Machine learning in Exoplanets

Ariel Summer School, Biarritz
September 2023

/imagine prompt: A planetary system being observed by scientists with the help of machines, minimalist

Ingo Waldmann

The Alan Turing Institute

UK SPACE AGENCY

esa

erc

European Research Council

Established by the European Commission

Science & Technology Facilities Council

UCL

Progress on AI has accelerated significantly

Mainly due to an exponential increase in compute

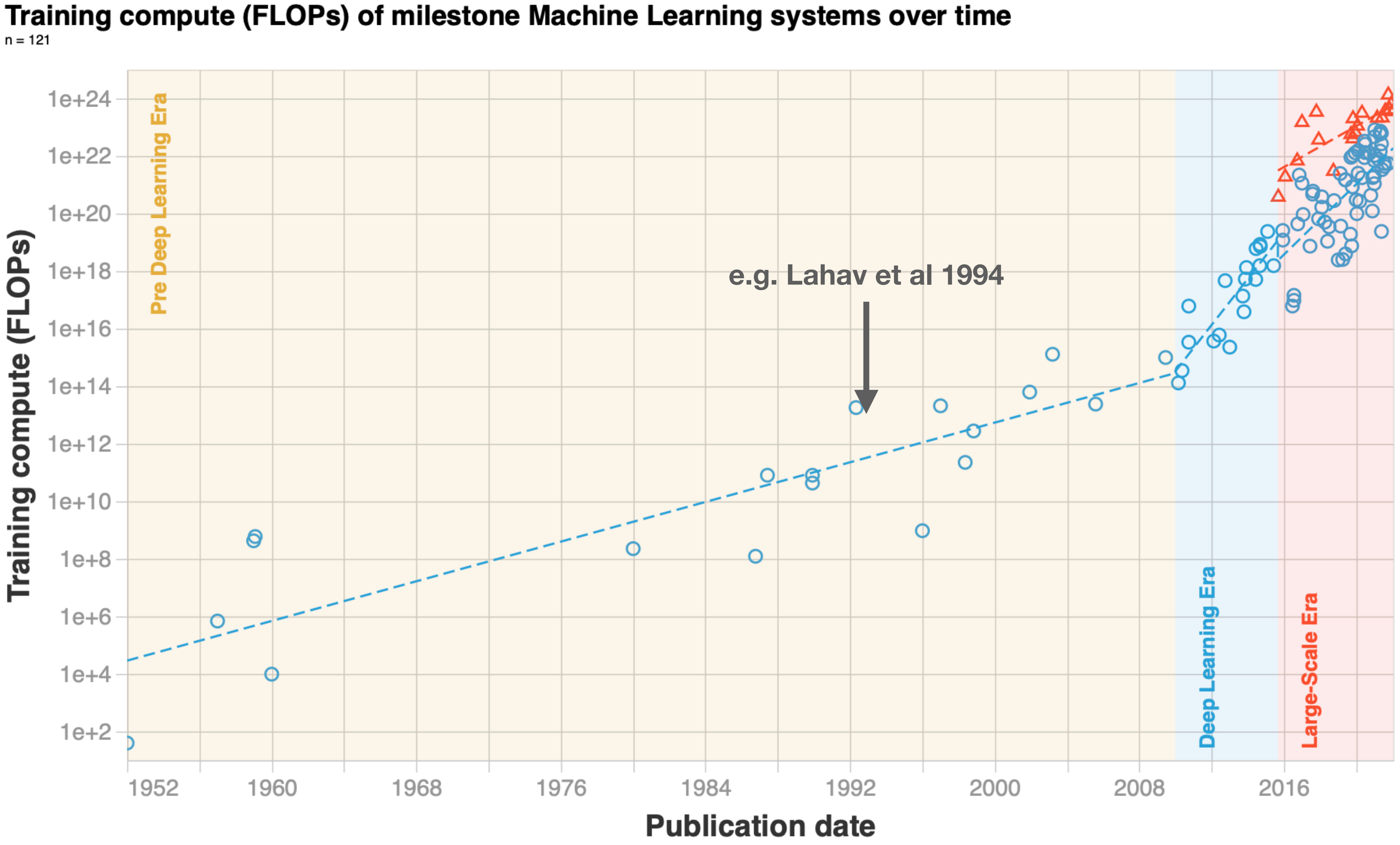


Figure 1: Trends in $n = 121$ milestone ML models between 1952 and 2022. We distinguish three eras. Notice the change of slope circa 2010, matching the advent of Deep Learning; and the emergence of a new large-scale trend in late 2015.

Savilla et al. 2022, arXiv: 2022:05924v2

GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models

Tyna Eloundou[1], Sam Manning[1,2], Pamela Mishkin*[1], and Daniel Rock[3]

March 27, 2023 arXiv: 2303.10130v4

Group	Occupations with highest exposure	% Exposure
Human α	Interpreters and Translators	76.5
GPT technology alone	Survey Researchers	75.0
	Poets, Lyricists and Creative Writers	68.8
	Animal Scientists	66.7
	Public Relations Specialists	66.7
Human β	Survey Researchers	84.4
GPT technology + some software augmentation	Writers and Authors	82.5
	Interpreters and Translators	82.4
	Public Relations Specialists	80.6
	Animal Scientists	77.8
Human ζ	Mathematicians	100.0
GPT technology + full software augmentation	Tax Preparers	100.0
	Financial Quantitative Analysts	100.0
	Writers and Authors	100.0
	Web and Digital Interface Designers	100.0

Humans labeled 15 occupations as "fully exposed."

Metric:
At least 50% of tasks will be automated/augmented

α = GPT only

β = GPT + 50% specialised software on top of GPT

ζ = GPT + specialised software on top of GPT

A significant number of tasks will be affected

Much of our work flow will change in the next years

Human estimate

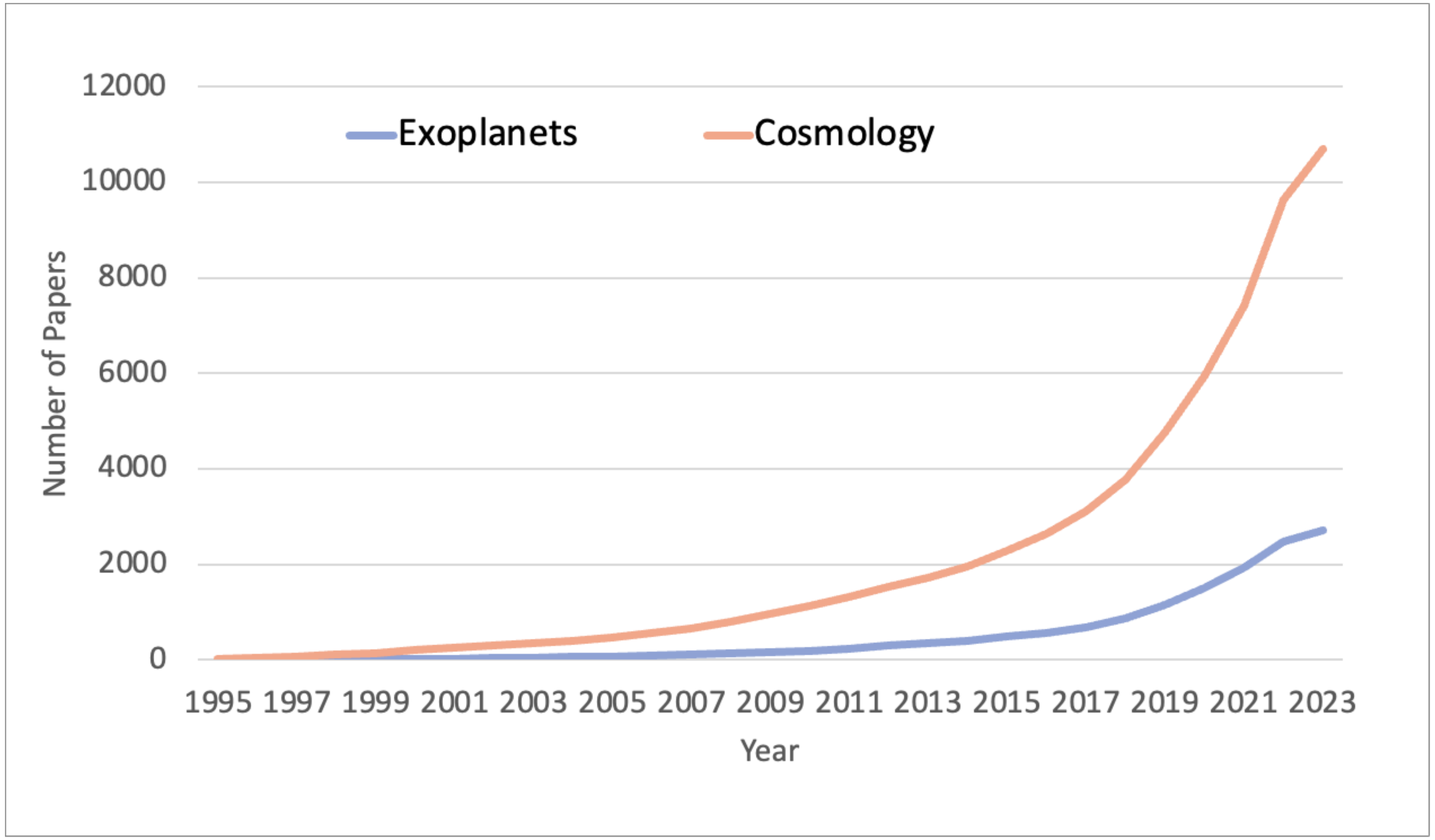
GPT4 estimate

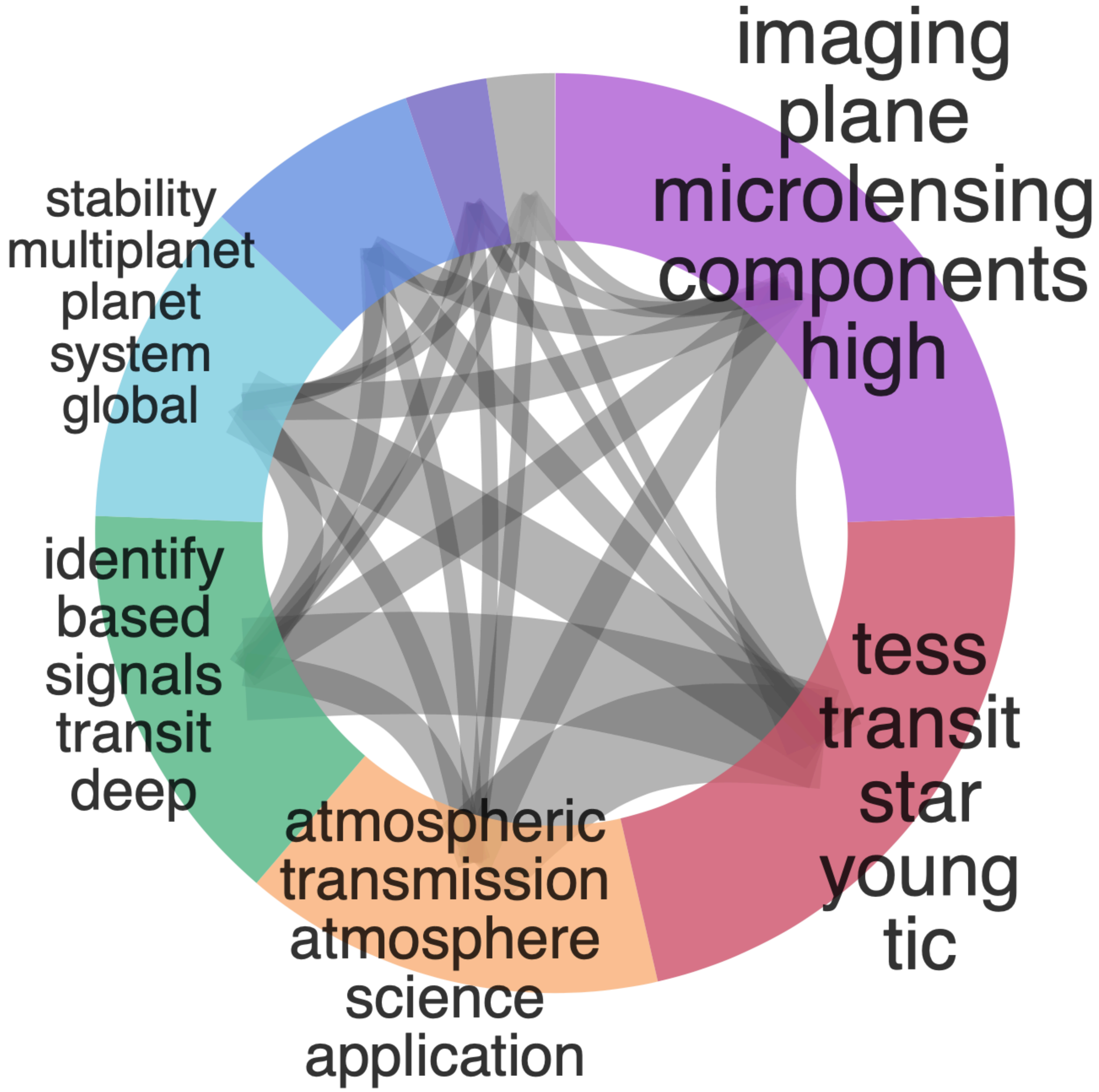
Job Zone	Preparation Required	Education Required	Example Occupations	Median Income	Tot Emp (000s)	H α	M α	H β	M β	H ζ	M ζ
1	None or little (0-3 months)	High school diploma or GED (otional)	Food preparation workers, dishwashers, floor sanders	\$30,230	13,100	0.03	0.04	0.06	0.06	0.09	0.08
2	Some (3-12 months)	High school diploma	Orderlies, customer service representatives, tellers	\$38,215	73,962	0.07	0.12	0.16	0.20	0.24	0.27
3	Medium (1-2 years)	Vocational school, on-the-job training, or associate's degree	Electricians, barbers, medical assistants	\$54,815	37,881	0.11	0.14	0.26	0.32	0.41	0.51
4	Considerable (2-4 years)	Bachelor's degree	Database administrators, graphic designers, cost estimators	\$77,345	56,833	0.23	0.18	0.47	0.51	0.71	0.85
5	Extensive (4+ years)	Master's degree or higher	Pharmacists, lawyers, astronomers	\$81,980	21,221	0.23	0.13	0.43	0.45	0.63	0.76

Table 6: Mean exposure to GPTs by job zone. For each job zone, we also present the median of median annual income for each constituting occupation in USD, and the total number of workers in all occupations for that job zone, in the thousands.

Eloundou et al. 2023

State of ML in Exoplanets

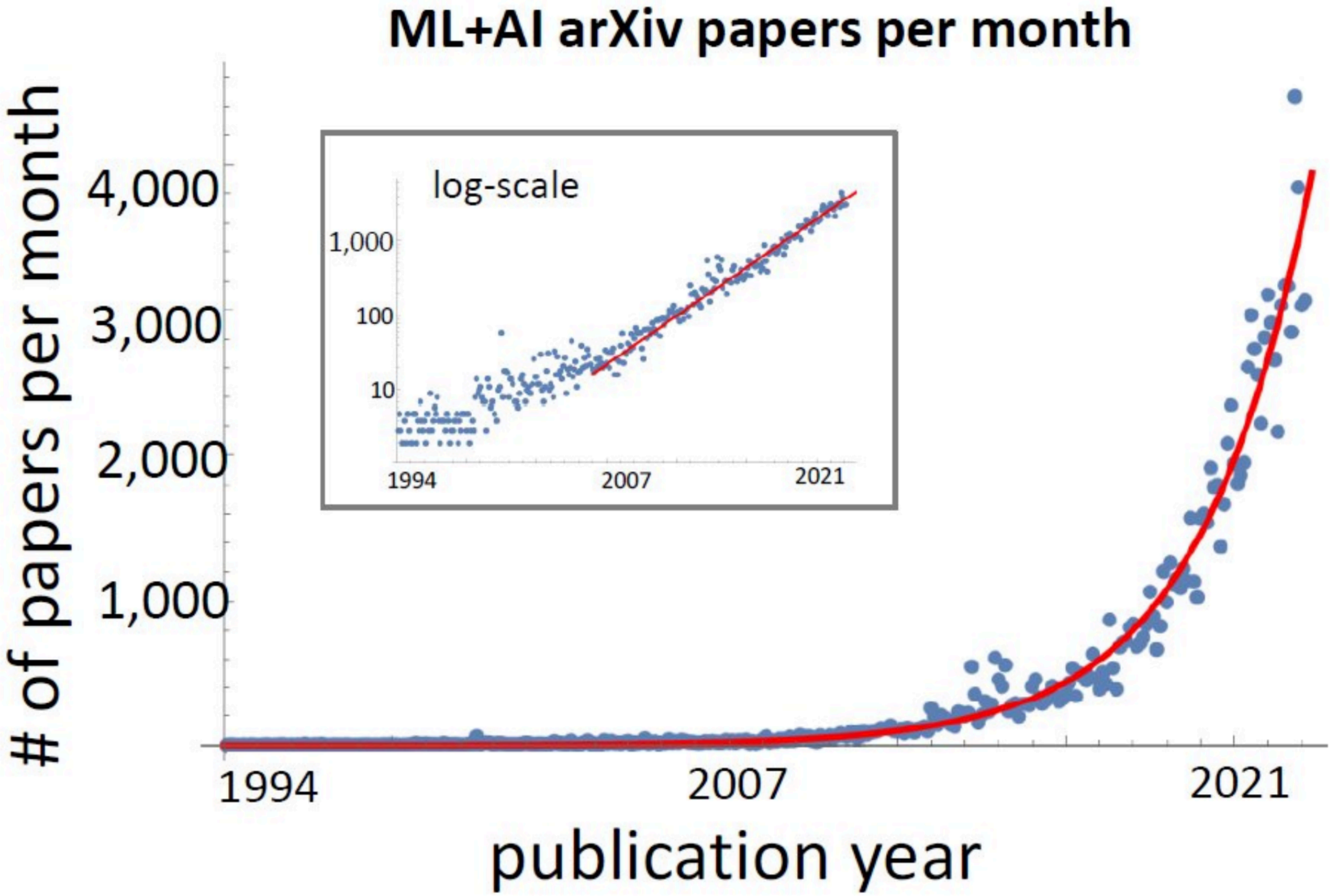




Search Term: (full:”extrasolar planet" or full:"exoplanet") and (full:"machine learning" or full:"artificial intelligence" or full:"neural network")

Generate using Astrophysics Data System

A lot of AI papers…



There's a lot of AI around …
We can't cover it all in 1.5 hours

What the Exoplanet Science addressable with AI?

- Realistic instrument noise simulations/ detrending
- Better data de-trending (instrumental noise and/or stellar)
- Faster and better inverse modelling (retrievals and light curve fitting)
- Faster generative models (e.g. chemistry, radiative transfer, circulation, condensation, etc)
- Many other things…

/imagine prompt: a woman working on a competition involving the discovery of new planets, with a blackboard in the background, filled with equations ,minimalist

Let's focus on some AI applications to Exoplanet Atmospheric retrievals

- What if we can train an AI to quickly and reliably classify and measure planet atmospheres?

Yip et al. 2022

A quick word on using AI

DON'T!

Only proceed if you really have to …

But if you have to use AI/ML
A quick cheat sheet:

- PCA, clustering and component separation, Random Forests…

    Use sklearn (https://scikit-learn.org/stable/index.html)

- Deep learning

    Use PyTorch (https://pytorch.org/)

- Probabilistic programming

    Use PyRo (https://pyro.ai/)

- Simulation based inference

    Use SBI (https://www.mackelab.org/sbi/)

- Great resources for models and tutorials

    HuggingFace (https://huggingface.co/)
    Papers With Code (https://paperswithcode.com/

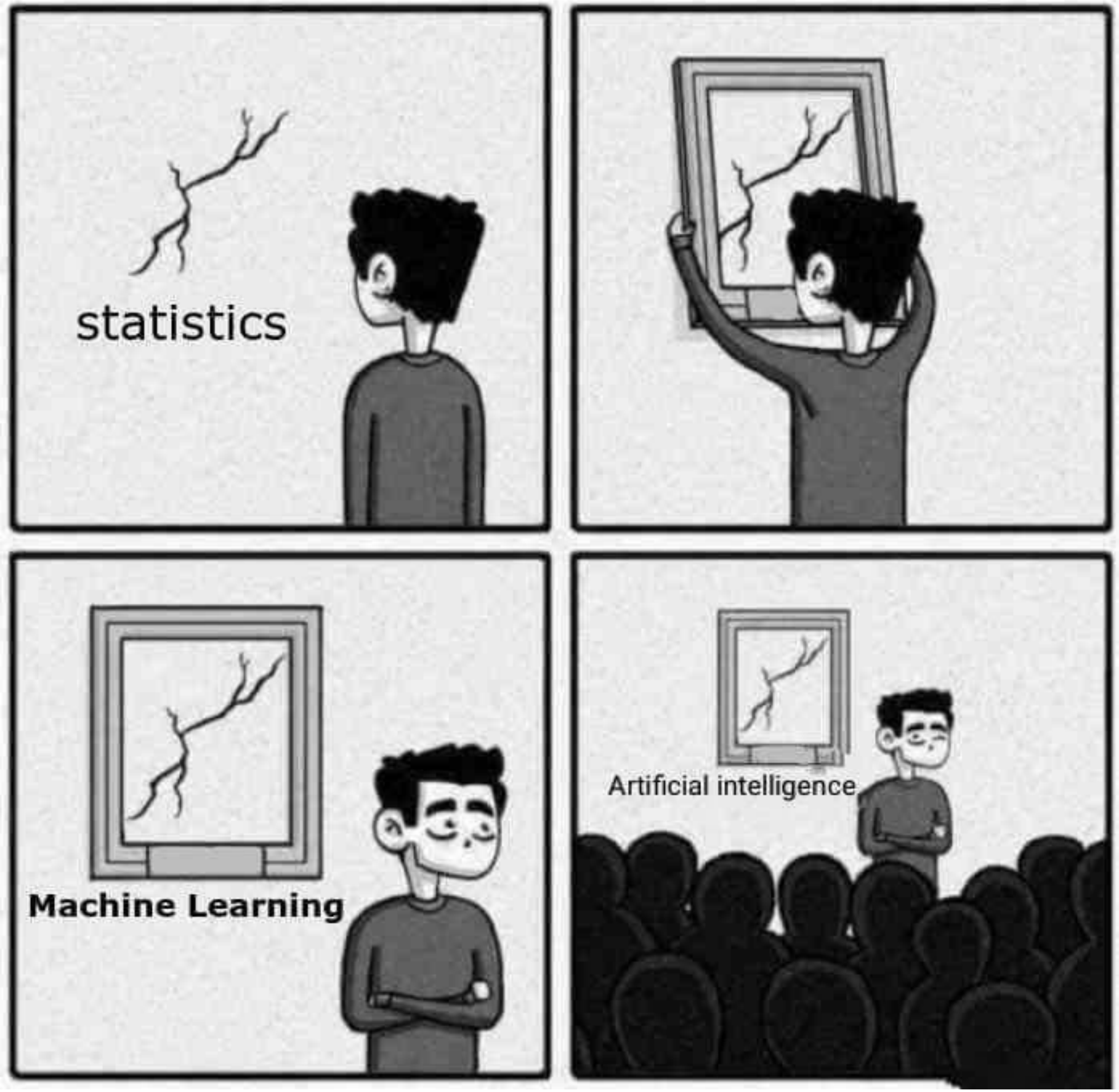
Agenda

Unsupervised learning

- Clustering (Nearest Neighbours, K-means)
- Component Separation (PCA, ICA)

(Self-)Supervised learning

- Random Forests
- Multi-layer perceptrons
- Autoencoders
- Bayesian Neural Networks
- Variational Inference
- Explainability



Let's reproduce some recent papers today

Supervised Machine Learning for Analysing Spectra of Exoplanetary Atmospheres

Reducing the complexity of chemical networks via interpretable autoencoders

T. Grassi[1,2,★], F. Nauman[3], J. P. Ramsey[4], S. Bovino[5], G. Picogna[1,2], and B. Ercolano[1,2]

…rte München, Scheinerstr. 1, D-81679 München, Germany
…rigin and Structure of the Universe, Boltzmannstr.2, D-85748 Garching bei München, Germany

Retrieving exoplanet atmospheric parameters using random forest regression

Patcharawee Munsaket[1*], Supachai Awiphan[2], Poemw…
and Eamonn Kerins[4]

[1]School of Physics, Institute of Science, Suranaree Univ…
Ratchasima 30000, Thailand

Disentangled Representation Learning for Astronomical Chemical Tagging

Damien de Mijolla[1], Melissa Kay Ness[2,3], Serena Viti[4,1], and Adam Joseph Wheeler[2]

[1] Department of Physics and Astronomy, University College London, Gower Street, WC1E 6BT, UK; ucapdde@ucl.ac.uk

Unsupervised Machine Learning for Exploratory Data Analysis of Exoplanet Transmission Spectra

RESEARCH ARTICLE

Molecular generative model based on conditional variational autoencoder for d… novo molecular design

Jaechang Lim[1], Seongok Ryu[1], Jin Woo Kim[1] and Woo Youn Kim[1,2*]

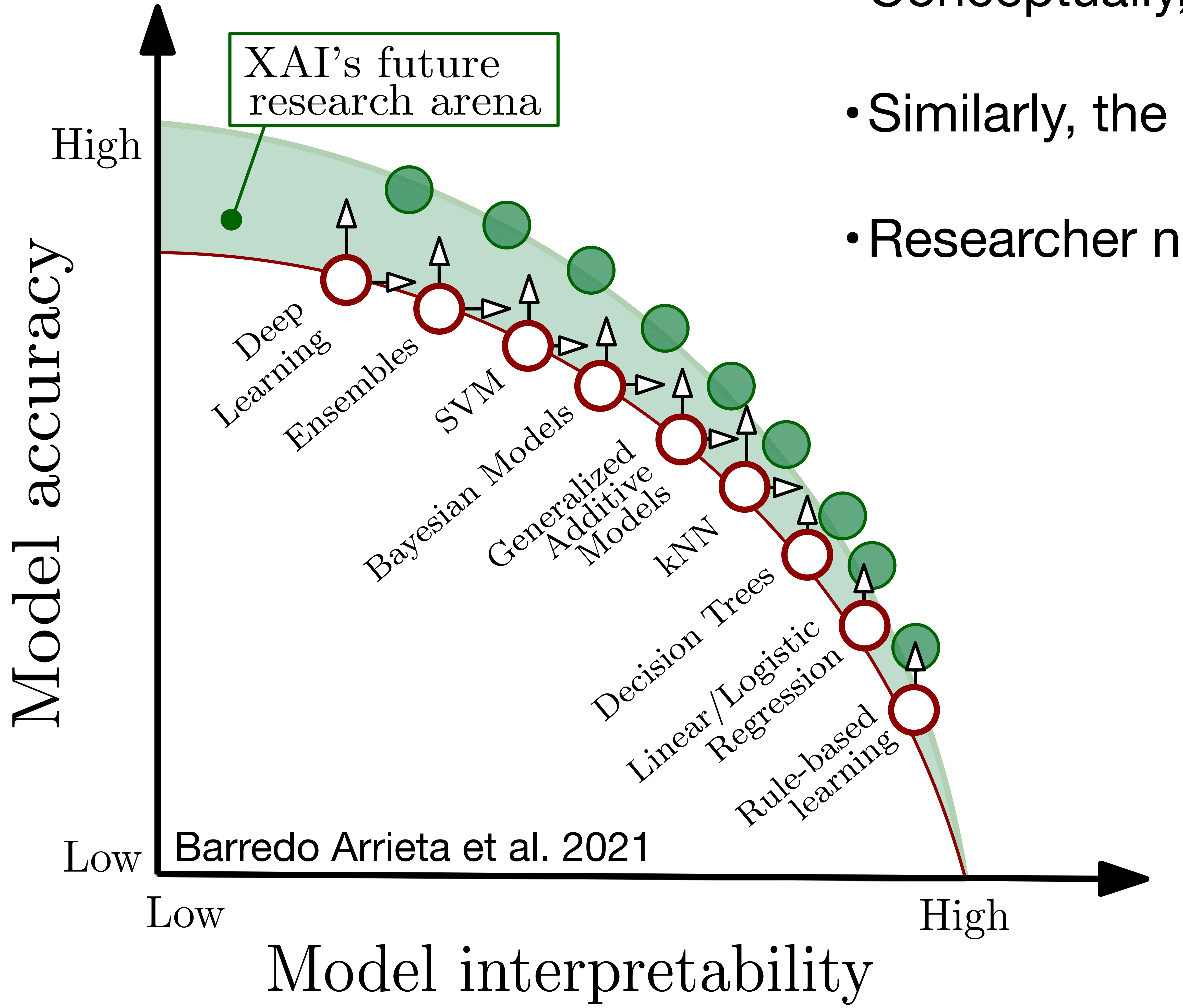
Accurate Machine-learning Atmospheric Retrieval via a Neural-network Surrogate Model for Radiative Transfer

Michael D. Himes[1], Joseph Harrington[2], Adam D. Cobb[3], Atılım Güneş Baydin[3], Frank Soboczenski[4], Molly D. O'Beirne[5], Simone Zorzan[6], David C. Wright[1], Zacchaeus Scheffer[1], Shawn D. Domagal-Goldman[7], and Giada N. Arney[7]

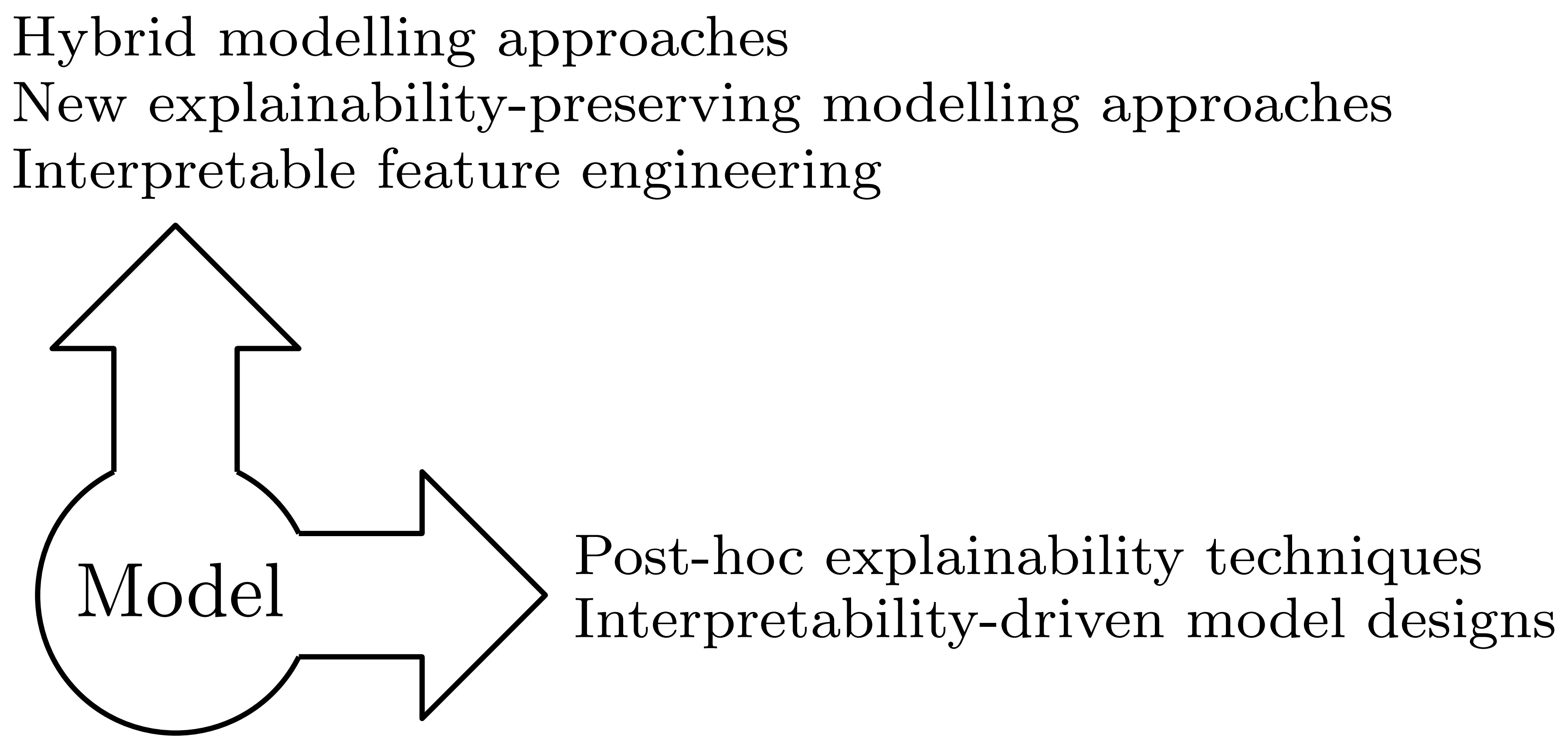
[1] Planetary Sciences Group, Department of Physics, University of Central Florida, USA; mhimes@knights.ucf.edu
[2] Planetary Sciences Group, Department of Physics and Florida Space Institute, University of Central Florida, USA

+ several others using similar techniques

A quick word on Explainability



- Conceptually, the more complex the model the harder to explain
- Similarly, the more complex the model, the more expressive
- Researcher needs to weigh up interpretability vs accuracy

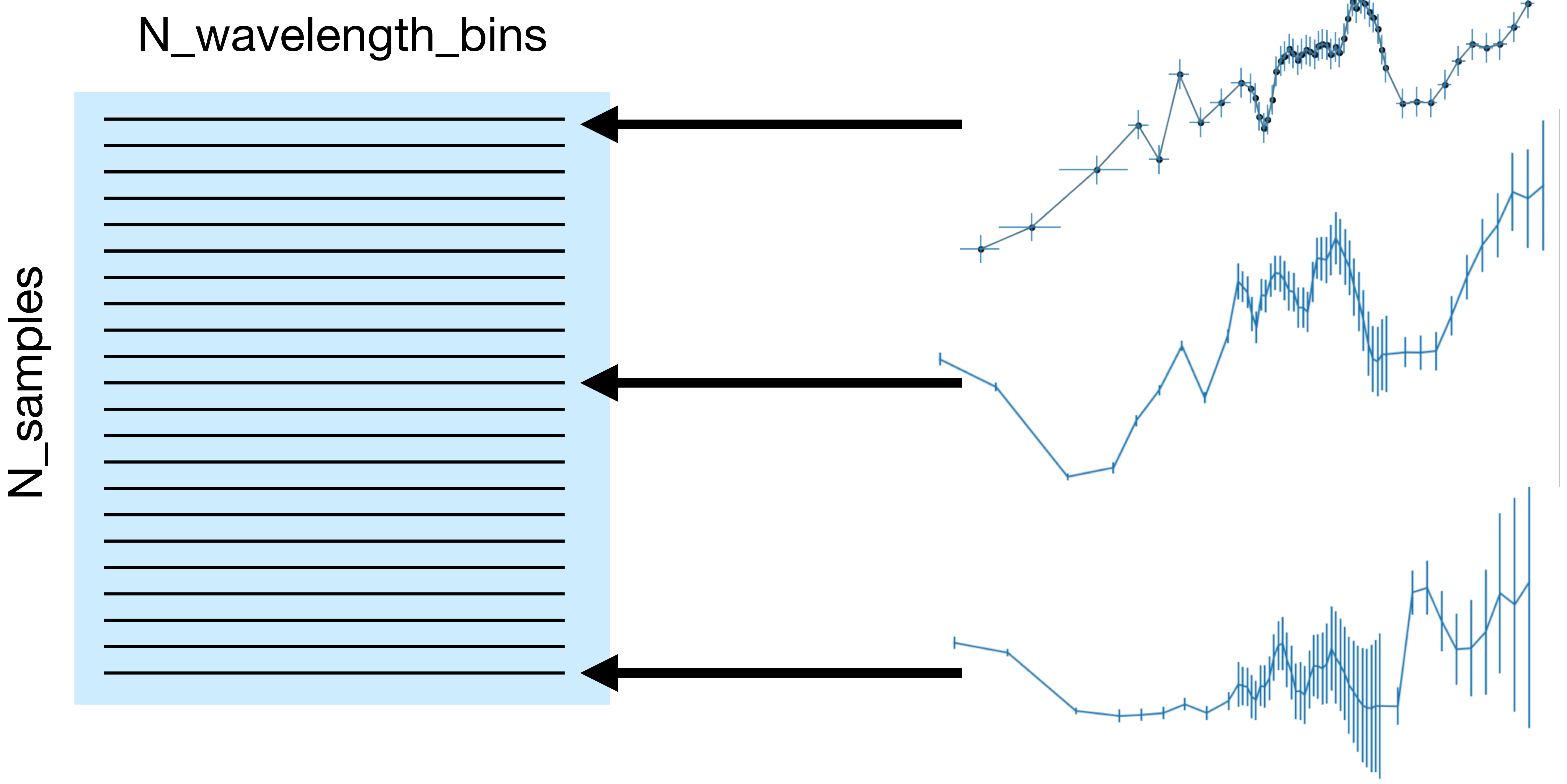


Introducing a generic data set

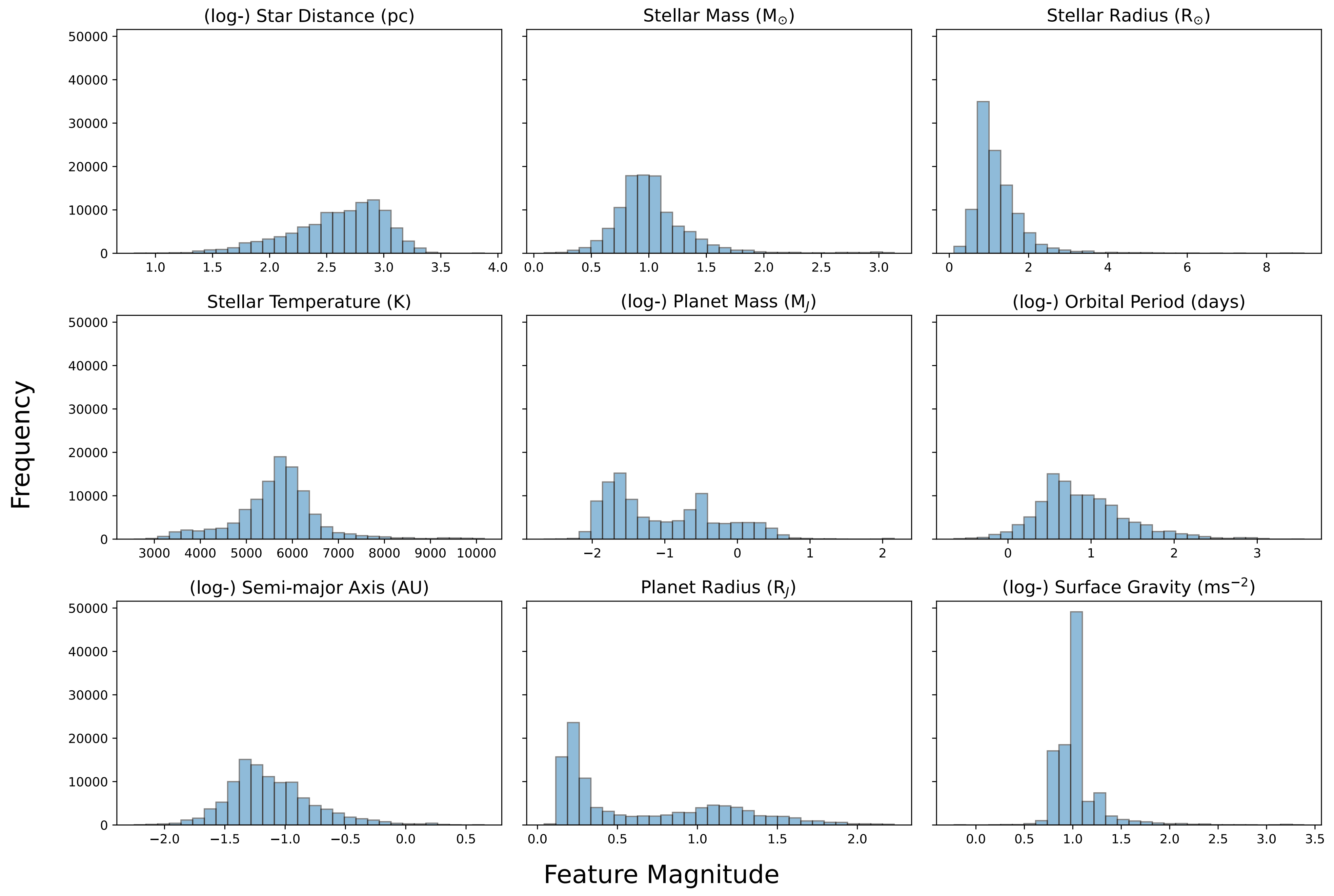
For most examples, our data set is an array of spectra

