Deep generative modelling for biomedical data

Siddharth Ramchandran[1]

CS-E5890 - Statistical Genetics and Personalised Medicine

Spring 2020

[1] - Dept. Of Computer Science, Aalto University

siddharth.ramchandran@aalto.fi

What is a generative model?

Given a dataset X, and associated labels Y:

Generative models

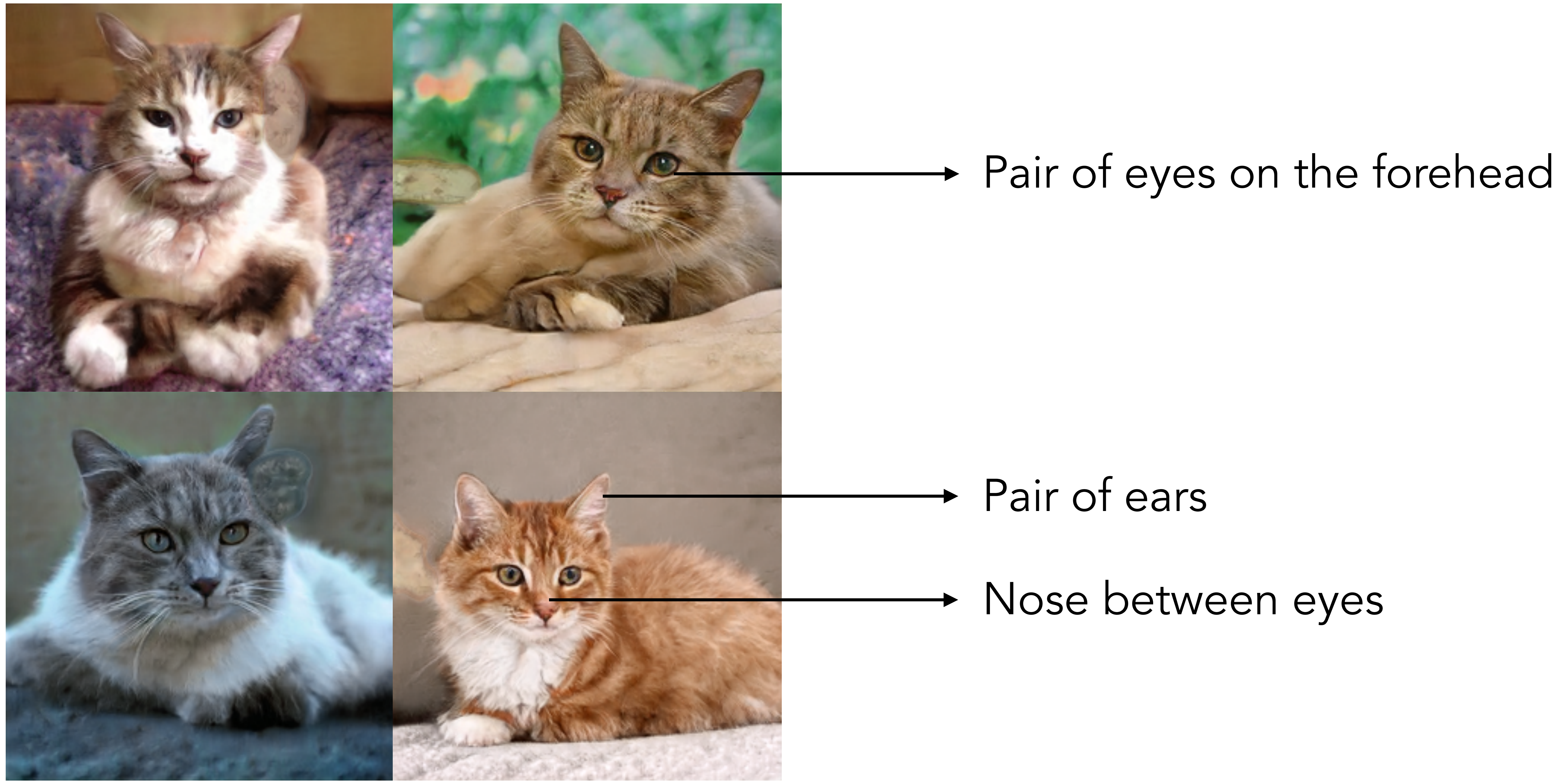
- capture the joint probability $p(X, Y)$ or $p(X)$ if there are no labels
- use probabilistic modelling to understand data generating mechanisms

Discriminative models

- capture the conditional probability $p(Y|X)$
- used to differentiate between different kinds of instances.

What is a generative model?

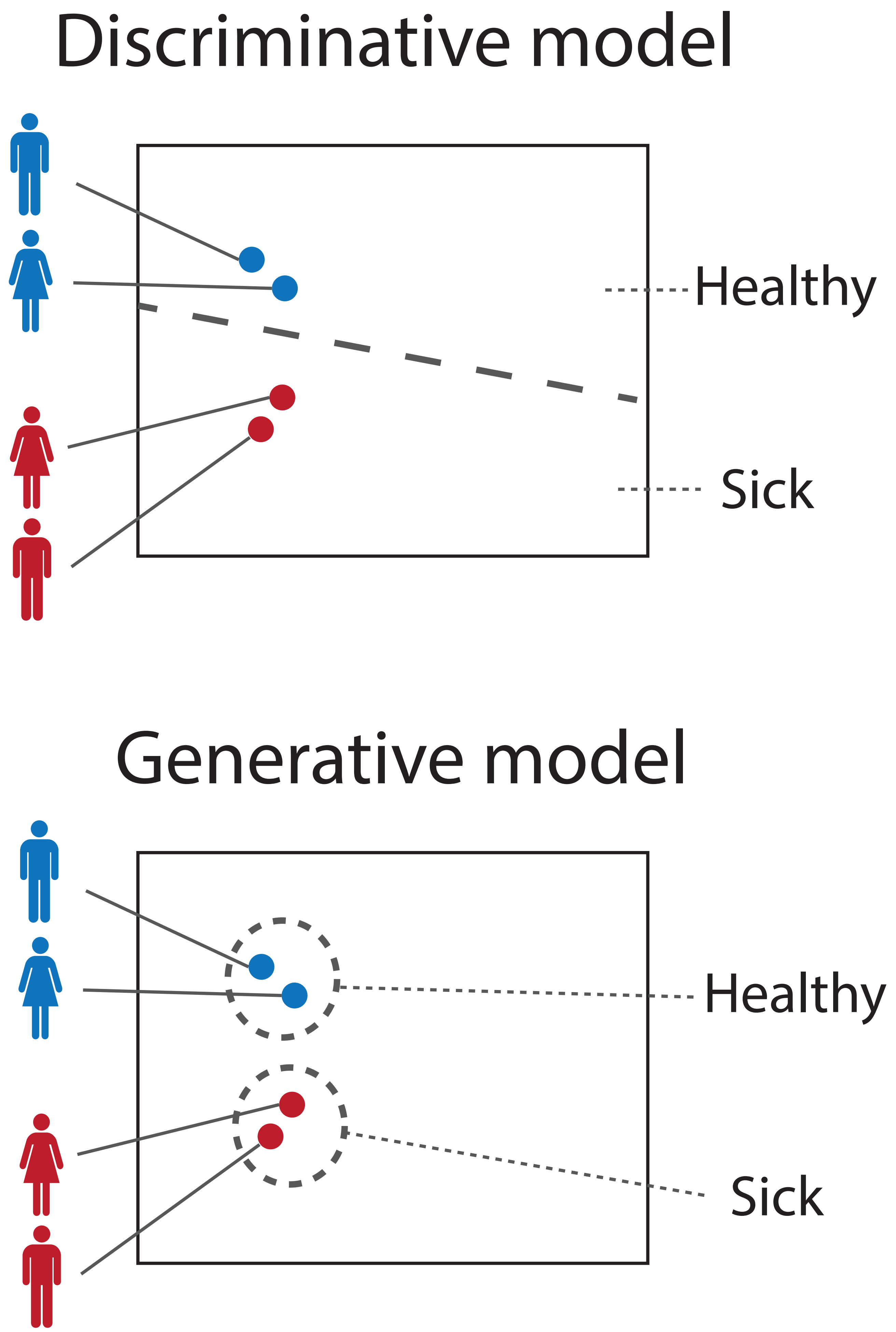
- Generative models tackle a much harder challenge.



Cats from a model that generates convincing “fake” data (Karras et. al, 2019)

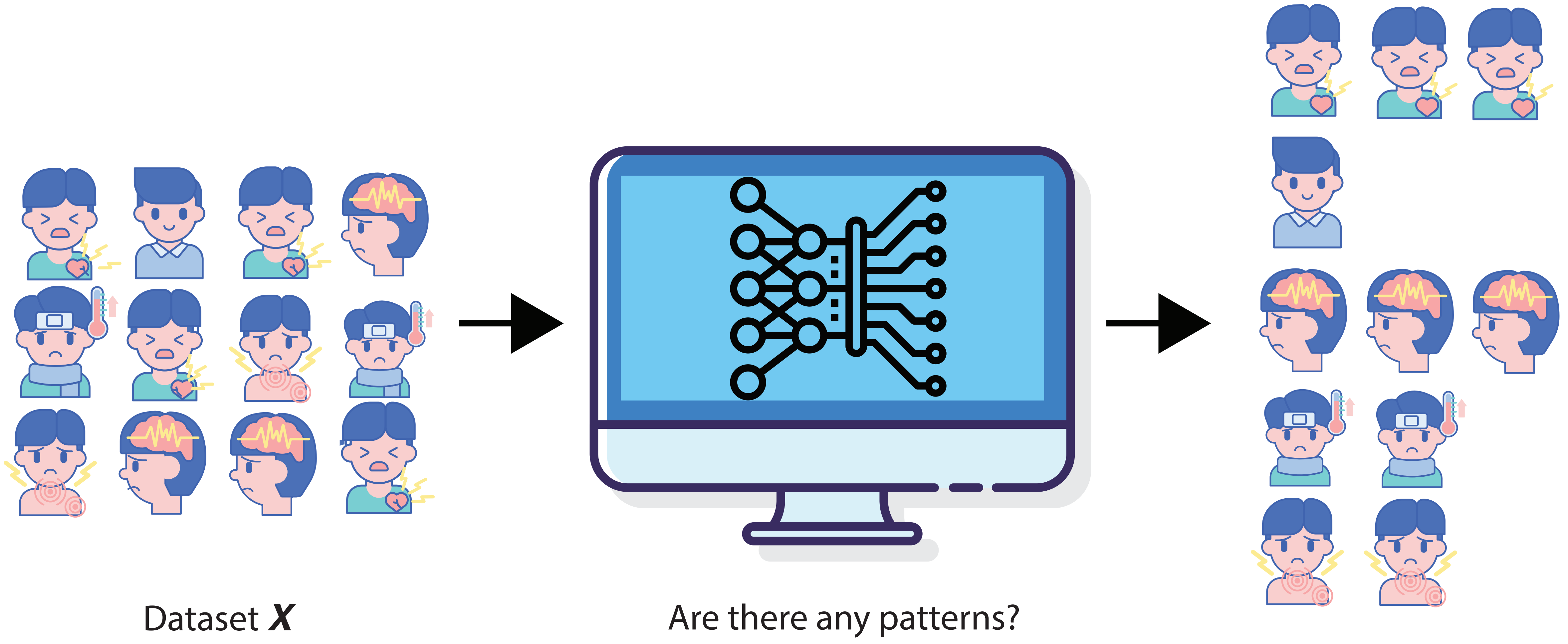
Generative vs. Discriminative

- Discriminative models try to learn the boundary in the data space (i.e. discriminate)
- Generative models try to "learn" the data in order to be able to replicate it.



