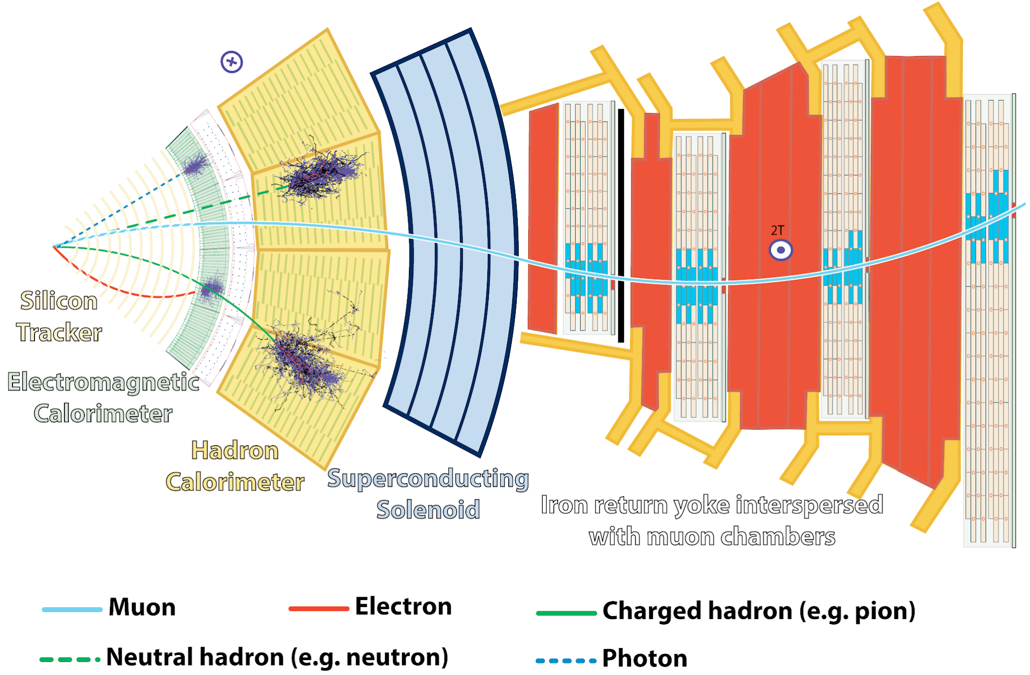
SPVCNN for clustering in HGCAL and HCAL

MIT: Jeff Krupa, Patrick McCormack, Zhijian Liu, Phil Harris, Song Han
UW: Alex Schuy, Haoran Zhao, Haotian Tang,, Shih-Chieh Hsu, Scott Huack

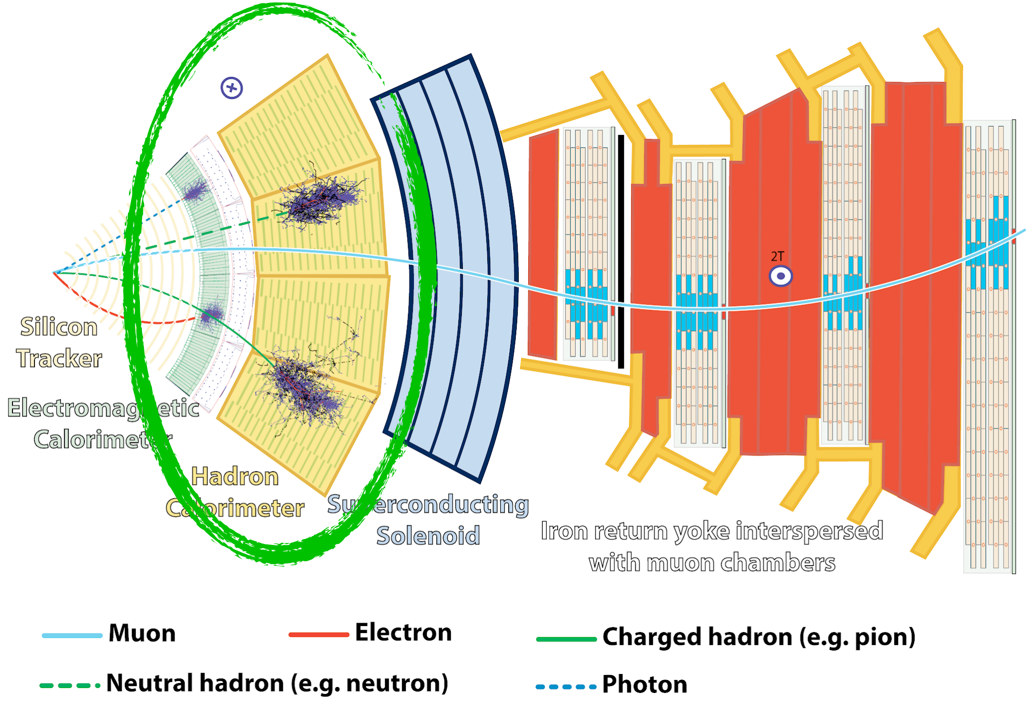
Fast Machine Learning Workshop 2023
25.09.2023

Overview



2T
Silicon
Tracker
Electromagnetic
Calorimeter
Hadron
Calorimeter
Superconducting
Solenoid
Iron return yoke interspersed
with muon chambers
Muon
Electron
Charged hadron (e.g. pion)
Neutral hadron (e.g. neutron)
Photon

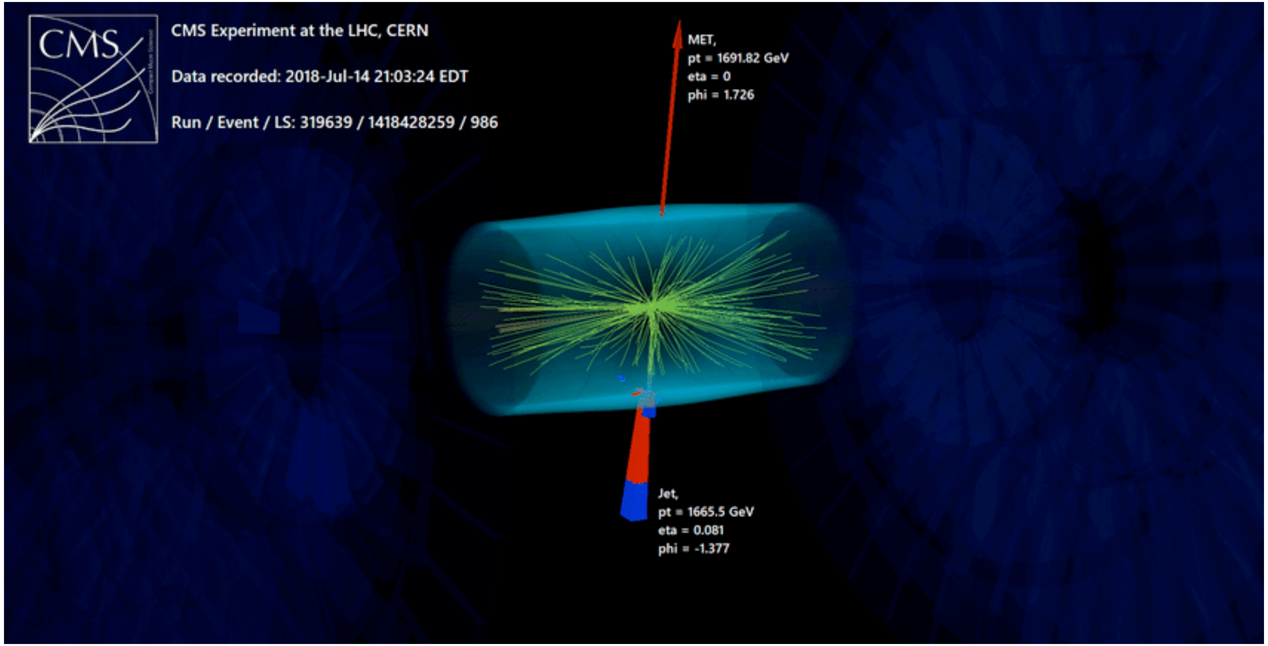
Overview



2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with muon chambers
Muon
Electron
Charged hadron (e.g. pion)
Neutral hadron (e.g. neutron)
Photon

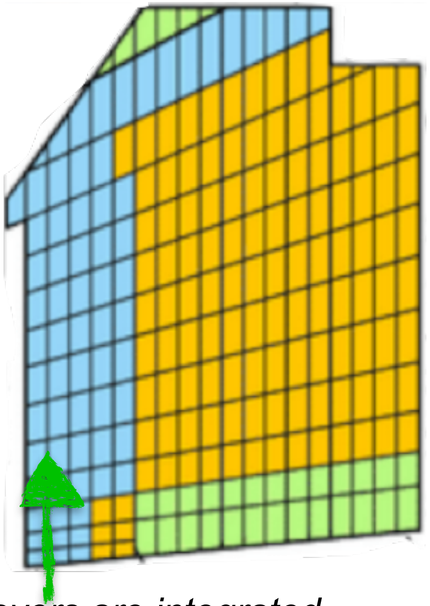
Hadron calorimeter

- Good resolution is important for physics observables



Approaches to clustering

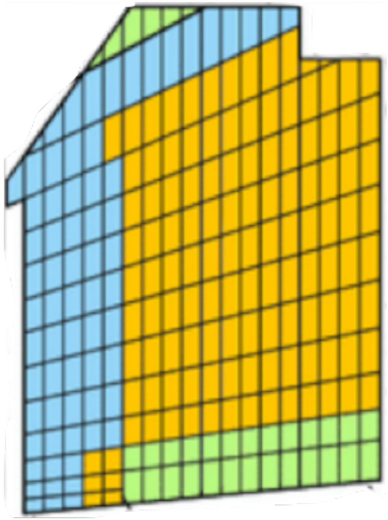
- Hadron calorimeter clustering is separated layer-by-layer (ParticleFlow "PF" clustering)
    - underutilizes depth and shape info available
- ML provides a natural way to introduce the depth/timing profile into clustering algorithms
    - could help with e.g. pileup suppression



