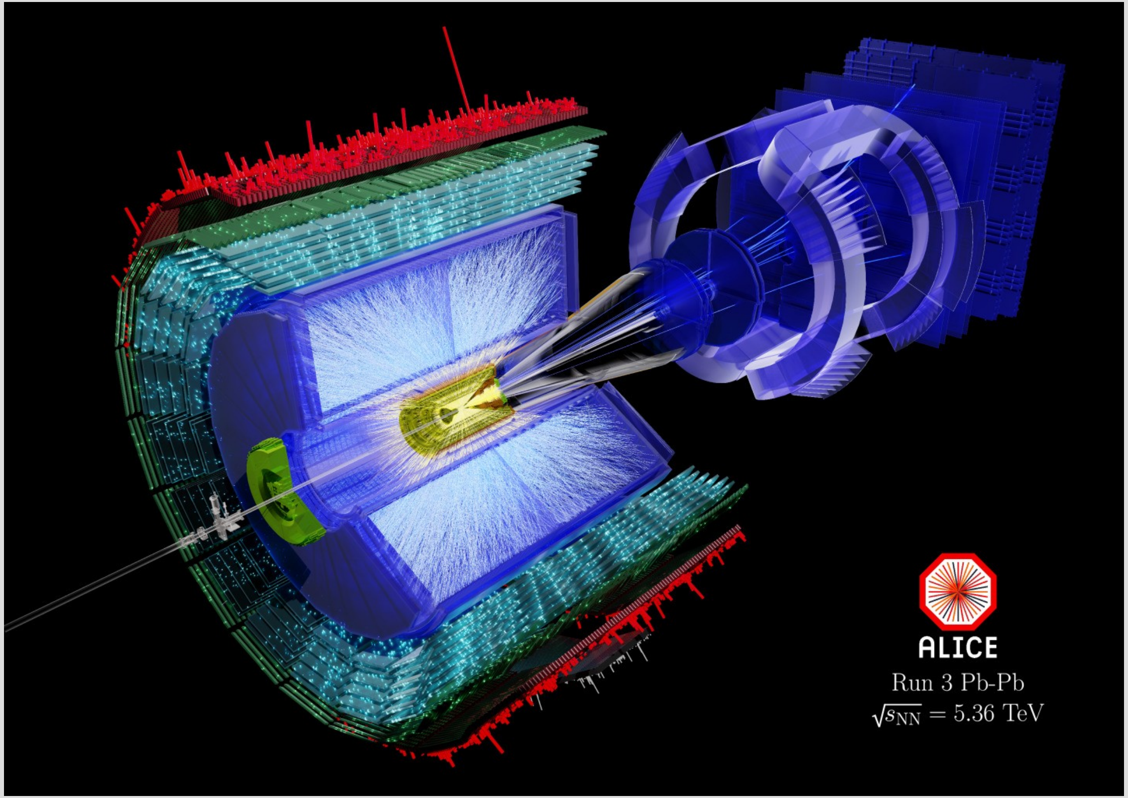
Neural Networks for calibrating particle identification in ALICE

Jeremy Wilkinson (GSI)

ALICE: A Large Ion Collider Experiment at CERN

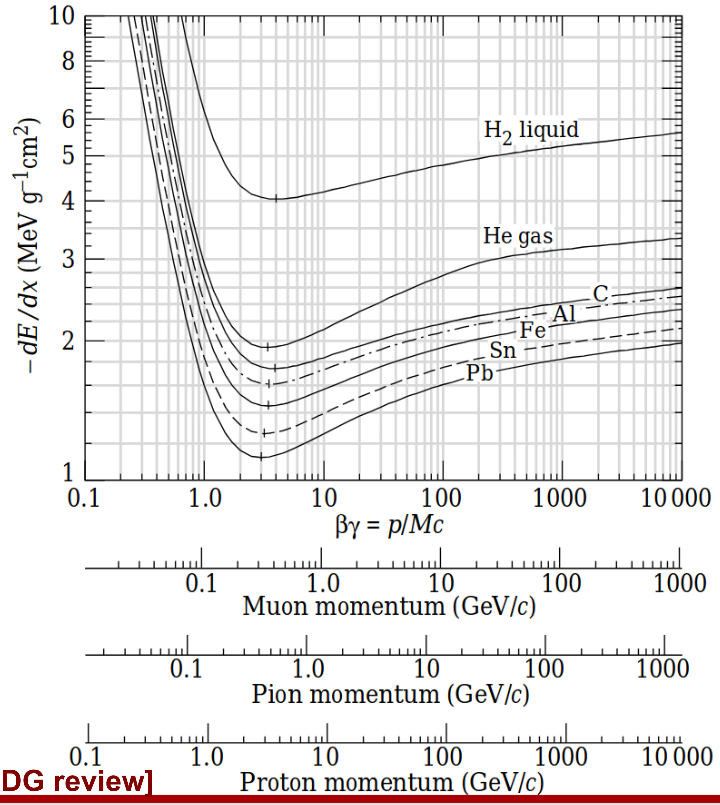
- ALICE is dedicated to the study of nuclear matter under extreme energy densities – Quark-Gluon Plasma (QGP)
- Pb ions collided at $\sqrt{s_{NN}}$ = 5.36 TeV at ~50kHz – typically O(10^3) tracks per central (head-on) collision
- Reconstruction of rare decays from product tracks using precise **tracking** capabilities (ITS, TPC) complemented by **particle identification (PID)** from TPC+TOF



