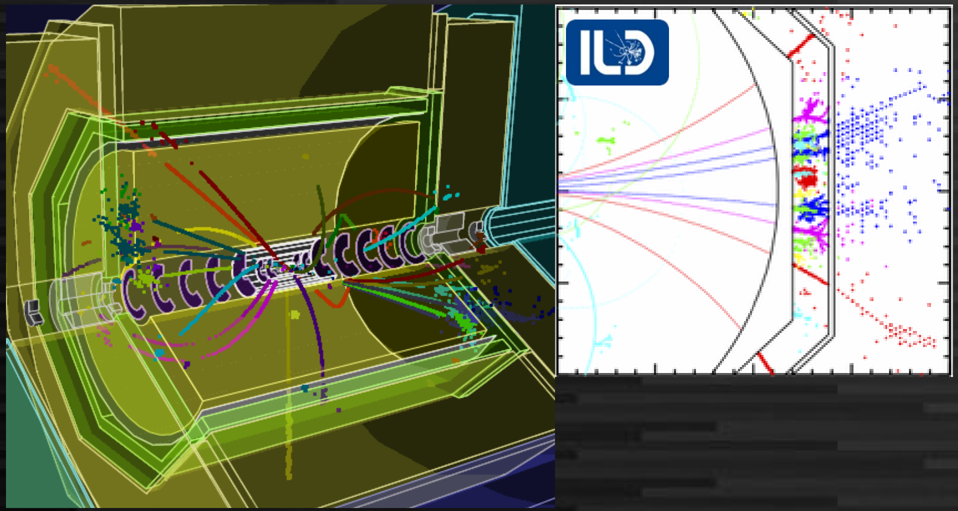
Development of particle flow algorithm with GNN for Higgs factories

Taikan Suehara / 末原　大幹
(ICEPP, The University of Tokyo)

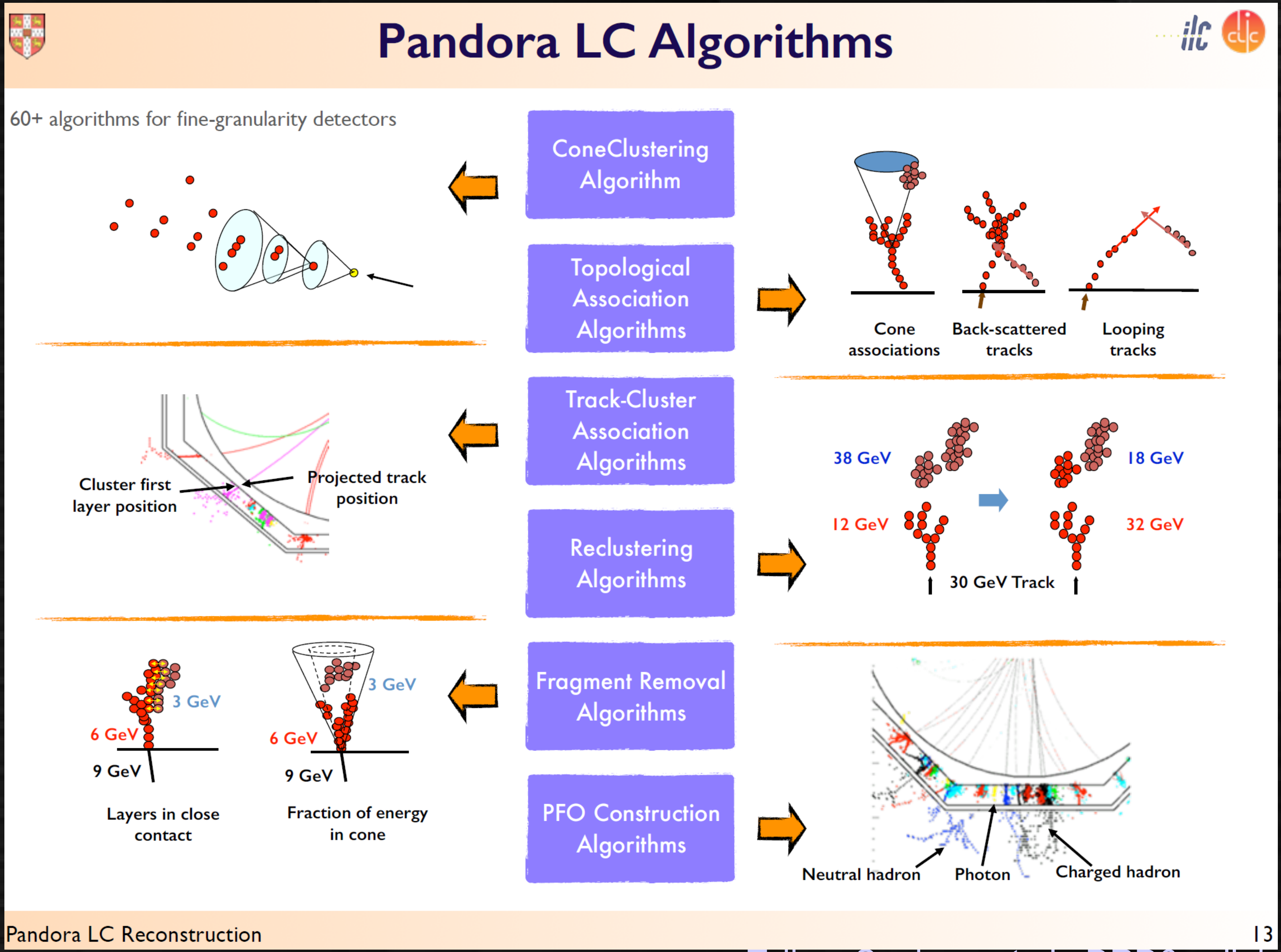
Collaborators: T. Murata (U. Tokyo), T. Tanabe (MI-6 Co.), L. Gray (Fermilab), P. Wahlen (IP Paris & ETHZ / internship at Tokyo)