Layers are integrated into depths

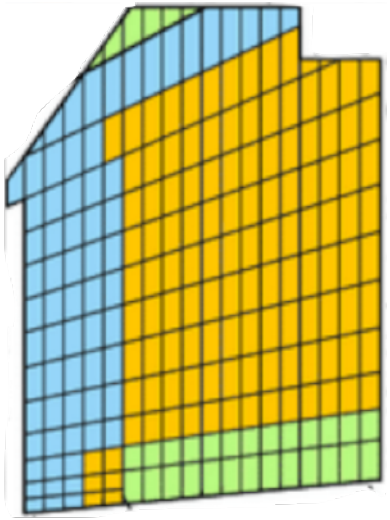
Detectors

HCAL
O(10k) channels

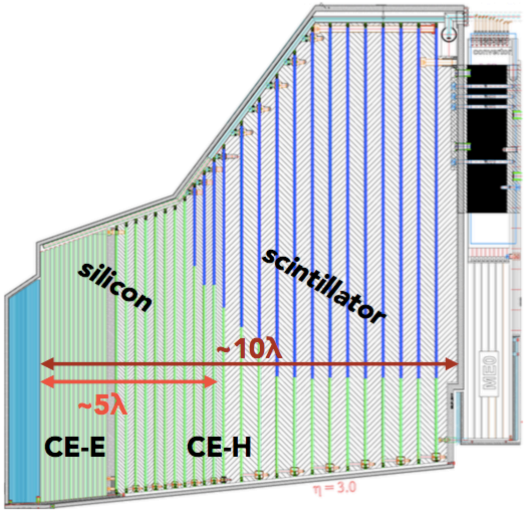


Detectors

HCAL
O(10k) channels

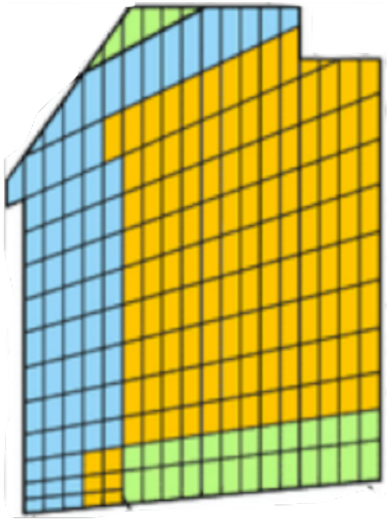


HGCAL
O(6M) channels

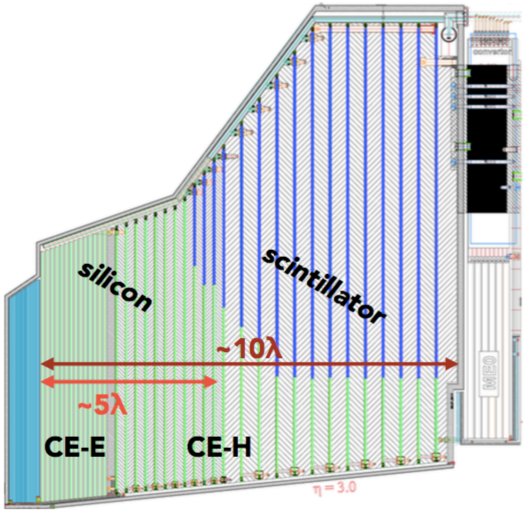


Detectors

HCAL
O(10k) channels



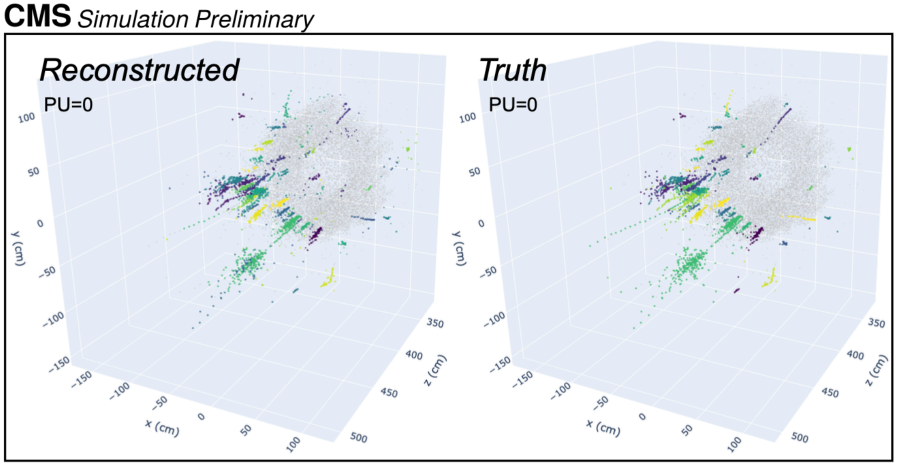
HGCAL
O(6M) channels



*Which **high throughput** algorithms can provide **good physics performance** in these detectors?*

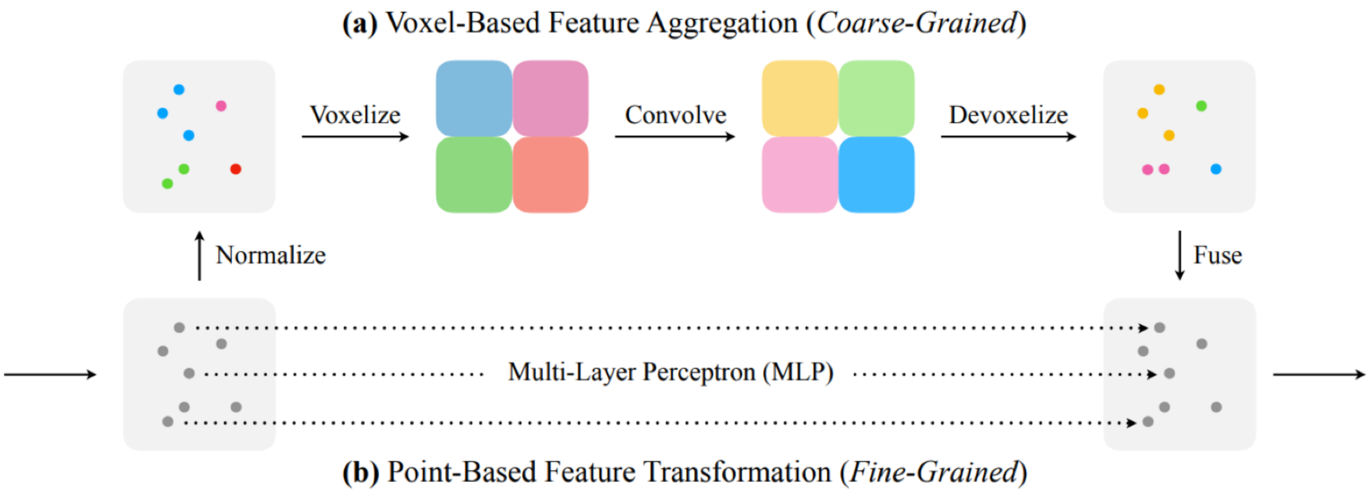
Graph Neural Networks

- Graphs have been successfully applied to this problem (eg. [GravNet]())
- Also rule-based methods (eg. [TICL]())
- We are approaching the problem with a computationally-efficient model for convolutions on sparse data



SPVCNN

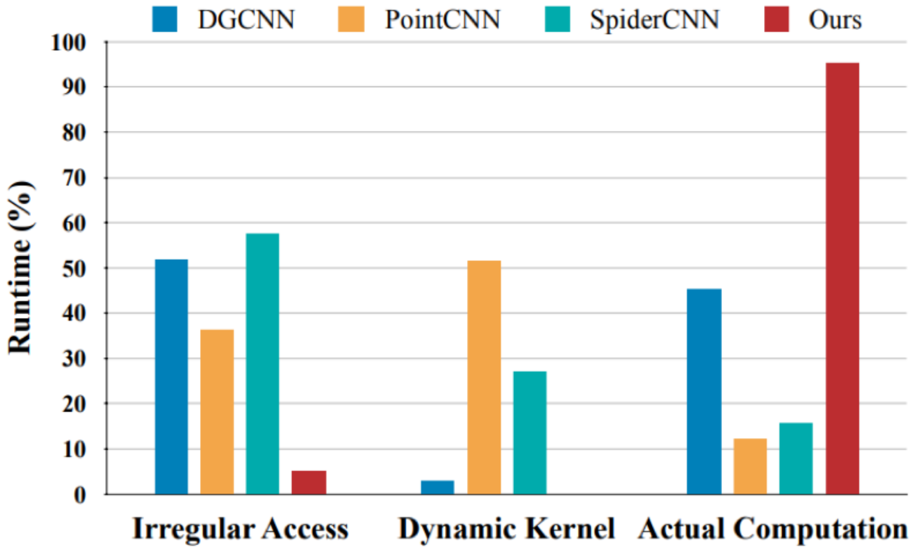
2007.16100



- Proven for semantic and instance segmentation in 3D vision tasks
  - low latency, high accuracy constraints (driverless cars)
- Sparse points are first voxelized and then convolved
  - HCAL event embedded into a 6D space using SPVConv blocks

SPVCNN

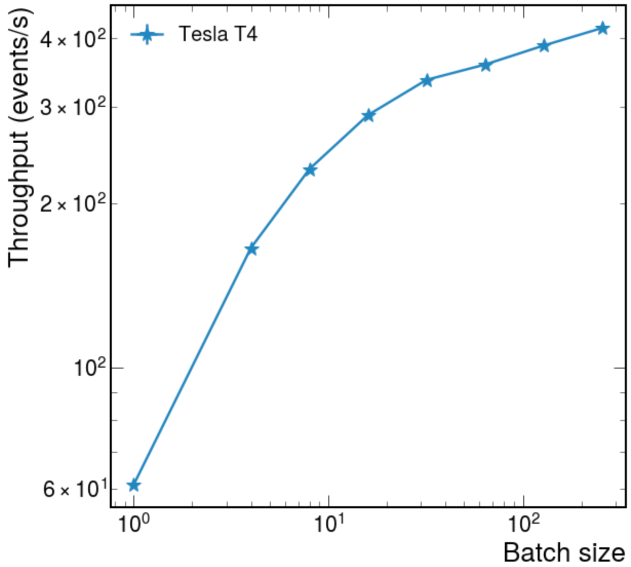
2007.16100



- **SPVCNN** is memory and computation efficient compared to leading CNNs (e.g. *voxel models* and *point cloud models*)
  - Low memory and computational overhead
  - No need to construct graph adjacency matrix

Throughput

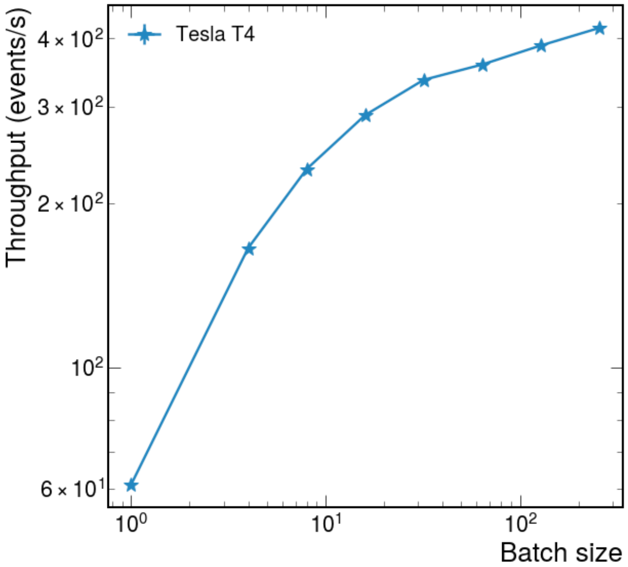
- SPVCNN can achieve 420 inferences/second on a single T4 GPU



- Using a GPU, SPVCNN is ~16x faster to form clusters than PF clustering on CPU
- O(1k) CPU threads can be served by a single GPU before GPU limits the workflow

