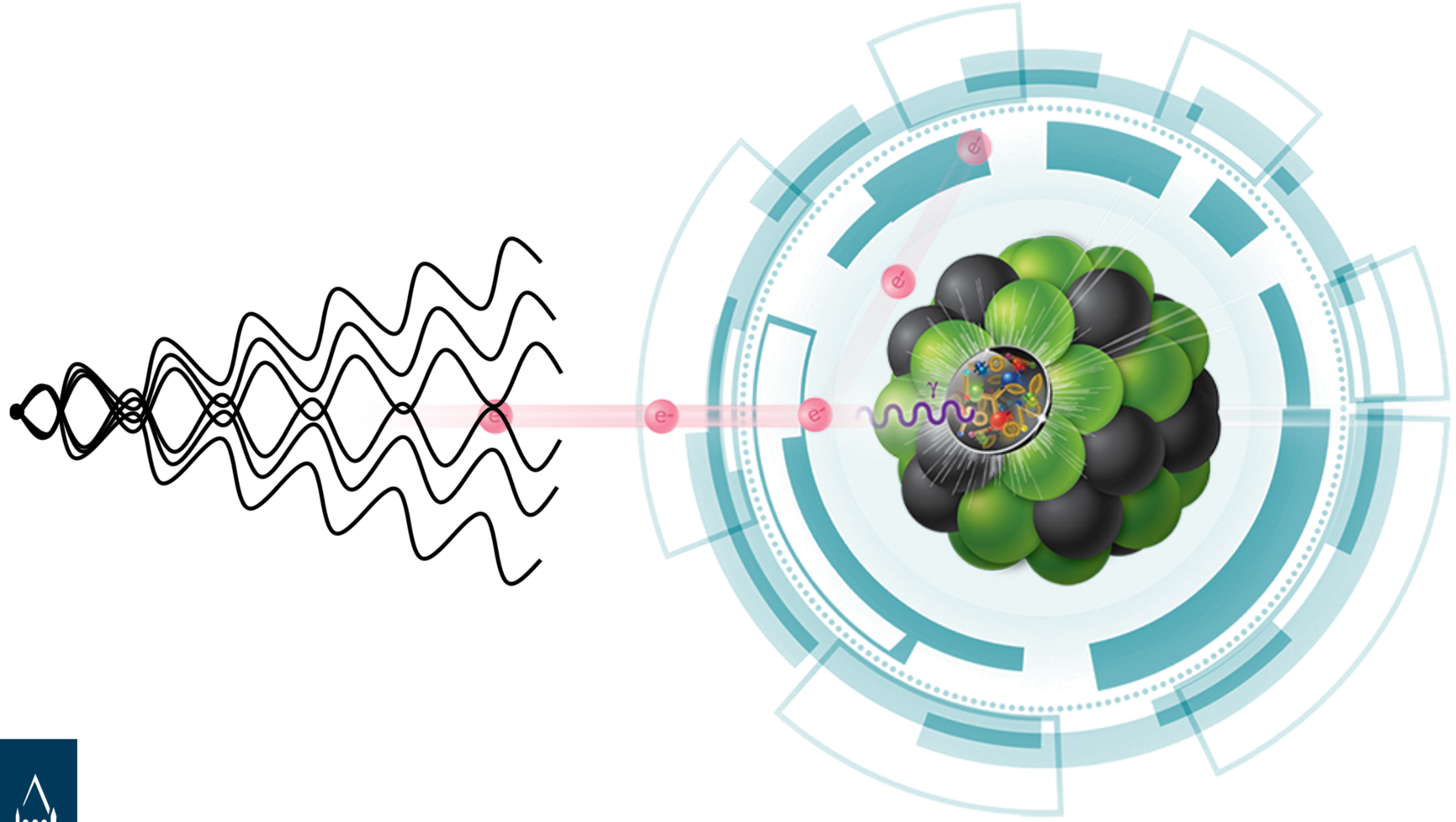
Synchrotron Radiation Studies



Rey Cruz-Torres
EIC Tracking Meeting
06/16/2022

Outline

- (Simulated) data structure
- Initial studies
- From 1 photon to an event

Outline

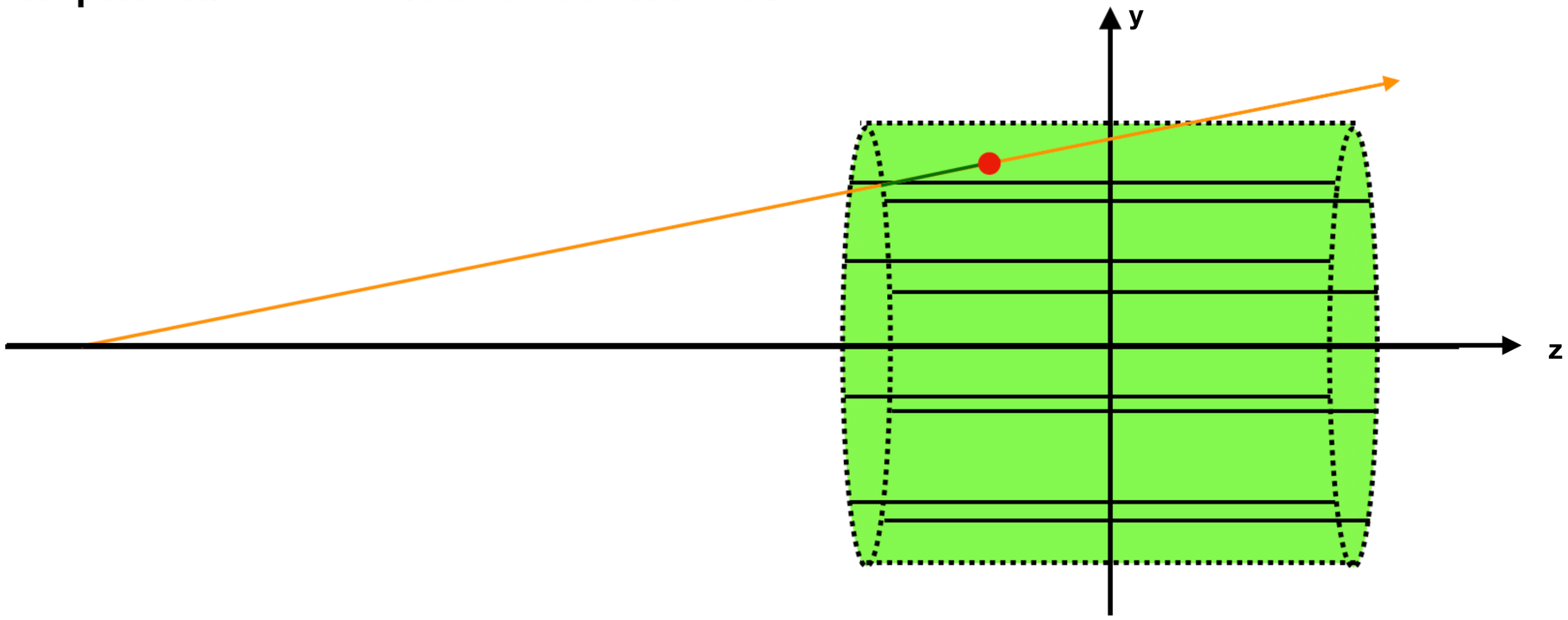
- (Simulated) data structure
- Initial studies
- From 1 photon to an event

Photons from Synchroton Radiation

Synrad+* simulation of photons due to the final electron forward magnets.

"virtual cylinder" comprised of rectangular facets placed just inside the IR beampipe and photons which pass through these facets are recorded.

A hepmc file is created for each facet



Data Structure

- Data stored in hepmc files (one for each facet)
- Each event corresponds to one photon that passed through virtual cylinder
- For each photon the following information is included:
  - Coordinate of hit in virtual cylinder (lab frame): x, y, z [mm]
  - Momentum of photon (lab frame): p_x, p_y, p_z, E [GeV]
  - Weight factor necessary to scale photons and extract photon flux

Two Photon Samples Available

Sample #1 (2021)

