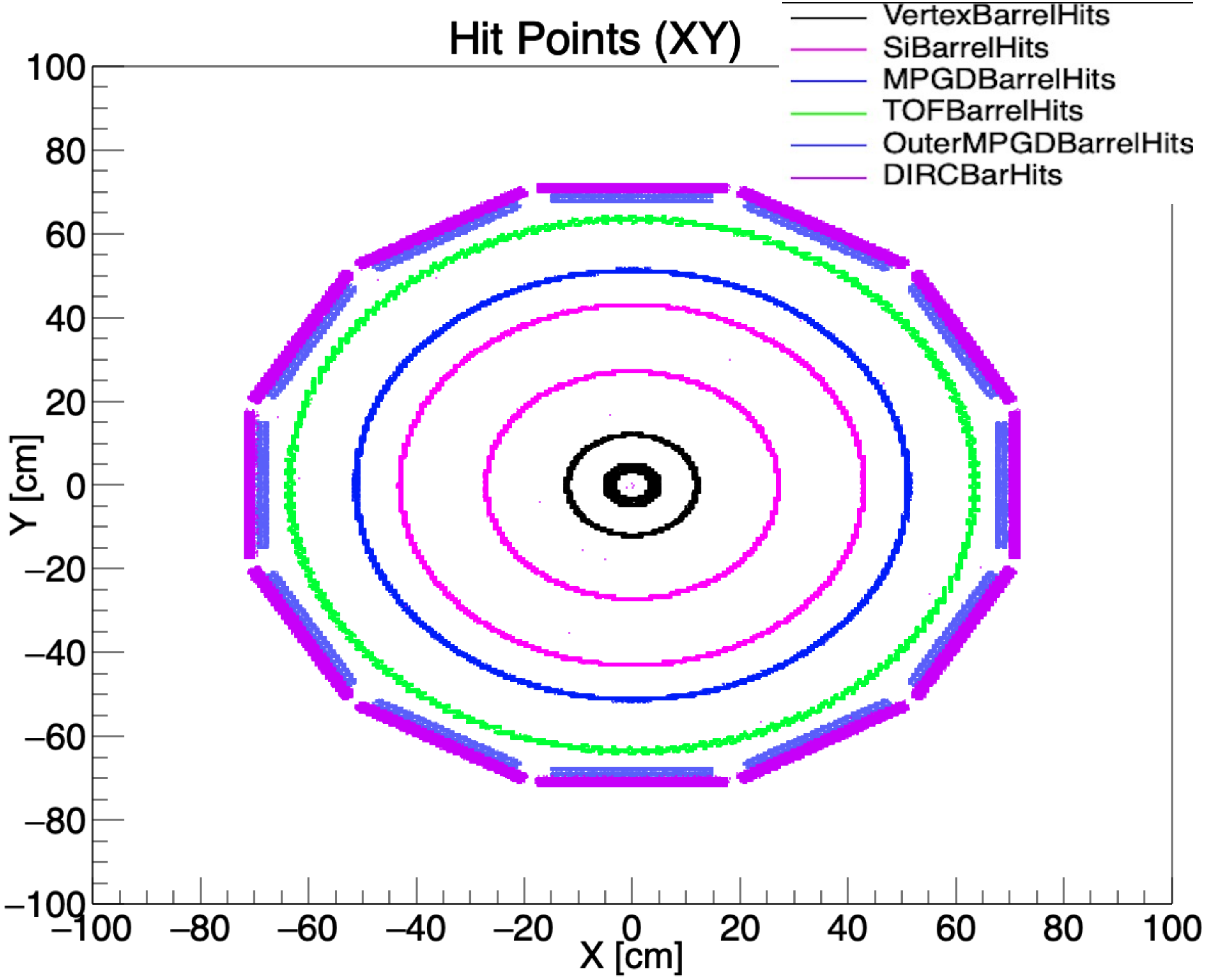
Assessing Angular Resolutions

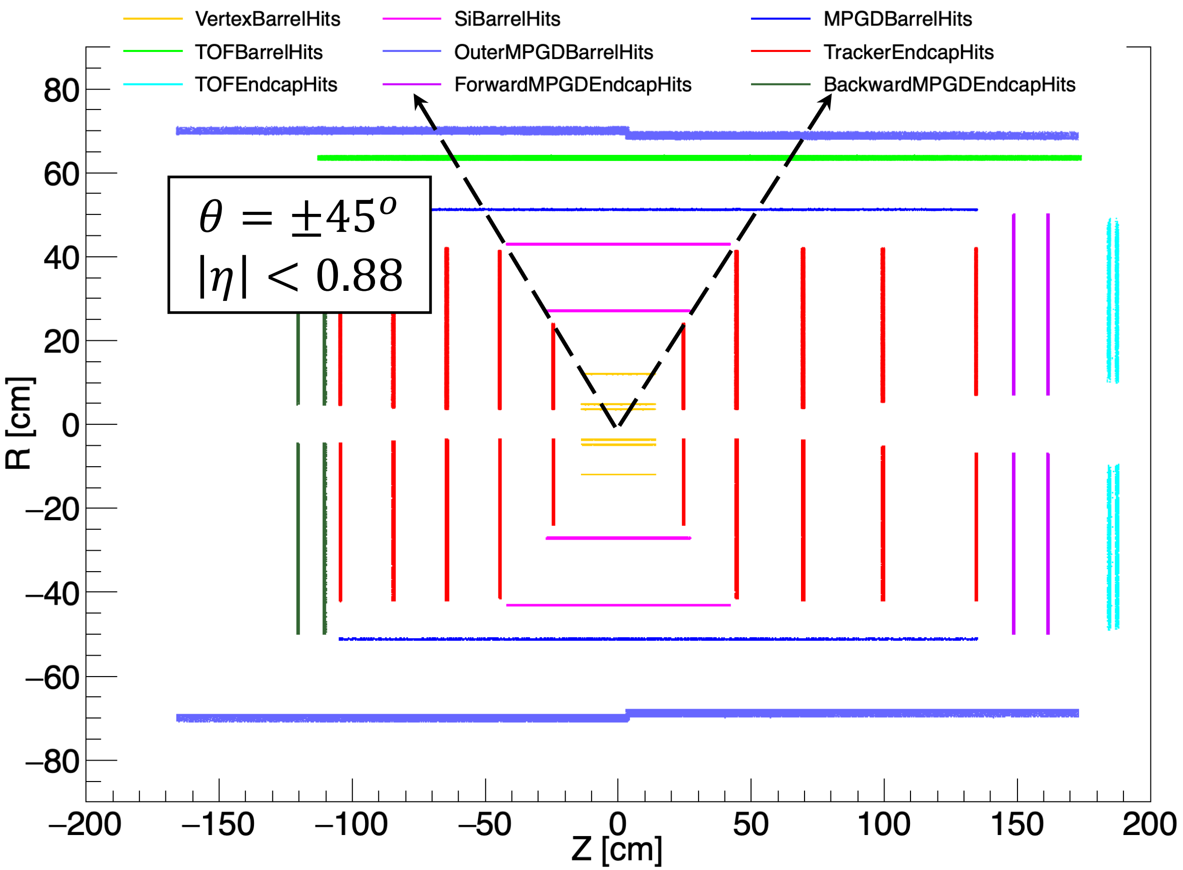
Matt Posik
Temple University

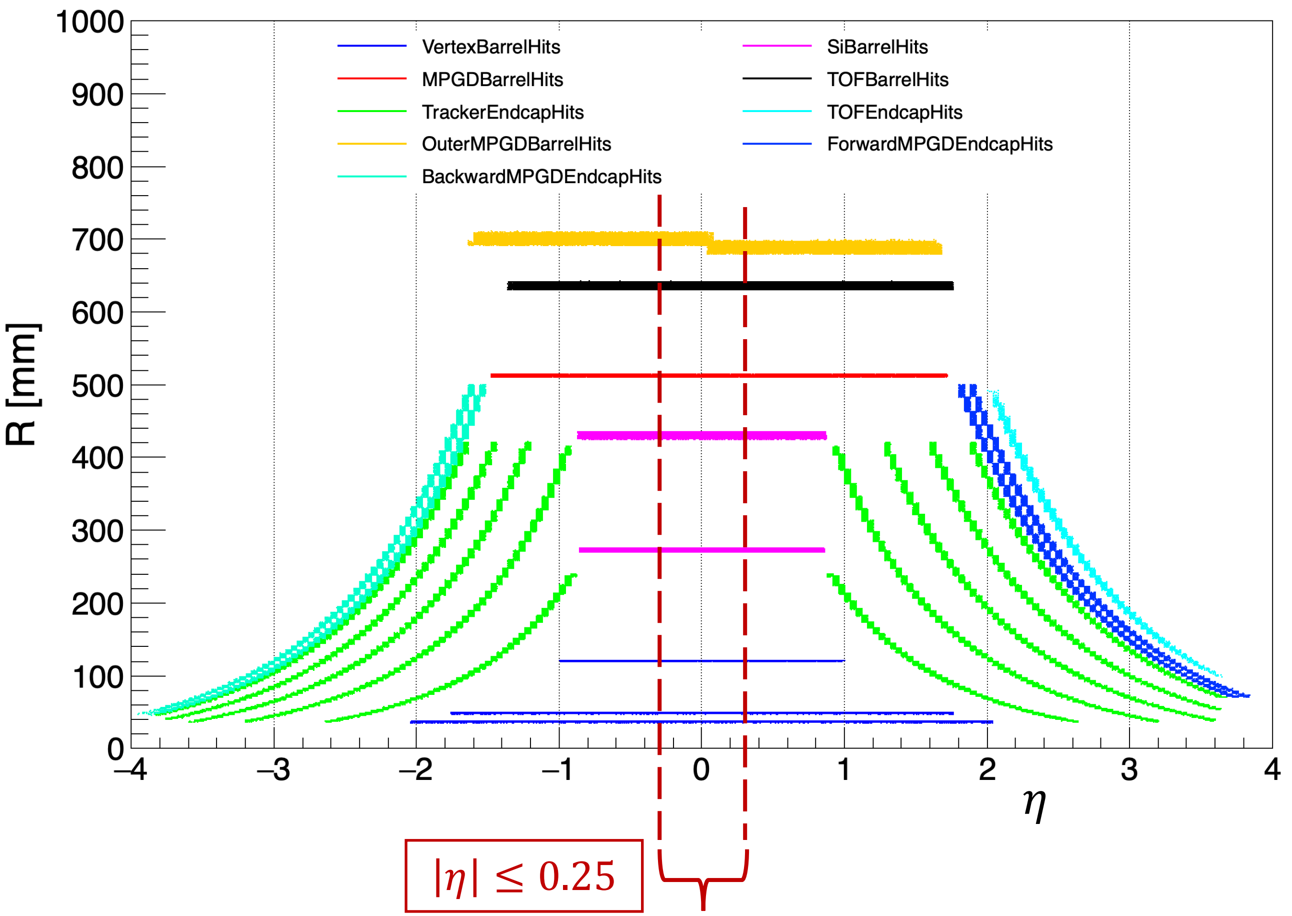
Simulation Details

- Software Version
  - ePIC = 23.07.2
  - Detector Configuration = Craterlake
  - EICRecon = v1.5.1
- Generator
  - Particle Gun = proton
  - ϕ (uniform) = $(0^o, 360^o)$
  - θ (uniform) = $(20^o, 160^o)$ / $(|\eta| \leq 1.73)$
  - p (uniform) = (0.3 GeV, 10.0 GeV)
- hpDIRC Mods
  - Make DIRC bars sensitive volume (provides DIRC hit)
  - Turn off optical photons

