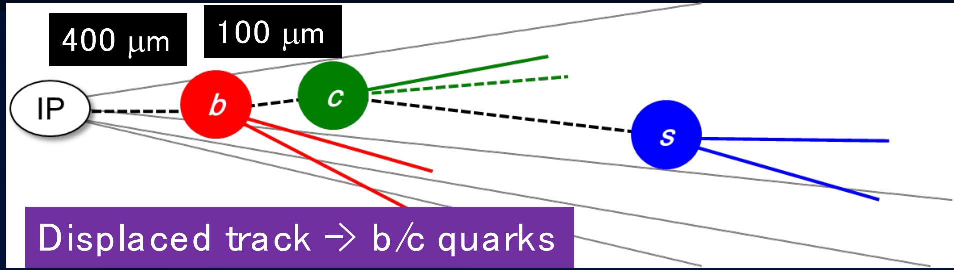
Application of Particle Transformer for Quark Flavor Tagging on Future Higgs Factories

Taikan Suehara (ICEPP, U. Tokyo),
Lai Gui (Summer student at Kyushu, from Imperial College London)

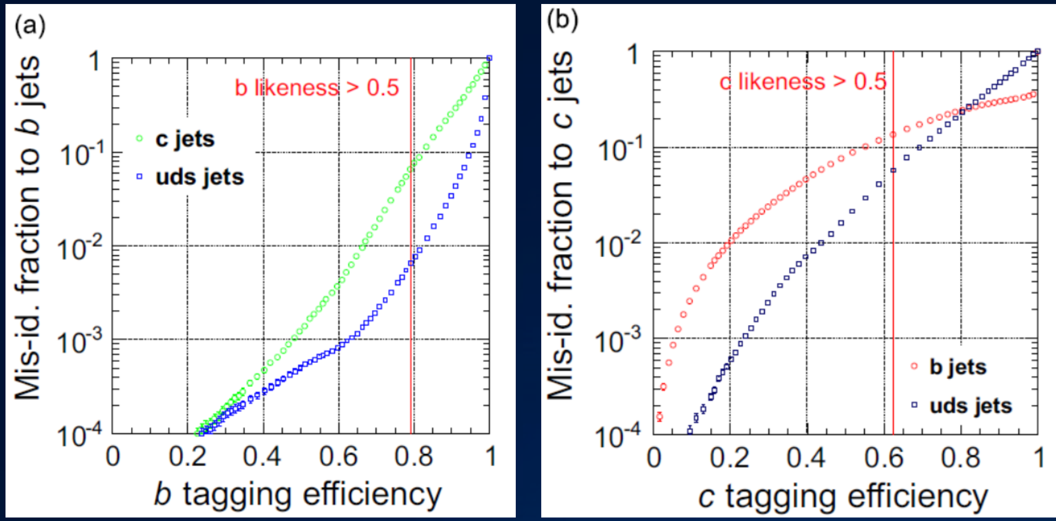
All results are preliminary: need to check reproducibility with shuffled events etc. (TBD)

Flavor tagging for Higgs factories

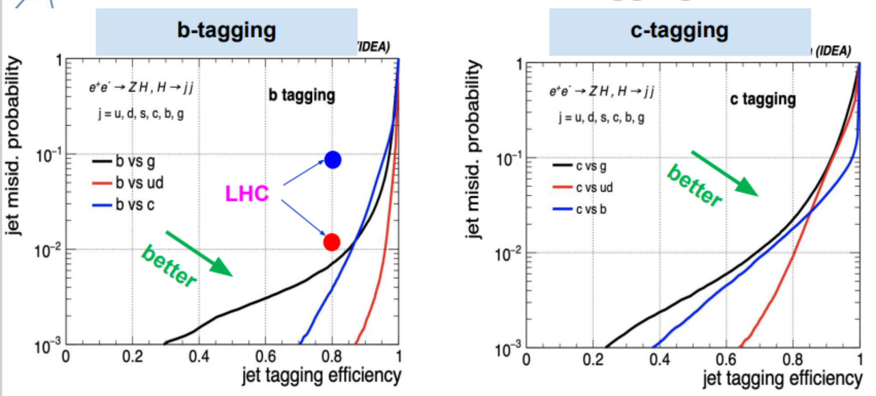


- Jet flavor tagging is essentially important for Higgs studies (including self coupling)
- LCFIPlus (published 2013)[1] was long used for flavor tagging
  - b-tag: ~80% eff., 10% c / 1% uds acceptance;
  - c-tag: ~50% eff., 10% b / 2% uds acceptance.
- Recently FCCee reported ~10x better rejection using 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

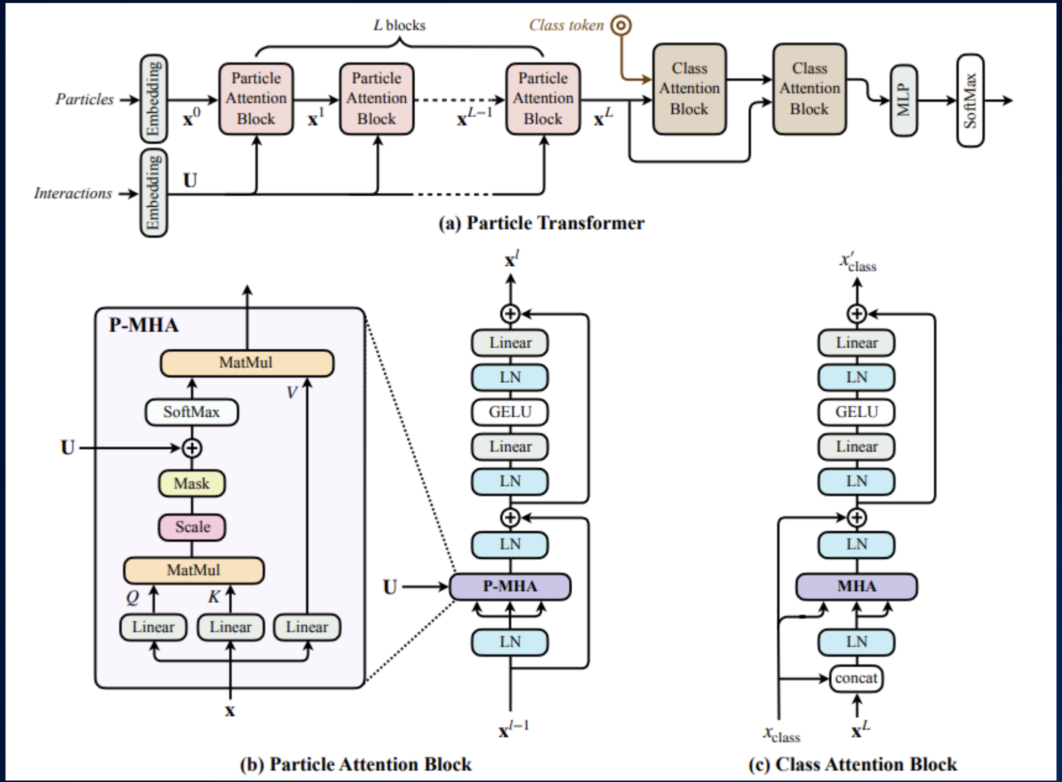


WP	Eff (b)	Mistag (g)	Mistag (ud)	Mistag (c)
Loose	90%	2%	0.1%	2%
Medium	80%	0.7%	<0.1%	0.3%

