Introduction to Machine Learning and Artificial Intelligence: Lecture III

Michael Kagan

SLAC NATIONAL ACCELERATOR LABORATORY

2nd COFI Advanced Instrumentation and Analysis Techniques School

December 10, 2023

The Plan

- Lecture 1
  - Introduction to Machine Learning fundamentals
  - Linear Models

- Lecture 2
  - Neural Networks
  - Deep Neural Networks
  - Inductive Bias and Model Architectures

- Lecture 3
  - Unsupervised Learning
  - Autoencoders
  - Towards generative modeling: Variation Autoencoders

Beyond Regression and Classification

Beyond Regression and Classification

- Not all tasks are predicting a label from features, as in classification and regression

- May want to model a high-dim. signal
  - Data synthesis / simulation
  - Density estimation
  - Anomaly detection
  - Denoising, super resolution
  - Data compression
  - …

- Often don't have labels → Unsupervised Learning

Unsupervised Learning

- Our goal is to study the data density $p(x)$
- Even w/o labels, aim to characterize the distribution

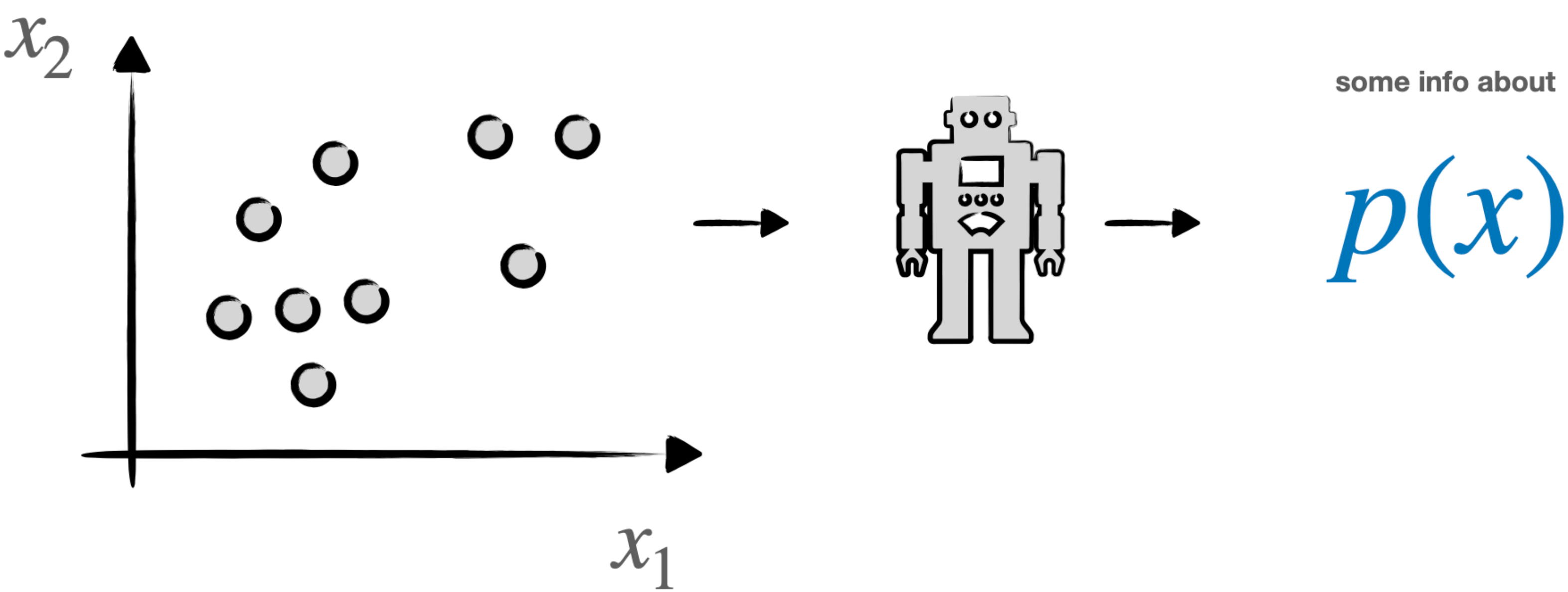


Image credit: L. Heinrich

Probability Models

A process

 $\rightarrow \mathbb{R}^2$

Generating new samples from randomness

A formula

$\mathbb{R}^2 \rightarrow \mathbb{R}$

$$p_{\mu,\Sigma}(x) = \frac{1}{\sqrt{|2\pi\Sigma|}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma(x-\mu)\right)$$

Evaluating the Probability for a given sample

"Understanding $p(x)$" – ability to do either or both of these

Image credit: L. Heinrich

Probability Models as Sampling a Process

- In many cases, we don't have a theory of the underlying process → *Can still learn to sample*

- Deep learning can be very good at this!

$$\text{face} \sim p(\text{face})$$

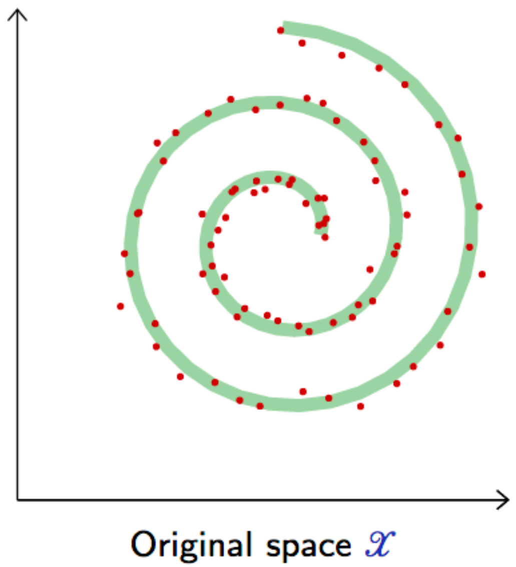


https://thispersondoesnotexist.com/

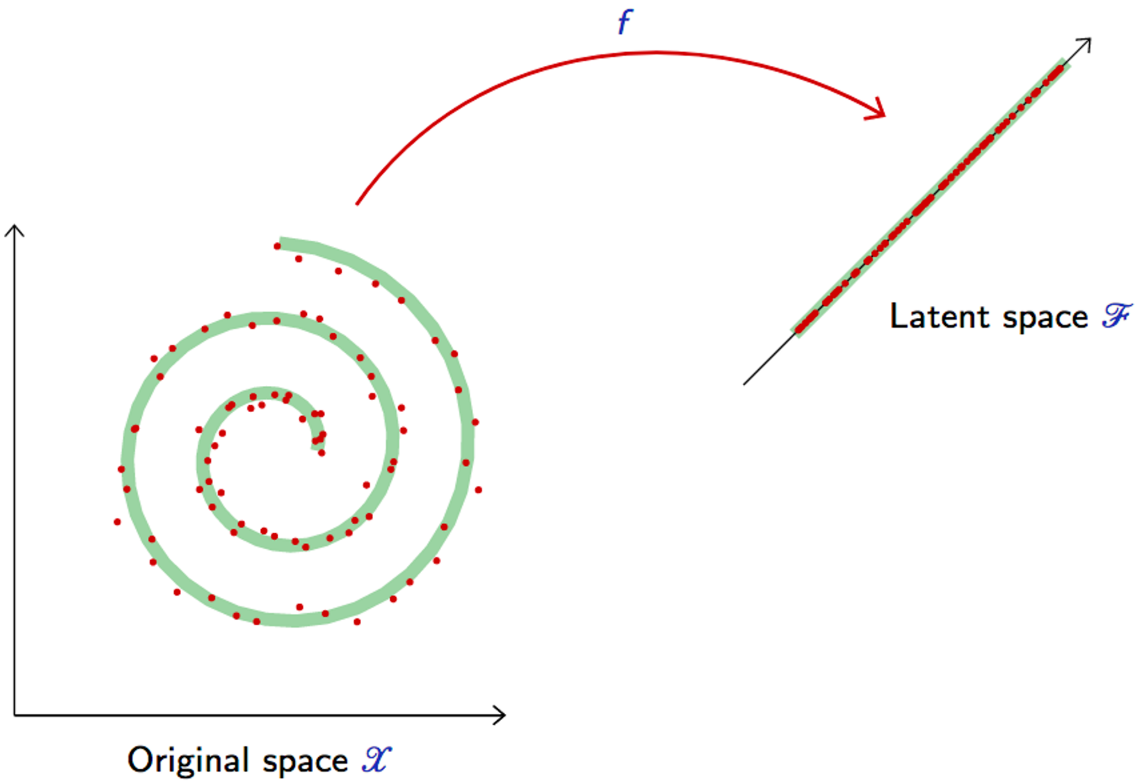
Learning Objective

- Unsupervised learning is more heterogeneous than supervised learning
- Many architectures, losses, learning strategies
- Often constructed so model converges to $p(x)$
  - Variational inference, Adversarial learning, Self-supervision, …
- Often framed as **modeling the lower dimensional "meaningful degrees of freedom"** that describe the data

Modeling Data and Meaningful Degrees of Freedom

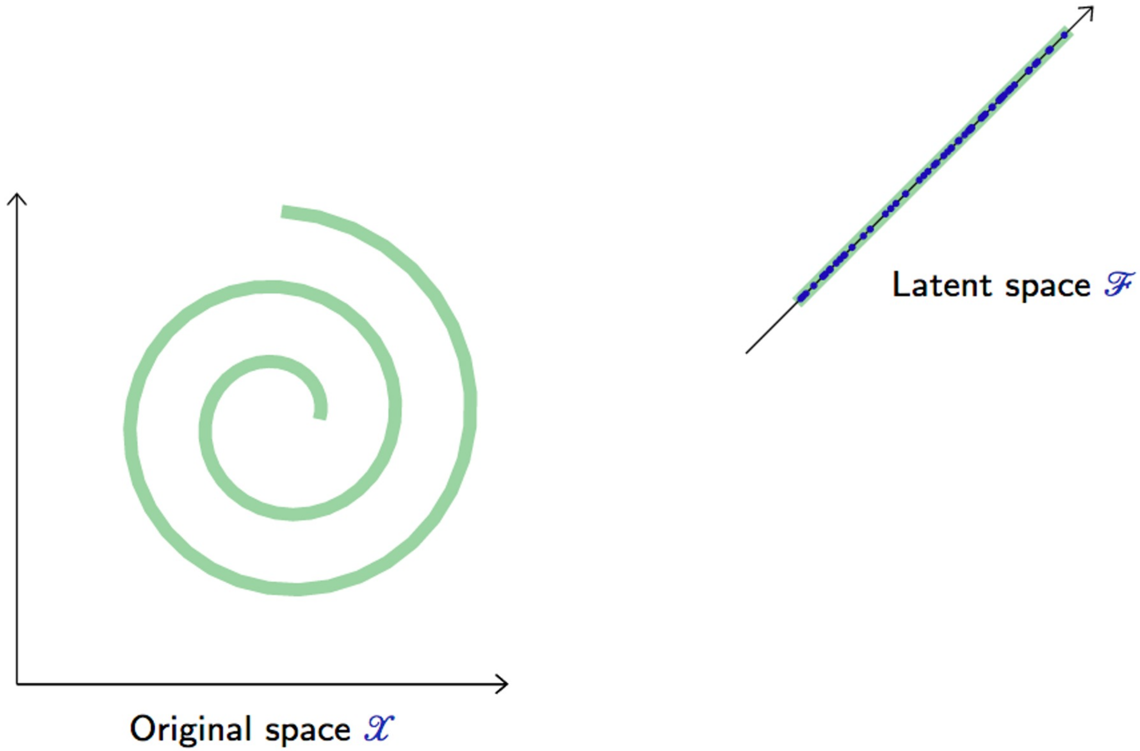


Modeling Data and Meaningful Degrees of Freedom



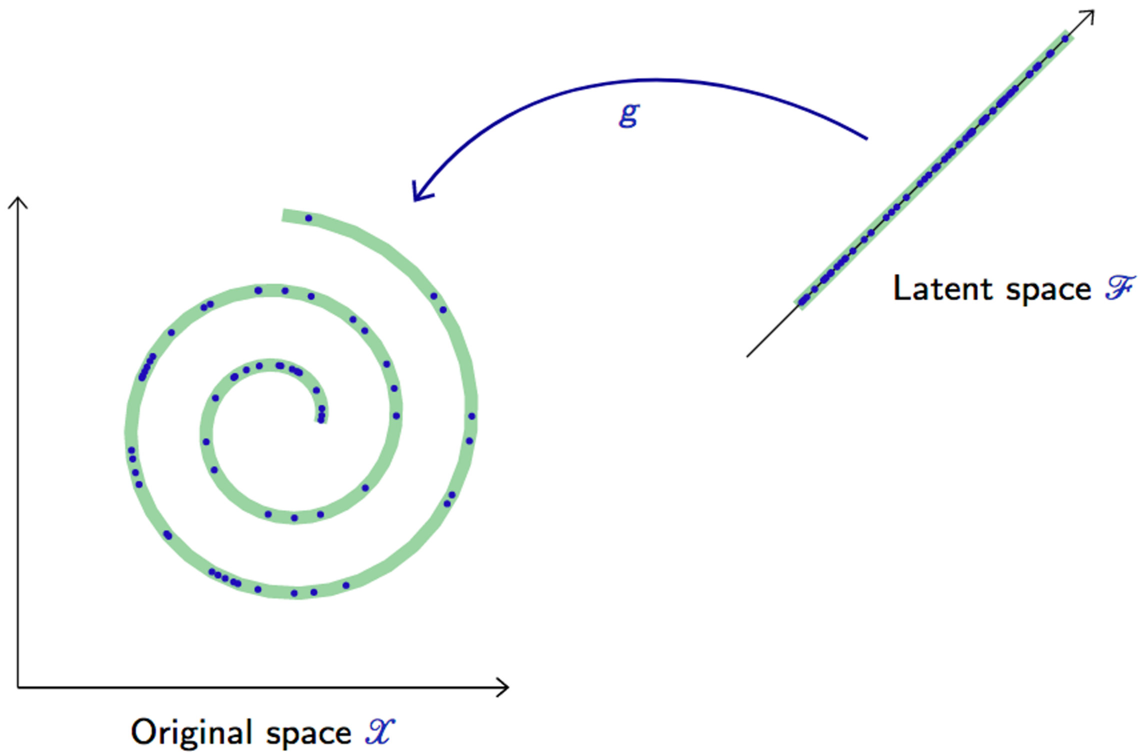
Fleuret, [Deep Learning Course]()

