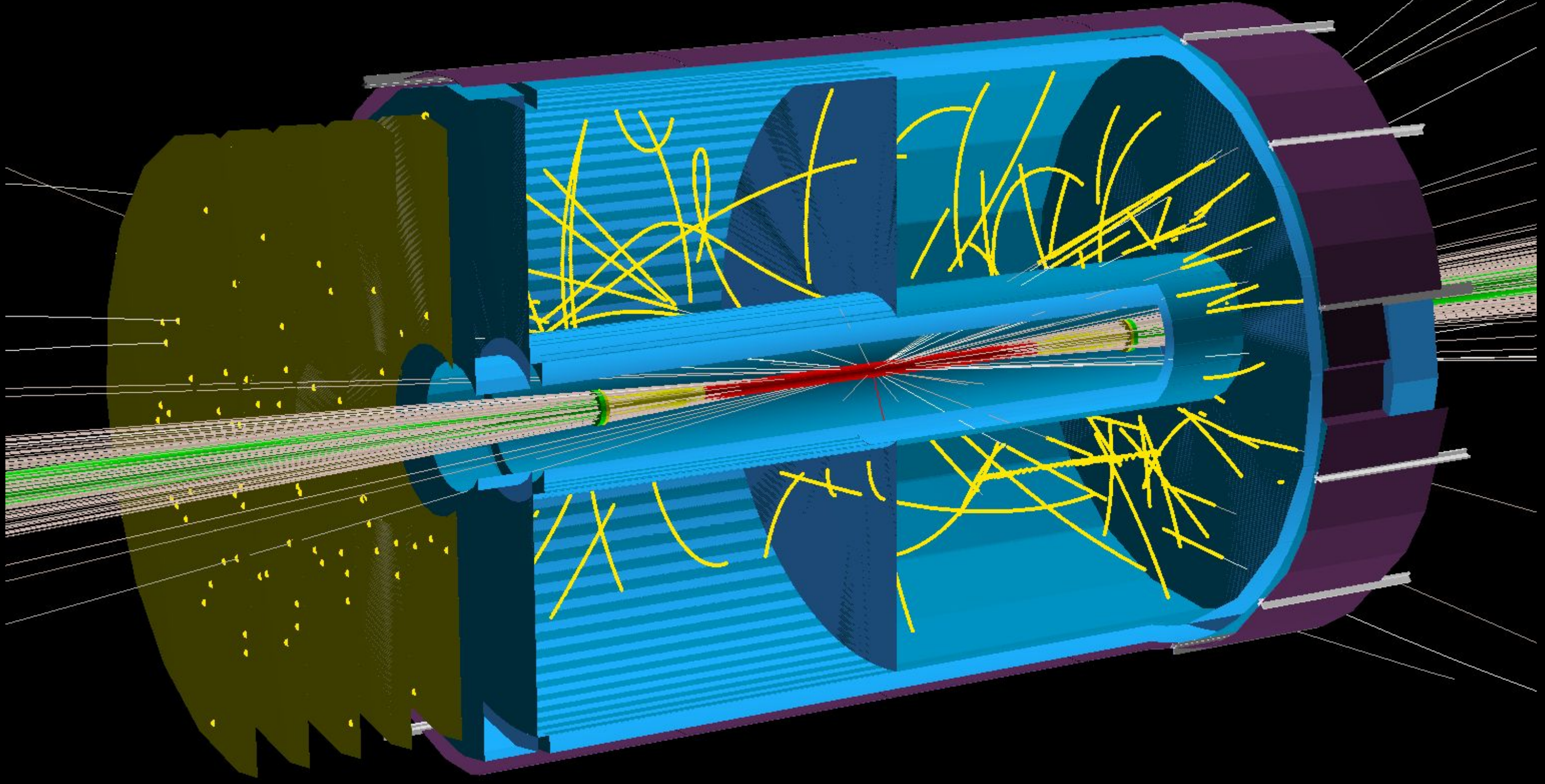
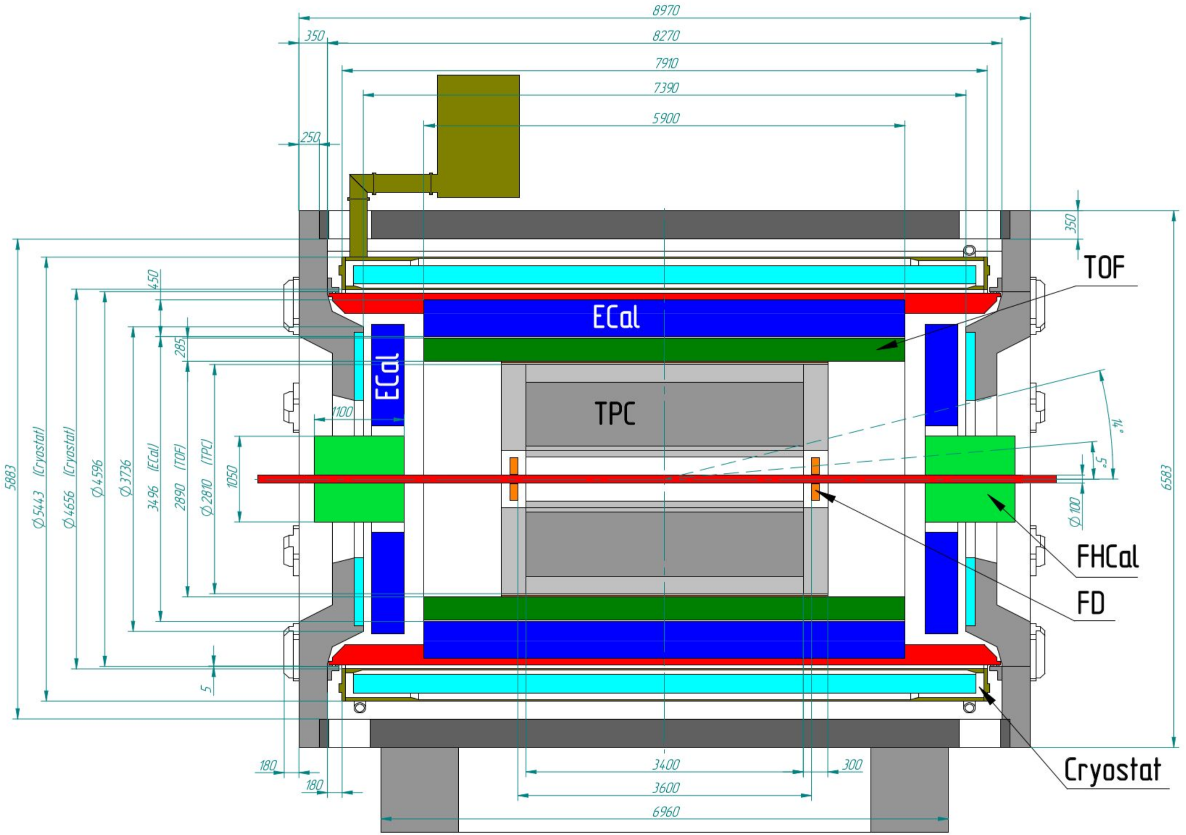
Implementation of ACTS-based track reconstruction for the forward detector in the MPD experiment at NICA



Evgeny Kryshen, Nazar Burmasov (PNPI, JINR)

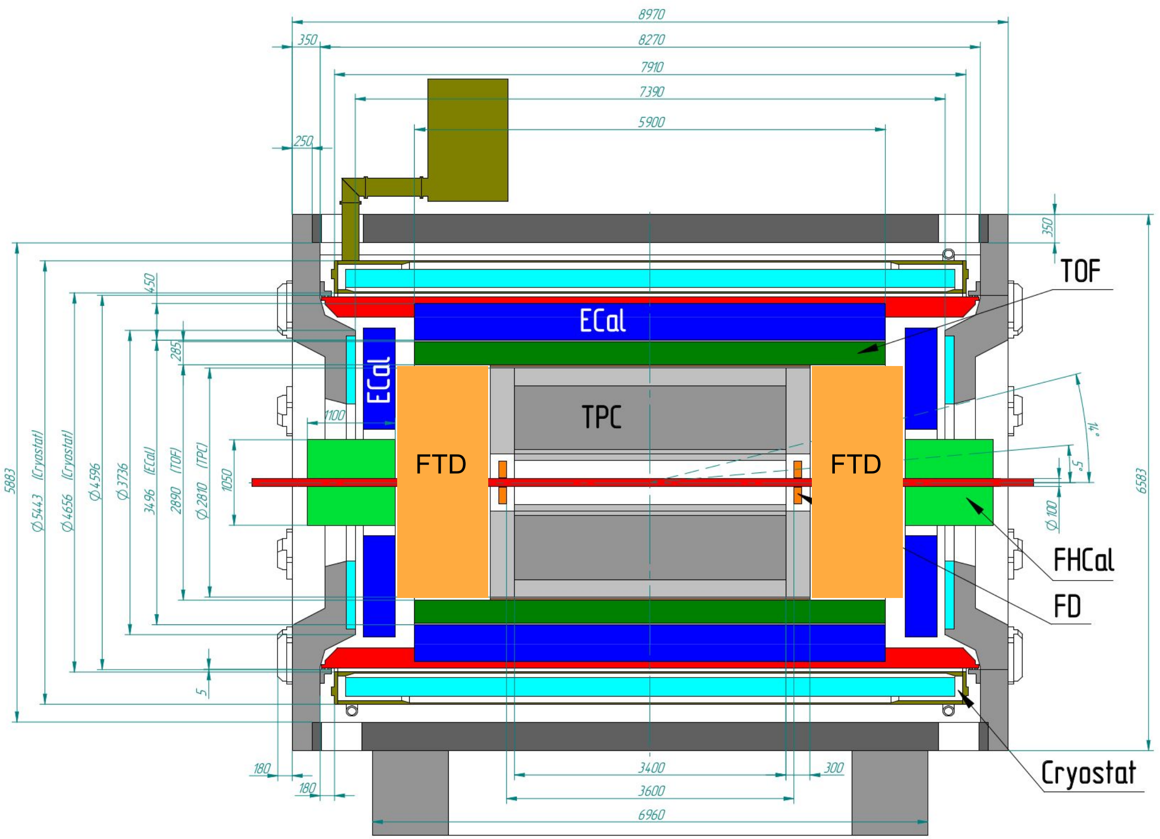
GRID, 8 July 2025

MPD: Stage I setup



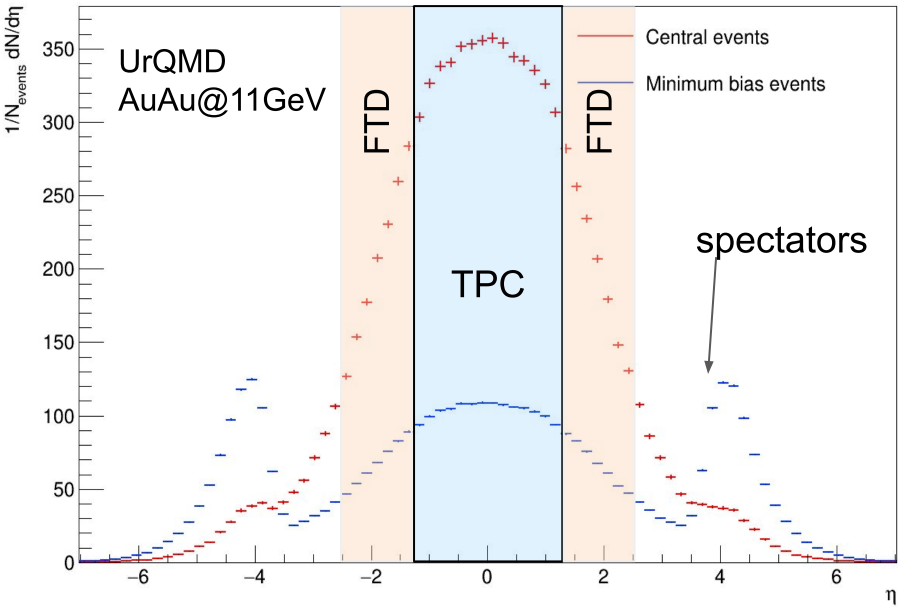
See talks on MPD by Arkady Taranenko (Tuesday) and Slavomir Hnatic (Thursday)

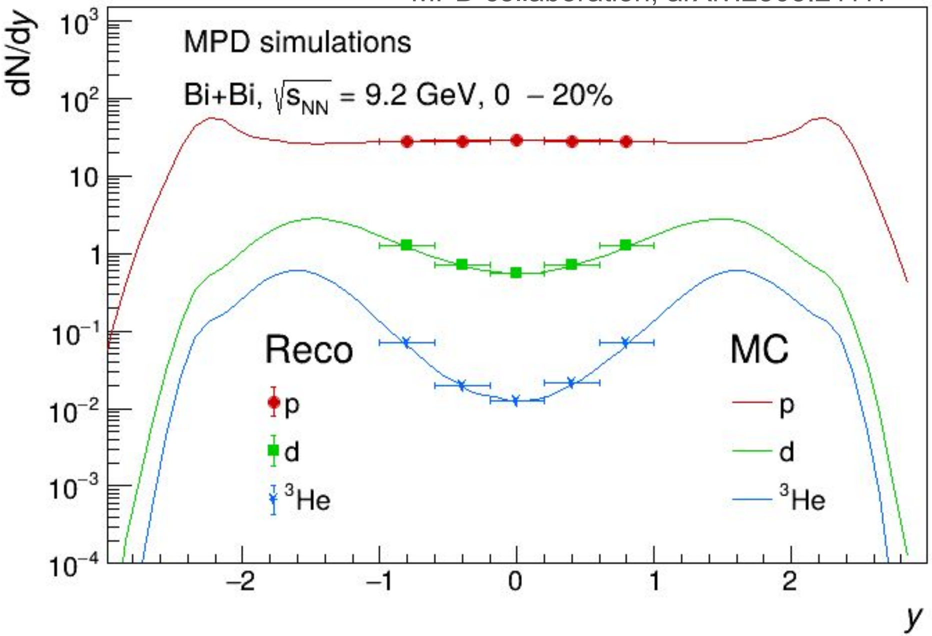
Extend rapidity coverage with forward tracker?



