# Deep Generative Models I

## Variational Autoencoders

Machine Perception

Otmar Hilliges

9 April 2020

Advanced Interactive Technologies

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

# Last week(s)

CNNs

RNNs

Fully convolutional networks

# Next week(s)

Discriminative models => generative models

Step I: (Variatonal autoencoders)

# **Supervised** vs Unsupervised learning

**Supervised Learning**

**Data:** $(\boldsymbol{x}, y)$
$\boldsymbol{x}$ is data, $y$ is label

**Goal**: Learn a function that maps $\boldsymbol{X} -> y$

**Examples:** Classification, regression,
Course content so far…

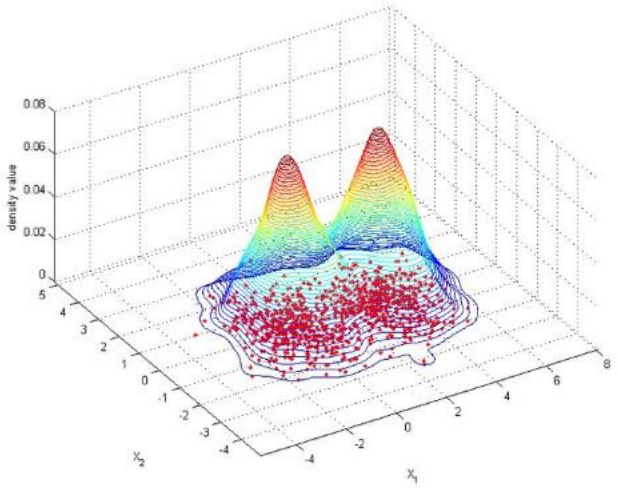
# Supervised vs **Unsupervised** learning

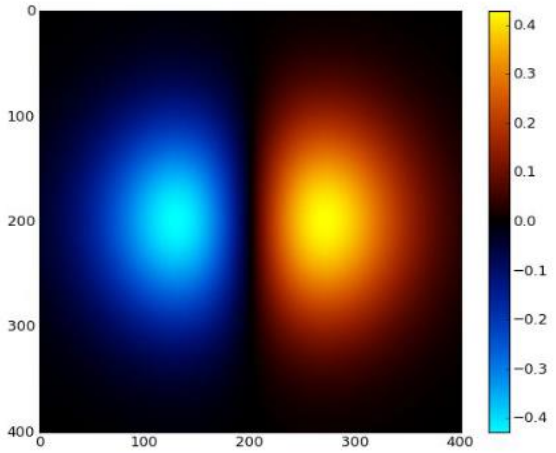
images (left and right) CC0 public domain

## Unsupervised Learning

**Data:** $\boldsymbol{X}$

$\boldsymbol{x}$ are data points. No labels



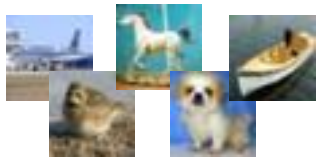




**Goal**: Learn some underlying *structure* of the data

**Examples:** clustering, feature learning, density estimation, etc.

# Generative Modelling

Given training data, generate new samples, drawn from "same" distribution



Training samples $\sim p_{data}(x)$

Generated samples $\sim p_{model}(x)$

We want $p_{model}(x)$ to be similar to $p_{data}(x)$

# Why generative modelling:

better classify

novel samples
- data scarcity

modelling capacity

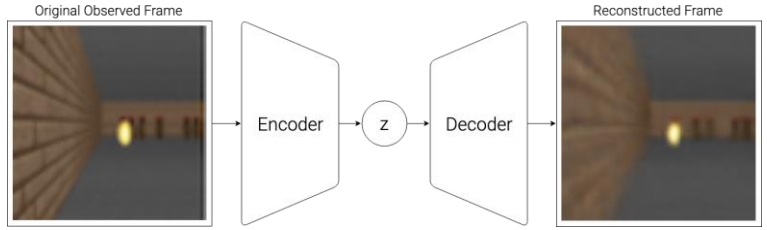
better generalization
- deal with imbalanced data

discover underlying factors
⇒ explore features

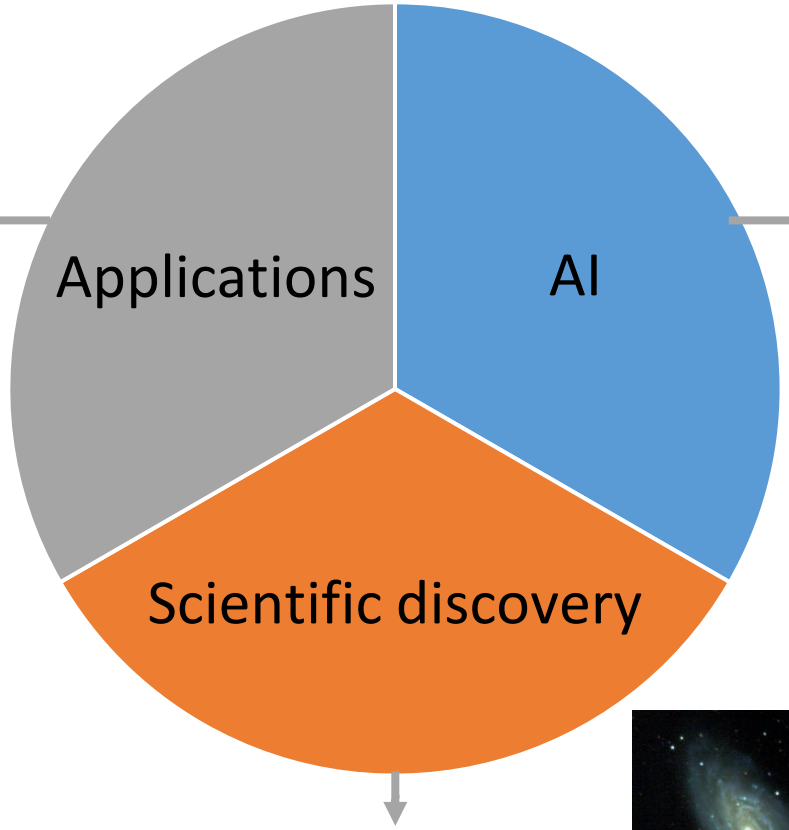
ability to sample

data augmentation

# Generative Modelling - Motivation




[Ha et al. NeurIPS ‘18]



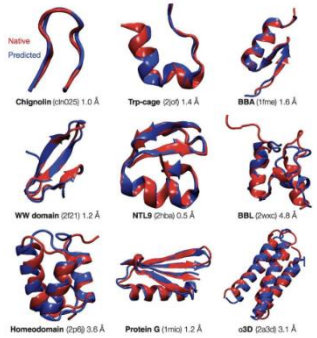

Super resolution
Compression
Text-to-speech

[SR-GAN. Ledig et al. CVPR ‘17]

Drug discovery
Astronomy
High-energy physics

[Celeste. Regier et al. MLR ‘15]

Planning
“Curiosity”
Model-based RL




[Gómez-Bombarelli, et al. ACS ’16]

# Learning-based Image Synthesis

2014

Goodfellow et.al, NIPS'14

2015

Radford et.al ICLR'16

2016

Liu et.al, NIPS'16

2017

Karras et.al, ICLR'18

2018

Karras et.al, arXiv 2018

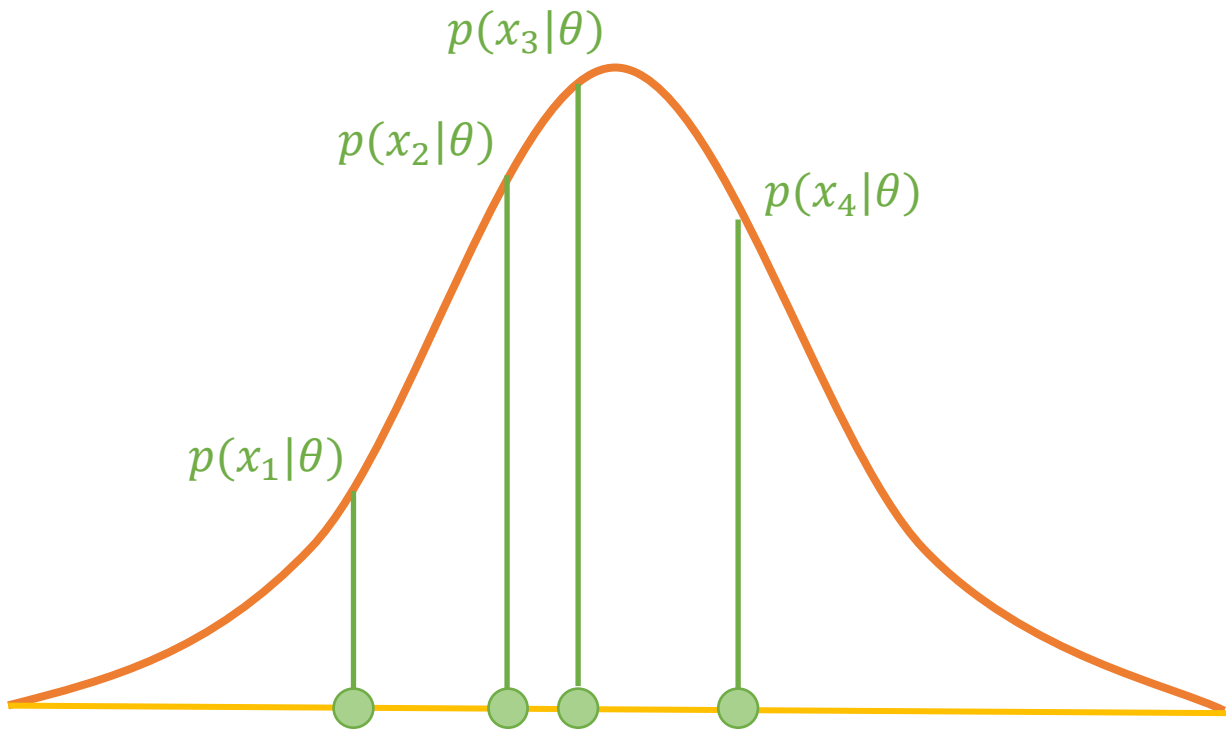
# Recap:
# Principles of Estimation

1. Explicit parametric models and likelihood
2. Implicit models

# Likelihood and Maximum Likelihood (MLE)

[Fisher, 1929]

- Likelihood is a function of the model parameters
- MLE extremely successful:
    - e.g. least squares and cross-entropy are MLE estimators



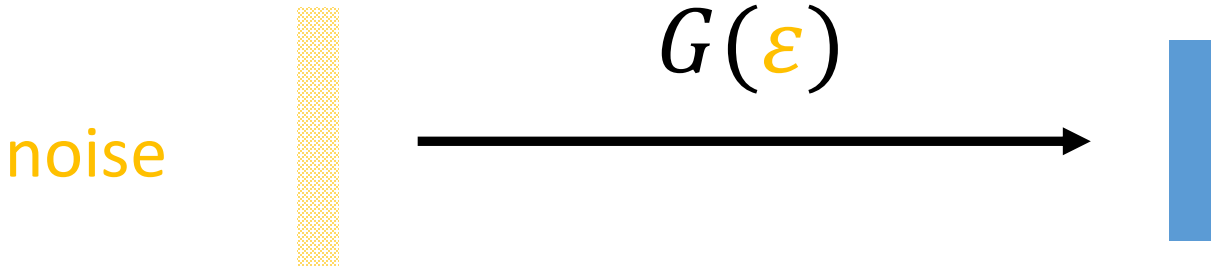

Goal: maximize likelihood function

$$L(\theta) = \prod_i p(x_i|\theta)$$

$$\log L(\theta) = \log \prod_i p(x_i|\theta)$$

$$\log L(\theta) = \sum_i \log p(x_i|\theta)$$

# Implicit Model / Neural Sampler / Likelihood-free Model



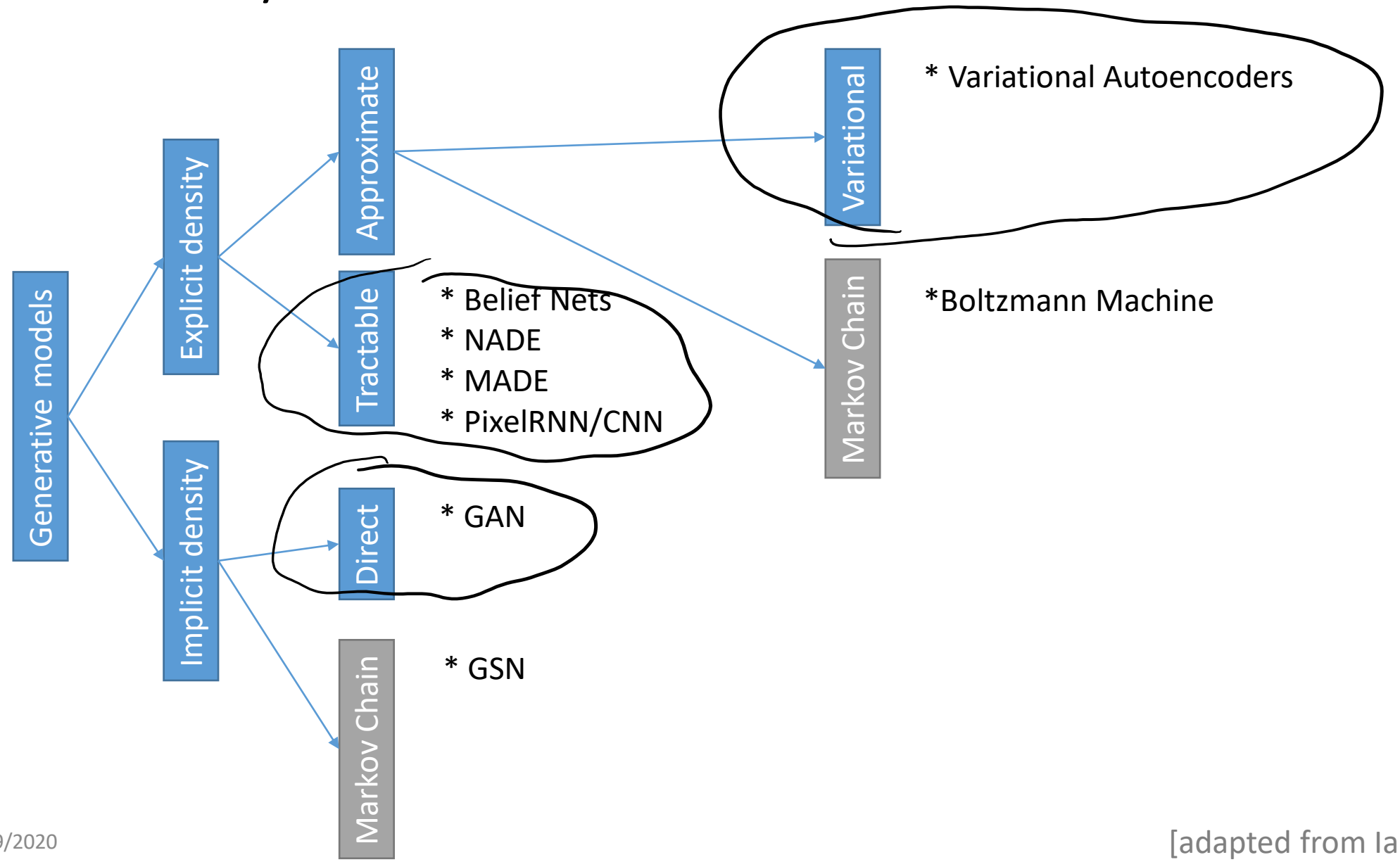

Pros:

- Highly expressive model class (universal)
- Density function not defined or intractable

Cons:

- Lack of theory and learning algorithms compared to explicit models

# Taxonomy of Generative Models





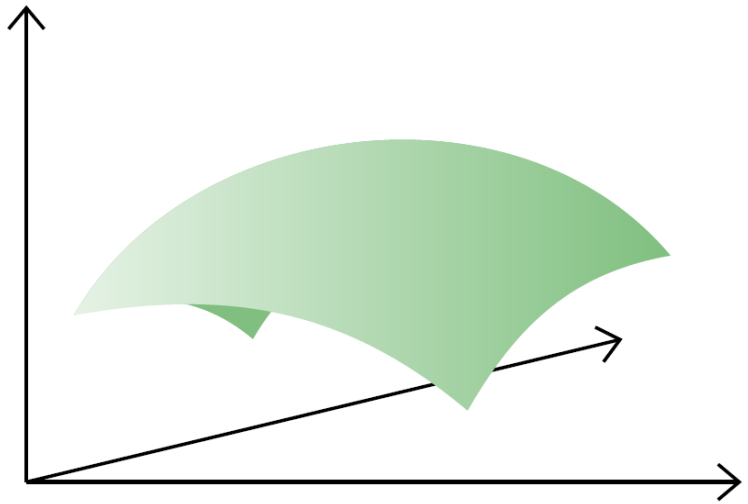

[adapted from Ian Goodfellow]

# Learning latent spaces

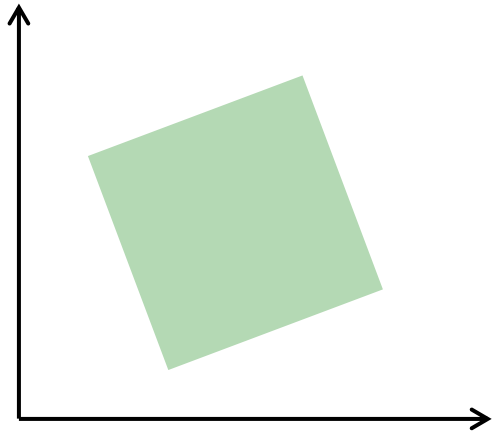
Finding “meaningful degrees of freedom" that describe the high-dimensional signal with lesser dimensions.