Files can he found here

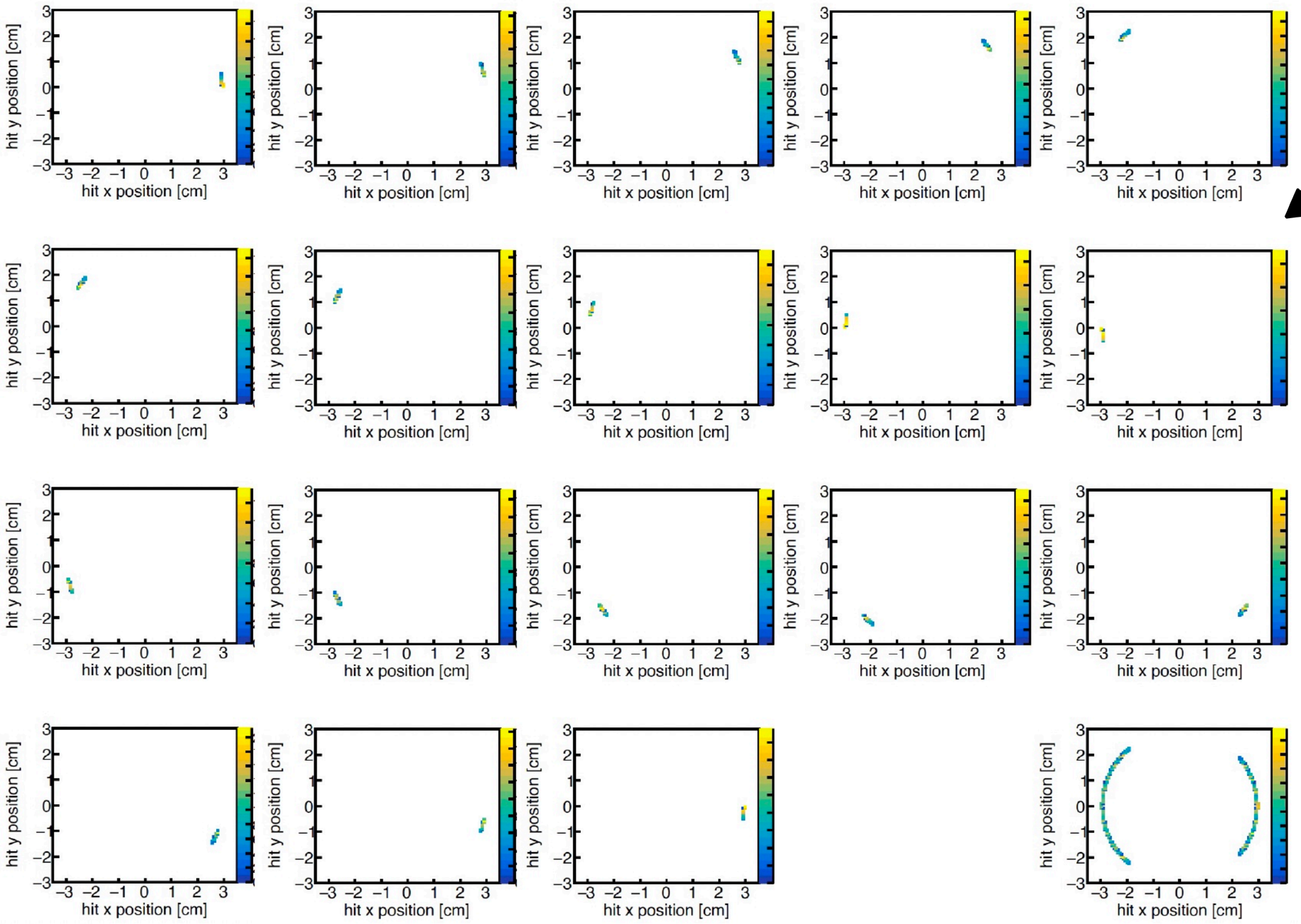
- "flipped" coordinate system (photons pointing in $+\hat{z}$)
- $18 \times 100k$ photons

Sample #2 (2022)

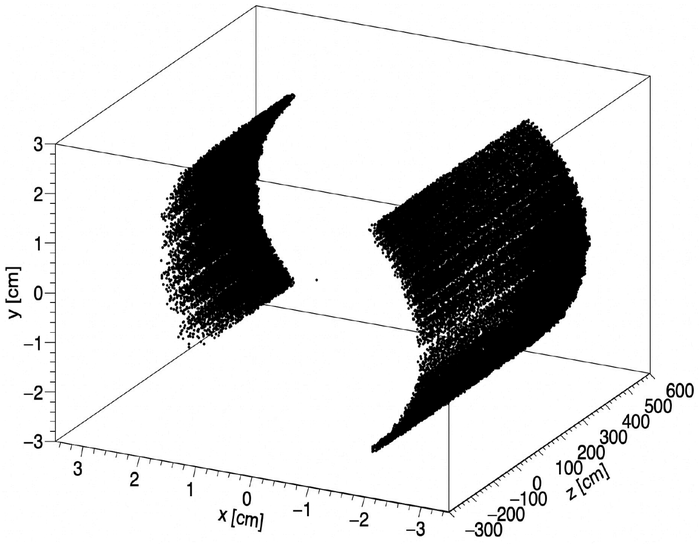
Files can he found here

- Right coordinate system
- $29 \times 1M$ photons
- Some features not well understood

Taking a peak at the data



Each plot corresponds to a different facet



Normalization

- Each facet contains N photons
- In order to get a photon flux (for a 2.5 mA electron beam of E = 10 GeV) each photon is weighed by:

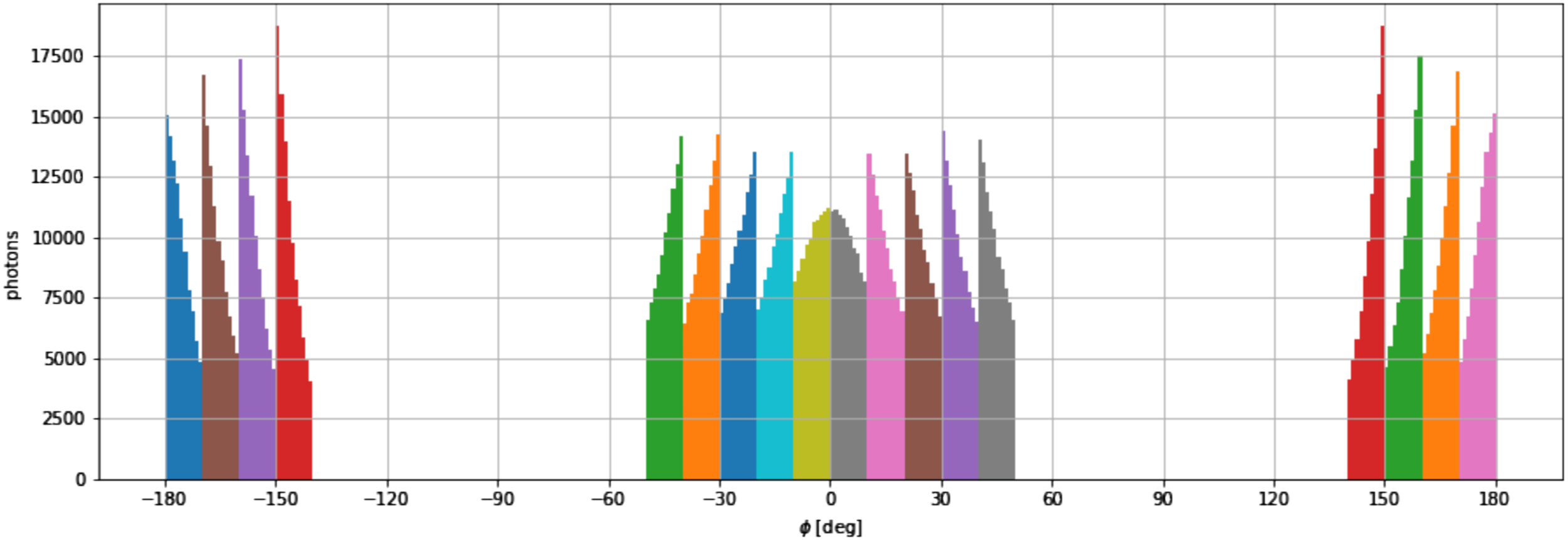
$$f_i = F\left[\frac{w_i}{\sum_{i=1}^{N} w_i}\right]$$

where w_i is a weight included in each facet file, $\sum_{i=1}^{N} w_i$ is the sum of the weights in a given facet file, and

F is a flux factor provided in the simulations. The resulting weight f_i should correspond to photons/s

