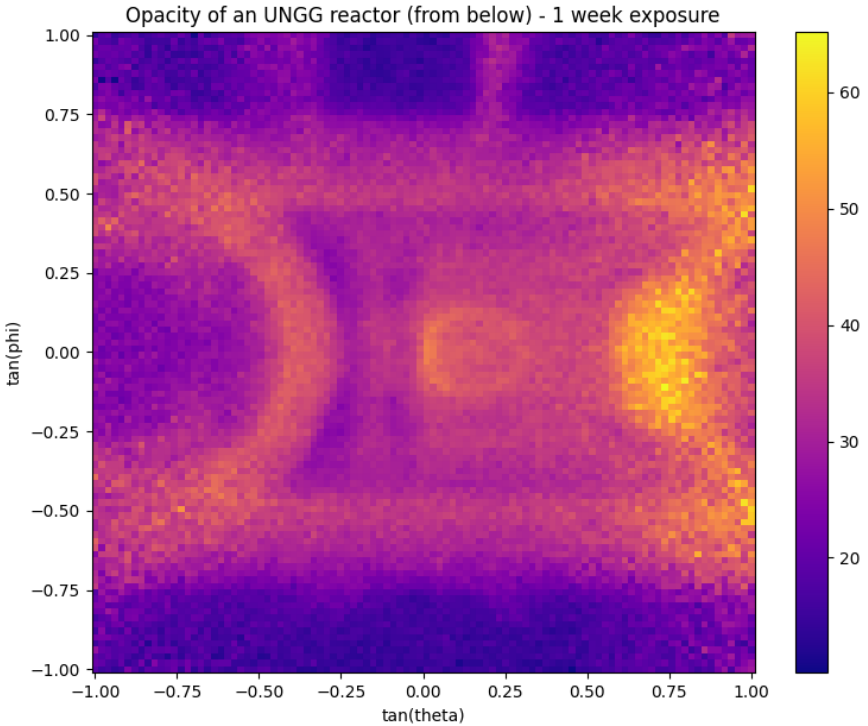
Irfu
Institut de recherche
sur les lois fondamentales
de l'Univers

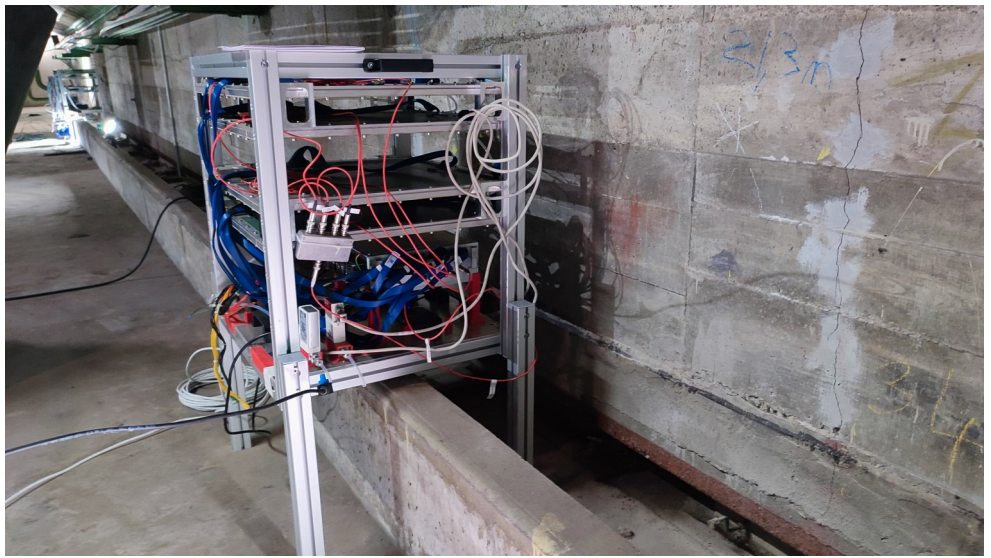
Convolutional neural networks demultiplexing in large Micromegas detectors for muography purposes

