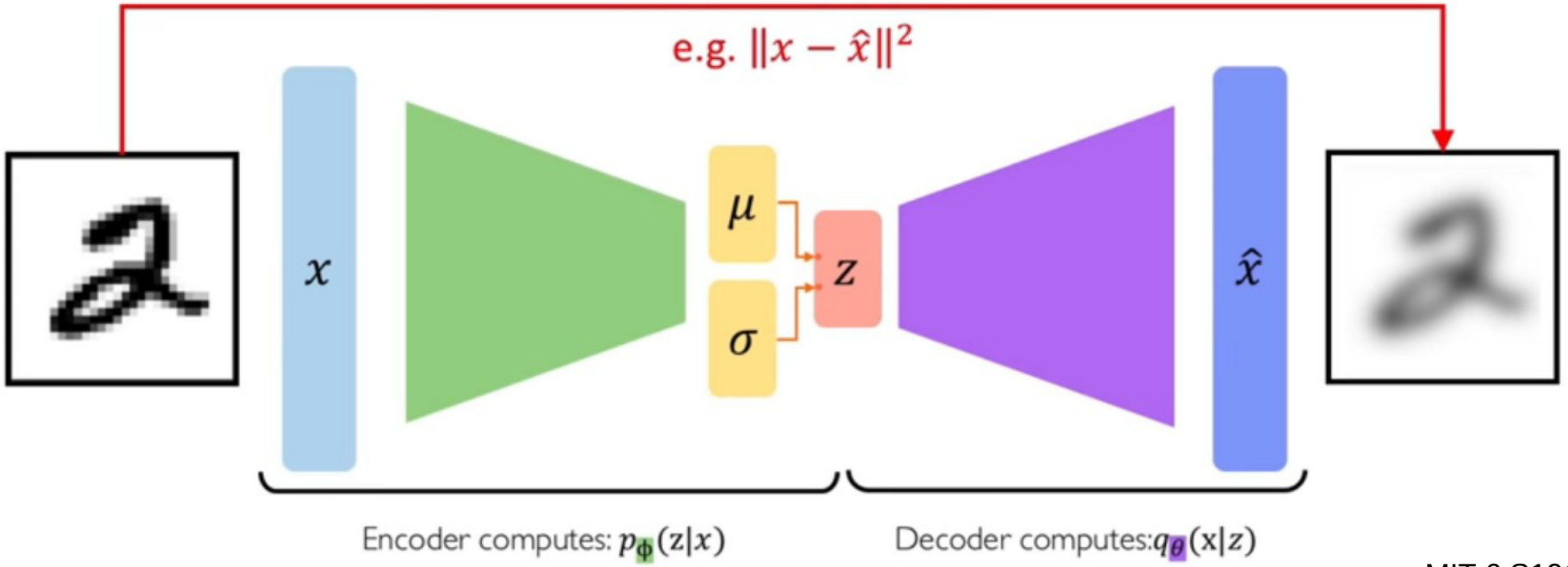
Variational Autoencoders



MIT 6.S191

$$\mathcal{L}(\phi, \theta) = \text{(reconstruction loss)} + \text{(regularization term)}$$

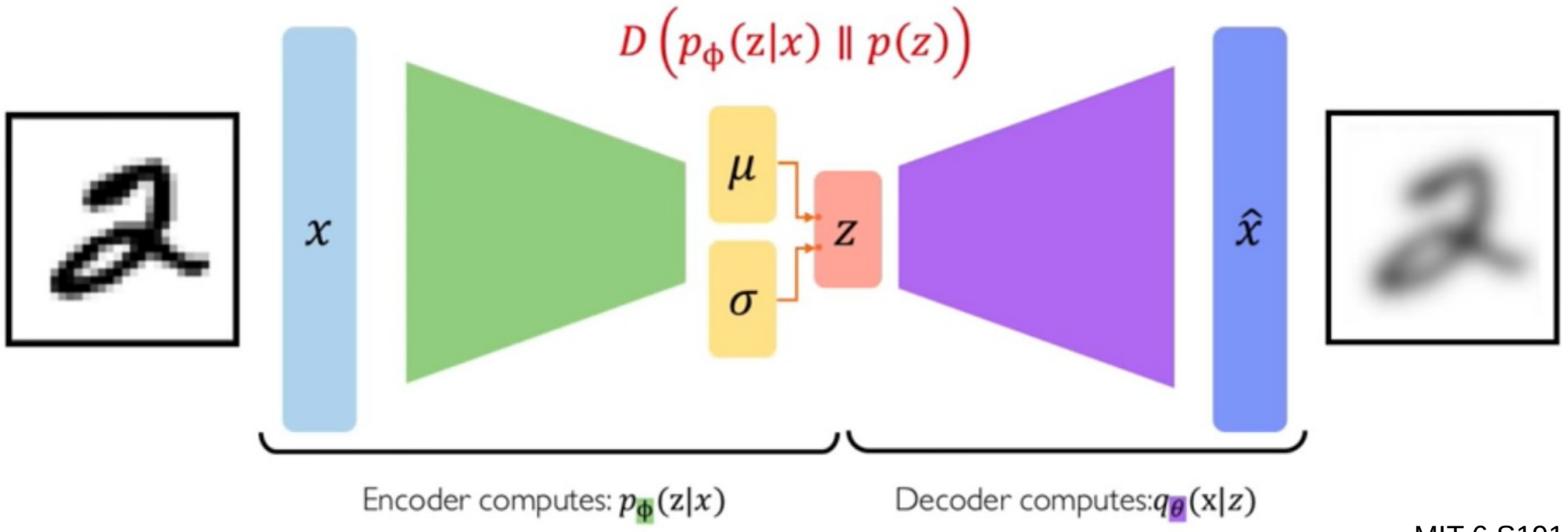
Variational Autoencoders



MIT 6.S191

$$\mathcal{L}(\phi, \theta, x) = \boxed{\text{(reconstruction loss)}} + \text{(regularization term)}$$

Variational Autoencoders



MIT 6.S191

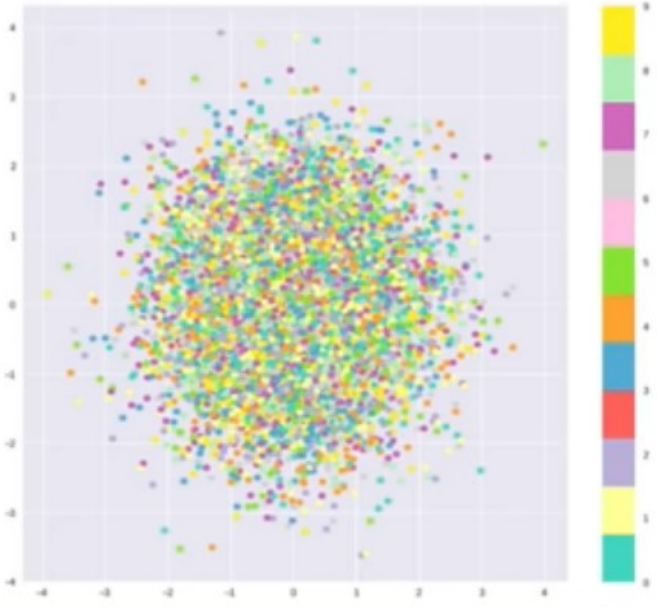
$$\mathcal{L}(\phi, \theta, x) = \text{(reconstruction loss)} + \boxed{\text{(regularization term)}}$$

Priors on the latent distribution

$$D\left(p_\phi(z|x) \parallel p(z)\right) = -\frac{1}{2}\sum_{j=0}^{k-1}\left(\sigma_j + \mu_j^2 - 1 - \log \sigma_j\right)$$