Unsupervised learning

- It tries to find interesting patterns/transformations of input data without target labels.
- In other words, dataset X is given, but labels Y are not.
- We try to model $p(X)$.



siddharth.ramchandran@aalto.fi

What is biomedical data?

- It comprises of observations of one or more variables of a patient or a sample of individuals which represent a population of interest.
- Some analysis have an outcome variable (or response variable) which defines the termination of the analysis.
- The attributes or variables can be quantitative (discrete and continuous) or qualitative (nominal and ordinal)
- Biomedical data is characterised by high-dimensionality and missing values.

Contents of biomedical data

Patient information: date of birth, sex, date of study entry/exit

Routine medical data: height, weight, blood pressure, cholesterol levels, medications used

Specialised laboratory data: proteins, lipids, metabolites, glycans, imaging. omics data such proteomics and metabolomics

Genetic data: genotype or sequencing. Gene expressions and epigenetic data (DNA methylation).

Examples of biomedical data

MIMIC-III collection: a freely-accessible critical care database.

- Anonymised health data associated with 61,532 ICU admissions.
- Demographics, vital signs, laboratory tests, medications, and more.

Biobank data: such as Helsinki Biobank and UK Biobank

- Genomics, demographics, chromosomes and patient electronic health records.
- The objectives can be, for example, disease stratification and temporal modelling.

What can the models be used for?

siddharth.ramchandran@aalto.fi

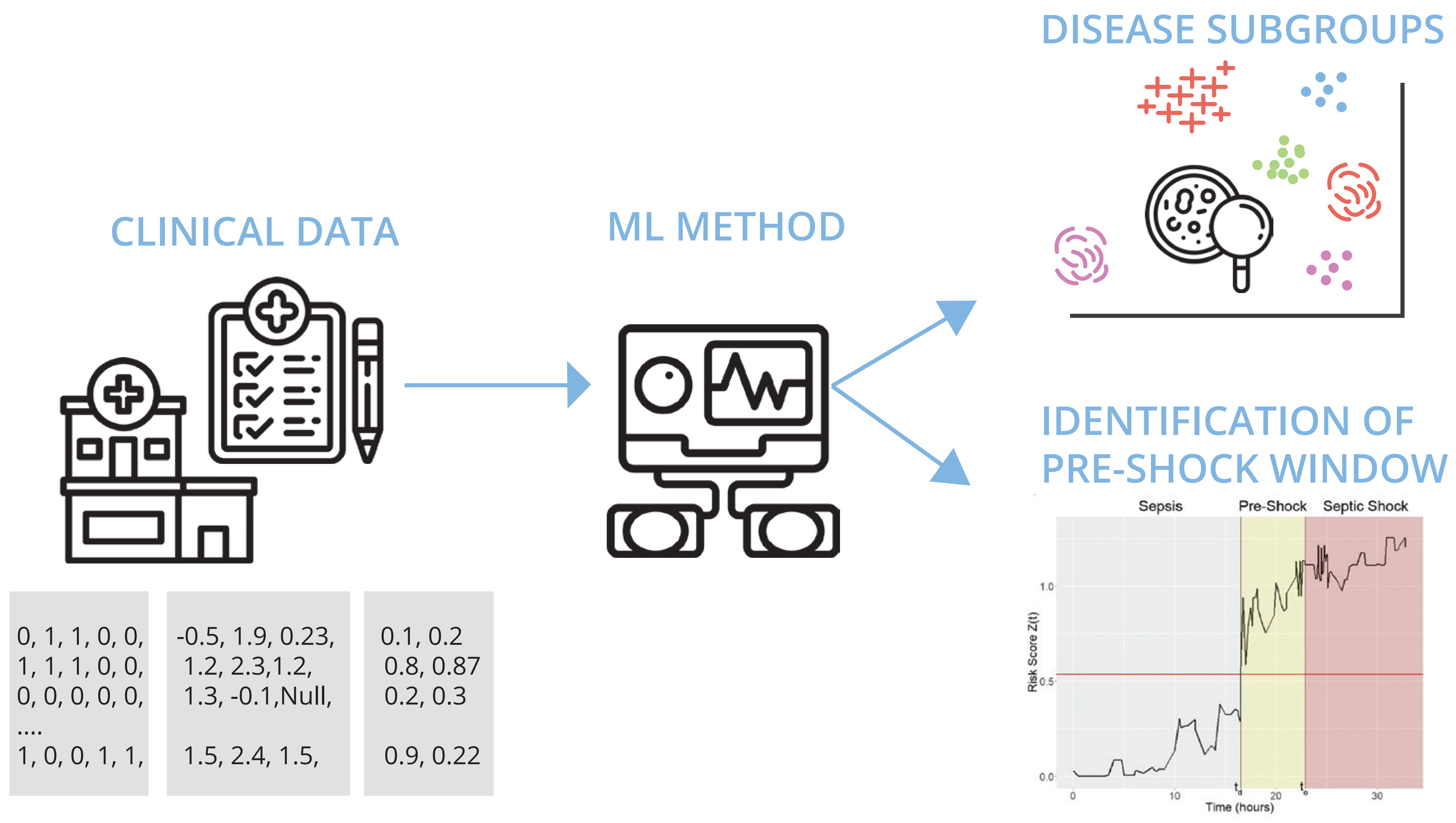
Data imputation

- Most longitudinal datasets comprise of a significant number of missing values.
- Make use of past and future information into account when predicting missing values

	♥	🖥	🌡	🕒
Patient 1	?	0.21	0.88	1
Patient 2	0.12	?	0.45	2.5
Patient 1	0.44	?	0.67	4
Patient 1	0.56	0.234	?	5

Down-stream tasks

- Perform disease stratification by identifying clusters (or disease sub-types) among patients.
- Perform classification. For example, identifying patient mortality or early detection of sepsis.
- Identify time windows of interest. For example, identification of pre-septic shock period.



0, 1, 1, 0, 0,
1, 1, 1, 0, 0,
0, 0, 0, 0, 0,
....
1, 0, 0, 1, 1,

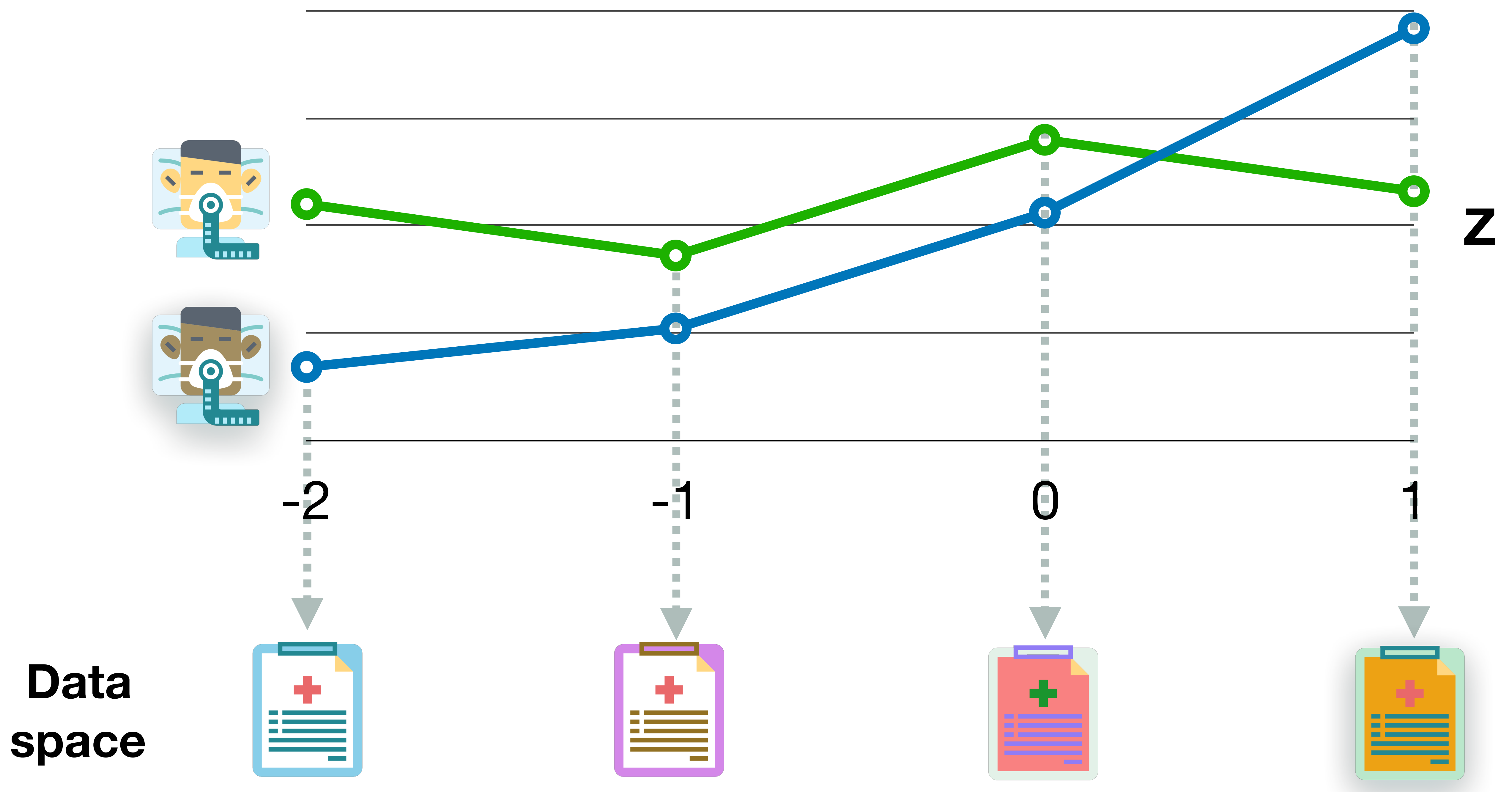
-0.5, 1.9, 0.23,
1.2, 2.3,1.2,
1.3, -0.1,Null,
1.5, 2.4, 1.5,

0.1, 0.2
0.8, 0.87
0.2, 0.3
0.9, 0.22

siddharth.ramchandran@aalto.fi

Disease progression

- We can model the non-linear evolution of a patient's health.
- Think of the generated low-dimensional representation as a latent physiological state of the patient with the original data space as the observable measurements



Popular DL models for generative modelling

Auto Encoders (This lecture.)

Used for dimensionality reduction, data imputation, drug discovery, etc.

