Machine Learning for Cluster Counting at the Edge in Future Drift Chambers

Online Data Compression

Liangyu Wu, Deniz Yilmaz, Julia Gonski
SLAC National Accelerator Laboratory

16 Sep 2025

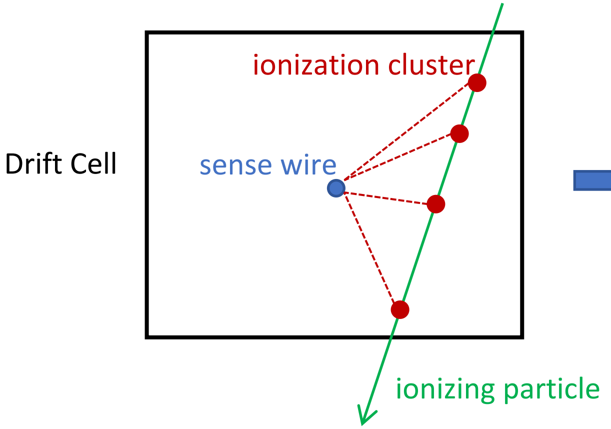
9th IDEA Study Group Meeting

Part 1

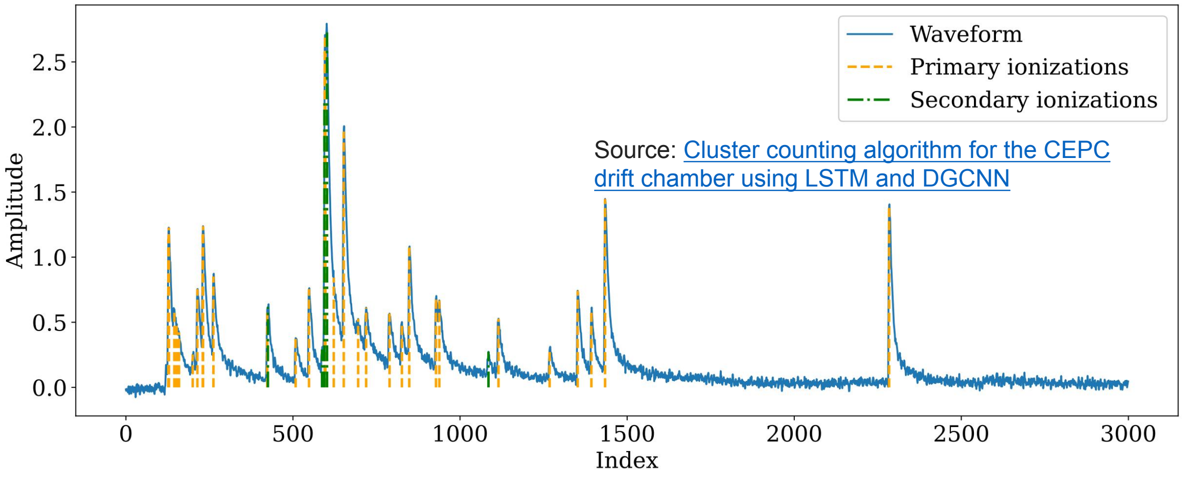
Introduction & Data Rate Challenge

Cluster Counting in the Future Drift Chamber

➢ Cluster Counting Technique



Fast Read-out



- 2000 ns with 1.5 GHz sampling rate
- Cell size: 18 mm x 18 mm

❑ Counting dN_{cl}/dx for PID

- Better and Robust Resolution

$$\boxed{\frac{\sigma_{dN_{cl}/dx}}{(dN_{cl}/dx)} = (\delta_{cl} \cdot L_{track})^{-1/2} = N_{cl}^{-1/2}}$$

90%He+10%iC_4H_{10}
δ_{cl} = 12.5 cm^{-1}
+
L_{track} = 2 m

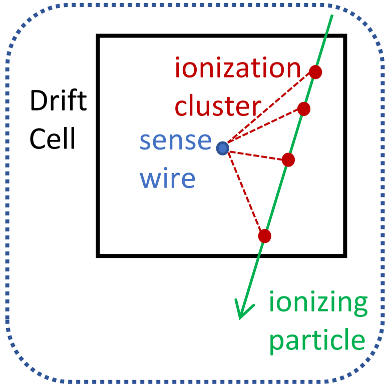
σ ≈ 2.0% (typical 4.3% for dE/dx)

Need an algorithm capable of effectively counting #clusters from the waveform!

Significant Data Rate Challenge

- Extreme Data Rates in Future Drift Chambers
  - High granularity designs are required to meet the stringent tracking and PID performance demands
  - Unprecedented challenges will arise with off-detector data rates above O(TB/s)
  - High data rates require significant power and pose challenges
    - Additional cooling needed for heat management
    - Increased energy use reduces budget for services like cooling and cabling

- Our Solution: ML-based compression at source



Analysis at the Edge →

One Single Number (#Clusters)

+ Possibly more ...

- Sufficient data reduction: O(TB/s) → O(GB/s)
- Requires an algorithm that meets both High accuracy & Low latency
- Deployable in front-end readouts

Part 2

Baseline Algorithms

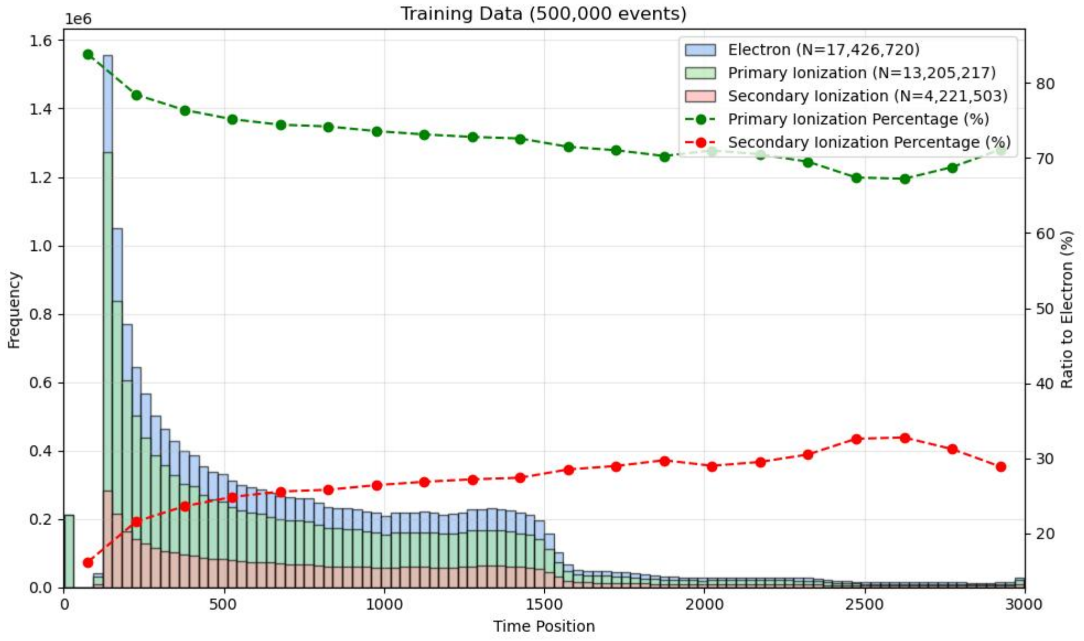
Basic Inspection of the Dataset

➢ Dataset Overview

Table 1. Summary of data sets used for training and testing ML-based cluster-counting algorithms.

Purpose	Algorithm	Particle	Number of Events	Momentum (GeV/c)
Training	peak-finding	$\pi^{\pm}$	5×10^5	$0.2 - 20.0$
Testing	peak-finding	$\pi^{\pm}$	5×10^5	$0.2 - 20.0$
Training	Clusterization	$\pi^{\pm}$	5×10^5	$0.2 - 20.0$
Testing	Clusterization	$\pi^{\pm}$	$1 \times 10^5 \times 7$	$5.0/7.5/10.0/12.5/15.0/17.5/20.0$
Testing	Clusterization	$K^{\pm}$	$1 \times 10^5 \times 7$	$5.0/7.5/10.0/12.5/15.0/17.5/20.0$

Source: [Cluster counting algorithm for the CEPC drift chamber using LSTM and DGCNN]



Event i	Shape	Note
‘wf_i’ (pulse shape)	(1, 3000)	2,000 ns time window (1.5 GHz)
‘mom’ (Momentum)	(1, 1)	0.2-20.0 [GeV/c]
‘tag_times’ (x-value)	(1, 300)	0 - 2999 (int)
’tag_values’ (peak info)	(1, 300)	0 - Background 1 - Primary ionization 2 - Secondary ionization

Several questions about this simulation dataset

- ❖ Why does the percentage of secondary ionization increase over time?
- ❖ What are the few ionizations occurring after 1000 ns?
- ❖ How can the drift velocity be inferred from the maximum drift time observed (approximately 1000 ns here)?

