A Deep-Learning Reconstruction Algorithm for Cluster Counting

[1]Guang Zhao, [2]Zhefei Tian, [1]Linghui Wu, [3]Brunella D'Anzi, [1]Mingyi Dong, [3]Gianluigi Chiarello, [3]Nicola De Filippis, [3]Francesco Grancagnolo, [1]Shuaiyi Liiu, [1]Shengsen Sun, [1]Shuiting Xin, [2]Zhenyu Zhang

CHEP 2023, Norfolk, USA, May 9, 2023

1. Institute of High Energy Physics
2. Wuhan University
3. Istituto Nazionale di Fisica Nucleare

Outline

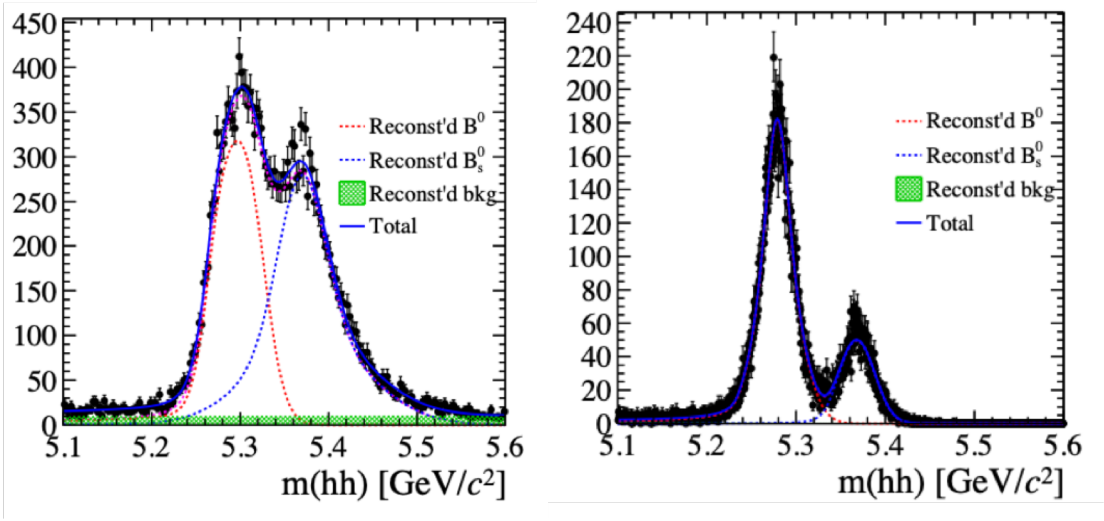
- Motivation
- Algorithm
  - Peaking finding algorithm
  - Clusterization
- Preliminary result on beam test data
- Summary and plan

PID in future lepton collider experiments

- Particle identification is essential for flavor physics and jet study
  - Reduce combination background
  - Improve mass resolution
  - Improve jet energy resolution
  - Benefit flavor tagging

Example of the impact of PID in heavy flavor decay reconstruction

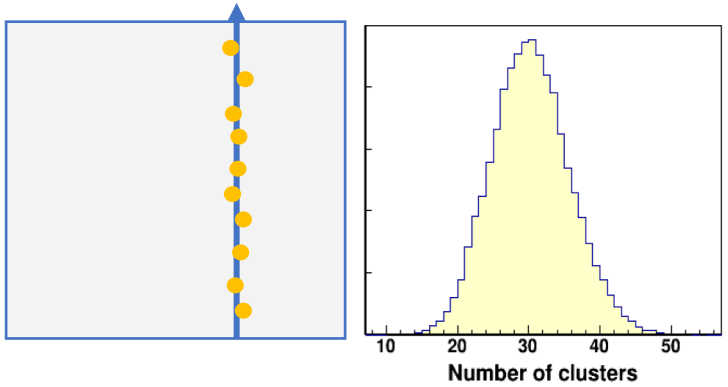
Disentangle the various $B_s^0(B^0) \to h^-h^+$ in same topology final-states.