IN2P3/IRFU Machine Learning workshop – September 2022

Baptiste Lefevre, baptiste.lefevre@cea.fr

Muography project at Irfu

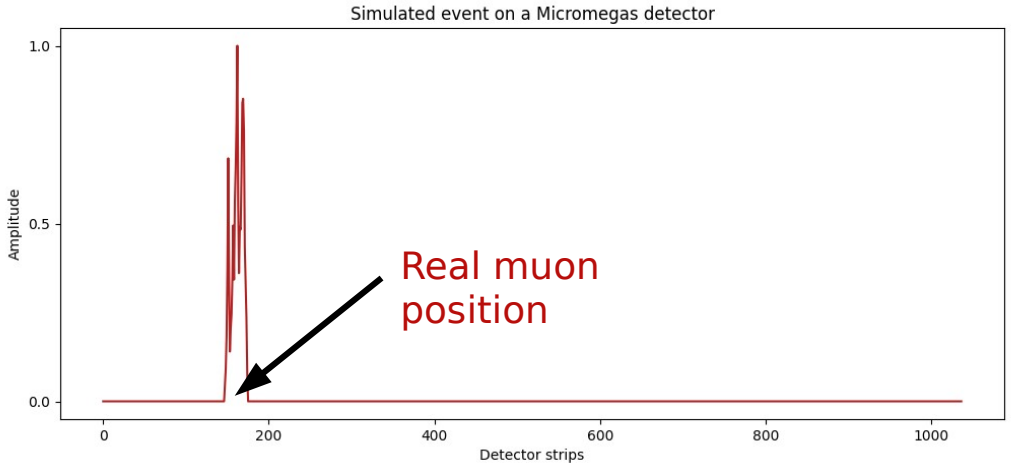


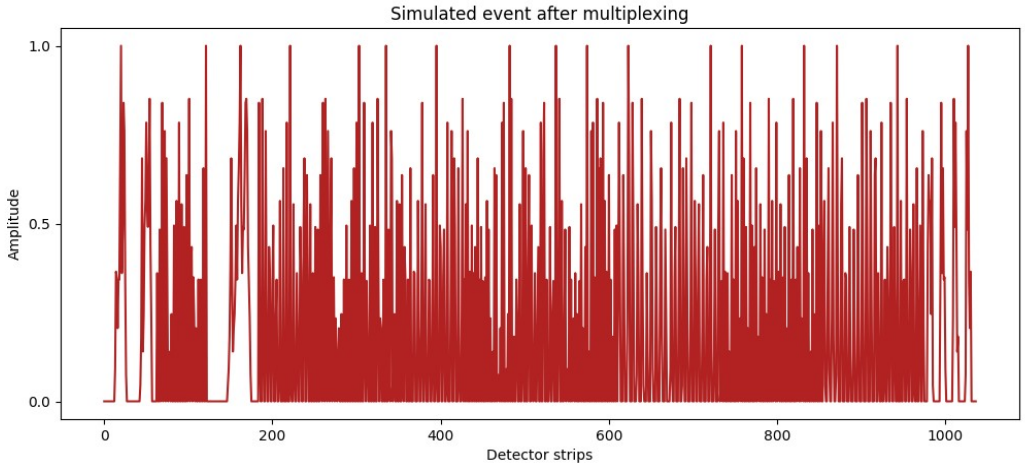


Picture of a muon telescope with 4 detectors below the same reactor

Detector multiplexing

Muon detection done with Micromegas gazeous detectors [1]





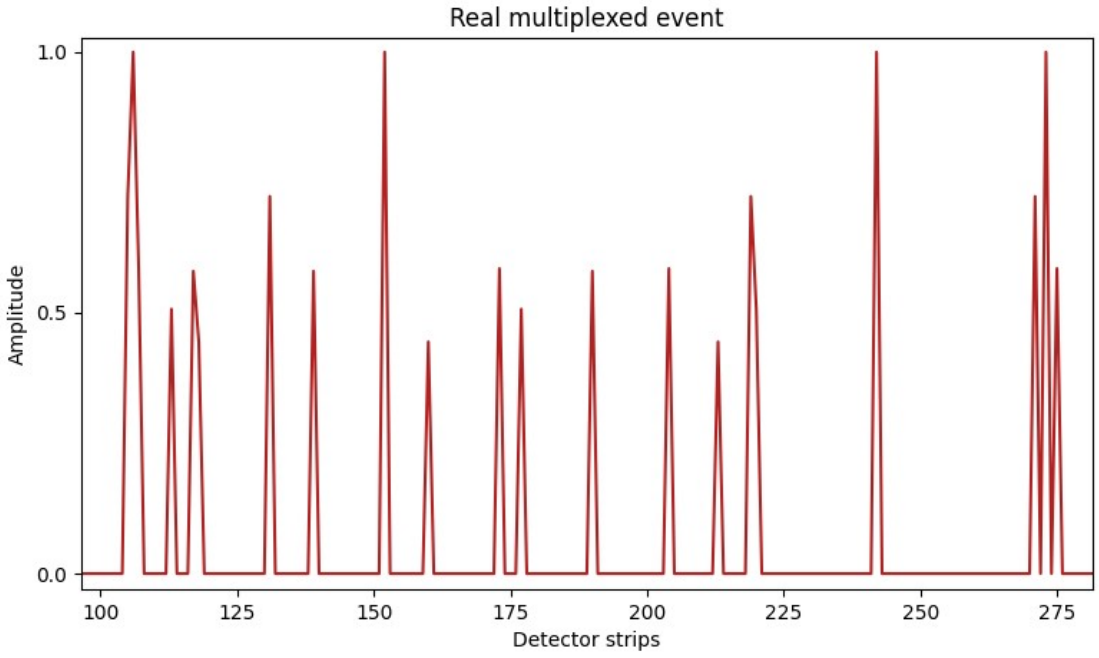
1037 strips on the detector
1037 amplitudes shall be read

