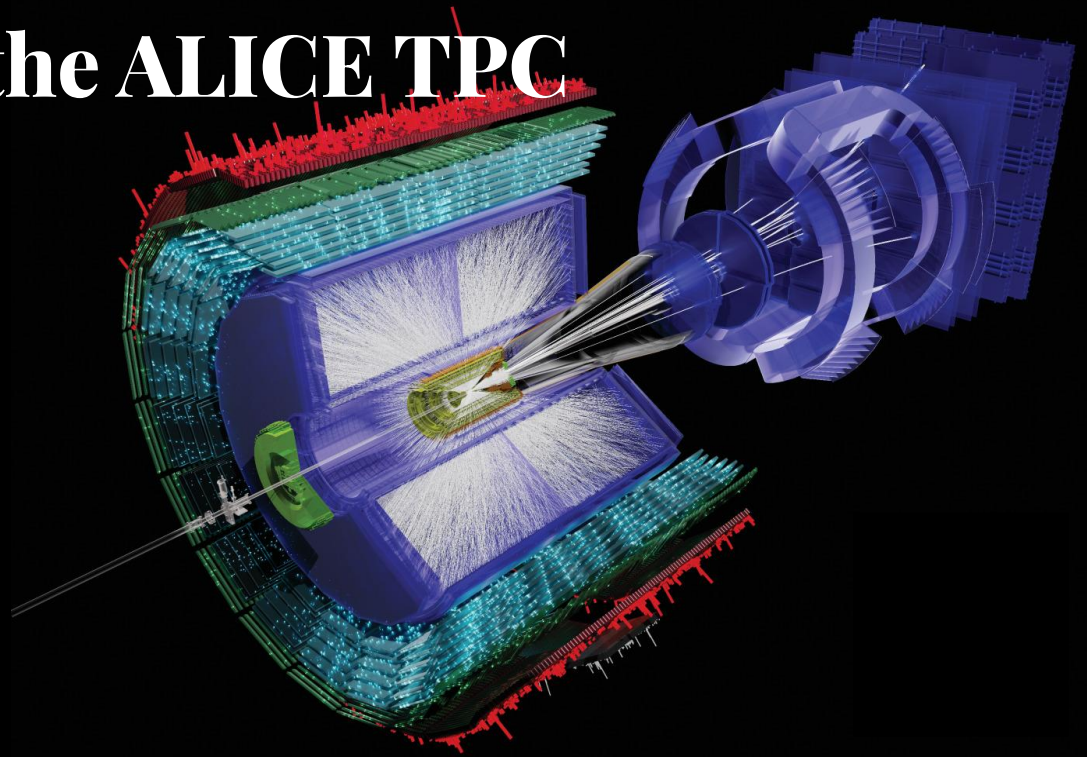
Machine learning based optimizations of the ALICE TPC d*E*/d*x* response

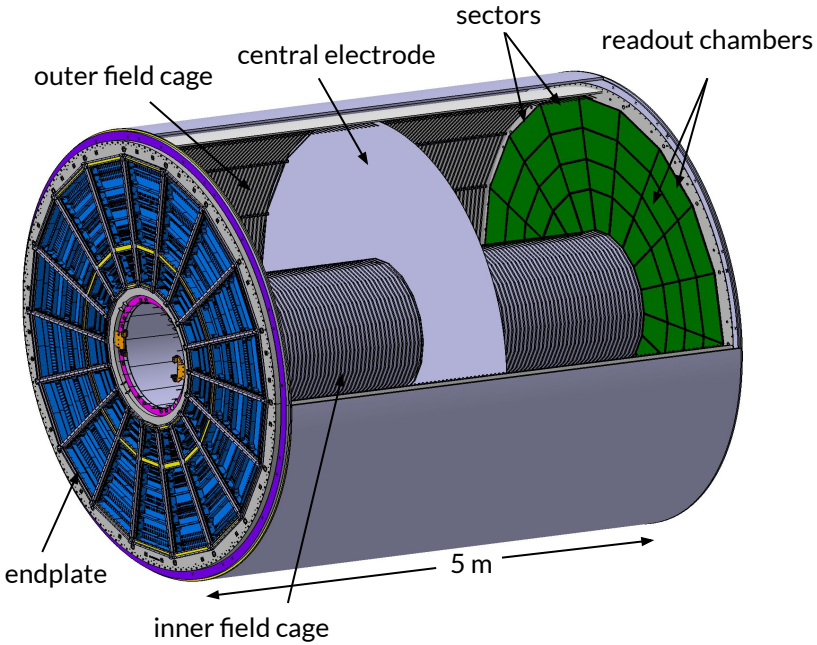


HADRON Conference 2023
Tuba Gündem
ALICE Collaboration
05.06.2023

Outline

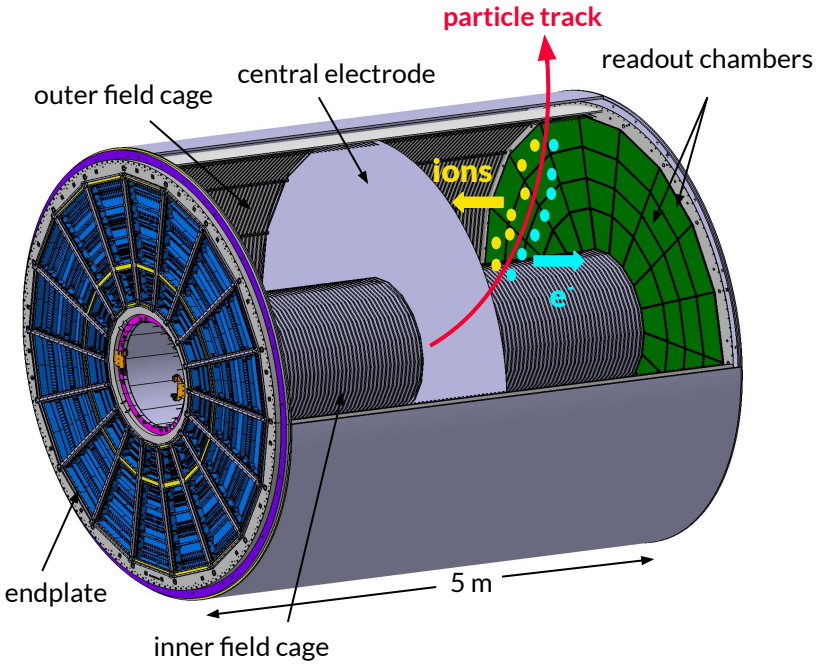
- ALICE TPC
- Specific Energy Loss (d*E*/d*x*) Measurement
- Particle Identification
- Machine Learning

ALICE TPC



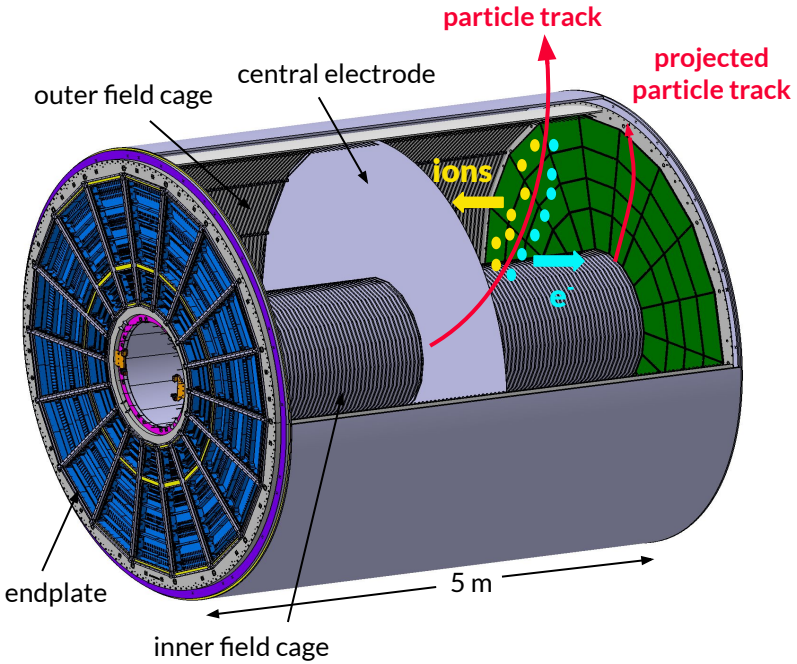
- The primary particle tracking and identification device in ALICE
  - Inner and outer field cages, high-voltage central electrode, endplates (18 sectors per side)