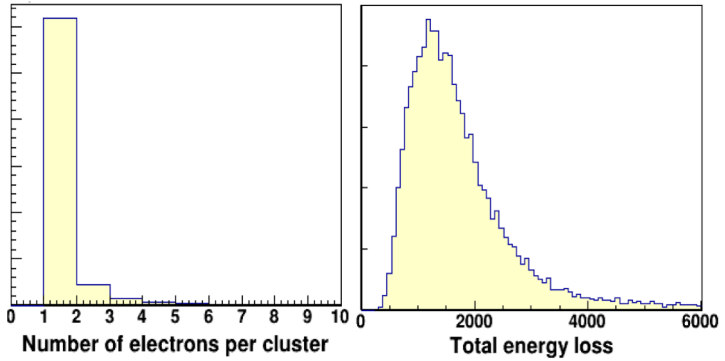
Simulation at CEPC by Shanzhen

Cluster counting vs dE/dx

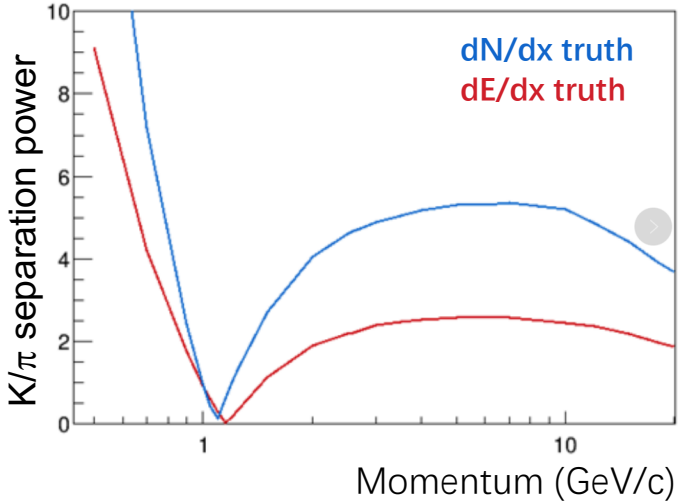
Primary ionization



Secondary ionization

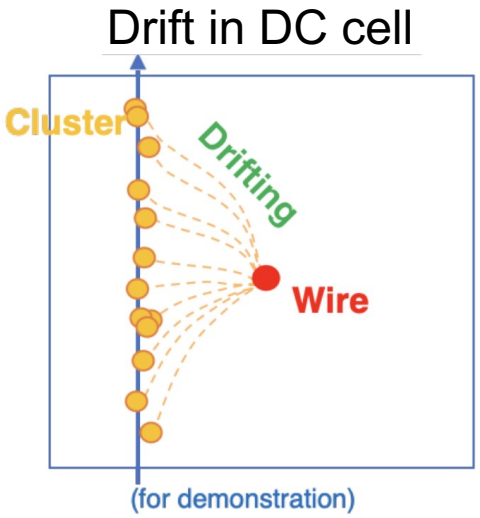


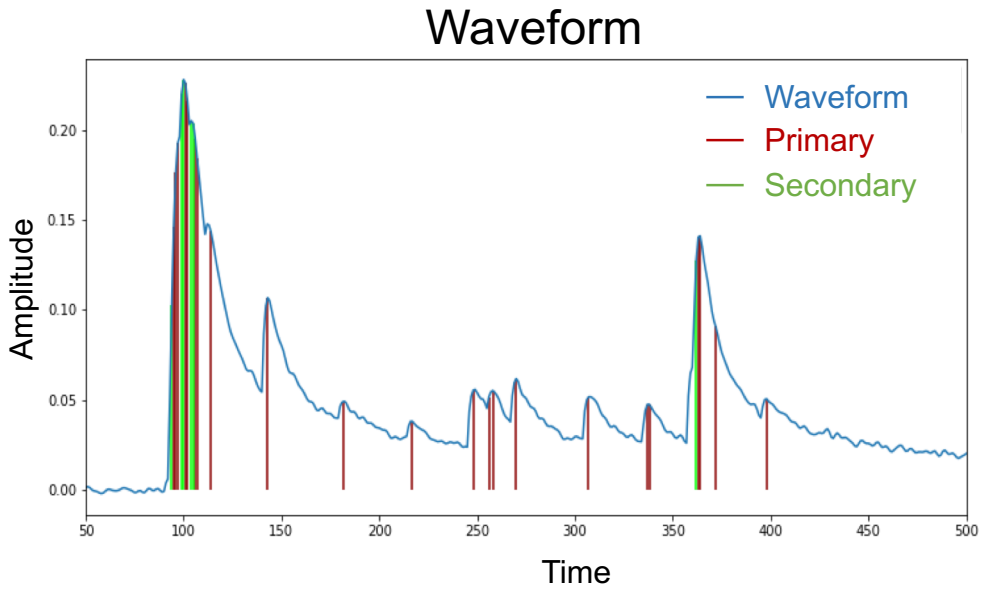
K/π separation power
dN/dx vs dE/dx



- dE/dx: Energy loss per unit length, Landau distribution, large fluctuation
- dN/dx: Number of primary ionization clusters per unit length, Poisson distribution, small fluctuation → cluster counting technique

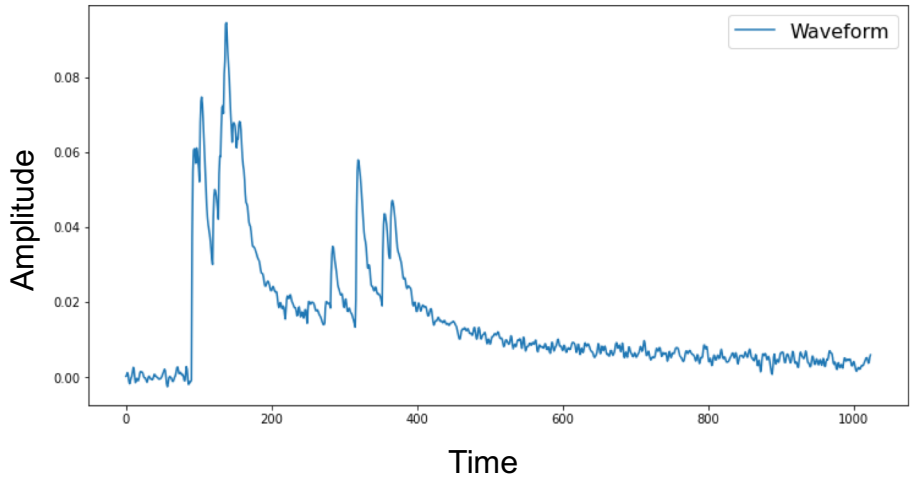
Cluster counting in drift chambers





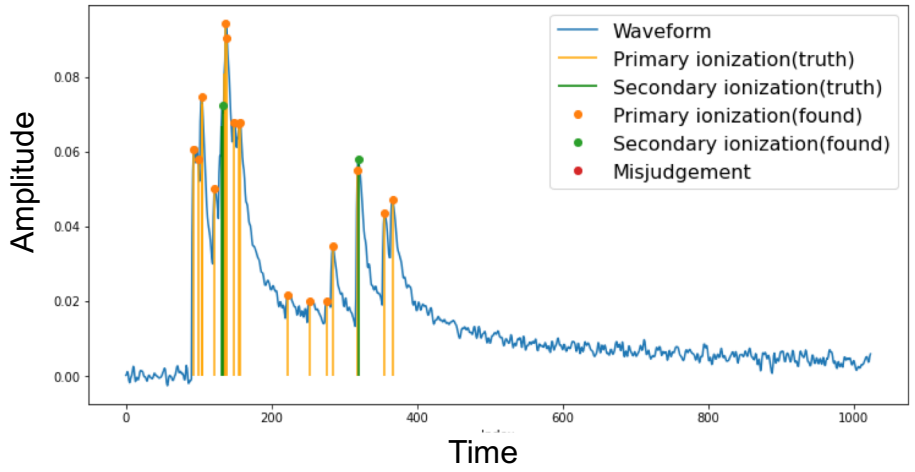
- **Tasks of reconstruction software**
  - Both **primary** and **secondary** ionization contribute peaks on the waveform
  - Find the number of peaks from **primary ionization**
- **Challenges**
  - High pile-up
  - Could be noisy
- **Machine learning** can make full use of the waveform information, could be effective

Two-step reconstruction algorithm



Step1. Peak Finding

Discriminate peaks (both primary and secondary) from the noises (classification problem)

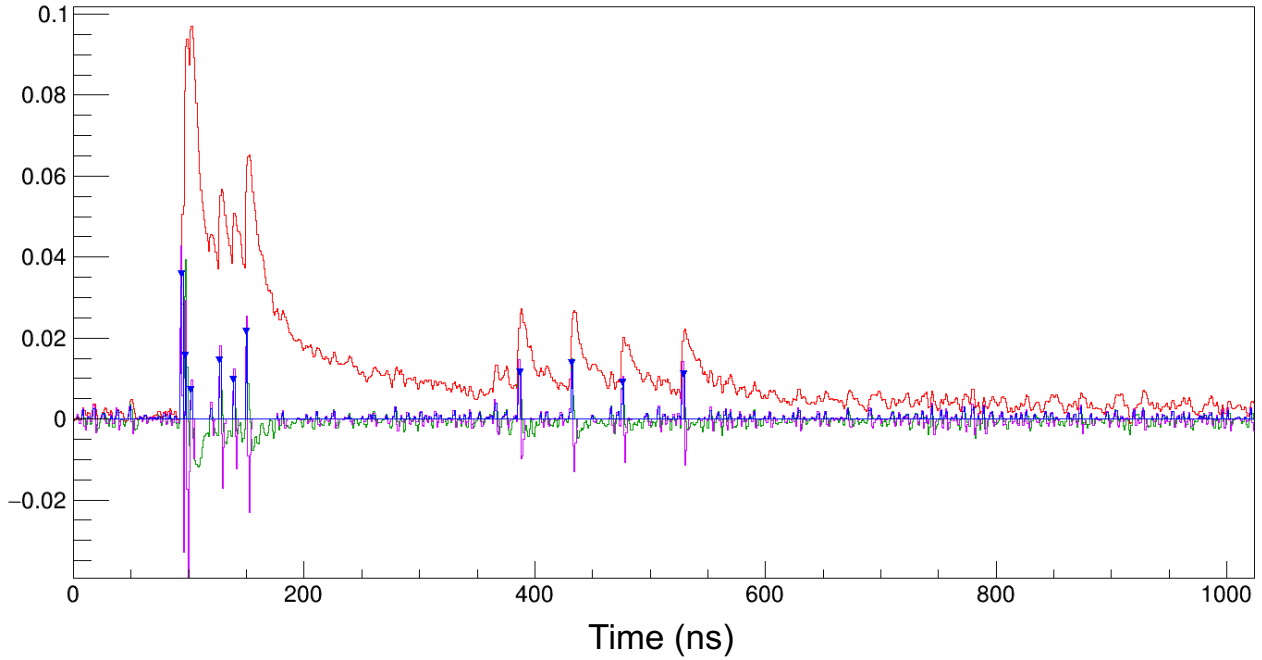


Step2. Clusterization:

Determine the number of clusters (N_{cls}) from the detected peaks (regression problem)

Derivative-based algorithm

Peak finding algorithm with 1st and 2nd order derivatives



- Idea: Peak detection by the slope change of the rising edge
- Advantages: Simple and fast
- Disadvantages: Lose efficiency for highly pile-up and noisy waveforms

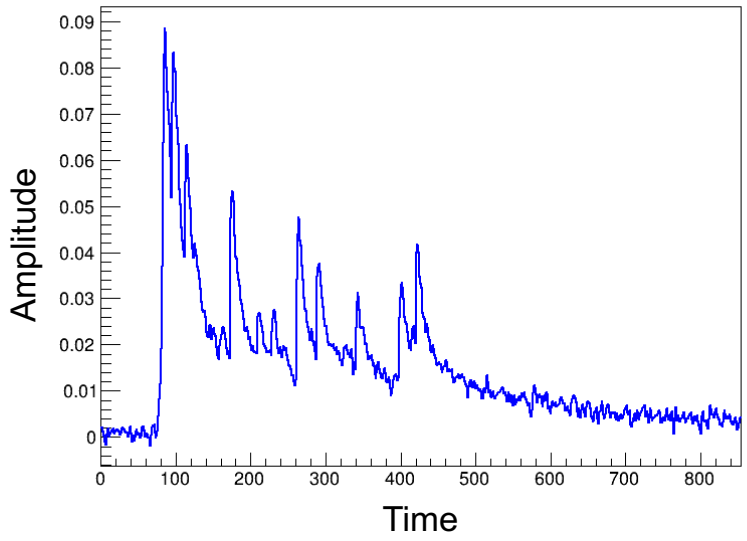
Deep-learning-based algorithm

- Traditional algorithm: Human input rules
- Deep learning: Learn rules from large amount of datasets

- Specifically, for cluster counting reconstruction:
  - Machine learning can make full use of the waveform information, not only information of pulse rising edge (e.g. derivative algorithm).
  - Machine learning can learn the hidden relationship in data (signal/noise characteristics, timing structure of primary/secondary peaks).
  - The reconstruction can easily be defined as classification and regression ⇒ apply mature ML tools like TensorFlow, Keras, PyTorch, etc.

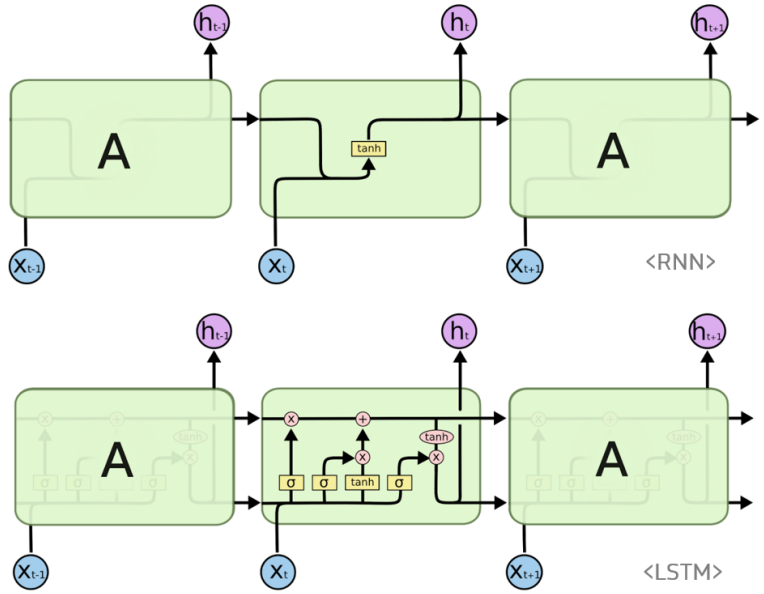
Step1: Peak finding

Waveform



- A classification problem to classify ionization signals and noises in the waveform
- The data of waveform is time sequence data, which is suitable for RNNs, especially LSTM
- Dataset:
  - Distribution of number of ionizations is flat: [1, 40]
  - 2,500 waveforms/point, 1,000,000 waveforms in total

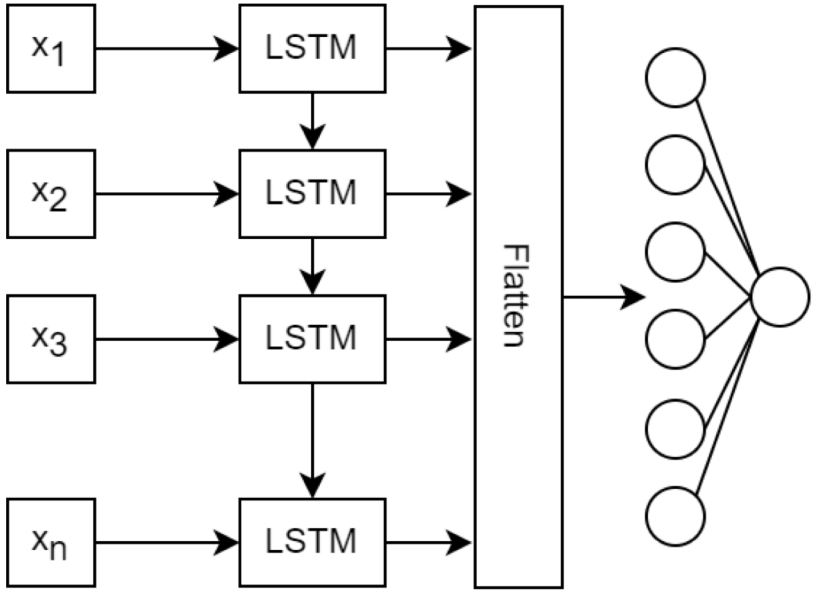
RNN (Recurrent Neural Network)



- With feedback loops, RNN has “memory”
- Well-suited to classifying based on time sequence data

