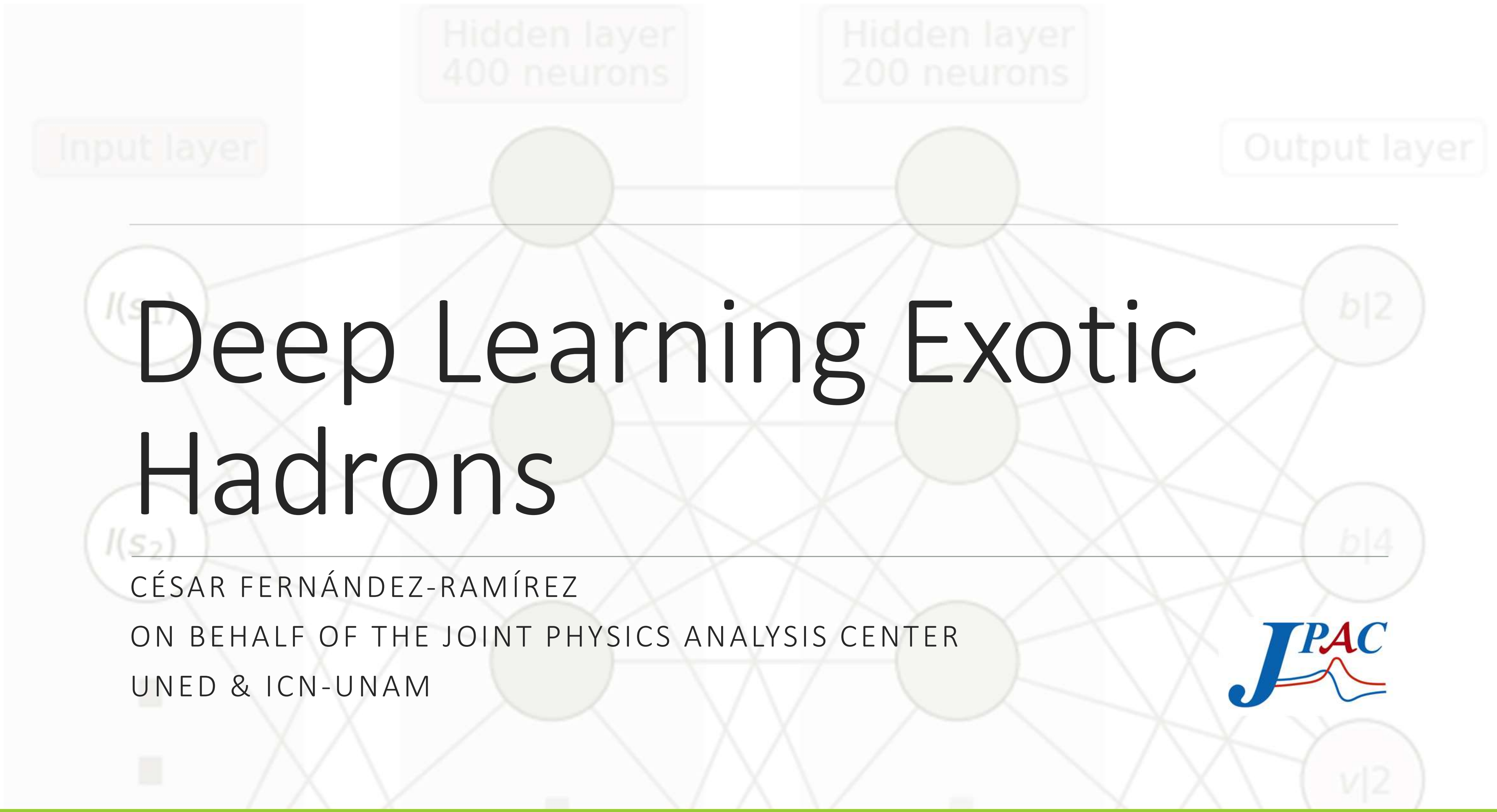


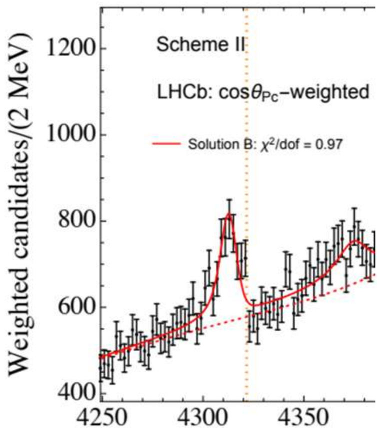
Deep Learning Exotic Hadrons

CÉSAR FERNÁNDEZ-RAMÍREZ
ON BEHALF OF THE JOINT PHYSICS ANALYSIS CENTER
UNED & ICN-UNAM

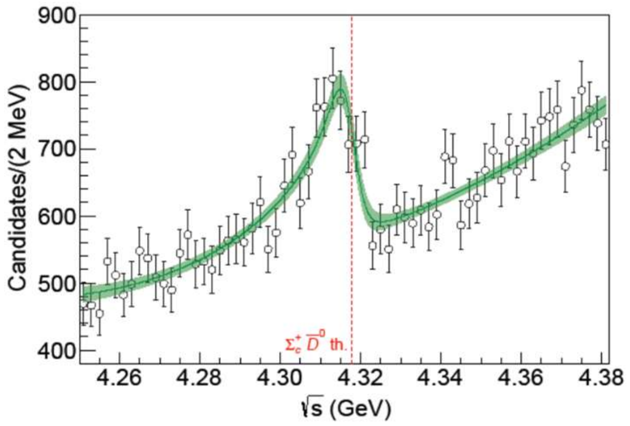
Standard approach to lineshape analysis

- Assume your model <u>is true</u>
- Fit data using χ^2
- Extract model parameters and get pole positions and compute uncertainties
- Asses the probability that those data were generated by your model