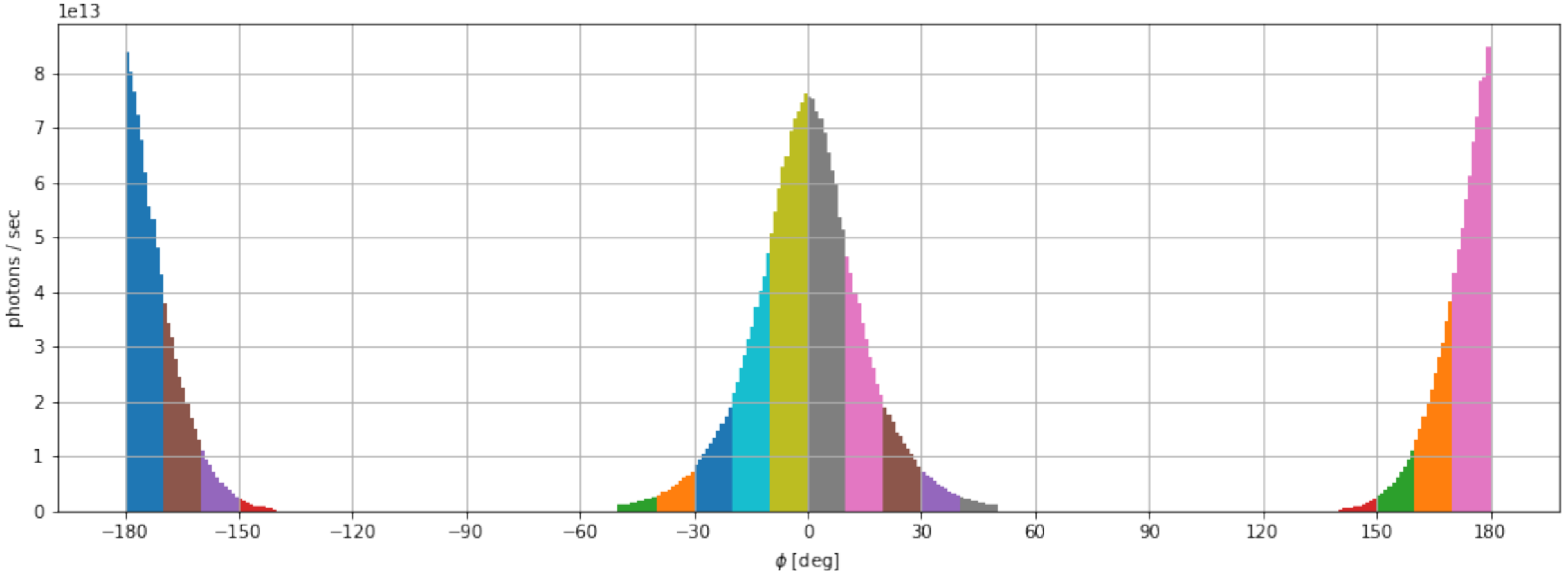
Phi distribution

Are we doing the normalization correctly?



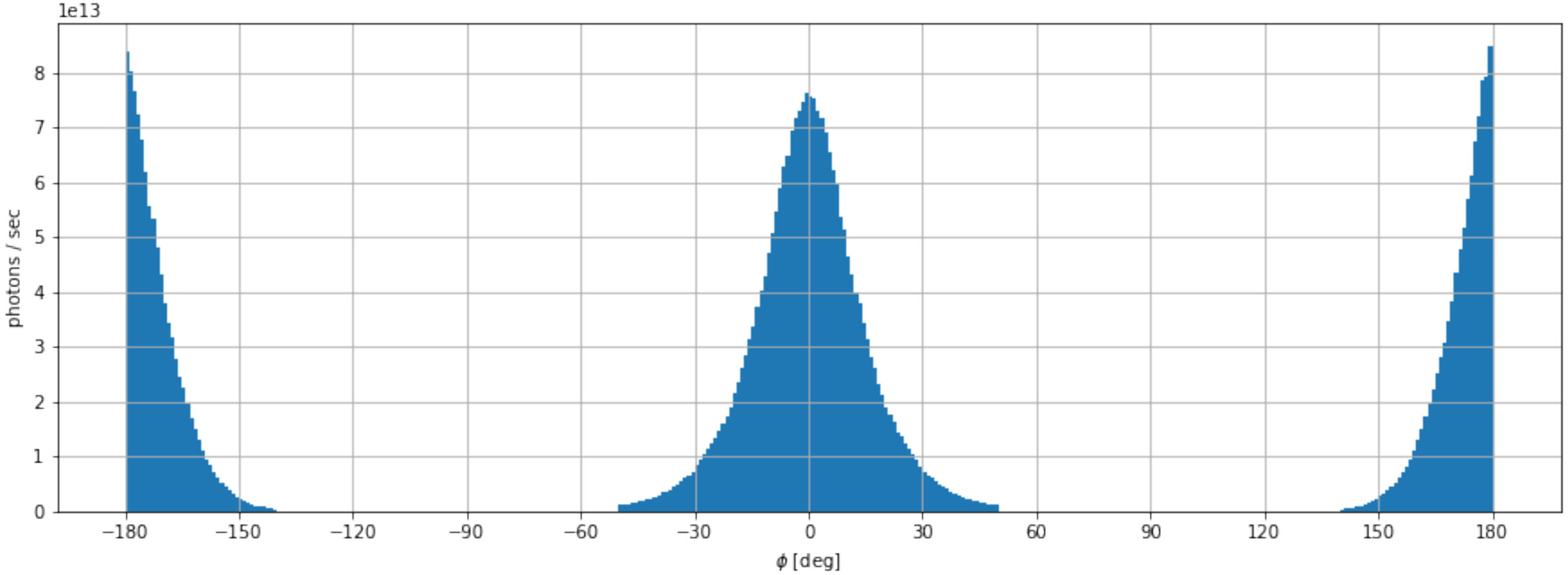
Phi distribution before normalization (equal total number of photons per facet file)