Introduction to TPC PID

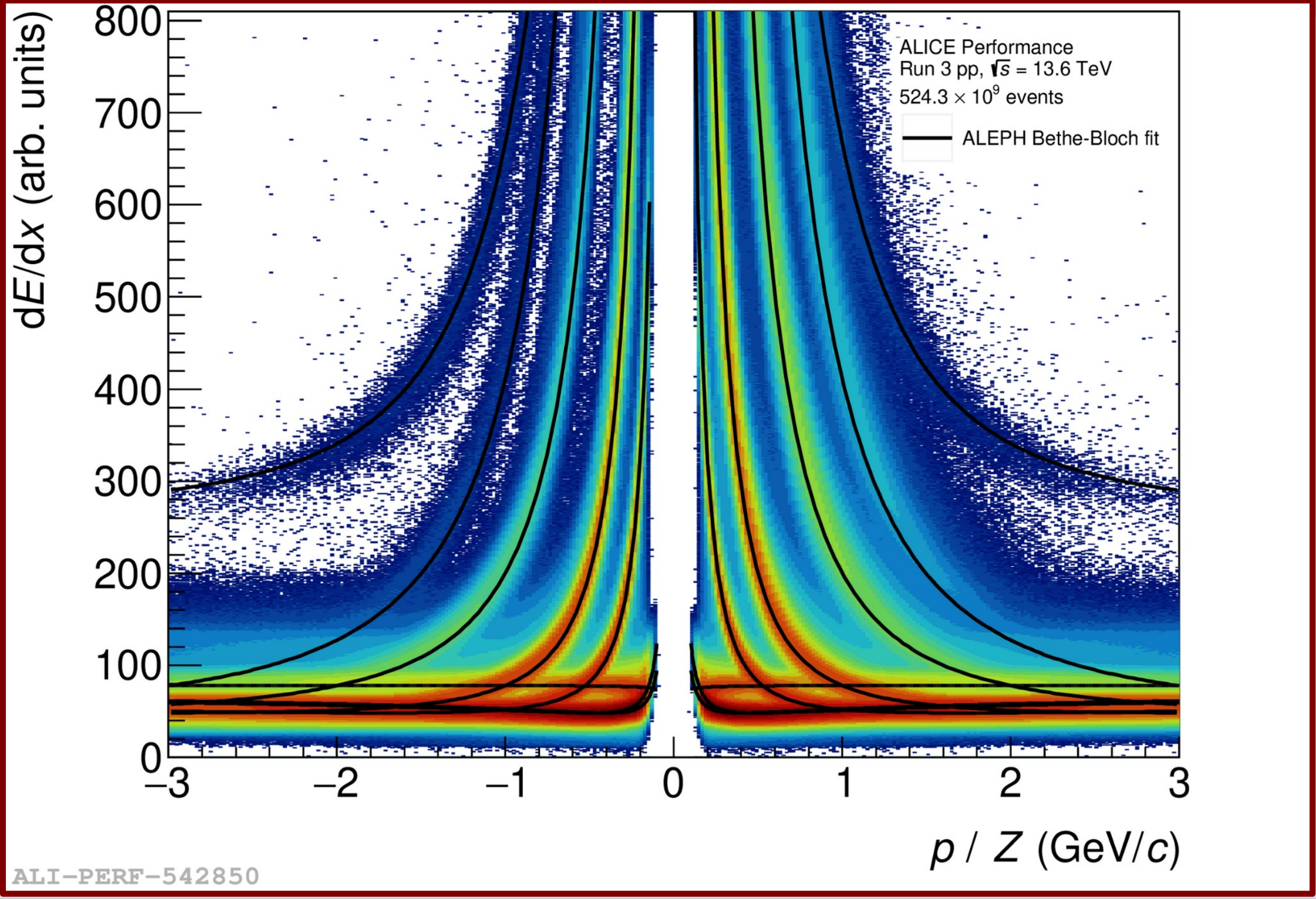
- ALICE Time Projection chamber: Main detector system for tracking + identification (PID) of charged particles at midrapidity
- Working principle for PID:
  - Charged particles lose energy as they traverse detector gas
  - Energy loss measured as specific energy deposit per unit length, dE/dx
  - Ideal case: Bethe-Bloch energy loss curve as function of βγ characteristic for all particle species, defined by properties of detector medium

  => Particles of different mass will have distinct Bethe-Bloch curves
- Selection based on n_σ, "number of standard deviations" in signal difference from expectation



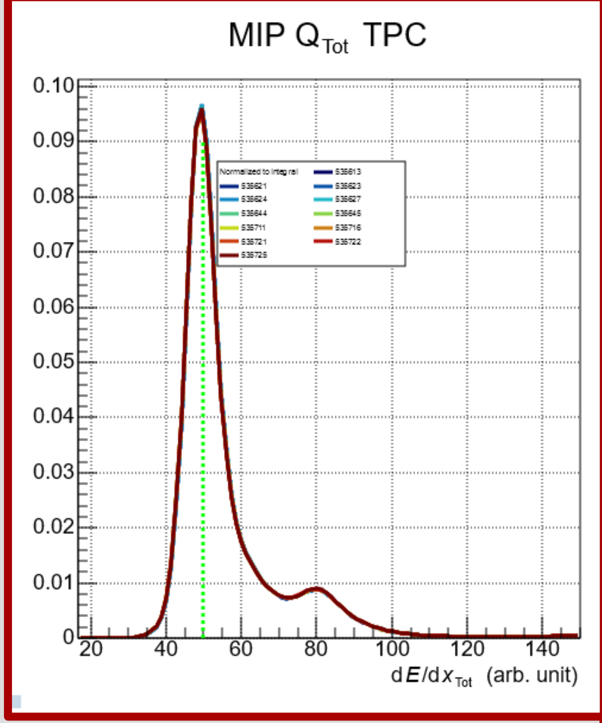
[PDG review]