Network structure and waveform processing

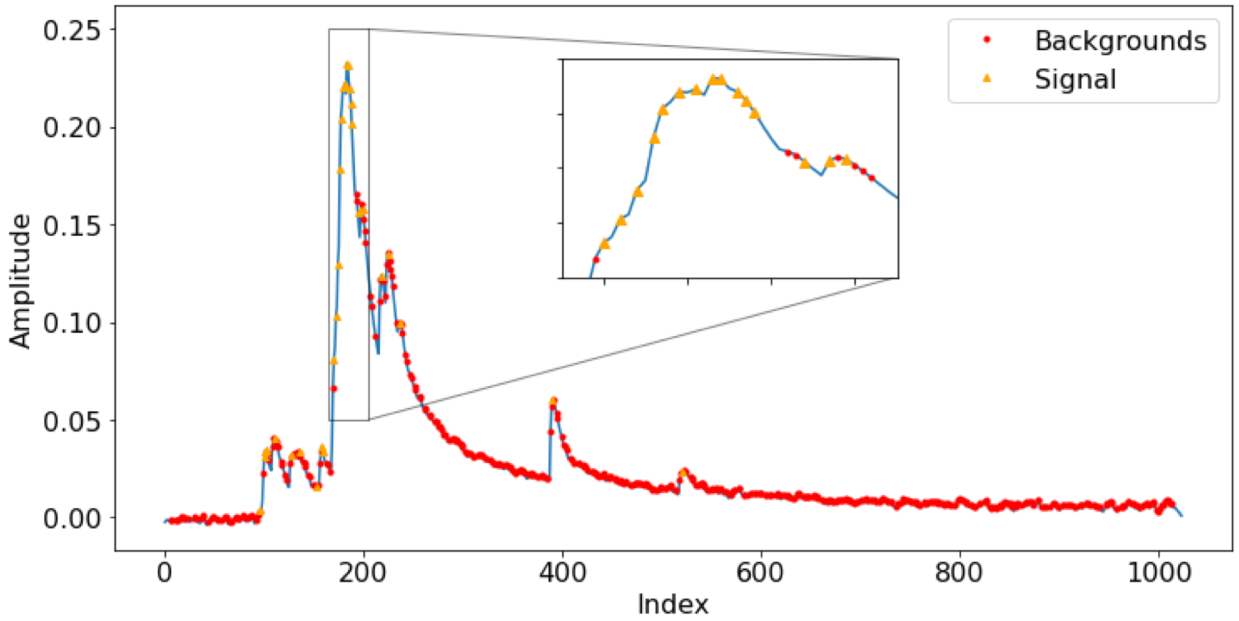
Long short-term memory (LSTM) model



- Labels: Signal or Noise.
- Features: Slide windows of peak candidates, with a shape of (15, 1)

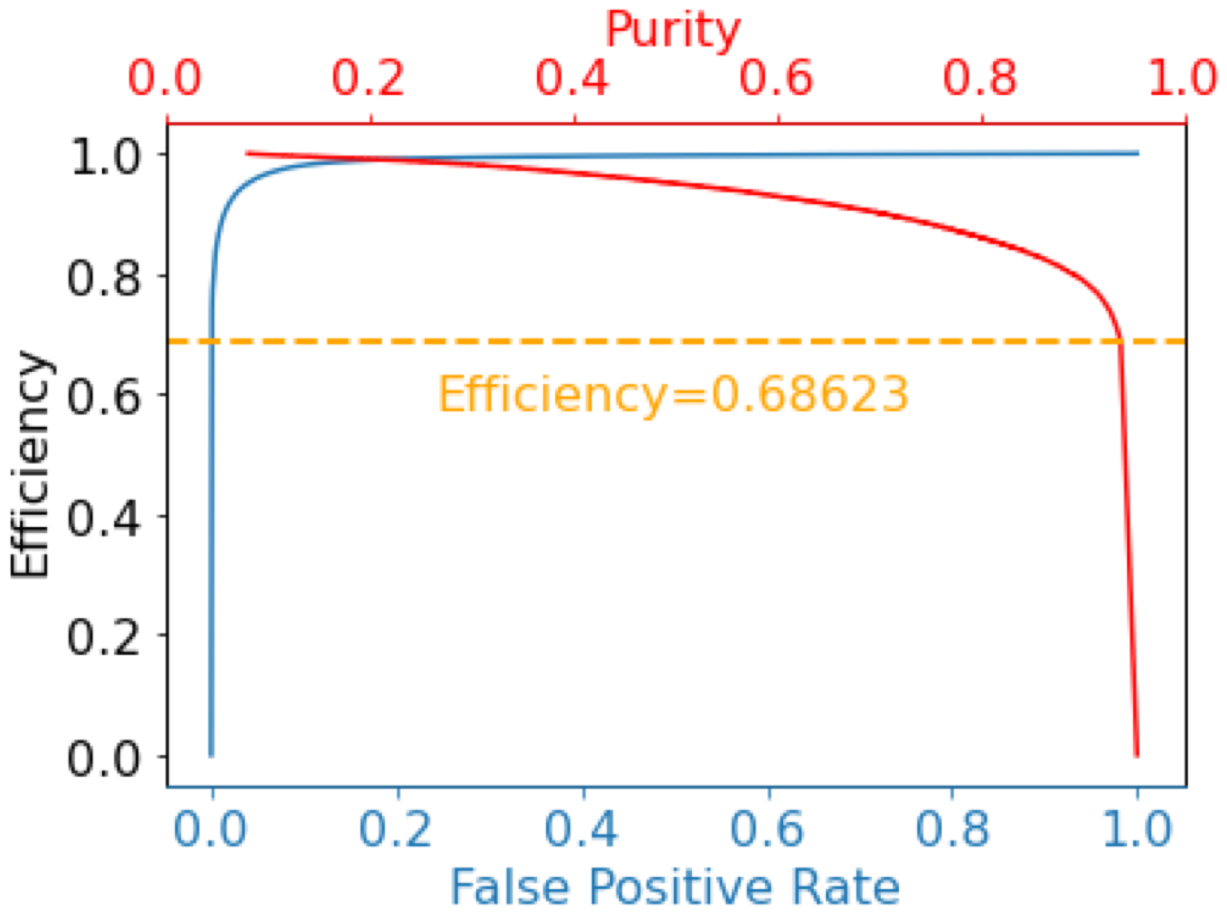
  ⇒ **A binary classification problem**

Processing of waveform



- Slide window samples: (−5, +9) bins
- Adding labels according to MC truth
- Balancing signal/noise samples