Phi distribution



Phi distribution after normalization

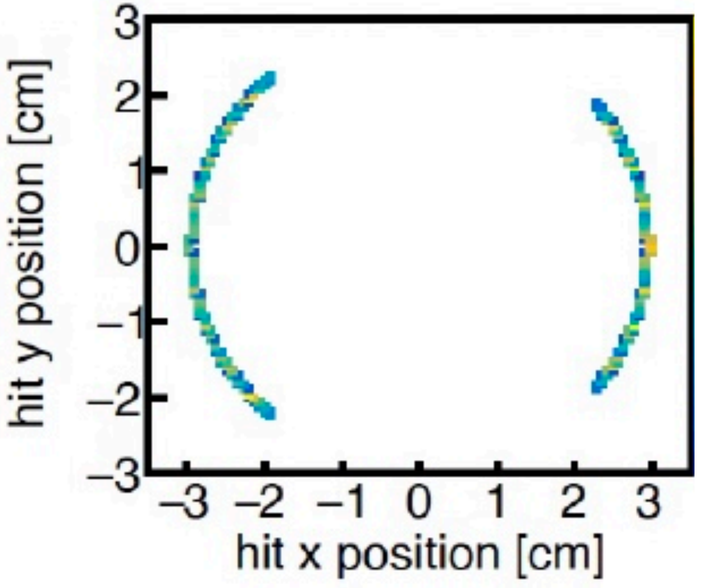
Phi distribution

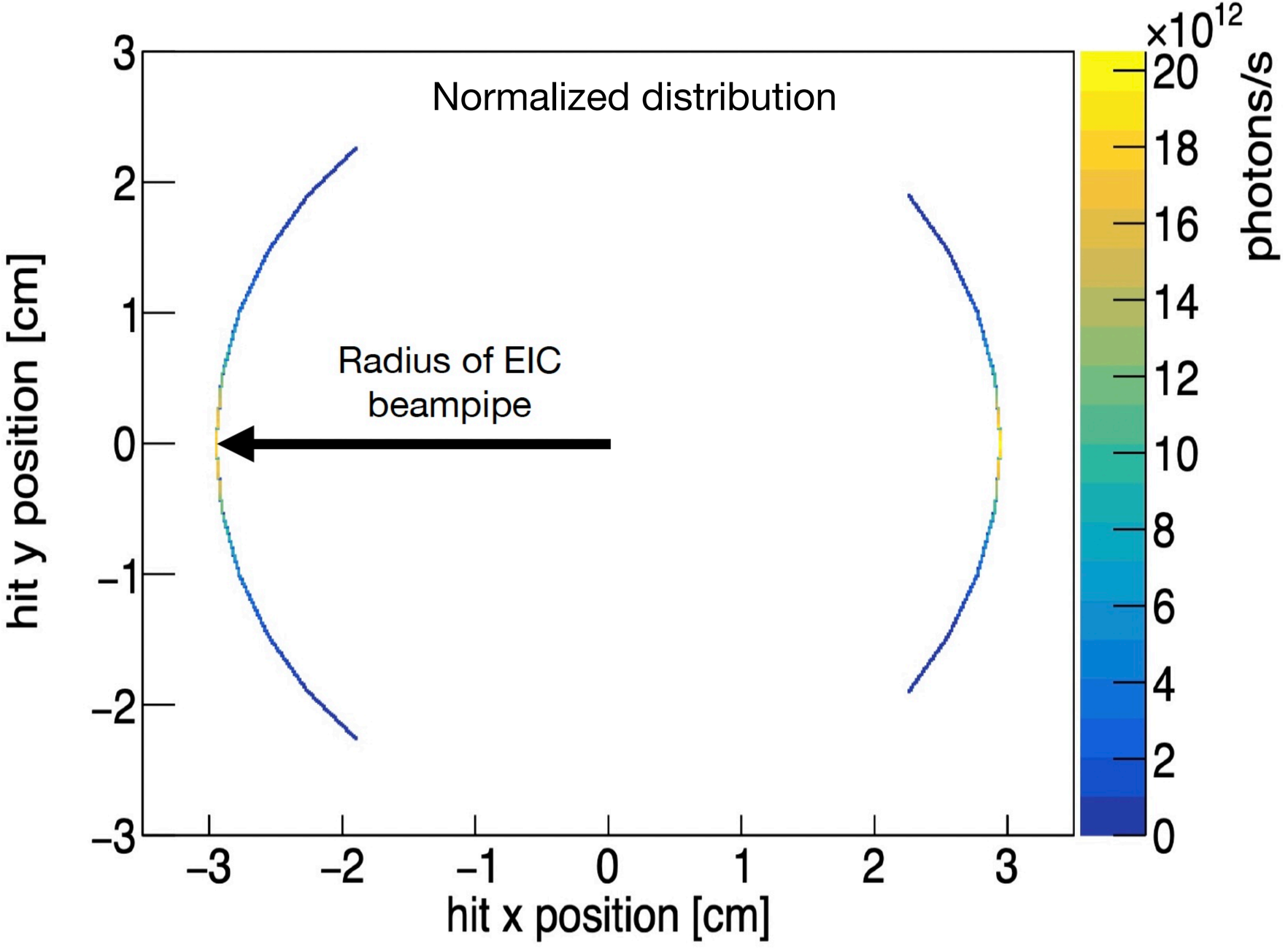


Phi distribution after normalization

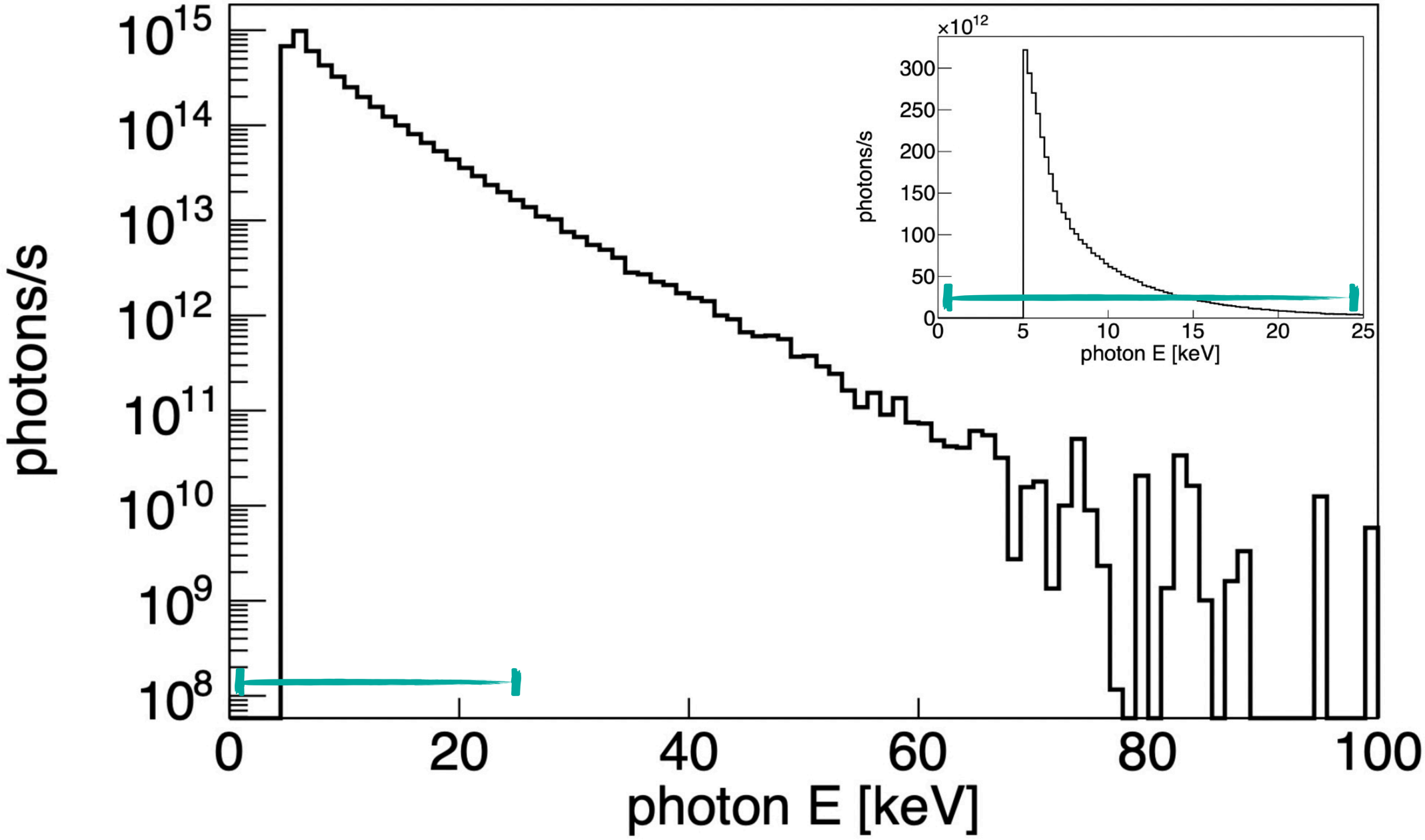
Normalized distribution

Not normalized distribution

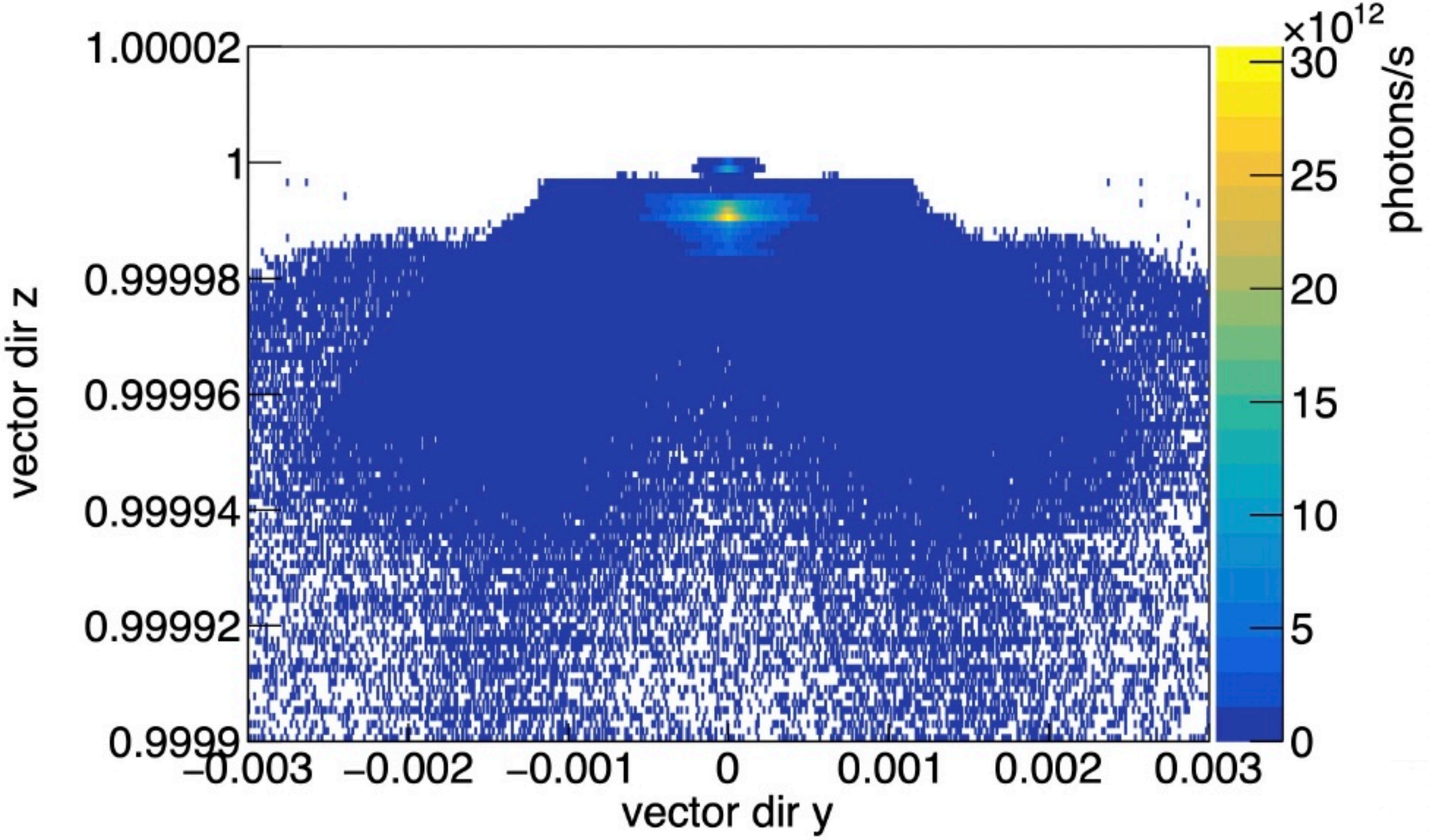




E spectrum


photons/s
10^{15}
10^{14}
10^{13}
10^{12}
10^{11}
10^{10}
10^{9}
10^{8}
0
20
40
60
80
100
photon E [keV]
$\times 10^{12}$
300
250
200
150
100
50
0
photons/s
0
5
10
15
20
25
photon E [keV]