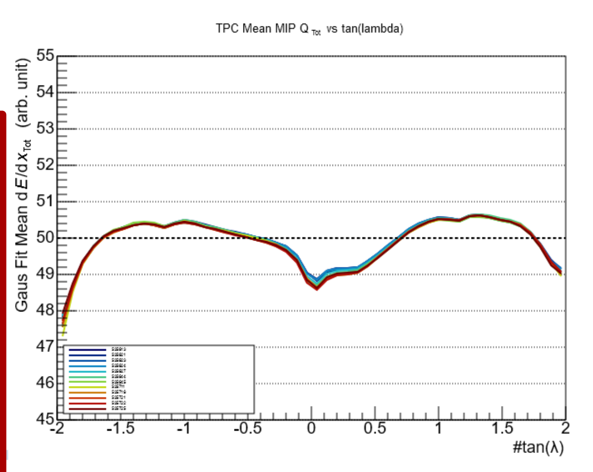
Introduction to TPC PID

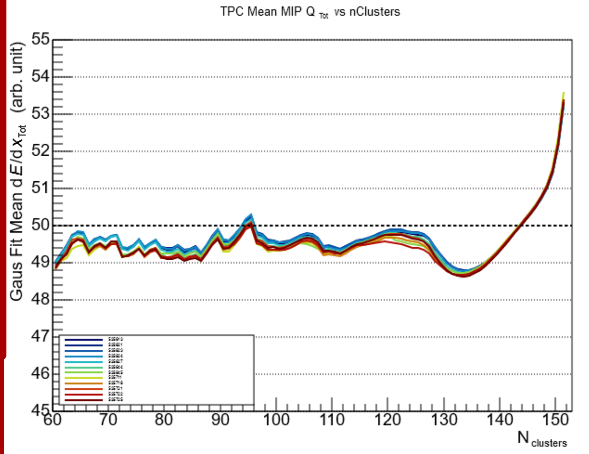


Introduction to TPC PID

- In reality, Bethe-Bloch description alone is not enough to fully describe PID response
- Measured signal and precision can depend strongly on:
  - Environmental conditions (gas temp/pressure)
  - Fluctuations in gain calibration
  - Number of clusters associated to a track
  - Detector occupancy / interaction rate
  - Region of detector / dead channels
  - and more!

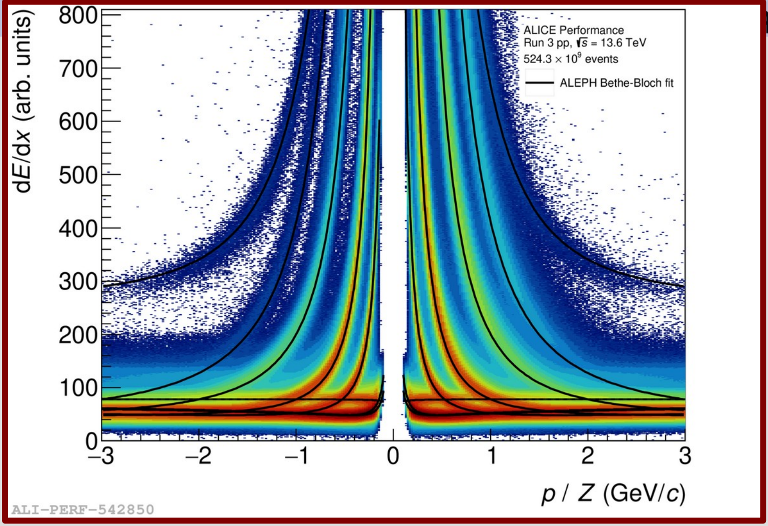


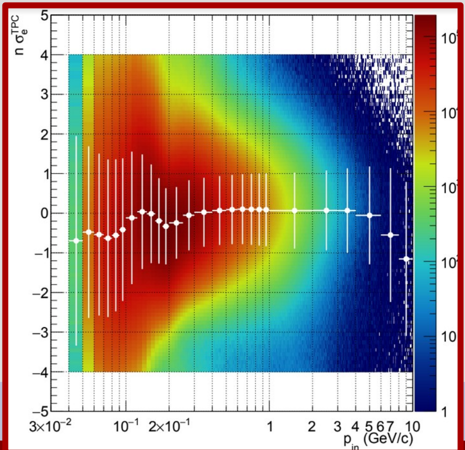




Goal of TPC PID in analysis software

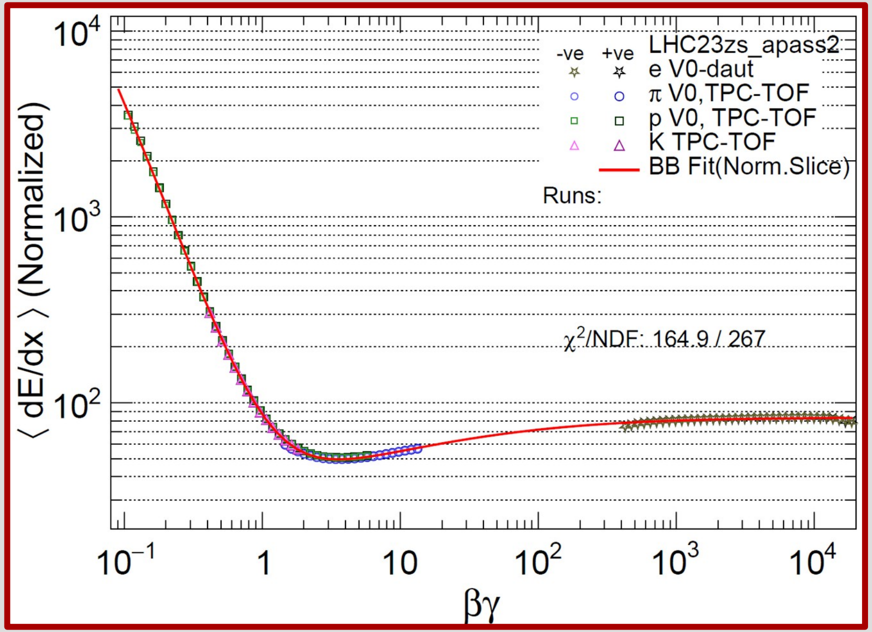
- Main goal: Post-calibration of dE/dx response at analysis time
  - → Provide expected mean + sigma for each track, based on knowledge of detector response
  - → Input signal + track/event parameters from standard data file, output n_σ for each species in PID table
  - → Analyser calls "track->tpcNSigma*Species*" for simple use in their own task - does not have to use raw d*E*/dx information in analysis
  - → Perform a multidimensional fit of relevant parameters rather than a simple product of independent 1D corrections

