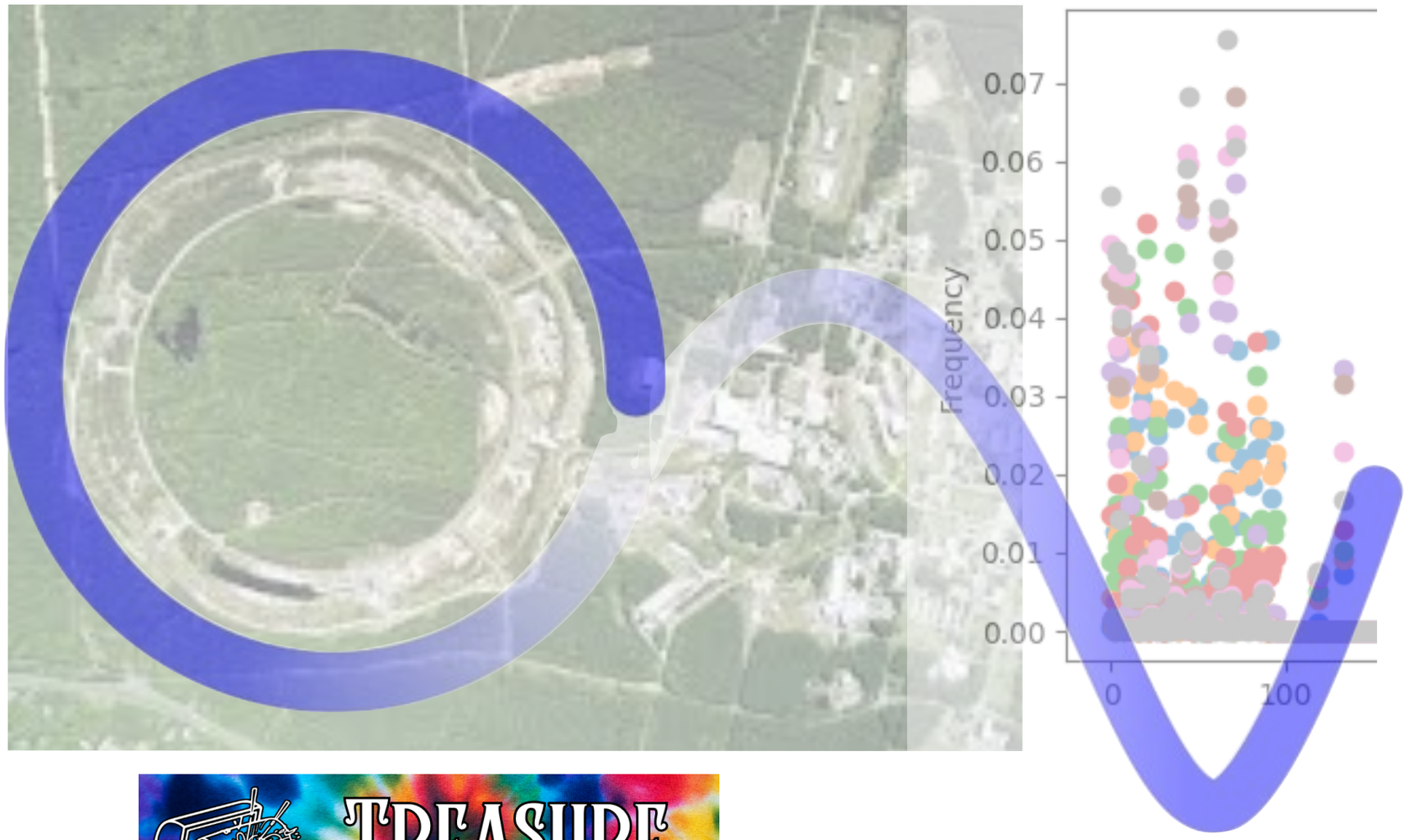
Tokens for Flavor Identification at the FCC-*ee*



Workshop

April 29, 2026

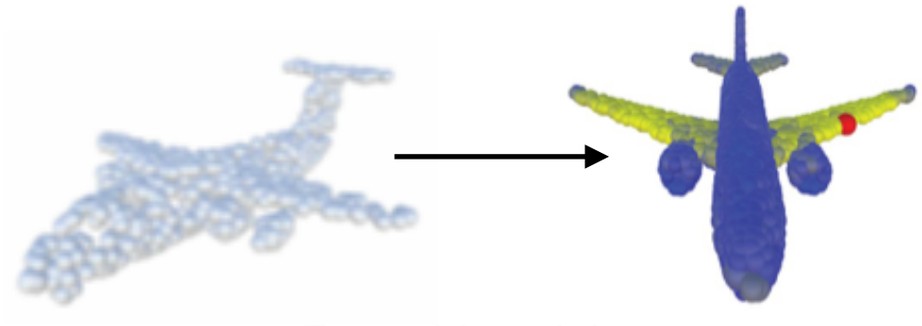
Andrea Sciandra

Brookhaven National Laboratory

FUTURE CIRCULAR COLLIDER

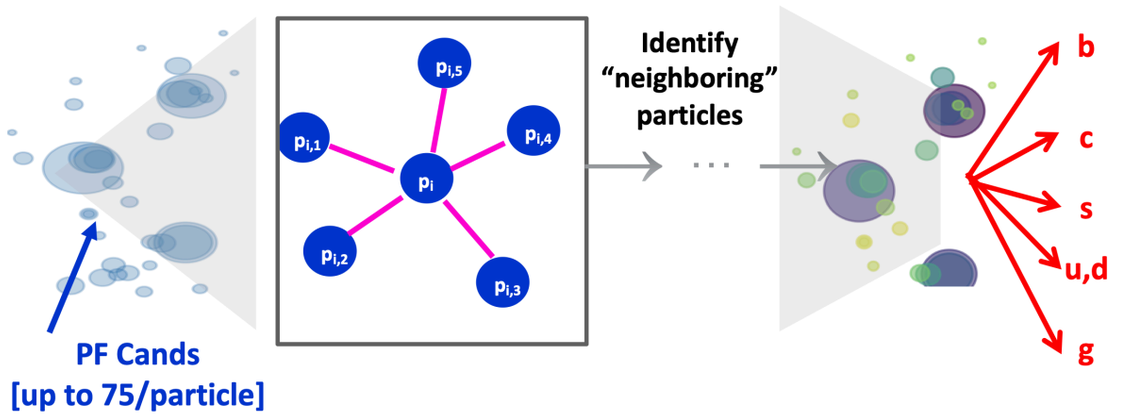
FCC-ee Flavor-Tagging in a Nutshell

- Graph-based tagger, where each jet = “cone” of reco particles traversing the detector
- Particle-flow (PF): particle candidates mutually exclusive & have 35 features associated with
  - E/p, position
  - Impact parameters, particle type
  - *Timing, cluster counting (dN/dx) info*
- Exclusive kT jet reco: unordered sets of particles with correlations & relationships
- Graph-Neural-Network for ParticleNet:
  - Identify properties of “particle cloud”, represented as a **graph**
  - Each particle: **node** of the graph; connections between particles: the **edges**
  - Learn local structures → move to more global ones



From this article

[O(40) properties/particle]
x [~50-100 particles/jet]
~O(1000) inputs/jet

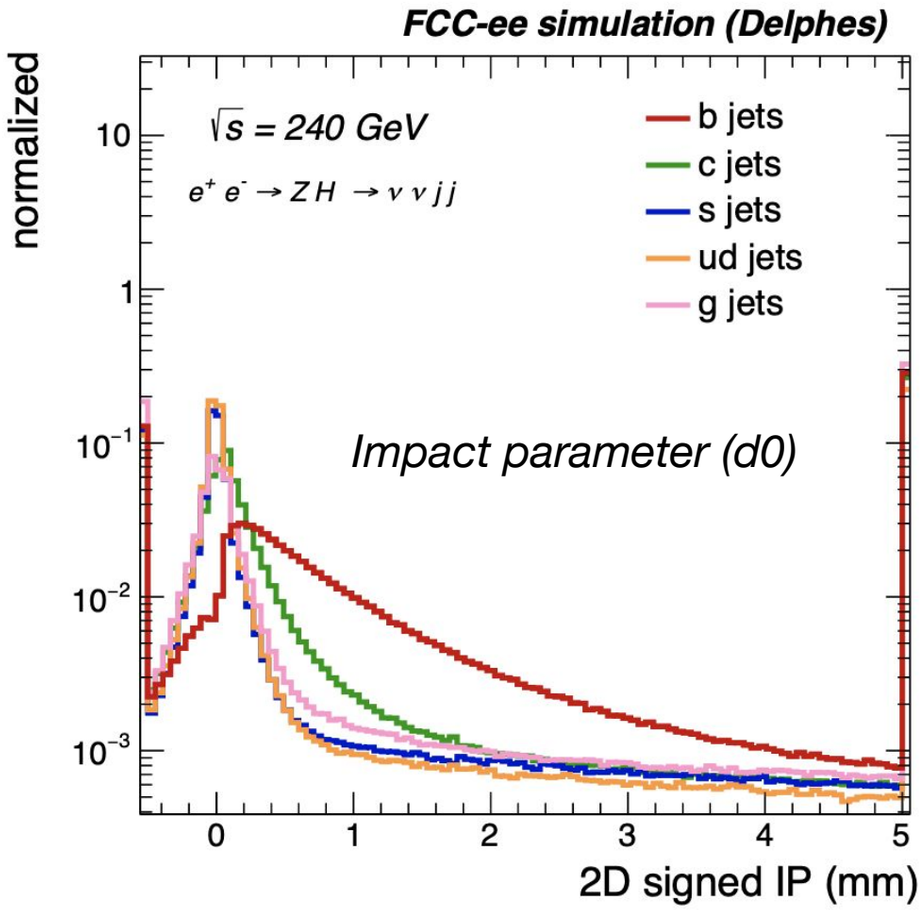


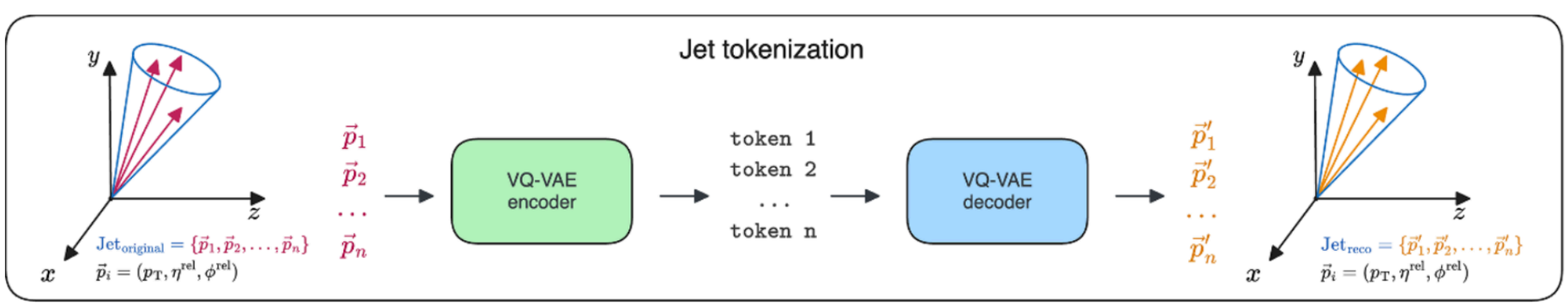
Learning Discrete Representations for Jet Tagging

- Modern jet taggers (e.g. graph transformers) use *continuous* PF *features*
- Powerful but:
  - computationally expensive
    - full training: 4 GPUs x 1 week
  - hard to interpret
- Idea, as for ATLAS/CMS jet tasks:
  - *map* PF candidates → *discrete tokens*
  - train *lightweight* models on tokens

Goal:

- Learn a *compact, physics-aware tokenizer*
- *Preserve jet flavor* information
- Compare with *continuous*-feature baselines



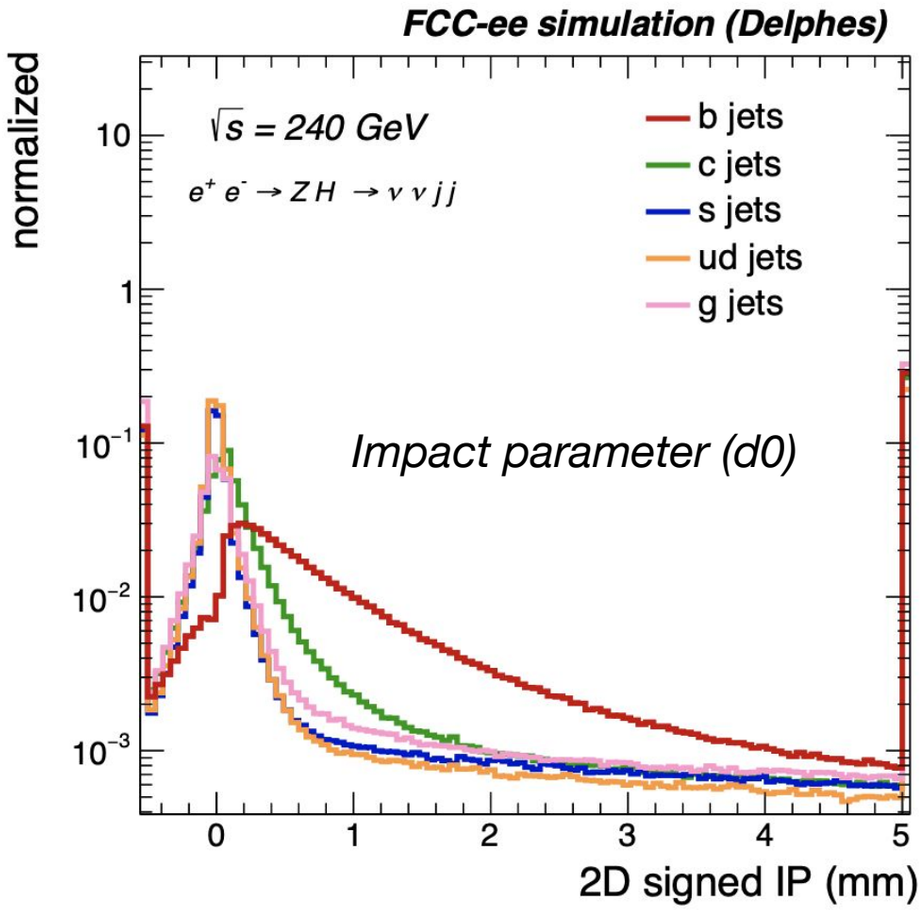


Learning Discrete Representations for Jet Tagging

- Modern jet taggers (e.g. graph transformers) use *continuous* PF *features*
- Powerful but:
  - computationally expensive
    - full training: 4 GPUs x 1 week
  - hard to interpret
- Idea, as for ATLAS/CMS jet tasks:
  - *map* PF candidates → *discrete tokens*
  - train *lightweight* models on tokens

Goal:

- Learn a *compact, physics-aware tokenizer*
- *Preserve jet flavor* information
- Compare with *continuous*-feature baselines



Disclaimer:

These are early-stage results from ongoing work over the past few weeks, as I continue building expertise in this area!

Workflow Overview

Developed full workflow, available on github

- ***Input***: official FCC-ee ROOT ntuples transformed into HDF5
  - up to O(10M) $H \to jj$ jets & 75 PF candidates / jet
  - 35 features per candidate → *standardized*
- Step 1: ***VQ-VAE tokenizer***
  - learns *codebook* of size K
  - assigns each *particle* → ***token***
- Step 2: ***Transformer*** classifier
  - sequence of *tokens* + *mask*
  - *predicts* jet flavor
    - 7 classes: b,c,s,d,u,g,τ

Model Architectures

VQ-VAE

- Encoder: Multi-Layer Perceptron (per particle)
  - F=35 → 64 → 64 → D (D = 16-128, def 64)
- Vector Quantization:
  - codebook size: (K = 16-512, def 256)
  - Exponential Moving Average updates (decay ~0.99)
- Decoder:
  - reconstruct PF features
- Optional supervision:
  - jet-level classification head
- Loss:

$$L = L_{rec} + L_{VQ} + \lambda L_{cls}$$

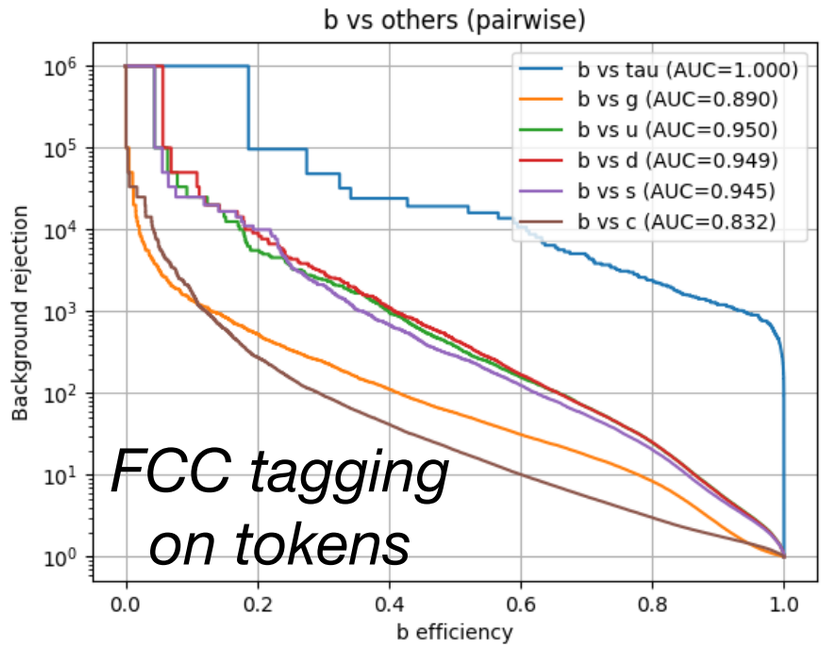
masked reco: per-particle MSE

commitment: EMA-updated codebook

jet-level weak supervision (λ~0-0.1)

(Basic) Transformer

- Input:
  - *tokens* (or continuous features baseline)
- Embedding:
  - token *embedding* OR linear projection (continuous case)
- Architecture:
  - standard transformer encoder
  - masked pooling
  - cross-entropy loss
- Output:
  - jet-level *classification* (7 flavors)



Preliminary Studies & Results

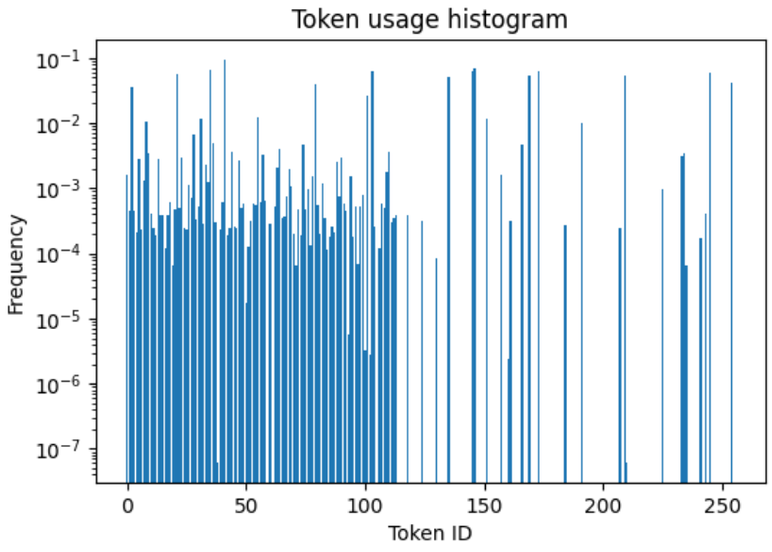
Tokenizer studies

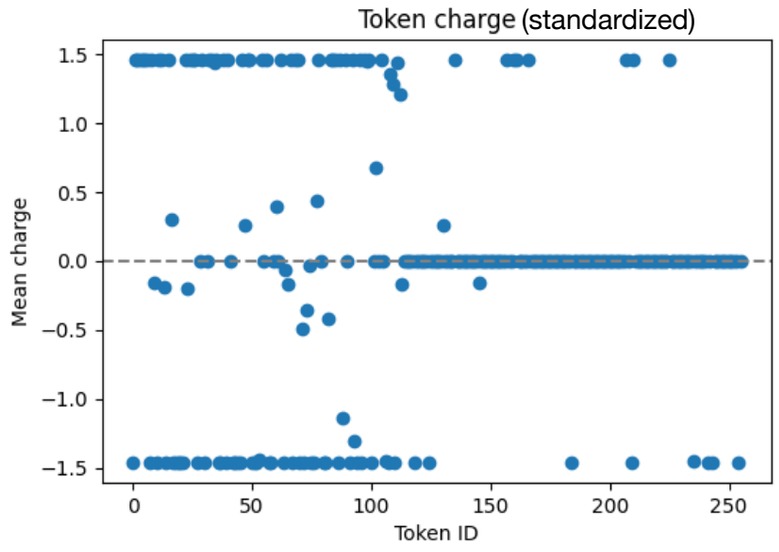
- Scan over:
  - codebook size (K = 16 → 512)
  - latent dim (D = 16 → 64)
  - number of PF features (5 → 35)
- Token diagnostics:
  - frequency vs token ID
  - dependence on jet momentum, PF charge
  - class-conditional usage

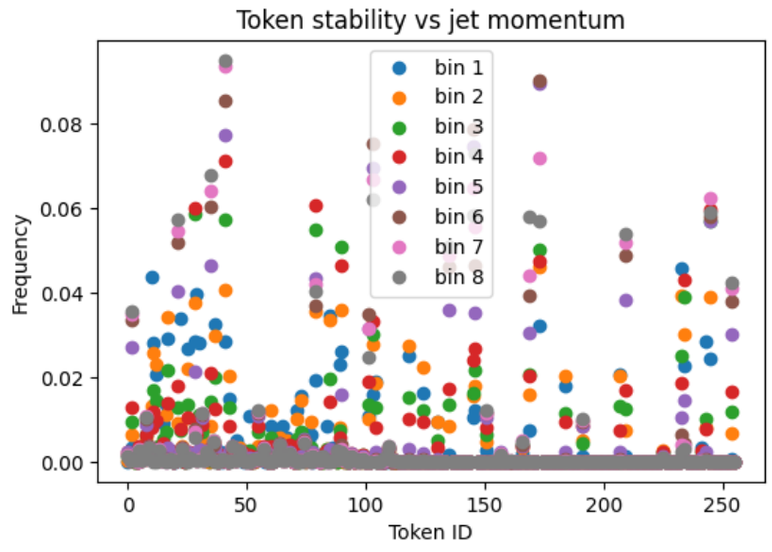
Training studies

- scaling:
  - up to ~10M jets
- stability:
  - standardization of inputs
  - label smoothing for supervised VQ-VAE
- memory / performance optimization:
  - batching, token caching

[0,5,10,15,20,30,40,50] GeV







Key Findings

1. Tokenizer behavior

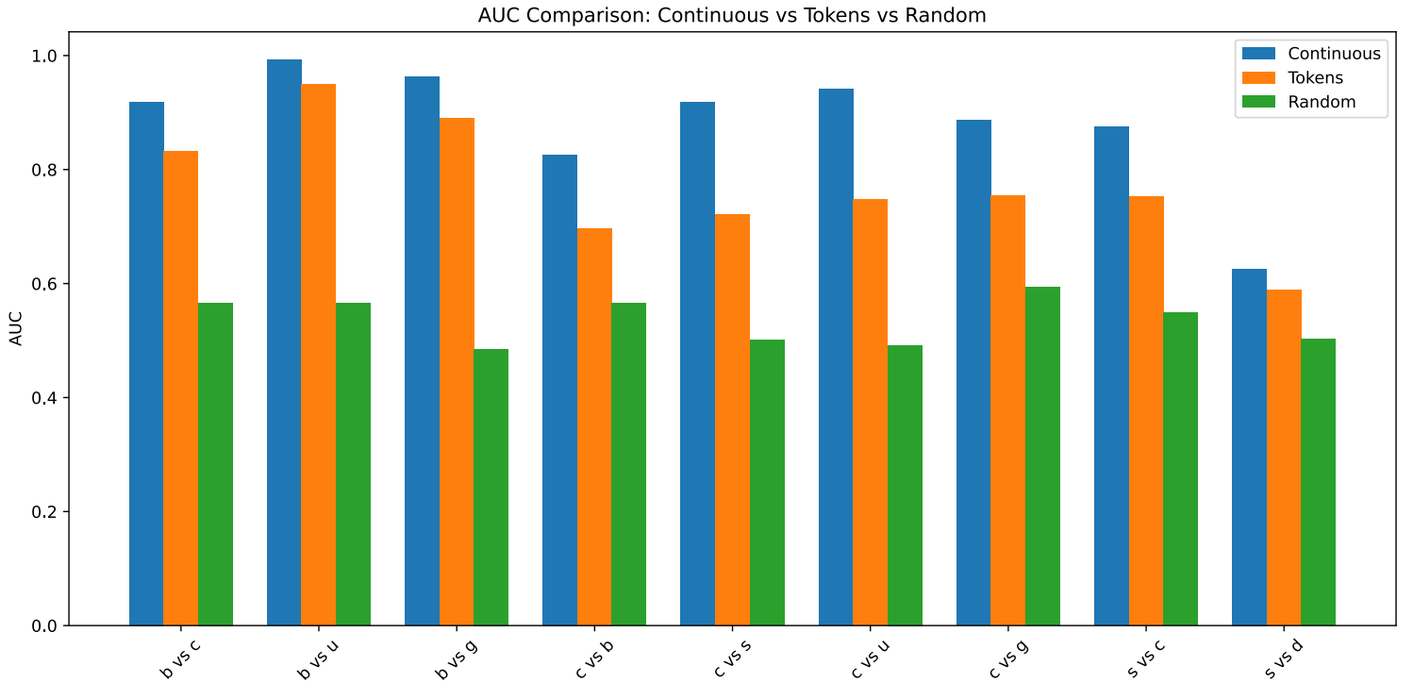
- VQ-VAE converges very quickly (~1-4 epochs)
- Large K → many *unused* tokens
- Token usage mostly *class-agnostic*
  - weak dependence on jet flavor

2. Supervised VQ-VAE

- naive supervision → classifier collapse
- *label smoothing* stabilizes training
- BUT:
  - jet-level classification does *not* strongly shape tokens

3. Downstream performance

- Transformer on tokens:
  - stable but *limited AUC gains*
  - weak scaling with more data
- *Continuous*-feature transformer:
  - significantly *better* performance



AUC: cont. vs tokens vs rnd

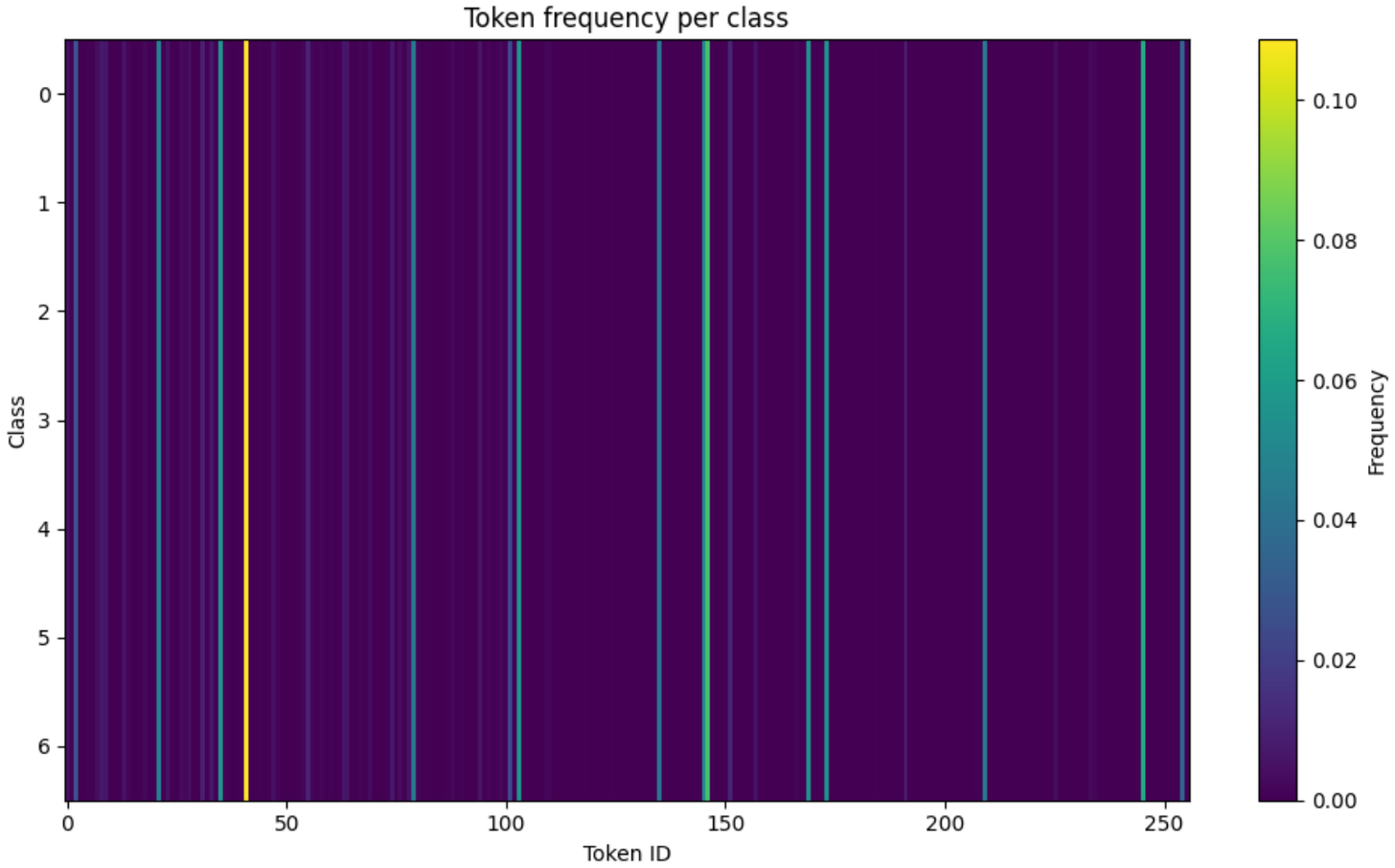
Comparing to *continuous* features:

- ***Continuous*** **>** ***Tokens*** **>>** ***Random***(ized tokens)
- Strong separations (e.g. b vs. u/g)
  - tokens still decent
  - small drop
- Weak/subtle separations (e.g. c vs. s, s vs. d)
  - tokens degrade a lot
  - closer to random

→ **Fine-grained structure lost**

This **VQ-VAE learns compressed reconstruction, not discrimination?**

More Key Findings, so far…



Token freq vs. class Heatmap

Extra studies/lessons learnt:

- Strong ***supervision*** "destroys" VQ-VAE loss behavior (gradient conflict)
- *Weak* (and/or *delayed*) supervision → no strong class structure, too weak to shape tokens
  - before pooling does not help either
- *Contrastive* loss: same class → closer tokens ; different class → separated
  - → ~no benefit, hard to make effective
- **Self-attention for context-aware tokens → small gains in performance**
- **Entropy/perplexity (token usage) ↑ ≠ performance ↑(usefulness)**

Future Directions

Short-term

- Quantify token *quality*:
  - conditional entropy H(class | token)
  - mutual information
- *Compare* systematically:
  - tokens vs continuous
  - reconstruction vs classification loss curve: where is tradeoff?
- *Diagnose which/where info is lost: hitting ceiling?*
  - *classifier on raw PF features does outperform tokens*
  - *input → **encoder** → **VQ** → tokens → TF*
- Try *richer pre-quantization representation* (currently, 35 → 64 → 64 → D=32-128)
  - deeper/wider encoder
- Try *conditional* tokenization (jet E bin)

Medium-term encoder improvements

- physics-aware *positional encoding* (e.g. ΔR)
- *multi-token* per particle
  - residual / multi-stage VQ (RQ-VAE) ?

Bridging the gap

- Compare to *hybrid* models?
  - token + continuous residuals
- Larger-scale training: match graph TF setup
  - multi-GPU, longer runs, etc…

BACKUP

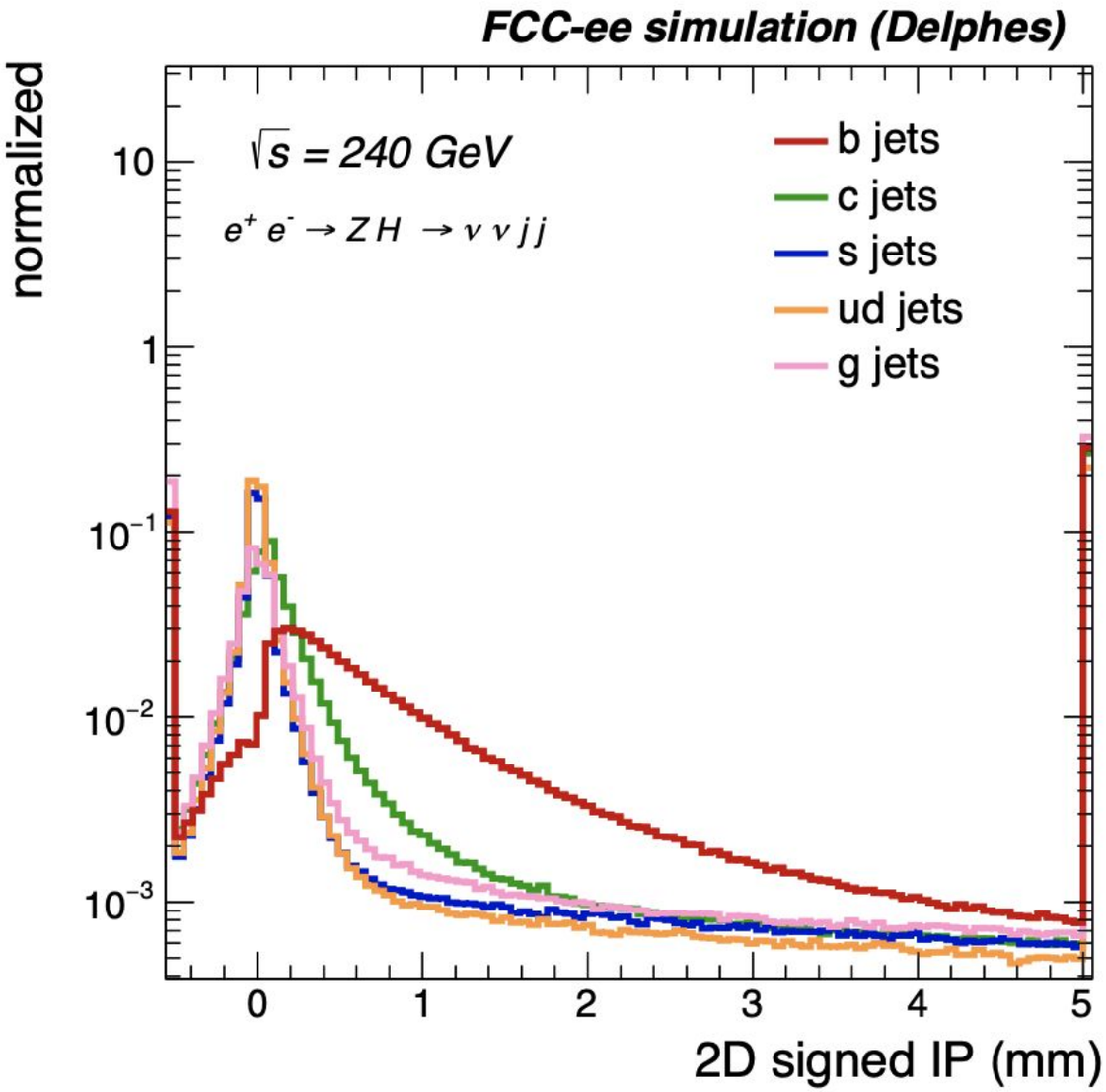
List of Input Variables

Variable	Description
	Kinematics
$E_{\text{const}}/E_{\text{jet}}$	energy of the jet constituent divided by the jet energy
θ_{rel}	polar angle of the constituent with respect to the jet momentum
ϕ_{rel}	azimuthal angle of the constituent with respect to the jet momentum
	Displacement
d_{xy}	transverse impact parameter of the track
d_z	longitudinal impact parameter of the track
SIP_{2D}	signed 2D impact parameter of the track
$\text{SIP}_{\text{2D}}/\sigma_{\text{2D}}$	signed 2D impact parameter significance of the track
SIP_{3D}	signed 3D impact parameter of the track
$\text{SIP}_{\text{3D}}/\sigma_{\text{3D}}$	signed 3D impact parameter significance of the track
d_{3D}	jet track distance at their point of closest approach
$d_{\text{3D}}/\sigma_{d_{\text{3D}}}$	jet track distance significance at their point of closest approach
C_{ij}	covariance matrix of the track parameters
	Identification
q	electric charge of the particle
$m_{\text{t.o.f.}}$	mass calculated from time-of-flight
dN/dx	number of primary ionisation clusters along track
`isMuon`	if the particle is identified as a muon
`isElectron`	if the particle is identified as an electron
`isPhoton`	if the particle is identified as a photon
`isChargedHadron`	if the particle is identified as a charged hadron
`isNeutralHadron`	if the particle is identified as a neutral hadron

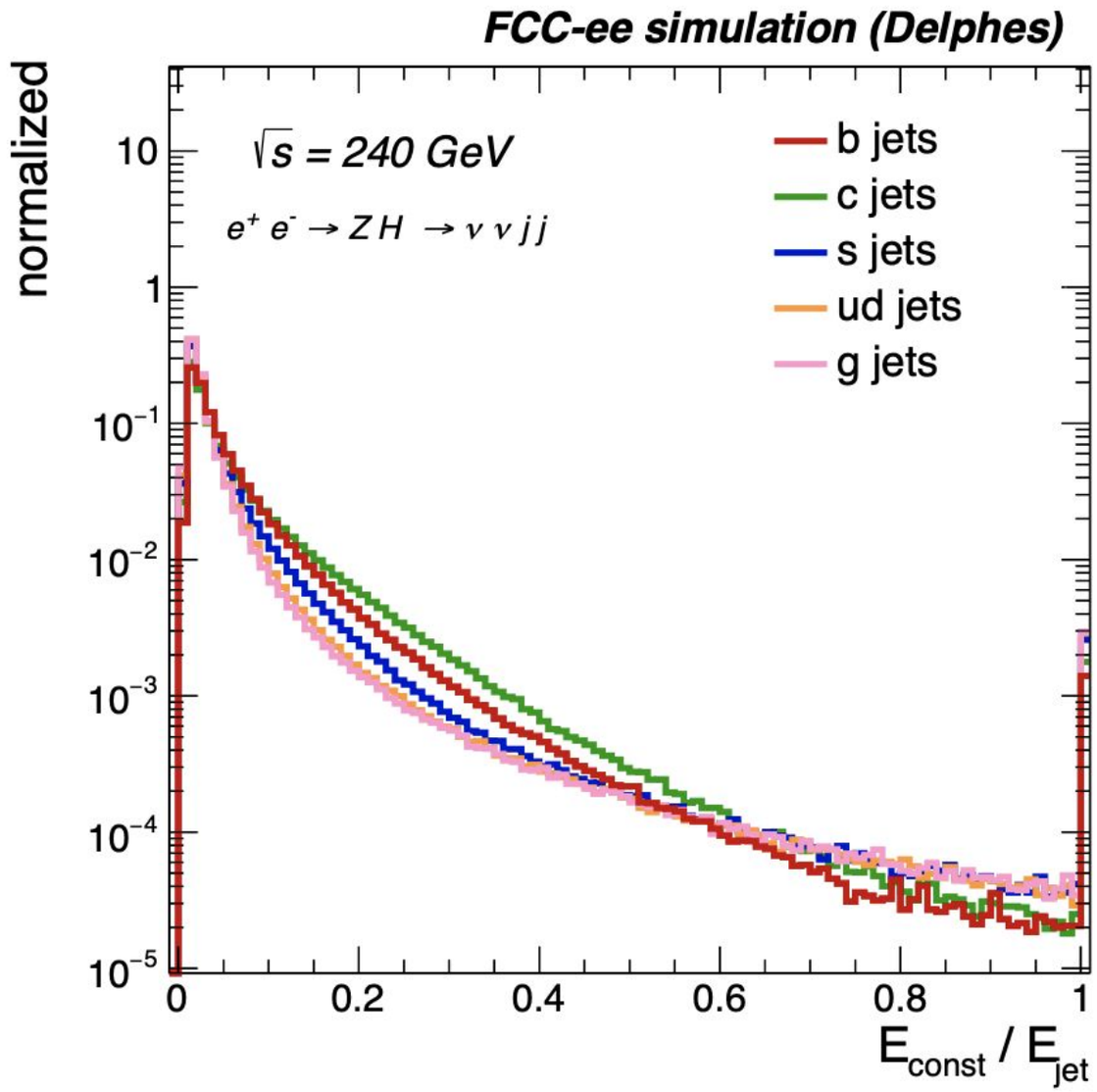
Input Variables

- Comparison of input distributions for different jet flavours

Impact parameter (d0)

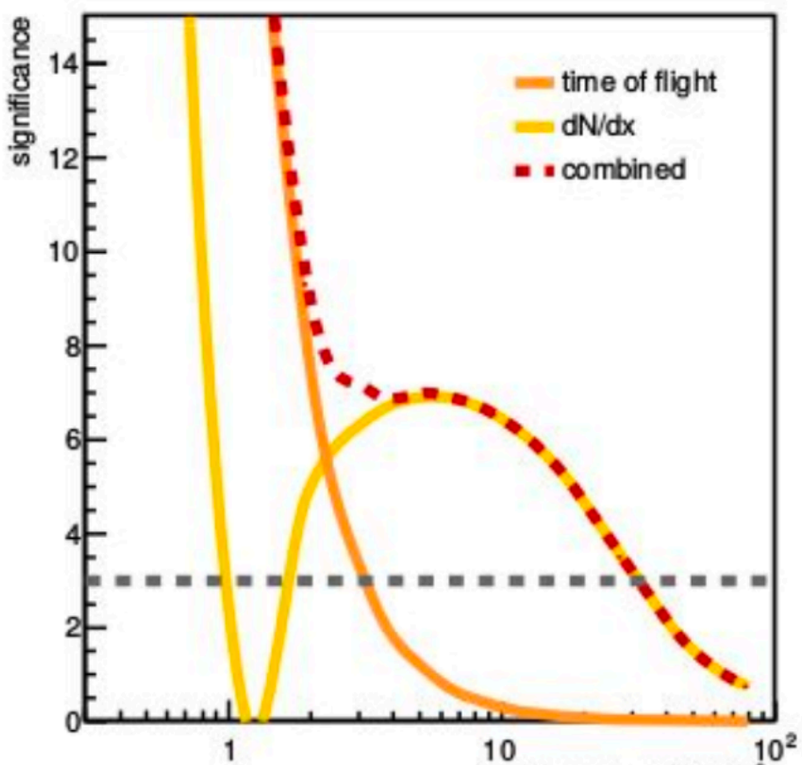


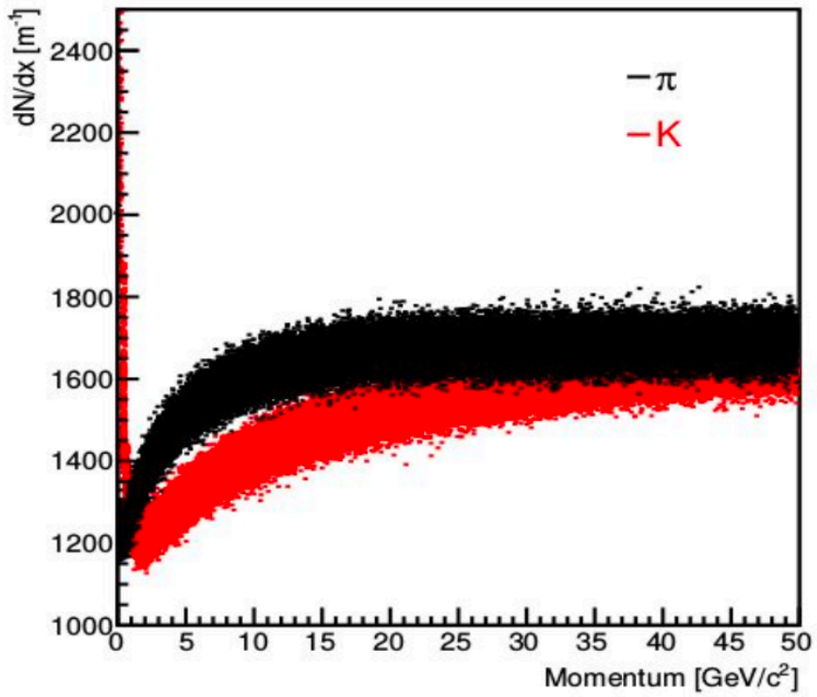
Constituent relative energy

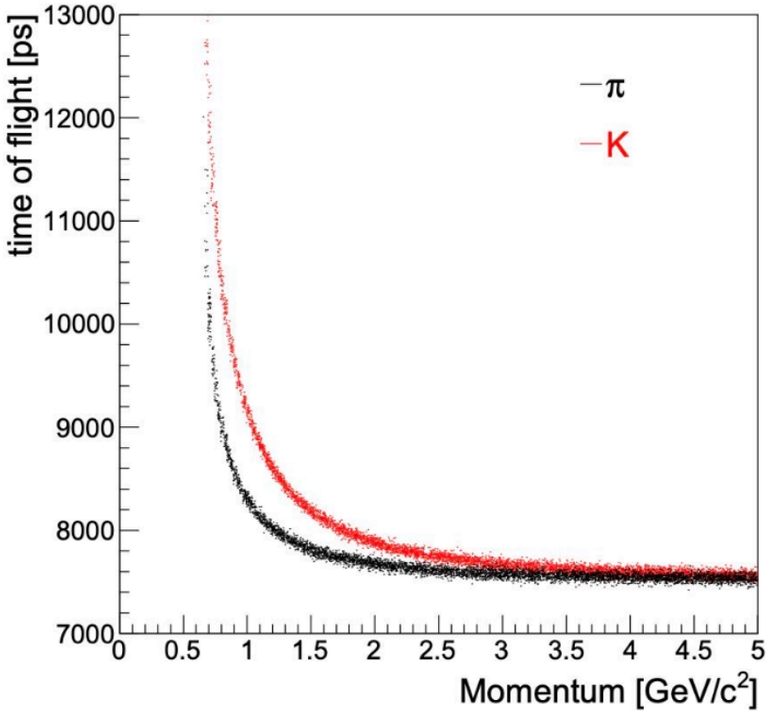


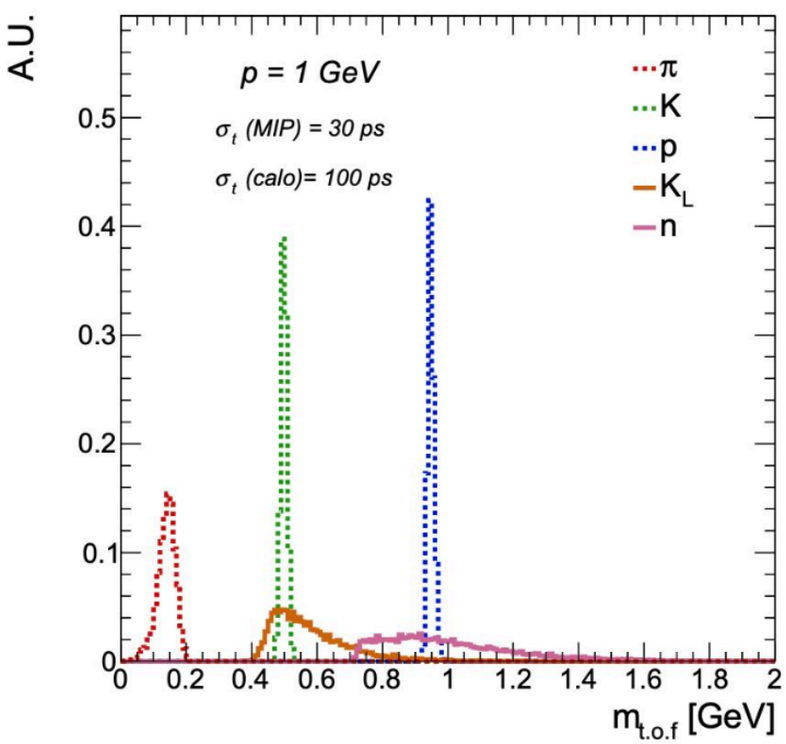
Particle ID: Cluster Counting & Time-of-Flight

- Count number of primary ionisation clusters along track path
- Time-of-Flight results in good K/π separation at low momenta
- Dedicated modules added in Delphes



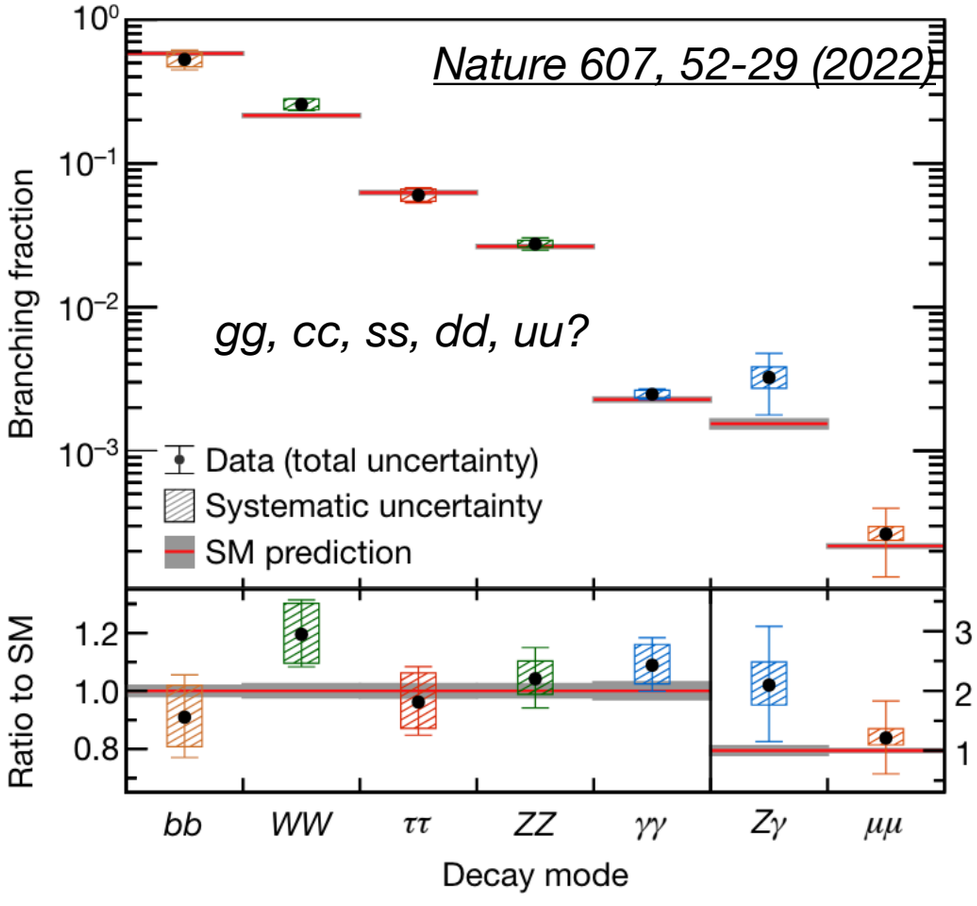


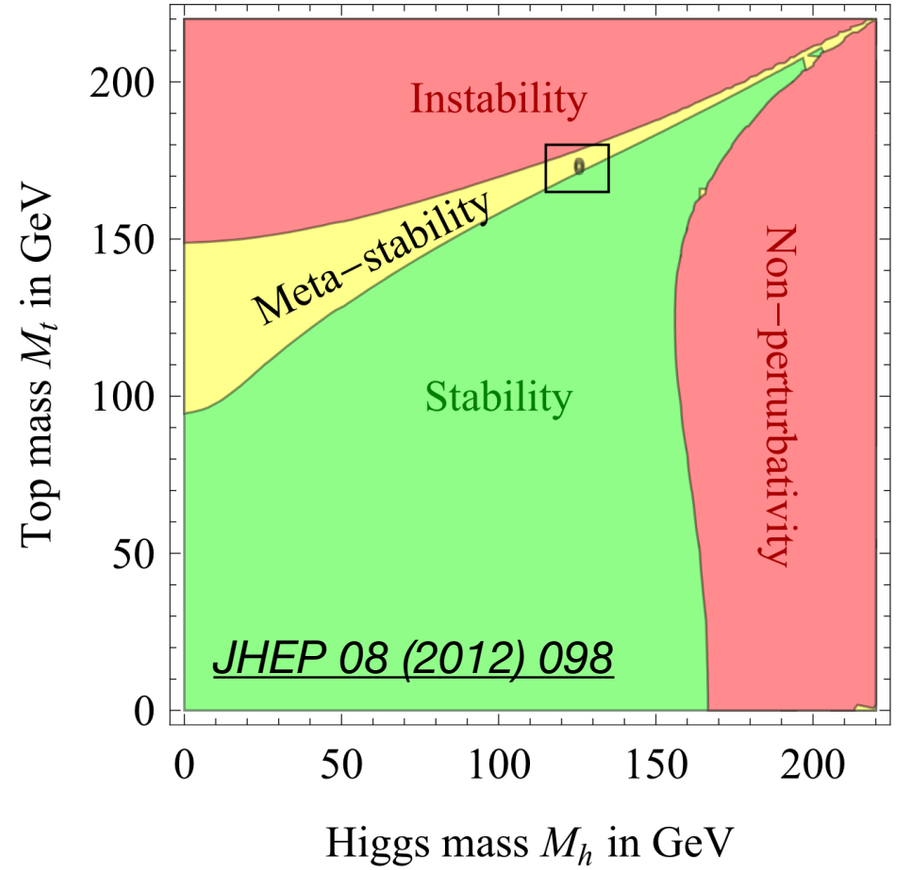




FCC-ee Tagger Motivation

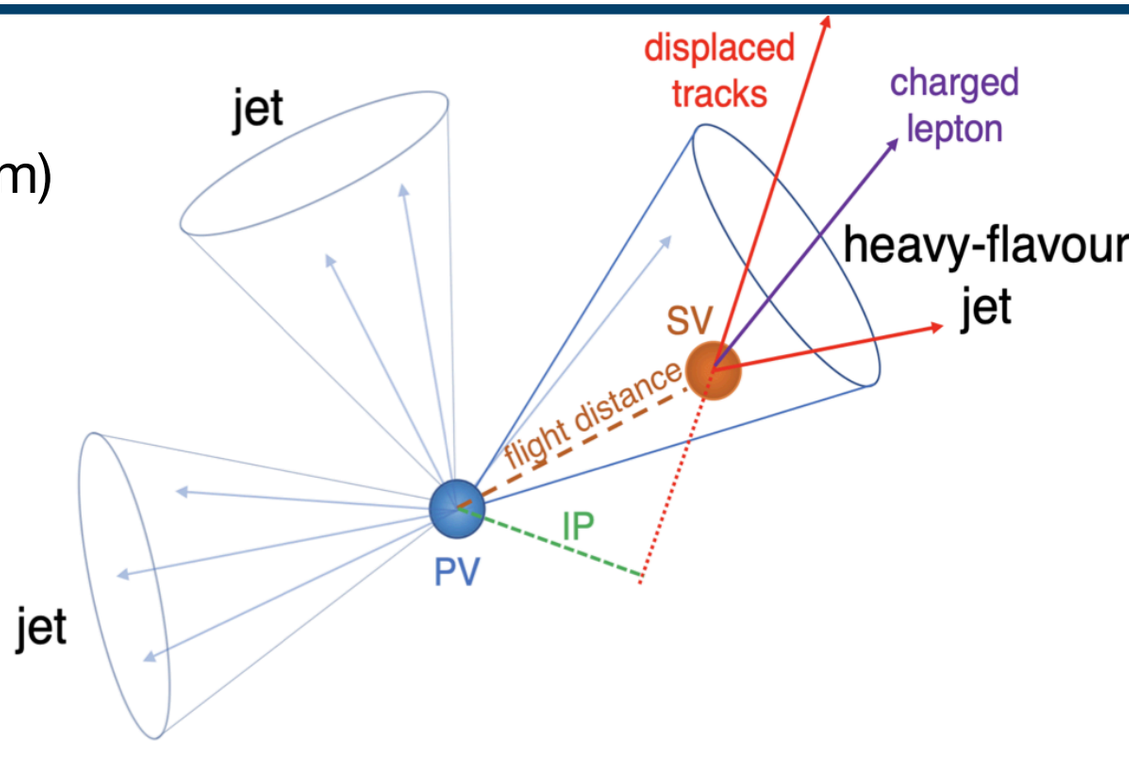
- Flavor tagging: very powerful tool, *serving Physics purpose*
  - Key for e^+e^- program!
  - Access **Higgs**-boson properties, hardly accessible at the (HL-)LHC
    - Challenging decay modes like *cc* and "impossible" hadronic decay modes: *gg*, *ss*, 1st generation quarks
- Precise determination of **top**-quark properties - provided sufficient COM energy
    - Mass, width, Yukawa
- **QCD**: strong coupling, hadronization modeling, tuning of MC, etc…
- Quark **flavour physics**, searches for FCNC, etc…





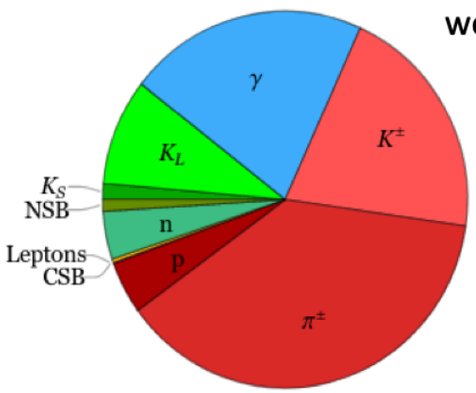
Reminder: Flavor-Tagging Principles

- ***Bottom & charm*** tagging based on:
  - Large lifetime (~1/0.1 ps) & decay length (~50-500 μm)
  - Displaced vertices/tracks
    - Tertiary vertex for B hadrons decaying to “charm hadron” or “D hadron”
  - Relatively large invariant mass
  - Specific track multiplicity (~5 charged particles on average)
  - Non-isolated charged leptons from semileptonic decays: 20(10)% in B(C)-hadrons decays
  - Tracker needs: good spatial resolution, small material budget
- ***Strange*** tagging, exploiting large Kaon content
  - Charged requiring K/π separation, neutral K_S->ππ, K_L
  - Benefitting from good PID: timing detectors, Cherenkov detectors, charged energy loss (silicon/gas)

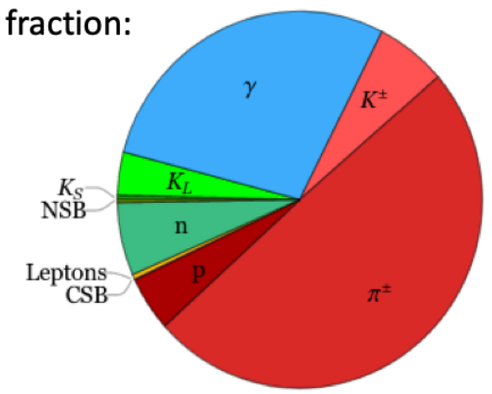


[2003.09517]

Momentum weighted fraction:



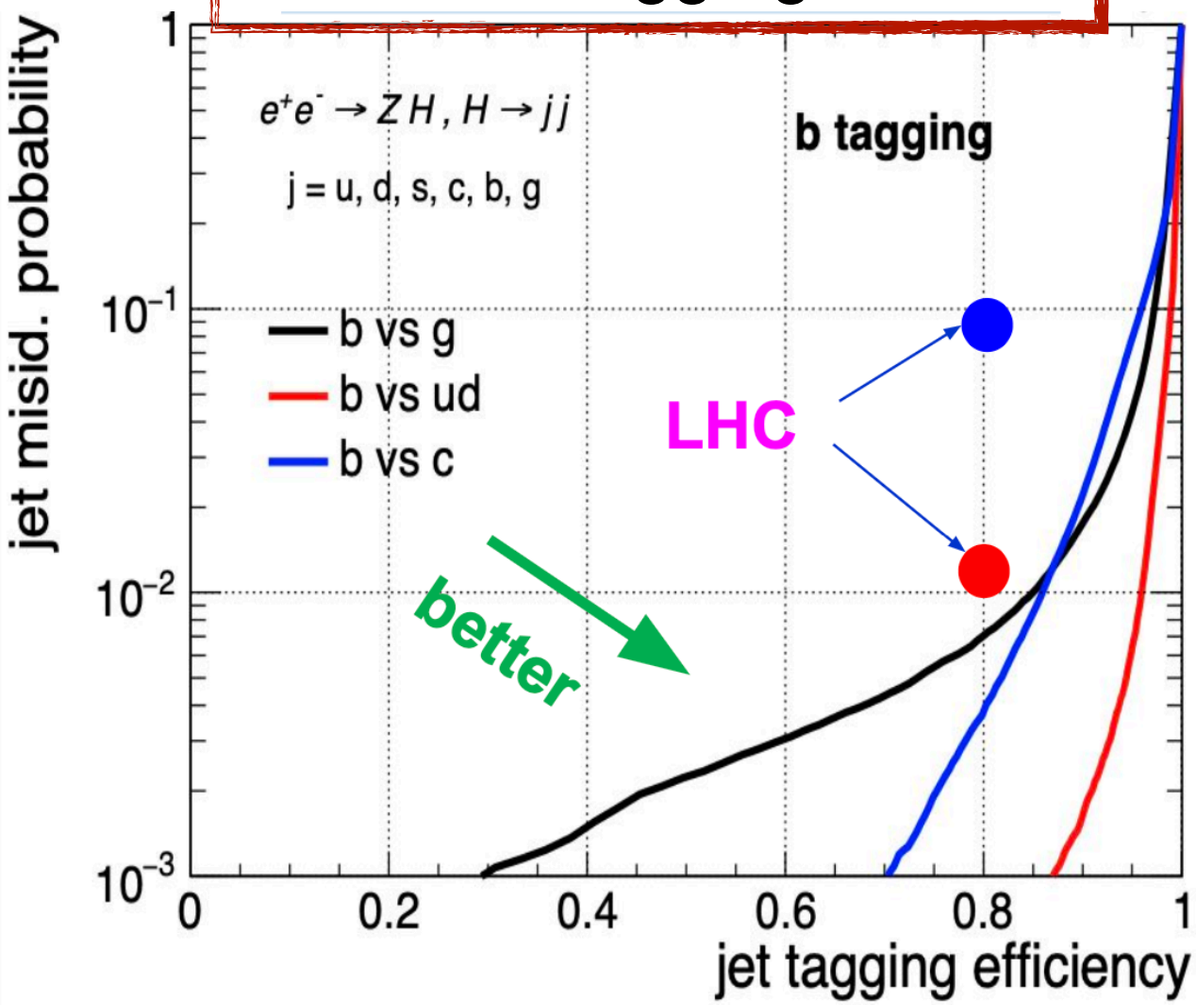
Strange $p_T = 45\,\text{GeV}$



Down $p_T = 45\,\text{GeV}$

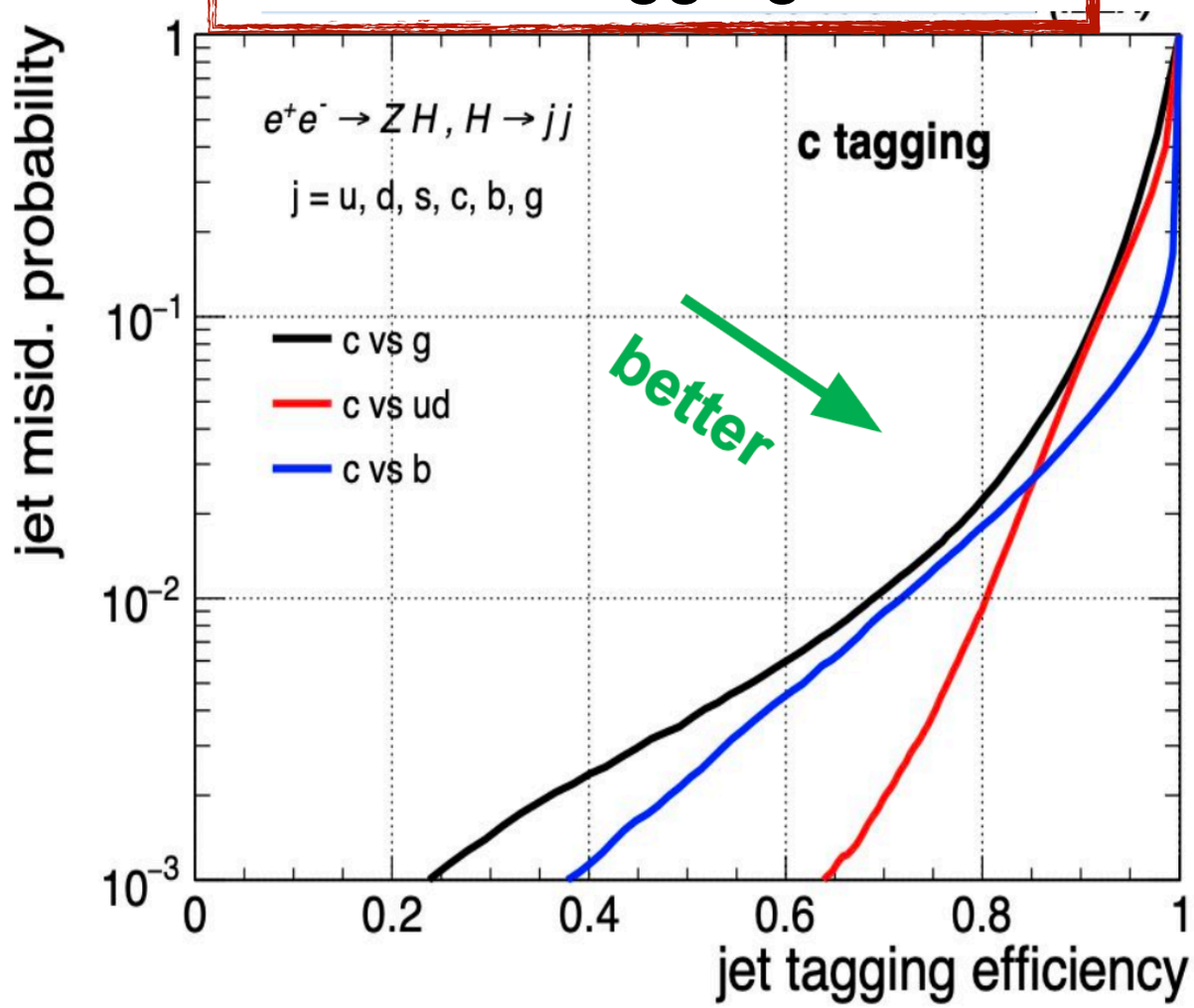
ParticleNet @FCC-ee: *b*/*c*-tagging Performance

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%

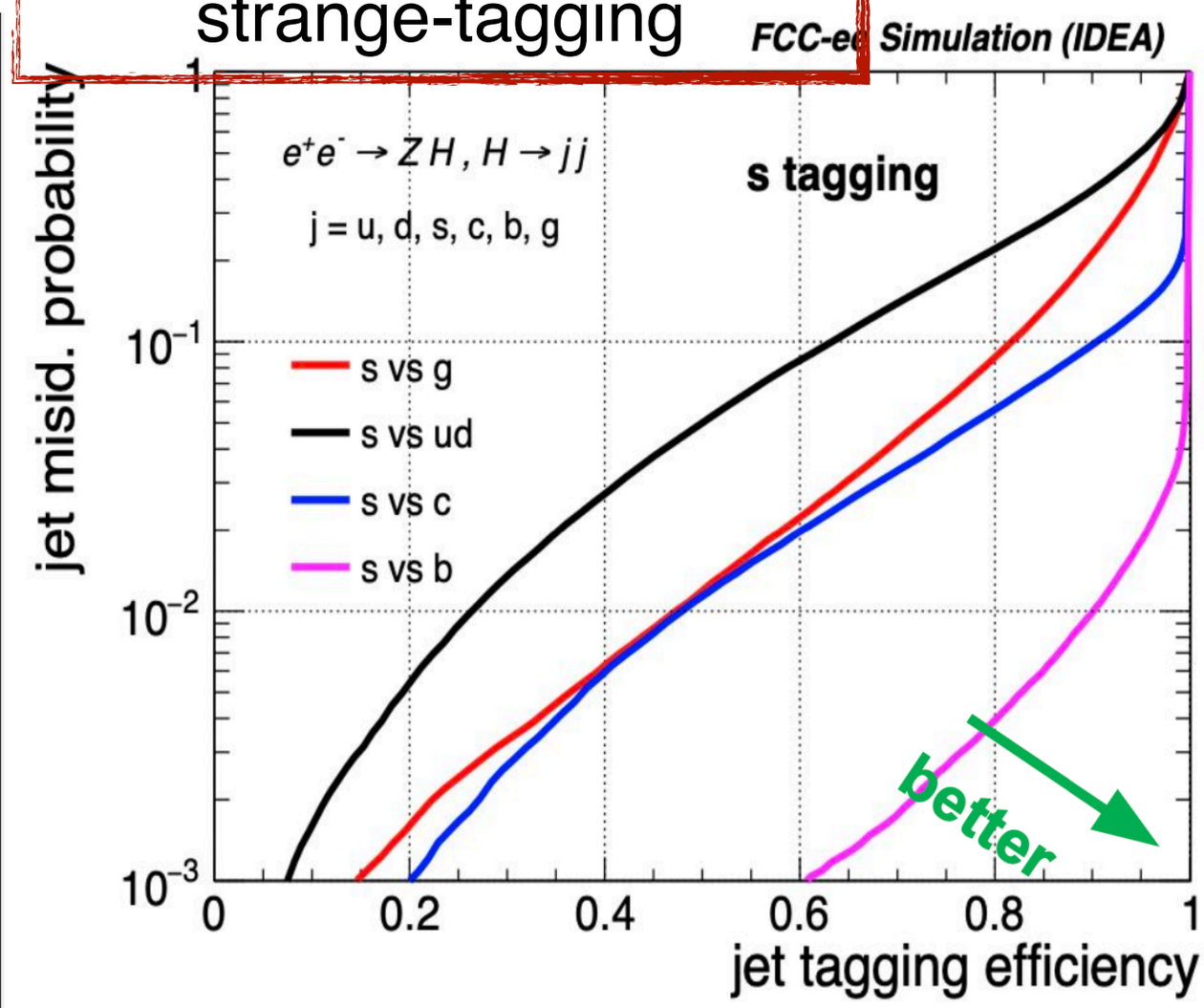
c-tagging



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

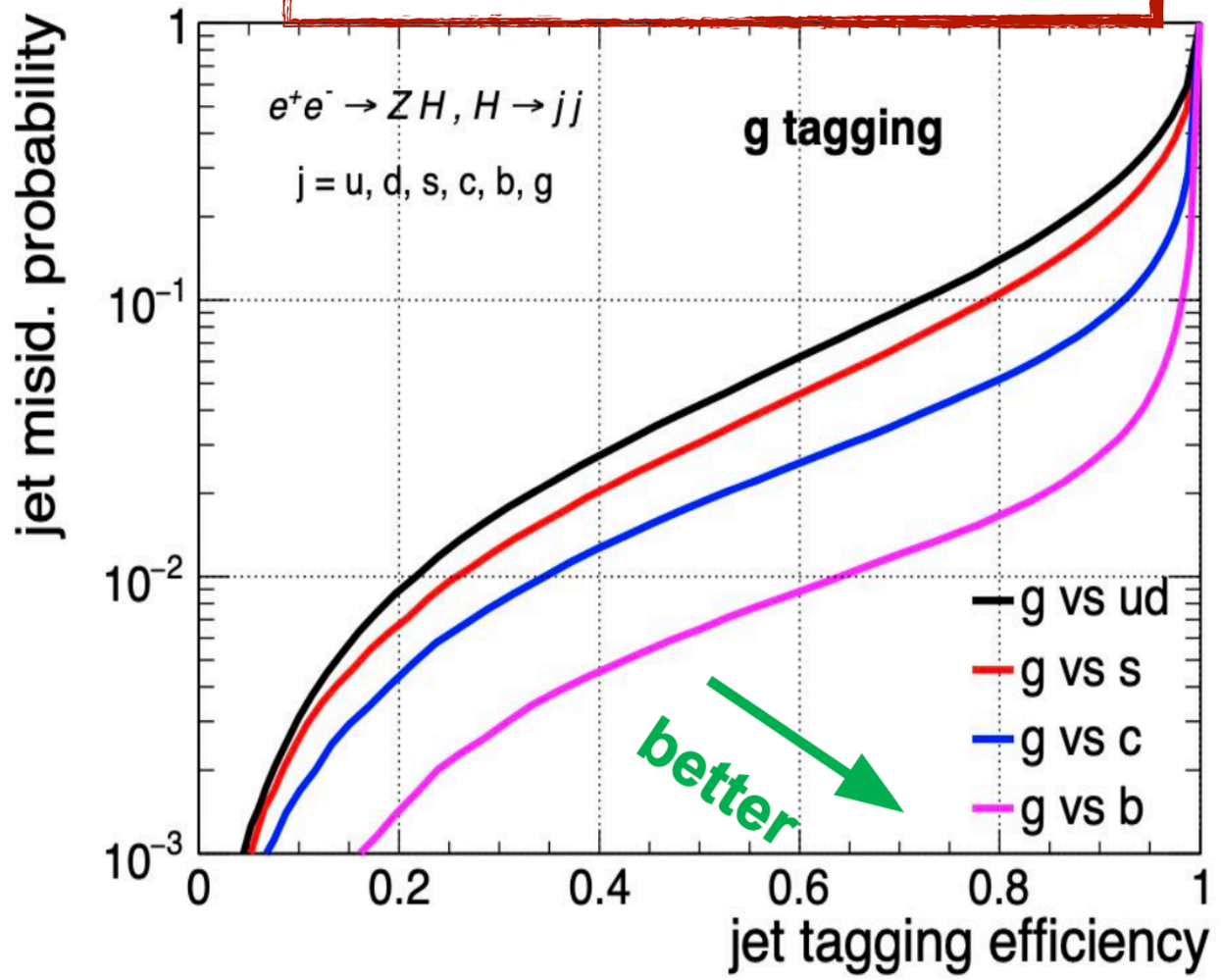
ParticleNet @FCC-ee: *s*/*g*-tagging Performance

strange-tagging



WP	Eff (s)	Mistag (g)	Mistag (ud)	Mistag (c)	Mistag (b)
Loose	90%	20%	40%	10%	1%
Medium	80%	9%	20%	6%	0.4%

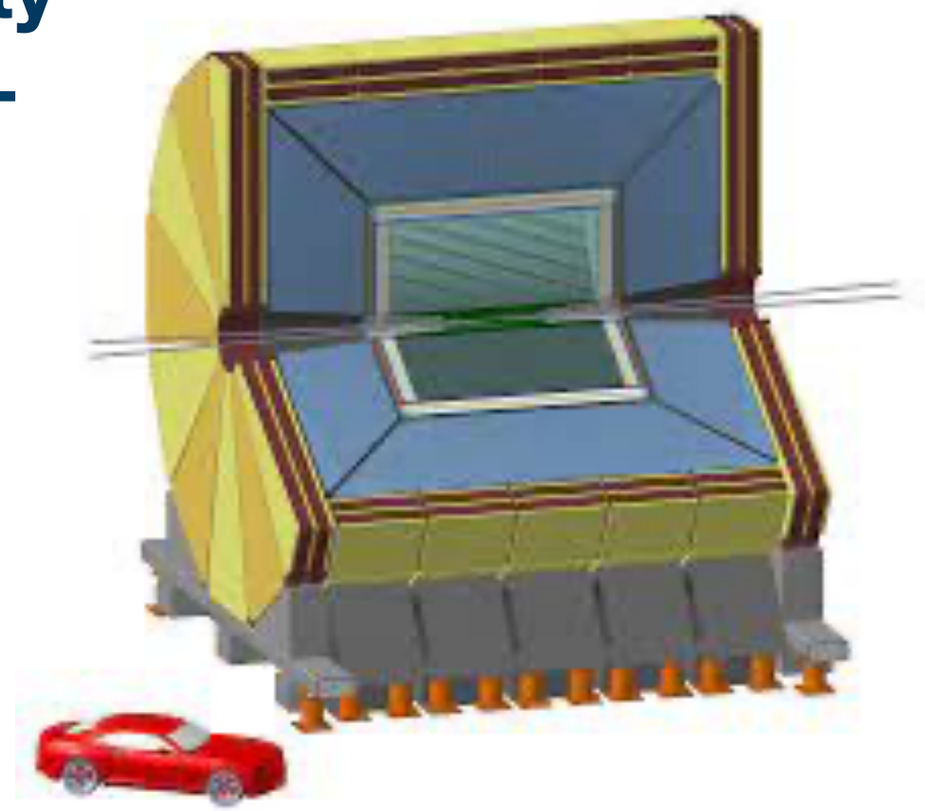
gluon-tagging

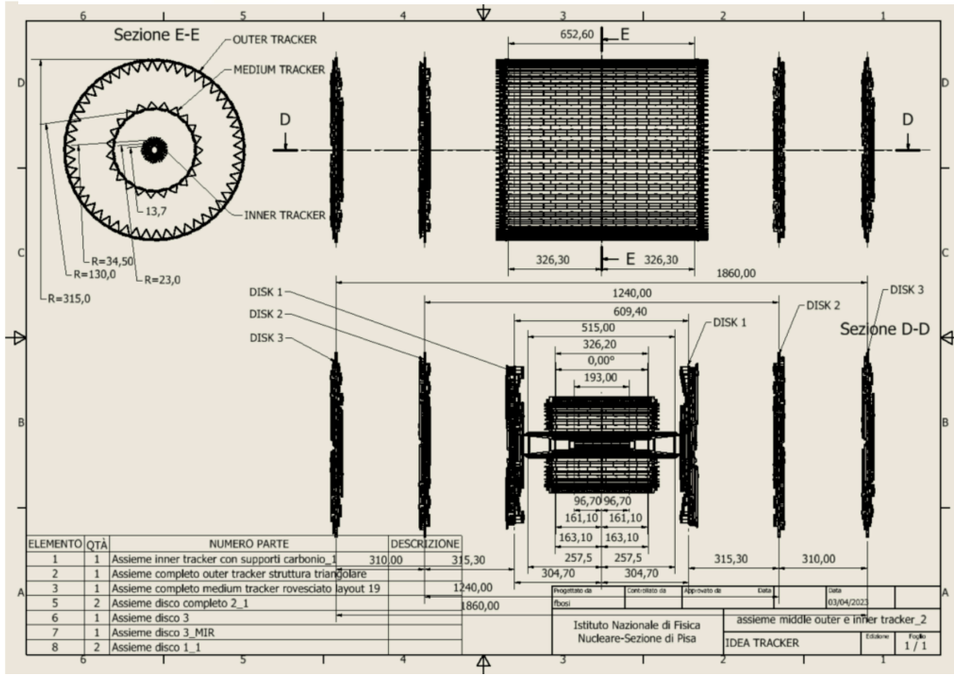


WP	Eff (g)	Mistag (ud)	Mistag (c)	Mistag (b)
Loose	90%	25%	7%	2.5%
Medium	80%	15%	5%	2%

The (IDEA) Tracker as an Opportunity

- Different possible detector scenarios, *tracker* particularly relevant to flavour-tagging
  - **PID capabilities**: timing, energy loss (gas/silicon)
  - **Amount (e.g. n. of layers)** & **quality of material**
  - **Hit resolution**
- Baseline IDEA detector as a well-established reference for detector-performance studies
  - Opportunity to access impact of detector configurations/ properties on physics performance
  - A lot already studied in the past [Eur. Phys. J. C 82, 646 (2022)]
  - **More & new studies on-going for Final State Report**
- Current IDEA pixel/tracking system:
  - beam pipe at 1cm, 3 *innermost* VTXD barrel layers: 1.2cm, 2cm, 3.15cm
  - PID: dN/dx + 30ps ToF system



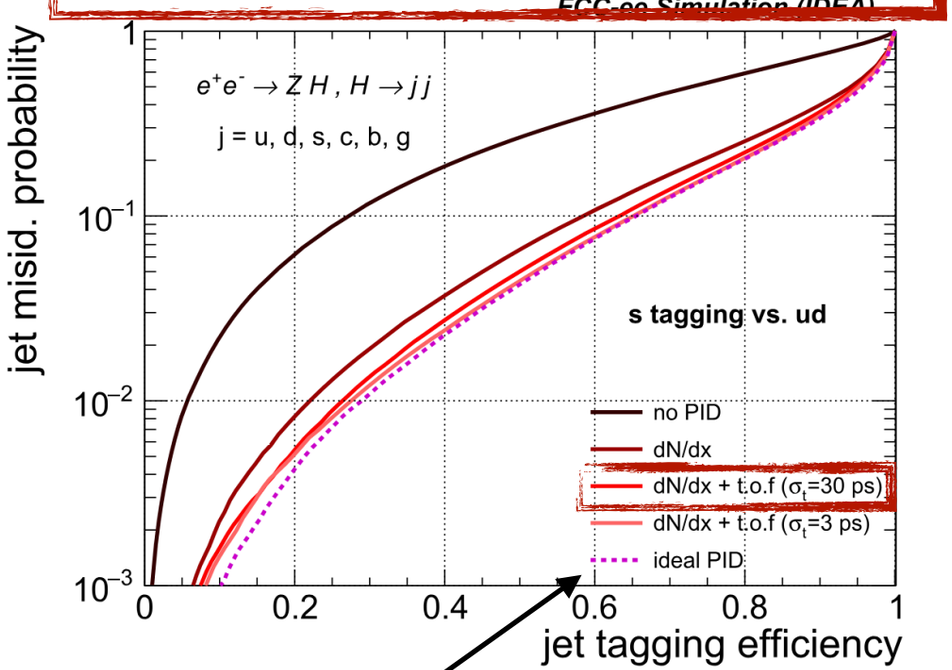


Latest IDEA tracker layout from F. Palla's talk

Impact of detector configurations

Eur. Phys. J. C 82, 646 (2022)

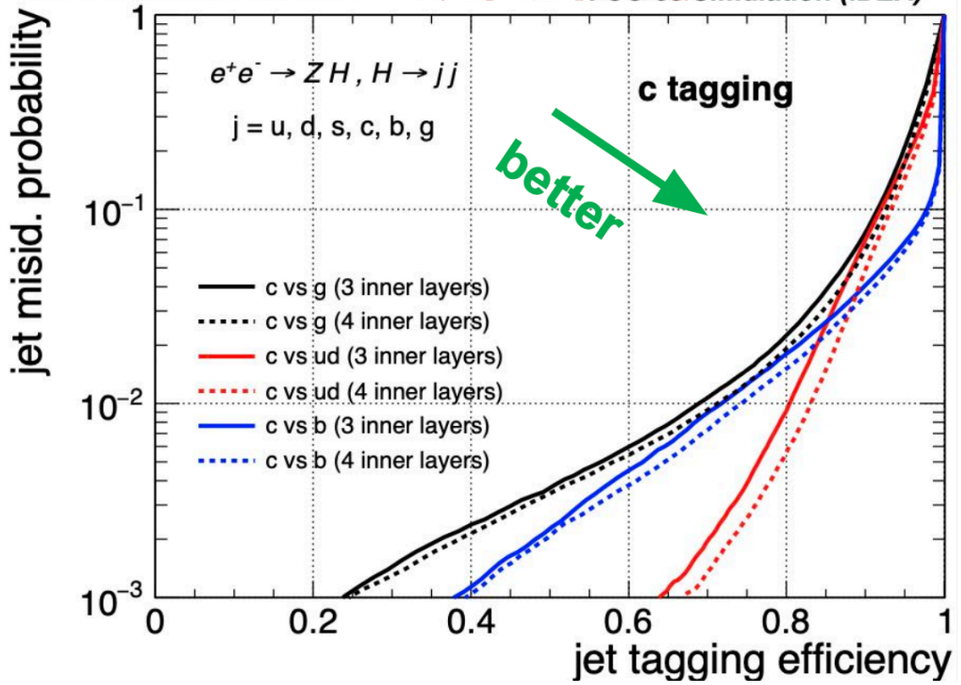
strange-tagging (PID)



"Ideal" PID from MC truth record

- dN/dx brings most of the gain additional gain w/ TOF (30ps)
  - TOF (3ps): marginal improvement
  - dN/dX + TOF(30ps) ~ perfect PID

c-tagging (PIX layers)

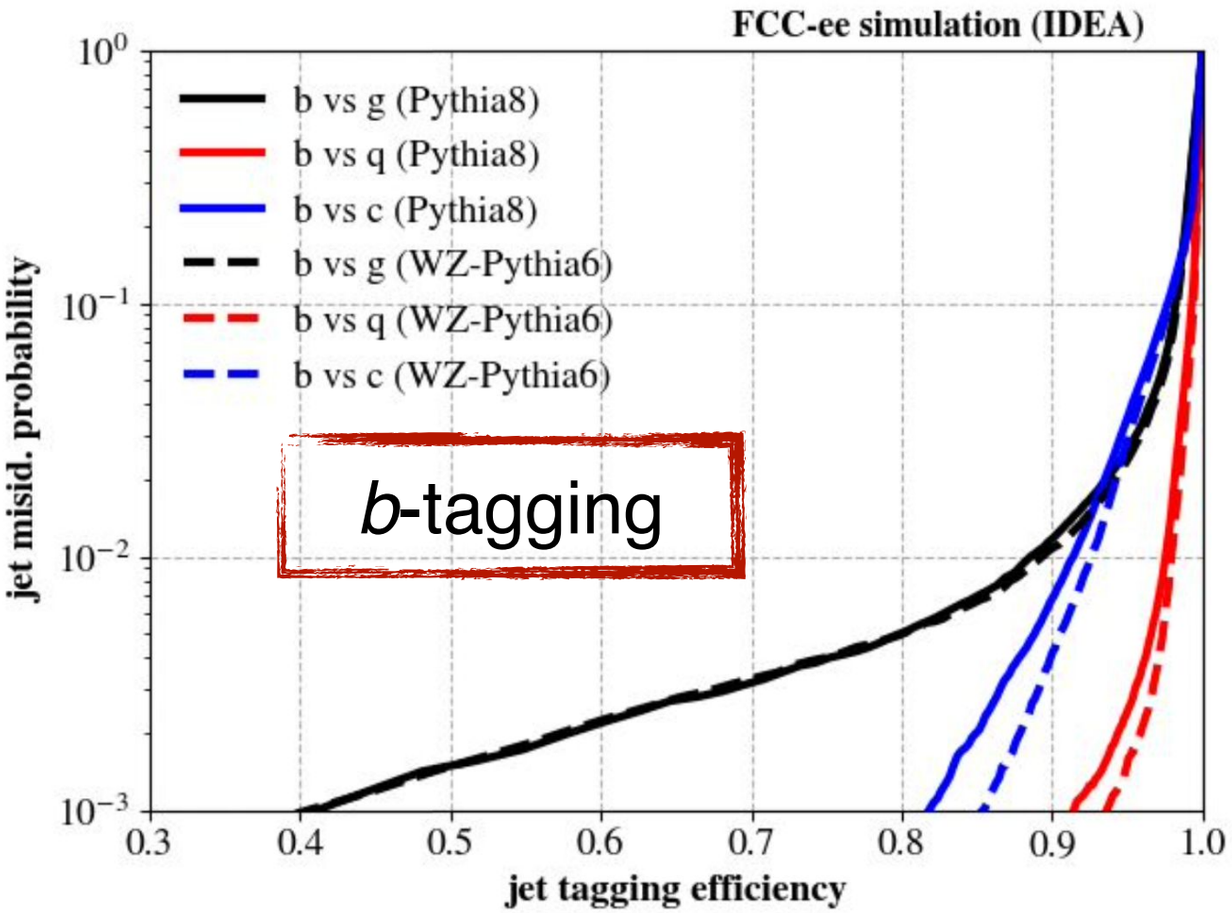


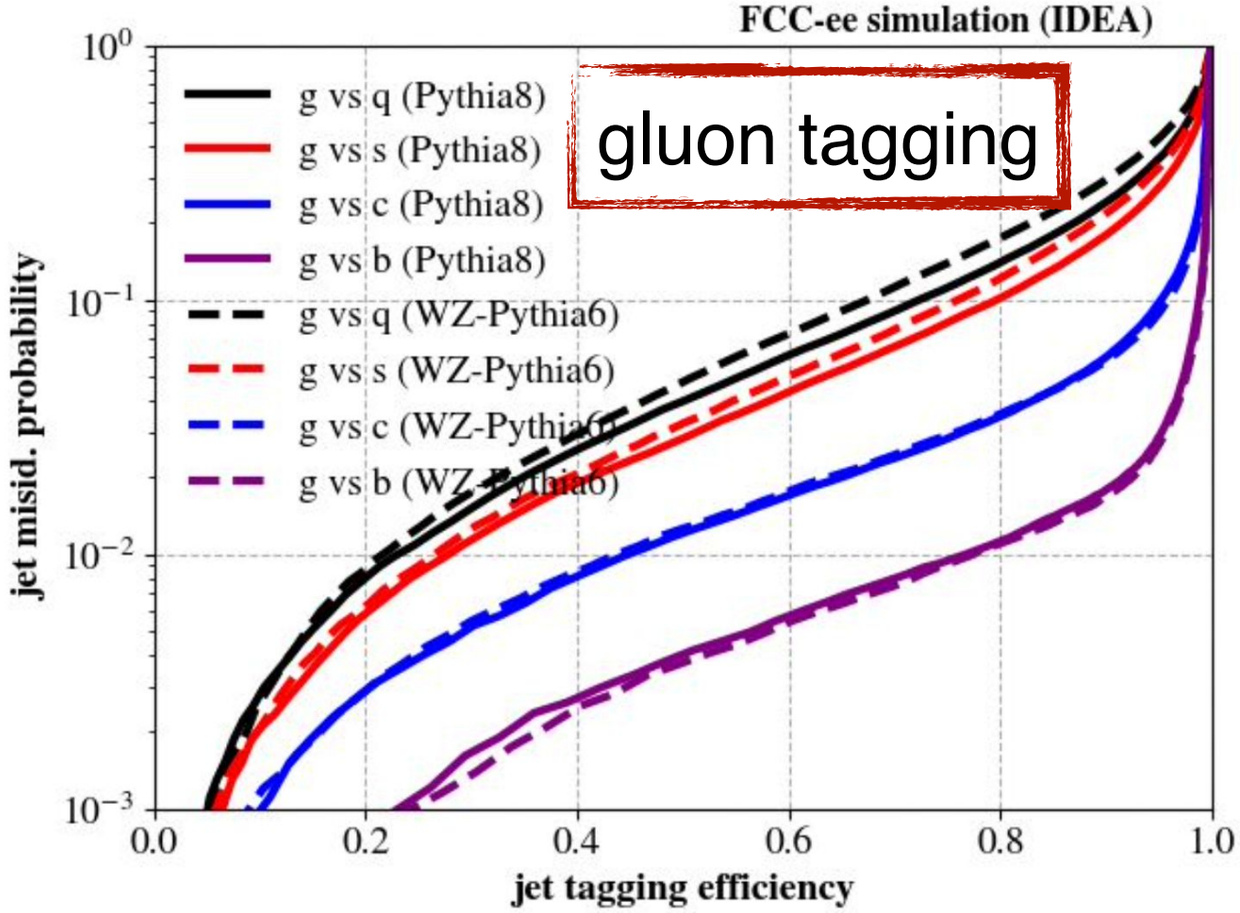
- Additional pixel layer 1 cm from beam pipe vs 1.5 cm (prev. detector layout):
  - improved BKG rejection in *c*-tagging
  - marginal/no improvement in *b*-tagging

Robustness against Modeling

Eur. Phys. J. C 82, 646 (2022)

- ParticleNet-*ee* trained using Pythia 8 samples
  - Tested on Pythia 8 samples (solid lines)
  - Tested on WZ+Pythia 6 samples (dashed lines)