# Autoencoders

An autoencoder consists of an encoder and a decoder.



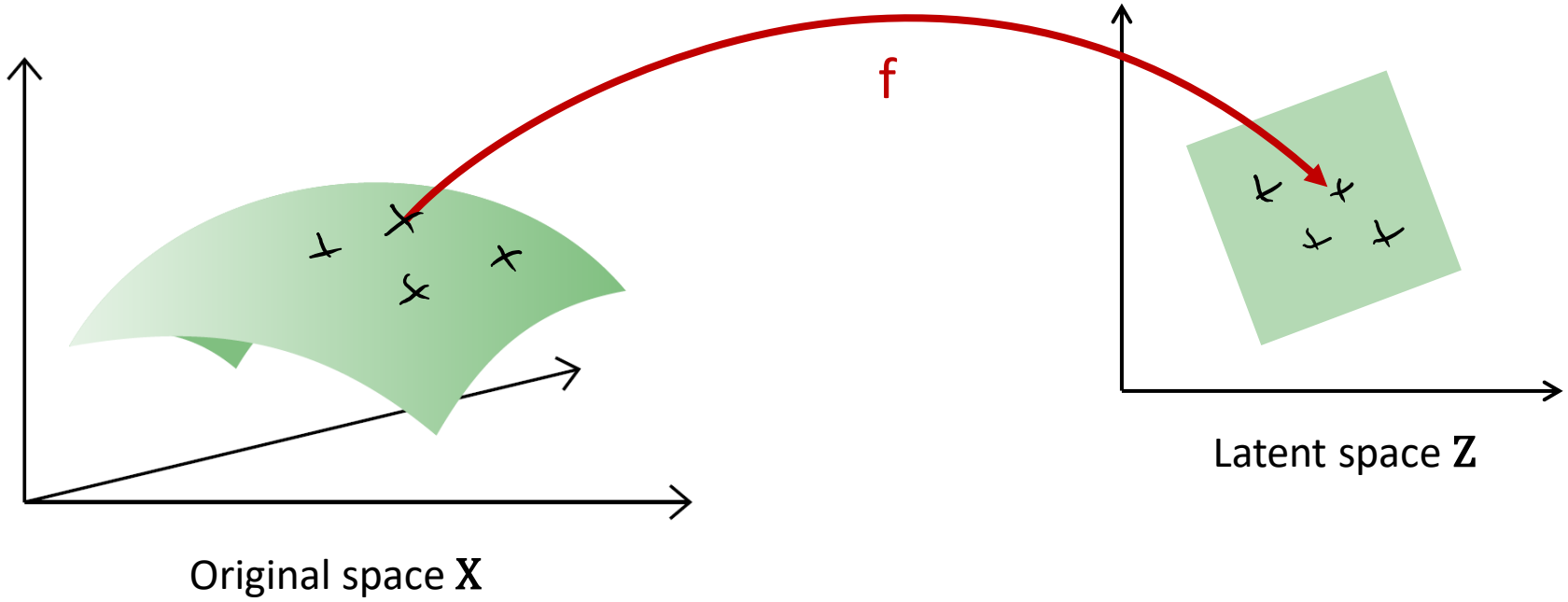
Original space $\mathbf{X}$



Latent space $\mathbf{Z}$

# Autoencoders

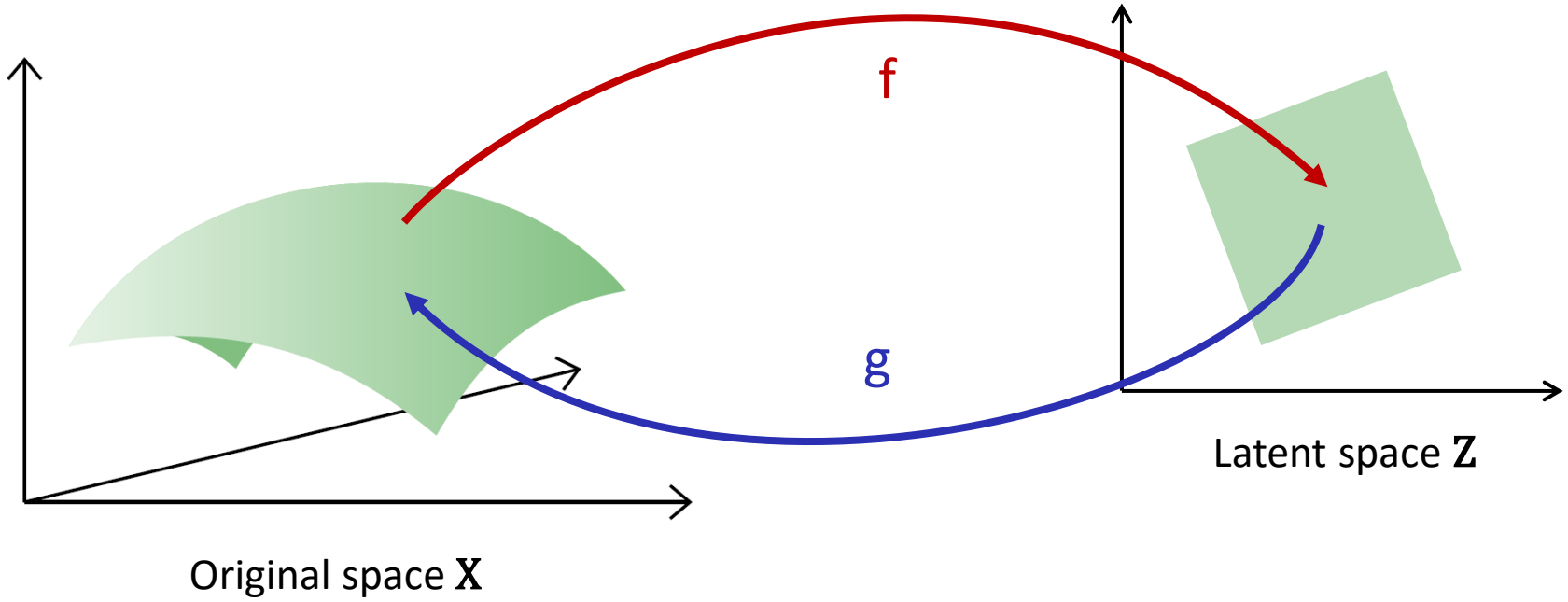
An autoencoder consists of an encoder and a decoder.




The encoder f projects the original input space $\mathbf{X}$ into a latent space $\mathbf{Z}$.

# Autoencoders

An autoencoder consists of an encoder and a decoder.

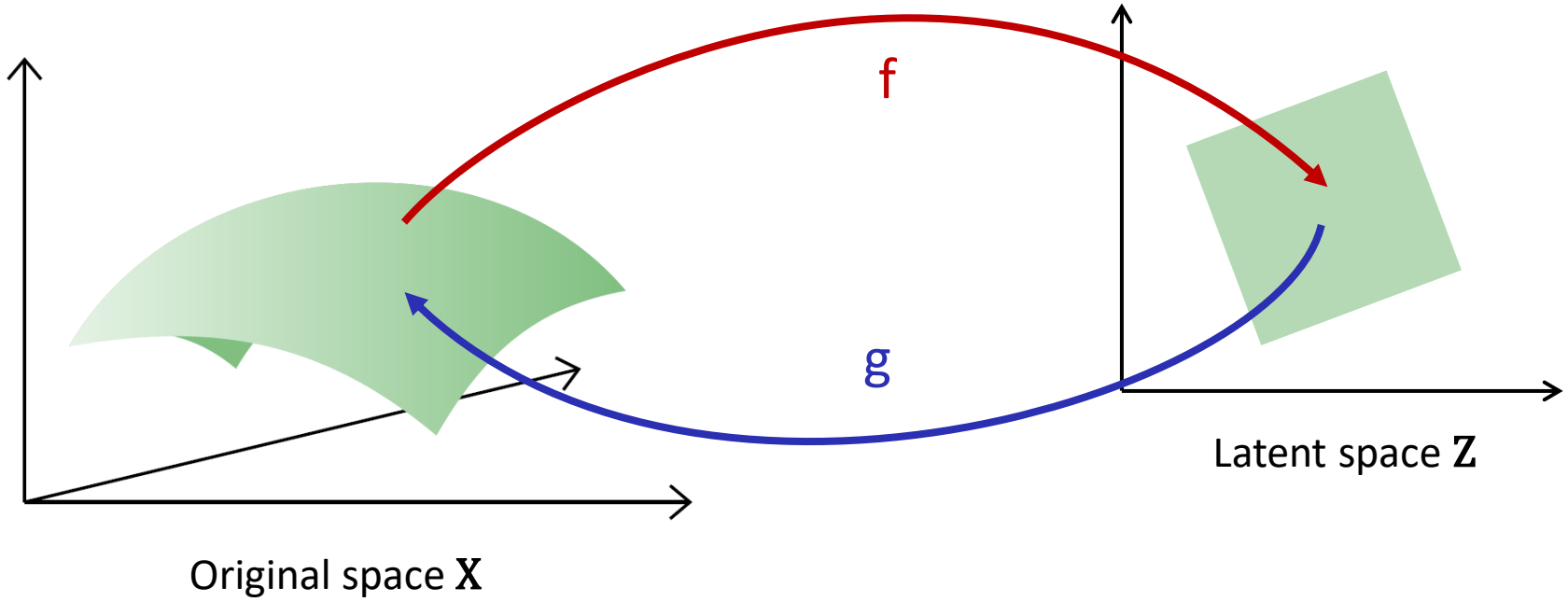



The encoder f projects the original input space $\mathbf{X}$ into a latent space $\mathbf{Z}$.

The decoder g maps samples from $\mathbf{Z}$ back to input space $\mathbf{X}$.

# Autoencoders

An autoencoder consists of an encoder and a decoder.

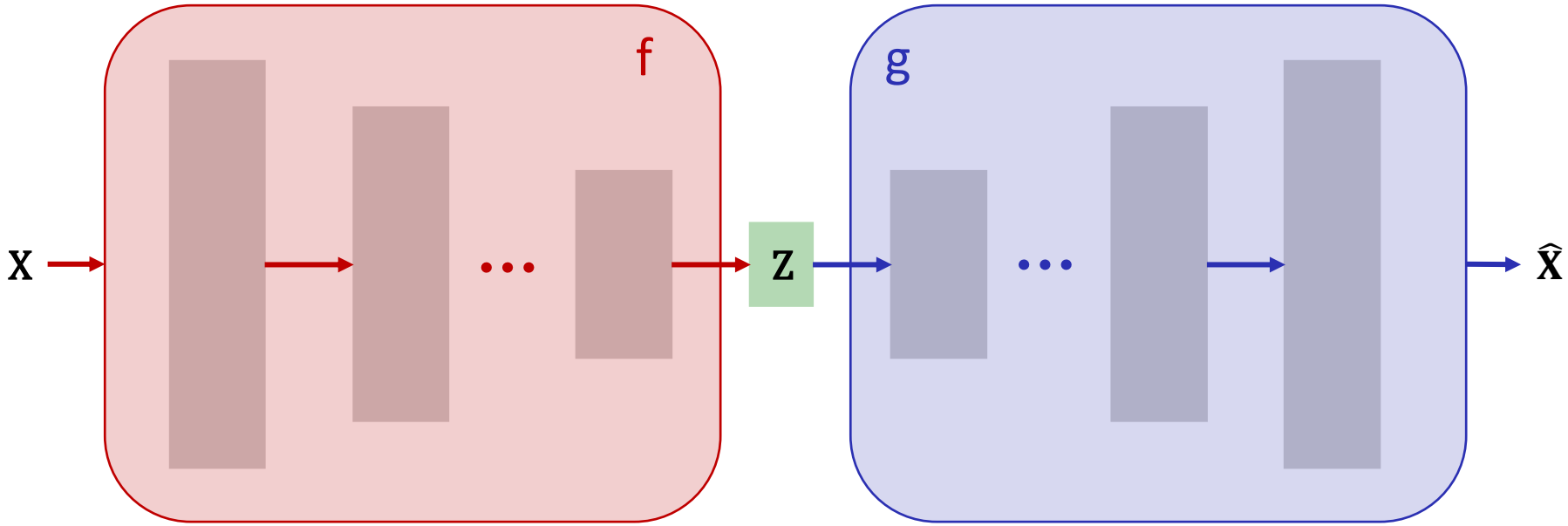



The encoder f projects the original input space $\mathbf{X}$ into a latent space $\mathbf{Z}$.

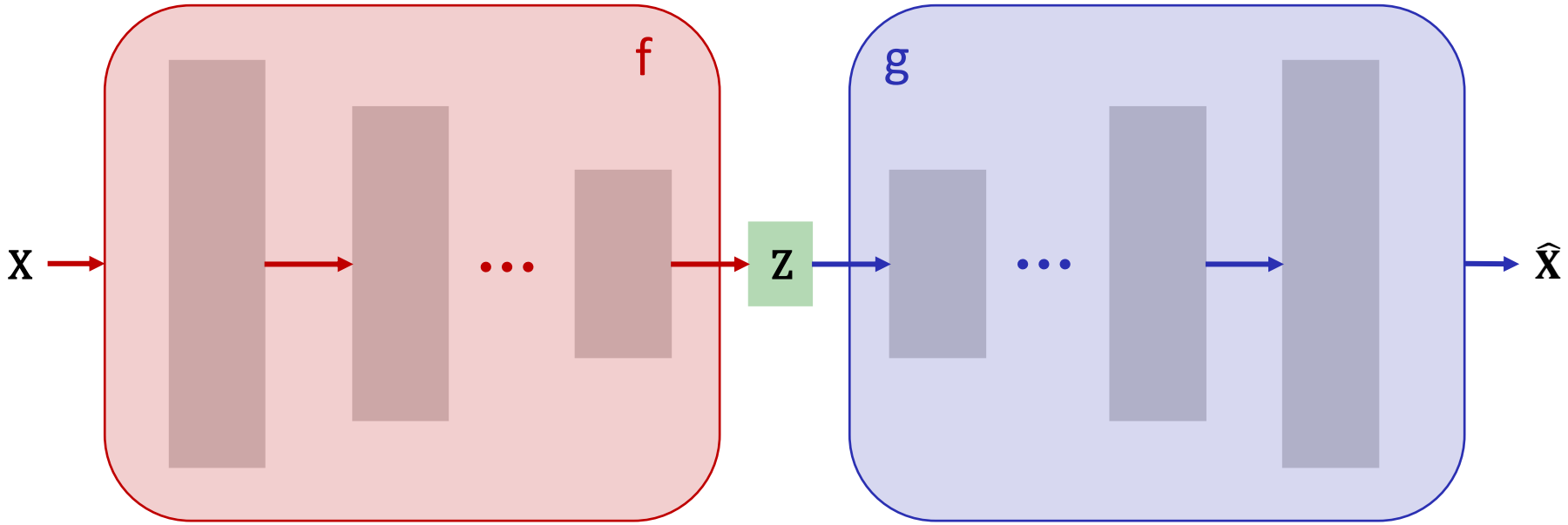
The decoder g maps samples from $\mathbf{Z}$ back to input space $\mathbf{X}$.

