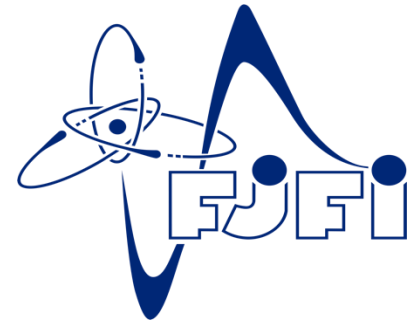
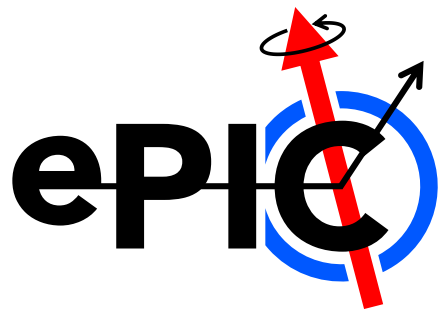


Two-Particle Position Resolution Study from Backward HCal

Leszek Kosarzewski, Alexandr Prozorov, **Subhadip Pal**



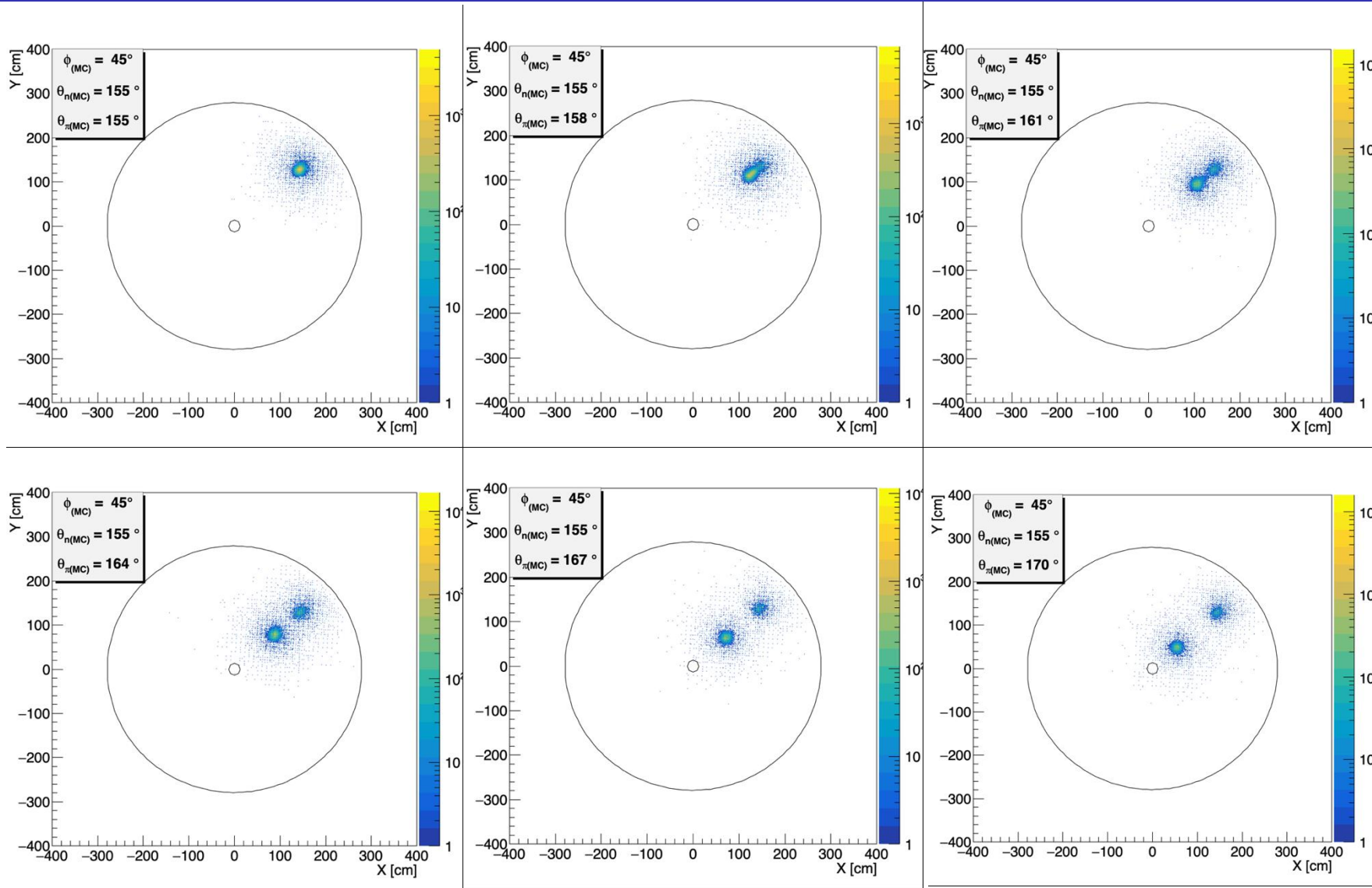
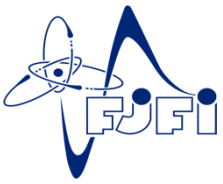
ePIC nHCal-DSC meeting – August 30, 2024

Objective : Use clusters to distinguish between neutron/pion shower reconstruction.

- $(1 n + 1 \pi) / \text{event.}$ ---- Standalone ddsim
- $\varphi = 45^\circ$
 - $\theta_n = 155^\circ$ ($\eta = -1.51$) ----- fixed
 - $\theta_\pi = 155^\circ$ ($\eta = -1.51$), 158° ($\eta = -1.64$),
 161° ($\eta = -1.79$), 164° ($\eta = -1.96$),
 167° ($\eta = -2.17$), 170° ($\eta = -2.44$)

- Only Backward HCal was taken into account [not the whole ePIC geometry – scattering effects neglected]
- $-4.14 < \eta < -1.18$
- Alternating Steel and Scintillator slices
- 10 cm. x 10 cm. Polystyrene tiles

Cluster Positions (xy coordinates)