Array of Spectra

Spectra for various compositions and planetary/stellar parameters



Your data set should cover a wide range of parameters



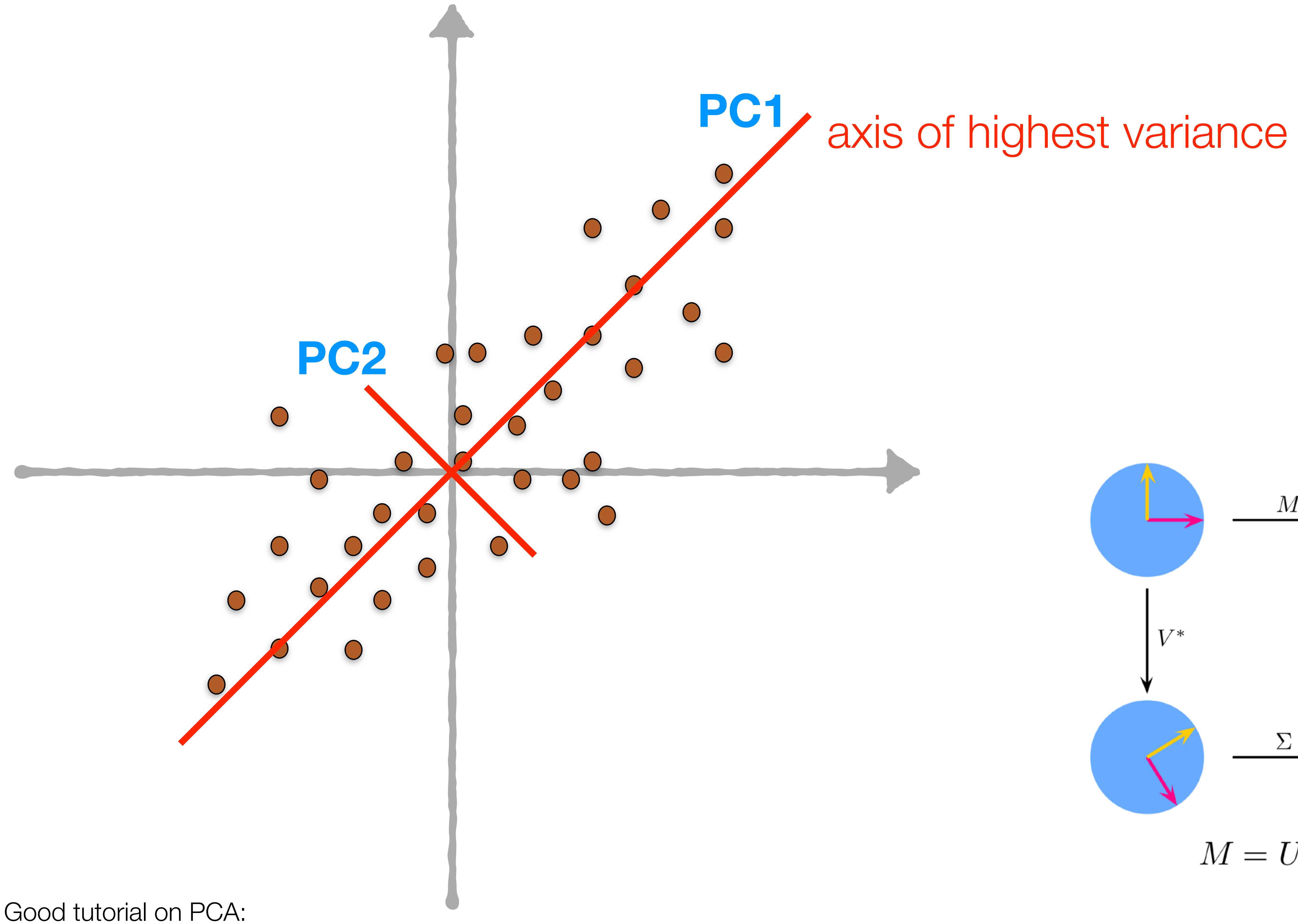
e.g. Changeat & Yip 2022

Clustering and PCA

/imagine prompt: separating components in a complex multi-dimensional space

Principal Component Analysis (PCA)

- PCA decomposition always exists
- Each component is orthogonal
- Is computational easy to compute (mostly)



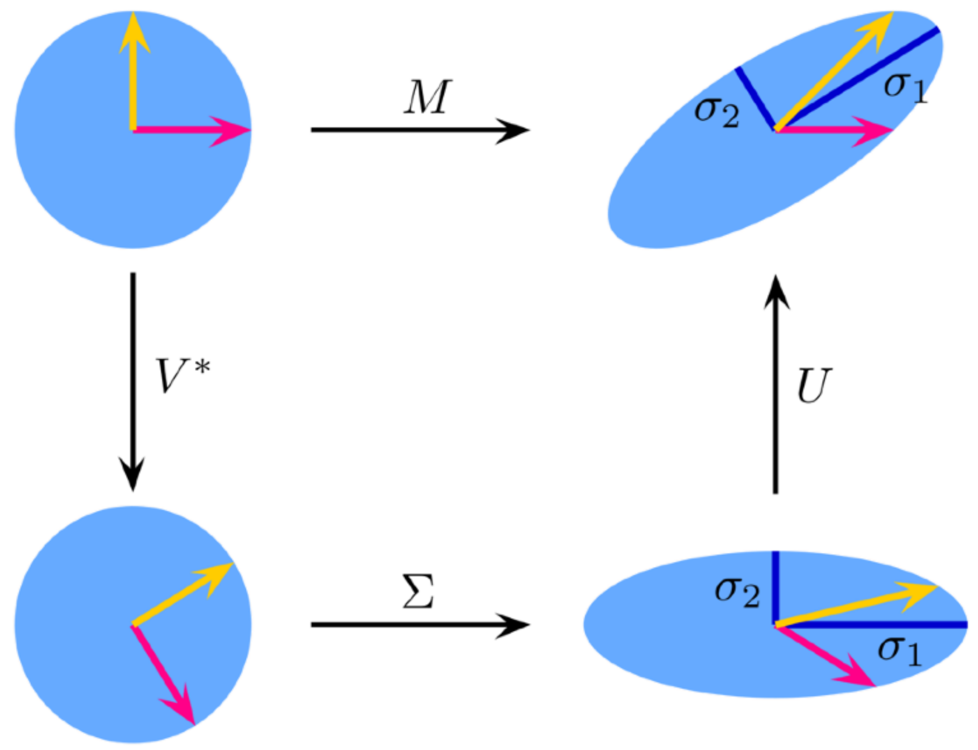
Single Value Decomposition

matrix of eigenvectors

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$$

data

diagonal matrix of eigenvalues



$$M = U \cdot \Sigma \cdot V^*$$

Projection to orthogonal axes

$$\begin{aligned} \mathbf{X} &= \mathbf{U}\Sigma\mathbf{V}^T \\ \mathbf{U}^T\mathbf{X} &= \Sigma\mathbf{V}^T \\ \mathbf{U}^T\mathbf{X} &= \mathbf{Z} \end{aligned}$$

z-score

Good tutorial on PCA:
http://arxiv.org/pdf/1404.1100v1.pdf

The Planetary Science Journal, 3:205 (12pp), 2022 September https://doi.org/10.3847/PSJ/ac880b

Unsupervised Machine Learning for Exploratory Data Analysis of Exoplanet Transmission Spectra

Konstantin T. Matchev, Katia Matcheva, and Alexander Roman
Physics Department, University of Florida, Gainesville, FL 32653, USA; matcheva@ufl.edu
Received 2022 April 7; revised 2022 July 1; accepted 2022 August 1; published 2022 September 1

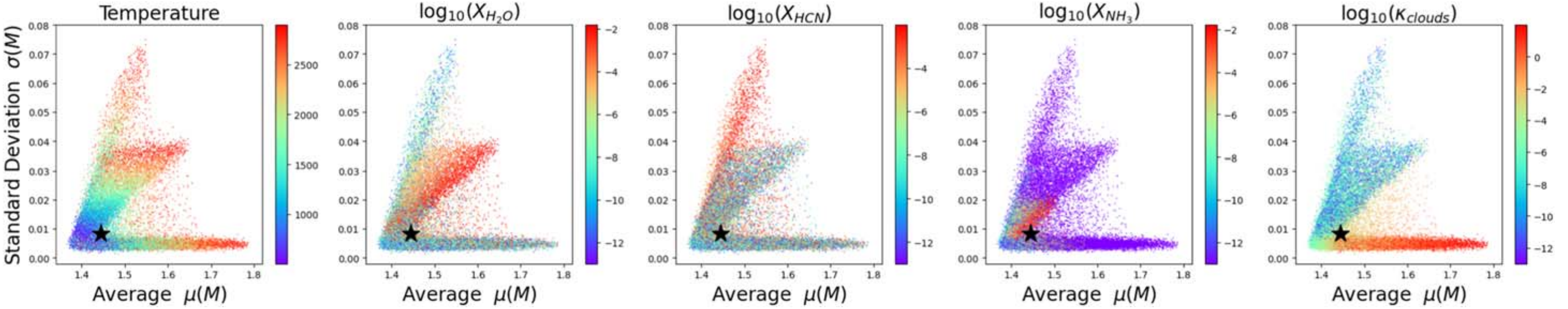


Figure 4. Scatter plots of 25,000 data points: the average $\mu(M)$ (plotted on the x-axis) vs. the standard deviation $\sigma(M)$ (plotted on the y-axis). In each panel, the points are color-coded by the value of one of the five target variables, indicated at the top. The black $\star$ symbol marks the location of the hot gas giant exoplanet WASP-12b.

Sklearn example of PCA

```
#importing PCA routine
from sklearn.decomposition import PCA
```

```
N = 1000 #data index to take the first 1000 spectra only

#running PCA
pca = PCA(n_components=3)
pca.fit(train_data[:N,0:13]) #performing the PCA transform
PCA_out = pca.fit_transform(train_data[:N,0:13]) #transforming your data into PCA space
```

Google Colab notebook:
https://bit.ly/ExoAI_PCA

https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

Spectral clustering of exoplanets

- Turns out that most of your information in your spectral data can be described by only 2 - 3 principal components

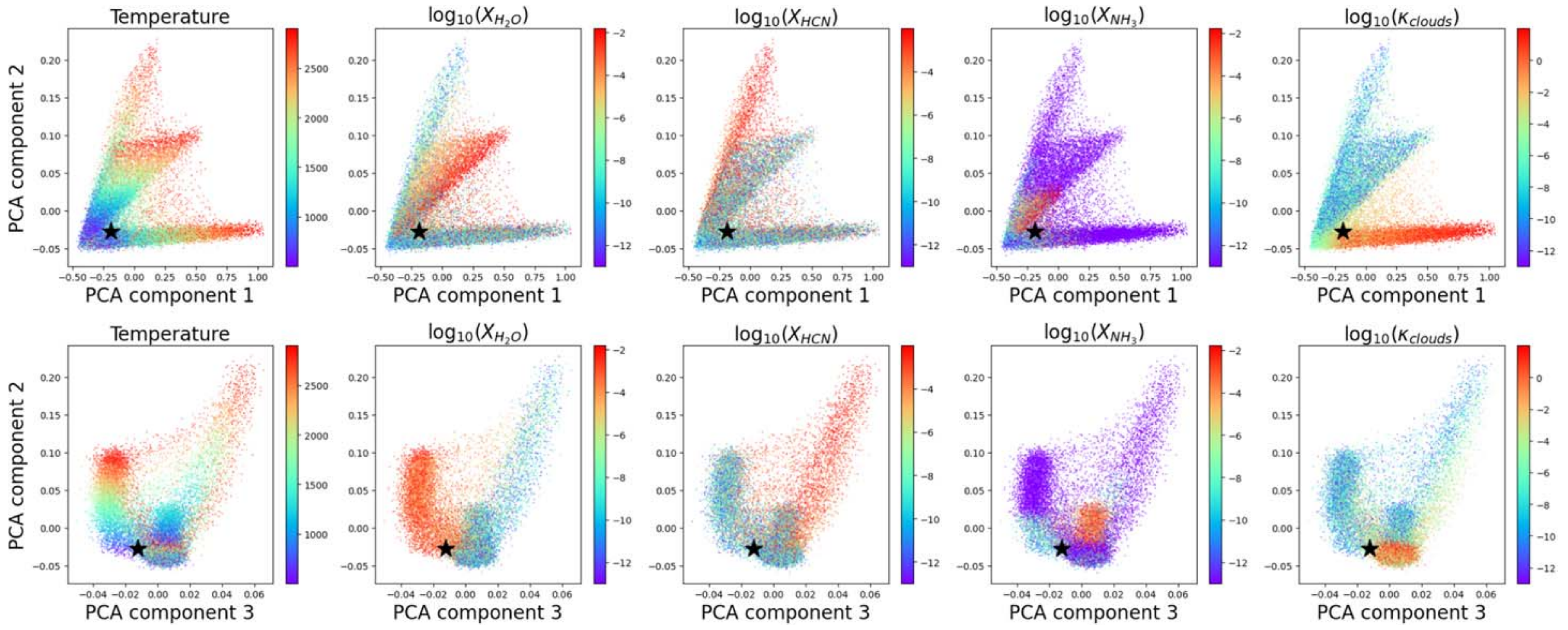
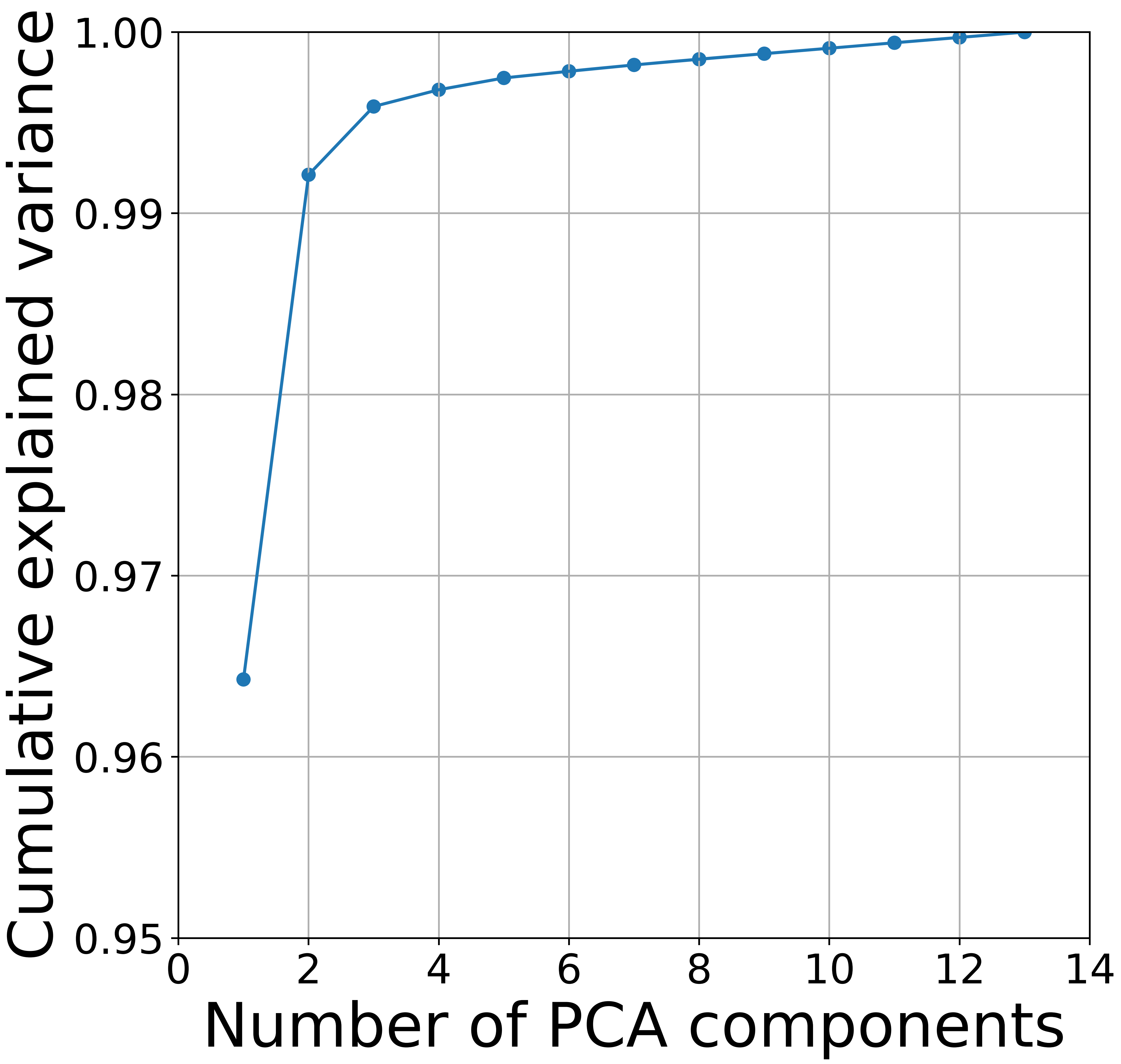


Figure 7. The same as Figure 4, but plotted in the plane of the first and second PCA components (top row) or the plane of the second and third PCA components (bottom row).



Matchev et al. 2022

Spectral clustering of exoplanets

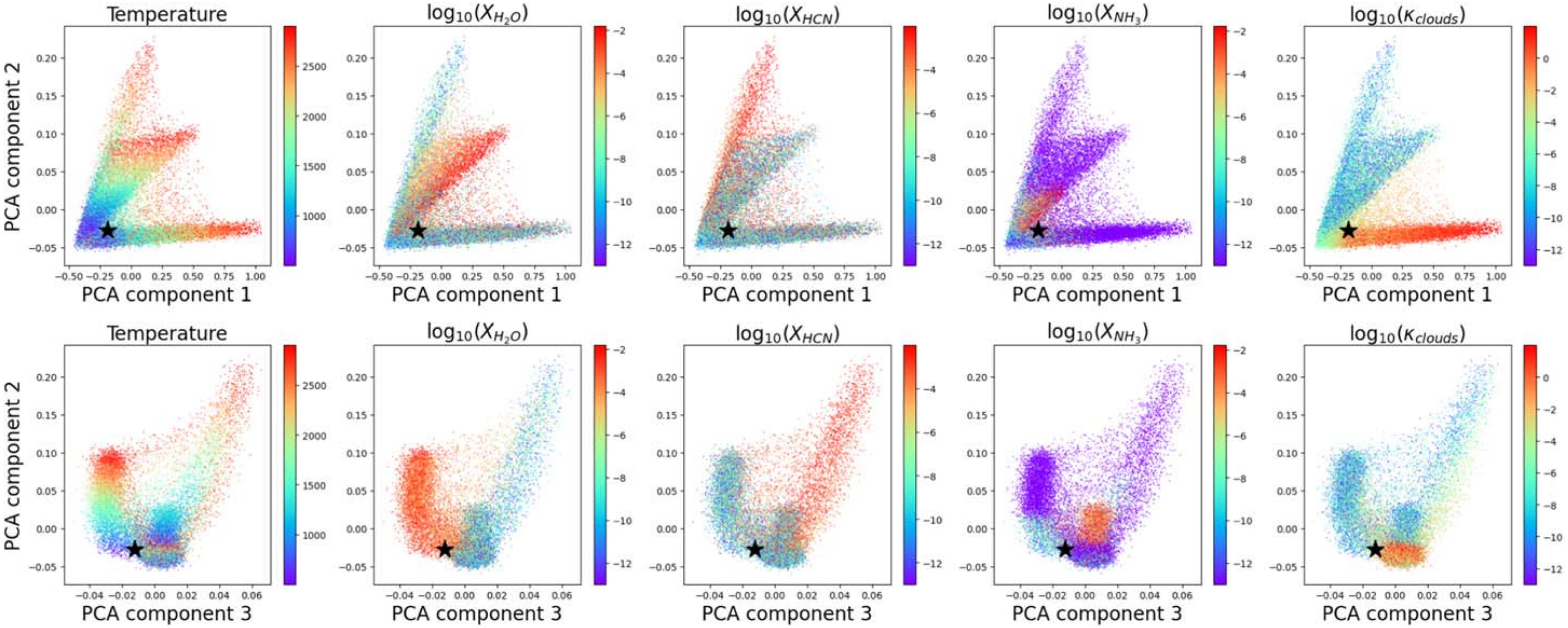


Figure 7. The same as Figure 4, but plotted in the plane of the first and second PCA components (top row) or the plane of the second and third PCA components (bottom row).

Matchev et al. 2022

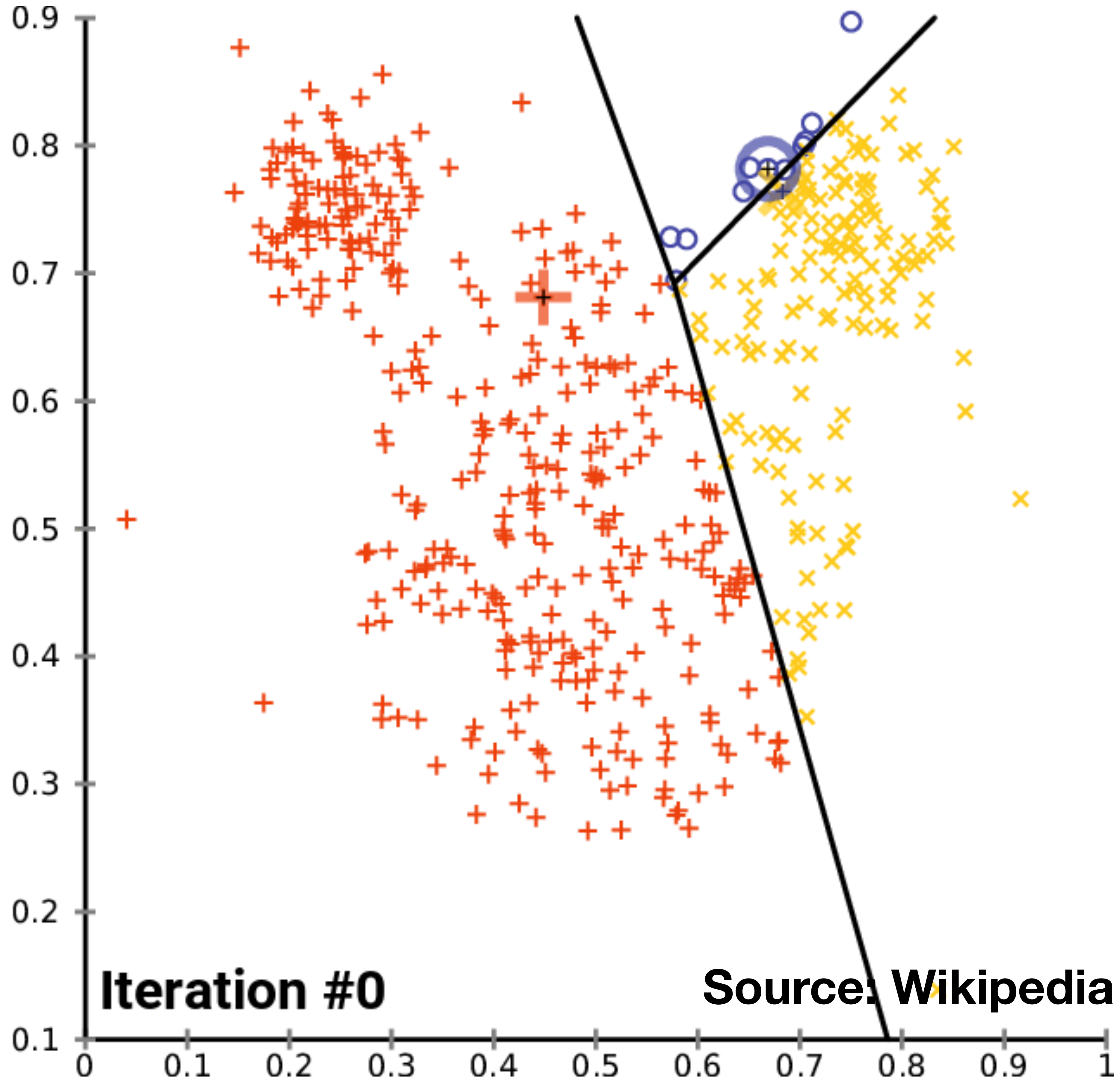
Clustering (k-means)

- The PCA components show distinct features… can we cluster them? - Yes

- Many clustering algorithms exist. All the good ones are on sklearn and you can try them all easily

K-means

- Given a number of given clusters, it calculates the position of the cluster mean (i.e. centre) that minimises both the distance of the surrounding points and the variance around the mean



Google Colab notebook:
https://bit.ly/ExoAI_PCA

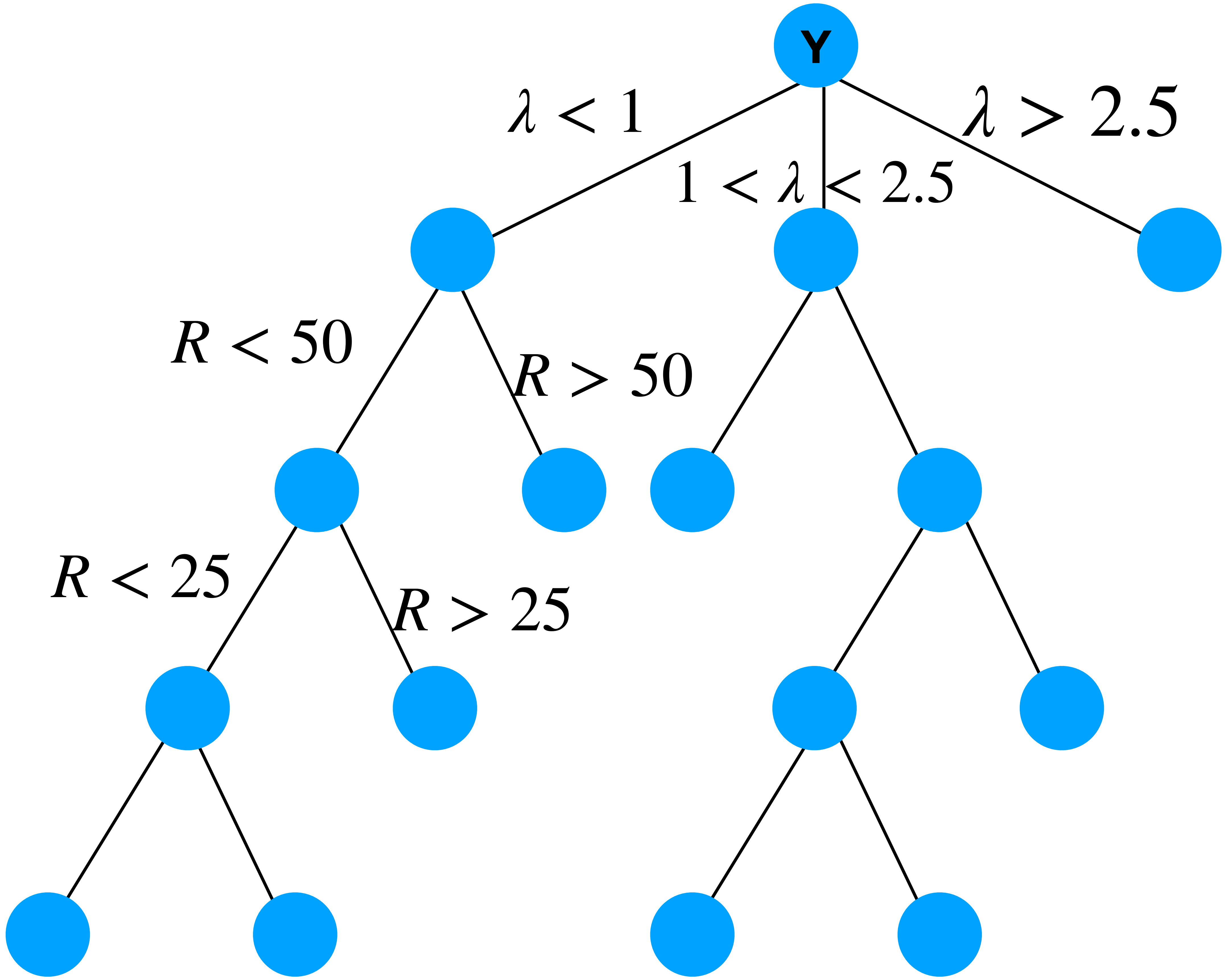
https://scikit-learn.org/stable/modules/clustering.html

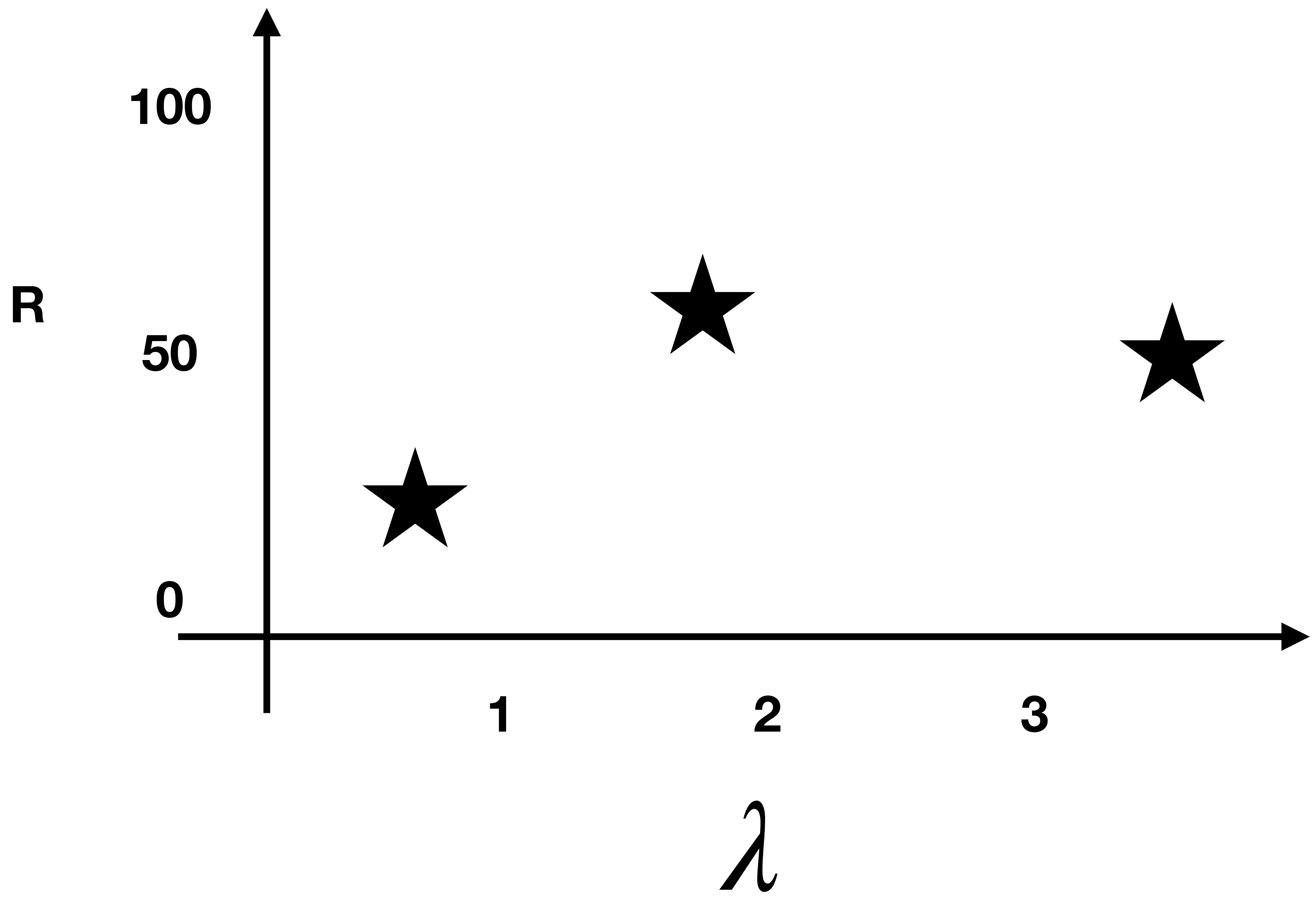
Random Forest regression

/imagine prompt: A random forest

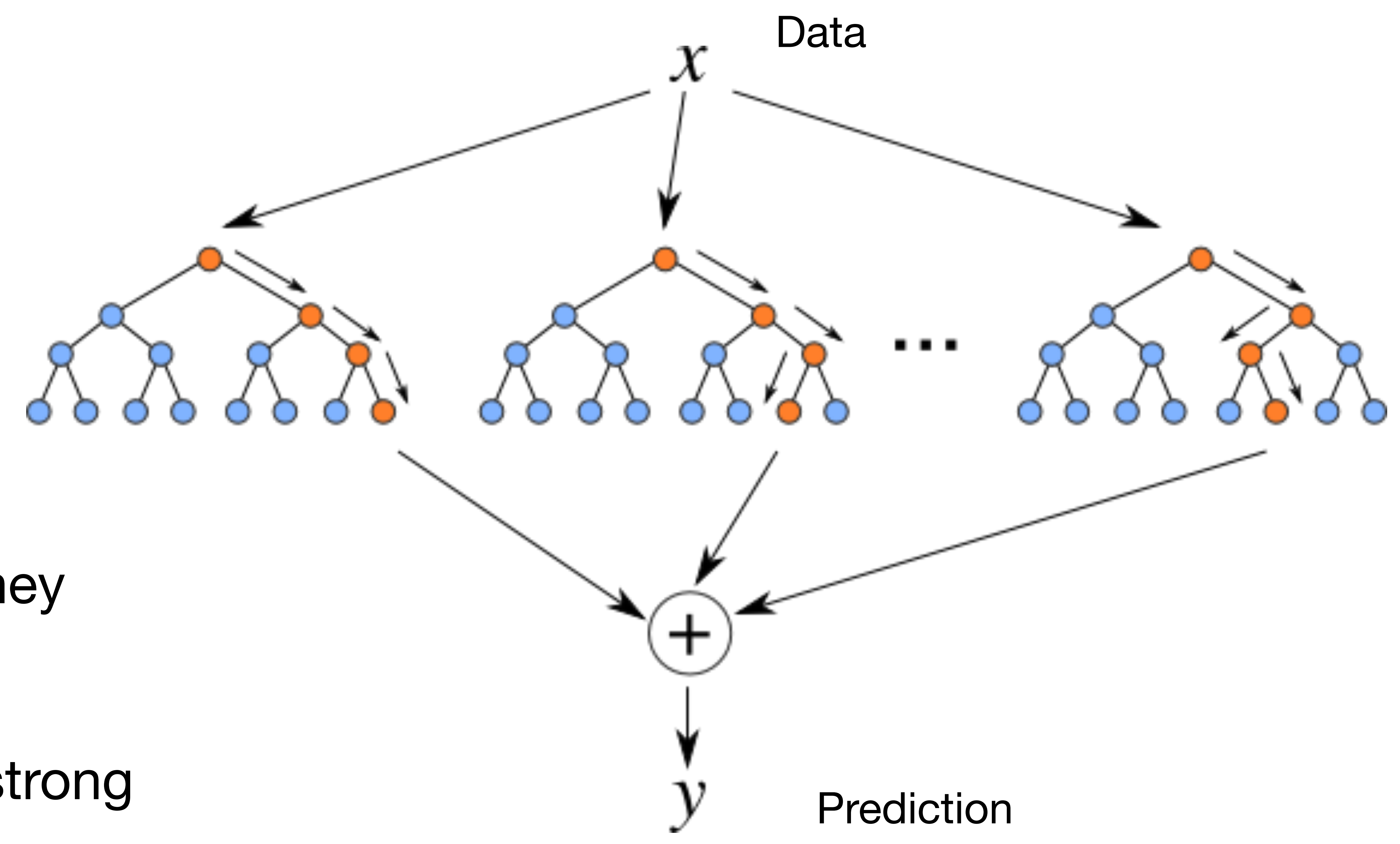
Decision tree regression

- Decision tress are a supervised machine learning technique
- They find a mapping between data (X) and labels/results (Y)





Random Forest regression

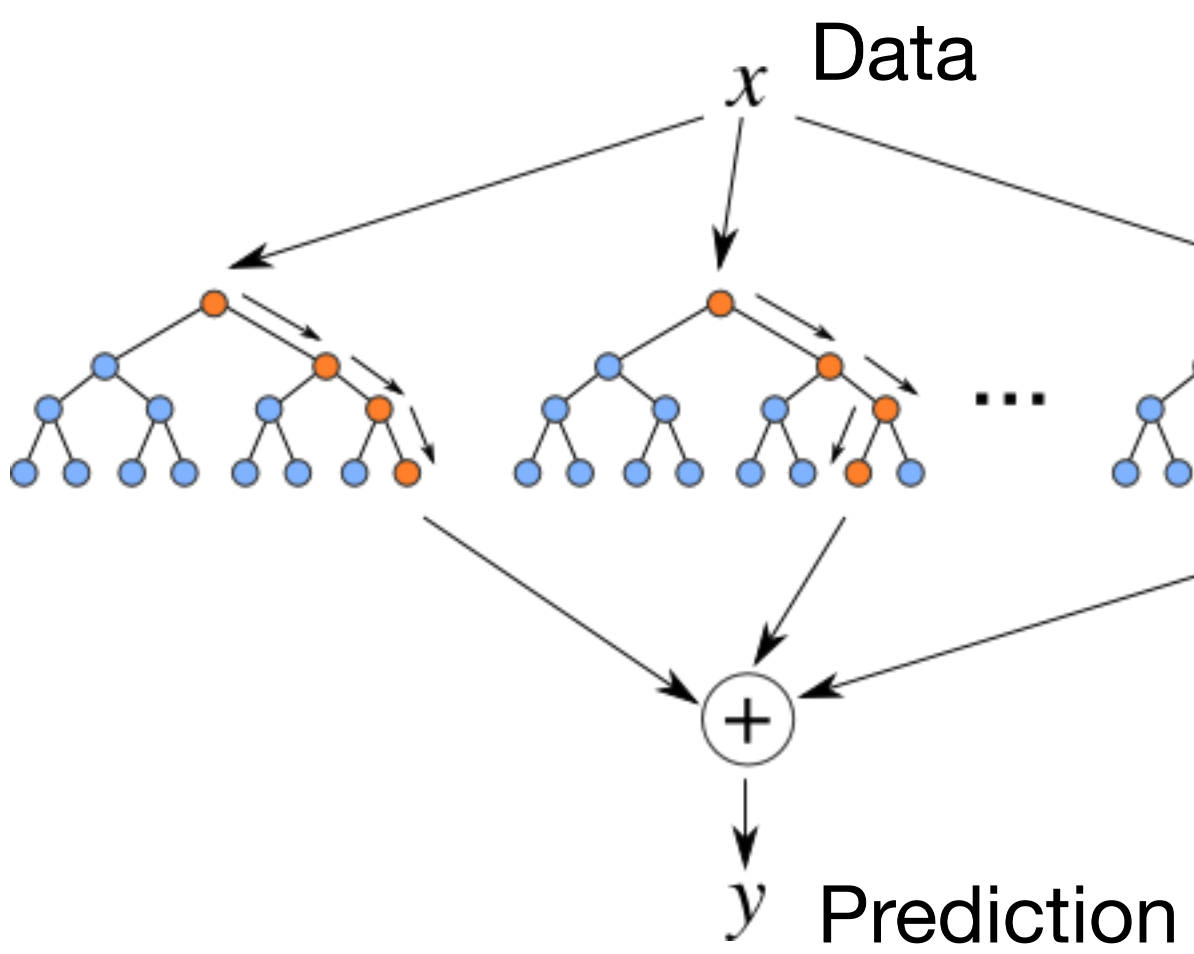


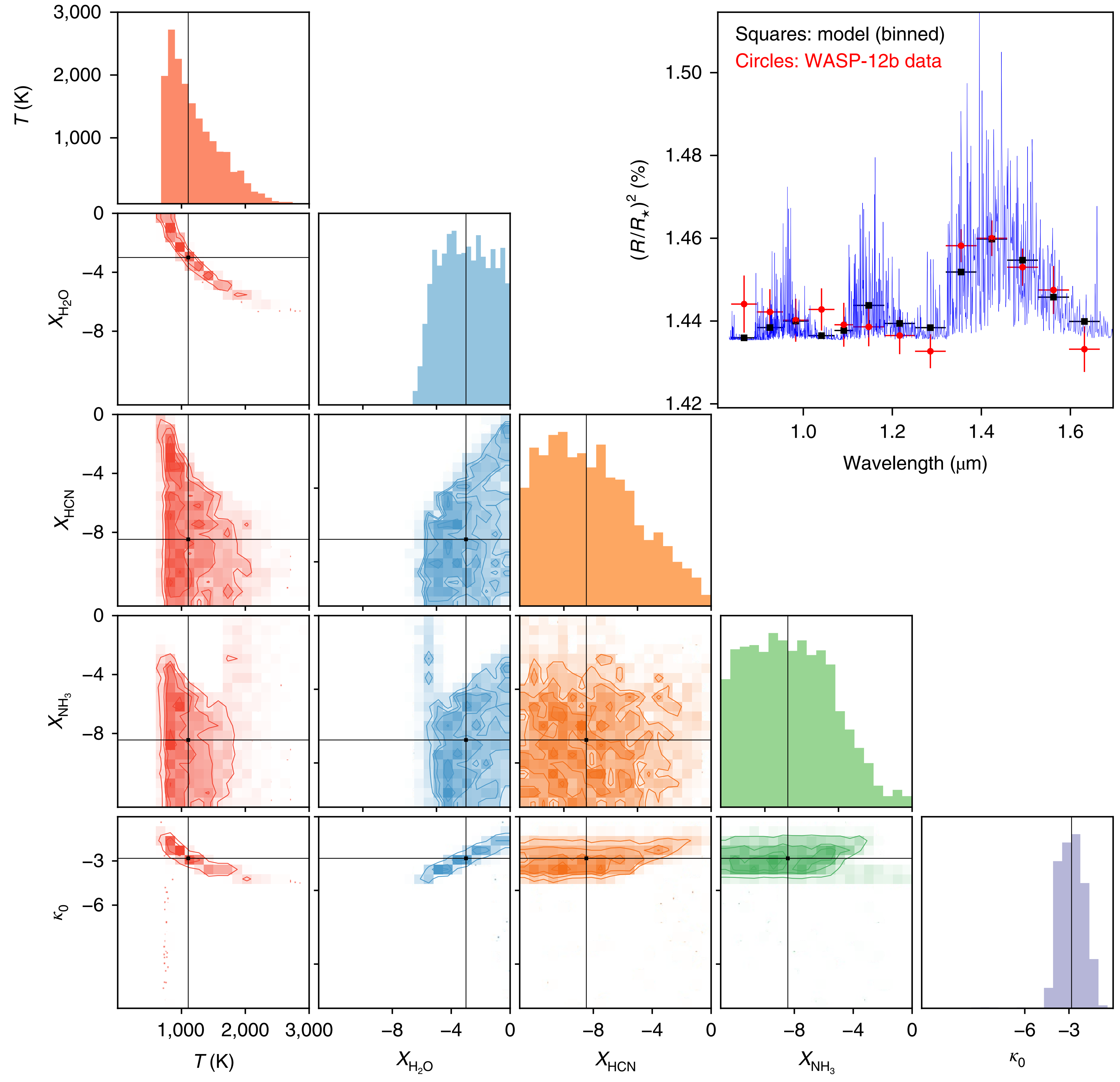
- Individual Trees are not very good predictors (they are called ‘weak predictors’)
- By averaging many weak predictors you get a strong predictor
- Averaging/summing many trees is called ‘ensembling’

