

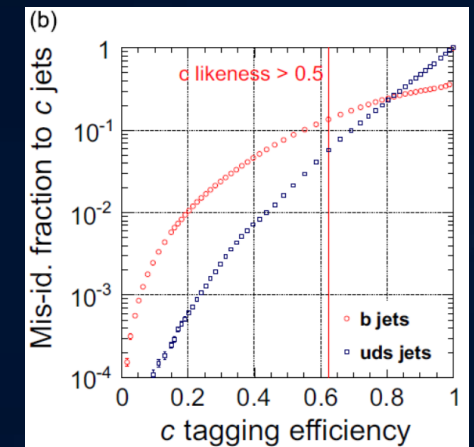
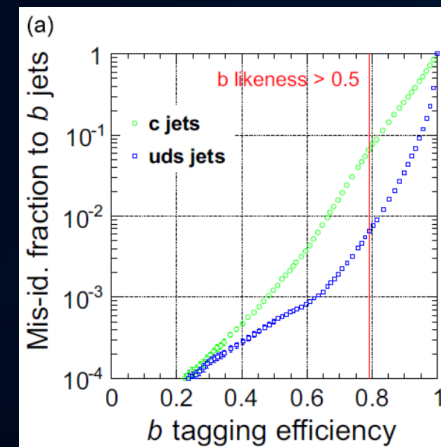
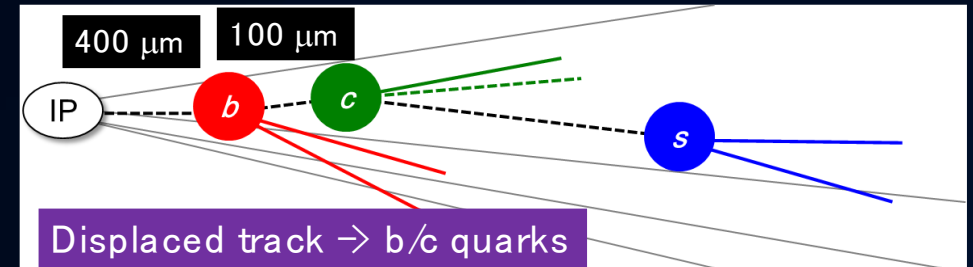
Application of Particle Transformer for Quark Flavor Tagging on Future Higgs Factories

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)

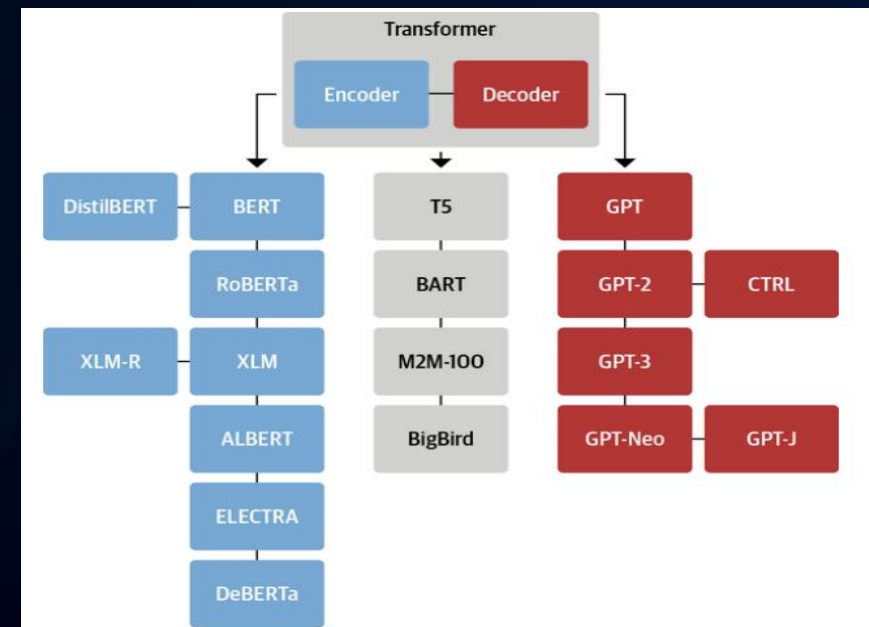
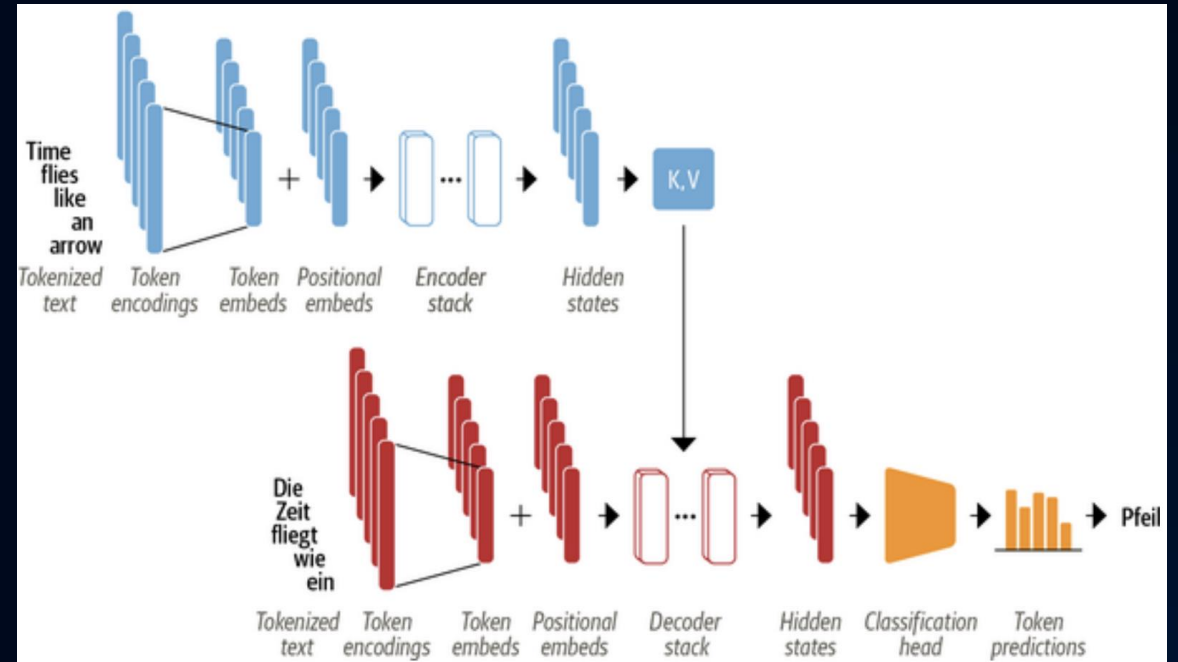
Background

- Precise measurements instrumentation and reconstruction software are essential for the ILC PROJECT.
- Various frameworks have been developed for jet flavor identification.
- LCFIPlus (published 2013)^[1] was successful in vertex finding, jet clustering and flavor tagging.
- Reached a reasonable performance of:
 - b-tag: 80% eff., 10% c / 1% uds acceptance;
 - c-tag: 50% eff., 10% b / 2% uds acceptance.

