UConn-RIKEN Update 11/25

Alessio I, Gursimran K.

UCONN
UNIVERSITY OF CONNECTICUT

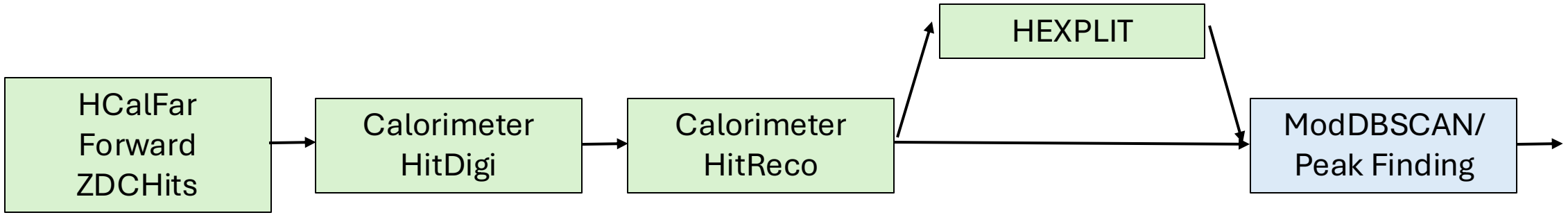
Overview

- We have been working on implementing our algorithms in EICrecon
- Primarily writing JAlgorithms, JFactories and plugins necessary to integrate them alongside the existing ZDC HCal algorithms
- Work on two different approaches
  - Modified DBSCAN
  - Peak Finding
- Modified DBSCAN provides a general algorithm to identify clusters in the data
  - Could be applied to more than just lambda
- Peak Finding is the much more restrictive option just for lambda reconstruction
- Much more work must happen before we are happy with the state of this group to share with collaborators

Reply from Miguel from UC Riverside

- We emailed Miguel to ask about the non-linearities in neutron EM shower in SiPM on tile
- His comments were twofold: one about the non-compensated nature of the SiPM on tile and an empirical one about neutron clustering
- As for the EM non-linearities, he was referring to how the Iron-Scintillator is non-compensated and how this must be algorithmically corrected
- For the neutron cluster, he has noticed that, due to the messy nature of the neutron clusters, algorithms will sometimes systematically discard the edge hits of the clusters

EICrecon



- CalorimeterHitCollection
- CalorimeterProtoClusterCollection

Modified DBSCAN

- ModDBSCAN is composed of two steps:
  - Step 1: Clusters are established where the energy of the core points is higher than the threshold
  - Step 2: The largest cluster is split into smaller ones using a change in energy density as you move outwards from the center of the cluster
- Uses its own JAlgorithm and JFactory
  - https://github.com/gursimrankainth/modDBSCAN
- The current version of modDBSCAN algorithm is only for step 1, once this is working well step 2 will be implemented.

Modified DBSCAN

- Overall Status: Still in debugging stage
  - Algorithm compiles with EICrecon build but issues arise when trying to use the plugin
  - Need to retune parameters for algorithm once it is running so that it performs well after HEXPLIT is applied
  - Need to add step 2 as well – will use another algorithm script to do this to prevent issues with integration with EICrecon framework (may also need to write a second config and factory script as well)
  - May also adjust step 2 so that the clusters are made based on change in energy density from the outside in instead of inside out