Inferred latent distribution

Fixed prior on latent distribution



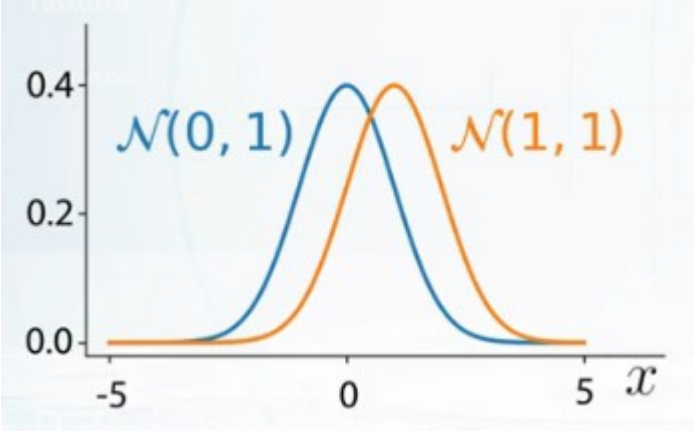
Common choice of prior:

$$p(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

- Encourages encodings to distribute encodings evenly around the center of the latent space
- Penalize the network when it tries to "cheat" by clustering points in specific regions (ie. memorizing the data)

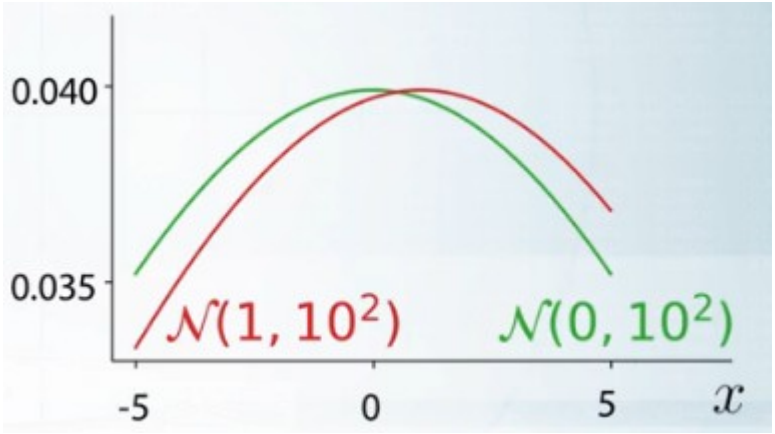
Kullback–Leibler divergence

- is a measure of how one probability distribution is different from a second, reference probability distribution



Parameters difference: 1

$$\mathcal{KL}(q_1 \parallel p_1) = 0.5$$



Parameters difference: 1

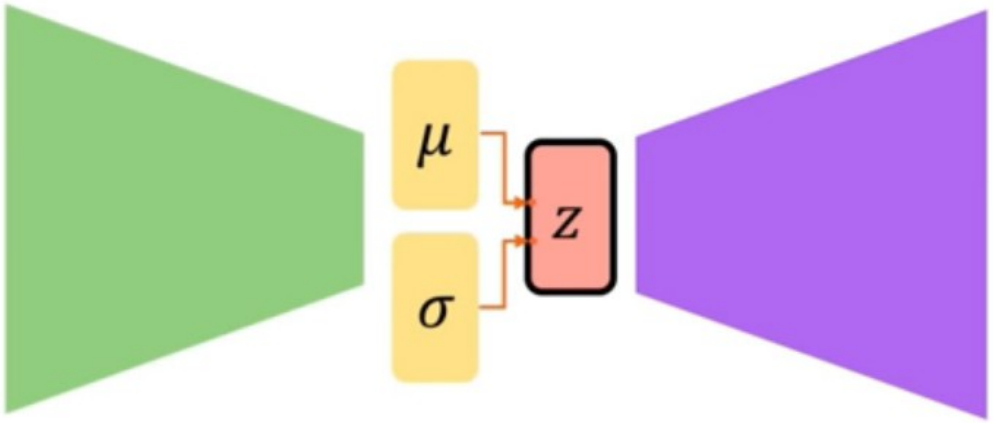
$$\mathcal{KL}(q_2 \parallel p_2) = 0.005$$

source

$$\mathcal{KL}(q \parallel p) = \int q(x) \log \frac{q(x)}{p(x)} dx$$

Not a distance, not symmetric!

Reparameterization trick



Key Idea:

~~$z \sim \mathcal{N}(\mu, \sigma^2)$~~

Consider the sampled latent vector z as a sum of

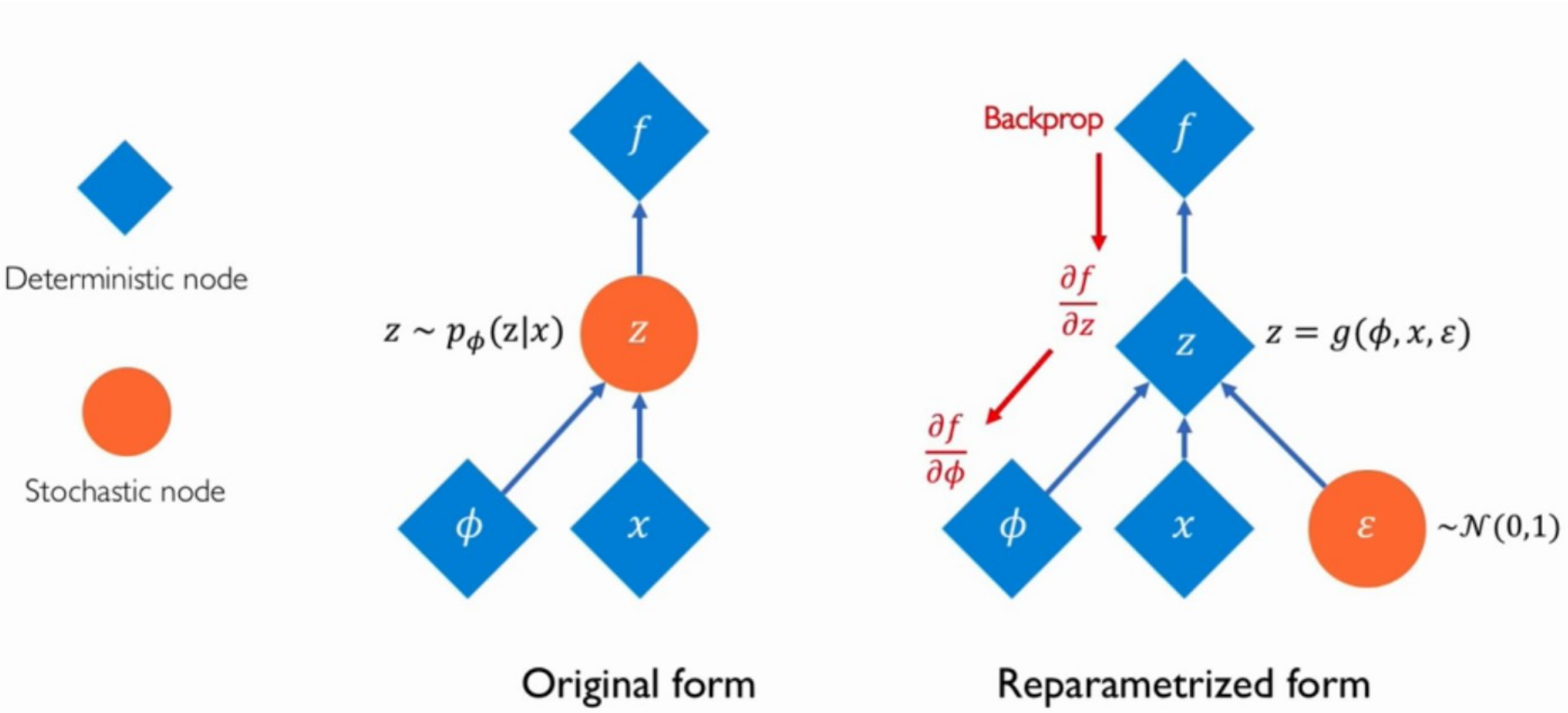
- a fixed μ vector,
- and fixed σ vector, scaled by random constants drawn from the prior distribution