Throughput

- SPVCNN can achieve 420 inferences/second on a single T4 GPU



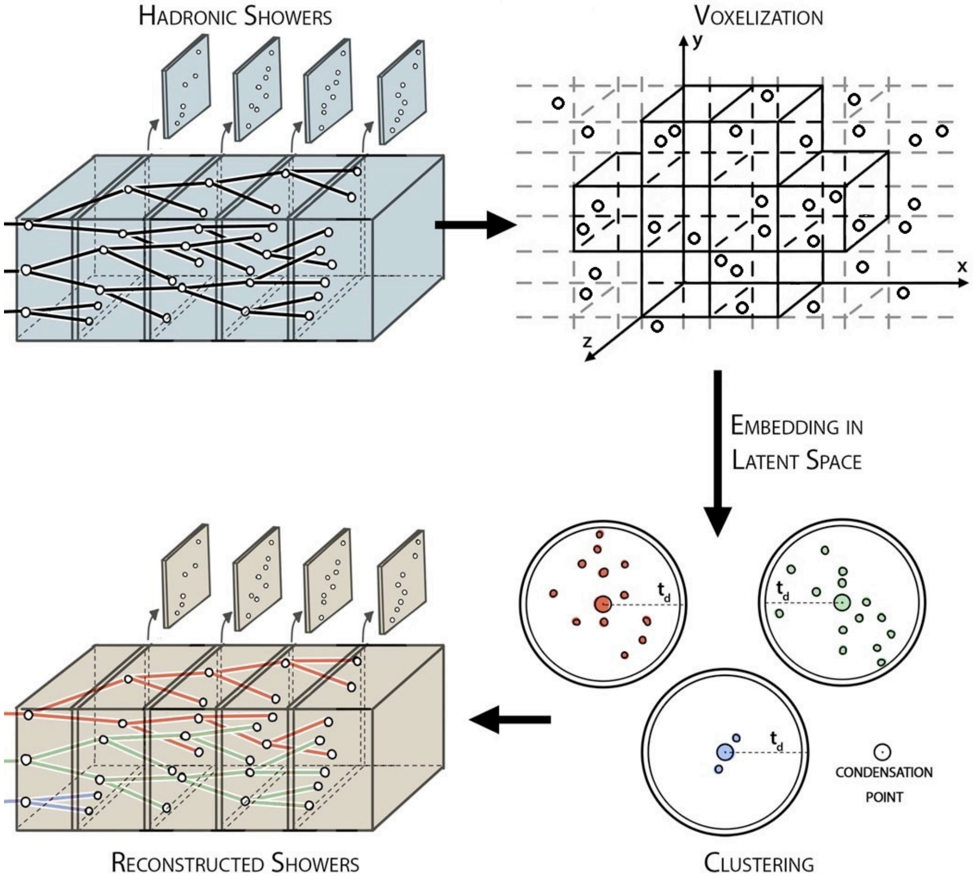
- Using a GPU, SPVCNN is ~16x faster to form clusters than PF clustering on CPU
- O(1k) CPU threads can be served by a single GPU before GPU limits the workflow

SPVCNN provides speedup on GPU and is integrated into CMS software

Object condensation loss

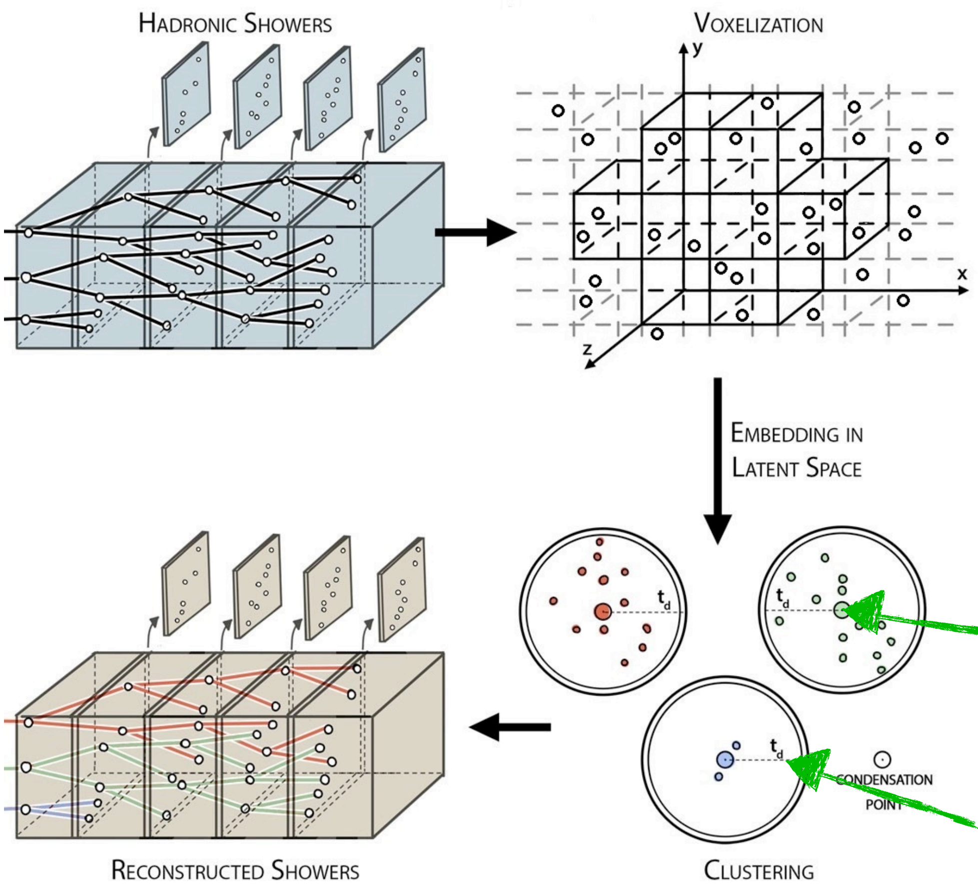
- We use the object condensation loss: [2002.03605]
- HCAL event is first embedded into a space using SPVCNN convolution blocks
    - Each hit is assigned a “condensation score” by the network
    - Hits are then ranked in descending condensation score, and assigned to condensation points (forming clusters)
    - Two noteworthy hyperparameters $\{t_d, t_\beta\}$
    - Loss is weighted by cluster energy

Embedding



Hadronic Showers
Voxelization
y
x
z
Embedding in
Latent Space
t_d
t_d
t_d
Condensation
Point
Clustering
Reconstructed Showers

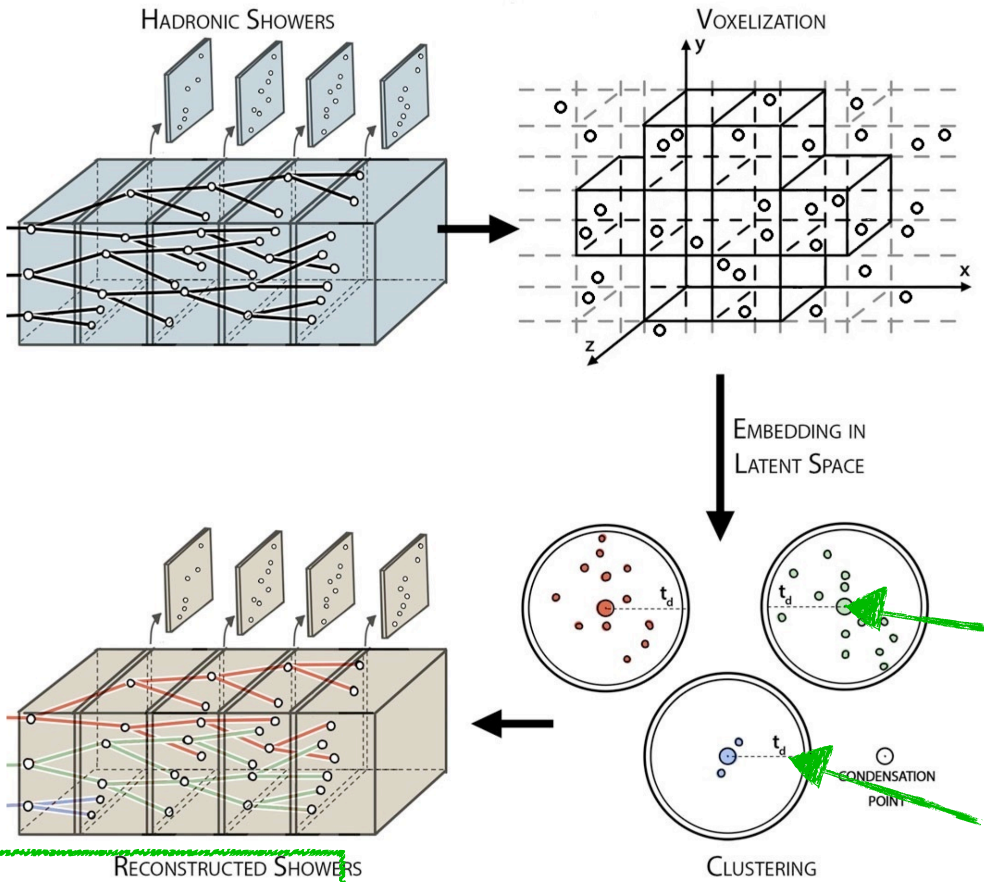
Embedding



t_β = smallest β value considered as a condensation point

t_d = max radius around condensation points

Embedding



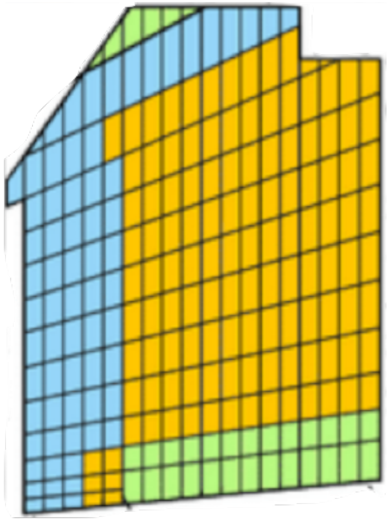
t_β = smallest β value considered as a condensation point

t_d = max radius around condensation points

hits are clustered inside embedded space

Results

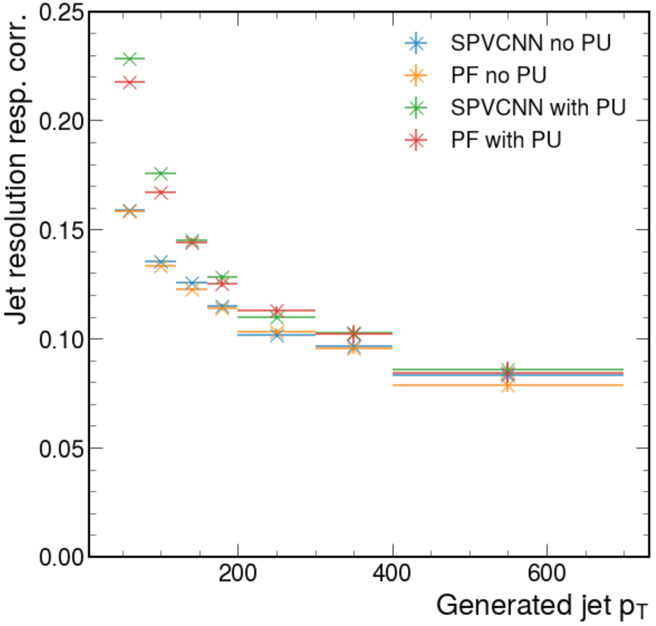
HCAL
O(10k) channels



HGCAL
O(6M) channels

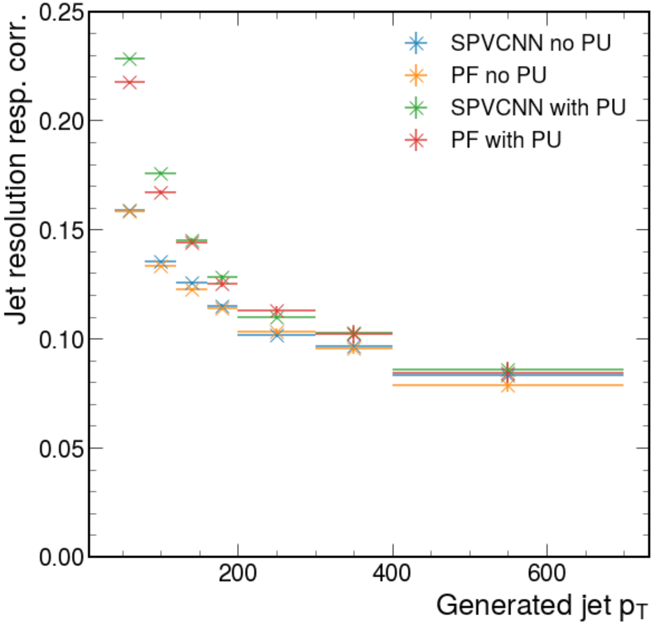
HCAL results: jet resolution

- Jet energy resolution for AK4 jets (note: no re-derived corrections)