Multiplexing
=
Sum 17 strips to 1 channel

61 channels
Only 61 amplitudes to read

[1] S. Procureur et al. Genetic multiplexing and first results with a 50 × 50 cm2 Micromegas, Nucl. Instrum. Meth. A 729 (2013) 888

Current demultiplexing method



- **Find connected areas with high amplitudes**
- **Tolerate some holes**
- **Look on other detectors for coherent track between them**

→ Many empirical parameters

→ Many steps and conditions…

→ Can be improved

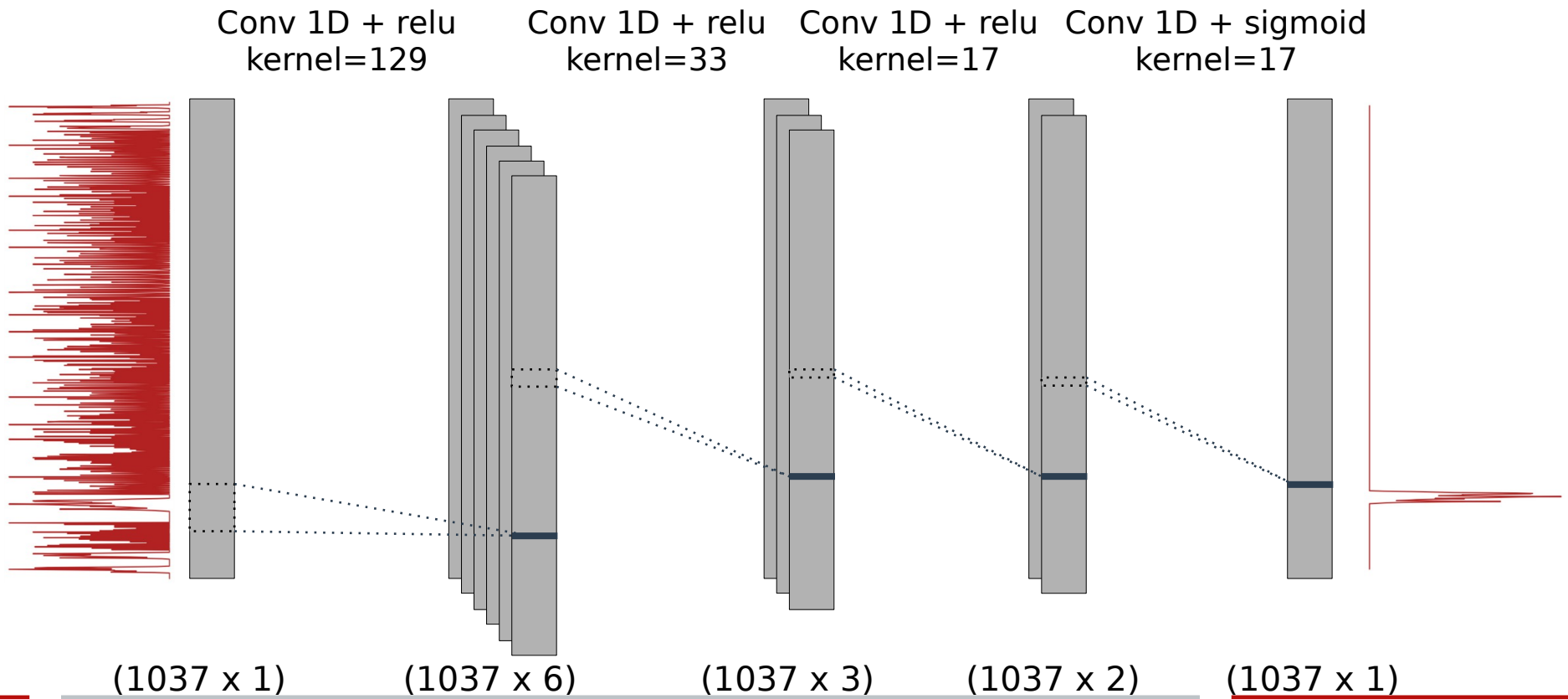
CNN Demultiplexing architecture

1-dimensional Convolution Layers [2]

- Very few parameters (easily applied to large input vectors)
