Generative Model

PHYS591000 Spring 2021

The slide is based on:

- MIT 6.S191[slide][video]
- Lil'Log: VAE, GAN

Supervised vs Unsupervised Learning

Supervised Learning

Goal: Learn function to map
x (data) to y (label)

Examples:

- Classification
- Regression
- Object Detection
- Semantic segmentation
- ...

Unsupervised Learning

Goal: Learn the hidden or underlying structure of the data without labels.

Examples:

- Clustering
- Feature extraction
- Dimensionality reduction
- ...

Generative Models

1701.00160

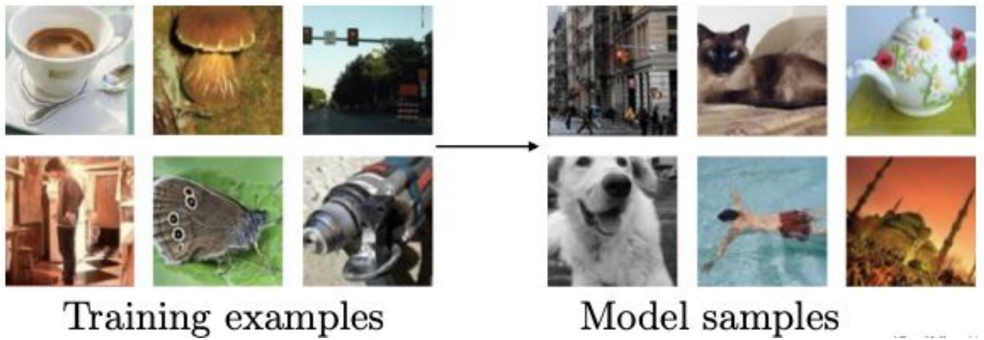
Goal: Take as input training samples from some distribution and learn a model that represent that distribution

How can we learn $P_{model}(x)$ similar to $P_{data}(x)$?

Density Estimation

$P_{data}(x)$ $P_{model}(x)$

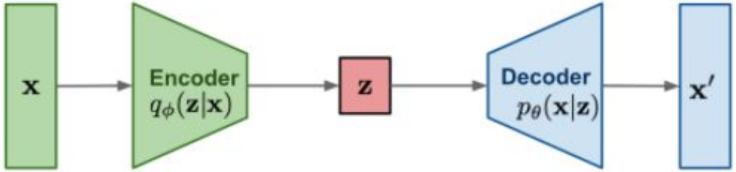
Sample Generation



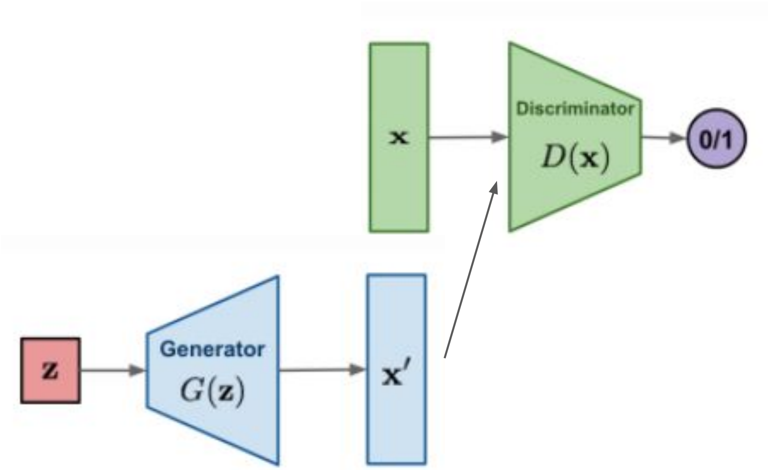
$P_{data}(x)$ $P_{model}(x)$

Generative Models

Autoencoders (AE) and
Variational Autoeconders (VAEs)