HCAL results: jet resolution

- Jet energy resolution for AK4 jets (note: no re-derived corrections)

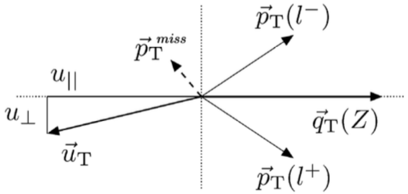


SPVCNN performs similarly in jet resolution to generic clustering. Currently re-deriving corrections

HCAL results: MET

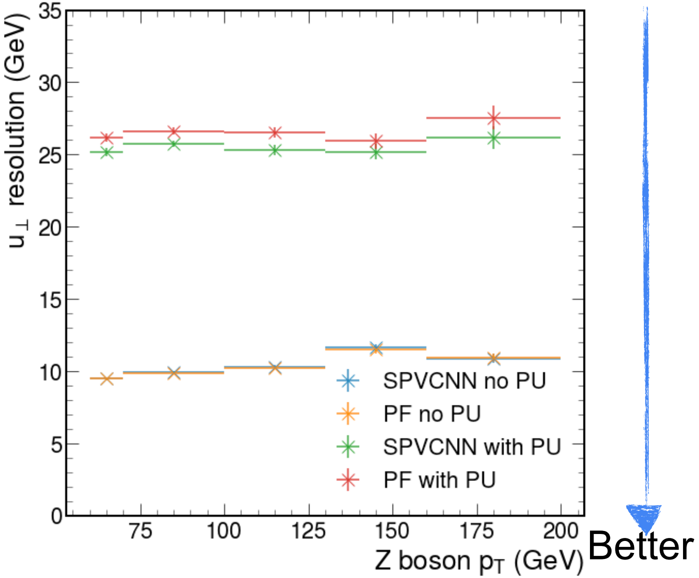
- How does SPVCNN clustering affect global observables like MET ?
    - generated Z(μμ)+jet in Run 3 with and without pileup to measure this

HCAL results: MET

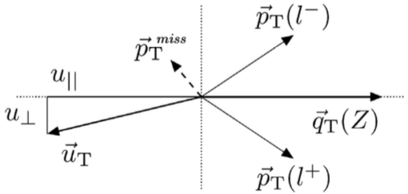


MET response = 1 - $\langle u_\parallel + Z_{pT}\rangle / \langle Z_{pT}\rangle$
$u_\parallel$ resolution = $\sigma(u_\parallel + Z_{pT})$
$u_\perp$ resolution = $\sigma(u_\perp)$

- How does SPVCNN clustering affect global observables like MET ?
  - generated Z(μμ)+jet in Run 3 with and without pileup to measure this

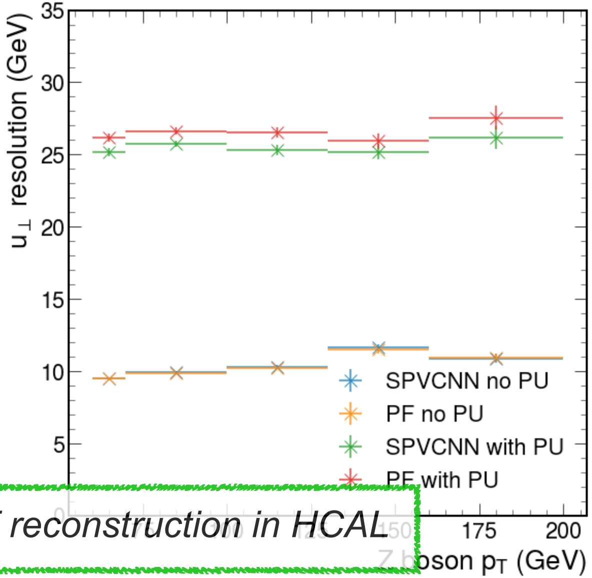


HCAL results: MET



MET response = 1 - $\langle u_\parallel + Z_{pT} \rangle / \langle Z_{pT} \rangle$
$u_\parallel$ resolution = $\sigma(u_\parallel + Z_{pT})$
$u_\perp$ resolution = $\sigma(u_\perp)$

- How does SPVCNN clustering affect global observables like MET ?
  - generated Z(μμ)+jet in Run 3 with and without pileup to measure this



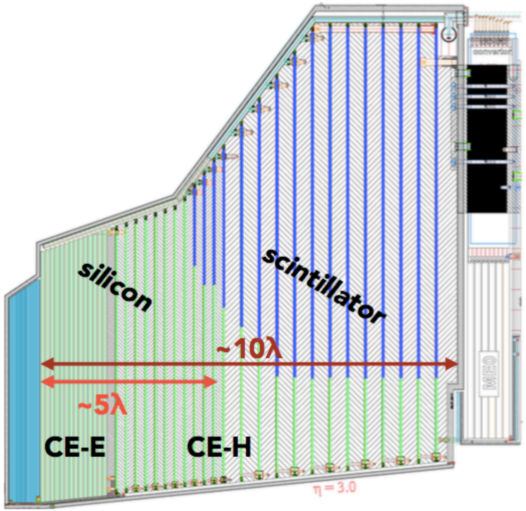
Better

SPVCNN performs similarly in MET reconstruction in HCAL

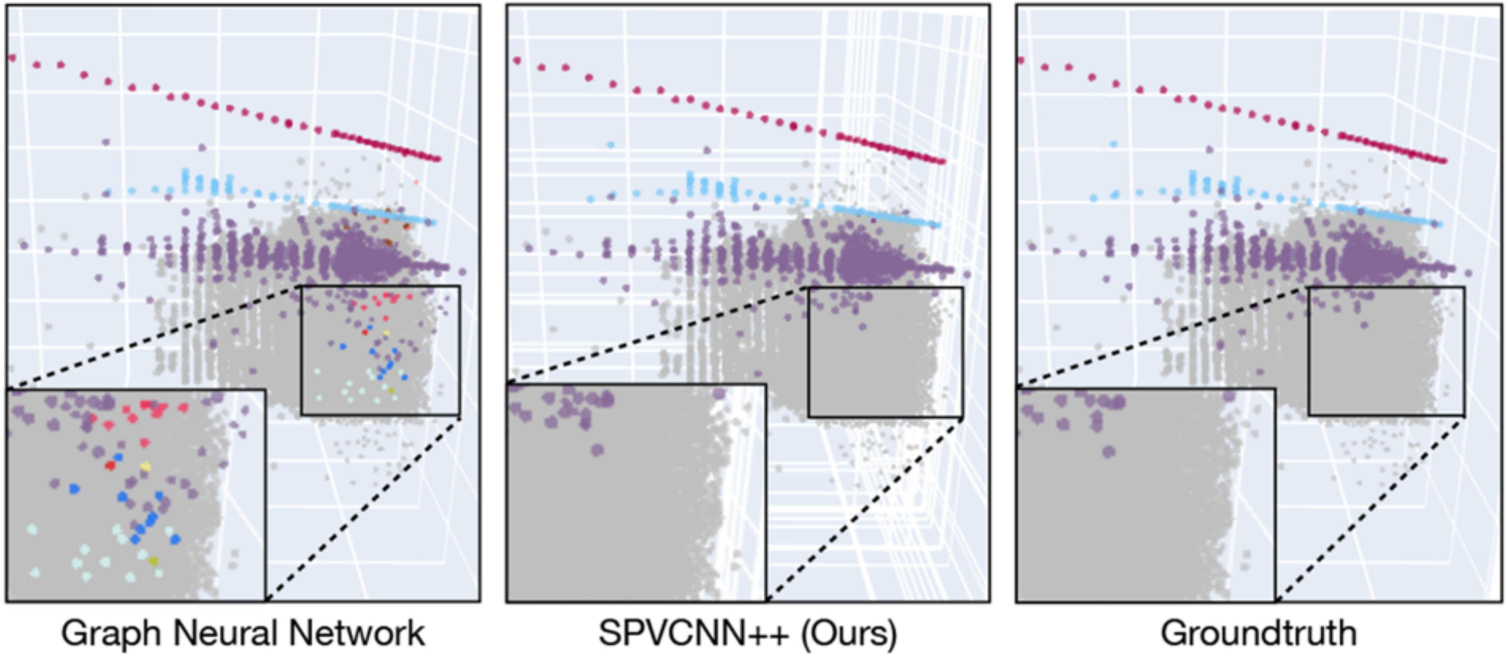
Detectors

HCAL
O(10k) channels

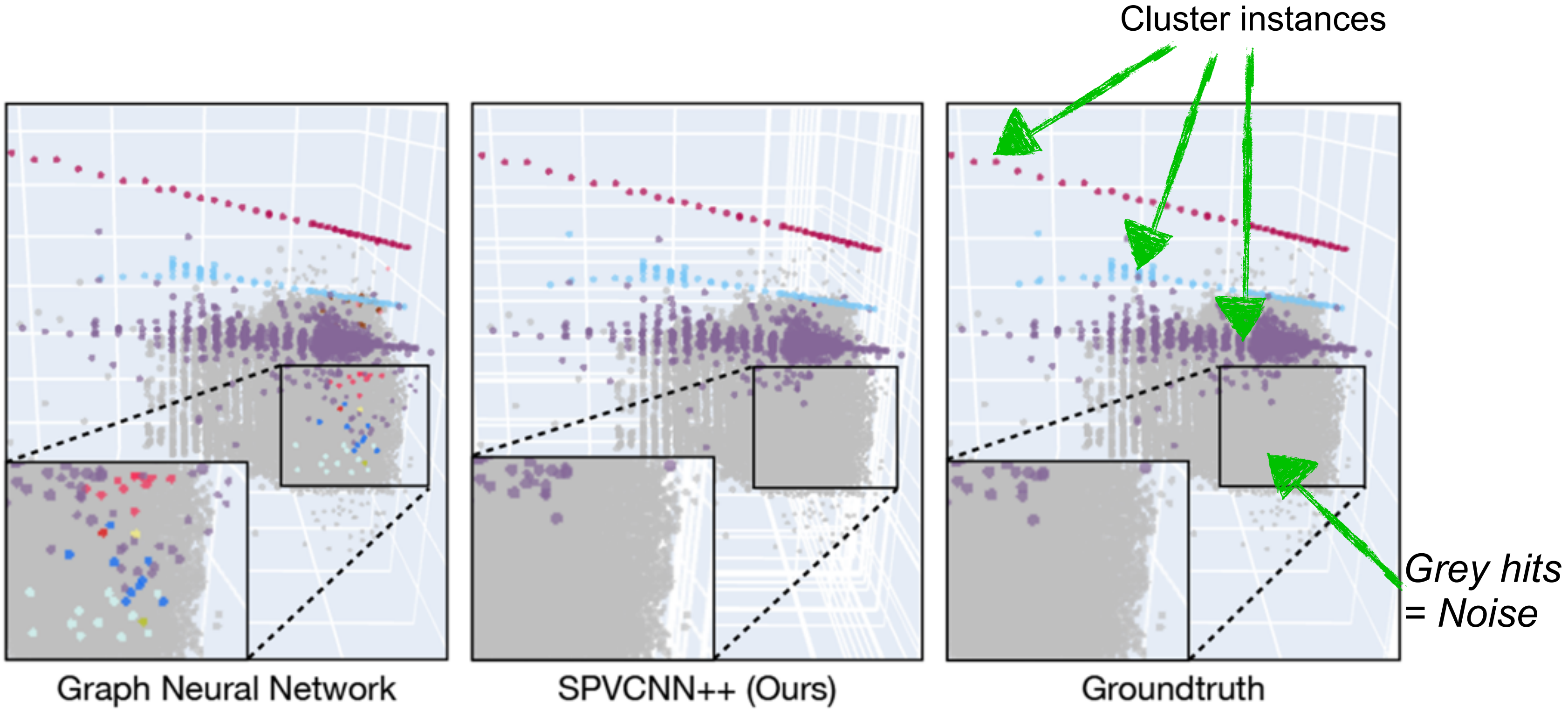
HGCAL
O(6M) channels



HGCAL Results: event display



HGCAL Results: event display



HGCAL Results: metrics

mIoU *= fraction of hits correctly identified as noise*
SQ *= overlap between reco-truth clusters for matched pairs*
RQ *= fraction of clusters that were matched.*
PQ *= SQ*RQ*