- If everything is fine, you can claim that the interpretation embedded in the model is a possible explanation of the data
- You can do this with different models with different underlying dynamics
- Compare models?

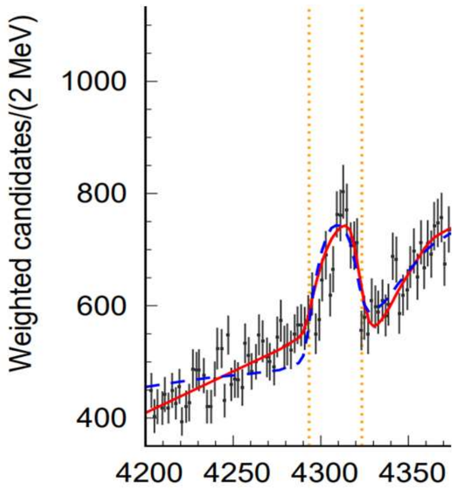
Example: $P_c(4312)$



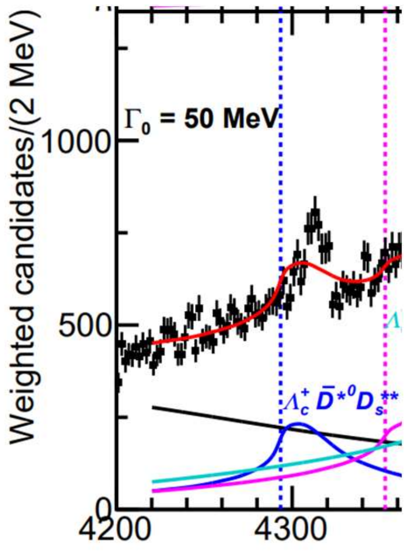
Molecule
Du et al., 2102.07159



Virtual
CFR et al. (JPAC), 1904.10021



Double-triangle (w. complex coupl. in the Lagrangian)
Nakamura, 2103.06817



Single triangle (ruled out)
LHCb, 1904.03947

Can machine learning help us?

