Tracking - II

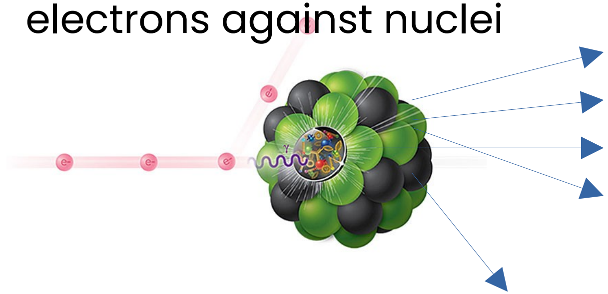
2nd European School on the Physics of the EIC and Related Topics

Benicàssim, July 23rd - 24th 2025

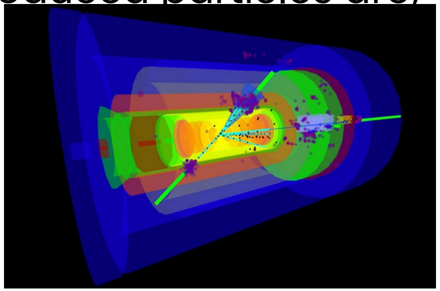
F. Bossù (CEA/Irfu)

From collisions to analysis

We start by colliding electrons against nuclei

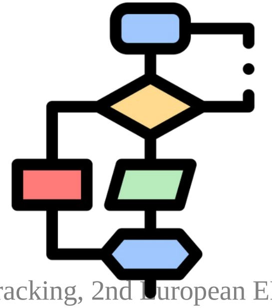


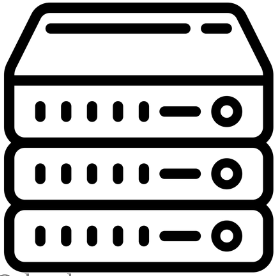
We need detectors to know where and what produced particles are;

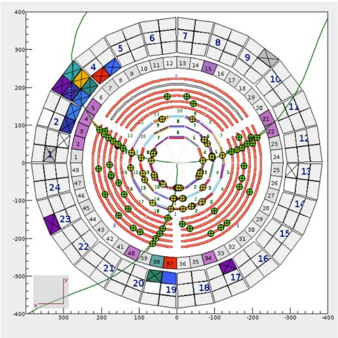


…electronics to read, decode and store signals from sensors;

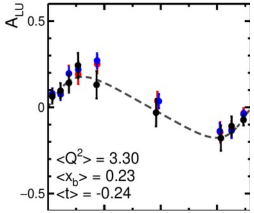
… computing and algorithms to extract features from raw data;





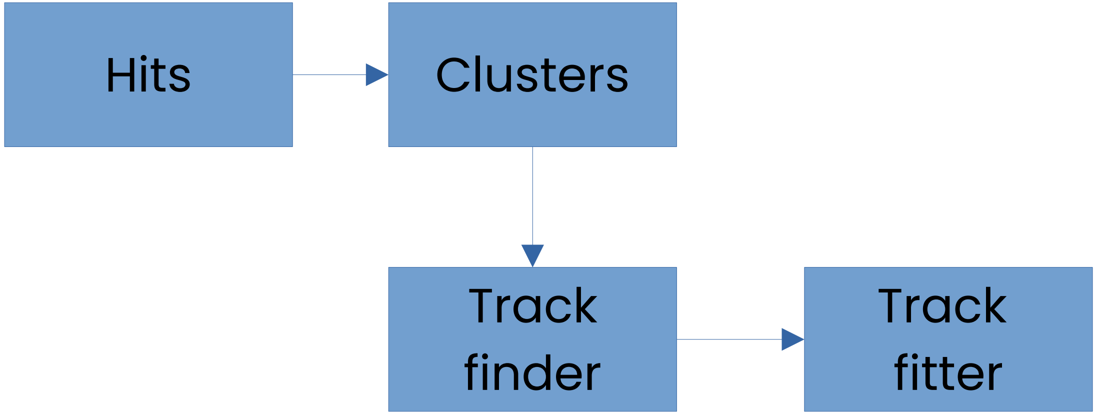


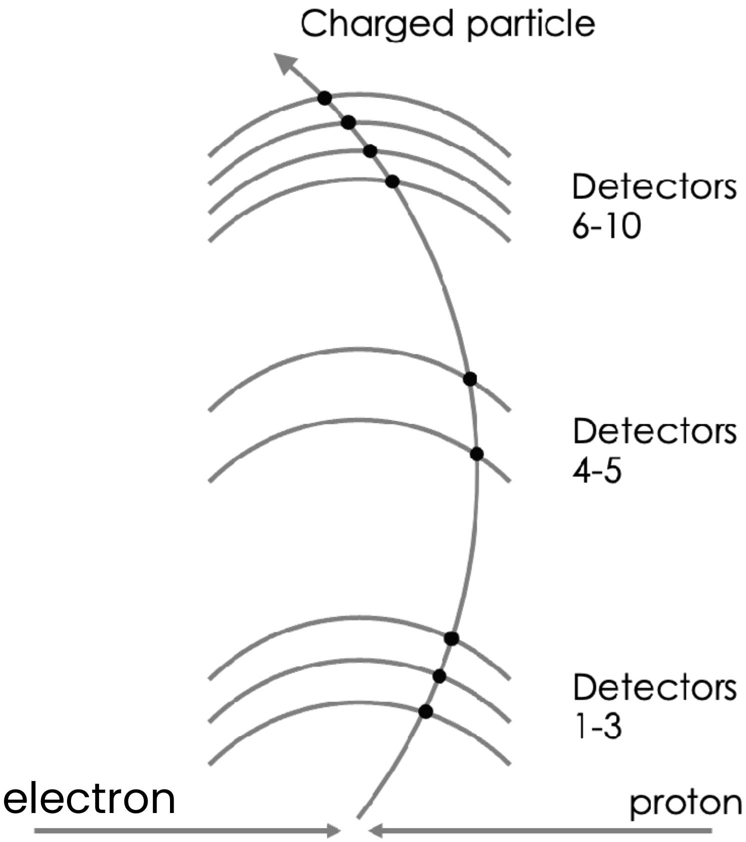
…and person-power to analyze the data and produce physics results



From detectors to tracks

- We learned how to detect where charged particles pass
- Now, from the detector information we need to find back the particle trajectory and extract its momentum



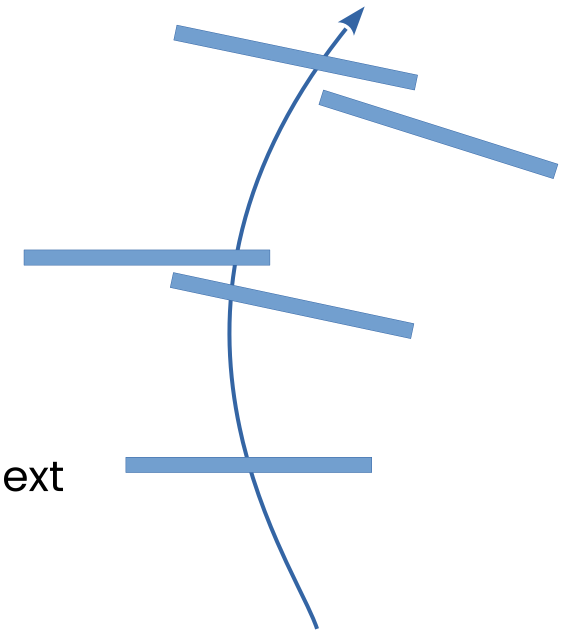


Track model and transport

- In a magnetic field, a charged particle trajectory can be described locally as an helix
- We need 5 parameters to describe it

$$\vec{p} = (l_1, l_2, \theta, \phi, Q/P)$$

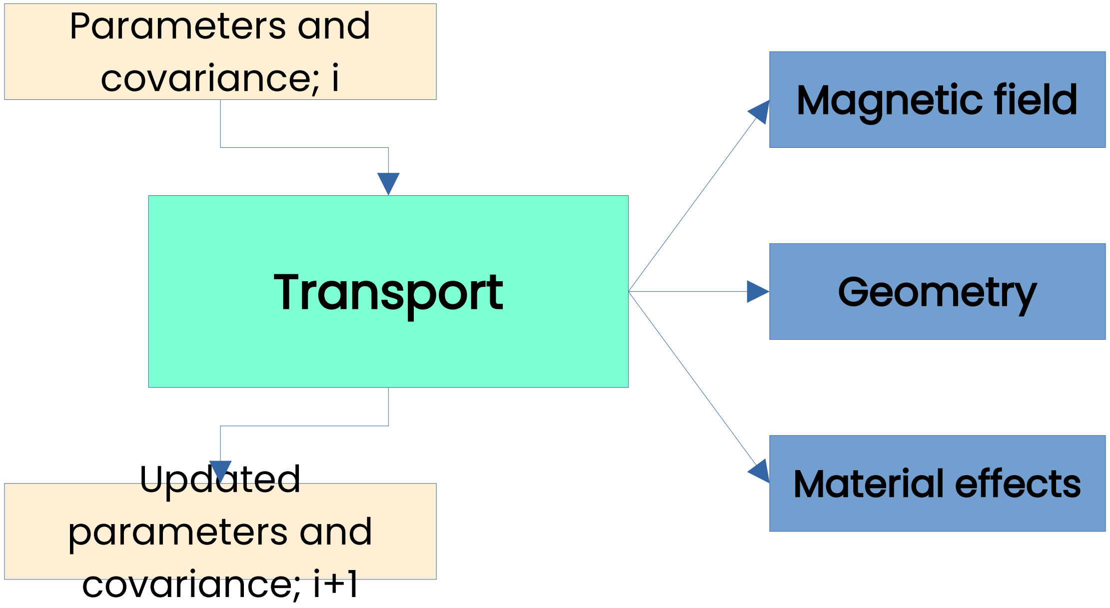
l1 et l2 local coordinates on a surface (can be a plane, a cylinder, a disc…), θ and ϕ, the direction, and Q/P the curvature

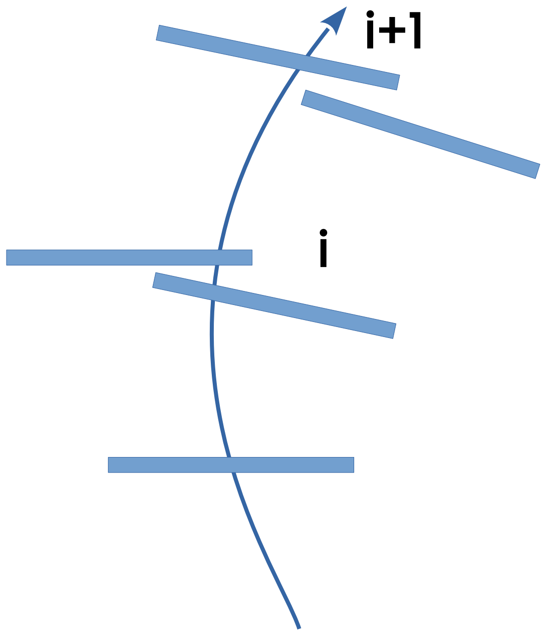


- From a given measurement, we need to transport the track to the next layer to know its intersection and the parameters at that surface
- In a constant B field, analytical formulas
- In an inhomogeneous field, numerical integration (Runge-Kutta) of differential equations

Track model and transport

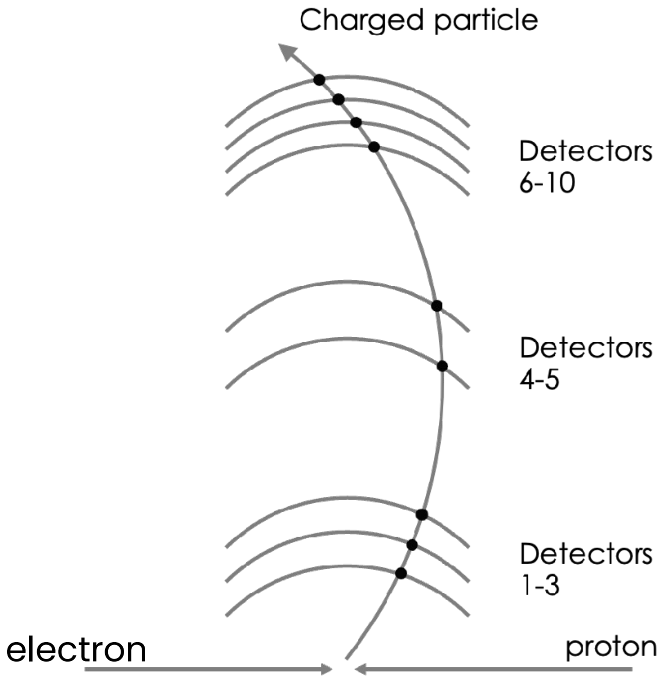
- We need an algorithm that propagates the track parameters and their covariances from layer to layer





