Gaseous Simulation Tasks

- ePIC tracker has 4 MPGD subsystems: Backward disks, Forward disks, inner and outer barrel layers
- General Gaseous Simulation Tasks:
  1. Implement more realistic geometries/services
  2. Digitize hits (convert hits into charge and timing info)
  3. Use digitized hits to form reconstructed hits (e.g. clustering)
  4. Use ACTS to fit tractlets from hits in MPGD and ToF layers

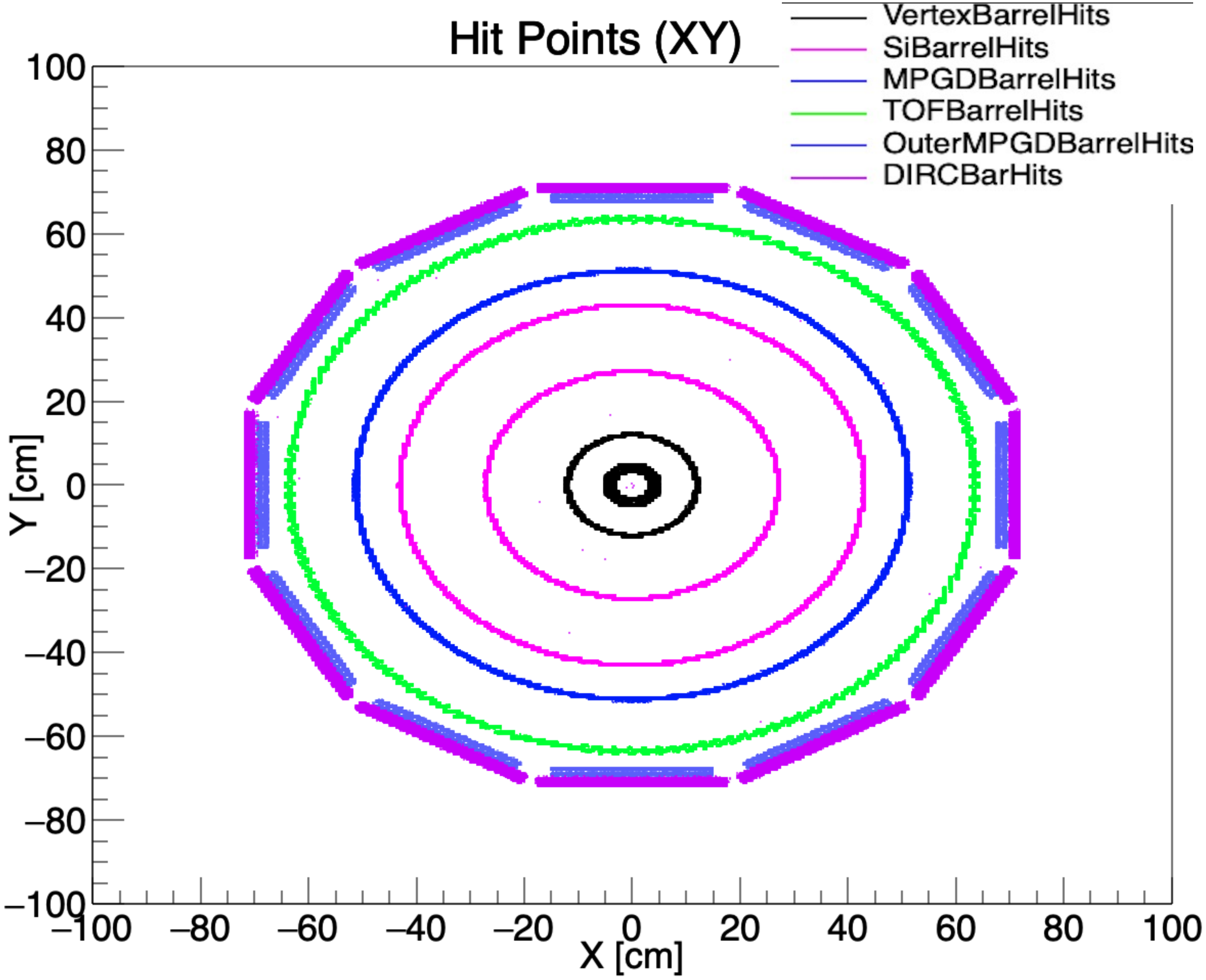
Please reach out if you have an interest in helping

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) = $(20^o, 160^o)$ /

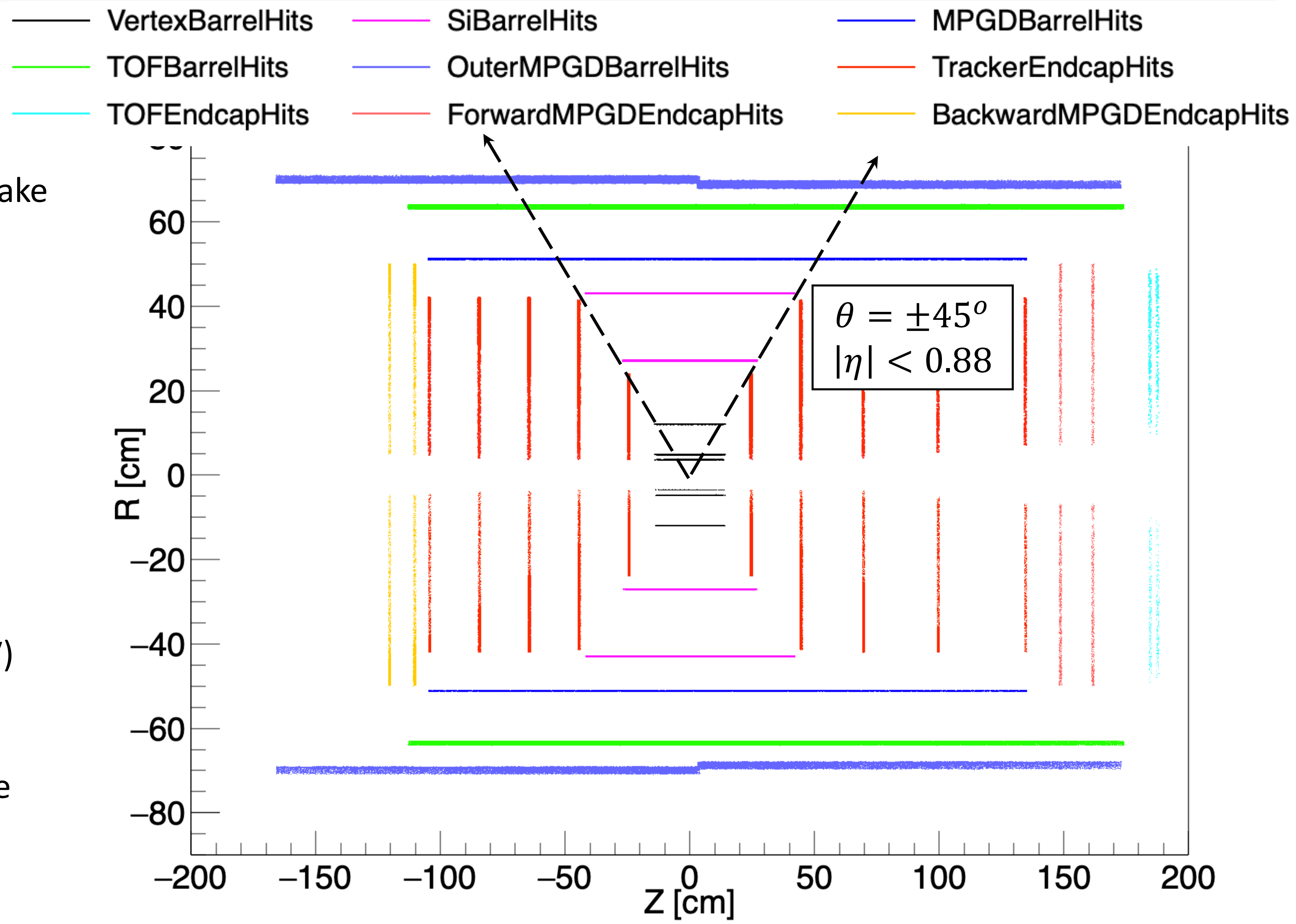
    $(|\eta| \leq 1.73)$
  - p (uniform) = (0.3 GeV, 10.0 GeV)
- hpDIRC Mods
  - Make DIRC bars sensitive volume

    (provides DIRC hit)
  - Turn off optical photons

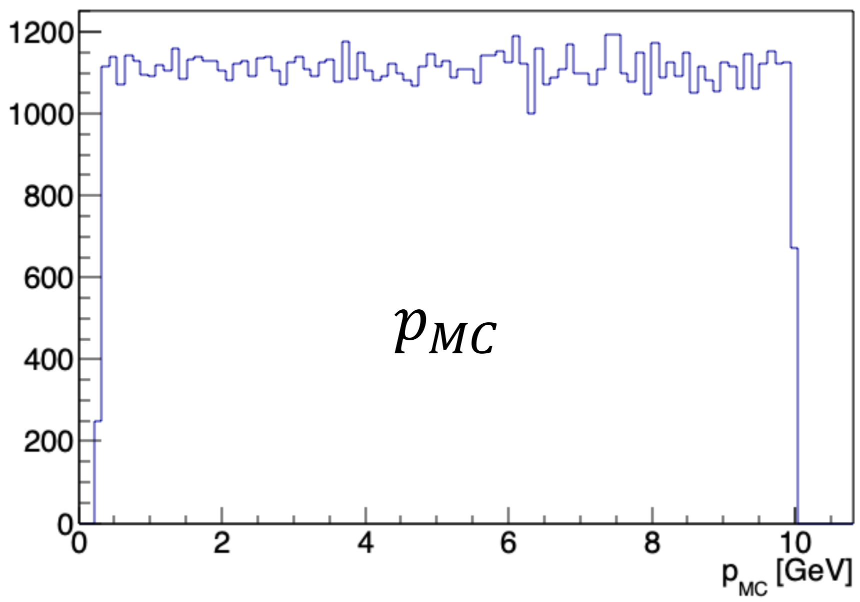


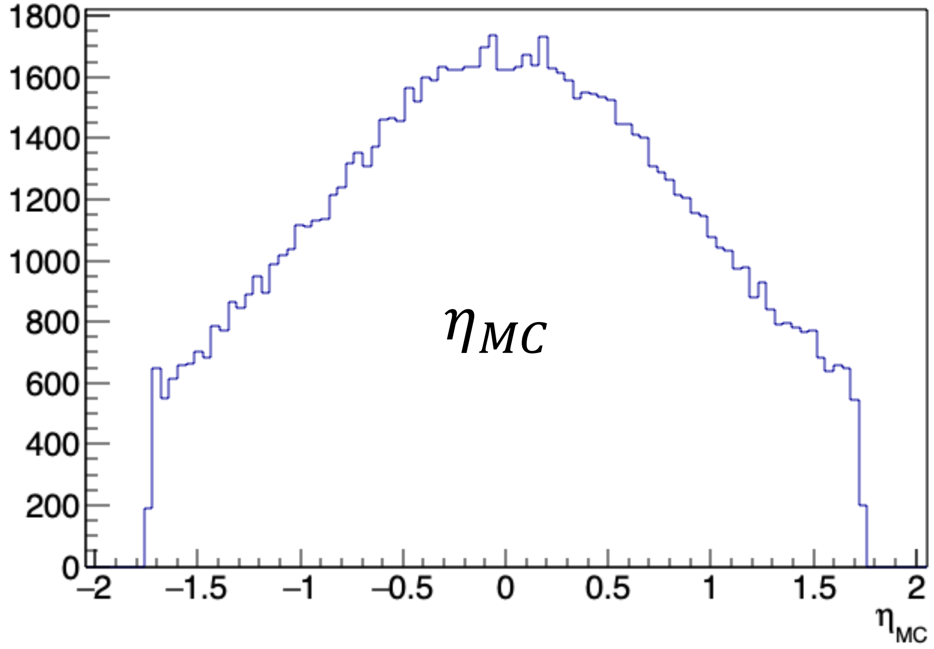
Simulation Details

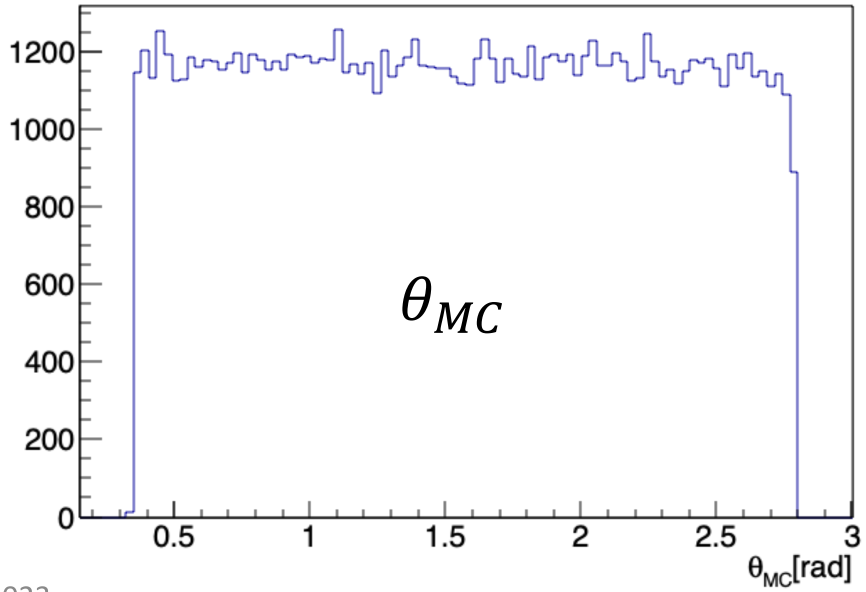
- Software Version
  - ePIC = 23.07.2
  - Detector Configuration = Craterlake
  - EICRecon = v1.5.1
- Generator
  - Particle Gun = proton
  - ϕ (uniform) = $(0^o, 360^o)$
  - θ (uniform) = $(20^o, 160^o)$ / $(|\eta| \leq 1.73)$
  - p (uniform) = (0.3 GeV, 10.0 GeV)
- hpDIRC Mods
  - Make DIRC bars sensitive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

