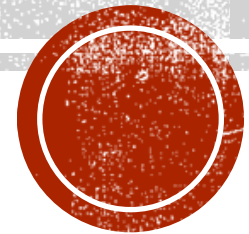# bHCAL Meeting — Acceptance Hole

Jan Vanek

University of New Hampshire

10/03/2025

OVERVIEW

- Study to determine origin of hole in bHCAL acceptance in ePIC simulation framework

- Simulation setup:
  1. Simulate single particles using npsim and pass through ePIC
     - Generated 10k π^+ at 2 GeV for each simulation pass
     - Different kinematic distributions in η and ϕ (details in corresponding slides)
  2. Reconstruct using EICRecon
  3. Fill histograms

DEFAULT SIMULATION

- Default simulation to re-produce the hole

- Particle gun setup:
  - π^+ at 2 GeV (10k)
  - Uniform in ϕ
  - $\theta = (33.5, 146)$ deg, with $\cos\theta$ distribution

- (top) Full ePIC detector

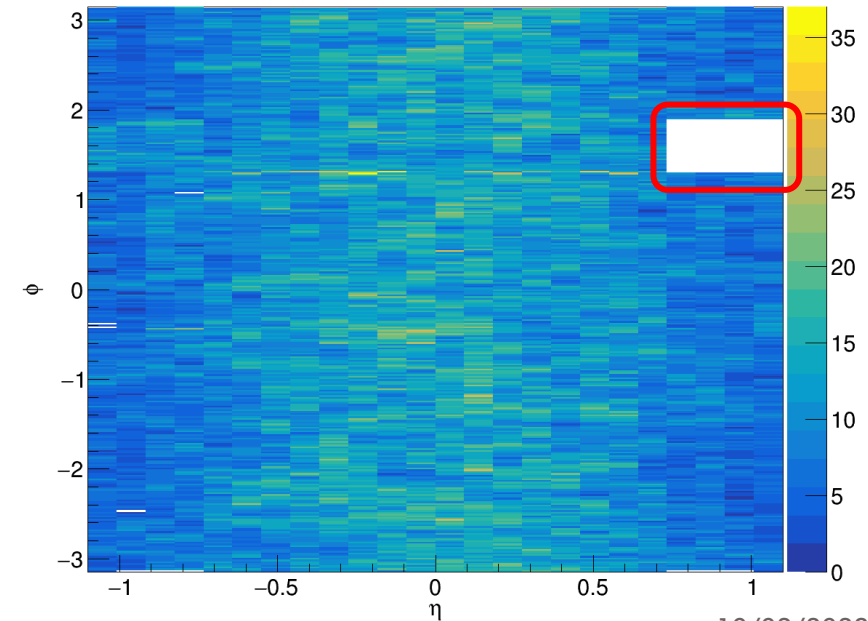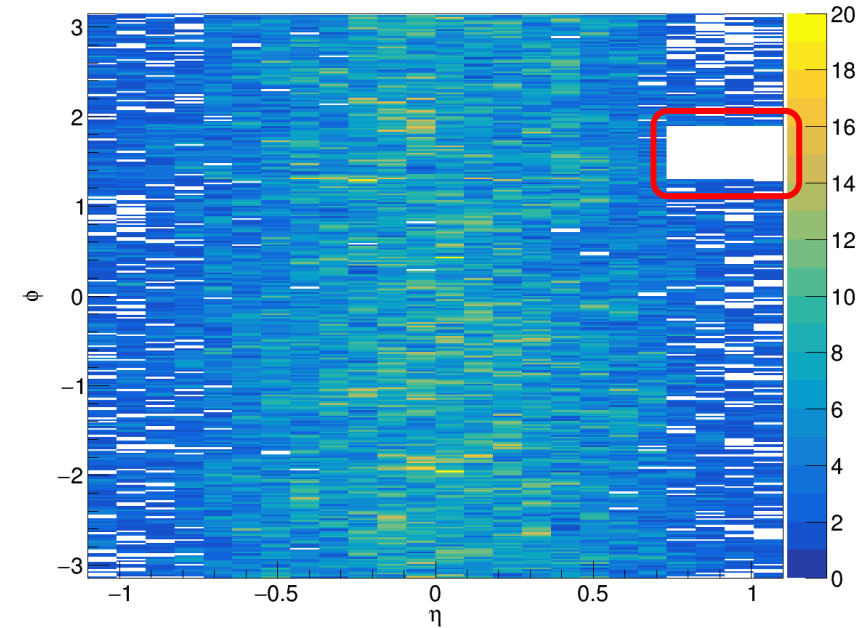- (bottom) bHCAL only
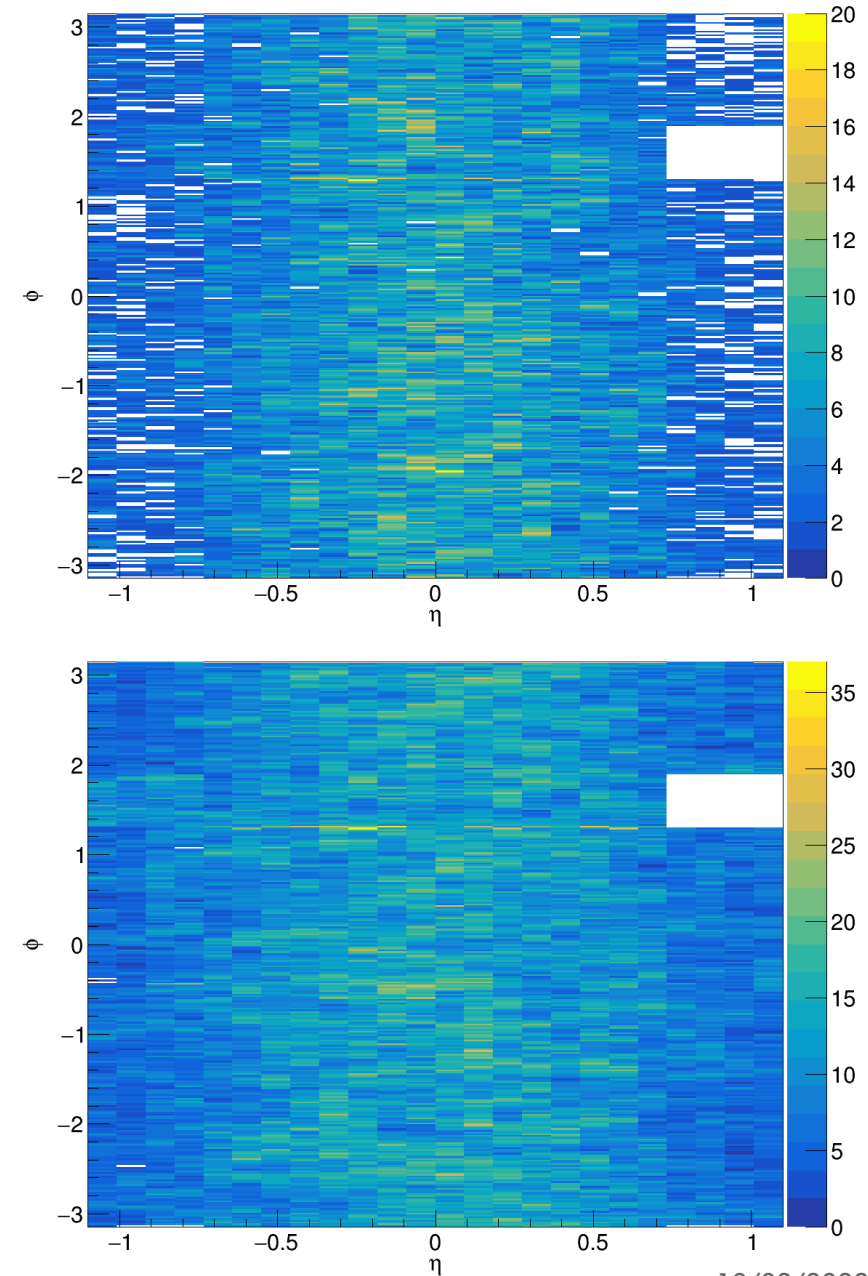
DEFAULT SIMULATION

