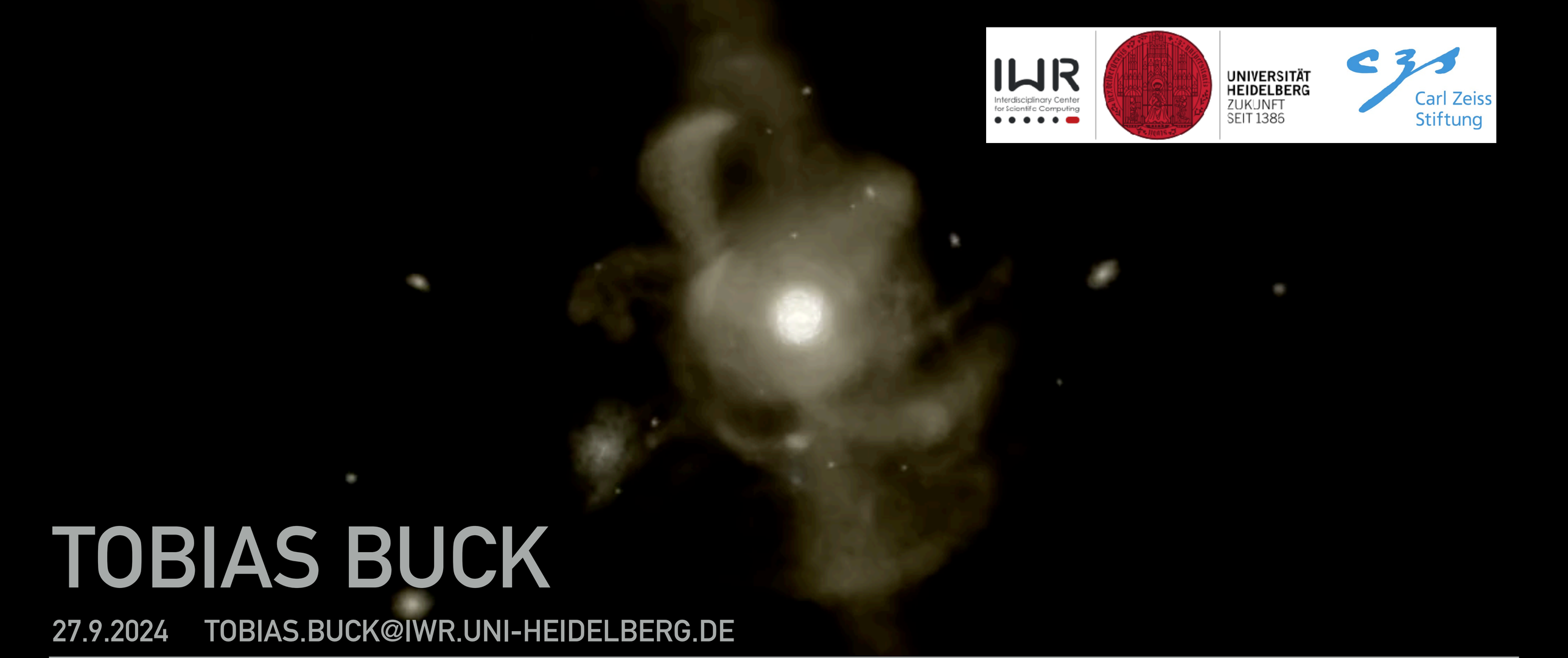


TOBIAS BUCK

27.9.2024 TOBIAS.BUCK@IWR.UNI-HEIDELBERG.DE

GENERATIVE AI

OUTLINE

1. Day 2: Generative AI
   A. Variational Autoencoder
   B. Normalizing Flows
   C. Score matching / diffusion models
   D. Simulation-Based Inference
   E. Model Comparison & Model Misspecification

Motivation

MODELING COMPLEX PROBABILITY DISTRIBUTIONS

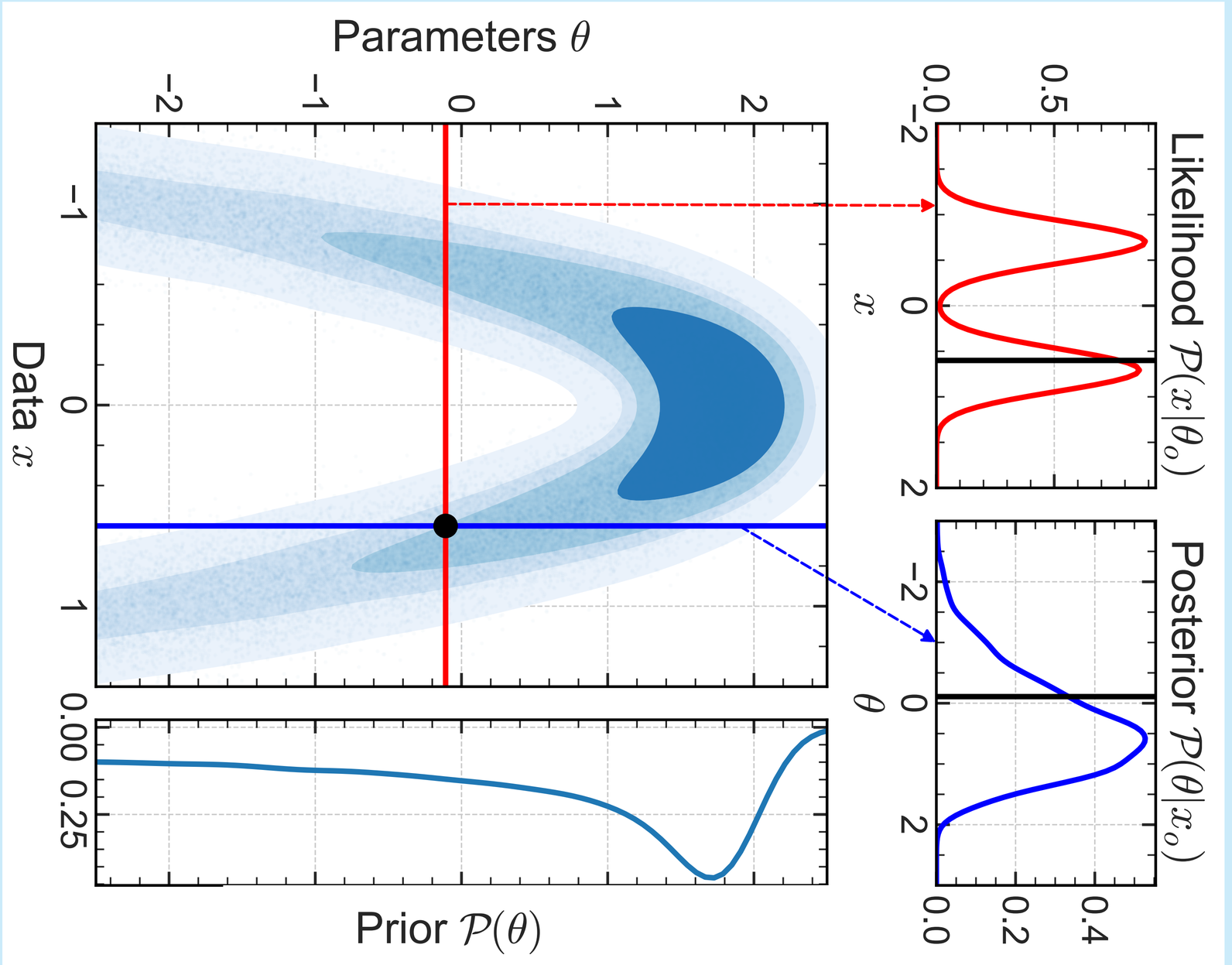
Bayesian inference aims at determining $p(\theta \,|\, \mathbf{x_o})$

$$p(\theta \,|\, \mathbf{x_0}) = \frac{p(\mathbf{x_0} \,|\, \theta) p(\theta)}{p(\mathbf{x_0})} \propto p(\mathbf{x_0} \,|\, \theta) p(\theta)$$

In astrophysics, $\mathbf{x_o}$ typically results from a large number of mechanisms/effects that transform the data and involve a large number of latent variables z, hence the marginal likelihood $p(\mathbf{x_o} \,|\, \theta)$ is intractable.

$$p(\mathbf{x_0} \,|\, \theta) = \int p(\mathbf{x_0} \,|\, \theta, z) p(z) dz$$

LIKELIHOODS, PRIORS AND POSTERIORS



MODELING COMPLEX PROBABILITY DISTRIBUTIONS

How can we approximate high-dimensional, complex probability distributions $p(\theta \mid \mathbf{x_o})$?

Goal:

- effectively: learn a model from the data!
- model $p(\theta \mid \mathbf{x_o})$ explicitly or implicitly
- sample and evaluate $p(\theta \mid \mathbf{x_o})$

Options:

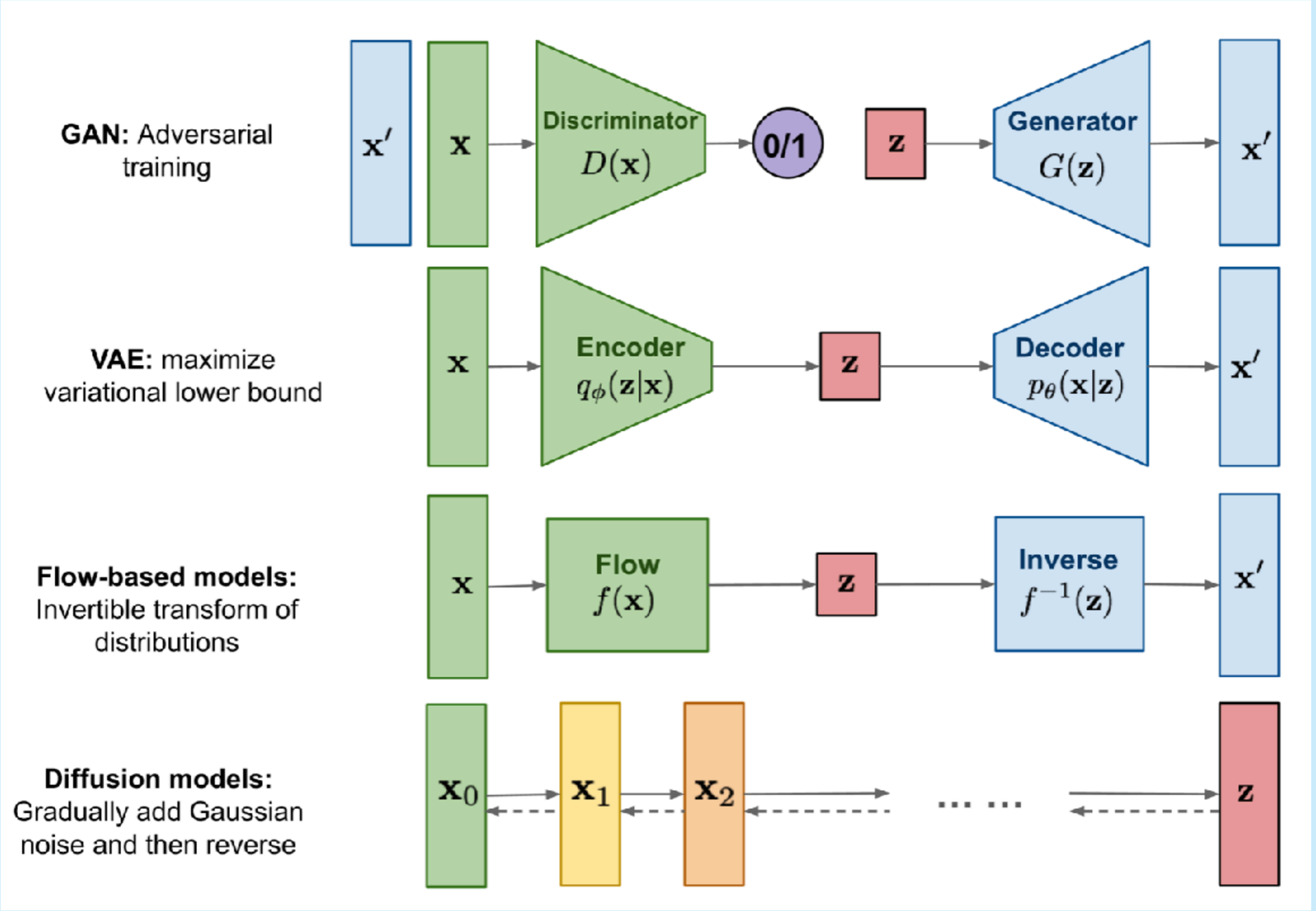
- normalizing flows
- VAEs
- GANs
- score matching / flow matching
- and possibly more

implicit model: architectural constraints

explicit model: prone to mode collapse

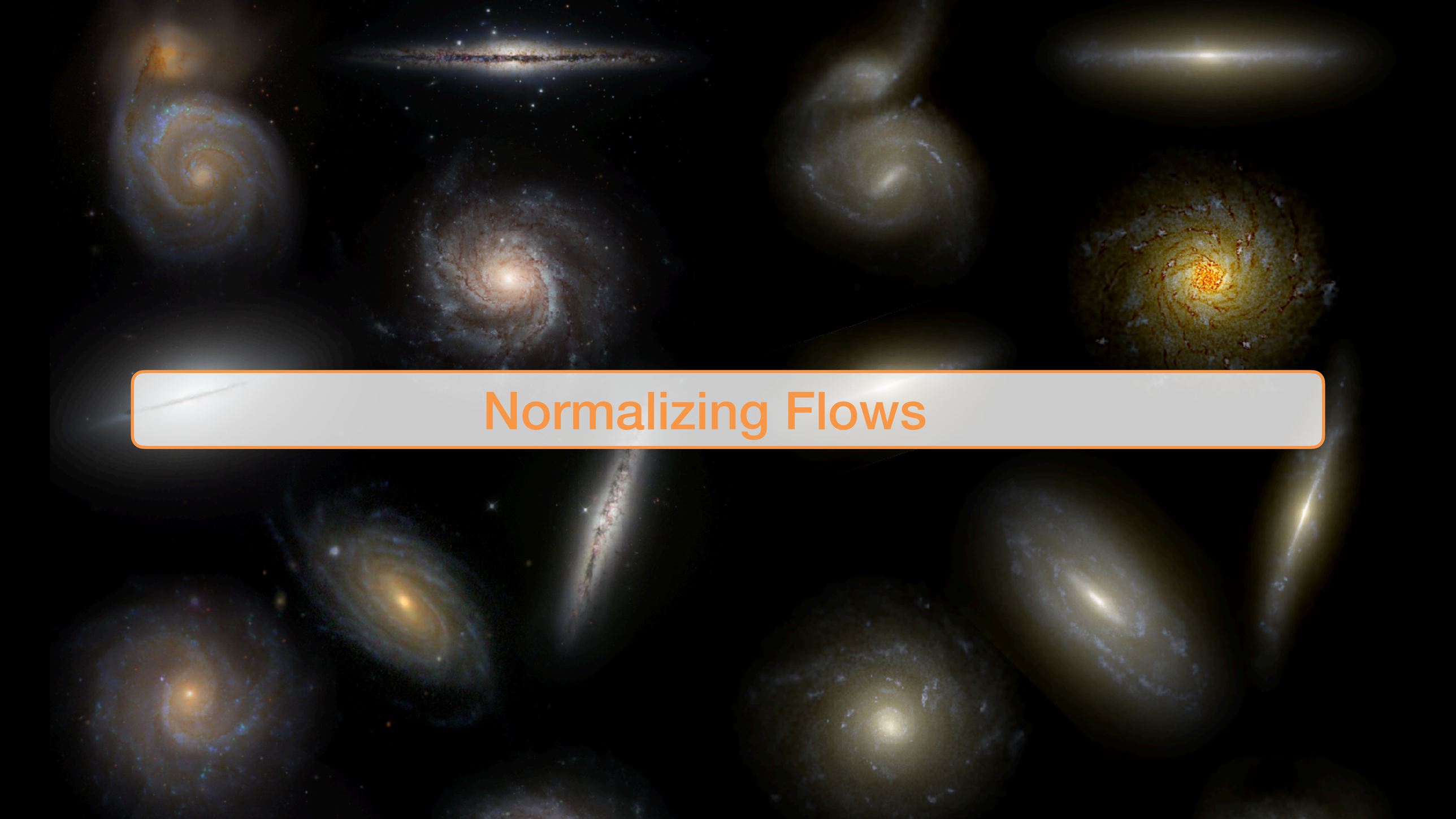
GENERATIVE AI FLAVOURS

"Creating noise from data is easy; creating data from noise is generative modeling.“ (Song+2020)



blog post by https://lilianweng.github.io/posts/2021-07-11-diffusion-models/


Likelihood-based models



Normalizing Flows

NORMALIZING FLOWS

How can we approximate high-dimensional, complex probability distributions $p(\theta \mid \mathbf{x_o})$?

Goal:

- model $p(\theta \mid \mathbf{x_o})$ explicitly
- sample and evaluate $p(\theta \mid \mathbf{x_o})$

Advantages of NF:

- exact log-likehood calculation
- density evaluation possible

Idea:

Transform a simple base distribution through a series of invertible transformations.

MODELING COMPLEX PROBABILITY DISTRIBUTIONS - NORMALIZING FLOWS

$$T = T_1 \circ T_2$$

NORMALISING FLOW THEORY

Normalizing Flows: Change of variables

$$1 = \int p_Z(z)dz$$

Transform: $\mathrm{x} = T(z)$

$$\int p_X(x)dx = 1 = \int p_Z(z)dz = \int p_Z(T^{-1}(x)) \left|\det J_{T^{-1}}(x)\right| dx$$

$$p_X(x) = p_Z(T^{-1}(x)) \left|\det J_{T^{-1}}(x)\right|$$

NORMALISING FLOW THEORY

Normalizing Flows: Definition

Base distribution $z \sim p_Z(z)$ $\xrightarrow{T}$ **Target distribution** $x = T(z)$

Defining Properties:

- T must be invertible
- T, T^{-1} differentiable

Diffeomorphism

$$T = T_N \circ T_{N-1} \circ \cdots \circ T_1$$

$$T^{-1} = T_1^{-1} \circ T_2^{-1} \circ \cdots \circ T_N^{-1}$$

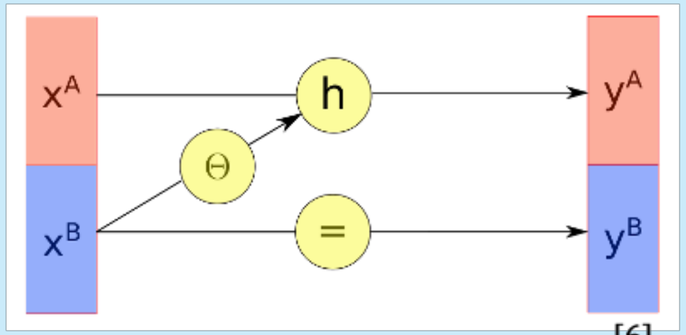
$$\det J_{T_2 \circ T_1}(z) = \det J_{T_2}(T_1(z)) \cdot \det J_{T_1}(z)$$

NORMALISING FLOW: IMPLEMENTING TRANSFORMATIONS

Coupling Flows

- Two partitions:

$$y^A = h\left(x^A, \theta(x^B)\right)$$
$$y^B = x^B$$



[6]

- Conditioner $\theta(x^B)$ does not have to be invertible
  → use Neural Networks for good expressivity
- Plethora of different coupling functions

NORMALISING FLOW: IMPLEMENTING TRANSFORMATIONS

Coupling Functions (How to choose h?)

- Additive $h(x;\theta) = x + \theta$
- Affine $h(x;\theta) = \theta_1 x + \theta_2$ (NICE, RealNVP, Glow, MAF, IAF)

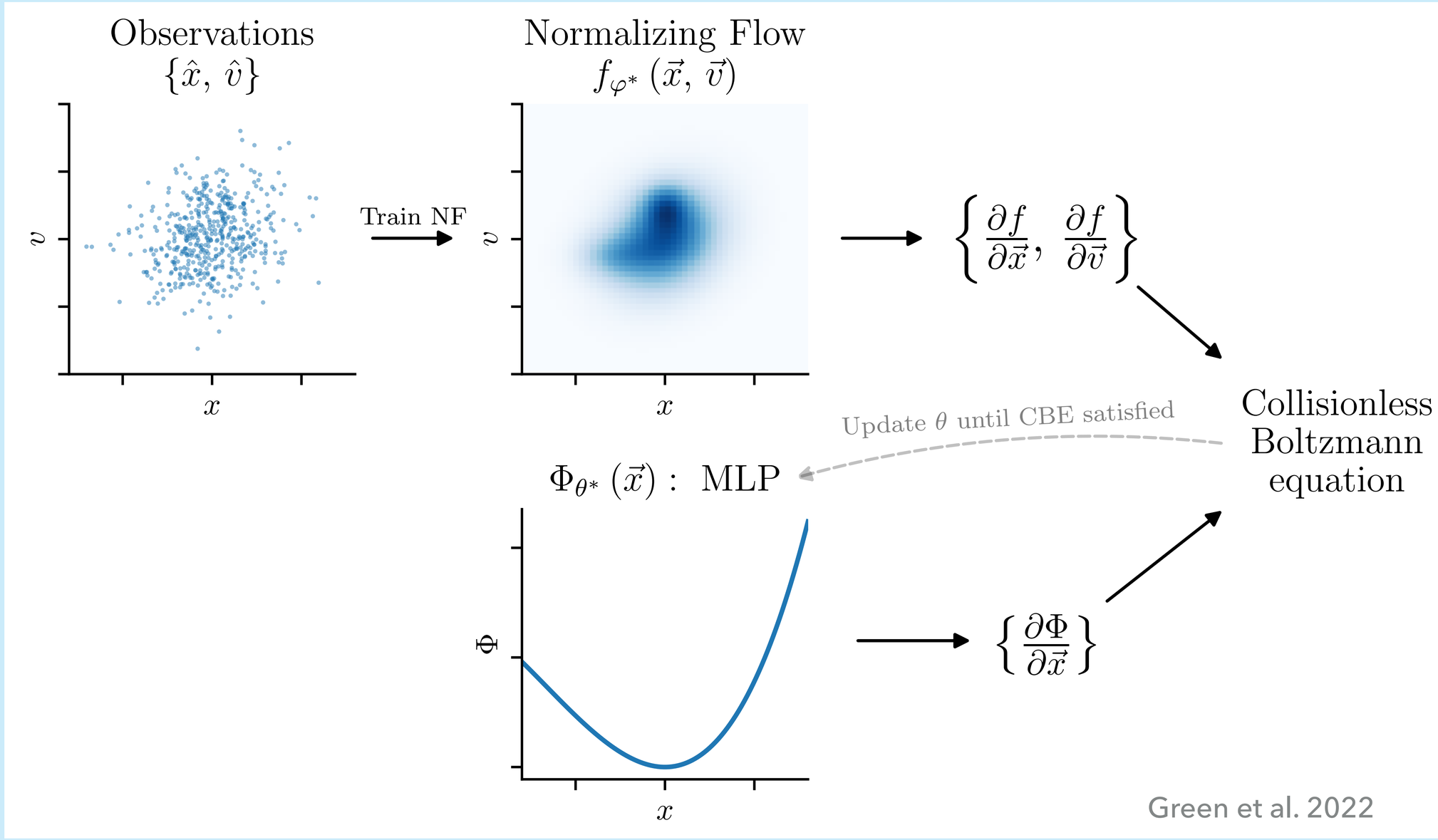
$$y_{1:d-1} = x_{1:d-1}$$

$$y_{d:n} = \exp(s_{d:n}(x_{1:d-1})) \cdot y_{d:n} + t_{d:n}(x_{1:d-1})$$

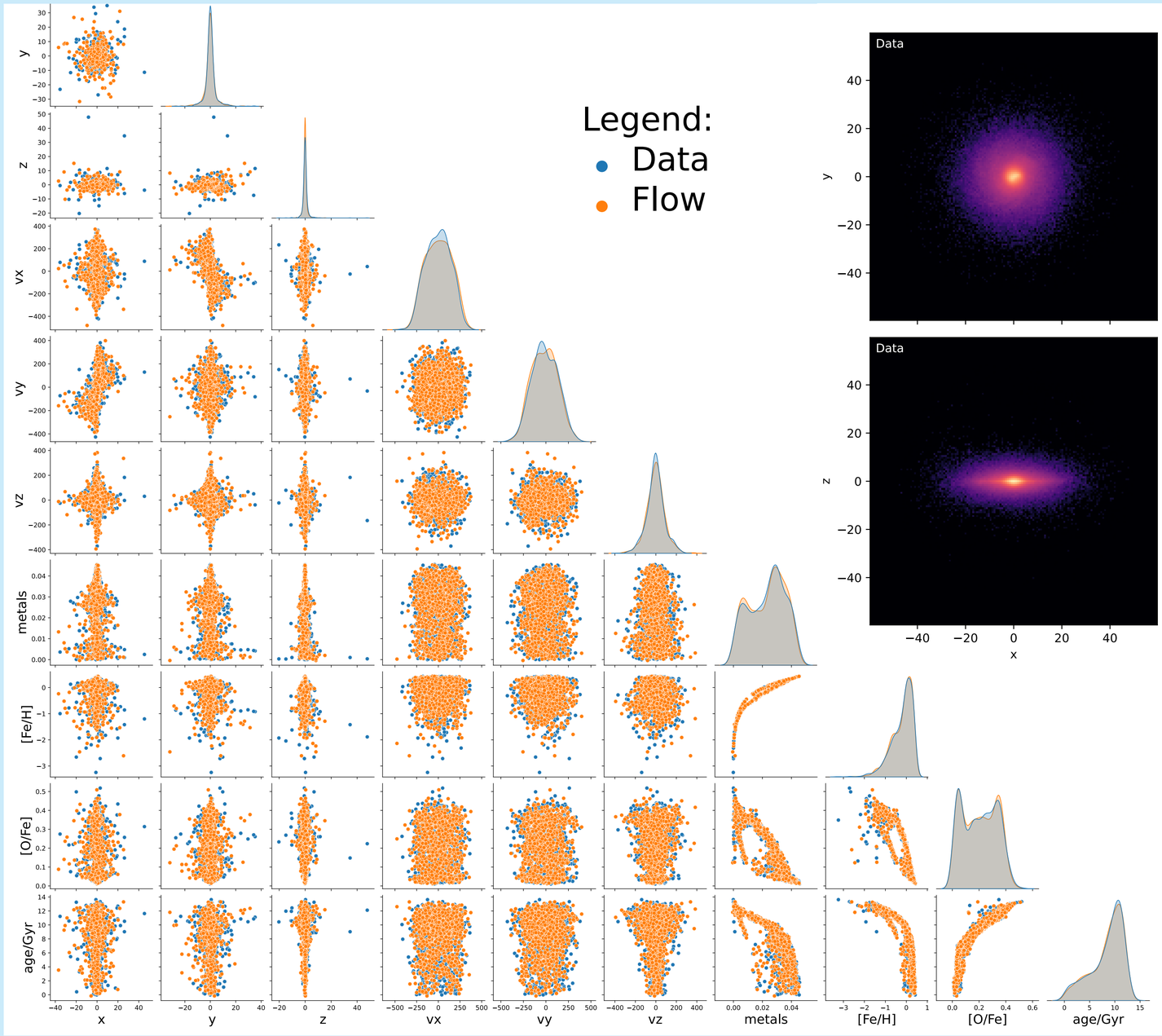
- Continuous mixture CDFs $h(x;\theta) = \theta_1 F(x, \theta_3) + \theta_2$
- Splines (RQ-NSF)
- Neural Networks

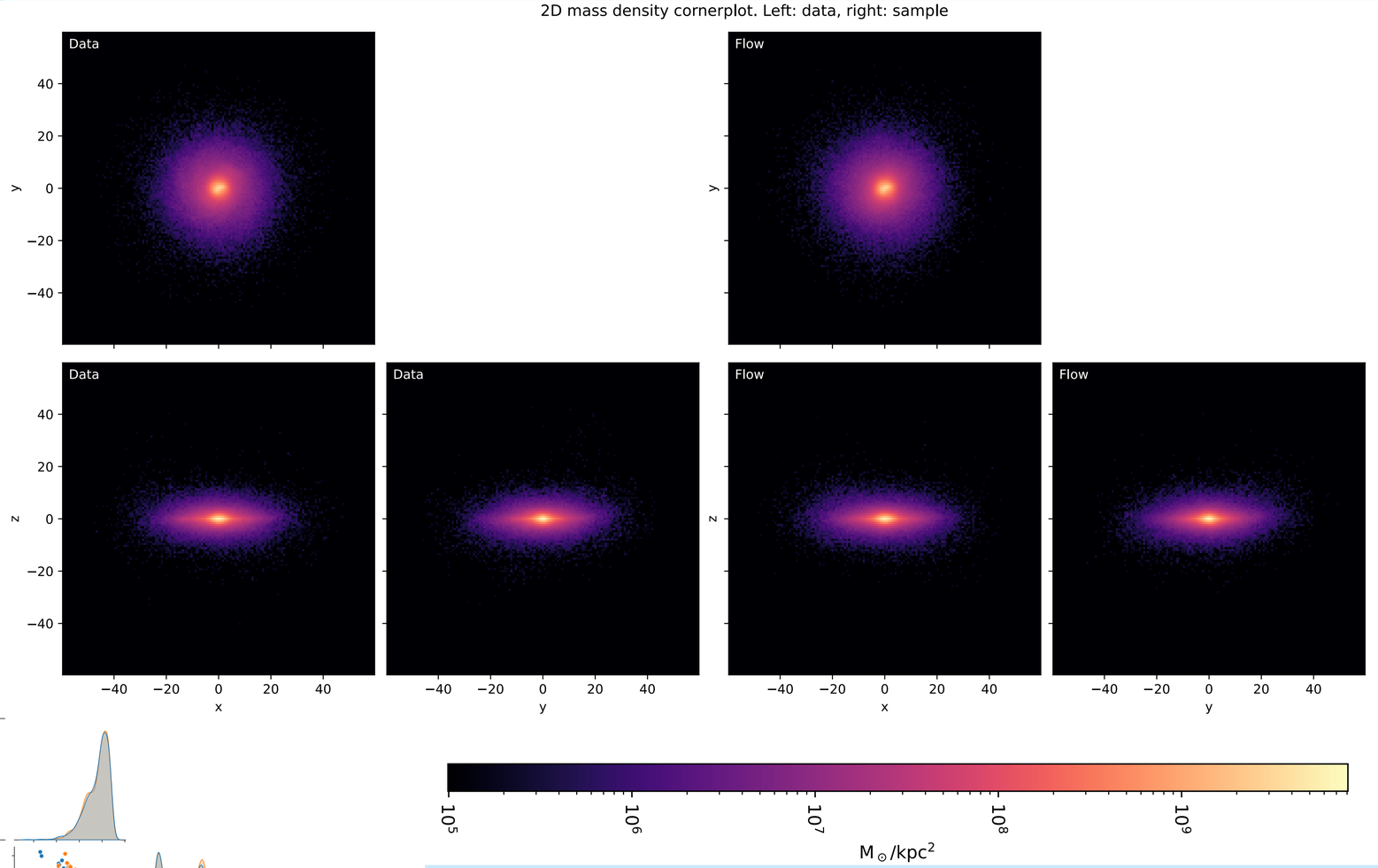
Important: after each coupling layer permute dimensions (or use 1x1 convolution like GLOW)

