Particle flow and flavor tagging with Deep Neutral Network

T. Suehara (ICEPP, U. Tokyo/Kyushu U.), L. Gui (ICL/Kyushu U.), S. Tsumura (Kyushu U./KEK), T. Tanabe (MI-6 Ltd.),
H. Nagahara, Y. Nakashima (Osaka U.), N. Takemura (Kyushu Tech.), L. Gray, T. Klijnsma (Fermilab)

Deep learning with Higgs factories

- Significant part of reconstruction is “pattern recognition”
  - Cut-based method should have limitation
  - DNN should take more information than human-tuning
- “Big data” detector for Higgs factories
  - Much more detector elements than before
  - Should fit with modern network with many learning weights
  - Also good for detector design
- Sensor → objects → physics
  should be more seamless with deep learning techniques
  - Event reconstruction is the heart of the chain

Today’s topics

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

Particle flow with DNN (ongoing work, no conclusion)

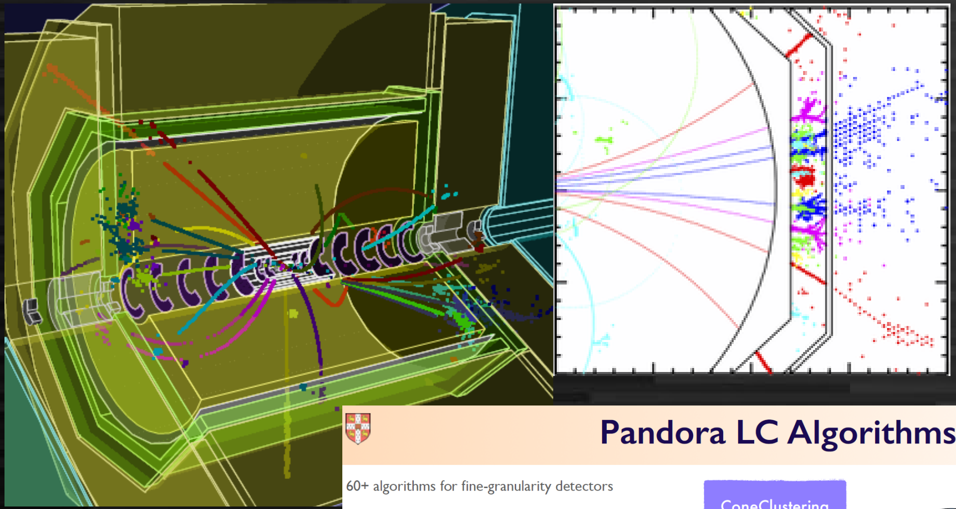
- Based on GNN developed for CMS HGCAL clustering
  - GravNet
  - Object condensation loss function
  - Timing information can be included → detector optimization
- Implementing track-cluster matching

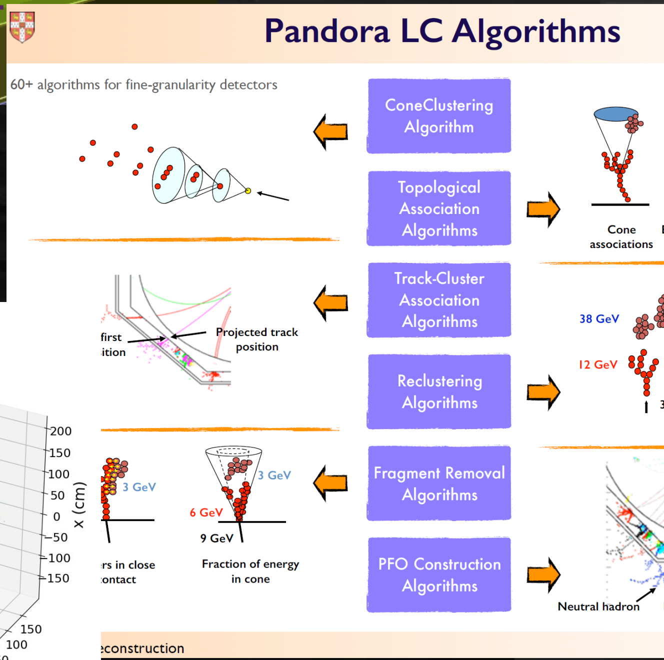
Flavor tagging with Particle Transformer (ParT)

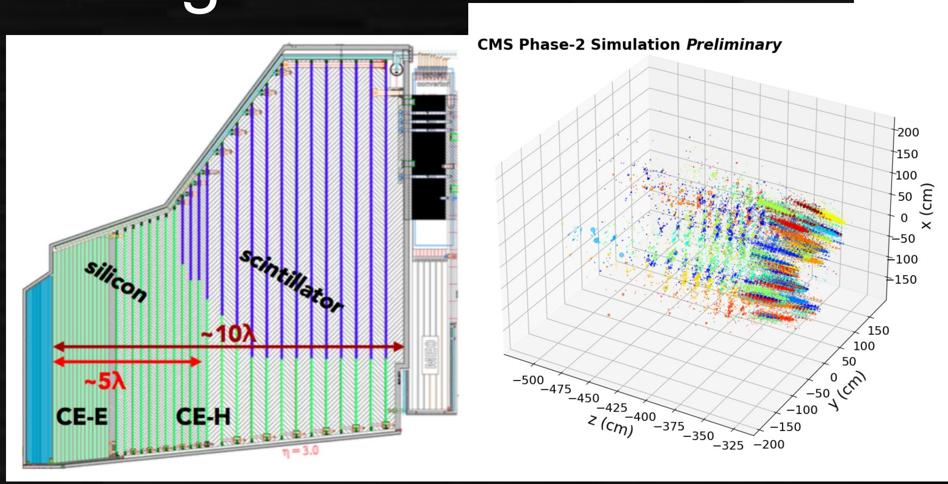
- Modern DNN-based jet flavor tagging developed for LHC
- Results reported by FCCee colleagues
  - Much better performance than current algorithm (LCFIPlus)
  - To be confirmed with full simulation
  - Big impact on Higgs studies

Particle flow with DNN: introduction

- Separation of cluster at calorimeter
  - Charged or neutral cluster
- Essential for jet energy resolution
- Current algorithm: PandoraPFA
  - Combination of various process
  - Not easy to optimize or adding more info
- CMS HGCal clustering
  - Similar to ILD calo
  - Good for starting point

