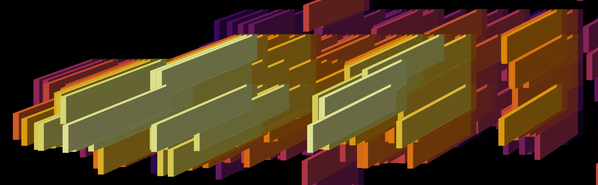
Neural network clusterization for the ALICE TPC online processing

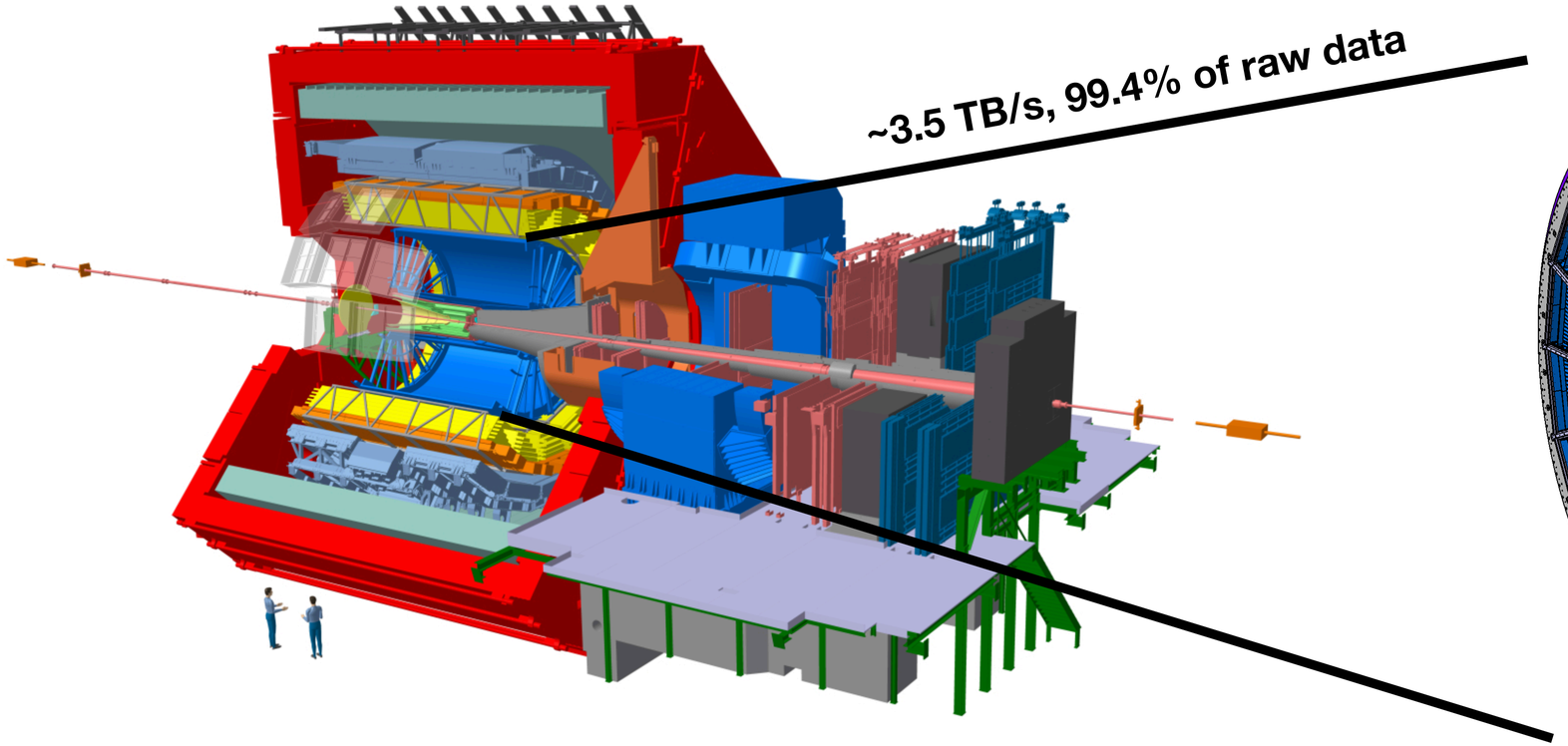
Christian Sonnabend - ALICE PDP group



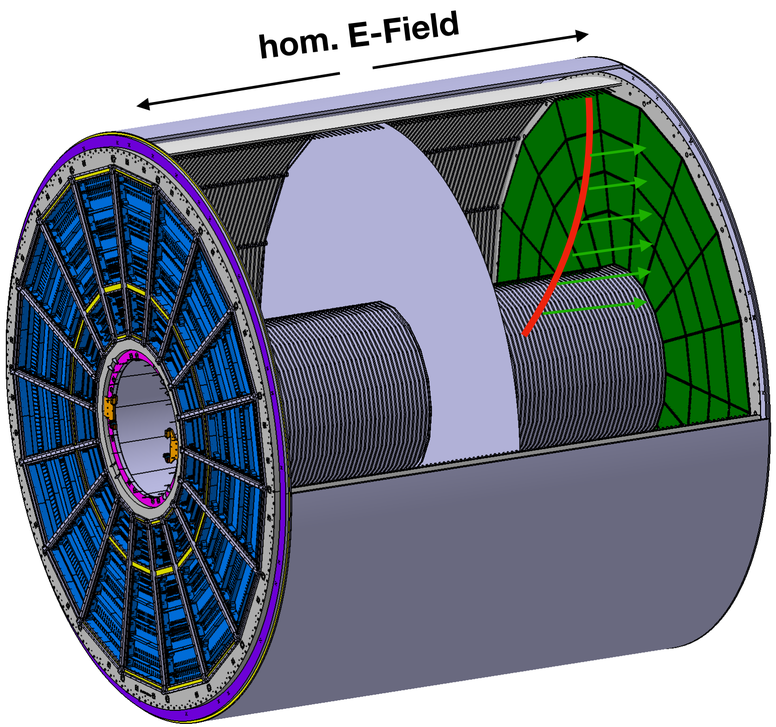
CHEP 2024, track 2: online processing - Krakow
??.??.????