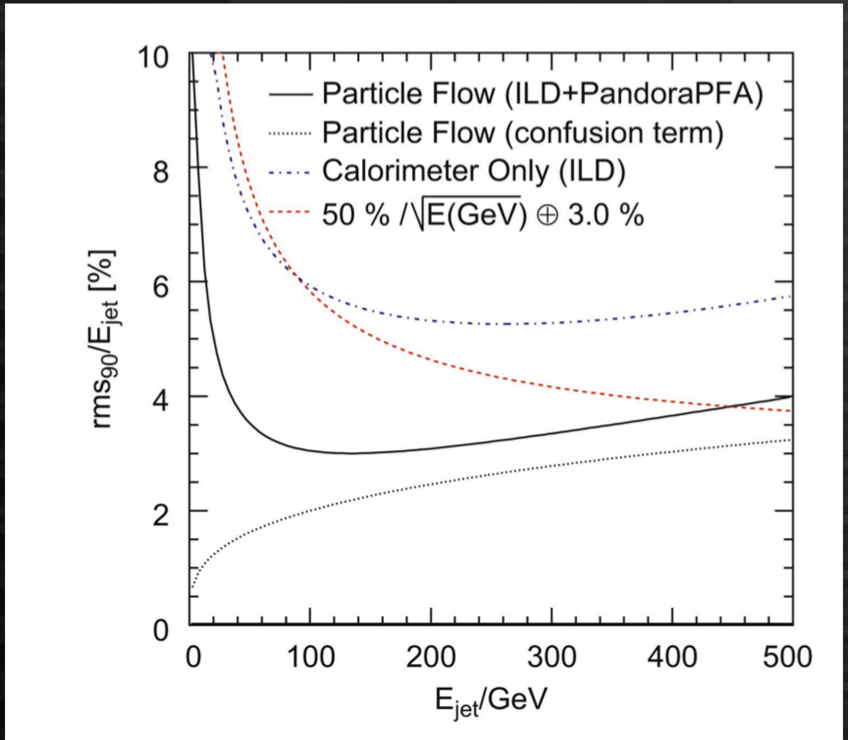
Particle flow for Higgs factories



- High granular calorimetry
  - 3D pixels for imaging EM/hadron showers at calorimeters
    - eg. 10^8 channels for ILD ECAL
  - Separation of particles inside jets
    → ~2x better energy resolution by separation of contribution from charged particles
    - Software algorithm essential (as well as hardware design)
- Particle Flow algorithm
  - Essential algorithm for high granular calorimetry
  - Complicated pattern recognition → good for DNN

Pandora ParticleFlow algorithm





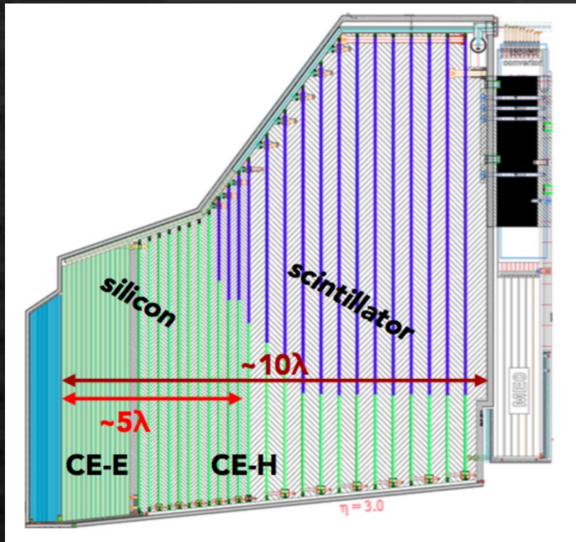
Widely used since 2008
Reasonably good performance up to ~50 GeV jets
Confusion dominates at higher energies

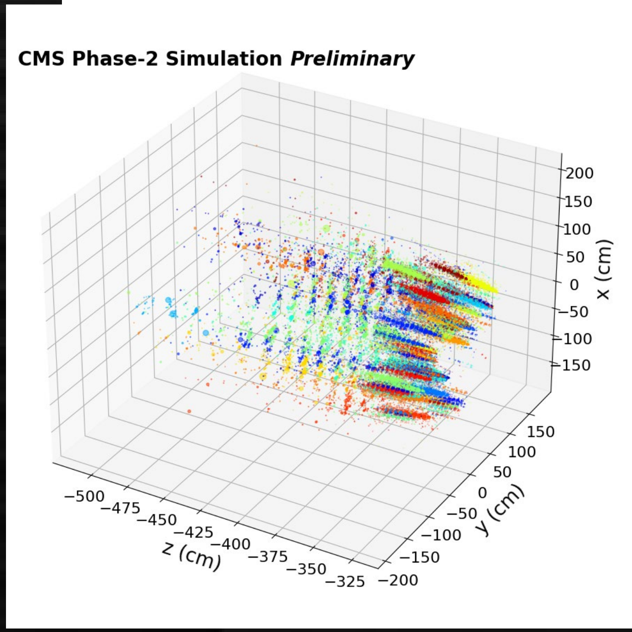
Motivations for DNN particle flow

- Performance improvement
  - Confusion dominant at jet energy > 100 GeV
  - More efficient way to separate cluster from charged particles should be investigated
- Integrate other functions
  - Software compensation, particle ID etc. closely related to PFA
- Detector optimization
  - Comparison with different detector settings
    - PandoraPFA too much depends on internal parameters
  - Effect of timing information to be investigated
    - With different timing resolution (1 ns, 100 ps, 10 ps, …)

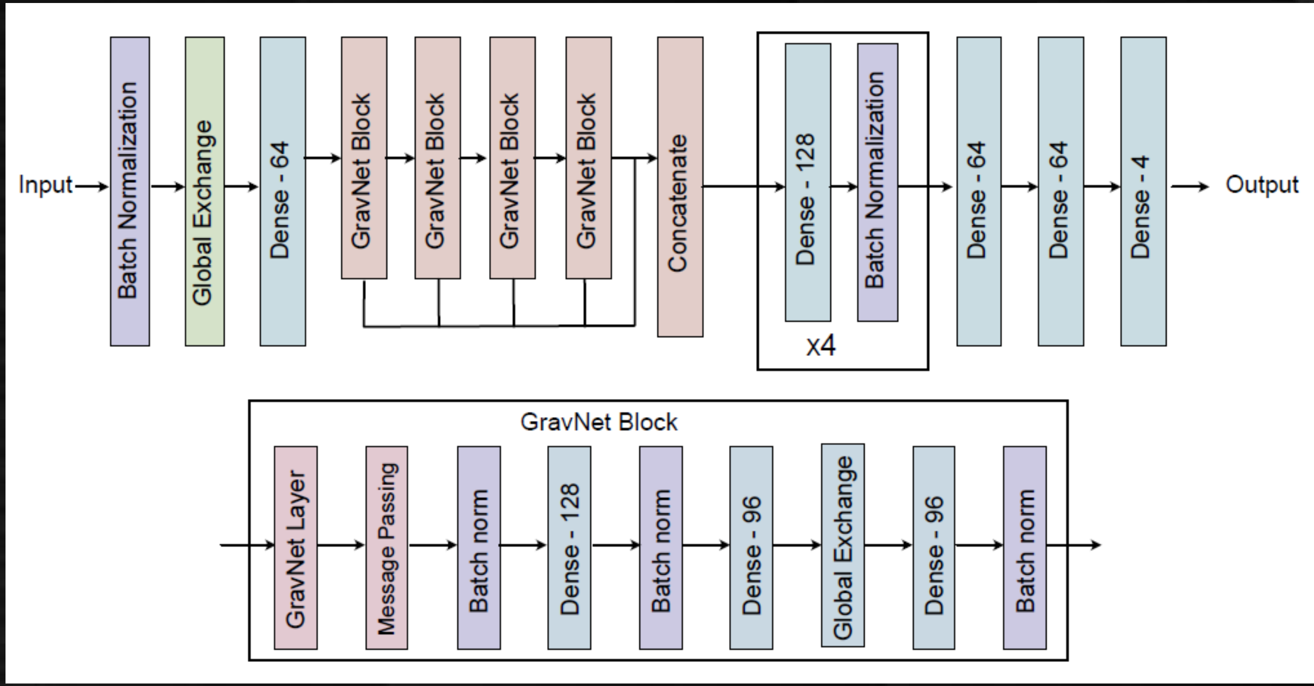
GravNet for CMS HGCAL

- CMS HGCAL
  - High granular forward calorimeter for HL-LHC upgrade at CMS
  - Similar to ILD calorimeter (silicon pixel + scintillator)
    - Inspired by CALICE development
- Reconstruction at HGCAL
  - Pileup/noise to be separated by software
  - Numerous particles from ~200 pileups
    - Difficult to handle: software algorithm critical
    - DNN reconstruction being investigated
      - Reasonable performance obtained up to ~50 pileups?