- Default simulation to re-produce the hole

- Particle gun setup:
  - π^+ at 2 GeV (10k)
  - Uniform in ϕ
  - $\theta = (33.5, 146)$ deg, with $\cos\theta$ distribution

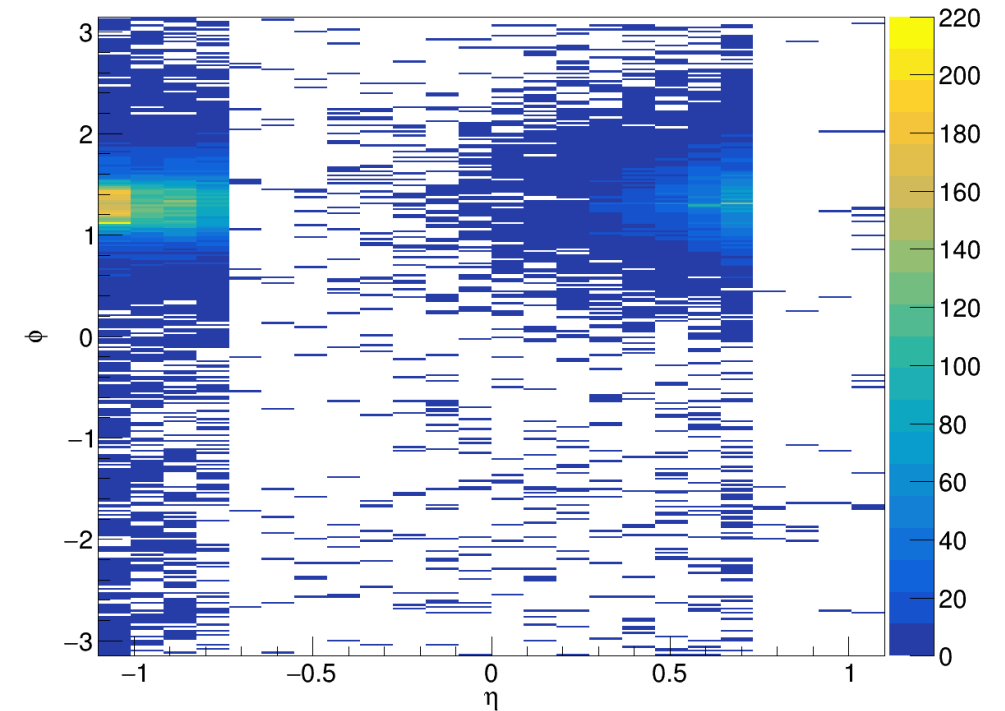- (top) Full ePIC detector

- (bottom) bHCAL only

- "Successfully" reproduced the acceptance hole

ACCEPTANCE SCAN 1

- Test simulation scanning bHCAL acceptance
  - Shooting π^+ to a window about the size of the hole
  - Changing position of the window

- Particle gun setup:
  - π^+ at 2 GeV (10k)
  - $\phi = (1.35, 1.80)$ rad, uniform
  - $\eta = (0.8, 1.1)$, uniform

ACCEPTANCE SCAN 2

- Test simulation scanning bHCAL acceptance
  - Shooting π^+ to a window about the size of the hole
  - Changing position of the window

- Particle gun setup:
  - π^+ at 2 GeV (10k)
  - $\phi = (-1.80, -1.35)$ rad, uniform
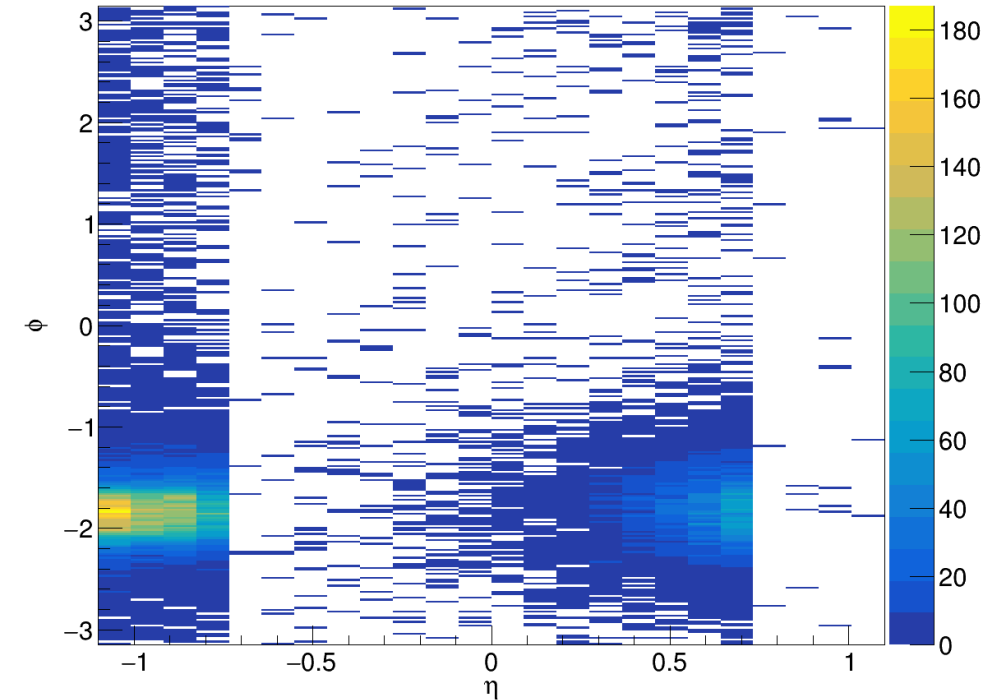 - $\eta = (0.8, 1.1)$, uniform