Marquez-Neila et al. 2018
see also e.g. Nixon & Madhusudan 2019

Using Random Forrests to classify a hot Jupiter

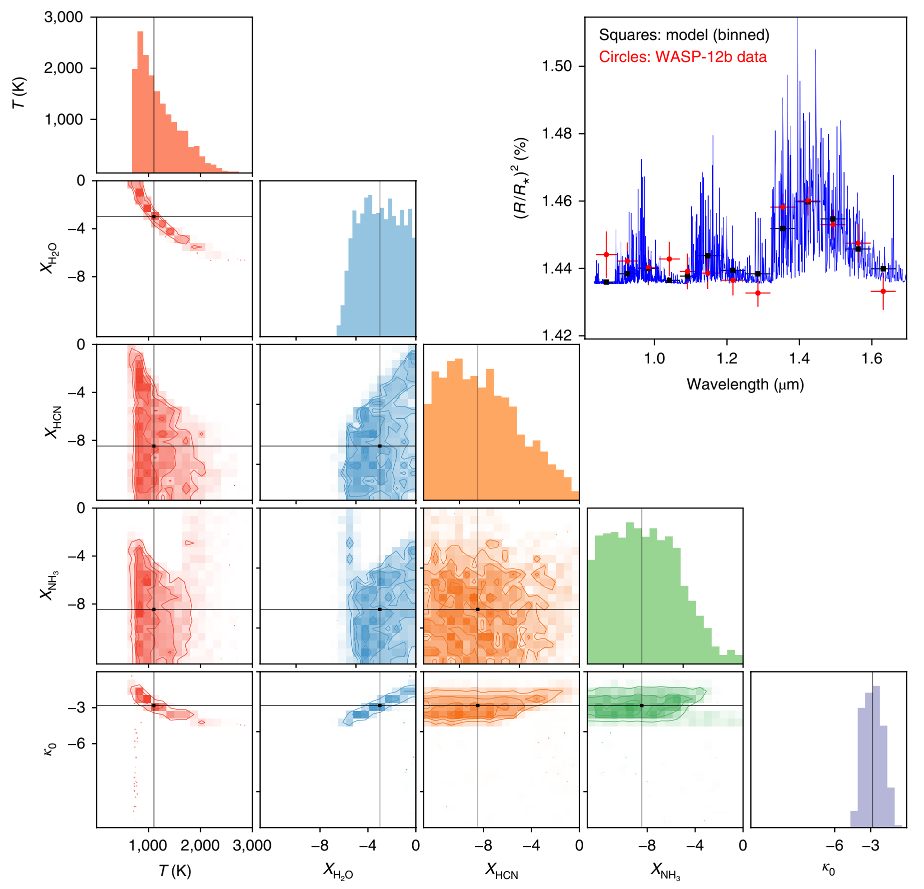
- Radom Forests are an ensemble of multiple decision trees
- One of the oldest and most stable machine learning methods
- Individual Forrests are by nature interpretable (ensembles not)
- Easy and fast to train
- Do not generalise as well as deep learning and struggle to cope with large data sets



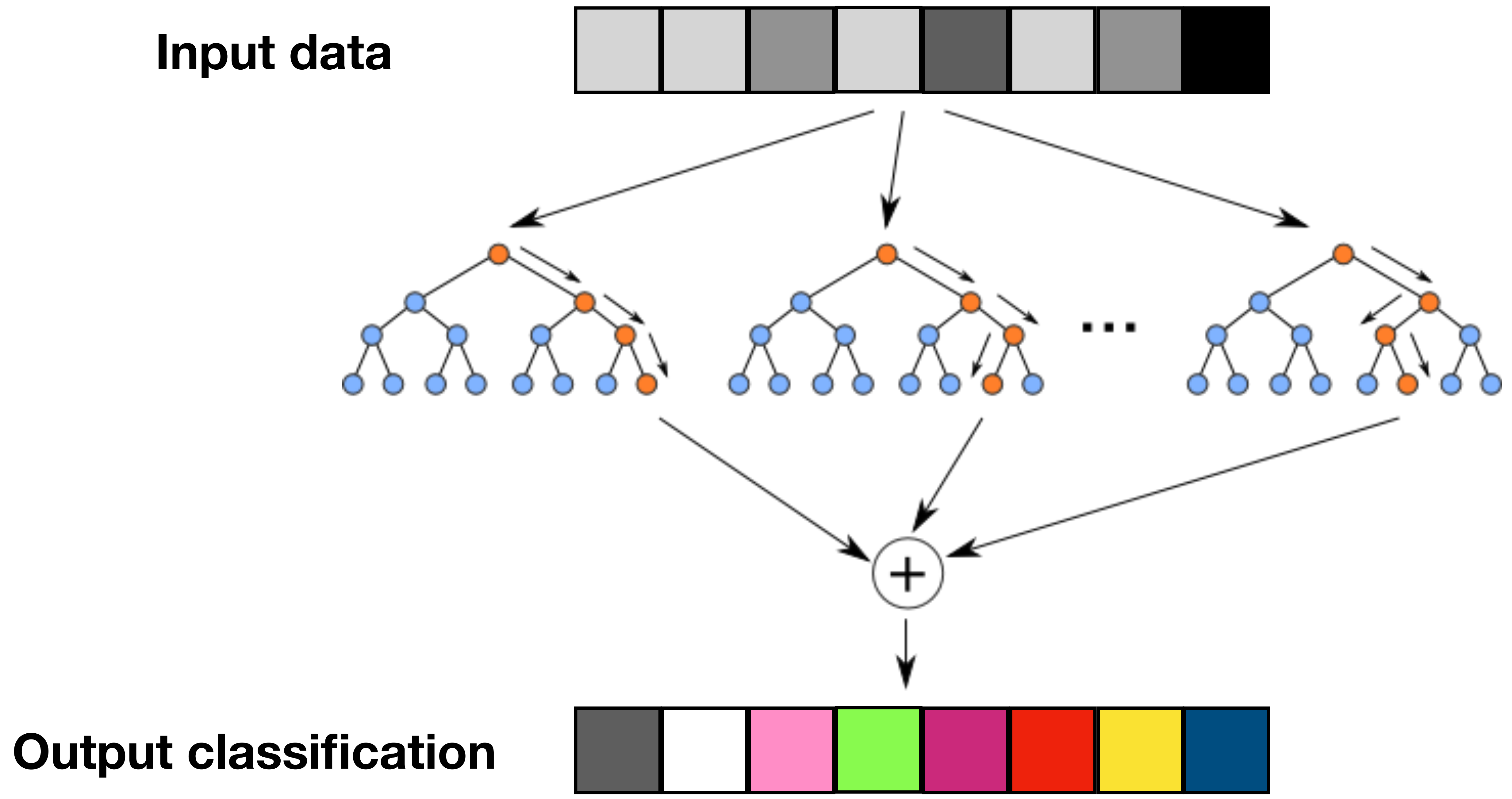


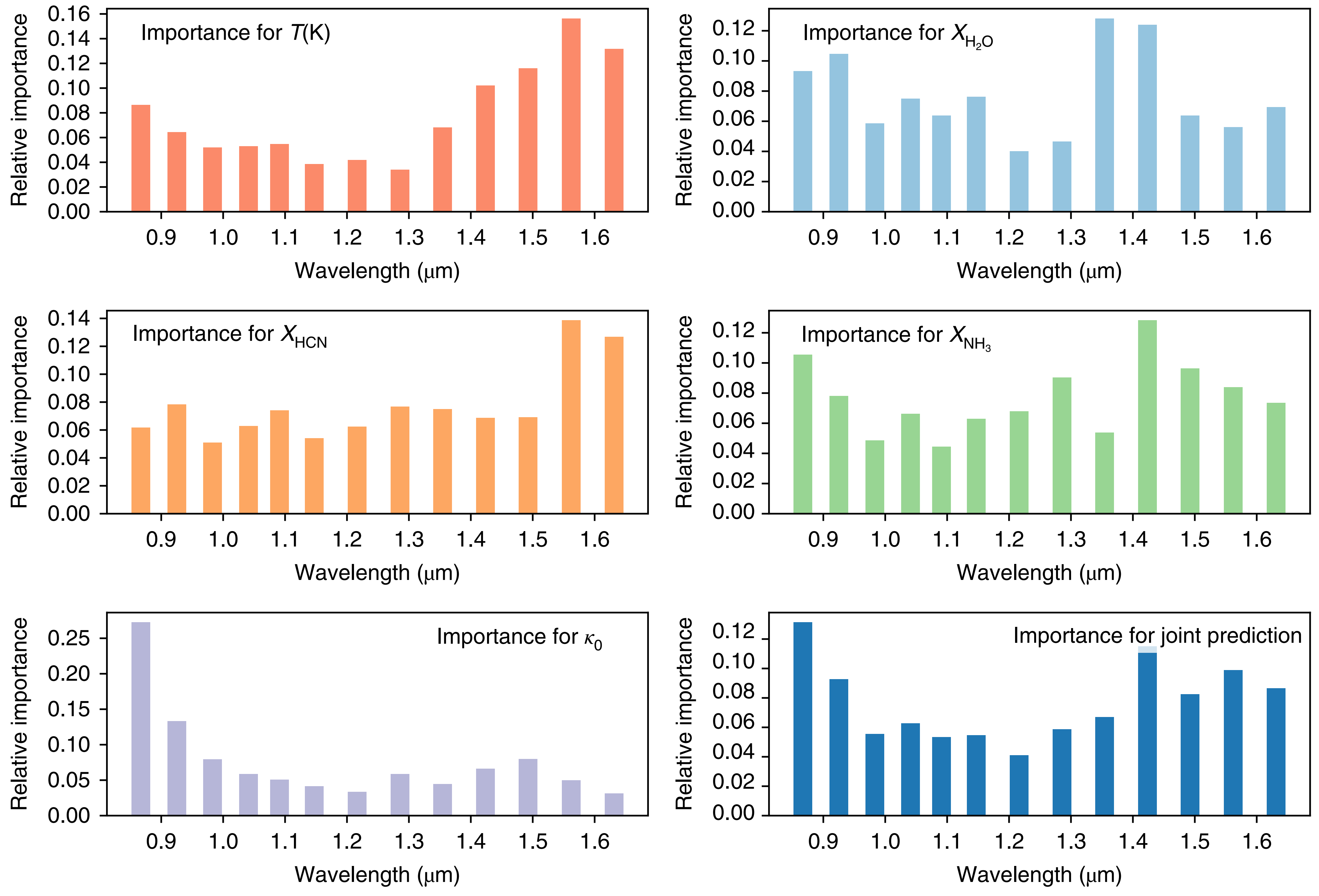
Marquez-Neila et al. 2018
see also e.g. Nixon & Madhusudan 2019

Feature importance in Random Forrests



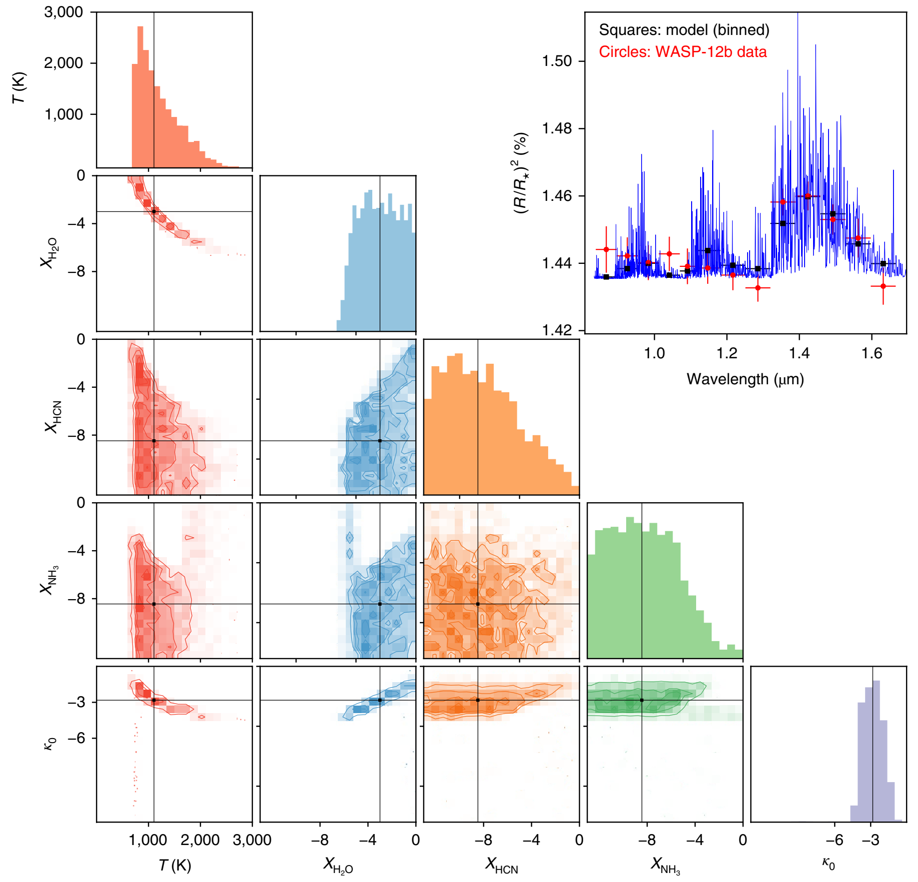
- Perturbation based analysis gives understanding of what data has greatest impact



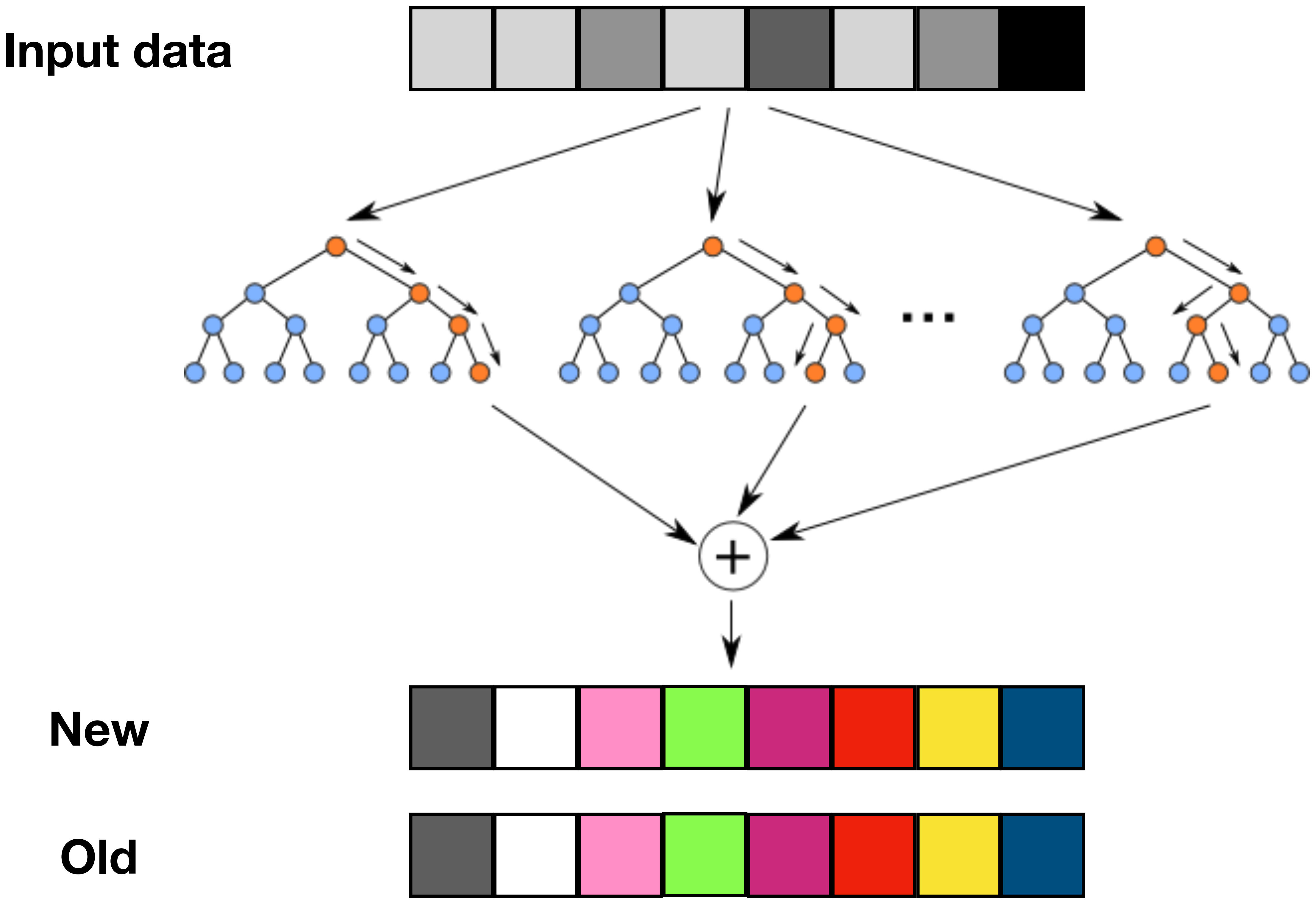


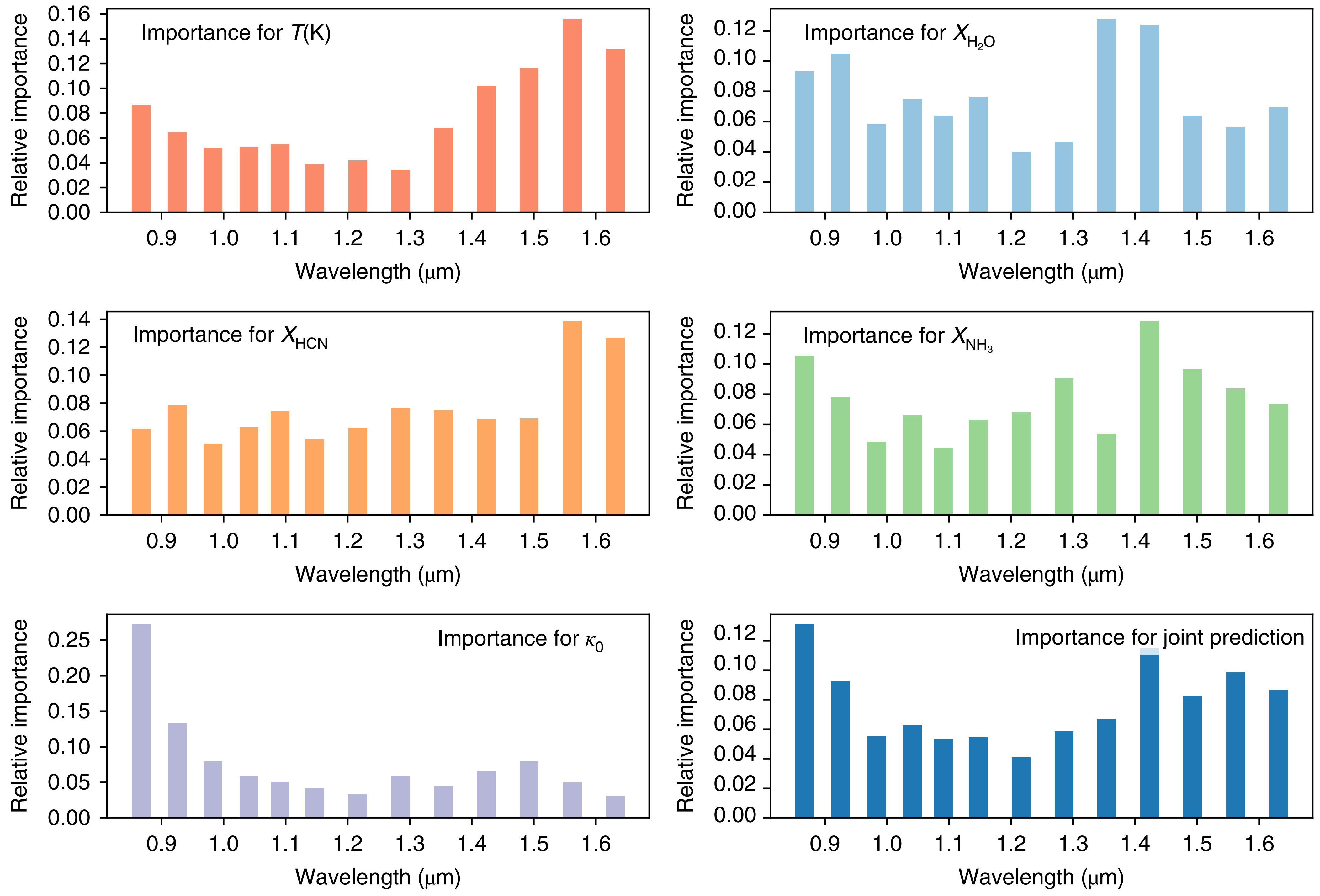
Marquez-Neila et al. 2018
see also e.g. Nixon & Madhusudan 2019

Feature importance in Random Forrests



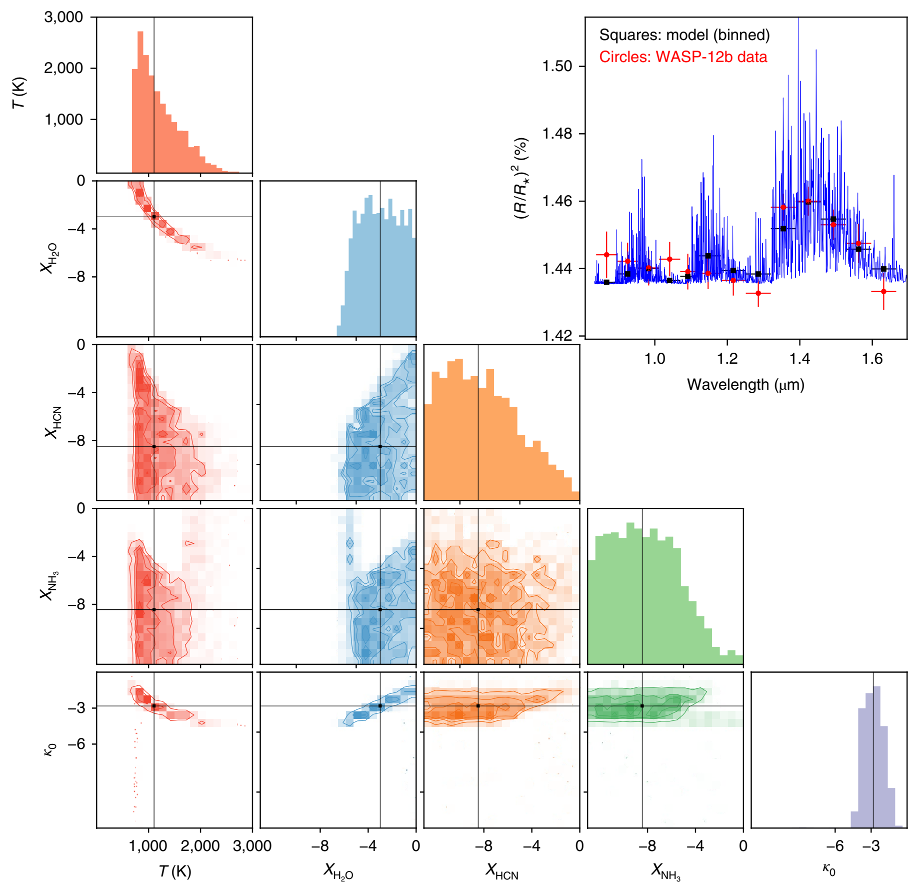
- Perturbation based analysis gives understanding of what data has greatest impact



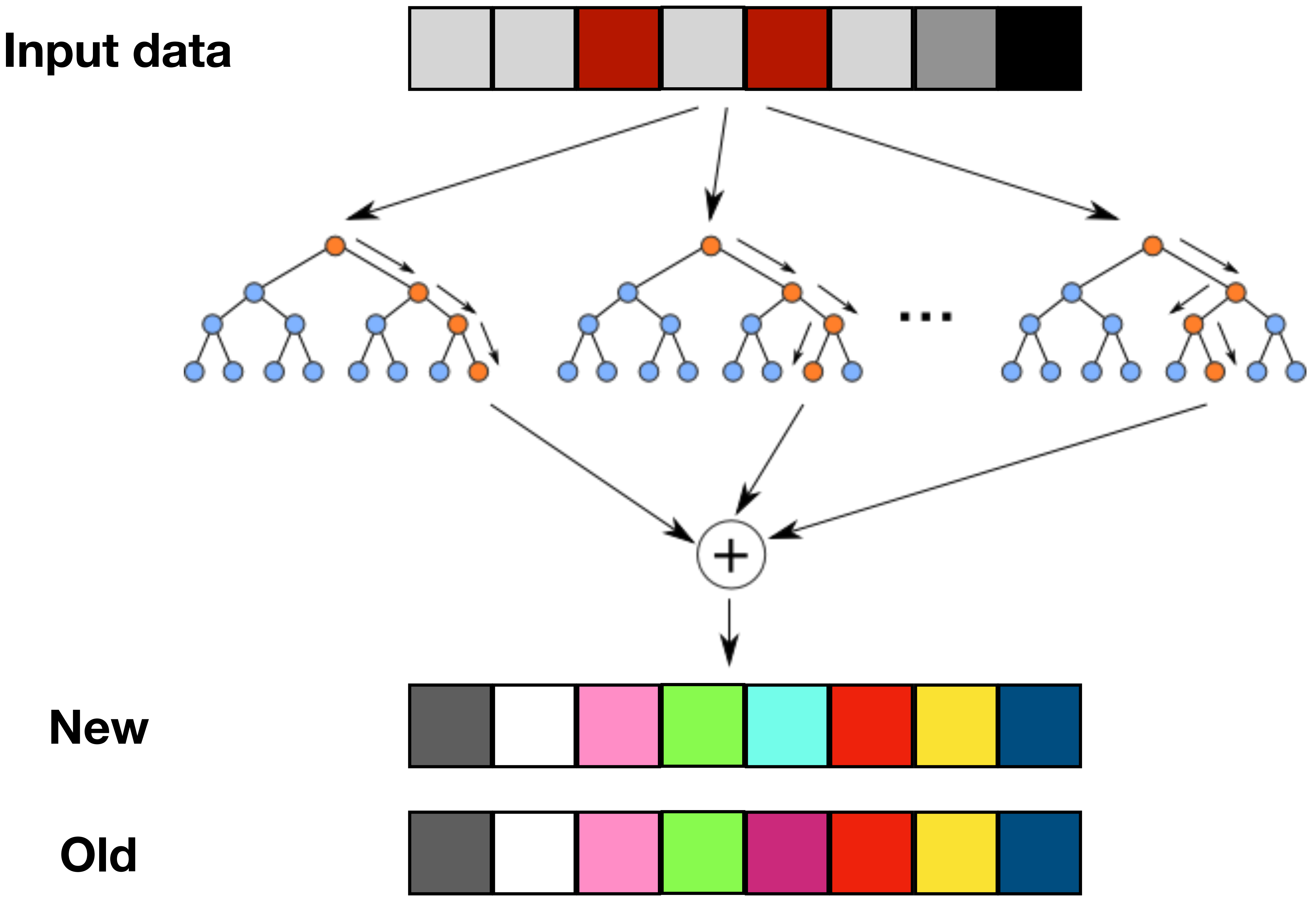


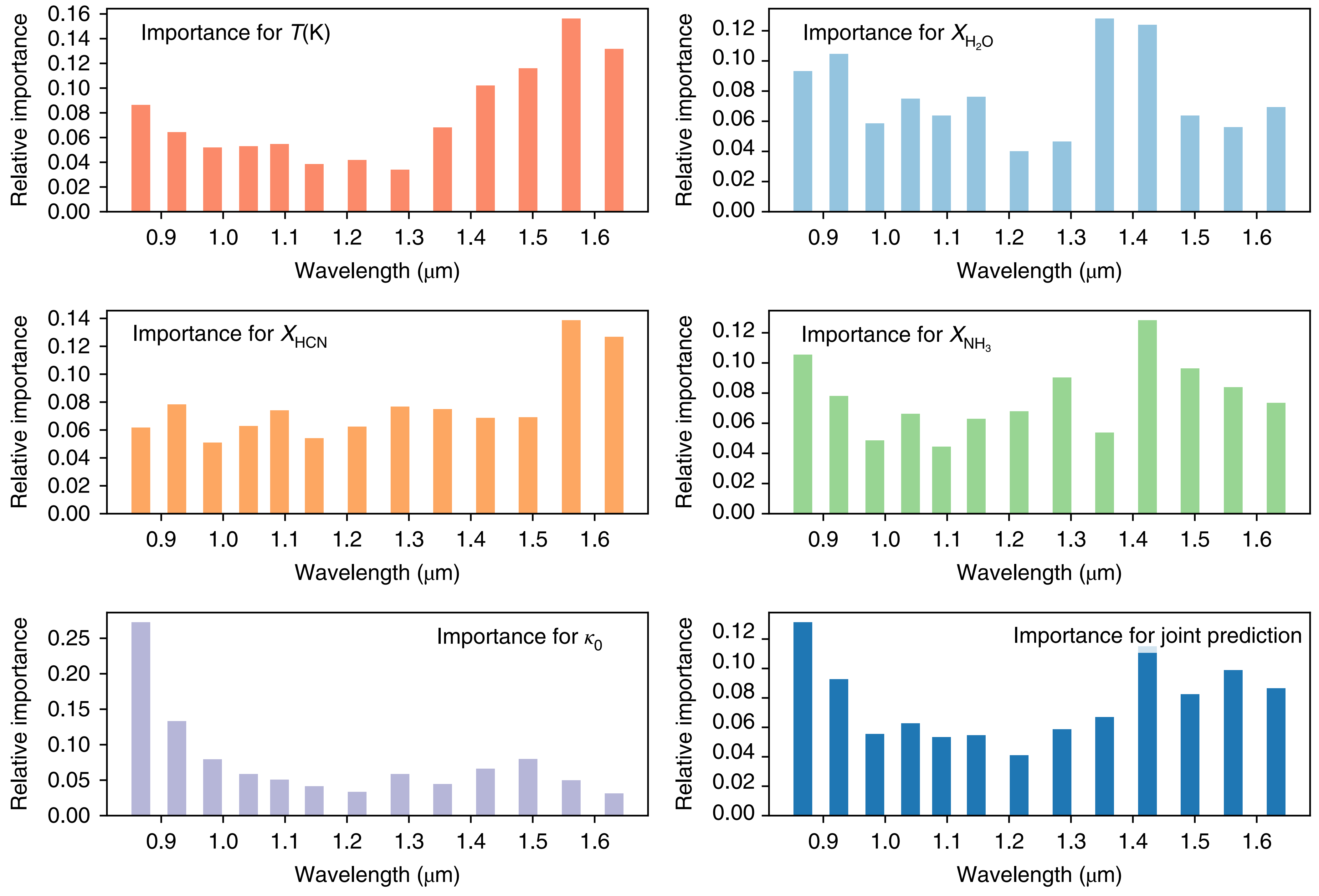
Marquez-Neila et al. 2018
see also e.g. Nixon & Madhusudan 2019

Feature importance in Random Forrests



- Perturbation based analysis gives understanding of what data has greatest impact

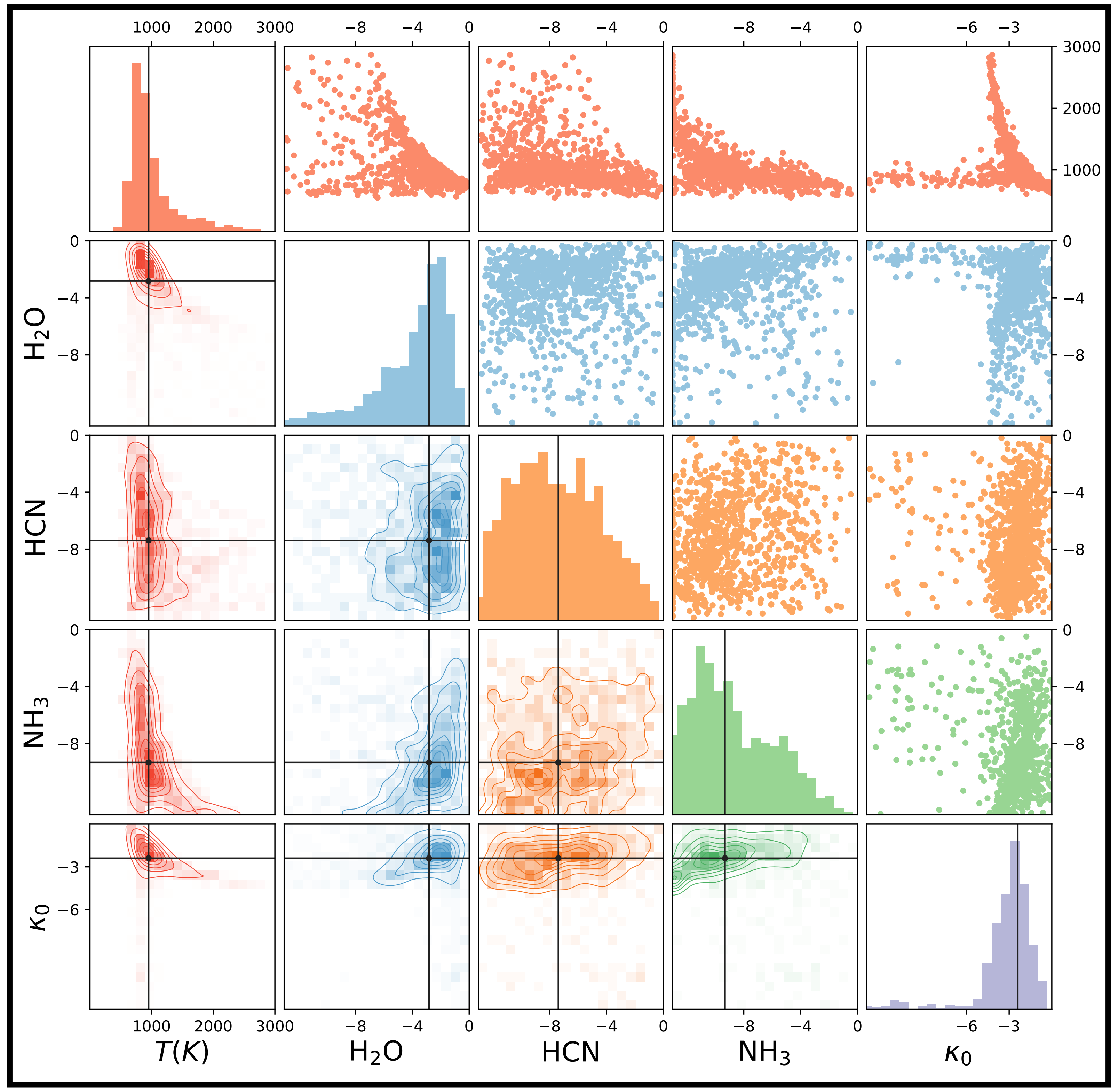


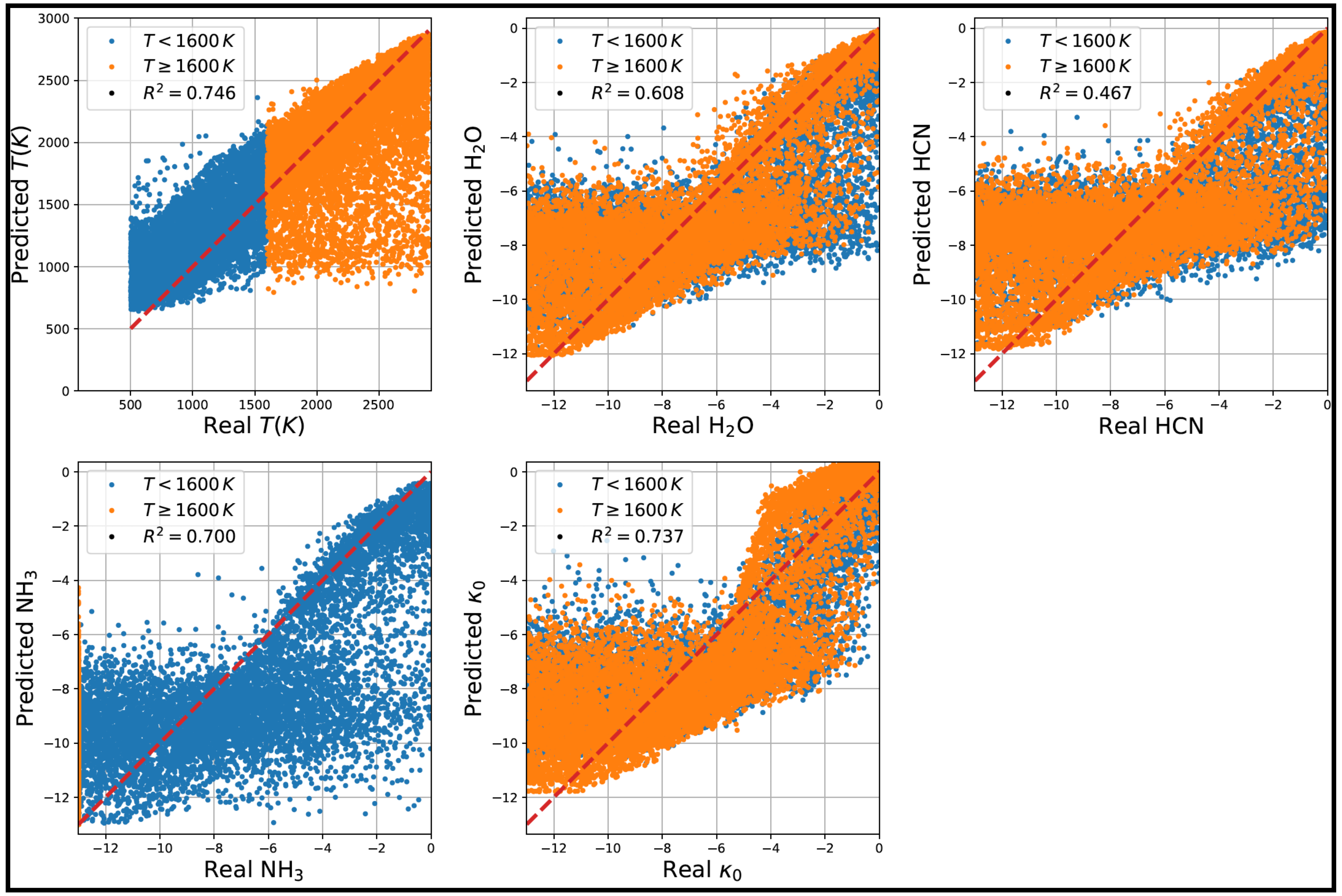


Marquez-Neila et al. 2018
see also e.g. Nixon & Madhusudan 2019

Using Random Forrests

By reading out the individual outputs, you can generate Parameter distributions
Note these are NOT formal Bayesian posterior distributions





Marquez-Neila et al. 2018
see also e.g. Nixon & Madhusudan 2019

Using Random Forrests

Pro:

- Easy to implement and fast to run
- Is in principle fully interpretable, also known as a ‘white box model’
- Can easily derive Feature Importance diagnostics
- Can provide a probability over parameters, can be extended into Bayesian framework

Con:

- Does not scale well with data size
- May not be expressive enough with a realistic number of trees
- Interpretability becomes difficult for many and deep trees

Using Random Forrests

```
>>> from sklearn.ensemble import RandomForestRegressor
>>> from sklearn.datasets import make_regression
>>> X, y = make_regression(n_features=4, n_informative=2,
...                        random_state=0, shuffle=False)
>>> regr = RandomForestRegressor(max_depth=2, random_state=0)
>>> regr.fit(X, y)
RandomForestRegressor(...)
>>> print(regr.predict([[0, 0, 0, 0]]))
[-8.32987858]
```

