

Machine Learning Lecture - I

Sanmay Ganguly
Indian Institute of Technology, Kanpur
sanmay@iitk.ac.in

22/07/2024



30-th Vietnam School of Physics : Particles and Dark Matter



Plan for the lectures

Lecture 1

1. Why do ML when we are physicists?
2. Setting up the language of general ML task.
3. Understanding basic ML architectures.

Lecture 2

1. Walking through the examples of HEP and relate them
2. with the ML language from lecture-1.

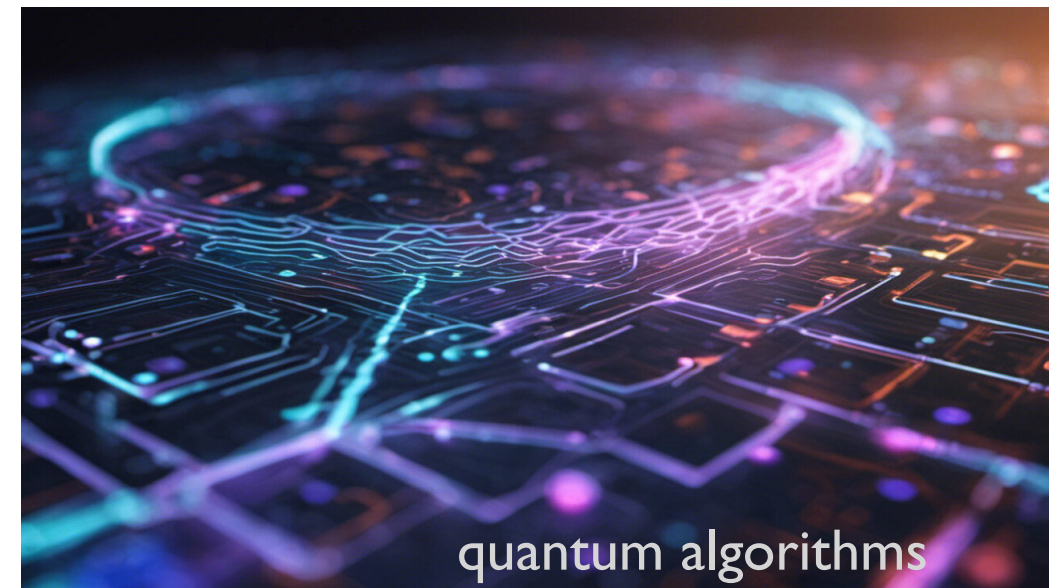
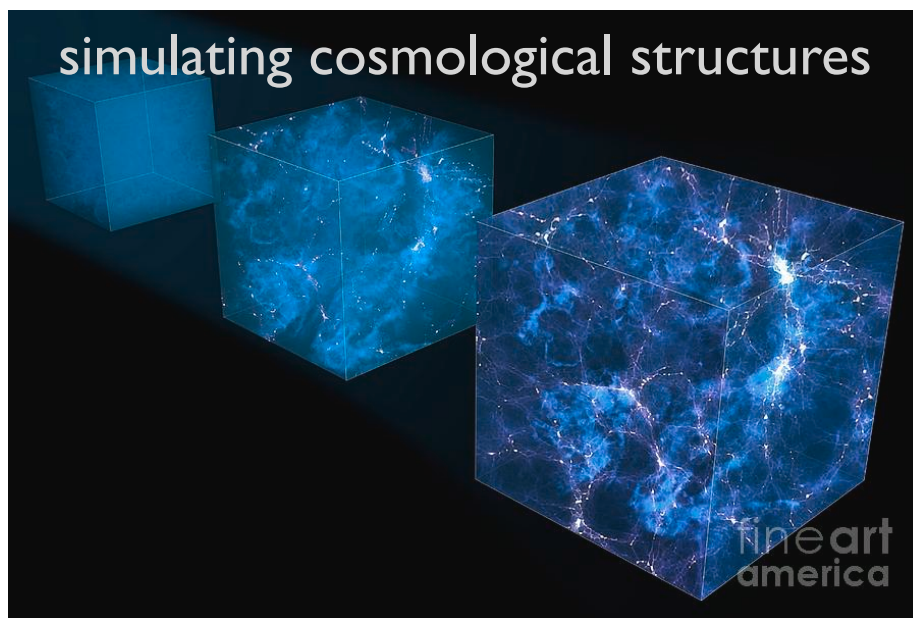
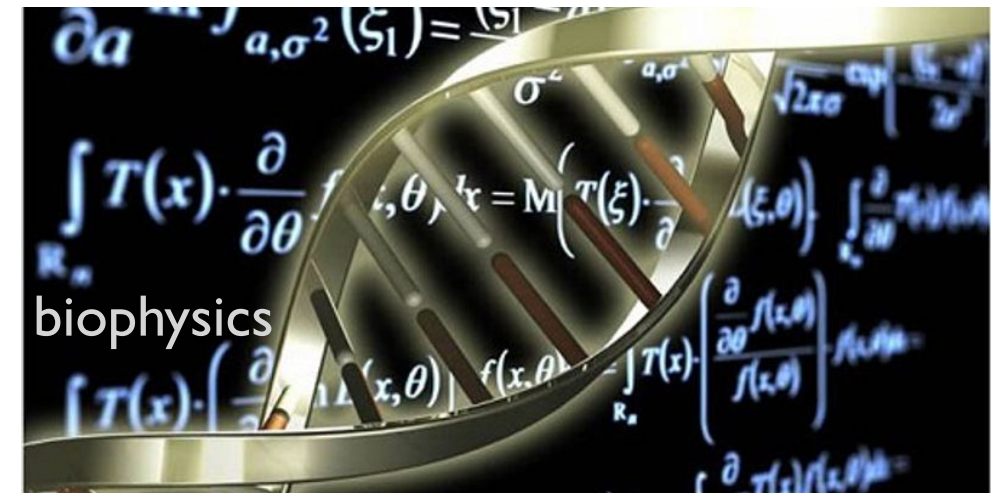
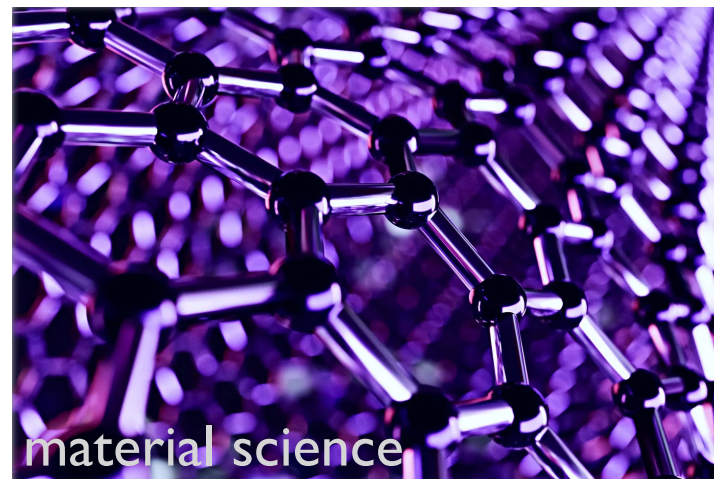
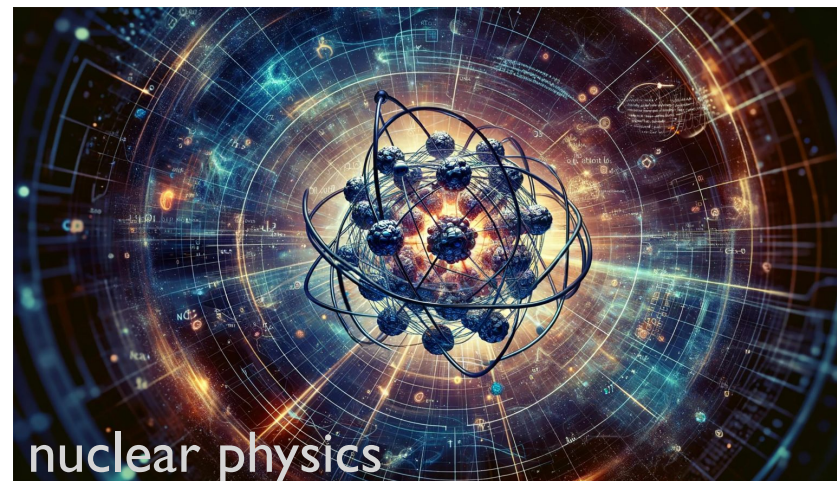
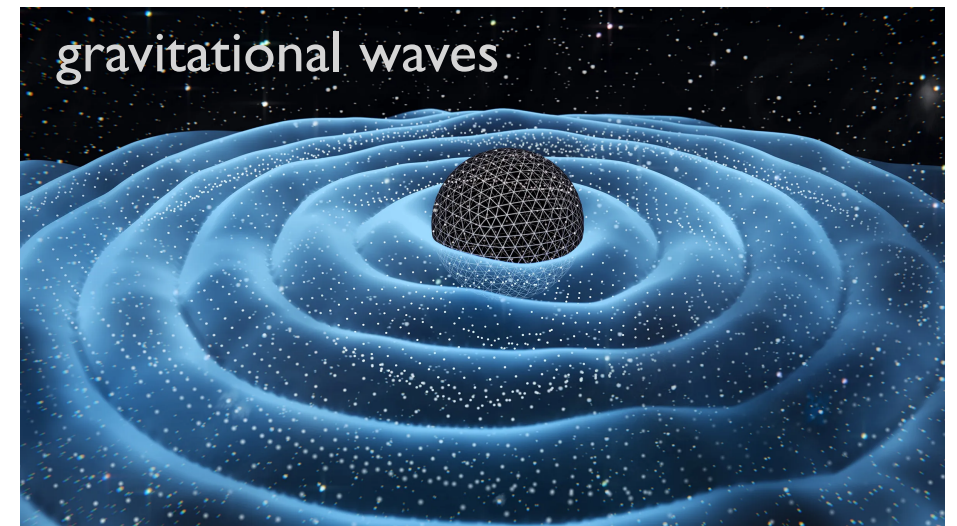
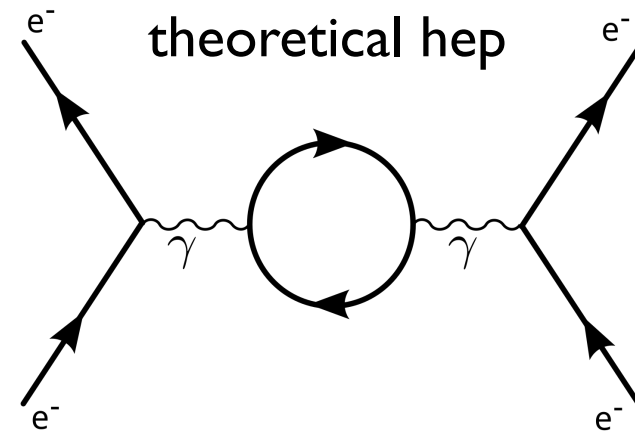
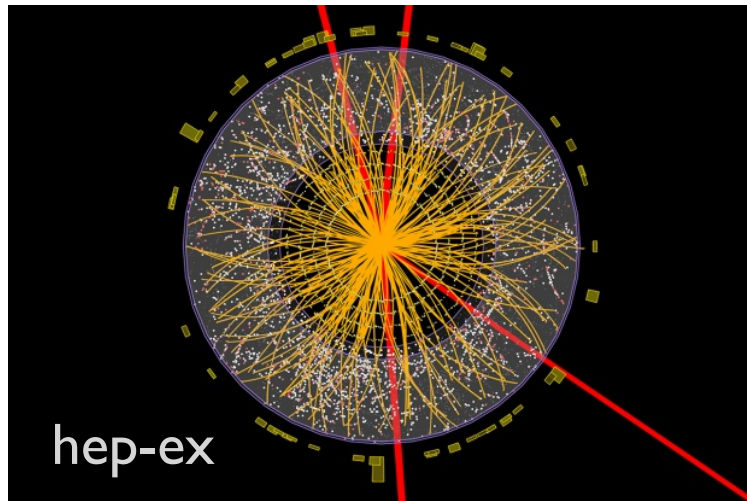
Lecture 3

1. Continue from lecture-2 and see where the future of ML X HEP is evolving.
2. Learning some basic PyTorch so that we can do some hands-on stuff.

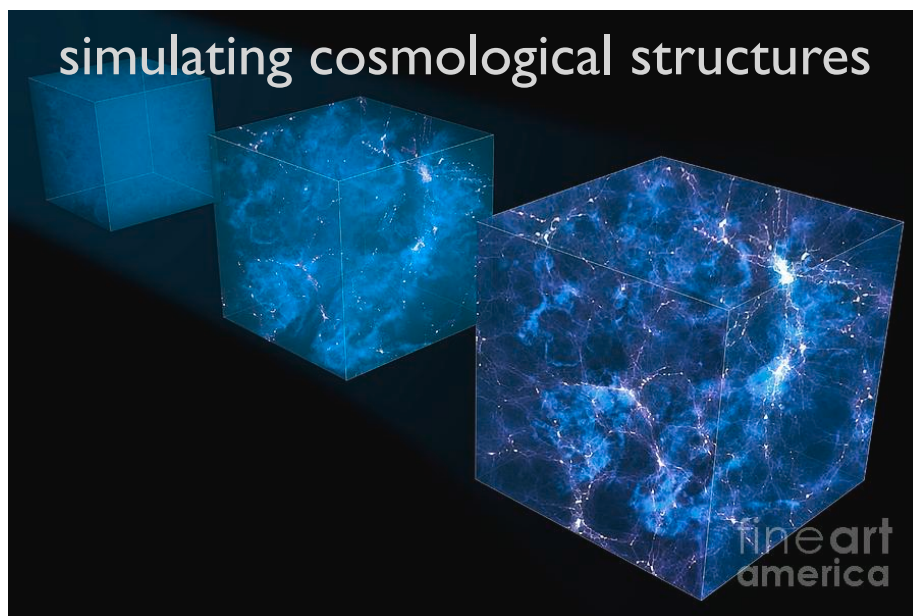
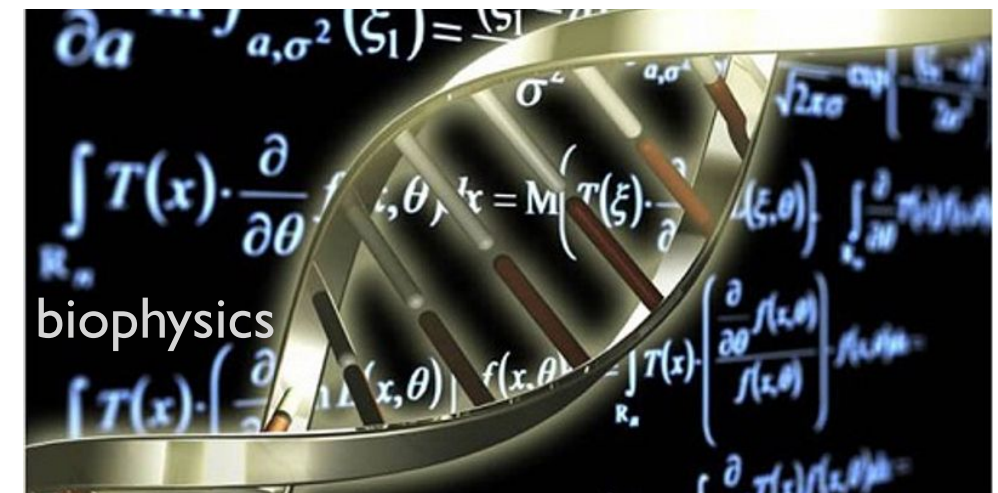
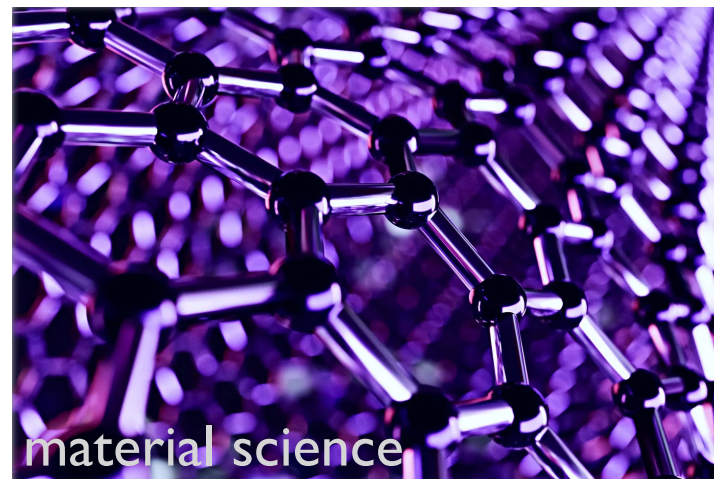
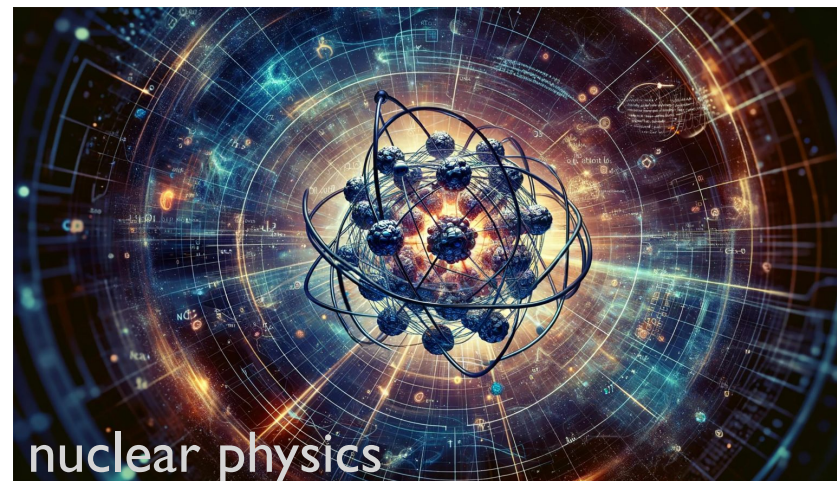
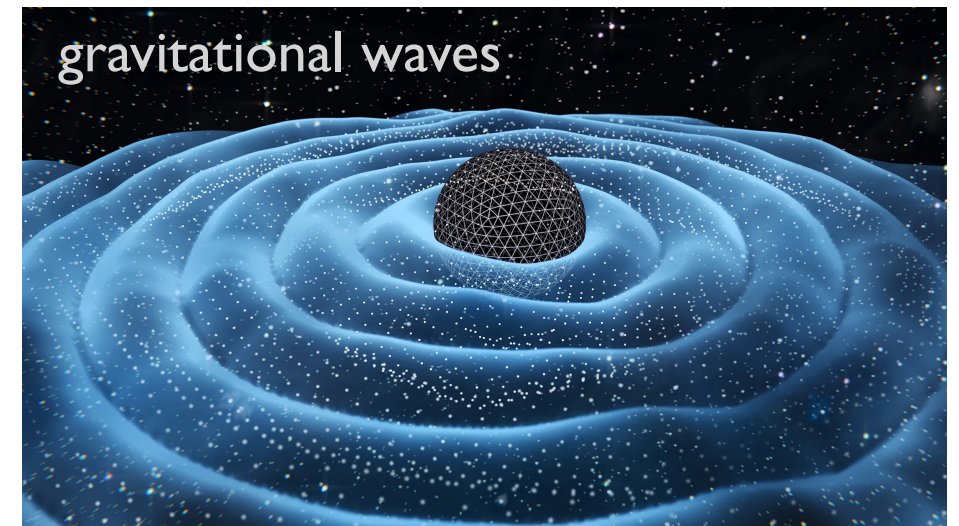
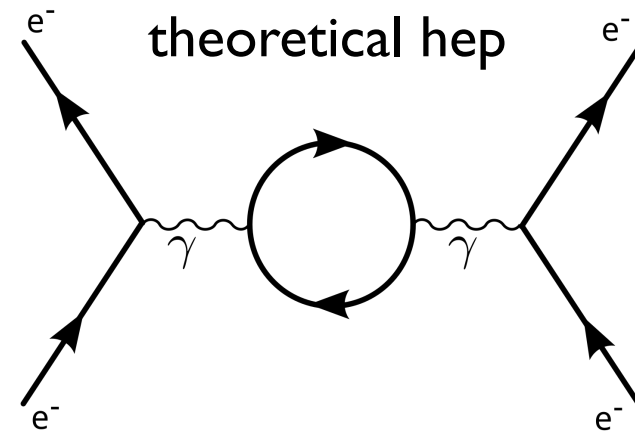
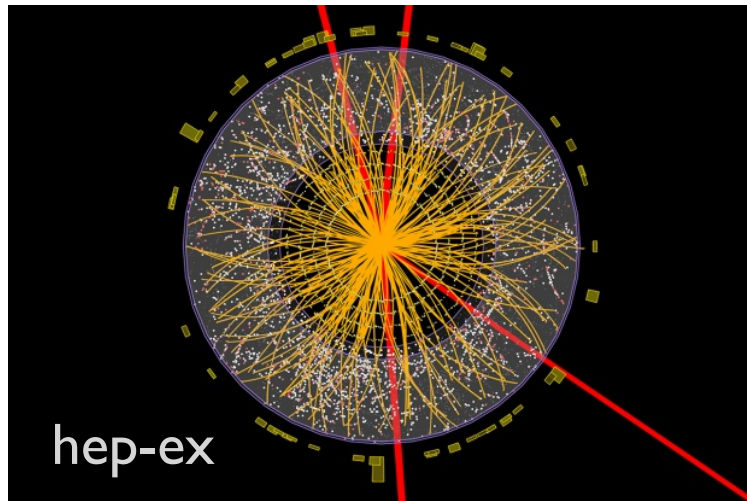
Lecture 4

1. Solving a jet tagging task using graph neural network from scratch.

We are a bunch of physicists : then why ML?

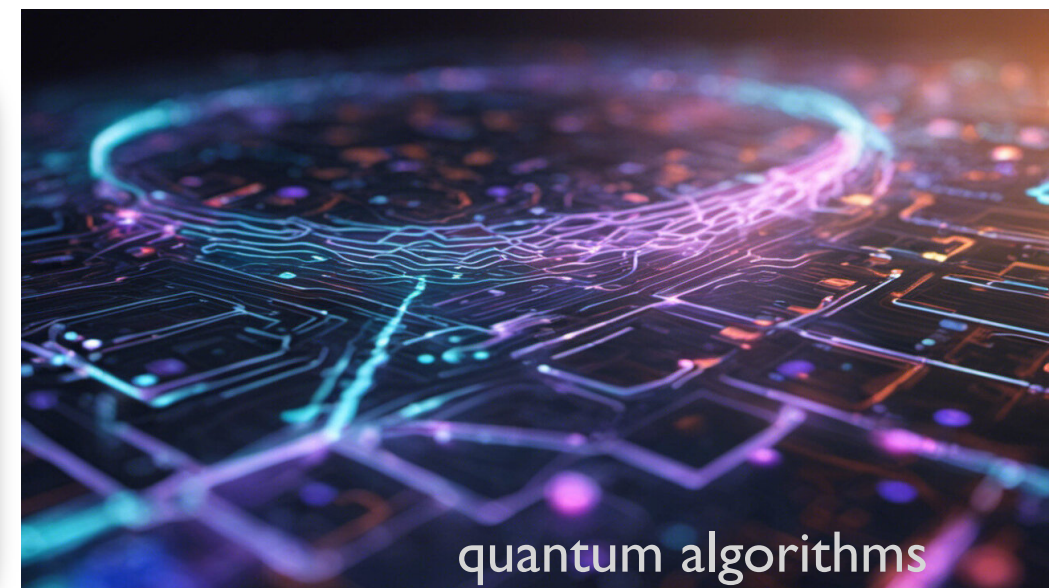


We are a bunch of physicists : then why ML?



and the coolest idea you
are about to bring ...

Are there any common
theme within all these
different branches?

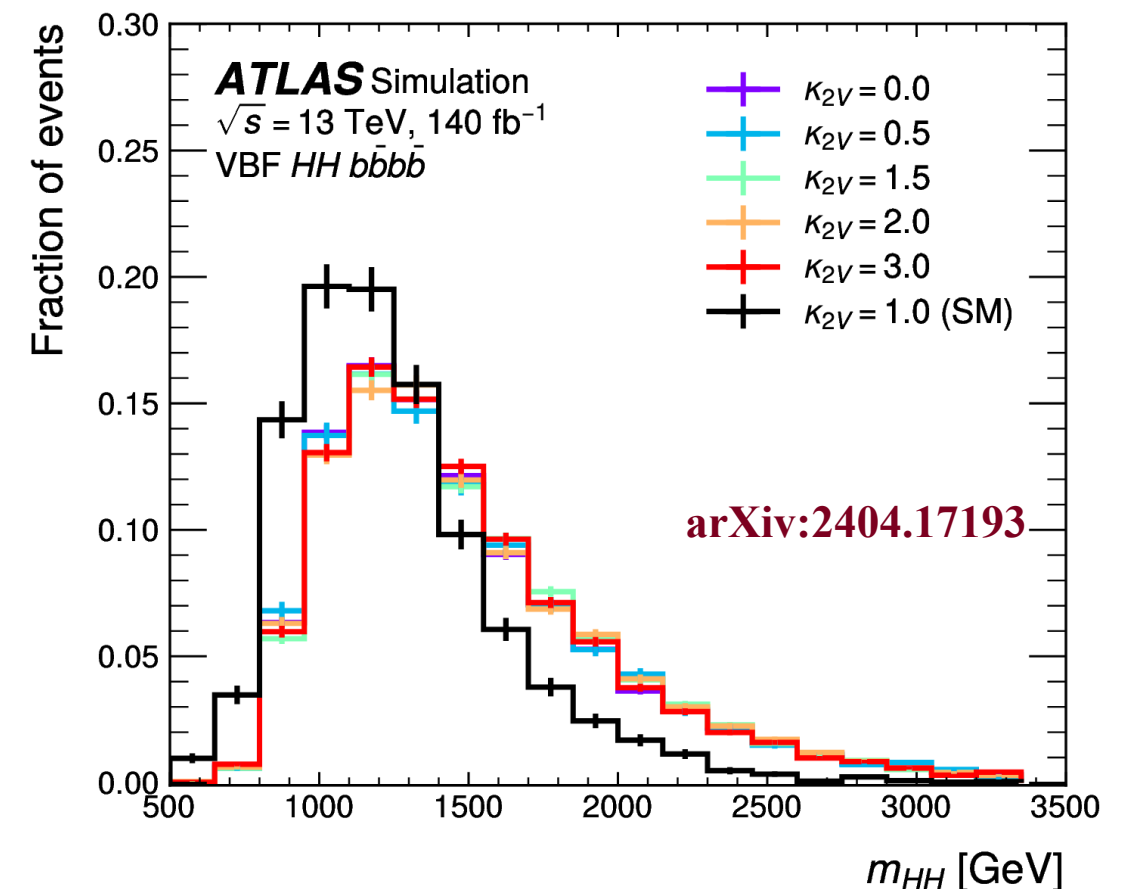
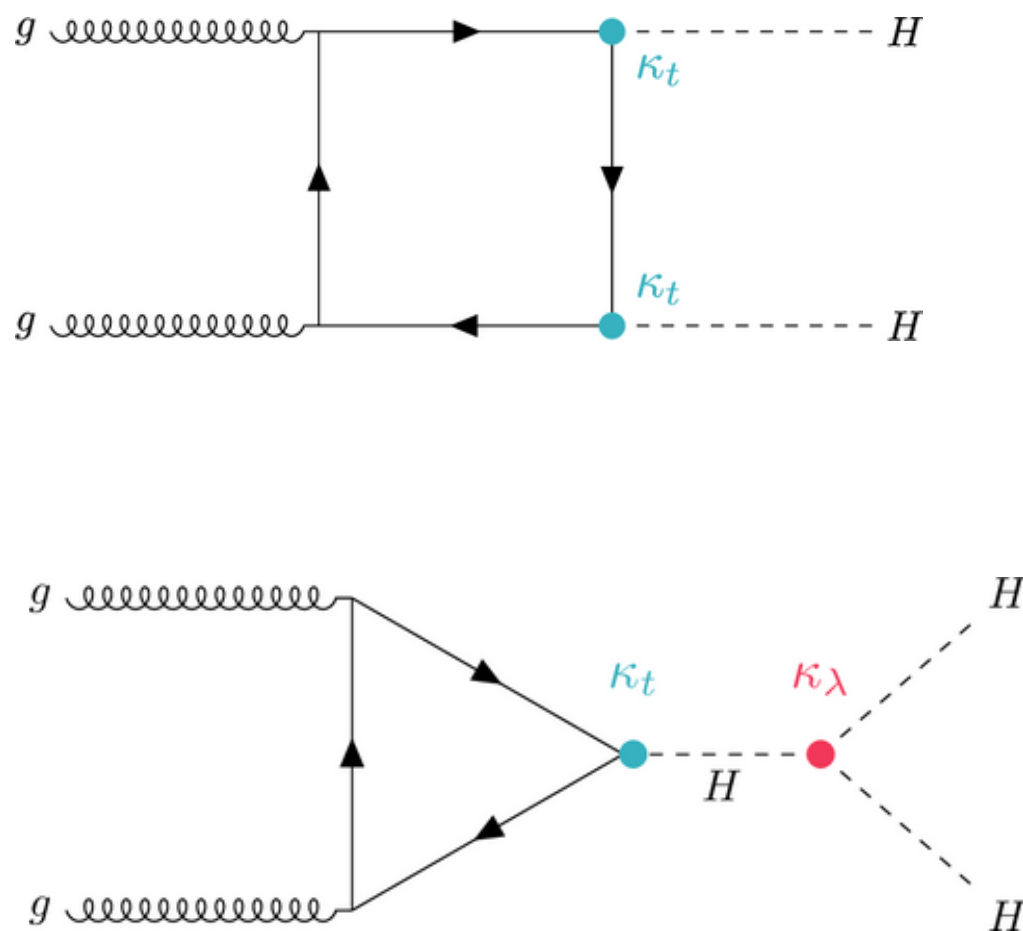


A common theme between all of them

Either we compute something :

$$f(x) \equiv f(x | \theta_{th})$$

$$\begin{aligned} \mathcal{L} = & -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} \\ & + i \bar{\Psi} \not{D} \Psi + h.c. \\ & + \bar{\Psi}_i y_{ij} \Psi_j \phi + h.c. \\ & + |D_\mu \phi|^2 - V(\phi) \end{aligned}$$

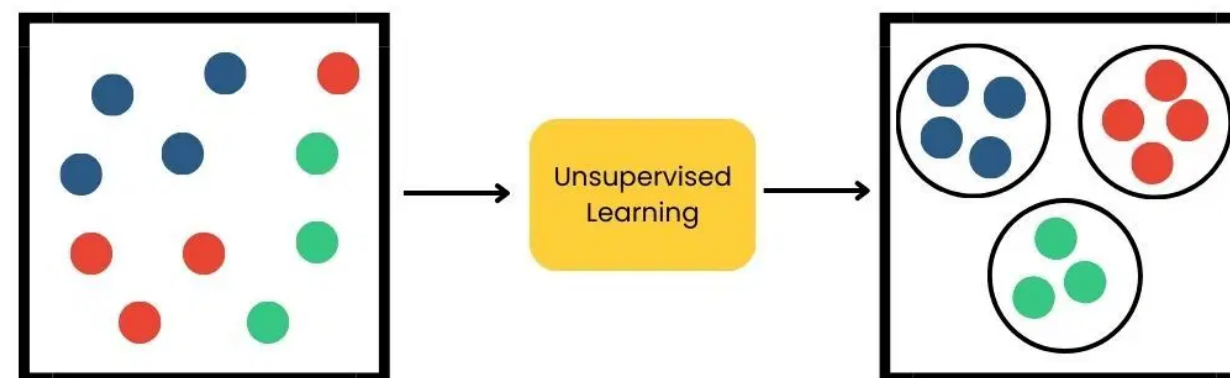


A common theme between all of them

Or we measure something and update our estimates :

$$p(\textit{theory} \mid \textit{data}) = \frac{p(\textit{data} \mid \textit{theory}) p(\textit{theory})}{p(\textit{data})}$$

And finally, we just do a measurement and do model independent statistical analysis (pattern identification) to achieve some understanding :



So why ML?

