NEDA PSA through Neural Networks

NEDA/DIAMANT data processing

Xavier FABIAN

IPN Lyon
Université Lyon 1, CNRS, IN2P3

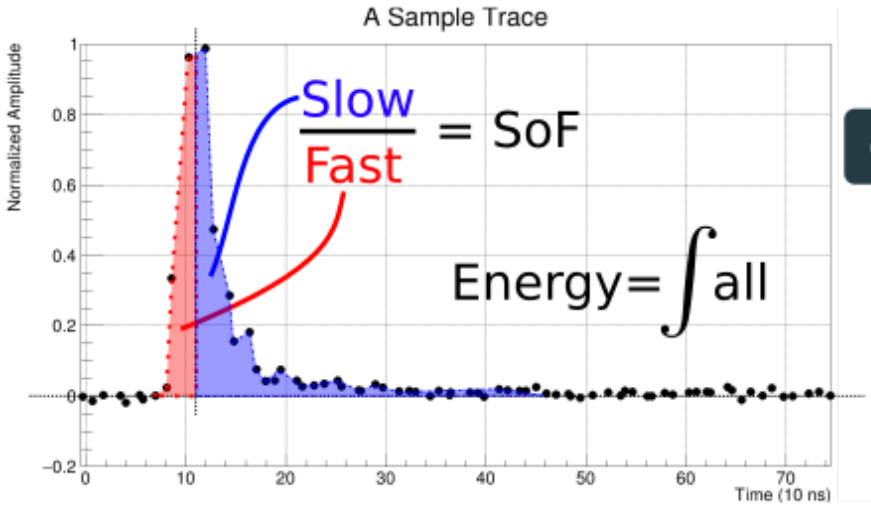
AGATA Data Analysis Workshop
Orsay, January 2019

Université Claude Bernard  Lyon 1

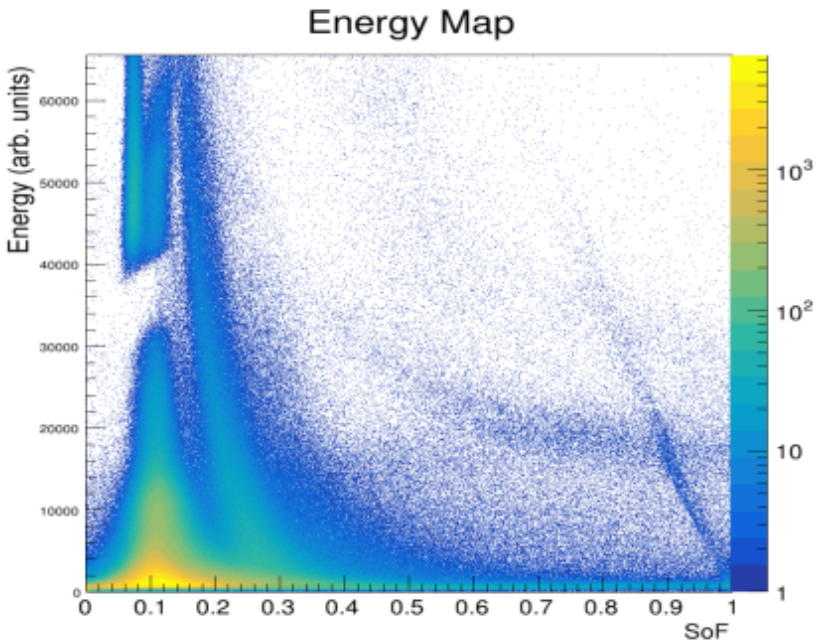
Menu

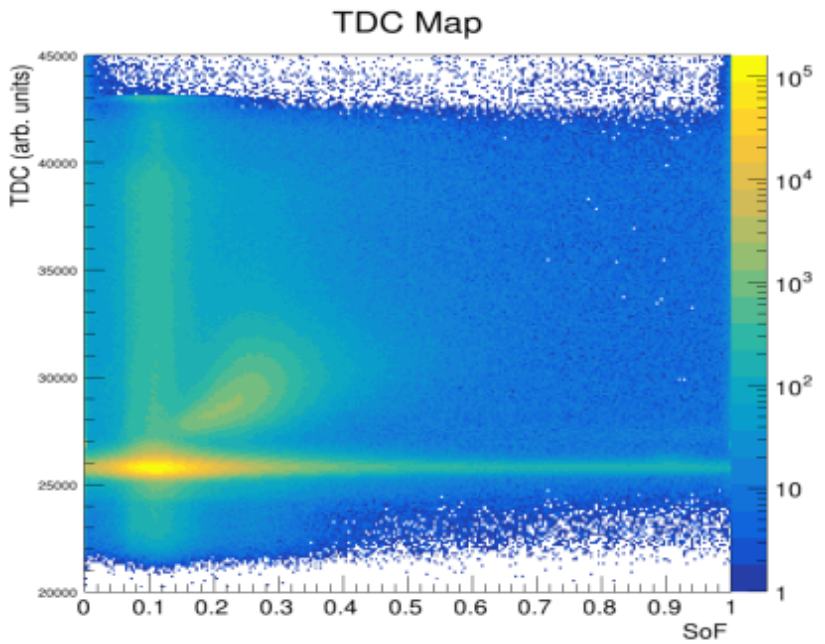
1. Introduction
2. Training
3. Performances
4. A Word on Neutron Wall