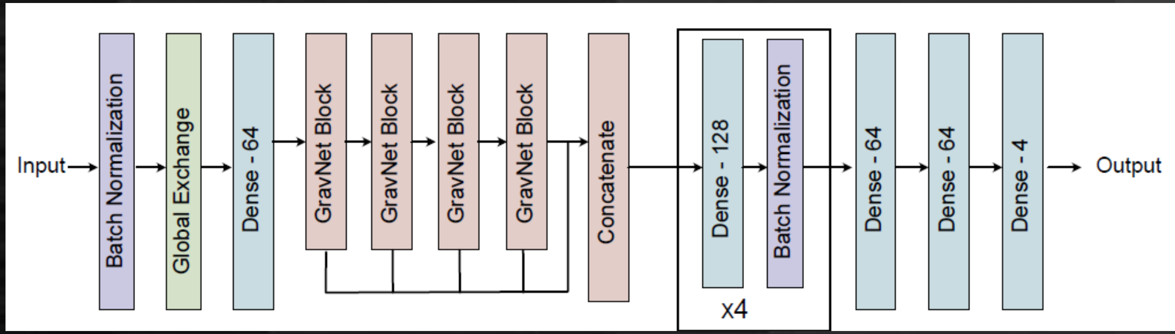






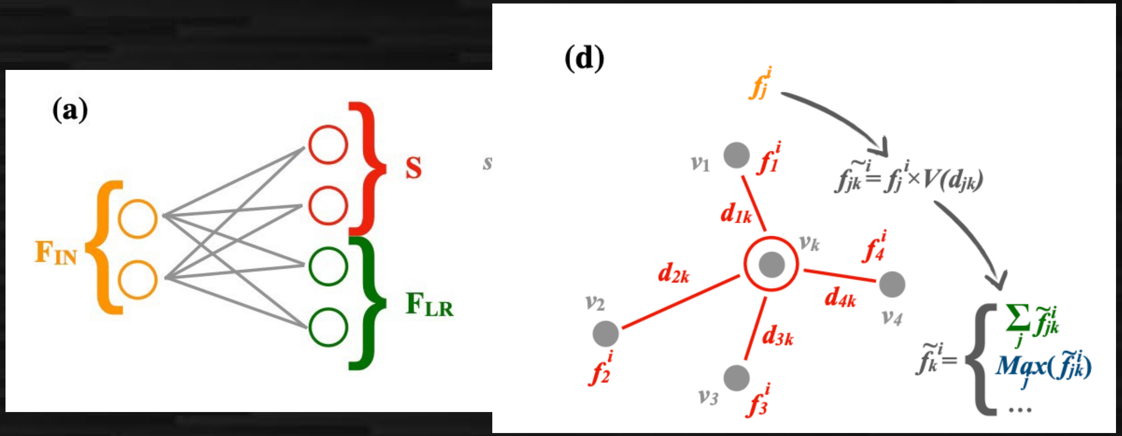
PFA: clustering algorithm

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



GravNet arXiv:1902.07987

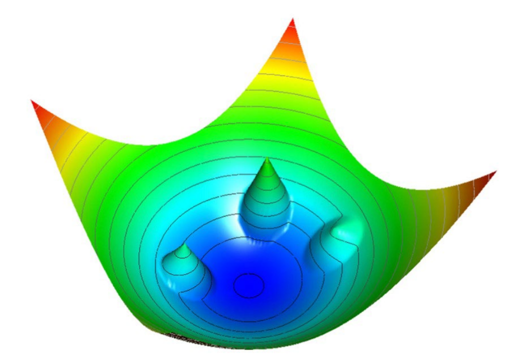
- The virtual coordinate (S) is derived from input variables with simple MLP
- Convolution using "distance" at S (bigger convolution with nearer hits)
- Concatenate the output with MLP



Object Condensation (loss function)

arXiv:2002.03605

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

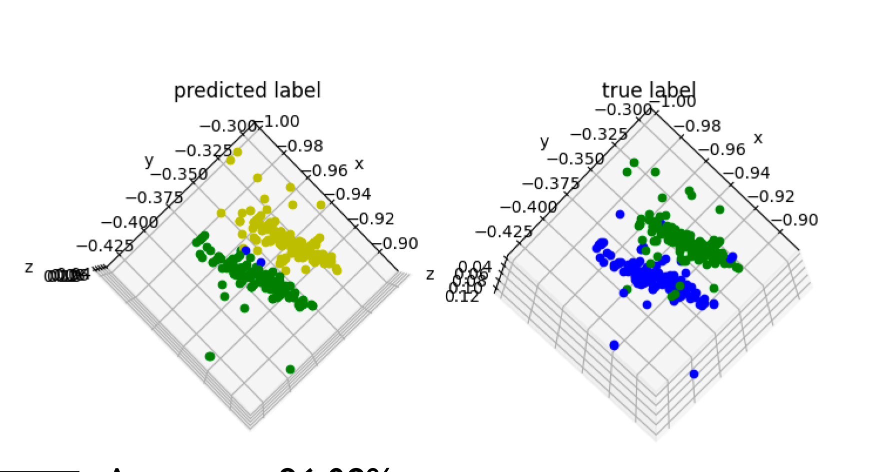


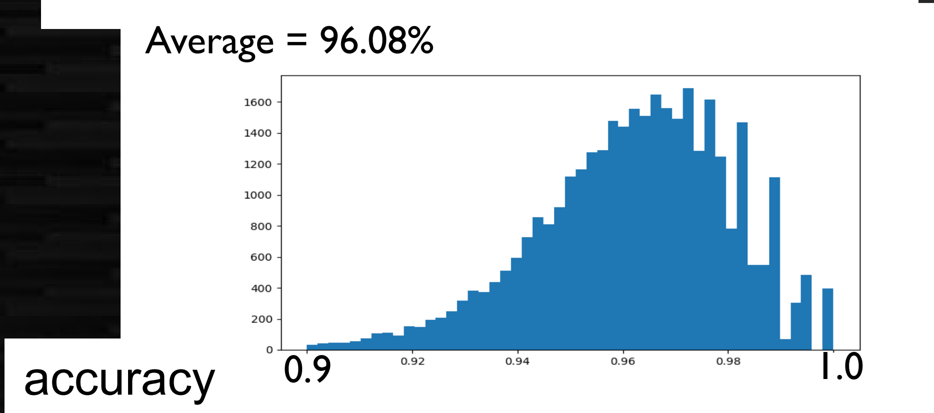
- Condensation point: The hit with largest β at each (MC) cluster
- L_V: Attractive potential to the condensation point of the same cluster and repulsive potential to the condensation point of different clusters
- L_β: Pulling up β of the condensation point
- L_p: Regression to output features

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

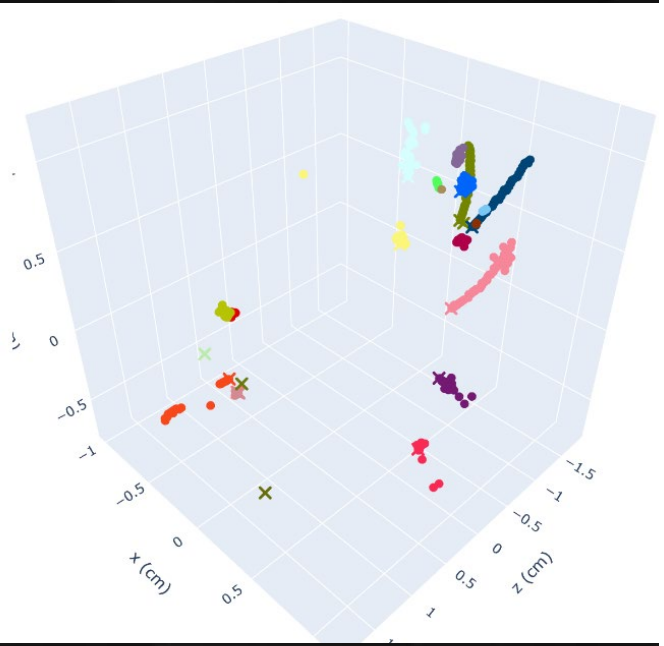
For details, refer eg. https://indico.slac.stanford.edu/event/7467/contributions/5948/attachments/2887/8032/230517-lcws2023-hlreco-suehara.pdf

Work in Progress: track-cluster matching

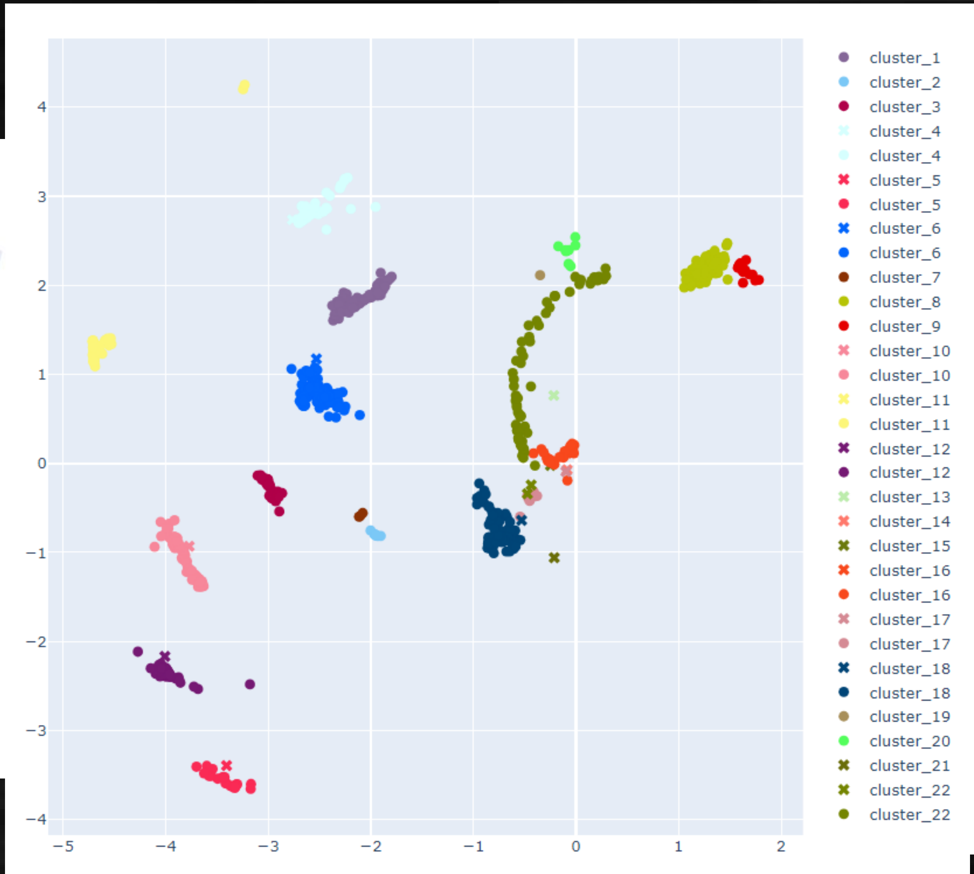
- PFA is essentially a problem “to subtract hits from tracks”
- HGCAL algorithm does not utilize track information
  - Only calorimeter clustering exists
- Simple extension to include track information
  - Adding “virtual hits” derived from track information
    - Hits at position where the track enters the calorimeter (from LCIO StackState)
  - Add a term to the object condensation loss function
    - Pulling up β of tracks (virtual hits) to promote them to condensation points (in addition to the usual beta-term, called beta-track term)
  - Evaluate fraction of (MC) charged clusters to be correctly assigned to clusters with tracks (virtual hits)