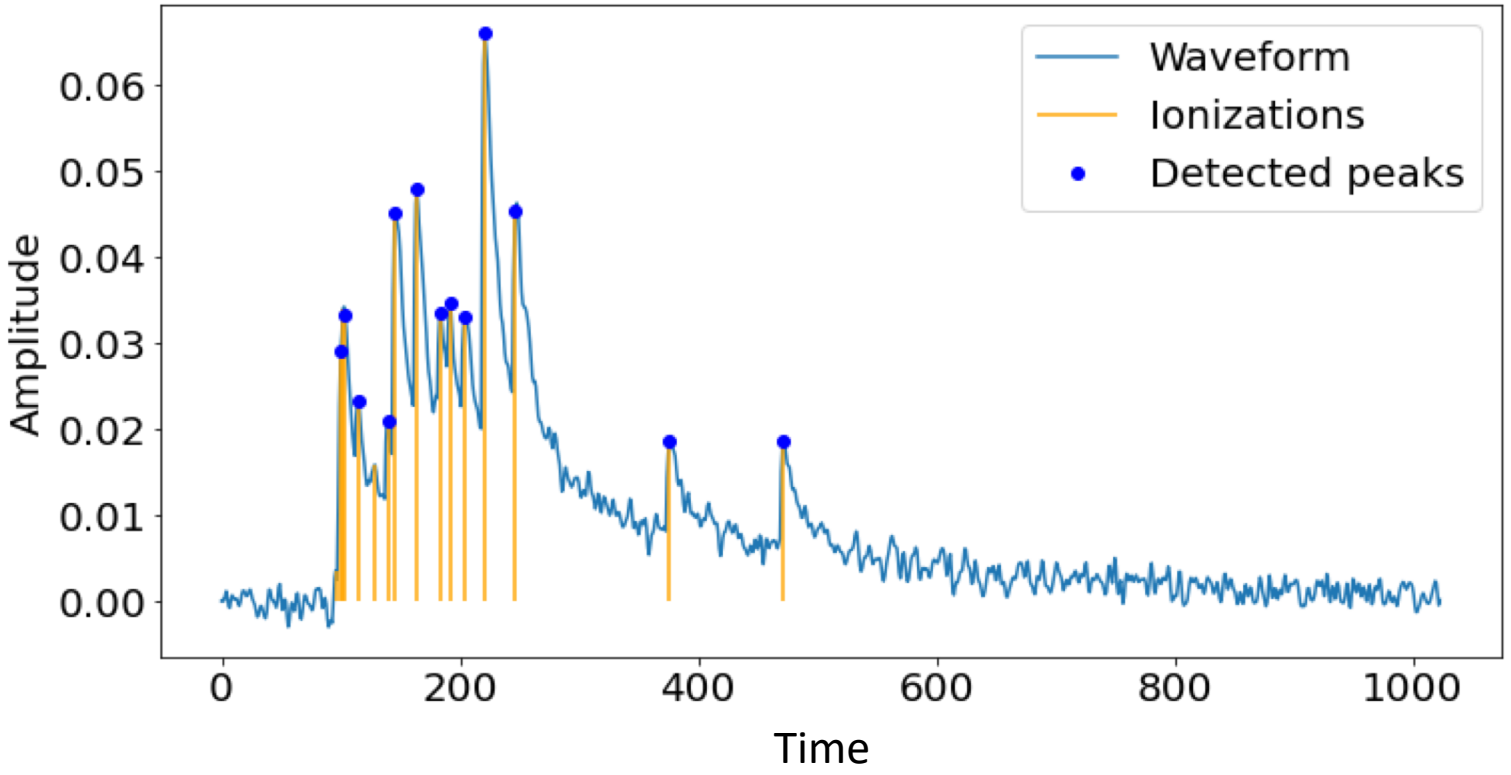
Model evaluation

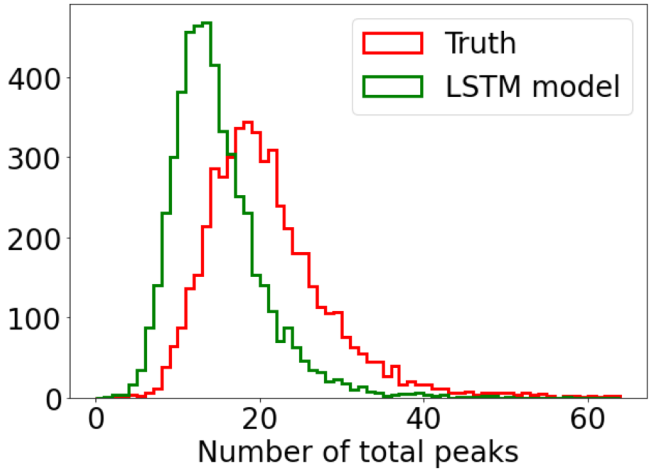


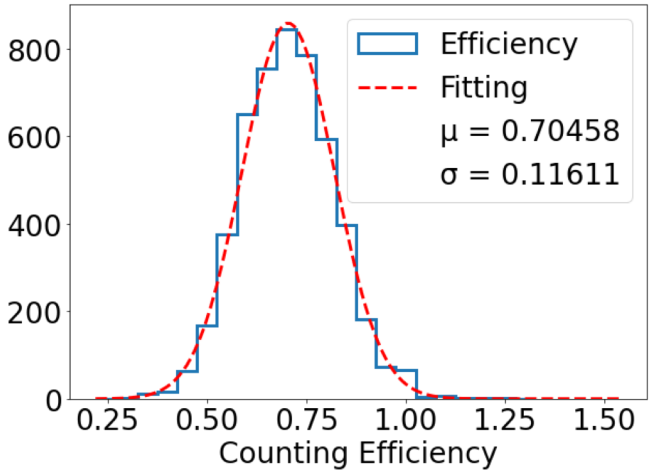
- Purity(Precision) = 0.9820 = TP/(TP+FP)
- Efficiency(Recall) = 0.6860 = TP/(TP+FN)
- False Positive Rate = 0.0005 = FP/(FP+TN)