Modeling Data and Meaningful Degrees of Freedom



Fleuret, [Deep Learning Course]

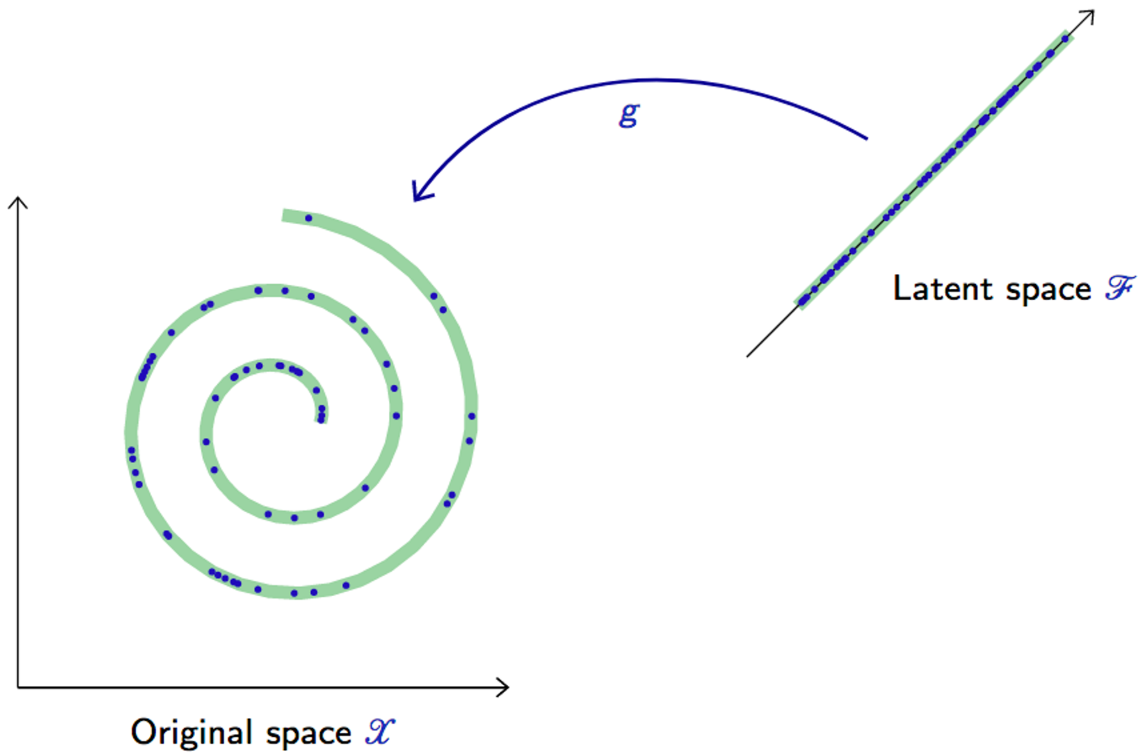
Modeling Data and Meaningful Degrees of Freedom



Fleuret, [Deep Learning Course]

Modeling Data and Meaningful Degrees of Freedom

How can we find the "meaningful degrees of freedom" in the data?



Fleuret, [Deep Learning Course]

Meaningful Representations

- Dimensionality Reduction / Compression

- Can we learn to:

    1. Compress the data to a *latent space* with smaller number of dimensions

    2. Recover the original data from this latent space?

- Latent space must encode and retain the important information about the data

Autoencoders

Autoencoders

- Map a space to itself through a compression

$$x \to z \to \hat{x}$$

Data

Latent space

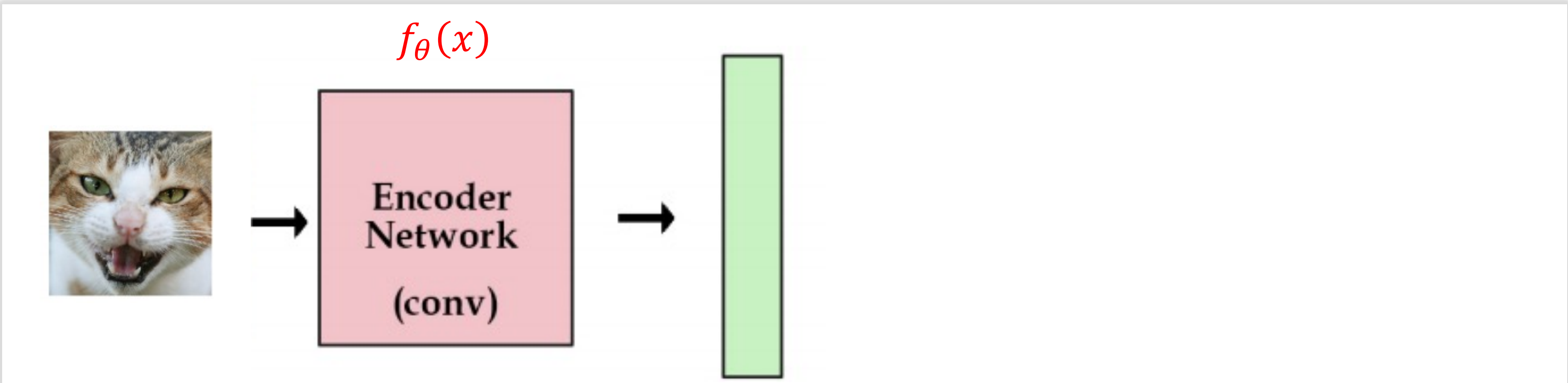
Reconstruction

Autoencoders

- Map a space to itself through a compression

$$x \to z \to \hat{x}$$

  - **Encoder**: Map from data to a lower dim. latent space
    - Neural network $f_\theta(x)$ with parameters θ

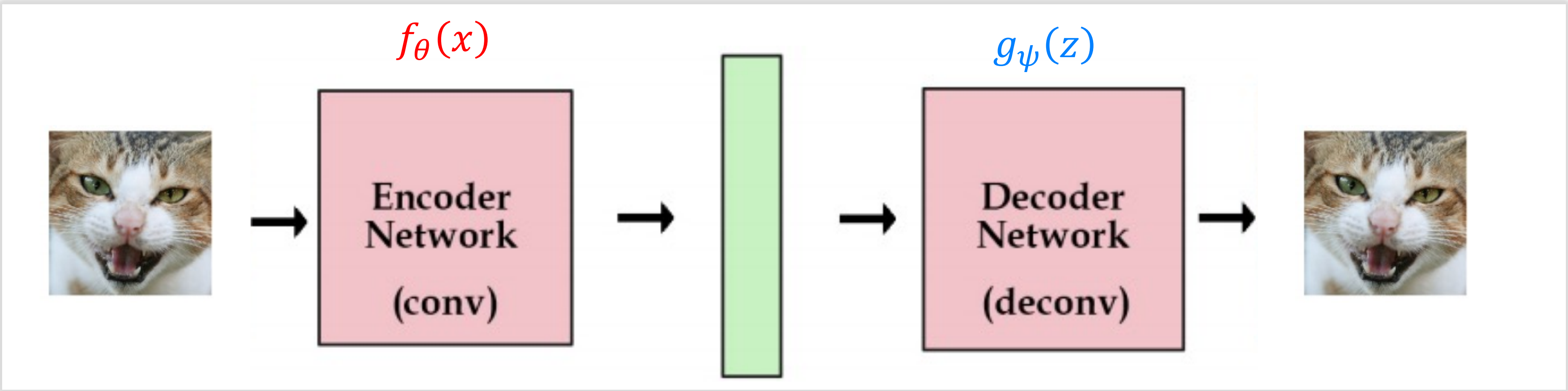


Autoencoders

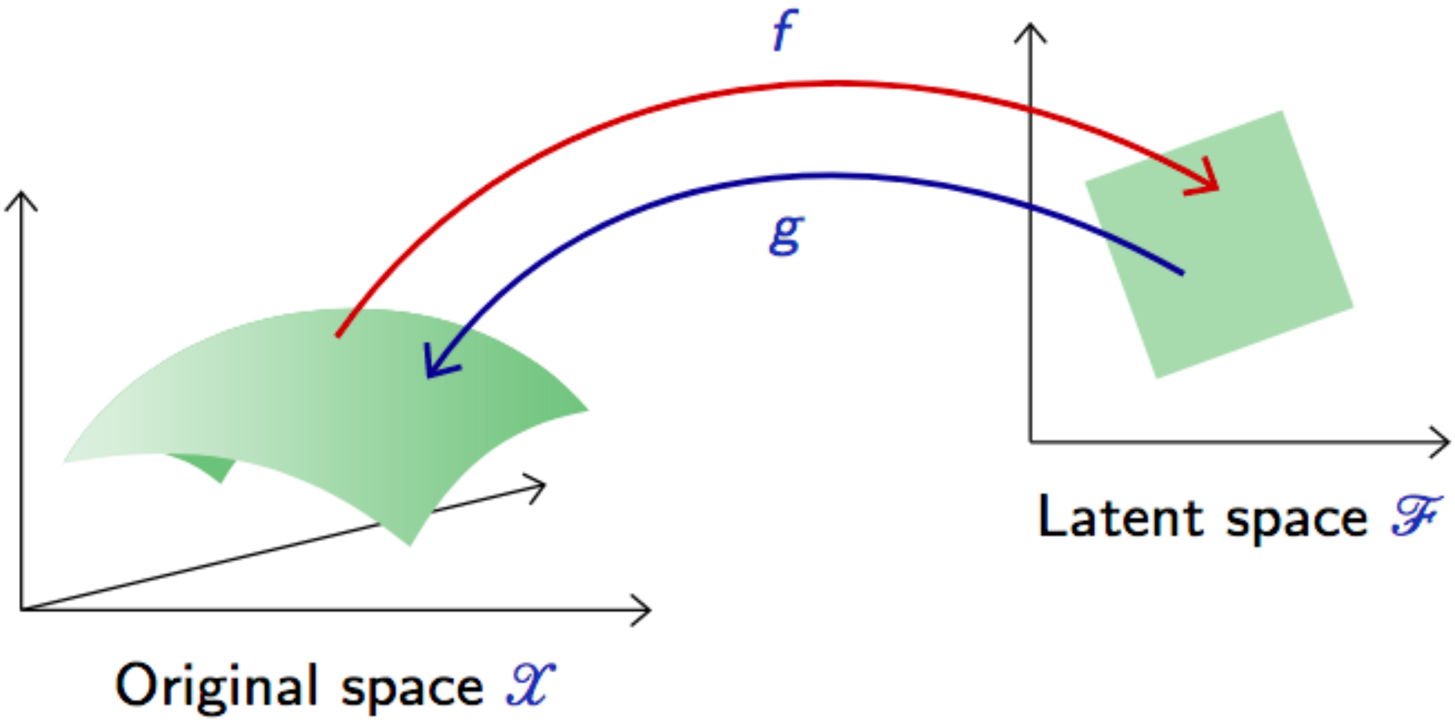
- Map a space to itself through a compression

$$x \to z \to \hat{x}$$

  - **Encoder**: Map from data to a lower dim. latent space
    - Neural network $f_\theta(x)$ with parameters θ
  - **Decoder**: Map from latent space back to data space
    - Neural network $g_\psi(z)$ with parameters ψ



Autoencoder Mappings



- Latent space is of lower dimension than data
- Model must learn a "good" parametrization and capture dependencies between components

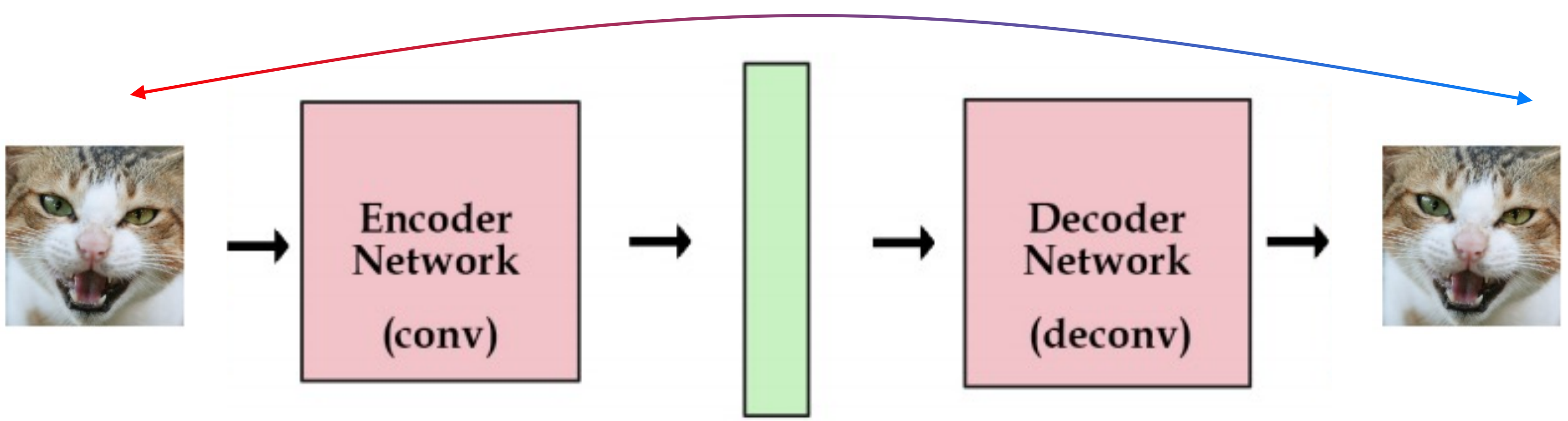
Fleuret, [Deep Learning Course]

Autoencoder Loss

$$L(\theta, \psi) = \frac{1}{N} \sum_n \left\| x_n - g_\psi(f_\theta(x_n)) \right\|^2$$

- **Loss**: mean *reconstruction loss* (MSE) between data and encoded-decoded data

- Min. over params. of encoder (θ) and decoder (ψ).