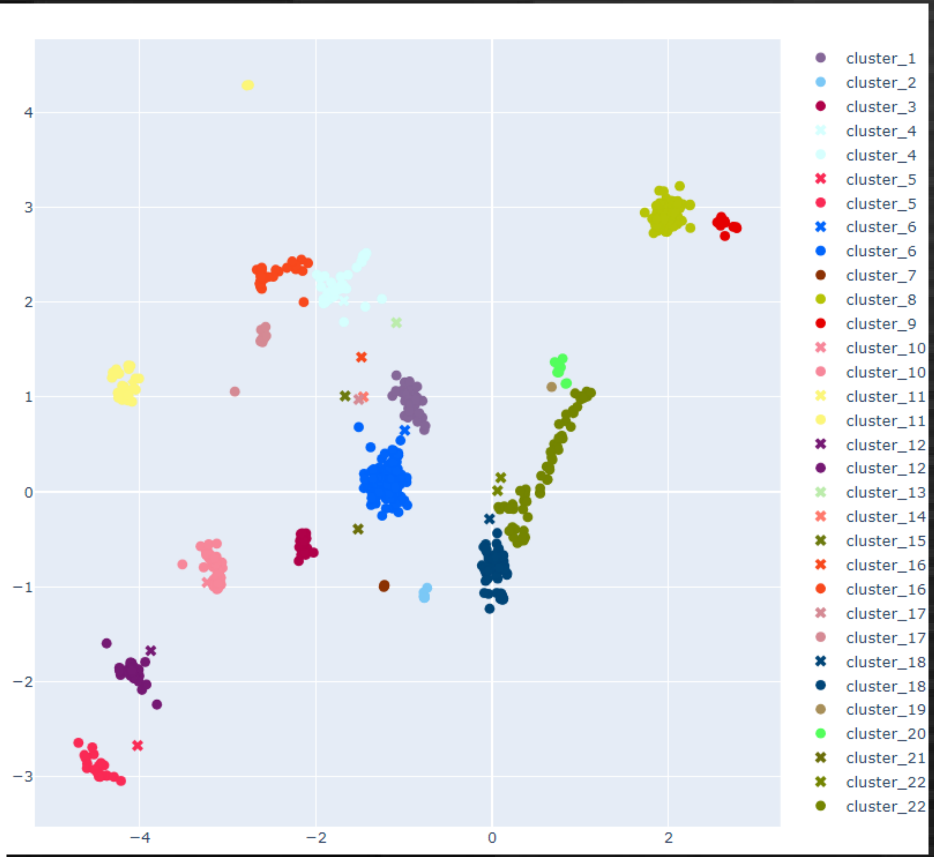
Preliminary results – event sample

10 Taus @ 10 GeV each



Real 3D coordinate

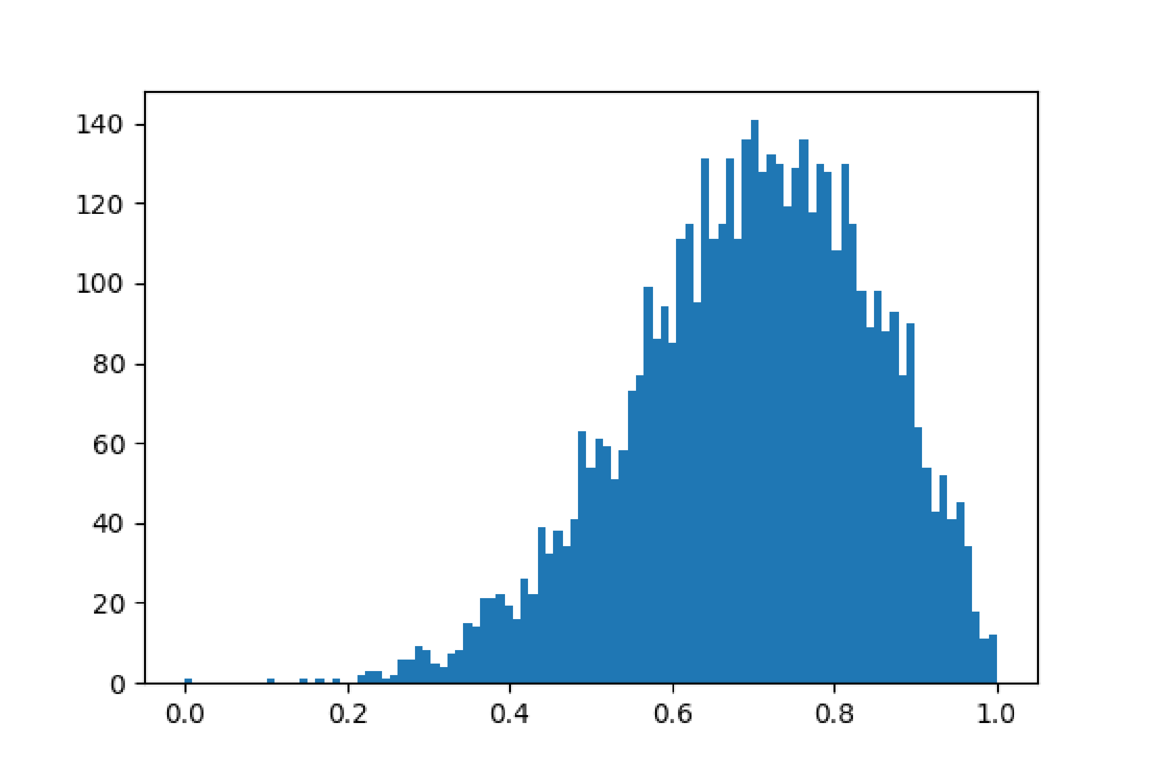




Hits on the virtual coordinate – colored by MC truth clusters
x refers virtual hits from tracks
left with beta-track term, right without beta-track term