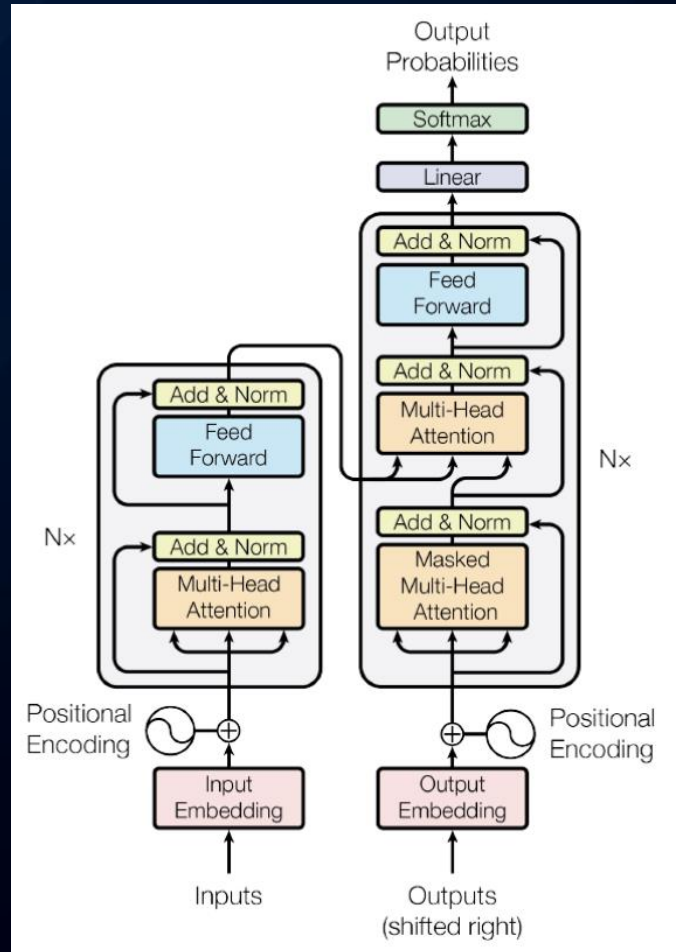


Transformer

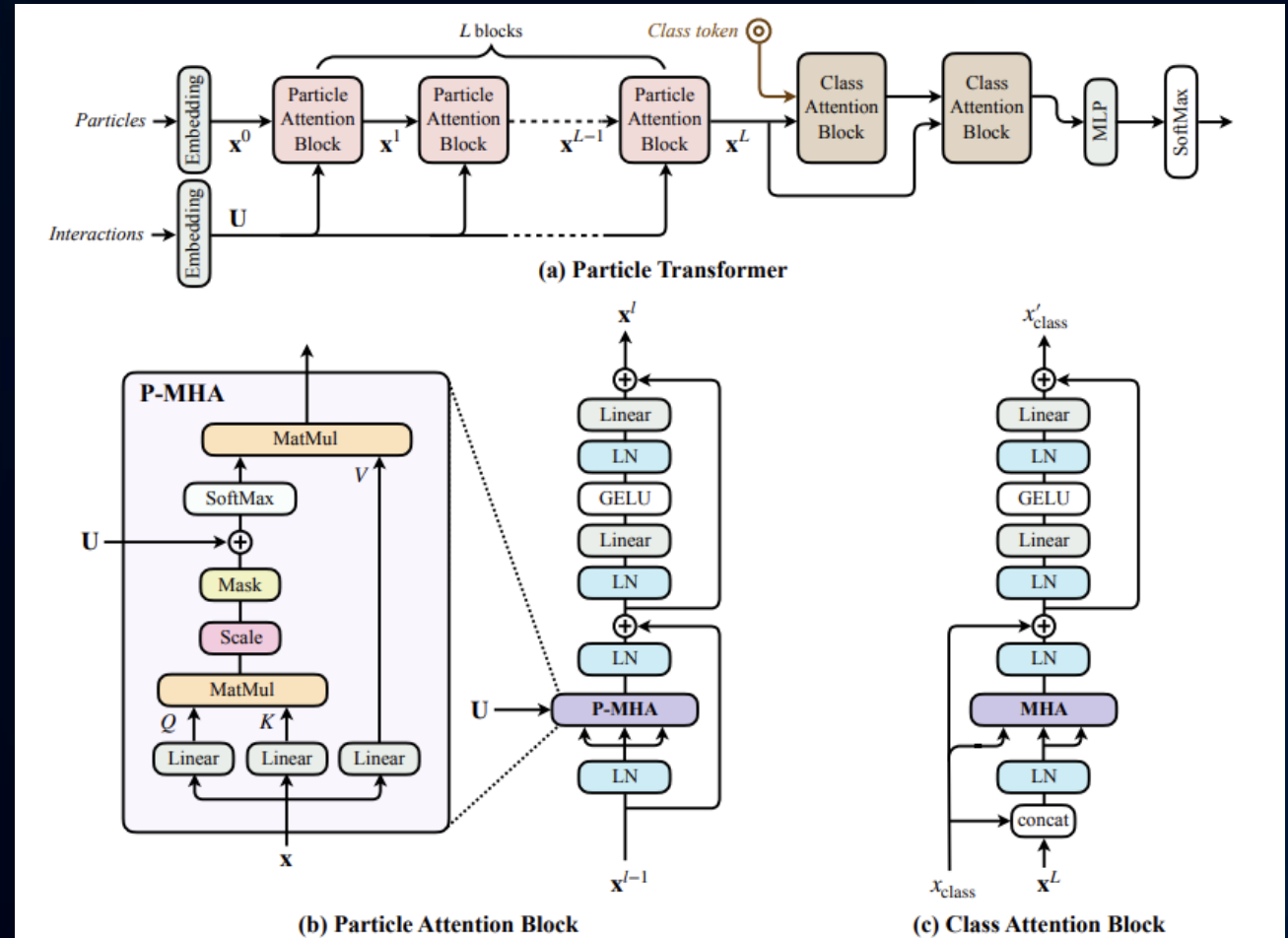
- Input is converted by the Encoder into a sequence of hidden states that is consisted of Token Embeds and Positional Embeds.
- This hidden state is then processed through layers of Self-Attention and Feed-Forward neural networks.
- The Self-Attention mechanism calculates the relative importance of each token relative to all the other tokens in the input sequence (Outperforms traditional RNN and CNN).
- The Decoder then outputs one token at a time, and this token is then added to the input to generate the next context iteratively.