NEDA PSA

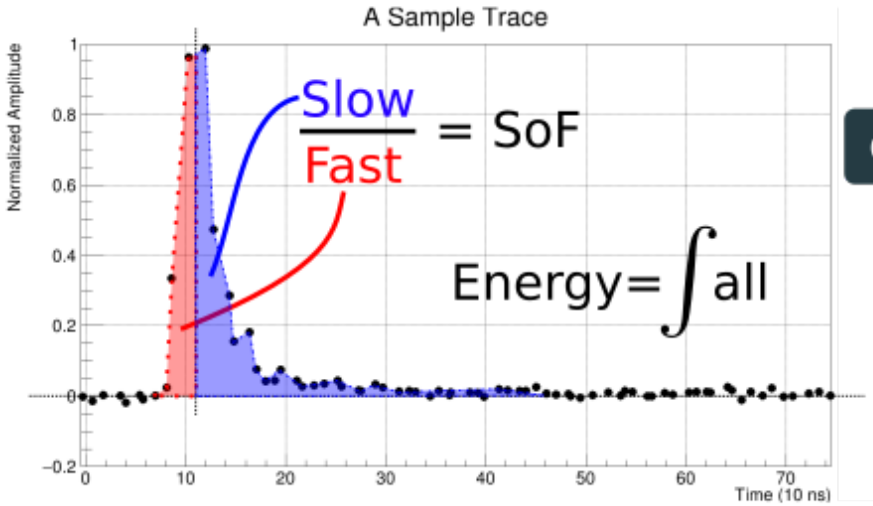


CCPSA actor computes Slow, Fast, Energy



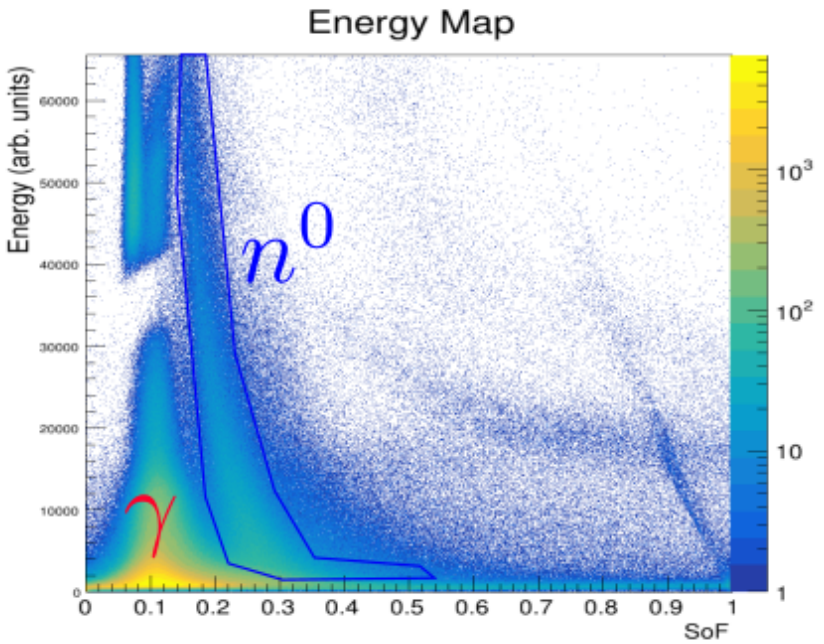


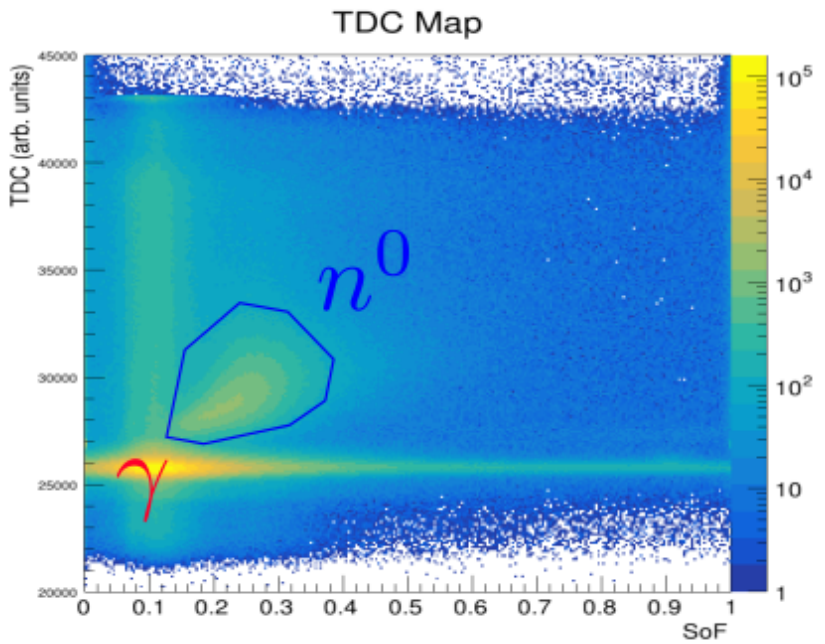
NEDA PSA



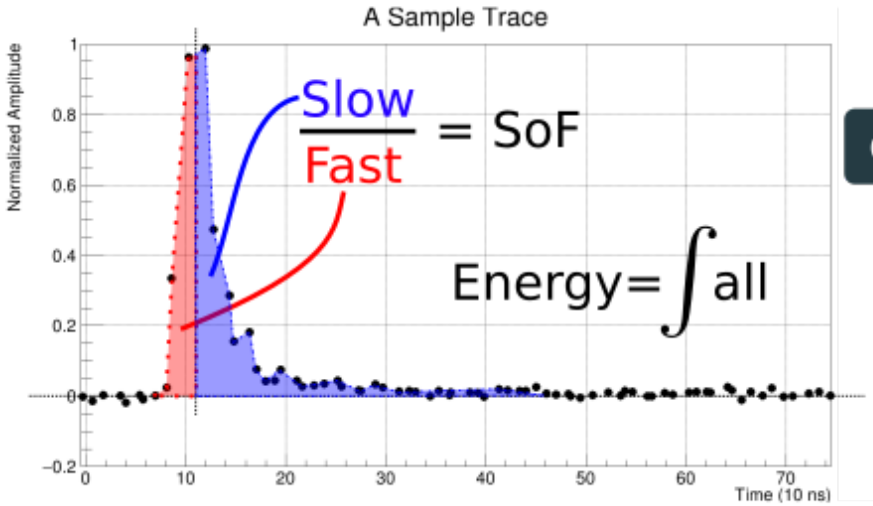
CCPSA actor computes Slow, Fast, Energy

Cuts for final selection required





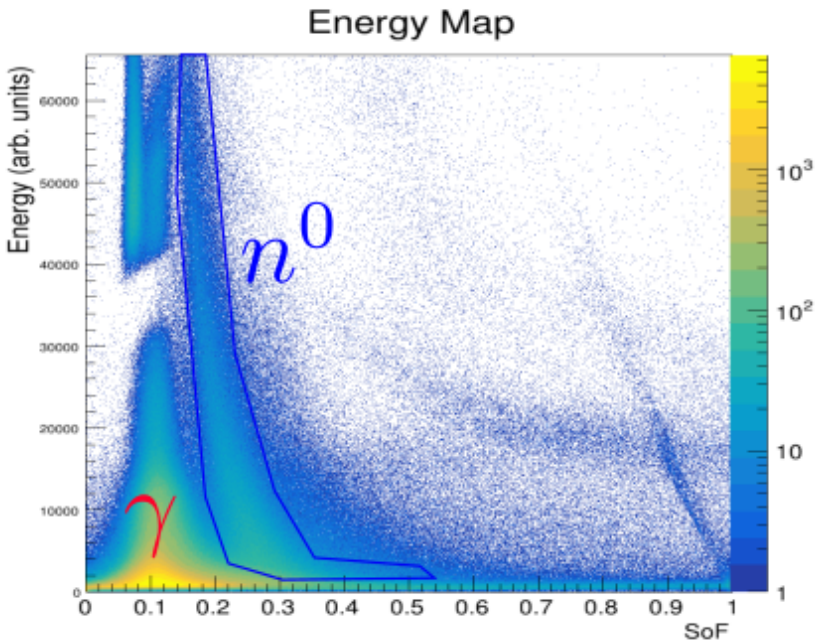
NEDA PSA

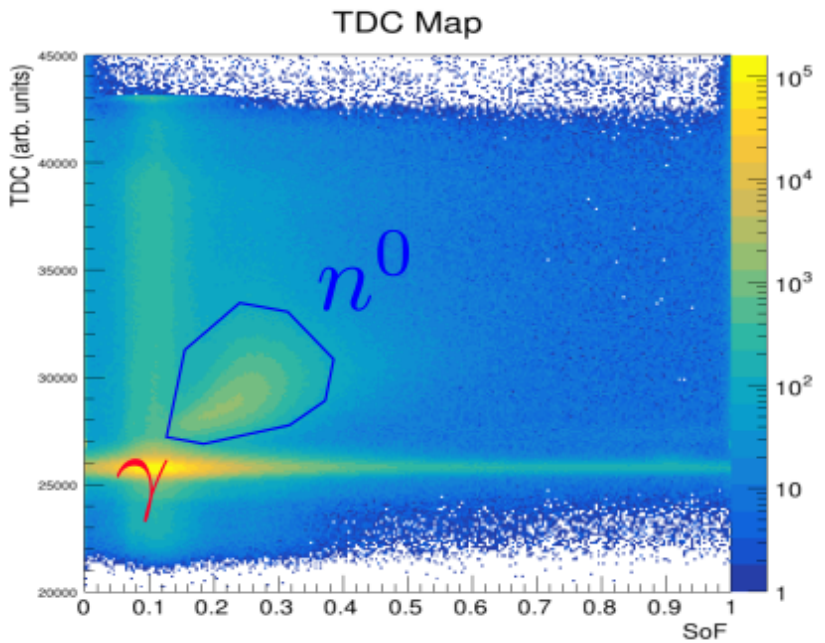


CCPSA actor computes Slow, Fast, Energy

Cuts for final selection required

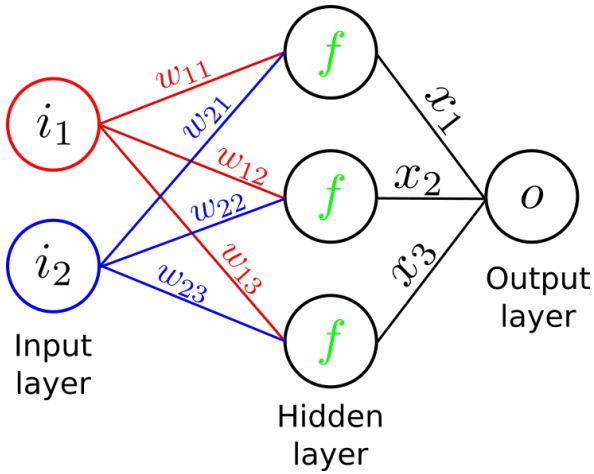
But a cut is "strongly binary"...





Neural Networks – Generalities

$$o = f\left(f\left(\begin{bmatrix} i_1 & i_2 \end{bmatrix}\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix}\right)\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) \Leftrightarrow$$



o: output i: input w/x: $1^{st}/2^{nd}$ set of weights f: activation function

Training

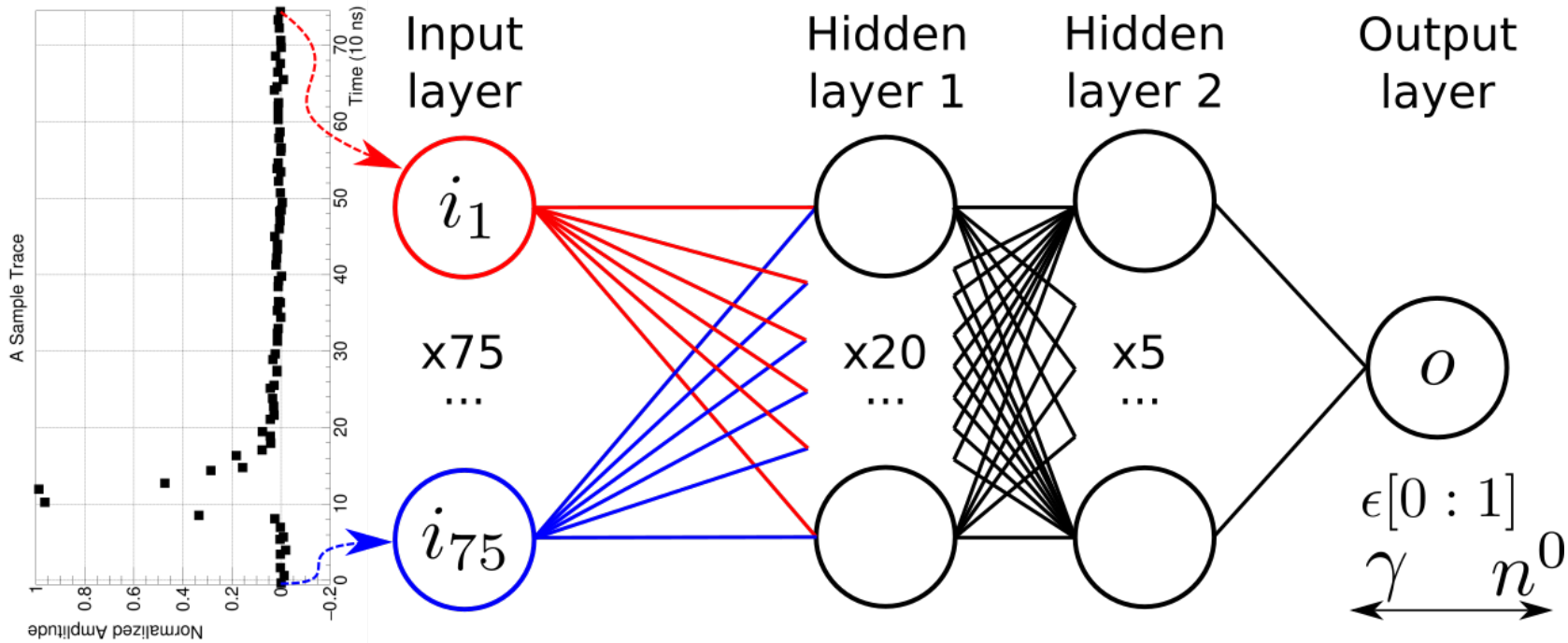
Adjust weights iteratively with a known input-to-output set

Output

Not boolean, but a probability/likeliness

Previous Work

TMultiLayerPerceptron (ROOT)



$\Rightarrow$ Interesting results[1], but computing time online-incompatible

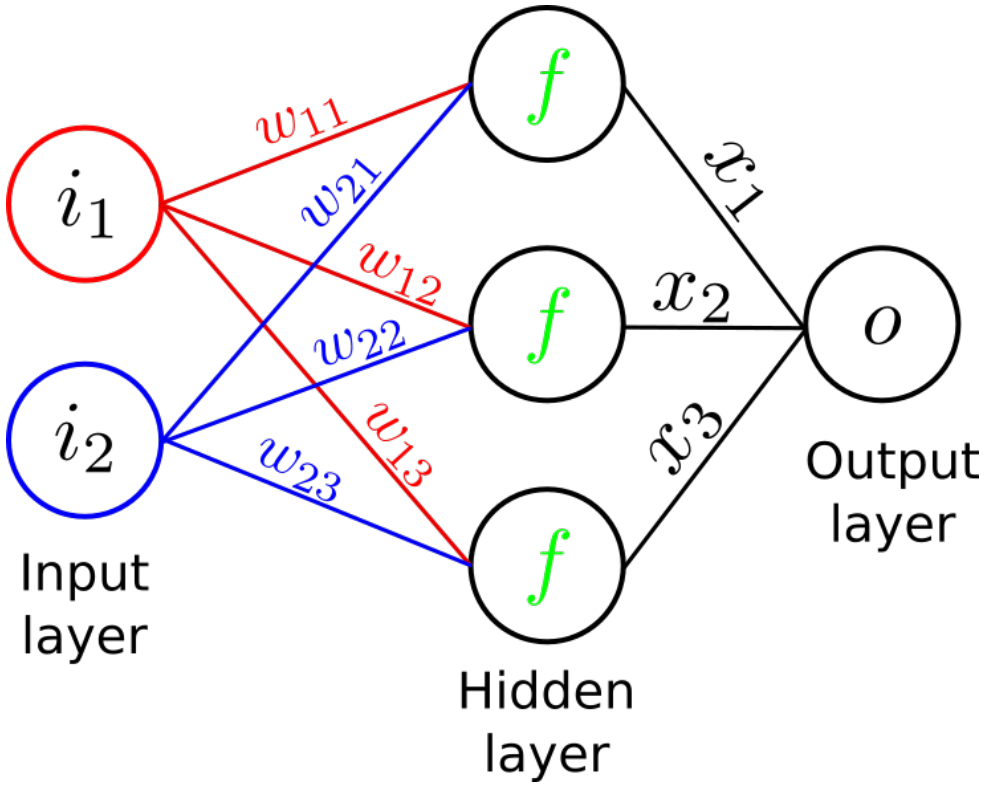
[1] P.-A. Söderström et al., in preparation

Integration in GANPRO

- Library by Google: optimized, documented, active community
- Multiple NN types available
  - Multilayer Perceptron
  - Recurrent NN
- GPU-compatible (Nvidia)
- C++-compatible
  - Integrated in GANPRO
  - NNPSA Filter: buffering & parallelism
  - ↪ has worked online!

Used Networks

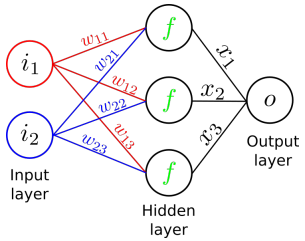
1. MultiLayer Perceptron (MLP)



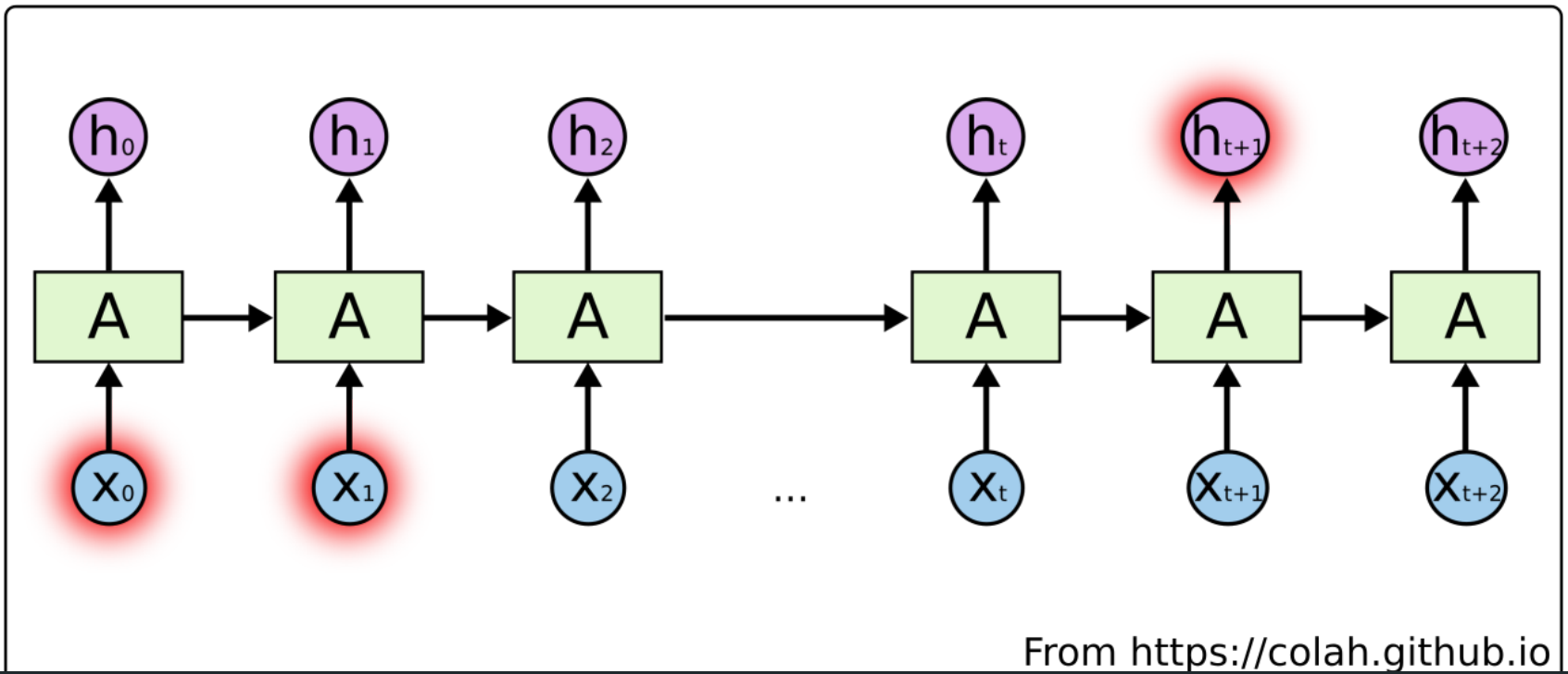
2. Recurrent NN (RNN) – Long Short Term Memory (LSTM)