ALICE TPC



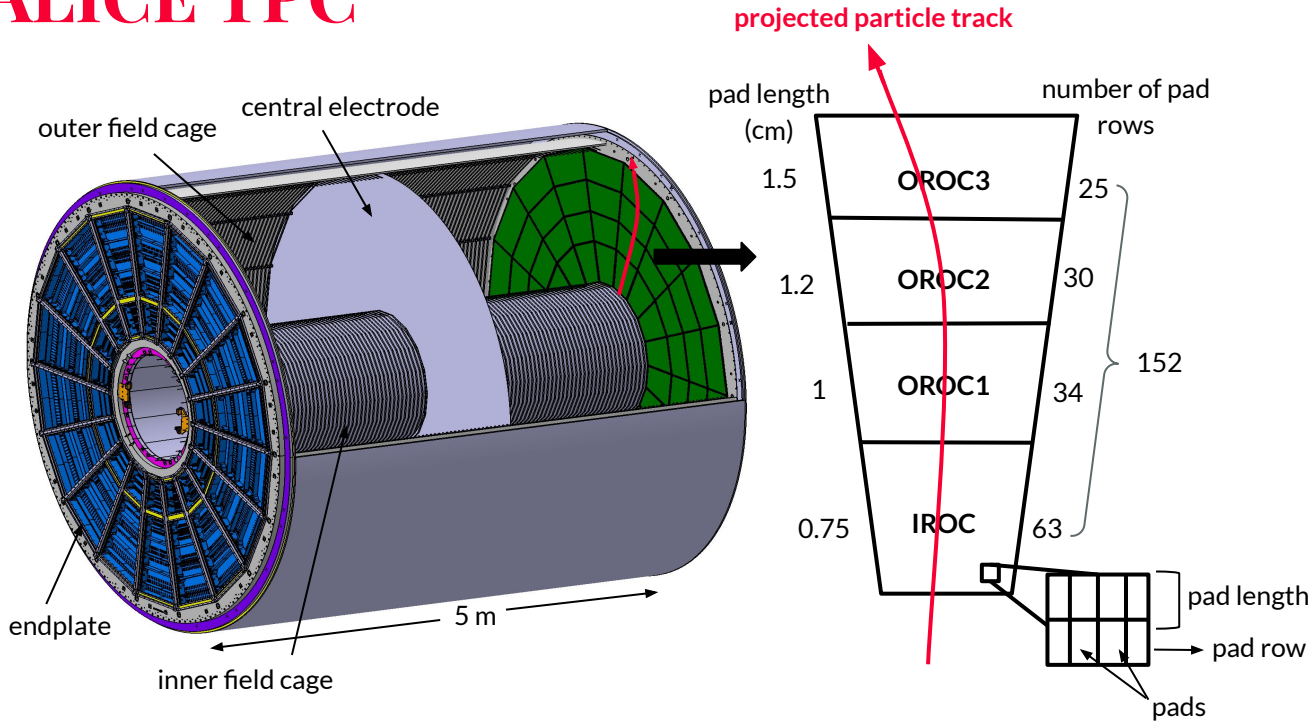
- It is a gas ionization detector
  - A charged particle loses energy by ionizing the gas

ALICE TPC



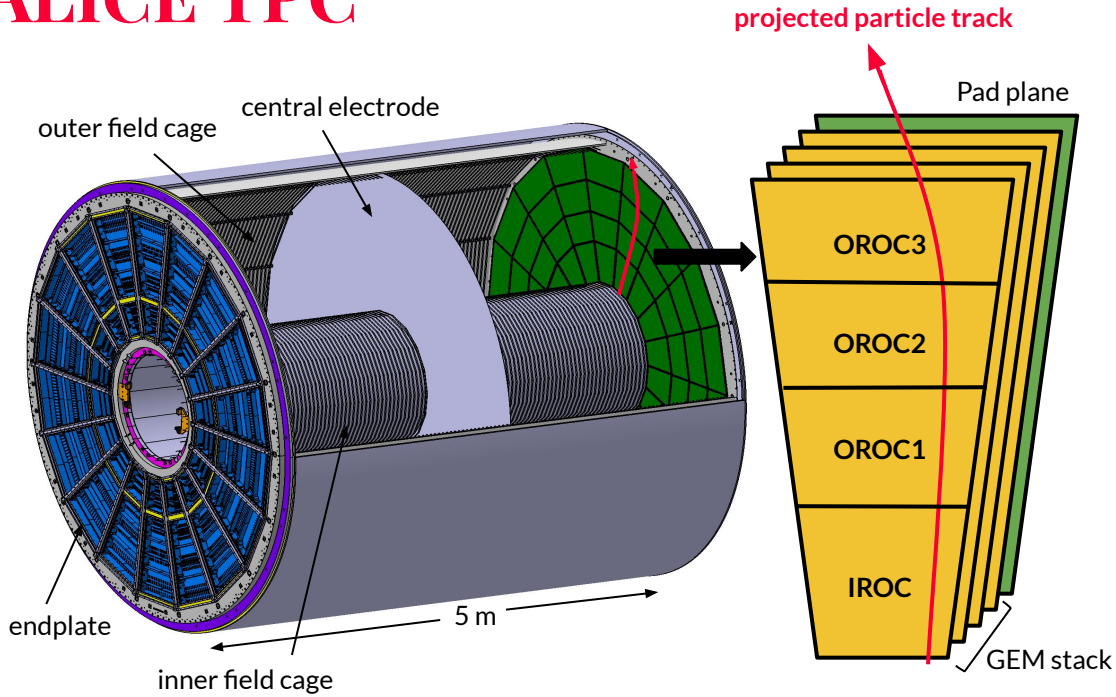
- The drifting electrons are projected onto readout planes

ALICE TPC



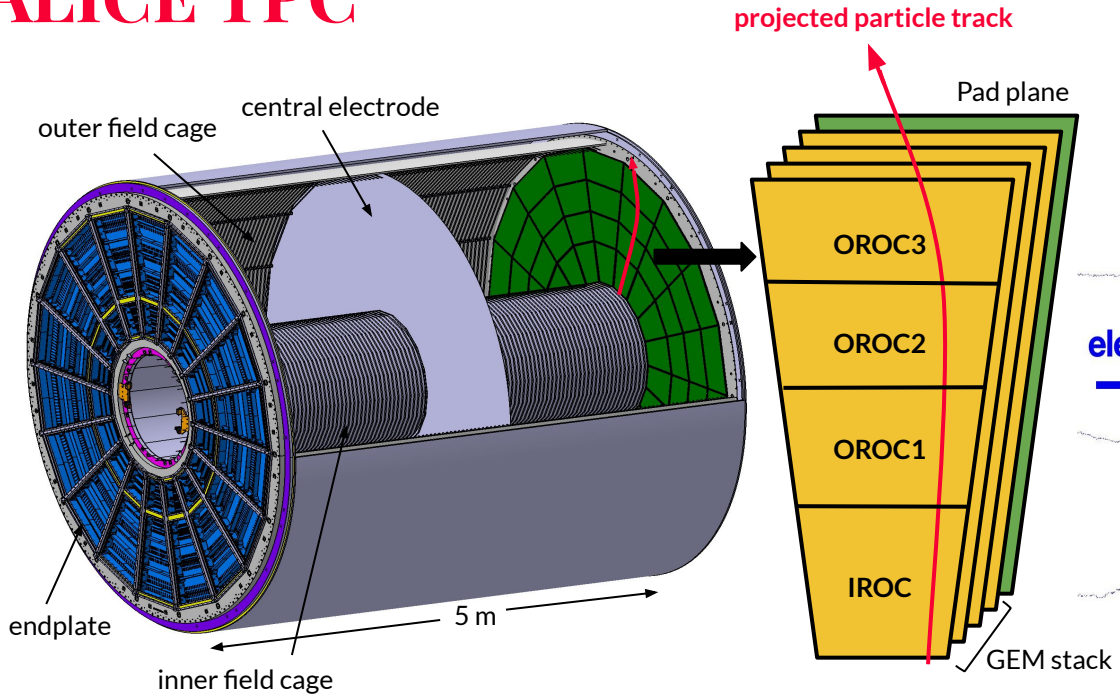
- The inner and outer readout chambers are denoted by IROC and OROCs, respectively
  - Different pad lengths and different numbers of pad rows