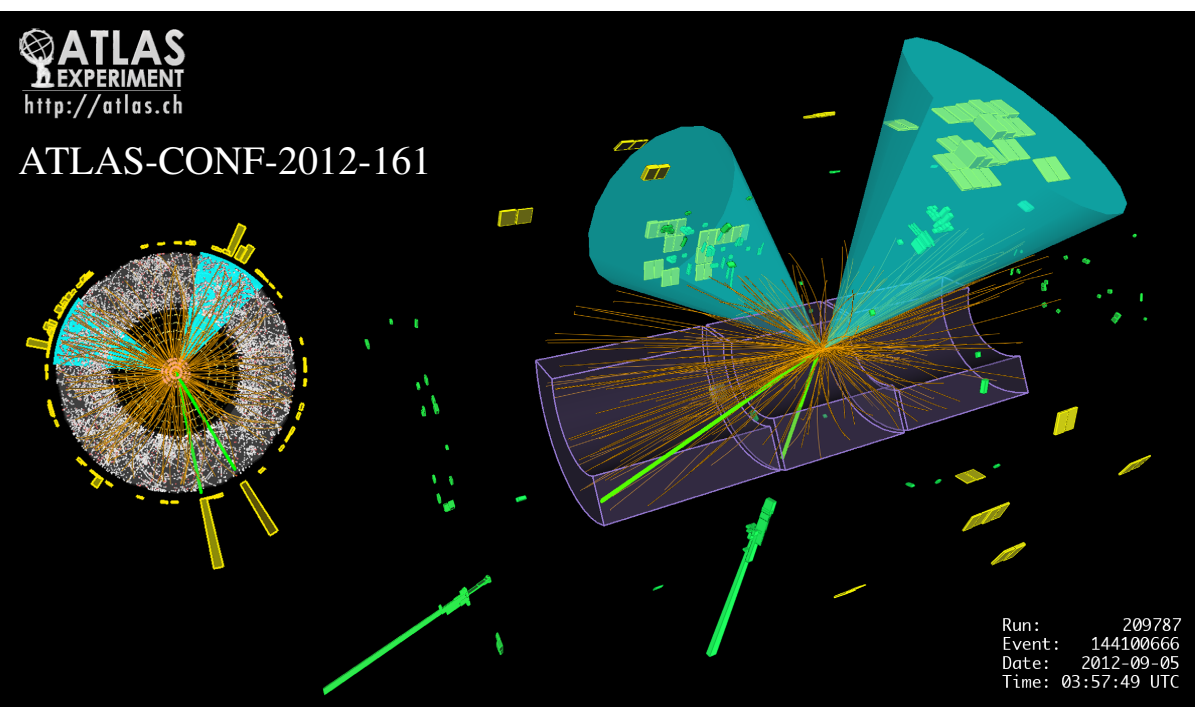
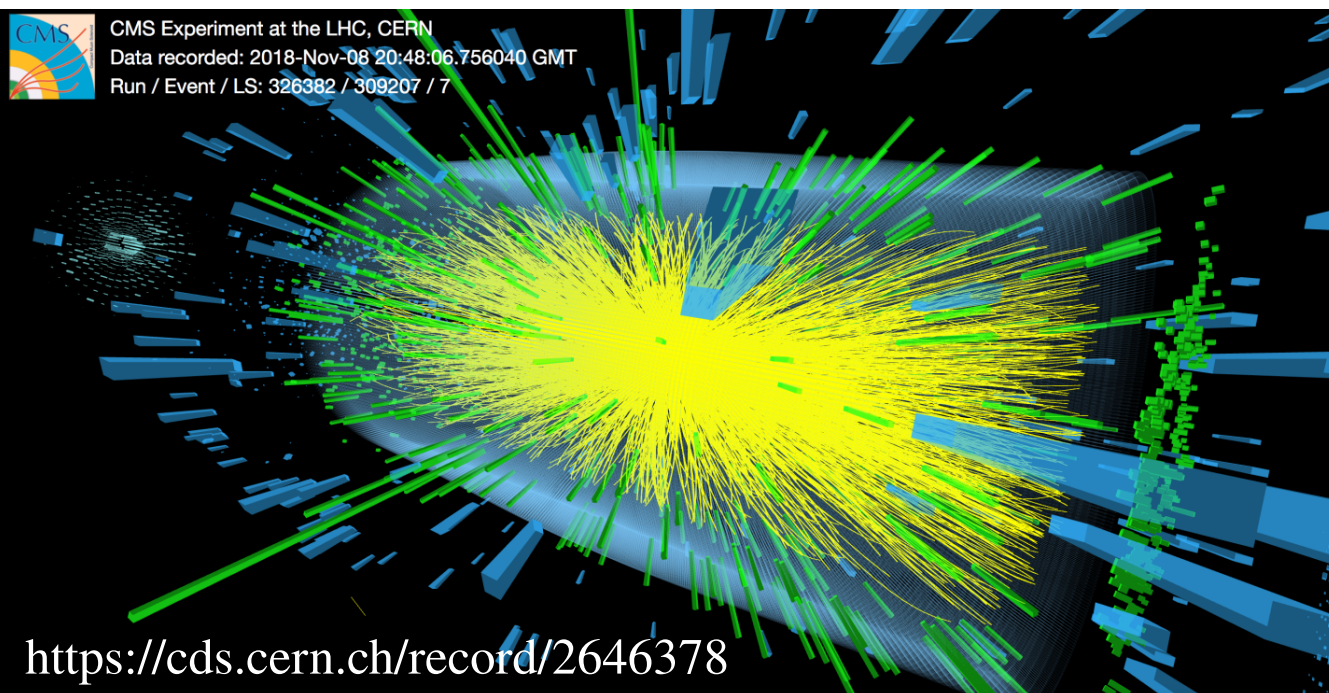
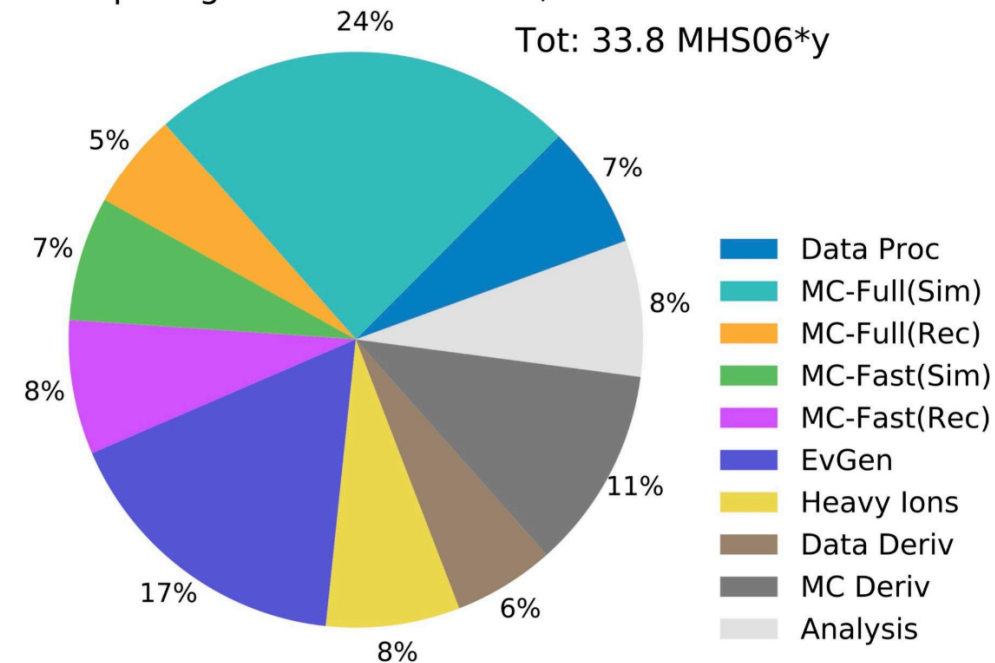
We often aim to solve hard computational expensive problems.

<https://cds.cern.ch/record/2802918>

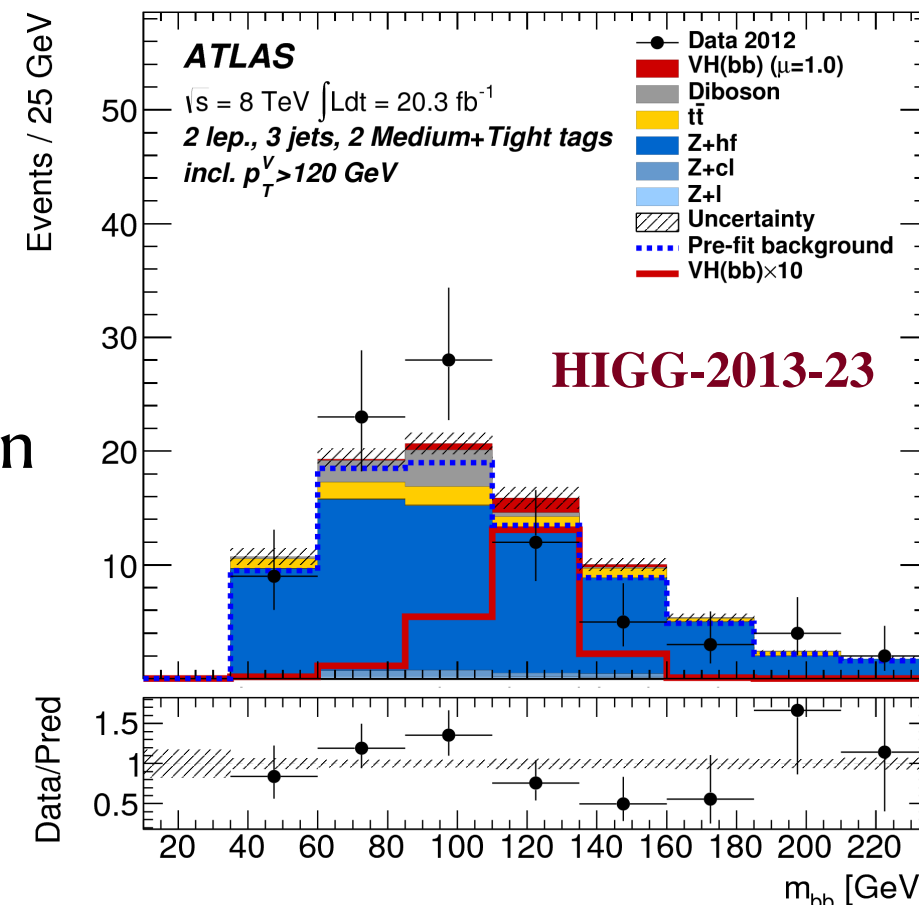
ATLAS Preliminary

2022 Computing Model - CPU: 2031, Conservative R&D

Tot: 33.8 MHS06*y

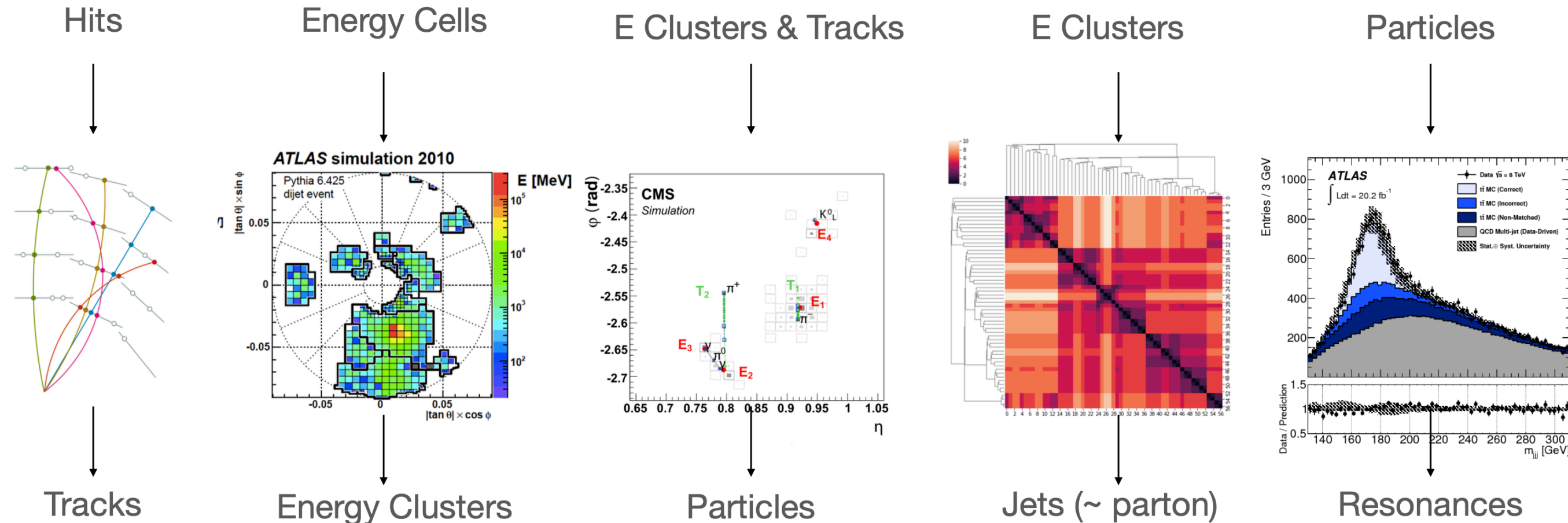


Find a good representation



Bringing the data into more tractable form

Fig. from Lukas Heinrich



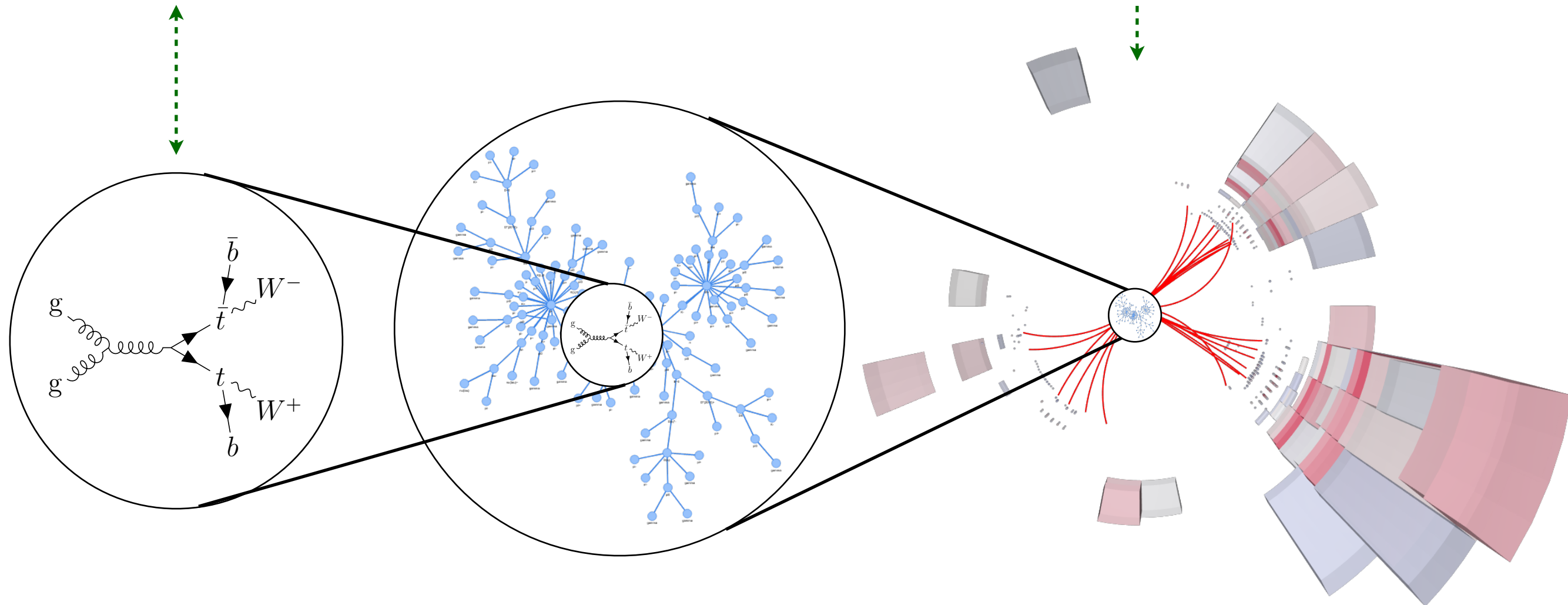
- ✓ On one hand ML is handy to produce large scale simulations.
- ✓ On the other hand, finding the right pattern in complex data is something ML is designed to do.
- ✓ In the era of data driven science, ML thus becomes an inevitable statistical tool.

A broad strategy towards physics inference

Guess the Lagrangian

← solving the inverse problem via ML

$$\frac{1}{\mathcal{L}} \frac{d^2 N}{dp_T d\eta}$$



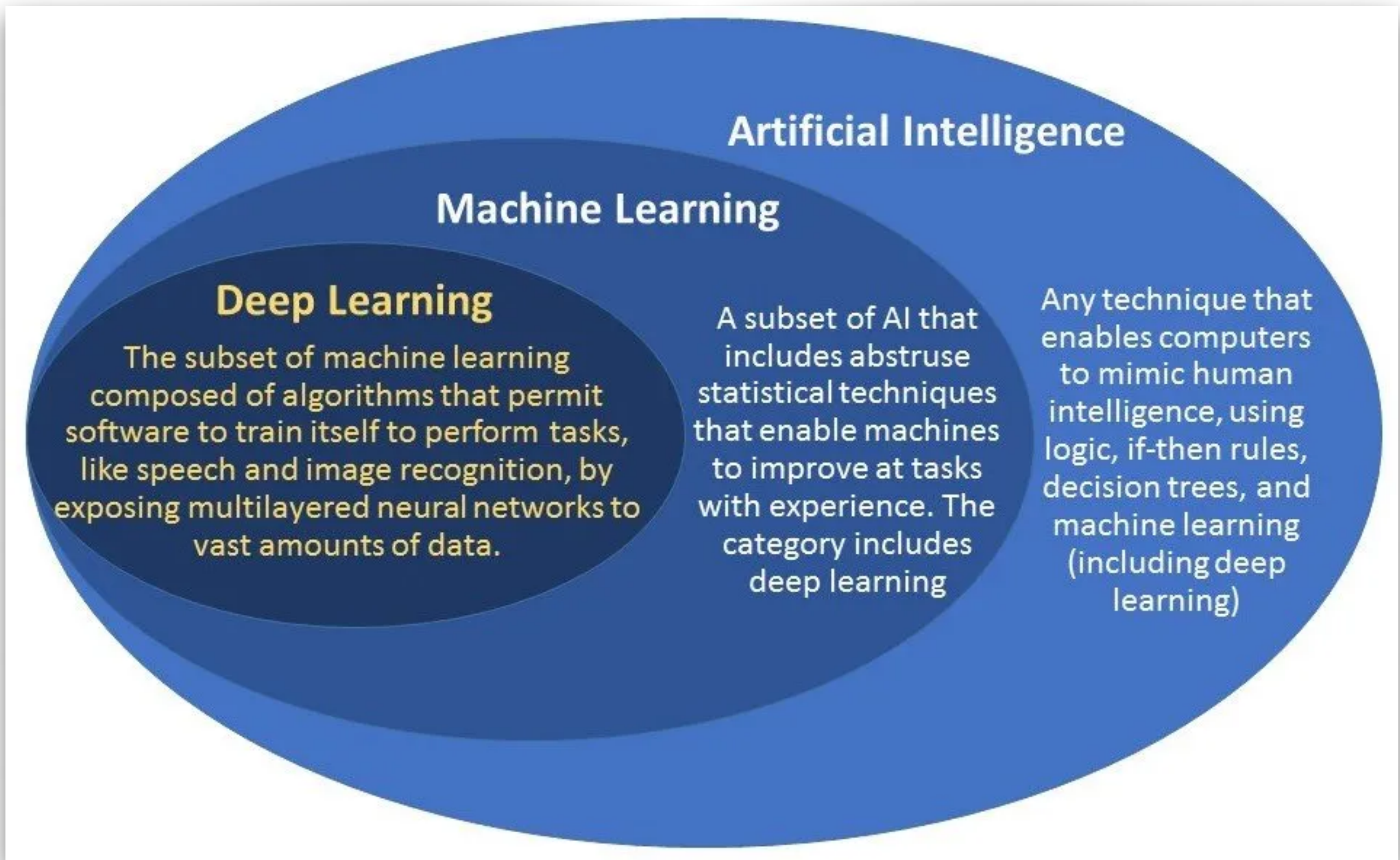
The physics at the core:
driven by the interaction
between quantum fields,
computed via perturbative
or lattice techniques.

Final Particles
Produced via hadronization
(no first principle analytic
techniques are available)

Detector output/Readout
Produced via hardware
or simulation

For complex hard tasks, ML serves the purpose in both direction

Let's understand : what is AI/ML/DL?

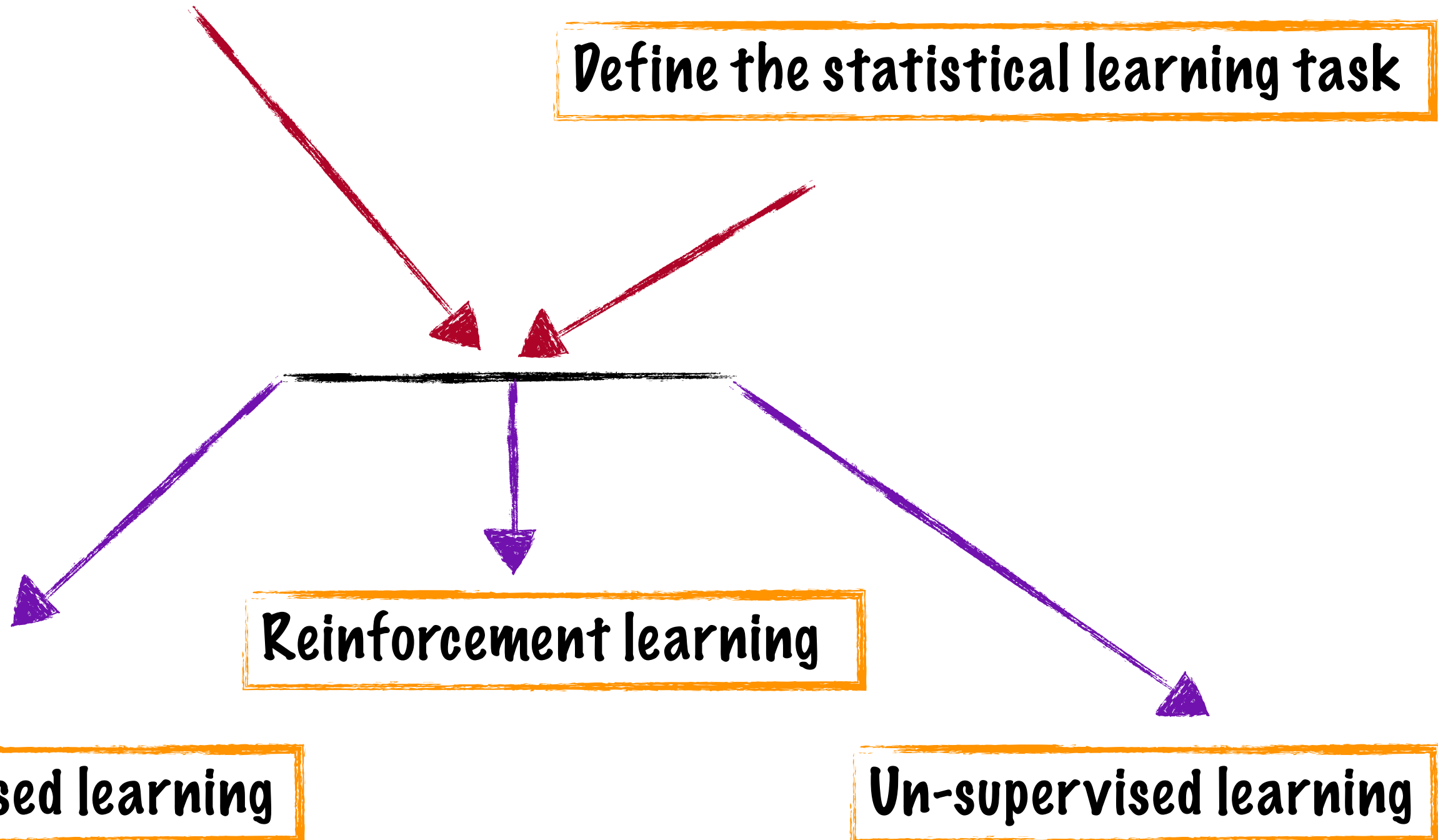


<https://medium.com/@clairedigitalogy/what-is-machine-learning-deep-learning-7788604004da>

The broad classes of ML task

Check the nature of data available
(labelled or unlabeled data)

Define the statistical learning task

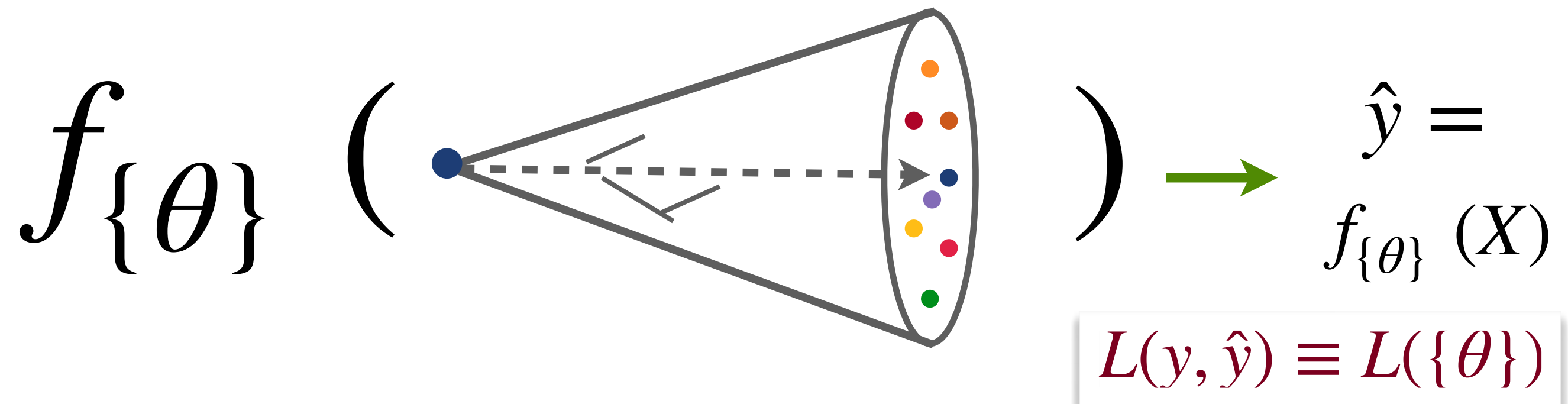


Supervised learning

Reinforcement learning

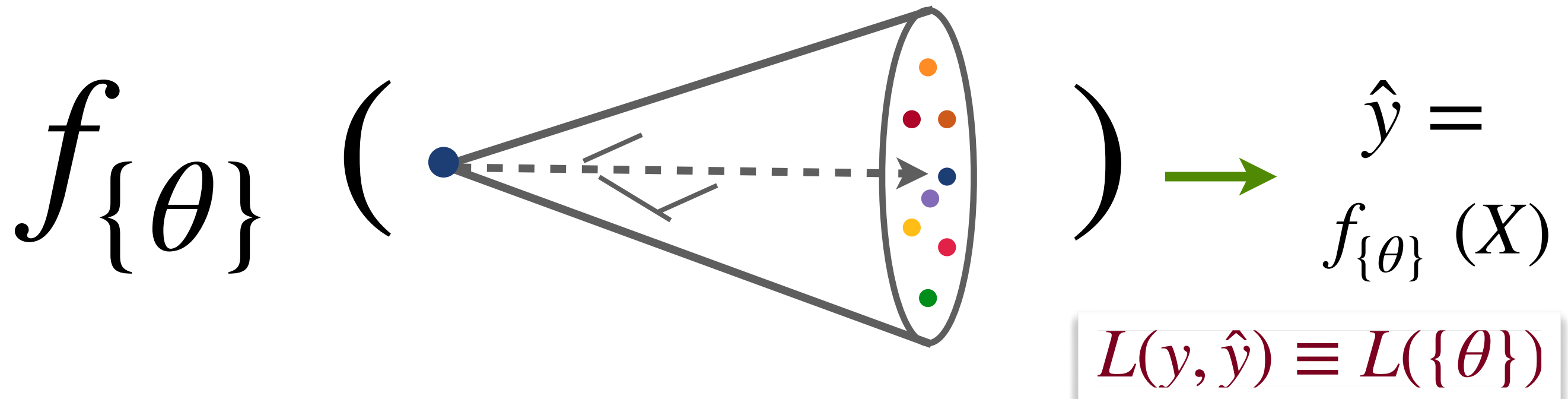
Un-supervised learning

ML@HEP : what's the broad task?



ML@HEP : what's the broad task?

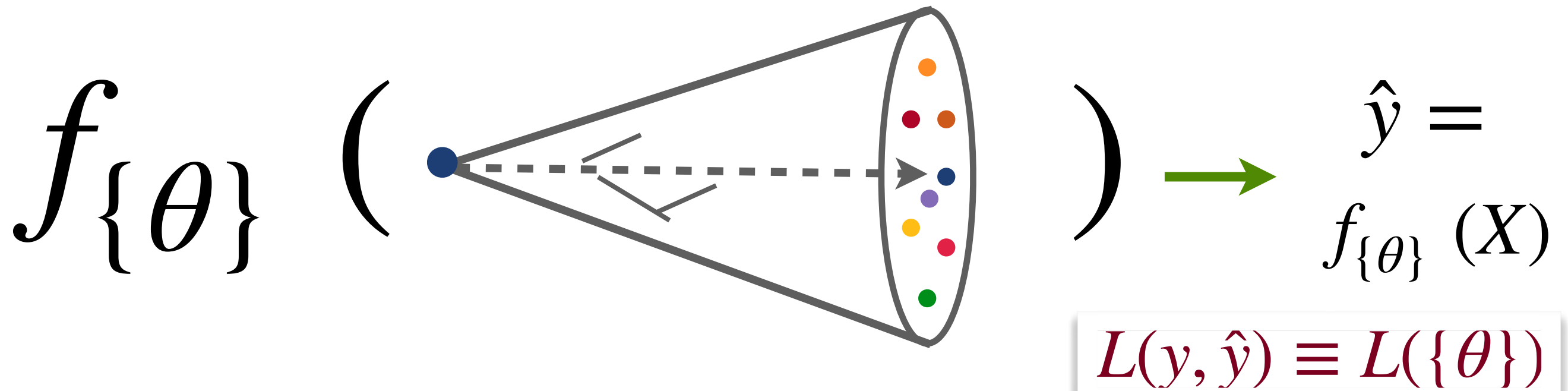
1. Decide the right representation of the data (images/graphs/trees..)



ML@HEP : what's the broad task?

2. Choose a NN model
(CNN/GNN/)

1. Decide the right representation
of the data (images/graphs/trees..)

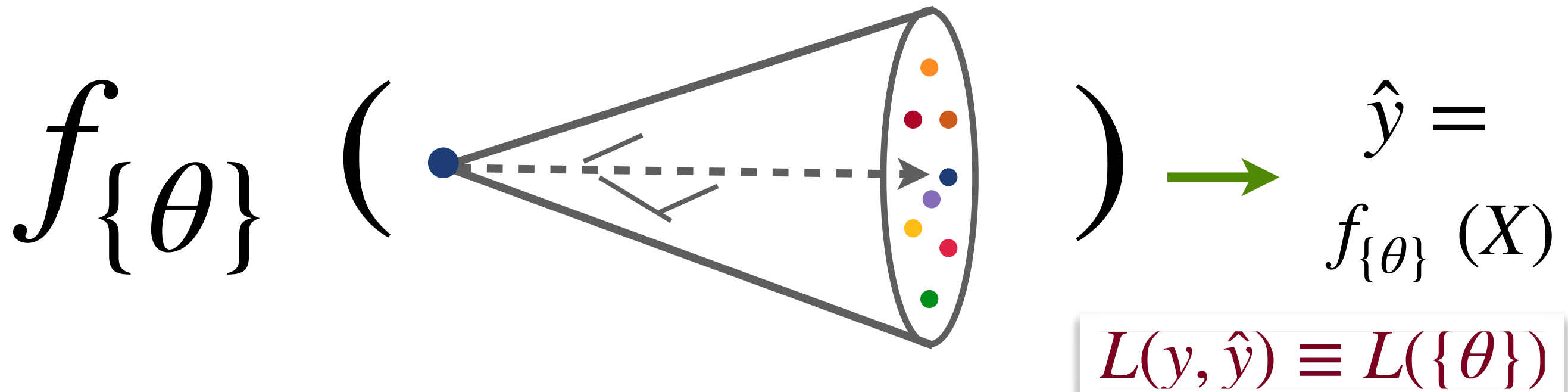


ML@HEP : what's the broad task?

2. Choose a NN model
(CNN/GNN/)

1. Decide the right representation
of the data (images/graphs/trees..)

3. With a defined learning task,
compute the loss function.

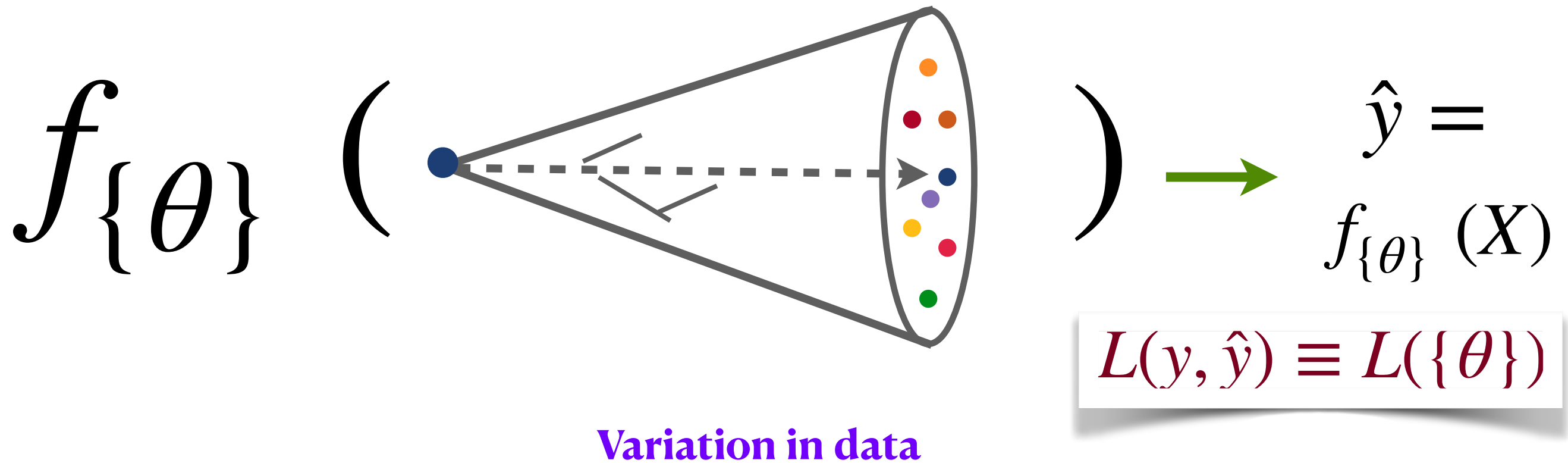


ML@HEP : what's the broad task?

2. Choose a NN model
(CNN/GNN/)

1. Decide the right representation
of the data (images/graphs/trees..)

3. With a defined learning task,
compute the loss function.

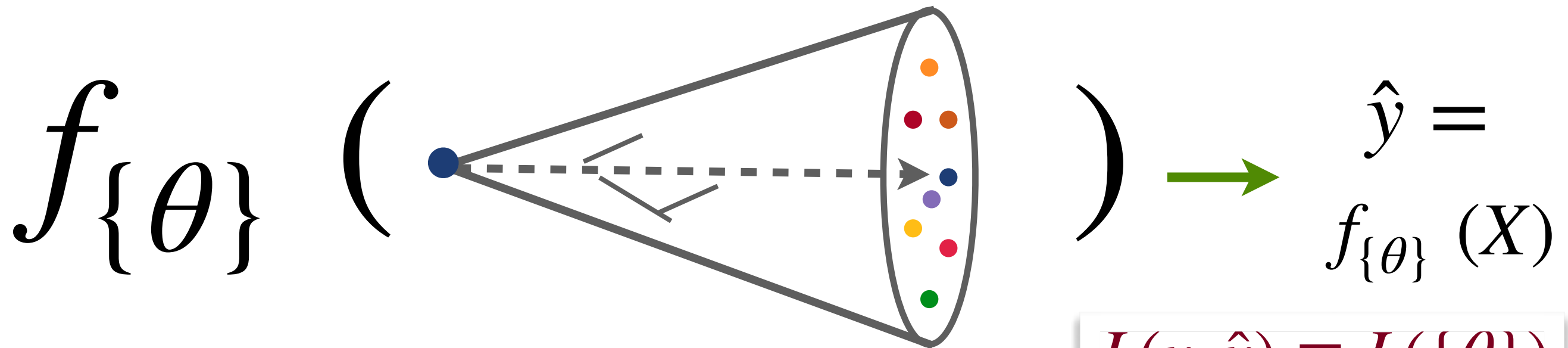


ML@HEP : what's the broad task?

2. Choose a NN model
(CNN/GNN/)

1. Decide the right representation
of the data (images/graphs/trees..)

3. With a defined learning task,
compute the loss function.



$$L(y, \hat{y}) \equiv L(\{\theta\})$$

Variation in data

Unsupervised

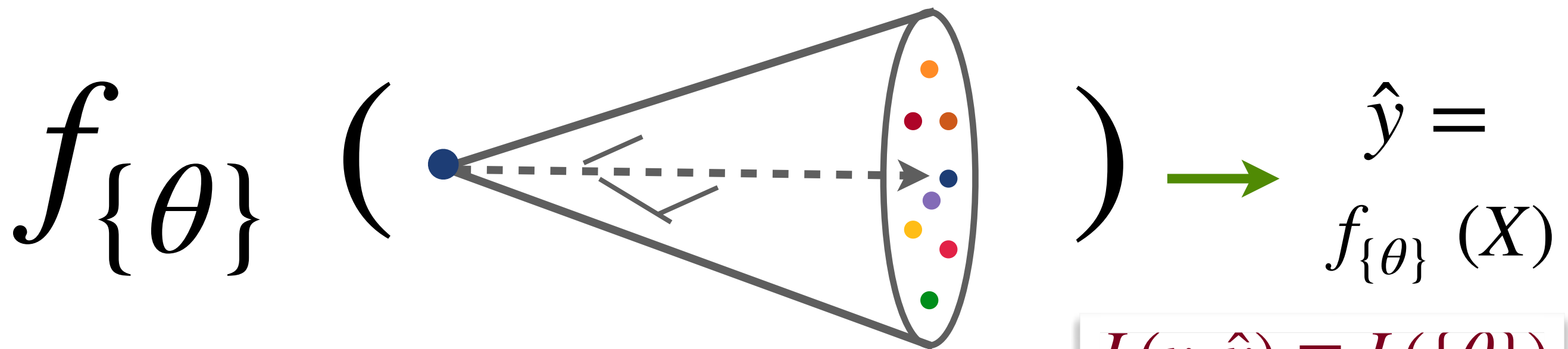
No-labels, the task is to
figure out $p(x)$ from which
the data is drawn. e.g. VAE

ML@HEP : what's the broad task?

