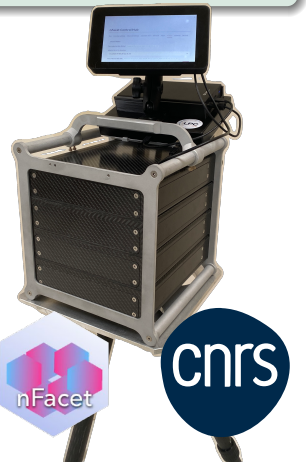
Machine learning techniques in the nFacet3D system

K. Béraud, G. Lehaut, A. Vacheret

nfacet@lpccaen.in2p3.fr

PhyNuBe4

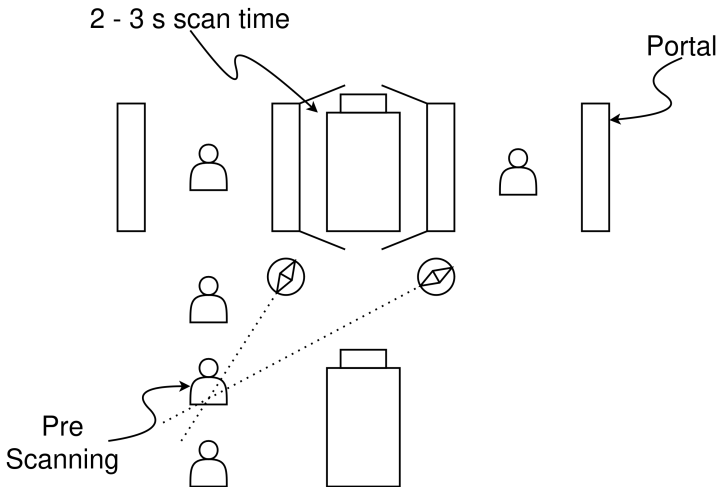


Context

Use case:

- Homeland security
- Radioprotection
- Radiotherapy

  → Localise and identify nuclear sources



Number of interactions:

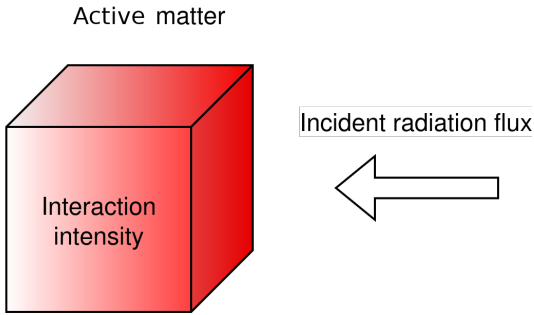
- $N(\vec{r}, E) = \sigma_{process}(\vec{r}, E)\, \phi(\vec{r}, \vec{\Omega}, E)$

Flux $\phi(\vec{r}, \vec{\Omega}, E)$:

- Position dependent
- Direction dependent
- Energy dependent

We wants:

- Gammas & Neutrons: $\vec{\Omega}$
- Neutrons: E

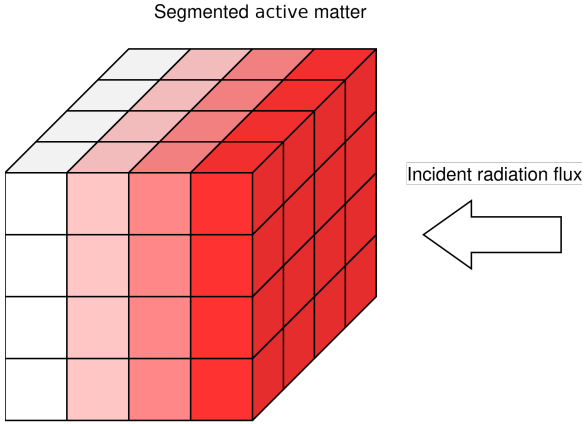


- Segmentation of the active volume

→ Active detection unit

Number of interactions per unit:

