PWA13/ATHOS8
Williamsburg, 2024-05-31

Towards machine learning the hadron spectrum

César Fernández Ramírez
Departamento de Física Interdisciplinar
Universidad Nacional de Educación a Distancia (UNED)

UNED Facultad de Ciencias

JPAC

EXOHAD
Exotic Hadrons Topical Collaboration

Towards machine learning the hadron spectrum?

César Fernández Ramírez
Departamento de Física Interdisciplinar
Universidad Nacional de Educación a Distancia (UNED)

UNED Facultad de Ciencias

JPAC

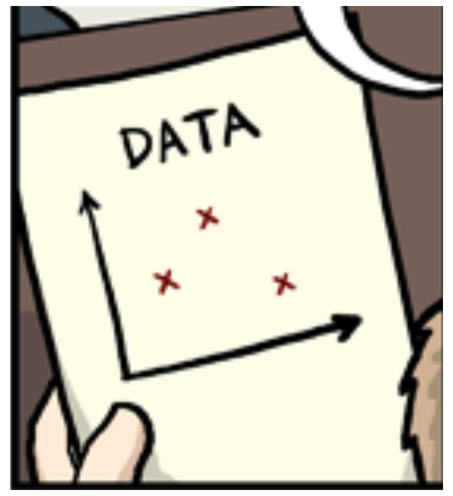
ExoHad
Exotic Hadrons Topical Collaboration

Outlook

- Motivation ➛ Explore new ways to learn the properties of the hadron spectrum
- Standard lineshape analysis
- Neural networks
- Benchmark
- Takeaways

Standard lineshape analysis

Top-down approach



Start from a model/theory

⇩

Compute amplitude

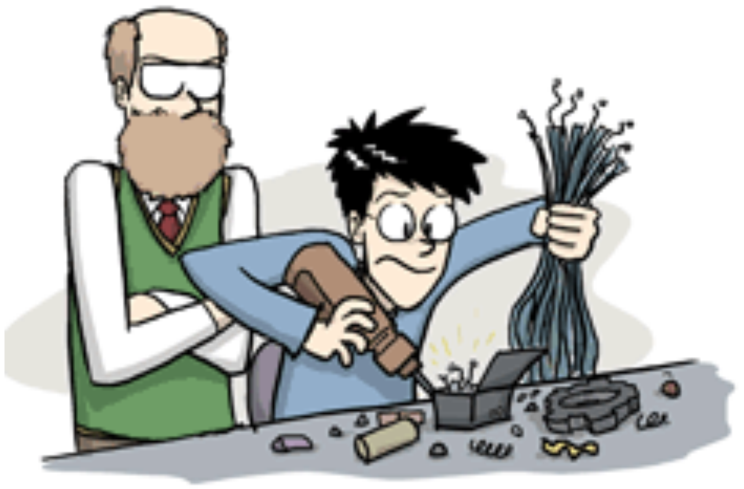
⇩

Compare to data (or not)

Predictive power 😎
Physics interpretation 😎
(within a model 😢)
Biased by hypothesis 🤪 ❓

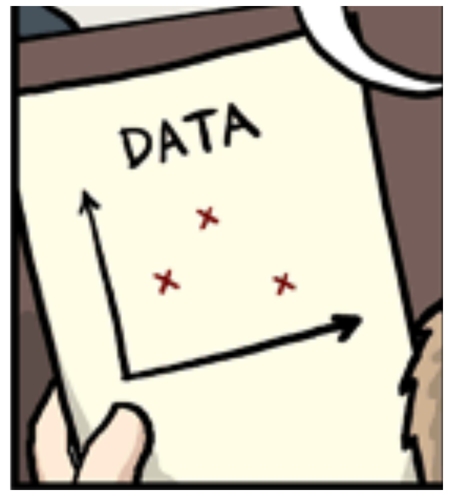
PhD comics

Bottom-up approach



Extract physics

Set of generic amplitudes



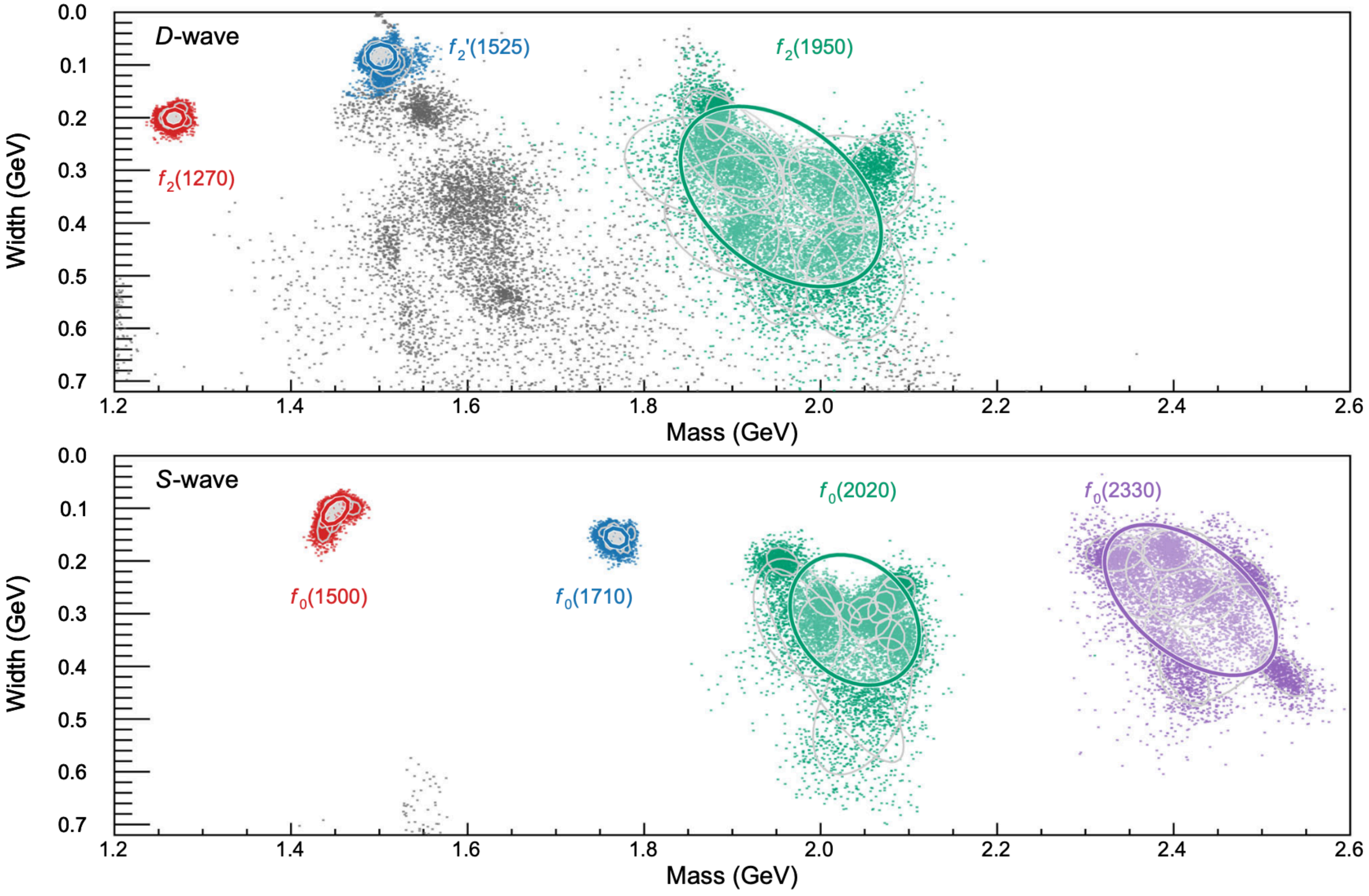
PhD comics

Start from data

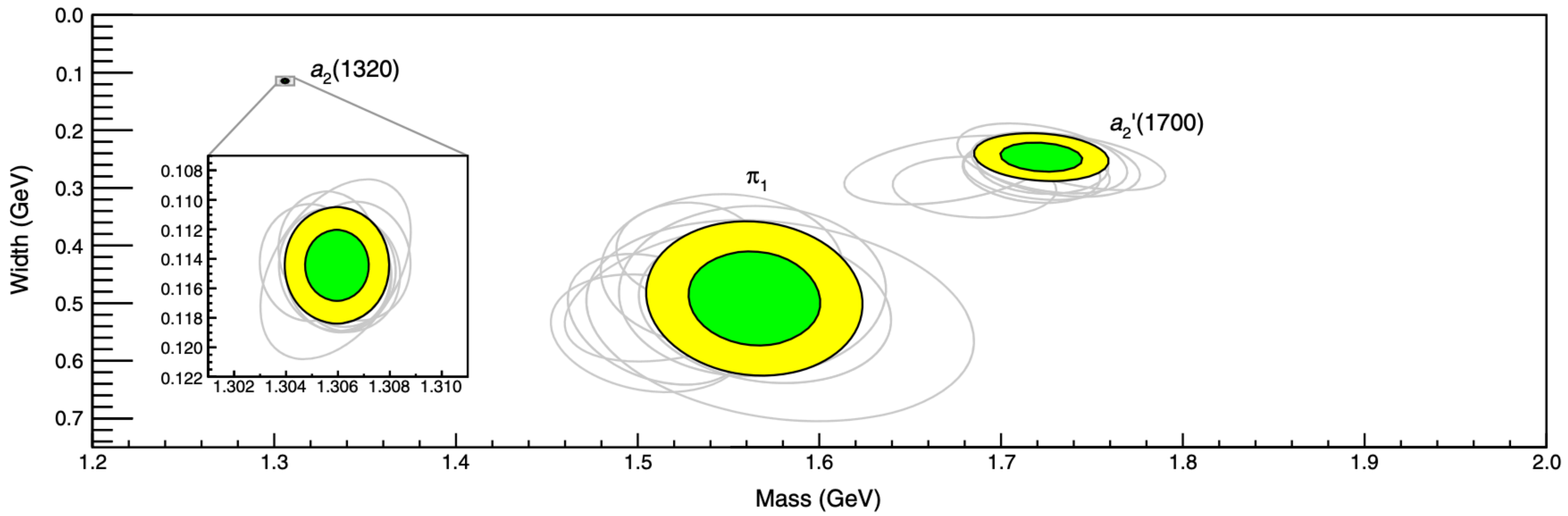
Less predictive 😔
Some interpretation 😕
Minimal bias 😎