NORMALISING FLOW: APPLICATION I



NORMALISING FLOW: APPLICATION II



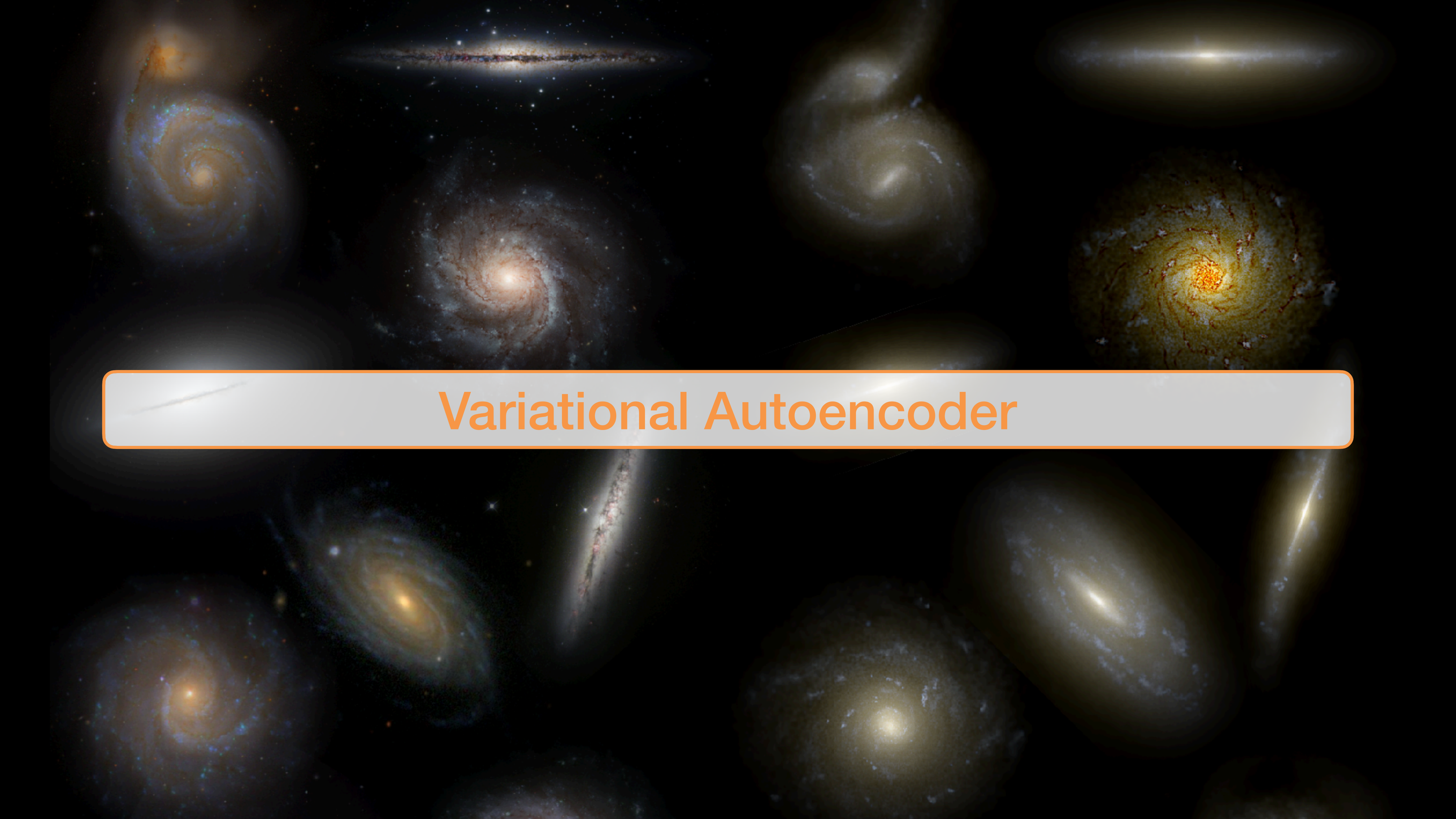


Wolf+Buck 2023

NORMALISING FLOW: APPLICATION III

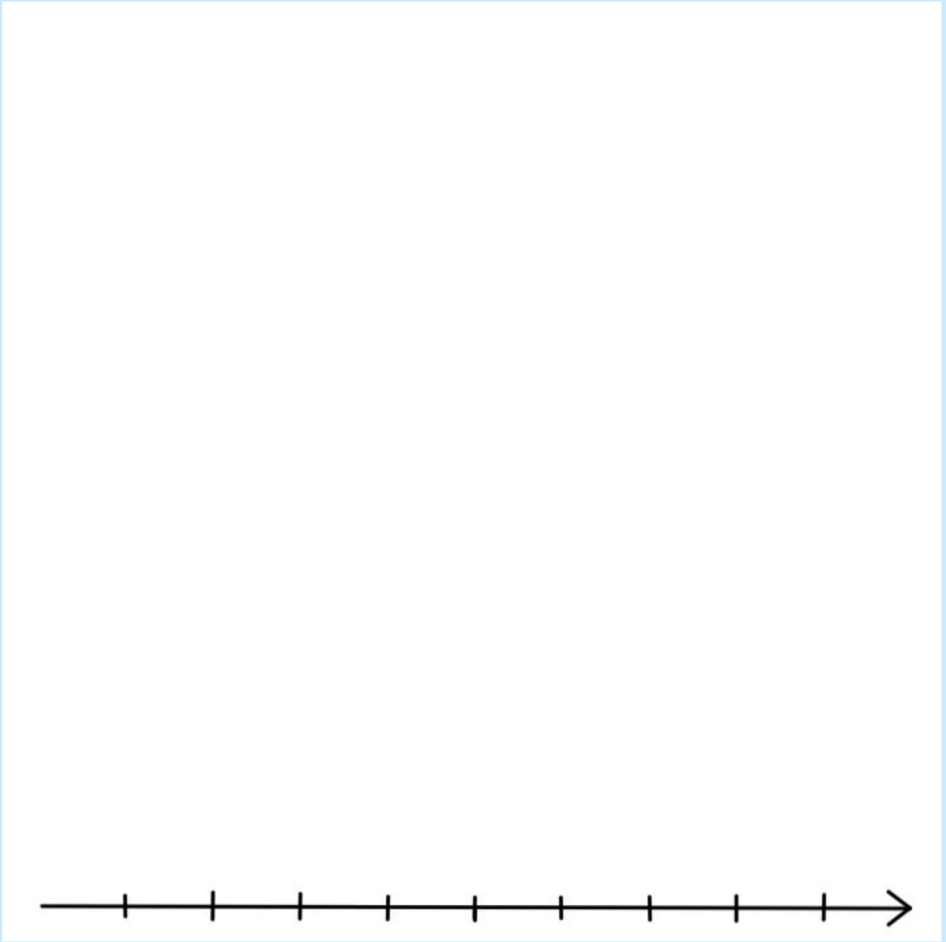
- Normalizing flows for random fields in cosmology (Rouhiainen+2021)
- Bayesian Stokes inversion with normalizing flows (Baso+2022)
- A Hierarchy of Normalizing Flows for Modelling the Galaxy-Halo Relationship (Lovell+2023)
- HIFlow: Generating Diverse Hi Maps and Inferring Cosmology while Marginalizing over Astrophysics Using Normalizing Flows (Hassan+2022)
- Normalizing Flows as an Avenue to Studying Overlapping Gravitational Wave Signals (Langendorff+2023)
- Charting Galactic Accelerations: When and How to Extract a Unique Potential from the Distribution Function (An+2021)
- Charting galactic accelerations II: how to ‘learn’ accelerations in the solar neighbourhood (Naik+2021)

- many many more…



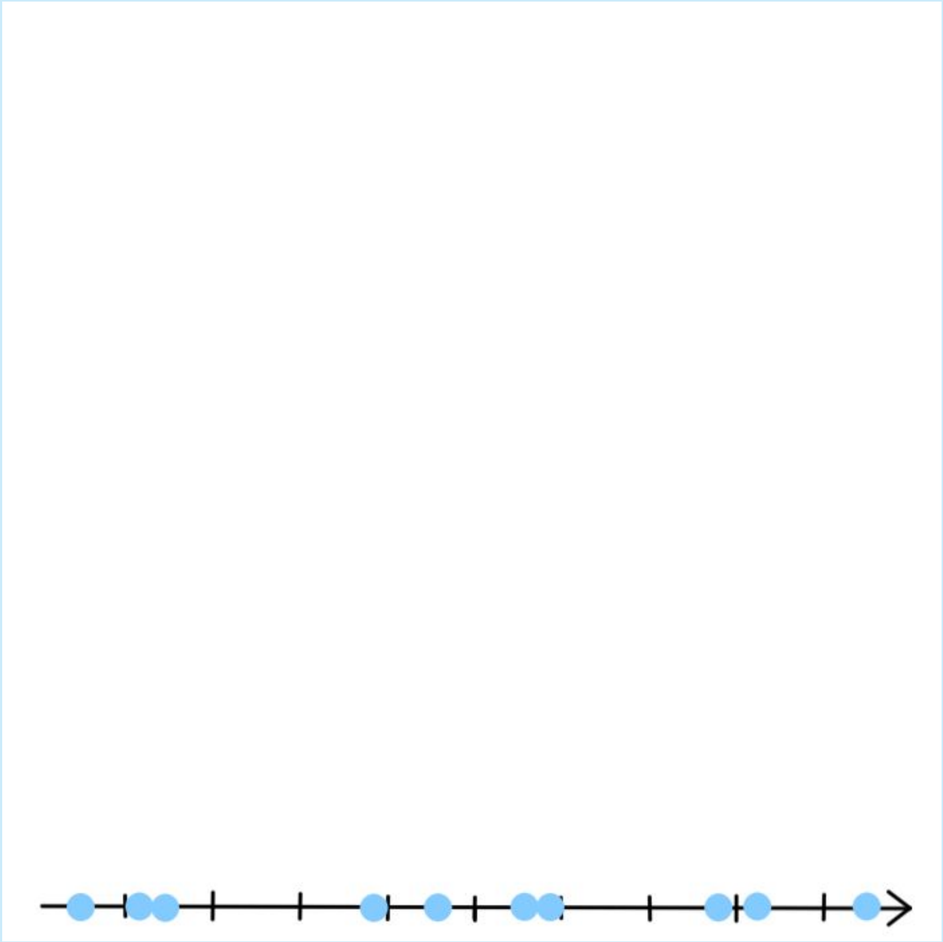
Variational Autoencoder

THE CURSE OF DIMENSIONALITY



Divide range into K intervals Δx_1

THE CURSE OF DIMENSIONALITY



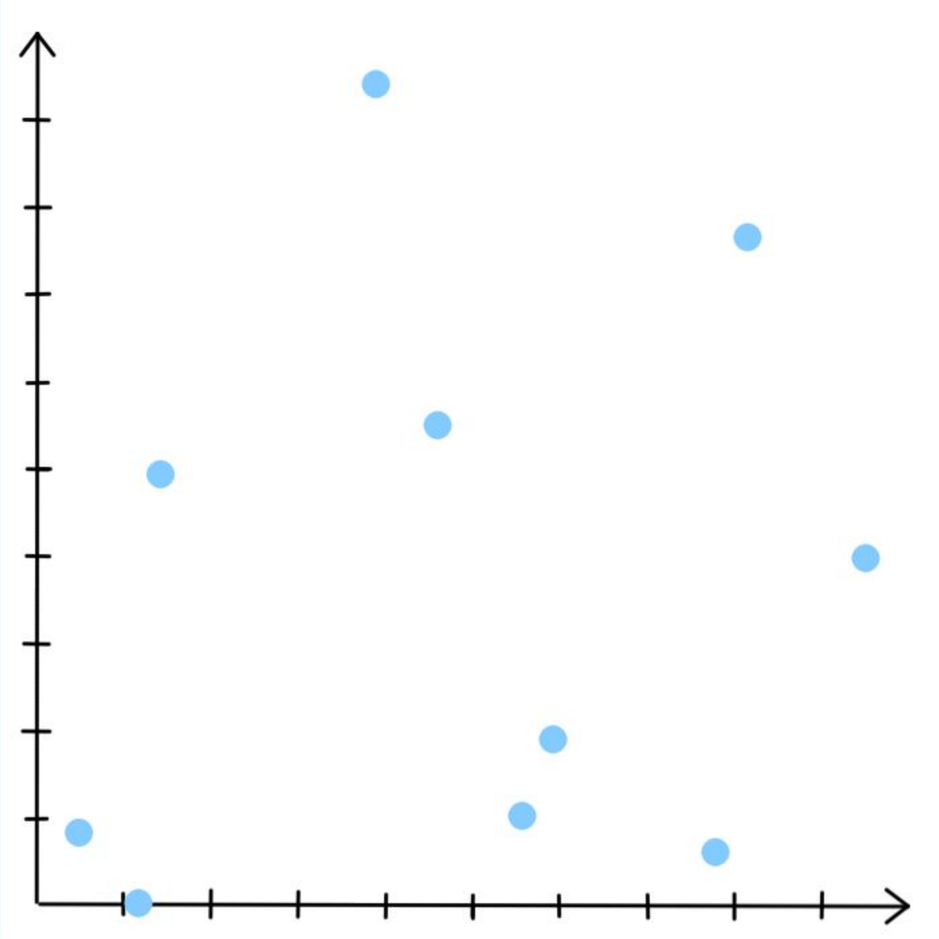
Divide range into K intervals Δx_1

Take data set of size N

On average: $n = N/K$ points per interval

$\rightarrow$ Here: $n = 10/10 = 1$

THE CURSE OF DIMENSIONALITY



Divide range into K intervals Δx_1

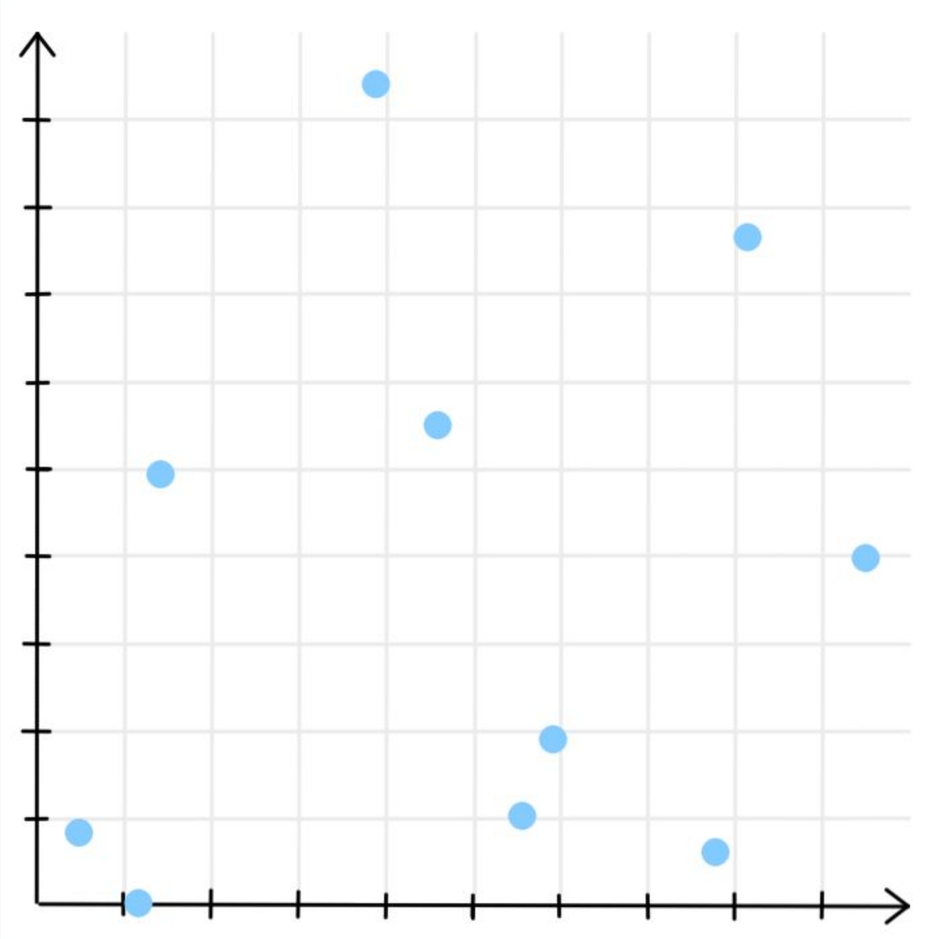
Take data set of size N

On average: $n = N/K$ points per interval

$\rightarrow$ Here: $n = 10/10 = 1$

For 2D: K^2 intervals of size $\Delta x_1 \Delta x_2$

THE CURSE OF DIMENSIONALITY



Divide range into K intervals Δx_1

Take data set of size N

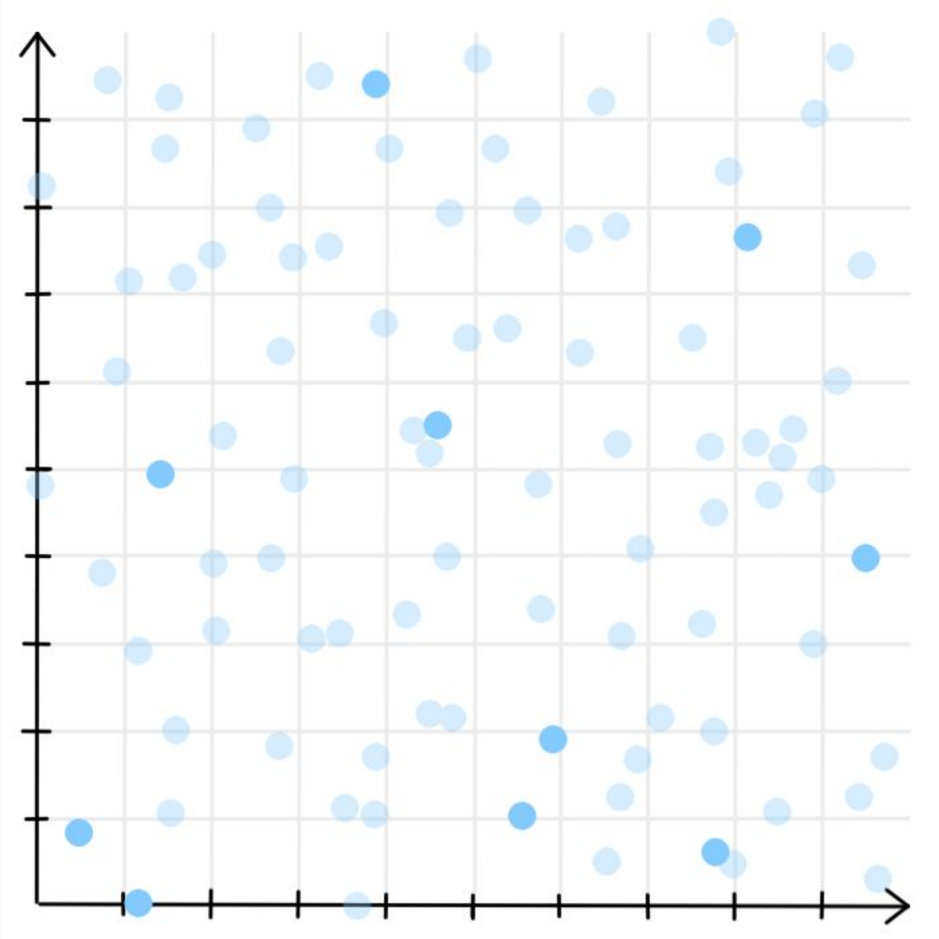
On average: $n = N/K$ points per interval

$\rightarrow$ Here: $n = 10/10 = 1$

For 2D: K^2 intervals of size $\Delta x_1 \Delta x_2$

On average: $n = 10/100 = 0.1$

THE CURSE OF DIMENSIONALITY



Divide range into K intervals Δx_1

Take data set of size N

On average: $n = N/K$ points per interval

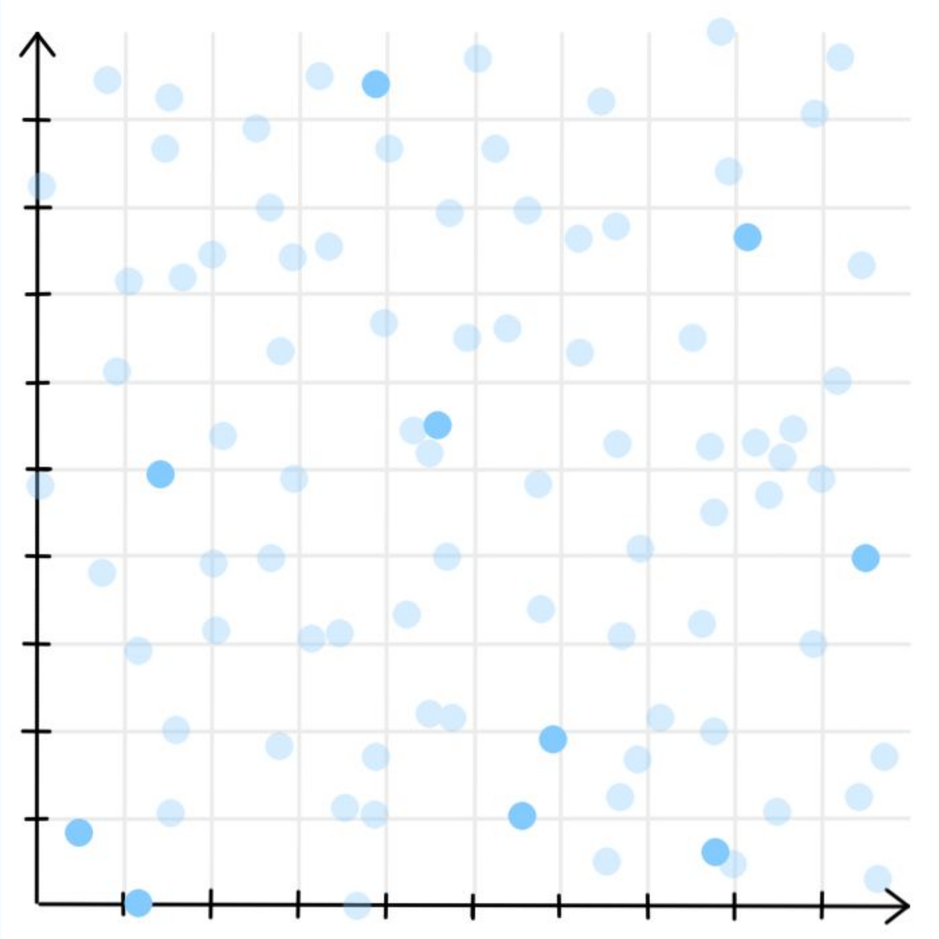
$\rightarrow$ Here: $n = 10/10 = 1$

For 2D: K^2 intervals of size $\Delta x_1 \Delta x_2$

On average: $n = 10/100 = 0.1$

For same n, $nK^2 = N^2 = 100$ data points needed

THE CURSE OF DIMENSIONALITY



Divide range into K intervals Δx_1

Take data set of size N

On average: $n = N/K$ points per interval

$\rightarrow$ Here: $n = 10/10 = 1$

For 2D: K^2 intervals of size $\Delta x_1 \Delta x_2$

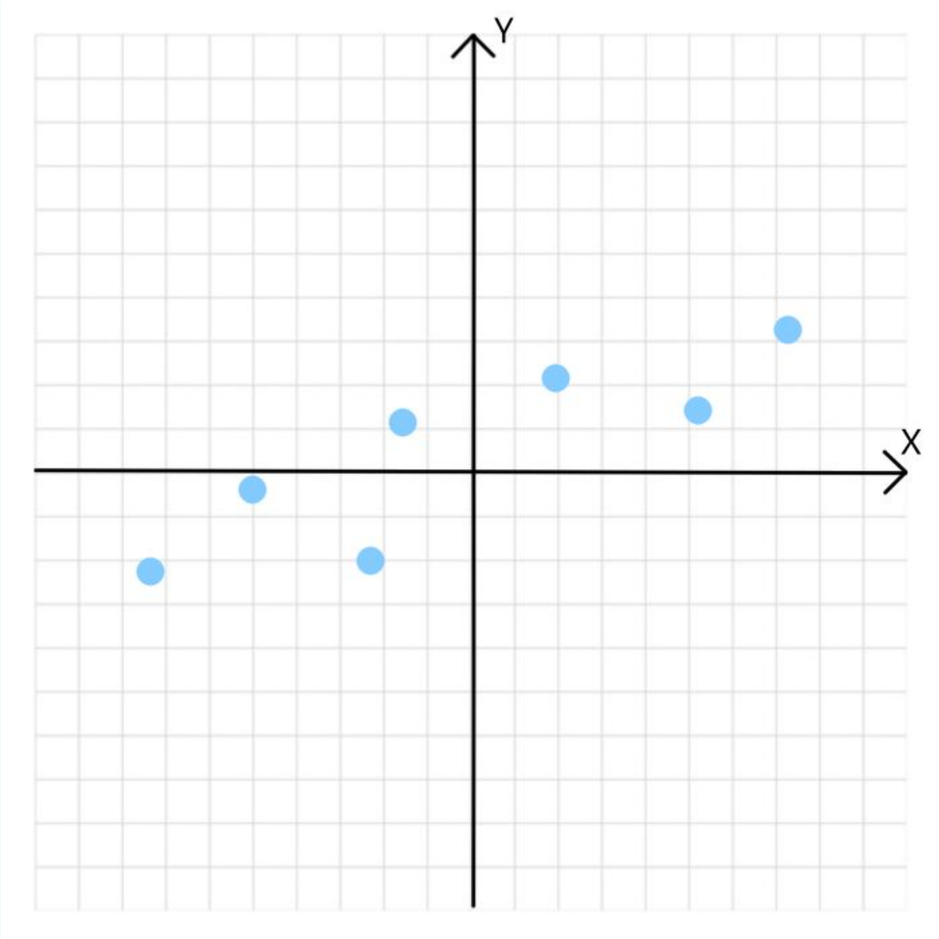
On average: $n = 10/100 = 0.1$

For same n, $nK^2 = N^2 = 100$ data points needed

For p dimensions: N^p data points required

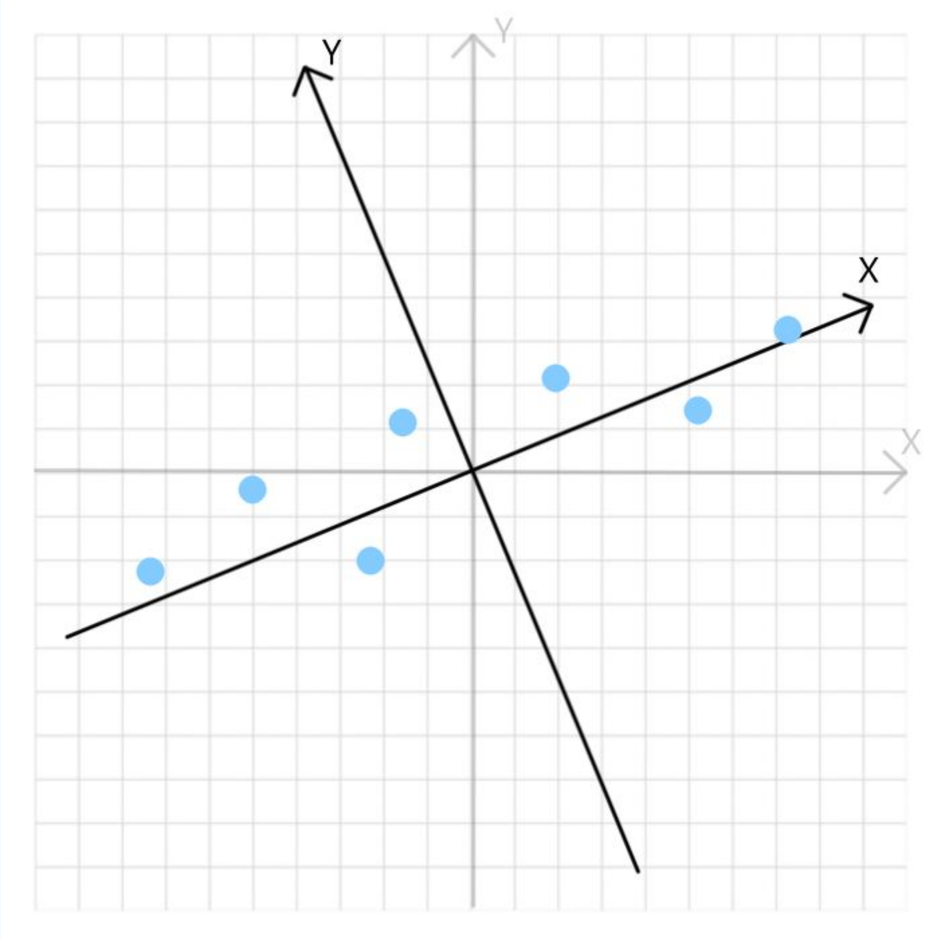
$\rightarrow p = 4$: 10.000 points

PRINCIPAL COMPONENT ANALYSIS



Maximise the variance along a new axis

PRINCIPAL COMPONENT ANALYSIS

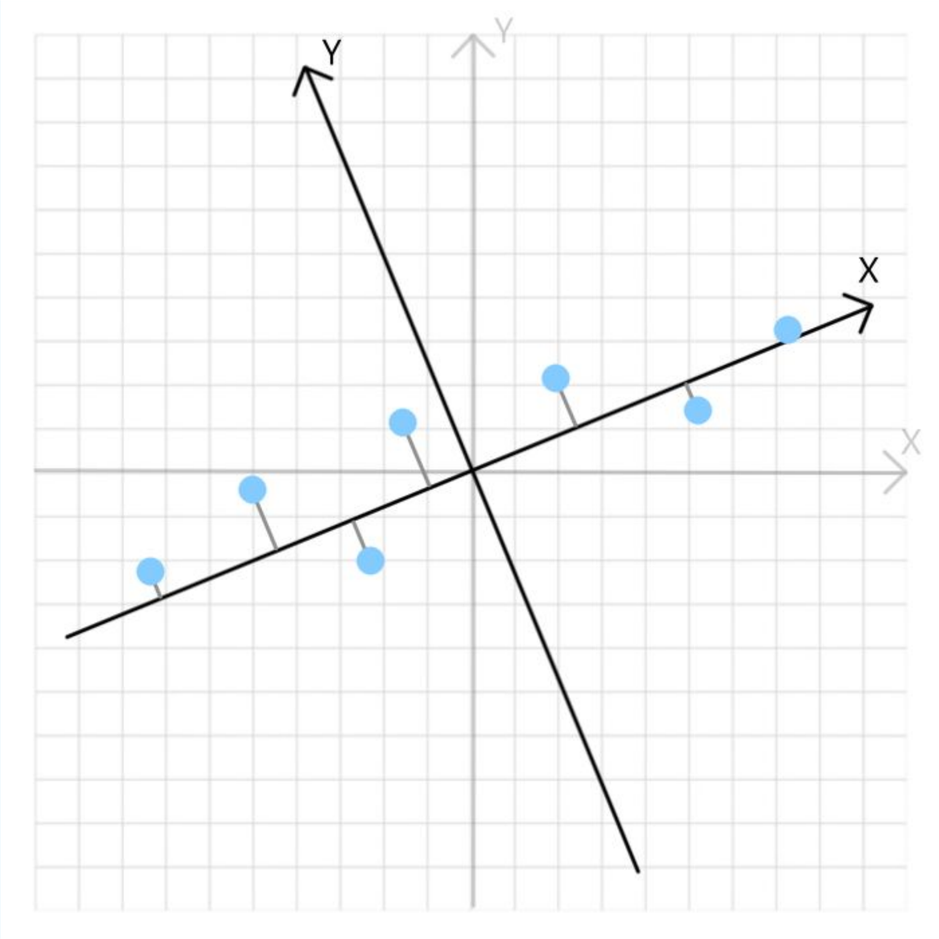


Maximise the variance along a new axis

Variance is maximised along x, and minimised along y

Neglect y to reduce dimensionality

PRINCIPAL COMPONENT ANALYSIS



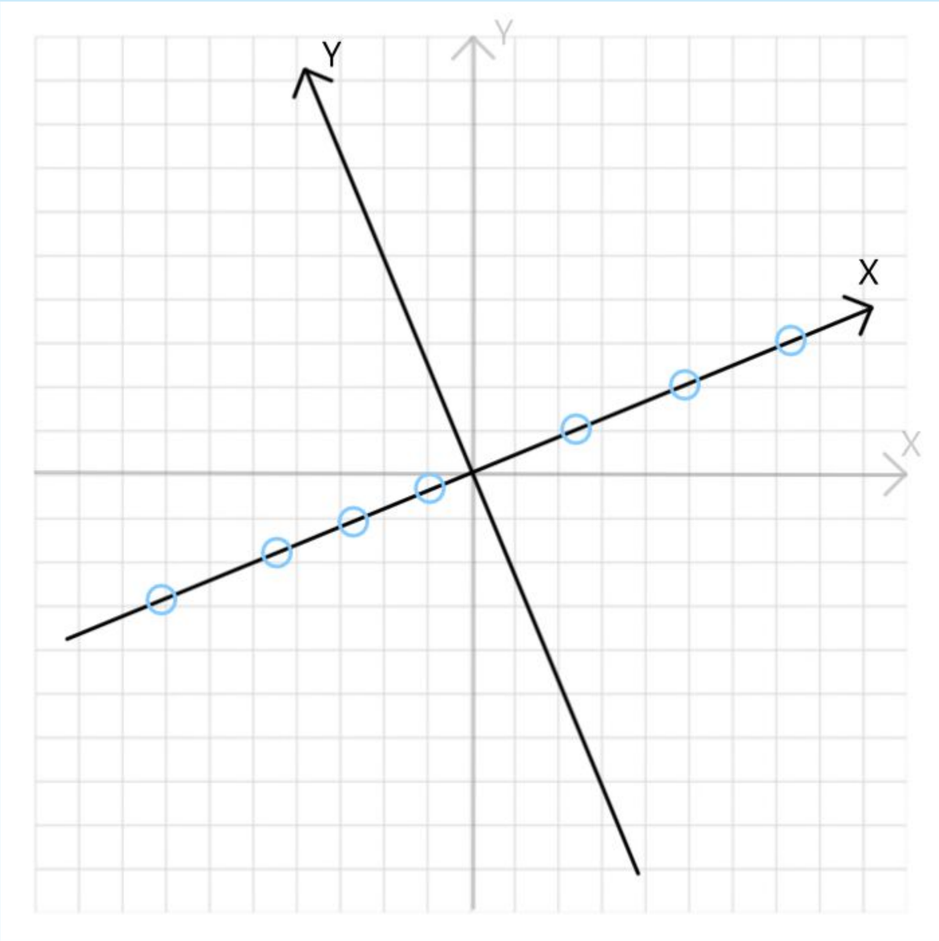
Maximise the variance along a new axis

Variance is maximised along x, and minimised along y

Neglect y to reduce dimensionality

→ Project data on x

PRINCIPAL COMPONENT ANALYSIS



Maximise the variance along a new axis

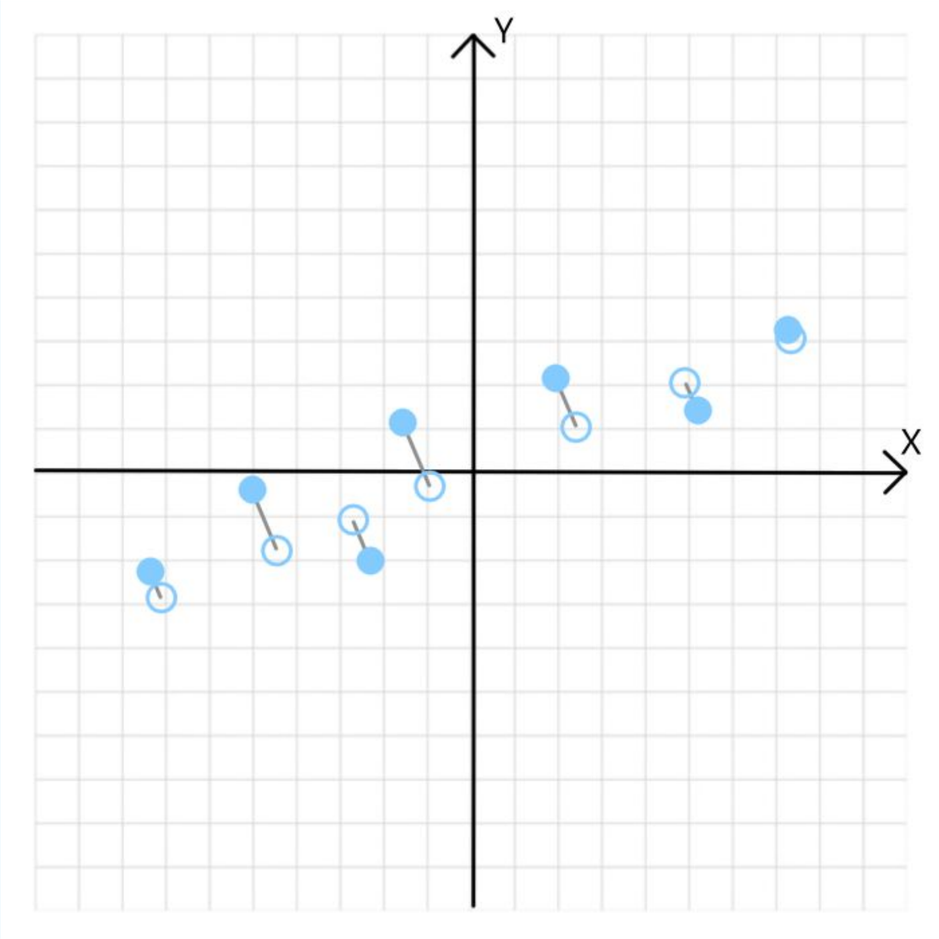
Variance is maximised along x, and minimised along y