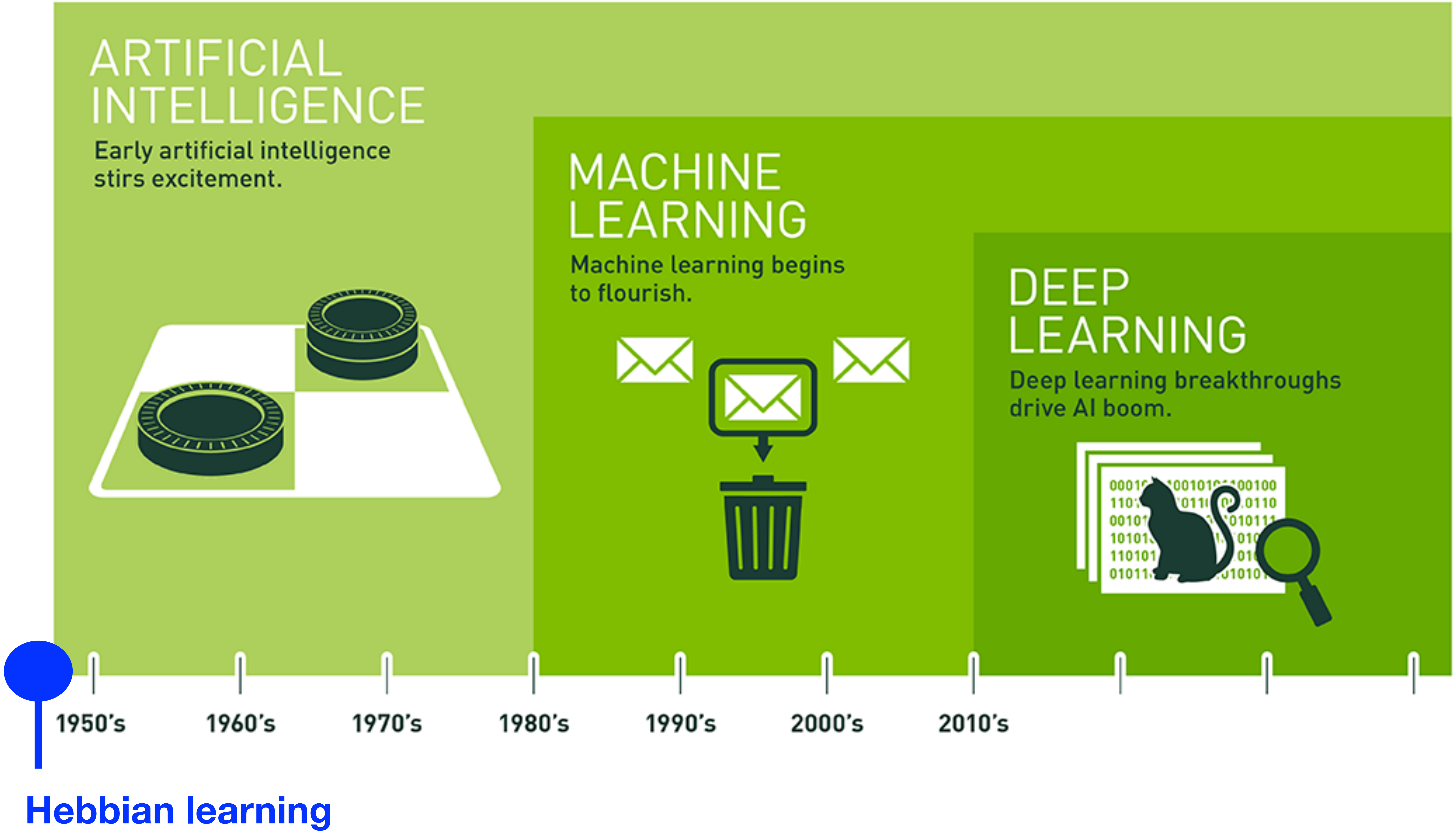
Google Colab notebook:
https://bit.ly/ExoAI_RF

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#

Interlude:
A very brief history of the origins of deep learning

/imagine prompt: theatre curtains for an interlude

Machine Learning and Deep Learning

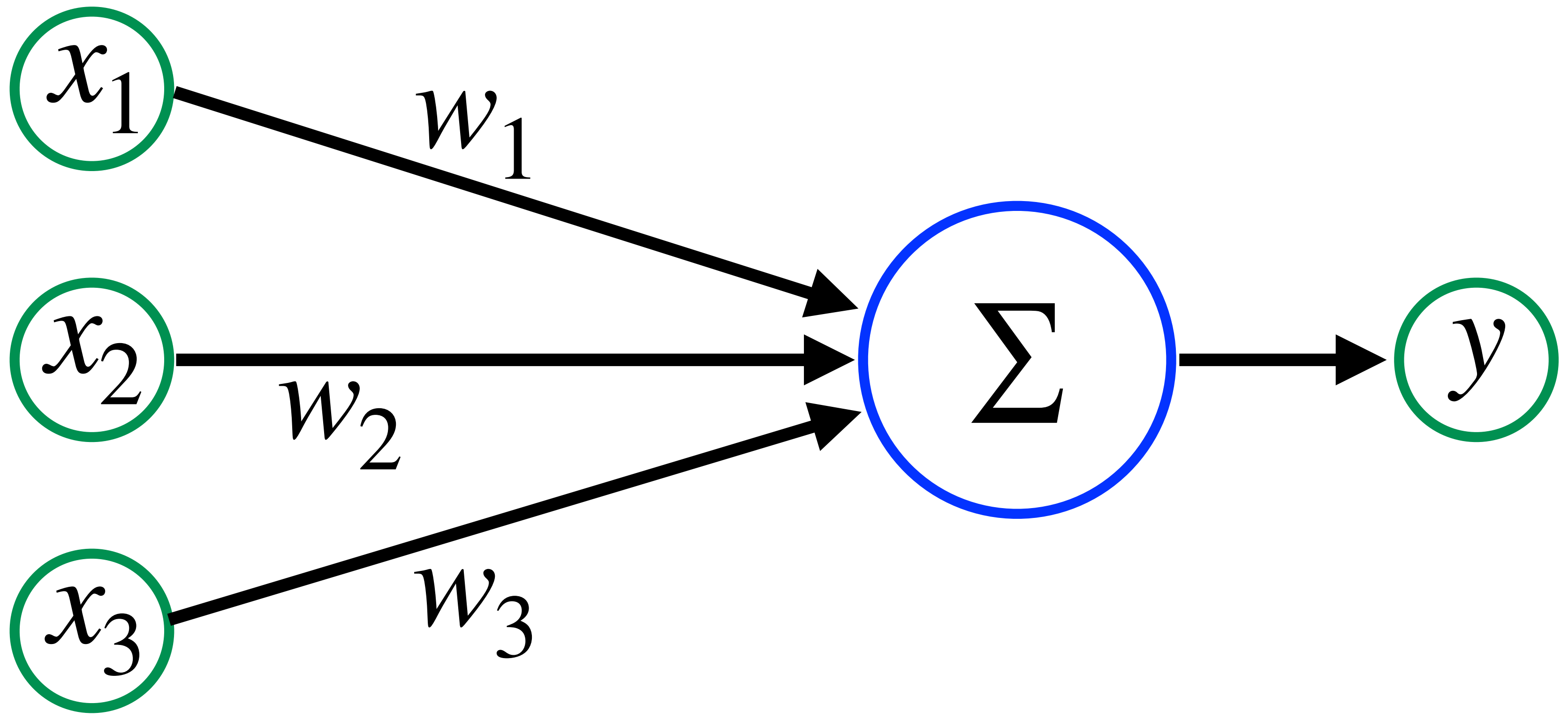


https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

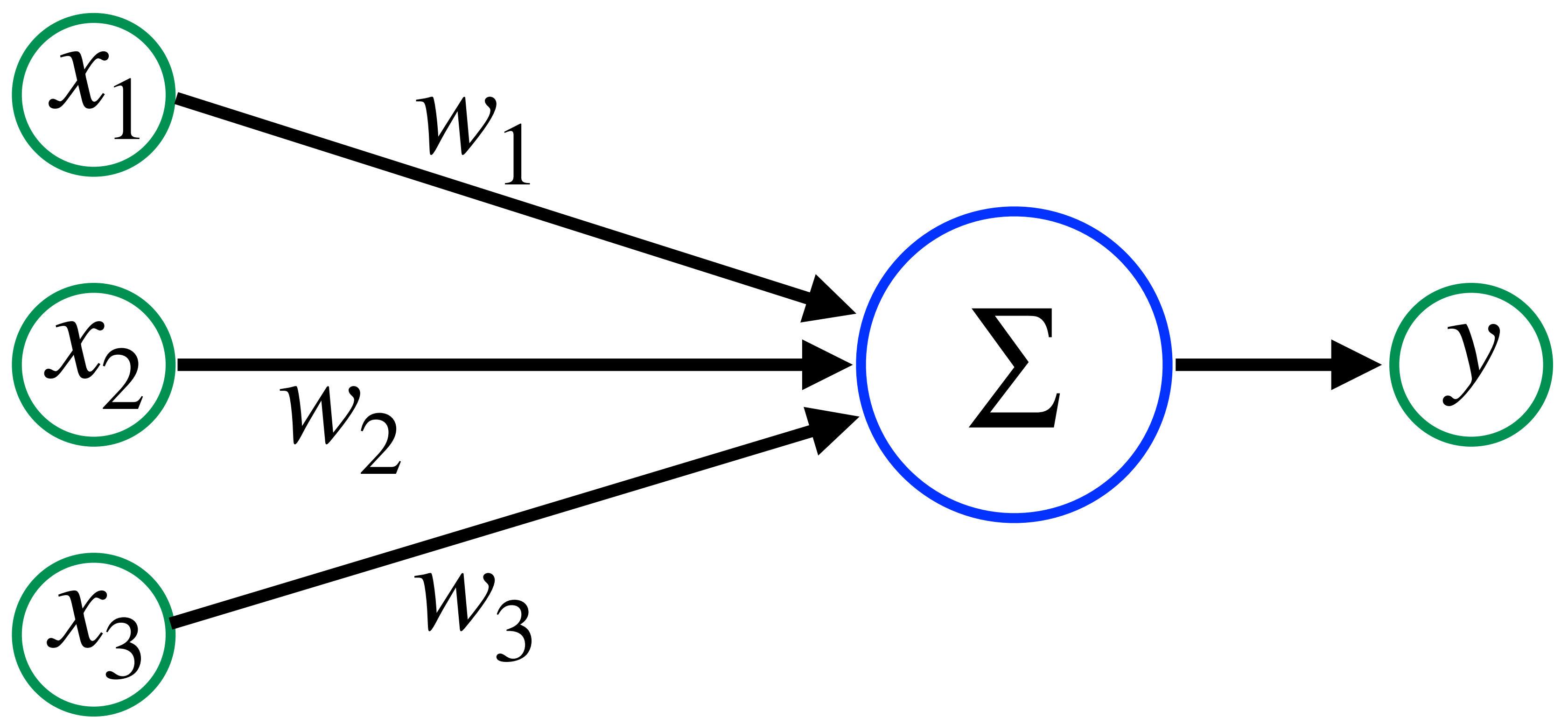
Hebbian learning and the perceptron

> “Let us assume that the persistence or repetition of a reverberatory activity tends to induce lasting cellular changes that add to its stability. … When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased”

Donald Hebb (The Organisation of Behaviour, 1949)

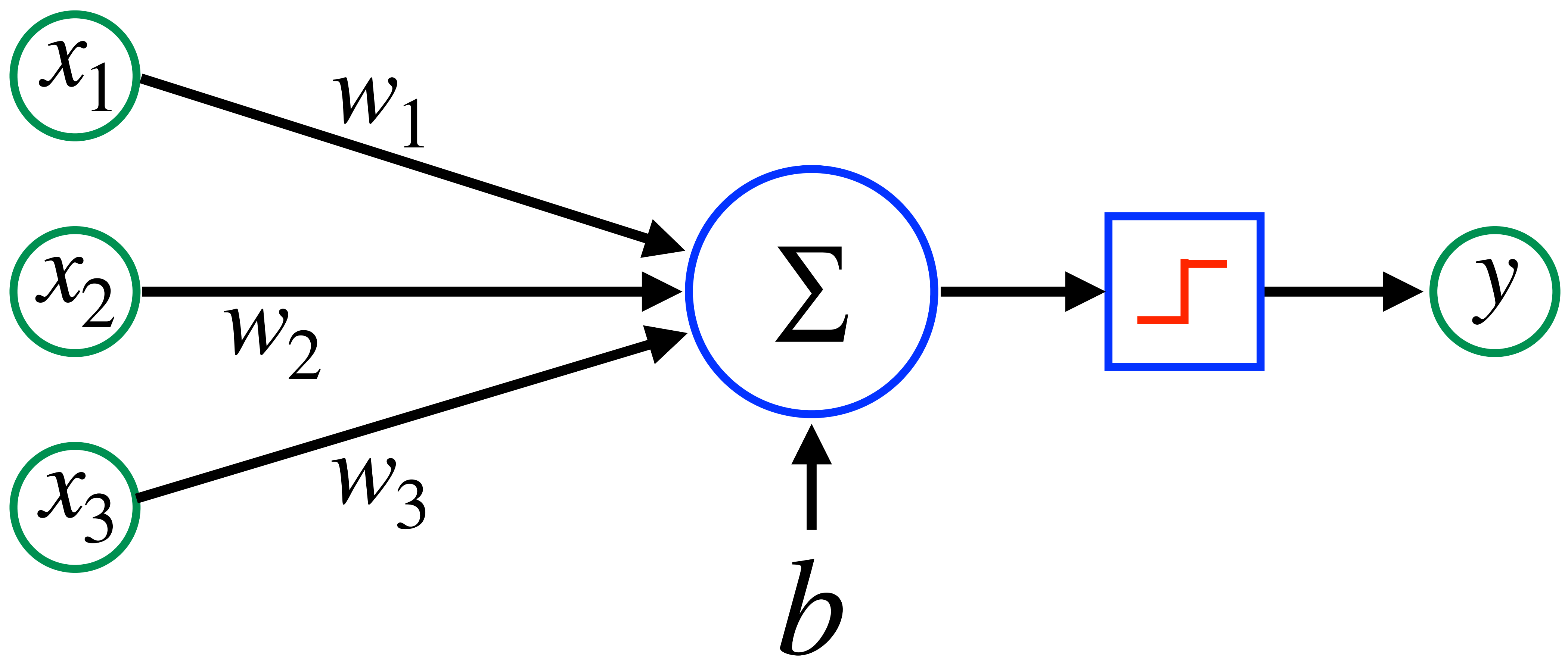


Hebbian learning and the perceptron



Perceptron

Rosenblatt (1958)

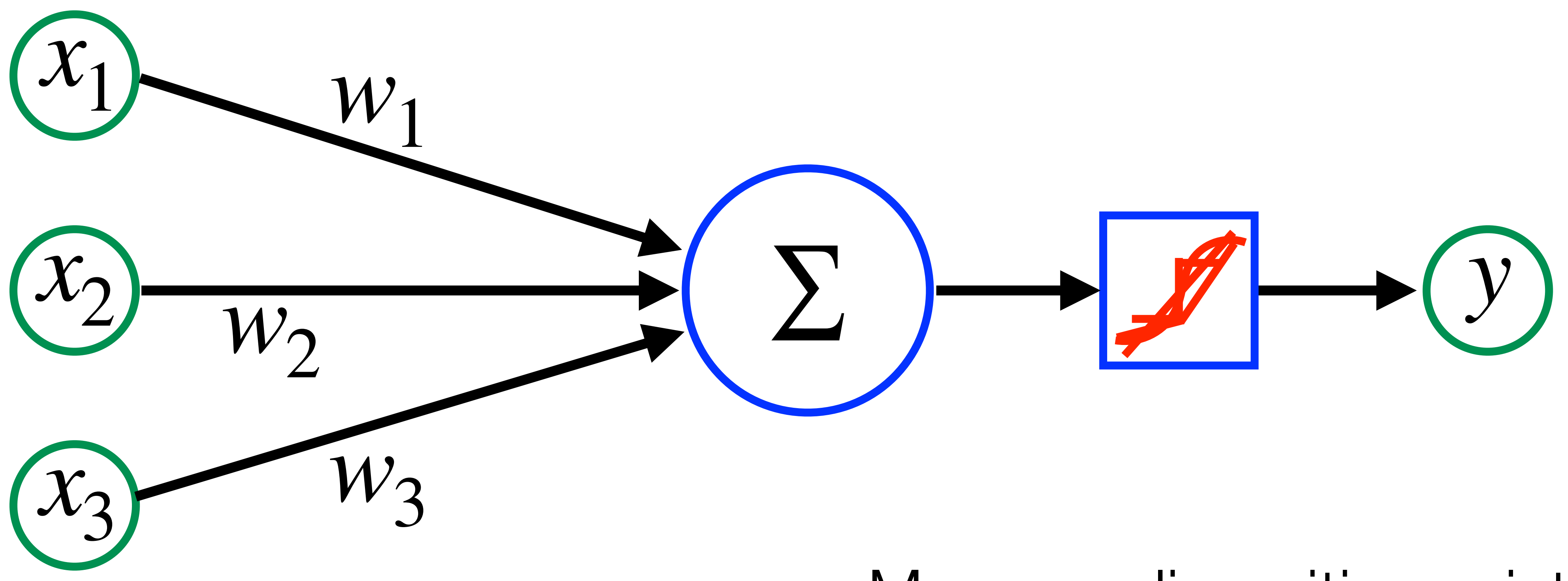


$$y = f(x) = \begin{cases} 1 & \text{if } \sum_i w_i x_i + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

Hebbian learning and the perceptron

Perceptron

Rosenblatt (1958)



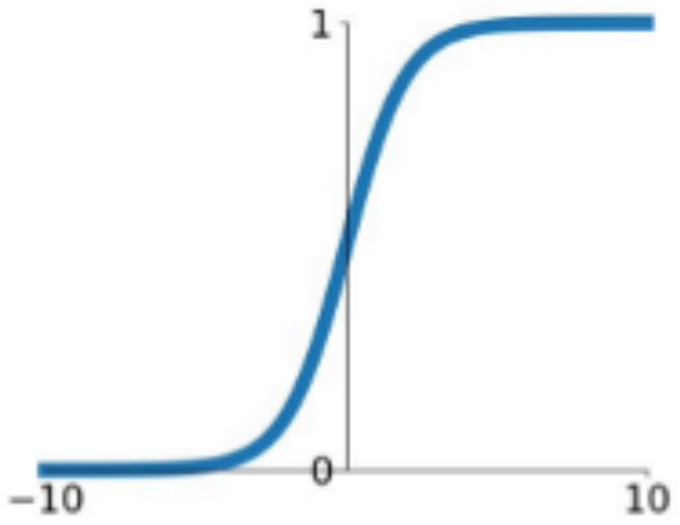
Many nonlinearities exist:

- tanh
- sigmoid
- RELU
- Leaky RELU

Many flavours of activation functions

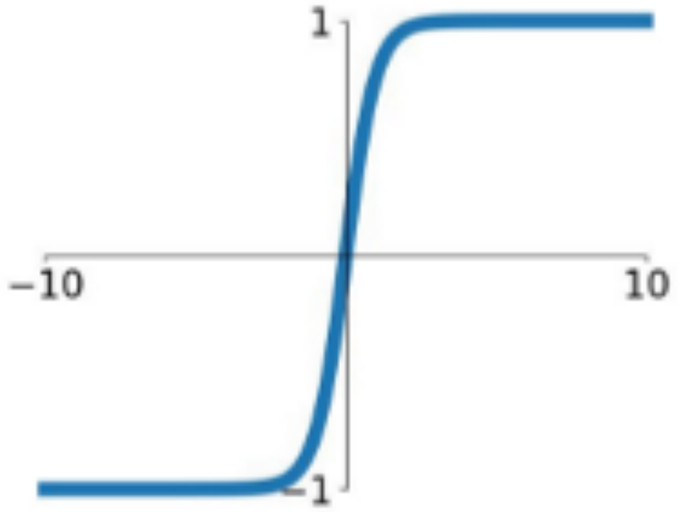
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



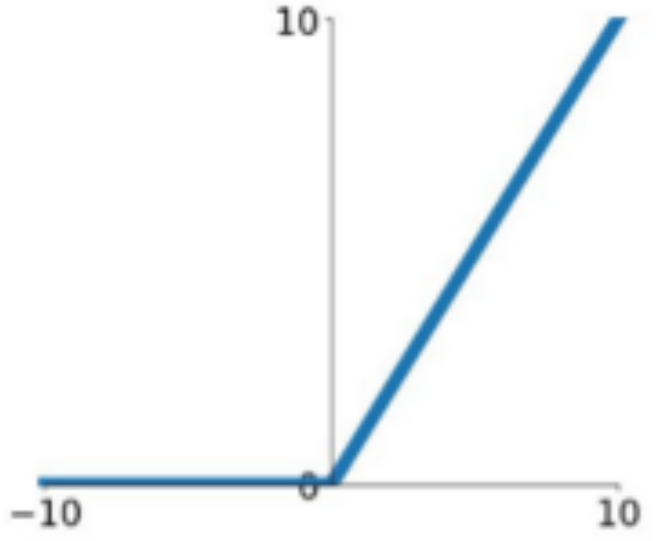
tanh

$\tanh(x)$



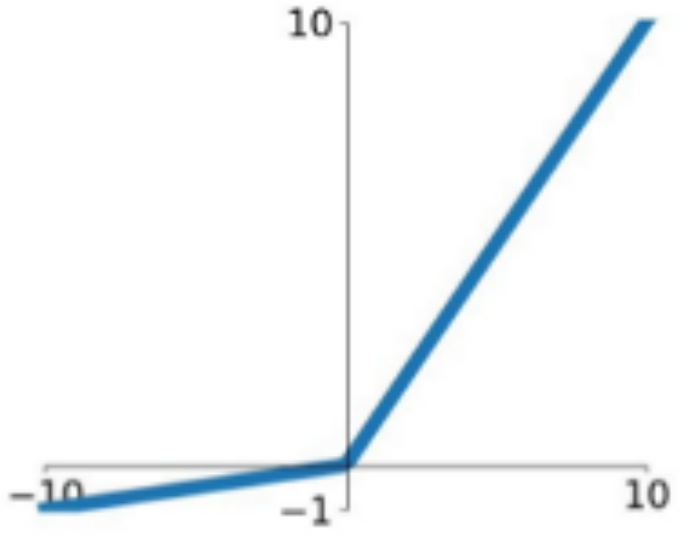
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

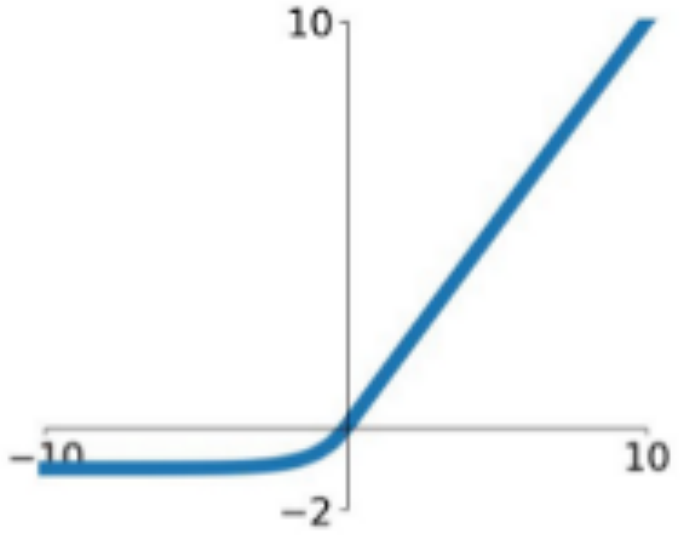


Maxout

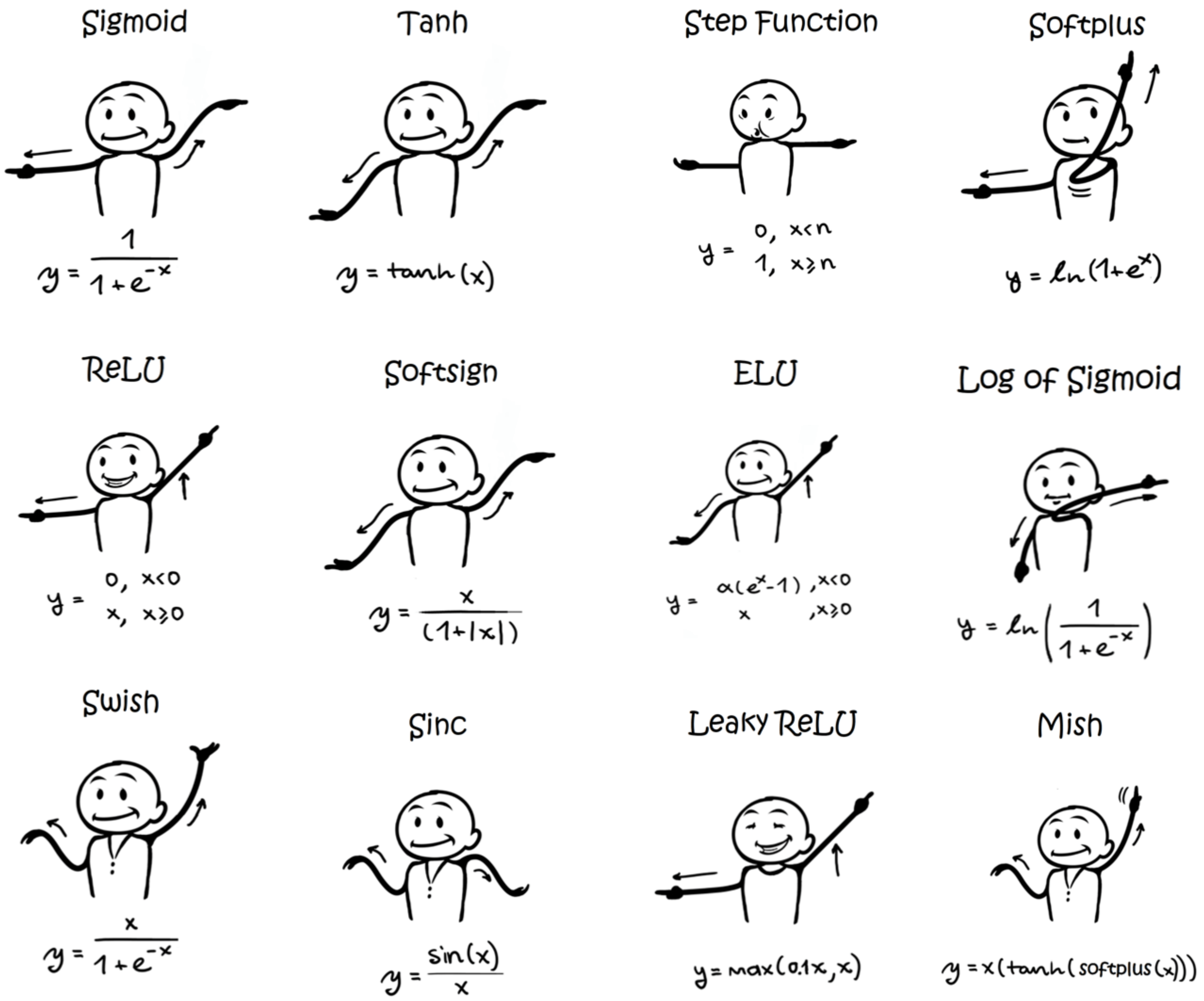
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

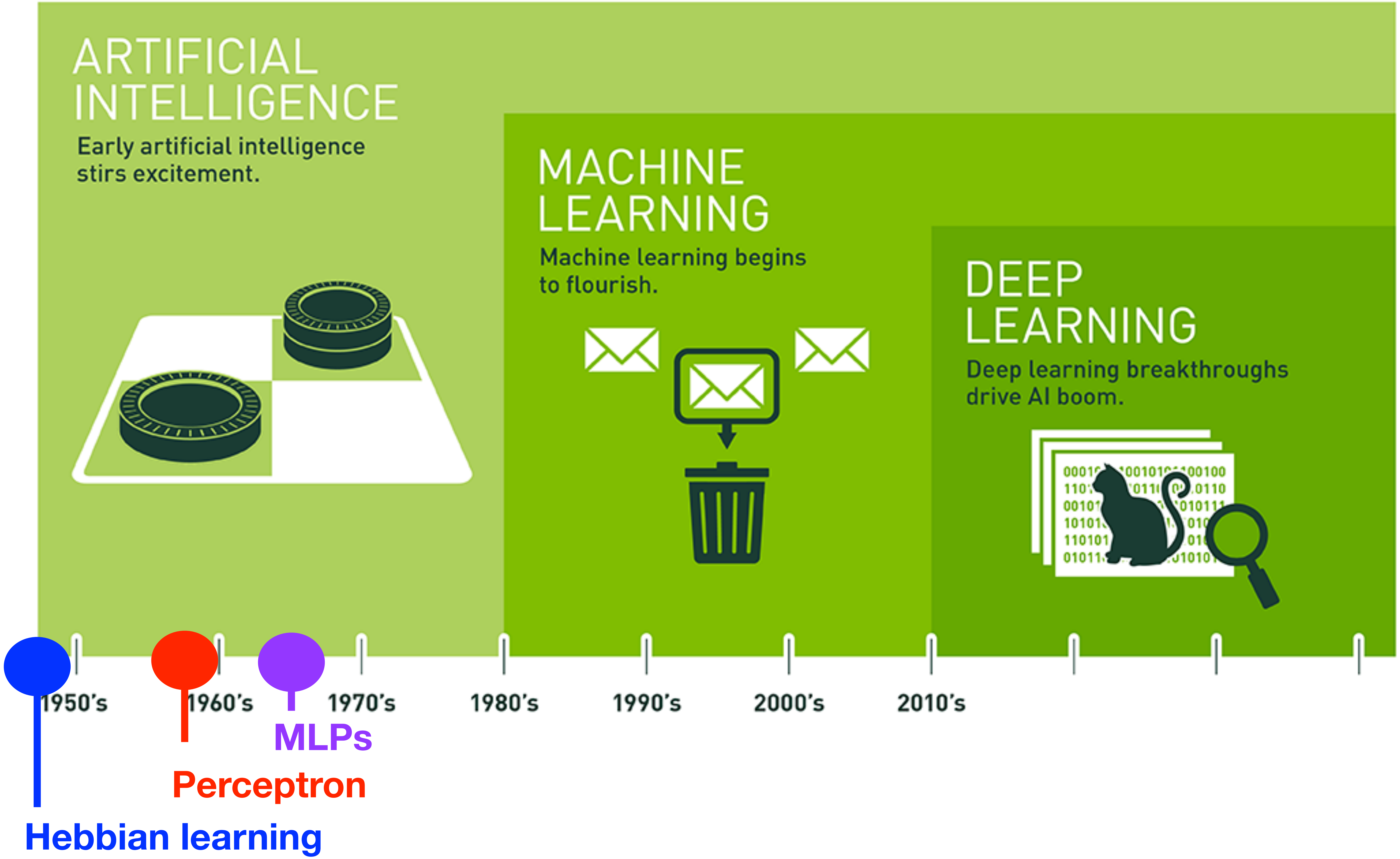
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Yes… there is also an activation function dance…



Machine Learning and Deep Learning

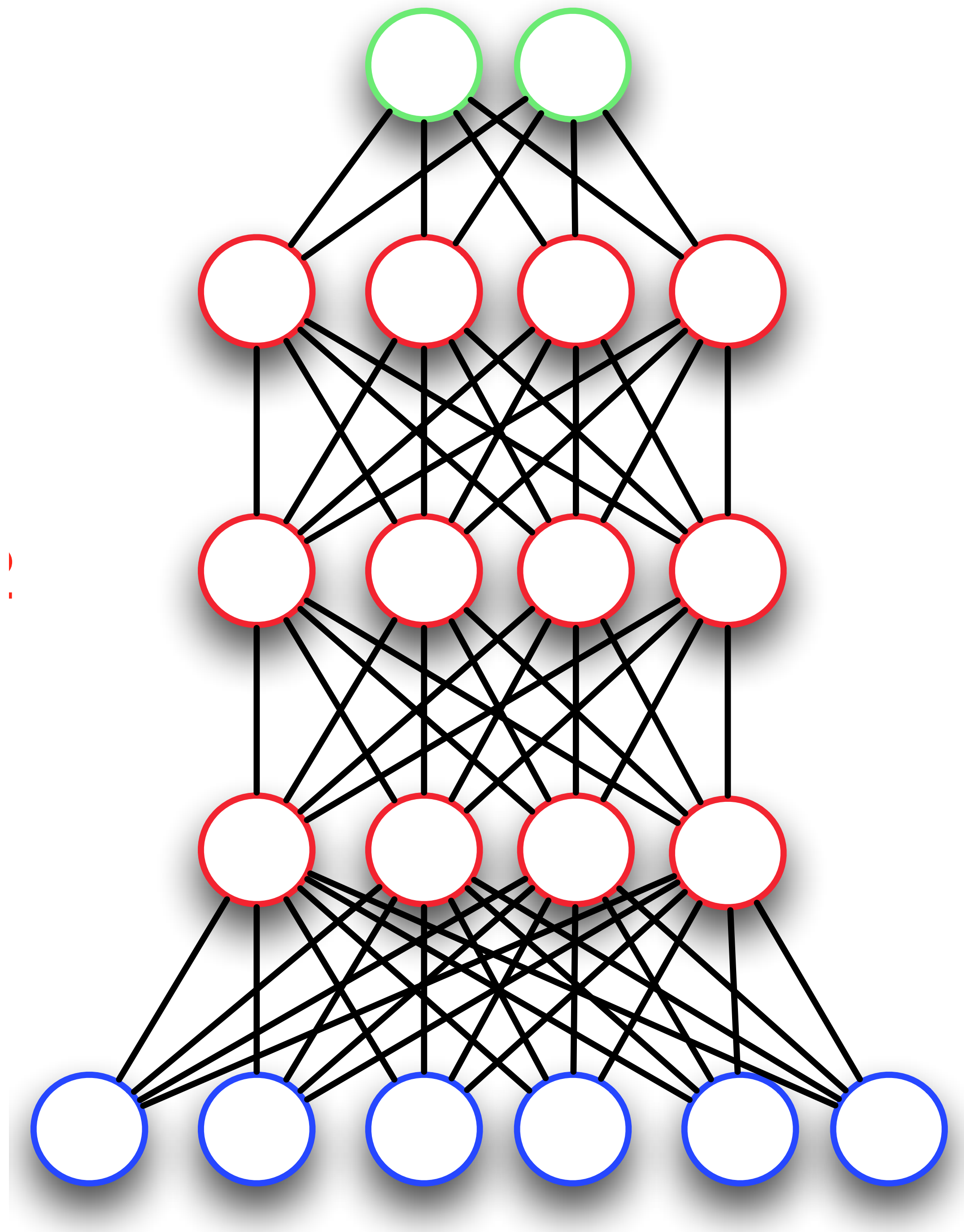


https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Multi-layer Perceptron

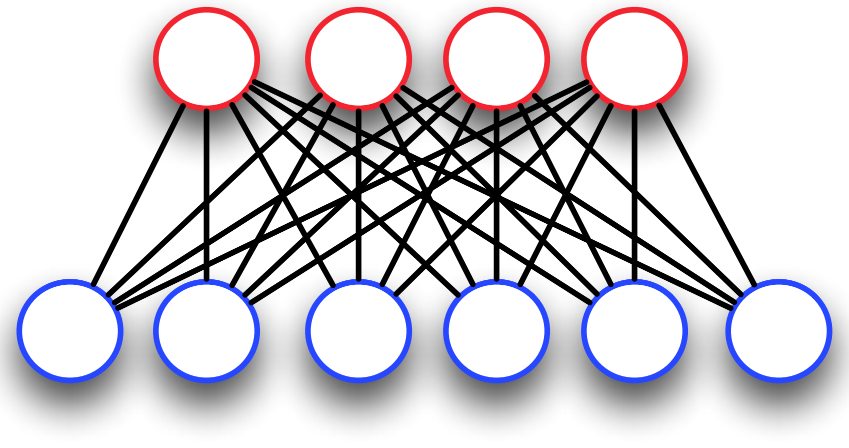
- First introduced by Rosenblatt in 1958 along with the Perceptron
- Usually trained by backpropagation (first introduced in 1970 as the inverse of automatic differentiation. Came back into fashion in the 2010s when GPUs became readily available.
- Calculate the derivative of the cost function $C(y, g(x))$ using chain rule

Multi-Layer Perceptron (MLP)



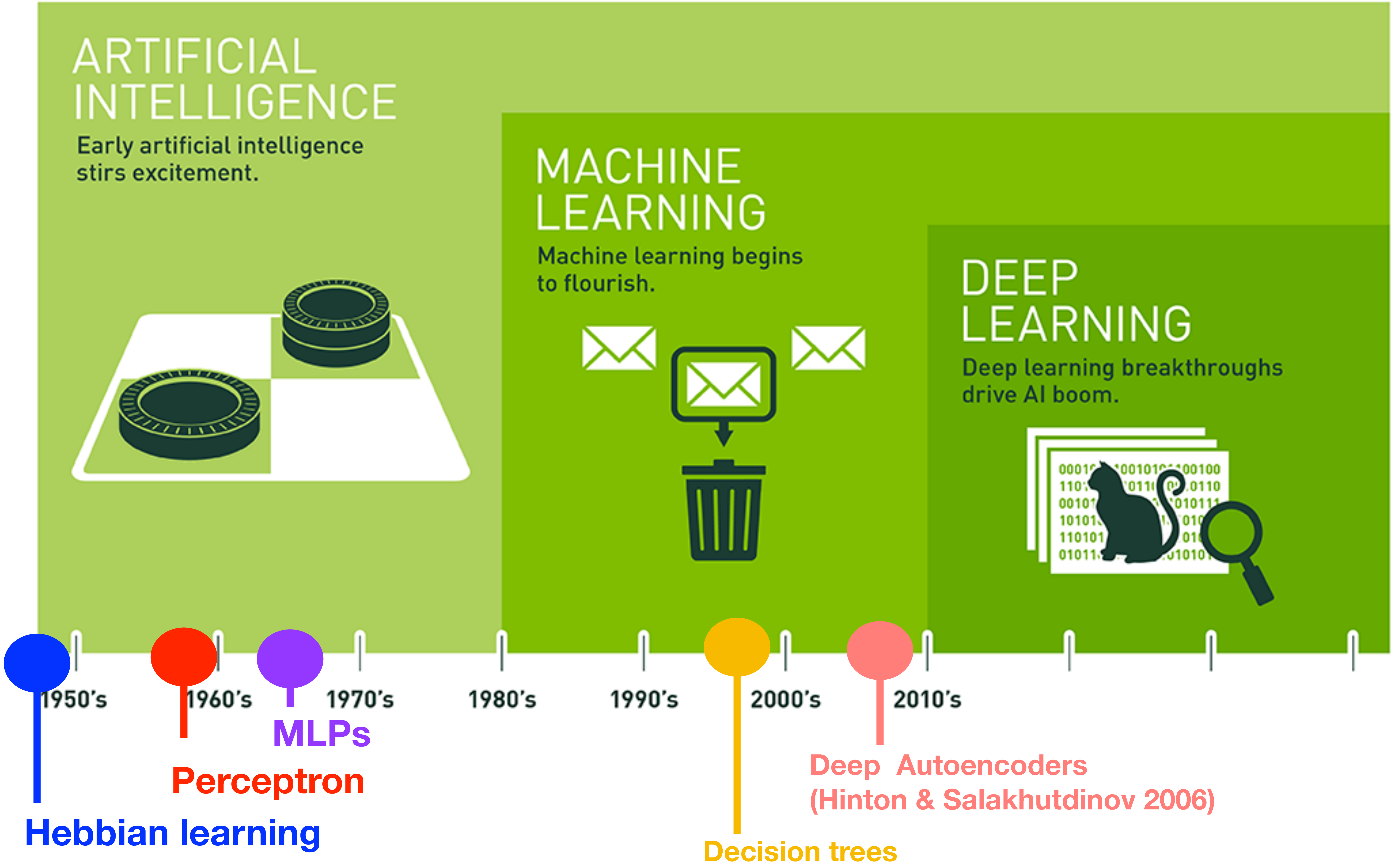
$$g(X) := f_L(f_{L-1}(\ldots f_0(W_0x + b_0)))$$