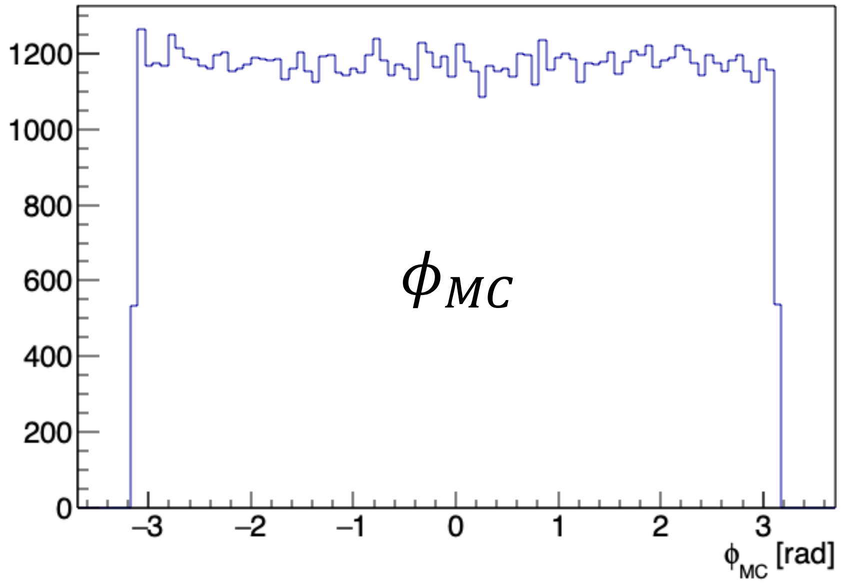


Simulation Distributions: Representative Sample

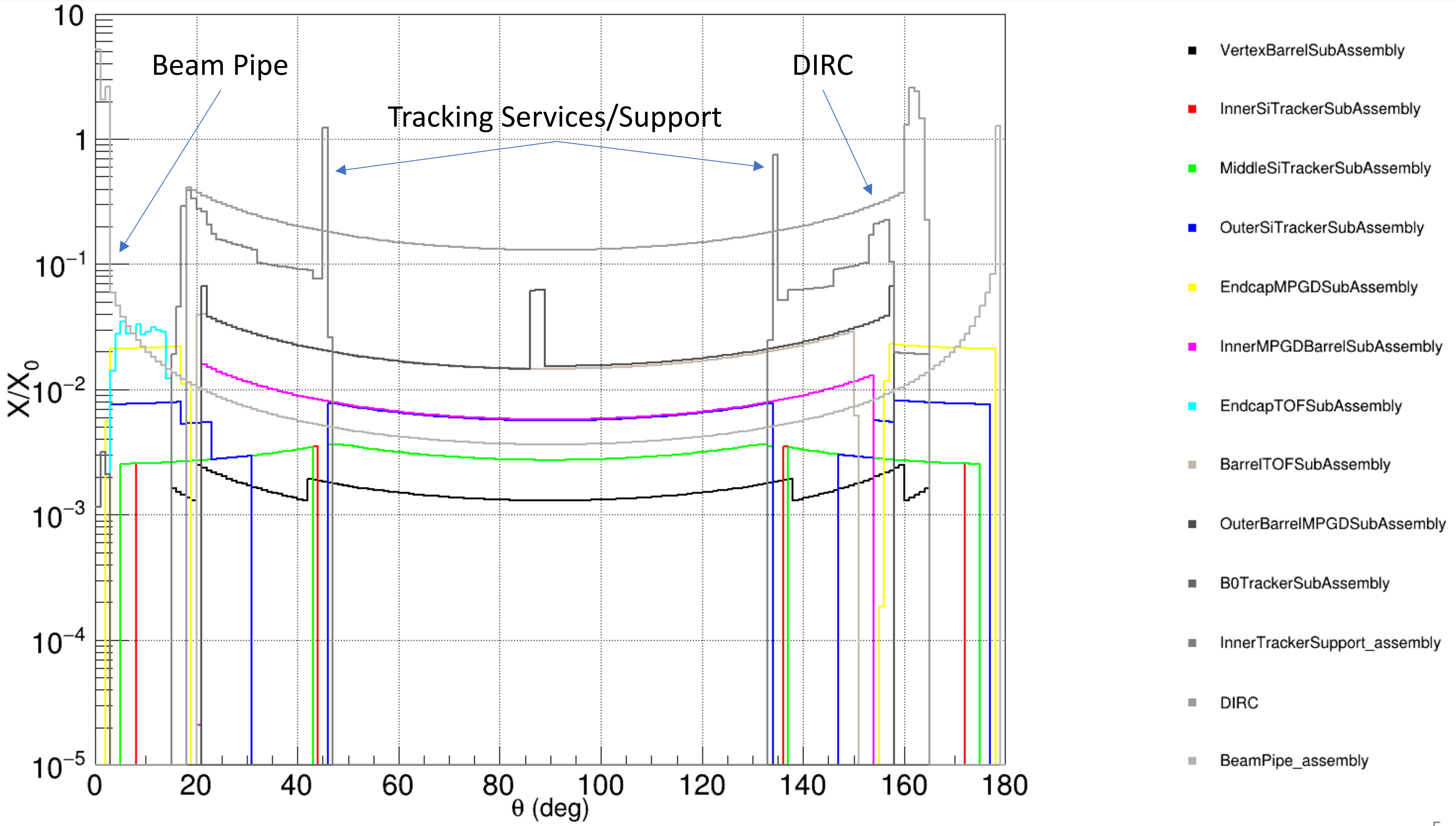






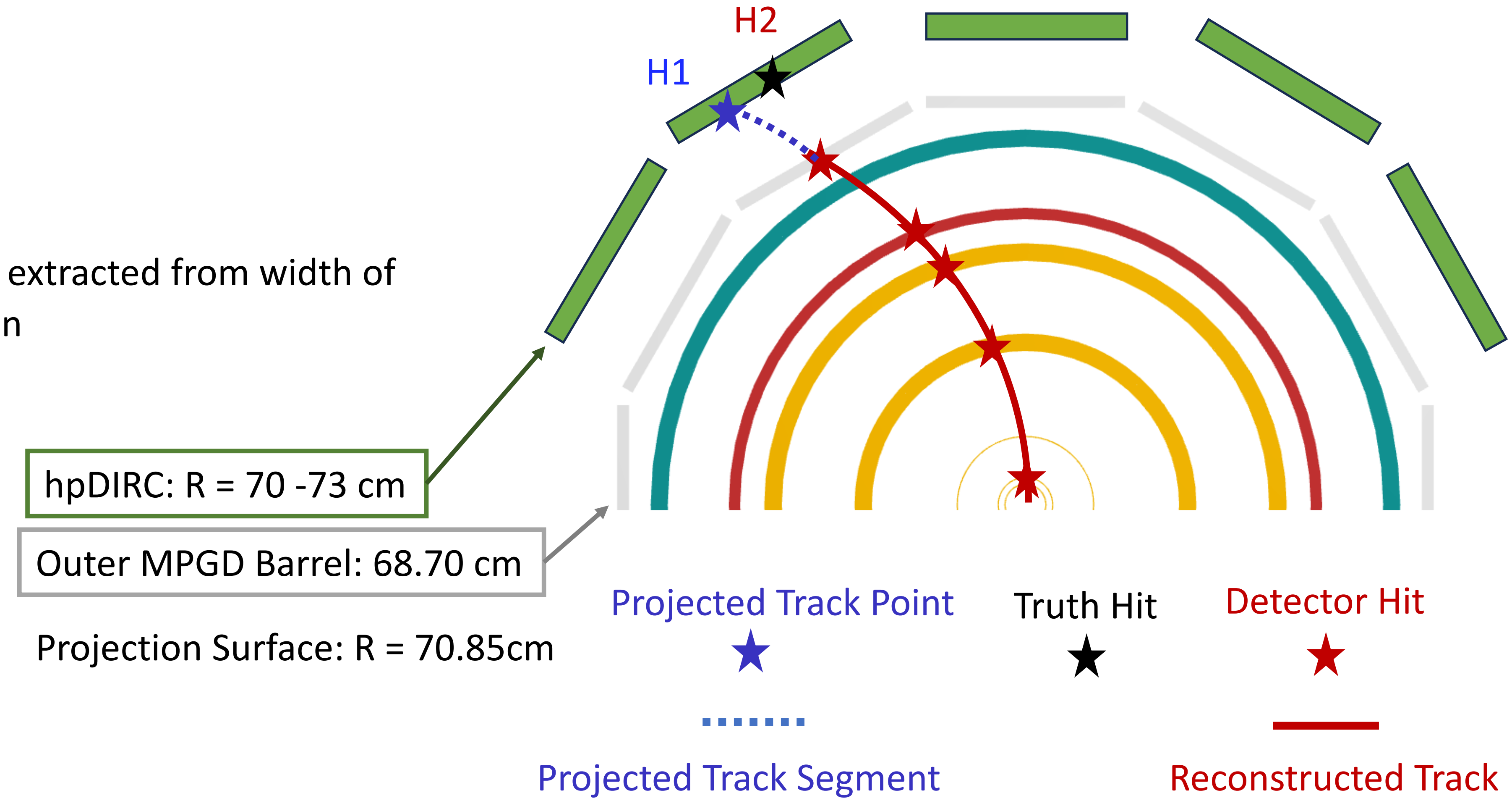


Material Budget



Angular Resolution Method 1

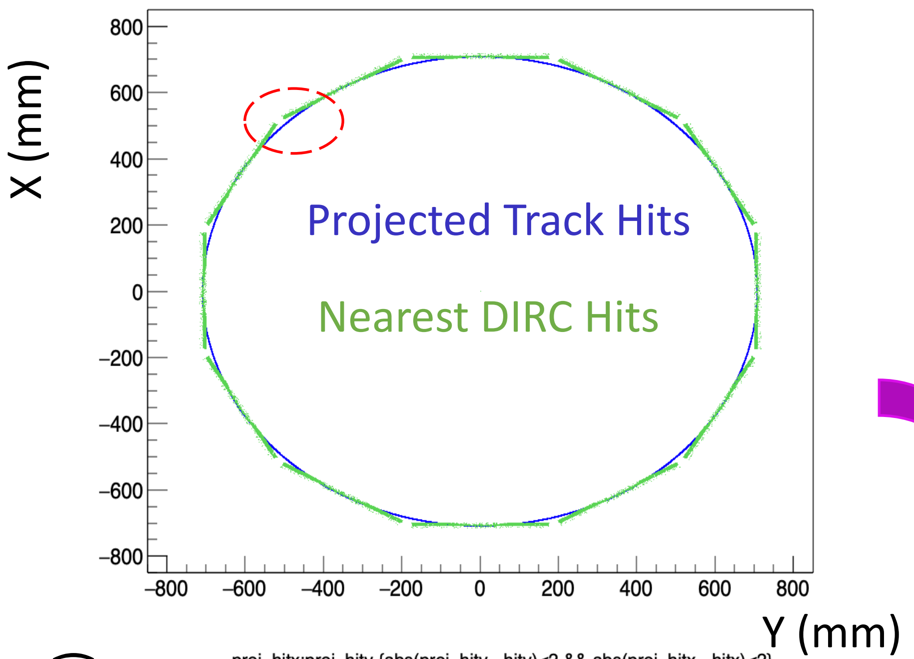
- Use projected position point vectors of projected track point (H1) and **nearest** DIRC hit (H2) to obtain angles:
  - Projected Point (x,y,z) hits → $\theta_{proj}, \phi_{proj}$
  - DIRC Point (x,y,z) hits → $\theta_{dirc}, \phi_{dirc}$
- Angular differences are:
  - $\theta_{proj} - \theta_{dirc}$
  - $\phi_{proj} - \phi_{dirc}$
- Angular resolution $\sigma_\theta, \sigma_\phi$ are extracted from width of assumed Gaussian distribution

