Jet flavor tagging and particle flow by DNN with ILD full simulation

arXiv:2410.08772 (Proc. ICHEP2024)
with some recent updates

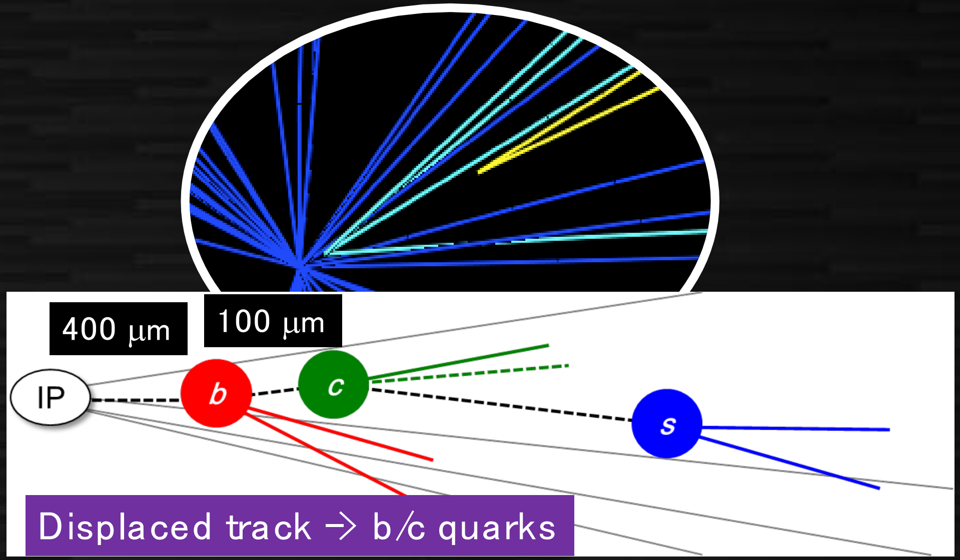
Taikan Suehara / 末原　大幹
(ICEPP, The University of Tokyo)

R. Tagami, T. Murata, W. Ootani, M. Ishino (ICEPP, UTokyo),
P. Wahlen (ETH/IP Paris/ILANCE UTokyo),
L. Gui (Imperial/Kyushu U.), T. Tanabe (MI-6 Ltd.)

Today's topics

All works done with ILD full simulation (plus FCCee Delphes for comparison)

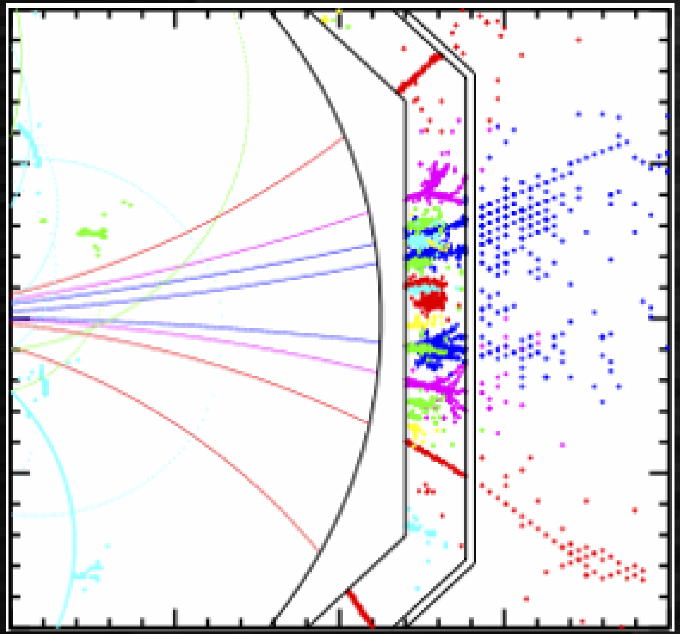
Flavor tagging with Particle Transformer (ParT)



Big impact on Higgs studies including self coupling

Strange tagging is also a scope

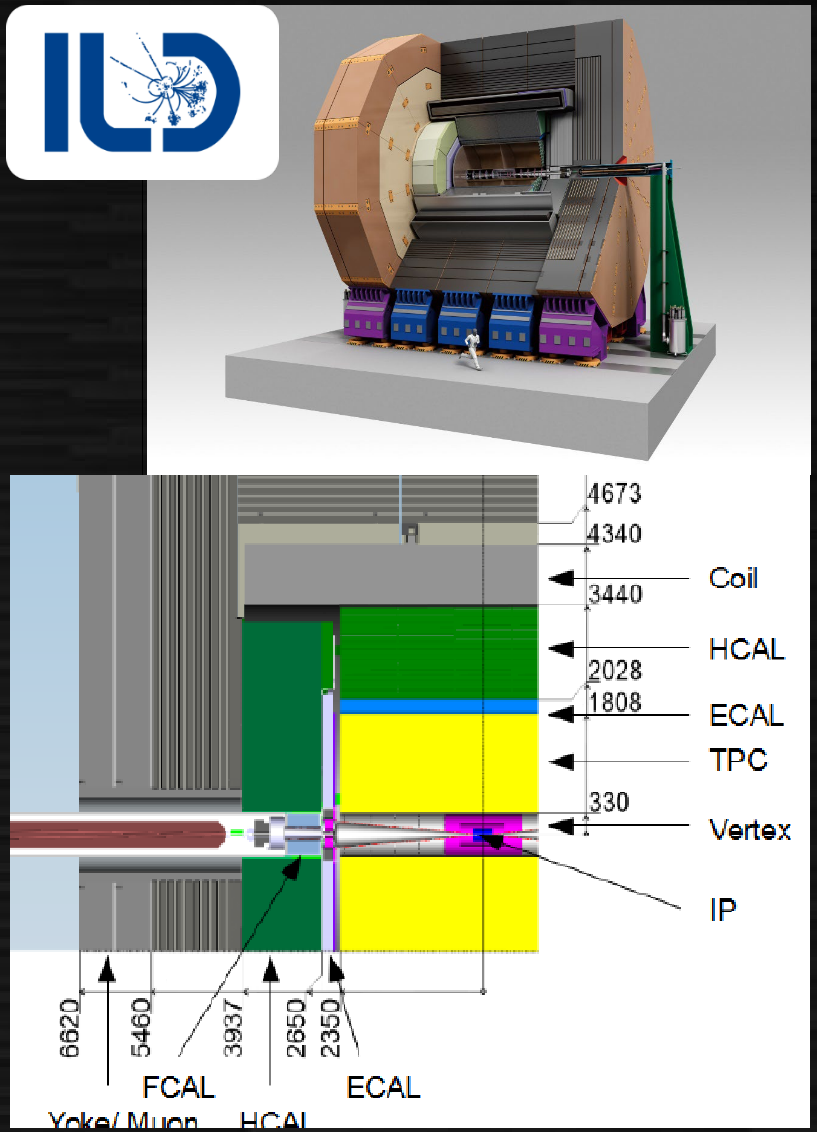
Particle flow with DNN



Key algorithm for particle flow detectors

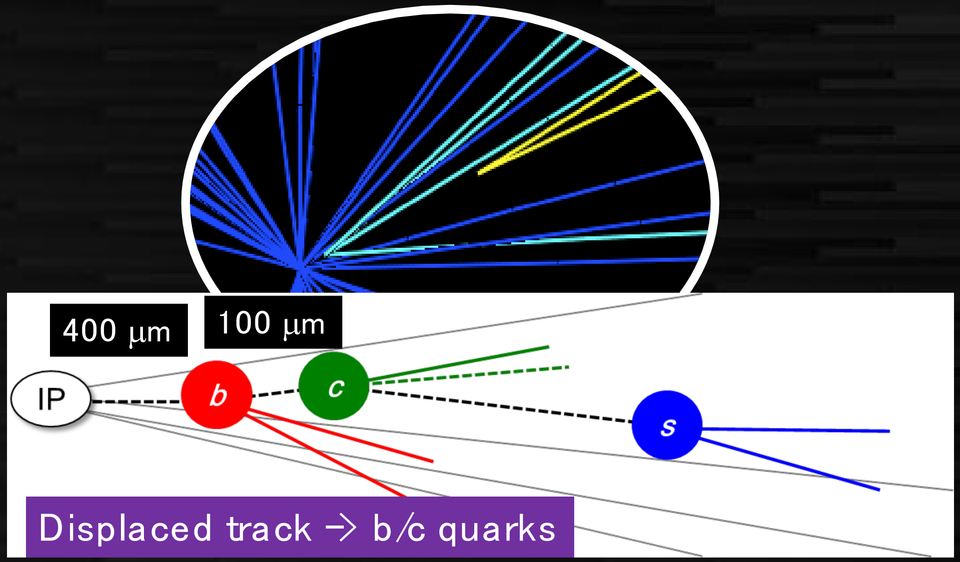
Essential for detector optimization

Detectors for Higgs factories (eg. ILD)



- Vertex detector: monolithic Si pixel
  - Impact parameter resolution: a few μm
  - Innermost layer: 15 mm from IP
- Tracking: Si only or Si + gas
- Hadron ID: dE/dx (or dN/dx), ToF, Cherenkov
  - Essential for strange tagging
- Calorimeter: highly granular calorimeter
  - ECAL: 5x5 mm^2 pixels, 20-30 layers, W absorber
  - HCAL: 1-9 cm^2 pixels, -48 layers, Fe/W absorber
  - Particle flow: separate particles inside jets
    - Hits sharing multiple particles not important
  - Solenoid outside HCAL

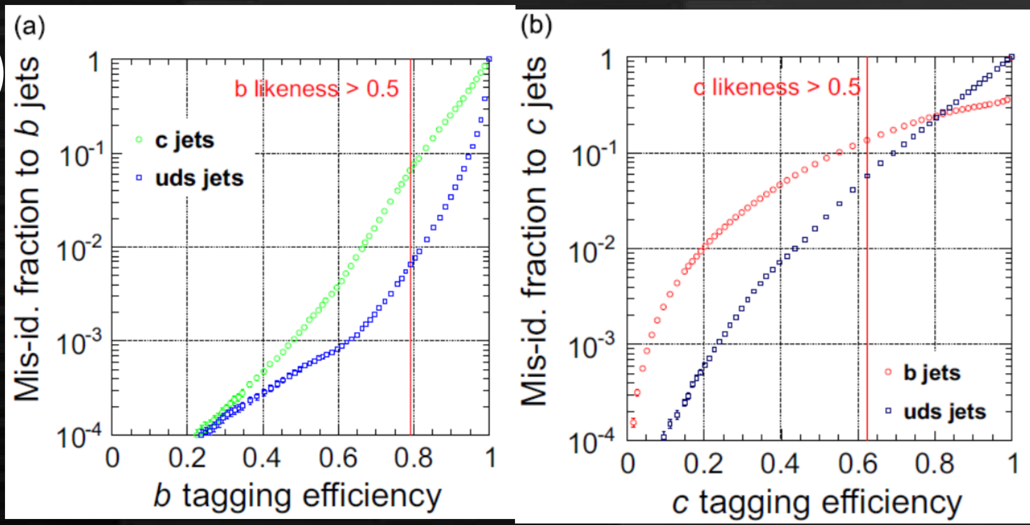
Flavor tagging with Particle Transformer (ParT)

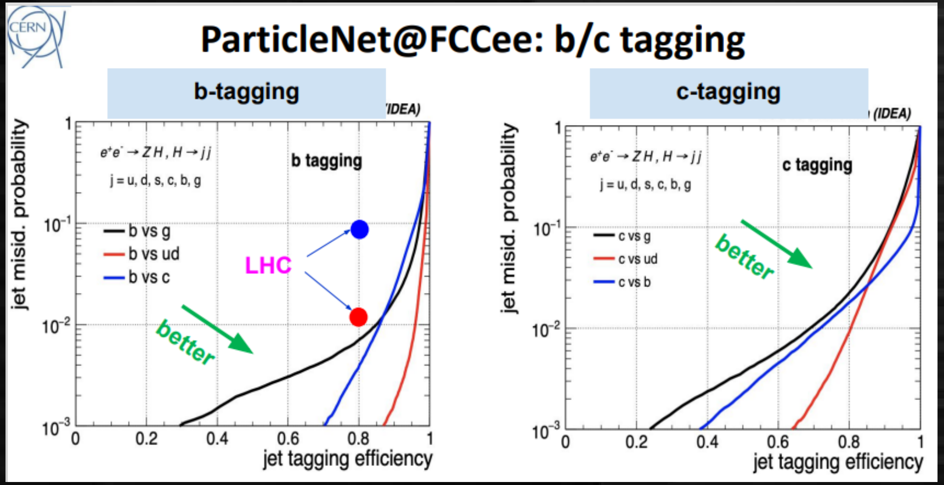


Flavor tagging for Higgs factories

- Jet flavor tagging is essentially important for Higgs studies (including self coupling)
- LCFIPlus (published 2013) was long used for flavor tagging
  - All physics performance in ILD/SiD/CLIC are based on LCFIPlus
- FCCee reported >10x better rejection using ParticleNet (GNN) in 2022
  - Delphes is used for simulation
- We studied DNN-based flavor tag with ILD full simulation to confirm it
  - Using latest algorithm: Particle Transformer (ParT)