Introduction

ALICE

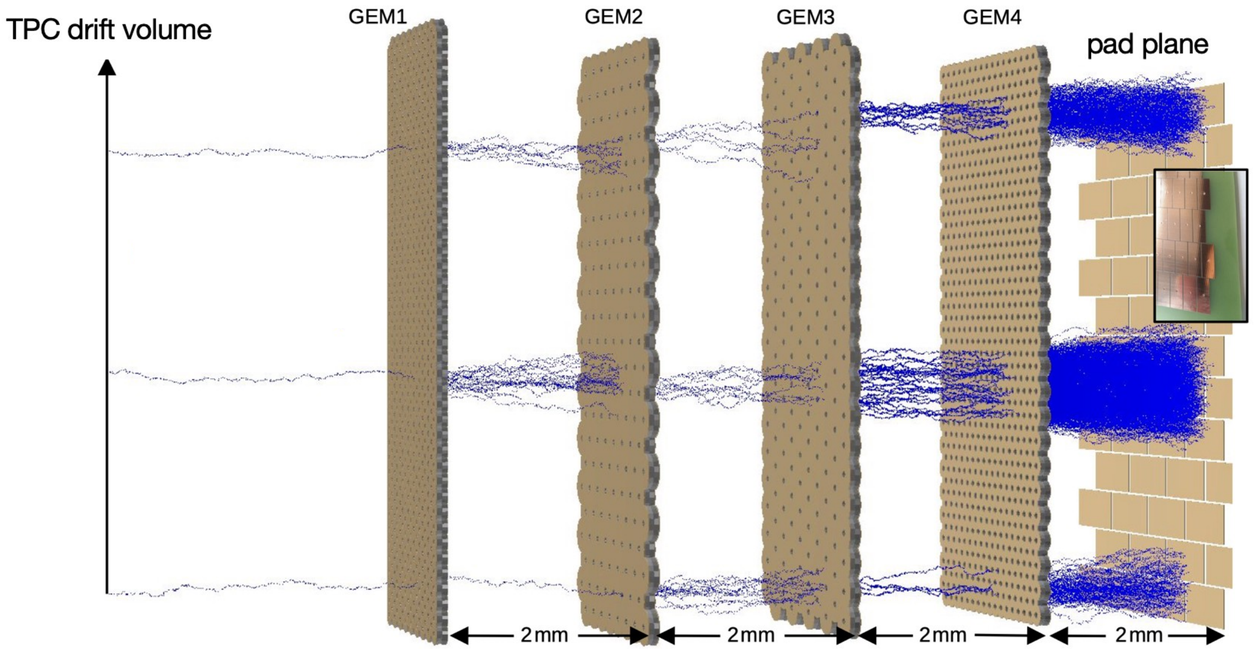


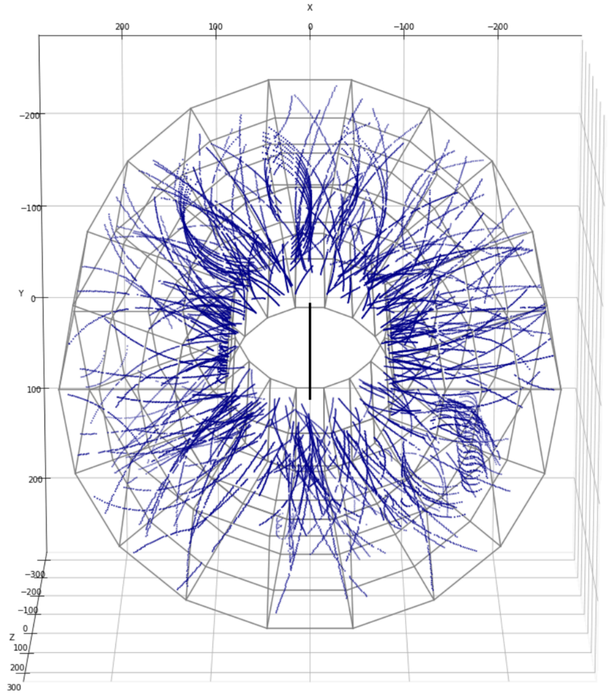
TPC - Time projection chamber



Central tracking and PID detector

Introduction





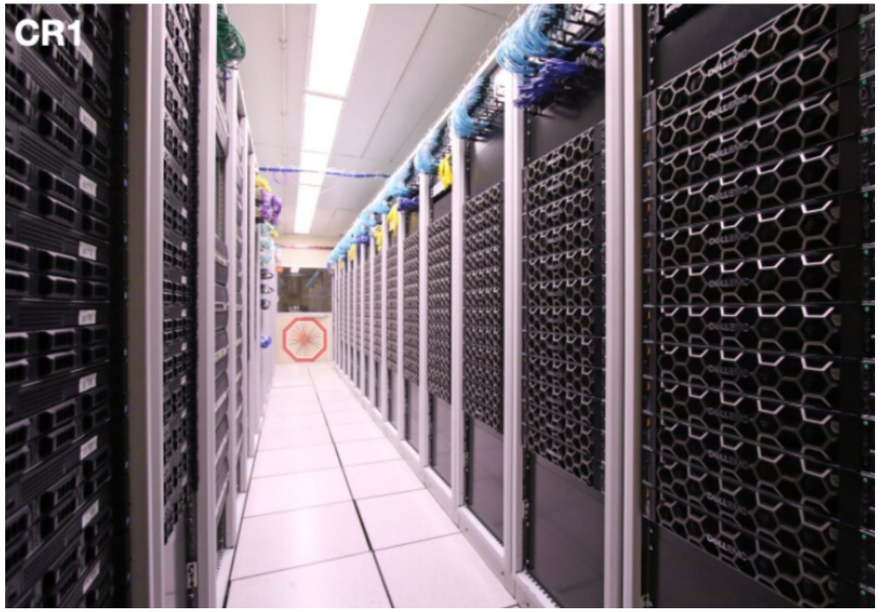
Raw electrons from tracks → **GEM amplification stage** → **Readout** → **Digits with measured charge value** → **Clusters**

From clusters we can do e.g. tracking or PID via dE/dx

Introduction

Hardware resources & constraints

- 350 EPNs (event processing nodes) for online reconstruction
- Each server: 8 MI50/MI100 GPUs, O(100) cores, O(1 TB) RAM
- Incoming data-rate: ~3.5 TB/s at peak load, ~50 mio. clusters/GPU/s

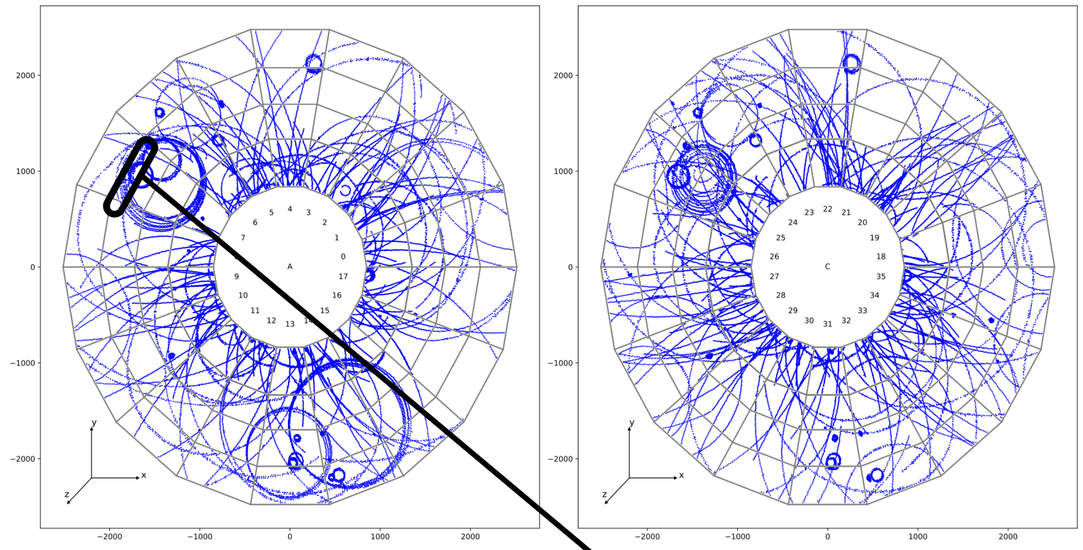


First level processors (FLP)

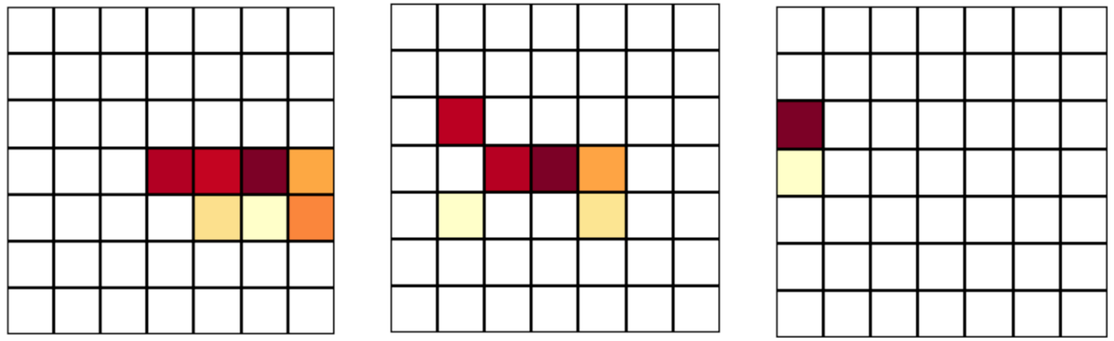
Event processing nodes (EPN)