Method	mIoU	SQ	RQ	PQ
GravNet	0.93	0.89	0.74	0.69
GravNet (optimized)*	0.93	0.90	0.83	0.76
SPVCNN++	**0.98**	**0.92**	**0.85**	**0.80**

** optimized = version of GravNet model tuned to maximize these metrics*

HGCAL Results: metrics

mIoU *= fraction of hits correctly identified as noise*
SQ *= overlap between reco-truth clusters for matched pairs*
RQ *= fraction of clusters that were matched.*
PQ *= SQ*RQ*

Method	mIoU	SQ	RQ	PQ
GravNet	0.93	0.89	0.74	0.69
GravNet (optimized)*	0.93	0.90	0.83	0.76
SPVCNN++	**0.98**	**0.92**	**0.85**	**0.80**

SPVCNN performs well on HGCAL according to metrics used in clustering tasks

** optimized = version of GravNet model tuned to maximize these metrics*

Conclusions

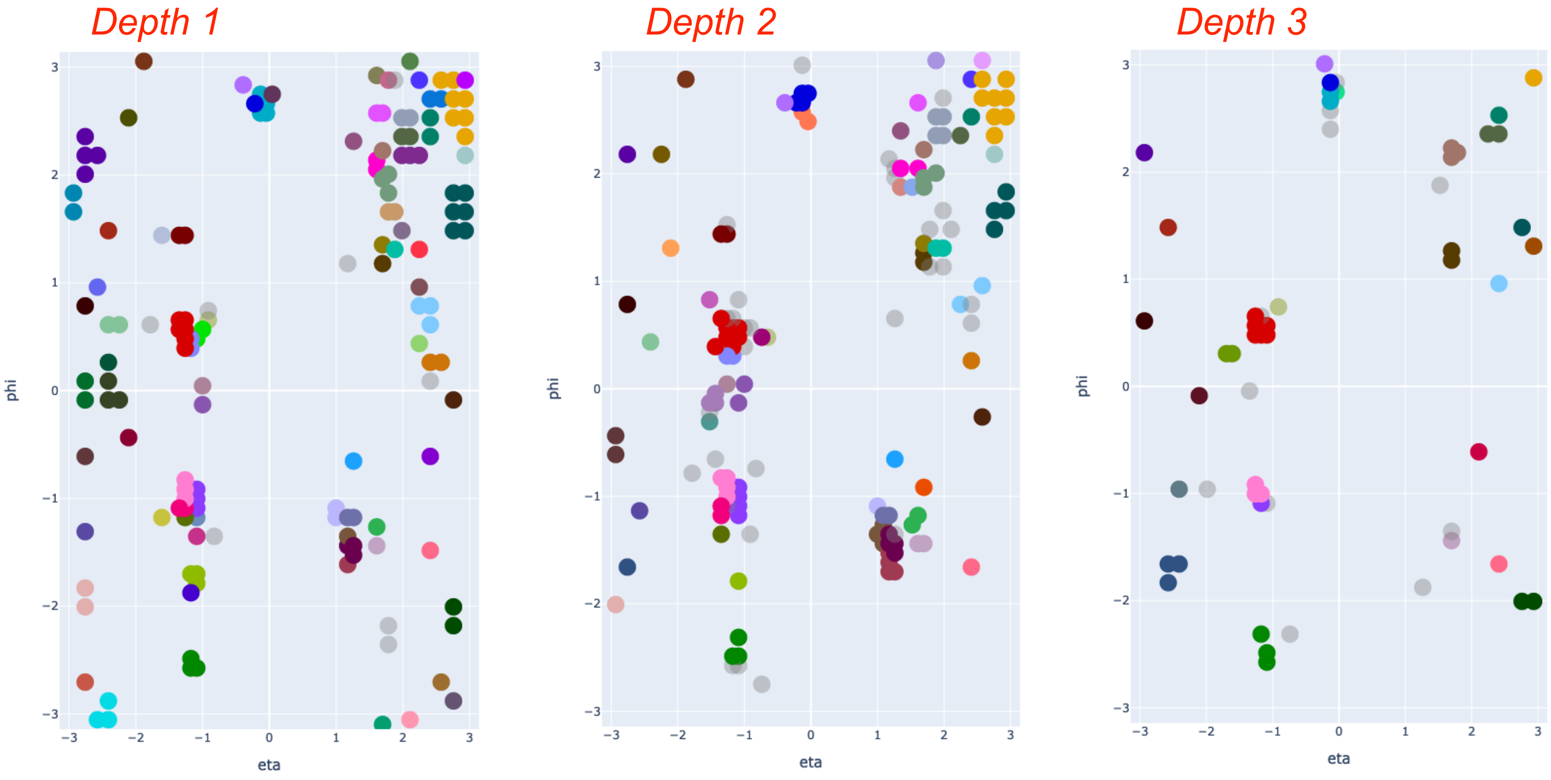
1. We introduced a **memory-efficient model** (SPVCNN) for clustering
   a. **High throughput** implementation on **GPU**
2. Clustering with SPVCNN yields physics performance **compatible with GravNet** for HGCAL and compatible with **generic PF clustering** for HCAL
3. Future plans:
   a. Can be deployed in CMS soon
   b. Finalize physics corrections and computing measurements
   c. Goal: Implementation for **HCAL+HGCAL @ HLT**

Backup

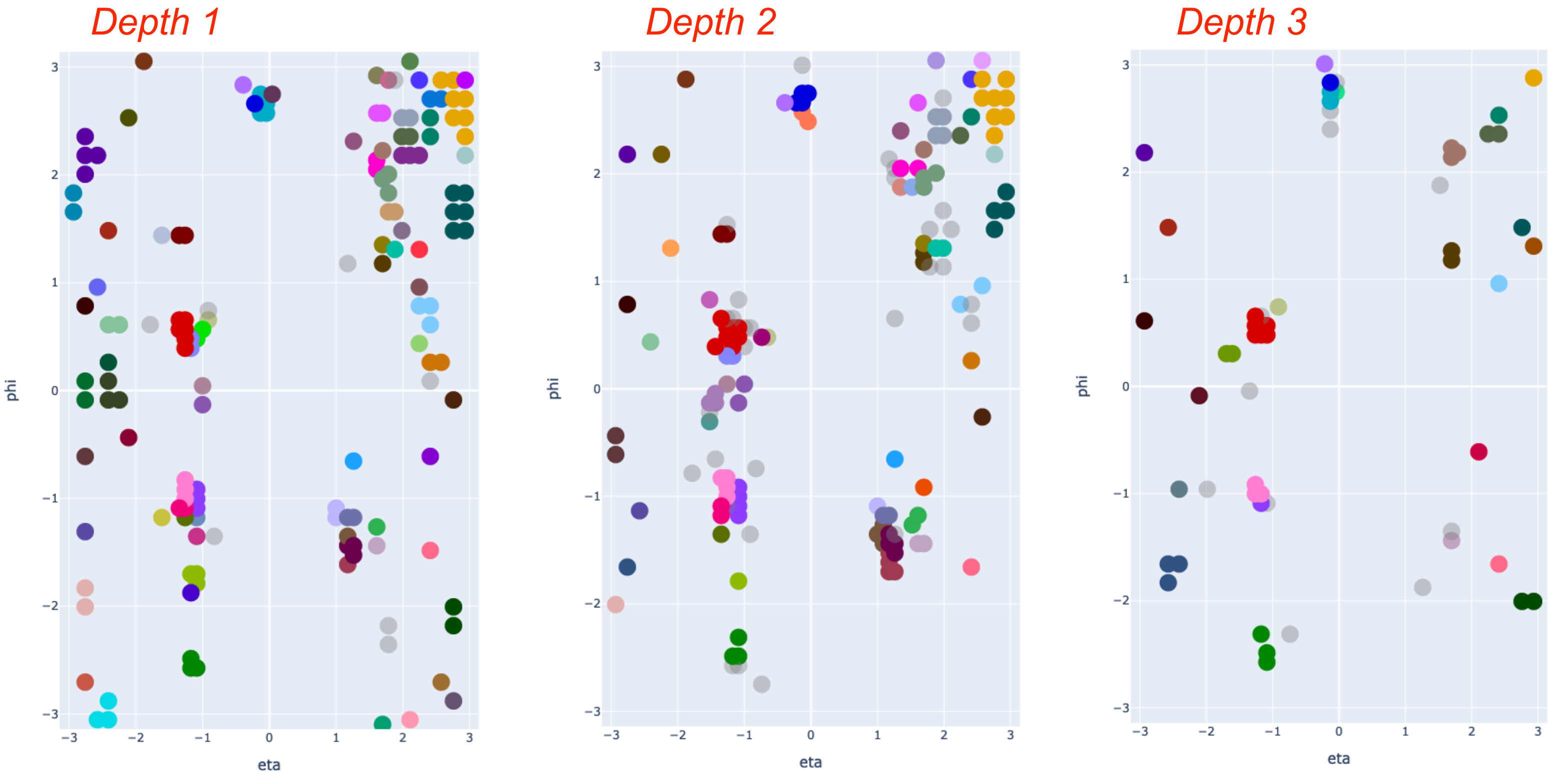
Integrating SPVCNN in CMSSW

- We integrated SPVCNN into CMSSW to test our workflow
    - We used SONIC + a GPU-enabled triton server
    - can also be run on local CPU resources
- This scheme largely **removes HCAL clustering time from offline**
    - Future goal: **HGCAL+HCAL implementation on HLT**

