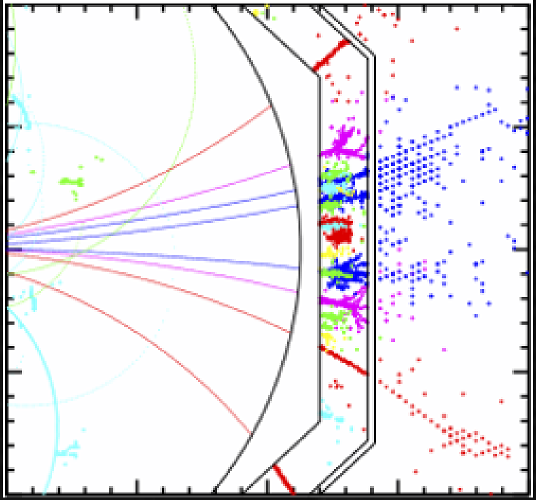
AI-based event reconstruction and detector design for Higgs factories

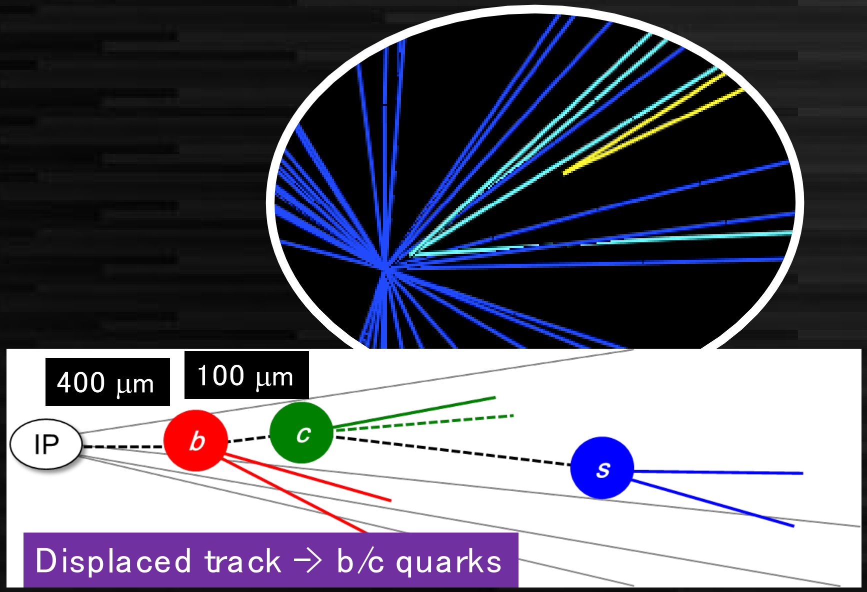
Previous results given in
arXiv:2410.08772 (Proc. ICHEP2024)

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

Today's topics

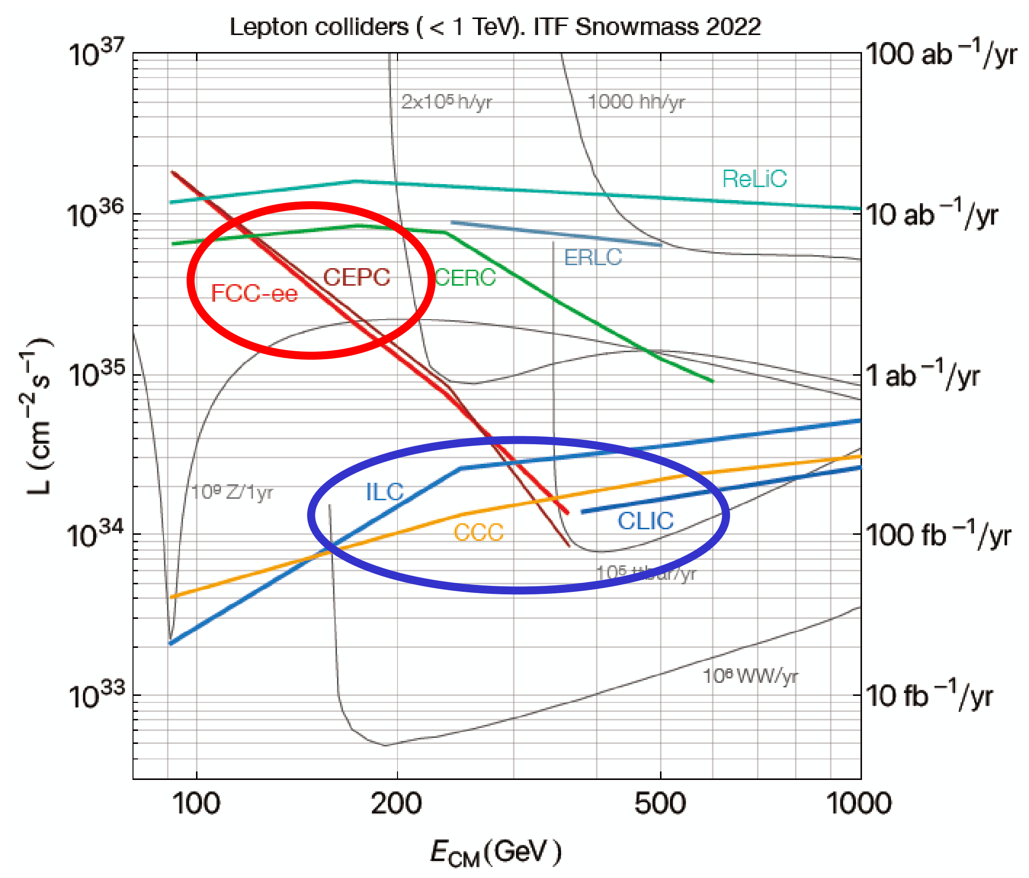
- Detectors and Reconstruction for Higgs factories
- ML topics (status and prospects)
  - Particle flow with GNN / Transformer
  - Flavor tagging with ParT
- Further info/prospects





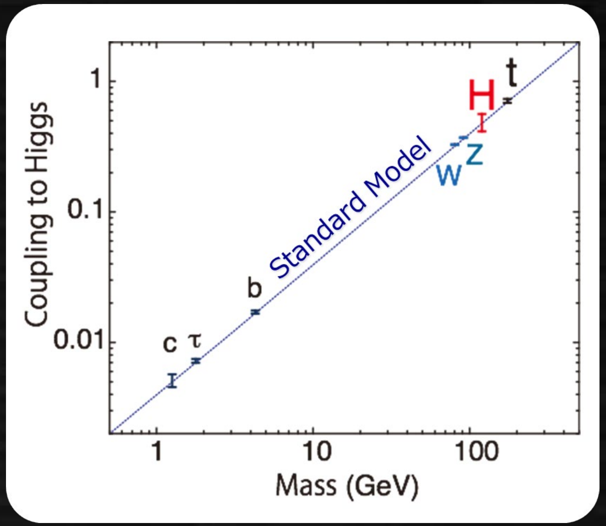
Higgs factories: status

- FCC at CERN: pushed as next CERN flagship
  – 91-365 GeV CM energy (2048-2063?)
- CEPC at China
  – Early realization expected (< 2040?)
- ILC/LCF: linear collider projects
  – 250/550/>1000 GeV operation expected
- LHeC/LEP3: recently proposed alternatives at CERN
  – Realization of Higgs factory with smaller budget (but less statistics/quality)

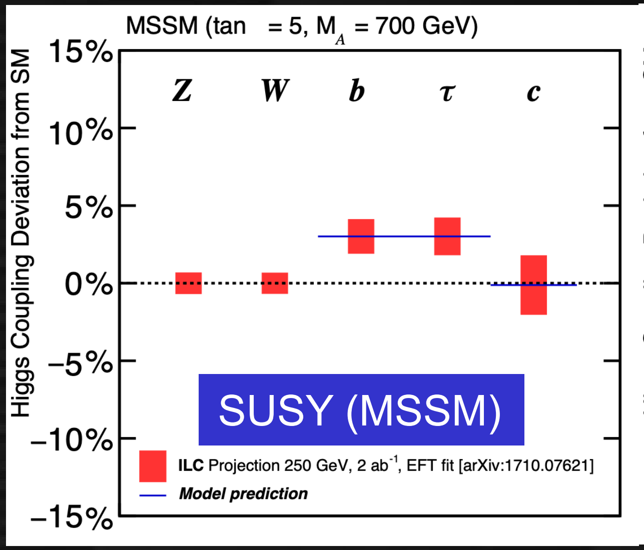


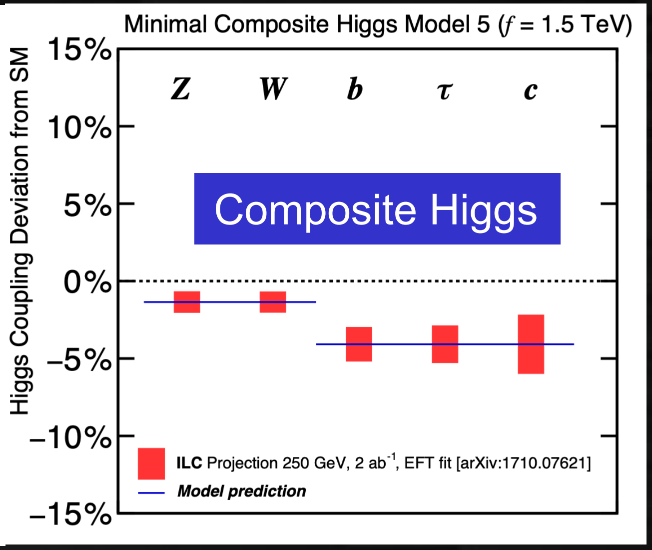
Note: Polarization boost physics at LCs

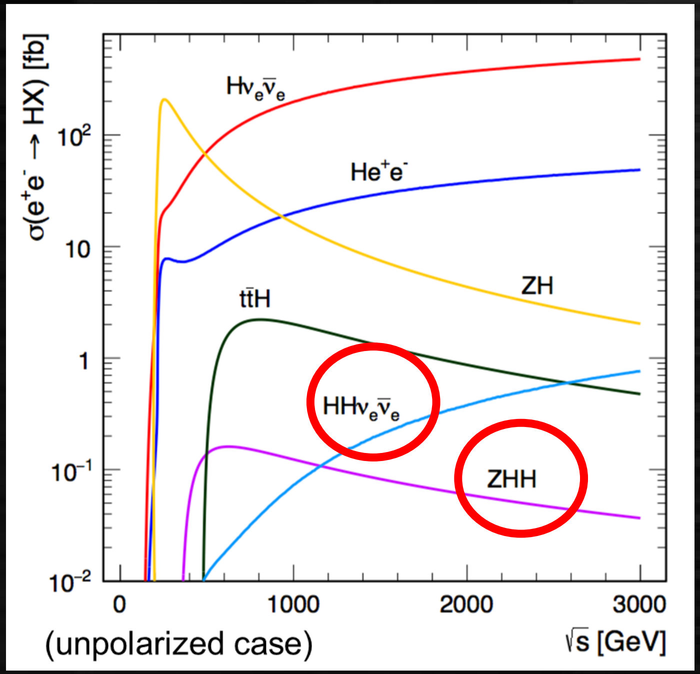
Physics at Higgs factories



Measurement of Higgs couplings: main probe for BSM in ~250 GeV Higgs factories
Higgs self coupling (via ZHH, nnHH) is one of the focus on higher energy e+e- colliders (realized by linear colliders)







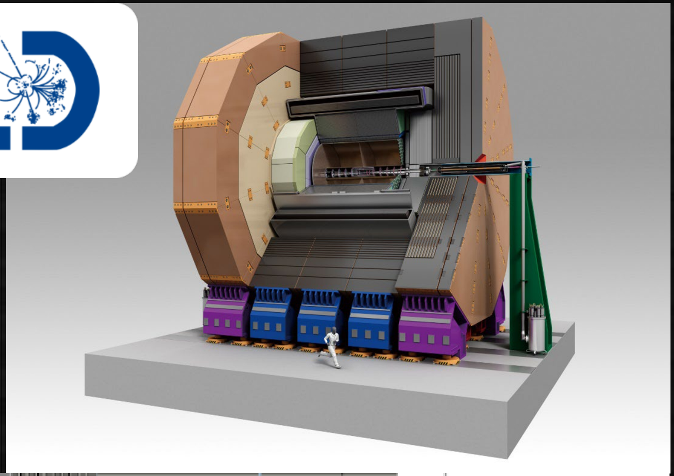
Separation of background critical for HH measurements

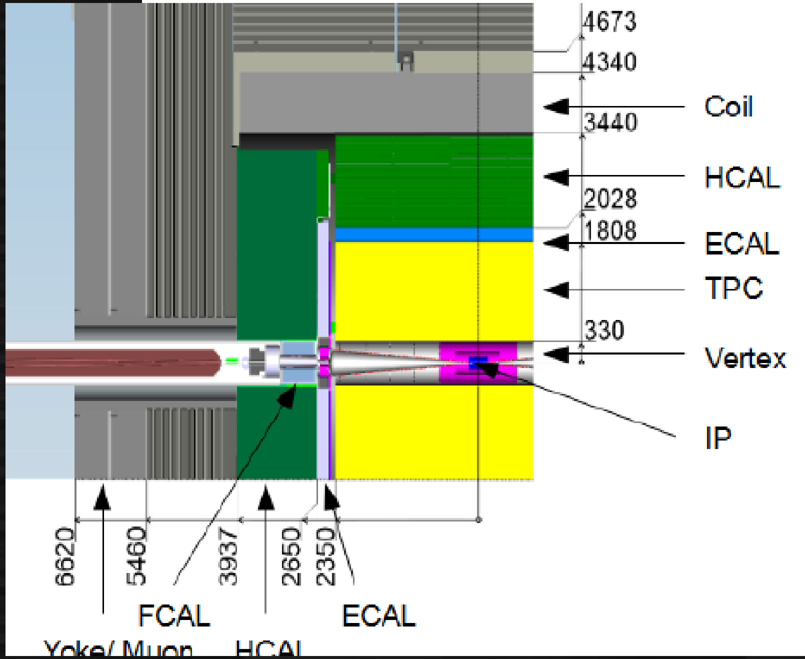
ttbar (a few 100 fb)
ZH, ZZH (1-10 fb)

~a few % Deviation of Higgs couplings expected with many TeV BSM scenarios
Separation of Higgs → b, c, g, s critical for sensitivity

Higgs self coupling: direct probe on Higgs potential
~11% λ_{ZHH} meas. expected at LCF (550 GeV, 8 ab^{-1})

A detector concept for Higgs factories: ILD





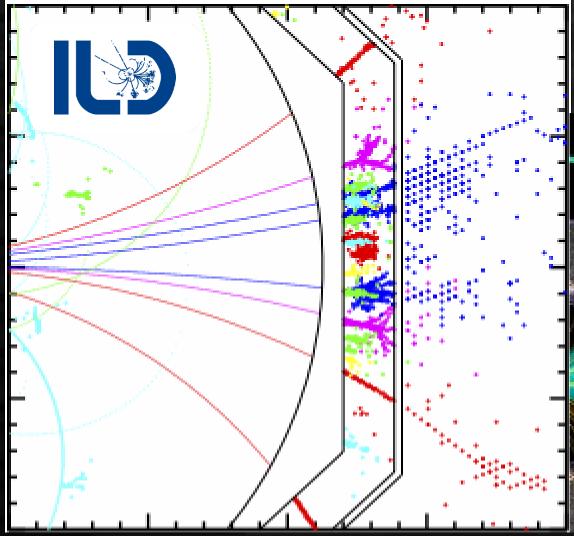
- Key concept: Particle Flow
  - Highly-granular calorimeters
  - Particle inside jets separated 1-by-1
  - Giving 2x better JER (~30%/sqrt(E [GeV]))
  - Optional ToF at calorimeter (~100 psec/hit)
- Tracker: silicon + TPC combined
  - Vertex: a few μm resolution at r ~ 15 mm
    - Significant impact on c-tagging (wrt. LHC)
  - TPC: good for dE/dx (discussed later)
    - Important for strange tagging
- Magnet (3.5T) outside HCAL
  - Minimal material before calorimeters

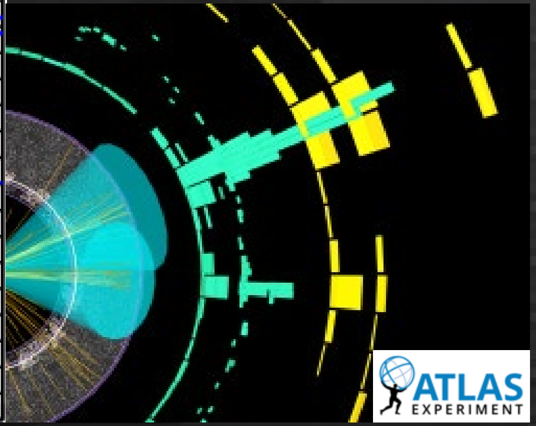
Particle flow concept

Separating particles inside jets to do track-cluster matching

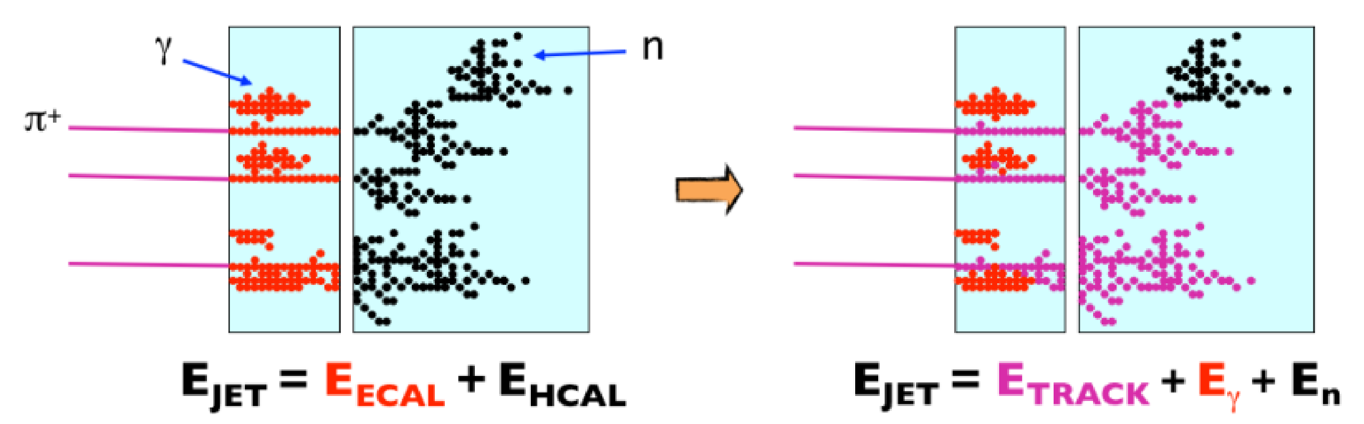
Requiring
- Highly-granular calorimeters
- Intelligent pattern recognition