Training data

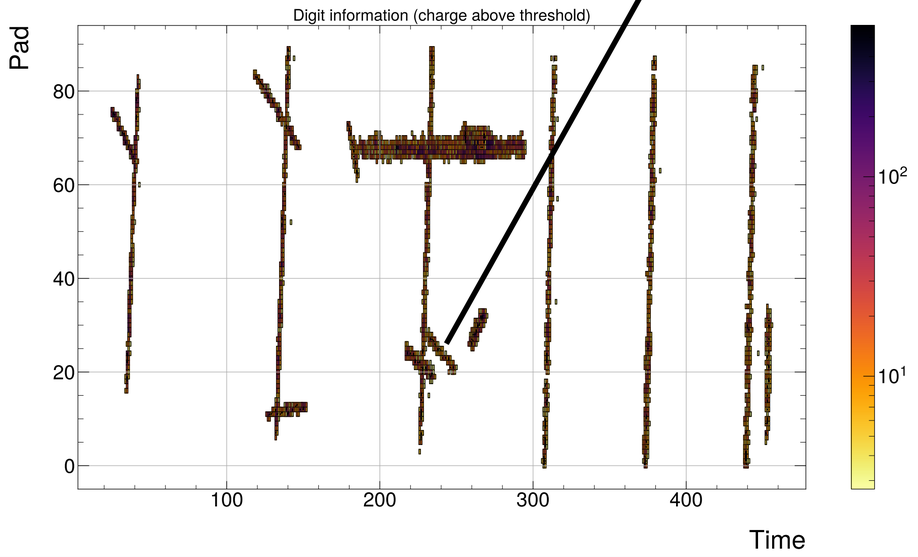
Data generation

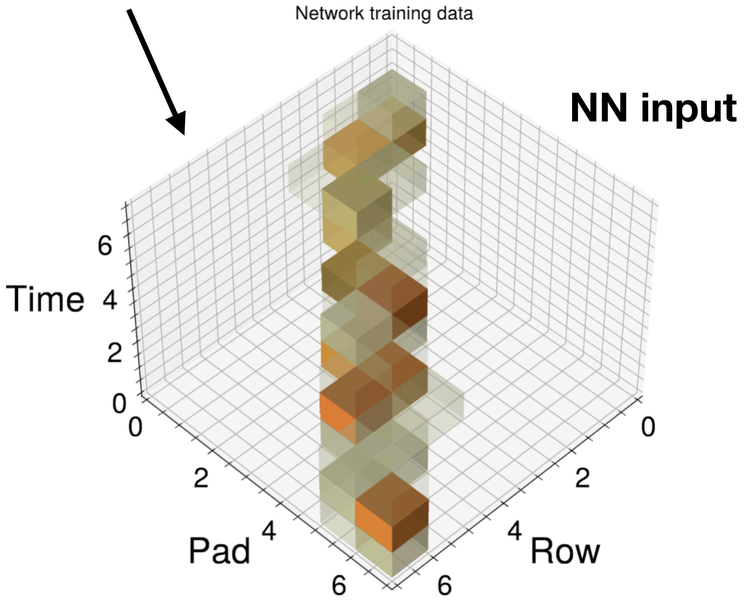


Raw TPC digit data



Adjacent rows



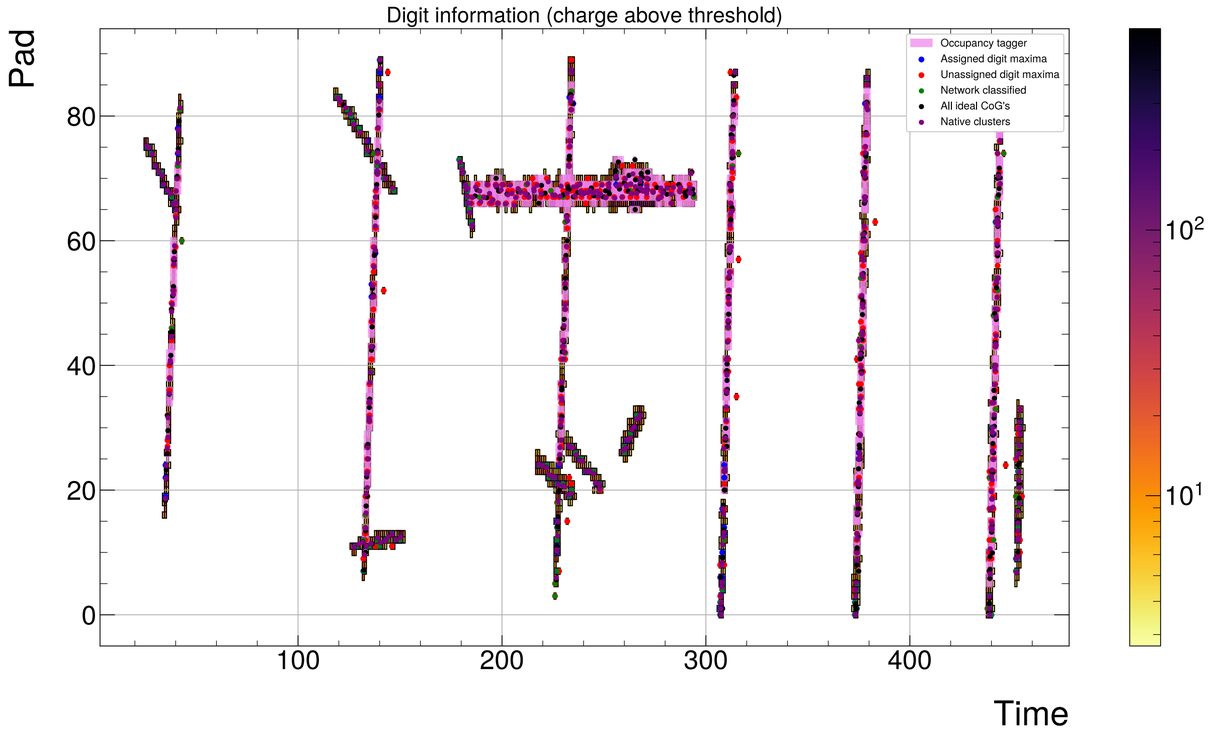


NN input

Assignment & training data selection

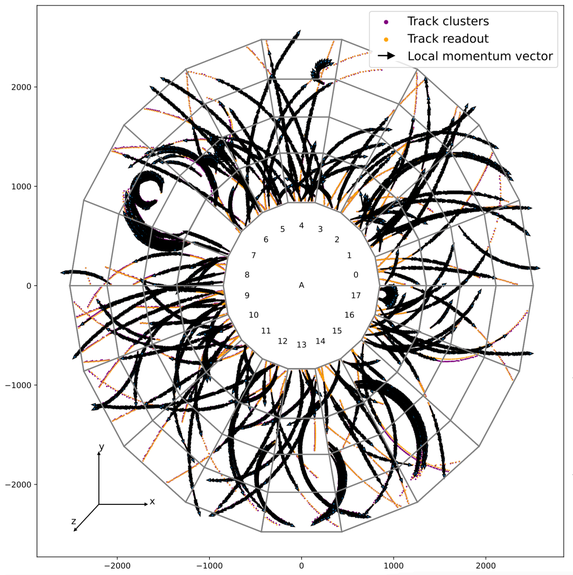
Simulation data

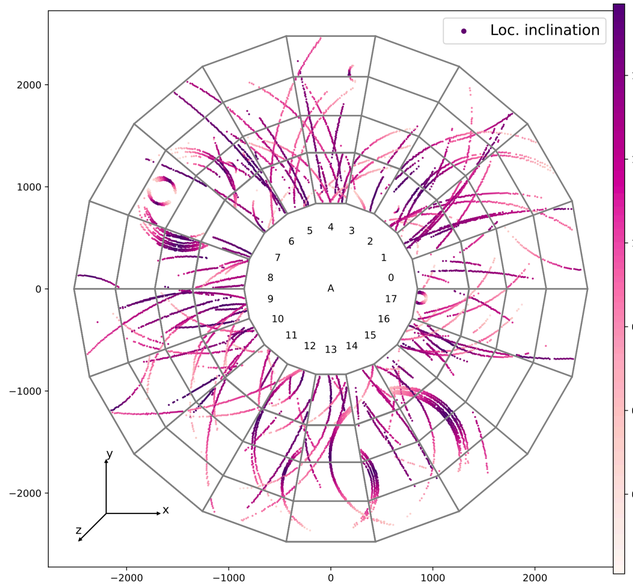
- Assign digit maxima with regular clusters
- Reject clusters if MC label occurs often in specified region of pad and time

