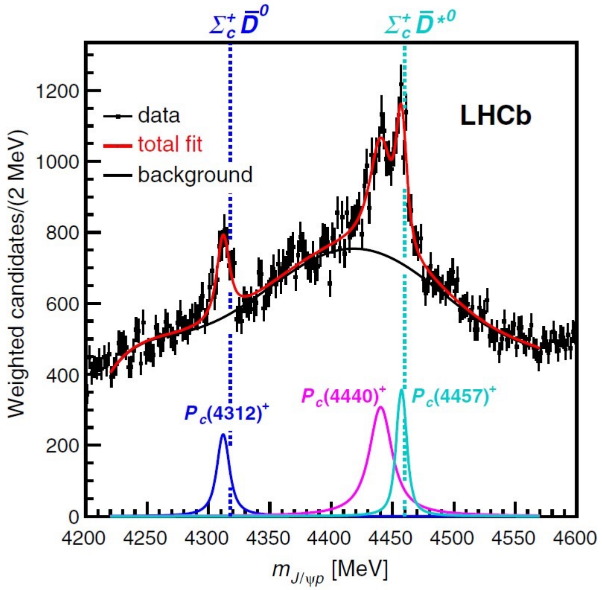
Extraction of S-matrix pole structure using deep learning

PhysRevD.102.016024
Few-Body Syst. 62,52
PhysRevD.104.036001

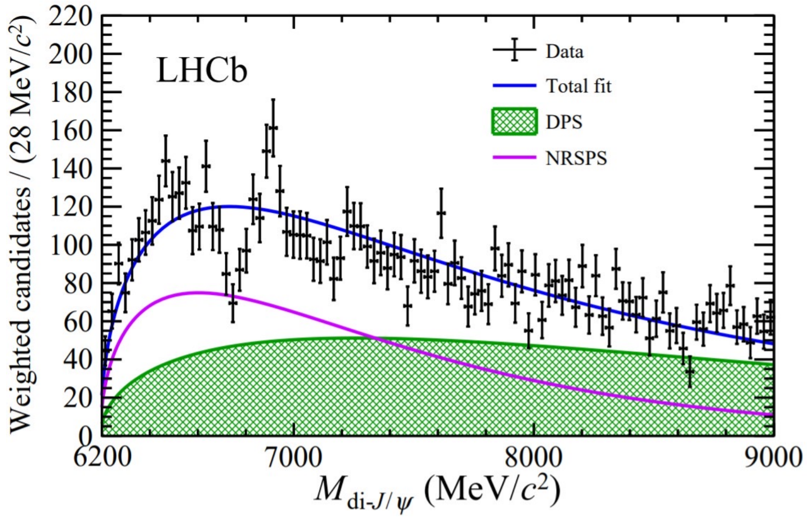
Denny Sombillo

In collaboration with:
Yoichi Ikeda (Kyushu University)
Toru Sato (RCNP, Osaka University)
Atsushi Hosaka (RCNP, Osaka University and ASRC, JAEA)

Motivation and Overview



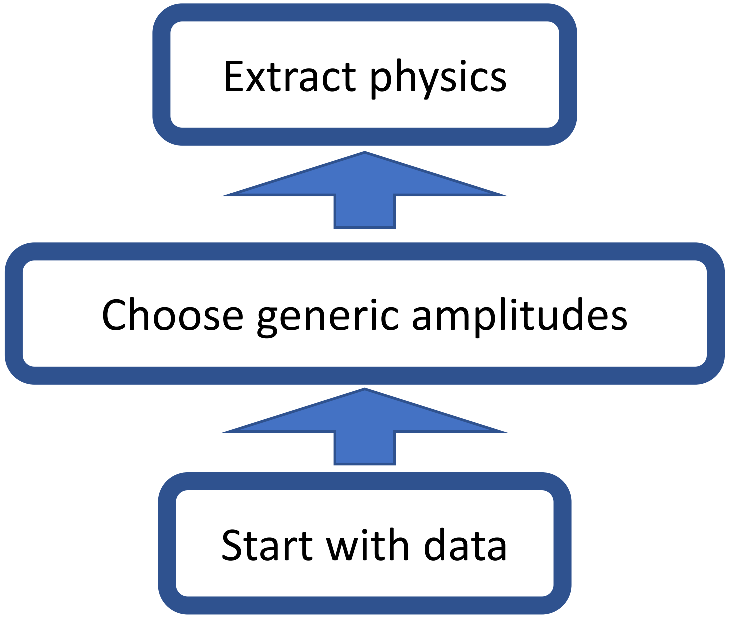
PhysRevLett.122.222001(2019)



j.scib.2020.08.032(2020)

F.-K.Guo's talk
PANIC2021 Sept 7th

Bottom-top approach:



A.Pilloni's talk
PANIC2021 Sept 5th

Explanation of the observed near-threshold/ threshold structures

- Molecular bound state
- Virtual state
- Threshold cusp
- Compact state

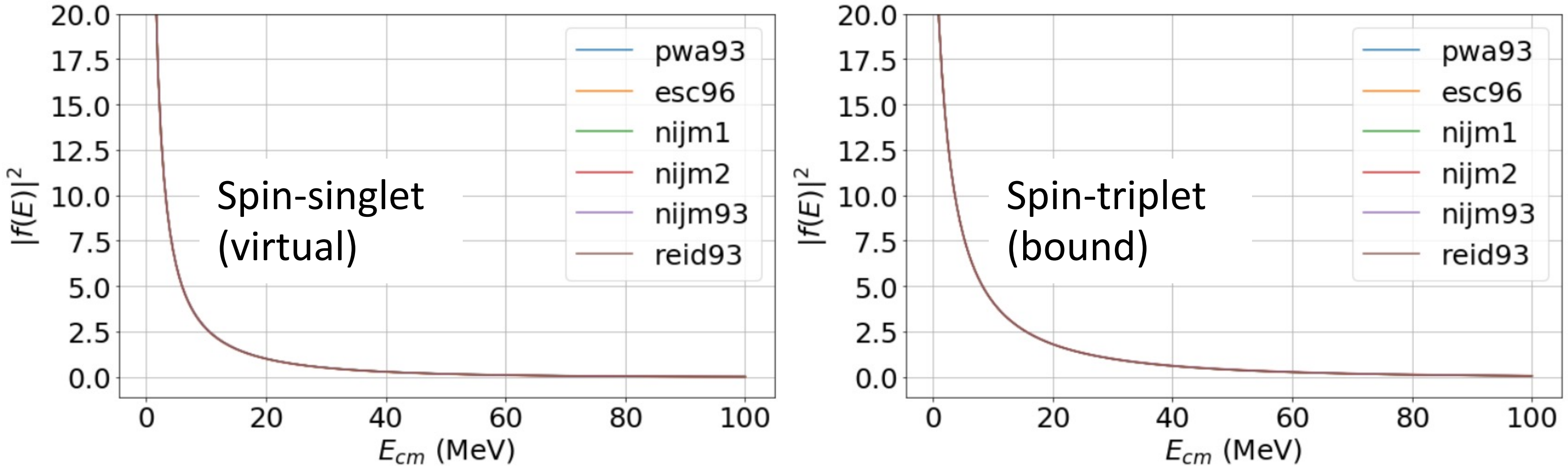
Study of compositeness is important
Kinugawa-san's talk PANIC2021 Sept 8th