Physics motivation

MPD collaboration, arXiv:2503.21117





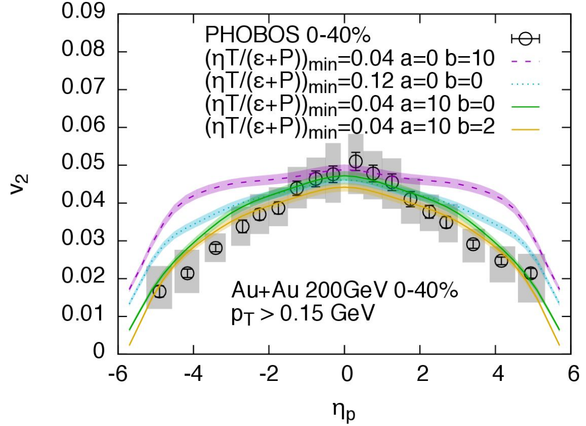
- TPC covers only ~55% of particle production yield in central events
➜ Forward tracker would allow us to cover more than 80%

- Production of light nuclei mainly at forward rapidities ➜ forward tracker would allow one to study the interplay of coalescence and baryon stopping mechanisms

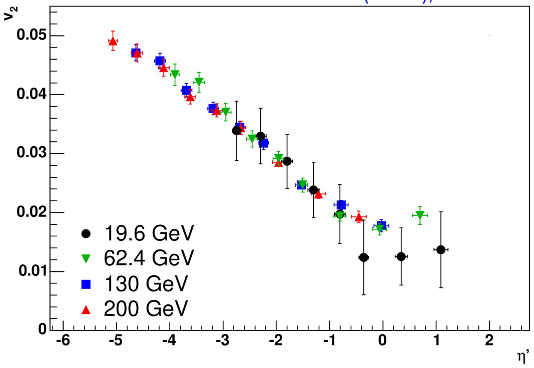
And more…

- the horn and the step effects at forward rapidities
- anisotropic flow: limiting fragmentation mechanism, temperature dependence of QGP etc.
- thermal photons via conversions on TPC endcaps
- global polarization of Λ hyperons: rapidity dependence?
- improve precision of centrality and reaction plane determination
- improved trigger efficiency for small systems
- possibility to access various observables of the SPD physics program
- aspects of non-perturbative QCD, e.g. diffractive studies, QCD instanton

PRL 116 (2016) 21, 212301



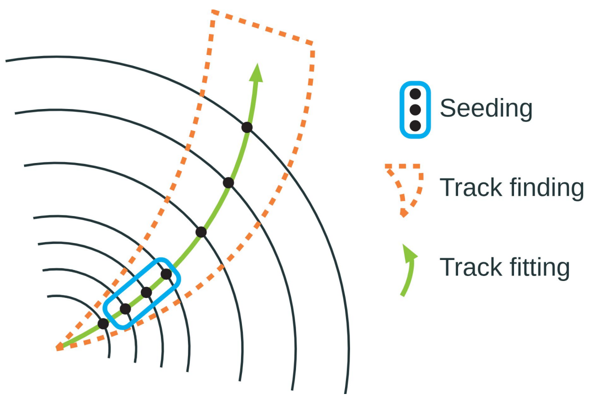
PRL 94 (2005), 122303



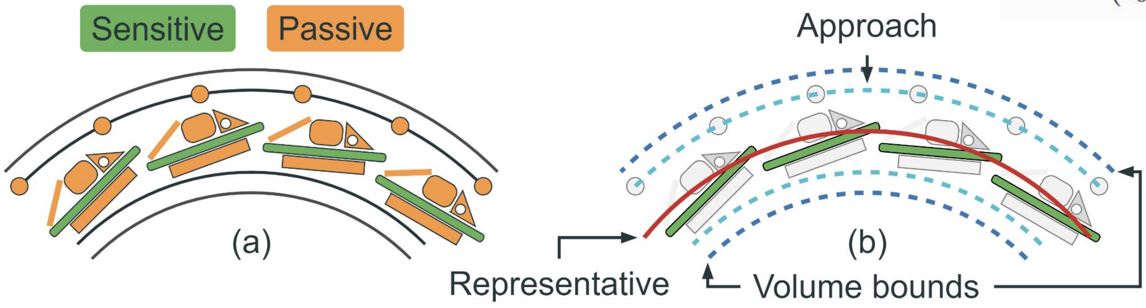
ACTS project

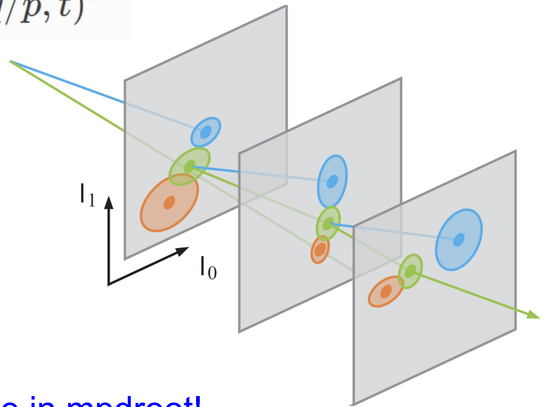
https://acts.readthedocs.io/

- A Common Tracking Software project
- Contains:
  - Box generator or interface to read external particles
  - Fatras (fast simulation tool) or interface to read hits
  - Digitization algorithm (smearing etc)
  - Seeding (several algorithms, including truth seeding)
  - Track finding/fitting with Combinatorial KF
- Accounting for energy losses, multiple scattering etc.
- Supporting multi-core execution, GPU etc.



$$\vec{x} = (l_0, l_1, \phi, \theta, q/p, t)^T$$





Using latest v41.1 from nicadist
Many thanks to Slavomir Hnatic and Jan Busa for integration of latest ACTS package in mpdroot!

Getting used to ACTS tracking algorithms…

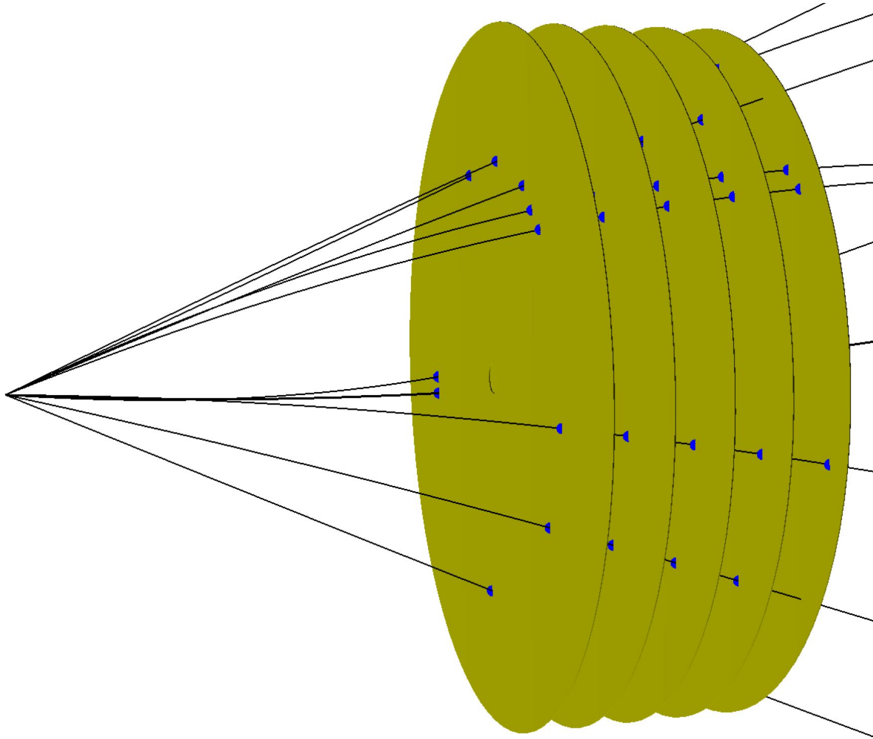
Getting used to ACTS tracking algorithms…

Considering “ideal” tracker:

- 5 tracking layers placed between 210 and 300 cm
- R_{inner} = 35.7 cm → η_{max} =2.47
- R_{outer} = 130 cm → η_{min} = 1.55
- Thickness per layer: 200 um silicon ~ 0.2% X_0
- Gaussian smearing in x and y with σ = 80 μm

Simulation config:

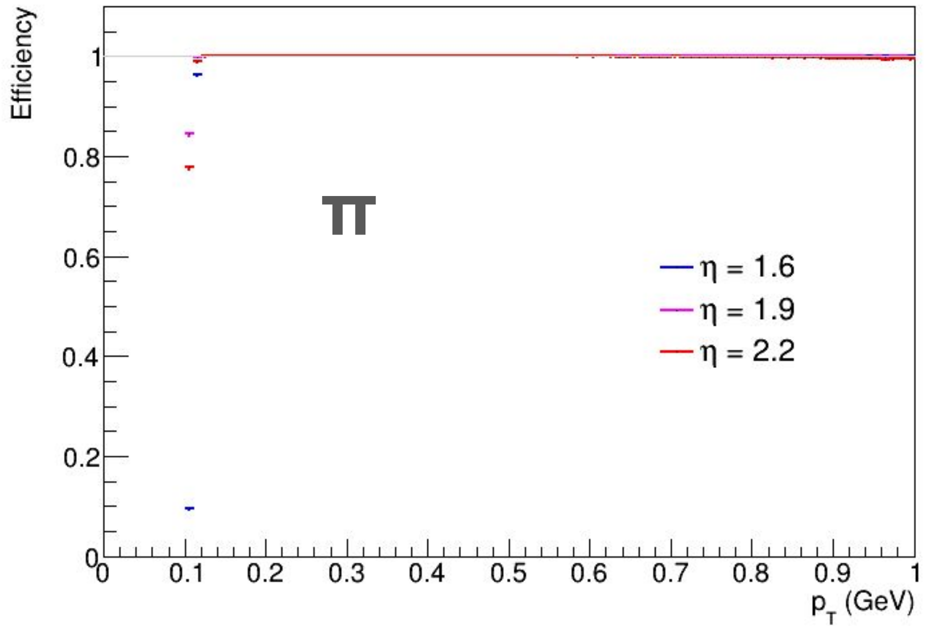
- Particle gun (π or p) with p_T from 0.1 to 1 GeV
- Build-in fatras transport (only EM processes)
- Seed finding using hits on first three layers (adopted seed finding algorithm for cylindrical layers)
- Track finding with combinatorial Kalman filter

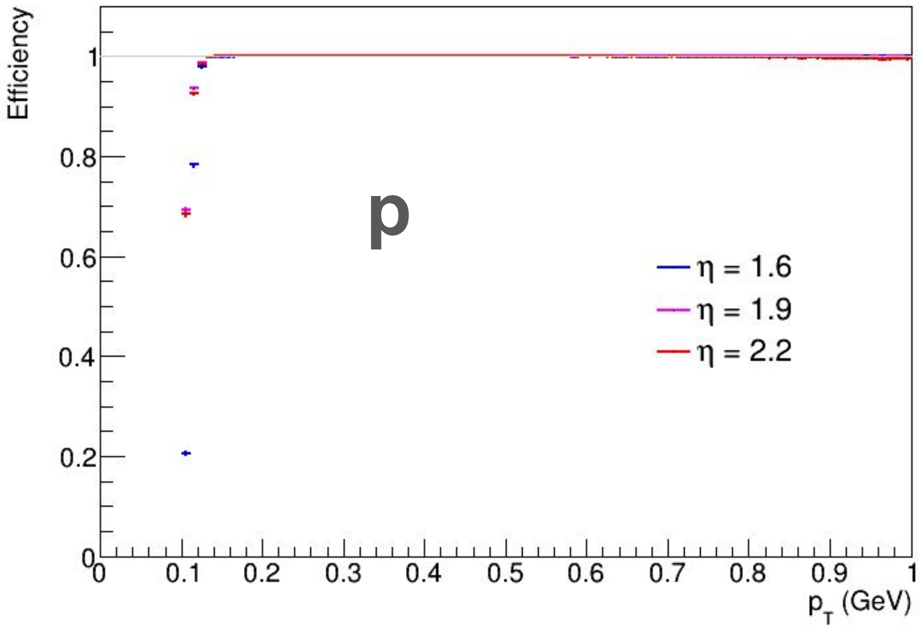


Study:

- seeding and tracking efficiency vs p_T and η
- p_T resolution vs p_T and η
- pulls (residuals normalized to estimated uncertainty)

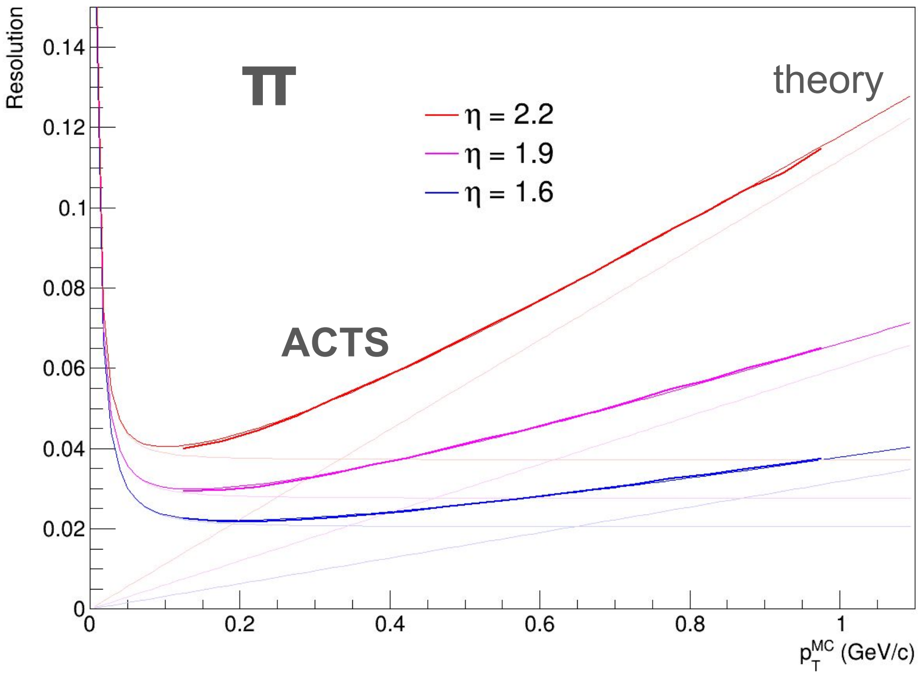
Tracking efficiency

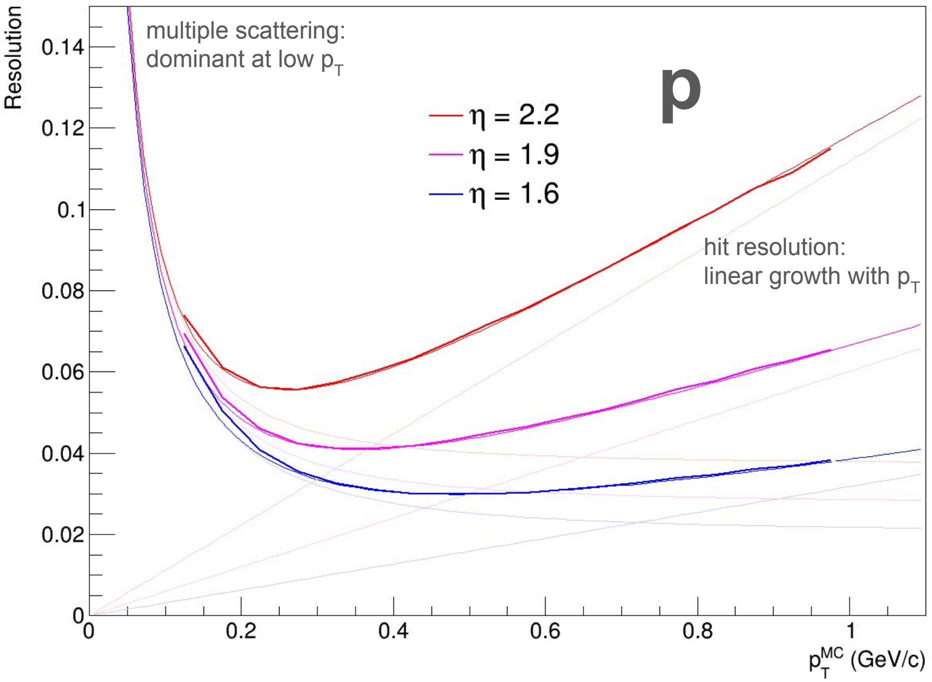




- Perfect efficiency for pions and protons in all eta regions
- Drop at 0.1 GeV due to limitation of the default seeding algorithm (curvature radius should be larger than $R_{max}/2$)