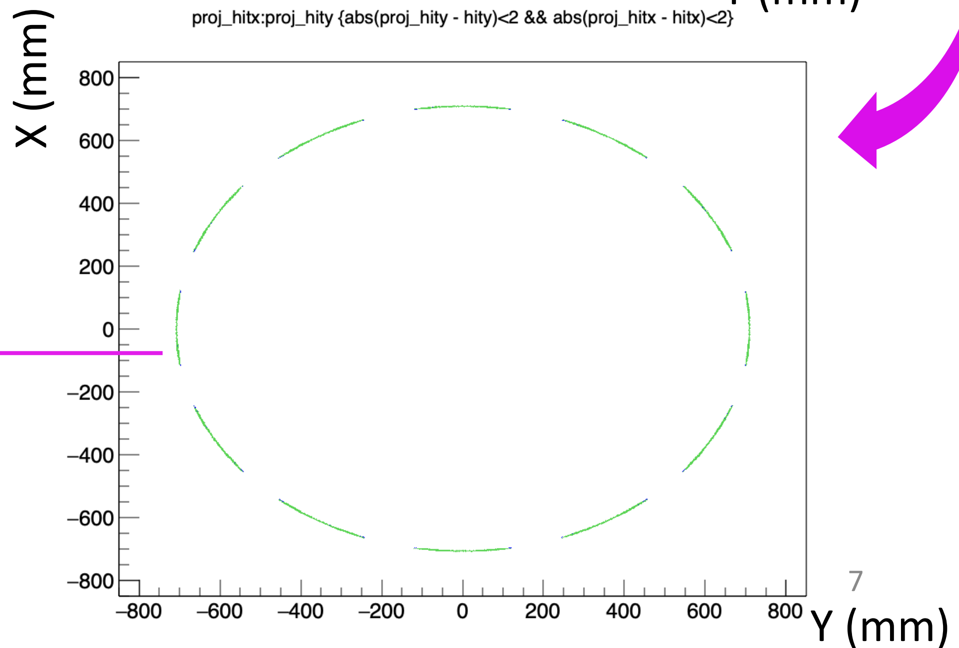


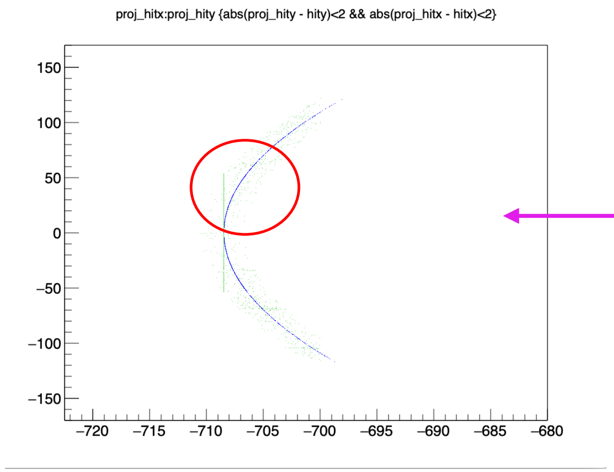
Acceptance Cut

- Implement a cut to include DIRC hits that are near the projected surface
  - $|(x_{proj} - x_{hit})| < 2\ mm$
  - $|(y_{proj} - y_{hit})| < 2\ mm$

- Cuts lead to improvement, but some DIRC hits still far from projected curved surface