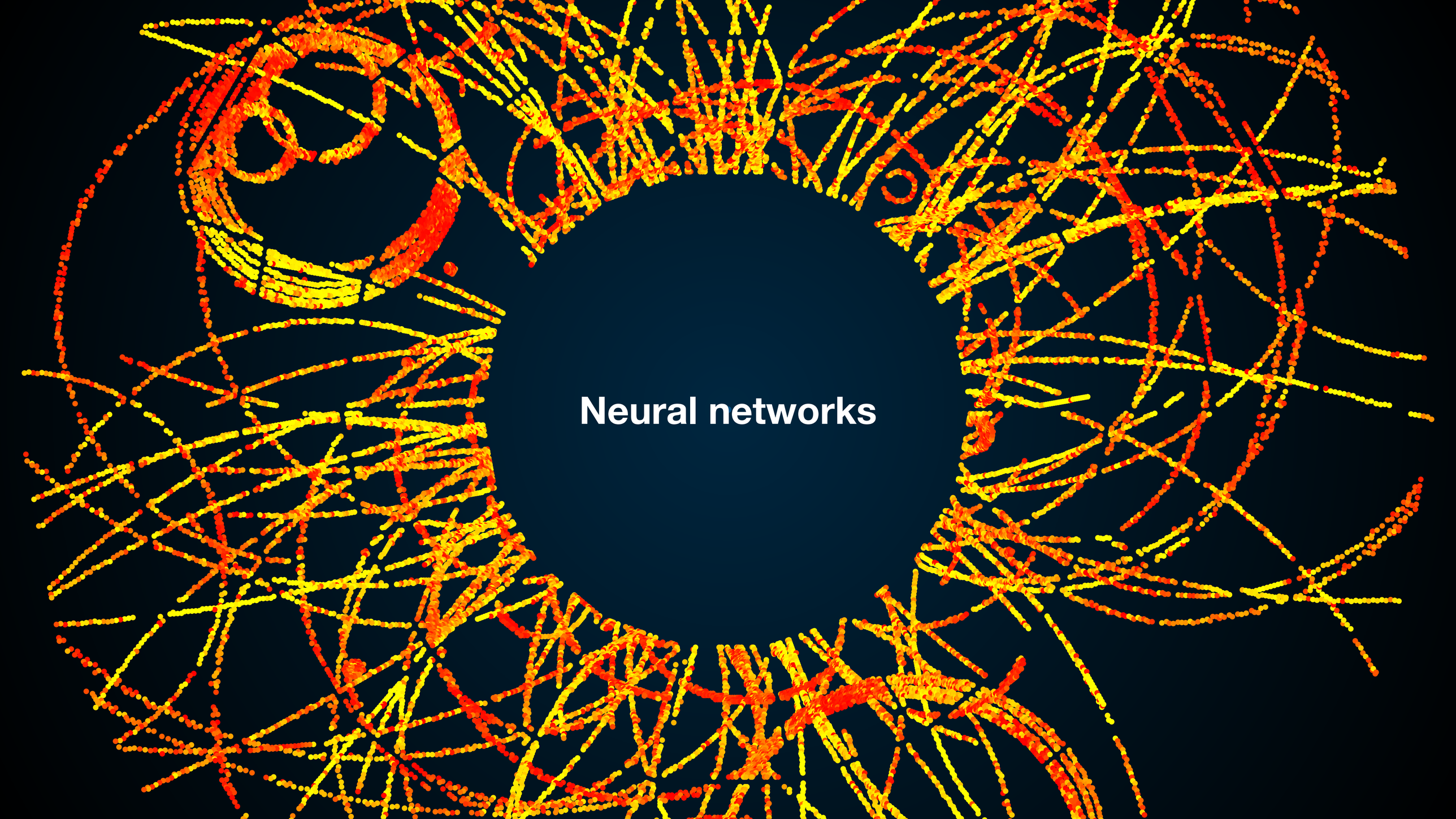


Real data

- Perform assignment between digit maxima and track paths
- Attach local momentum vector after reconstruction and reject clusters where loc. inclination angle is too high





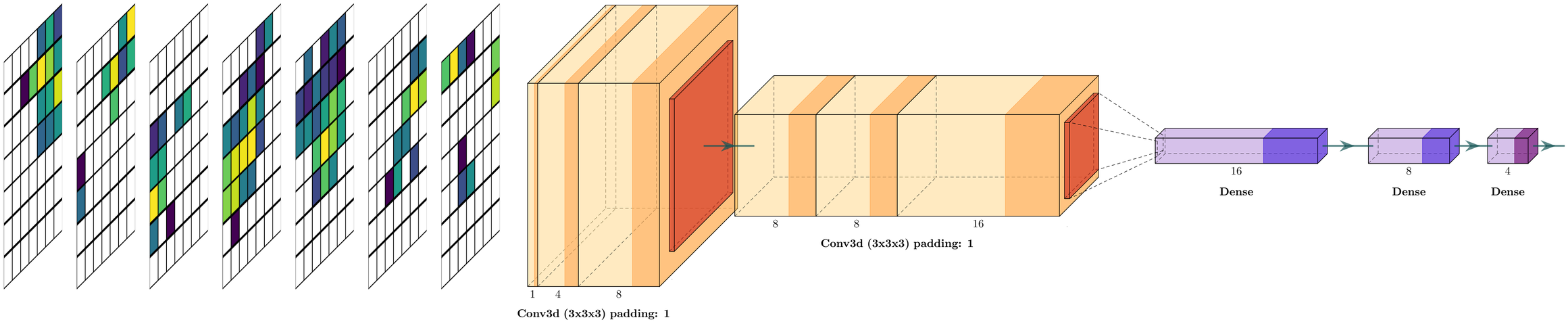

Neural networks

Neural networks

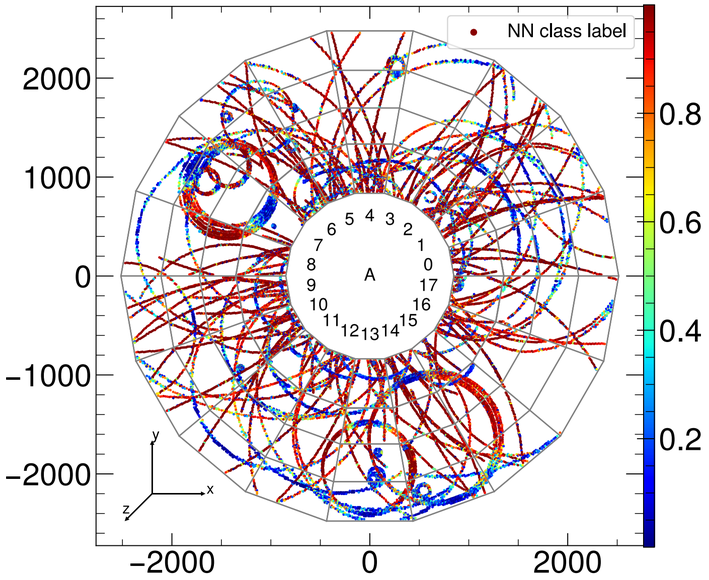
Tasks at hand

- Classification: Should digit maximum be converted to cluster or be rejected
- Regression: Predict cluster position, sigma, total charge and momentum vector
- Splitting: Should a cluster be split into two or more clusters