Generative Adversarial Networks

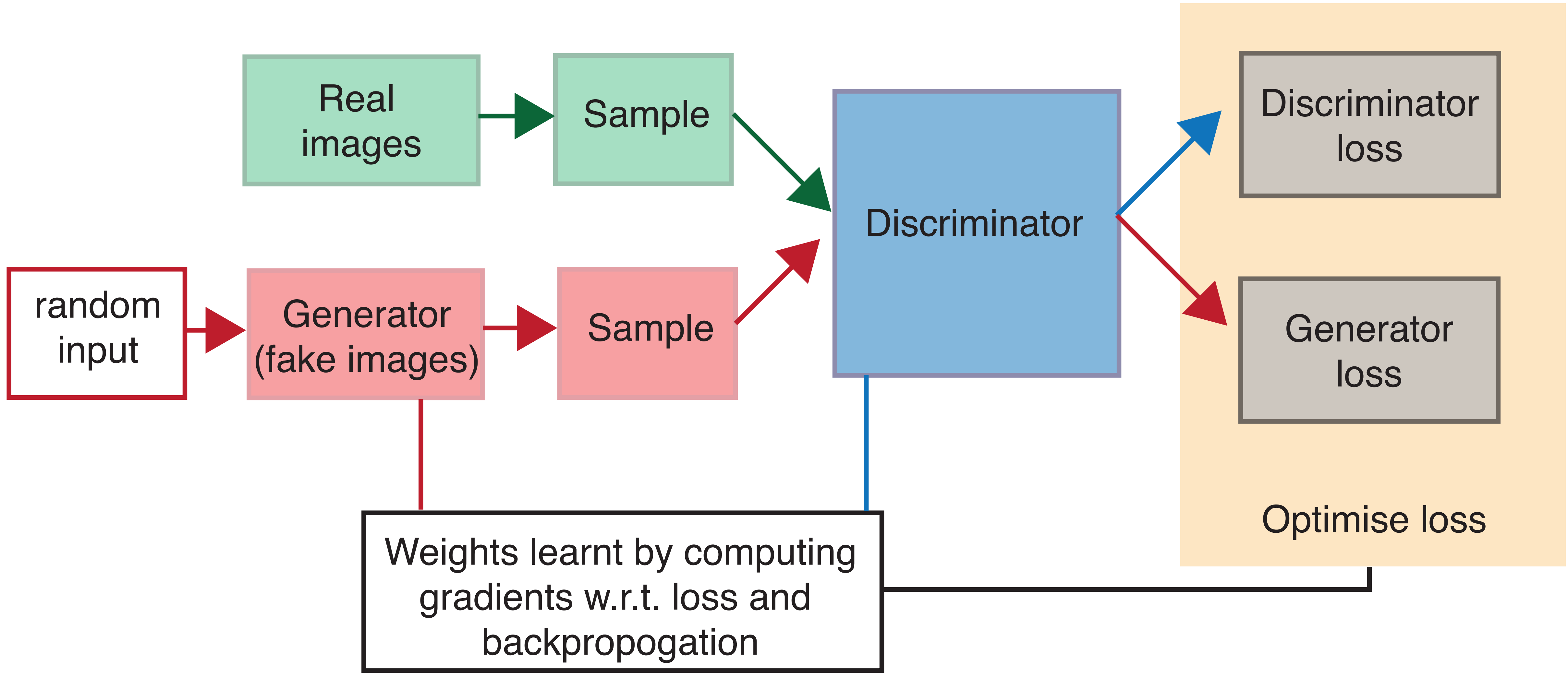
Used mainly for generative tasks such as image generation, data generation, etc.

Gaussian Process Latent Variable Models

Similar applications as auto encoders.

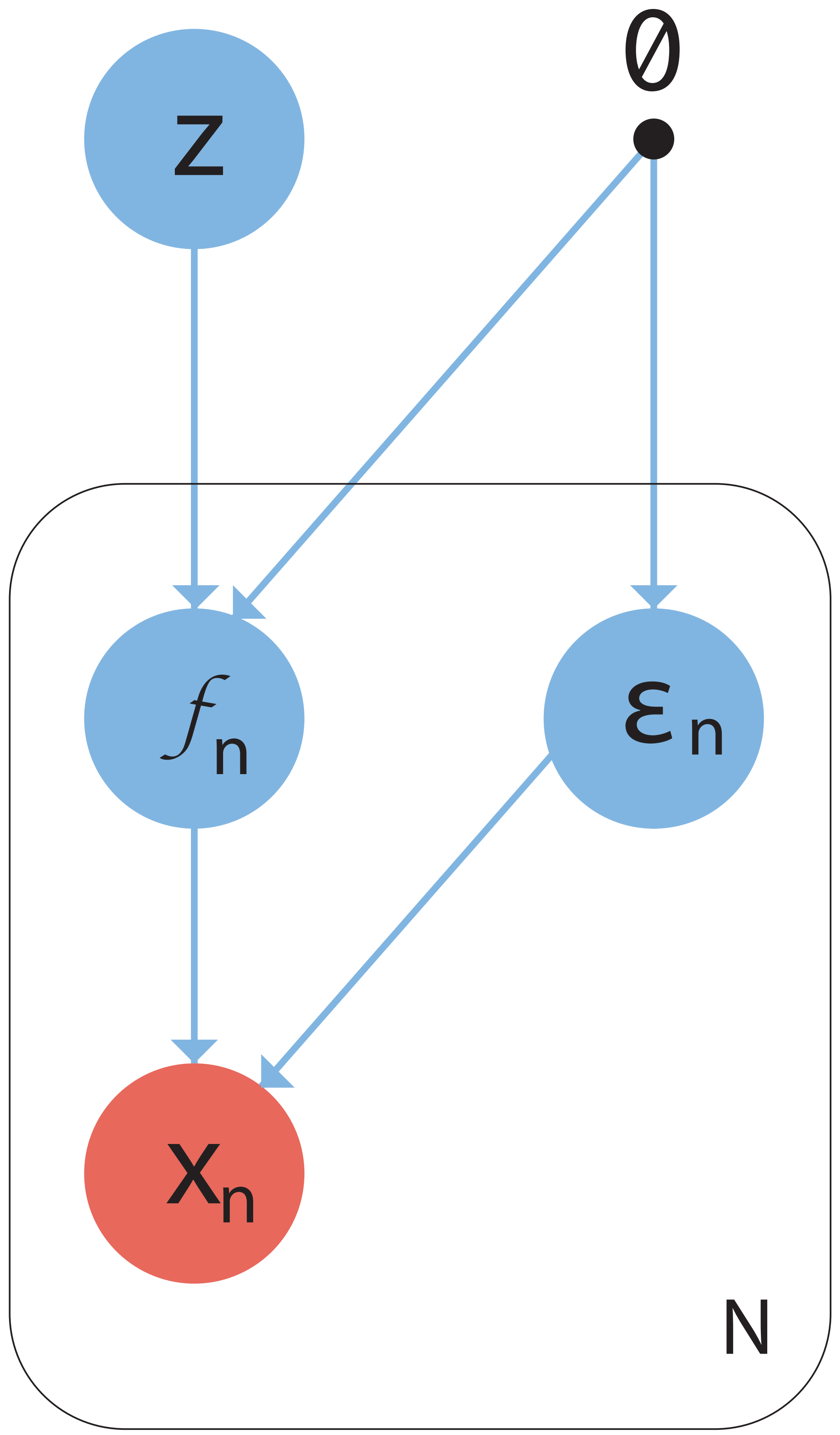
Generative Adversarial Network

- GANs are a popular adversarial technique.
- The generator learns to generate plausible data.
- The discriminator tries to distinguish the generator's fake data from the real data. Penalises generator for producing implausible results.



Gaussian Process Latent Variable Model

- Learns a low-dimensional latent representation, Z.
- It is a special case of PPCA where output dimensions are linear and i.i.d.
- GPLVM removes assumptions of linearity.
- Originally for Gaussian distributed data. Extension proposed for multi-likelihood setting.



siddharth.ramchandran@aalto.fi

Latent modelling - learning low dimensional representations

siddharth.ramchandran@aalto.fi

Recap - Principal Component Analysis

- A popular technique for linear dimensionality reduction.
- Projects a number of possibly correlated variables into a smaller number of uncorrelated variables (principal components).
- $X* = XP*$ where $P*$ is the matrix of eigenvectors sorted in decreasing order of eigenvalues.

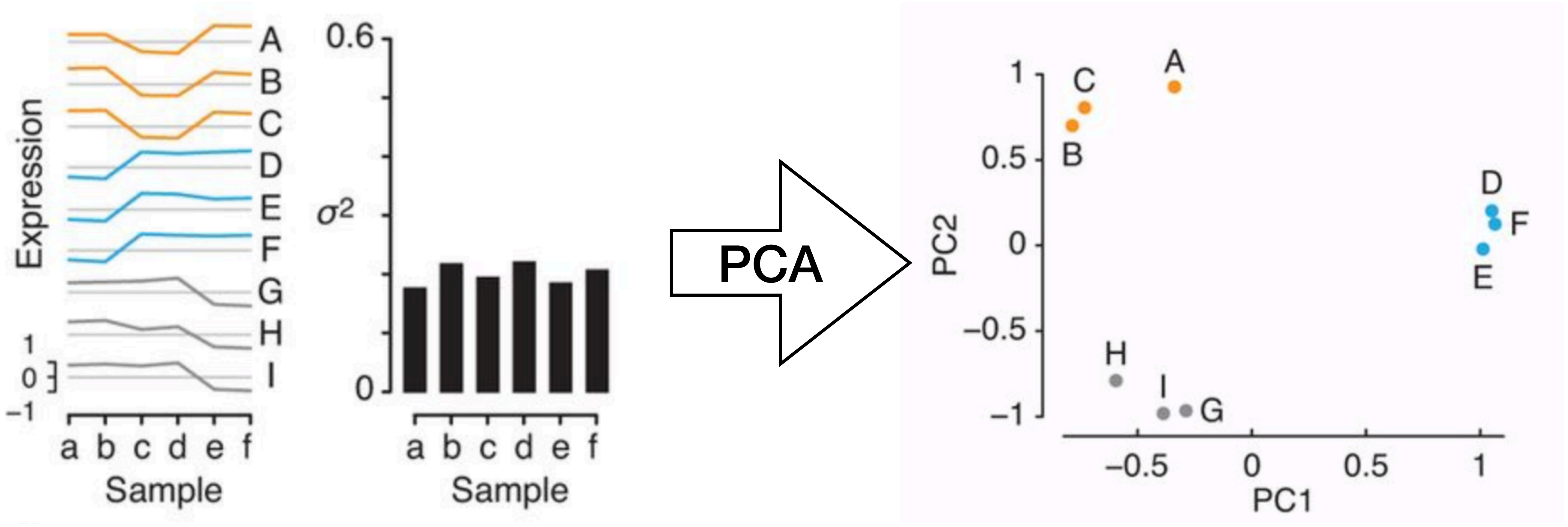


Figure modified from (Lever et al. 2019)

- Not probabilistic as it has no likelihood model
- Does not handle missing values properly
- Computationally intensive
- Not robust to outliers