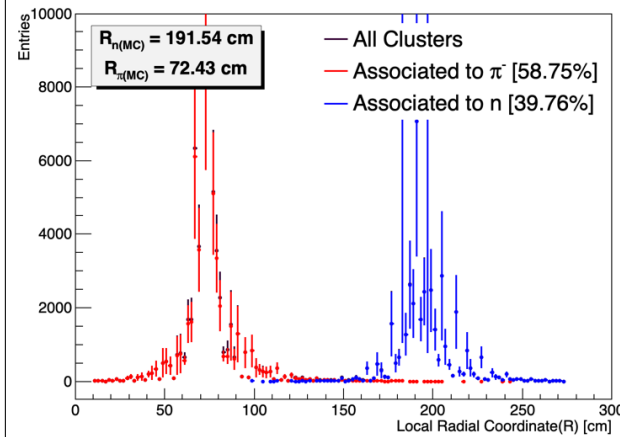
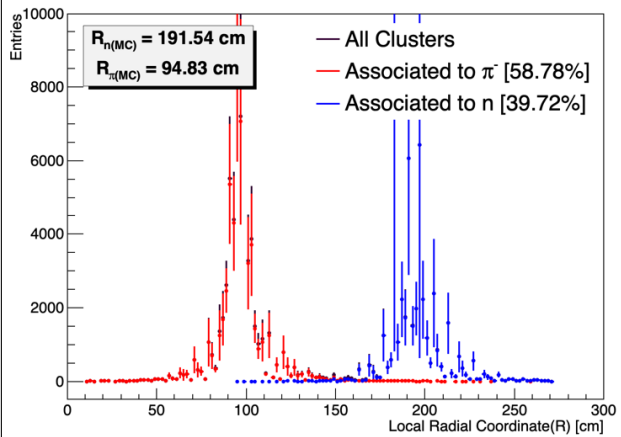
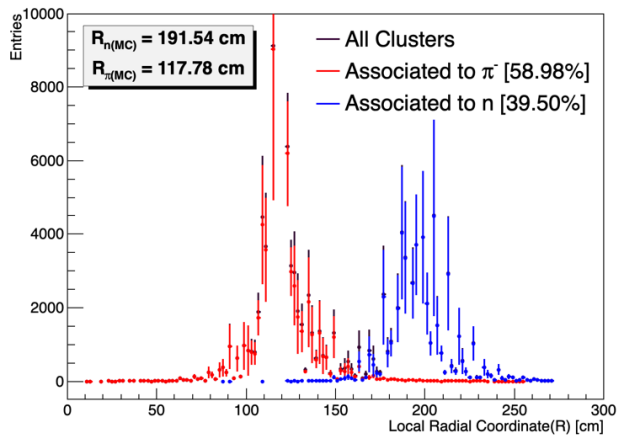
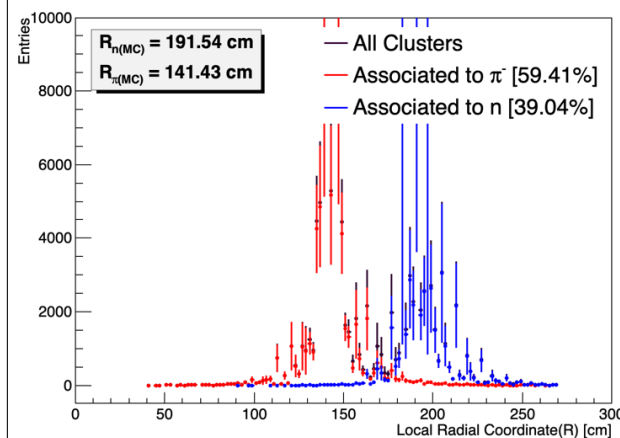
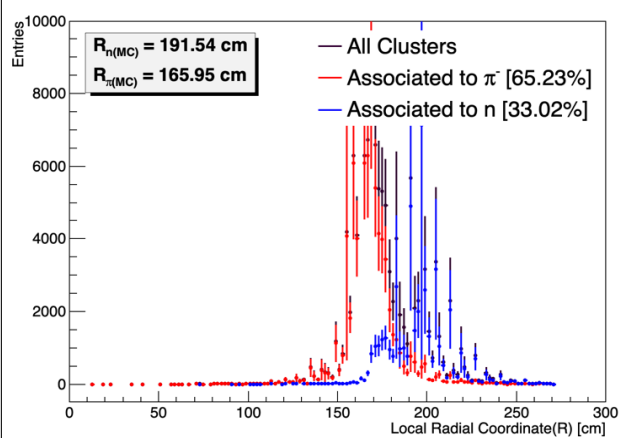
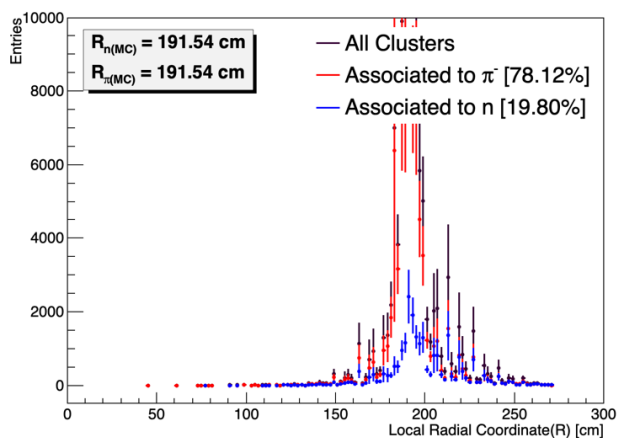
Cluster (x,y) are shown along with simulated angular coordinates

$p = 1 \text{ GeV}/c$

[neutron showers in outer region; pion showers in inner region]

Distributions are becoming more smeared more distinguishable as $(\theta_\pi - \theta_n)$ increases...

Cluster Radial Coordinates



$p = 1 \text{ GeV}/c$

~~$\approx 80\%$ of the clusters associated with pions~~

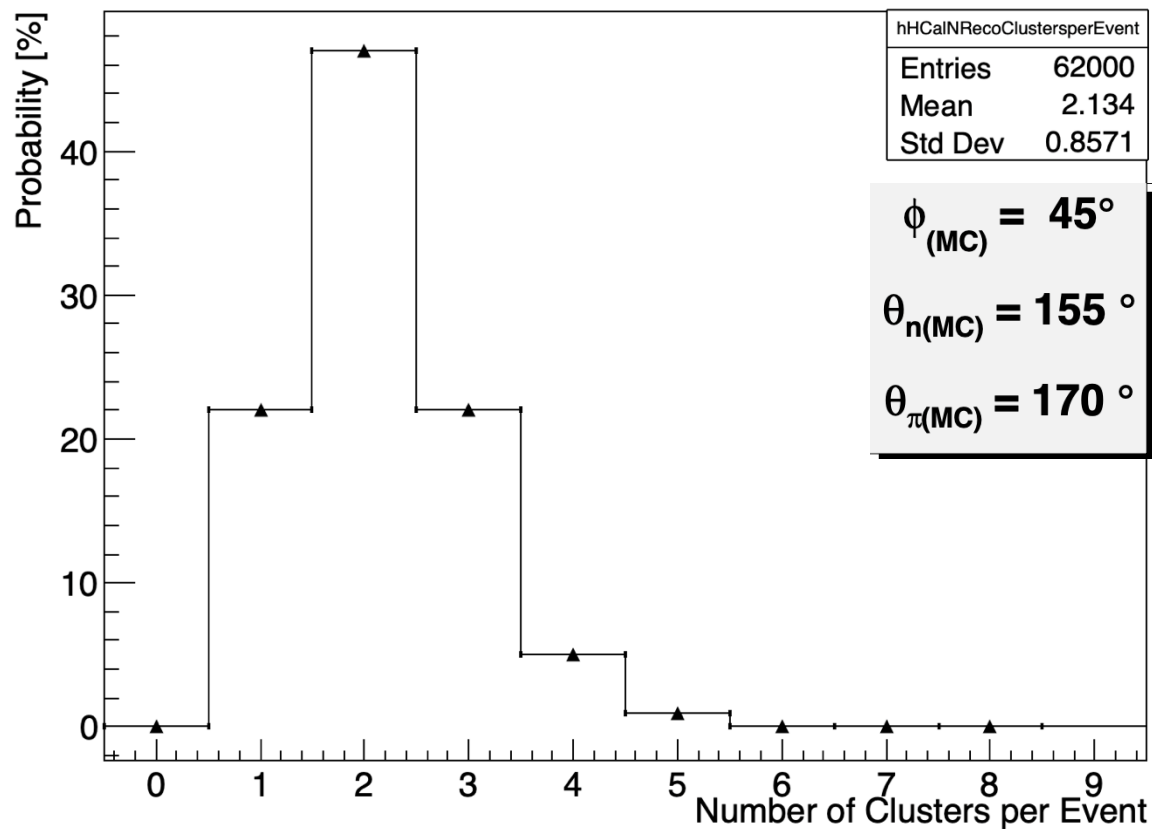
~~$\approx 20\%$ of the clusters associated with neutrons~~

~~Neutron Clusters start to shift inwards as $(R_n - R_\pi)$ increases~~

Percentages (fraction of clusters identified as π -/n clusters) are based on ClusterMCParticle associations [better performance as the π -n distance increases]