The network



Rather complicated network
with ~30 hidden layers

“Object condensation” loss function
is applied (shown in next page)

Input/output obtained for each hit at calorimeter

Input: Features at each hit (position, energy deposit, timing)

Output: “condensation coefficient” β, position at virtual coordinate (2-dim)
optional output of features such as energy, PID (not used now)

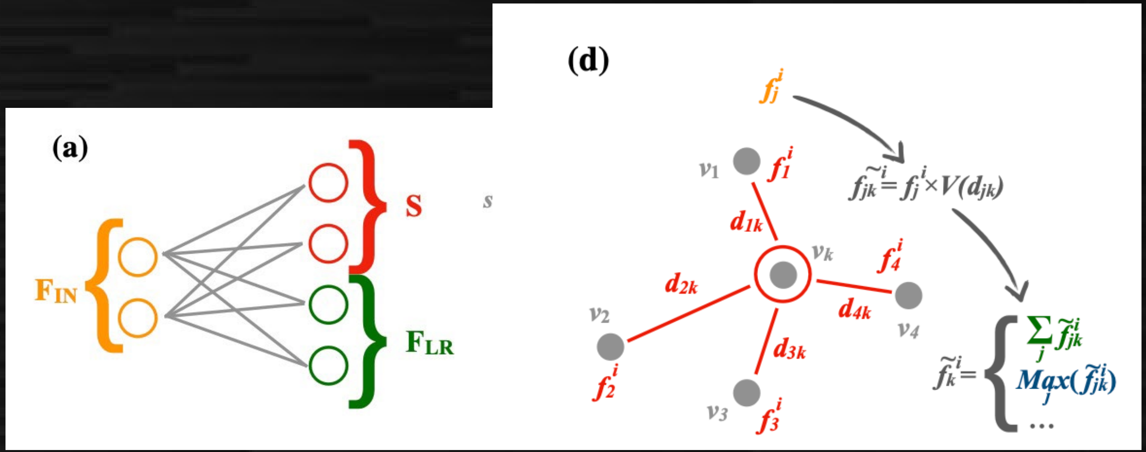
Dense (fully-connected layer) inside each hit, GravNet connects hits

GravNet and Object Condensation

GravNet

arXiv:1902.07987

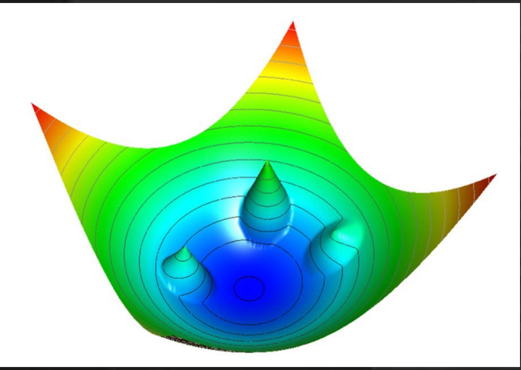
- The virtual coordinate (S) is derived from input variables with simple MLP
- Convolution using "distance" at S (bigger convolution with nearer hits)
- Repeat 2 times and concatenate the output with simple MLP



Object Condensation (loss function)

arXiv:2002.03605

$$L = L_p + s_C(L_\beta + L_V)$$



- Condensation point: The hit with largest β at each (MC) cluster
- L_V: Attractive potential to the condensation point of the same cluster and repulsive potential to the condensation point of different clusters
- L_β: Pulling up β of the condensation point
- L_p: Regression to output features (energy etc.) → currently not used

What we implemented: track-cluster matching

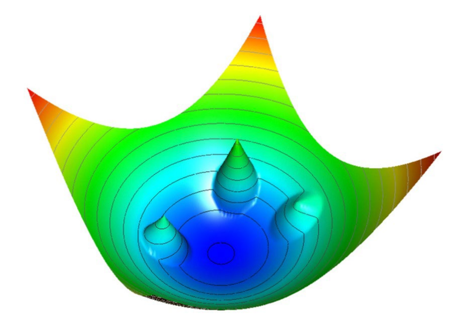
- PFA is essentially a problem “to subtract hits from tracks”
- HGCAL algorithm does not utilize track information
  - Only calorimeter clustering exists
- Putting tracks as “virtual hits”
  - Located at entry point of calorimeter
  - Having “track” flag (1=track, 0=hit)
  - Energy deposit = 0
- Modification on object condensation to forcibly treat tracks as condensation points (details next page)
  - Also modifying clustering algorithm to avoid double-track clusters

Current number of parameters: ~420K

Object condensation and our implementation

Object condensation loss function (the function to minimize)

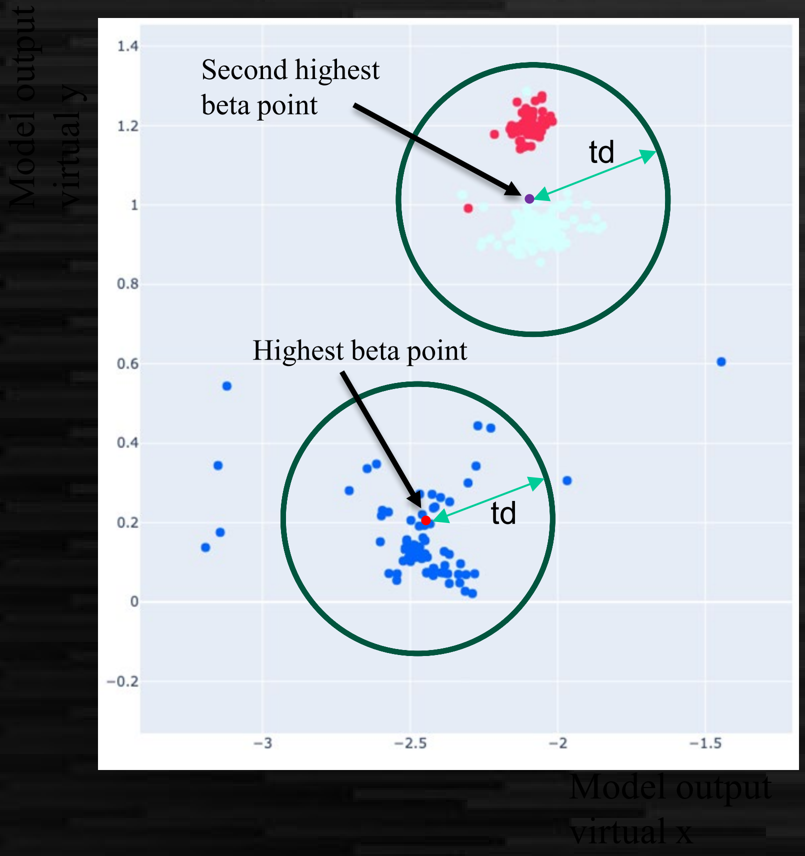
$$L = L_p + s_C(L_\beta + L_V)$$



- Condensation point: The hit with largest β at each (MC) cluster
  → For each MC cluster having a track,
  the track is forcibly the condensation point regardless of β
- L_V: Attractive potential to the condensation point of the same cluster and repulsive potential to the condensation point of different clusters (no modification)
- L_β: Pulling up β of the condensation point (up to 1)
  (no modification, but β of tracks become spontaneously close to 1)
- L_p: Regression to output features (energy etc.) → currently not used