Used Networks

1. MultiLayer Perceptron (MLP)

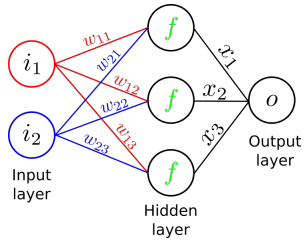


2. Recurrent NN (RNN) – Long Short Term Memory (LSTM)

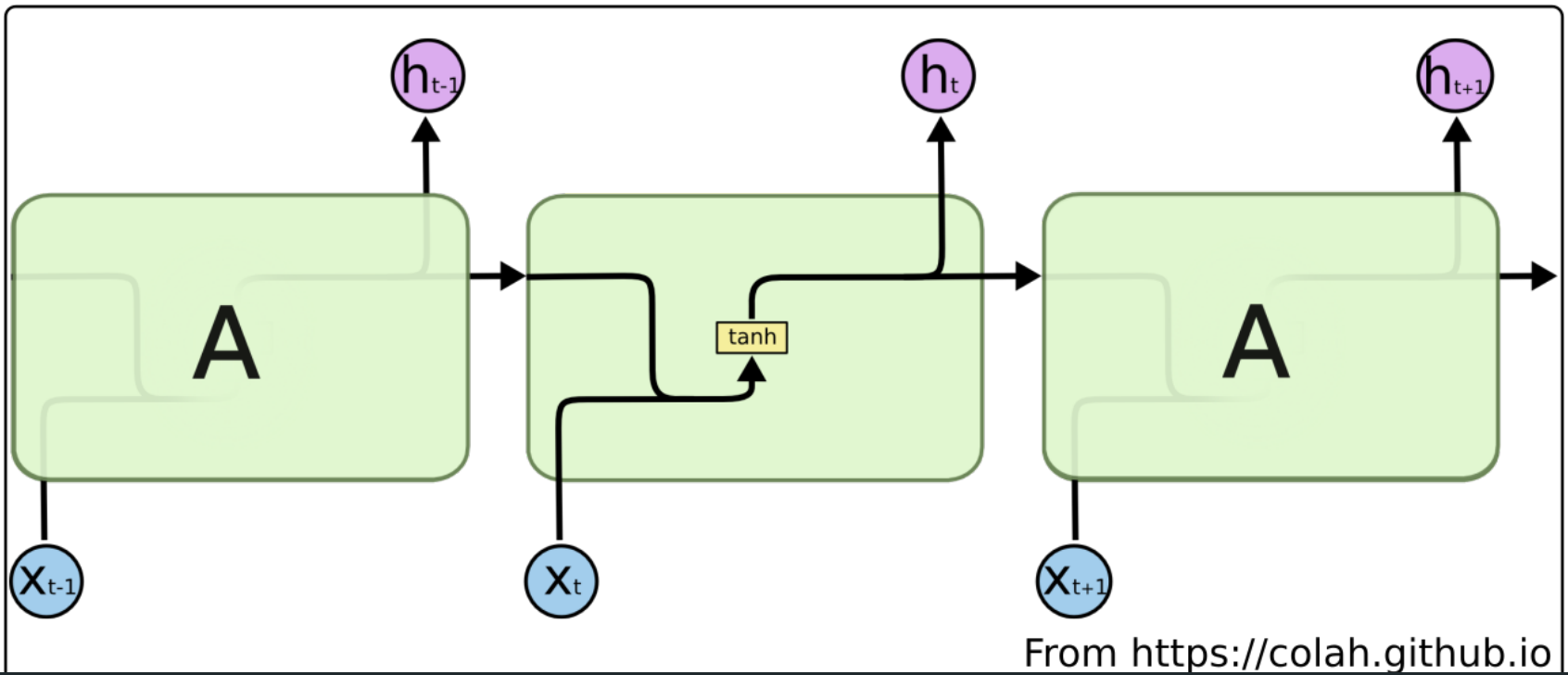


Used Networks

1. MultiLayer Perceptron (MLP)

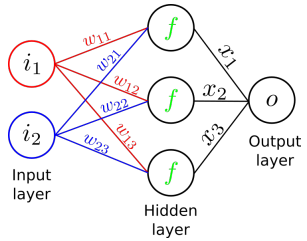


2. Recurrent NN (RNN) – Long Short Term Memory (LSTM)

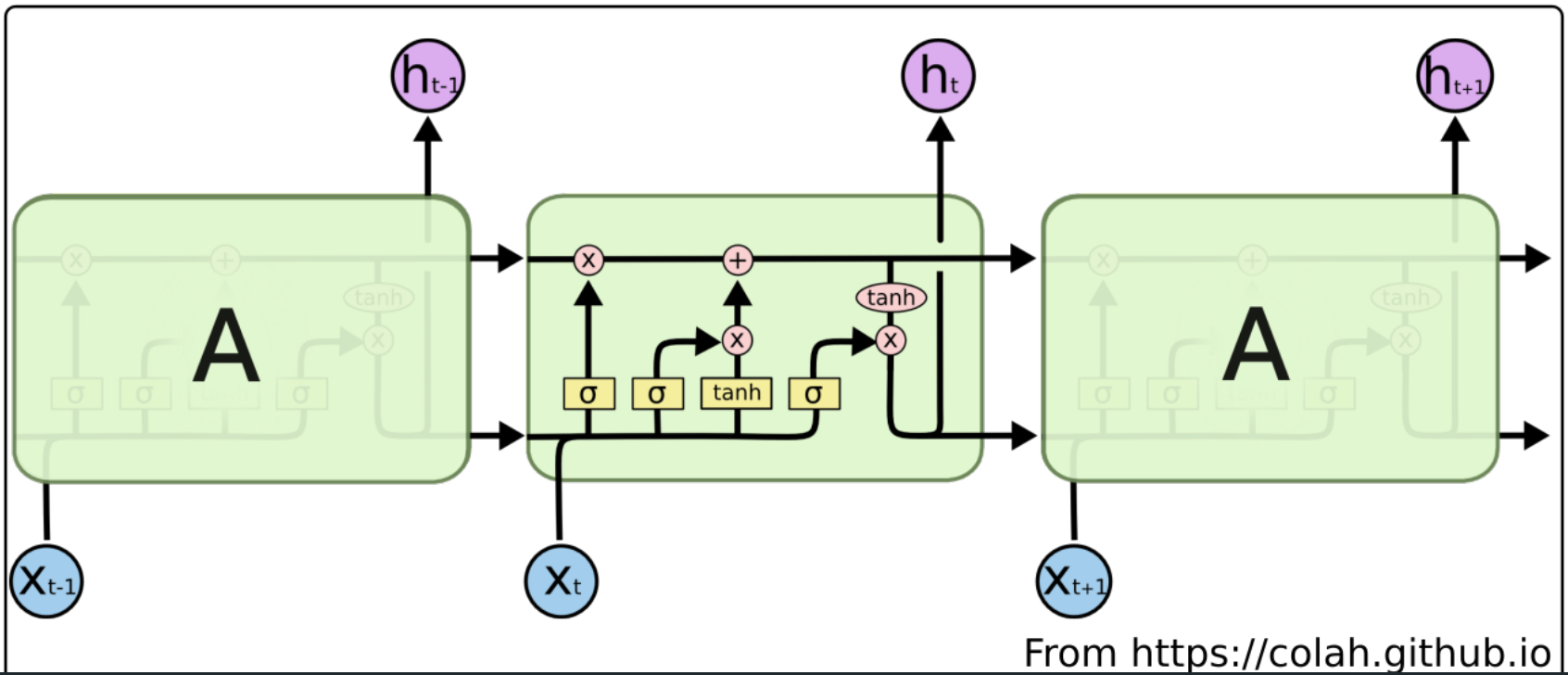


Used Networks

1. MultiLayer Perceptron (MLP)



2. Recurrent NN (RNN) – Long Short Term Memory (LSTM)



Training

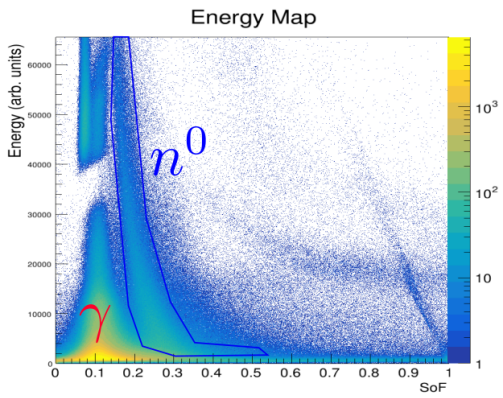
Crucial step, sets the discrimination quality

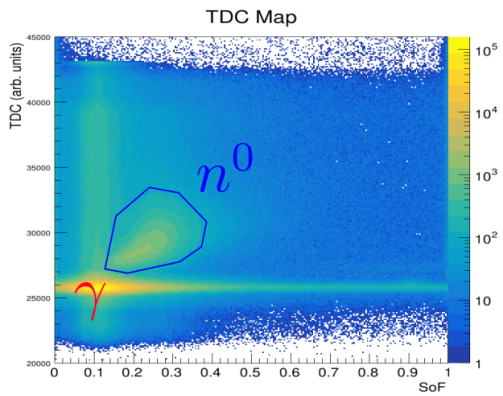
Sensitivity to t_0

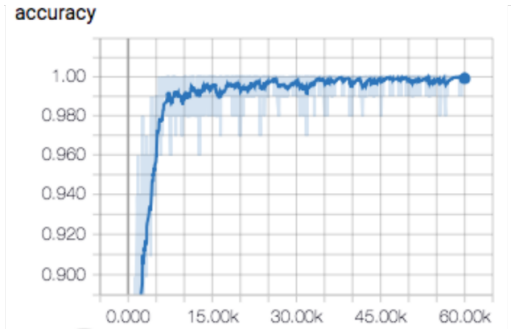
Training with blocks of 100 signals and averaged error

Training dataset

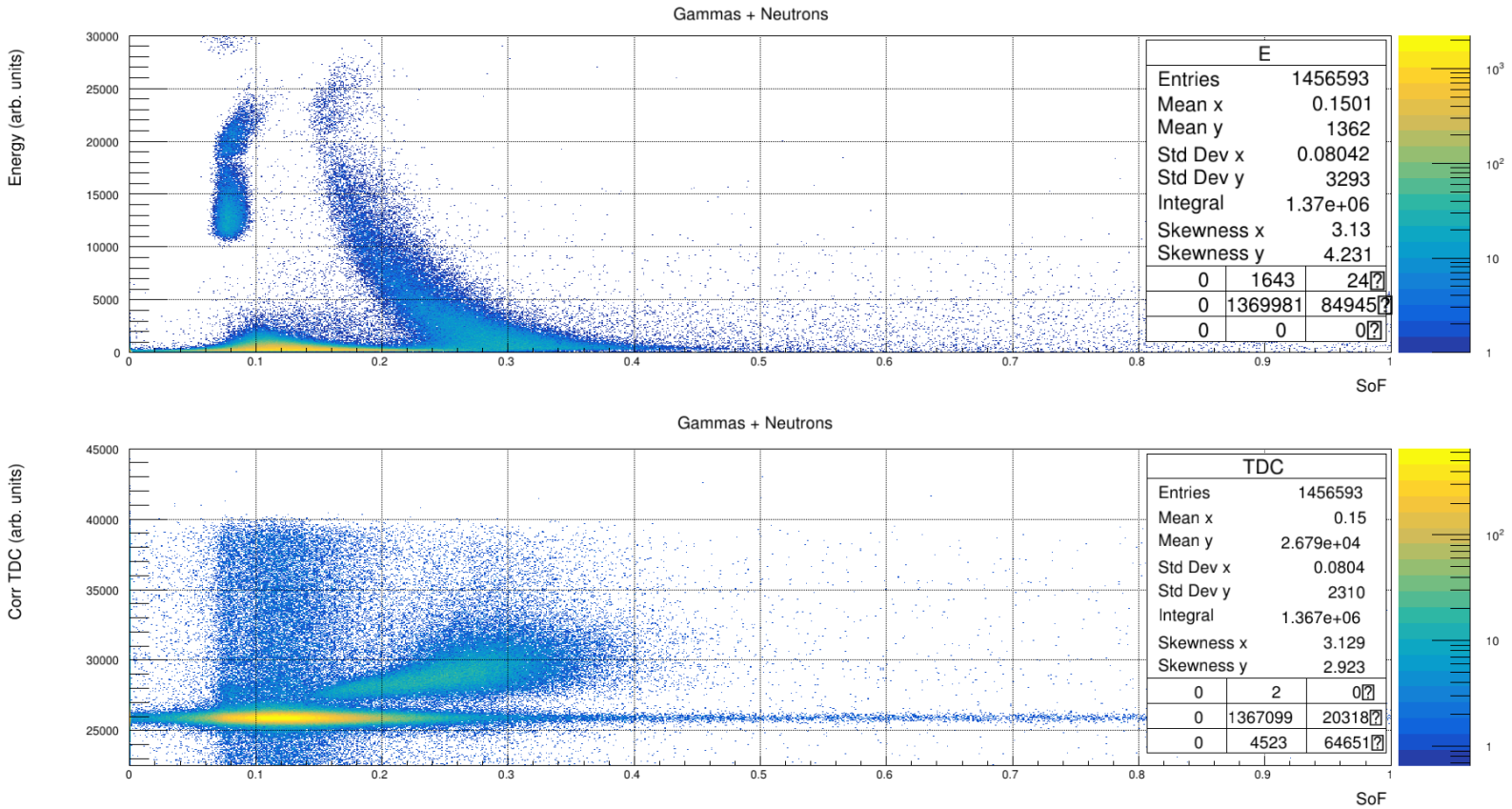
- Ideal: events labelled with 100% confidence
- For now: CCPSA cuts $\Rightarrow$ fine tuning mandatory!