Momentum resolution





$$\left.\frac{\Delta p_{\mathrm{T}}}{p_{\mathrm{T}}}\right|_{\mathrm{m.s.}} = \frac{N}{(N+1)(N-1)} \frac{0.0136\,\mathrm{GeV}}{0.3\beta B L} \sqrt{\frac{d_{\mathrm{tot}}}{X_0 \cos\theta}} \left(1 + 0.038 \ln \frac{d}{X_0 \cos\theta}\right)$$

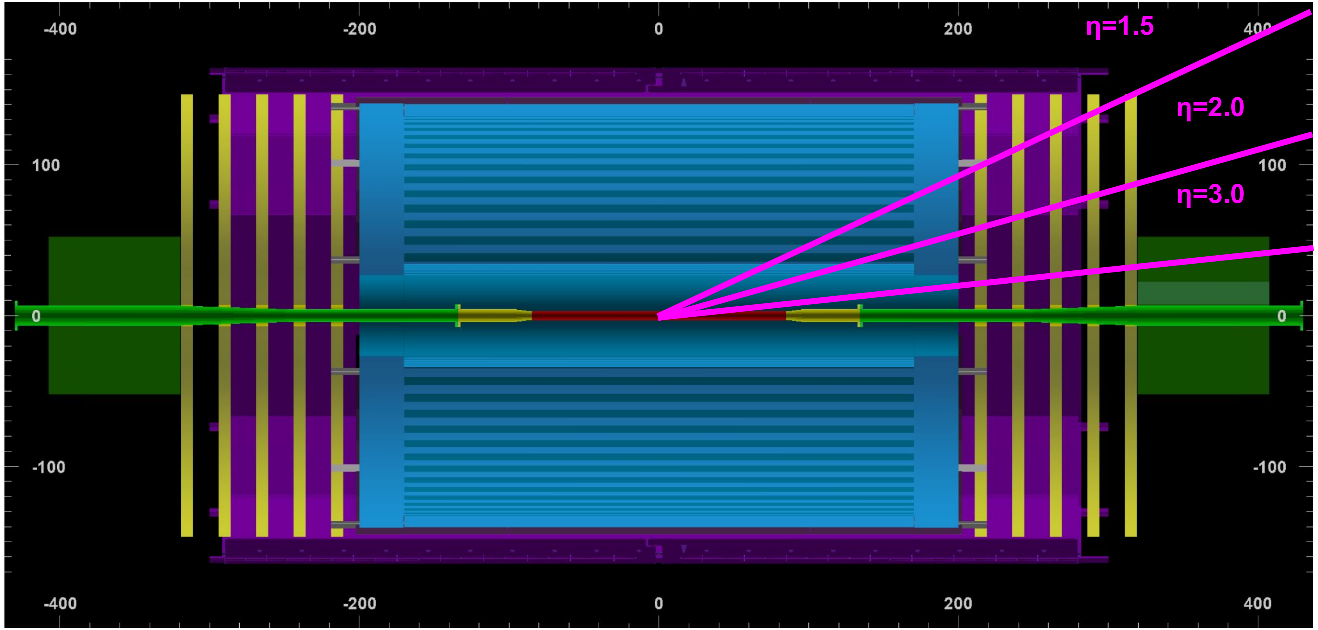
$$\left.\frac{\Delta p_{\mathrm{T}}}{p_{\mathrm{T}}}\right|_{\mathrm{res.}} = \frac{\sigma_{\mathrm{hit}}\, p_{\mathrm{T}}}{0.3 B L^2} \sqrt{\frac{720 N^3}{(N-1)(N+1)(N+2)(N+3)}}$$

- momentum resolution is within 3-11% depending on p_T and pseudorapidity
- strong effect of multiple scattering for protons at low p_T
- perfect agreement of ACTS fits with theory

formulae from Drasal and Riegler, NIM A910 (2018) 127, adopted to the forward tracker case

Towards more realistic tracking

FTD simulations in mpdroot



Basic FTD geometry and hit producer embedded in mpdroot

- 5 tracking layers placed between 210 and 300 cm
- Thickness per layer: 0.2% X_0
- Gaussian smearing in x and y with σ = 100 μm

Geometry hierarchy in ACTS: volumes

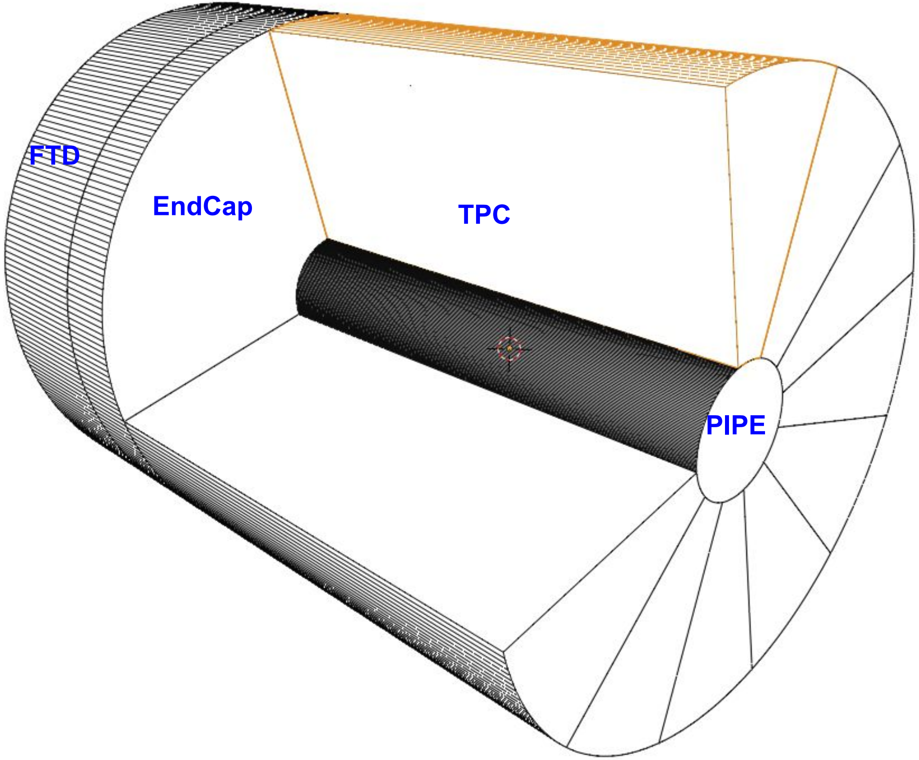
Volumes:

z binning

- BARREL
  - PIPE
  - TPC:
    - TPC0
    - TPC1
    - …
    - TPC11
- EndCap
- FTD

r binning (PIPE, TPC)

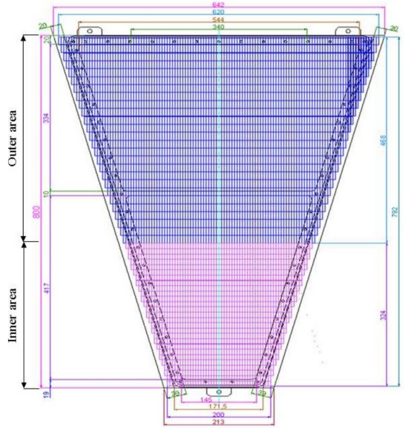
phi binning (TPC0 … TPC11)

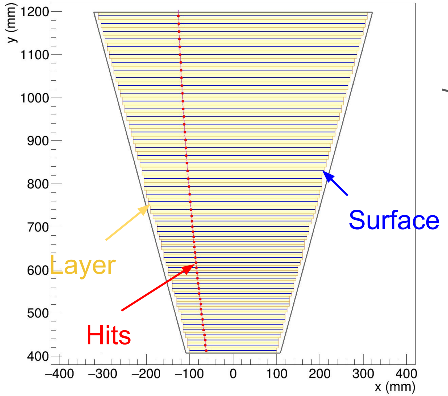


Fully connected geometry:

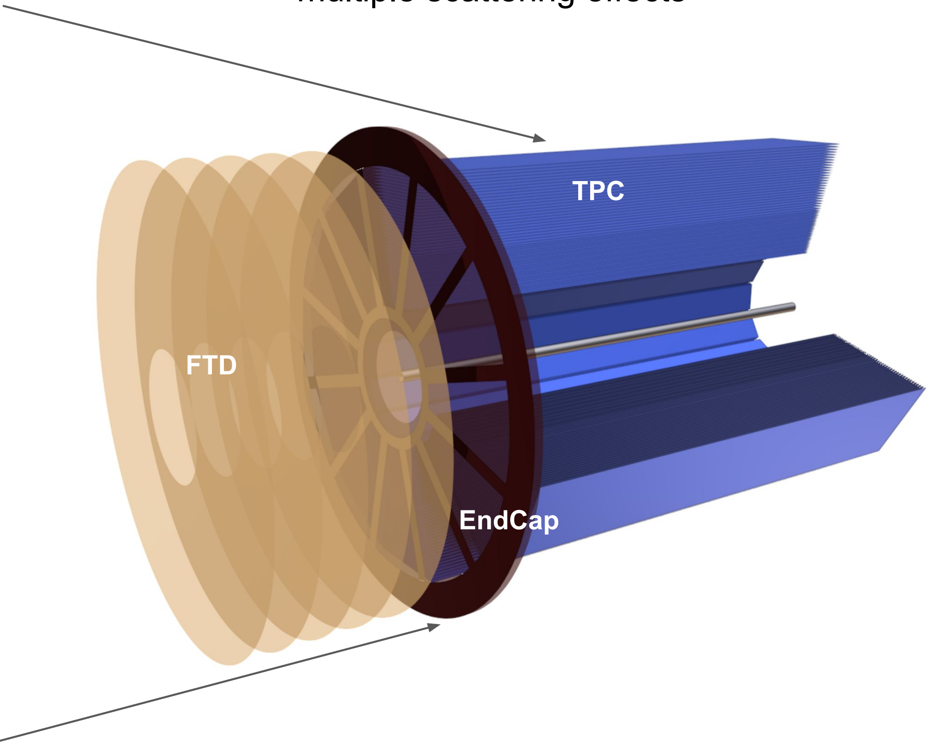
- Common boundary surfaces are glued (e.g. FTD and EndCap)
- If boundary is shared by several volumes, volumes must be attached to boundary (e.g. TPC0… TPC11 to pipe boundary)