*[g ∘ f]* approximates the identity function on the data.

# Autoencoders



# Autoencoders

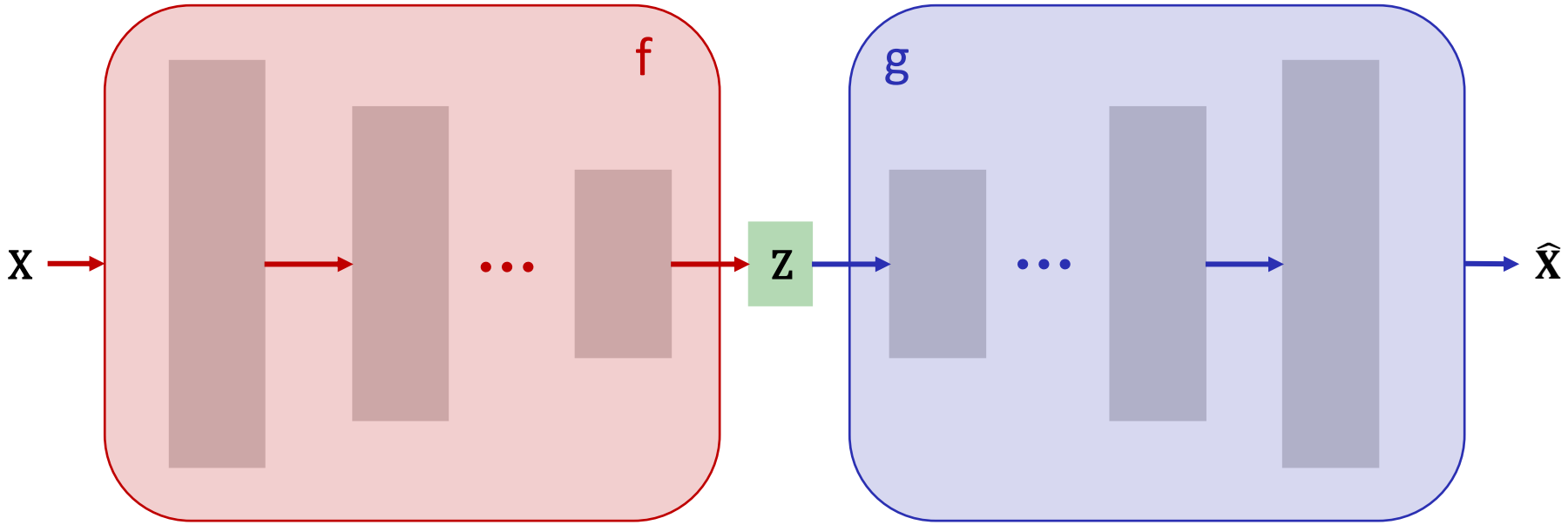



Given two parametrized mappings $f(\cdot\,;\theta_f)$ and $g(\cdot\,;\theta_g)$ training consists of minimizing

$$\hat{\theta}_f, \hat{\theta}_g = \underset{\theta_f,\theta_g}{\operatorname{argmin}} \sum_n^N \|x_n - \hat{x}_n\|^2$$

$$\hat{\theta}_f, \hat{\theta}_g = \underset{\theta_f,\theta_g}{\operatorname{argmin}} \sum_n^N \|x_n - g(f(x_n))\|^2$$

# Autoencoders

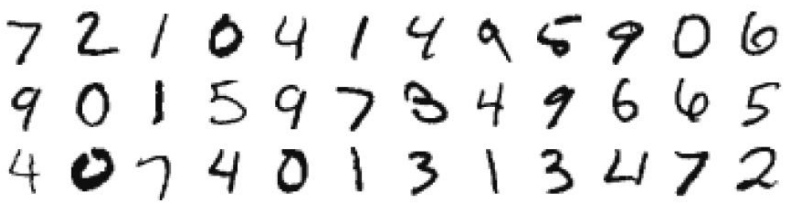



设有 activation 时，就是 PCA

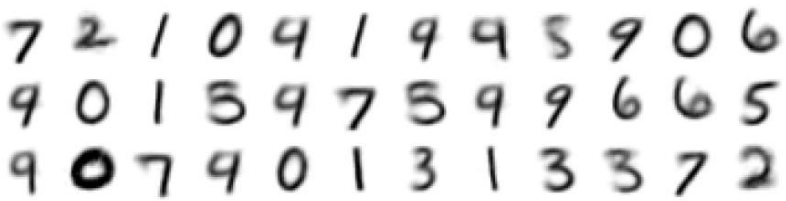
A simple example of such an autoencoder would be with both f and g linear, in which case the optimal solution is given by PCA.

# Autoencoders

$X$ (original samples) $d = 1024$

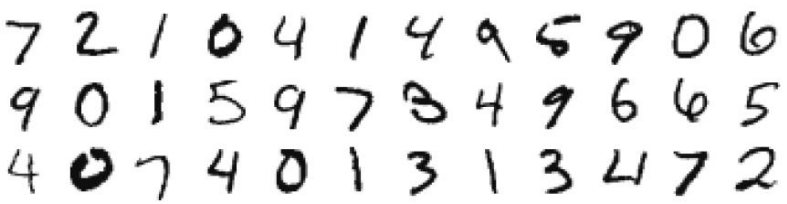


$g \circ f(X)$ (CNN, $d = 2$)

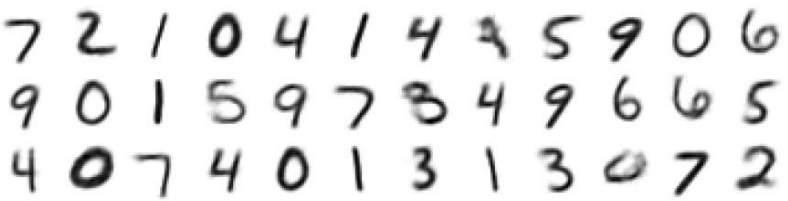


$g \circ f(X)$ (PCA, $d = 2$)

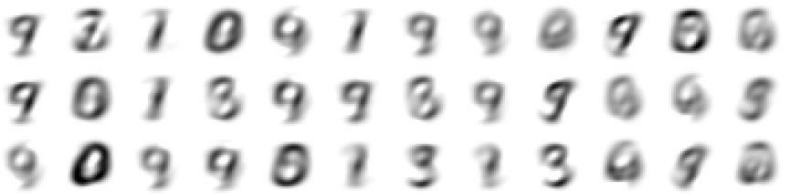
$X$ (original samples) $d = 1024$



$g \circ f(X)$ (CNN, $d = 4$)

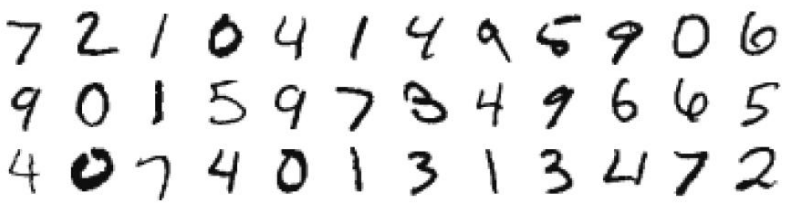


$g \circ f(X)$ (PCA, $d = 4$)

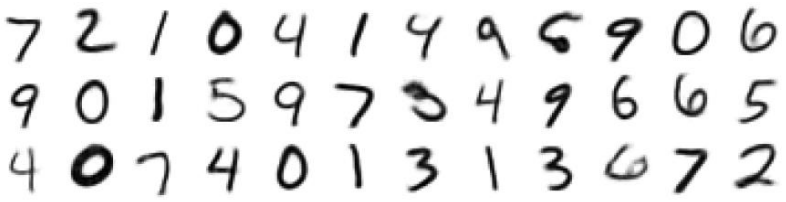


# Autoencoders

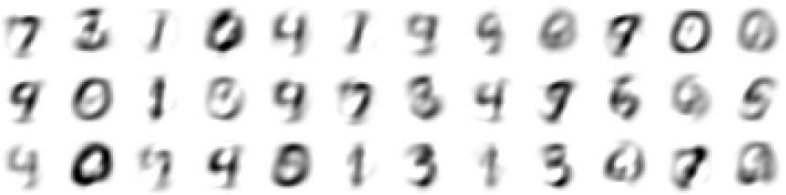
$X$ (original samples) $d = 1024$



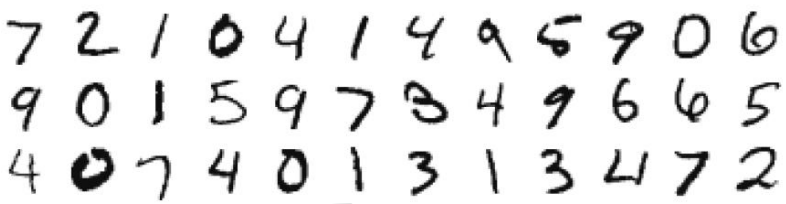
$g \circ f(X)$ (CNN, $d = 8$)



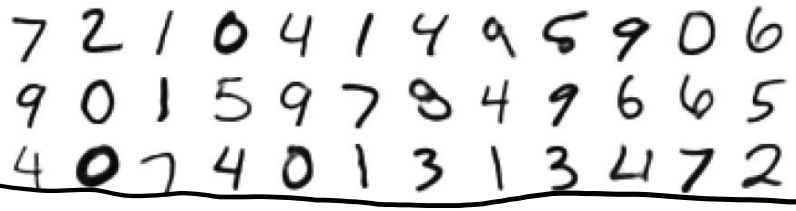
$g \circ f(X)$ (PCA, $d = 8$)



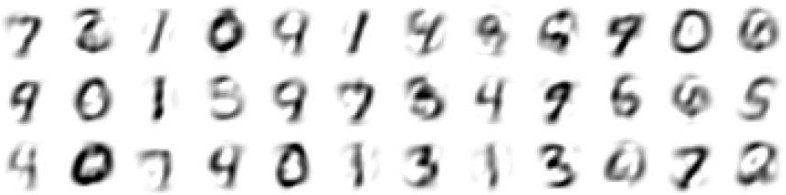
$X$ (original samples) $d = 1024$



$g \circ f(X)$ (CNN, $d = 16$)



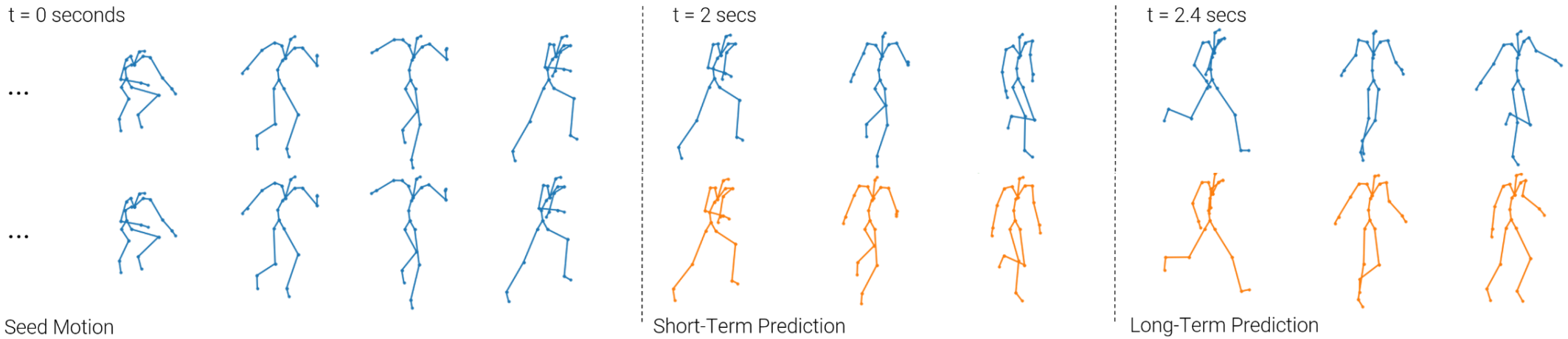
$g \circ f(X)$ (PCA, $d = 16$)



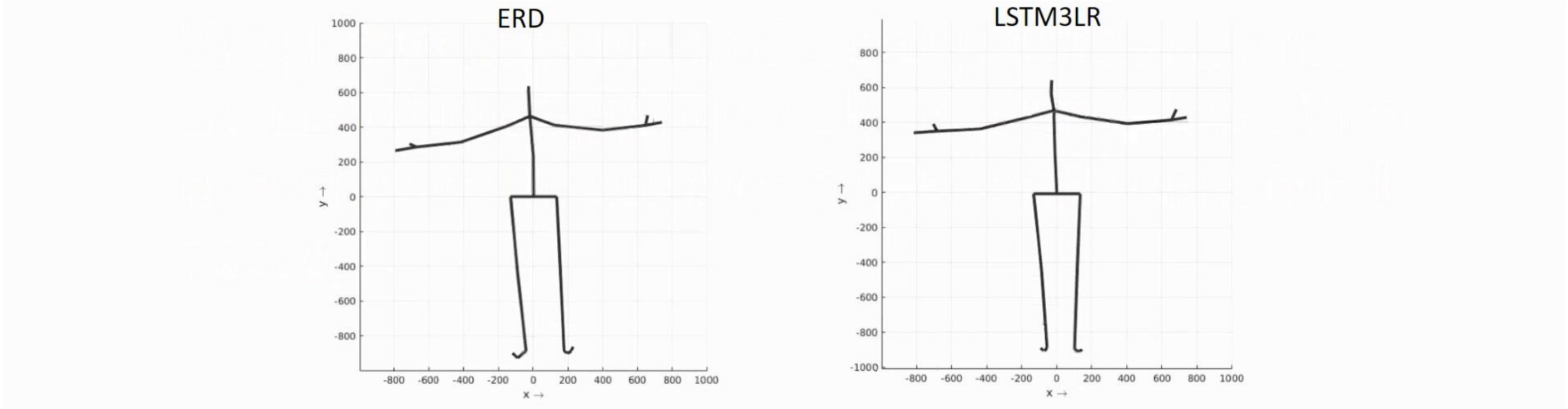
# Example:
# 3D Human Motion Modelling

[Partha Ghosh et al. 3DV 2017]

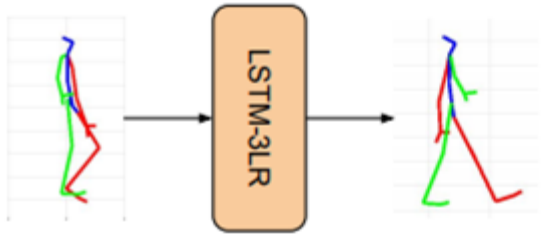
# Problem

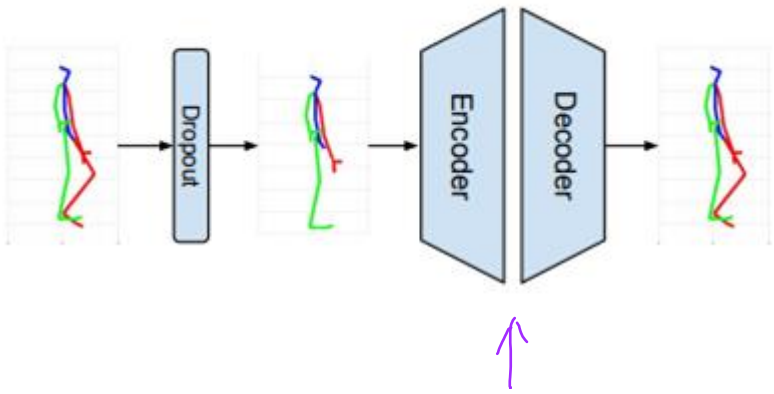

t = 0 seconds
...
...
Seed Motion
t = 2 secs
Short-Term Prediction
t = 2.4 secs
Long-Term Prediction