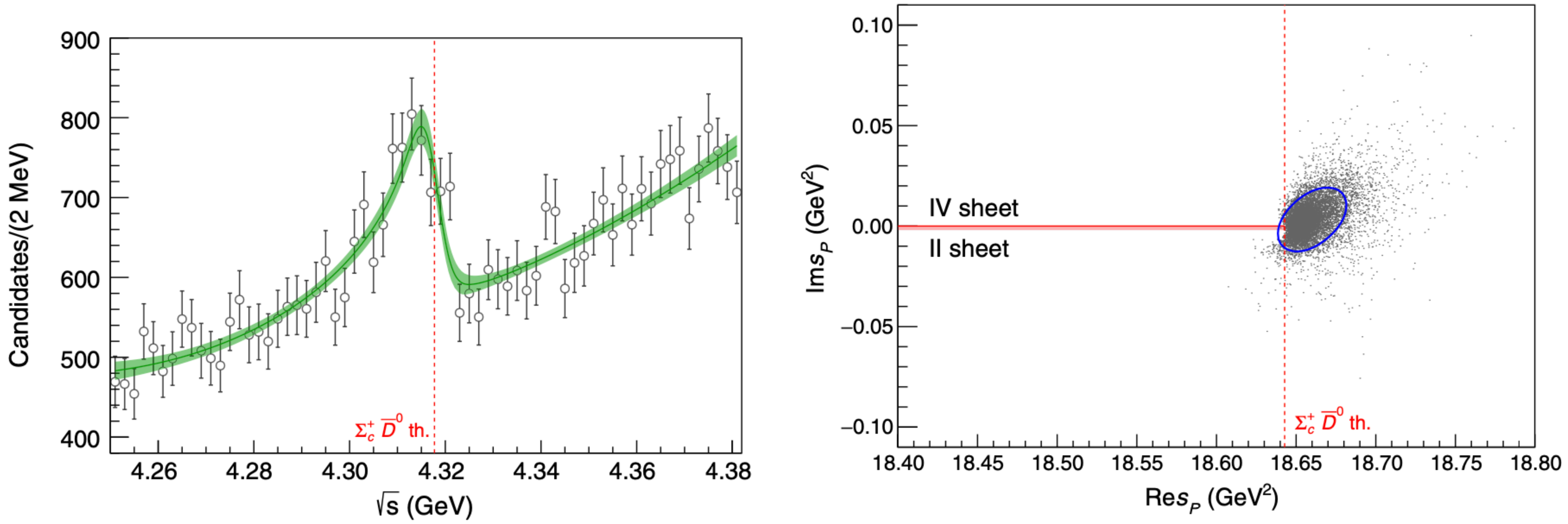
Examples

Rodas et al (JPAC) EPJC 82 (2022) 80



Rodas et al (JPAC) PRL 122 (2019) 042002





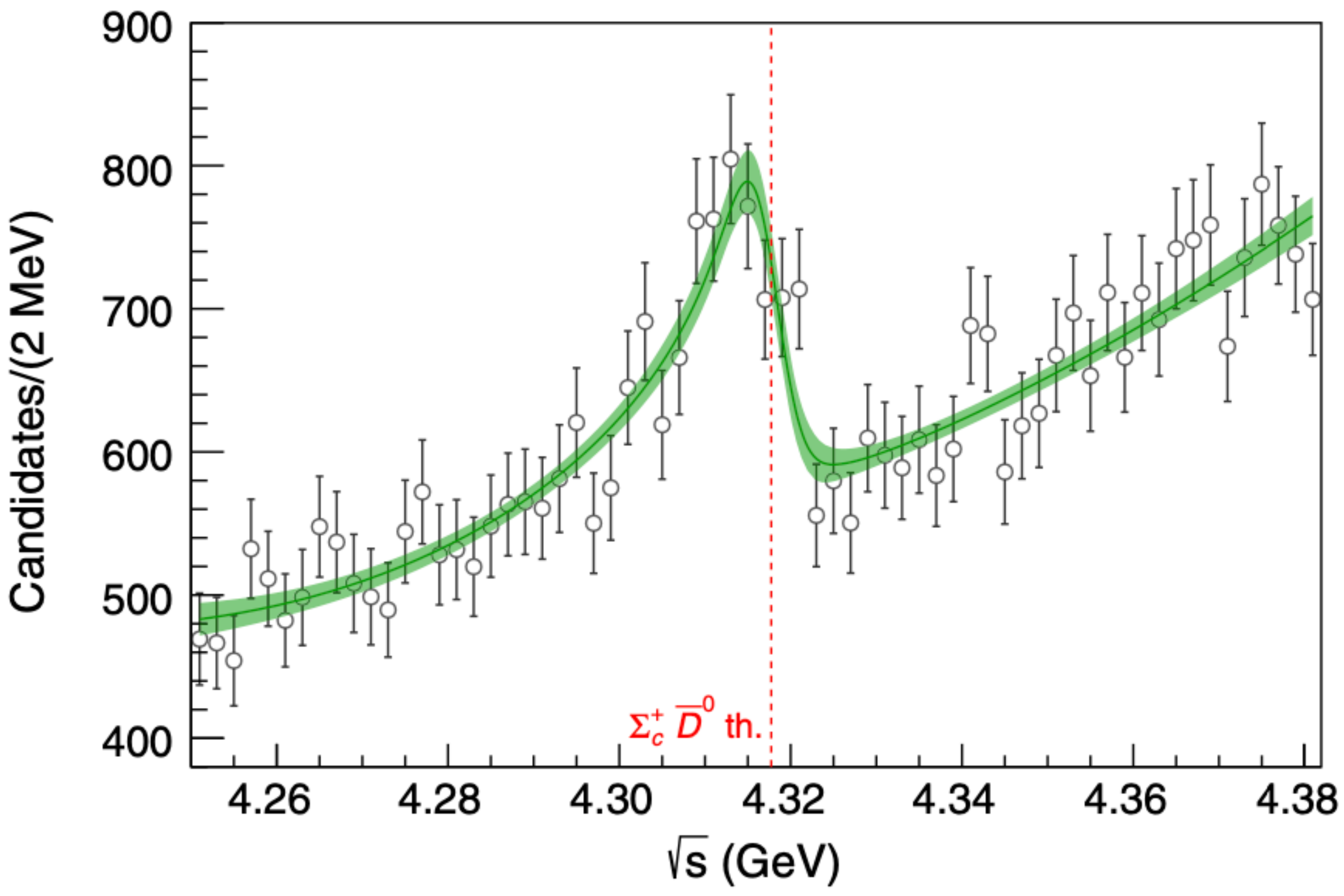
CFR et al (JPAC) PRL 123 (2019) 092001

Standard approach to resonant lineshape analysis

- Take an amplitude, it has parameters to be determined
- Fit data using Maximum Likelihood or χ^2
- Extract parameters, get pole positions and compute uncertainties
- Assess the probability that those data were generated by your amplitude
- If χ^2 is reasonable, one can claim that the physical interpretation of the data is possible
- One can do this with different amplitudes that represent different underlying dynamics
- Compare amplitudes? Compare dynamics?

J/Psi projection data

- We focus on Sigma-D threshold
- Only one partial wave contributes to the signal
- The threshold is responsible for the dynamics
- Other singularities are irrelevant

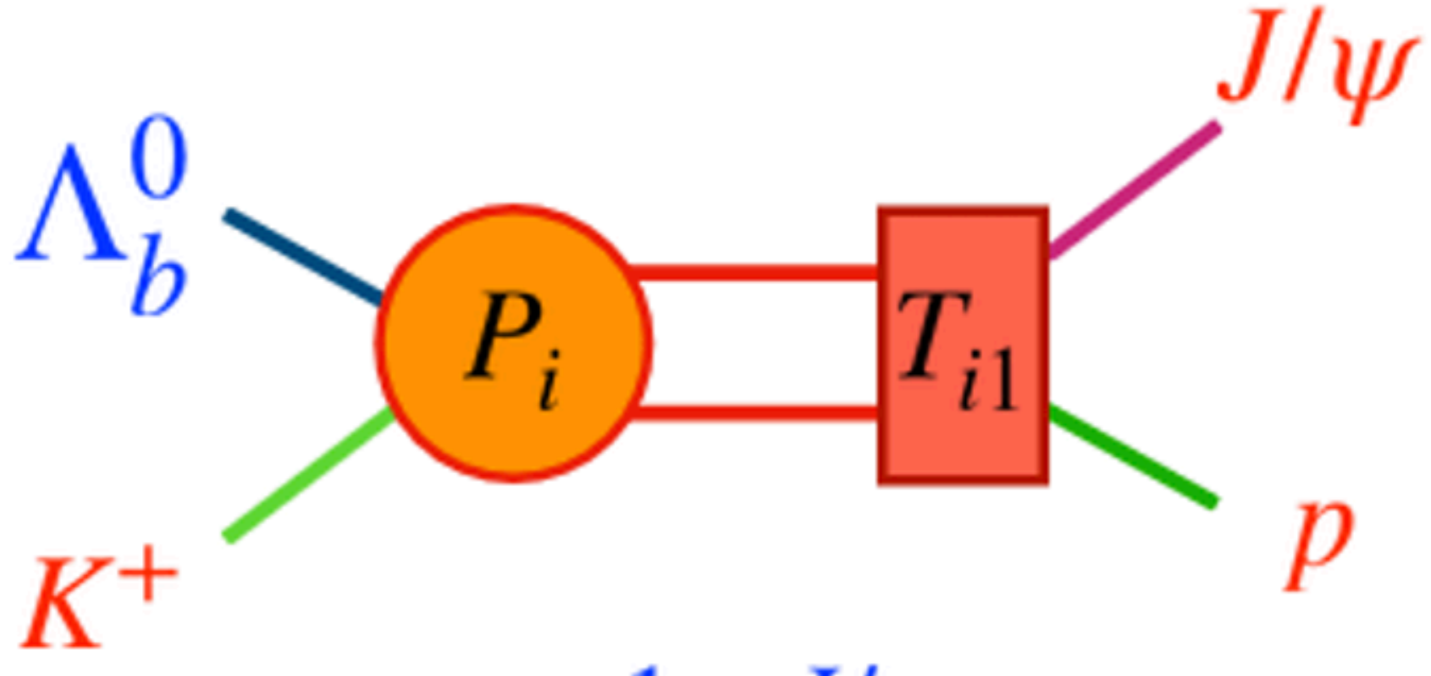


CFR et al (JPAC) PRL 123 (2019) 092001

Near-threshold model (two channels)

$$\frac{dN}{d\sqrt{s}} = \rho(s)\left[\,|F(s)|^2 + B(s)\right]$$

$$F(s) =$$



$1 : J/\psi p$

$2 : \Sigma_c^+ \bar{D}^0$

$$F(s) = P_1(s)T_{11}(s) \qquad \left(T^{-1}\right)_{ij} = M_{ij} - ik_i\delta_{ij}$$