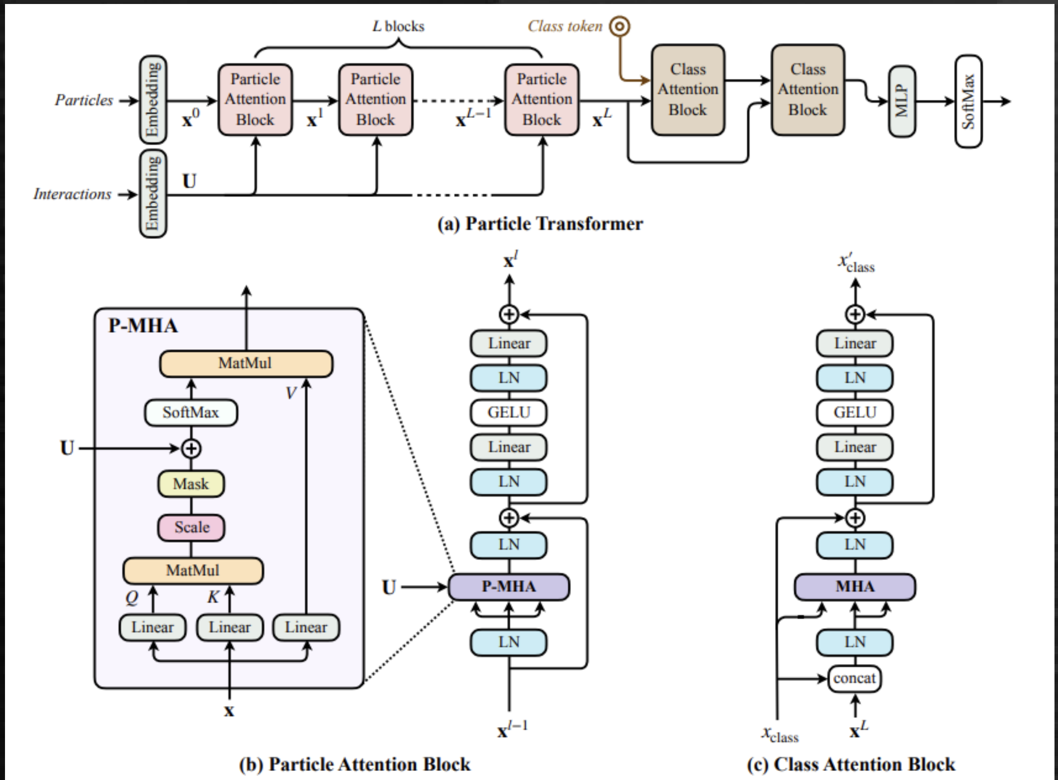
LCFIPlus performance plots





Particle Transformer (ParT)

- Transformer: self-attention-based algorithm intensively used for NLP (e.g. chatGPT)
  - Weak biasing: possible to train big samples efficiently (with more learnable weights) but demanding big training sample for high performance
- ParT is a new Transformer-based architecture for Jet tagging, published in 2022.
  - Pair-wise variable (angle, mass etc.) is added to plain Transformer encoder to boost attention
- Surpasses the performance of ParticleNet
  - ParticleNet only looks “neighbor” particles while Transformer uses attention to learn where to look



	All classes	
	Accuracy	AUC
PFN	0.772	0.9714
P-CNN	0.809	0.9789
ParticleNet	0.844	0.9849
ParT	**0.861**	**0.9877**

Performance with JetClass event classification (100M sample)

Data Samples and Input Variables

Data samples

- ILD full simulation

  1. e+ e- → qq (at 91 GeV) (used in LCFIPlus study)

  q = b,c,uds

  j=b,c,u,d,s,g

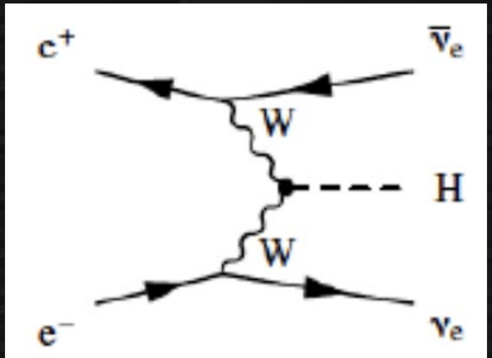
  2. e+ e- → ννH → ννjj (at 250 GeV) (2020 production)

1M jets (500k events) for each flavor

- FCCee fast simulation (Delphes with IDEA detector):

  e+ e- → ννH → ννjj (at 240 GeV)

10M jets (5M events) each flavor



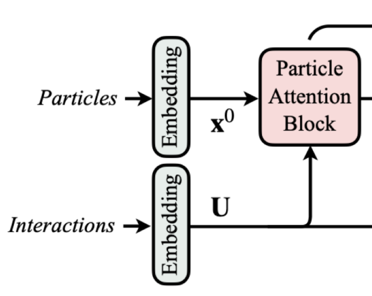
80% for training
5% for validation
15% for test

Input variables

Particles: for every track/neutral

- Impact parameters (6)
  – 2D/3D, from primary vertex
- Jet distance (2)
  – Displacement from jet axis
- Covariant matrix (15)
- Kinematics (4)
  – Energy fraction, angles, charge
- Particle ID (6)
  – Probability (or binary selection) of e, μ, hadron, gamma, neutral hadron

Input of ParT



Interactions: for every particle pair

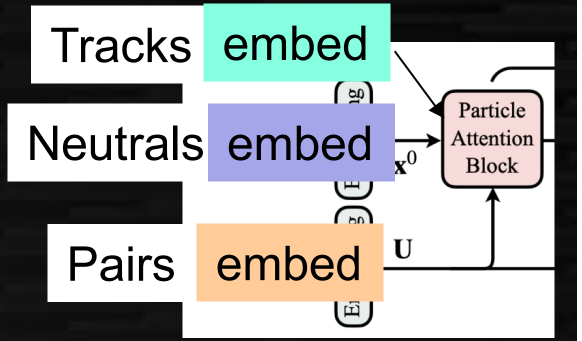
- δR^2, k_t, Z, mass

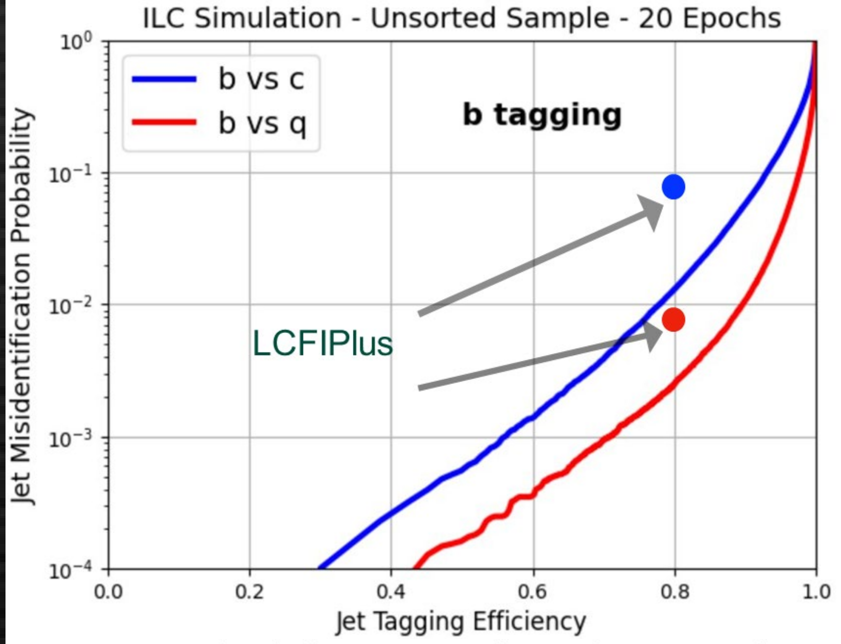
Improvements wrt. LCFIPlus

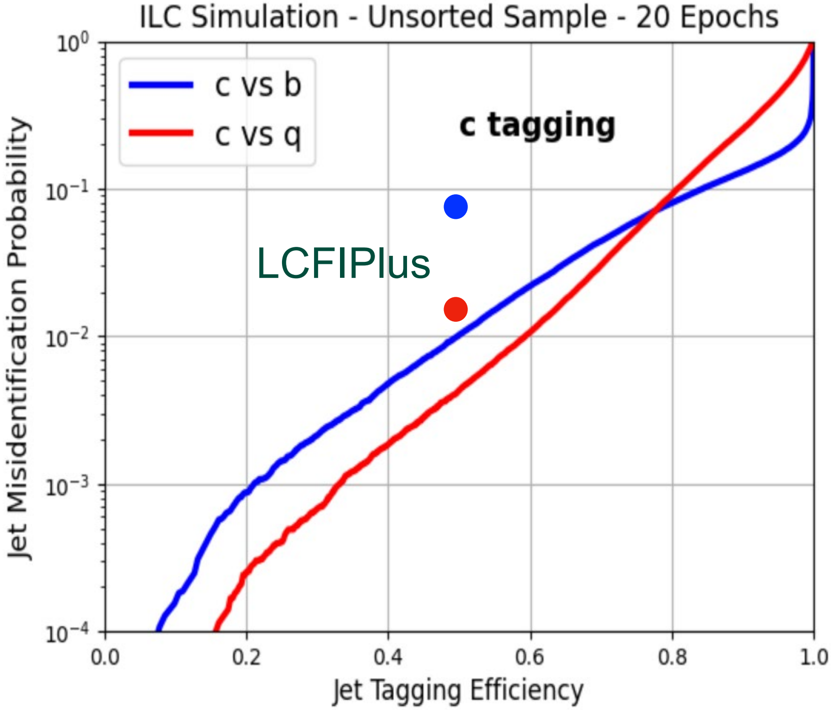
- Factor (3-9) improvement at ParT from LCFIPlus without any tuning
- Another factor (max 3) improvement by tuning
  - Optimizing input variables
  - Separate embedding for tracks/neutrals

	b-tag 80% eff.		c-tag 50% eff.	
background	c jets	uds jets	b jets	uds jets
+LCFIPlus (BDT)	6.3%	0.79%	7.4%	1.2%
*ParT (initial)	1.3%	0.25%	1.0%	0.43%
**ParT (improved)	0.48%	0.14%	0.86%	0.34%

+LCFIPlus (BDT) 250 GeV nnqq
*ParT (initial) 91 GeV qq, default settings
**ParT (improved) 250 GeV nnqq, b/c/d separation







Comparison with FCCee results

Conditions

- FCCee data provided by M. Selvaggi (as ROOT files including input variables)
- Processed with our script (using weaver (by H. Qu) and based on provided configuration)

Results

- FCCee 1M gives ~2x better
  - Comparable with reduced inputs
- FCCee 4/6/8M gives much better
  - Sample size dependence needs to be investigated with ILD (maybe difficult with full simulation)
  - (JetClass has 100M events)

1M: 800k jets for training
4/6/8M: 4/6/8M jets for training

Sample / sample size	b-tag 80% eff.		c-tag 50% eff.	
background	c jets	uds jets	b jets	uds jets
ILD full-sim 1M (optimized)	0.48%	0.14%	0.86%	0.34%
FCCee Delphes 1M (reduced)	0.47%	0.12%	0.64%	0.10%
FCCee Delphes 1M (full)	0.21%	0.054%	0.36%	0.059%
FCCee Delphes 4M	0.045%	0.025%	0.20%	0.033%
FCCee Delphes 6M	0.014%	0.010%	0.13%	0.022%
FCCee Delphes 8M	0.007%	0.006%	0.076%	0.021%

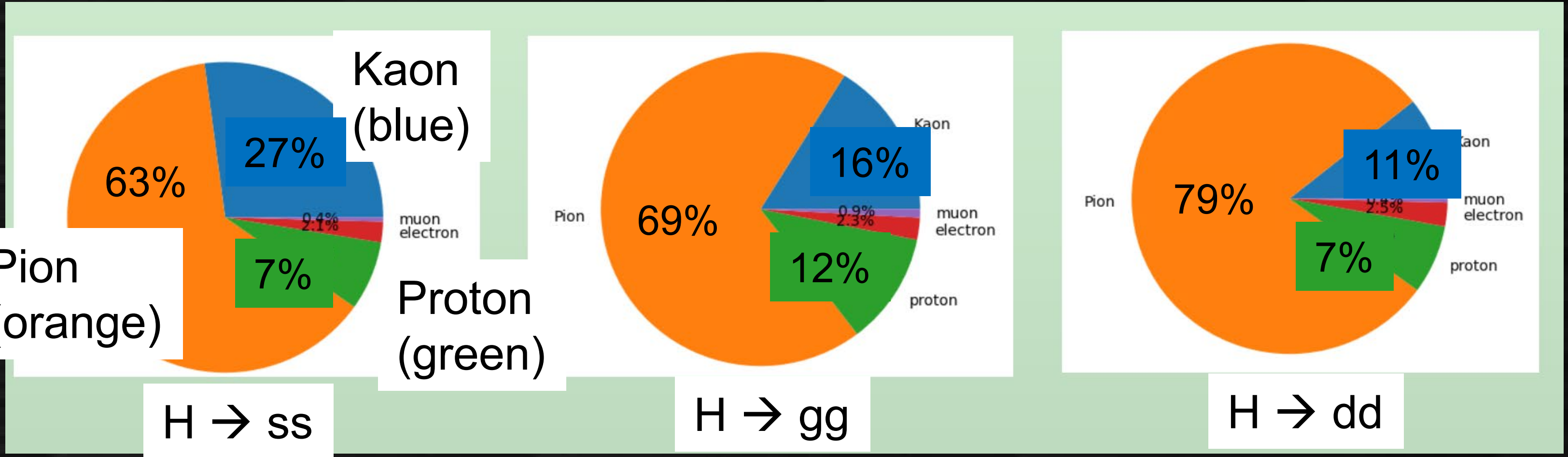
We see mild consistency between ILD and FCC!