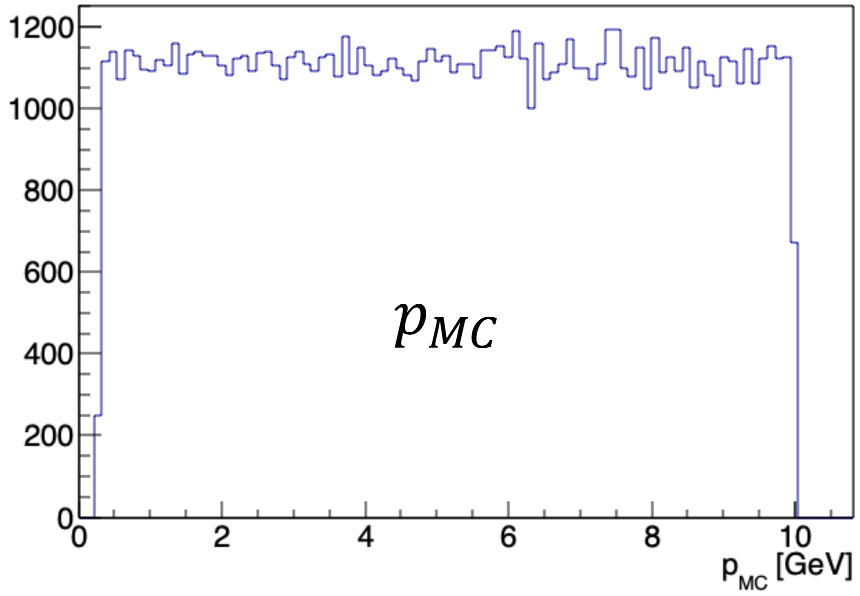


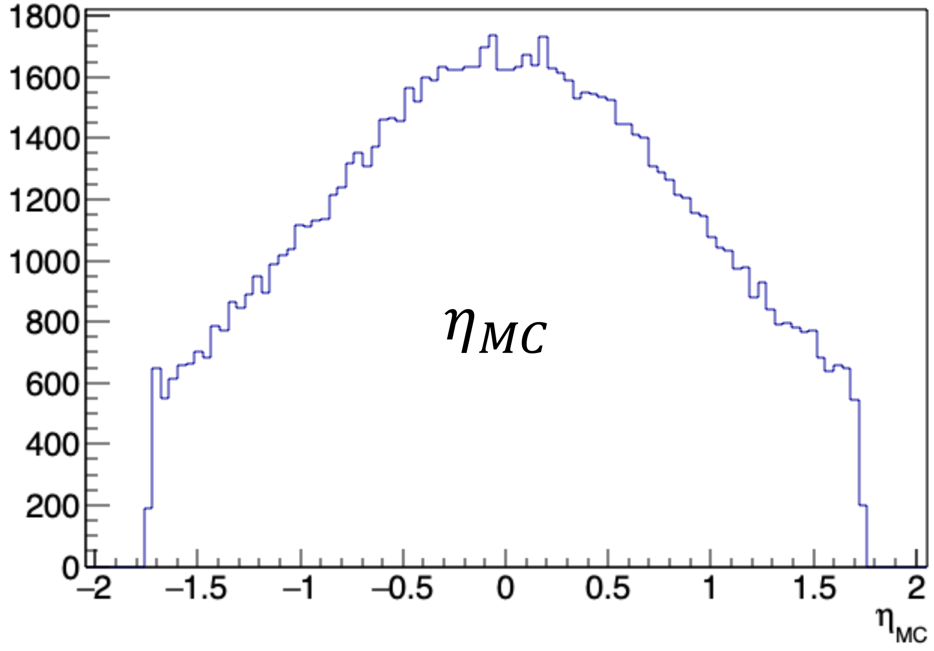
Simulation Details

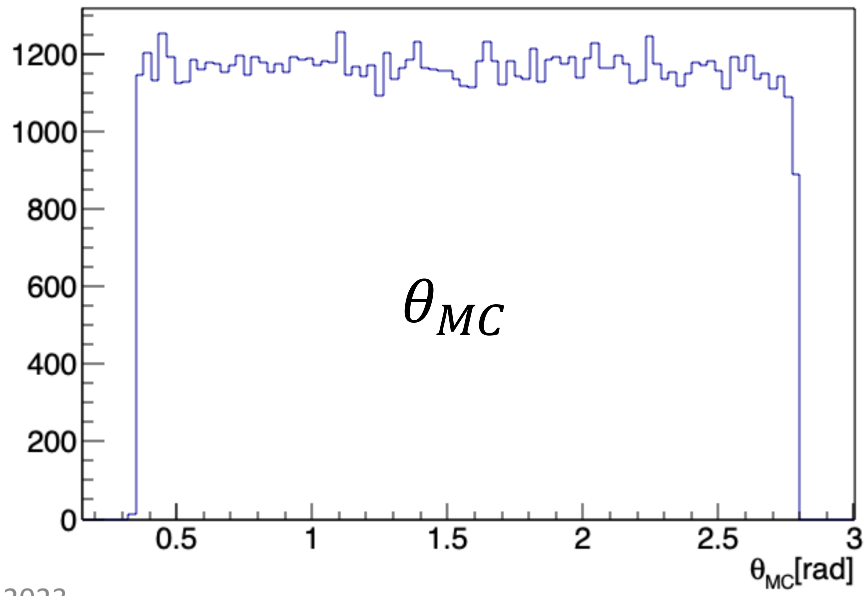


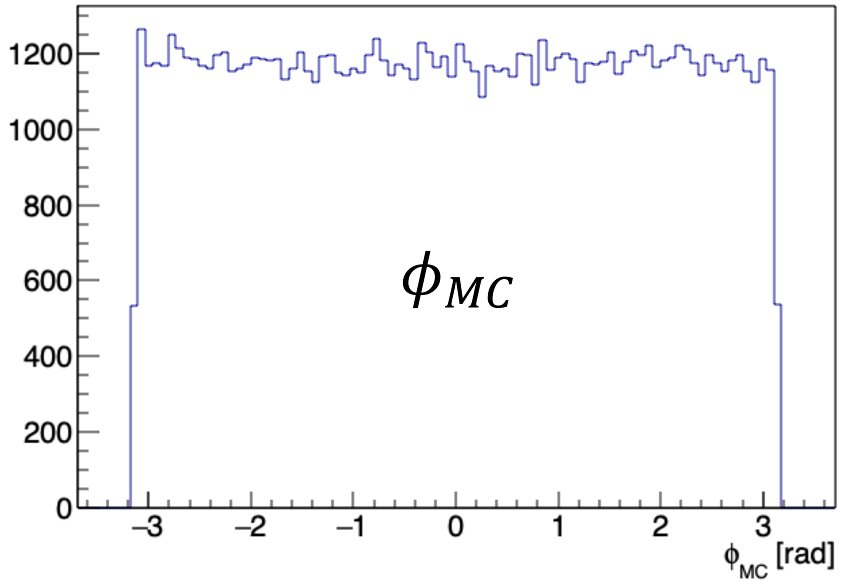


Simulation Distributions: Representative Sample

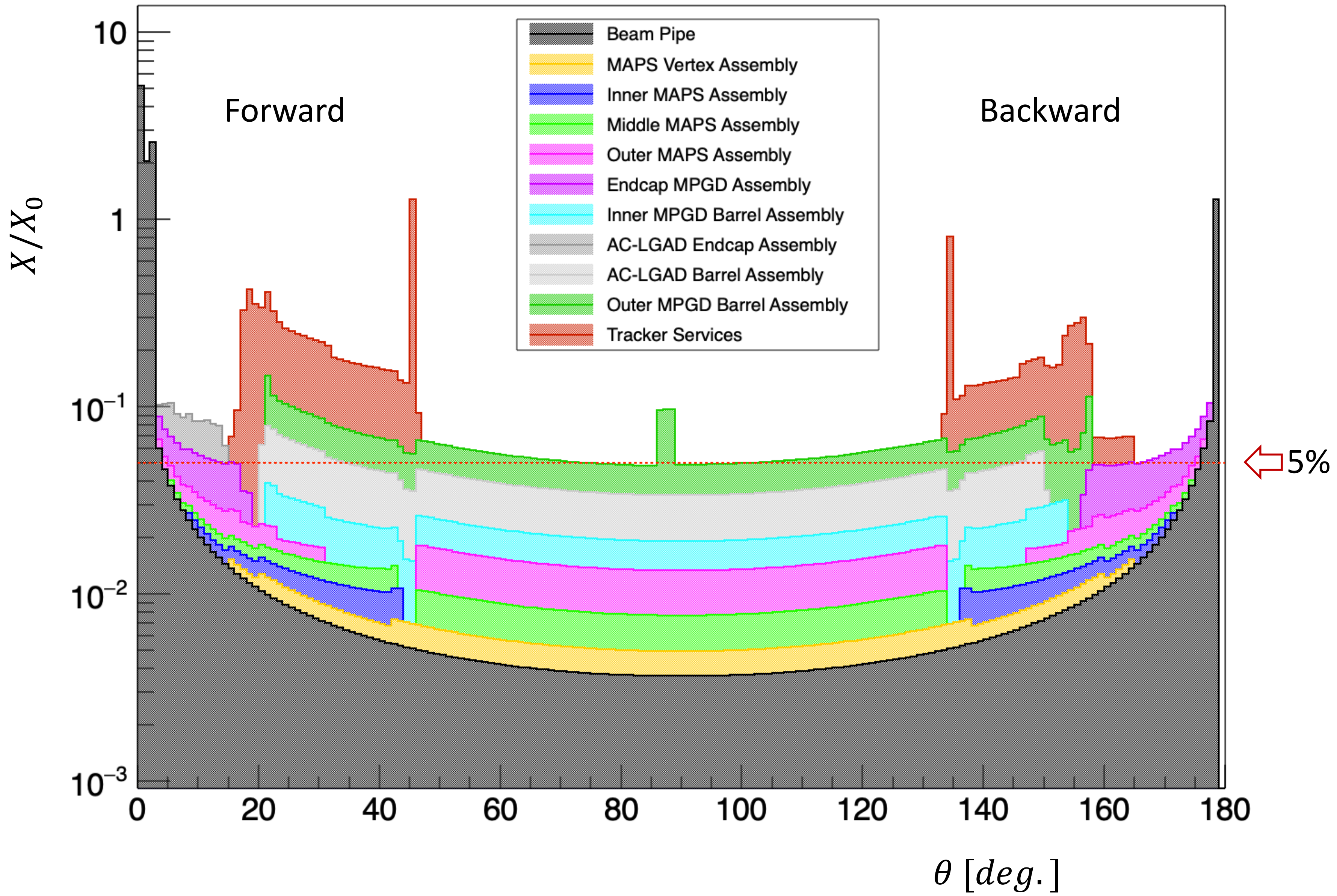






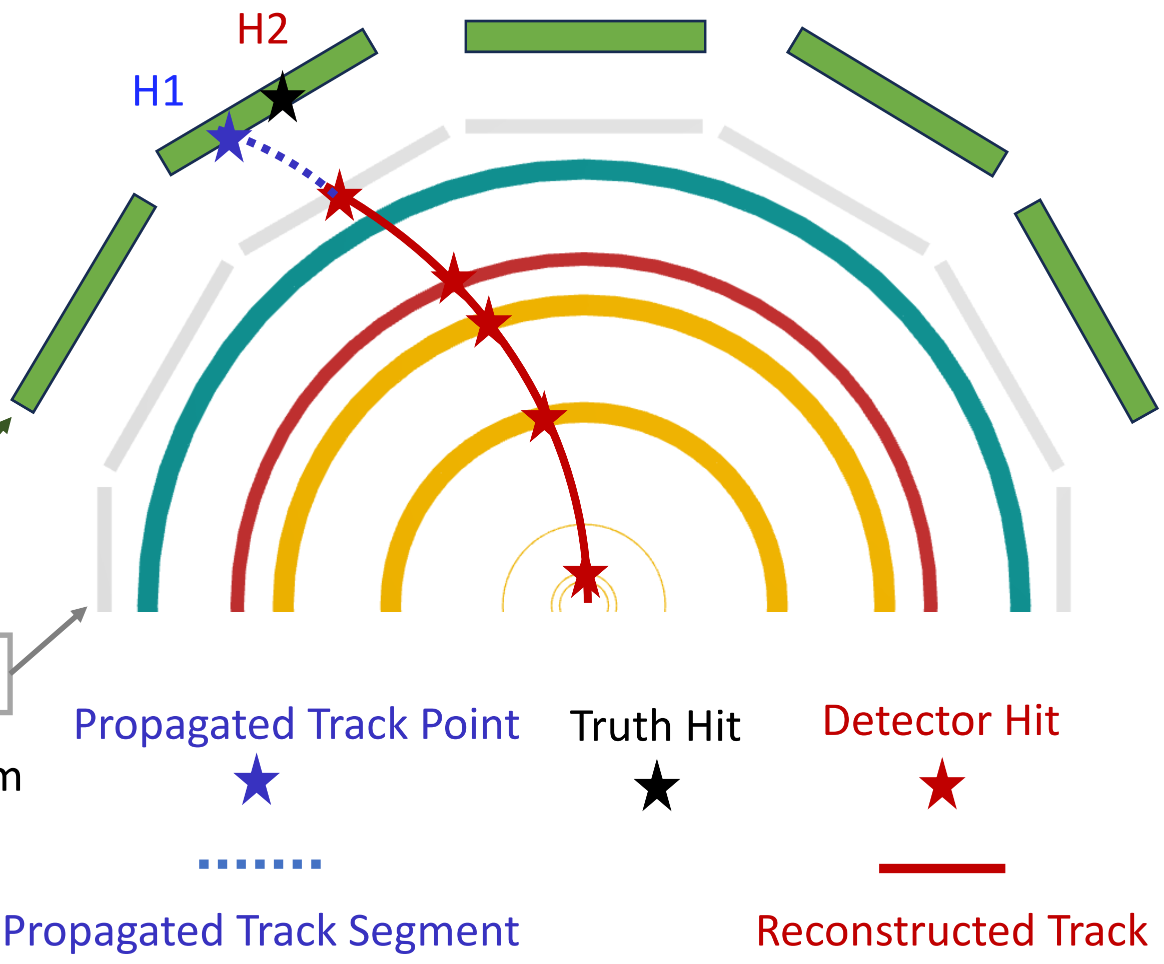


Material Budget



Angular Resolution Method 1

- Propagate trajectory to specified projection surface, track point (H1), and compare to **nearest** DIRC hit (H2) to obtain angles:
  - DIRC point is the true particle hit
  - Propagated trajectory track point (x,y,z) → θ_{prop} , ϕ_{prop}
  - DIRC Point (x,y,z) hits → θ_{dirc} , ϕ_{dirc}
- Angular differences are:
  - $\theta_{prop} - \theta_{dirc}$
  - $\phi_{prop} - \phi_{dirc}$
- Angular resolution $\sigma_\theta, \sigma_\phi$ are extracted from width of assumed Gaussian distribution