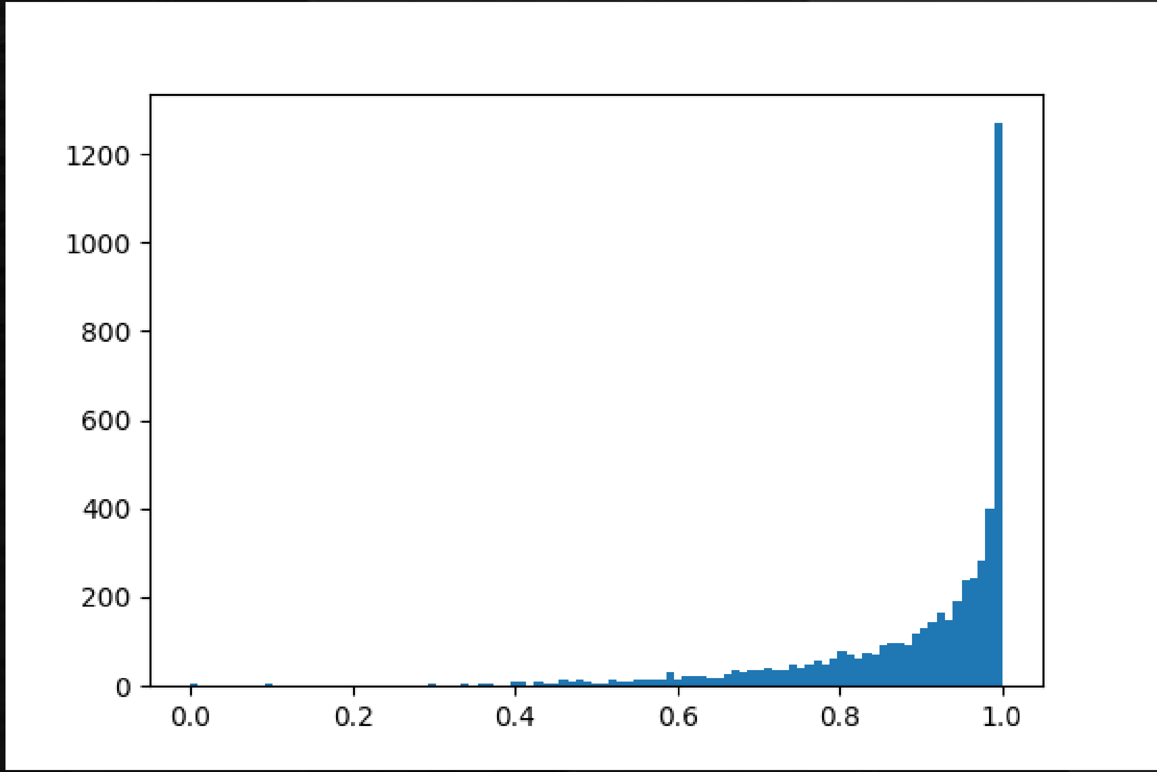
Evaluation of performance

Fraction of charged hit predicted correctly



Average: 0.698 (somehow low)

Fraction of neutral hit predicted correctly



Average: 0.890

Pred charged hit: associated cluster having at least one track

Ideas of improvement (short term / ongoing)

- Tuning of beta-track term (how to promote tracks to condensation point) → being done
- Utilizing track momentum
  - At clustering (additional conditional association, being tried)
  - Regression of momentum at the network (not done yet)
- Adding classification of track hit or not (being done)
  - Interaction/tuning with other terms

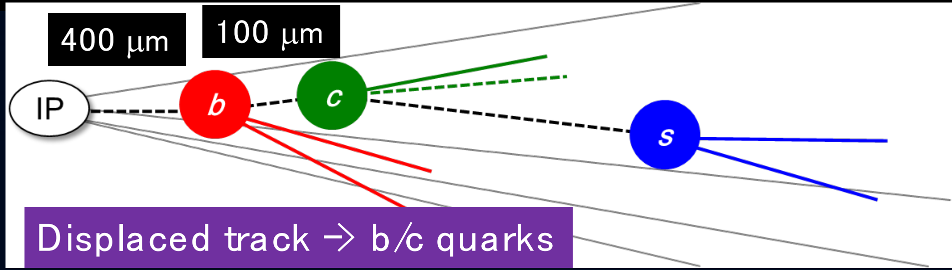
- Fully-DNN clustering (currently clustering is done with traditional way)

Application of Particle Transformer for Quark Flavor Tagging on Future Higgs Factories

Taikan Suehara (Kyushu → ICEPP Tokyo),
Lai Gui (Summer student at Kyushu, from ICL)

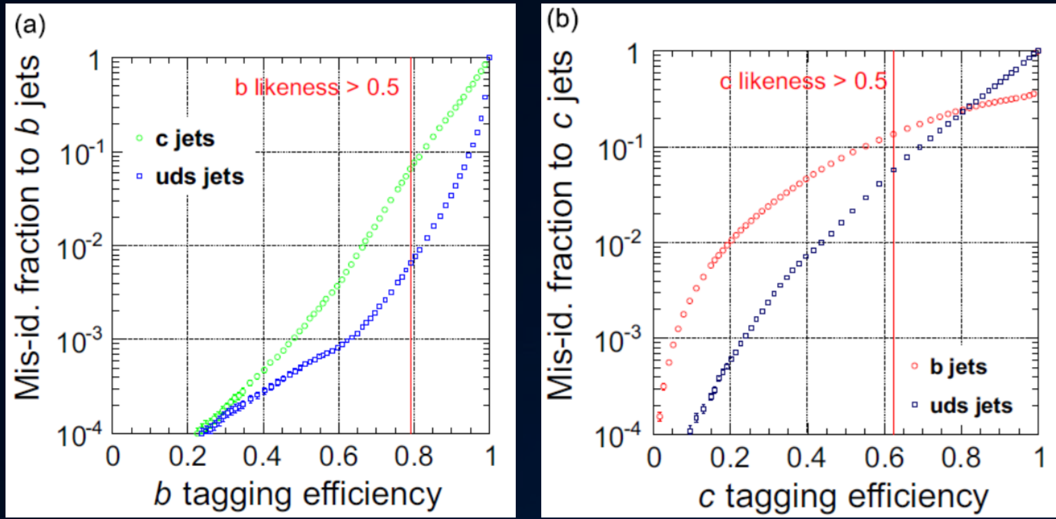
All results are preliminary: need to check reproducibility with shuffled events etc. (TBD)

Flavor tagging for Higgs factories



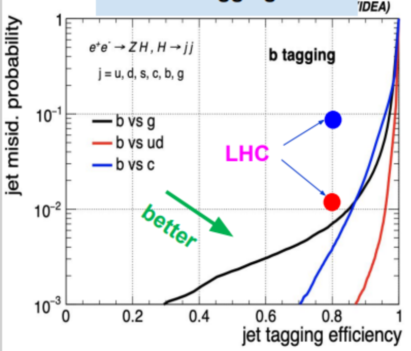
- Jet flavor tagging is essentially important for Higgs studies (including self coupling)
- LCFIPlus (published 2013)[1] was long used for flavor tagging
  - b-tag: ~80% eff., 10% c / 1% uds acceptance;
  - c-tag: ~50% eff., 10% b / 2% uds acceptance.
- Recently FCCee reported ~10x better rejection using ParticleNet (GNN)
  - To be confirmed with full simulation (with latest algorithm: Particle Transformer (ParT)

  → If good, consider to apply to physics analyses hopefully with common framework



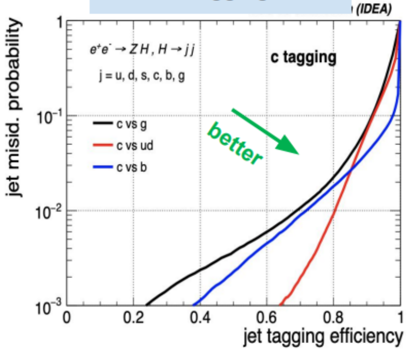
ParticleNet@FCCee: b/c tagging

b-tagging



WP	Eff (b)	Mistag (g)	Mistag (ud)	Mistag (c)
Loose	90%	2%	0.1%	2%
Medium	80%	0.7%	<0.1%	0.3%