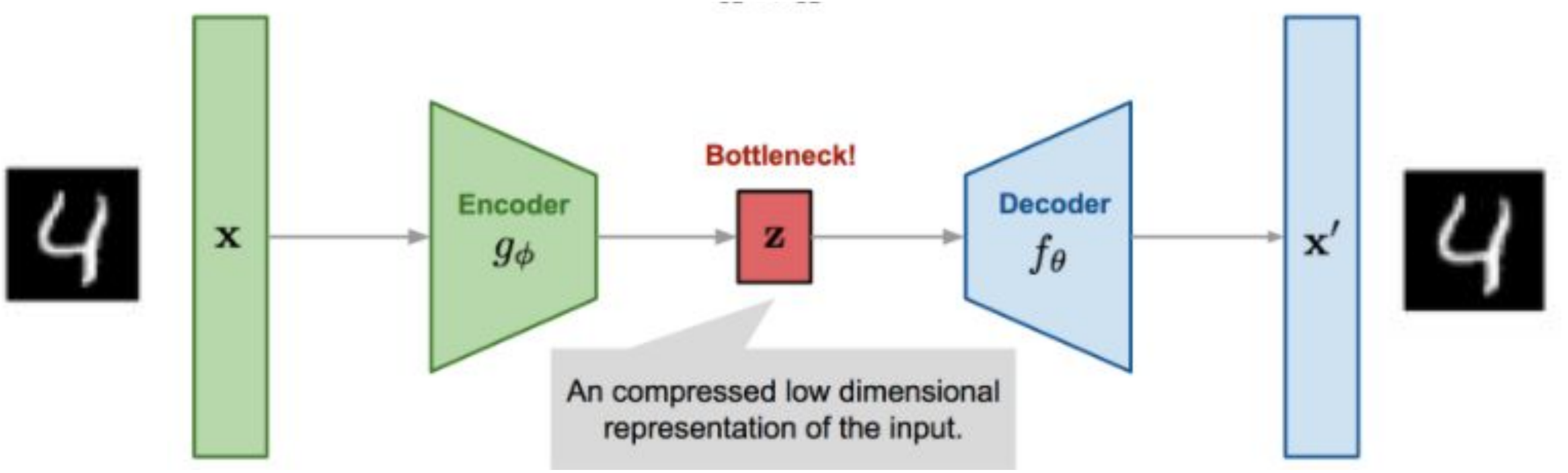
Generative Adversarial Network (GAN)



Autoencoders and Variational Autoencoders

Autoencoder

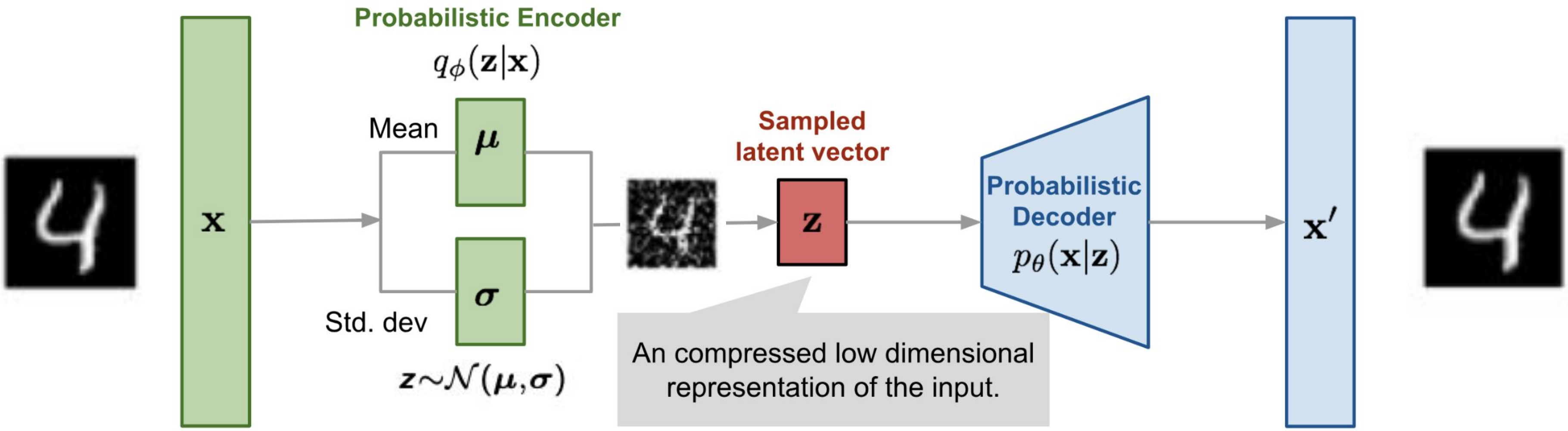
Bottleneck hidden layer: forces network to learn a compressed latent representation



$$\boldsymbol{x} \xrightarrow{\text{encoder}} \boldsymbol{z} \xrightarrow{\text{decoder}} \boldsymbol{x}' \qquad \mathcal{L}_{AE} = (\boldsymbol{x} - \boldsymbol{x}')^2$$

Variational Autoencoders (VAEs)