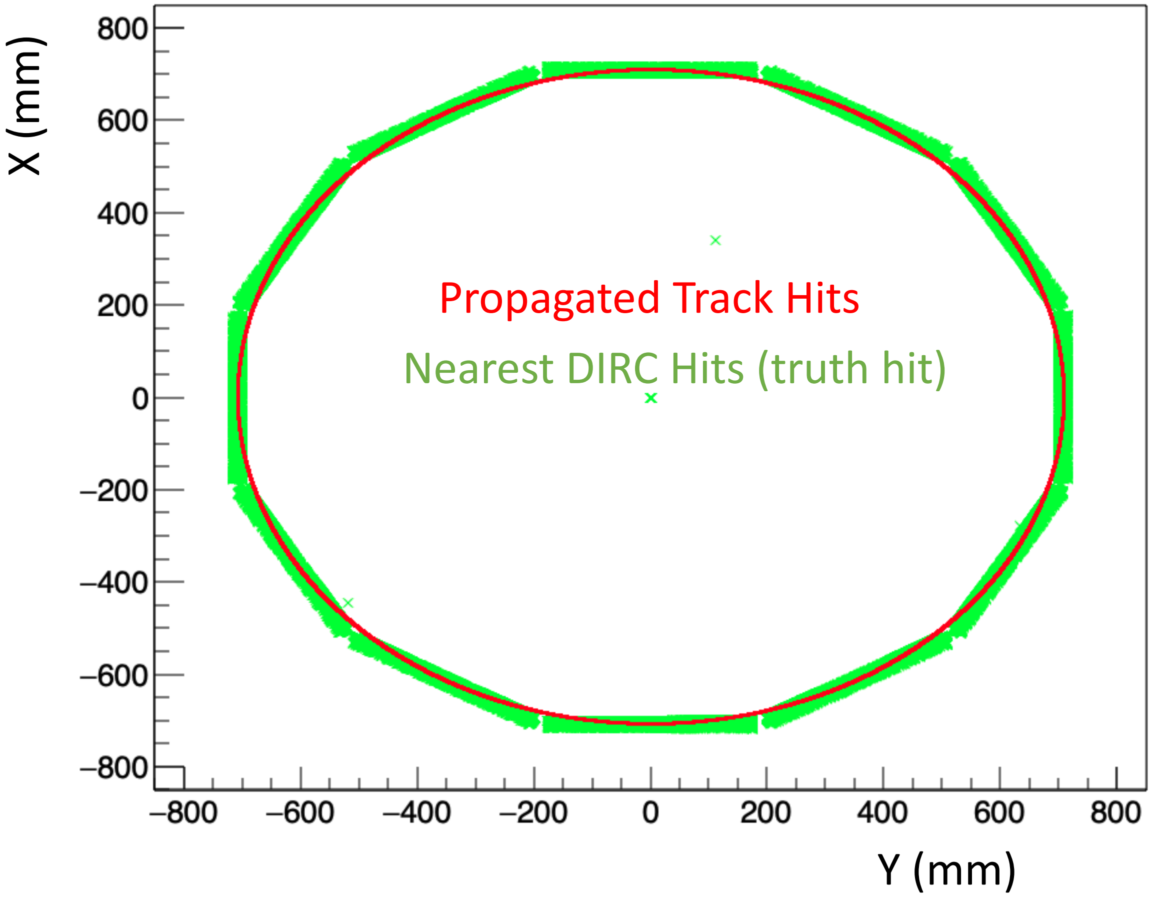
hpDIRC: R = 70 -73 cm

Outer MPGD Barrel: 68.70 cm

Projection Surface: R = 70.85cm

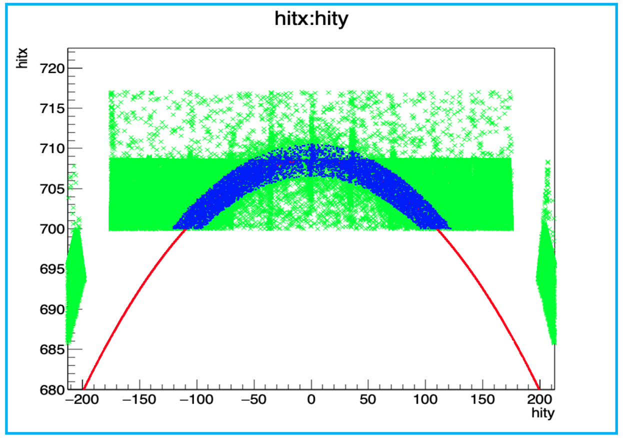
Acceptance Cut

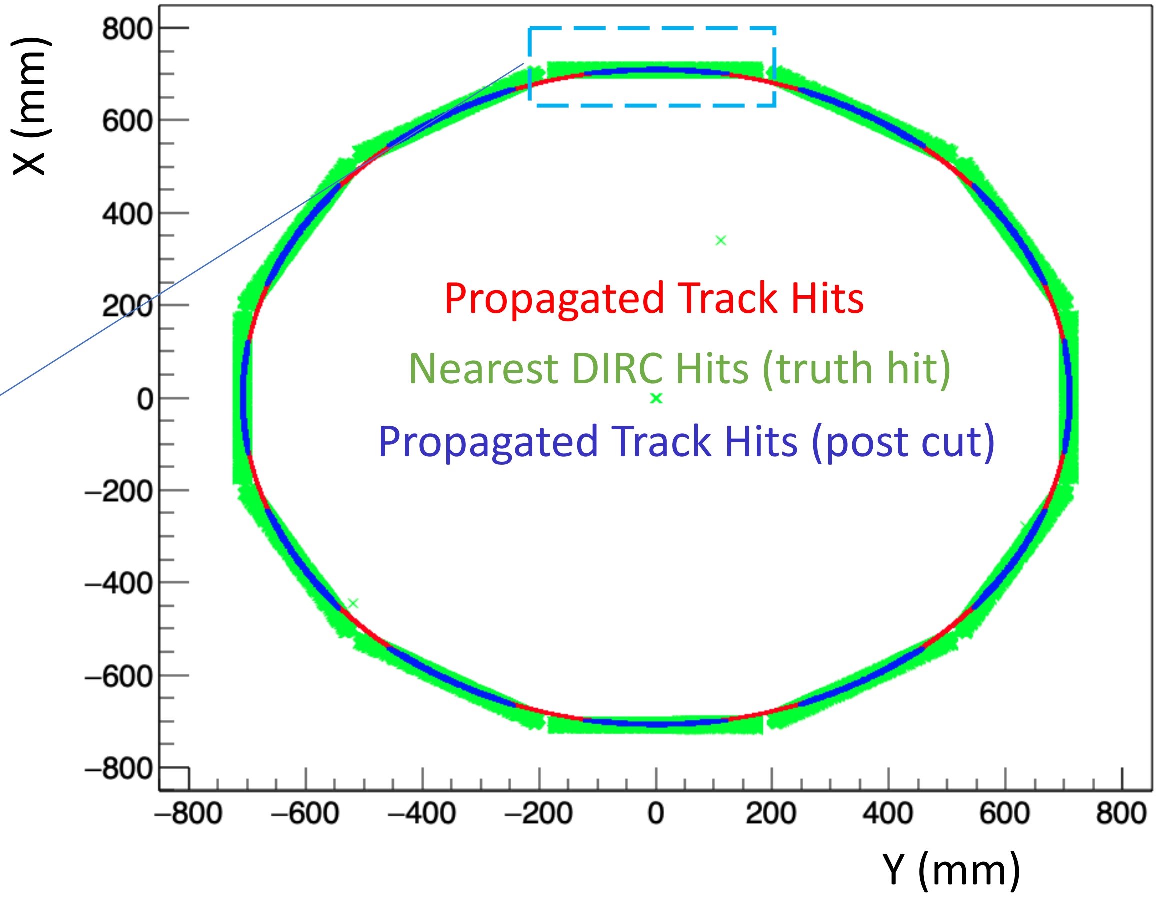
- DIRC bars are rectangular and propagation surface is cylindrical.
  - Need to address geometrical mismatch
- Implement a cut to include DIRC hits that are near the propagation surface
  - $|(x_{prop} - x_{hit})| < 2\ mm$
  - $|(y_{prop} - y_{hit})| < 2\ mm$



Acceptance Cut

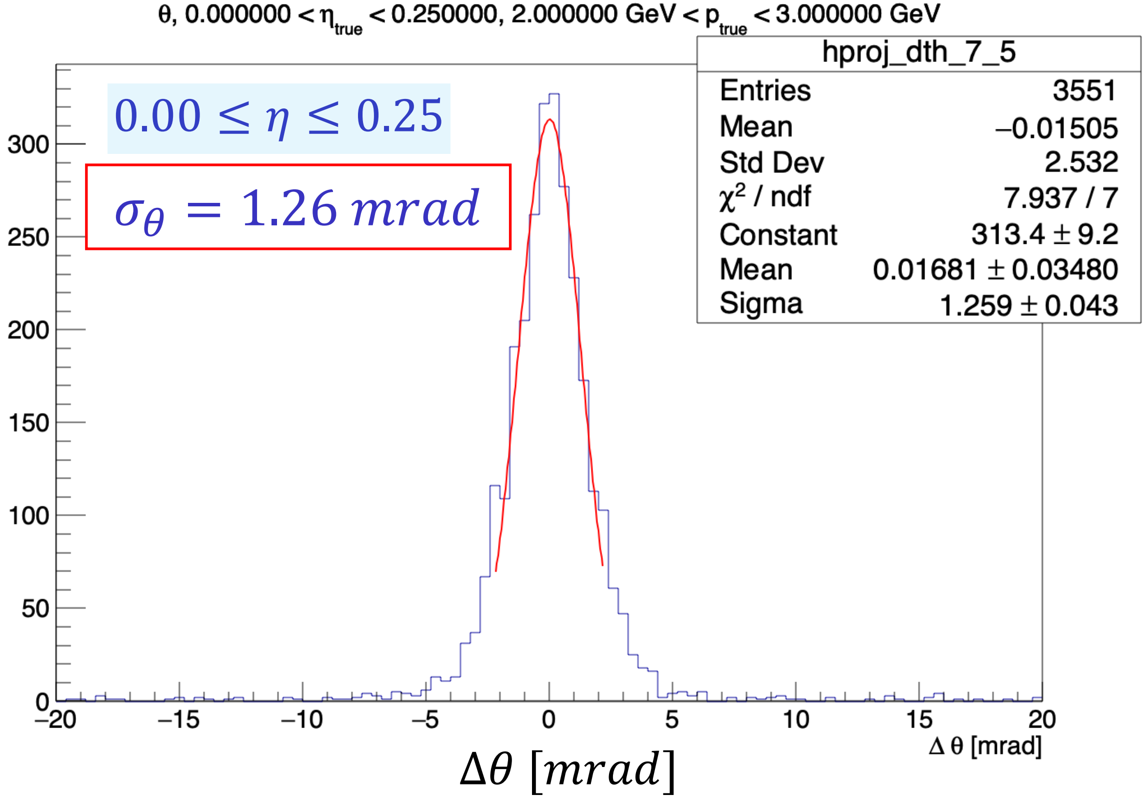
- DIRC bars are rectangular and propagation surface is cylindrical.
  - Need to address geometrical mismatch
- Implement a cut to include DIRC hits that are near the propagation surface
  - $|(x_{proj} - x_{hit})| < 2\ mm$
  - $|(y_{proj} - y_{hit})| < 2\ mm$
- Cuts lead to improvement between truth hit and propagated hit