Coordinate system



e- in these simulations

e- @ EIC

This needs to be accounted for at some point in the simulations

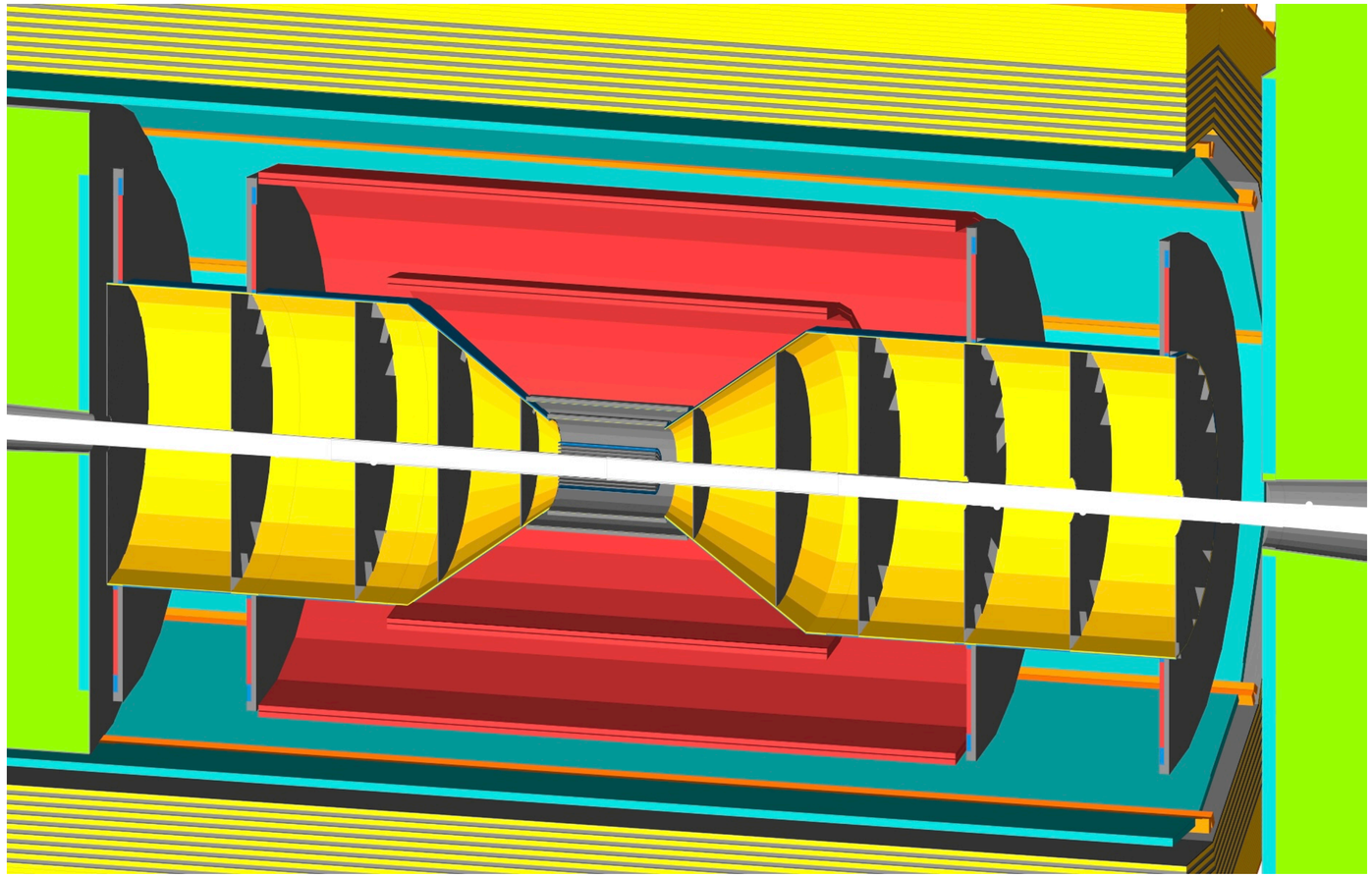
Simplest to address it from the get-go (translate hepmc files into rotated ones)

Outline

- (Simulated) data structure
- Initial studies
- From 1 photon to an event

Propagating photons through Geant

Propagate each individual photon through the GEANT detector and normalize each resulting hit by the photon weight



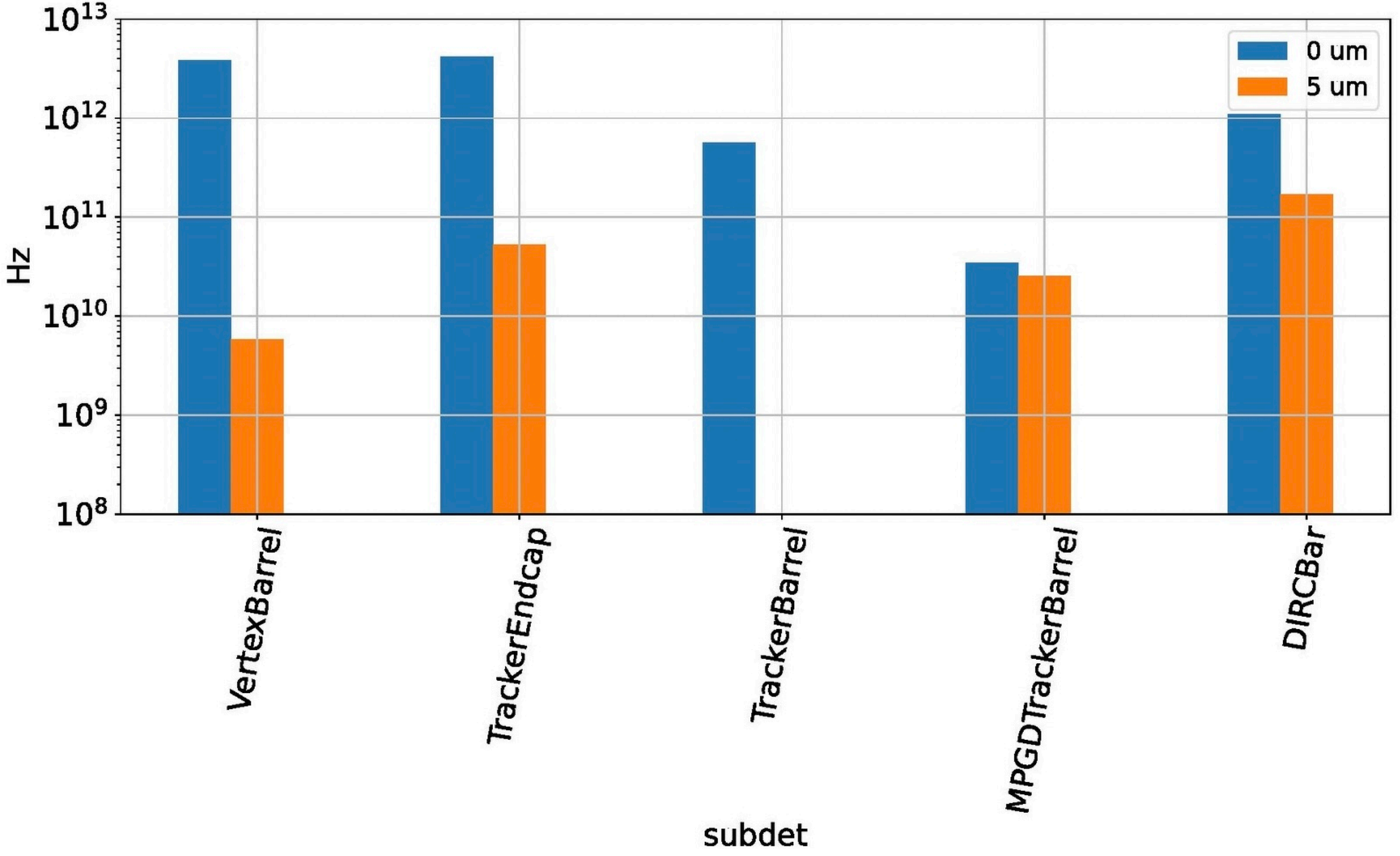
Using the ATHENA detector in DD4HEP

```
npsim
--runType batch
--numberOfEvents Nphotons
--compactFile ${DETECTOR_PATH}/
${JUGGLER_DETECTOR}.xml
--inputFiles path/to/input.hepmc
—outputFile path/to/output.root
```

Run this command for each facet file, and then merge the output

No reconstruction, just Geant propagation. This produces an output root file with hit information.