- First explorations of Deep neural networks as classifiers for hadron spectroscopy:
  Sombillo et al., 2003.10770, 2104.141782, 2105.04898

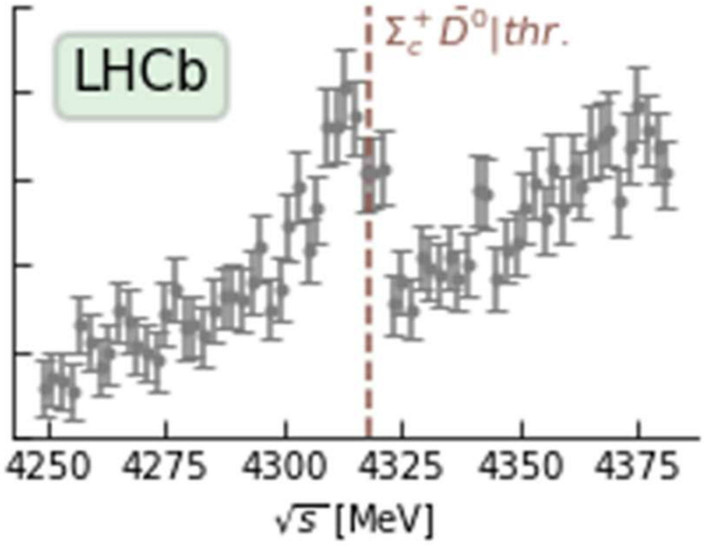
- **Specific questions:**
  - **Can we train a neural network to analyze a lineshape and get as a result what is the probability of each posible dynamical explanation?**
  - **If posible, what other information can we gain by using machine learning techniques?**

- Still far away from answering those question but we are advancing
- Benchmark case applying to lineshape: $P_c(4312)$ Ng et al. (JPAC), 2110.13742

Outlook

- ❑ Standard approach to $P_c(4312)$ (benchmark case):
  CFR et al. (JPAC), 1904.10021
- ❑ Deep neural networks: classifiers
- ❑ Application to our benchmark case
- ❑ Additional information: SHAP values

J/Ψp projection data

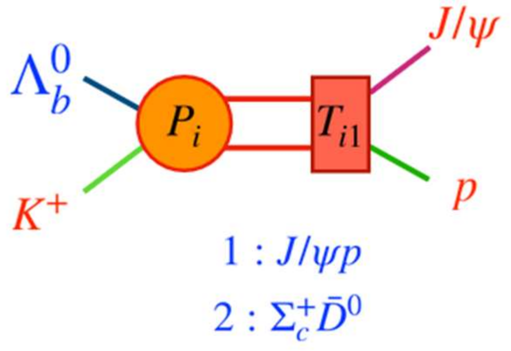


LHCb, 1904.03947

- We focus on $\Sigma_c^+\bar{D}^0$threshold region where the $P_c(4312)$ signal appears
- We asume only one partial wave contributes
- The threshold is responsible for the dynamics
- Other singularities are irrelevant

Near-threshold theory

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

$F(s) =$ 

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

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

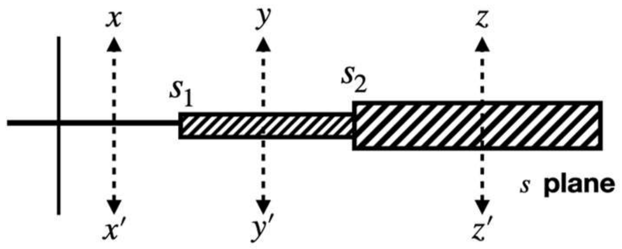
Matrix elements M_{ij} are singularity free and can be Taylor expanded

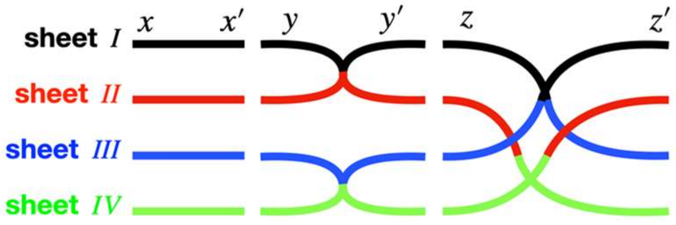
Frazer, Hendry, PR134 (1964) B1307

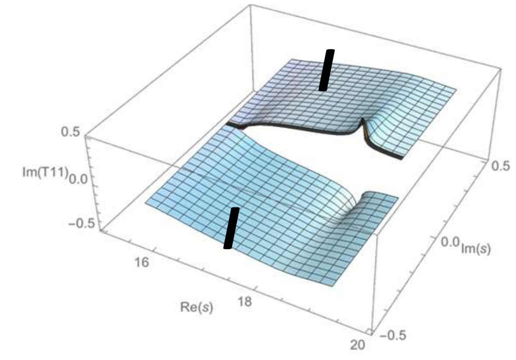
$c_{ij} = 0$ is the scattering length approximation