Surfaces in TPC, FTD, Pipe and EndCap volumes

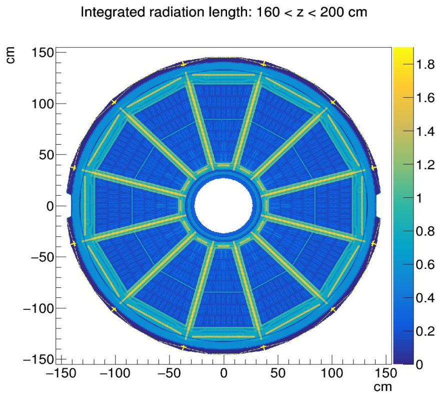


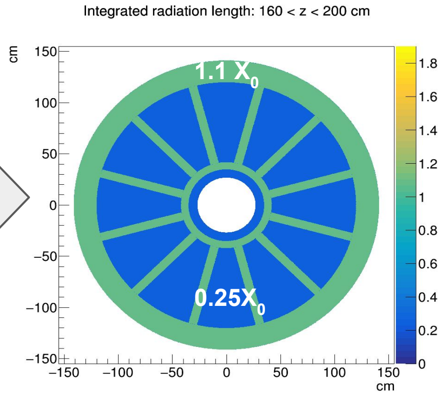


- Surfaces are used to account for multiple scattering effects



- Measurements (TPC hits) must be attached to surfaces

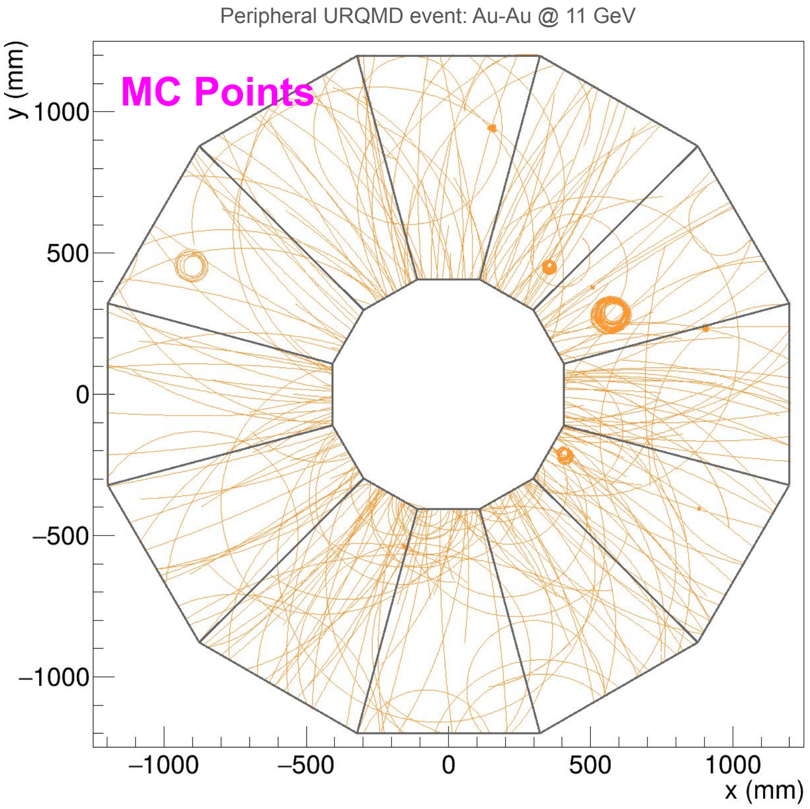


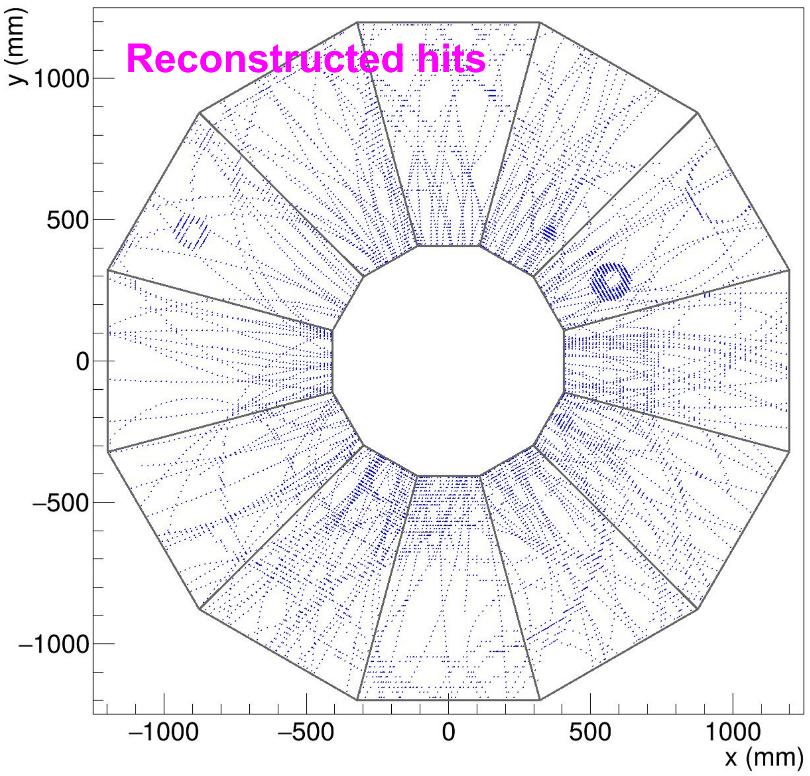


- Endcap: toy model with ROC-like and Frame-like layers

TPC tracking

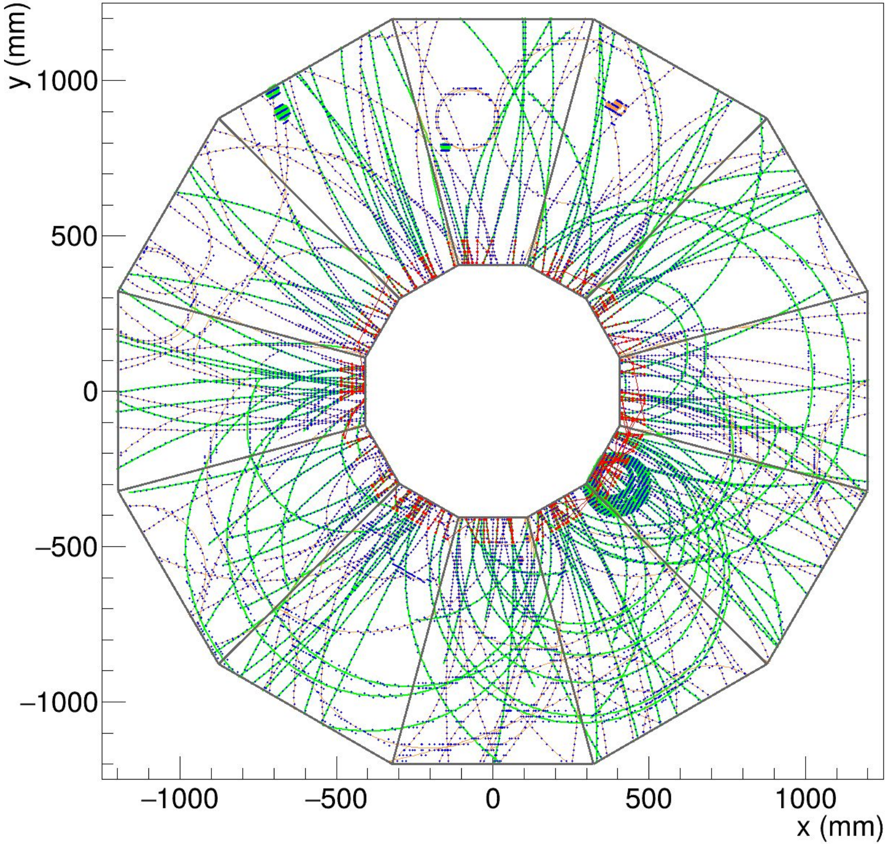
Typical MC and reconstructed hit distributions in TPC

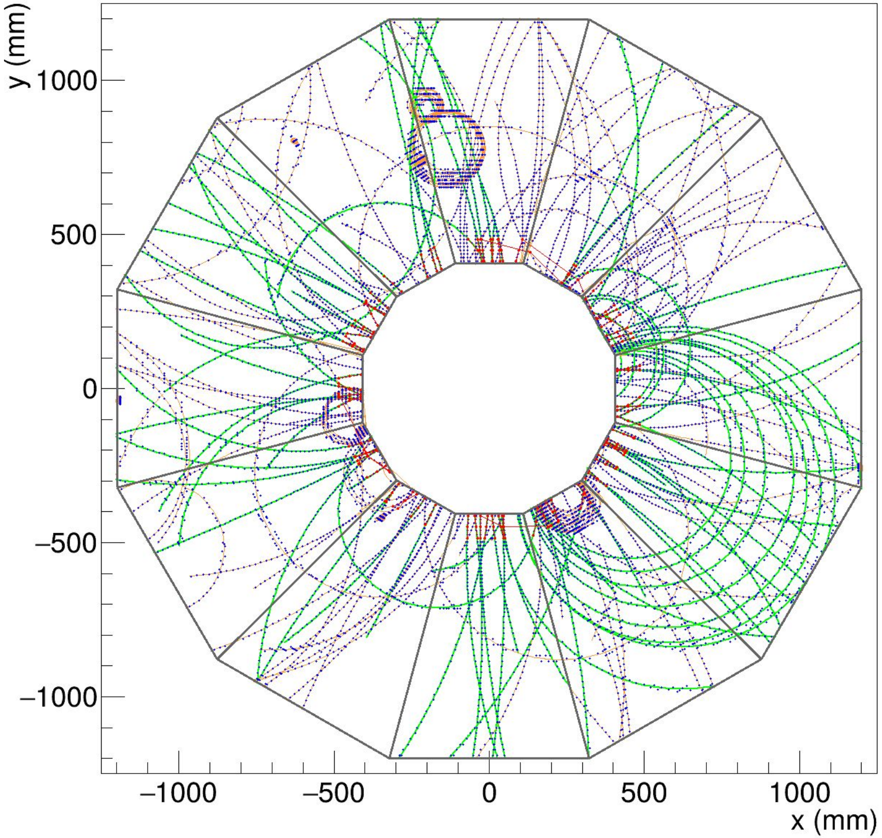




- McTracks, McPoints, and TPC hits converted to ACTS format
- Using realistic hits from MLEM clustering algorithm

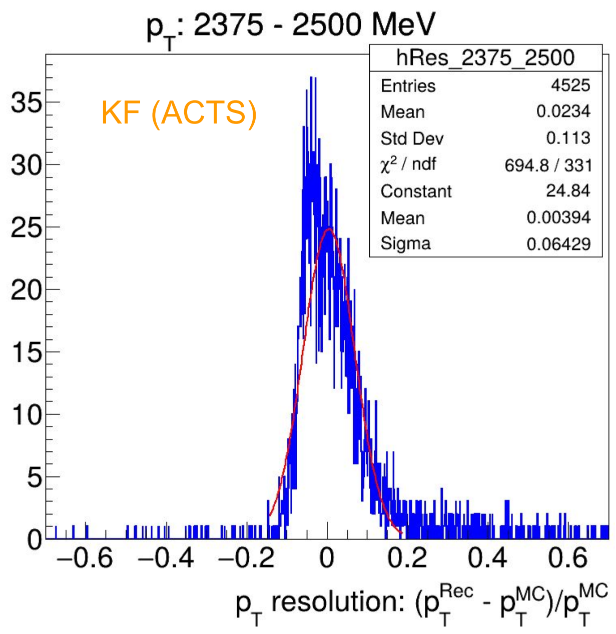
Examples from UrQMD generator (AuAu @ 11 GeV)



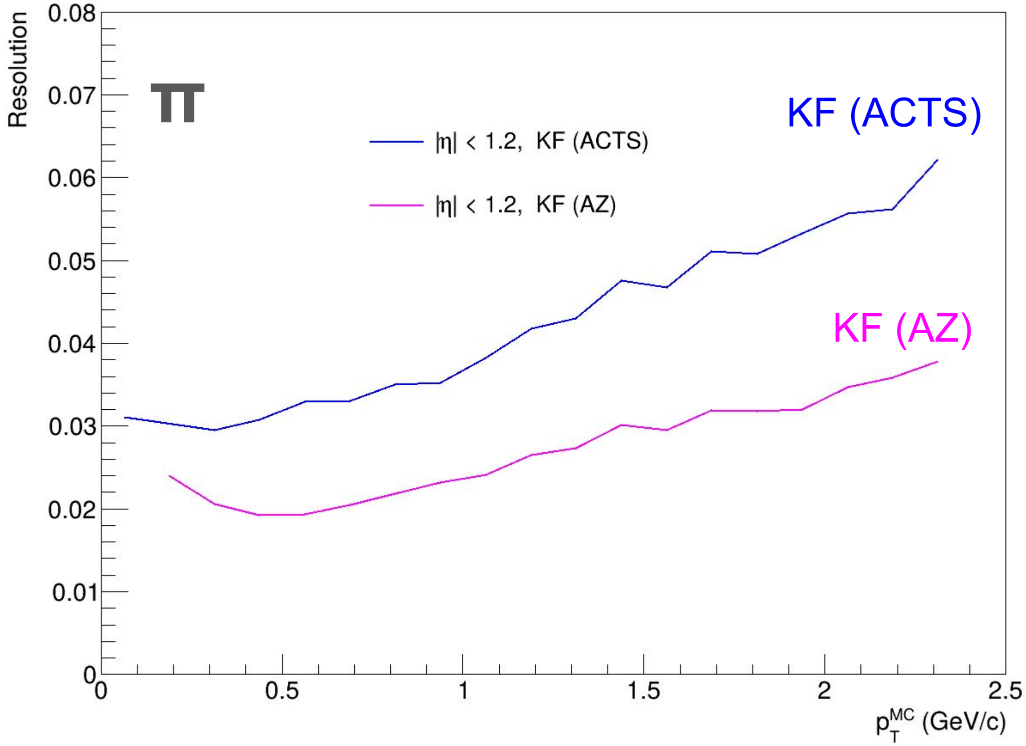


red: seeds green: reconstructed tracks

Momentum resolution



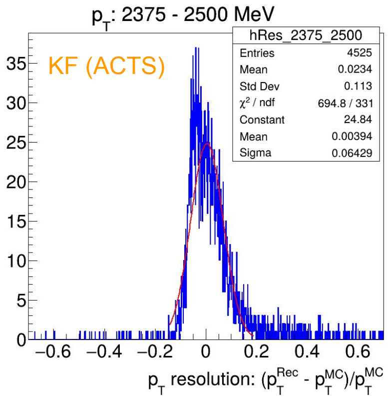
Significant tails and biases in p_T residual plots with the KF from ACTS

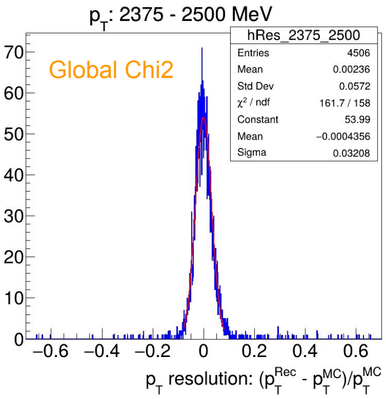


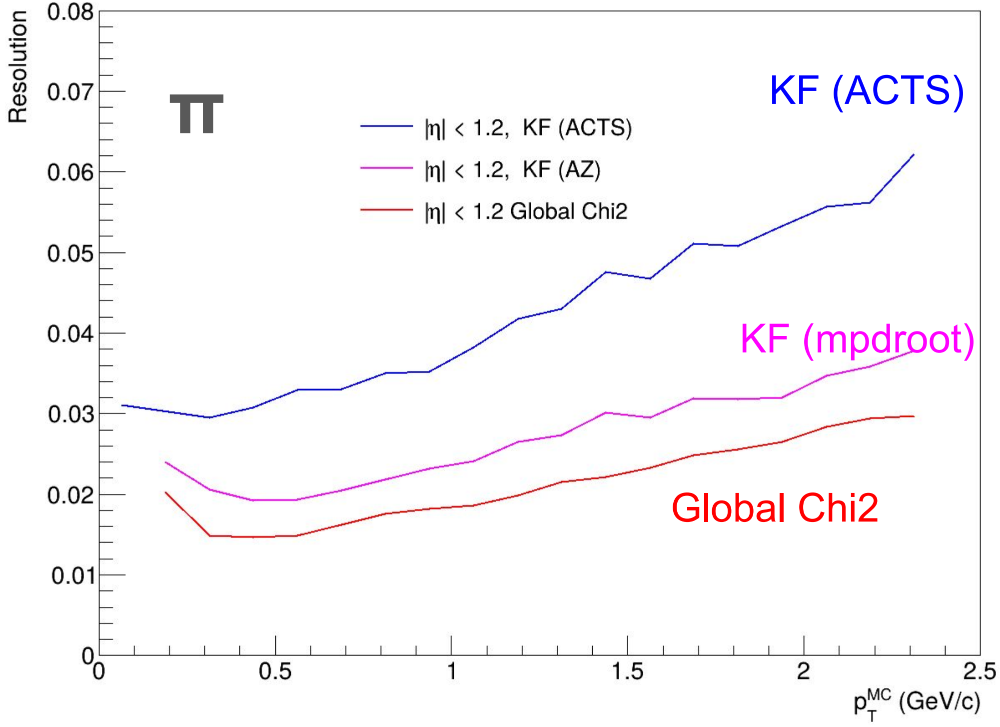
Momentum resolution with KF from ACTS significantly worse compared to KF implementation in mpdroot (by A. Zinchenko)

Refitting with Global Chi2 fitter

- Custom refitting algorithm developed to explore different fitting options (KF, Global Chi2 etc)
- Much better residuals with Global Chi2
- p_T resolution from Global Chi2 fitter appears to be much better compared to KF from ACTS and also slightly better compared tom KF from mpdroot



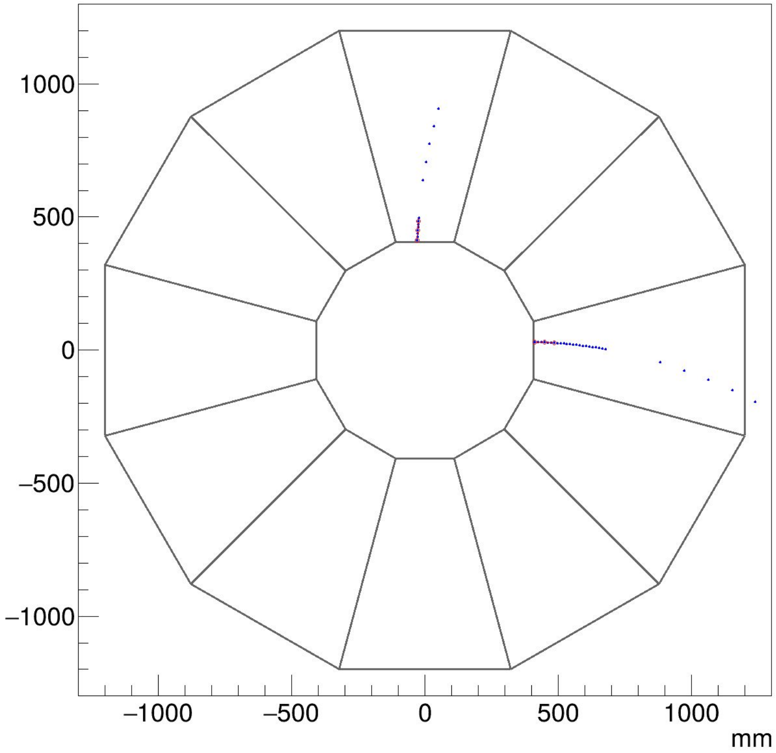


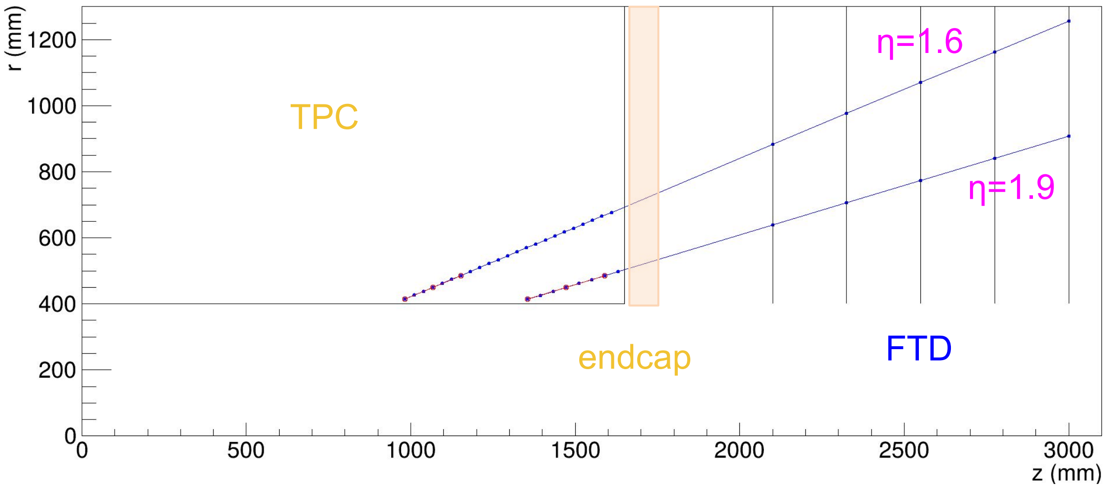


- The refitting algorithm can also be used to refit reconstructed tracks with different mass hypotheses

Towards TPC+FTD tracking

FTD tracking with TPC seeds





- FTD hits converted to ACTS format
- Use TPC seeds (1,4,7 padrows)
- Apply combinatorial KF to attach TPC and FTD measurements to tracks

FTD tracking performance: efficiency (Boxgen)

Seeding:

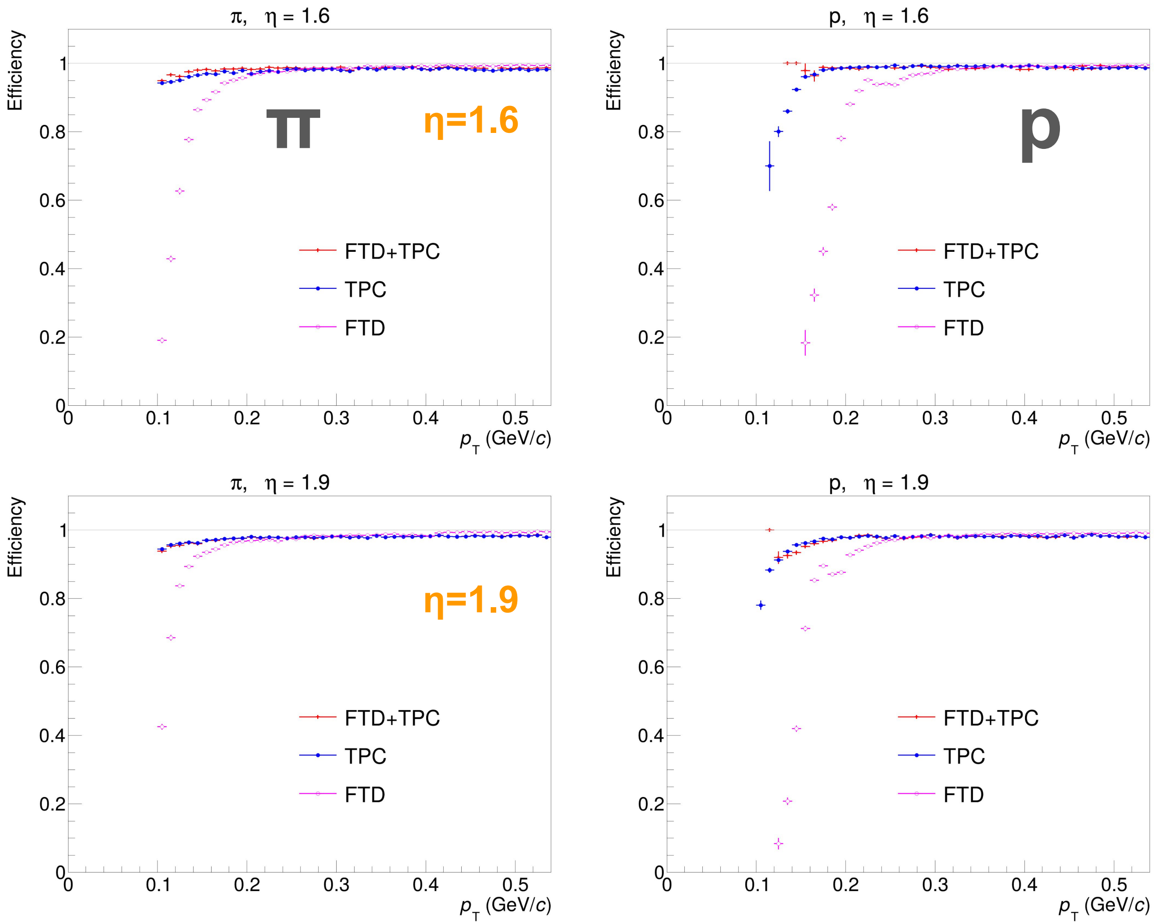
- TPC: for TPC and FTD+TPC
- FTD: for FTD

Shown trackable efficiencies.
Minimum requirements:

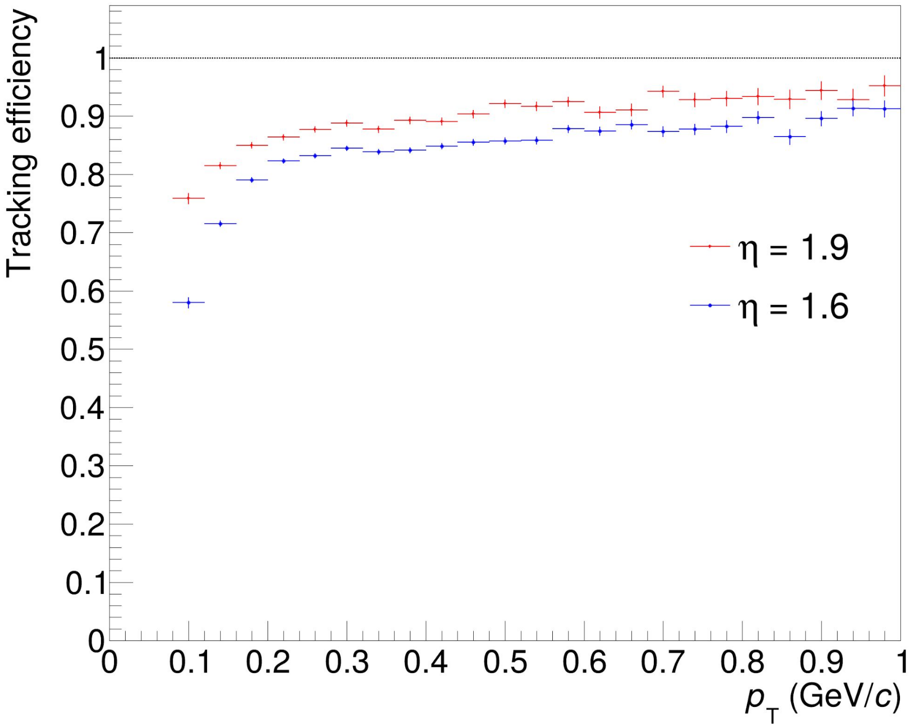
- TPC: hits in 1,4,7 padrows
- TPC+FTD: hits in 1,4,7 padrows, at least 3 hits in FTD
- FTD: 5 hits in FTD
- ALL: stay away from end-cap frame (~110% X_0)

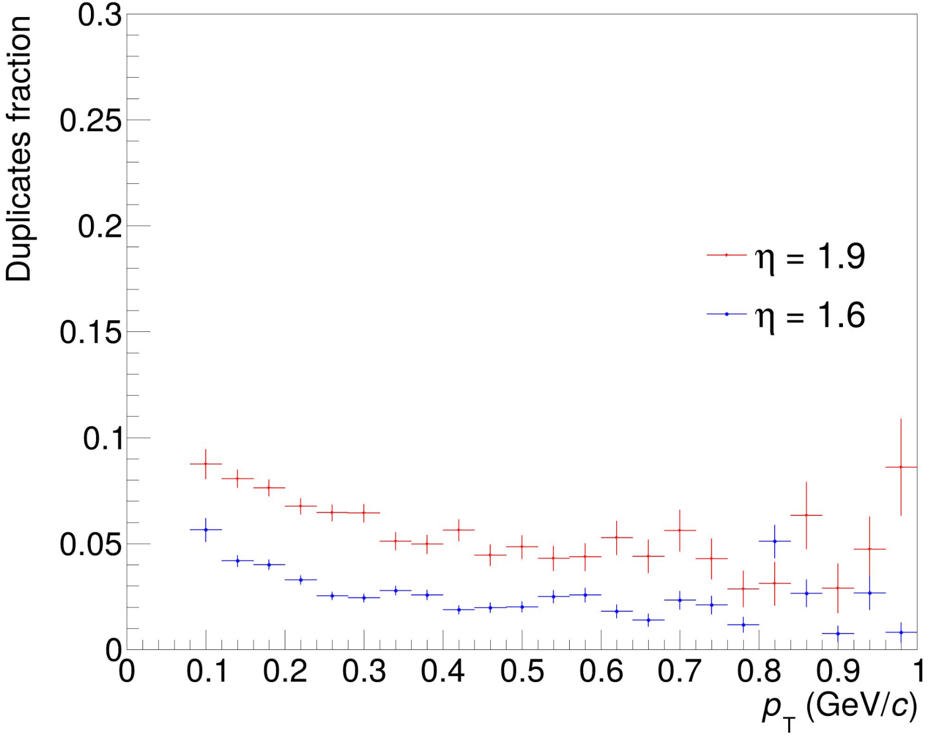
Results:

- Close to perfect efficiencies for single-track boxgen



TPC+FTD tracking performance (UrQMD)



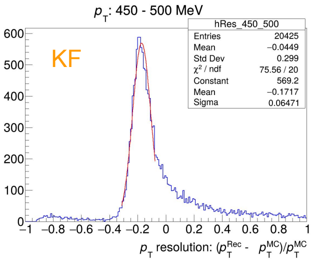


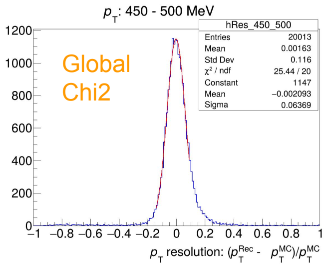
- Efficiency with standard 1-4-7 seeding

FTD(+TPC) tracking resolution

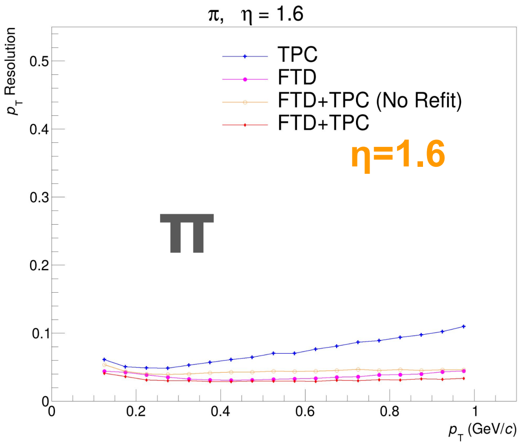
- KF: Biased momentum estimate with long tails
- Global Chi2: Much better Gaussian-like distributions
- FTD significantly improves momentum resolution, especially at large eta
- Combined FTD+TPC fit further improves momentum resolution
- TPC-FTD-matching helps to improve DCA resolution

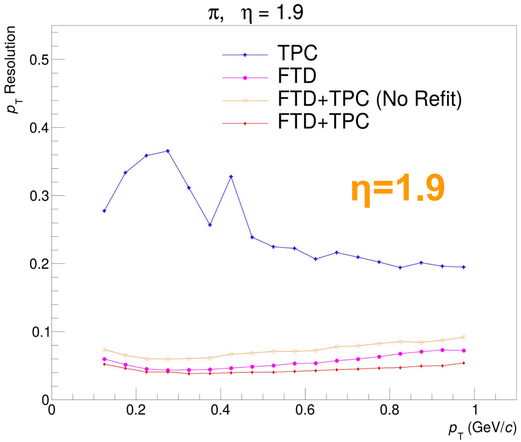
Resolution from TPC-only fit for pions at eta = 1.6



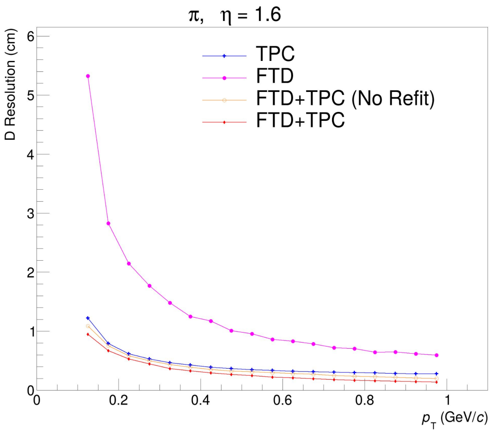


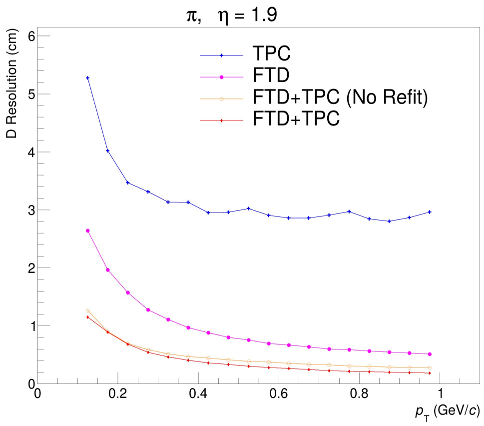
Momentum resolution





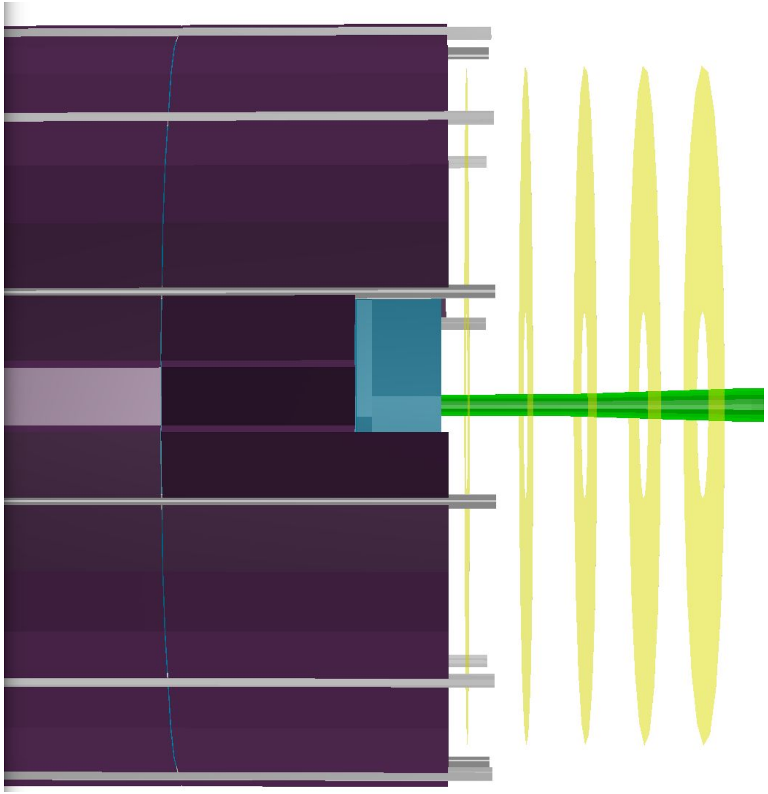
DCA resolution

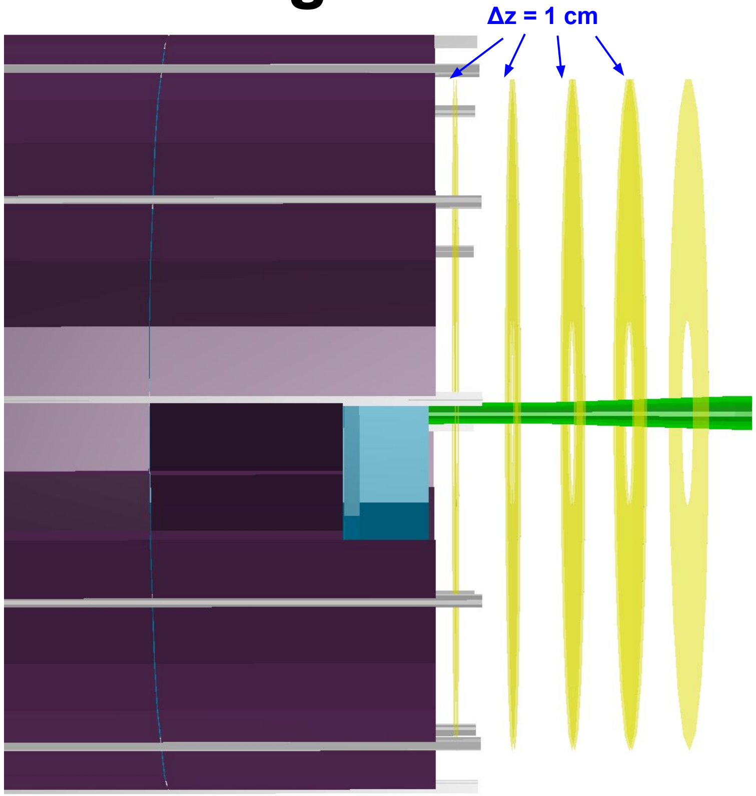




ACTS tracking
in strip-like forward detector

2D tracking vs 1D tracking

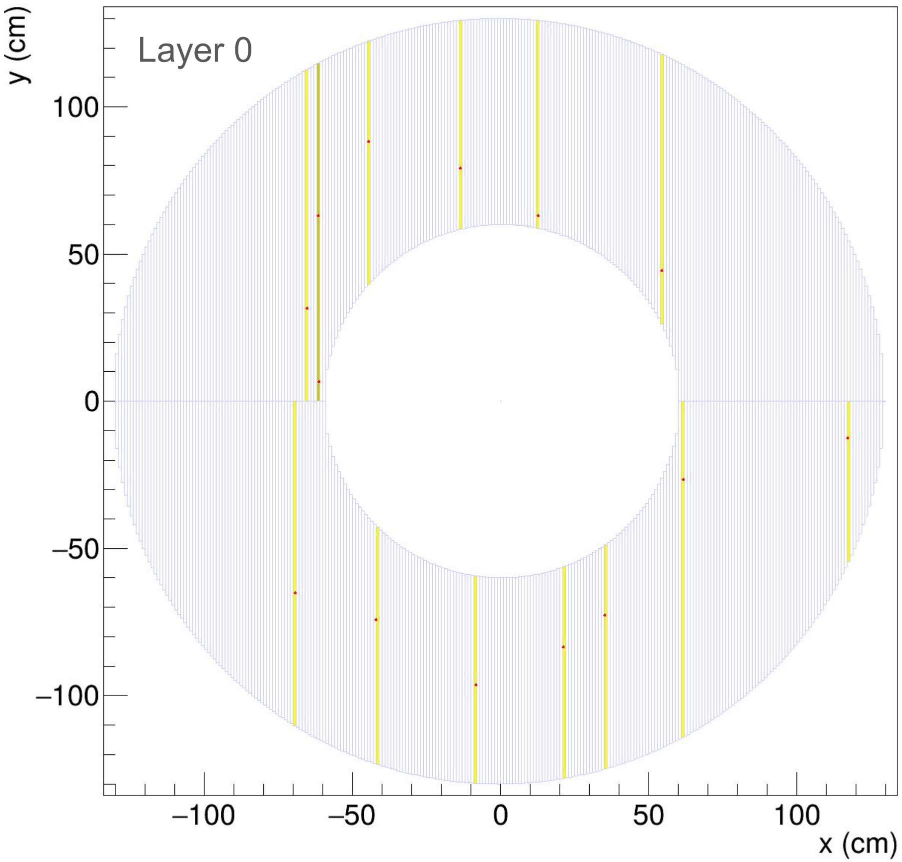


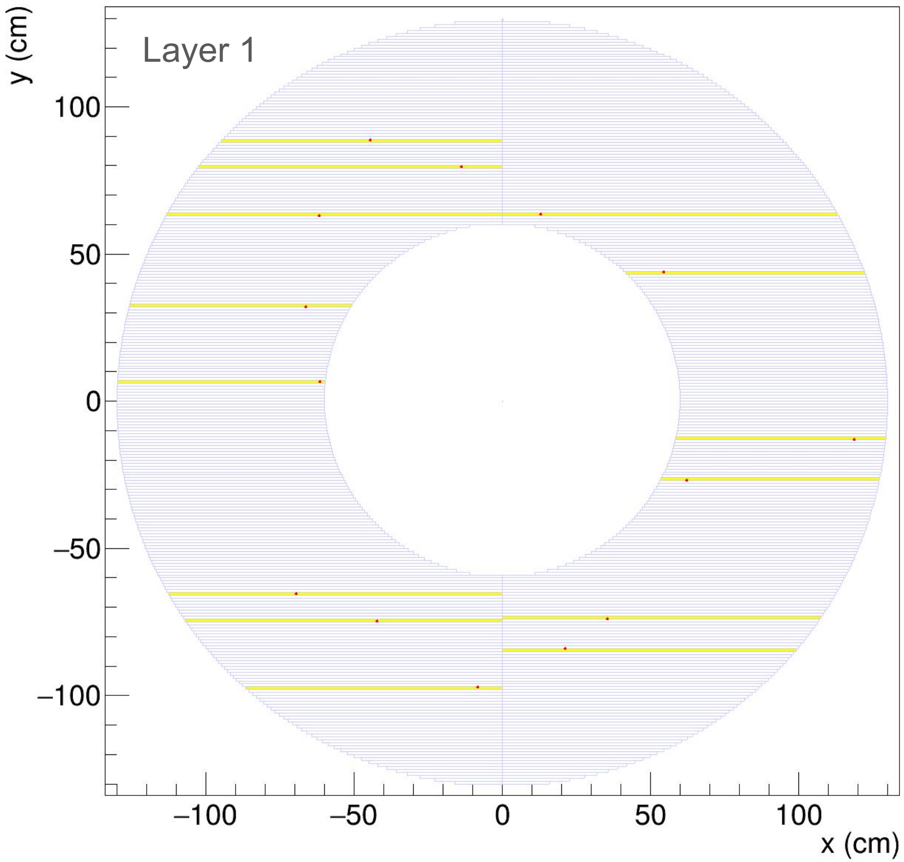


- 5 stations with pixel-like 2D layers

- 4 stations with 2 strip-like 1D layers + 1 pixel-like 2D layer
- In strip-like stations:
  first layer measures x coordinate
  second layer measures y coordinate

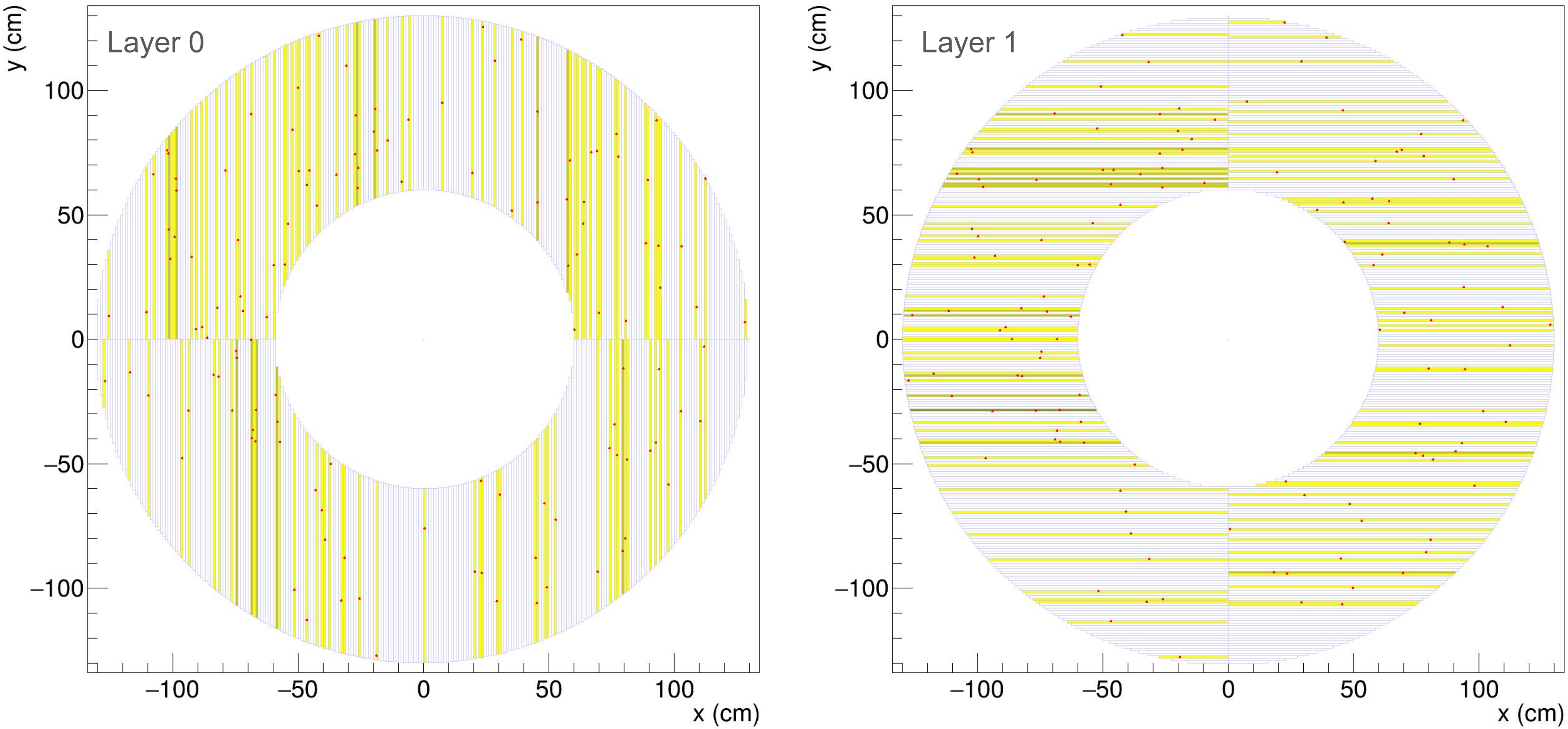
Typical UrQMD event in strip-like forward detector station





Simple strip-like geometry with 1 cm strips, second layer rotated by 90 degrees

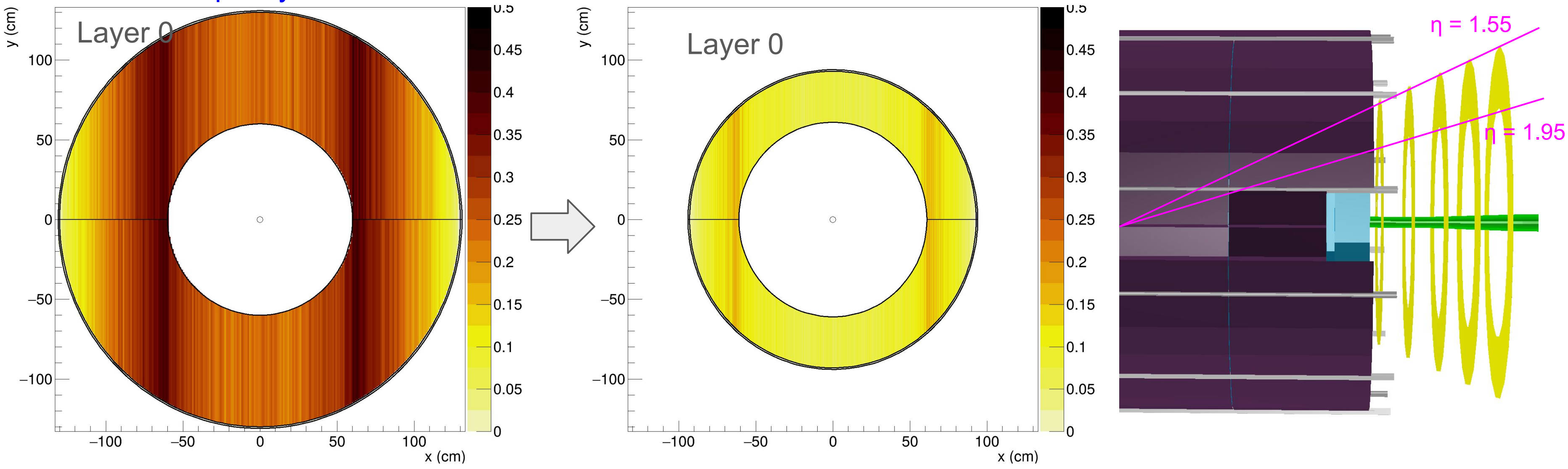
High-multiplicity UrQMD event example



Too high occupancy… Consider thinner/shorter/segmented strips?