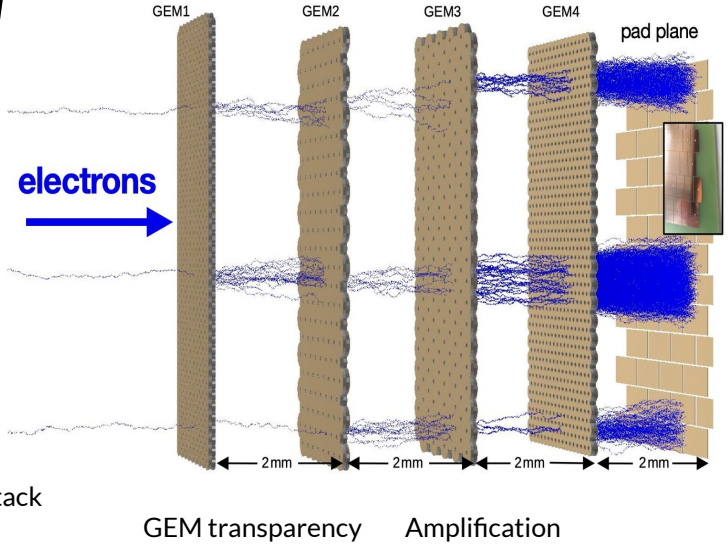
ALICE TPC



- For the gas gain, GEM stacks are used

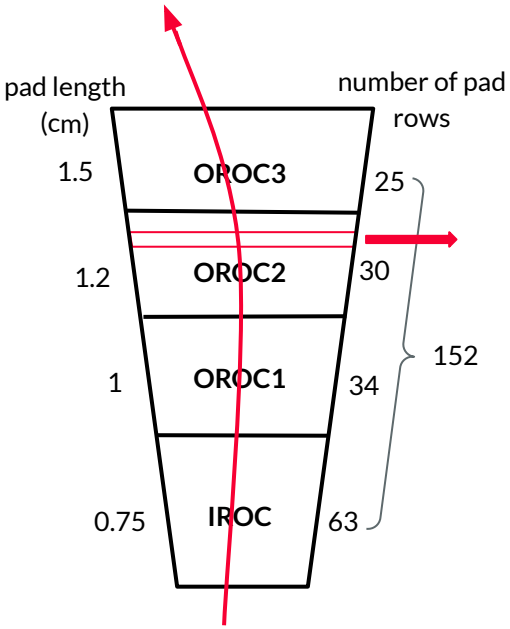
ALICE TPC

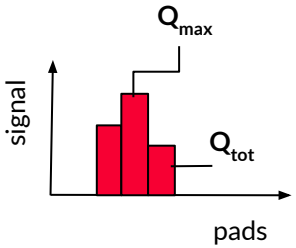




- GEM transparency affects the energy loss resolution
- The amplified charge then induces a signal on the pad plane
- The total charge of the signal is proportional to the specific energy loss of this particle

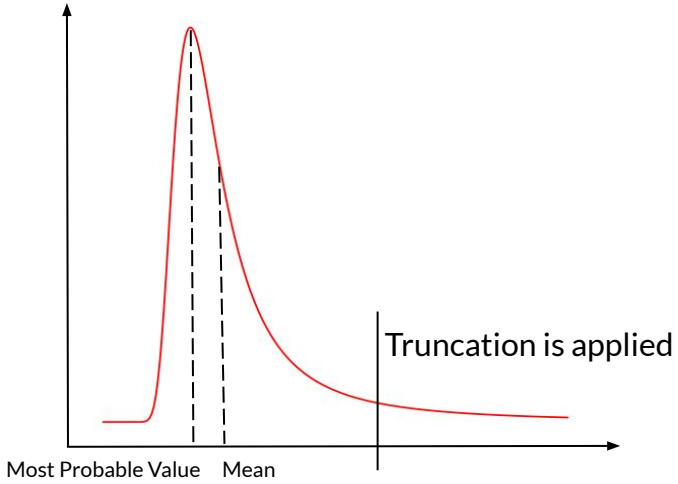
Specific Energy Loss Measurement





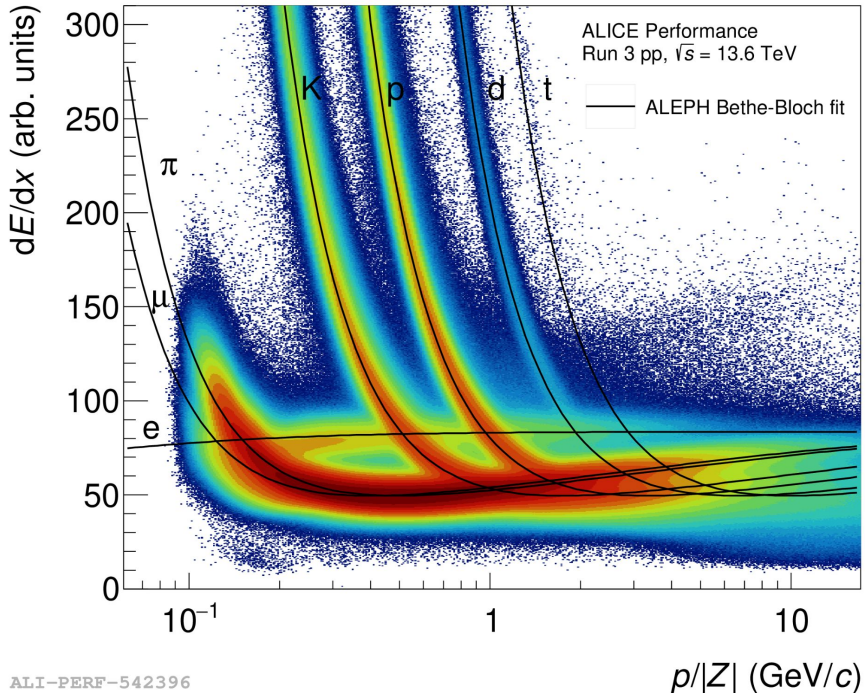
A total of 152 Q measurements

Landau distribution



- The fluctuations of the signal are described by the Landau distribution
- Clustering in pad and time direction is performed
- The specific energy loss (d*E*/d*x*) is calculated using a truncated mean method on the cluster charge (Q_{tot} or Q_{max})
- The truncation value is 60% in the ALICE TPC

Particle Identification (PID)



- The average behaviour of the dE/dx of a charged particle is described by the Bethe-Bloch formula

- ALEPH parametrization:

$$\langle dE/dx \rangle_{ALEPH} = p1\,(p2 - \log(p3 + (\beta\gamma)^{-p5})/\beta^{p4} - 1)\, z^{fz}\, f_{MIP}$$

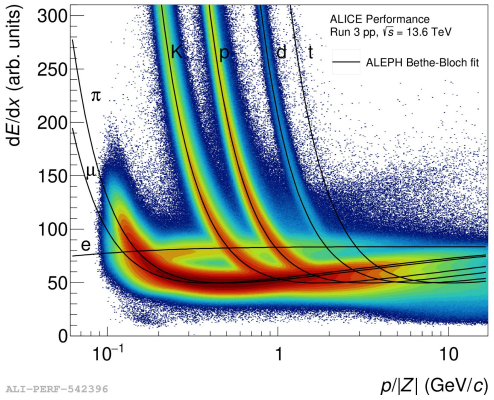
- Momentum:

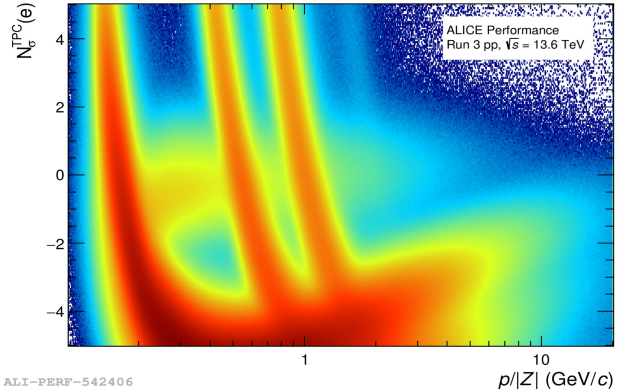
$$p = \beta\gamma \,.\, m$$

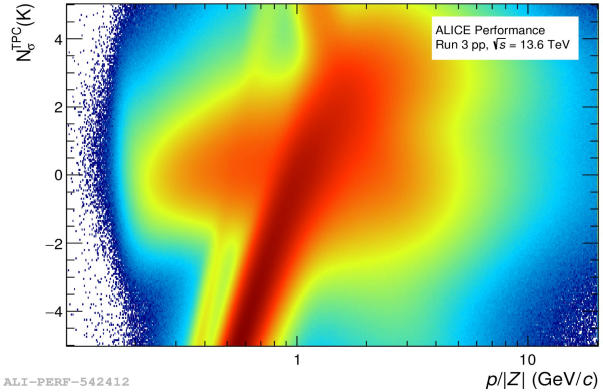
Particle Identification (PID)