—> Make it fast enough for online processing

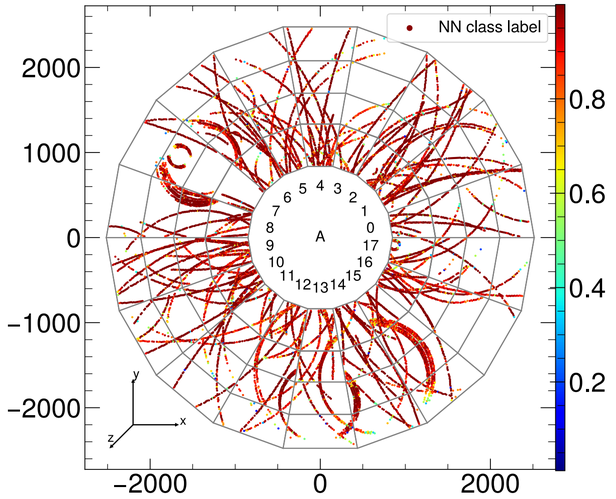


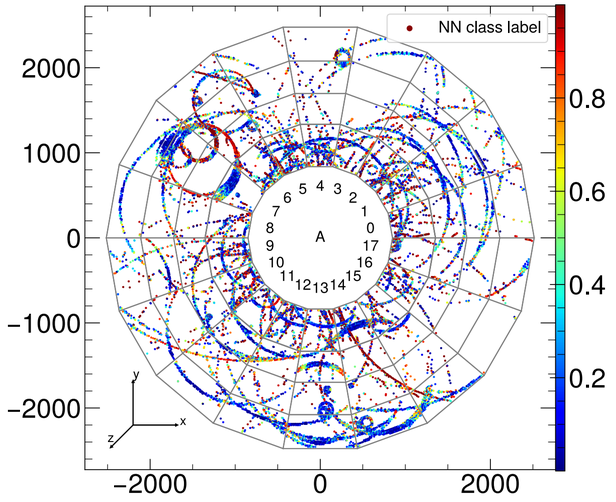
Classification network performance

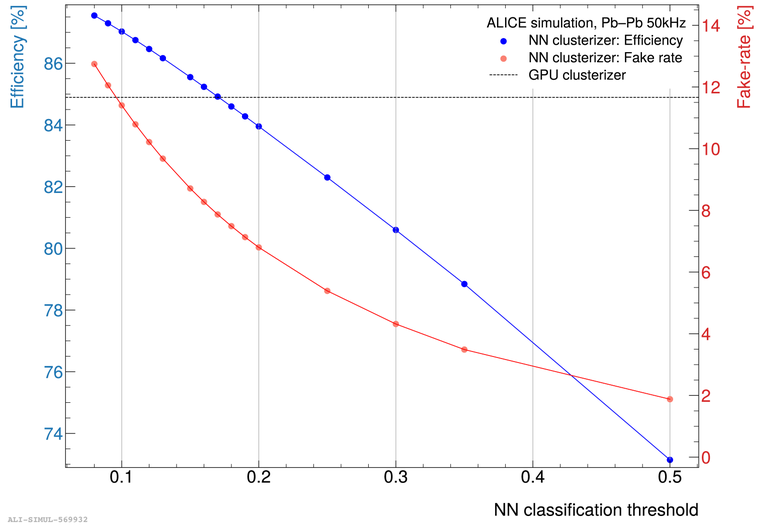


Should accept

Should reject

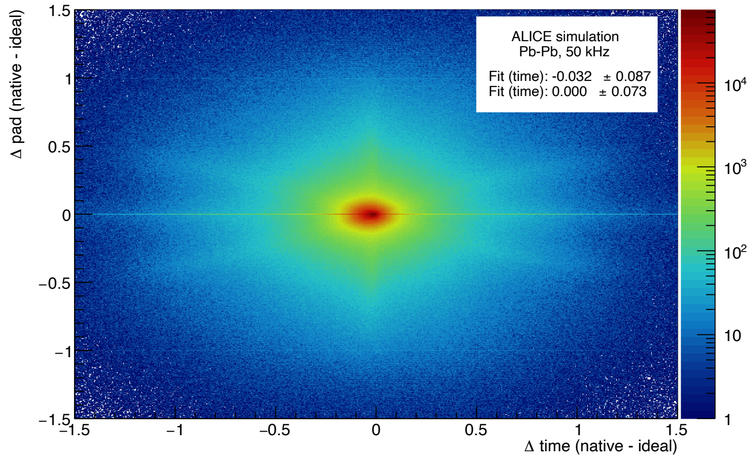
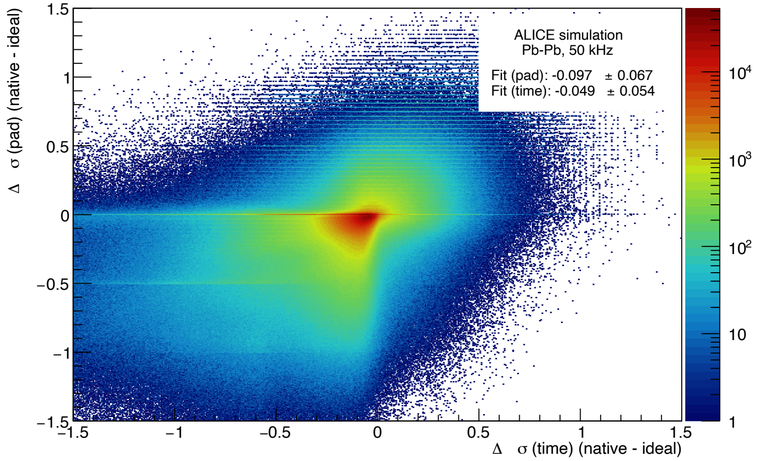
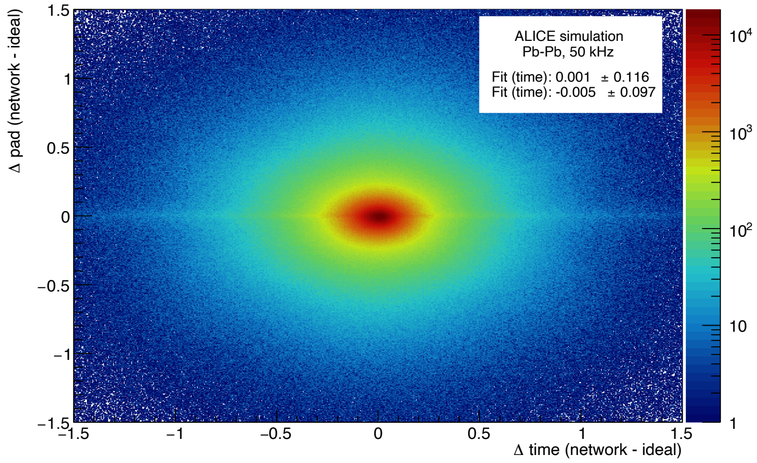
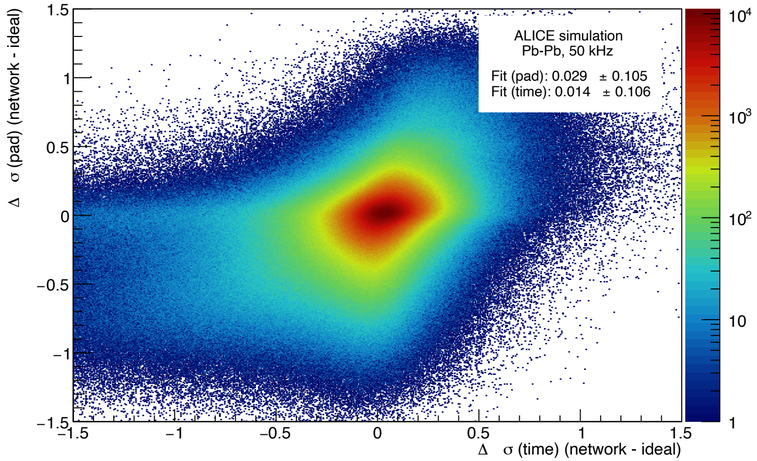






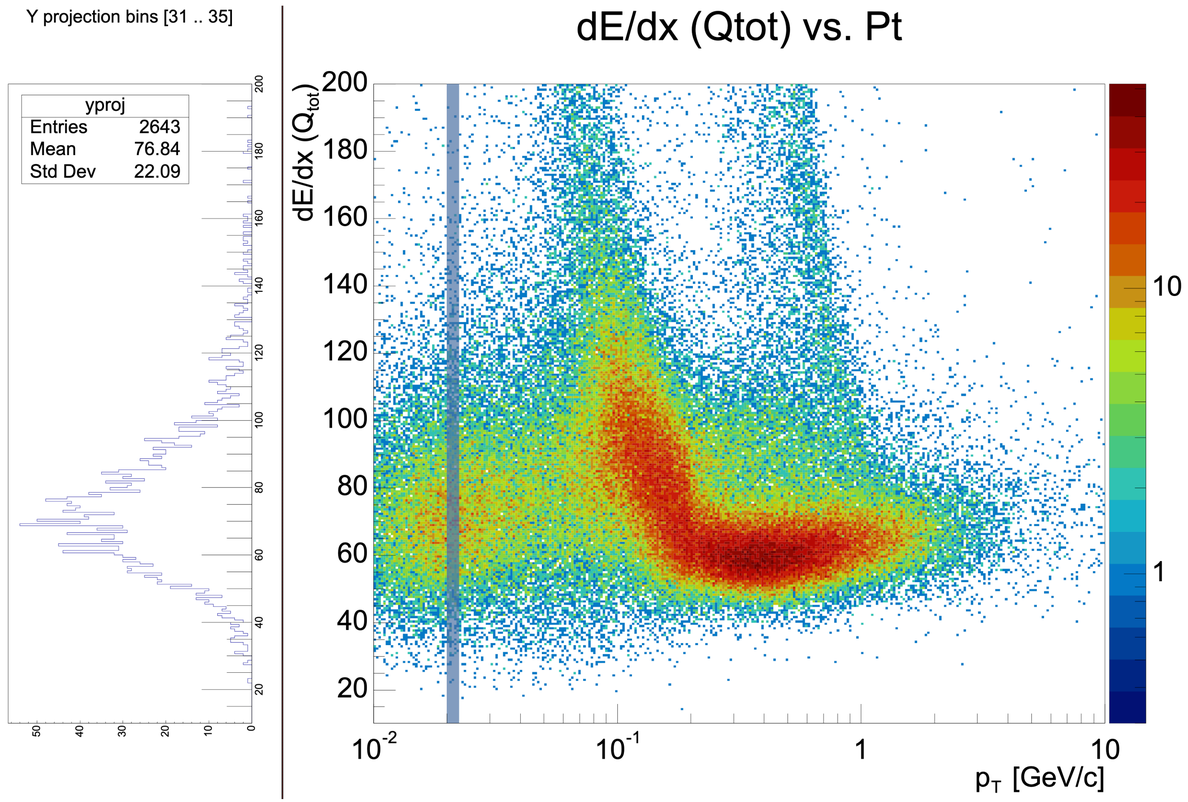
NN can reduce clusterization fake-rate by O(30%)!