Clusters are in the expected positions

Number of Reco Clusters per Event

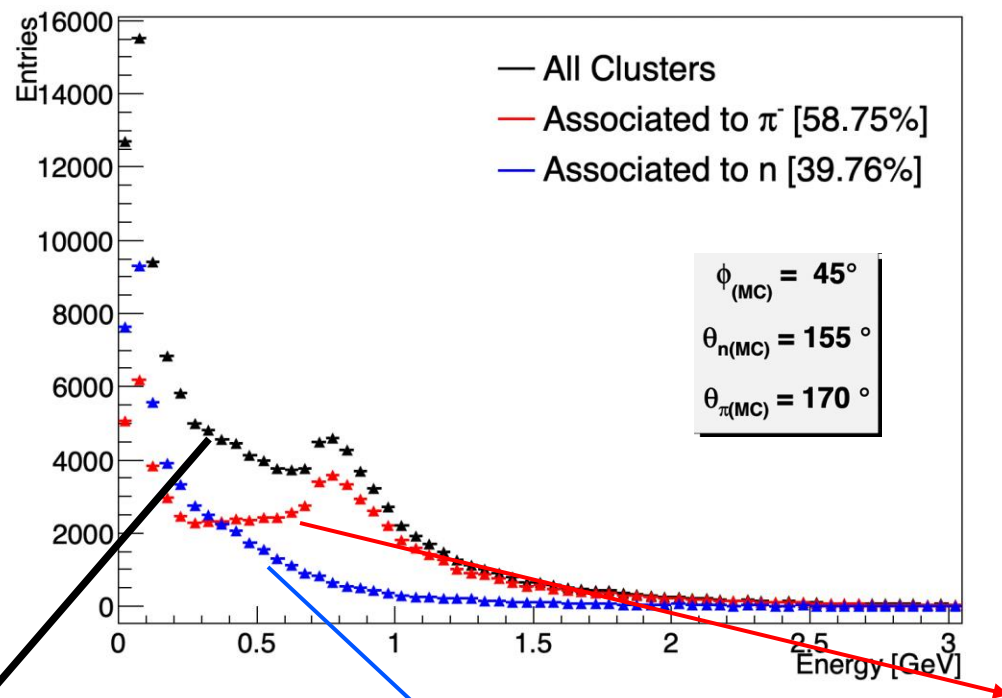


Expectation: 2 clusters per event

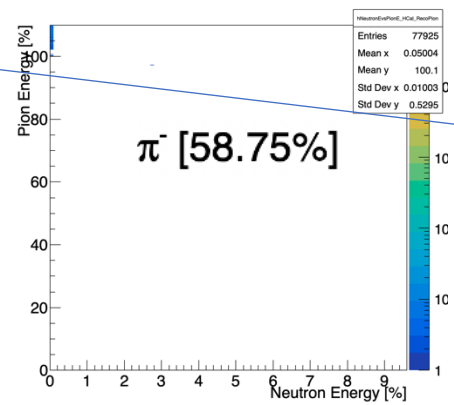
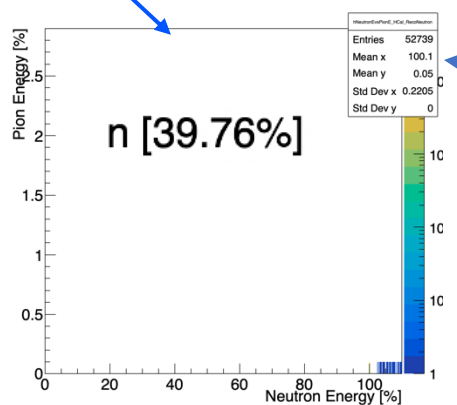
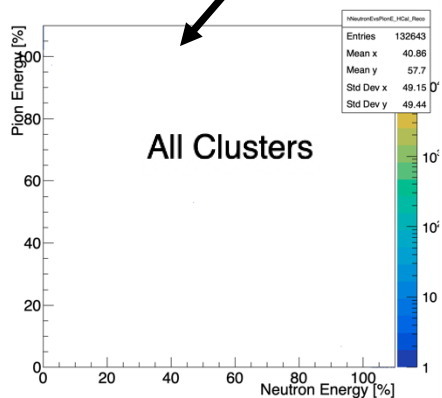
Observation: ~~> 90%~~
~~events have 1 reconstructed cluster~~
Mean = 2.134

~~Clusters are being merged...?~~
Better Performance.

Cluster Reconstruction



- Cluster energies have been traced back to the constituent `recoHits` mergedRecoHits [can be accessed via `cluster.getHits()`] which were tagged as pion/neutron hits based on the most energetic hit contribution of the mapped simHit [can be accessed by comparing cellIDs] of the mergedRecoHits.



Cluster-MCparticle Association works well.

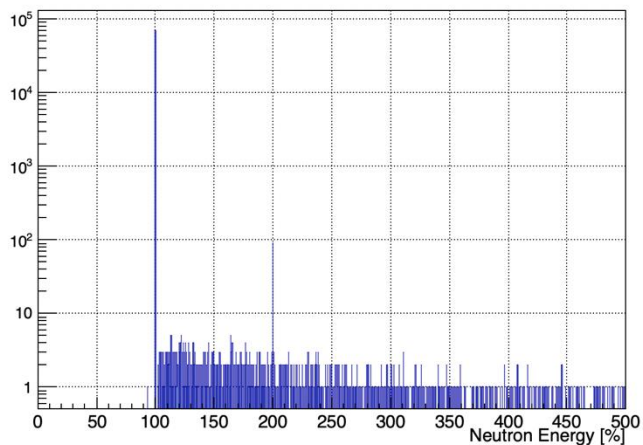
$$Pion(Neutron)Energy\% = \frac{\sum Pion(Neutron)RecohitEnergy}{ClusterEnergy} \times 100$$

Cluster Reconstruction

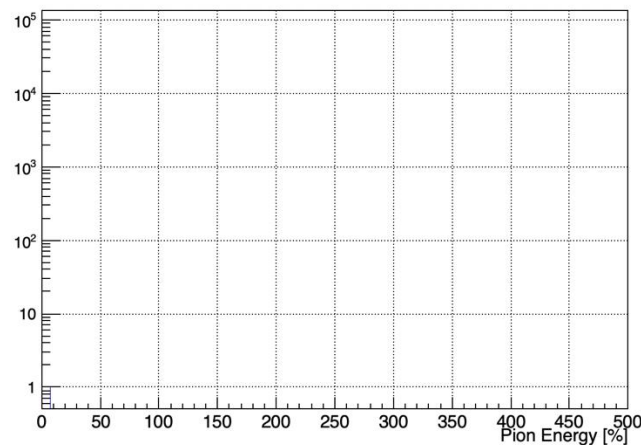


Neutron Clusters

ProjectionX of biny=[0,5000] [y=-0.10..500.00]

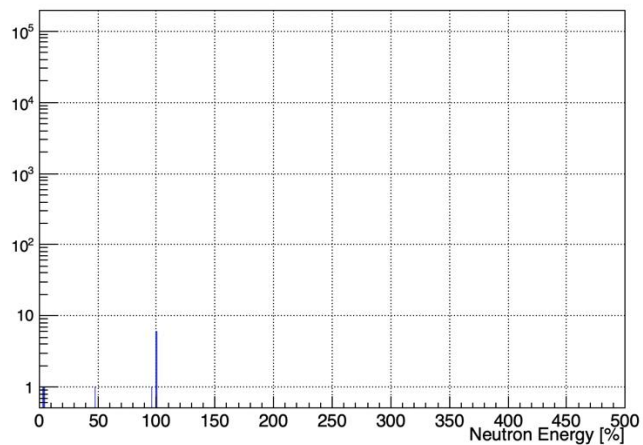


ProjectionY of binx=[0,5000] [x=-0.10..500.00]

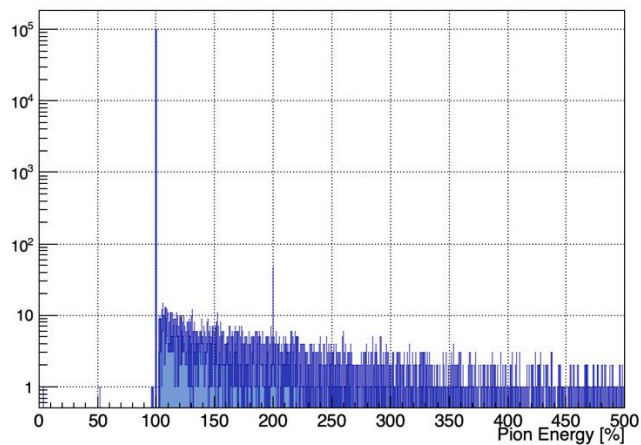


Pion Clusters

ProjectionX of biny=[0,5000] [y=-0.10..500.00]