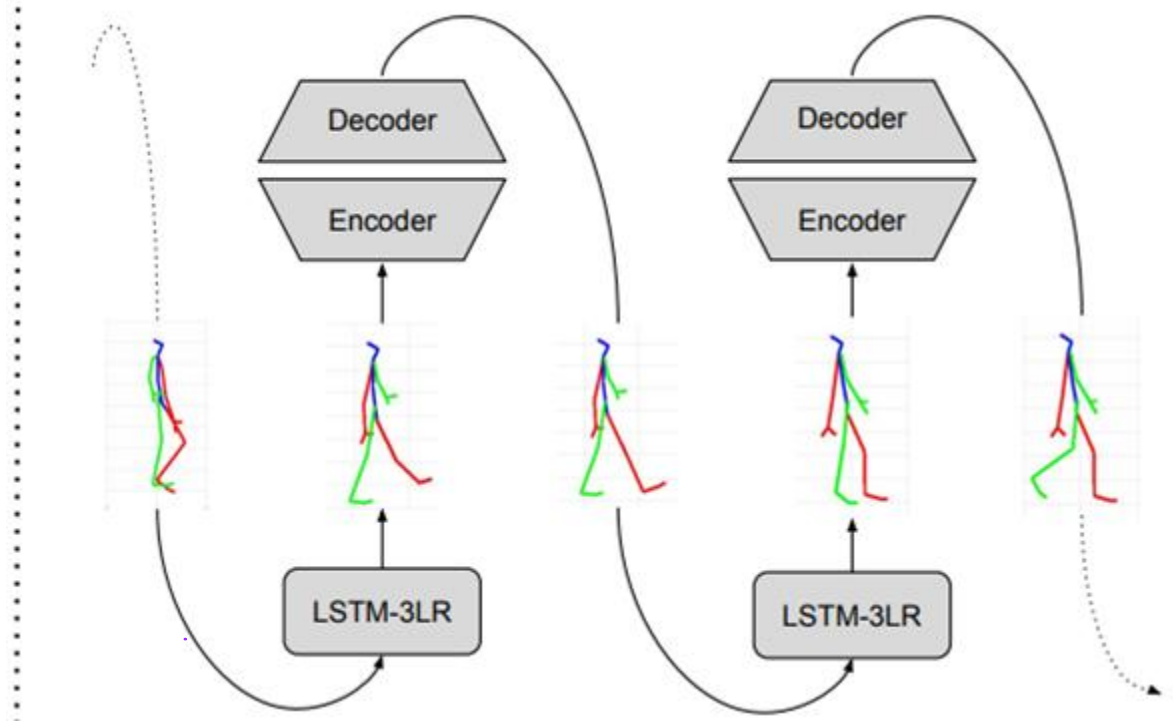
# Qualitative comparison



# Learning human motion models for long-term predictions

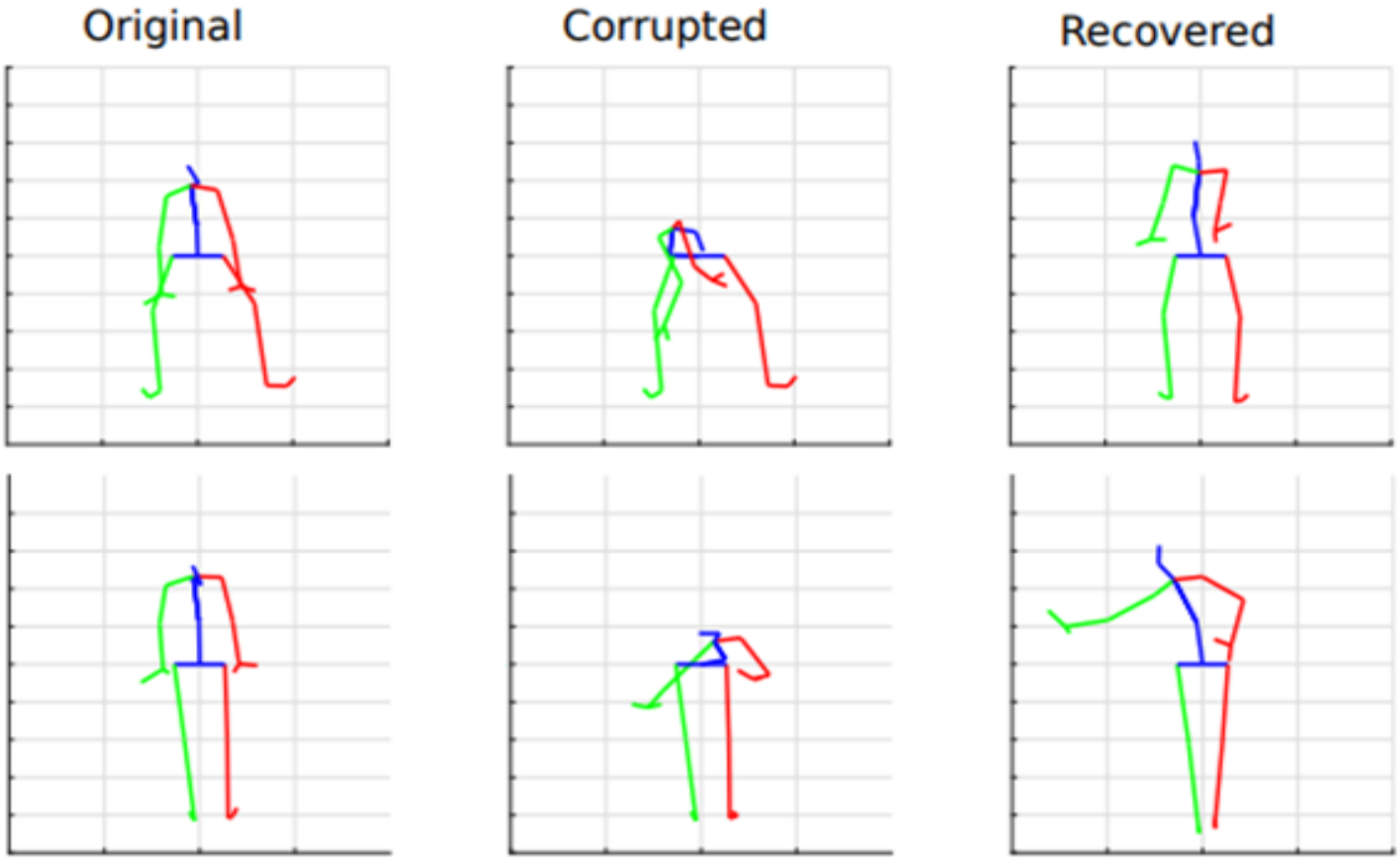


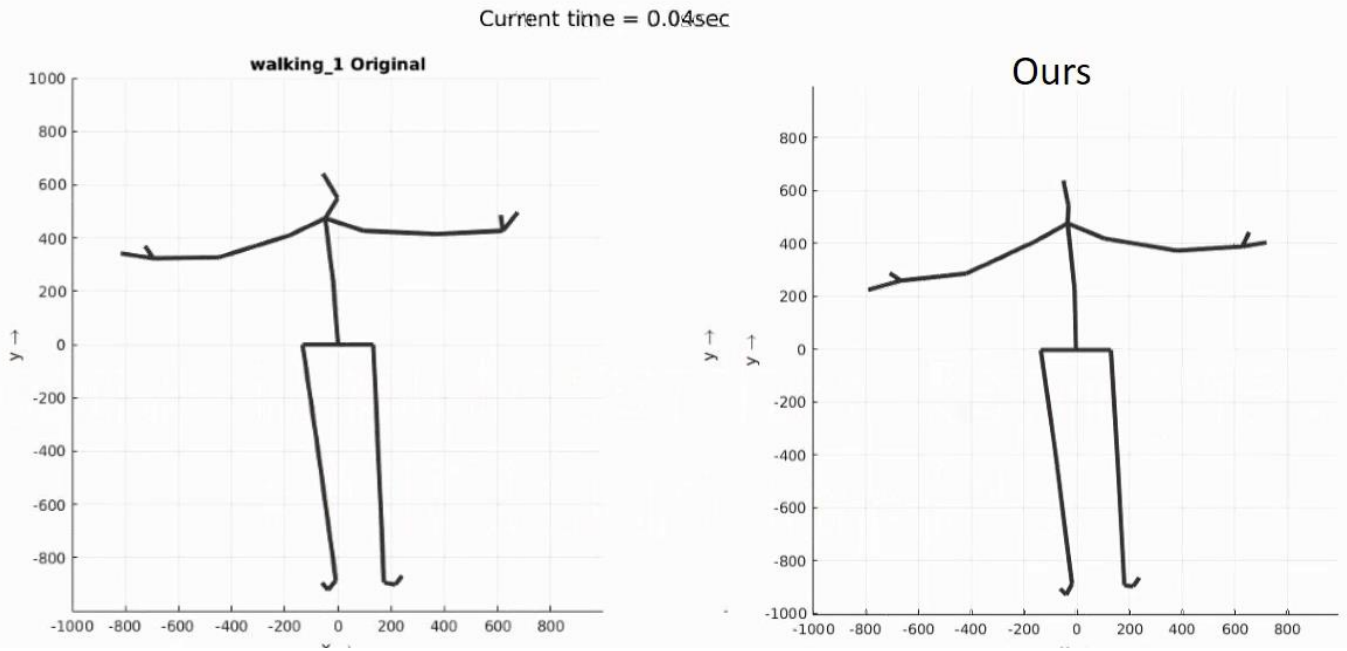

In this case, latent space is in higher-d.

# The auto-encoder's role



# Qualitative comparison

Grey = Seed Sequence
Colored = Predicted



# The Generative Story: Autoencoders

By introducing a simple density model (Gaussian) over the latent space $\mathrm{Z}$ the decoder $\mathrm{g}$ can be used to *generate* new data (e.g., images).

$$f(x) \sim N(\hat{\mu}, \hat{\sigma} I)$$

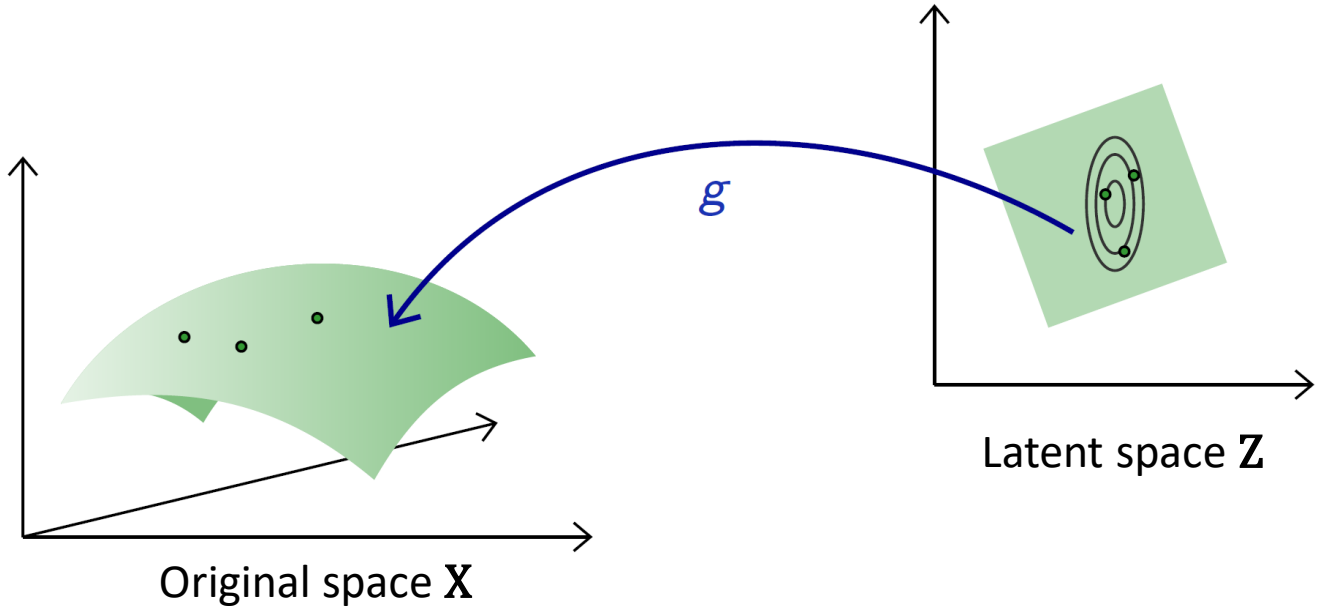
where $\hat{\mu}$ and $\hat{\sigma}$ are mean and standard deviation estimated on training data.

# Autoencoders: Generative Capabilities

By introducing a simple density model (Gaussian) over the latent space $\mathrm{Z}$ the decoder $\mathrm{g}$ can be used to *generate* new data (e.g., images).

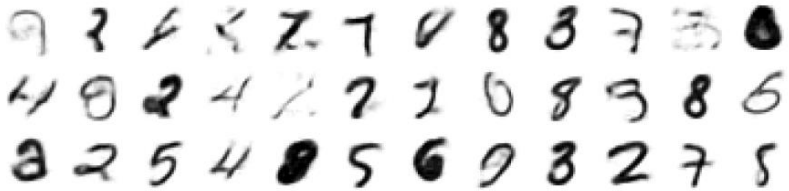
$$f(x) \sim N(\hat{\mu}, \hat{\sigma} I)$$

where $\hat{\mu}$ and $\hat{\sigma}$ are mean and standard deviation estimated from training data.

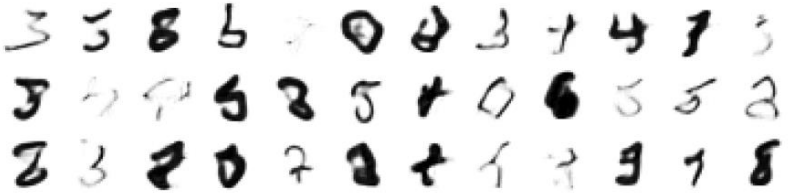


# Autoencoders

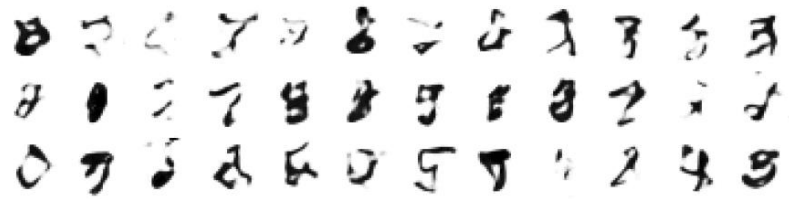
Autoencoder sampling ($d = 8$)



Autoencoder sampling ($d = 16$)

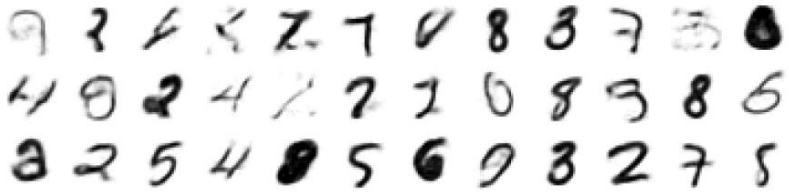


Autoencoder sampling ($d = 32$)

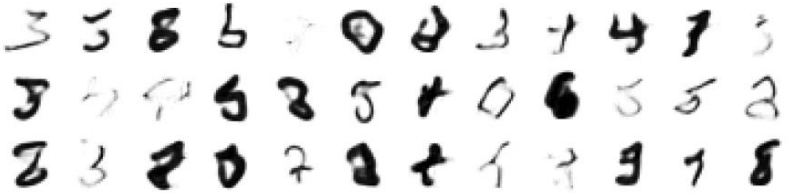


# Autoencoders

Autoencoder sampling ($d = 8$)



Autoencoder sampling ($d = 16$)



Autoencoder sampling ($d = 32$)

The decoder is not able to generate good quality samples because the density model on the latent space Z is too simple and inadequate.