HCAL Results: event display



HCAL Results: event display

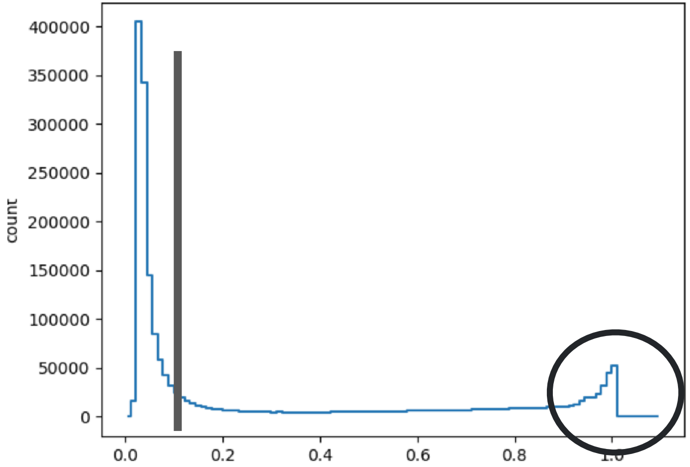


SPVCNN trained for PF targets creates contiguous, multi-depth clusters

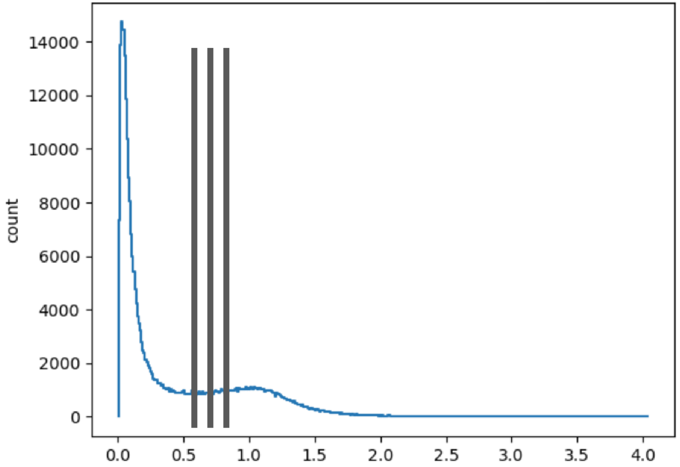
Hyperparameters

- From the training dataset, we choose t_β after the spike, $t_\beta = 0.1$
  - For too-small values, each hit will be considered its own cluster
- We choose t_d {0.6,0.7,0.8}

Object condensation score for all hits



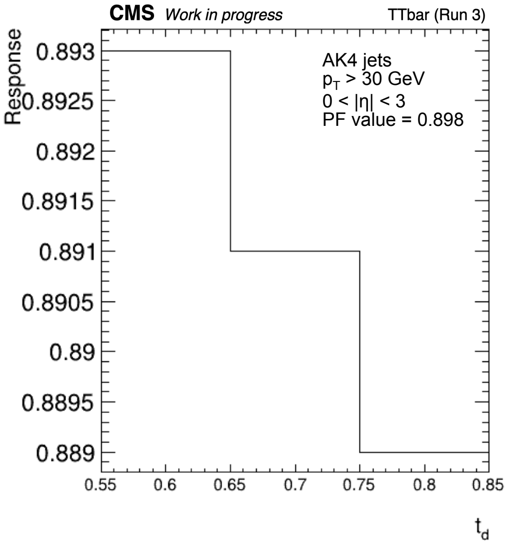
In truth clusters, most hits lie within distance ~.7 of the condensation point

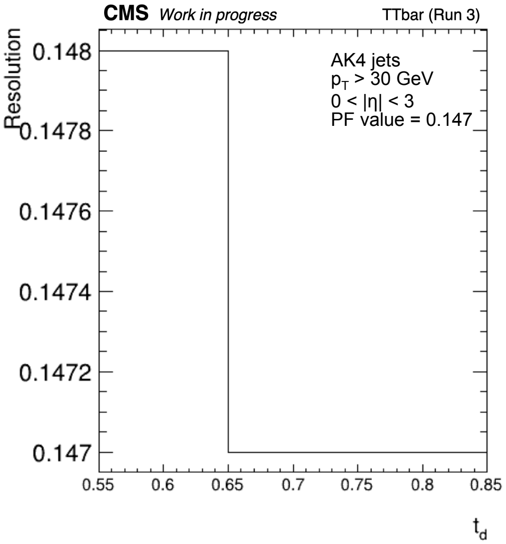


The the networks thinks these are good candidates values–we want to cluster around them!

Hyperparameters

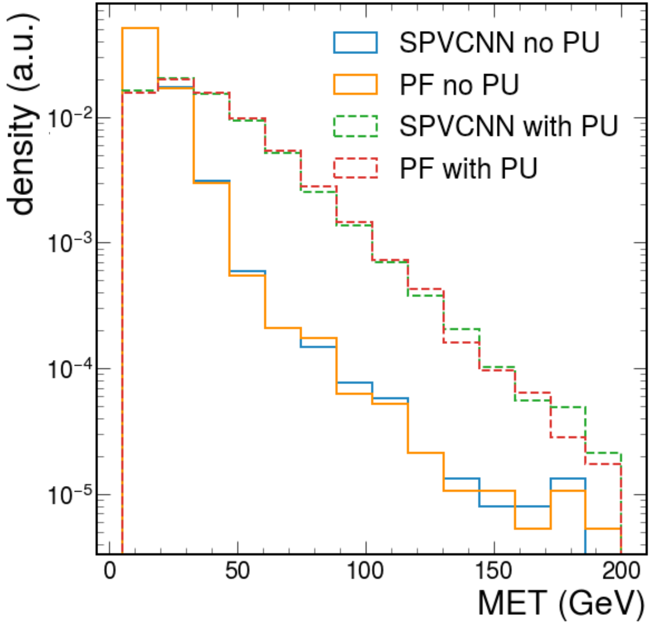
- We also choose based on high-level objects
- Jet energy scale and resolution vs. t_d for AK4 jets (TTbar run3)
  - Relatively insensitive to t_β
  - We choose t_d = 0.7 as a starting point



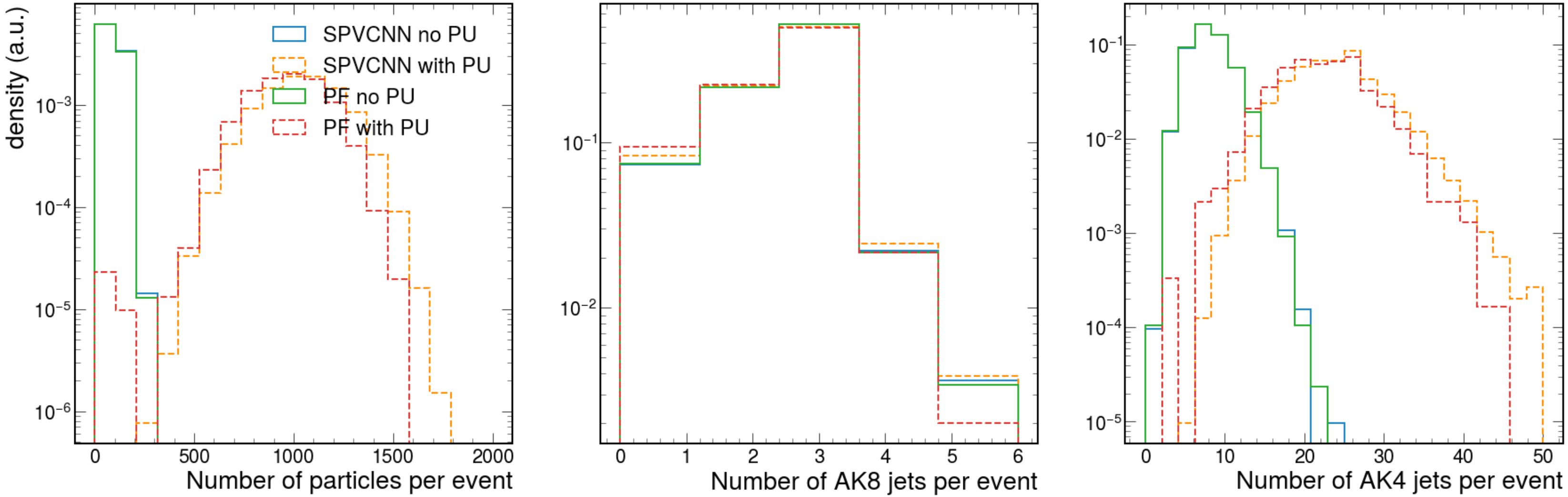


HCAL results: MET

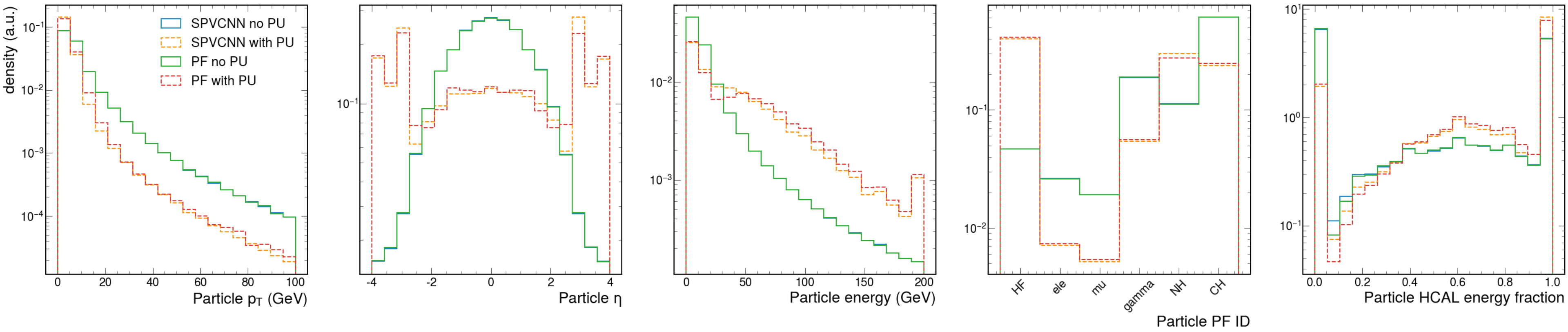
- How does SPVCNN clustering affect global observables like MET ?
  - generated Z(μμ)+jet in Run 3 with and without pileup to measure this



HCAL Results (with pileup)

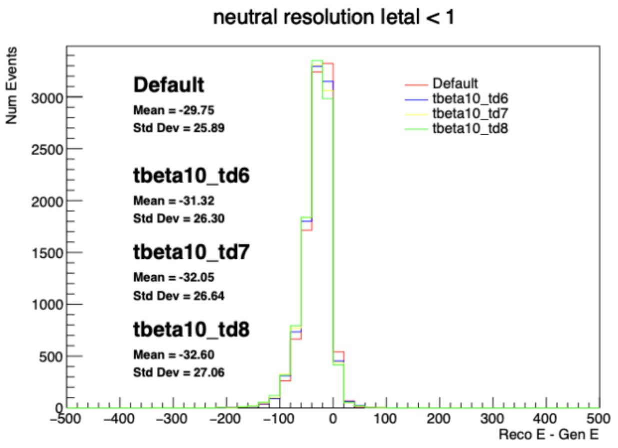


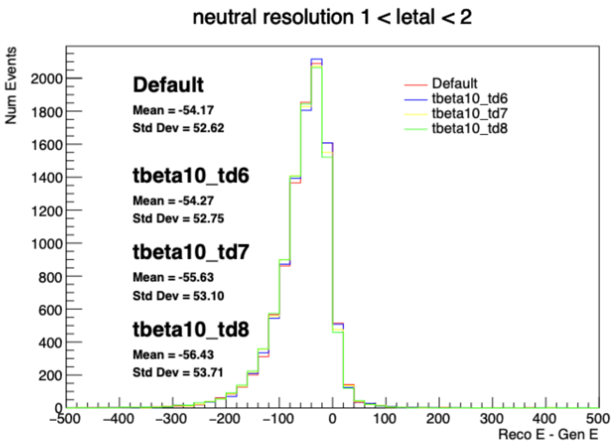
HCAL Results (with pileup)