Neglect y to reduce dimensionality

→ Project data on x

That’s the new dimensionally reduced data set

PRINCIPAL COMPONENT ANALYSIS



Maximise the variance along a new axis

Variance is maximised along x, and minimised along y

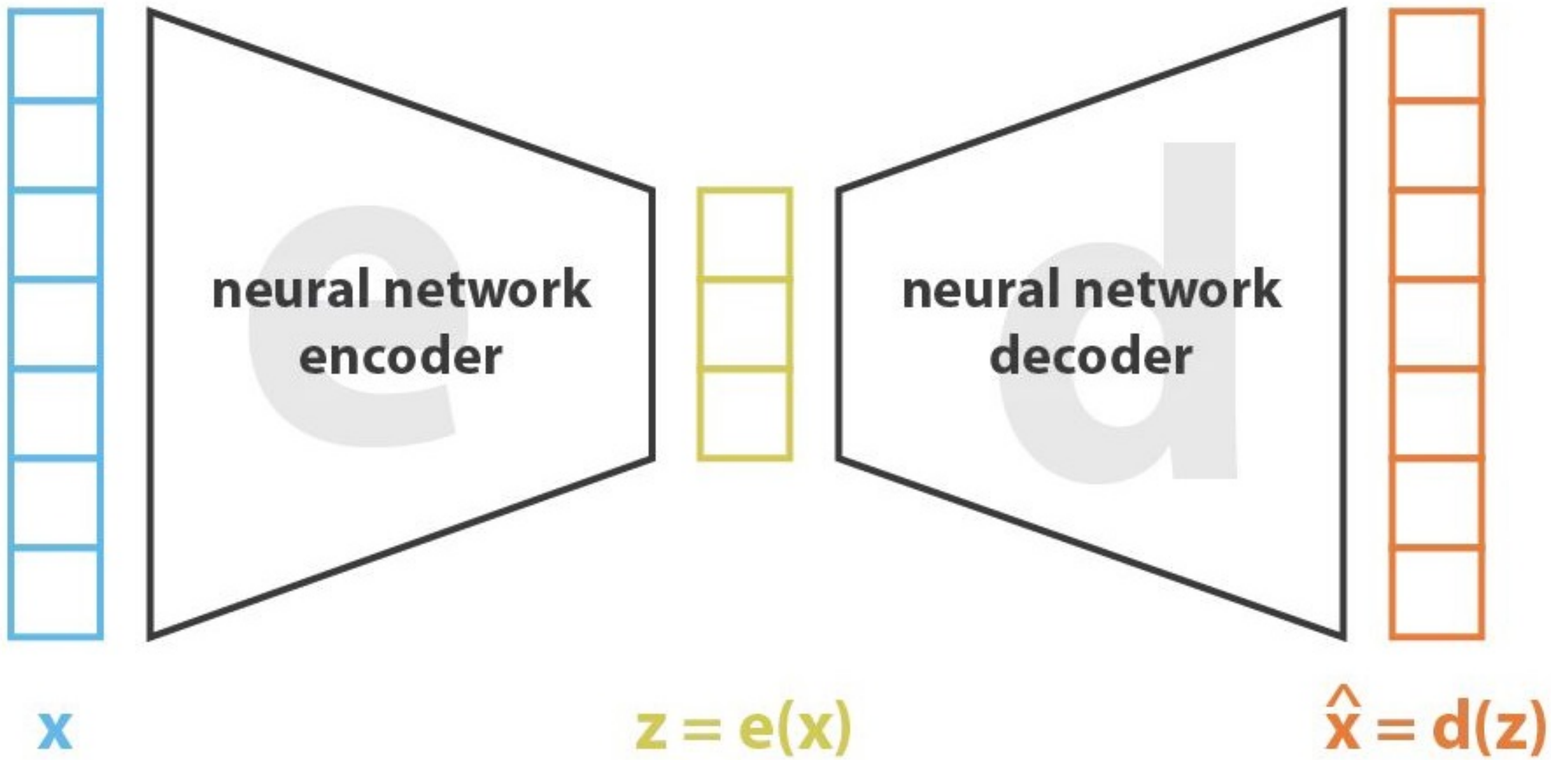
Neglect y to reduce dimensionality

→ Project data on x

That’s the new dimensionally reduced data set

Information (Reconstruction) loss is minimised

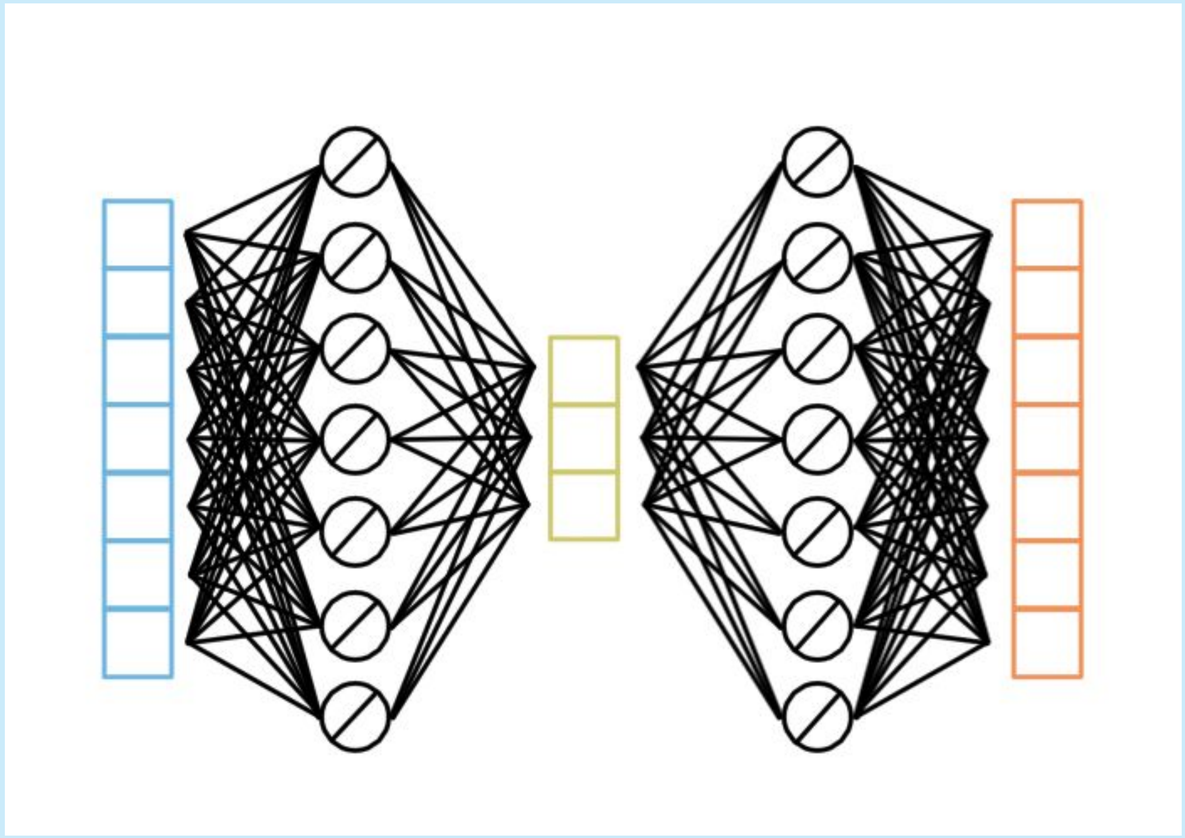
AUTOENCODER



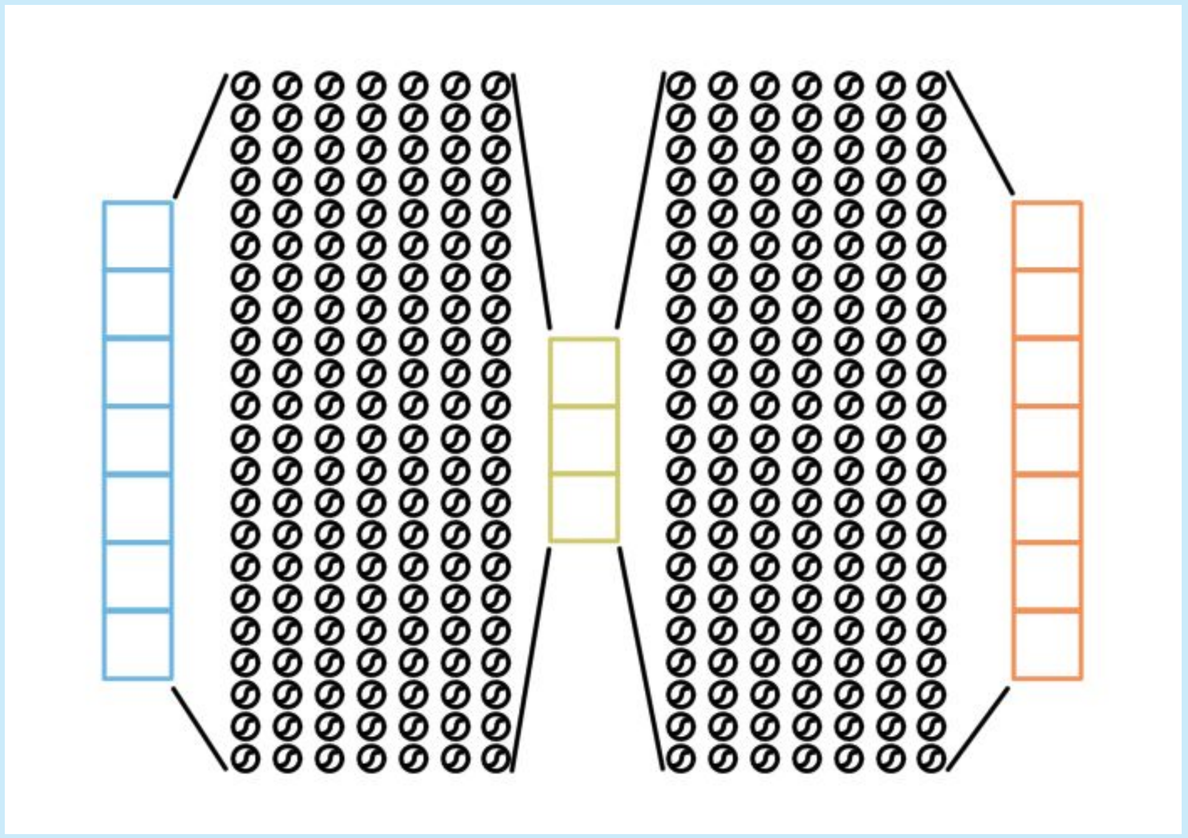
$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

AUTOENCODER

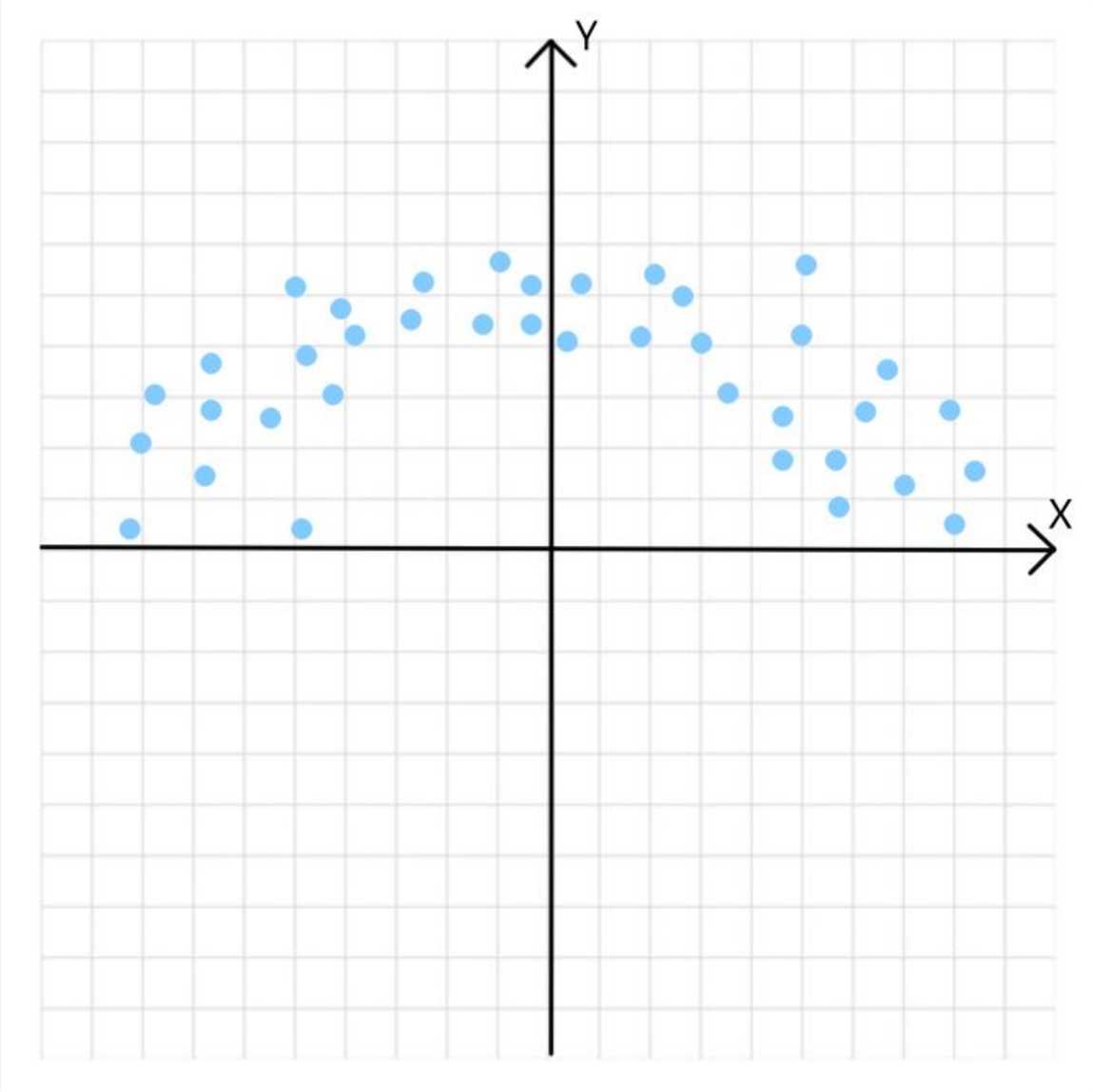


Linear transformation of inputs; very similar to PCA

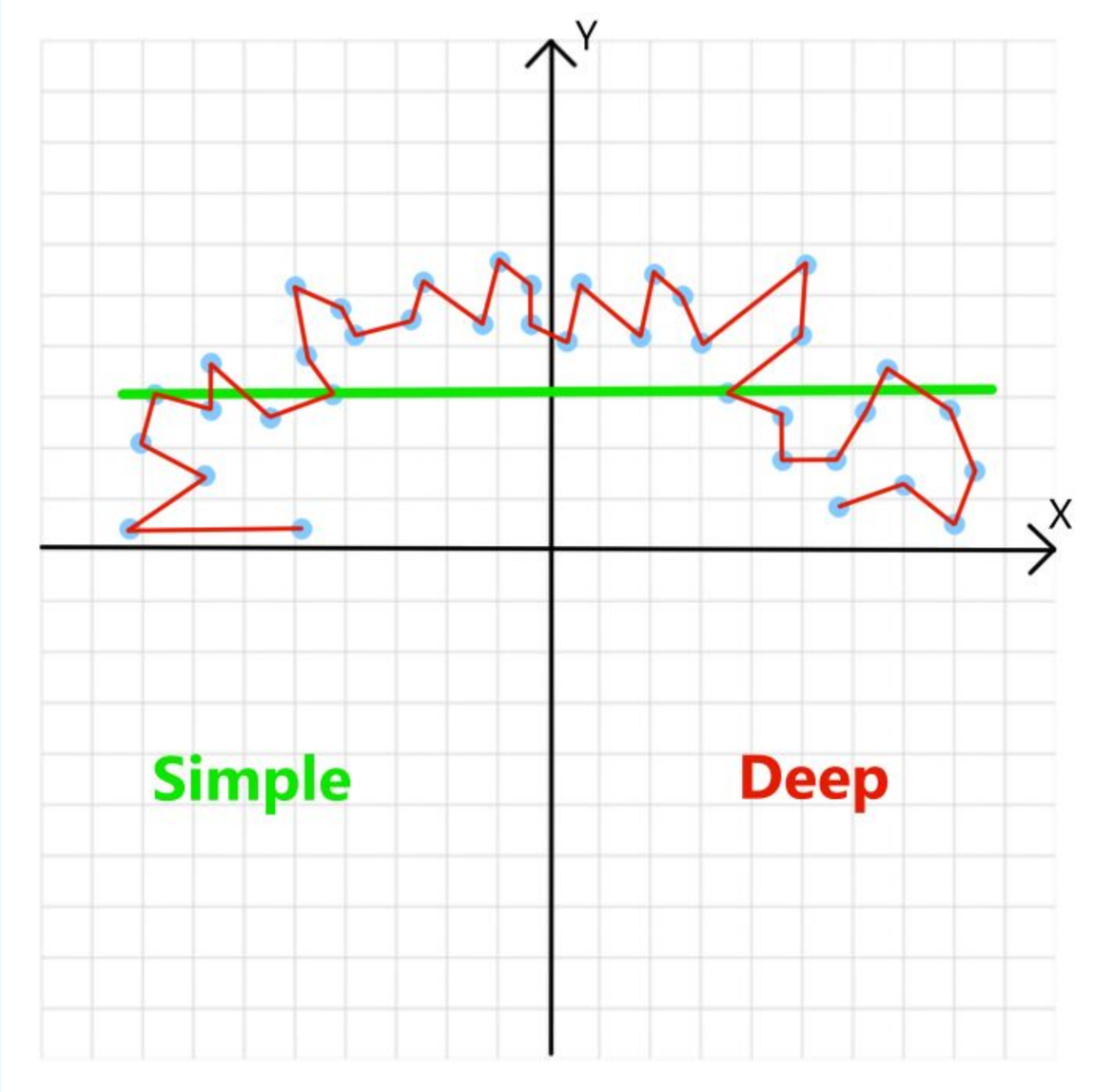


Infinitely deep network has the capacity for lossless compression into a single dimension

AUTOENCODER: LOSSLESS COMPRESSION



AUTOENCODER: LOSSLESS COMPRESSION



AUTOENCODER: LOSSLESS COMPRESSION

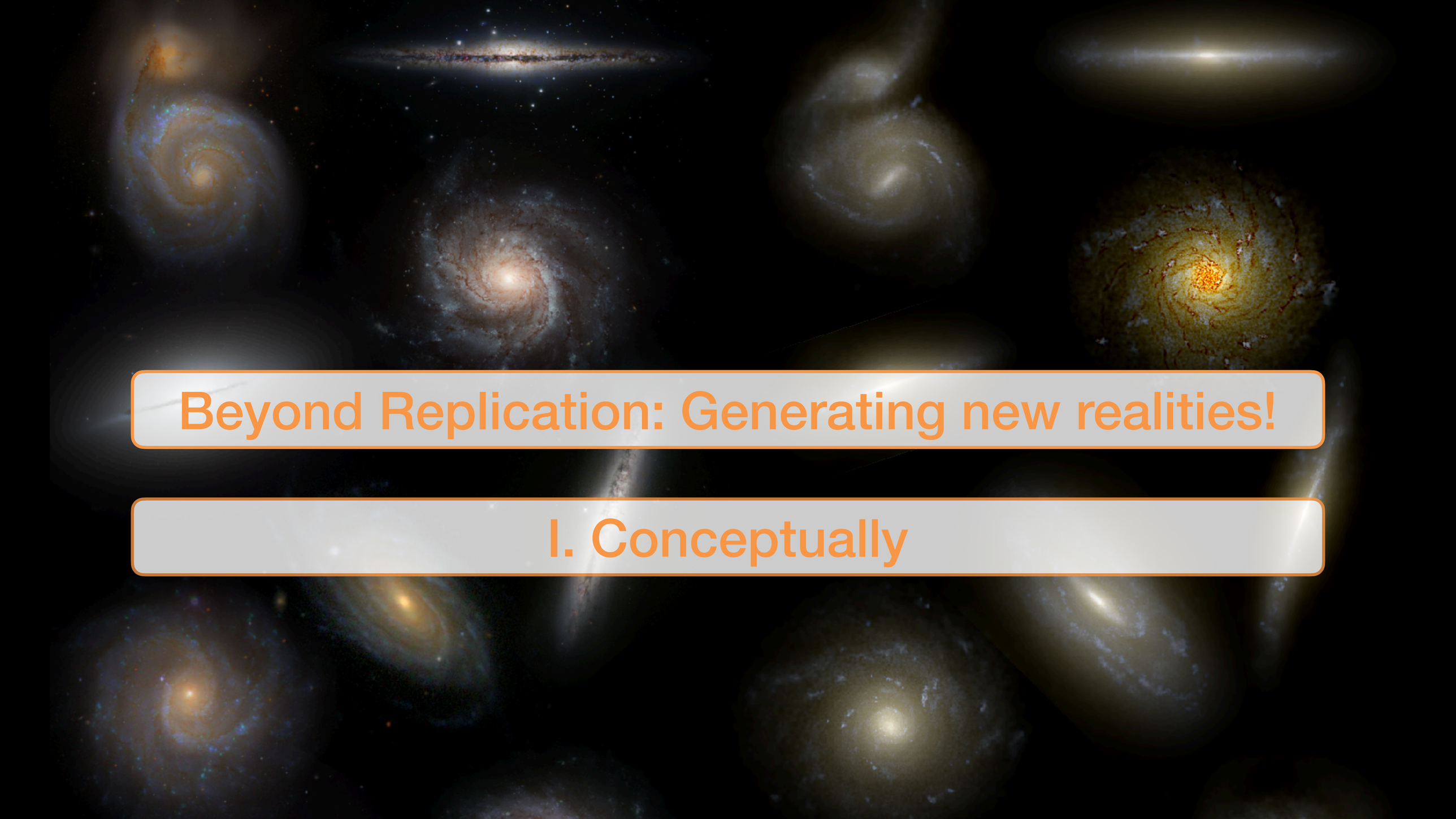
Is lossless compression to a single dimension always what we want?

↓

Strong compression suffers from a lack of interpretability and generalisation

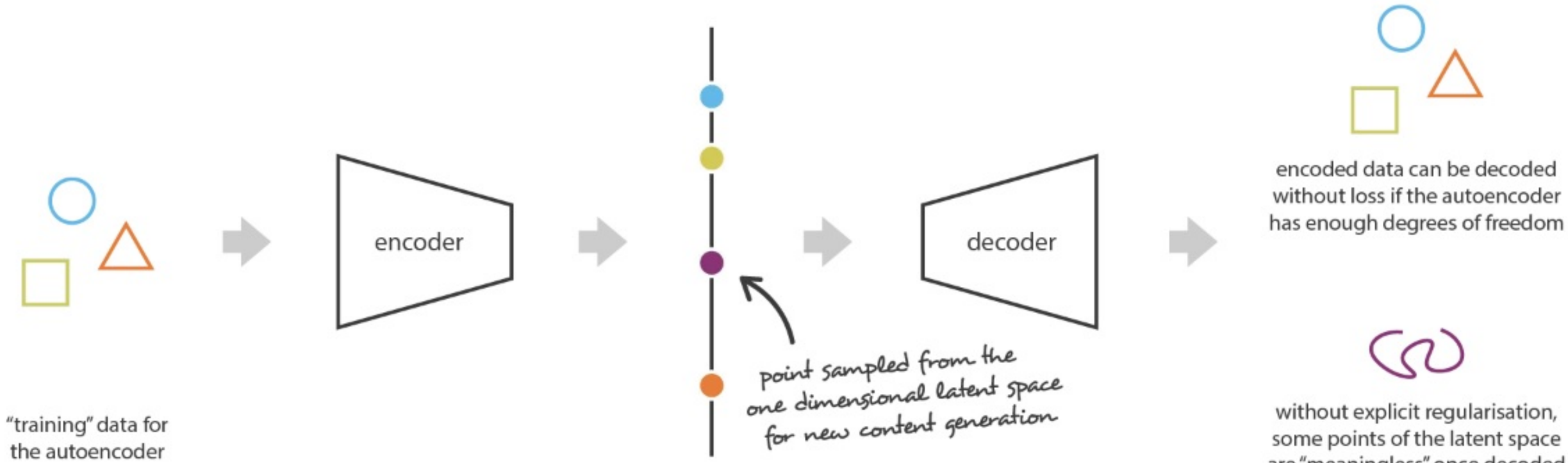
↓

Reduce dimensionality, while also keeping the the data structure interpretable


Beyond Replication: Generating new realities!
I. Conceptually

GENERATION WITH AUTOENCODERS

The Autoencoder is trained to maximise reconstruction, not to result in a meaningful latent space.

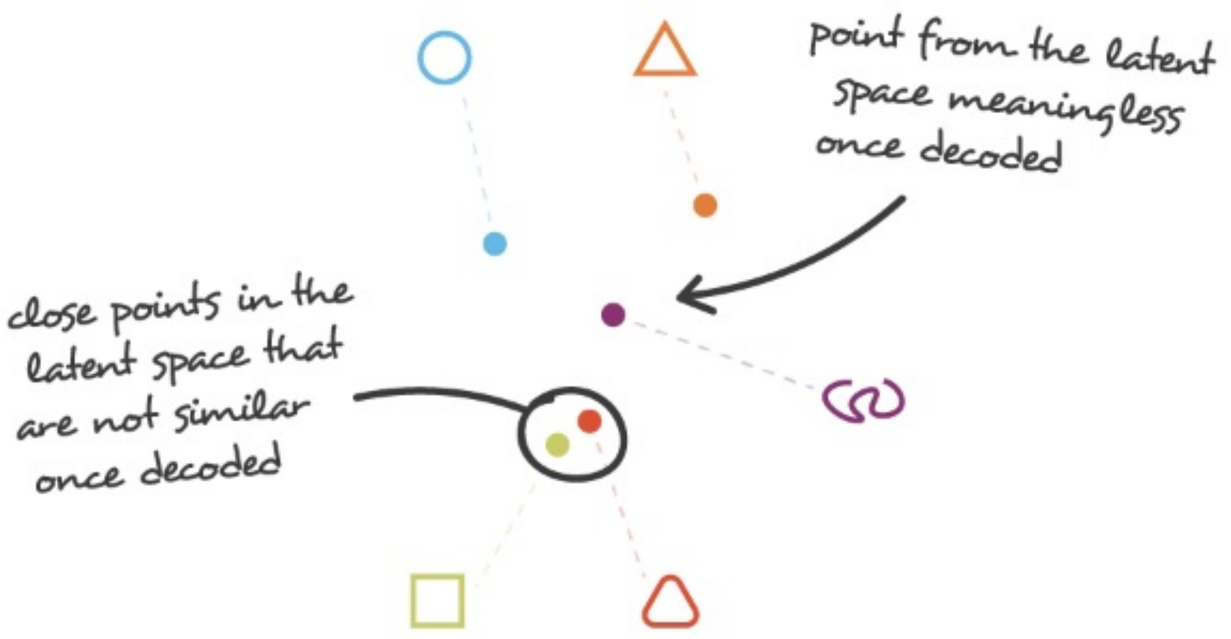


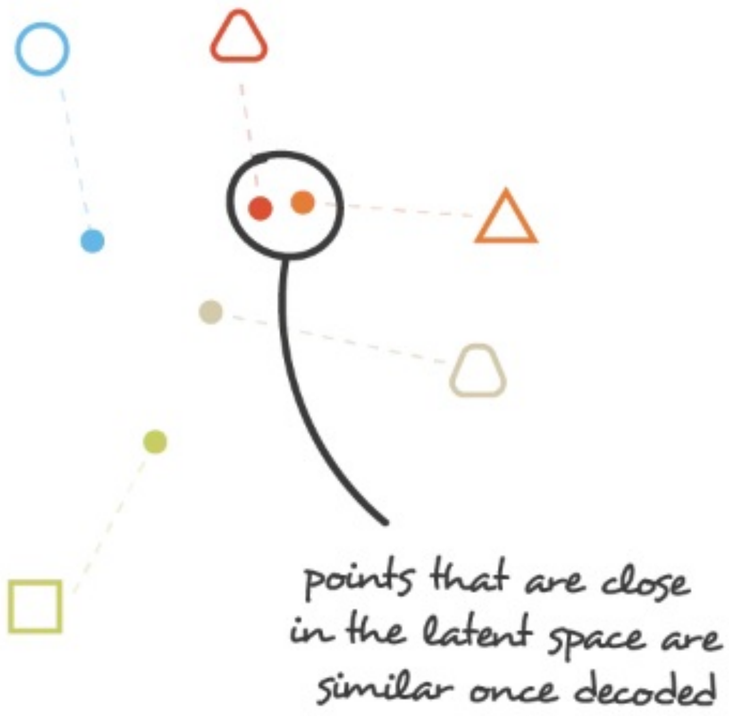
→ Regularization

blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

GENERATION WITH AUTOENCODERS
WHAT DO WE WANT TO ACHIEVE?