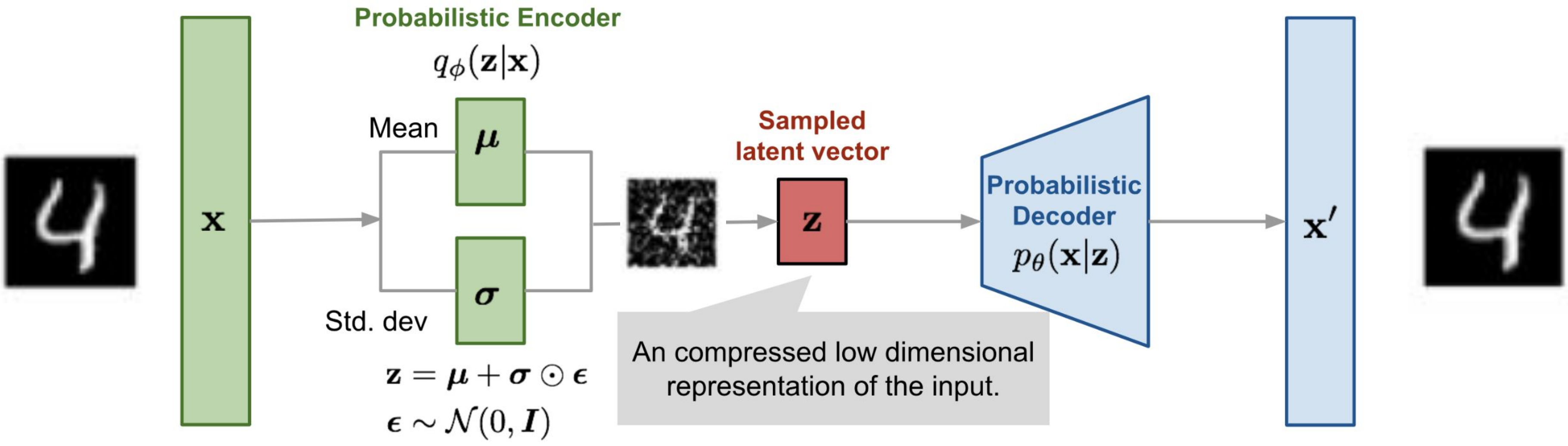
VAEs are a probabilistic variation on autoencoders?
Sample from the mean and standard deviation to compute latent sample



$$\text{VAE: } \boldsymbol{x} \xrightarrow{\text{encoder}} \begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\sigma} \end{pmatrix} \xrightarrow[z \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})]{\text{sample}} \boldsymbol{z} \xrightarrow{\text{decoder}} \boldsymbol{x}' \qquad \mathcal{L}_{VAE} = \mathcal{L}_{AE} + \mathcal{L}_{lat}$$

β-Variational Autoencoders

Parametrization tricks: A fixed μ vector and a fixed σ vector scaled by random constants ε drawn from the prior distribution (usually Normal Gaussian)



$$\beta\text{- VAE: } \boldsymbol{x} \xrightarrow{\text{encoder}} \begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\sigma} \end{pmatrix} \xrightarrow[z = \mu + \sigma \odot \varepsilon]{\text{sample}} \boldsymbol{z} \xrightarrow{\text{decoder}} \boldsymbol{x}' \qquad \mathcal{L}_{VAE} = \mathcal{L}_{AE} + \mathcal{L}_{lat}$$

β-VAE

Reconstruction Loss   Regularization term

$$\beta\text{-VAE: } \boldsymbol{x} \xrightarrow{\text{encoder}} \begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\sigma} \end{pmatrix} \xrightarrow[z = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \varepsilon]{\text{sample}} \boldsymbol{z} \xrightarrow{\text{decoder}} \boldsymbol{x}' \qquad \mathcal{L}_{VAE} = \boxed{\mathcal{L}_{AE}} + \boxed{\mathcal{L}_{lat}}$$

Loss enforces Gaussian latent space

$$\begin{aligned} \mathcal{L}_{VAE} &= \mathcal{L}_{AE} + \beta \cdot \boxed{\mathrm{KL}(q_x(z) | \mathcal{N}(0,1))} \qquad \leftarrow \text{similarity measure} \\ &= \mathcal{L}_{AE} + \frac{\beta}{2} \sum_j 1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2 \end{aligned}$$