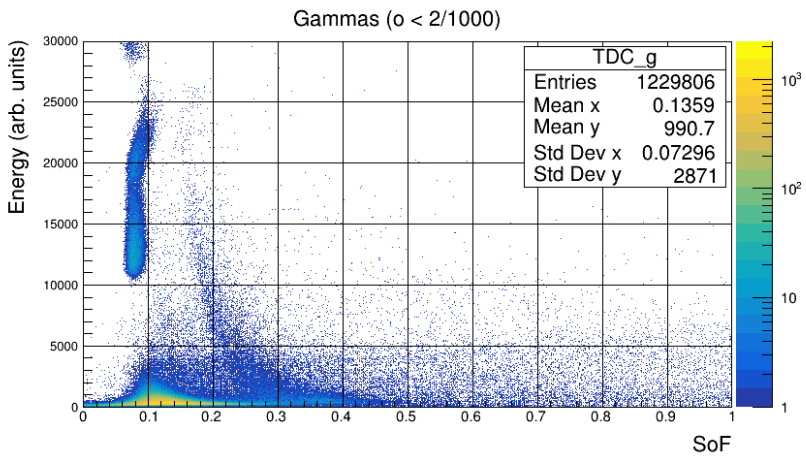


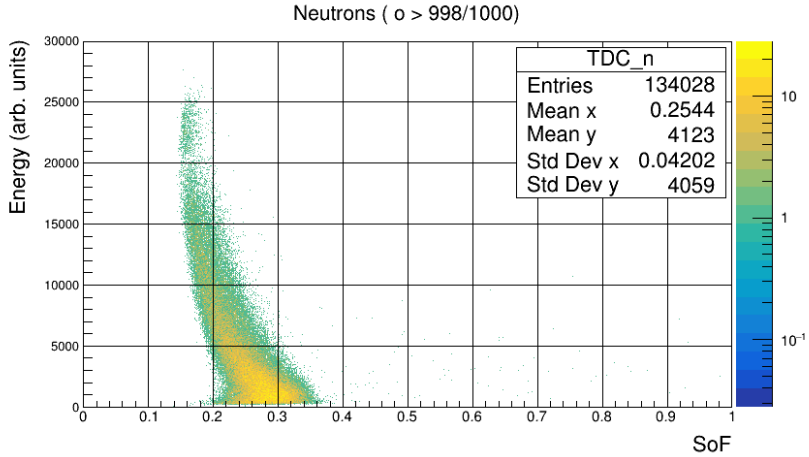


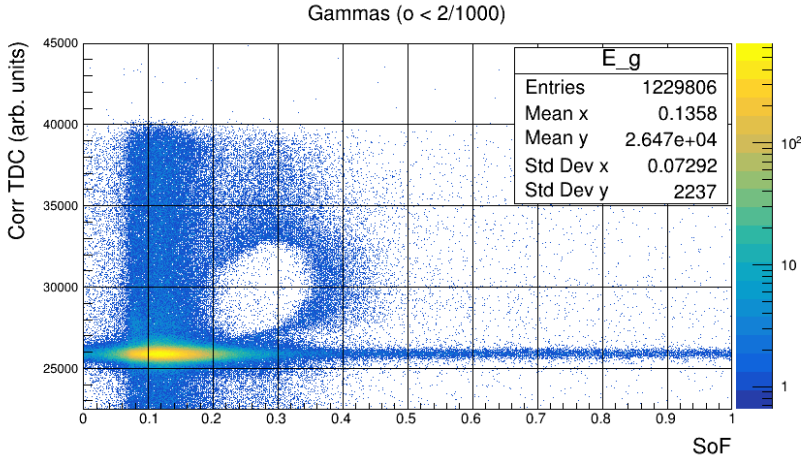
Selectivity Example

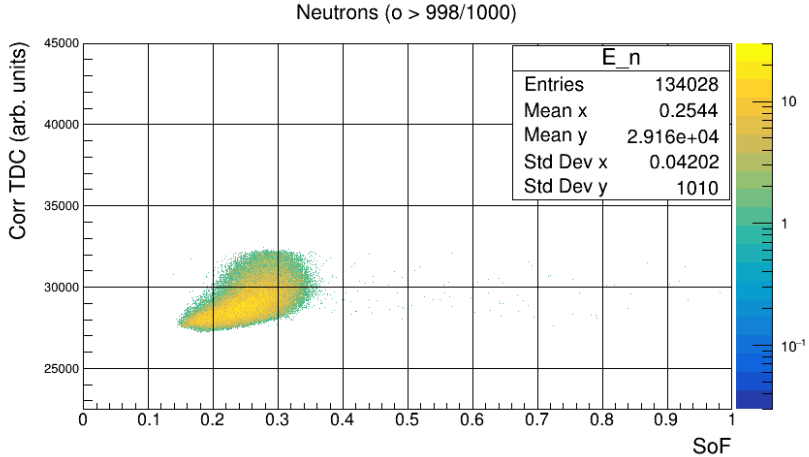


Selectivity Example



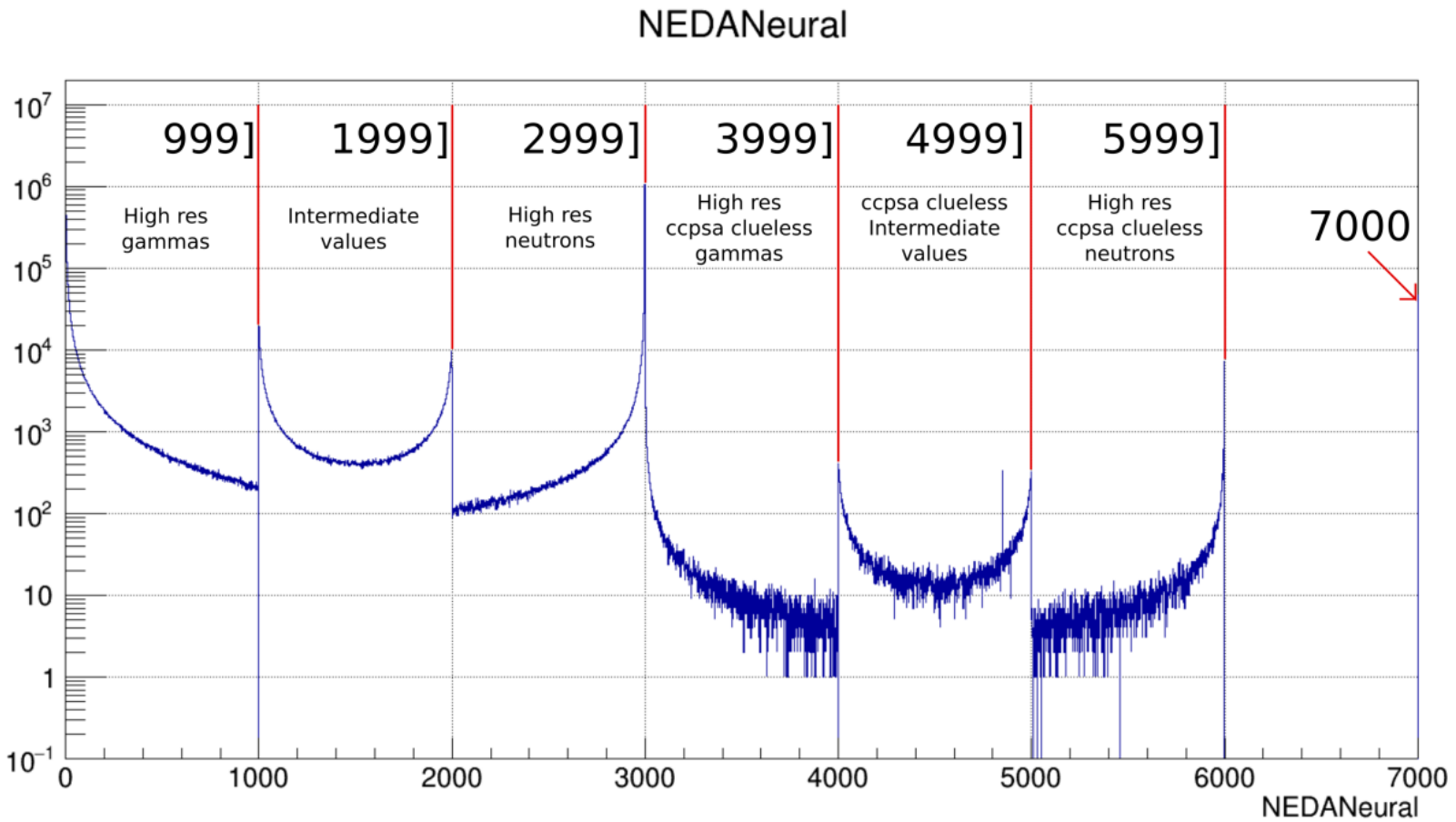




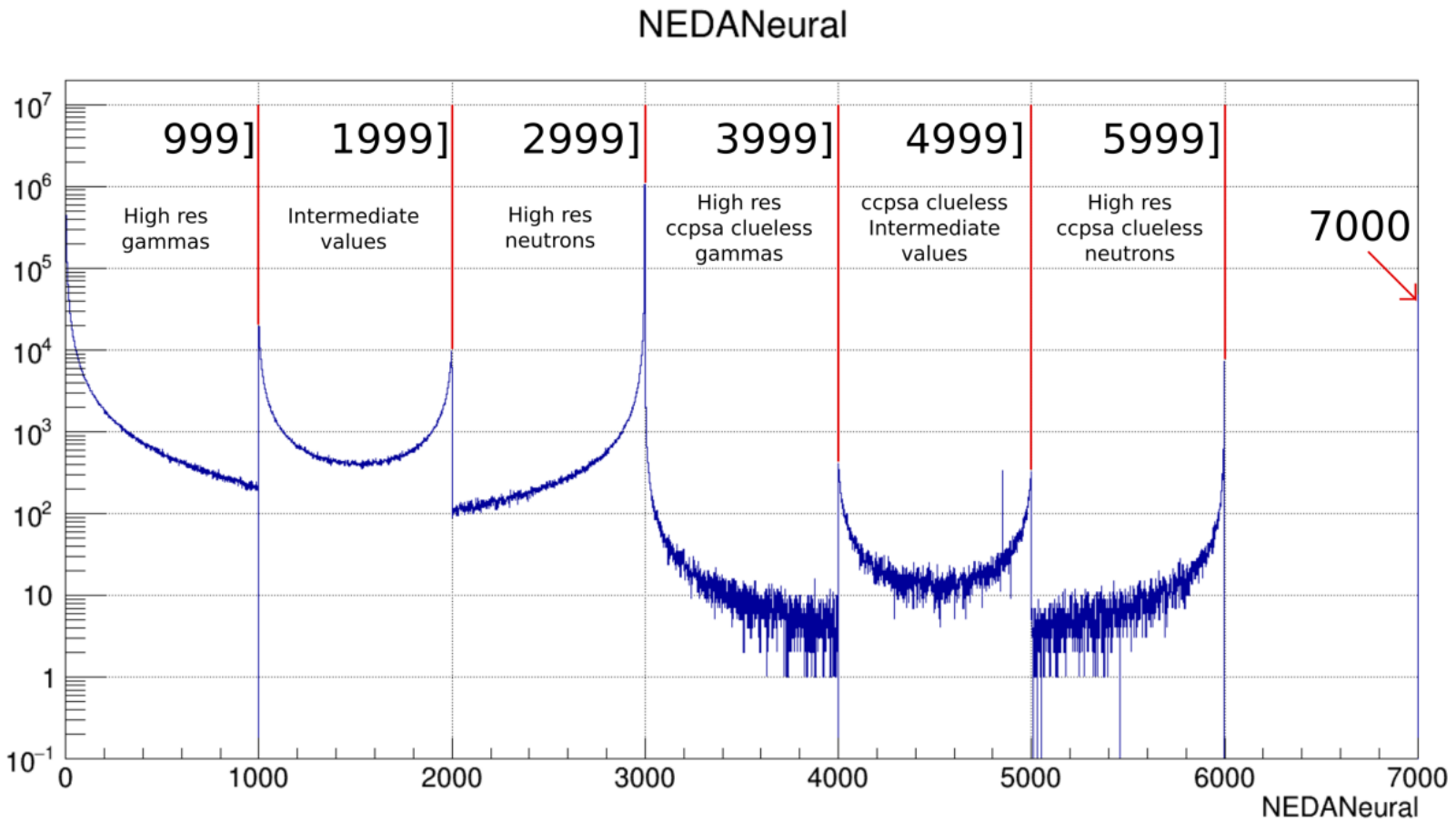


$\Rightarrow$ Extreme condition on output values is compatible with training cuts