Realizable in deep learning framework (with some modification)

Proof-of-principle

Can a trained deep neural network distinguish virtual and bound enhancements of the low-energy nucleon-nucleon scattering without invoking the existence of deuteron?

DLBS, YI, TS, AH PhysRevD.102.016024(2020)
DLBS, YI, TS, AH. Few-Body Syst 62,52(2021)

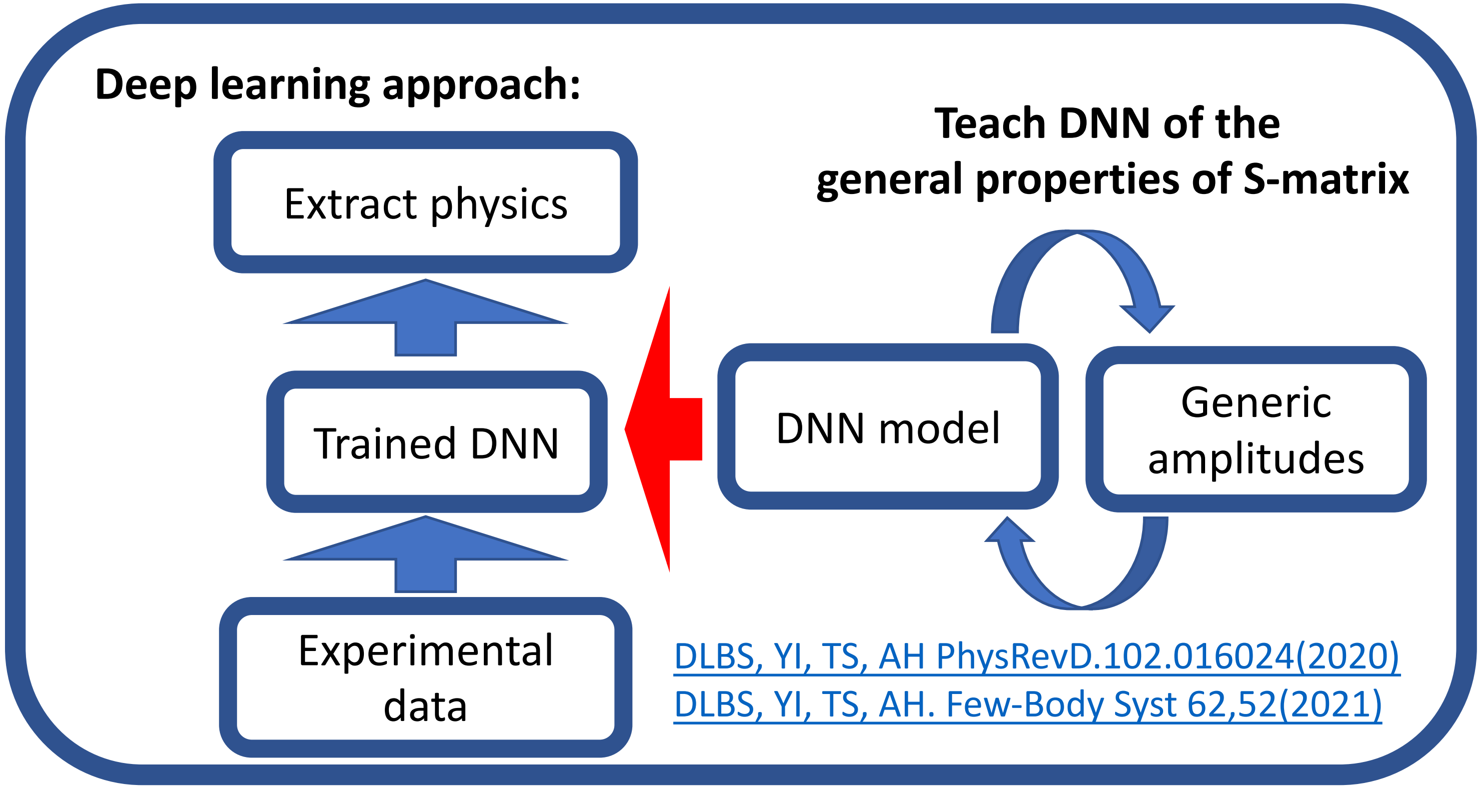


http://nn-online.org/NN/?page=nnphs2

The unbalanced distribution of singularities in the complex momentum plane should manifests on the scattering region above the threshold.

We just don't know how. → Let deep neural network figure it out for us.

Proof-of-principle



Trained Deep Neural Network's inference output:

	PWA93	ECS96	NijmI	NijmII	Nijm93	Reid93
$^1S_0$	virtual	virtual	virtual	virtual	virtual	virtual
$^3S_1$	bound	bound	bound	bound	bound	bound

Proof of principle

- Generic properties of S-matrix is sufficient to generate the training dataset.
- Threshold enhancements can be studied using deep learning approach.