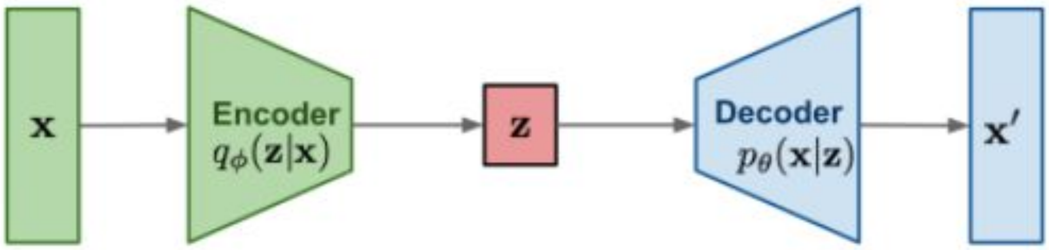
Kullback-Leibler (KL) divergence

- KL is a measure of how one probability distribution is different from a second
- Normal Gaussian encourages encodings to distribute encodings evenly around the center of the latent space

VAE summary

MIT 6.S191

- Compress representation to something we can use to learn
- Reconstruction allows for unsupervised learning (no labels!)
- Reparameterization trick enables training
- Interpret hidden latent variables using perturbation
- Generating new examples

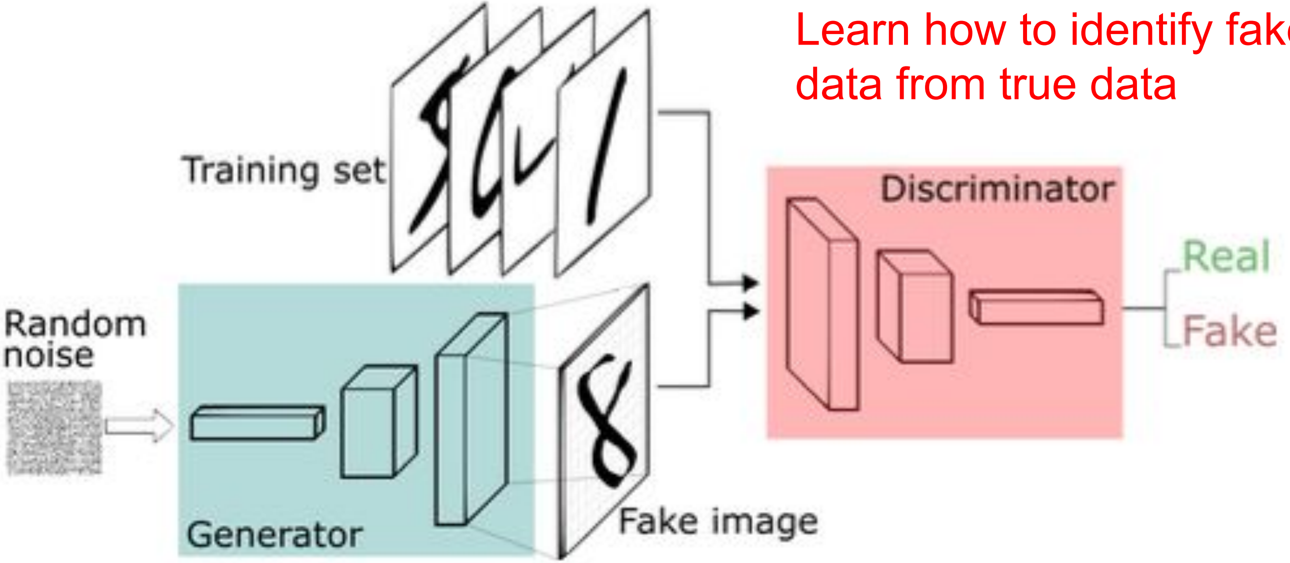


Generative Adversarial Network

Generative Adversarial Networks (GAN)