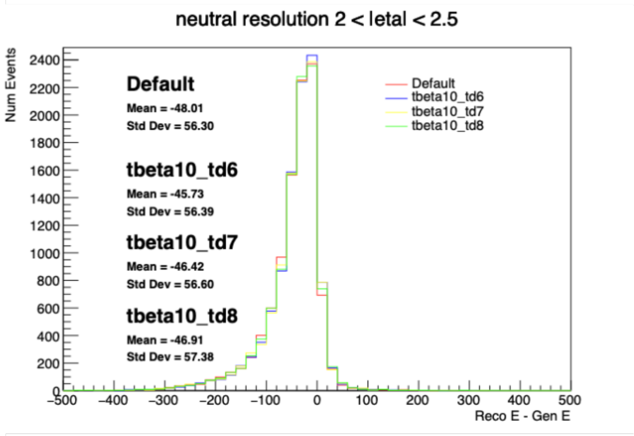


HCAL Results (zero pileup)

- Resolution of particles defined by Reco E - gen E
- Matches PF nicely

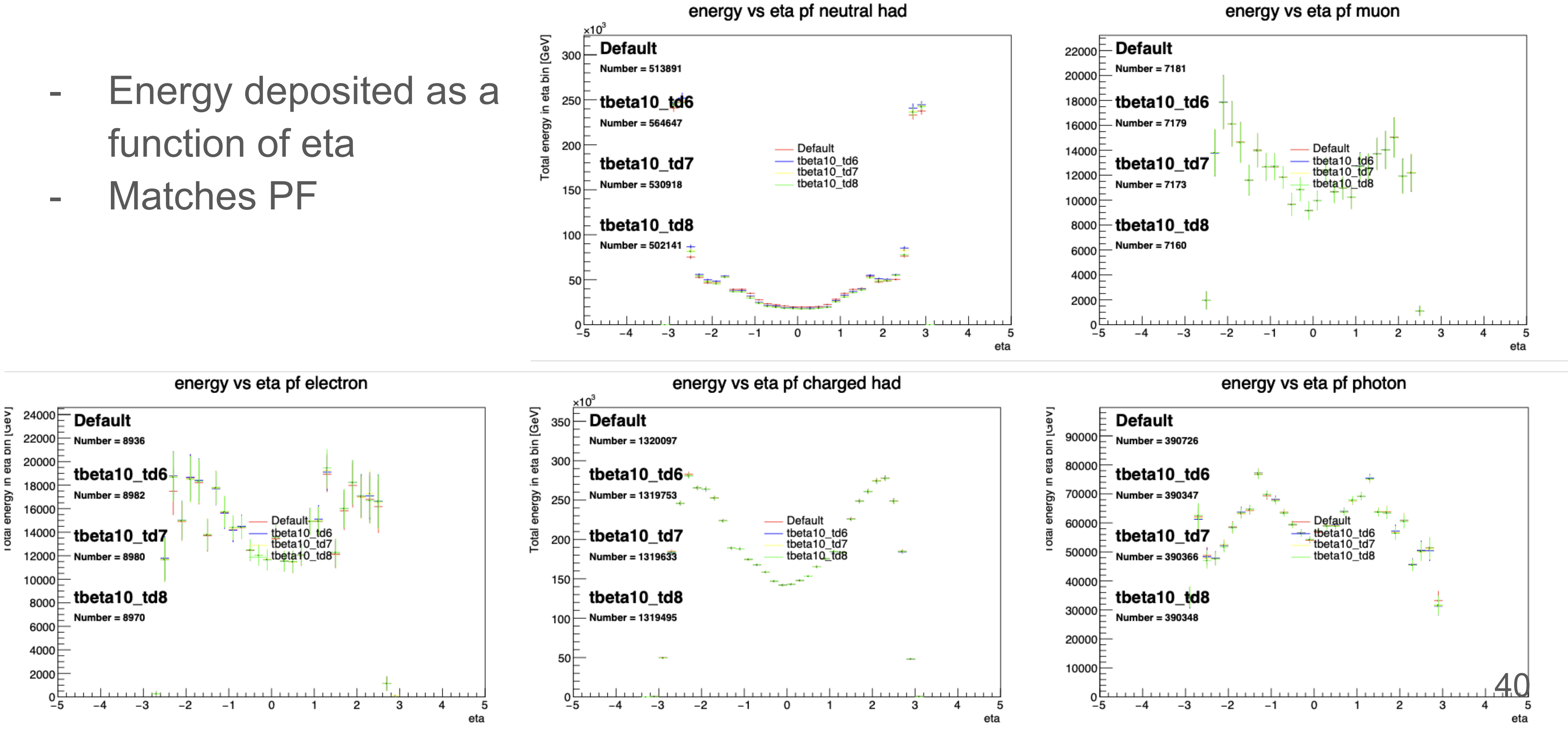






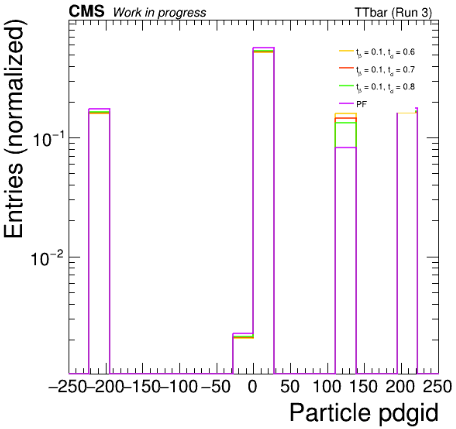
HCAL Results (zero pileup)

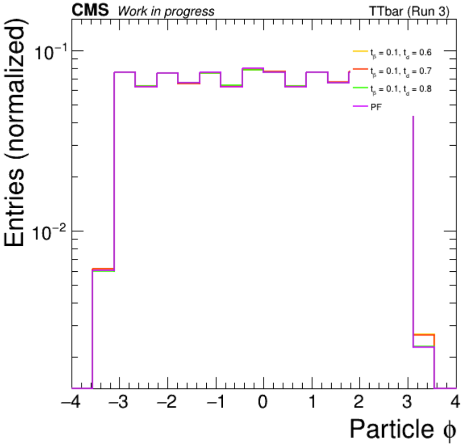
- Energy deposited as a function of eta
- Matches PF

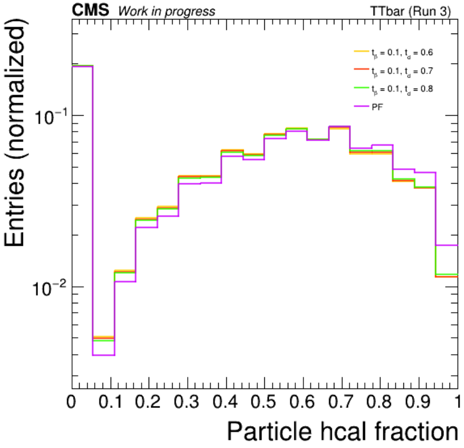


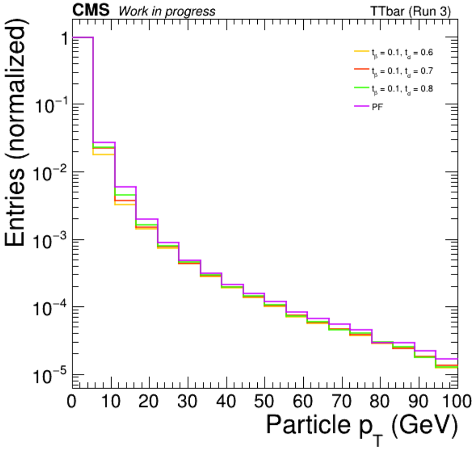
HCAL results (with pileup)

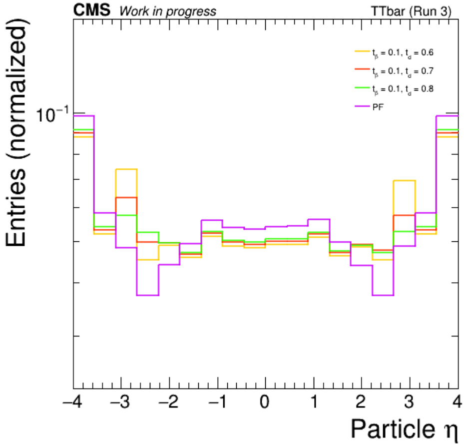
- Larger number of neutral hadrons
  - Larger difference for $\{t_d, t_\beta\}$ than in zero PU case





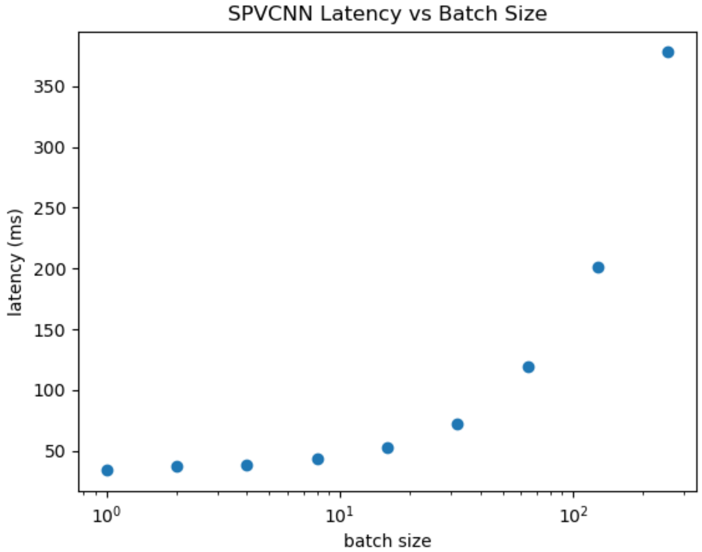


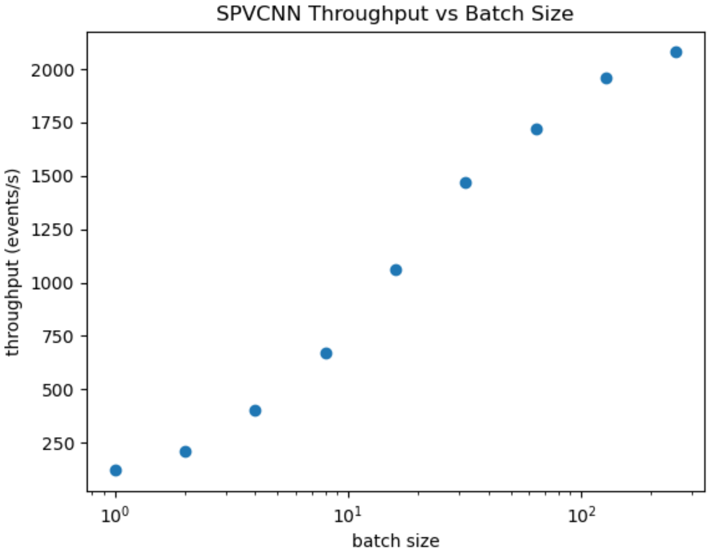




Latency checks

- Measured with 4xNVIDIA 3090-TIs
- Preliminary, unoptimized





Condensation loss

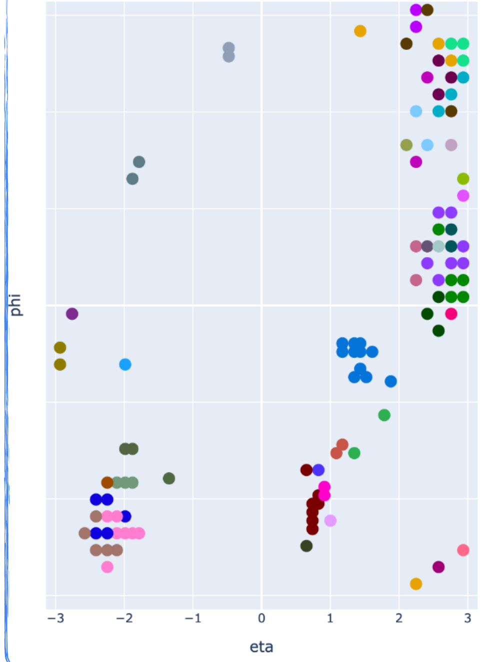
- Define a charge $q = \tanh^2(\beta)+q_{min}$ where $\beta \sim [0,1]$ is condensation score for each hit that is a parameter
- Loss is made of:
  - Repulsive term (push points and condensation points belonging to different objects apart from each other)
  - Attractive term (bring points and their condensation points together)
  - Beta term (break potential degeneracies from repulsive/attractive term, avoiding trivial solutions)
- To make clusters:
  - Order all hits by decreasing β values. Go down the list, clustering points within t_d of the condensation point (if the condensation point has been clustered, ignore it).
  - Once the β value reaches t_β, the clustering is complete.