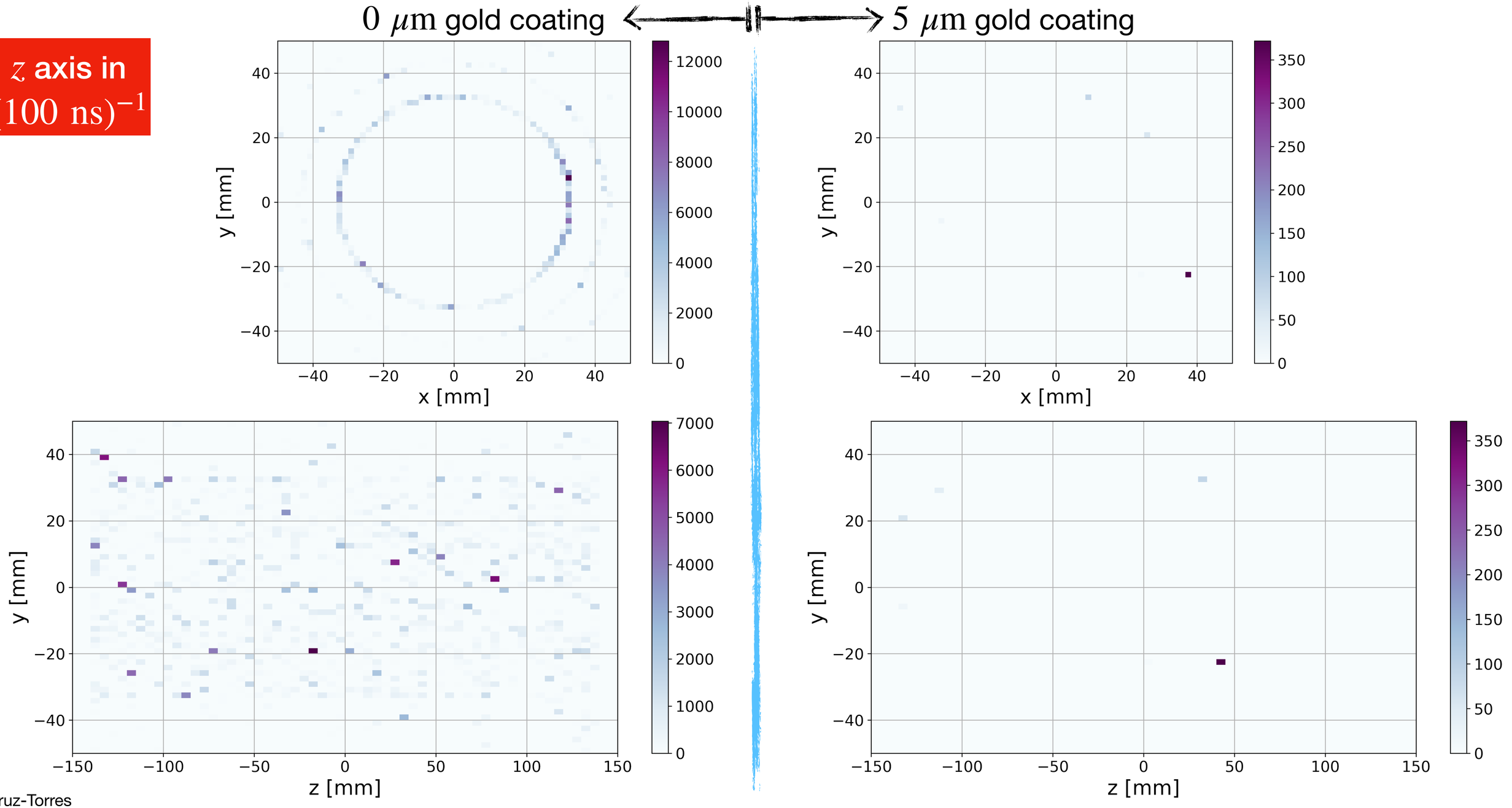
Rate from Synchrotron Radiation



Hit Distribution in Vertexing Layers

z axis in $(100\ \text{ns})^{-1}$

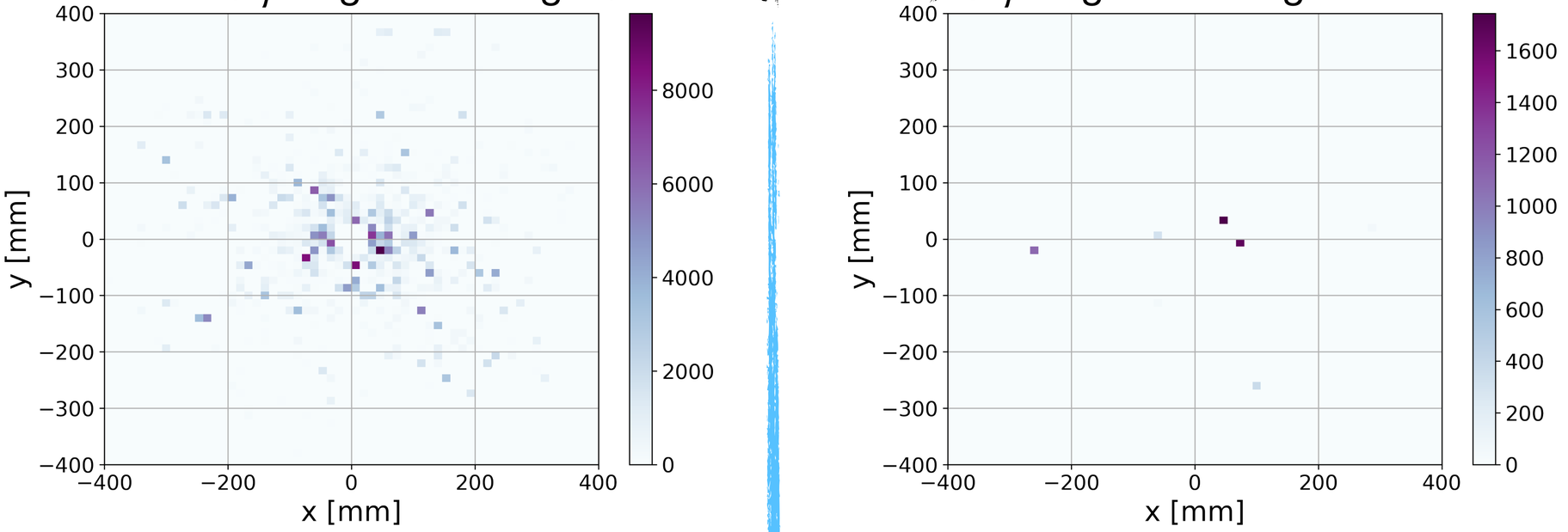
0 μm gold coating ⟷ 5 μm gold coating

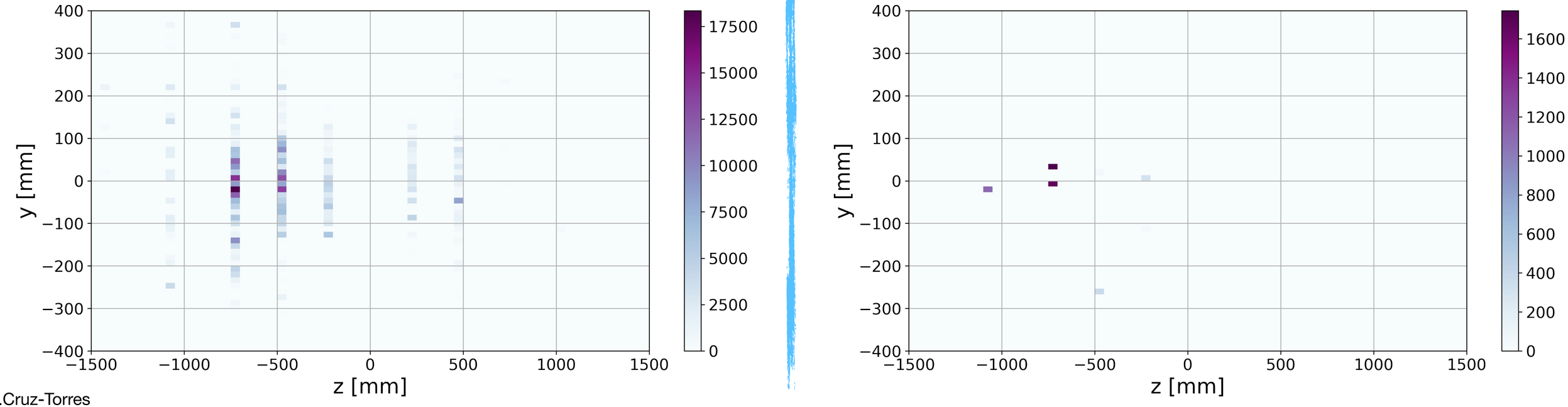


Hit Distribution in Tracker Endcap Layers

z axis in $(100\ \text{ns})^{-1}$

$0\ \mu$m gold coating ⟷ $5\ \mu$m gold coating



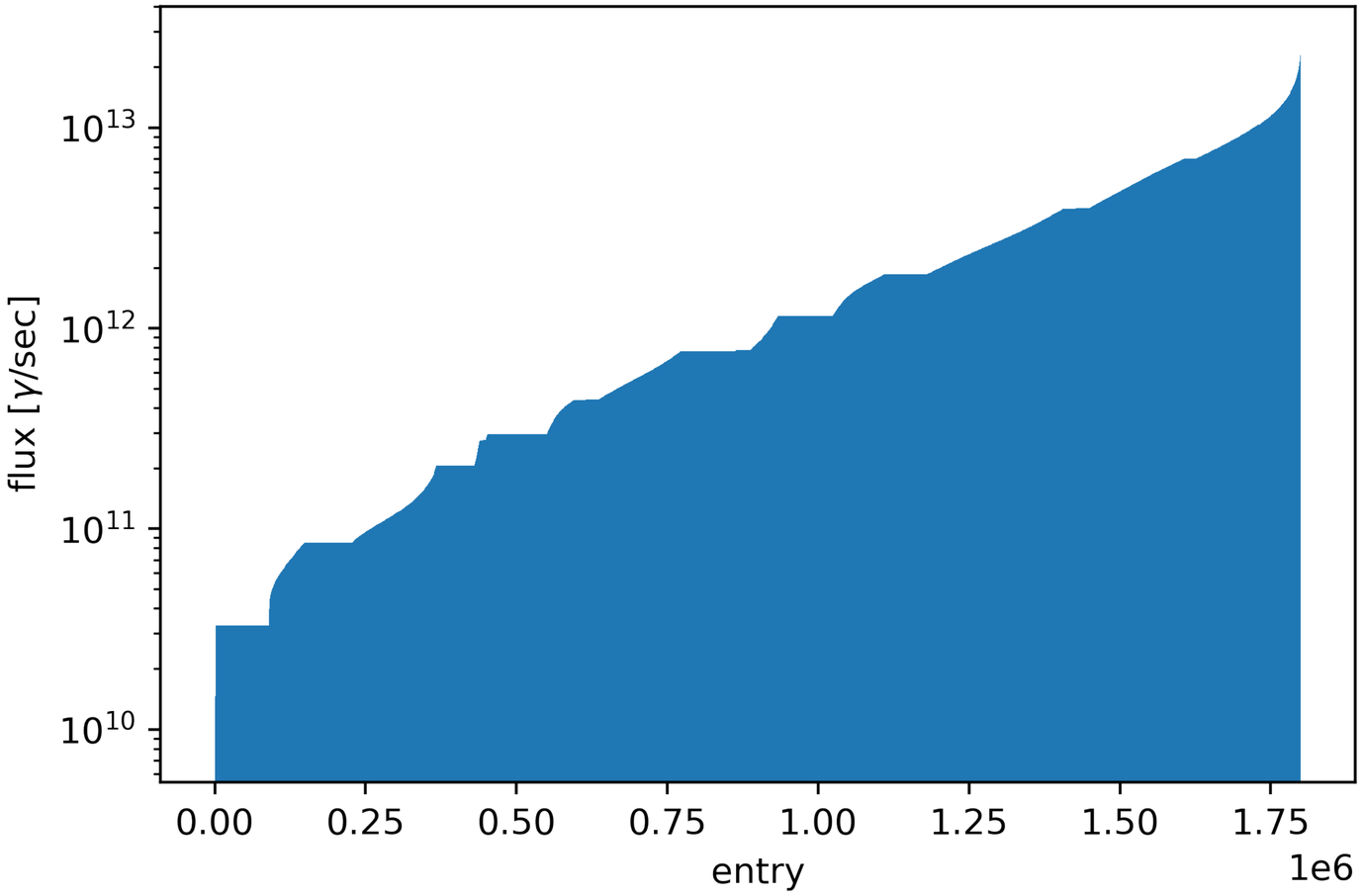


Outline

- (Simulated) data structure
- Initial studies
- From 1 photon to an event

Generator

Sample consists of 1.8M photons, each with a flux (γ/sec) weight