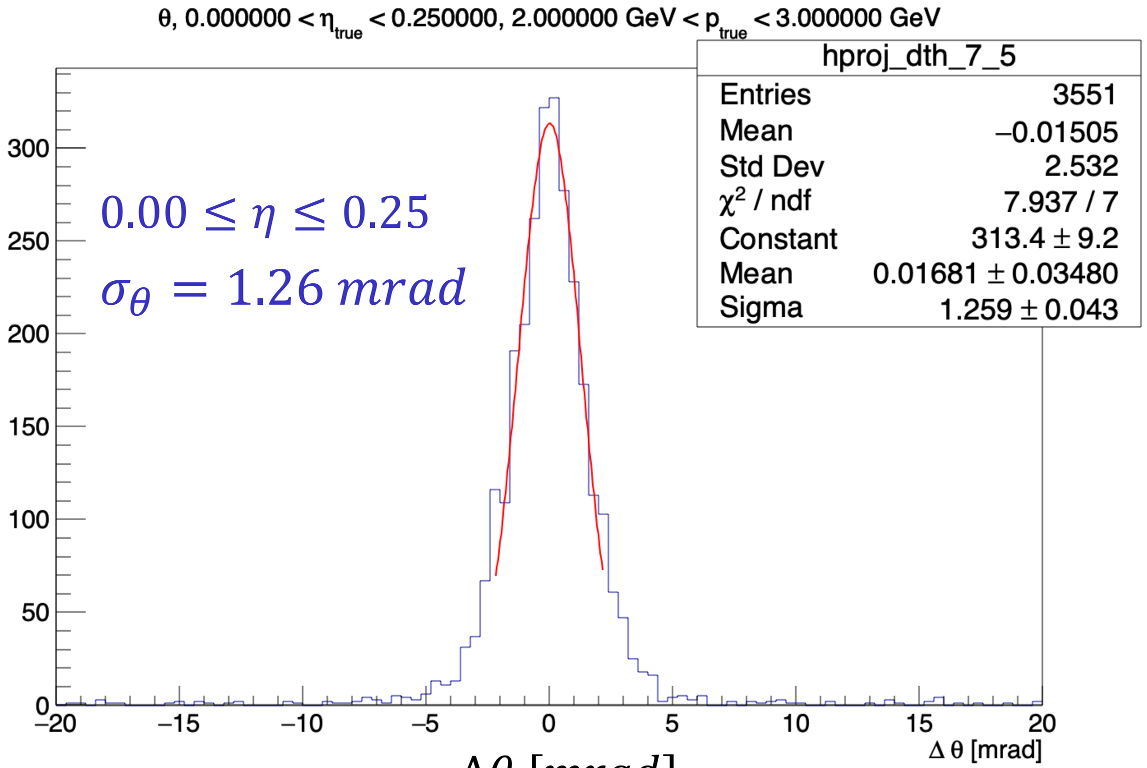




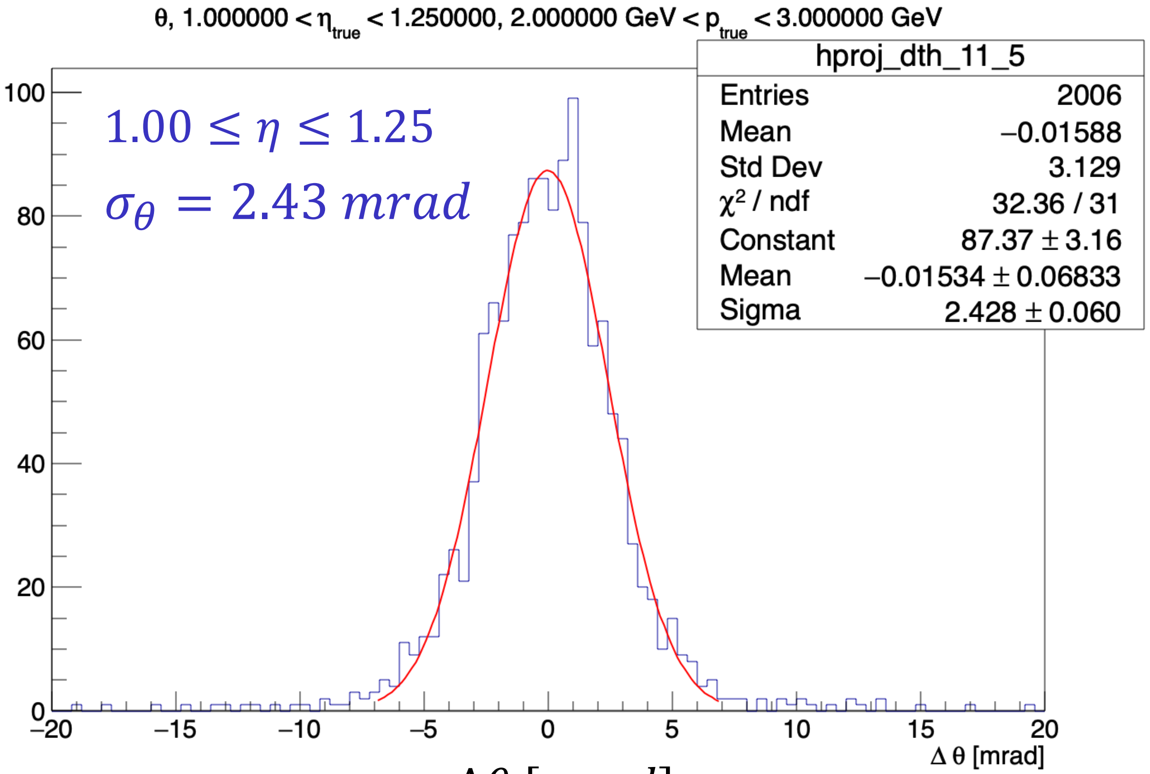


Method 1: Extracting θ Angular Resolution

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



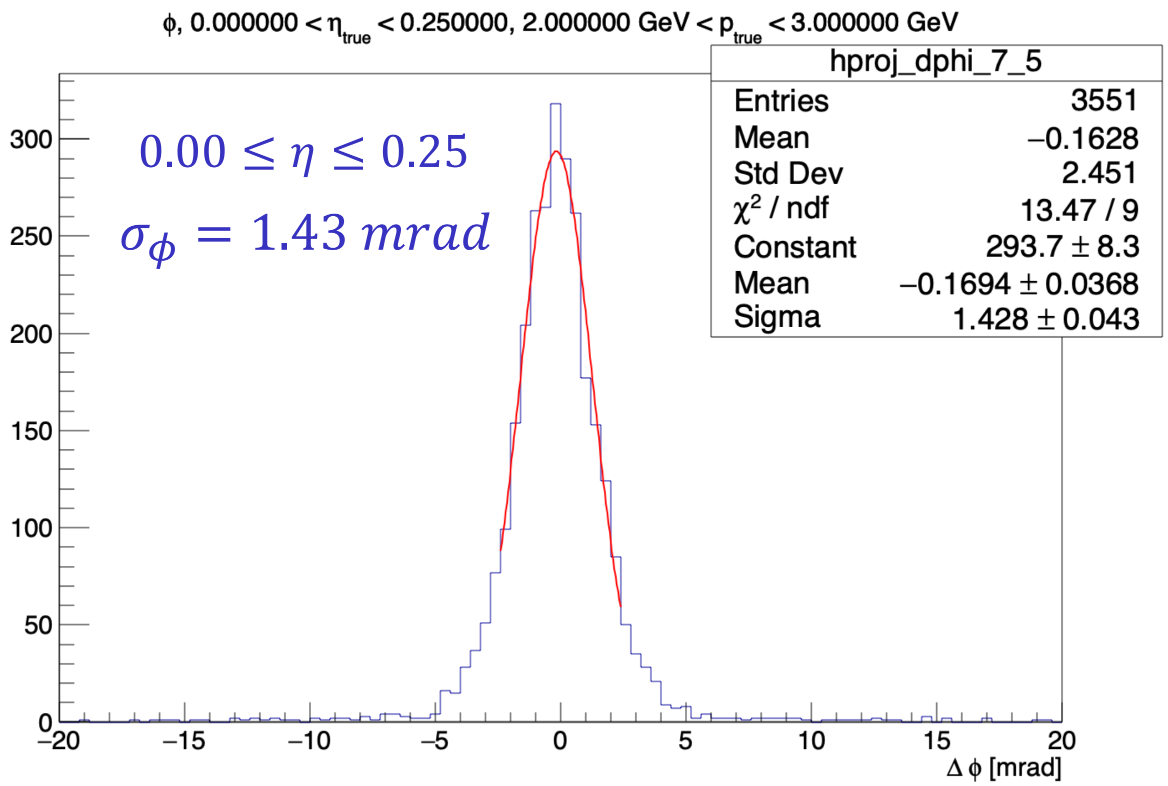
$\Delta\theta\ [mrad]$



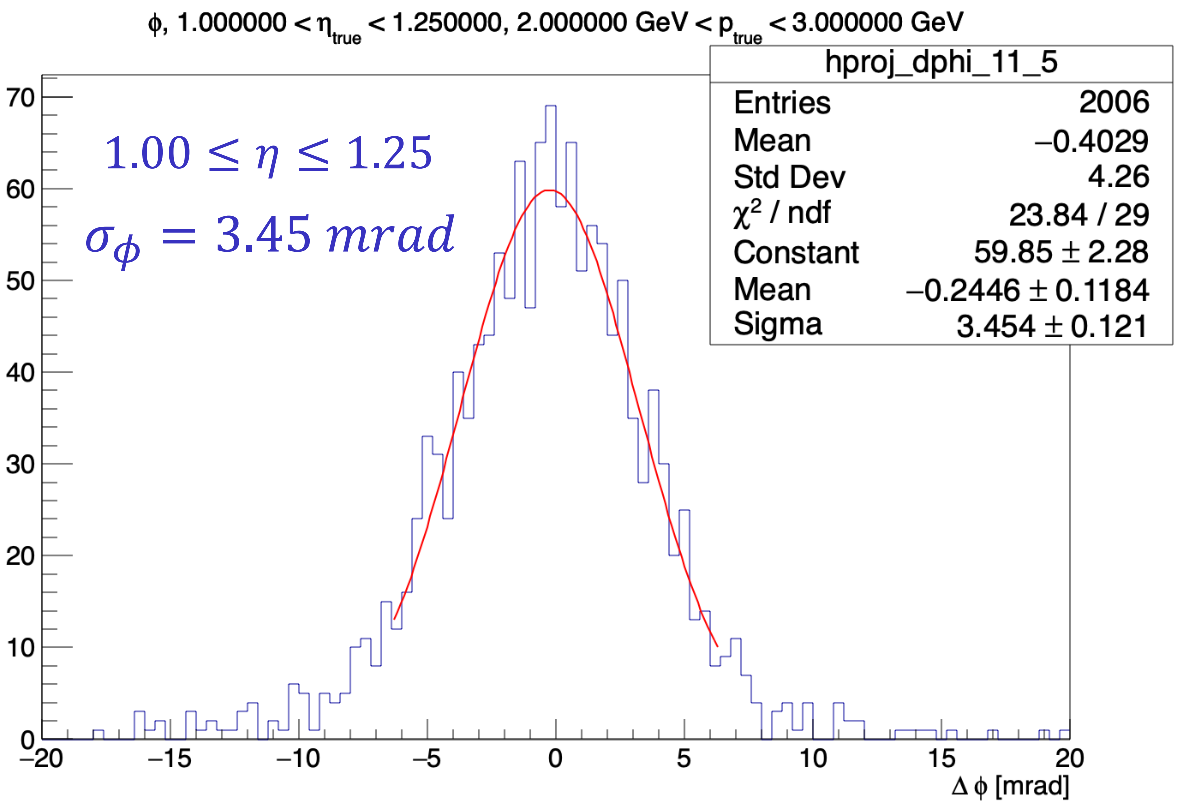
$\Delta\theta\ [mrad]$

Method 1: Extracting ϕ Angular Resolution

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



$\Delta\phi\ [mrad]$

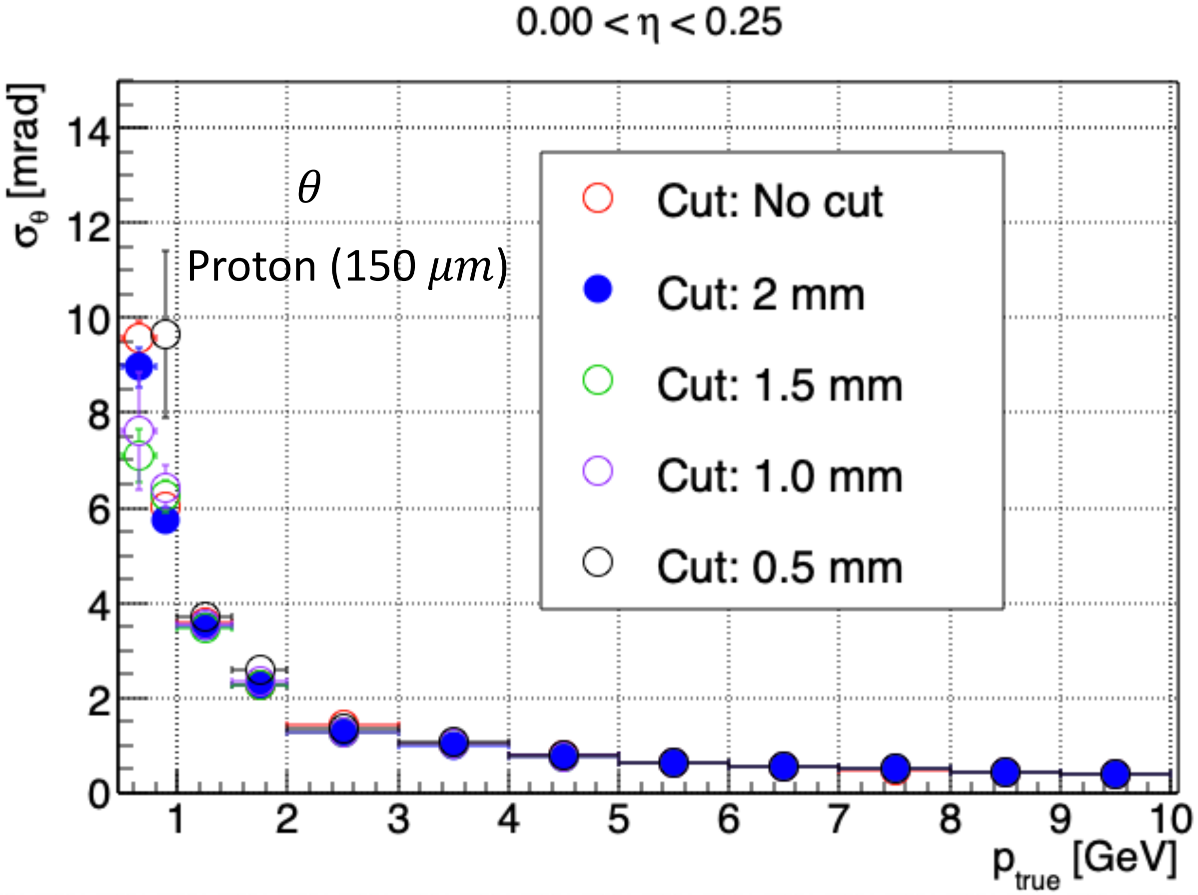


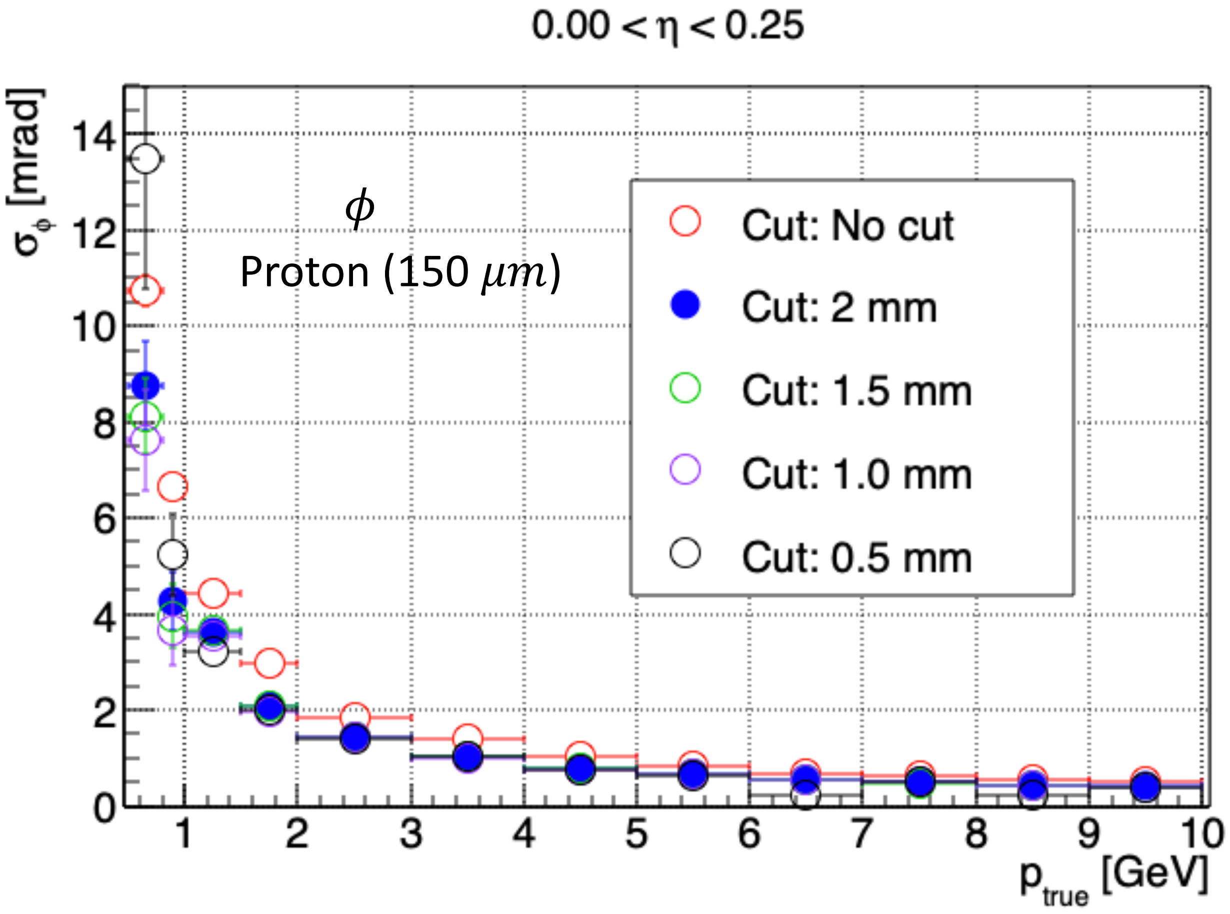
$\Delta\phi\ [mrad]$

Cut Sensitivity

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

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

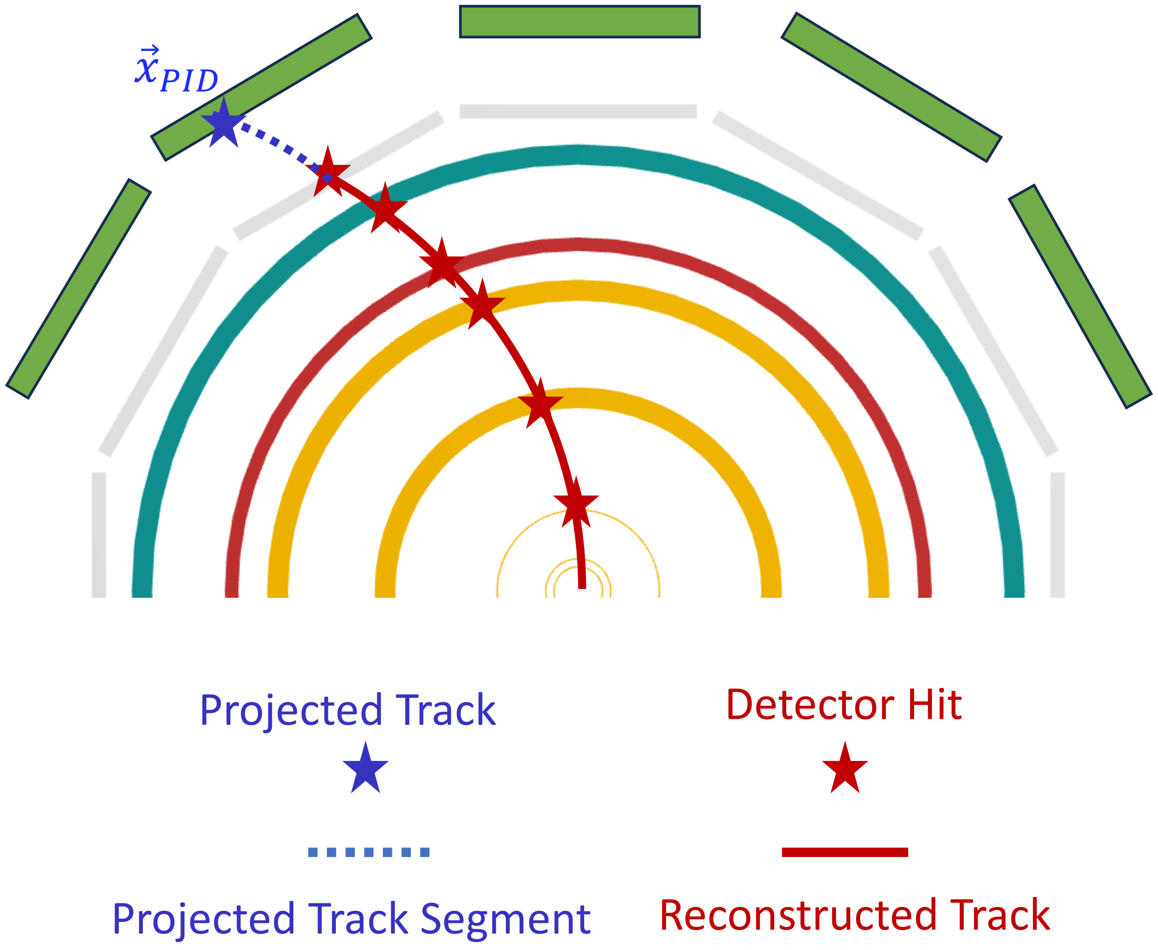




Method 2

- Track Errors

  - Use projected track state vector a 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
    - $var(\theta), var(\phi), cov(\theta, \phi)$