To make a generative model by having two neural networks compete with each other

Learn how to identify fake data from true data

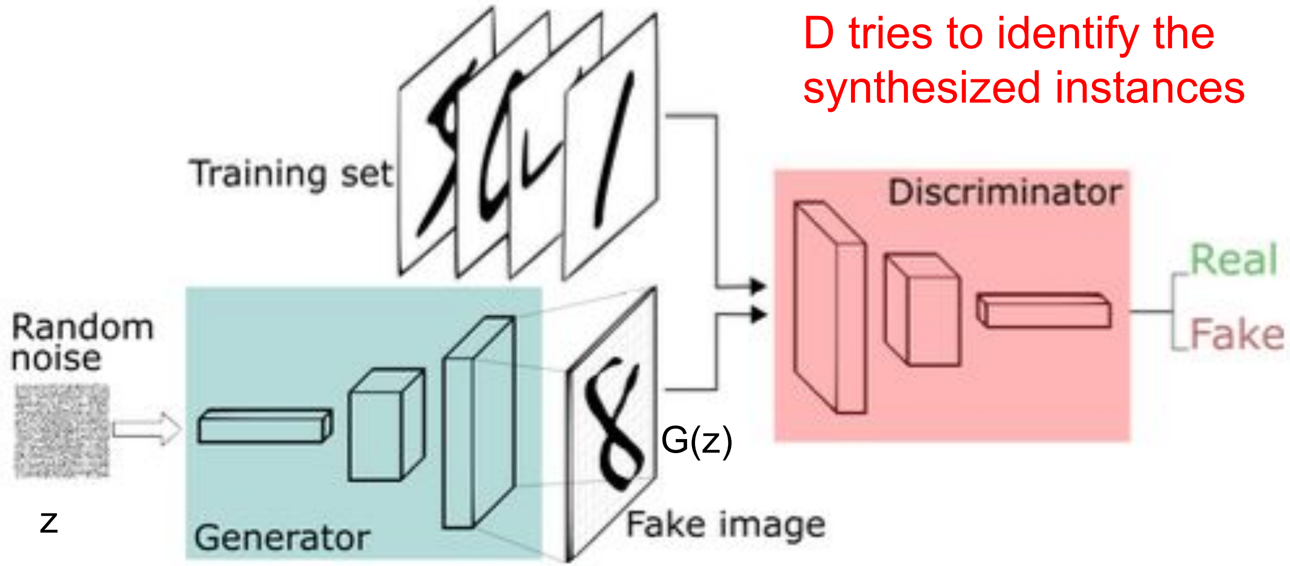


Learn data distribution

Generative Adversarial Networks

Training: adversarial objectives for D and G
Goal: G reproduces the true data distribution

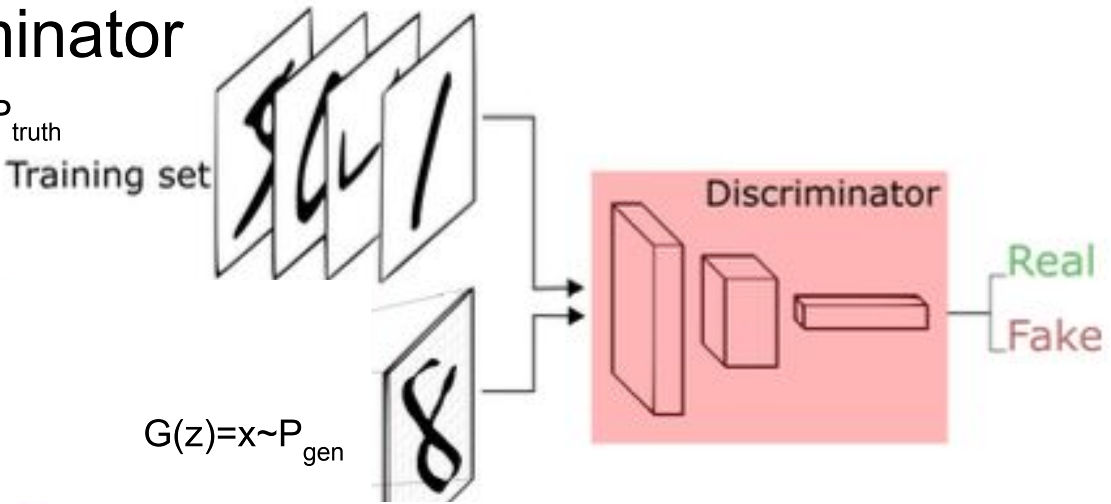
D tries to identify the synthesized instances



G tries to synthesize fake instances that fool D

Training Discriminator

$x \sim P_{truth}$



$G(z) = x \sim P_{gen}$

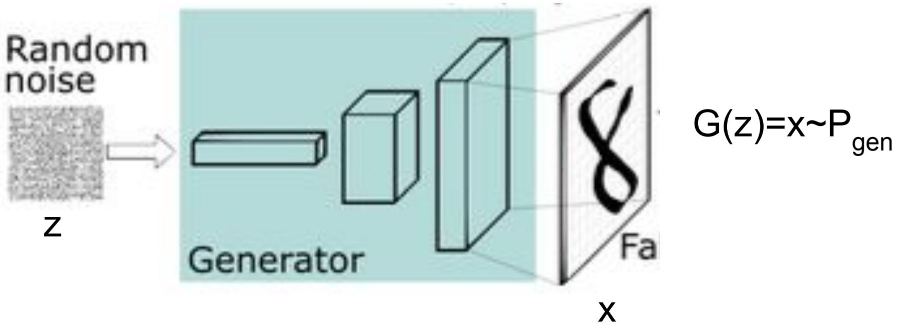
Discriminator

$$\underset{D}{\arg\max}\; L_D = \langle \log D(x) \rangle_{x \sim P_{Truth}} + \langle \log(1 - D(x)) \rangle_{x \sim P_{Gen}}$$

