Applied Machine Learning for gamma-neutron discrimination: studies

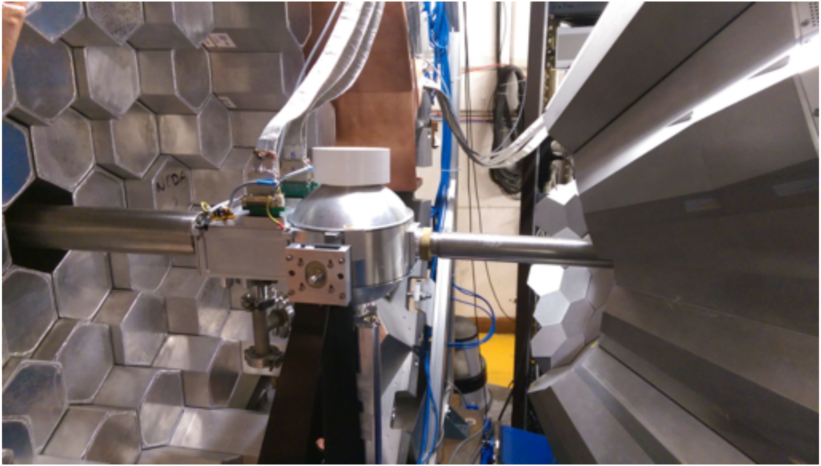
Xavier FABIAN

Univ Lyon, Univ Claude Bernard Lyon 1
CNRS/IN2P3, IP2I Lyon, F-69622, Villeurbanne, France

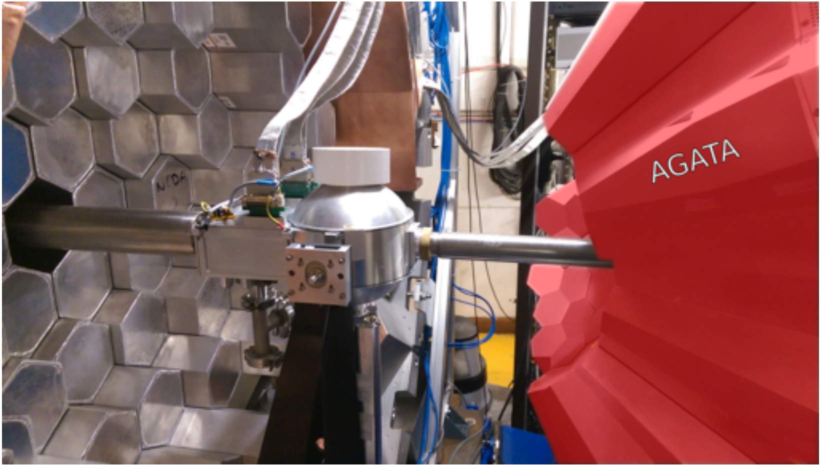
Journée(s) Machine Learning et Physique Nucléaire
Orsay, October 2019