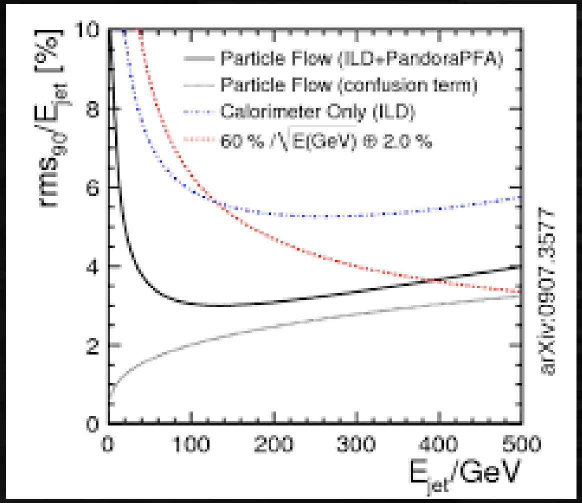
Different granularity on ILD - ATLAS





Developed in ILC, first full application in CMS HGCAL at HL-LHC (partial use already in ATLAS/CMS)





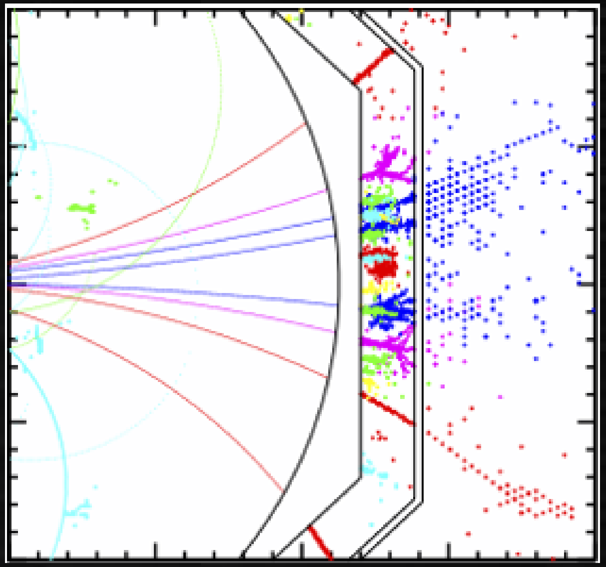
Possible to obtain jet energy resolution of

$$\frac{\delta E_{\mathrm{jet}}}{E_{\mathrm{jet}}} \cong \frac{30\%}{\sqrt{E_{\mathrm{jet}}[\mathrm{GeV}]}}$$

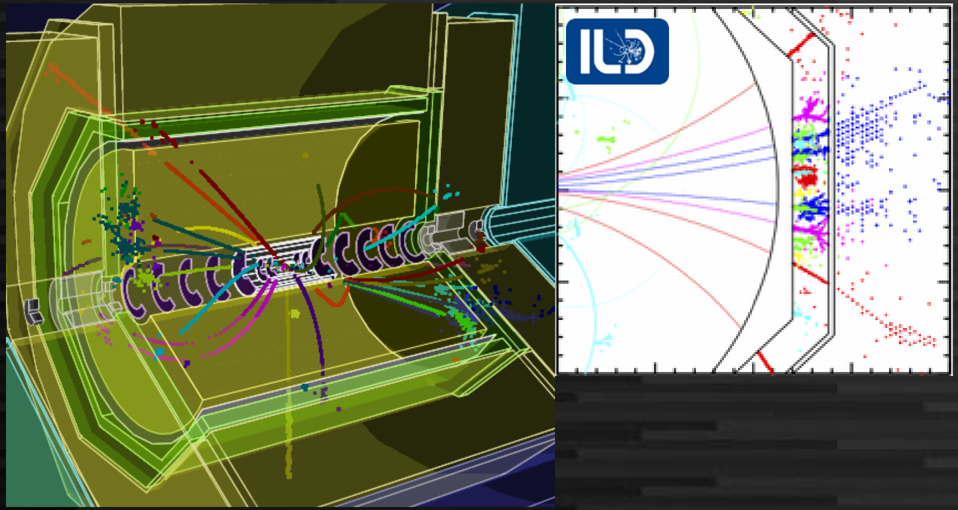
~2 times better than calo-only

Particle flow with DNN

Today's main topic

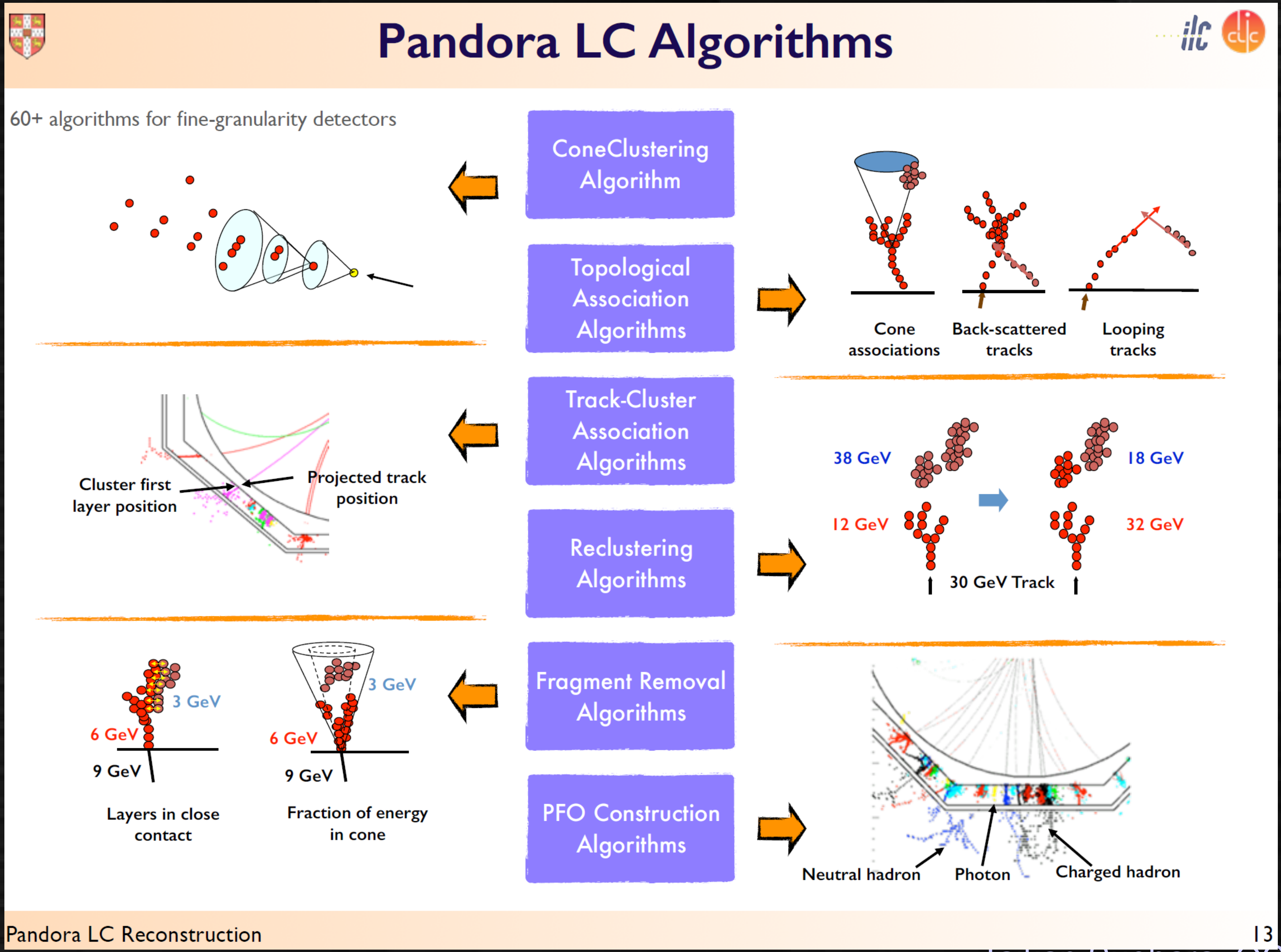


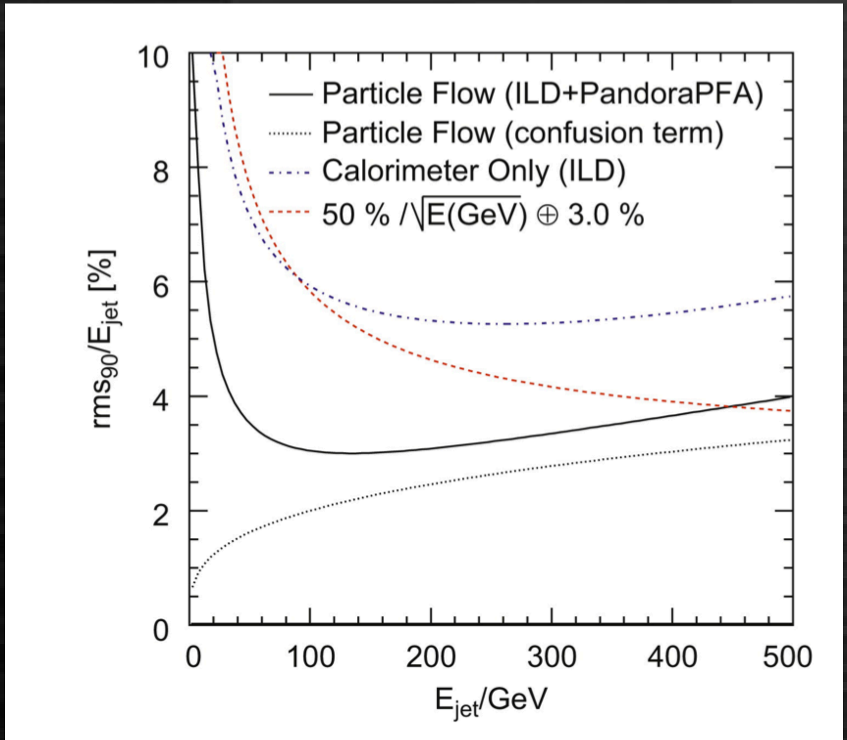
Particle flow for Higgs factories



- 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



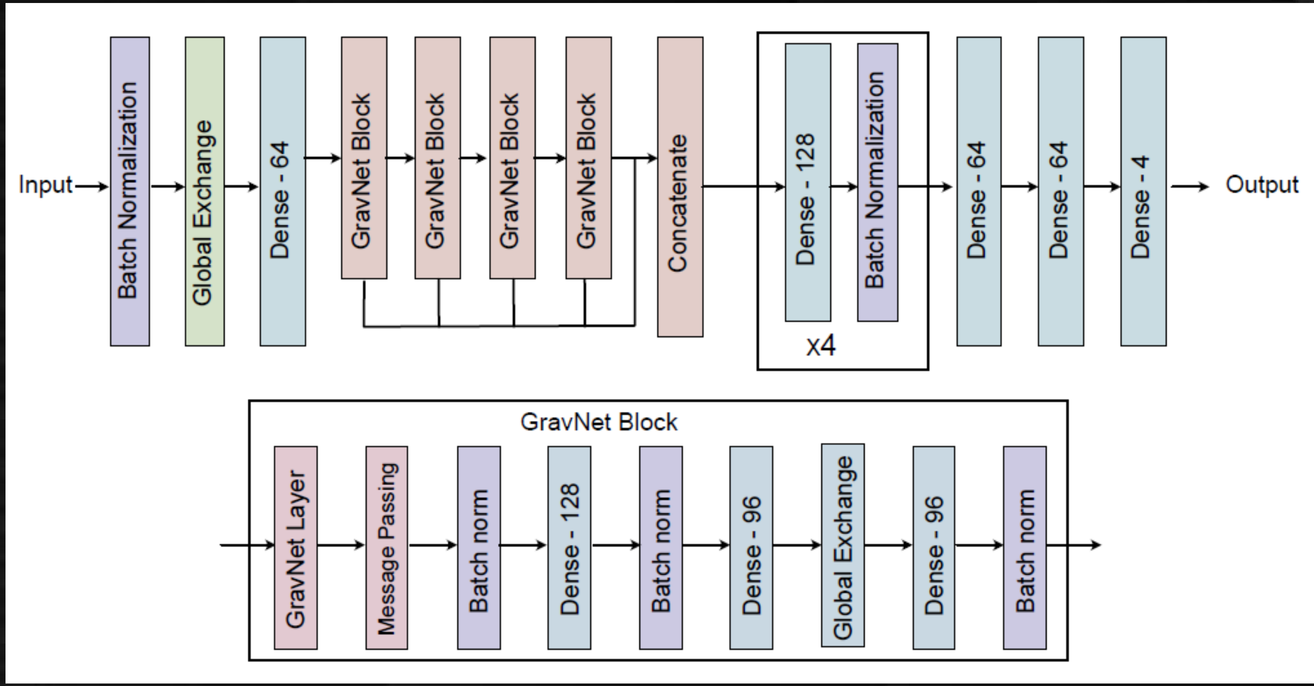


Widely used since 2008
Reasonably good performance up to ~50 GeV jets
Confusion dominates at higher energies

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, …)

The network



Initially developed as CMS HGCAL clustering algorithm (without tracks)

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)

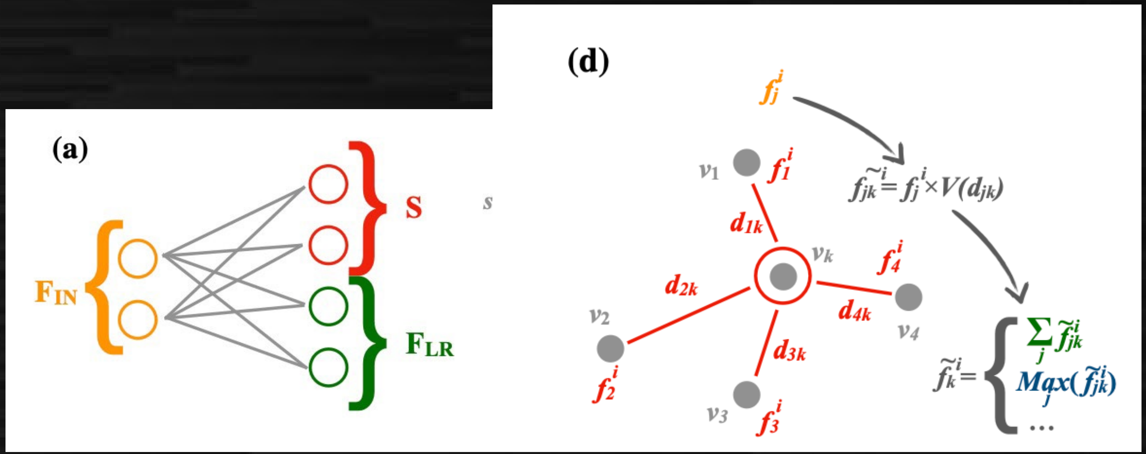
Output: “condensation coefficient” β, position at virtual coordinate (2-dim)
optional output of features such as energy, PID (not used now)

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

GravNet and Object Condensation

GravNet arXiv:1902.07987

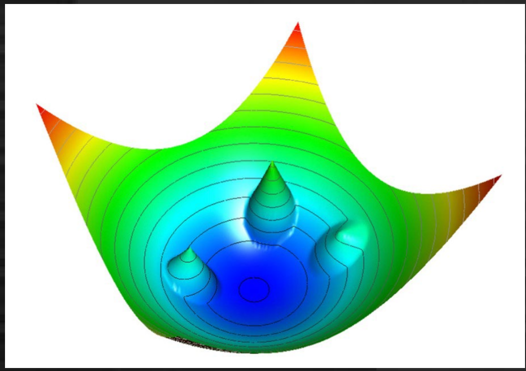
- The virtual coordinate (S) is derived from input variables with simple MLP
- Convolution using "distance" at S (bigger convolution with nearer hits)
- Repeat 2 times and concatenate the output with simple MLP



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.)