x-axis: every photon in the sample (1 per bin)

Define an integration window (IW)

integral = 0

while integral < IW:

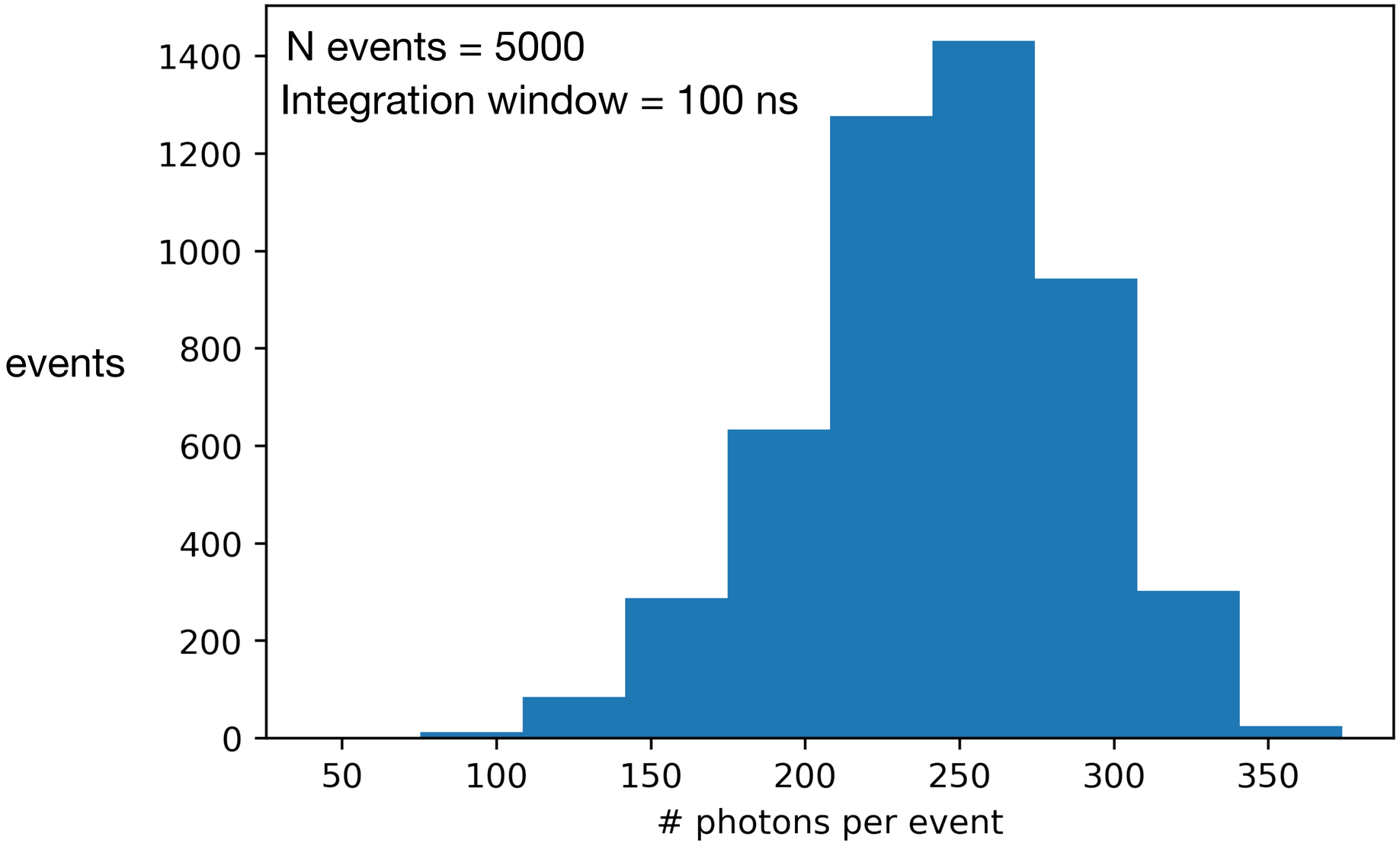
    Randomly sample photon, add it to event

    integral += 1/flux

return event

```
def generate_an_event(integration_window):
    event = []
    integrated_so_far = 0.
    while integrated_so_far < integration_window:
        x = h1_df.FindBin(h1_df.GetRandom())
        if x >= 1800000:
            continue
        photon = df.iloc[x]
        integrated_so_far += 1./photon['NormFact']
        event.append(photon)
    return event
```

