Muon efficiency & ID in the nHCal

June 17, 2025

Caroline Riedl, Kira Richards (UIUC)

Muons in the nHCal

- **Goals**: muon efficiency and ID in the nHCal
- **Tools**: particle gun submission framework provided by Sasha Prozorov,
  - https://github.com/aprozo/epic_full_chain_batch
  - https://chat.epic-eic.org/main/pl/c319sepsqbg5ibgetxpzk686ia
- **Status and issues** as of June 17, 2025:
  - Can run Sasha's framework and produce simulated data
  - MCParticle - Cluster matching issue resolved as of June 6, 2025, https://github.com/eic/EICrecon/issues/1908, and should be active in the central productions starting from starting from the 25.06.0 container
  - Exploring clusters and hits in older productions and new particle gun data

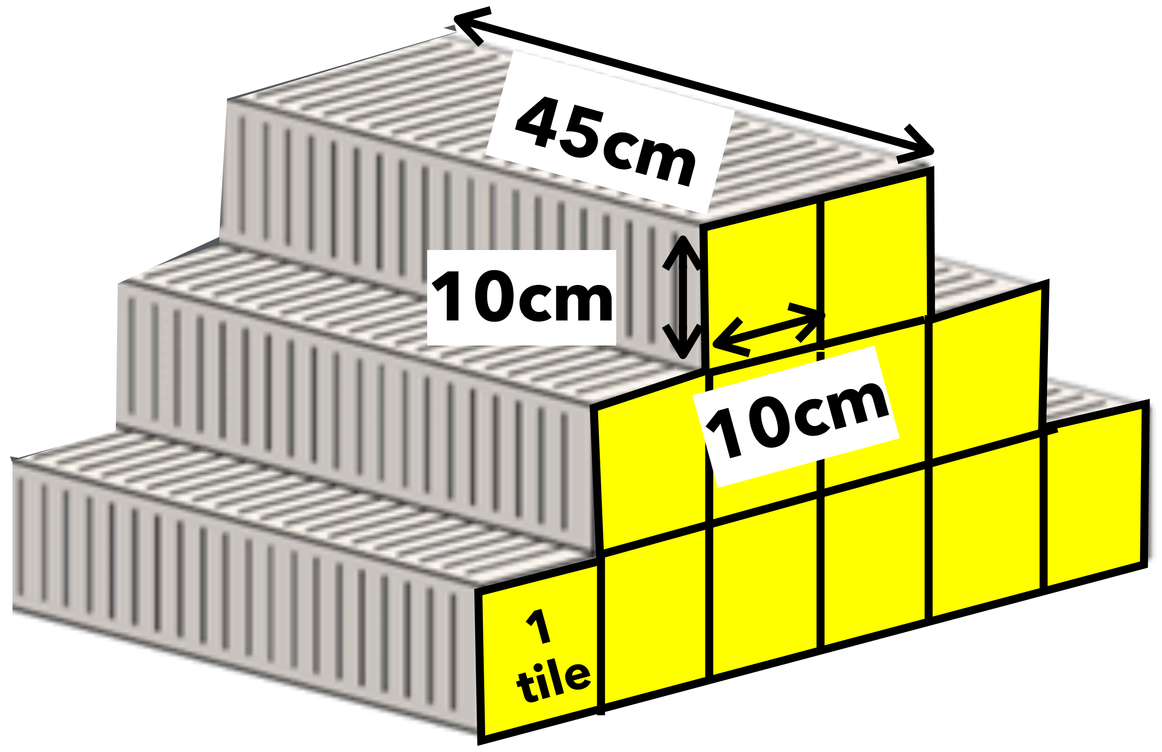
Caroline’s experience with Sasha’s framework

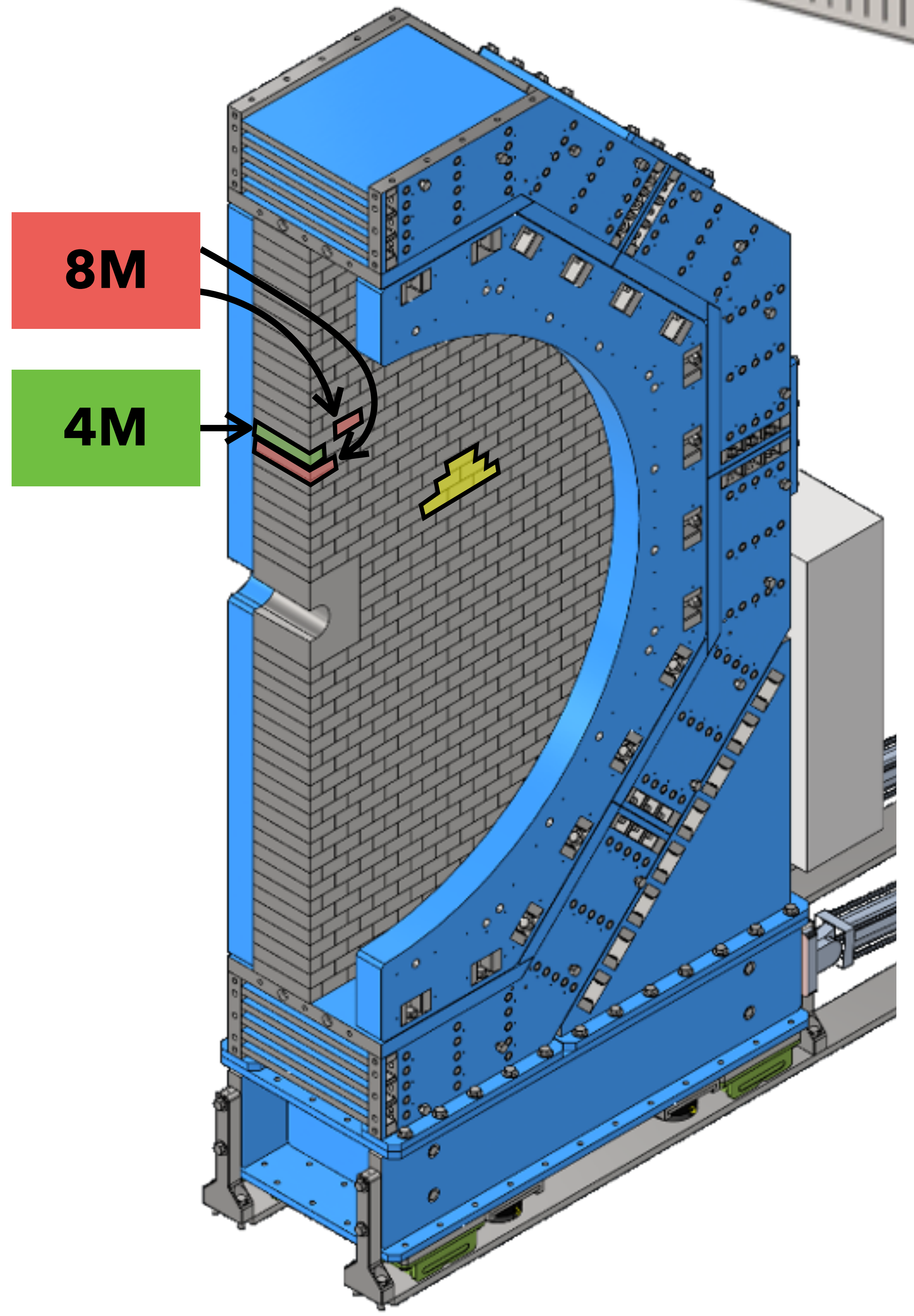
- cd /eic/u/criedl/eic
- git clone https://github.com/aprozo/epic_full_chain_batch
- cd epic_full_chain_batch
- Start the script from within the Linux shell (not in eic-shell!)
- Runs ddsim with nHCal-only geometry (epic_backward_hcal_only.xml)
  - env TILE_SIZE=5 N_LAYERS=11 SCINTILLATOR_THICKNESS=0.8 ABSORBER_THICKNESS=3 MOMENTUM=1 ./run_submit.sh
    - Produces 10 files *edm4hep.root [DEFAULT_JOBS=10]
    - Produces 10 files *edm4eic.root [often fewer files - some fail]
    - Output is written into /gpfs02/eic/criedl - is not backed up, use only for data (not code)
- Analyse the *edm4eic.root files using https://github.com/criedl77/nHCal_VM

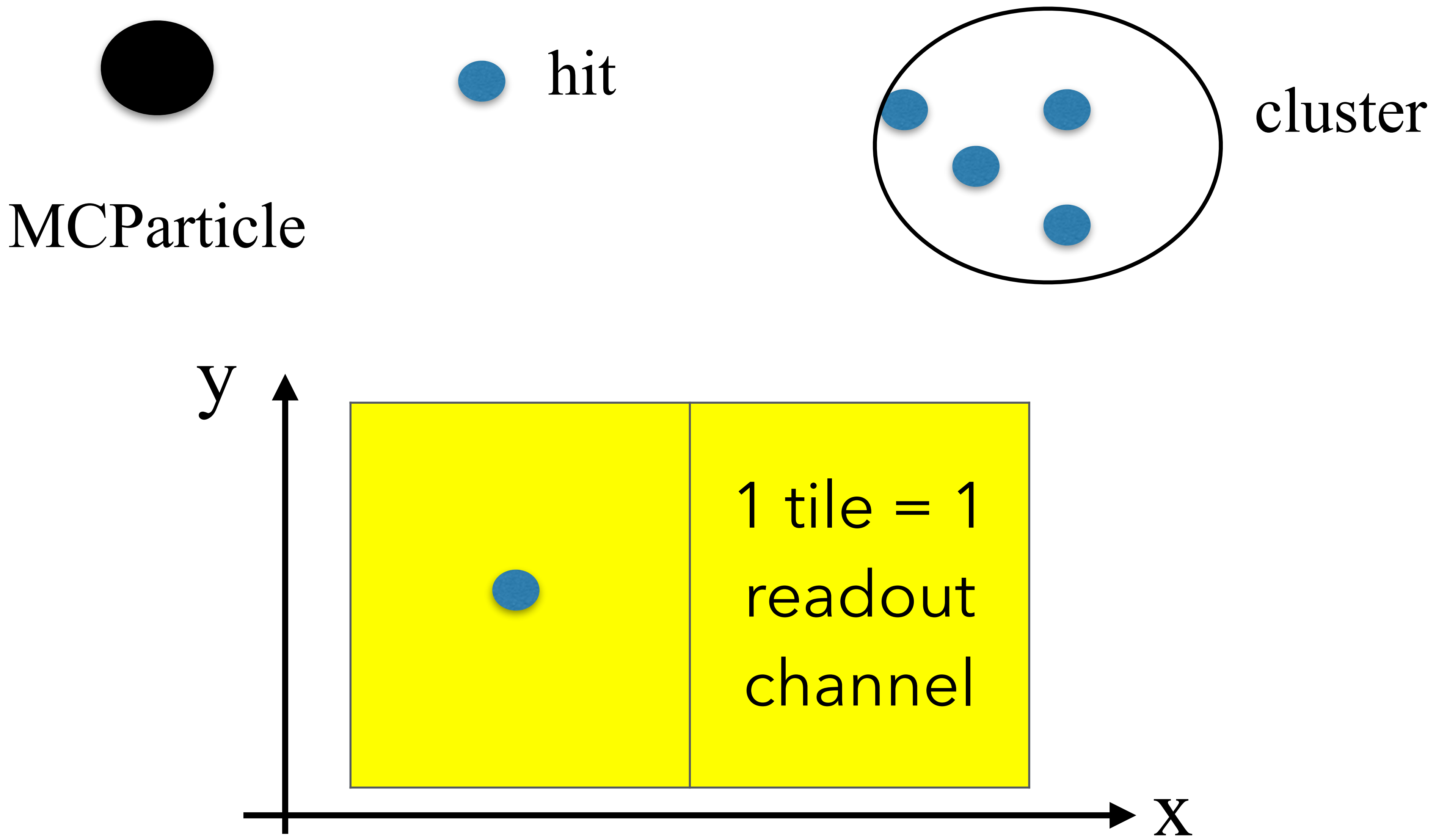
Data

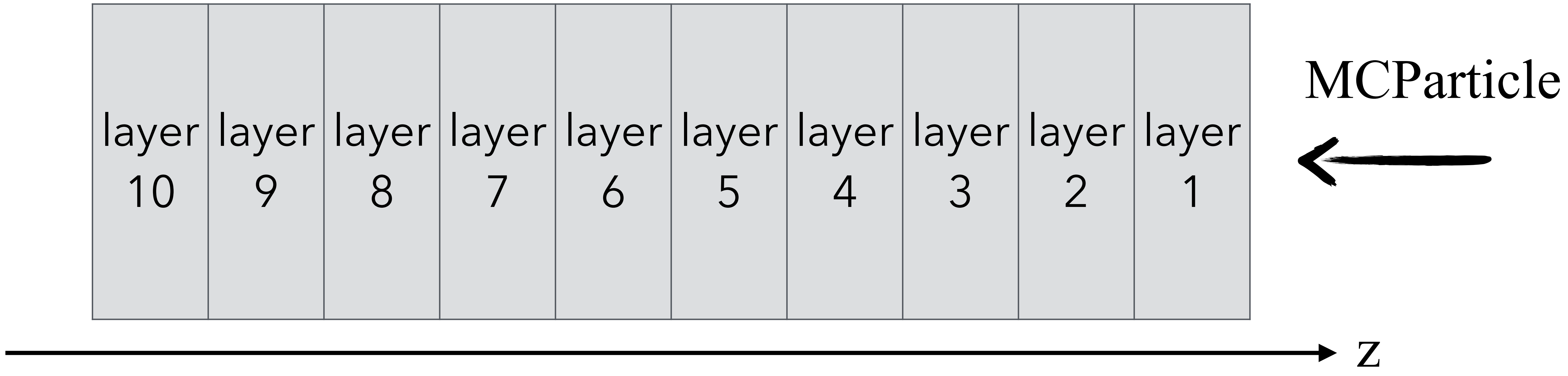
- My particle gun data are located here:
  - /gpfs02/eic/criedl/output/nhcal_only_*
- To develop my code, I am using these older productions for which the cluster associations work:
  - /volatile/eic/EPIC/RECO/24.08.1/epic_craterlake/DIS/NC/18x275/minQ2=1
  - /volatile/eic/EPIC/RECO/24.08.1/epic_craterlake/EXCLUSIVE/DIFFRACTIVE_PHI_ABCONV/Sartre/Coherent

default design:
10 layers of each
4 cm steel +
4 mm scintillator

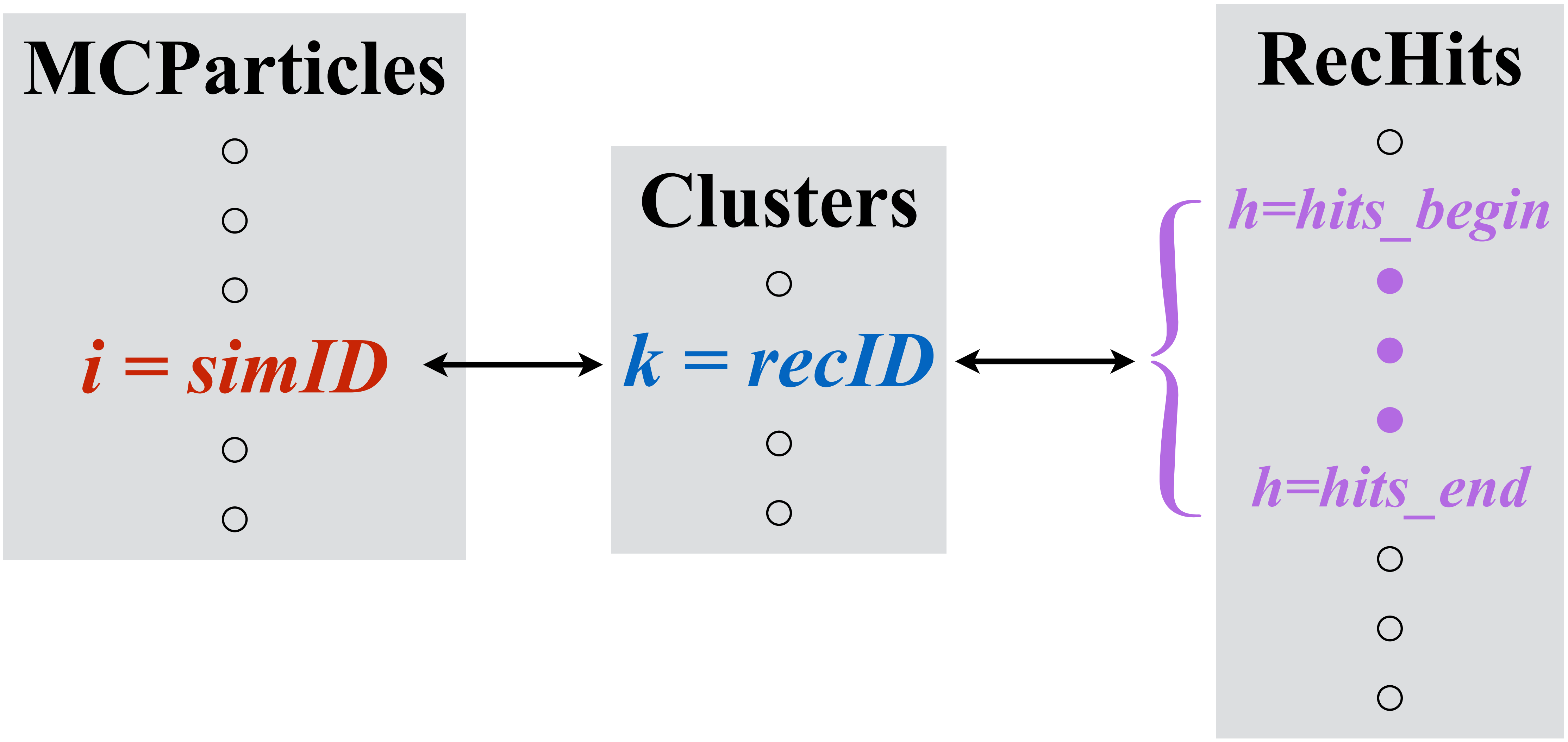








MCParticles, Clusters and Hits in EICrecon



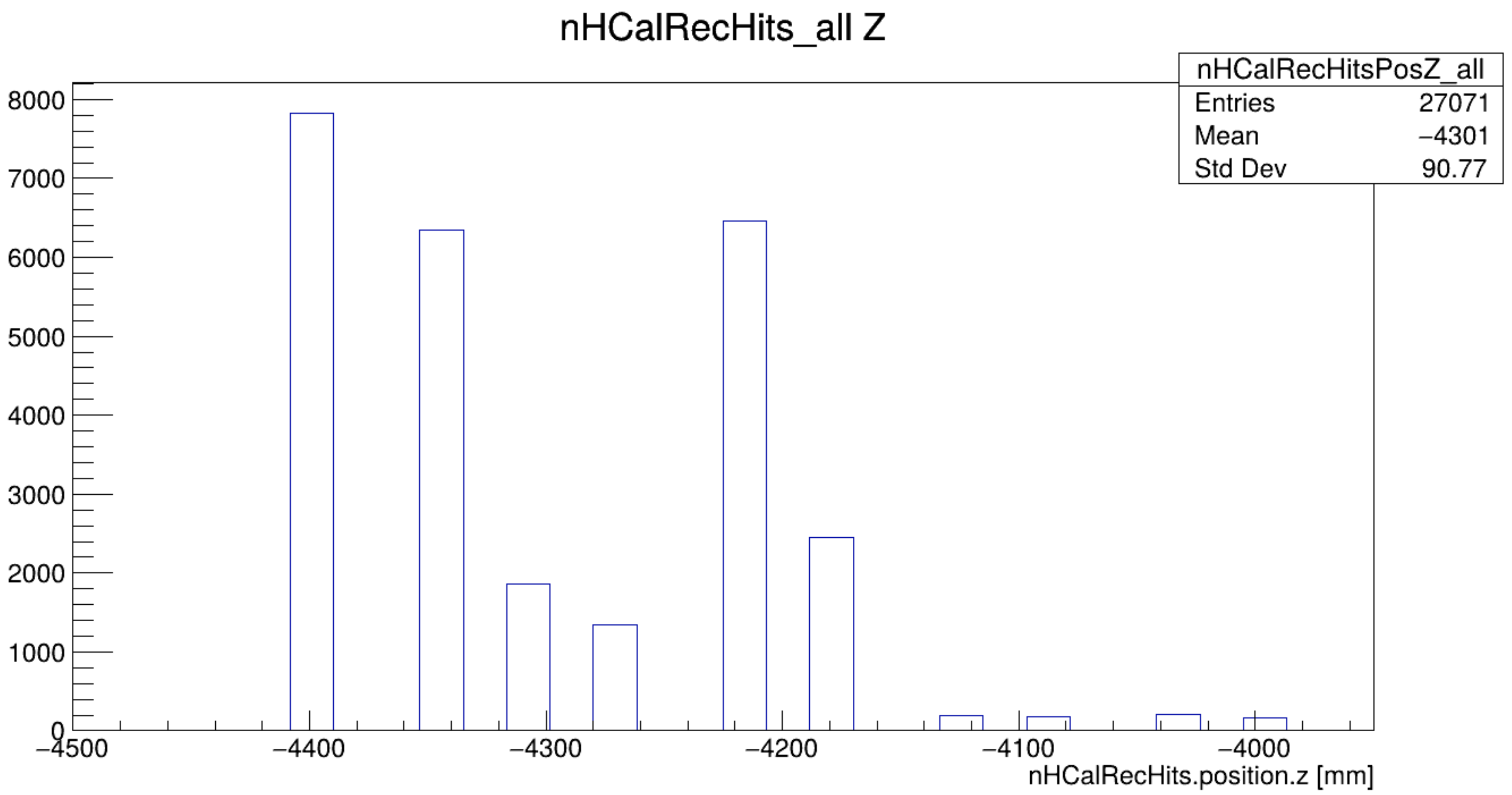
MCParticles, Clusters and Hits

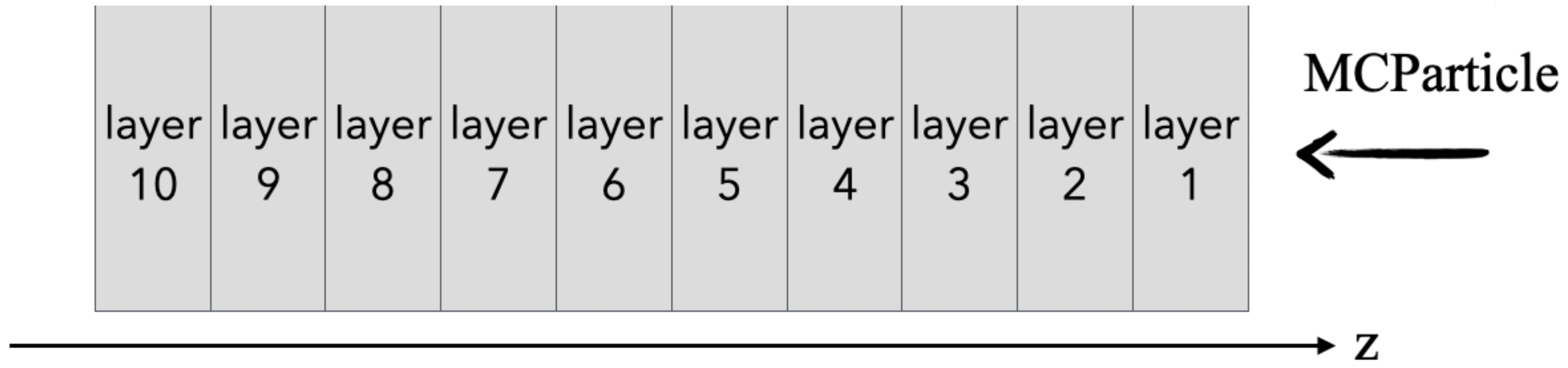
```
// Get generated particle information (after GEANT):
  TTreeReaderArray<int>    partGenStat(tree_reader, "MCParticles.generatorStatus");
  TTreeReaderArray<double> partMomX(tree_reader, "MCParticles.momentum.x");
  TTreeReaderArray<double> partMomY(tree_reader, "MCParticles.momentum.y");
  TTreeReaderArray<double> partMomZ(tree_reader, "MCParticles.momentum.z");
  TTreeReaderArray<int>    partPdg(tree_reader, “MCParticles.PDG");

// nHCal clusters: fParentName edm4eic::ClusterData
  TTreeReaderArray<unsigned int> nHCalClustersnHits(tree_reader, "HcalEndcapNClusters.nhits");
  TTreeReaderArray<float> nHCalClustersE(tree_reader, "HcalEndcapNClusters.energy");
  TTreeReaderArray<float> nHCalClustersPosX(tree_reader, "HcalEndcapNClusters.position.x");
  TTreeReaderArray<float> nHCalClustersPosY(tree_reader, "HcalEndcapNClusters.position.y");
  TTreeReaderArray<float> nHCalClustersPosZ(tree_reader, “HcalEndcapNClusters.position.z");

// Get associations between MCParticles and Clusters:
// edm4eic::MCRecoClusterParticleAssociation = "Association between a Cluster and a MCParticle"
// simID = index of corresponding MCParticle (position in MCParticles array)
// recID = index of corresponding Cluster (position in Clusters array)
  TTreeReaderArray<unsigned int> recoAssocClusters_nHCal(tree_reader, "HcalEndcapNClusterAssociations.recID");
  TTreeReaderArray<unsigned int> simuAssocClusters_nHCal(tree_reader, "HcalEndcapNClusterAssociations.simID");


// nHCal reconstructed hits: fParentName edm4eic::CalorimeterHitData
  TTreeReaderArray<float> nHCalRecHitsE(tree_reader, "HcalEndcapNRecHits.energy");
  TTreeReaderArray<float> nHCalRecHitsPosX(tree_reader, "HcalEndcapNRecHits.position.x");
  TTreeReaderArray<float> nHCalRecHitsPosY(tree_reader, "HcalEndcapNRecHits.position.y");
  TTreeReaderArray<float> nHCalRecHitsPosZ(tree_reader, "HcalEndcapNRecHits.position.z");

// Hit navigators for clusters and hits:
  TTreeReaderArray<unsigned int> nHCalClustershits_begin(tree_reader, "HcalEndcapNClusters.hits_begin");
  TTreeReaderArray<unsigned int> nHCalClustershits_end(tree_reader, "HcalEndcapNClusters.hits_end");
```

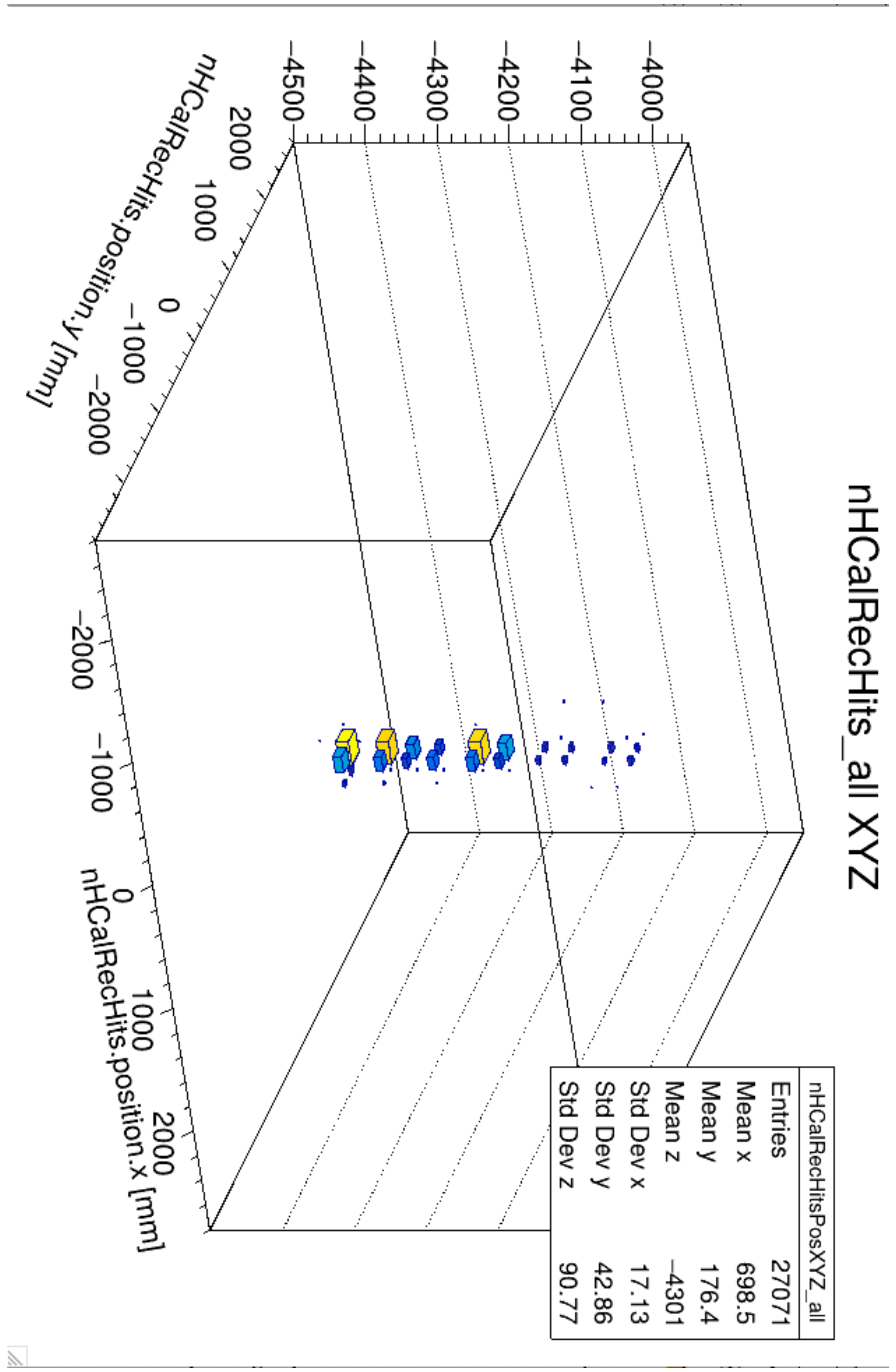
Particle Gun - hit level

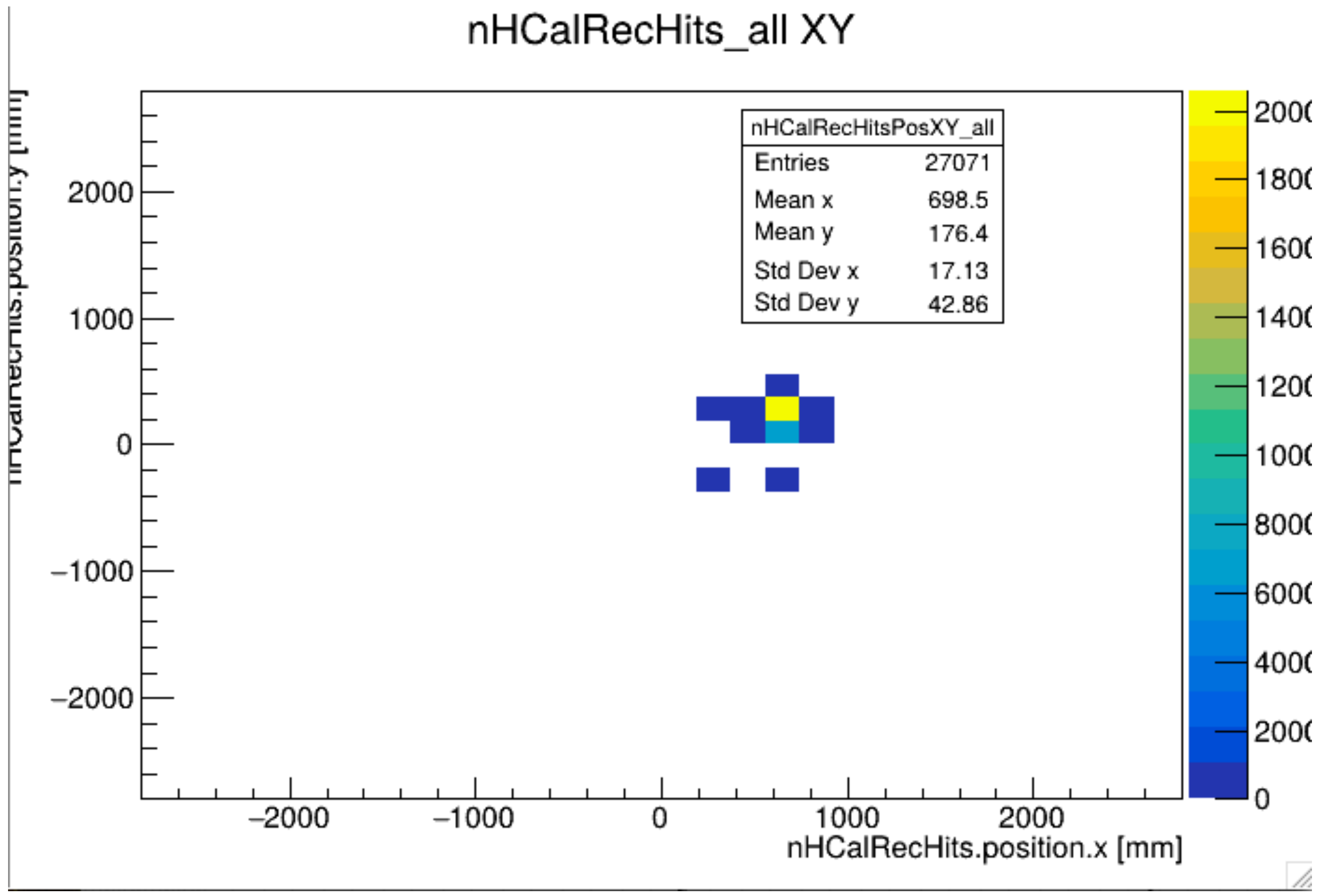
- env TILE_SIZE=5 N_LAYERS=10 SCINTILLATOR_THICKNESS=0.8 ABSORBER_THICKNESS=4 MOMENTUM=1 PHI=45 THETA=170 PARTICLE=mu- NUMBER_OF_EVENTS=1000 ./run_submit.sh

```
Number of generated electrons +-: 123
Number of generated muons +-: 8000
```



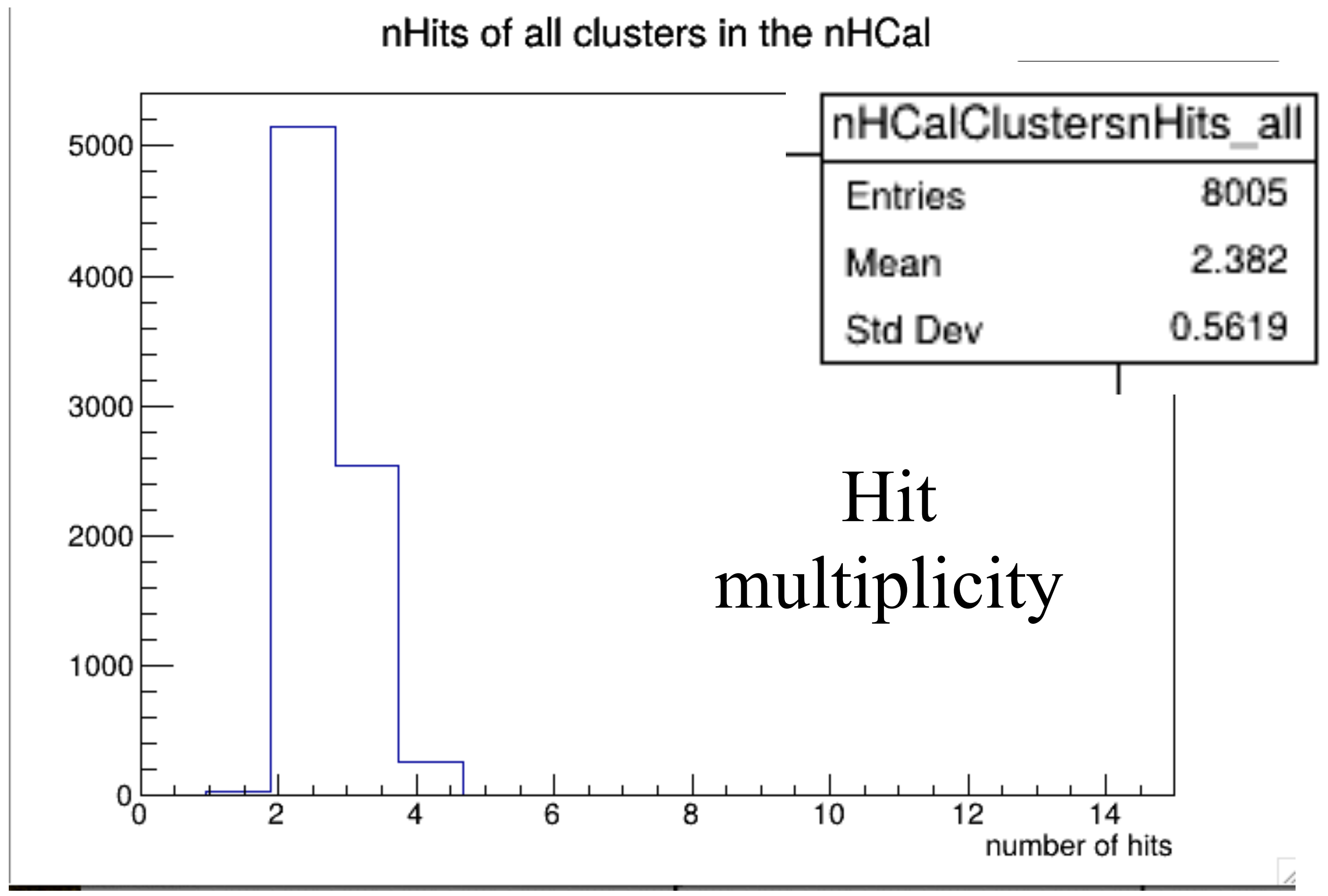


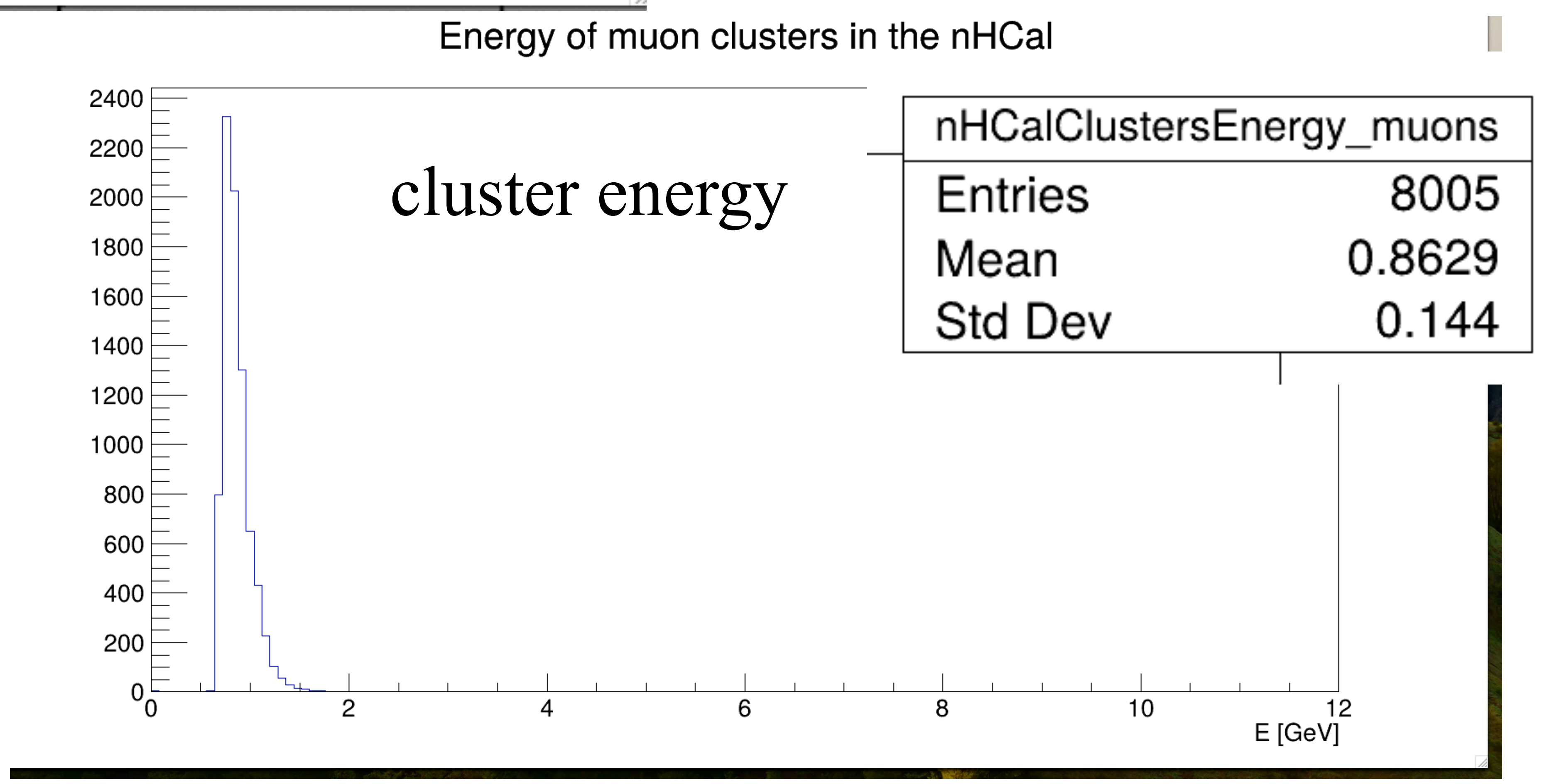




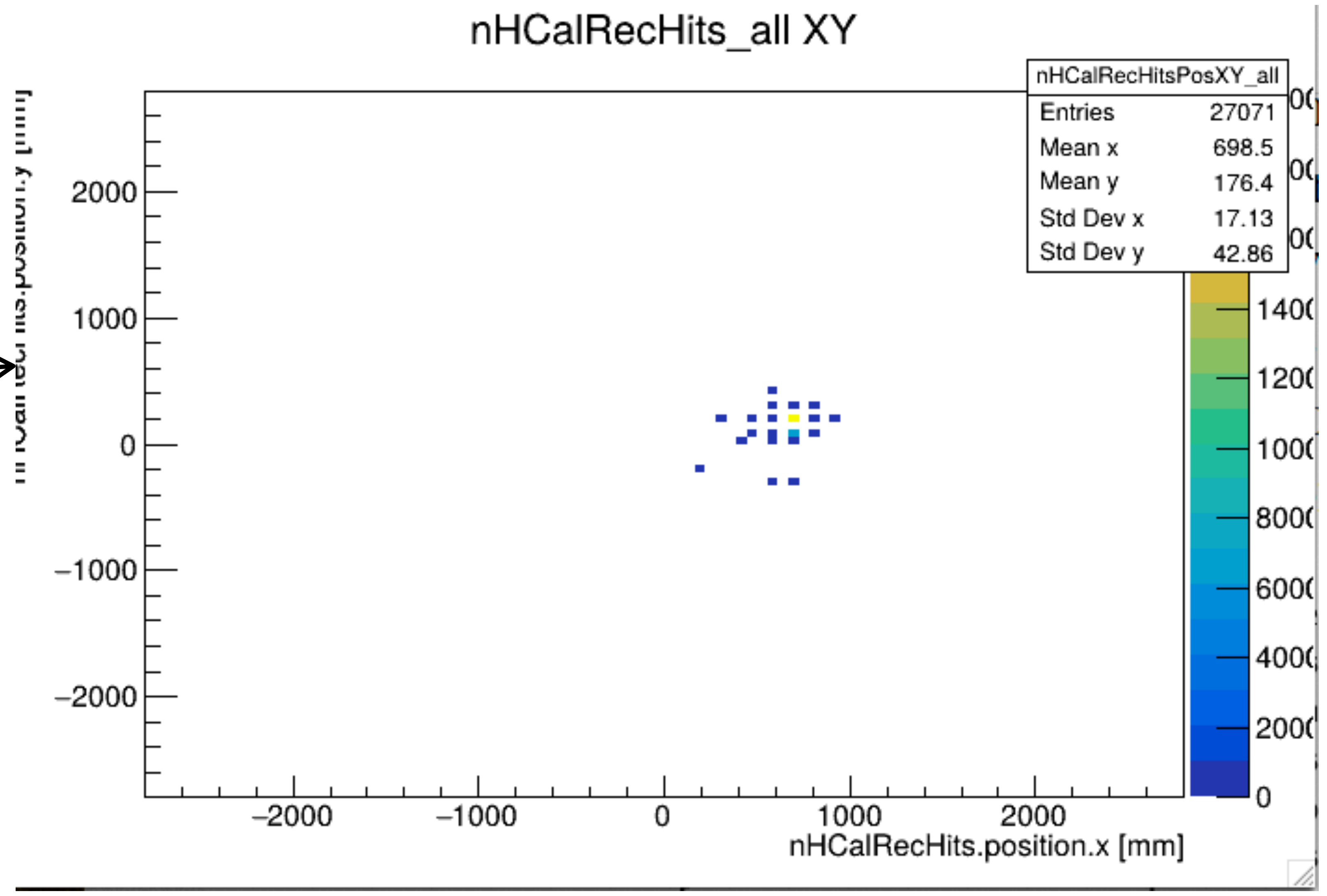
Particle Gun - from hits to clusters

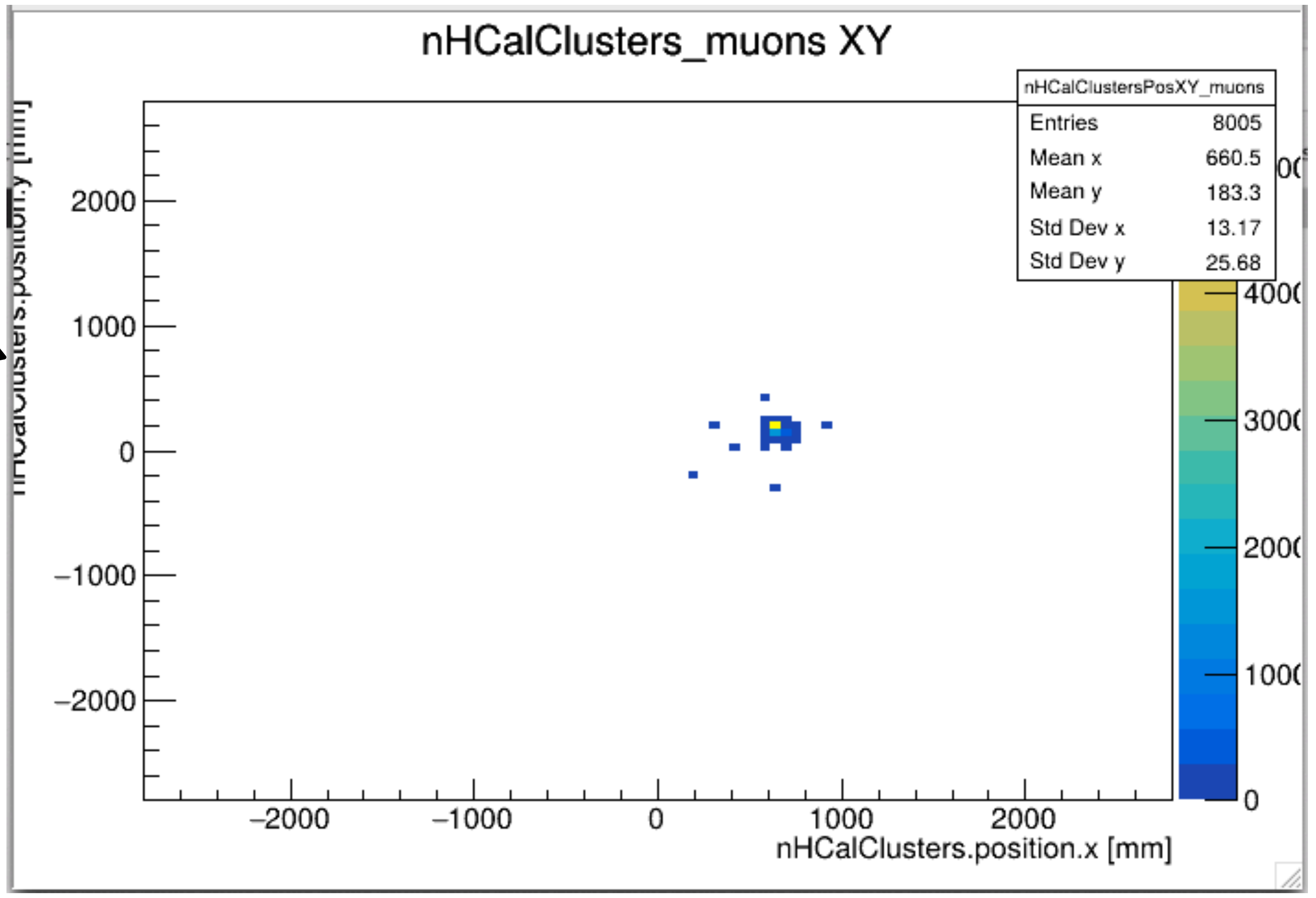
- env TILE_SIZE=5 N_LAYERS=10 SCINTILLATOR_THICKNESS=0.8 ABSORBER_THICKNESS=4 **MOMENTUM=1** PHI=45 THETA=170 PARTICLE=mu- NUMBER_OF_EVENTS=1000 ./run_submit.sh
- Mainly 2 or 3 hits make up one cluster





(x, y) distributions look reasonable at first sight
top: hits
bottom: clusters

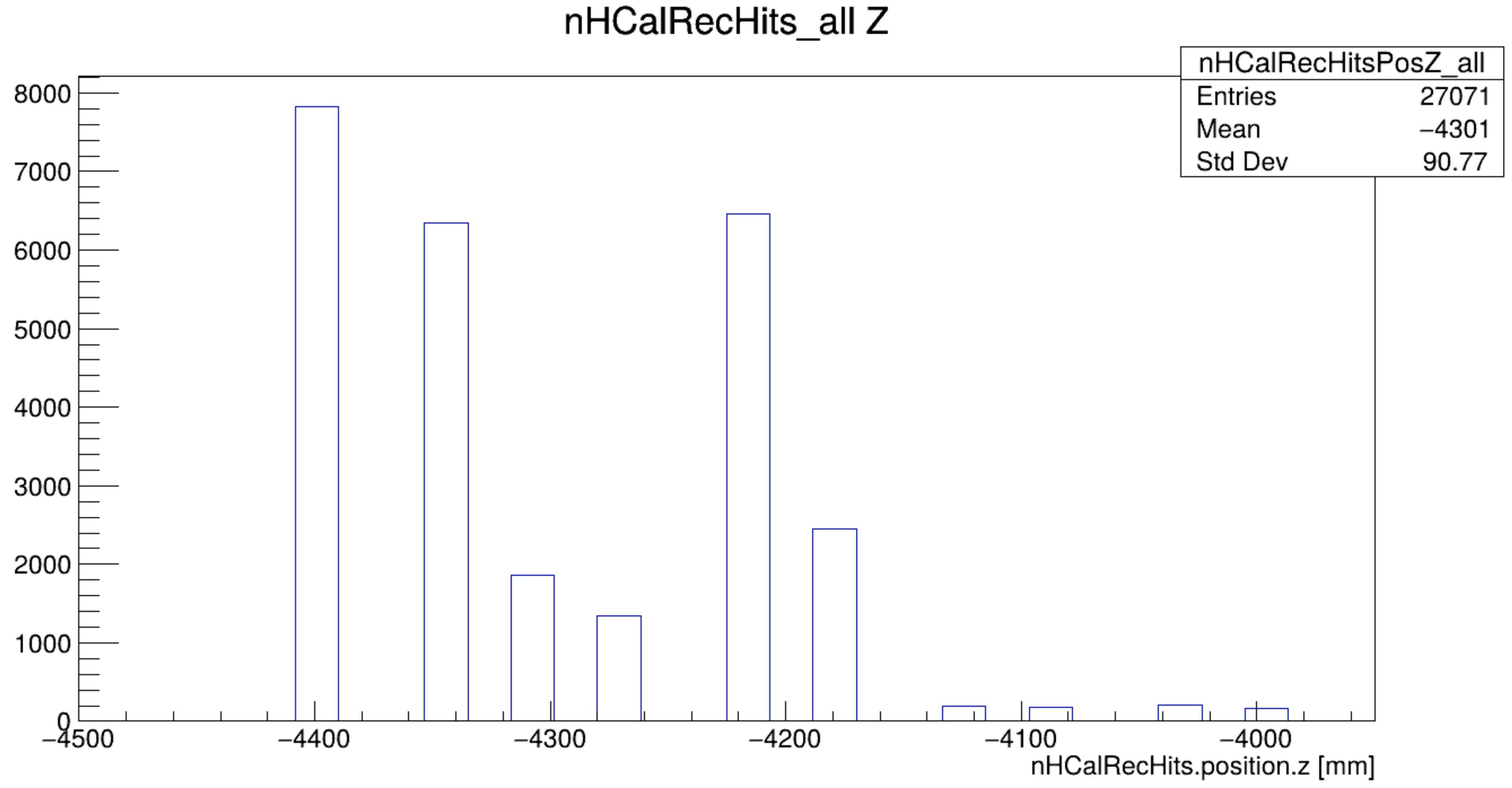




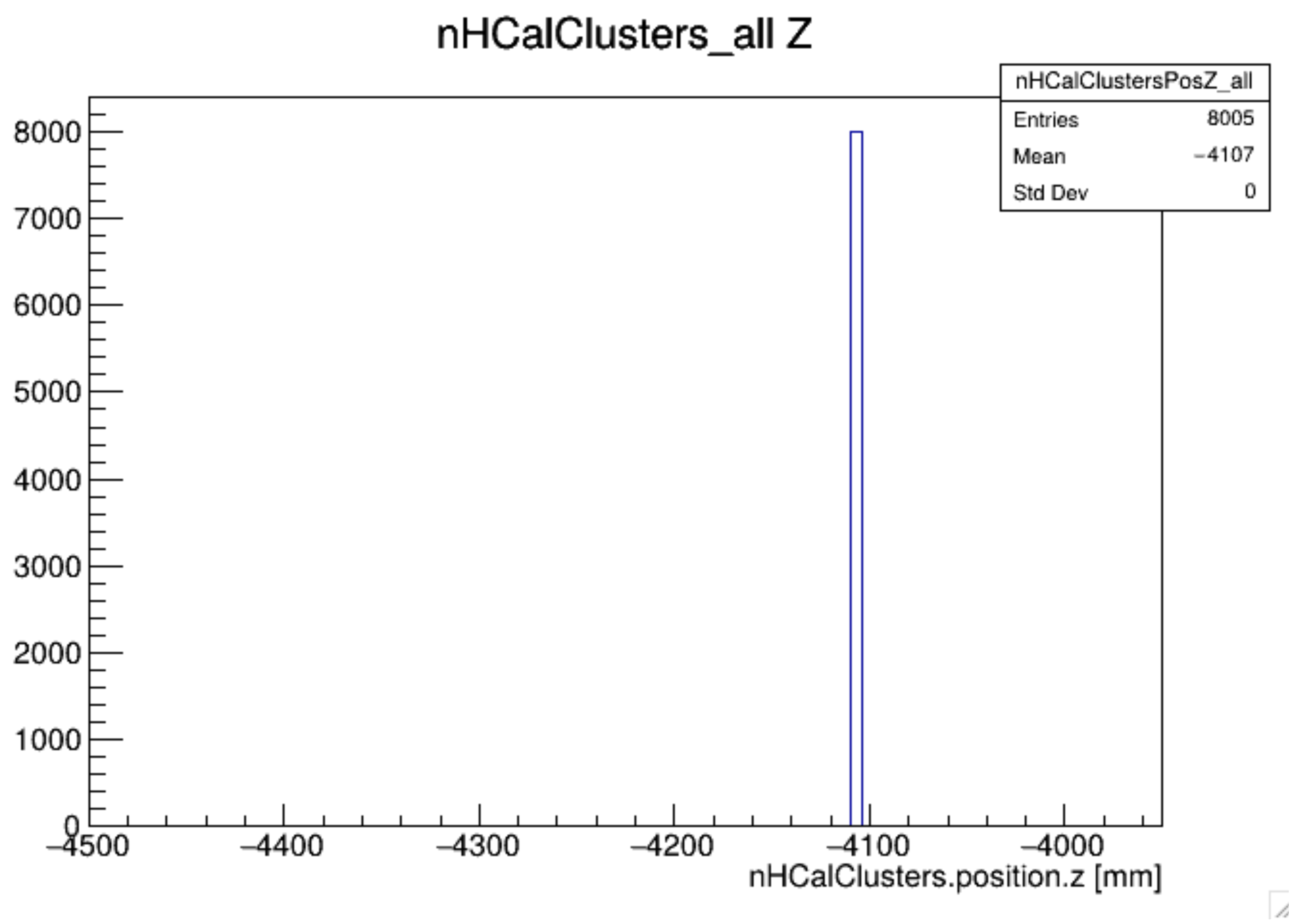
Clustering in z?

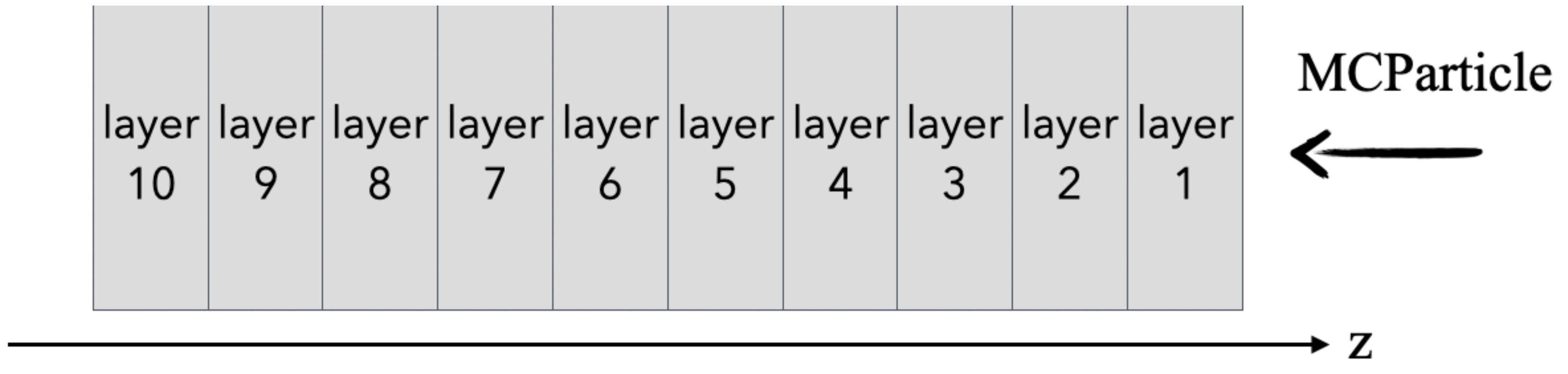
- env TILE_SIZE=5 N_LAYERS=10 SCINTILLATOR_THICKNESS=0.8 ABSORBER_THICKNESS=4 **MOMENTUM=1** PHI=45 THETA=170 PARTICLE=mu- NUMBER_OF_EVENTS=1000 ./run_submit.sh

hits



clusters

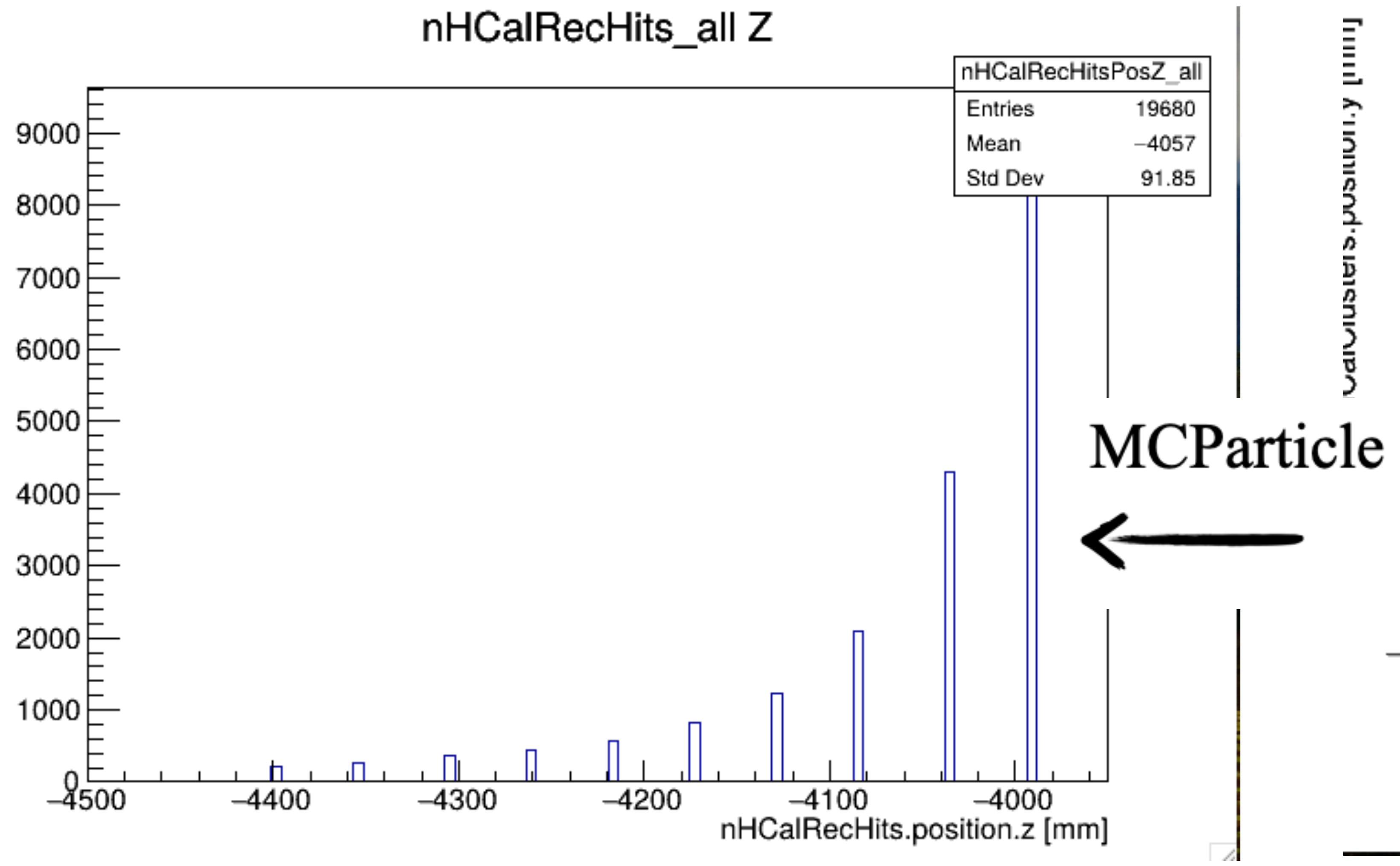


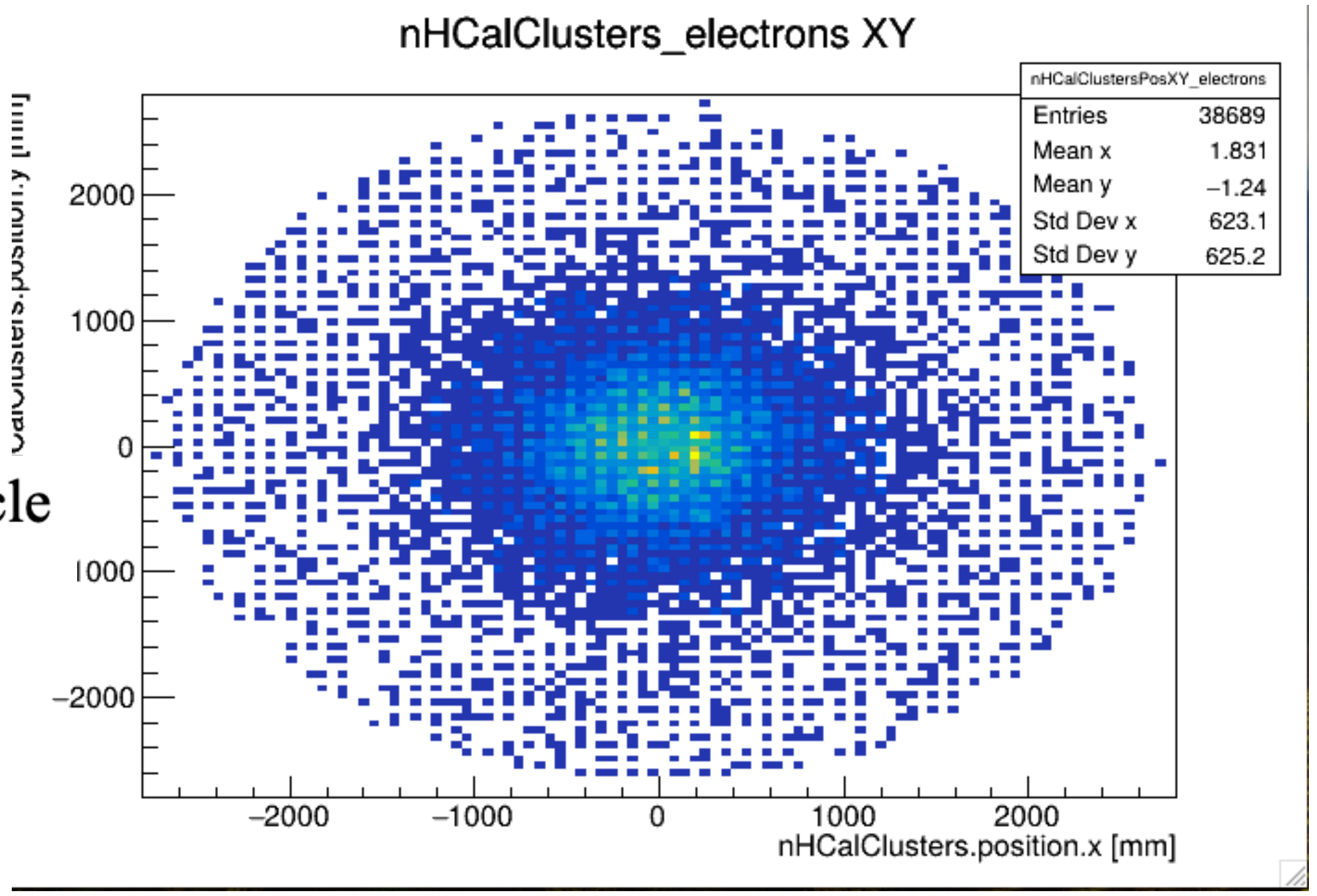


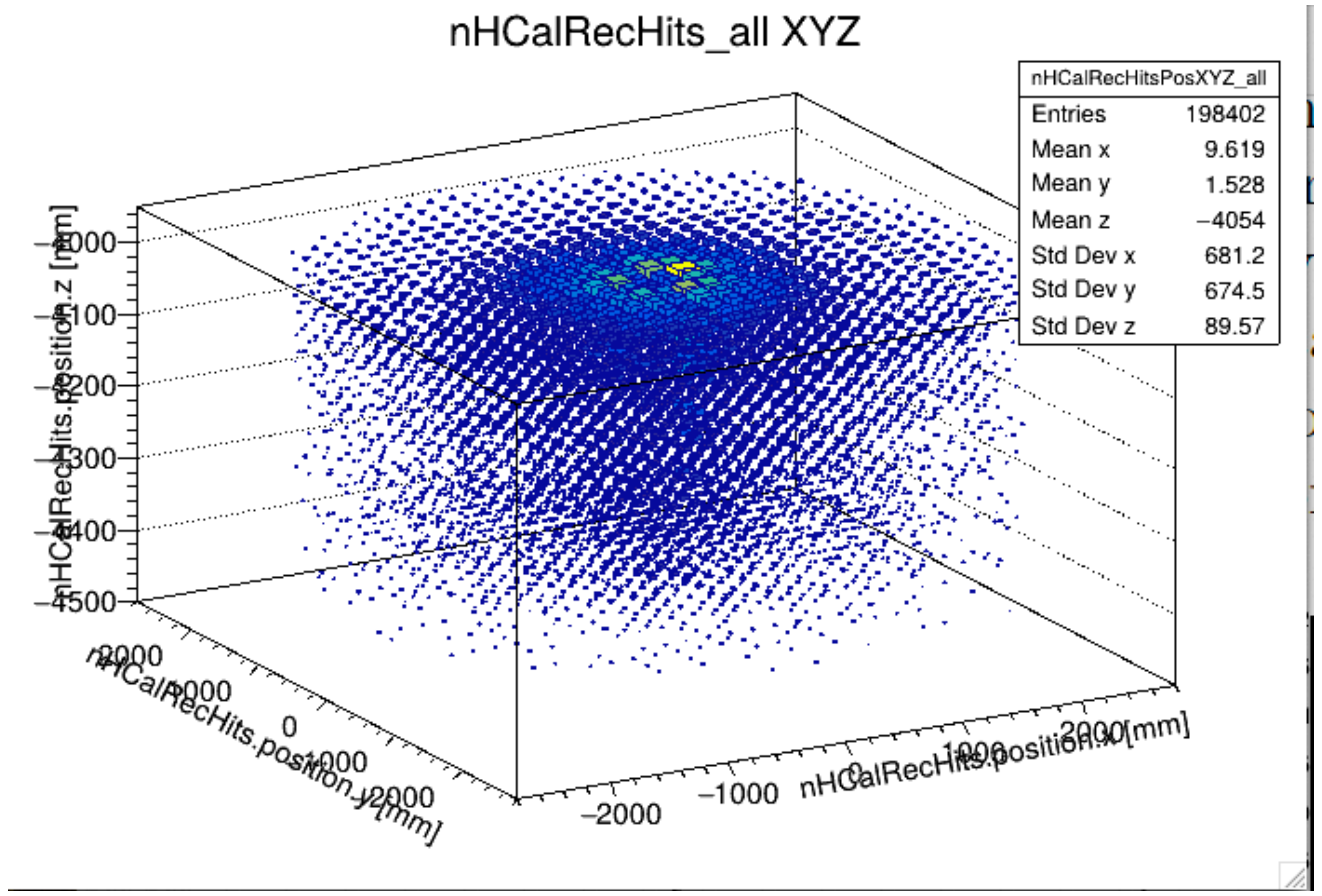
why does it all cluster to z~-4100mm?

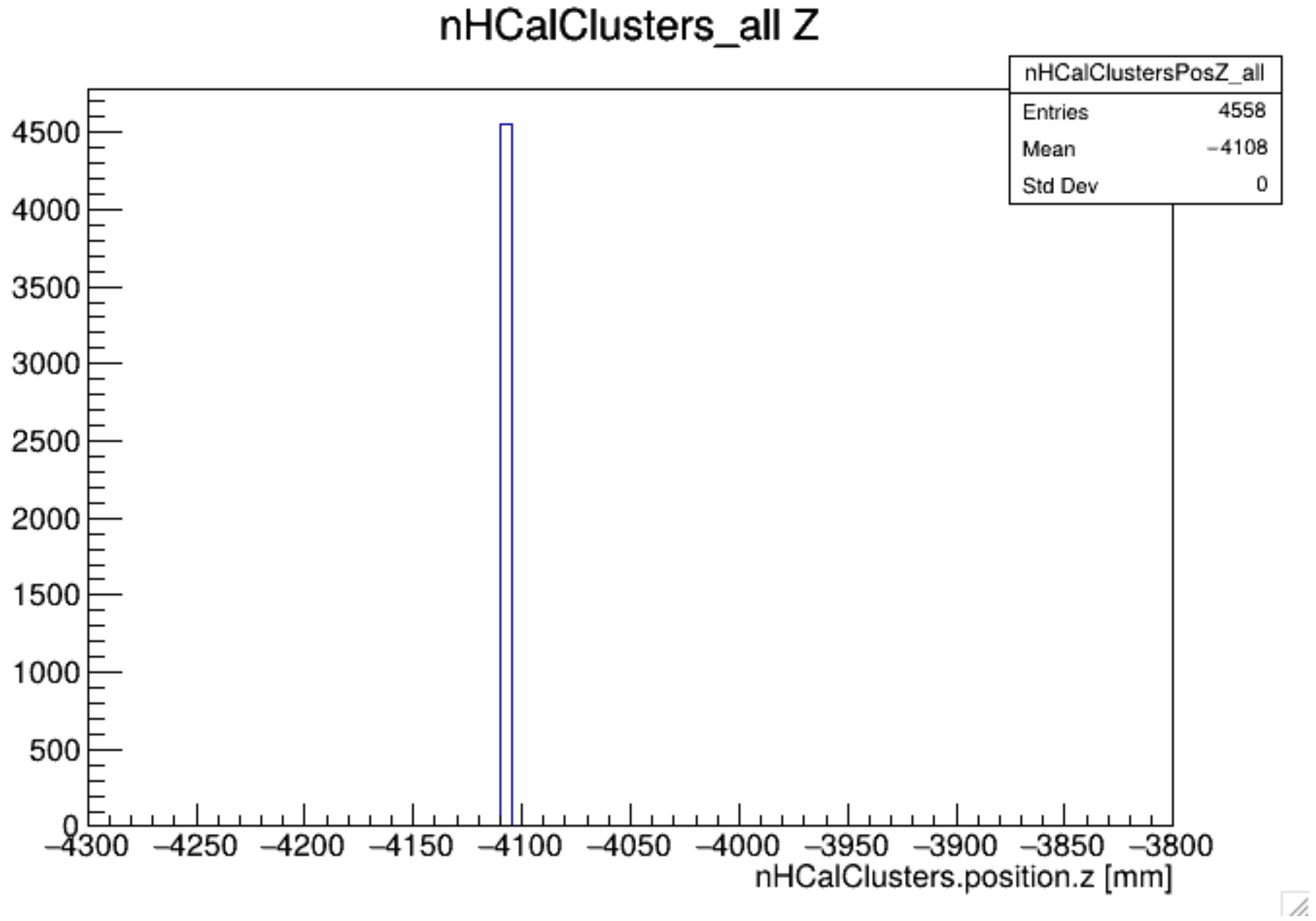
quick look at DIS sample

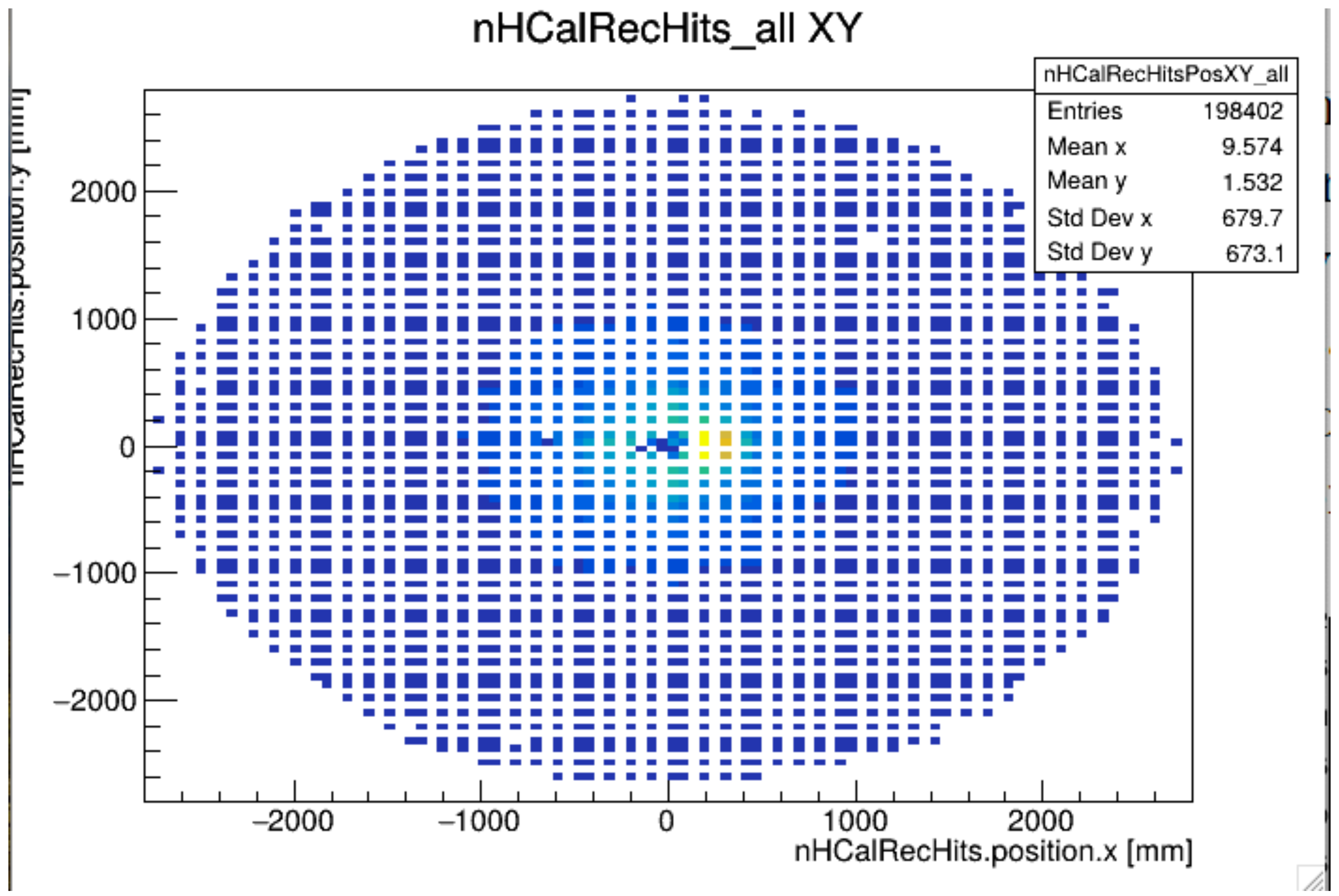
- DIS_NC_18x275_minQ2=1_24.08.1_100runs

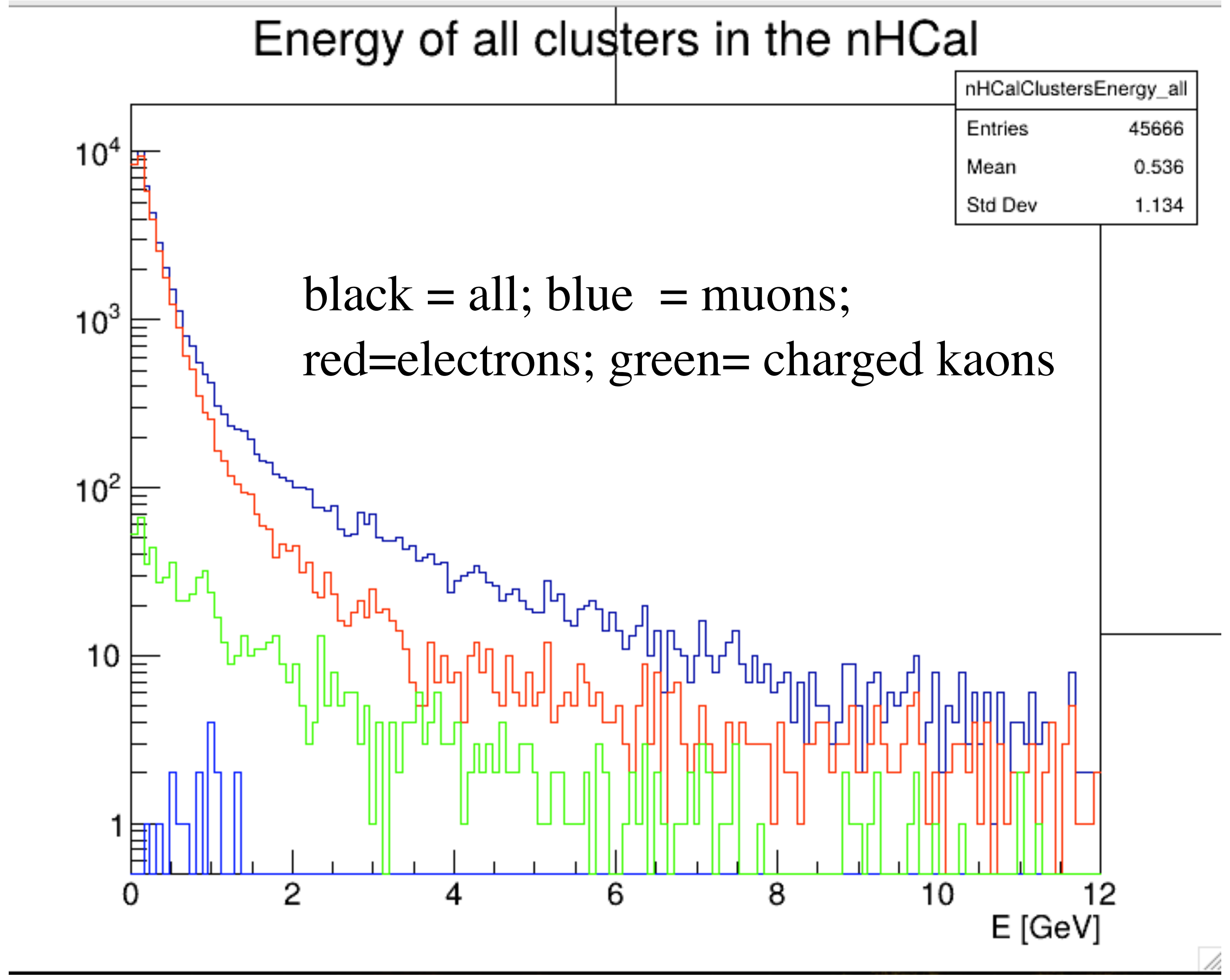






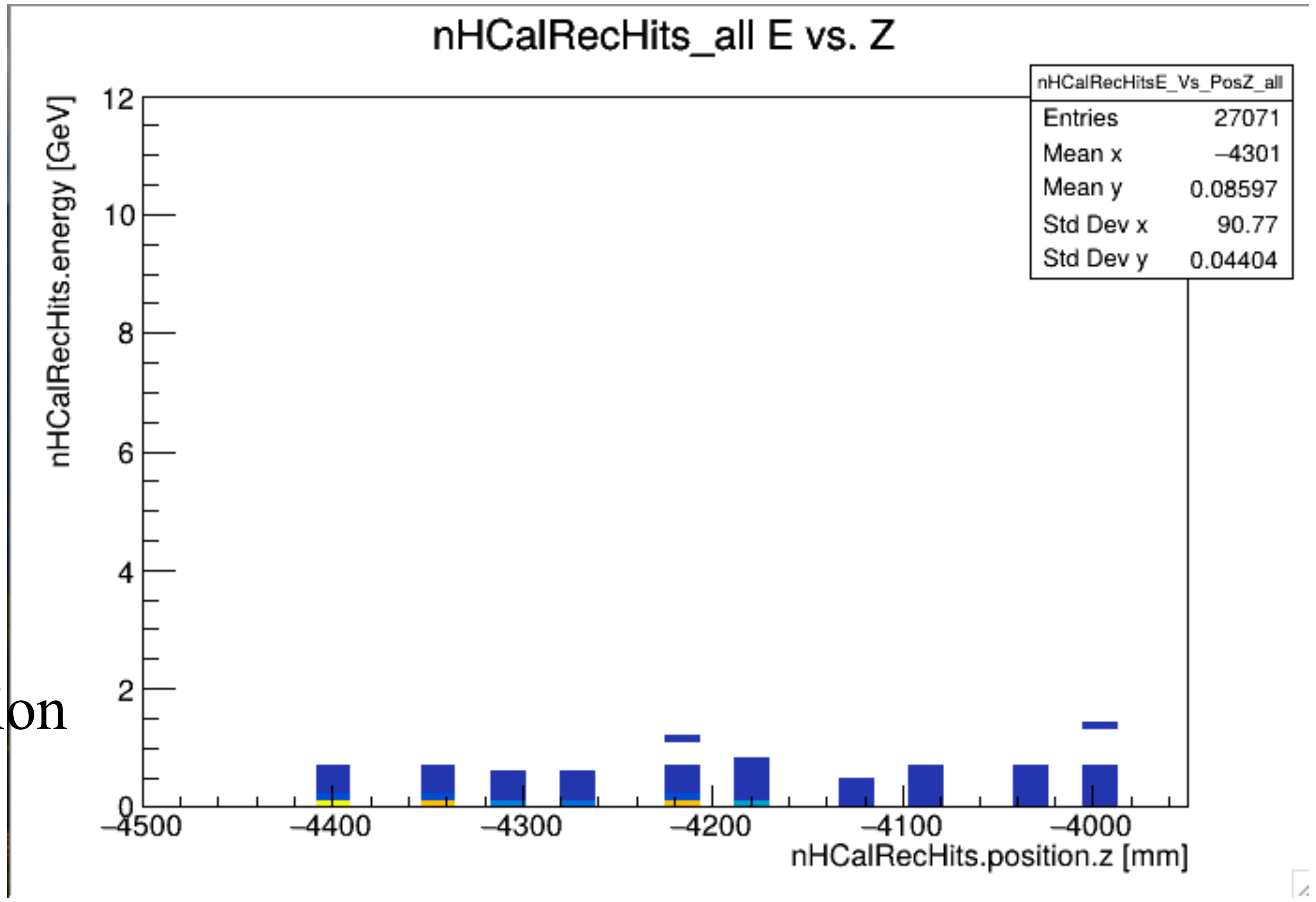






Steps: muon efficiency in the nHCal

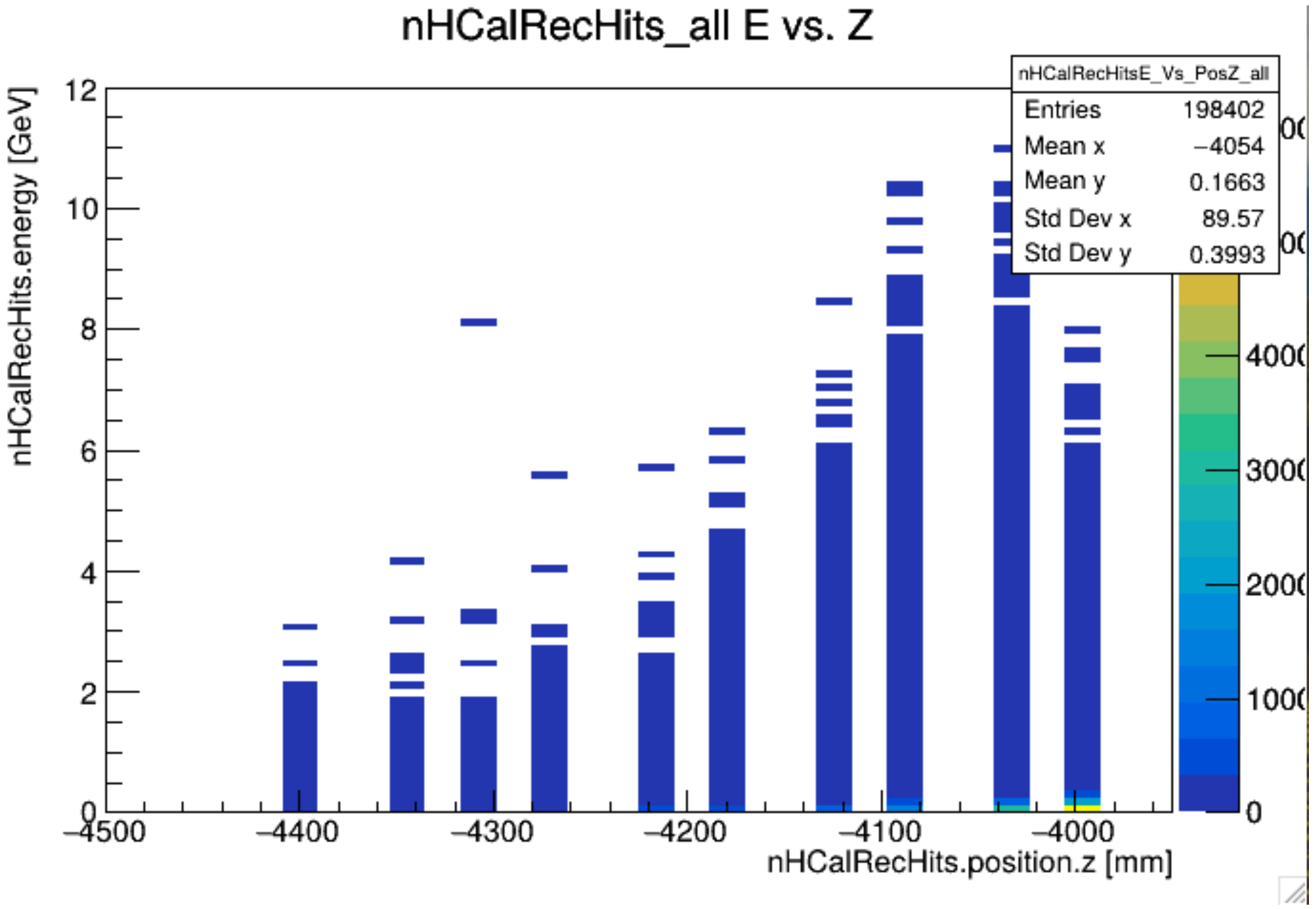
- Generate particle gun data.
  - Over a wide range of ϕ and η, and various momenta 1…~10 GeV… The particle gun generates for ϕ=fixed, η=fixed, or ϕ=fixed and η=flat, or vice versa.
- Define MIP signal in a given layer.
  - Start with muon at theta=0 and detector close to begin of beam (to avoid that muon hits anything else)
  - Define MIP threshold - how?
  - If say 6 out of 10 layers say "yes, it's muon / MIP", tag on particle level.