Single layer



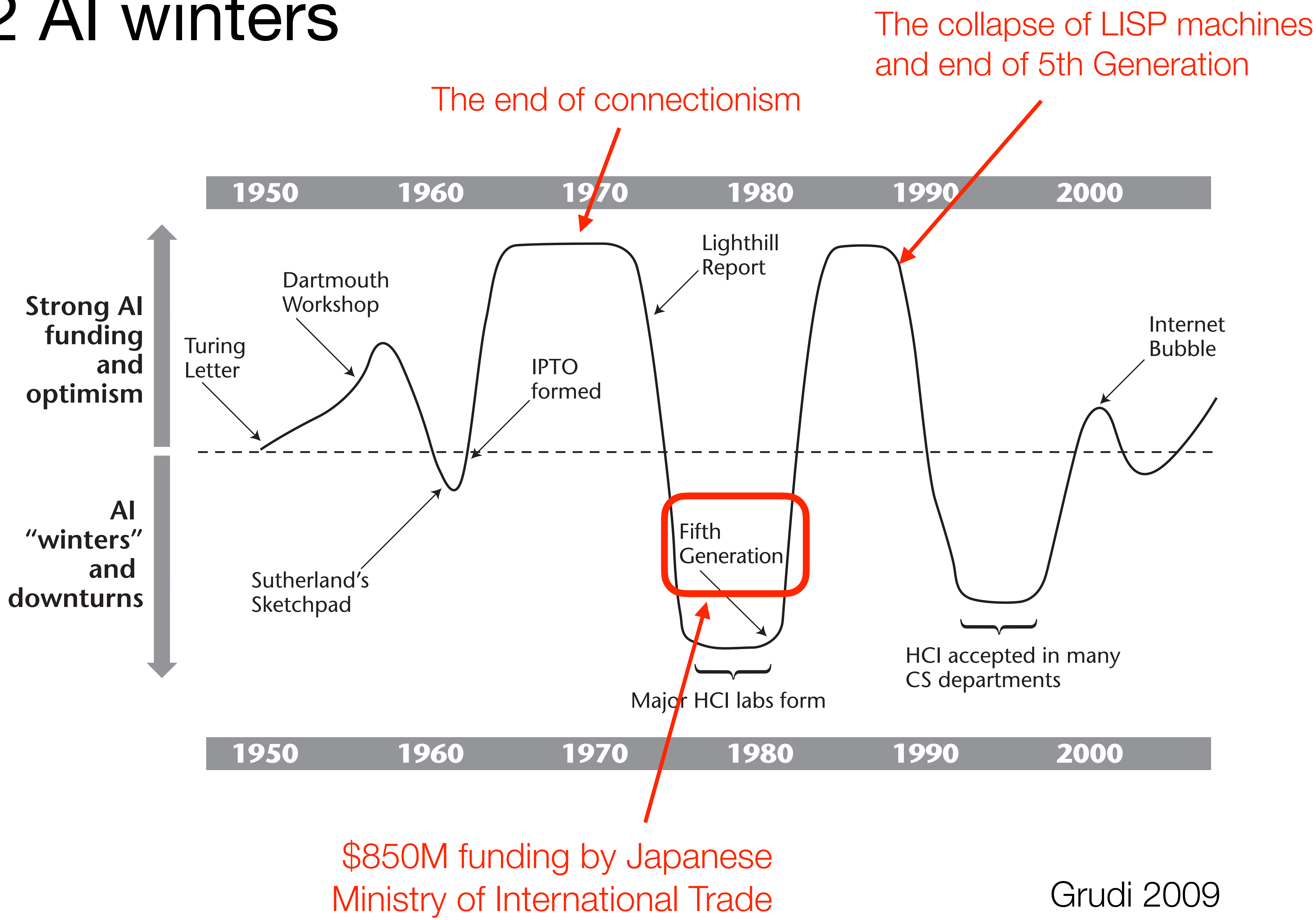
$$y = f\left(\sum_i w_i x_i + b\right)$$

Machine Learning and Deep Learning



https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

The 2 AI winters

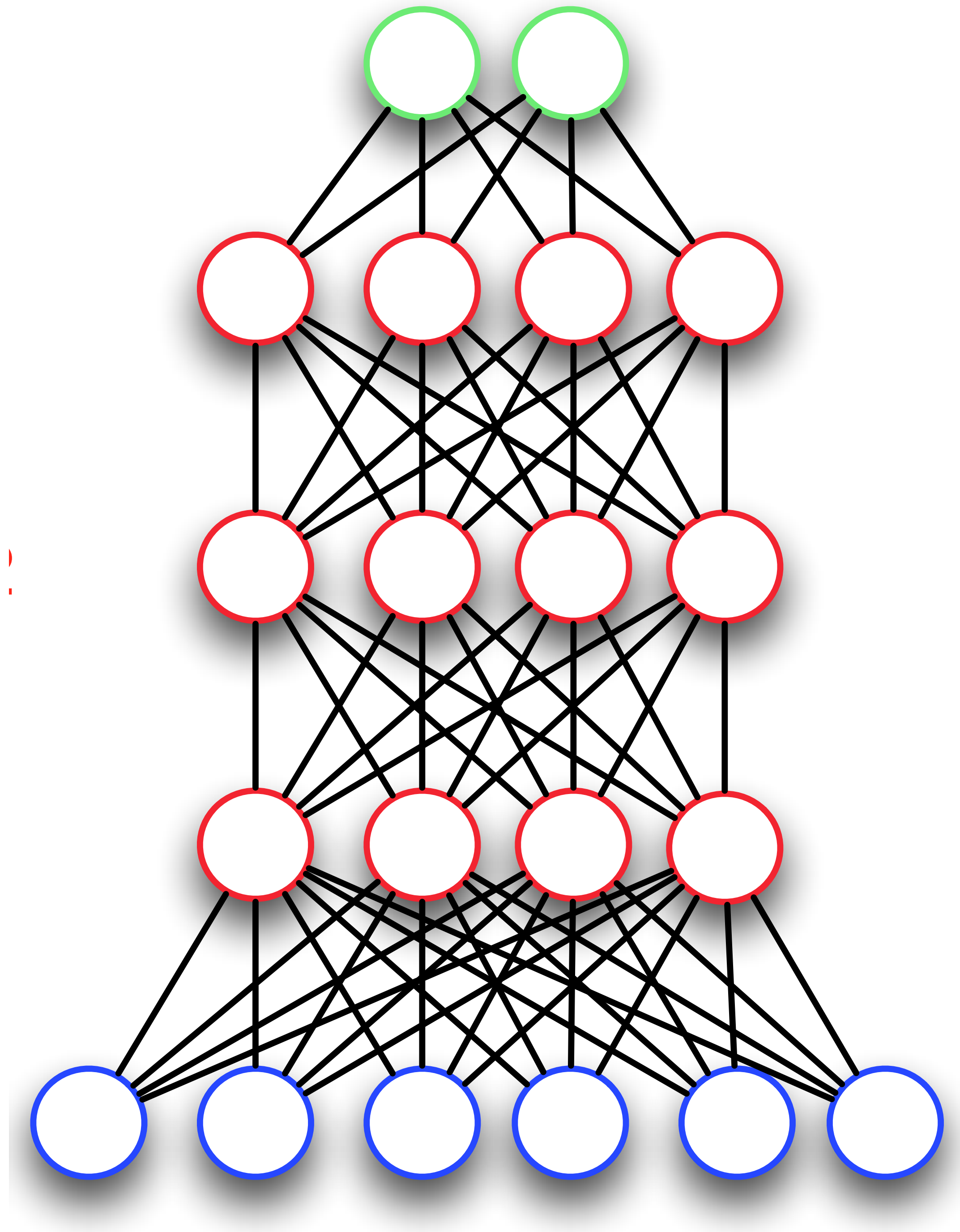


Grudi 2009

Feed forward nets & Autoencoders

/imagine prompt: a feed forward neural network modelling exoplanet atmospheres

Let’s simplify our pictograms



Multi-layer perceptron (MLP)
Feed forward network

Accurate Machine-learning Atmospheric Retrieval via a Neural-network Surrogate Model for Radiative Transfer

Michael D. Himes[1], Joseph Harrington[2], Adam D. Cobb[3], Atılım Güneş Baydin[3], Frank Soboczenski[4], Molly D. O'Beirne[5], Simone Zorzan[6], David C. Wright[1], Zacchaeus Scheffer[1], Shawn D. Domagal-Goldman[7], and Giada N. Arney[7]

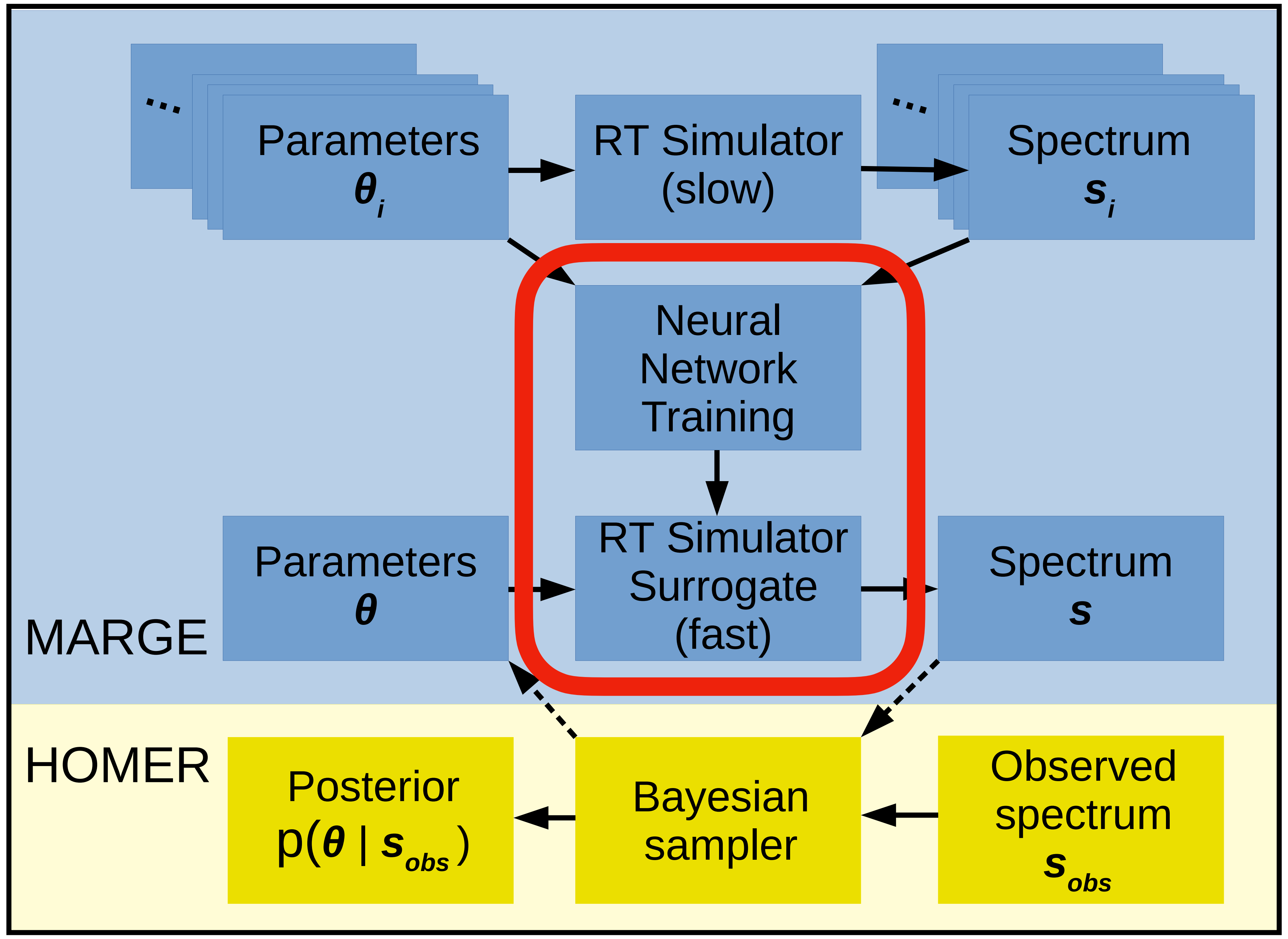
[1] Planetary Sciences Group, Department of Physics, University of Central Florida, USA; mhimes@knights.ucf.edu
[2] Planetary Sciences Group, Department of Physics and Florida Space Institute, University of Central Florida, USA
[3] Department of Engineering Science, University of Oxford, UK
[4] SPHES, King's College London, UK
[5] Department of Geology and Environmental Science, University of Pittsburgh, USA
[6] ERIN Department, Luxembourg Institute of Science and Technology, Luxembourg
[7] NASA Goddard Space Flight Center, Greenbelt, MD, USA

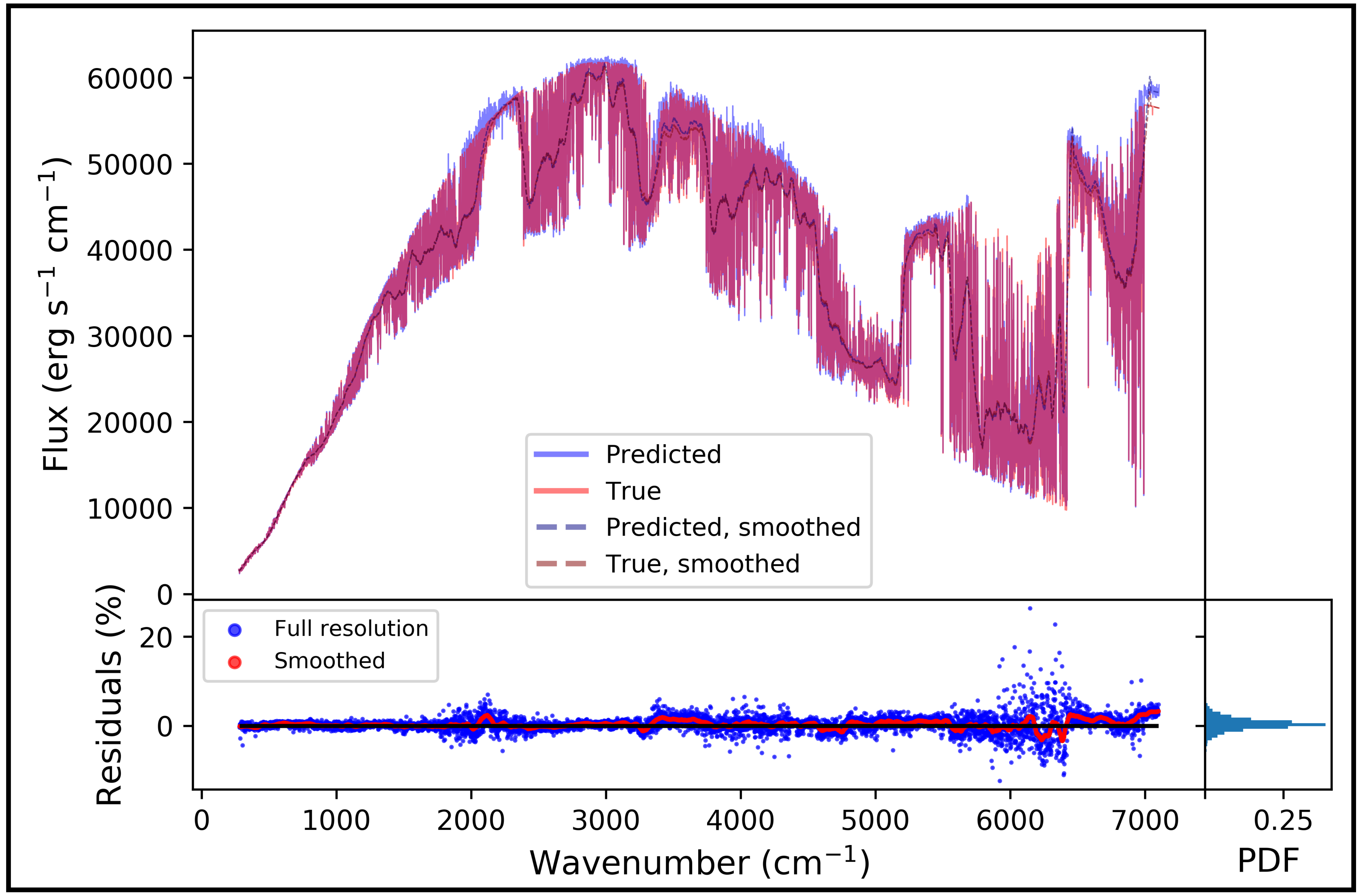
Received 2020 March 4; revised 2021 January 22; accepted 2021 February 4; published 2022 April 25

Neural Network surrogate models

Replacing the slow astrophysical model with a faster Neural Network surrogate

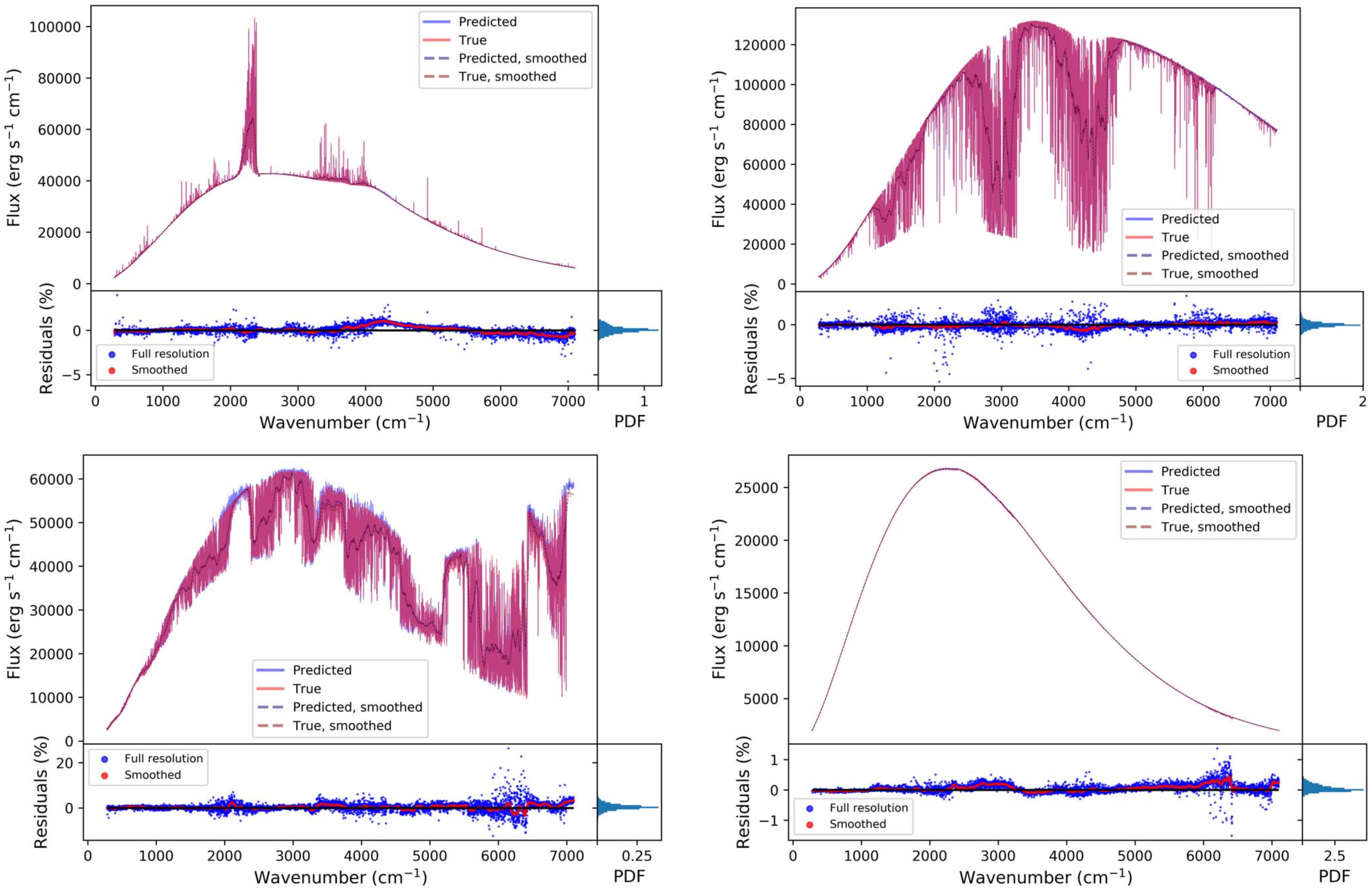
Maintaining traditional Bayesian sampling to calculate parameter distributions