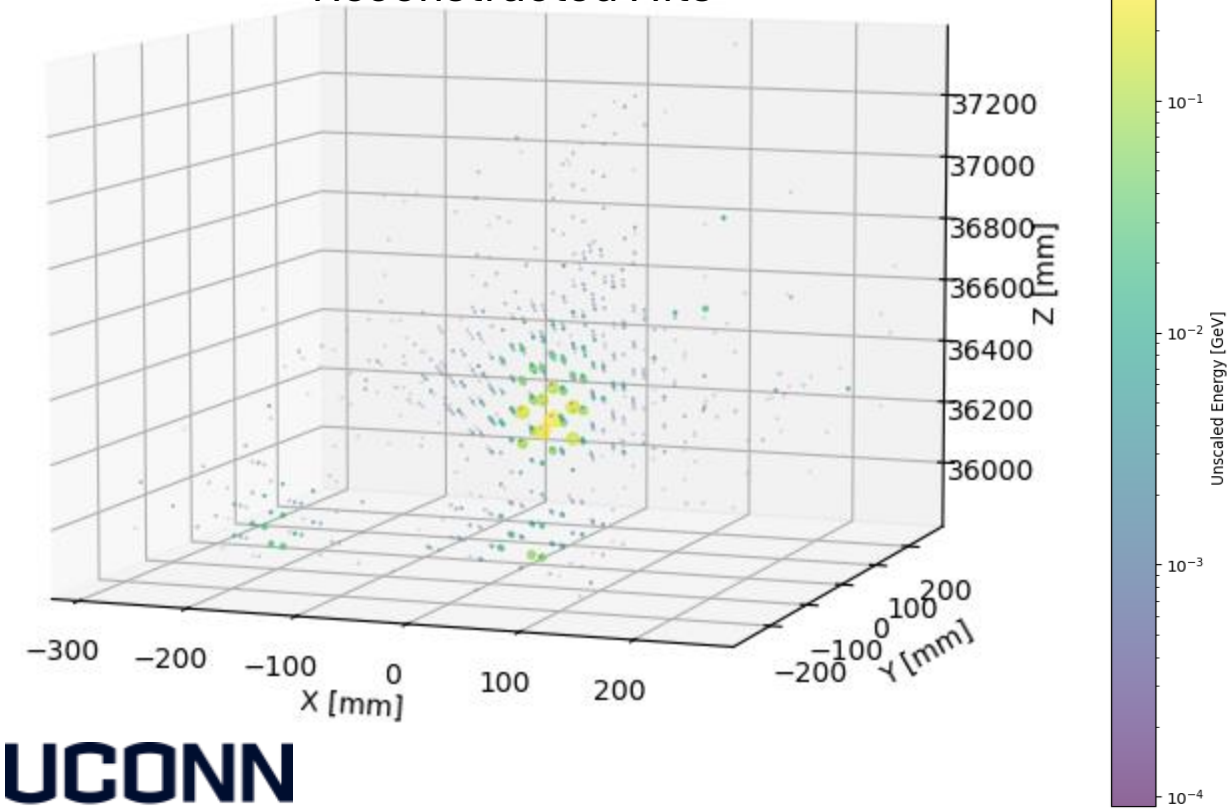
Peak Finding

- Peak Finding is composed of three stages:
  - Determining local energy maximums
  - Identifying and grouping local energy maximums
  - Expanding clusters from local energy maximums
- Previously, I used assumptions about the relative depths of gamma and neutrons to inform steps 2 and 3
  - Those assumptions were for ZDC with ECal+HCal, do not apply for ZDC with SiPM on tile only
- Currently step 1 is implemented and I am still working on steps 2 and 3
  - This will require a better familiarity with the data before we can decide what the best approach would be

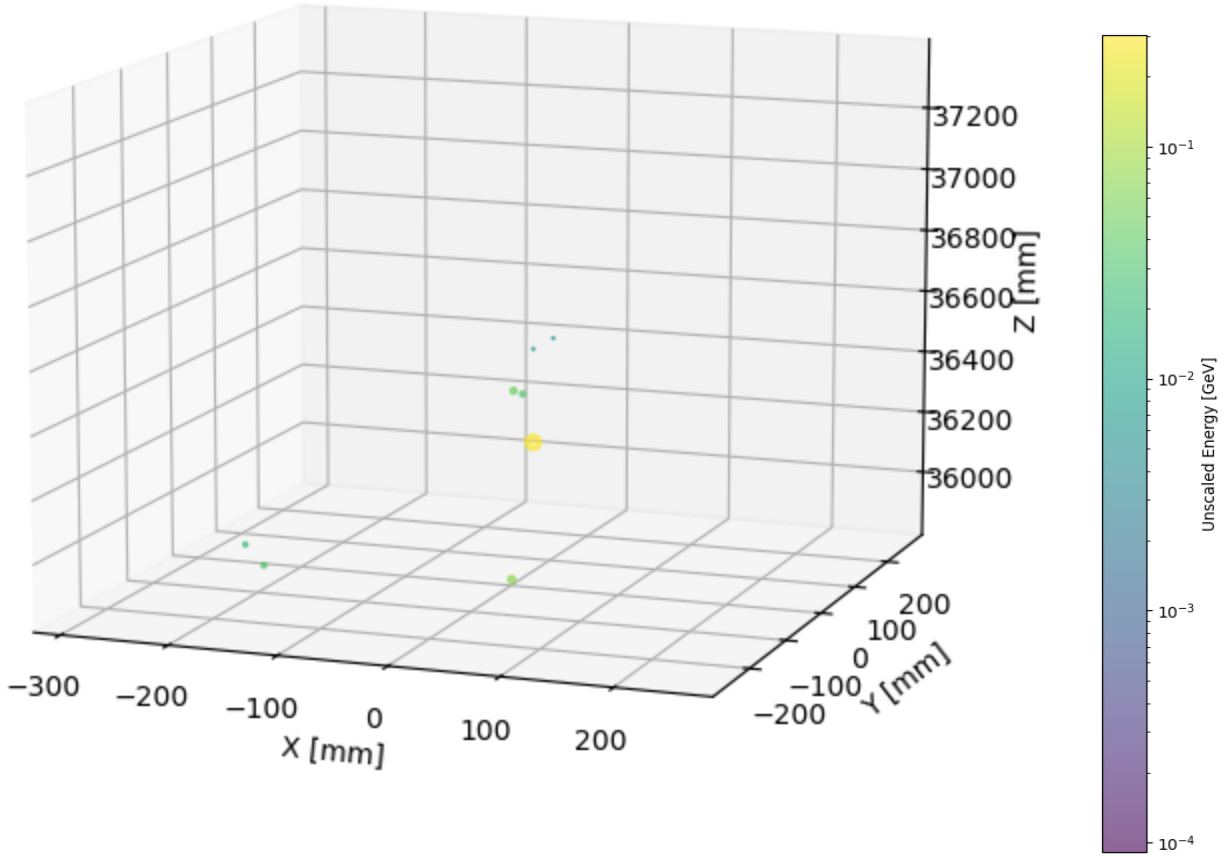
Local Maximums for ZDC (ECal+HCal)