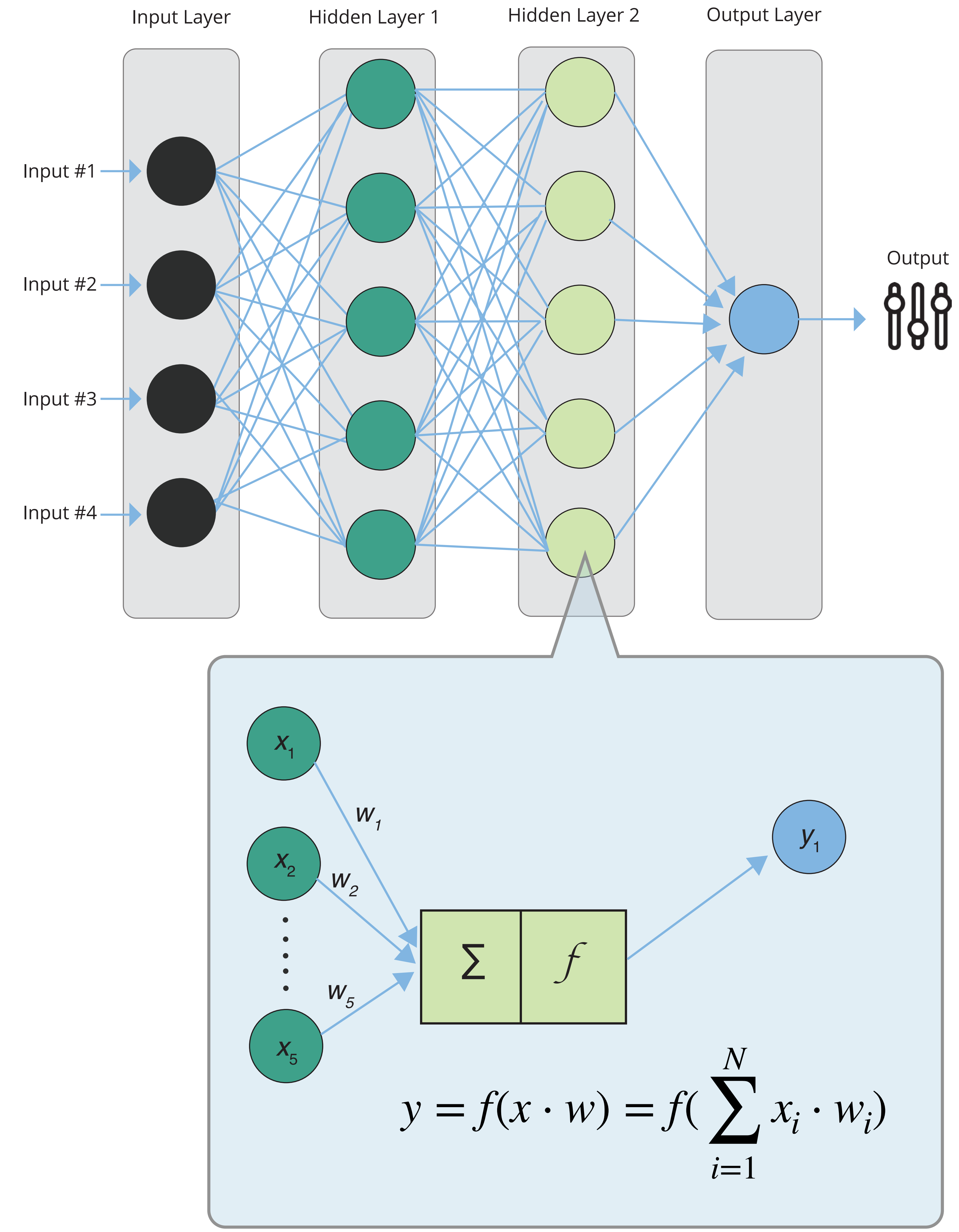
Recap - Artificial Neural Networks

- A Multi-Layered perceptron is one of the simplest forms of an Artificial Neural Network.

- The *forward pass* refers to the computation of the output given the input and weights.

- Need to learn the weights given training examples (i.e. train the network).

Architecture of simple Multi-Layered Perceptron

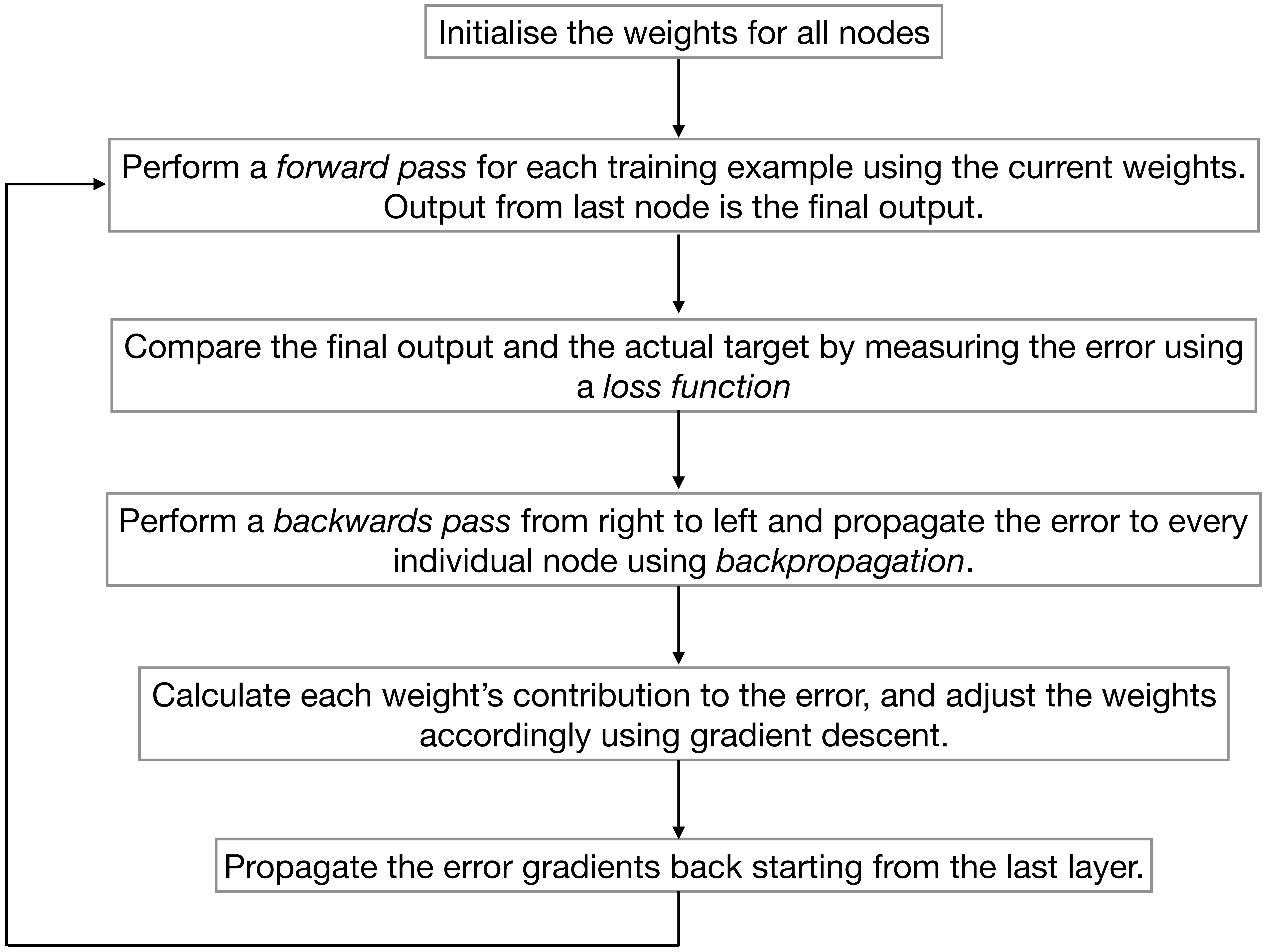


$$y = f(x \cdot w) = f(\sum_{i=1}^{N} x_i \cdot w_i)$$

siddharth.ramchandran@aalto.fi

Learning the weights of a MLP

- *Backpropagation* with *gradient descent* is the key ingredient in deep learning.
- The learning is repeated over several iterations or *epochs*.
- Training once the *loss function* does not decrease further (appears to have converged). However, learning is not necessarily complete.



Initialise the weights for all nodes

Perform a *forward pass* for each training example using the current weights. Output from last node is the final output.

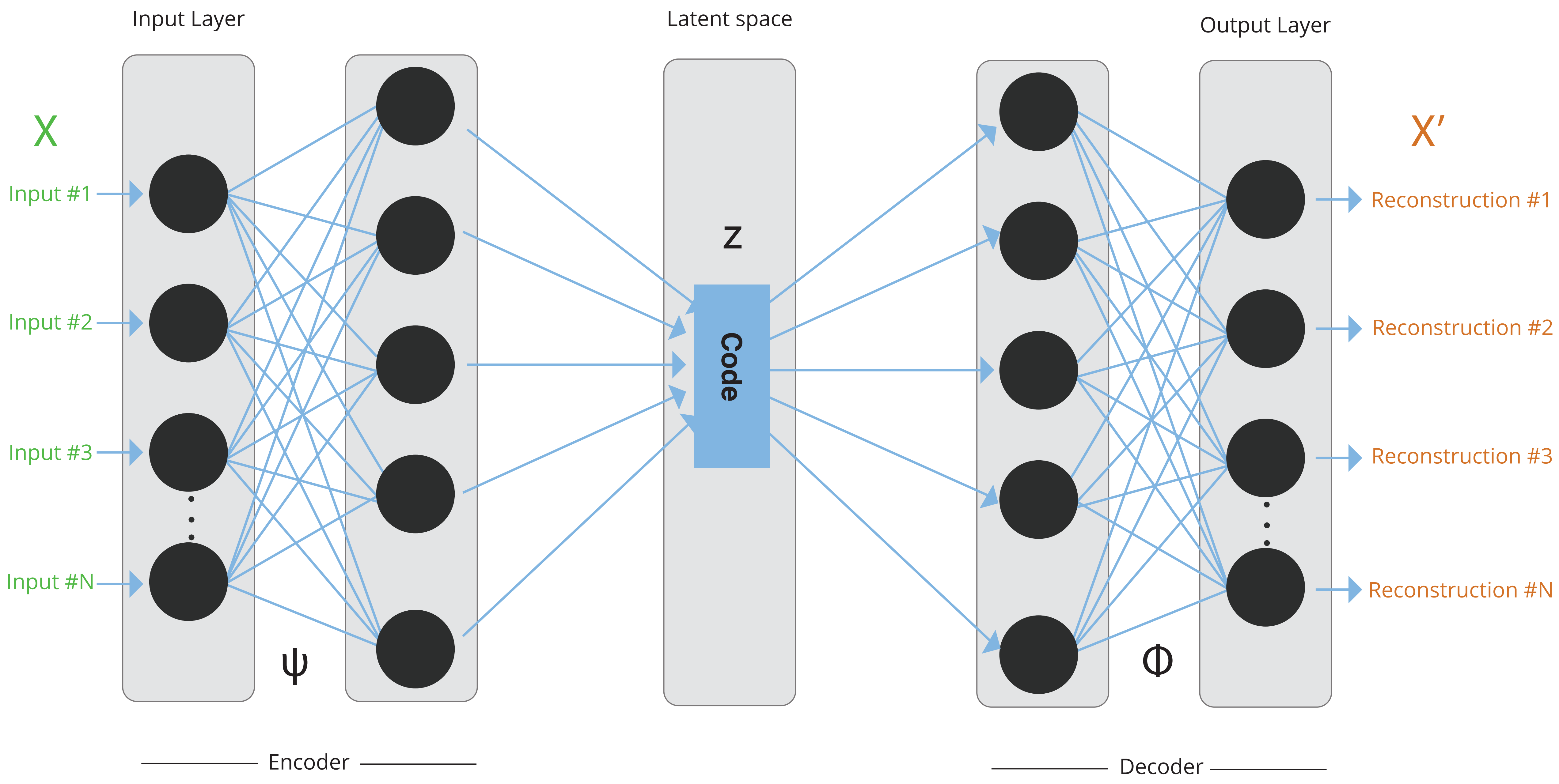
Compare the final output and the actual target by measuring the error using a *loss function*

Perform a *backwards pass* from right to left and propagate the error to every individual node using *backpropagation*.

Calculate each weight's contribution to the error, and adjust the weights accordingly using gradient descent.

Propagate the error gradients back starting from the last layer.

The Autoencoder



$$\text{Loss} = \| X - X' \|^2 = \| X - d(Z) \|^2 = \| X - d(e(X)) \|^2$$

- An ANN that learns to copy its input to the output. It has an internal latent layer (*code*) that can act as an information bottleneck.
- The *latent space* preserves only the most relevant aspects of the data.
- The reconstruction error (*loss*) is minimised by gradient descent over the parameters of the two neural networks (i.e. *Backpropagation* of the error).