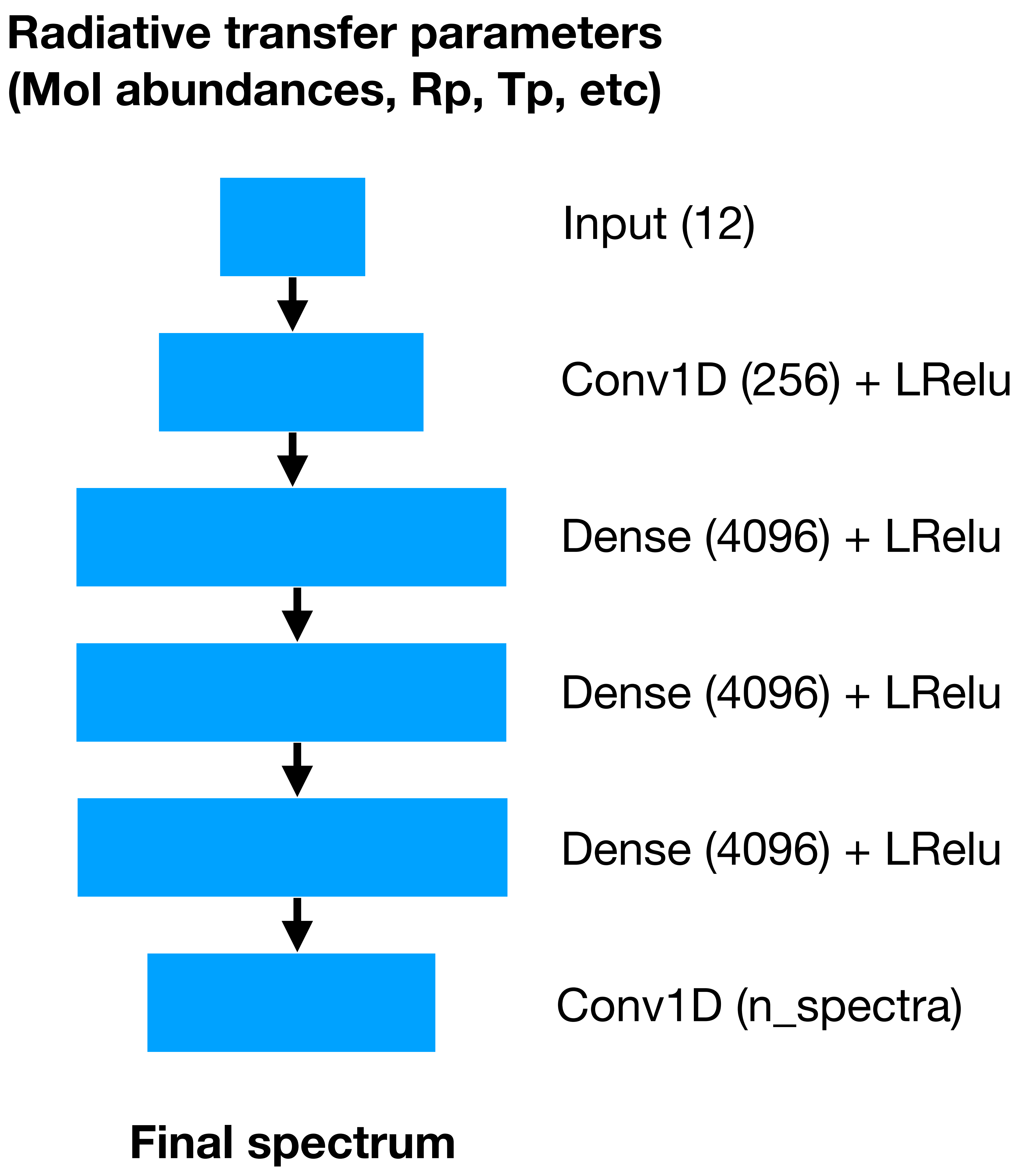
Himes et al. 2021

Doing a good job at approximating the radiative transfer model



Himes et al. 2021

Reproducing their model in PyTorch




```python
import torch
import torch.nn as nn

#setting up model
model = nn.Sequential(
    nn.Linear(12,256)
    nn.Conv1D(256, 4096,1,stride=1),
    nn.LeakyReLU(),
    nn.Linear(4096, 4096),
    nn.LeakyReLU(),
    nn.Linear(4096, 4096),
    nn.LeakyReLU(),
    nn.Conv1D(n,1,stride=1),
)

#definining loss function
loss_fn = nn.MSELoss()
loss = loss_fn(spectra, forward_model_parameters)

#defining optimizer
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)

#training the model over num_epochs cycles
for n in range(num_epochs):
    y_pred = model(X)
    loss = loss_fn(y_pred, y)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

King & Ba 2014

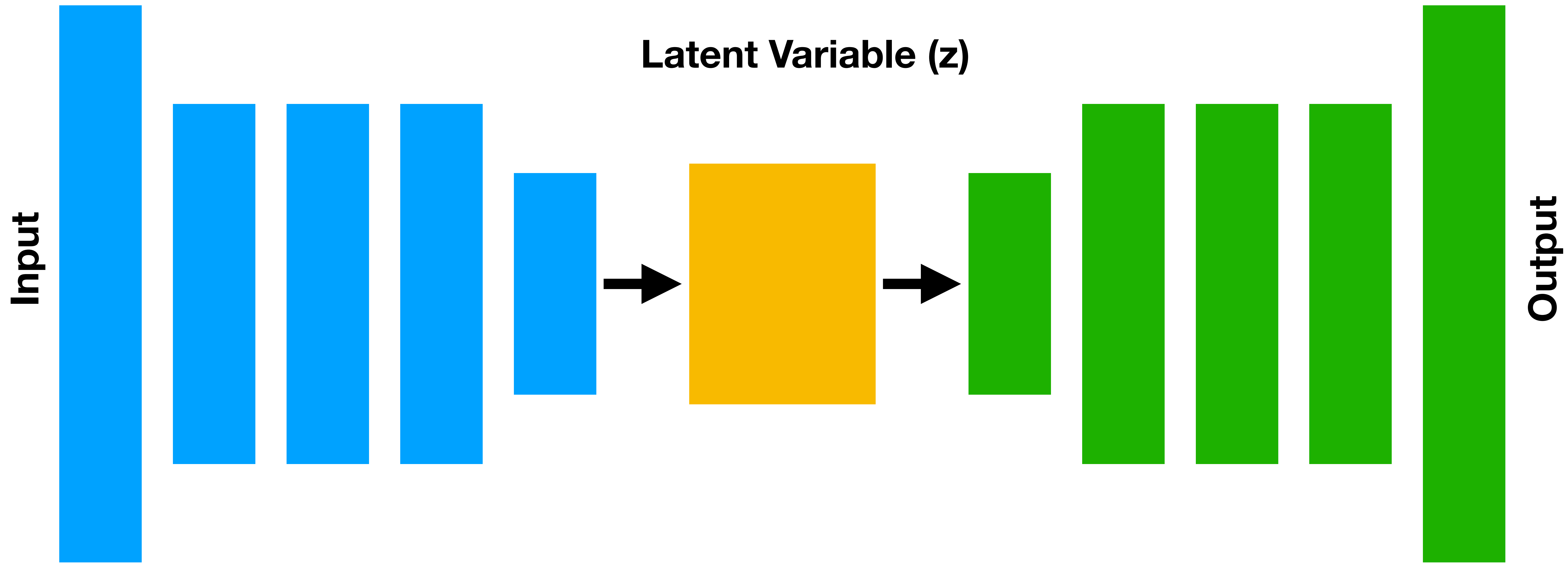
Himes et al. 2021

Autoencoders

Multi-layer perceptron (MLP)
Feed forward network

Read chapter 14 in: https://www.deeplearningbook.org/

Autoencoders



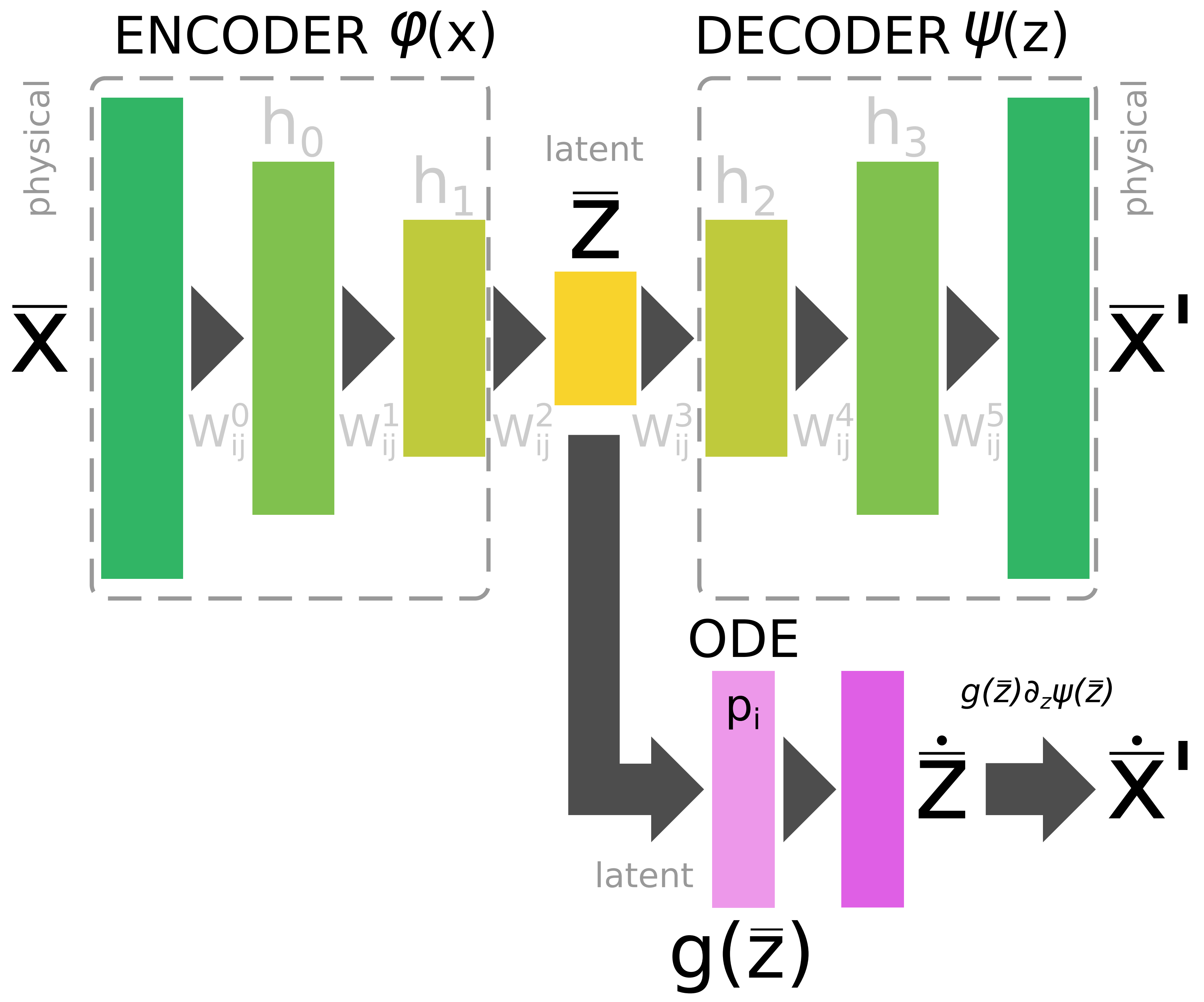
- Output must equal input
- Self-supervised learning
- Non-linear data compression and clustering
- Not probabilistic but variational autoencoders are an easy modification

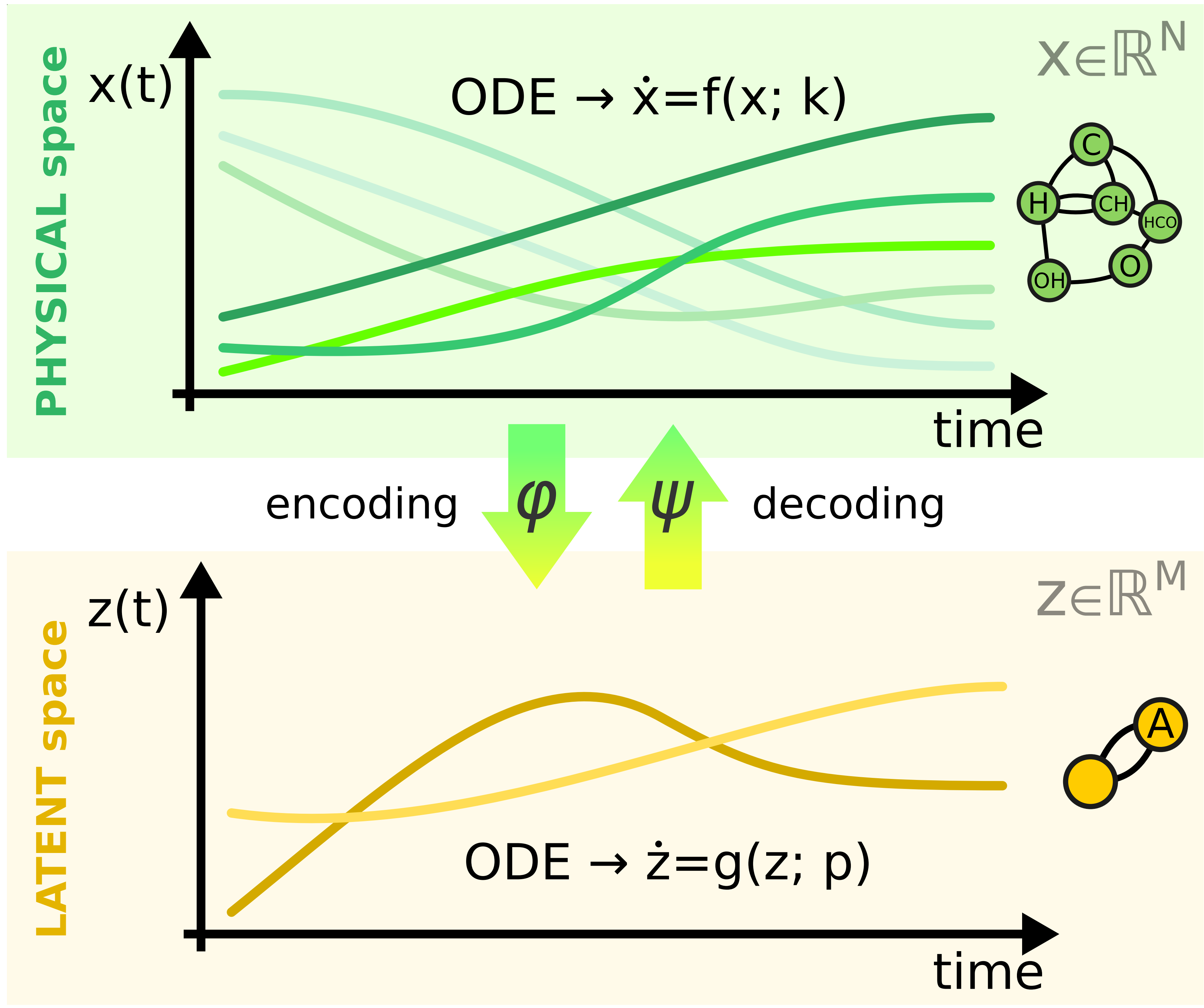
Cost function = Input - Output

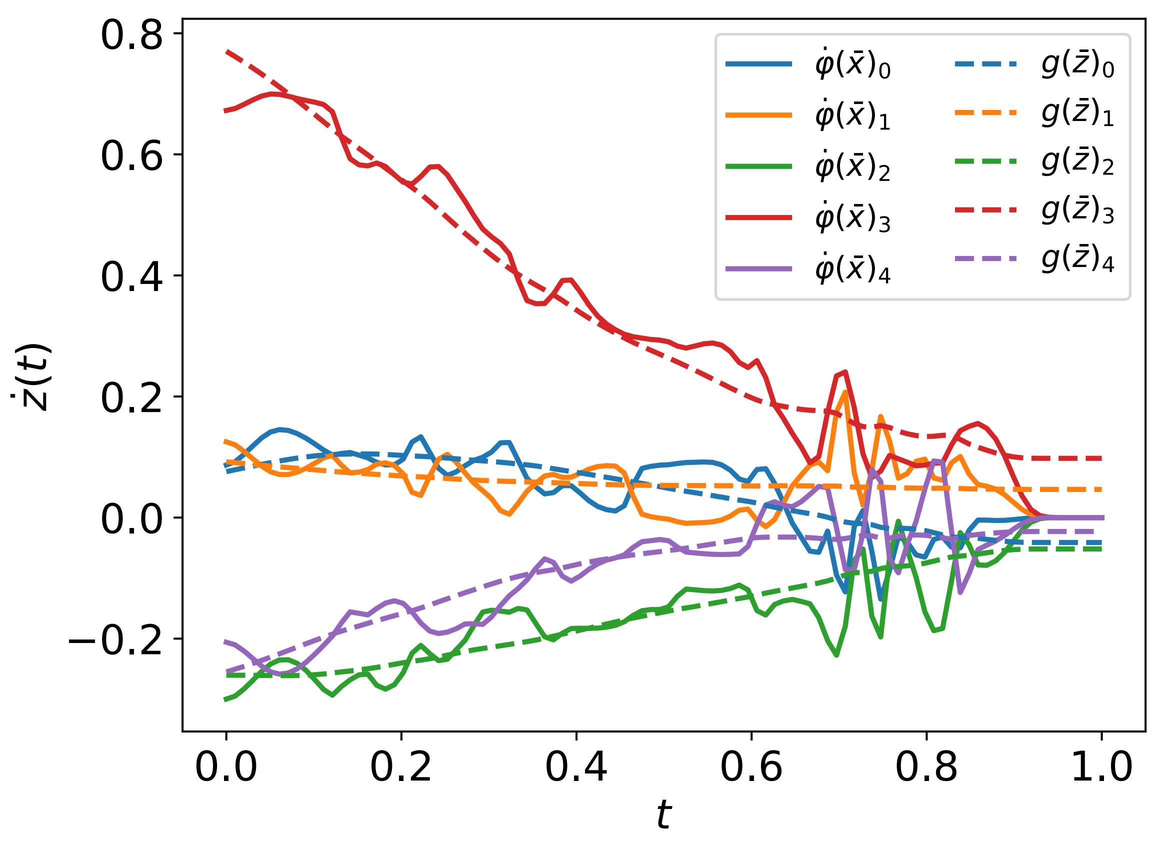
Read chapter 14 in: https://www.deeplearningbook.org/

Modelling time evolution of ODEs in chemical networks

Evolving complex chemical networks in compressed latent space



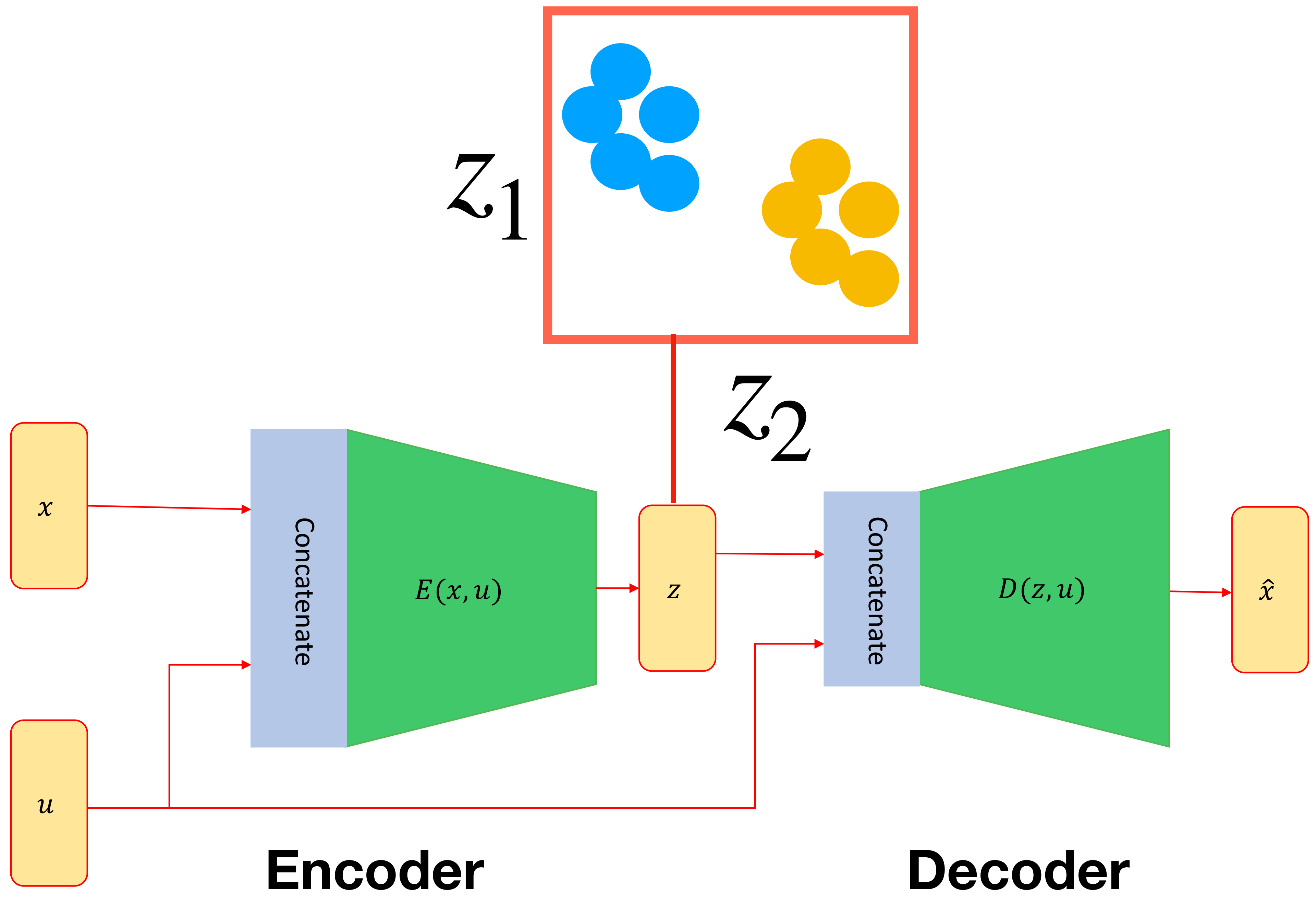




Grassi et al. 2021

Disentangling Complex Chemistry in Astrochemistry

- Using Conditional Autoencoders to transform data (x) to a lower dimensional representation (z)
- Latent variables (z) should ideally cluster in a physically interpretable way
- Enforcing statistical separation using loss function is an example of active explainability



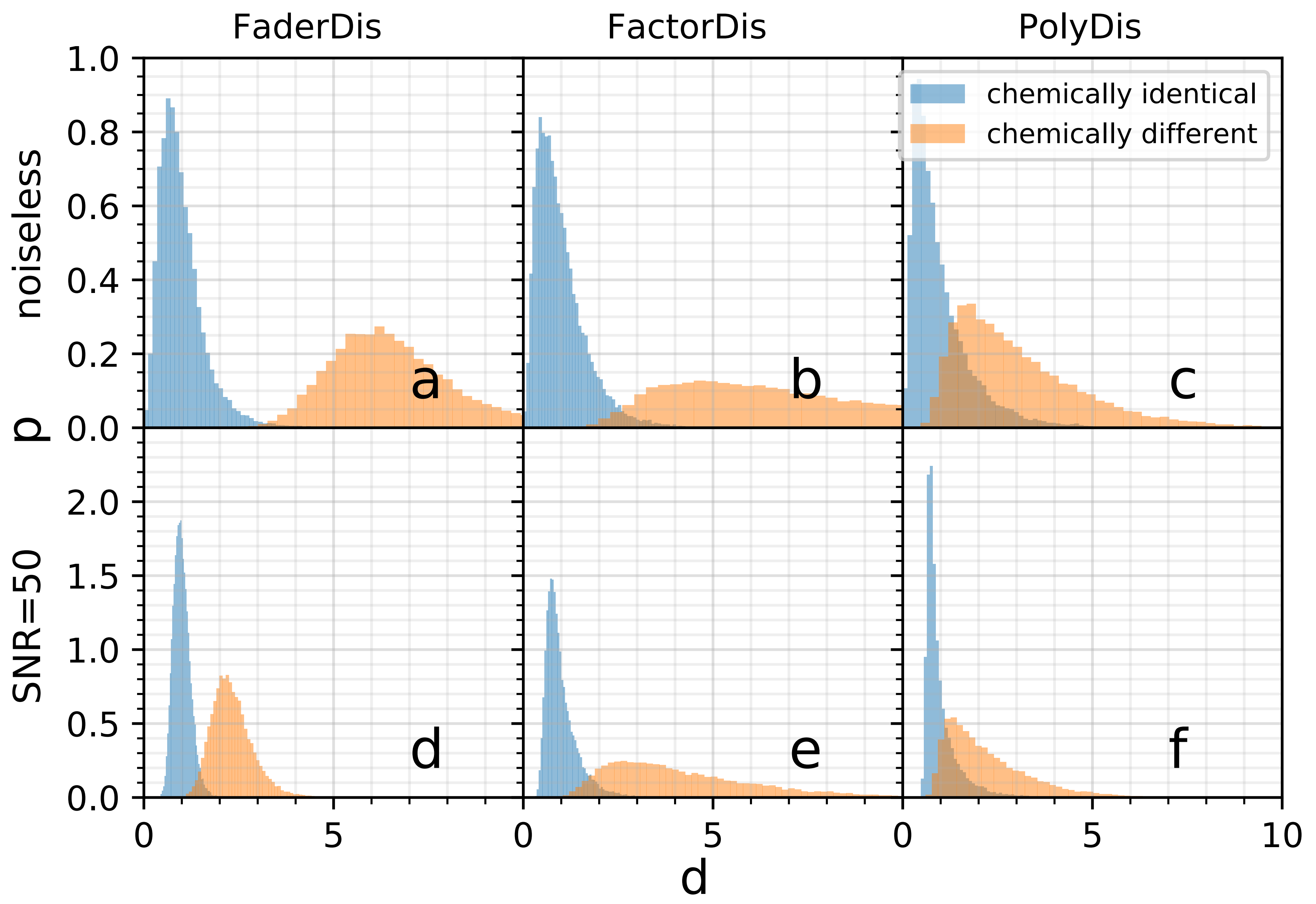


Figure 2. Distribution of scaled euclidian distances, d, for a sample of chemically identical pairs of stars (blue) and fully randomly sampled pairs of stars (orange). For each model, a scaling is applied to the latents such that the mean distance of chemically identical stars is 1. Each model includes $T_{\rm eff}$, $\log g$ and [Fe/H], as the parameters to disentangle from the chemical factors of variation. The top row is evaluated using the noiseless test dataset, the bottom with noise of order SNR=50 added. The first column is evaluated using the FaderDis method, the second using the FactorDis method and the final row using the PolyDis method (after PCA with 50 components).

Lim et al. 2018

Bayesian Neural Networks

/imagine prompt: Thomas Bayes explaining a neural network to his students in the 18th century

Recap: The Bayes theorem

Posterior **Likelihood** **Prior**

$$P(\theta | D) = \frac{P(D | \theta) P(\theta)}{P(D)}$$

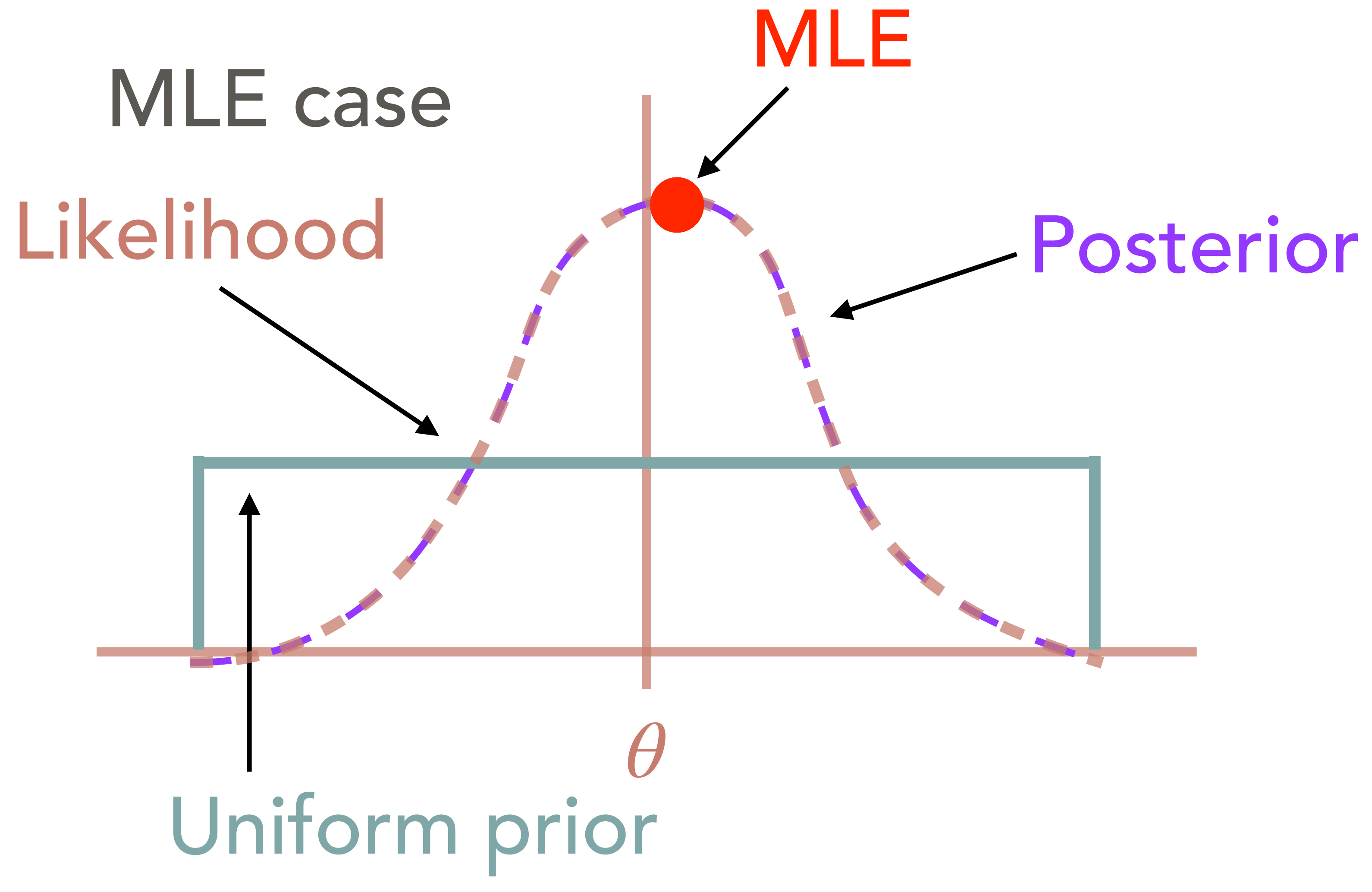
Evidence

$$P(D) = \int P(D | \theta) P(\theta) d\theta$$

Maximum Likelihood Estimation (MLE) and Maximum A Posteriori (MAP)

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

- MLE and MAP are almost the same thing and only differ by the prior distribution



Maximum Likelihood Estimate

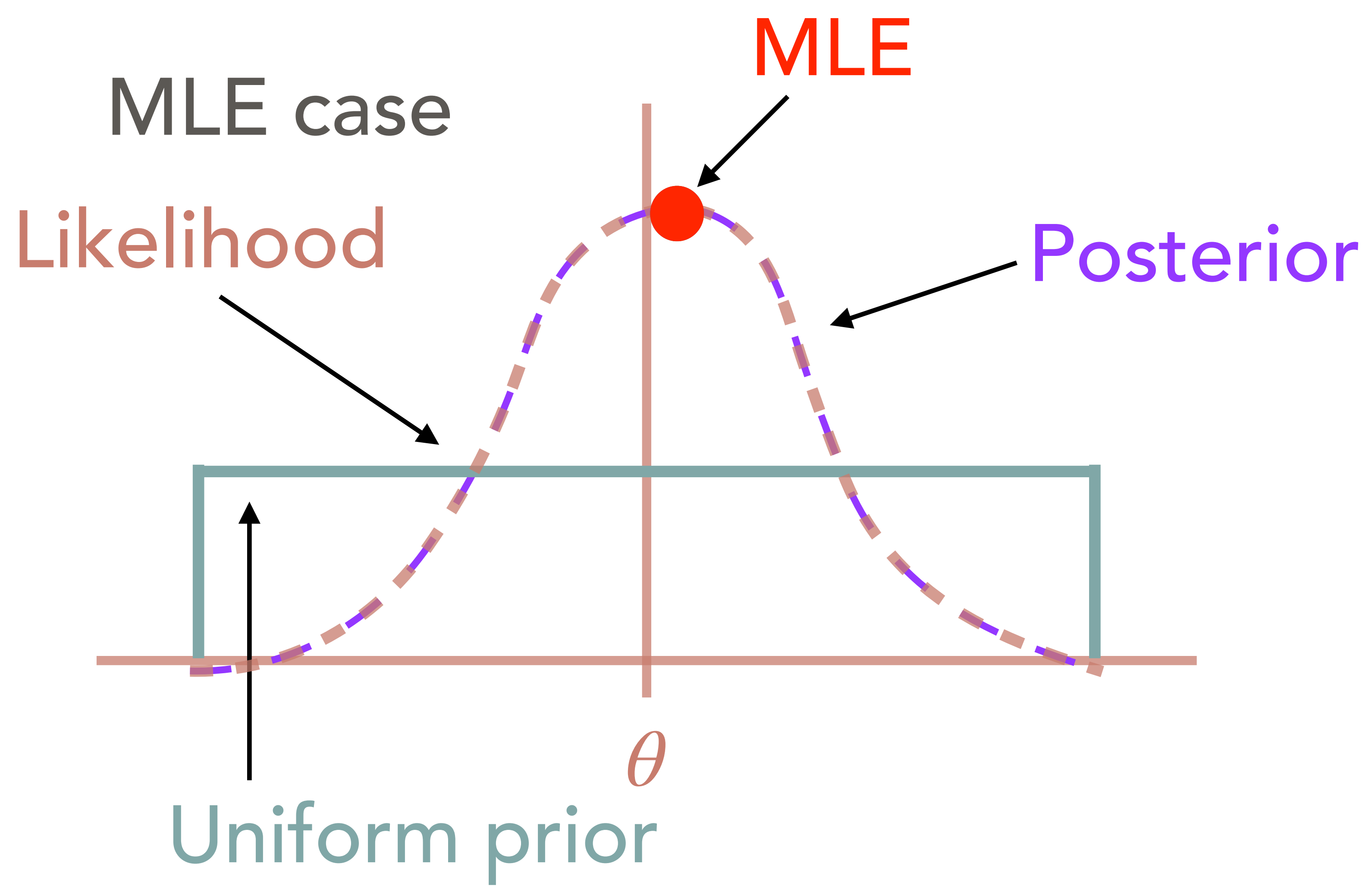
$$\hat{\theta}_{MLE}(D) = \underset{\theta}{\operatorname{argmax}}\, P(D|\theta)$$

- It's literally the maximum of the likelihood.
- In the case of a Gaussian likelihood, it's equivalent to the lowest χ^2

Maximum Likelihood Estimation (MLE) and Maximum A Posteriori (MAP)

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{P(D)}$$

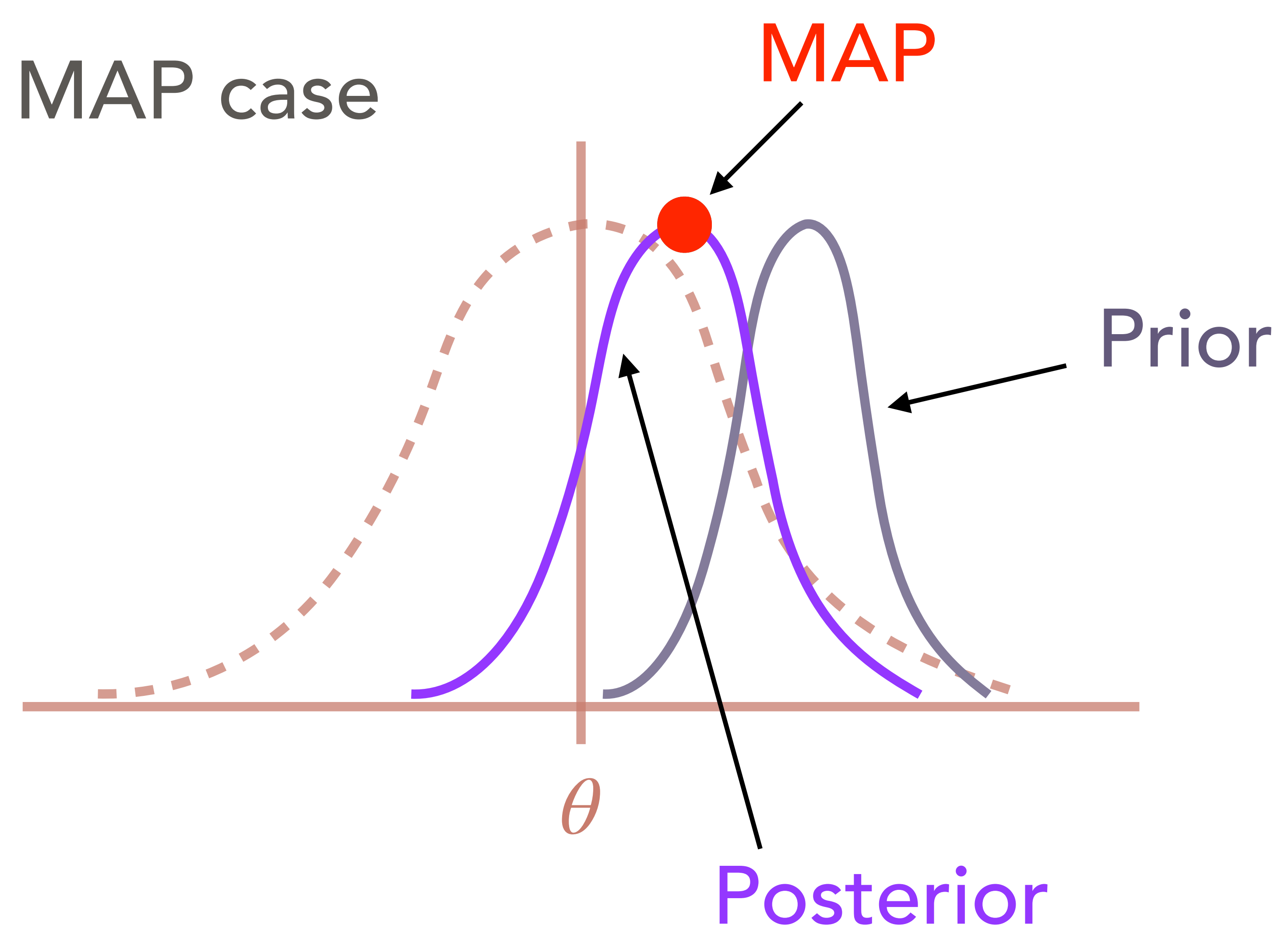
- MLE and MAP are almost the same thing and only differ by the prior distribution



<u>Maximum Likelihood Estimate</u>

$$\hat{\theta}_{MLE}(D) = \underset{\theta}{\operatorname{argmax}}\, P(D \mid \theta)$$

- It's literally the maximum of the likelihood.
- In the case of a Gaussian likelihood, it's equivalent to the lowest χ^2



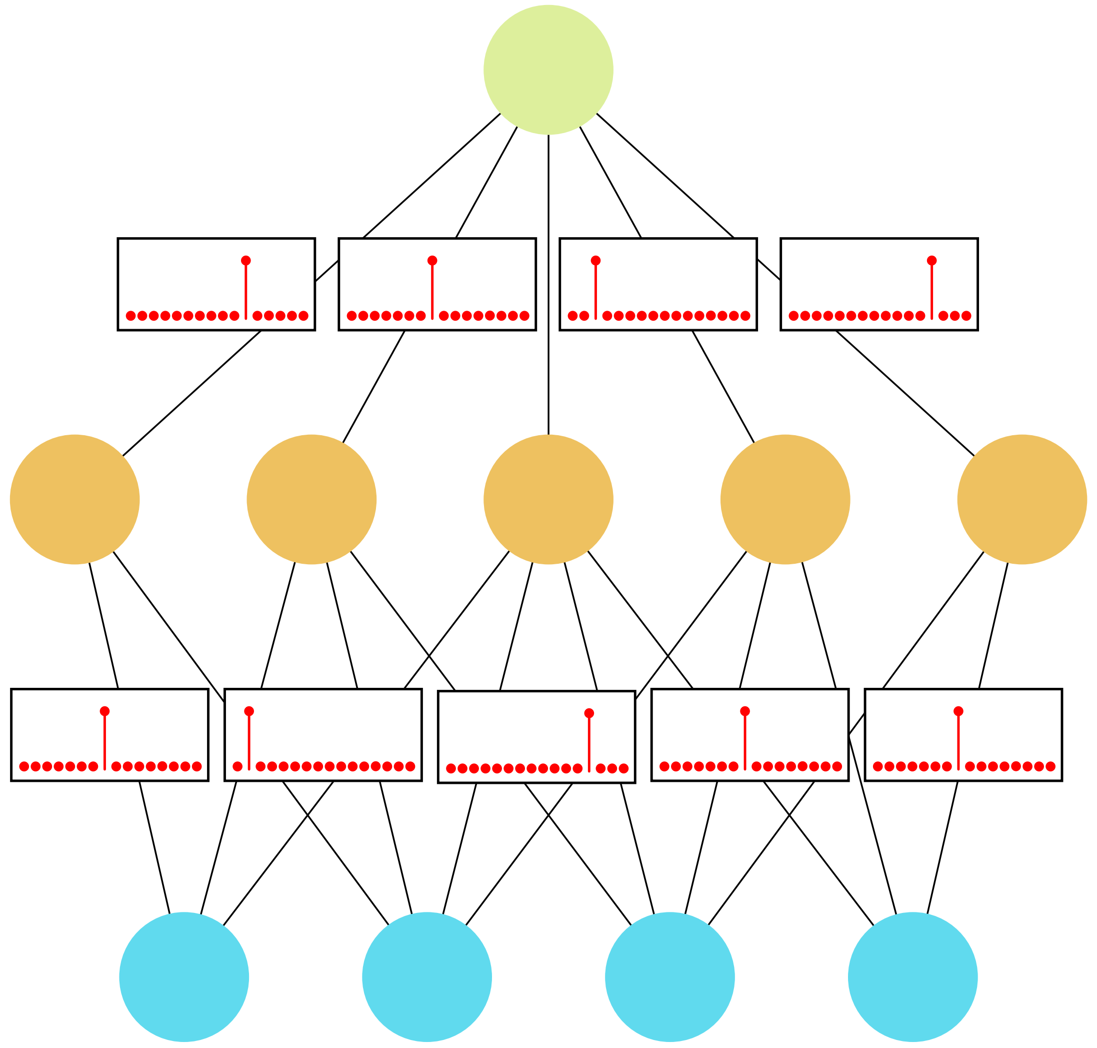
<u>Maximum A Posteriori</u>

$$\begin{aligned}\hat{\theta}_{MAP}(D) &= \underset{\theta}{\operatorname{argmax}}\, P(\theta \mid D) \\ &= \underset{\theta}{\operatorname{argmax}}\, P(D \mid \theta)P(\theta)\end{aligned}$$

- It's literally the maximum of the posterior.
- MLE is a special case of MAP

Going back to our MLP

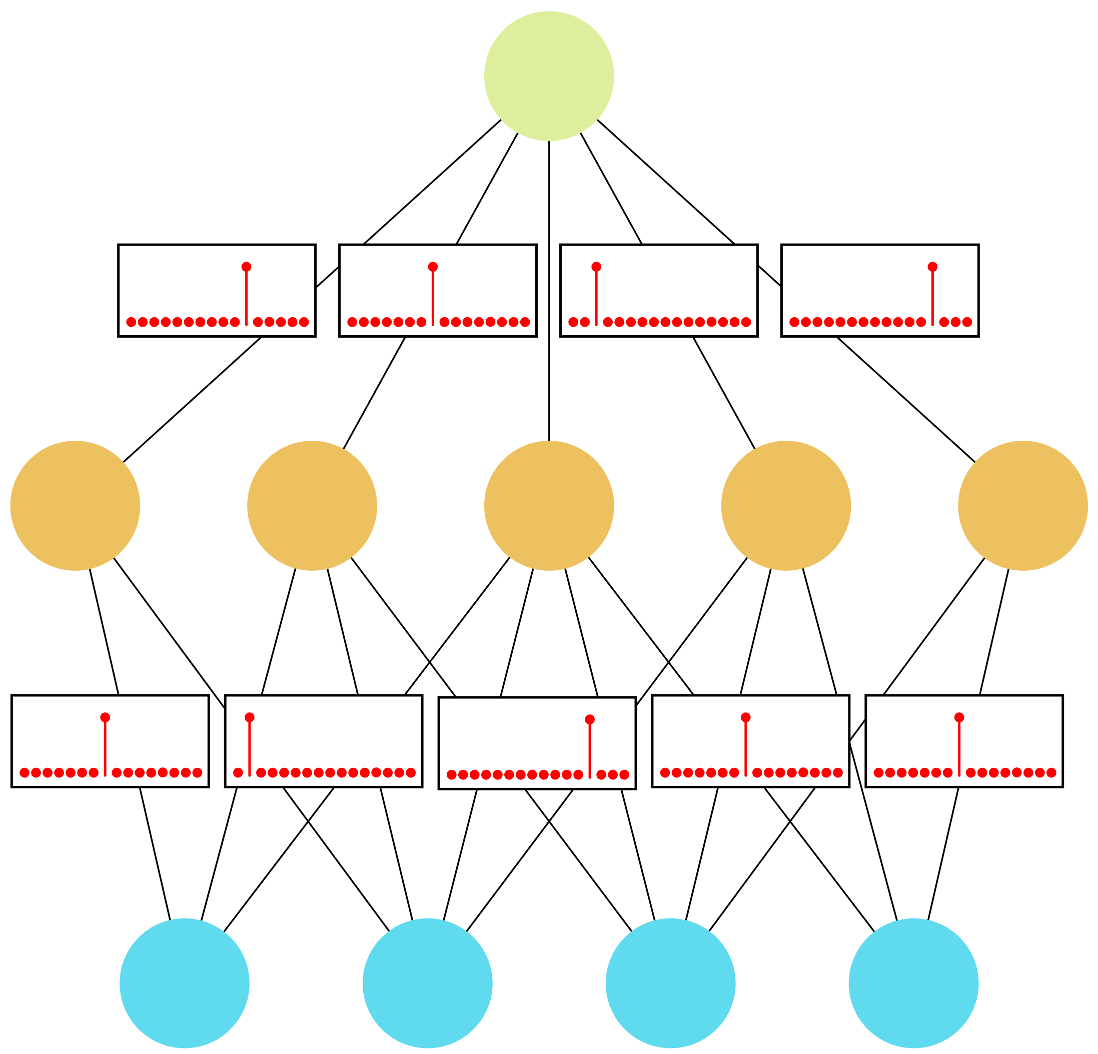
Multi-layer perceptron (MLP)
Feed forward network



- After convergence, standard MLP obtains a single value
- In effect, it converges to the Maximum Likelihood (MLE) value.
- There is no uncertainty on the output or the individual values in the network
- Does not capture epistemic uncertainty (uncertainty due to the model itself)

Two excellent papers: Jospin et al. (2022, arXiv: 2007.06823); Goan & Fookes (2020, arXiv: 2006.12024)

Going back to our MLP



$$y = l_n$$

$$l_i = s_i(W_i l_{i-1} \; + \; b_i)$$

$$l_0 = x,$$

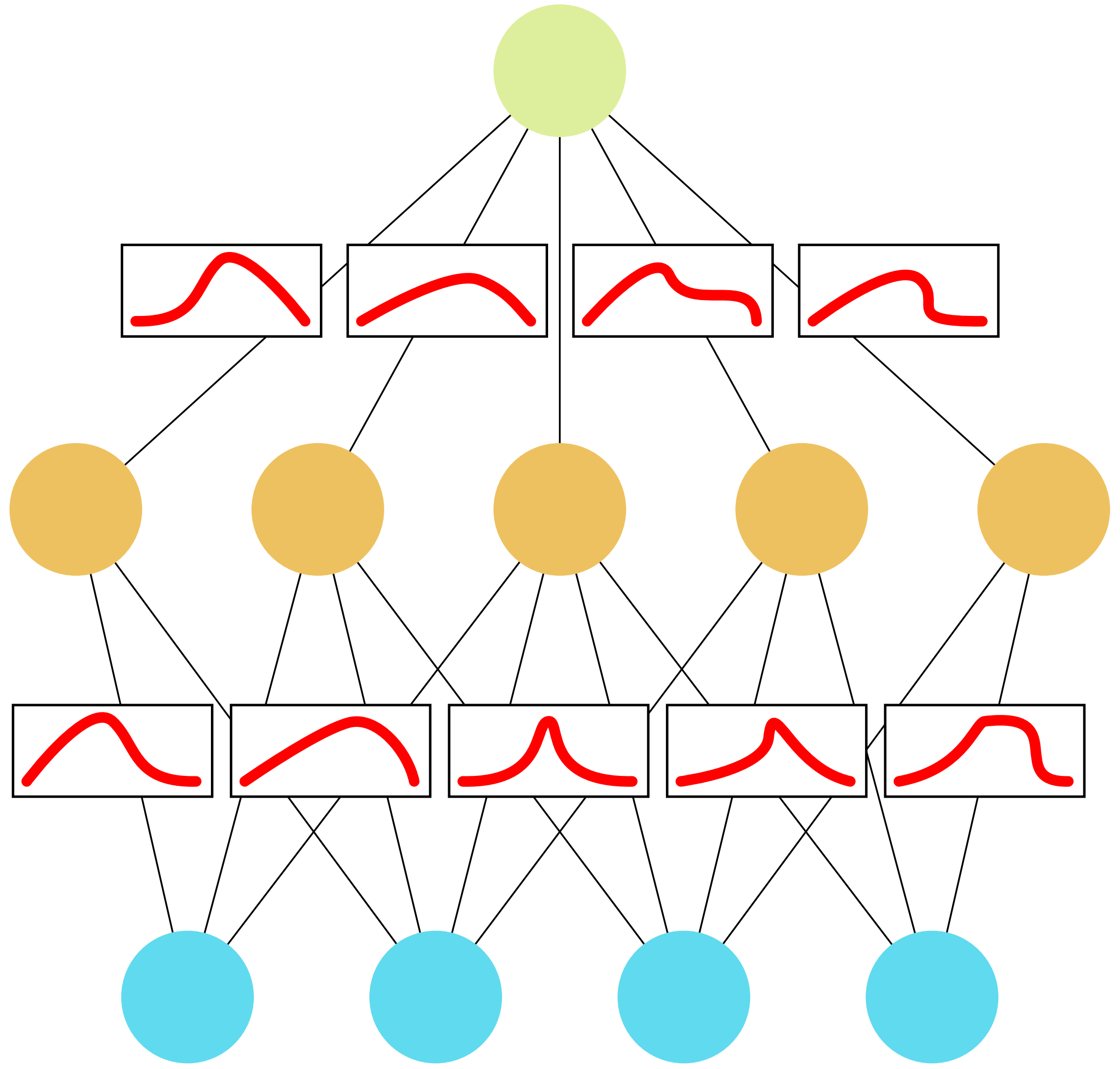
l = layer
x = input
y = output
W = weights matrix
b = biases
s = activation function

Let's collect all model parameters in theta:

$$\theta = (W, b)$$

Two excellent papers: Jospin et al. (2022, arXiv: 2007.06823); Goan & Fookes (2020, arXiv: 2006.12024)

Adding uncertainties to our weights



$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}),$$
$$\boldsymbol{y} = \Phi_{\boldsymbol{\theta}}(\boldsymbol{x}) \; + \; \boldsymbol{\epsilon},$$

Φ = our approximate model

ϵ = random noise

$\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b})$