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	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{99\%}$	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{99.5\%}$	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{50\%}$
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

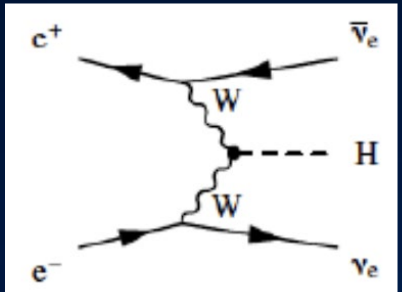
Data Used For Investigation

- 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

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

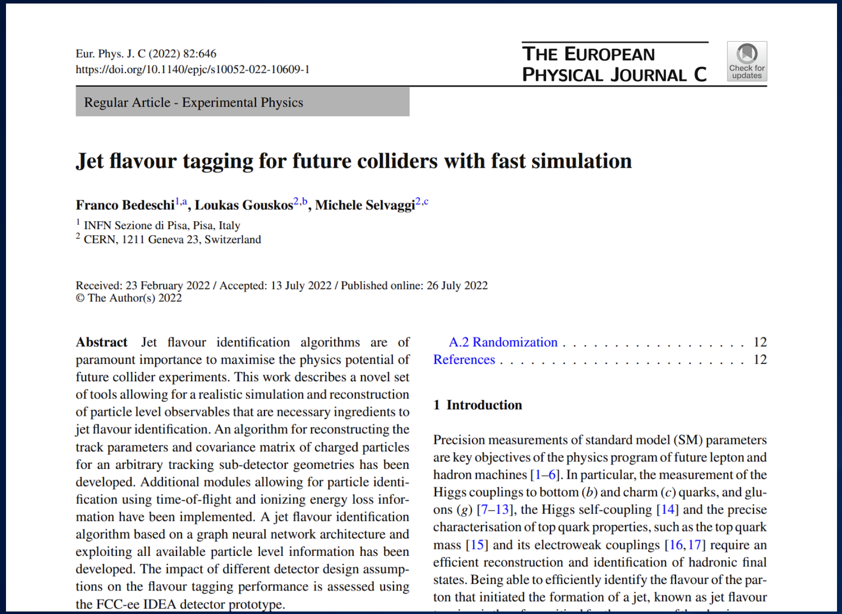


- FCCee fast simulation (Delphes with IDEA detector):

  e+ e- ⟶ ννH ⟶ ννqq (at 240 GeV)

  With 10M jets (5M events) each

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



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

THE EUROPEAN PHYSICAL JOURNAL C

Check for updates

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

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

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

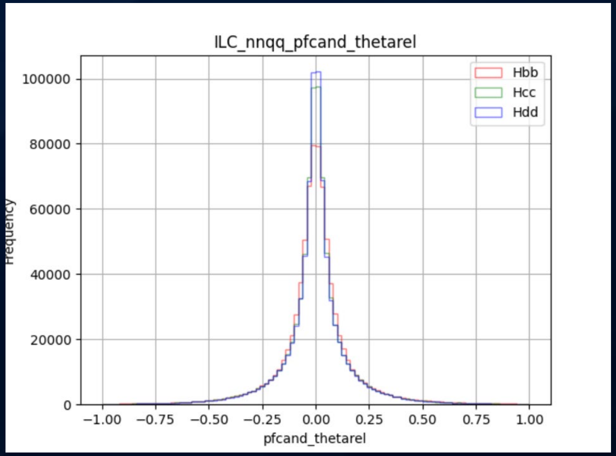
*Fraction of the particle energy wrt. jet energy (log is taken)

- Track Errors (15):

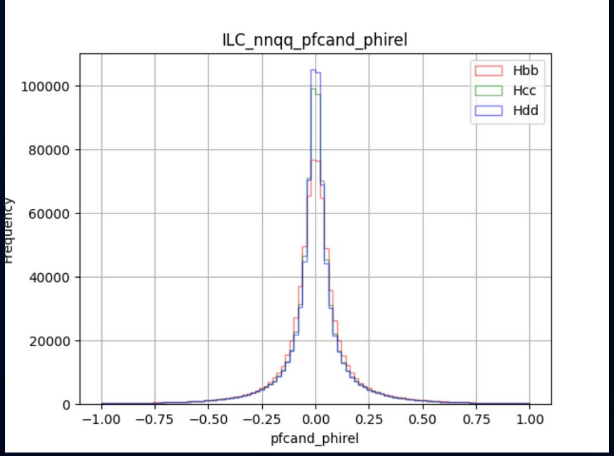
pfcand_dptdpt
pfcand_detadeta
pfcand_dphidphi
pfcand_dxydxy
pfcand_dzdz
pfcand_dxydz
pfcand_dphidxy
pfcand_dlambdadz
pfcand_dxyc
pfcand_dxyctgtheta
pfcand_phic
pfcand_phidz
pfcand_phictgtheta
pfcand_cdz
pfcand_cctgtheta

*each element of covariant matrix
0 for neutrals

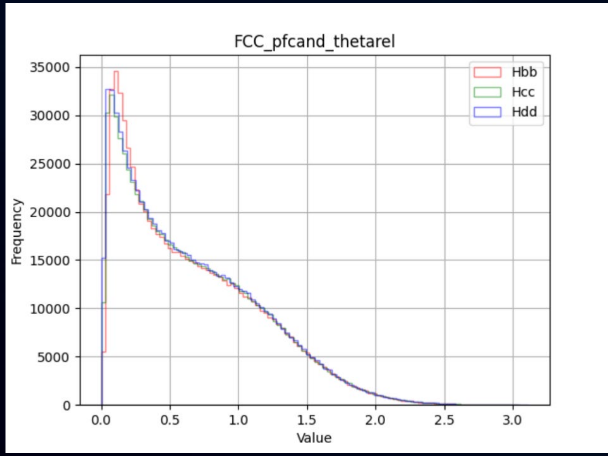
ILD vs. FCC – theta/phi distribution



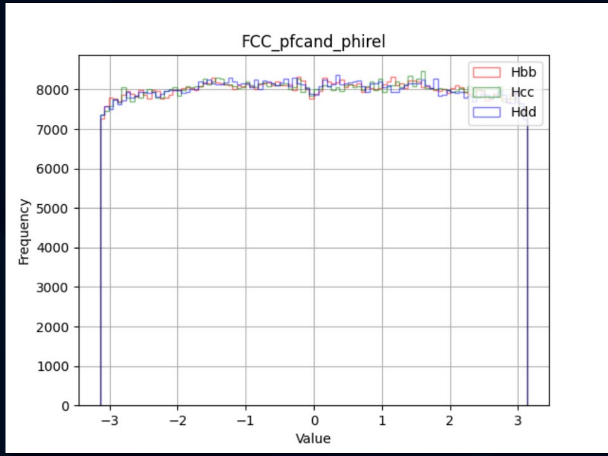
ILD theta



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.