Vanilla Autoencoders are not enough …

Autoencoders are great because:

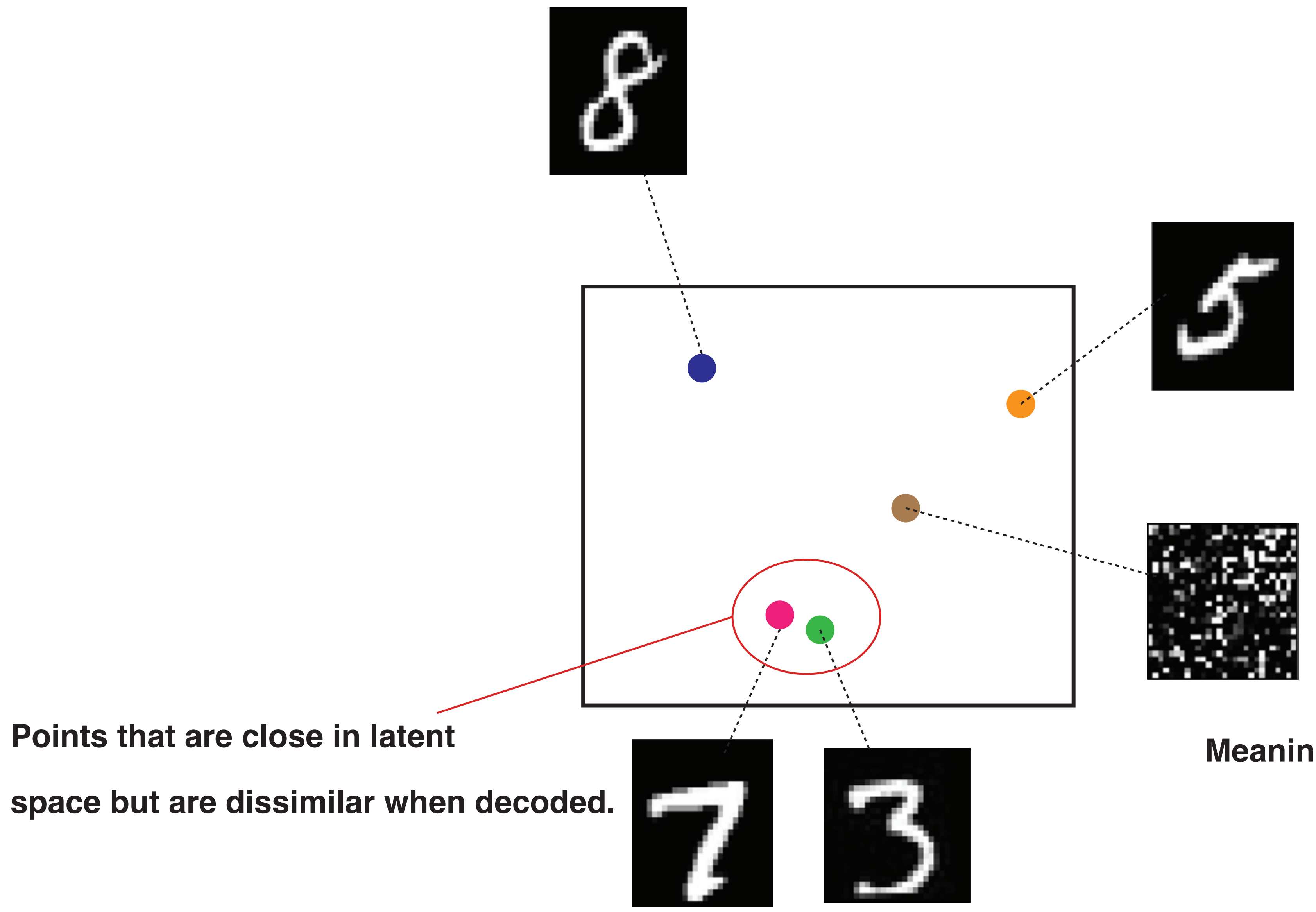
- Identifies compact representations and reconstruct their inputs well.
- Fast to train and are a simple concept that leverages the power of DL.

However, we can do better:

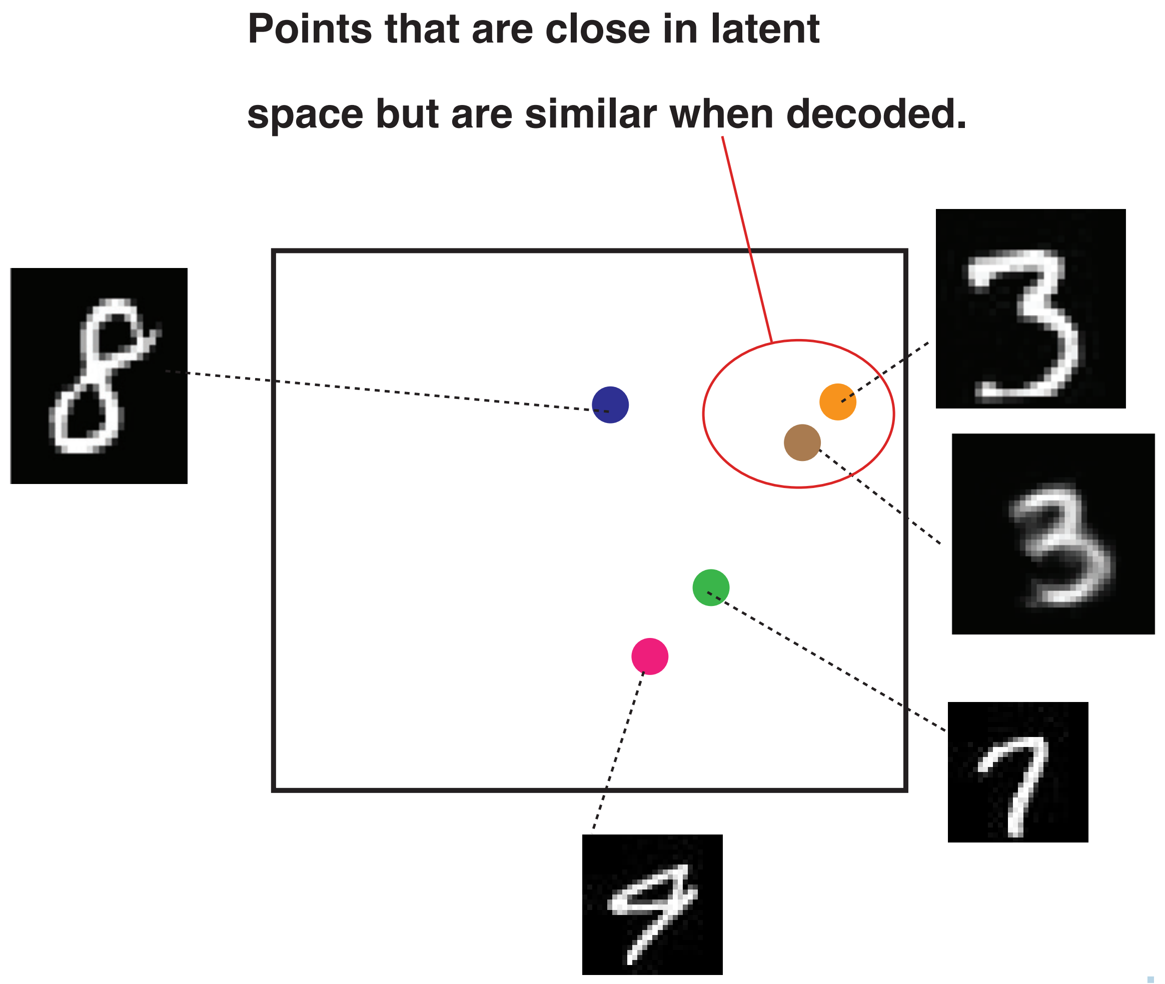
- Explore variations of existing data in a desired way by exploring the latent space. Hard to use the decoder for generative purposes.
- The latent space of vanilla Autoencoders may not be continuous or allow easy interpolation. There is a lack of structure in the latent space.

Irregular vs. Regular latent space

- **Continuity:** two points that are close in the latent space do not give completely dissimilar reconstructions.
- **Completeness:** a point sampled from the latent space should give meaningful content once decoded.


Points that are close in latent space but are dissimilar when decoded.
Meaningless reconstruction

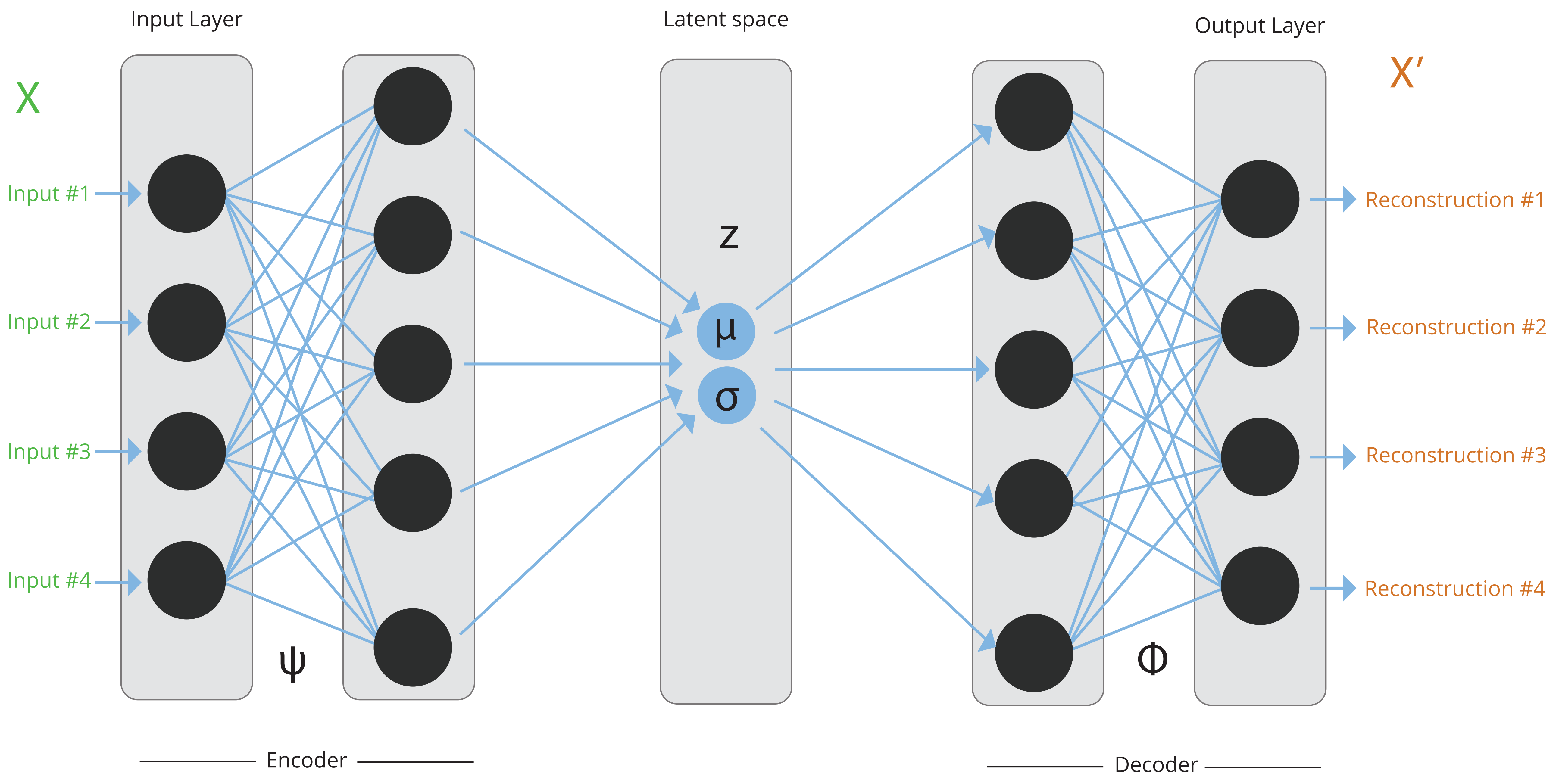
Irregular latent space (overfit)


Points that are close in latent space but are similar when decoded.