		Prediction	
		Sig	Noise
Truth	Sig	TP	FN
	Noise	FP	TN

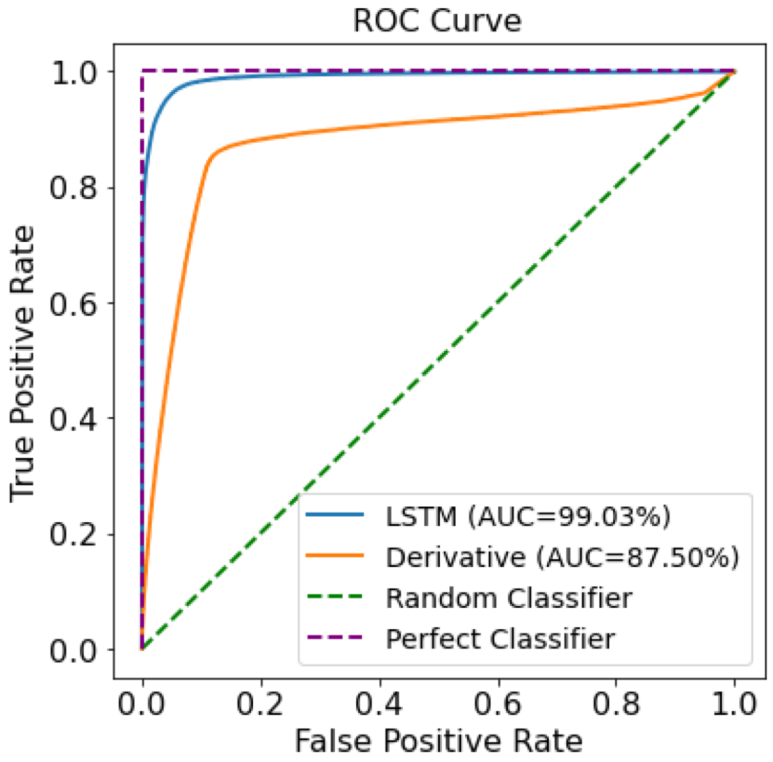
Evaluation by waveforms





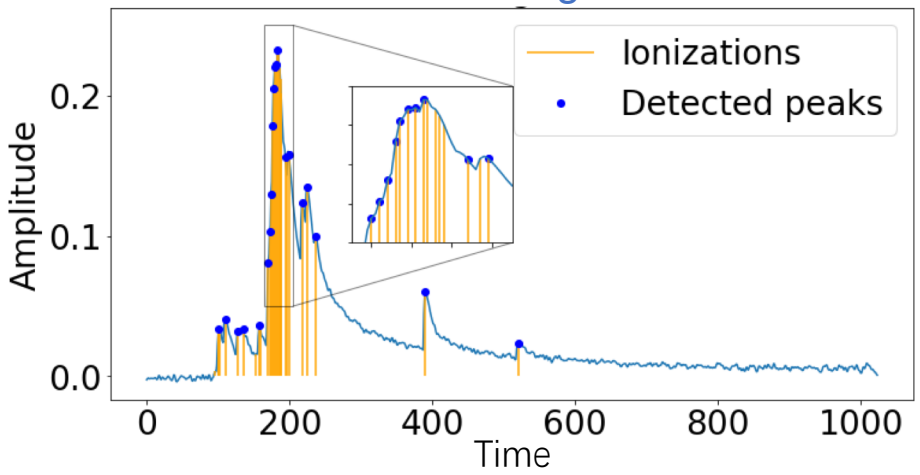


Comparison between LSTM and derivative model

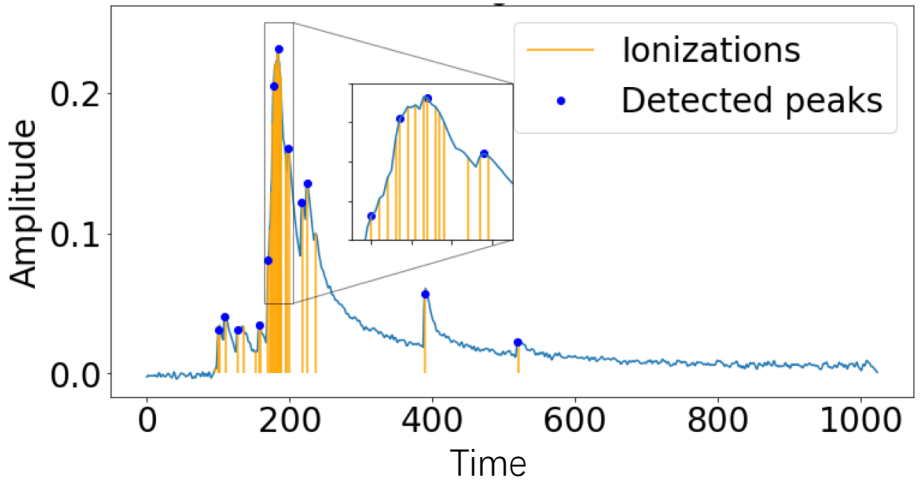


Better AUC for LSTM, due to the better pile-up recovery ability of the LSTM model

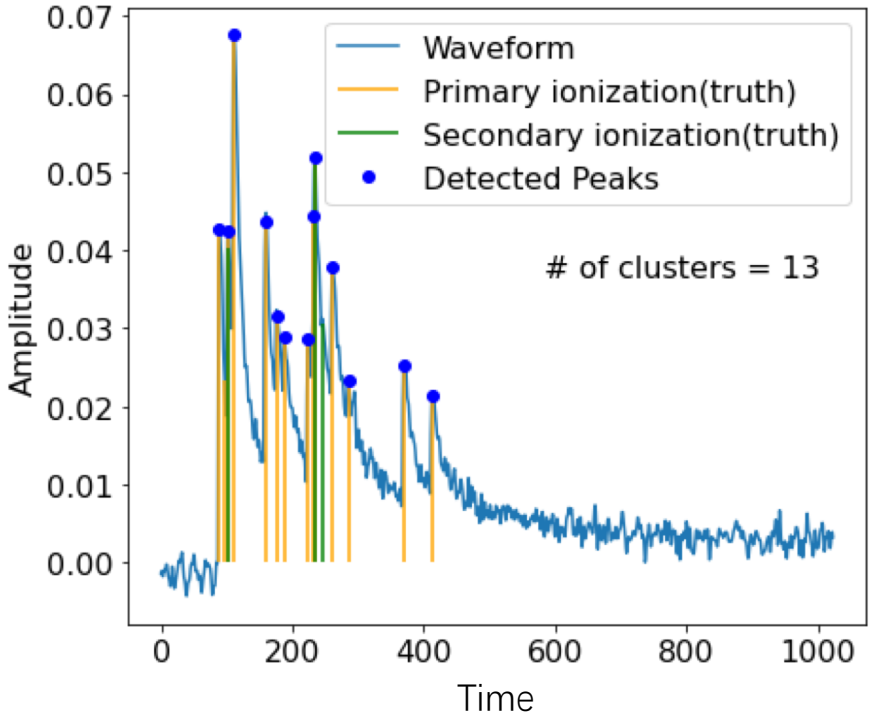
LSTM algorithm



Derivative algorithm

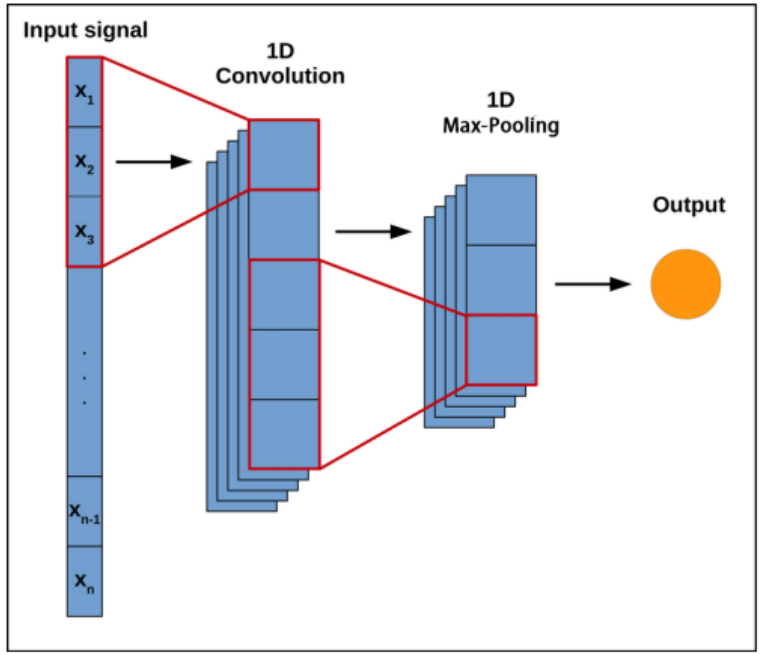


Step2: Clusterization



- A regression problem to predict N_{cls}
- The peaks found by peak finding algorithm would be training sample of this algorithm

CNN (Convolutional Neural Network)