$$\Rightarrow z = \mu + \sigma \odot \varepsilon$$

where $\varepsilon \sim \mathcal{N}(0,1)$

MIT 6.S191

Reparameterization trick

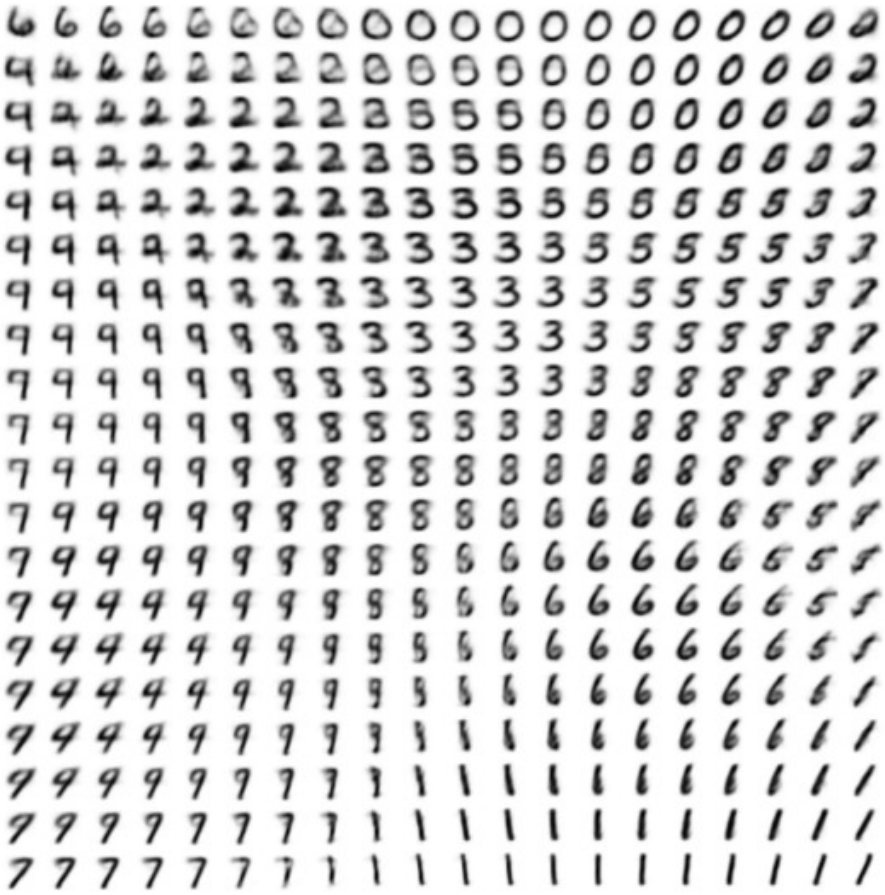


MIT 6.S191

Latent perturbation

Increase ordecrease a single latent variable, keeping all others fixed.