Regression network performance

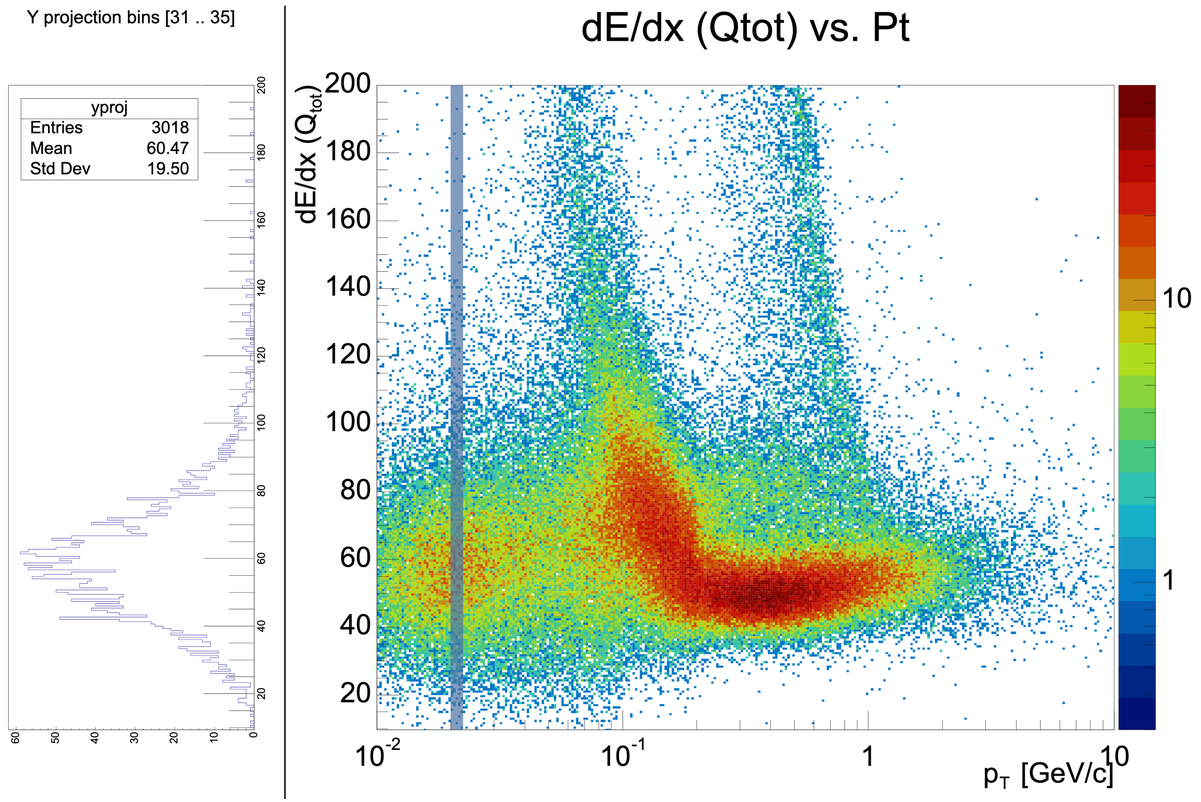
Pad vs. time	Center of gravity estimate	Resolution (σ) estimate
GPU cluster finder		
NN cluster finder		