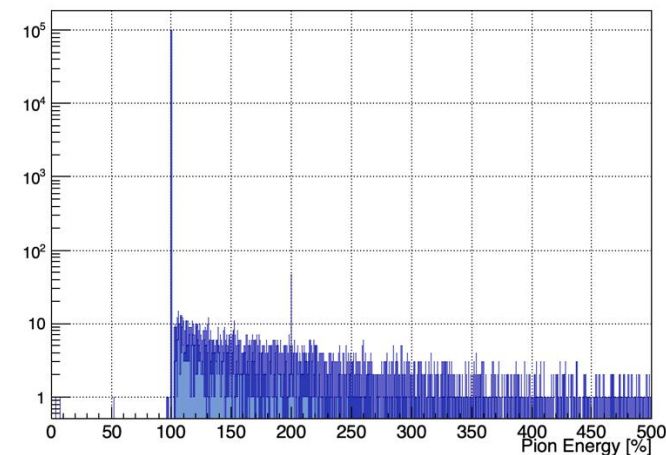


ProjectionY of binx=[0,5000] [x=-0.10..500.00]

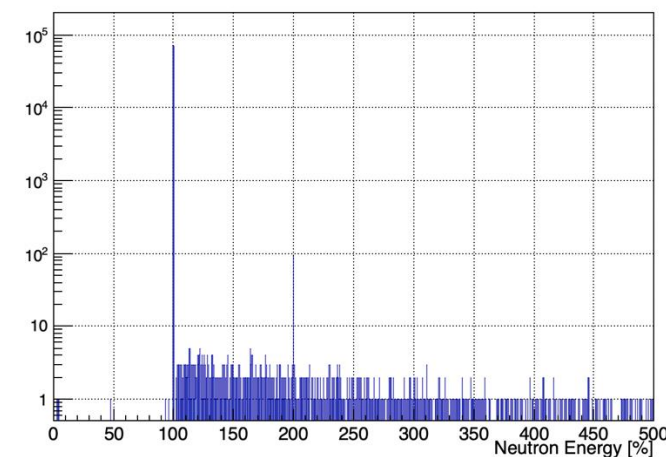


All Clusters

ProjectionY of binx=[0,5000] [x=-0.10..500.00]



ProjectionX of biny=[0,5000] [y=-0.10..500.00]



*discussion on next slides with an example

Cluster Reconstruction



Processing event 84959/85000... # **Event containing 2 clusters**

```
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (0, 0, 16, 59, 3)
Cluster: energy = 0.721966, nHits = 6
Rec Hit: energy = 0.0263415, time = 17.3, pos.z() = -4107.5
Sim Hit: energy = 0.000220031
Contrib: energy = 0.000220031, time = 17.303
Particle: energy = 1.00969, pdg = -211, gen status = 1
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (0, 1, 24, 89, 3)
Cluster: energy = 0.721966, nHits = 6
Rec Hit: energy = 0.132992, time = 14.29, pos.z() = -4107.5
Sim Hit: energy = 0.00127383
Contrib: energy = 0.00121976, time = 14.2872
Particle: energy = 1.00969, pdg = -211, gen status = 1
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (0, 2, 23, 84, 3)
Cluster: energy = 0.721966, nHits = 6
Rec Hit: energy = 0.462582, time = 13.83, pos.z() = -4107.5
Sim Hit: energy = 0.00295112
Contrib: energy = 0.000721806, time = 14.039
Particle: energy = 1.00969, pdg = -211, gen status = 1
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (0, 3, 30, 102, 3)
Cluster: energy = 0.721966, nHits = 6
Rec Hit: energy = 0.0424034, time = 14.62, pos.z() = -4107.5
Sim Hit: energy = 0.000395054
Contrib: energy = 3.9479e-06, time = 68.6456
Particle: energy = 1.00969, pdg = -211, gen status = 1
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (0, 4, 35, 132, 3)
Cluster: energy = 0.721966, nHits = 6
Rec Hit: energy = 0.0841643, time = 14.22, pos.z() = -4107.5
Sim Hit: energy = 0.000807706
Contrib: energy = 0.000807706, time = 14.2187
Particle: energy = 1.00969, pdg = -211, gen status = 1
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (0, 5, 37, 134, 3)
Cluster: energy = 0.721966, nHits = 6
Rec Hit: energy = 0.090589, time = 14.8, pos.z() = -4107.5
Sim Hit: energy = 0.000865865
Contrib: energy = 0.00016734, time = 14.7966
Particle: energy = 1.00969, pdg = -211, gen status = 1
```

```
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (1, 0, 16, 59, 3)
Cluster: energy = 0.117107, nHits = 6
Rec Hit: energy = 0.0263415, time = 17.3, pos.z() = -4107.5
Sim Hit: energy = 0.000220031
Contrib: energy = 0.000220031, time = 17.303
Particle: energy = 1.00969, pdg = -211, gen status = 1
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (1, 1, 24, 89, 3)
Cluster: energy = 0.117107, nHits = 6
Rec Hit: energy = 0.132992, time = 14.29, pos.z() = -4107.5
Sim Hit: energy = 0.00127383
Contrib: energy = 0.00121976, time = 14.2872
Particle: energy = 1.00969, pdg = -211, gen status = 1
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (1, 2, 23, 84, 3)
Cluster: energy = 0.117107, nHits = 6
Rec Hit: energy = 0.462582, time = 13.83, pos.z() = -4107.5
Sim Hit: energy = 0.00295112
Contrib: energy = 0.000721806, time = 14.039
Particle: energy = 1.00969, pdg = -211, gen status = 1
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (1, 3, 30, 102, 3)
Cluster: energy = 0.117107, nHits = 6
Rec Hit: energy = 0.0424034, time = 14.62, pos.z() = -4107.5
Sim Hit: energy = 0.000395054
Contrib: energy = 3.9479e-06, time = 68.6456
Particle: energy = 1.00969, pdg = -211, gen status = 1
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (1, 4, 35, 132, 3)
Cluster: energy = 0.117107, nHits = 6
Rec Hit: energy = 0.0841643, time = 14.22, pos.z() = -4107.5
Sim Hit: energy = 0.000807706
Contrib: energy = 0.000807706, time = 14.2187
Particle: energy = 1.00969, pdg = -211, gen status = 1
[Contrib #0] (cluster, reco hit, sim hit, contrib, partide) ID = (1, 5, 37, 134, 3)
Cluster: energy = 0.117107, nHits = 6
Rec Hit: energy = 0.090589, time = 14.8, pos.z() = -4107.5
Sim Hit: energy = 0.000865865
Contrib: energy = 0.00016734, time = 14.7966
Particle: energy = 1.00969, pdg = -211, gen status = 1
```

Constituent RecHit energies can be > Cluster energies
*continued discussion on next slide

Cluster 1:
cluster.getEnergy(): 0.721966
hcalreco_neutronE: 0
hcalreco_pionE: 0.839073

Pion energy contribution > cluster energy

Cluster 2:
cluster.getEnergy(): 0.117107
hcalreco_neutronE: 0
hcalreco_pionE: 0.839073

*Rec hits are mergedRechts
[look at the pos.z()]

Cluster energy is determined after doing a weighted sum of the mergedhits. If the weight is too small, cluster energy can be < a constituent mergedhit energy.

```
for (unsigned i = 0; i < pcl.getHits().size(); ++i) {
  const auto& hit = pcl.getHits()[i];
  const auto weight = pcl.getWeights()[i];
  debug("hit energy = {} hit weight: {}", hit.getEnergy(), weight);
  auto energy = hit.getEnergy() * weight;
  totalE += energy;
  time += (hit.getTime() - time) * energy / totalE;
  cl.addToHits(hit);
  cl.addToHitContributions(energy);
  const float eta = edm4hep::utils::eta(hit.getPosition());
  if (eta < minHitEta) {
    minHitEta = eta;
  }
  if (eta > maxHitEta) {
    maxHitEta = eta;
  }
}
cl.setEnergy(totalE / m_cfg.sampFrac);
cl.setEnergyError(0.);
cl.setTime(time);
cl.setTimeError(timeError);
```