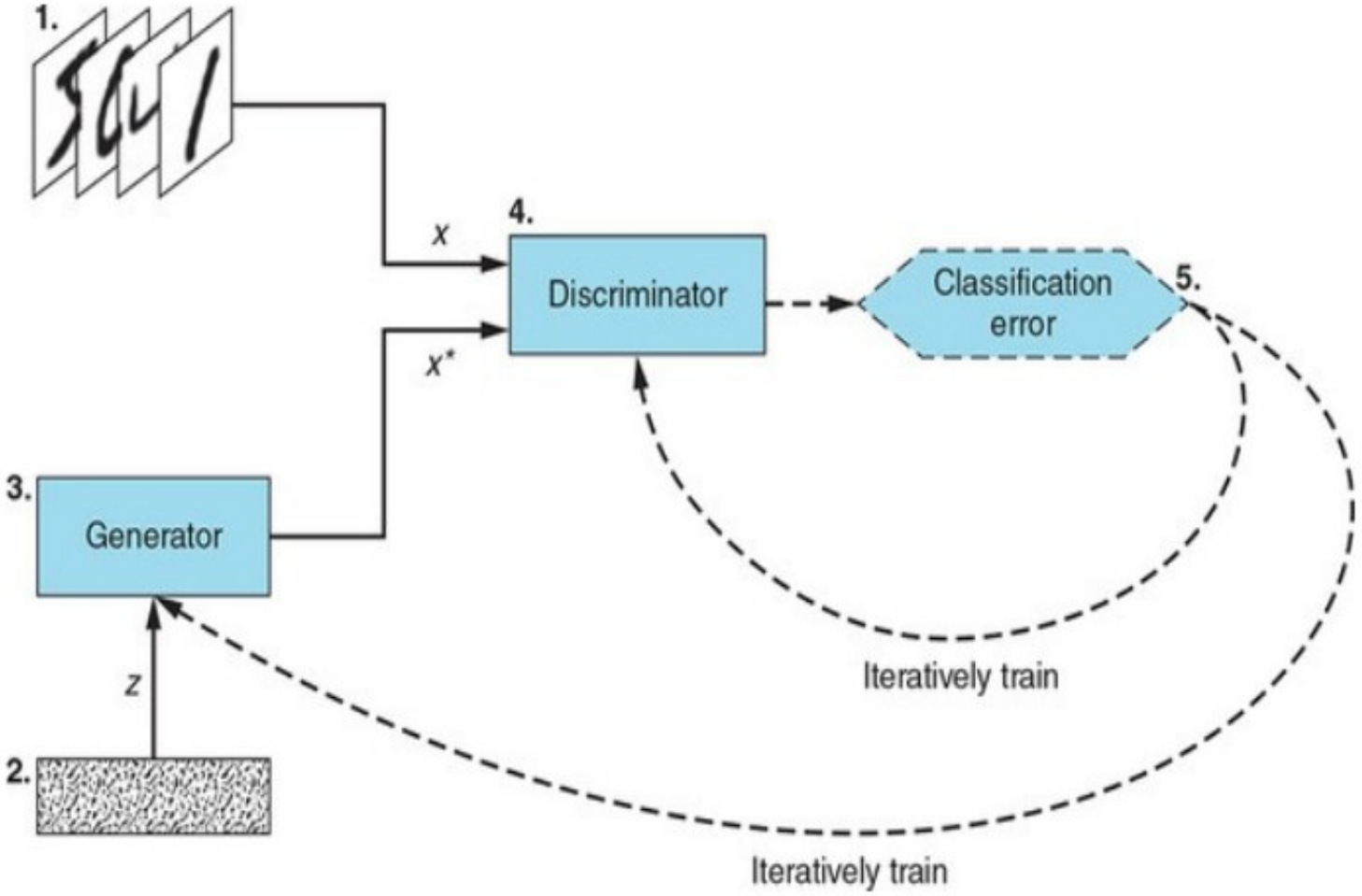
(a) Learned Frey Face manifold



(b) Learned MNIST manifold

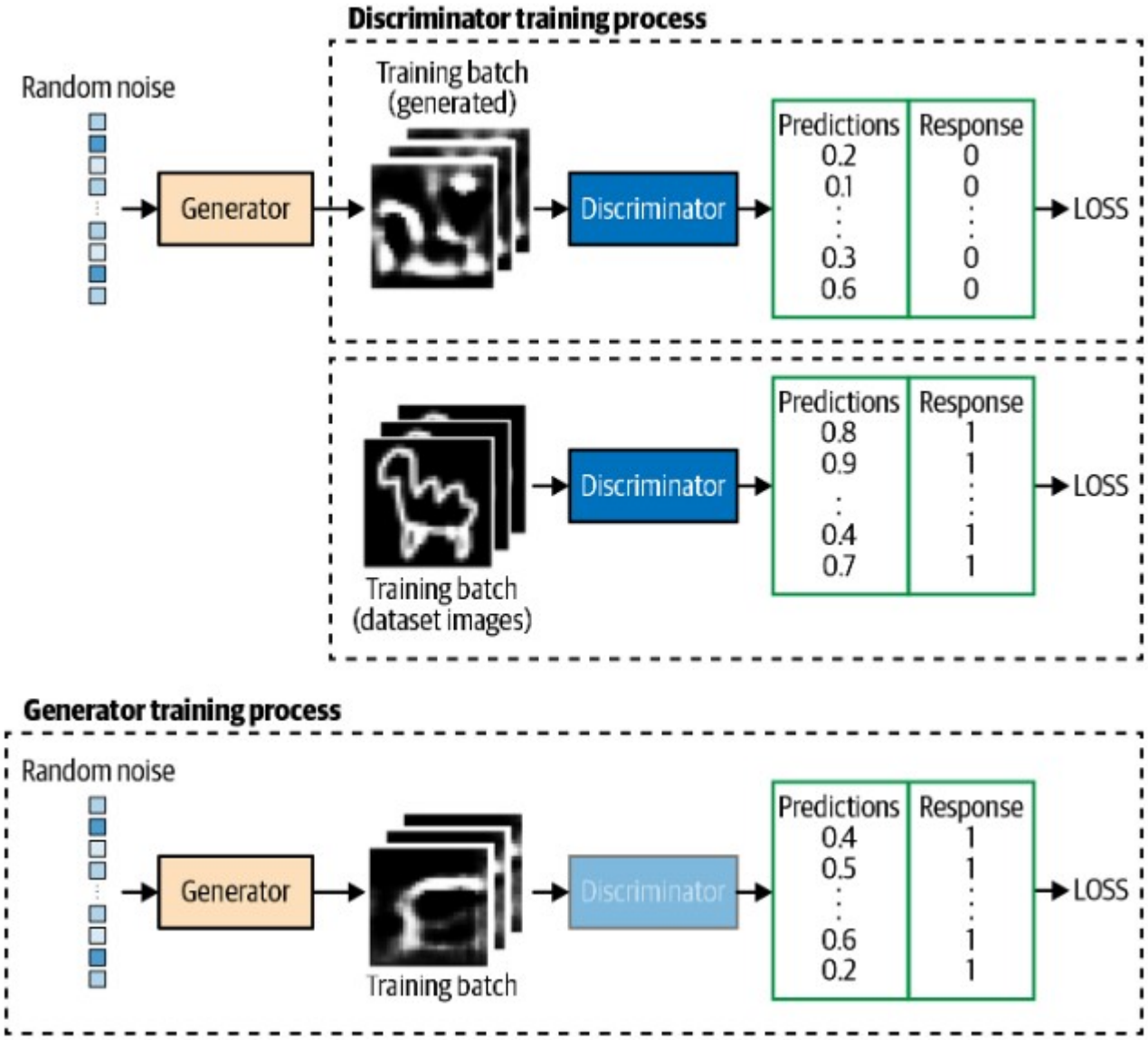
source

Generative Adversarial Neural Networks (GANs)



source

Training GANs



source

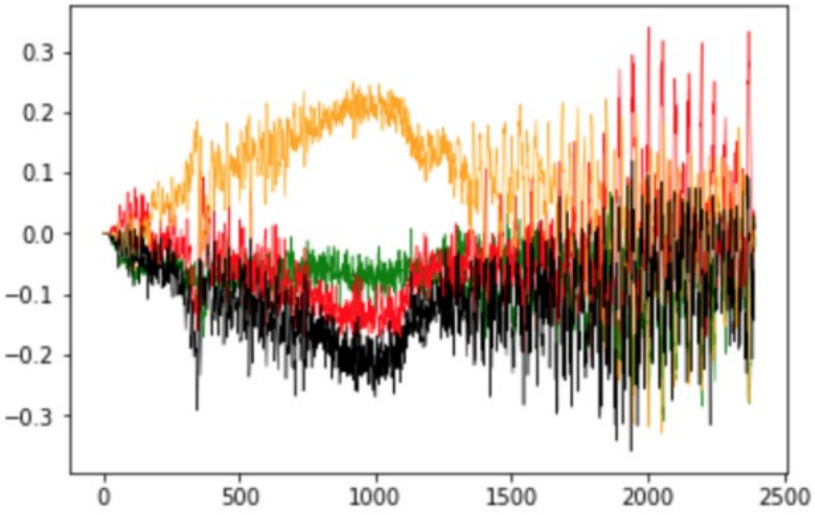
When the training loop is meant to stop?

- The Generator produces fake examples that are indistinguishable from the real data in the training dataset

- The Discriminator can at best randomly guess whether a particular example is real or fake

Challenges for training GANs

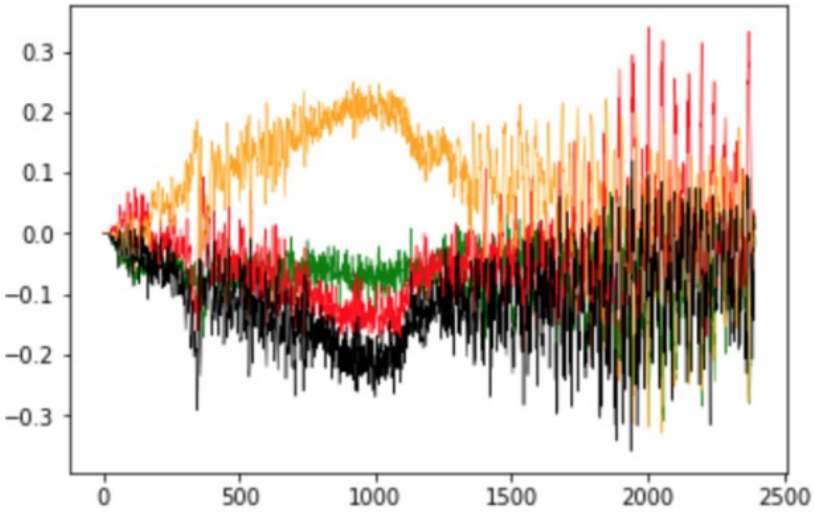
- oscillating loss



source

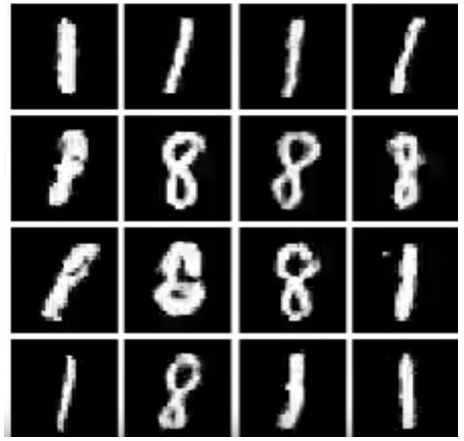
Challenges for training GANs

- oscillating loss



source

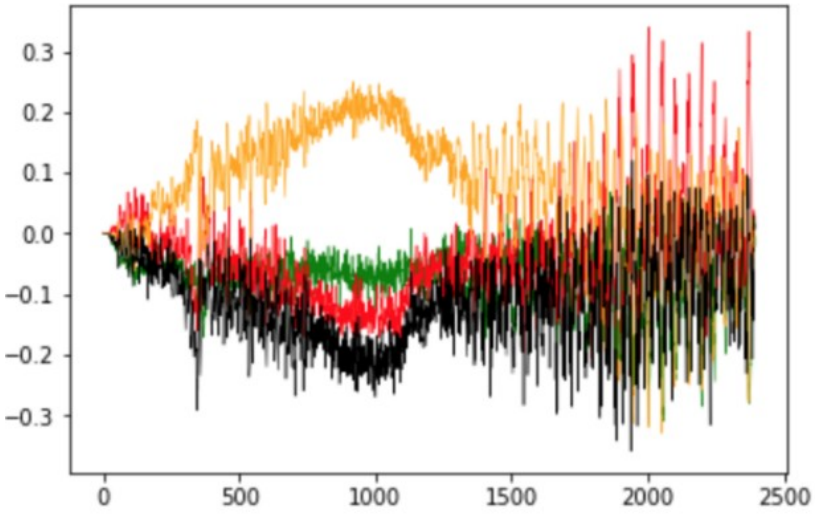
- mode collapse



source

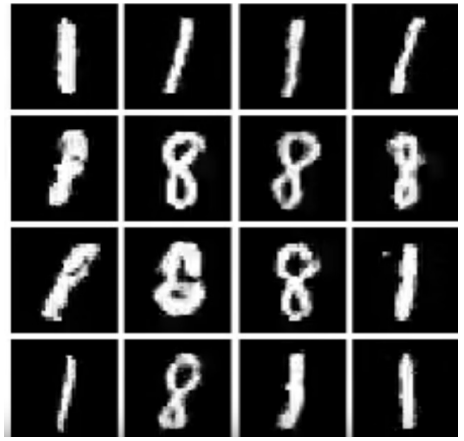
Challenges for training GANs

- oscillating loss



source

- mode collapse



source

- vanishing gradients

- failure to converge

Overview

1) What are Deep Generative Models (DGM)

2) What are they good for

3) Some principles behind DGM

4) How to evaluate DGM

5) Challenges

How to asses the quality and diversity of the generated synthetic images?