ACCEPTANCE SCAN 3

- Test simulation scanning bHCAL acceptance
  - Shooting π^+ to a window about the size of the hole
  - Changing position of the window

- Particle gun setup:
  - π^+ at 2 GeV (10k)
  - $\phi = (-1.80, -1.35)$ rad, uniform
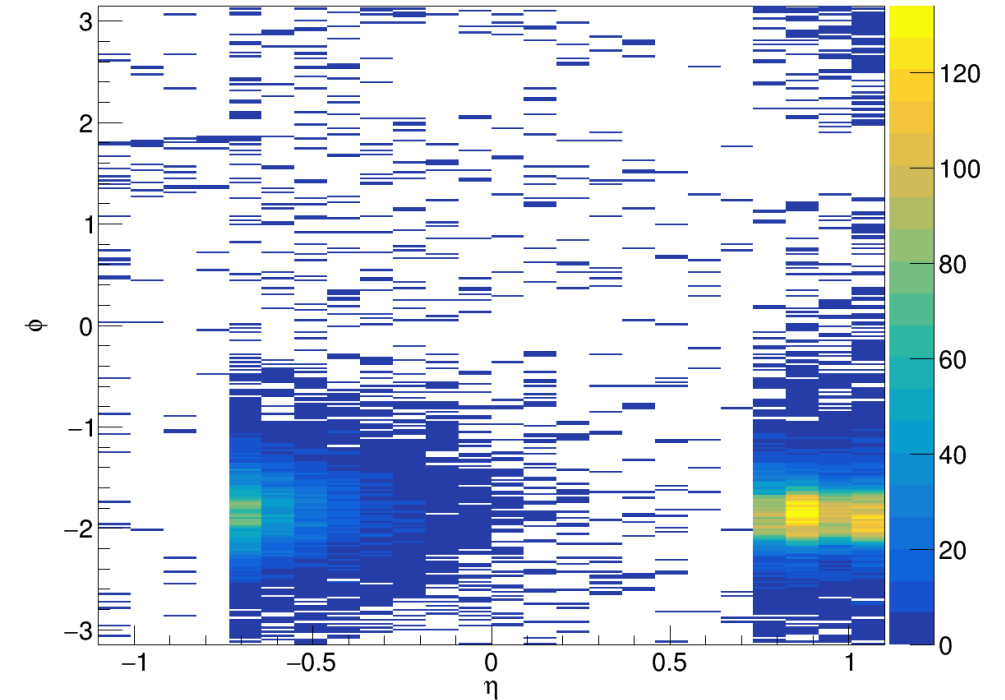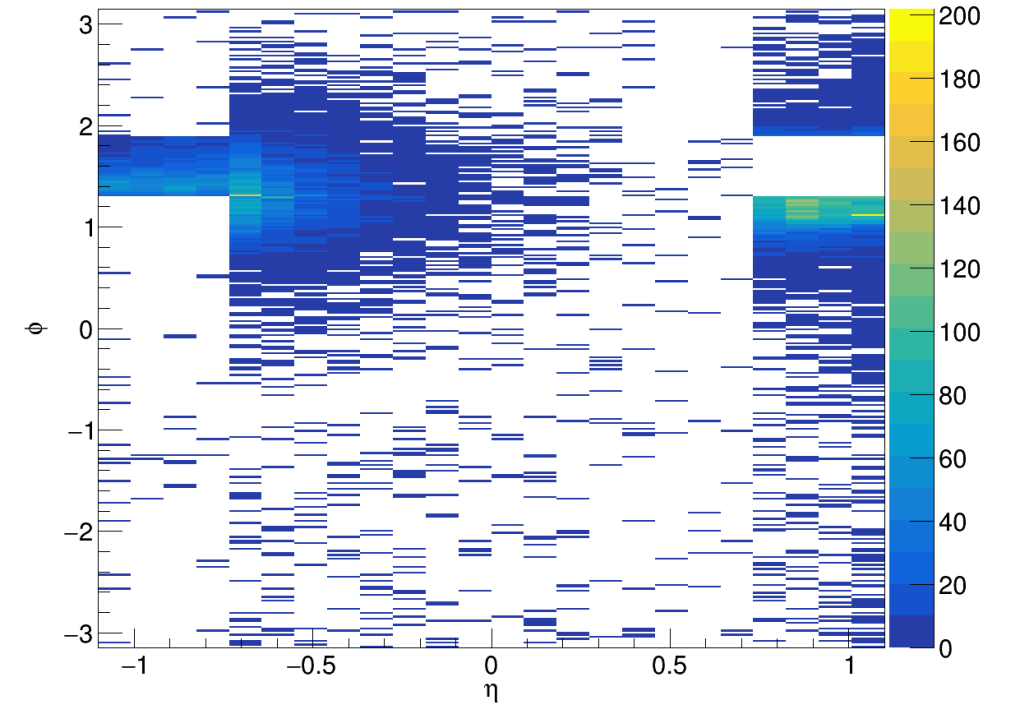 - $\eta = (-1.1, -0.8)$, uniform

ACCEPTANCE SCAN 4

- Test simulation scanning bHCAL acceptance
  - Shooting π^+ to a window about the size of the hole
  - Changing position of the window

- Particle gun setup:
  - π^+ at 2 GeV (10k)
  - $\phi = (1.35, 1.80)$ rad, uniform
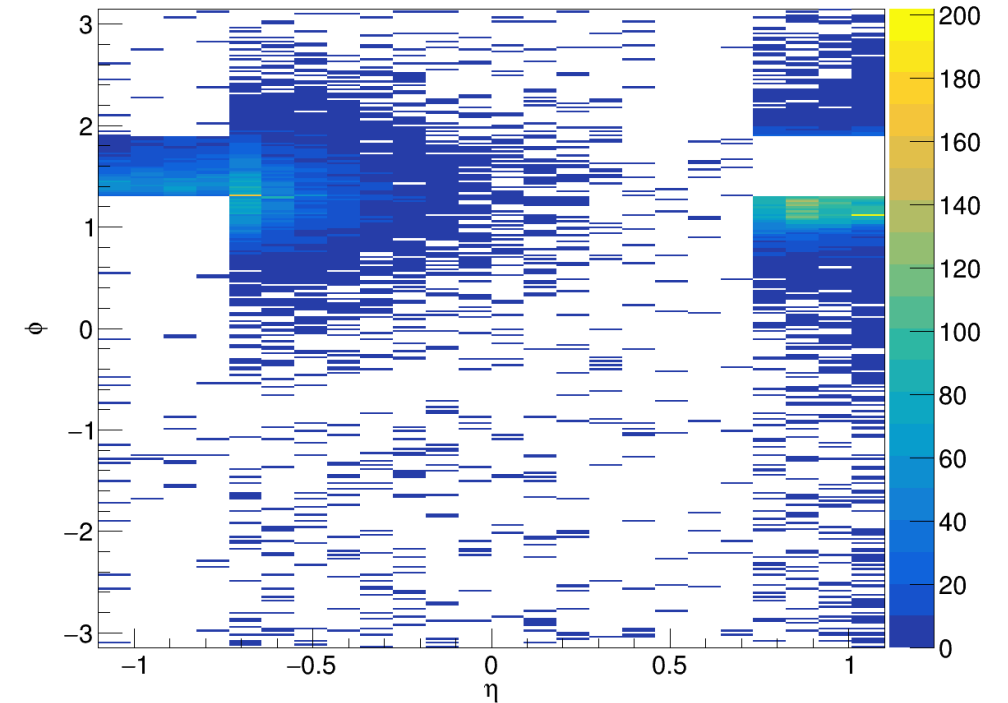 - $\eta = (-1.1, -0.8)$, uniform