Inverse of the scattering length

↓

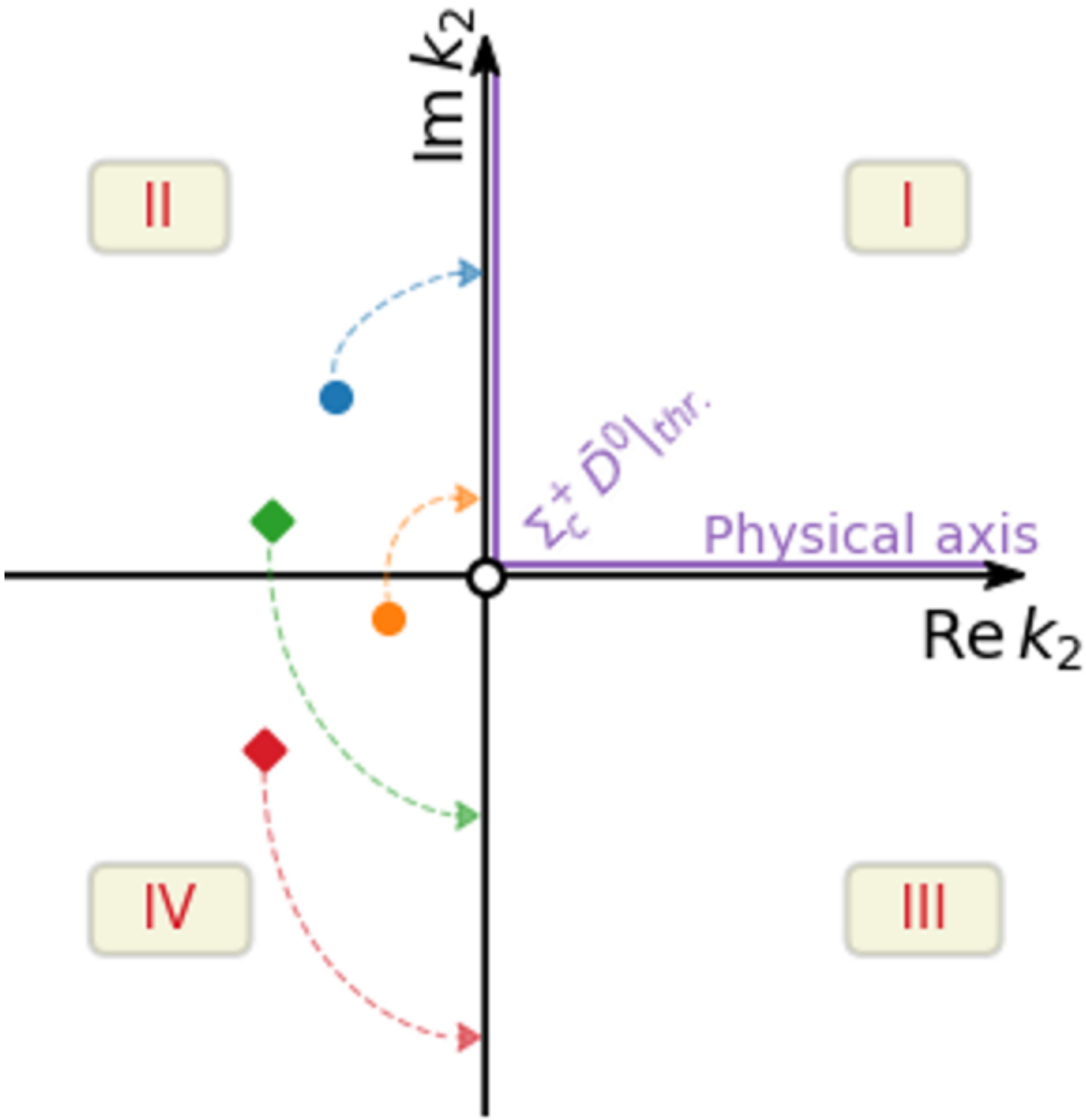
$$M_{ij}(s) = m_{ij}$$

Matrix elements M_{ij} are singularity free and can be Taylor expanded

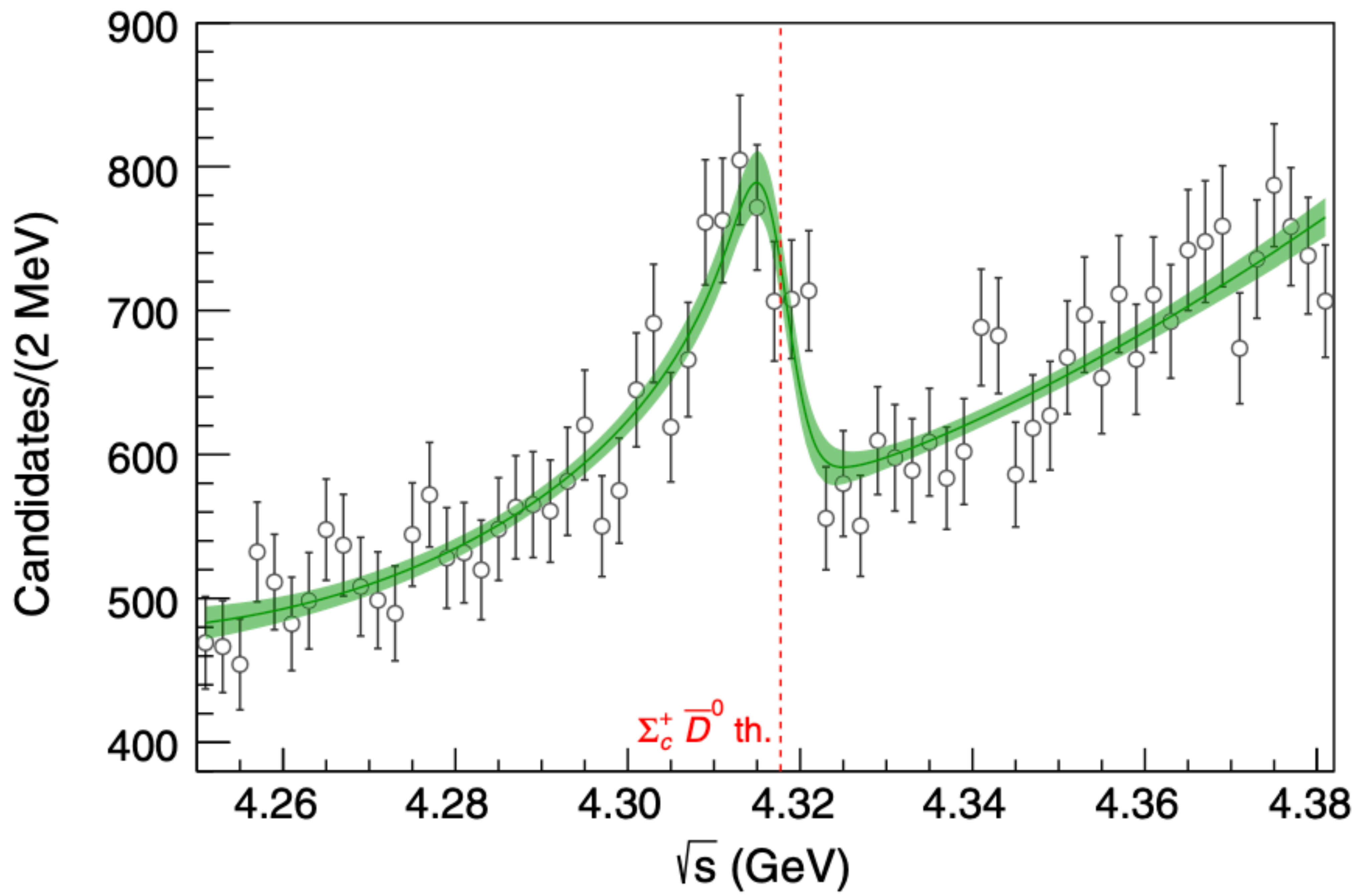
Frazer, Hendry, PR134 (1964) B1307

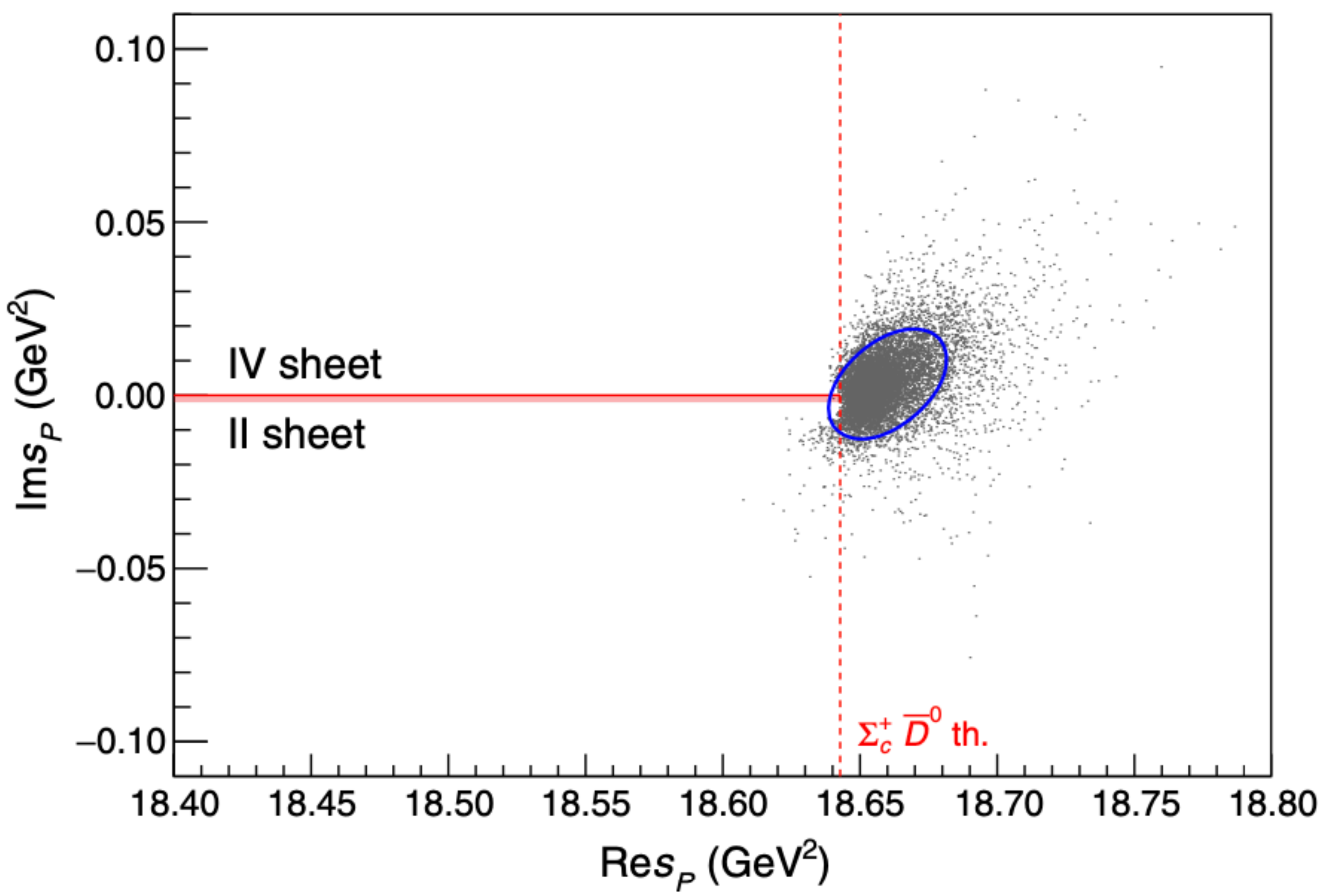
Virtual and bound states

- Bound state on IV RS: b|4
- Virtual state on IV RS: v|4
- Bound state on II RS: b|2
- Virtual state on II RS: v|2



Result





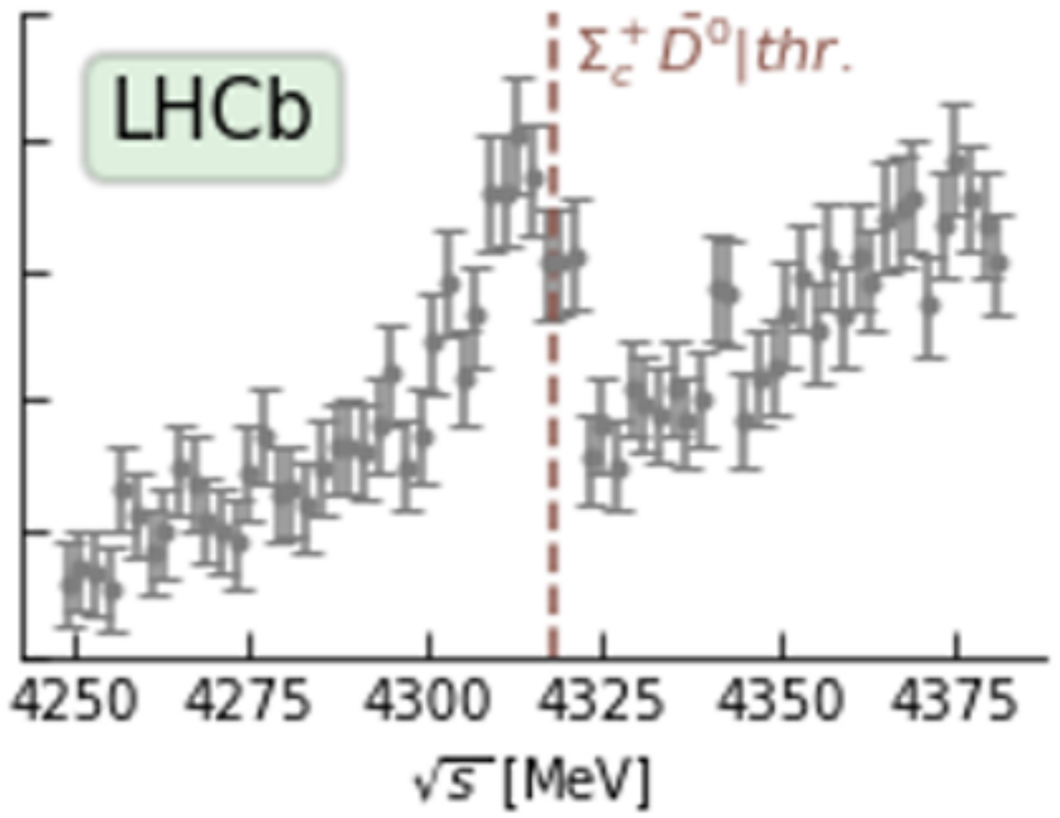
Interpretation obtained:
Virtual state on IV RS (v|4)