How to evaluate Generative Models

Problems with human annotations

- subjectivity

How to evaluate Generative Models

Problems with human annotations

- subjectivity

- have high variance

How to evaluate Generative Models

Problems with human annotations

- subjectivity

- have high variance

- time consuming / expensive

How to evaluate Generative Models

Problems with human annotations

- subjectivity

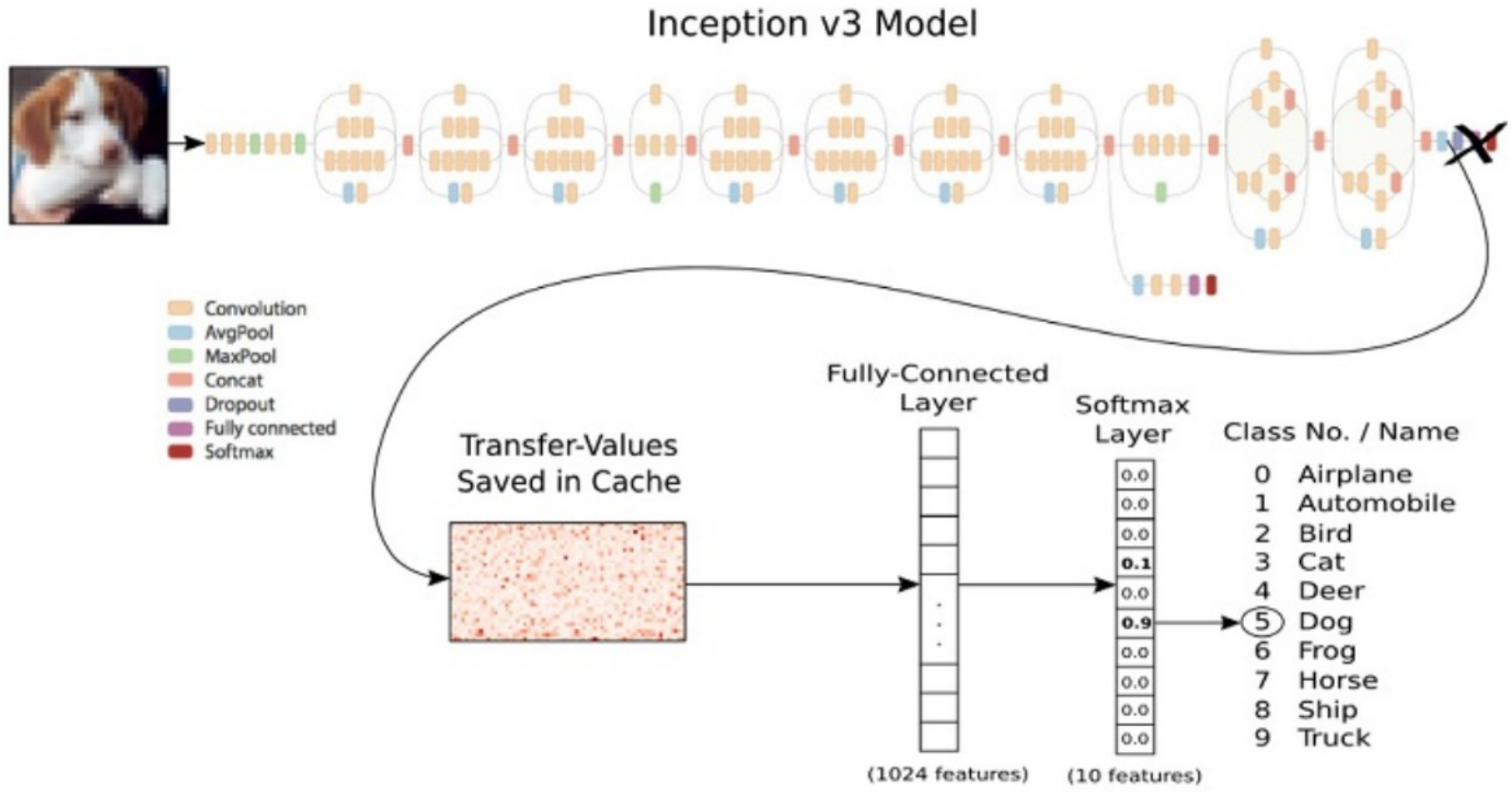
- have high variance

- time consuming / expensive

- generative models can overfit and underfit in the same time

- humans can hardly detect mode collapse or overfitting

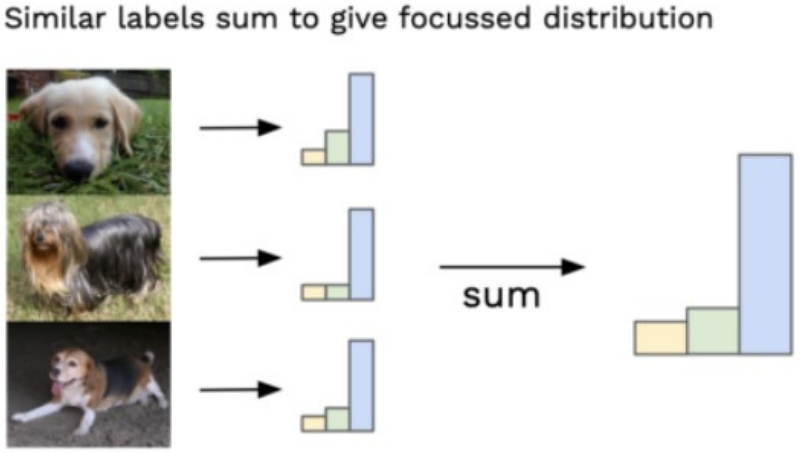
The Inception Score (IS)