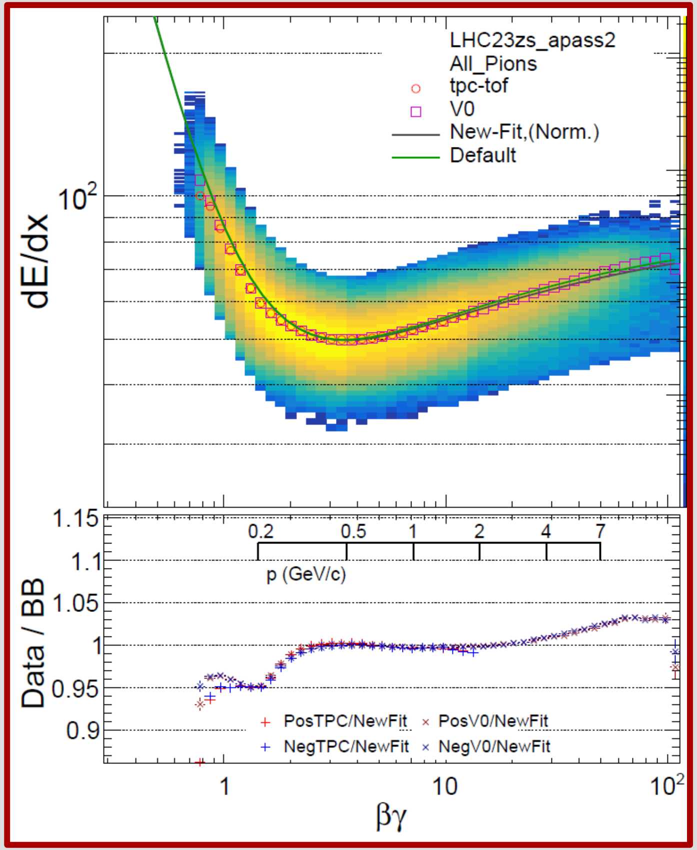




Basic dE/dx parameterisation

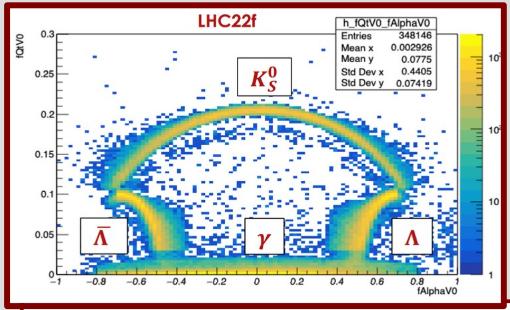
- Determination of BB function parameters: Lightweight ROOT fit for dE/dx parameters for all species simultaneously.
- The BB parameters are obtained from fitting⟨dE/dx⟩vs $\beta\gamma$ with a parametrized (ALEPH) function.

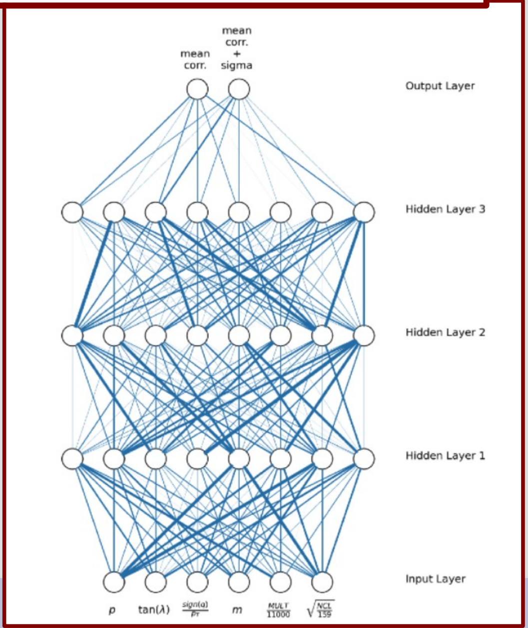




Neural network corrections for TPC PID in Run 3

- Main procedure for Run 3: 6D correction of expected mean + width of PID distribution for each species
- Inputs: Bethe-Bloch parametrisation, momentum, incident angle, charge/transverse momentum, species mass, TPC occupancy, number of clusters
- Output: Correction factor for expected BB curve
- Training performed on clean samples of input tracks from so-called “V^0” decays
- Performed per “chunk” of reconstructed data (~week of data taking) with similar data-taking conditions