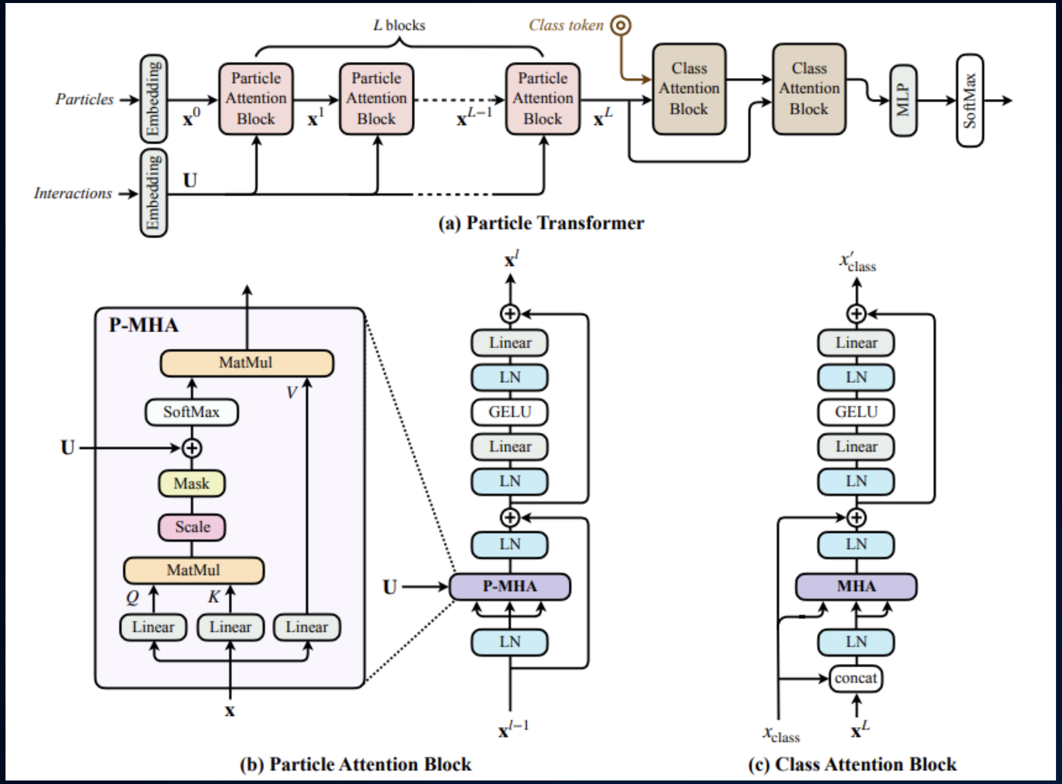
c-tagging



WP	Eff (c)	Mistag (g)	Mistag (ud)	Mistag (b)
Loose	90%	7%	7%	4%
Medium	80%	2%	0.8%	2%

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[2].
- Surpasses the performance of previous architectures
- Easily usable with TTree input and XML steering file



Performance on event categorization (ie. not direct flavor tagging but flavor information is essential for the categorization)

	All classes		$H \to b\bar{b}$	$H \to c\bar{c}$	$H \to gg$	$H \to 4q$	$H \to \ell\nu qq'$	$t \to bqq'$	$t \to b\ell\nu$	$W \to qq'$	$Z \to q\bar{q}$
	Accuracy	AUC	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{99\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{99.5\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$
PFN	0.772	0.9714	2924	841	75	198	265	797	721	189	159
P-CNN	0.809	0.9789	4890	1276	88	474	947	2907	2304	241	204
ParticleNet	0.844	0.9849	7634	2475	104	954	3339	10526	11173	347	283
ParT	**0.861**	**0.9877**	**10638**	**4149**	**123**	**1864**	**5479**	**32787**	**15873**	**543**	**402**
ParT (plain)	0.849	0.9859	9569	2911	112	1185	3868	17699	12987	384	311

Data Used For Investigation

- ILD full simulation:

  1. e+ e- ⟶ qq (at 91 GeV)
     (DBD sample used for initial LCFIPlus study)
  2. e+ e- ⟶ ννH ⟶ ννqq (at 250 GeV)
     (2020 production, process ID: 410001-410006)

  With 1M jets (500k events) each

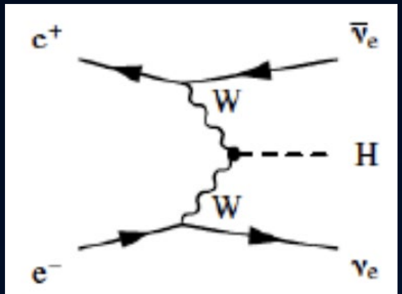
- FCCee fast simulation (Delphes with IDEA detector):

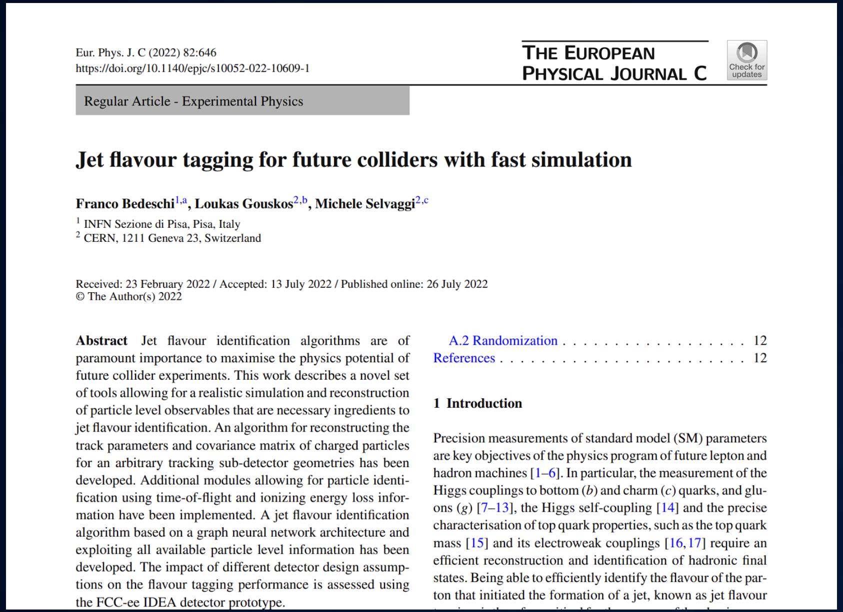
  e+ e- ⟶ ννH ⟶ ννqq (at 240 GeV)

  With 10M jets (5M events) each

- 80% are used for training, 5% for validation, 15% for test

$$\left\{ \begin{array}{l} \text{q = b,c,uds} \\ \text{ν = neutrino} \end{array} \right\}$$




Eur. Phys. J. C (2022) 82:646
https://doi.org/10.1140/epjc/s10052-022-10609-1

THE EUROPEAN PHYSICAL JOURNAL C

Regular Article - Experimental Physics

Jet flavour tagging for future colliders with fast simulation

Franco Bedeschi[1,a], Loukas Gouskos[2,b], Michele Selvaggi[2,c]
[1] INFN Sezione di Pisa, Pisa, Italy
[2] CERN, 1211 Geneva 23, Switzerland

Received: 23 February 2022 / Accepted: 13 July 2022 / Published online: 26 July 2022
© The Author(s) 2022

Abstract Jet flavour identification algorithms are of paramount importance to maximise the physics potential of future collider experiments. This work describes a novel set of tools allowing for a realistic simulation and reconstruction of particle level observables that are necessary ingredients to jet flavour identification. An algorithm for reconstructing the track parameters and covariance matrix of charged particles for an arbitrary tracking sub-detector geometries has been developed. Additional modules allowing for particle identification using time-of-flight and ionizing energy loss information have been implemented. A jet flavour identification algorithm based on a graph neural network architecture and exploiting all available particle level information has been developed. The impact of different detector design assumptions on the flavour tagging performance is assessed using the FCC-ee IDEA detector prototype.

