Joint Physics Analysis Center Review 2022

AI/ML@JPAC

César Fernández Ramírez

#SOMOS2030

GOBIERNO DE ESPAÑA

MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL

CONACYT

PAPIIT - PAPIME

JPAC

Instituto de Ciencias Nucleares UNAM

UNED

Standard approach to data analysis of a resonance lineshape

- Take an amplitude, it has parameters to be determined by data
- Fit data using χ^2
- Extract parameters and 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?

AI/ML

Machine Learning provides methods to learn from data to perform tasks

Widely used nowadays thanks to:

- *Improvement in algorithms and hardware*

Examples:

- *Neural networks, Random forest, Genetic algorithms, …*

Widely used in physics

- *Fitters, universal interpolators, classifiers*

Explainability

- *NNs are usually black boxes, so statistical methods have been developed to study them*

AI/ML/NNs in hadron physics

Experiments

- *Particle ID, …*

A(I)DAPT

- *Exploring NNs to extract amplitudes from data. LRDR funded*

Universal interpolators

- *NNPDF*

Regression

- *Lattice QCD*

Classifiers

- *Hadron spectroscopy (this talk)*

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., 2003.10770, 2104.141782, 2105.04898*

If possible…

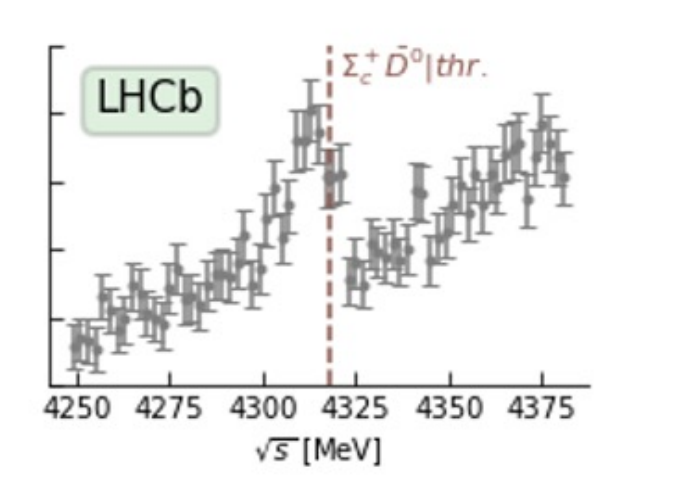
- *What other information can we gain by using machine learning techniques?*

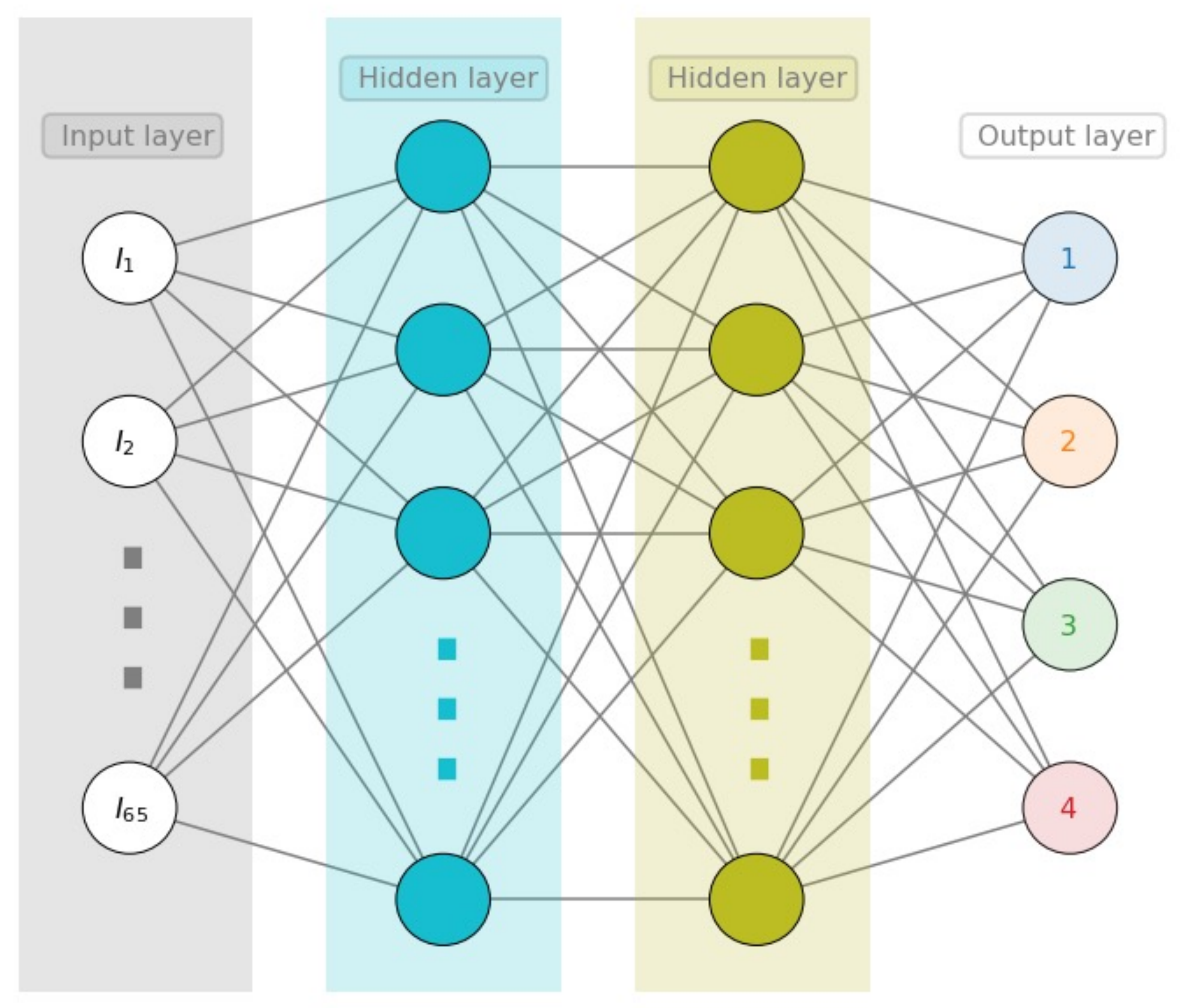
Benchmark case

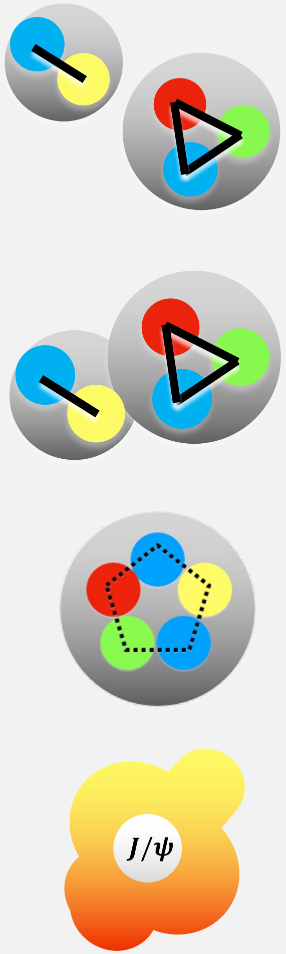
- *The $P_c(4312)$ lineshape: Ng et al. (JPAC)* 2110.13742

Still far away from answering this question but we are advancing

Holy Grail: AI as a tool for physics discovery

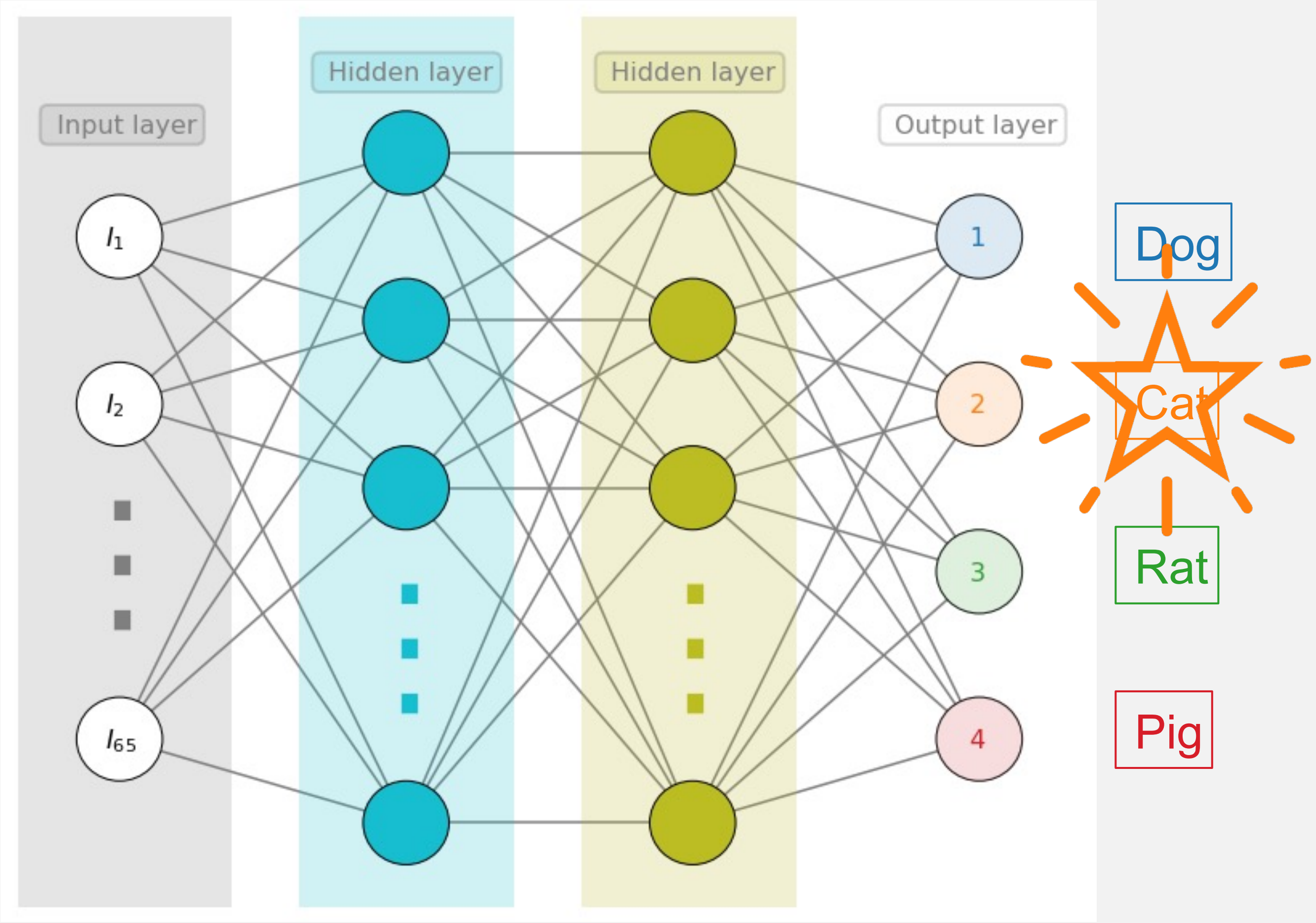






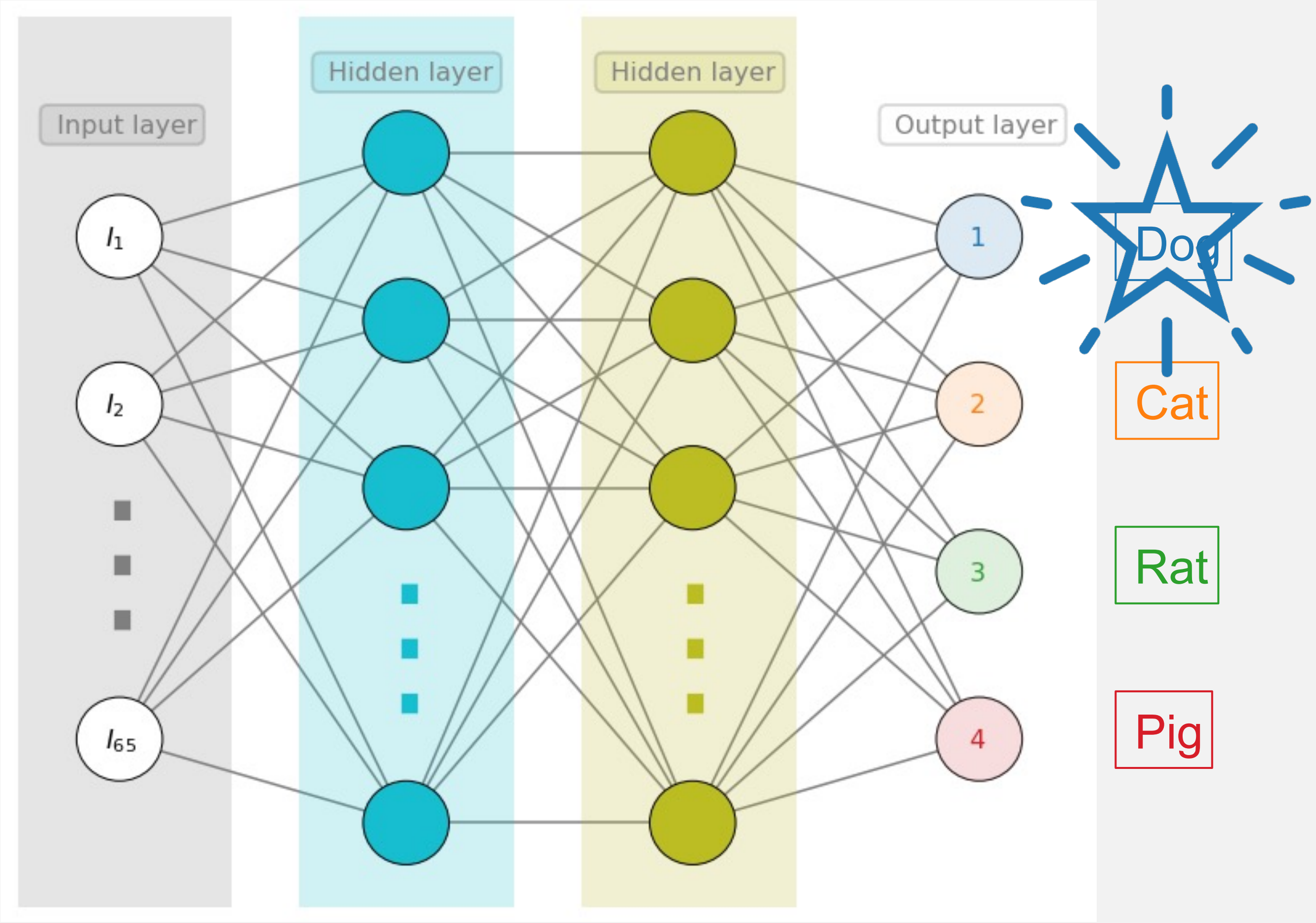
Neural networks as classifiers

Training



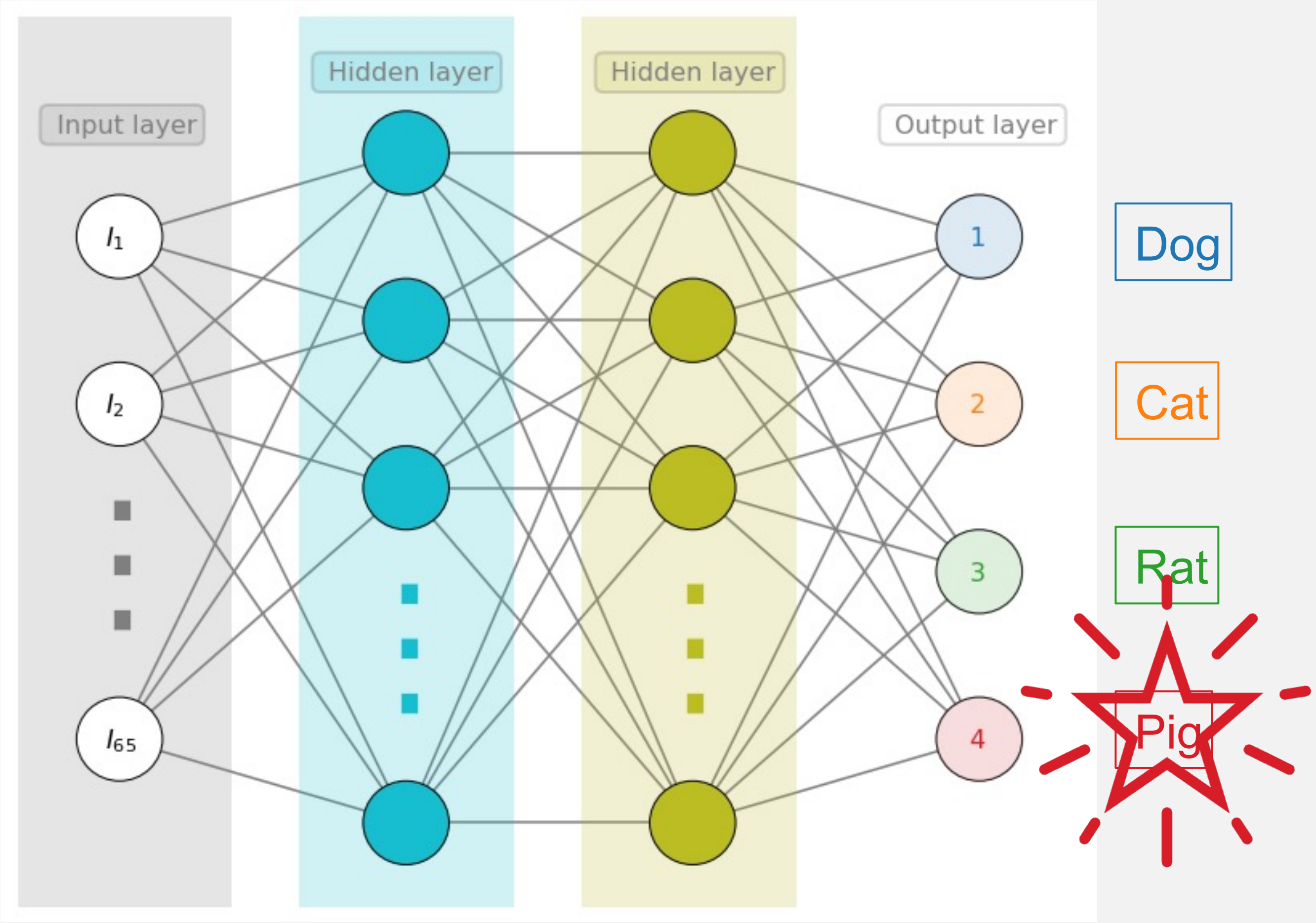
Neural networks as classifiers

Training



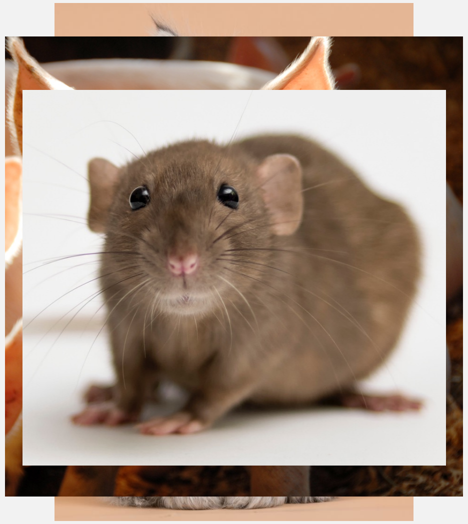
Neural networks as classifiers

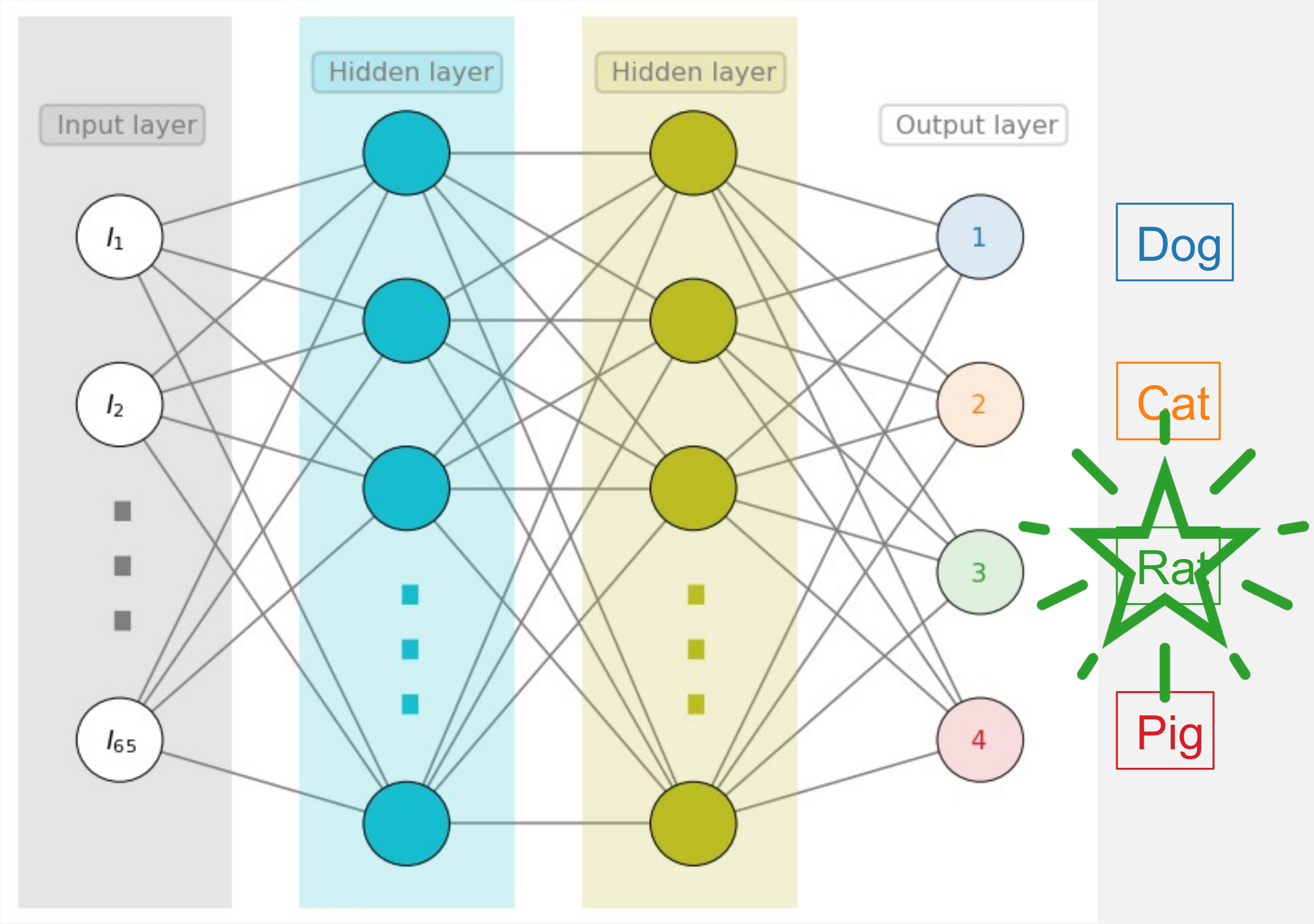
Training



Neural networks as classifiers

Training

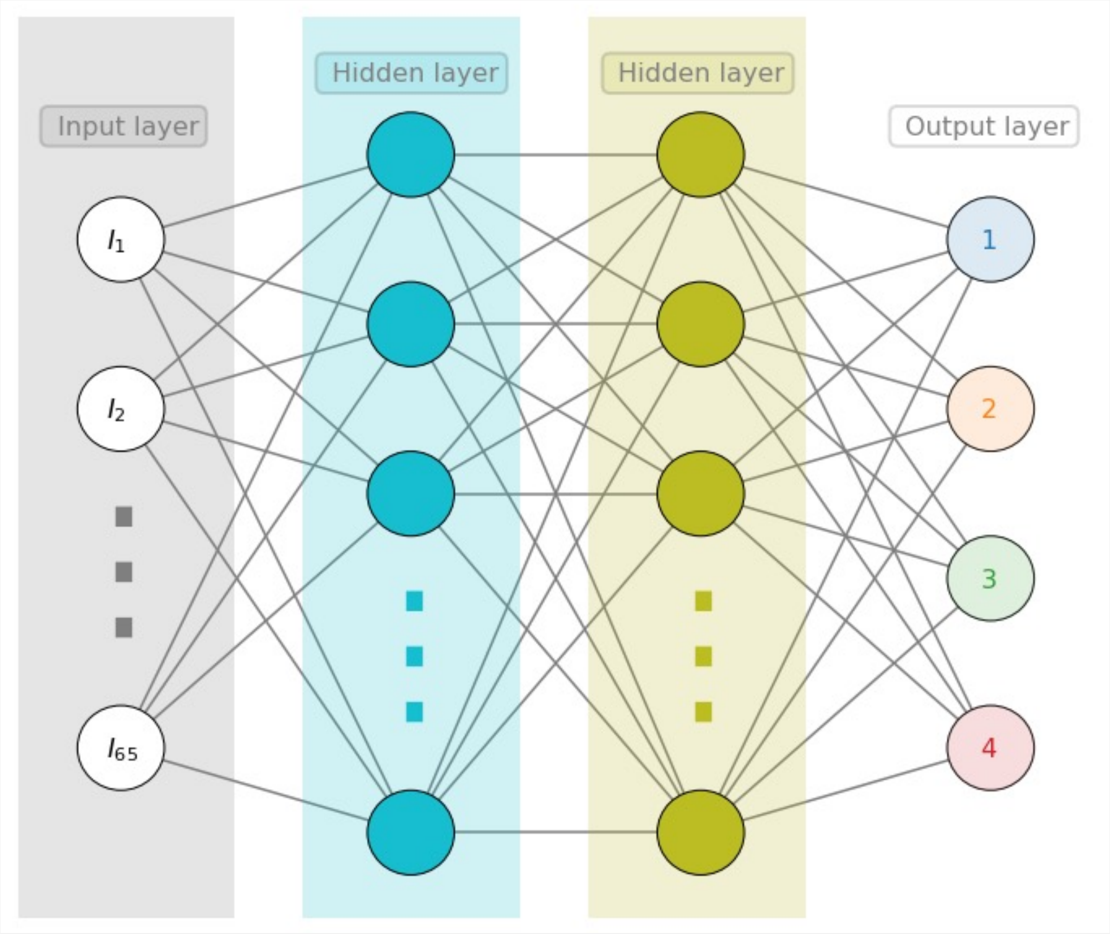




10

Neural networks as classifiers

New picture



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

Disclaimer: Made up percentages

Building a benchmark

Training the neural network

- *We choose a model that we fully understand to teach the NN about lineshapes*

Comparison

- *Simple enough to perform a direct comparison between standard and NN approaches*

Experimental data

- *We use the model on data that we know well*

Error analysis

- *Implement uncertainites both in the training and the data analysis*

Building the training set

10^5 training curves

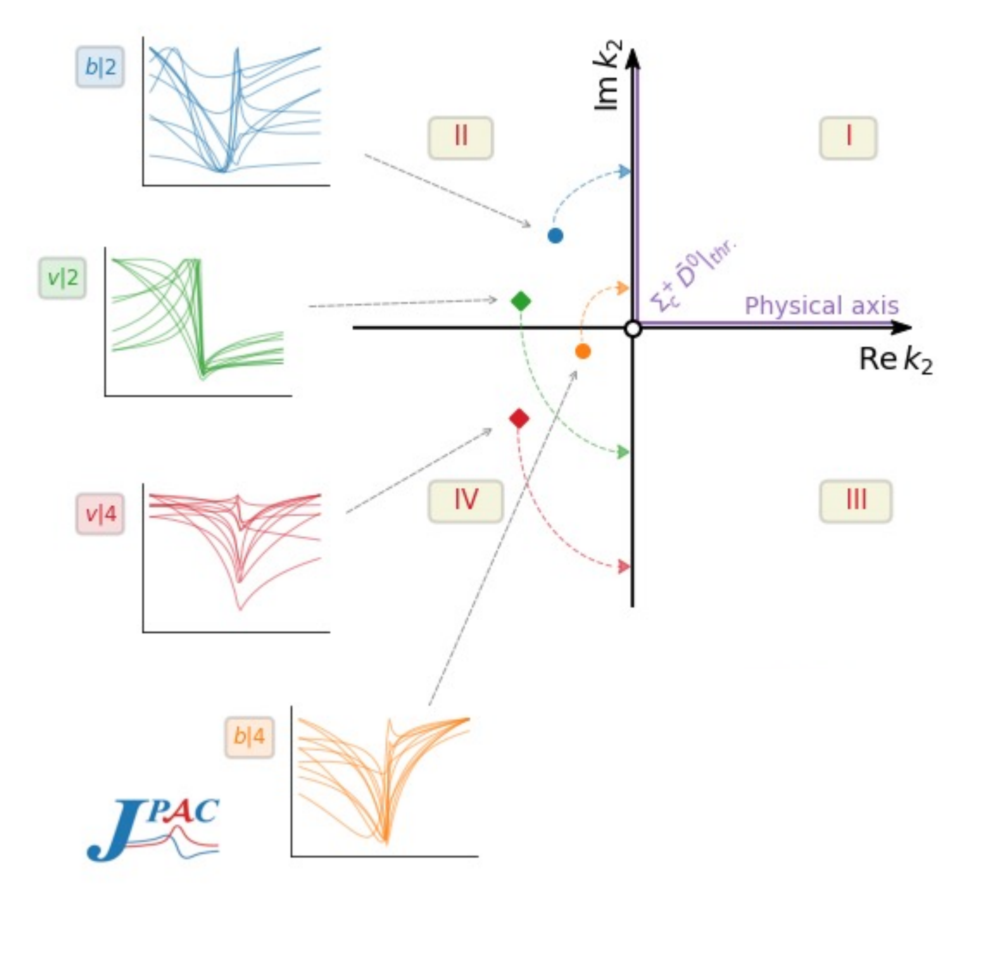
- *Generated by randomly setting parameter values in a wide range*
- *Curves are computed at the experimental energies*

Convolution

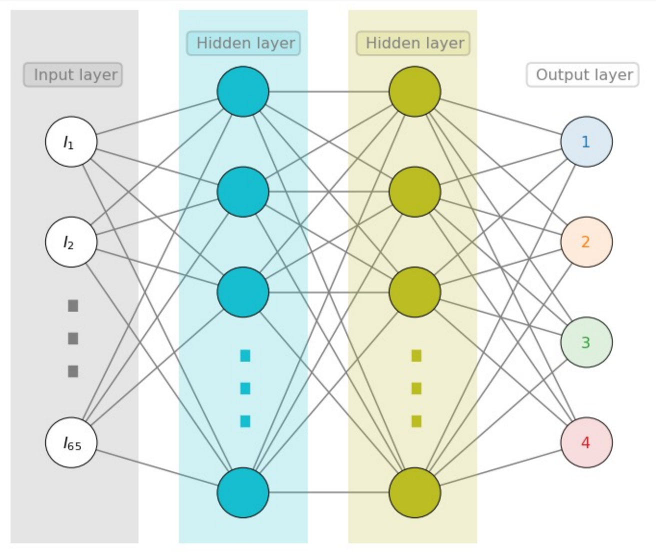
- *Model is convoluted with the experimental resolution*

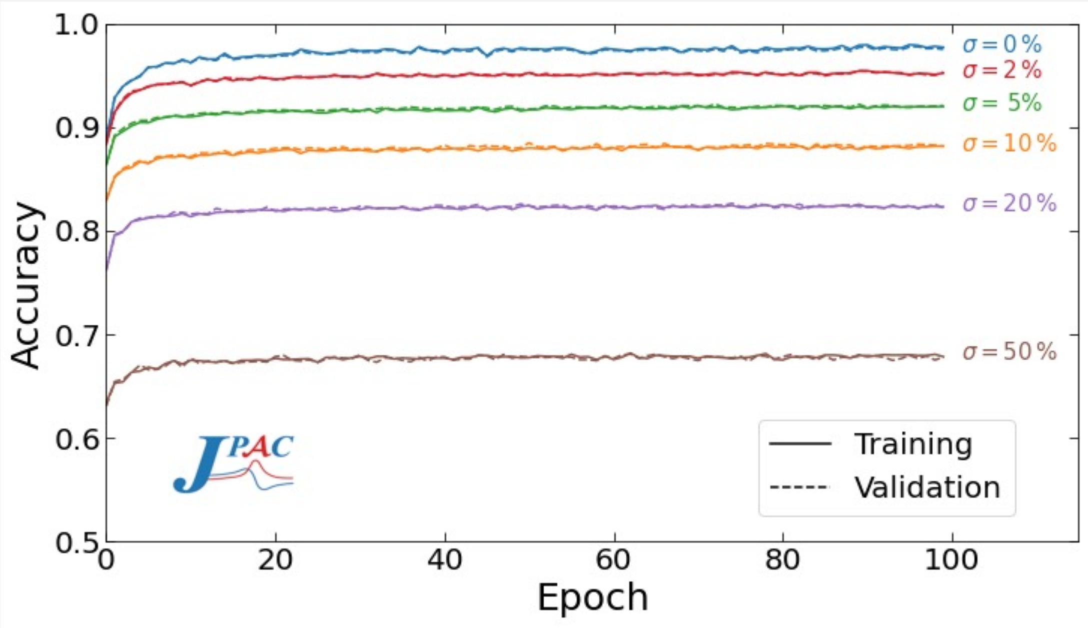
Gaussian noise

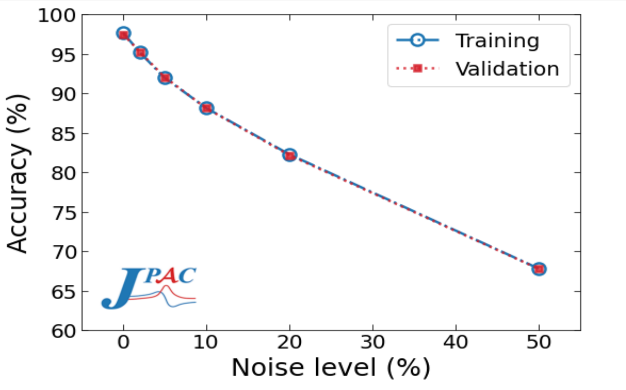
- *Included to mimic uncertainties*
- *Compare "blurry" to "blurry"*

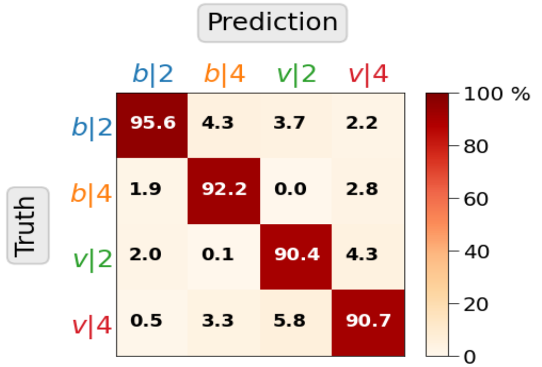


Training / validation









Training determines the parameters (weights) of the NN

Experimental uncertainties: bootstrap

Associate a distribution to each experimental datapoint

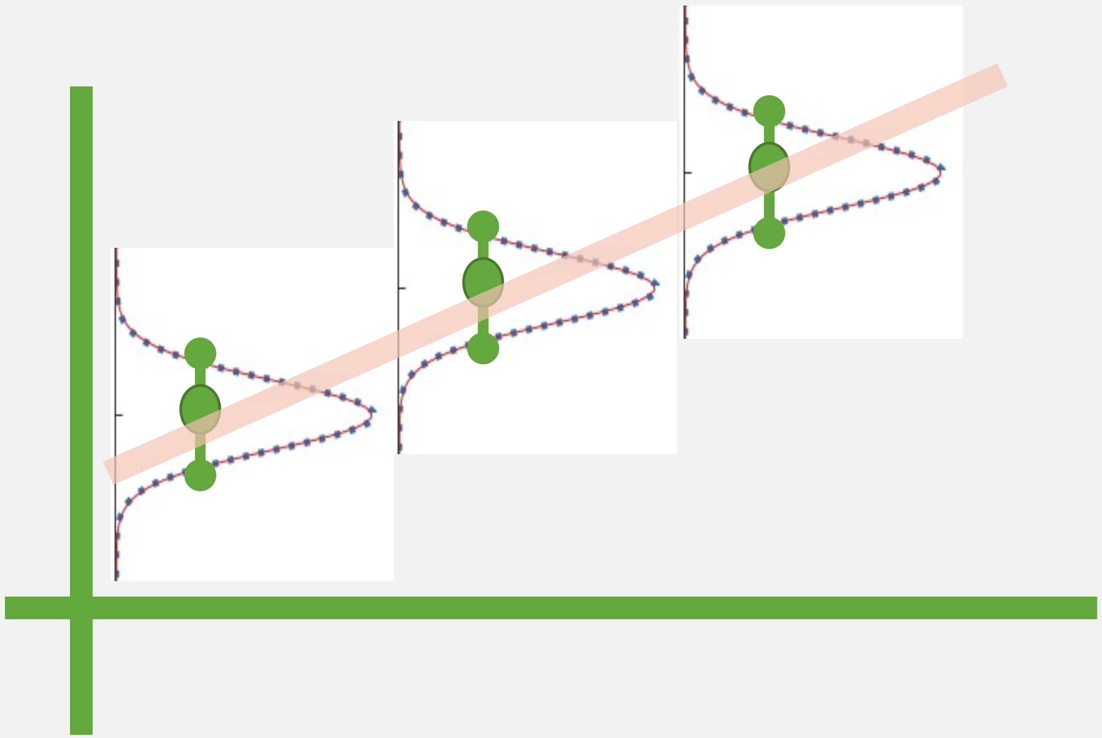
- *Typically a Gaussian with mean and sigma from experiment*

Monte Carlo

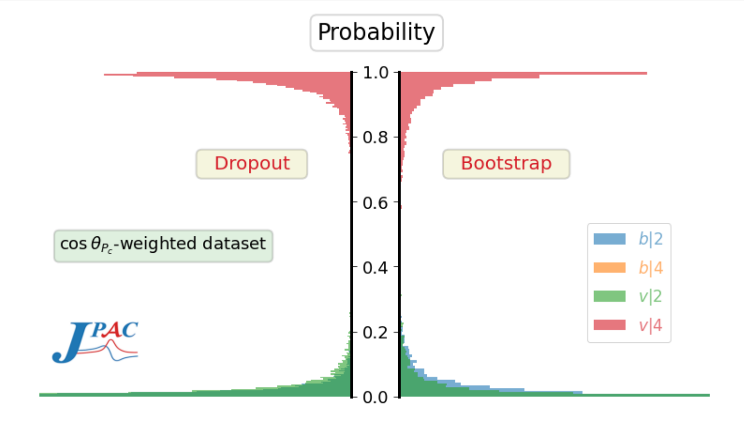
- *Generate pseudodata according to the chosen distribution*

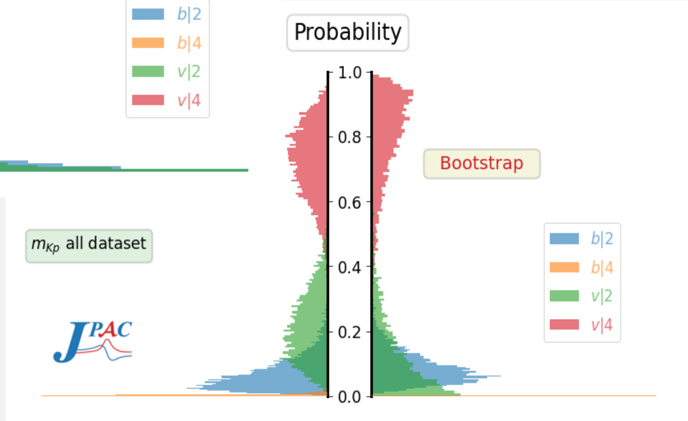
Run statistics on the pseudodatasets

- *Compute distributions, mean, standard deviation, quantiles, …*



Applying the NN to LHCb data





	$b\|2$	$b\|4$	$v\|2$	$v\|4$
$\cos\theta_{P_c}$-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%

We pass the data through the NN

- *We pass the three LHCb datasets through the same NN to obtain three answers*

Uncertainties

- *Bootstrap and dropout*

Obtain proability distributions

- *Sanity check: We recover the same result as with the standard approach: v|4*

What we get from the NN

The NN is comprehensive exploring the parameter space

Standard χ^2fit are often unstable and NN can help exploring the parameter space

Rather than testing a single model hypothesis, the NN tests various models all at once

But, there's more…

Explainability

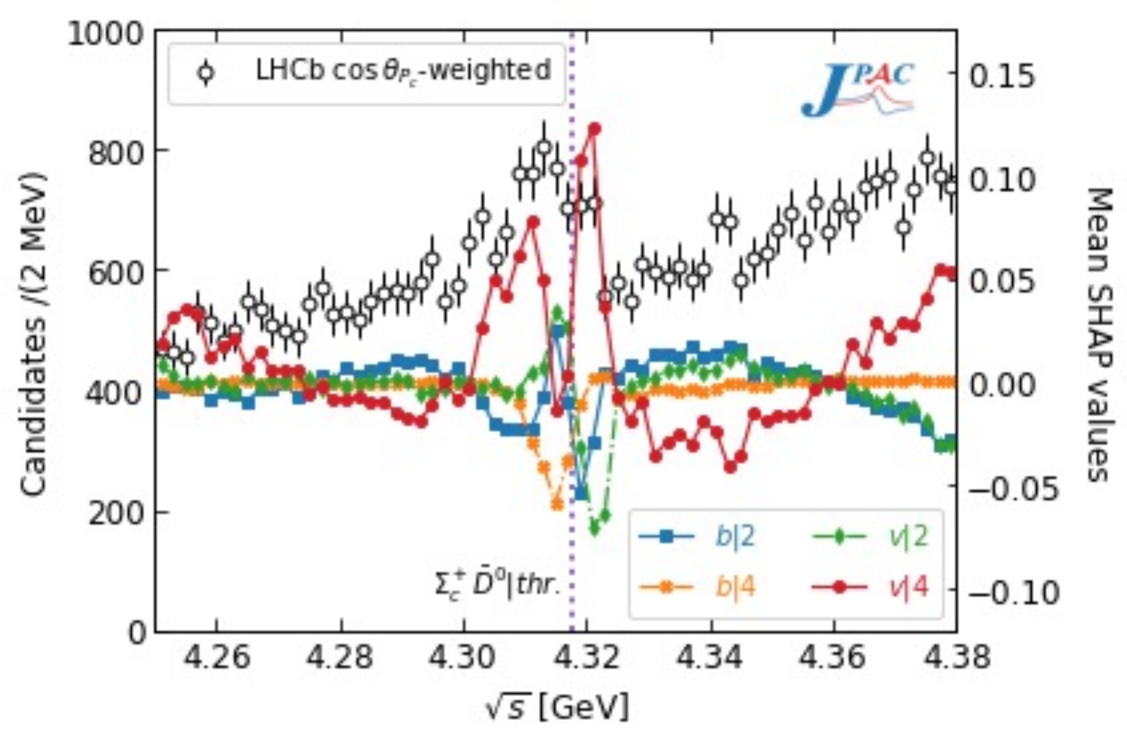
SHAP values

- *SHapley Additive exPlanations*

Inherited from game theory

Application

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



Takeaways

We tested a relatively simple, ML based application, and we are positively surprised by the results

- *Ng et al. (JPAC), 2110.13742*

We started as skeptics and became excited about AI/ML

- *Engaging topic for prospect students*

NN is not a substitution of the canonical approach to analyzing data

We are (hopefully) just in the begining…