- Fill "all" and "MIP" histograms (ϕ, η) - Efficiency = MIP/all. Also plot versus momentum.
- Figure out correct uncertainties.
- Start with default nHCal design. Once established, vary design parameters

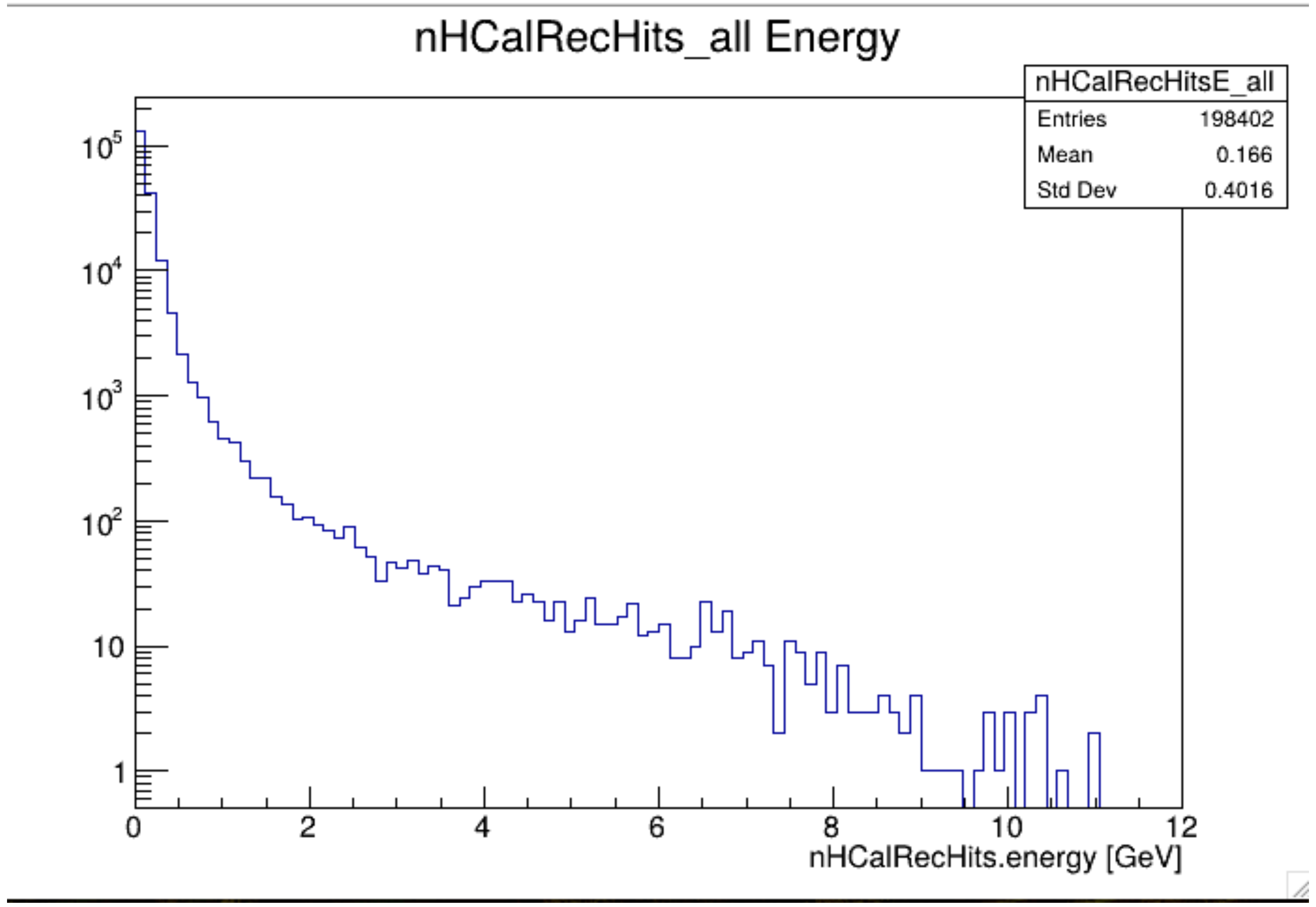


Can't take cluster information - all collapsed to 1 z-location

Hit energies and distributions for DIS

- DIS_NC_18x275_minQ2=1_24.08.1_100runs





Planned design variations for the nHCal

- see table.

5x5 or 10x10	scintillator thickness	0.4				0.6	0.8
[cm]	absorber thickness	4	3	2	1.52	2	4
10 layers		45					
12 layers		54					
13 layers			46				
15 layers		68					
20 layers				50		54	58
28 layers					57		

gross length

nHCal default design (10x10)

LFHCal configuration (5x5)