A.2 Randomization 12
References 12

1 Introduction

Precision measurements of standard model (SM) parameters are key objectives of the physics program of future lepton and hadron machines [1–6]. In particular, the measurement of the Higgs couplings to bottom (b) and charm (c) quarks, and gluons (g) [7–13], the Higgs self-coupling [14] and the precise characterisation of top quark properties, such as the top quark mass [15] and its electroweak couplings [16,17] require an efficient reconstruction and identification of hadronic final states. Being able to efficiently identify the flavour of the parton that initiated the formation of a jet, known as jet flavour

https://link.springer.com/article/10.1140/epjc/s10052-022-10609-1

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)

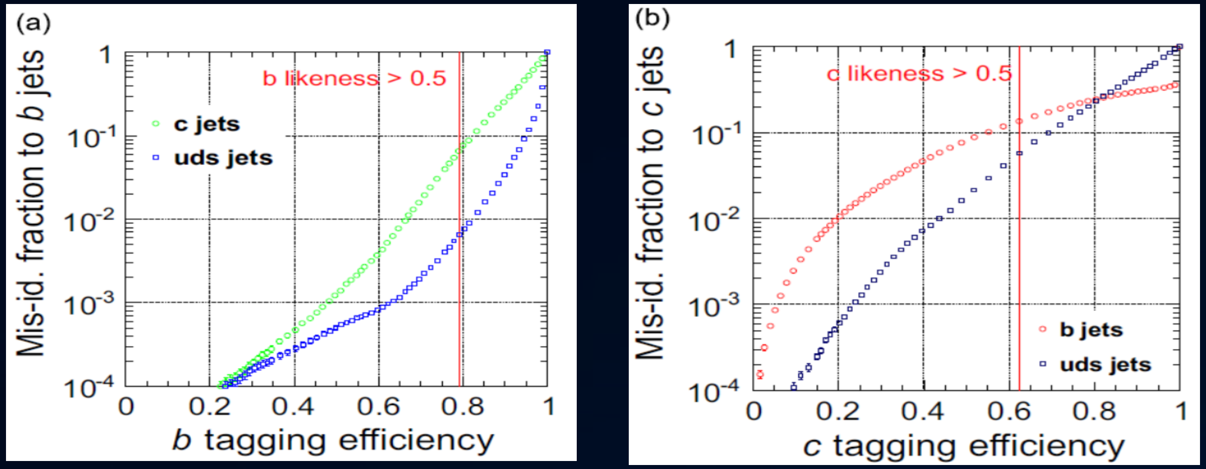
- Track Errors (15):

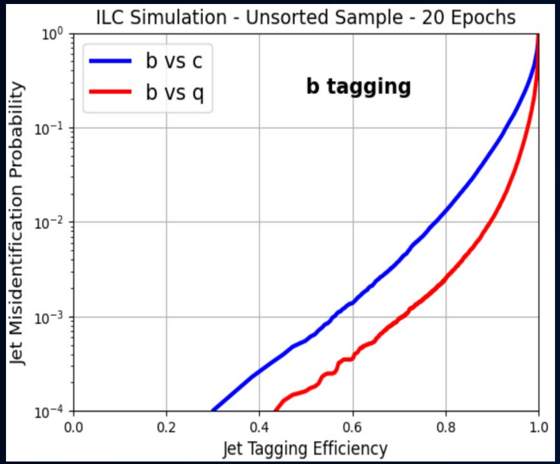
pfcand_dptdpt
pfcand_detadeta
pfcand_dphidphi
pfcand_dxydxy
pfcand_dzdz
pfcand_dxydz
pfcand_dphidxy
pfcand_dlambdadz
pfcand_dxyc
pfcand_dxyctgtheta
pfcand_phic
pfcand_phidz
pfcand_phictgtheta
pfcand_cdz
pfcand_cctgtheta

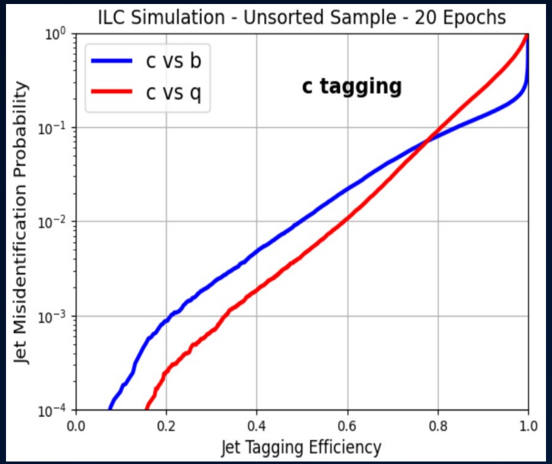
*each element of covariant matrix
0 for neutrals

Application of ParT to ILD data
(ILD qq 91 GeV, 0.8M jets for training)

- Jet tagging performance is greatly improved by ParT immediately.
- The performance is improved by 4.05 – 9.80 times compared to LCFIPlus with the same set of data.
- 20 epochs are taken,
  200 epochs do not help improving performance but give overtraining



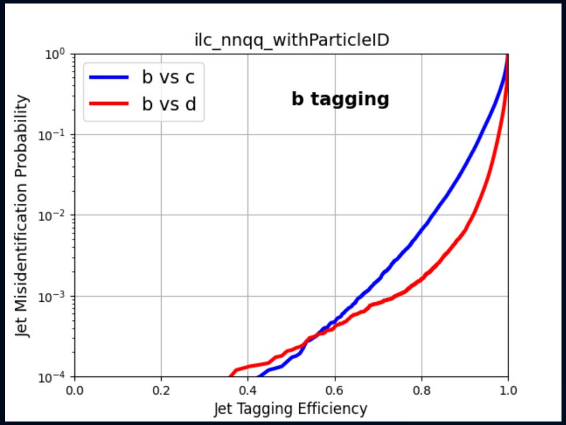


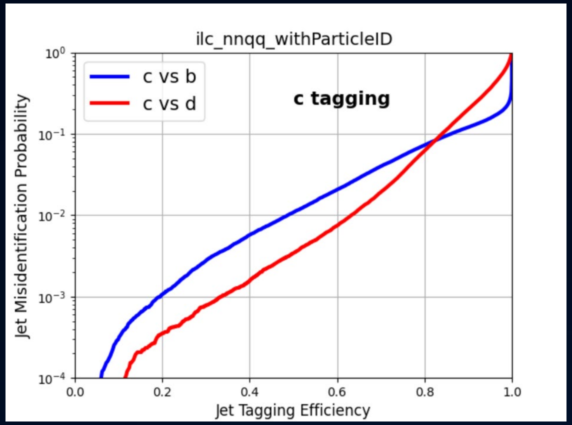


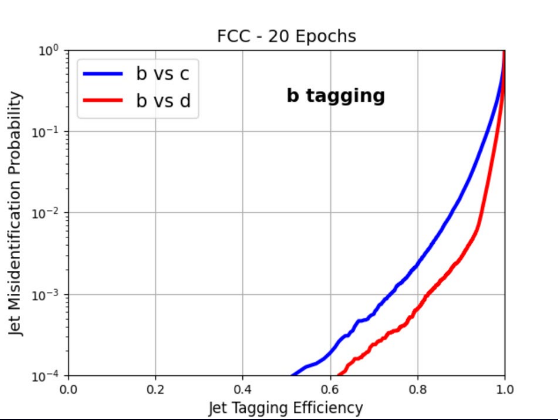
	b-tag 80% eff.		c-tag 50% eff.	
Method	c-bkg acceptance	uds-bkg acceptance	c-bkg acceptance	uds-bkg acceptance
LCFIPlus	10%	1%	10%	2%
ParT	1.29%	0.25%	1.02%	0.43%

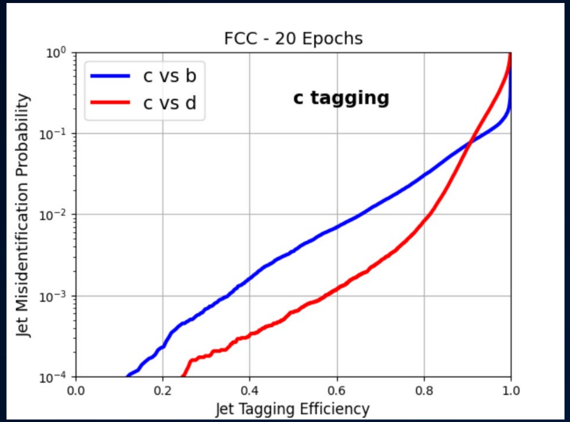
Comparison with FCC data[3]

- Trained with same condition as ILD data for fair comparison. (800k data size, 20 epochs, etc.)
- FCC data has ~ 3 times the performance compared to ILD data.
- Possible cause of the difference:
  - Particle ID: too pessimistic for ILD
  - Definition of some variables
    - Theta, phi etc.
  - Difference on full and fast sim
    - Especially different on tails of distributions
  - Assumed detector resolution (?)