D’s decisions over real data are accurate

D’s output $< D(G(z)) > = < D(x)_{x \sim Pgen} >$ close to zero over fake data

Training Generator

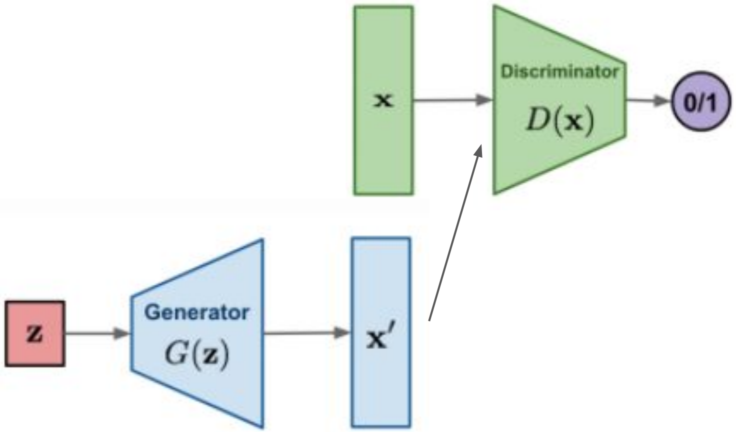


Generator

$$\underset{G}{\arg\min}\ L_G = \left\langle \log(1 - D(x)) \right\rangle_{x \sim P_{Gen}}$$

G is trained to increase the chance of D producing a high probability for a fake example.

Training: min-max game

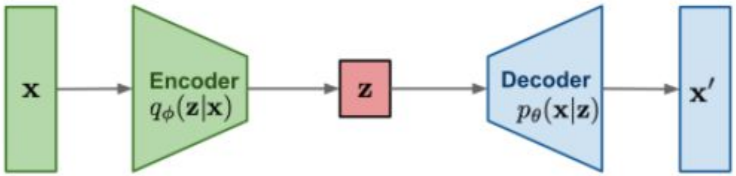


$$\arg\min_{G}\max_{D}\; L_D = \langle \quad \log D(x) \rangle_{x \sim P_{Truth}} + \langle \quad \log(1 - D(x)) \rangle_{x \sim P_{Gen}}$$

Generative Models

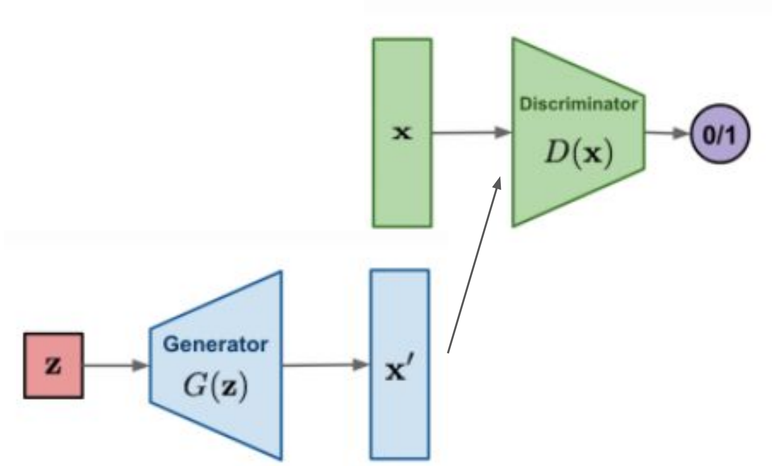
Autoencoders and Variational Autoencoders (VAEs)

Learn lower-dimensional **latent space** and **sample** to generate input reconstructions



Generative Adversarial Network (GAN)