Clustering algorithm

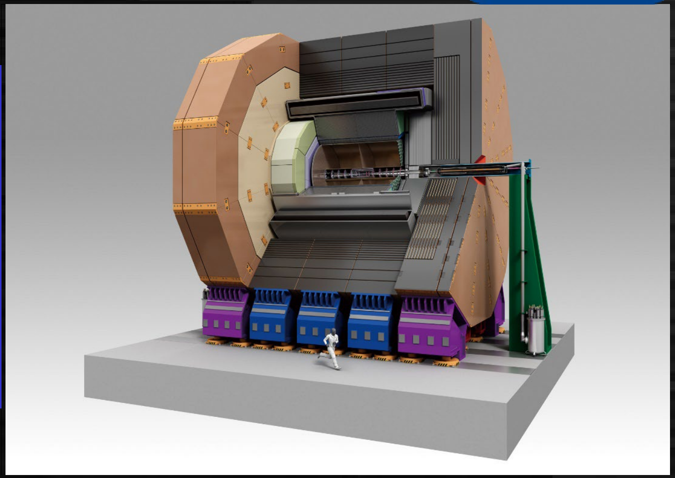
- Output of the network is position and β of each hit → need clustering
- Hits that are within a certain distance (td) from the highest β point assume as a cluster
- Continues clustering until all hits are clustered or β of remaining hits are below threshold (tbeta)
- td/tbeta are tunable parameters



Our samples for performance evaluation

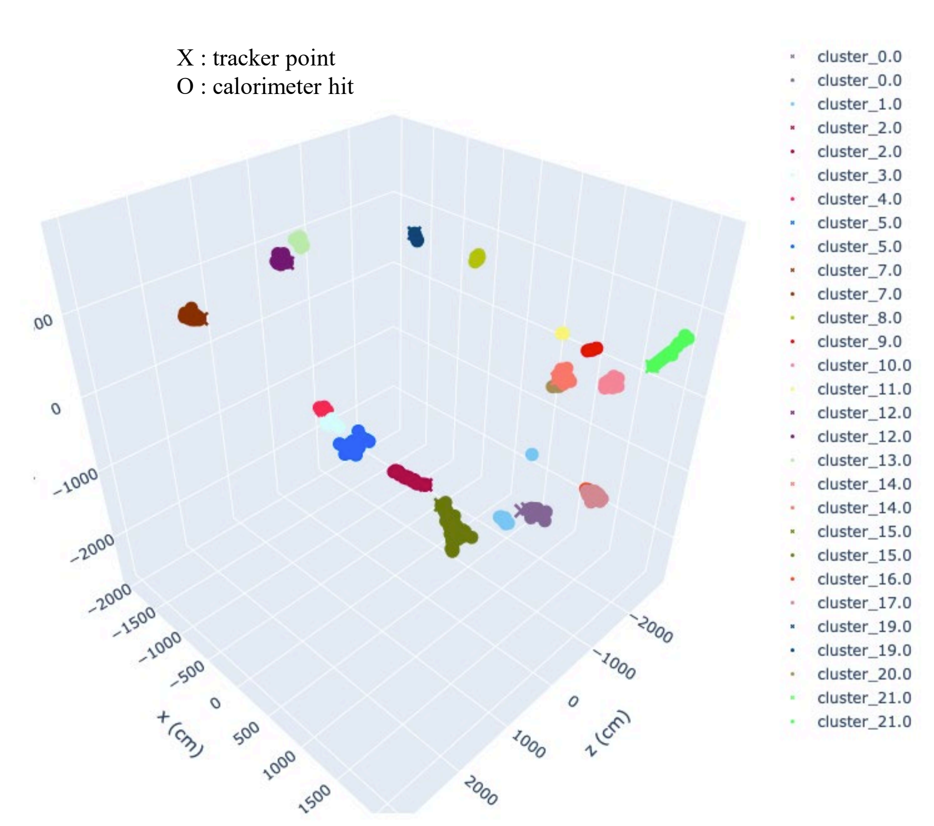
- ILD full simulation with SiW-ECAL and AHCAL
  - ECAL: 5 x 5 mm^2, 30 layers, Tungsten/silicon sandwich (24 X_0)
  - HCAL: 30 x 30 mm^2, 48 layers, Iron/scintillator sandwich (6 λ)
  - 10 Taus overlayed with random direction
    - 100k events, 10 GeV x 10 taus / event → 1 million taus (~13 GB)
    - 1M events with variable energies up to 100 GeV to be tested (~500 GB)
  - qq (q=u, d, s) sample at 91 GeV
    - ~75k events
    - Official sample for PFA calibration
    - A few 10 GB each

Taus: good mixture of hadrons, leptons and photons with some isolation
Good for training