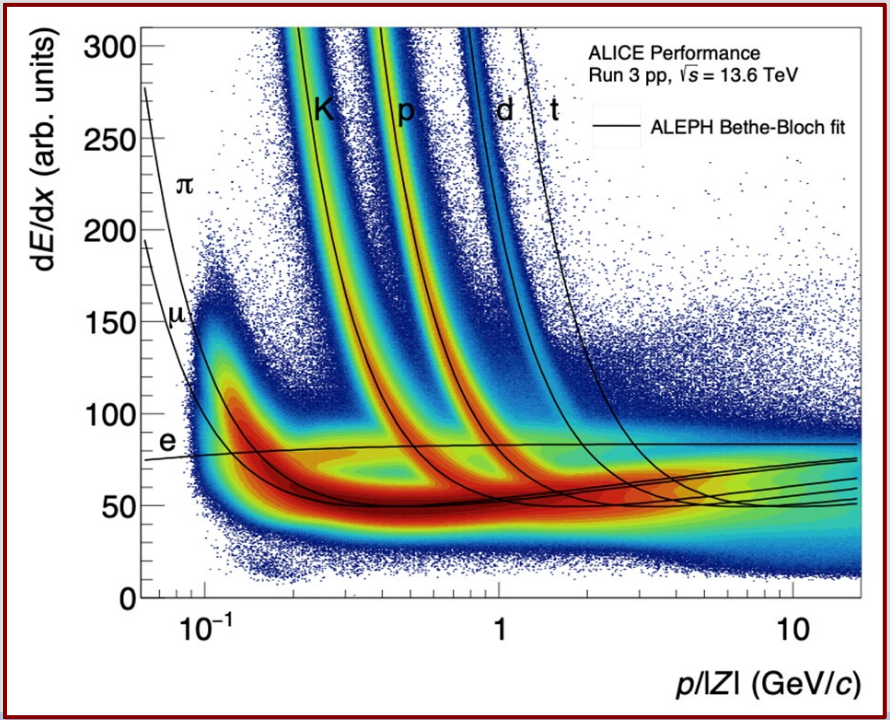


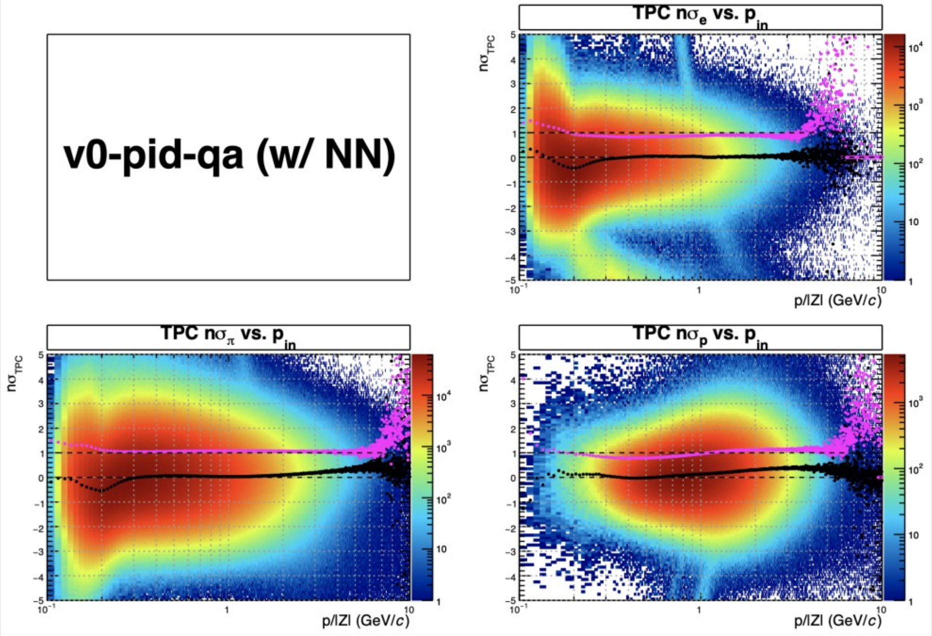
Implementation in O² analysis software

1. Skimming of processed data performed on LHC Grid using customised V^0 selection tasks
2. Train neural networks ⇒ PyTorch: Flexibility and easy GPU support, using GPU resources on GSI computing farm
3. Interface between Python and C++ for Neural Network training/inference: ONNXRuntime - Open Neural Network Exchange. Allows portability of trained networks between different codebases + simple loading at runtime
4. ONNX models uploaded to ALICE Calibration & Conditions Database (CCDB); Automatic file-fetching from CCDB and application at runtime
5. Corrections applied to <dE/dx> and resolution before saving nσ information to TPC tables; application is transparent to analyser
6. Dedicated QA tasks can be attached to check performance for all species

Performance in pp collisions

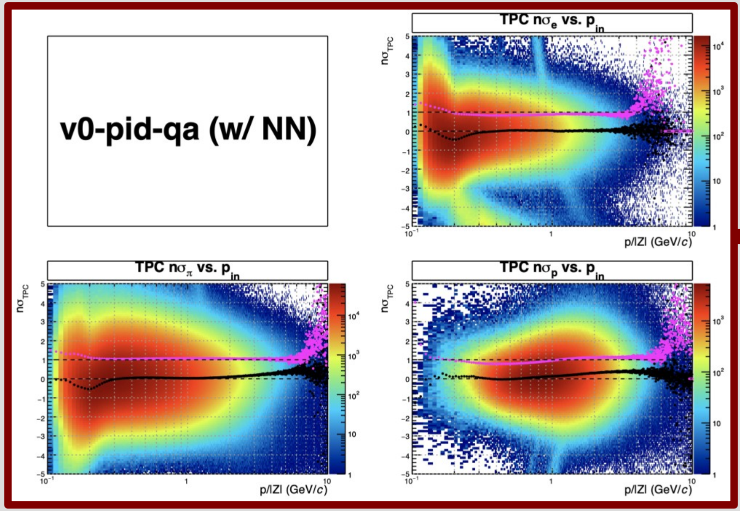
- Production of pp collisions from 2023 data-taking
- Pure BB function captures distributions reasonably well, but with deviations at high momenta for pions and protons