Baseline Algorithms for Cluster Counting

- Main Idea

Peak Finding Algorithm + Cluster Counting Algorithm

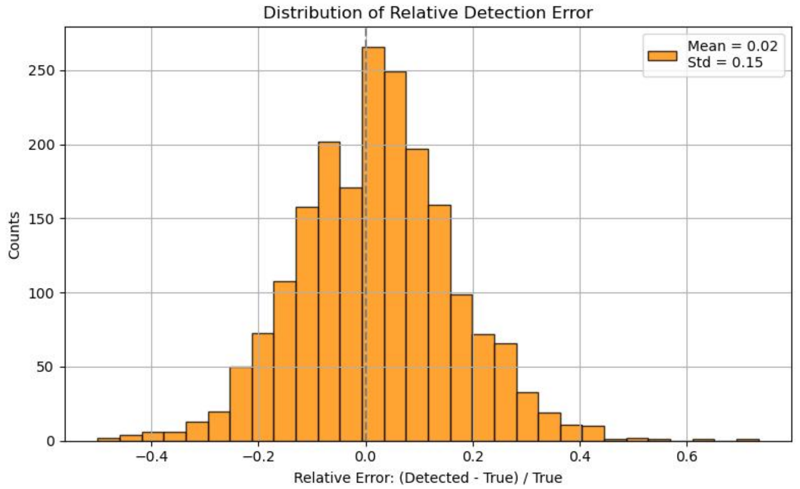
(Identify peak candidates) (Merge peaks into clusters)

D2 Algorithm

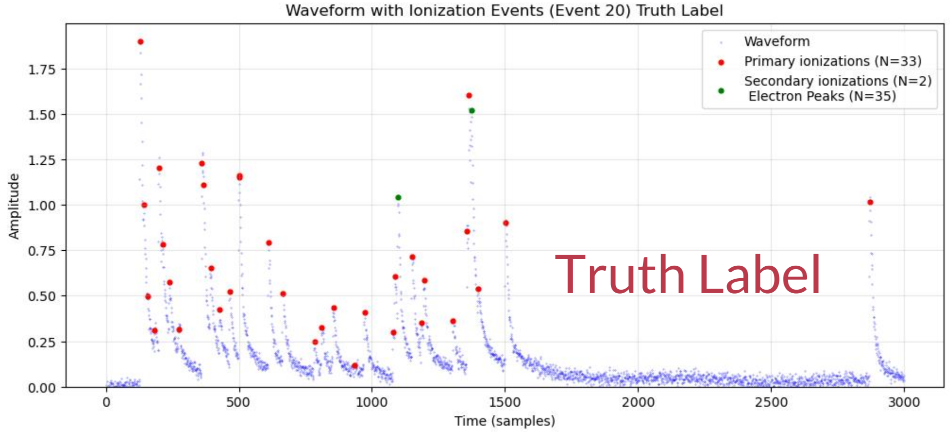
- Threshold-based derivative approach
  - T_1 (Amplitude threshold to suppress noise)
  - T_2 (2nd-derivative threshold for peak confirmation)

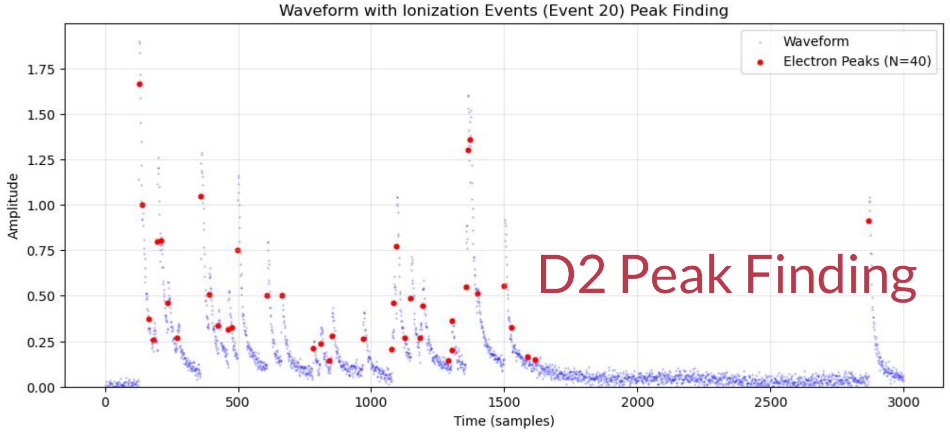
  Requires experimentation
- Return a list of detected peaks

Mean Percent Error 2%



Event Display





Baseline Algorithms for Cluster Counting

- Main Idea

Peak Finding Algorithm + Cluster Counting Algorithm

(Identify peak candidates) (Merge peaks into clusters)

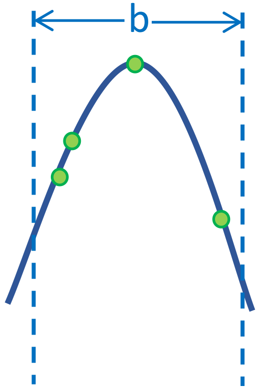
D2 Algorithm

Peak-finding

return lists of peaks

Fixed-Clusterization Algorithm (FCA)

- Input
  - b: Number of units to look forward in time window
  - c: Clusterization factor
  - d: Maximum number of peaks in a window
- 3D scanning for parameters determination

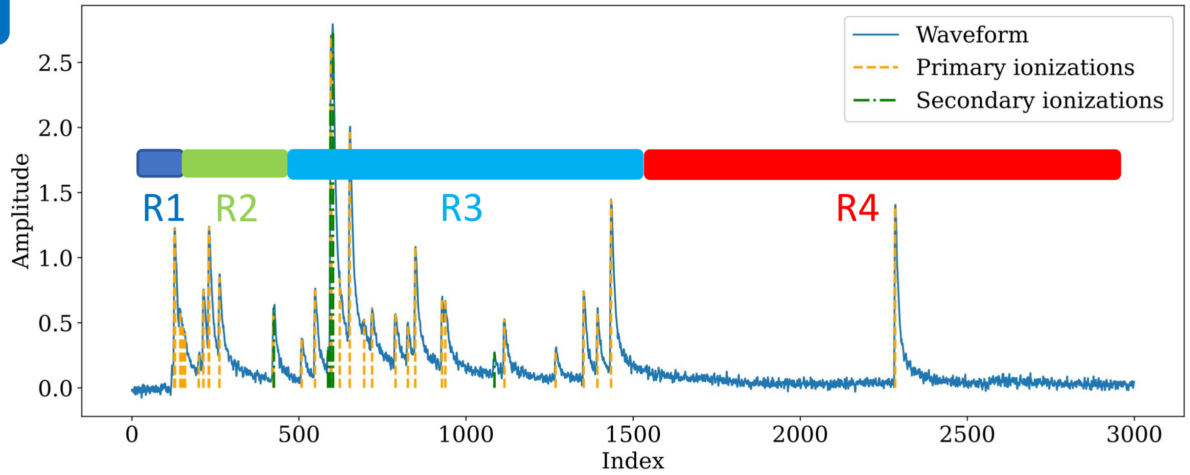


- #clusters = 4/c
- #clusters ≤ d

Adapted-Clusterization Algorithm (ACA)

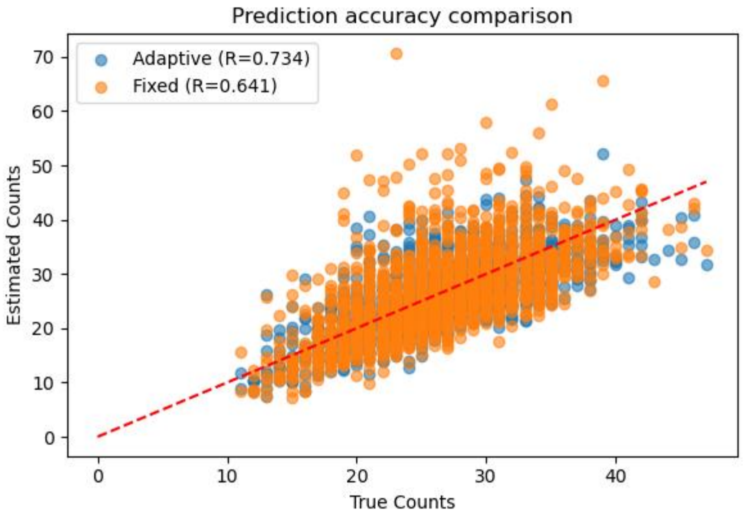
- Input:
  - b,c,d → b-map, c-map, d-map
- Scaning the parameters region by region:
  - R1: (0, 130)
  - R2: (130, 400)
  - R3: (400, 1550)
  - R4: (1550, 3000)