2. Choose a NN model
(CNN/GNN/)

1. Decide the right representation
of the data (images/graphs/trees..)

3. With a defined learning task,
compute the loss function.



$$L(y, \hat{y}) \equiv L(\{\theta\})$$

Variation in data

Unsupervised

Semi-supervised

No-labels, the task is to
figure out $p(x)$ from which
the data is drawn. e.g. VAE

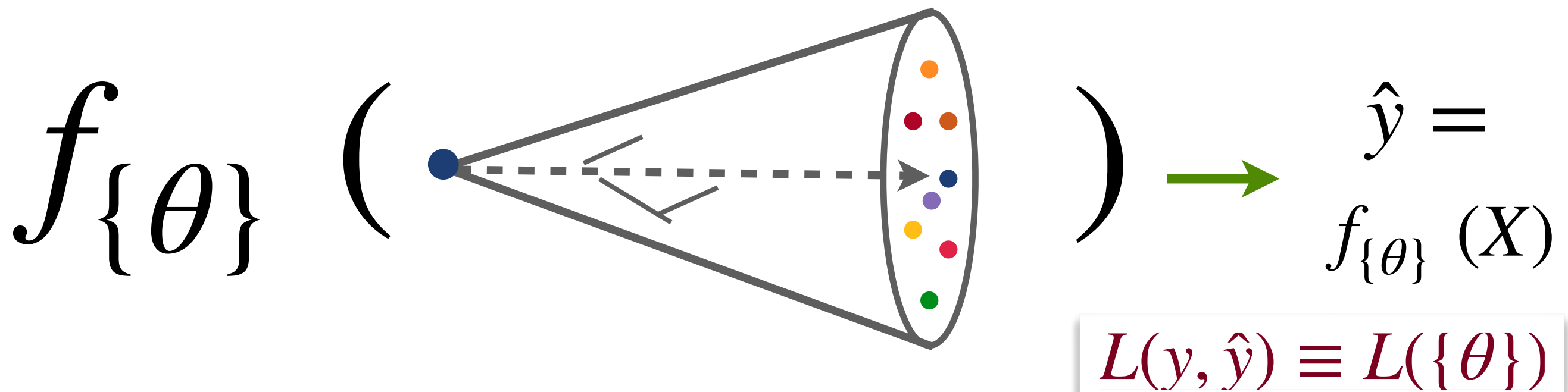
Noisy labels. estimate :
 $p(s\text{-enriched})/p(s\text{-depleted})$

ML@HEP : what's the broad task?

2. Choose a NN model
(CNN/GNN/)

1. Decide the right representation
of the data (images/graphs/trees..)

3. With a defined learning task,
compute the loss function.



Variation in data

Unsupervised

Semi-supervised

Weakly-supervised

No-labels, the task is to figure out $p(x)$ from which the data is drawn. e.g. VAE

Noisy labels. estimate : $p(s\text{-enriched})/p(s\text{-depleted})$

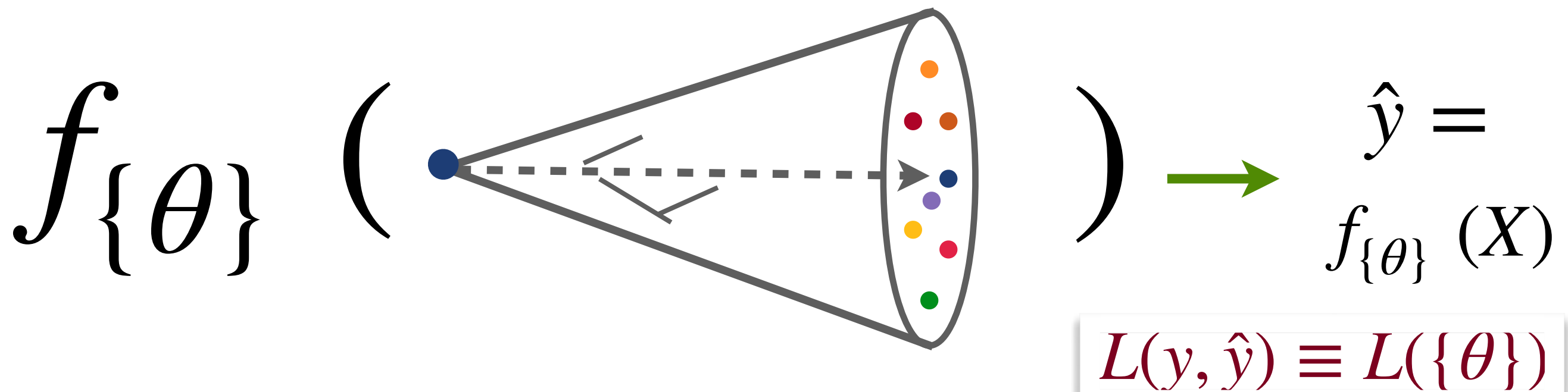
Partial labels. e.g. simulating : SM bkg vs many NP signals.

ML@HEP : what's the broad task?

2. Choose a NN model
(CNN/GNN/)

1. Decide the right representation
of the data (images/graphs/trees..)

3. With a defined learning task,
compute the loss function.



Variation in data

Unsupervised

Semi-supervised

Weakly-supervised

Supervised

No-labels, the task is to figure out $p(x)$ from which the data is drawn. e.g. VAE

Noisy labels. estimate : $p(s\text{-enriched})/p(s\text{-depleted})$

Partial labels. e.g. simulating : SM bkg vs many NP signals.

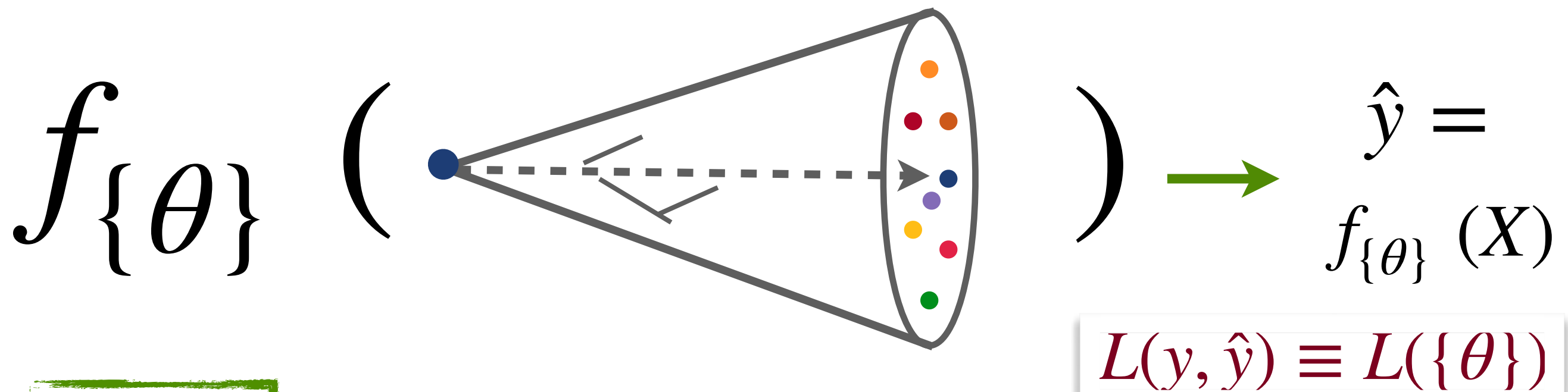
Learning on all the well labeled data.

ML@HEP : what's the broad task?

2. Choose a NN model
(CNN/GNN/)

1. Decide the right representation
of the data (images/graphs/trees..)

3. With a defined learning task,
compute the loss function.



Self-supervised

Variation in data

Unsupervised

Semi-supervised

Weakly-supervised

Supervised

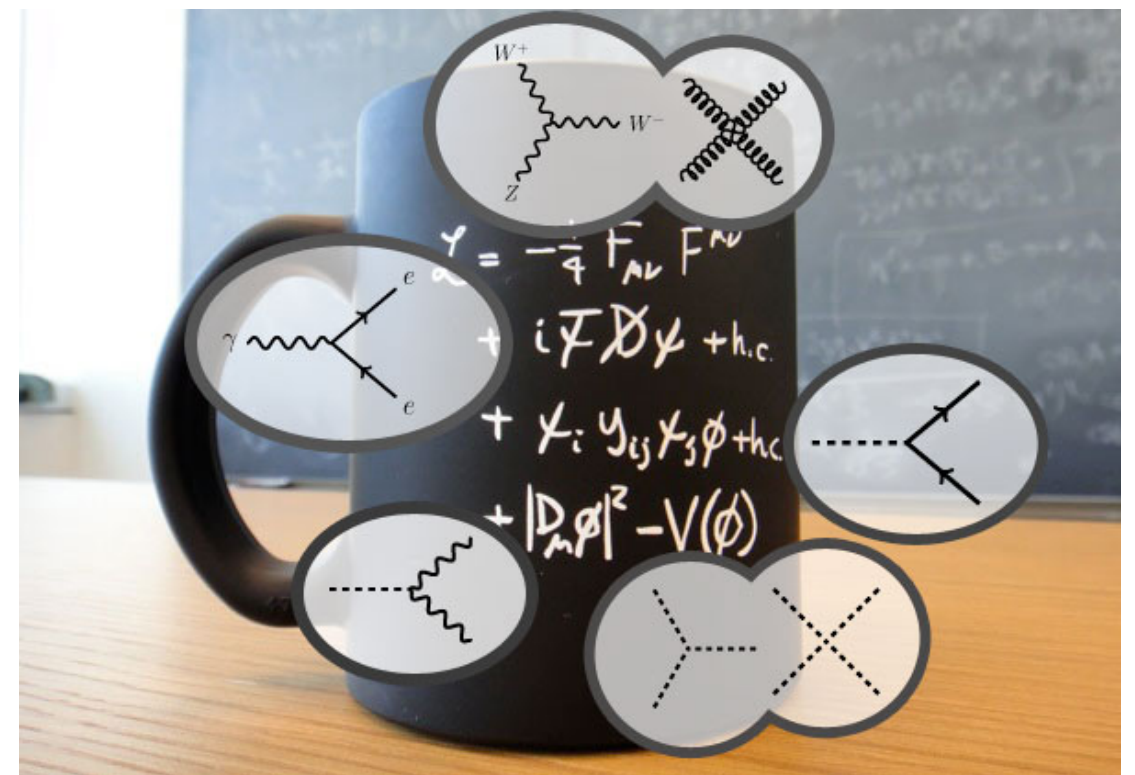
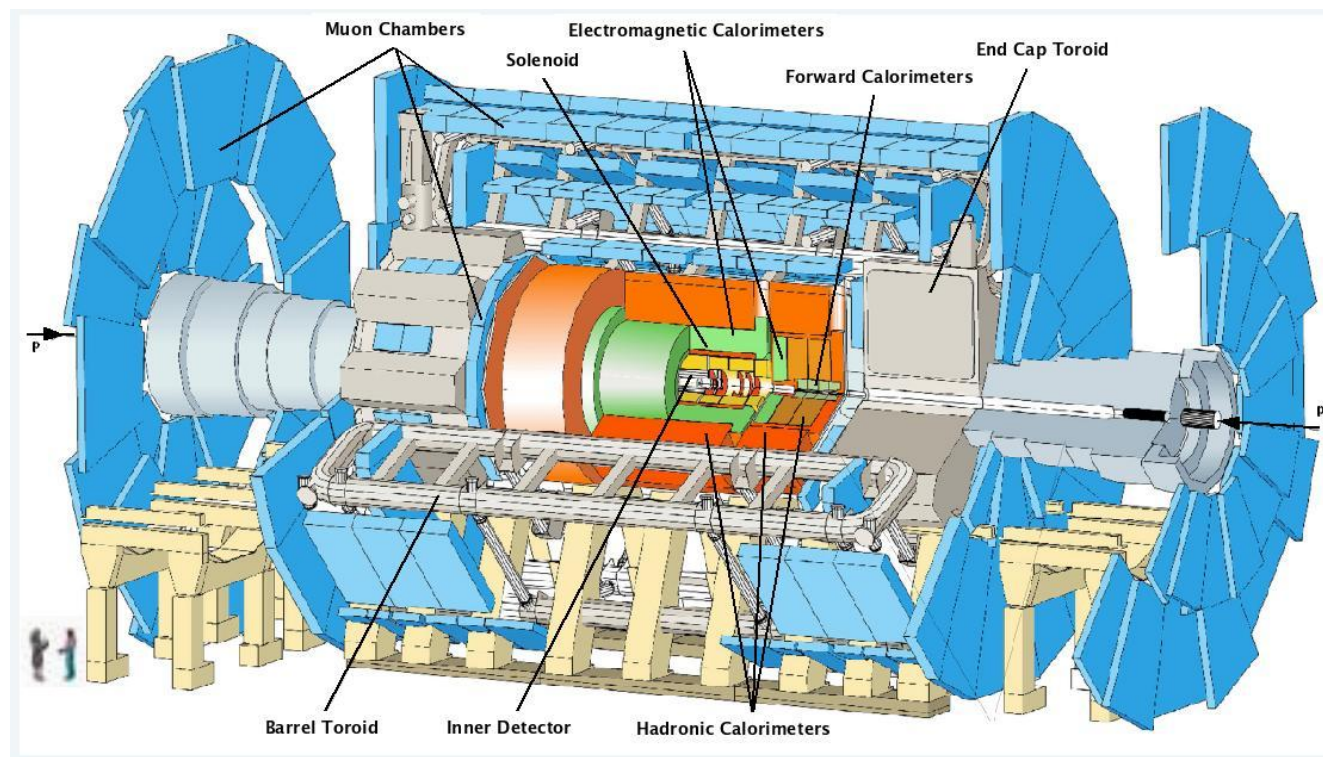
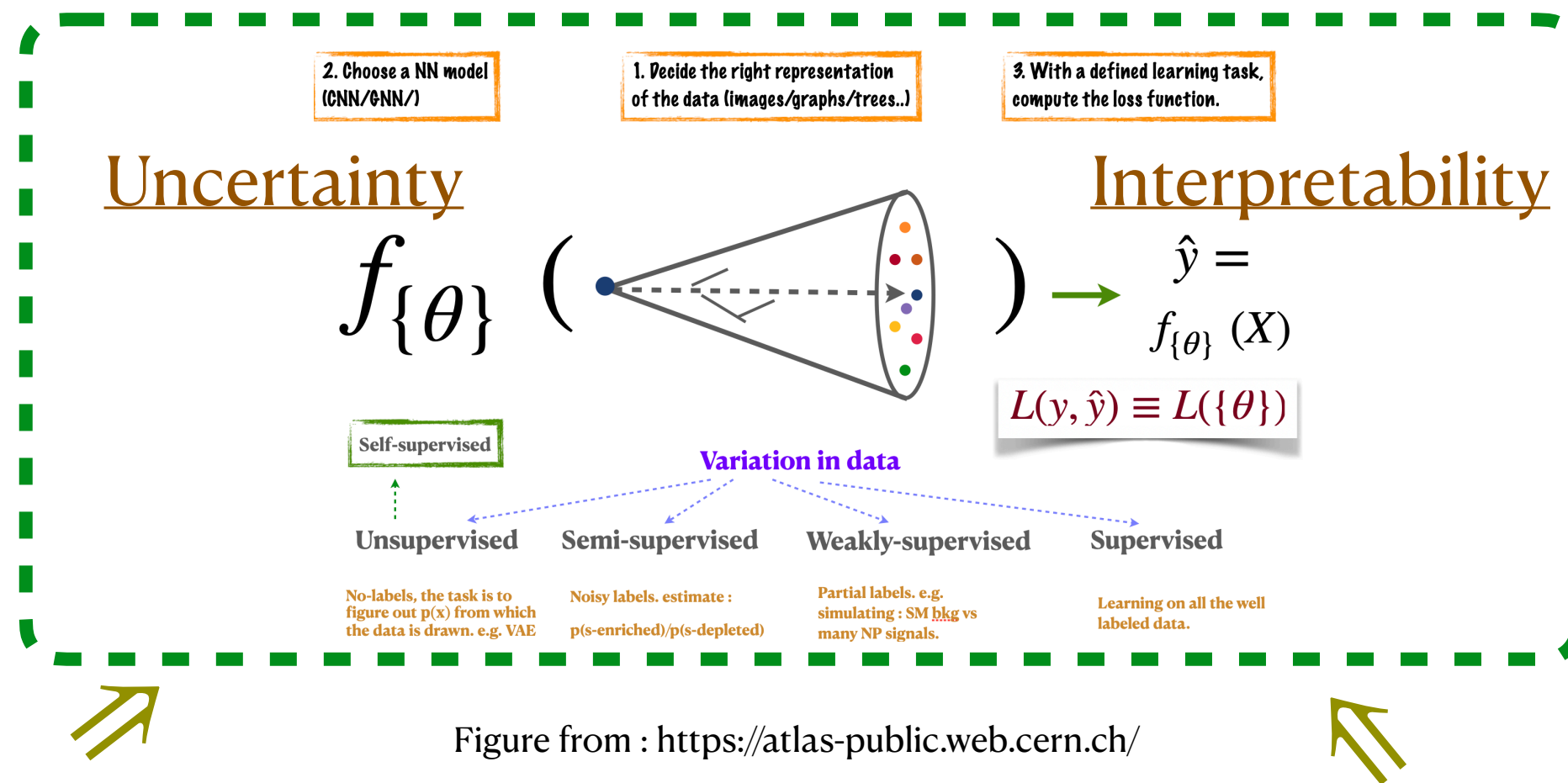
No-labels, the task is to figure out $p(x)$ from which the data is drawn. e.g. VAE

Noisy labels. estimate : $p(s\text{-enriched})/p(s\text{-depleted})$

Partial labels. e.g. simulating : SM bkg vs many NP signals.

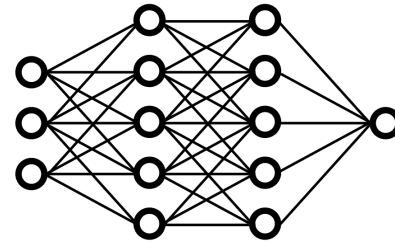
Learning on all the well labeled data.

Looking the problem through ML lens

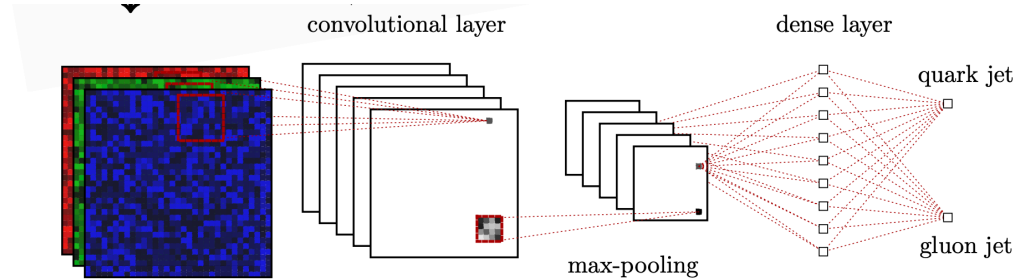
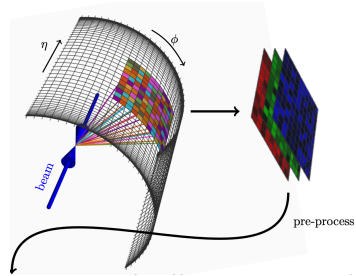


Data representation $\Leftrightarrow$ NN correspondence

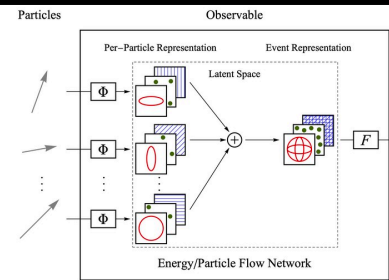
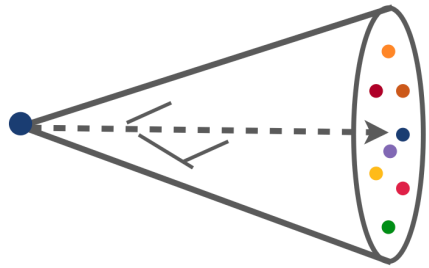
$$J = \{p_1^\mu, p_2^\mu, \dots\}$$



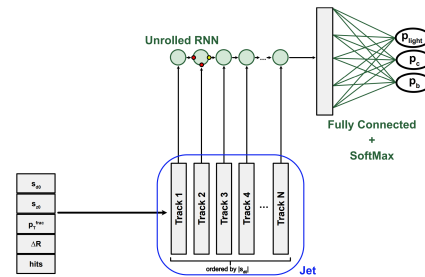
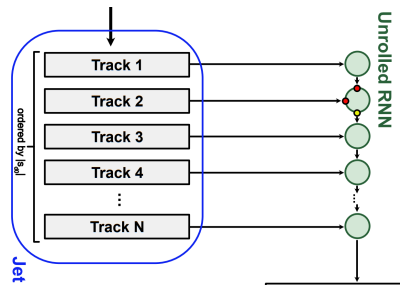
Ordered set
DNN



Grid
CNN

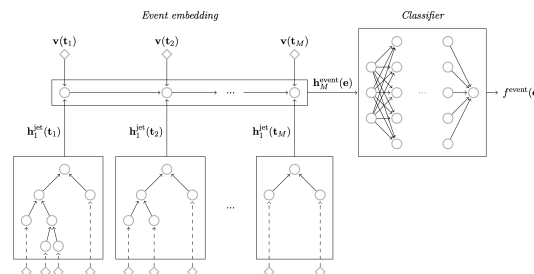
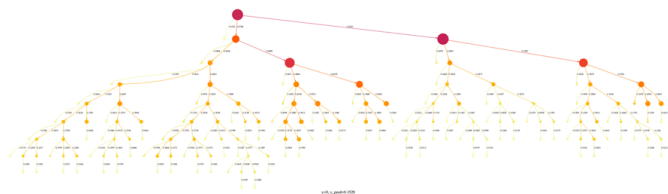


Unordered set
Deepest

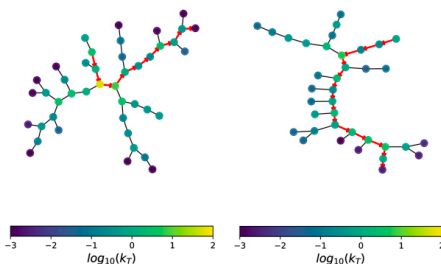


Sequential data
RNN

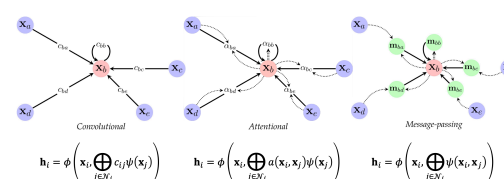
ML4Jets



Tree structure
Deepset/GNN

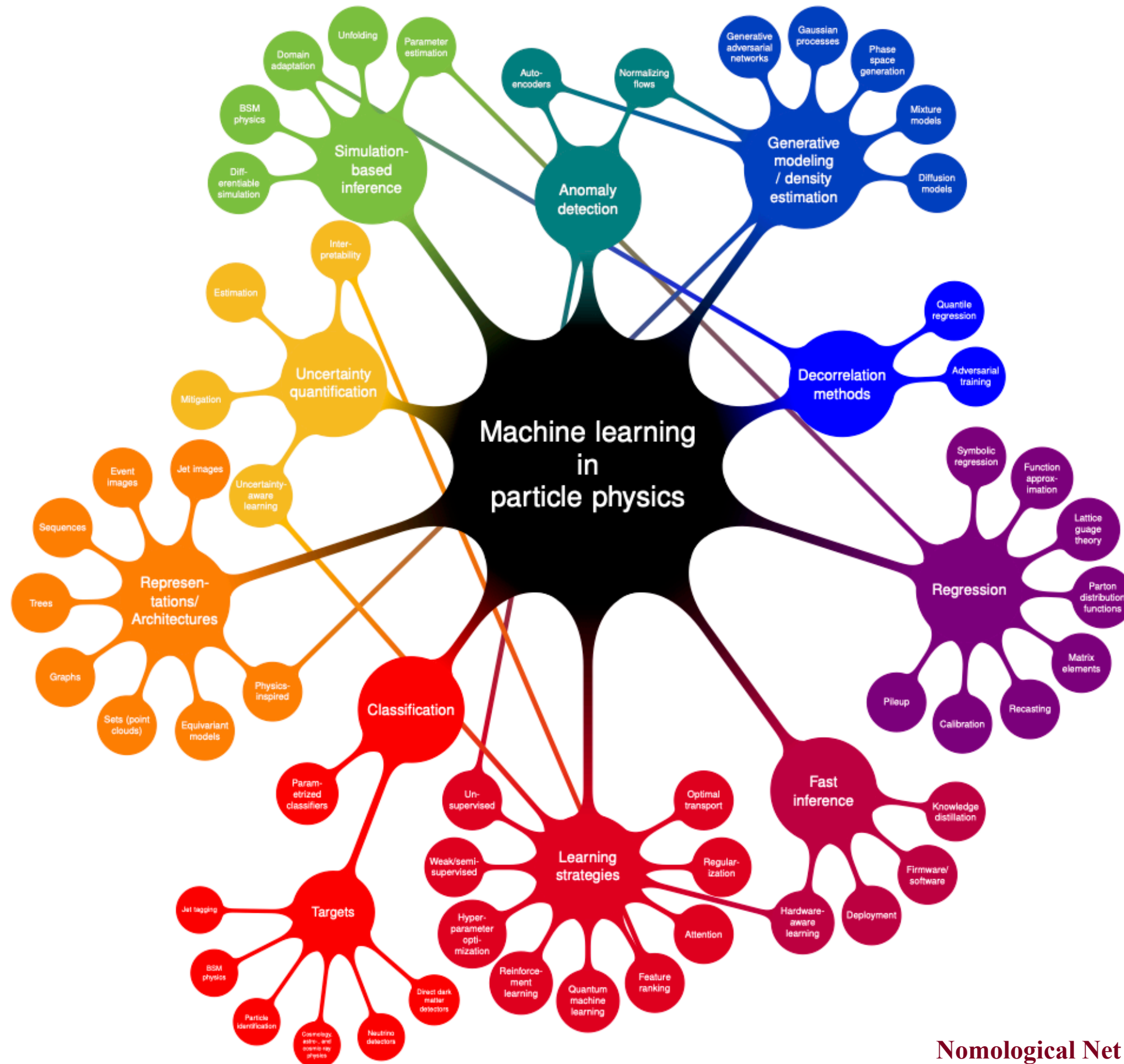


The three "flavours" of GNN layers



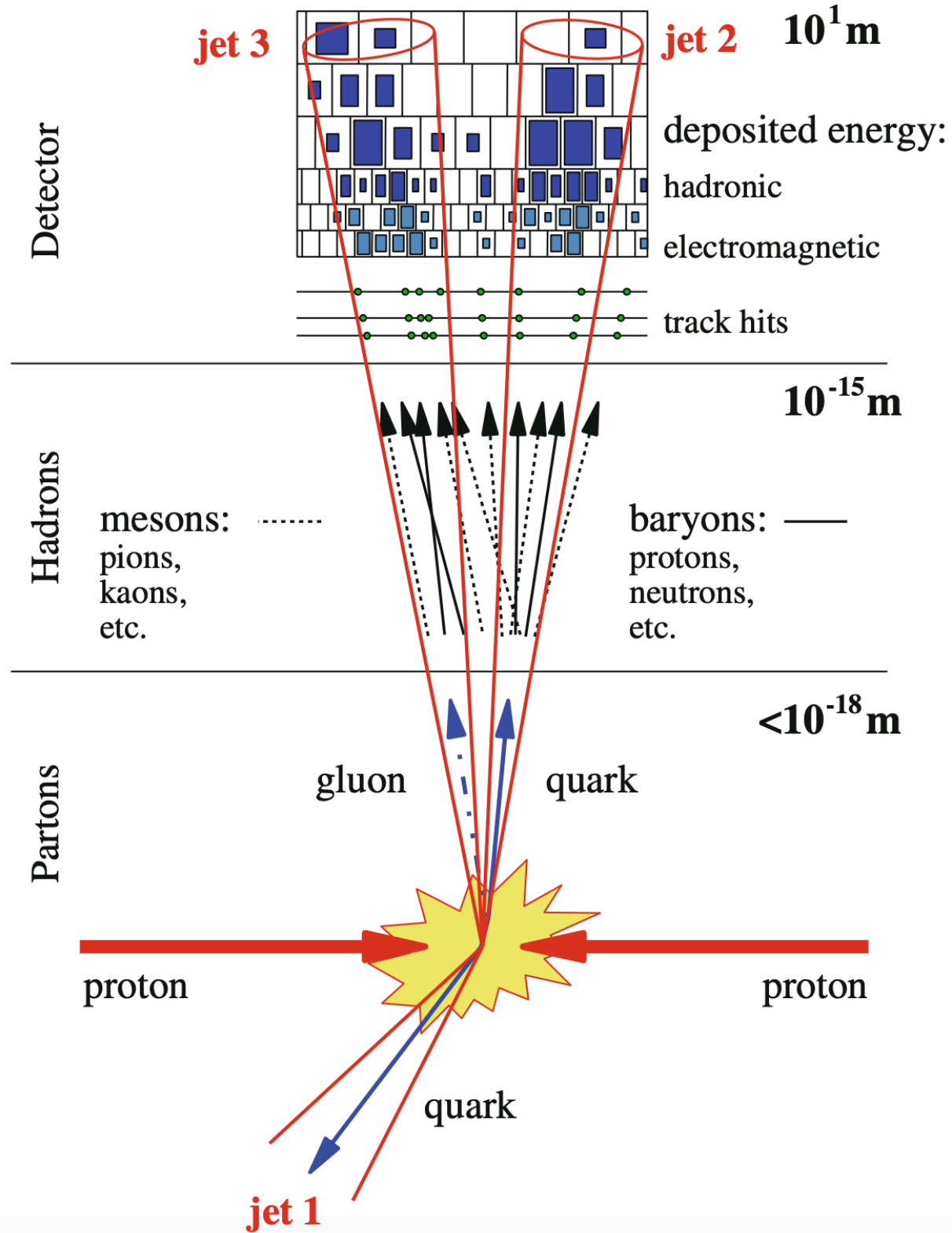
Graph
GNN

Application of ML in HEP



Nomological Net for ML in Particle Physics

Jet tagging : a popular case



$$\{p_1, p_2, \dots, p_n\}$$

Jet Algorithm (for CA, kT, anti-kT)

$$\{j_1, j_2, \dots, j_k\}$$

$$\{p_1, p_2, \dots, p_n\} = F(q)$$

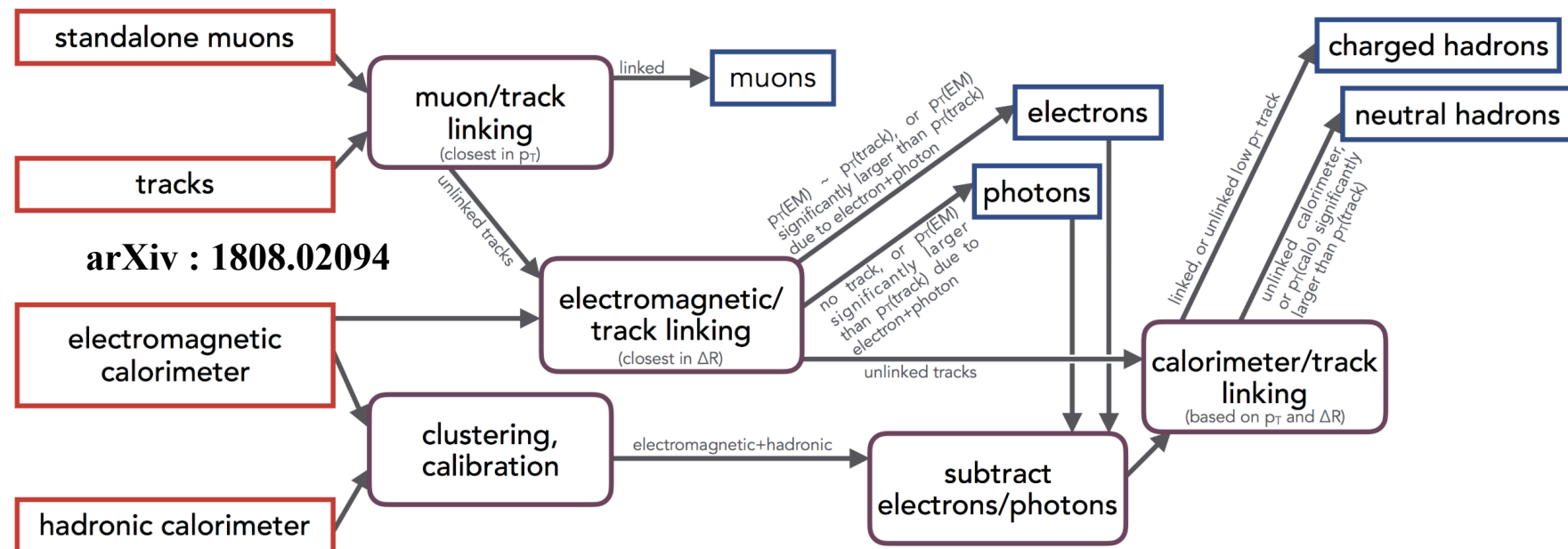
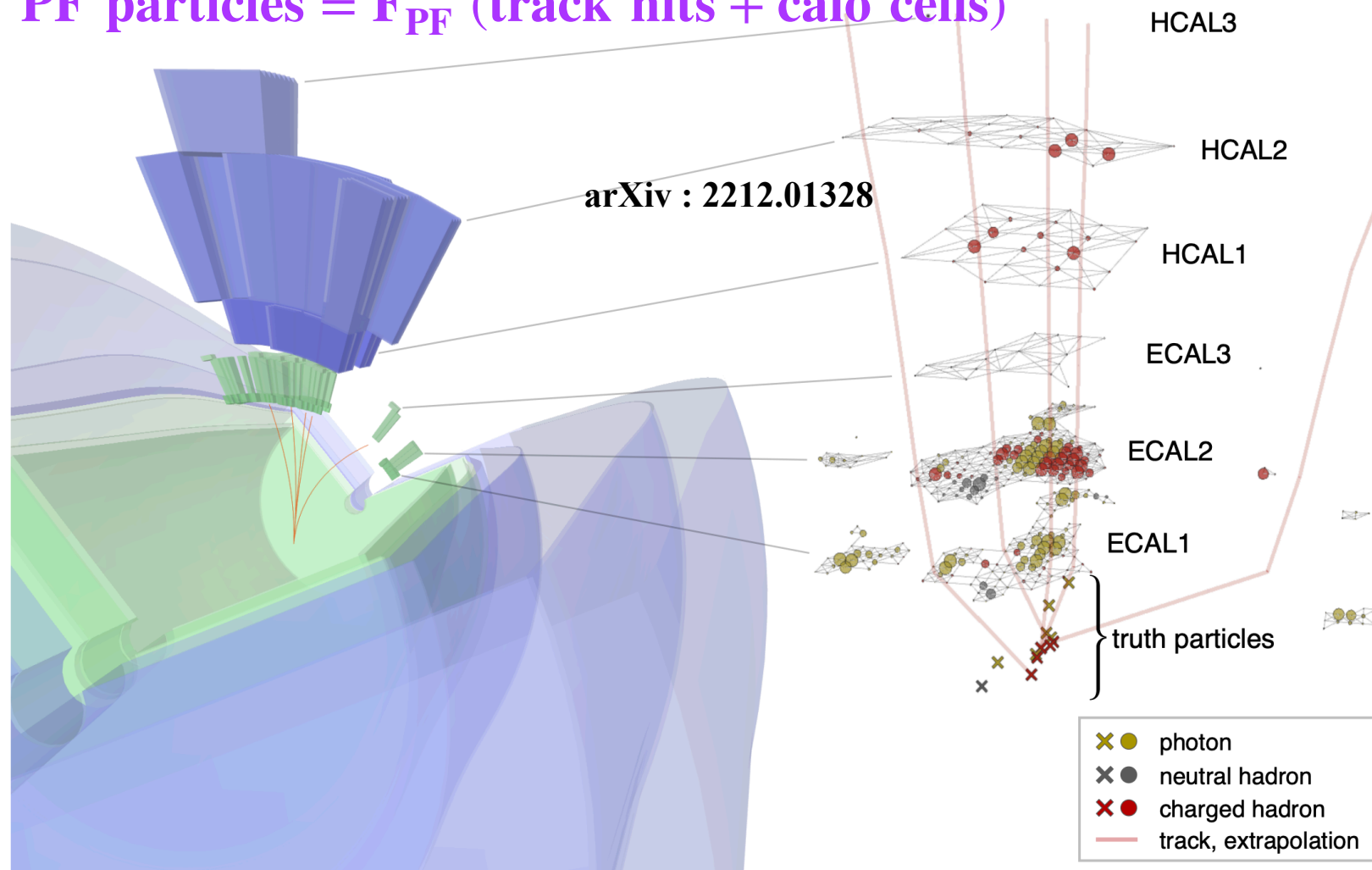
The forward problem is not computable from first principle