Comparison between regular Transformer and Particle Transformer



Regular Transformer

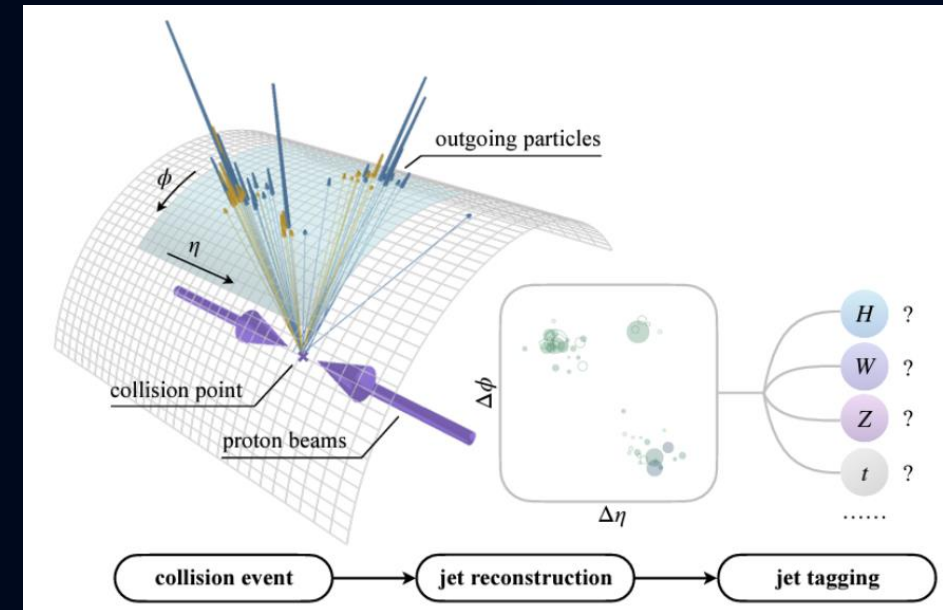


Particle Transformer

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

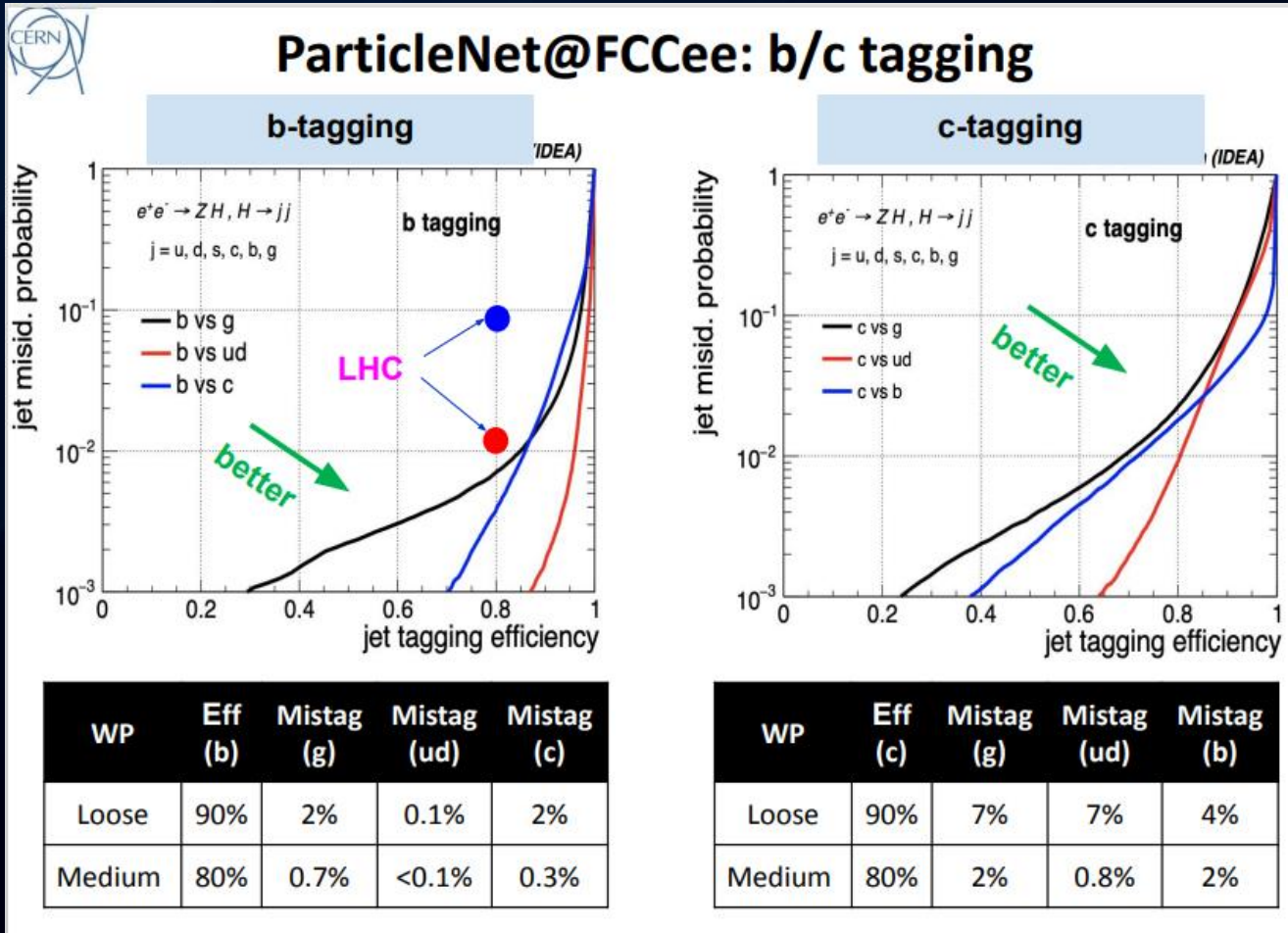
Particle Transformer (ParT)

- A new Transformer-based architecture for Jet tagging, published in 2022^[2].
- It analyses the readings collected after collision events to reconstruct jets. (Illustration of CERN LHC p-p collisions)
- Surpasses the performance of previous architectures by a large margin. Values below are rejection ratio (inverse of acceptance ratio).



	All classes		$H \rightarrow b\bar{b}$	$H \rightarrow c\bar{c}$	$H \rightarrow gg$	$H \rightarrow 4q$	$H \rightarrow \nu qq'$	$t \rightarrow bqq'$	$t \rightarrow bl\nu$	$W \rightarrow qq'$	$Z \rightarrow q\bar{q}$
	Accuracy	AUC	Rej _{50%}	Rej _{50%}	Rej _{50%}	Rej _{50%}	Rej _{99%}	Rej _{50%}	Rej _{99.5%}	Rej _{50%}	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

ParticleNet at FCCee



- Superb performance!
 - $\sim x10$ better than LCFIPlus
- Fast simulation (Delphes)
 - How different with Full-sim?
 - “Bad tracks” affect a lot in flavor tagging
- Dependence on detector performance?

→ Confirmation with Full-sim is very important!
 → trial with ILD full simulation with latest algorithm (Particle Transformer)

2nd ECFA HF workshop on reconstruction, 11-12 July 2023

<https://indico.cern.ch/event/1283129/>

14 Sep. 2023

Full paper: <https://link.springer.com/article/10.1140/epjc/s10052-022-10609-1>

Data Used For Investigation

- ILD full simulation:
 1. $e^+ e^- \rightarrow qq$ (at 91 GeV)
(DBD sample used for initial LCFIPlus study)
 2. $e^+ e^- \rightarrow \nu\nu H \rightarrow \nu\nu qq$ (at 250 GeV)
(2020 production, process ID: 410001-410006)

With 1M jets (500k events) each

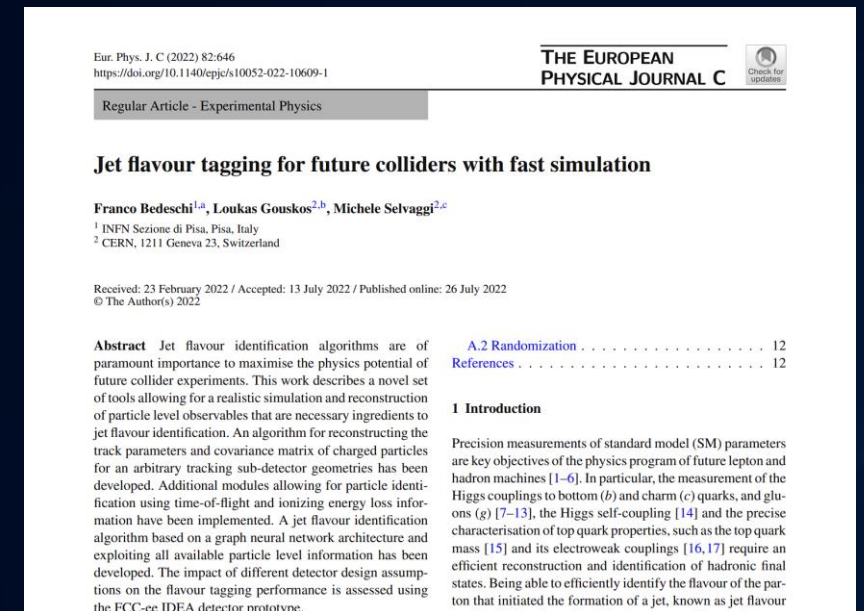
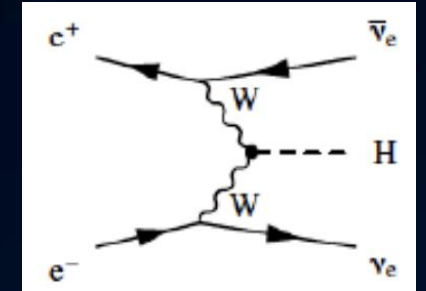
- FCCee fast simulation (Delphes with IDEA detector):