- Lambda MC straight at ZDC
- Hits on the left are now the reconstructed hits
- Z axis weighting is log

Reconstructed Hits



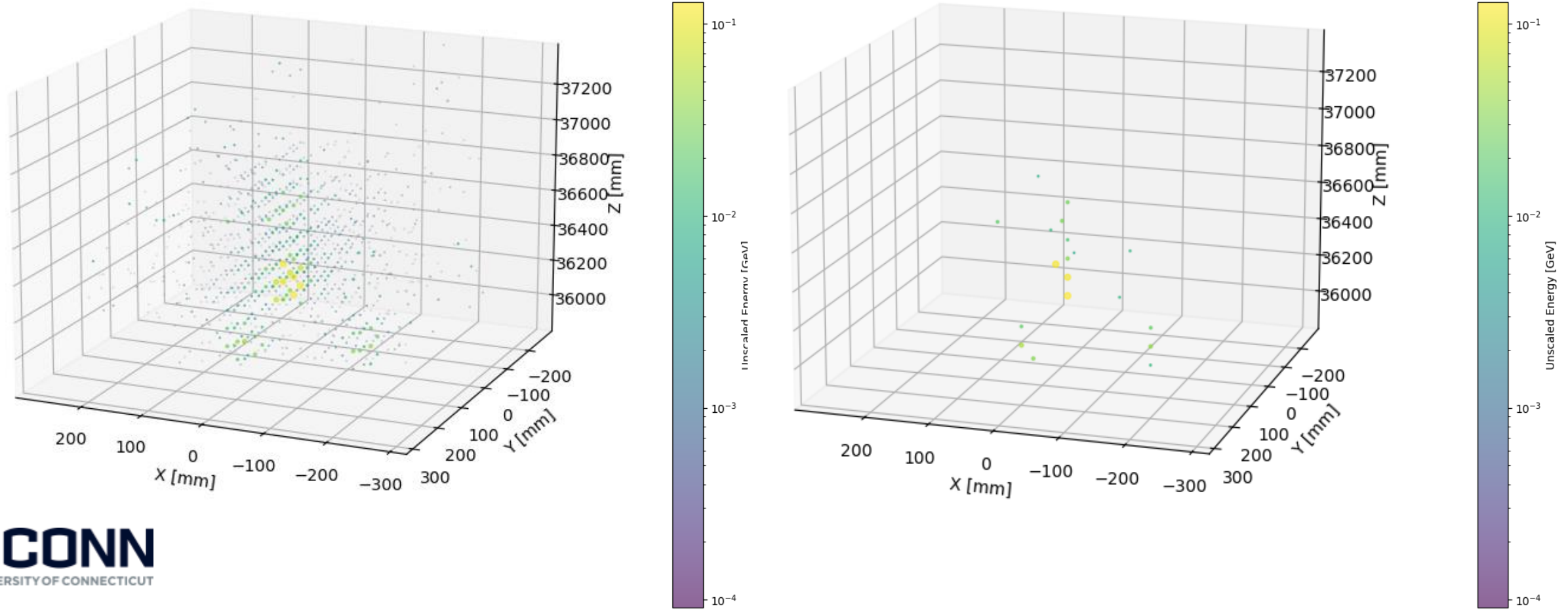
Local Maximum Hits



UCONN
UNIVERSITY OF CONNECTICUT

Local Maximums for ZDC (HCal)

- Lambda MC straight at ZDC - generated with only SiPM on tile



UCONN
UNIVERSITY OF CONNECTICUT

Identifying Gamma Clusters

- The easiest way to reconstruct the lambda is to identify the gamma clusters first and then isolate neutron from remaining hits
- Previously, a combination of ECal peaks + HCal peak would allow me to identify gamma peaks from neutron peaks
- As we are primarily interested in SiPM on tile only, a new approach is needed
  - We need a new analysis for event categorization
  - We need more simulations for statistics; maybe on JLab farm?
- Either: adjust algorithm values for SiPM on tile simulations or take a different approach

Outlook

- We will continue working on implementing our algorithms in EICrecon
  - Is our goal to implement these algorithms into the main branch of EICrecon or to get them in a state we can give to colleagues?
  - If the former, we will need to start speaking more to our colleagues in charge of EICrecon so that our code is up to their standards and conventions
- Once the algorithms are done, we can start comparing our algorithms to:
  - ImagingTopoCluster + RecoCoG: lowest level clustering possible
  - GNN from UC Riverside Team: We will have to ask Miguel and Sebouh for the weights and model
- Potentially implement complete lambda reconstruction with ZDC in EICrecon