Regular latent space

The Variational Autoencoder

- An input is encoded as a distribution over the latent space instead of a single point.
- The training (i.e. the mean and covariance) is regularised to avoid overfitting.
- The encoded distributions are Gaussians. Hence, the encoder returns the mean and covariance matrix that describes the Gaussians.



input $\xrightarrow{\text{encode}}$ **latent distribution** $\xrightarrow{\text{sample}}$ **sampled latent representation** $\xrightarrow{\text{decode}}$ **reconstruction**

X $\qquad p(Z|X)$ $\qquad z \sim p(Z|X)$ $\qquad x' = d(z)$

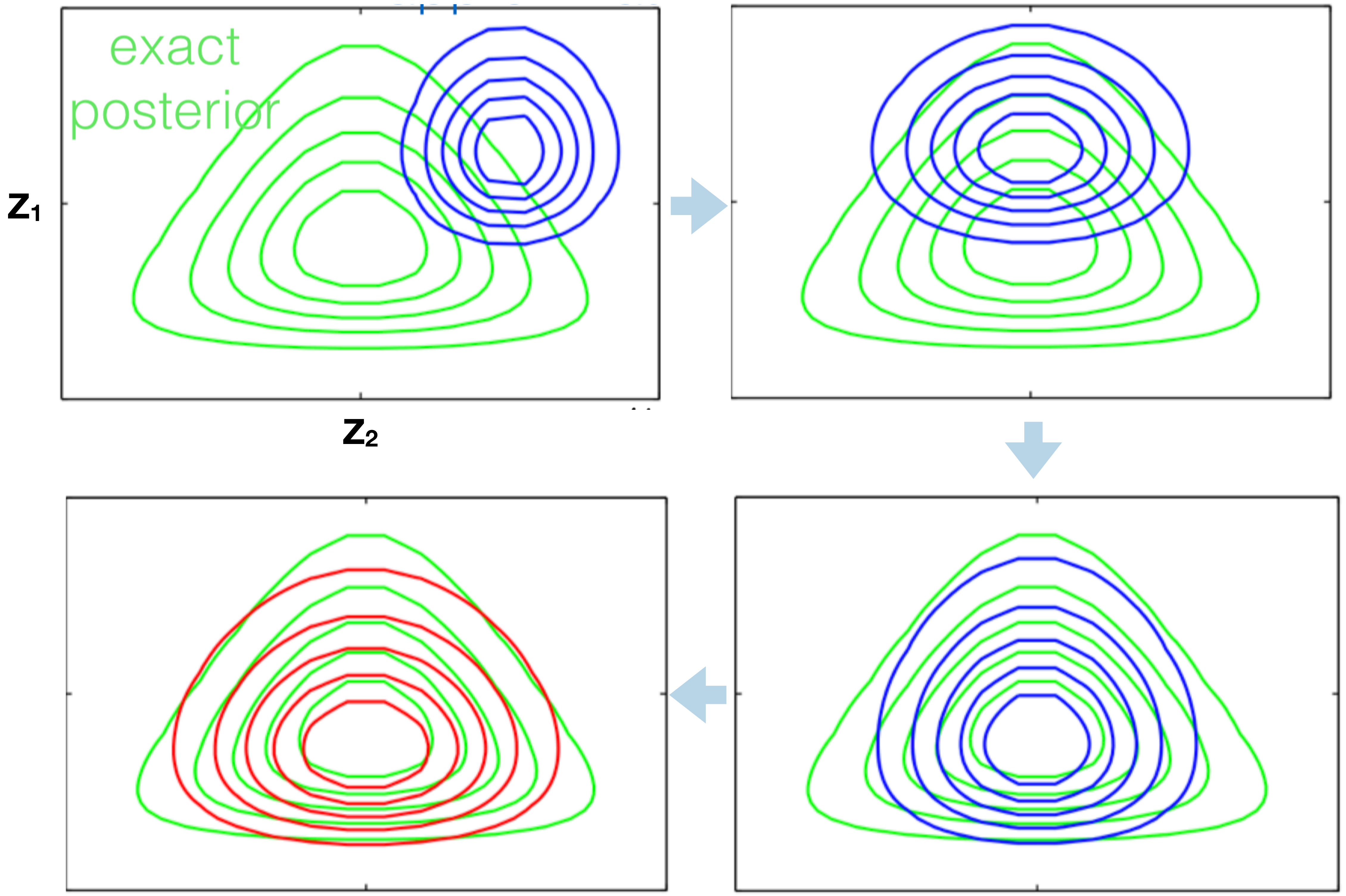
The regularised loss

$$\textbf{Loss} = \overbrace{||X - \mathbf{d}(\mathbf{e}(X))||^2}^{\text{Reconstruction loss}} + \overbrace{KL(\underbrace{q(z|y)}_{\mathcal{N}(\mu,\sigma^2)\ \textbf{(from encoder)}}||\underbrace{p(z)}_{\mathcal{N}(0,\boldsymbol{I})})}^{\text{Regularisation term}}$$

- The latent space is regularised by forcing the distributions returned by the encoder to be close to the standard normal distribution (*regularisation term*).
- Meanwhile, the reconstruction term encourages better reconstruction performance.
- The regularisation term is vital to obtain *continuity* and *completeness*.

A statistical view of VAEs: Variational Inference

- Variational inference is a technique to approximate complex distributions.
- Makes use of the Kullback-Leibler divergence between an approximation and target to choose the best from a family of approximations.
- Best approximation found by gradient descent over the parameters that describe the family.



Variational inference in VAEs

- We approximate $p(Z|X)$ with a Gaussian distribution, $q_\psi(Z)$ whose mean and covariance are from the encoder neural network ($q_\psi(Z|X)$ as this amortised V.I.)
- Recall that ψ and Φ corresponds to the encoder weights and decoder weights.
- Makes use of the *reparametrisation trick*: $z = \mu_\psi(x) + \epsilon\sigma_\psi^2(X)$ where $\epsilon \sim \mathcal{N}(0,I)$

$$q_\psi(Z) = \mathcal{N}(\mu_\psi(X), \sigma_\psi^2(X))$$

Minimise KL divergence between approximate and target distribution:

$$(\mu_\psi^*, \sigma_\psi^{2*}, \psi^*, \Phi^*) = \textbf{arg min}_{\mu_\psi, \sigma_\psi^2, \psi, \Phi} KL(\underset{\text{approximation}}{\underline{q_\psi(Z)}}\,||\,\underset{\text{target}}{\underline{p(Z|X)}})$$

The KL divergence can be re-written as:

$$= \textbf{arg min}_{\mu_\psi, \sigma_\psi^2, \psi}(\mathbb{E}_{z\sim q_\psi}(\log q_\psi(z)) - \mathbb{E}_{z\sim q_\psi}(\log \underset{\text{Baye's theorem on } p(Z|X)}{\underline{\frac{p(x|z)p(z)}{p(x)}}}))$$

Expanding above:

$$= \textbf{arg min}_{\mu_\psi, \sigma_\psi^2, \psi}(\mathbb{E}_{z\sim q_\psi}\underline{(\log q_\psi}(z)) - \mathbb{E}_{z\sim q_\psi}\underline{(\log p(z)}) - \mathbb{E}_{z\sim q_\psi}(\log p(x|z)) + \mathbb{E}_{z\sim q_\psi}\underline{(\log p(x)})$$

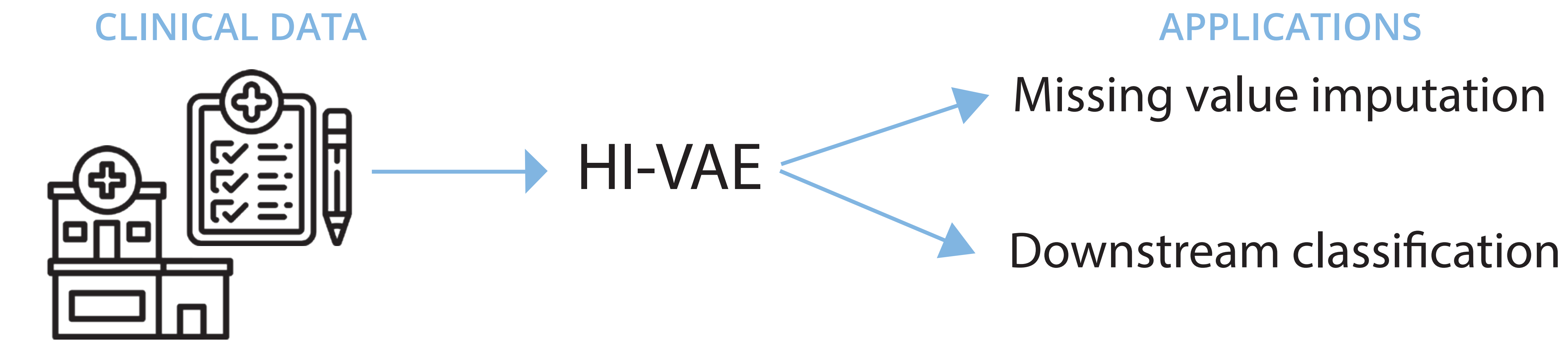
Using the KL definition:

$$= \textbf{arg max}_{\mu_\psi, \sigma_\psi^2, \psi}(\overset{\text{Reconstruction loss (-MSE)}}{\overline{\underset{\text{Using reparameterisation trick}}{\underline{\mathbb{E}_{z\sim q_\psi}}}[\log p(x|z)]}} - \overset{\text{Regularisation term}}{\overline{KL(q_\psi(z)\,||\,p(z))}})$$

Recent advancements

siddharth.ramchandran@aalto.fi

Handling incomplete heterogeneous data using VAEs



CLINICAL DATA

HI-VAE

APPLICATIONS

Missing value imputation

Downstream classification

0, 1, 1, 0, 0,	-0.5, 1.9, 0.23,	0.1, 0.2
1, 1, 1, 0, 0,	1.2, 2.3,1.2,	0.8, 0.87
0, 0, 0, 0, 0,	1.3, -0.1,Null,	0.2, 0.3
....		
1, 0, 0, 1, 1,	1.5, 2.4, 1.5,	0.9, 0.22

Patient records comprising of Binomial, Gaussian and Beta distributed data and missing values.