Data	Particle ID	Impact Parameters	Jet Distance	Track Errors	c-bkg acceptance @ b-tag 80% eff.	b-bkg acceptance @ c-tag 50% eff.
ILD (vvqq 250 GeV)					0.64%	1.09%
FCC					0.23%	0.35%

ILD (vvqq 250 GeV) vs. FCC with partial variables

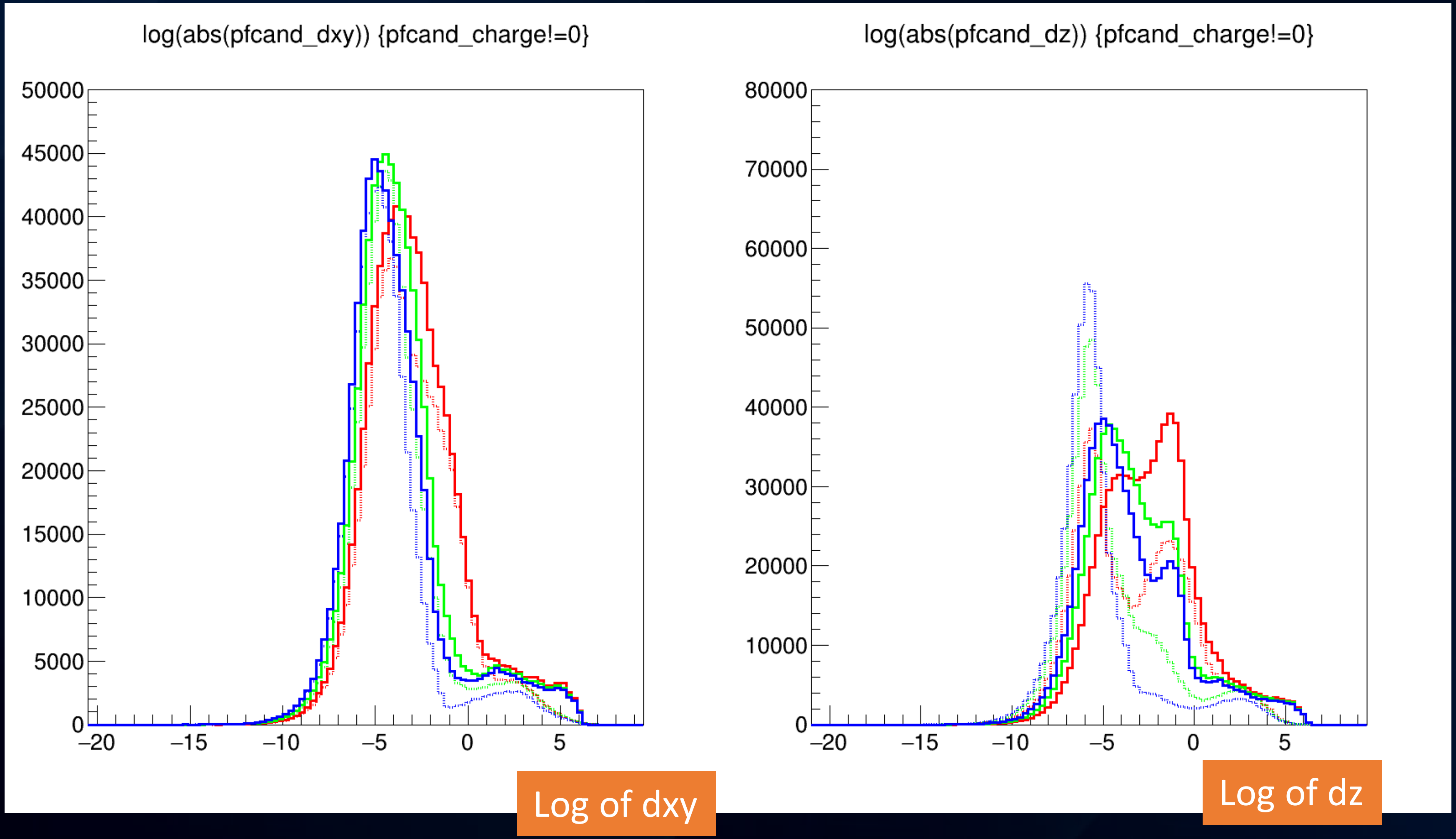
800 kjet for training, 20 epochs

					c-bkg acceptance @ b-tag 80% eff.		b-bkg acceptance @ c-tag 50% eff.	
Plot Index	Particle ID	Impact Parameters	Jet Distance	Track Errors	ILD	FCC	ILD	FCC
(1)	◍	◍	◍	◍	0.64%	0.23%	1.09%	0.35%
(2)	✕	◍	◍	◍	0.62%	0.47%	1.14%	0.64%
(3)	✕	◍	◍	✕	0.71%	0.24%	1.24%	0.35%
(4)	✕	◍	✕	◍	0.63%	0.75%	1.19%	0.80%
(5)	✕	◍	✕	✕	0.79%	0.77%	1.28%	0.80%
(6)	✕	✕	◍	◍	9.69%	2.64%	6.91%	1.58%

Observations:

1. PID gives significant effect on FCCee, not ILD (due to easy PID in ILD)
2. Track errors are rather harmful in FCCee
3. Difference on b-tag is small with only impact parameters (5), but still see difference in c-tag
4. (of course) significantly losing performance without impact parameter (but still ~ LCFIPlus)

Difference in impact parameters

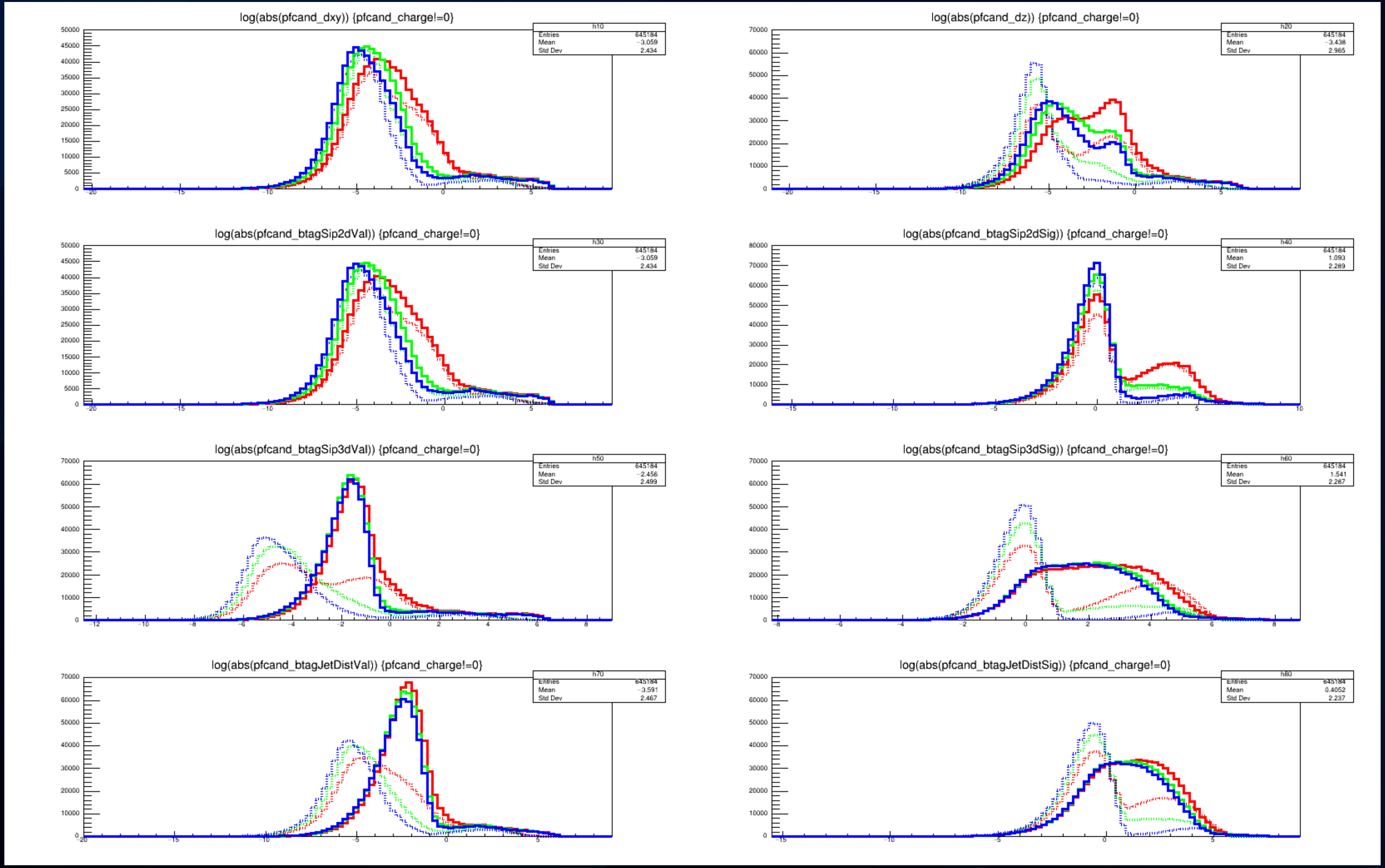


Dotted – FCCee
Solid – ILD

Red – nnbb
Green – nncc
Blue – nndd

Significant difference on dz seen
- beam spot smearing?

