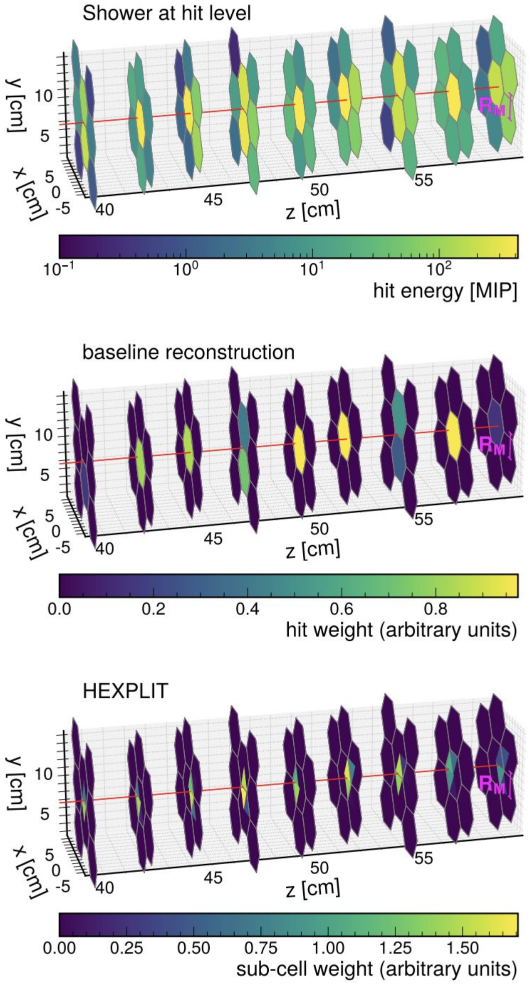
The HEXPLIT algorithm in EICrecon

Dr. Sebouh J. Paul
12/12/2023

Intro

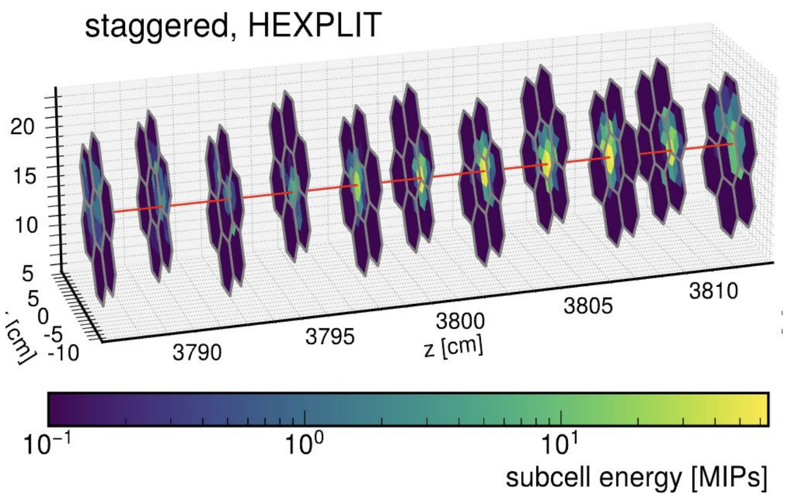
- HEXPLIT* was developed to improve the position reconstruction in sampling calorimeters by taking advantage of staggered layouts
  - Improves spatial resolution by a factor of ~2
- The HCal Insert and the SiPM-on-tile ZDC are examples of detectors that can take advantage of this algorithm
- The algorithm was originally written as a stand-alone program in Python
- Now a C++ implementation has been added to the EICrecon on a branch**