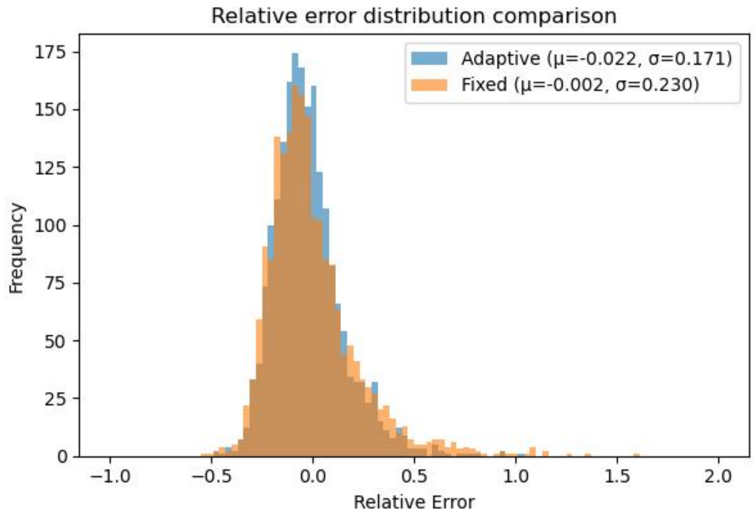
Requires experimentation



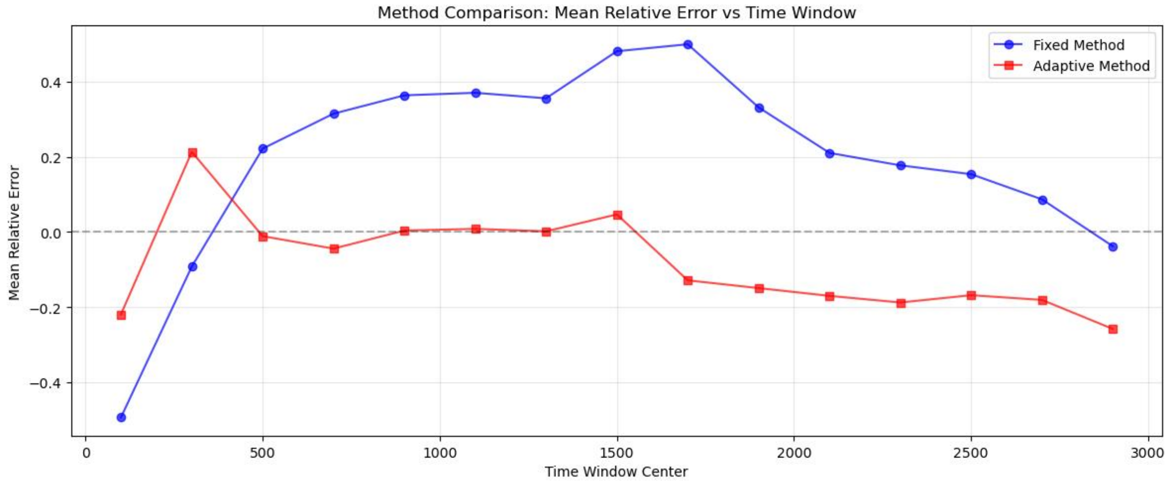
Baseline Algorithms for Cluster Counting

➤ Performance Comparison of ACA and FCA





- The ACA demonstrates higher prediction accuracy and resolution compared to the FCA



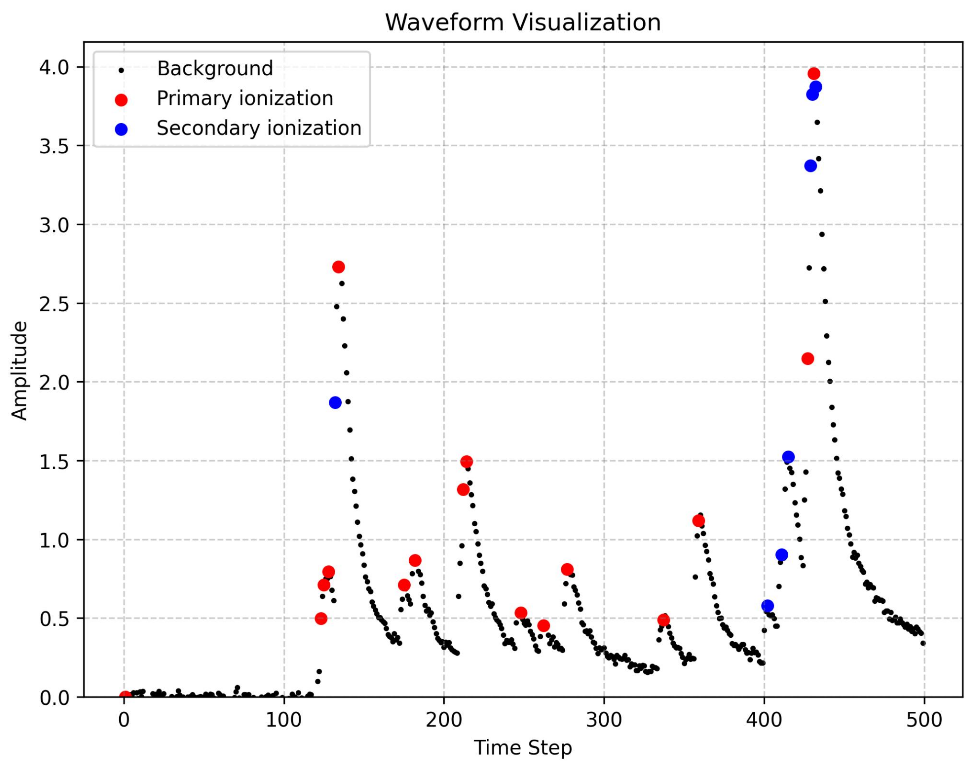
- ACA consistently outperforms FCA across the time scale
- "Learning" features from different time regions proves effective (suggesting that ML may yield better results)

Part 3

ML Algorithm Attempts

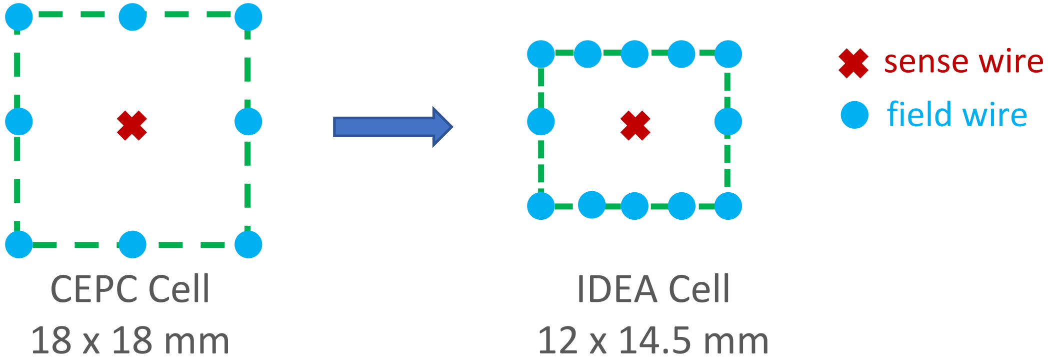
Truncated Waveforms as Input

➢ Sample Reduction



- 3,000→500 samples to emulate the reduction of the CEPC cell size
- Maintain the 1.5 GHz sampling rate
- Corresponding to a maximum drift time of ~350 ns
  (Aligns with IDEA DCH of 300-600 ns)
- Track length analysis
  - 12.5 clusters per waveform after truncation
  - 9.3 mm track length per cell (derived from 12.6 clusters/cm)

Assuming v_{drift} = 2 cm/μs, the cell radius will be roughly 6.7 mm (after truncation)