ACCEPTANCE SCAN – OBSERVATIONS

- The acceptance hole is 4x30 tiles (ηxϕ)
  - Size exactly corresponds to the "chimney tiles"
  - Hole is on the opposite side from the chimney tiles

ACCEPTANCE SCAN – OBSERVATIONS

- The acceptance hole is 4x30 tiles (ηxϕ)
  - Size exactly corresponds to the "chimney tiles"
  - Hole is on the opposite side from the chimney tiles

ACCEPTANCE SCAN – OBSERVATIONS

- The acceptance hole is 4x30 tiles (ηxϕ)
  - Size exactly corresponds to the "chimney tiles"
  - Hole is on the opposite side from the chimney tiles

- Hits in 4 outer rows of tiles in η get recorded with the wrong sign
  - Hits that should have positive η are observed with negative η, and vice versa
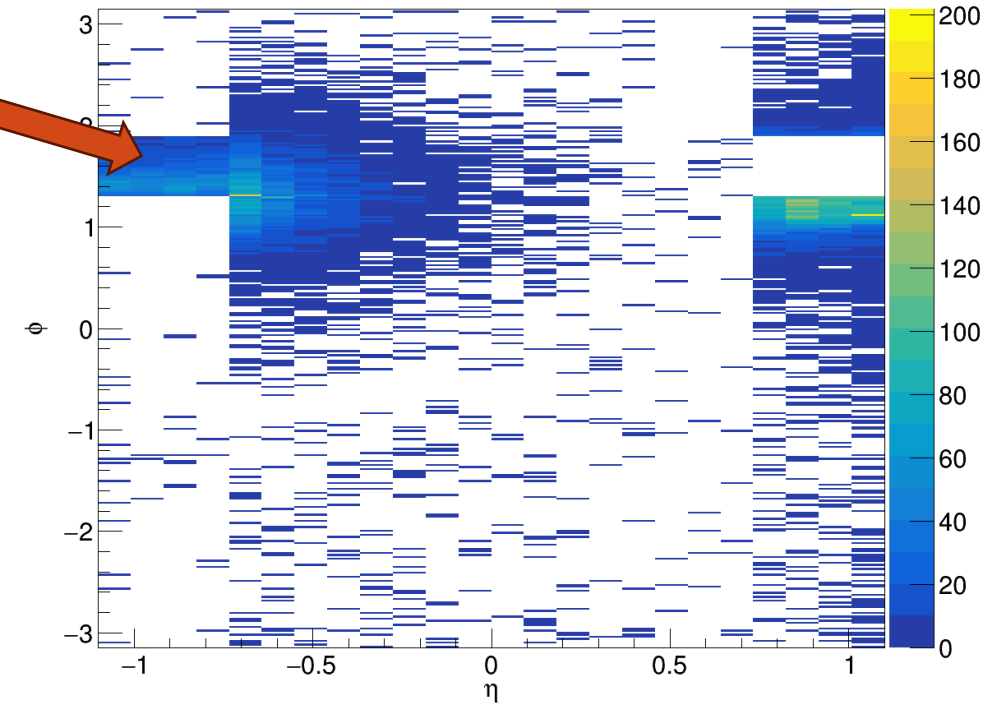 - Exception are the chimney tiles, that appear to be recorded correctly

ACCEPTANCE SCAN – OBSERVATIONS

- The acceptance hole is 4x30 tiles (ηxϕ)
  - Size exactly corresponds to the "chimney tiles"
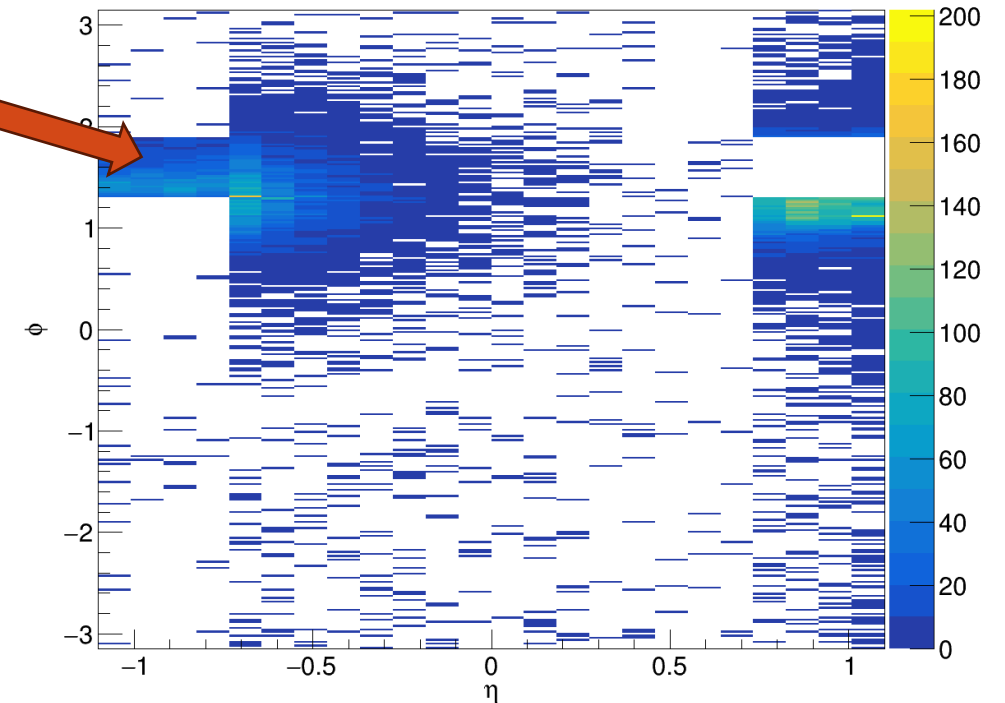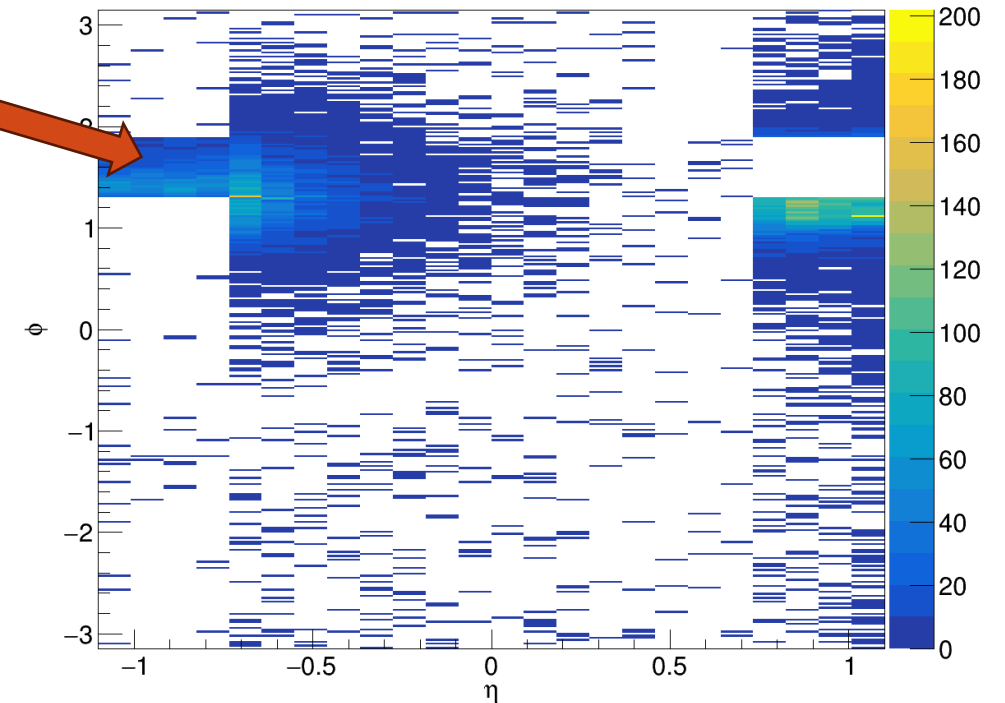 - Hole is on the opposite side from the chimney tiles