Autoencoder Loss

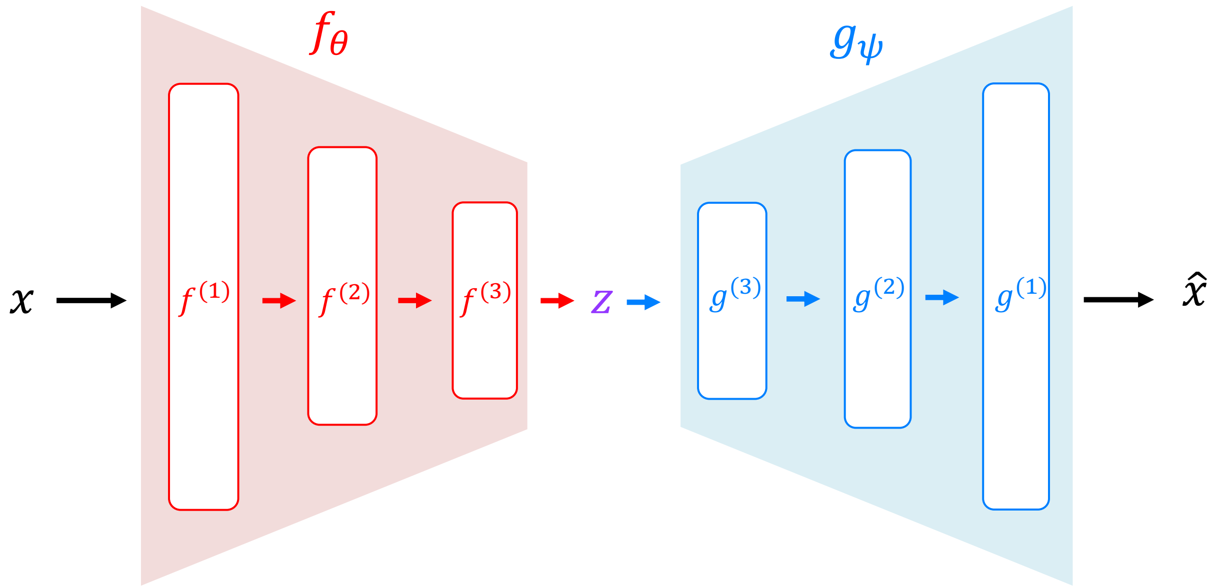
$$L(\theta, \psi) = \frac{1}{N} \sum_n \left\| x_n - g_\psi\left(f_\theta(x_n)\right) \right\|^2$$

- Loss: mean *reconstruction loss* (MSE) between data and encoded-decoded data

- Min. over params. of encoder (θ) and decoder (ψ).

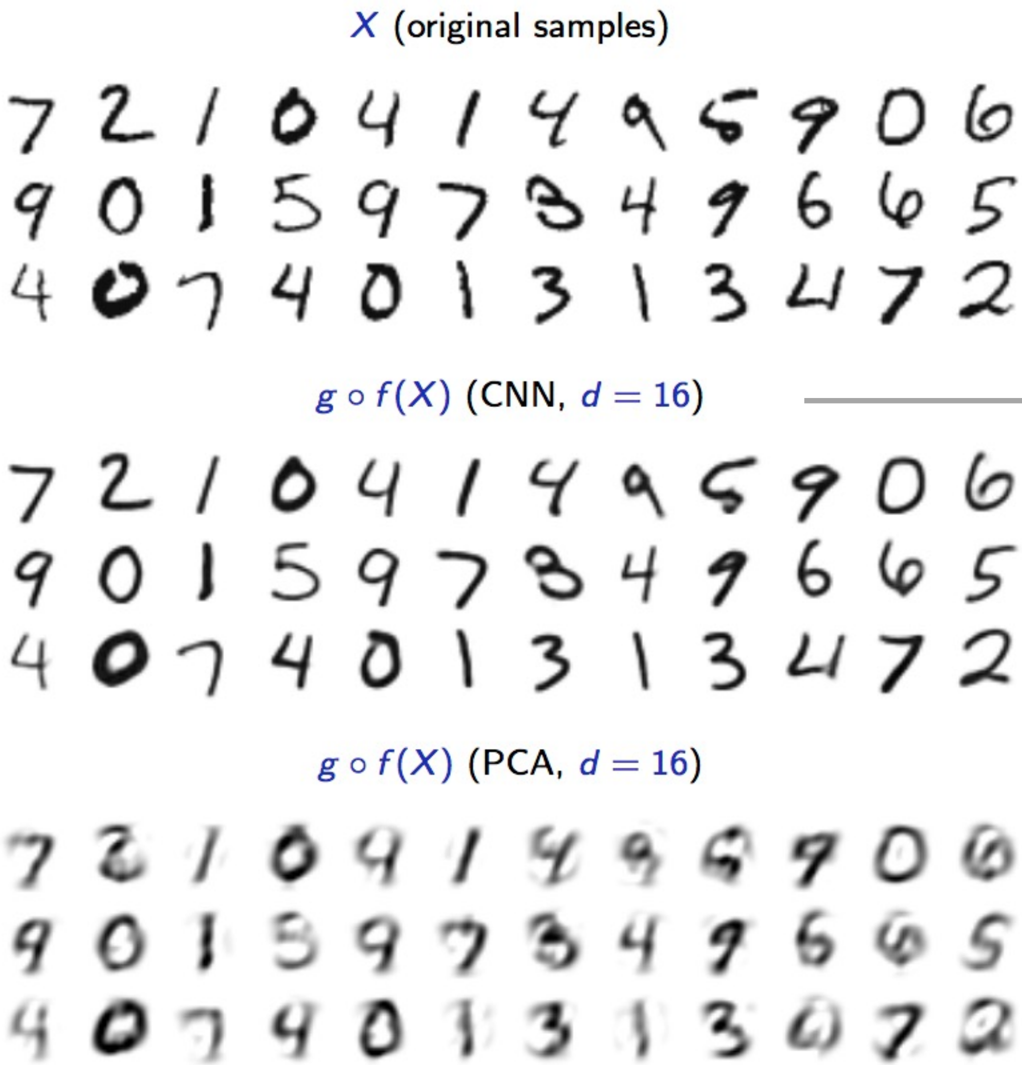
- NOTE: if $f_\theta(x)$ and $g_\psi(z)$ are linear, optimal solution given by Principle Components Analysis

Deep Autoencoder



- When f_θ and g_ψ are multiple neural network layers, can learn complex mappings
  - f_θ and g_ψ can be Fully Connected, CNNs, RNNs, etc.
  - Choice of network structure will depend on data

Deep Convolutional Autoencoder

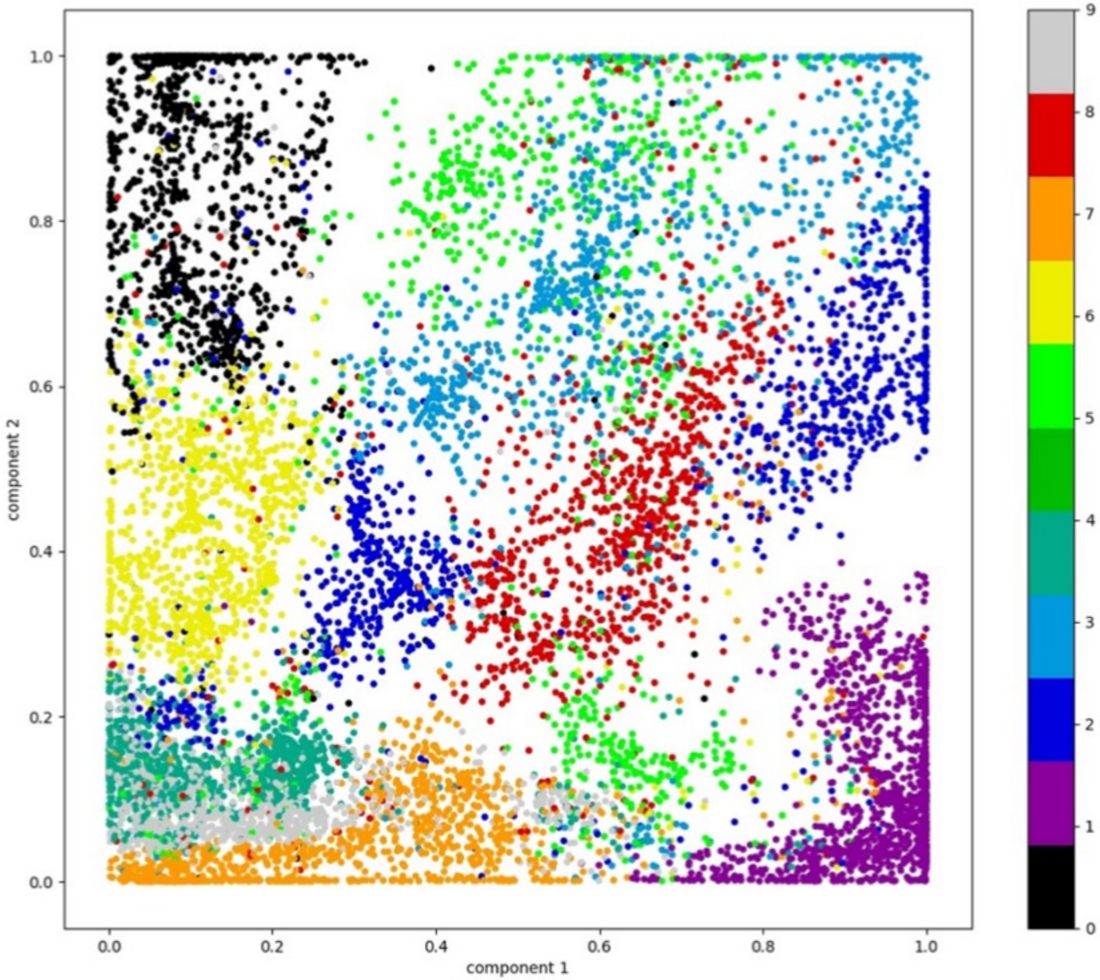


f_θ and g_ψ are five convolutional layers

Fleuret, [Deep Learning Course]

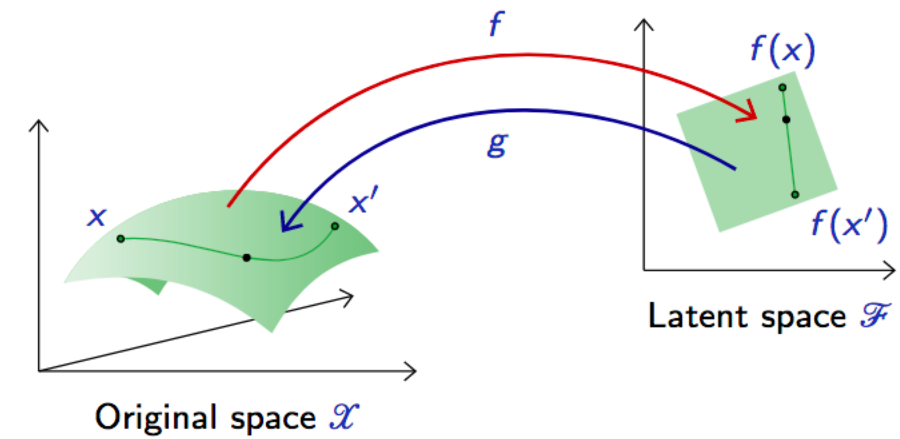
The Latent Space

- Can look at latent space to see how the model arranges the data



Interpolating in Latent Space

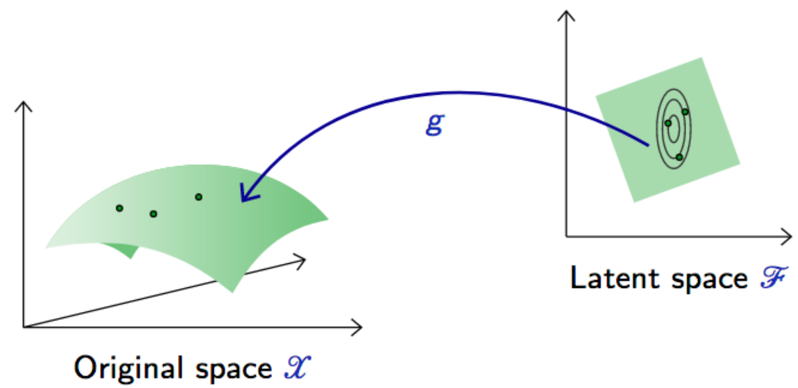
$$\alpha \in [0,1], \quad \xi(x, x', \alpha) = g((1-\alpha)f(x) + \alpha f(x')).$$