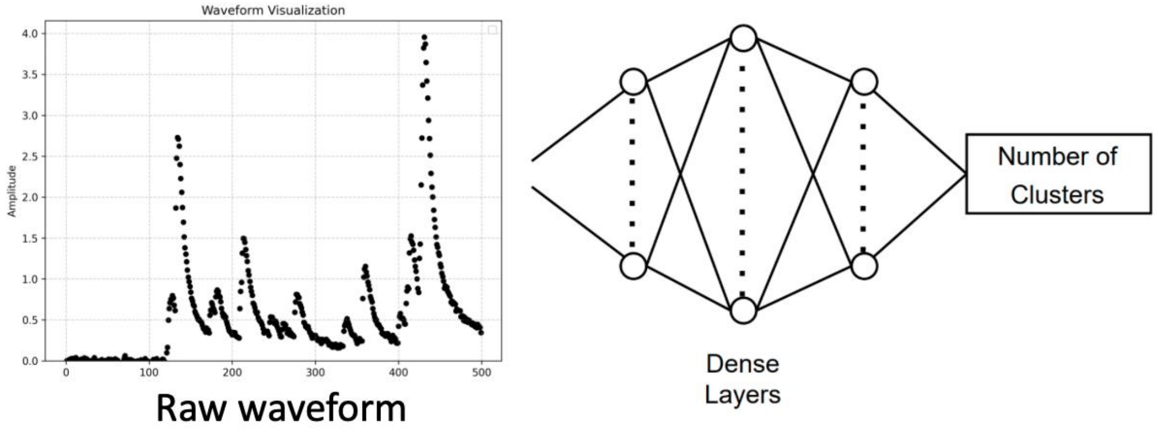
ML-Based Cluster Counting Paradigms

- Configurations
  - Classical (Non-ML) Strategy
    - D2 + FCA
    - D2 + ACA

Peak finding algorithm followed by a clusterization algorithm

  - ML Strategy
    - DNN direct regressor

Directly predict #clusters for each waveform



On-Chip DNN Regressor

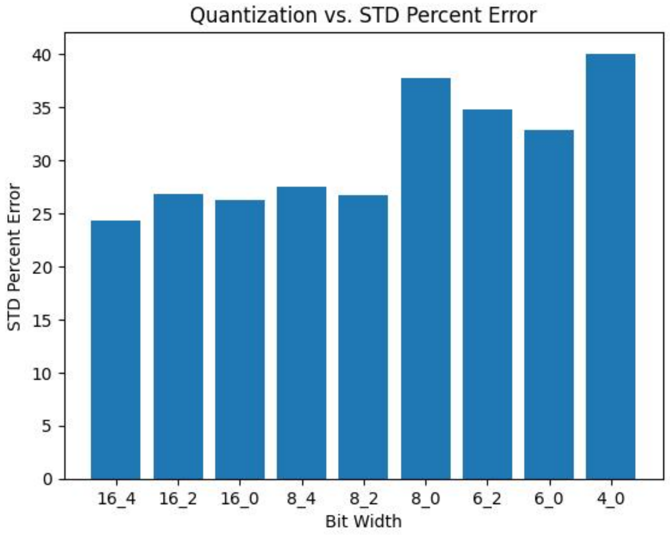
Compression Strategies for Edge Deployment

- Model Quantization and Pruning
  - Goal: To develop a model that meets **strict resource and latency requirements** for real-time processing

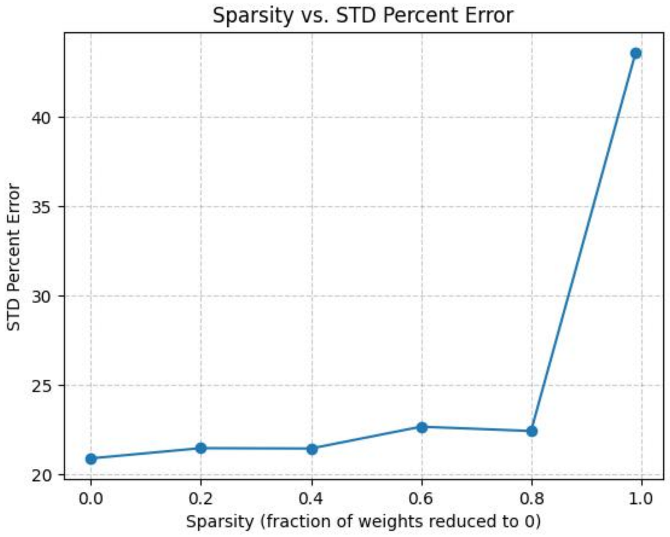
ML Model Training → **Quantization** → **Pruning**

- 400k for training
- 100k for validation
- 100k for testing

Lowering the precision of model weights and activations

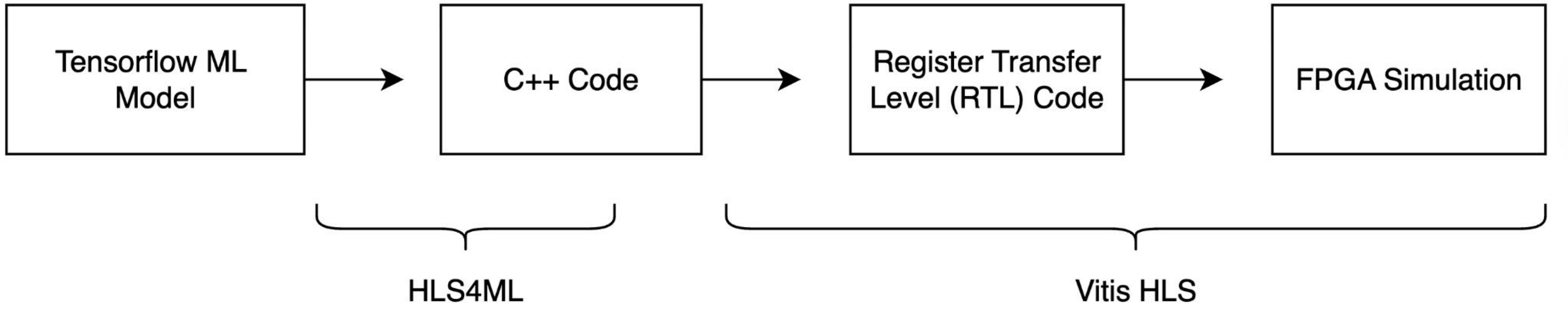


Eliminating model weights and activations that have minimal impact on predictions



Hls4ml for Model Evaluation on FPGAs

➢ Workflow for Model Evaluation



hls4ml [Documentation](#)

- Translate trained models into HLS code for FPGA implementation
- Configuration
  - Reuse factor = 1
  - io_parallel

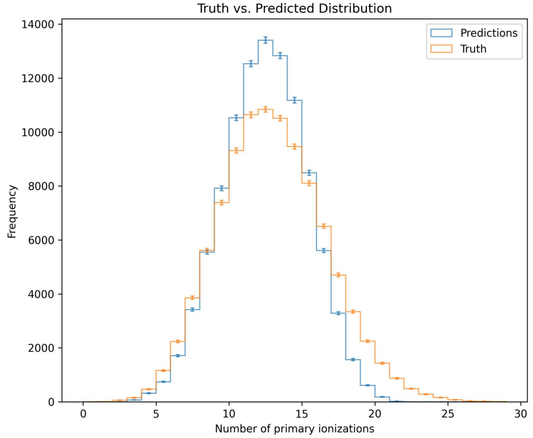
➢ Evaluation Report

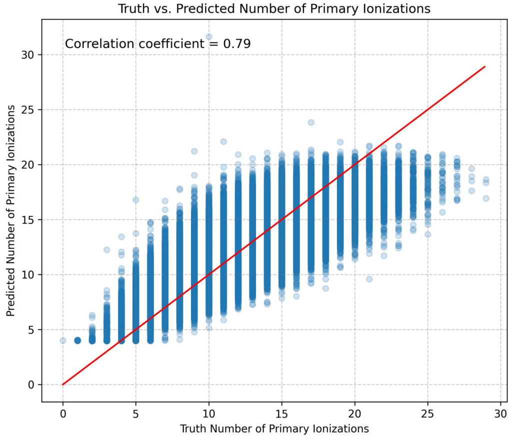
Model	Latency [ns]	LUTs	FFs	DSPs
DNN 8-32-8	55	183,726	44,946	3,399
HLS4ML Quantized <10,5> DNN 8-32-8	55	83,417	22,029	39
HLS4ML 60% Pruned; Quantized to <10,5> DNN	45	127,818	26,002	19

- Latency of 50 ns enables real-time applications in future colliders with O(10) ns bunch crossing rates

Model Performance Summary

- Baseline DNN Model
  - Without QAT or pruning

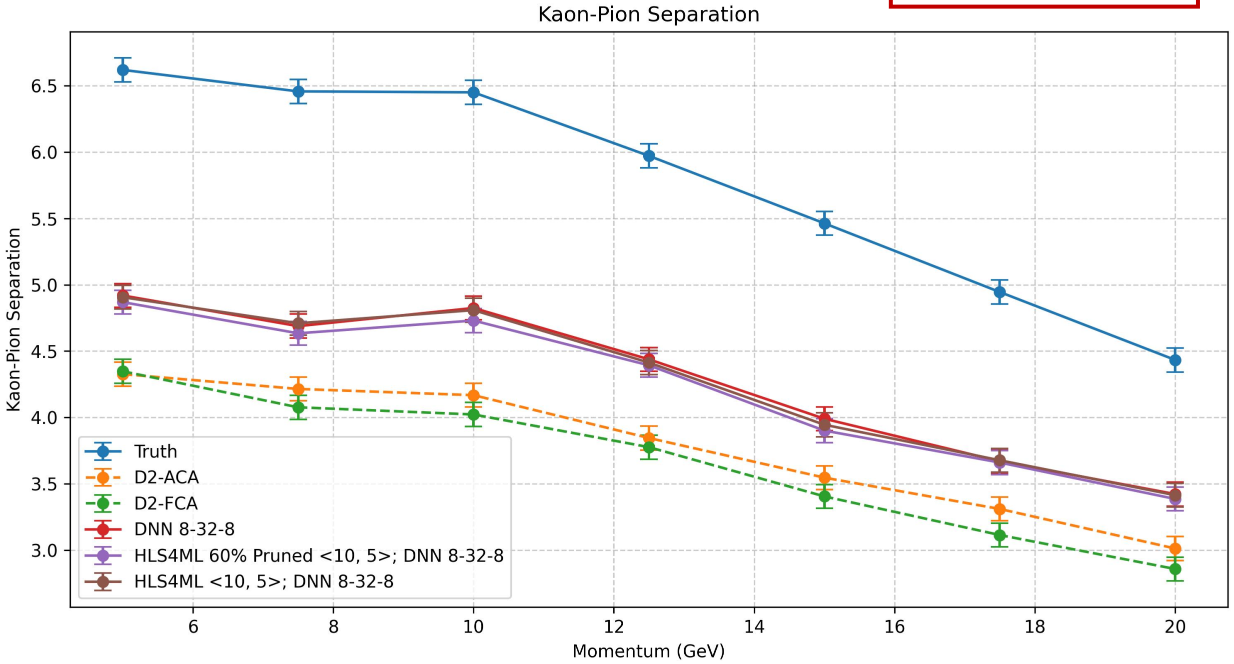




- Overall, the model's predictions closely match the truth labels, with some deviations at the peaks
- The 2D plot of predictions vs. truth **shows a strong correlation**