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;
    }
```

...

Projects track to specified ACTS surface

Assess track point information at specified ACTS surface

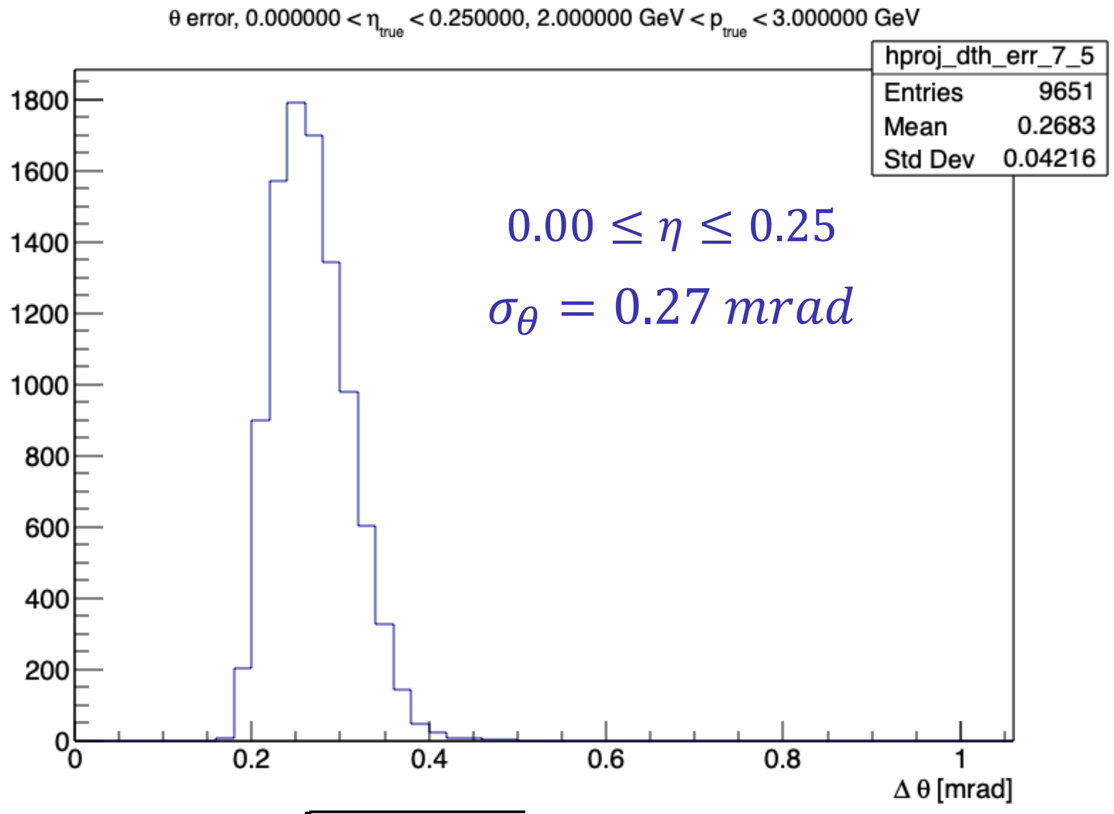
```
    std::unique_ptr<edm4eic::TrackPoint> projection_point;
    try {
        // >>> try to propagate to surface <<<
        projection_point = m_propagation_algo.propagate(trajectory, m_DIRC_center_surface);
    }
    catch(std::exception &e) {
        throw JException(e.what());
    }