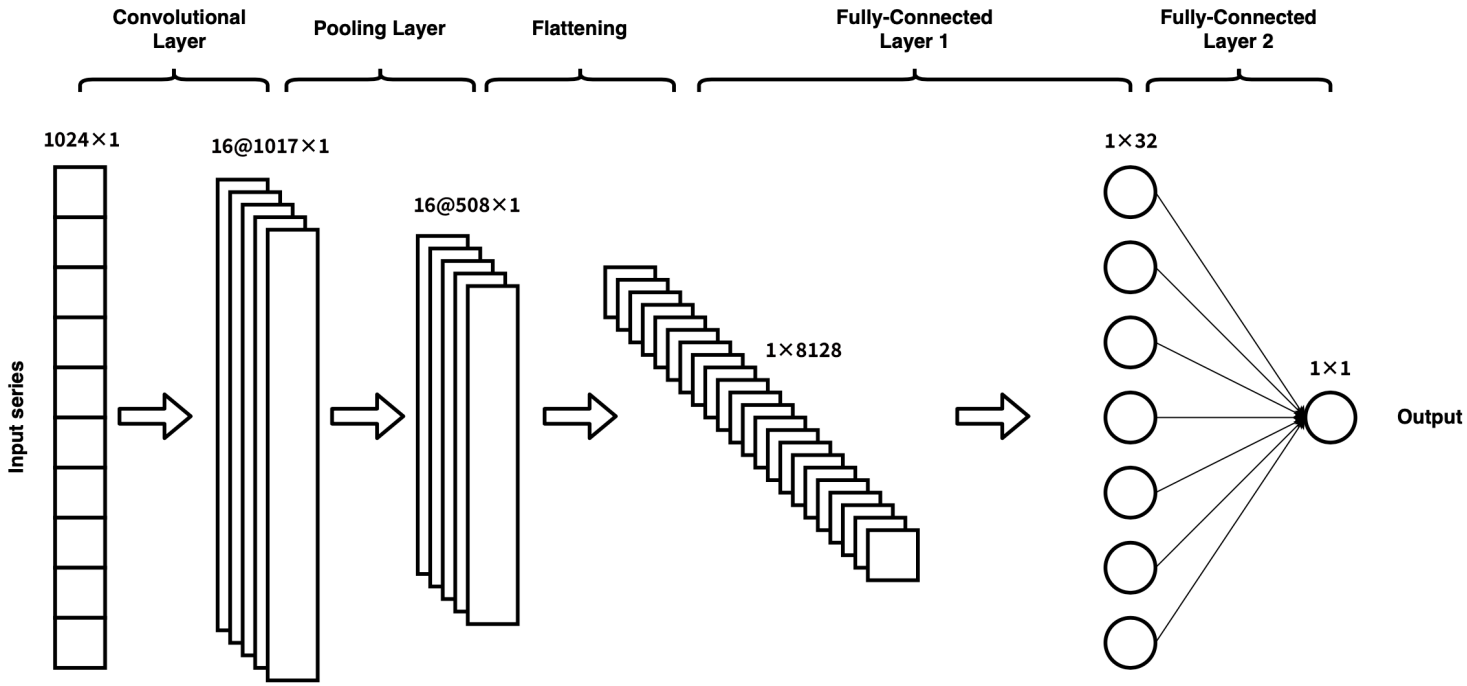


- Extracting features form local input patches
- 1D CNN can handle sequence data.

Network structure

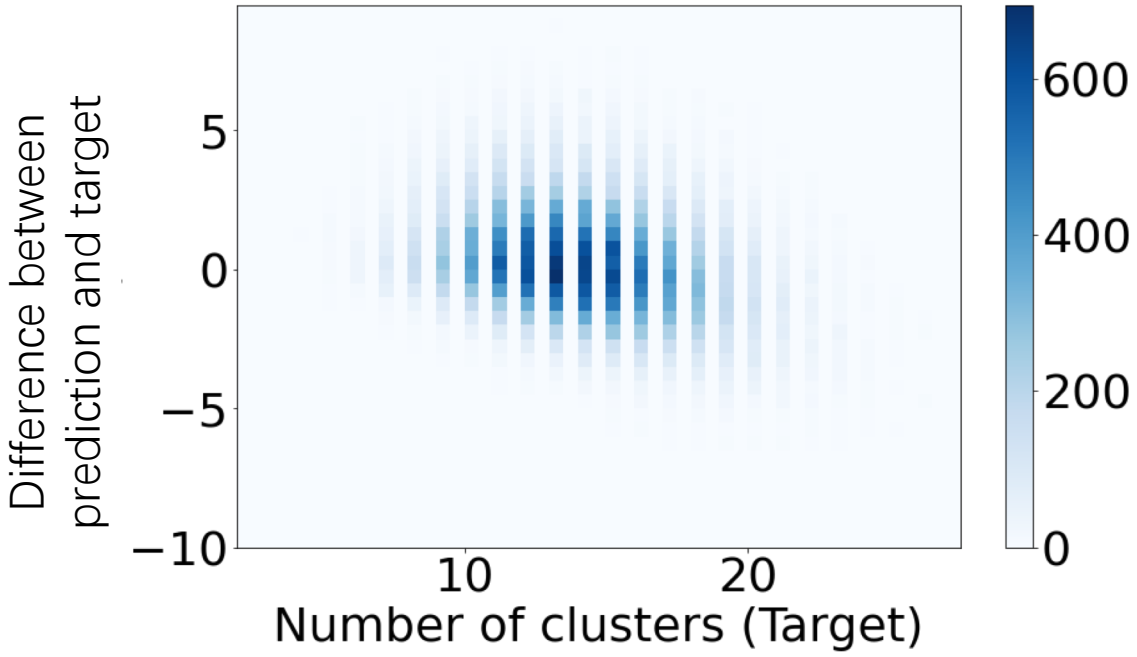
- Labels: Number of clusters from MC truth
- Features: Time list of the detected times in the previous step. Encoding in an (1024, 1) array.