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:

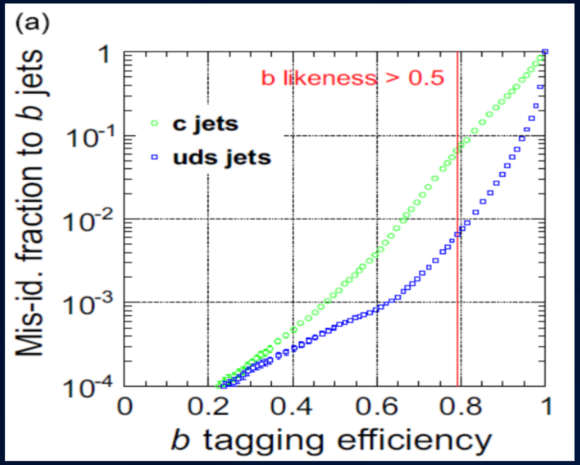
  - pfcand_px
  - pfcand_py
  - pfcand_pz

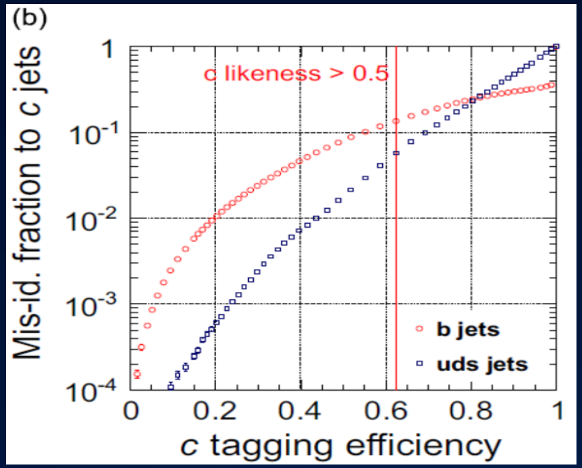
- But it's possible to transfer ILD's interaction to FCC's form for fair comparison:

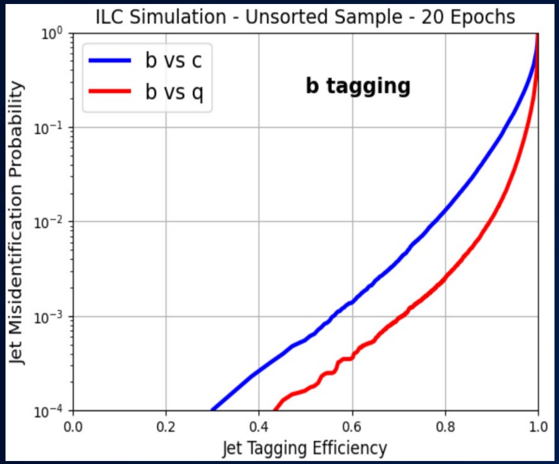
$$p = \sqrt{{p_x}^2 + {p_y}^2 + {p_z}^2}$$

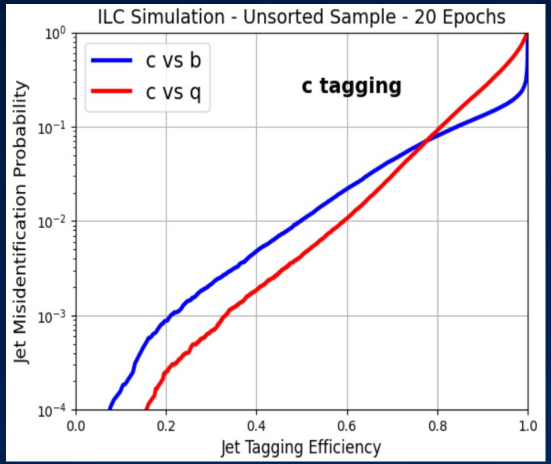
Application of ParT to ILD data
(ILD qq 91 GeV, 0.8M jets for training)

- 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





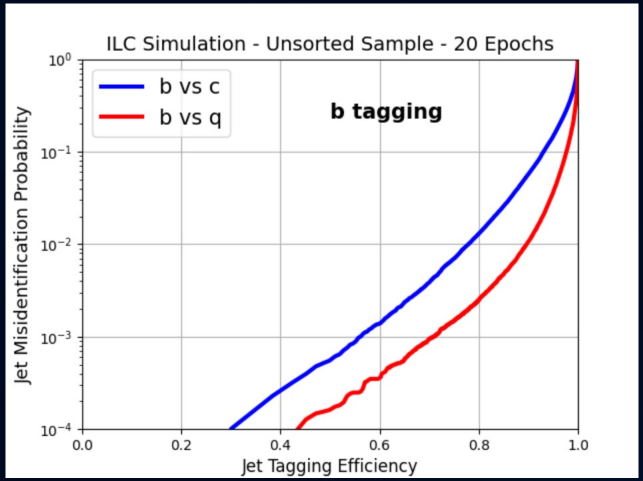


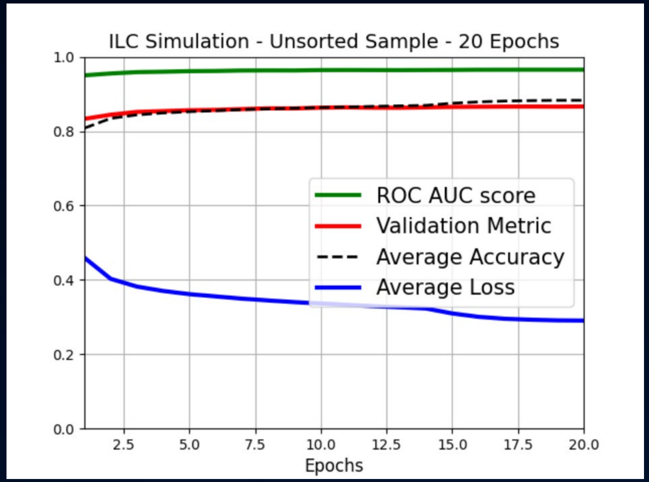


	b-tag 80% eff.		c-tag 50% eff.	
Method	c-bkg acceptance	uds-bkg acceptance	c-bkg acceptance	uds-bkg acceptance
LCFIPlus	10%	1%	10%	2%
ParT	1.29%	0.25%	1.02%	0.43%