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

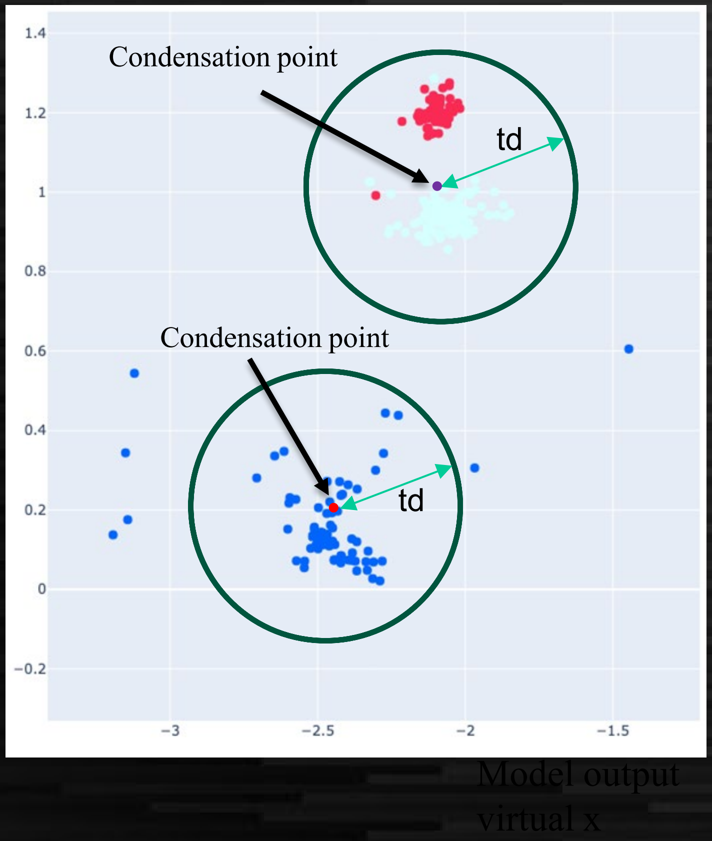
L_β: Pulling up β of the condensation points / tracks

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
- List all condensation points with β > tbeta
- Associate hits to condensation points if they are within a distance (td) from the condensation point at the output coordinate
  - If hits can be associated to multiple condensation points, the nearest one is taken
- Take the highest β point from the remaining hits, and cluster neighbor hits as similar to the previous step
- td/tbeta are tunable parameters



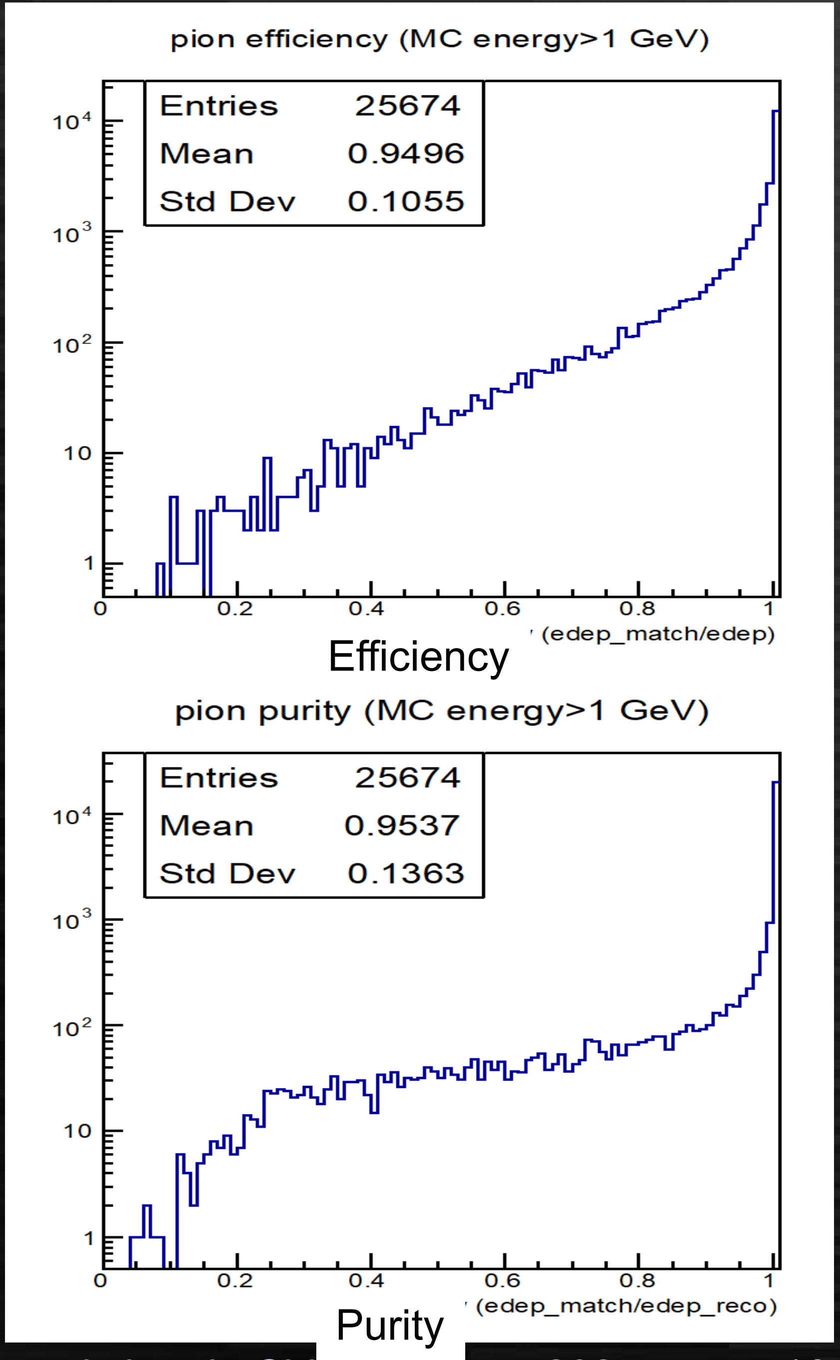
Our samples for performance evaluation

- 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)
  - ZH → $\nu\nu$qq (q=u/d) sample at 250 GeV
    - For energy regression

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

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 clusters may connect to one MC cluster → encourage splitting too much
- 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)



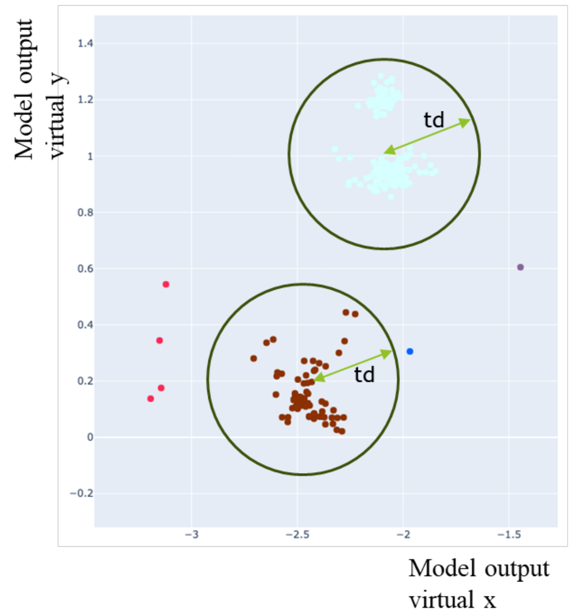
Optimization of performance

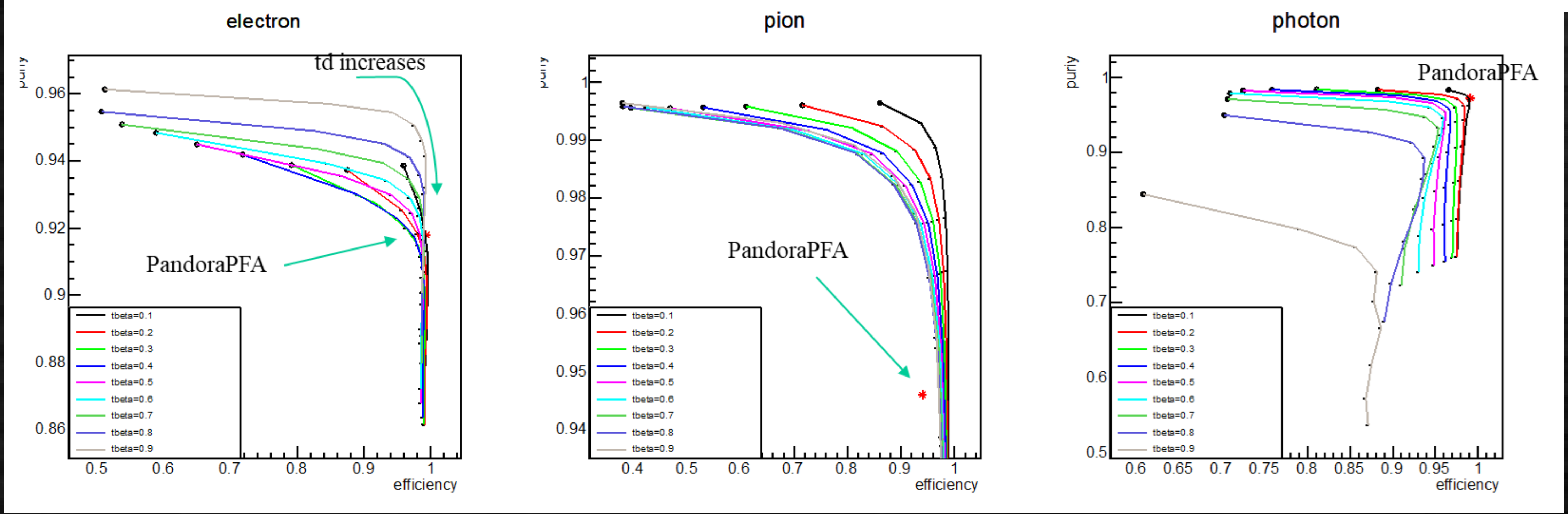
Output dimension of the coordinate

- 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)

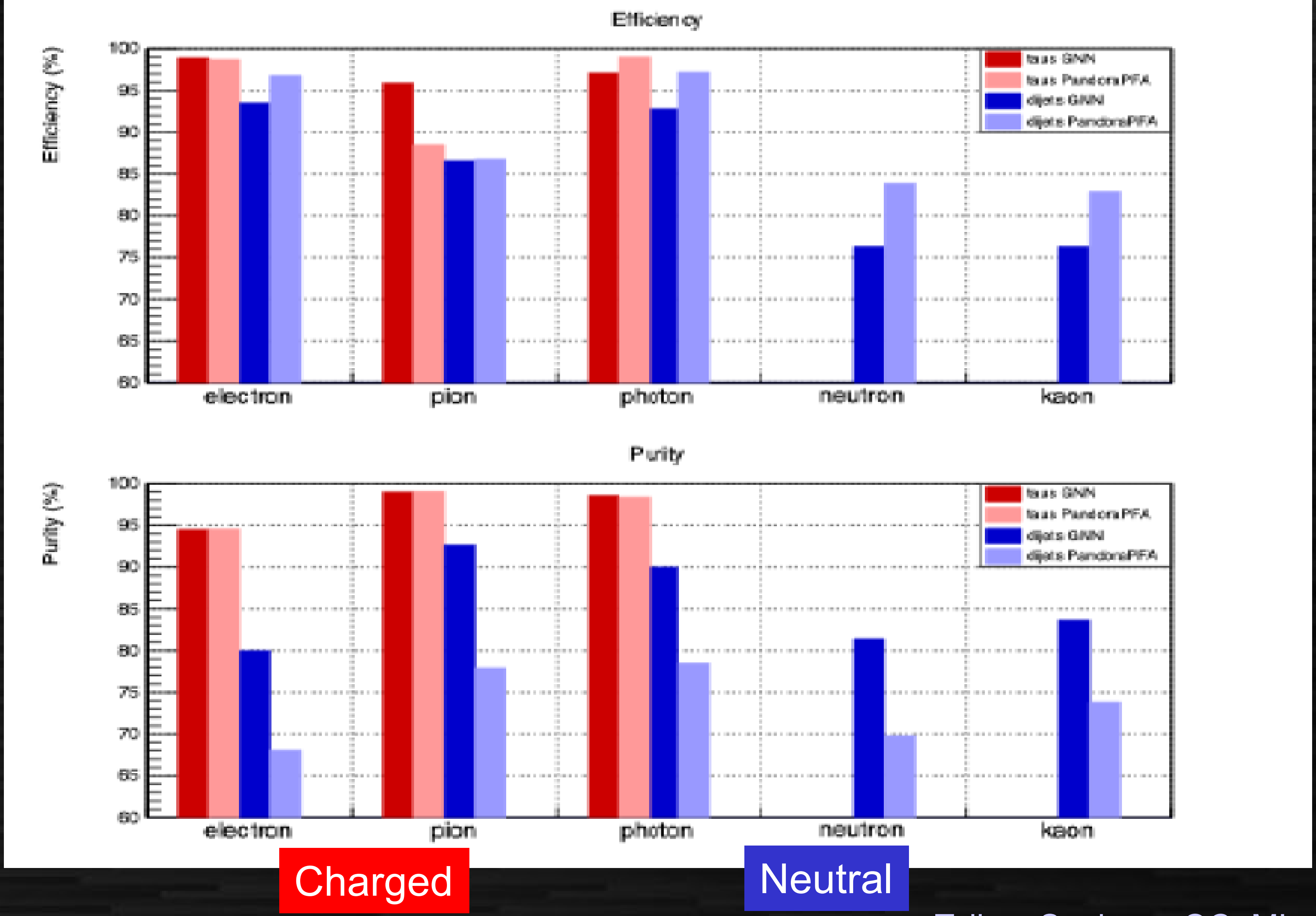
- td: radius which hits are treated as coming from the same cluster
- tbeta: threshold of beta to form clusters





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

Efficiency and purity: comparison with Pandora



Red: 10 taus
Blue: di-jets
Thick color: GNN-based PFA
Thin color: PandoraPFA

Pion: GNN > Pandora

Electron, photon, neutron, kaon:
Efficiency: GNN < Pandora
Purity: GNN > Pandora

Overall:
competitive performance achieved
Pion reconstruction is especially important in jet reconstruction
→ good expectation

Energy regression: ongoing work

Add E_{tr} and E_{hit} to the output of the network (for each hit)
Add terms (1, 2) to object condensation loss

Cluster energy (MC vs reco) at 10 taus event

truth clustering

real clustering

Two additional loss term

1. E_{tr} at condensation points to be regressed to MC cluster energy

$$L_{E,charged} = \frac{1}{2}\sum_i \left(E_{truth,i} - E_{pred,cond,i}\right)^2$$

2. Sum of E_{hit} of all energies to be regressed to MC cluster energy

$$L_{E,nerutral} = \frac{1}{2}\sum_i \left(E_{truth,i} - \sum_j E_{pred,calo,i,j}\right)^2$$

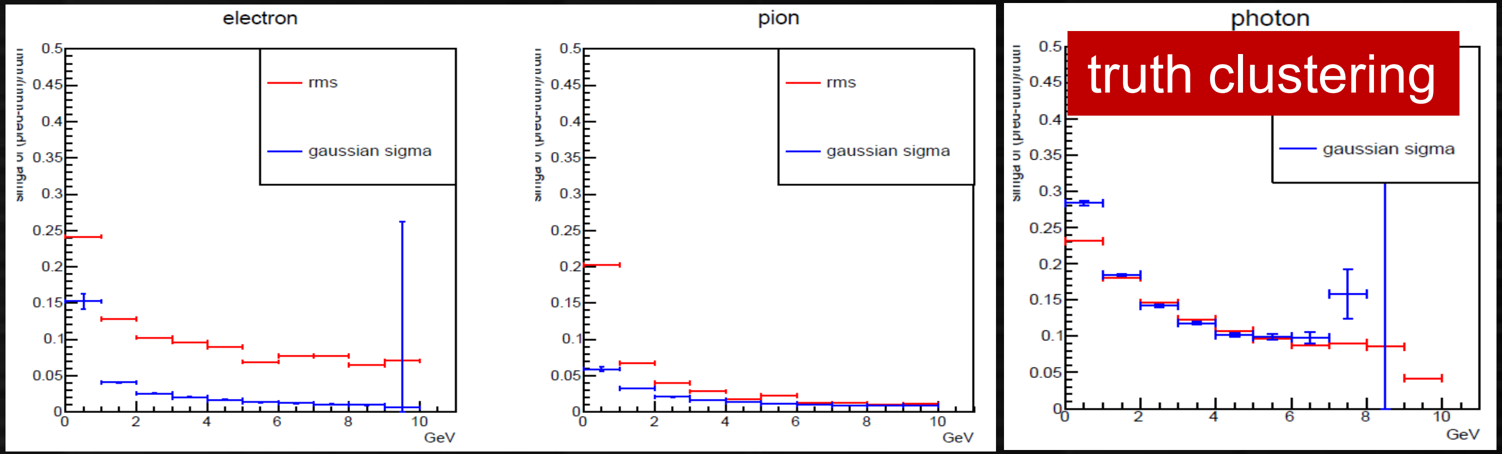
3. Use E_{tr} for charged clusters and use sum of E_{hit} for neutral clusters

Energy regression: ongoing work

Add E_{tr} and E_{hit} to the output of the network (for each hit)

Add terms (1, 2) to object condensation loss

Cluster energy (MC vs reco) at 10 taus event



truth clustering

real clustering

Two additional loss term

1. E_{tr} at condensation points to be regressed to MC cluster energy

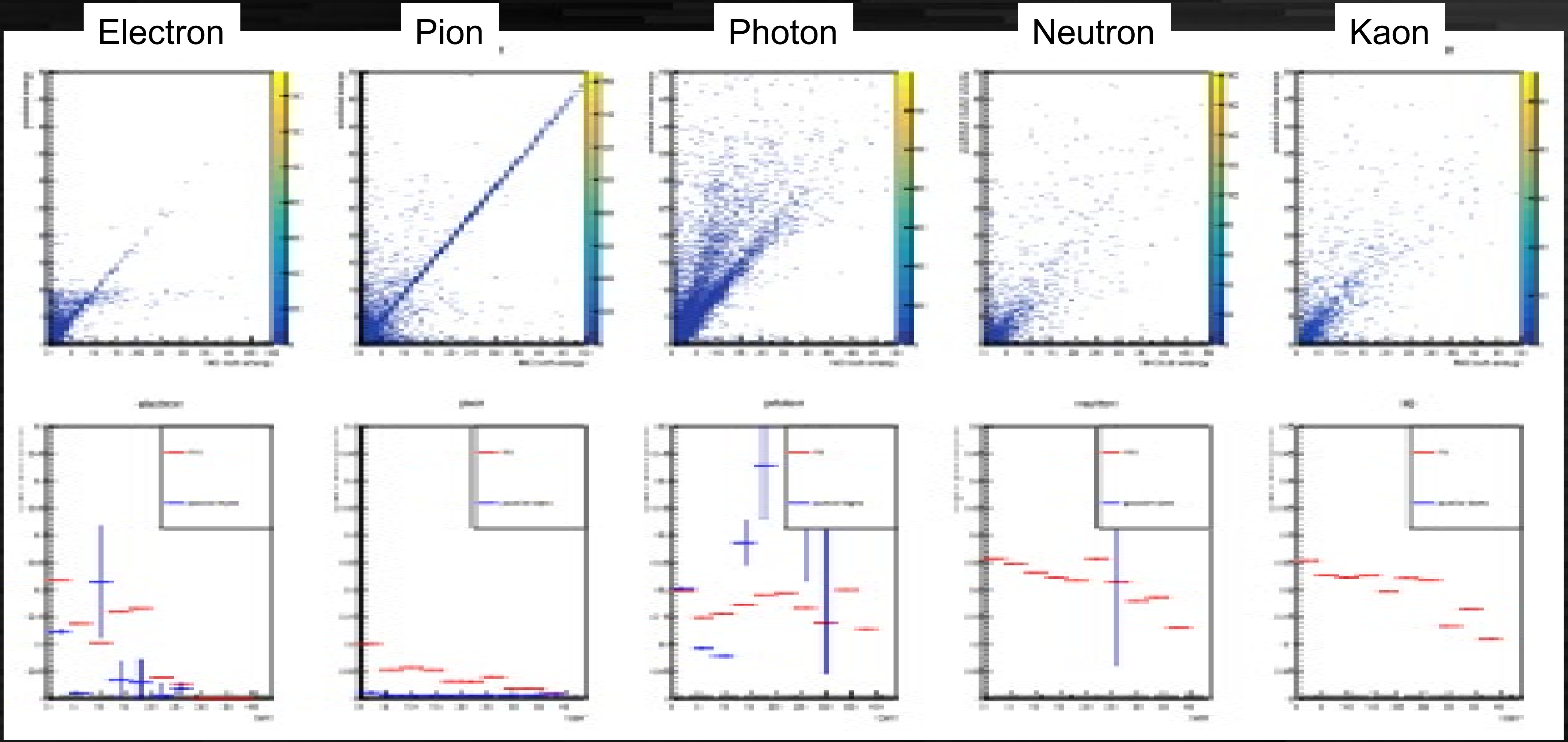
$$L_{E,charged} = \frac{1}{2}\sum_i \left(E_{truth,i} - E_{pred,cond,i}\right)^2$$