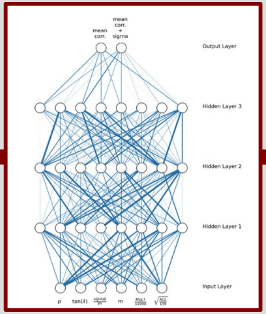


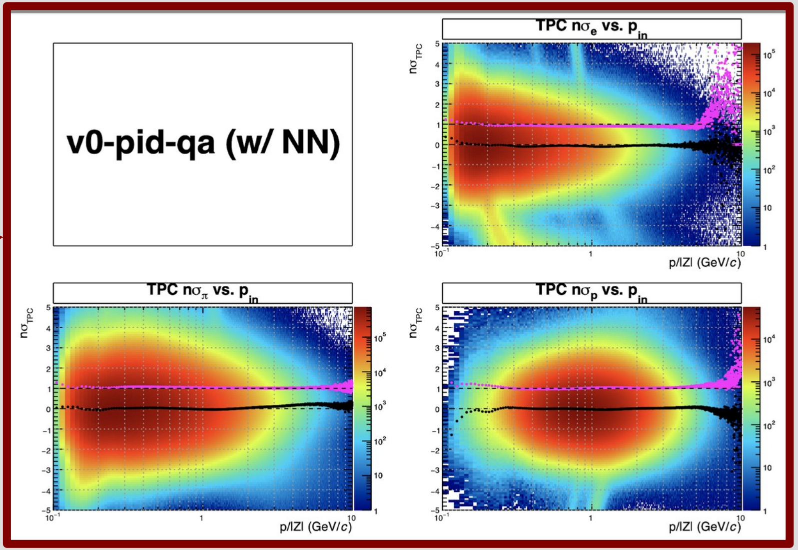


Performance in pp collisions with neural network

After applying NN corrections:
- Significant improvement over full kinematic range, particularly in low-momentum region
- Reliable mean correction and sigma estimation in all regions of clear separation in the TPC

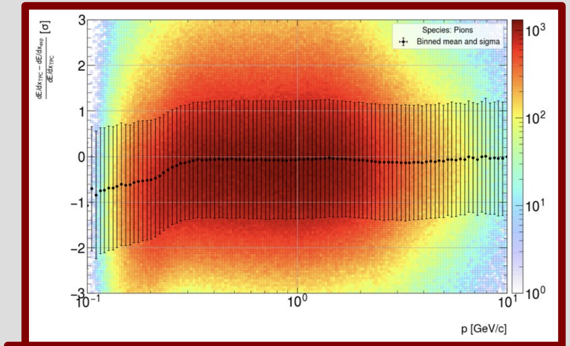


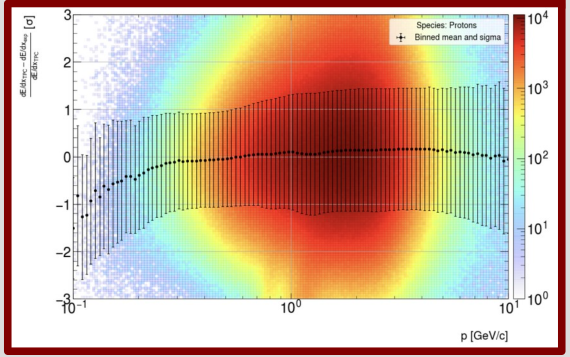


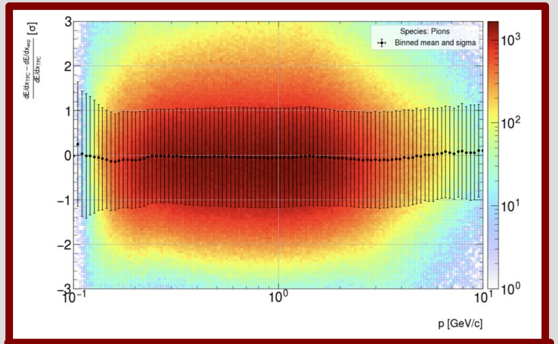


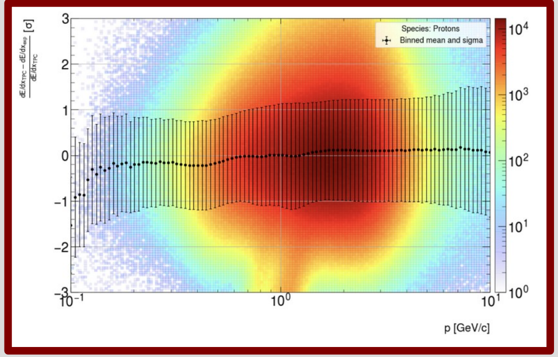
Performance in Pb-Pb collisions

- Pb-Pb collisions: High detector occupancy, much more challenging environment to produce clean signals for training
- Most recent reconstruction pass for 2023 Pb-Pb data shows strong improvement in low-momentum tails + resolution estimation when NN corrections are applied