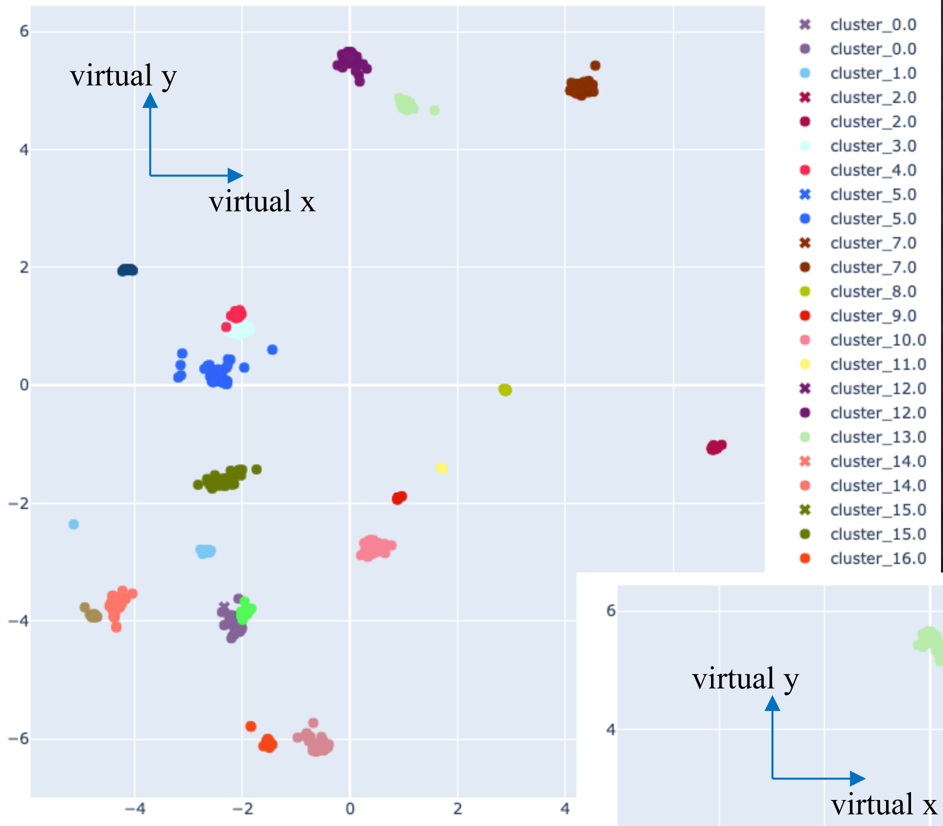
Event display

10 Taus
@ 10 GeV each



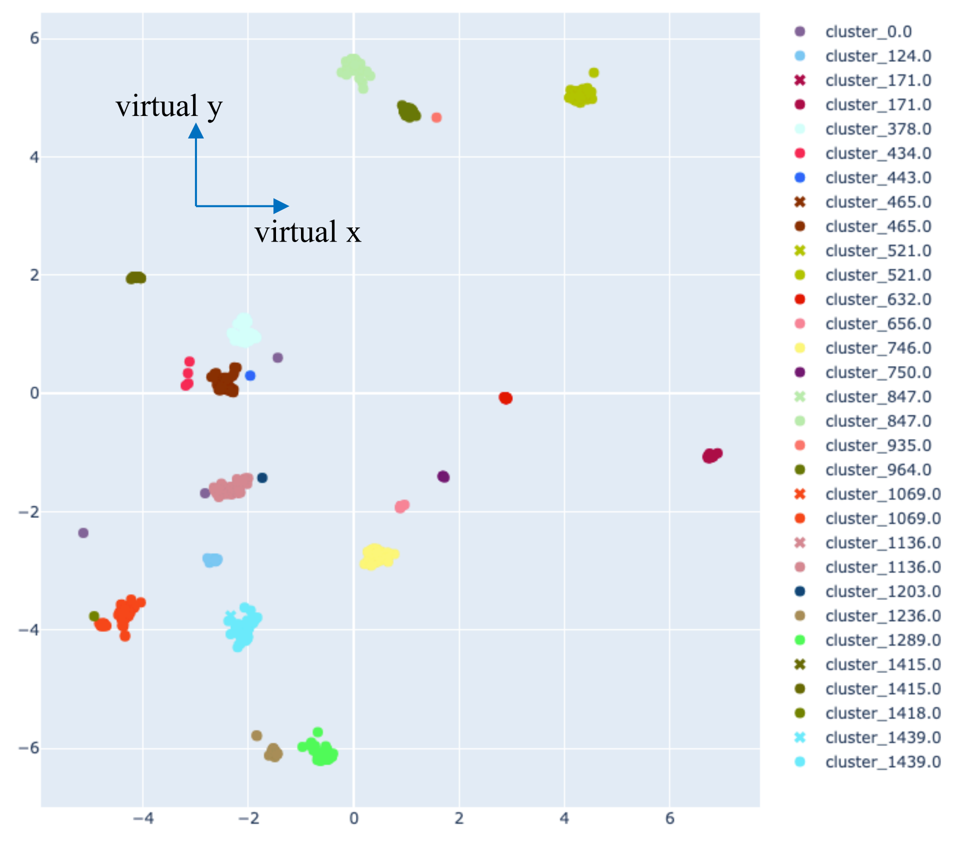
Input features
Real coordinate in detector

Colored by true clusters



Colored by
true clusters

Output features
Virtual coordinate

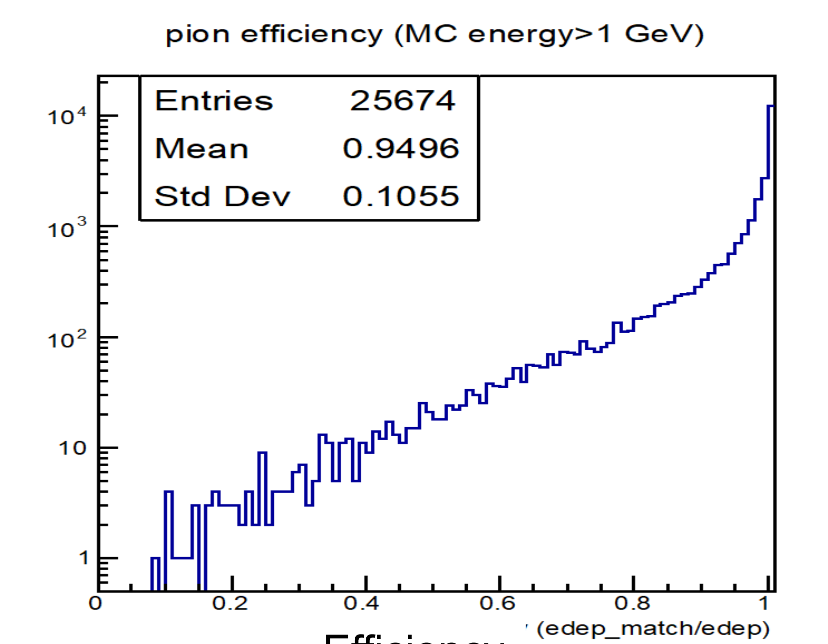


Colored by
reconstructed clusters

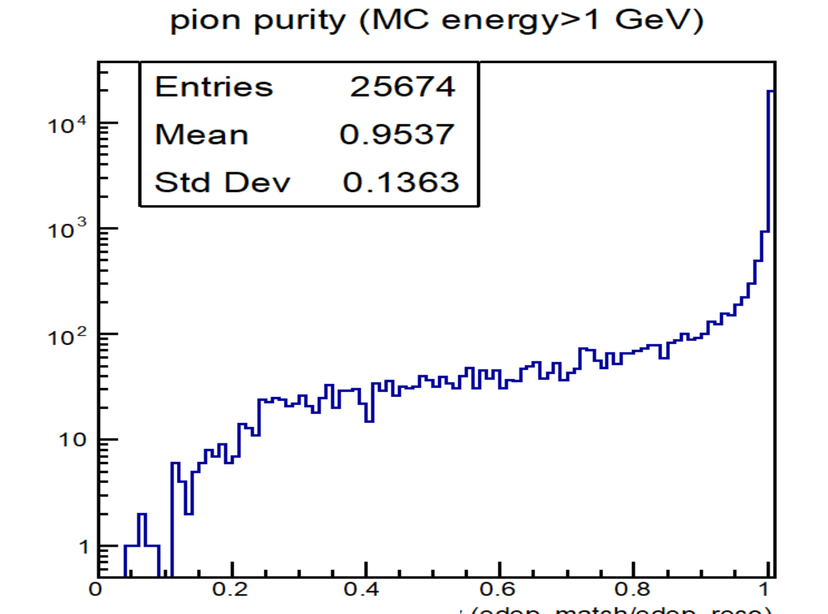
Taikan Suehara et al., DRD6 collabo

Quantitative evaluation

- Make 1-by-1 connection of MC and reconstructed cluster
  - Reconstructed cluster with highest fraction of hits from the MC is taken
  - Multiple reconstructed cluster may connect to one MC cluster
- Quantitative comparison with PandoraPFA
  - Compared "efficiency" and "purity" of particle flow
    - Efficiency : (reconstructed cluster energy that matches the MC cluster) / (MC cluster energy)
    - Purity : (reconstructed cluster energy that matches the MC cluster) / (reconstructed cluster energy)