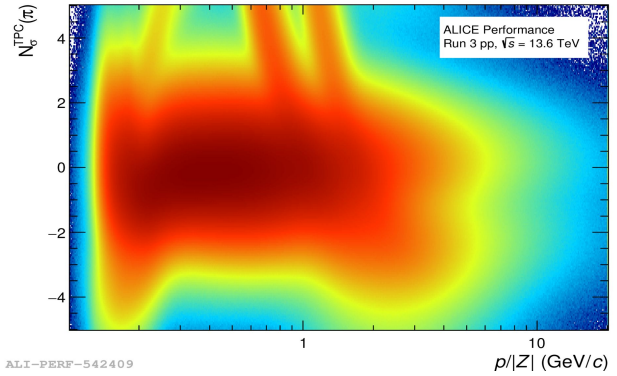
- Nσ-bands are defined around the average for PID

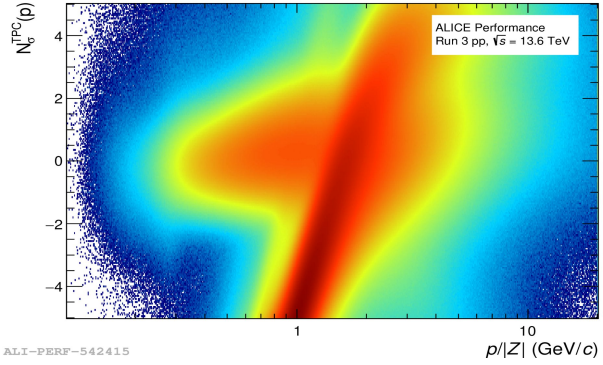
$$N\sigma = (dE/dx_{meas} - dE/dx_{exp})/\sigma_{exp}$$











Optimizing the PID Performance

- The Bethe-Bloch parameterization is not know a priori
  - A fit to data is required
- The Bethe-Bloch fit is not sufficient to describe all particles species in full phase space
  - A multidimensional correction is needed
- Resolution (sigma) parametrization requires a multidimensional description
- The d*E*/d*x* estimator might be improved by optimizing the truncation range separately on readout regions
- Different Machine Learning (ML) techniques can be used to carry out these tasks

Machine Learning

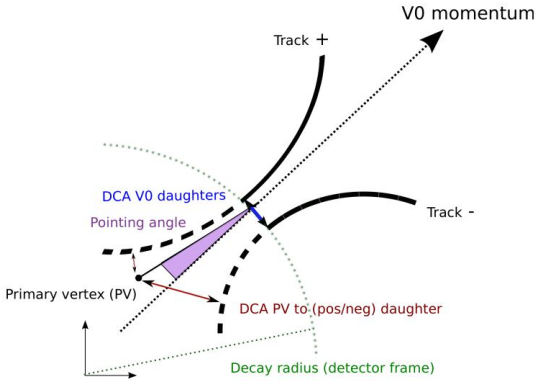
PID Response Parameterization

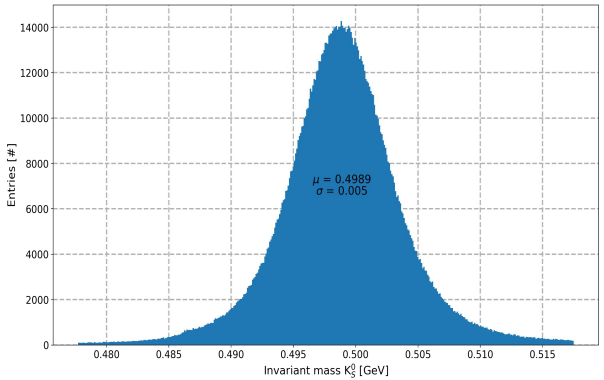
1. Update the Bethe-Bloch parameters
    a. using Optuna (full dataset, no skimming)
    b. using ROOT (skimmed clean samples available)
2. Skim data with optimal Bethe-Bloch parameters
3. Train Neural Networks (NN) for mean and sigma estimation
4. Mean+Sigma NN (relatively small, to optimize CPU usage) is stored
5. Application on analysis with the Open Neural Network Exchange (ONNX)

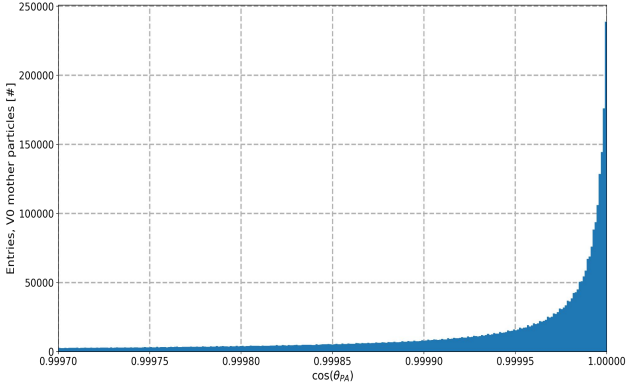
Full chain (1-5) is run per data taking period (several weeks of data taking)

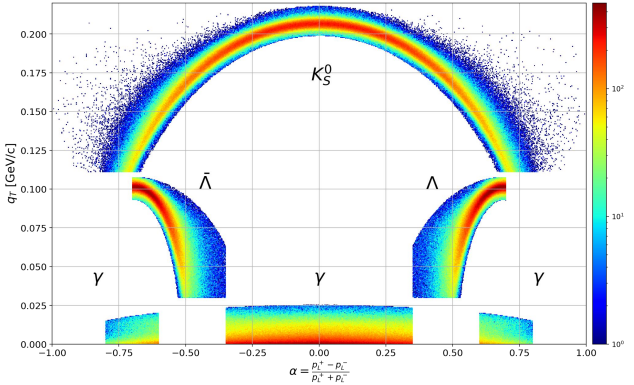
Data Preparation

- Clean samples of pions, protons and electrons
  - V0 selection based on
    - invariant mass
    - pointing angle (θ_{PA})
    - Armenteros-Podolanski selections
      - excluding overlapping regions



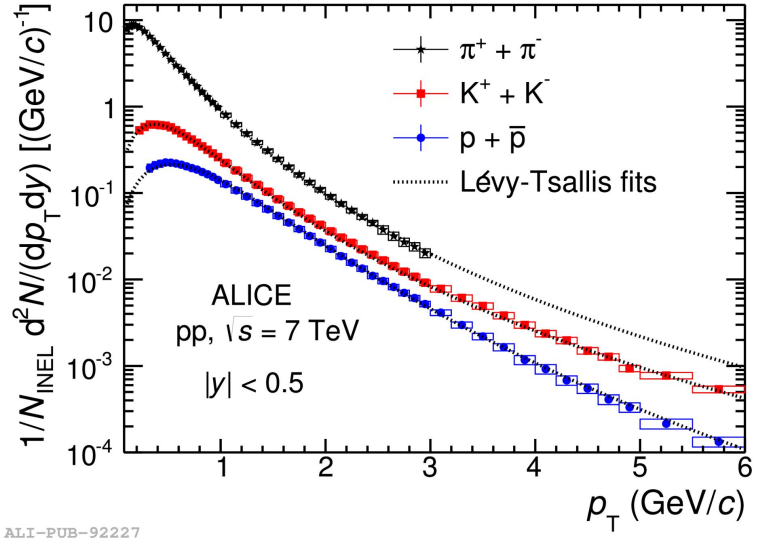






Data Preparation

- Skimming
  - Random downsampling of data using Tsalis/Hagedorn fit along the transverse momentum p_T
  - In order to reduce computational complexity skimmed data is used for the PID response parameterization
    - PetaBytes -> GigaBytes



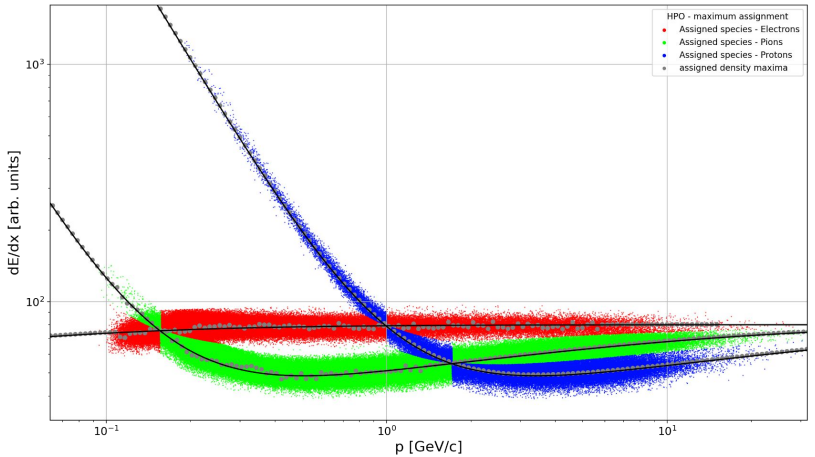
Bethe-Bloch Parameterization with Optuna