    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;
```

Method 2: Extracting θ Angular Resolution

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

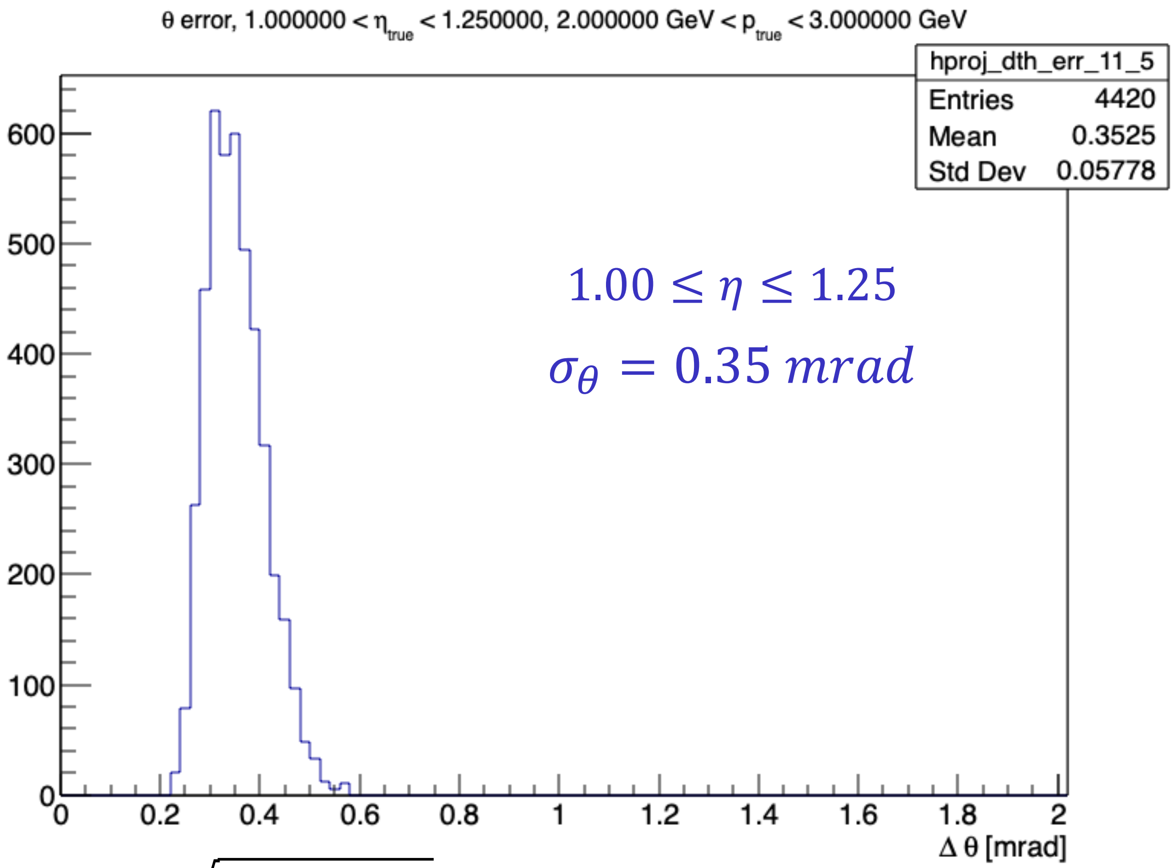
$$2.00\ GeV \leq p \leq 3.00\ GeV$$



$0.00 \leq \eta \leq 0.25$

$\sigma_\theta = 0.27\ mrad$

$\sqrt{var < \theta >}\ [mrad]$



$1.00 \leq \eta \leq 1.25$

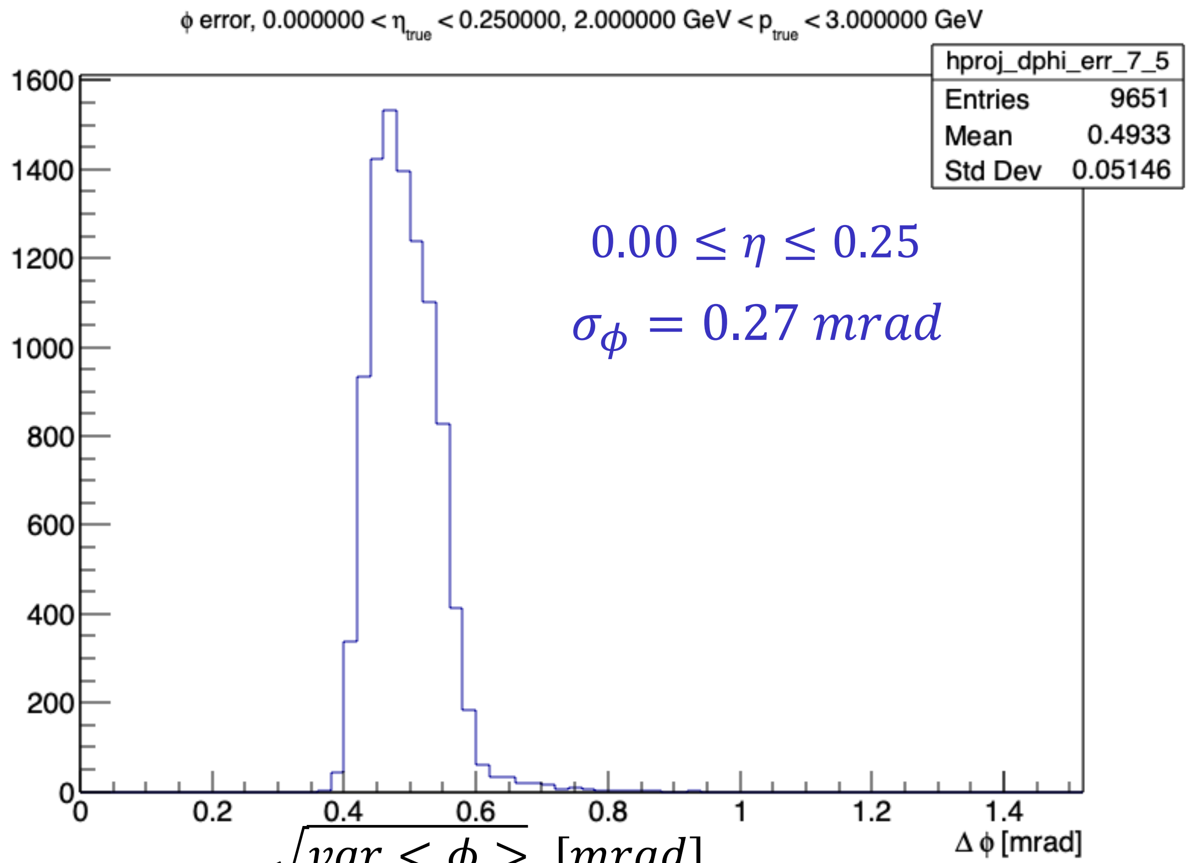
$\sigma_\theta = 0.35\ mrad$

$\sqrt{var < \theta >}\ [mrad]$

Method 2: Extracting ϕ Angular Resolution

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

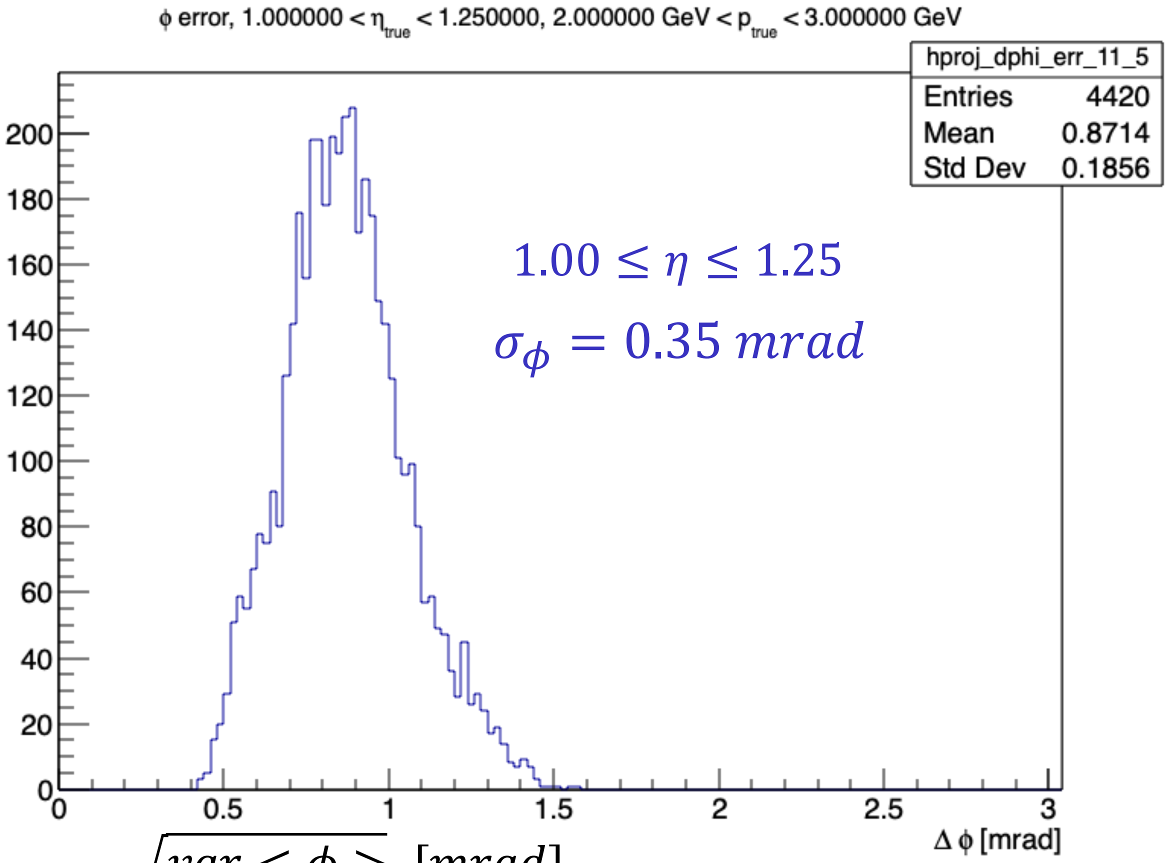
$2.00\ GeV \leq p \leq 3.00\ GeV$



$0.00 \leq \eta \leq 0.25$

$\sigma_\phi = 0.27\ mrad$

$\sqrt{var < \phi >}\ [mrad]$



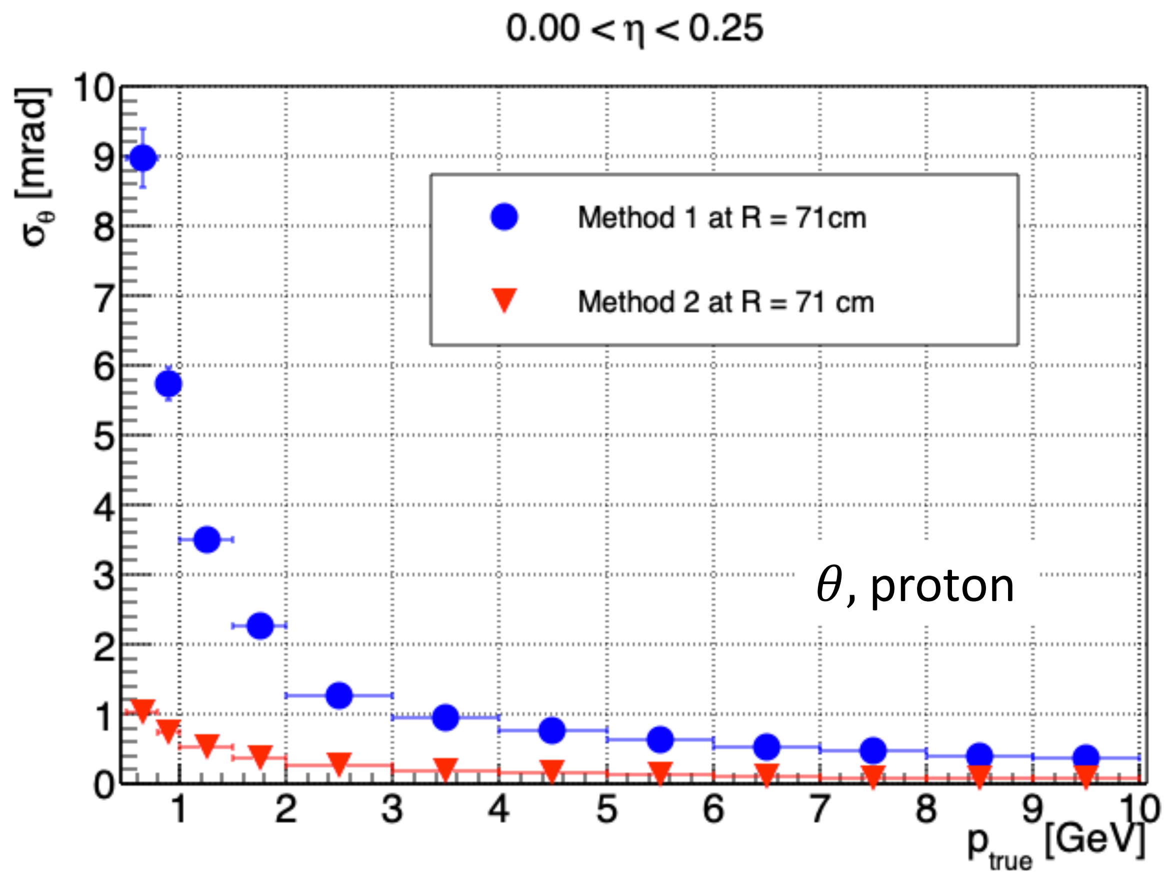
$1.00 \leq \eta \leq 1.25$

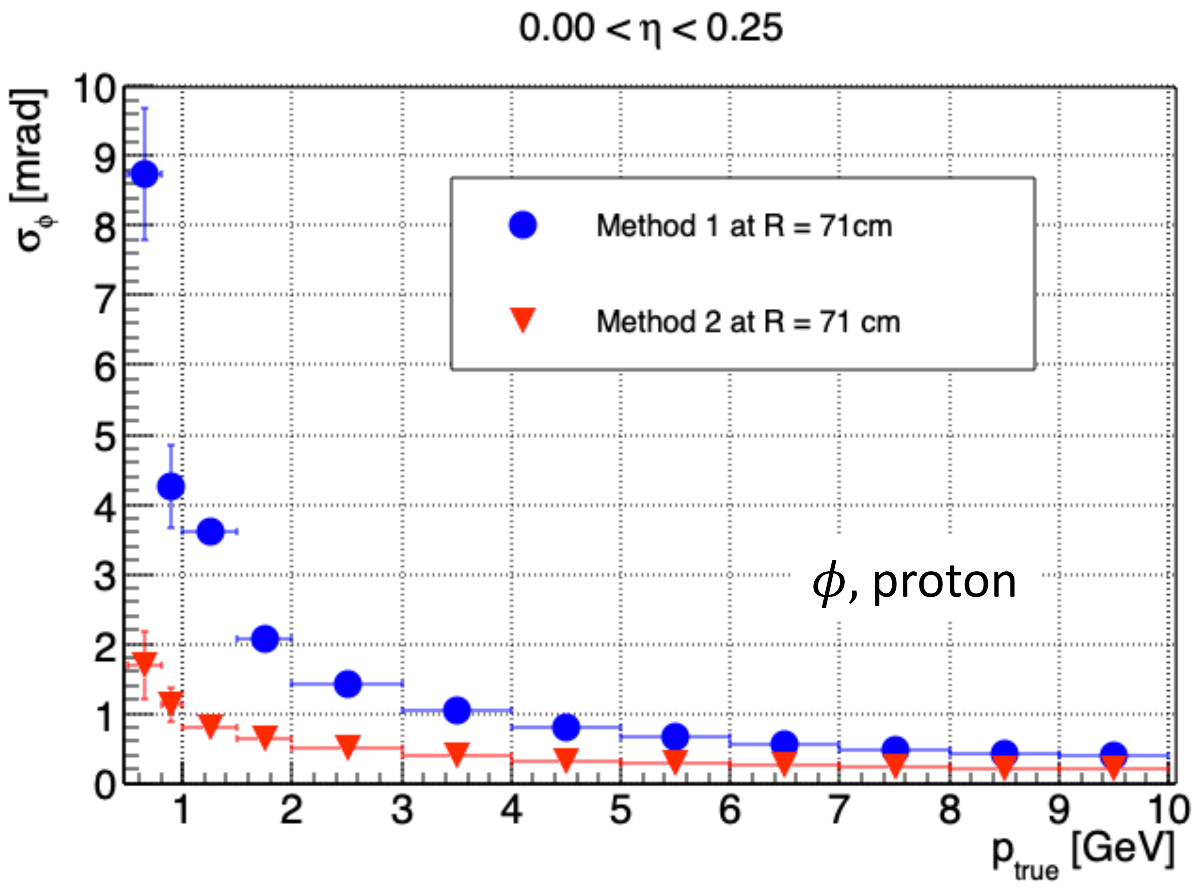
$\sigma_\phi = 0.35\ mrad$

$\sqrt{var < \phi >}\ [mrad]$

Method 1 vs. Method 2

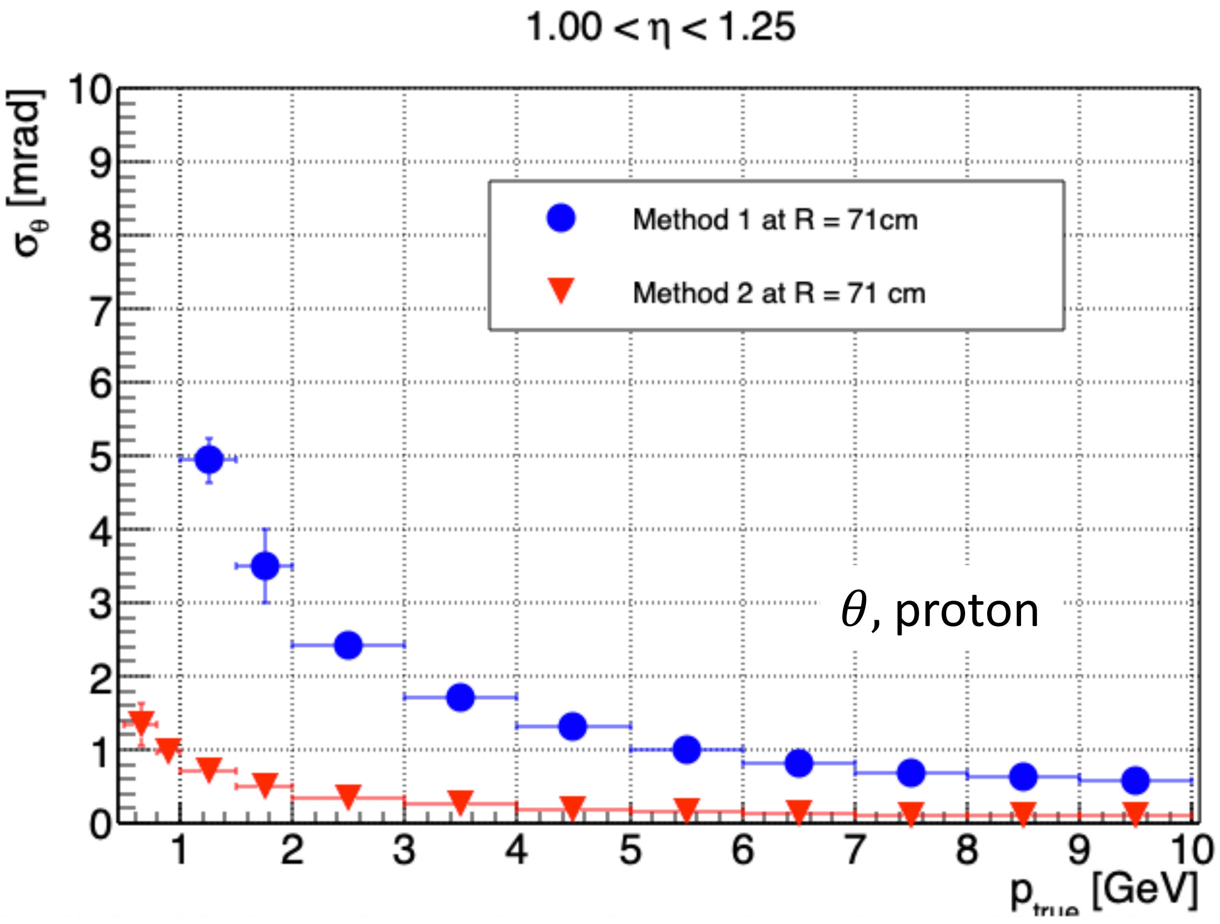
- ❑ Use difference between MC and Reconstructed tracks to obtain angular resolutions
- ➢ **Clear angular resolution difference between Method 1 and Method 2**