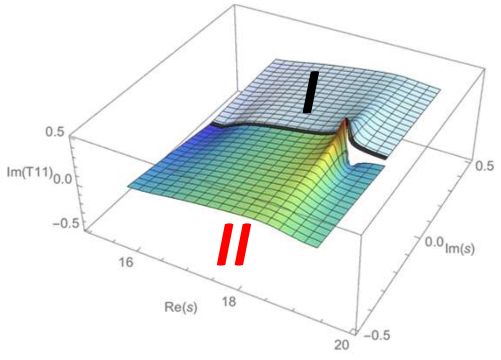
Riemann sheets structure

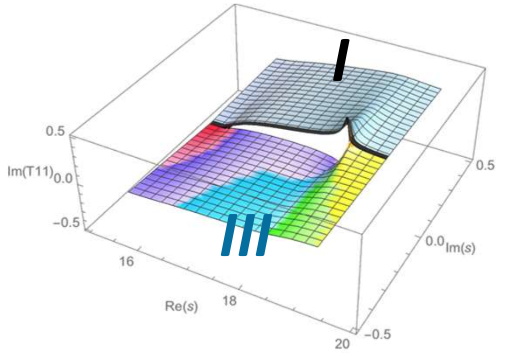
$$\begin{bmatrix} J/\psi p \to J/\psi p & J/\psi p \to \Sigma_c \bar{D}^0 \\ \Sigma_c \bar{D}^0 \to J/\psi p & \Sigma_c \bar{D}^0 \to \Sigma_c \bar{D}^0 \end{bmatrix}$$