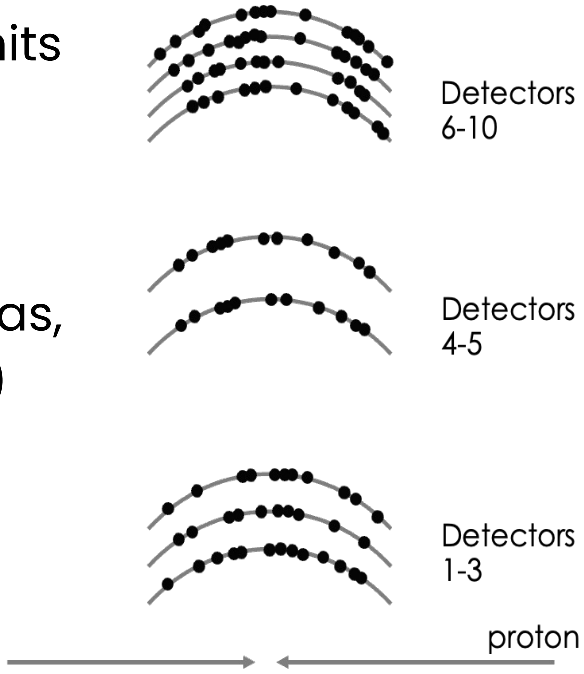
Finding tracks

- It may looks easy, but…



In reality we have many hits from:

- Multiple particles
- Beam induced backgrounds (beam-gas, synchrotron radiation…)
- Intrinsic detector noise

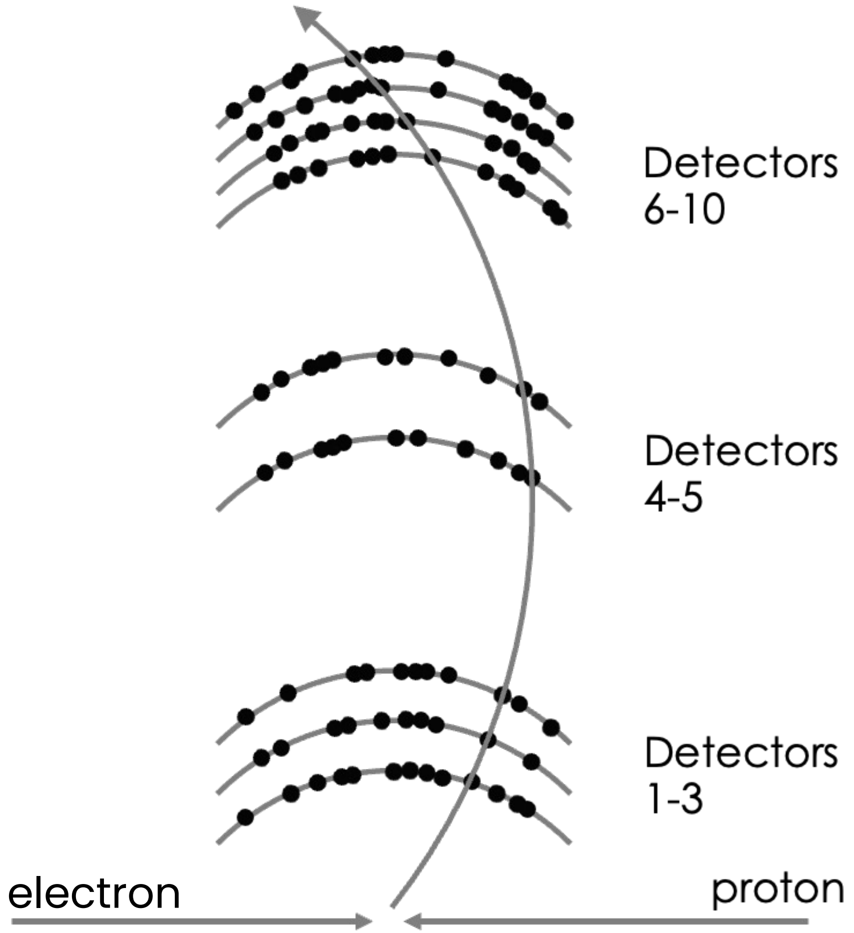


Finding tracks

We need an algorithm that looks through all the hits and sort them out in track candidates.

Key parameters for **a track finding algorithm**:

- Efficiency: what fraction of *true* particles have been found
    - Hit matching: a particle is found if a fraction of true hits are used
    - Parameter matching: a particle is found if rec parameters are close to the true ones
- Ghost rate: how many tracks are reported that are not true particles
- Clone tracks: particles reconstructed more than once, due to some hit sharing or kinks in track



Finding tracks

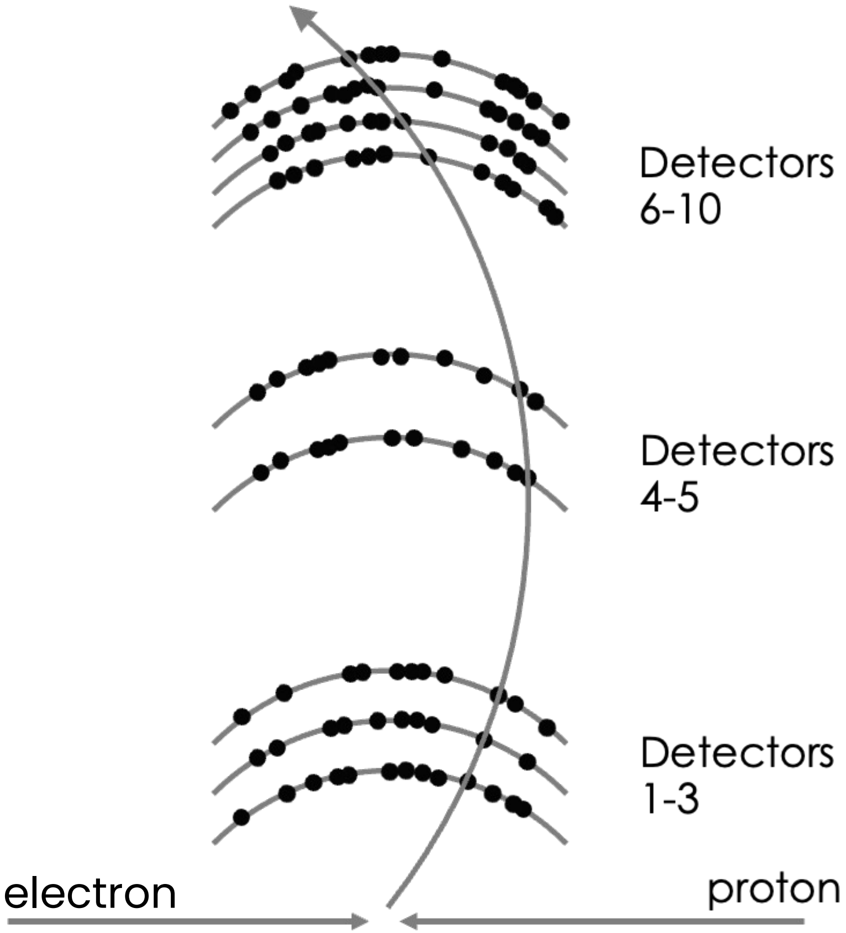
Two main approaches:

Global methods:

- Treat hits in all detector layers simultaneously
- Find all tracks at the same time

Local methods:

- Start with construction of seeds, sub-sets of hits
- Add hits by extrapolation the seed trajectory through the other detector layers
- Improve seeds after each added hit



Track finding – Hough transform

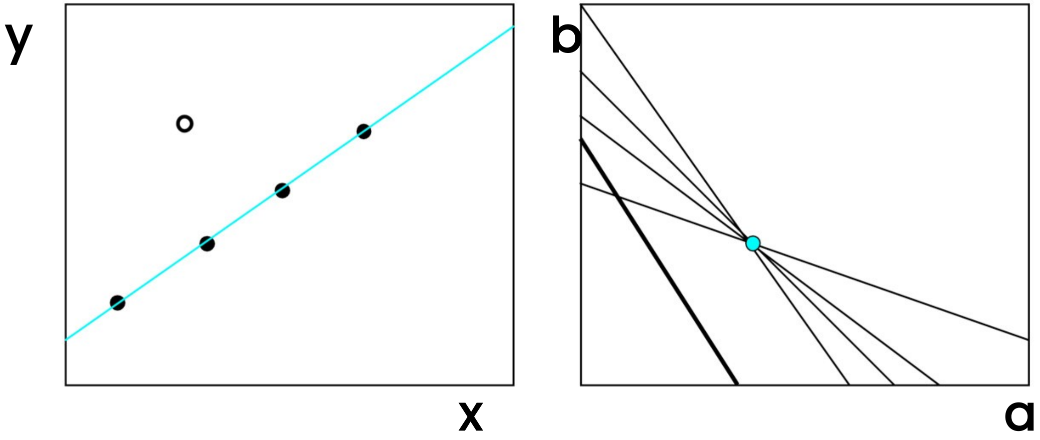
Hough transform takes all the hits at once

- Measured space points → lines in feature space

Example in 2D

Track model: $y = a\,x + b$;

- measurements: (x_i, y_i) ;
- Features: $b_i = y_i - a_i * x_i$



Finding tracks:

- For each data point (x_i, y_i) build a line in feature space
  - Fill an histogram using the line
- Look for maxima in the parameter space histogram

Easy to implement.

It works in higher dimensions (for circles too) but becomes quickly time consuming to find the maxima

Track finding – Neural networks

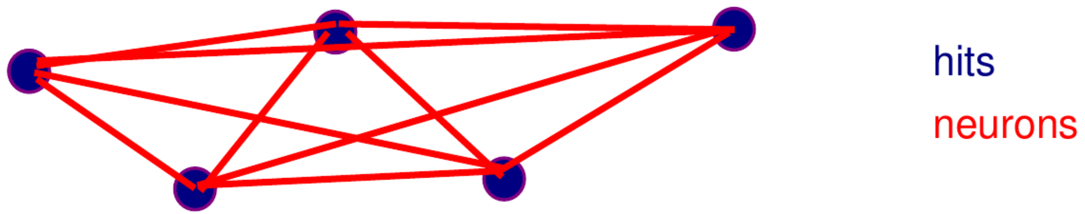
Artificial neural networks look for global pattern using local (neighbor) rules.