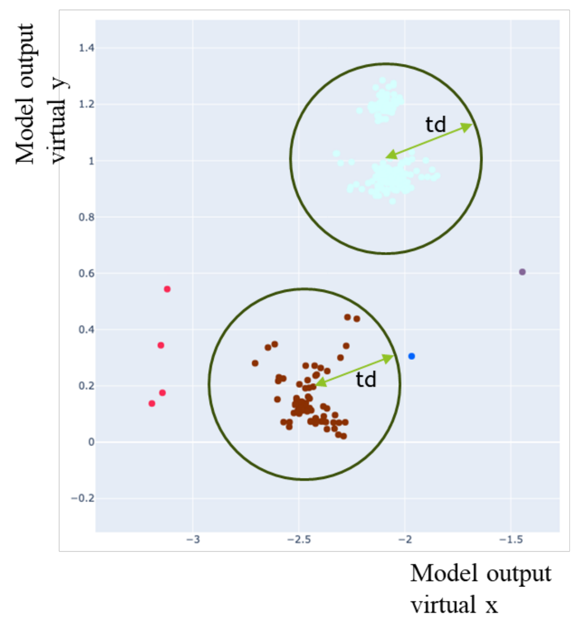


Efficiency



Purity

Optimization of performance

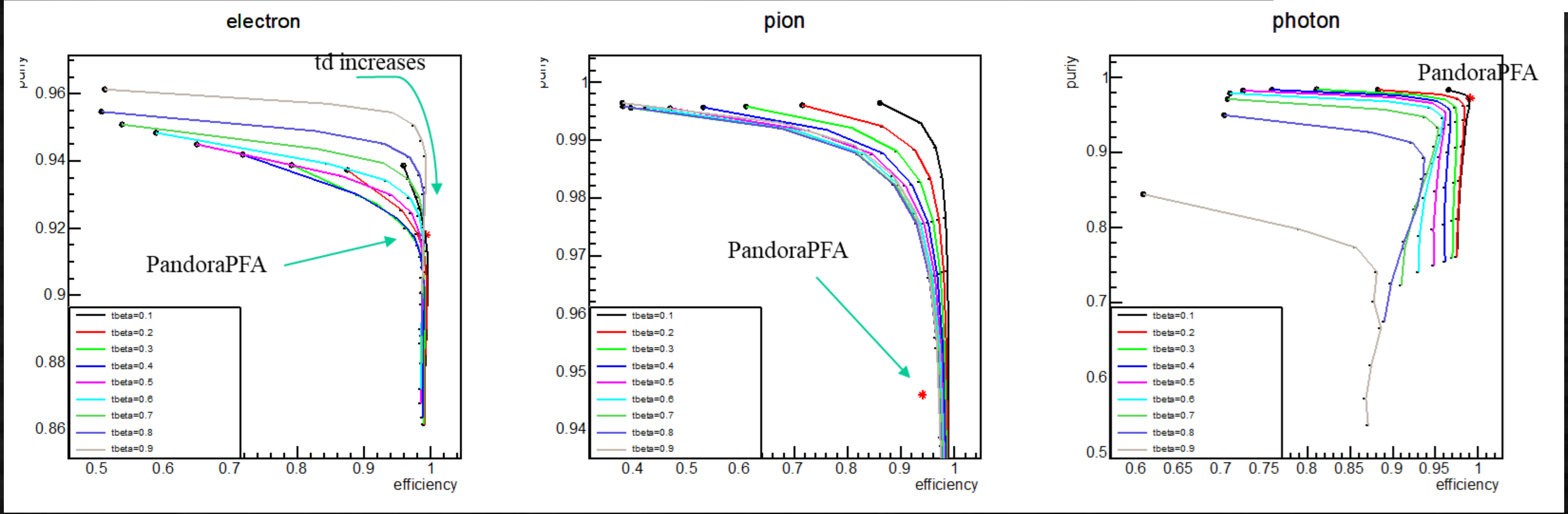


Output dimension of the coordinate

- The initial work done with output coordinate dimension D = 2 (for visibility)
- Tried D=3,4,8,16 → D=4 selected

Clustering parameters (td, tbeta)

- td: radius which hits are treated as coming from the same cluster
- tbeta: threshold of beta to form clusters



Scanning result: tbeta=0.1, td=0.3/0.4 selected

Results on efficiency and purity

Algorithm train/test	Electron eff.	Pion eff.	Photon eff.	Electron pur.	Pion pur.	Photon pur.
GravNet 10 taus/10 taus	99.1%	96.5%	99.0%	91.8%	98.9%	97.1%
PandoraPFA 10 taus	99.3%	94.0%	99.1%	91.8%	94.6%	97.2%
GravNet jets/jets	94.5%	93.1%	95.2%	77.4%	93.2%	92.4%
PandoraPFA jets	80.2%	90.4%	79.0%	75.0%	90.6%	77.7%
PandoraPFA jets (ILCSoft truth)	96.7%	95.5%	96.4%	97.1%	90.4%	97.7%

At least in our measure, performance of GravNet-based algorithm exceeds PandoraPFA

→ Promising as full PFA (but energy regression to be done)

Definition of MC truth clusters needs to be tuned (see ILCSoft truth)

Energy regression: in progress

Add “energy” to the output of the network (for each hit)
Add a term to object condensation

E_i : true cluster energy
ε_i : predicted cluster energy
β_i : condensation factor