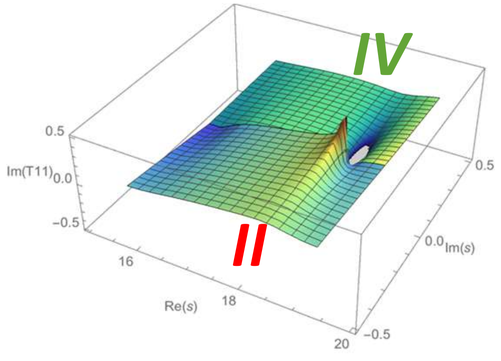






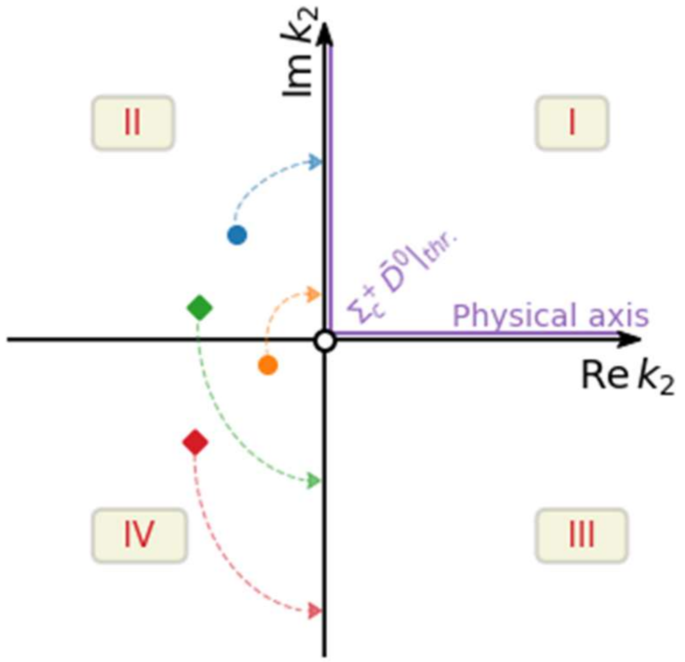






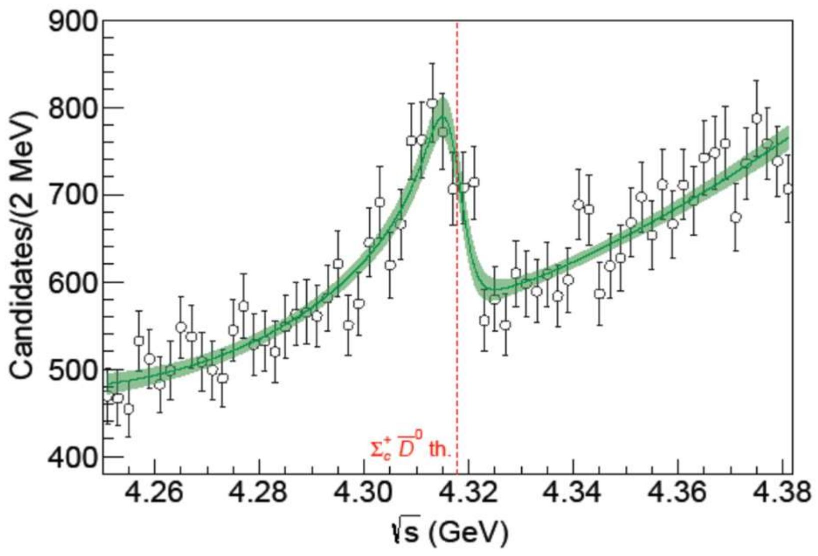
Virtual and bound states

- Under the the scattering length approximation the physical interpretation is given by the sign of the m_{22} parameter. Four options:
- Bound state on IV RS: b|4
- Virtual state on IV RS: v|4
- Bound state on II RS: b|2
- Bound state on II RS: v|2