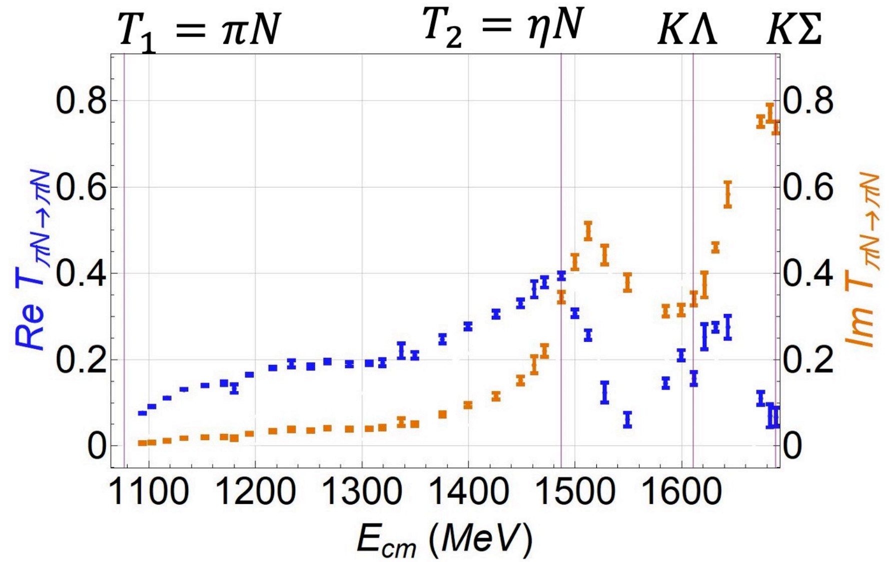
Deep learning as a model-selection framework

Pole-based models: How many poles in each Riemann sheet are needed to reproduce the experimental data?

Model-space restriction: maximum of 4 poles

For coupled two-channel scattering: 35 models.

Label	S-matrix pole configuration
0	no nearby pole
1	1 pole in [bt]
2	2 poles in [bt]
⋮	⋮ ⋮ ⋮ ⋮
32	1 pole in [bt], 2 poles in [bb] and 1 pole in [tb]
33	1 pole in [bt], 1 pole in [bb] and 2 poles in [tb]
34	1 pole in [bt], 1 pole in [bb] and 1 pole in [tb]

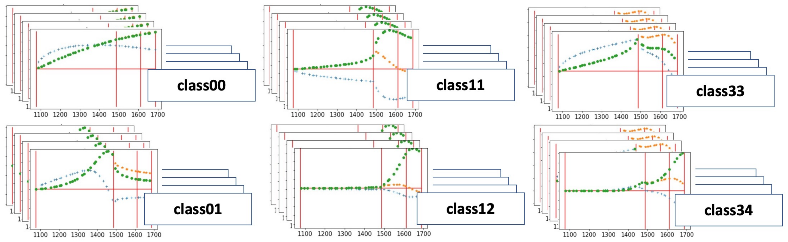


Note that because of the error bars, we expect that the experimental data can be described by **more than one model**.

Deep learning as a model-selection framework

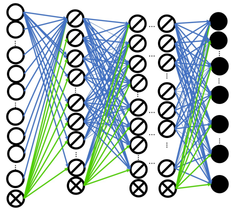
Proposed deep learning analysis approach:

Generating the training dataset (Model space)



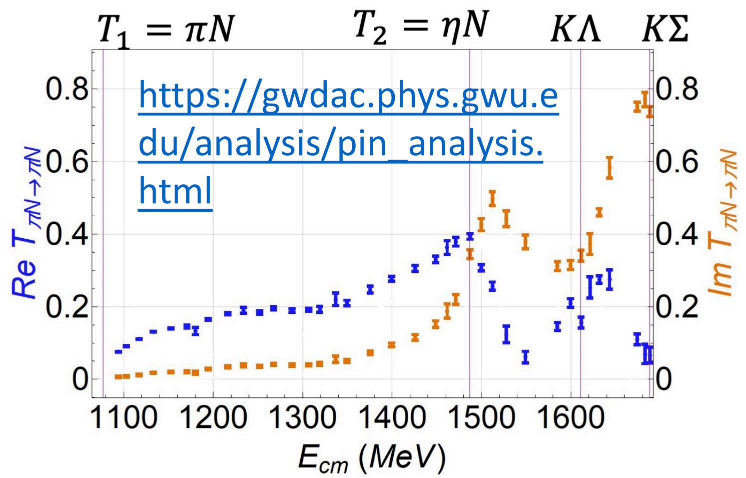
- Use general properties of S-matrix
- Include energy uncertainty

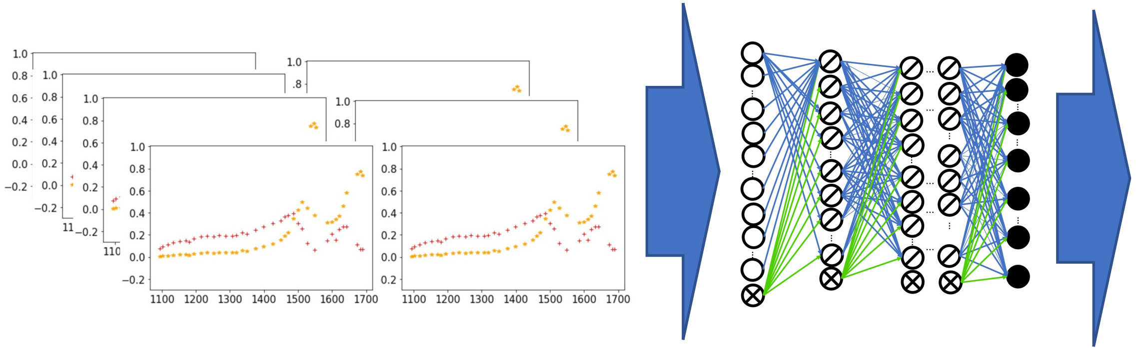
Optimization of deep neural network (DNN) model



Apply trained-DNN on experimental data

- Use error bars in the data to generate inference amplitudes





Count the number of outcomes

Generation of training dataset (model space)

General form of S-matrix:

- Hermitian below the lowest threshold
- Unitarity
- Analyticity

https://doi.org/10.1063/1.1703698
https://doi.org/10.1098/rspa.1960.0096

$$S_{11}(p_1,p_2) = \prod_m \frac{D_m(-p_1,p_2)}{D_m(p_1,p_2)}; \qquad S_{11} = 1 + 2iT_{11}$$

$$S_{22}(p_1,p_2) = \prod_m \frac{D_m(p_1,-p_2)}{D_m(p_1,p_2)} \qquad S_{22}S_{11} - S_{12}^2 = \prod_m \frac{D_m(-p_1,-p_2)}{D_m(p_1,p_2)}$$