- Traditional VAEs do not handle heterogeneous (different likelihoods) or missing data.
- A VAE assumes that the data is Gaussian distributed.
- HI-VAE includes likelihood models for real-valued data, count data, categorical data, and ordinal data.
- Incomplete data is handled by *input drop-out recognition* distribution.

siddharth.ramchandran@aalto.fi

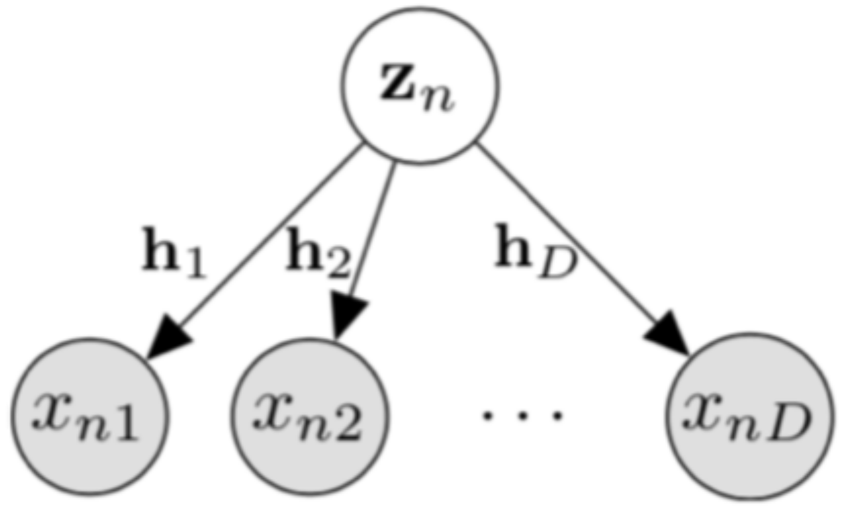
HI-VAE: Heterogeneous data

- Accommodate a variety of likelihood models - one per attribute.
- A deep neural network (the decoder) $h_d(\cdot)$, models the likelihood parameters γ_{nd}
- Assumes a factorised decoder for simplicity.

Factorised decoder:

$$p(x_n, z_n) = p(z_n)\prod_d p(x_{nd} \mid z_n)$$

The generative model:



Every likelihood model can be written as:

$$p(x_{nd} \mid z_n) = p(x_{nd} \mid \gamma_{nd} = \boldsymbol{h}_d(z_n))$$

Gaussian likelihood model:

$$\gamma_{nd} = \{\mu_d(z_n), \sigma_d^2(z_n)\}$$

$$p(x_{nd} \mid \gamma_{nd}) = \mathcal{N}(\underline{\mu_d}(z_n), \underline{\sigma_d^2}(z_n))$$

From neural network

Poisson likelihood model:

$$\gamma_{nd} = \underline{\lambda_d}(z_n)$$

Mean parameter

$$p(x_{nd} \mid \gamma_{nd}) = \text{Poiss}(\underline{\lambda_d}(z_n))$$

From neural network

HI-VAE: Incomplete data

- The recognition model needs to be flexible enough to handle any combination of observed and missing attributes.
- The distribution of latent variables z_n depends only on the observed attributes x_n^o.
- The loss is only computed over the observed data.

Factorised decoder:

$$p(x_n, z_n) = p(z_n) \prod_d p(x_{nd} \mid z_n)$$

Separating the contributions:

$$p(\boldsymbol{x}_n \mid z_n) = \prod_{d \in \mathcal{O}_n} \underbrace{p(x_{nd} \mid z_n)}_{\text{Observed data}} \prod_{d \in \mathcal{M}_n} \underbrace{p(x_{nd} \mid z_n)}_{\text{Missing data}}$$

The recognition model:

$$q(z_n, \boldsymbol{x}_n^m \mid \boldsymbol{x}_n^o) = q(z_n \mid \boldsymbol{x}_n^o) \prod_{d \in \mathcal{M}_n} p(x_{nd} \mid z_n)$$

Input drop-out recognition distribution:

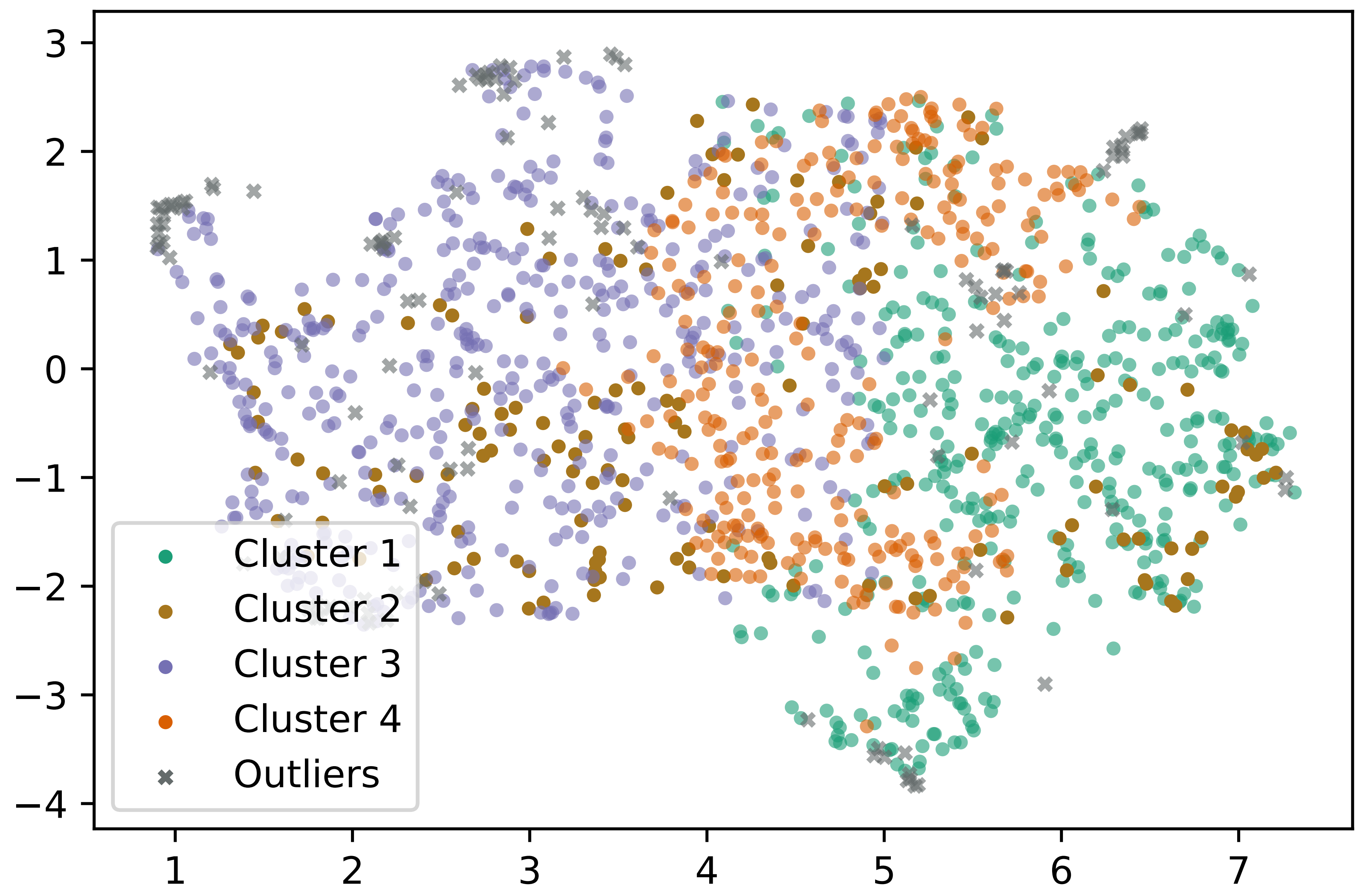
$$q(z_n \mid \boldsymbol{x}_n^o) = \mathcal{N}(\underline{\boldsymbol{\mu}_q}(\tilde{\boldsymbol{x}}_n), \underline{\boldsymbol{\Sigma}_q}(\tilde{\boldsymbol{x}}_n))$$

From deep neural network (encoder)

$\tilde{\boldsymbol{x}}_n$ resembles the original observed $\boldsymbol{x}_n$, but missing dimensions are replaced by zeros.

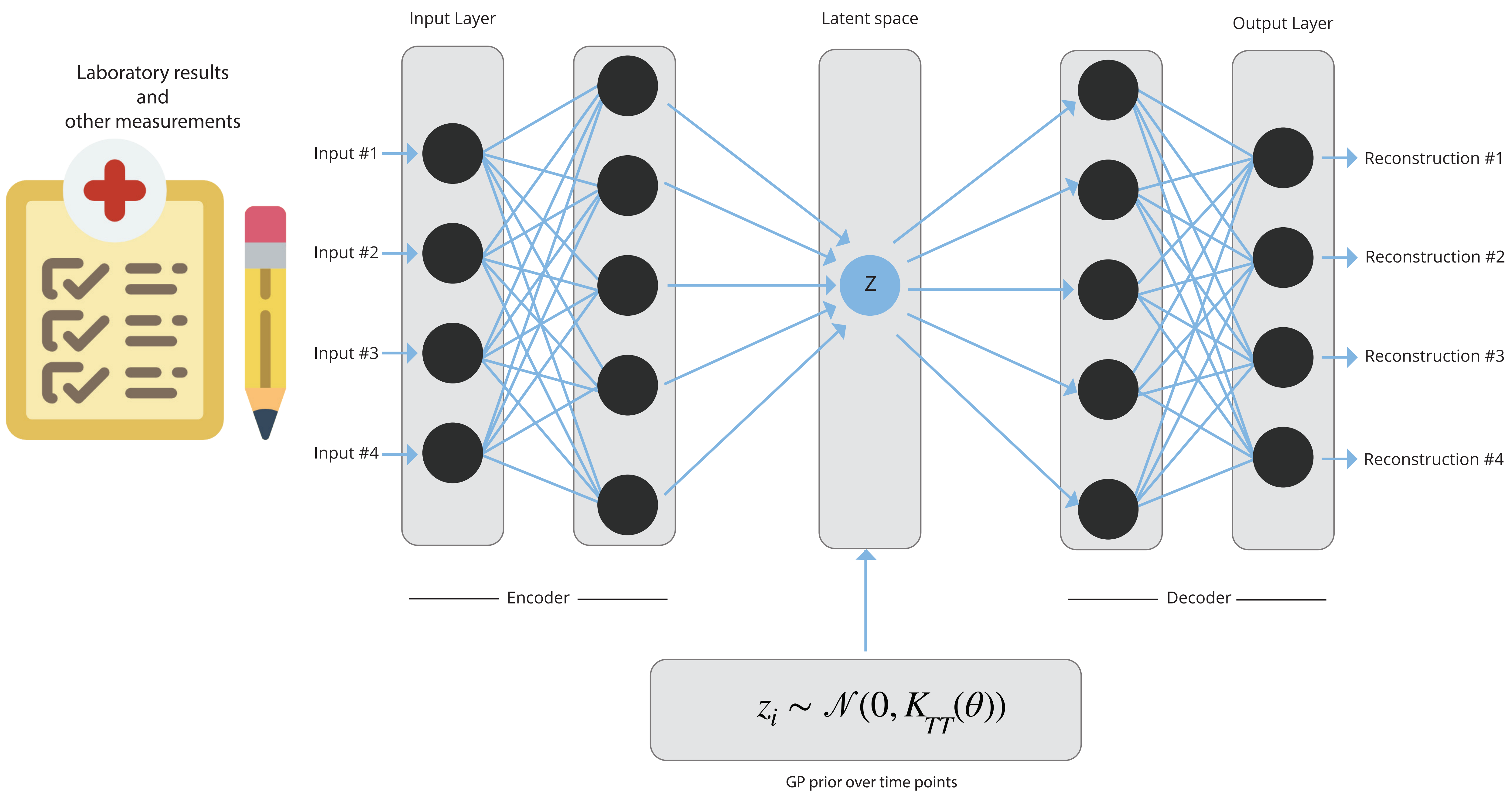
An Illustrative example

- Latent space learnt using a similar method from (Ramchandran et. al, 2019).
- Cluster labels identified using Bayesian GMM in a latent space with 8 dimensions.
- Projected on to a 2 dimensional space using UMAP.



Parkinson's data from the Helsinki Biobank

What about time series data?