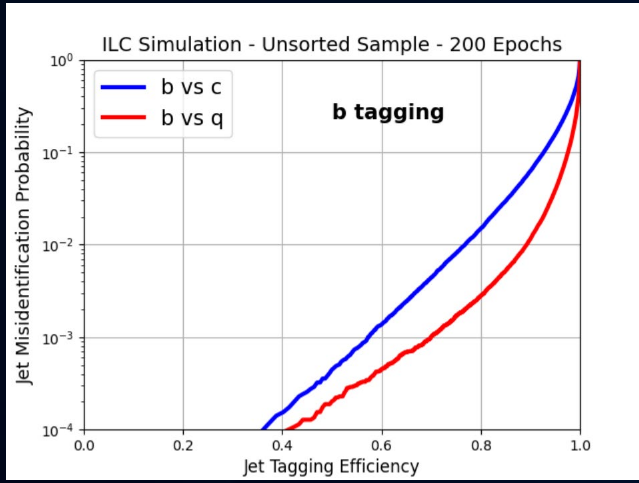
Training parameters - epochs

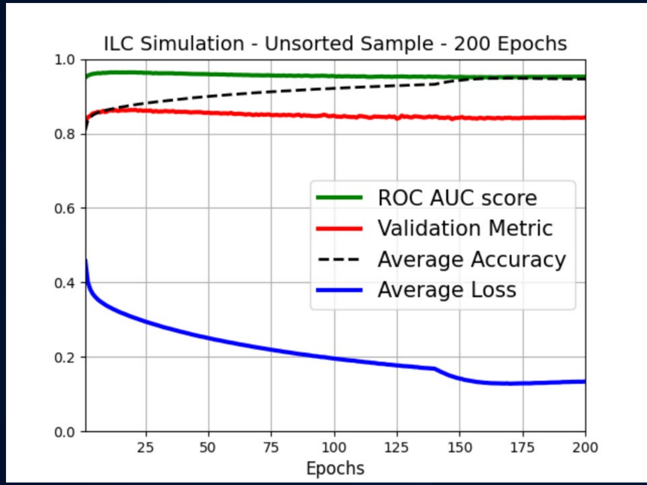
- Run on NVIDIA TITAN RTX (memory: 24 GB)
  - 20 Epochs: 3 hours
  - 200 Epochs: 30 hours
- No significant improvement in tagging efficiency
- Both ROC AUC score and Validation Metric reaches a maximum around 20 epochs.
- Overtraining after 20 epochs.
- Hence 20 epochs of training is selected to avoid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





20 epochs (ILD qq 91 GeV)

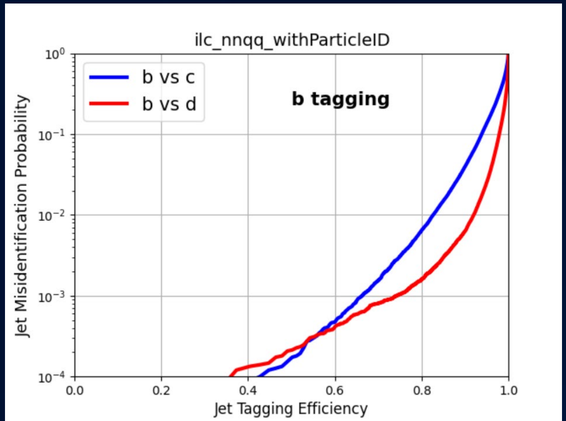


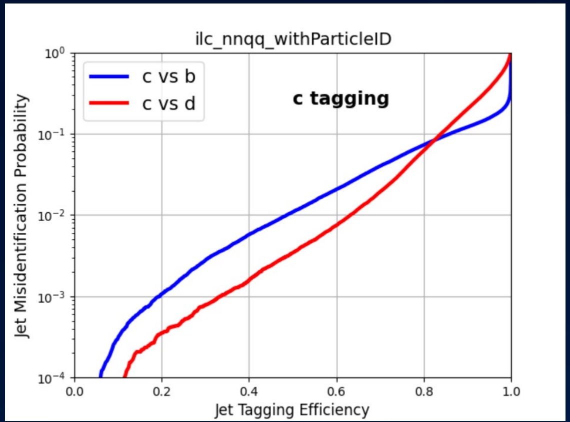


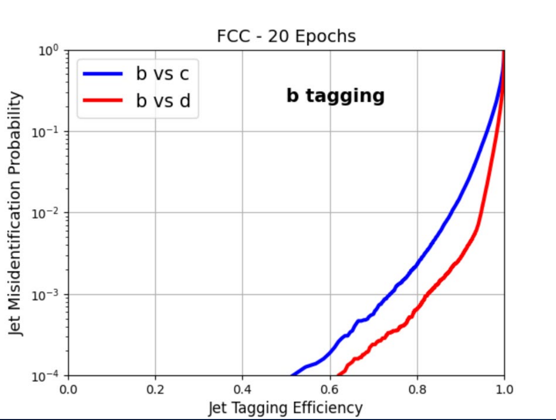
200 epochs (ILD qq 91 GeV)

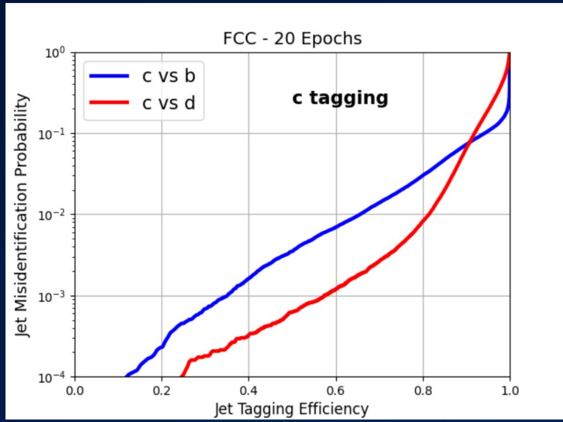
Comparison with FCC data[3]

- 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 (?)





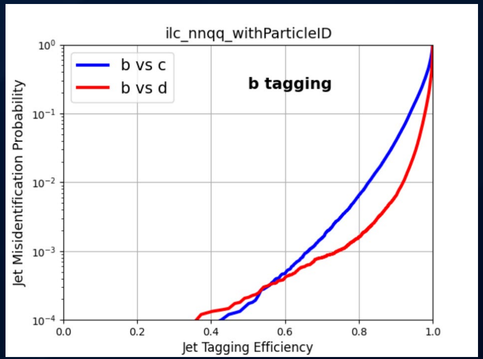




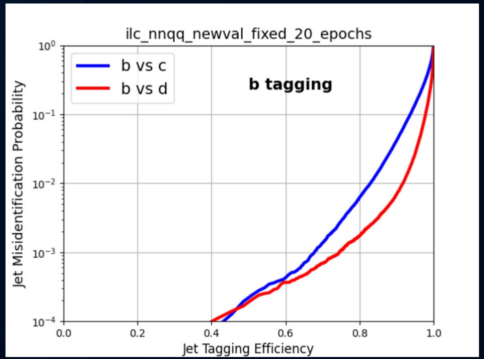
Data	Particle ID	Impact Parameters	Jet Distance	Track Errors	c-bkg acceptance @ b-tag 80% eff.	b-bkg acceptance @ c-tag 50% eff.
ILD (vvqq 250 GeV)	*	*	*	*	0.64%	1.09%
FCC	*	*	*	*	0.23%	0.35%

Effect of different parameters: ILD (vvqq 250 GeV)

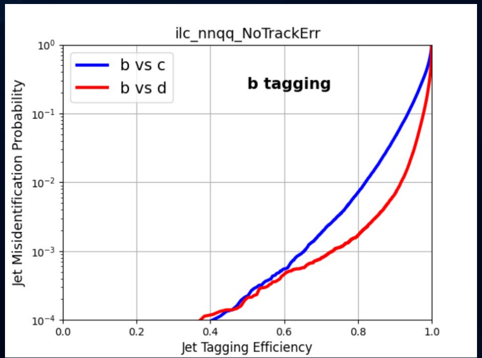
(1)