source

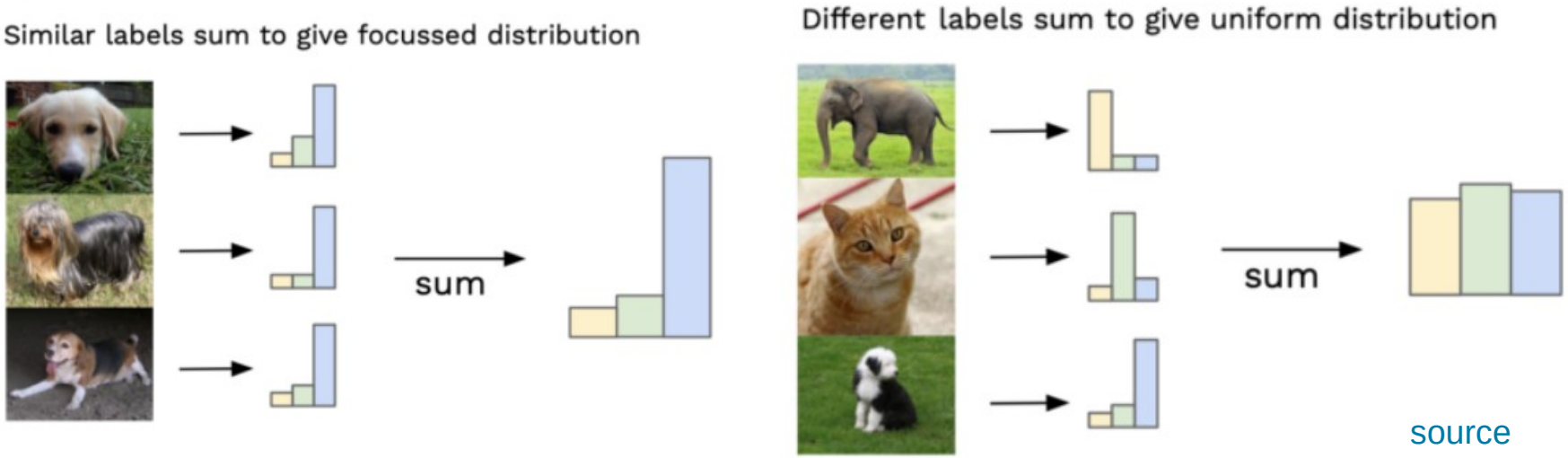
Assumption: we have a good probabilistic classifier c(y|x)

The Inception Score (IS)



- high conficence for the correct class

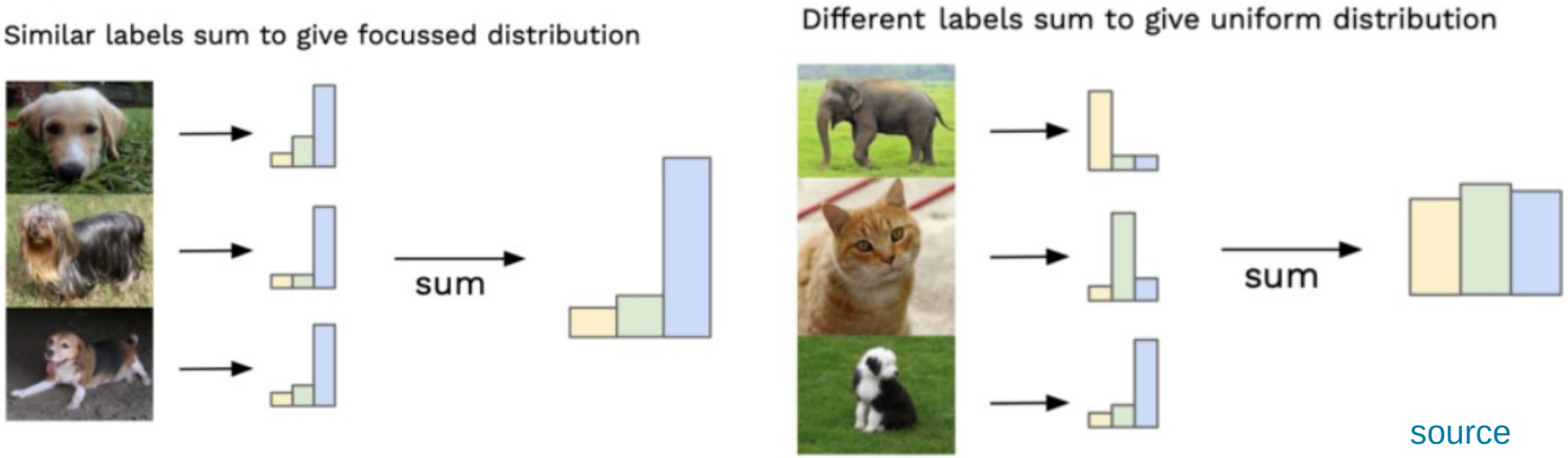
The Inception Score (IS)



- high conficence for the correct class

- variety in the output of the generator

The Inception Score (IS)



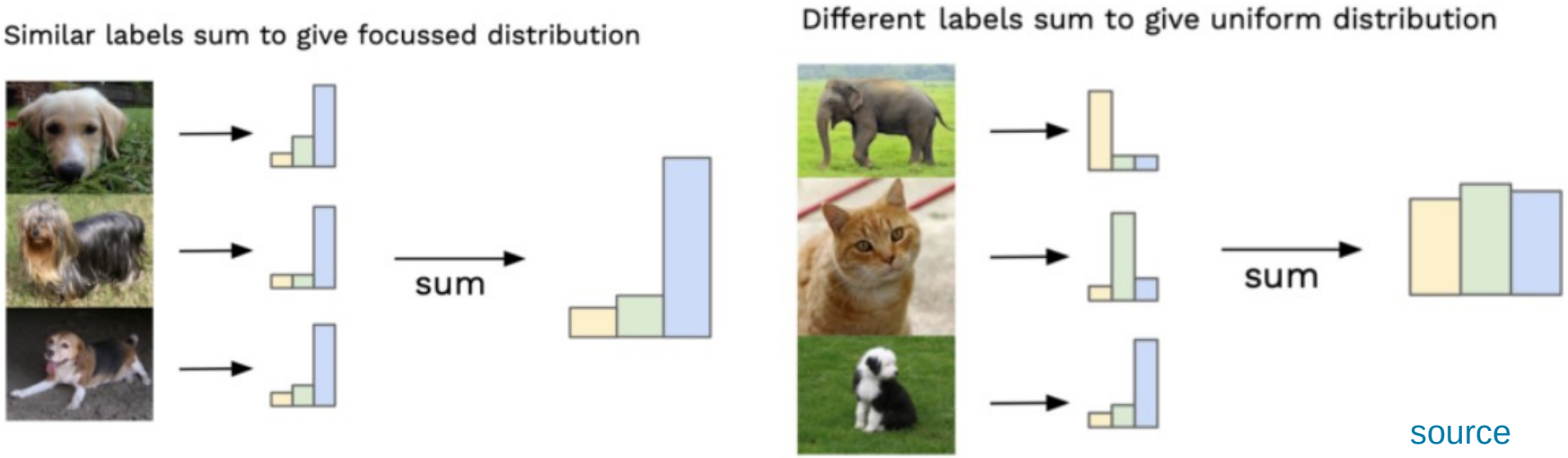
- high conficence for the correct class

- variety in the output of the generator

$$S = \exp\left(E_{\mathbf{x}\sim p}\left[\int c(y|\mathbf{x})\log c(y|\mathbf{x})\mathrm{d}y\right]\right)$$

Sharpness should be high

The Inception Score (IS)



- high conficence for the correct class

$$S = \exp\left(E_{\mathbf{x}\sim p}\left[\int c(y|\mathbf{x}) \log c(y|\mathbf{x}) \mathrm{d}y\right]\right)$$

Sharpness should be high

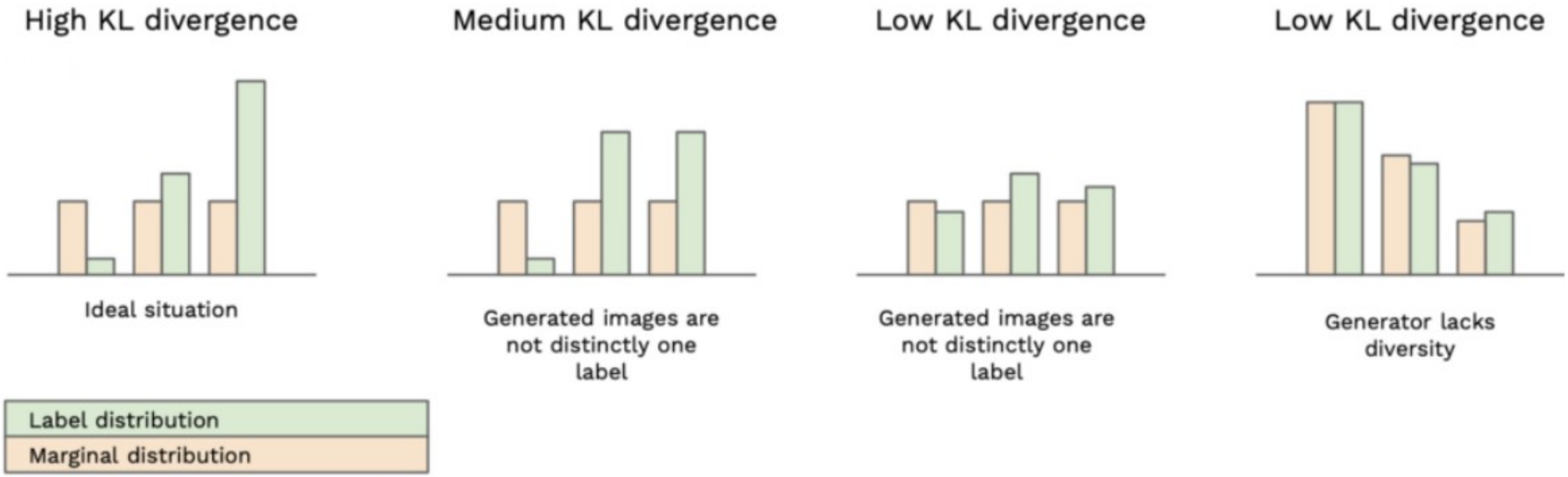
- variety in the output of the generator