- K/π-Separation Summary

$$S = \frac{\left| \frac{dN}{dx}_{\pi} - \frac{dN}{dx}_{K} \right|}{\frac{\sigma_{\pi}+\sigma_{K}}{2}}$$



- All the DNN models outperform classical algorithms across all tested ranges
- Under a track length of 2 m, **all ML models achieve K/π separation powers exceeding 3σ**
- Compared to the truth (the best ideal limit), there remains significant potential for ML performance improvement

Future Directions

- Additional Model Variants
  - Autoencoder-based compression with off-detector decoding
  - On-chip split-DNN
  - VAE-based anomaly detection
  - ...
- More Realistic IDEA DCH Simulation
  - More detector details, electronic parameters, and transfer functions ...
  - Further exploration of ML performance under different gas fill conditions
  - ...
- Power Consumption Evaluation of Chip Designs
  - Ensuring compatibility with future drift chamber specifications

Summary

- Future drift chambers will face the challenge of handling TB/s data rates. Machine learning applied at the source for cluster counting shows significant potential for real-time data reduction in next-generation drift chambers.

- ML-based cluster counting techniques have been demonstrated, showing improved pion-kaon separation performance compared to traditional methods.

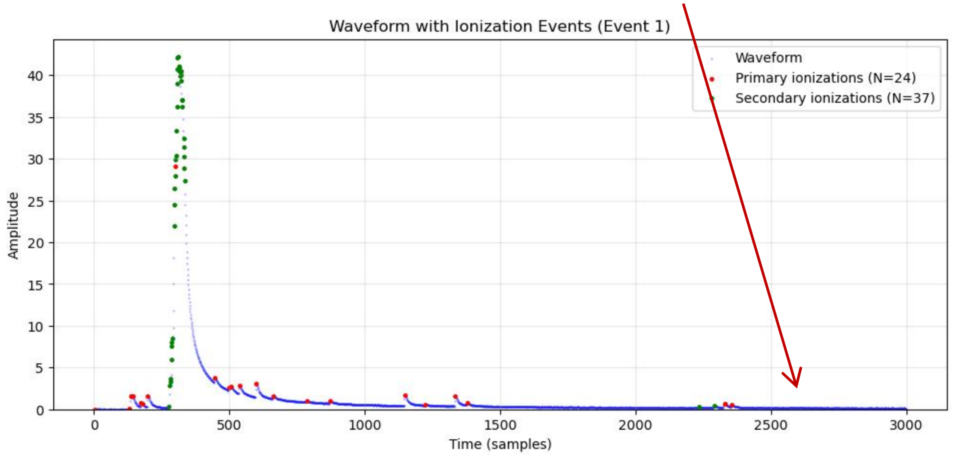
- Compressed ML models have been shown to be viable for deployment in front-end readout ASICs. Findings from FPGA synthesis studies validate this potential, demonstrating effective performance and resource utilization.

- Accurate Garfield++ simulations that closely represent the IDEA drift chamber are essential for advancing further research into ML applications and detector design optimization in the future.

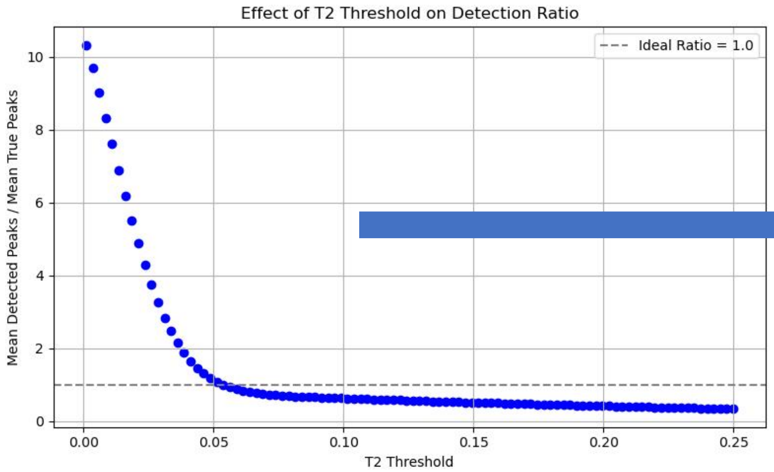
Back ups

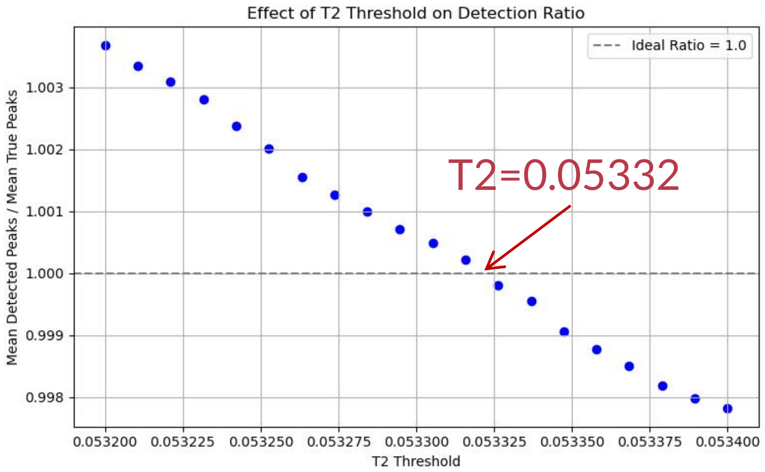
D2 Threshold Determination

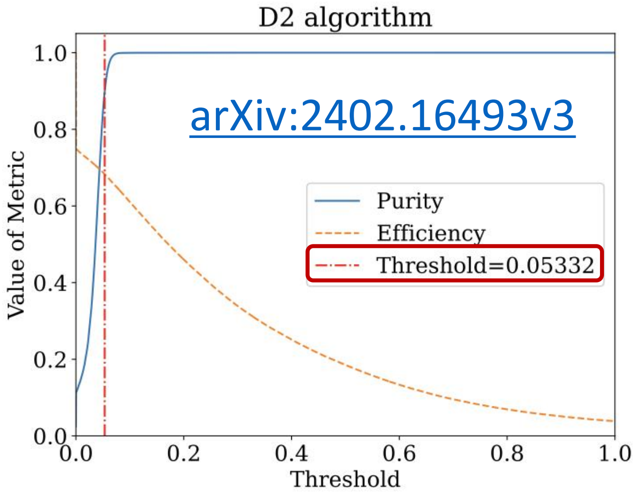
- T1: 3x RMS of the average noise amplitudes



- T2: Accuracy vs. T2 Threshold Scanning



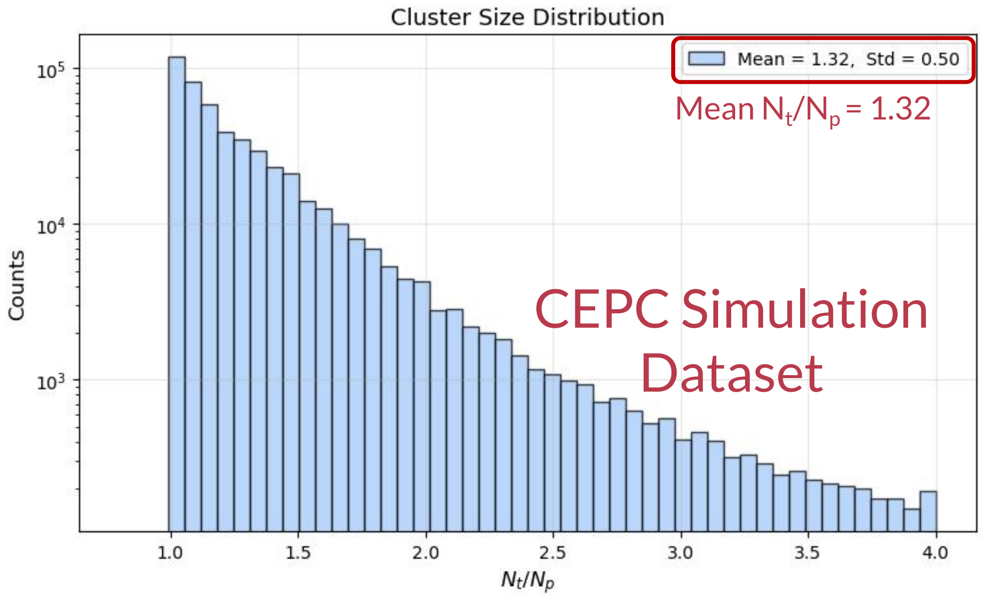




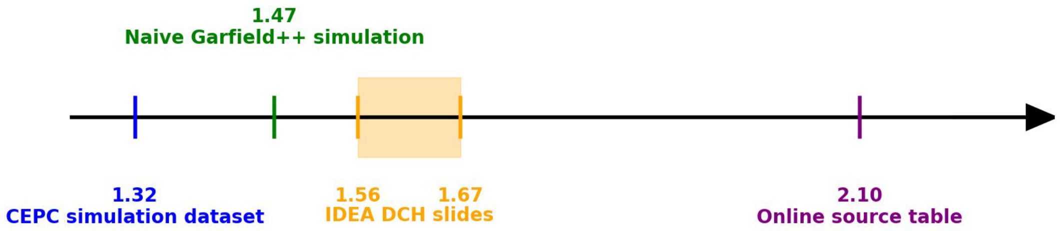
Perfect match!