- Hyperparameter Optimization (HPO) framework - Optuna
  - Designed for the automation and the acceleration of the optimization studies
  - How does it work?
    - Define a function to be minimized
      - Error or score functions
    - Define number of trials for the optimization
    - In each trial Optuna will generate the hyperparameters in user defined ranges
    - Optuna will give the best parameters depending on the error or score values
- The parameters of the Bethe-Bloch fit are considered as hyperparameters

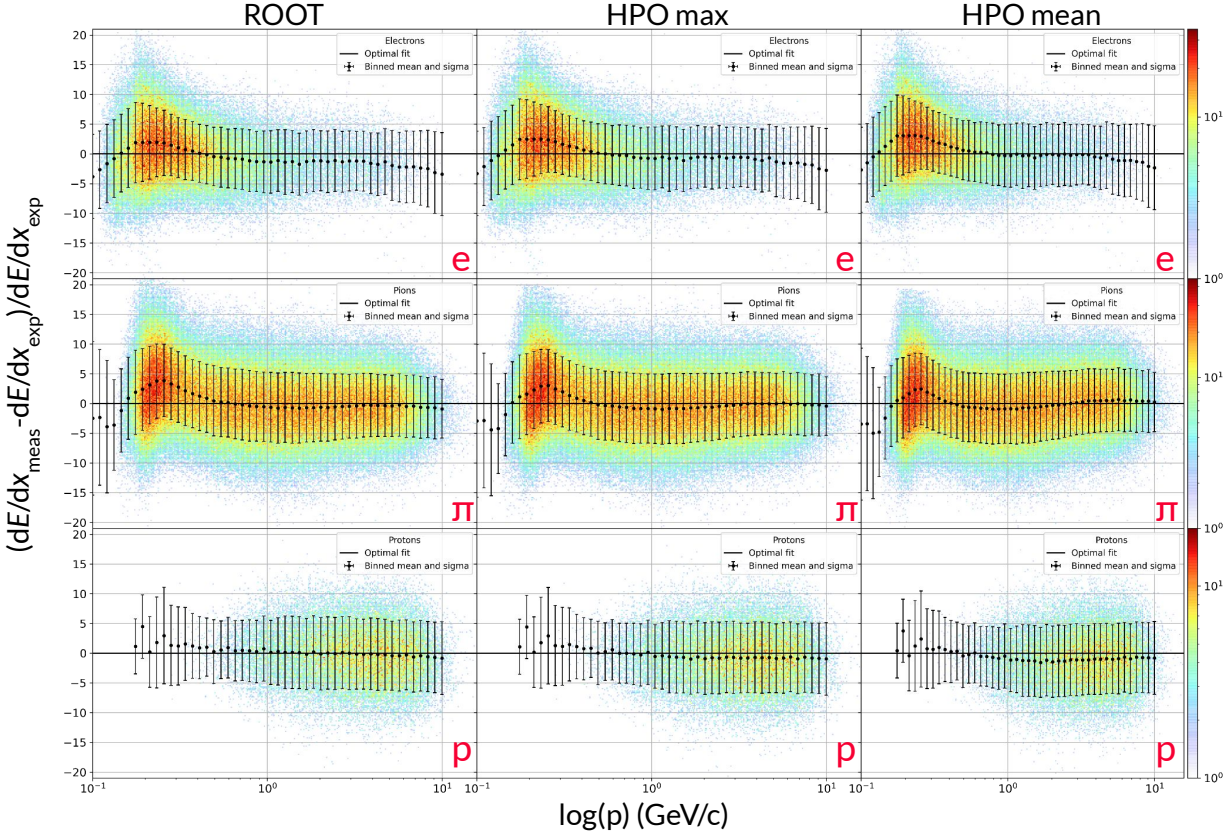
Bethe-Bloch Parameterization with Optuna

- Full dataset and range for Bethe-Bloch parameters are given to Optuna
- For each set of parameters
  - PID is assigned based on proximity to the nearest Bethe-Bloch curve (black line)
  - For each particle species, the mean or density maxima (grey points) is evaluated along the βγ binning
  - The score function is calculated
    - Mean squared sum of mean or max values to current Bethe-Bloch curve
- Optuna tries to optimize this score
- This runs on CPU (several CPU hours)

Bethe-Bloch fit : $p1\,(p2 - \log(p3 + (\beta\gamma)^{-p5})/\beta^{p4} - 1)$



Bethe-Bloch Parameterization Performance



- dE/dx_{exp} : Bethe-Bloch fit from ROOT (clean samples) or HPO (no clean samples)
- Fits are in good agreement with both mean and max estimations
- Residual deviations at low momenta
  - Further correction is needed

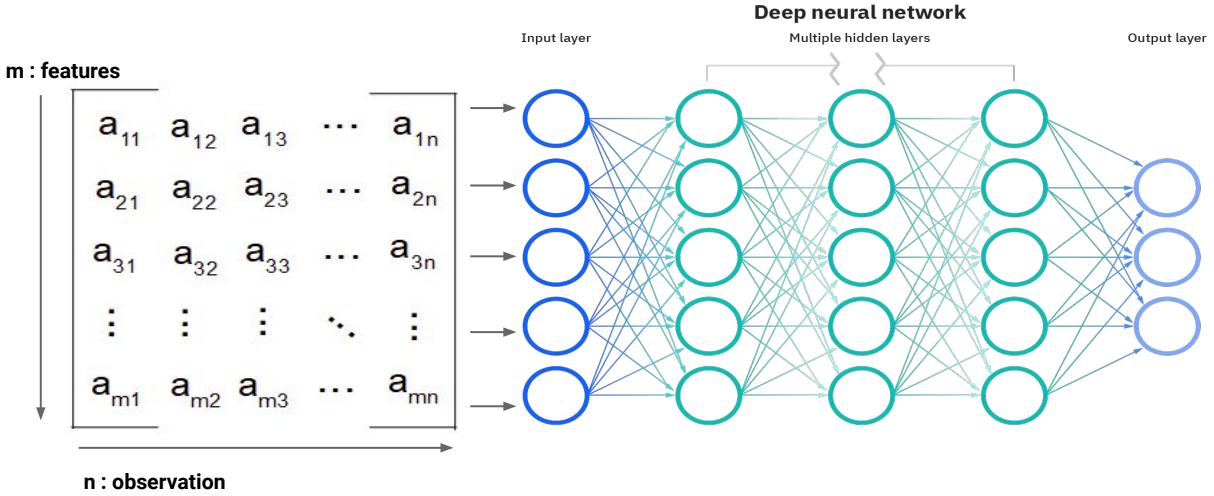
Mean Correction to Bethe-Bloch Parameterization

- Mean correction using Neural Networks in 6D phase space
- Network training is done on GPU using Pytorch
- Fully connected, 6 input layers, 3 hidden layer, 8 neurons per layer
- Loss function: Mean Squared Error (MSE)
- Activation function: tanh
- $dE/dx_{corr} = dE/dx_{exp} \cdot NN_{mean}$

NN learns the ratio

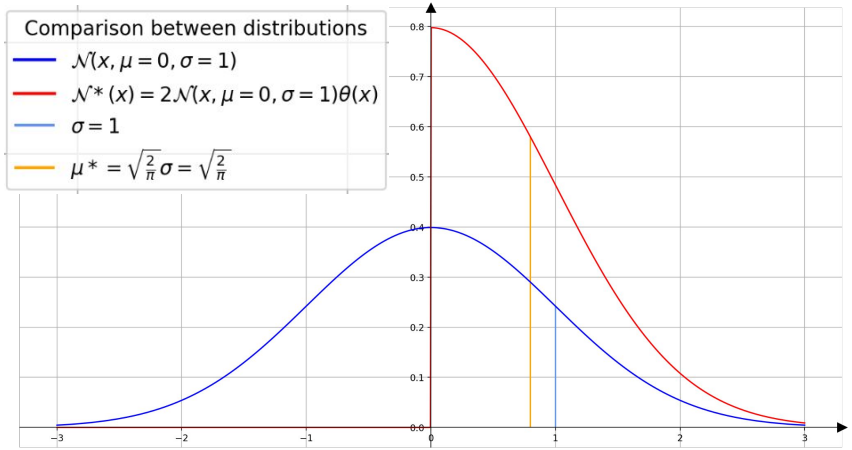
$$\frac{(dE/dx)_{meas}}{(dE/dx)_{exp}}$$

momentum
inclination angle
q/p_T
mass
multiplicity
number of clusters



Resolution Parameterization

- The data distribution for each particle species can be well approximated by a Gaussian distribution in d*E*/d*x* for every slice in momentum
- NN can learn the mean description of the data
  - For the resolution (sigma) parameterization the data is transformed in a way that the mean description is is proportional to sigma of d*E*/d*x*