2. Sum of E_{hit} of all energies to be regressed to MC cluster energy

$$L_{E,nerutral} = \frac{1}{2}\sum_i \left(E_{truth,i} - \sum_j E_{pred,calo,i,j}\right)^2$$

3. Use E_{tr} for charged clusters and use sum of E_{hit} for neutral clusters

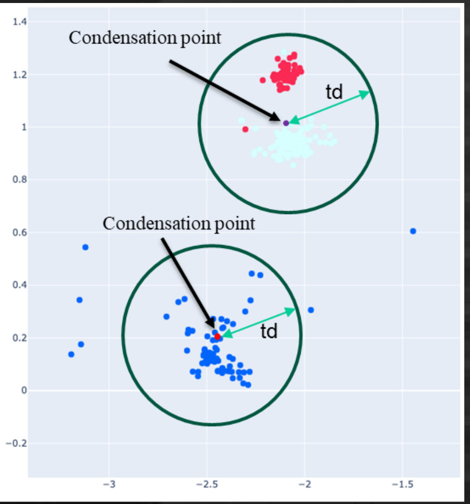
Energy regression: di-jet sample

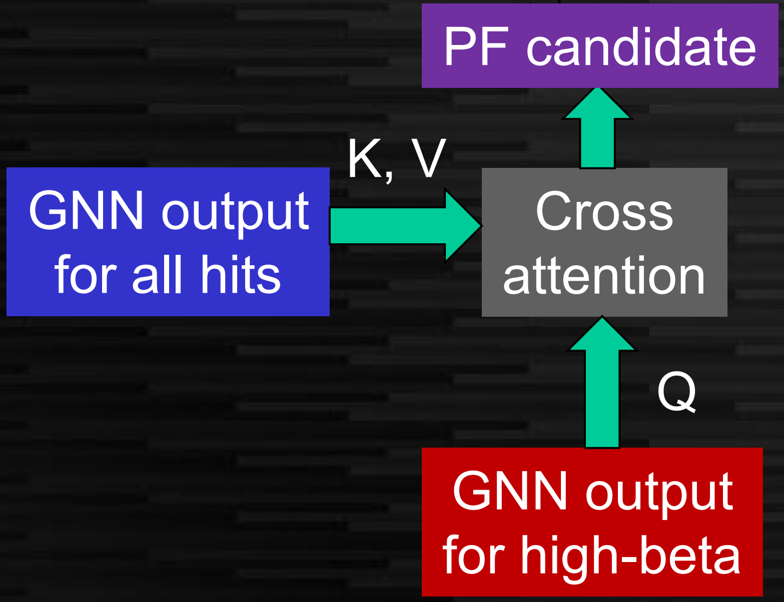


Performance not very good: possible reason is the difference between training loss (with truth clustering) and evaluation criteria (with realistic clustering)

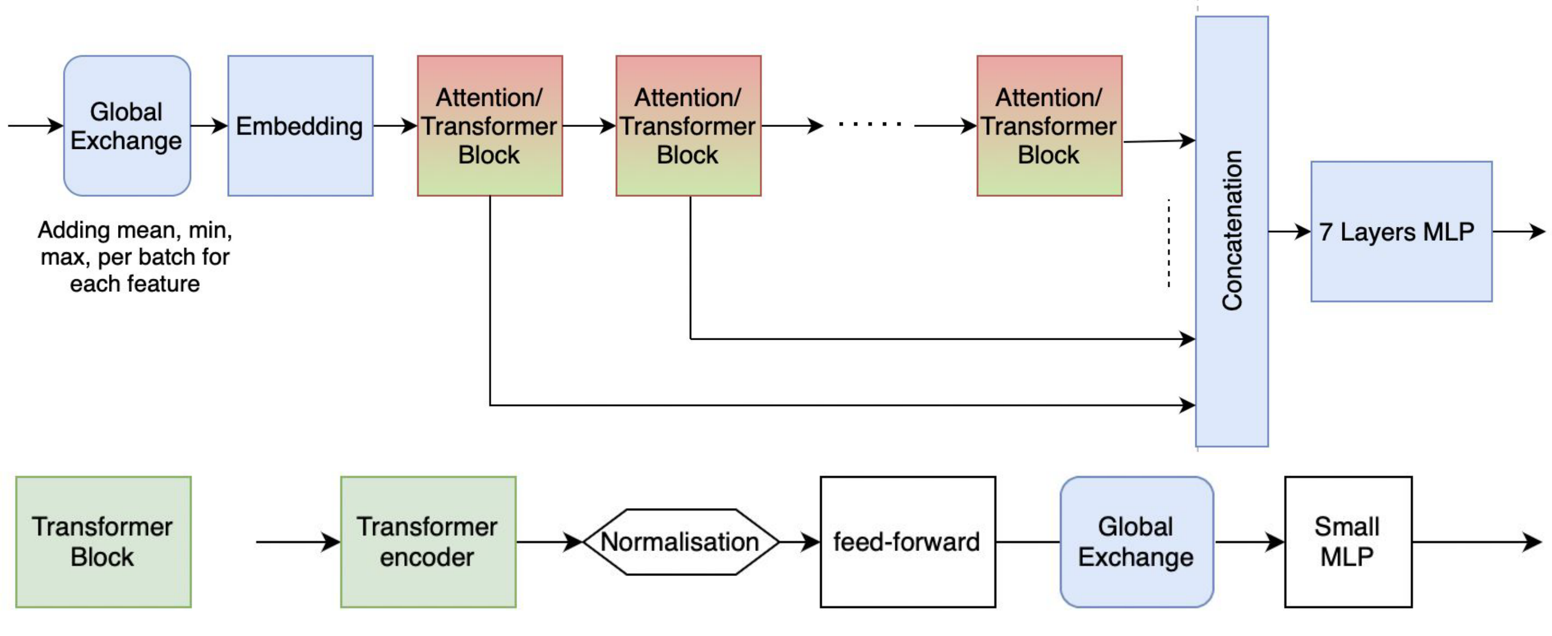
Prospects: DNN-based clustering

- Current issue: clustering not intelligent
  - Simply gathering hits around cond-point
  - Not based on ML – issue on energy regression
- Implementing ML-based clustering
  - Use high-beta points as “query” of transformer
    → particle candidate (pfcand)
  - Cross attention of hits to pfcand
  - Derive particle properties (or tagged as fake)
    - Attention weights used for hit-particle mapping
  - Eventually unified to single network
- Under investigation





Replacing GravNet with Transformer



Work by internship student (S. Barbu)
Summarized in BOOST2025 poster
https://indico.physics.brown.edu/event/18/contributions/396/

Using similar structure to GravNet but replace GravNet block with transformer encoder block

Use the same loss (object condensation)

Metric	Tr-model	GSA-GravNet	GravNet	Improvement (Tr-model-GravNet)
Number of parameters (M)	2	0.9	0.4	
Electron Efficiency/Purity	99.4 / 91.3	98.2 / 91.5	98.9 / 94.5	+0.5 / -3.2
Pion Efficiency/Purity	98.0 / 98.5	95.7 / 98.4	95.9 / 99.0	+2.1 / -0.5
Photon Efficiency/Purity	97.2 / 97.1	93.3 / 97.1	97.1 / 98.6	+0.1 / -1.5

Performance comparison with 10 taus

Some improvement seen in pion efficiency – details to be checked

Yet another prospects: treating real data

- Real calorimeter suffers from inhomogeneous response
  - Channel-by-channel gain difference (to calibrate/correct)
  - Dead channels, noisy channels
- Can ML be used for correction from MC to data?
  - "FiLM" technique – additional (small) ML to derive linear (ax+b) correction (FiLM calculates a and b to each hit)
  - Train initial network with simulation and train additional FiLM layer later by real data
  - Under investigation (to try calibration of test beam data)

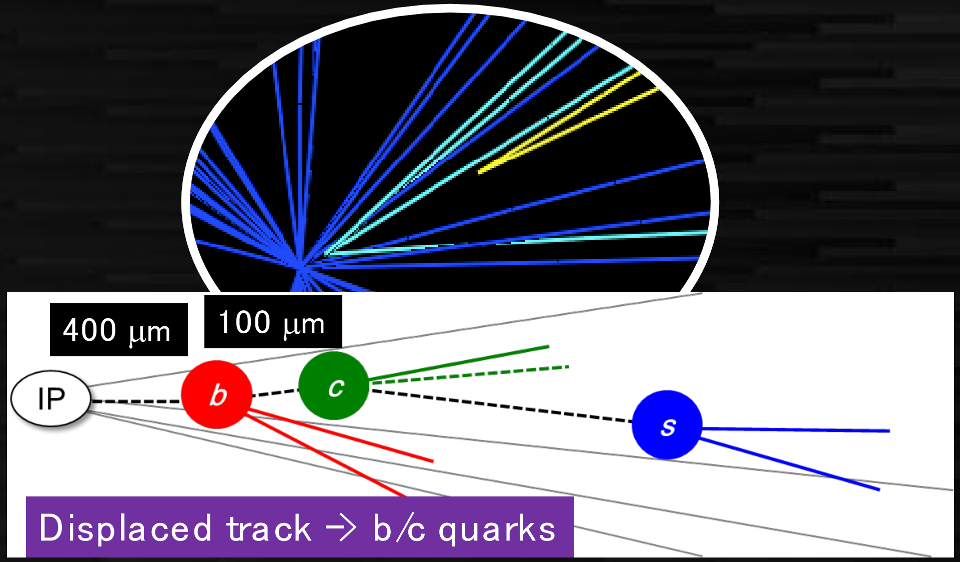
→ Towards "modeling" of imperfect detector response

Hits → Particles → Jets → Physics

- Hits → Particles: Particle flow
  - Actively developing…
  - Particle ID after PFlow: 1-by-1 correspondence (cf. Manqi's talk)
    - Starting collaboration
- Particle → Jets: Jet clustering
  - ML-based implementation not used yet
  - Transformer-based clustering can be used? Same as Pflow?
  - Jet flavor tagging: the most established ML application
- Particle/Jets → Physics
  - Better to think of using particles as well as jets
  - Doing some study on jet pairing from particle + jets
  - Particle (or even hits) → Physics: one-pass "full analysis" ML can be a central brain??

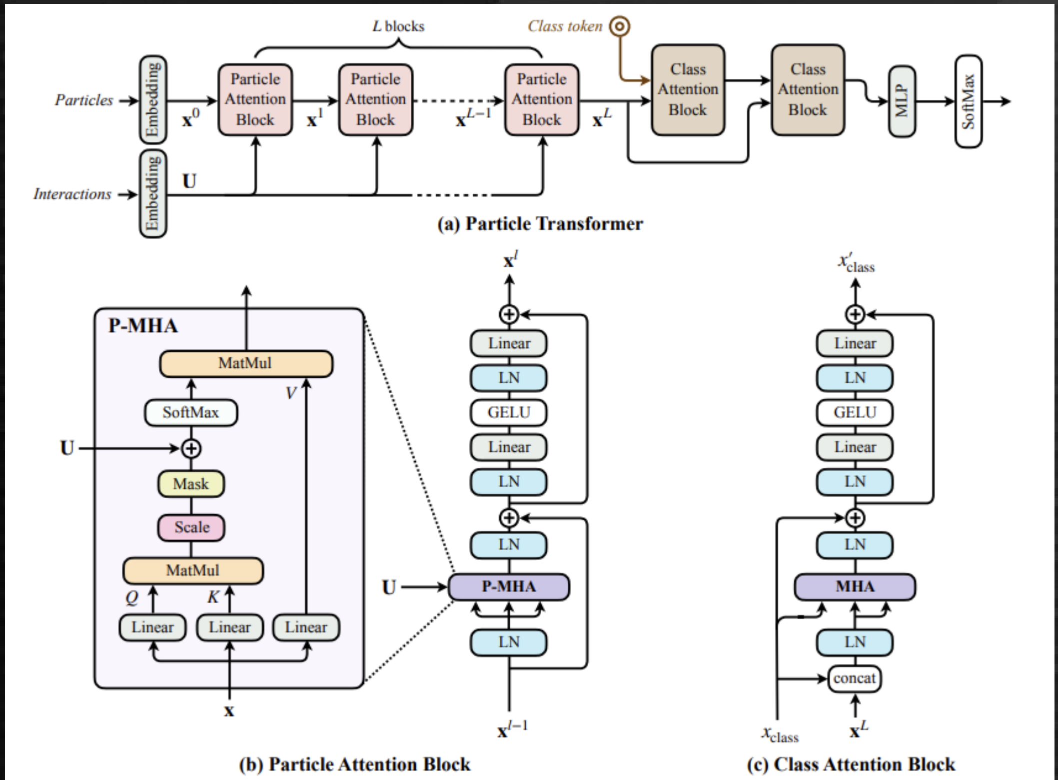
Flavor tagging with Particle Transformer (ParT)

Just a brief summary (due to time constraint)
Full presentation in BOOST2025
https://indico.physics.brown.edu/event/18/contributions/418/



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

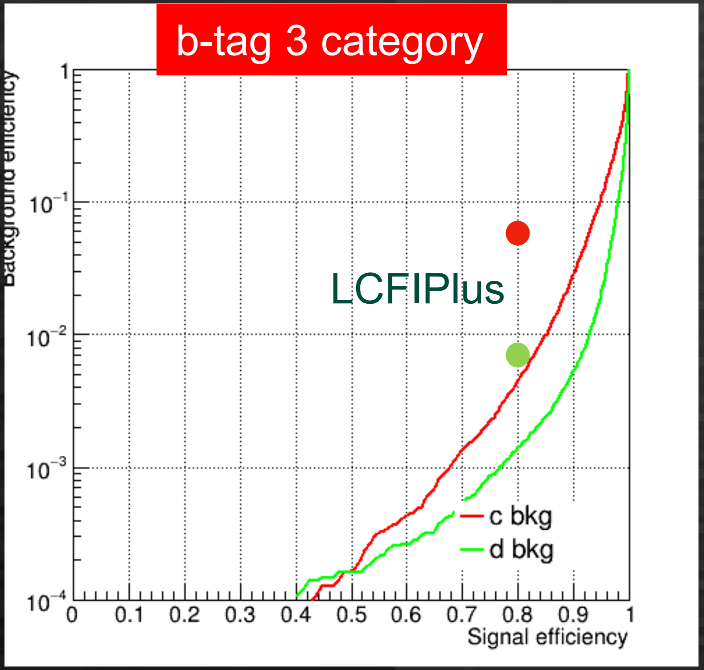


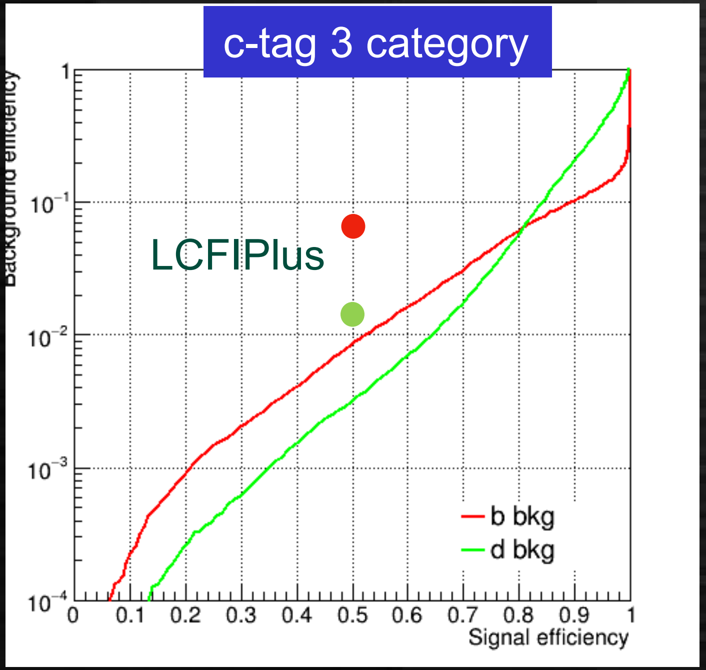
	All classes	
	Accuracy	AUC
PFN	0.772	0.9714
P-CNN	0.809	0.9789
ParticleNet	0.844	0.9849
ParT	**0.861**	**0.9877**

Performance with JetClass event classification (100M sample)