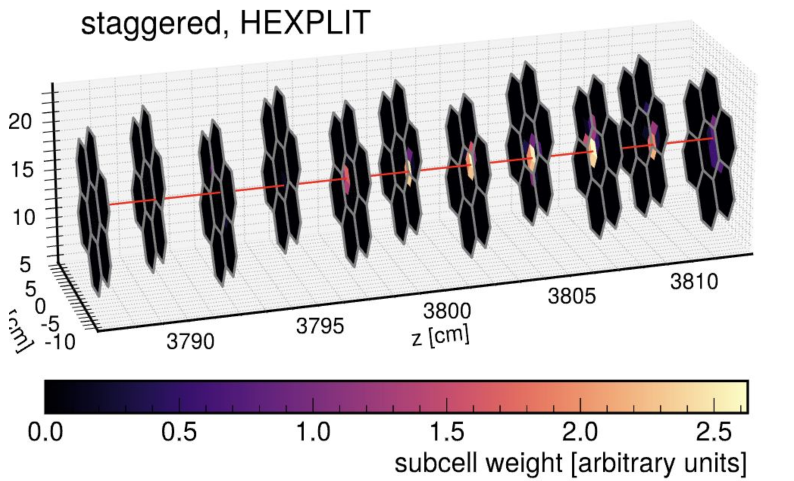


*arXiv:2308.06939
** https://github.com/eic/EICrecon/blob/sipmzdc/src/algorithms/calorimetry/HEXPLIT.cc

Two components to HEXPLIT (separate algorithms)

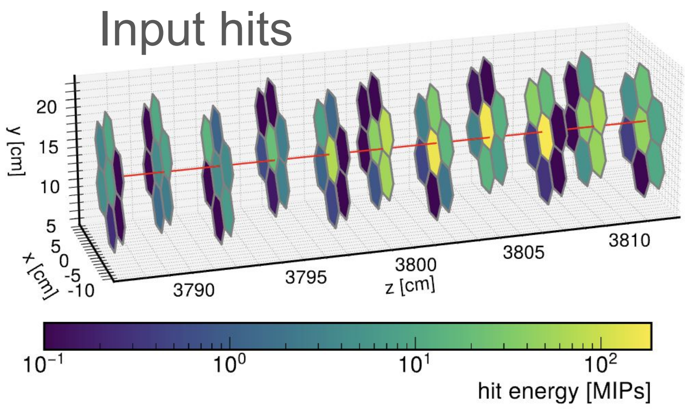
- Subcell reweighting
  - Creating subcell hits with energies determined by those of the overlapping hits in neighboring layers,
- Position determination
  - Determine the position of the center of the shower by weighting each (subcell) hit by the log of its energy, with a cutoff determined by the total shower energy
- These algorithms are respectively called **HEXPLIT** and **LogWeightReco** in the EICrecon
  - LogWeightReco can run independent of whether the input hits are normal hits or subcell hits
- These algorithms and their factories have been added in EICrecon

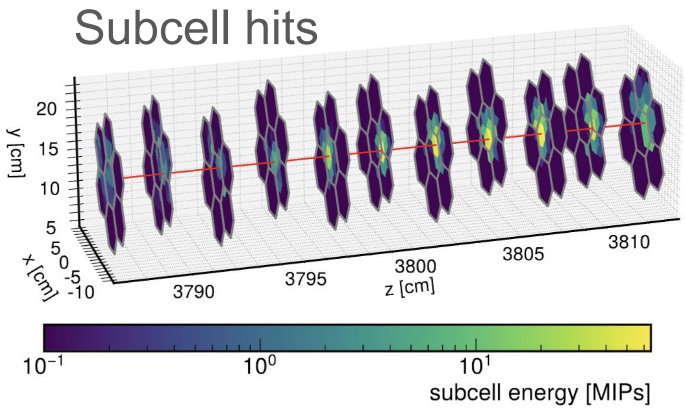




HEXPLIT algorithm

Core Portion of Neutron Shower





Formula for subcell reweighting

$$W_i = \prod_{j=1}^{N-1} \max(E_j, \delta),$$

Product over overlapping cells, j, in neighboring layers

$$E_i = E_{\text{tile}} W_i / \sum_j W_j .$$

Energy in a given subcell, i

δ = energy threshold, set to 1 MIP.