- Strides to 1 and same padding → Keep input vector size
- Keep analysis "local" in the data → Could get multiple events
- Easier to interpret → May be seen as a simple filtering

[2] Keras API, https://keras.io/api/layers/convolution_layers/convolution1d/

CNN Demultiplexing architecture



Training data

→ **Real demultiplexed data are biased by the current algorithm**

→ **Training will be made on simulations**

→ Simulations are produced by Sébastien Procureur

→ **Model is tested afterwards on real data after demultiplexing**

Custom constraints

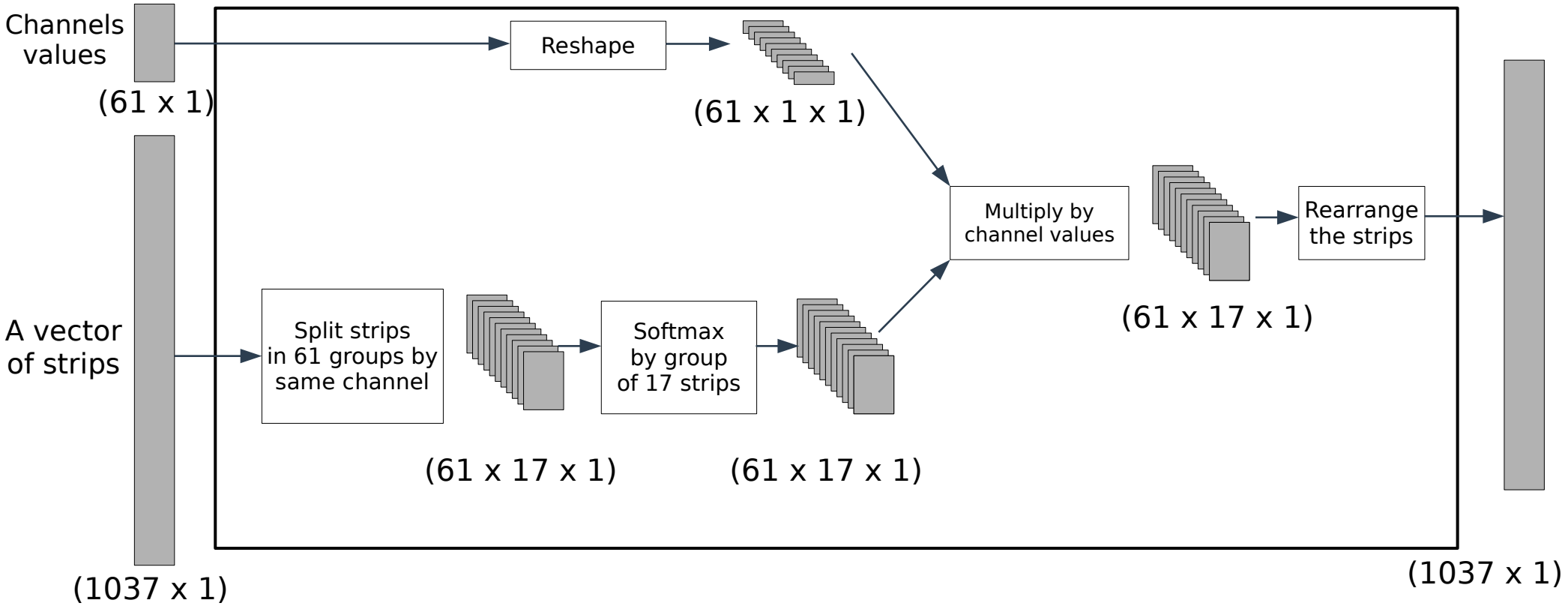
- **A custom layer [3] to force to respect the multiplexing pattern**