blue curve: original distribution

red curve: transformed distribution

orange line: NN estimation which is proportional to original sigma

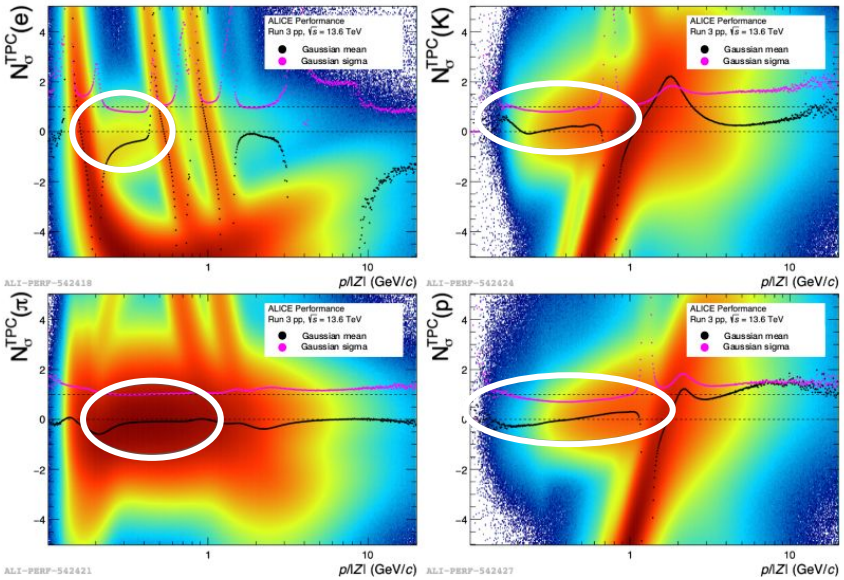
μ = sqrt(2/π) σ

Neural Network Corrections Performance

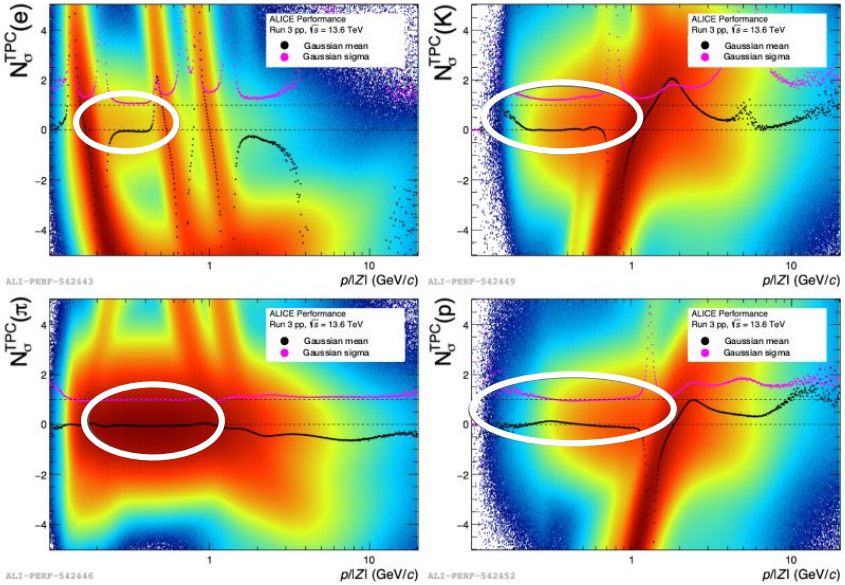
- NN corrections via ONNX for the analysis code for mean and sigma
- Comparison between Nσ plots

$$N\sigma = (dE/dx_{meas} - dE/dx_{exp})/\sigma_{exp}$$

Bethe-Bloch only

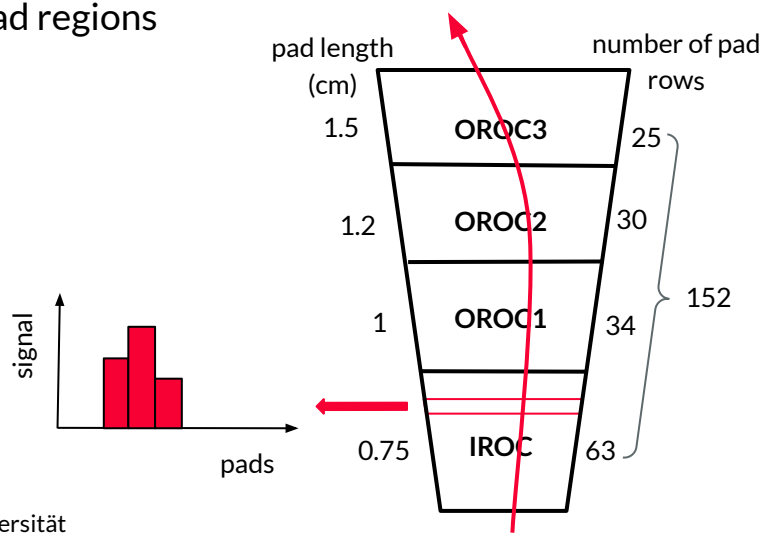


Bethe-Bloch + NN corrections



Optimization of d*E*/d*x* Estimator

- Toy Monte Carlo to simulate signal on the pad plane of the TPC readout
- Truncated mean calculations with truncation range from 30-100% for different regions
- Use Random Forest Regression model to estimate input of the simulation from truncated mean values
- Estimate best truncation combination from different pad regions
- Ongoing study



Toy Monte Carlo

- Fast microscopic simulation
- Sequence of processes

Input

$\langle dN_{prim,in}/dx \rangle$: Mean number of primary electrons per cm, per track

Studied in range [10-510] (1/cm)

Primary ionization

N_{prim}: Number of primary electrons along the particle trajectory

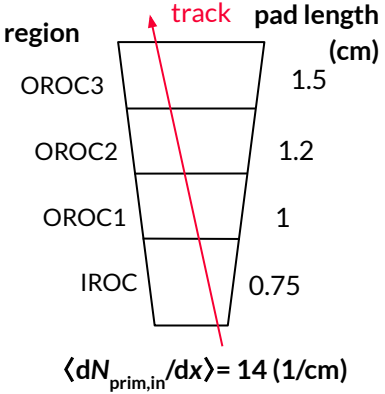
Poisson distribution

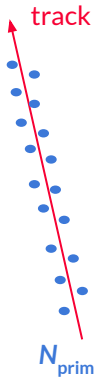
Total ionization

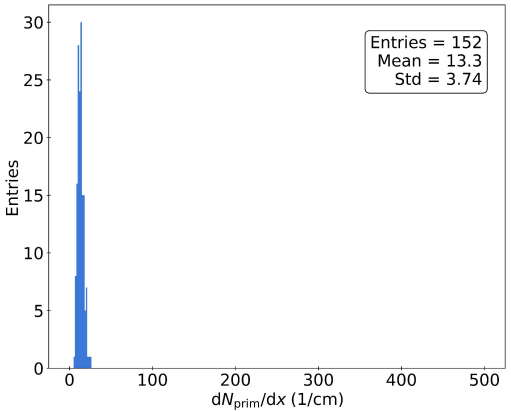
N_{tot}: $\sum^{Nprim} N_{sec}$

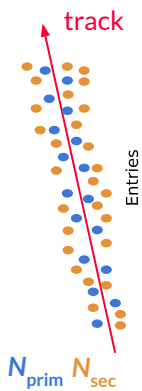
N_{sec}: Number of secondary electrons (power law) generated for each primary electron along the particle trajectory

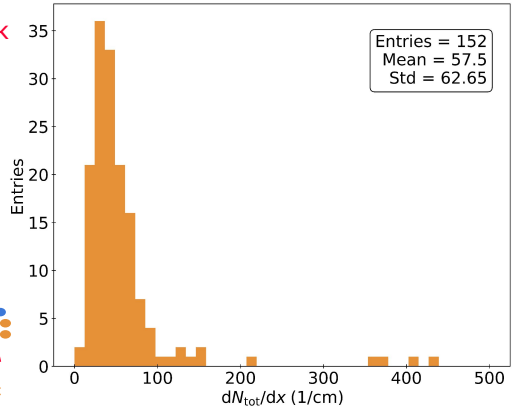
Simulating 1 track











Toy Monte Carlo

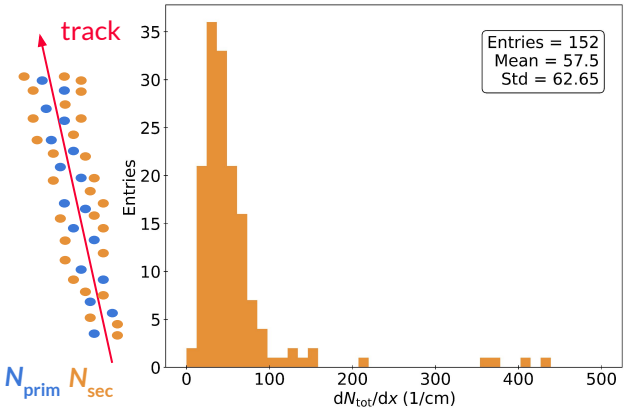
- Fast microscopic simulation
- Sequence of processes