The question of jet tagging is how do we define the inverse problem?

$$q = F^{-1}\left(\{p_1, p_2, \dots, p_n\}\right) ?$$

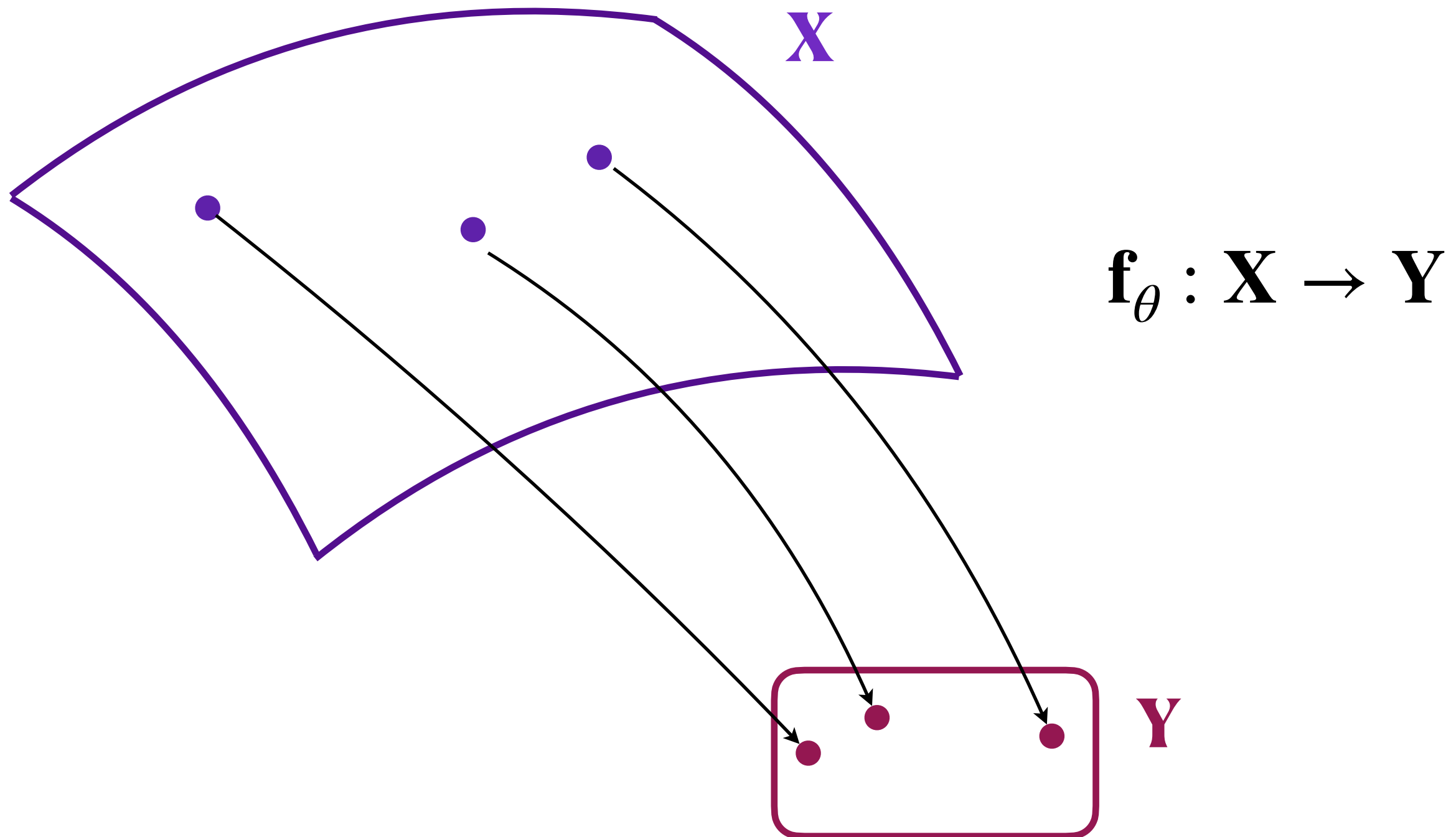
The PF reconstruction algorithm

PF particles = F_{PF} (track hits + calo cells)



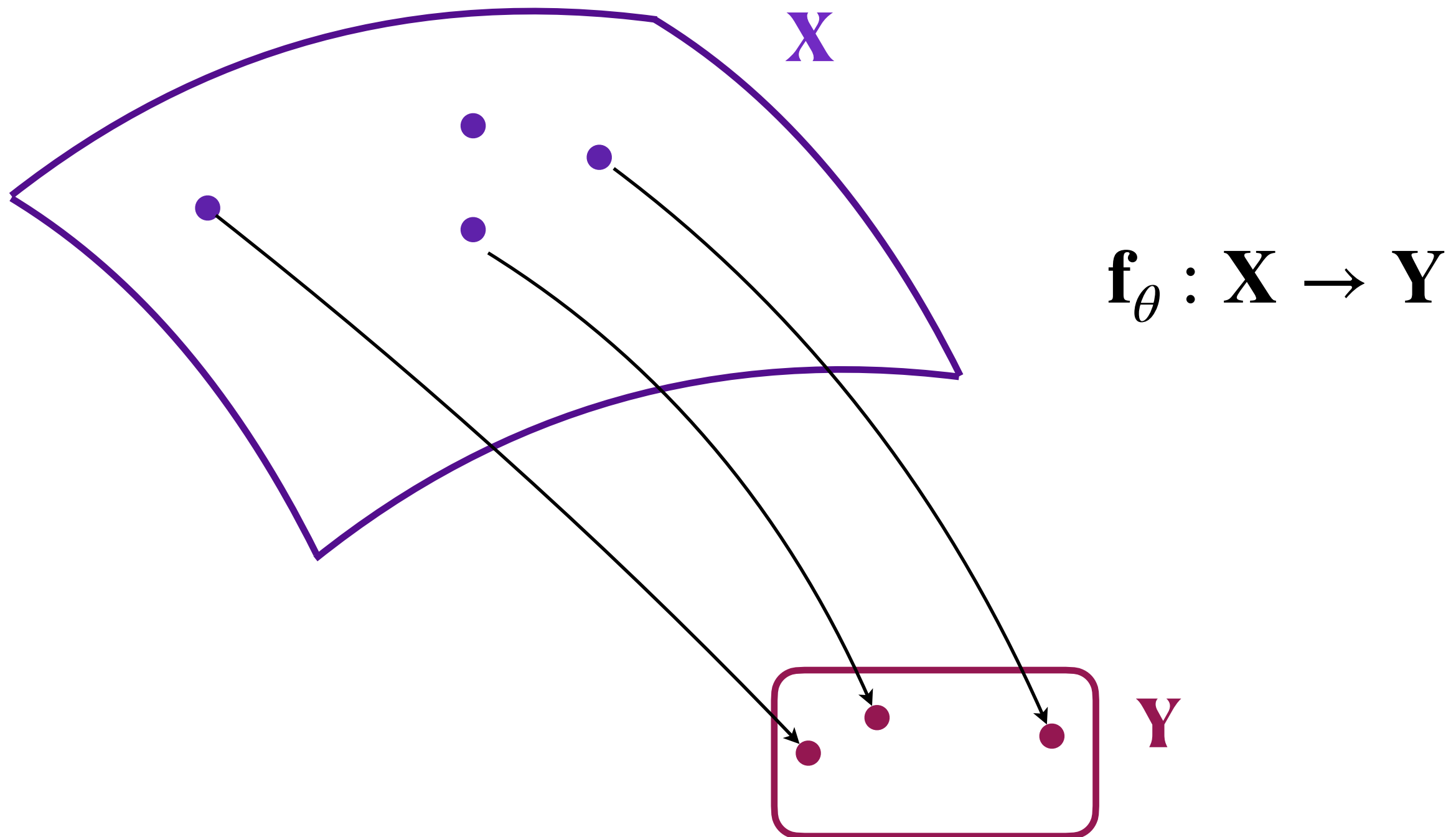
Supervised learning

We have a labelled dataset in hand : $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), (\mathbf{x}_3, \mathbf{y}_3) \dots$



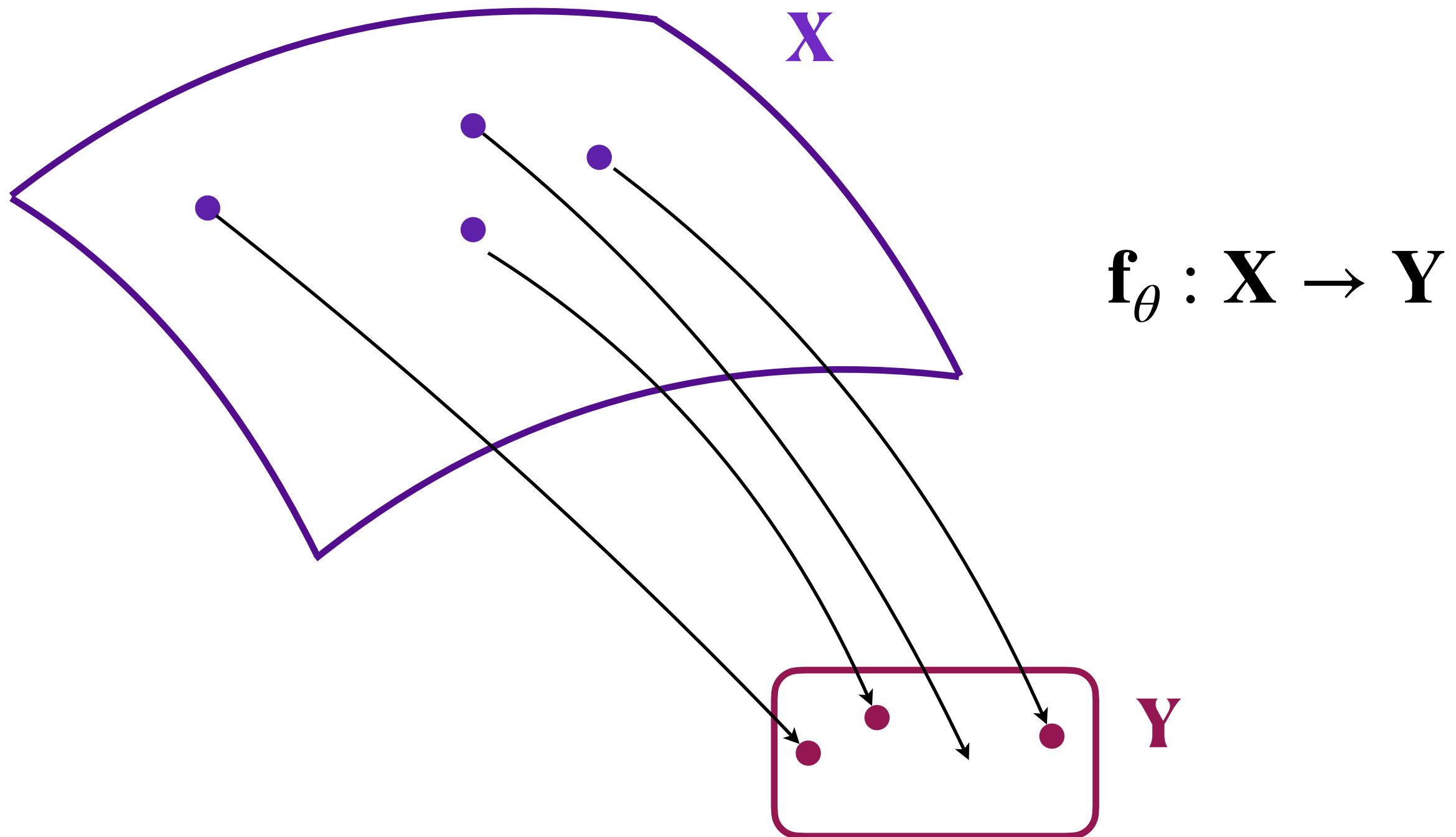
Supervised learning

We have a labelled dataset in hand : $(\mathbf{x}_1, \mathbf{y}_1)$, $(\mathbf{x}_2, \mathbf{y}_2)$, $(\mathbf{x}_3, \mathbf{y}_3)$...



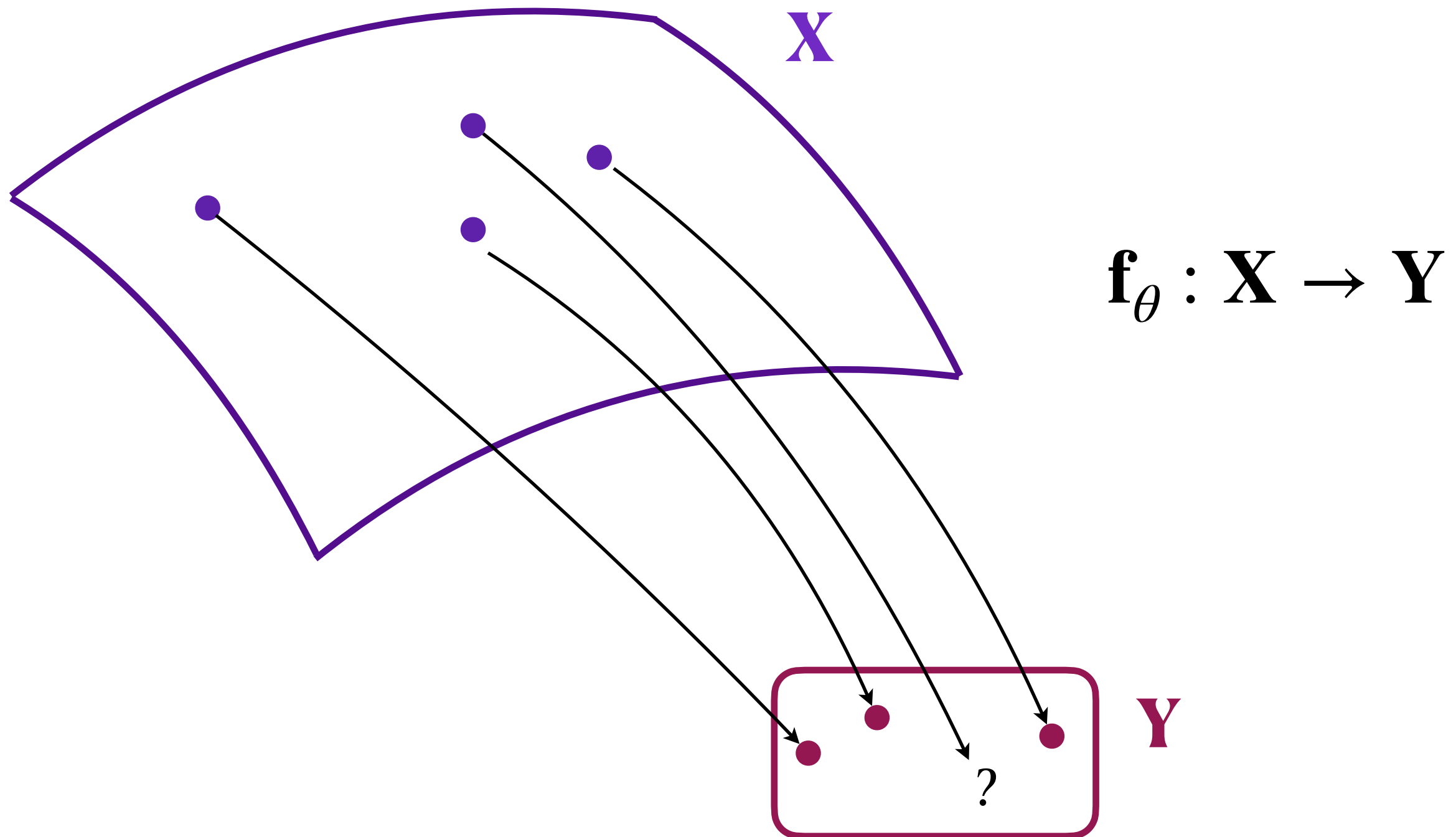
Supervised learning

We have a labelled dataset in hand : $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), (\mathbf{x}_3, \mathbf{y}_3) \dots$

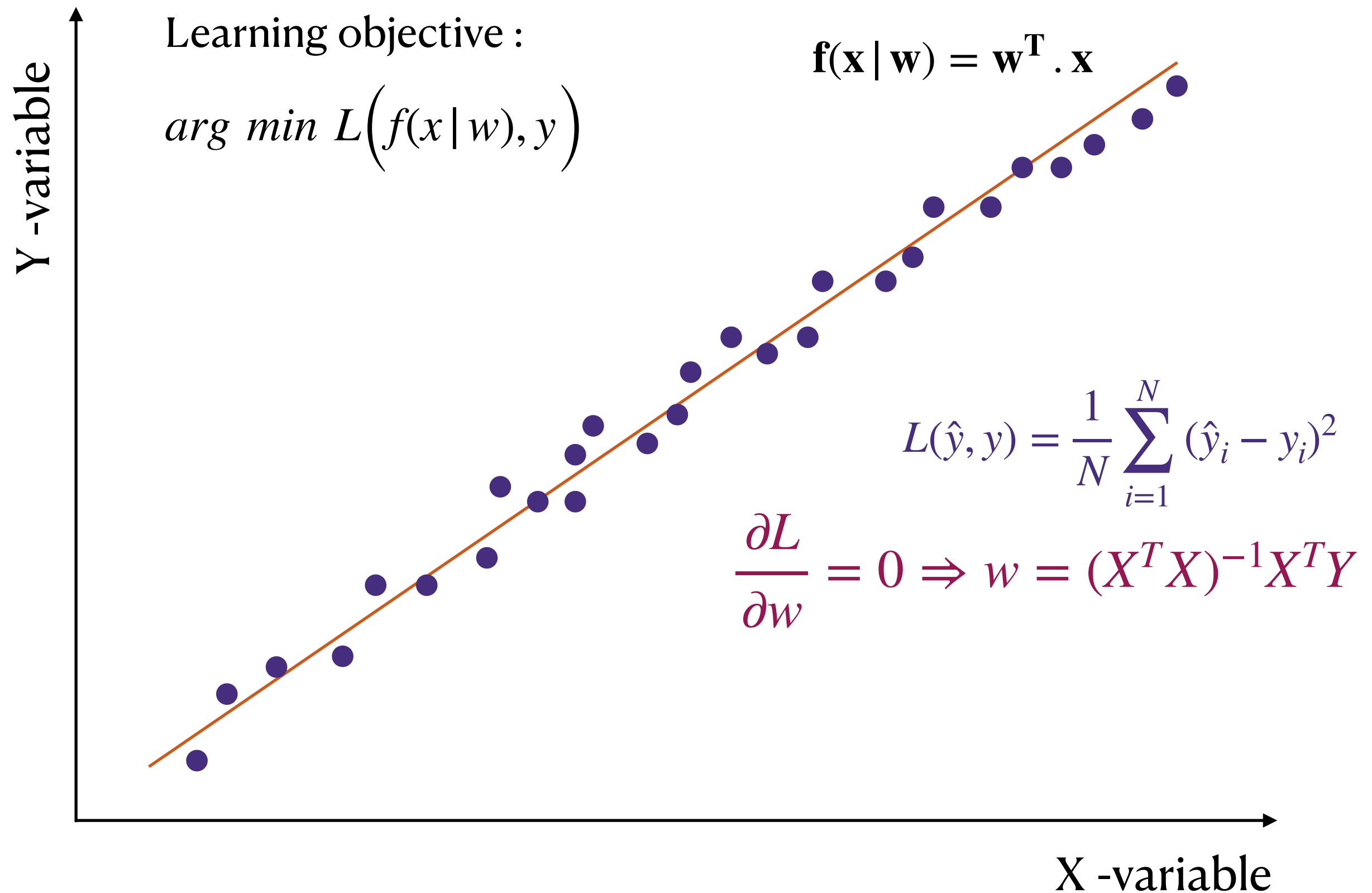


Supervised learning

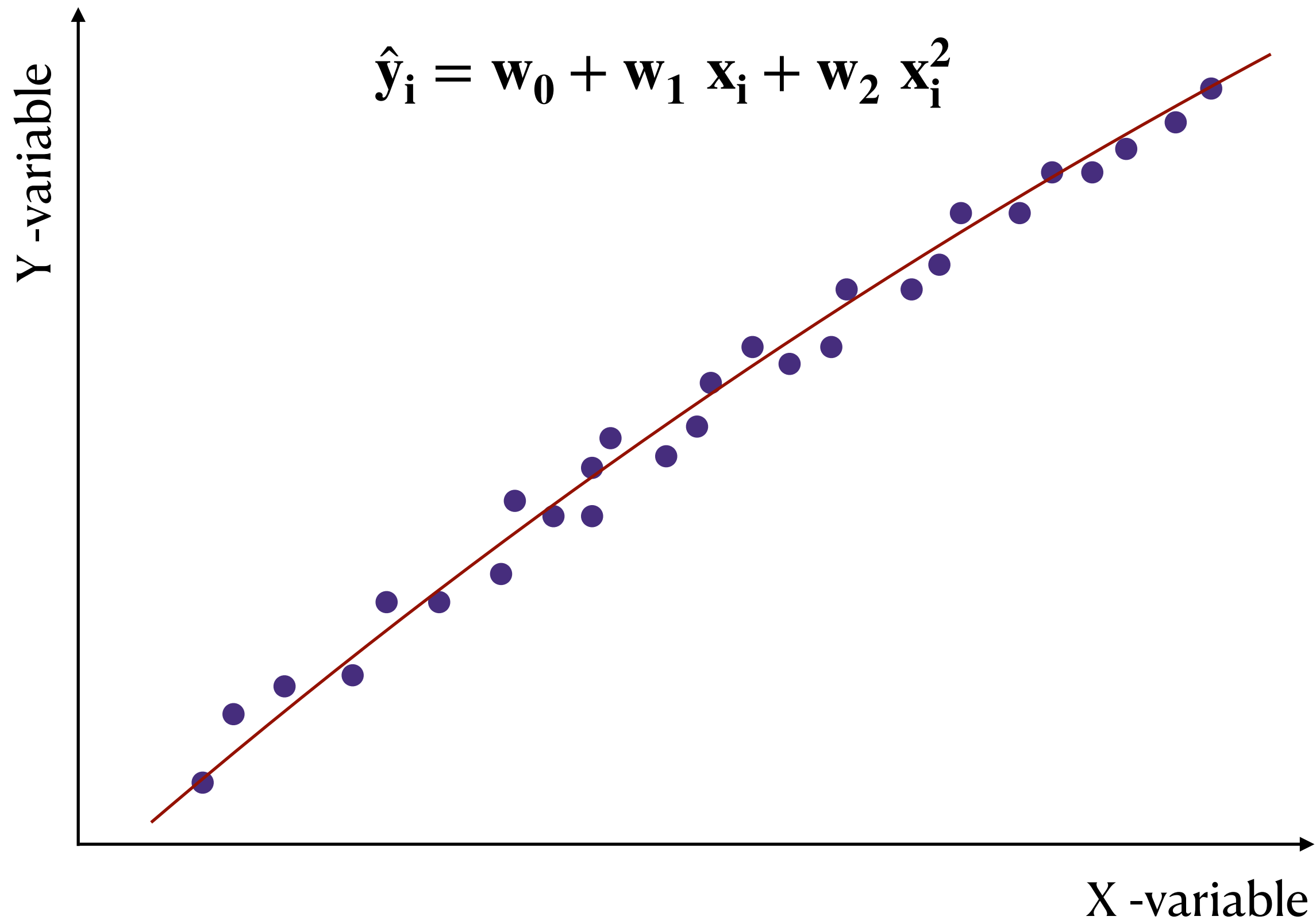
We have a labelled dataset in hand : $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), (\mathbf{x}_3, \mathbf{y}_3) \dots$



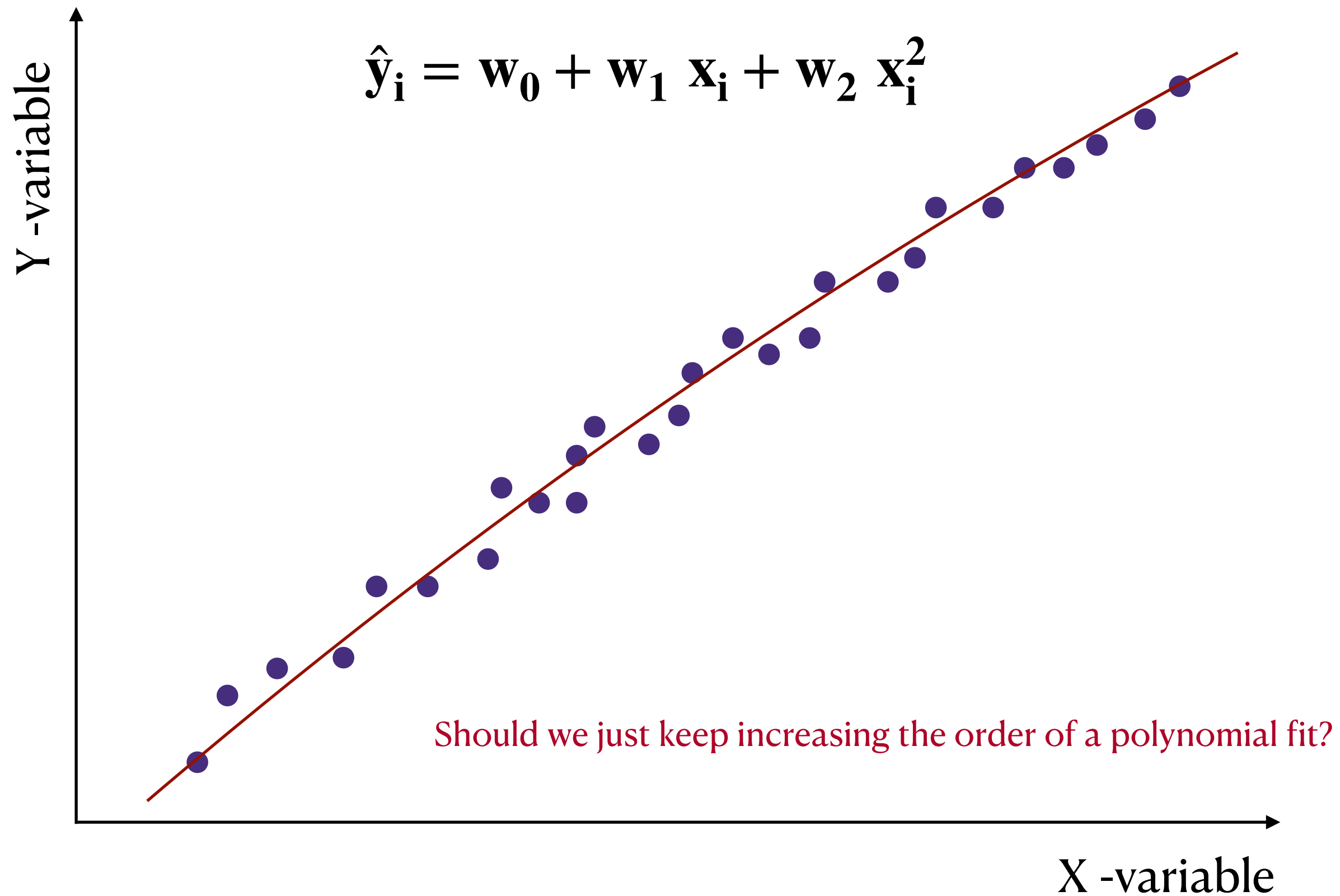
Let's learn through an example



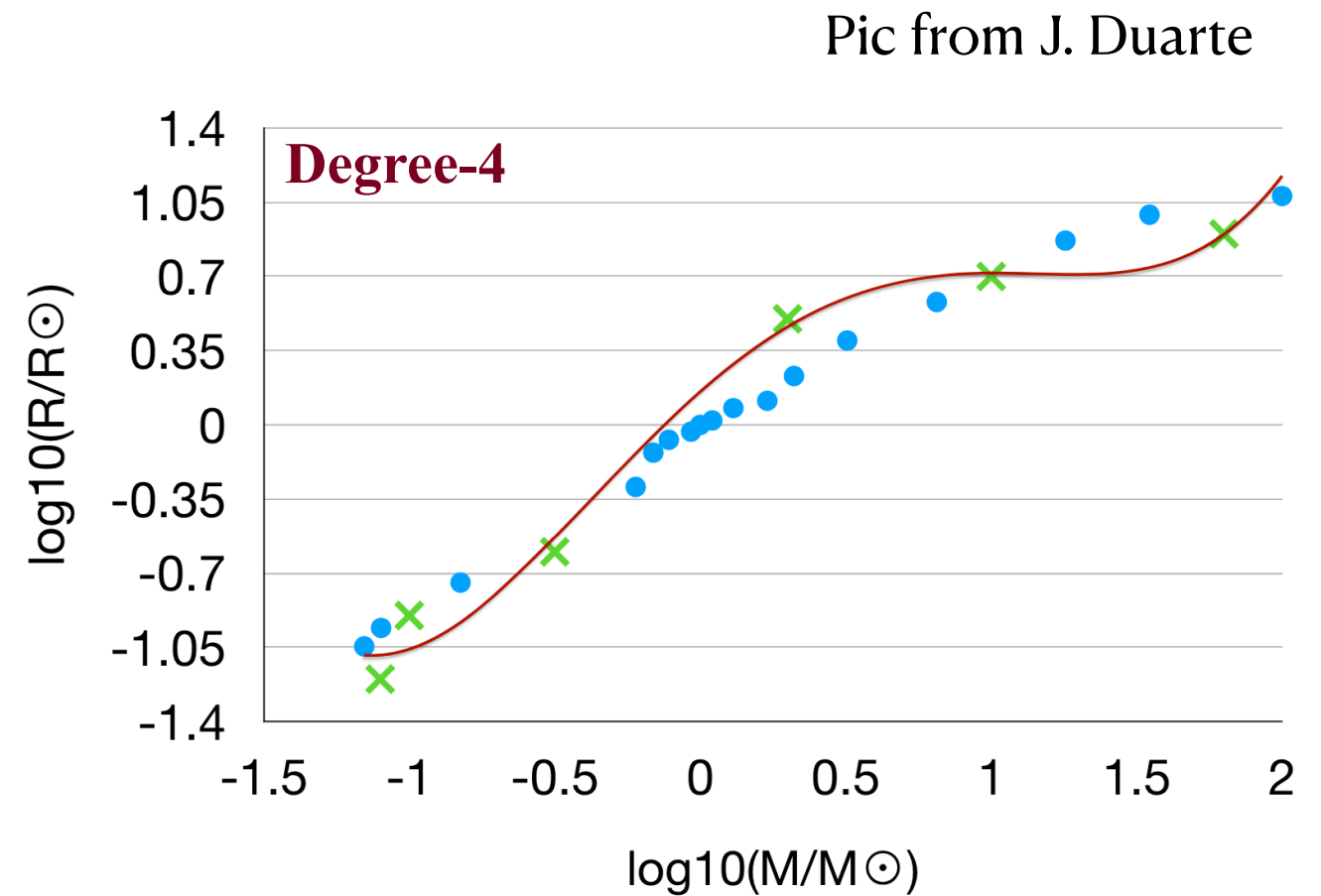
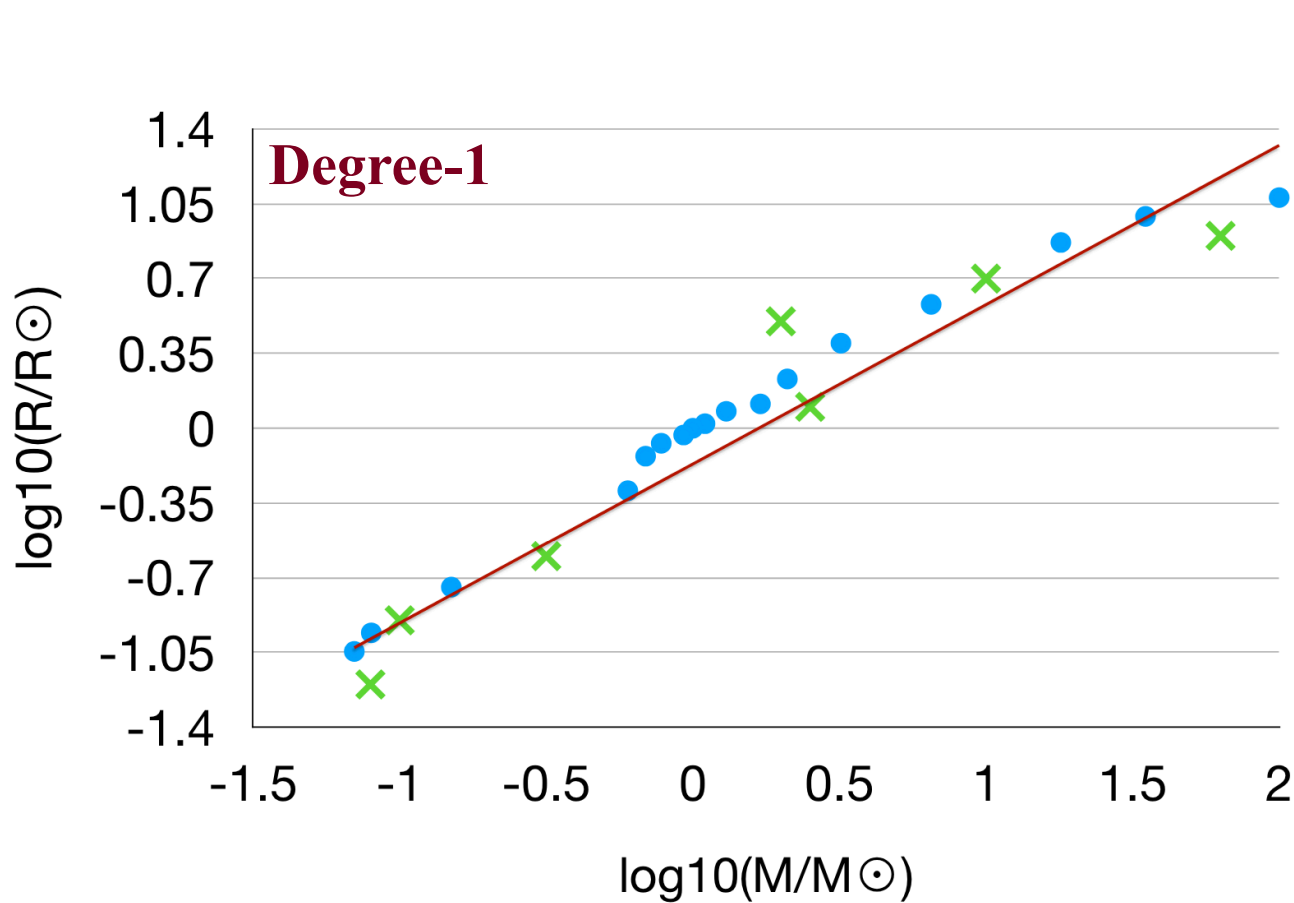
Let's learn through an example