- Neurons are the lines connecting two hits
- Update the state of a neuron based on connected neurons

Main feature: can be built robust against noise and inefficiencies

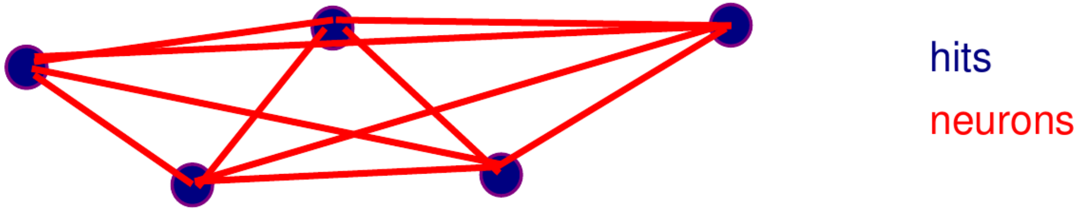
Example Denby-Peterson neural network:

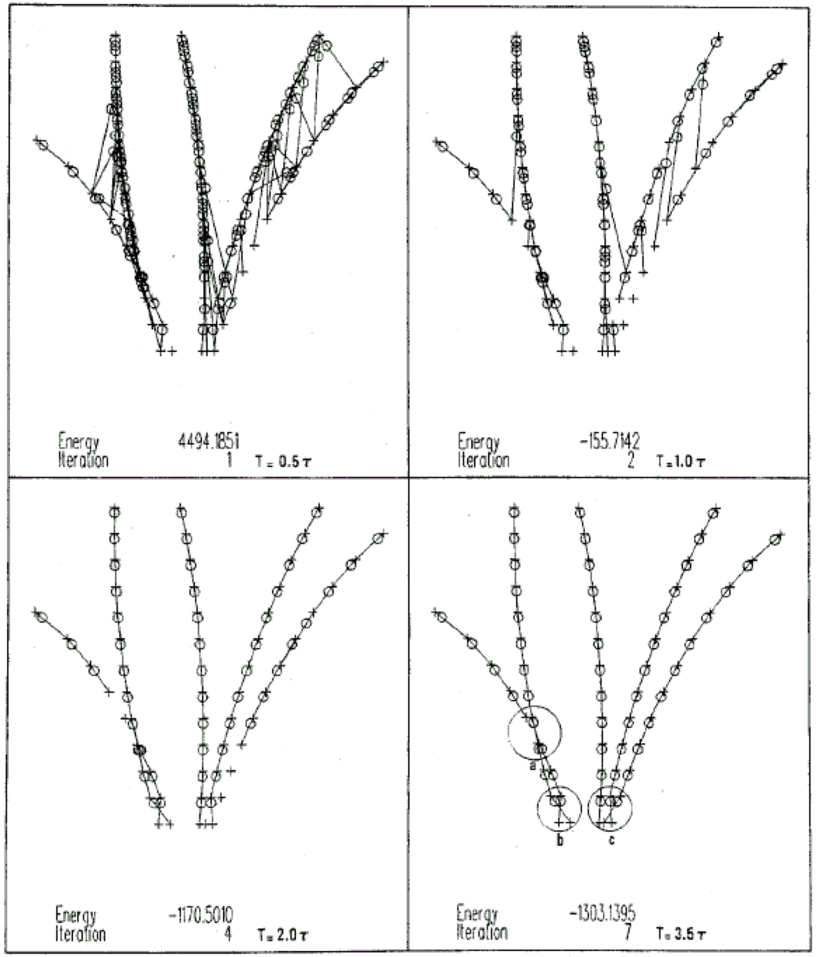
- Define an “energy” function of number of connected neurons, angle between neurons, …
- Update the neuron states to try to minimize the energy



Track finding – Neural networks

Example of a Demby-Peterson network evolution

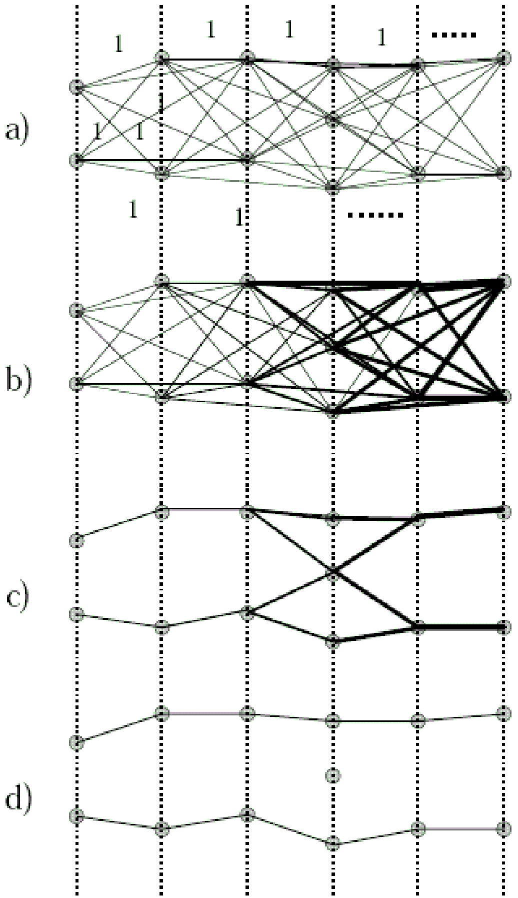




Track finding – Cellular Automata

Simpler to Denby-Peterson.

Example:
CATS algorithm
used in Hera-B



Build a cellular automata:

- A cell is the link between two hits
- Each cell is initialize with a state S=1
- Define a rule about which cells can belong to the same track. Ex, just on angle

Evolution:

- Update until stable, all cell states according to

$$S = 1 + \max\left\{S_{neighbours} \middle| \theta_{ijk} < \theta_{max}\right\}$$

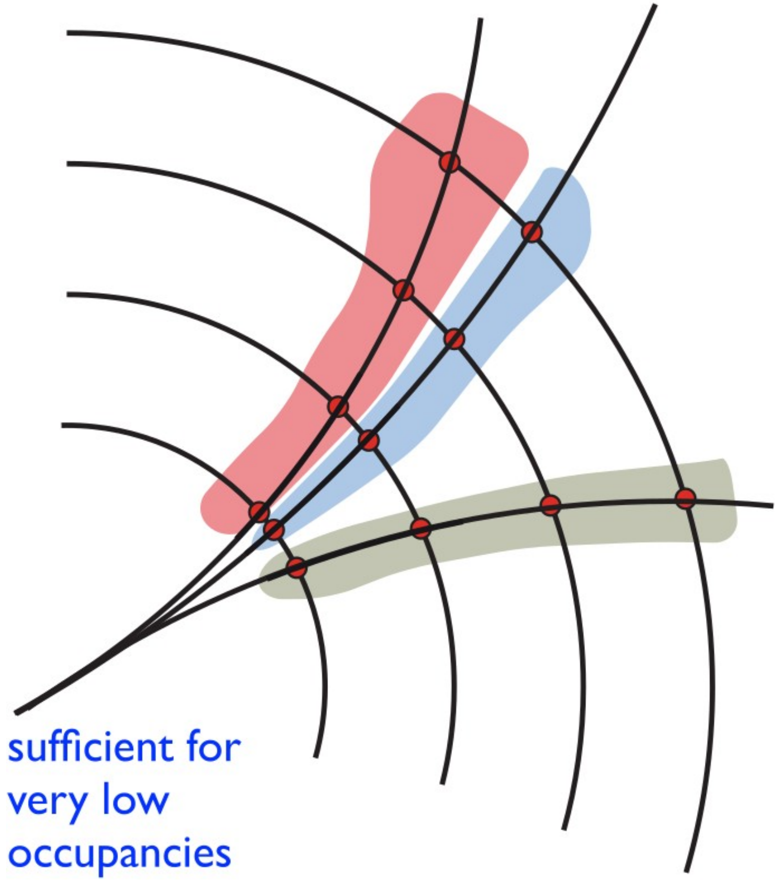
Track finding:

- Start from cells with highest states and follow neighbors with Sn = S - 1

Track finding – Local methods

Track Roads

- Find seeds ~ combination of 2-3 hits
- Build roads along the likely trajectory
- Select hits on various layers inside the road to build the track candidate



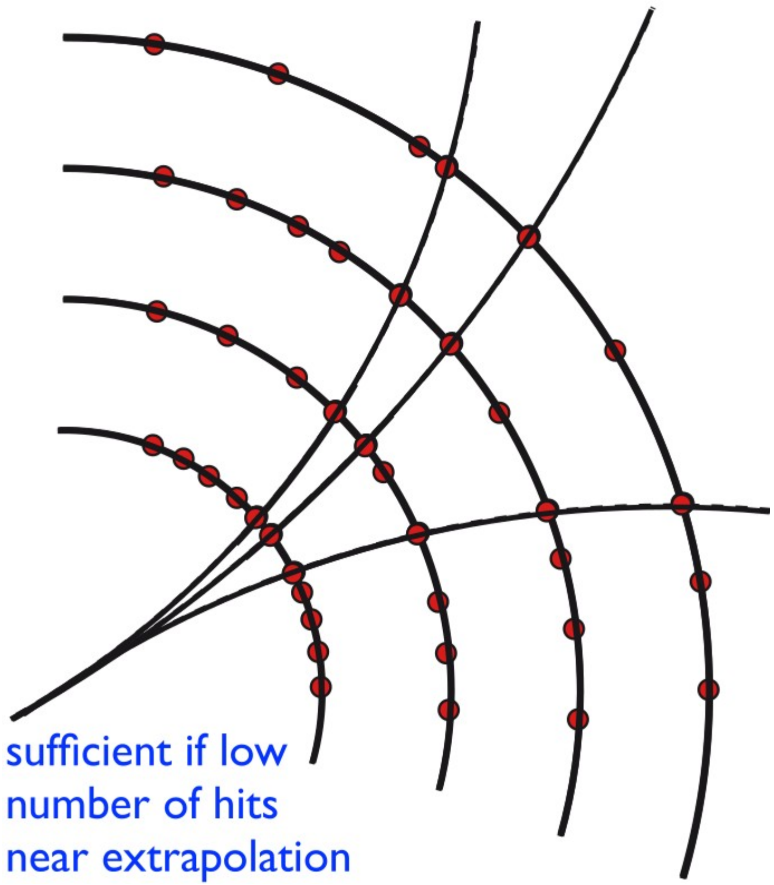
Track finding – Local methods

Track Roads

- Find seeds ~ combination of 2-3 hits
- Build roads along the likely trajectory
- Select hits on various layers inside the road to build the track candidate

Track following

- Find seeds ~ combination of 2-3 hits
- Extrapolate seed trajectories
- Select hits on various layers close to the seed trajectory to build the track candidate



Track finding – Local methods

Track Roads

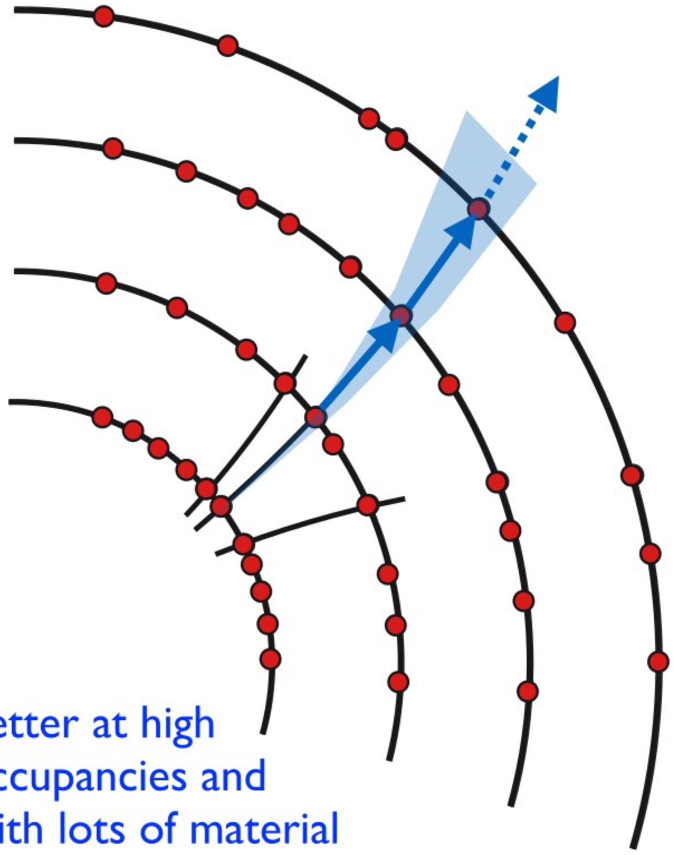
- Find seeds ~ combination of 2-3 hits
- Build roads along the likely trajectory
- Select hits on various layers inside the road to build the track candidate

Track following

- Find seeds ~ combination of 2-3 hits
- Extrapolate seed trajectories
- Select hits on various layers close to the seed trajectory to build the track candidate