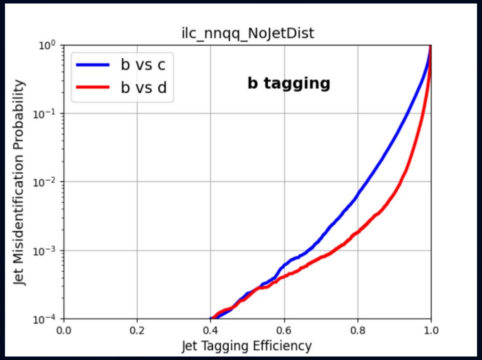
(2)



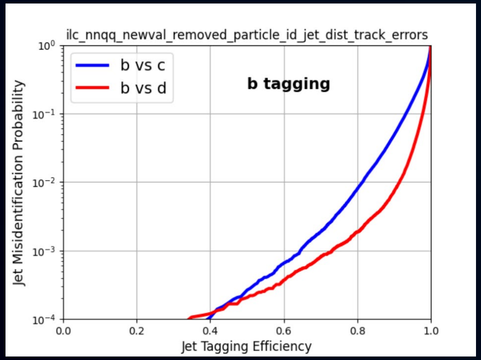
(3)



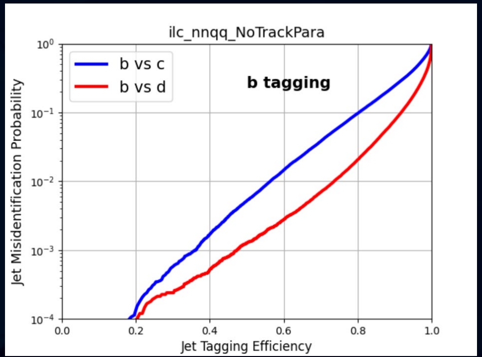
(4)



(5)



(6)

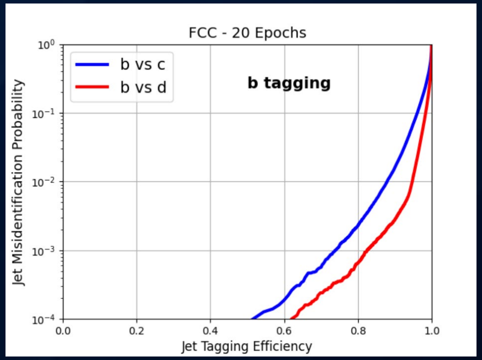


Plot Index	Particle ID	Impact Parameters	Jet Distance	Track Errors	c-bkg acceptance @ b-tag 80% eff.	b-bkg acceptance @ c-tag 50% eff.
(1)	*	*	*	*	0.64%	1.09%
(2)	-	*	*	*	0.62%	1.14%
(3)	-	*	*	-	0.71%	1.24%
(4)	-	*	-	*	0.63%	1.19%
(5)	-	*	-	-	0.79%	1.28%
(6)	-	-	*	*	9.69%	6.91%

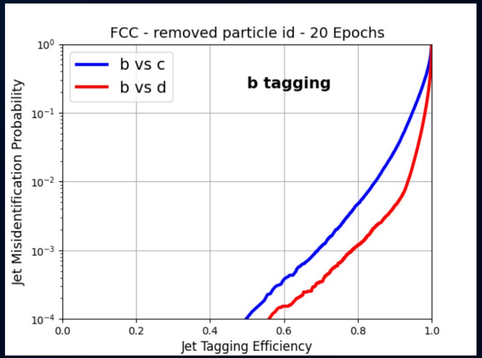
- Impact parameter gives most significance in affecting the training performance.
- The other parameters are about the similar significance (not significant impact).

Effect of different parameters: FCC

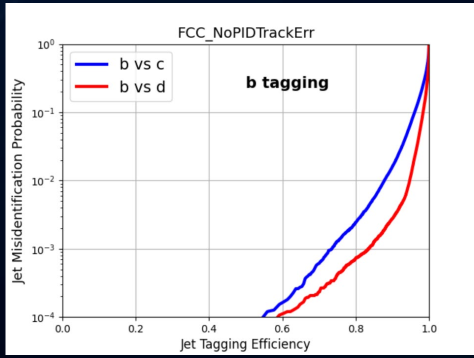
(1)



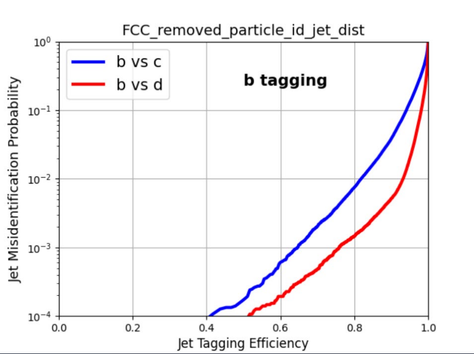
(2)



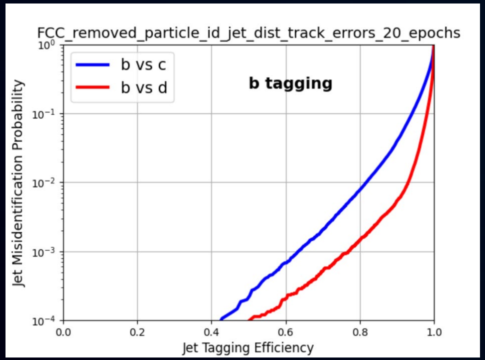
(3)



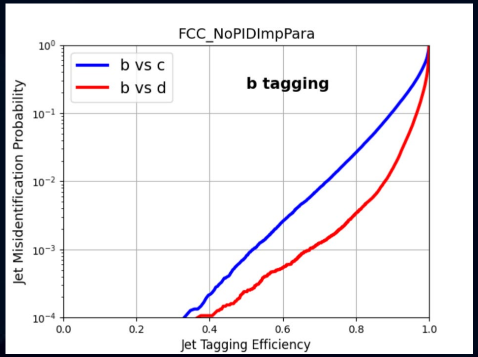
(4)



(5)



(6)



Plot Index	Particle ID	Impact Parameters	Jet Distance	Track Errors	c-bkg acceptance @ b-tag 80% eff.	b-bkg acceptance @ c-tag 50% eff.
(1)	*	*	*	*	0.23%	0.35%
(2)	-	*	*	*	0.47%	0.64%
(3)	-	*	*	-	0.24%	0.35%
(4)	-	*	-	*	0.75%	0.80%
(5)	-	*	-	-	0.77%	0.80%
(6)	-	-	*	*	2.64%	1.58%

- Effect of Impact Parameters also significant.
- Both Particle ID and Jet Distance give significant impacts.
- Removal of track errors improves performance, could be a result of too many variables of Track Errors (15) shifting away the contribution of others. Further investigation should be conducted.