Let's learn through an example

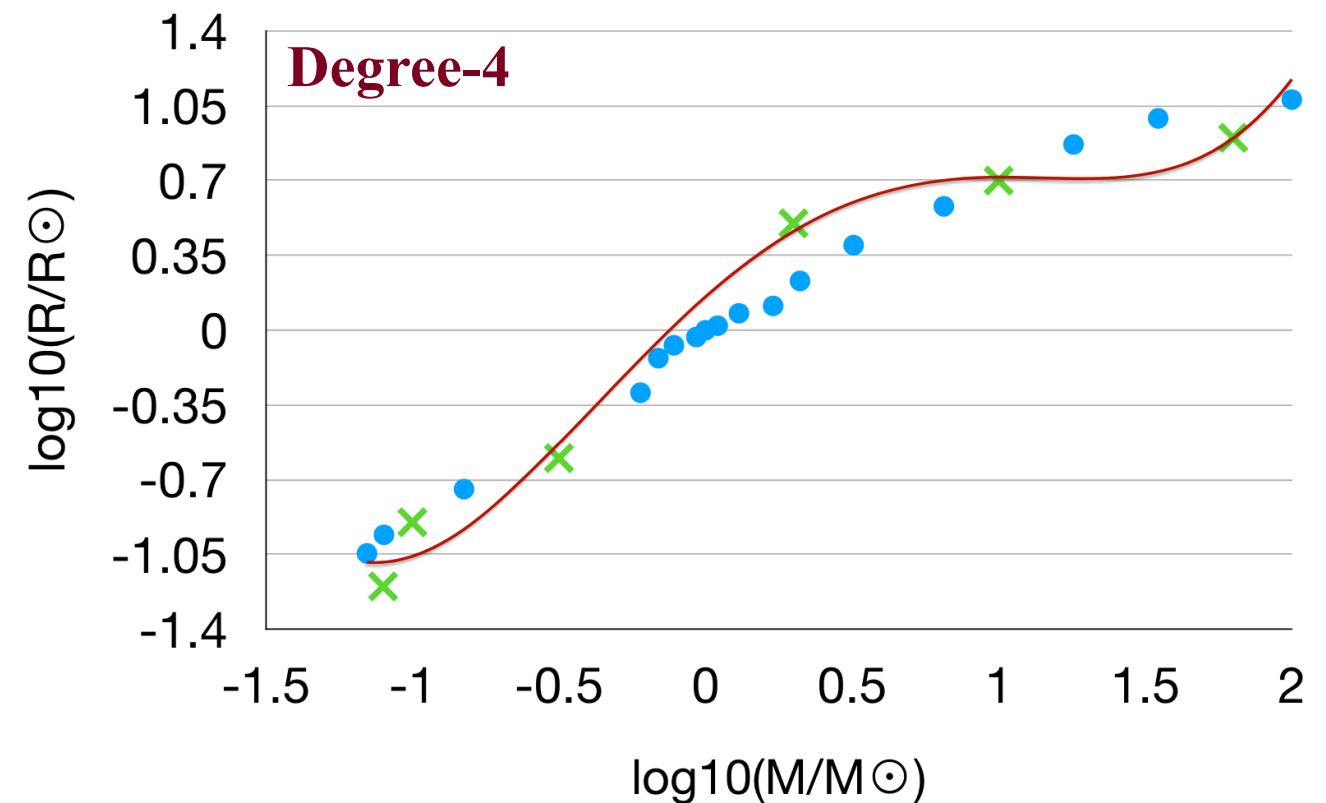
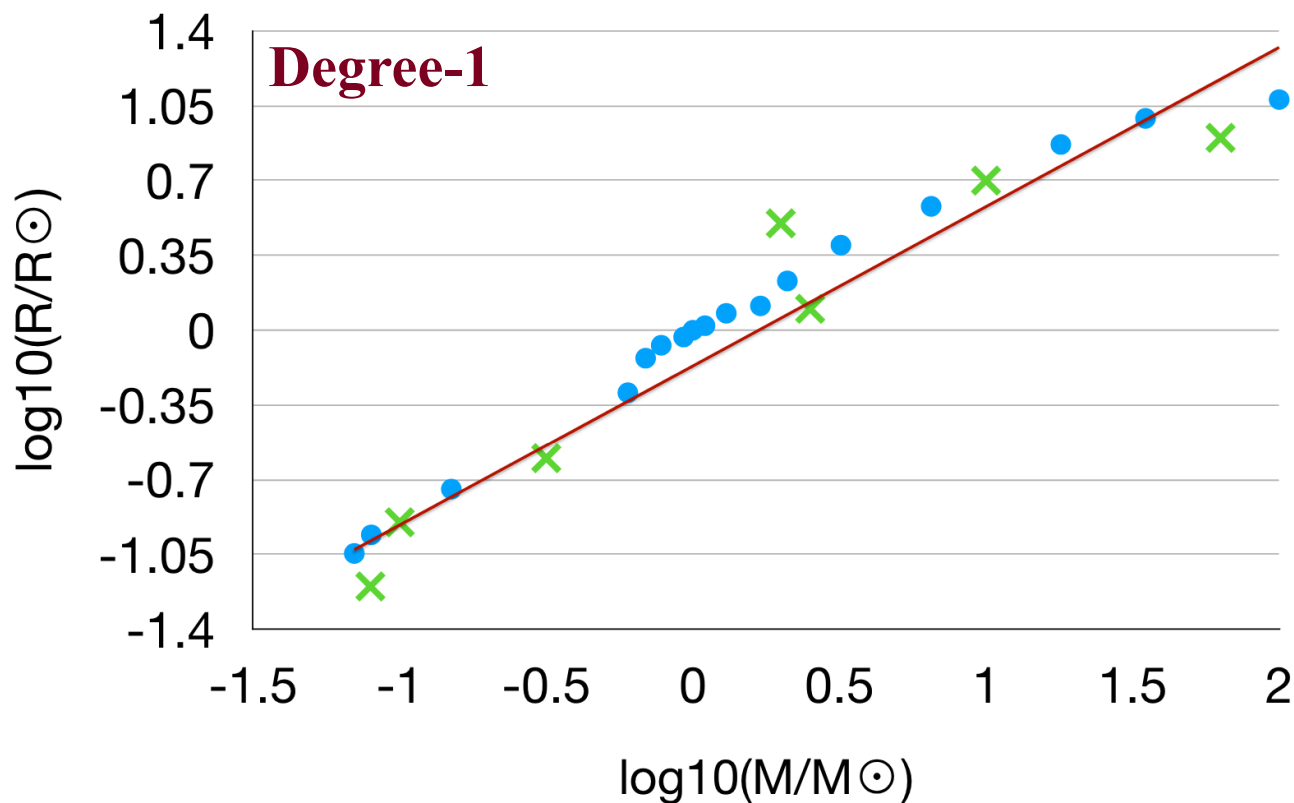


The issue is a model doesn't generalize



The issue is a model doesn't generalize

Pic from J. Duarte



Will a polynomial always work?

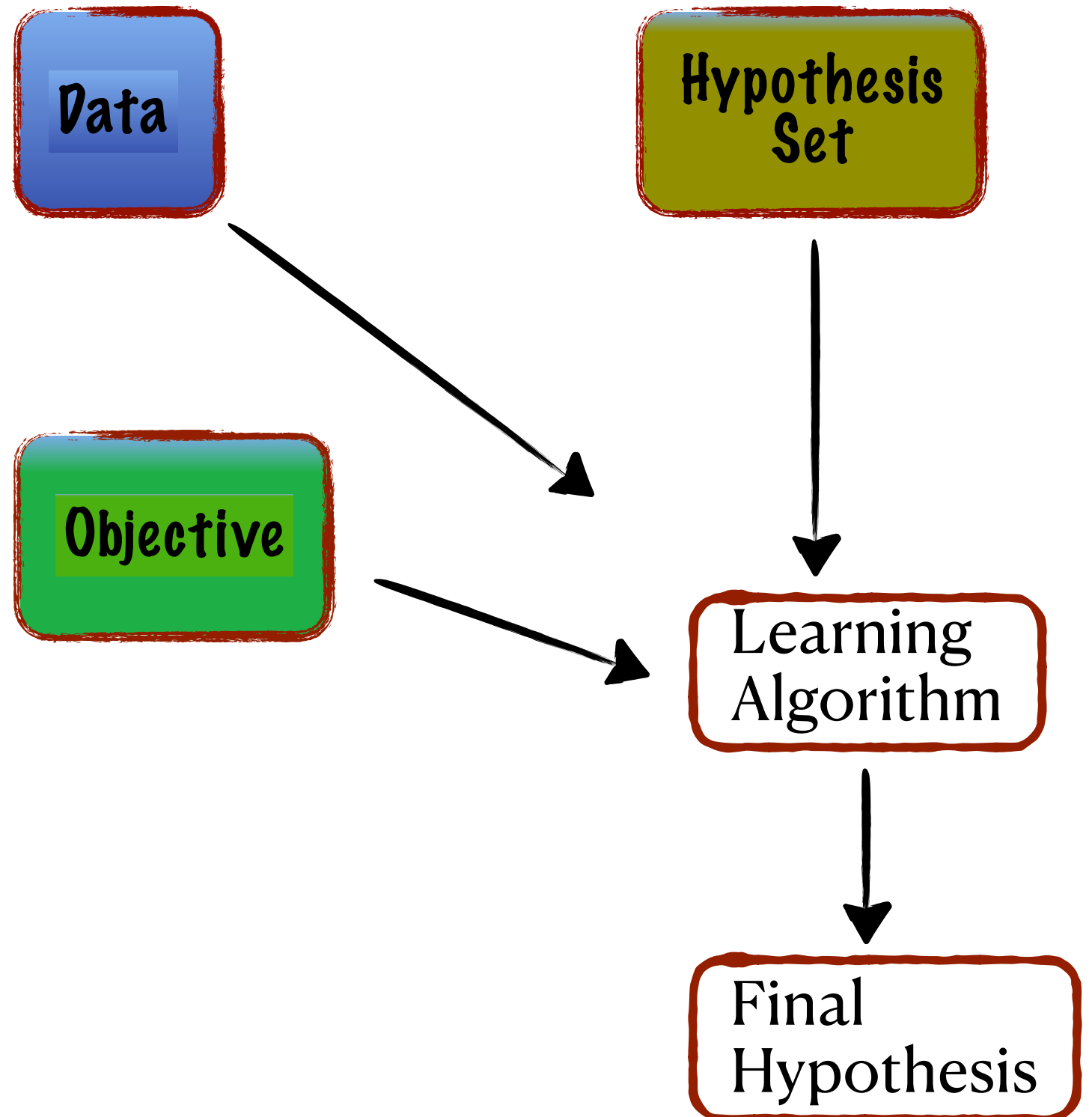
What's the recipe for writing an inductive bias ?

When do we know if we are overfitting?

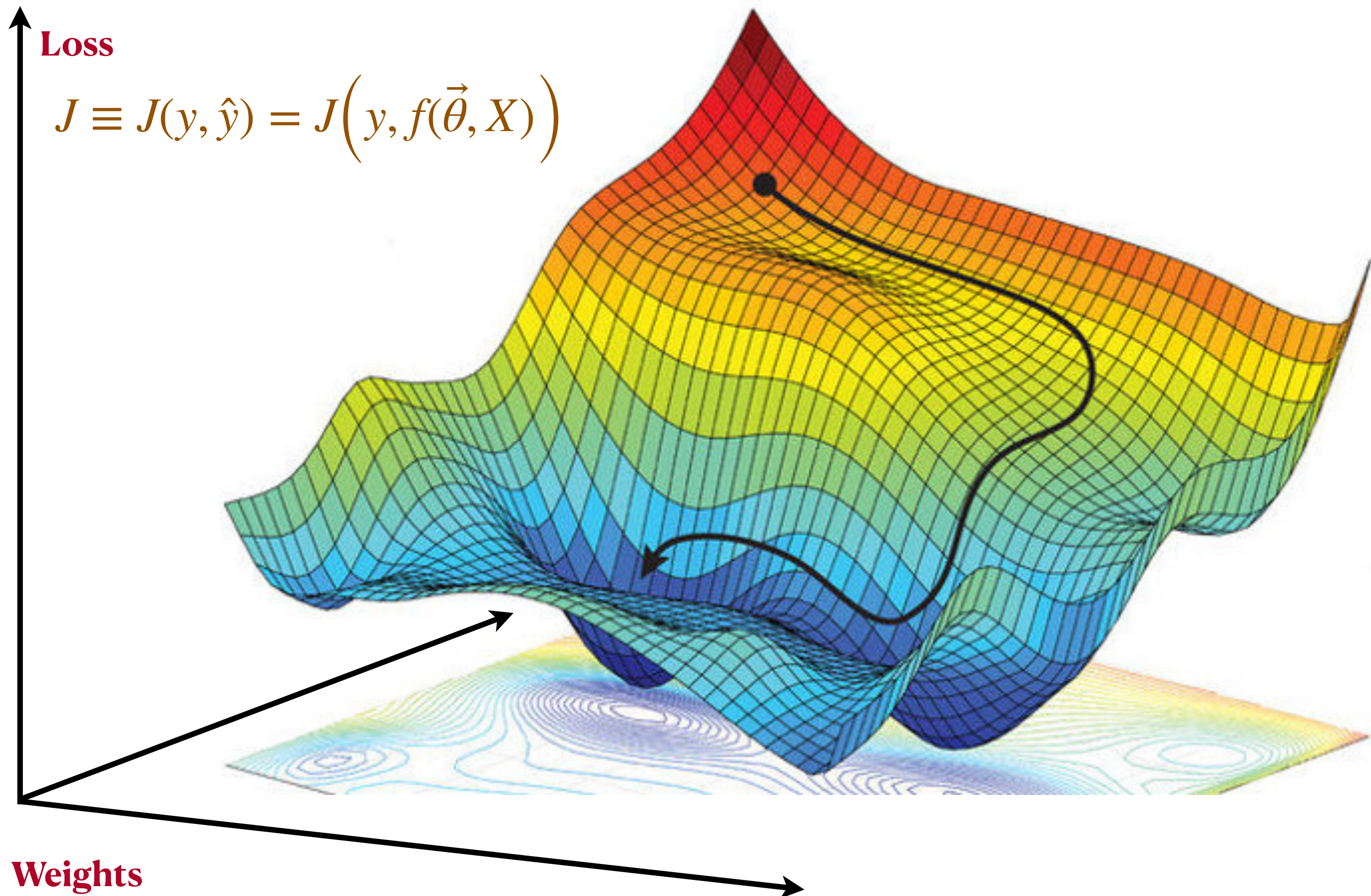
How do we club all of it inside a general principle of statistical learning?

Statistical learning : the big picture

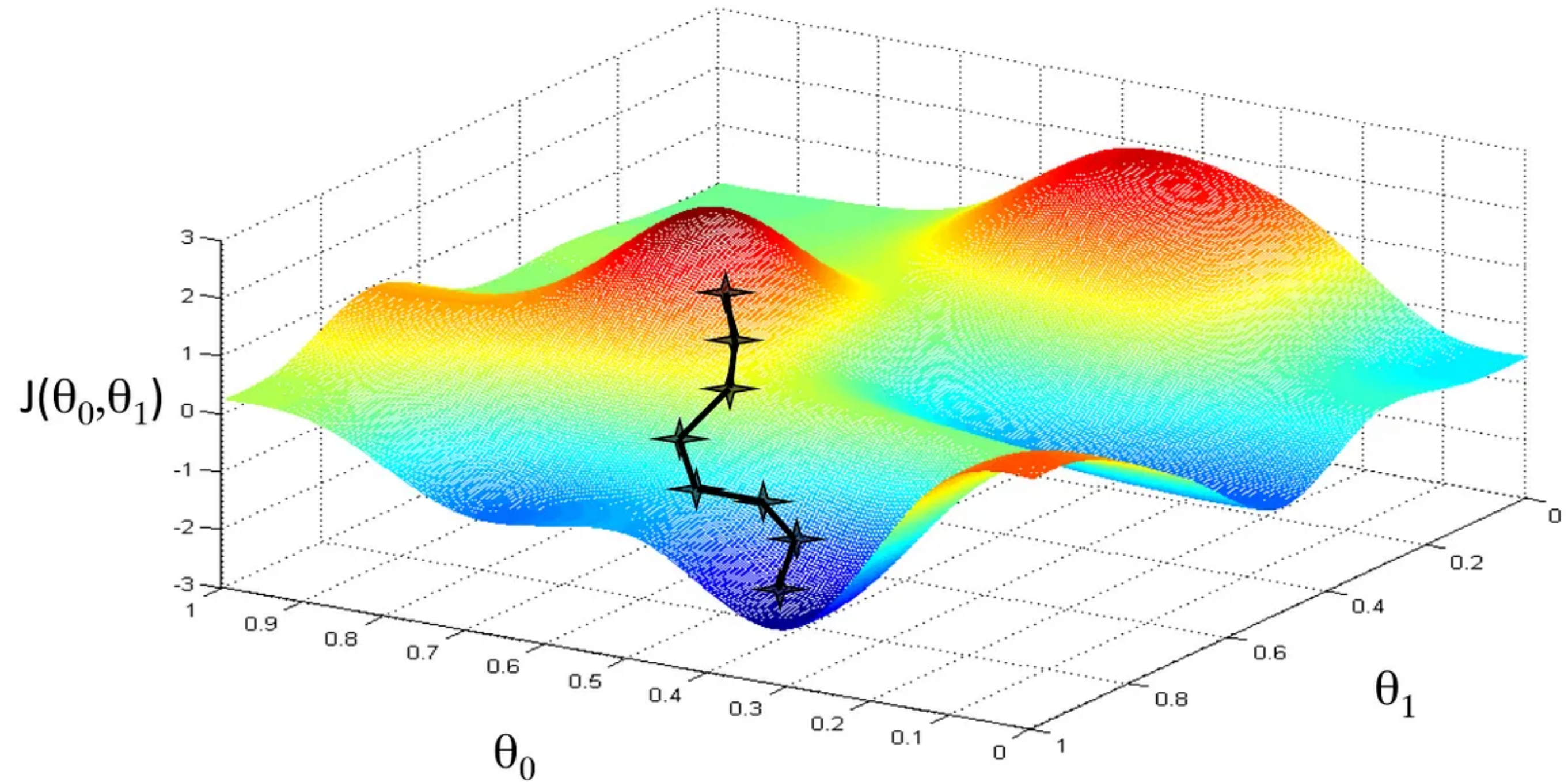
- ✓ Understand and prepare the data to be used for training.
- ✓ Propose a search space for hypothesis test. (a.k.a define the algorithm)
- ✓ Define the objective.
Will refer to this as loss function.
- ✓ Provide a learning algorithm to select the best one.



The parameter optimization : gradient descent



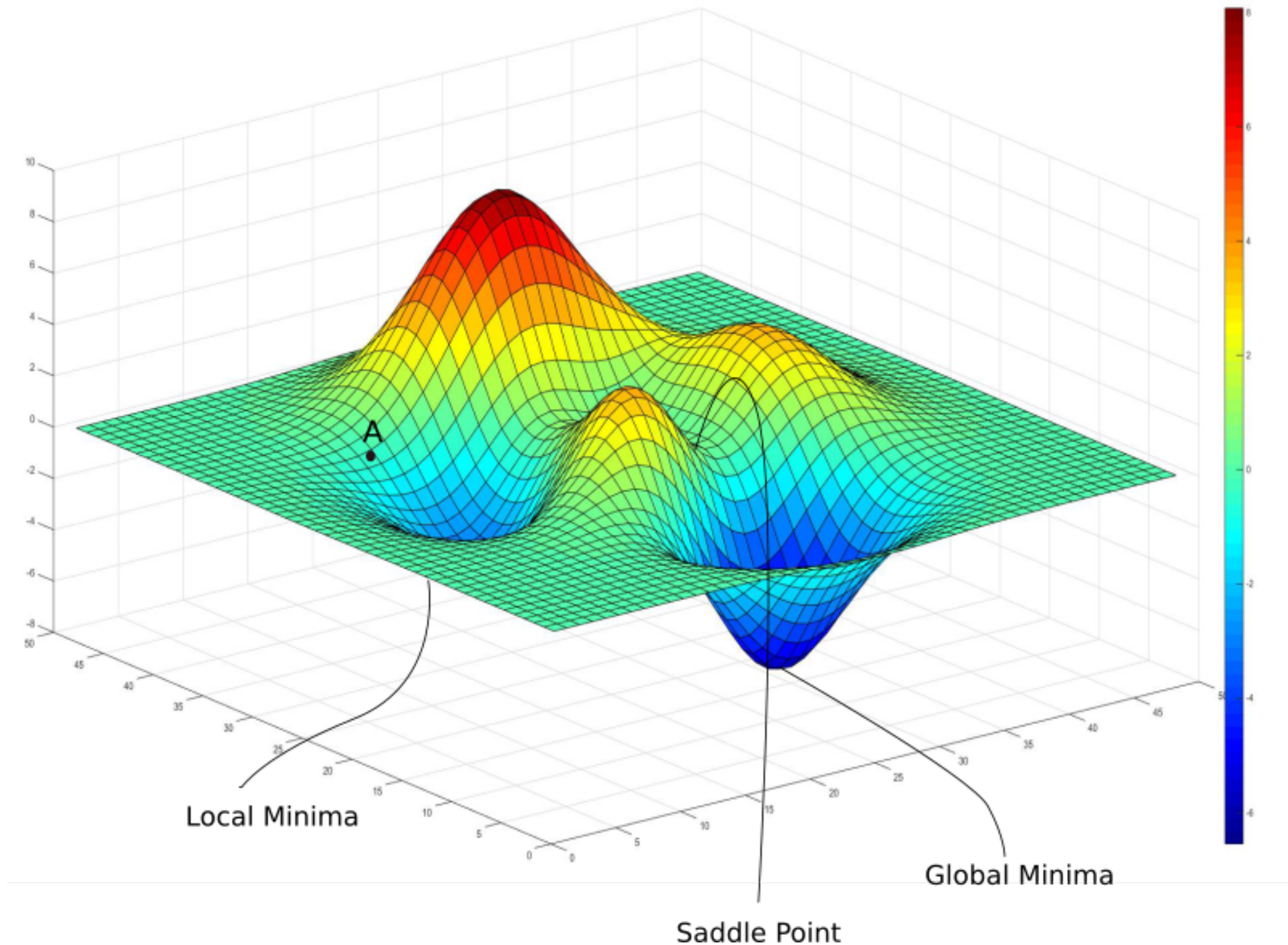
The parameter optimization : gradient descent



$$\theta'_i = \theta_i - \alpha \cdot \frac{\partial J}{\partial \theta_i}$$

Stochastic gradient descent

<https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/>



What if the network converges to a local minima?

Stochastic gradient descent

<https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/>

Repeat Until Convergence {
$$\omega_j \leftarrow \omega_j - \alpha * \nabla_w \sum_1^m L_m(w) . \hat{w}_j$$

Gradient descent
}

Stochastic Gradient descent

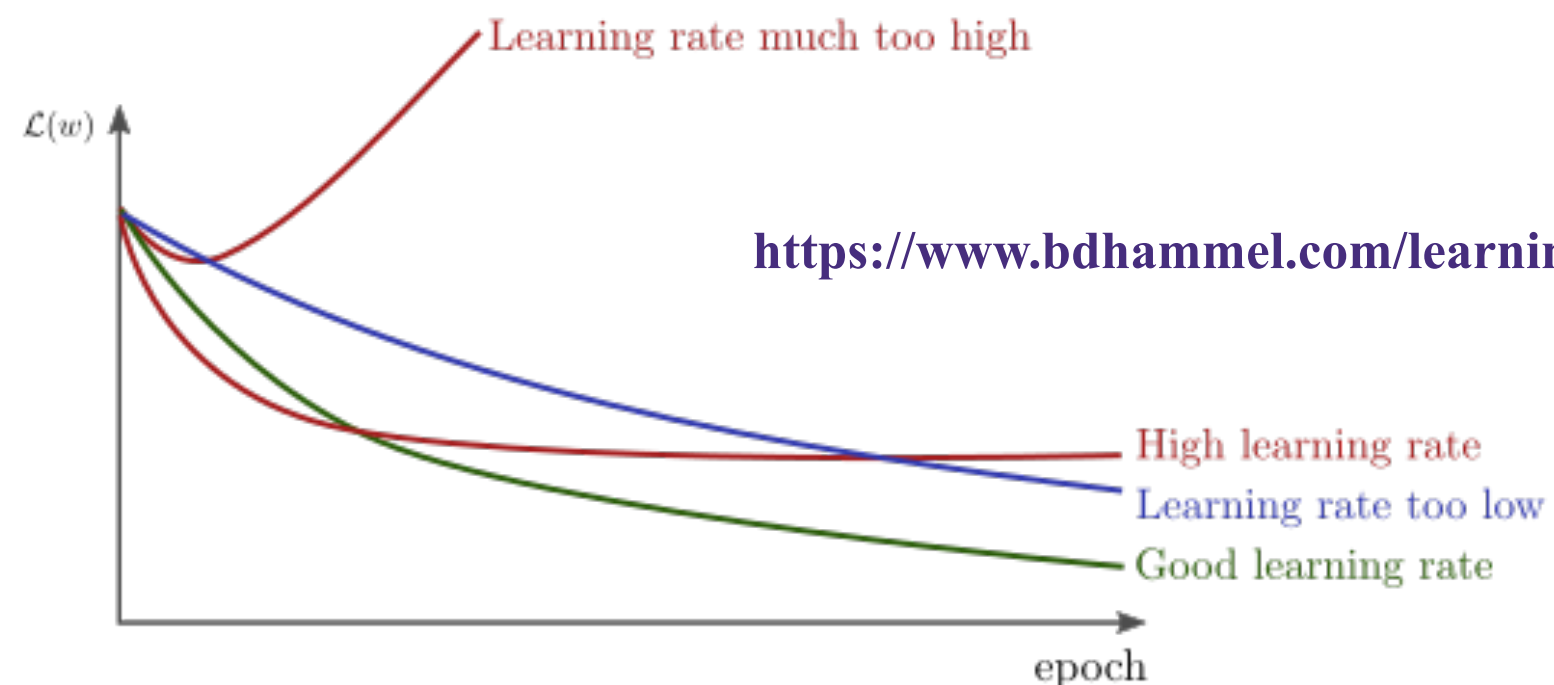
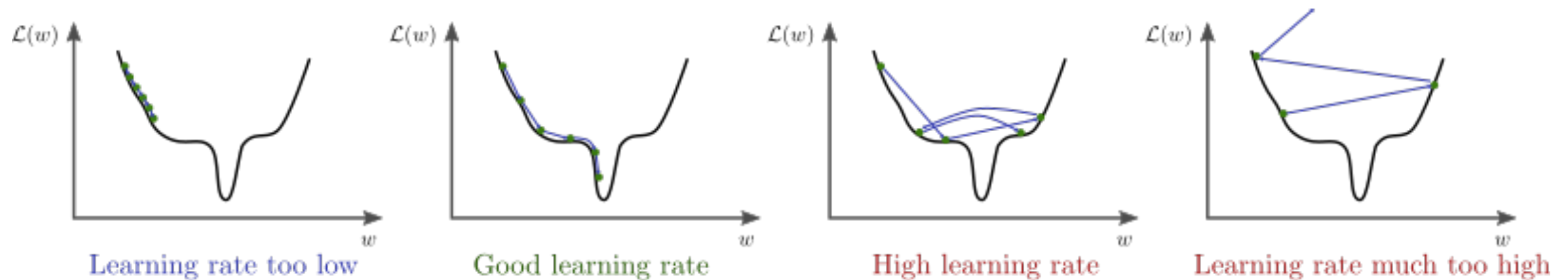
Repeat Until Convergence {
 for i = 1...m {

$$\omega \leftarrow \omega - \alpha * \nabla_w L_m(w)$$

 }
}

Few words about learning rates

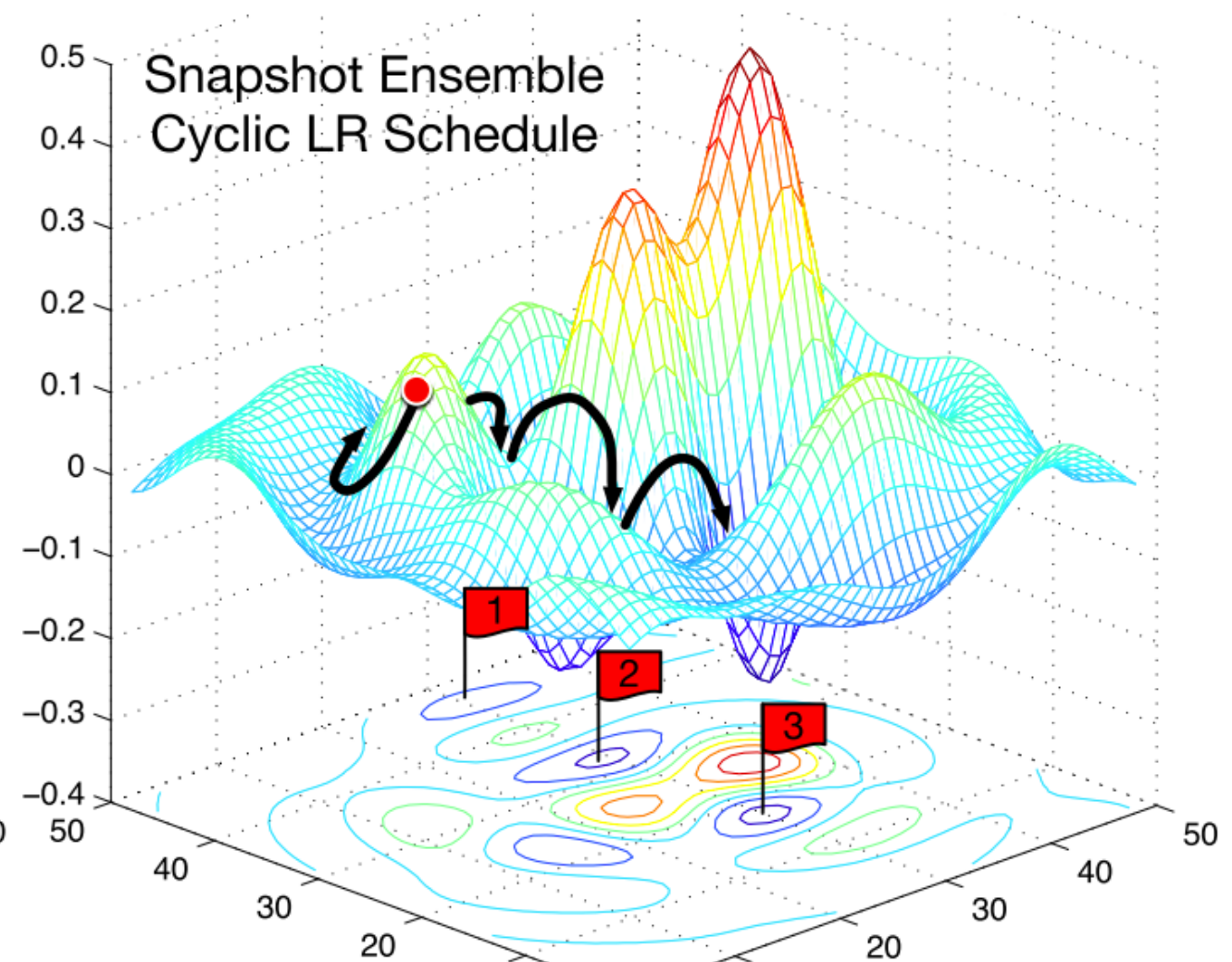
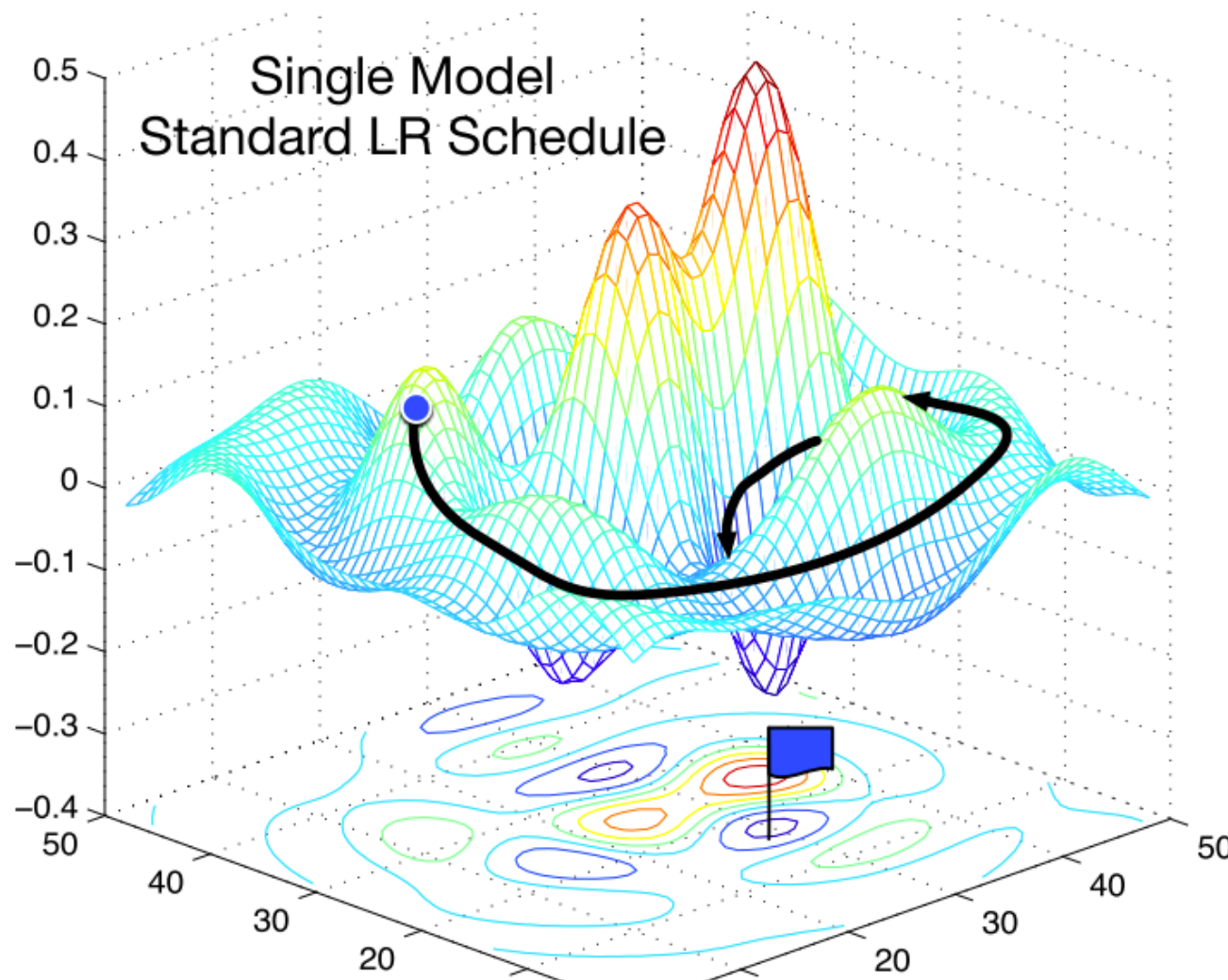
$$w_{t+k} = w_t - \eta \frac{1}{n} \sum_{j < k} \sum_{x \in B} \nabla l(x, w_{t+j})$$
$$\hat{w}_{t+k} = w_t - \hat{\eta} \frac{1}{kn} \sum_{j < k} \sum_{x \in B} \nabla l(x, w_{t+j})$$