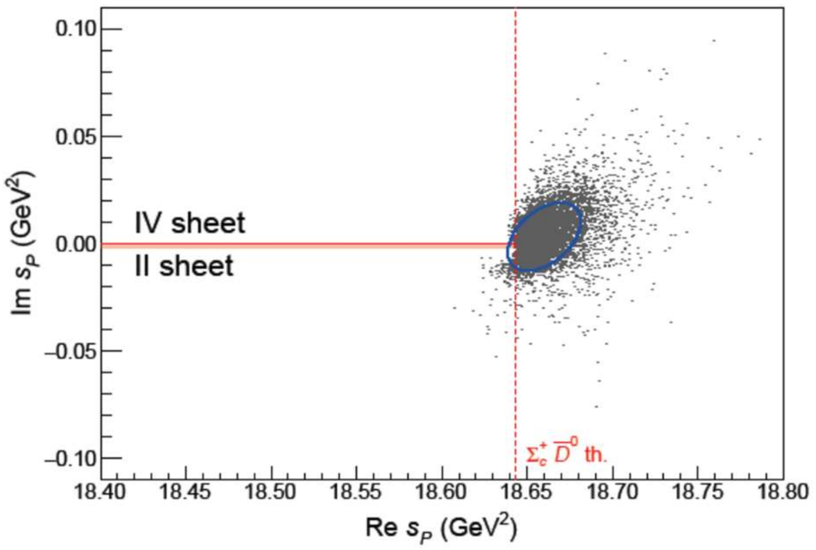


Standard approach is still needed

Amplitude



Pole

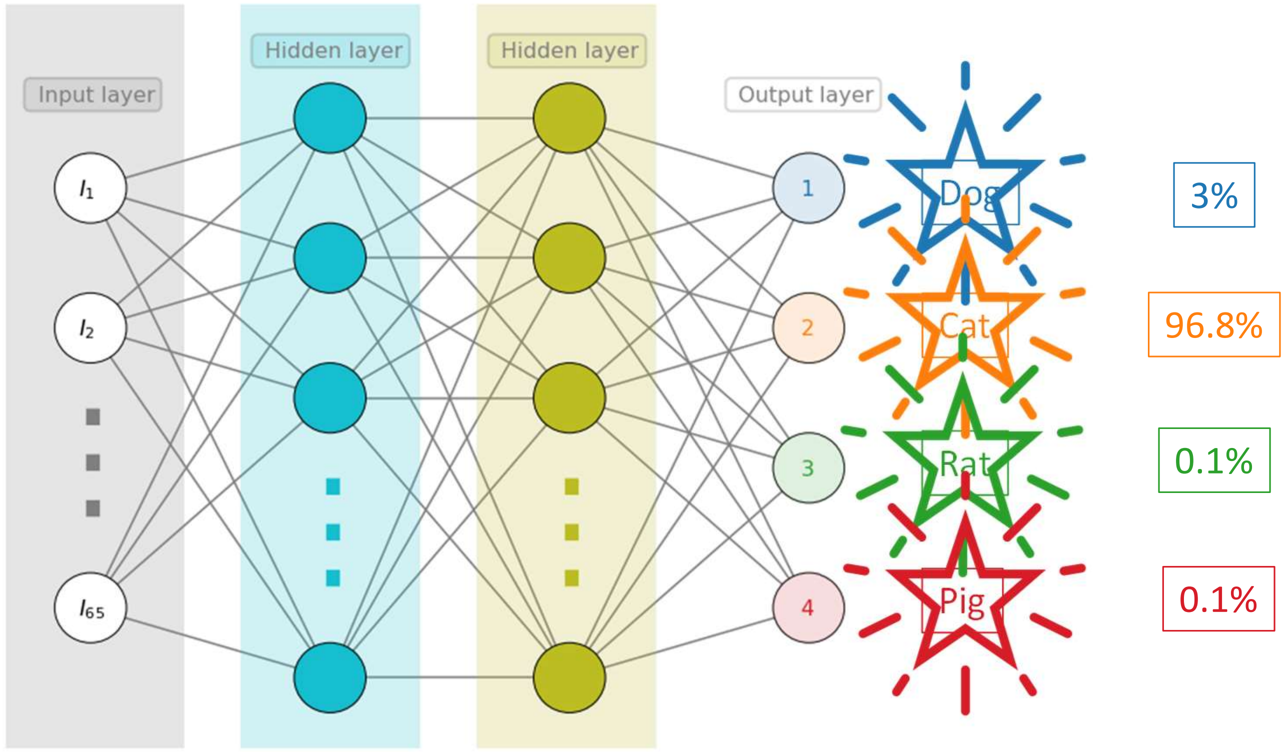


- Interpretation obtained: Virtual state on IV RS (v|4)