Difference in impact parameters



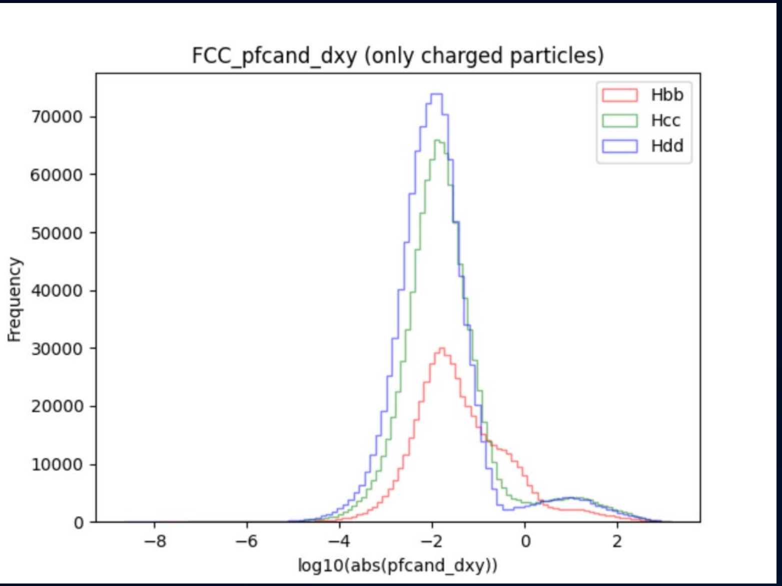
Dotted – FCCee
Solid – ILD

Red – nnbb
Green – nncc
Blue – nndd

Significant difference on dz seen
- beam spot smearing?

Potential Improvement: log(abs)

Particle ID	Impact Parameters	Jet Distance	Track Errors	c-bkg acceptance @ b-tag 80% eff.	b-bkg acceptance @ c-tag 50% eff.
✗	◍	◍	◍	**0.62%**	**1.14%**
✗	◍ +log(abs)	◍ +log(abs)	◍ +log(abs)	**0.54%**	**1.06%**
✗	◍	◍ +log(abs)	◍ +log(abs)	**0.79%**	**1.33%**
✗	◍	◍ +log(abs)	◍	**0.78%**	**1.36%**
✗	◍ +log(abs)	◍	◍	**0.47%**	**1.03%**
✗	log(abs)	log(abs)	log(abs)	**0.82%**	**1.32%**
✗	◍	log(abs)	log(abs)	**0.80%**	**1.37%**
✗	◍	◍	log(abs)	**0.82%**	**1.38%**



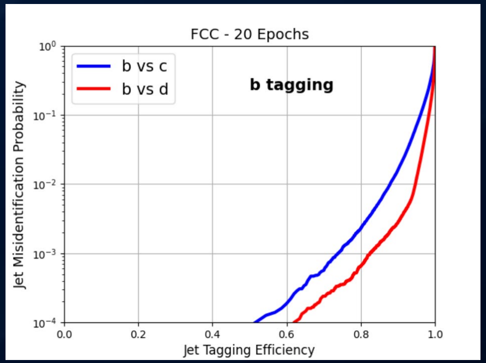
Impact Parameter

ML prefers “gaussian-like” distribution
Not sensitive to small values
(because of linear weighting)

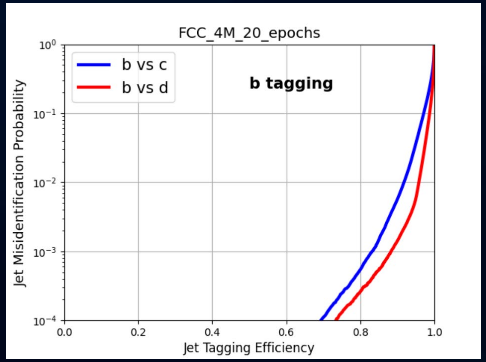
Track errors or impact parameters should convert with e.g. log function
→ slightly improving performance
(but not much as expected…)

Sample size affects performance (FCCee sample)

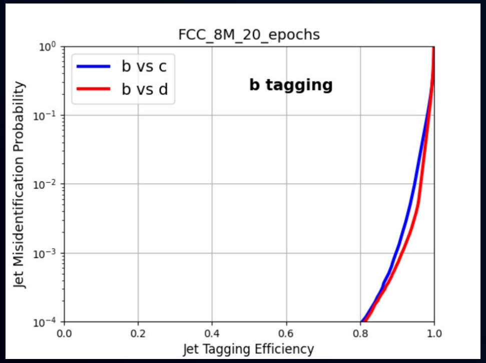
(1)



(2)



(3)



Plot Index	Particle ID	Impact Parameters	Jet Distance	Track Errors	Training Sample size	c-bkg acceptance @ b-tag 80% eff.	b-bkg acceptance @ c-tag 50% eff.
(1)	◍	◍	◍	◍	800k	0.23%	0.35%
(2)	◍	◍	◍	◍	4M	0.054%	0.20%
(3)	◍	◍	◍	◍	8M	Unreasonably good, TBC	

- Training performance significantly improved with bigger data sample size
- Training sample size change of FCC data:

  800k -> 4M : 4 times better performance (b-tagging)

  4M -> 8M: 5 times better performance (b-tagging)
- This non-linearity of increase in performance should be further investigated.
- Bigger data size of ILD should be obtained for better performance, as well as comparison with FCC data for further investigation on its behaviour.

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

Things to do

1. Share the data with FCC people
2. Confirm the difference in more detail
   (especially on sample size)
3. Include better particle ID on ILD based on recent PID developments
4. Optimizing input parameters
   (should be agreed with FCCee for fair comparison)
5. Trying fast simulation of ILD (SGV) and try to use for pretraining
   (alternatively prepare 10 M jets with full simulation)
6. Strange tagging – including π/K/p separation variables
7. Preparing inference procedure to be used for physics analyses
   (cooperation with software group essential)

Summary of all

- DNN-based PFA and flavor tagging are being investigated
- For PFA:
  - First implementation of track-cluster matching done, need more tuning for comparison with existing PandoraPFA
  - To be done in ~a half year
- For flavor tagging:
  - ParT based flavor tagging gives ~10x better performance than LCFIPlus → need to replace
  - Optimization still to be done
  - Incorporation to analysis framework desired
  - Fine-tuning is powerful: to investigate how to use it for analysis

Summary / long-term plans

- New DNN-based particle flow algorithm is under development based on clustering at CMS HGCAL study
- Track-cluster matching is being implemented, statistical results will come soon
  - Energy regression with track momentum information will be the next step of implementation
- Medium/long term plans (or just hopes)
  - Can be extended to any analyses using cluster/jet information using the PFA as "a foundation model"
    - Such as Particle ID, Jet clustering, even physics analyses directly
  - "Differentiate" detector parameters/designs for optimization

Summary

- Particle Transformer seems very promising in quark flavour tagging.
- Its performance can be further improved by adjusting the input parameters.
- Bigger data set is required for better training outcomes.
- Fine-tuning is effective with the model, but only for similar data setups.
- It’s maybe time to start thinking of how to apply to physics analyses.
- Its application on other reconstruction algorithms should be explored.