Autoencoder interpolation ($d = 8$)

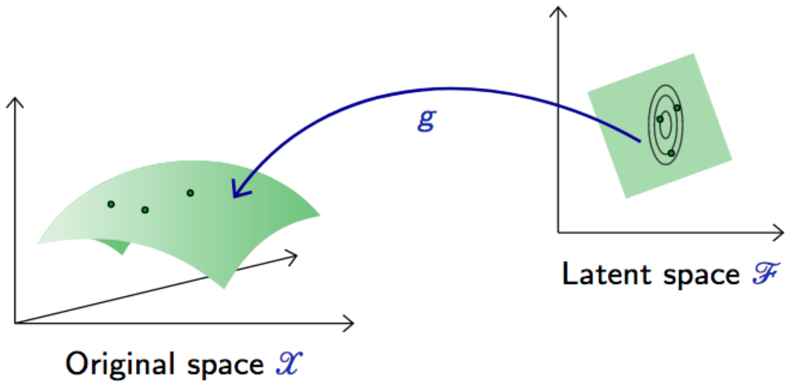
Can We Generate Data with Decoder?

- Can we sample in latent space and decode to generate data?

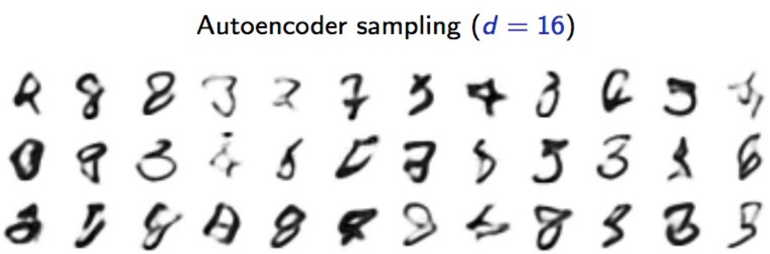


Can We Generate Data with Decoder?

- Can we sample in latent space and decode to generate data?

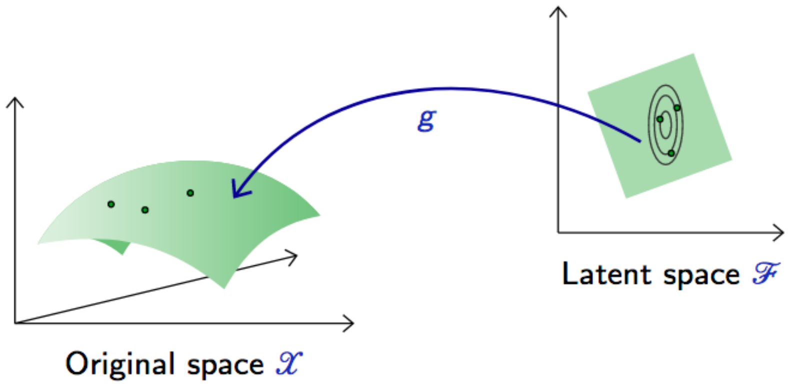


- What distribution to sample from in latent space?
  - Try Gaussian with mean and variance from data

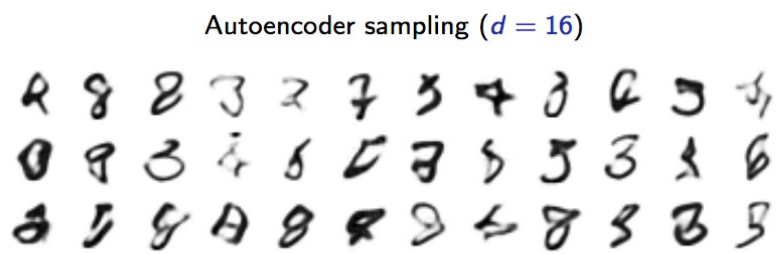


Can We Generate Data with Decoder?

- Can we sample in latent space and decode to generate data?



- What distribution to sample from in latent space?
  - Try Gaussian with mean and variance from data



- Don't know the right latent space density

Fleuret, Deep Learning Course

Generative Models

Generative Models Goal

A generative model is a probabilistic model q that can be used as a simulator of the data.

Goal: generate synthetic, realistic high-dimension data

$$x \sim q(x; \theta)$$

that is as close as possible to the unknown data distribution $p(x)$ for which we have empirical samples.

i.e. want to recreate the raw data distribution (such as the distribution of natural images).

Generative Models

- Generative models aim to:
  - Learn a distribution $p(x)$ that explains the data
  - Draw samples of plausible data points

- Explicit Models
  - Can evaluate the density $p(x)$ of a data point x

- Implicit Models
  - Can only sample $p(x)$, but not evaluate density

Variational Autoencoders

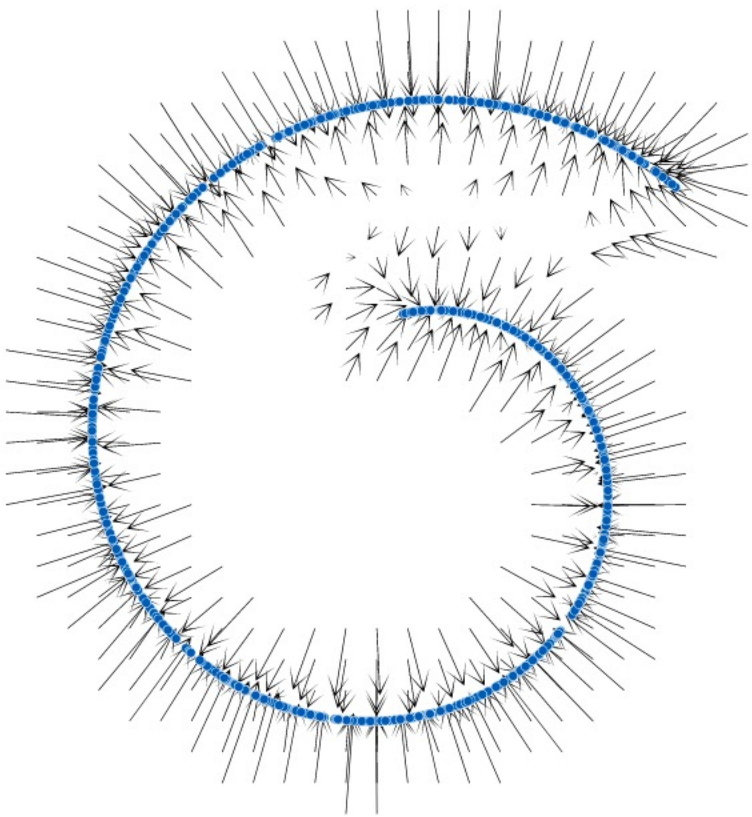
Denoising Autoencoders

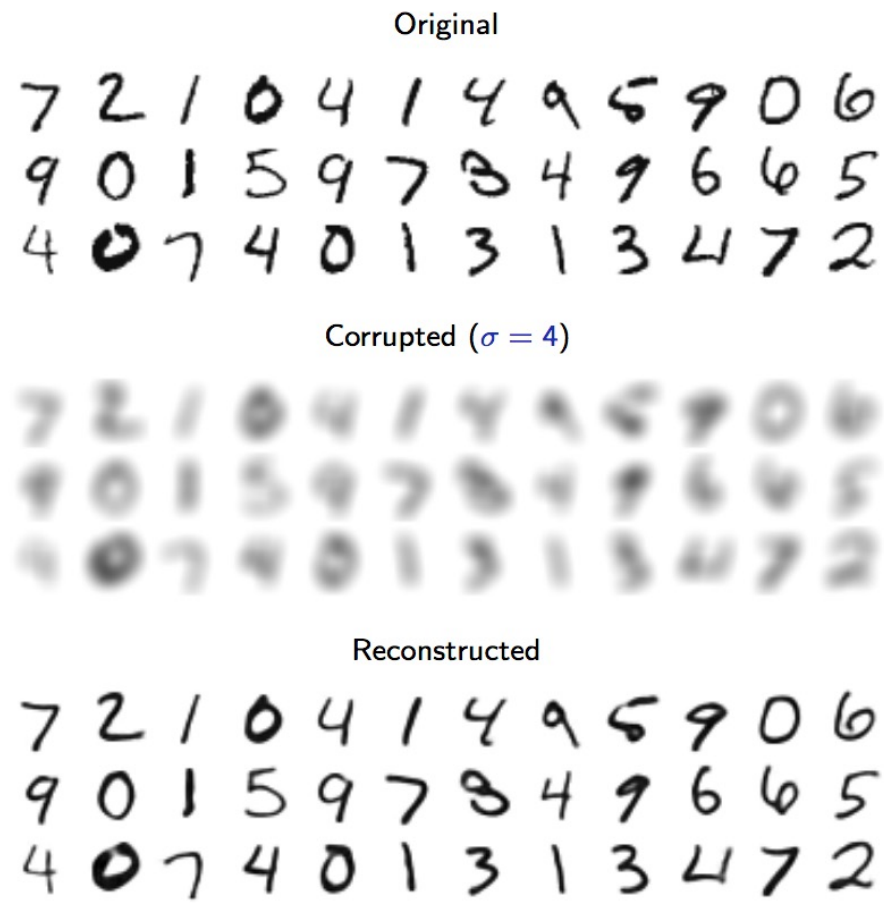
- Learn a mapping from corrupted data space $\tilde{X}$ back to original data space
  - Mapping $\phi_w(\tilde{X}) = X$
  - ϕ_w will be a neural network with parameters w
- Loss:

$$\mathrm{L} = \frac{1}{N}\sum_n \|x_n - \phi_w(x_n + \epsilon_n)\|$$

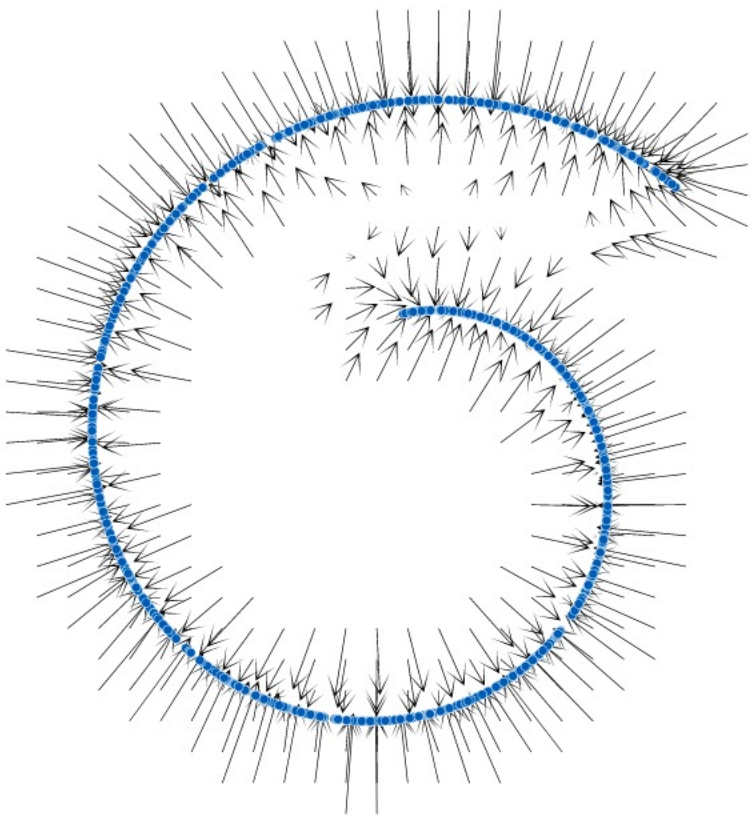
Perturbation, e.g. Gaussian noise

Denoising Autoencoders Examples



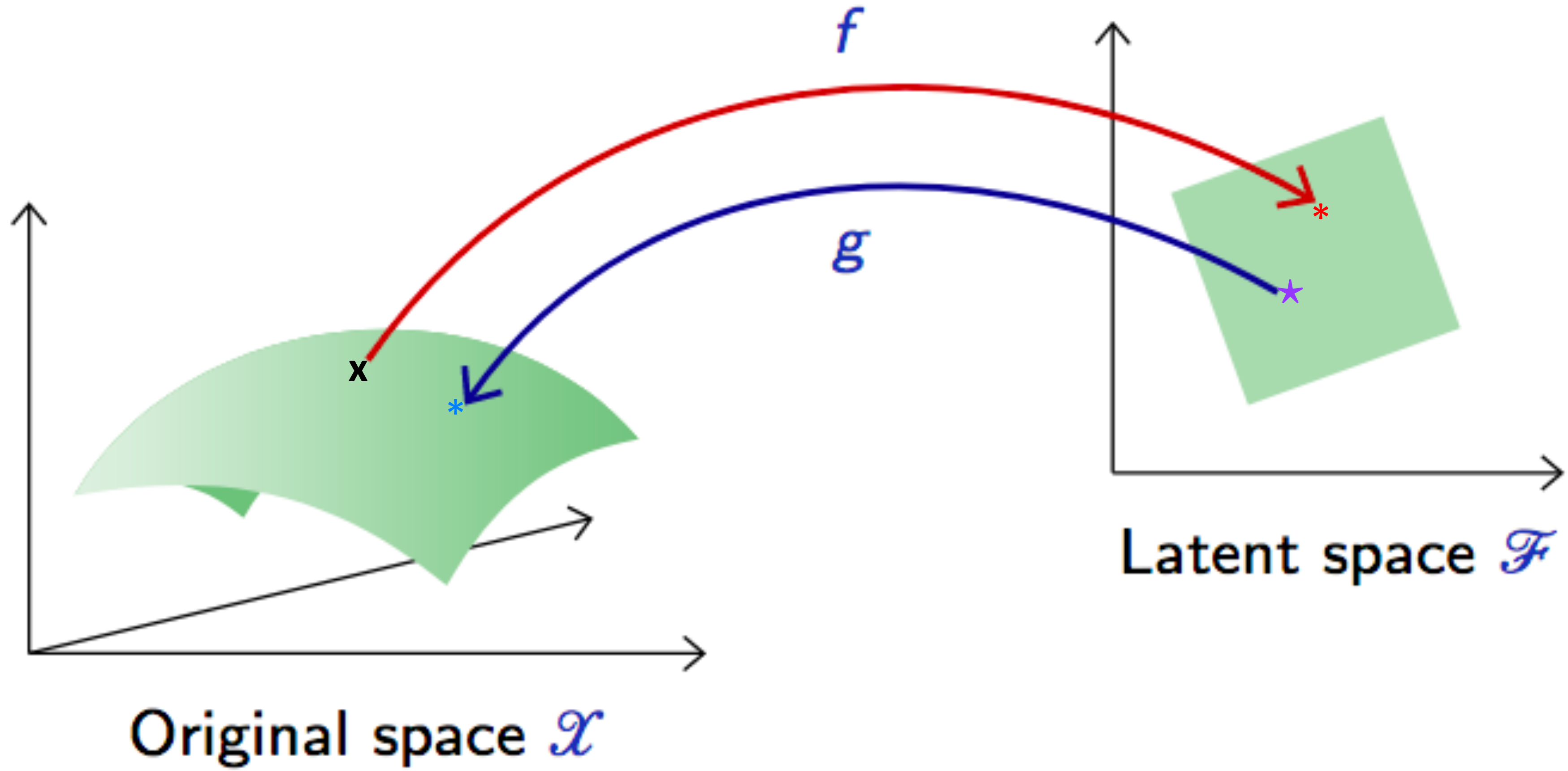


Denoising Autoencoders Examples

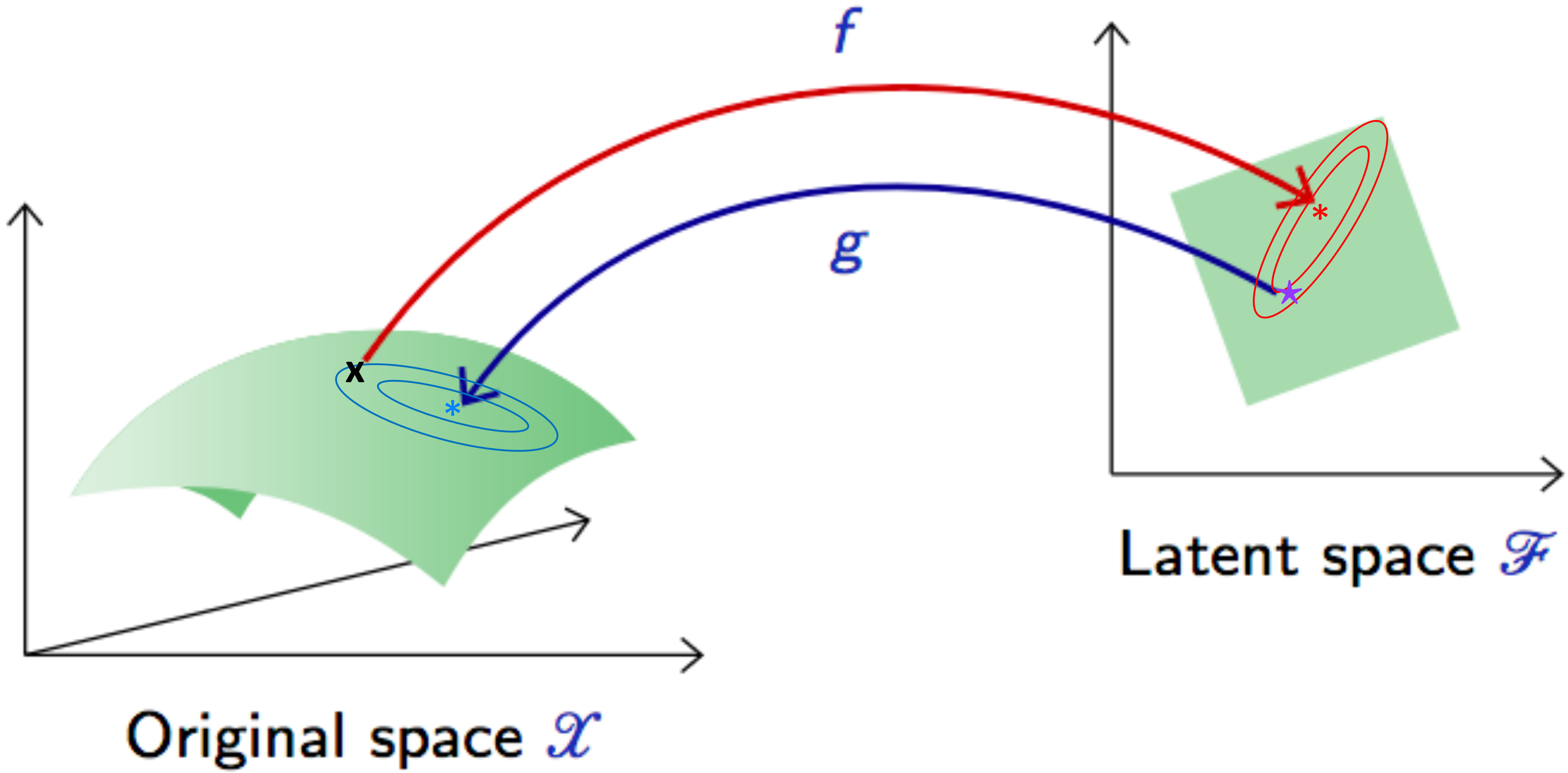


- **Autoencoder learns the average behavior**
- What if we care about these variations?
- Can we add a notion of variation in the autoencoder?

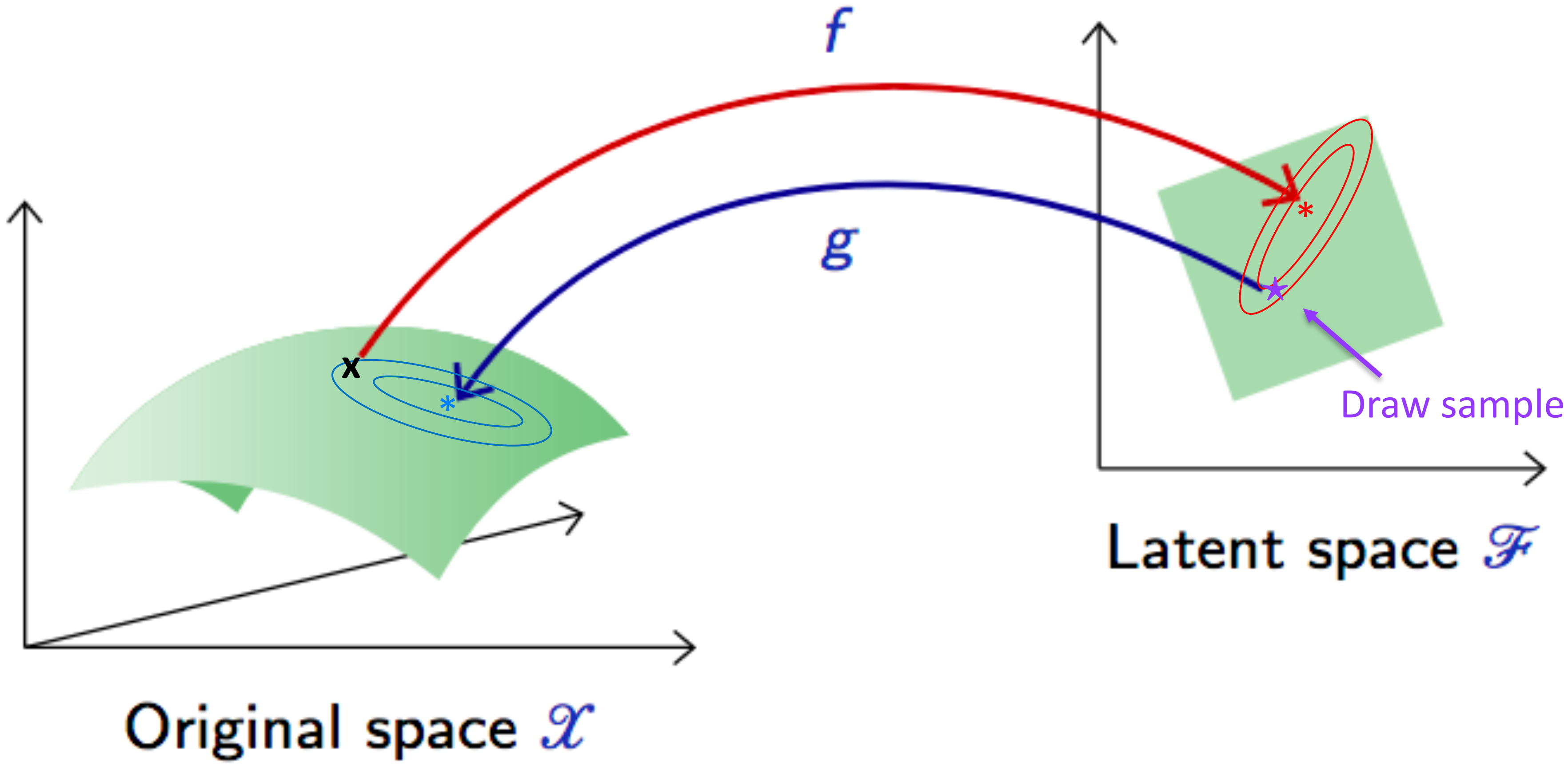
Autoencoder



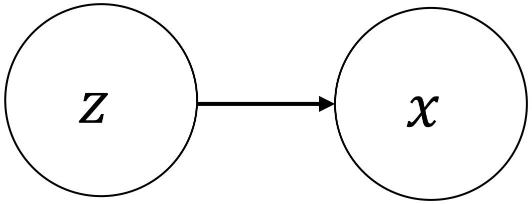
Variational Autoencoder



Variational Autoencoder



Latent Variable Models



- Observed random variable x depends on unobserved latent random variable z

- Joint probability: $p(x,z) = p(x|z)p(z)$

- $p(x|z)$ is stochastic generation process from $z \rightarrow x$

From Deterministic to Probabilistic Autoencoder

- Probabilistic relationship between data and latents

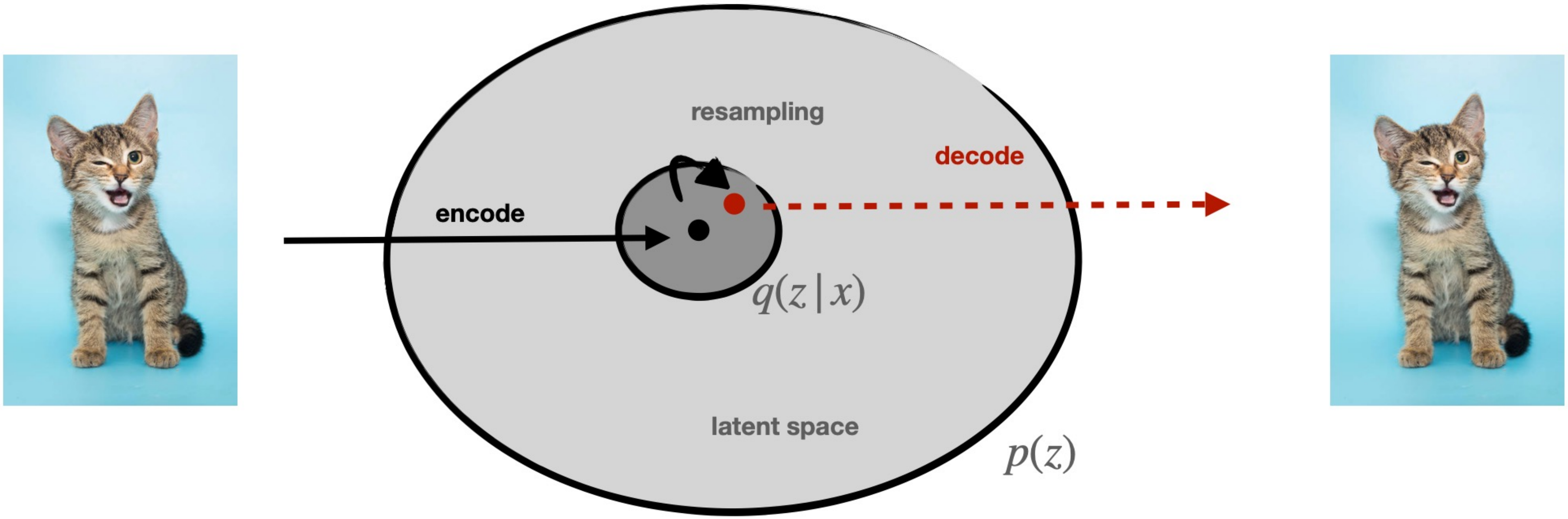
$$x, z \sim p(x, z) = p(x|z)p(z)$$

- Autoencoding

$$x \rightarrow q(z|x) \underset{sample}{\Longrightarrow} z \rightarrow p(x|z)$$

  - **Encoder:** Learn what latents can produced data: $q(z|x)$
  - **Decoder:** Learn what data is produced by latent: $p(x|z)$