D_x = training input data

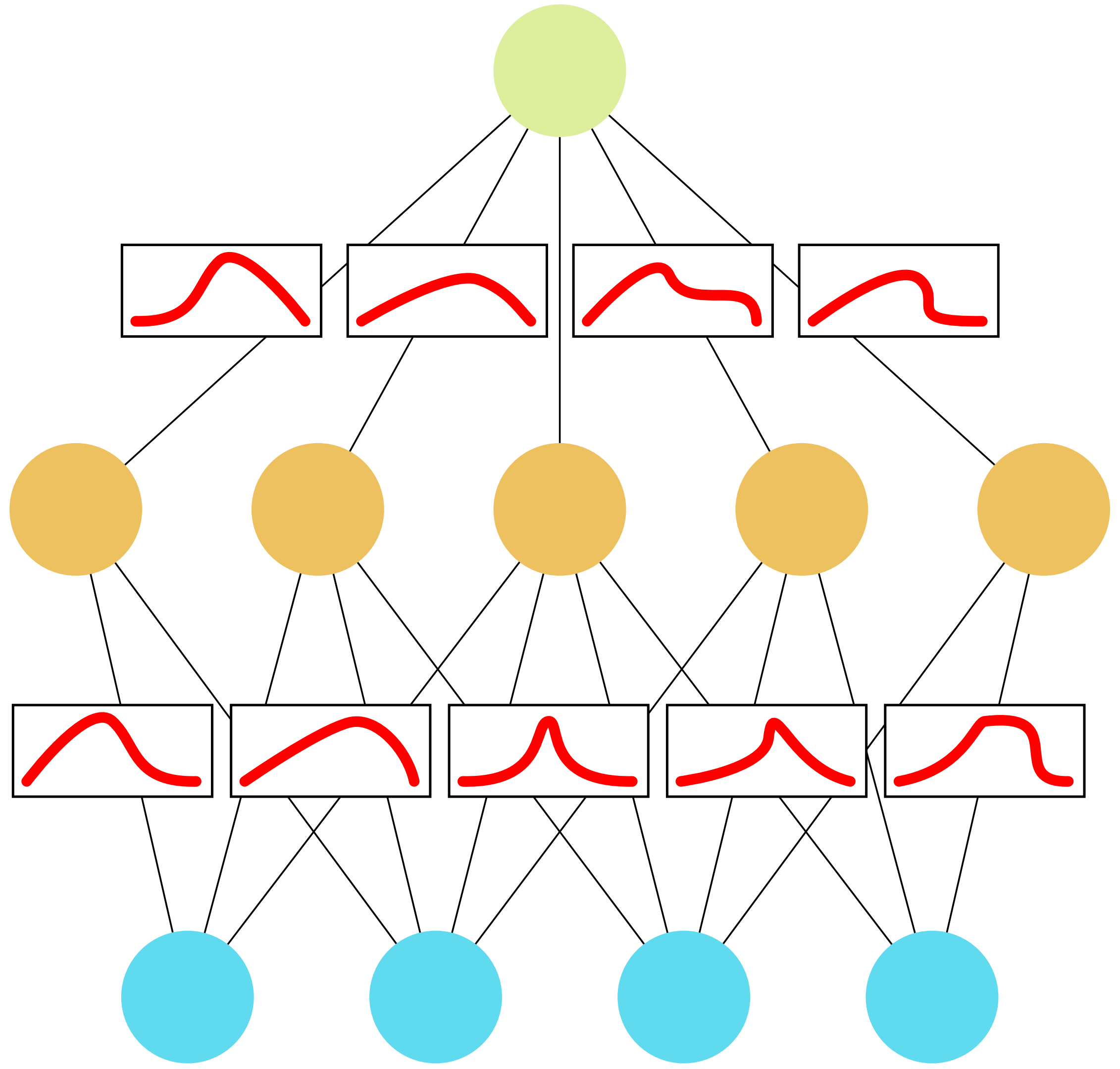
D_y = training output data

- By setting the weights to be distributions, we make the model probabilistic
- We can now compute the posterior of the parameters θ over the training data D

$$p(\boldsymbol{\theta}|D) = \frac{p(D_{\boldsymbol{y}}|D_{\boldsymbol{x}}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} p(D_{\boldsymbol{y}}|D_{\boldsymbol{x}}, \boldsymbol{\theta}')p(\boldsymbol{\theta}')d\boldsymbol{\theta}'} \propto p(D_{\boldsymbol{y}}|D_{\boldsymbol{x}}, \boldsymbol{\theta})p(\boldsymbol{\theta})$$

Jospin et al. (2022, arXiv: 2007.06823

Adding uncertainties to our weights



- Given $p(\theta \,|\, D)$ we can compute the probability of y given x assuming D: $p(y \,|\, x, D)$

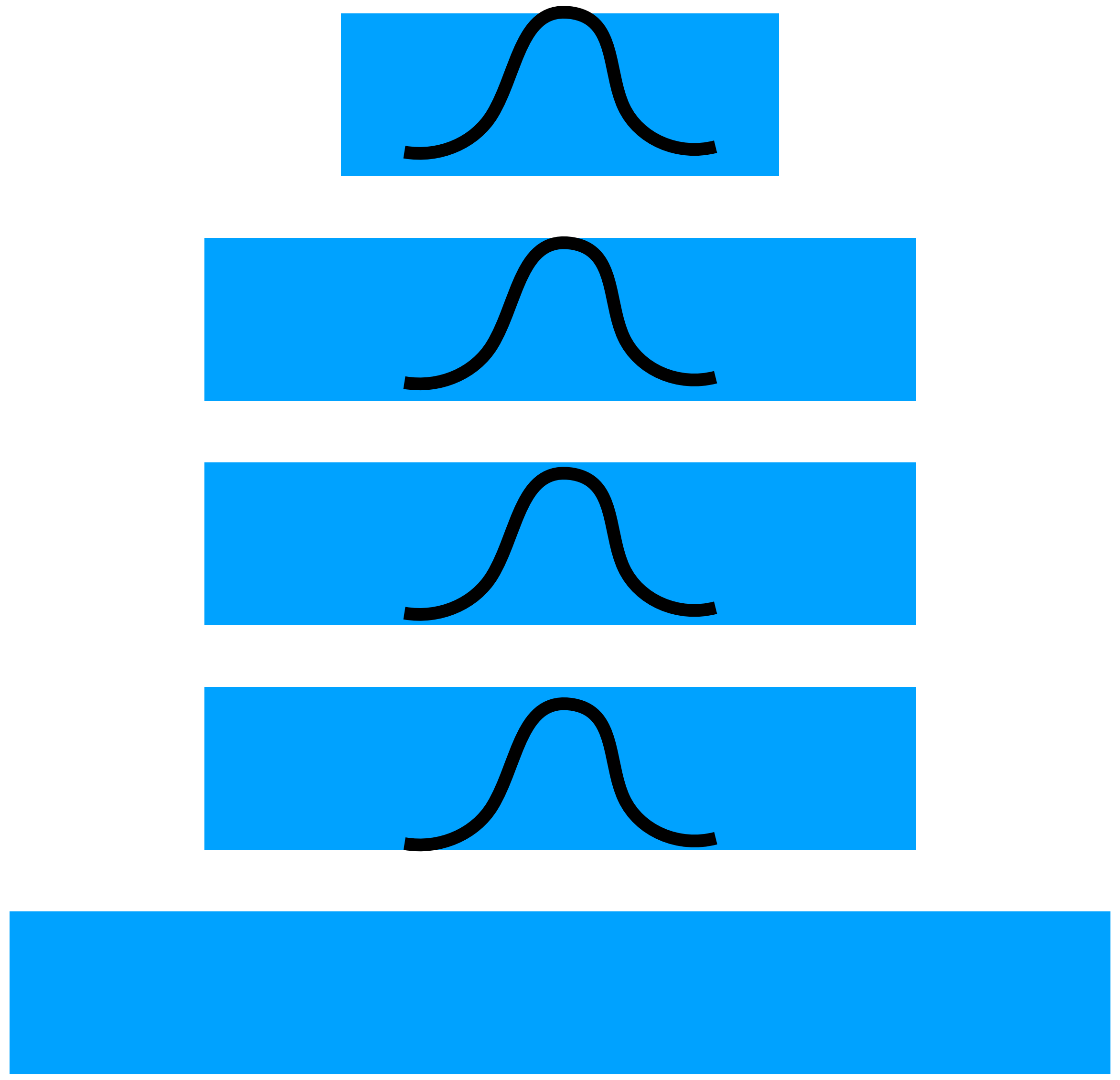
$$p(\boldsymbol{y}|\boldsymbol{x}, D) = \int_{\boldsymbol{\theta}} p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta'})p(\boldsymbol{\theta'}|D)d\boldsymbol{\theta'}$$

The integral $p(y \,|\, x, \theta)$ is very hard to calculate. It is usually sampled or approximated using Variational Inference (e.g. Normalising Flows)

We will discuss Variational Inference if we have time…

Jospin et al. (2022, arXiv: 2007.06823

What’s happening in practice



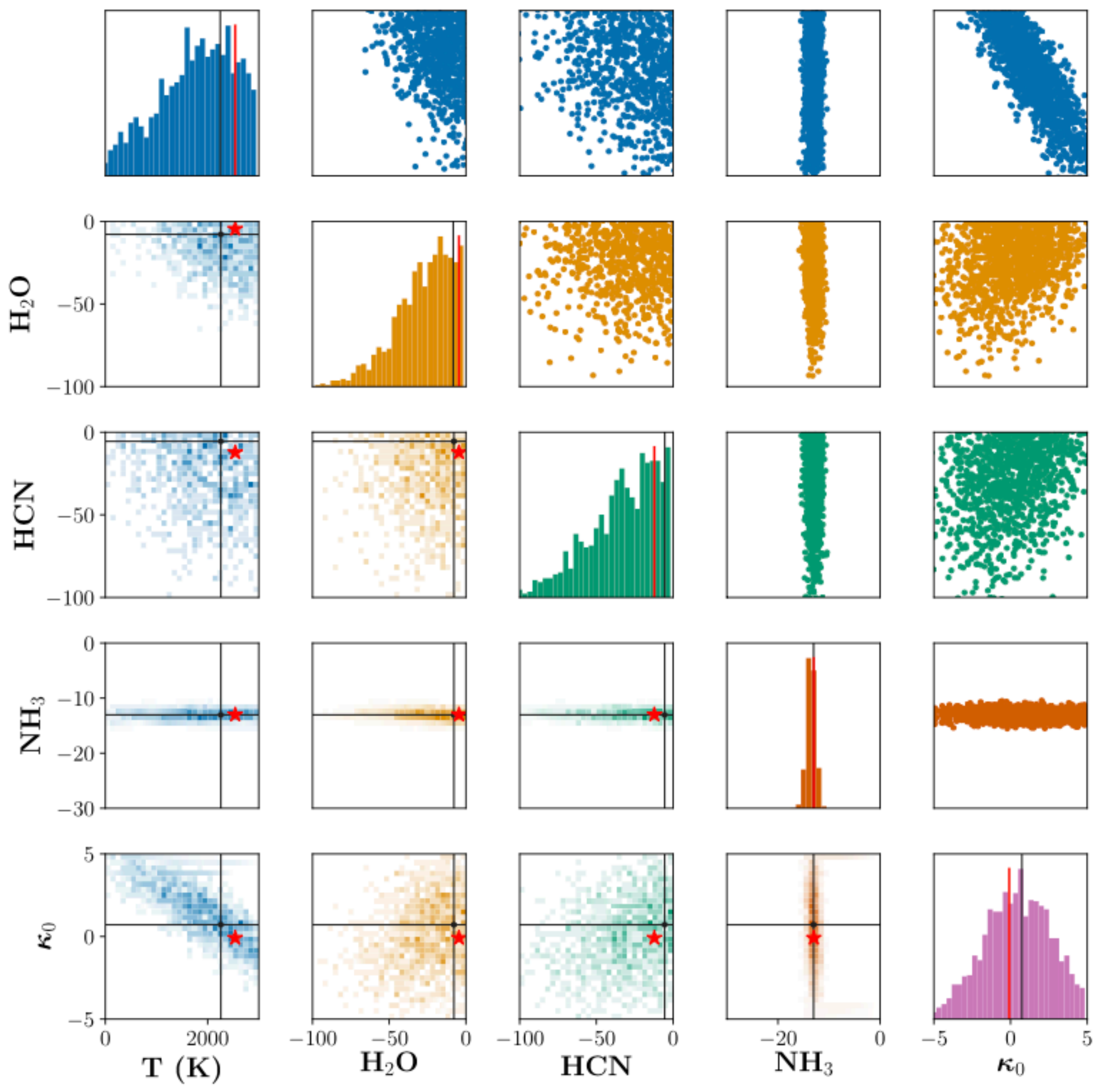
- Replace your standard feed forward layers with probabilistic layers
- The easiest way is to use PyRo or torchnbnn that implements this for you
- We can now compute the posterior of the parameters θ over the training data
- We don’t actually need ALL layers to be probabilistic but only the last layer needs to be

Google Colab notebook:
https://bit.ly/ExoAI_BNN

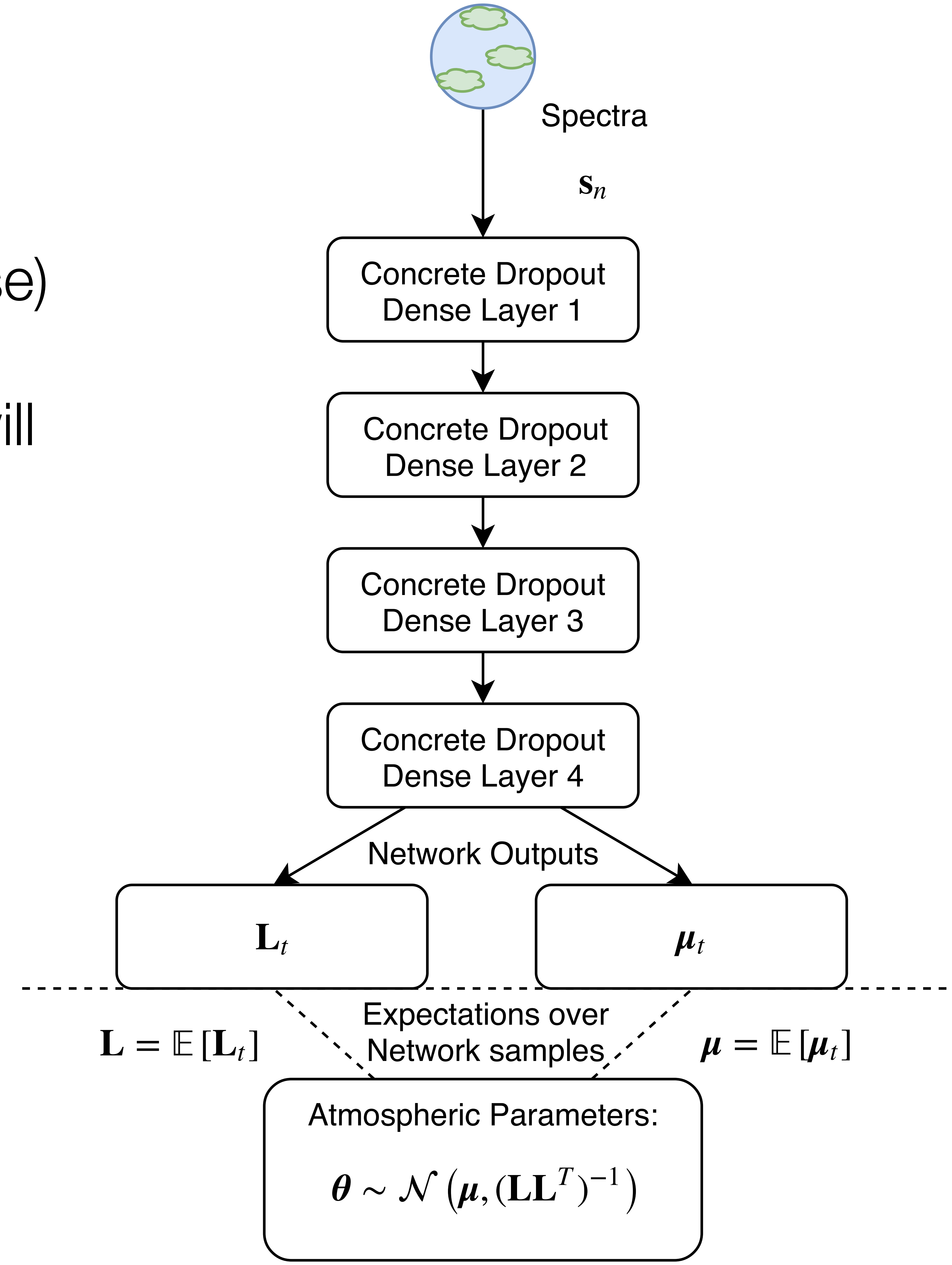
Ensemble Bayesian Networks

- Provides an estimate of network uncertainty (epistemic noise)

- Running many networks in an ensemble (average results) will create a stronger predictor



(b) plan-net Ensemble



Plan-net: Cobb et al. (2019)

Do we have more time?

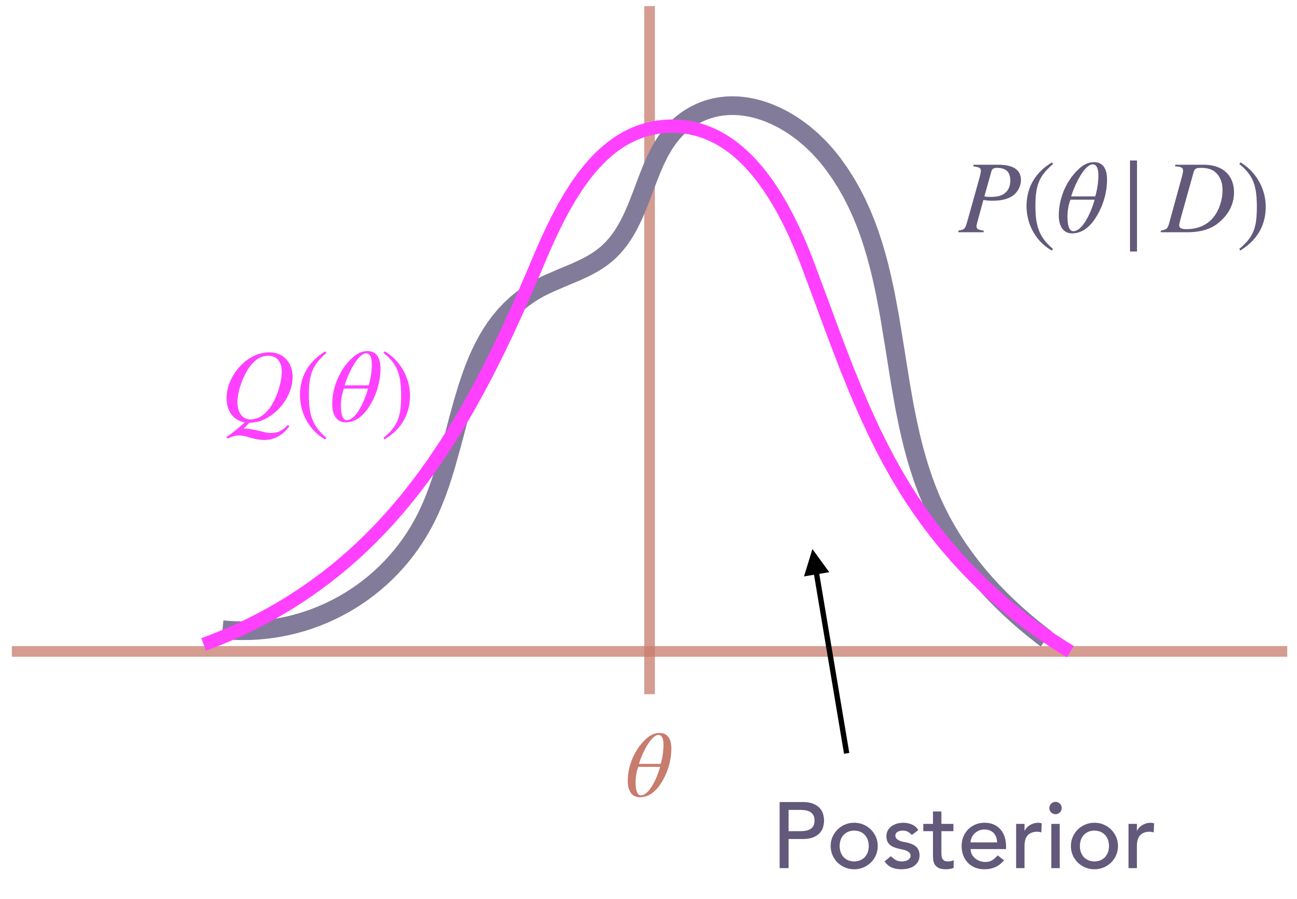
/imagine prompt: a surrealist painting showing the flow of time with astronomers

Variational Inference

- If $P(\theta | D)$ is intractable, don't sample from it, replace it with an approximation

$$P(\theta | D) = \frac{P(D | \theta)P(\theta)}{P(D)} \qquad P(D) = \int P(D | \theta)P(\theta)d\theta$$

- Instead of sampling an intractable posterior, we can replace it with an approximate distribution $Q(\theta)$
- The idea is to minimise the statistical difference between $Q(x)$ and $P(\theta | D)$ -> This becomes a fitting, not a sampling problem!
- $Q(x)$ can be any function but often is a multivariate Gaussian

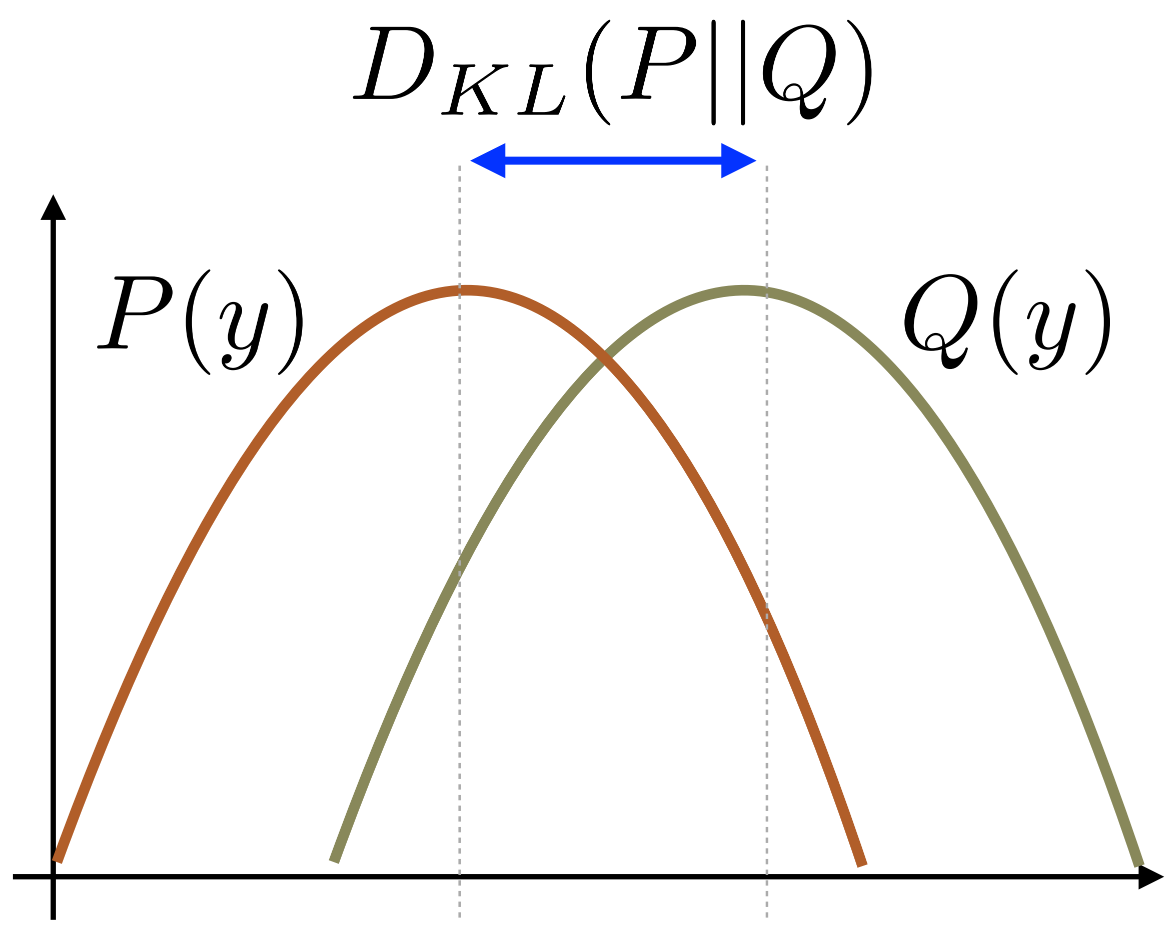


Bei et al. 2018, Ganguly & Earp 2021

Reminder: Kullback-Leibler Divergence

- Claude Shannon derived information entropy in 1948
- Derived by Salomon Kullback and Richard Leiber in 1951
- KLD is the most fundamental measure of information theory
- KLD was devised to measure the expected extra information needed if you want to model the right distribution, P, but you assume the wrong distribution Q.
- It measures the 'distance' between two probability distributions
- Note that $D_{KL}(P||Q) \neq D_{KL}(Q||P)$!!

$$D_{KL}(P||Q) = \int P(y) \log \left(\frac{P(y)}{Q(y)} \right) dy$$



Berger et al (2009), Kullback & Leibler (1951), John & Draper (1975)

Variational Inference

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

$$P(D) = \int P(D|\theta)P(\theta)d\theta$$

$$P(\theta, D) = P(\theta|D)P(\theta)$$

- VI poses the following minimisation:

$$Q^*(\theta) = \underset{Q(\theta)\in\mathbb{Q}}{\text{argmin}}\ D_{KL}(Q(\theta)\,||\,P(\theta|D))$$

- VI poses the following minimisation:

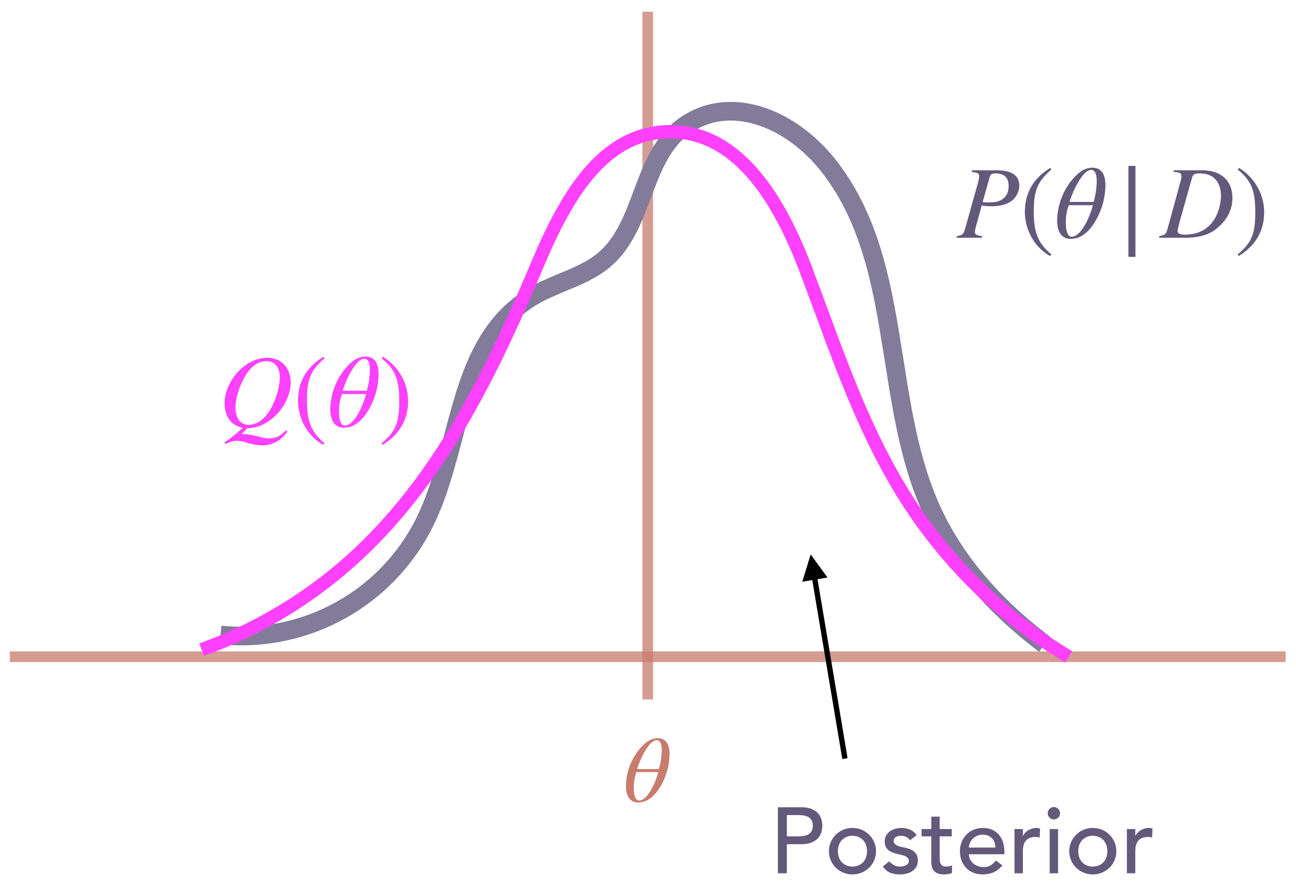
$$D_{KL}(Q(\theta)\ \|\ P(\theta|D)) = \mathbb{E}_Q\left[\log\frac{Q(\theta)}{P(D|\theta)}\right]$$

$$= \mathbb{E}_Q\left[\log Q(\theta)\right] - \mathbb{E}_Q\left[\log P(\theta|D)\right]$$

$$= \mathbb{E}_Q\left[\log Q(\theta)\right] - \mathbb{E}_Q\left[\log P(\theta, D)\right] + P(D)$$

$$= \underbrace{-\left(\mathbb{E}_Q\left[\log P(\theta, D)\right] - \mathbb{E}_Q\left[\log Q(\theta)\right]\right)}_{\text{ELBO}} + P(D)$$

Hard to compute,
Easy to ignore…



Berger et al (2009), Kullback & Leibler (1951), John & Draper (1975)

Variational Inference

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

$$P(\theta, D) = P(\theta|D)P(\theta)$$

- If you continue the maths you get

$$D_{KL}(Q(\theta) \,||\, P(\theta|D)) = \mathbb{E}_Q\left[\log \frac{Q(\theta)}{P(D|\theta)}\right]$$

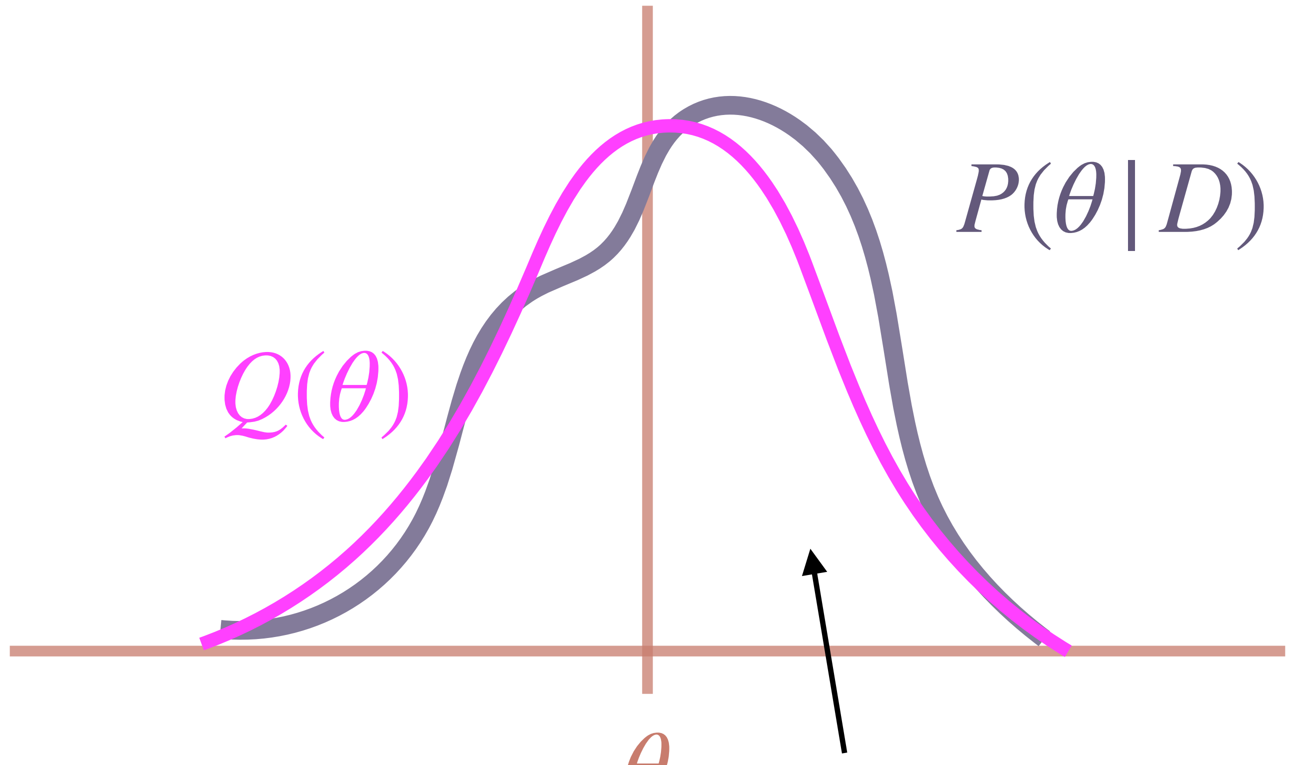
$$= -\underbrace{\left(\mathbb{E}_Q\left[\log\, P(\theta, D)\right] - \mathbb{E}_Q\left[\log\, Q(\theta)\right]\right)}_{\text{ELBO}} + P(D)$$

$$\text{ELBO}(Q) = \mathbb{E}[\log P(\theta)] + \mathbb{E}[\log P(D|\theta)] - \mathbb{E}[\log Q(\theta)]$$

$$= \mathbb{E}[\log P(D|\theta)] - D_{KL}(Q(\theta)\,||\,P(\theta))$$

Expectation of your likelihood

Distance of $Q(\theta)$ from Prior $P(\theta)$

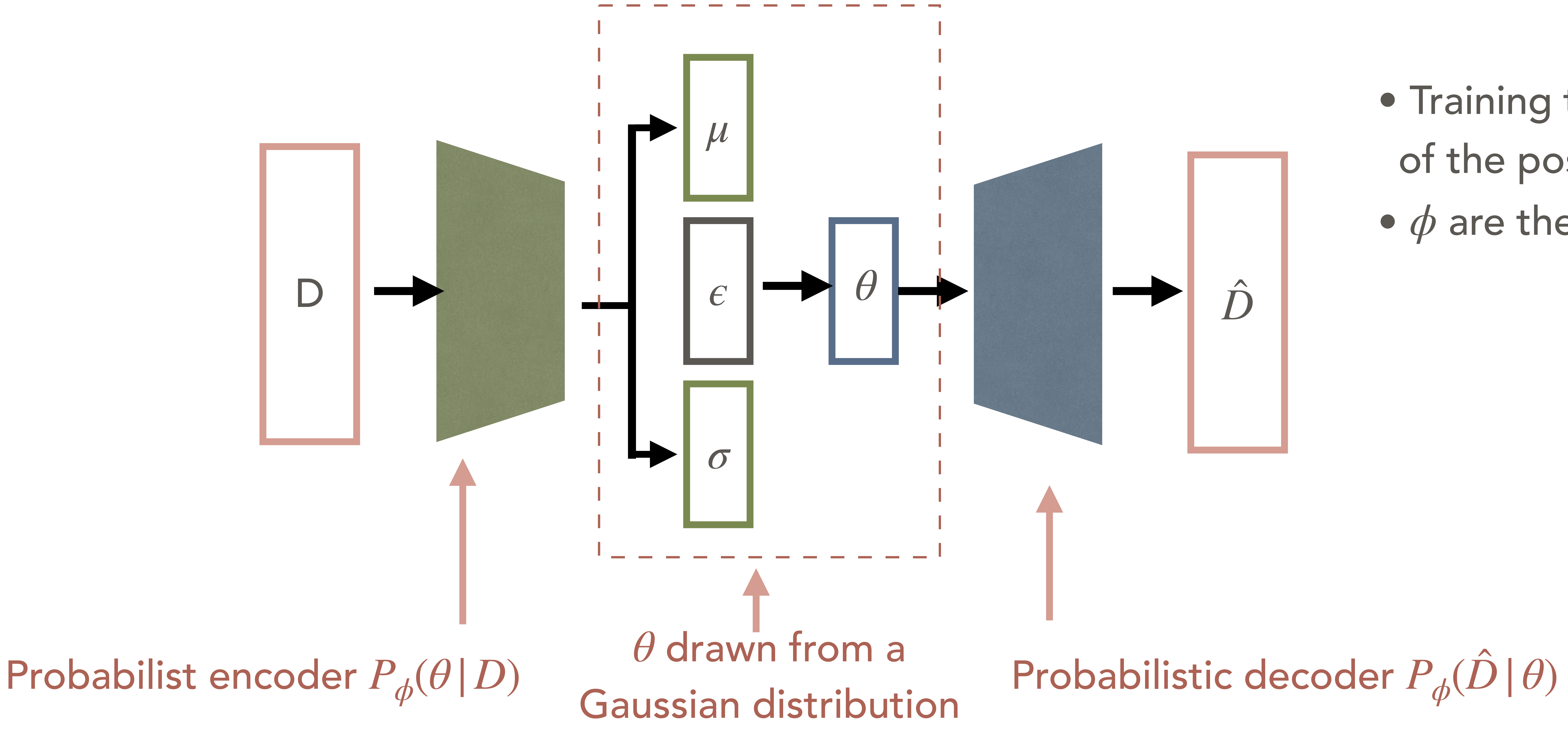


Blei et al. 2018

Variational Inference in Variational Autoencoders

- How do we calculate this? Using ML
- Variational Autoencoders (VAE) use VI very successfully

$$\mathrm{ELBO}(Q) = \mathbb{E}[\log P(\theta)] + \mathbb{E}[\log P(D|\theta)] - \mathbb{E}[\log Q(\theta)]$$
$$= \mathbb{E}[\log P(D|\theta)] - D_{KL}(Q(\theta)||P(\theta))$$

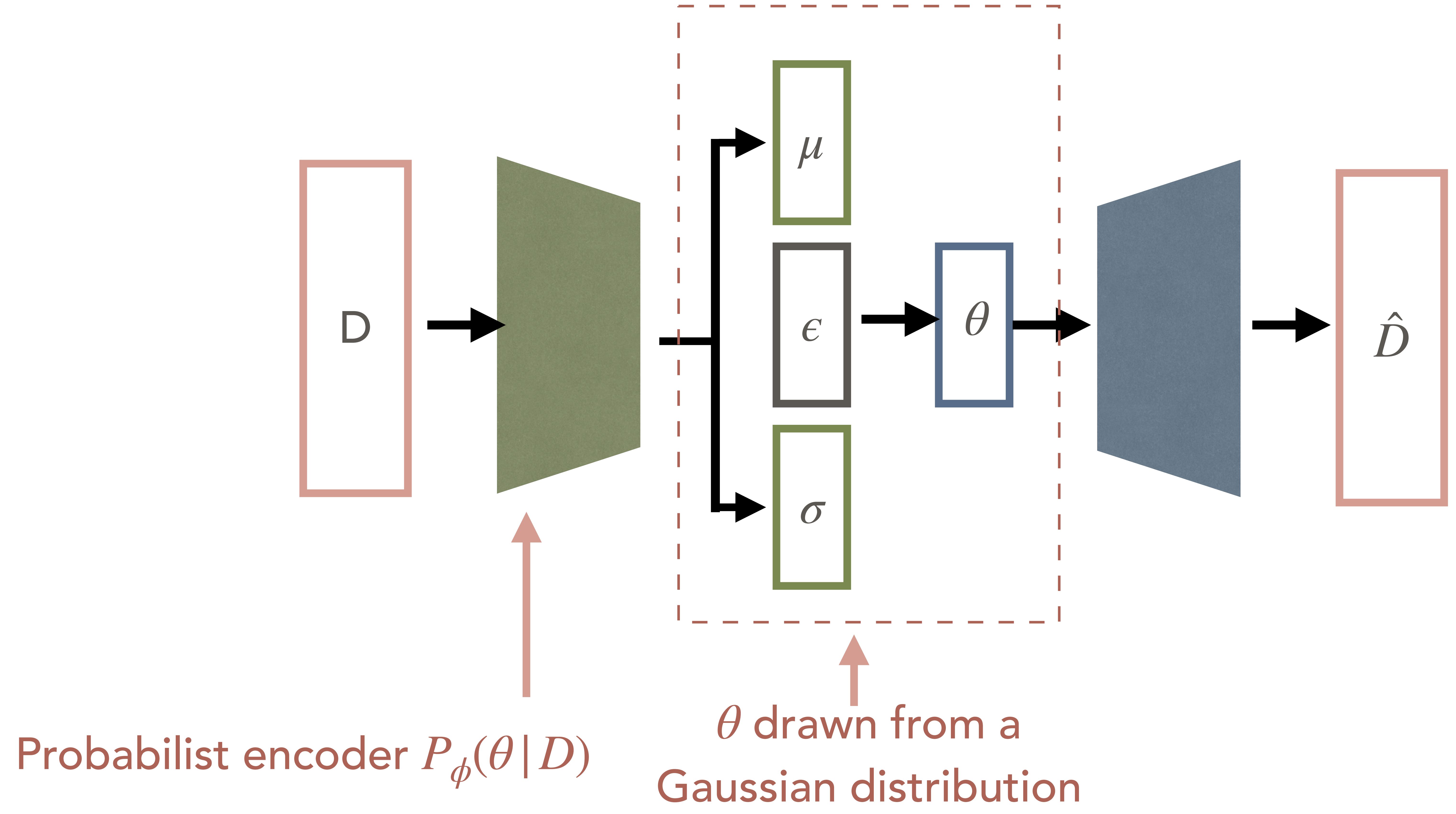


- Training the VAE learns an approx. of the posterior $P_\phi(\theta|D)$
- ϕ are the parameters of the VAE

Blei et al. 2018

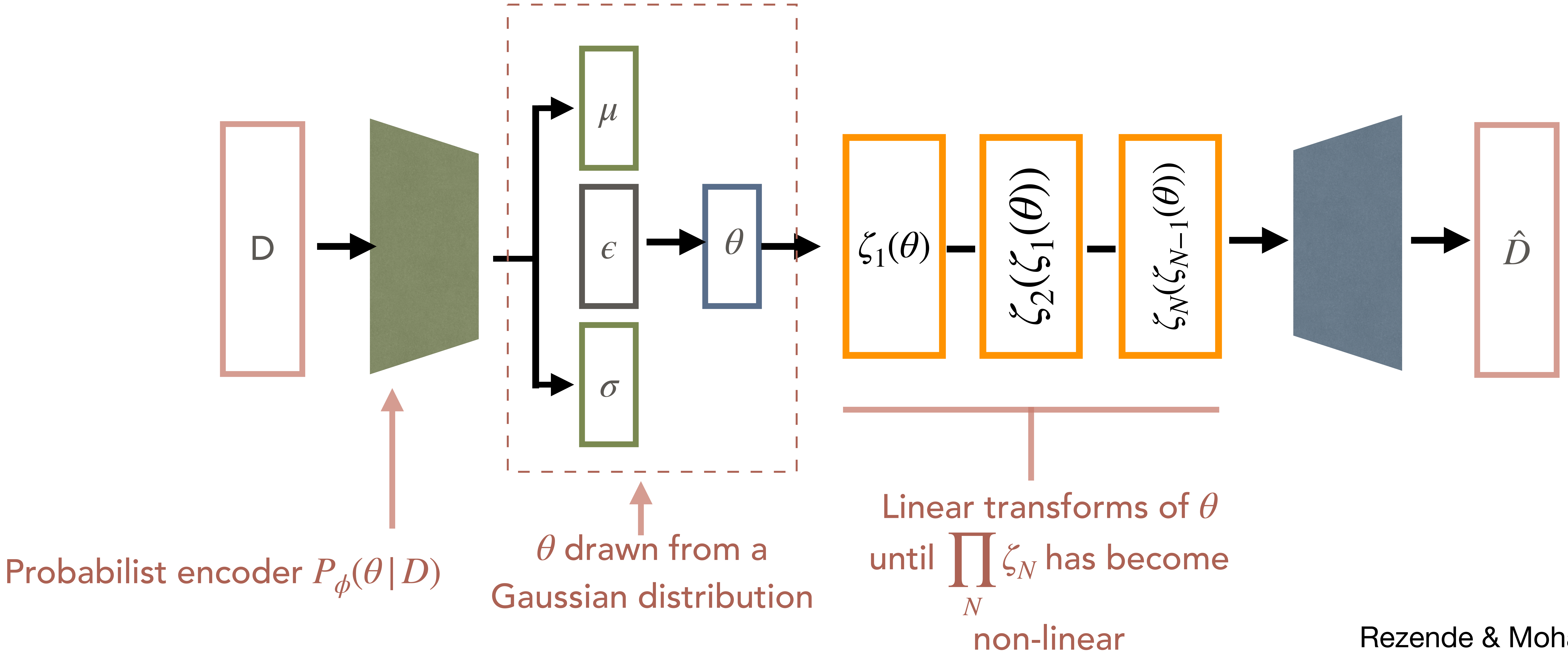