Continuity and Completeness





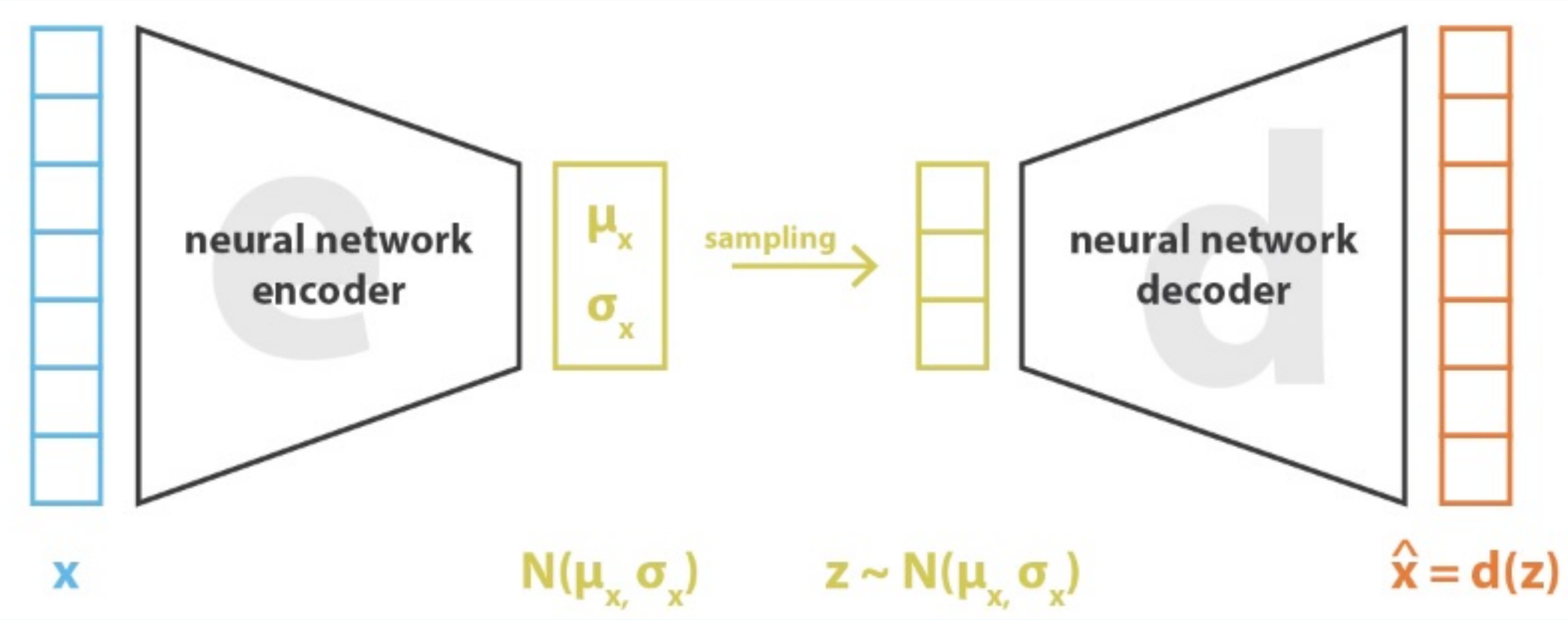
irregular latent space

regular latent space

blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

GENERATION WITH AUTOENCODERS
HOW DO WE ACHIEVE OUR GOALS?

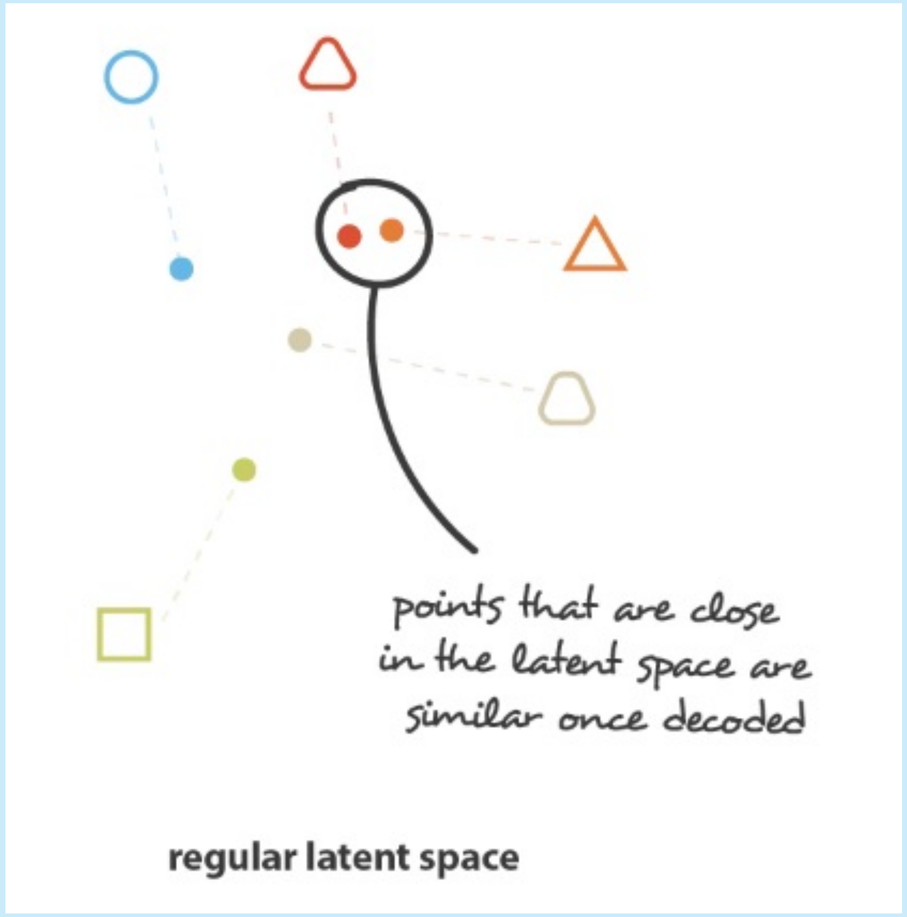
Solution: Encoder gives parameters used for sampling

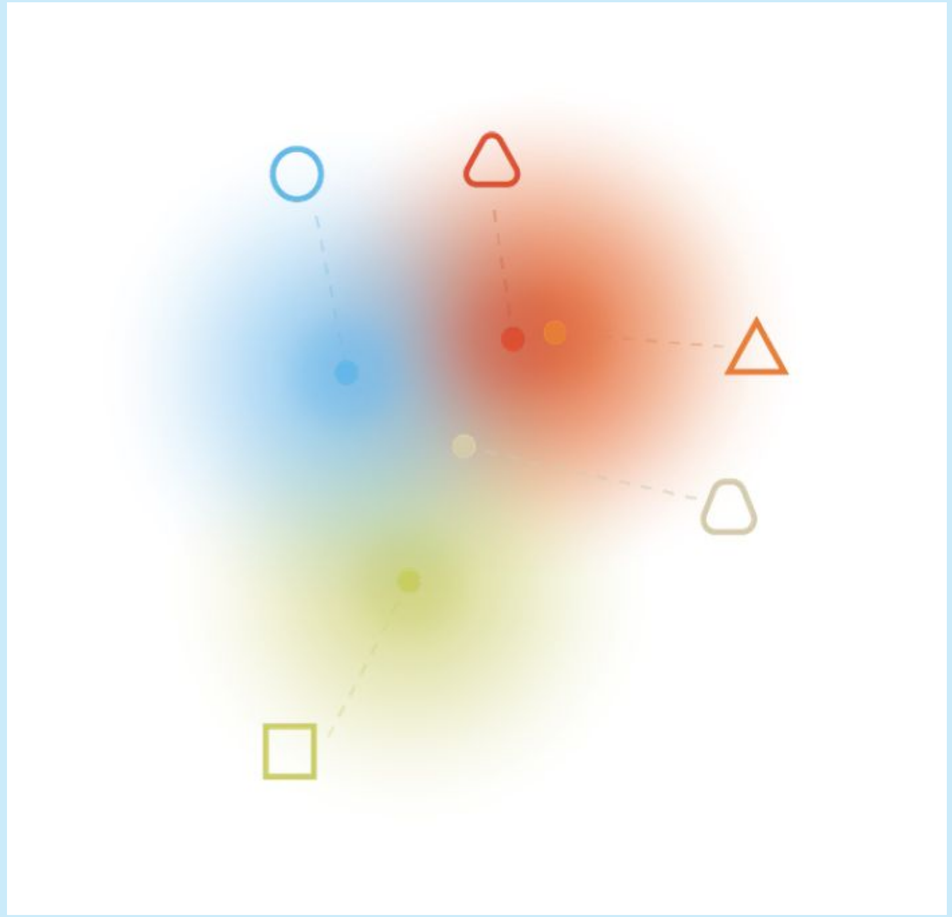


blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

GENERATION WITH AUTOENCODERS
HOW DO WE ACHIEVE OUR GOALS?

Solution: Encoder gives parameters used for sampling

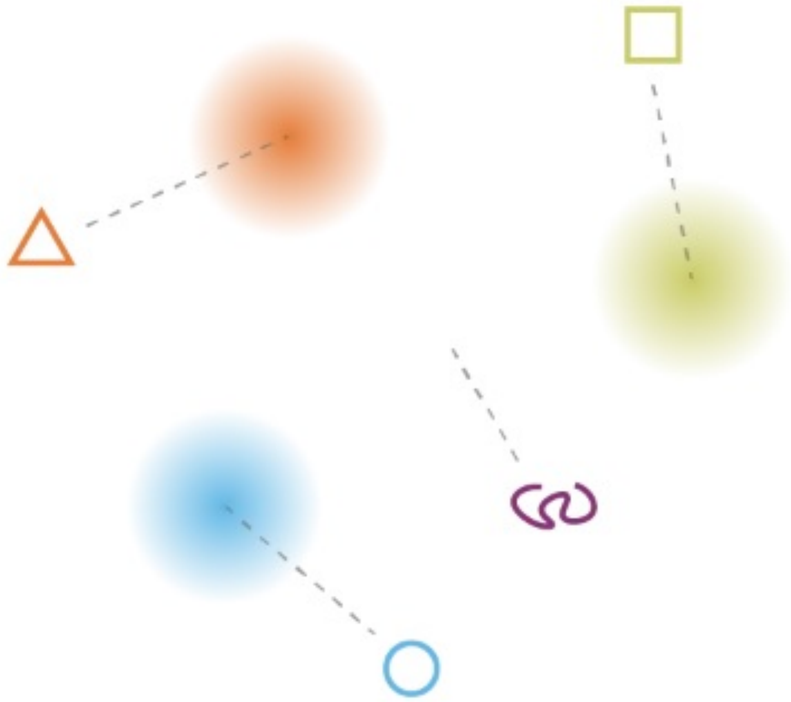




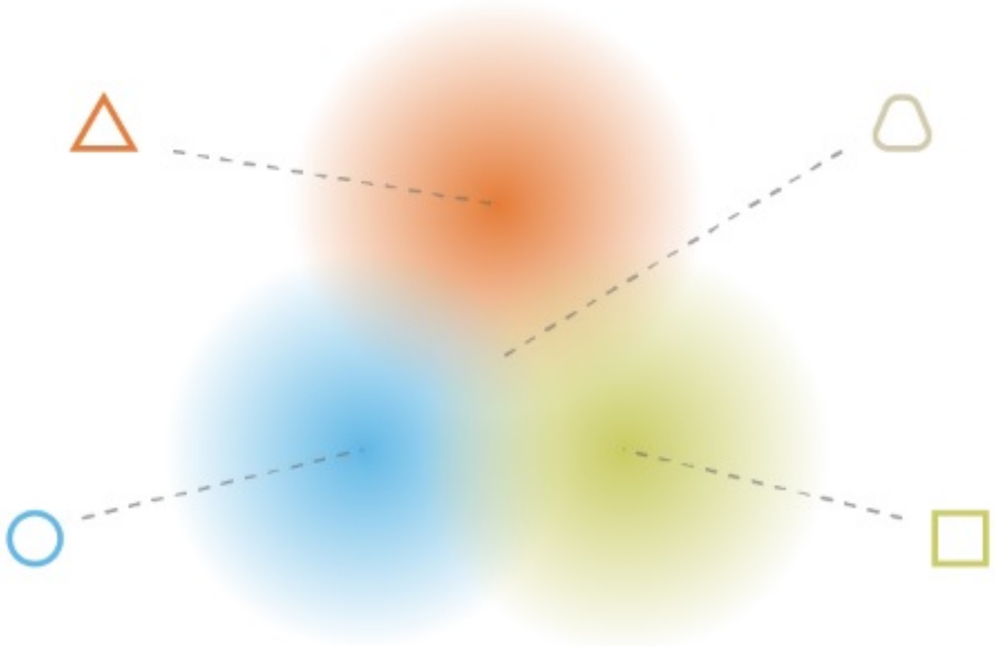
modified from blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

GENERATION WITH AUTOENCODERS
HOW DO WE ACHIEVE OUR GOALS?

Problem: Still only trained for reconstruction…



what can happen without regularisation

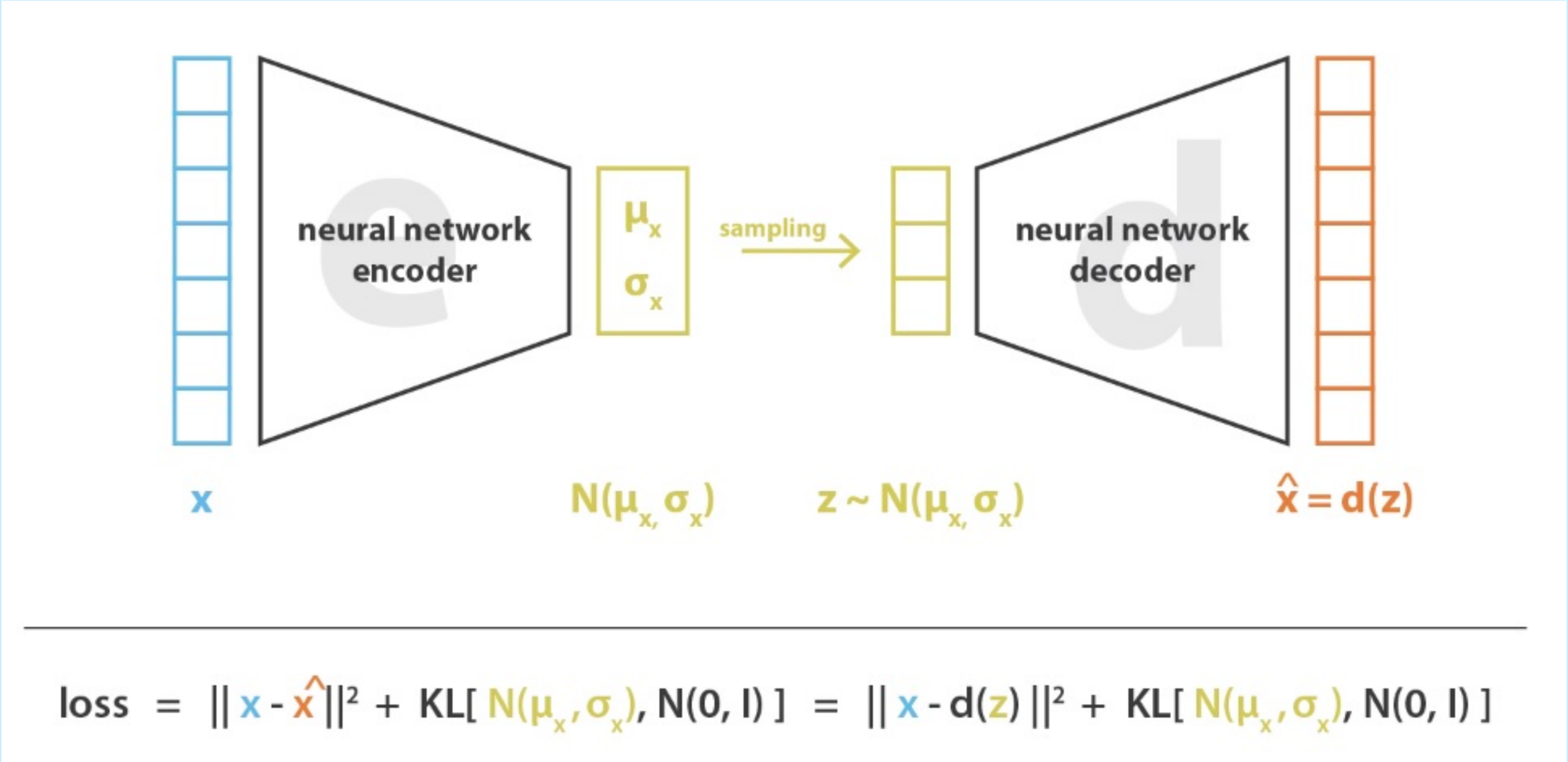


what we want to obtain with regularisation

blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

GENERATION WITH AUTOENCODERS
HOW DO WE ACHIEVE OUR GOALS?

Solution: Adjust the loss term, to enforce a certain distribution

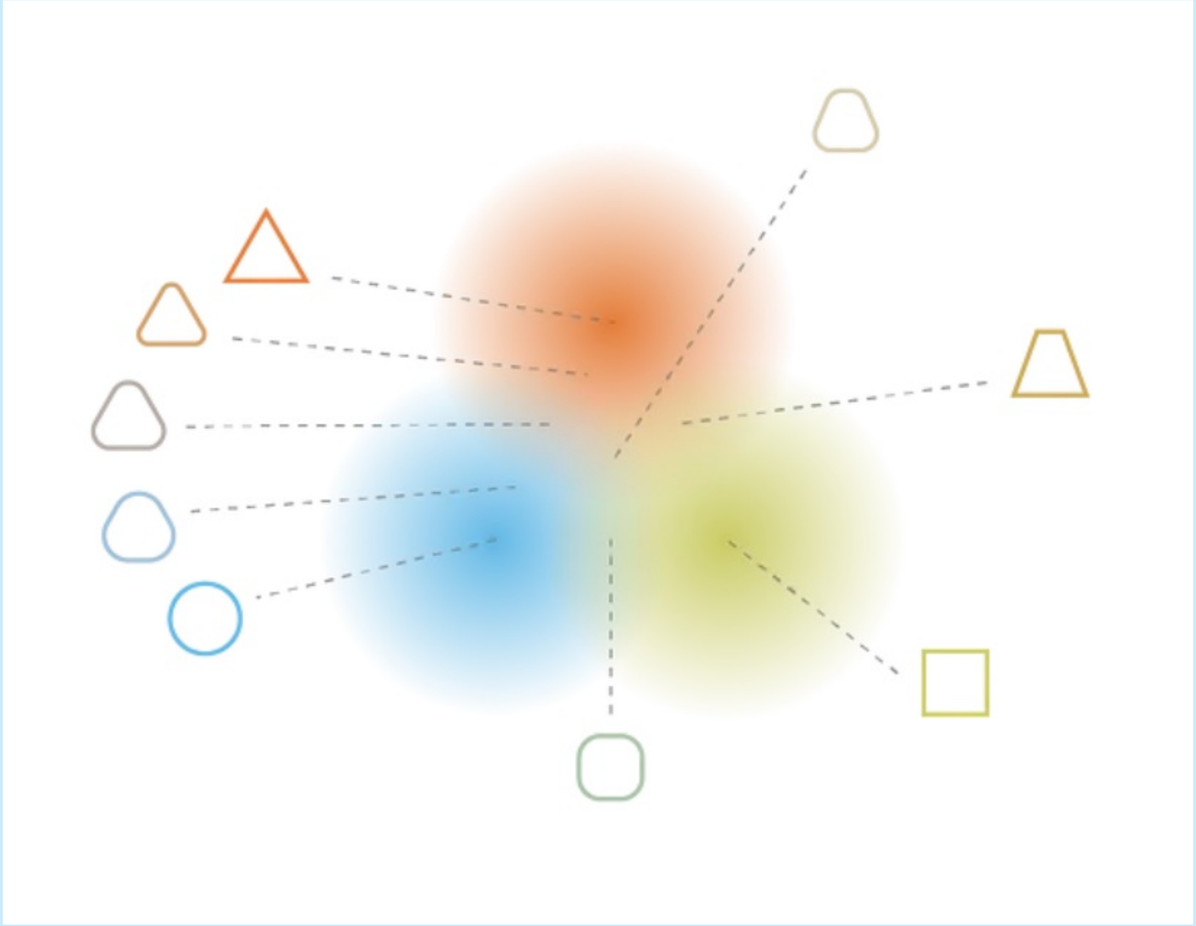


blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

GENERATION WITH AUTOENCODERS

HOW DO WE ACHIEVE OUR GOALS?

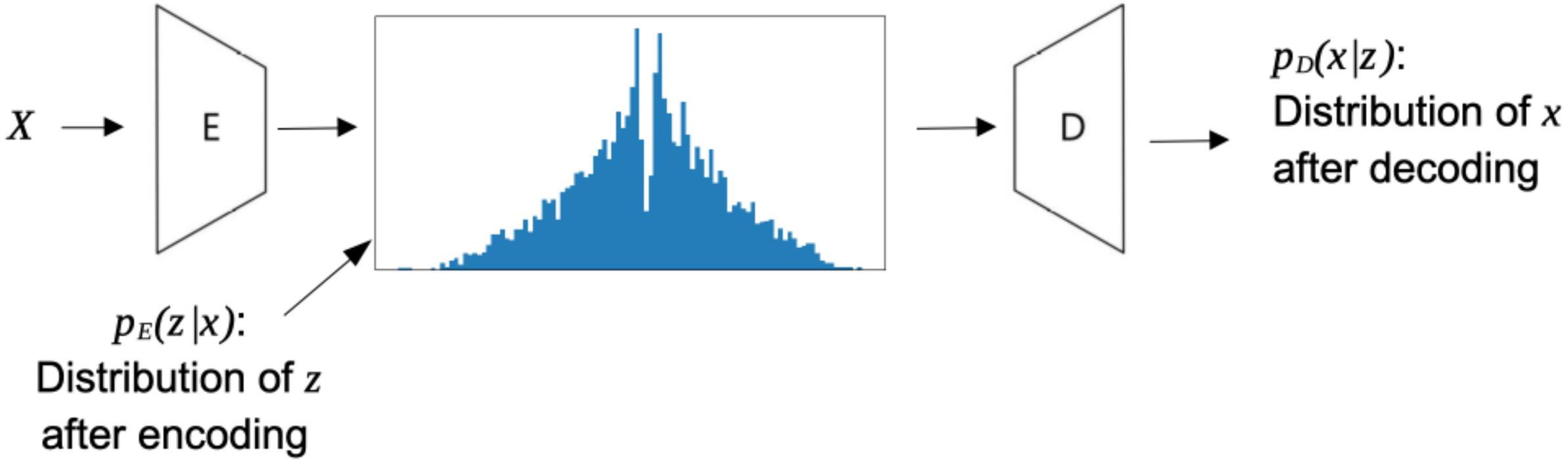
Solution: Adjust the loss term, to enforce a certain distribution



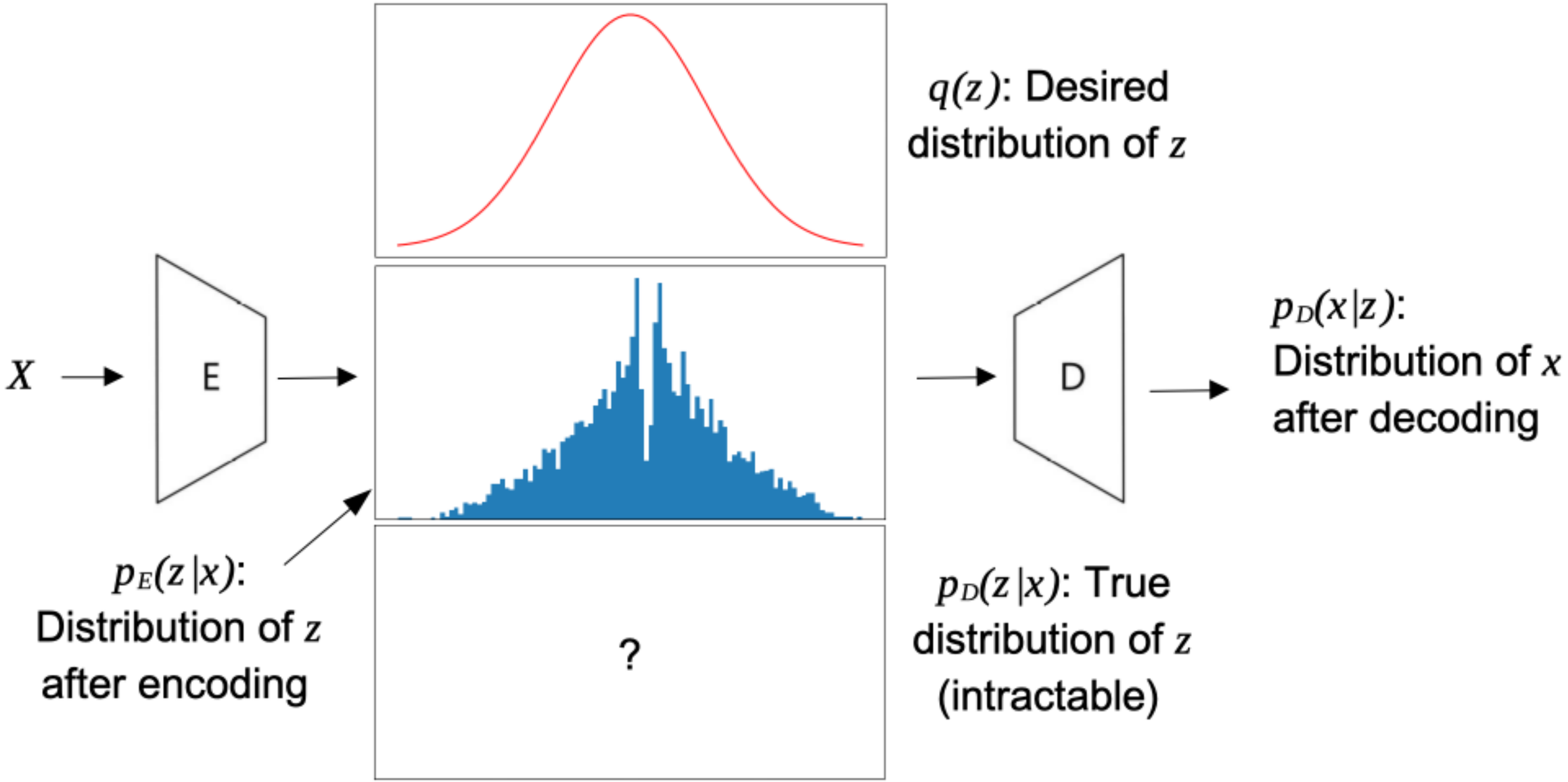
blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

II. Mathematically

VARIATIONAL AUTOENCODERS: BASICS



VARIATIONAL AUTOENCODERS: BASICS



VARIATIONAL AUTOENCODERS: MATHEMATICAL BACKGROUND

Kullback-Leibler (KL) Divergence: a measure for the similarity of two distributions p and q

$$\mathrm{KL}(\mathrm{p}(\mathrm{x})\,||\,\mathrm{q}(\mathrm{x})) = \int_{-\infty}^{+\infty} \mathrm{p}(\mathrm{x})\,\log\left(\frac{\mathrm{p}(\mathrm{x})}{\mathrm{q}(\mathrm{x})}\right)\mathrm{dx}$$

Properties:

Always non-negative, due to Jensen's inequality

Exactly 0 only if the argument of the logarithm is 1, i.e. the two distributions are equal

VARIATIONAL AUTOENCODERS: MATHEMATICAL BACKGROUND

Goal: Find optimal encoder and decoder to minimise the difference between original and approximation:

$$\hat{\mathrm{E}}, \hat{\mathrm{D}} = \arg\min_{\mathrm{E,D}} \left(\mathrm{KL}\left(p^*\left(x\right) || p\left(x\right) \right) \right)$$

VARIATIONAL AUTOENCODERS: MATHEMATICAL BACKGROUND

Goal: Find optimal encoder and decoder to minimise the difference between original and approximation:

$$\hat{\mathrm{E}}, \hat{\mathrm{D}} = \arg\min_{\mathrm{E,D}} \left(\mathrm{KL}\left(p^*\left(x\right) || p\left(x\right)\right)\right)$$

$$\begin{aligned}
\mathrm{KL}\left(p^*\left(x\right) || p\left(x\right)\right) &= \int_{-\infty}^{+\infty} p^*\left(x\right) \log\left(\frac{p^*\left(x\right)}{p\left(x\right)}\right) \mathrm{d}x \\
&= \int_{-\infty}^{\infty} p^*\left(x\right) \log\left(p^*\left(x\right)\right) \mathrm{d}x - \int_{-\infty}^{\infty} p^*\left(x\right) \log\left(p\left(x\right)\right) \mathrm{d}x
\end{aligned}$$

VARIATIONAL AUTOENCODERS: MATHEMATICAL BACKGROUND

Goal: Find optimal encoder and decoder to minimise the difference between original and approximation:

$$\hat{\mathrm{E}}, \hat{\mathrm{D}} = \arg\min_{\mathrm{E,D}} \left(\mathrm{KL}\left(p^*\left(x\right) || p\left(x\right)\right)\right)$$

$$\begin{aligned}\mathrm{KL}\left(p^*\left(x\right) || p\left(x\right)\right) &= \int_{-\infty}^{+\infty} p^*\left(x\right) \log\left(\frac{p^*\left(x\right)}{p\left(x\right)}\right) \mathrm{d}x \\ &= \int_{-\infty}^{\infty} p^*\left(x\right) \log\left(p^*\left(x\right)\right) \mathrm{d}x - \int_{-\infty}^{\infty} p^*\left(x\right) \log\left(p\left(x\right)\right) \mathrm{d}x\end{aligned}$$

Independent of model!

$$\mathbb{E}_{x \sim p^*(x)}\left[-\log\left(p\left(x\right)\right)\right]$$

NEED TO MAXIMISE THE LOG-LIKELIHOOD

$$
\begin{aligned}
&\log\left(p\left(x\right)\right) \\
&= \log\left(p\left(x\right)\right) \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right)\,\mathrm{d}z \\
&= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \log\left(p\left(x\right)\right) \mathrm{d}z \\
&= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \log\left(p\left(x\right) \cdot \frac{p_{\mathrm{E}}\left(z|x\right)}{p_{\mathrm{E}}\left(z|x\right)} \cdot \frac{q\left(z\right)}{q\left(z\right)} \cdot \frac{p_{\mathrm{D}}\left(x|z\right)}{p_{\mathrm{D}}\left(x|z\right)}\right) \mathrm{d}z \\
&= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \left[\log\left(\frac{p_{\mathrm{E}}\left(z|x\right)}{q\left(z\right) p_{\mathrm{D}}\left(x|z\right) / p\left(x\right)}\right) - \log\left(\frac{p_{\mathrm{E}}\left(z|x\right)}{q\left(z\right)}\right) + \log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right] \mathrm{d}z \\
&= \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right) || p_{\mathrm{D}}\left(z|x\right)\right) - \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right) || q\left(z\right)\right) + \mathbb{E}_{z \sim p_{\mathrm{E}}\left(z|x\right)}\left[\log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right]
\end{aligned}
$$

NEED TO MAXIMISE THE LOG-LIKELIHOOD

$$\log\left(p\left(x\right)\right)$$

Insert 1

$$= \log\left(p\left(x\right)\right) \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right)\, \mathrm{d}z$$

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \log\left(p\left(x\right)\right) \mathrm{d}z$$

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \log\left(p\left(x\right) \cdot \frac{p_{\mathrm{E}}\left(z|x\right)}{p_{\mathrm{E}}\left(z|x\right)} \cdot \frac{q\left(z\right)}{q\left(z\right)} \cdot \frac{p_{\mathrm{D}}\left(x|z\right)}{p_{\mathrm{D}}\left(x|z\right)}\right) \mathrm{d}z$$