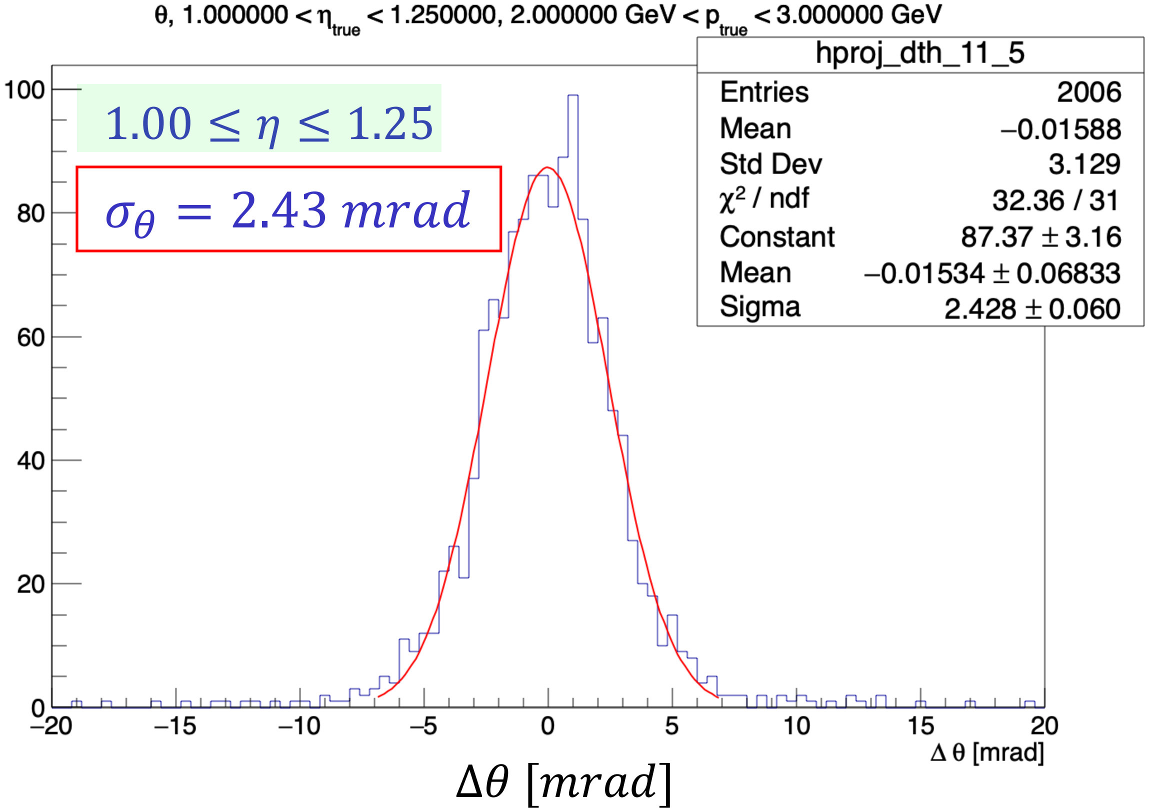




Method 1: Extracting θ Angular Resolution

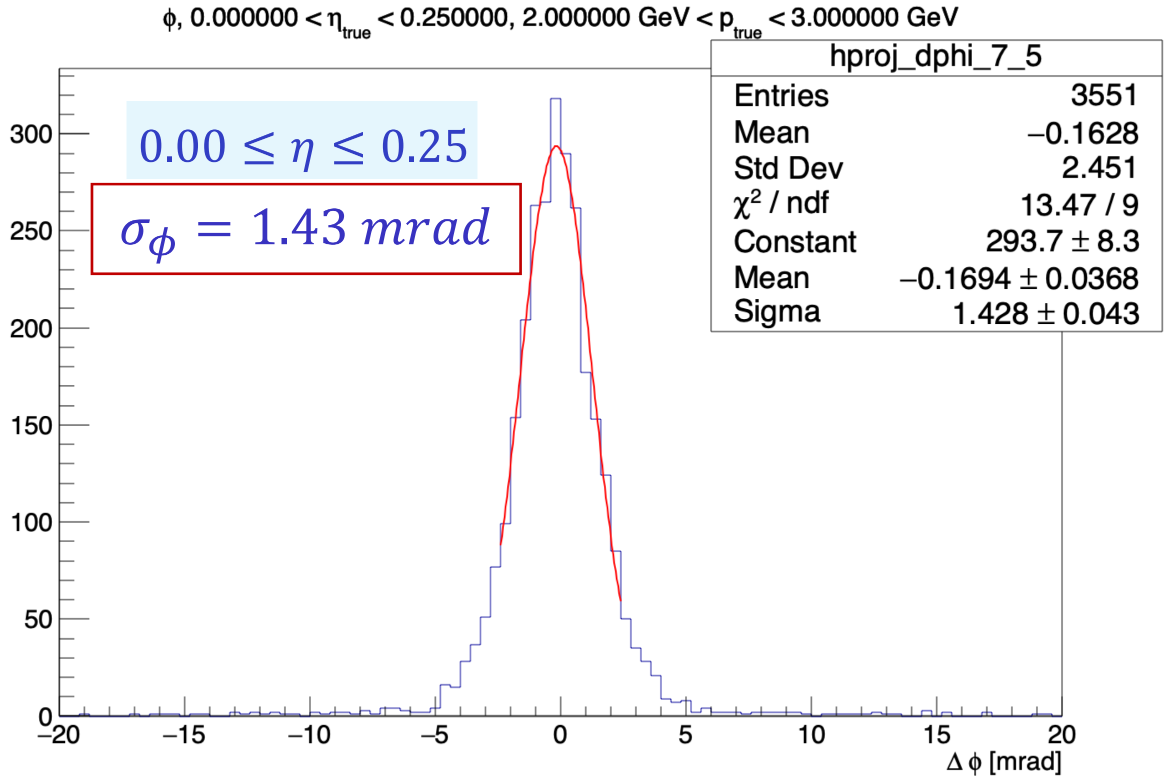
$2.00\ GeV \leq p \leq 3.00\ GeV$



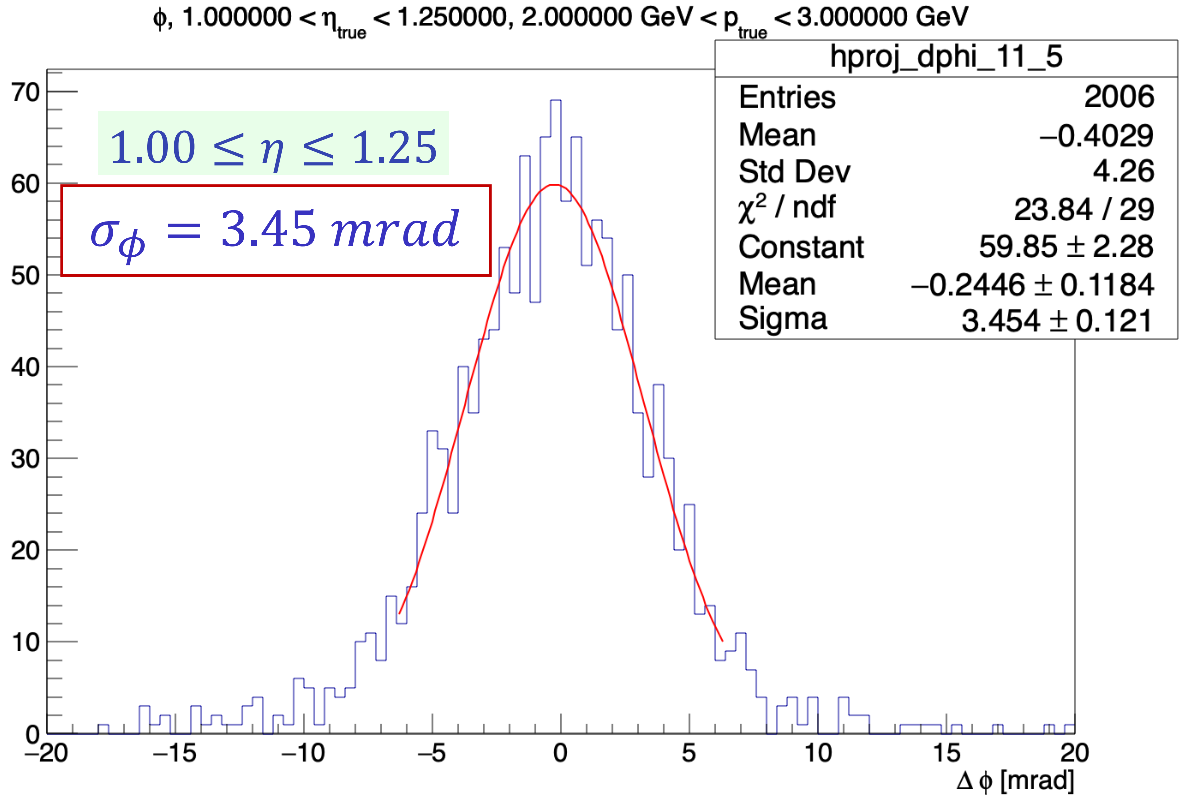


Method 1: Extracting ϕ Angular Resolution

$2.00\ GeV \le p \le 3.00\ GeV$



$\Delta\phi\ [mrad]$

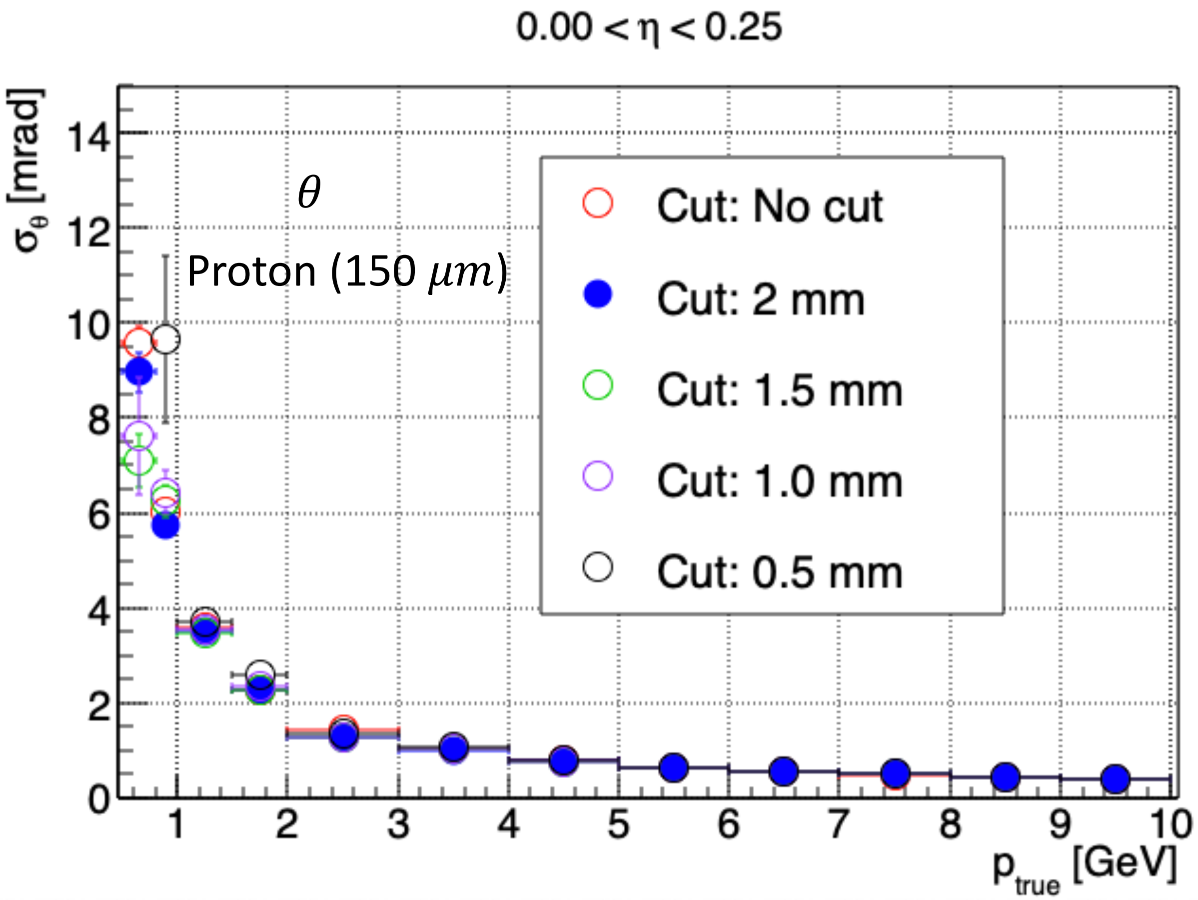


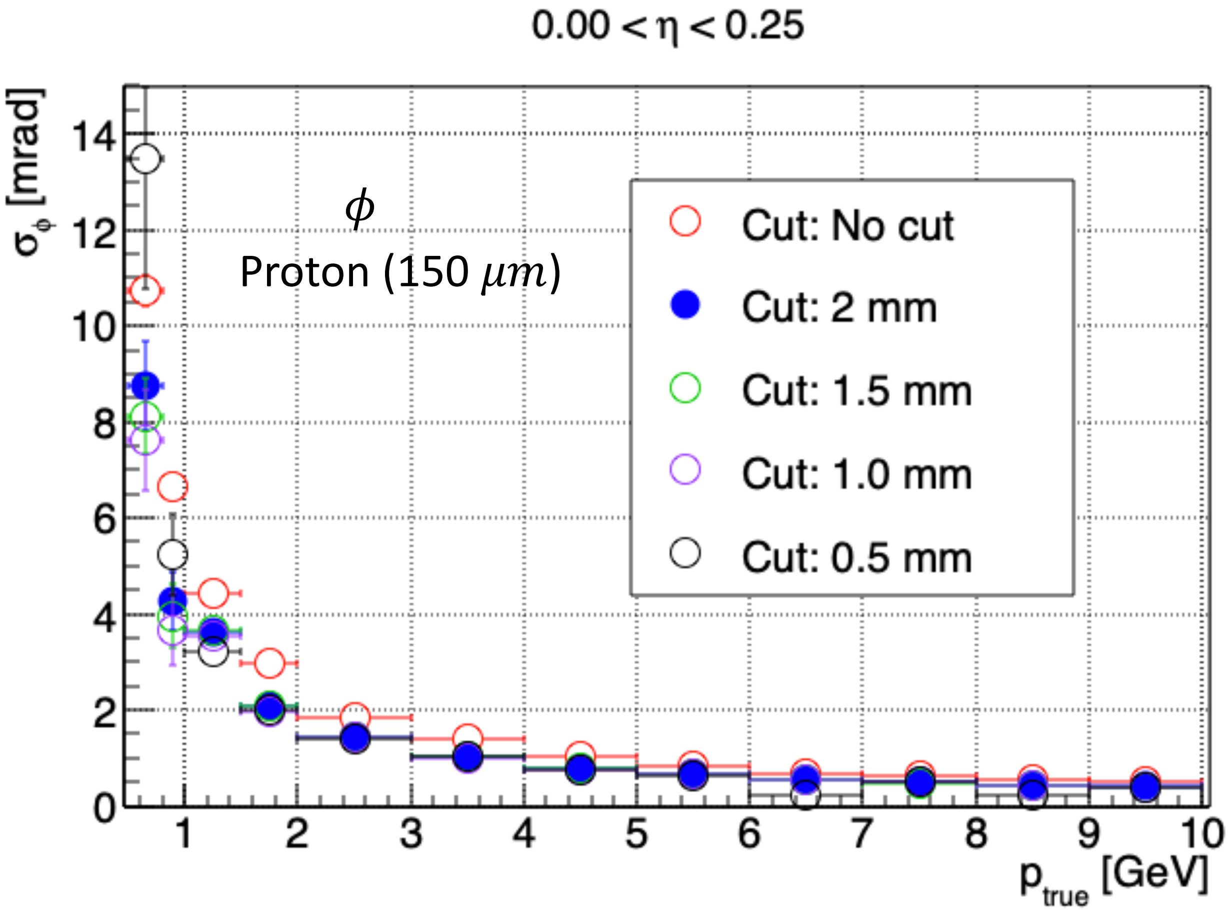
$\Delta\phi\ [mrad]$

Cut Sensitivity

- Cut Sensitivity
  - No much improvement below 2mm – similar trend for other η regions

➢ **Generally, cut provides better resolution, mainly at lower momentum**





Method 2

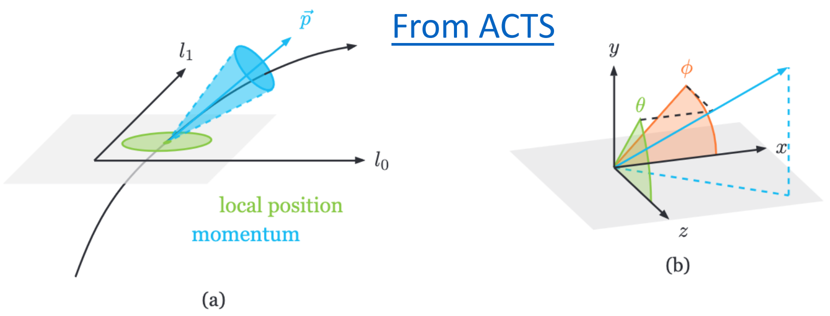
❑ Track Errors

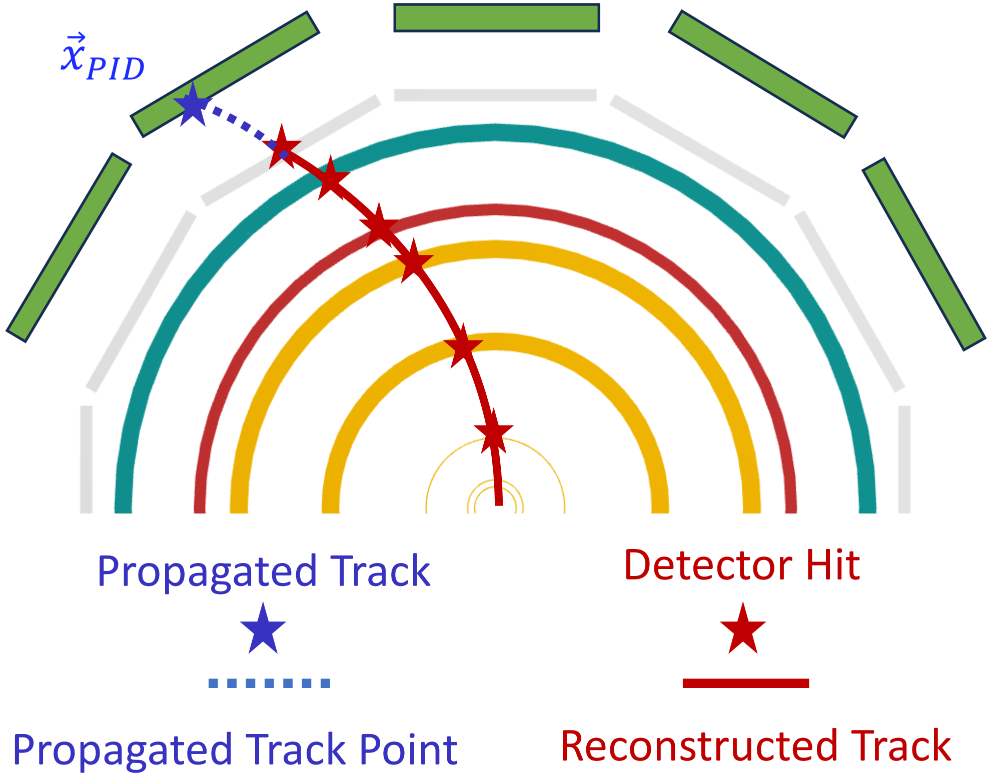
- Use propagated trajectory and track point vector to get track direction impacting PID surface
  - $\vec{x}_{PID} = \left(l_0, l_1, \theta, \phi, \frac{q}{p}\right)$
- Obtain track direction uncertainty from covariance matrix, C

$$C = \begin{bmatrix} \sigma^2(l_0) & \mathrm{cov}(l_0, l_1) & \mathrm{cov}(l_0, \phi) & \mathrm{cov}(l_0, \theta) & \mathrm{cov}(l_0, q/p) \\ . & \sigma^2(l_1) & \mathrm{cov}(l_1, \phi) & \mathrm{cov}(l_1, \theta) & \mathrm{cov}(l_1, q/p) \\ . & . & \sigma^2(\phi) & \mathrm{cov}(\phi, \theta) & \mathrm{cov}(\phi, q/p) \\ . & . & . & \sigma^2(\theta) & \mathrm{cov}(\theta, q/p) \\ . & . & . & . & \sigma^2(q/p) \end{bmatrix}$$

From ACTS

From ACTS





Method 2: Track Projection Code

- Code Snippets

Loops through trajectories

```
// Get trajectories from tracking
auto trajectories = event->Get<ActsExamples::Trajectories>("CentralCKFActsTrajectories");
//auto trajectoriesSeed = event->Get<ActsExamples::Trajectories>("CentralCKFSeededActsTrajectories");

// Iterate over trajectories
m_log->debug("Propagating through {} trajectories", trajectories.size());
for (size_t traj_index = 0; traj_index < trajectories.size(); traj_index++) {
    auto &trajectory = trajectories[traj_index];
    m_log->trace(" -- trajectory {} --", traj_index);

    // The trajectory entry indices and the multiTrajectory
    const auto& mj        = trajectory->multiTrajectory();
    const auto& trackTips = trajectory->tips();
    if (trackTips.empty()) {
      m_log->trace("Empty multiTrajectory.");
      continue;
    }
```

Gets trajectories

...

Propagated trajectory to specified ACTS surface

Assess track point information at specified ACTS surface

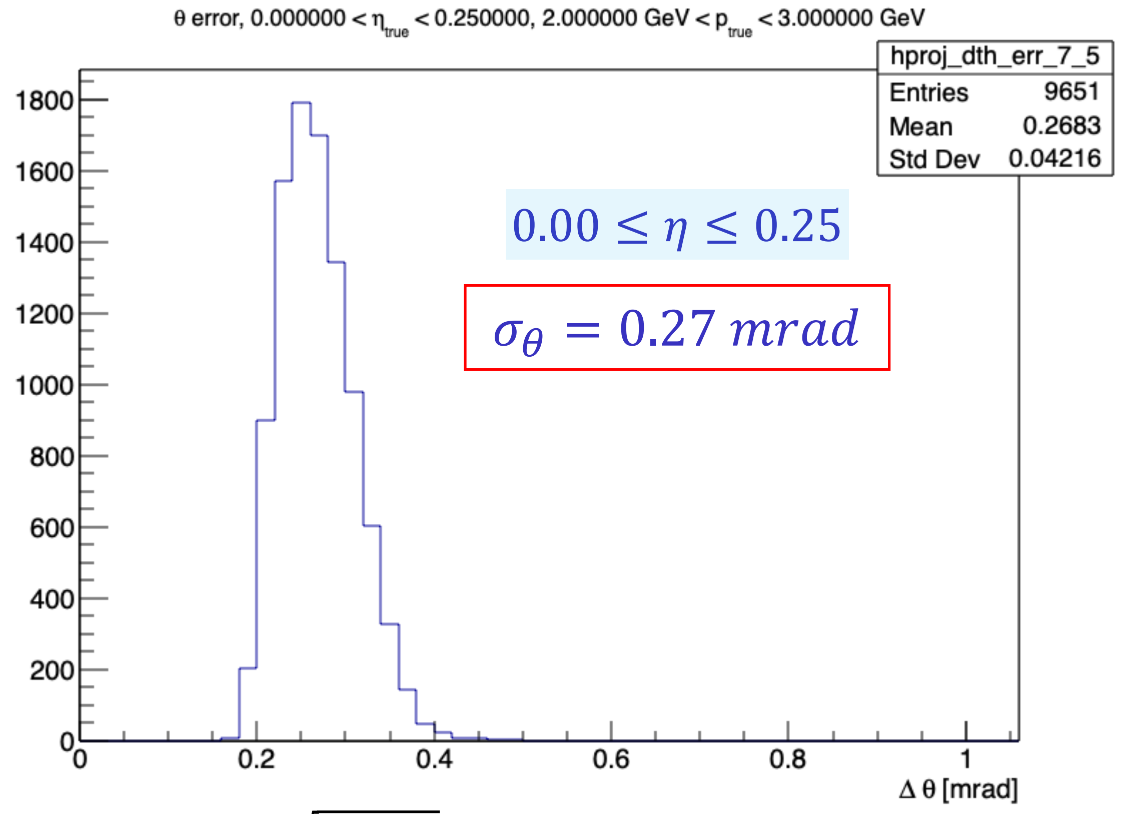
```
    std::unique_ptr<edm4eic::TrackPoint> projection_point;
    try {
        // >>> try to propagate to surface <<<
        projection_point = m_propagation_algo.propagate(trajectory, m_DIRC_center_surface);
    }
    catch(std::exception &e) {
        throw JException(e.what());
    }

    if(!projection_point) {
        m_log->trace("   could not propagate (center)!", traj_index);
        continue;
    }
    // Now go through reconstructed tracks points
    auto pos    = projection_point->position;
    auto length =  projection_point->pathlength;
    auto mom    =  projection_point->momentum;
    TVector3 proj_mom_vec(mom.x,mom.y, mom.z);
    auto momp = sqrt(mom.x*mom.x + mom.y*mom.y + mom.z*mom.z);
    auto phi  = atan2(mom.y,mom.x);
    auto th   = proj_mom_vec.Theta();
    auto eta  = -log(tan(atan2(sqrt(mom.x*mom.x + mom.y*mom.y),mom.z)/2.0));
    m_log->trace("   {:>10} {:>10.2f} {:>10.2f} {:>10.2f} {:>10.2f}", traj_index, pos.x, pos.y, pos.z, length);
    auto theta_center = projection_point->theta;
    auto phi_center   = projection_point->phi;
    auto theta_center_error = projection_point->directionError.xx;
    auto phi_center_error = projection_point->directionError.yy;
    auto theta_phi_center_error = projection_point->directionError.xy;
```

$\sigma^2(\theta), \sigma^2(\phi)$

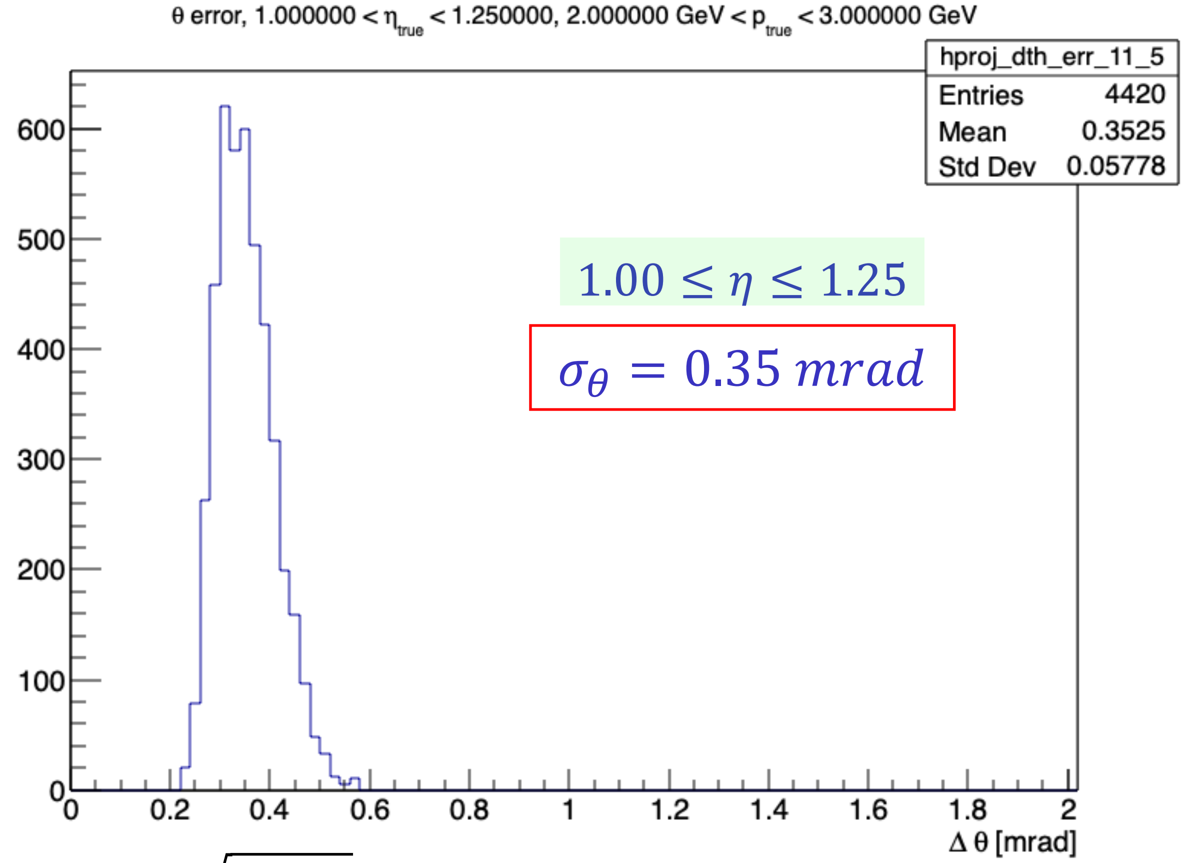
Method 2: Extracting θ Angular Resolution

- Histogram sqrt(variance), variance obtained from covariance matrix
  - Histogram mean = angular uncertainty
  - Histogram RMS = error bar

$2.00\ GeV \leq p \leq 3.00\ GeV$



$\sqrt{\sigma^2(\theta)}\ [mrad]$

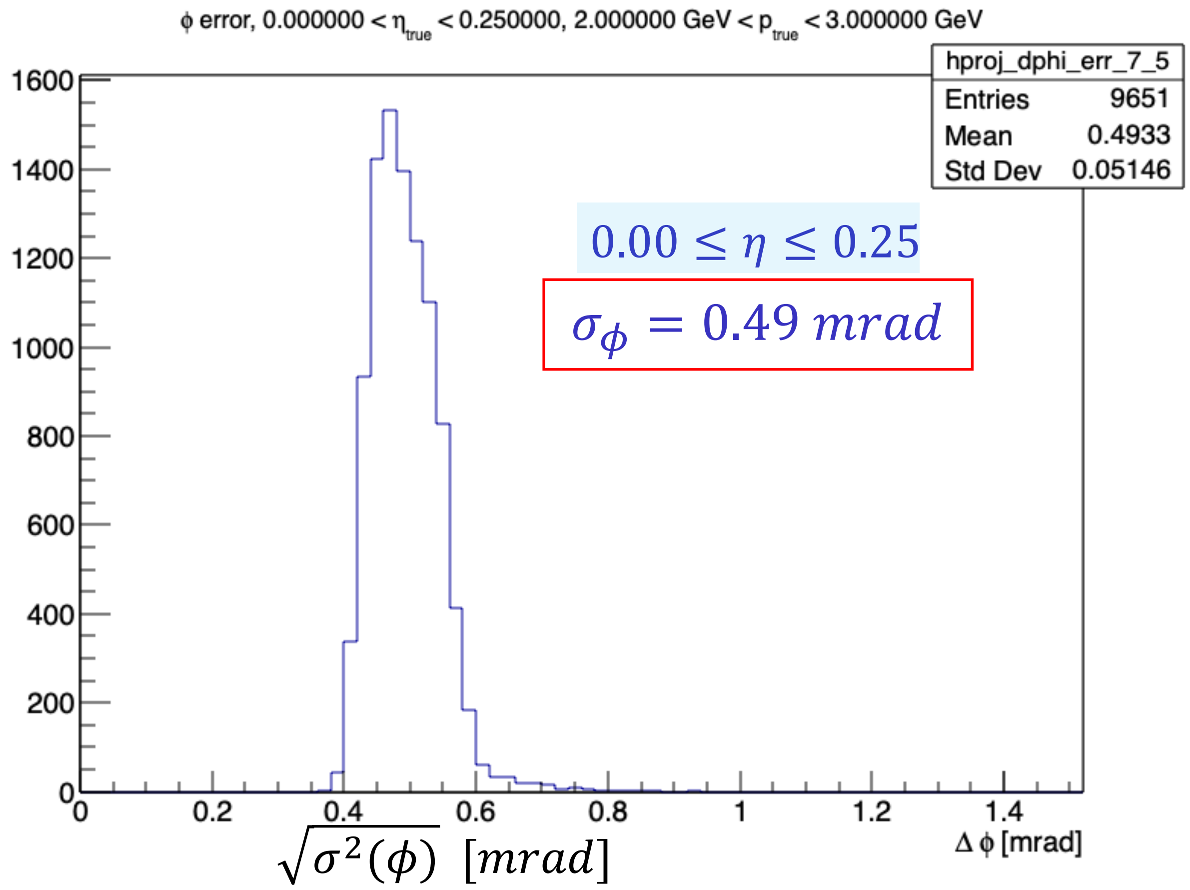


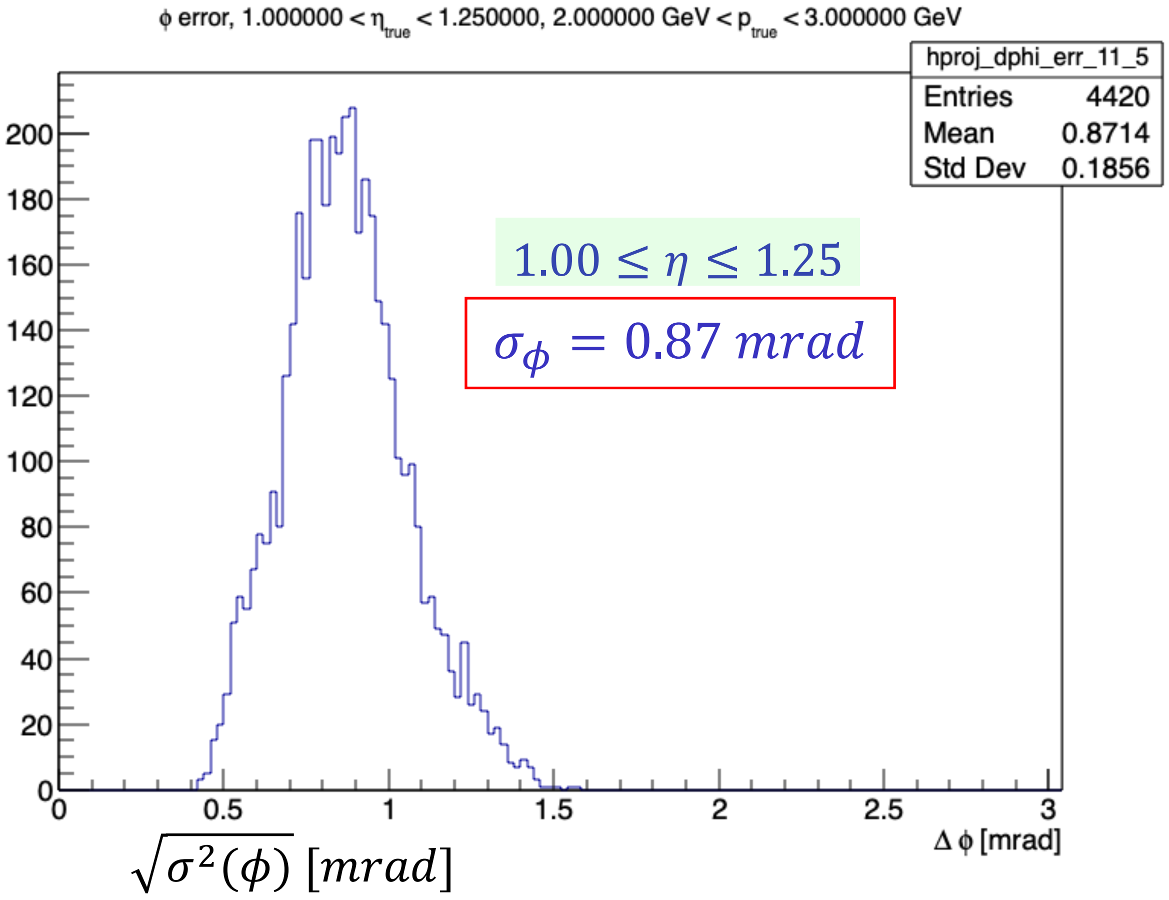
$\sqrt{\sigma^2(\theta)}\ [mrad]$

Method 2: Extracting ϕ Angular Resolution

- Histogram sqrt(variance), variance obtained from covariance matrix
  - Histogram mean = angular uncertainty
  - Histogram RMS = error bar

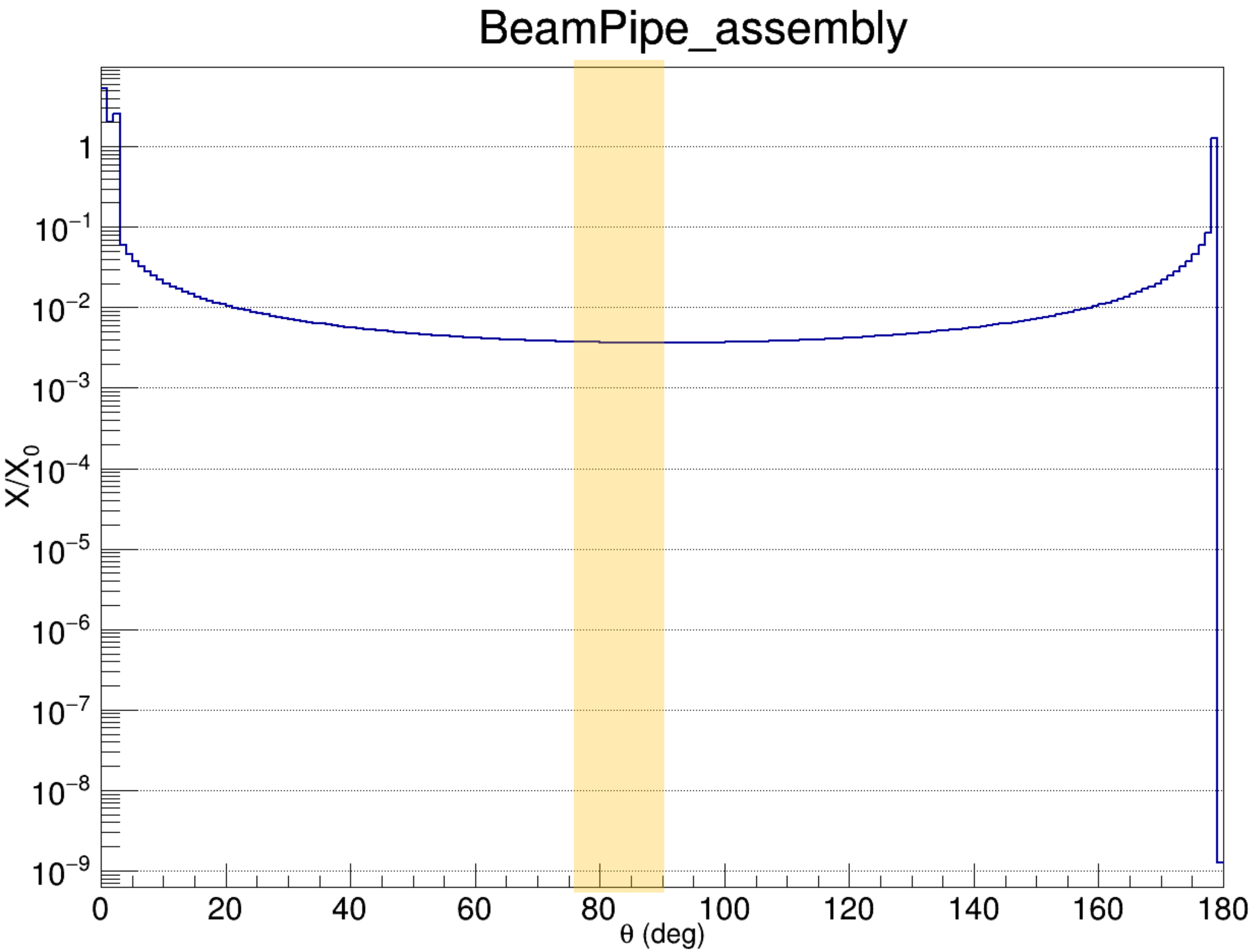
$2.00\ GeV \leq p \leq 3.00\ GeV$





Validating Method 2: Multiple Scatter from Beam Pipe

- Use effective X/X0 in multiple scatter calculation for comparison to $0.00 \leq \eta \leq 0.25$ simulation bin