<https://www.bdhammel.com/learning-rates/>

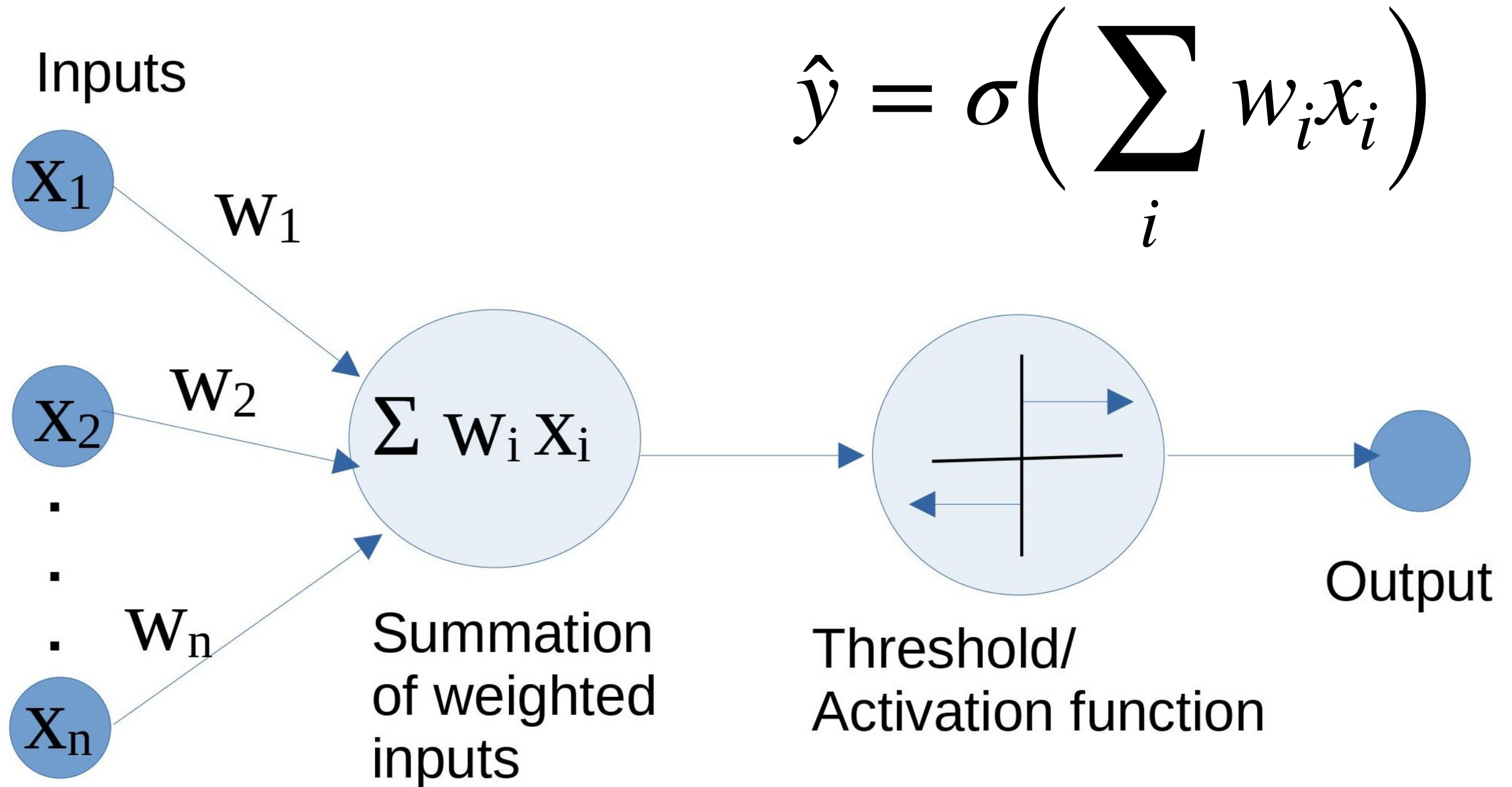
Few words about learning rates

$$\eta = d_{\text{model}}^{-0.5} \cdot \min(\text{step_num}^{-0.5}, \text{step_num} \cdot \text{warmup_steps}^{-1.5})$$



<https://www.ruder.io/deep-learning-optimization-2017/>

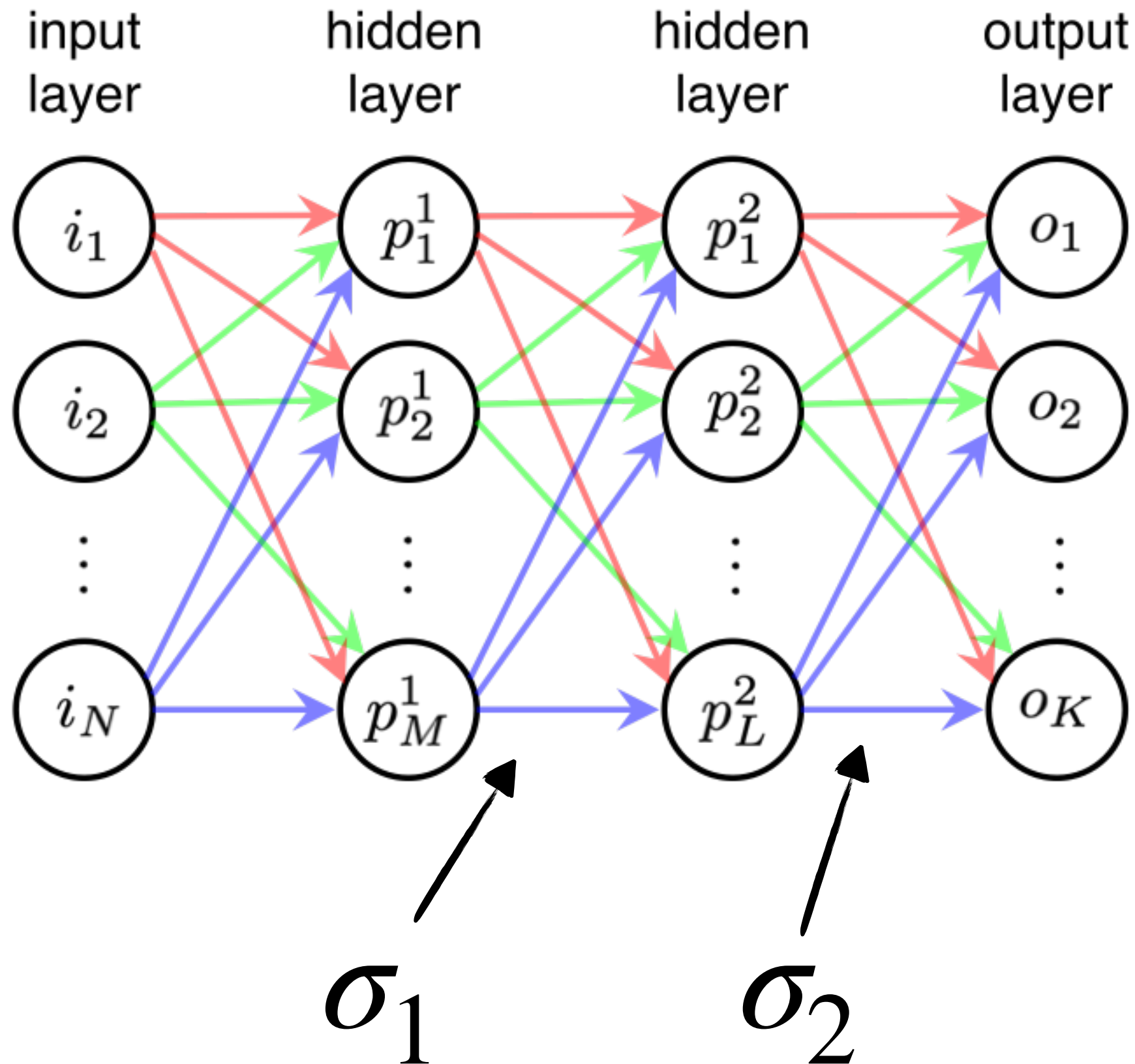
What's the strategy to fit general functions?



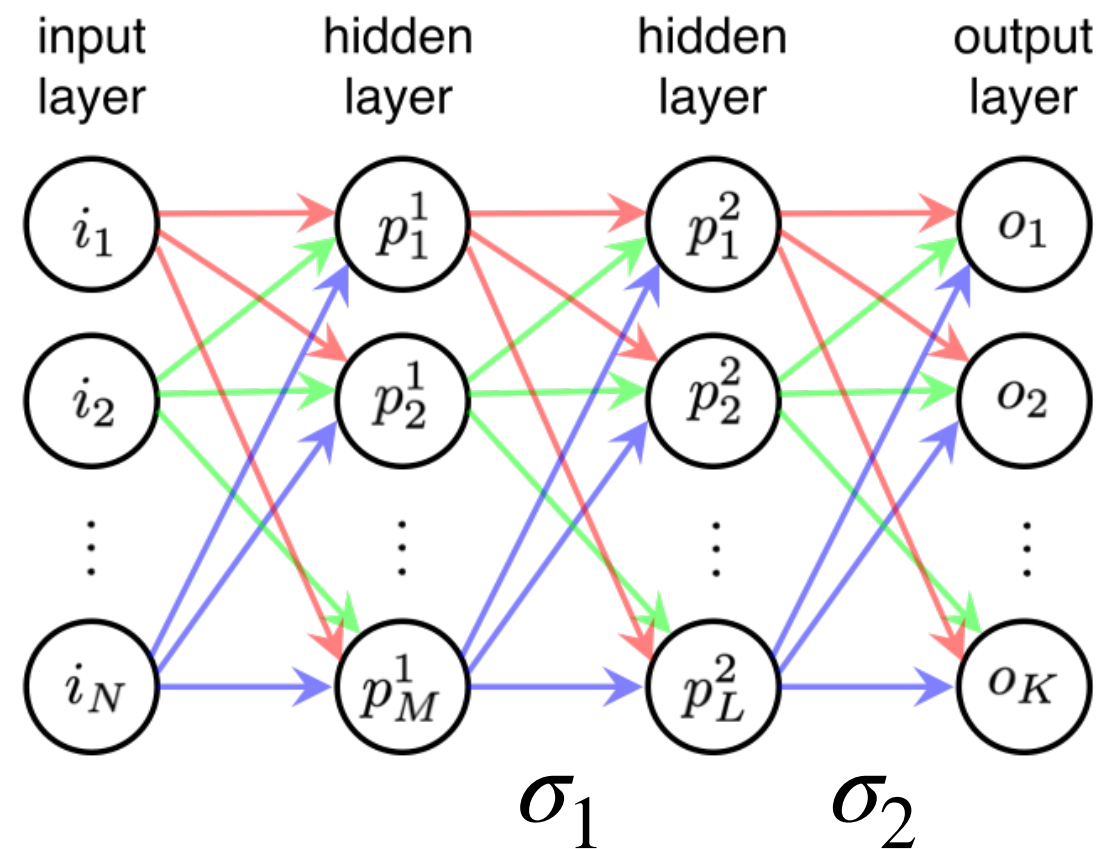
A single layer perceptron

<https://www.analyticsvidhya.com/blog/2021/06/beginners-guide-to-universal-approximation-theorem/>

What's the strategy to fit general functions?



What's the strategy to fit general functions?

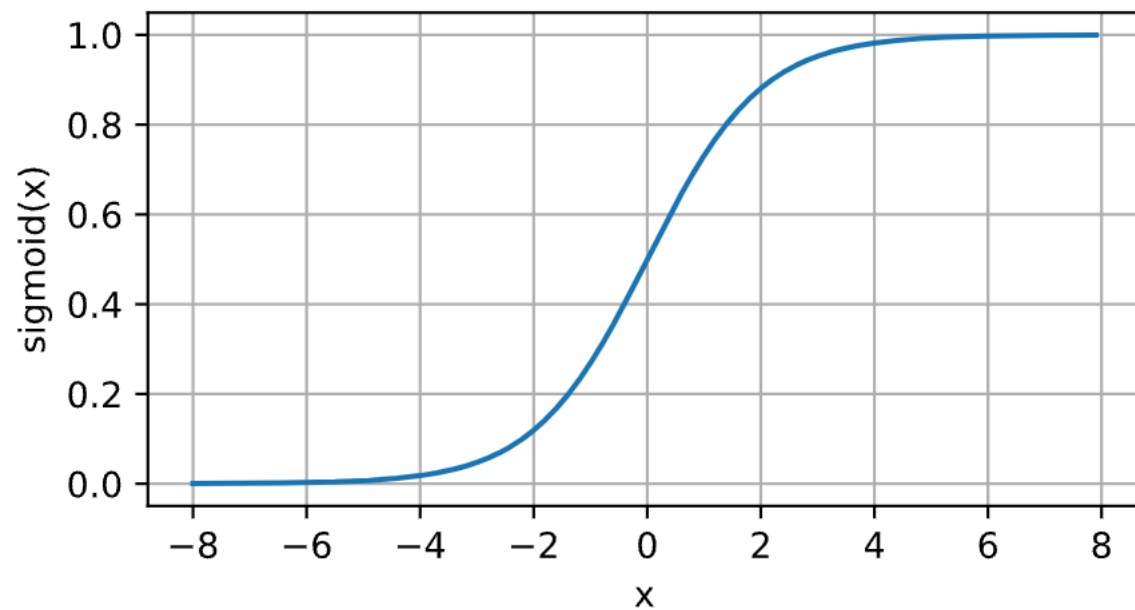


$$p_{\alpha}^1 = \sum_{\beta} W_{\alpha\beta}^1 i_{\beta} + b_{\alpha}^1 \quad p_{\alpha}^1 = \sigma_1 (p_{\alpha}^1)$$
$$p_{\alpha}^2 = \sum_{\beta} W_{\alpha\beta}^2 p_{\beta}^1 + b_{\alpha}^2 \quad p_{\alpha}^2 = \sigma_2 (p_{\alpha}^2)$$

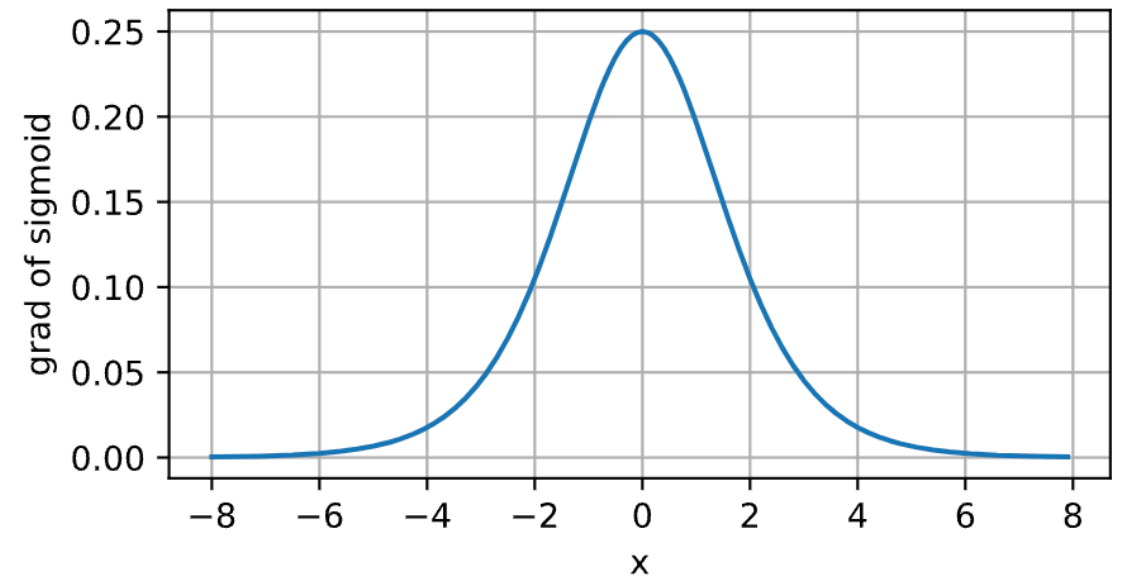
Dimensions of the features changes.

A thing or two about activation functions

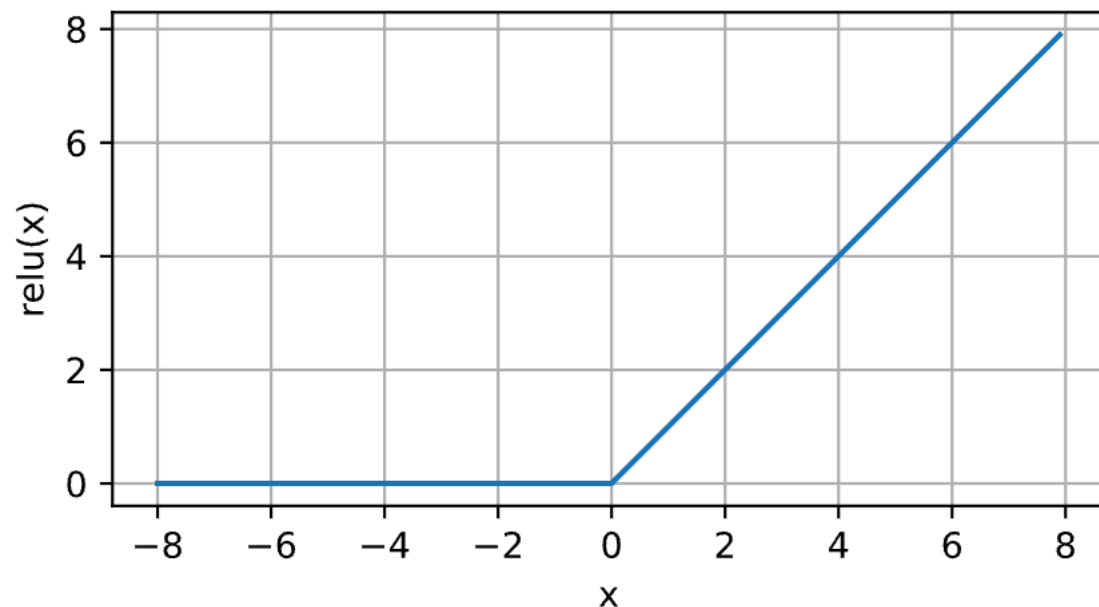
$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}$$



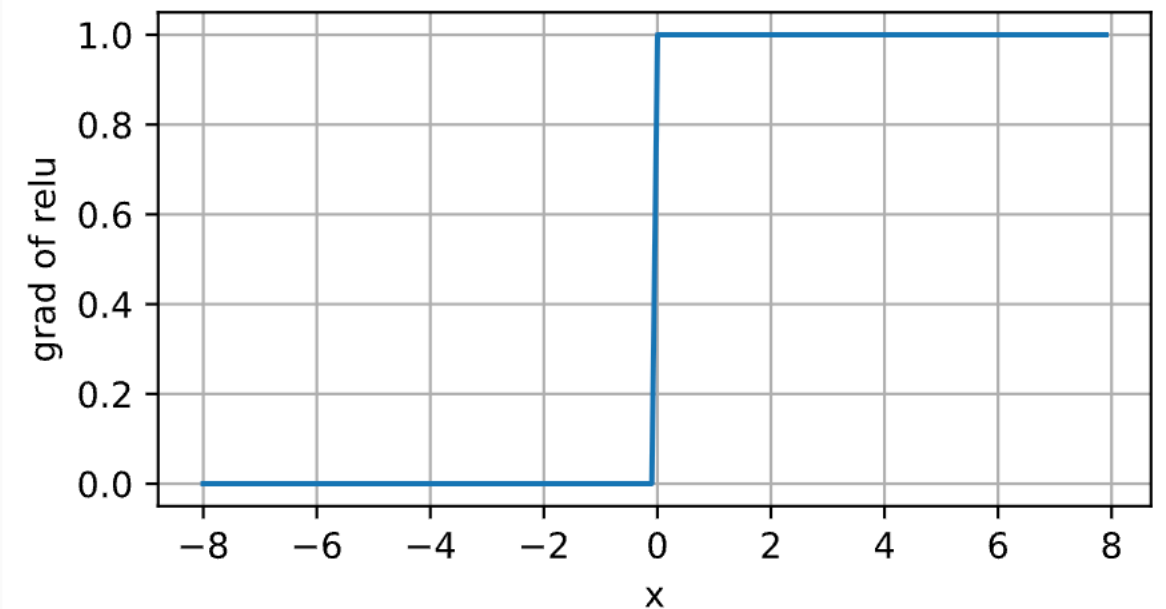
$$\frac{d}{dx} \text{Sigmoid}(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$



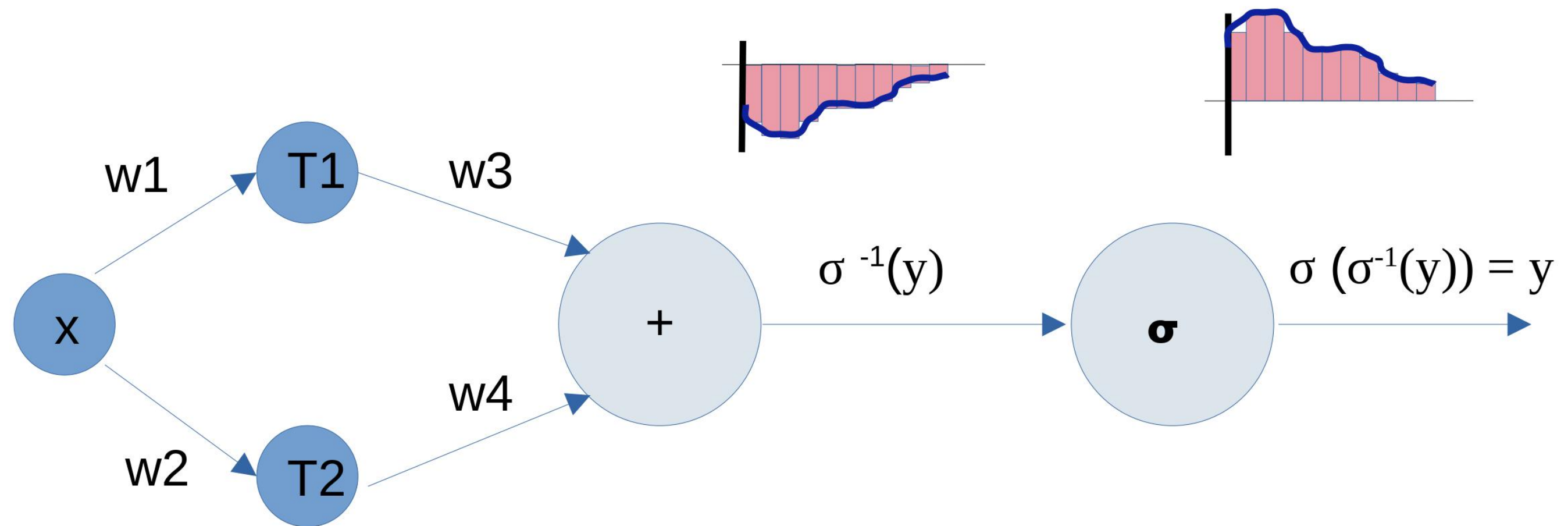
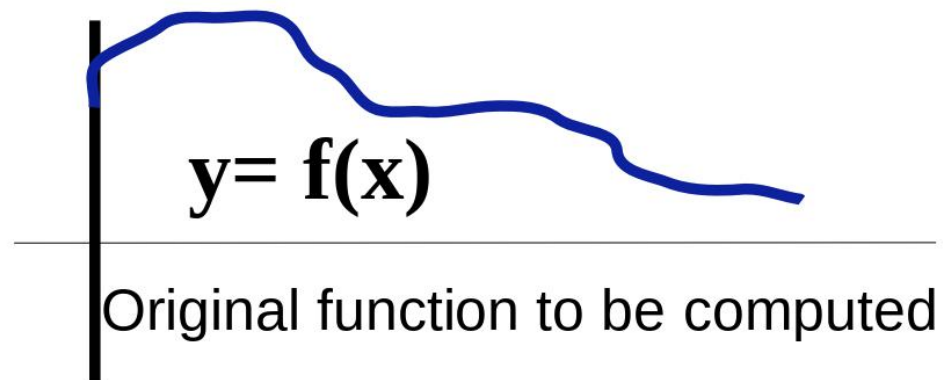
$$\text{ReLU}(x) = \max(x, 0)$$



$$\frac{d}{dx} \text{ReLU}(x) = \Theta(x)$$

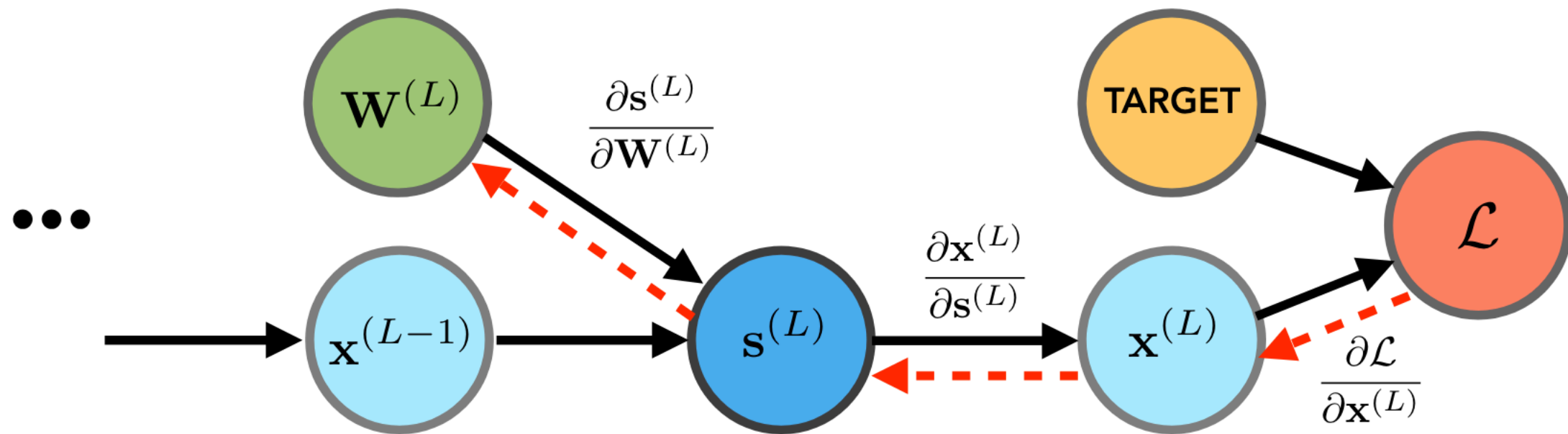


What's the strategy to fit general functions?



<https://www.analyticsvidhya.com/blog/2021/06/beginners-guide-to-universal-approximation-theorem/>

Backpropagation & autograd



$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{x}^{(L)}} \frac{\partial \mathbf{x}^{(L)}}{\partial \mathbf{s}^{(L)}} \frac{\partial \mathbf{s}^{(L)}}{\partial \mathbf{W}^{(L)}}$$

depends on the
form of the loss

derivative of the
non-linearity

$$\frac{\partial}{\partial \mathbf{W}^{(L)}} (\mathbf{W}^{(L)\top} \mathbf{x}^{(L-1)}) = \mathbf{x}^{(L-1)\top}$$

note $\nabla_{\mathbf{W}^{(L)}} \mathcal{L} \equiv \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L)}}$ is notational convention

Figure credit J. Duarte

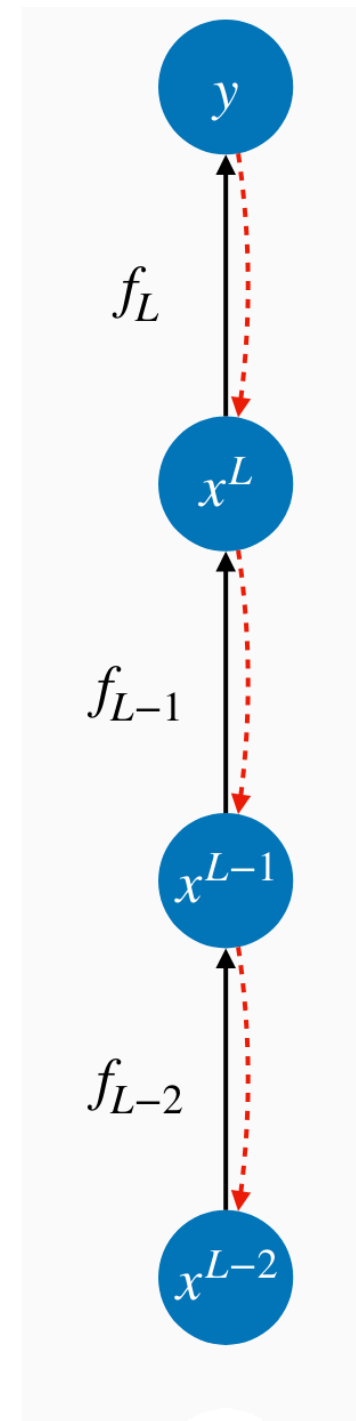
Backpropagation & autograd

$$y = f_L(\dots f_2(f_1(x))\dots)$$



$$\begin{aligned} y &= f_L(x_L) \\ x_L &= f_{L-1}(x_{L-1}) \\ &\vdots \\ x_1 &= f_1(x) \end{aligned}$$

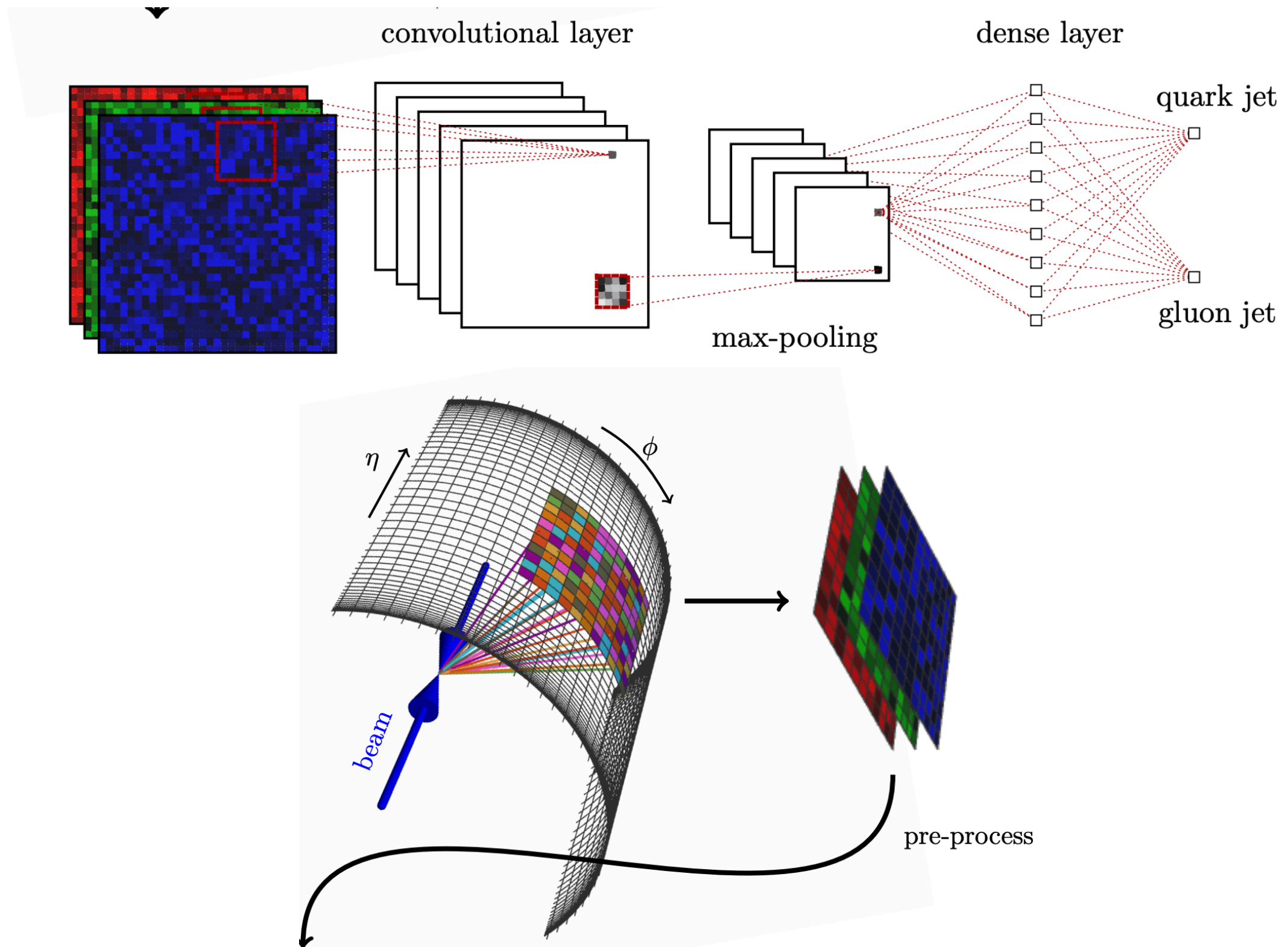
We want to compute $\frac{\partial y}{\partial x_l}$ for all $l \in \{1, \dots, L\}$



$$\begin{aligned} \frac{\partial y}{\partial x_L} &= \frac{\partial y}{\partial x_L} \frac{\partial x_L}{\partial x_{L-1}} \\ \frac{\partial y}{\partial x_{L-1}} &= \frac{\partial y}{\partial x_{L-1}} \frac{\partial x_{L-1}}{\partial x^{L-2}} \\ \frac{\partial y}{\partial x_{L-2}} &= \frac{\partial y}{\partial x_{L-2}} \frac{\partial x_{L-2}}{\partial x^{L-3}} \end{aligned}$$

At each step we can reuse the computation of the previous step!