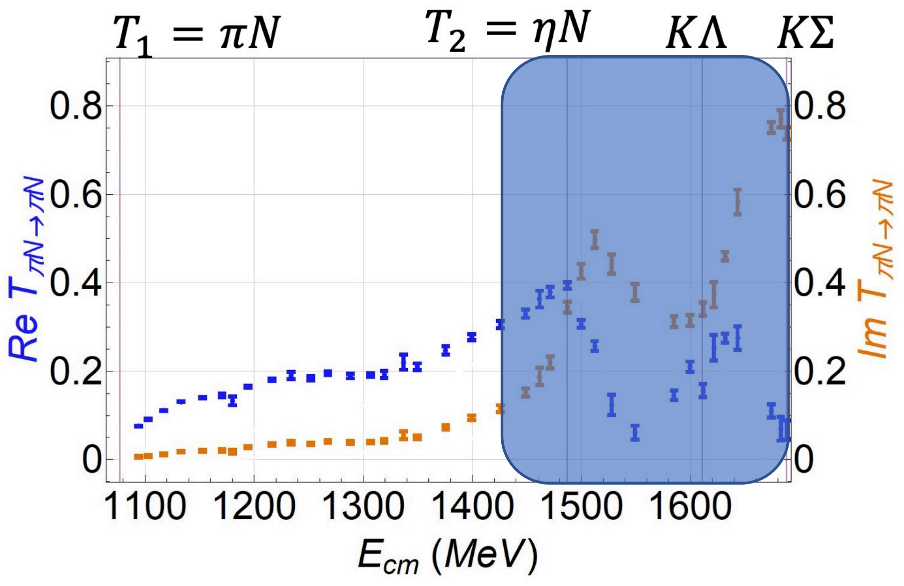
The available experimental data will determine the relevant S-matrix element.

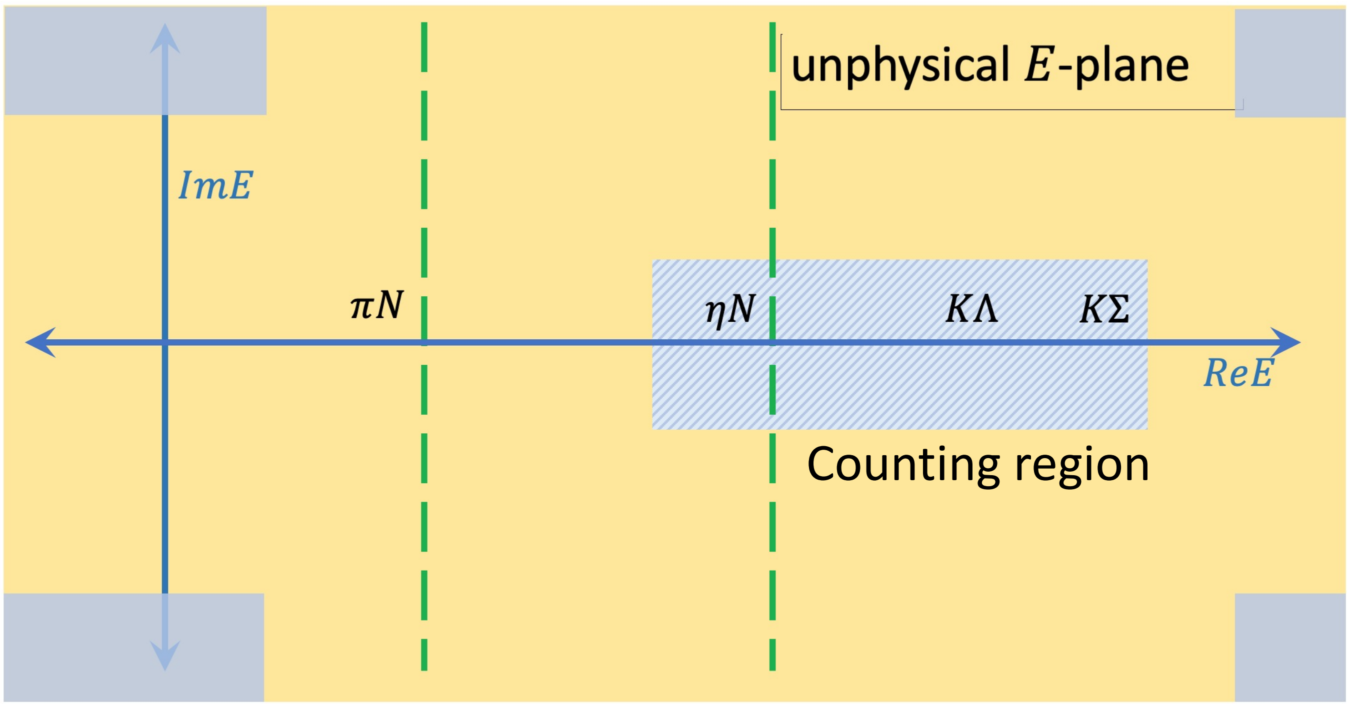
Ensure that one $D_m(p_1,p_2)$ will produce one pole (conjugate pair) on the desired Riemann sheet.

DLBS,YI,TS,AH,PhysRevD.104.036001(2021)

Generation of training dataset (model space)

We limit the region of analysis.





Region where poles are generated and counted:

$$\eta N - 50 \leq Re\ E_{pole} \leq \eta N + 200$$
$$-200 \leq Im\ E_{pole} \leq 200$$

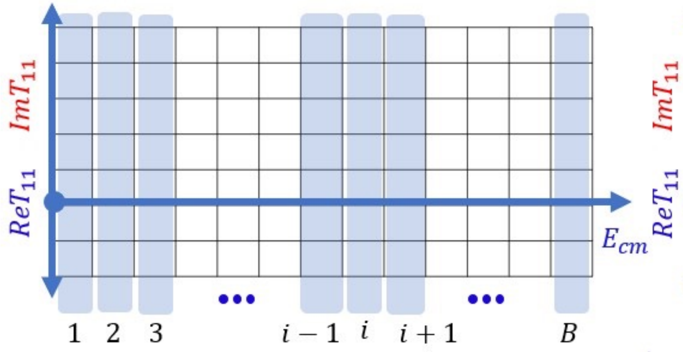
Region where poles are generated but not counted:

$$Re\ E_{pole} < \pi N - 100 \quad and \quad Re\ E_{pole} > \eta N + 500$$
$$700 < |Im\ E_{pole}| < 2000$$

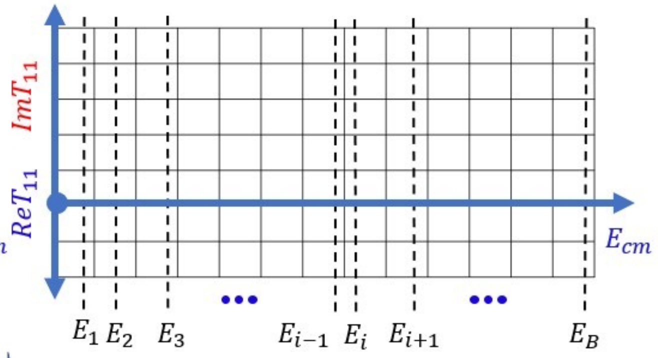
$$S_{11}(p_1, p_2) = \prod_j \frac{D_j(-p_1, p_2)}{D_j(p_1, p_2)} \prod_n \frac{D_n(-p_1, p_2)}{D_n(p_1, p_2)}$$