# Autoencoder: Visualizing MNIST Embedding

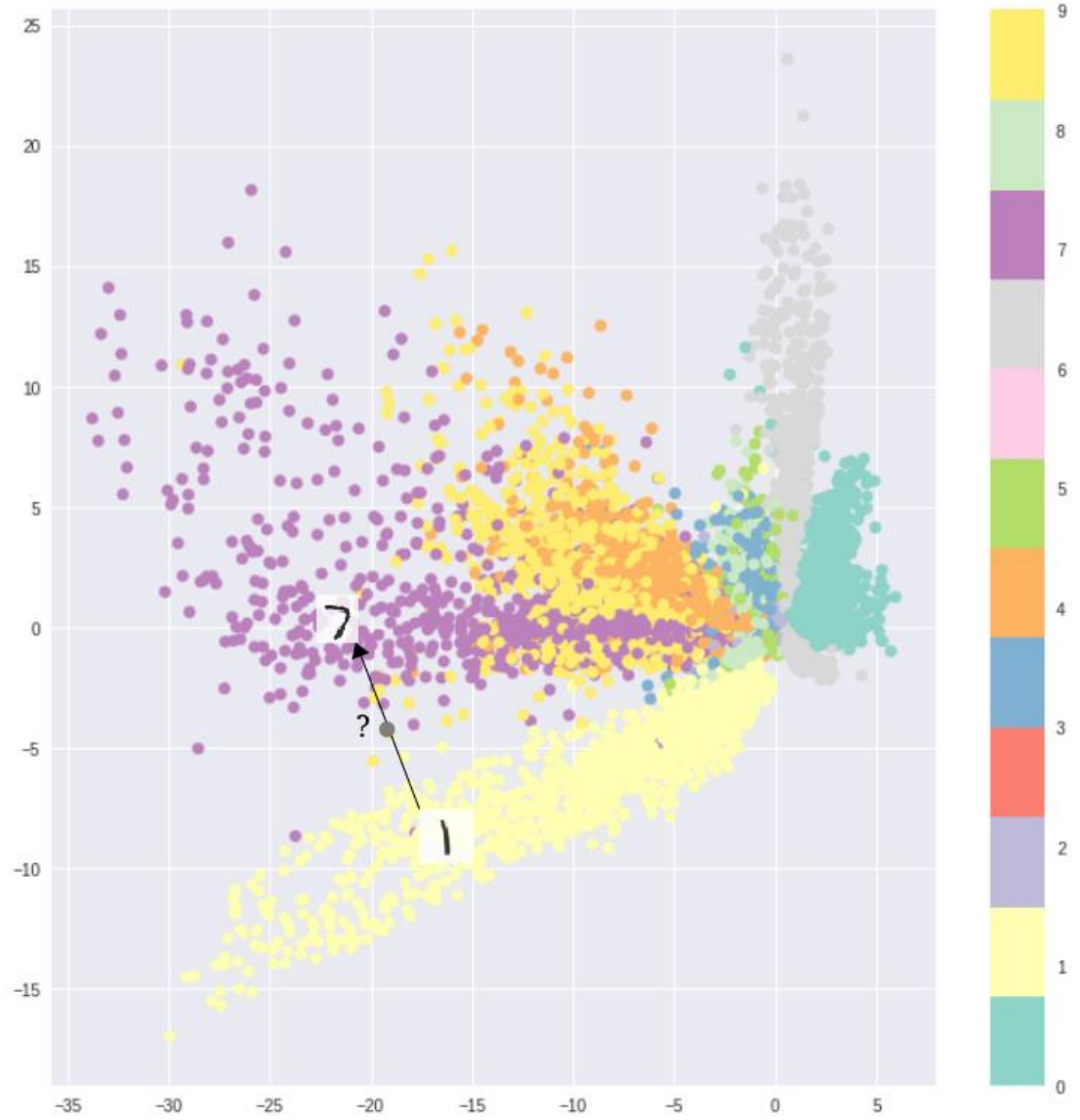



Image source: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

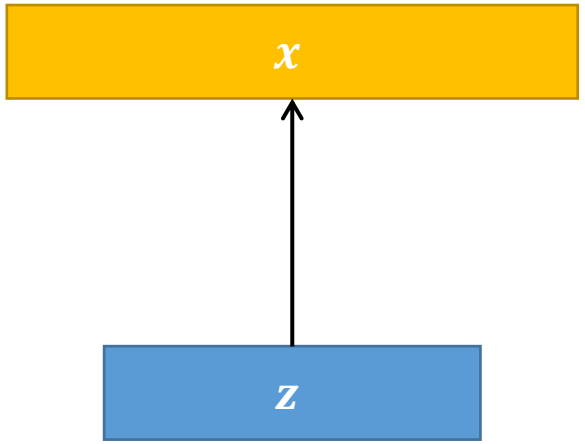
# Variational autoencoders

Probabilistic version of autoencoders:

Allows for sampling from the learned model to generate data with natural variation

Assume training data $\mathbb{D}=\{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ is generated from underlying latent distribution $z$

Sample from conditional $p_{\Theta^*}(x|z^{(i)})$




Sample from prior $p_{\Theta^*}(z)$

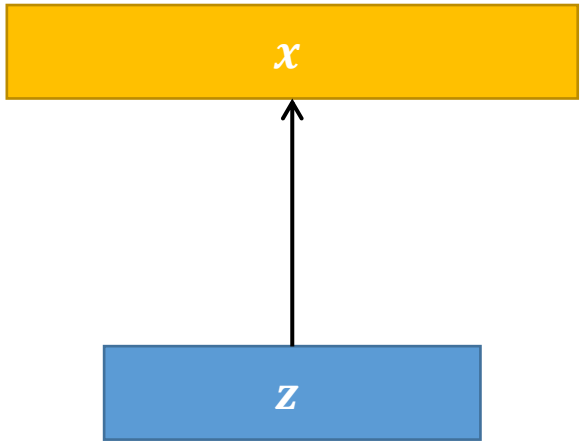
Intuition:
$\boldsymbol{x}$ is the desired output (e.g. an image),
$\boldsymbol{z}$ are the latent factors that control generation process of $\boldsymbol{x}$:
orientation, scale, object properties, etc.

# Variational autoencoders: Modelling

We want to capture this process via estimation of the parameters $\Theta^*$ of this generative model.

How to best represent prior and conditional?

Sample from conditional $p_{\Theta^*}(x|z^{(i)})$




Sample from prior $p_{\Theta^*}(z)$

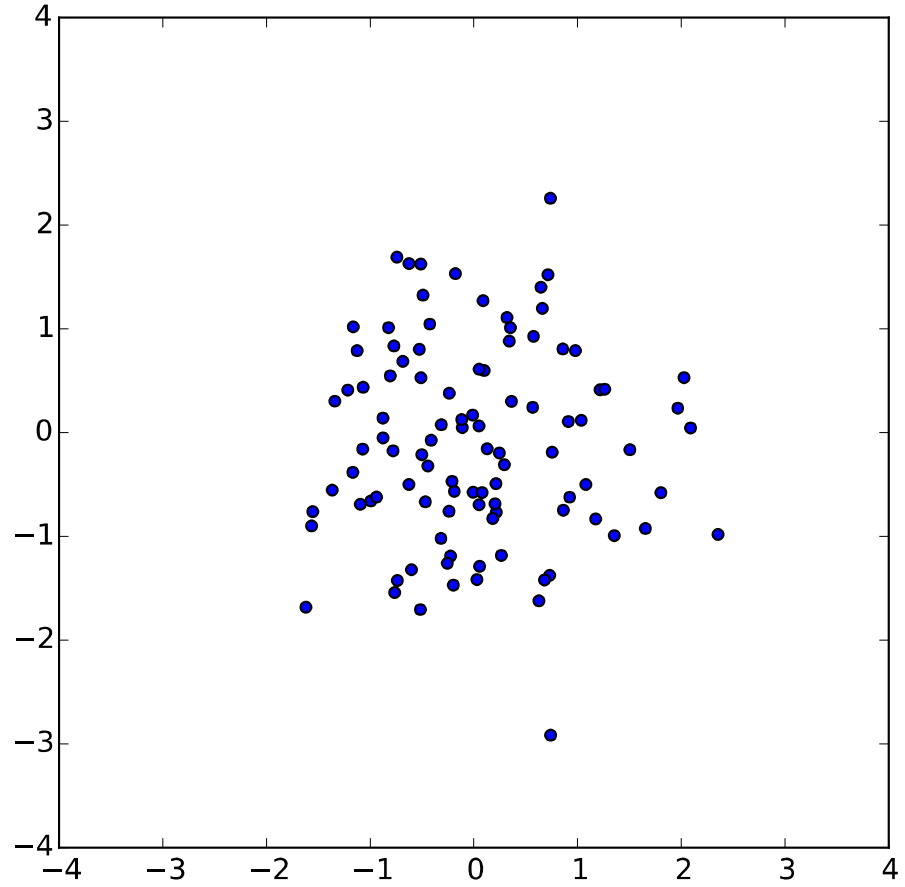
Conditional $p(\boldsymbol{x}|\boldsymbol{z})$ is generally *complex* since it needs to generate outputs (e.g., images)
=> parametrize via neural network

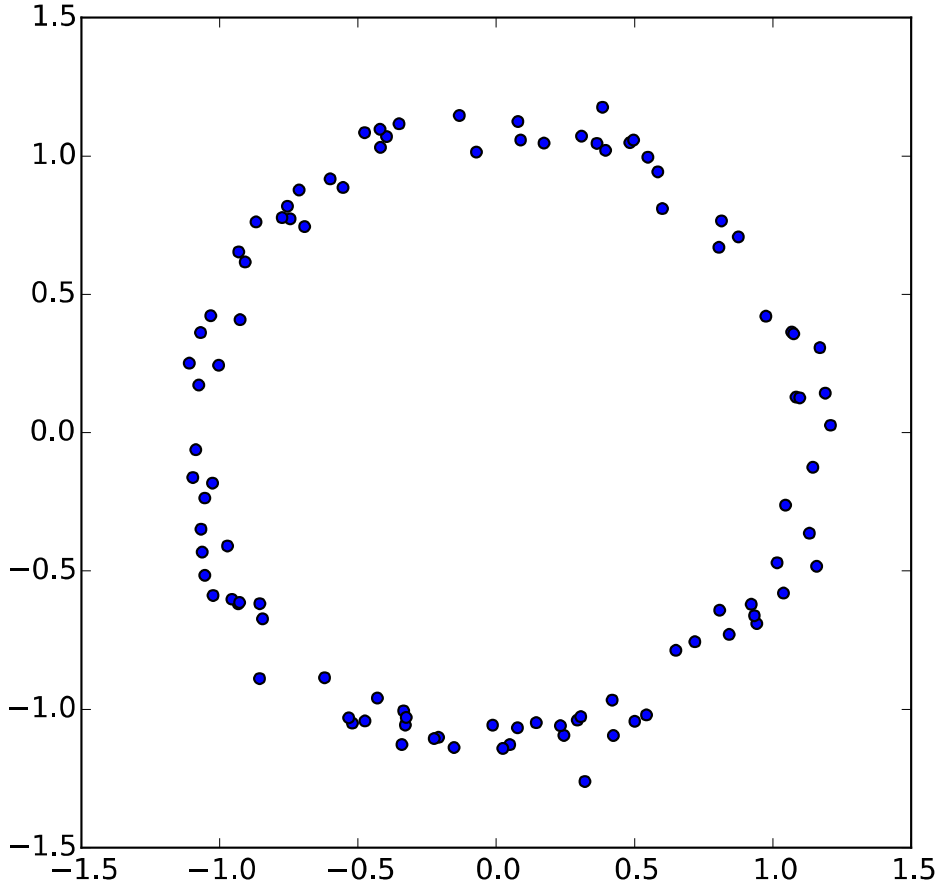
Choose prior $p(\boldsymbol{z})$ to be a *simple* distribution.
For example, a Gaussian.

# Re-Mapping of Random Variables

$z$

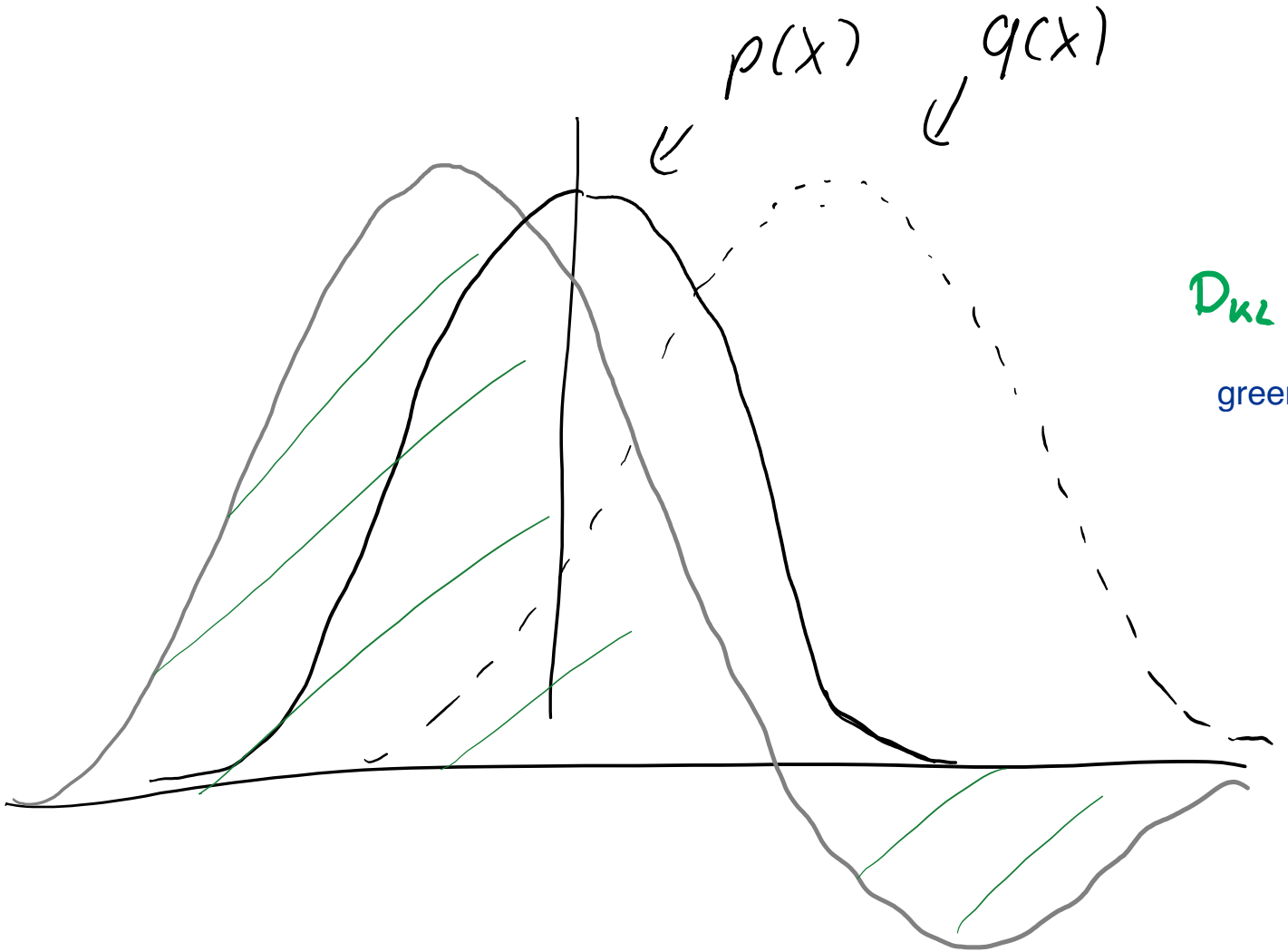
$$g(z) = \frac{z}{10} + \frac{z}{\|z\|}$$



# Kullback-Leibler divergence

measure dissimilarity between 2 distribution




$$D_{KL}(P||Q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx$$

green part

# Non-negativity (Proof)

We show that $-D_{KL} \leq 0$:

$$-D_{KL} = -\int_x p(x) \log \frac{p(x)}{q(x)} \, dx$$

$$= \int_x p(x) \log \frac{q(x)}{p(x)} \, dx$$

$$\overset{*}{\leq} \log \int_x p(x) \frac{q(x)}{p(x)} \, dx$$

$$= \log \int_x q(x) \, dx$$

$$= \log 1 = 0 \quad \blacksquare$$

* Jensen's inequality

$$\mathbb{E}(\varphi(x)) \leq \varphi(\mathbb{E}(x))$$

iff $\varphi$ concave

log concave?

$D' = \frac{1}{x}$ $\frac{1}{x}$

$D'' = -\frac{1}{x^2} \Rightarrow$ strictly decreasing for all $x \neq 0$ $-\frac{1}{x^2}$

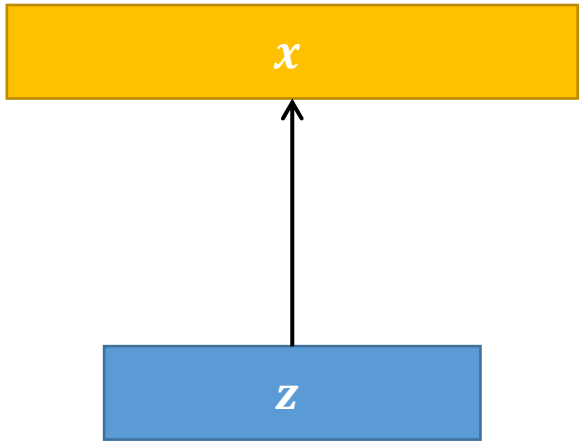
$\Rightarrow$ concave!