FCCee configurations:

- Simulation: Delphes (IDEA geometry)
- Input: Kinematic/Impact parameter/Track error /Particle ID (including TOF and dn/dx) (not with reduced)
- Slight difference with ILD variables (e.g. interaction)

Strange tagging

- High-momentum kaon in jet is a clue to strange jets
  - Contamination from g→ss give relatively low momentum
- dE/dx is essential for Particle ID in ILD
  - As well as ToF, but only effective in low energy tracks (which are less important in strange tagging)
- Using newly-developed comprehensive PID
  - Giving much better separation than previous PID



Fractions of tracks having > 5 GeV

More Kaons in ss
More protons in gg

Fraction of true particles

True particle (columns) vs. CPID prediction (rows), ↑ 3 < p < 5 GeV

CPID prediction	K	π	p	e	μ
K	0.65	0.04	0.20	0.04	0.10
π	0.08	0.90	0.04	0.32	0.28
p	0.26	0.04	0.76	0.09	0.08
e	0.00	0.00	0.00	0.53	0.01
μ	0.01	0.02	0.00	0.01	0.53

↑ p > 5 GeV

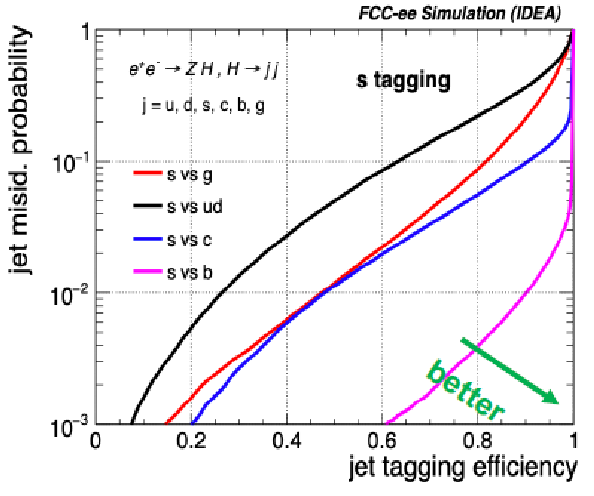
CPID prediction	K	π	p	e	μ
K	0.74	0.07	0.20	0.13	0.16
π	0.07	0.89	0.03	0.40	0.37
p	0.18	0.03	0.76	0.09	0.06
e	0.00	0.00	0.00	0.38	0.01
μ	0.01	0.01	0.00	0.01	0.40

Strange tagging: initial results

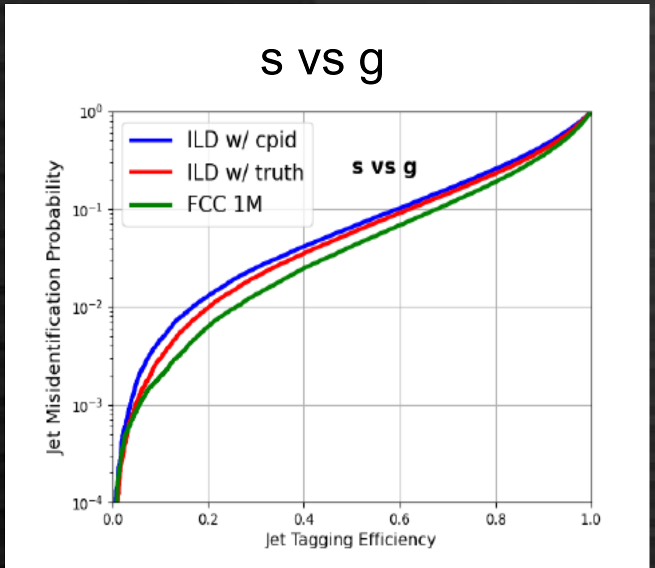
- First results obtained with CPID
  - No significant improvements from old PID: investigating
  - Compared with truth PID: some difference
  - FCC (1M) better than ILD Truth PID
    - Reason needs to be investigated (maybe non-perfect assignment of truth PID)
- Still needs study

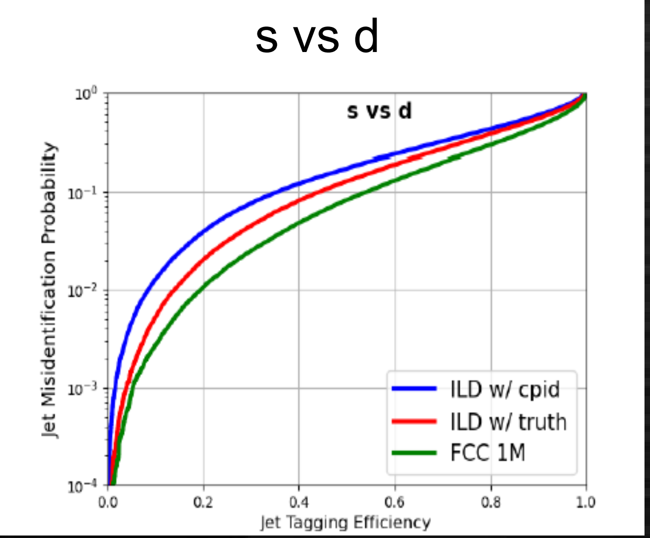
	s-tag 80% eff.	
Method	g-bkg acceptance (%)	d-bkg acceptance (%)
ILD full sim. CPID	25.7	42.7
ILD full sim. Truth PID	23.2	38.0
FCC 1M (PID+tof)	20.3	29.6

Strange tagging performance



FCCee plot (in their study)

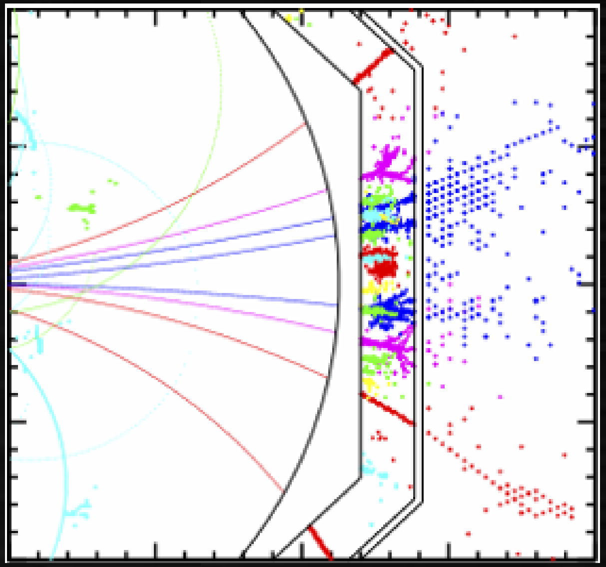




Flavor tagging: summary and plans

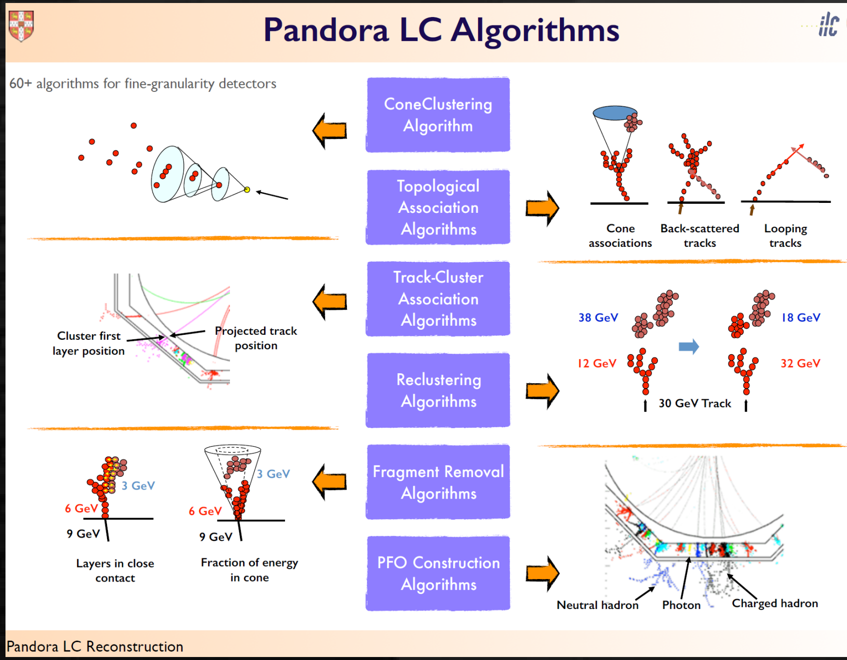
- Significantly better performance of flavor tagging with ParT
  - Implementation to the reconstruction framework foreseen to be applied to real physics analysis
    - Porting with Onnx framework available, performance currently degraded, under investigation
  - Further optimization still possible
- Strange tagging under investigation
  - Performance still needs to be understood more
  - To be fixed soon → to be used for physics analysis (e.g. H→ss)
  - Dependence on PID performance to be investigated
    - Coming with various detector configurations

Particle flow with DNN



Particle flow in Higgs factories

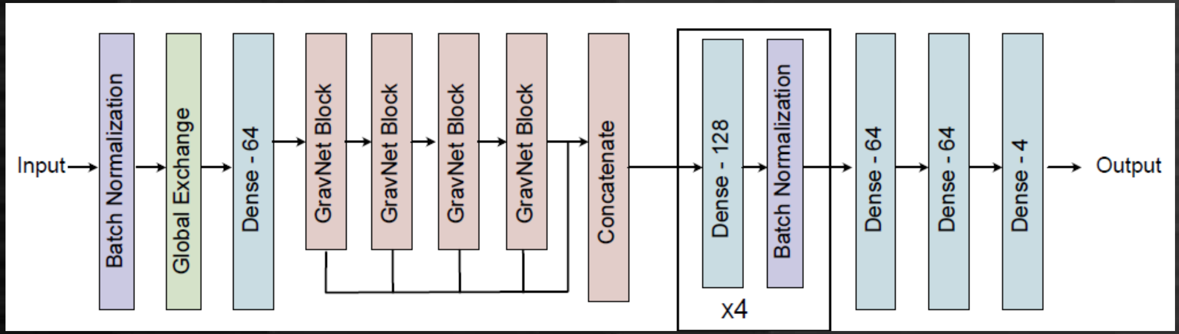
- PandoraPFA is used since 2008 as standard for >15 years
  - Good-old technology but fully tuned only minor modifications since 2008
  - Exceeding PandoraPFA is a long-lasting target for development of PFA
    - Several algorithms gave challenge but no algorithm significantly exceeds the performance and thus not replaced



- Our primary target is to exceed PandoraPFA
  - In addition, DNN-based algorithm has many benefits
    - eg. Easier adaptation to geometries and additional features

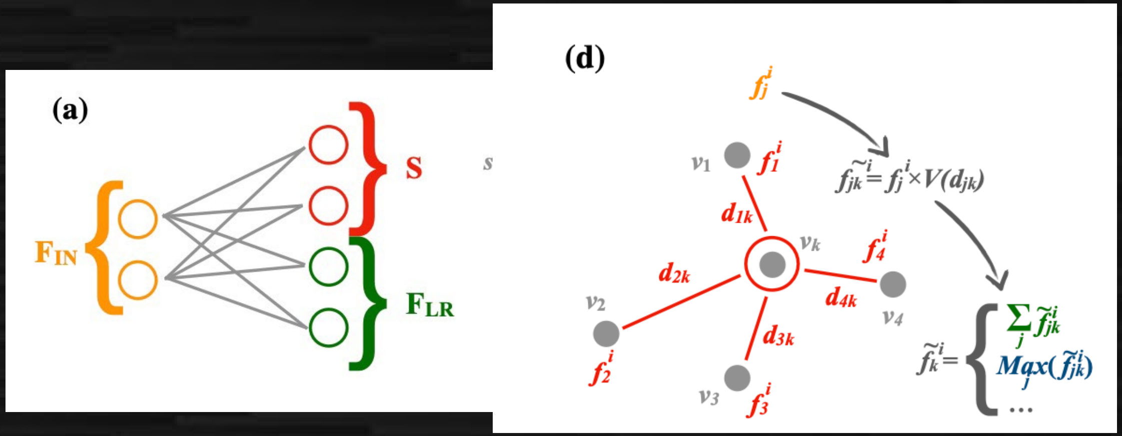
GNN-based PFA

- Originally developed for CMS HGCAL
- Input: position/energy/timing of each hit
- Output: virtual coordinate and β for each hit



GravNet arXiv:1902.07987

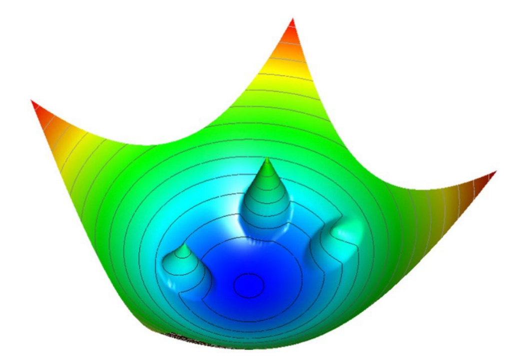
- The virtual coordinate (S) is derived from input variables with simple MLP
- Convolution using "distance" at S (bigger convolution with nearer hits)
- Concatenate the output with MLP



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

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

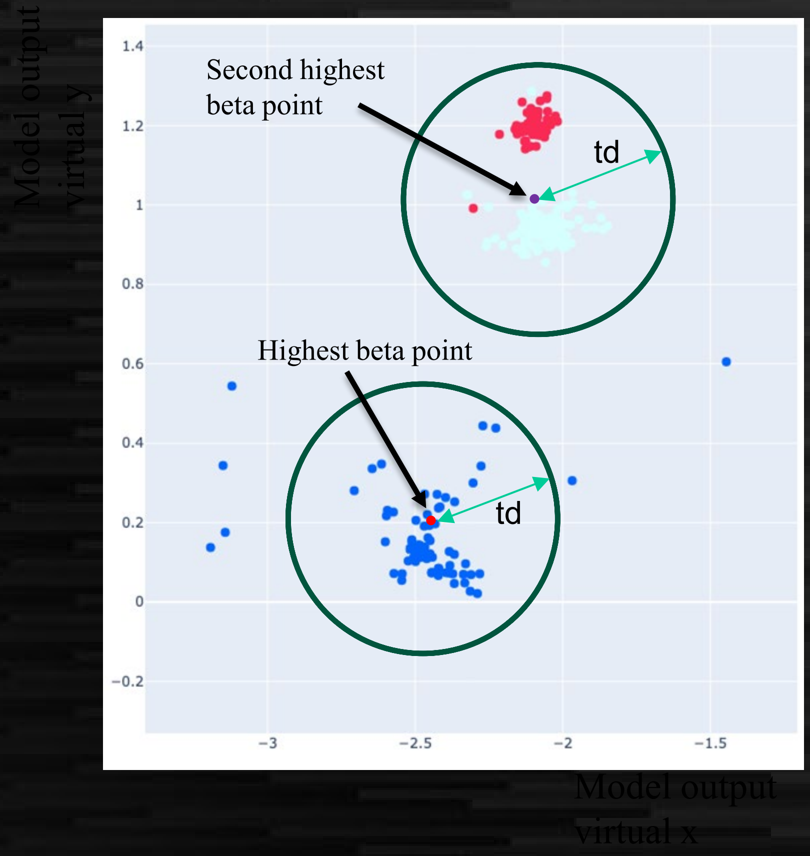
$$L = L_p + s_C(L_\beta + L_V)$$

L_V: attractive/repulsive potential to condensation points / tracks
L_β: Pulling up β of the condensation points / tracks
Tracks are prioritized over other condensation points