$e^+ e^- \rightarrow \nu\nu H \rightarrow \nu\nu qq$ (at 240 GeV)

With 10M jets (5M events) each

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

$\left\{ \begin{array}{l} q = b, c, u, d, s \\ \nu = \text{neutrino} \end{array} \right\}$



<https://link.springer.com/article/10.1140/epjcs/10052-022-10609-1>

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

- Impact Parameter (6):

{ pfcand_dxy
pfcand_dz
pfcand_btagSip2dVal
pfcand_btagSip2dSig
pfcand_btagSip3dVal
pfcand_btagSip3dSig

- Jet Distance (2):

{ pfcand_btagJetDistVal
pfcand_btagJetDistSig

- Particle ID (6):

{ pfcand_isMu
pfcand_isEl
pfcand_isChargedHad
pfcand_isGamma
pfcand_isNeutralHad
pfcand_type

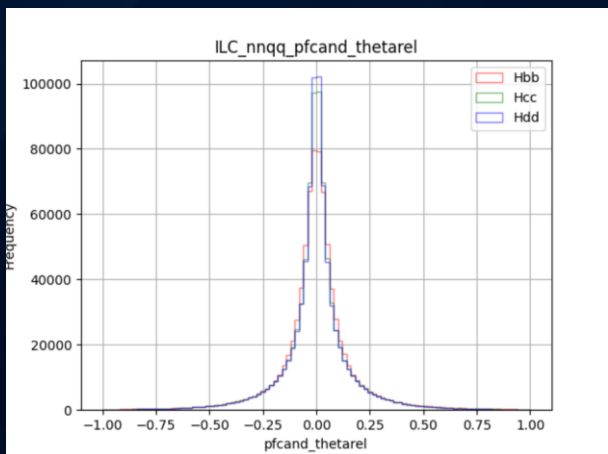
- Kinematic (4):

{ pfcand_erep_log
pfcand_thetarel
pfcand_phirel
pfcand_charge

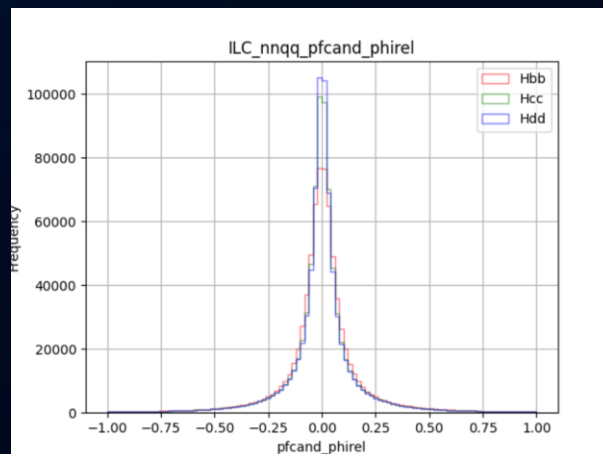
- Track Errors (15):

{ 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

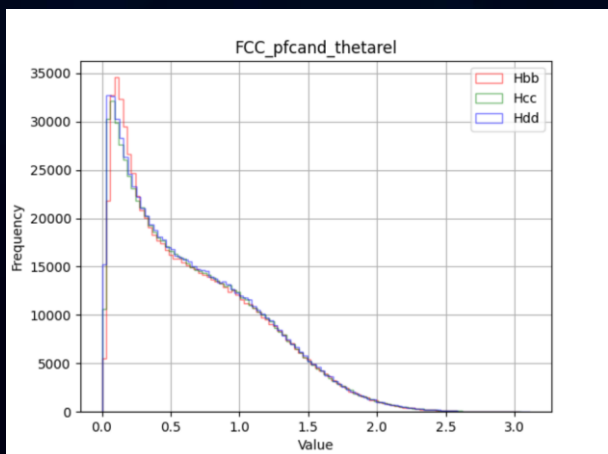
ILD vs. FCC – theta/phi distribution



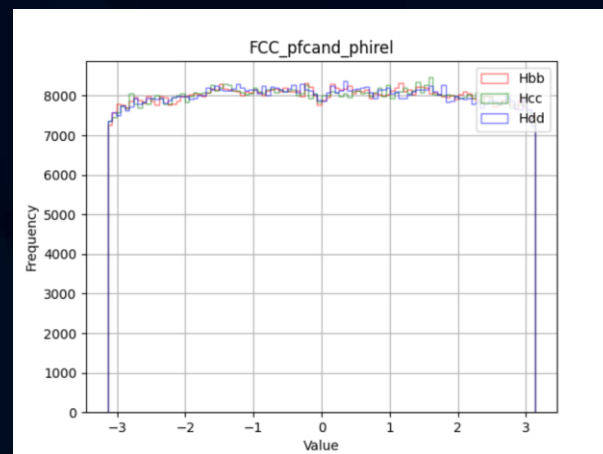
ILD theta



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.

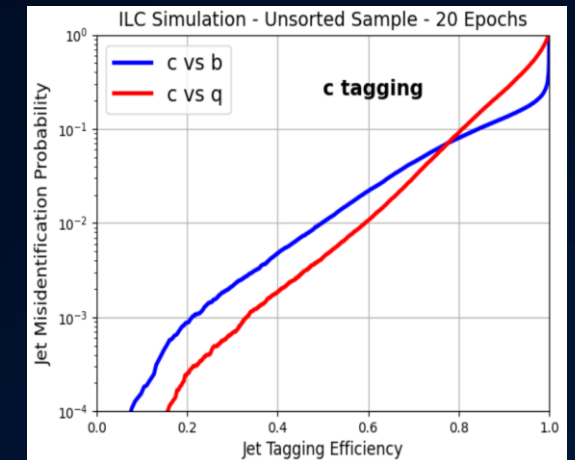
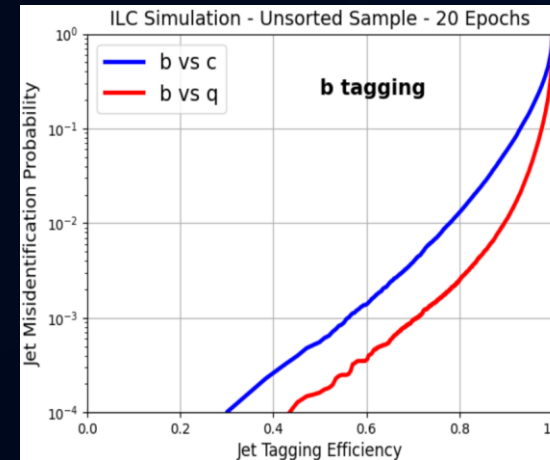
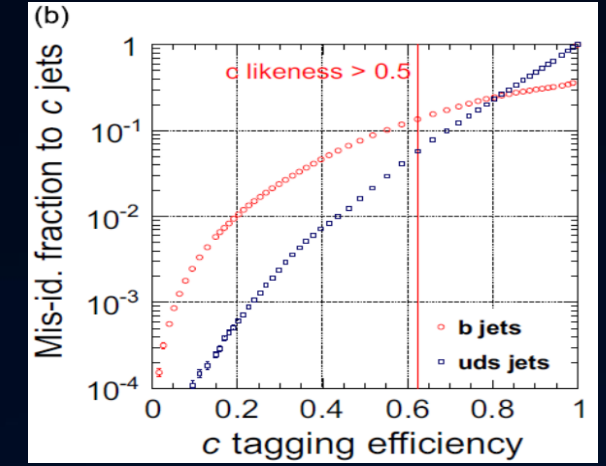
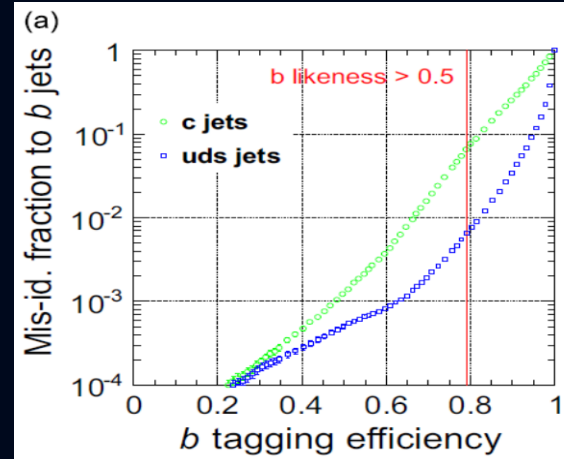
Input Variables - Interactions