  The sum of 17 strips from the same channel is known

  "Demultiplexing normalization"

[3] With Tensroflow Layer API, https://www.tensorflow.org/api_docs/python/tf/keras/layers/Layer

Custom constraints
Demultiplexing normalization



Training results

Optimizer :

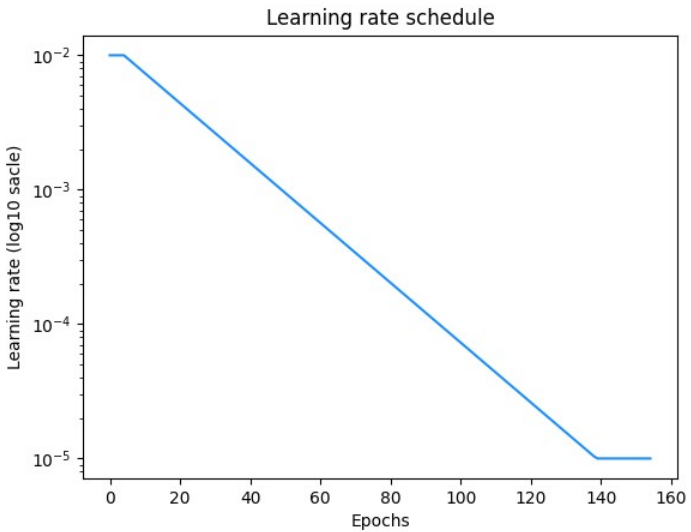
Adam [4]

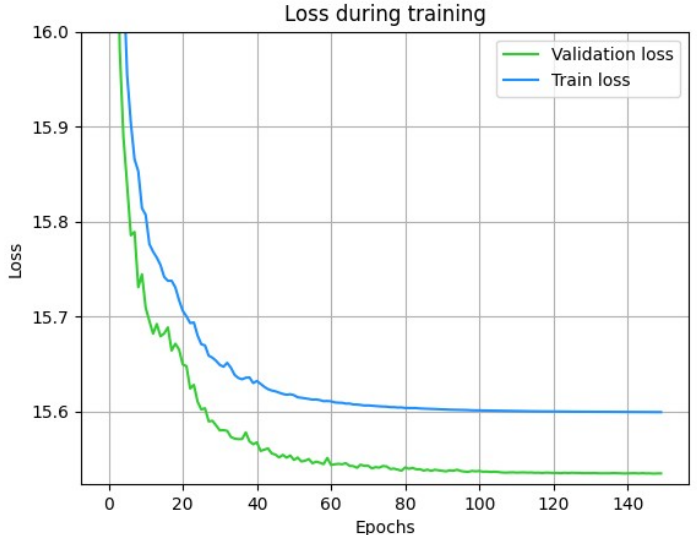
Learning rate scheduled from 10^{-2} to 10^{-5}

Epochs :

Limited by early stopping on the validation loss

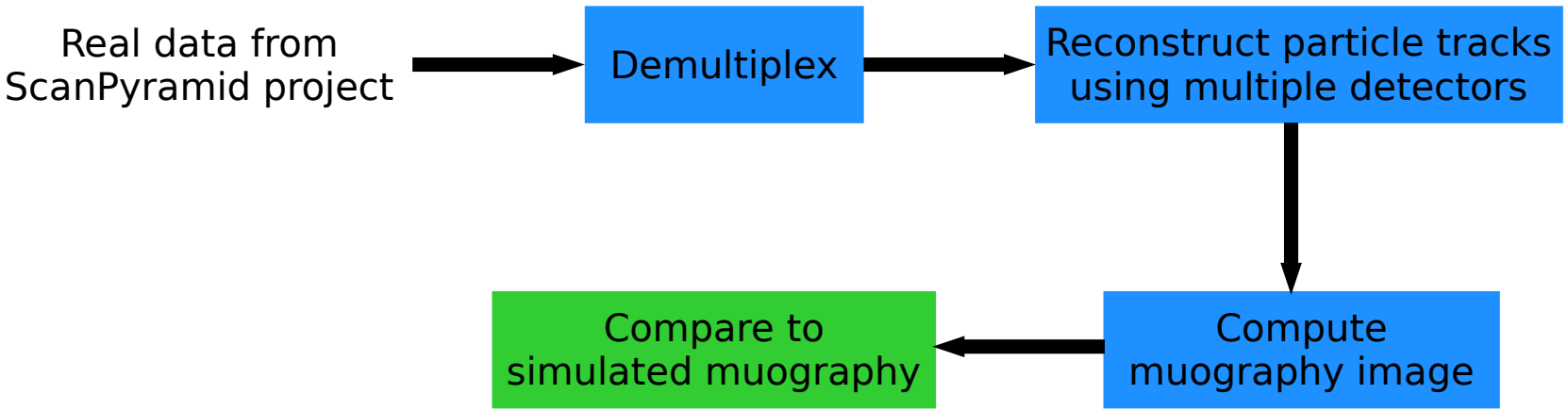
Typically takes between 60 and 150 epochs



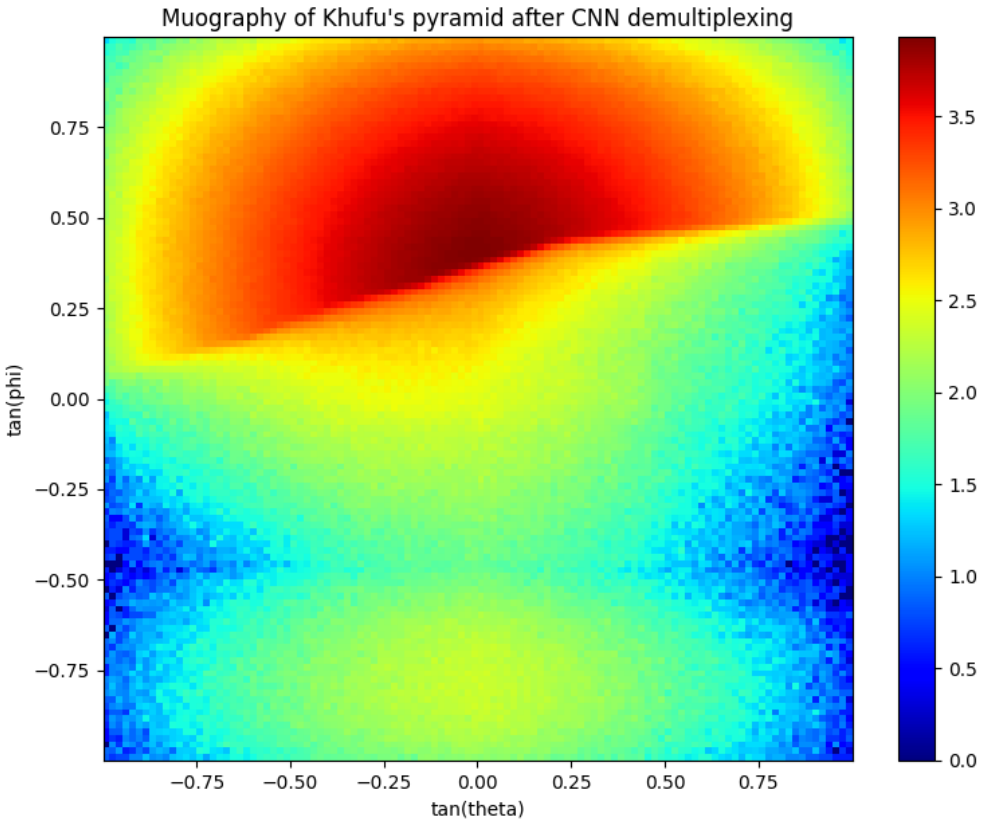


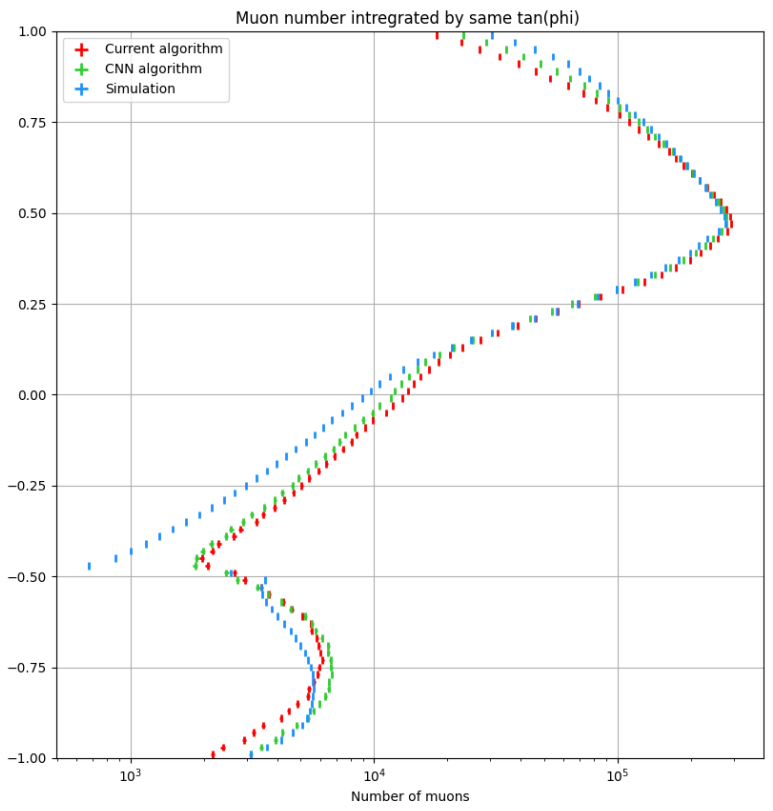
[4] Implemented in Keras API, https://keras.io/api/optimizers/adam/

Results on real data

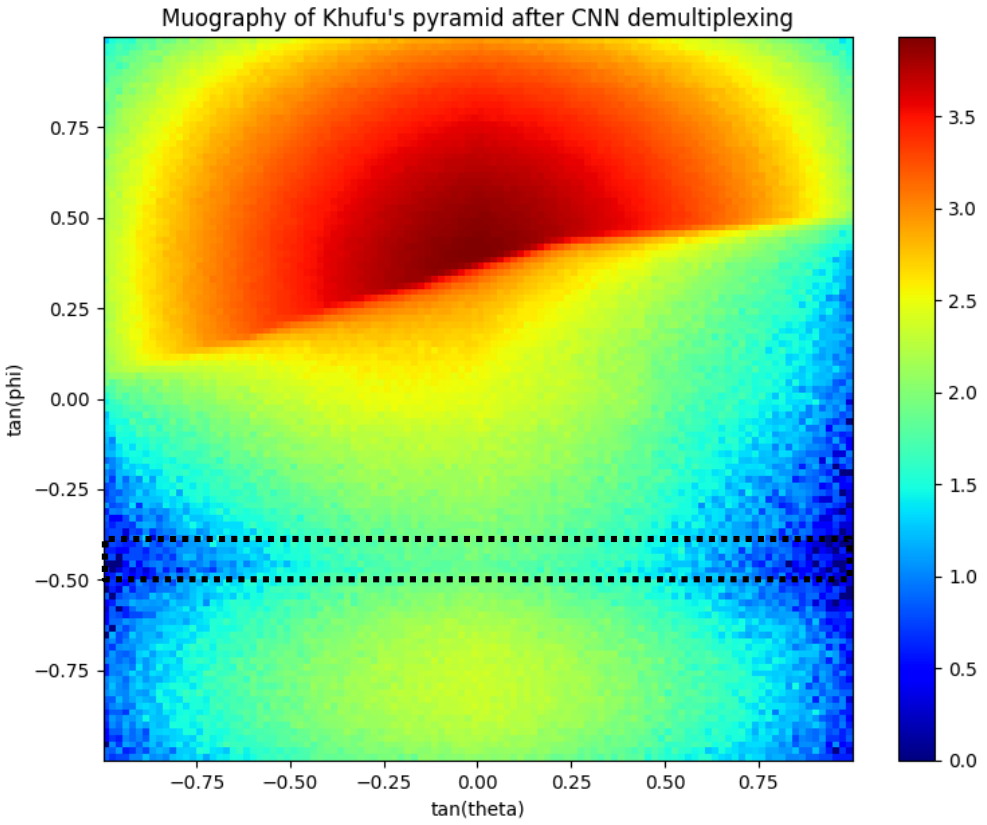


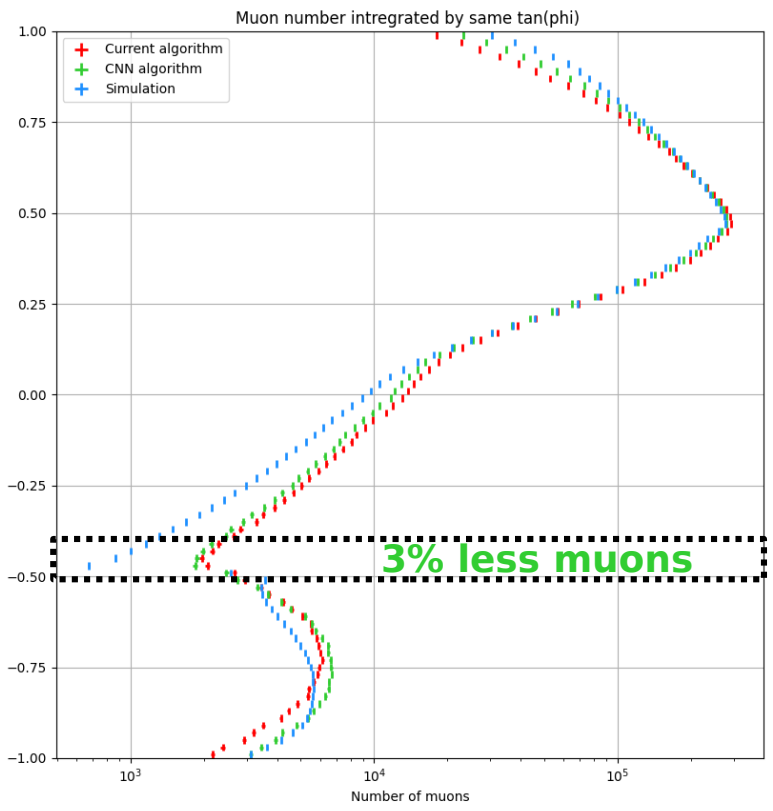
Results on real data



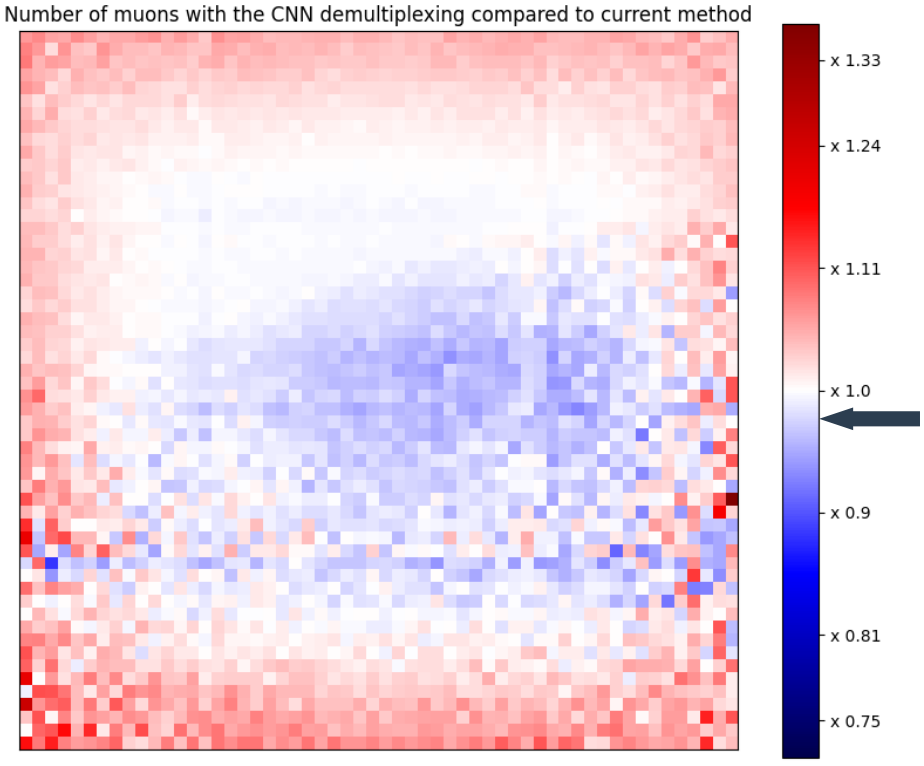


Results on real data





Results on real data



- More muons at large angles
  → on the borders of the muography
- Less muons through the pyramid
  → pyramid shape recognizable

Issues

Demultiplexing normalization :

- Makes it more difficult to de-noise
- Has not proven to be effective on real data

Training is only done on simulated data, we would need either to

- Increase simulations quality
- Build a non-multiplexed detector to acquire real data
- Find a way to use real data during training

How to use real data during training ?

- Only used after training+analysis to compare to expected muon angle distributions

Conclusion

- **Convolutional NN have proven to be efficient for this application**
- **Work remain to :**
  - Find a way to use real data
  - Use constraints from the demultiplexing
  - Characterize the performances of this solution

Baptiste Lefevre, baptiste.lefevre@cea.fr