- $N_i(E) = \int N(\vec{r}, E) d\vec{r}$



nFacet

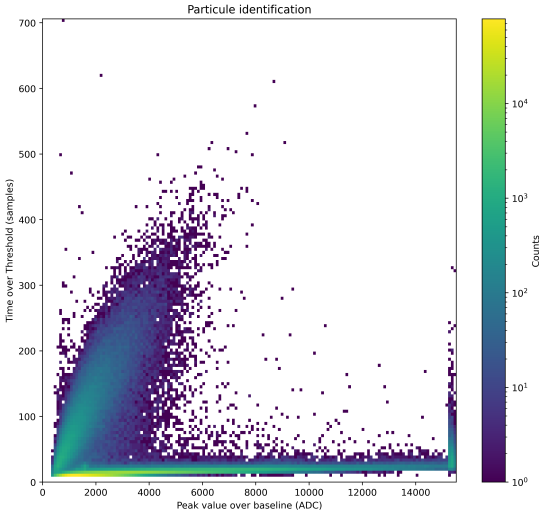
Detection unit

- 5 cm voxel
- Total unit: 64
- Layout: 4x4x4

Scintillators:

- PVT: neutron moderation and gamma detection
- ^{6}LiF:ZnS(Ag): neutron detection

That’s a lot of information!



Models

Generic approach

Tasks:

- Source direction: **DeepCompass**
- Neutron fluence: **nFluence**

Supervised approach:

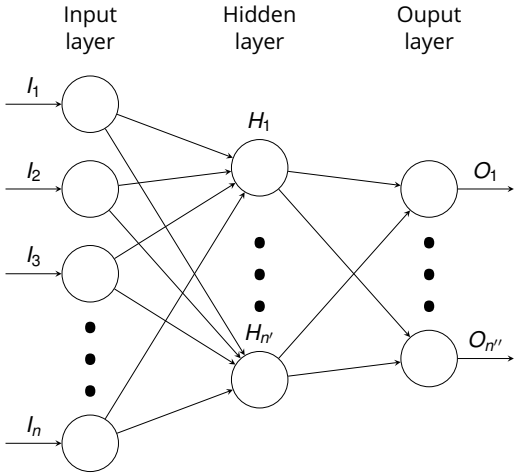
- Fully connected Neural Net

→ Light and fast

Input:

- Fraction of counts per cubes

→ Gammas or Neutrons



Models

Details

DeepCompass

- Model size: 33k
- Training dataset size: 4M

nFluence

- Model size: 120k
- Training dataset size: 20M

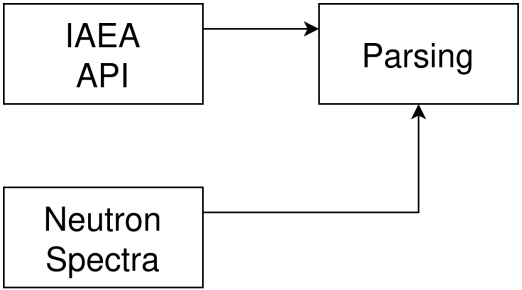
- Activation function: PReLU
- Regularisation: Monte-Carlo Dropout
- Trained on simulated data