Current number of parameters: ~420K

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

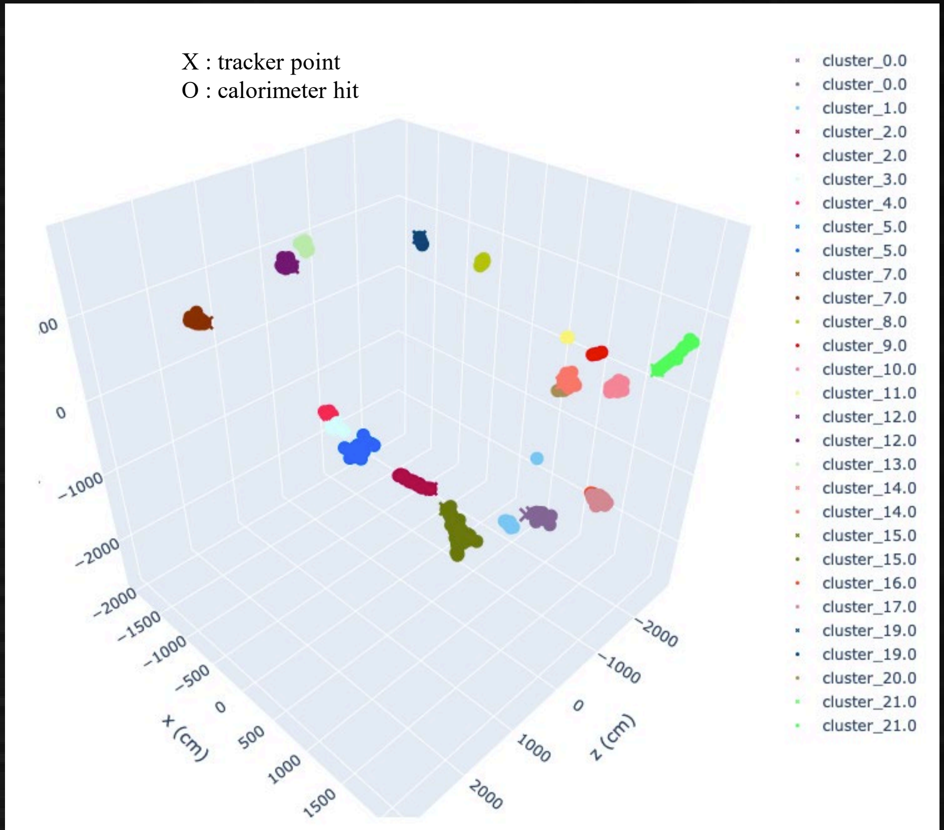


Our samples for performance evaluation

- ILD full simulation with SiW-ECAL and AHCAL
  - ECAL: 5 x 5 mm^2, 30 layers, HCAL: 30 x 30 mm^2, 48 layers
  - Taus overlayed with random direction
    - 100k events, 10 GeV x 10 taus / event → 1 million taus
  - qq (q=u, d, s) sample at 91 GeV
    - ~75k events
    - Official sample for PFA calibration (other energies available)
  - Converted to awkward array stored in HDF5 format
    - A few 10 GB each

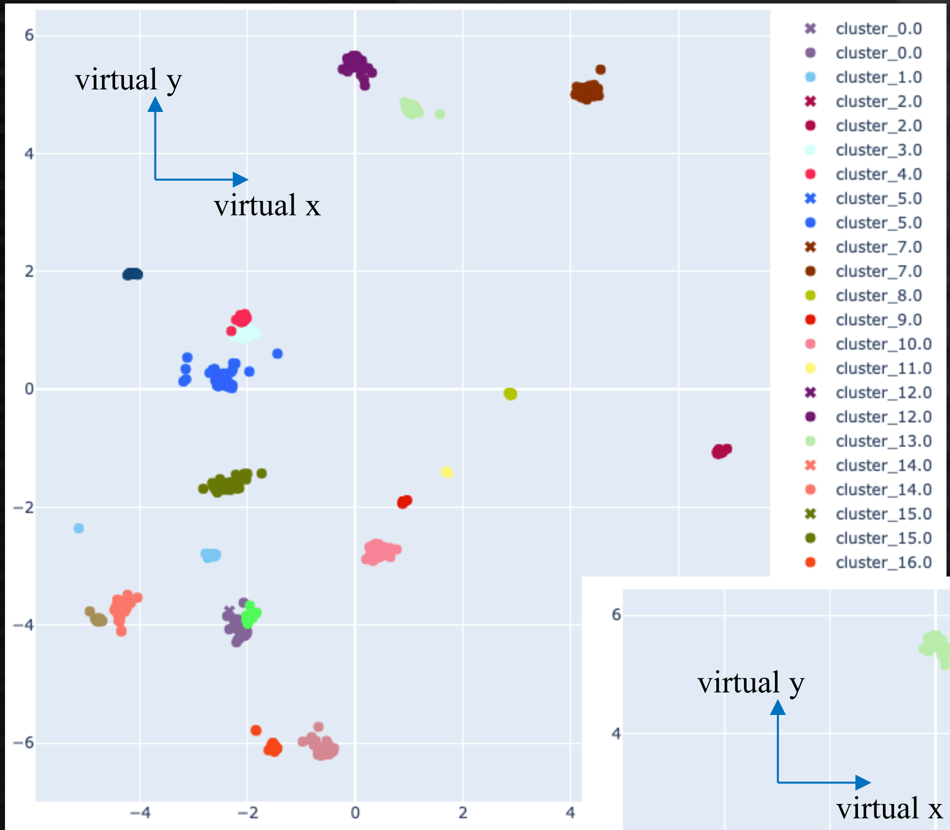
Taus: good mixture of hadrons, leptons and photons with some isolation
Good for training

Event display



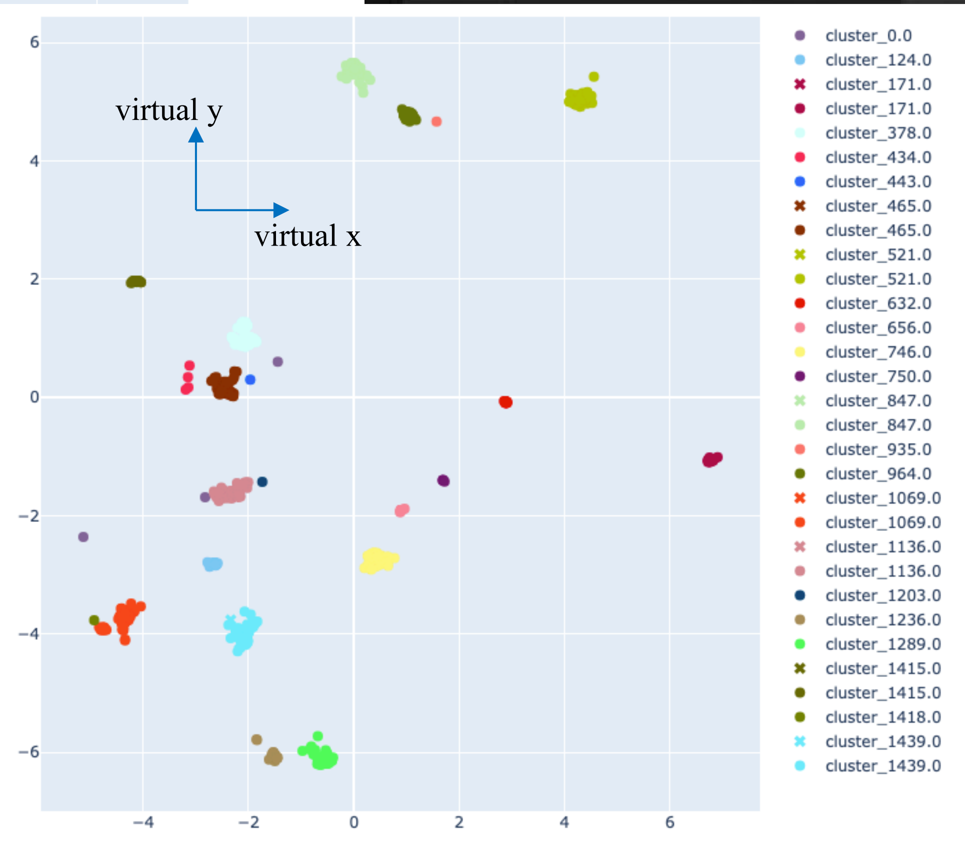
Input features
Real coordinate in detector

Colored by true clusters



Output features
Virtual coordinate

Colored by
true clusters

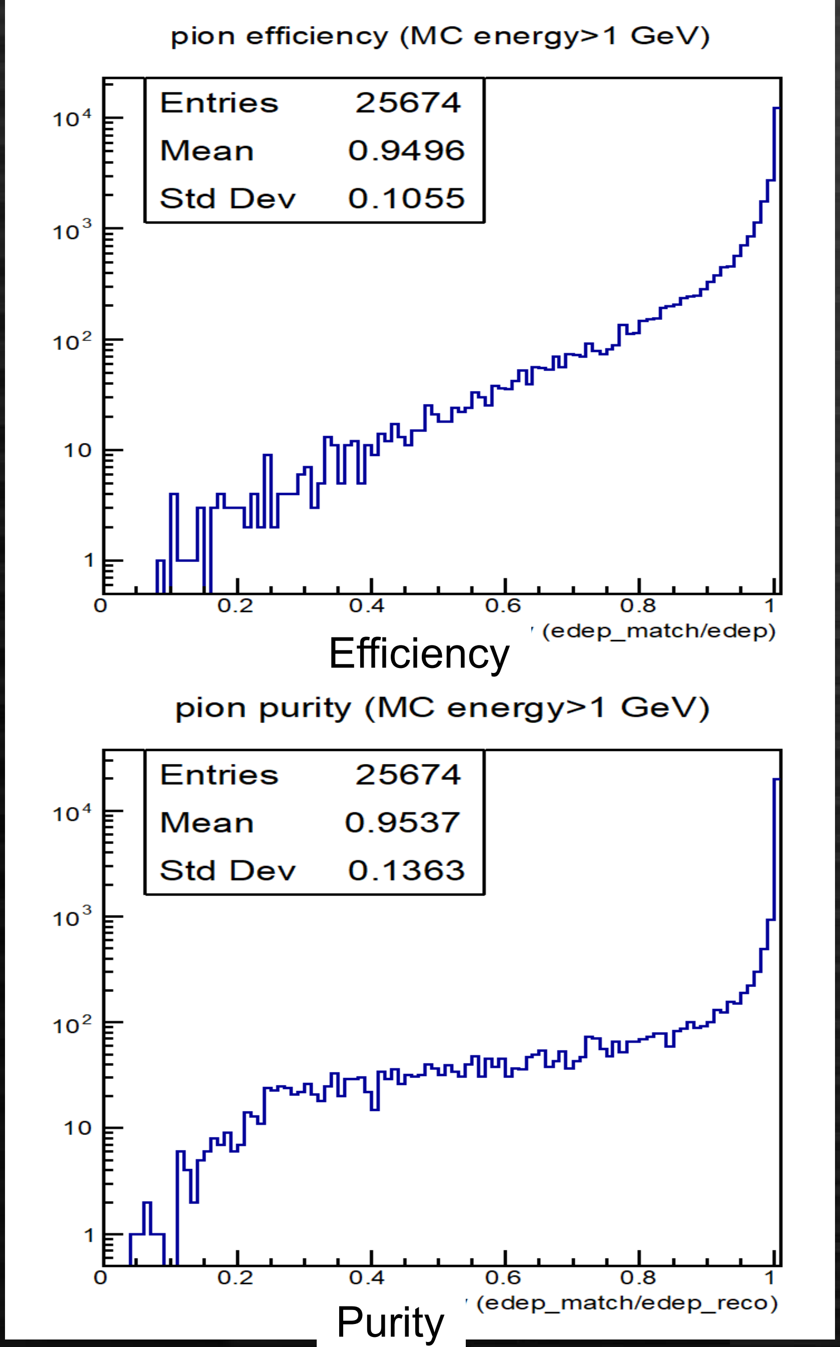


Colored by
reconstructed clusters

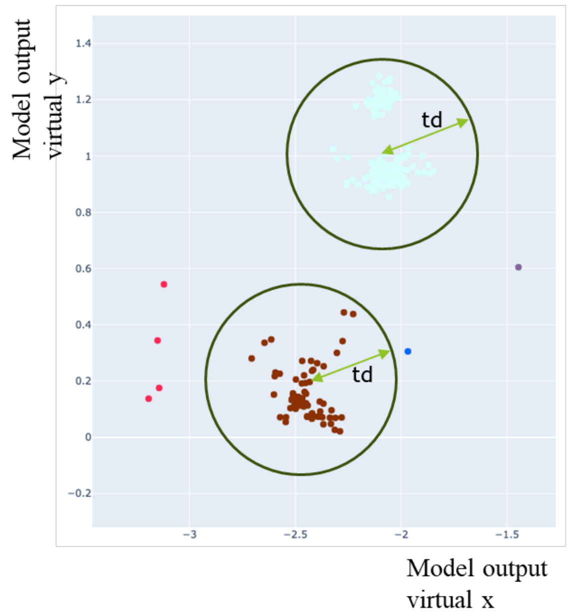
Taikan Suehara

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)