Variational Autoencoder



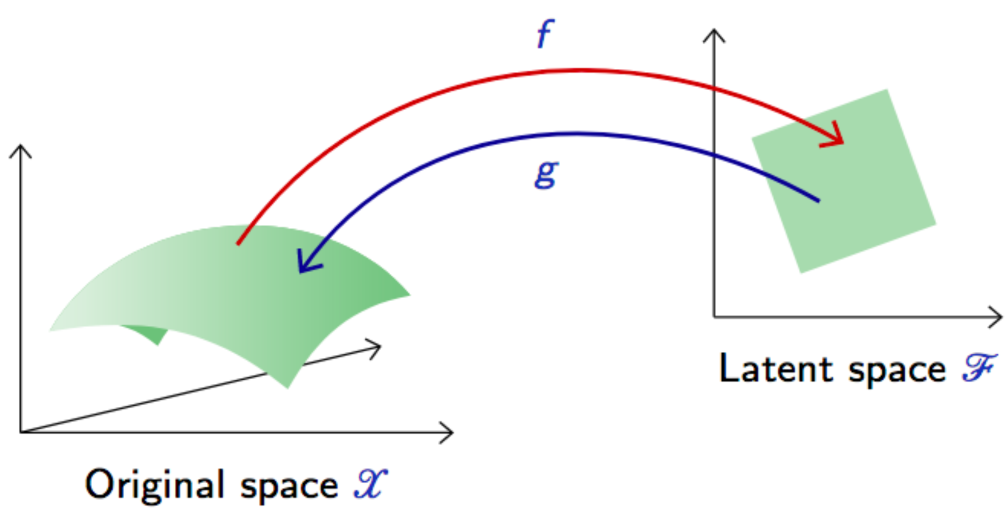
- Close-by points must decode to similar images

Image credit: L. Heinrich

Encoding

- Typical encoder maps input x to "average" point in latent space

$$f(x) = \mu(x)$$

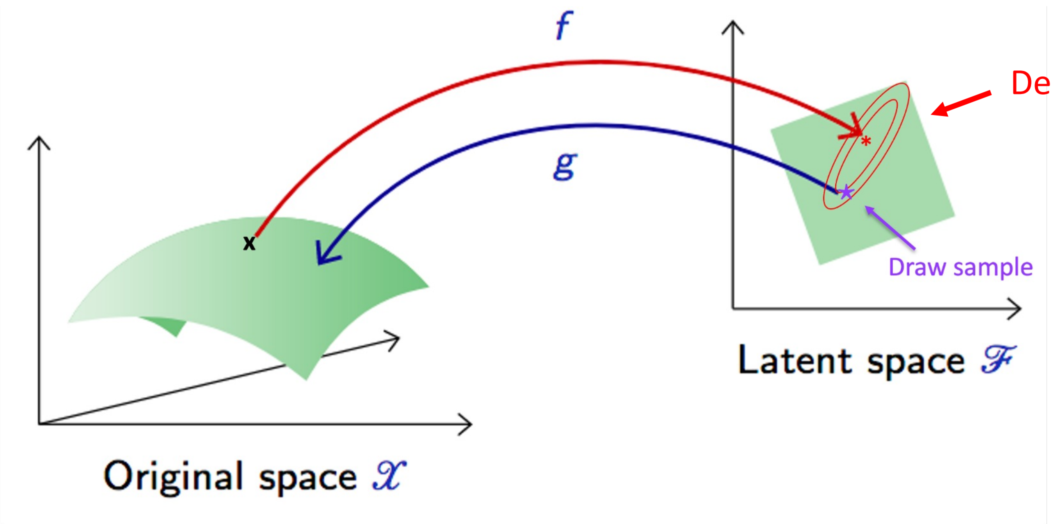


Encoding

- A VAE Encoder has two outputs: mean & variance function

$$f_\psi(x) = \{\mu_\psi(x), \sigma_\psi(x)\}$$

ψ are parameters of the NN



Encoding

- A VAE Encoder has two outputs: mean & variance function

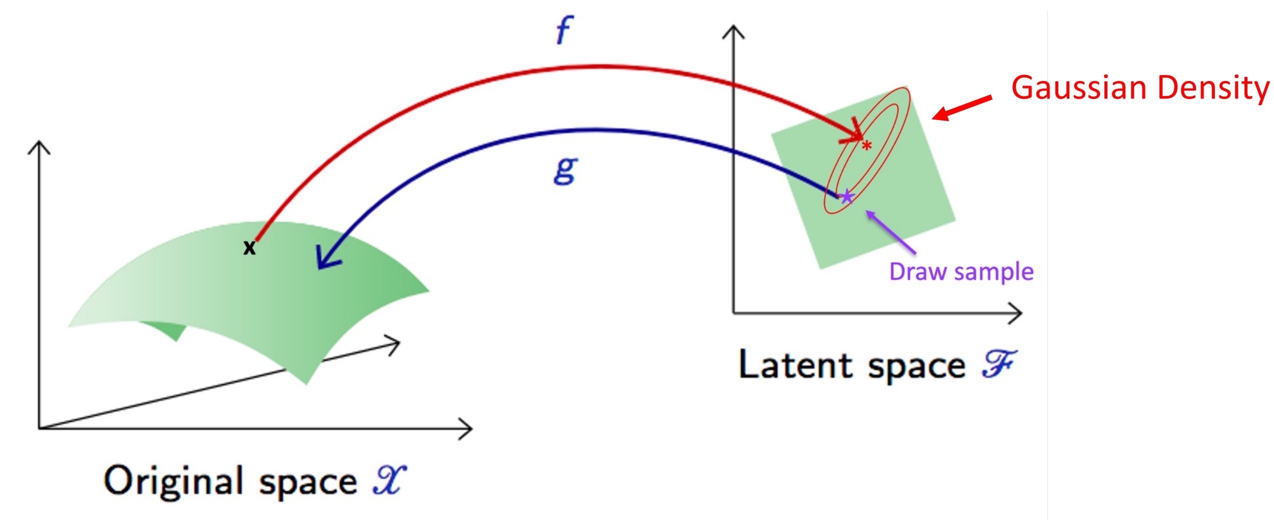
$$f_\psi(x) = \{\mu_\psi(x), \sigma_\psi(x)\}$$

ψ are parameters of the NN

- What is the probability of a point in latent space?

$$p_\psi(z|x) = N(z \mid \mu_\psi(x), \sigma_\psi(x))$$

Could choose different density
Gaussian is easiest



Encoding

- A VAE Encoder has two outputs: mean & variance function

$$f_\psi(x) = \{\mu_\psi(x), \sigma_\psi(x)\}$$

ψ are parameters of the NN

- What is the probability of a point in latent space?

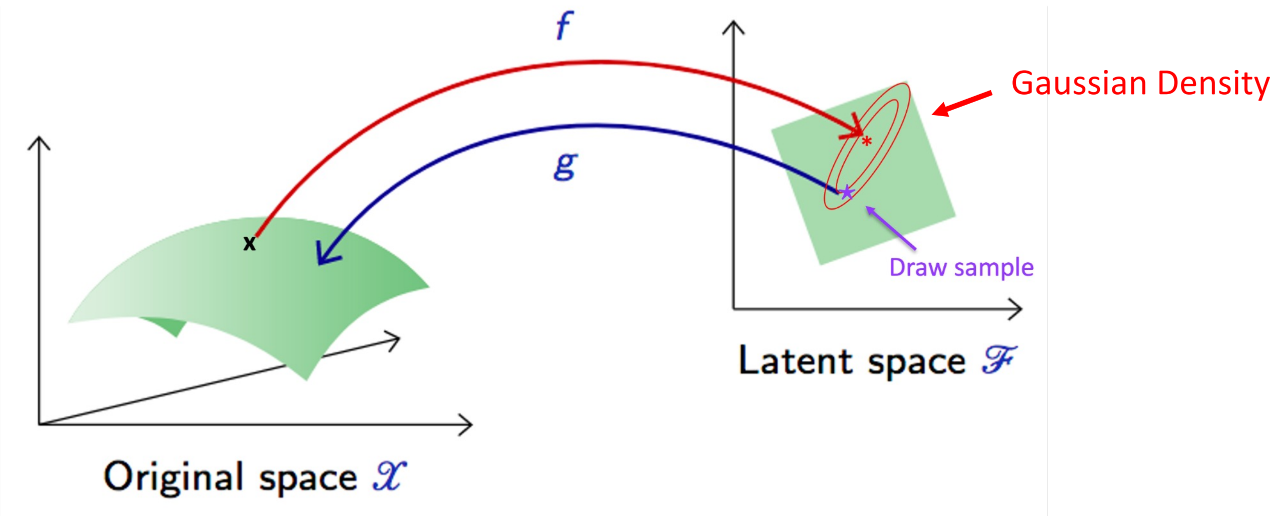
$$p_\psi(z|x) = N(z \mid \mu_\psi(x), \sigma_\psi(x))$$

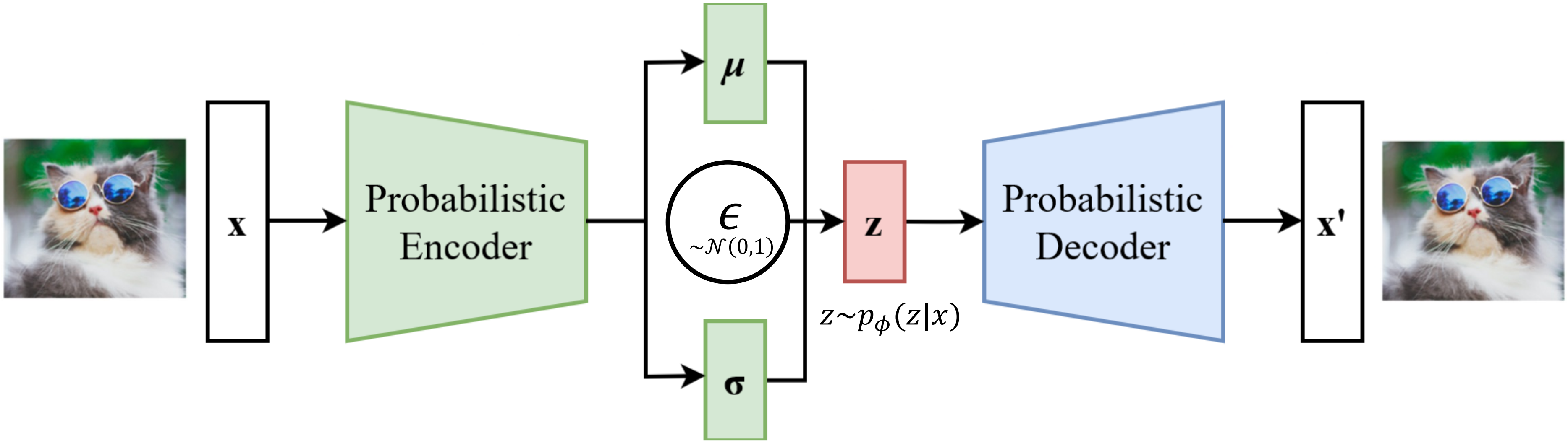
Could choose different density
Gaussian is easiest

- How do we draw a sample in latent space?

$$z = \sigma_\psi(x) * \epsilon + \mu_\psi(x) \qquad \epsilon \sim N(0, I)$$

Re-parameterization trick





Kingma, Welling, [1312.6114](1312.6114)
Rezende, Mohamed, Wierstra, [1401.4082](1401.4082)