- Hits in 4 outer rows of tiles in η get recorded with the wrong sign
  - Hits that should have positive η are observed with negative η, and vice versa
  - Exception are the chimney tiles, that appear to be recorded correctly

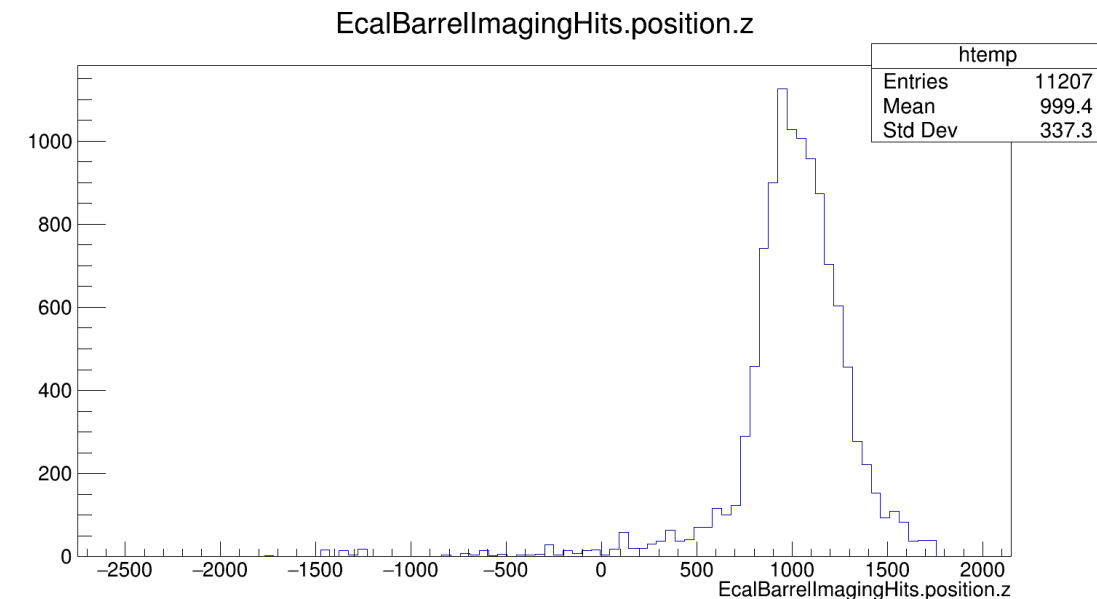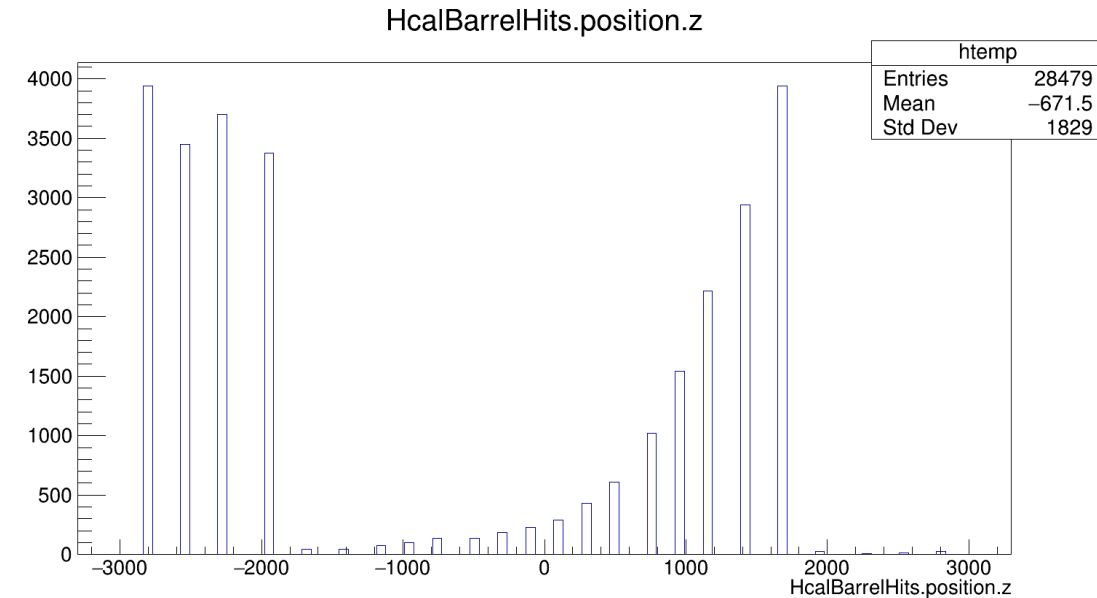- So, how does the hole appear?
  - If a particle hits the hole, it is wrongly recorded on the opposite side, i.e. in the chimney tiles
  - If a particle hits the chimney tile, it is recorded correctly
  - As a result, nothing gets recorded in the hole

MISMATCH ORIGIN?