Total ionization

N_{tot}: $\sum^{Nprim} N_{sec}$

N_{sec}: Number of secondary electrons (power law) generated for each primary electron along the particle trajectory

Simulating 1 track

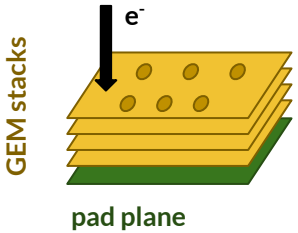


GEM transparency

GEM transparency: Studied in range [50%-100%]

$$\frac{\text{electrons passing through GEM holes}}{\text{electrons approaching}}$$

Transparency fluctuations: uniform, mean=1



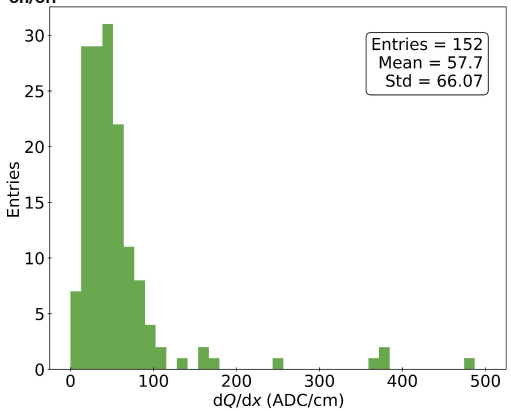
GEM transparency = 50%

Amplification

Q: $\sum^{Ntot}$ transparency fluctuations + gas gain fluctuations

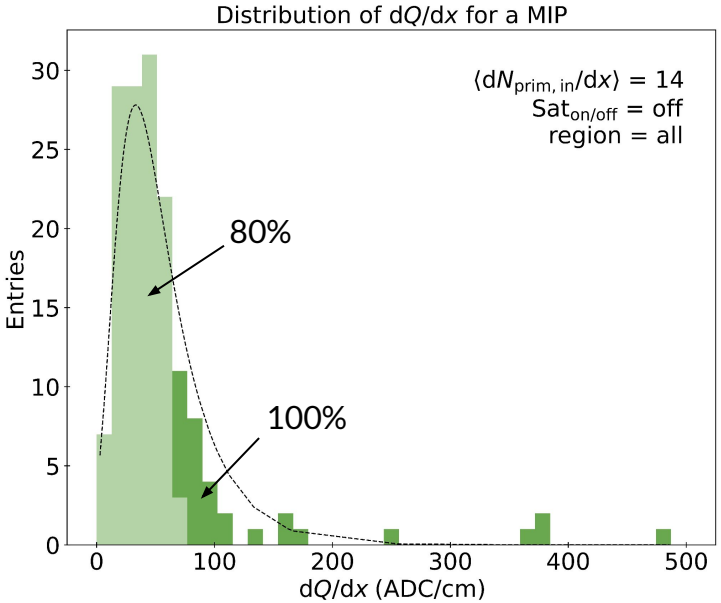
Gas gain fluctuations: exponential, mean=1

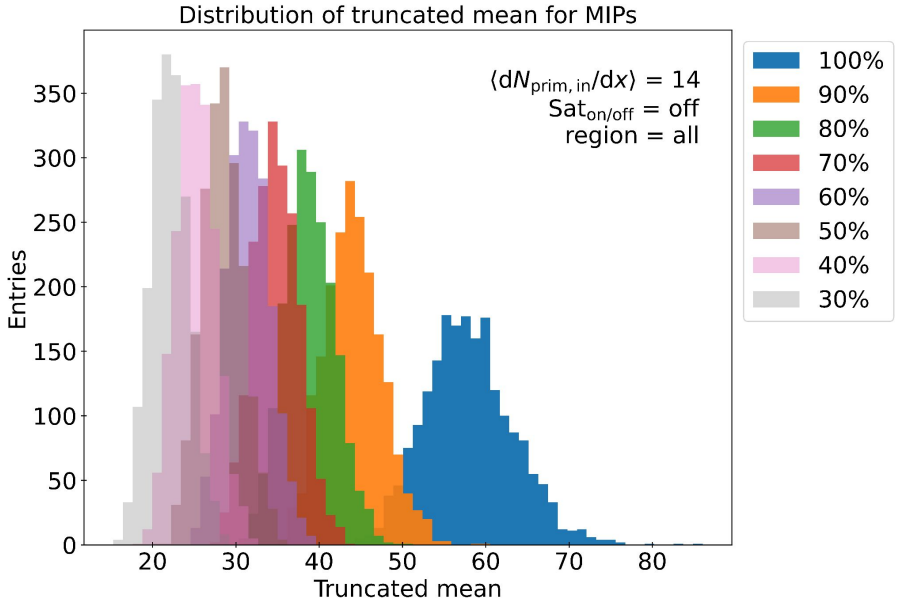
Sat$_{on/off}$: Saturation cut



Truncation

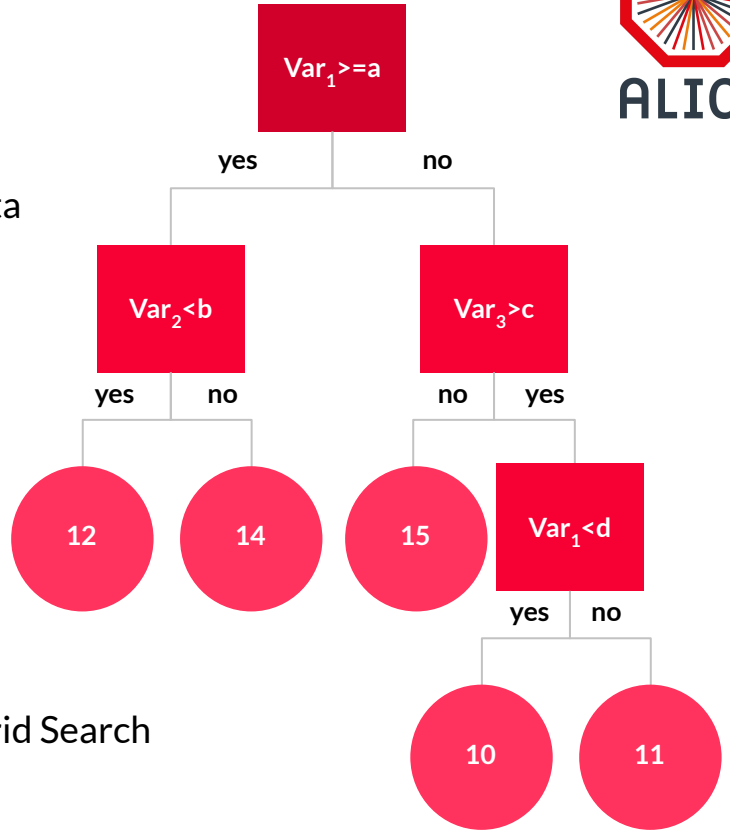
- dQ/dx (dQ/dx ≈ dE/dx) values are sorted for each track and truncated by each percentage value:
  - 30, 40, 50, 60, 70, 80, 90, 100 %
- Mean is calculated for different regions:
  - IROC, OROC1, OROC2, OROC3 and all