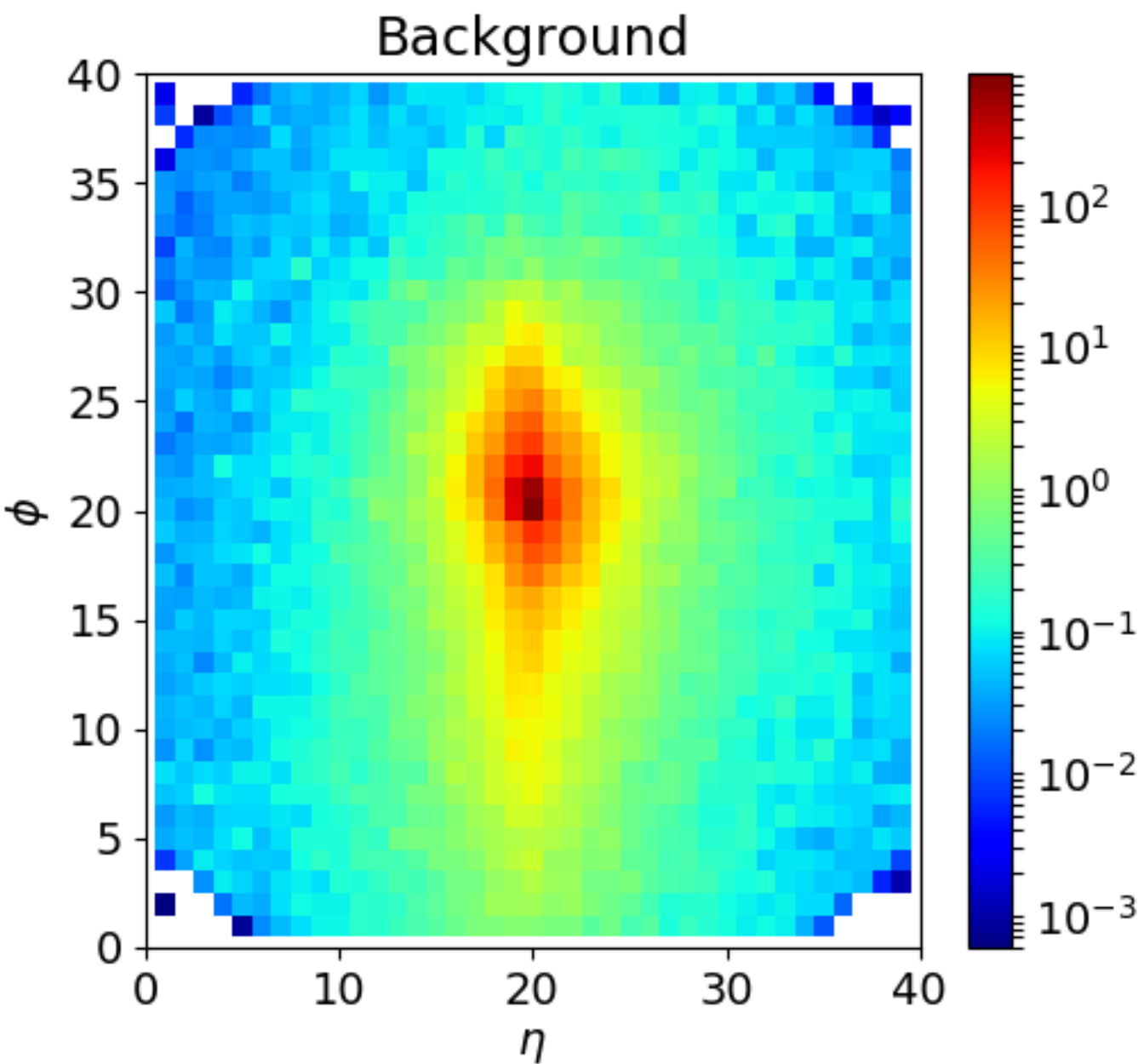
Architectures : CNN



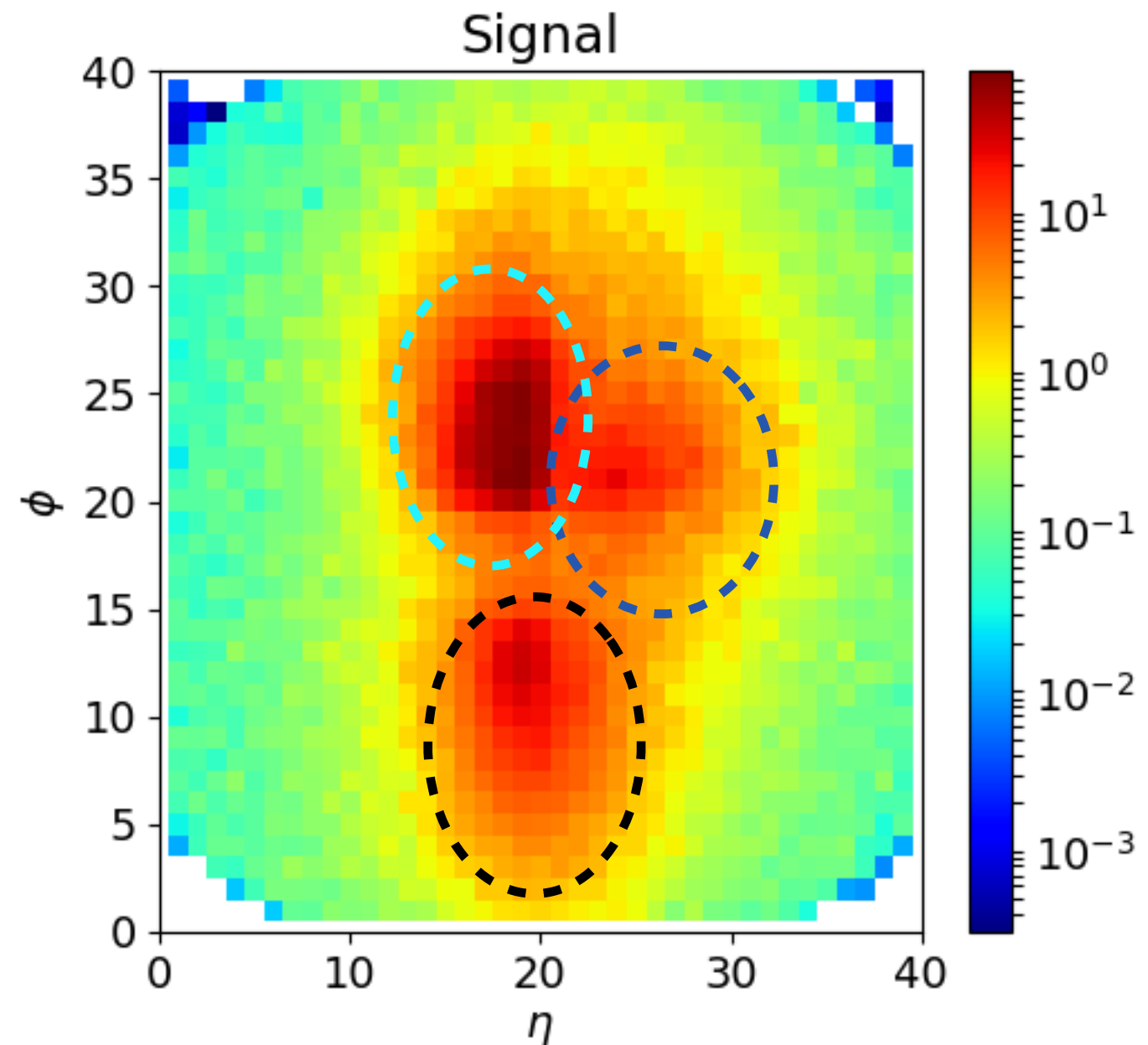
An example jet representation

Tagging objects

arXiv : 1902.09914



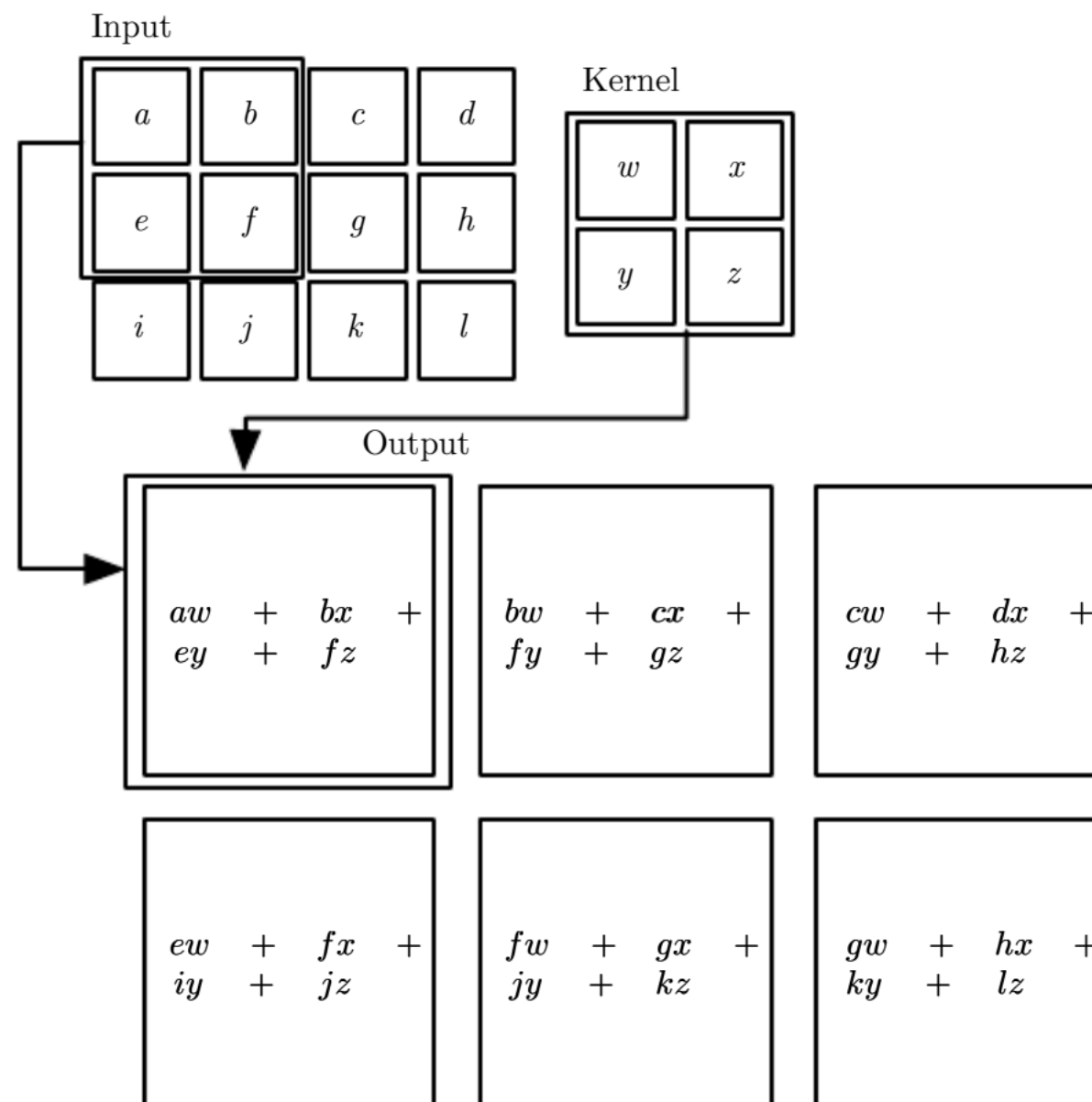
1-prong gluon background



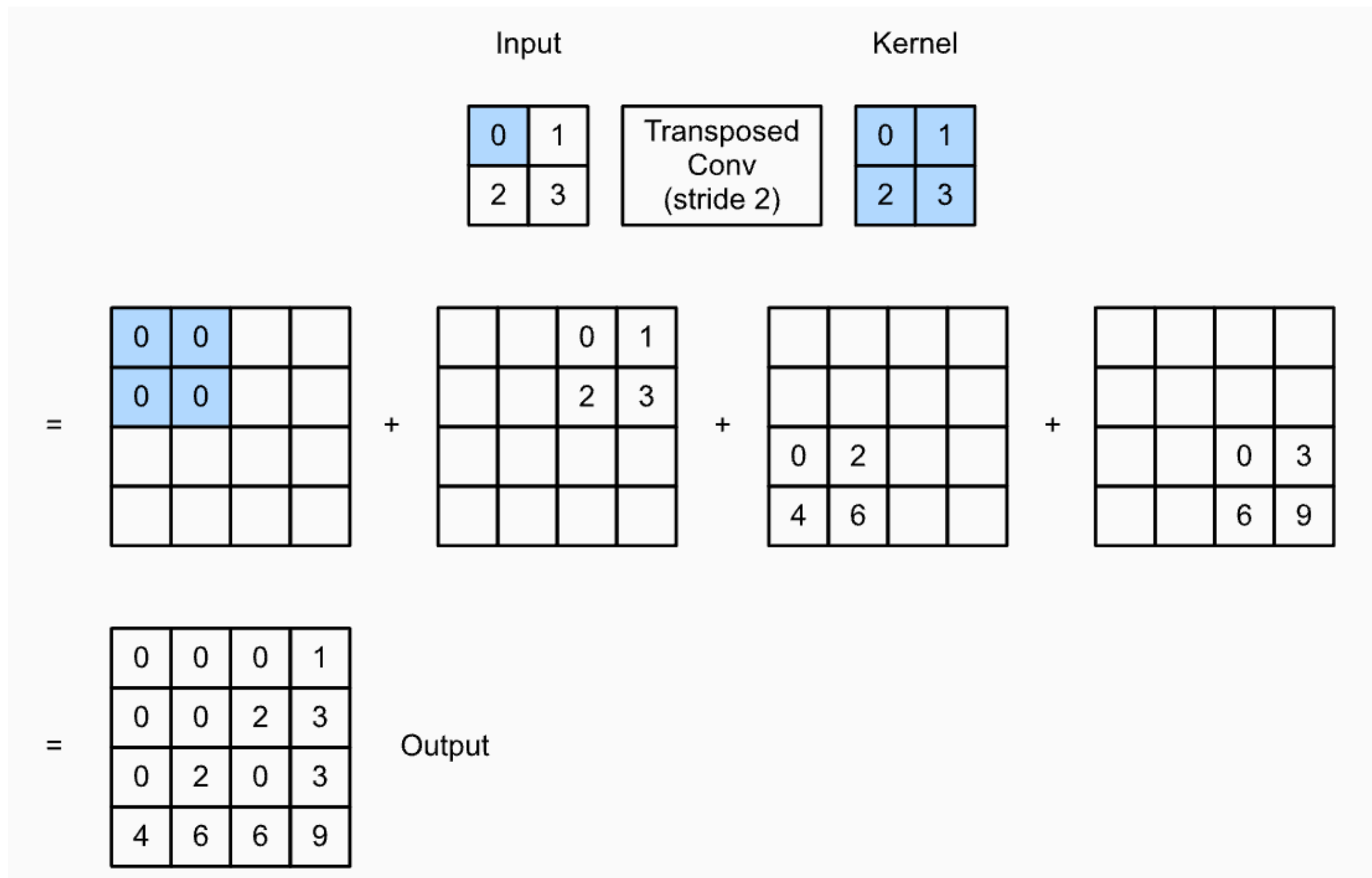
3-prong top signal

Architectures : CNN

$$S(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n)$$



Architectures : Transposed Convs



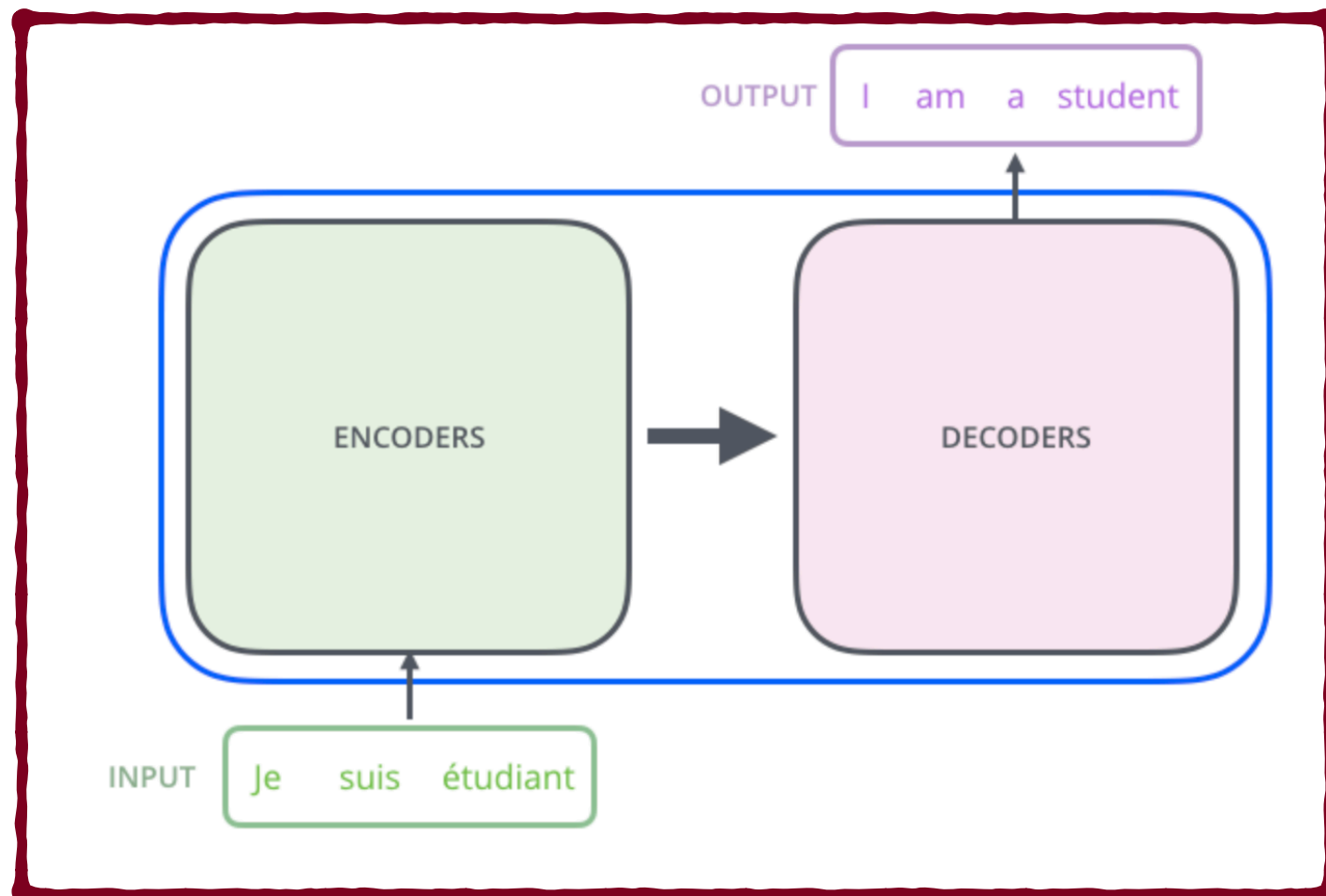
https://d2l.ai/chapter_computer-vision/transposed-conv.html

Sequential data learning



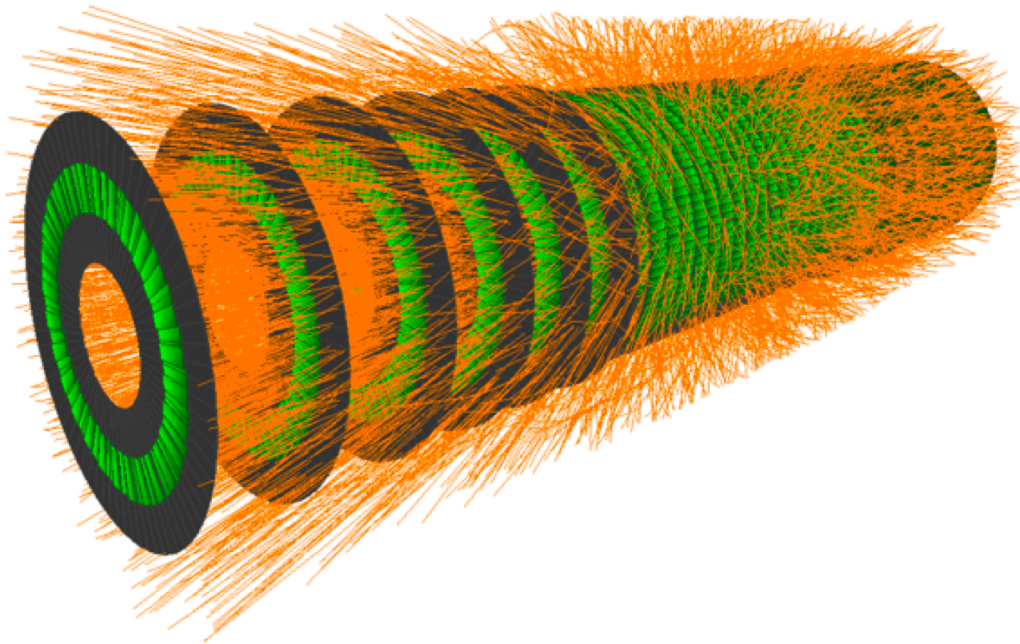
What are the general principles to code the transformer ?

<https://jalammar.github.io/illustrated-transformer/>

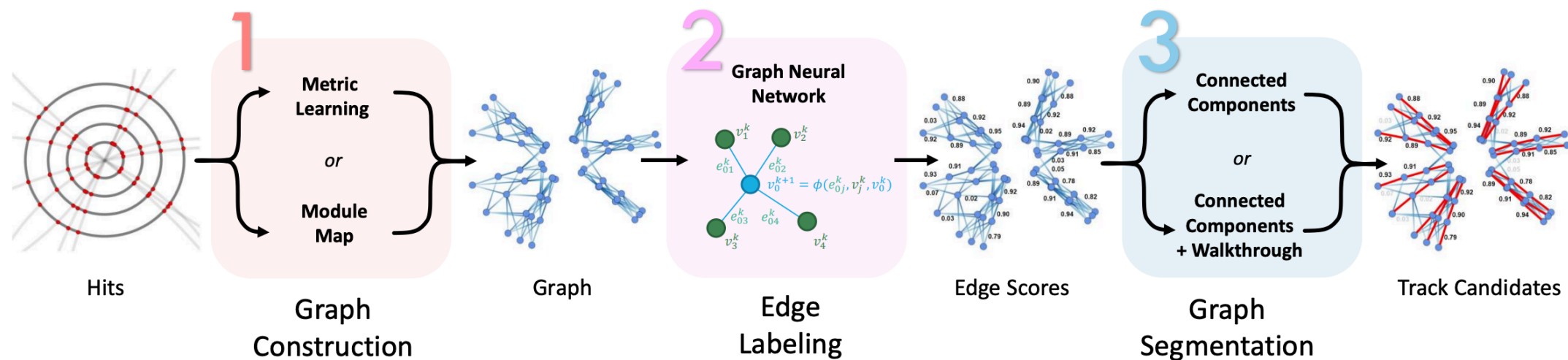
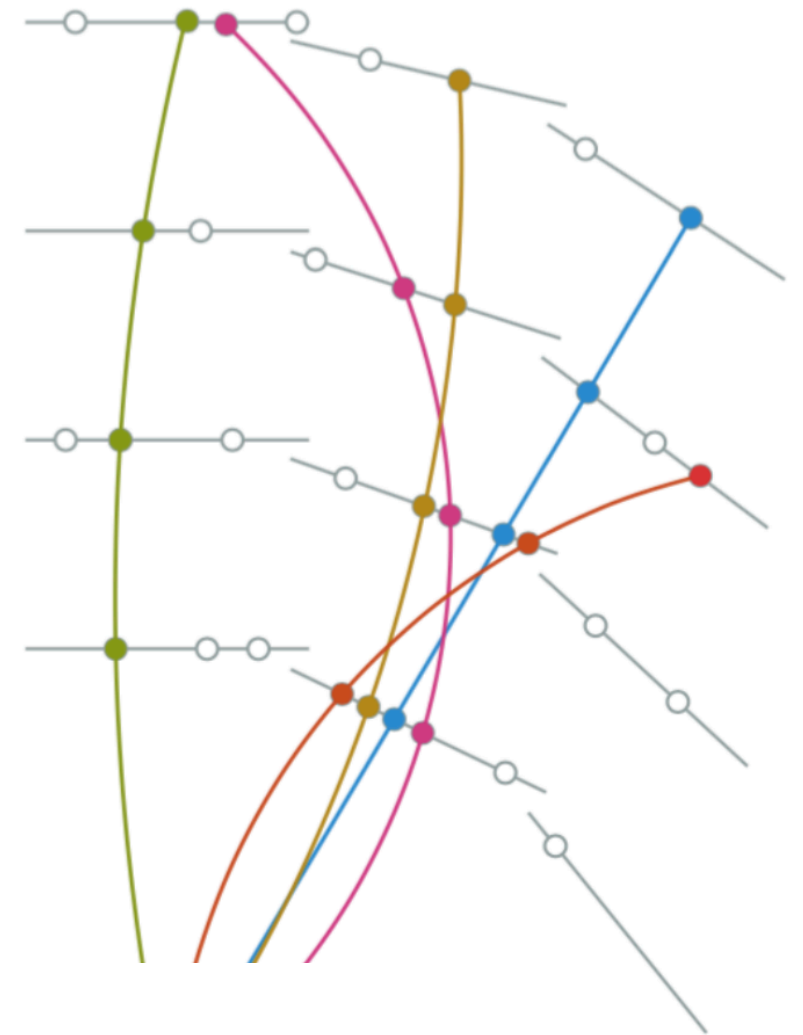


An encoder - decoder turned out to be serving this purpose best.

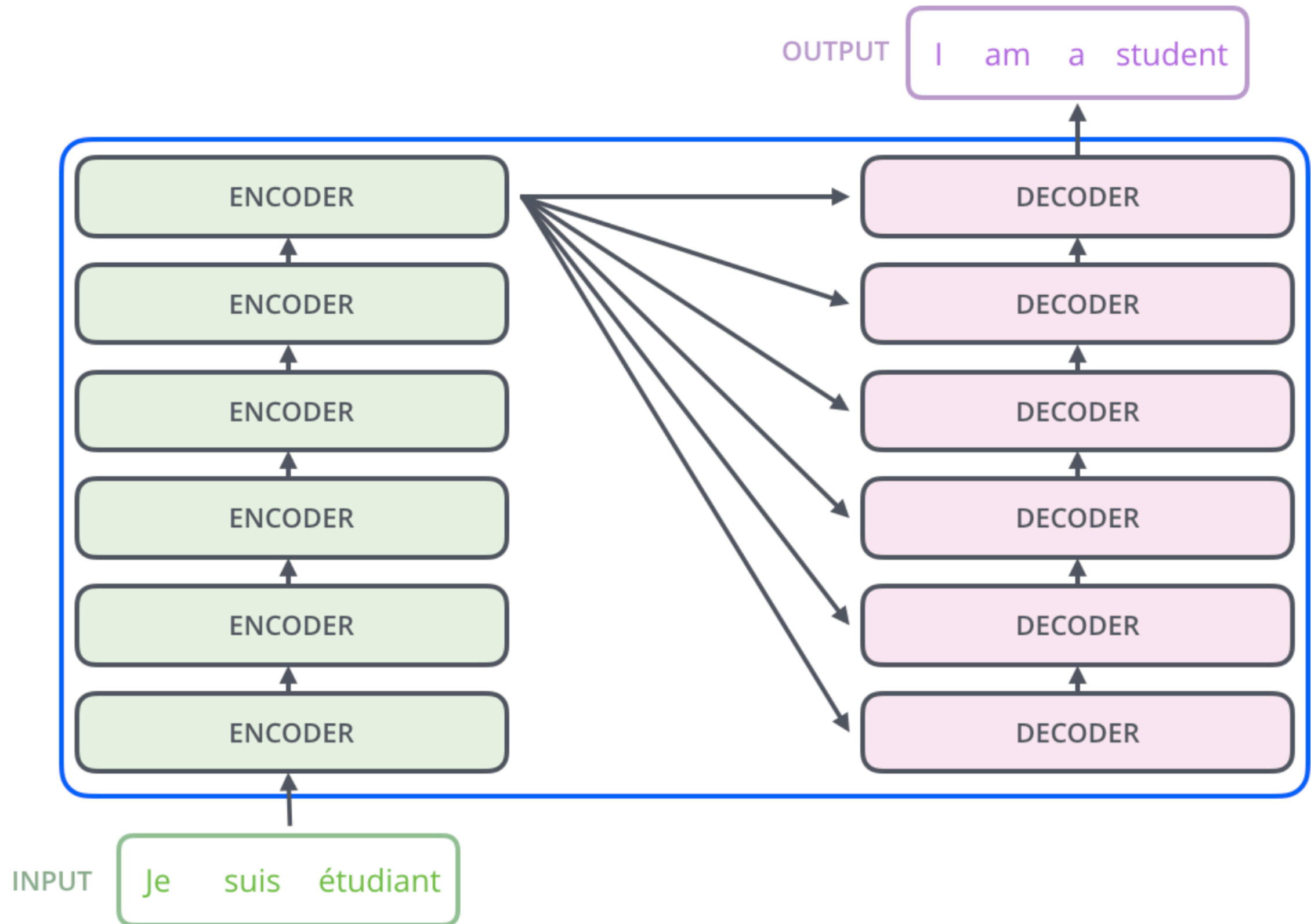
Tracking & ML



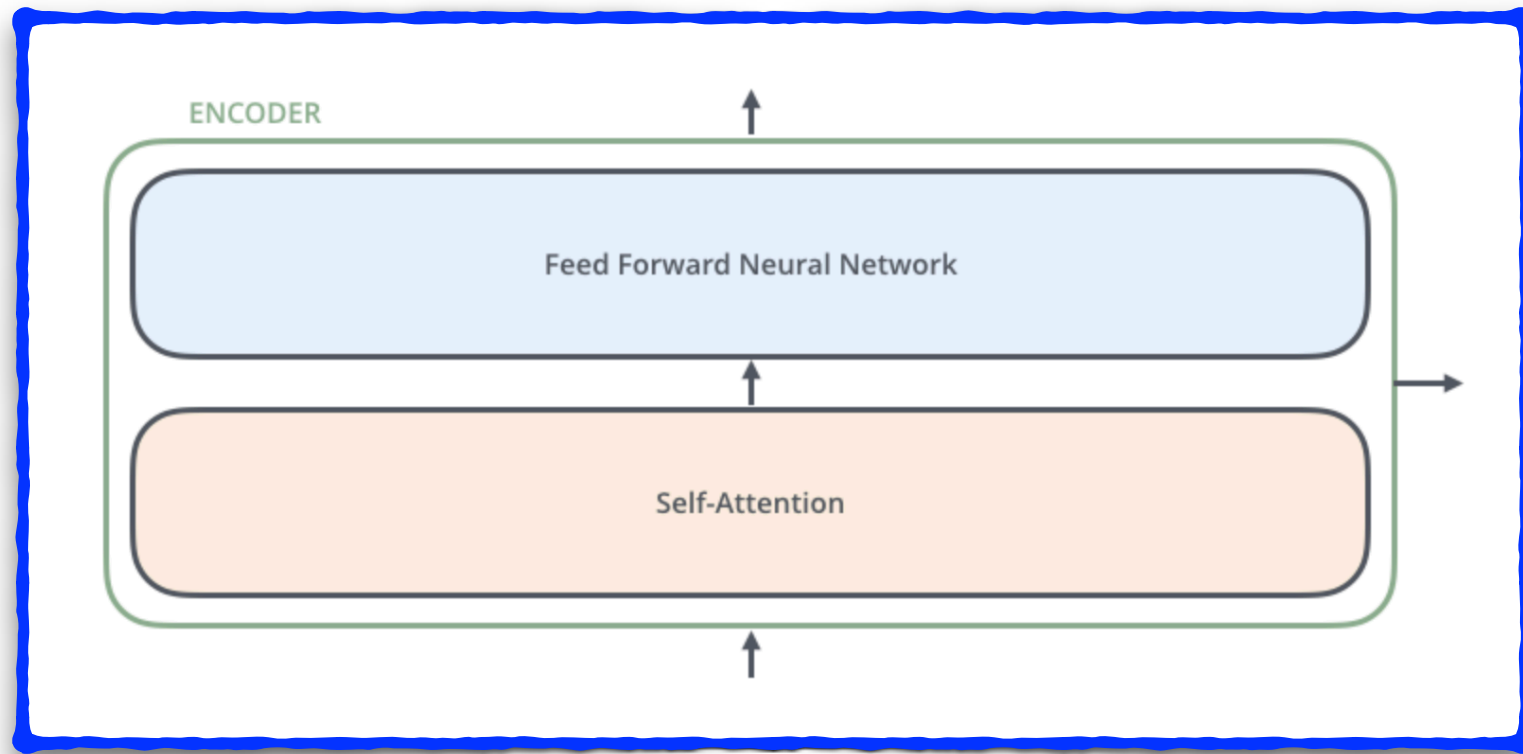
An exponentially large edge finding problem



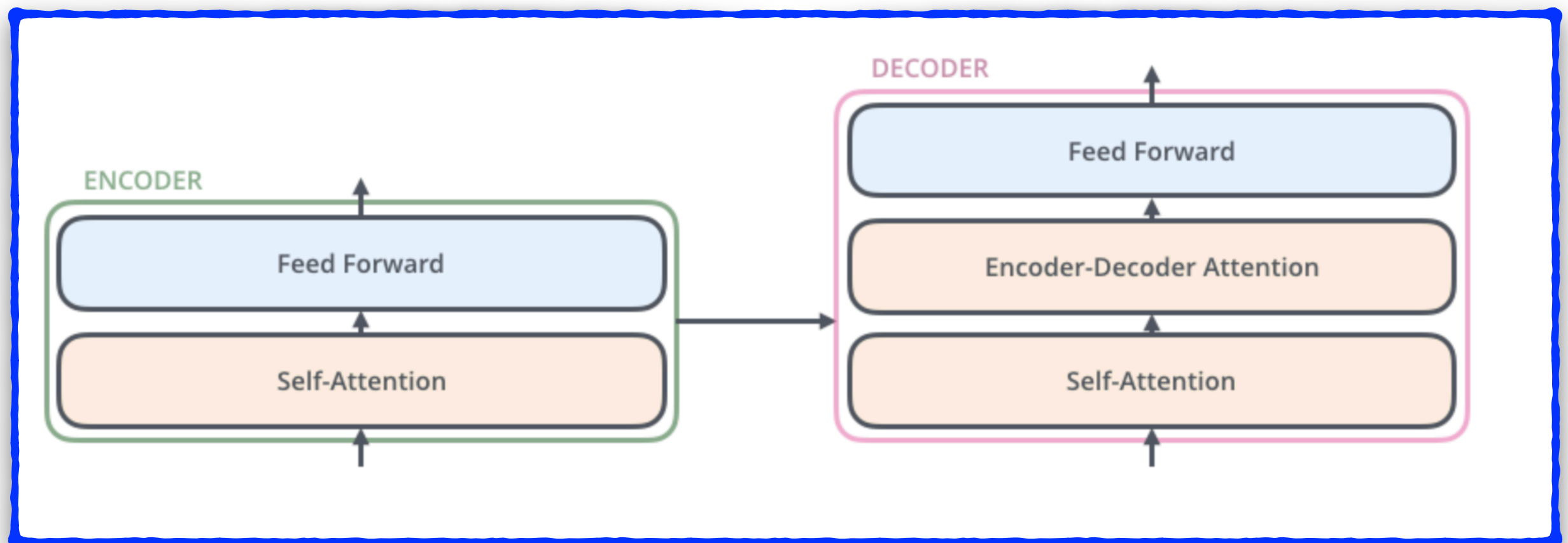
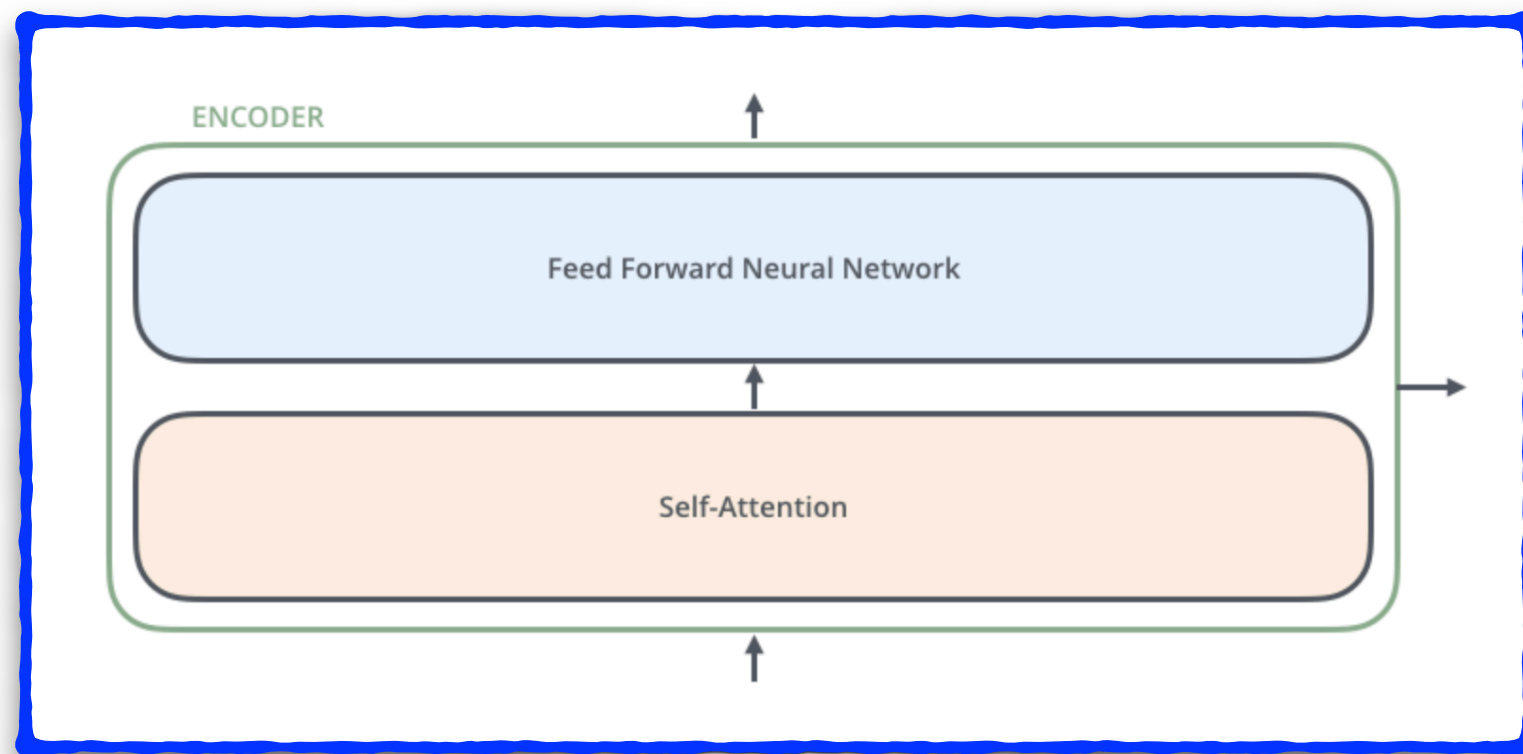
Sequence to Sequence mapping



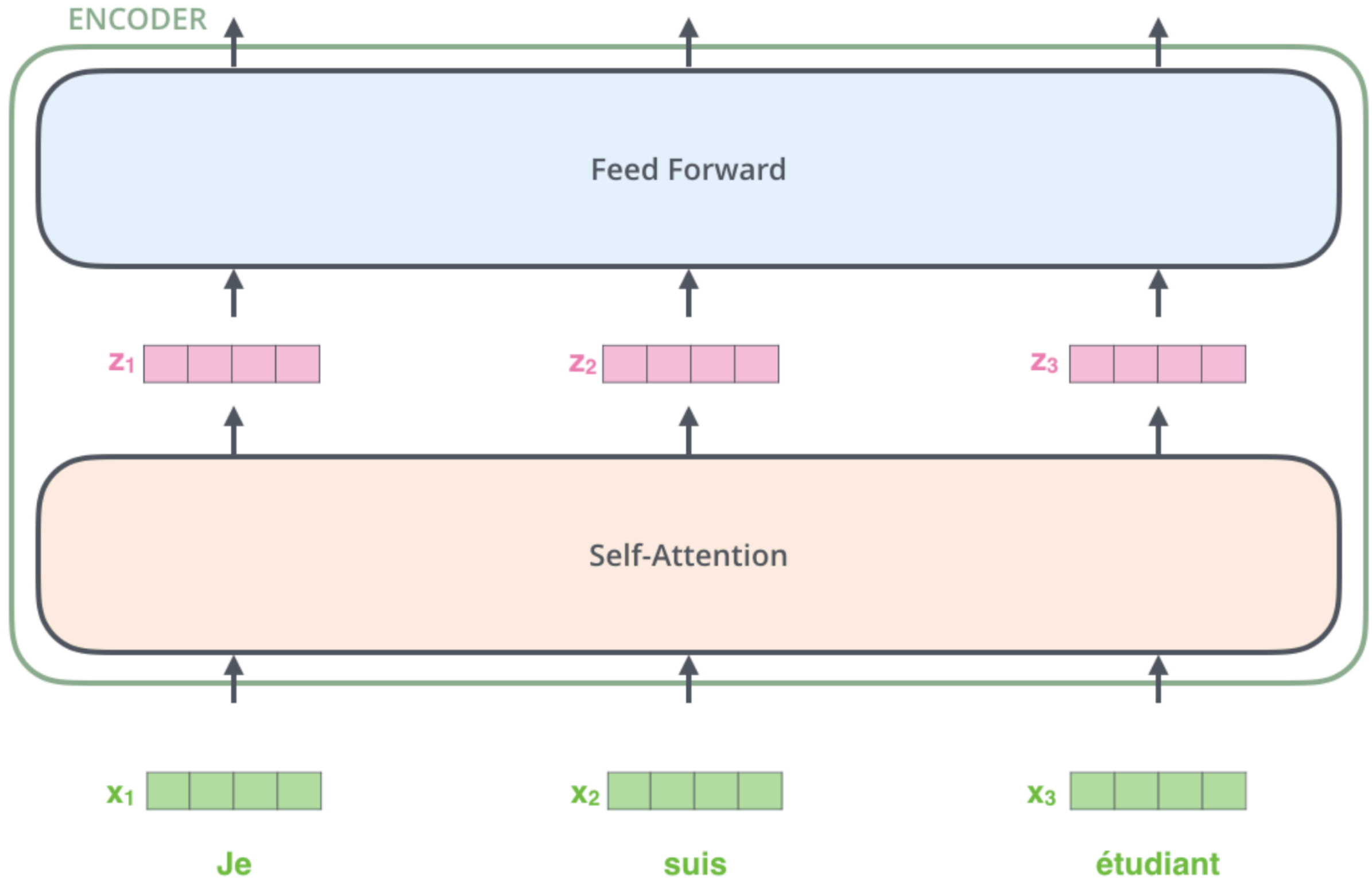
Encoder feeds the decoder



Encoder feeds the decoder

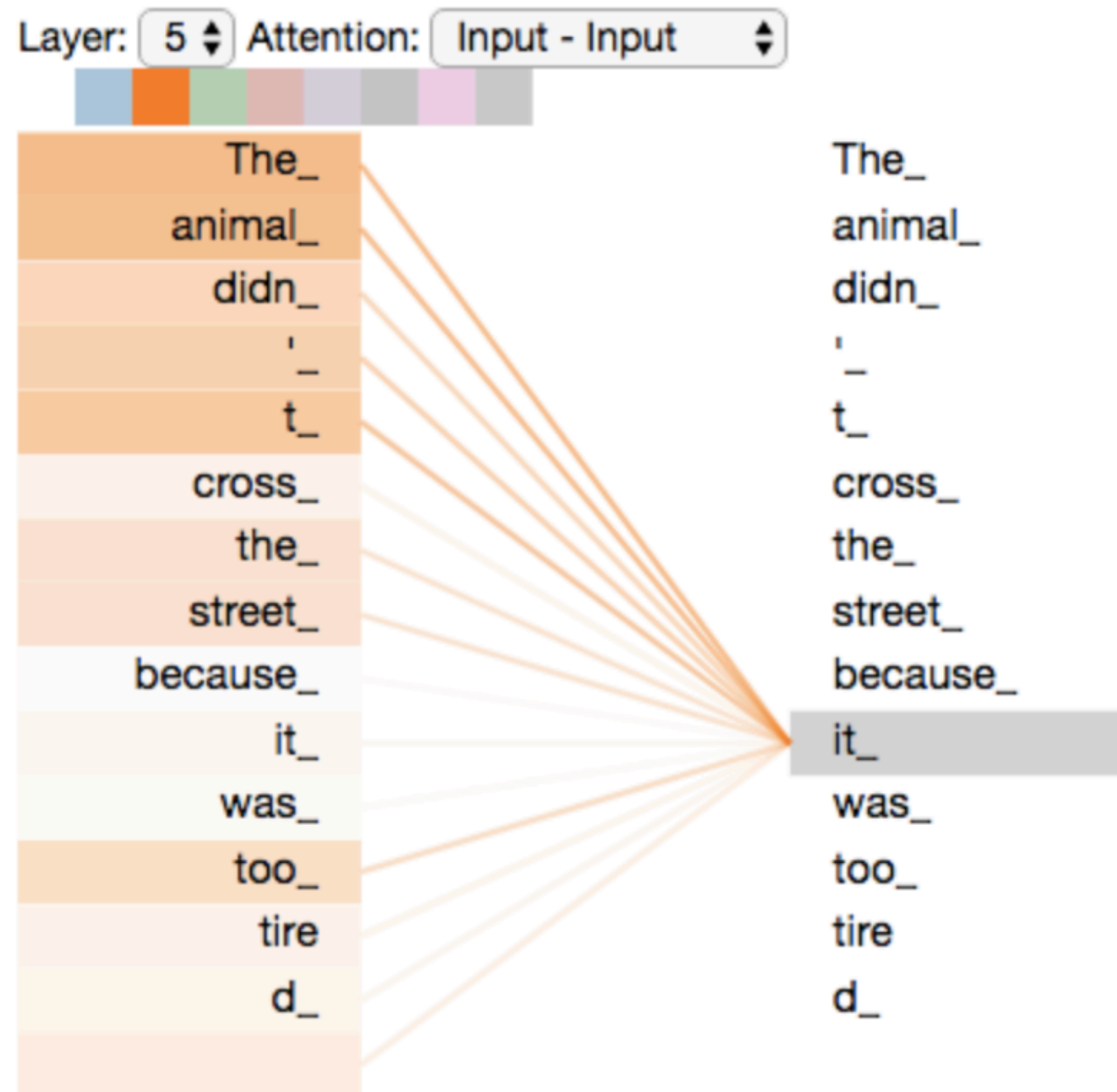


Encoder feeds the decoder

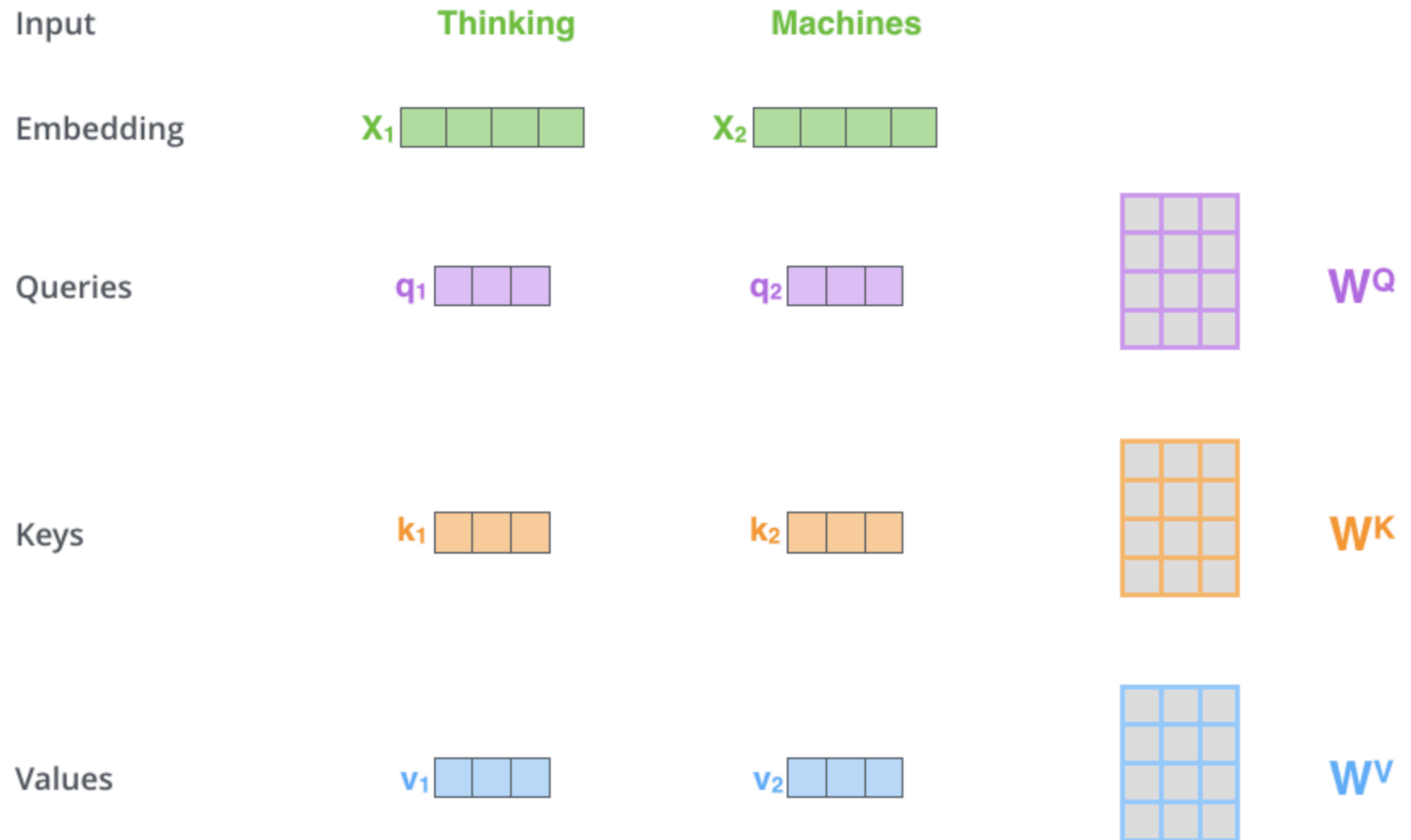


The concept of self-attention

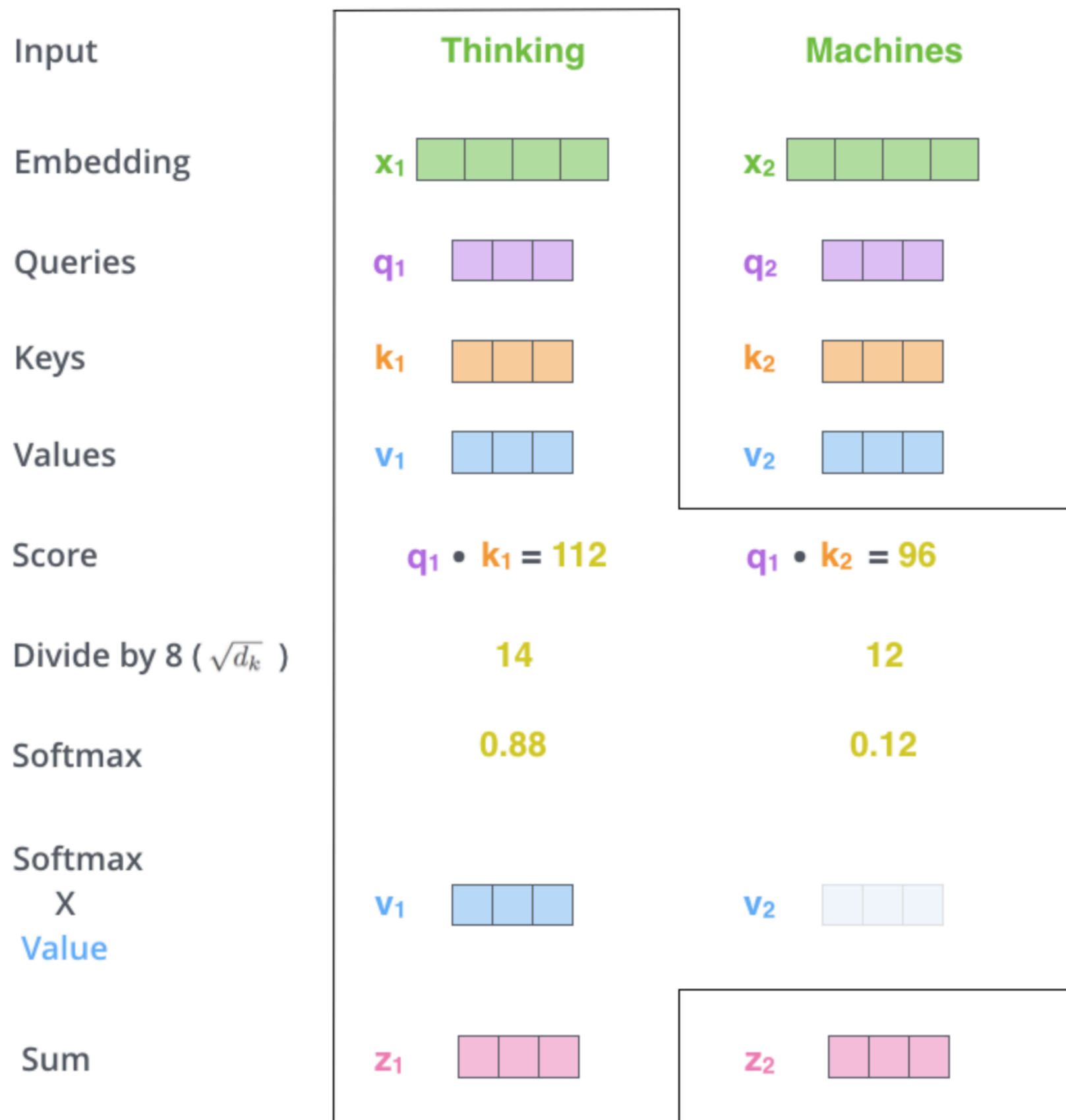
"The animal didn't cross the street because it was too tired"



How self attention works?



How self attention works?

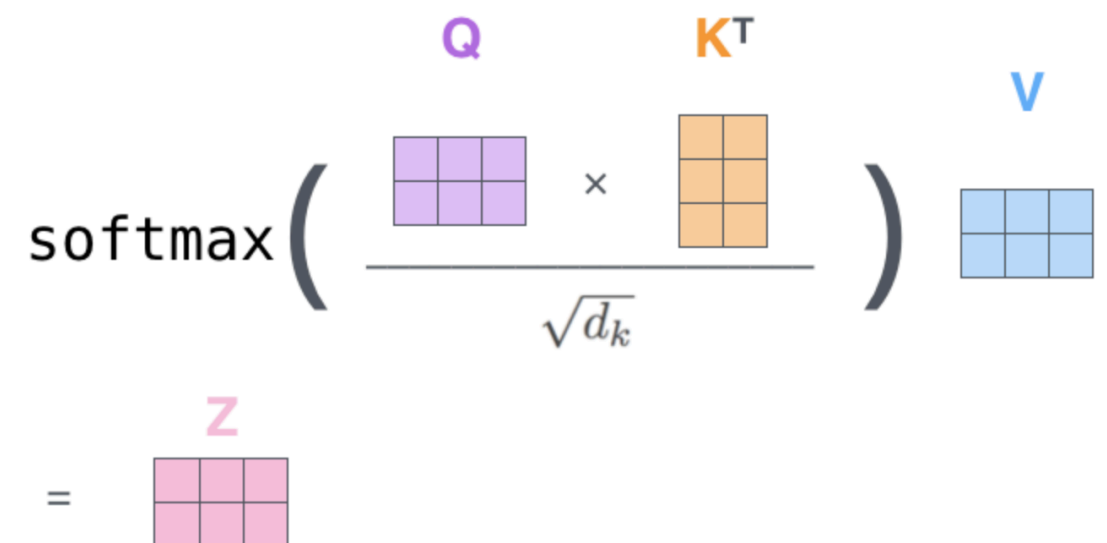


Computing the attention values

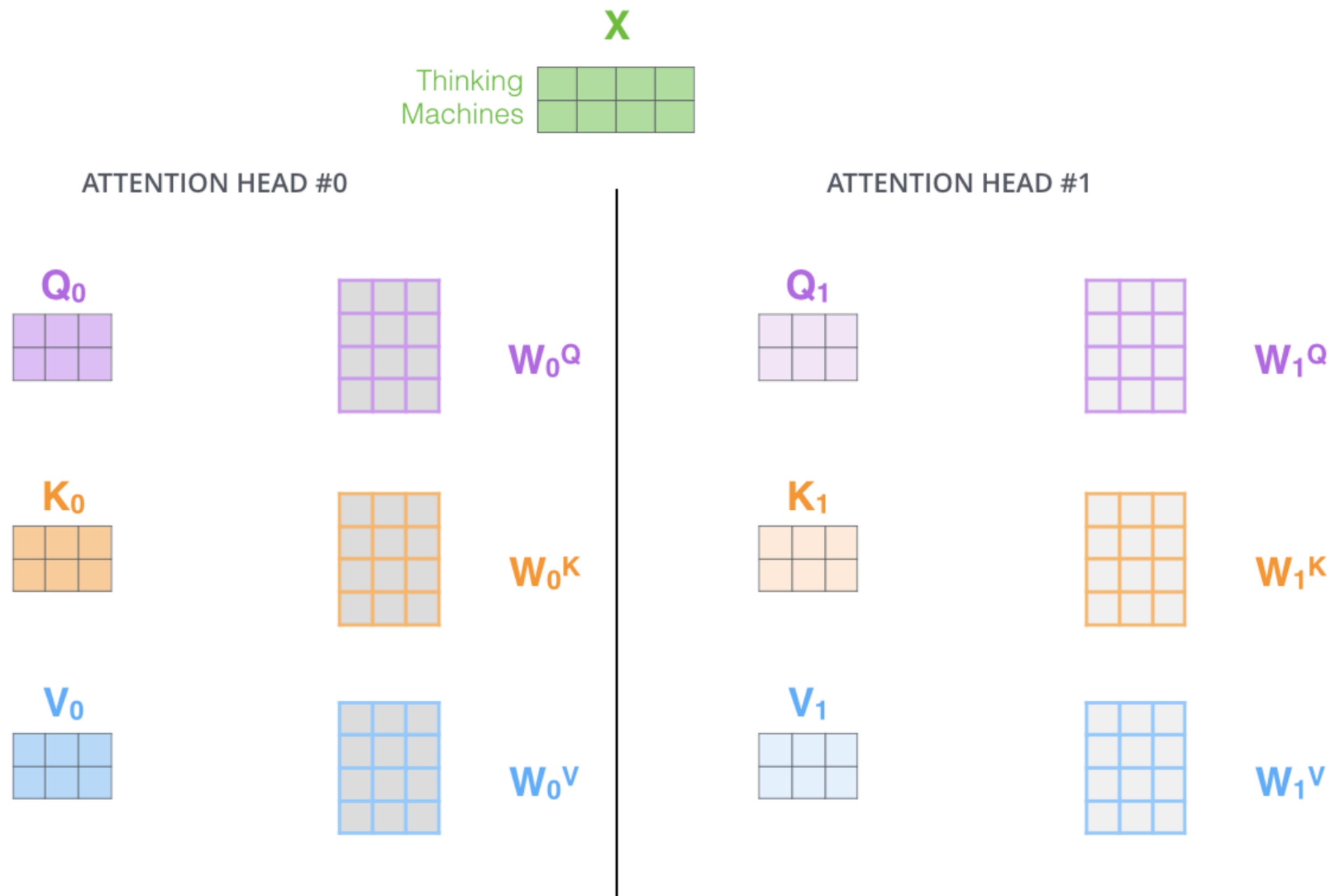
$$\mathbf{X} \times \mathbf{W}^Q = \mathbf{Q}$$


$$\mathbf{X} \times \mathbf{W}^K = \mathbf{K}$$


$$\mathbf{X} \times \mathbf{W}^V = \mathbf{V}$$

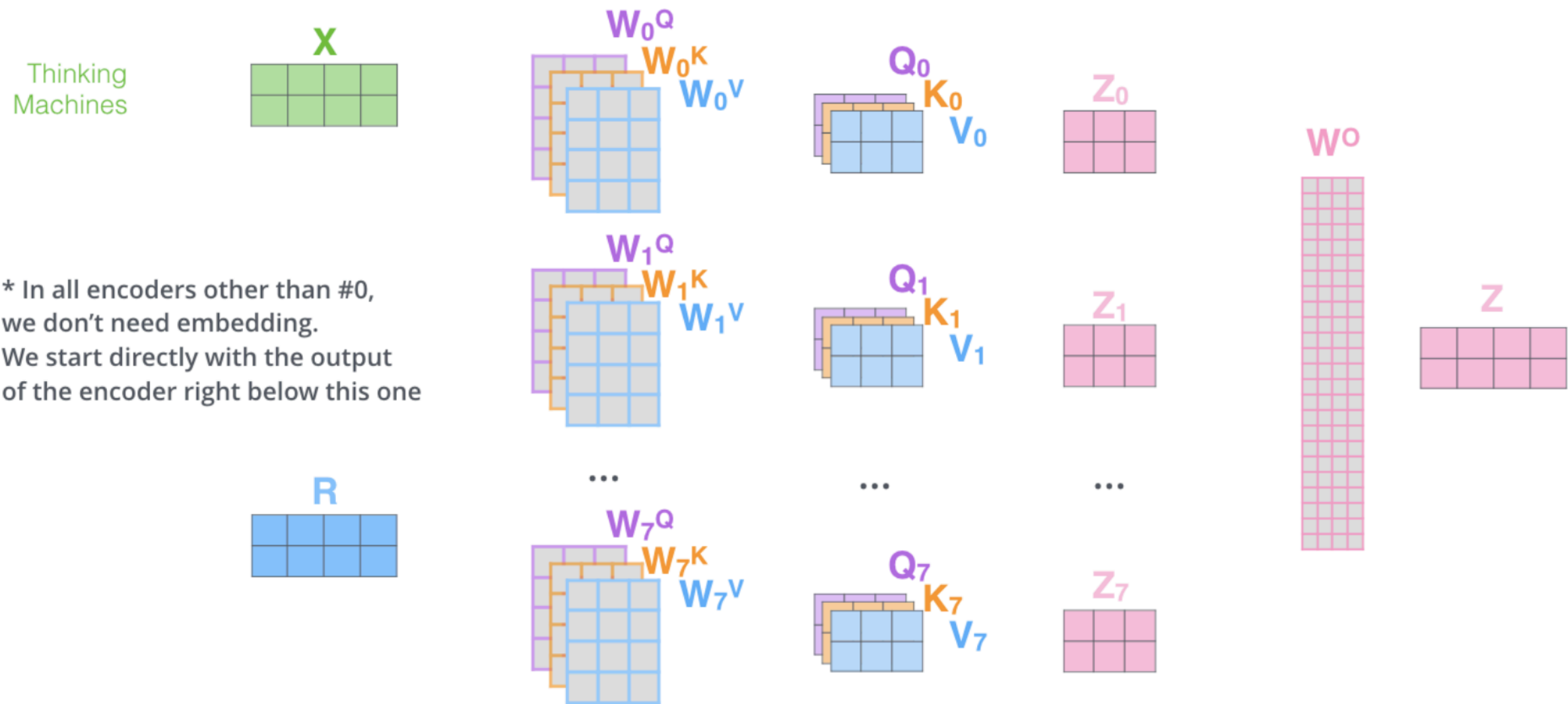

$$\text{softmax}\left(\frac{\mathbf{Q} \times \mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}$$

$$= \mathbf{Z}$$

Computing the multi-head attention

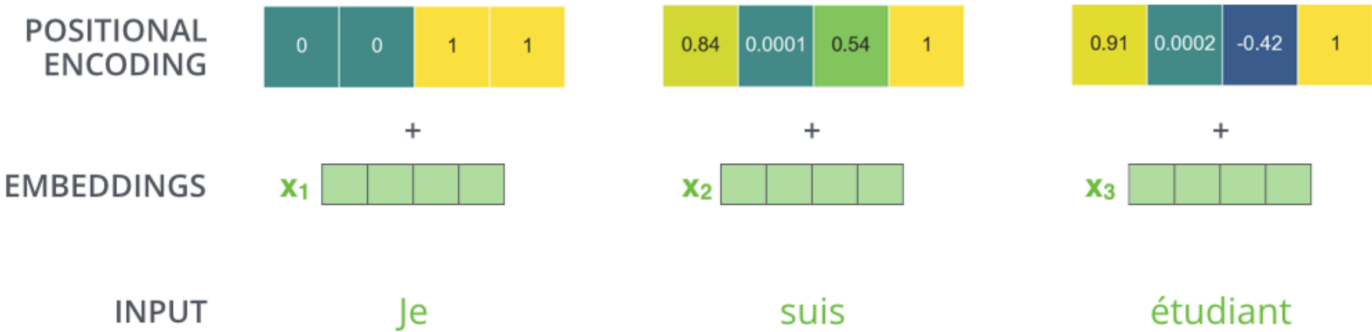


Computing the multi-head attention

- 1) This is our input sentence*
- 2) We embed each word*
- 3) Split into 8 heads. We multiply X or R with weight matrices
- 4) Calculate attention using the resulting $Q/K/V$ matrices
- 5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

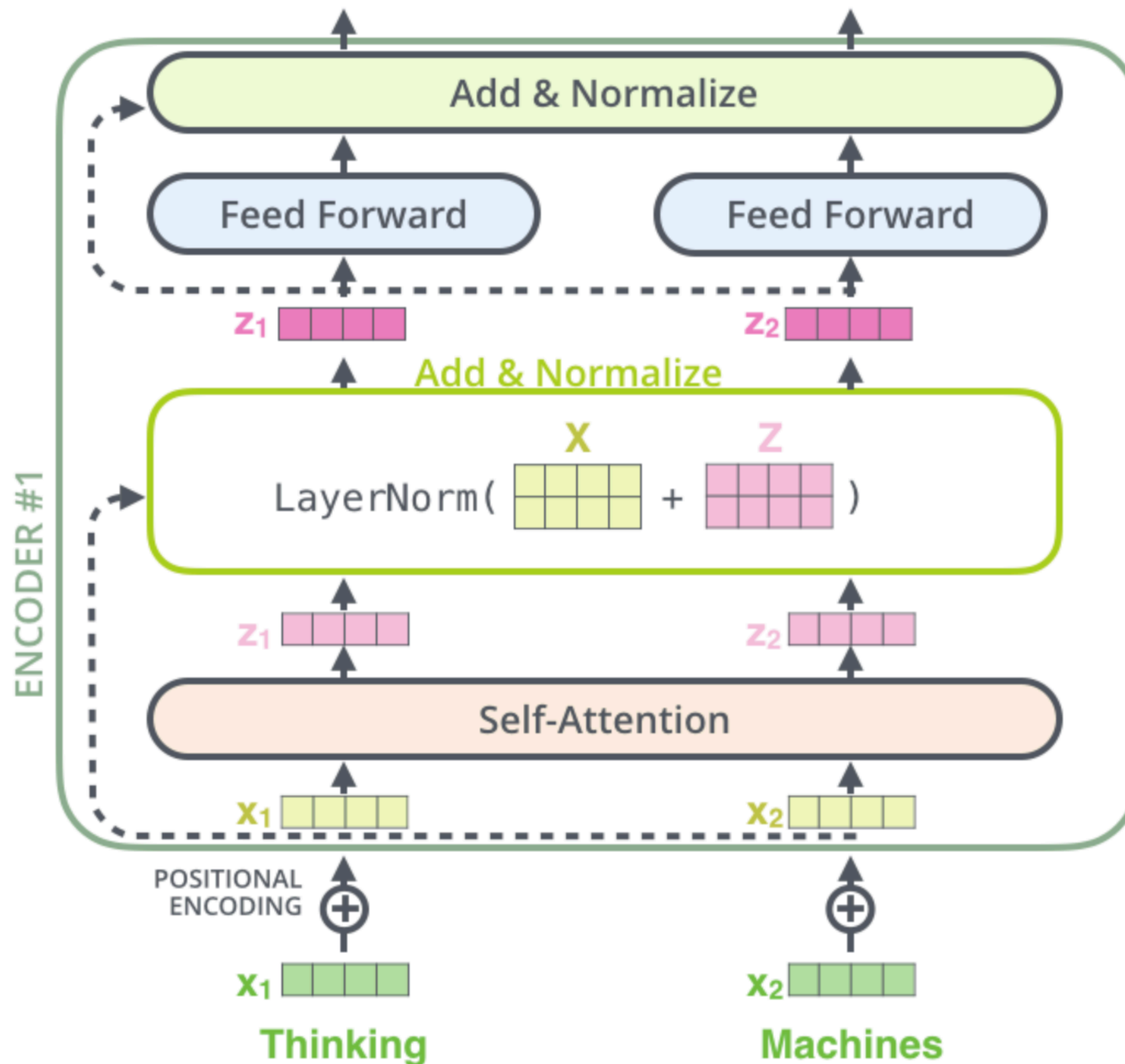


* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

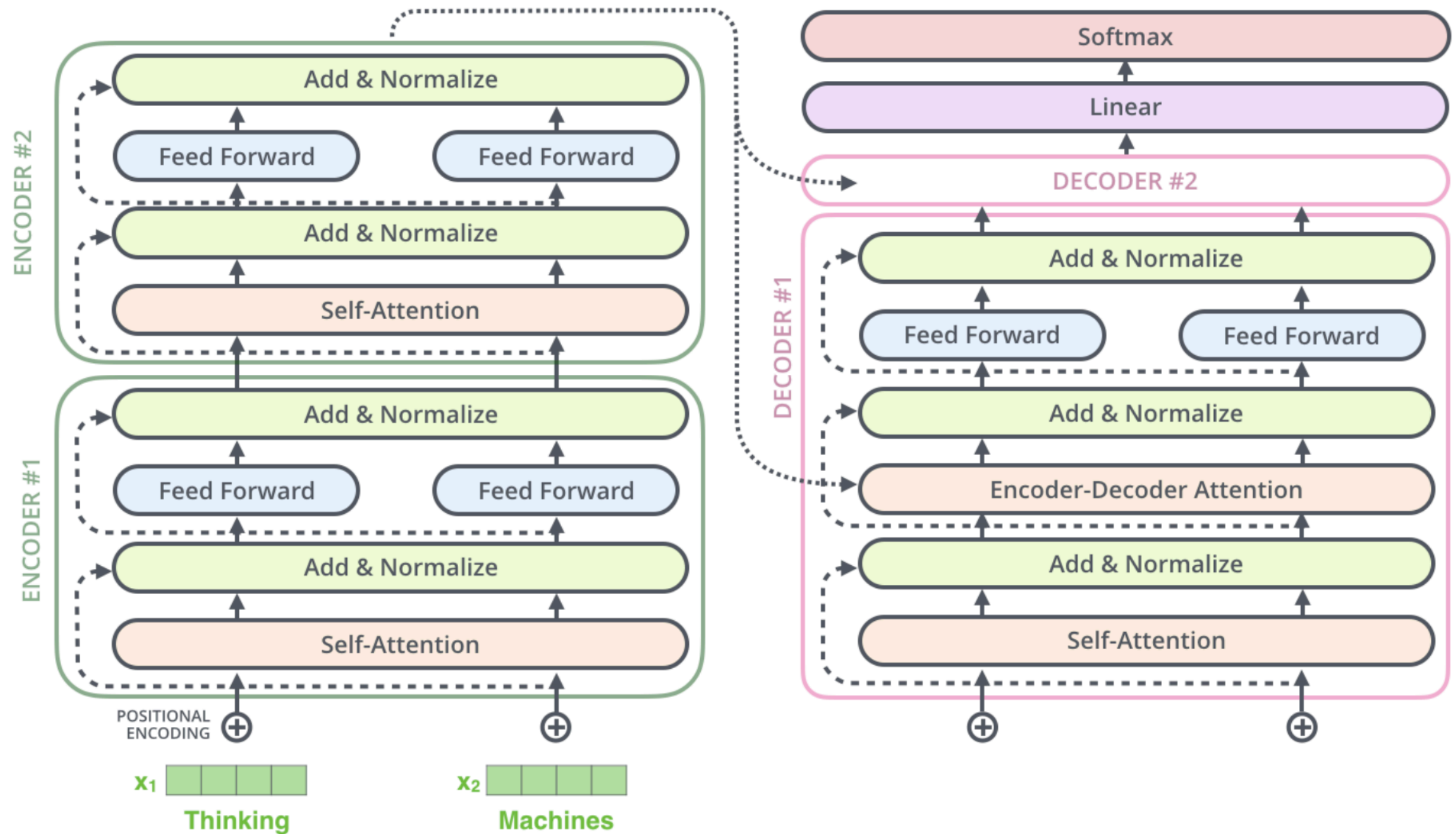


Add the time stamp to ensure positional encoding

Let's see how the encoder looks



The simplest encoder-decoder net



Transformer architecture