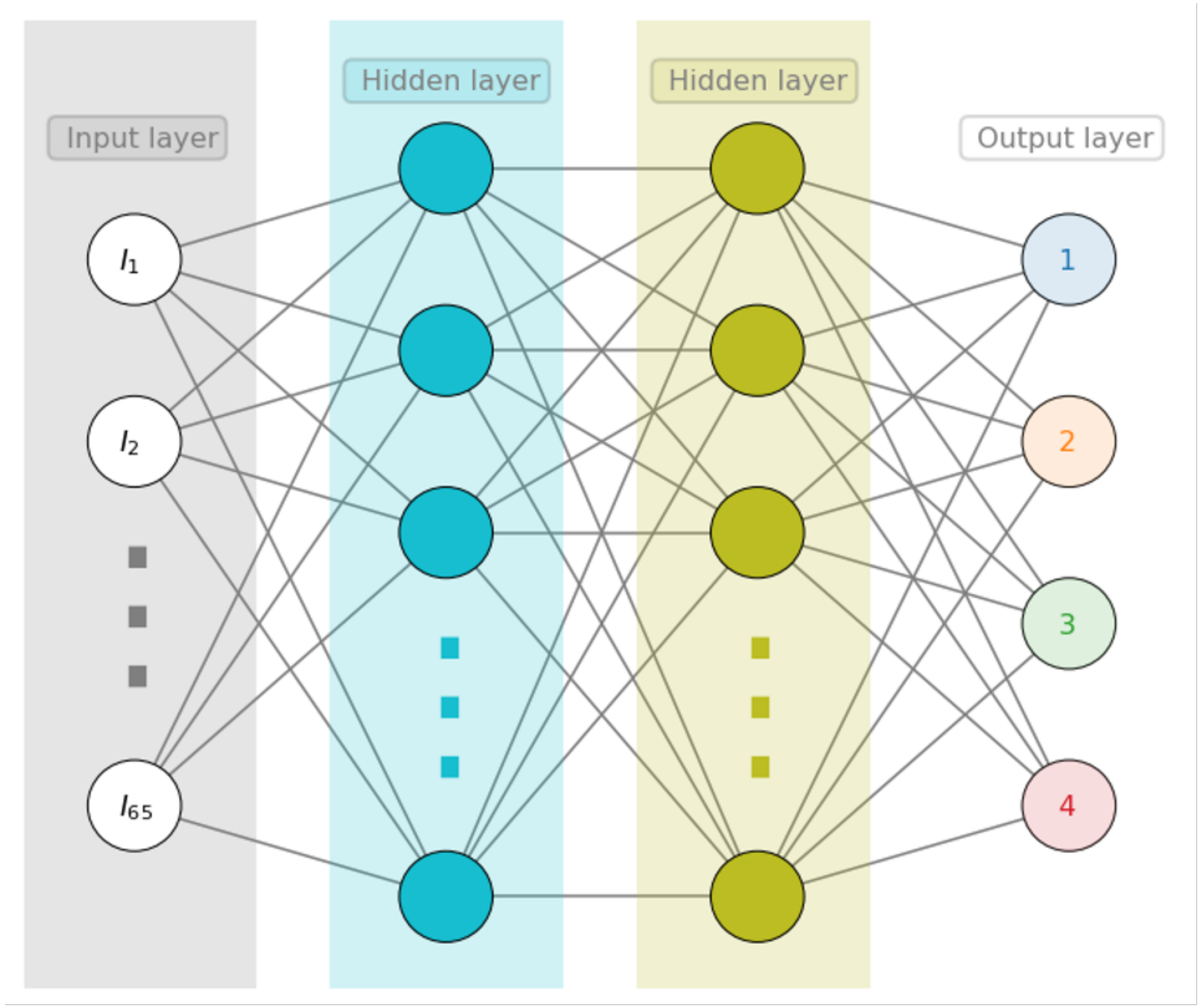
$M = 4319.7 \pm 1.6$ MeV

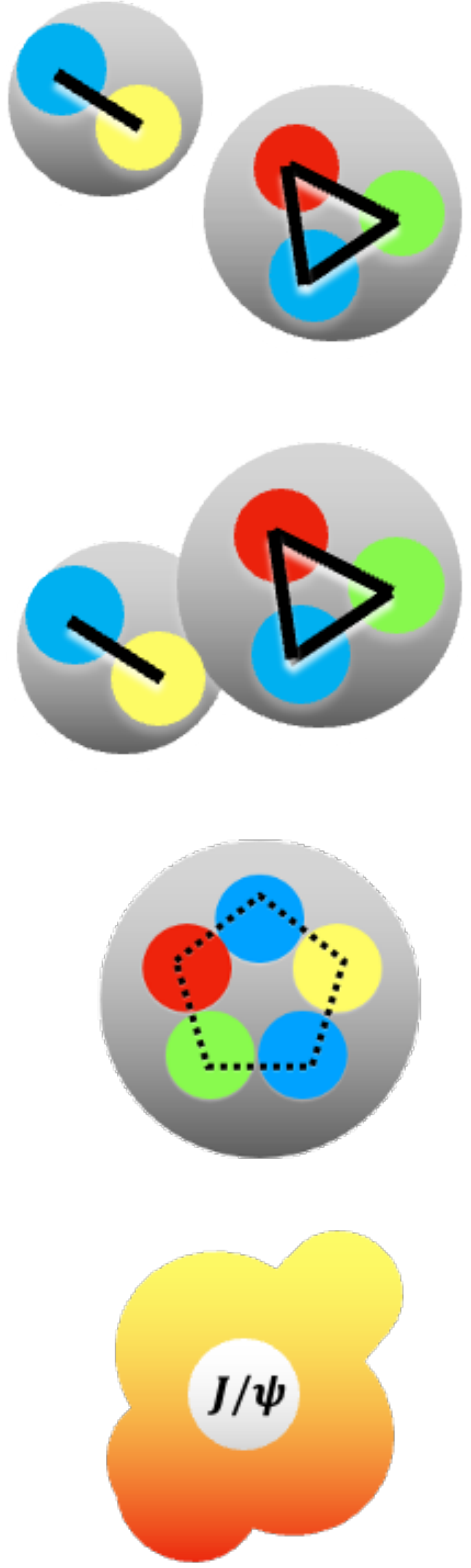
$\Gamma = -0.8 \pm 2.4$ MeV

Neural networks

Holy Grail: AI as a tool for physics discovery

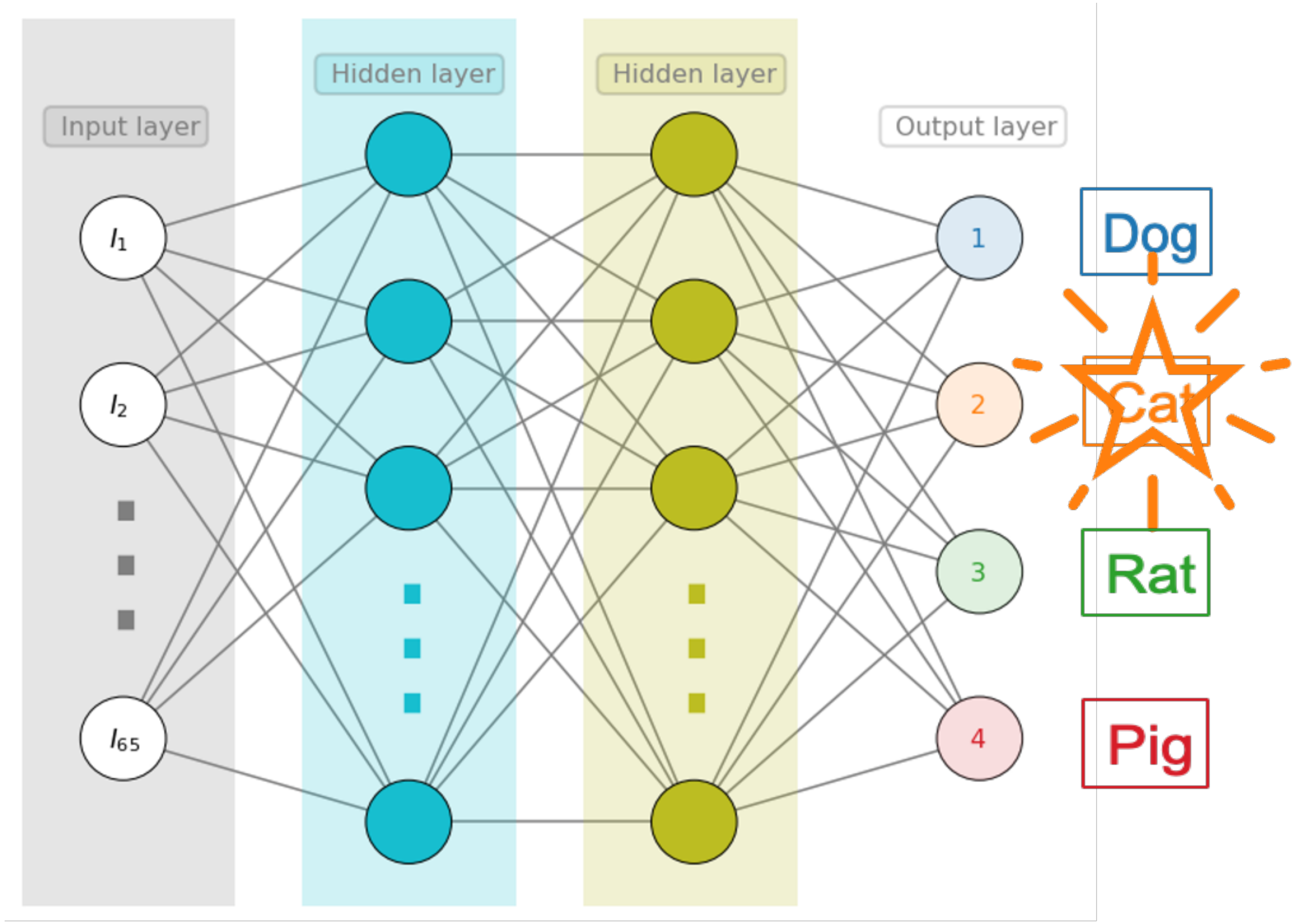






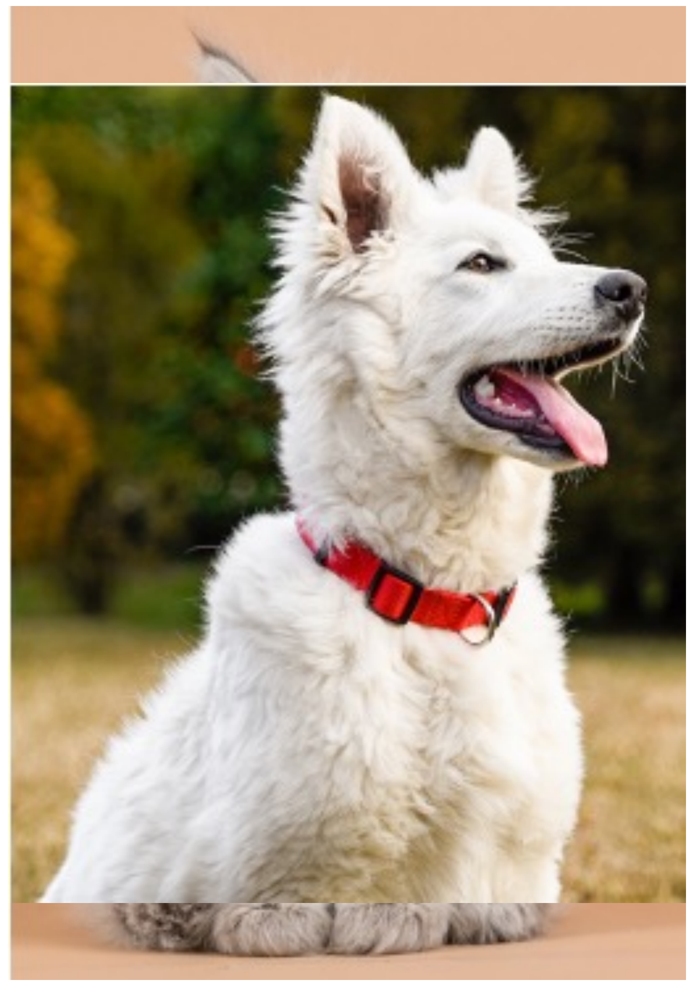
Neural networks as classifiers

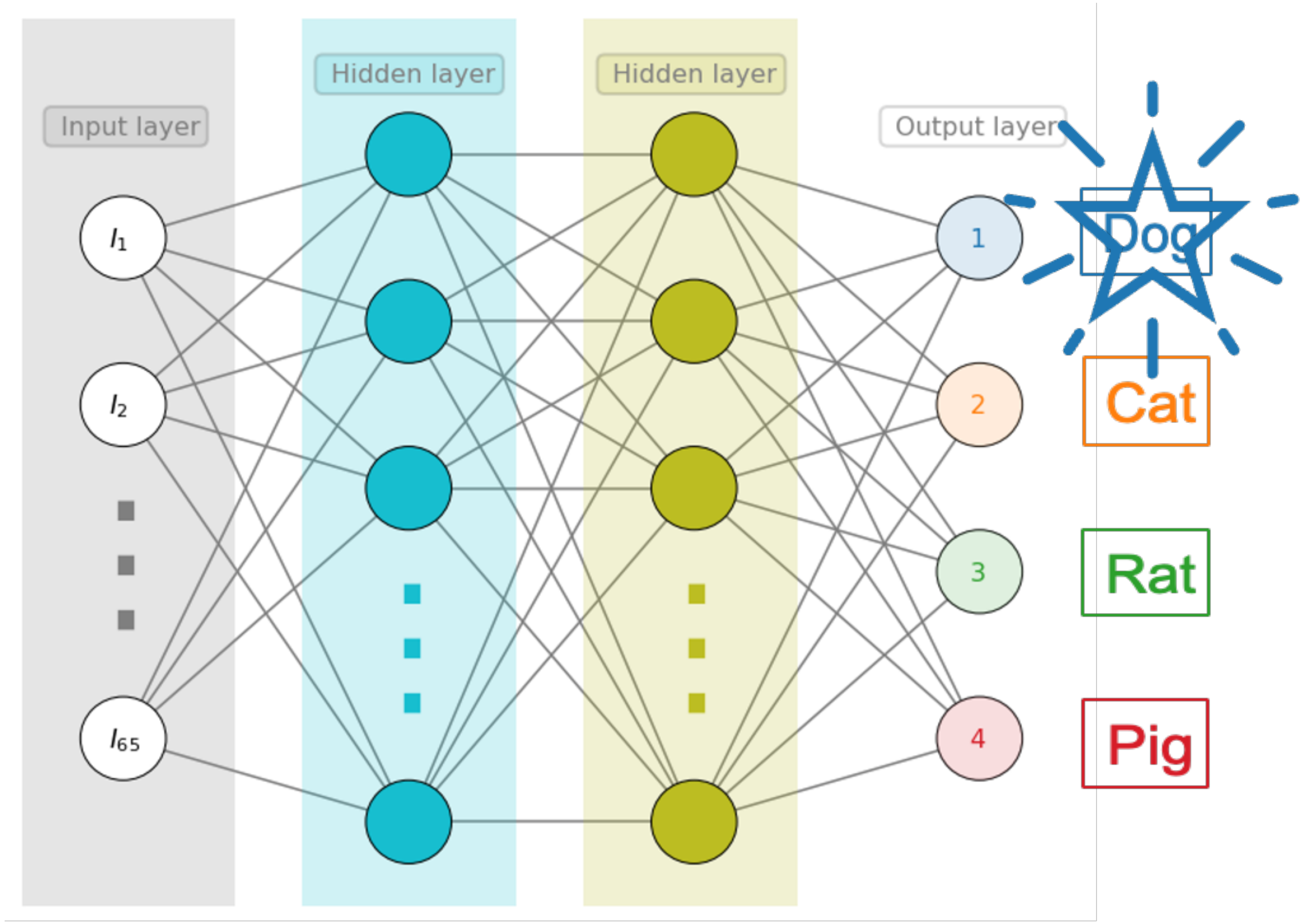
Training



Neural networks as classifiers

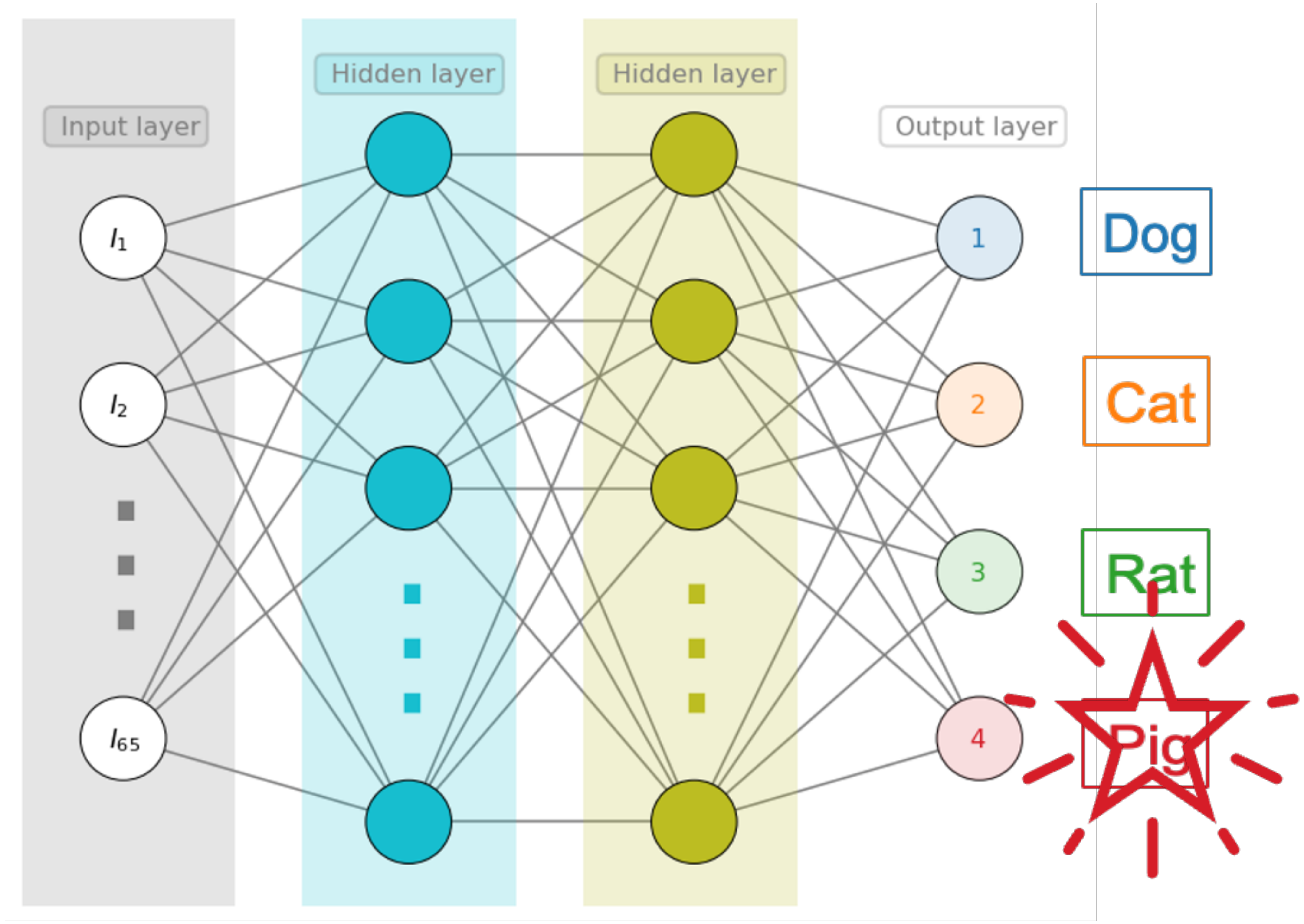
Training





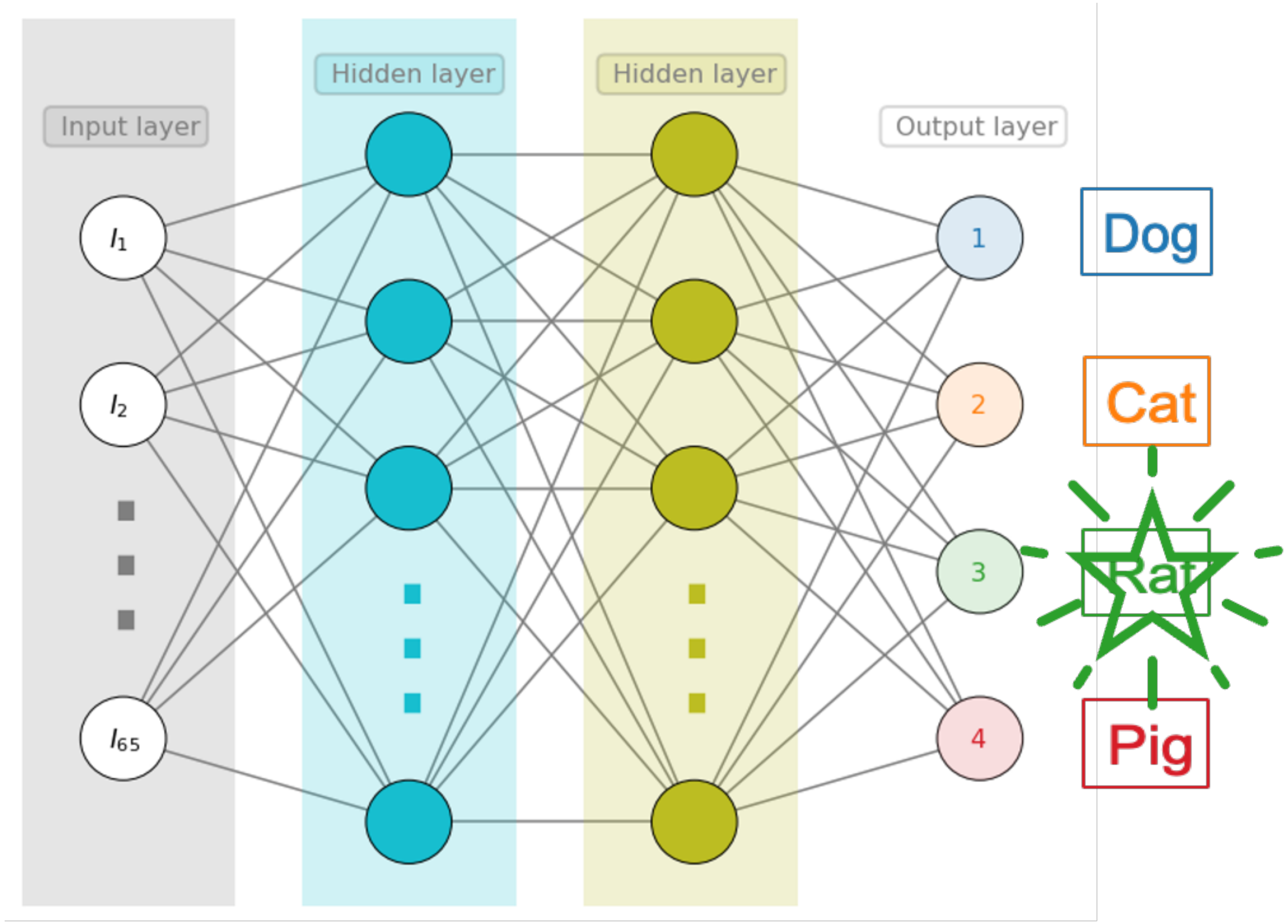
Neural networks as classifiers

Training



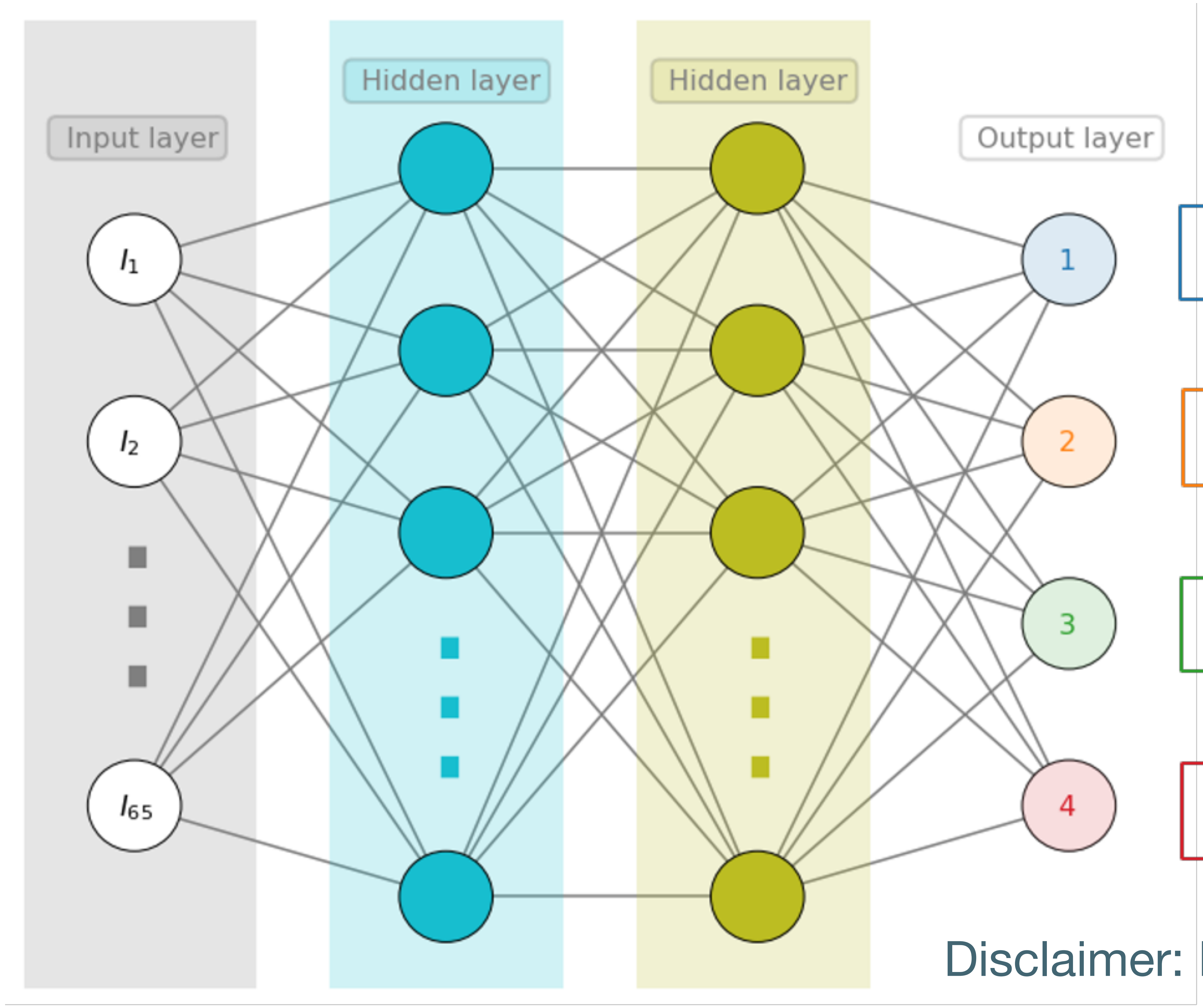
Neural networks as classifiers

Training



Neural networks as classifiers

New picture



Dog	3%
Cat	96.8%
Rat	0.1%
Pig	0.1%

Disclaimer: Made up percentages

Warning! Be aware of bias

- What you get depends completely on your training
- If you train your network to identify {🐶,🐱,🐷,🐹} don't expect it to identify 🐼!
- The neural network does no know what a 🐼 is