If a mergedRecoHit is far away from a local maxima; it will have a less weight to the cluster corresponding to that local maxima.

```
for (std::size_t idx : group) {
  size_t j = 0;
  // calculate weights for local maxima
  for (std::size_t cidx : maxima) {
    double energy = hits[cidx].getEnergy();
    double dist = edm4hep::utils::magnitude(transverseEnergyProfileMetric(hits[cidx], hits[idx]));
    weights[j] = std::exp(-dist * transverseEnergyProfileScaleUnits / m_cfg.transverseEnergyProfileScale) * energy;
    j += 1;
  }

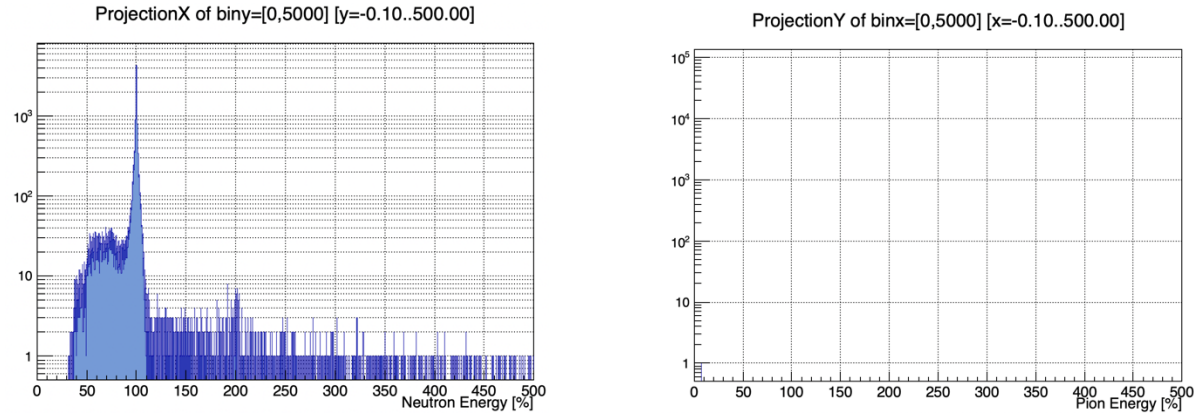
  // normalize weights
  vec_normalize(weights);

  // ignore small weights
  for (auto& w : weights) {
    if (w < 0.02) {
      w = 0;
    }
  }
  vec_normalize(weights);

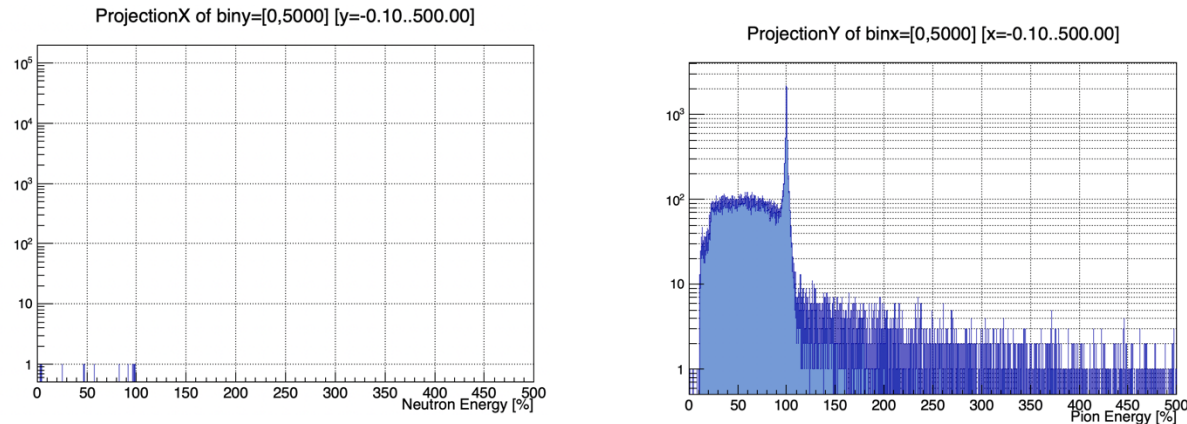
  // split energy between local maxima
  for (size_t k = 0; k < maxima.size(); ++k) {
    double weight = weights[k];
    if (weight <= 1e-6) {
      continue;
    }
    pcls[k].addToHits(hits[idx]);
    pcls[k].addToWeights(weight);
  }
}
```

*no of local maxima = no of clusters

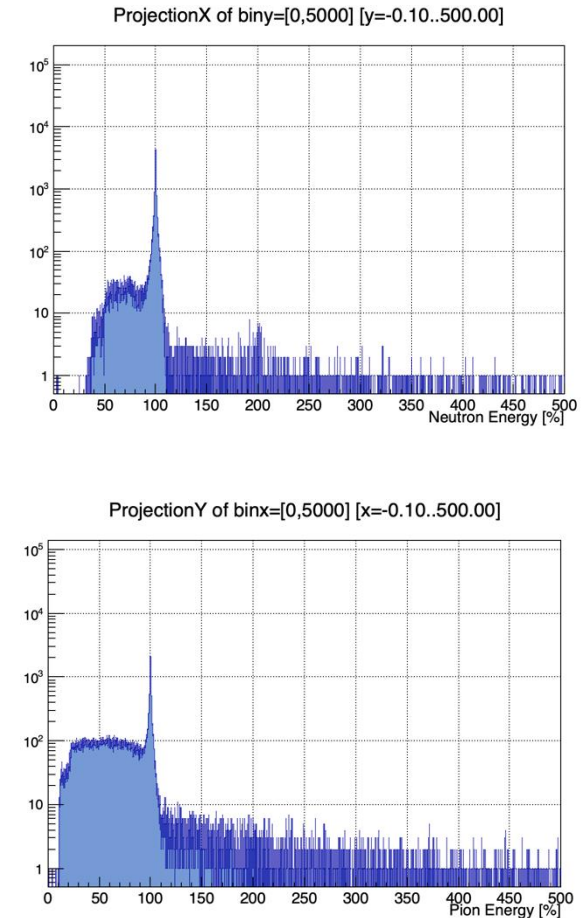
Neutron Clusters



Pion Clusters

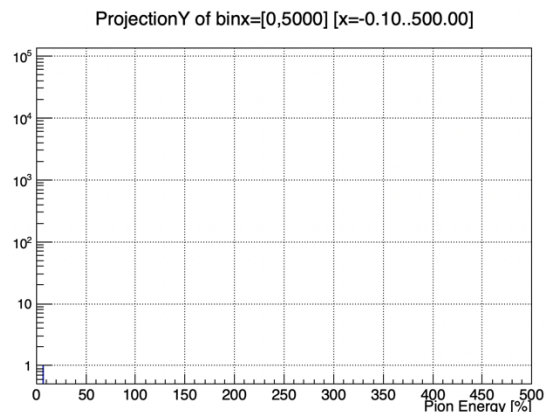
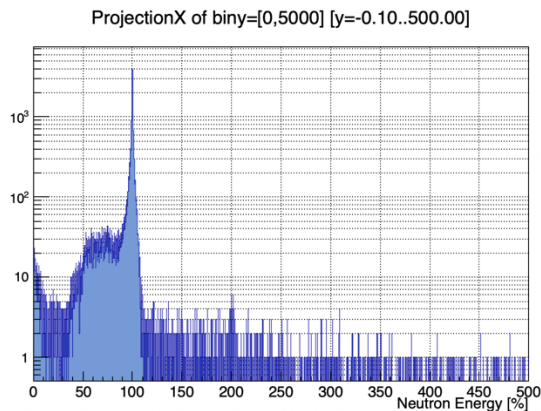


All Clusters

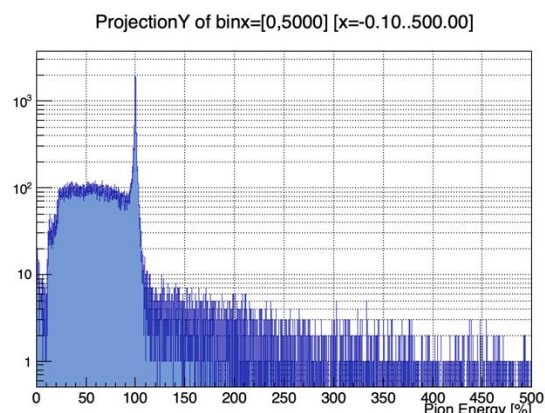
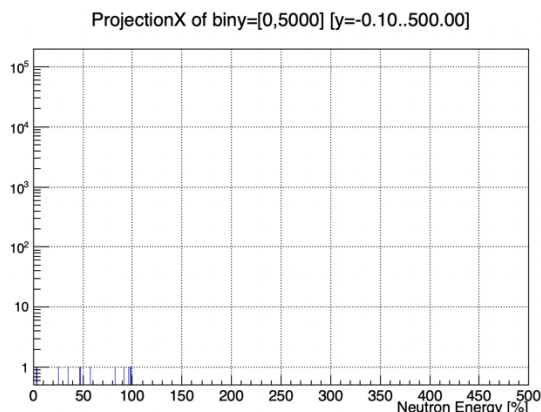


$$E_{Pion(Neutron)}\% = \frac{\sum E_{Pion(Neutron)}^{HitContribution} / sampling\ fraction}{Cluster\ Energy} \times 100$$

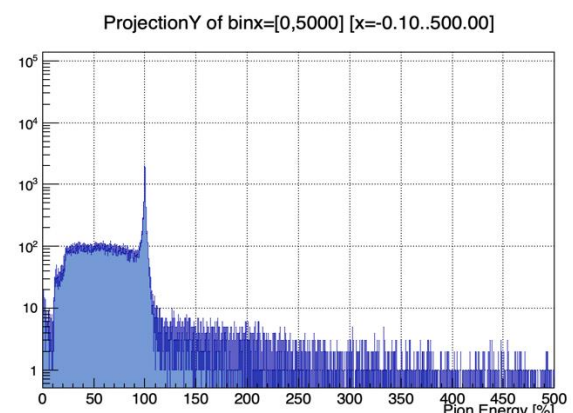
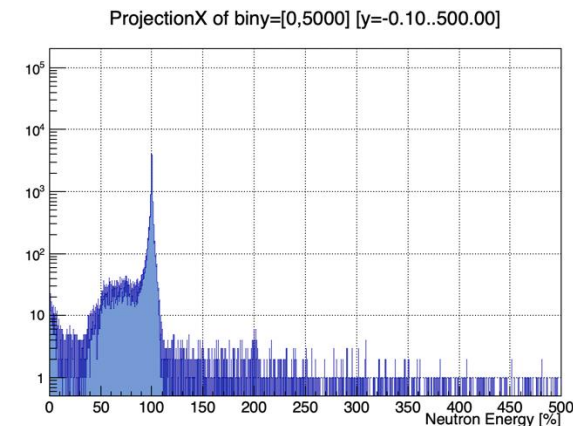
Neutron Clusters



Pion Clusters



All Clusters



$$E_{Pion(Neutron)}\% = \frac{\sum E_{Pion(Neutron)}^{HitContribution} / sampling\ fraction}{Cluster\ Energy} \times 100$$

if (sim.getEnergy() < 0.1875e-3) continue; [time info not stored in simhits]
 if (contrib.getTime() > 100) continue;

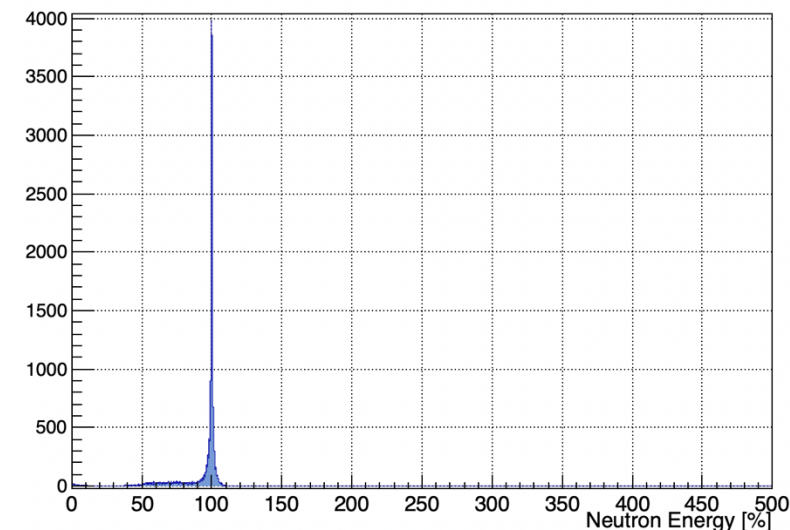
Cluster Reconstruction



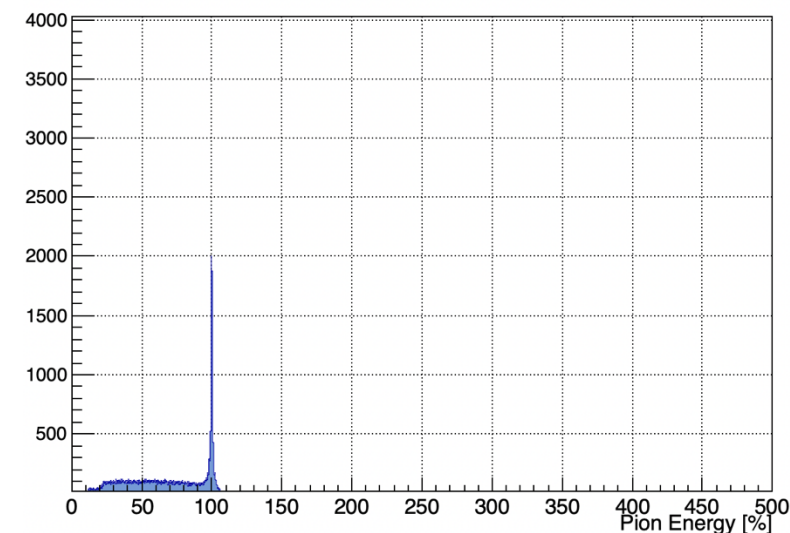
```
app->Add(new J0mniFactoryGeneratorT<CalorimeterHitDigi_factory>(  
    "HcalEndcapNRawHits",  
    {"HcalEndcapNHits"},  
#if EDM4EIC_VERSION_MAJOR >= 7  
    {"HcalEndcapNRawHits", "HcalEndcapNRawHitAssociations"},  
#else  
    {"HcalEndcapNRawHits"},  
#endif  
    {  
        .tRes = 0.0 * dd4hep::ns,  
        .capADC = HcalEndcapN_capADC,  
        .capTime = 100, // given in ns, 4 samples in HGCR0C  
        .dyRangeADC = HcalEndcapN_dyRangeADC,  
        .pedMeanADC = HcalEndcapN_pedMeanADC,  
        .pedSigmaADC = HcalEndcapN_pedSigmaADC,  
        .resolutionTDC = HcalEndcapN_resolutionTDC,  
        .corrMeanScale = "1.0",  
        .readout = "HcalEndcapNHits",  
    },  
    app // TODO: Remove me once fixed  
));  
app->Add(new J0mniFactoryGeneratorT<CalorimeterHitReco_factory>(  
    "HcalEndcapNRecHits", {"HcalEndcapNRawHits"}, {"HcalEndcapNRecHits"},  
    {  
        .capADC = HcalEndcapN_capADC,  
        .dyRangeADC = HcalEndcapN_dyRangeADC,  
        .pedMeanADC = HcalEndcapN_pedMeanADC,  
        .pedSigmaADC = HcalEndcapN_pedSigmaADC,  
        .resolutionTDC = HcalEndcapN_resolutionTDC,  
        .thresholdFactor = 0.0,  
        .thresholdValue = 41.0, // 0.1875 MeV deposition out of 200 MeV max (per layer) --> adc = 10 + 0.1875 / 200 * 32768 == 41  
        .sampFrac = "0.0095", // from latest study - implement at level of reco hits rather than clusters  
        .readout = "HcalEndcapNHits",  
    },  
    app // TODO: Remove me once fixed
```

```
if (sim.getEnergy() < 0.1875e-3) continue;  
if (contrib.getTime() > 100) continue;
```

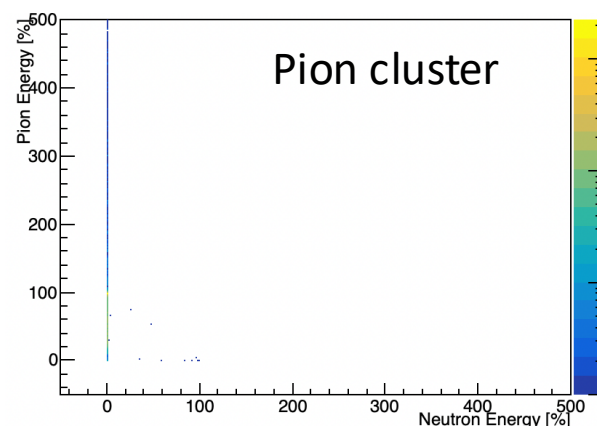
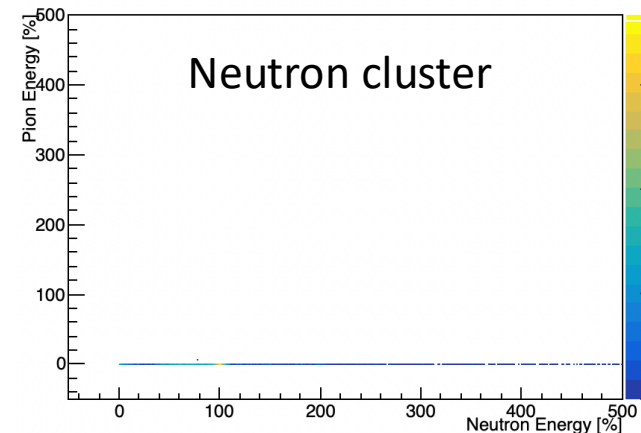
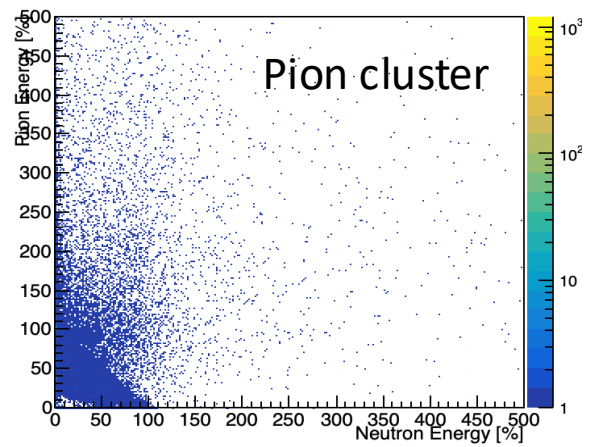
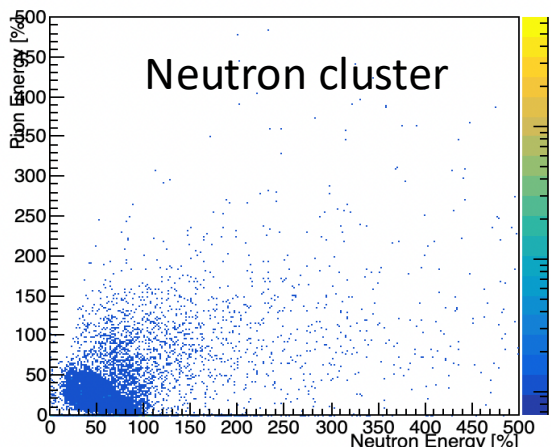
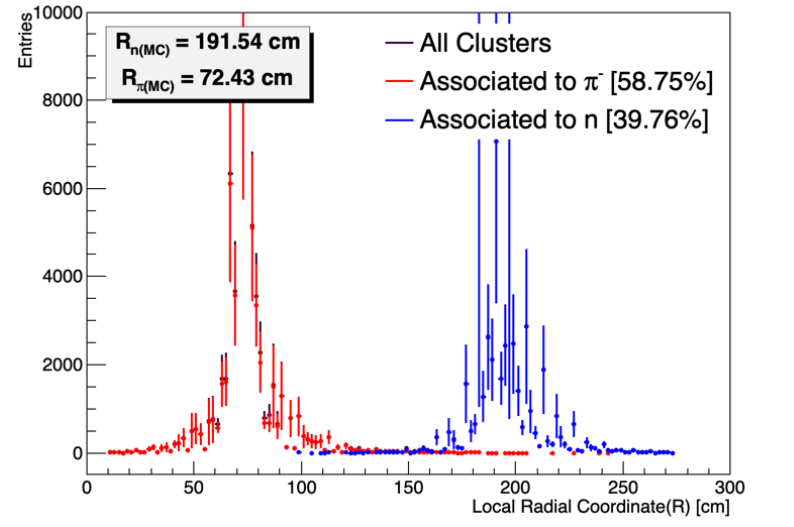
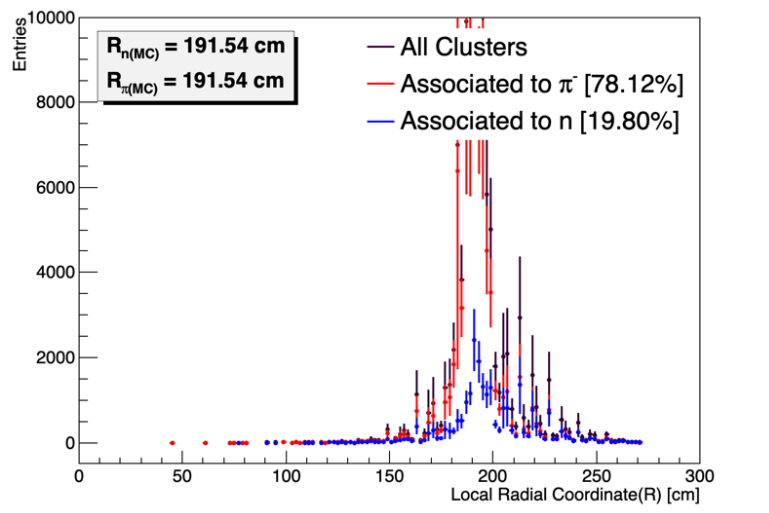
ProjectionX of biny=[0,5000] [y=-0.10..500.00]



ProjectionY of binx=[0,5000] [x=-0.10..500.00]