$$D = \exp\left(-E_{\mathbf{x}\sim p}\left[\int c(y|\mathbf{x}) \log c(y) \mathrm{d}y\right]\right)$$

Diversity should be high

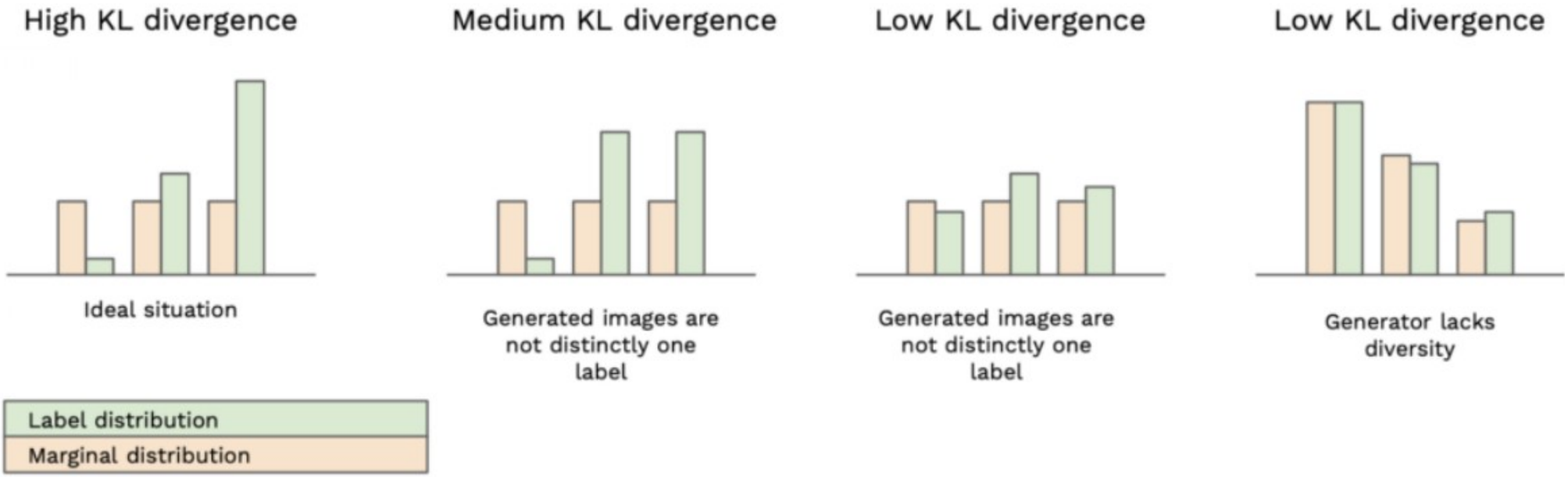
$c(y) = E_{x\sim p}[c(y|x)]$

The Inception Score (IS)



source

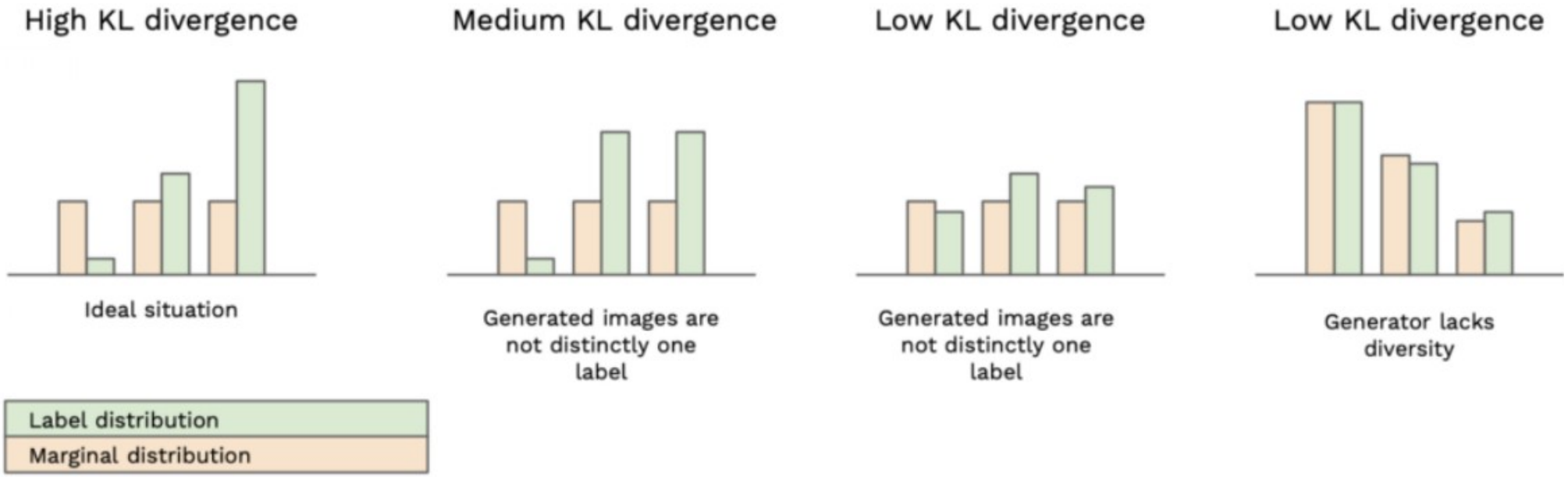
The Inception Score (IS)



source

$$\mathrm{IS}(G) = \exp\left(\mathop{\mathbb{E}}_{\mathbf{x}\sim p_g}\left[D_{\mathrm{KL}}(p(y|\mathbf{x}) \parallel p(y))\right]\right)$$

The Inception Score (IS)



Higher IS, better!

source

$$\mathrm{IS}(G) = \exp\left(\mathop{\mathbb{E}}_{\mathbf{x}\sim p_g}\left[D_{\mathrm{KL}}(p(y|\mathbf{x}) \parallel p(y))\right]\right)$$

Issues with IS

Score depends on weight initialization

	Network		
	IV2 TF	IV3 Torch	IV3 Keras
CIFAR-10	11.237±0.11	9.737±0.148	10.852±0.181
ImageNet Validation	63.028±8.311	63.702±7.869	65.938±8.616
Top-1 Accuracy	0.756	0.772	0.777

source

Issues with IS

Score depends on weight initialization

	Network		
	IV2 TF	IV3 Torch	IV3 Keras
CIFAR-10	11.237±0.11	9.737±0.148	10.852±0.181
ImageNet Validation	63.028±8.311	63.702±7.869	65.938±8.616
Top-1 Accuracy	0.756	0.772	0.777

source

Misleading results if Inception trained on another dataset than ImageNet

Fails to capture mode collapse inside a class

source

Overfitting may cause a good IS score.