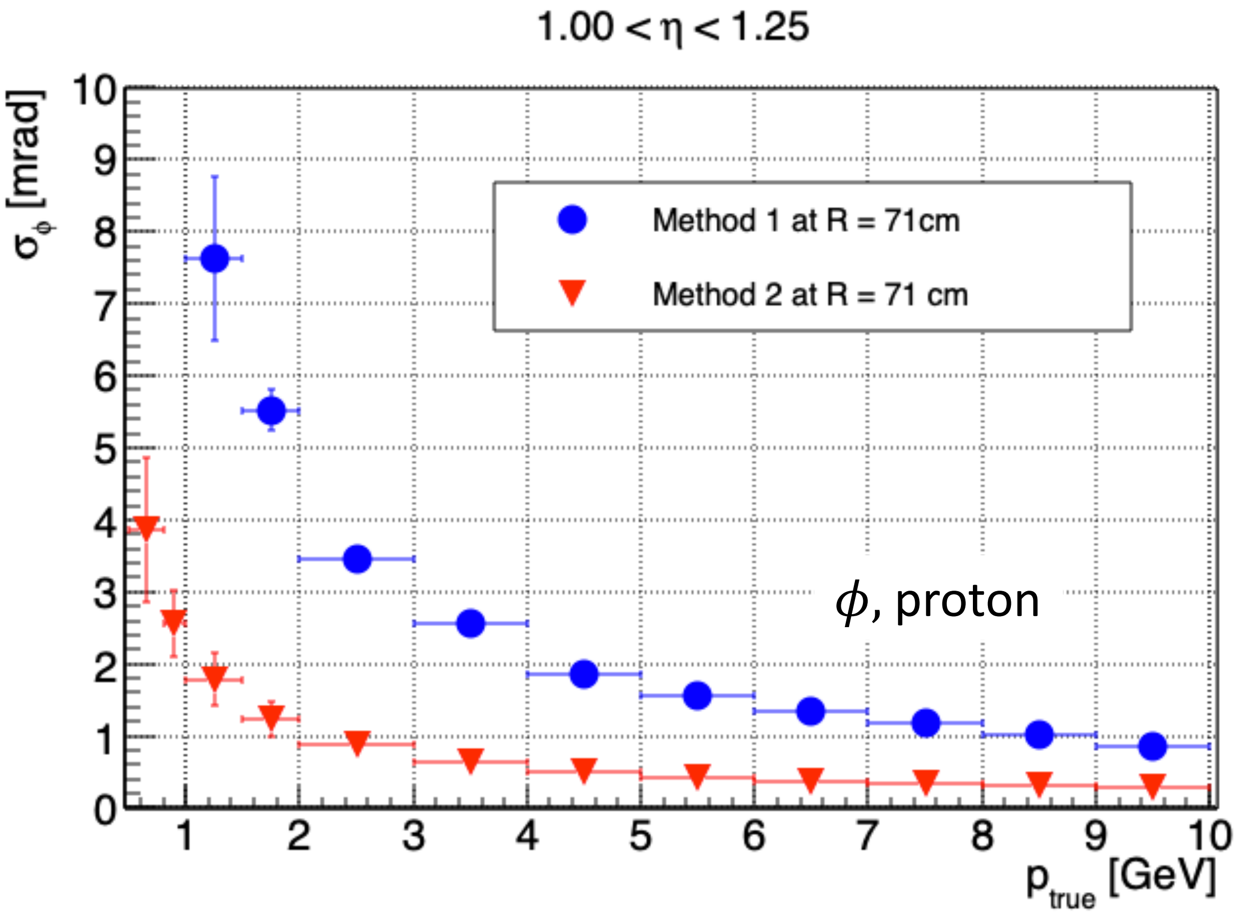




Method 1 vs. Method 2

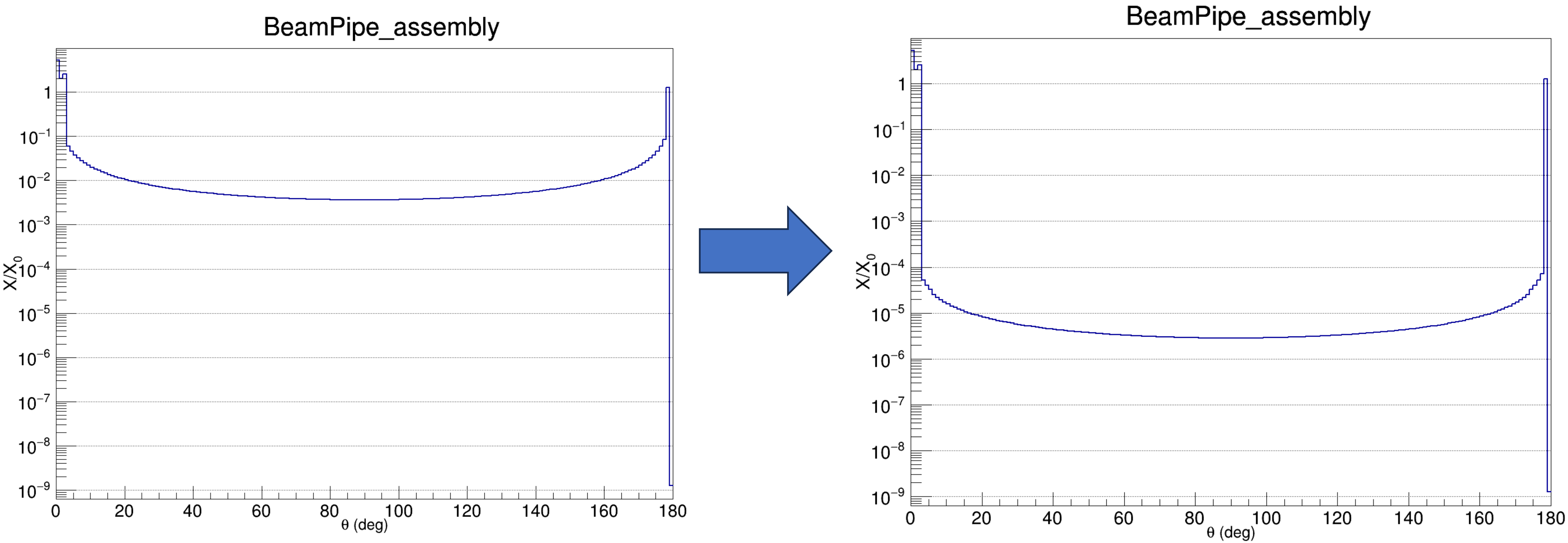
- ❑ Use difference between MC and Reconstructed tracks to obtain angular resolutions
- ➢ **Clear angular resolution difference between Method 1 and Method 2**





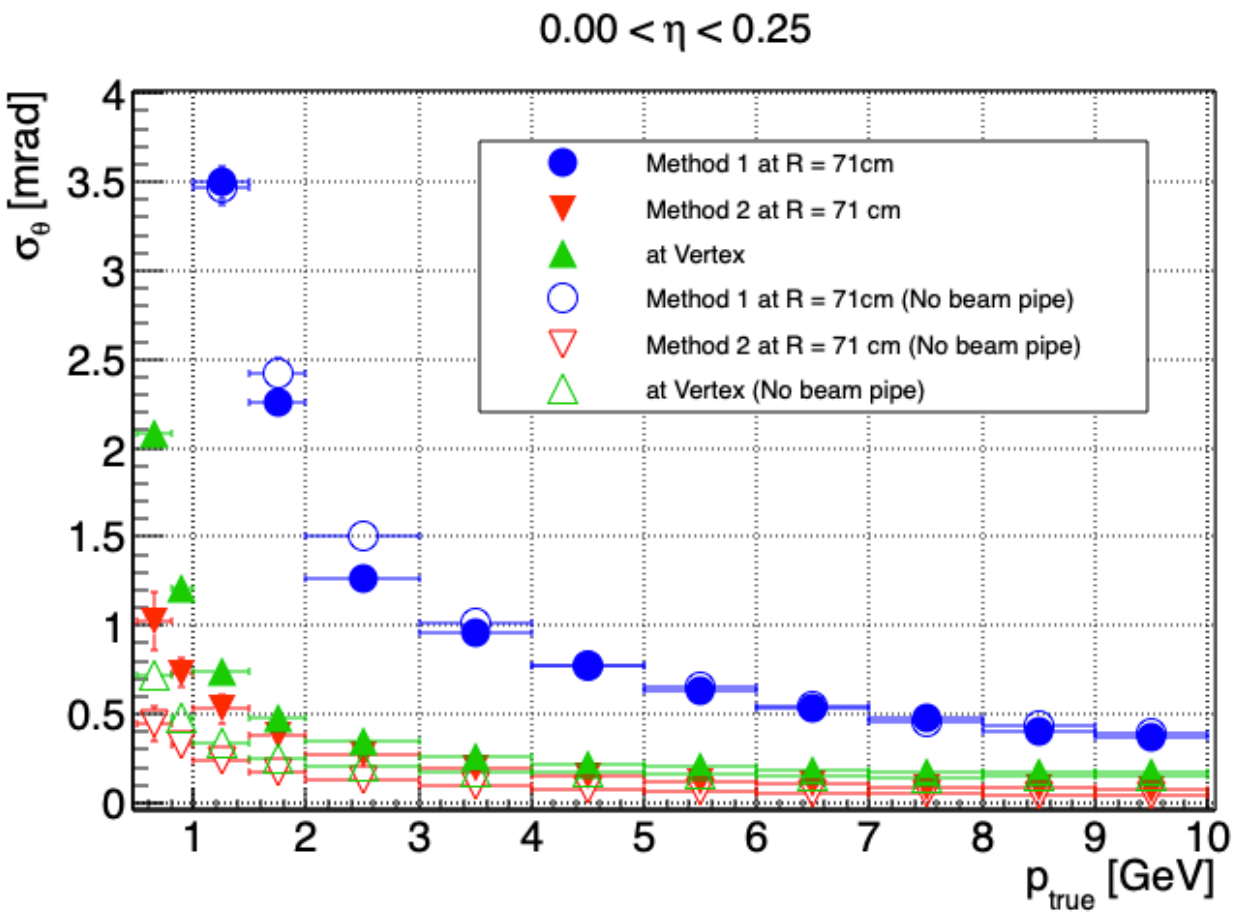
Effect of Beam Pipe

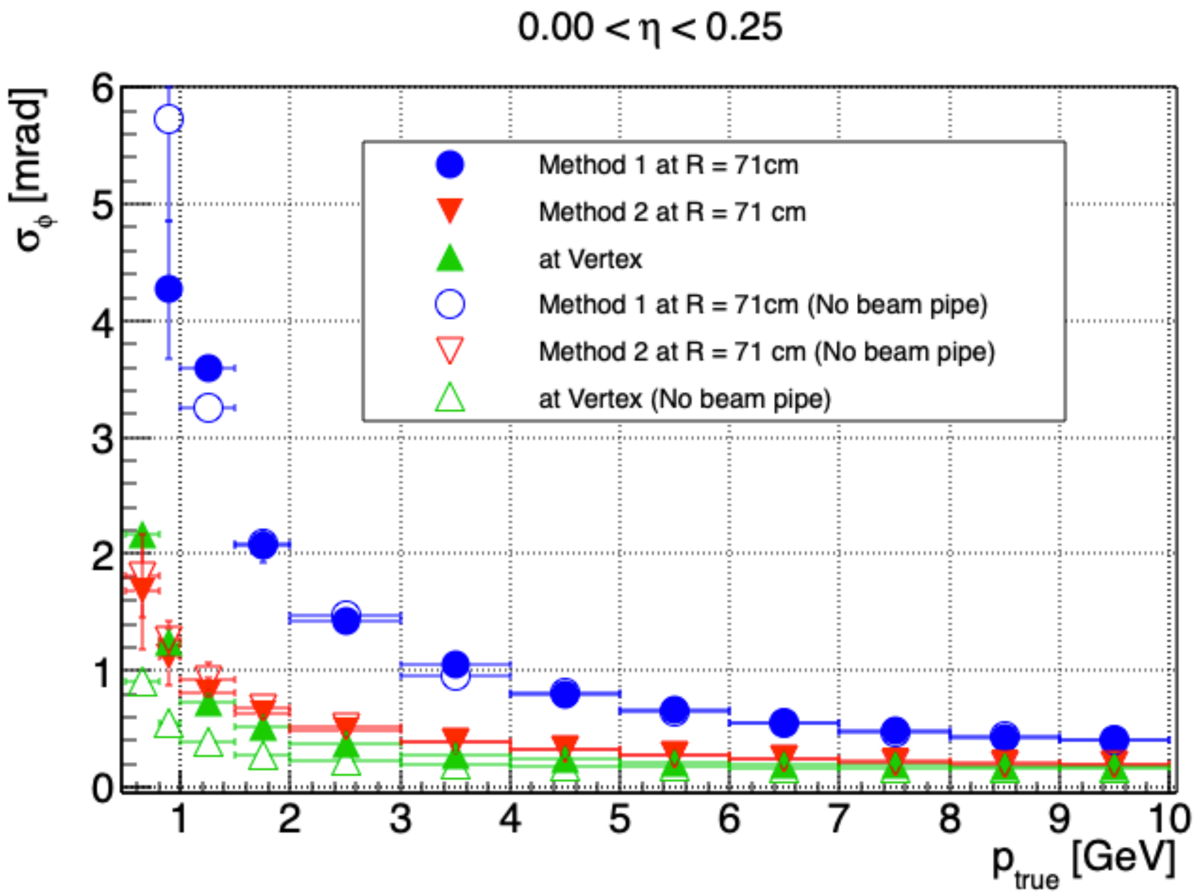
- Investigate effect of reducing beam pipe material



Effect of Beam Pipe

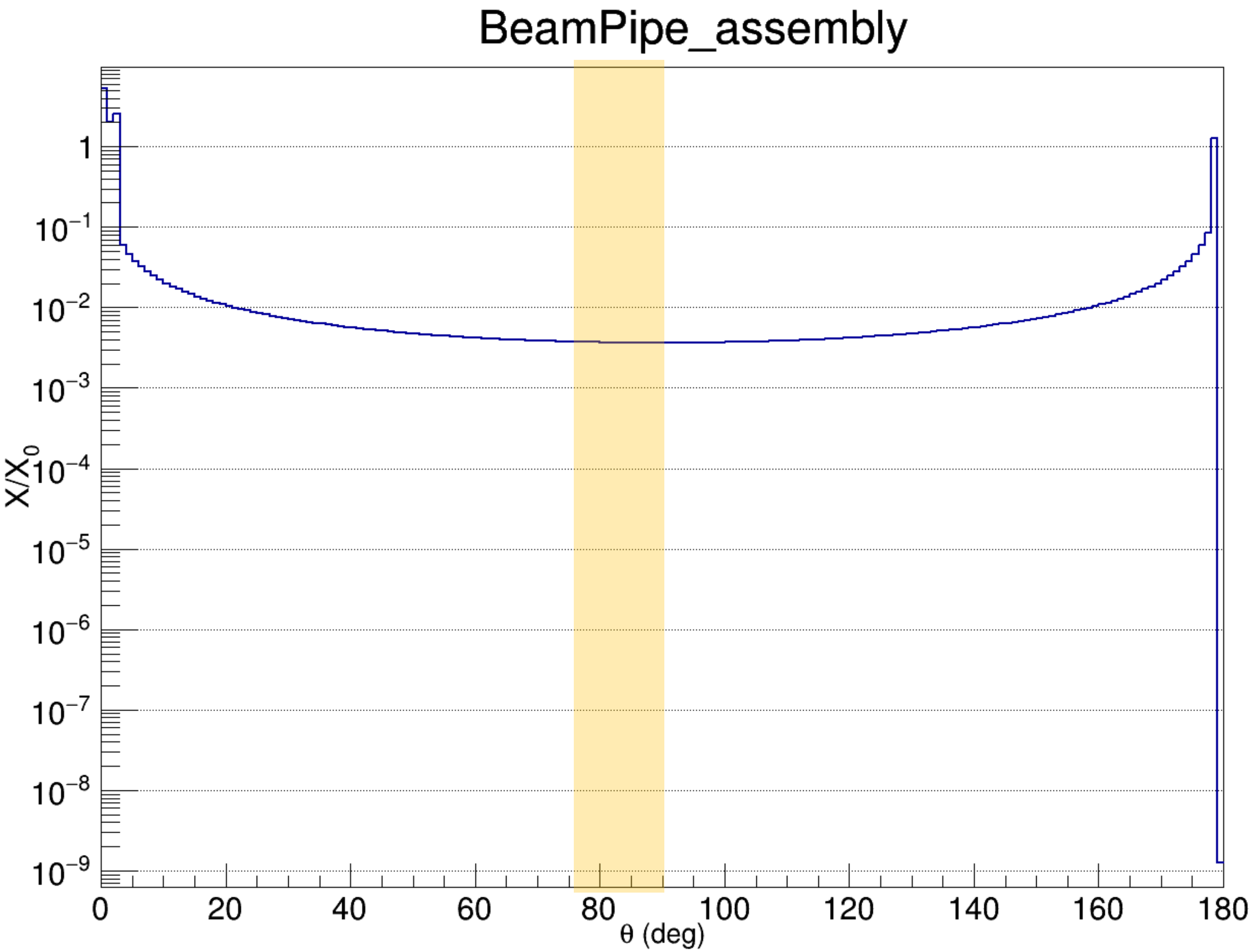
- **Closed markers = with beam pipe, Open markers = "without" beam pipe**
  - Expect beam pipe material to affect angular resolution wrt vertex more than wrt to another surface
  - Beam pipe material affects angular resolution
    - at the vertex the most and from Method 1 the least





Multiple Scatter from Beam Pipe Calculation

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



PDG

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

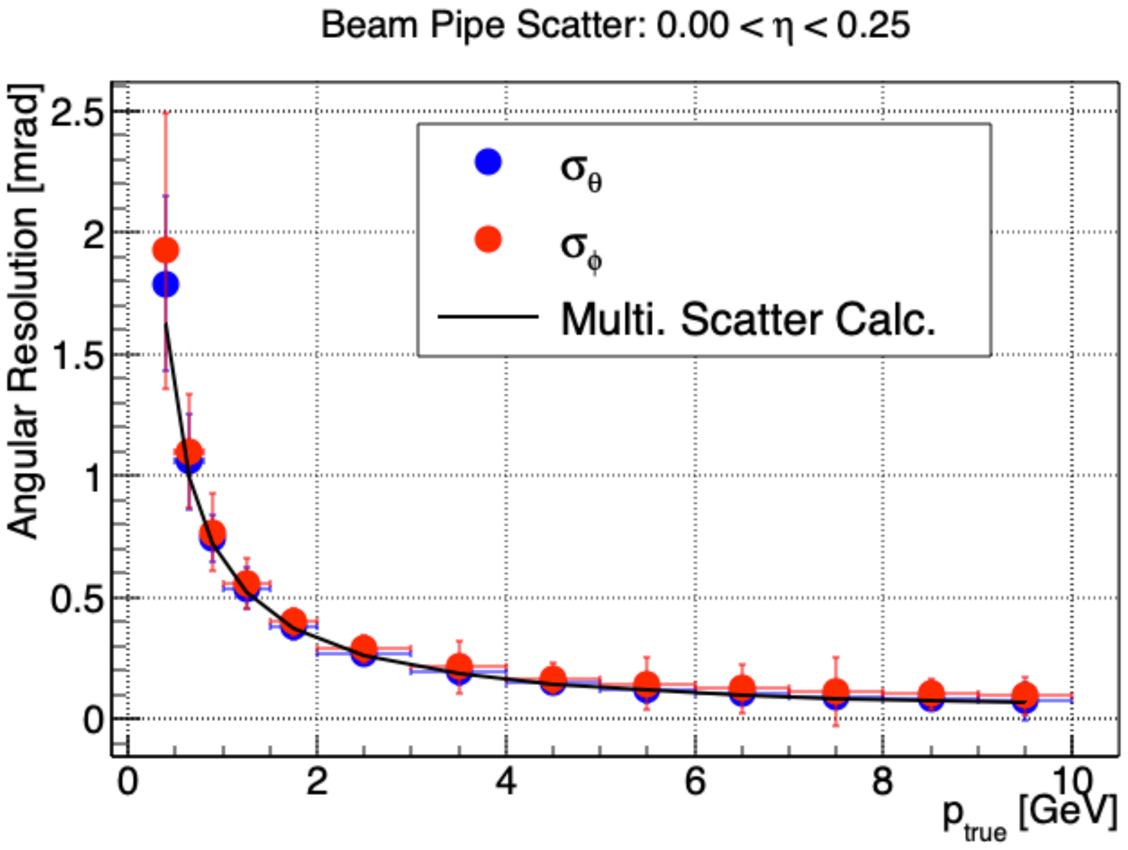
$$\theta_0 = \frac{13.6\ {\rm MeV}}{\beta c p}\, z \sqrt{\frac{x}{X_0}}\left[1 + 0.088 \log_{10}\left(\frac{x z^2}{X_0 \beta^2}\right)\right]$$