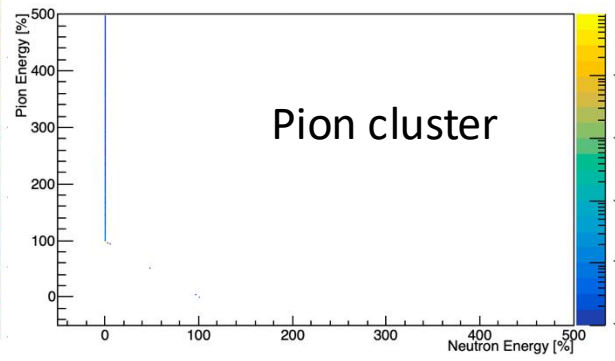
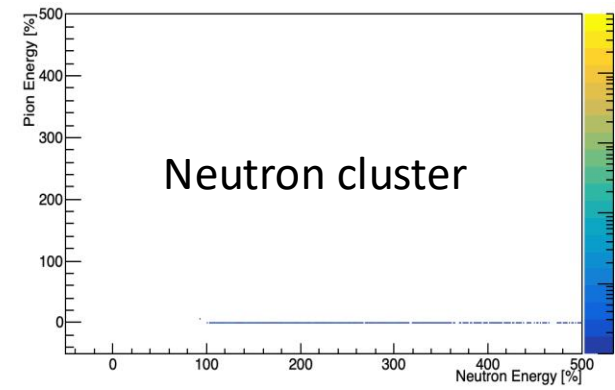
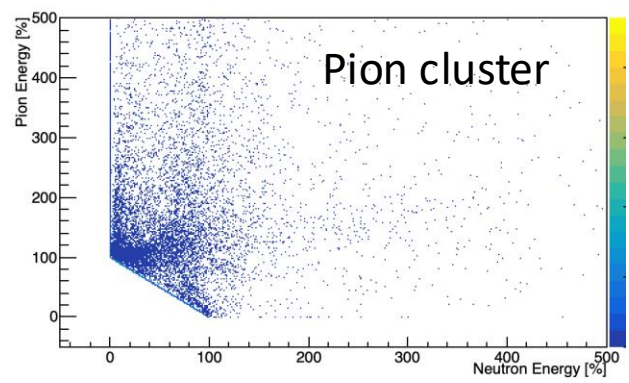
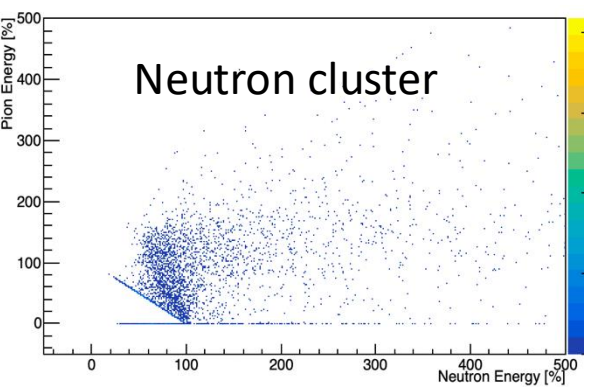
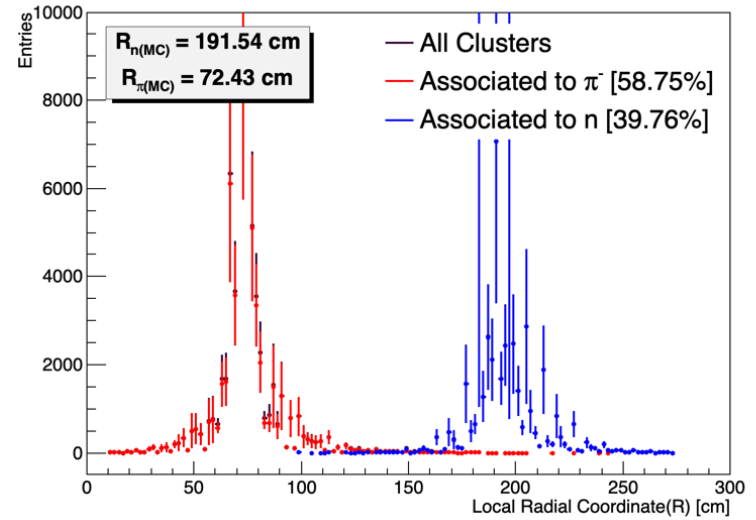
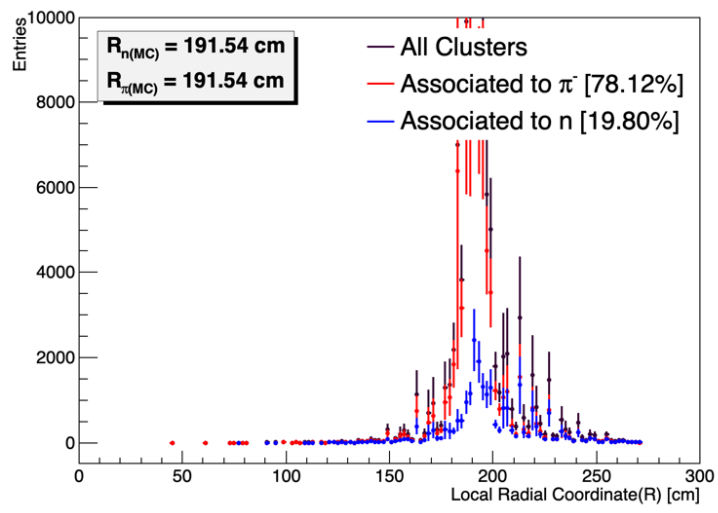
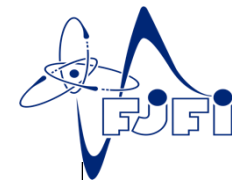
Cluster Reconstruction



$$E_{Pion(Neutron)}\% = \frac{\sum E_{Pion(Neutron)}^{HitContribution} / sampling\ fraction}{Cluster\ Energy} \times 100$$

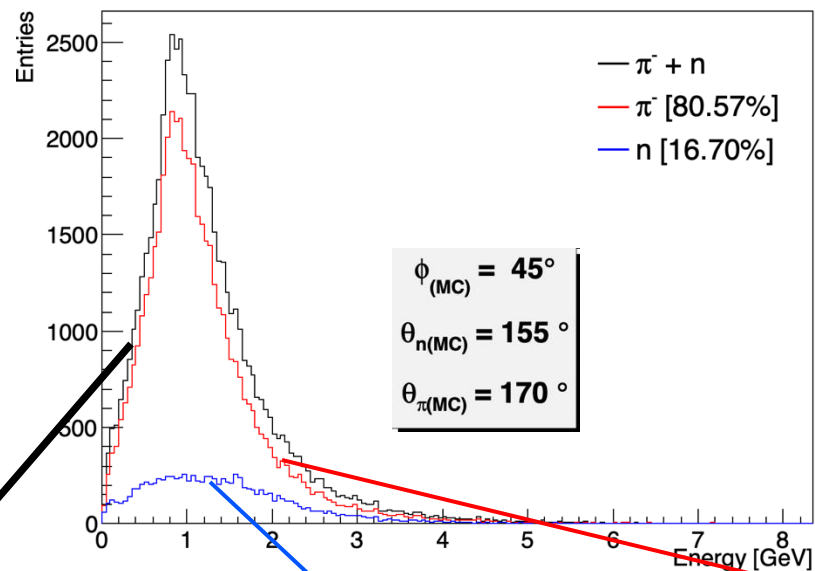
```
if (sim.getEnergy() < 0.1875e-3) continue; [time info not stored in simhits]
if (contrib.getTime() > 100) continue;
```

Cluster Reconstruction

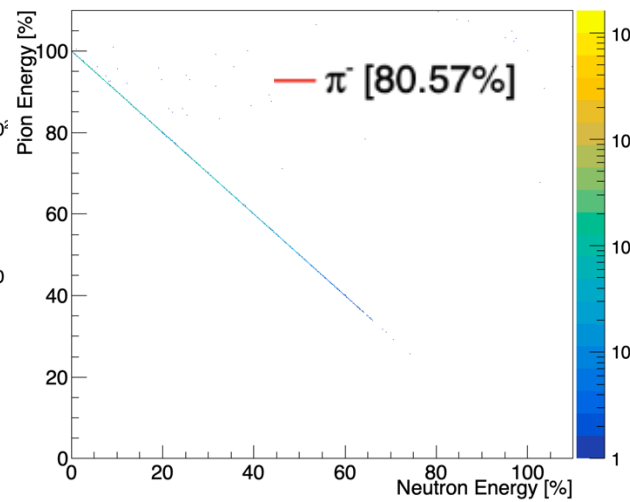
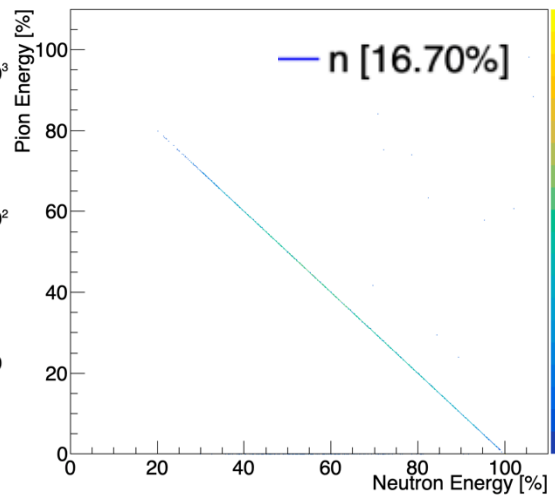
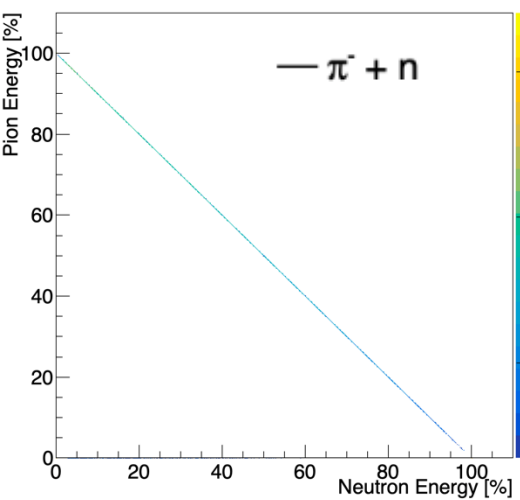


$$Pion(Neutron)Energy\% = \frac{\sum Pion(Neutron)RecohitEnergy}{ClusterEnergy} \times 100$$

Cluster Reconstruction



old



Thank You