Reducing occupancy…

Occupancy in central Au-Au events at 11 GeV

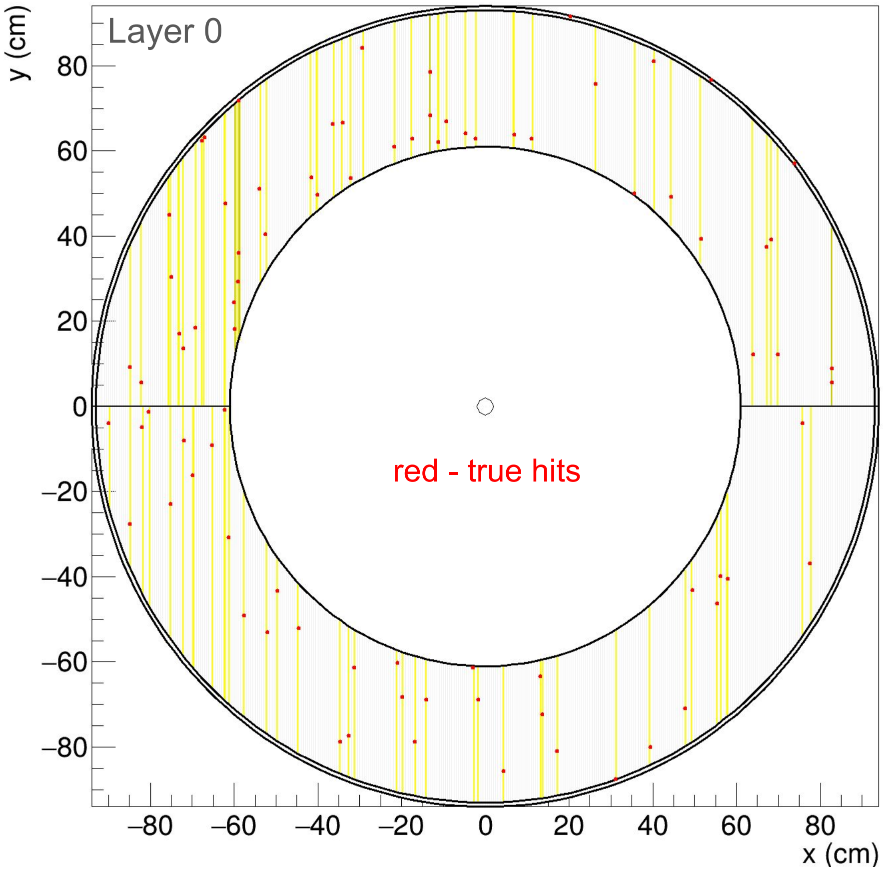


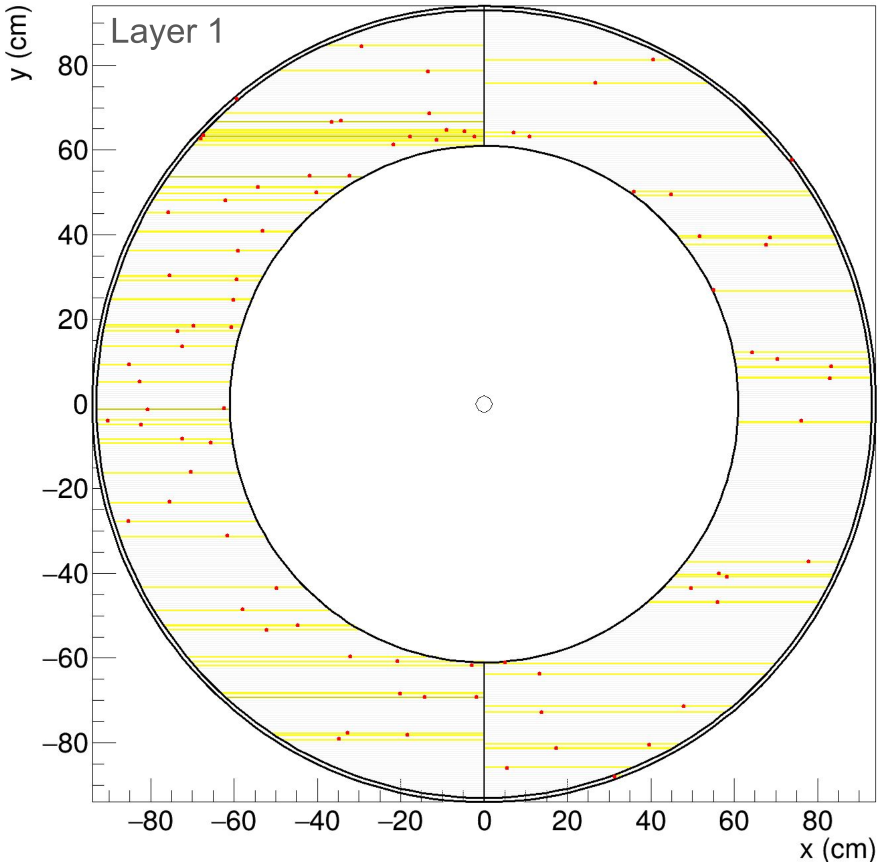
long 1 cm strips: occupancy up to 42%

reduced acceptance, 5-mm strips: occupancy below 19%

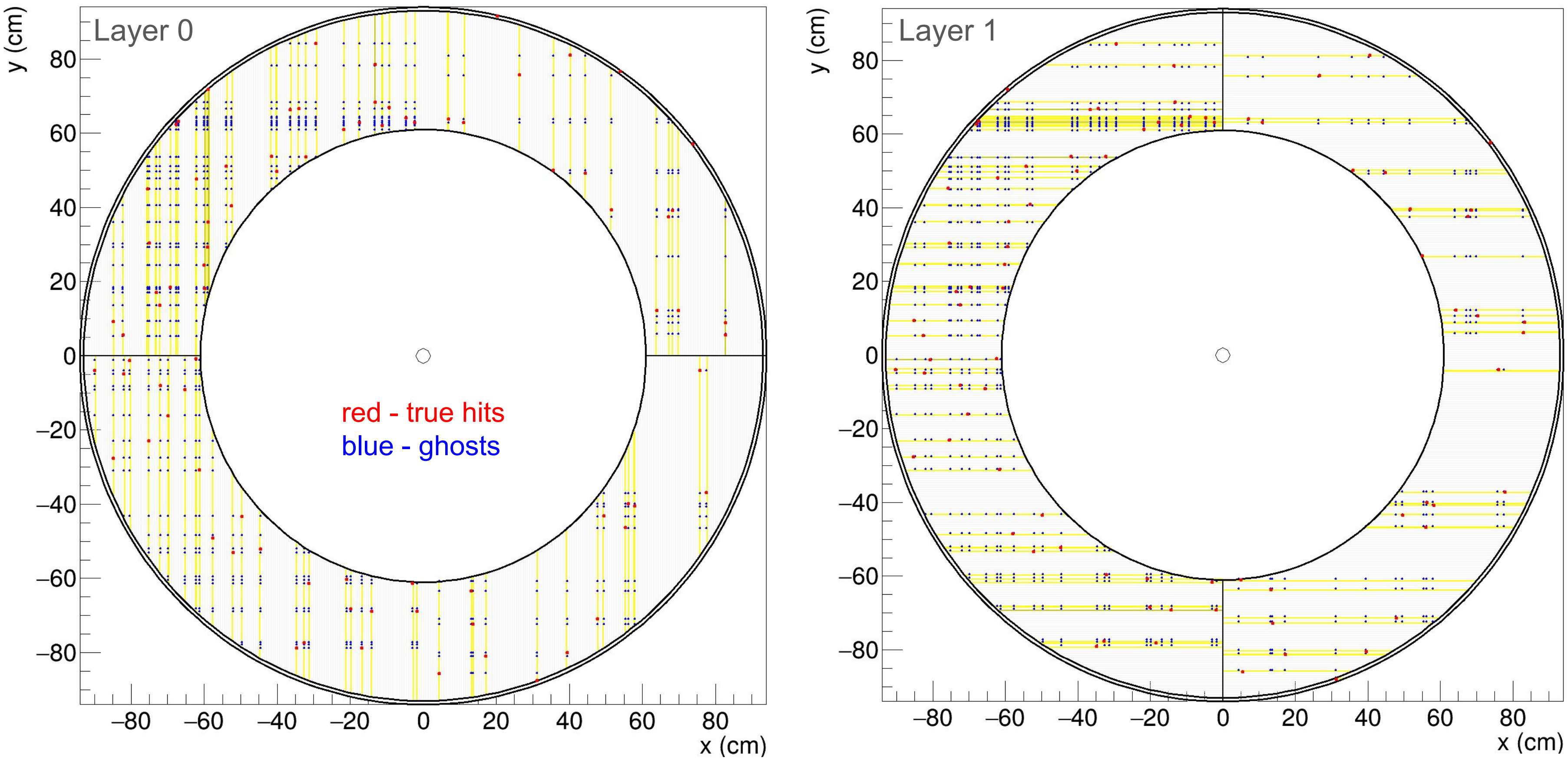
- Strip width can be reduced, e.g. 5-mm straw tubes or MSGCs
- Reducing acceptance of all stations to $1.55 < \eta < 1.95$:
  - $\eta < 1.55$: tracks can be reconstructed in TPC with reasonable p_T resolution (better than 10%)
  - $\eta > 1.95$: large material budget in TPC endcaps - need dedicated study/detector technology

High-multiplicity UrQMD event example with 5mm strips



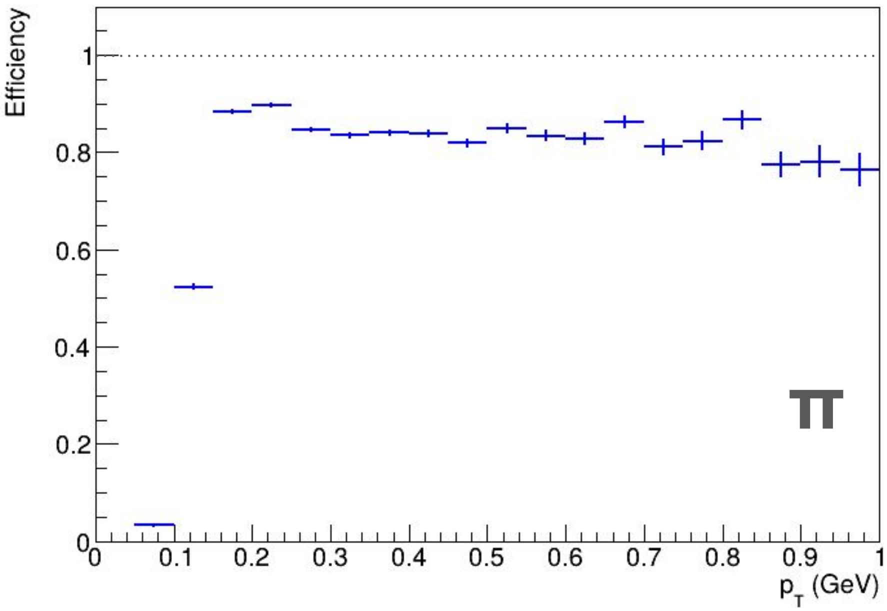


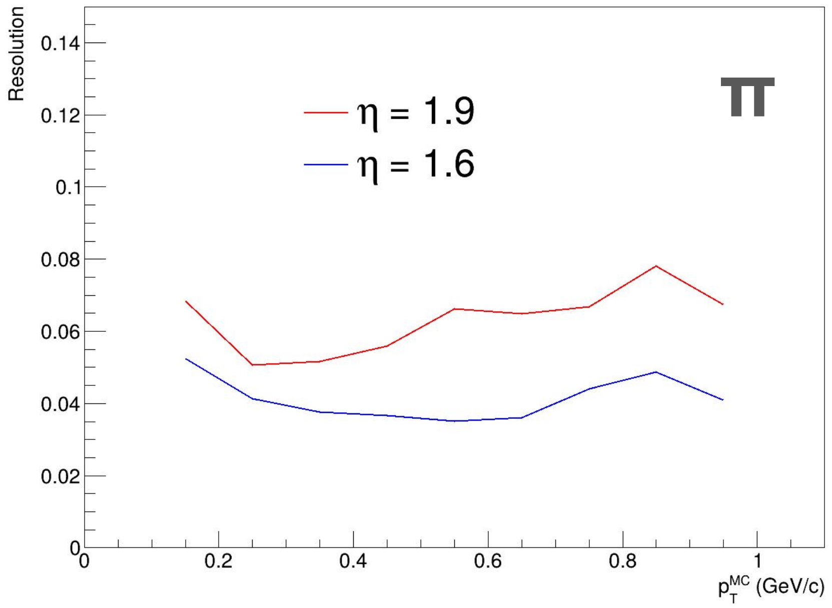
Problem of ghosts



- Ghosts: spacepoints build from all possible intersections of strips fired by different particles