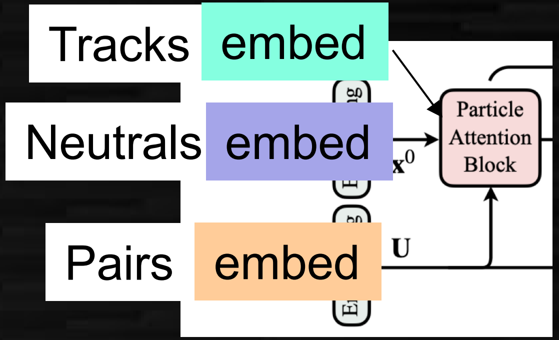
Improvements wrt. LCFIPlus

- 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



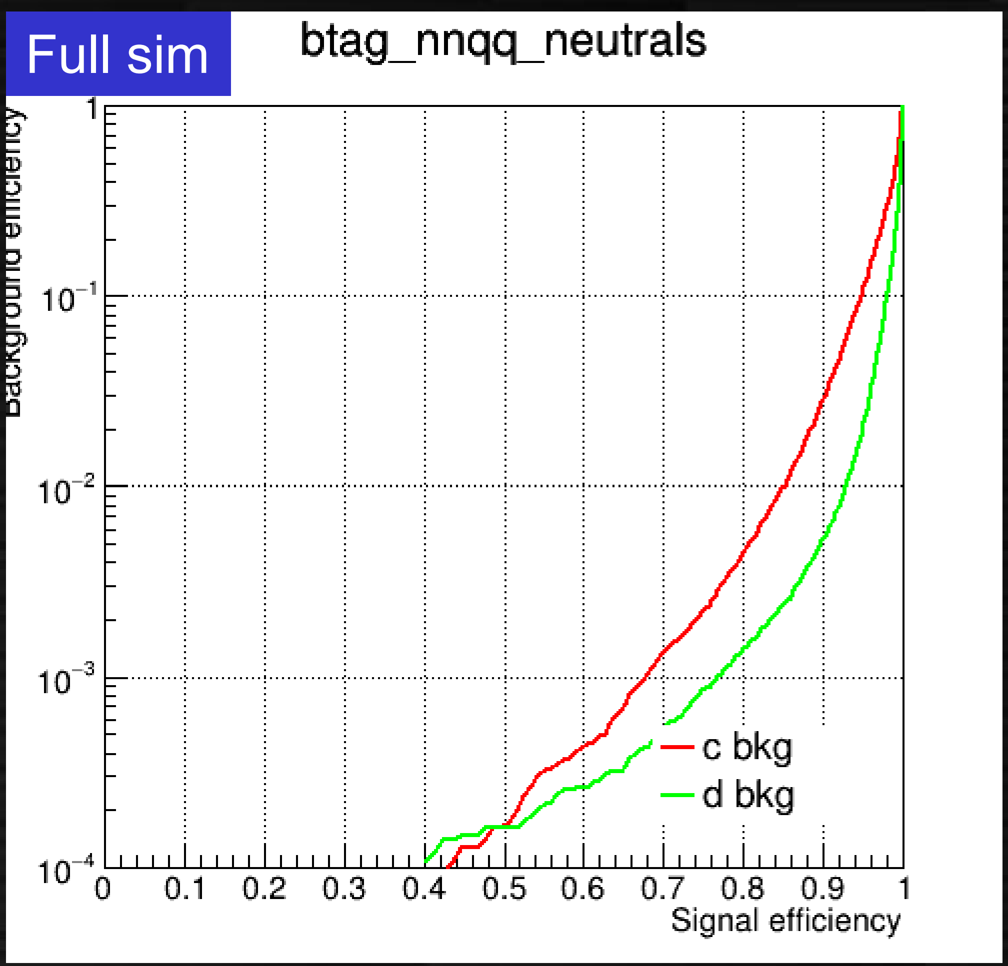


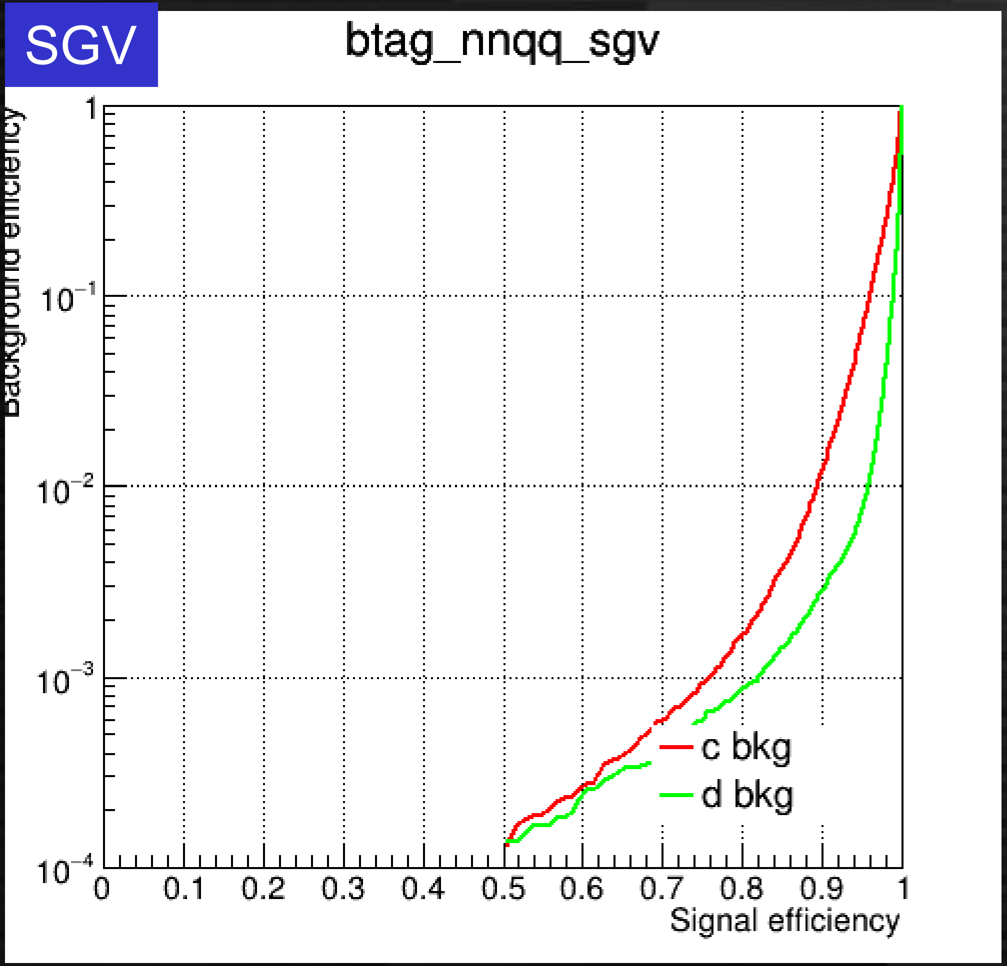
	b-tag 80% eff.		c-tag 50% eff.	
background	c jets	uds jets	b jets	uds jets
+LCFIPlus (BDT)	6.3%	0.79%	7.4%	1.2%
*ParT (initial)	1.3%	0.25%	1.0%	0.43%
**ParT (improved)	0.48%	0.14%	0.86%	0.34%



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

Comparison with ILC fastsim (SGV) vs full sim





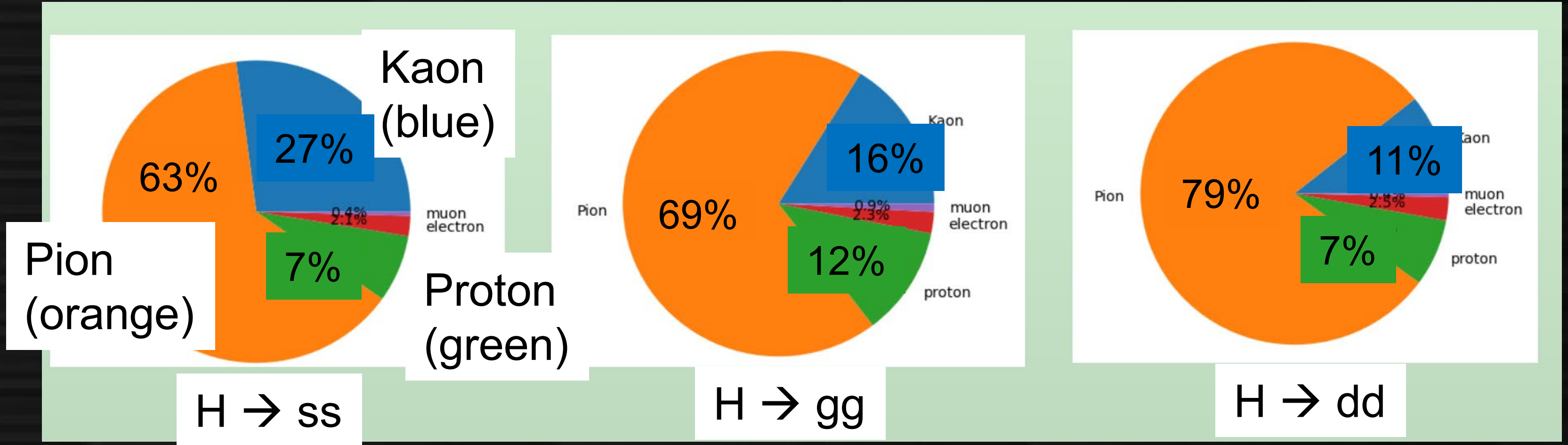
SGV gives better performance (the sample size is same)

Maybe related to difference of neutral particles (confusion may not be included)

Comparison on scaling raw to be checked

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

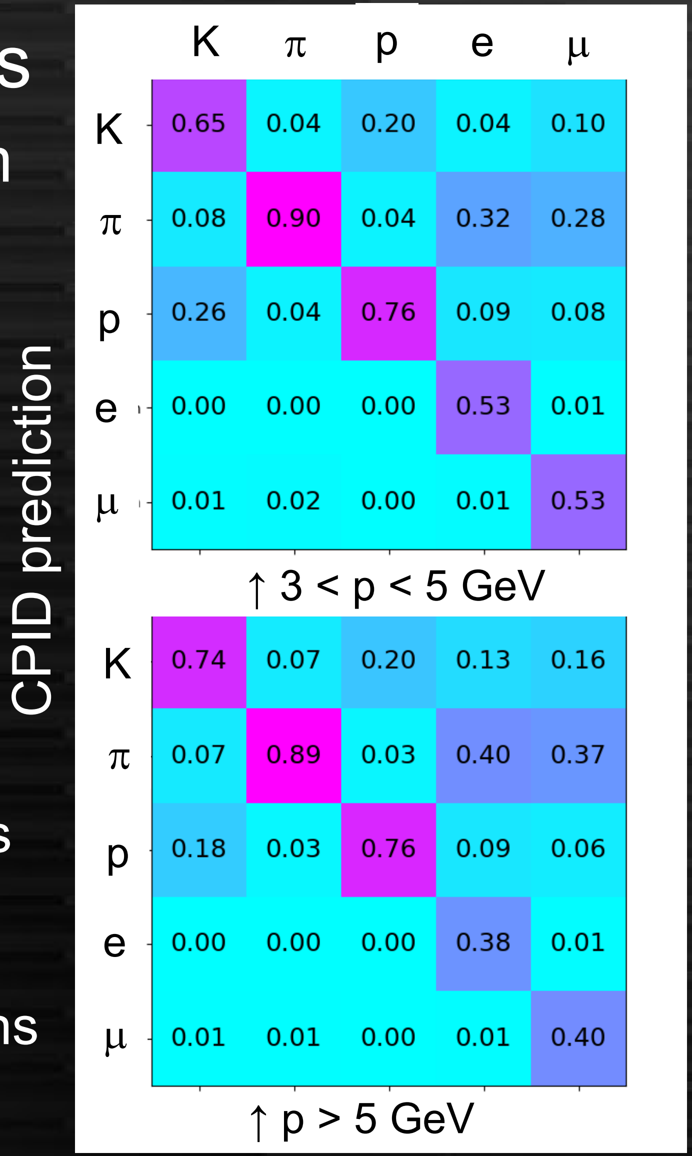


Fractions of tracks having > 5 GeV

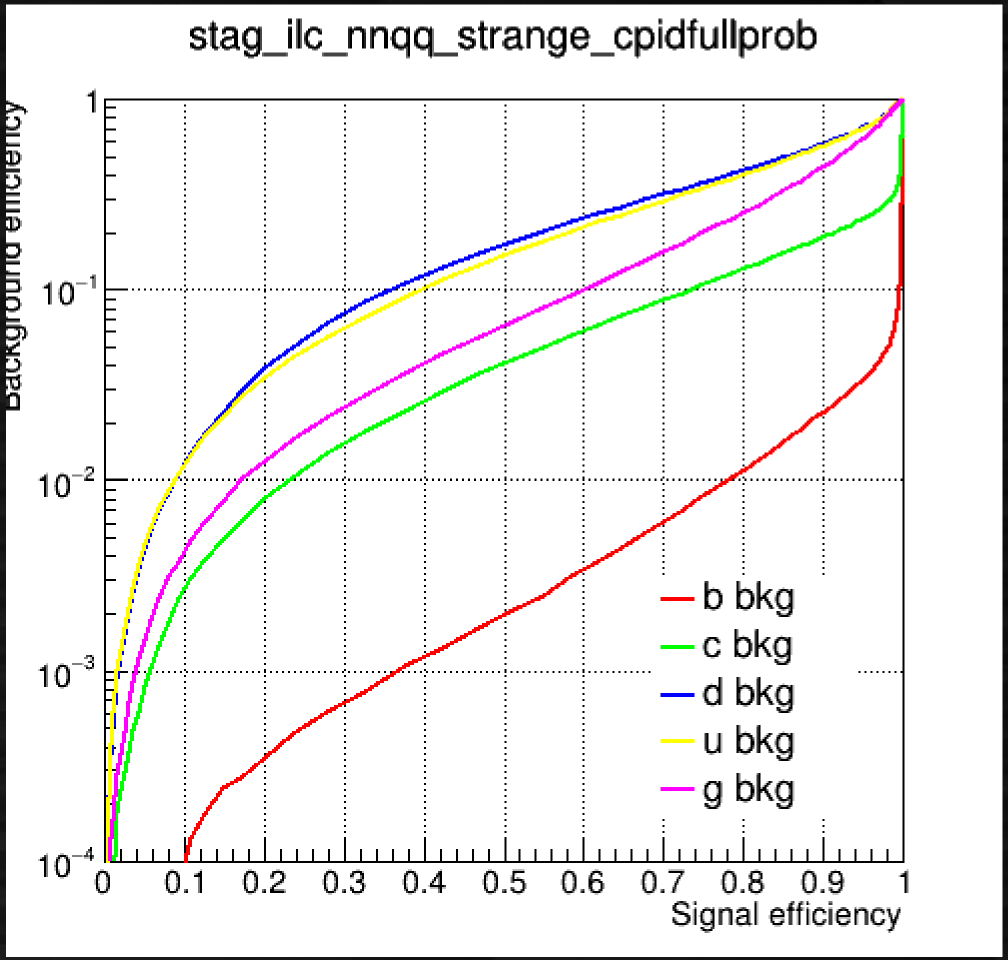
More Kaons in ss
More protons in gg

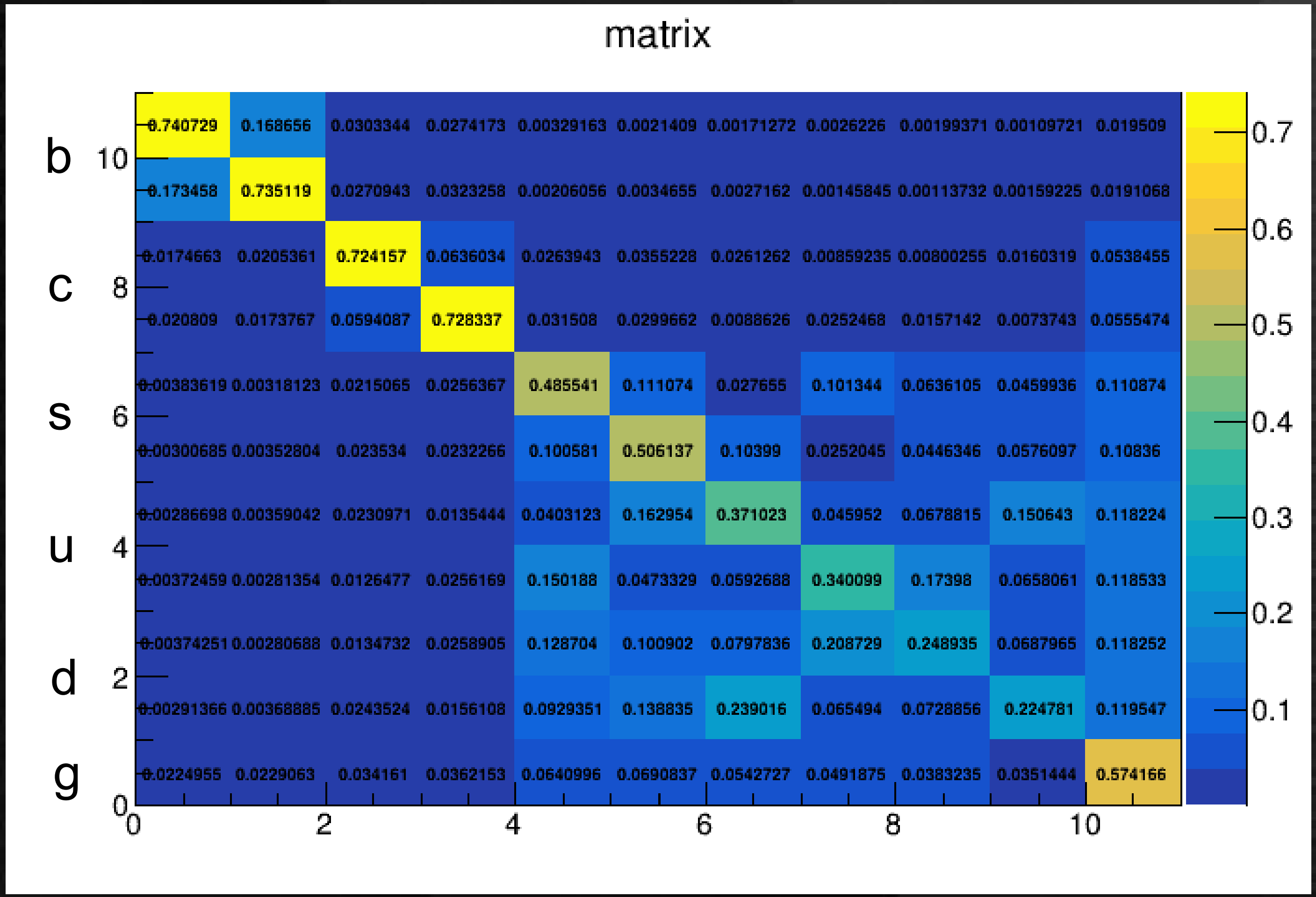
Fraction of true particles

True particle



11-category q/qbar tag (nnqq sample)





CPID (K/π/p probability)
with 100 ps TOF (x 10 hits) + dE/dx

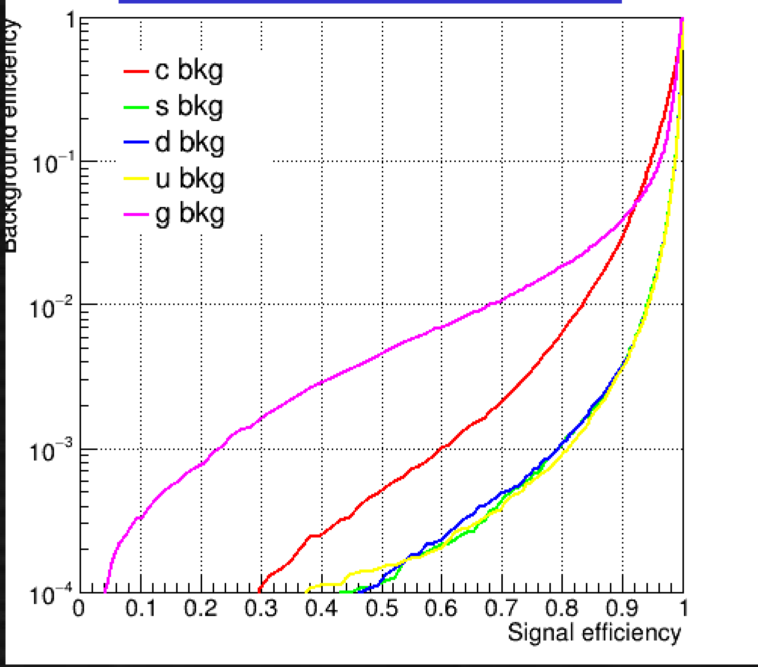
Vertical: truth jet PDG, horizontal: predicted jet PDG
PDG with highest score taken