Neural network performance

Neural network performance

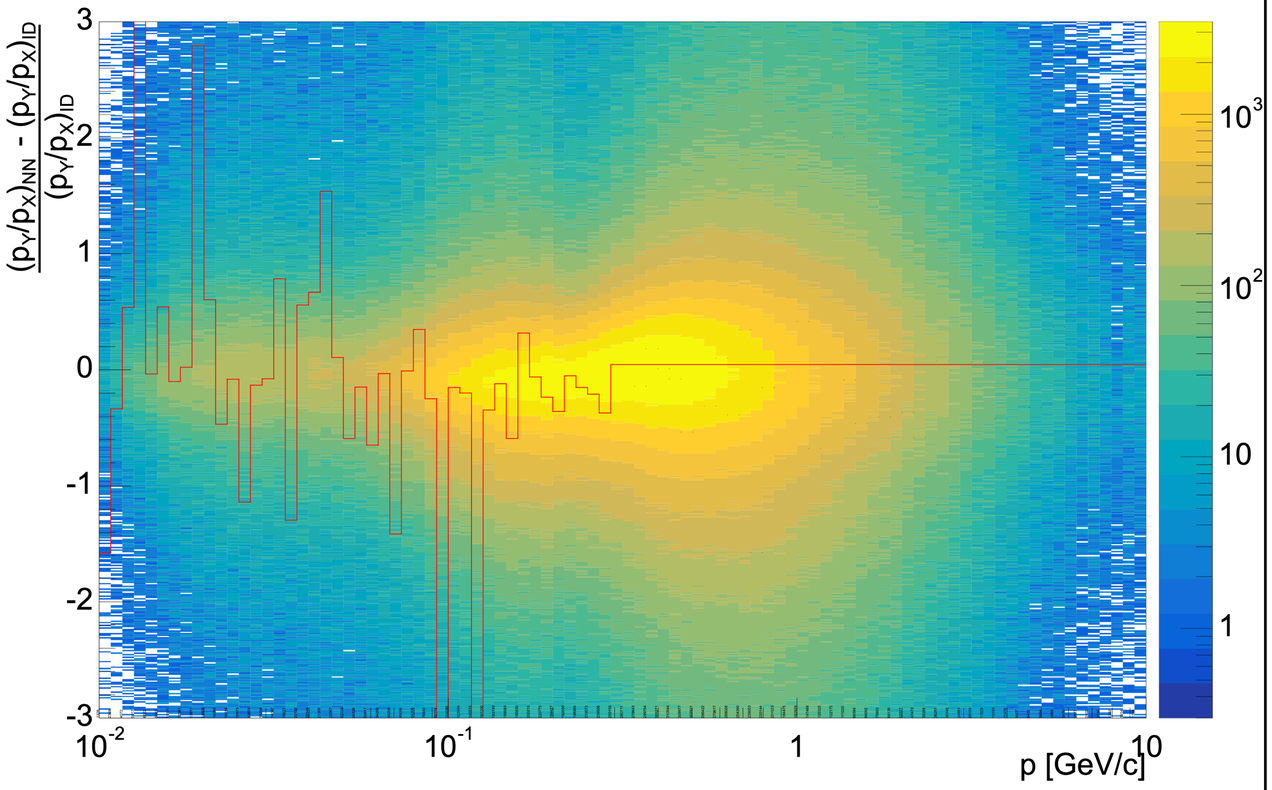


GPU clusterizer performance

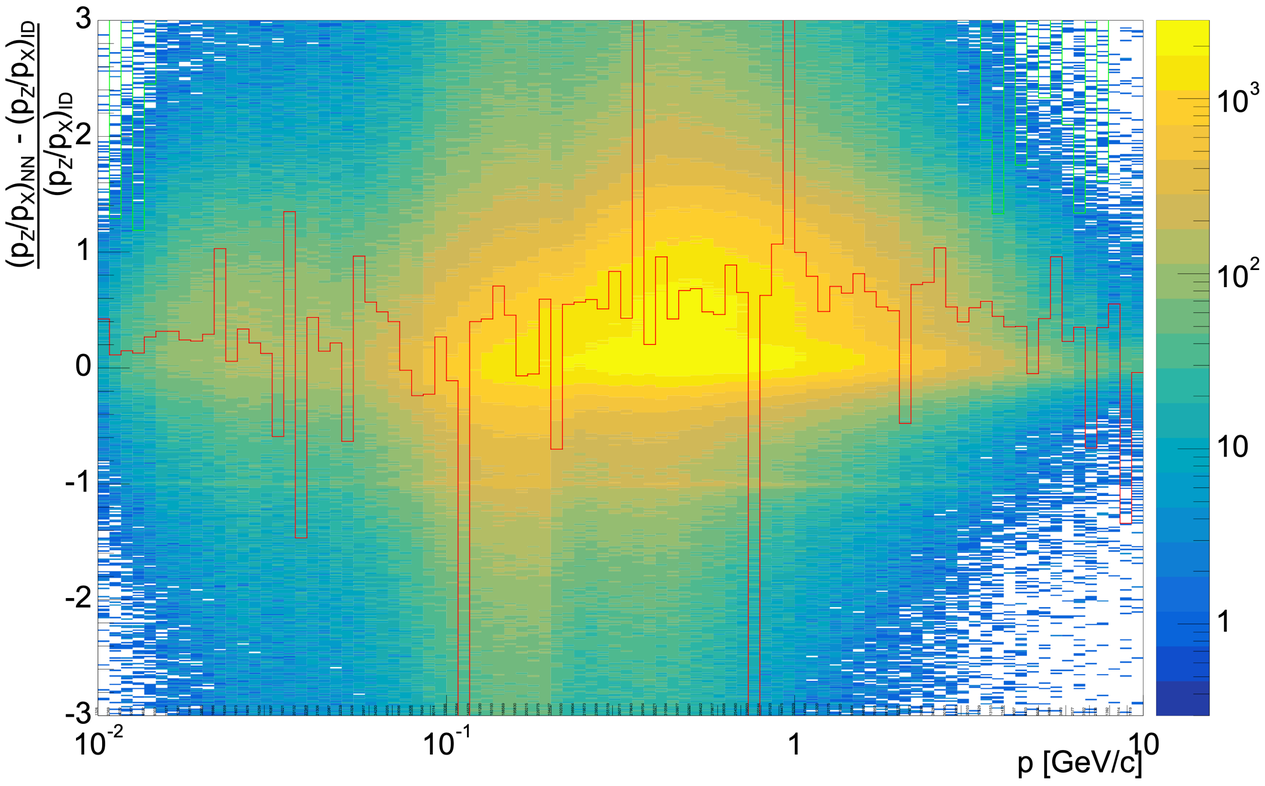


Neural network performance

NN: p_Y/p_X performance

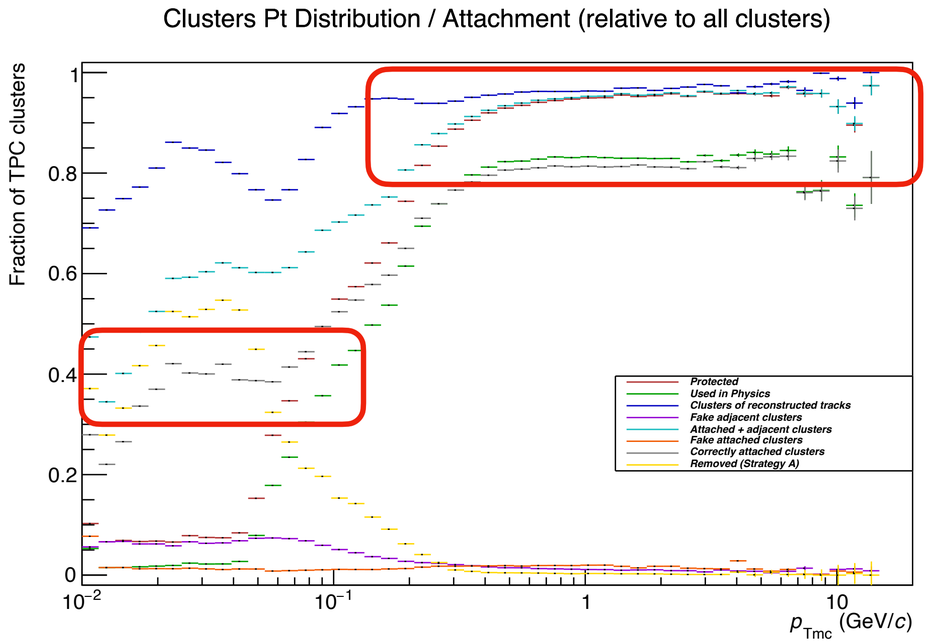


NN: p_Z/p_X performance



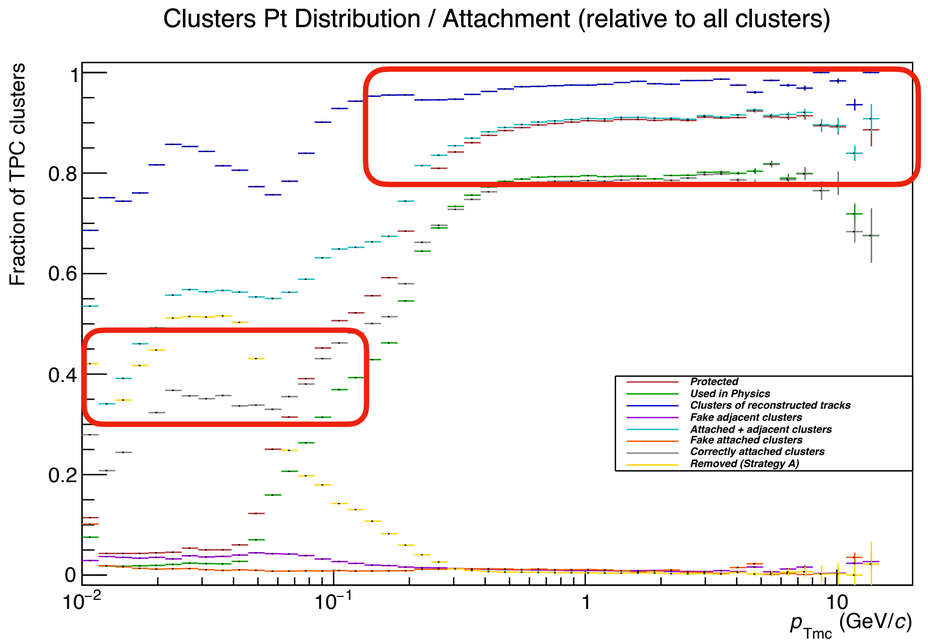
Clusterization performance

Neural network performance



Total number of clusters: 17.9 mio.

GPU clusterizer performance

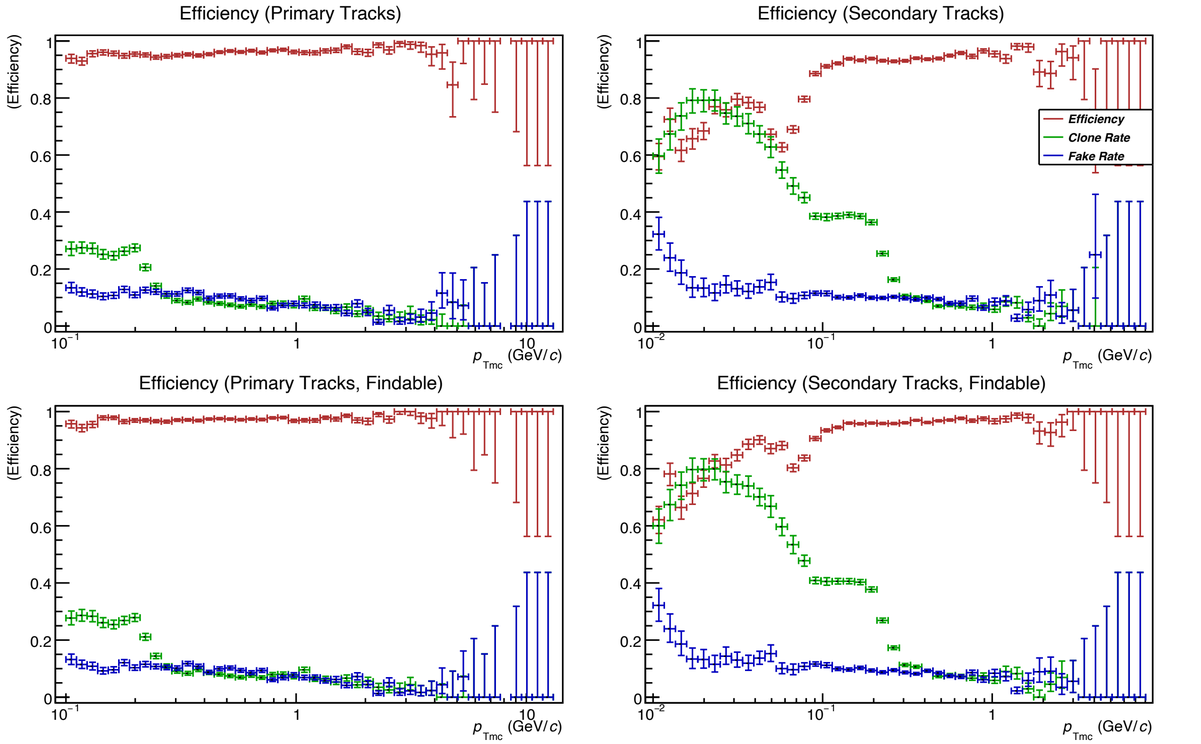


Total number of clusters: 21.4 mio.

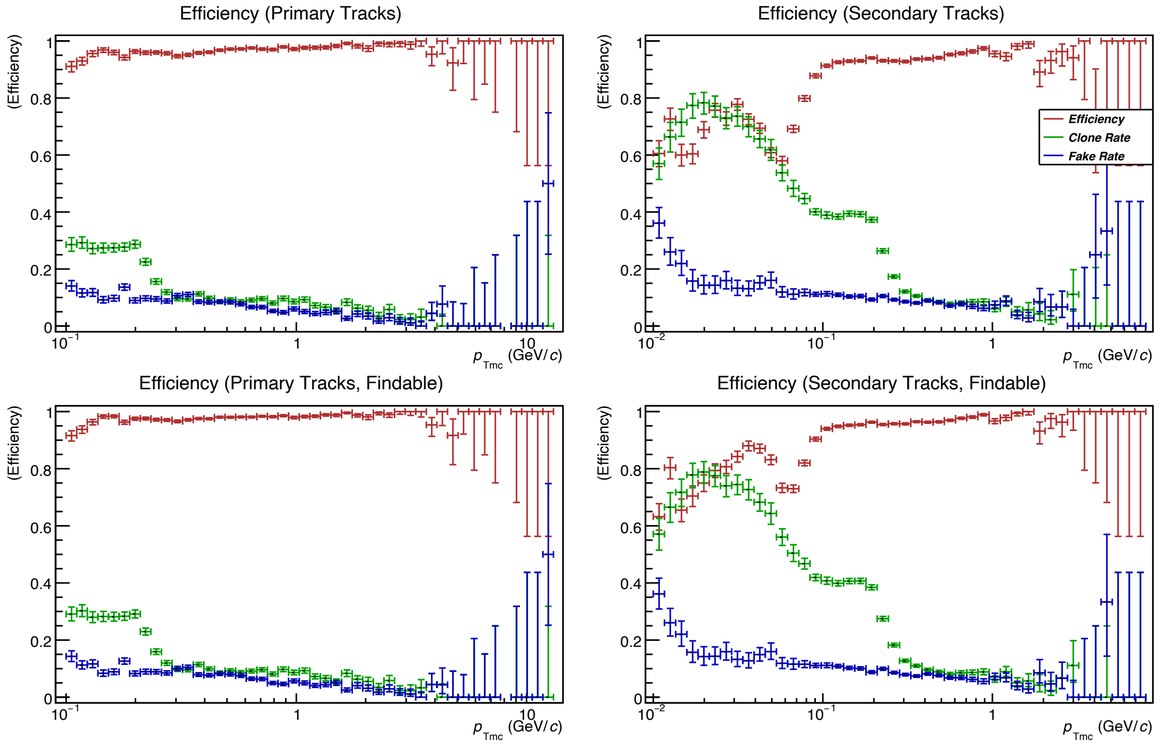
Significantly improved cluster attachment ratios after NN application:
More clusters used in physics, higher efficiency (O(10%))

Tracking performance

Neural network performance



GPU clusterizer performance



Tracking performance

Adding ratio plots of tracking efficiency here

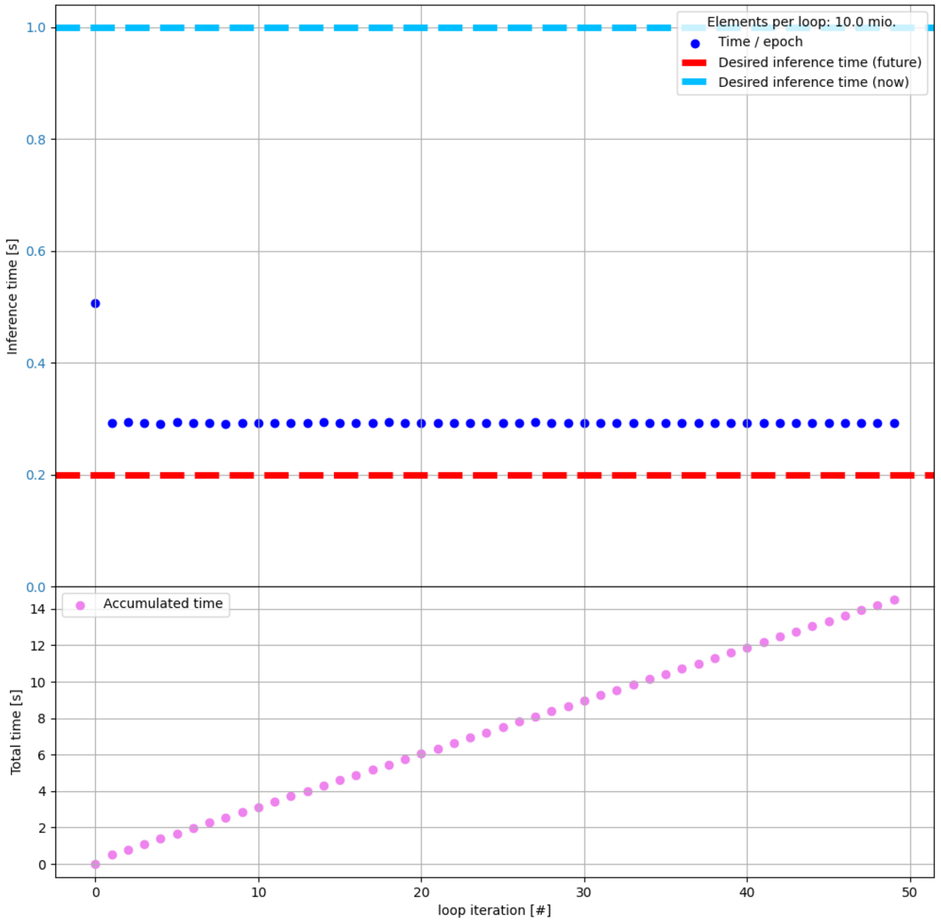
Processing speed & Design choices

Goal: Inference needs to be fast enough for online processing

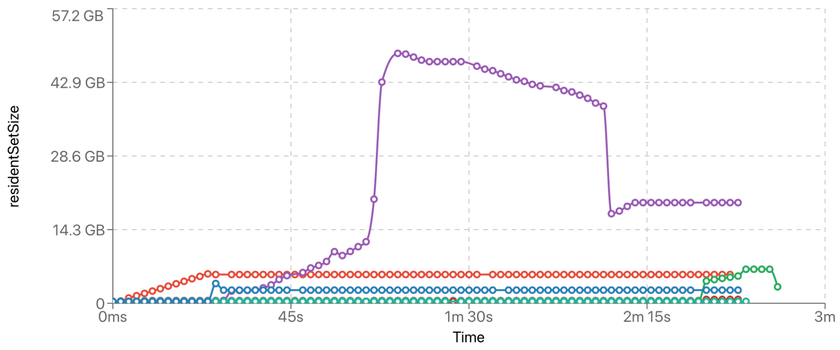
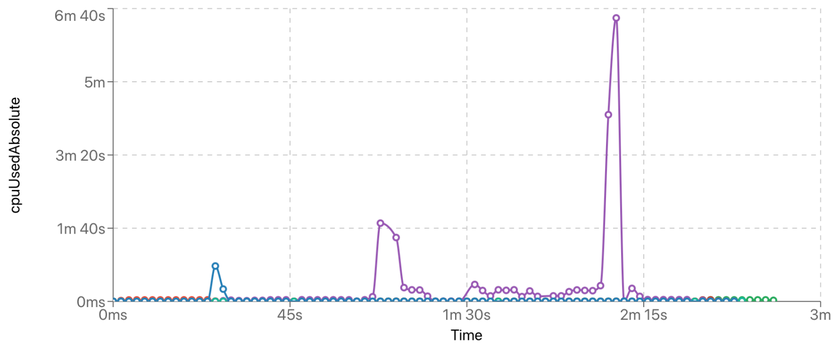
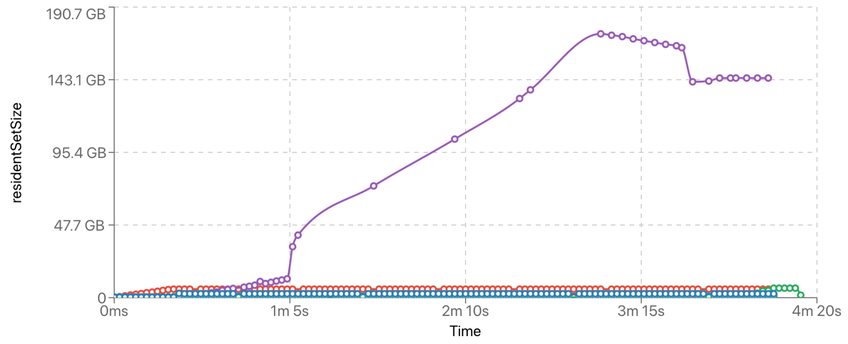
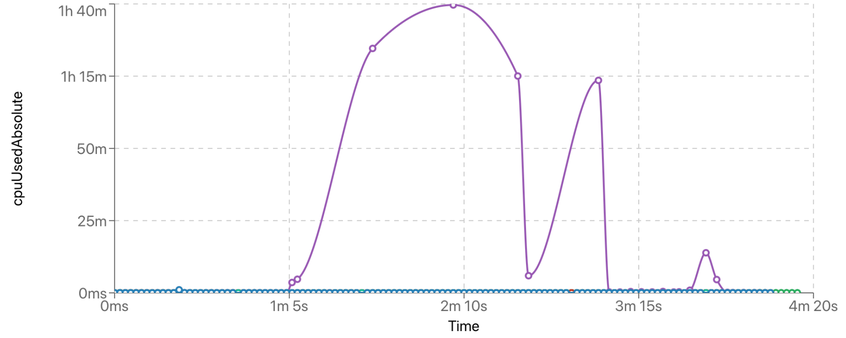
- Trade-off: precision <-> speed -> Use Float16 implementation
- Measured in clusterization code: ~30 mio. clusters / s

Design choices

- NN design choices: Fully-connected or 2D convolutional layers are well optimised
- Inference framework: ONNX runtime with build options for MI50 & MI100 GPU's



Processing speed & Design choices

Pad vs. time	Memory consumption	CPU time
GPU cluster finder		
NN cluster finder		

Conclusion

Classification network

- Successfully rejects clusters that are not used in tracking
  - This could reduce effective data-size by O(20%)!
- To-do: Predict cluster splitting -> Limited in training data

Regression network

- For single clusters: Comparable performance to current clusterizer
- Novel: Predict momentum of cluster (apparently with great success!)
- To-do: Can this be done well also for clusters that need to be split?

Thank you for listening!

Topics:
- O2 processing chain
- MC clusterizer & Ideal data
- Momentum data & Real tracks
- Training data selection
- Classification network
- Regression network
- Implementation on GPU & Speed