- Attempt to determine, when the problem occurs:
  - Npsim vs. EICRecon vs. histogram filling

- (top) Location of bHCAL hits in z in edm4hep output from npsim
  - Example from Acceptance scan 2 (Slide 6)

- (bottom) Location of bECAL hits in z in edm4hep output from npsim
  - From the same file

- **The issue appears already in the first step in the simulation chain and is exclusive to the bHCAL**

HcalBarrelHits.position.z

htemp	
Entries	28479
Mean	−671.5
Std Dev	1829

EcalBarrelImagingHits.position.z

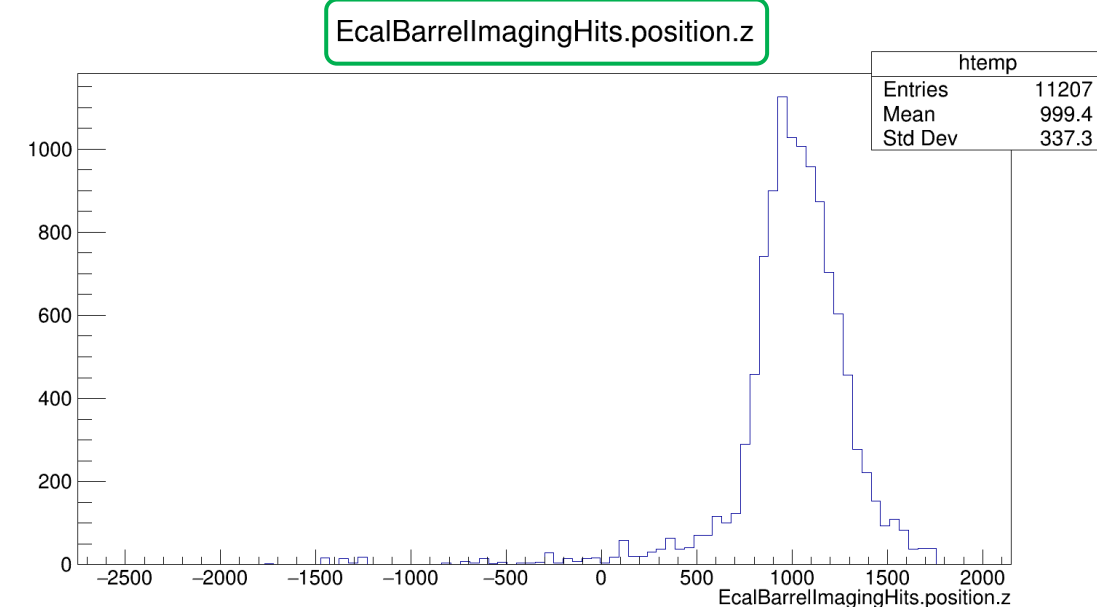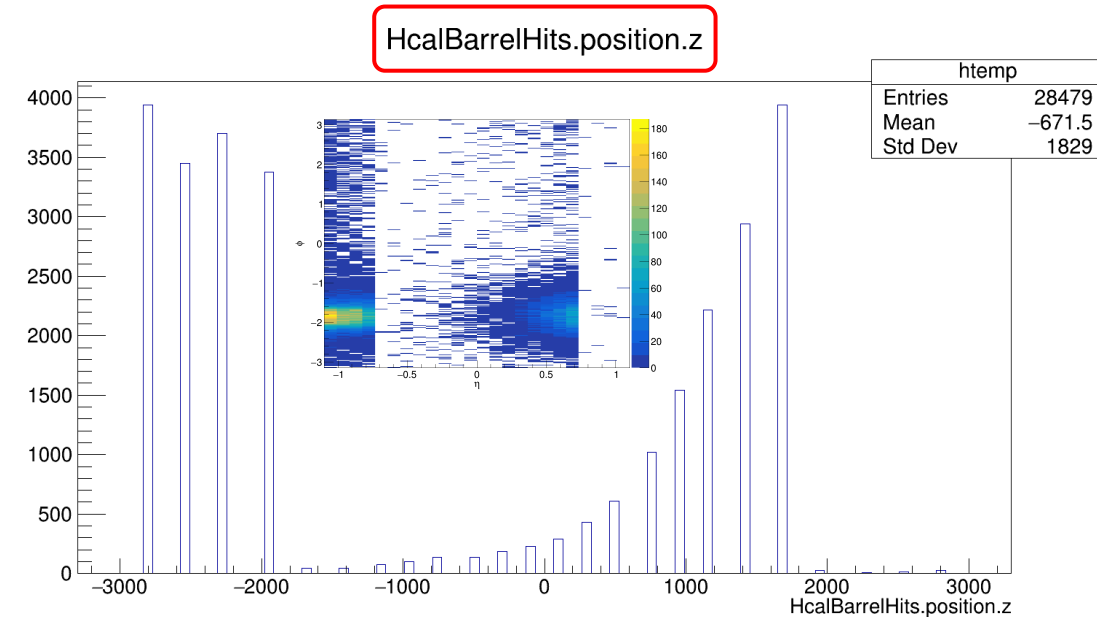| htemp	
Entries	11207
Mean	999.4
Std Dev	337.3

MISMATCH ORIGIN?

- Attempt to determine, when the problem occurs:
  - Npsim vs. EICRecon vs. histogram filling

- (top) Location of bHCAL hits in z in edm4hep output from npsim
  - Example from Acceptance scan 2 (Slide 6)

- (bottom) Location of bECAL hits in z in edm4hep output from npsim
  - From the same file