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

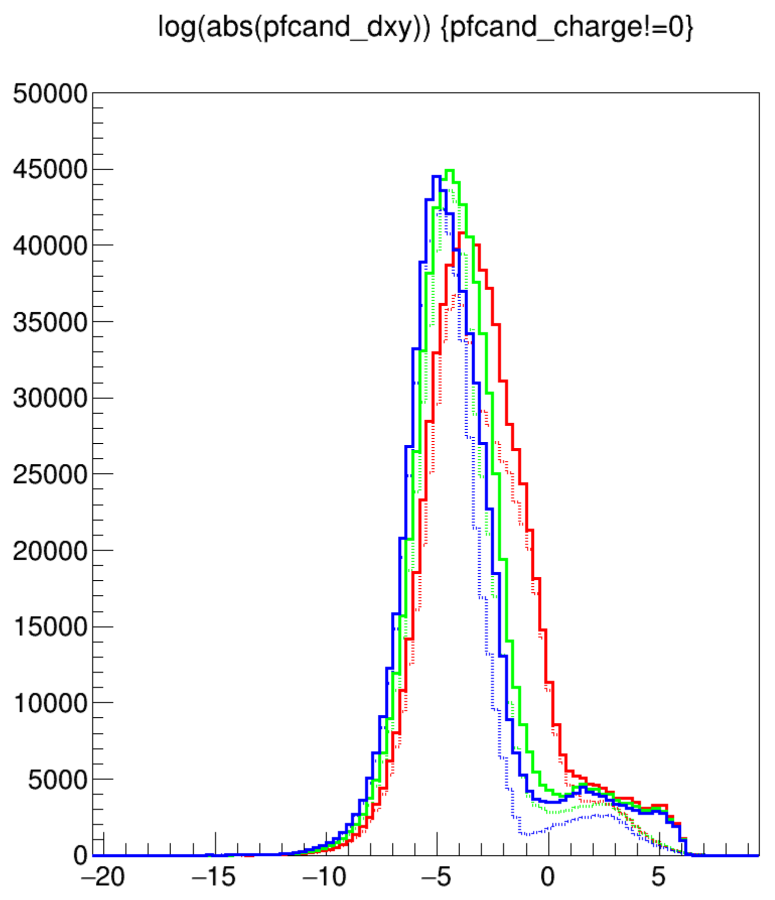
800 kjet for training, 20 epochs

					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%

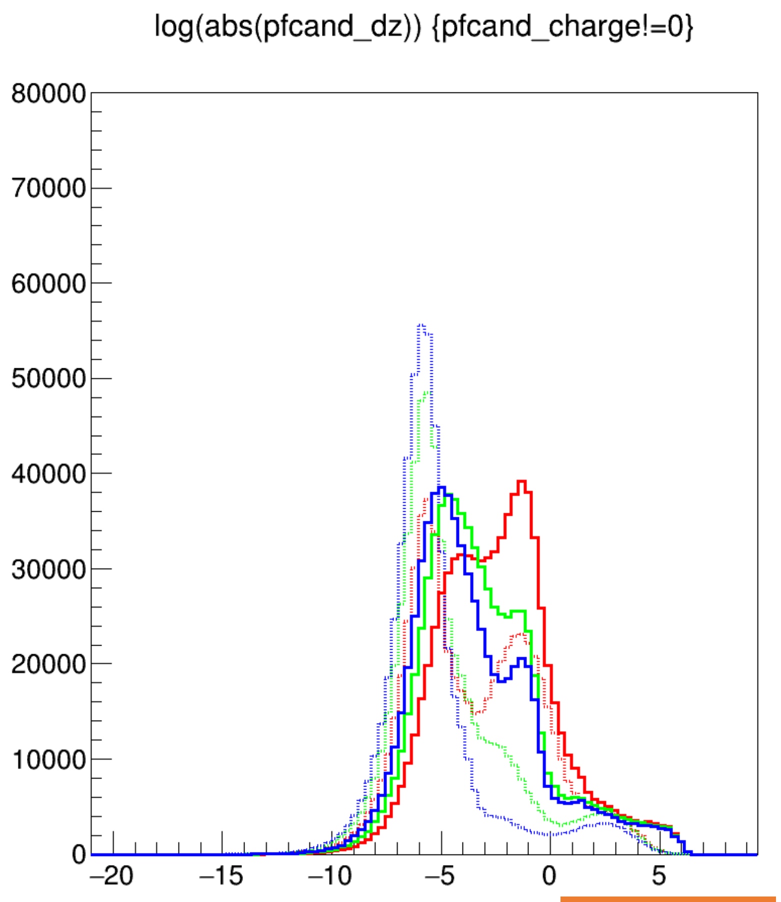
Observations:

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



Log of dxy



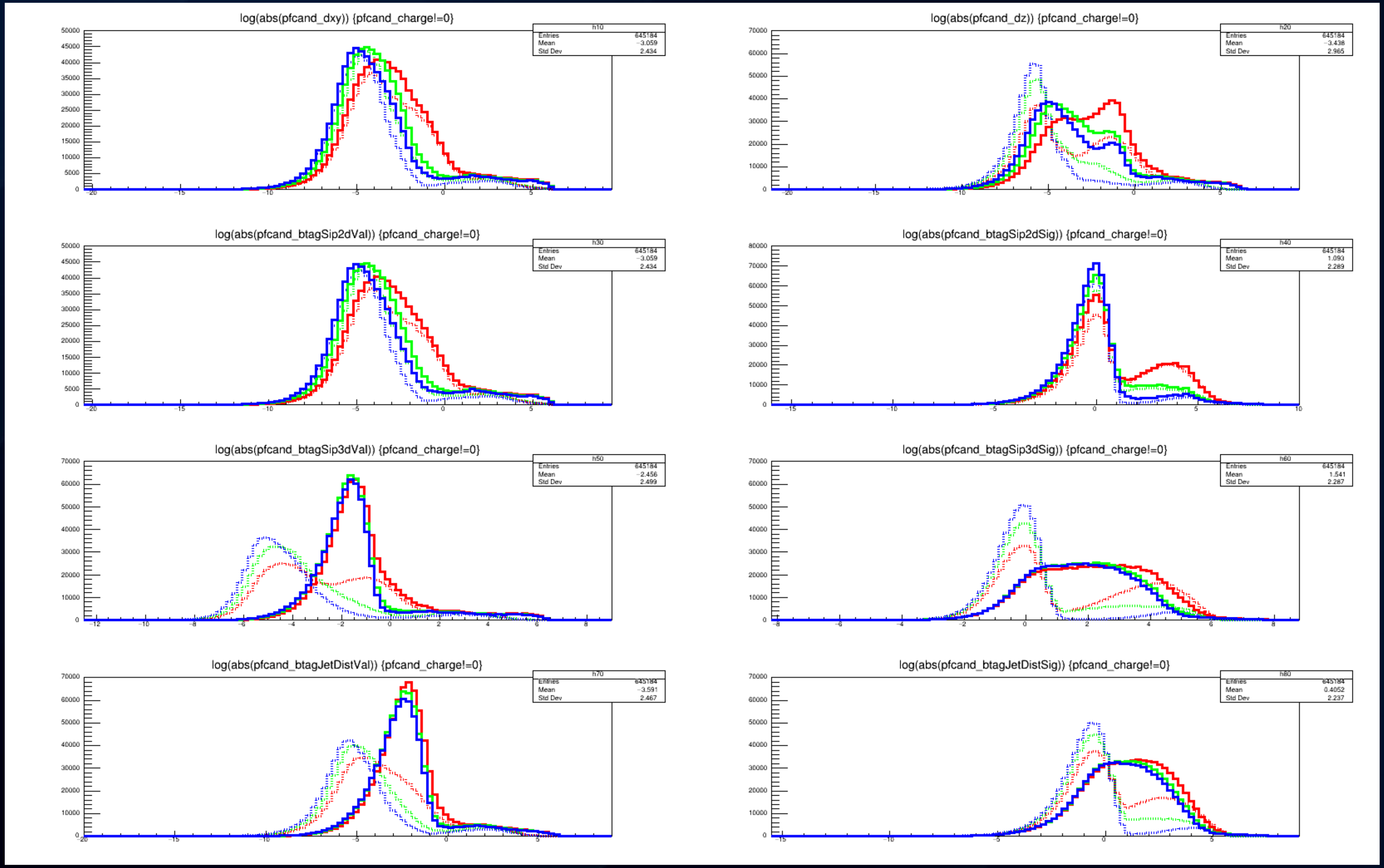
Log of dz

Dotted – FCCee
Solid – ILD

Red – nnbb
Green – nncc
Blue – nndd

Significant difference on dz seen
- beam spot smearing?

Difference in impact parameters

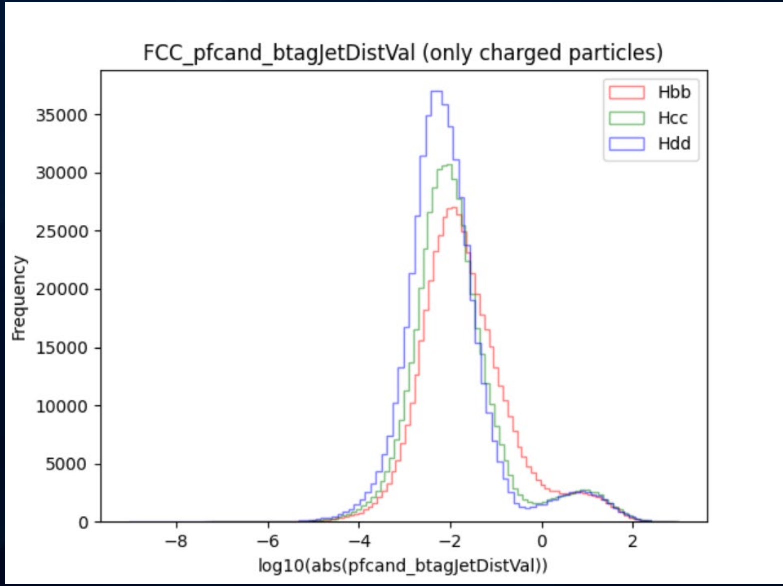


Dotted – FCCee
Solid – ILD

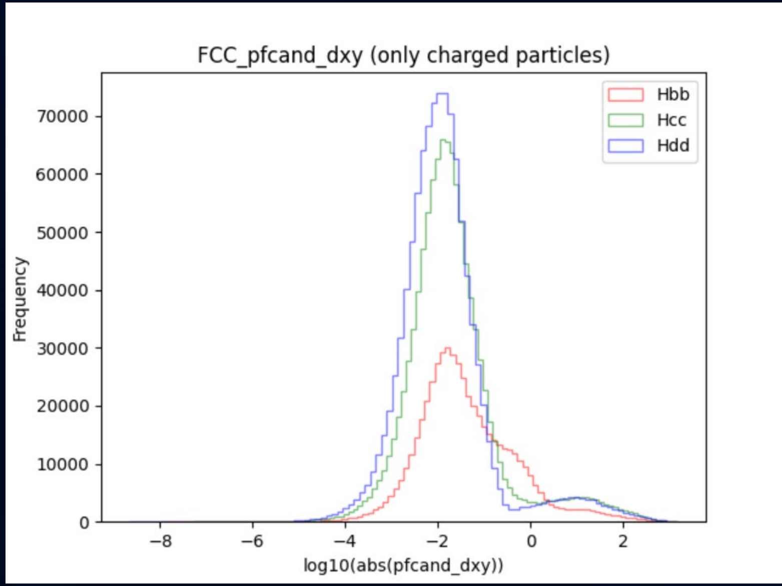
Red – nnbb
Green – nncc
Blue – nndd

Significant difference on dz seen
- beam spot smearing?

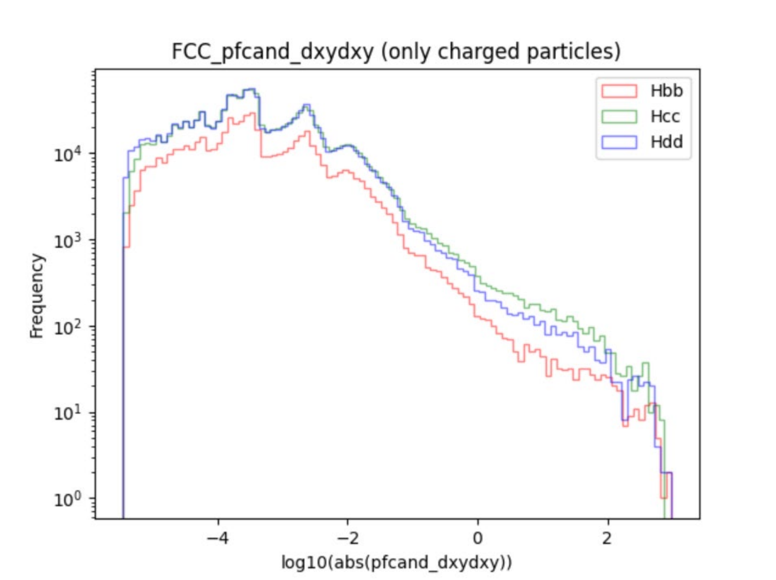
Potential Improvement: log(abs)



Jet Distance



Impact Parameter

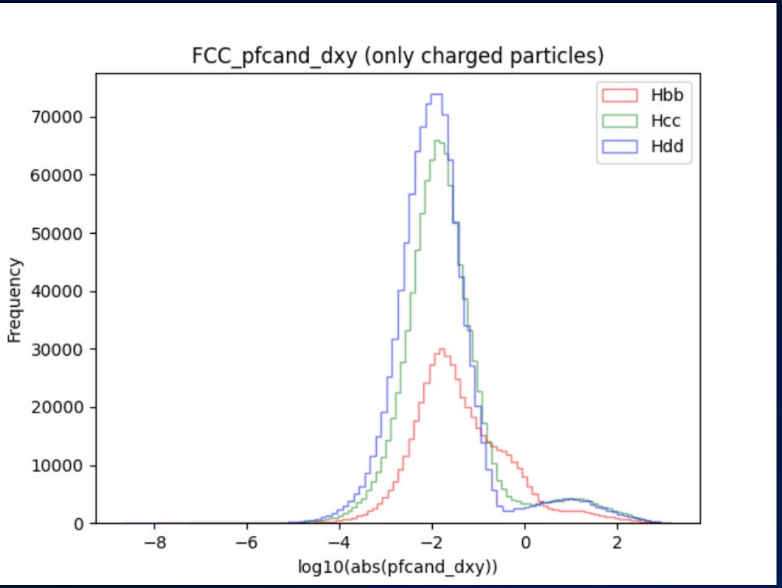


Track Errors

- Some example distribution of log(abs) the three parameters
- All very small (largely gathering around 10^{-2})
- Hence log(abs) potentially spreads out the distribution and make it more readable by the architecture
- Can potentially improve the performance?