Decoding

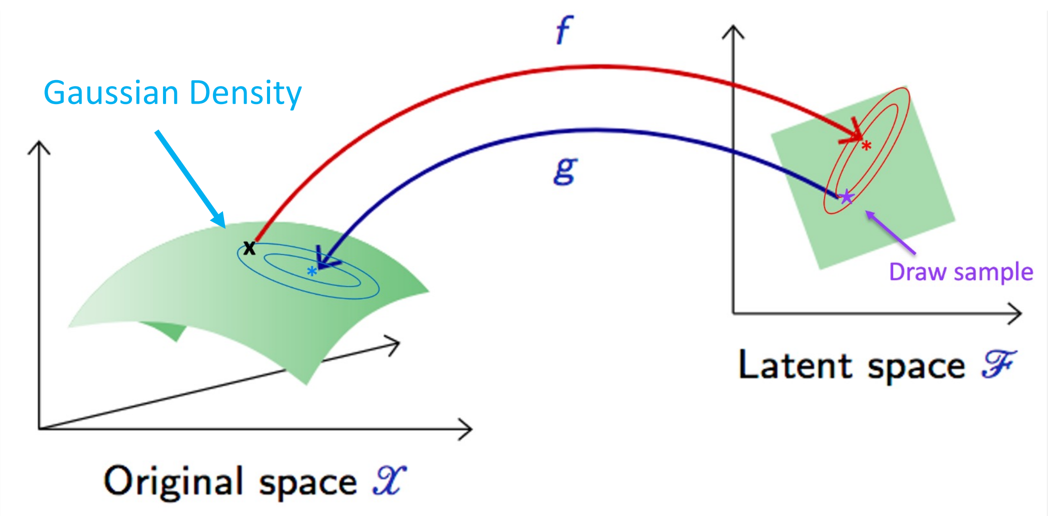
- Same as autoencoder

$$g_\theta(z) \equiv \mu_\theta(z)$$

θ are parameters of the NN

- Likelihood of an observation x

$$p_\theta(x|z) = N(x \mid \mu_\theta(z), I)$$



Decoding

- Same as autoencoder

$$g_\theta(z) \equiv \mu_\theta(z)$$

θ are parameters of the NN

- Likelihood of an observation x

$$p_\theta(x|z) = N(x \mid \mu_\theta(z), I)$$

- "**Reconstruction Loss**": Maximum likelihood

$$L_{reco} = \mathbb{E}_{z\sim q(z|x)}[\log p(x|z)]$$

Decoding

- Same as autoencoder

$$g_\theta(z) \equiv \mu_\theta(z)$$

θ are parameters of the NN

- Likelihood of an observation x

$$p_\theta(x|z) = N(x \mid \mu_\theta(z), I)$$

- "**Reconstruction Loss**": Maximum likelihood

$$L_{reco} = \mathbb{E}_{z\sim q(z|x)}[\log p(x|z)] \approx \frac{1}{N} \sum_{z_i \sim q(z|x)} \log N(x \mid g_\theta(z_i), I)$$

Decoding

- Same as autoencoder

$$g_\theta(z) \equiv \mu_\theta(z)$$

θ are parameters of the NN

- Likelihood of an observation x

$$p_\theta(x|z) = N(x \mid \mu_\theta(z), I)$$

- "**Reconstruction Loss**": Maximum likelihood

$$L_{reco} = \mathbb{E}_{z\sim q(z|x)}[\log p(x|z)] \approx -\frac{1}{N}\sum_{z_i\sim q(z|x)} \left(x - g_\theta(z_i)\right)^2$$

Same as the autoencoder loss

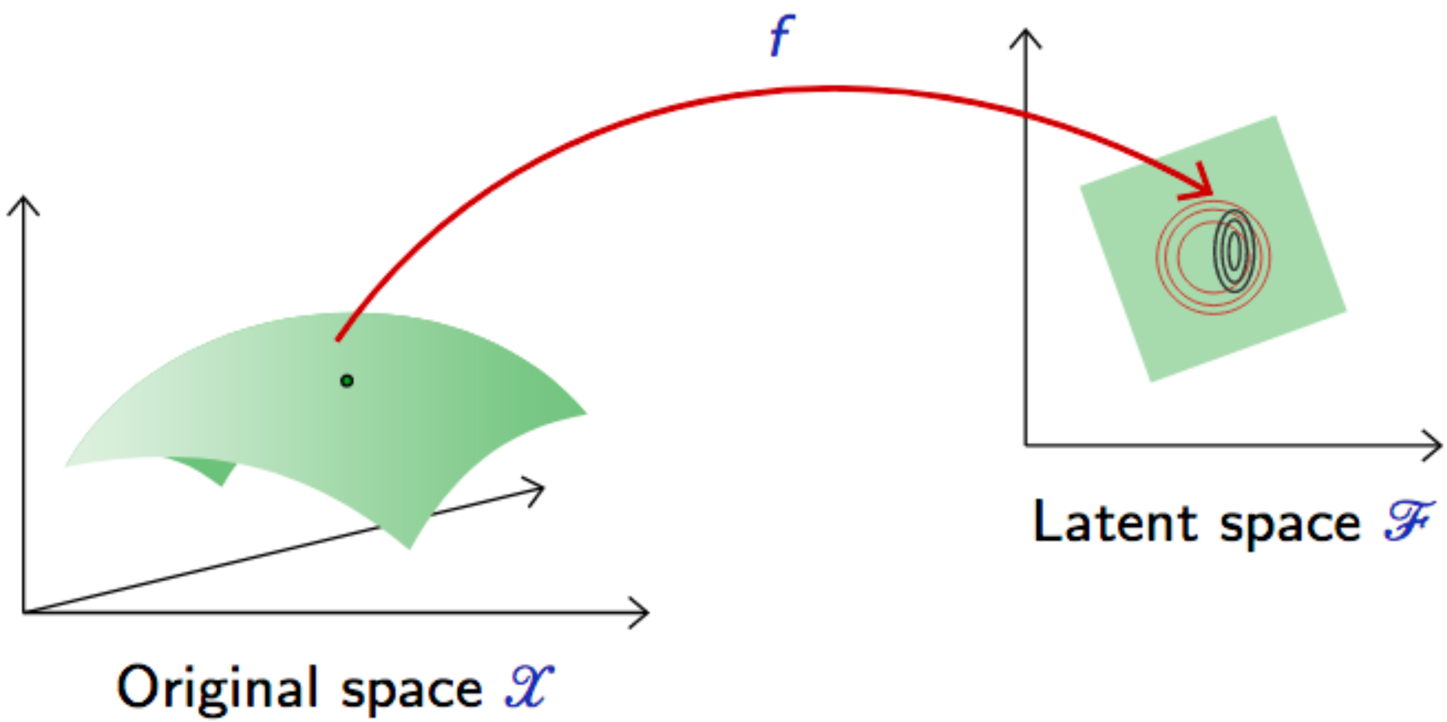
Variational Autoencoder Training Loss

- How do we make sure system doesn't collapse to an autoencoder (i.e. VAE encoder only predicts mean)?

Variational Autoencoder Training Loss

- How do we make sure system doesn't collapse to an autoencoder (i.e. VAE encoder only predicts mean)?

- Use prior $p(z)$ for the latent space distribution, **need to ensure the encoder is consistent with prior**



Variational Autoencoder Training Loss

- Constrain difference between distributions with **Kullback–Leibler divergence**

$$D_{KL}[q(z|x)|p(z)] = \mathbb{E}_{q(z|x)}\left[\log\frac{q(z|x)}{p(z)}\right] = \int q(z|x)\log\frac{q(z|x)}{p(z)}\,dz$$

  - $D_{KL}[q|p] \geq 0$ and is only 0 when $q = p$

Variational Autoencoder Training Loss

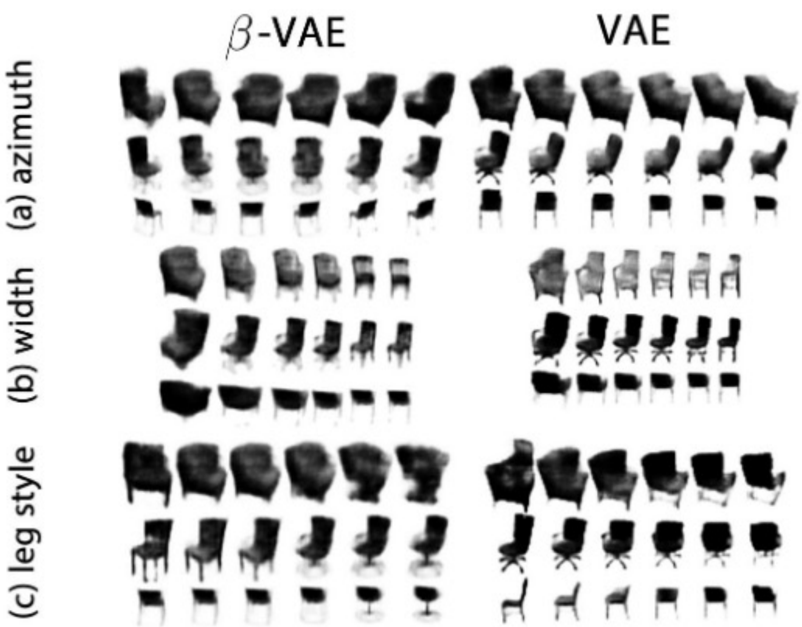
- Constrain difference between distributions with **Kullback–Leibler divergence**

$$D_{KL}[q(z|x)|p(z)] = \mathbb{E}_{q(z|x)}\left[\log\frac{q(z|x)}{p(z)}\right] = \int q(z|x)\log\frac{q(z|x)}{p(z)}\,dz$$

- VAE full objective

$$\max_{\theta,\psi} L(\theta,\psi) = \max_{\theta,\psi}\Big[\underbrace{\mathbb{E}_{q_\psi(z|x)}[\log p_\theta(x|z)]}_{\text{Reconstruction Loss}} - \underbrace{D_{KL}[q_\psi(z|x)|p(z)]}_{\text{Regularization of Encoder}}\Big]$$

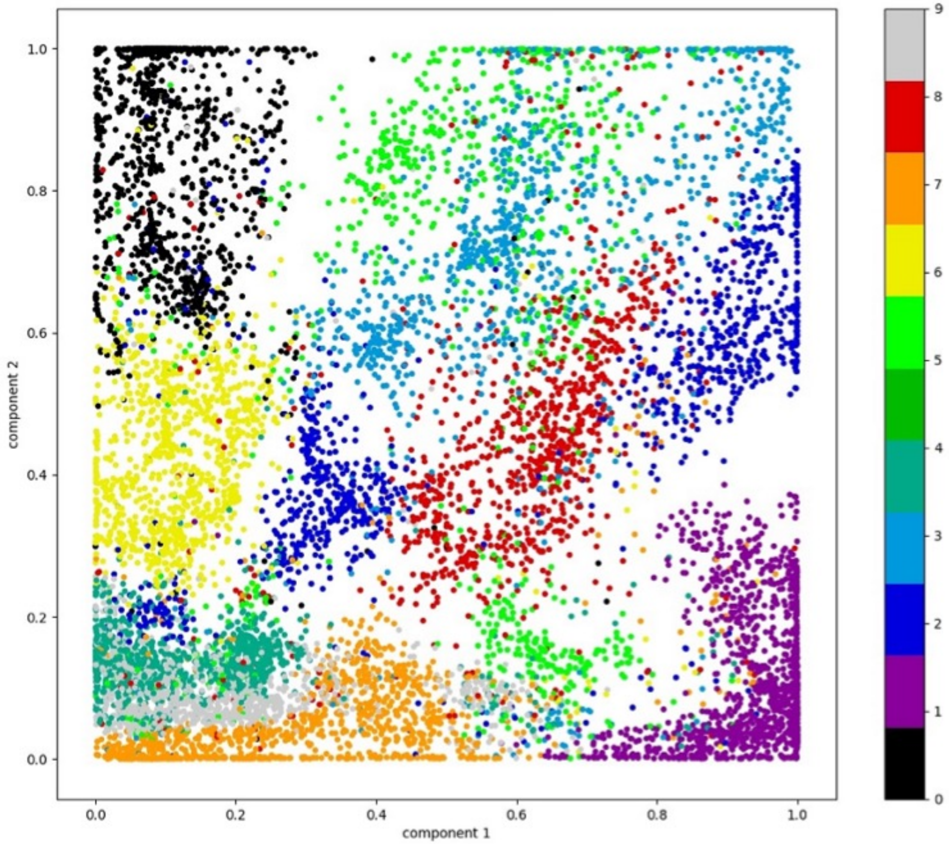
Examples



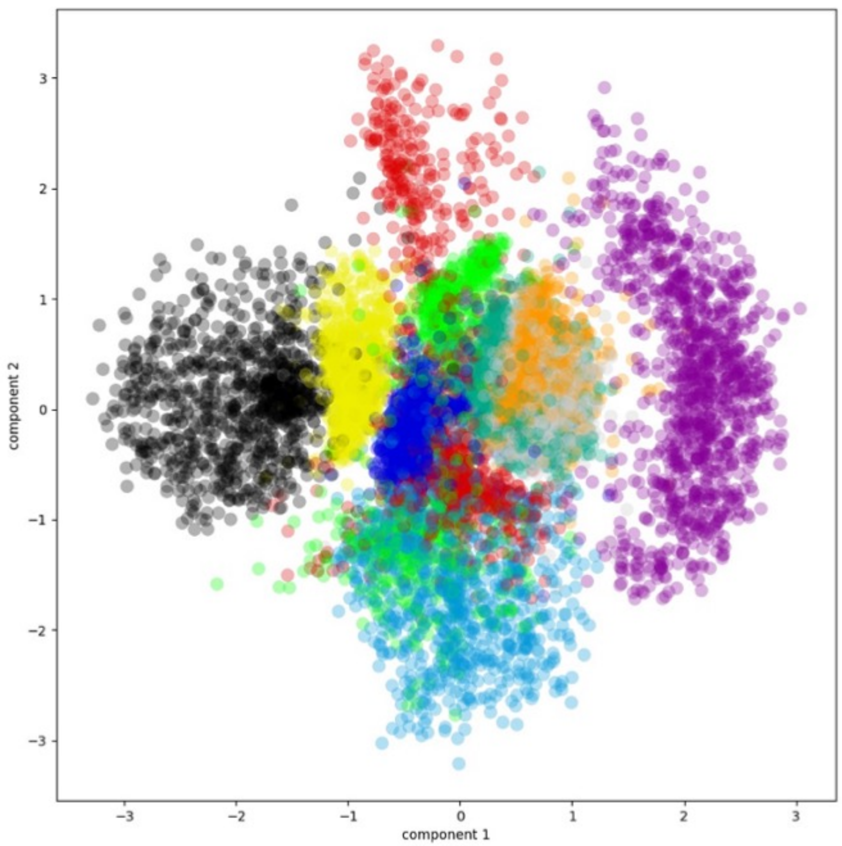
Higgins et al., [2017]