Basic Inspection of the Dataset

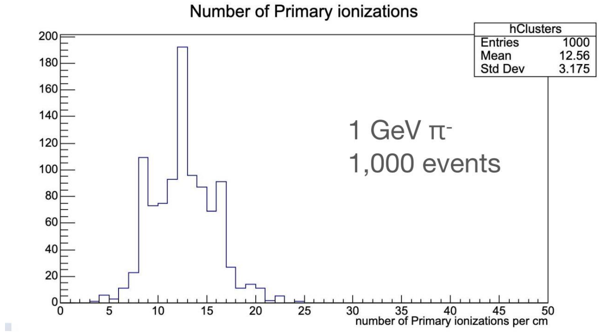
➢ Puzzles about the Cluster Size?

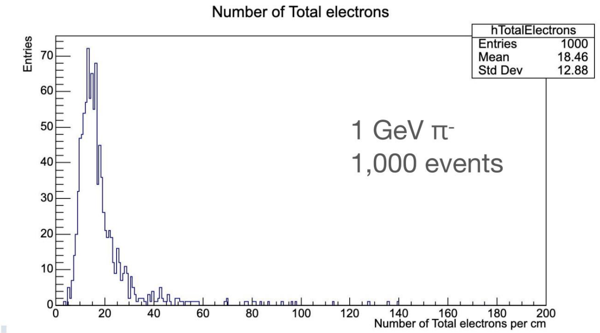


❑ Cluster Size of Helium-isobutane Mixture



- Naive 3x3 cell array Garfield++ simulation [18.46/12.56 = **1.47**]





- Online source (NTP, MIPs) [26.7 / 12.7 = **2.10**]

Gas	Density *10^{-3} (g/cm^3)	N_p (cm^{-1})	N_t(cm^{-1})
He-iC_4H_{10} (90-10)	0.42	12.7	26.7

- Dr. Gravili's slides on IDEA DCH 2024 (Page 14)

Results for cluster size distribution (∼ 1.61), in reasonable agreement overall:

- GARFIELD++: ∼ 1.56
- Test beam analysis: ∼ 1.67
- He experimental measurements: ∼ 1.6

Basic Inspection of the Dataset

➢ K/π-Separation Power Defination?

❑ Version A

- Source: Cluster counting algorithm for the CEPC drift chamber using LSTM and DGCNN

The K/π-separation power is defined as

$$S = \frac{\left|\left(\frac{dN}{dx}\right)_\pi - \left(\frac{dN}{dx}\right)_K\right|}{\left(\sigma_\pi + \sigma_K\right)/2},$$

- Source: Simulation of particle identification with the clustercounting technique

The separation power for two particles, labelled for simplicity p_1 and p_2, with different masses and same momentum, is evaluated with the relation (3.1) [1]:

$$n_{\sigma_E} = \frac{\Delta_{p_1} - \Delta_{p_2}}{< \sigma_{p_1,p_2} >} \tag{3.1}$$

where Δ_{p_1} and Δ_{p_2} are the measurements of the deposited energy, σ_E is the resolution in the ionization measurement (*energy resolution*) given by the variance of *Gaussian* distribution of the truncated mean values and $< \sigma_{p1,p2} >$ is the average of the two resolutions:

$$< \sigma_{p1,p2} > = \frac{\sigma_{E,p1} + \sigma_{E,p2}}{2} \tag{3.2}$$

❑ Version B

- Source: Production of charged pions, kaons and (anti-)protons in Pb-Pb and inelastic pp collisions at $\sqrt{s_{NN}}$ = 5.02 TeV

TOF and HMPID. The separation power is defined as follows:

$$Sep_{(\pi,\mathrm{K})} = \frac{\Delta_{\pi,\mathrm{K}}}{\sigma_\pi} = \frac{|\langle signal\rangle_\pi - \langle signal\rangle_\mathrm{K}|}{\sigma_\pi}; \quad Sep_{(\mathrm{K,p})} = \frac{\Delta_{\mathrm{K,p}}}{\sigma_\mathrm{K}} = \frac{|\langle signal\rangle_\mathrm{K} - \langle signal\rangle_\mathrm{p}|}{\sigma_\mathrm{K}} \tag{3}$$

❑ Version C

- Source: Charged Hadron Identification with dE/dx and Time-of-Flight at Future Higgs Factories

resolution of 5 % or better. The separation power S is the relative distance between the Bethe-Bloch bands, defined as $S = |\mu_1 - \mu_2| / \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2}}$ with μ_i and σ_i being the mean and width of the band of particle i, respectively. Figure 3 shows the π/K and K/p separation power. $S > 3$ is achieved for particle momenta between about 2 and 20 GeV in the default detector model IDR-L.

❑ Version D

- Source: Detector Requirements Analysis on the Pion-Kaon Separation

$$S_{\pi K} = \sqrt{\frac{(I_\pi - I_K)^2}{\sigma_{I_\pi}^2 + \sigma_{I_K}^2} + \frac{(T_\pi - T_K)^2}{\sigma_{T_\pi}^2 + \sigma_{T_K}^2}}$$

Following Version A in this research, but may consider Version D in the future

Significant Data Rate Challenge

- Basic Parameters (e.g. IDEA DCH)
  - Number of sense wires: N_{wires} = 56,448
  - Sampling rate (assumed): f_s = 1.5 GHz
  - ADC resolution (assumed): 12 bits/sample
  - Maximum drift time (assumed): 500 ns
  - Read both ends of the wires

- Continuous Data Rate (Triggerless mode)

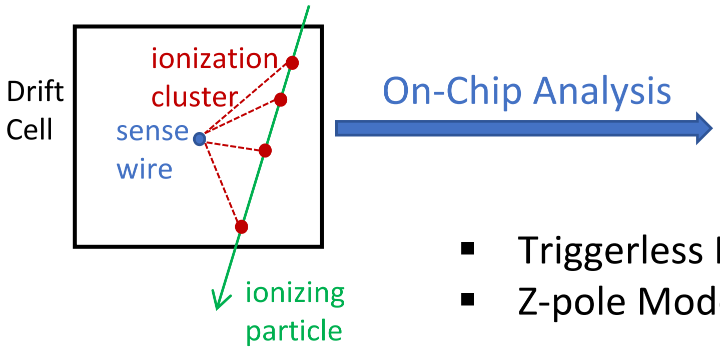
$$\begin{aligned} R_{\text{total}} &= N_{\text{wires}} \times f_s \times \text{bits per sample} \times 2 \\ &= 56,448 \times 1.5 \times 10^9 \times 12 \text{ bits/s} \times 2 \\ &= 2.03 \times 10^{15} \text{ bits/s} \\ &= 254 \text{ TB/s} \end{aligned}$$

- 100 kHz Trigger Data Rate (Z-pole mode)

$$\begin{aligned} R_{\text{triggered}} &= N_{\text{wires}} \times f_{\text{trigger}} \times t_{\text{drift}} \times f_s \times \text{bits per sample} \times 2 \\ &= 56,448 \times 10^5 \times 500 \times 10^{-9} \times 1.5 \times 10^9 \times 12 \text{ bits/s} \times 2 \\ &= 10.16 \times 10^{13} \text{ bits/s} \\ &= 12.7 \text{ TB/s} \end{aligned}$$

- TB/s is way too high!