Competing **generator** and **discriminator** networks

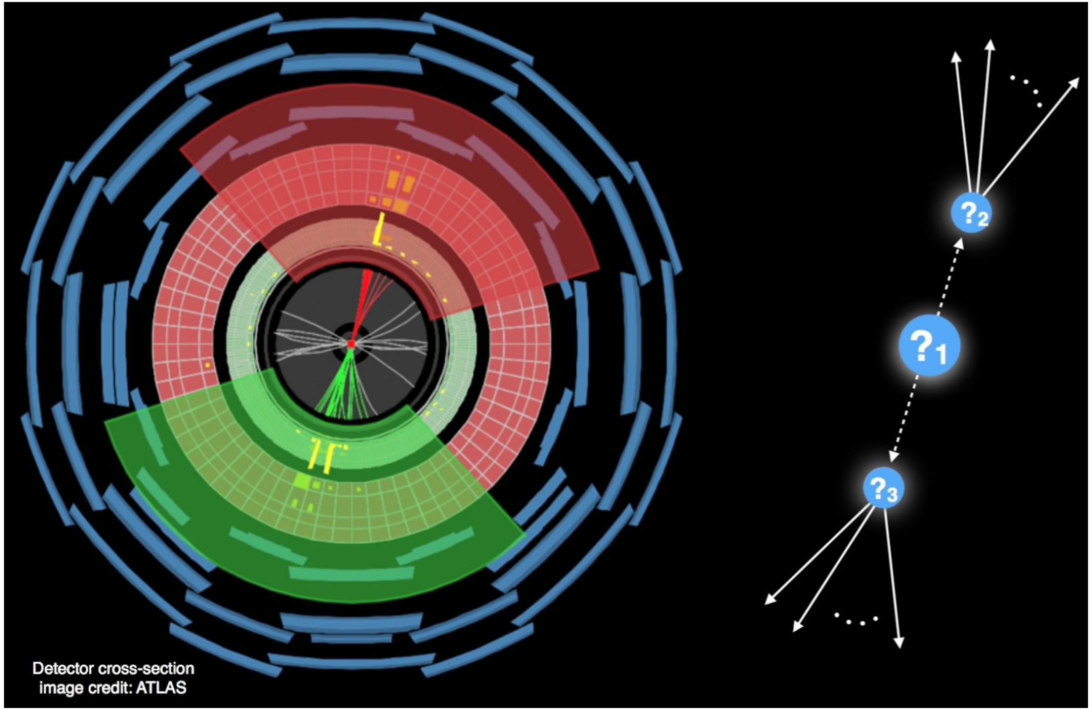


Which face is real?