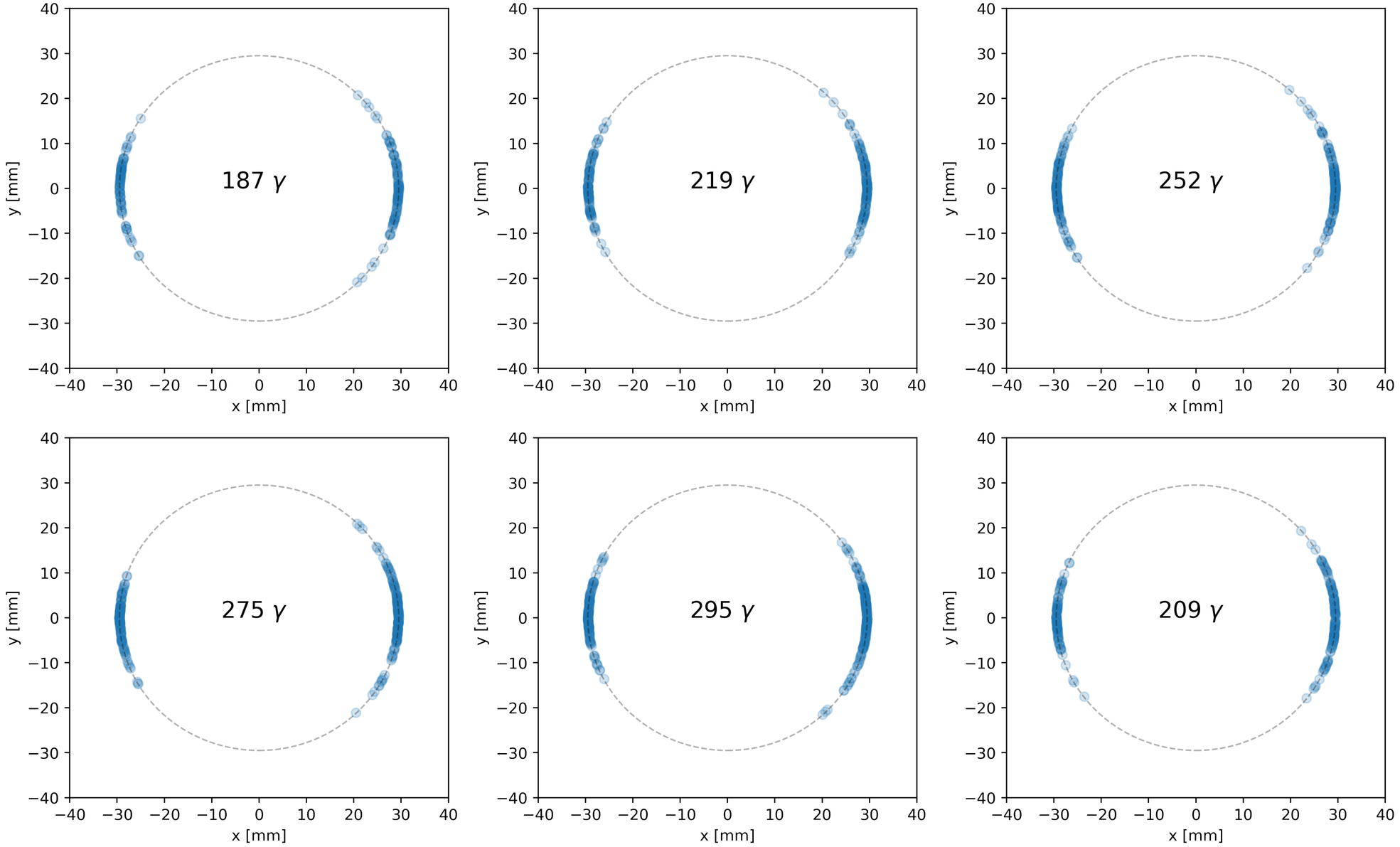
Resulting number of photons per event



before interaction with gold coating

For each photon in the event we have: p_x, p_y, p_z, x, y, z

Sample synchrotron-radiation background events



before interaction with gold coating

Integration window = 100 ns

Output data

Storing results in hepmc format using the pyhepmc-ng package

Examples in: https://github.com/scikit-hep/pyhepmc/tree/master/tests

```
import pyhepmc_ng as hep
```

```
for i in range(n_events):
    event = generate_an_event(integration_window)

    evt = hep.GenEvent(hep.Units.GEV, hep.Units.MM)
    evt.event_number = i+1
```

Generate new synchrotron-radiation event

Generate new hepmc event “container”

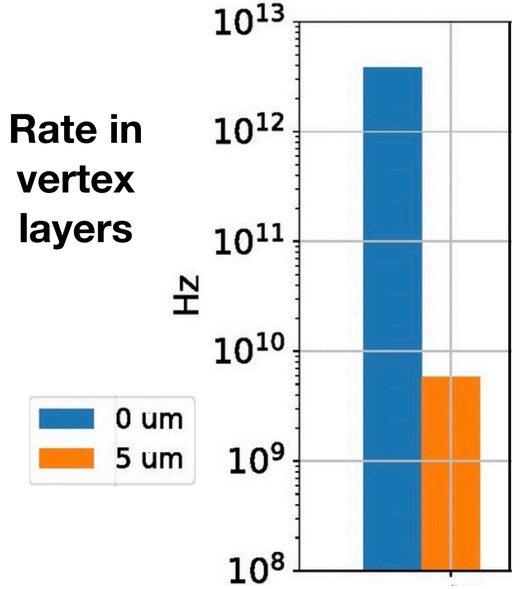
```
# loop over each photon in the event
for g in range(len(event)):

    # hep.GenParticle((px,py,pz,E),pdgid,status)
    p1 = hep.GenParticle((px,py,pz,E),22,g+1)
    p1.generated_mass = 0.
    evt.add_particle(p1)
    # make sure vertex is not optimized away by WriterAscii
    v1 = hep.GenVertex((x,y,z,0));
    v1.add_particle_out(p1)
    evt.add_vertex(v1)
```

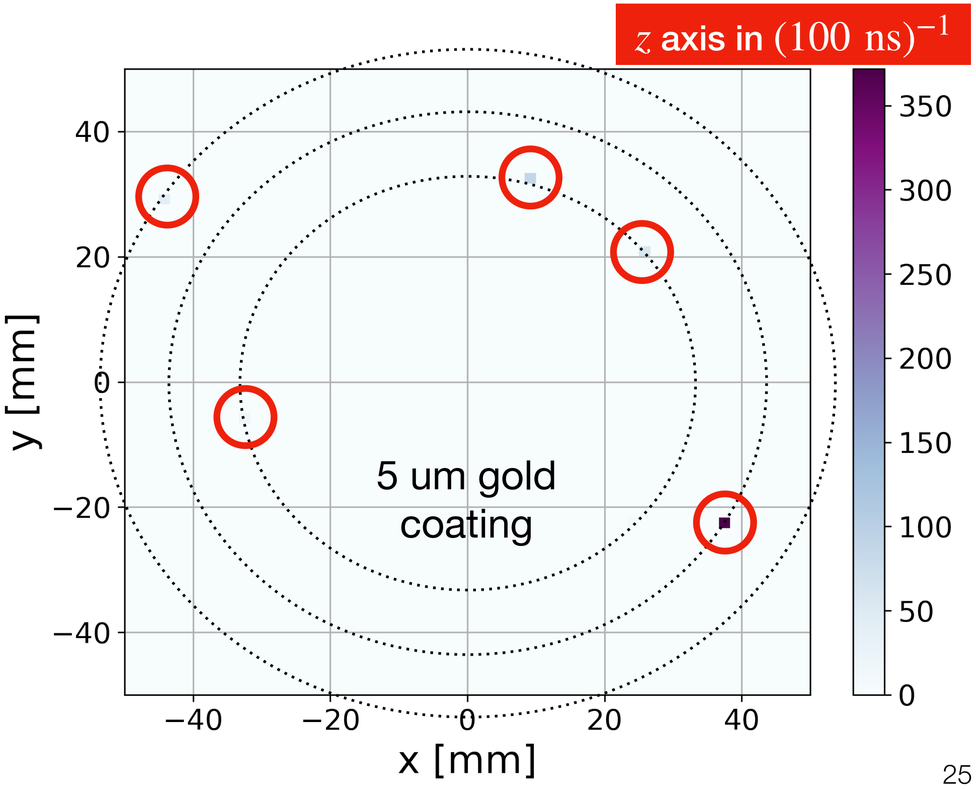
Add photons to hepmc

Add vertices to hepmc

Summary 1/2

- Rotated photons from Synrad+ hepmc files to the correct EIC coordinate system
- Propagated these photons through the Geant4 simulation of the ATHENA detector (in DD4HEP)
- Normalized the resulting hits to get hit rates
- Few hits with high weights cause seemingly high synchrotron-radiation rates, but with high uncertainty
- Files with more statistics seemingly have some unexpected features





Summary 2/2

- Wrote synchrotron-radiation background event generator based on the available Synrad+ simulations

- These are soft raw photon events before traversing the beampipe material. Their interaction with the gold coating as well as energy thresholds will significantly reduce the number of photons creating signal in the detectors

- Next steps: embed simulated “signals” into these background events and try, e.g. track finding

- Updates in beamline magnets -> somewhat different synchrotron radiation profile. New simulations are currently being run.

Backup

New Higher Statistics Files

Work done with UC Berkeley undergrad Benjamen Sterwerf