PDG

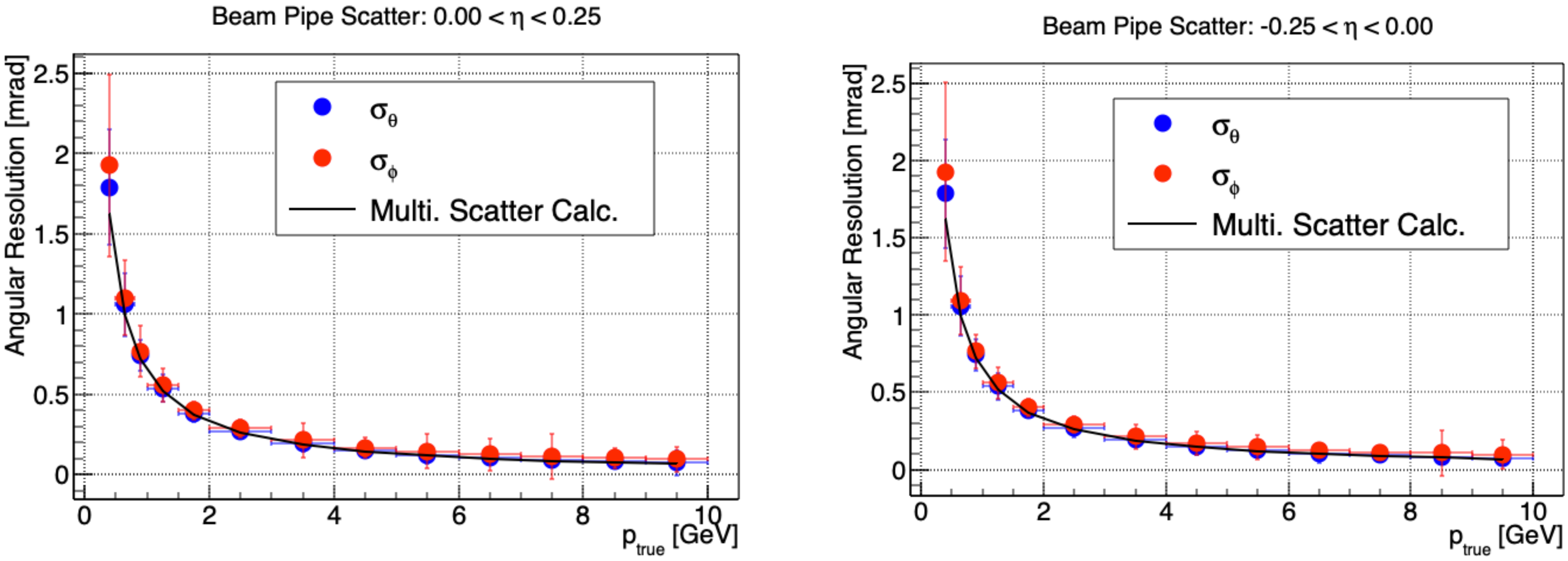
$$\theta_0 = \theta_{\rm plane}^{\rm rms} = \frac{1}{\sqrt{2}}\theta_{\rm space}^{\rm rms} \ ,$$

$$\theta_0 = \frac{13.6\ {\rm MeV}}{\beta c p} z \sqrt{\frac{x}{X_0}} \Big[1 + 0.088 \log_{10}(\frac{x z^2}{X_0 \beta^2})\Big]$$
$$= \frac{13.6\ {\rm MeV}}{\beta c p} z \sqrt{\frac{x}{X_0}} \Big[1 + 0.038 \ln(\frac{x z^2}{X_0 \beta^2})\Big]$$

- $z = c = \beta = 1$
- $76^o \leq \theta \leq 90^o$
  - **Avg: X/X0 = 0.003675**

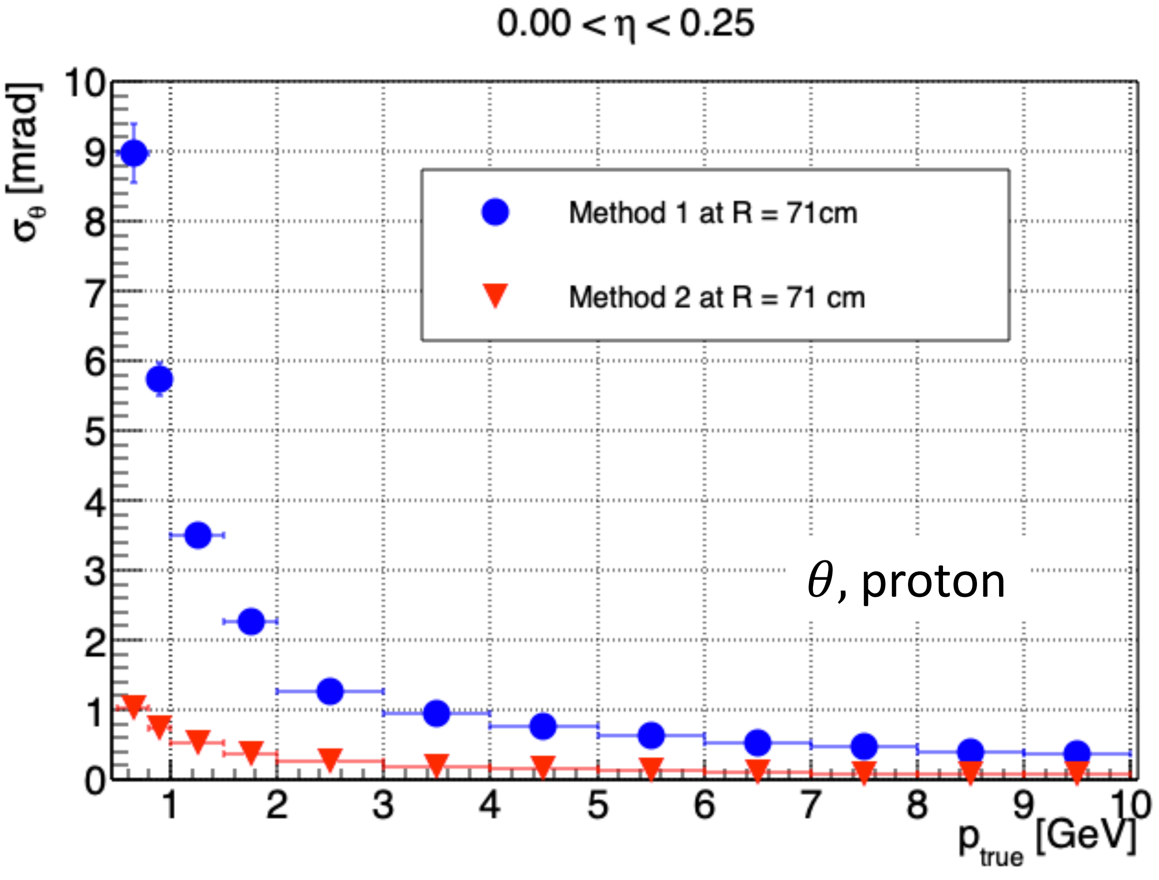
Validating Method 2: Multiple Scatter from Beam Pipe

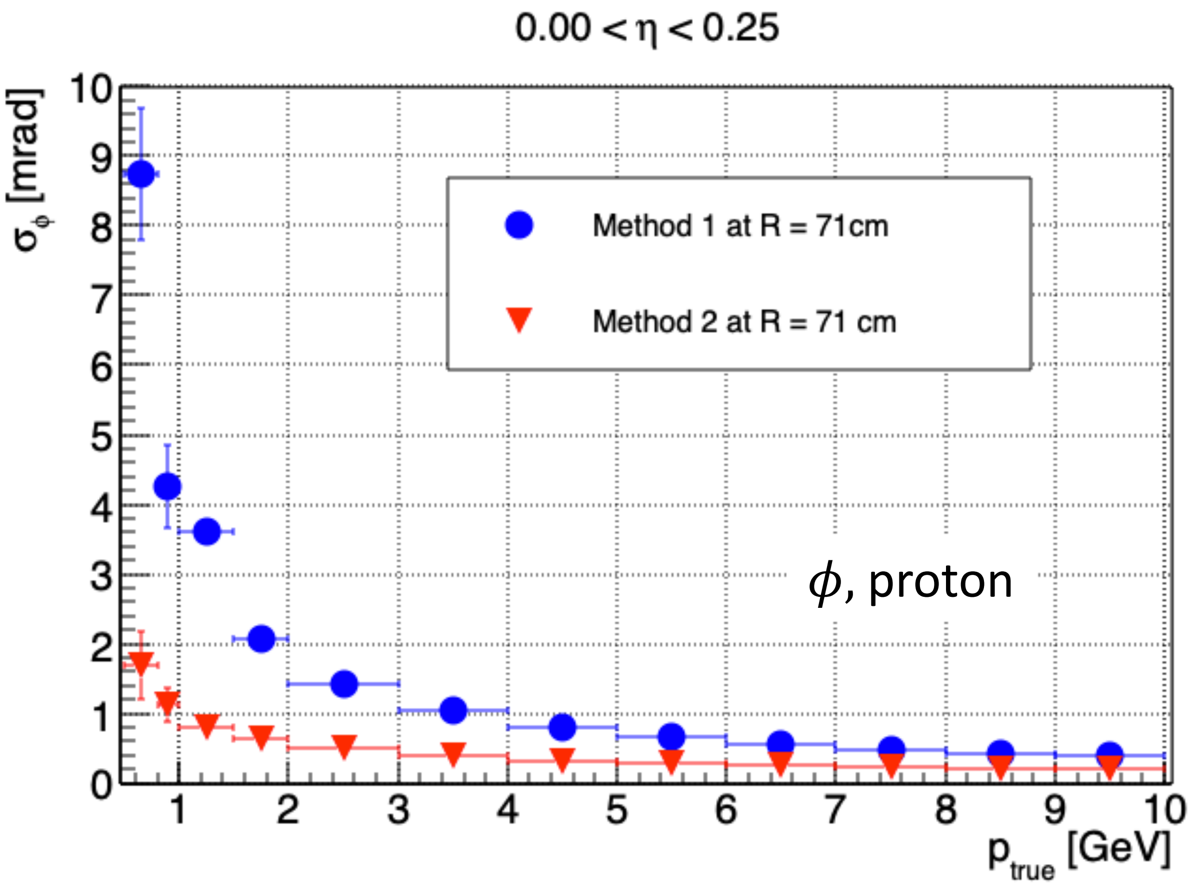
- Place a projection surface in between beam pipe and first Si vertex layer
  - Measure multiple scattering through beam pipe in simulation using Method 2
- **Good agreement between Method 2 and hand calculation**