Generation of training dataset (model space)

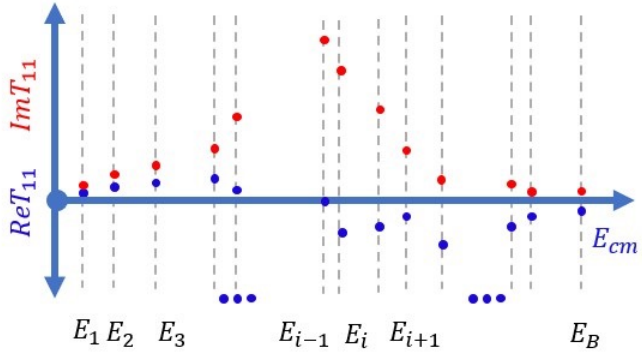
To simulate the limited energy resolution.



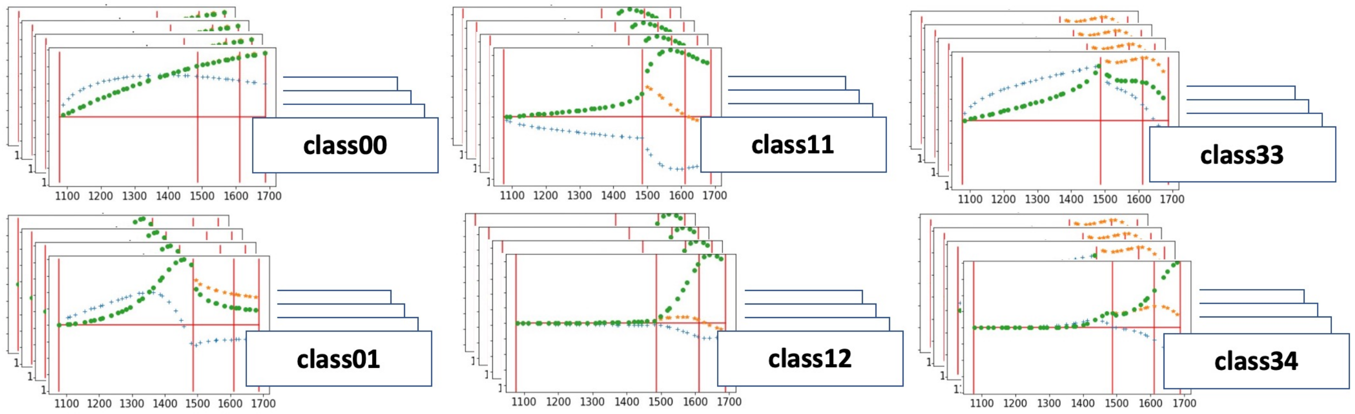
(1) Divide the energy range into *B* bins



(2) Choose a random energy value in each bin



(3) Calculate the amplitude for each random energy value



(4) Label each amplitude according to its pole-configuration

Label	S-matrix pole configuration
0	no nearby pole
1	1 pole in $[bt]$
2	2 poles in $[bt]$
⋮	⋮ ⋮ ⋮ ⋮
32	1 pole in $[bt]$, 2 poles in $[bb]$ and 1 pole in $[tb]$
33	1 pole in $[bt]$, 1 pole in $[bb]$ and 2 poles in $[tb]$
34	1 pole in $[bt]$, 1 pole in $[bb]$ and 1 pole in $[tb]$

Optimization of DNN model

- Start with small easy examples.
  Elman 1993
- Slowly introduce new class (pole models) until all examples are presented.
- Perform regular training loop

Chosen DNN architecture

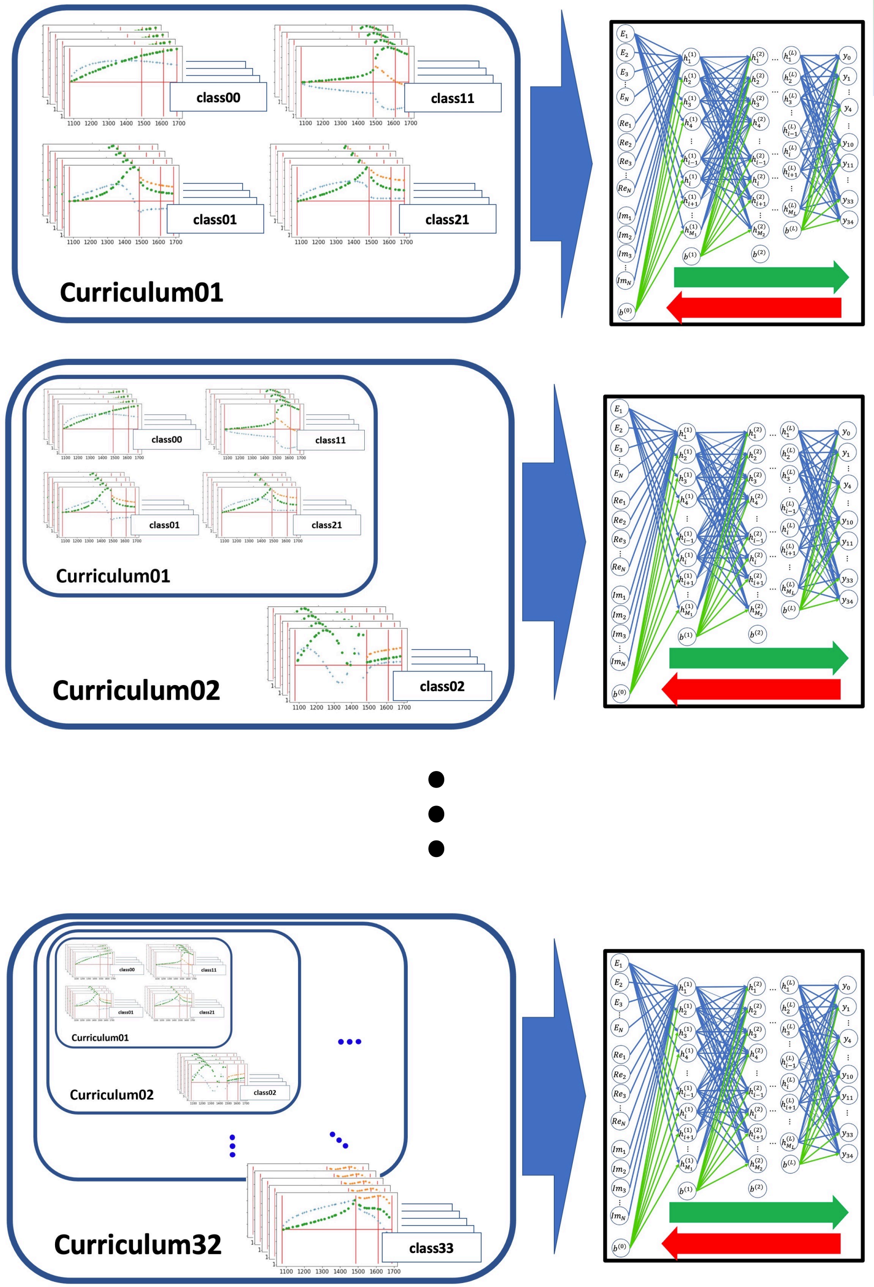
Layer	Number of nodes	Activation Function
Input	111+1	
1st	200+1	ReLU
2nd	200+1	ReLU
3rd	200+1	ReLU
Output	35	Softmax