Optimization of performance

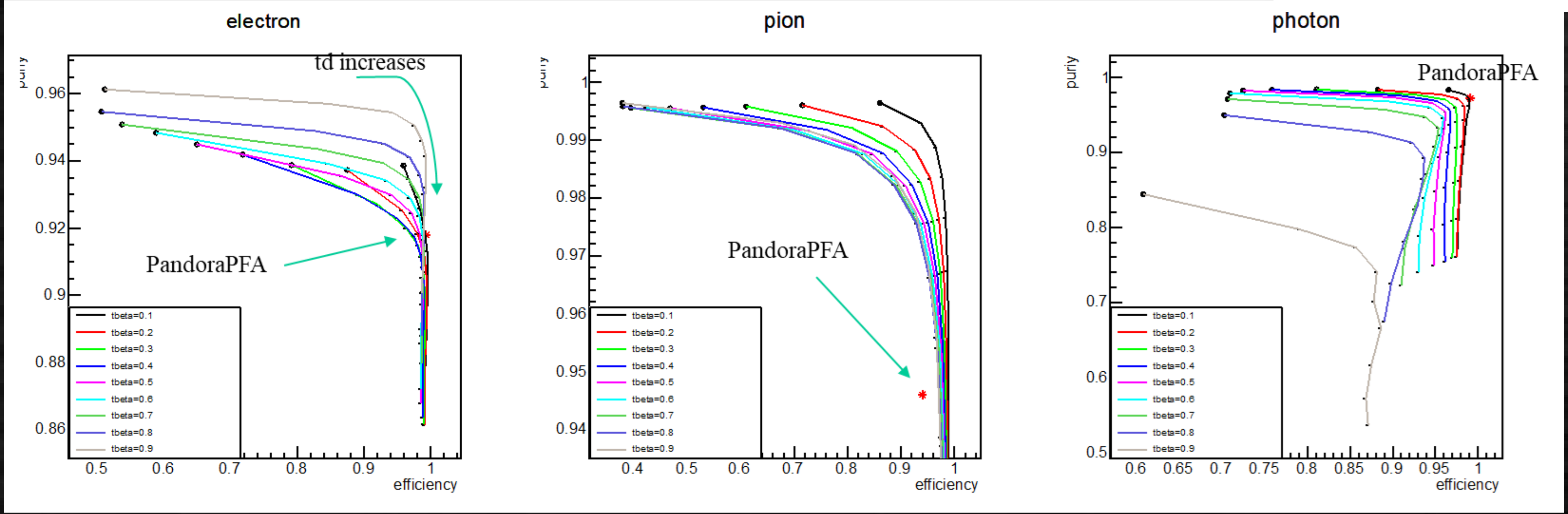


Output dimension of the coordinate

- 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

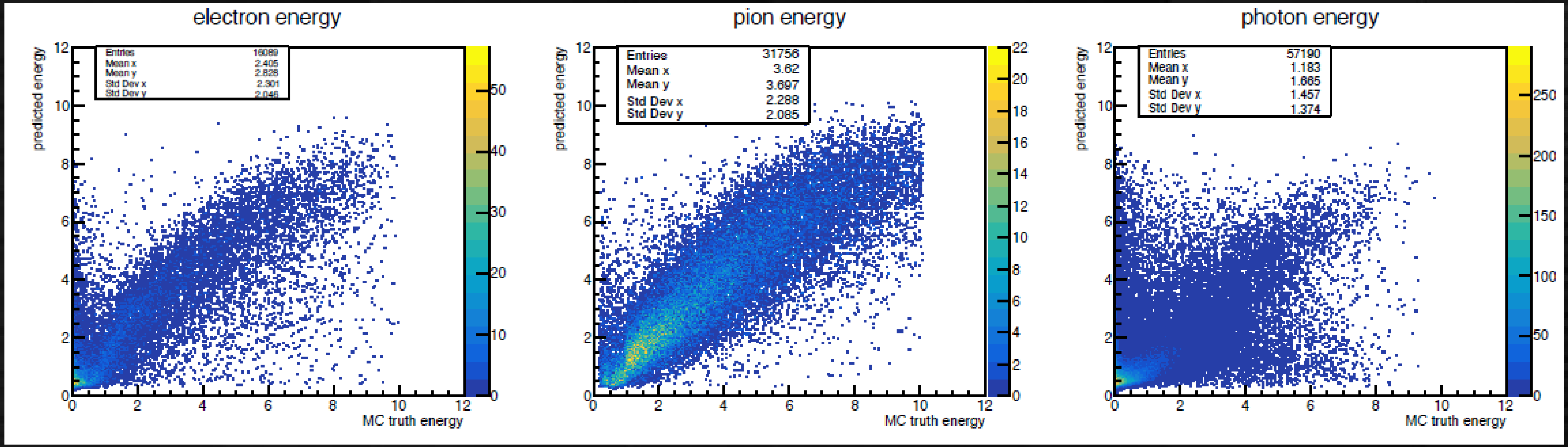
④ $L_E = \sum_i \theta_i (E_i - \varepsilon_i)^2$ $\qquad \theta_i = 1$ if the point is condensation point

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

ε_i : energy related variable

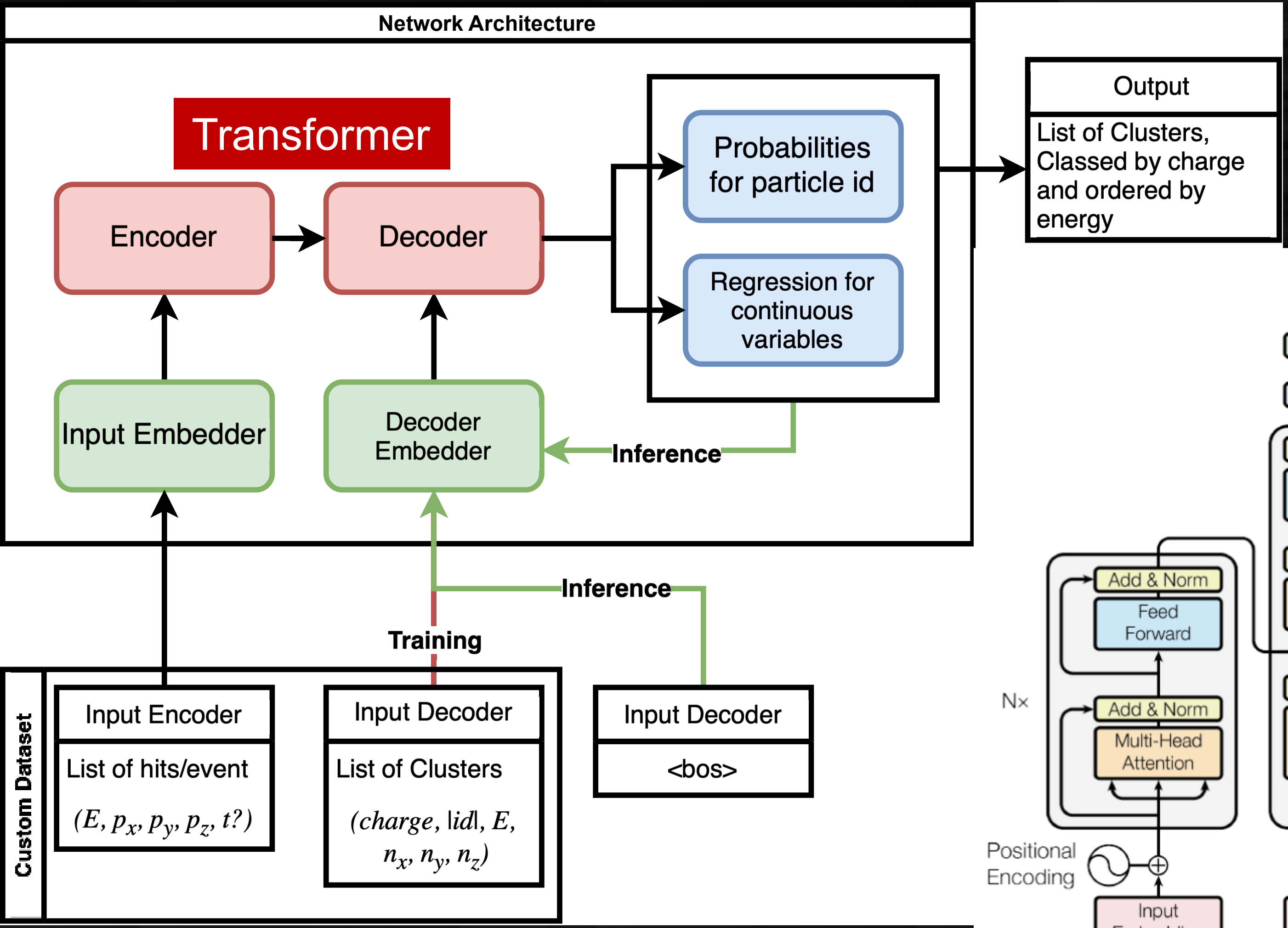
Reasonable correlation to MC energy seen
Performance still to be tuned

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

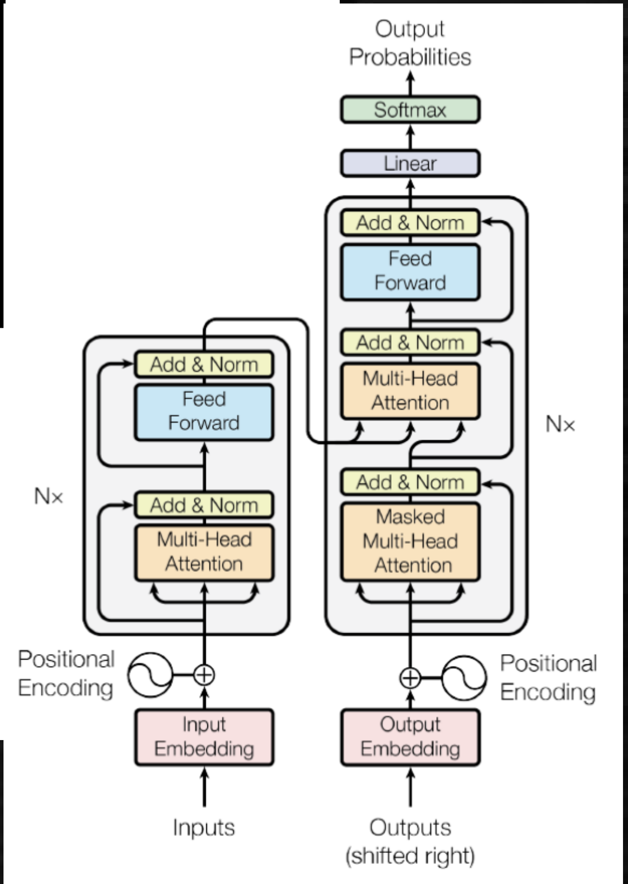


More NLP-like model: transformer

Submitted to E1 in FY2025-26



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

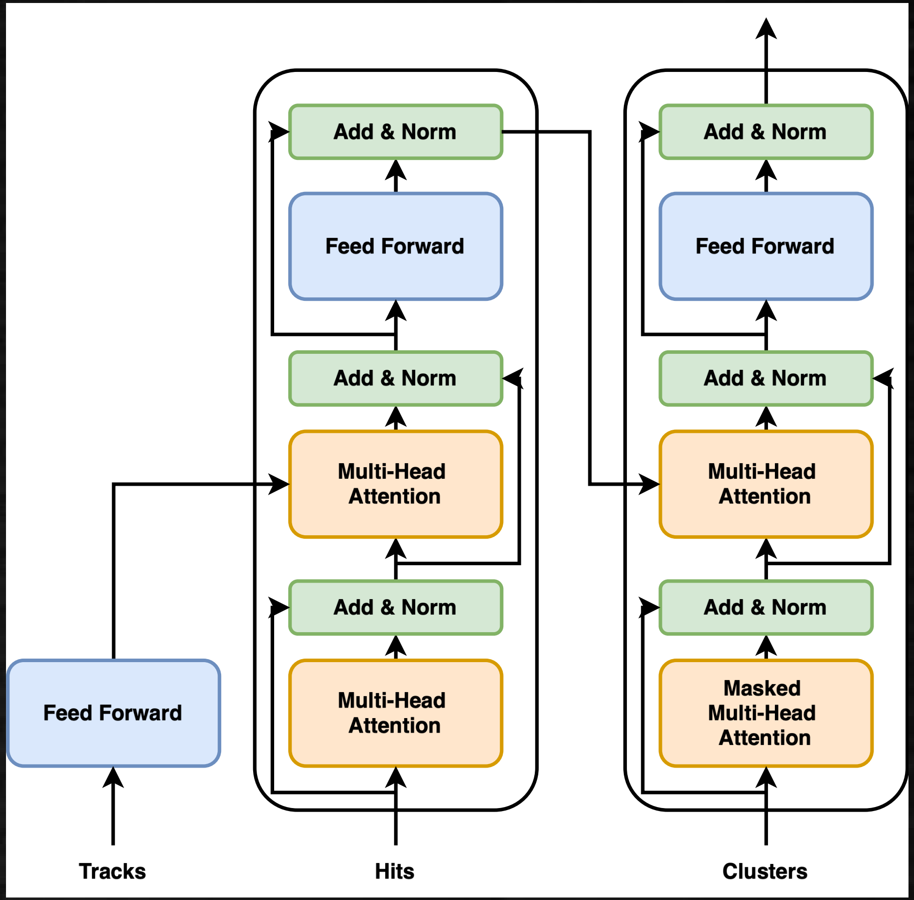
Output info of next cluster

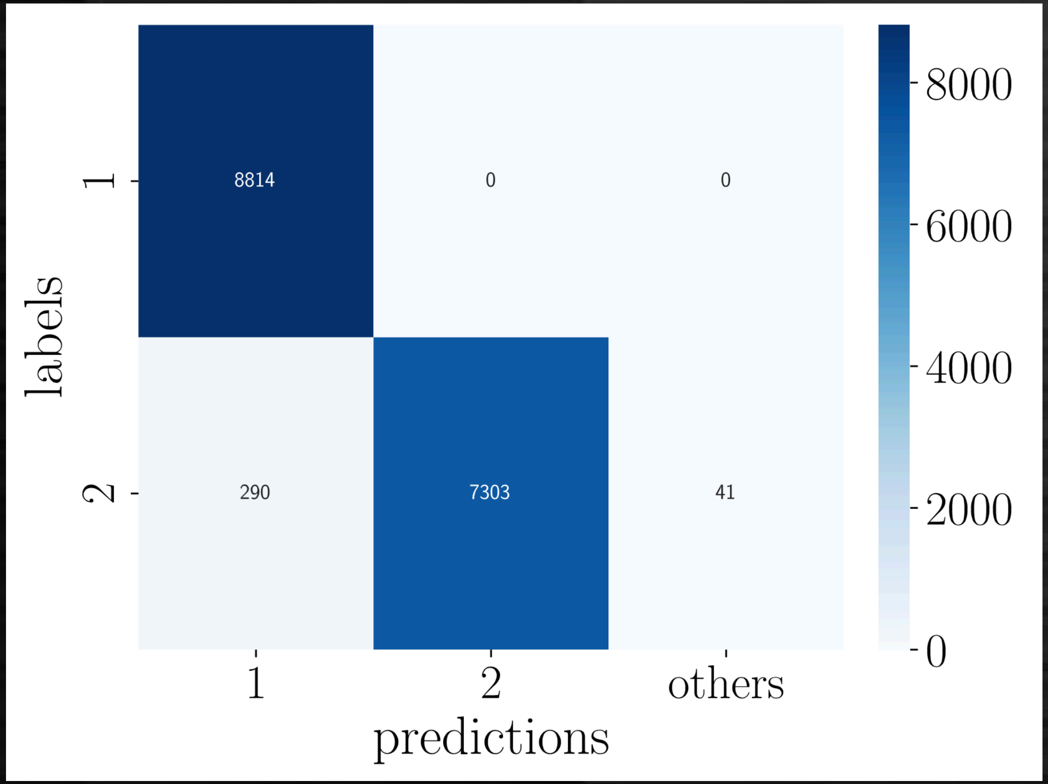
(training) MC truth clusters

(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)