Method 1 vs. Method 2

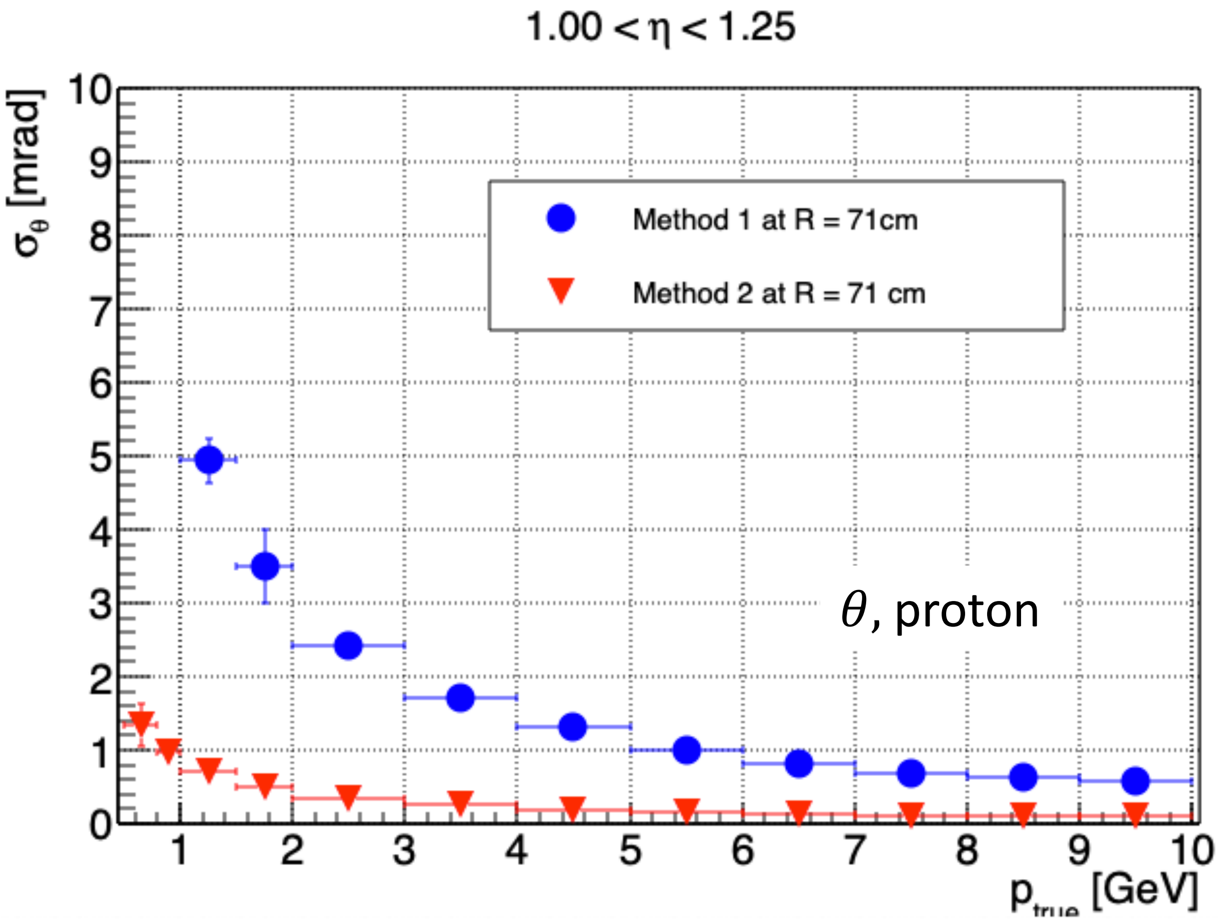
- **Clear angular resolution difference between Method 1 and Method 2**

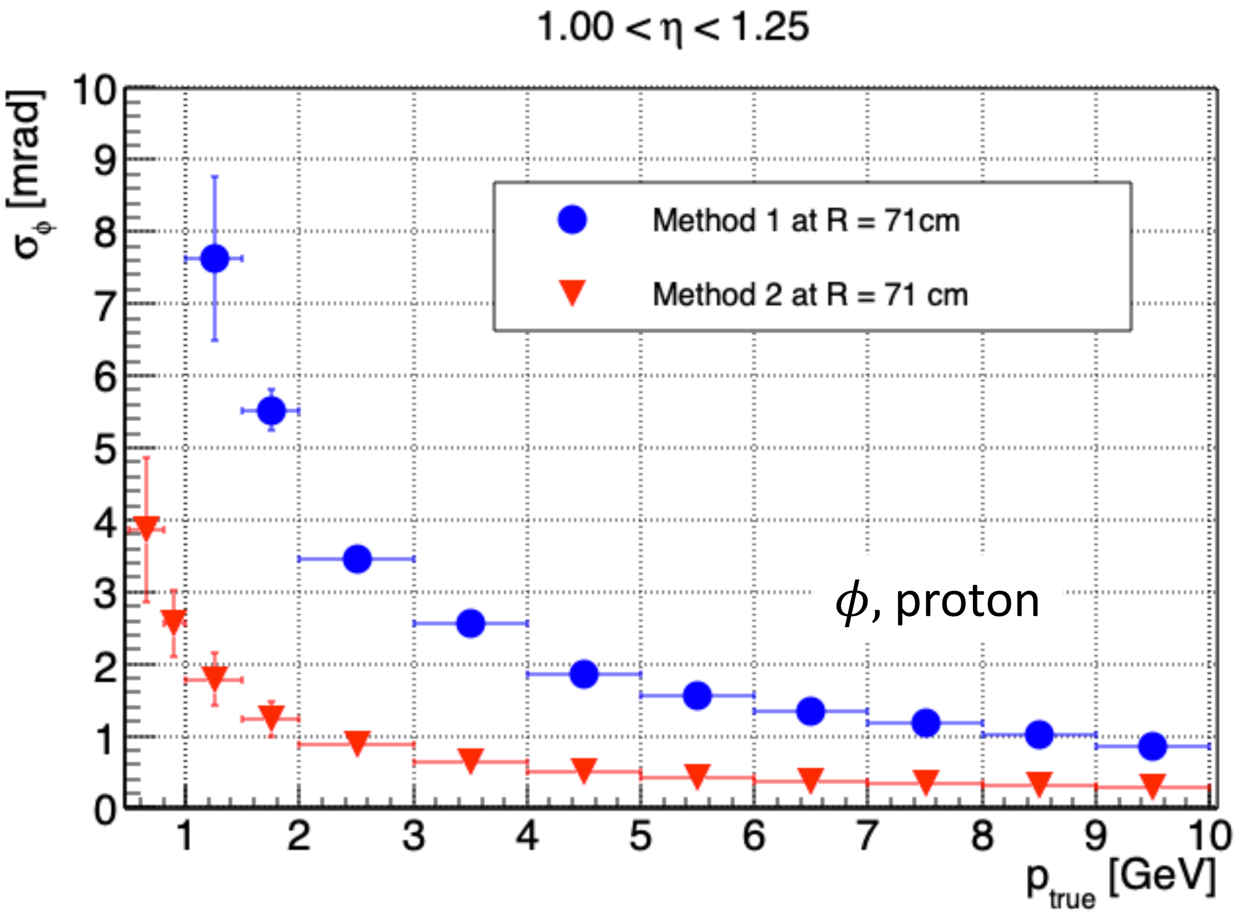




Method 1 vs. Method 2

- Clear angular resolution difference between Method 1 and Method 2

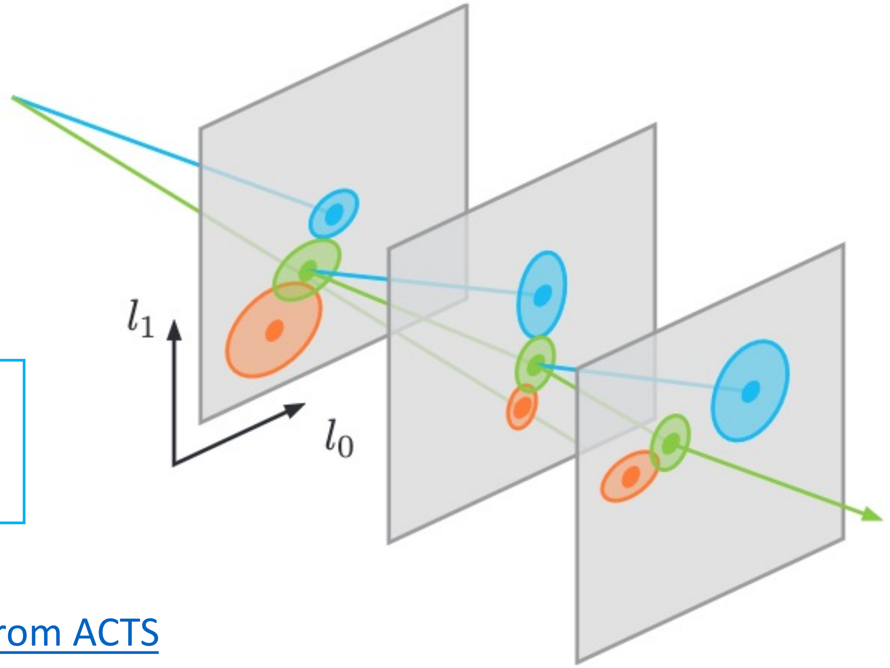




Difference Between Methods

- Propagation of trajectories begin at the vertex ($\vec{0}$, for this study) and propagate outward
- Method 1 takes difference between propagated trajectory track point and the true hit (via DIRC hit) to extract angular resolution
- Method 2 assigns uncertainty at each surface from Kalman Filter
  - Gives uncertainty related to KF (filtering uncertainty)
  - Doesn’t know where true hit location is
- ➢ Method 1 closer to true angular resolutions

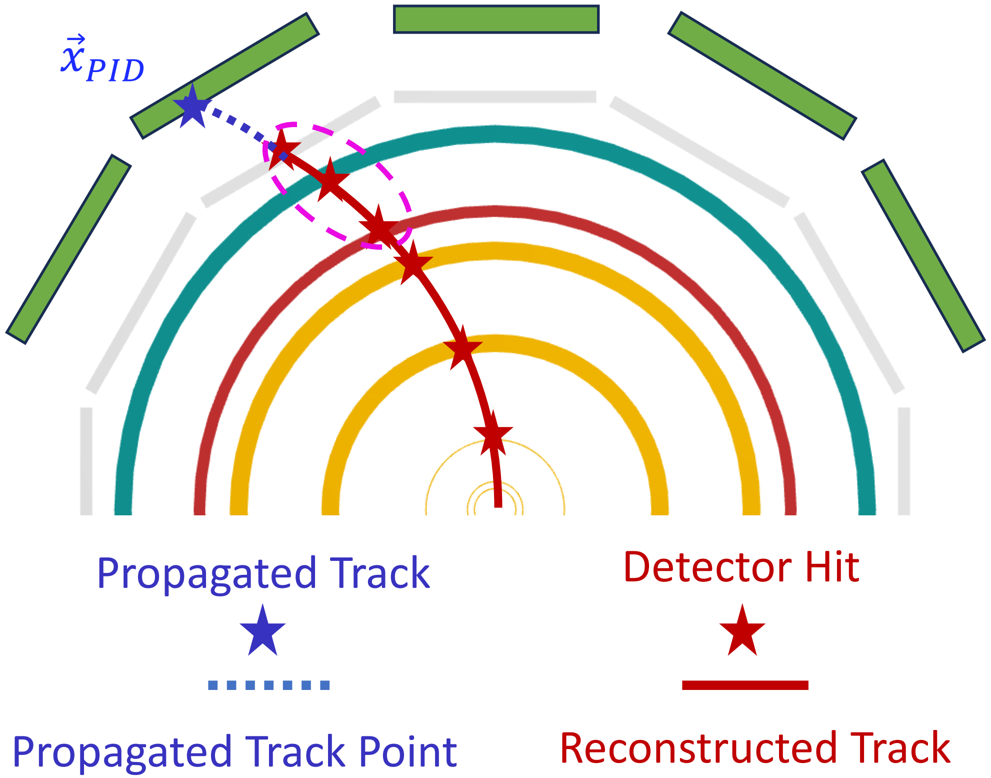
Illustration of the Kalman Filter. Two of the three stages, the prediction and the filtering are shown. The filtering updates the prediction with information from the measurement.



From ACTS

Other Methods

- Other assessments of the angular resolutions involve propagations not starting at vertex
  1. Form tracklet using the outer layers (MPGD+ToF), then propagate it to the projected surface
     - Requires looking for seeds/fitting in the outer layers
     - Are there enough layers for tracklet fitting (requires 3 seeds)?
       - 2 layers in backward region, 3 in the barrel and forward regions.
       - BIC could provide an additional hit in the barrel region
  2. Propagate from outer track state
     - Requires modification to propagation algo to take track state rather than trajectory
- Other thoughts?



Summary

- Applying cuts in Method 1 can remove geometrical mismatches between DIRC and projected surface
- Cause of the difference between Method 1 and 2 is better understood
  - Method 2 provides uncertainty on KF, where as Method 1 compares the propagated track point to true particle location
  - Method 1 provides the more realistic angular resolutions
- Validation and cross checks for Method 2
  - Resolutions due to only the beam pipe agree with multiple scattering calculations
  - In Progress: Cross check with Fast Simulations being produced by Shyam
- Development work will be needed if we want to move away from trajectories which begin propagation at the vertex