DLBS,YI,TS,AH,PhysRevD.104.036001(2021)

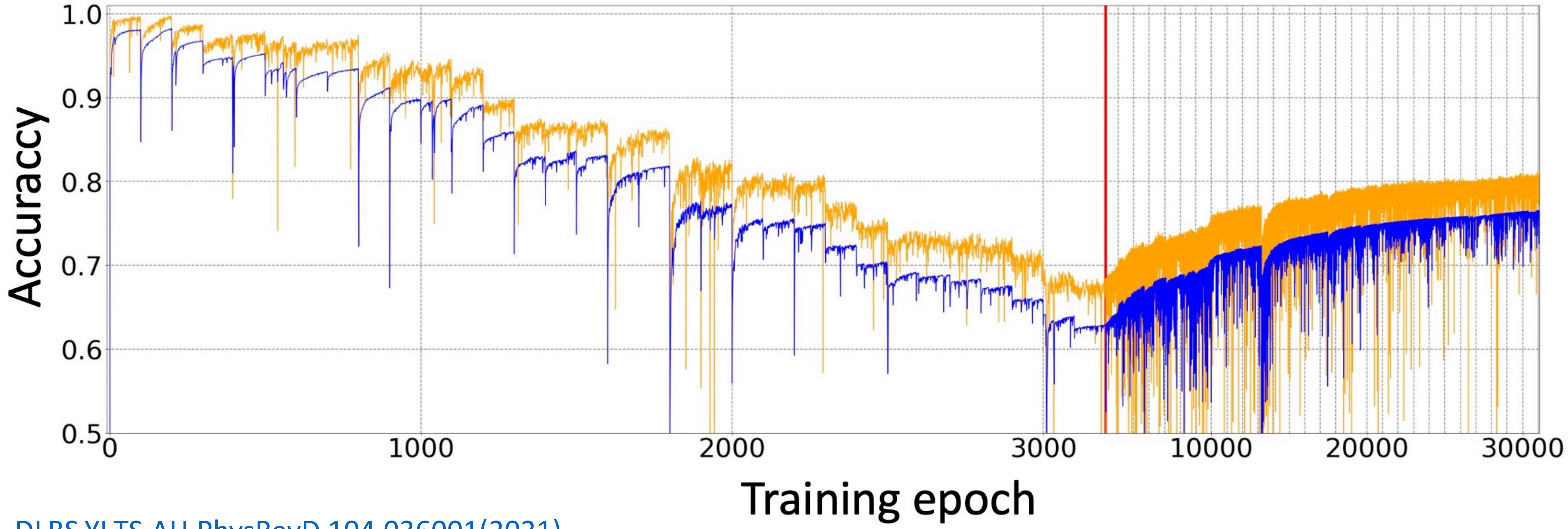
Curriculum01:
At-most-**one**-pole models

Curriculum02:
Curriculum01 + 1 new class in the **two**-pole models

Curriculum32:
Curriculum31 + last class in the **four**-pole models



Optimization of DNN model

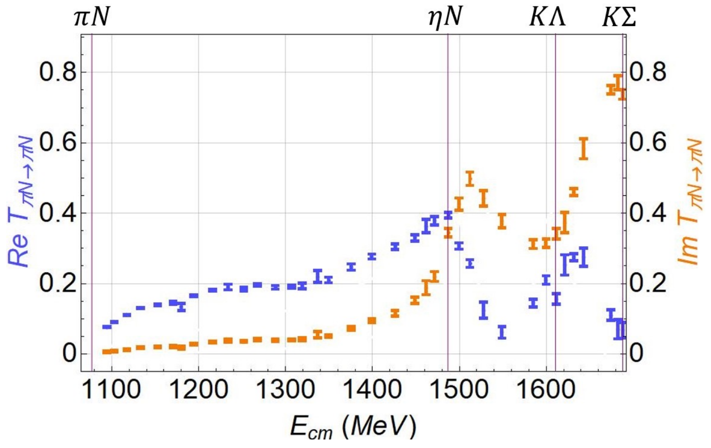


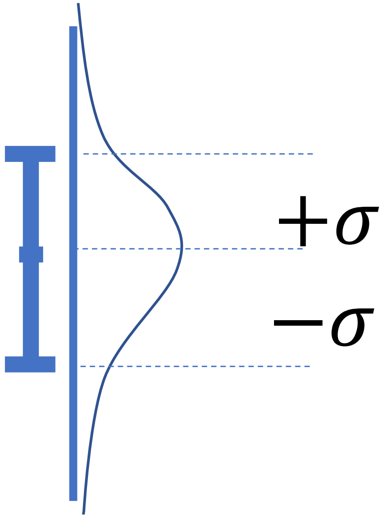
DLBS,YI,TS,AH,PhysRevD.104.036001(2021)