Can machine learning help us?

- The question:
  - Can we train a neural network to analyze a lineshape and get as a result what is the probability of each possible characterization?
- First explorations of neural networks as classifiers for hadron spectroscopy
  - *Sombillo et al. PRD 102 (2020) 016024,104 (2021) 036001*
- If possible...
  - What other information can we gain by using machine learning techniques?
- Benchmark case
  - The Pc(4312) lineshape: *Ng et al (JPAC) PRD 105 (2022) L091501*
- Still far away from answering the question but we are advancing

Building a benchmark

Building a benchmark

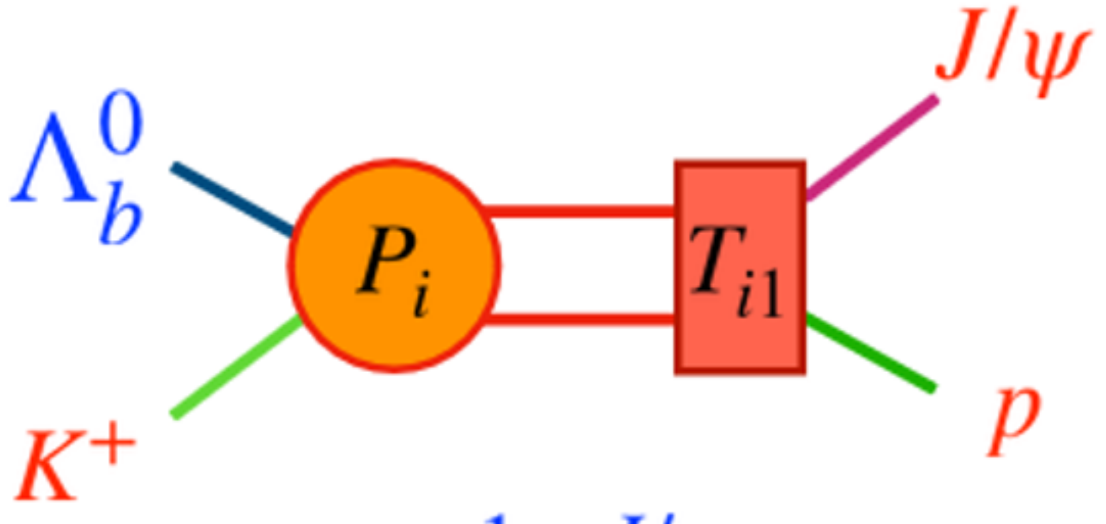
- We shoose a model that we fully uderstand to teach the NN about lineshapes
- Simple enough to perfomr comparison between standard and NN approaches
- We use the model on data that we know very well
- Implement uncertainties in both the training and the data analysis

Ng et al (JPAC) PRD 105 (2022) L091501

Model for the training set

$$\frac{dN}{d\sqrt{s}} = \rho(s)\left[\,|F(s)|^2 + B(s)\right]$$

$F(s) =$



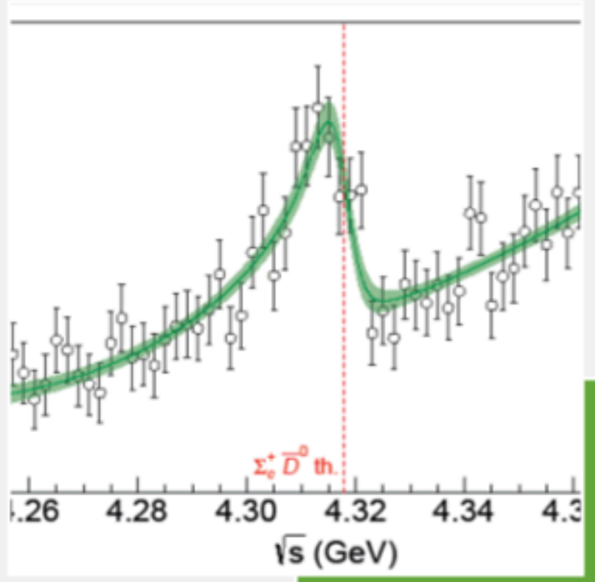
$1 : J/\psi p$

$2 : \Sigma_c^+ \bar{D}^0$

$$F(s) = P_1(s) T_{11}(s) \qquad \left(T^{-1}\right)_{ij} = M_{ij} - i k_i \delta_{ij}$$

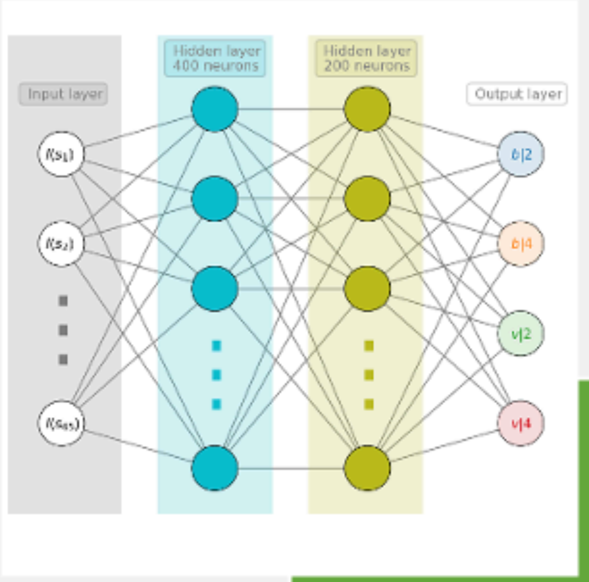
$$M_{ij}(s) = m_{ij}$$

Dictionary



AMPLITUDE ANALYSIS

a. Datapoints (lineshape)
b. Model convoluted with experimental resolution
c. Experimental uncertainties
d. Physical interpretation
e. Objective (minimization) function

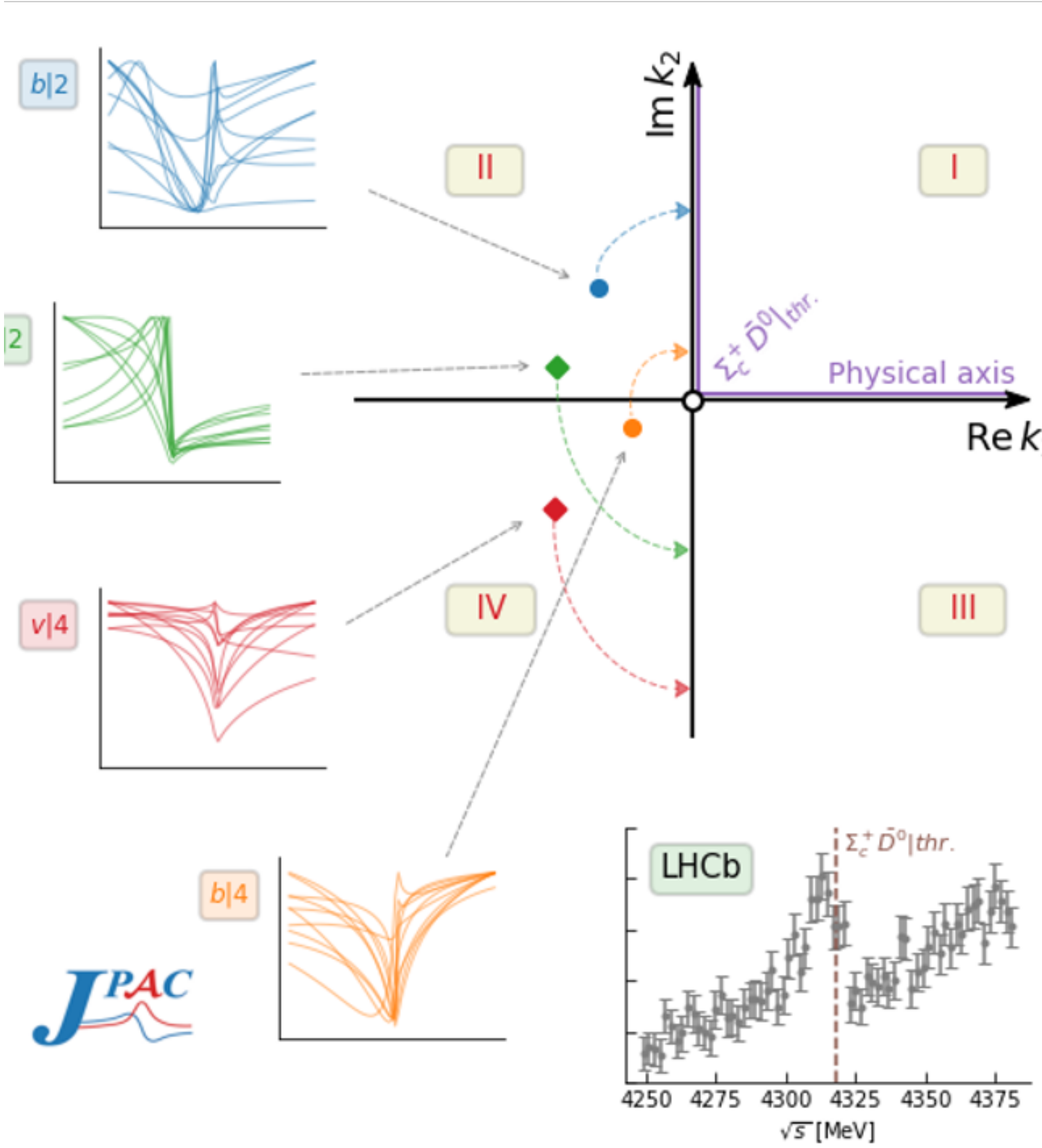


MACHINE LEARNING

a. Features
b. Training set with noise and convolution
c. Bootstrap
d. Classes
e. Optimize the NN (weights)

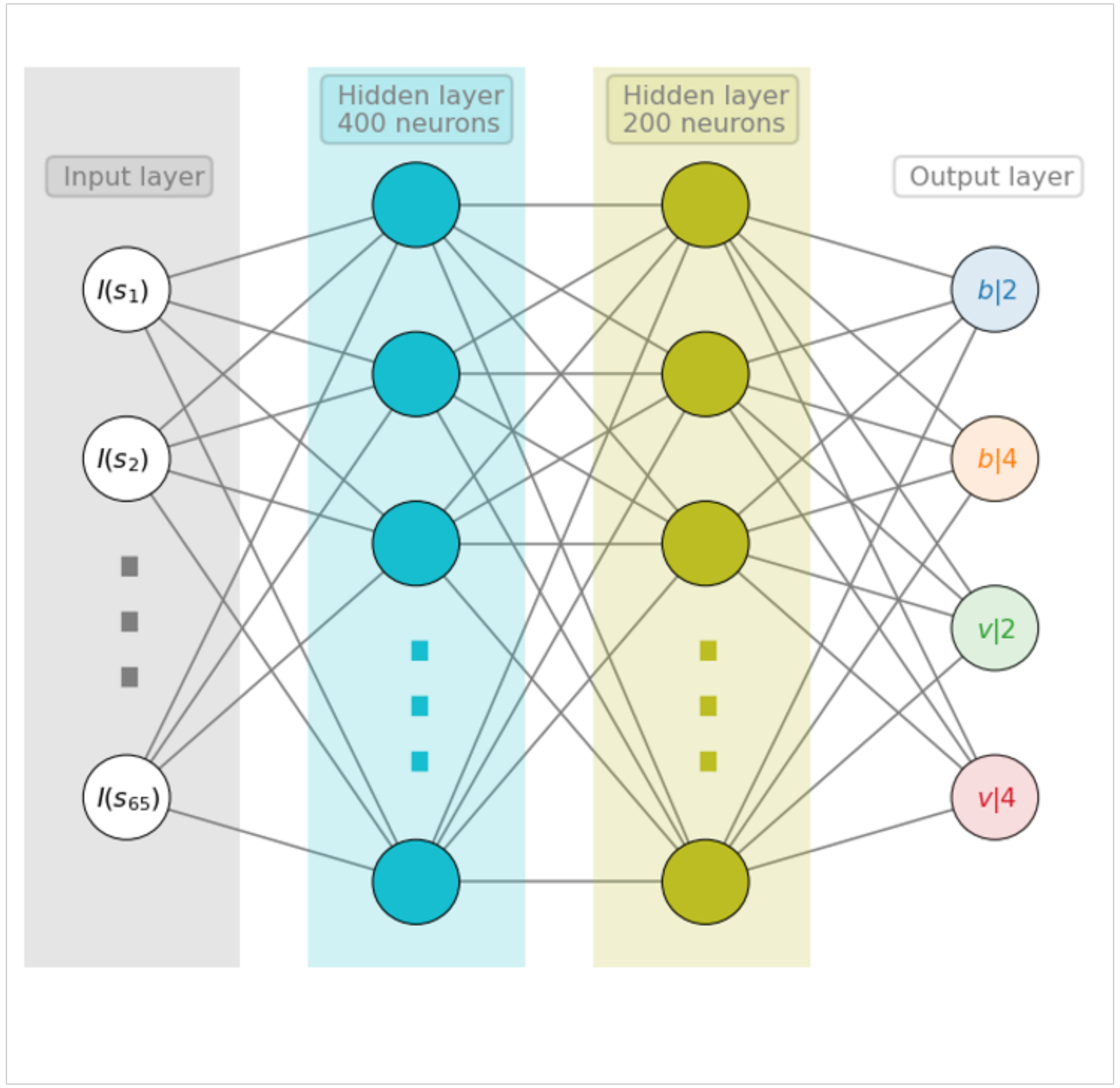
Building the training set

- 10^5 training curves
- Generated by randomly setting parameter values in a wide range
- Curves are computed at the experimental energies
- The lineshapes are convoluted with the experimental resolution
- Gaussian noise included to mimic uncertainties
- Compare "blurry to blurry"

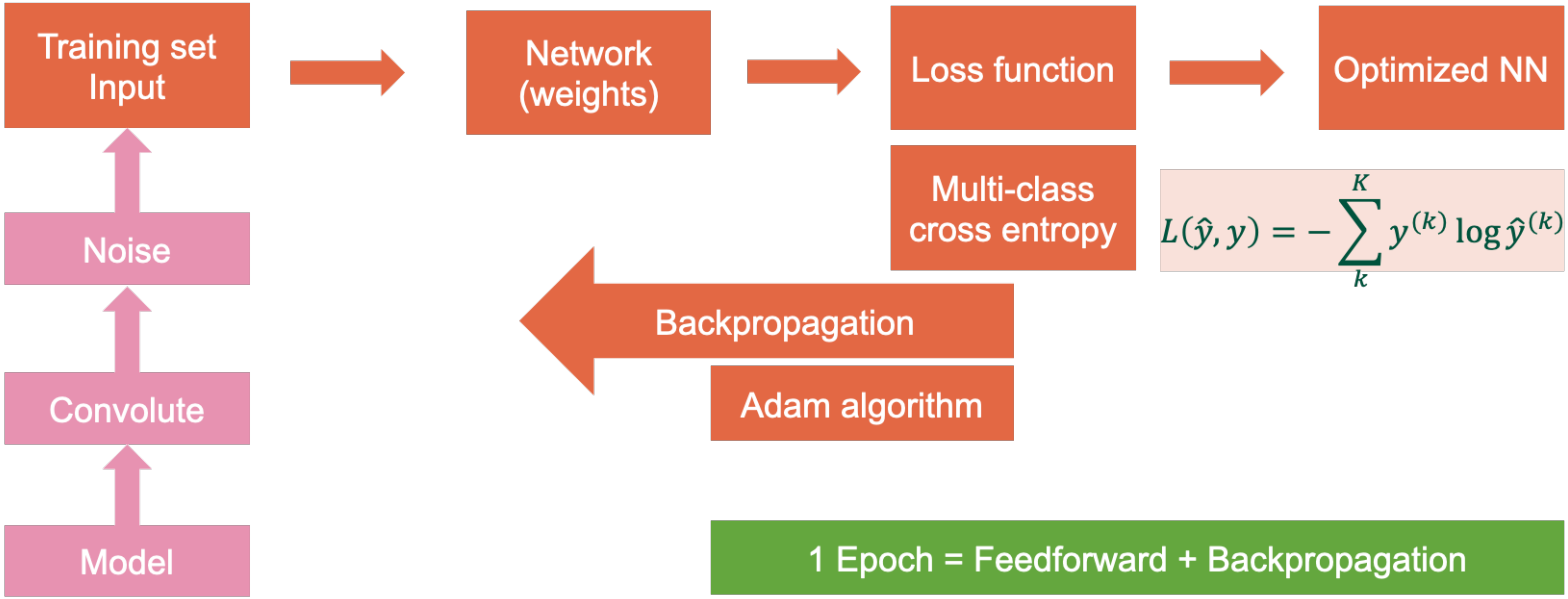


Neural network architecture

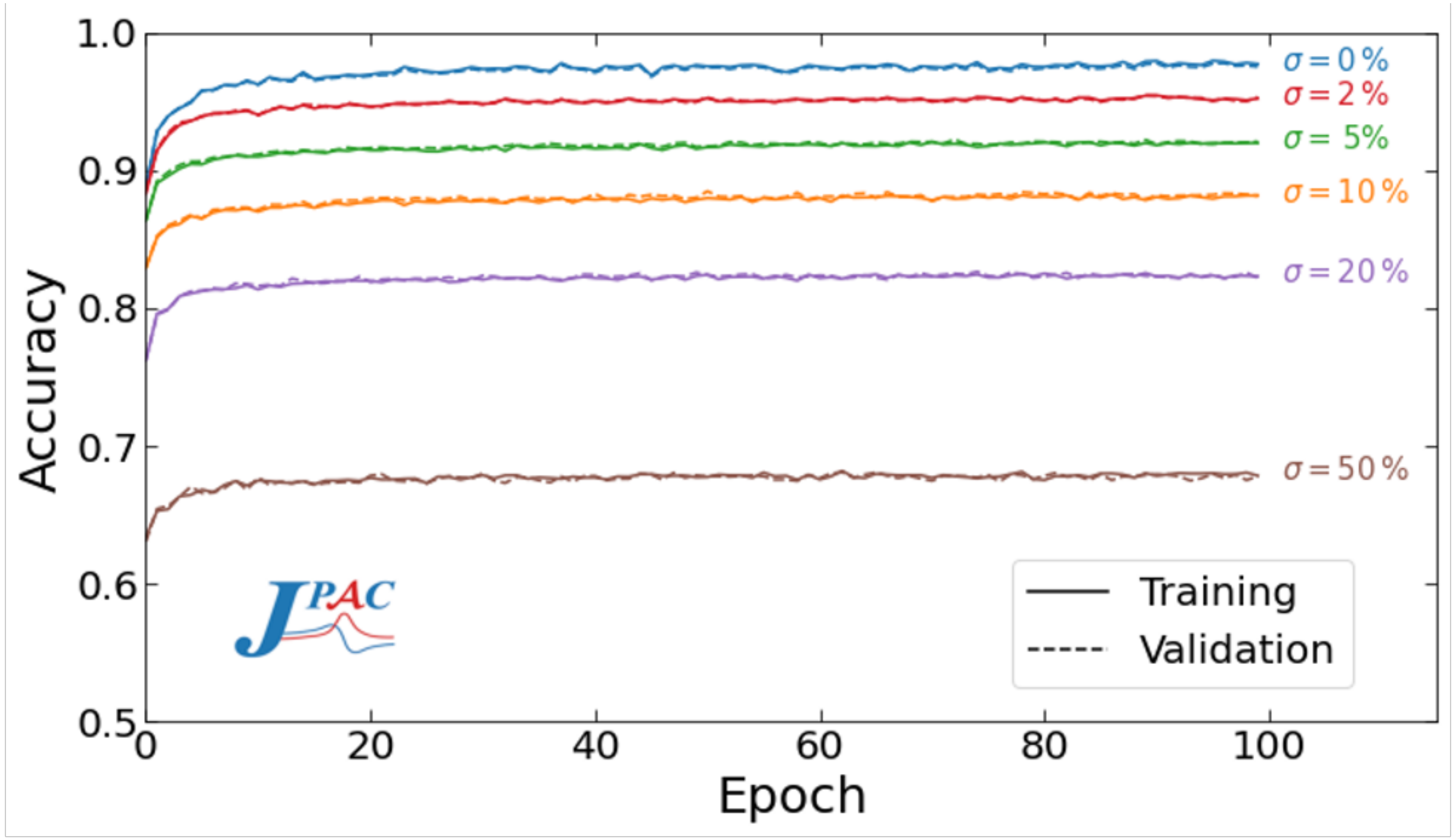
Layer	Input	Operation	Output
Input layer			(B,65)
Dense	(B,65)		(B,400)
Dropout	(B,400)	20%	(B,400)
ReLu	(B,400)	$f(x) = \max(0, x)$	(B,400)
Dense	(B,400)		(B,200)
Dropout	(B,200)	50%	(B,200)
ReLu	(B,200)	$f(x) = \max(0, x)$	(B,200)
Dense	(B,200)		(B,4)
Output layer	(B,4)	$p(x_i) = e^{x_i} / \sum e^{x_j}$	(B,4)

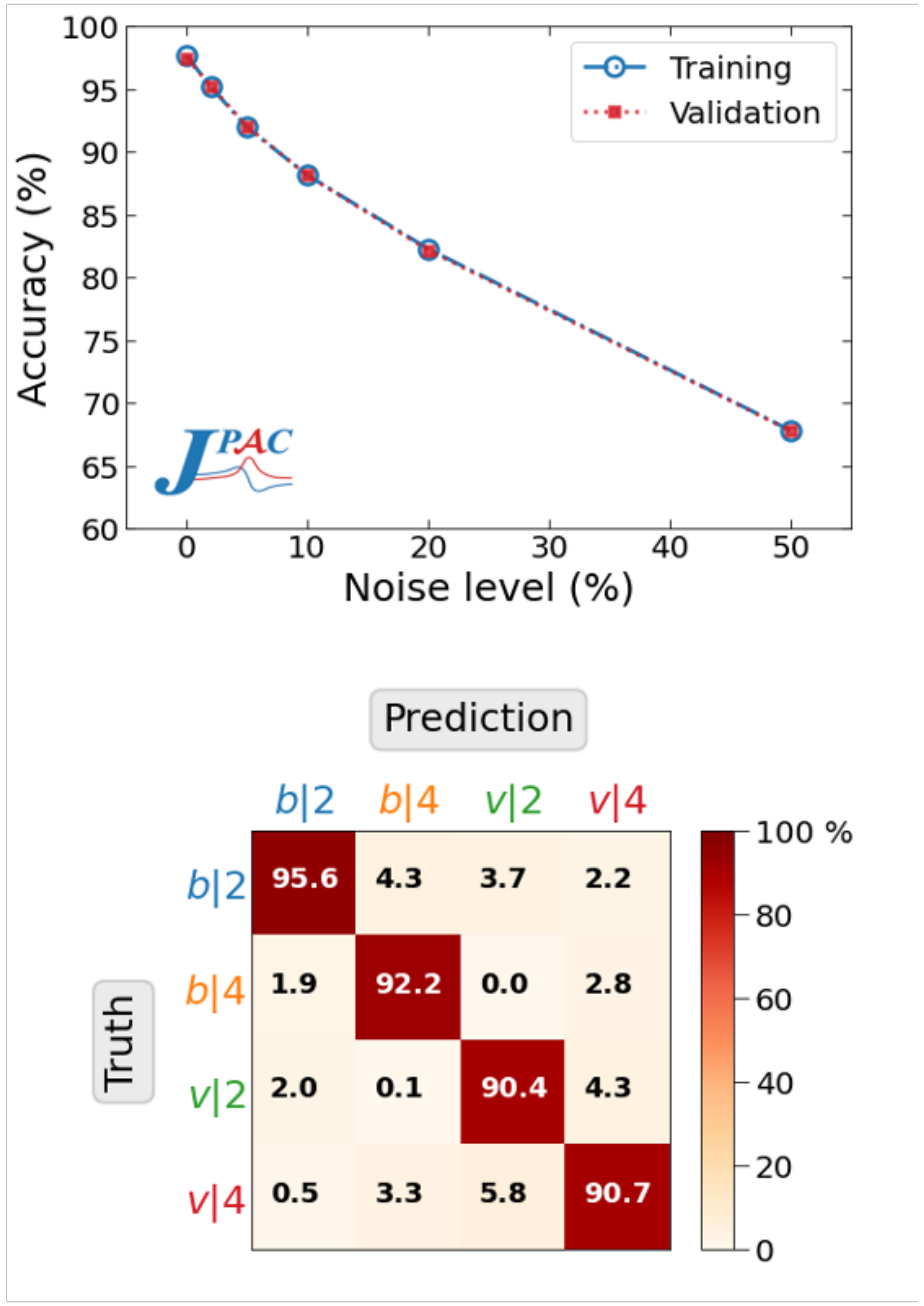


Optimizing (training) the neural network: fitting weights



Training


1.0
0.9
0.8
0.7
0.6
0.5
Accuracy
0
20
40
60
80
100
Epoch
σ = 0 %
σ = 2 %
σ = 5%
σ = 10 %
σ = 20 %
σ = 50 %
Training
Validation
JPAC

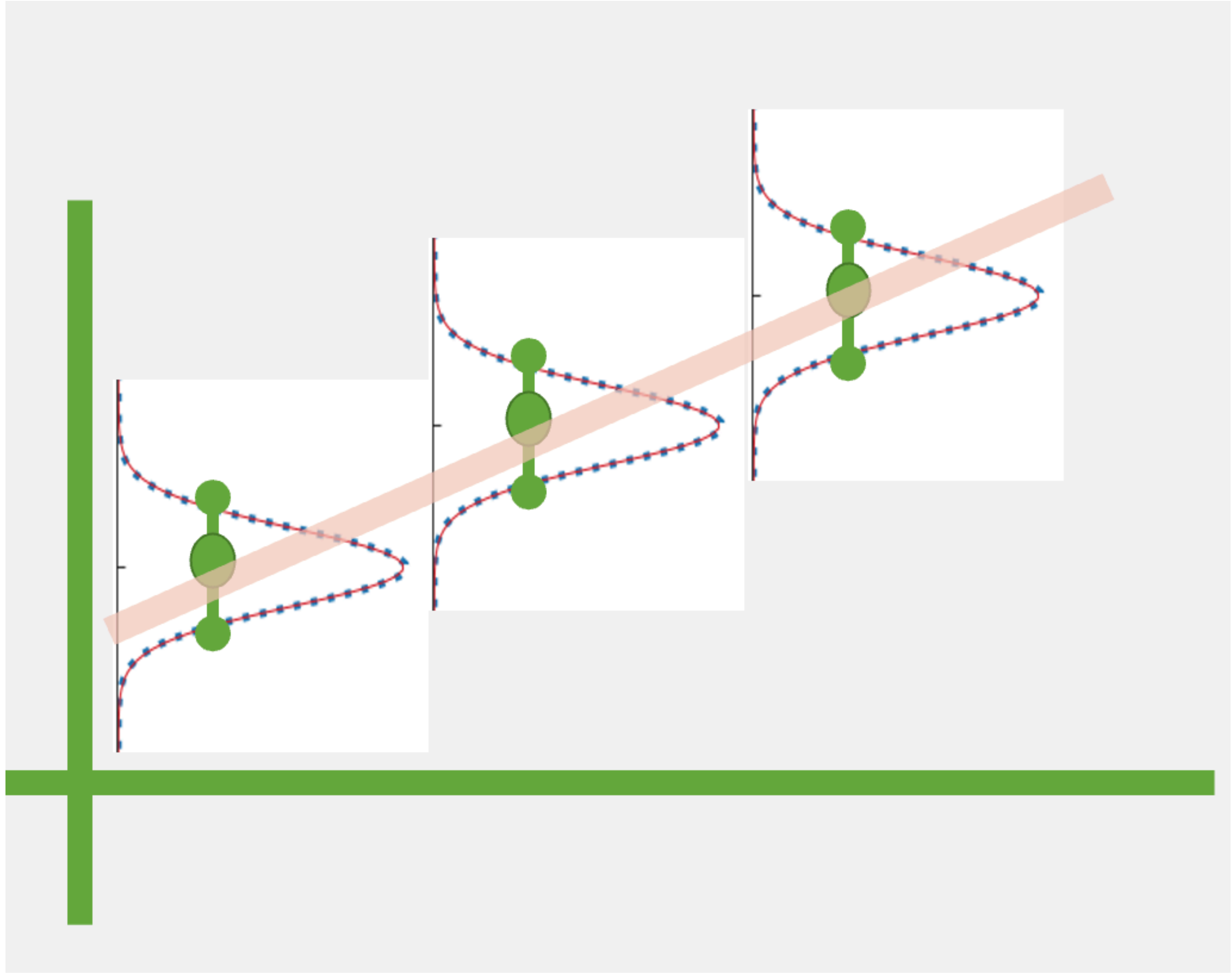

100
95
90
85
80
75
70
65
60
Accuracy (%)
0
10
20
30
40
50
Noise level (%)
Training
Validation
JPAC
Prediction
Truth
b|2
b|4
v|2
v|4
95.6 4.3 3.7 2.2
1.9 92.2 0.0 2.8
2.0 0.1 90.4 4.3
0.5 3.3 5.8 90.7
100 %
80
60
40
20
0

Experimental uncertainties

- Associate a distribution to each experimental datapoint: typically a Gaussian with mean and sigma from experiment

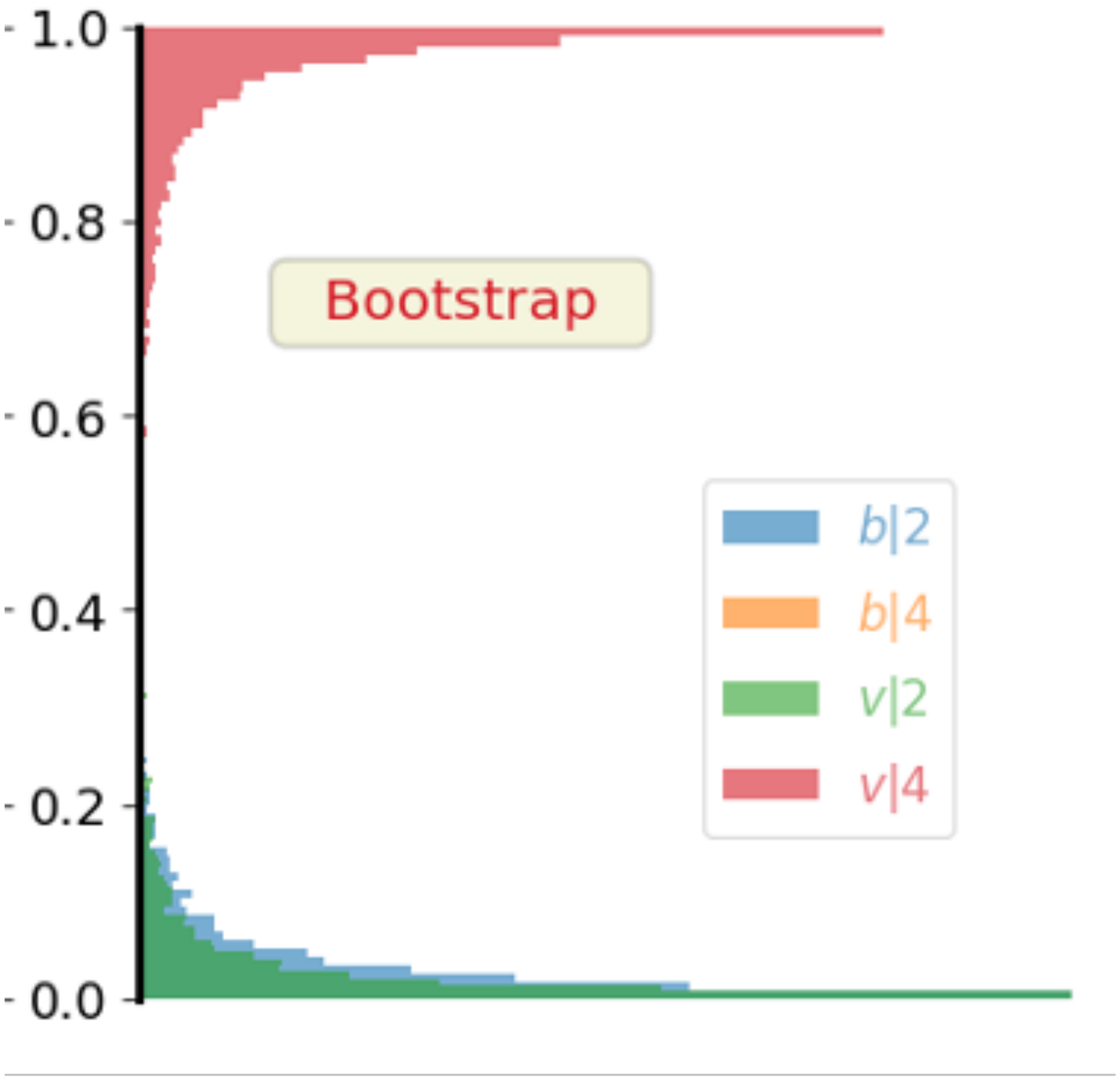
- Monte Carlo. Generate pseudodata according to the chose distribution

- Run statistics on the pseudodatasets. Compute distributions, mean, standard deviation, quantiles...

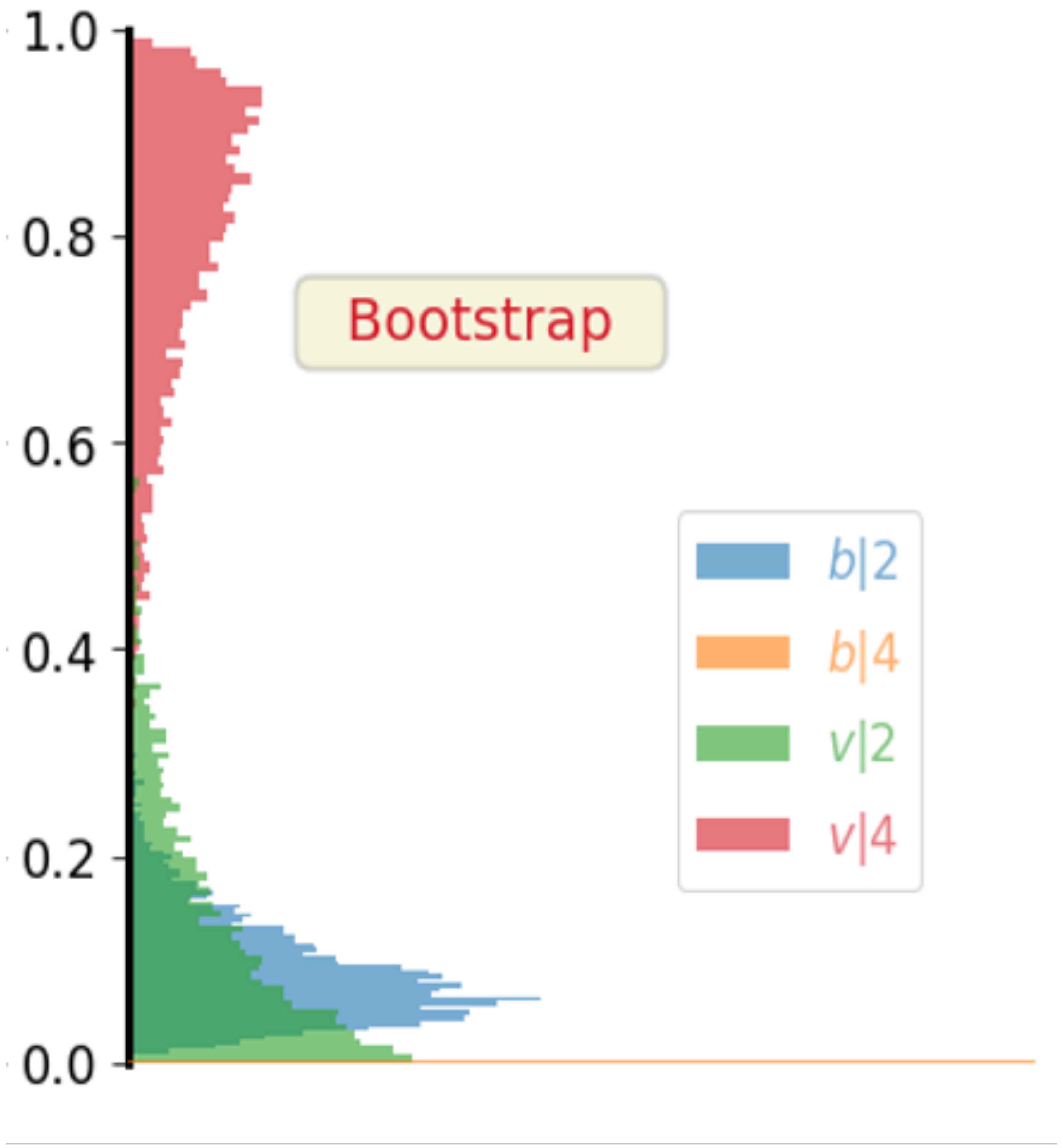


Applying NN to data

- We pass the three dataset through the NN
- Uncertainties using bootstrap
- Obtain probability distributions
- We unsurprisingly recover the same result as the standard approach



Three datasets analyzed with the same network

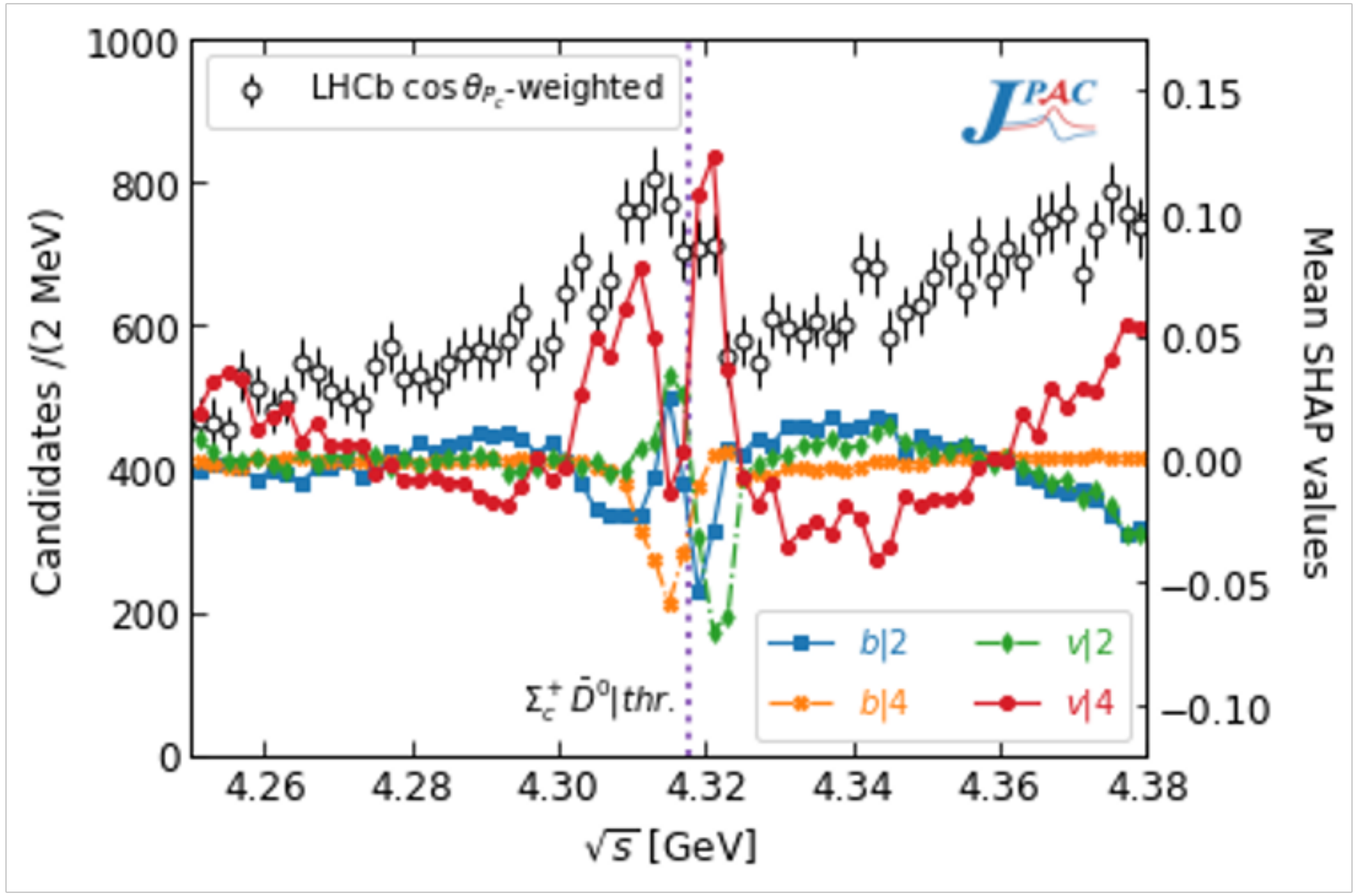


	$b\|2$	$b\|4$	$v\|2$	$v\|4$
$\cos\theta_{P_c}$-weighted	0.6%	$<$ 0.01%	1.1%	98.3%
$m_{Kp} > 1.9\,\mathrm{GeV}$	1.4%	$<$ 0.1%	1.6%	97.0%
m_{Kp} all	5.4%	$<$ 0.1%	21.0%	73.6%

Explainability

- SHAP values

- Allows to determine how a given feature in the input layer (in our case an experimental datapoint) impacts the decision made by the network in the output layer (the classes)



Takeaways

- We tested a relatively simple, ML based application
- Neural networks are not a substitution of the canonical approach to analyzing data
- Neural networks provide a way to truly compare interpretations and gain physics insight
- We are (hopefully) in the begining

Always remember