Normalising flows. Making VI non-Gaussian

- Normalising flows extend the central Gaussian assumption to arbitrary complex distributions
- The do this by repeatedly learning consecutive linear transformations of θ



Normalising flows. Making VI non-Gaussian

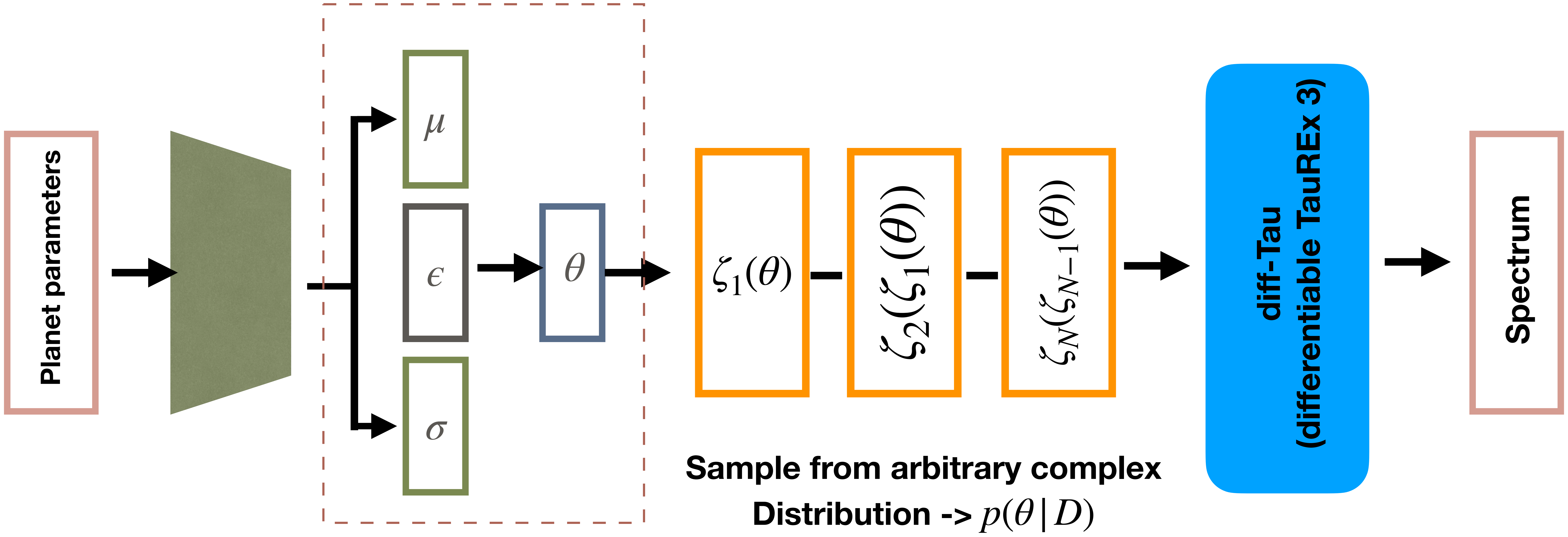
- Normalising flows extend the central Gaussian assumption to arbitrary complex distributions
- The do this by repeatedly learning consecutive linear transformations of θ



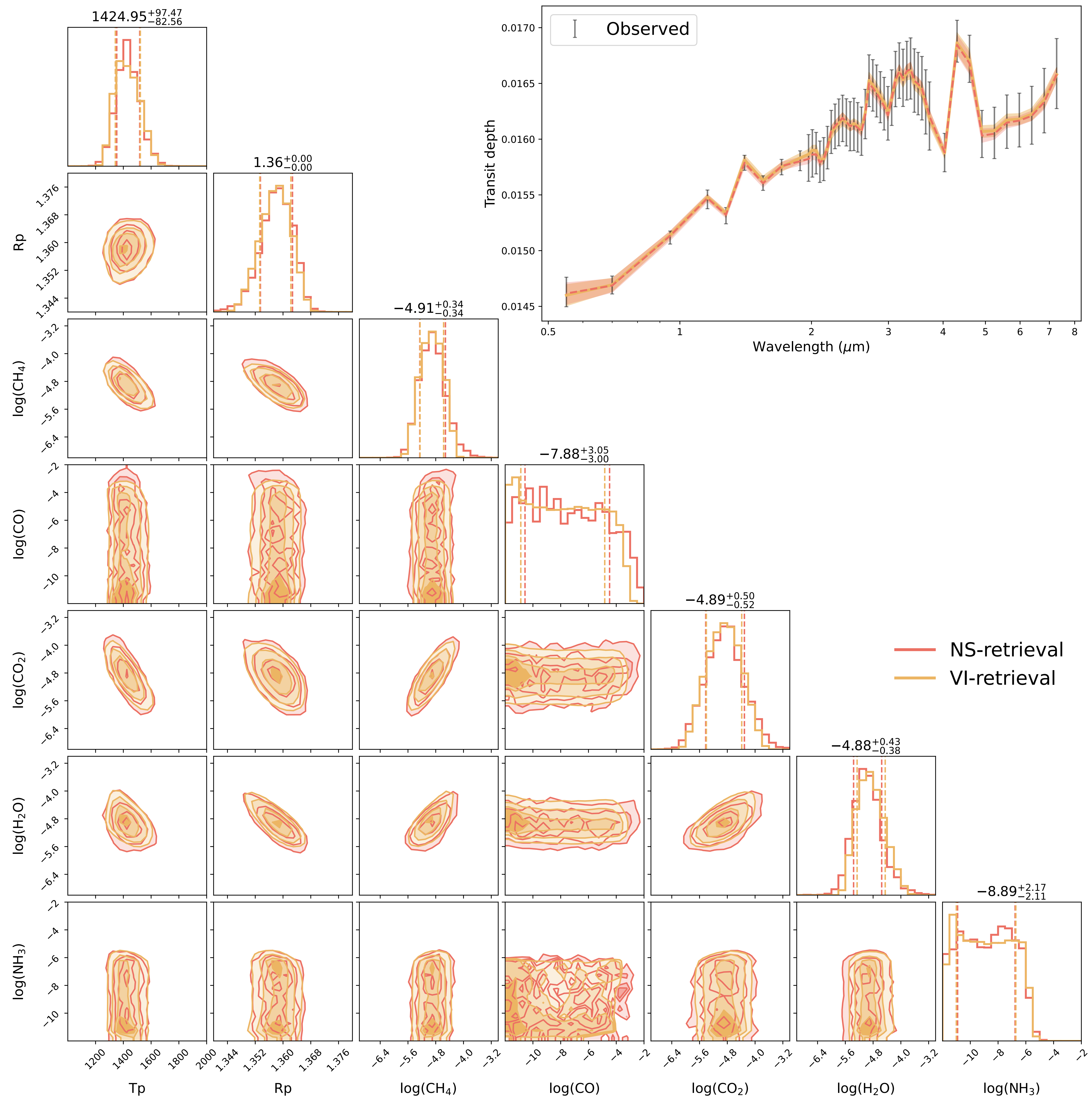
Rezende & Mohamad (2015)

Normalising flows learning Atmospheric retrievals

- Normalising flows extend the central Gaussian assumption to arbitrary complex distributions
- The do this by repeatedly learning consecutive linear transformations of θ



Yip et al (2022)



Variation Inference vs Sampling results

- Equivalent posteriors to traditional retrievals
- 75% fewer forward models required
- Full formal treatment of observational errors
- Full ability to do Bayesian model selection

Model	ELBO	Ref	$\log_{10}(\mathcal{B})$
Flat line	62.74	62.83	315.66
No Methane	345.37	347.18	33.03
Complete	378.40	380.20	N/A
Overspecified Model	374.00	377.74	4.4

Yip et al. (2022)

Variational Inference in Variational Autoencoders

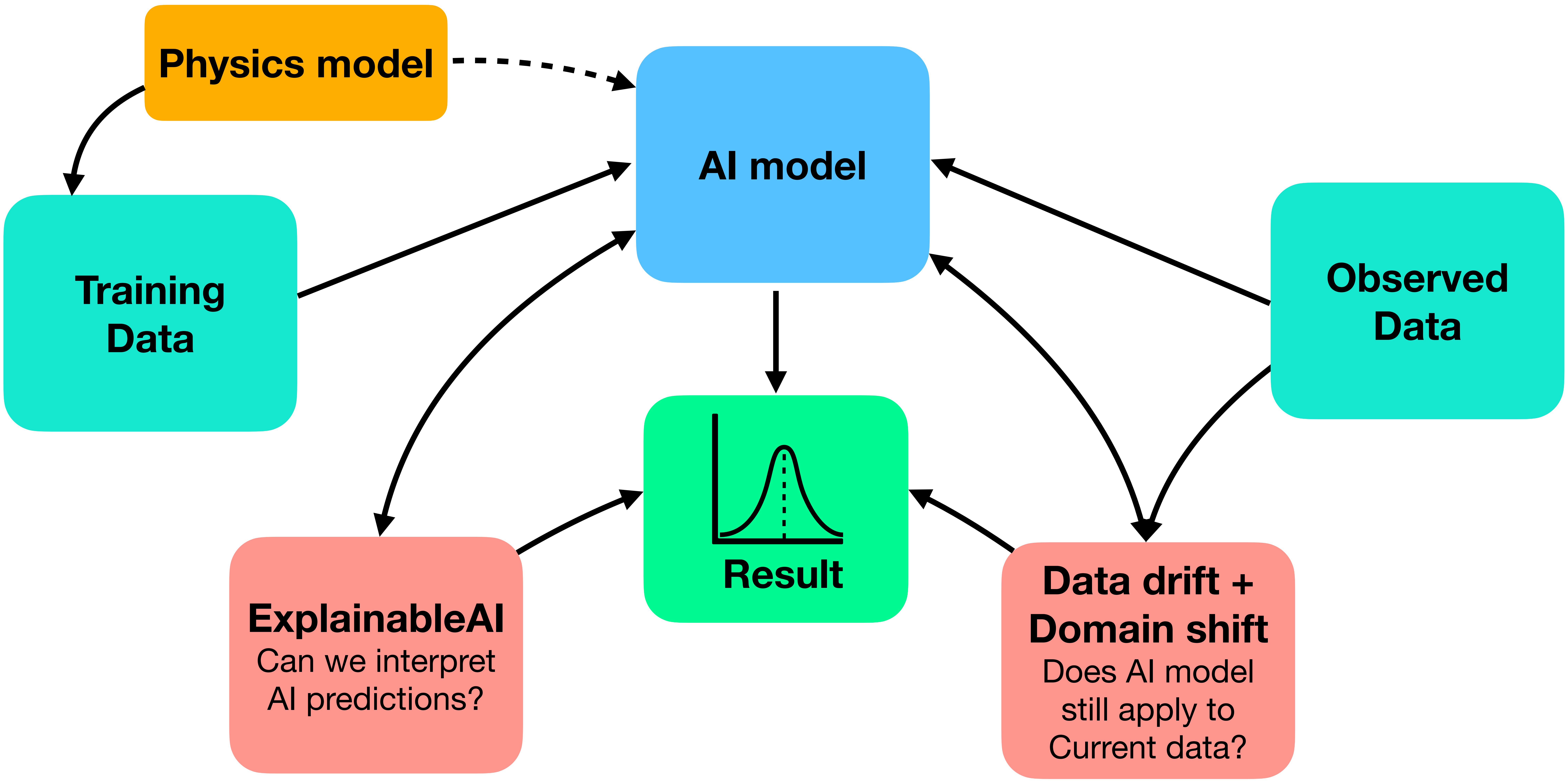
- VAEs are not the only way to do VI but its the most focused on at the moment
- Most distributions used to approximate $P(\theta | D)$ are multivariate Gaussians but more complex distributions can be implemented or iteratively learned
- Normalising Flows allow the transformation from Gaussians to arbitrary complex distributions by iteratively applying linear transformations to the Gaussian dists.
- Good blogpost on NFs: https://towardsdatascience.com/introduction-to-normalizing-flows-d002af262a4b

Kobyzev et al. 2019

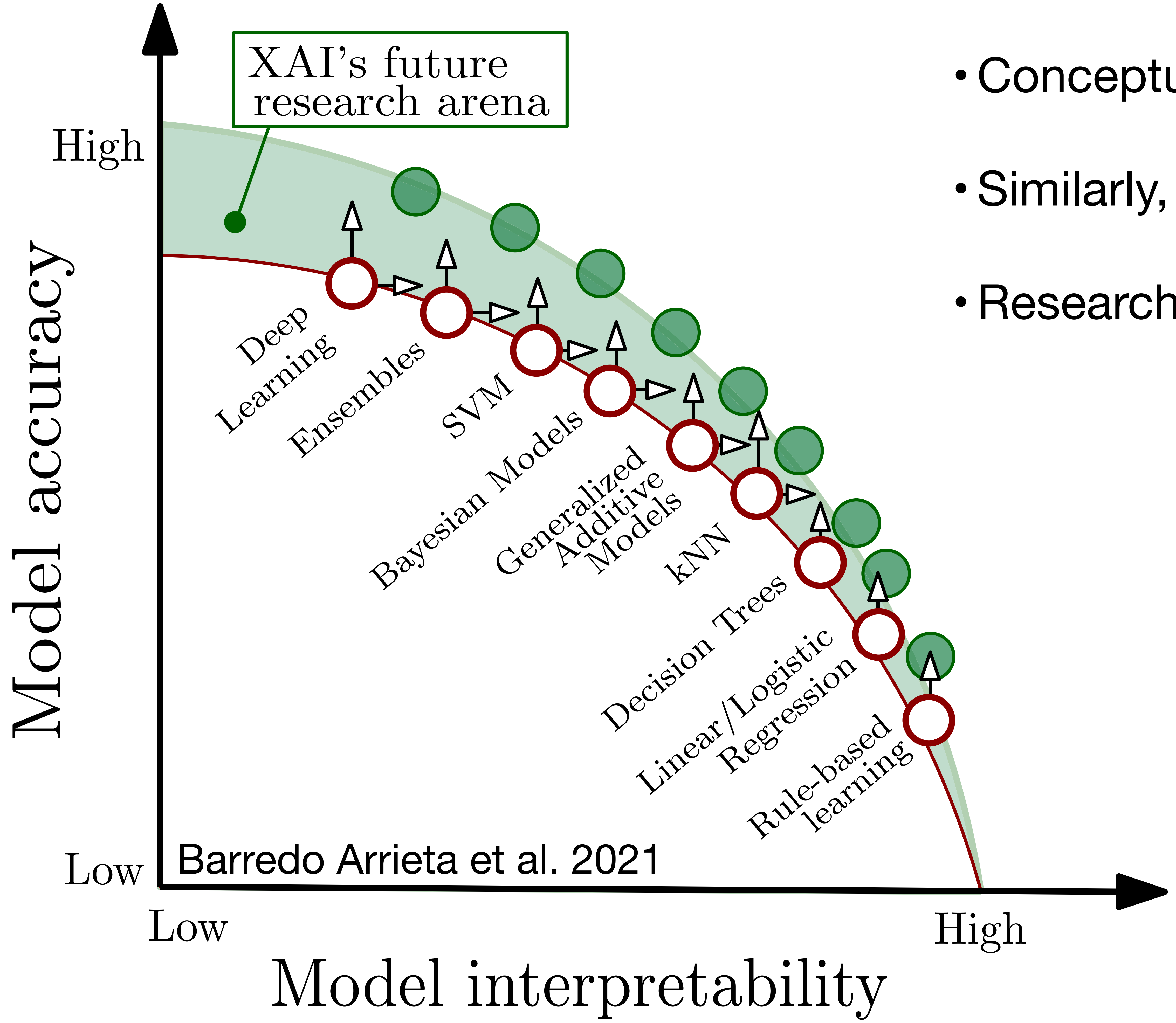
The need for explainability

/imagine prompt: An impression of the pitfalls and dangers of using AI

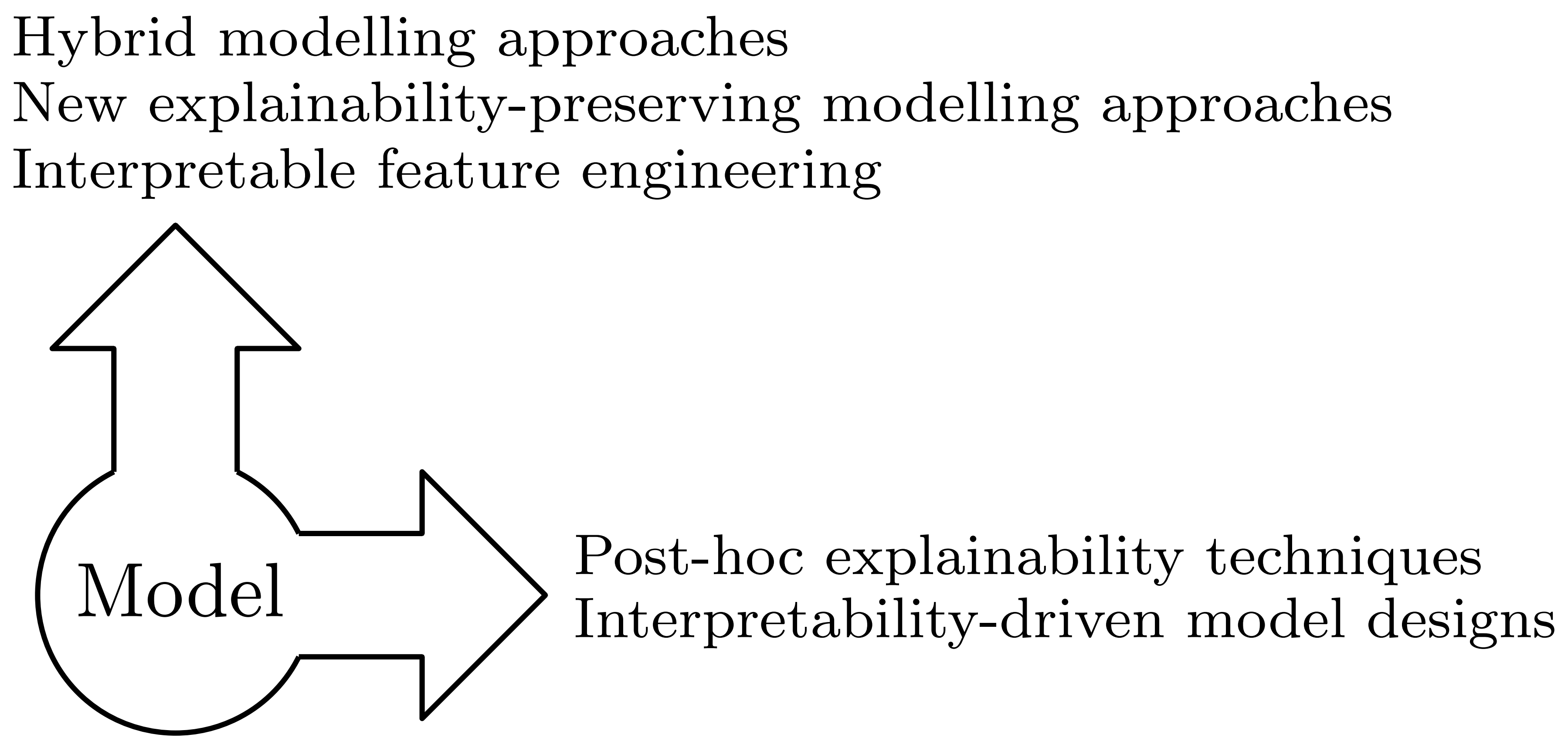
It’s not only about publishing a paper…



Power vs Explainability

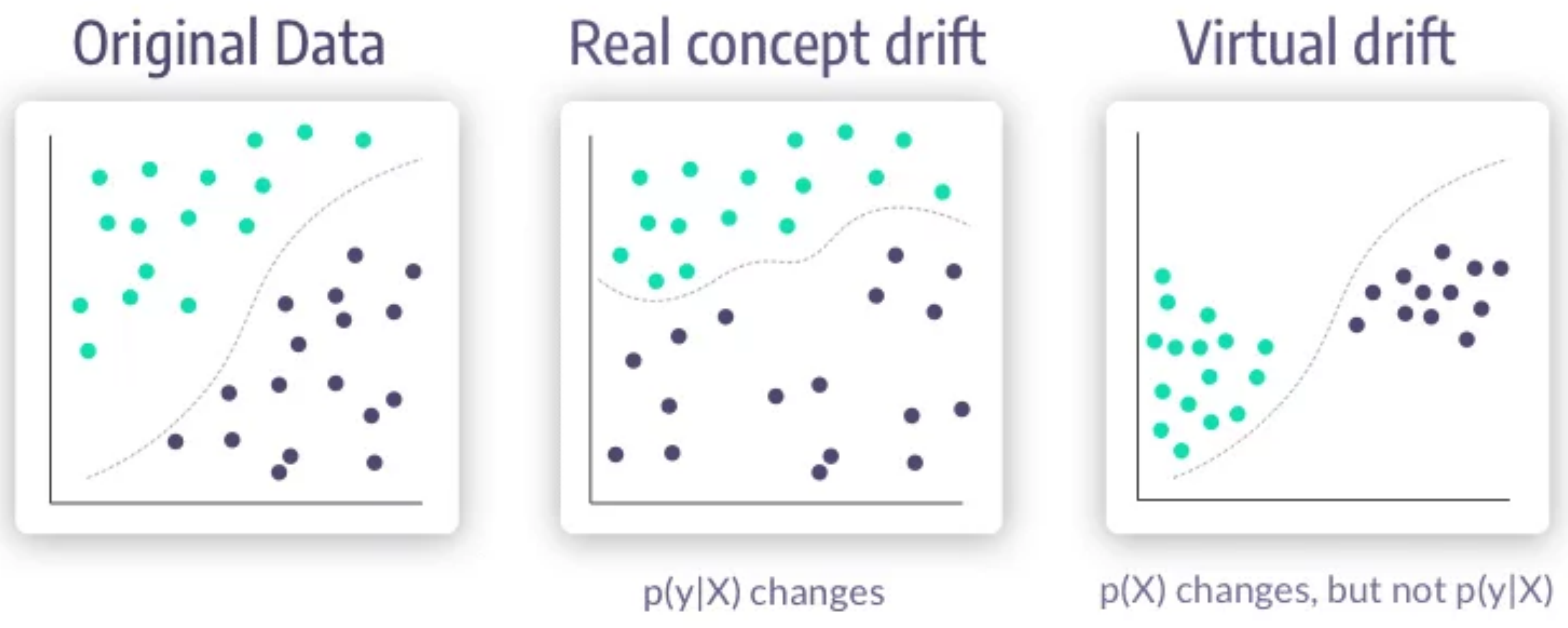


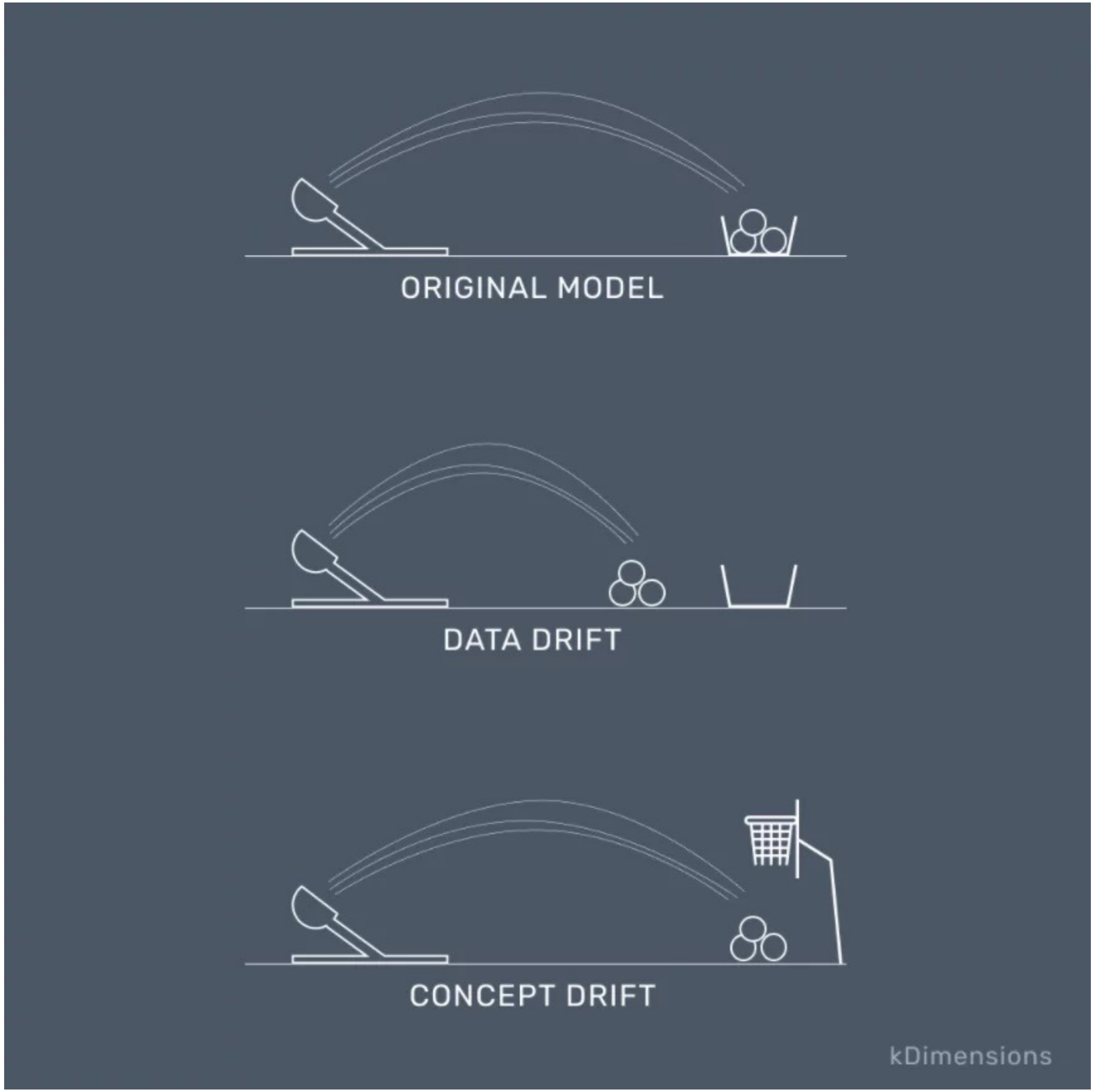
- Conceptually, the more complex the model the harder to explain
- Similarly, the more complex the model, the more expressive
- Researcher needs to weigh up interpretability vs accuracy



Concept and Data drift

- Is your model trained on simulations? Are those representative of the data?
- Is your observation/instrument changing?
- Is your data changing in imperceptible ways?
- Is the science question changing?





Hierarchy of Explainability

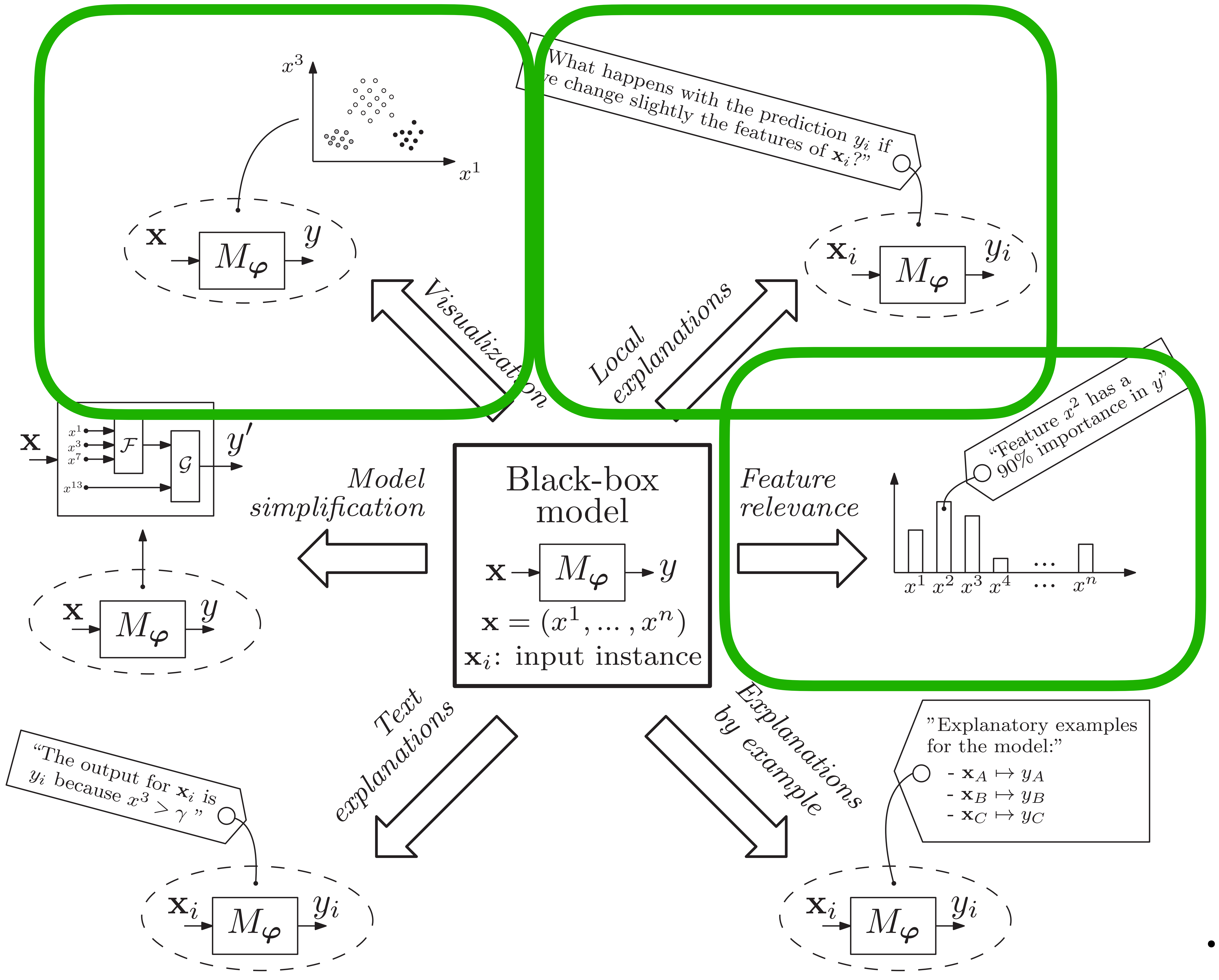
Passive vs Active, Local vs Global Explanations

Dimension 1 — Passive vs. Active Approaches	
Passive	Post hoc explain trained neural networks
Active	Actively change the network architecture or training process for better interpretability

Dimension 2 — Type of Explanations (in the order of increasing explanatory power)	
To explain a prediction/class by	
Examples	Provide example(s) which may be considered similar or as prototype(s)
Attribution	Assign credit (or blame) to the input features (e.g. feature importance, saliency masks)
Hidden semantics	Make sense of certain hidden neurons/layers
Rules	Extract logic rules (e.g. decision trees, rule sets and other rule formats)

Dimension 3 — Local vs. Global Interpretability (in terms of the input space)	
Local	Explain network's *predictions on individual samples* (e.g. a saliency mask for an input image)
Semi-local	In between, for example, explain a group of similar inputs together
Global	Explain the network *as a whole* (e.g. a set of rules/a decision tree)

Zhang et al. 2021



Barredo Arrieta et al. 2020, Information Fusion, 58, 82

A number of approaches

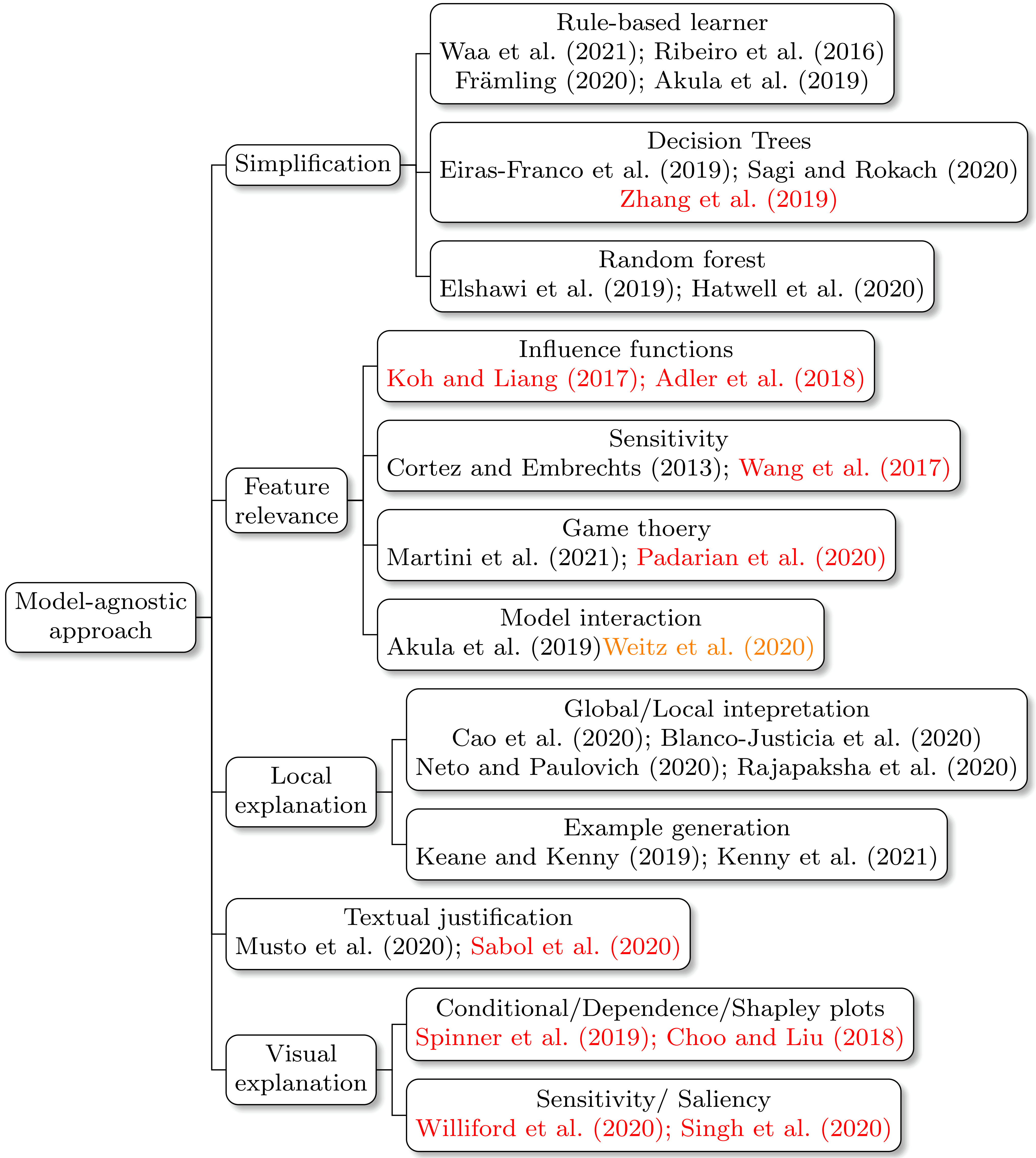
- Very fast developing field
- Large number of approaches
- Huge body of literature, see references below for good reviews

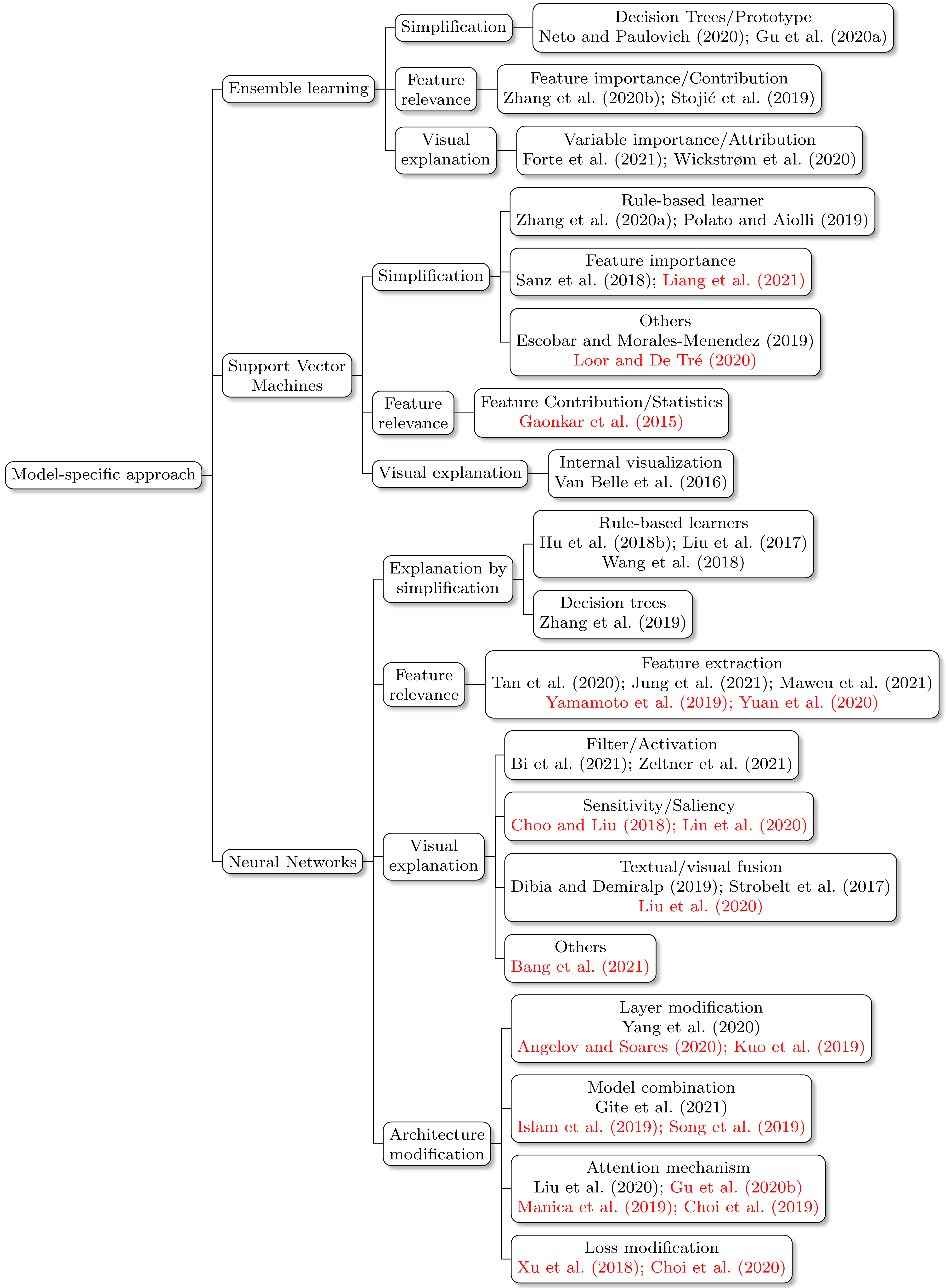
Most recent review papers

- Miller 2019
- Guidotti et al. 2019
- Carvaho et al 2019
- Guo 2020
- Tjoa & Guan 2020
- Meske et al 2020
- Arietta et al 2020
- Ivanovs et al 2021
- Langer et al 2021
- Sokol & Flach 2021
- Zhang et al 2021

- See Minh et al 2021 (Artificial Intelligence Review) for a review of review papers

Many approaches to XAI!





Barredo Arrieta et al. 2020, Minh et al. 2021

Localised explainability may sometimes not be enough

The Blind Man and the Elephant parable

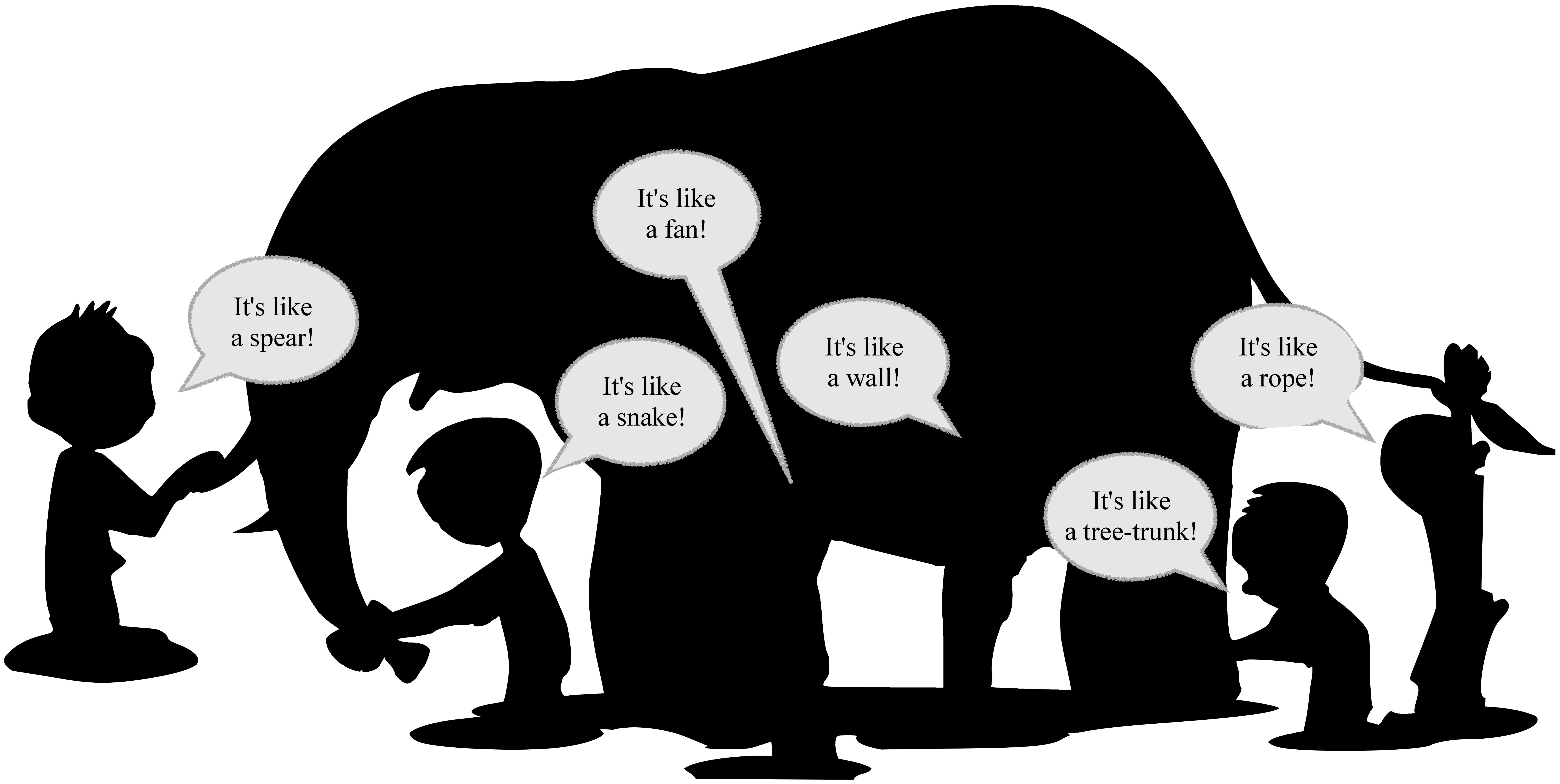


Figure from Sokol & Flach 2021, arXiv: 2112.14466

But if you have to use AI/ML
A quick cheat sheet:

- PCA, clustering and component separation, Random Forests…

    Use sklearn (https://scikit-learn.org/stable/index.html)

- Deep learning

    Use PyTorch (https://pytorch.org/)

- Probabilistic programming

    Use PyRo (https://pyro.ai/)

- Simulation based inference

    Use SBI (https://www.mackelab.org/sbi/)

- Great resources for models and tutorials

    HuggingFace (https://huggingface.co/)
    Papers With Code (https://paperswithcode.com/

Want to try yourself on some AI now?

Have a look at the Ariel Machine Learning Data Challenge

/imagine prompt: A female scientist analysing an alien in the war of the worlds

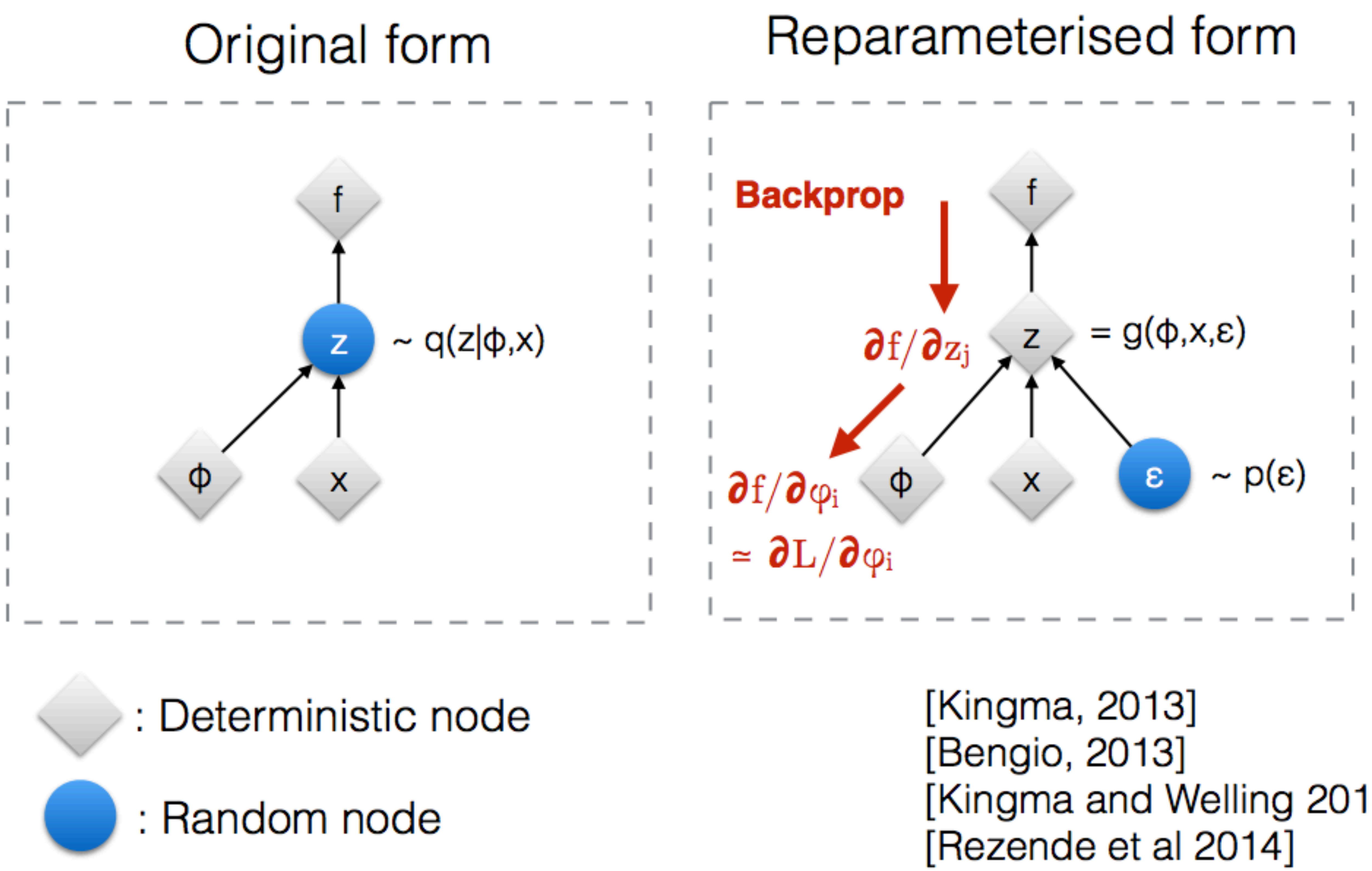
Done!

Any questions?

/imagine prompt: a hyper realistic photo of a group of students cheering that the boring lecture is finally over

Extra slides

Reparameterisation trick for VAEs



[Kingma, 2013]
[Bengio, 2013]
[Kingma and Welling 2014]
[Rezende et al 2014]