M= 4319.7±1.6 MeV	Γ= —0.8±2.4 MeV

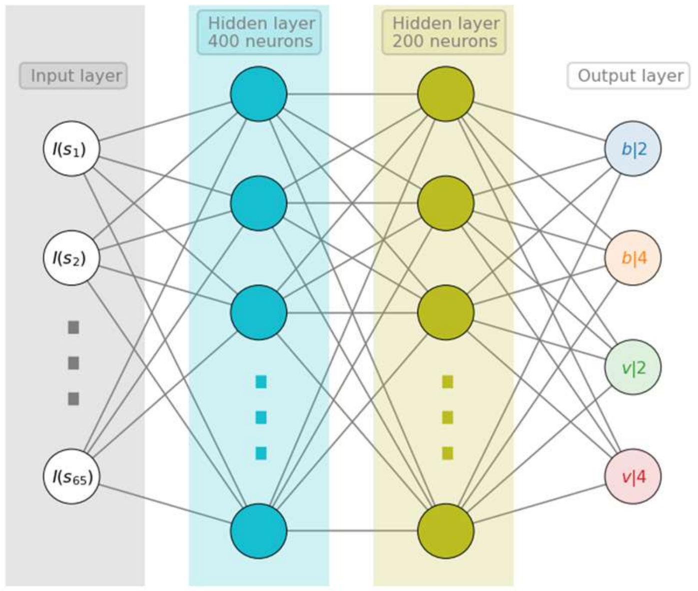
Neural networks as classifiers

New picture



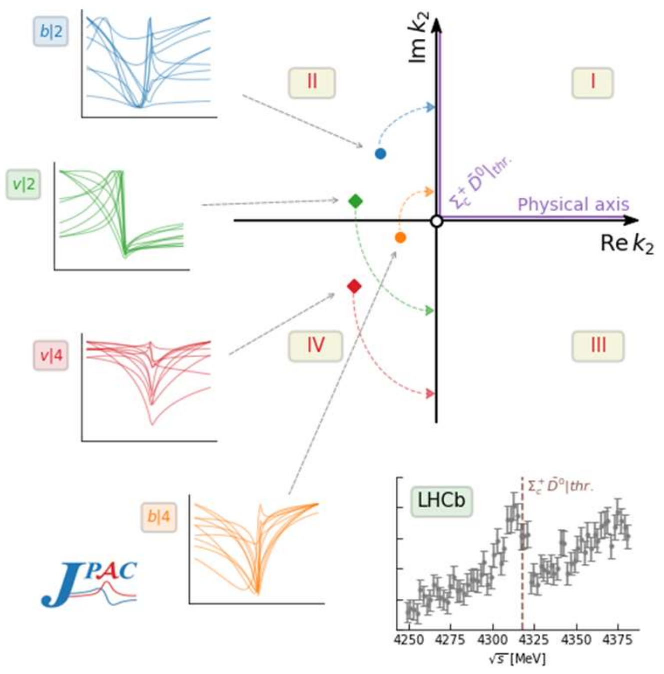
The DNN

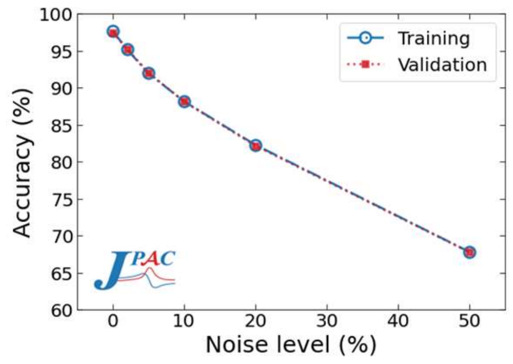
- The input layer contains 65 experimental datapoints
- Two hidden layers with 400 and 200 neurons respectively
- Output layer, four classes: v|4, b|4, b|2, v|2
- Probabilities are obtained using the softmax function $p(x_i) = \frac{e^{x_i}}{\sum e^{x_j}}$
- Implemented with PyTorch

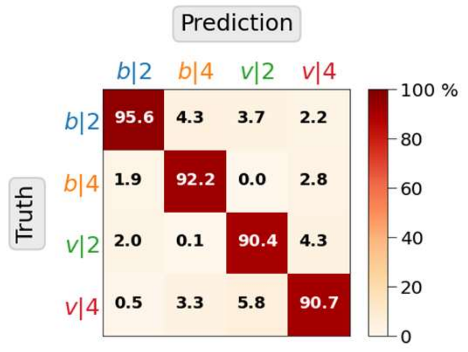


Build a training set and train your DNN

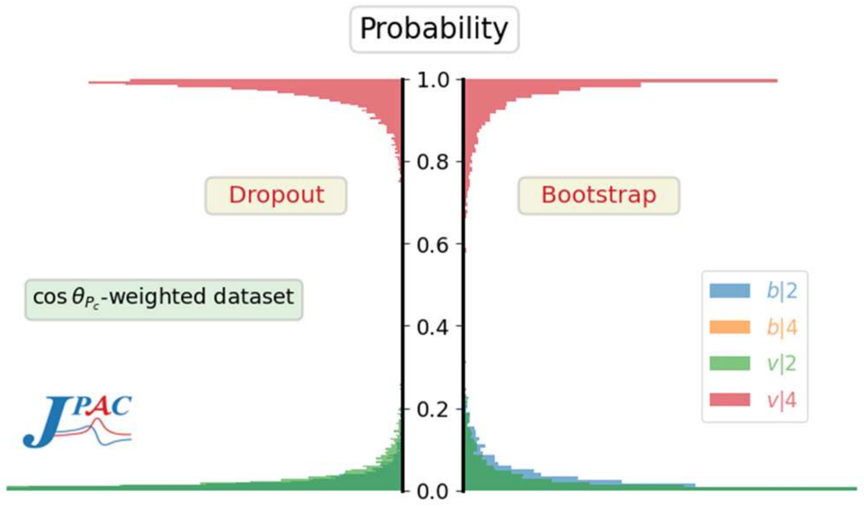
- ❑ 10^5 training curves generated by randomly choosing parameter values
- ❑ To mimic uncertainties a a 5% noise is included in the training set
- ❑ The lineshapes are convoluted with the experimental resolution
- ❑ Experimental uncertainty limits the accuracy of the DNN