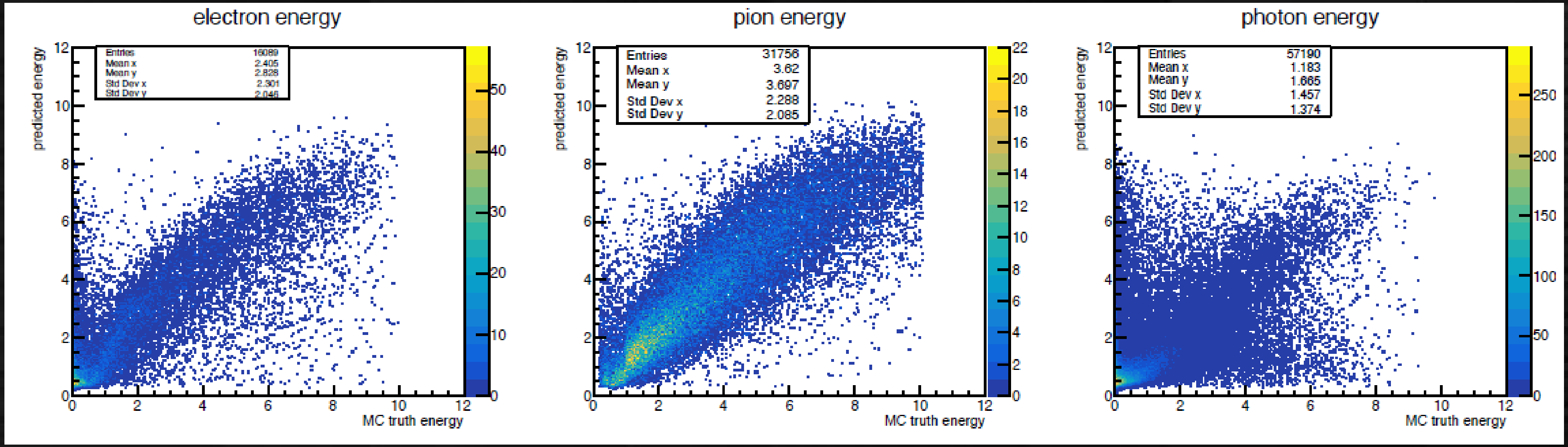
④ $L_E = \sum_i \theta_i (E_i - \varepsilon_i)^2$ $\theta_i = 1$ if the point is condensation point

⑤ $L_E = \frac{\sum_i (\beta_i E_i - \varepsilon_i)^2}{\sum_i \beta_i^2}$ summation of all hits

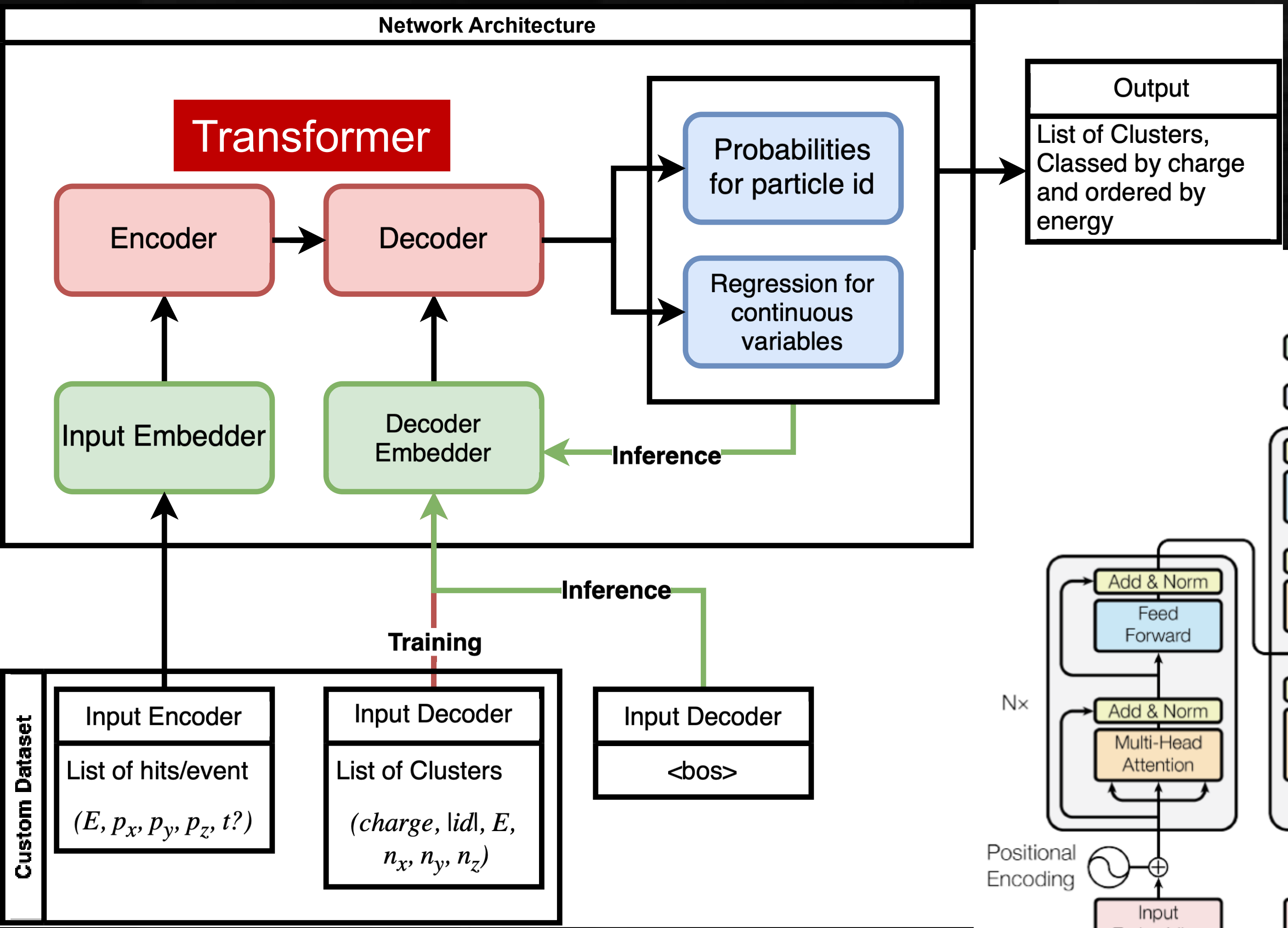
ε_i : energy related variable

Reasonable correlation to MC energy seen
Performance still to be tuned

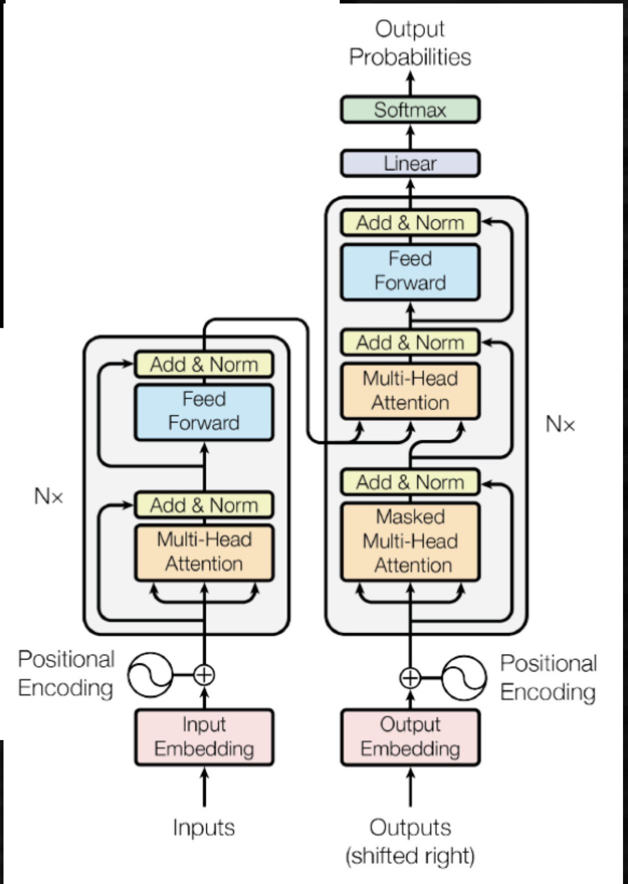
Cluster energy (MC vs reco) at 10 taus event with LE no. 4, without track momenta