Particle flow: 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…

Overall summary

- High level reconstruction @ ILD has a lot of room to incorporate with DNN to improve performance
  - Also easier to use for detector optimization
- Flavor tagging with ParT significantly better than LCFIPlus
  - To be applied to physics analysis
  - Strange tagging also under investigation
- Particle flow with GNN gives competitive performance
  - Energy regression to be done
  - Hope to replace PandoraPFA in ~a few years
  - NLP-like method also being investigated

Backup

Results on efficiency and purity (another view)

Algorithm train/test	Electron eff.	Pion eff.	Photon eff.	Electron pur.	Pion pur.	Photon pur.
GravNet 10 taus/10 taus	98.8%	99.6%	99.1%	92.6%	99.3%	97.7%
PandoraPFA 10 taus	99.3%	94.0%	99.1%	91.8%	94.6%	97.2%
GravNet jets/jets	94.6%	93.1%	95.2%	77.4%	93.1%	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%

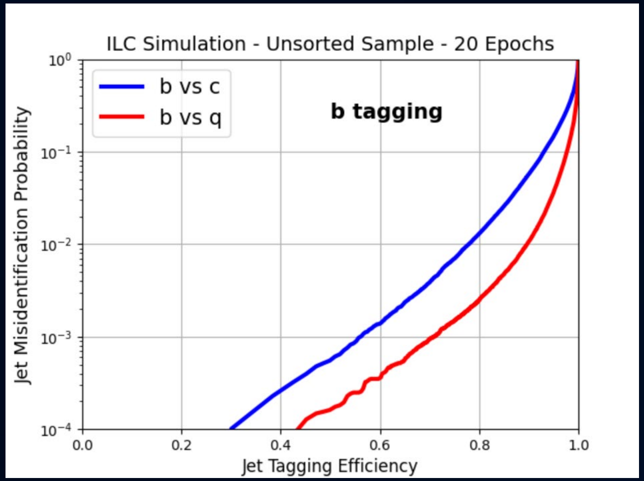
At least in our measure, performance of GravNet-based algorithm exceeds PandoraPFA
→ Promising as full PFA (but energy regression to be done)
Definition of MC truth clusters needs to be tuned (see ILCSoft truth)

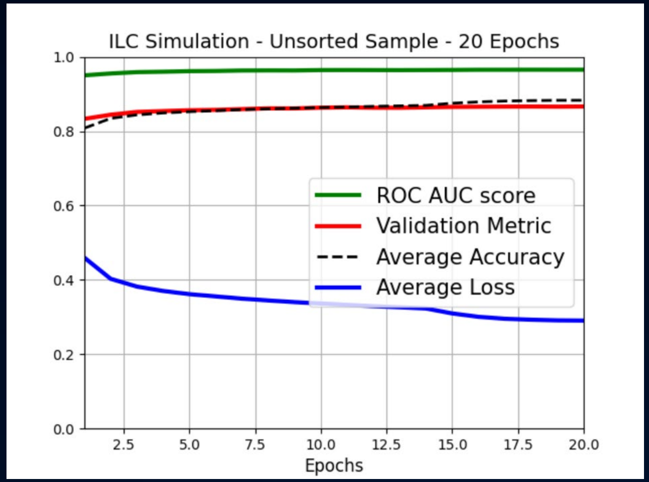
Software for Particle Transformer

- Public in github, with instruction provided
  - https://github.com/jet-universe/particle_transformer
- Input: ROOT files for training (80%), validation (5%), test (15%)
  - Input variables can be provided via steering file (XML)
    - Input for each particle (tracks, neutral clusters)
    - Input for “interaction” → currently momentum only
    - Input for “coordinate” → theta/phi plan wrt. jet axis
- Output: ROOT files including evaluation results (likeness) for test events
  - To be analyzed with ROOT or so
- We implemented a processor (inside LCFIPlus) to produce ROOT files for input as much as compatible to FCCee variables
  - Except for PID values, which are not fully implemented
- Easy for testing, but not direct to be used for physics analyses

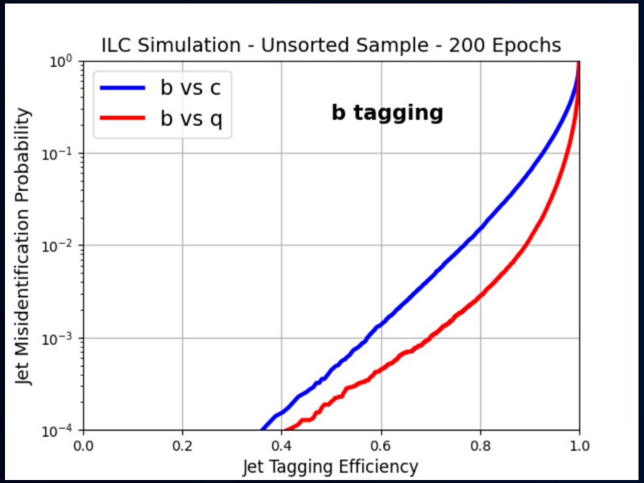
Training parameters - epochs

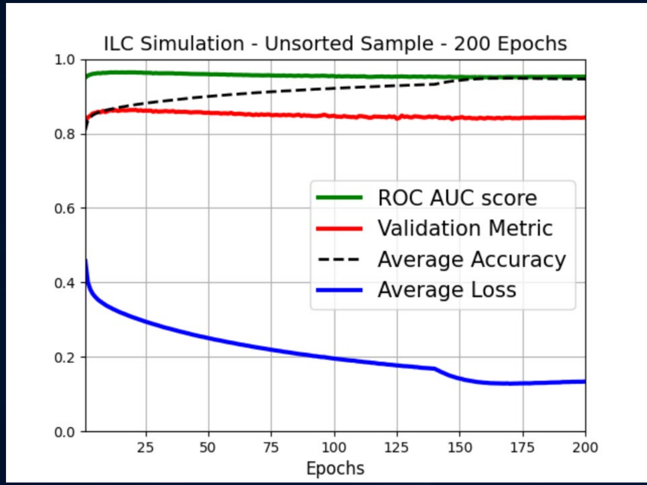
- Run on NVIDIA TITAN RTX (memory: 24 GB)
  - 20 Epochs: 3 hours
  - 200 Epochs: 30 hours
- No significant improvement in tagging efficiency
- Both ROC AUC score and Validation Metric reaches a maximum around 20 epochs.
- Overtraining after 20 epochs.
- Hence 20 epochs of training is selected to avoid overtraining.





20 epochs (ILD qq 91 GeV)





200 epochs (ILD qq 91 GeV)

Input Variables - Features

*Naming follows FCCee scheme – may not express exact meaning

- Impact Parameter (6):
  pfcand_dxy
  pfcand_dz
  pfcand_btagSip2dVal
  pfcand_btagSip2dSig
  pfcand_btagSip3dVal
  pfcand_btagSip3dSig

*d0/z0 and 2D/3D impact parameters, 0 for neutrals

- Jet Distance (2):
  pfcand_btagJetDistVal
  pfcand_btagJetDistSig

*Displacement of tracks from line passing IP with direction of jet
0 for neutrals

- Particle ID (6):
  pfcand_isMu
  pfcand_isEl
  pfcand_isChargedHad
  pfcand_isGamma
  pfcand_isNeutralHad
  pfcand_type

* Not including strange-tagging related variables (TOF, dE/dx etc.)
* Simple PID for ILD, not optimal

- Kinematic (4):
  pfcand_erel_log
  pfcand_thetarel
  pfcand_phirel
  pfcand_charge

*Fraction of the particle energy wrt. jet energy (log is taken)

- Track Errors (15):
  pfcand_dptdpt
  pfcand_detadeta
  pfcand_dphidphi
  pfcand_dxydxy
  pfcand_dzdz
  pfcand_dxydz
  pfcand_dphidxy
  pfcand_dlambdadz
  pfcand_dxyc
  pfcand_dxyctgtheta
  pfcand_phic
  pfcand_phidz
  pfcand_phictgtheta
  pfcand_cdz
  pfcand_cctgtheta

*each element of covariant matrix
0 for neutrals

Input Variables - Interactions

- FCC data uses p (scalar momentum) as interaction:

  - pfcand_p

- ILD data contains p_x, p_y, p_z (vector momentum) as interaction:

  - pfcand_px
  - pfcand_py
  - pfcand_pz

- But it's possible to transfer ILD's interaction to FCC's form for fair comparison:

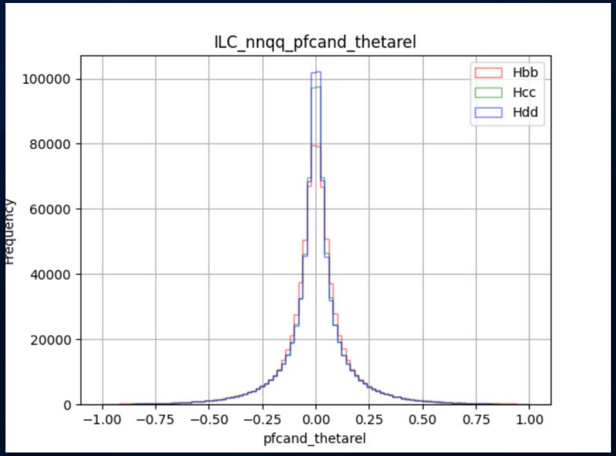
$$p = \sqrt{{p_x}^2 + {p_y}^2 + {p_z}^2}$$

Use px, py, pz instead of p (Interaction)

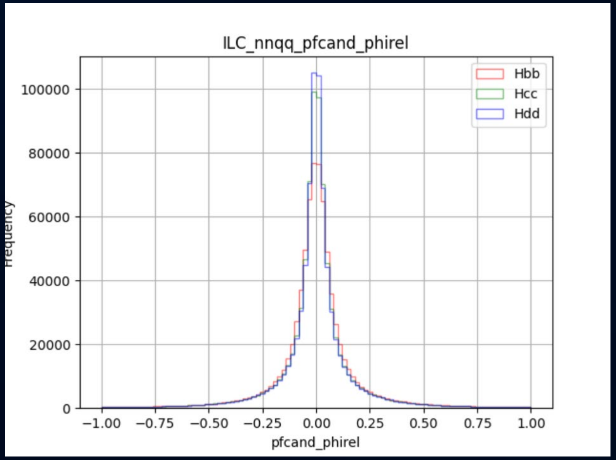
				c-bkg acceptance @ b-tag 80% eff.		b-bkg acceptance @ c-tag 50% eff.	
Particle ID	Impact Parameters	Jet Distance	Track Errors	p	p_x p_y p_z	p	p_x p_y p_z
✗	●	●	●	0.62%	0.49%	1.14%	1.01%
✗	● +log(abs)	● +log(abs)	● +log(abs)	0.54%	0.52%	1.06%	1.00%
✗	● +log(abs)	●	●	0.47%	0.50%	1.03%	0.97%

- ILD (ννqq 250 GeV) data shows that application of px, py, pz has better performance than p.
- However, application of log(abs) of the parameters becomes less significant.
- Can be because that application of px, py, pz changes the way log(abs) interacts with other parameters.
- Other potential treatments can be investigated.