LogWeightReco algorithm

Core Portion of Neutron Shower

Reconstruct shower from (subcell) hits

$$\vec{x}_{\text{recon}} = \frac{\sum_{i \in \text{subcells}} \vec{x}_i w_i}{\sum_{i \in \text{subcells}} w_i}$$

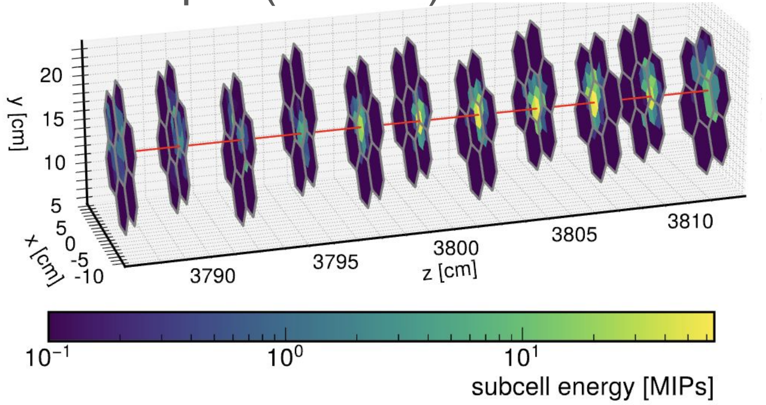
$$w_i = \max\left(0, w_0 + \ln \frac{E_i}{E_{\text{tot}}}\right)$$

w_0 is cutoff parameter, fine-tuned as a function of the reconstructed particle energy

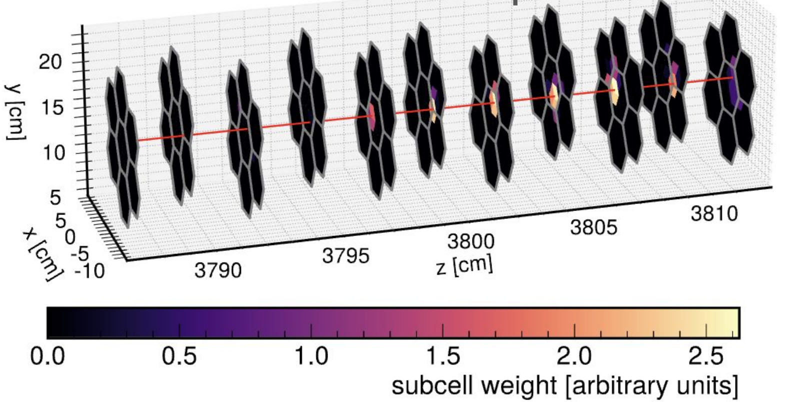
$$w_0 = \mathtt{w0\_a} + \mathtt{w0\_b} \log \frac{E_{\text{recon}}}{\mathtt{E0}} + \mathtt{w0\_c} \left(\log \frac{E_{\text{recon}}}{\mathtt{E0}}\right)^2$$