- 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)

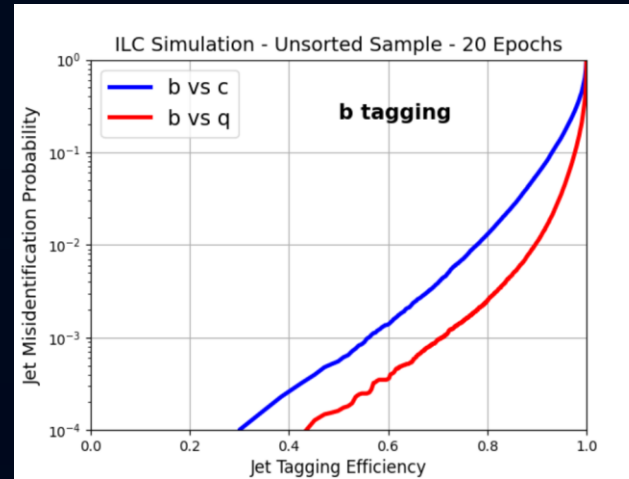
- Jet tagging performance is greatly improved by ParT immediately.
- The performance is improved by 4.05 – 9.80 times compared to LCFIPlus with the same set of data.
- Can this performance to be further improved?



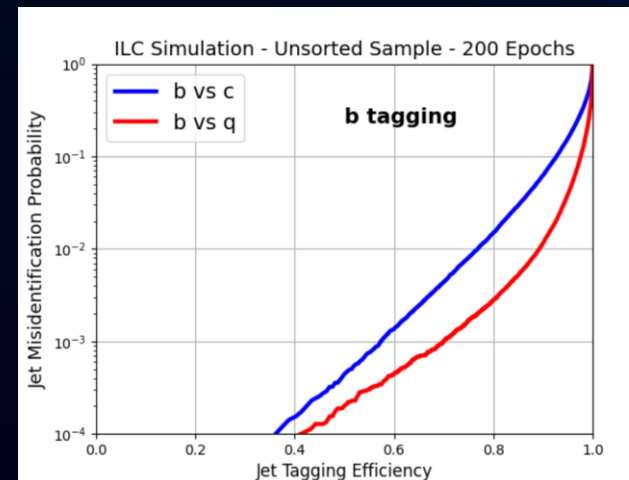
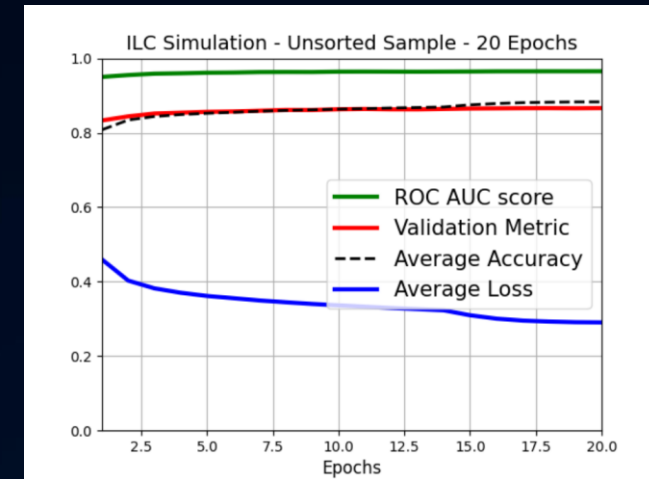
Method	b-tag 80% eff.		c-tag 50% eff.	
	c-bkg acceptance	uds-bkg acceptance	c-bkg acceptance	uds-bkg acceptance
LCFIPlus	10%	1%	10%	2%
ParT	1.29%	0.25%	1.02%	0.43%

Training parameters - epochs

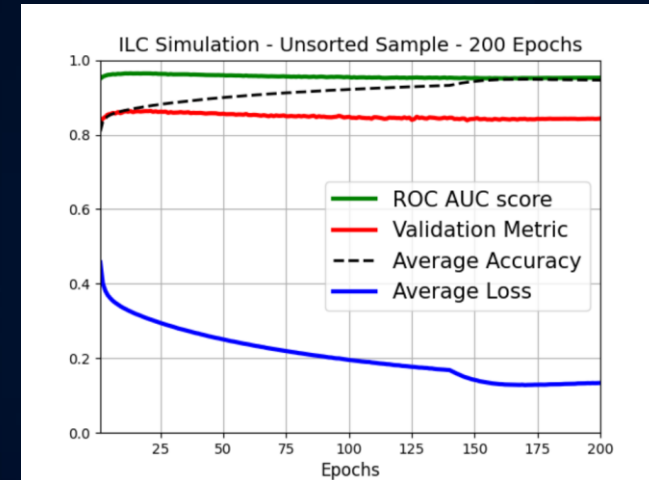
- Run on NVIDIA TITAN RTX (memory: 24 GB)
 - 20 Epochs: 3 hours
 - 200 Epochs: 30 hours
- No significant improvement in tagging efficiency
- Both ROC AUC score and Validation Metric reaches a maximum around 20 epochs.
- Overtraining after 20 epochs.
- Hence 20 epochs of training is selected to avoid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



20 epochs (ILD qq 91 GeV)