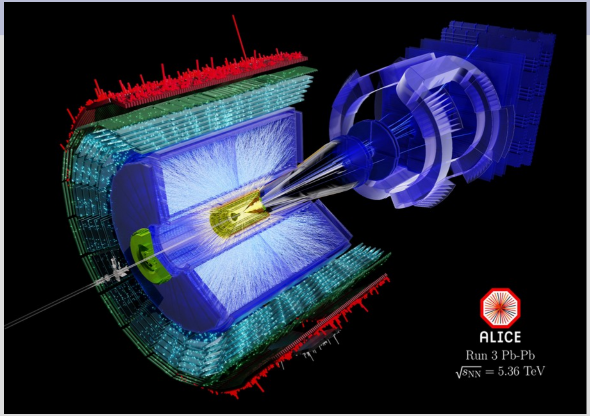


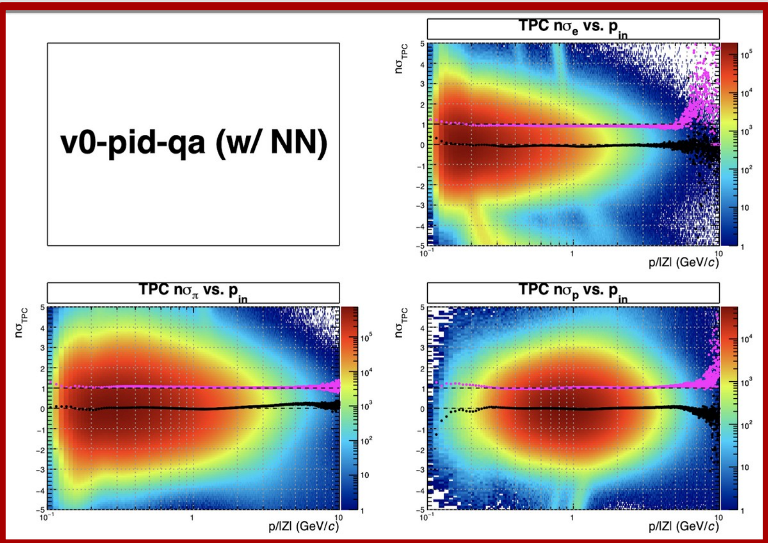




Summary

- Neural network provides a robust 6-dimensional correction to PID response in ALICE TPC – key input for many physics analyses
- Use of ONNXRuntime allows simple interface + portability between Python-based neural network training and C++-based O^2 analysis framework
- Availability of GPU resources at GSI allows faster training; inference still possible + performant on CPUs on LHC Grid
- Corrections with this method released for trigger skimming on most recent datasets, with refinements added for each new reconstruction pass of data





-- backup --

Method in LHC Runs 1+2 (2009-2018)

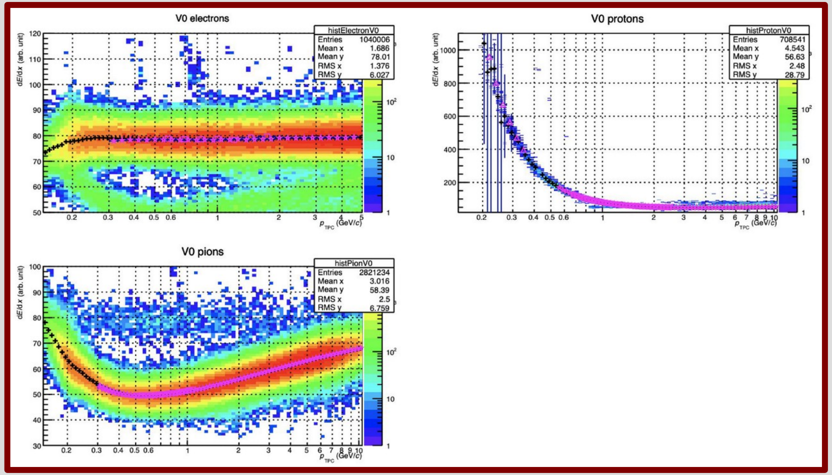
- dE/dx splines of mean detector response were fit from clean samples of "V⁰" decays (Λ, K^0_S , γ)

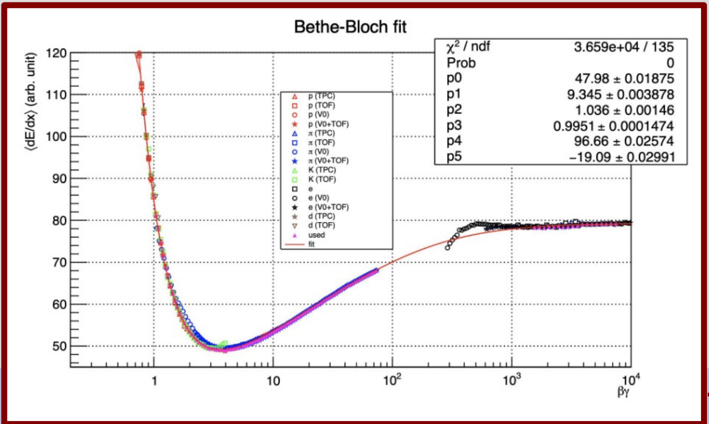
  → Clean samples of electrons, pions, protons using cuts on decay kinematics + topology

  → Other species (K, d) are included using TOF response cuts where available

  → Series of separate correction factors layered on top of each other based on momentum region, incident angle, multiplicity, ...