$$E_{\text{recon}} = \sum_{i \in \text{hits}} E_i / \mathtt{sf}$$

Input (subcell) hits

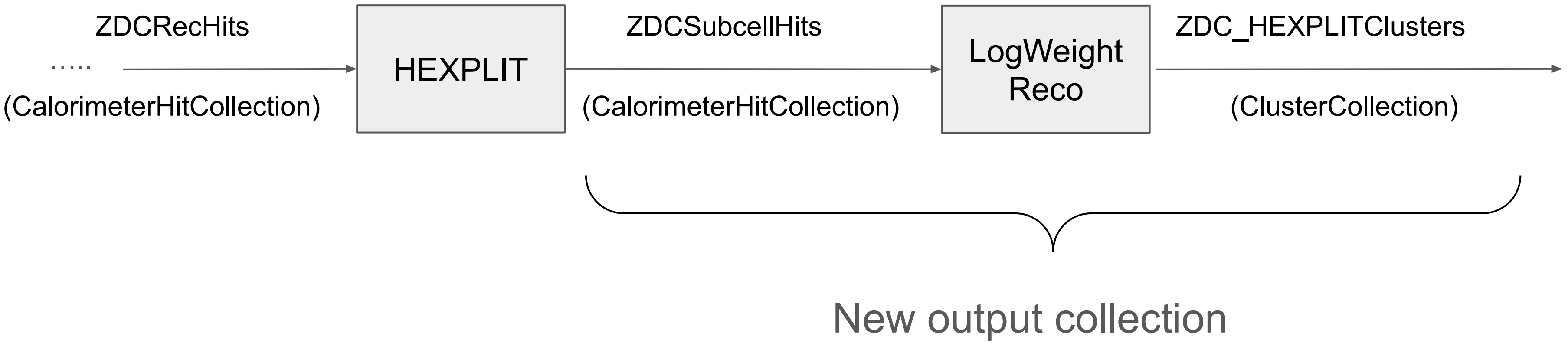


Contribution to cluster position



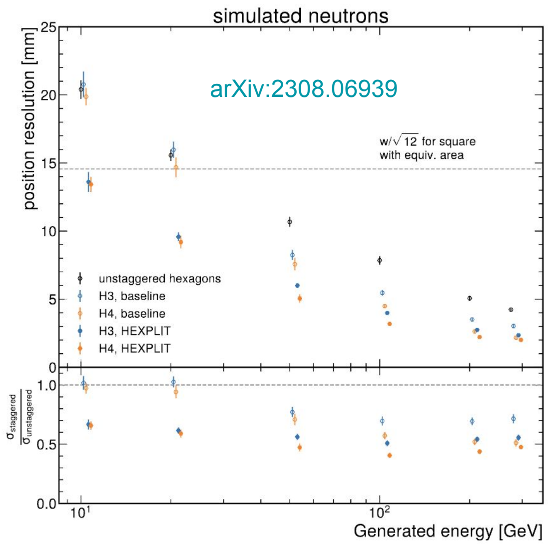
ZDC plugin (for SiPM-on-tile ZDC)

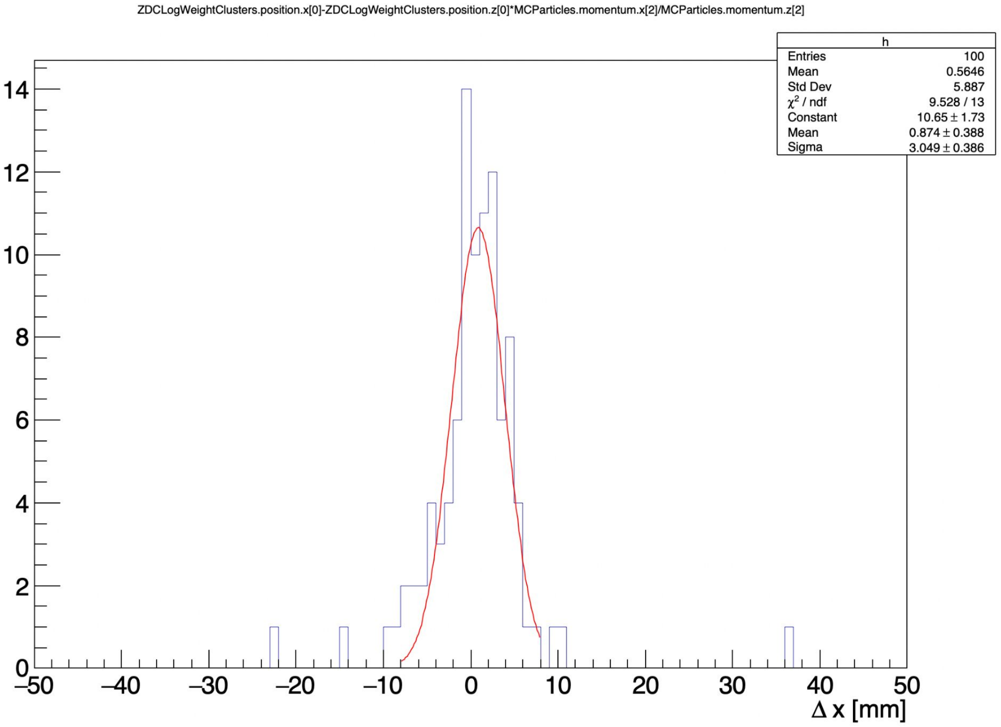
Added these algorithms' factories in the chain of factories in the ZDC SiPM-on-tile's reconstruction



Performance

Position obtained with C++ implementation is consistent with the results we got earlier with the Python implementation.





Summary

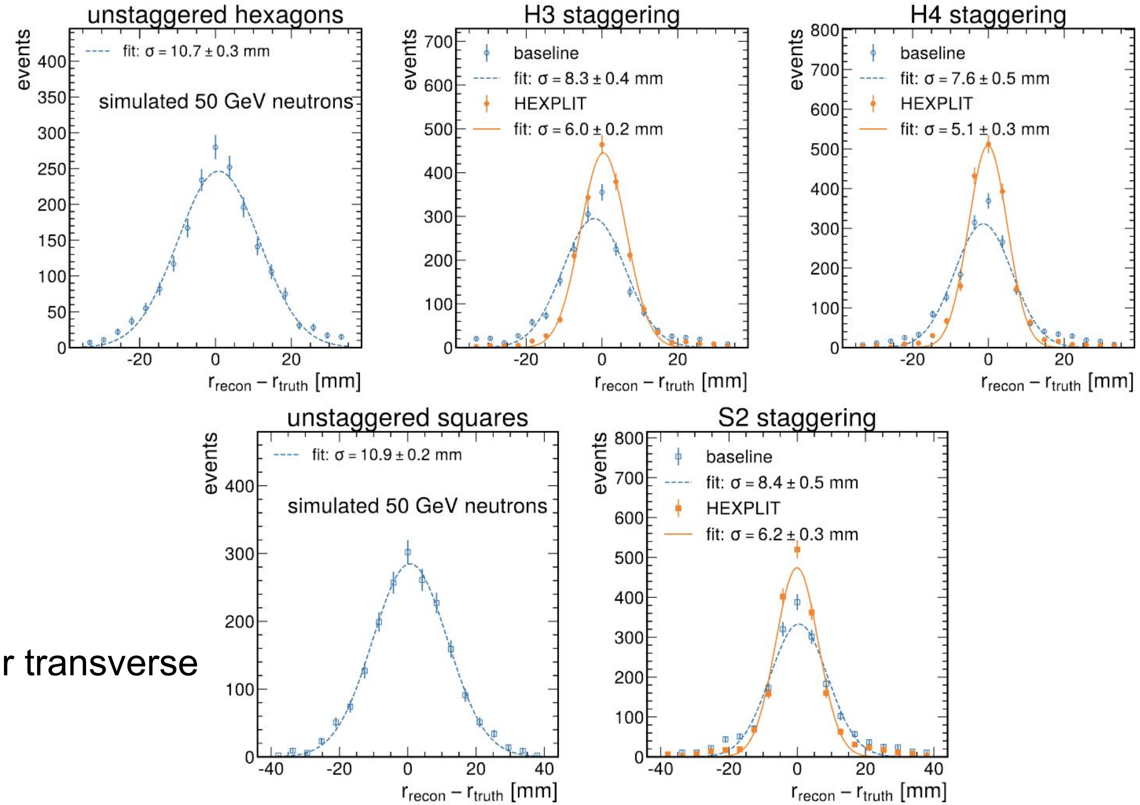
- HEXPLIT has been incorporated into EICrecon
- Shower-position reconstruction is now its own separate algorithm
- Both algorithms could be used in the context of either the SiPM-on-tile ZDC or the calorimeter insert.
- These have been included in the sipmzdc branch; to be pull-requested into the main branch soon.

Backup slides

Neutron-shower performance for the ZDC-like* calorimeter

- Factor of 2 improvement

arXiv:2308.06939



*Simulations in this paper used much larger transverse dimensions to avoid edge effects.

Energy dependence of position resolution

- H4 staggering improves the resolution by up to 60%, when utilizing the HEXPLIT algorithm