Progressive track finder

- Find seeds ~ combination of 2-3 hits
- Extrapolate seed trajectories to the next layer
- Select best hit and update trajectory
- Repeat until the last layer to build candidates



better at high occupancies and with lots of material

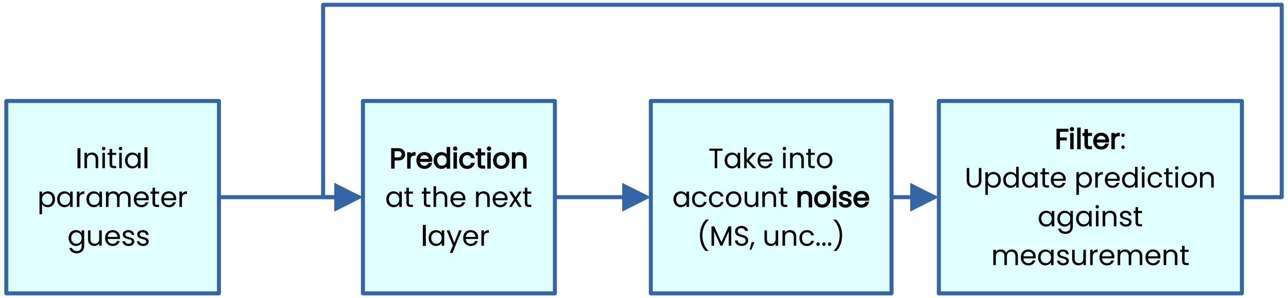
Track fitting

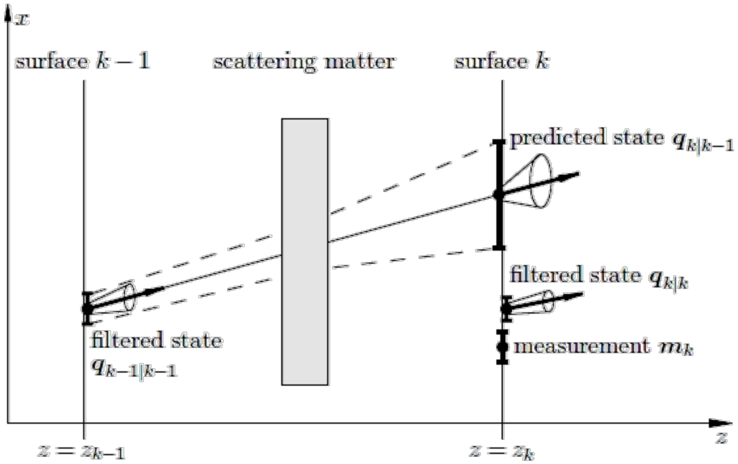
The track finder provides us a list of sets of measurements, i.e. the track candidates

- We know the track model, then we can use to good old χ^2 minimization for a global fit.
- It can also handle material effects, such as multiple scattering, but it is computationally hard: it needs the inversion of large matrices.

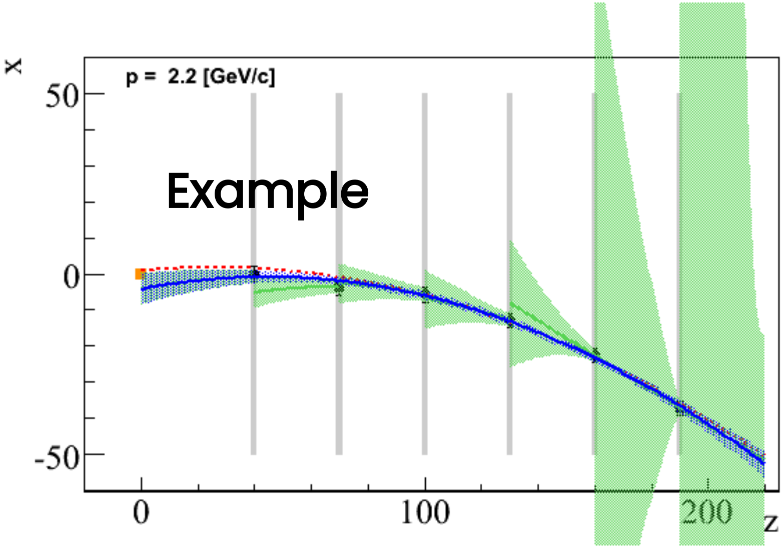
Alterative option: use a **Kalman Filter**.

- It is a *progressive* way of performing a least square fit



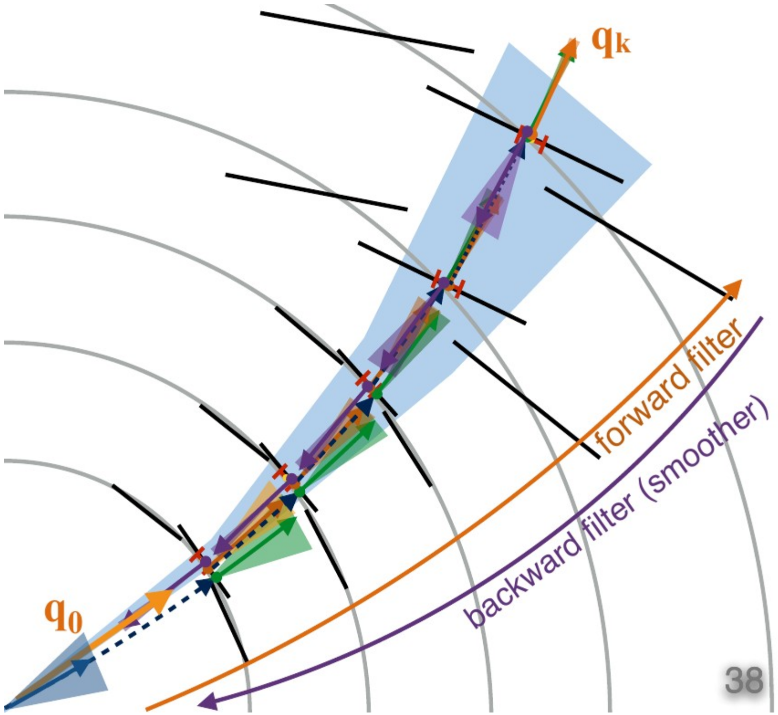


Track fitting



A global fit provides the best track state at everywhere along the trajectory

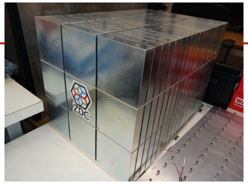
The Kalman Filter will provide the best state only after processing all measurements

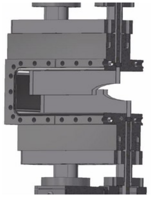


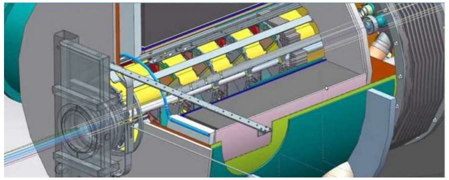
We need to perform twice the KF: forward and backward. Smoothing (weighted average at every node)

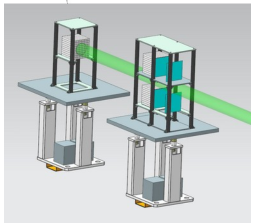
The ePIC detector

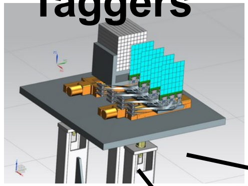
90m long set of detectors.
Tight integration with the accelerator in the interaction region

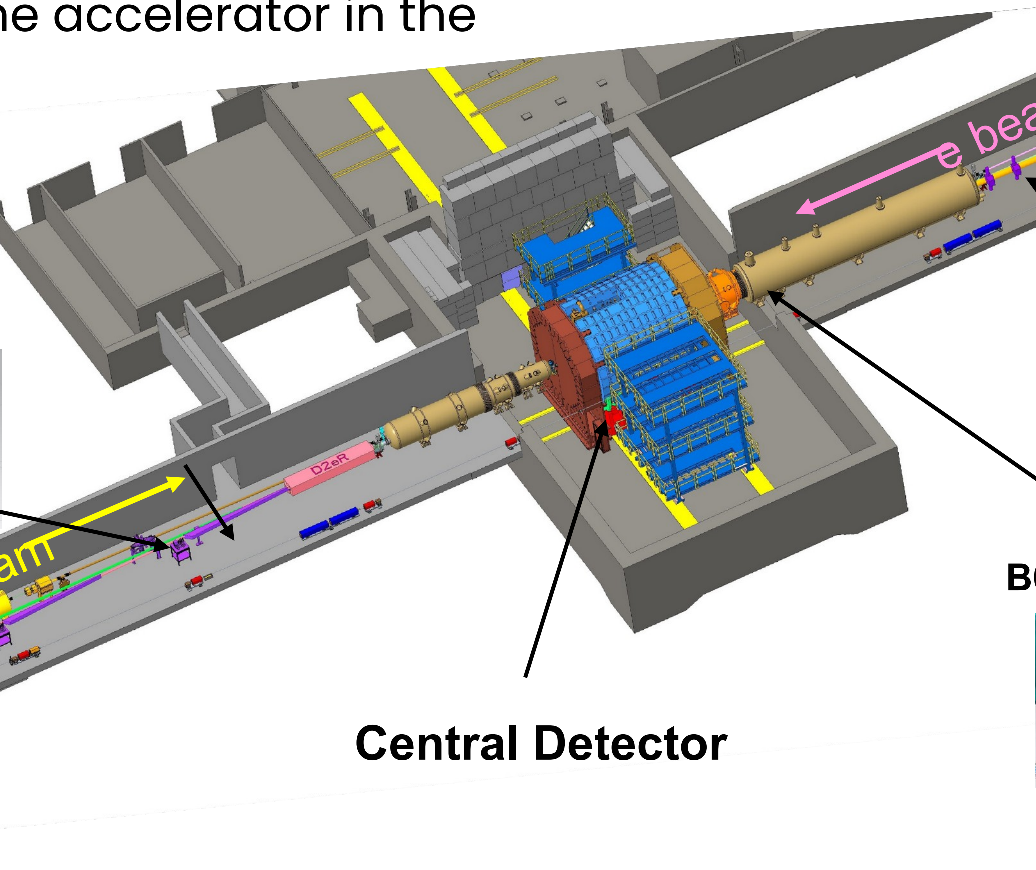












The ePIC central detector

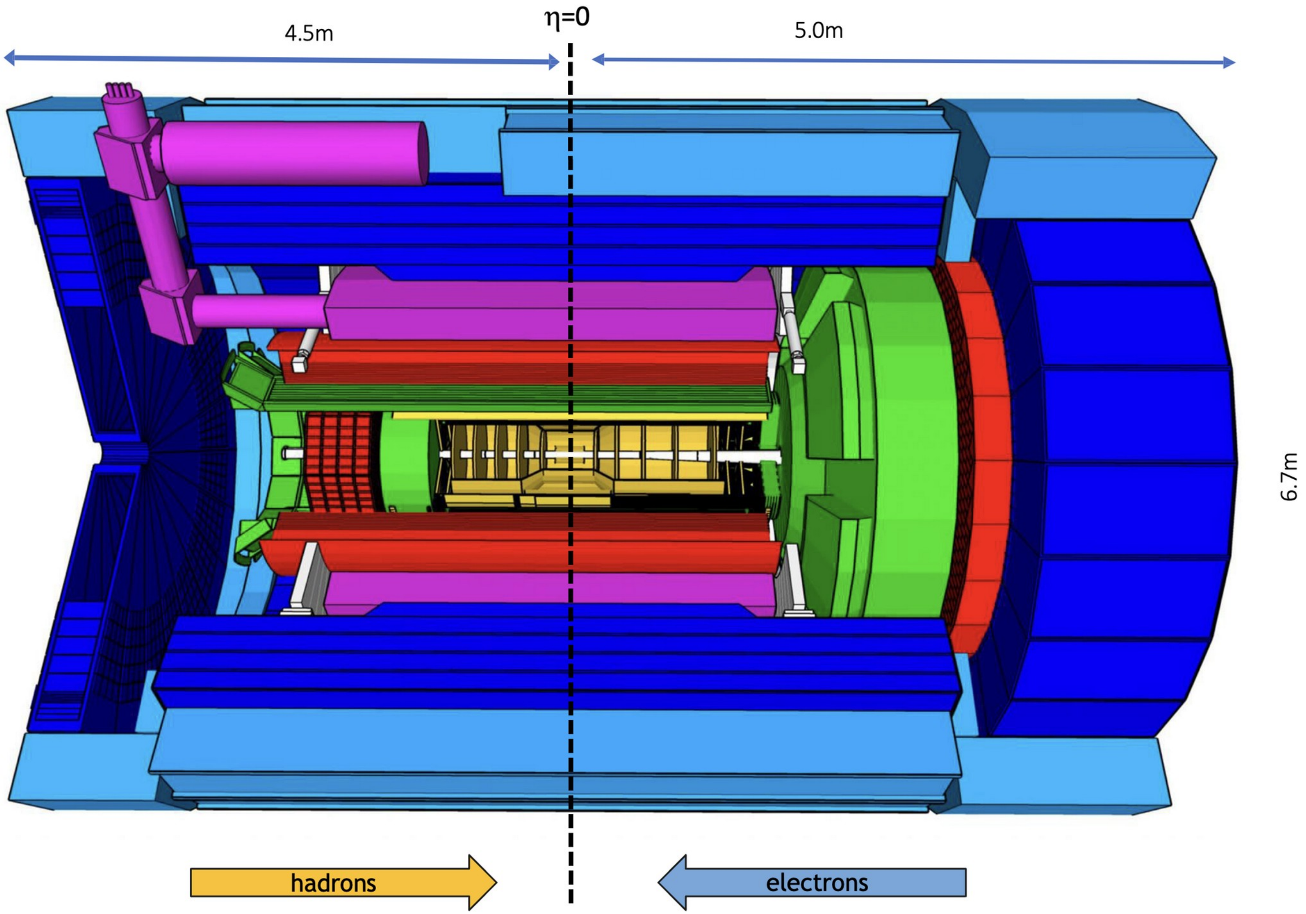
Magnet
- New 1.7 T SC solenoid, 2.8 m bore diameter