Application to H → ss ongoing

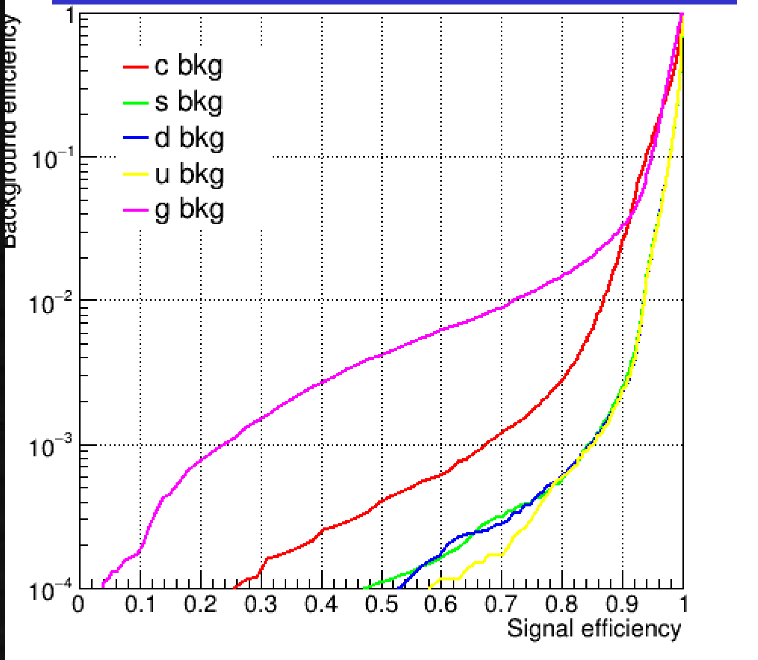
Comparison of fast/full sim and scaling law

B-tag performance, 6-category

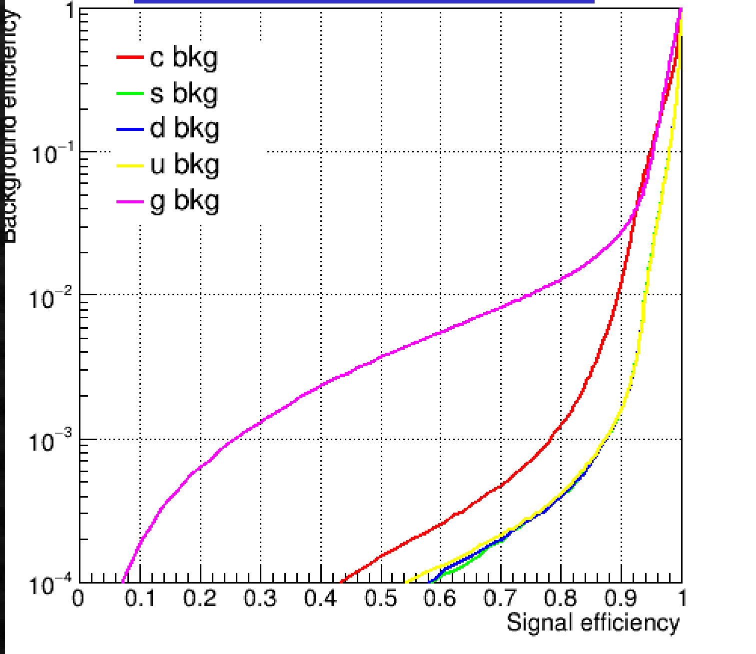
Full sim 1M sample



Fast sim (SGV) 1M sample



SGV 10M sample



b-tag 80% eff.	c bkg.	s bkg.	d bkg.	u bkg.	g bkg.
Full sim 1M	0.627%	0.105%	0.106%	0.088%	1.839%
Fast sim 1M	0.289%	0.059%	0.063%	0.061%	1.511%
Fast sim 10M	0.120%	0.039%	0.039%	0.041%	1.274%

Prospects on flavor tagging

- Checking "scaling law" with full simulation
  - 10M fast simulation tested, full sim to be produced
- Optimize input variables / network
  - "vertex information" to be included
- Direct application to event categorization
  - To reduce effect of misclustering in physics analysis
  - "Particles" + "Jets" input: how to combine?
  - How to integrate flavor tag results?
- One pass reconstruction + analysis
  → aiming for "one for all" network

Summary

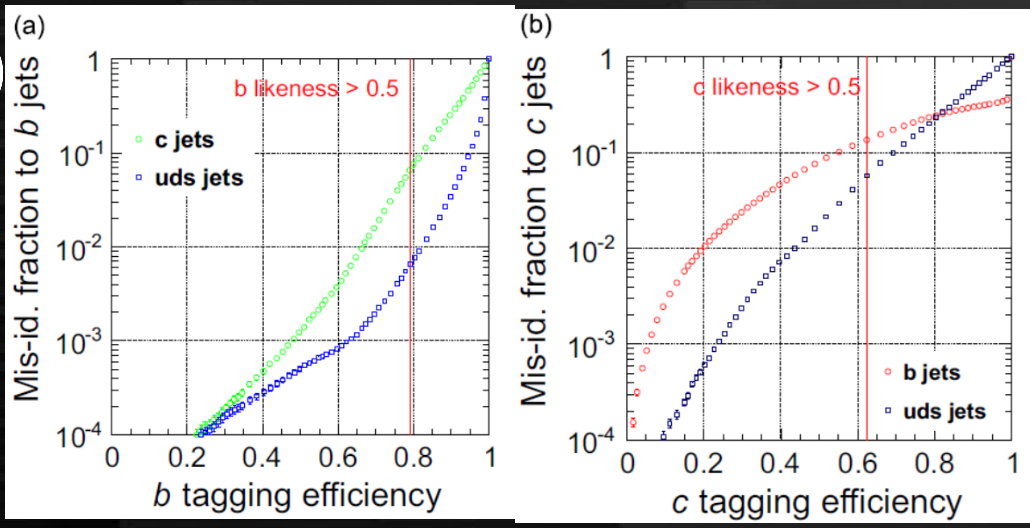
- AI-based reconstruction became essential for collider analyses
  - Higgs factory studies as well as LHC studies
- Flavor tag: already a game changer
  - ~10x better performance (both in LHC and HF)
  - Want to know “ultimate performance” with sufficient data/parameters
  - Combination with jet clustering / physics analyses to be pursued
- Particle flow: less significant improvements but still important development towards one-pass reconstruction
  - Also important for detector optimization/modelling
- Complicated task – room for more (JP-CN) collaboration!

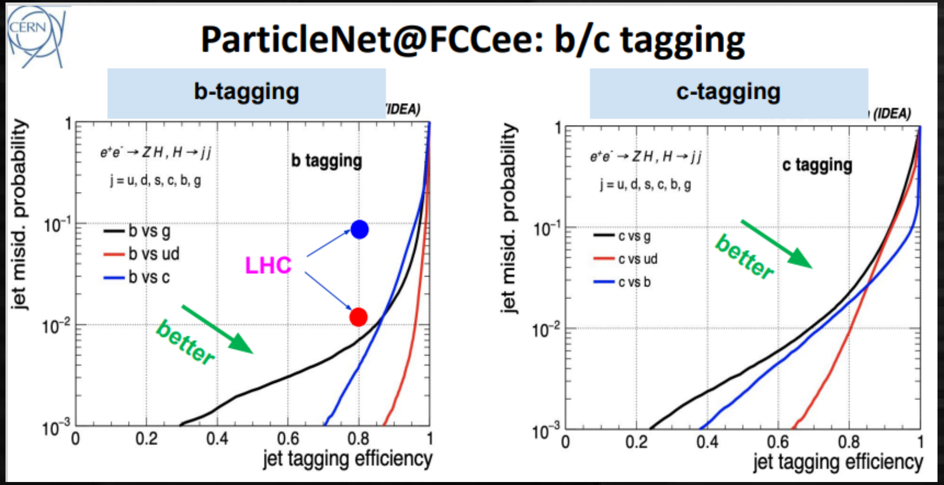
Backup

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





Data Samples and Input Variables

Data samples

- ILD full simulation

  1. e+ e- → qq (at 91 GeV) (used in LCFIPlus study)
  2. e+ e- → ννH → ννjj (at 250 GeV) (2020 production)

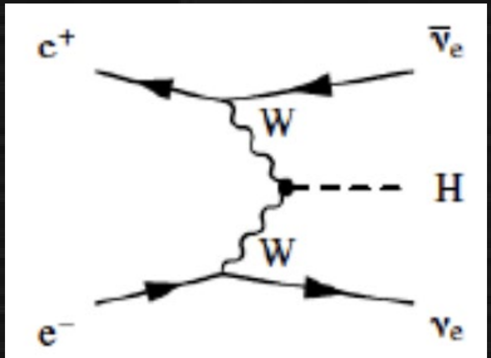
q = b,c,uds
j=b,c,u,d,s,g

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

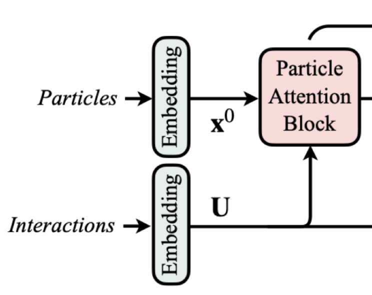
Particles: for every track/neutral

- 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

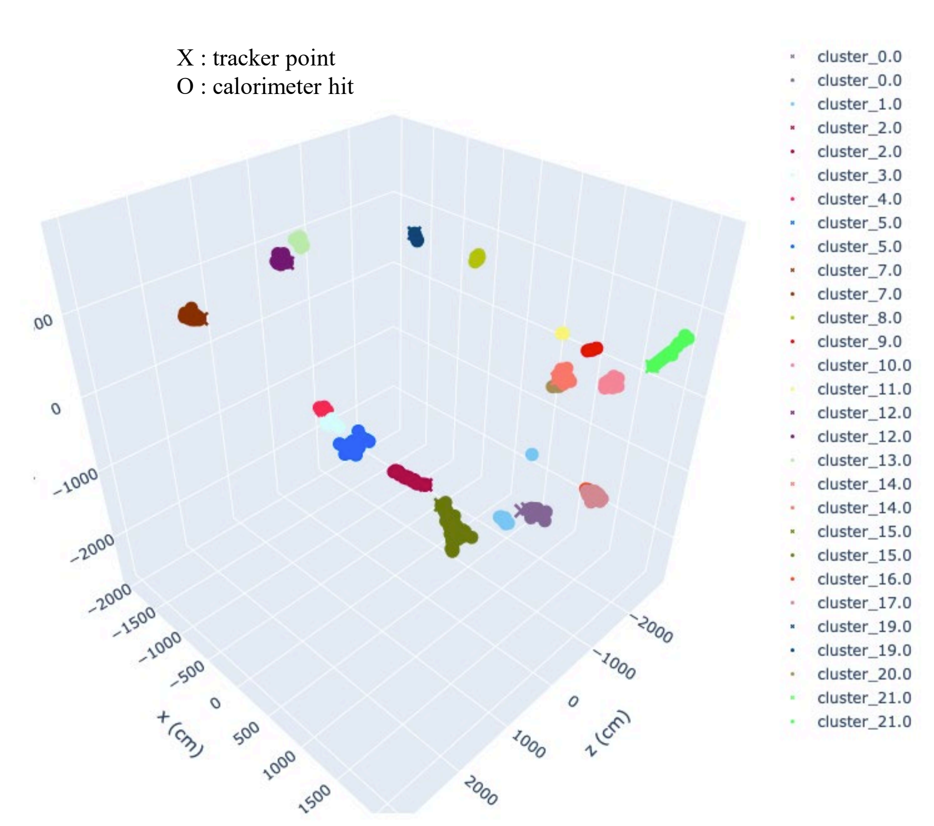
Interactions: for every particle pair

- δR^2, k_t, Z, mass

Input of ParT

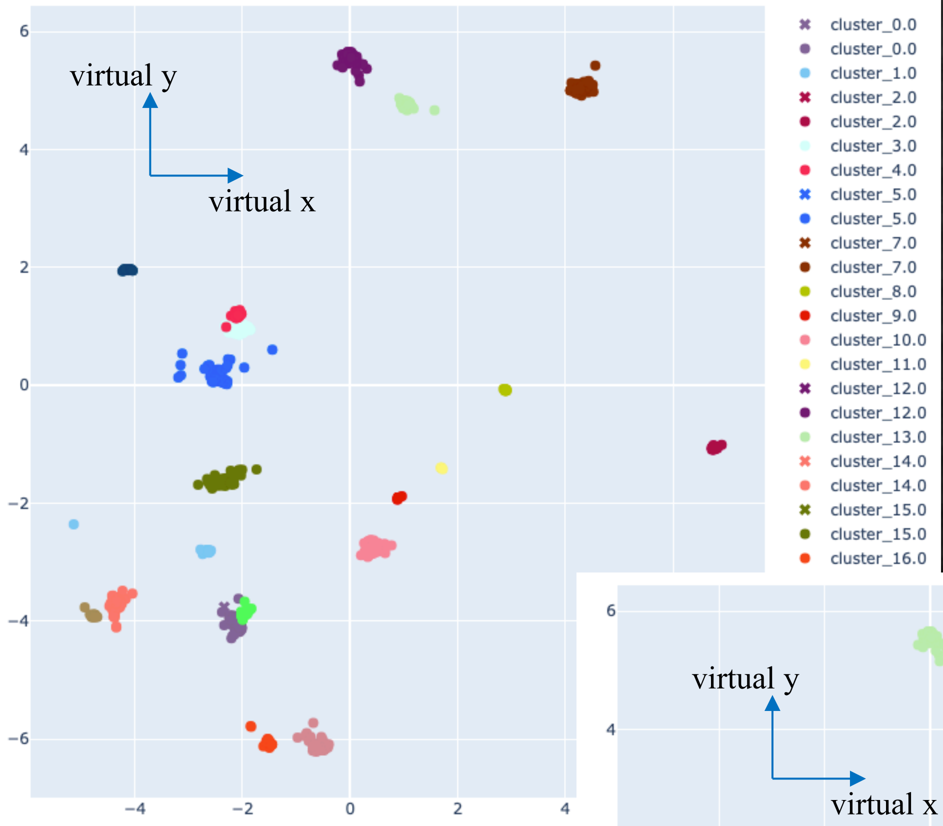


Event display



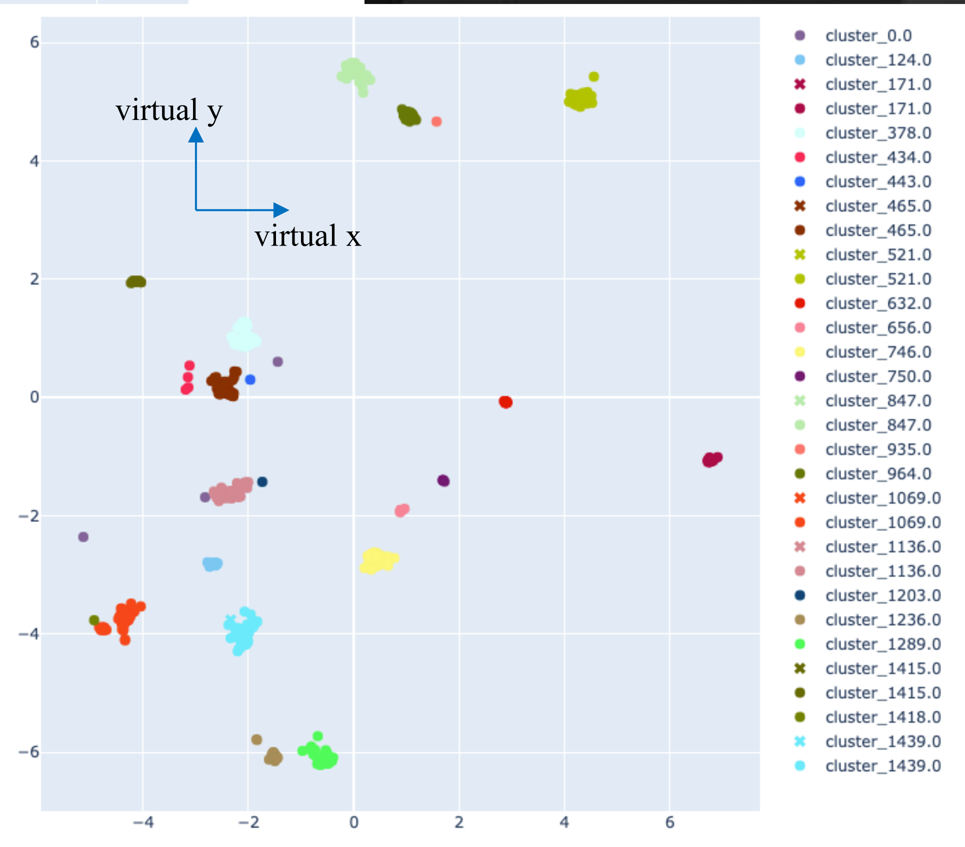
Input features
Real coordinate in detector

Colored by true clusters



Colored by
true clusters

Output features
Virtual coordinate



Colored by
reconstructed clusters

Taikan Suehara, QC

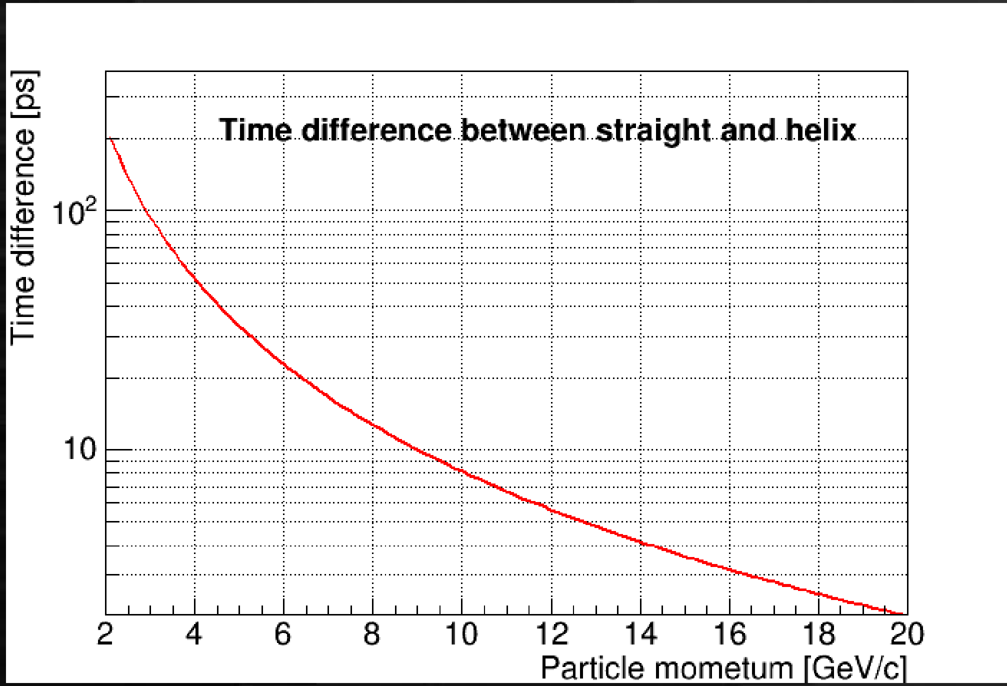
Summary and prospects for PFA

- GNN-based particle flow has possibility to replace PandoraPFA
  - Clustering performance is comparable with current optimization
  - Energy regression is being tried (reasonable performance with truth clustering)
- Possible improvements on algorithm (study ongoing)
  - Clustering algorithm (possibly with additional network)
  - Transformer-based network (in various ways)
- Test bench for detector design/optimization
  - Effects/advantages on new variables/measurements
    - Timing information (how much precision required?)
    - Particle ID (dE/dx, tof, …)
    - Pixel size (silicon pads vs scintillator vs MAPS), detector size, magnetic field etc.
- Application to physics analyses