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \left[\log\left(\frac{p_{\mathrm{E}}\left(z|x\right)}{q\left(z\right) p_{\mathrm{D}}\left(x|z\right) / p\left(x\right)}\right) - \log\left(\frac{p_{\mathrm{E}}\left(z|x\right)}{q\left(z\right)}\right) + \log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right] \mathrm{d}z$$

$$= \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right) || p_{\mathrm{D}}\left(z|x\right)\right) - \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right) || q\left(z\right)\right) + \mathbb{E}_{z \sim p_{\mathrm{E}}\left(z|x\right)}\left[\log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right]$$

NEED TO MAXIMISE THE LOG-LIKELIHOOD

$$\log\left(p\left(x\right)\right)$$

Insert 1

$$= \log\left(p\left(x\right)\right) \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right)\, \mathrm{d}z$$

Log independent of z

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \log\left(p\left(x\right)\right) \mathrm{d}z$$

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \log\left(p\left(x\right) \cdot \frac{p_{\mathrm{E}}\left(z|x\right)}{p_{\mathrm{E}}\left(z|x\right)} \cdot \frac{q\left(z\right)}{q\left(z\right)} \cdot \frac{p_{\mathrm{D}}\left(x|z\right)}{p_{\mathrm{D}}\left(x|z\right)}\right) \mathrm{d}z$$

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \left[\log\left(\frac{p_{\mathrm{E}}\left(z|x\right)}{q\left(z\right) p_{\mathrm{D}}\left(x|z\right) / p\left(x\right)}\right) - \log\left(\frac{p_{\mathrm{E}}\left(z|x\right)}{q\left(z\right)}\right) + \log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right] \mathrm{d}z$$

$$= \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right) || p_{\mathrm{D}}\left(z|x\right)\right) - \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right) || q\left(z\right)\right) + \mathbb{E}_{z \sim p_{\mathrm{E}}\left(z|x\right)}\left[\log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right]$$

NEED TO MAXIMISE THE LOG-LIKELIHOOD

$$\log\left(p\left(x\right)\right)$$

Insert 1

$$= \log\left(p\left(x\right)\right) \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right)\, \mathrm{d}z$$

Log independent of z

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \log\left(p\left(x\right)\right) \mathrm{d}z$$

Insert 1 again

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \log\left(p\left(x\right) \cdot \frac{p_{\mathrm{E}}\left(z|x\right)}{p_{\mathrm{E}}\left(z|x\right)} \cdot \frac{q\left(z\right)}{q\left(z\right)} \cdot \frac{p_{\mathrm{D}}\left(x|z\right)}{p_{\mathrm{D}}\left(x|z\right)}\right) \mathrm{d}z$$

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \left[\log\left(\frac{p_{\mathrm{E}}\left(z|x\right)}{q\left(z\right) p_{\mathrm{D}}\left(x|z\right) / p\left(x\right)}\right) - \log\left(\frac{p_{\mathrm{E}}\left(z|x\right)}{q\left(z\right)}\right) + \log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right] \mathrm{d}z$$

$$= \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right) || p_{\mathrm{D}}\left(z|x\right)\right) - \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right) || q\left(z\right)\right) + \mathbb{E}_{z \sim p_{\mathrm{E}}\left(z|x\right)}\left[\log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right]$$

NEED TO MAXIMISE THE LOG-LIKELIHOOD

$$\log\left(p\left(x\right)\right)$$

Insert 1

$$= \log\left(p\left(x\right)\right) \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right)\, \mathrm{d}z$$

Log independent of z

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \log\left(p\left(x\right)\right) \mathrm{d}z$$

Insert 1 again

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \log\left(p\left(x\right) \cdot \frac{p_{\mathrm{E}}\left(z|x\right)}{p_{\mathrm{E}}\left(z|x\right)} \cdot \frac{q\left(z\right)}{q\left(z\right)} \cdot \frac{p_{\mathrm{D}}\left(x|z\right)}{p_{\mathrm{D}}\left(x|z\right)}\right) \mathrm{d}z$$

$$= \int_{-\infty}^{+\infty} p_{\mathrm{E}}\left(z|x\right) \left[\log\left(\frac{p_{\mathrm{E}}\left(z|x\right)}{q\left(z\right) p_{\mathrm{D}}\left(x|z\right) / p\left(x\right)}\right) - \log\left(\frac{p_{\mathrm{E}}\left(z|x\right)}{q\left(z\right)}\right) + \log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right] \mathrm{d}z$$

Bayes Theorem

$$= \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right) || p_{\mathrm{D}}\left(z|x\right)\right) - \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right) || q\left(z\right)\right) + \mathbb{E}_{z \sim p_{\mathrm{E}}\left(z|x\right)}\left[\log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right]$$

DERIVING THE ELBO

$$\log\left(p\left(x\right)\right)$$
$$= \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right)||p_{\mathrm{D}}\left(z|x\right)\right) - \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right)||q\left(z\right)\right) + \mathbb{E}_{z\sim p_{\mathrm{E}}\left(z|x\right)}\left[\log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right]$$

DERIVING THE ELBO

Intractable

$$\log\left(p\left(x\right)\right)$$
$$= \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right)||p_{\mathrm{D}}\left(z|x\right)\right) - \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right)||q\left(z\right)\right) + \mathbb{E}_{z\sim p_{\mathrm{E}}\left(z|x\right)}\left[\log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right]$$

Variational Gap

Evidence Lower Bound (ELBO)

MAXIMISING THE ELBO

$$\mathrm{ELBO} = \mathbb{E}_{z\sim p_{\mathrm{E}}(z|x)}\left[\log\left(p_{\mathrm{D}}\left(x|z\right)\right)\right] - \mathrm{KL}\left(p_{\mathrm{E}}\left(z|x\right)||q\left(z\right)\right)$$

Reconstruction Error:

Original data x should be a highly probable output

Optimum reached for perfect reconstruction without variability, i.e. deterministic model

→ No generative abilities

How close is the latent space distribution to the desired distribution

Optimum reached for z independent of x

→ No reconstruction abilities

EXPLICIT LOSS EXPRESSION FOR VAE

$$\hat{\mathrm{E}}, \hat{\mathrm{D}} = \arg\min_{\mathrm{E,D}} \left(-\mathrm{ELBO} \right)$$

with

$$\mathrm{ELBO} = \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(p_{\mathrm{D}} \left(x|z \right) \right) \right] - \mathrm{KL} \left(p_{\mathrm{E}} \left(z|x \right) || q \left(z \right) \right)$$

EXPLICIT LOSS EXPRESSION FOR VAE

$$\hat{\mathrm{E}}, \hat{\mathrm{D}} = \arg\min_{\mathrm{E,D}} \left(-\mathrm{ELBO} \right)$$ with $$\mathrm{ELBO} = \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(p_{\mathrm{D}} \left(x|z \right) \right) \right] - \mathrm{KL} \left(p_{\mathrm{E}} \left(z|x \right) || q \left(z \right) \right)$$

Assumptions:

Standard Normal Prior: $$q \left(z \right) = \mathcal{N} \left(\mathrm{o}, \mathbb{1} \right)$$

Gaussian Encoder Distribution: $$p_{\mathrm{E}} \left(z|x \right) = \mathcal{N} \left(z|\mu_{\mathrm{E}} \left(x \right), \mathrm{Diag} \left(\sigma \left(x \right)^{2} \right) \right)$$

Gaussian Decoder with Uncertainty: $$p_{\mathrm{D}} \left(x|z \right) = \mathcal{N} \left(x|\mu_{\mathrm{D}} \left(z \right), \beta^{2} \mathbb{1} \right)$$

EXPLICIT LOSS EXPRESSION FOR VAE

$$\hat{\mathrm{E}}, \hat{\mathrm{D}} = \arg\min_{\mathrm{E,D}} \left(-\mathrm{ELBO} \right)$$

with

$$\mathrm{ELBO} = \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(p_{\mathrm{D}} \left(x|z \right) \right) \right] - \mathrm{KL} \left(p_{\mathrm{E}} \left(z|x \right) || q \left(z \right) \right)$$

$$\begin{aligned} \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(p_{\mathrm{D}} \left(x|z \right) \right) \right] &= \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(\mathcal{N} \left(x|\mu_{\mathrm{D}} \left(z \right), \beta^2 \mathbb{1} \right) \right) \right] \\ &= \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[-\frac{\left(x - \mu_{\mathrm{D}} \left(z \right) \right)^2}{2\beta^2} + \log \left(2\pi\beta^2 \right)^{d/2} \right] \end{aligned}$$

EXPLICIT LOSS EXPRESSION FOR VAE

$$\hat{\mathrm{E}}, \hat{\mathrm{D}} = \arg\min_{\mathrm{E,D}} \left(-\mathrm{ELBO} \right)$$

with

$$\mathrm{ELBO} = \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(p_{\mathrm{D}} \left(x|z \right) \right) \right] - \mathrm{KL} \left(p_{\mathrm{E}} \left(z|x \right) || q \left(z \right) \right)$$

$$\begin{aligned} \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(p_{\mathrm{D}} \left(x|z \right) \right) \right] &= \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(\mathcal{N} \left(x|\mu_{\mathrm{D}} \left(z \right), \beta^2 \mathbb{1} \right) \right) \right] \\ &= \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[-\frac{\left(x - \mu_{\mathrm{D}} \left(z \right) \right)^2}{2\beta^2} + \log \left(2\pi\beta^2 \right)^{d/2} \right] \end{aligned}$$

$$\mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(p_{\mathrm{D}} \left(x|z \right) \right) \right] = -\frac{1}{M} \sum_{k=1}^{M} \left(\frac{\left(x_k - \mu_{\mathrm{D,\,k}} \left(z \right) \right)^2}{2\beta^2} \right)$$

EXPLICIT LOSS EXPRESSION FOR VAE

$$\hat{\mathrm{E}}, \hat{\mathrm{D}} = \arg\min_{\mathrm{E,D}} \left(-\mathrm{ELBO} \right)$$

with

$$\mathrm{ELBO} = \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(p_{\mathrm{D}} \left(x|z \right) \right) \right] - \mathrm{KL} \left(p_{\mathrm{E}} \left(z|x \right) || q \left(z \right) \right)$$

$$\begin{aligned} -\mathrm{KL} \left(p_{\mathrm{E}} \left(z|x \right) || q \left(z \right) \right) &= -\mathrm{KL} \left(\mathcal{N} \left(z|\mu_{\mathrm{E}} \left(x \right), \mathrm{Diag} \left(\sigma \left(x \right)^2 \right) \right) || \mathcal{N} \left(0, \mathbb{1} \right) \right) \\ &= -\frac{1}{2} \sum_{j=1}^{d} \left(\mu_j \left(x \right)^2 + \sigma_j \left(x \right)^2 - \log \left(\sigma_j \left(x \right)^2 \right) - 1 \right) \end{aligned}$$

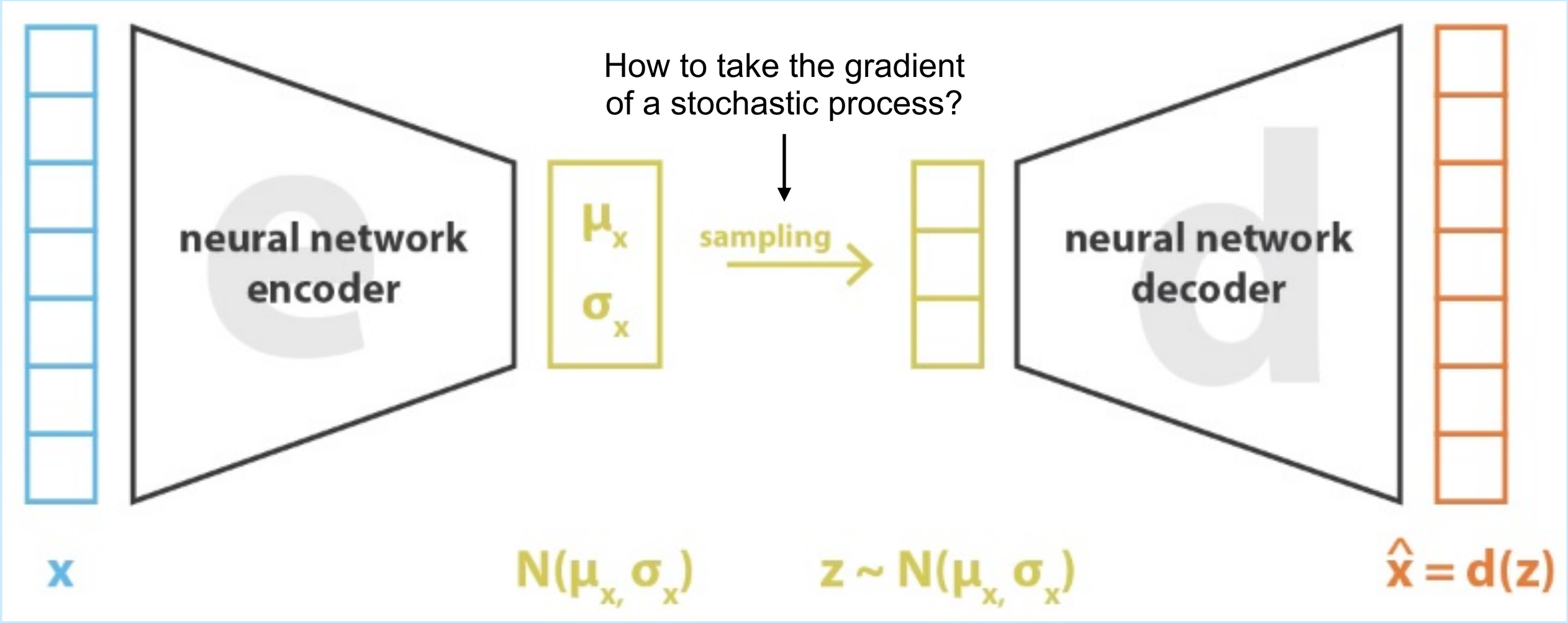
EXPLICIT LOSS EXPRESSION FOR VAE

$$\hat{\mathrm{E}}, \hat{\mathrm{D}} = \arg\min_{\mathrm{E,D}} \left(-\mathrm{ELBO} \right)$$ with $$\mathrm{ELBO} = \mathbb{E}_{z \sim p_{\mathrm{E}}(z|x)} \left[\log \left(p_{\mathrm{D}} \left(x|z \right) \right) \right] - \mathrm{KL} \left(p_{\mathrm{E}} \left(z|x \right) || q \left(z \right) \right)$$

Loss Function for training a VAE

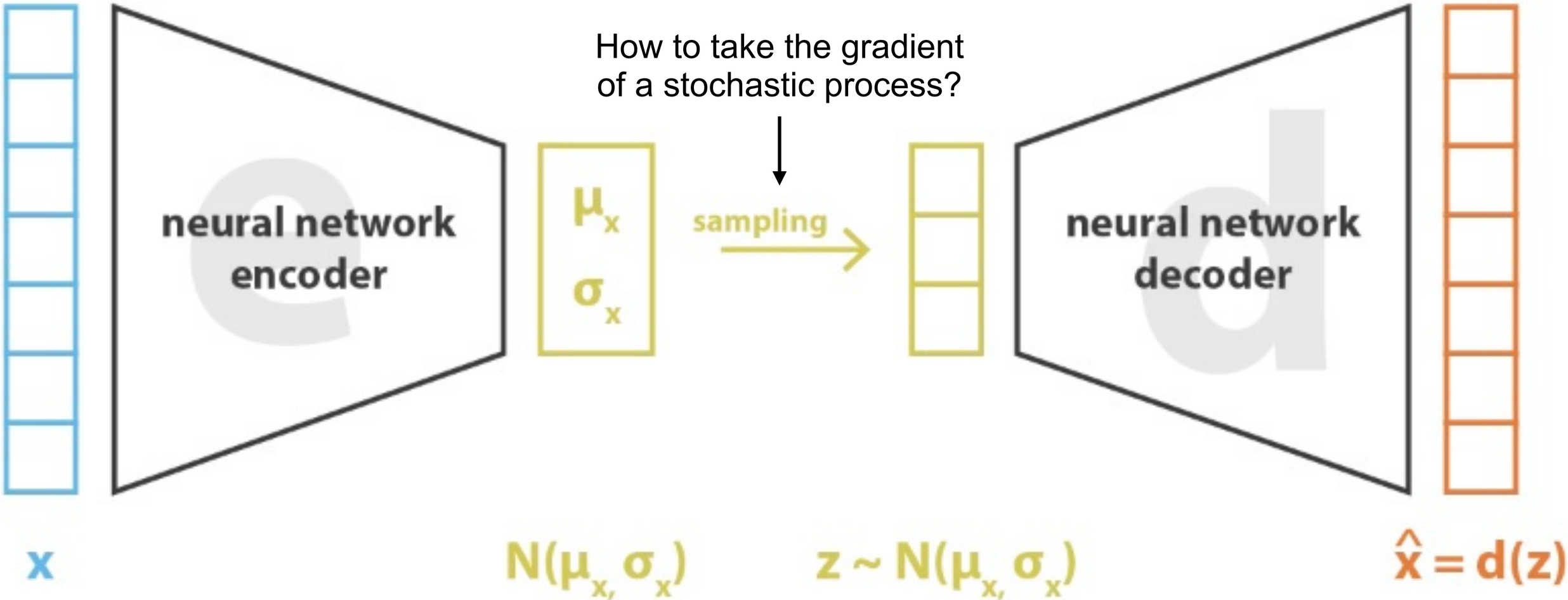
$$-\mathrm{ELBO} = \frac{1}{2\beta^2} \cdot \frac{1}{M} \sum_{k=1}^{M} \left(\left(x_k - \mu_{\mathrm{D,\,k}} \left(z \right) \right)^2 \right) + \frac{1}{2} \sum_{j=1}^{d} \left(\mu_j \left(x \right)^2 + \sigma_j \left(x \right)^2 - \log \left(\sigma_j \left(x \right)^2 \right) - 1 \right)$$

BUT… ONE MORE PROBLEM



blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

BUT… ONE MORE PROBLEM

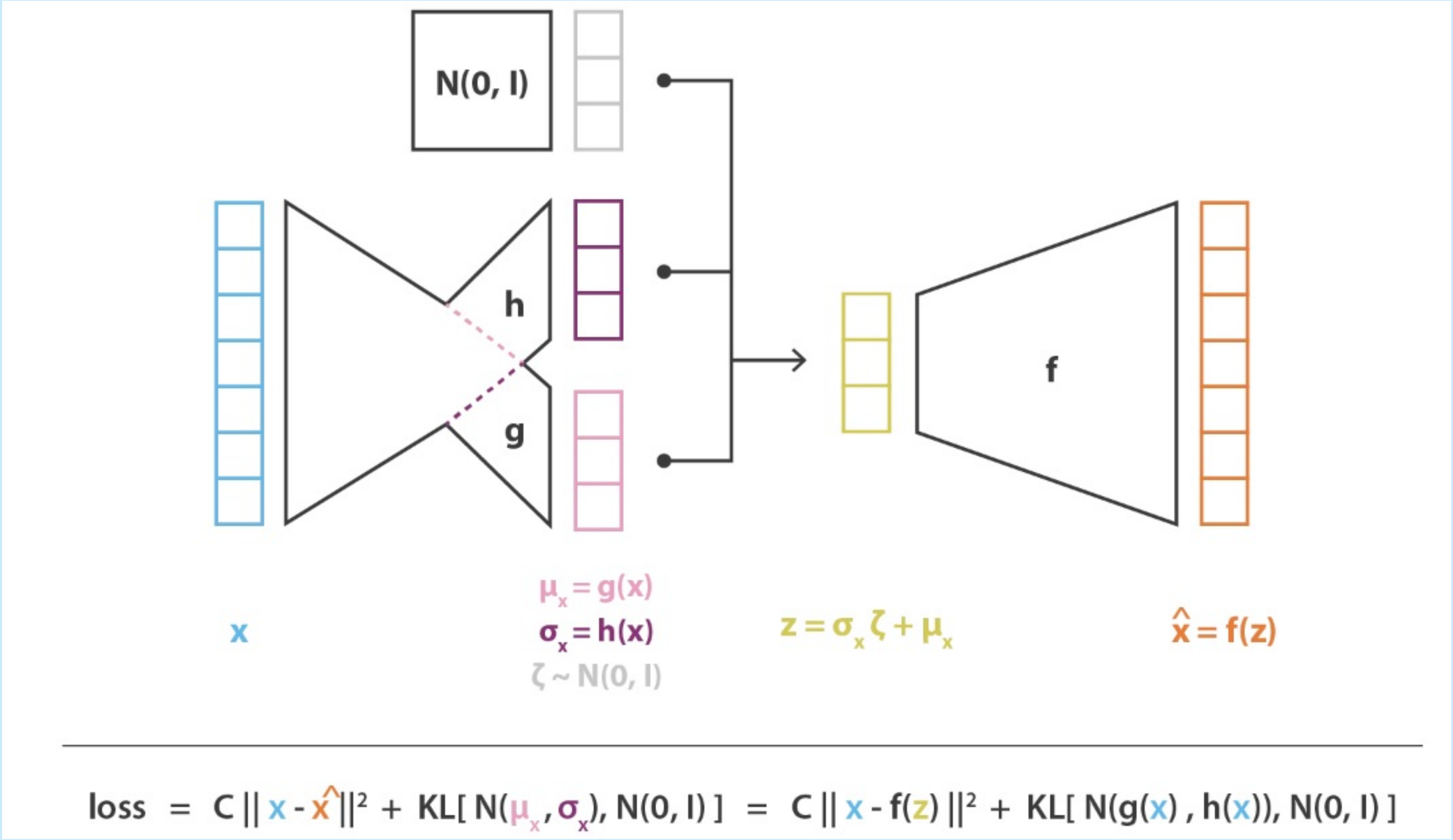


Reparameterisation Trick:

$$z \sim \mathcal{N}(\mu, \sigma) \rightarrow z = \mu + \zeta \cdot \sigma \text{ with } \zeta \sim \mathcal{N}(0, 1)$$

blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

BUT… ONE MORE PROBLEM



$$\text{loss} = C \,||\, x - \hat{x} \,||^2 + KL[\, N(\mu_x, \sigma_x), N(0, I) \,] = C \,||\, x - f(z) \,||^2 + KL[\, N(g(x), h(x)), N(0, I) \,]$$

blog post by https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

III. Practically: Code examples

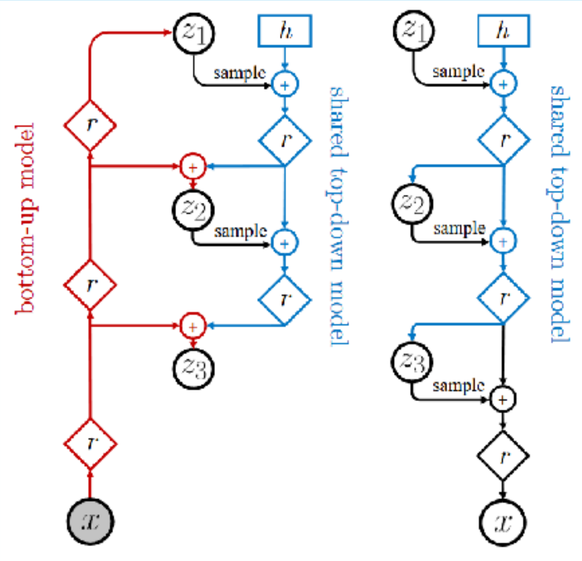
DIFFERENT VAE FLAVOURS

1. AVAE (Adversarial Variational Autoencoder)
   → Combines VAEs and GANs to minimise problems with each model
   → Avoid GAN model collapse with latent space
   → Create more realistic samples with additional “realism” loss

2. β-VAE

3. k-spares VAE

4. NVEA (Nouveau Variational Autoencoder)



Astro Examples of VAE

DIFFERENT VAE APPLICATIONS IN ASTRO

1. Dimensionality Reduction of SDSS Spectra with Variational Autoencoders (Portillo+2020)

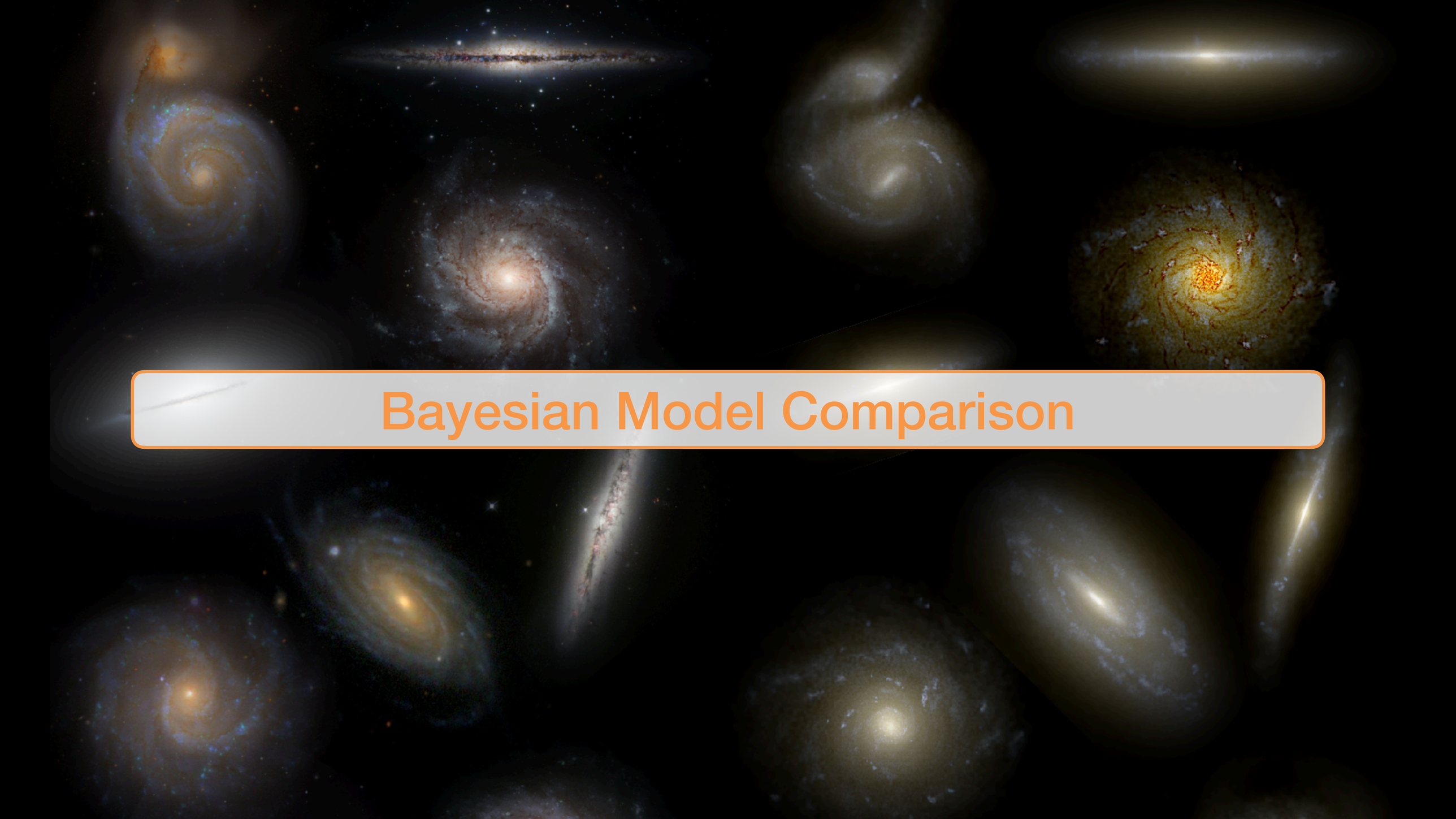
2. Deblending galaxies with variational autoencoders: A joint multiband, multi-instrument approach (Arcelin+2020)

3. Bayesian decomposition of the Galactic multi-frequency sky using probabilistic autoencoders (Milosevic+2021)

4. Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy (Gabbard+2022)

5. PELS-VAE: Physics-Enhanced Latent Space Variational Autoencoder for physically informed light curves of periodic variable stars (Martínez-Palomera+2020)

5. Model comparison and model misspecification for galaxy simulations (Zhou+2024 subm.)



Bayesian Model Comparison

MODEL COMPARISON & MODEL MISSPECIFICATION

Bayesian model selection assigns posterior probabilities $\mathrm{p}(\mathcal{M}_k \,|\, \mathbf{d})$ to models $\mathcal{M}_k \in \{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$ (instead of to values of their parameters $\boldsymbol{\theta}_k$), conditional on observed data $\mathbf{d}$. The conventional approach is to compute the marginal likelihood (or *evidence*) $\mathrm{p}(\mathbf{d} \,|\, \mathcal{M}_k)$, which is the average likelihood $\mathrm{p}(\mathbf{d} \,|\, \boldsymbol{\theta}_k)$ of parameters distributed according to the prior $\mathrm{p}(\boldsymbol{\theta}_k)$:

$$\mathrm{p}(\mathbf{d} \,|\, \mathcal{M}_k) = \int \mathrm{p}(\mathbf{d} \,|\, \boldsymbol{\theta}_k)\, \mathrm{p}(\boldsymbol{\theta}_k)\, \mathrm{d}\boldsymbol{\theta}_k \tag{1}$$

(where the presence of $\mathcal{M}_k$'s parameters $\boldsymbol{\theta}_k$ implies conditioning on $\mathcal{M}_k$ in the right-hand side). The prior belief in the model, $\mathrm{p}(\mathcal{M}_k)$, is then updated to its posterior probability in accordance with Bayes' theorem: $\mathrm{p}(\mathcal{M}_k \,|\, \mathbf{d}) \propto \mathrm{p}(\mathbf{d} \,|\, \mathcal{M}_k)\, \mathrm{p}(\mathcal{M}_k)$, normalised over all models considered.