Frehet Inception Distance

- authors consider that the coding units (of Inception v3) follow a multidimensional Gaussian

Frehet Inception Distance

- authors consider that the coding units (of Inception v3) follow a multidimensional Gaussian

- the difference for synthetic and real-world images is measured by the Fréchet distance (aka Wasserstein-2 distance)

Frehet Inception Distance

- authors consider that the coding units (of Inception v3) follow a multidimensional Gaussian

- the difference for synthetic and real-world images is measured by the Fréchet distance (aka Wasserstein-2 distance)

$$\mathrm{FID} = ||\mu_r - \mu_g||^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

- $X_r \sim \mathcal{N}(\mu_r, \Sigma_r)$ and $X_g \sim \mathcal{N}(\mu_g, \Sigma_g)$ are the 2048-dimensional activations of the Inception-v3 pool3 layer for real and generated samples respectively

Frehet Inception Distance

Numpy code:

```python
# calculate frechet inception distance
def calculate_fid(act1, act2):
    # calculate mean and covariance statistics
    mu1, sigma1 = act1.mean(axis=0), cov(act1, rowvar=False)
    mu2, sigma2 = act2.mean(axis=0), cov(act2, rowvar=False)
    # calculate sum squared difference between means
    ssdiff = numpy.sum((mu1 - mu2)**2.0)
    # calculate sqrt of product between cov
    covmean = sqrtm(sigma1.dot(sigma2))
    # check and correct imaginary numbers from sqrt
    if iscomplexobj(covmean):
        covmean = covmean.real
    # calculate score
    fid = ssdiff + trace(sigma1 + sigma2 - 2.0 * covmean)
    return fid
```

source

Lower FID, better!

Frehet Inception Distance

Pros of FID

- sensitive to intraclass sample omissions

- captures better the perception

- FID scores correlate with better-quality images when systematic distortions were applied

Frehet Inception Distance

Pros of FID

- sensitive to intraclass sample omissions

- captures better the perception

- FID scores correlate with better-quality images when systematic distortions were applied

Cons of FID

- depends on ImageNet training

- changes with the training dataset

Kernel Inception Distance

Maximum Mean Discrepancy

- compare samples from 2 distributions ***p*** and ***q*** by computing differences in their moments (mean, variance, etc.)

Kernel Inception Distance

Maximum Mean Discrepancy

- compare samples from 2 distributions ***p*** and ***q*** by computing differences in their moments (mean, variance, etc.)

- use a suitable kernel (Gaussian) to measure similarity between points

$$MMD(p,q) = E_{\mathbf{x},\mathbf{x}'\sim p}[K(\mathbf{x},\mathbf{x}')] + E_{\mathbf{x},\mathbf{x}'\sim q}[K(\mathbf{x},\mathbf{x}')] - 2E_{\mathbf{x}\sim p,\mathbf{x}'\sim q}[K(\mathbf{x},\mathbf{x}')]$$

- we compare the similarity between samples from ***p*** and ***q*** individually to the samples from the mixture of ***p*** and ***q***

How to evaluate Generative Models

HYPE (Human Eye Perceptual Evaluation)

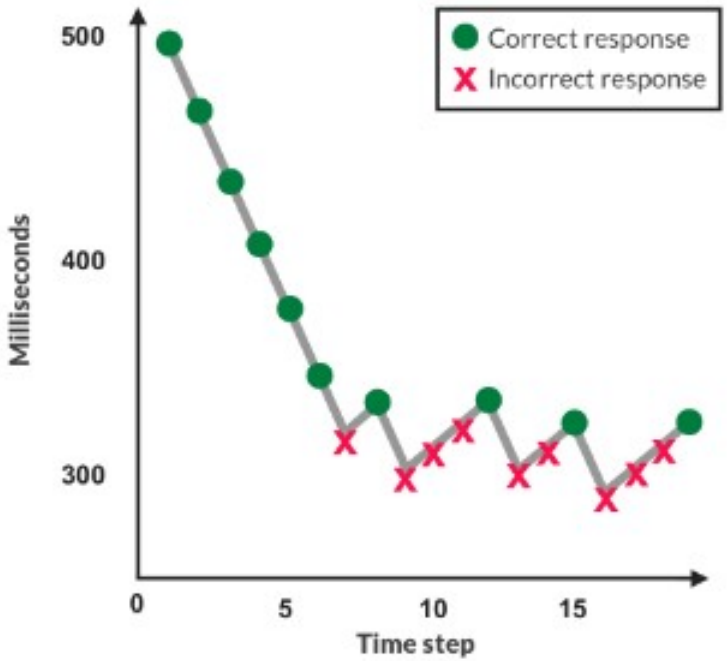


Figure 3: The adaptive staircase method shows images to evaluators at different time exposures, decreasing when correct and increasing when incorrect. The modal exposure measures their perceptual threshold.

source

We can (try to) approximate HYPE

How ?

Using embeddings of an Inception v3 pre-trained on ImageNet

$$d_{uv}^{\text{test}} = \text{NearestNeighborDistance}(z_{uv}^{\text{test}}, \{z_{uv}^{(1)}, \cdots, z_{uv}^{(n)}\}) = \min_{i=1,\cdots,n} \|z_{uv}^{\text{test}} - z_{uv}^{(i)}\|_2$$

source

A FID but with embeddings of more layers...

Overview

1) What are Deep Generative Models (DGM)

2) What are they good for

3) Some principles behind DGM

4) How to evaluate DGM

5) Challenges

Challenges

Inductive Bias versus Data

modeling spectrum

Generative models
(e.g. Bayesian networks)
+ lots of inductive bias
+ out of domain generalization

Inductive bias + data → predictions

Discriminative Models
(e.g. Deep learning)
+ lots of data
+ limited domain

PTSGM@ECCV2020

Challenges

Generalization

Humans generalize from a single example?

Training example (“water bear”)

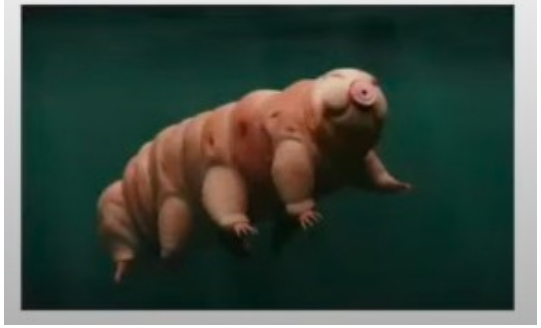
Challenges

Generalization

Humans generalize from a single example?

Training example ("water bear")

Test examples



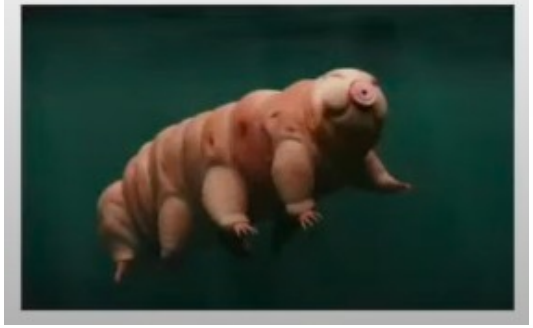
Challenges

Generalization

Humans generalize from a single example?

Training example ("water bear")

Test examples



Humans have much more background knowledge:

- Laws of physics, causality, biology, psychology, sociology, …

Challenges

Causality

Causal mechanisms generalize much better, and add understanding

Without an understanding of cause and effect (e.g. only correlations) you can not make optimal decisions.

Judea Pearl
Turning award 2012

"This ingredient [causality] *should allow computer systems to choreograph a* ***parsimonious*** *and* ***modular*** *representation of their environment,* ***interrogate*** *that representation,* ***distort*** *it through acts of imagination, and finally answer* ***"What if?"*** *kinds of questions."*

Challenges

Curse of dimensionality

- many signals are high-dimensional and representing the complete density of class is data-hard

- how to model the corner cases / boundaries ?

Thank you!