  Our Solution: ML-based compression at source



One Single Number (#Clusters)

- Triggerless Mode: 254 TB/s → 125 GB/s
- Z-pole Mode: 12.7 TB/s → 6.25 GB/s

Clusterization Algorithms Summary

Note:
1. **True peaks** - Truth-level peak-finding results
2. **HIGH** - Peak-dense regions [150, 500)

Non-ML Results:

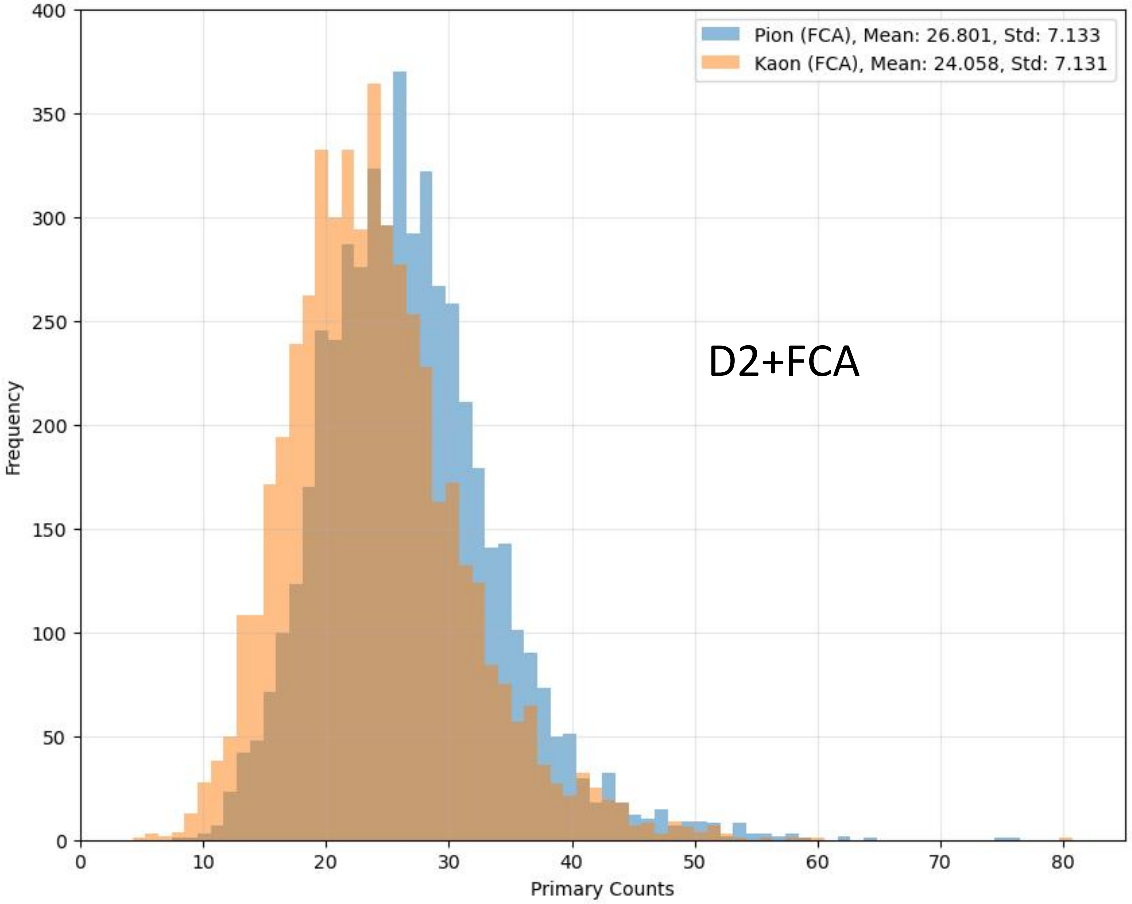
	D2 + FCA	D2 + ACA	D2/1.32	True peaks + FCA	True peaks + ACA	D2 + FCA (HIGH)	D2 + ACA (HIGH)
Mean Relative Error	0.1%	-2.2%	0.4%	-0.3%	-1.5%	-17.7%	4.5%
Std	0.231	0.171	0.291	0.135	0.116	0.233	0.291
Correlation	0.640	0.734	0.591	0.781	0.844	/	/

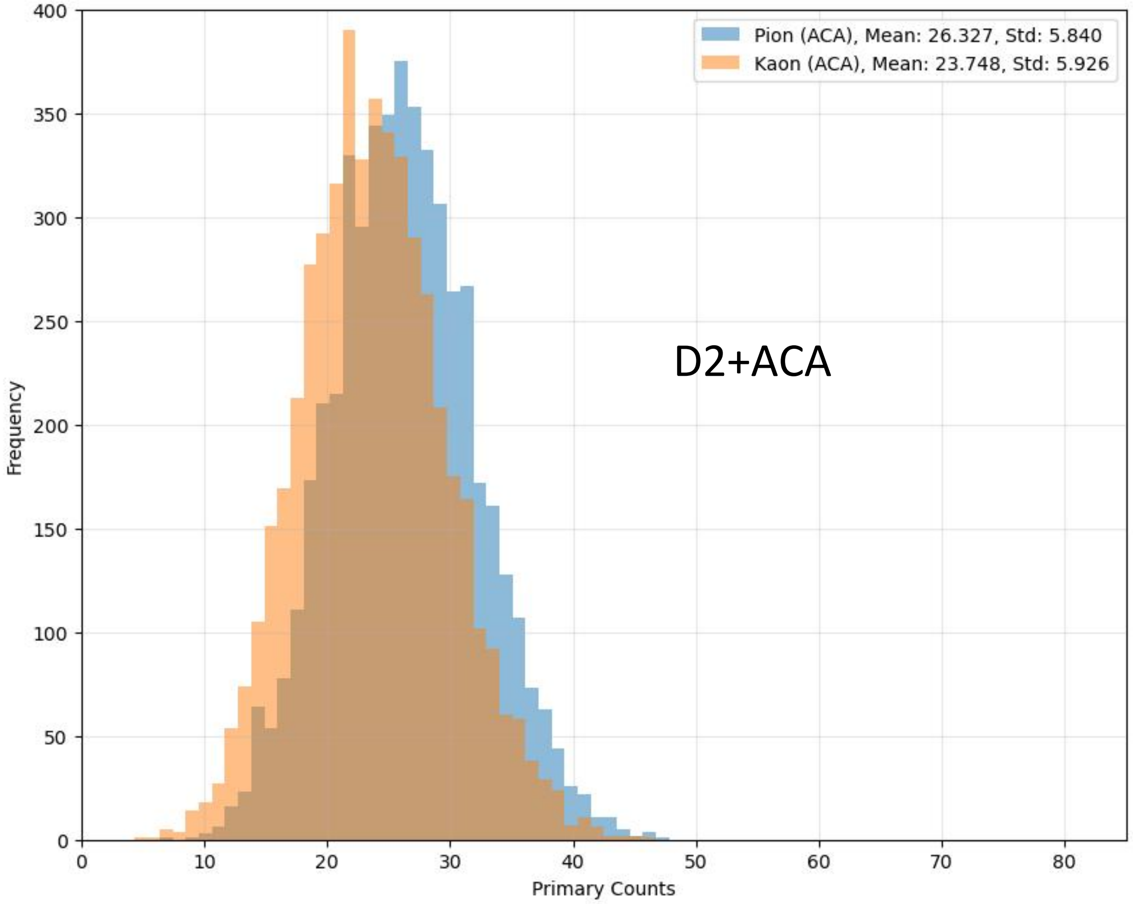
ML Results:

	Direct N-Regressor	CNN Non-Regressor
Mean Relative Error	-1.04%	7.71%
Std	0.113	0.169
Correlation	0.85	0.76

- The direct #cluster-regressor model seems to be promising!

10 GeV Momentum K/π Separation (unnormalized)

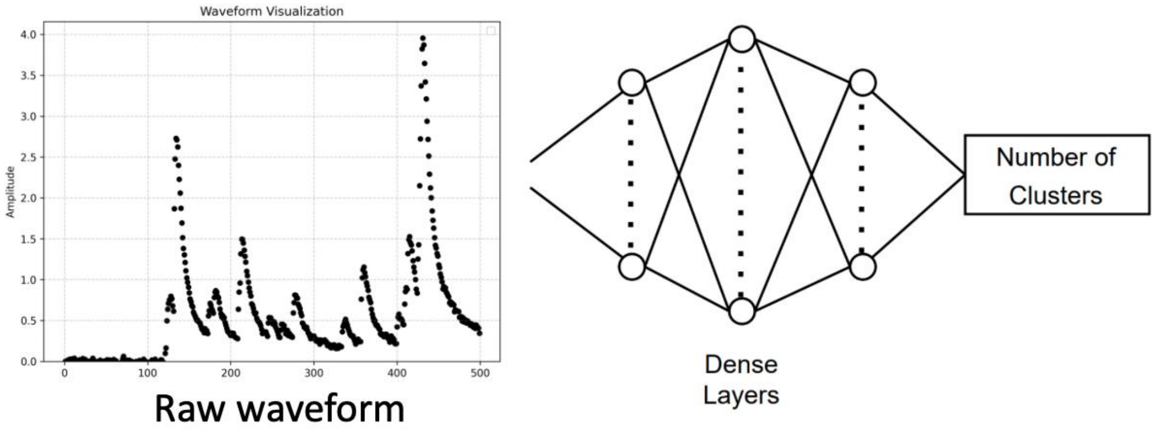




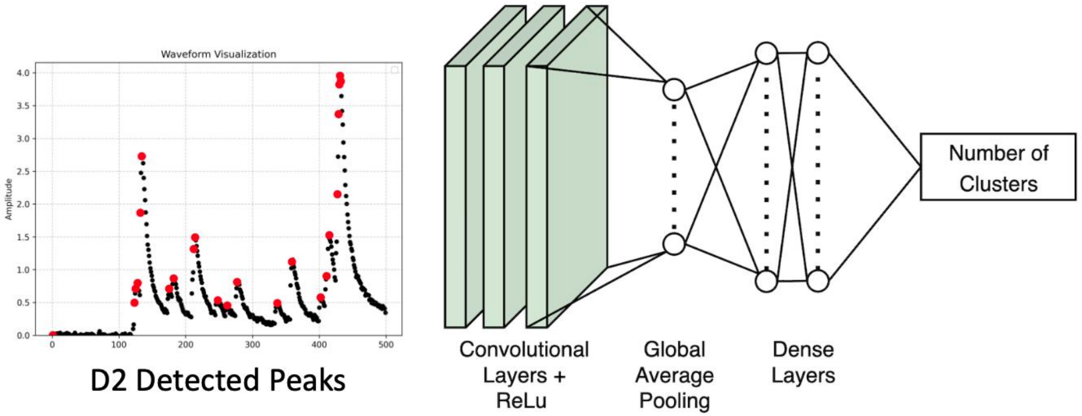
More ML-Based Cluster Counting Paradigms