MODEL COMPARISON & MODEL MISSPECIFICATION

Methods

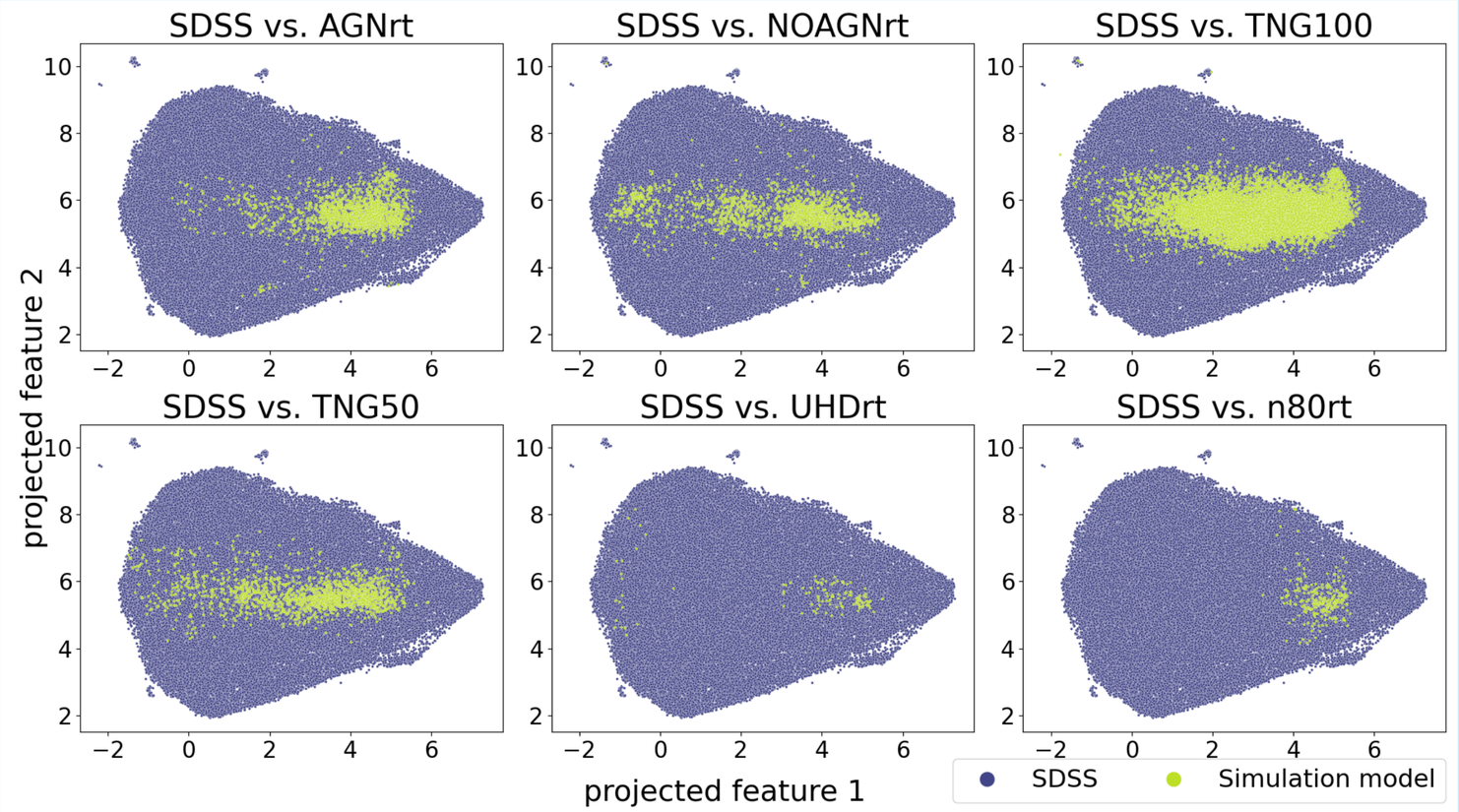
- Jeffrey+Wandelt 2023: loss functions for two-way model comparison with an emphasis on recovering accurate extreme Bayes factors
- Radev+2021: estimate a Dirichlet distribution over an arbitrary number of models using a NN and variational optimisation
- Elsemüller+2023 and Karchev+2024: use a cross-entropy loss for multi-class posterior probabilities
- Macciò+2022: Model selection for star formation prescriptions in cosmo sims
- Zhou+2024: Model misspecification plus model comparison for low simulation budget applications
- Jin+2024: Model comparison of cosmo sims via GANomaly scores

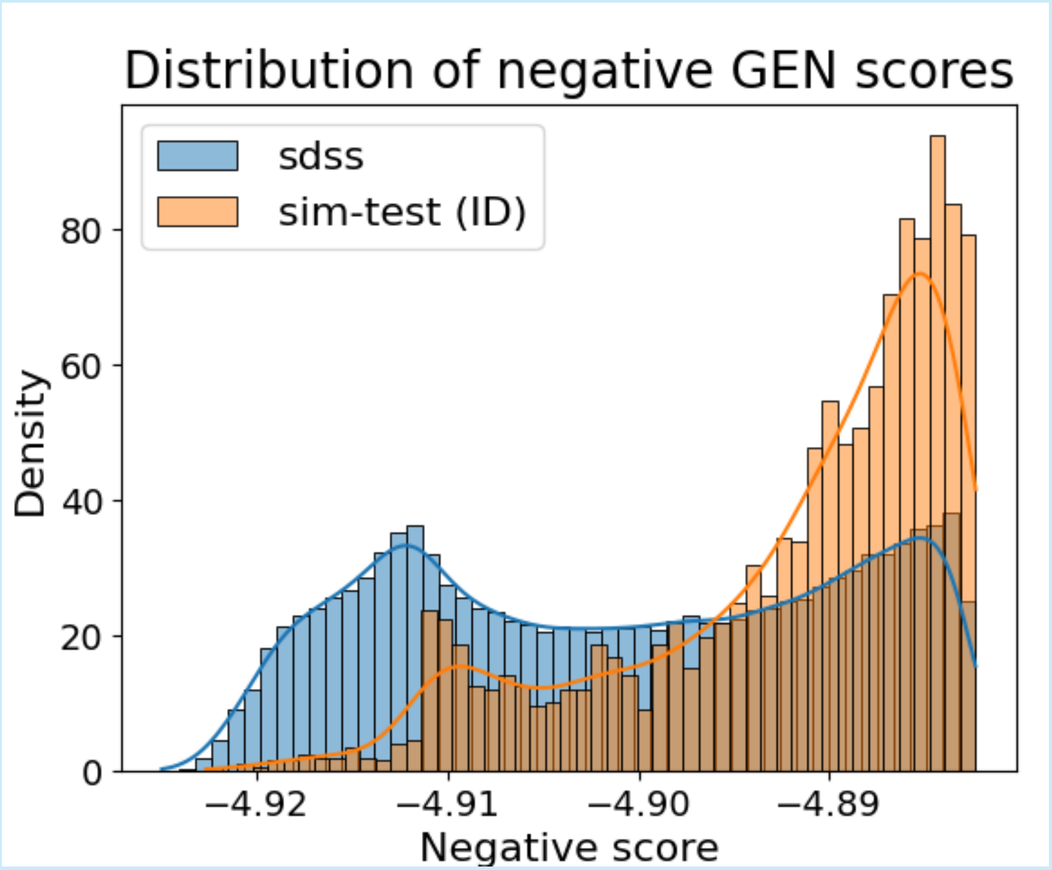
MODEL COMPARISON & MODEL MISSPECIFICATION

Idea:

- Train embedding network on ~600.000 SDSS images, then encode simulated SDSS images
- Train simulation classifiers on embeddings, apply to real SDSS images

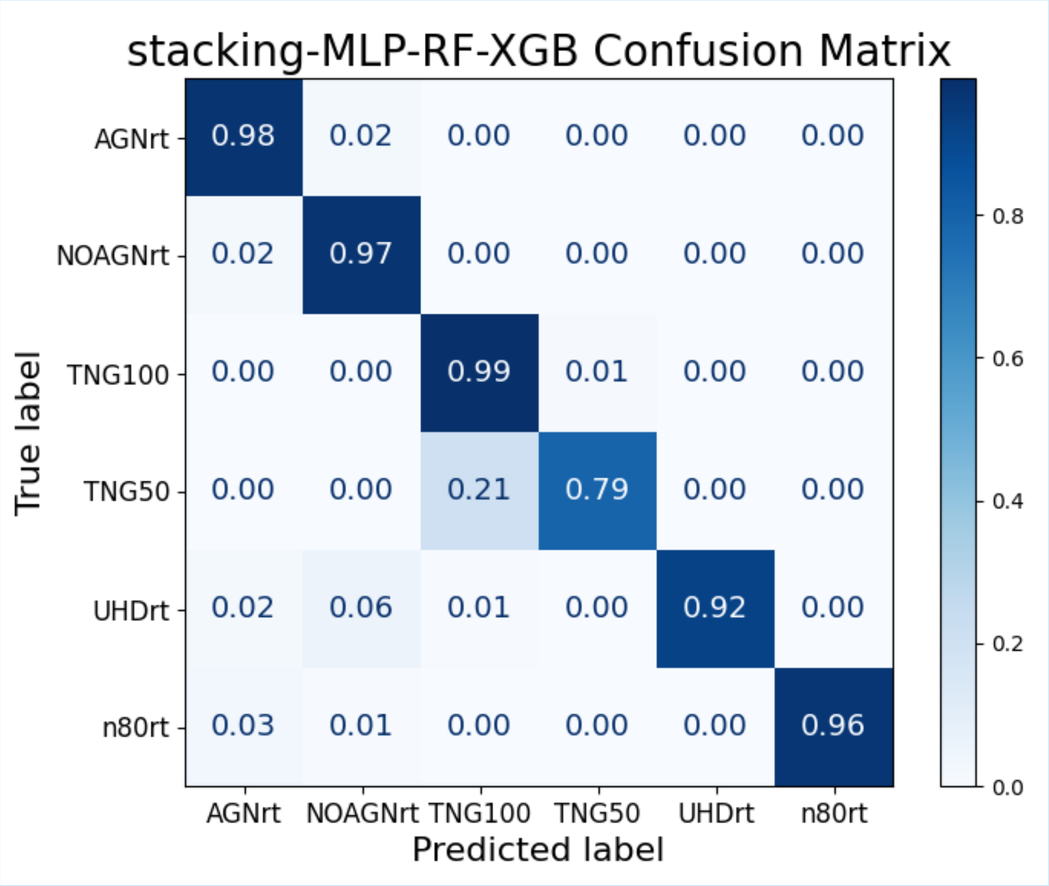
UMAP projection of k-sparse encoding of sims and Jobs

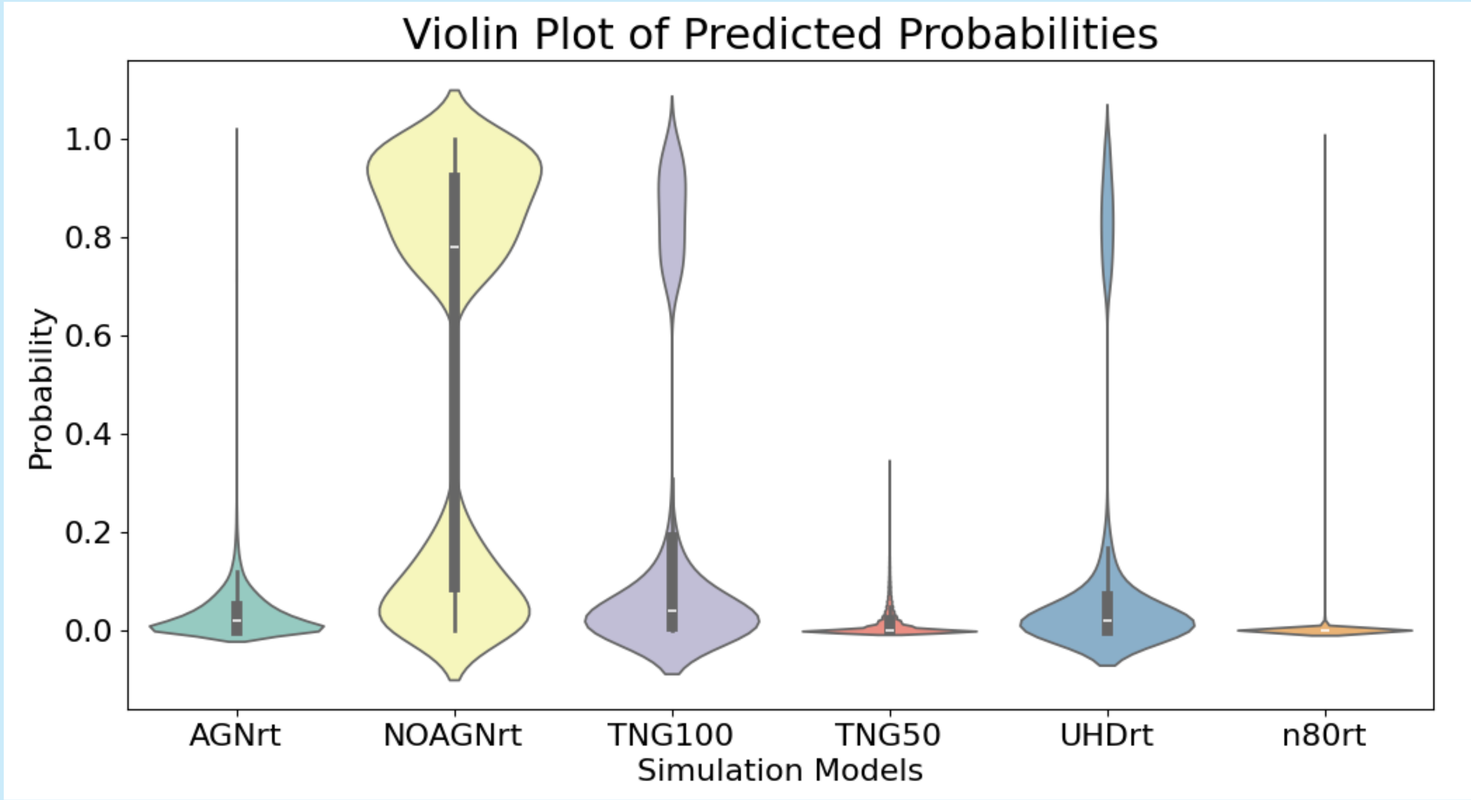




Zhou,Buck+2024 subm.

MODEL COMPARISON & MODEL MISSPECIFICATION



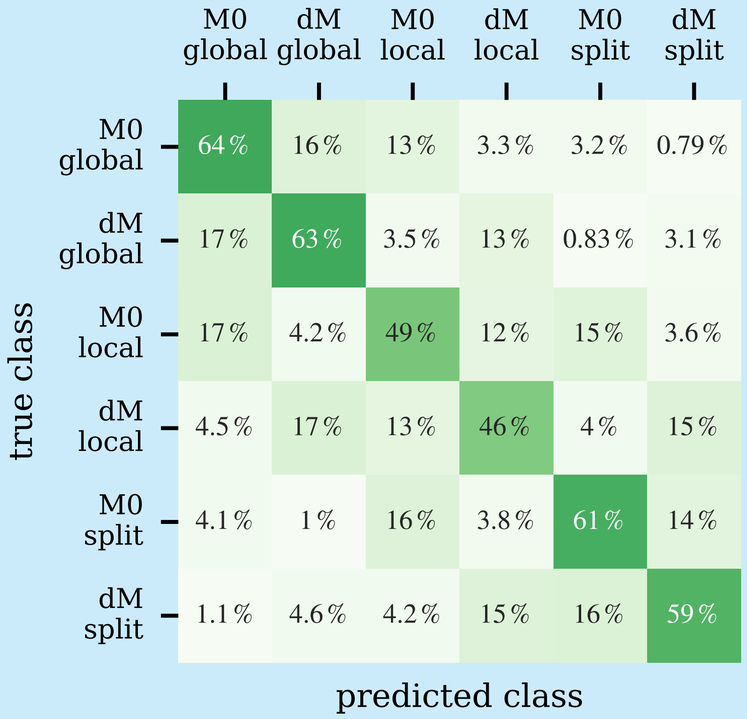


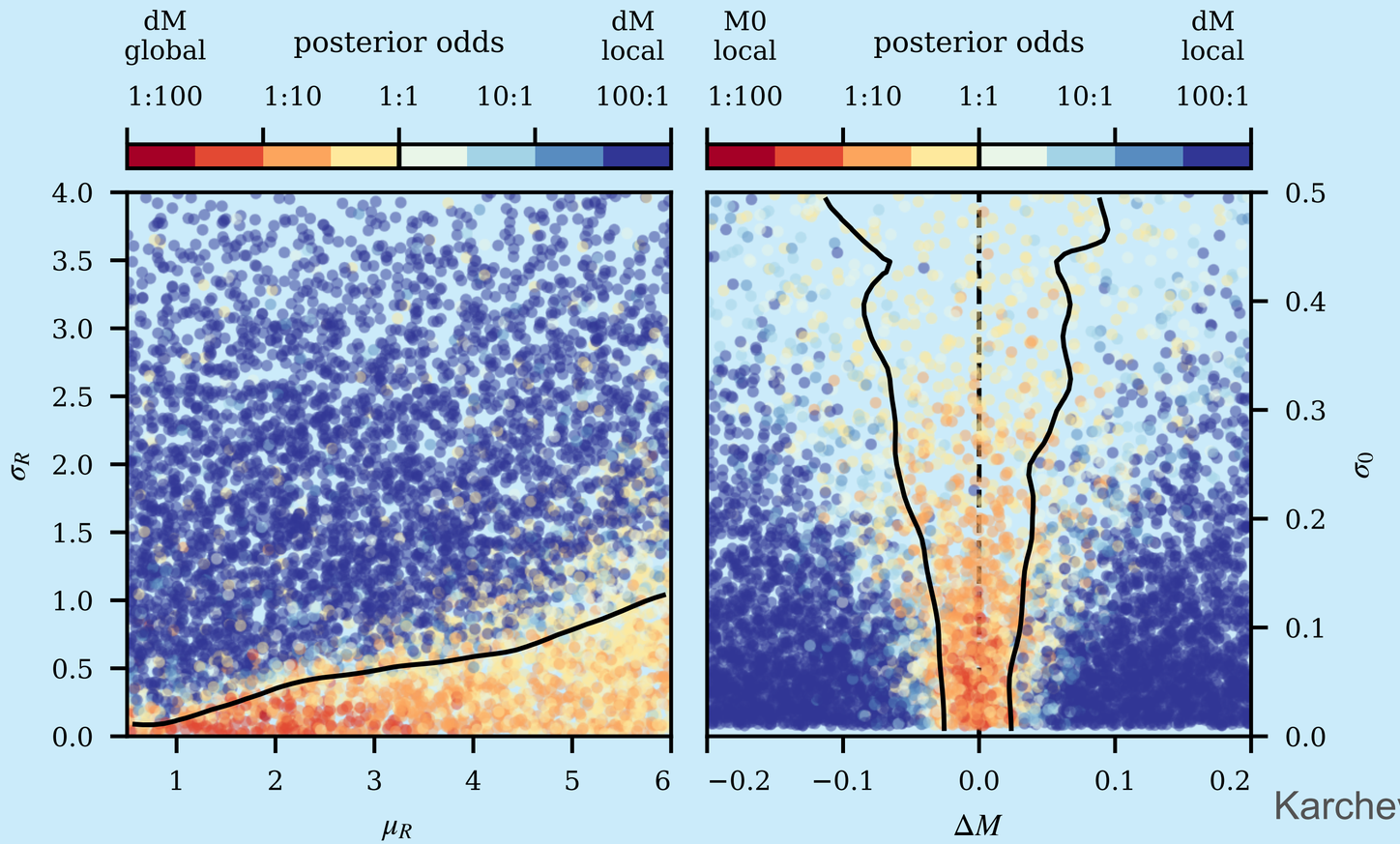
Zhou,Buck+2024 subm.

MODEL COMPARISON/SELECTION

SimSIMS: Simulation-based Supernova Ia Model Selection with thousands of latent variables

Konstantin Karchev[1] **Roberto Trotta**[1,2,3,4] **Christoph Weniger**[5]





Karchev+2024

WHERE TO FIND THE LECTURE MATERIAL

https://github.com/TobiBu/graddays


Score Matching and diffusion models

SCORE MATCHING AND DIFFUSION MODELS

What is the score $s(\mathbf{x})$ of a pdf $p(\mathbf{x})$?

$$s(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

Gibbs measure: $$p(x) = \frac{1}{\mathcal{Z}} \exp(-\beta E(x))$$

$$\log p(x) = -\beta E(x) - \log \mathcal{Z}$$

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = -\beta \nabla_x E(x)$$

SCORE MATCHING AND DIFFUSION MODELS

What is the score $s(\mathbf{x})$ of a pdf $p(\mathbf{x})$?

$$s(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

Gibbs measure: $$p(x) = \frac{1}{\mathcal{Z}} \exp(-\beta E(x))$$

$$\log p(x) = -\beta E(x) - \log \mathcal{Z}$$

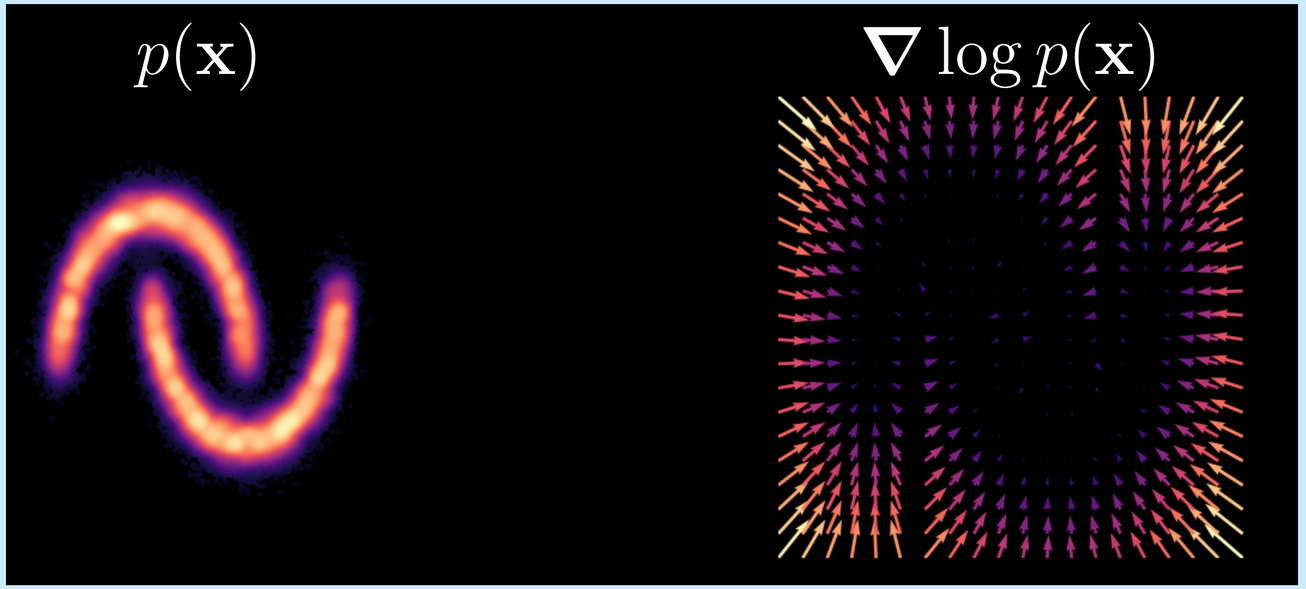
$$\boxed{\nabla_{\mathbf{x}} \log p(\mathbf{x}) = -\beta \nabla_x E(x)}$$

local and easier to learn

SCORE MATCHING AND DIFFUSION MODELS

What is the score $s(\mathbf{x})$ of a pdf $p(\mathbf{x})$?

$$s(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})$$



Idea: Learn $p(\mathbf{x})$ solely from data samples, then sample new instances.

blog post by https://yang-song.net/blog/2021/score/
and https://jmtomczak.github.io/blog/16/16_score_matching.html

TRAINING OBJECTIVE FOR SCORE MATCHING

train via Fisher divergence

$$\mathbb{E}_{p(x)}[\,||\nabla_{\mathbf{x}}\log p(\mathbf{x}) - s_\theta(x)||_2^2\,]$$

however, this needs access to the data score $\nabla_x \log p(x)$

score matching circumvents this!

additionally, score matching does not require $s_\theta(x)$ to be an actual score function of any normalised distribution.

only requirement: vector valued function of same dimensional input/output

But how does this work?

TRAINING OBJECTIVE FOR SCORE MATCHING

train via Fisher divergence

$$\mathbb{E}_{p(x)}[\,||\,\nabla_{\mathbf{x}}\log p(\mathbf{x}) - s_\theta(x)\,||_2^2]$$

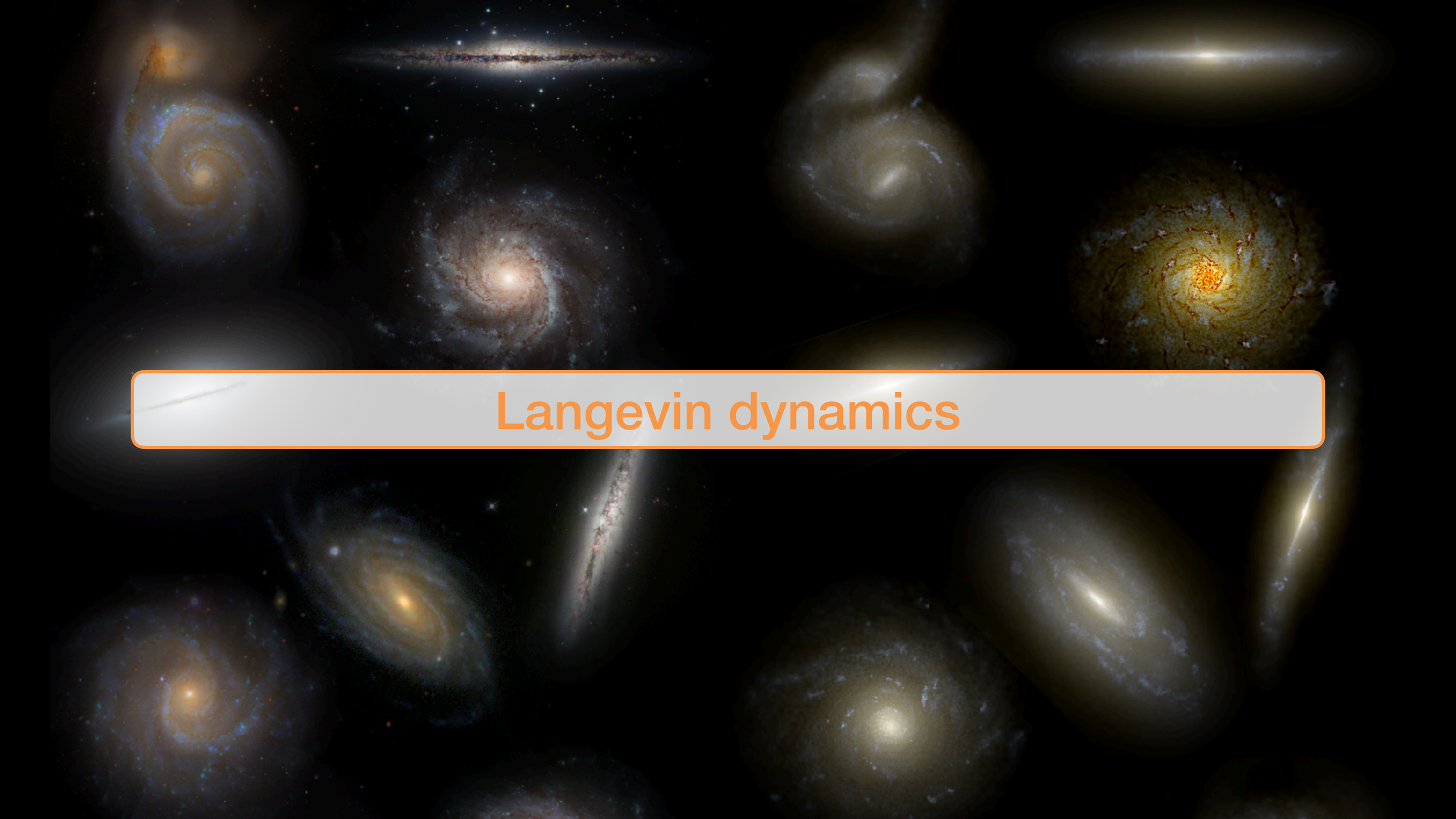
using integration by parts and some math magic we end up with (Hyvärinen+2005)

$$\mathbb{E}_{p(x)}[\,||\,\frac{1}{2}s_\theta(x)\,||_2^2 + \boxed{\text{trace}(\,\nabla_{\mathbf{x}}\log \text{p}(\mathbf{x}))}]$$

$$\approx \sum_{i=1}^{N}[\frac{1}{2}||\,s_\theta(x)\,||_2^2 + \boxed{\text{trace}(\,\nabla_{\mathbf{x}}\log \text{p}(\mathbf{x}))}]$$

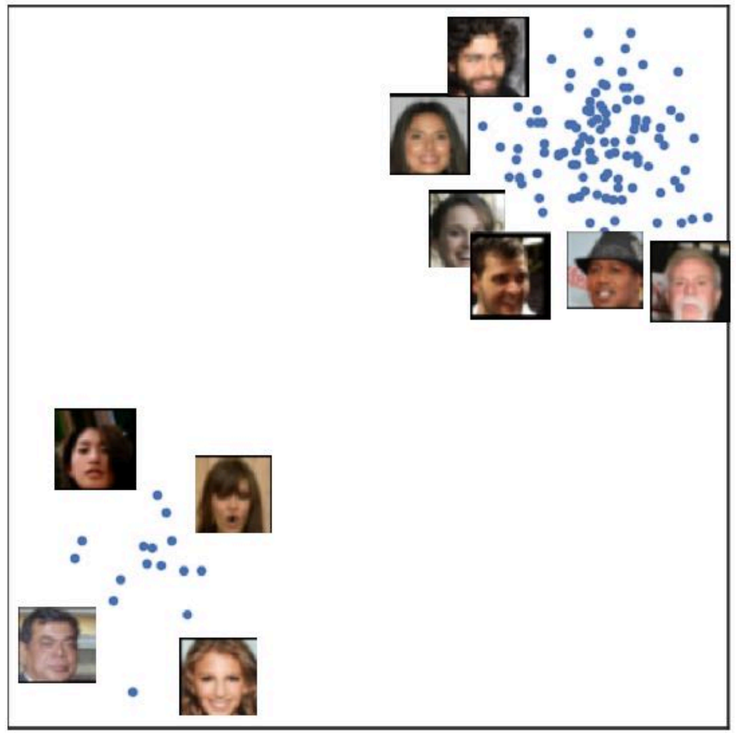
Trace of the Jacobian of s
Not scalable,
but can be efficiently
approximated in high d

But how does sampling then work? → Langevin dynamics (Welling&Teh 2011)



Langevin dynamics

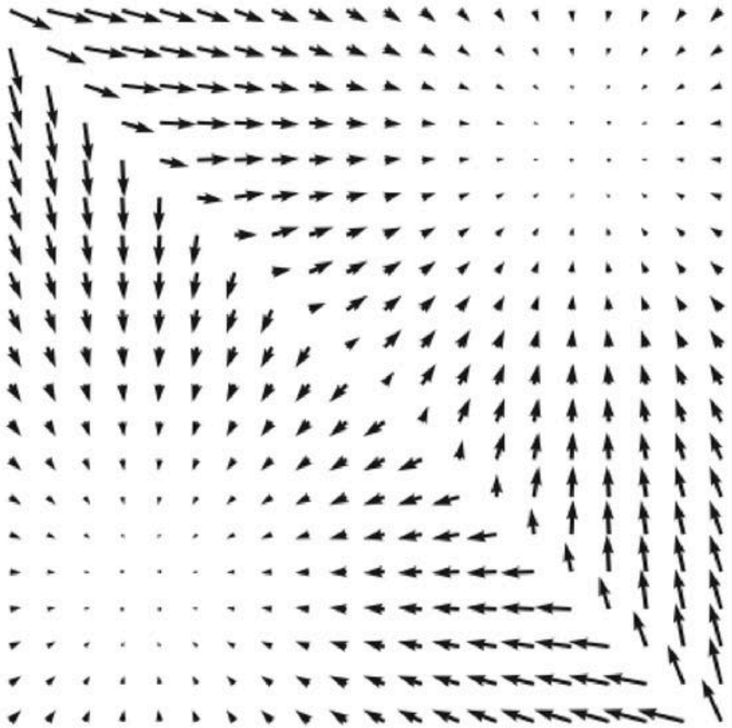
LANGEVIN DYNAMICS CONCEPT



Data samples

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$$

score matching



Scores

$$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

Langevin dynamics

New samples

blog post by https://yang-song.net/blog/2021/score/

MATH OF LANGEVIN DYNAMICS

Langevin dynamics provides an MCMC procedure to sample from a distribution $p(x)$ using only its score function $\nabla_x \log p(x)$

specifically, it initialises the chain from an arbitrary prior distribution $x_0 \sim \pi(x)$ and iterates

$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p(x) + \sqrt{2\epsilon} z_i, \quad i = 0,1,\ldots,K$$

where $z_i \sim \mathcal{N}(0,I)$

When $\epsilon \to 0$ and $K \to \infty$ the x_K from above converge to samples from $p(x)$

MATH OF LANGEVIN DYNAMICS

Langevin dynamics provides an MCMC procedure to sample from a distribution $p(x)$ using only its score function $\nabla_x \log p(x)$

specifically, it initialises the chain from an arbitrary prior distribution $x_0 \sim \pi(x)$ and iterates

$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p(x) + \sqrt{2\epsilon} z_i, \quad i = 0,1,\ldots,K$$