Data Generation

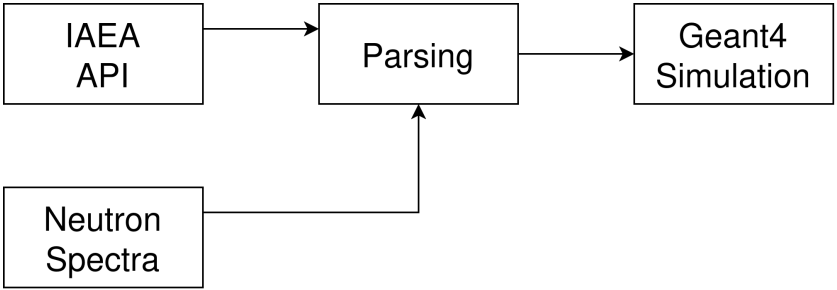
IAEA
API

Neutron
Spectra

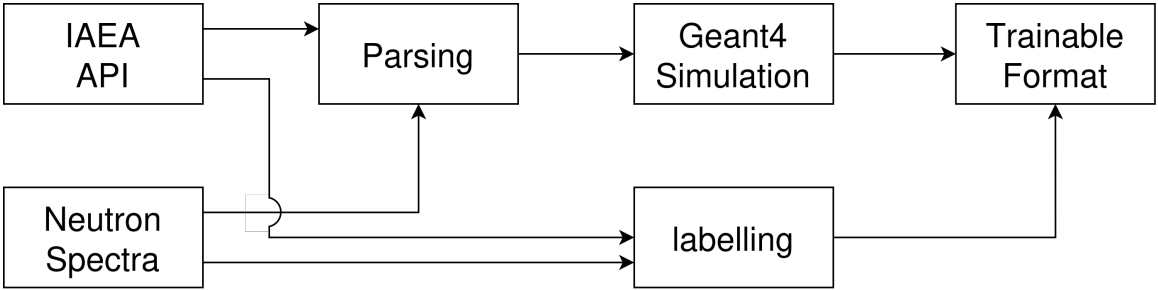
Data Generation



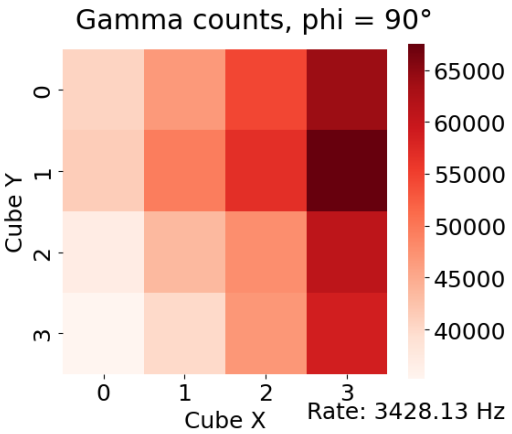
Data Generation



Data Generation



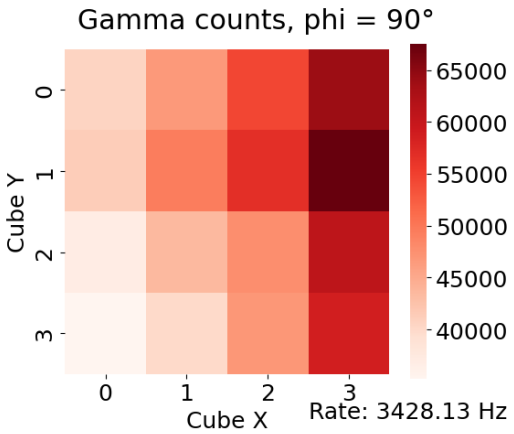
DeepCompass

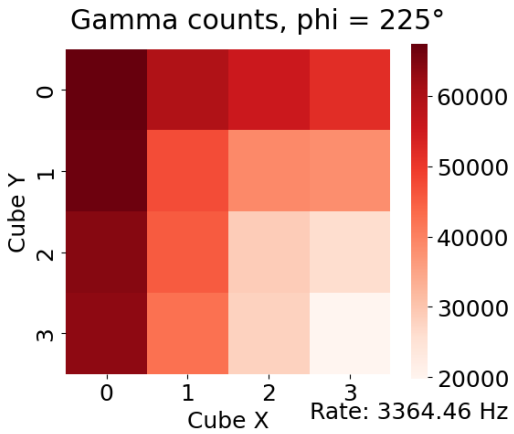


- Gamma or neutrons attenuation encode direction

→ Spatial distribution is direction dependant!

DeepCompass





DeepCompass

Output:

- Cartesian direction vector

→ Maximum error below 6° and mostly below 1°!

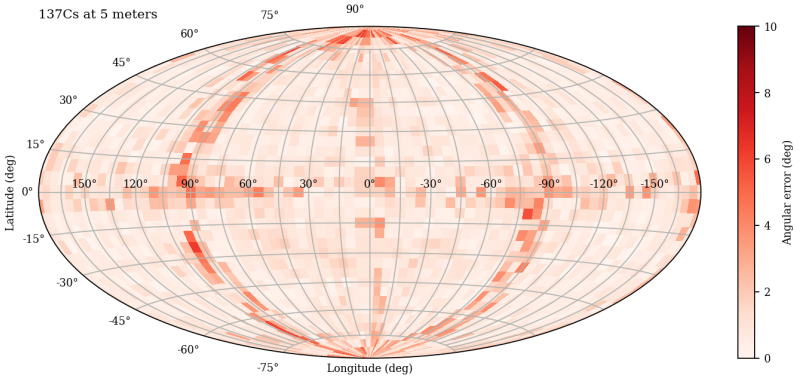
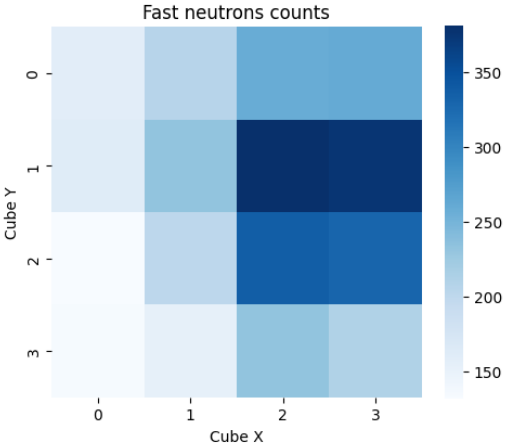


Figure: Angular error on direction prediction of a simulated ^{137}Cs source.