Tracking
- Si Vertex Tracker MAPS wafer-level stitched sensors (ALICE ITS3)
- Si Tracker MAPS barrel and disks
- Gaseous tracker: MPGDs (μRWELL, MMG) cylindrical and planar

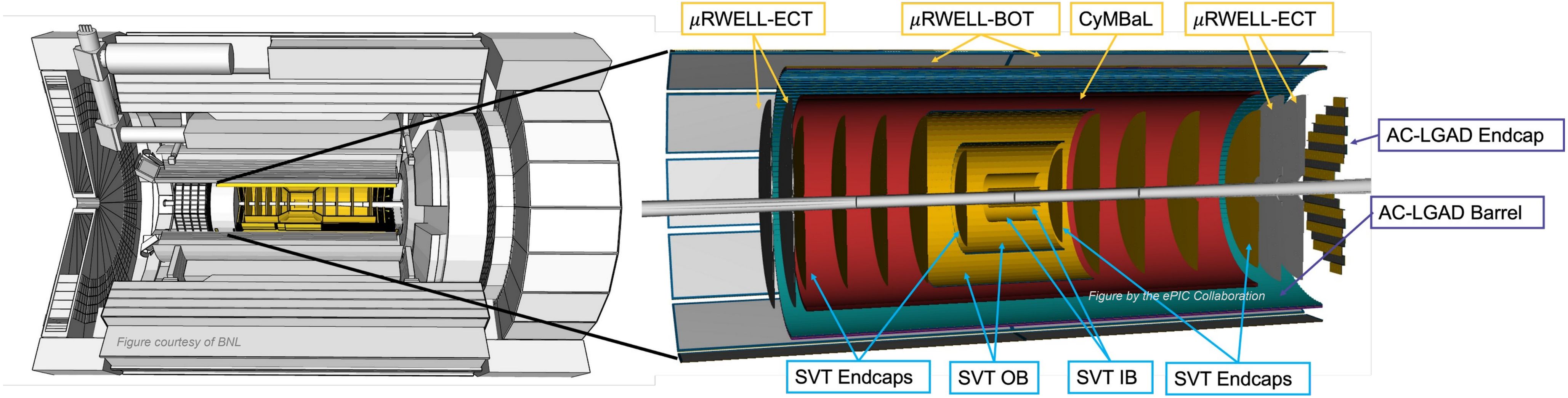
PID
- high performance DIRC (hpDIRC)
- dual RICH (aerogel + gas) (forward)
- proximity focussing RICH (backward)
- ToF using AC-LGAD (barrel+forward)

EM Calorimetry
- imaging EMCal (barrel)
- W-powder/SciFi (forward)
- $PbWO_4$ crystals (backward)

Hadron calorimetry
- FeSc (barrel, re-used from sPHENIX)
- Steel/Scint – W/Scint (backward/forward)



The ePIC tracking system



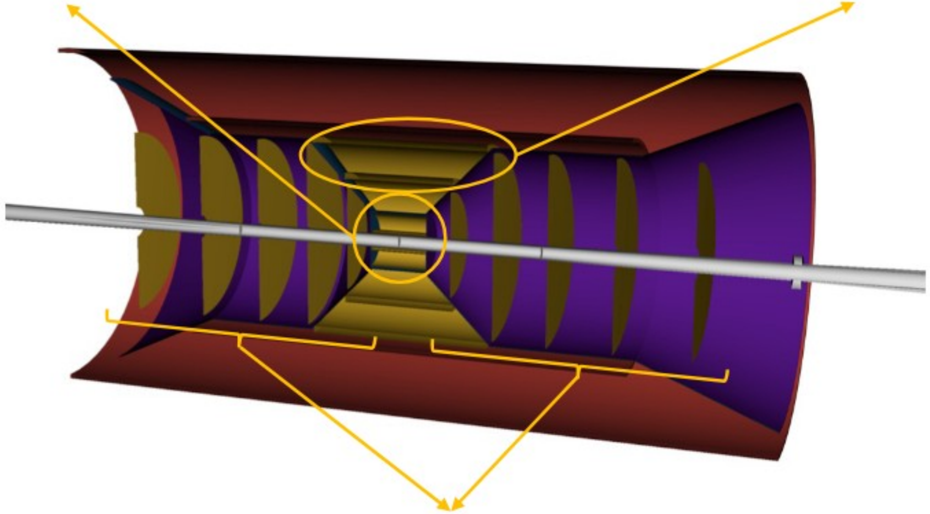
Tracking Requirements:

- High vertex resolution 20μm/pT⊕ 5μm
- Excellent momentum resolution 0.05%pT⊕0.5%
- Low material budget
- Good pattern recognition efficiency
- Sufficient time resolution to resolve 10ns bunch crossing
- Good angular resolution for the DIRC

ePIC SVT

Inner Barrel (IB)
- 2 curved silicon vertex layers
- 1 curved dual-purpose layer

Outer Barrel (OB)
- 1 stave-based sagitta layer
- 1 stave-based outer layer



Electron/Hadron Endcaps (EE, HE)
- 5 disks on either side of the IP

Total (active) area ~ 8.5 m²

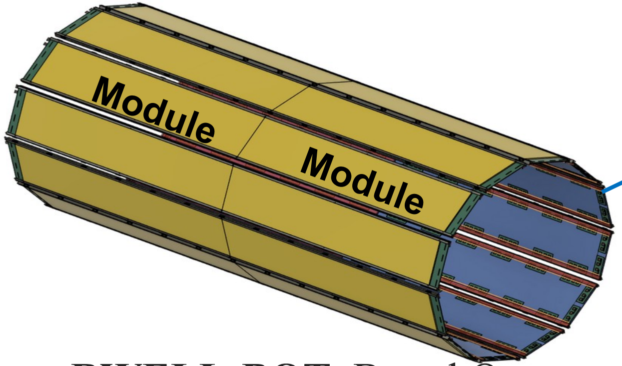
Inner Barrel (IB)
Three layers, L0, L1, L2,
Radii of ~ 36, 48, 120 mm
Length of approx. 27 cm
x/X_0 ~ 0.05% per layer
Curved, thinned, wafer-scale sensor

Outer Barrel (OB)
Two layers, L3, L4
Radii of ~ 27 and 42 cm
X/X_0 ~0.25% and ~0.55%
More conventional structure w. staves

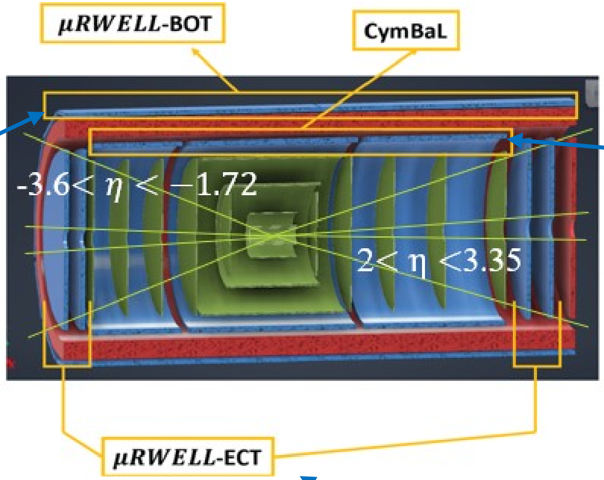
Electron/Hadron Endcaps (EE, HE)
Two arrays with five disks
x/X_0 ~0.25% per disk
More conventional structure w. halves

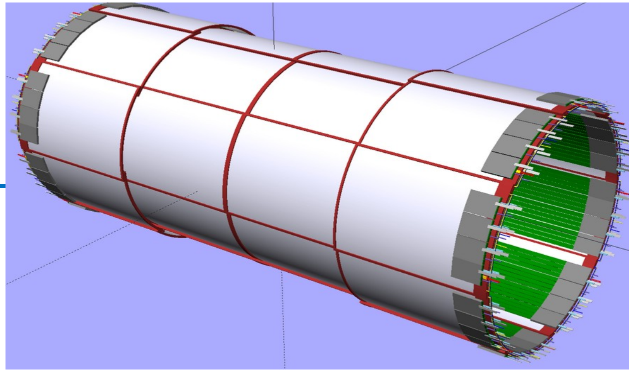
Based on the MAPS technology currently developed for the ALICE ITS3 Upgrade
Pixel size ~20x20μm
Integration time ~2μm

ePIC Micro-pattern gaseous detectors

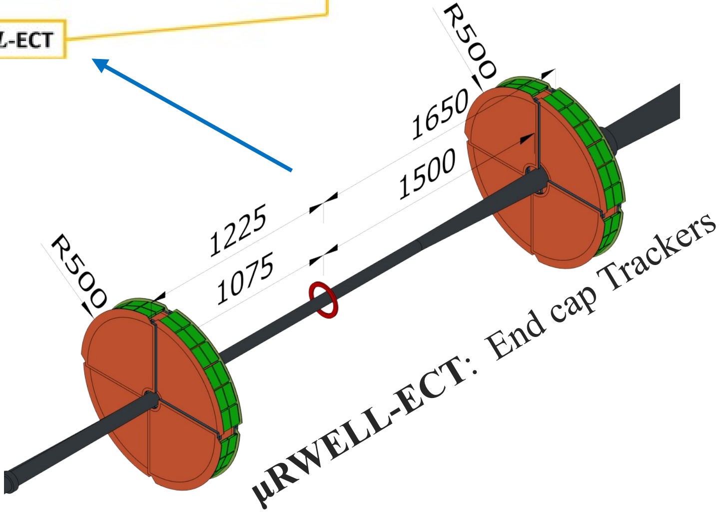


µRWELL-BOT: Barrel Outer Tracker





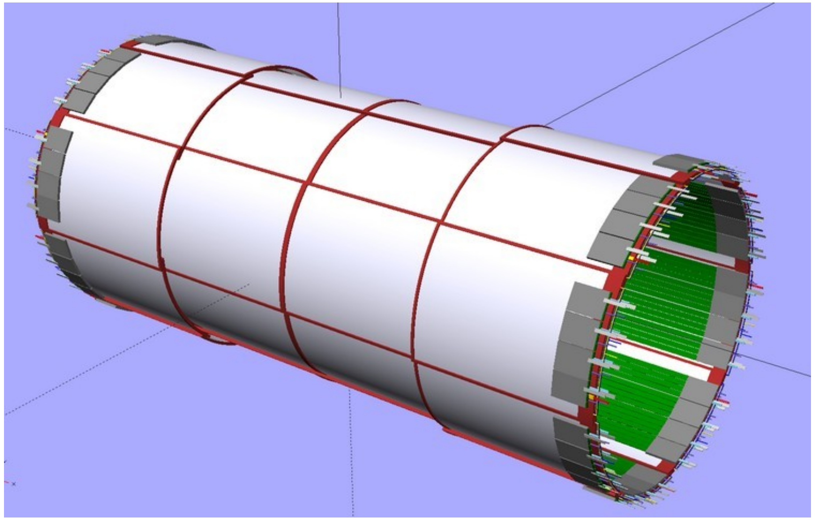
CyMBaL: Inner Barrel Layer



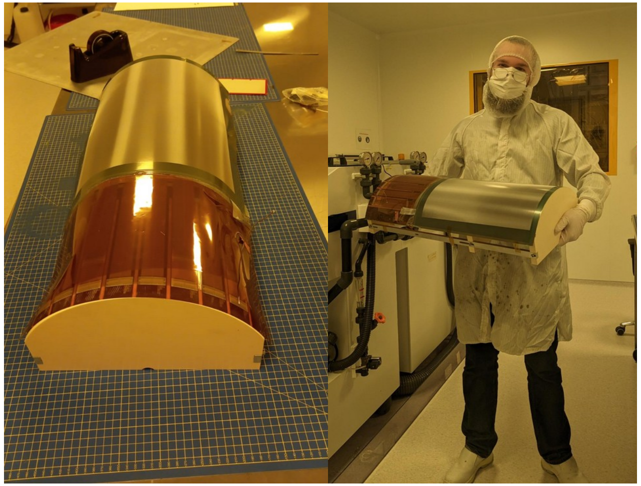
ePIC MPGD trackers provide:

- **Good timing capability (~10 – 20 ns)** to help pattern recognition in high background.
- additional hit points capabilities for tracking
- Precision tracking to enhance hpDIRC performance

ePIC MPGDs – CyMBaL



Based on cylindrical resistive Micromegas technology used in CLAS12 at JLab



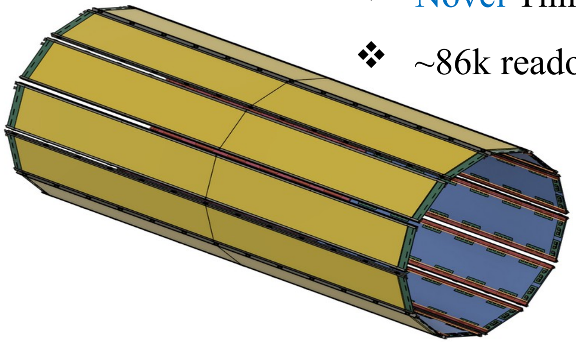
- **32 module: 8 modules in ϕ times 4 modules in z**
- Overlaps in ϕ and in z for hermeticity
- 1024 readout channels/module
- **32K readout channels**
- Rmin = 55 cm; Rmax = 60.5 cm

Module dimensions
Z = 67 cm
R*phi = 48 cm
Active zone dimensions
Z = 59 cm
R*phi = 46 cm

ePIC MPGDs – Barrel Outer Tracker

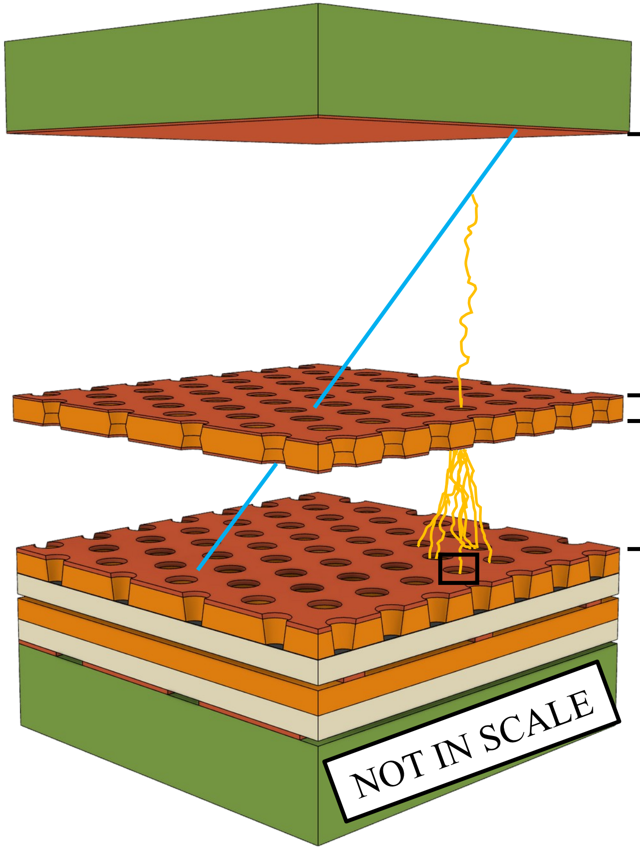
24 planar detector modules:

- 12 sectors in r*φ × 2 modules in z ➔ No overlaps
- R_{min} = 72.5 cm; R_{max} = 75 cm
- Novel Thin-gap GEM-µRWELL hybrid technology
- ~86k readout electronic channels



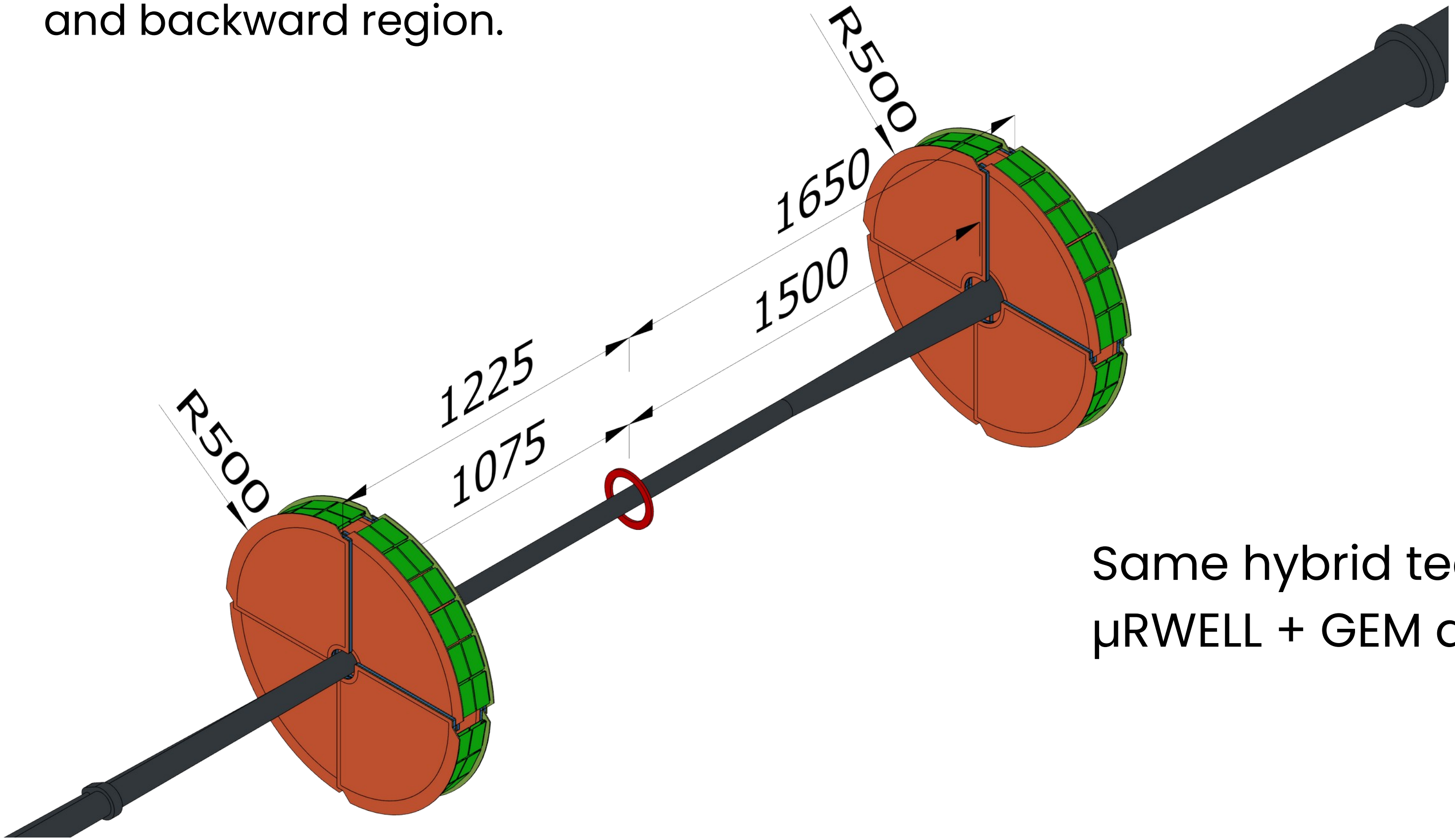
Module

- Overall dimension: 180 cm in z × 36 cm in r*φ
- Active area: 170 cm in z × 33 cm in r*φ
- 14 FEBs / module Total: 3,584 chs / module



ePIC MPGDs – End Cap Tracker

Two pairs for disks in the forward and backward region.

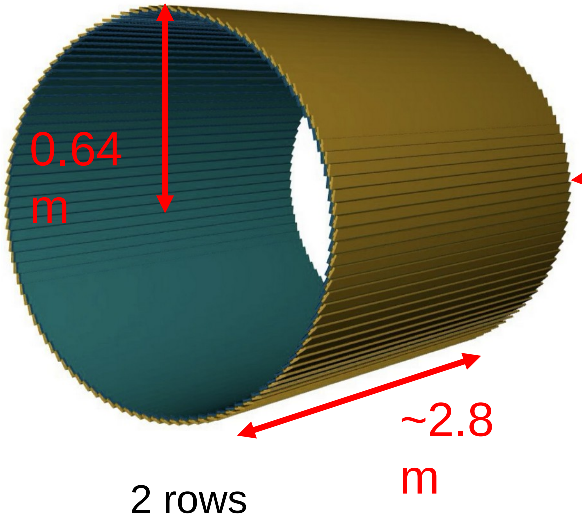


Same hybrid technology µRWELL + GEM as for BOT

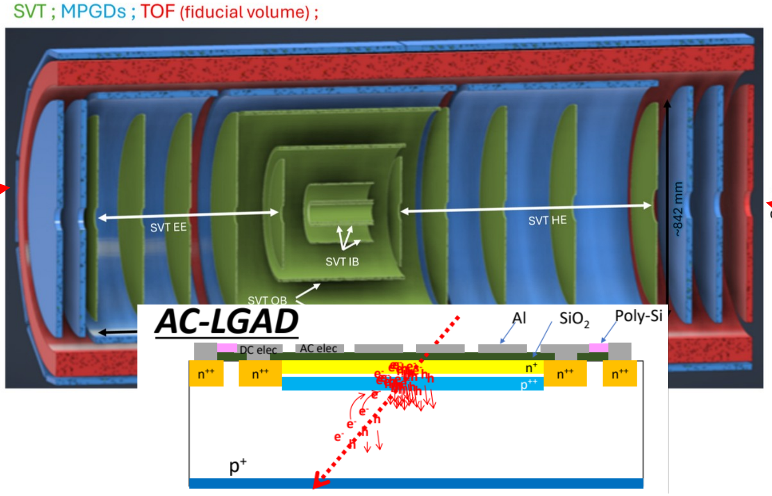
ePIC AC-LGAD: tracking and time of flight

Barrel-TOF (BTOF)

-1.33<η<1.74

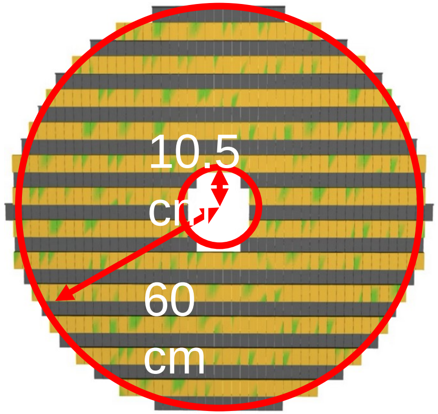


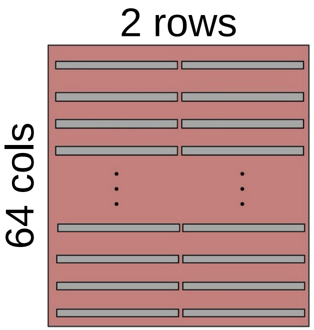
Resolutions: σt ~ 30ps; σx < 30μm;