Example: timing information

Timing information can be utilized in many ways

- Particle ID by ToF (e.g. pi/K/p separation)
  - Essential for strange tag
  - Should be good for PFA as well
- Separation of helix and straight path
  - Charged and neutral particles
- Off-axis photons (but need ~1psec resolution)
  - Should be useful for flavor tagging
    - b/c separation by mass



Performance on e.g. PID/PFA heavily depends on reconstruction software

- For PID: simple introducing timing info and check the performance should be easy
  - With any timing smearing
  - For non-ML, need to implement new algorithms and heavily tune it

Software for Particle Transformer

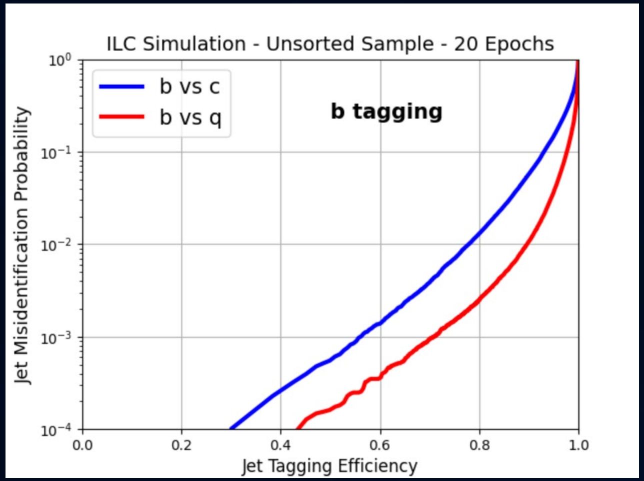
- Public in github, with instruction provided
  - https://github.com/jet-universe/particle_transformer
- Input: ROOT files for training (80%), validation (5%), test (15%)
  - Input variables can be provided via steering file (XML)
    - Input for each particle (tracks, neutral clusters)
    - Input for “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

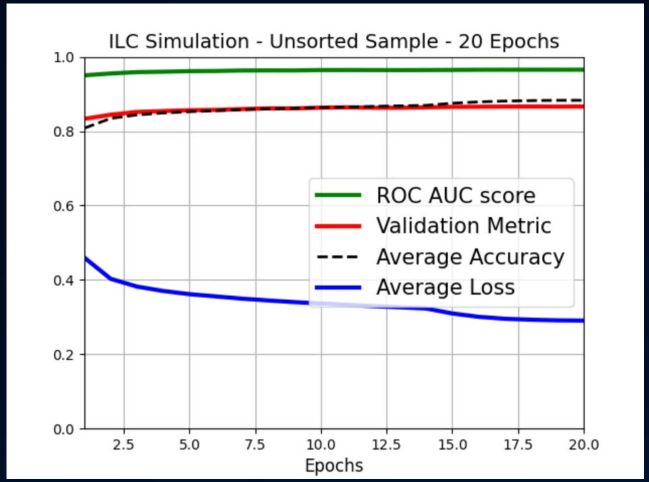
Software for Particle Transformer

- 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

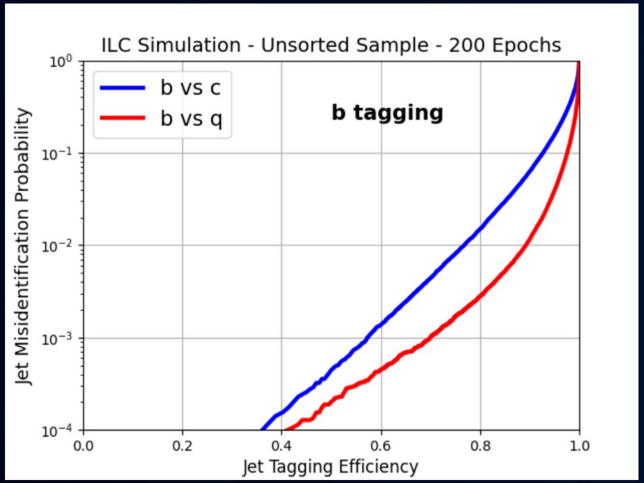
Training parameters - epochs

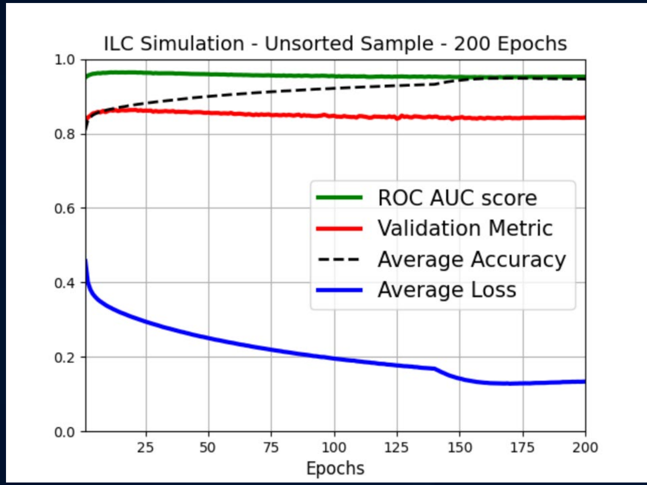
- Run on NVIDIA TITAN RTX (memory: 24 GB)
  - 20 Epochs: 3 hours
  - 200 Epochs: 30 hours
- No significant improvement in tagging efficiency
- Both ROC AUC score and Validation Metric reaches a maximum around 20 epochs.
- Overtraining after 20 epochs.
- Hence 20 epochs of training is selected to avoid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

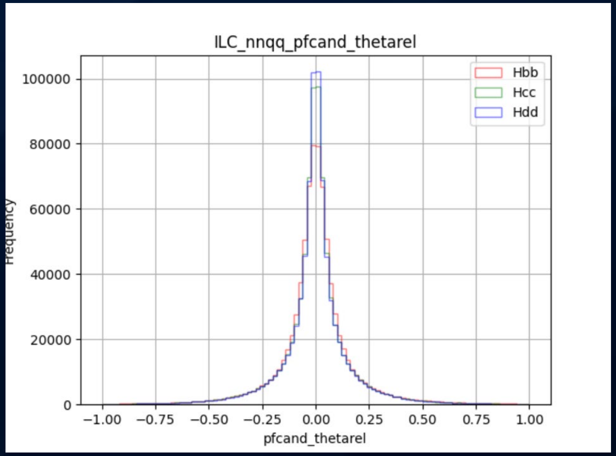
$$p = \sqrt{{p_x}^2 + {p_y}^2 + {p_z}^2}$$

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

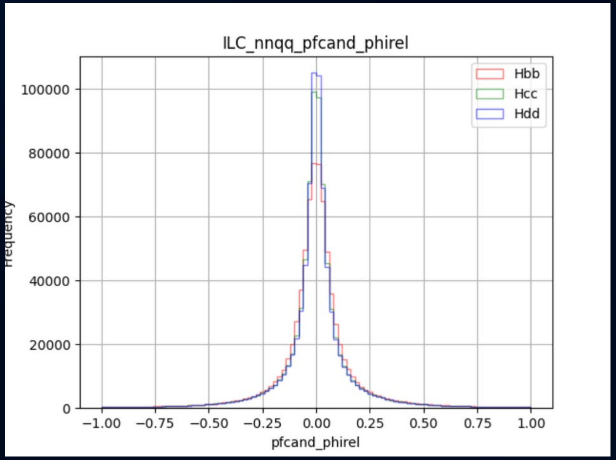
				c-bkg acceptance @ b-tag 80% eff.		b-bkg acceptance @ c-tag 50% eff.	
Particle ID	Impact Parameters	Jet Distance	Track Errors	p	p_x p_y p_z	p	p_x p_y p_z
✗	●	●	●	0.62%	0.49%	1.14%	1.01%
✗	● +log(abs)	● +log(abs)	● +log(abs)	0.54%	0.52%	1.06%	1.00%
✗	● +log(abs)	●	●	0.47%	0.50%	1.03%	0.97%

- ILD (vvqq 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.

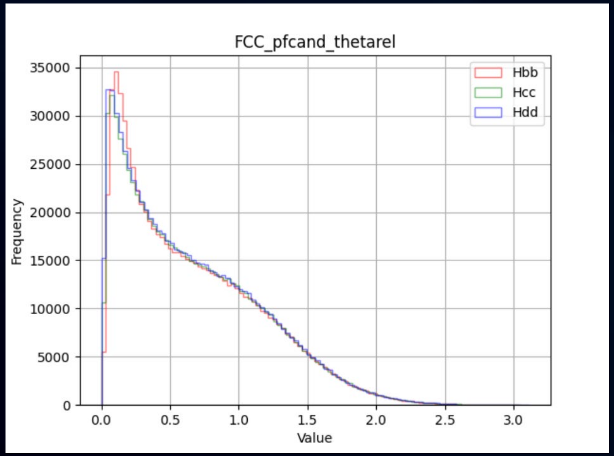
ILD vs. FCC – theta/phi distribution



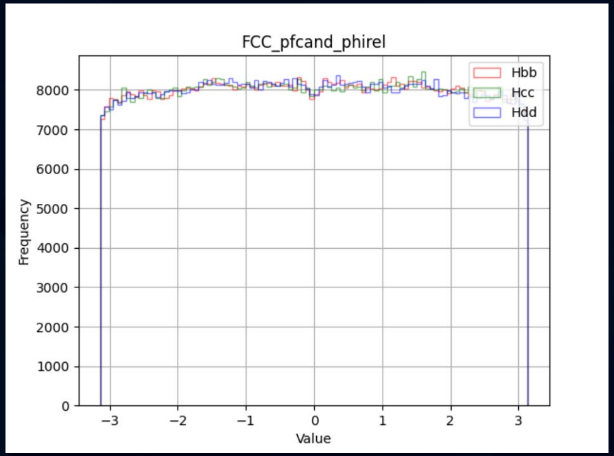
ILD theta



ILD phi



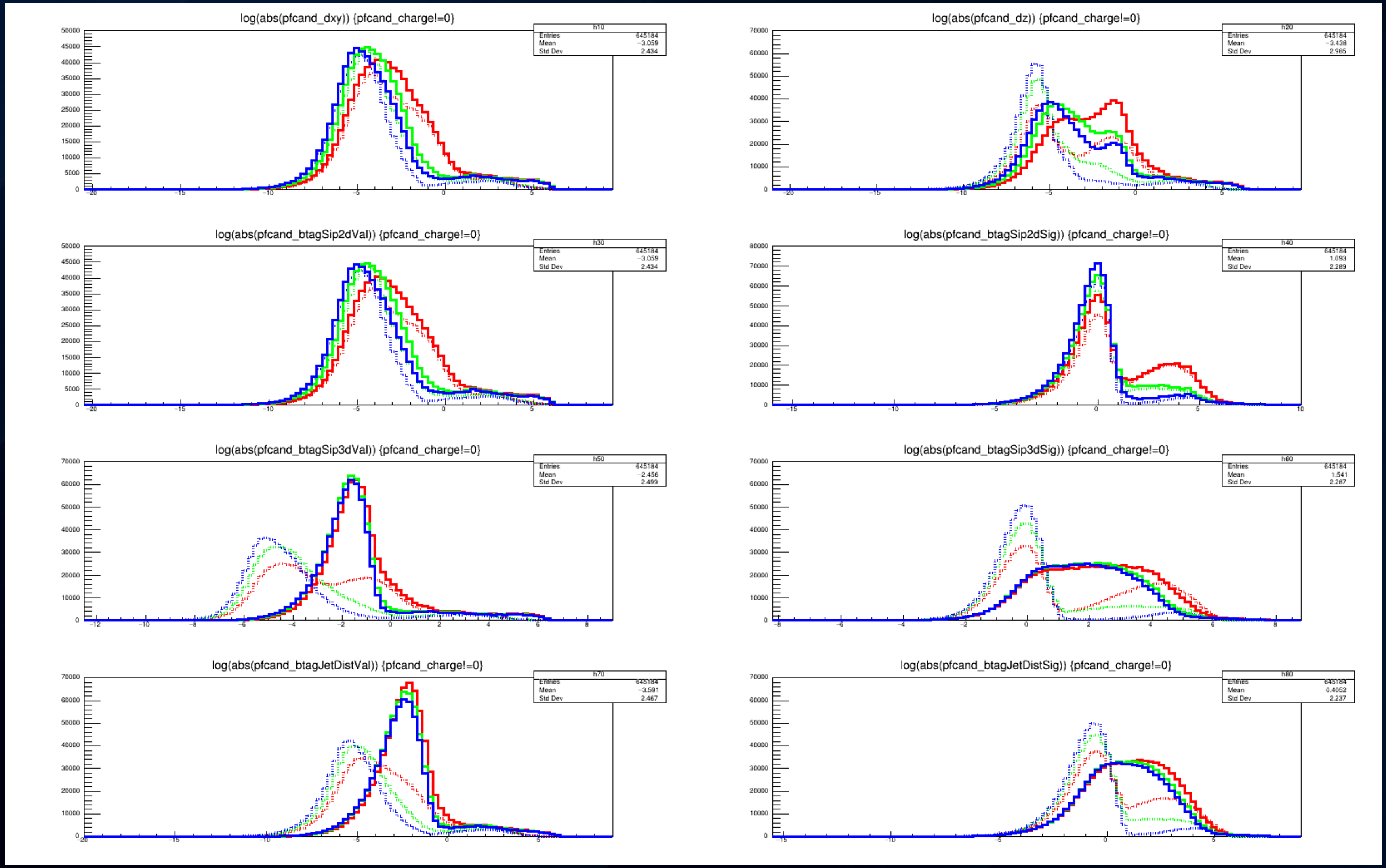
FCC theta



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?

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

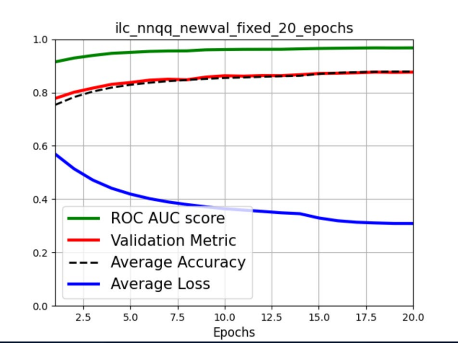
Fine tuning

							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%

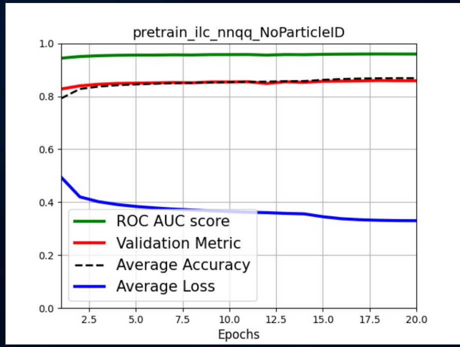
- Use result of 8M FCC data to train ILD 800k data
- Improves performance only when setups are similar
- Training of same setup (pretrain ILD 91 GeV data with ILD 250 GeV data) gives best performance
- Further investigation should be conducted on how to maximise the outcome for fine-tuning between different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

Fine tuning – Training curves

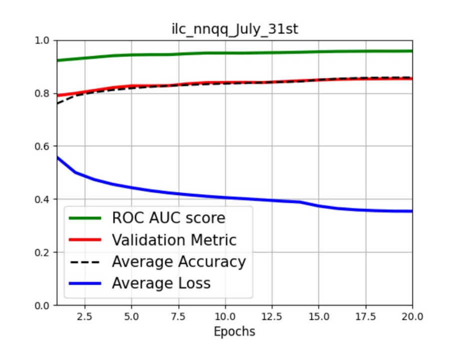
(1)



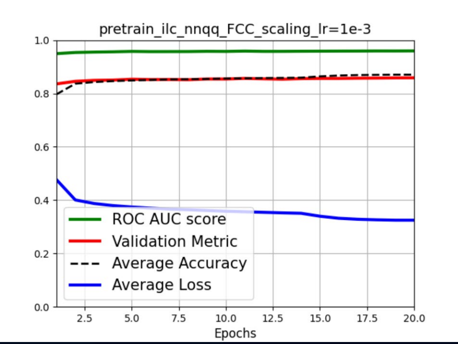
(2)



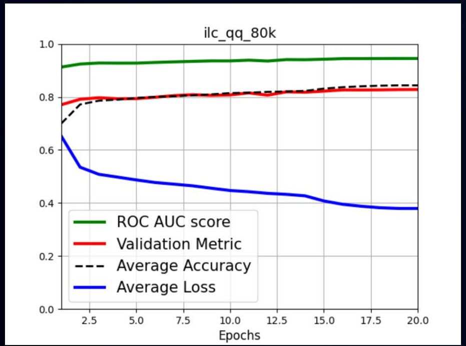
(3)



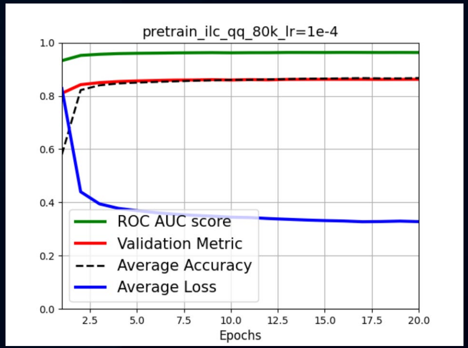
(4)



(5)