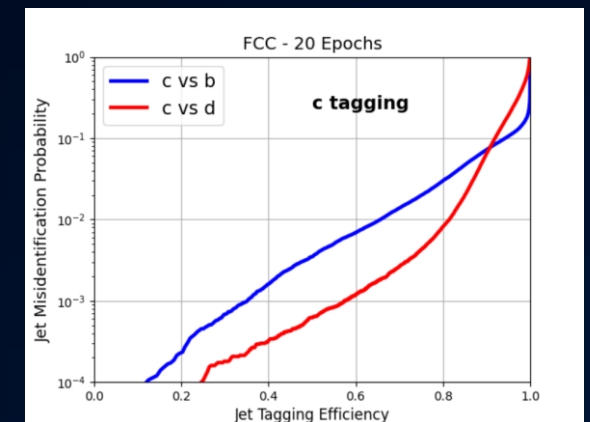
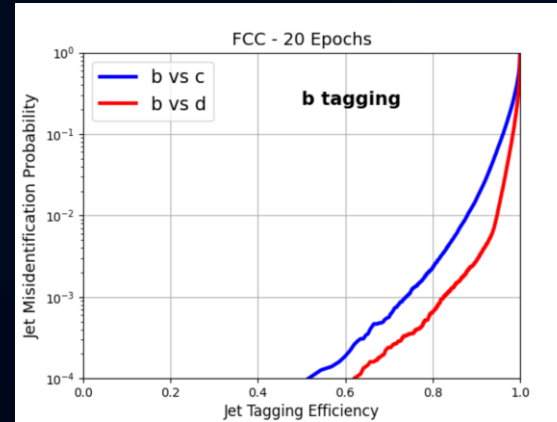
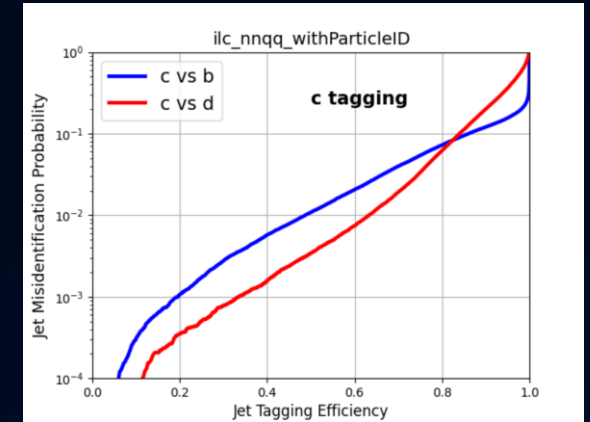
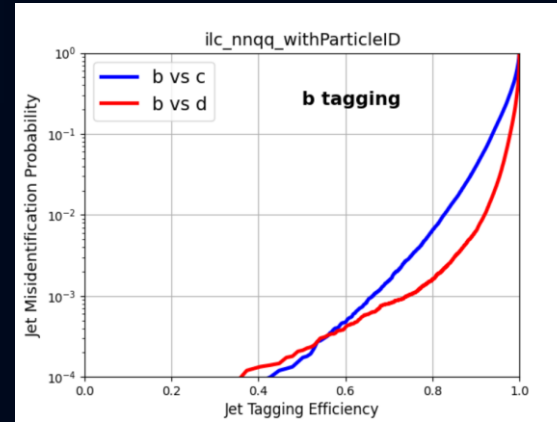


200 epochs (ILD qq 91 GeV)



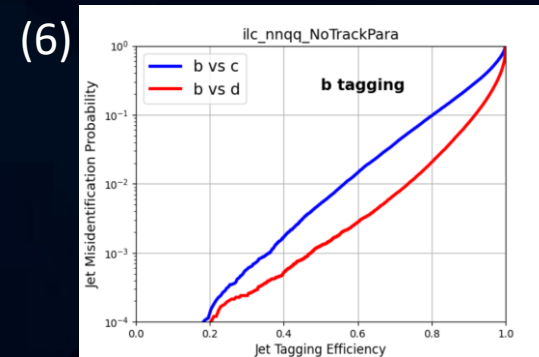
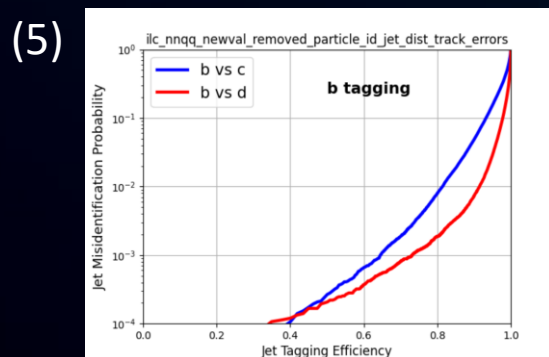
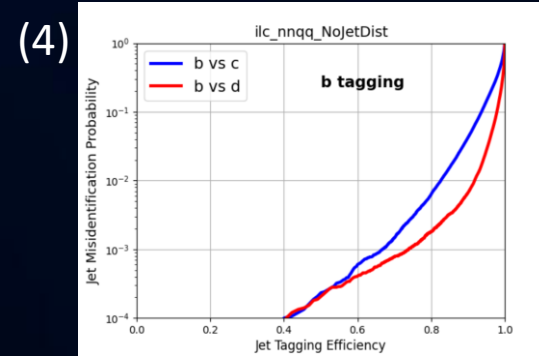
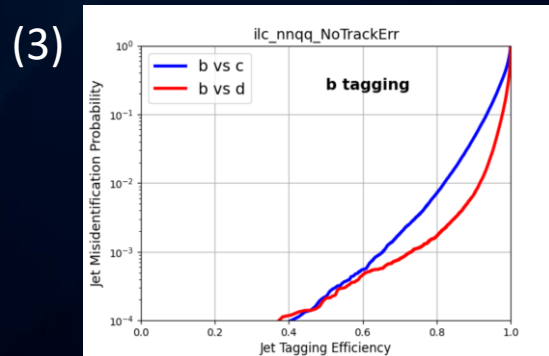
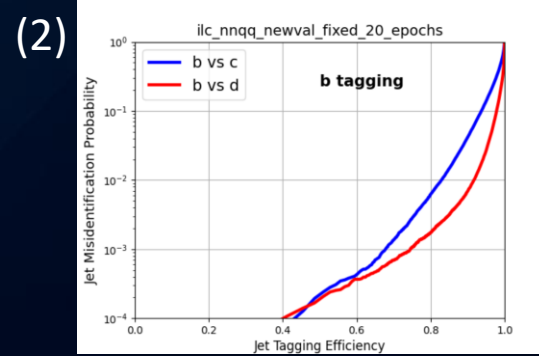
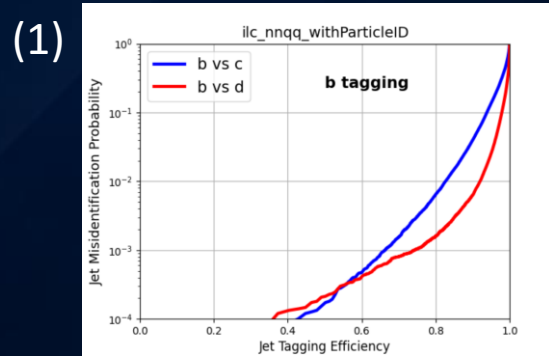
Comparison with FCC data^[3]

- 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.
- We would like to understand what factors caused this difference.