- **The issue appears already in the first step in the simulation chain and is exclusive to the bHCAL**
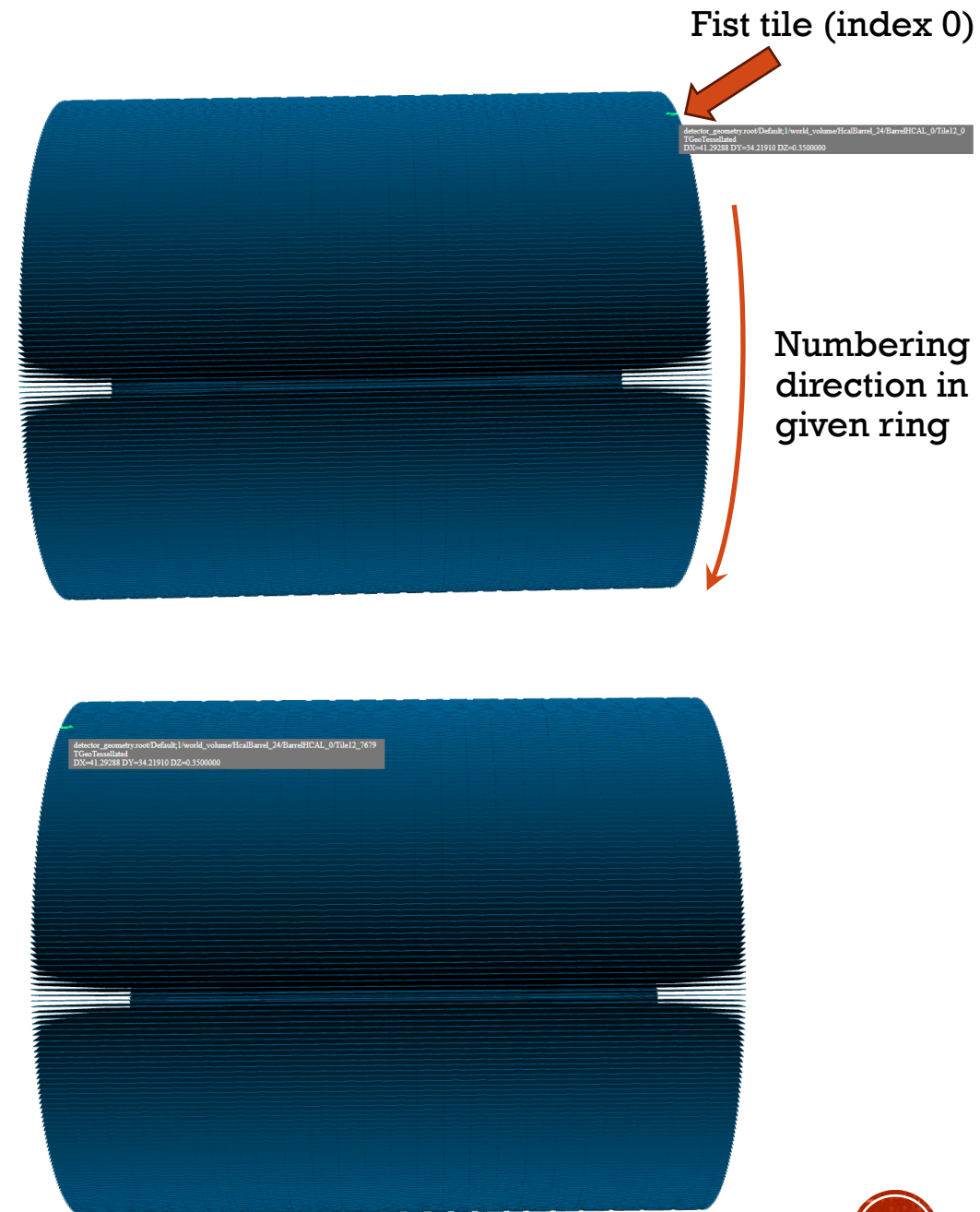
HcalBarrelHits.position.z

htemp	
Entries	28479
Mean	−671.5
Std Dev	1829

EcalBarrelImagingHits.position.z

htemp	
Entries	11207
Mean	999.4
Std Dev	337.3

TILE PLACEMENT

- Investigated tile placement in bHCAL model
  - In cpp macro that builds the detector
  - In exported detector geometry

- Key observations
  - Changing indexing of tiles in the cpp does not seem to affect the hole
    - Checked with locally compiled ePIC detector with various changes to the indexing of the tiles
  - The default indexing seems to be correct
    - The tile indices start in the outer ring at positive η and then increase by 1 in each η ring

Fist tile (index 0)
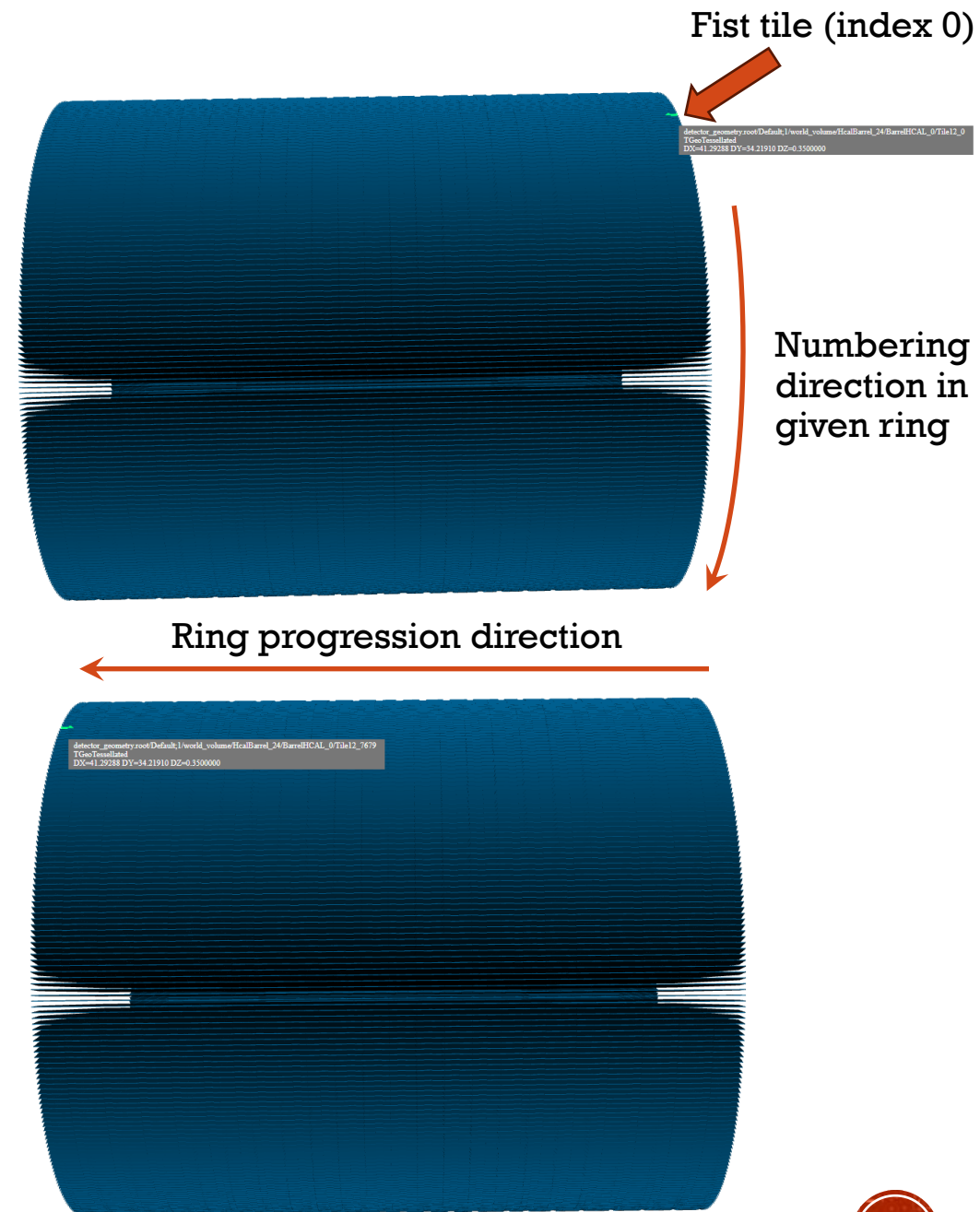
Numbering direction in given ring