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…)

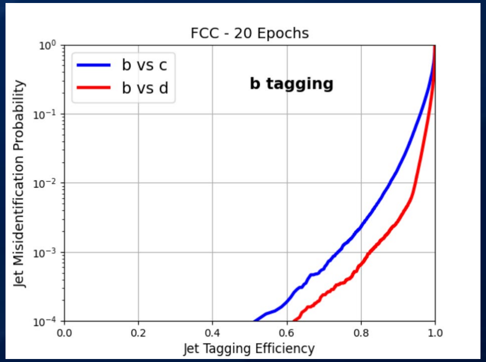
Use px, py, pz instead of p (Interaction)

				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%

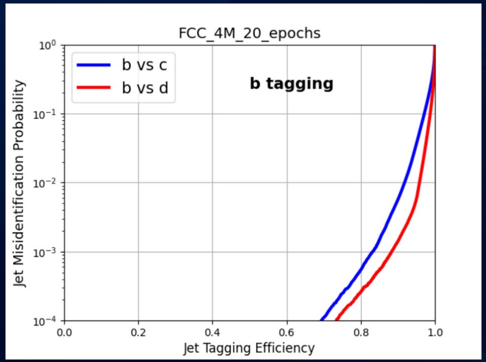
- ILD (ννqq 250 GeV) data shows that application of px, py, pz has better performance than p.
- However, application of log(abs) of the parameters becomes less significant.
- Can be because that application of px, py, pz changes the way log(abs) interacts with other parameters.
- Other potential treatments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

Sample size affects performance (FCCee sample)

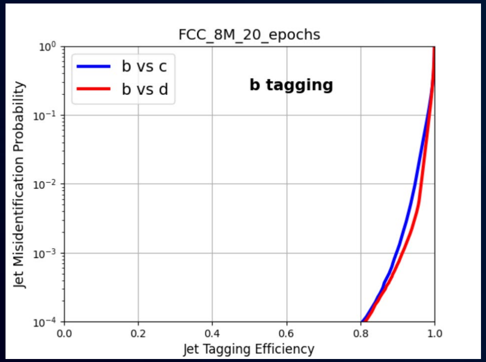
(1)



(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

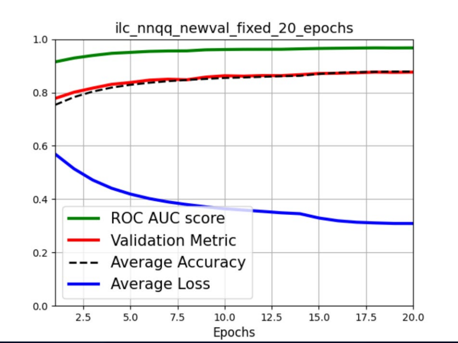
Two objectives
- Pretrained with fast sim and fine-tune with full sim
- Pretrained with large central production and fine-tune with dedicated physics samples in each analysis

							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%

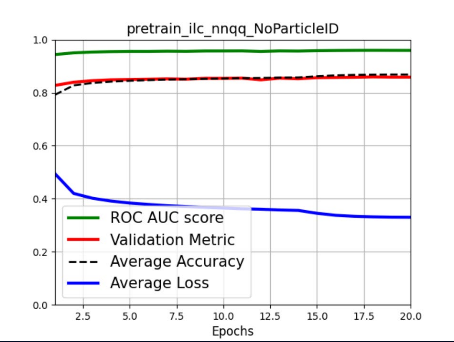
- Use result of 8M FCC data to train ILD 800k data
- Improves performance only when setups are similar
- Training of same setup (pretrain ILD 91 GeV data with ILD 250 GeV data) gives best performance
- Further investigation should be conducted on how to maximise the outcome for fine-tuning between different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

Fine tuning – Training curves

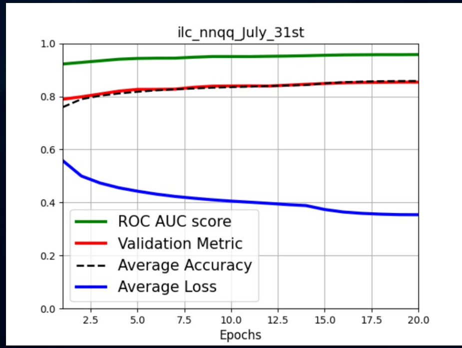
(1)



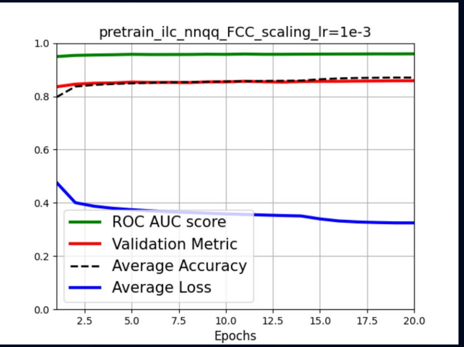
(2)



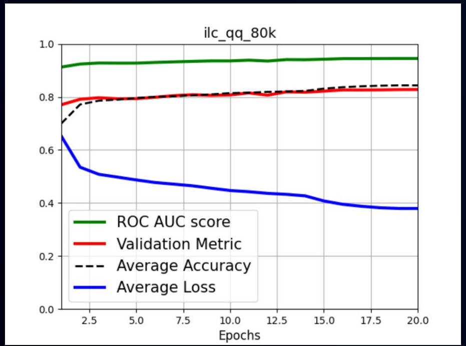
(3)



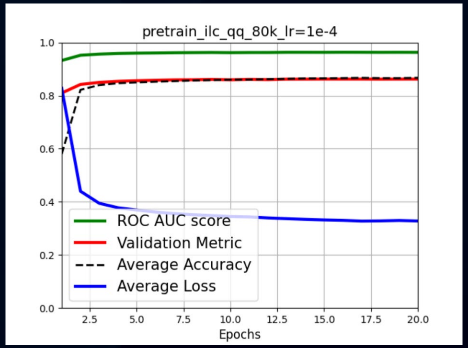
(4)



(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

Things we are working on / plan to do

1. Share the data with FCC people – where to upload?
2. Confirm uncertainty of training and sample
   - With individual training of the same sample
   - Shuffling training/validation/test samples
3. Optimizing input parameters (transformation of variables etc.)
   - should be agreed with FCCee for fair comparison
4. Trying fast simulation of ILD (SGV) and try to use for pretraining
   (alternatively prepare 10 M jets with full simulation)
5. Include better particle ID on ILD based on recent PID developments
6. Strange tagging – including π/K/p separation variables
7. Preparing inference procedure to be used for physics analyses
   (cooperation with software group essential)
8. Try similar but different structure like plain Transformer, Graphormer etc.

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.

Reference List

[1] https://doi.org/10.1016/j.nima.2015.11.054

[2] https://arxiv.org/abs/2202.03772

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