Output values

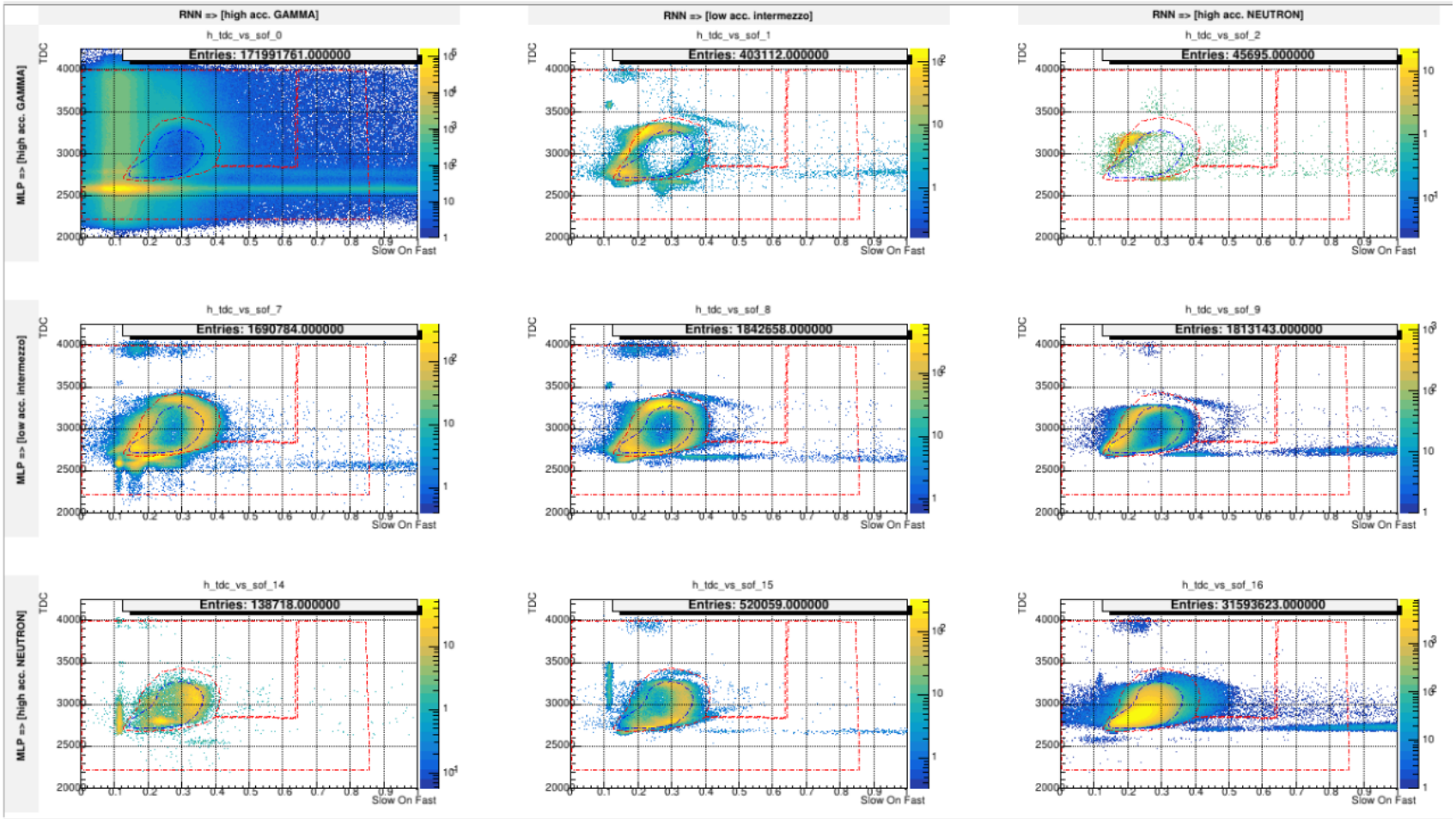


Output values

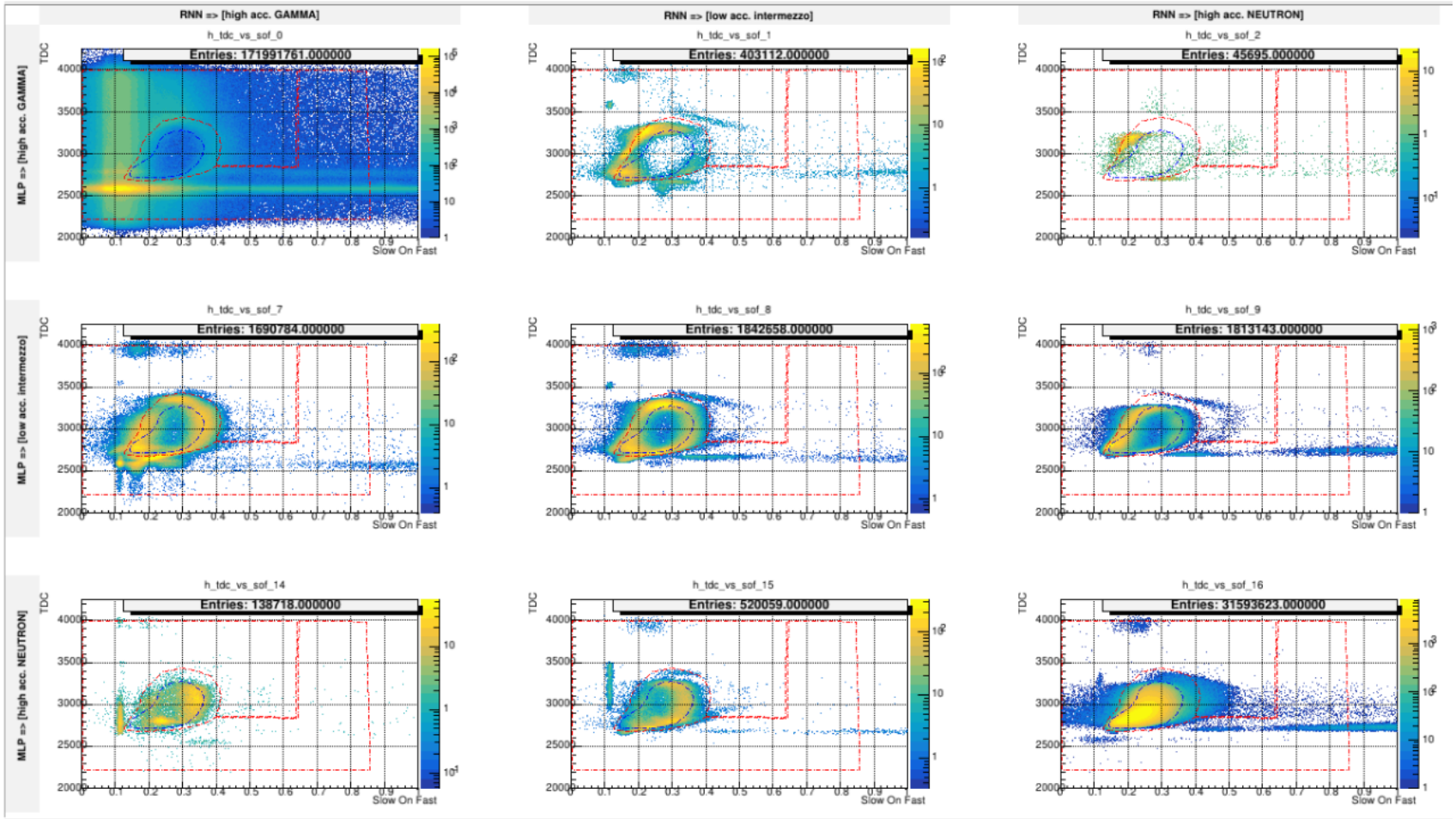


⇒ What data shows up for MLP vs RNN decision?

"Cross-cases"

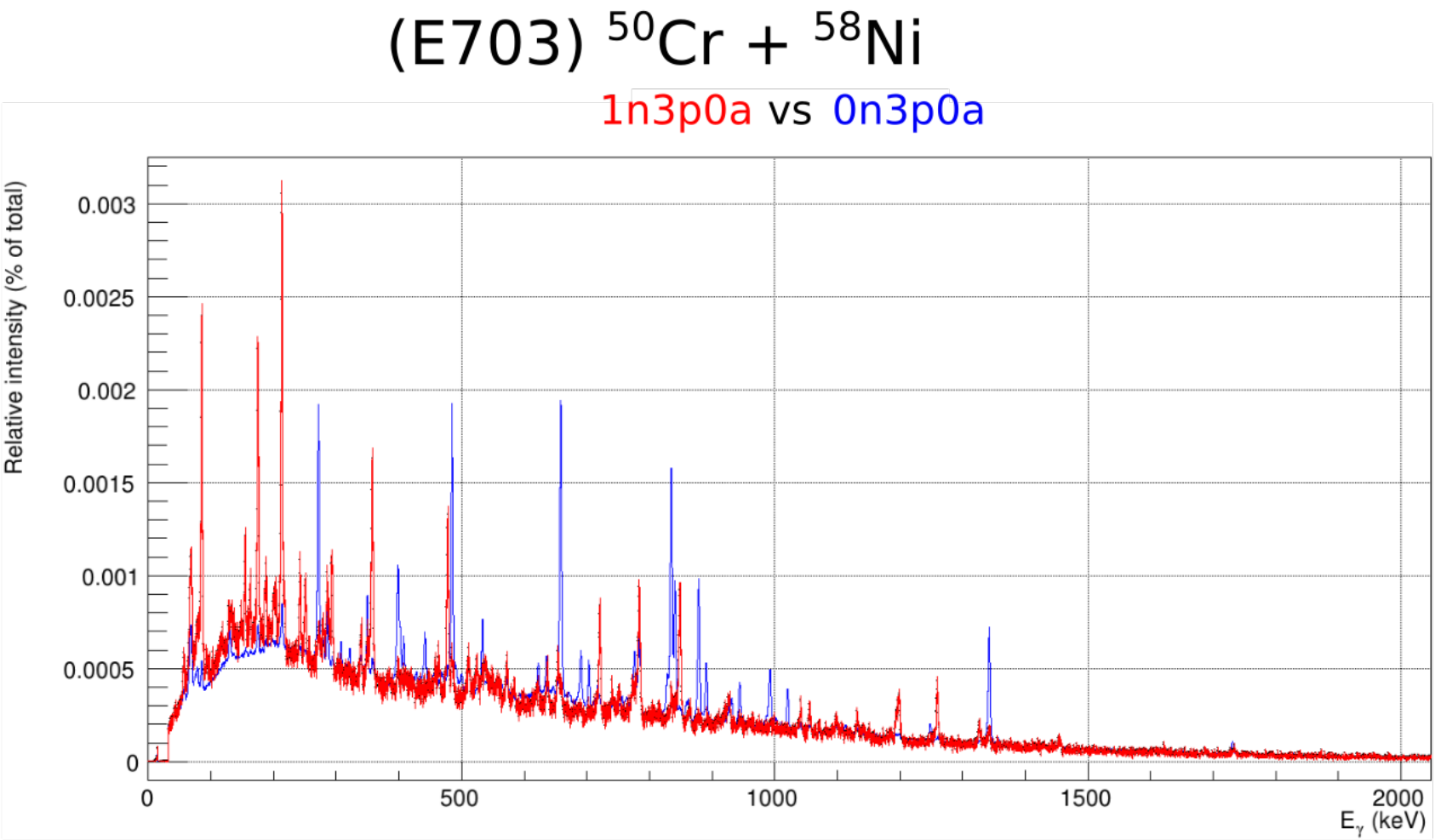

RNN => [high acc. GAMMA]
RNN => [low acc. intermezzo]
RNN => [high acc. NEUTRON]
MLP => [high acc. GAMMA]
MLP => [low acc. intermezzo]
MLP => [high acc. NEUTRON]
h_tdc_vs_sof_0
Entries: 171991761.000000
h_tdc_vs_sof_1
Entries: 403112.000000
h_tdc_vs_sof_2
Entries: 45695.000000
h_tdc_vs_sof_7
Entries: 1690784.000000
h_tdc_vs_sof_8
Entries: 1842658.000000
h_tdc_vs_sof_9
Entries: 1813143.000000
h_tdc_vs_sof_14
Entries: 138718.000000
h_tdc_vs_sof_15
Entries: 520059.000000
h_tdc_vs_sof_16
Entries: 31593623.000000
TDC
Slow On Fast

"Cross-cases"

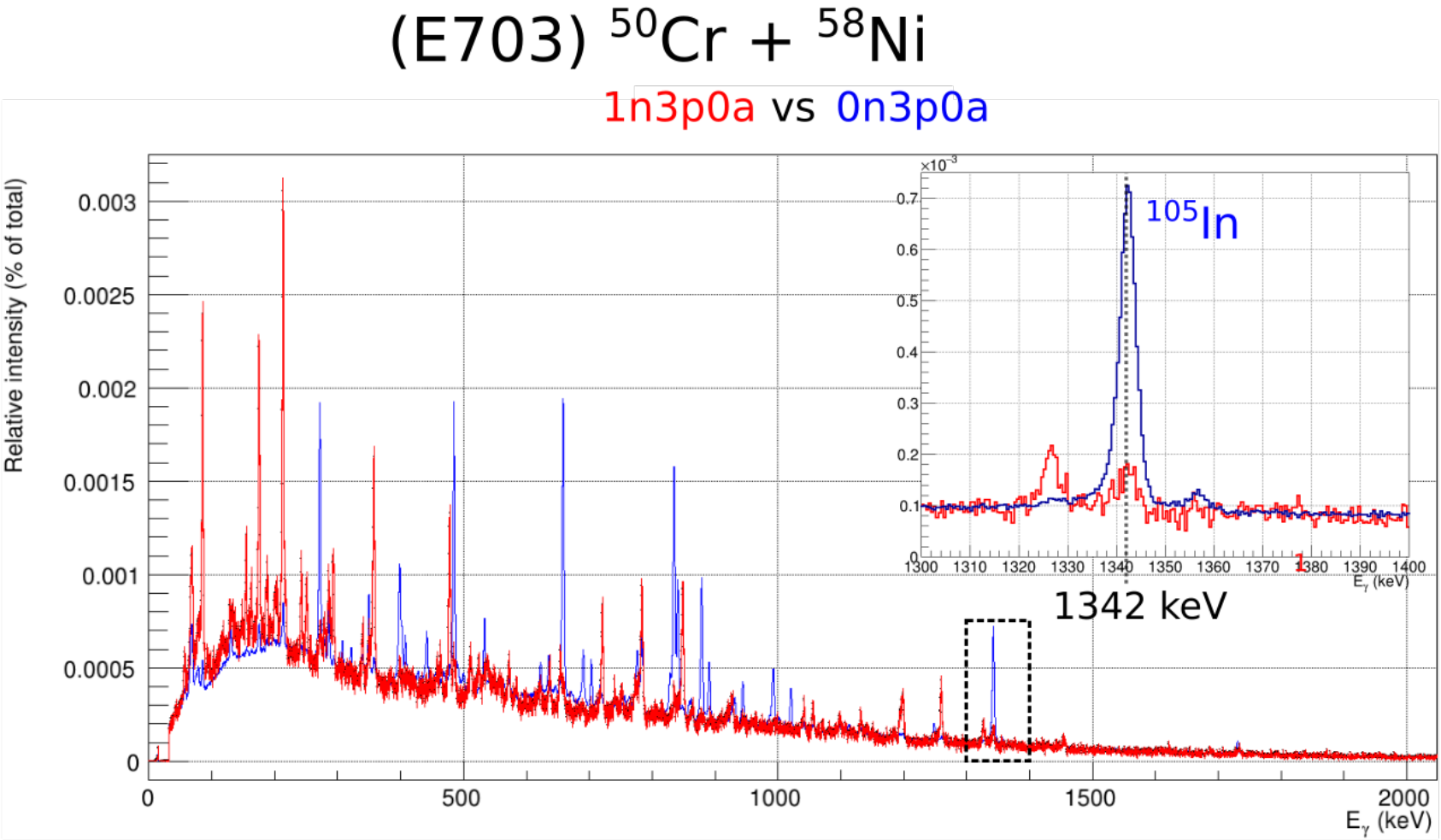


=> Who's right ?

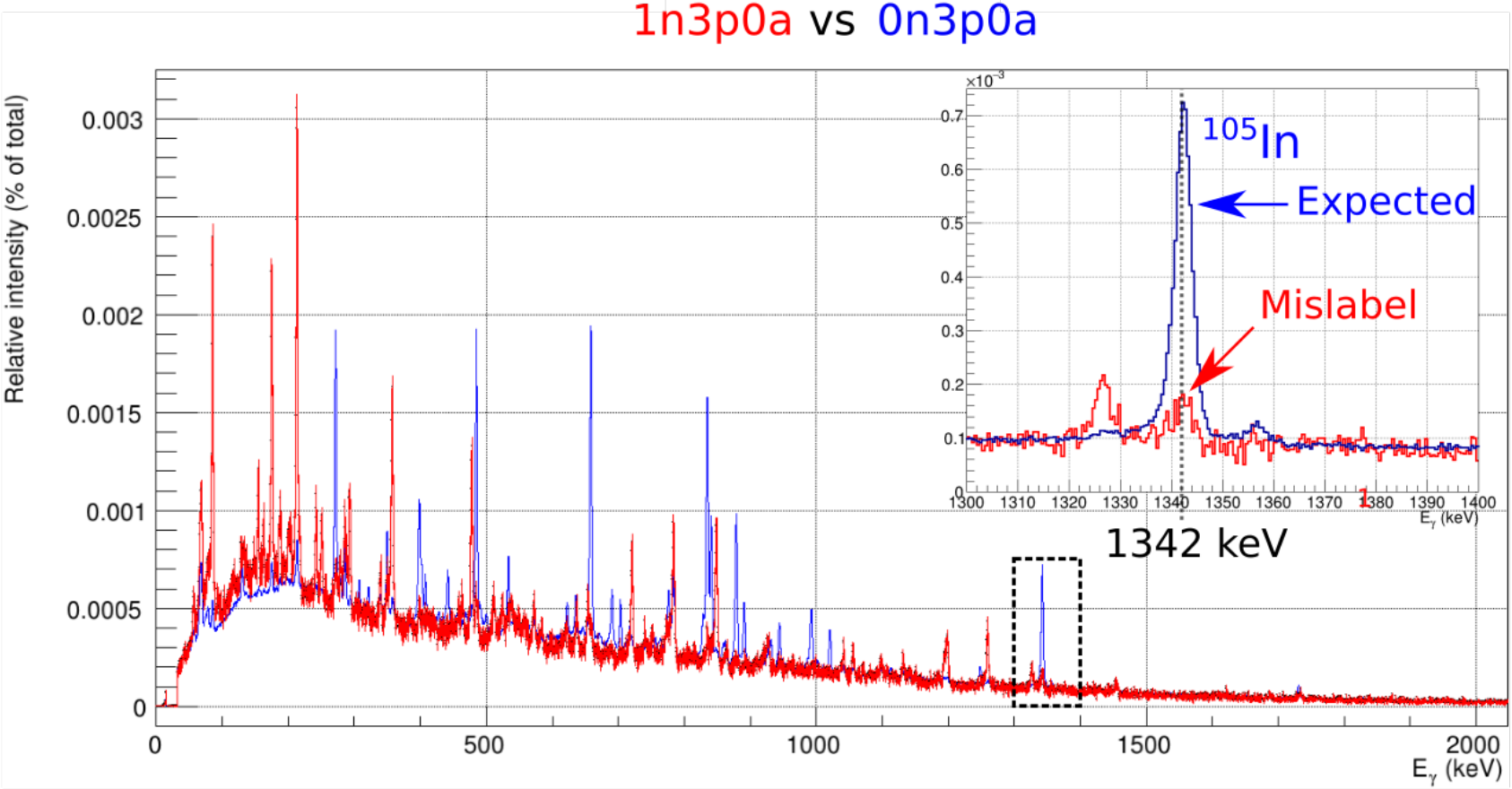
Quantify Success



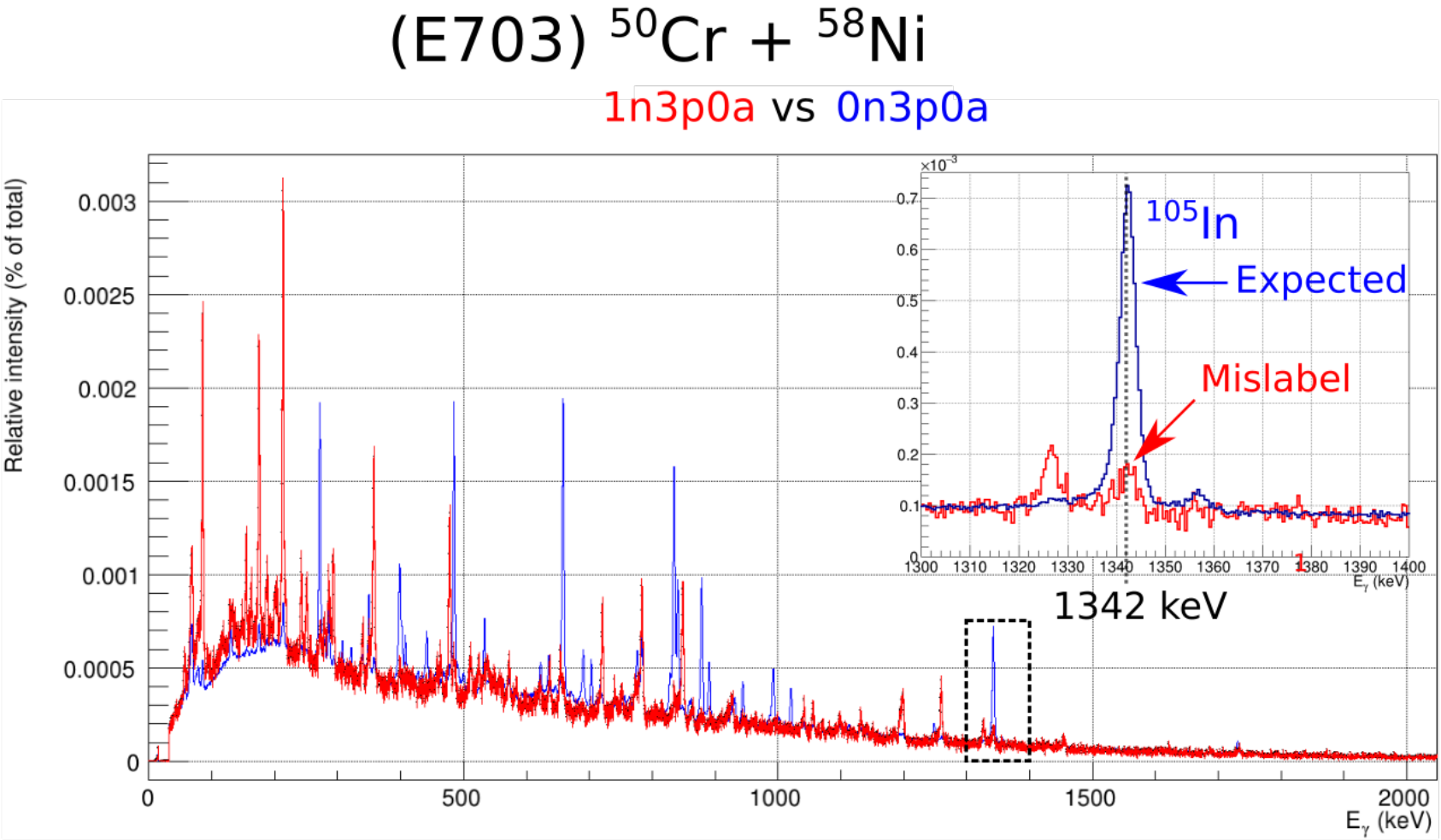
Quantify Success



Quantify Success

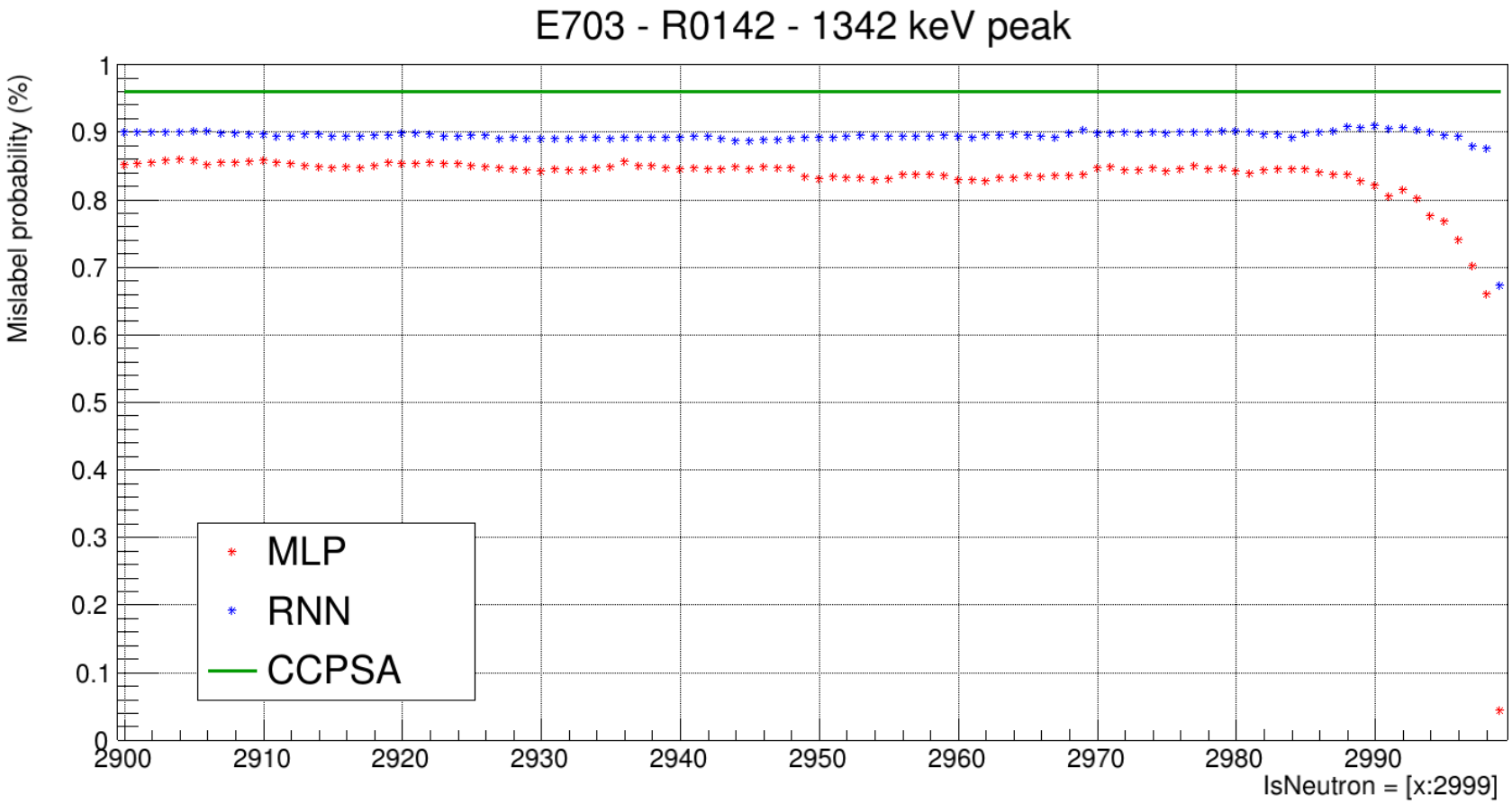


Quantify Success

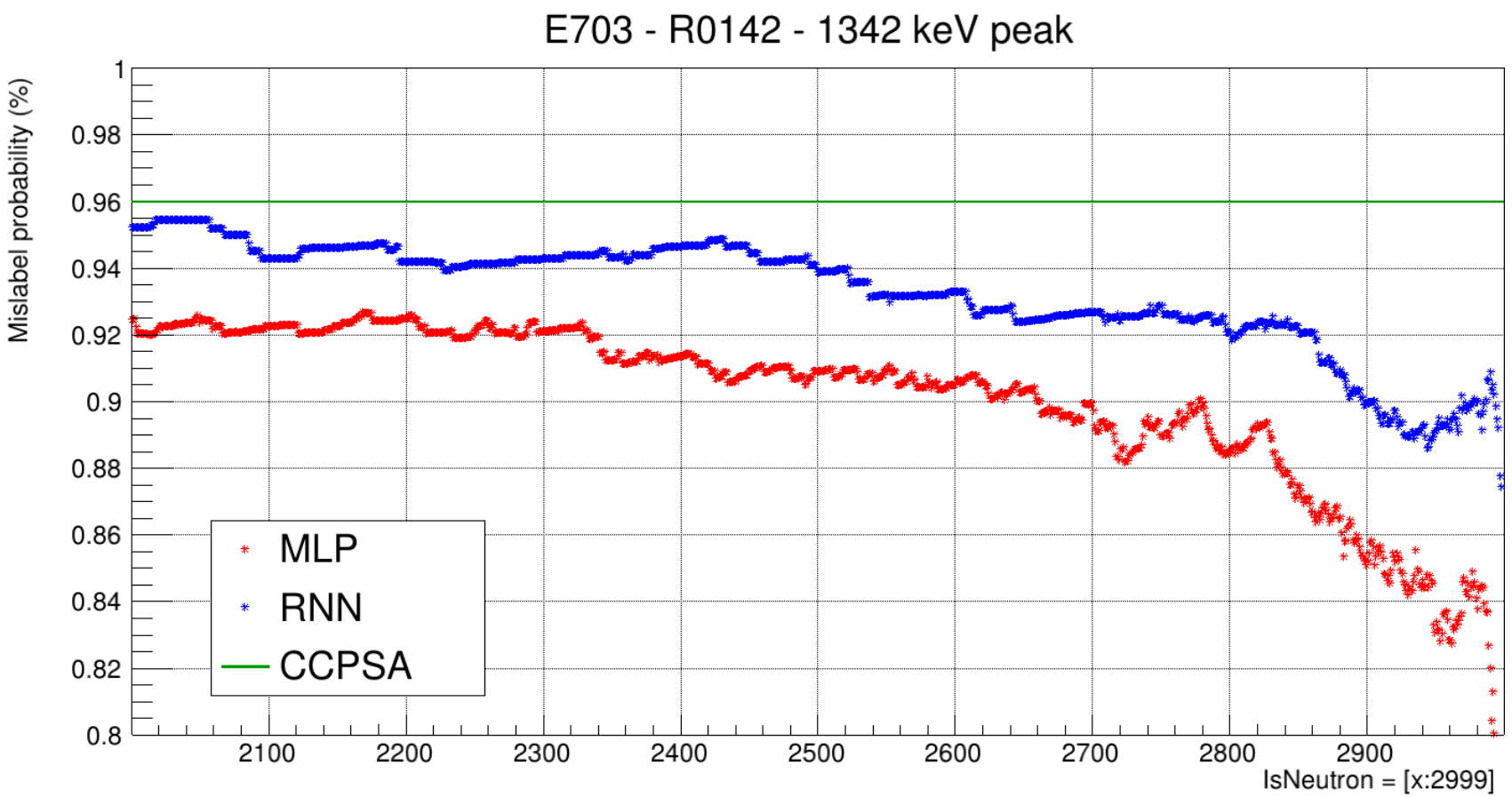


=> Computation of a mislabel probability

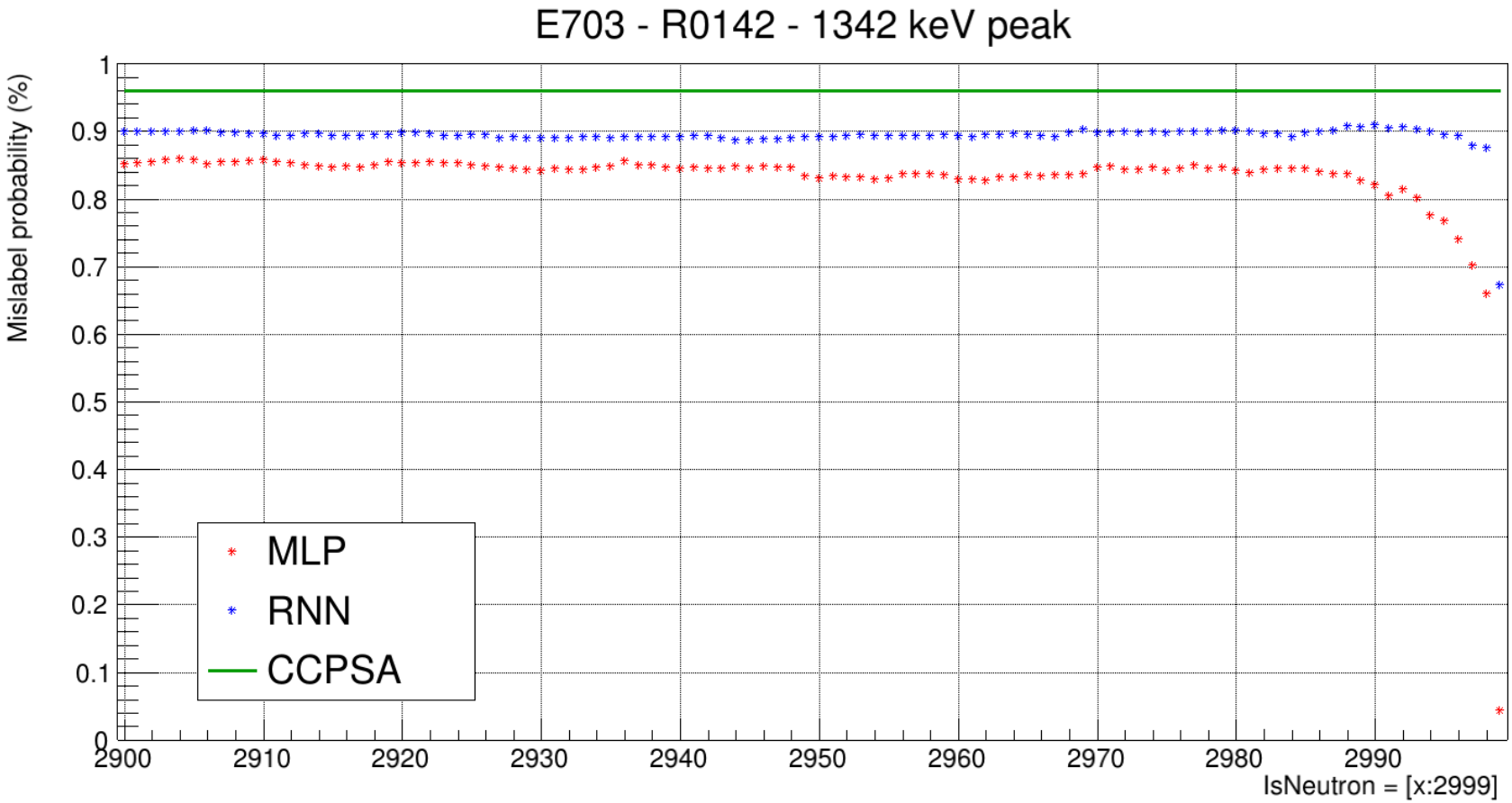
Mislabel probability



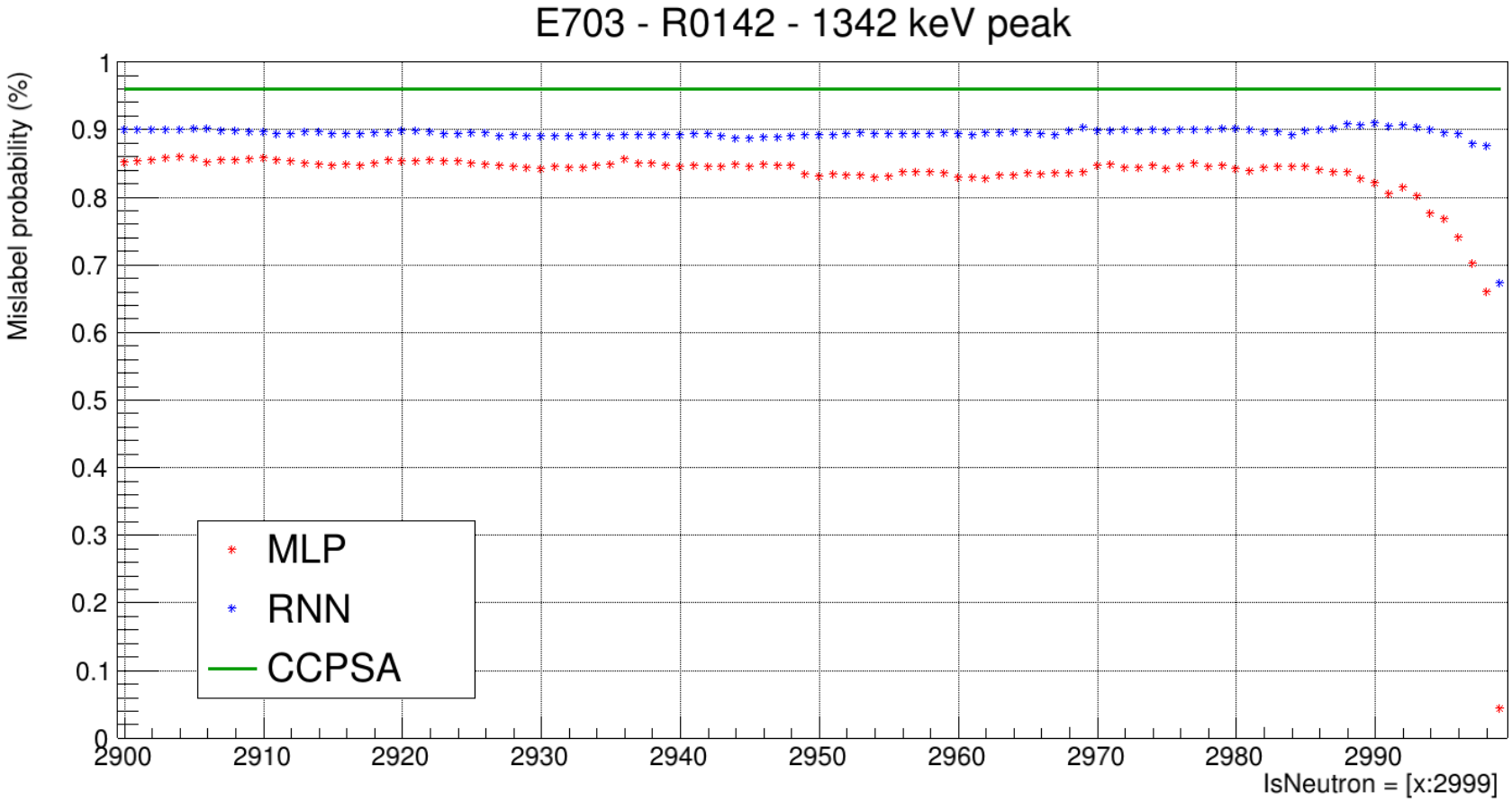
Mislabel probability



Mislabel probability

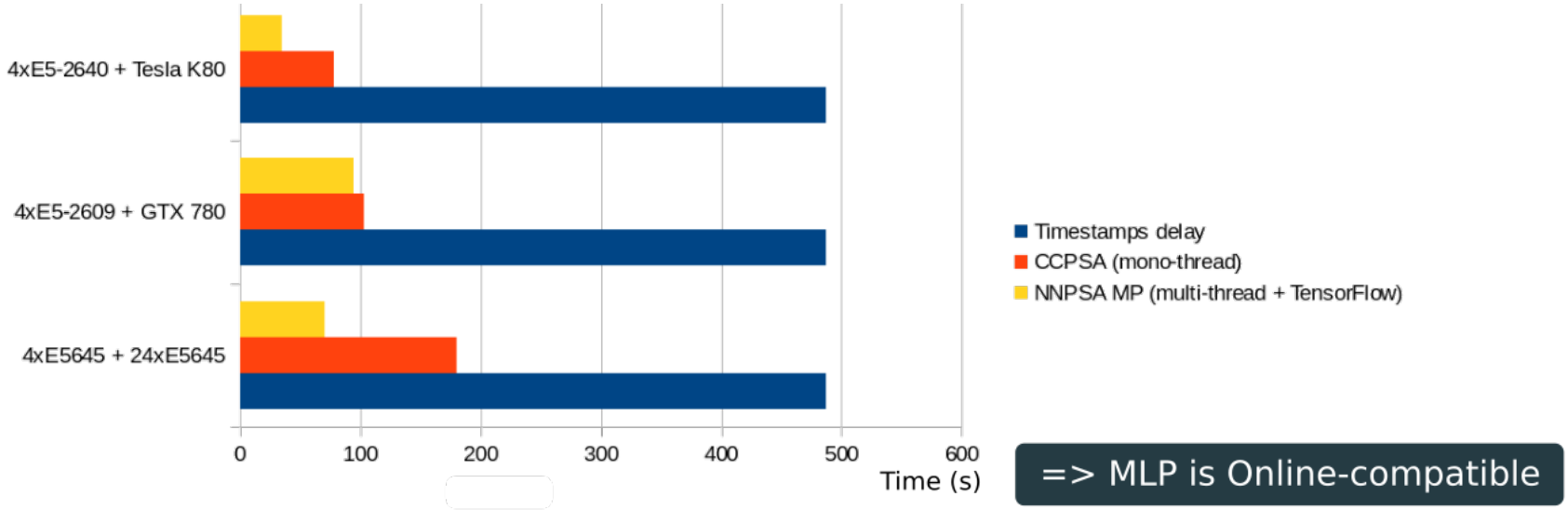


Mislabel probability



$\Rightarrow$ 0.1% was achieved!

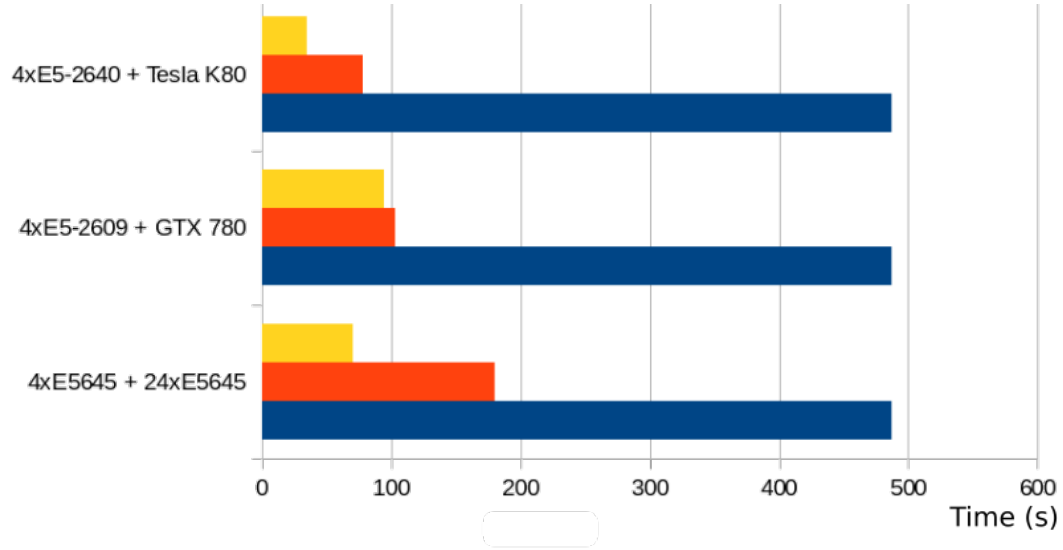
Computing Time

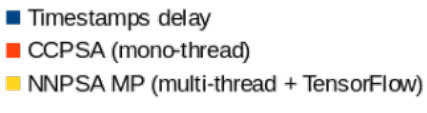


=> MLP is Online-compatible

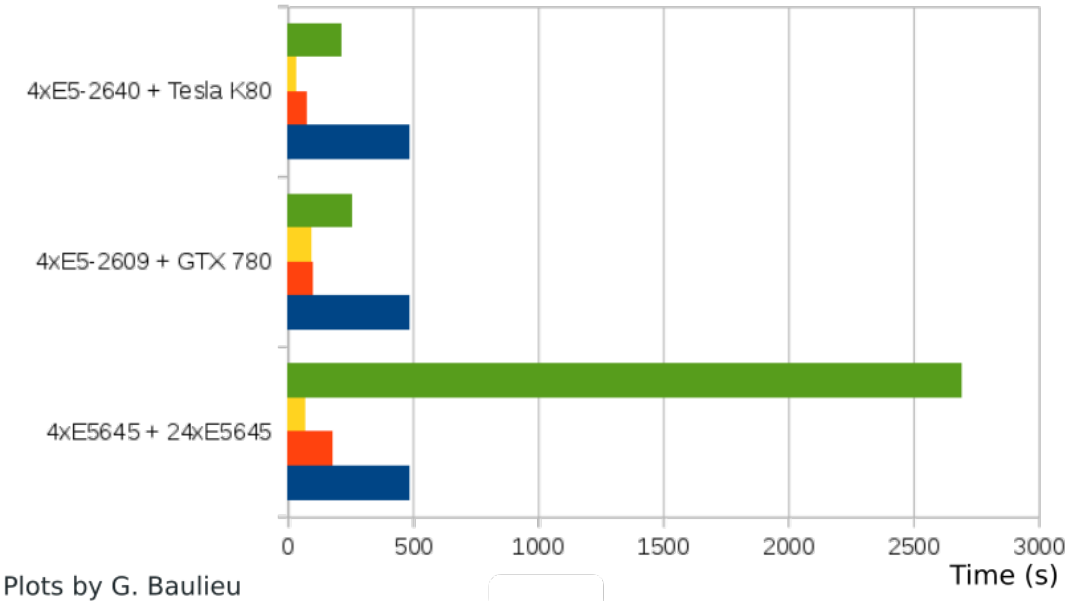
Plots by G. Baulieu

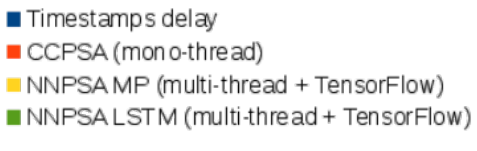
Computing Time





=> MLP is Online-compatible



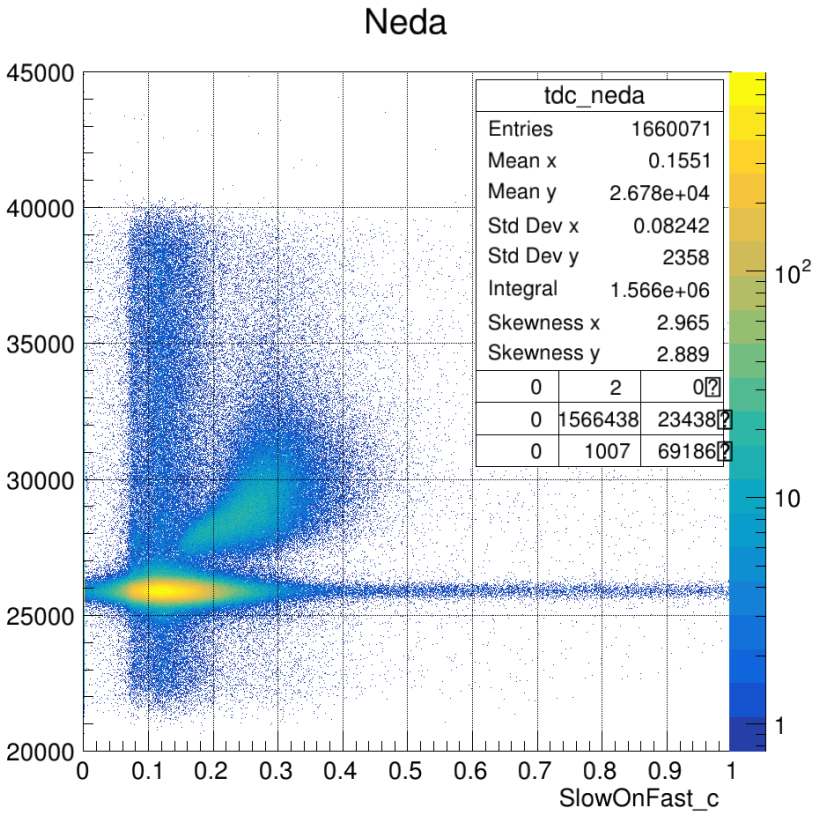


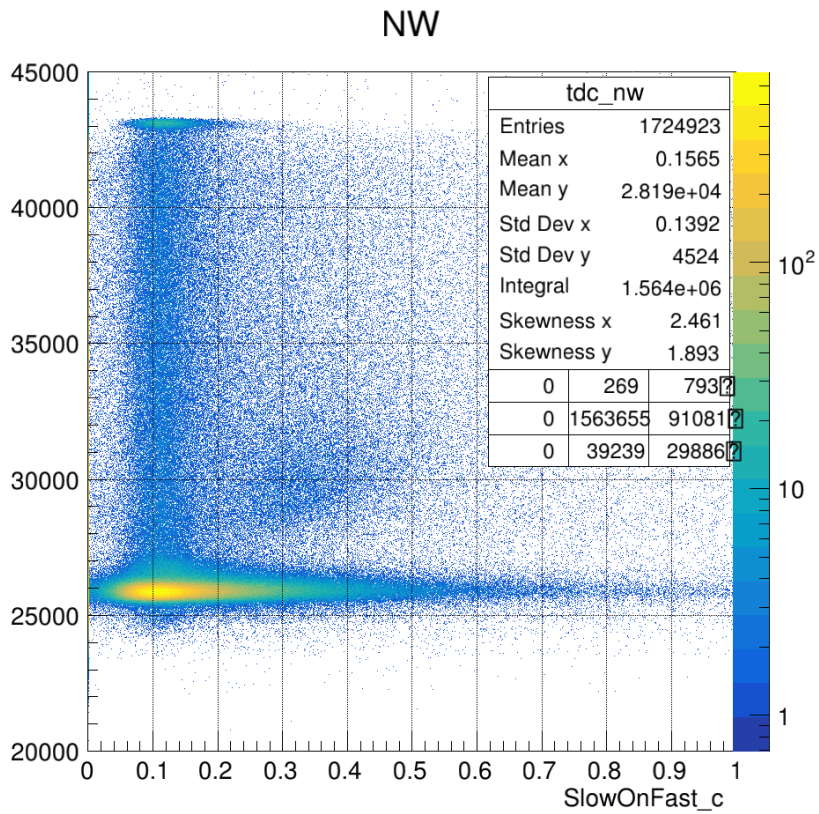
=> GPU required for RNN

Plots by G. Baulieu

Neutron Wall

Same networks as NEDA with a dedicated training





NNPSA actor is 2-buffer/2-network ready

Summary

Conclusions

- NNPSA actor operational
  - MLP: Online-compatible
  - RNN: GPU required
- 0.1% mislabel probability was achieved!

ToDo list

- Need training data of quality
- Need better encoding of NN output value
- Need to check the gamma spectra of the cross-cases
- Need a straightforward way to choose the final quality

IPN Lyon NN taskforce:

G. Baulieu, L. Ducroux, X. Fabian, O. Stezowski