Comparing Latent Spaces

Autoencoder

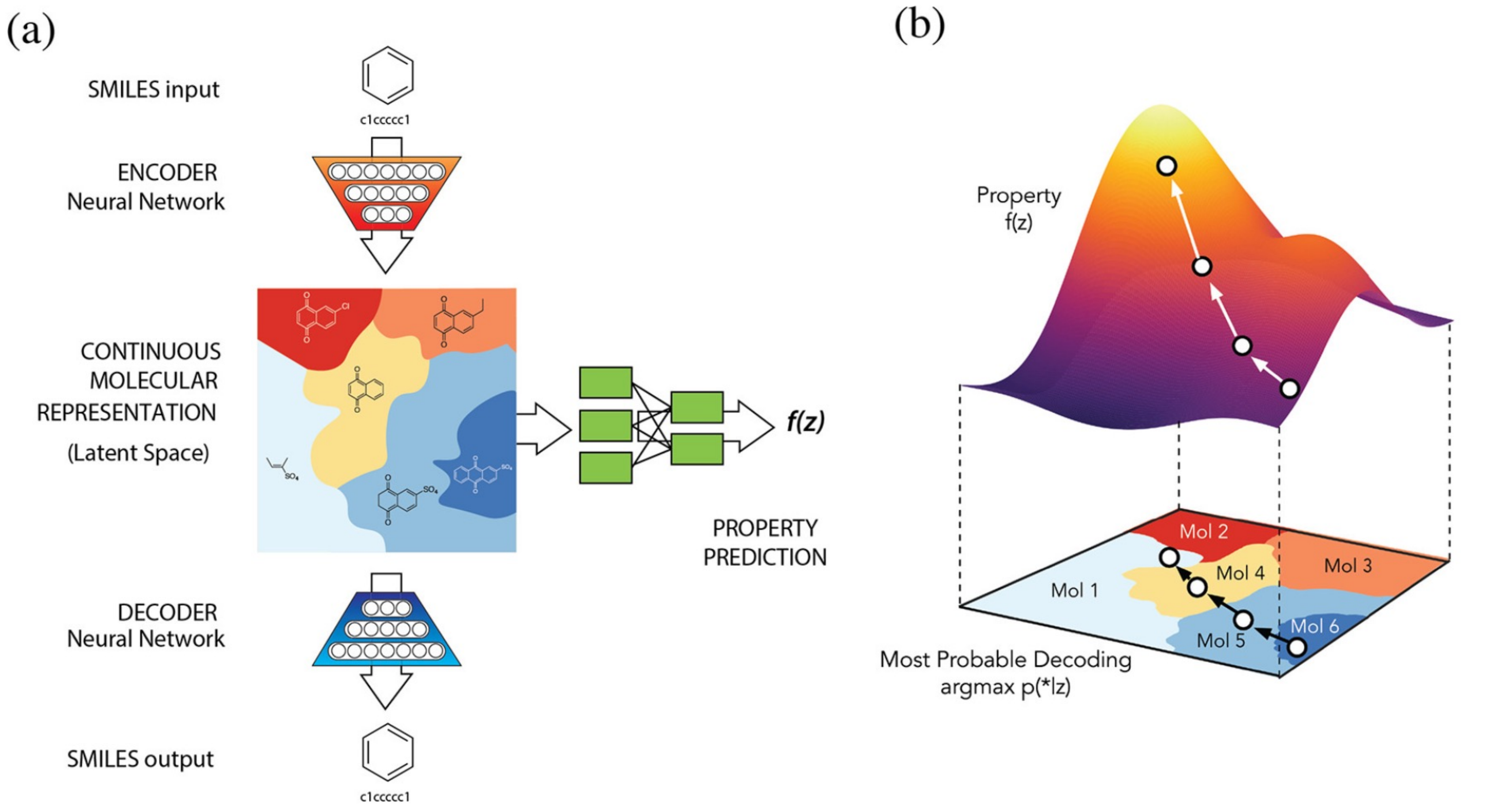


Variational Autoencoder



Data: MNIST data set of hand-written digits

Examples



Design of new molecules with desired chemical properties.
(Gomez-Bombarelli et al, 2016)

Slide credit: [G. Louppe]

What have we learned?

- In generative modeling,
want to learn the lower dimensional degrees of freedom that describe the features of the data

- "Degrees of freedom" are modeled with a latent distribution (kept simple for convenience) and complex neural network mappings

- Need to think about **probabilistic systems**

- Design loss around this probabilistic model

The Zoo of Generative Models…

GAN: Adversarial training

VAE: maximize variational lower bound

Flow-based models: Invertible transform of distributions

Diffusion models: Gradually add Gaussian noise and then reverse

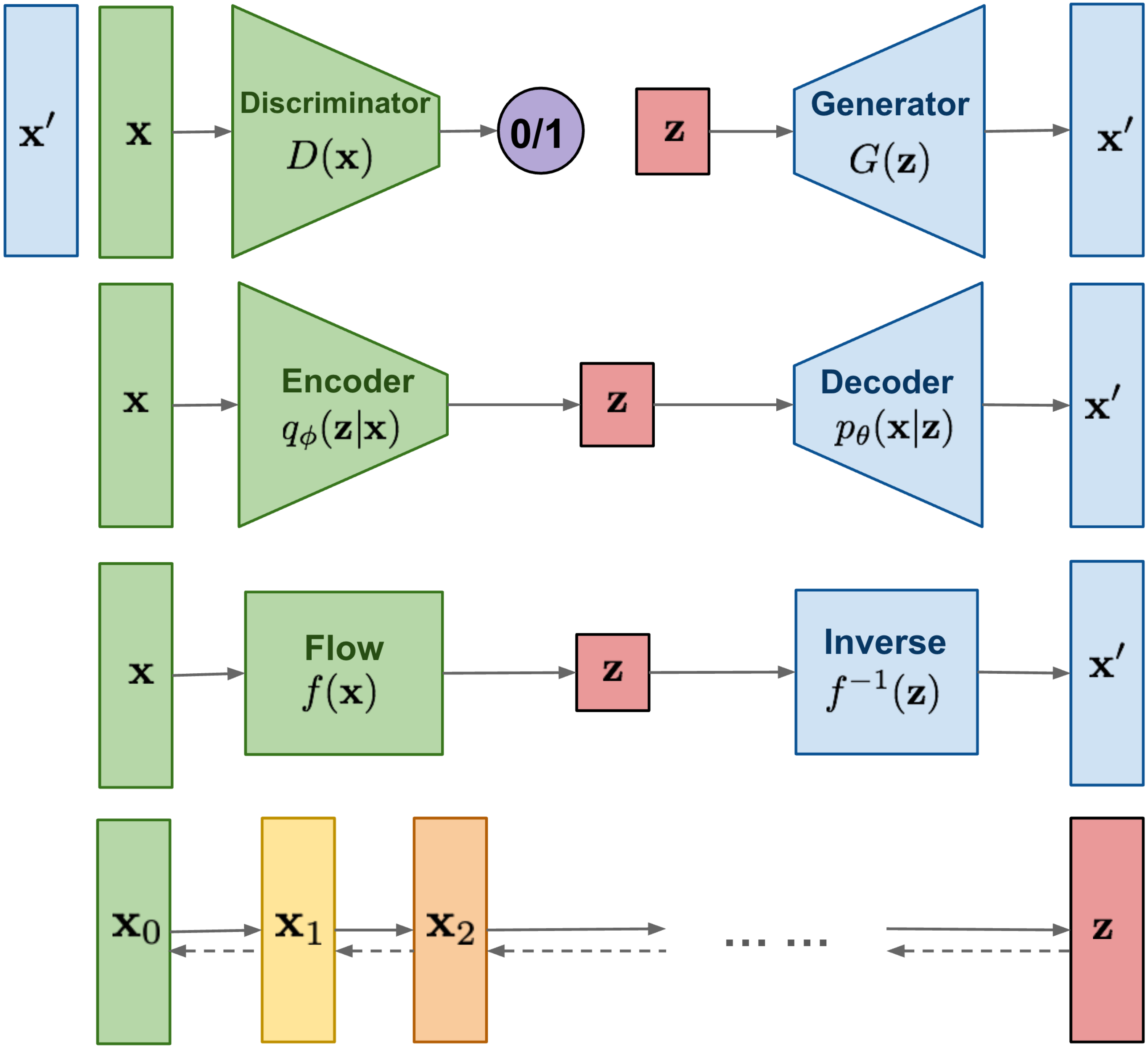
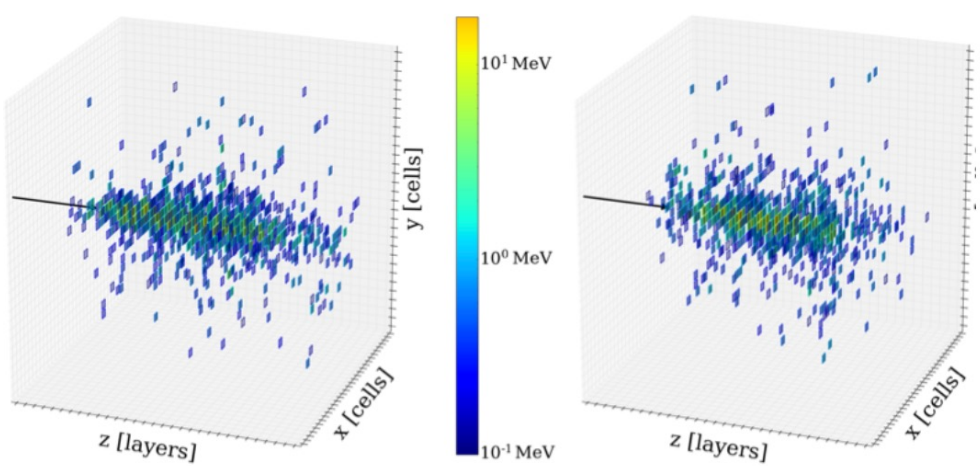


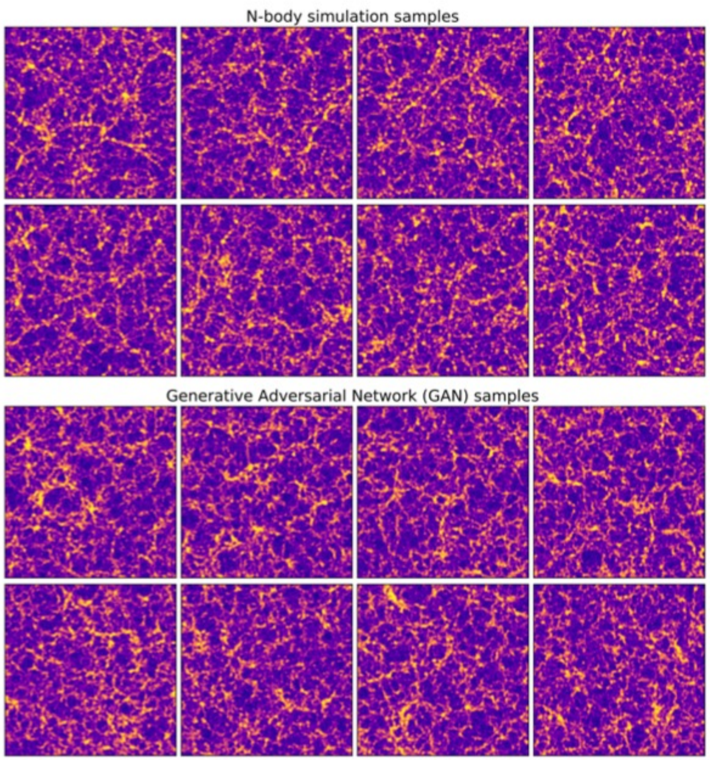
Image credit: Lilian Weng

Generative Models in Physics

- Often studied for fast approximate simulation, simulation-based inference, optimization, anomaly detection, …
  - See talks by [G. Kasieczka](), [D. Shih](), [B. Nevin](), [A. Edelen]()



2005.05334



1801.09070

Summary

- Deep neural networks are an extremely powerful class of models

- We can express our inductive bias about a system in terms of model design, and can be adapted to a many types of data

- Even beyond classification and regression, deep neural networks allow powerful unsupervised learning and Generative modeling!