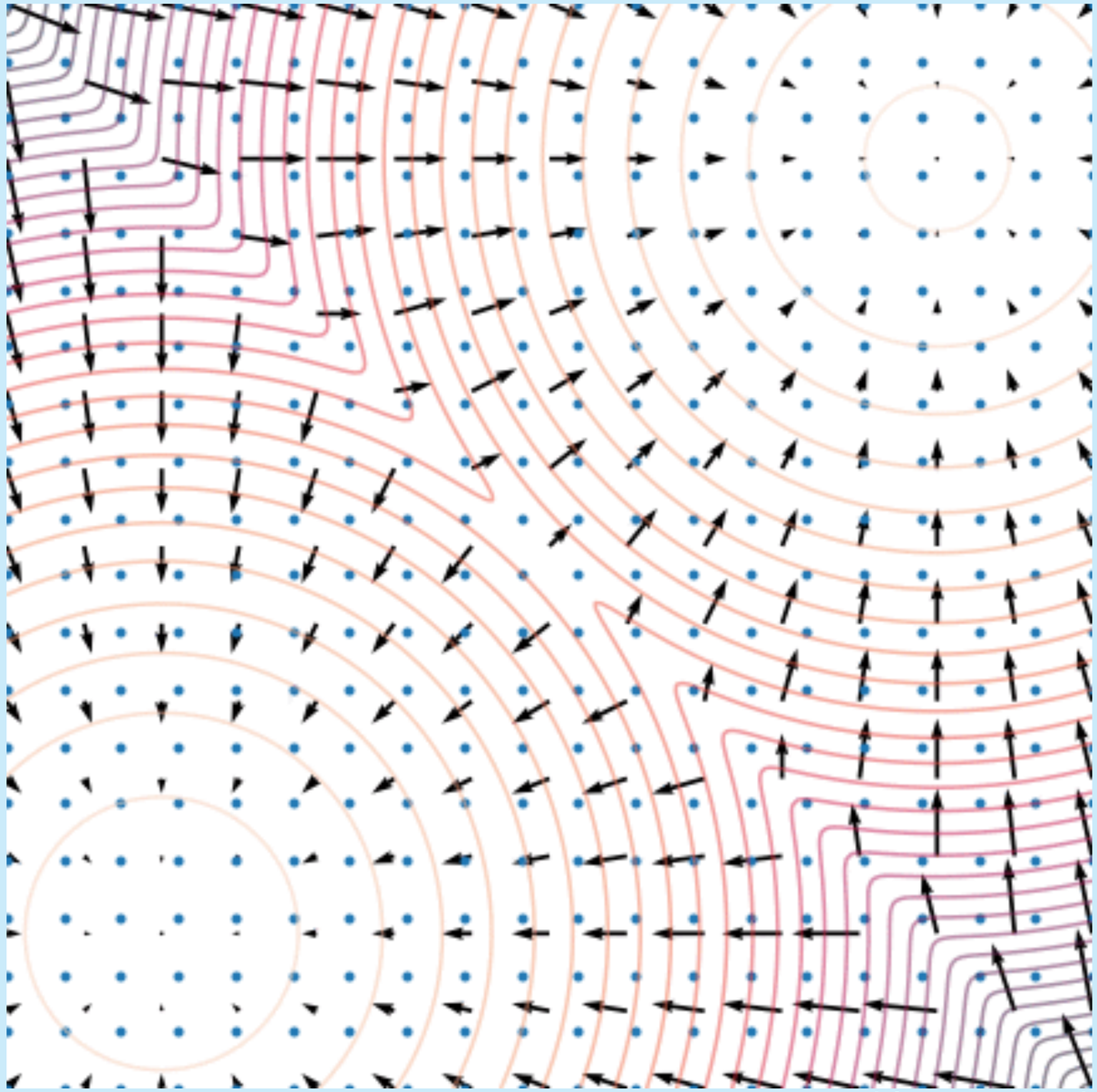
where $z_i \sim \mathcal{N}(0,I)$

Learned from data

When $\epsilon \to 0$ and $K \to \infty$ the x_K from above converge to samples from $p(x)$

Note: Langevin dynamics accesses $p(x)$ only through $\nabla_x \log p(x) \approx s_\theta(x)$

LANGEVIN DYNAMICS IN ACTION



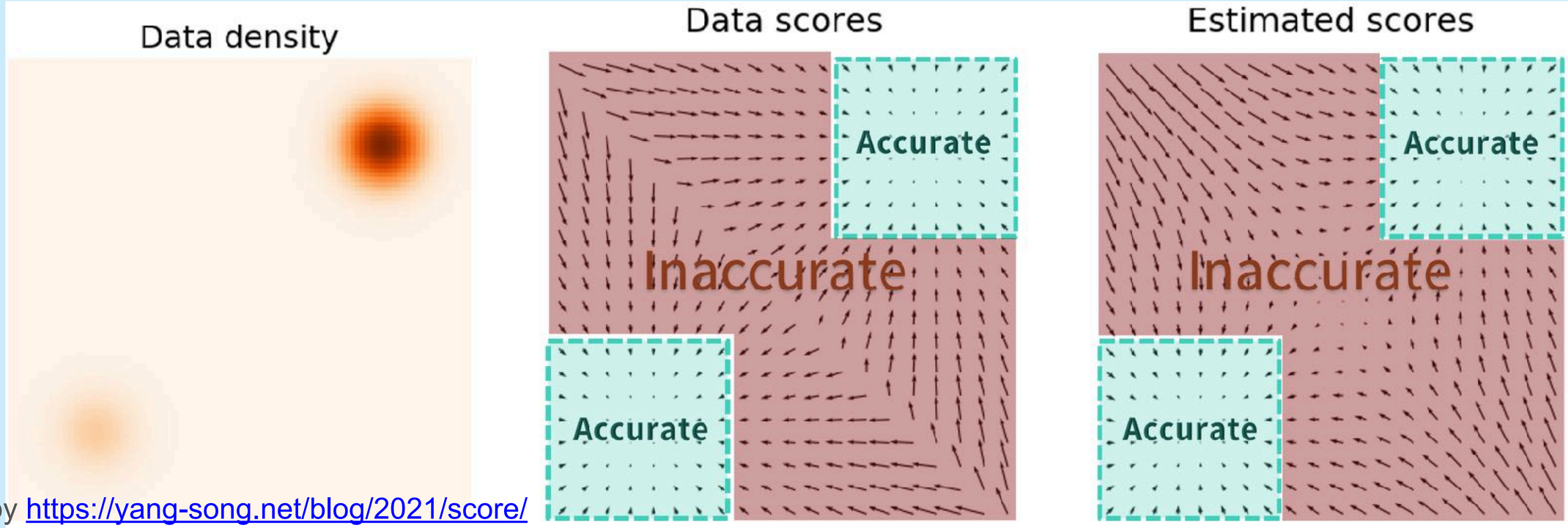
blog post by https://yang-song.net/blog/2021/score/

LANGEVIN DYNAMICS PROBLEM SOLVED?

NO! Estimated score function only accurate for high density regions
this is expected since score matching minimises the Fisher divergence

$$\mathbb{E}_{p(x)}[\,||\nabla_{\mathbf{x}}\log p(\mathbf{x}) - s_\theta(x)||_2^2] = \int p(x)\,||\nabla_{\mathbf{x}}\log p(\mathbf{x}) - s_\theta(x)||_2^2 \mathrm{dx}$$

since here l_2 norm is weighted by $p(x)$ its value is largely ignored in low density regions

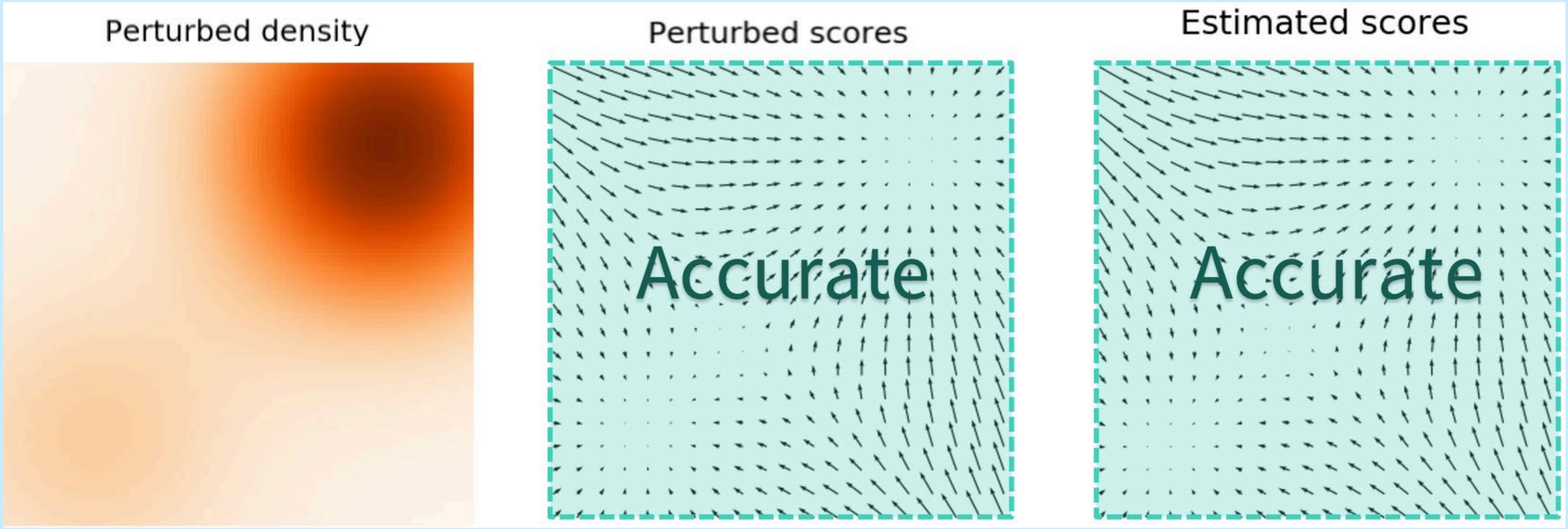


blog post by https://yang-song.net/blog/2021/score/

SOLUTION: SCORE MATCHING WITH MULTIPLE NOISE PERTURBATIONS

Perturb the data with noise and train score model on noisy data

But: How to choose suitable noise perturbations?



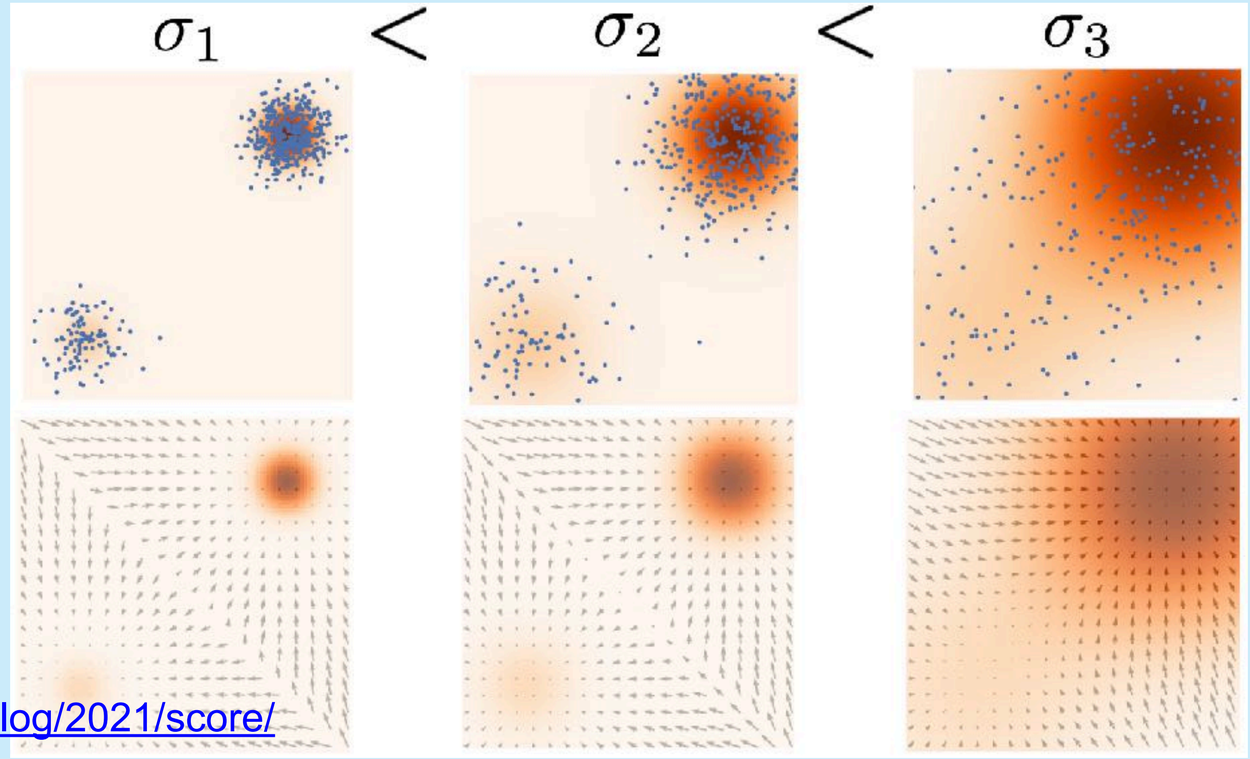
blog post by https://yang-song.net/blog/2021/score/

SOLUTION: SCORE MATCHING WITH MULTIPLE NOISE PERTURBATIONS

Perturb the data with noise of increasing $\sigma_1 < \sigma_2 < \sigma_3 \ldots < \sigma_L$

sample $x \sim p_{\sigma_i}(x)$ by sampling $x \sim p(x)$ and computing $x + \sigma_i z$ with $z \sim \mathcal{N}(0,I)$

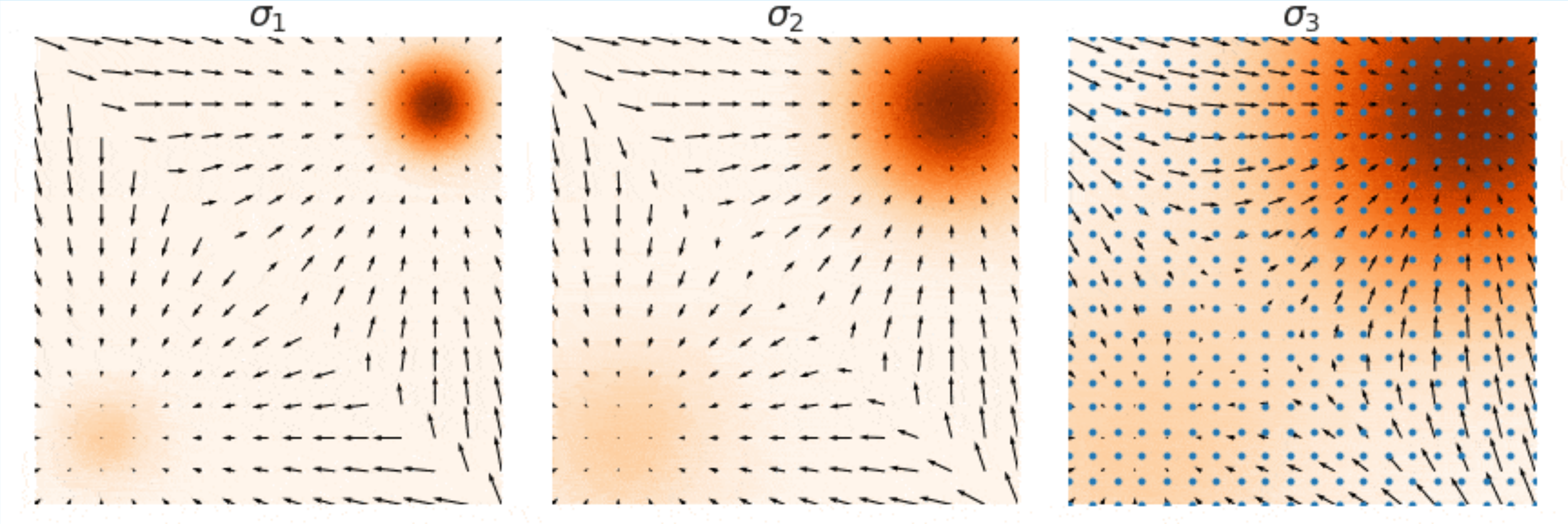
then train **noise conditional score-based model** $s_\theta(x, i)$



blog post by https://yang-song.net/blog/2021/score/

SAMPLING VIA ANNEALED LANGEVIN DYNAMICS

In the limit of infinite noise schedules → higher quality samples, exact log-likehood computation
and more important: **controllable generation for inverse problem solving**

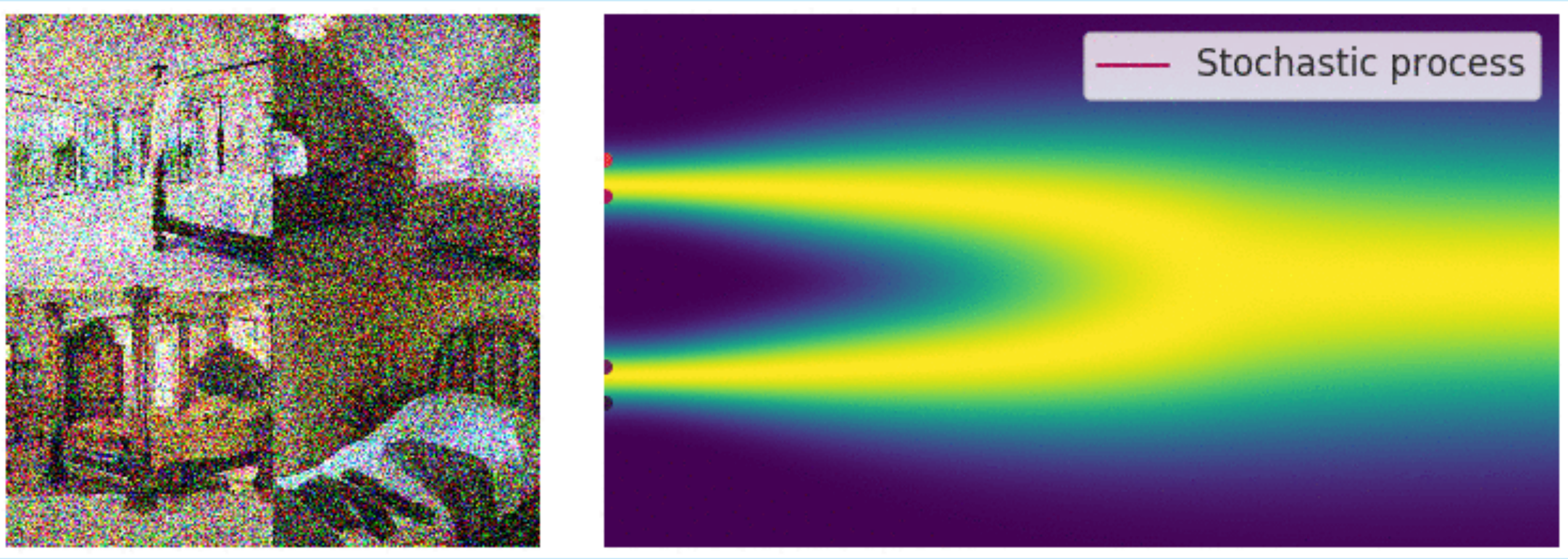


blog post by https://yang-song.net/blog/2021/score/

SCORE-BASED GENERATIVE MODELING WITH STOCHASTIC DIFFERENTIAL EQUATIONS

In the limit of infinite noise schedules → noise perturbation procedure is a continuous-time stochastic process

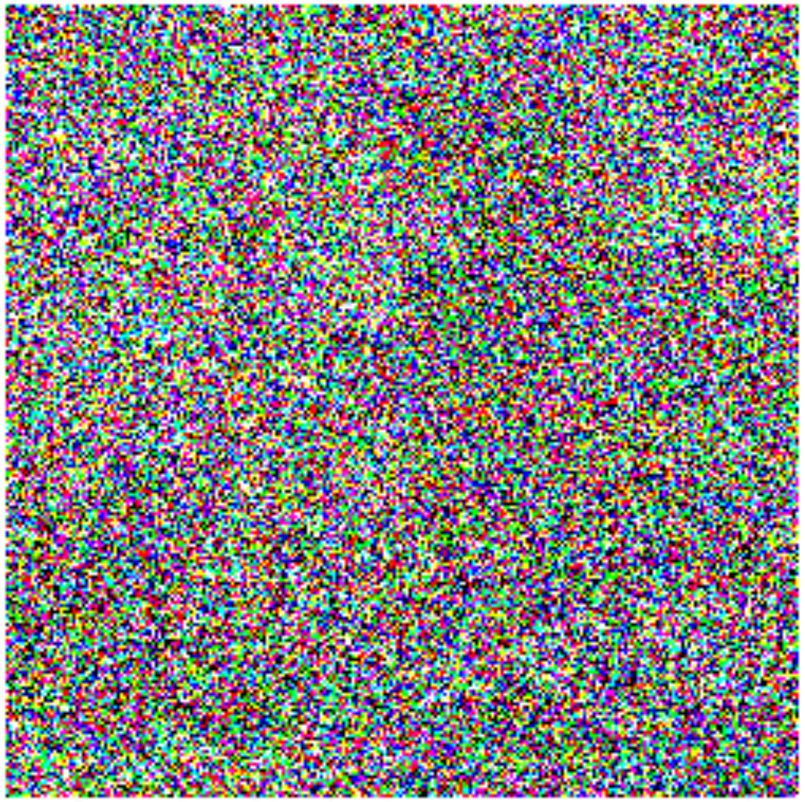
In general, the exact form of the SDE is part of the model, e.g. Variance Exploding SDE (VE SDE), the Variance Preserving SDE (VP SDE), and the sub-VP SDE (Song+2021)

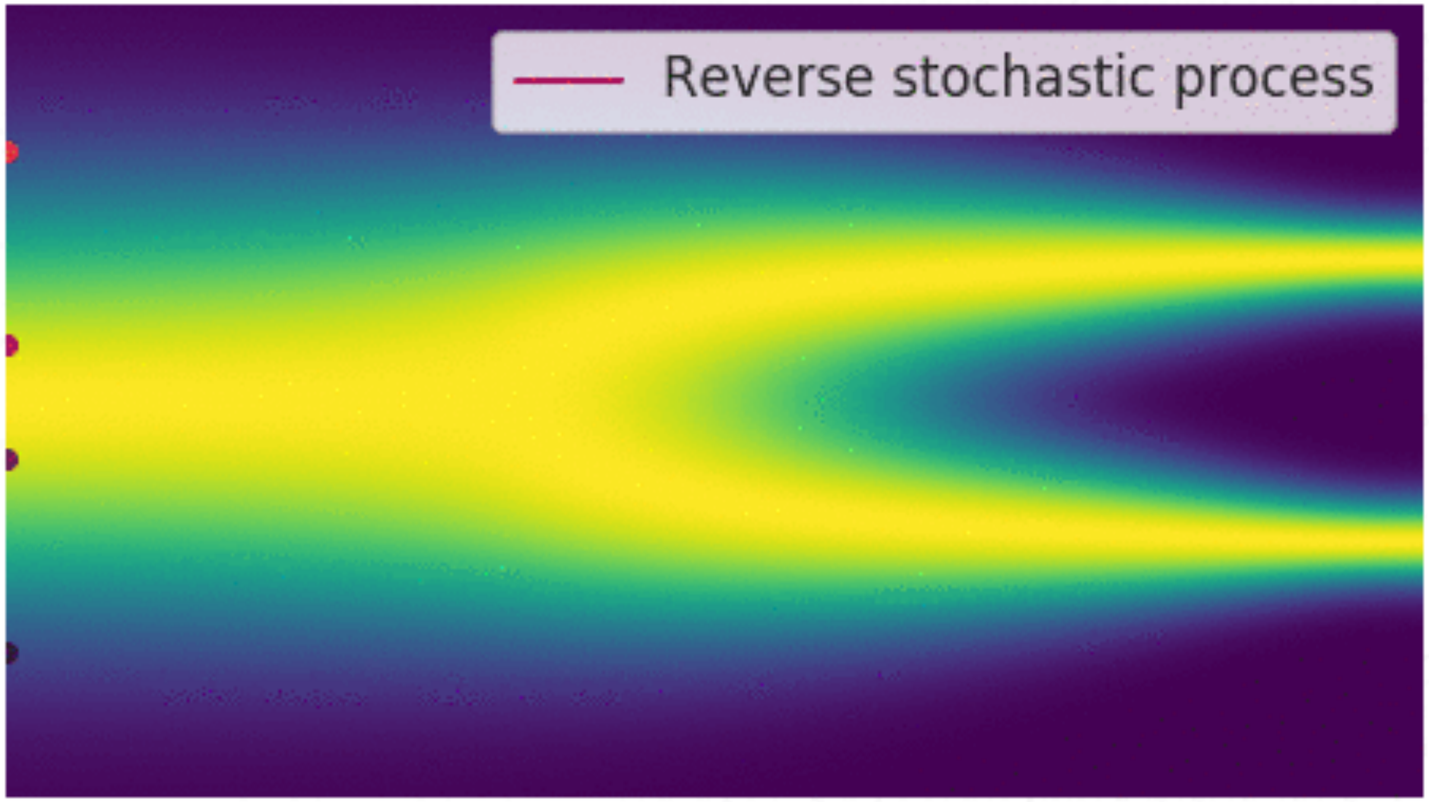


blog post by https://yang-song.net/blog/2021/score/

HOW TO SAMPLE WITH SCORE-BASED MODELS BASED ON SDES?

Note! any SDE has a corresponding reverse SDE!





blog post by https://yang-song.net/blog/2021/score/

THE CONCEPT BEHIND SDE BASED SCORE MATCHING

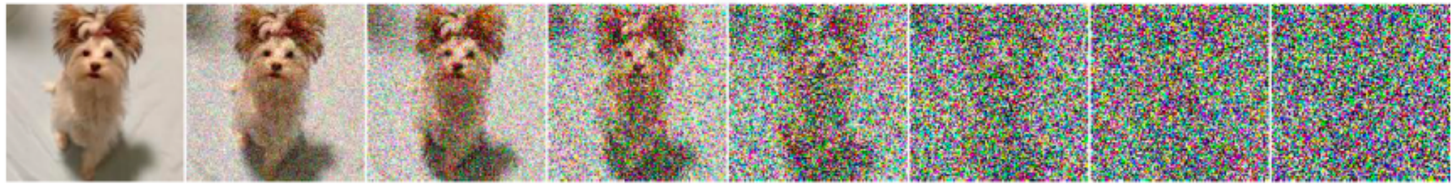
Sohl-Dickstein et al. (2015), Ho et al. (2020), Song et al. (2021)

Forward SDE: $$\mathrm{d}\mathbf{z}_t = -\frac{1}{2}\beta(t)\mathbf{z}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{w}_t$$

Easy

$t = 0$

$\mathbf{z}_0 \sim p_{\text{target}}$



$t = 1$

$\mathbf{z}_1 \sim \mathcal{N}(0, I_d)$

Hard

Backward SDE: $$\mathrm{d}\mathbf{z}_t = \left[-\frac{1}{2}\beta(t)\mathbf{z}_t - \beta(t)\,\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\right]\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\bar{\mathbf{w}}_t$$

Score function

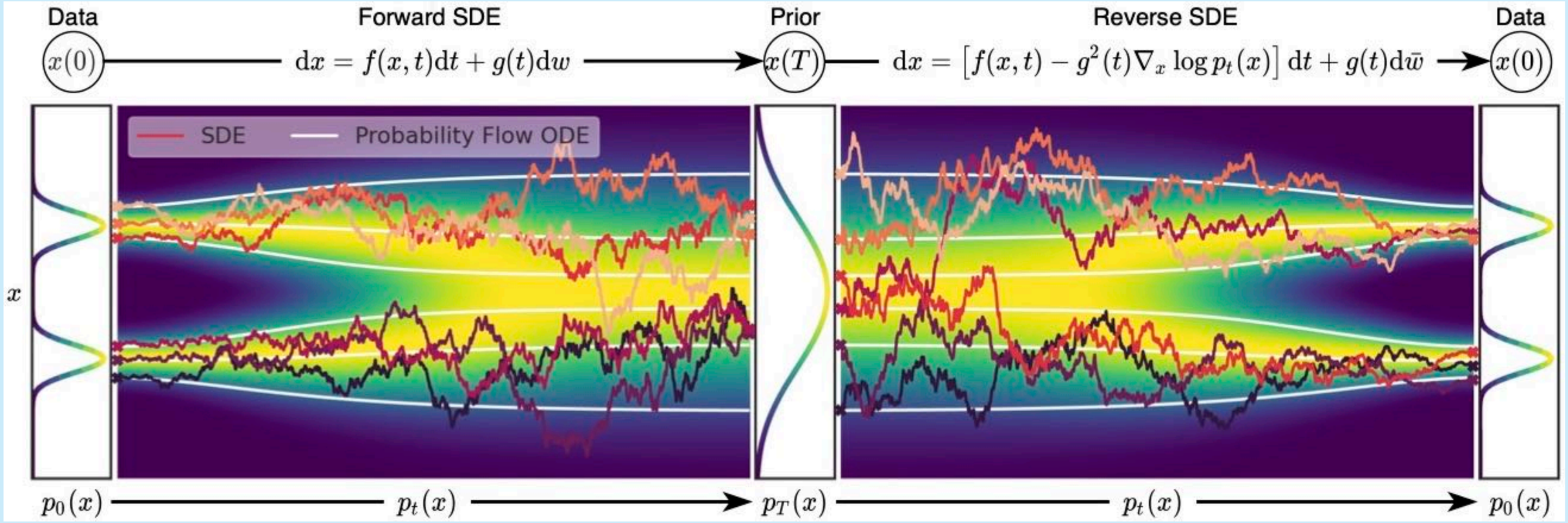
Solve with your favourite SDE solver, e.g. Euler-Maruyama, stochastic Runge-Kutta, etc.
Jolicoeur-Martineau+2021 for adaptive step size SDE solver

adapted from: slides by Bruno Régaldo-Saint Blancard and blog post by https://jmtomczak.github.io/blog/16/16_score_matching.html

SIDE NOTE: PROBABILITY FLOW ODES

Any SDE can be converted into an ordinary differential equation (ODE) without changing its marginal distribution

$$dx = \left[f(x,t) - \frac{1}{2} g^2(t) \nabla_x \log p_t(x) \right] dt$$



blog post by https://yang-song.net/blog/2021/score/

SIDE NOTE: PROBABILITY FLOW ODES

Any SDE can be converted into an ordinary differential equation (ODE) without changing its marginal distribution

$$\mathrm{dx} = \left[\mathrm{f(x,t)} - \frac{1}{2}\mathrm{g}^2(\mathrm{t})\,\nabla_{\mathrm{x}}\log \mathrm{p_t(x)}\right]\mathrm{dt}$$

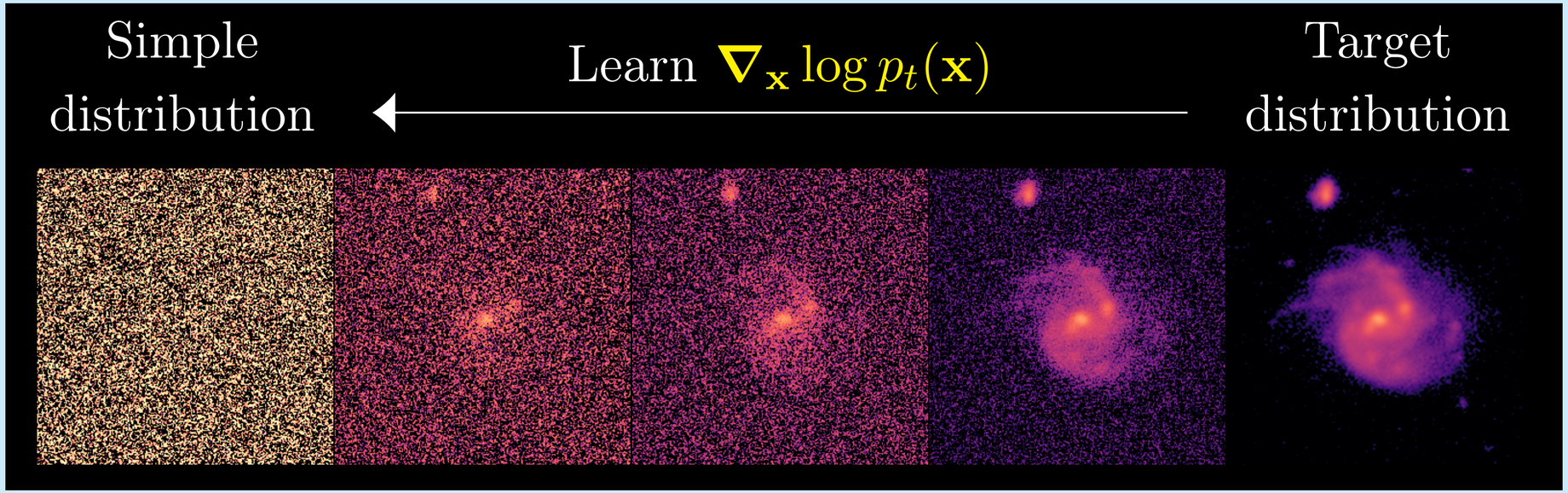
Why is this interesting?

when replacing $\nabla_x \log p_t(x)$ with approximation $s_\theta(x, t)$ the probability flow ODE becomes a special case of a neural ODE (Chen+2018)

in particular its an example of a continuous normalizing flow (Gerathwohl+2019)