Menu

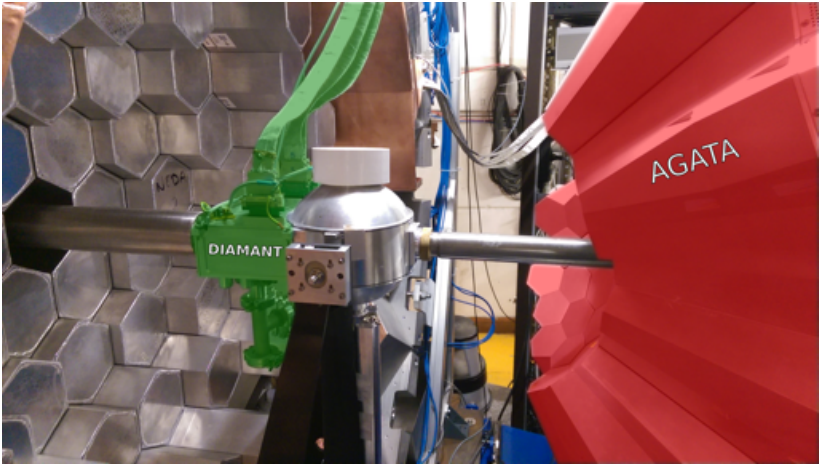
2018 Setup



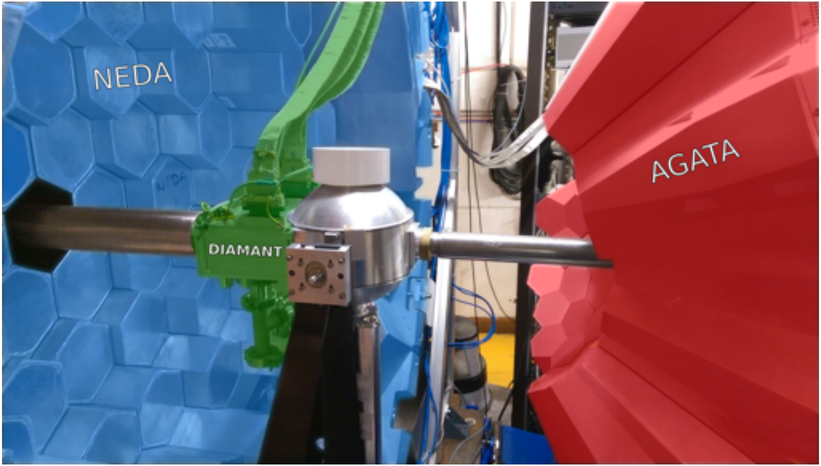
2018 Setup



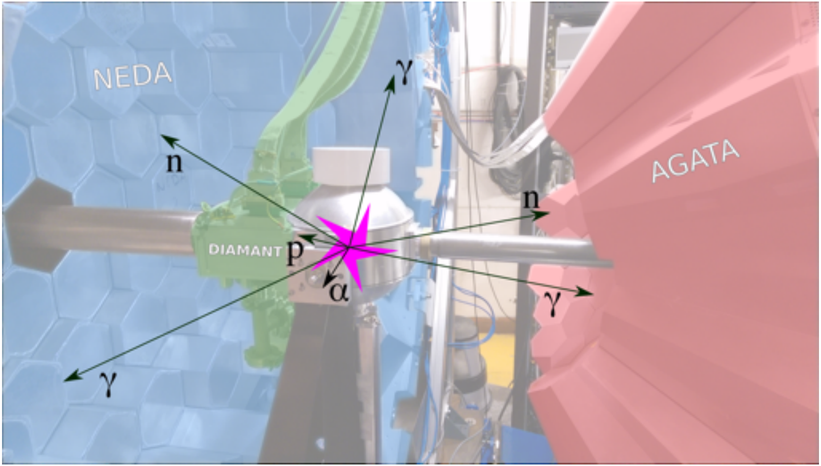
2018 Setup



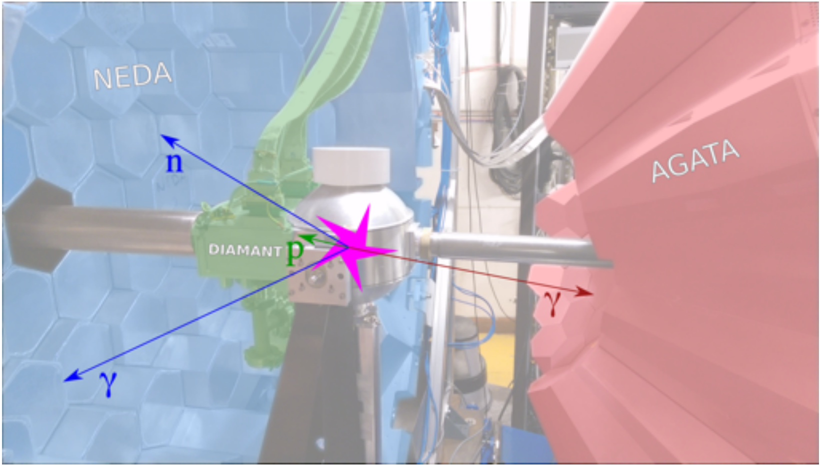
2018 Setup



2018 Setup



2018 Setup



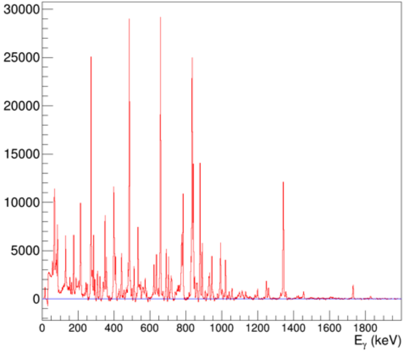
2018 Setup

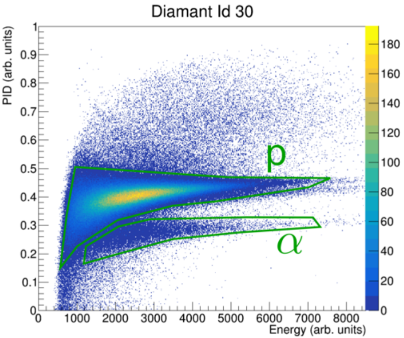
AGATA

- 30 Germanium crystals, 36 segments per crystal
- γ detector array

DIAMANT

- 60 scintillators
- proton and α filter

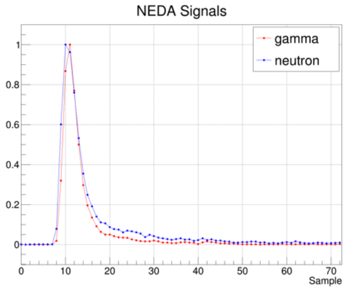


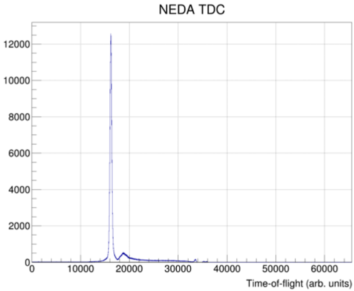


2018 Setup

NEDA

- 54 scintillators (+42 for Neutron Wall, unused here)
- Filters a number of neutrons using:

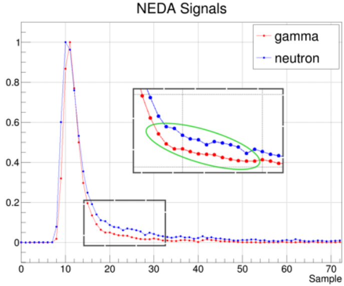


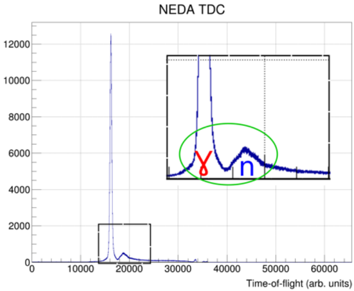


2018 Setup

NEDA

- 54 scintillators (+42 for Neutron Wall, unused here)
- Filters a number of neutrons using:



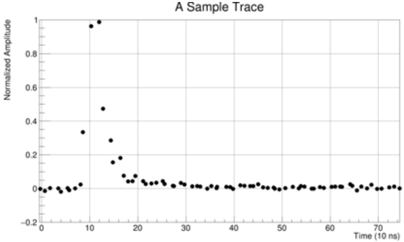


$\Rightarrow$ *n*-γ discrimination is based on:

- Signal: *n* slow component $>$ γ slow component
- TDC: *n* time-of-flight $>$ γ time-of-flight