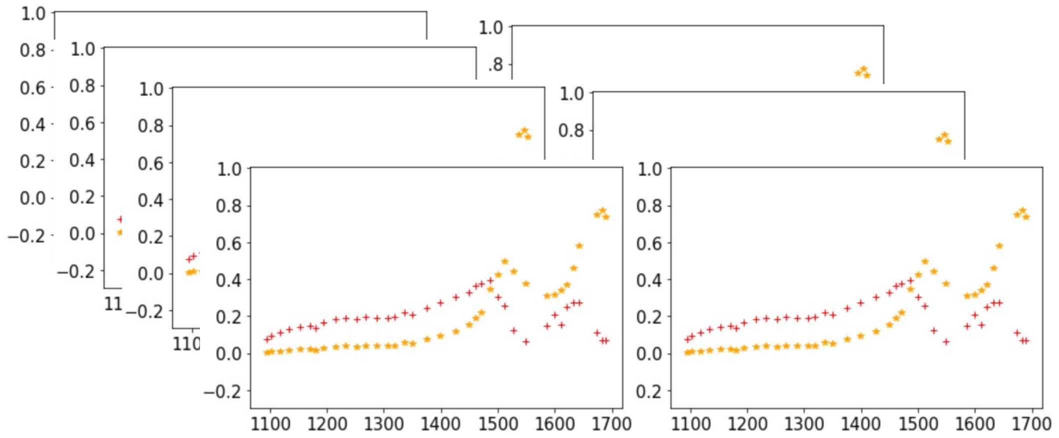
Post-curriculum training accuracy: 76.5%
Post-curriculum testing accuracy: 80.4%

We now have a DNN that can identify the appropriate pole-based model for a given data.

Inference stage: Application







- Pick random points in each error bar
- No model-fitting step is needed
- Generate 10^6 amplitudes
- Feed to the trained DNN
- Count the number of outcomes

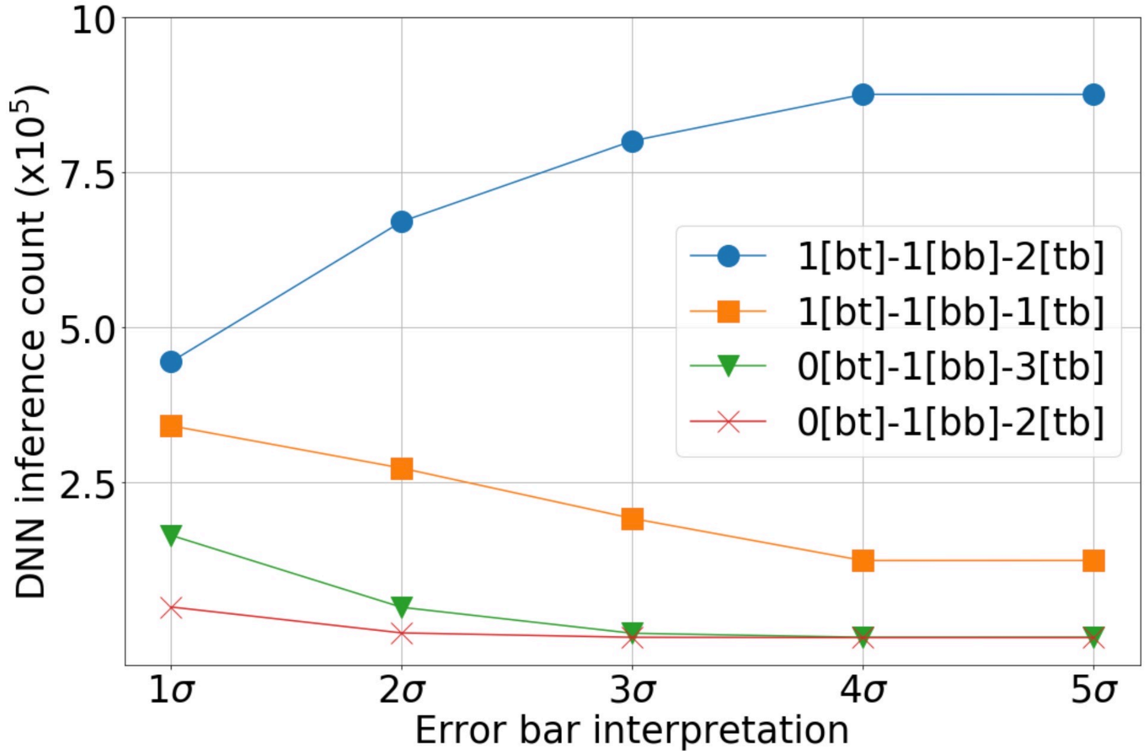
DNN inference on 10^6 amplitudes using 1 error bar = 1σ

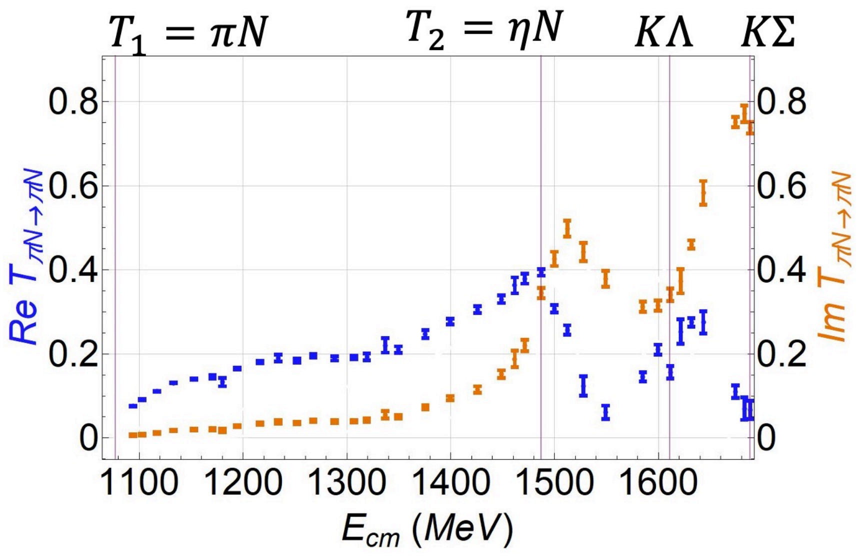
- 44.6% 1[bt]-1[bb]-2[tb]
- 34.1% 1[bt]-1[bb]-1[tb]
- 16.4% 0[bt]-1[bb]-3[tb]
- 4.9% 0[bt]-1[bb]-2[tb]

DLBS,YI,TS,AH,PhysRevD.104.036001(2021)