TILE PLACEMENT

- Investigated tile placement in bHCAL model
  - In cpp macro that builds the detector
  - In exported detector geometry

- Key observations
  - Changing indexing of tiles in the cpp does not seem to affect the hole
    - Checked with locally compiled ePIC detector with various changes to the indexing of the tiles
  - The default indexing seems to be correct
    - The tile indices start in the outer ring at positive η and then increase by 1 in each η ring
    - Once a ring is complete, the indices continue at nearest η ring at smaller η ("to the left")

Fist tile (index 0)

Numbering direction in given ring
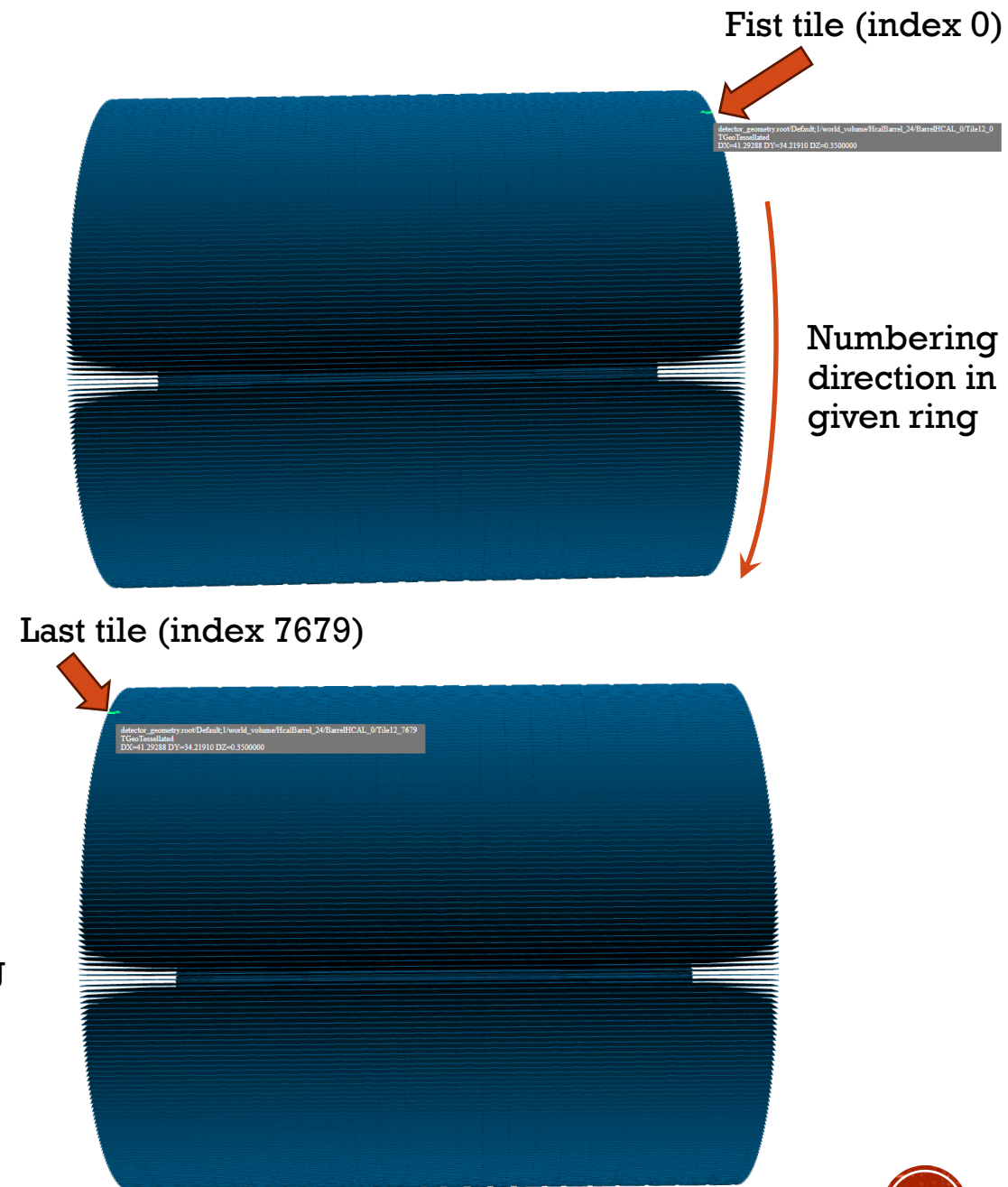
Ring progression direction

TILE PLACEMENT

- Investigated tile placement in bHCAL model
  - In cpp macro that builds the detector
  - In exported detector geometry

- Key observations
  - Changing indexing of tiles in the cpp does not seem to affect the hole
    - Checked with locally compiled ePIC detector with various changes to the indexing of the tiles
  - The default indexing seems to be correct
    - The tile indices start in the outer ring at positive η and then increase by 1 in each η ring
    - Once a ring is complete, the indices continue at nearest η ring at smaller η ("to the left")
    - Last tile is on the opposite side from the first tile

Fist tile (index 0)

detector_geometry.root/Default;1/world_volume/HcalBarrel_24/BarrelHCAL_0/Tile12_0
TGeoTessellated
DX=41.29288 DY=34.21910 DZ=0.3500000

Numbering direction in given ring

Last tile (index 7679)

detector_geometry.root/Default;1/world_volume/HcalBarrel_24/BarrelHCAL_0/Tile12_7679
TGeoTessellated
DX=41.29288 DY=34.21910 DZ=0.3500000

SUMMARY AND OUTLOOK

- Reproduced acceptance hole

- The hole appears to be associated with "chimney tiles" of the bHCAL

- The issue is larger than simple acceptance hole
  - Hits in 4 outside η rings are recorded on the wrong side of the detector in η
  - Exception seem to be the "chimney tiles" which record hits as expected

- The issue appears already in the first step of the simulation chain when running npsim
  - Mismatch visible in output trees from npsim
  - Seems to be exclusive to bHCAL, as hits in other detectors are OK
    - Checked with bECAL

- Tile indexing in detector model seems to be OK
  - Checked both cpp macro and exported detector model

- **To-do:**
  - Determine precise origin of the mismatching issue

THANK YOU FOR ATTENTION