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%

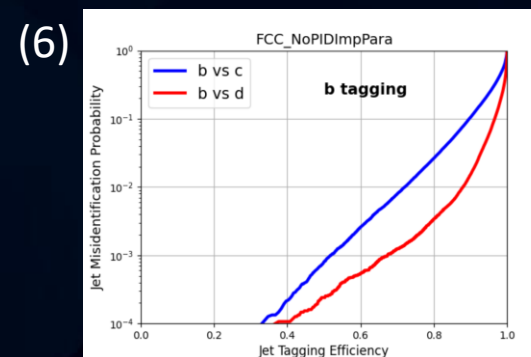
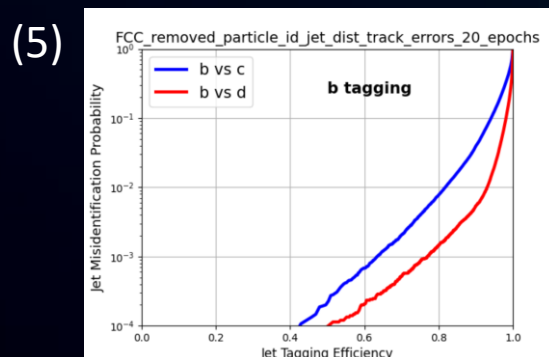
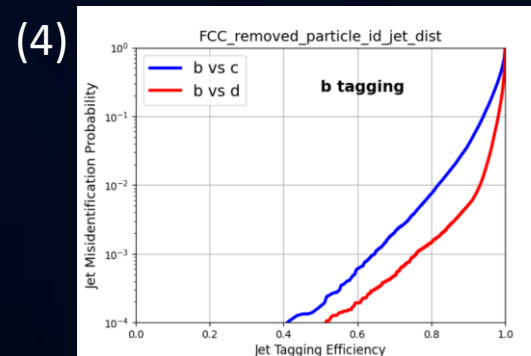
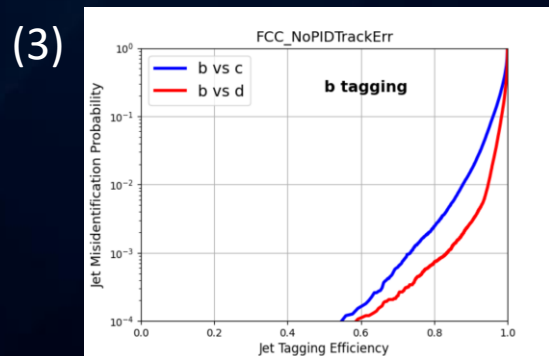
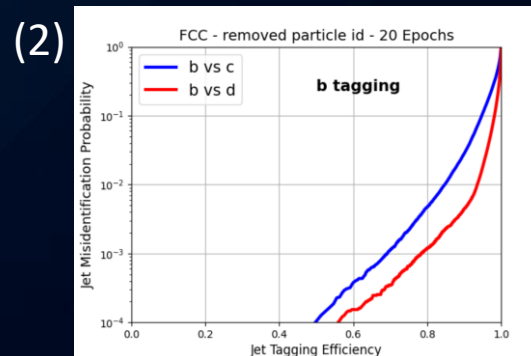
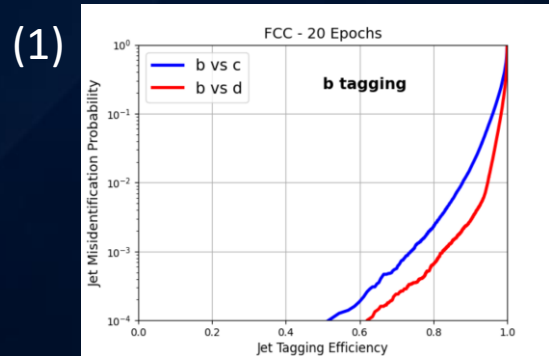
Effect of different parameters: ILD ($\nu\nu qq$ 250 GeV)



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



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%

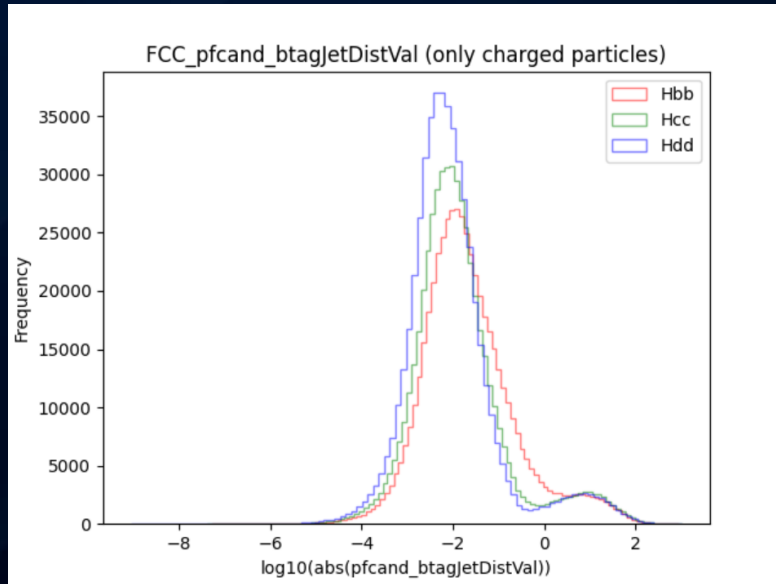
- Effect of Impact Parameters also significant.
- Both Particle ID and Jet Distance give significant impacts.
- Removal of track errors improves performance, could be a result of too many variables of Track Errors (15) shifting away the contribution of others. Further investigation should be conducted.

ILD ($\nu\nu qq$ 250 GeV) vs. FCC

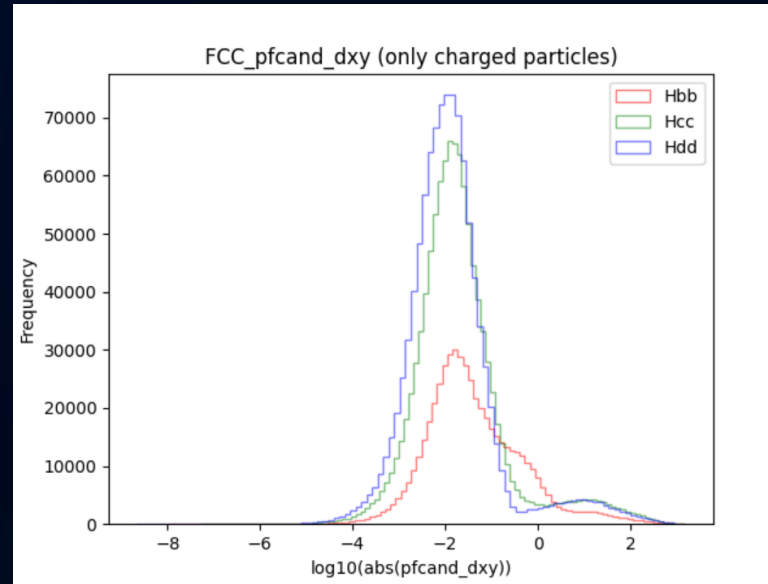
					c-bkg acceptance @ b-tag 80% eff.		b-bkg acceptance @ c-tag 50% eff.	
Plot Index	Particle ID	Impact Parameters	Jet Distance	Track Errors	ILD	FCC	ILD	FCC
(1)	●	●	●	●	0.64%	0.23%	1.09%	0.35%
(2)	✗	●	●	●	0.62%	0.47%	1.14%	0.64%
(3)	✗	●	●	✗	0.71%	0.24%	1.24%	0.35%
(4)	✗	●	✗	●	0.63%	0.75%	1.19%	0.80%
(5)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

- Overall, ILD data is performing slightly worse than FCC data in ParT training.
- There are three potential factors:
 1. FCC has rather ideal detector response as a result of fast simulation
 2. FCC's Impact Parameter has potentially better resolution
 3. The Particle ID of ILD is rather simple, not yet including the recent development
- For (5), when the input variable is reduced to be only Impact Parameters, the performance for b-tagging becomes very similar, while FCC does better in c-tagging
- This potentially indicates that resolution of Impact Parameter is more crucial for c-tagging than b-tagging (since charm hadrons decay faster than heavier bottom hadrons)

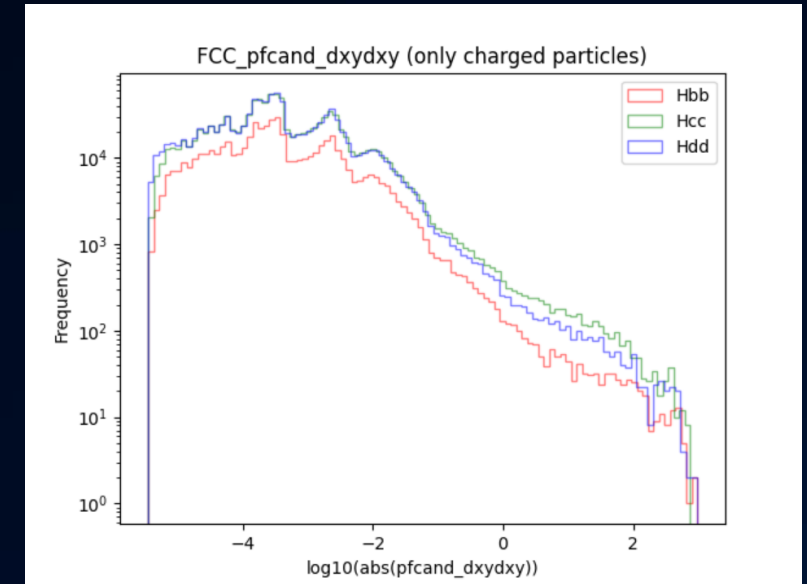
Potential Improvement: log(abs)



Jet Distance



Impact Parameter



Track Errors










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

Potential Improvement: $\log(\text{abs})$

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%

- Adding $\log(\text{abs})$ to three parameters of ILD (vvqq 250 GeV) does improve performance.
- However, the addition of $\log(\text{abs})$ of Jet Distance and Track Errors only decreases the performance.
- Can be a result of too many parameters lowers the weight of contribution of impact parameter in the model, which is more significant.
- Addition of only $\log(\text{abs})$ of Impact Parameters gives the best performance.
- Also tried replacing the original values with $\log(\text{abs})$.
- Performance decreased – possible loss of directional information.