Inference stage: Application

DNN inference on 10^6 amplitudes using Gaussian distribution for each error bar



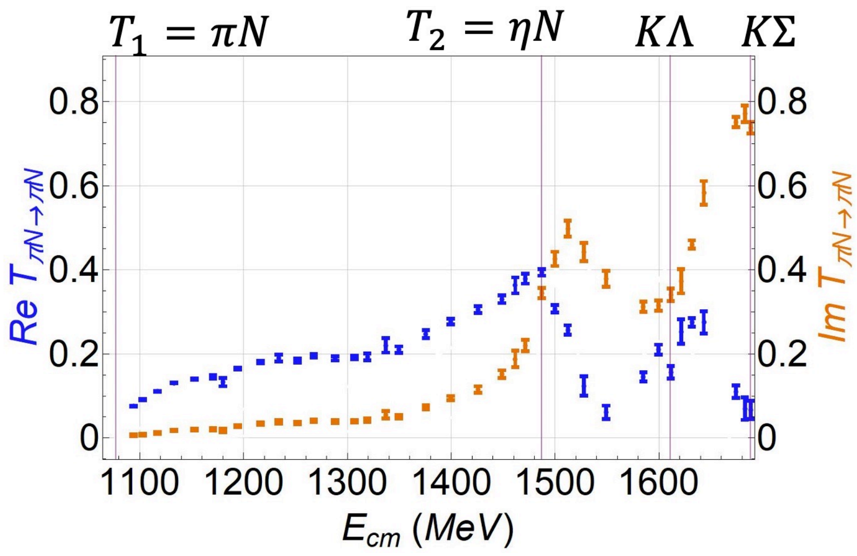


DNN inference on 10^6 amplitudes using uniform distribution for each error bar

- 60.3% 1[bt]-1[bb]-2[tb]
- 30.9% 1[bt]-1[bb]-1[tb]
- 7.5% 0[bt]-1[bb]-3[tb]
- 1.3% 0[bt]-1[bb]-2[tb]

DLBS,YI,TS,AH,PhysRevD.104.036001(2021)

Inference stage: Application



DNN inference on 10^6 amplitudes using uniform distribution for each error bar

- 60.3% 1[bt]-1[bb]-2[tb]
- 30.9% 1[bt]-1[bb]-1[tb]
- 7.5% 0[bt]-1[bb]-3[tb]
- 1.3% 0[bt]-1[bb]-2[tb]

- Detected [bb] (3rd RS) pole
  - Enhancement between $K\Lambda$ and $K\Sigma$ thresholds.
- Detected [bt] (2nd RS) pole
  - Far from ηN threshold but within the counting region
  - Might be shadow of the detected [bb]
- Detected [tb] (4th RS) poles
  - Only poles left to explain ηN enhancement

Conclusion

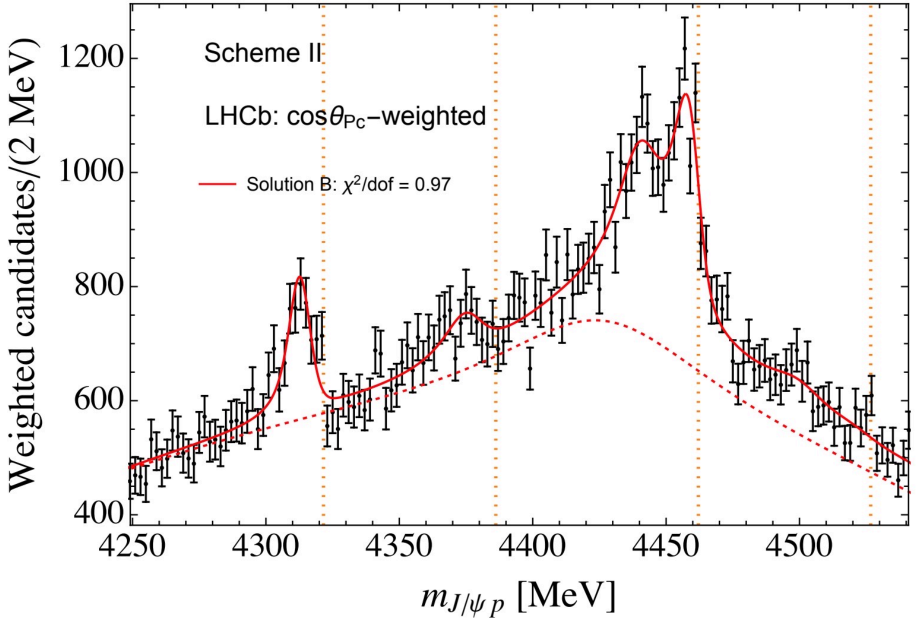
Summary:

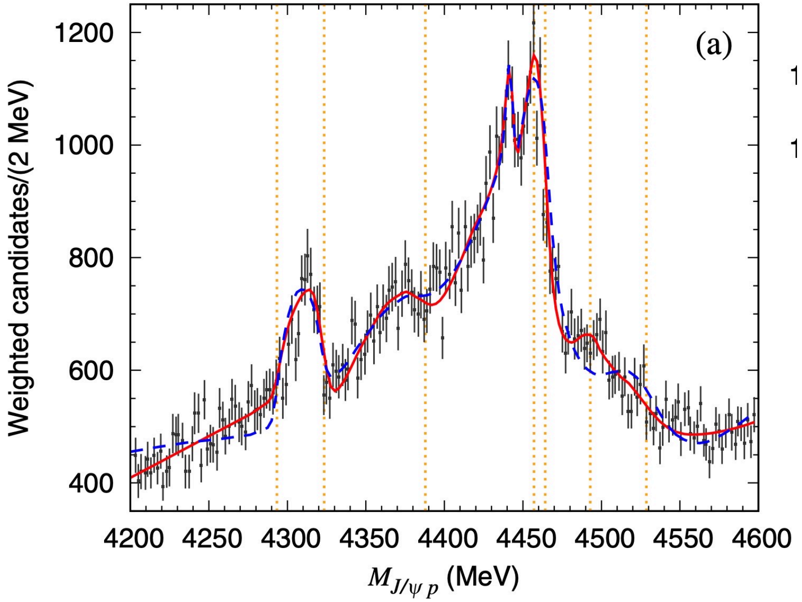
- We can teach a deep neural network to recognize the pole structure of an amplitude.
- Deep learning framework allows us to have a unified treatment to execute the model selection process.

Outlook:

- Extract pole position and the Riemann sheet using deep learning approach.
    - Training dataset can contain different backgrounds.
    - Uncertainties in the pole position can be obtained using the experimental data's error bars.

Further future plan





PhysRevLett.124.072001 Molecular picture of Pc-states

PhysRevD.103.L111503 Double triangle singularities

Same data, almost the same quality of fit, two conflicting models.
Which one is a better description of data?

Maybe deep learning framework
can give us an unbiased answer.

Stay tuned!

Thank you for listening!