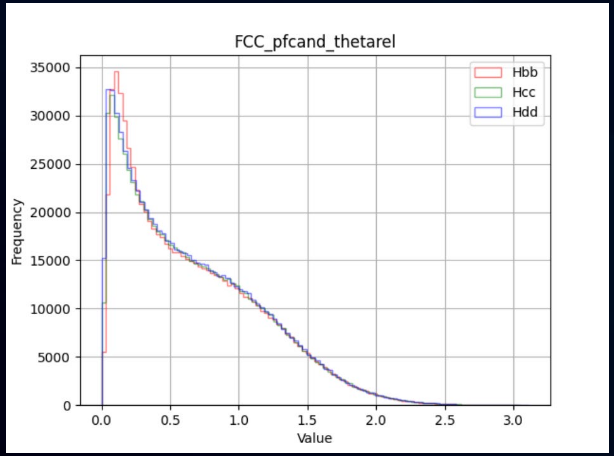
ILD vs. FCC – theta/phi distribution



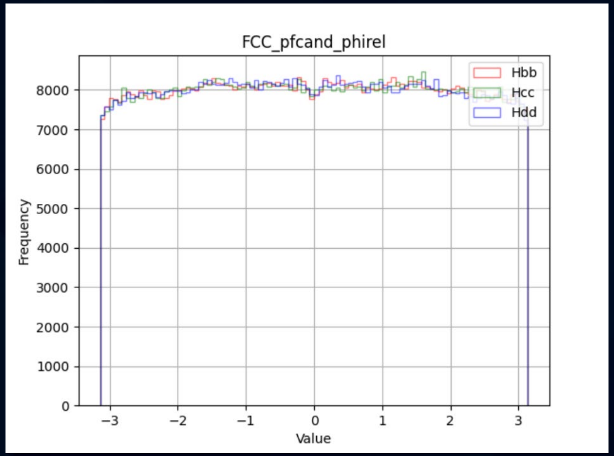
ILD theta



ILD phi



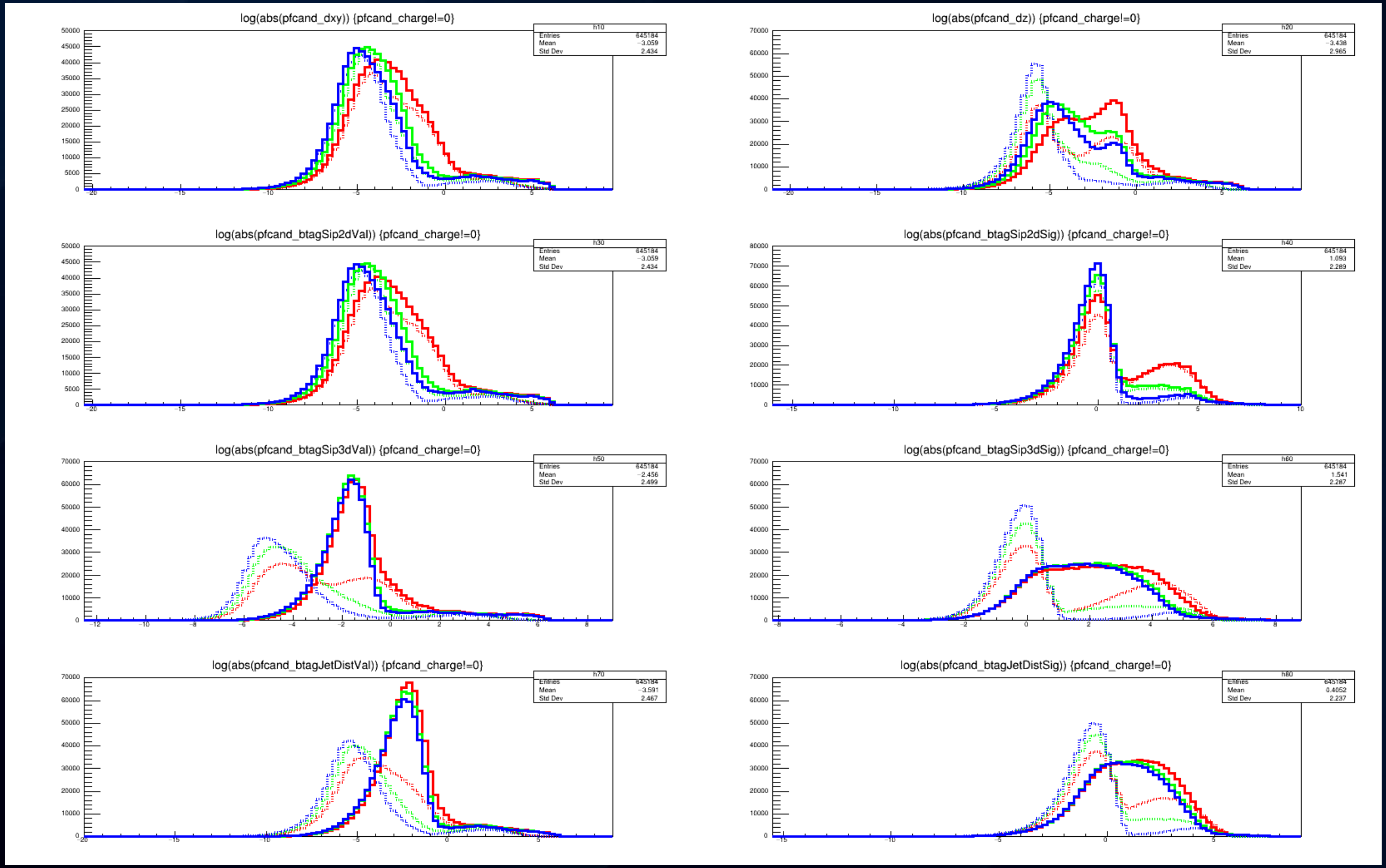
FCC theta



FCC phi

- ILD theta/phi are calculated from the difference between particle and jet theta/phi in the frame of the detector.
- FCC theta/phi are obtained from relative trace of the particle compared to the jet.
- This can cause some differences in the interaction of other parameters in the model.

Difference in impact parameters



Dotted – FCCee
Solid – ILD

Red – nnbb
Green – nncc
Blue – nndd

Significant difference on dz seen
- beam spot smearing?

Fine tuning

Two objectives

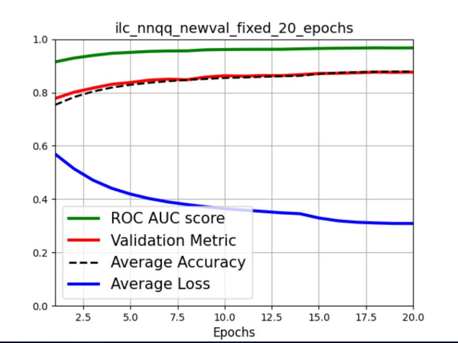
- Pretrained with fast sim and fine-tune with full sim
- Pretrained with large central production and fine-tune with dedicated physics samples in each analysis

							c-bkg acceptance @ b-tag 80% eff.		b-bkg acceptance @ c-tag 50% eff.	
Particle ID	Impact Parameters	Jet Distance	Track Errors	Fine-Tuning Sample	Training Sample	Similar theta/phi ?	No Fine-Tuning	With Fine-Tuning	No Fine-Tuning	With Fine-Tuning
✗	●	●	●	FCC 240 GeV (8M)	ILD 250 GeV (800k)	✗	0.62%	1.37%	1.14%	1.95%
✗	●	●	●	FCC 240 GeV (8M)	ILD 250 GeV (800k)	●	1.77%	1.32%	2.22%	2.01%
●	●	●	●	ILD 250 GeV (800k)	ILD 91 GeV (80k)	●	4.49%	0.97%	3.79%	1.53%

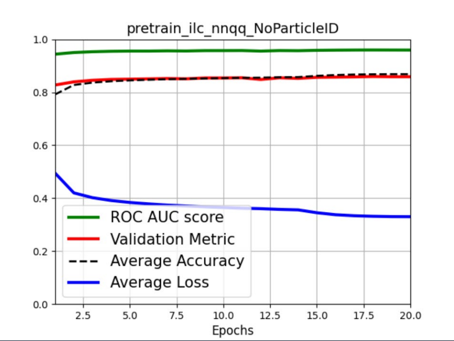
- Use result of 8M FCC data to train ILD 800k data
- Improves performance only when setups are similar
- Training of same setup (pretrain ILD 91 GeV data with ILD 250 GeV data) gives best performance
- Further investigation should be conducted on how to maximise the outcome for fine-tuning between different data sets

Fine tuning – Training curves

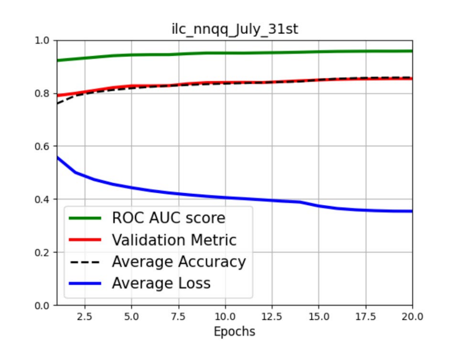
(1)



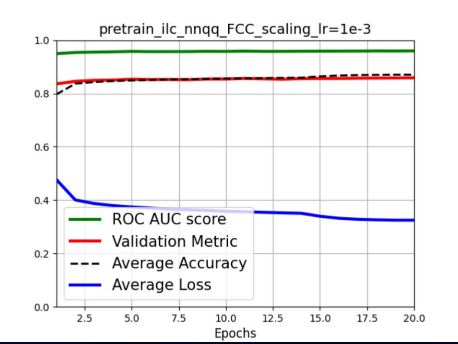
(2)



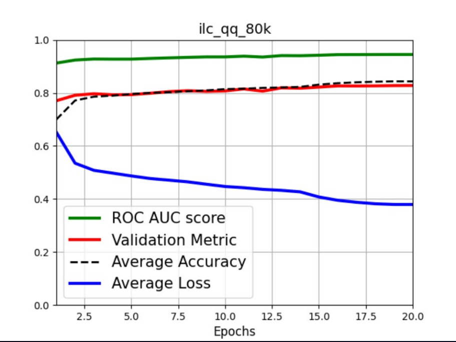
(3)



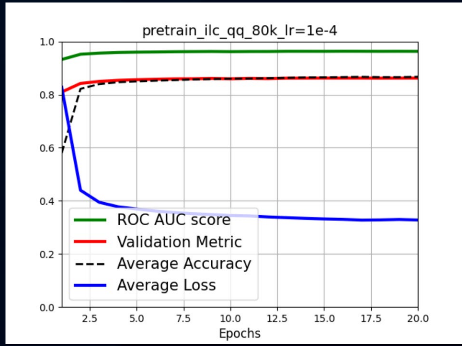
(4)



(5)



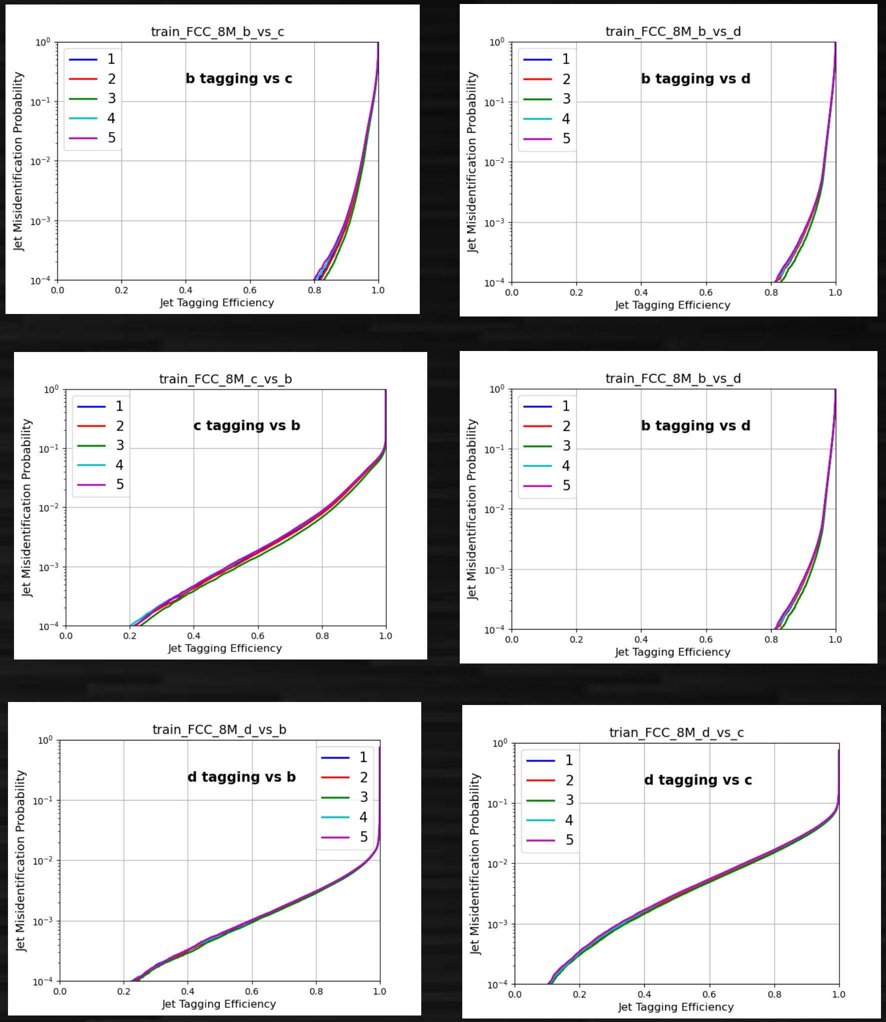
(6)



							Plot Indices	
Particle ID	**Impact Parameters**	**Jet Distance**	**Track Errors**	**Fine-Tuning Sample**	**Training Sample**	**Similar theta/ phi?**	**No Fine-Tuning**	**With Fine-Tuning**
✗	●	●	●	FCC 240 GeV (8M)	ILD 250 GeV (800k)	✗	(1)	(2)
✗	●	●	●	FCC 240 GeV (8M)	ILD 250 GeV (800k)	●	(3)	(4)
●	●	●	●	ILD 250 GeV (800k)	ILD 91 GeV (80k)	●	(5)	(6)

- With fine-tuning, the training is obviously accelerated for the initial epochs (even for those with worse eventual performance)
- This is particularly obvious between plots (5) & (6) – similar simulation setup data

Multiple Training Runs

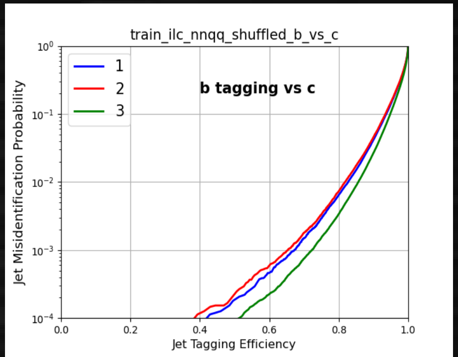


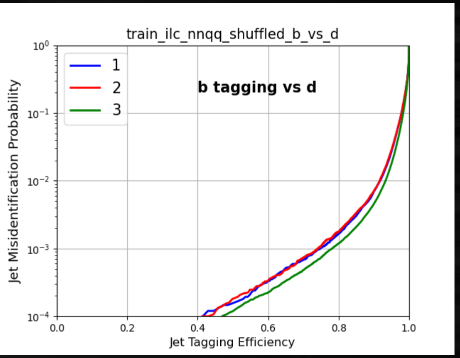
5 times training of FCC_8M data

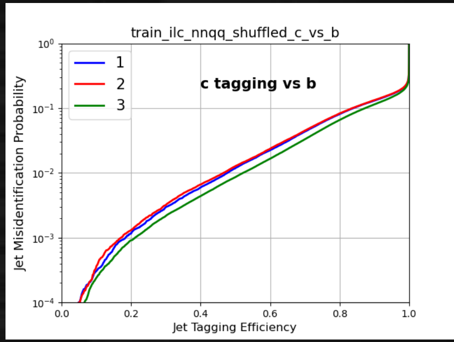
- Multiple training runs don't give significant impacts on results.
- The smaller data size is, the bigger impacts on results multiple runs give.
- The results of no Particle ID trainings varies more than those of with Particle ID.