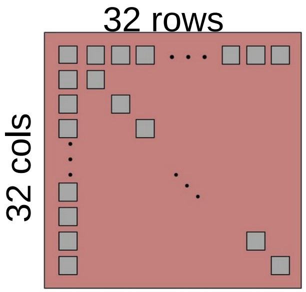
Forward-TOF (FTOF)

1.84<η<3.61



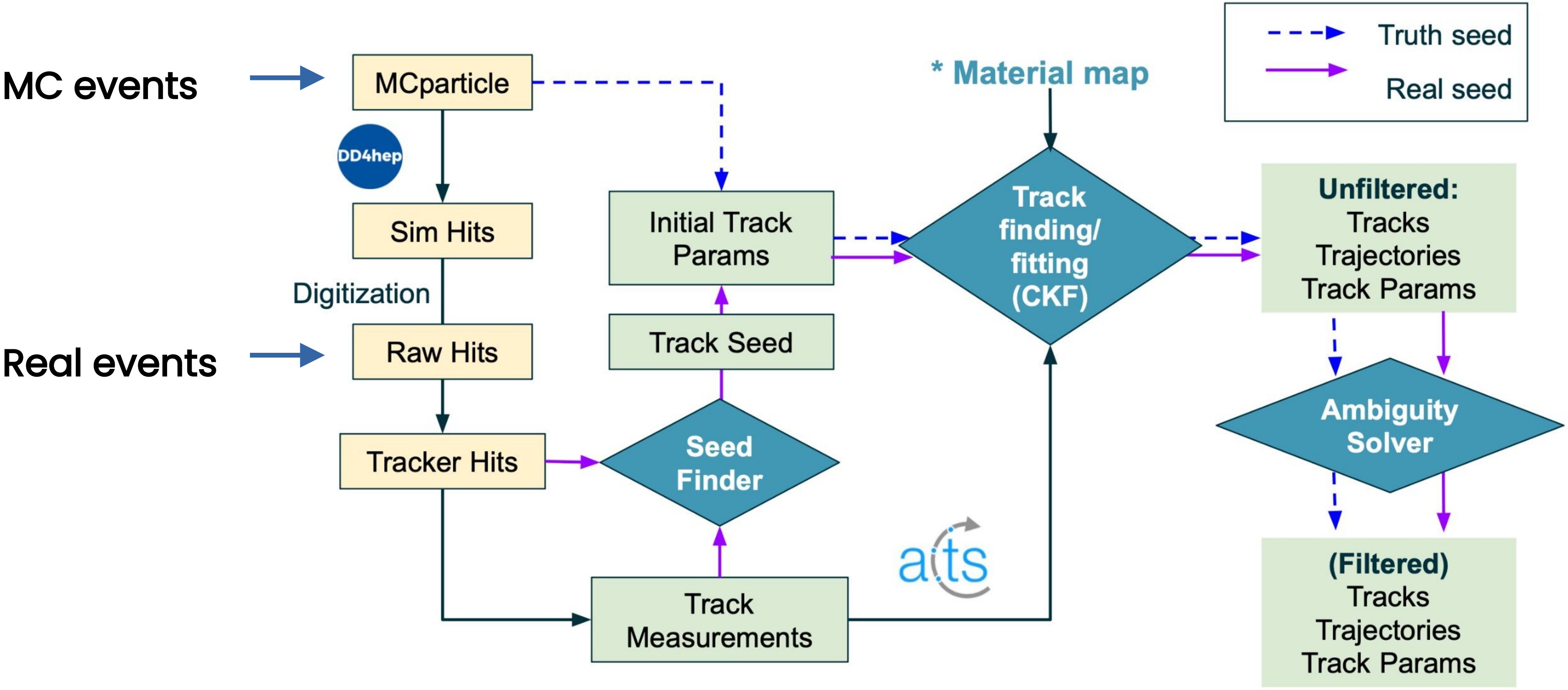


- **Strip-type for BTOF**
  - ~12 m^2
  - ~2.4 M channels
  - Strips: 0.05x10 mm2



- **Pixel-type for FTOF**
  - ~1.1 m^2
  - ~3.6 M channels
  - Pixel: 0.05x0.05mm2

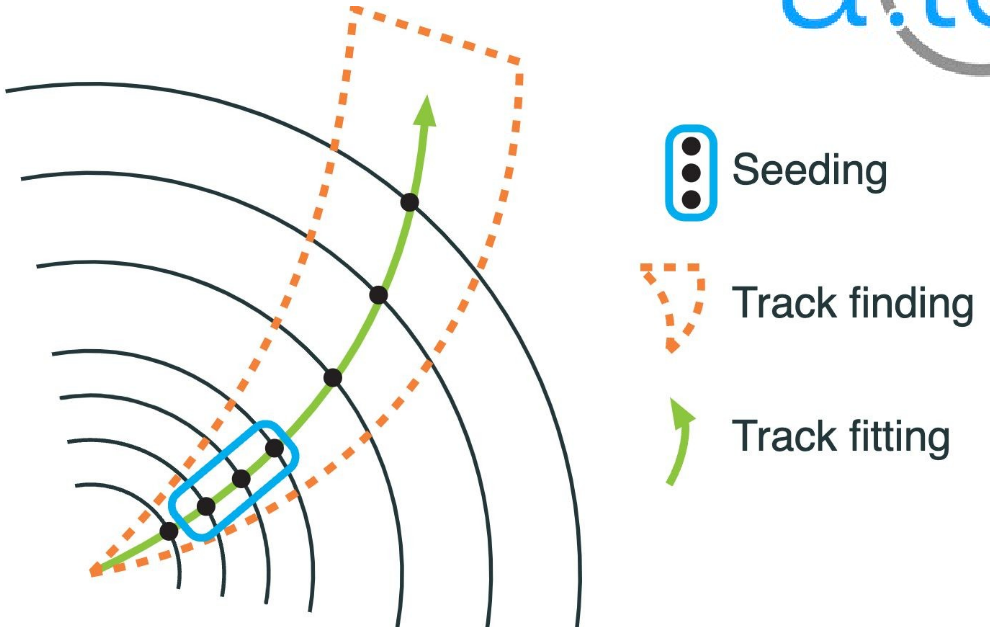
ePIC Tracking algorithm stack

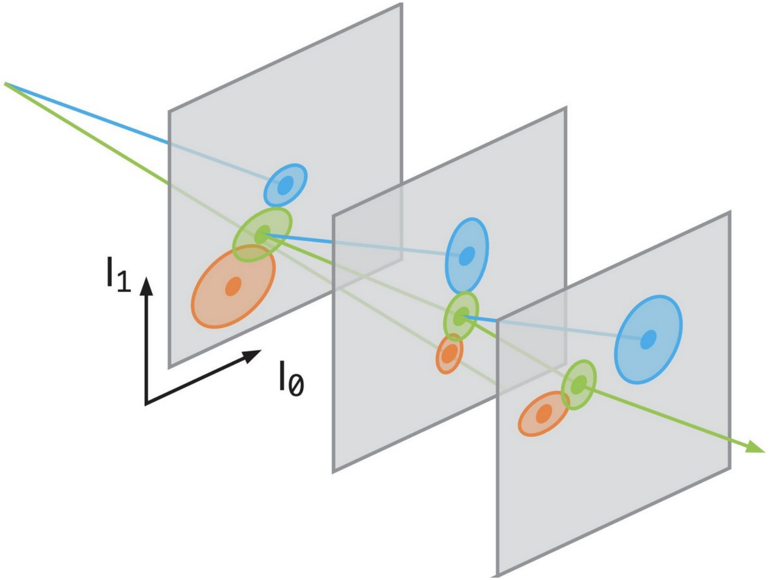


Track finding and fitting in ePIC

Combinatorial Kalman Filter (CKF) implemented in the software library ACTS (developed by ATLAS)

From an initial seed, all possible (acceptable) hits are considered, accept the best chi2, update the KF parameters for the next propagation

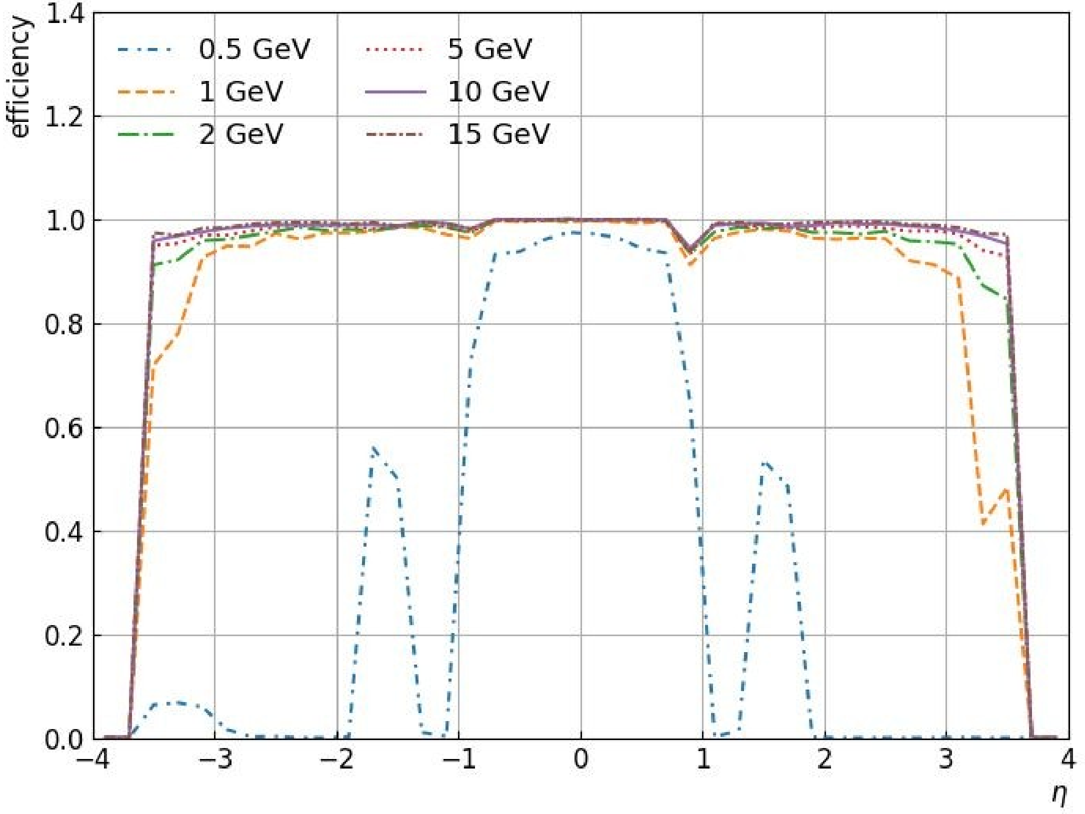




ePIC Tracking performance

The CKF in ePIC is fully efficient for pions with p>0.5 GeV/c, over the pseudorapidity range [-3.5,3.5]