- Flexible Configurations
  - Classical (Non-ML) Strategy
    - D2 + ACA/FCA
      [Peak-finding algorithm followed by clustering]
  - Pure ML Strategy
    - CNN direct regressor
      [Challenging to meet O(20 ns) latency requirement]
    - DNN direct regressor
  - Classical + ML Hybrid Strategy
    - D2 + CNN direct regressor
      [On-Chip PF + off detector ML cluster counting]

- On-Chip DNN Regressor

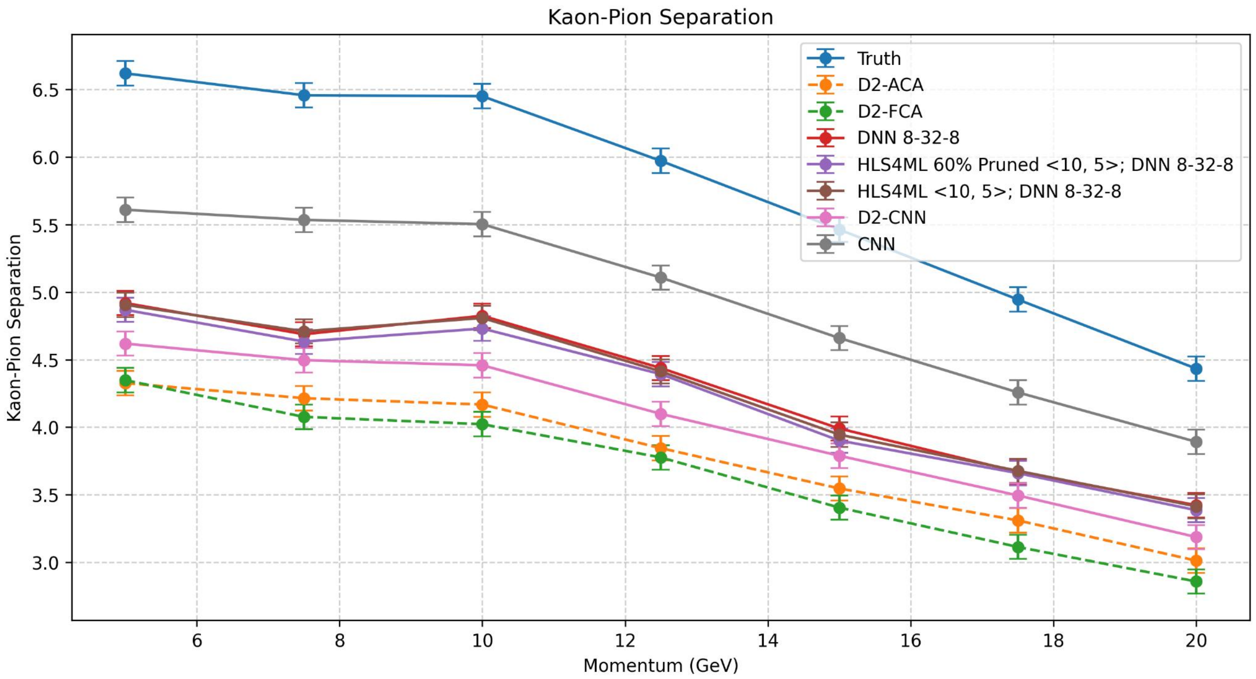


- D2 + CNN Regressor

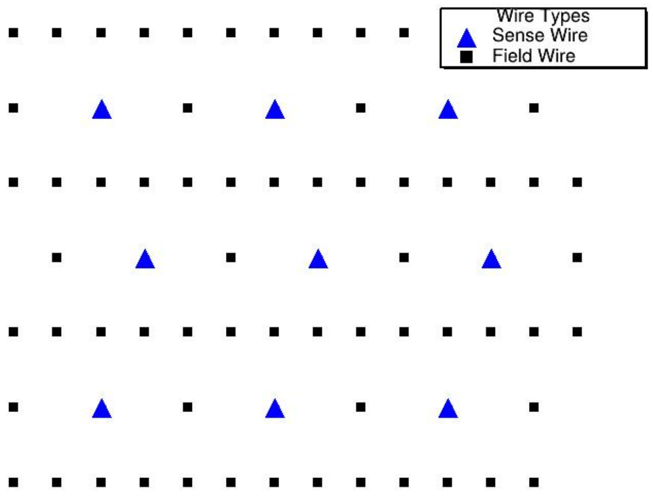


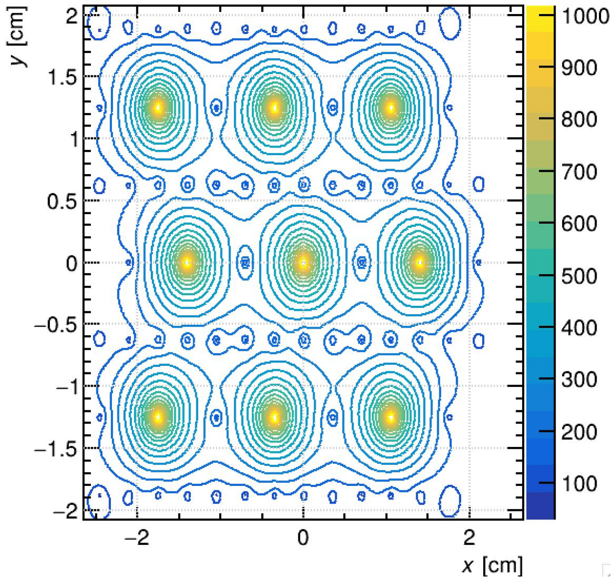
More ML-Based Cluster Counting Paradigms

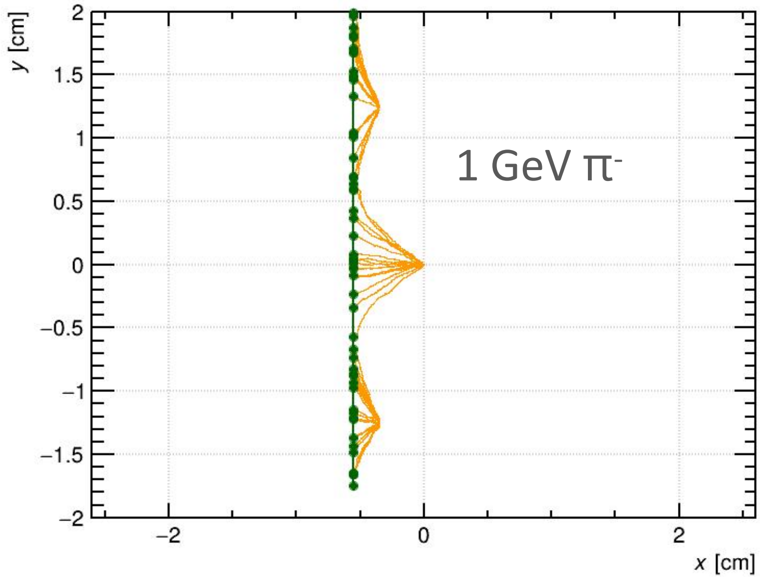
- Flexible Configurations
  - Classical (Non-ML) Strategy
    - D2 + ACA/FCA
      [Peak-finding algorithm followed by clustering]
  - Pure ML Strategy
    - CNN direct regressor
      [Challenging to meet O(20 ns) latency requirement]
    - DNN direct regressor
  - Classical + ML Hybrid Strategy
    - D2 + CNN direct regressor
      [On-Chip PF + off detector ML cluster counting]



IDEA DCH-like Cell Array Simulation







3x3 Cell Array

- 20 μm diameter sense wires
- 40 μm diameter field wires
- Cell X:Y = 14 mm: 12.5 mm

Potential Contours

- Hinge the layers together
- Cells offset by 1/4 cell between them
- No +/- stereo angles in this exercise

Event Display

- Ionization generated in each cell is properly drifted to the corresponding sense wire