# Variational autoencoders: Training

We want to capture this process via estimation of the parameters $\Theta^*$ of this generative model.

How to find parameters $\Theta$?

Sample from conditional
$p_{\Theta^*}(x|z^{(i)})$




Sample from prior
$p_{\Theta^*}(z)$

$$p_\Theta(\boldsymbol{x}) = \int_z p_\Theta(\boldsymbol{x}|\boldsymbol{z}) p_\Theta(\boldsymbol{z}) dz$$

Learn model parameters that maximize likelihood of the data

Problem?

# Training VAEs: Intractability

Data likelihood: $p_\Theta(\boldsymbol{x}) = \int_z p_\Theta(\boldsymbol{x}|\boldsymbol{z}) p_\Theta(\boldsymbol{z}) d\boldsymbol{z}$

Z是infinite 所以不可积

Integral over continuous random variable

Decoder (NN)

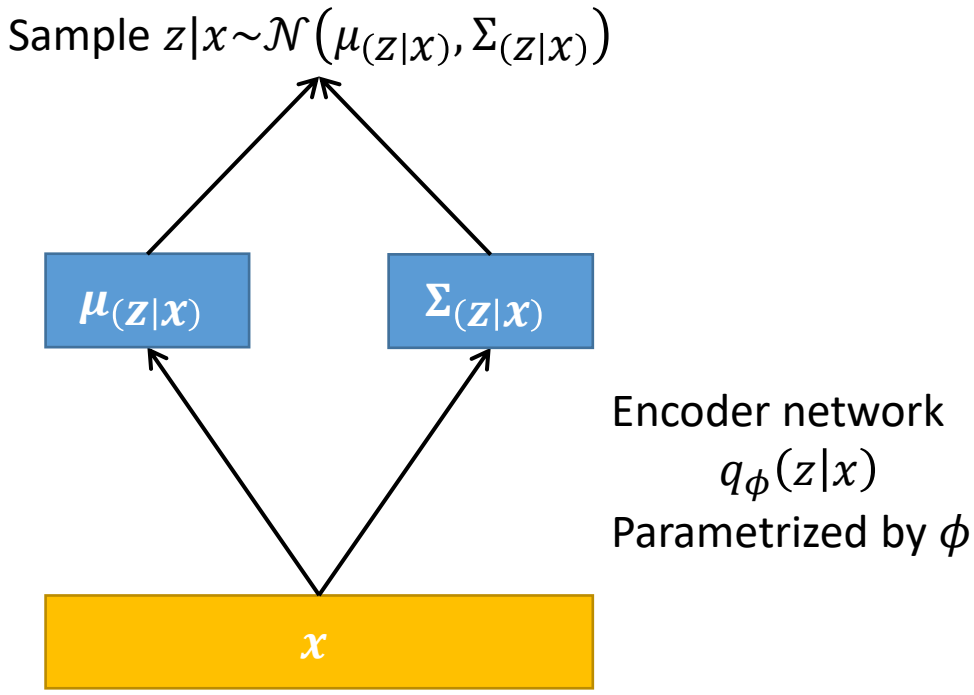
Isotropic normal distribution

Posterior density also intractable: $p_\Theta(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_\Theta(\boldsymbol{x}|\boldsymbol{z}) p_\Theta(\boldsymbol{z})}{p_\Theta(\boldsymbol{x})}$

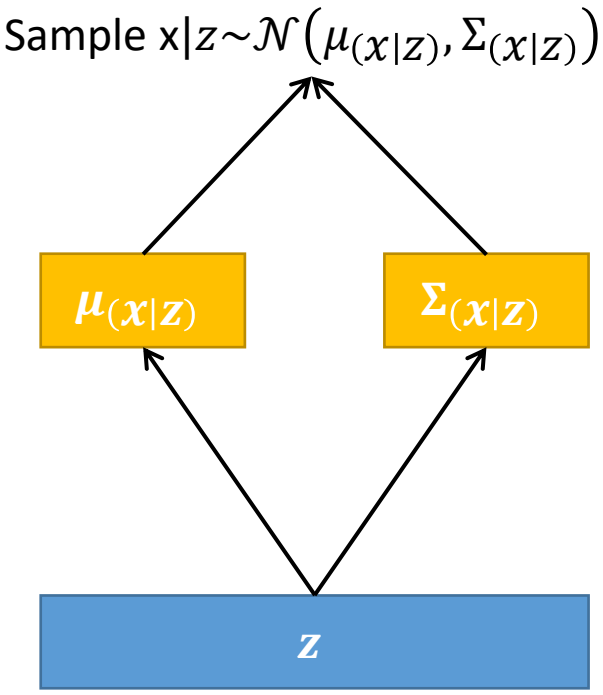
Intractable data likelihood

Solution: define additional encoder network $q_\Theta(\boldsymbol{z}|\boldsymbol{x})$ to approximate $p_\Theta(\boldsymbol{z}|\boldsymbol{x})$

# VAEs: Encoder & Decoder Networks




Encoder network
$q_\phi(z|x)$
Parametrized by $\phi$




Decoder network
$p_\Theta(x|z)$
Parametrized by $\Theta$

# Data log-likelihood

$$\log p_\theta(x^{(i)}) =$$

$$= \mathbb{E}_{z \sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(\cdot) \text{ doesn't depend on } z)$$

$$= \mathbb{E}_z\left[\log \frac{p_\theta(x^{(i)}|z)\,p_\theta(z)}{p_\theta(z|x^{(i)})}\right] \quad (\text{Bayes' rule})$$

$$= \mathbb{E}_z\left[\log \frac{p_\theta(x^{(i)}|z)\,p_\theta(z)}{p_\theta(z|x^{(i)})}\,\frac{q_\phi(z|x^{(i)})}{q_\phi(z|x^{(i)})}\right] \quad (\text{mult by constant})$$

$$= \mathbb{E}_z\left[\log p_\theta(x^{(i)}|z)\right] + \mathbb{E}_z\left[\log \frac{p_\theta(z)}{q_\phi(z|x^{(i)})}\right] + \mathbb{E}_z\left[\log \frac{q_\phi(z|x^{(i)})}{p_\theta(z|x^{(i)})}\right] \quad (\text{product rule logarithms})$$

(continued on next slide)

$$= E_z[\log p_\theta(x^{(i)}|z)] - E_z\left[\log \frac{q_\phi(z|x^{(i)})}{p_\theta(z)}\right] + E_z\left[\log \frac{q_\phi(z|x^{(i)})}{p_\theta(z|x^{(i)})}\right] \quad \left(\log \frac{x}{y} = -\log \frac{y}{x}\right)$$

✓ (with sampling) ✓ closed-form if Gaussian ✓

$$= E_z[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)}) \| p_\theta(z)) + D_{KL}(q_\phi(z|x^{(i)}) \| p_\theta(z|x^{(i)}))$$

$\log p_\theta(x^{(i)}|z)$: ↑ NN, tractable

$q_\phi(z|x^{(i)})$: ↑ tractable

$p_\theta(z)$: ↓ "make approx. posterior as similar as possible to prior"

$D_{KL}(q_\phi(z|x^{(i)}) \| p_\theta(z|x^{(i)}))$: ✗ $\geq 0$ (see proof earlier)

目的是: opt. only tractable term: $q_\phi$, $p_\theta$

不想要处理 $p(x)$, $p(z|x)$

‖ reconstruction ‖

goes high if our sample is similar to true distribution

loss function $\mathcal{L}(x^{(i)}; \theta, \phi)$

$$\log(p_\theta(x^{(i)})) \geq \mathcal{L}$$

evidence lower bound "ELBO"

⇒ data is at least as likely as $\mathcal{L}$

Traing procedure:

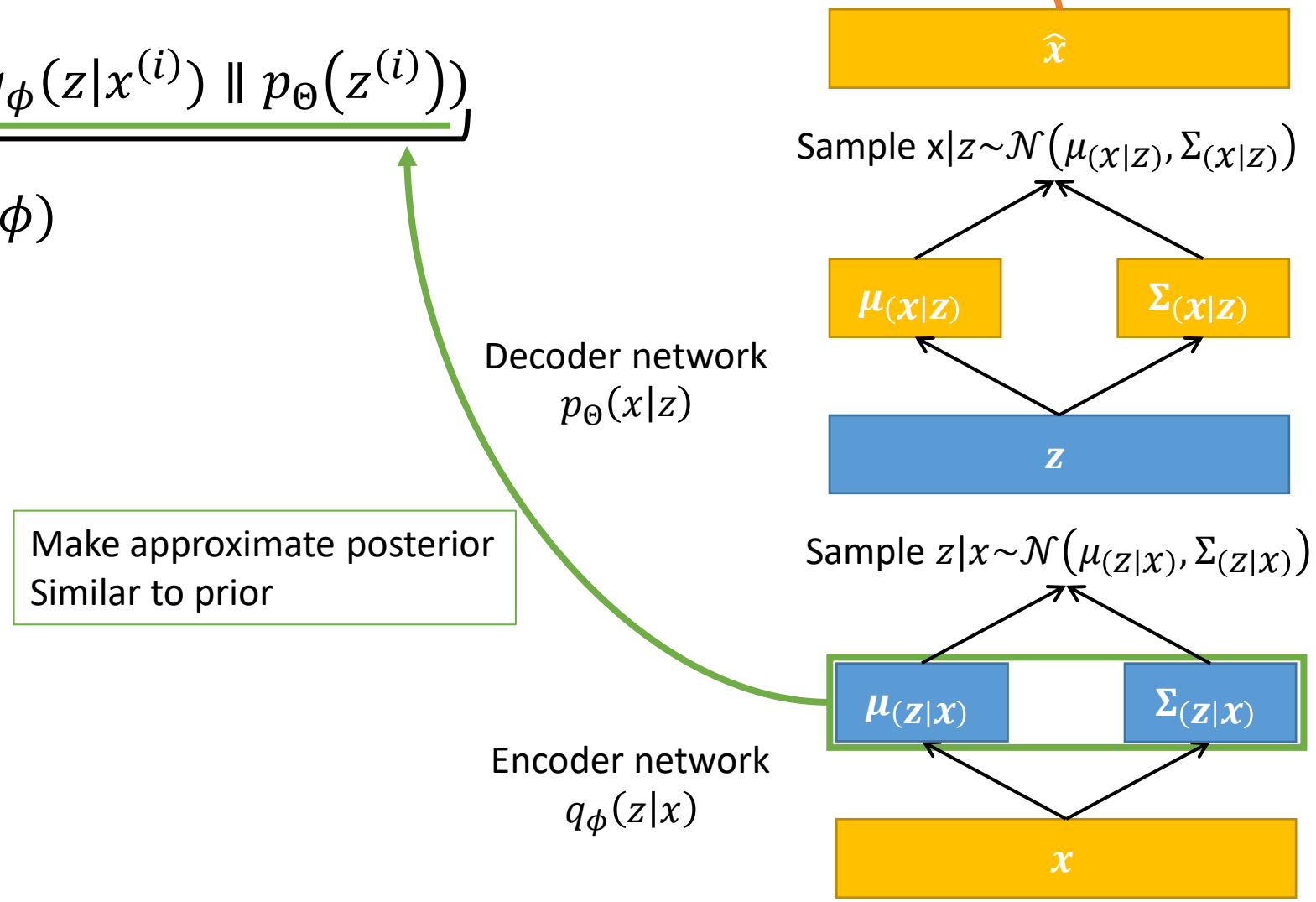
$$\theta^*, \phi^* = \text{argmax} \sum_1^N \mathcal{L}(x^{(i)}; \theta, \phi)$$

# Putting it all together (Forward-pass)

$$\underbrace{\mathrm{E}_z\left[\log p_\Theta\left(x^{(i)} \middle| z\right)\right] - D_{KL}\left(q_\phi(z|x^{(i)}) \parallel p_\Theta\left(z^{(i)}\right)\right)}_{\mathcal{L}(x^{(i)}, \Theta, \phi)}$$

Evidence lower bound

Maximize reconstruction likelihood



# Training VAEs

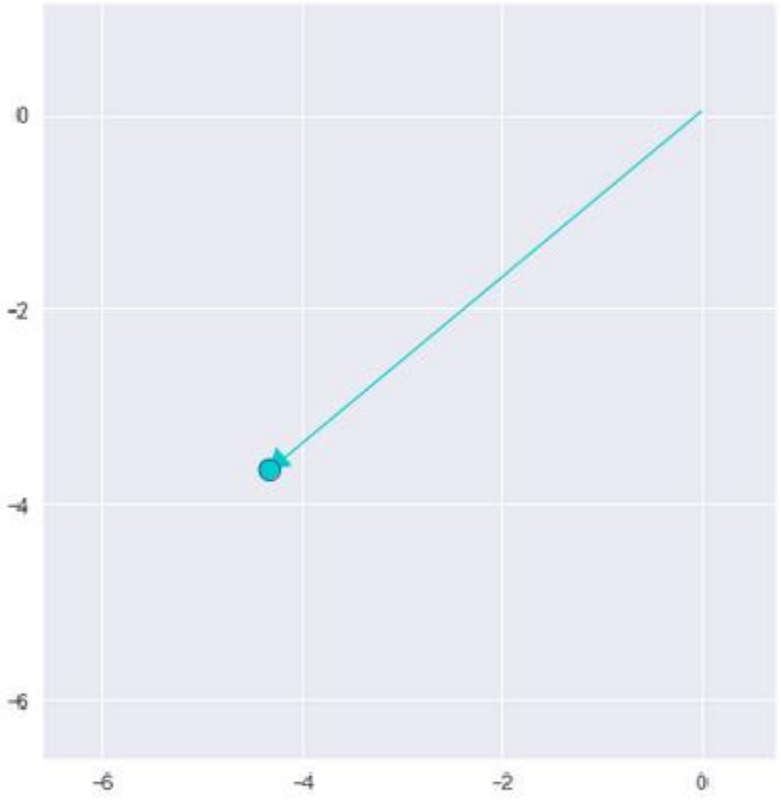
Training of VAEs is just backprop.

But wait, how do gradients flow through $z$?

Or any other random operation (requiring sampling)?

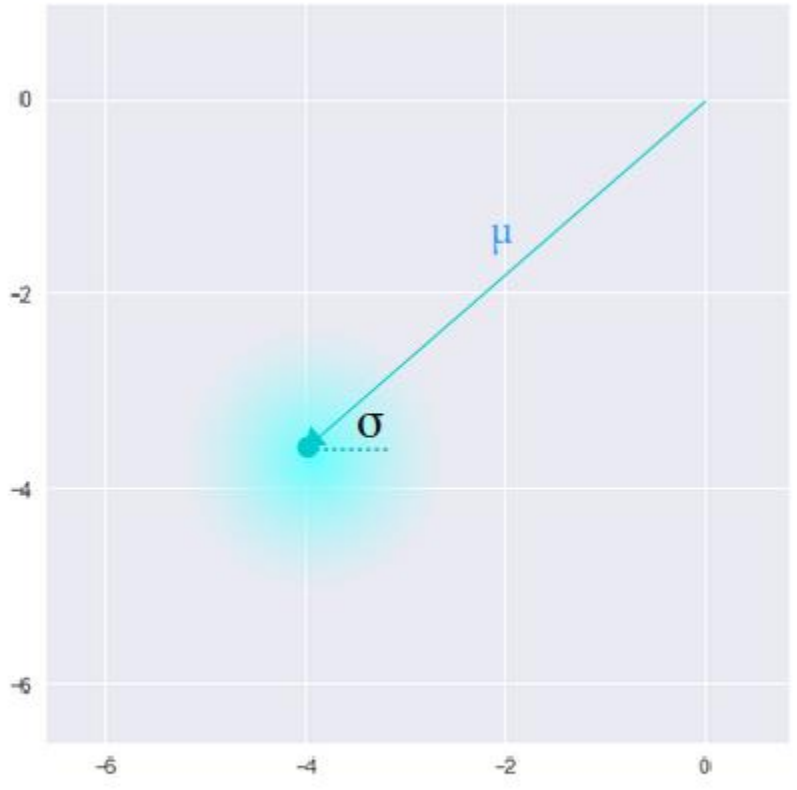
(see blackboard)

# Contrasting Autoencoders and VAEs




Standard Autoencoder
(direct encoding coordinates)

$x \rightarrow z$




Variational Autoencoder
(μ and σ initialize a probability distribution)

predict $\mu$, $\sigma$ of latent space

Image source: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

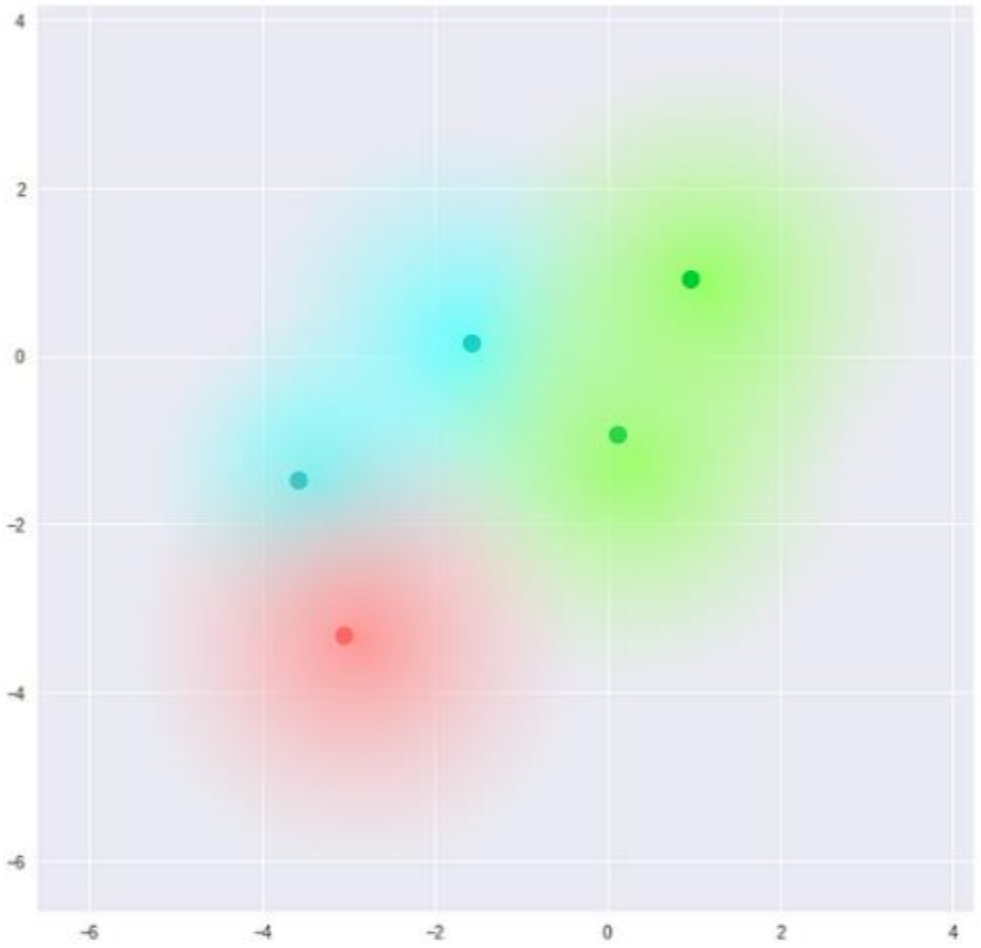
# Effect of sampling

density packed

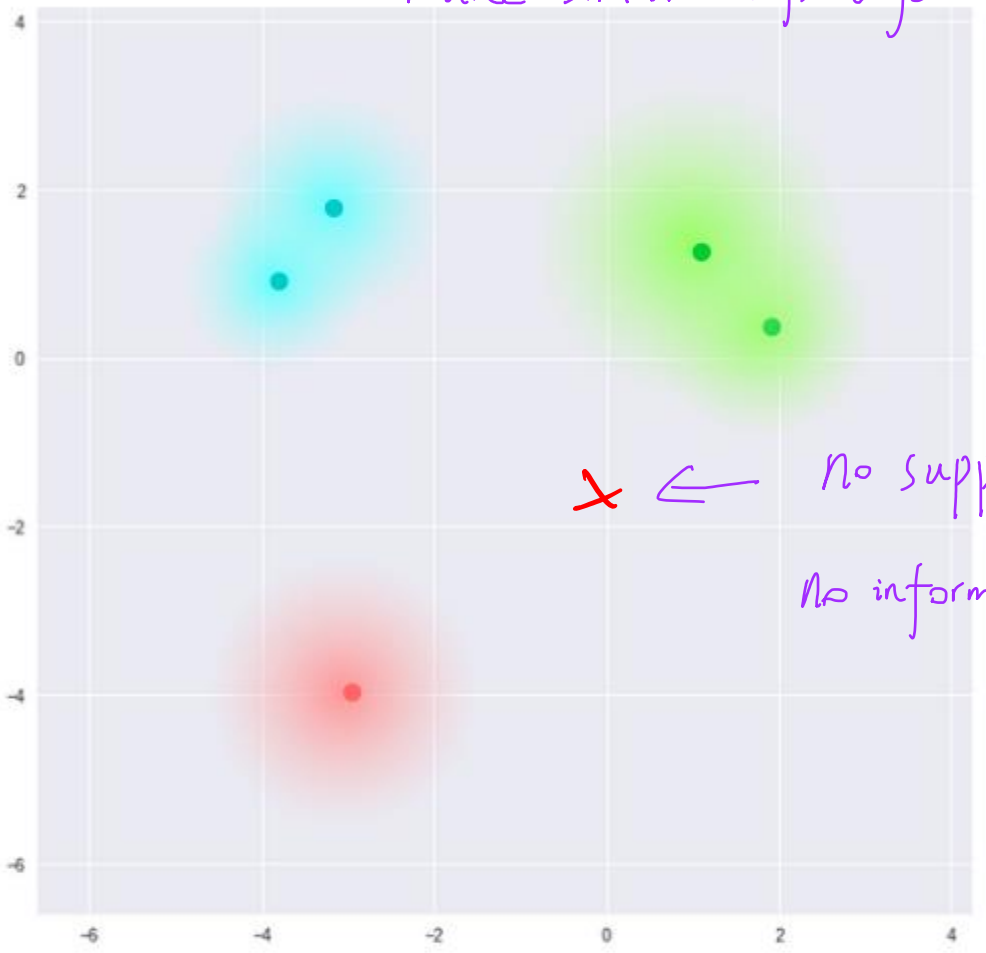
only construction term $\log p(x|z)$

reconstruction loss will push cluster apart

make similar things together




What we require




x ← no support

No information

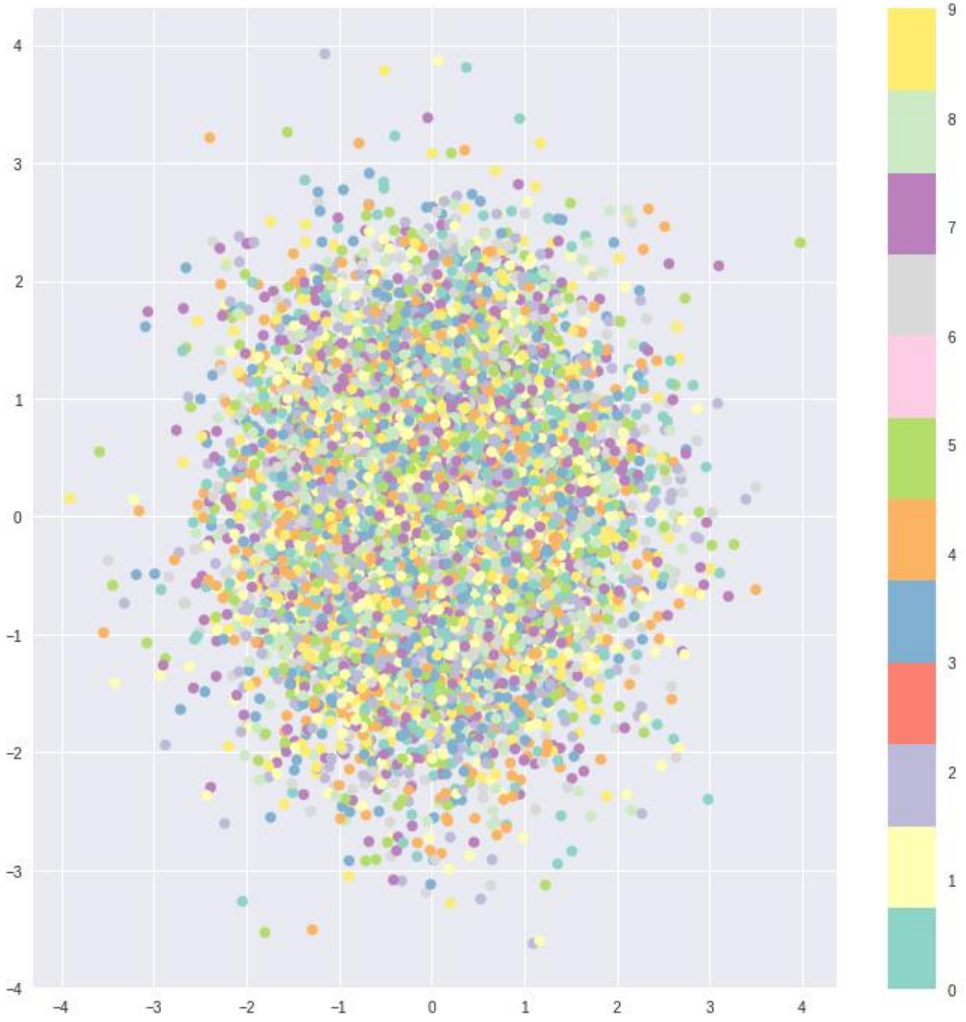
What we may inadvertently end up with

我们可能会无意中得到的结果

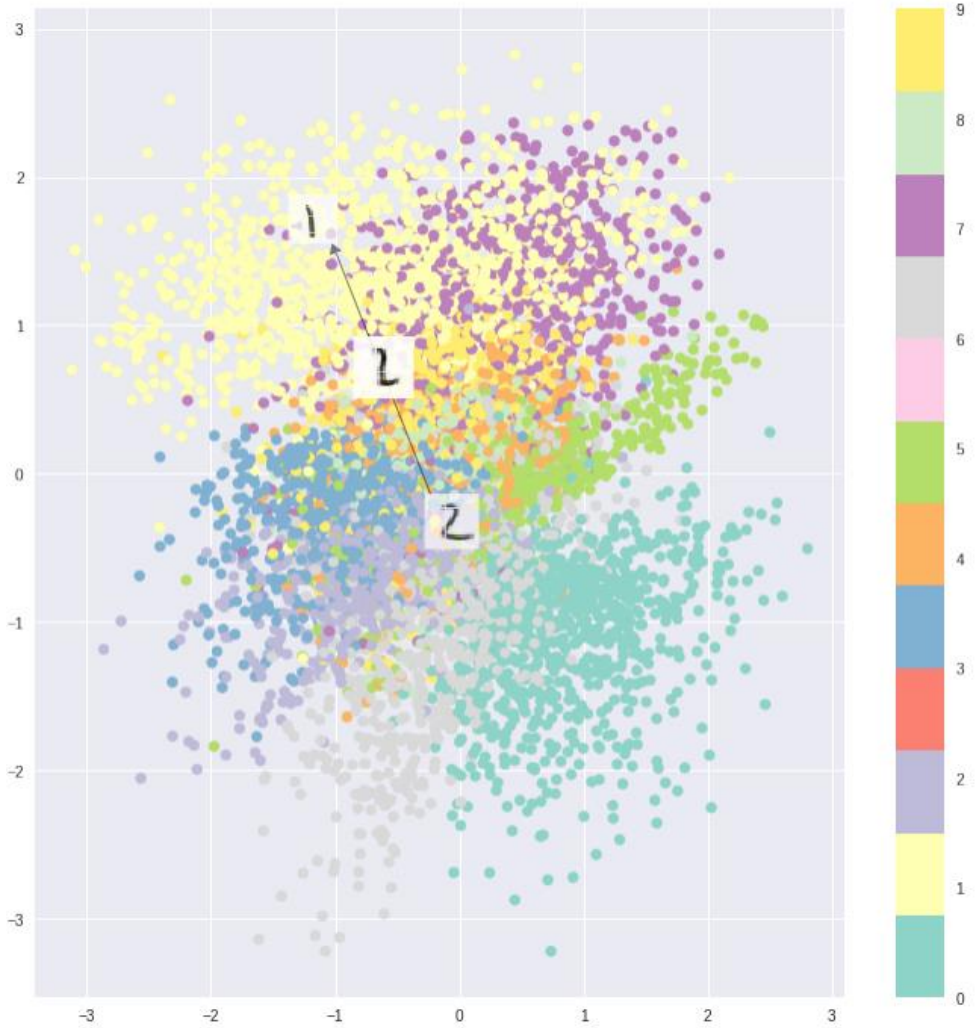
Image source: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

# VAE loss

have support, draw sample everywhere



$D_{KL}(q_\phi(z|x^{(i)}) \parallel p_\Theta(z^{(i)}))$




$\mathrm{E}_z[\log p_\Theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)}) \parallel p_\Theta(z^{(i)}))$

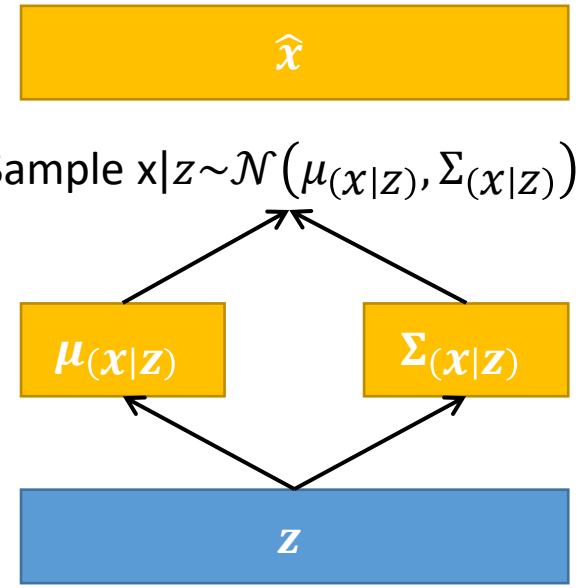
Image source: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

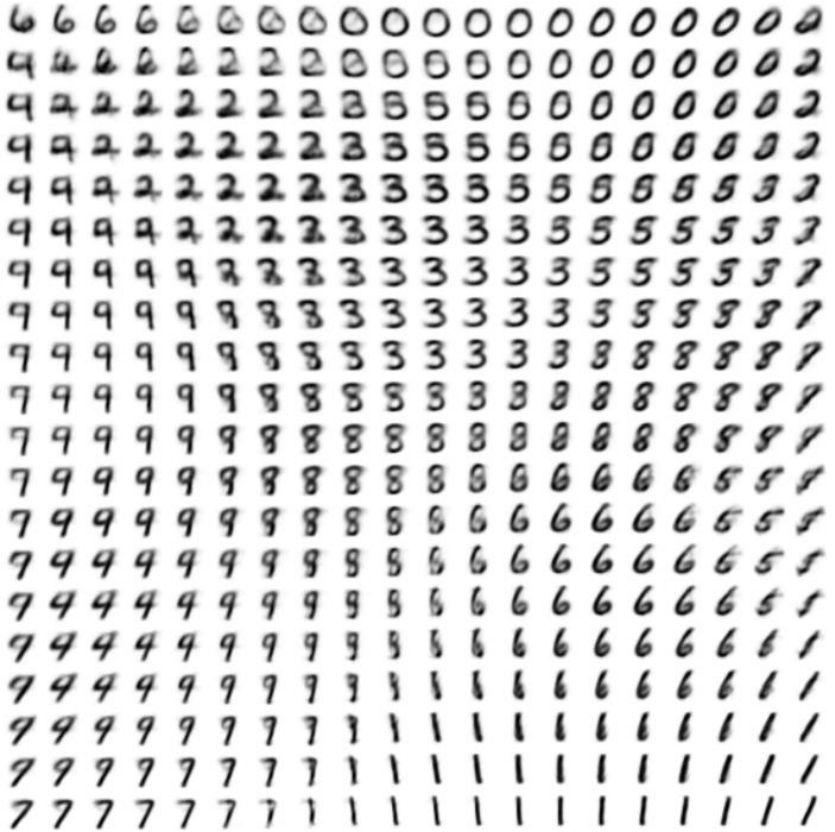
# Stopped here 9 April 2020

# Generating Data

At runtime only use decoder network. Sample z from prior.

Decoder network
$p_\Theta(x|z)$

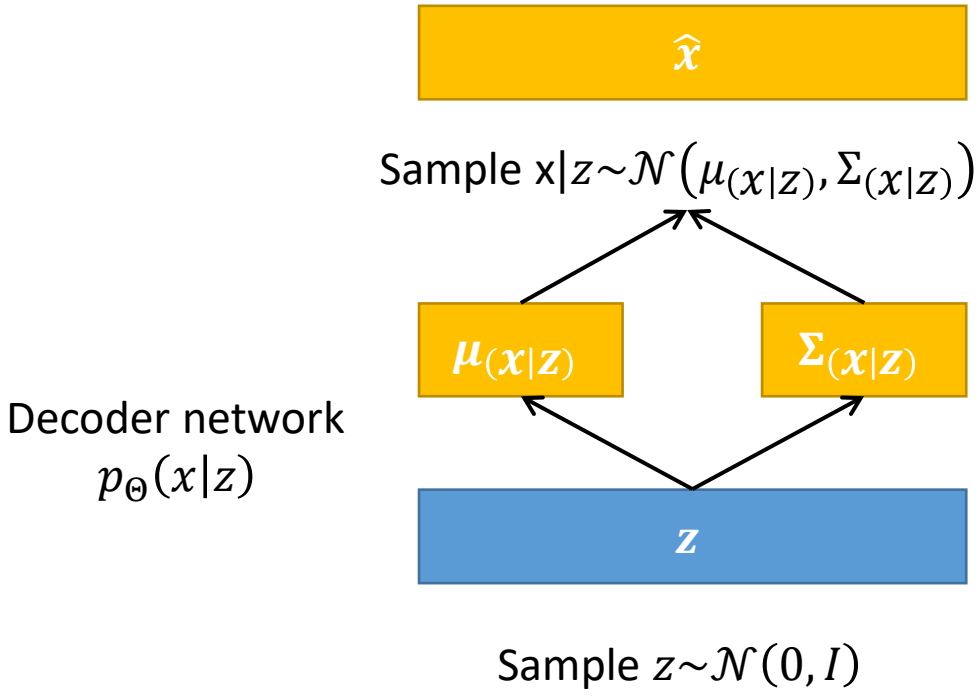


[Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014]

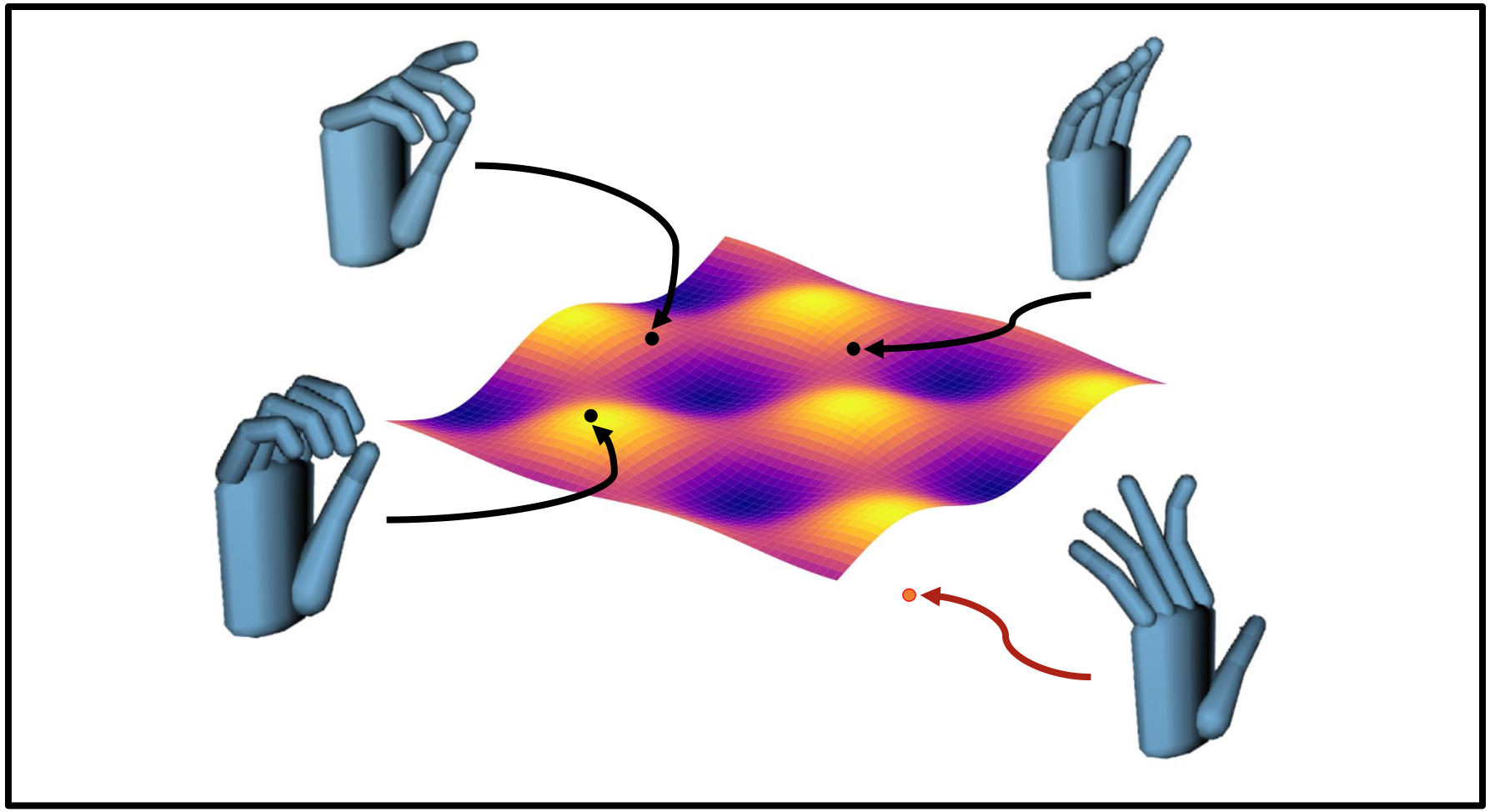
# Generating Data

At runtime only use decoder network. Sample z from prior.



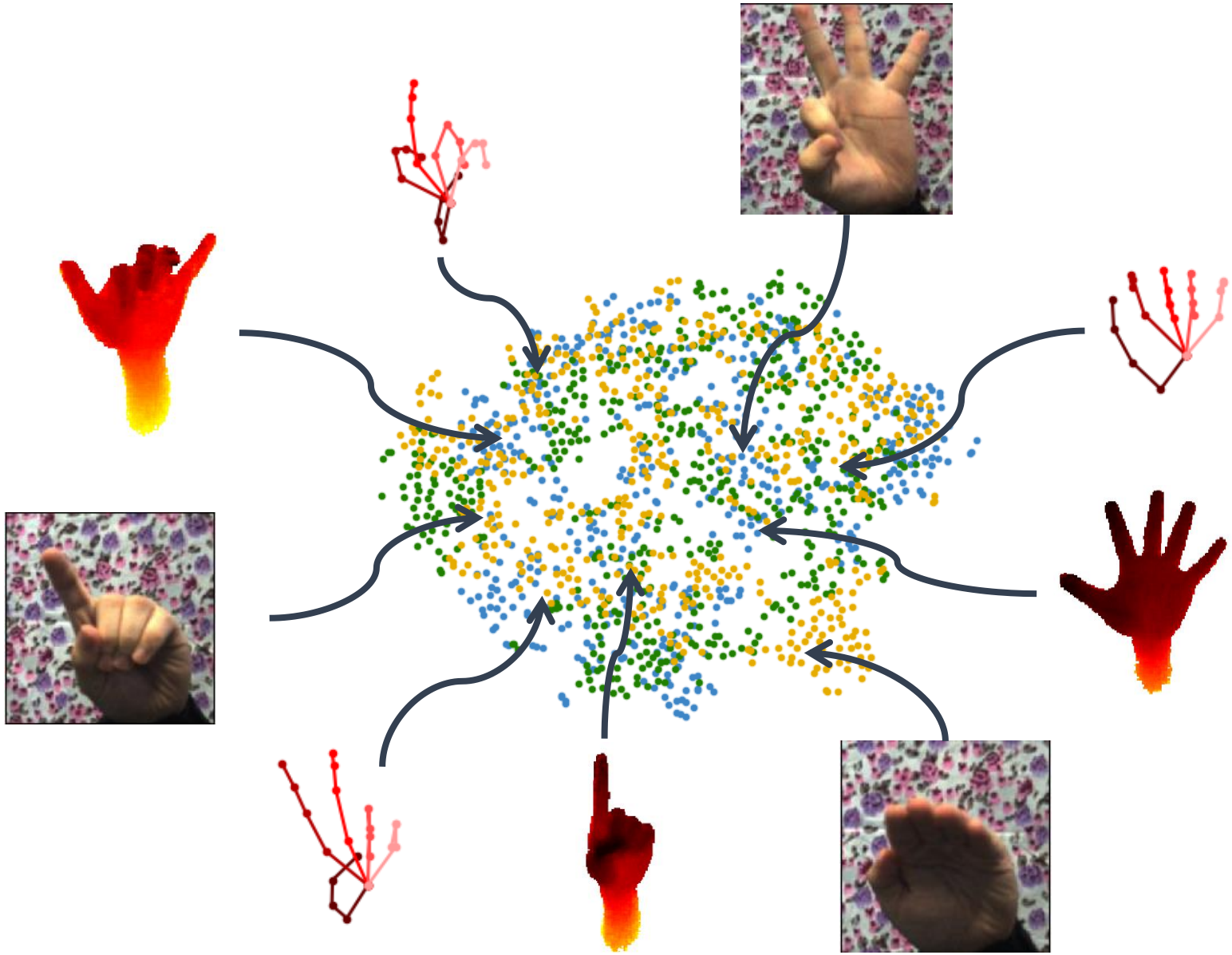

# Applications

# Assumption: Manifold of valid hand poses

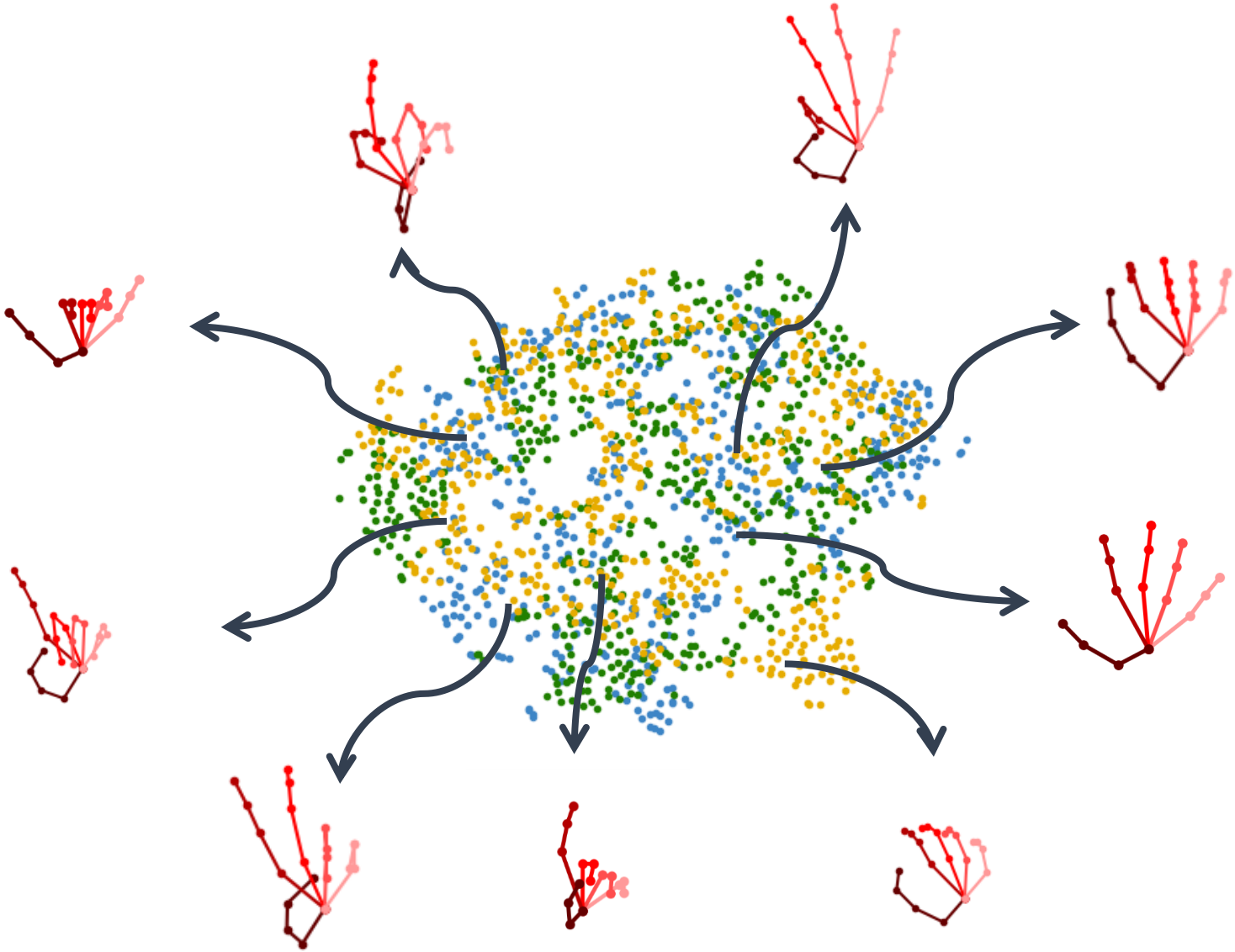


Adopted from [Tagliasacchi et al. 2015]

# Cross-modal latent space

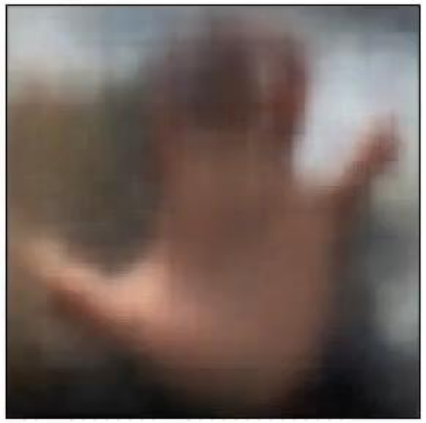


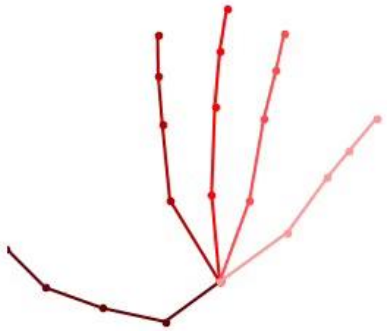
# Cross-modal latent space



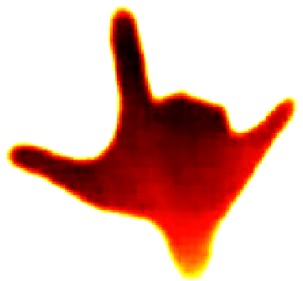
[Spurr et al.: CVPR ‘18]

# Synthetic hands: Latent Space Walk





[Synthetic samples RGB]



[Synthetic samples depth]

Digital ink promises to combine the flexibility and aesthetics of handwriting and the ability to process, search and edit digital text.

---

Character recognition converts handwritten text into a digital representation, albeit at the cost of losing style and personalized appearance due to the technical difficulties of separating the interwoven components of content and style.

---

In this paper we propose a novel generative neural network architecture that is capable of disentangling style from content and thus making digital ink editable.

---

Our model can synthesize arbitrary text while giving users control over the visual appearance.

---

For example allowing for style transfer without changing the content editing of digital ink at the character level and other application scenarios such as spellchecking and correction of handwritten text.

---

We furthermore contribute a new dataset of handwritten text with fine grained annotations at the character level and report results from an initial user evaluation.

[Aksan et al. CHI '18]

# Limitations of VAEs

Tendency to generate blurry images.

Believed to be due to injected noise and weak inference models (Gaussian assumption of latent samples to simplistic, model capacity to weak)

More expressive model → substantially better results (e.g Kingma et al. 2016, *Improving variational inference with inverse autoregressive flow*)

# Next Time

Deep Generative Modelling II:

Generative Adversarial Networks (GANs)