More NLP-like model: transformer



Planned structure for PFA



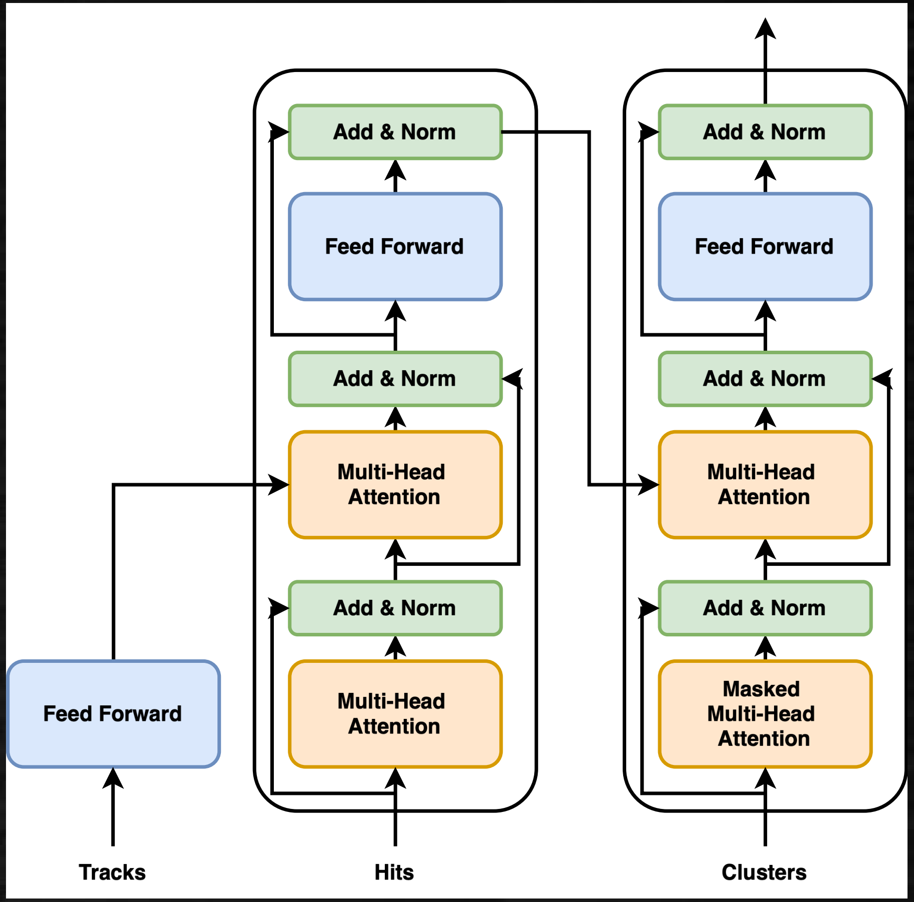
Transformer

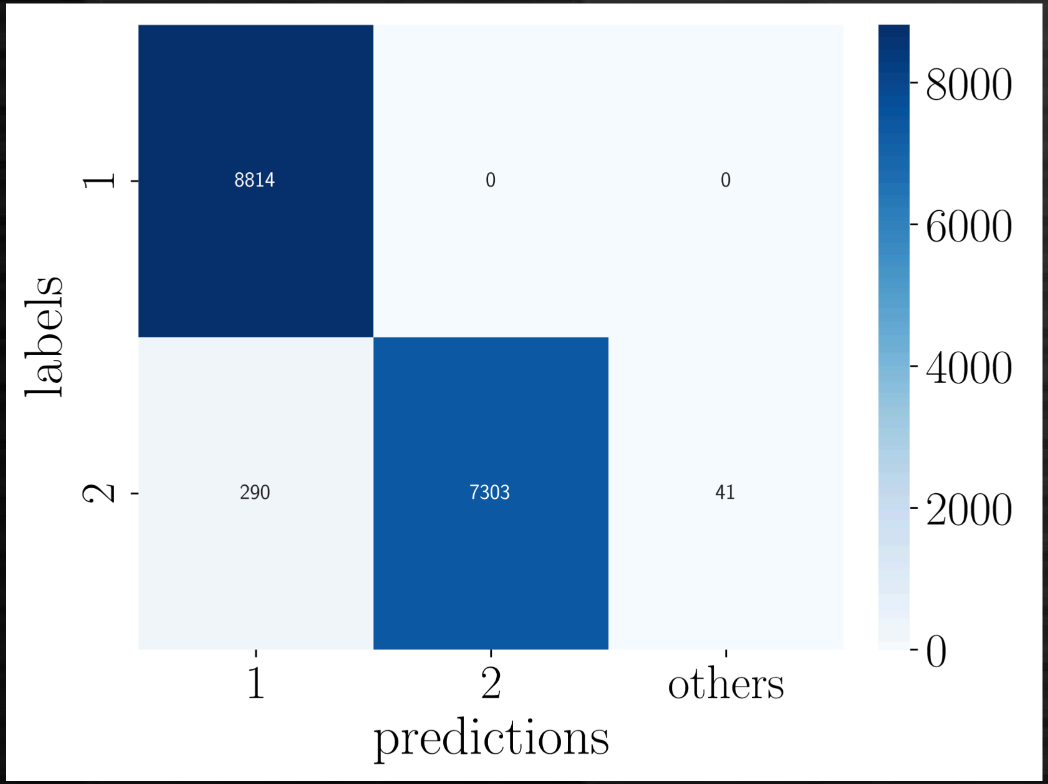
Transformer: training relation among elements (hits in PFA) with (multi-head) self-attention mechanism (used in GPT etc.)

Encoder: accumulate info of all hits/tracks by transformer

Decoder:
Input cluster info one by one
Output info of next cluster
(training) MC truth clusters
(inference) just provide <bos> to derive first cluster, using output as next input until <eos> obtained
(Inspired by translation NN)

Transformer-based PFA: some quick view





Separation of single and double photons - random opening angle – not too bad but worse than GNN-based study now

Proposal from collaborator: should investigate independent training of encoder part by e.g. masking some particles in each event (as often done in NLP)

Summary and plans

First target achieved!

- GNN-based particle flow has possibility to replace PandoraPFA
  - Performance seems significantly exceeded at least in our measure
  - Difference on MC truth definition to ILCSoft to be investigated
    - (ILCSoft uses MCParticlesSkimmed while our method uses MCParticle collection)
- Regression of cluster energy being investigated
  - Necessary for complete PFA
  - Jet energy resolution would be compared with PandoraPFA
- Possible improvements
  - Momenta of tracks currently not used (improvements of clustering possible)
  - Incorporation of timing information etc.
- Another new idea to “ask network the next cluster” being tried
- Implementation to analysis: maybe not in the ECFA timescale…