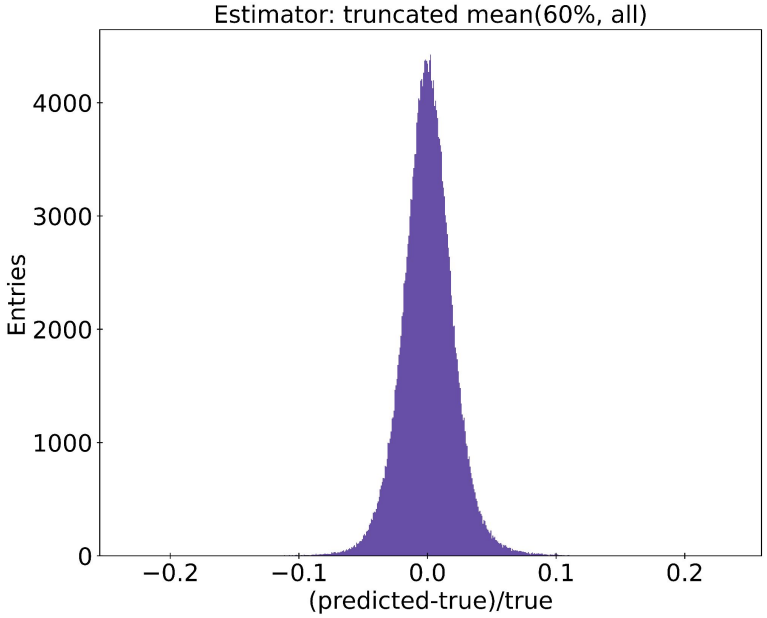
Random Forest Regression

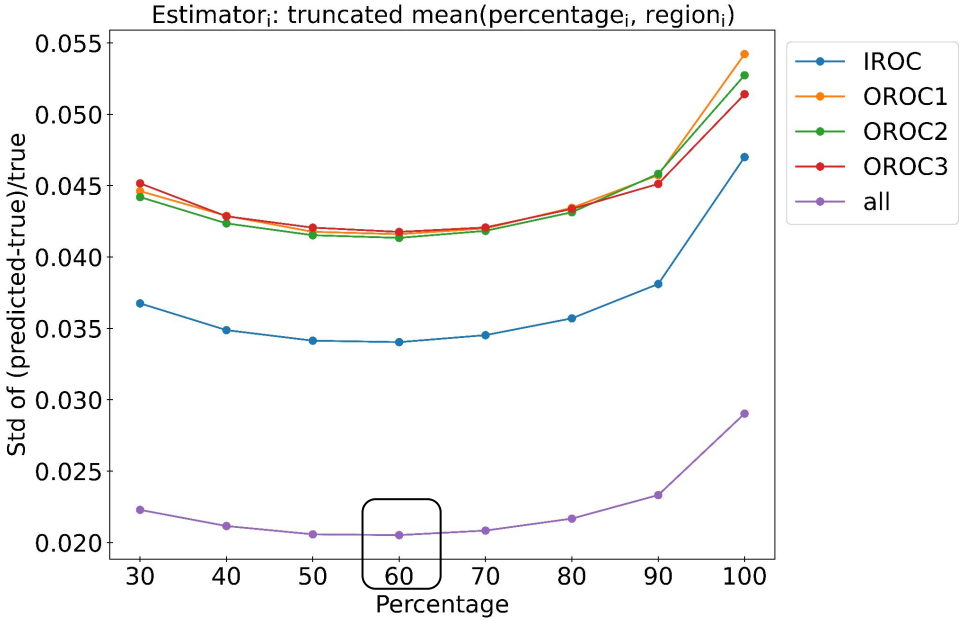
- Ensemble of decision trees
- Each tree is created from a randomly sampled subset of data
- Each tree makes its own prediction
  - Var_i : truncated mean values
  - Prediction target: $\langle dN_{prim,in}/dx \rangle = 14$ (for a MIP)
- These predictions are averaged to produce a single result
- Number of trees and depth of a tree are optimized using Grid Search



Random Forest Regression

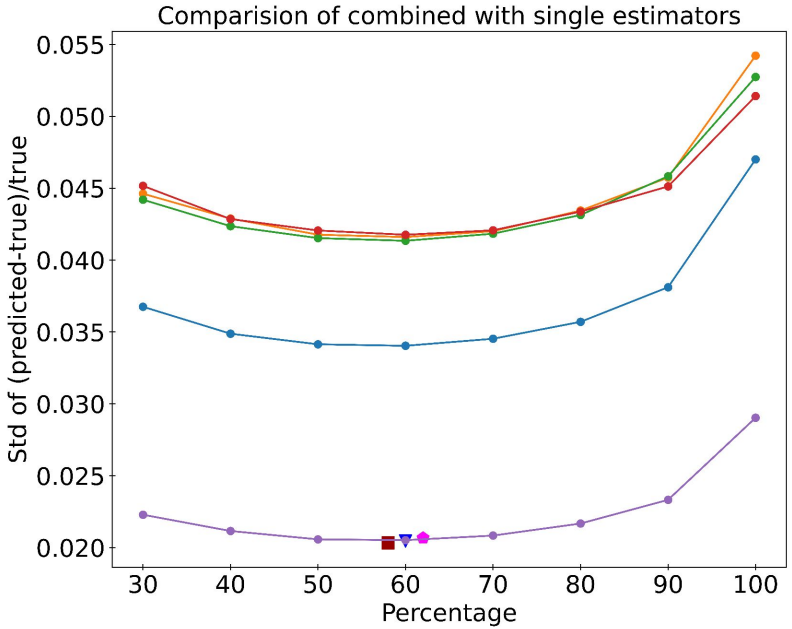
- Prediction of $\langle dN_{prim,in}/dx \rangle$ with a single estimator
  - Estimator$_i$: truncated mean (percentage$_i$, region$_i$)
  - Best single estimator ⟶ truncated mean (60%, all)





Random Forest Regression

- Prediction of $\langle dN_{prim,in}/dx \rangle$ with a combined estimator
  - Estimator$_i$: truncated mean[(percentage$_i$, IROC), (percentage$_i$, OROC1), (percentage$_i$, OROC2), (percentage$_i$, OROC3)]



(60%, IROC)
(60%, OROC1)
(60%, OROC2)
(60%, OROC3)

Best single estimators from different regions

Combined estimators:

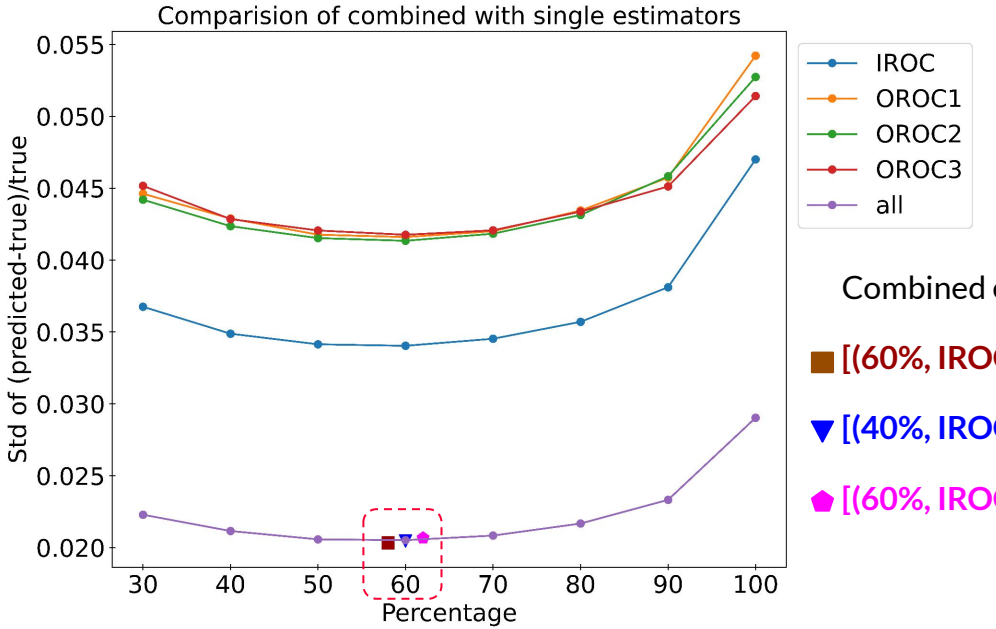
■ [(60%, IROC), (60%, OROC1), (60%, OROC2), (60%, OROC3)]

▼ [(40%, IROC), (50%, OROC1), (60%, OROC2), (70%, OROC3)]

⬟ [(60%, IROC), (50%, OROC1), (40%, OROC2), (30%, OROC3)]

Random Forest Regression

- Prediction of $\langle dN_{prim,in}/dx \rangle$ with a combined estimator
  - Estimator$_i$: truncated mean[(percentage$_i$, IROC), (percentage$_i$, OROC1), (percentage$_i$, OROC2), (percentage$_i$, OROC3)]



Combinations of pad regions with different truncation percentages do not provide a better description of the simulation.

Combined estimators:

■ **[(60%, IROC), (60%, OROC1), (60%, OROC2), (60%, OROC3)]**

▼ **[(40%, IROC), (50%, OROC1), (60%, OROC2), (70%, OROC3)]**

⬟ **[(60%, IROC), (50%, OROC1), (40%, OROC2), (30%, OROC3)]**

Summary & Outlook

- PID response parameterization
  - Bethe-Bloch parameterization is obtained from Optuna or ROOT fit (clean samples)
  - Mean and sigma are estimated in multidimensional space from real data using NNs
  - Application on analysis with ONNX
- Optimization of d*E*/d*x* estimator using Random Forest Regression
  - Effects of different truncation ranges on pad regions studied
    - Toy Monte Carlo - More detector effects to be added
  - Combinations of different truncated mean (percentage, region) values studied

Thank you
for your attention