HCAL: training target

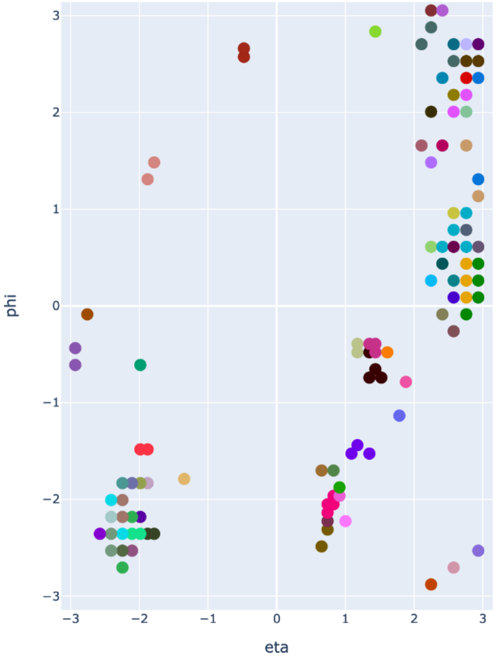
- Also interested in HCAL to include depth and timing information
- We train on Run 3 TTbar (with and without PU)
- Making truth definition for HCAL is a challenging task
    - Initially, we used a custom truth definition:
        - For each *RecHit*, find the *SimTrack* whose *SimHits* constituted the largest fraction of the total simulated energy in the *RecHit* HCAL cell.
            - cluster label = this *SimTrack* ID
        - Using this truth-level definition as a training target gives **improved jet response and resolution metrics** (relative to PF), **however we get discontiguous clusters that are not easy for SPVCNN to reproduce**
            - Leads to a large number of clusters and large number of particles (mostly neutral hadrons)

HCAL: event display

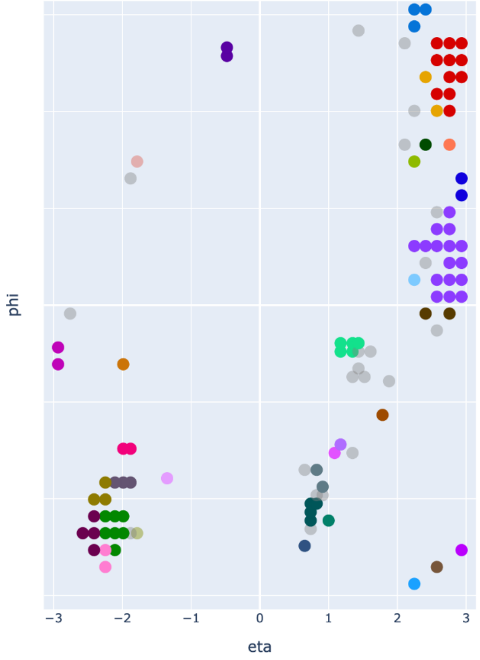
Original truth definition



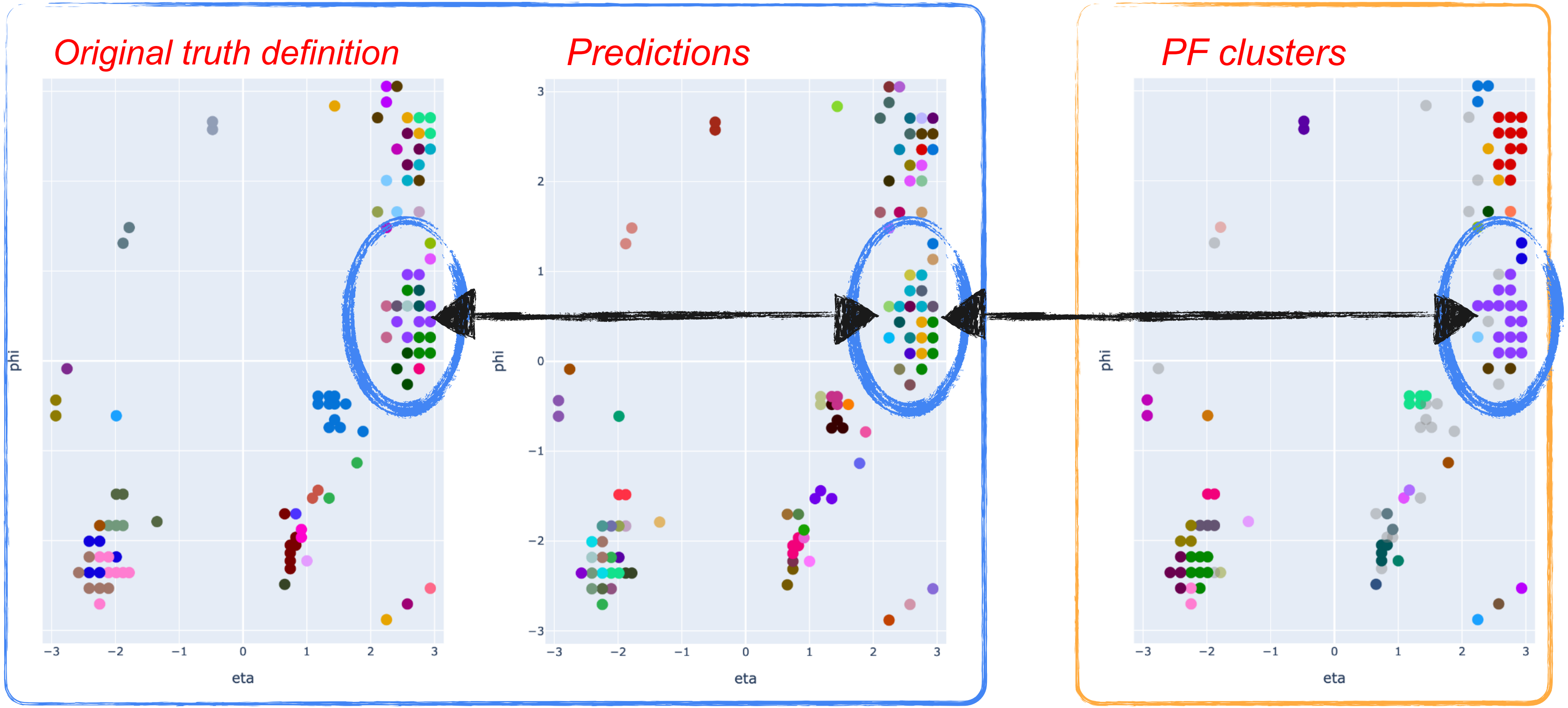
Predictions



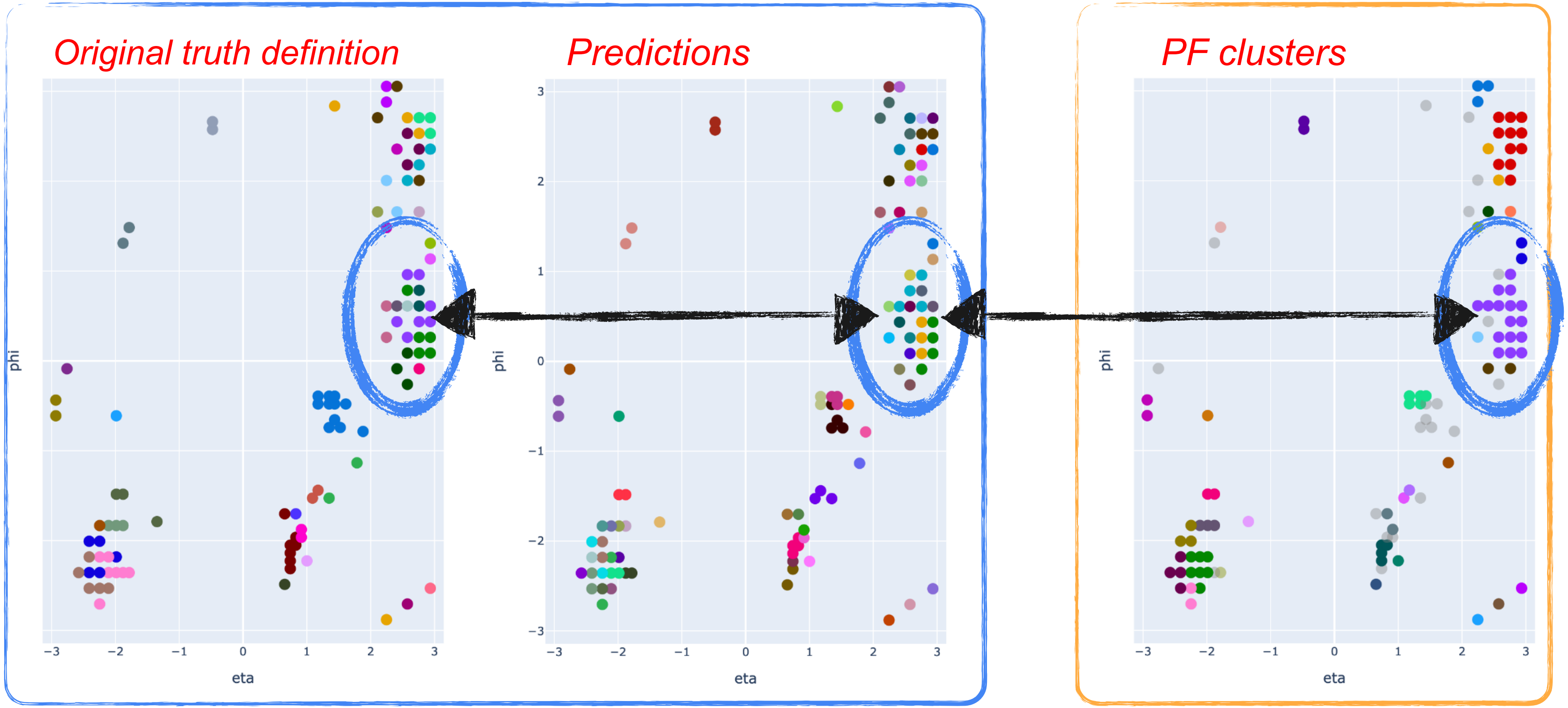
PF clusters



HCAL: event display



HCAL: event display



For now, we use PF HCAL cluster labels as the target

HCAL: training target

- This naive approach is not a good training target for SPVCNN
    - High-density regions with many co-linear particles → interleaved clusters with discontinuities → not reconstructable
- We are converging on a reasonable ground truth definition for HCAL

HGCAL Dataset

- We first use the same HGCAL dataset used for GravNet
    - Ditau in endcaps
    - zero pileup
    - O(20k) hits per event
    - Truth cluster definition: **same as GravNet**
        - energy deposits from reconstructed hits are traced to particles using Geant4 tracking
        - particles that cannot be reasonably distinguished due to detector granularity are merged together