2018 Setup

NEDA



$$s(t) = A\left[e^{-t/\tau_1} - e^{-t/\tau_0} + R\left(e^{-t/\tau_2} - e^{-t/\tau_0}\right)\right] \text{ for } t > t_0$$

A: signal amplitude

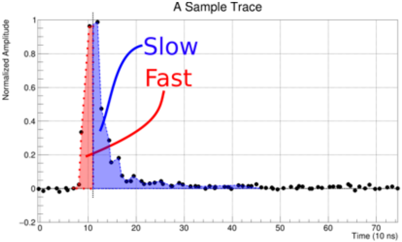
τ_0, τ_1, τ_2: decay constants (depends on the scintillator)

R: ratio of excited scintillation processes (different for γ and n)

t_0: signal alignement

2018 Setup

NEDA



$$s(t) = A\left[e^{-t/\tau_1} - e^{-t/\tau_0} + R\left(e^{-t/\tau_2} - e^{-t/\tau_0}\right)\right] \text{ for } t > t_0$$

A: signal amplitude

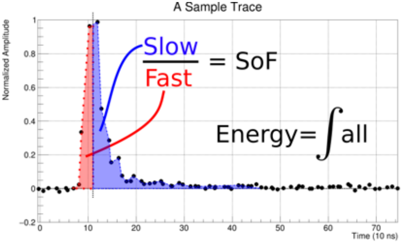
τ_0, τ_1, τ_2: decay constants (depends on the scintillator)

R: ratio of excited scintillation processes (different for γ and n)

t_0: signal alignement

2018 Setup

NEDA



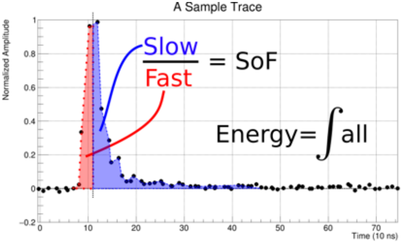
- Signal $\rightarrow$ SoF and Energy

$$s(t) = A\left[e^{-t/\tau_1} - e^{-t/\tau_0} + R\left(e^{-t/\tau_2} - e^{-t/\tau_0}\right)\right] \text{ for } t > t_0$$

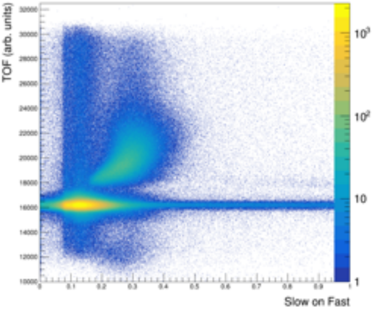
A: signal amplitude
τ_0, τ_1, τ_2: decay constants (depends on the scintillator)
R: ratio of excited scintillation processes (different for γ and n)
t_0: signal alignement

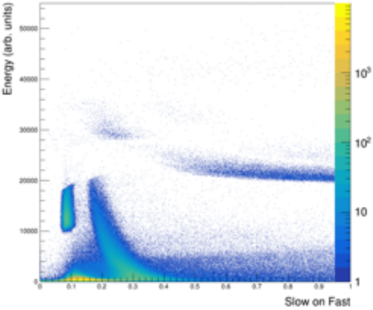
2018 Setup

NEDA



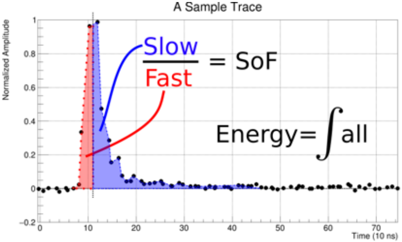
- Signal → SoF and Energy



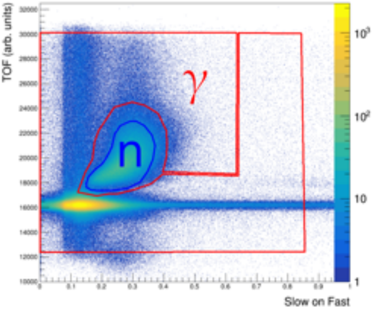


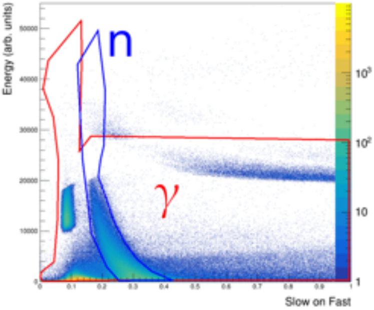
2018 Setup

NEDA



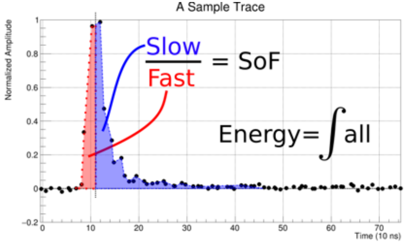
- Signal → SoF and Energy
- Classical charge-comparison algorithm: geometrical cuts = Our "Truth"



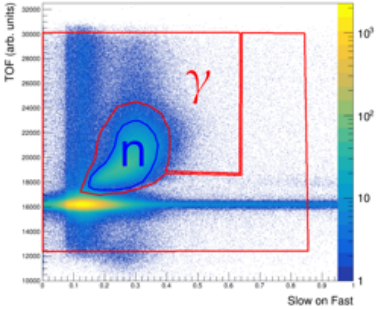


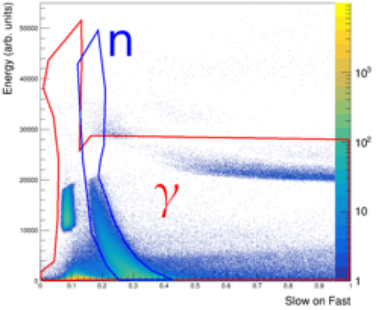
2018 Setup

NEDA



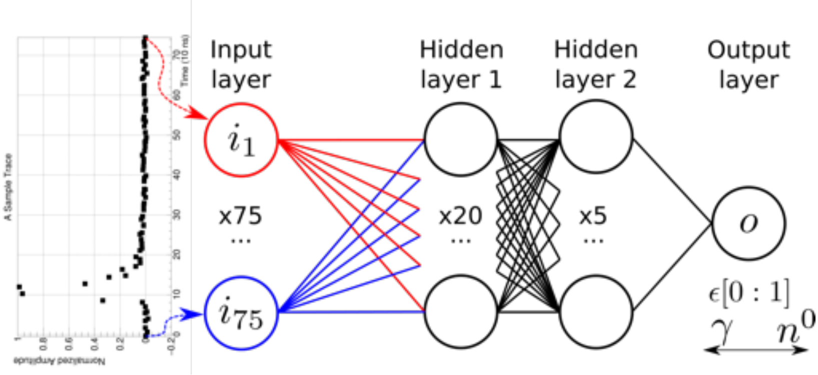
- Signal → SoF and Energy
- Classical charge-comparison algorithm: geometrical cuts = Our "Truth"
- Mislabel rate? Flexibility?





Previous collaboration work

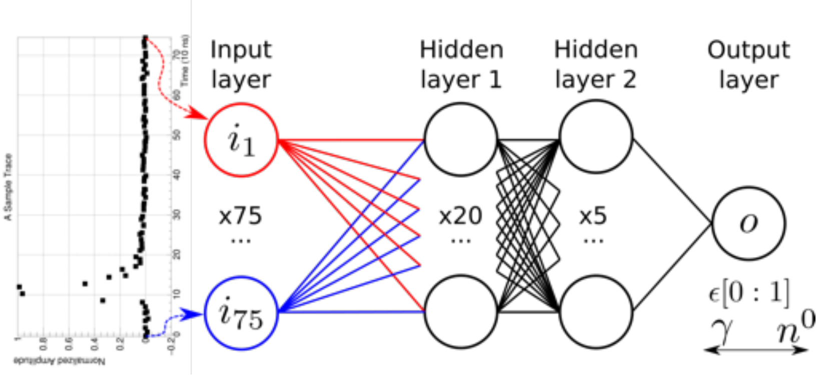
TMultiLayerPerceptron (ROOT)



Söderström *et al.* 2019. Neutron detection and γ-ray suppression using artificial neural networks with the liquid scintillators BC-501A and BC-537. NIM A. Volume 916:238-245. https://doi.org/10.1016/j.nima.2018.11.122

Previous collaboration work

TMultiLayerPerceptron (ROOT)

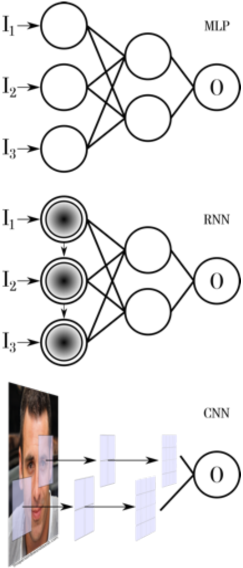


Söderström *et al.* 2019. Neutron detection and γ-ray suppression using artificial neural networks with the liquid scintillators BC-501A and BC-537. NIM A. Volume 916:238-245. https://doi.org/10.1016/j.nima.2018.11.122

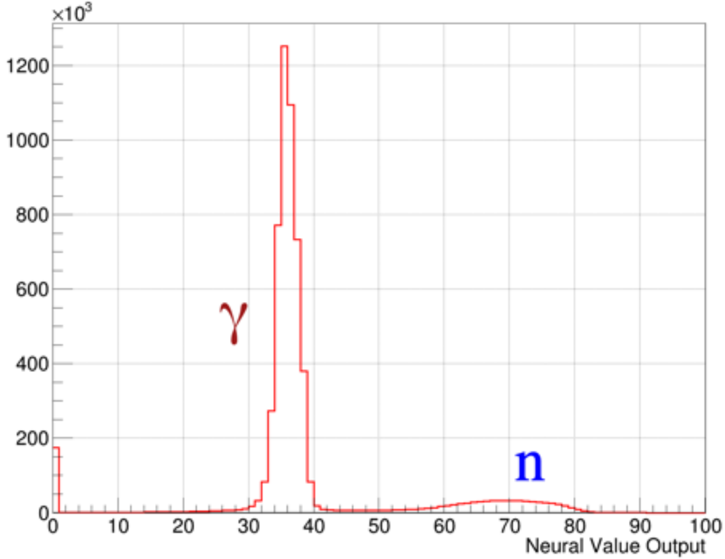
⇒ Interesting results, but online incompatible & what about other NN?

Investigated Neural Networks

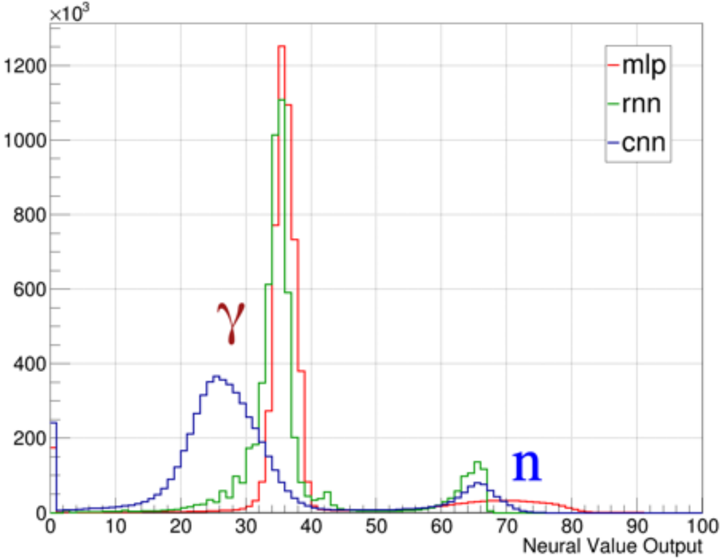
- Input layer = 75 neurons
  - First 73 signal samples
  - Energy
  - Time-of-flight
- Three architectures
  1. MLP: MultiLayer Perceptron
     The classical reference
  2. RNN: Recurrent Neural Network
     Ideal for time series
  3. CNN: Convolutionnal Neural Network
     Image recognition
- Output layer = 1 neuron
  - A value in [(γ)0; 100(n)]



Output

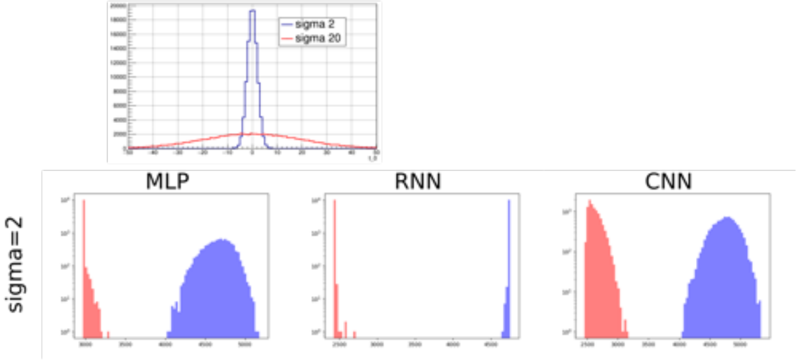

×10^3
1200
1000
800
600
400
200
0
0 10 20 30 40 50 60 70 80 90 100
Neural Value Output
γ
n

Output

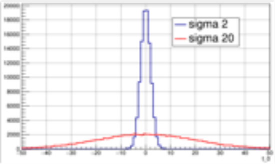


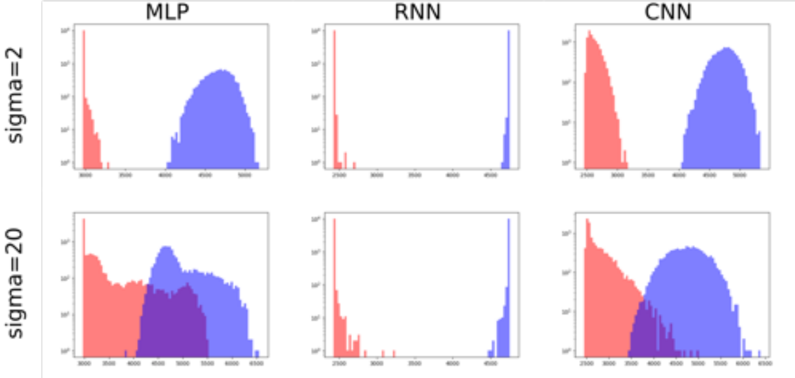
×10^3
1200
1000
800
600
400
200
0
0
10
20
30
40
50
60
70
80
90
100
Neural Value Output
mlp
rnn
cnn
γ
n

t_0



t_0

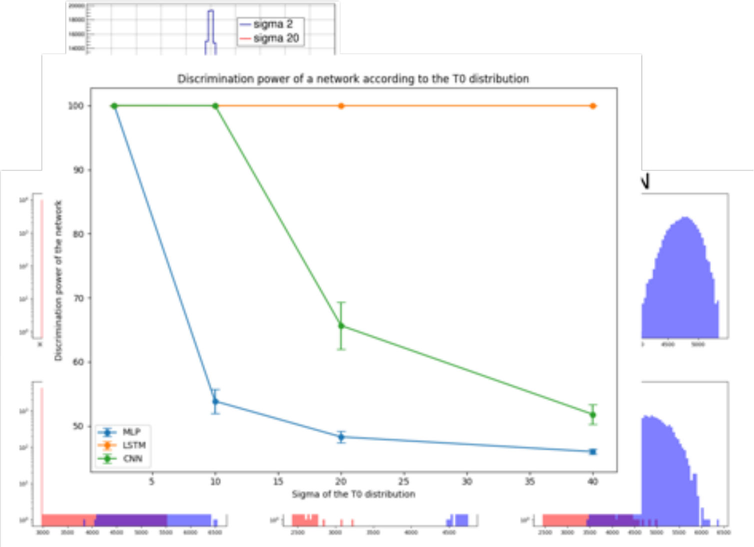




t_0 RNN resilient!

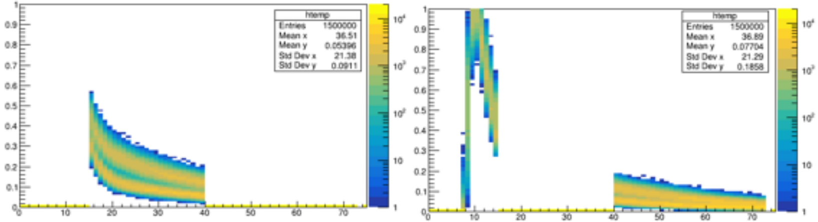
sigma=2

sigma=20



Truncated signal

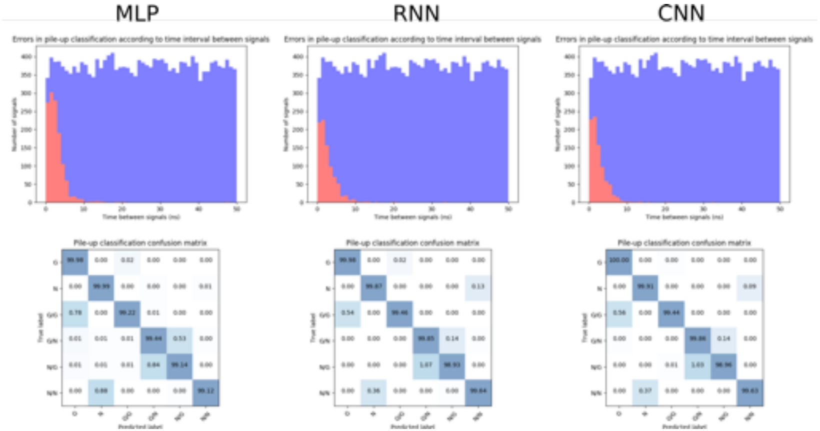
MLP can work with part of the signal:



. . . but signals need to be thoroughly pre-processed

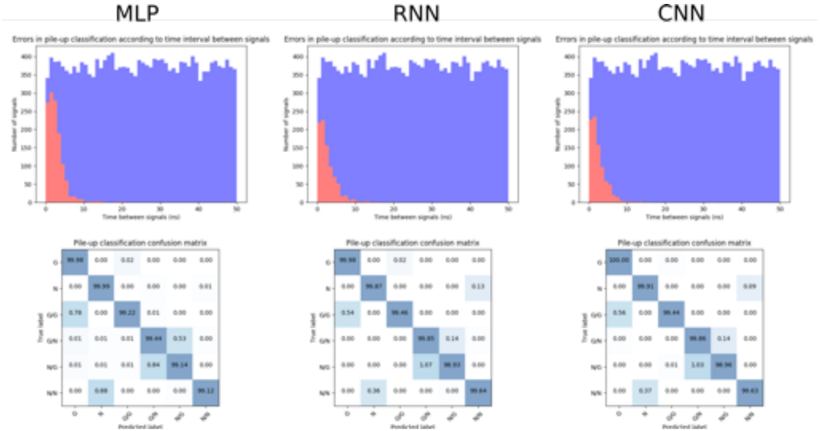
Pileup identification

- Done using simulated signals



Pileup identification

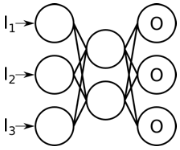
- Done using simulated signals

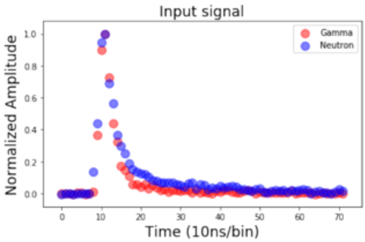


$\Rightarrow$ Works nicely, but signals need to be at least slightly separated

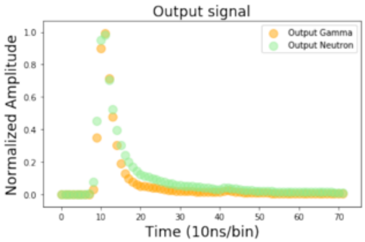
Related study: autoencoder

- Unsupervised learning
- Size of bottleneck? 4 required here.
  $\Rightarrow$ Linked with signal formula?
- Usages:
  - Noise suppression
  - Data compression





$\rightarrow$



Work of K. Zougagh

Other important considerations

The crucial step(s) of training

Inference time & Online compatibility

Practicability, Usage, Examples, . . .

Other important considerations

The crucial step(s) of training

Inference time & Online compatibility

Practicability, Usage, Examples, . . .

$\Rightarrow$ G.Baulieu's talk !

Investigated data

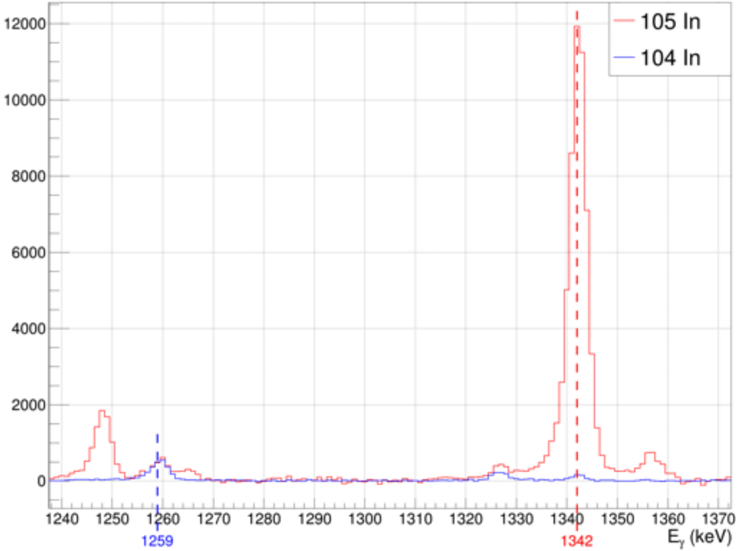
- AGATA NEDA DIAMANT 2018 campaign
- Experiment E703: ^{50}Cr $\rightarrow$ ^{58}Ni
- Runs 142+143 (~2×10^9 events):
  - Detectors stability checked
  - Time-aligned, Time gates active
  - DIAMANT: 0 α, 3 protons
  - NEDA: one event (most of the data & avoid combinatorics)
    Can be either a γ or a *n*, goal = test filter quality

Investigated data

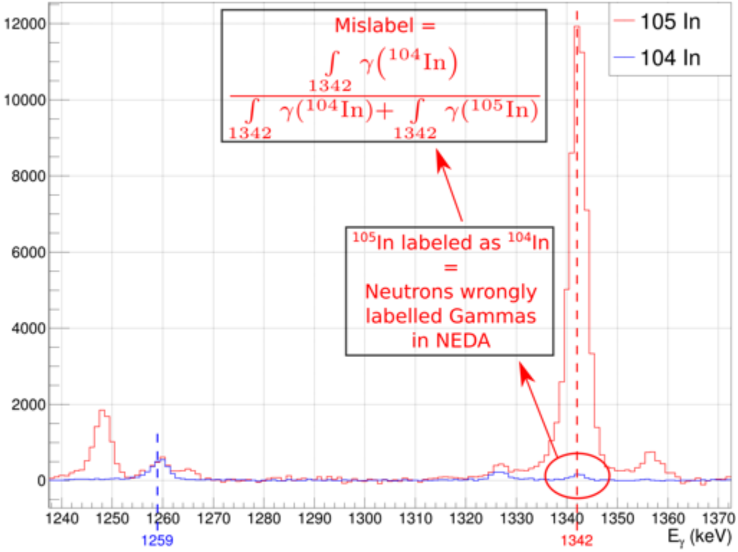
- AGATA NEDA DIAMANT 2018 campaign
- Experiment E703: $^{50}\mathrm{Cr} \rightarrow {}^{58}\mathrm{Ni}$
- Runs 142+143 (~2×10^9 events):
  - Detectors stability checked
  - Time-aligned, Time gates active
  - DIAMANT: 0 α, 3 protons
  - NEDA: one event (most of the data & avoid combinatorics)
    Can be either a γ or a n, goal = test filter quality

$\Rightarrow$ Compare ^{104}In and ^{105}In AGATA γ spectra to compute NEDA's neural networks n-γ discrimination quality

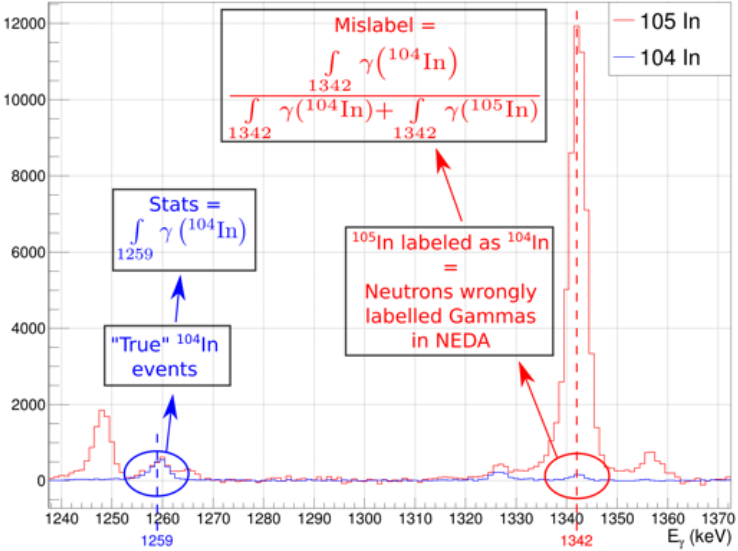
^{104}In vs ^{105}In — Geometrical cuts



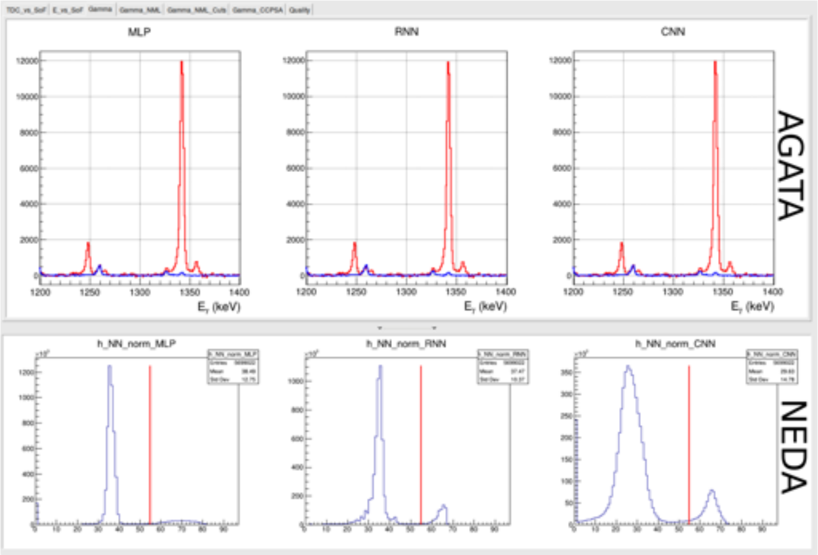
^{104}In vs ^{105}In — Geometrical cuts



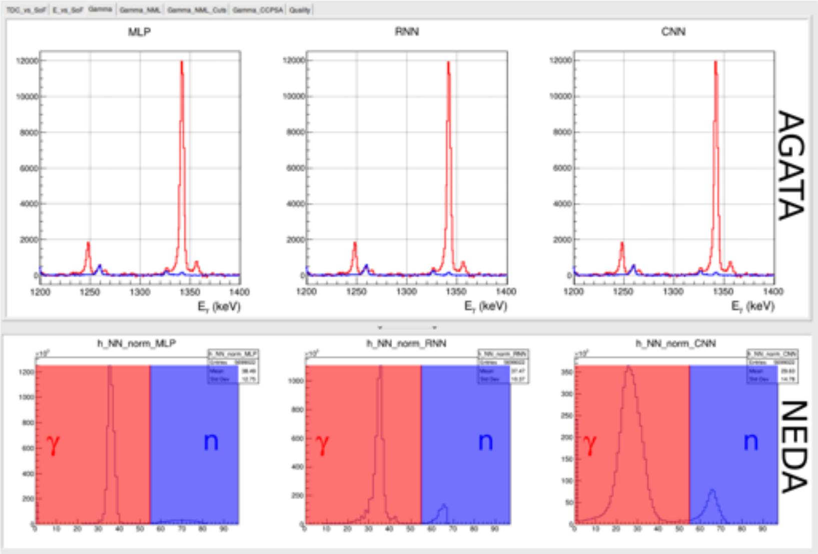
^{104}In vs ^{105}In — Geometrical cuts



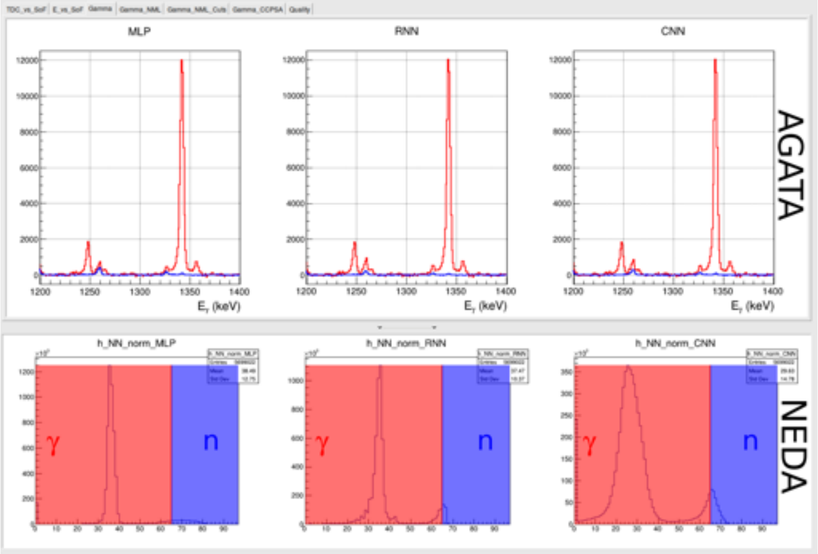
γ selection with NN



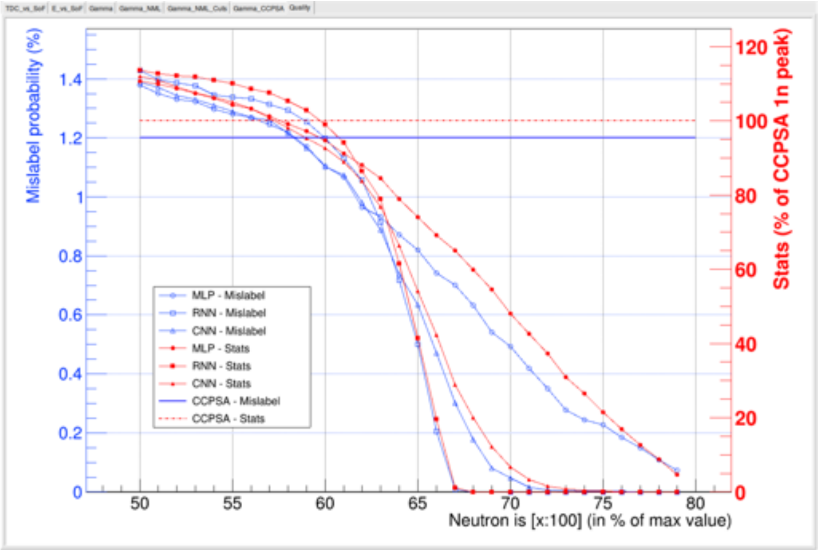
γ selection with NN



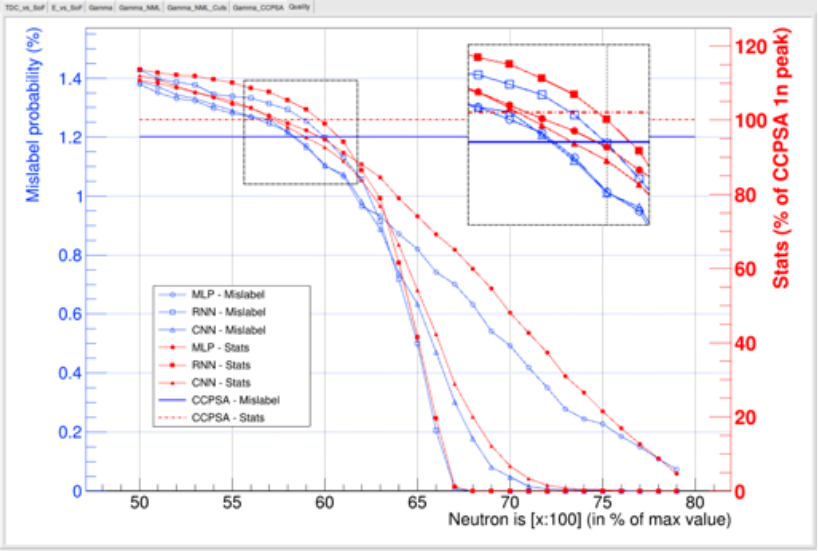
γ selection with NN



Quality vs Stats tradeoff



Quality vs Stats tradeoff

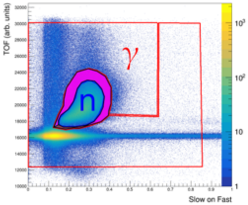


Wrapping-up

- Three Neural Networks (MLP, RNN, CNN) implemented
- Discriminating information from NEDA:
  - Sampled signal
  - Time-of-Flight
- Trained using truth label from classical CC algorithm (cuts)
- Multiple features were tested:
  - RNN is not sensitive to misaligned t_0
  - MLP only requires a part of the signal
  - Proper pileup identification
  - Autoencoder: 4 neurons required
- Mislabel probability vs stats of NEDA computed using AGATA
  $\Rightarrow$ Convergence towards training algo, user has flexibility
- Further results in Guillaume's talk!

Perspectives

- NN extrapolation skills
  $\Rightarrow$ Study of γ spectra associated to NEDA's "No Man's Land"
- Towards a variationnal autoencoder
- Future objective: apply developed skills to a more ambitious task
  $\Rightarrow$ AGATA signals



The End

Lyon IP2I task force:

- Guillaume Baulieu
- Laurent Ducroux
- Jérémie Dudouet
- Xavier Fabian
- Olivier Stézowski

Many thanks to all the people involved in the AGATA, NEDA and DIAMANT collaborations!

Questions?