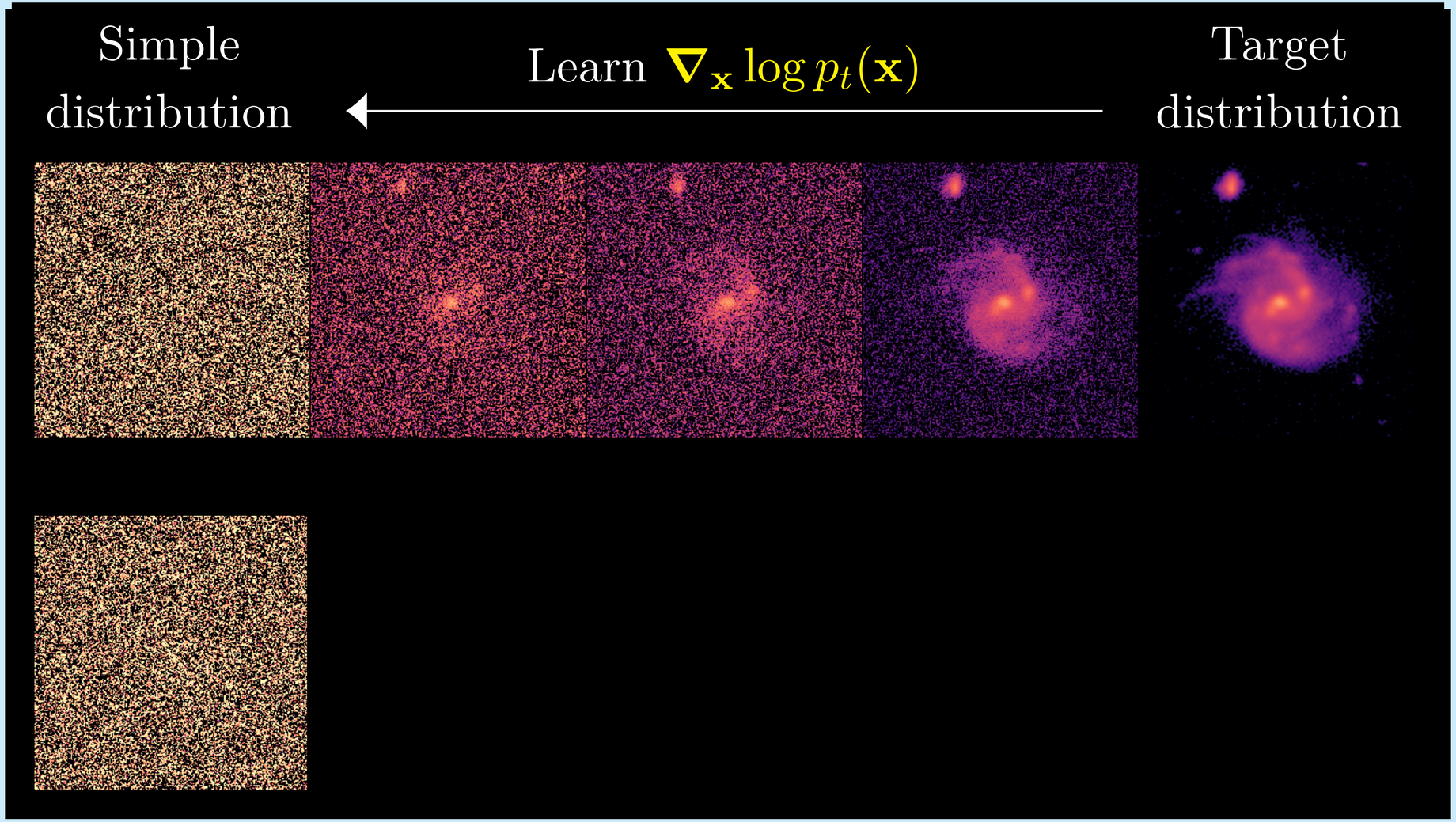
blog post by https://yang-song.net/blog/2021/score/

TRAINING SCORE MATCHING MODELS



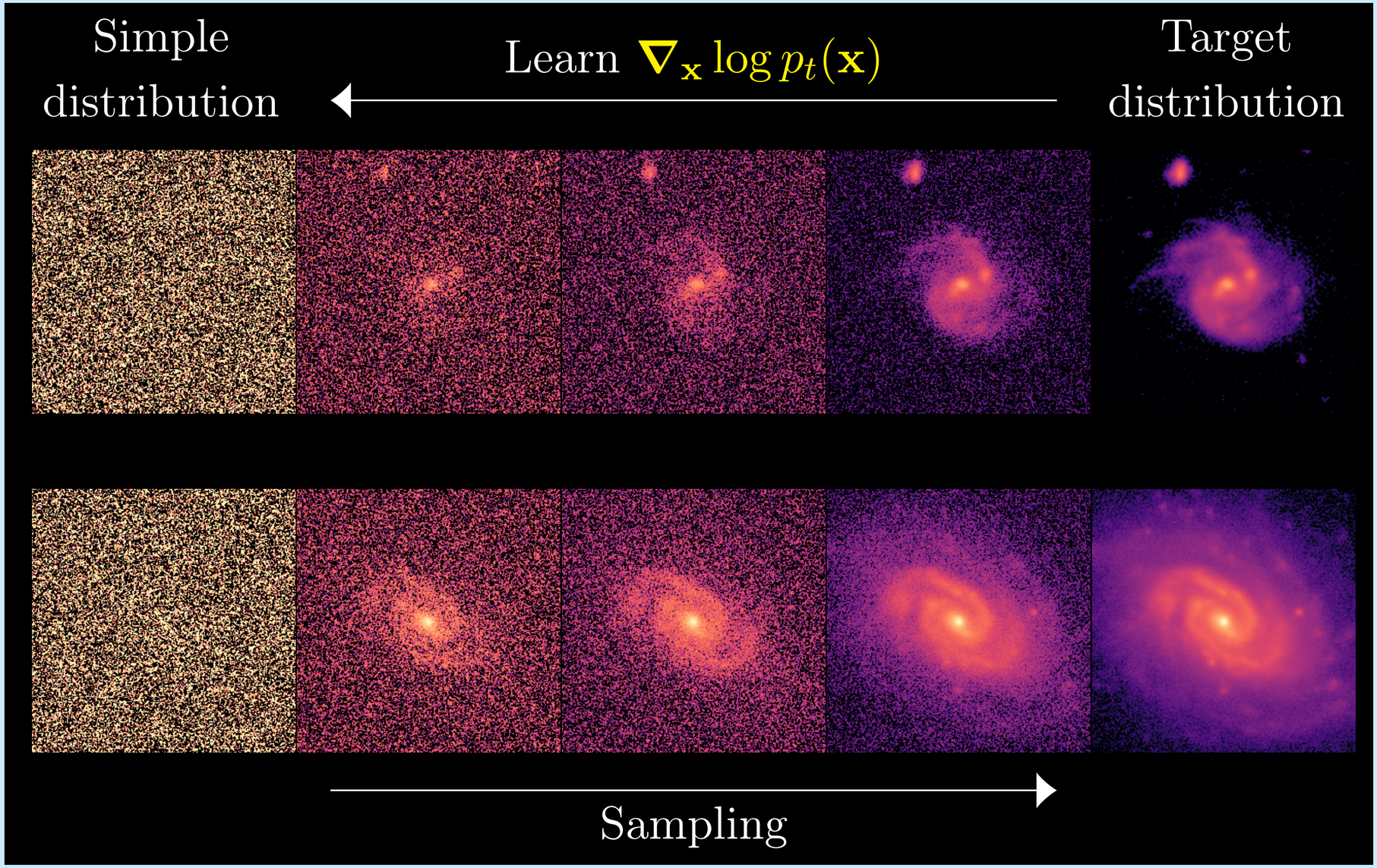
slides adapted from Alexandre Adams

TRAINING SCORE MATCHING MODELS



slides adapted from Alexandre Adams

TRAINING SCORE MATCHING MODELS



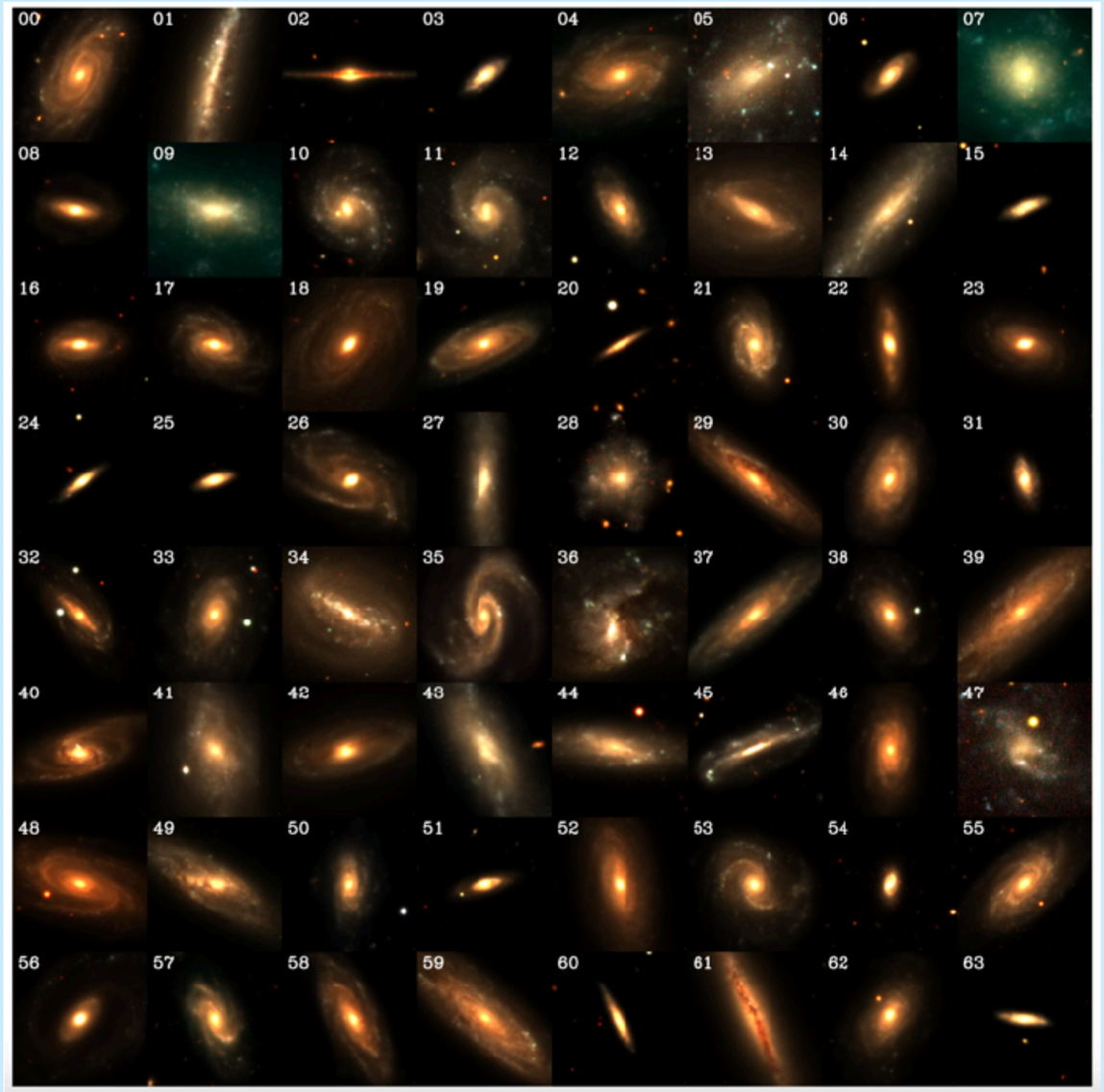
slides adapted from Alexandre Adams

Applications to Astrophysics

APPLICATION OF SCORE MATCHING AND DIFFUSION MODELS

Learn generative model purely from data!

Smith+2021



http://www.mjjsmith.com/thisisnotagalaxy/

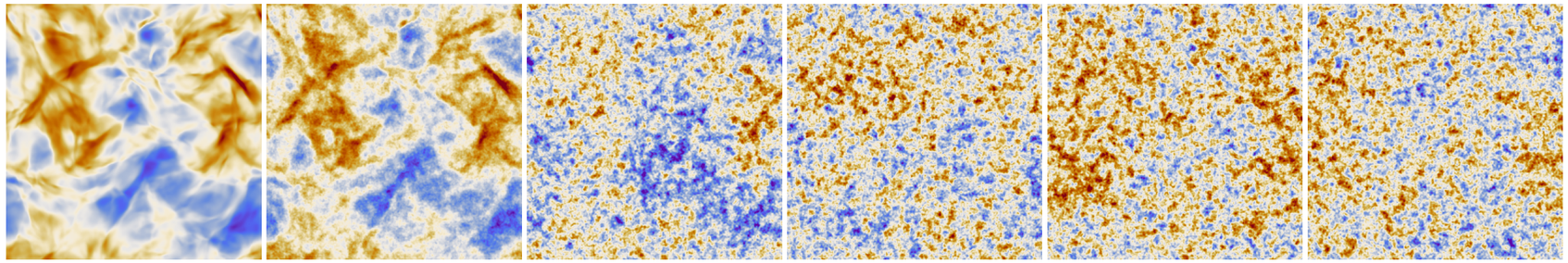
DIFFUSION MODELS AS A MAPPING $p_{\text{dust}} \leftrightarrow p_{\text{CMB}}$

Forward SDE: $$\mathrm{d}\mathbf{z}_t = -\frac{1}{2}\beta(t)\mathbf{z}_t\,\mathrm{d}t + \sqrt{\beta(t)}\boldsymbol{\Sigma}_{\phi}^{\frac{1}{2}}\,\mathrm{d}\mathbf{w}_t$$

Easy

$t = 0$

$\mathbf{z}_0 \sim p_{\text{dust}}$



$t = 1$

$\mathbf{z}_1 \sim p_{\text{CMB}}$

$\mathcal{T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$

Hard

Backward SDE: $$\mathrm{d}\mathbf{z}_t = \left[-\frac{1}{2}\beta(t)\mathbf{z}_t - \beta(t)\boldsymbol{\Sigma}_{\phi}\,\underbrace{\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)}_{\text{Score function}}\right]\mathrm{d}t + \sqrt{\beta(t)}\boldsymbol{\Sigma}_{\phi}^{\frac{1}{2}}\,\mathrm{d}\overline{\mathbf{w}}_t$$

Heurtel-Depeignes+2023

slides adapted from Bruno Régaldo-Saint Blancard

DIFFUSION MODELS AS A MAPPING $p_{\text{dust}} \leftrightarrow p_{\text{CMB}}$

Component Separation ?

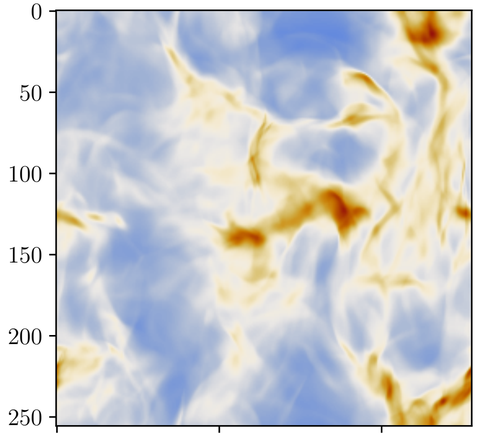
Visual assessment:

Dust $\mathbf{x}$

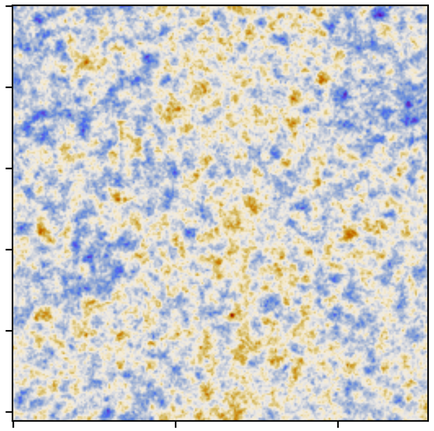
CMB ϵ

Observation y

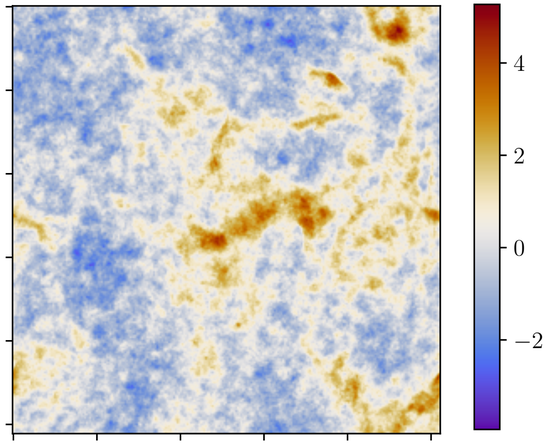
True



+



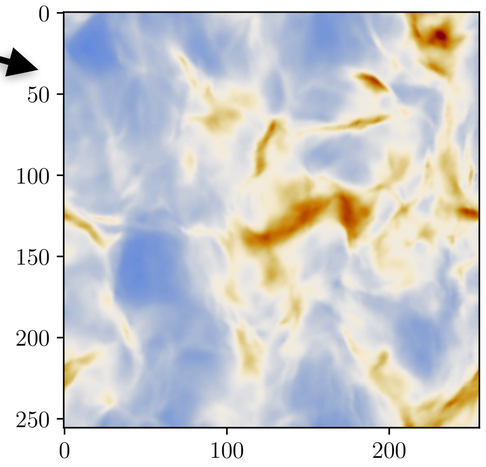
=



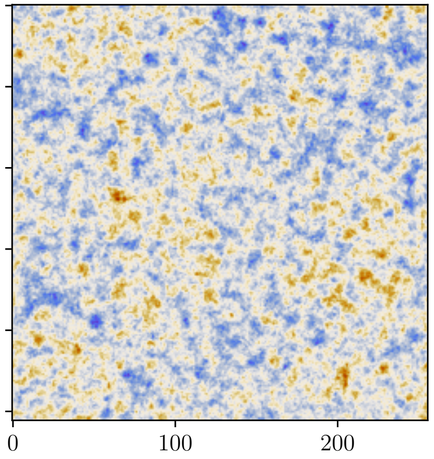
Sample from

$p(\mathbf{x} \mid \mathbf{y}, \phi)$

Reconstructed



+



=

Heurtel-Depeignes+2023

slides adapted from Bruno Régaldo-Saint Blancard

POSTERIOR SAMPLES WITH DIFFUSION MODELS

Posterior $p(x \mid y)$ with observation y is given by Bayes' theorem:

$$\log p(x \mid y) = \log p(y \mid x) + \log p(x) - \log p(y)$$

with $p(y \mid x)$ being the likelihood
and $p(x)$ the prior.

hence the score is given by:

$$\nabla_x \log p(x \mid y) = \nabla_x \log p(y \mid x) + \nabla_x \log p(x) - \nabla_x \log p(y)$$

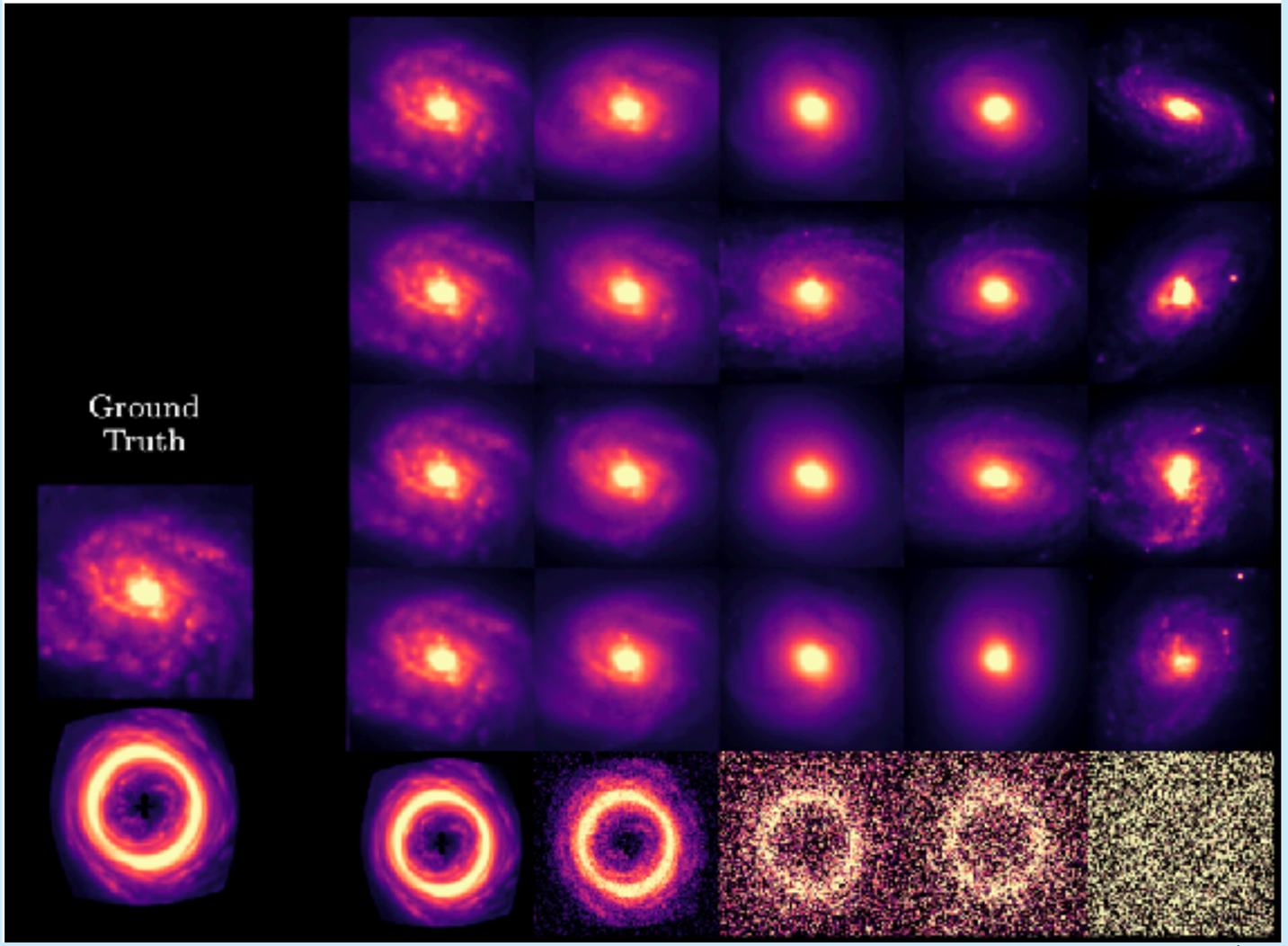
To a good approximation, we can calculate the likelihood score analytically or numerically from the known forward process.

This is the score we learnt with Score Matching from data!

=0 because it does not depend on x.

POSTERIOR SAMPLES WITH DIFFUSION MODELS

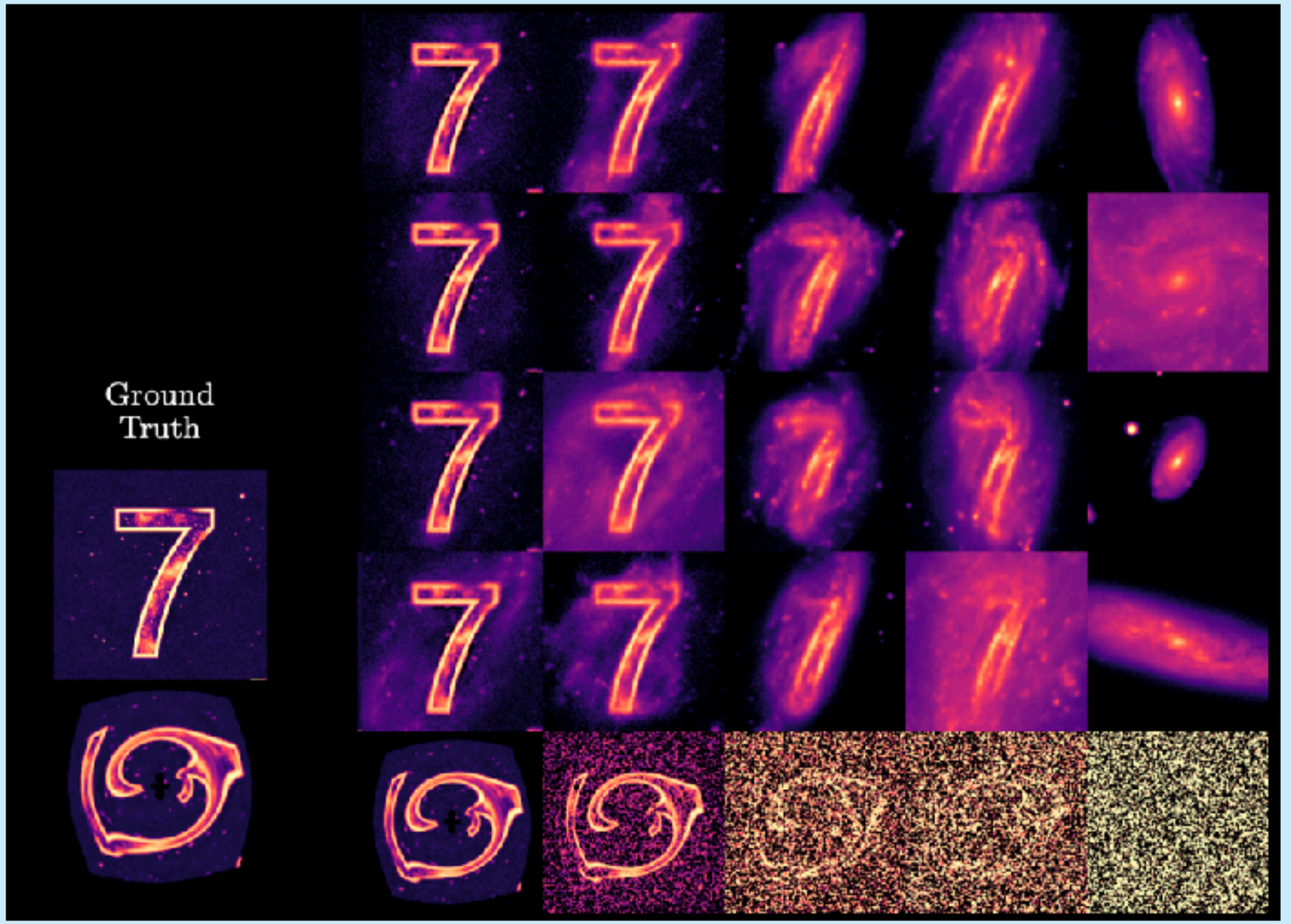
Posterior samples of source galaxies in strong gravitational lenses with score-based priors



Adam+2022

POSTERIOR SAMPLES WITH DIFFUSION MODELS

Posterior samples for out-of-distribution galaxies



posterior samples

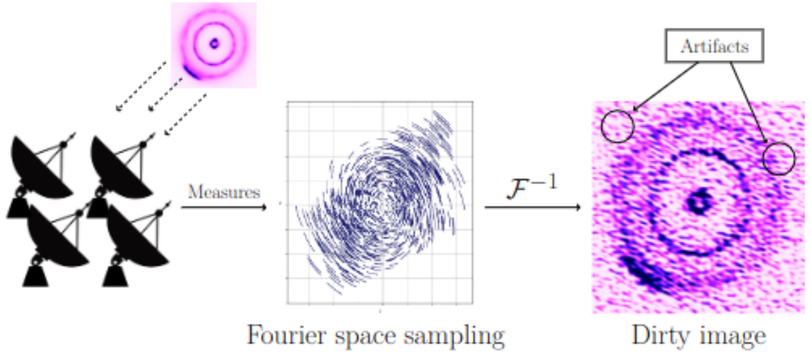
Adam+2022

less informative likelihood

MORE APPLICATIONS OF DIFFUSION MODELS

Bayesian Imaging for Radio Interferometry with Score-Based Priors

Noé Dia[1,2,4] M. J. Yantovski-Barth[1,2,4] Alexandre Adam[1,2,4] Micah Bowles[5]
Pablo Lemos[1,2,3,4] Anna M. M. Scaife[5,6] Yashar Hezaveh[1,2,3,4,7,8]
Laurence Perreault-Levasseur[1,2,3,4,7,8]
[1]Université de Montréal [2]Ciela Institute [3]Flatiron Institute [4]Mila [5]University of Manchester
[6]The Alan Turing Institute [7]Trottier Space Institute [8]Perimeter Institute



THE ASTROPHYSICAL JOURNAL LETTERS

OPEN ACCESS

Beyond Gaussian Noise: A Generalized Approach to Likelihood Analysis with Non-Gaussian Noise

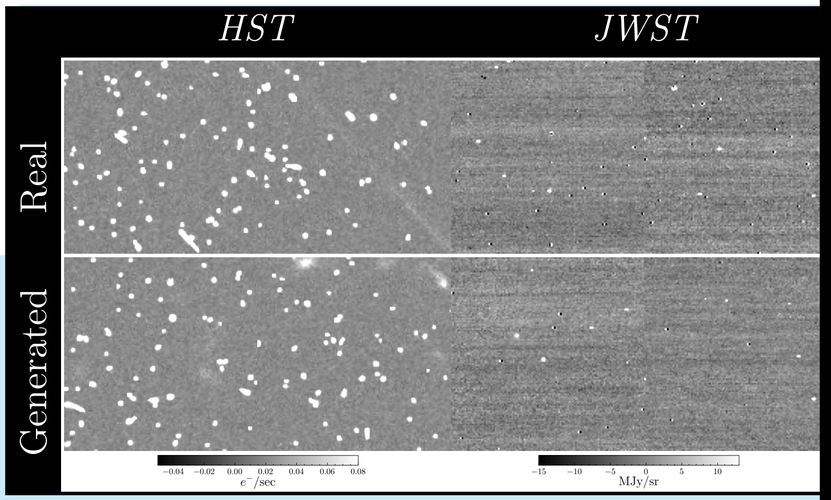
Ronan Legin[5,1,2,3], Alexandre Adam[5,1,2,3], Yashar Hezaveh[1,2,3,4], and Laurence Perreault-Levasseur[1,2,3,4]

Published 2023 June 6 • © 2023. The Author(s). Published by the American Astronomical Society.

The Astrophysical Journal Letters, Volume 949, Number 2

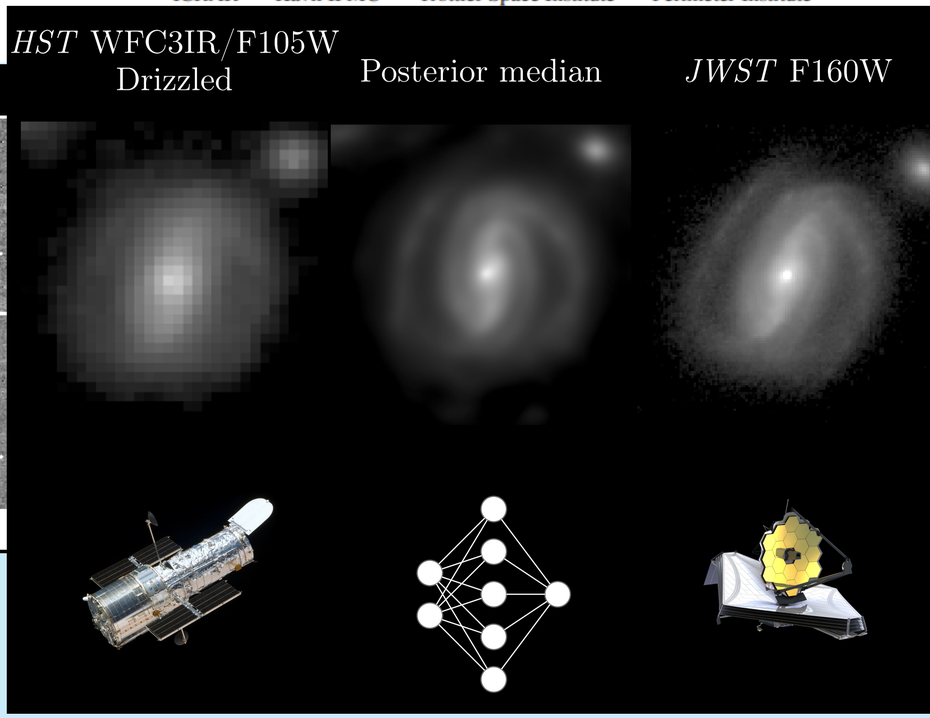
Citation Ronan Legin *et al* 2023 *ApJL* **949** L41

DOI 10.3847/2041-8213/acd645



Echoes in the Noise: Posterior Samples of Faint Galaxy Surface Brightness Profiles with Score-Based Likelihoods and Priors

Alexandre Adam[1,2,4] Connor Stone[1,2,4] Connor Bottrell[5,6] Ronan Legin[1,2,4]
Yashar Hezaveh[1,2,3,4,7,8] Laurence Perreault-Levasseur[1,2,3,4,7,8]
[1]Université de Montréal [2]Ciela Institute [3]CCA, Flatiron Institute [4]Mila
[5]ICRAR [6]Kavli IPMU [7]Trottier Space Institute [8]Perimeter Institute



WHERE TO FIND THE LECTURE MATERIAL

https://github.com/TobiBu/graddays

SUMMARY & CONCLUSION

Main take away:

scientific motivated inductive bias helps to be more robust, more data efficient and better interpretable

My personal message:

Write better code!
Share more data!
Build more open-source software!

This will accelerate research cycles and lets you engage with peers early on!


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS