

# ML for Data selection

**M.Nebot** on behalf of Lewis Lappin, Marin Mlinarevic, F. Muheim

DUNE DS/PP WG meeting  
2nd March 2021



## Motivation (MPhys projects)

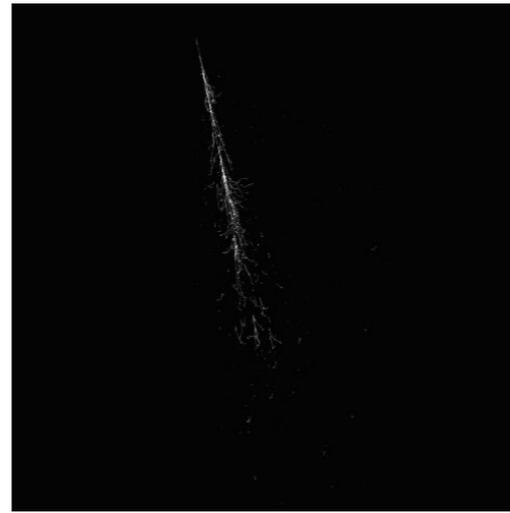
- ML based R&D plans for Data Selection interested some undergrad students.

*“For DS you either go around checking different possible TCs and how to ensure, which threshold, you use to form a trigger decision or you use a few students/life to evaluate all possibilities == train a NN to do it.”*

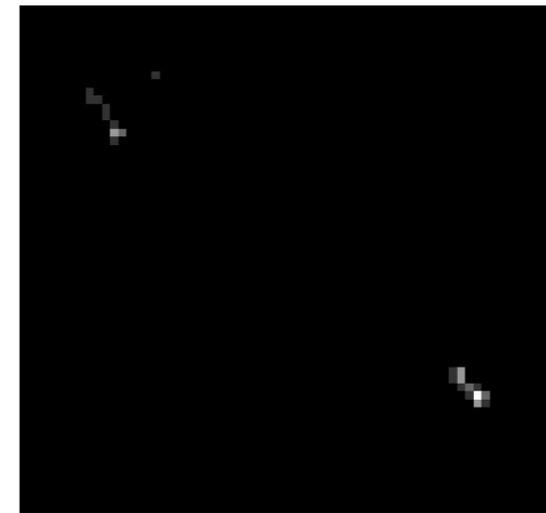
- This was setup as preliminary studies (MPhys projects), unfortunately COVID-19 brought it to an abrupt end (hopefully just a pause).
- We would like to continue this studies trying to use TC to NN. With possible outputs as: TDecision, event classification (HLF), ROI (#APA, time window ...), early pointing resolution ...

# Event classification using deep learning

- 20k Marley + gaushit generated SN + 20k radiological events into 2D hits histograms (1500x1500 pixels)



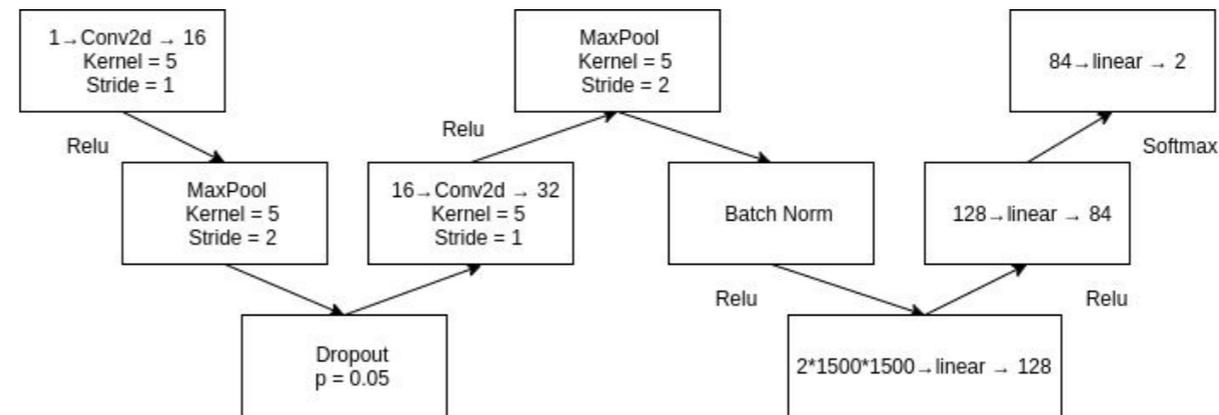
High Energy Electron



Supernova Neutrino

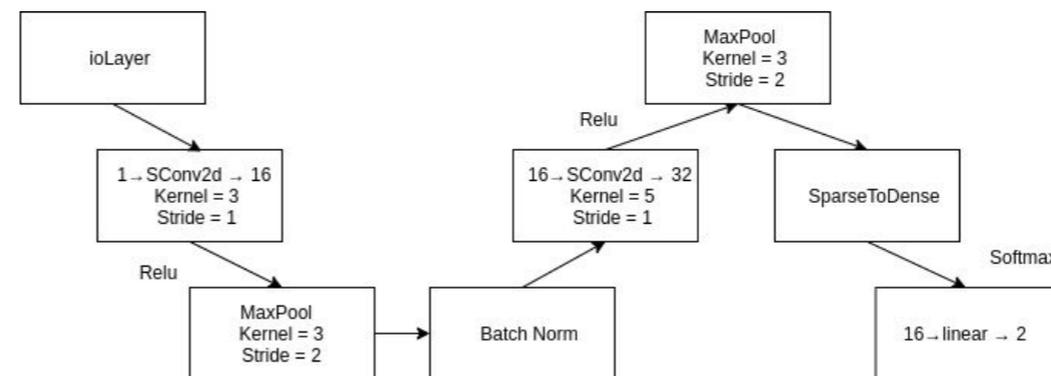
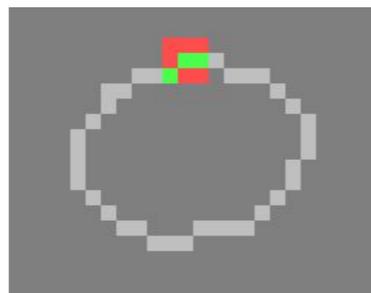
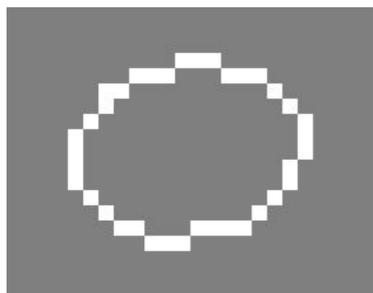


Background



PyTorch CNN  
Batch size 128

Intel® Xeon® Gold 5120 Processor.  
14 cores and 196 GB of RAM.



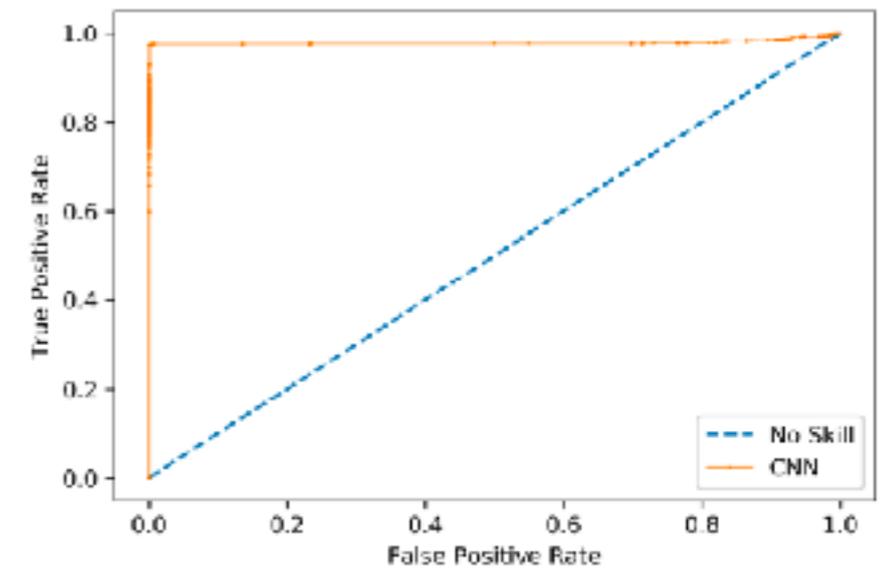
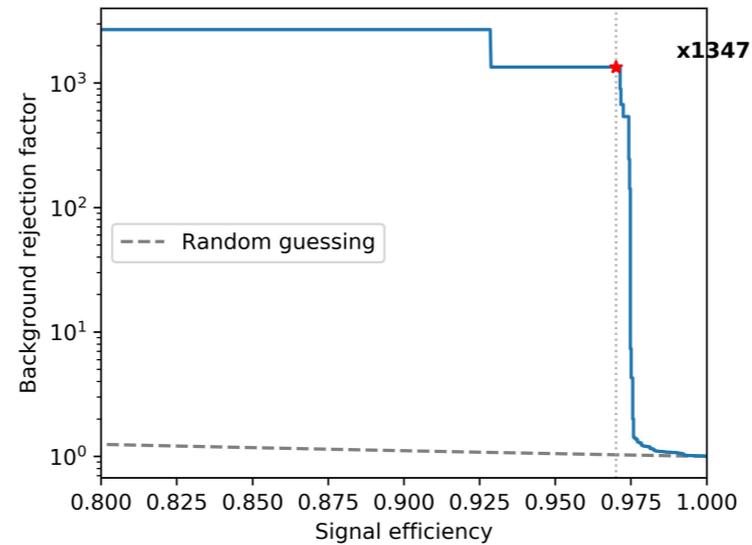
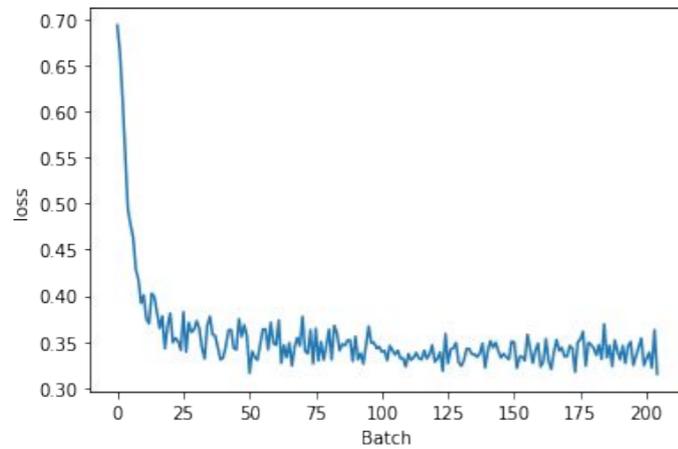
Facebook's SparseConvNet

Credit:  
<https://github.com/facebookresearch/SparseConvNet>

# Event classification using deep learning

## Results - CNN

98% Total accuracy  
 98% Signal  
 97% Background



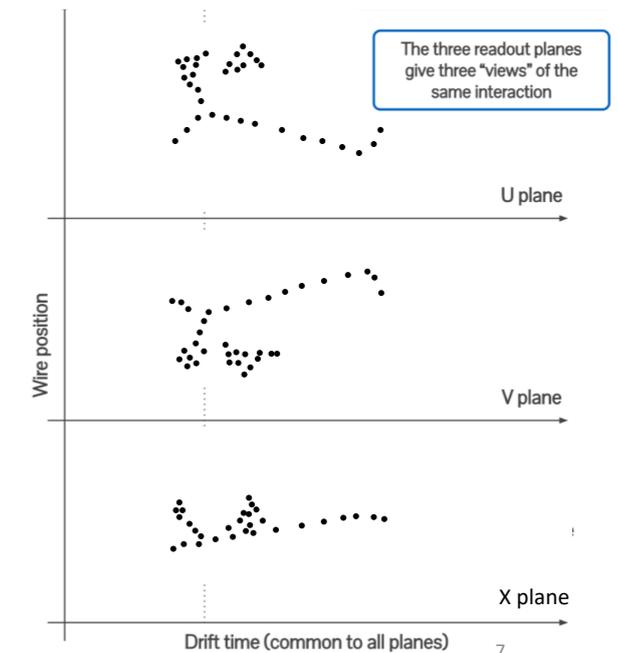
16

Network	Computation Time (s)	Peformance increase
CNN	$187 \pm 24$	1
Sparse CNN	$0.025 \pm 0.005$	7480

# Neural networks for event reconstruction

- Graph network for pattern recognition

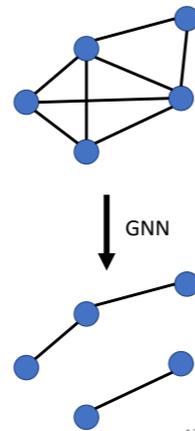
- Uses hits as inputs (wire position, drift time, width, amplitude, collected charge)
- Cluster hits coming from same particle together – wire position and drift time give 2D image (“view”)
- Identify interaction vertices
- Identify hierarchies (parent-daughter links)
- Match 2D clusters into 3D trajectories



A. Smith, The Pandora multi-algorithm approach to pattern recognition (slides DESY, 16/9/2019)

## Graph neural networks (GNNs)

- Graph is a set of nodes with some features, with edges describing relationships between them
  - Natural representation of LArTPC data:
    - Nodes = hits
    - Node features = hit wire, hit time
    - Edges connect hits into clusters/tracks
    - Edge features: distance between hits (in wire and time)
- Variable input size (number of hits in event)
- Efficient representation of sparse data
- Can classify nodes, edges or whole graph
- My GNN: classify edges as correct or wrong connections



12

## Edge-classifying GNN architecture



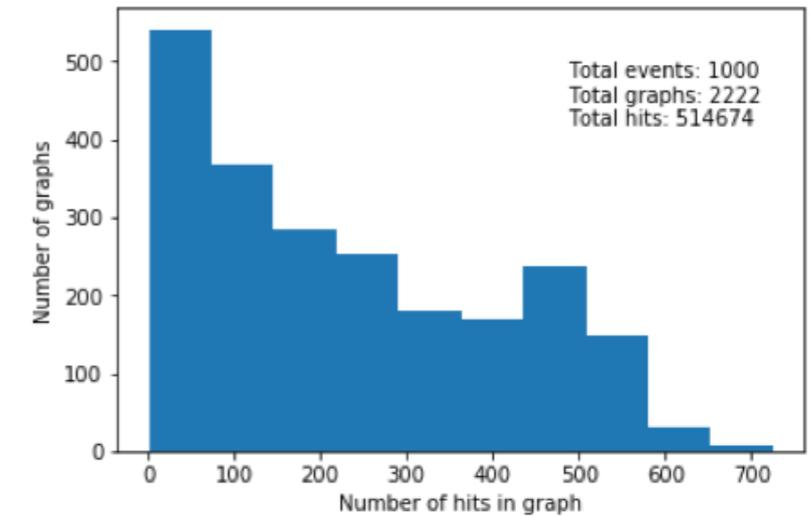
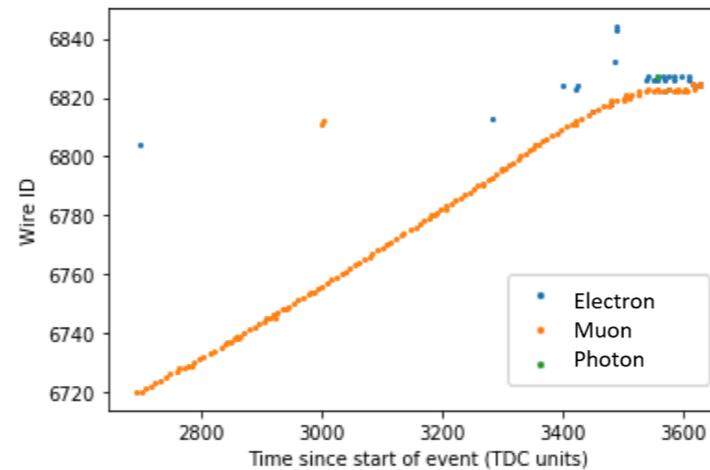
- 2 multi-layer perceptron components:
  - Edge network computes weights for every edge using the features of the two nodes it connects
  - Node network computes new features for every node using aggregated features of edges
  - Each has 2 hidden layers with 128 nodes and ReLU activation
- 8 iterations; each propagates features from each node further through graph
- Sigmoid activation on final layer
- Implemented using DeepMind’s graph\_nets

13

# Neural networks for event reconstruction

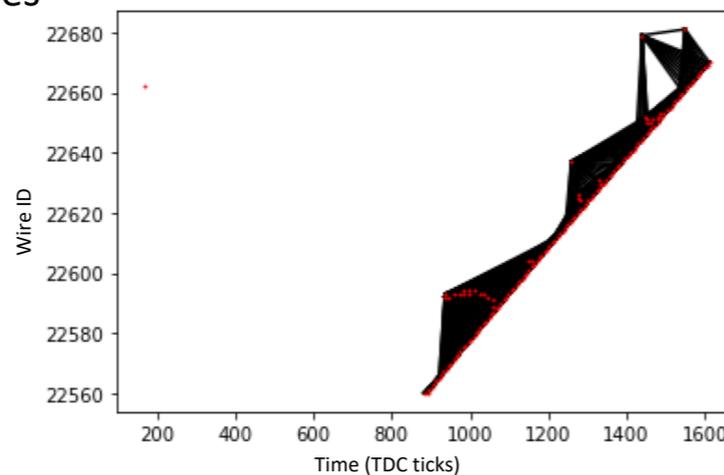
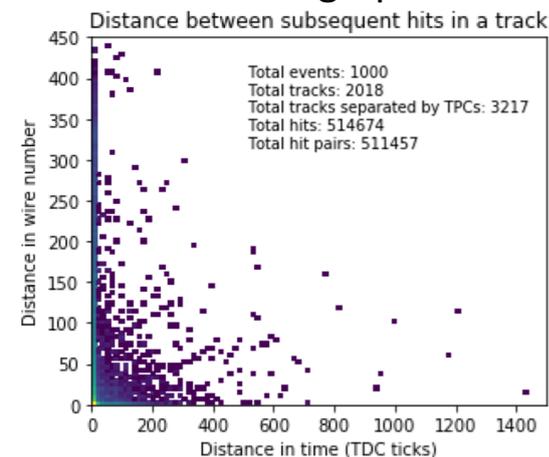
## Simulation

- LArSoft
  - Monte Carlo generator
  - Particle propagation and interactions in LAr
  - Detector effects
- Detector:
  - Geometry:  $1 \times 2 \times 6$  APAs
  - 24 frames with 480 collection wires each
- Generated 1000 isotropic muons with energy 0.1 GeV – 5 GeV



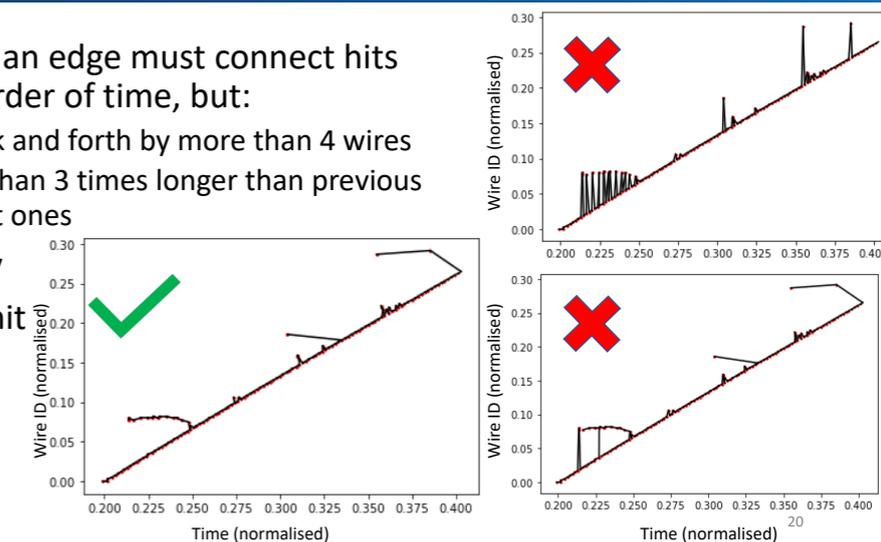
## Graph construction

- Separate graph for each frame, only collection wires
- Discarded graphs with  $< 4$  nodes



## Target definition

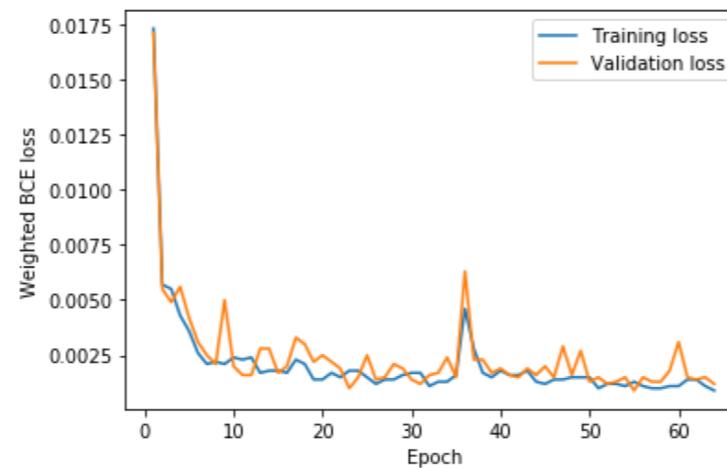
- To be labelled 1, an edge must connect hits within track in order of time, but:
  - Can't jump back and forth by more than 4 wires
  - Can't be more than 3 times longer than previous and subsequent ones
- Each hit can only connect to one hit further in time
- Else label is 0



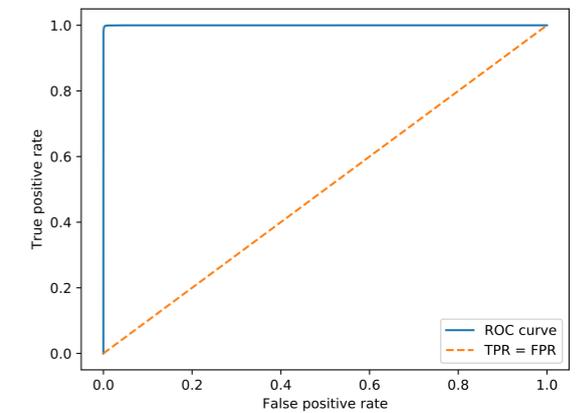
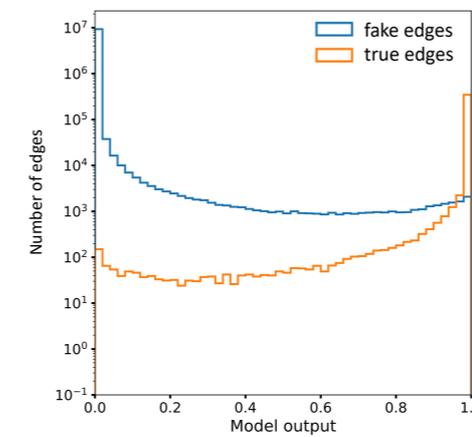
# Neural networks for event reconstruction

## GNN training

- 2147 graphs split into 3 sets:
  - Training: 56%
  - Validation: 14%
  - Testing: 30%
- Train target:
  - 360 000 true edges
  - 9.5 million fake edges
  - True/fake ratio: 0.04
- Loss: binary cross entropy, with weight of 0.04 for false edges
- Batches of 9
- 65 epochs

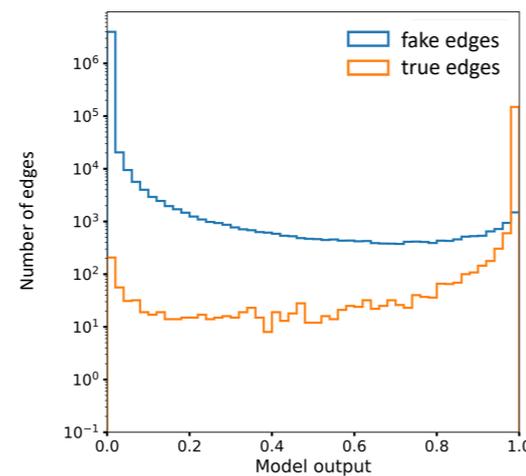


- Best cut on model output: 0.47 (TPR = 99.7%, FPR = 0.3%)



## GNN results – test data

- Accuracy with discrimination threshold of 0.47: 99.7%



Confusion matrix (normalised by true label):

True label	True edge	0.4%	99.6%
	False edge	99.7%	0.3%
		False edge	True edge

Predicted label

23

## Summary

- Two studies of the implementation of NN for event classification and reconstruction performed.
- The comparison of SparseNN and CNN was not fully completed but laid out a starting point.
- Successful implementation of a GraphNN, larger data set would be needed for better performance studies.
- Although results shown are encouraging, there's still lots to do. We'll take over with improvements (apart from fixing things):  
Metric comparison, Trigger candidates ...