https://www.whichfaceisreal.com/methods.html

Lab this week

Dijet Production



LHCO2020

pp → W' (3.5 TeV) → X +Y

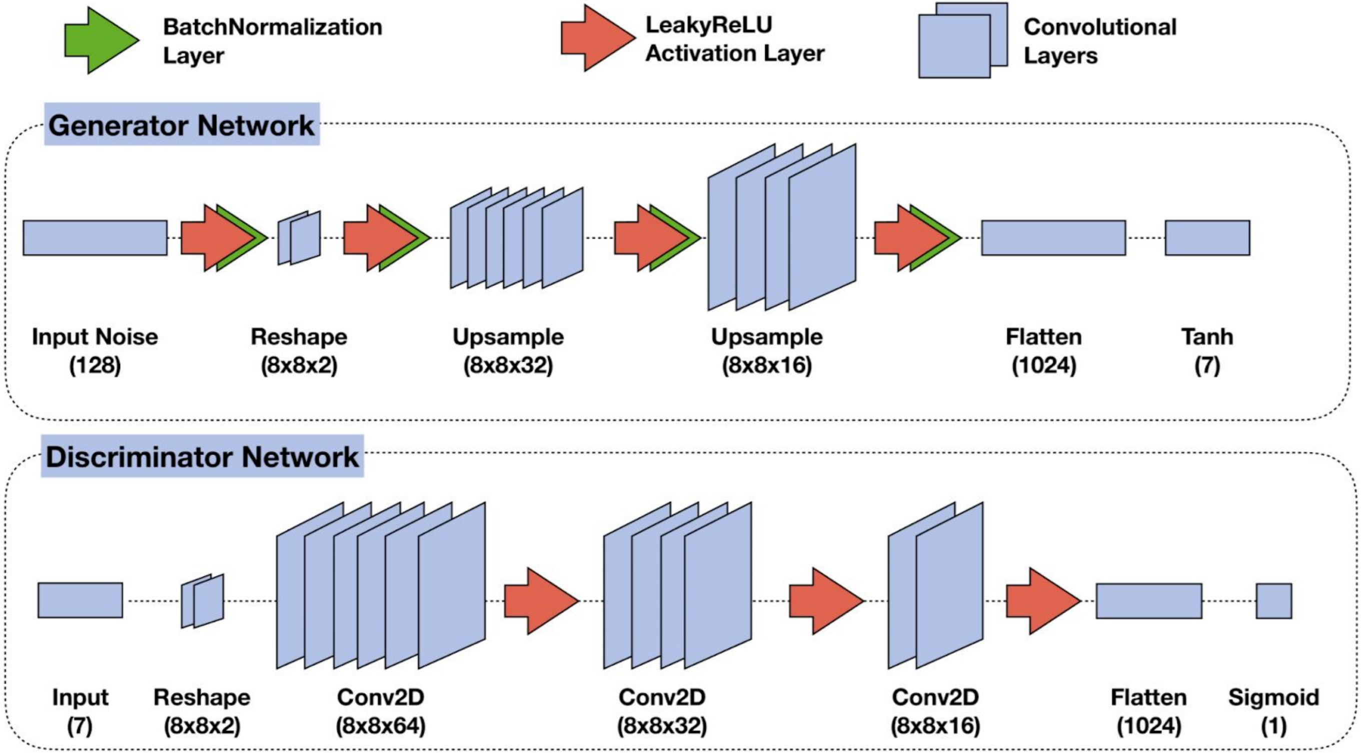
X→qq and Y→qq

mX= 500 GeV

mY=100 GeV

Anti-kT fat-jet R = 1, cuts of pT > 1.2 TeV and |η| < 2.5

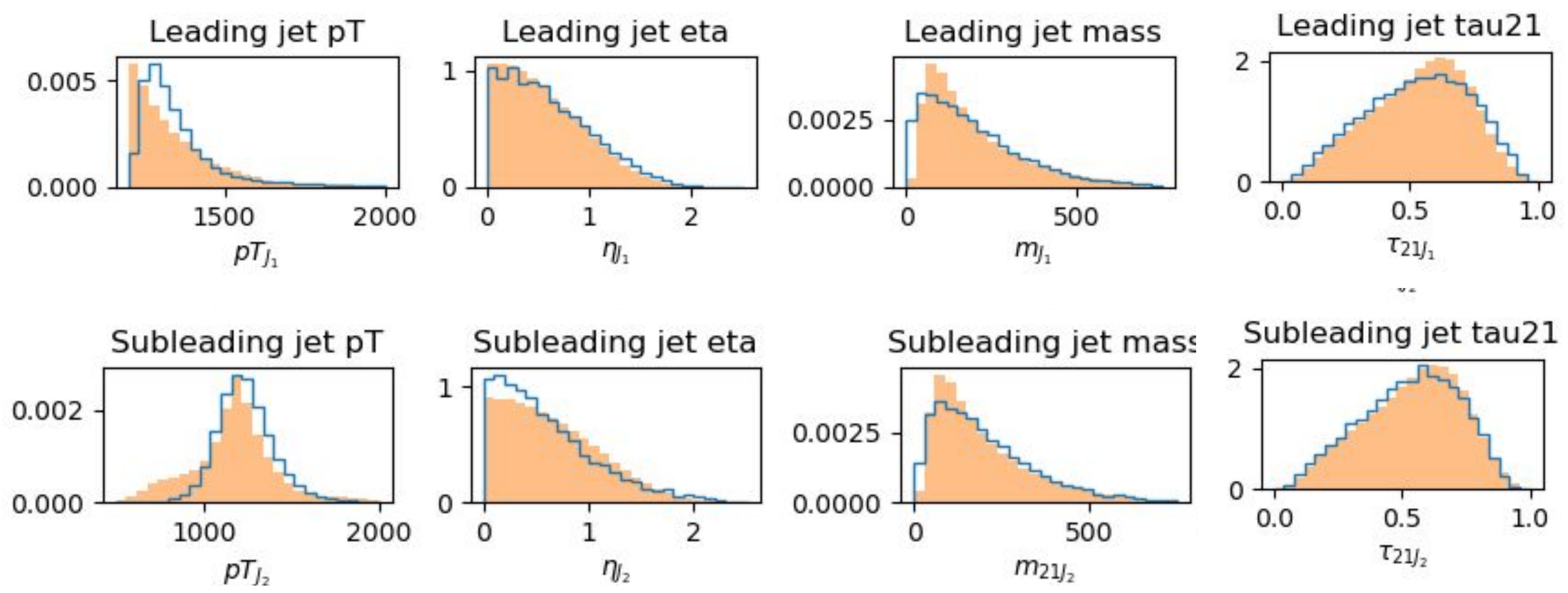
DijetGAN (code)



Source: arXiv:1903.02433

GAN results

Orange: input data   Blue: Generated Data



References

- Lil'Log: VAE, GAN, Flow-based
- A. Butta, Machine learning for particle physicists, Vietnam 2020
- Deep Generative Model for Fundamental Physics, BIDS March 2021
- CS 236 Fall 2019, Ermon & Grover, “Deep Generative Models”
- MIT 6.S191 A. Soleimany, “Introduction to Deep Learning” Lecture 4 [slide][video]