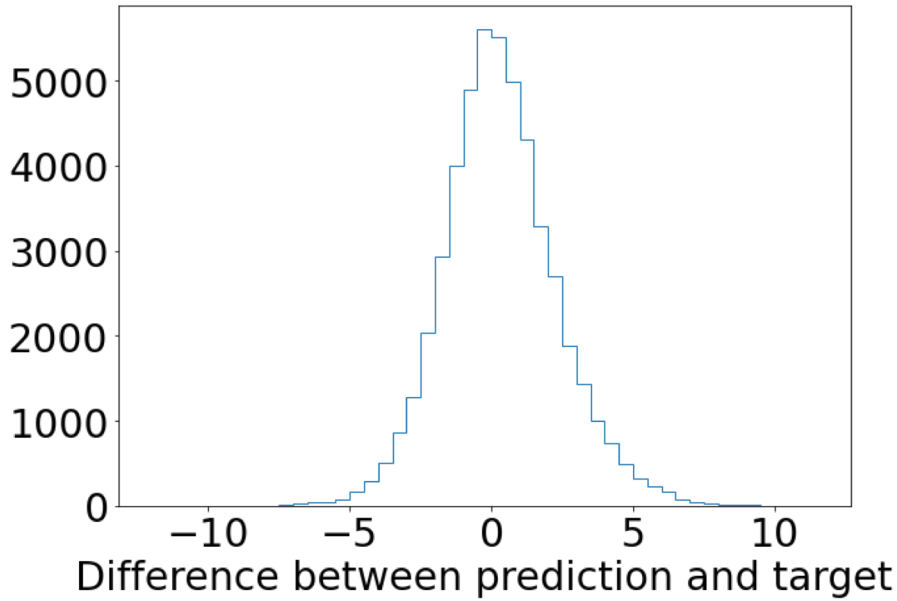
⇒ A regression problem



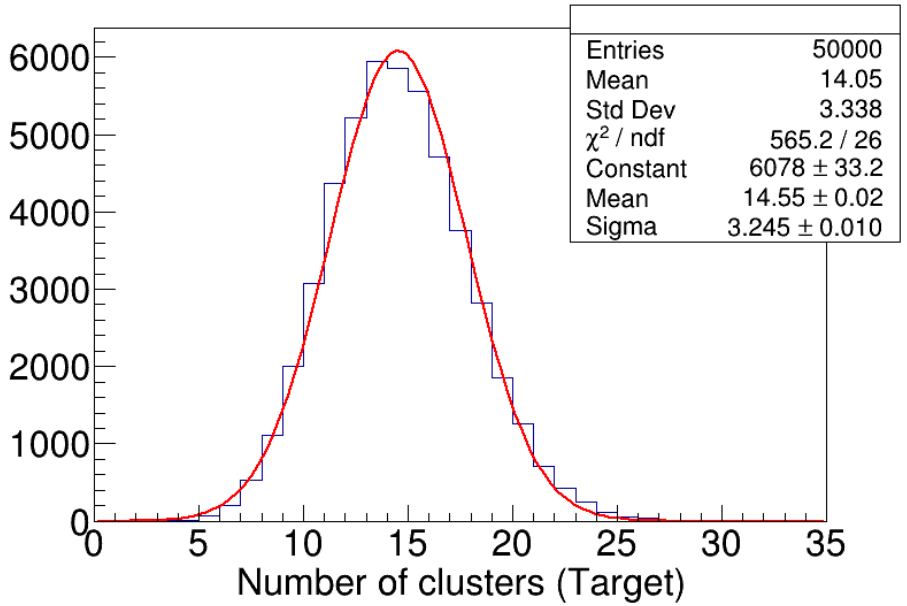
Model evaluation

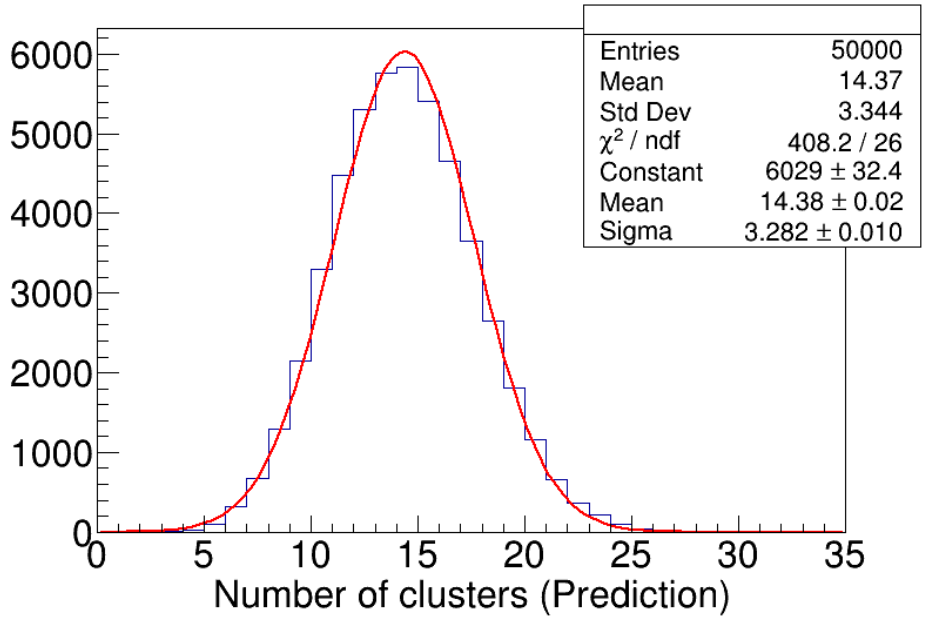
- Test using samples with mean number of ionizations fixed
- The difference between predicted and true values is small and stable





Final results of the reconstruction

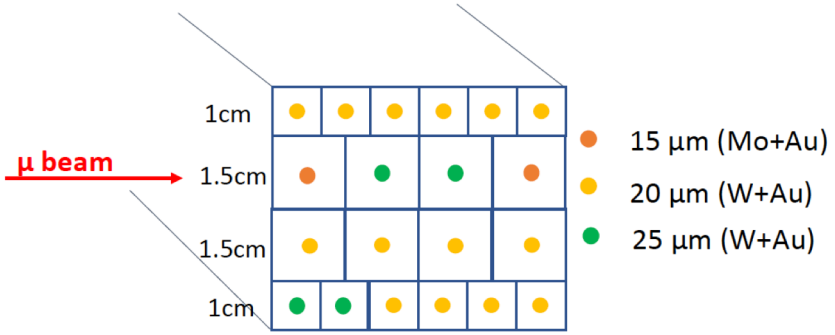


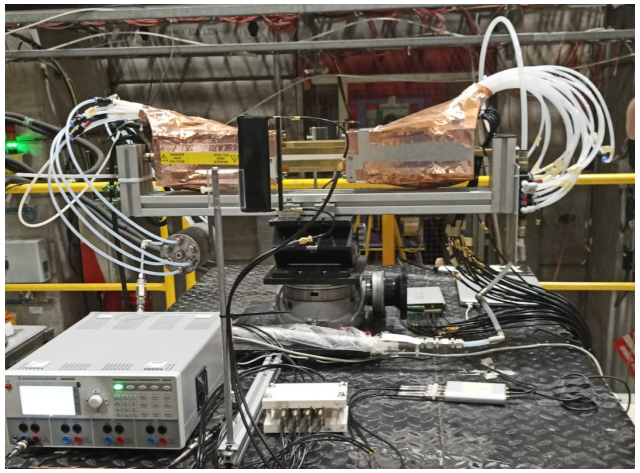


- Single cell resolution (σ/μ) ~ 22.8% (22.3% in truth)
- Good Gaussian distribution
- The relative error is quite similar to the truth value, which implies stable efficiency

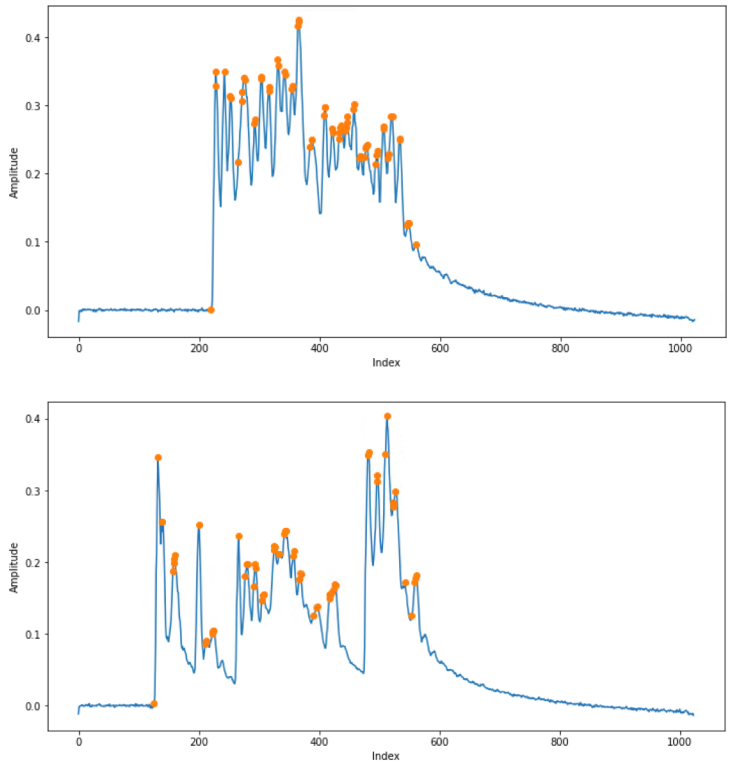
Applying NN on beam test data

- Beam tests organized by INFN group
- Cooperation between INFN and IHEP on data analysis is ongoing





Preliminary results of peak finding



- Clusterization under optimization

Summary and plan

- A two-step deep-learning-based cluster counting algorithm for drift chambers is developed
  - The peak finding algorithm shows better signal purity and efficiency than derivative algorithm
  - The clusterization algorithm gives Gaussian distributed N_{cls}
  - By applying the algorithms, single cell resolution is close to the MC truth level
  - Preliminary result with beam test data seems good
- Next
  - Optimization with beam test data
  - Implementation of the algorithm on online FPGA

Thanks!

Backup

Dataset

- **Simulated waveforms:**
  - The total # of ionizations distribution is flat: [1, 40]
  - 2,500 waveforms/point, 1,000,000 waveforms in total.