FTD tracking efficiency in central events

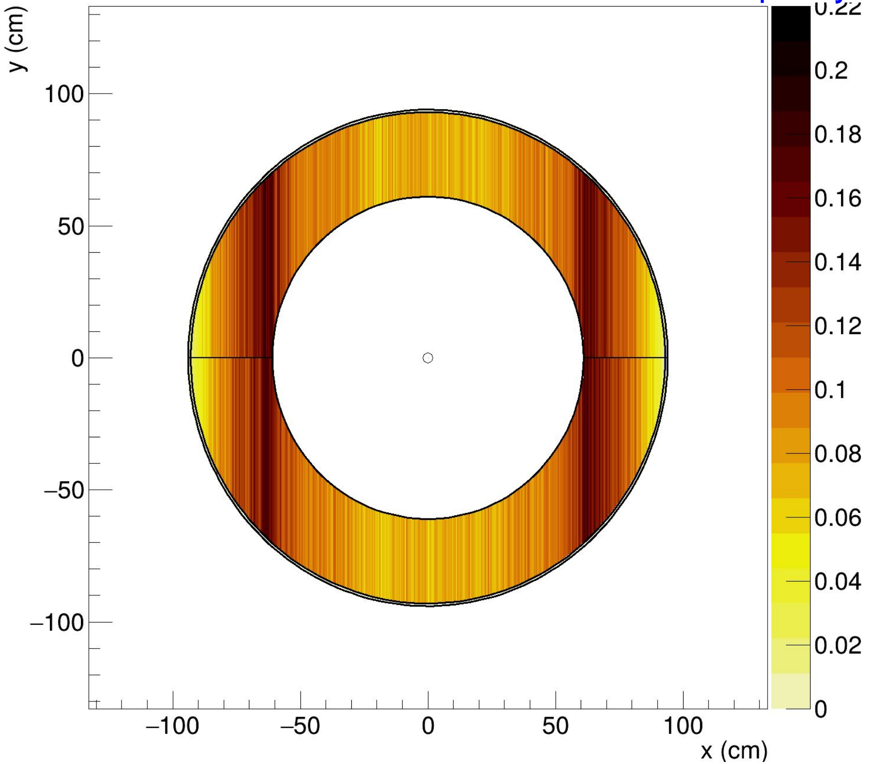




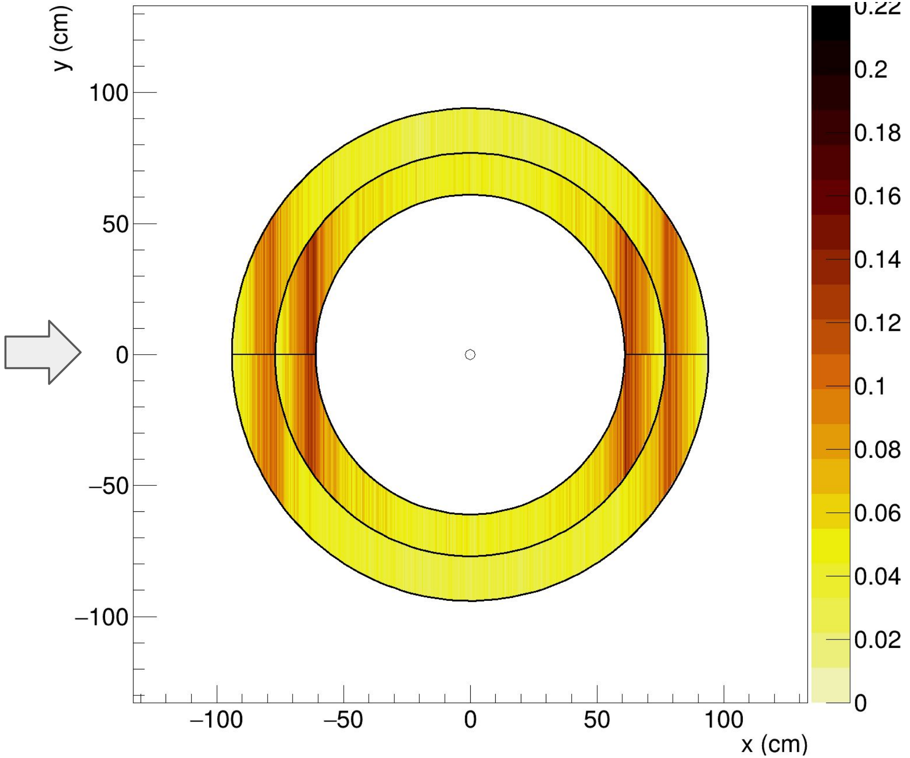
- Seeds using spacepoints (including ghosts) at 1, 3 and 5 stations
- Analysing tracks with 9 MC hits (all layers)
- Reconstruction efficiency (>7 true hits): 80-85%
- Fraction of fake tracks (<6 true hits): ~2.3%
- Momentum resolution similar to 2D-hit setup

Segmented strips (2 rings)

Occupancy in central events

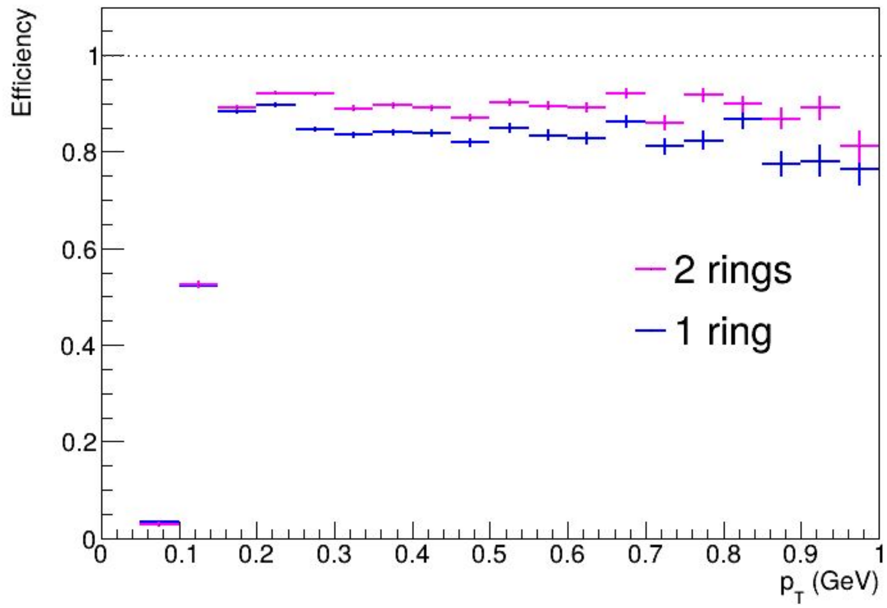


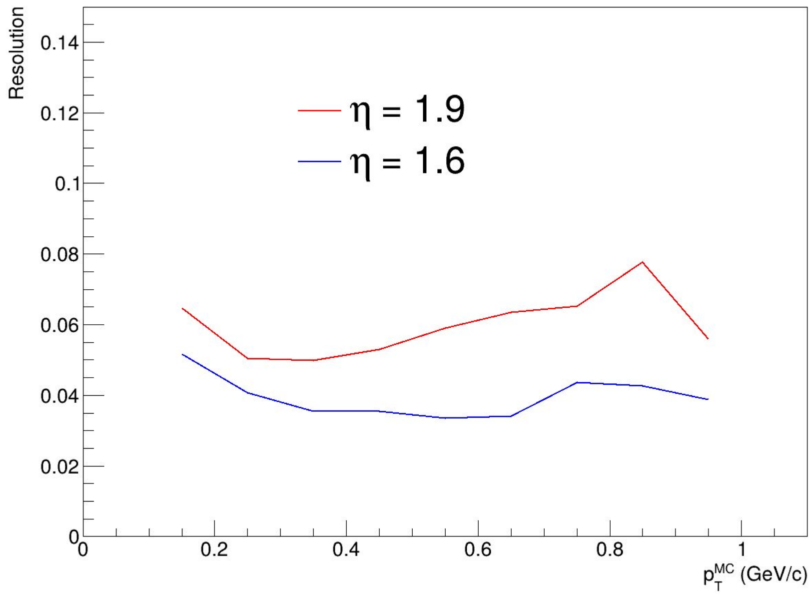
Occupancy up to 19%



Occupancy < 14%

FTD tracking efficiency with 2 rings





- Improved reconstruction efficiency (>7 true hits): ~90%
- Reduced fraction of fake tracks (<6 true hits): ~1.3%
- Further finetuning and optimization ongoing

Code structure

```
ftd
  tracking
    CMakeLists.txt
    MpdFtdActsTracker.cxx
    MpdFtdActsTracker.h
    MpdFtdDetector.cxx
    MpdFtdDetector.h
    MpdFtdSpacePointMaker.cxx
    MpdFtdSpacePointMaker.h
    MpdFtdToActsConverter.cxx
    MpdFtdToActsConverter.h
    MpdRefittingAlgorithm.cxx
    MpdRefittingAlgorithm.h
  CMakeLists.txt
  MpdFtd.cxx
  MpdFtd.h
  MpdFtdGeo.cxx
  MpdFtdGeo.h
  MpdFtdHit.cxx
  MpdFtdHit.h
  MpdFtdHitProducer.cxx
  MpdFtdHitProducer.h
  MpdFtdLinkDef.h
  MpdFtdPoint.cxx
  MpdFtdPoint.h
```

Code structure:

- MpdFtd.h/cxx - FTD geometry and MC point processing
- MpdFtdGeo.h/cxx - FTD geometry settings
- MpdFtdPoint.h/cxx - MC point container for FTD
- MpdFtdHit.h/cxx - reco hit container for FTD
- MpdFtdHitProducer.h/cxx - ideal hit producer for FTD
- Tracking subfolder (compiled if ACTS package is found):
  - MpdFtdDetector.h/cxx - ACTS geometry (TPC + FTD)
  - MpdFtdToActsConverter.h/cxx - converts mpdroot info to ACTS format
  - MpdFtdActsTracker.h/cxx - FairTask: ACTS tracking in TPC and/or FTD
  - MpdRefittingAlgorithm.h/cxx - Refitting with GlobalChi2 fitter (+mass hypothesis)
  - MpdFtdSpacePointMaker.h/cxx - Making space points from 1D hits in FTD

Running procedure:

```
FairTask* ftdHitProducer = new MpdFtdHitProducer();
fRun->AddTask(ftdHitProducer);

FairTask* ftdActsTracker = new MpdFtdActsTracker();
fRun->AddTask(ftdActsTracker);
```

Fork in git: https://git.jinr.ru/ekryshen/mpdroot

Algorithms used in MpdFtdActsTracker

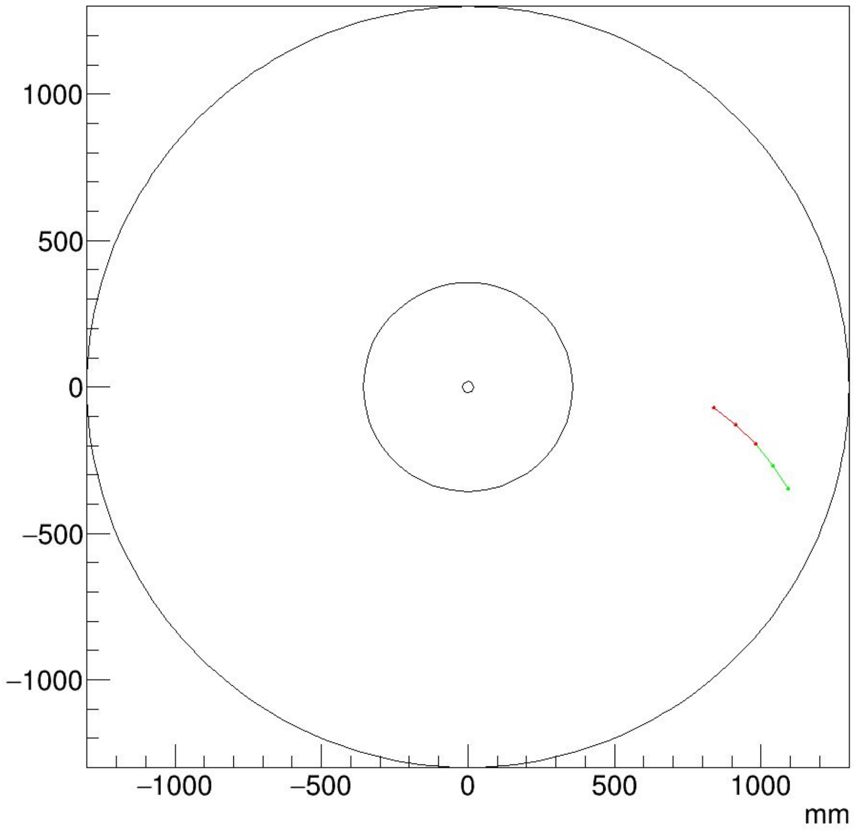
Trying to use native ACTS algorithms when possible

```
// create algorithms
fConverter = new MpdFtdToActsConverter(converterCfg, logLevel);
fSpacePointMaker = new ActsExamples::MpdFtdSpacePointMaker(spCfg, logLevel);
fSeedingAlgorithm = new ActsExamples::SeedingAlgorithm(seedingCfg, logLevel);
fTrackParamsEstimationAlgorithm = new ActsExamples::TrackParamsEstimationAlgorithm(paramsEstimationCfg, logLevel);
fTrackFindingAlgorithm = new ActsExamples::TrackFindingAlgorithm(trackFindingCfg, logLevel);
if (fDoRefit) fTrackRefittingAlgorithm = new ActsExamples::MpdRefittingAlgorithm(trackRefitCfg, logLevel);
fTrackTruthMatcher = new ActsExamples::TrackTruthMatcher(trackTruthMatcherCfg, logLevel);
fRootParticleWriter = new ActsExamples::RootParticleWriter(particleWriterCfg, logLevel);
fRootSimHitWriter = new ActsExamples::RootSimHitWriter(simhitWriterCfg, logLevel);
fRootMeasurementWriter = new ActsExamples::RootMeasurementWriter(measWriterCfg, logLevel);
fRootSpacepointWriter = new ActsExamples::RootSpacepointWriter(spWriterCfg, logLevel);
fRootSeedWriter = new ActsExamples::RootSeedWriter(seedWriterCfg, logLevel);
fRootTrackStatesWriter = new ActsExamples::RootTrackStatesWriter(trackStatesWriterCfg, logLevel);
fRootTrackSummaryWriter = new ActsExamples::RootTrackSummaryWriter(trackSummaryWriterCfg, logLevel);
if (fDoRefit) fRootTrackRefitSummaryWriter = new ActsExamples::RootTrackSummaryWriter(trackRefitSummaryWriterCfg, logLevel);
```

Conclusions and outlook

- Strong physics potential of the forward tracker
- ACTS - powerful track reconstruction tool
- Reasonable performance of TPC+FTD tracking with ACTS
- Developed custom refitting algorithm in ACTS framework
- Developed custom spacepoint producer from 1D strip-like hits
- Working towards realistic strip-like FTD geometry

Example event: pion 110 MeV at η = 1.6



Visualization: hits in xy plane

- green - findable primary (5 hits, p_T > 100 MeV)
- red - found seed

Seeding algorithm:

- xy plane: helix pointing to (x,y) ~ (0,0).
  impact parameter in r < impactMax ~ rMin
- rz plane: angular difference between two doublets consistent with expected mult. scattering
- selection on impact parameter in z direction