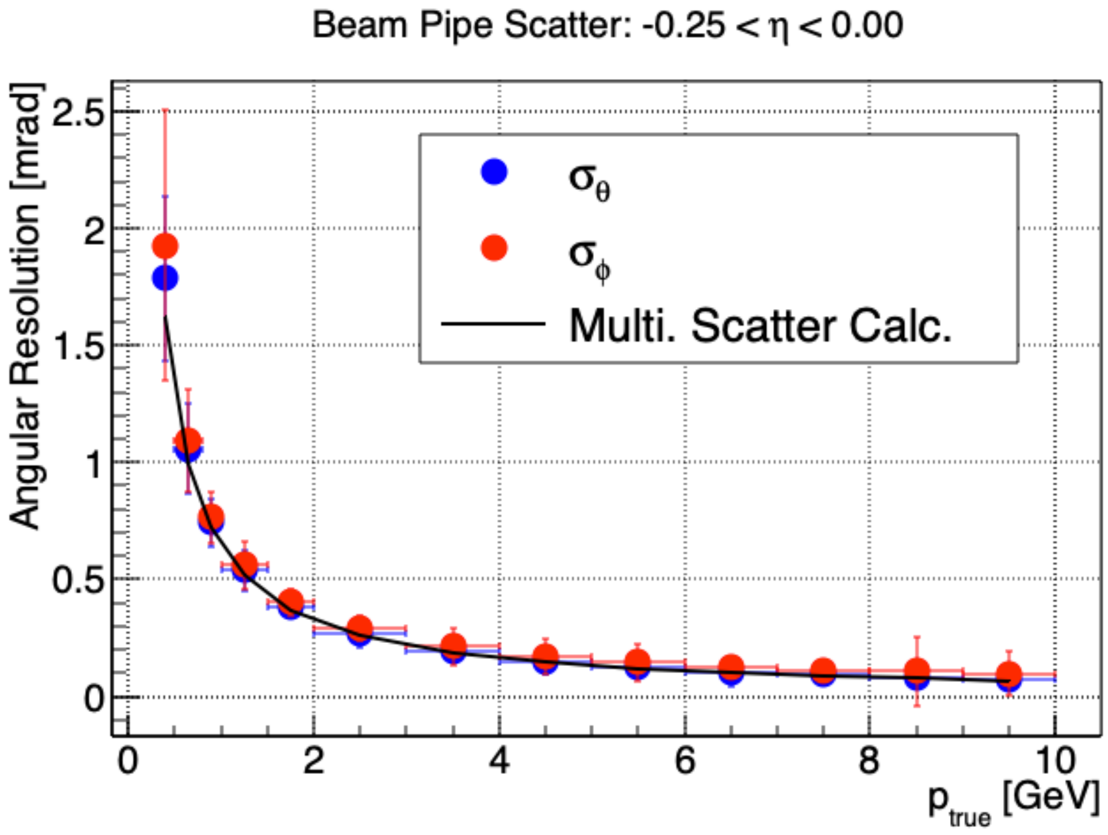
$$= \frac{13.6\ {\rm MeV}}{\beta c p}\, z \sqrt{\frac{x}{X_0}}\left[1 + 0.038 \ln\left(\frac{x z^2}{X_0 \beta^2}\right)\right]$$

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

Multiple Scatter from Beam Pipe Calculation

- ❑ 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**

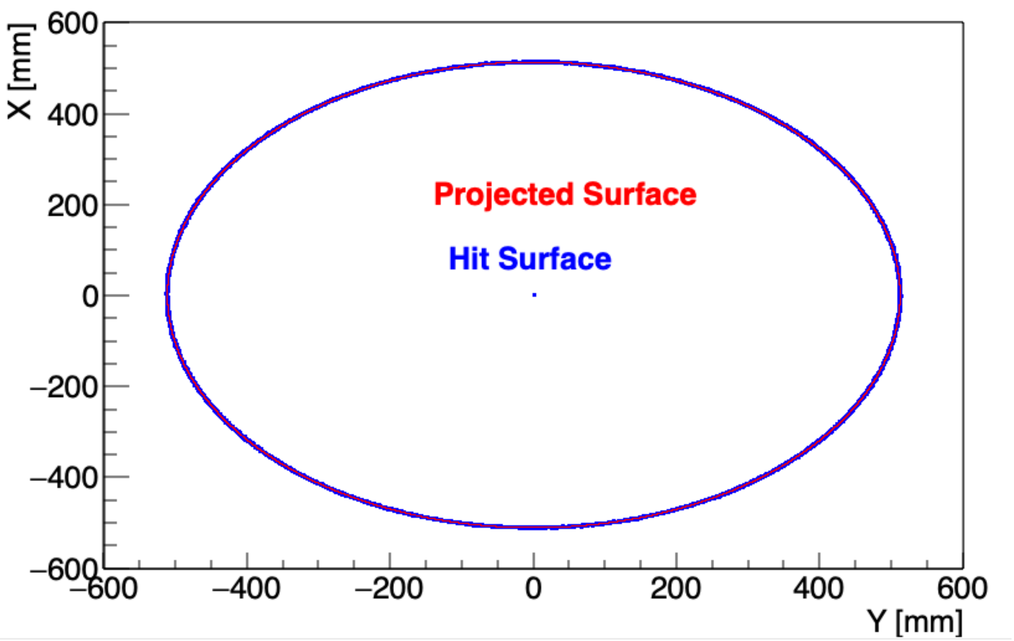


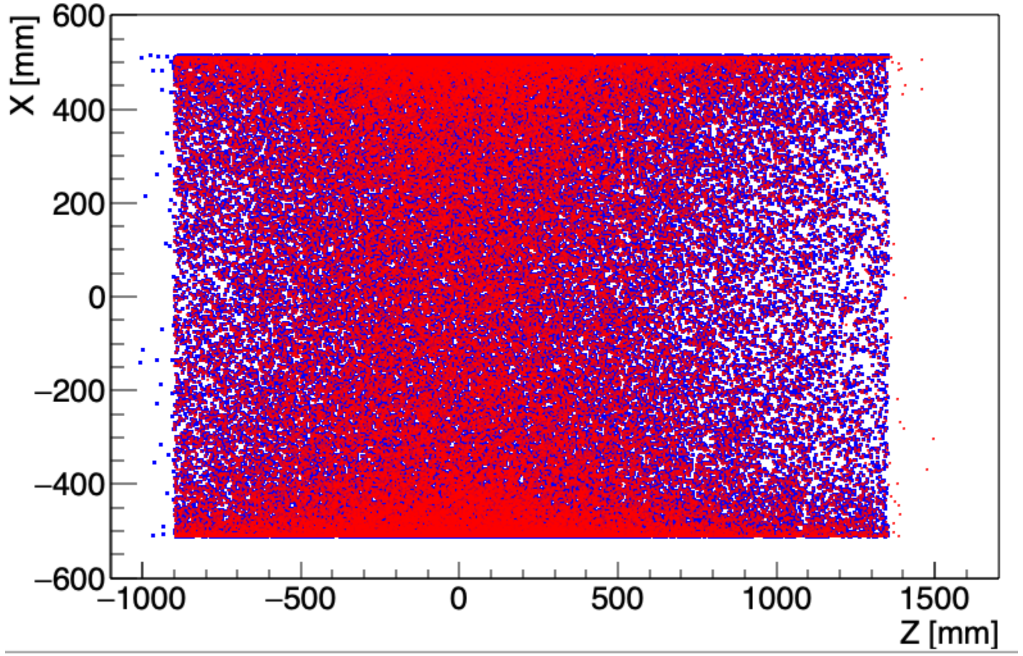


Method 1 Test

- Does geometrical mismatch between planar DIRC bars and cylindrical projected surface lead to discrepancy between Method 1 and 2?
- Test:
  - Project surface to inner MPGD barrel layer (R = 51 cm)
  - Both layers are cylindrical – Eliminates geometric mis-match
  - Compare resolutions calculated from Method 1 and Method 2

Need some more time to look over results
Will present follow up





Summary

- Method 1 and 2 give different results
- Method 2 agrees with multiple scattering calculations
- Method 1 has mismatch between detector and projected surface
  - Follow up: Compare the two methods using matching detector and projection surfaces

    → e.g. inner MPGD