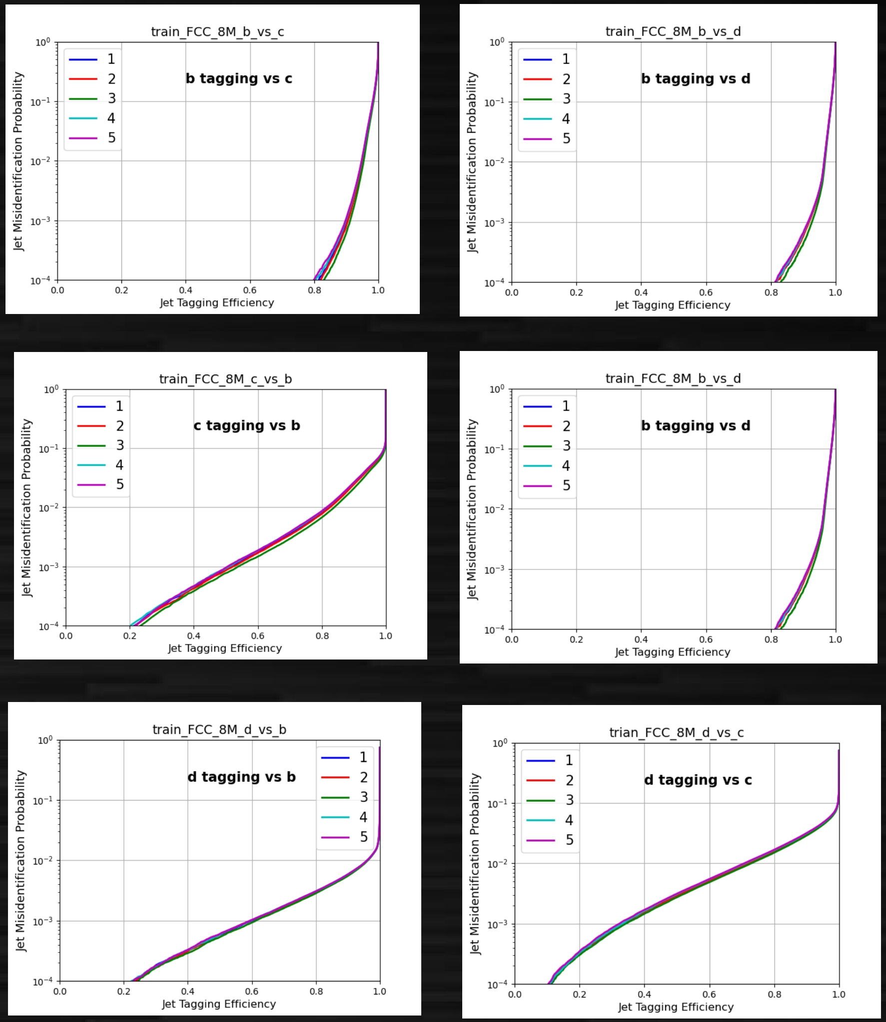
(6)



							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

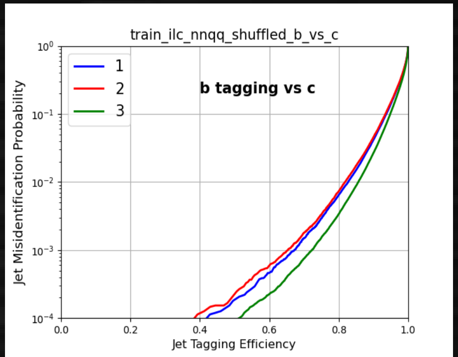


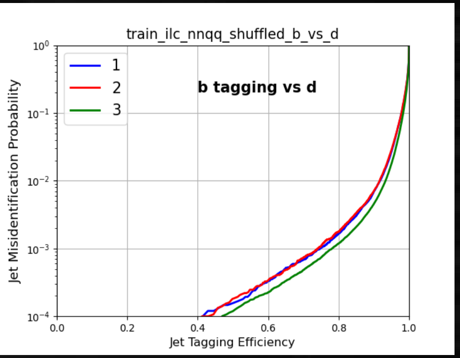
5 times training of FCC_8M data

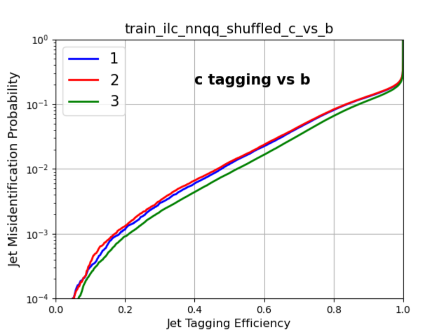
- Multiple training runs don't give significant impacts on results.
- The smaller data size is, the bigger impacts on results multiple runs give.
- The results of no Particle ID trainings varies more than those of with Particle ID.

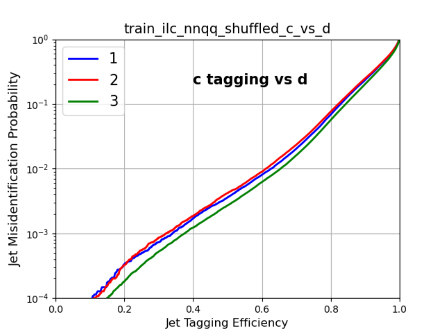
data	Particle ID	b vs c 0.8 Score	variation
FCC 4M	○	4.82e-4	0.43e-4
FCC 8M	○	8.14e-5	1.58e-5
FCC 4M	×	1.69e-3	0.14e-3
FCC 8M	×	7.04e-4	3.49e-4

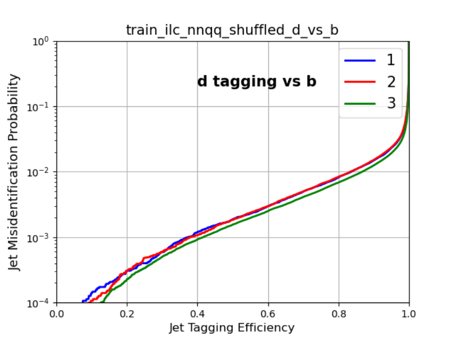
Data Shuffled

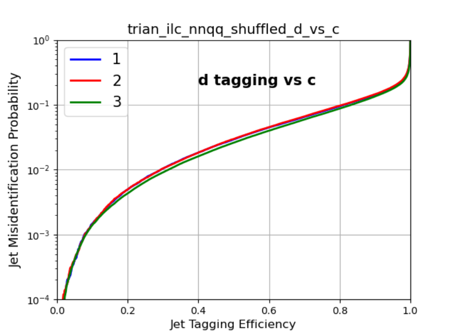












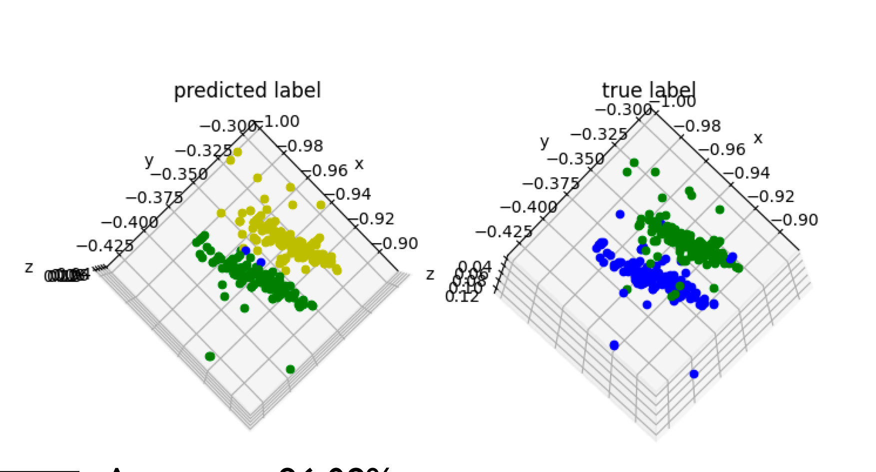
- ILC nnqq dataset
  - 80% training, 5% validation, 15% test
- Shuffled the order of train/test/val making root files
  - Pattern 1: train/val/test
  - Pattern 2: val/train/test
  - Pattern 3: train/test/val

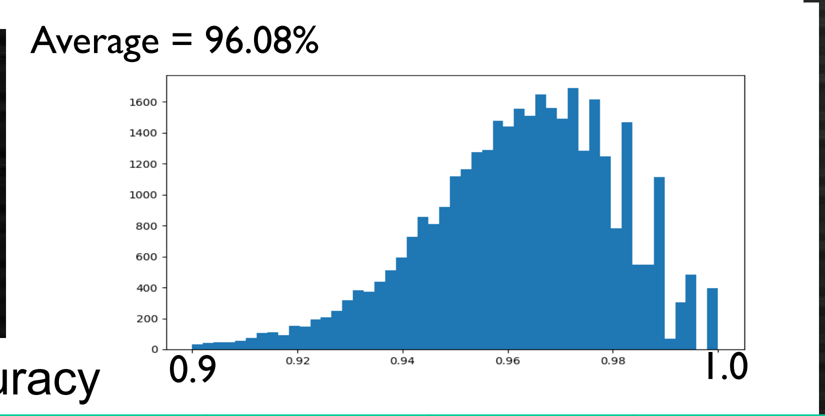
data	b vs c 0.8 score
Shuffle pattern 1	0.00647
Shuffle pattern 2	0.00734
Shuffle pattern 3	0.00338

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)



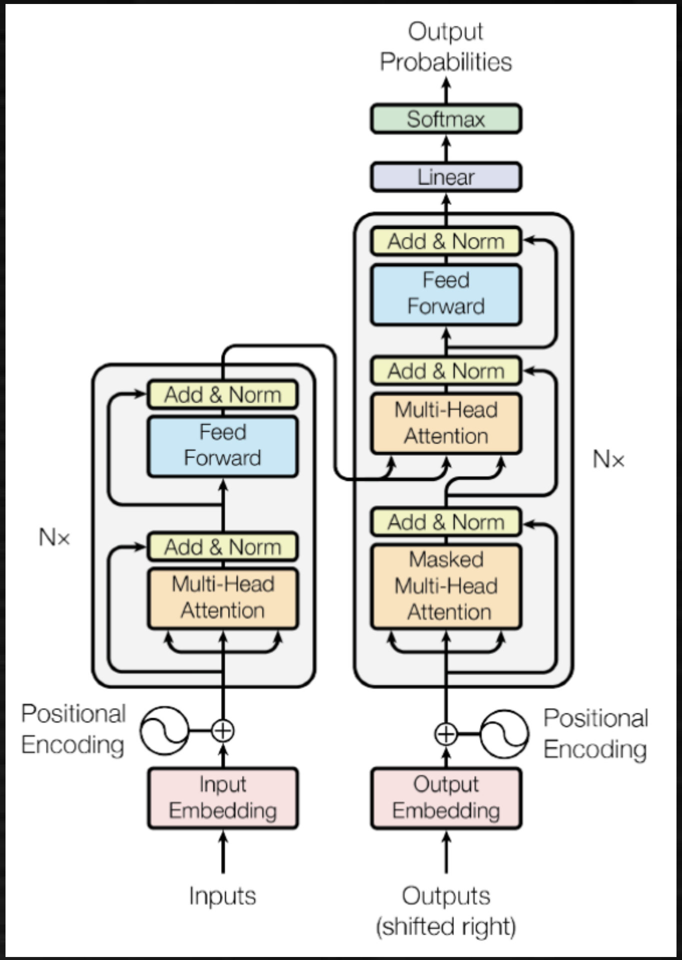


Reasonable
performance seen

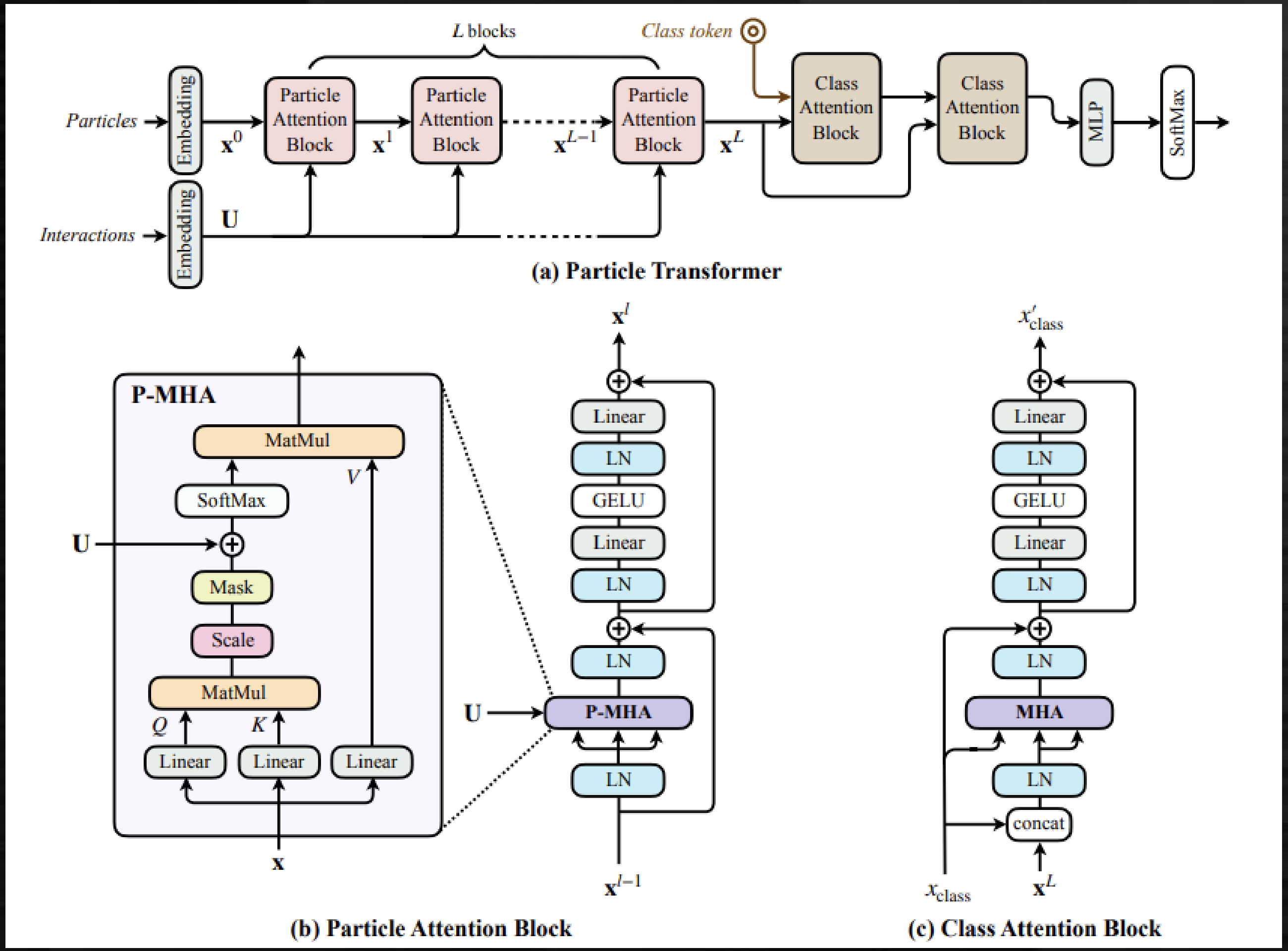
Angle[mrad]	30	60	90	120	150
Accuracy[%]	96.08	98.64	99.30	99.68	99.56

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



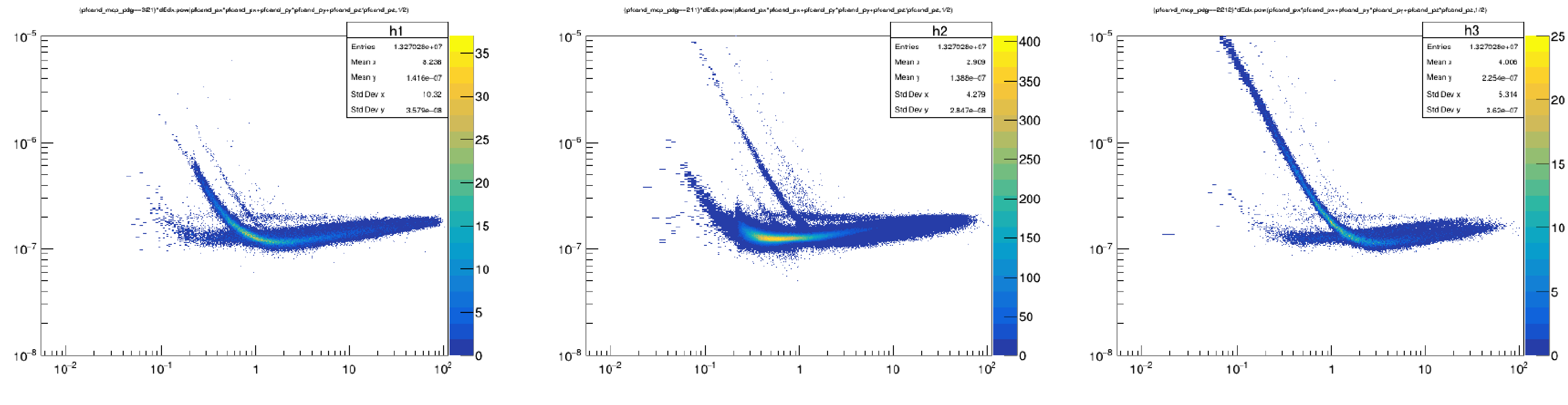
Particle Transformer

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)

Inference within LCFIPlus

- Training done in python/weaver framework
  - New LCFIPlus algorithm (MLMakeNtuple) to create input ROOT files
  - ROOT files used for training ParT
    - nnqq 250 GeV, ~1M jets / each flavor
    - MC/jet matching inside LCFIPlus (only for q/qbar training)
      - Color-singlet tagging by RecoMCTruthLink, q/g identified based on angle
        - » If multiple jets assigned to the same q/g, jet with highest energy taken
  - Training with GPU (~a half day for 20 epochs with Tesla V100)
- Weights (checkpoint) converted to onnx
  - Using onnx 1.15.0, onnxruntime 1.17.1 (to be compatible with key4hep)
- Inference with CPU in LCFIPlus framework
  - New processor MLInferenceWeaver with onnx files (uploaded in LCFIPlusConfig)
- Currently on private repository (pulling to official repository being processed)
  - LCFIPlus github with ParT, https://github.com/suehara/LCFIPlus/tree/onnx
  - LCFIPlusConfig with weight/steering files, https://github.com/suehara/LCFIPlusConfig

ILC: International Development Team

See LCWS2023: https://indico.slac.stanford.edu/event/7467/

WG3 physics group hosts series of physics meetings
https://agenda.linearcollider.org/category/266/
(Next: July 13th)
Mailing list subscription:
https://agenda.linearcollider.org/event/9154/

Established in 2020: aiming for ILC pre-lab
Pre-lab proposal in 2021
https://arxiv.org/abs/2106.00602
→ MEXT expert panel (2021)
- Not mature enough for proceeding to pre-lab
  - Mainly in international situation
- Accelerator technology should be developed in preparation for next step

→ Two steps towards pre-lab
- International Technology Network (ITN)
  - Collaboration framework with US/Europe
  - Doing time-critical works of pre-lab
  - Japanese part is funded by MEXT
- International Expert Panel
  - Among researchers connected to FA
  - Discussing how to proceed “global” projects