  → Result is spline of expected dE/dx as function of $\beta\gamma$, stored in analysis software and used at analysis time

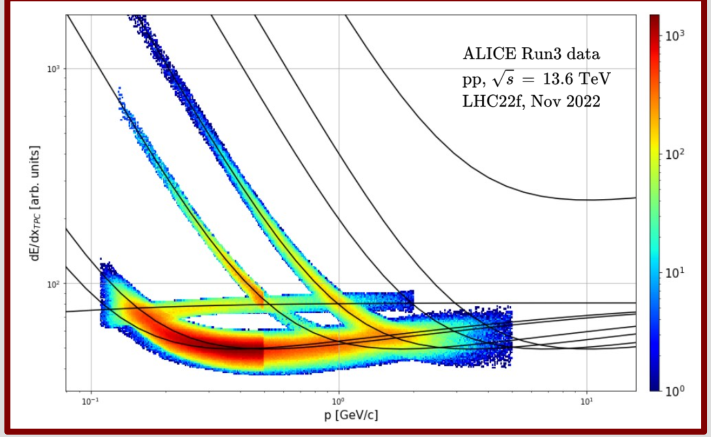


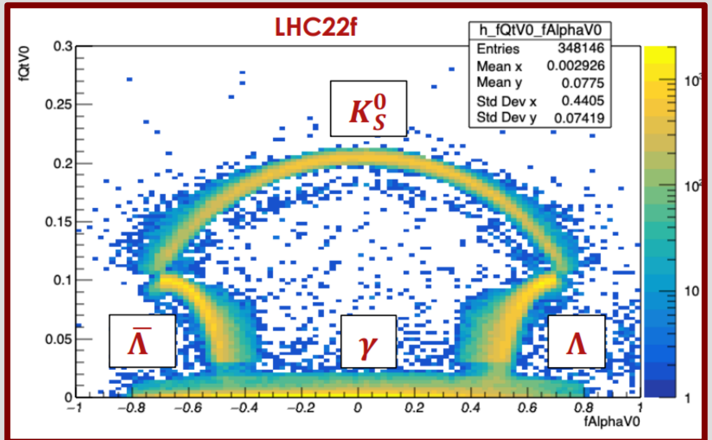


Basic dE/dx parameterisation in O²

- V0 / TPC only / TPC+TOF used for initial selection ("basic skimming") for first pass of BB fit
- Custom tree skimmer on Hyperloop:

  - V^0 decays of Λ and K_0 deliver pions and protons; ɣ-conversion produces electron sample
  - TPC / TPC+TOF cuts to enrich samples in regions of clear separation between species
- Requirements: Representative input sample, "basic" initial BB parameters (sampled from LHC22m)
- Sampling full period due to varying conditions between individual runs (interaction rate, MIP point, …)
- Mean estimation + hyperparameter optimization framework for the calibration of the initial parameters

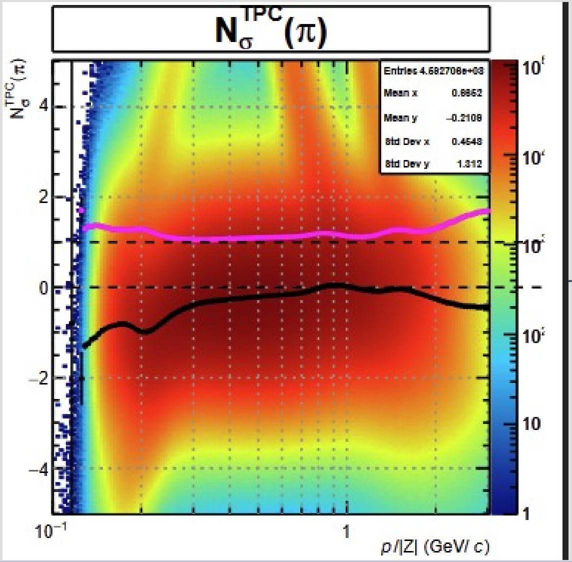


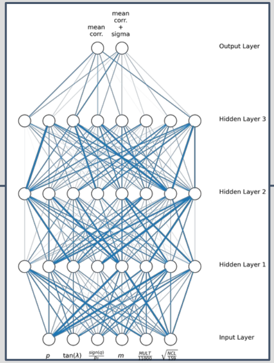


Mean correction with neural networks

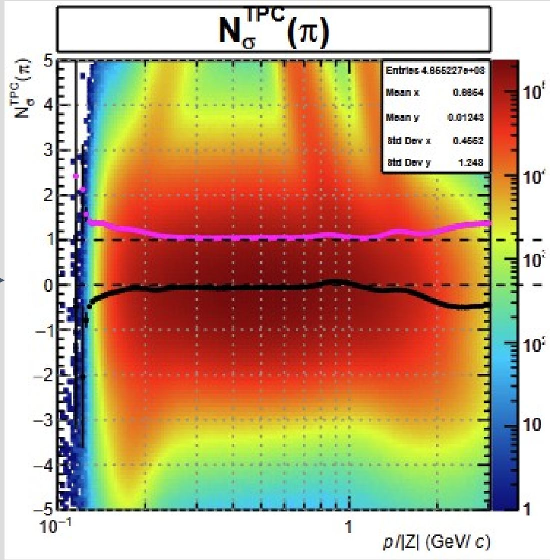
- Input: Track parameters
  (p, tan(λ), sign(q)/p_T, mass hypothesis, normalized multiplicity, normalized number of clusters)
- Output: dE/dx of identified particles as a ratio to the Bethe-Bloch parametrisation as correction factor to expected curve

Bethe-Bloch parametrisation





Bethe-Bloch + NN



MC: TuneOnData implementation

- Monte Carlo: Required for correction of efficiencies in analysis selections

- However, anchoring does not reflect real distribution of signals – data calibrations `

- Solution: “Tuning” of MC signal based on expectations from data:
  - MC truth used to define particle species
  - MC signal sampled randomly by a Gaussian with expected signal + width defined by BB+NN from data
  - Sampled signal used for all n_σ calculations instead of stored value