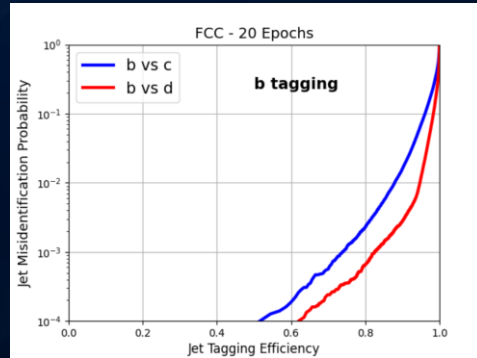
Use p_x , p_y , p_z instead of p (Interaction)

				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%	

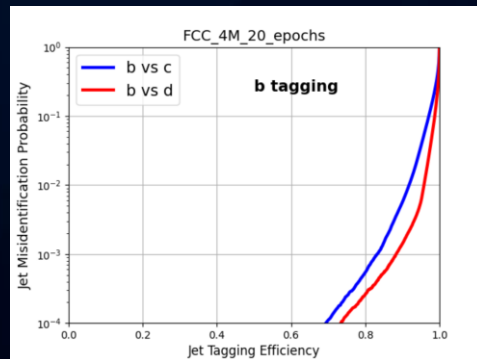
- ILD (vvqq 250 GeV) data shows that application of p_x , p_y , p_z has better performance than p .
- However, application of $\log(\text{abs})$ of the parameters becomes less significant.
- Can be because that application of p_x , p_y , p_z changes the way $\log(\text{abs})$ interacts with other parameters.
- Other potential treatments can be investigated.

Sample size affects performance (FCCee sample)

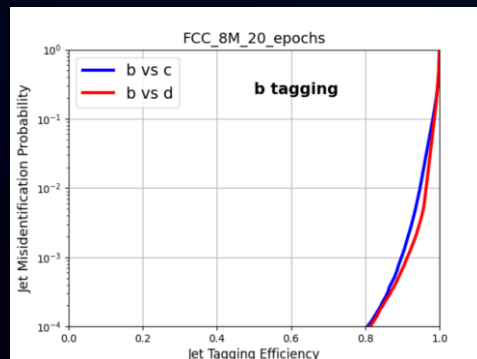
(1)



(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	0.0076%	0.10%

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

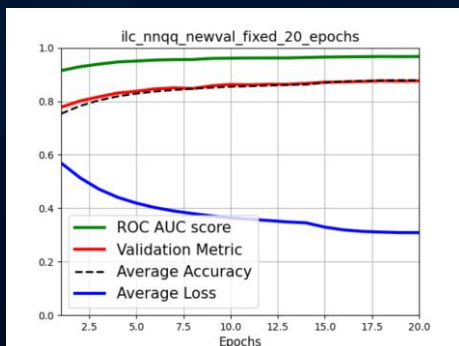
- Pretrained with fast sim and fine-tune with full sim
- Pretrained with large central production and fine-tune with dedicated physics samples in each analysis

							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%

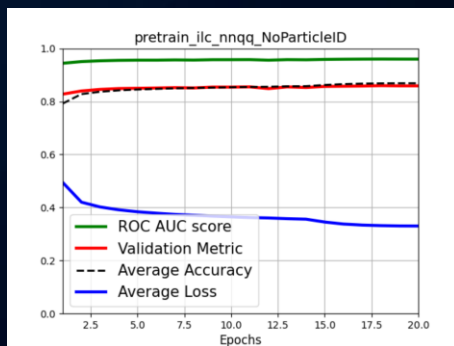
- Use result of 8M FCC data to train ILD 800k data
- Improves performance only when setups are similar
- Training of same setup (pretrain ILD 91 GeV data with ILD 250 GeV data) gives best performance
- Further investigation should be conducted on how to maximise the outcome for fine-tuning between different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

Fine tuning – Training curves

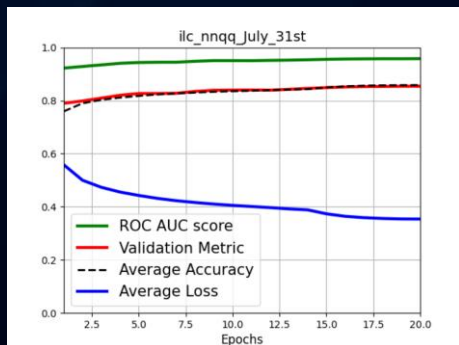
(1)



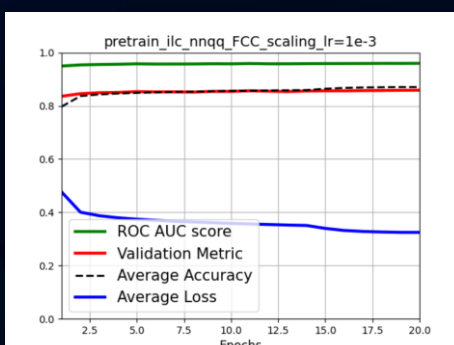
(2)



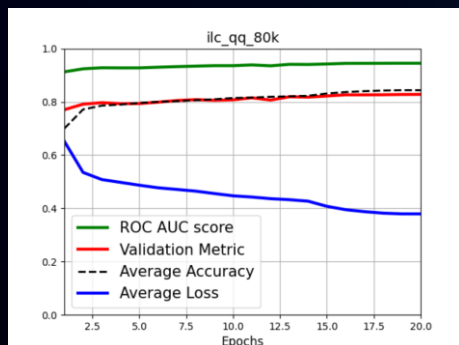
(3)



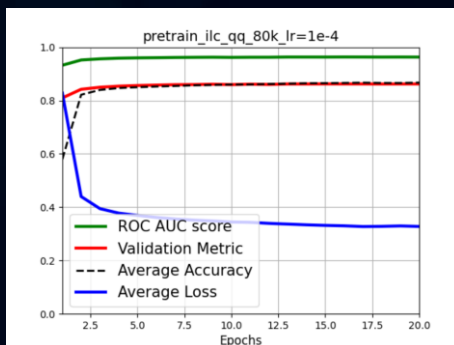
(4)



(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

Potential Further Investigation

1. Application to real physics data (e.g. Higgs identification)
2. Potentially combine LCFIPlus with ParT to further improve performance
3. Train with bigger sample of ILD
4. Fast simulation data of ILD can be potentially used for pretraining for the full simulation data
5. Particle ID for ILD data can be better implemented by applying the timing and dE/dx measurement (can also be used for testing accuracy of detectors required by examining the strange-tagging performance)
6. Applying transformer to other reconstruction algorithms (e.g. particle flow) and investigate on its wider usage

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>