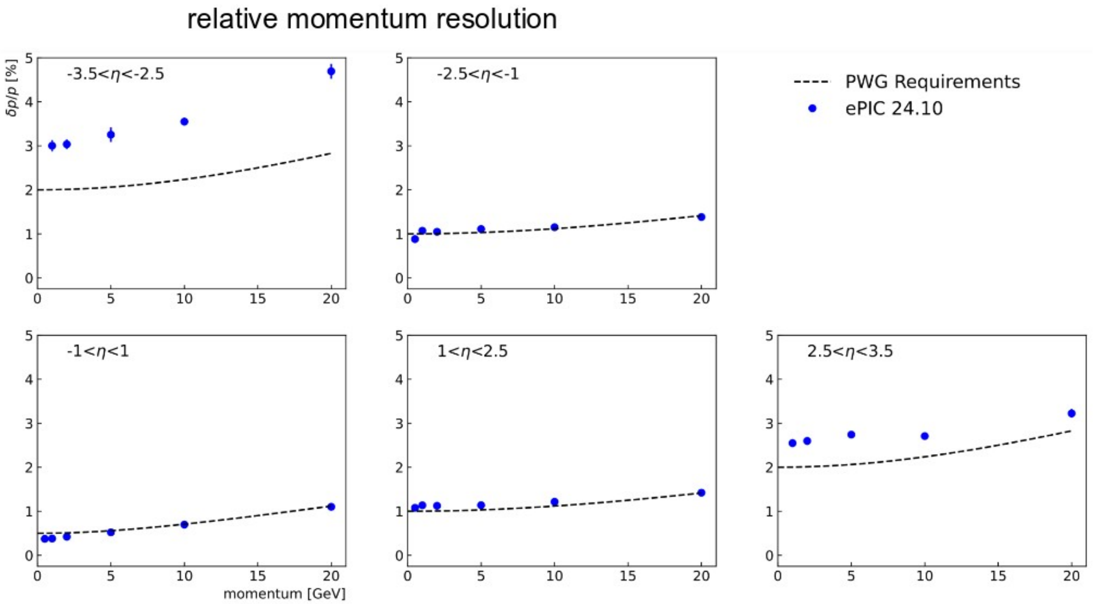
Some inefficiencies in the transition between the central and the backward/forward regions. Expected due to the detector geometry



ePIC Tracking performance

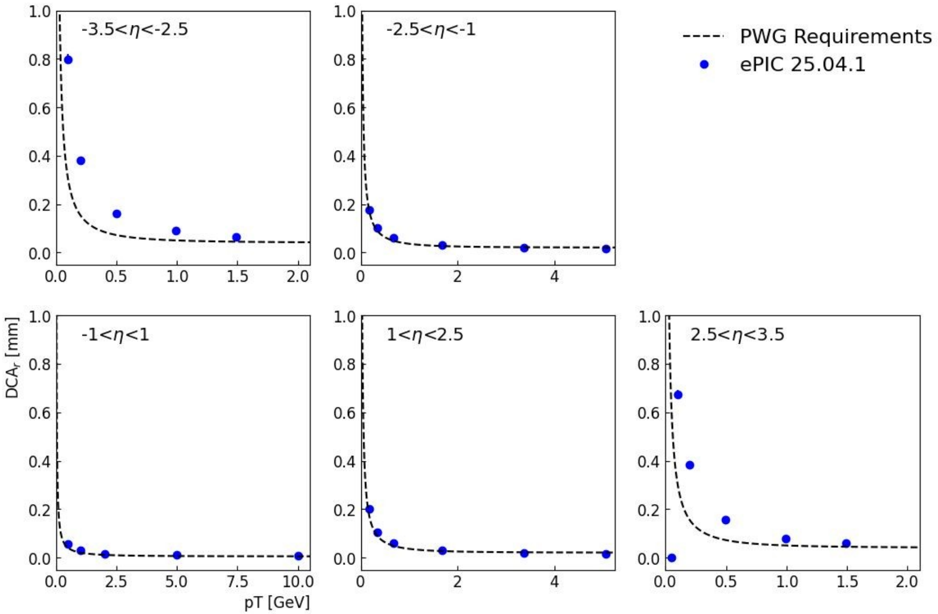
Single particle (pion) momentum resolution, shows good performance.

Most forward and backward,MS is the main limiting factor



The pointing resolution at the vertex is within requirements.

Simulations with DIS events plus background are ongoing



ePIC environment

	EIC	RHIC	LHC → HL-LHC
Collision species	$\vec{e}+\vec{p}$, $\vec{e}+A$	$\vec{p}+\vec{p}/A$, $A+A$	$p+p/A$, $A+A$
Top x-N C.M. energy	140 GeV	510 GeV	13 TeV
Bunch spacing	10 ns	100 ns	25 ns
Peak x-N luminosity	10^{34} cm^{-2} s^{-1}	10^{32} cm^{-2} s^{-1}	10^{34} → 10^{35} cm^{-2} s^{-1}
x-N cross section	**50 µb**	**40 mb**	**80 mb**
Top collision rate	500 kHz	10 MHz	1-6 GHz
$dN_{ch}/d\eta$ in p+p/e+p	0.1-Few	~3	~6
Charged particle rate	**4M N_{ch}/s**	**60M N_{ch}/s**	**30G+ N_{ch}/s**

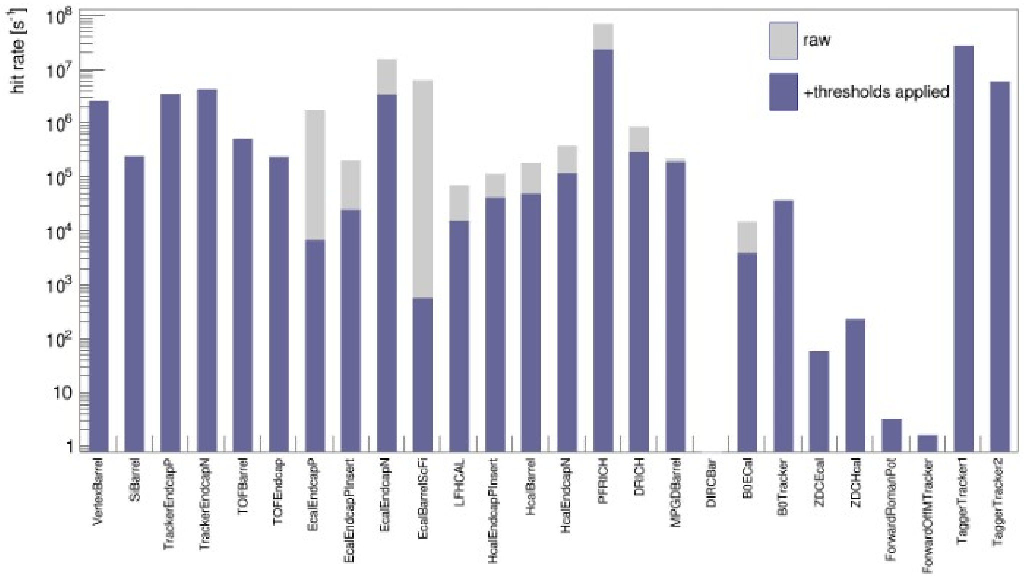
Collisions rate at EIC are small w.r.t. LHC, despite high luminosity

The number of particle per event is also small

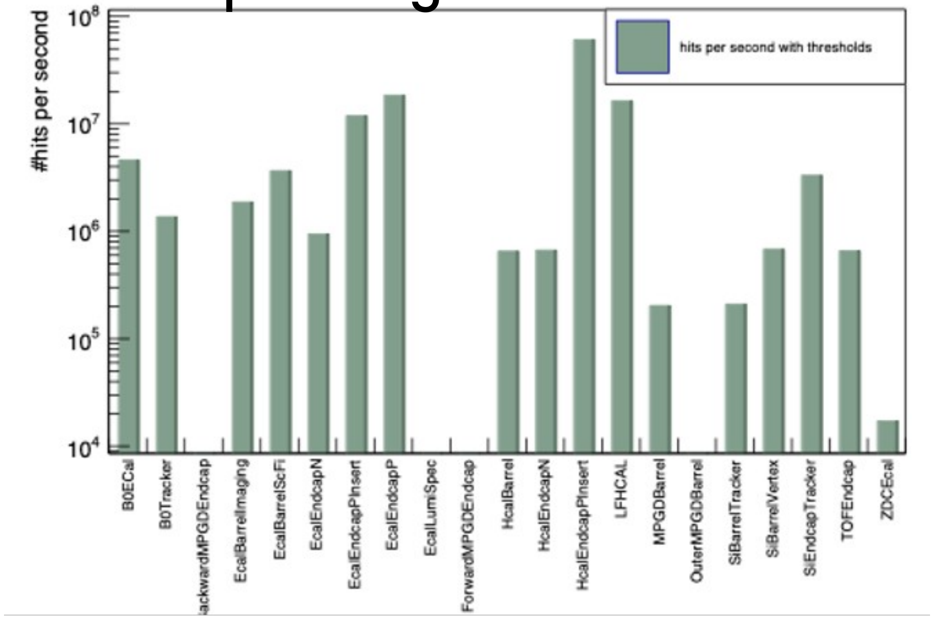
We can stream 100% of the data: no trigger

ePIC environment

electron-gas

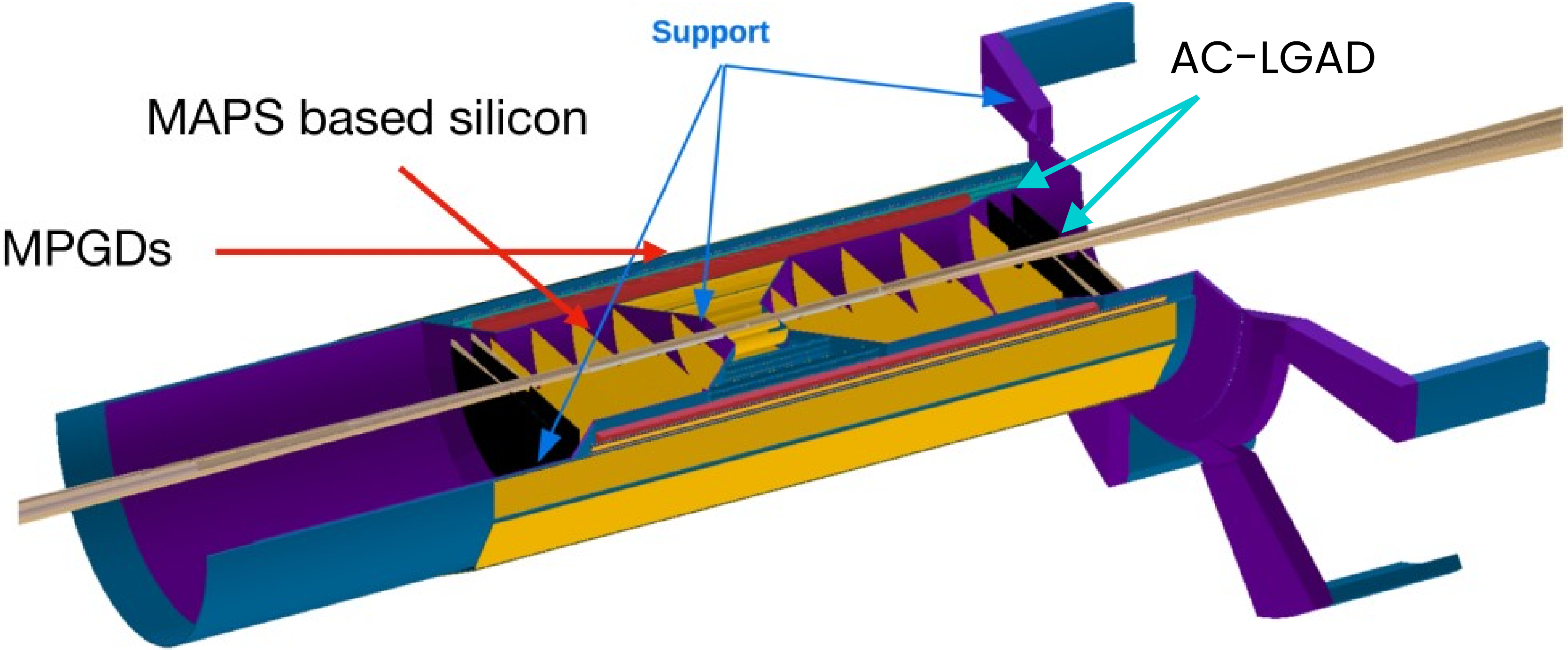


proton-gas



Backgrounds from beam gas events can add millions of hits per seconds in the ePIC detectors.
Synchrotron radiation is also a major concerns, active studies in the beam pipe design are ongoing to limit it

ePIC 4D tracking



SVT has great momentum reconstruction capabilities, but, with an integration time of 2-5μm, it can “see” O(500) bunch crossings
MPGDs (~10ns) and AC-LGAD (~30ps) will help associate tracks to the correct crossing

Summary

- Tracking in NHEP experiments is complex task that involves many aspects: detector technologies, electronics, engineering, software, …
- Two lectures are just for a very shallow overview
- EIC and ePIC are an important opportunity to make use of new technologies (not just for tracking), both in hardware and in software

Resources and Credits

- First place to go: PDG reviews https://pdg.lbl.gov/
- https://www.epic-eic.org/

I’ve stolen many plots and slides from many people

Credits: J. Osborn; J. Kaminski; C. Niebuhr; W. Hulsbergen; J. Baudot; F. Ragusa; M. Elsing; S. Lee; E. Sichtermann; S. Yano; K. Gnanvo; ...