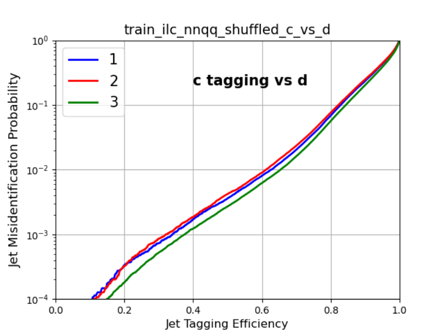
data	Particle ID	b vs c 0.8 Score	variation
FCC 4M	○	4.82e-4	0.43e-4
FCC 8M	○	8.14e-5	1.58e-5
FCC 4M	×	1.69e-3	0.14e-3
FCC 8M	×	7.04e-4	3.49e-4

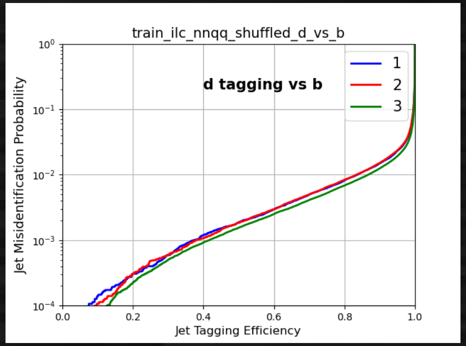
Data Shuffled

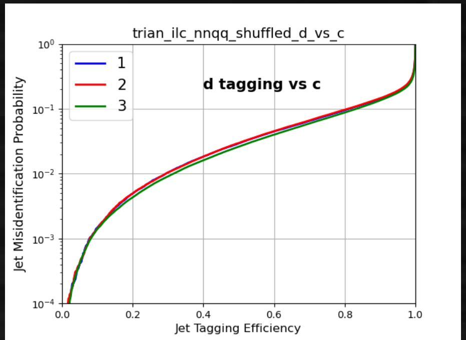












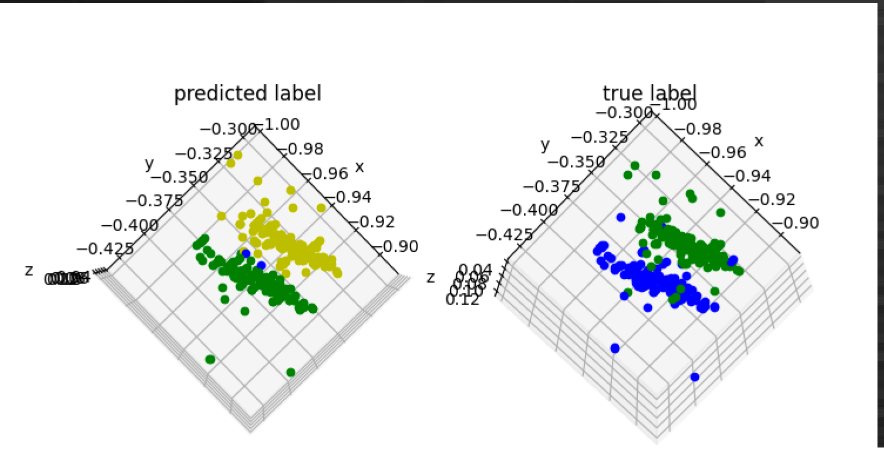
- ILC nnqq dataset
  - 80% training, 5% validation, 15% test
- Shuffled the order of train/test/val making root files
  - Pattern 1: train/val/test
  - Pattern 2: val/train/test
  - Pattern 3: train/test/val

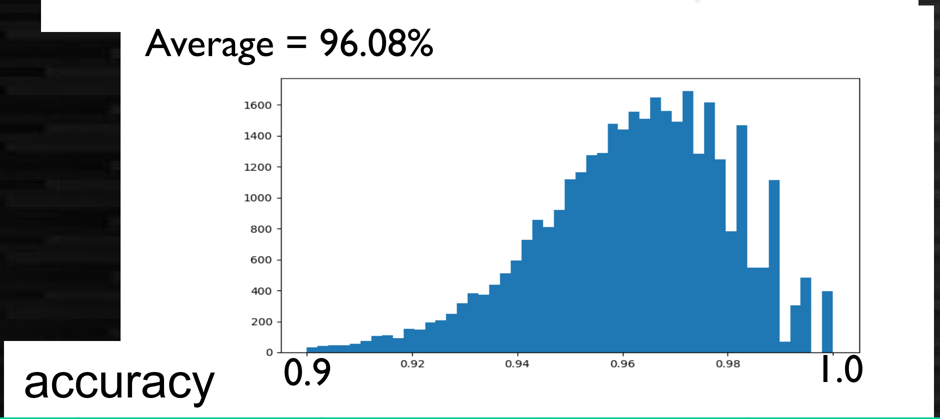
data	b vs c 0.8 score
Shuffle pattern 1	0.00647
Shuffle pattern 2	0.00734
Shuffle pattern 3	0.00338

Importing to ILD full simulation

- Prepare features from ILD full simulation
  - With recent versions (> v02-02)
- Input features: (x, y, z, edep)
- True cluster info
  from MCParticle and LCRelation
- Produced events
  - Two photons
    (5/10 GeV, fixed opening angles)
  - (n x) taus (5/10 GeV)
- Evaluation
  - Fraction of hits associated to the correct cluster (accuracy)

Example of a two-photon event
(5 GeV, 30 mrad)



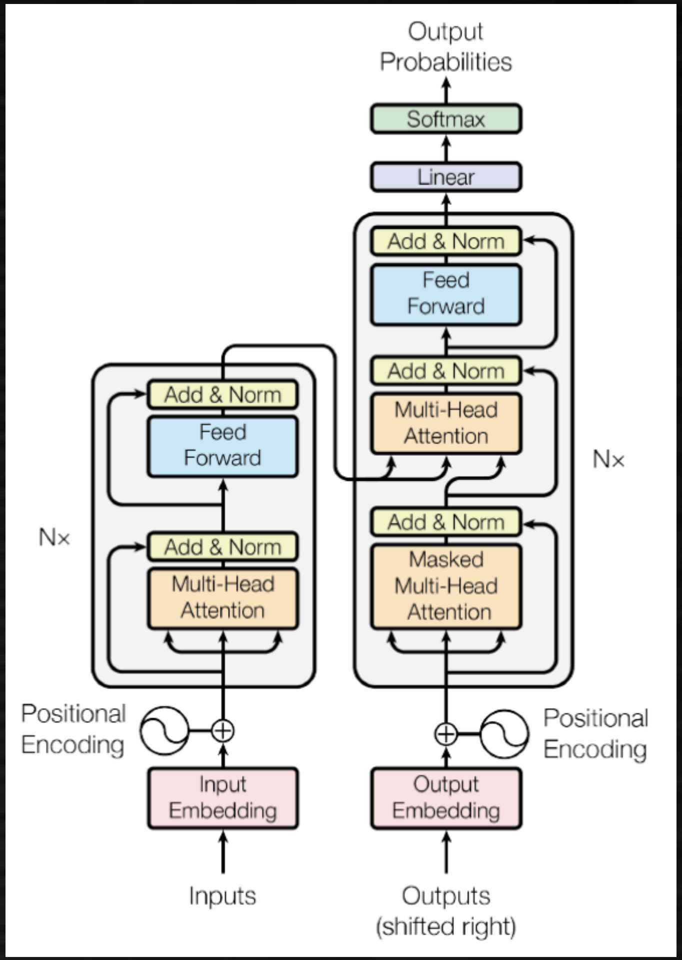


Reasonable
performance seen

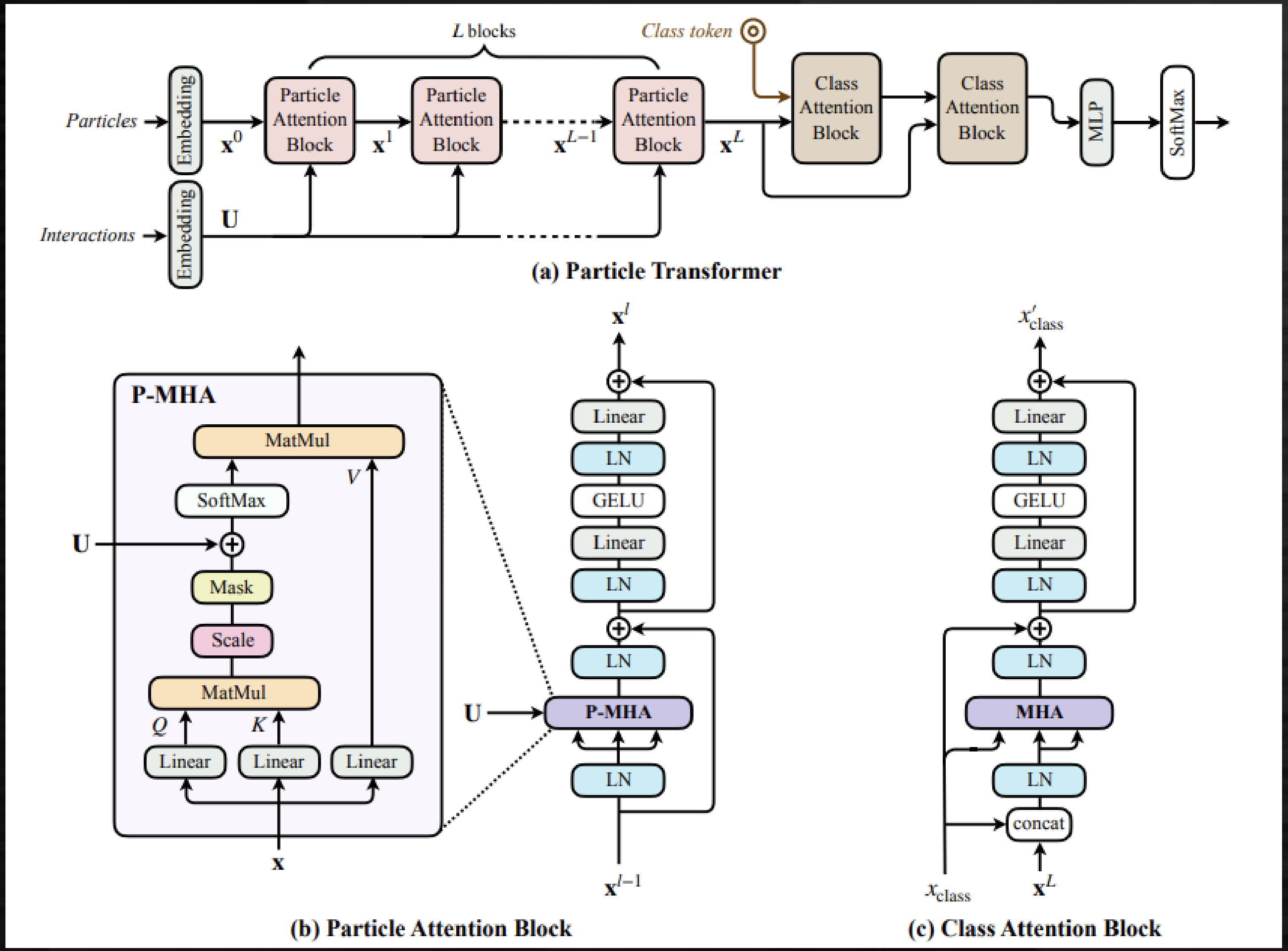
Angle[mrad]	30	60	90	120	150
Accuracy[%]	96.08	98.64	99.30	99.68	99.56

For details, refer eg. https://indico.slac.stanford.edu/event/7467/contributions/5948/attachments/2887/8032/230517-lcws2023-hlreco-suehara.pdf

Comparison between regular Transformer and Particle Transformer



Regular Transformer



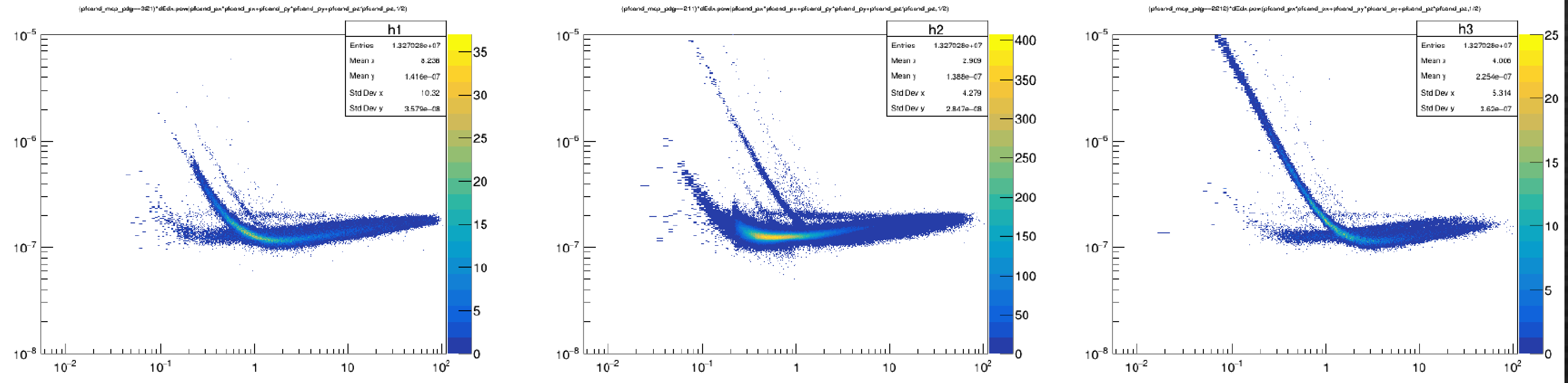
Particle Transformer

Note:
MHA – MultiHeadAttention
P-MHA – Augmented version of MHA by Particle Transformer that involves Interactions Embeddings instead of Positional Embeddings

Progress in strange tag

	s vs c	s vs g	s vs u
0.8 efficiency	0.138	0.288	0.466

Current performance with ParT
(under investigation yet)



Kaon Pion Proton

dE/dx inside strange jets (separated by MC PID)