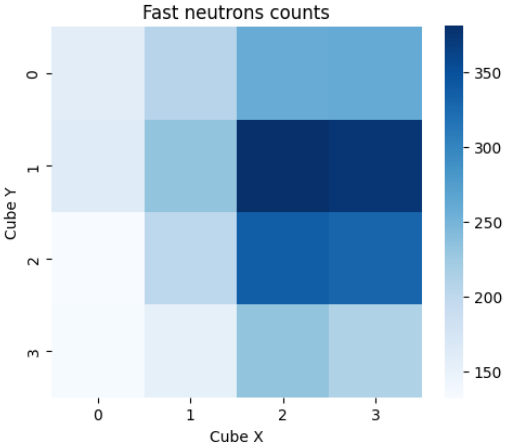
nFluence

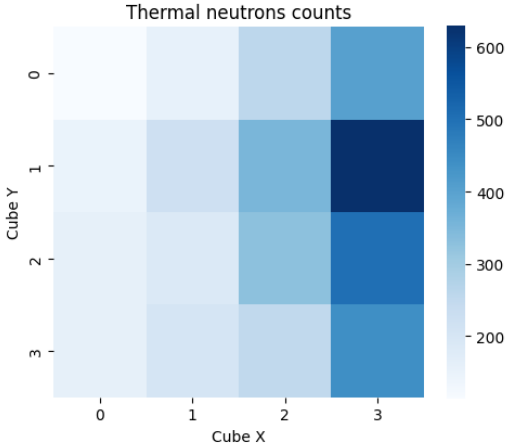


- Neutron signals are not energy correlated...

→ But spatial distribution is!

nFluence


Fast neutrons counts
Cube Y
0
1
2
3
Cube X
0
1
2
3
350
300
250
200
150


Thermal neutrons counts
Cube Y
0
1
2
3
Cube X
0
1
2
3
600
500
400
300
200

nFluence

Output:

- Binned fluence

→ Sensible to both fast and thermal neutrons at the same time!

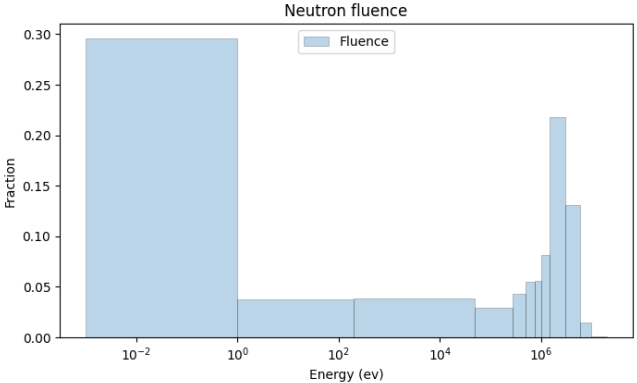


Figure: ^{240}Pu neutron fluence from MOX assembly.

nFluence

→ Sensible to moderator!

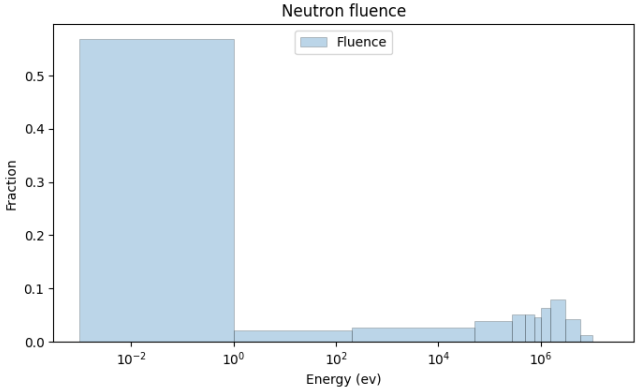


Figure: Thermalised ^{240}Pu neutron fluence from MOX assembly.

Summary

We use NN to exploit complex data:

- The main source direction is reconstructed
  - 6° worst resolution
  - Mainly 1° resolution
- The neutron fluence is reconstructed
  - Sensible to fast neutrons
  - Sensible to thermal neutrons
  - Sensible to shielding

→ **Neural Nets are effective methods for difficult inverse problems!**
Model error can now be quantified.

Thanks for your attention

Questions? Comments?