- VAEs have been described for i.i.d datapoint with missing values.
- We can create low-dimensional representations of high-dimensional time series that evolves smoothly over time according to a Gaussian process.
- This can be used for reliable dimensionality reduction and data imputation.

siddharth.ramchandran@aalto.fi

Gaussian process prior VAE

From the vanilla VAE (loss to maximise):

$$\textbf{Loss} = \mathbb{E}_{z\sim q_\psi}[\log p(x\,|\,z)] - KL(q_\psi(z)\,||\,\underbrace{p(z)}_{\text{Replace with Gaussian process prior.}})$$

Replacing the prior with a GP prior computed over time:

$$p(Z\,|\,\theta) = \Pi_{i=1}^{L}\mathcal{N}(z_l\,|\,0,\underbrace{K_{TT}}_{\text{Squared exponential kernel}}(\theta))$$

θ corresponds to the GP hyper-parameters.

- Make use of a Gaussian process instead of a standard normal distribution as the prior on the latent space.
- Model temporal correlations in the reduced representation using a GP. This decouples the handling of missing values from instantaneous correlations between the different feature dimensions
- Kernel computed over time (age or time to disease).

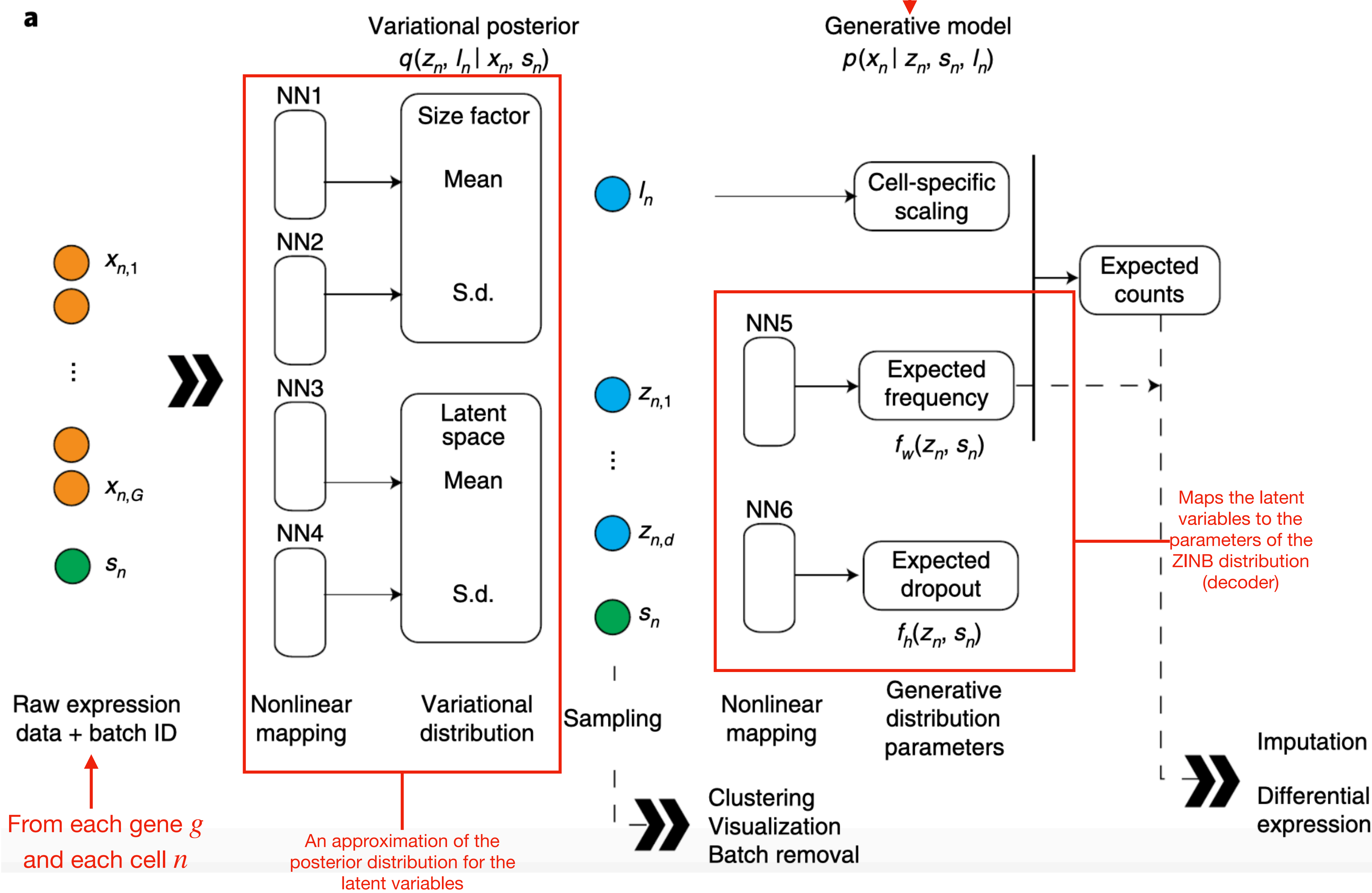
siddharth.ramchandran@aalto.fi

Single-cell variational inference

- ScVI is a fully probabilistic approach for the normalisation and analysis of scRNA-seq data. scRNA-seq can measure gene expression levels for each cell in a sample.
- Each cell is represented as a point in a low-dimensional latent space that can be used for visualisations and clustering.
- It is based on a hierarchical Bayesian model with the conditional distributions specified by deep neural networks.

Explicitly models 2 key nuisance factors: *library size* and *batch effects*.

Each x_{ng} modelled as a sample drawn from a zero-inflated negative binomial (ZINB) distribution



Maps the latent variables to the parameters of the ZINB distribution (decoder)

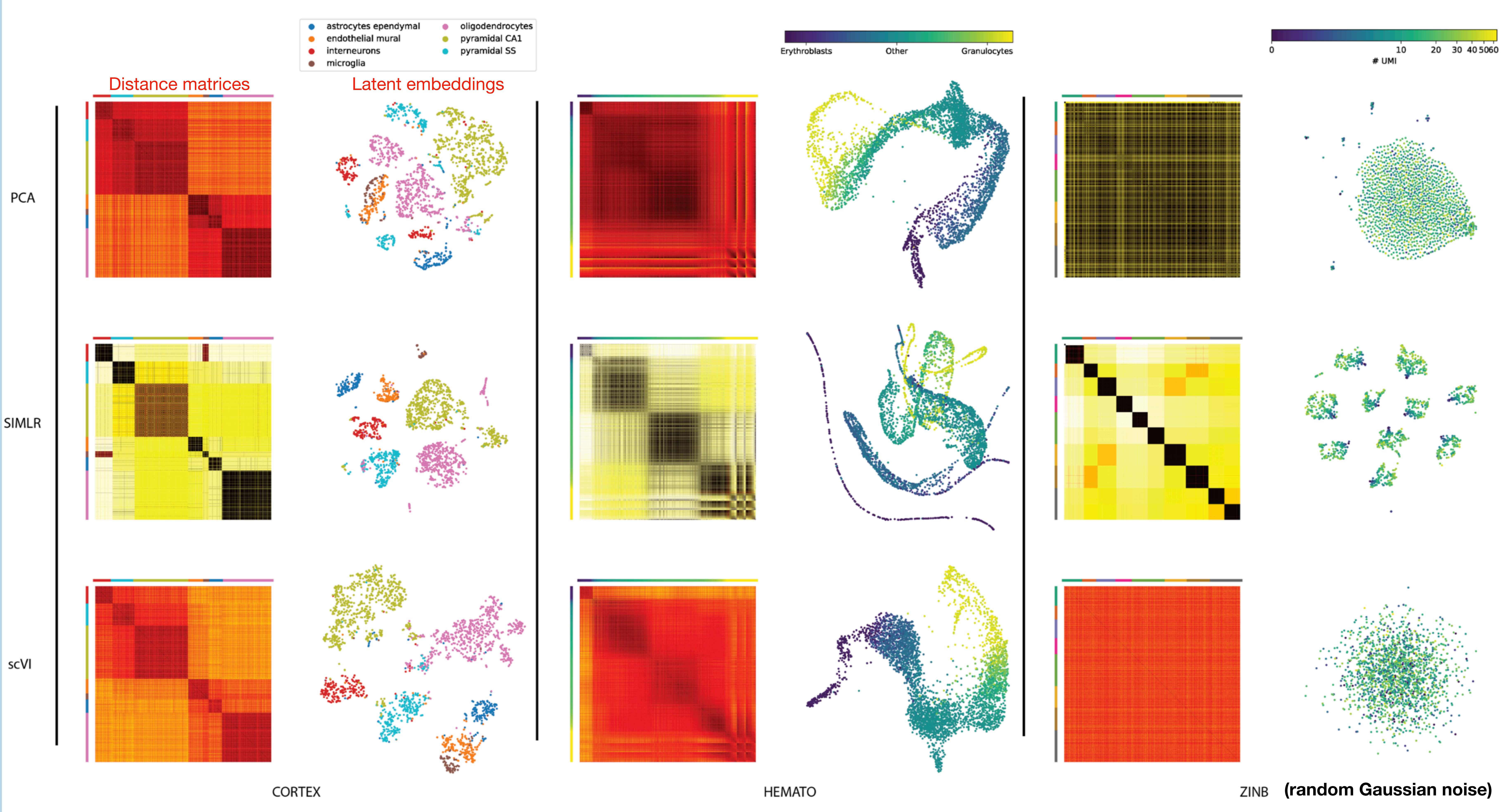
From each gene g and each cell n

An approximation of the posterior distribution for the latent variables (encoder)

ZINB combines the negative binomial distribution and the logit distribution.

Figure modified from (Lopez et al. 2019)

An illustrative example



Distance matrices

Latent embeddings

(random Gaussian noise)

Figure from (Lopez et al. 2019)

Useful resources

- **Introduction to biomedical data:** https://pm2.phs.ed.ac.uk/BDS/lecture01.pdf
- **Artificial neural networks:** https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6
- **Generative modelling:** https://towardsdatascience.com/generative-deep-learning-lets-seek-how-ai-extending-not-replacing-creative-process-fded15b0561b
- **GANs:** https://developers.google.com/machine-learning/gan
- **VAEs:** https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73
- **Variational Bayes:** https://people.csail.mit.edu/tbroderick/tutorial_2018_icml.html

References

- Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4401-4410).
- Woolson, R. F., & Clarke, W. R. (2011). *Statistical methods for the analysis of biomedical data (Vol. 371)*. John Wiley & Sons.
- Lever, J., Krzywinski, M. & Altman, N.(2017). Principal component analysis. *Nature Methods* 14, 641–642.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.
- Ramchandran, S., Koskinen, M., & Lähdesmäki, H. (2019). *Latent Gaussian process with composite likelihoods for data-driven disease stratification.* arXiv preprint arXiv:1909.01614.
- Casale, F. P., Dalca, A., Saglietti, L., Listgarten, J., & Fusi, N. (2018). Gaussian process prior variational autoencoders. In *Advances in Neural Information Processing Systems* (pp. 10369-10380).
- Fortuin, V., Rätsch, G., & Mandt, S. (2019). *Multivariate time series imputation with variational autoencoders.* arXiv preprint arXiv:1907.04155
- Nazabal, A., Olmos, P. M., Ghahramani, Z., & Valera, I. (2018). Handling incomplete heterogeneous data using VAEs. arXiv preprint arXiv:1807.03653.
- Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., & Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12), 1053-1058.

Some visual assets are from flaticon (https://www.flaticon.com) and unDraw (https://undraw.co).

Thank You.

Aalto University
School of Science
and Technology

siddharth.ramchandran@aalto.fi

Questions?

Aalto University
School of Science
and Technology

siddharth.ramchandran@aalto.fi