	$b\|2$	$b\|4$	$v\|2$	$v\|4$
$b\|2$	95.6	4.3	3.7	2.2
$b\|4$	1.9	92.2	0.0	2.8
$v\|2$	2.0	0.1	90.4	4.3
$v\|4$	0.5	3.3	5.8	90.7



Applying the DNN

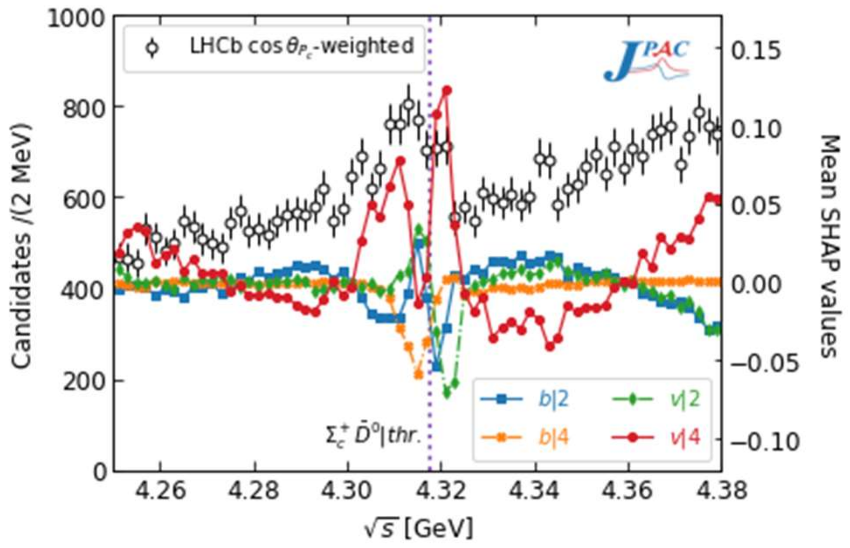
- We pass the three LHCb datasets (full, cut in m_{Kp} and weighted) through the DNN to obtain an answer. Note that we pass the three dataset through the same DNN
- To account for uncertainties we use two procedures
  - Bootstrap
  - Dropout
- We (unsurprisingly) recover the same result as with the standard approach: v|4
- But we are learning new things…

What we get from the DNN

- The DNN targets specific regions of the parameter space (which yield stable solutions) that might be difficult to reach during optimization or might require high-resolution data.
- Standard χ^2fit can be indeed unstable, and a small change in the input data can induce large changes in the parameter values and therefore in the physics interpretation.
- Rather than testing a single model hypothesis as a χ^2 fit would, the DNN determines the probability of each of the classes of interest, given the experimental uncertainties. The latter is possible, since the DNN learns the subtle classification boundary between the different classes.
- But, there’s more…

SHapley Additive exPlanations (SHAP) values

- Technique inherited from game theory
- Allows to determine how a given feature in the input layer (in our case an experimental datapoint) impacts the decisión made by the DNN in the output layer (the classes)



Takeaways

- ❑ Deep neural networks open new possibilities to answer the question on the underlying nature of a given resonance
- ❑ Possibility to gain physics insight on how the data impact the obtained interpretation
- ❑ DNN does NOT substitute the standard approach. They are complementary (we still want the amplitude and the pole position)
- ❑ DNN allows a true comparison among interpretations
- ❑ The BIG objective: Be able to train the DNN with every amplitude possibility we can devise with our twisted minds and throw the data to it, returning probabilities for each class
- ❑ Uncharted territory so we are taking baby steps: Ng et al. (JPAC), 2110.13742
- ❑ We are (hopefully) just in the begining…