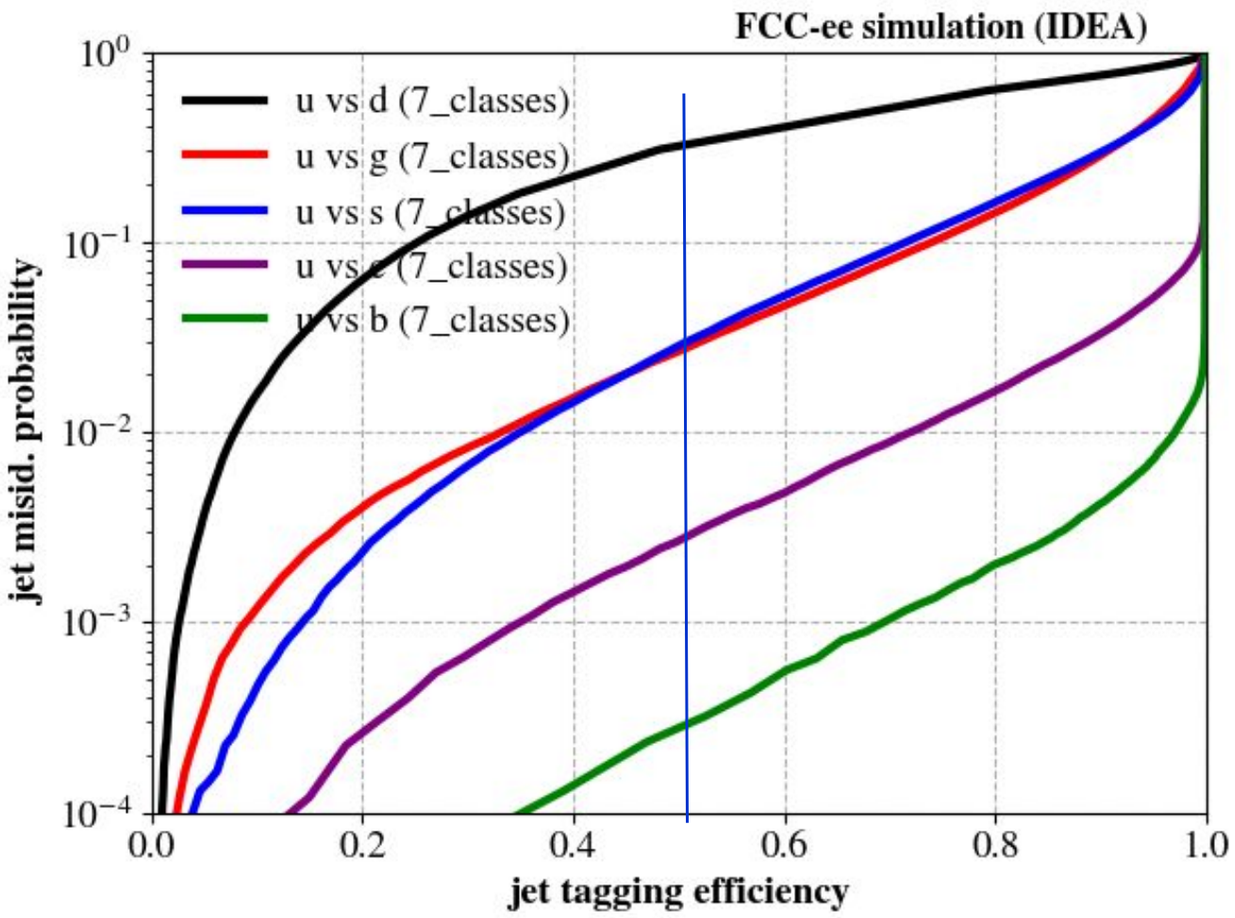




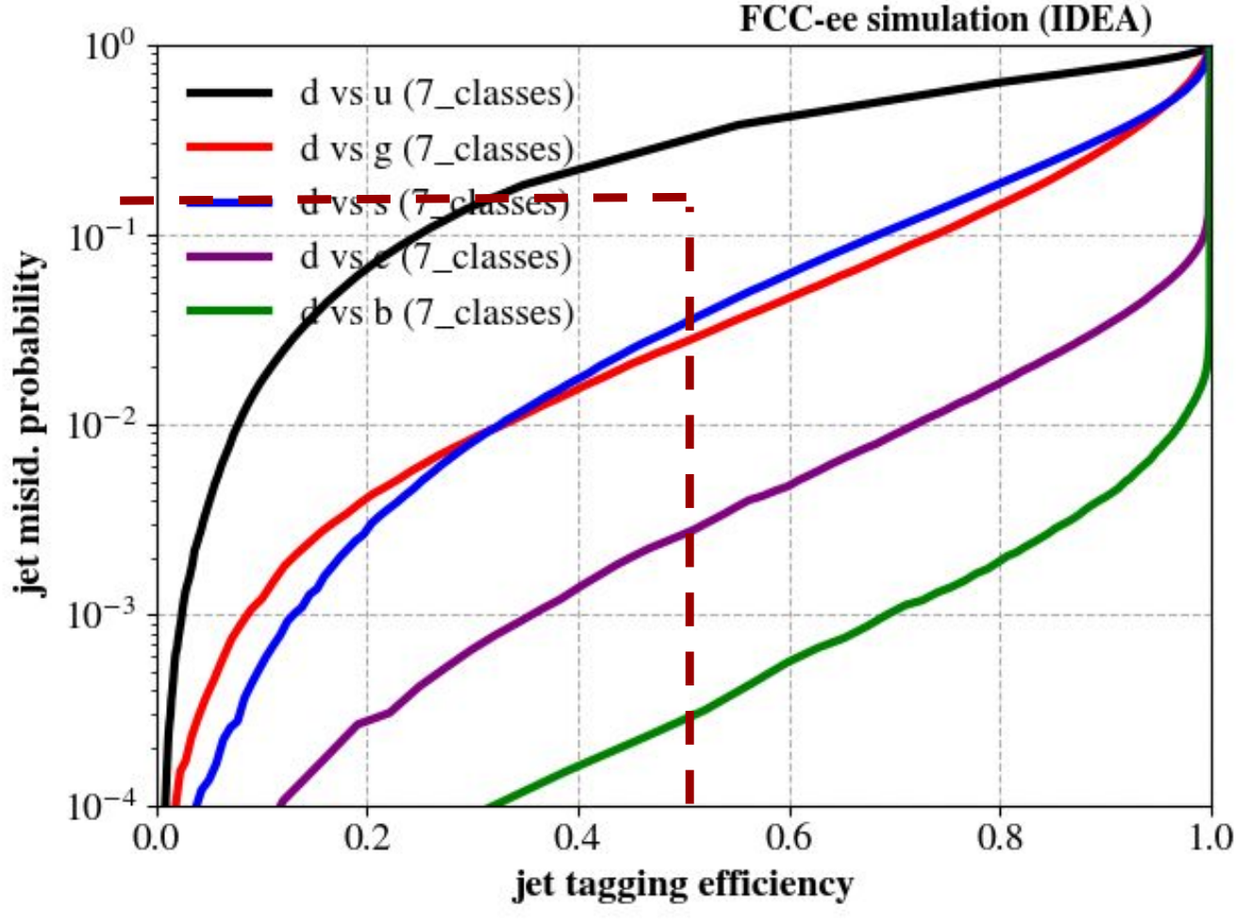
- Modest dependence on choice of generator
- More parton showers being explored (Sherpa, Herwig, etc…)

The Future: Up & Down Tagging!

Up-tagging



Down tagging



- Up vs. Down discrimination seems ~possible thanks to jet charge
- 50% signal efficiency for 30% bkg efficiency (better than random coin toss)

H->bb/cc/ss/gg Analysis Strategy in a Nutshell

- Signal signature: *H->jj, j=b,c,s,g,τ*
- Main background processes: *WW/ZZ/Z, qqH, H->WW, H->ZZ*
- Key ingredients:
  - Jet reconstruction: N=2 Durham kT exclusive algorithm
  - ParticleNet jet tagger with 4 categories: *b,c,s,g*
- Analysis:
  - Event pre-selection: lepton veto, cos(ϑ)
  - Categorisation based on tagger scores
  - Fit with floating 10% background normalisation uncertainty (to be constrained) & 4 signal strengths (i.e. *H->bb,cc,ss,gg*)

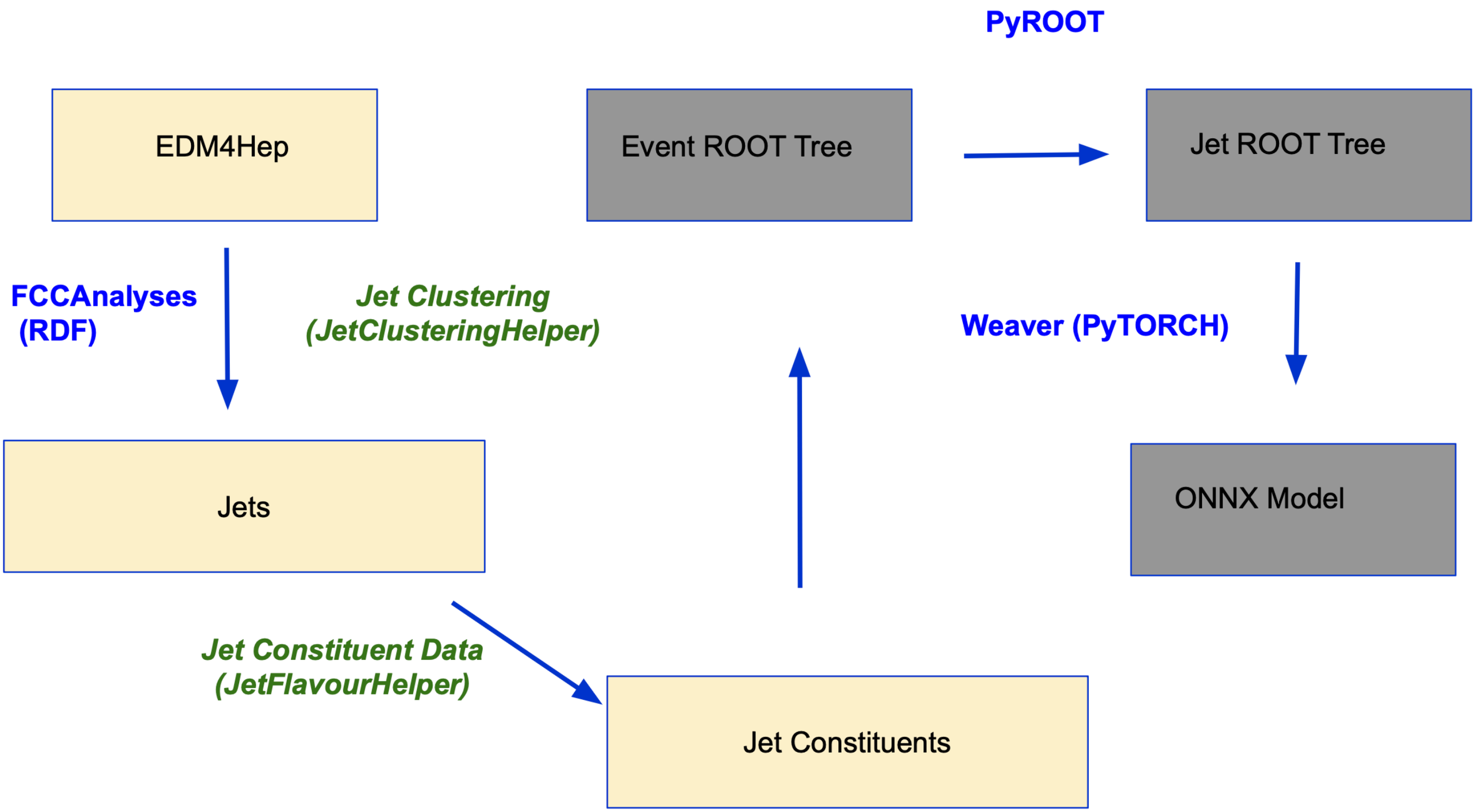
Results using only **vvH** channel:

Hgg : 1.1%
Hss : 150 %
Hcc : 2.7%
Hbb : 0.5%

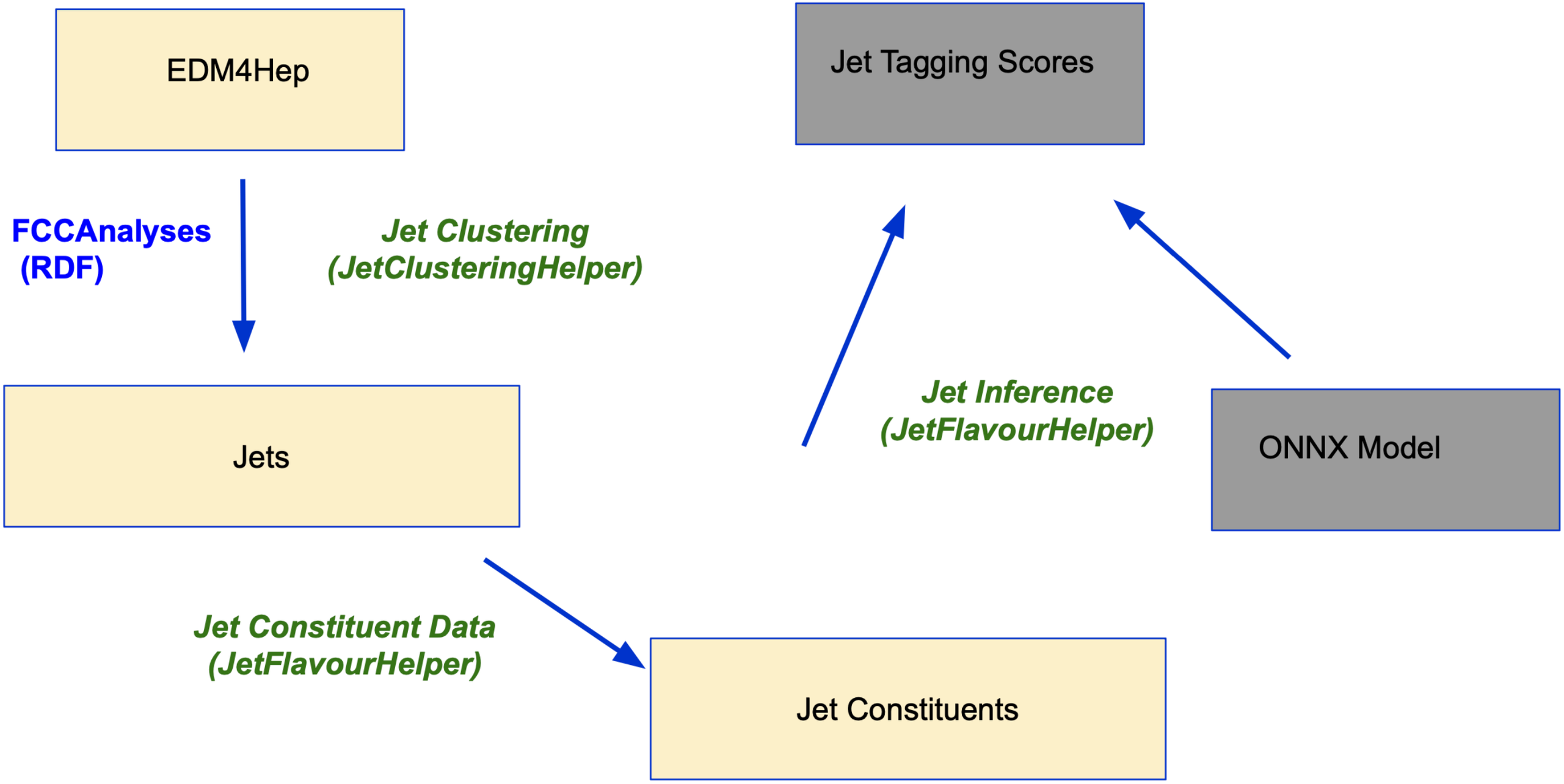
Event yields in the **S-like highest purity** category (with **122 < mvis < 128**) for **7.2 ab-1**

	Hss	Hgg	Hbb	Hcc	Htautau	HWW	HZZ	ZZ	WW	Zqq
N	10	10	0	0	0	8	10	300	150	80
S/B	1	1	0	0	0	1	1	1/30	1/15	1/10

Training the Model



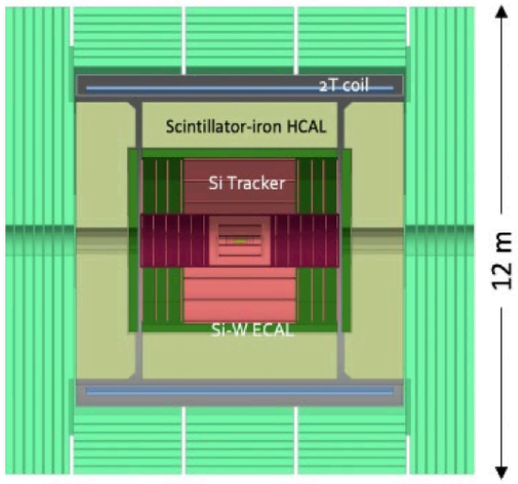
Inference


EDM4Hep
FCCAnalyses
(RDF)
Jet Clustering
(JetClusteringHelper)
Jets
Jet Constituent Data
(JetFlavourHelper)
Jet Constituents
Jet Tagging Scores
Jet Inference
(JetFlavourHelper)
ONNX Model

Current Detector Concepts

From Marc-André's talk

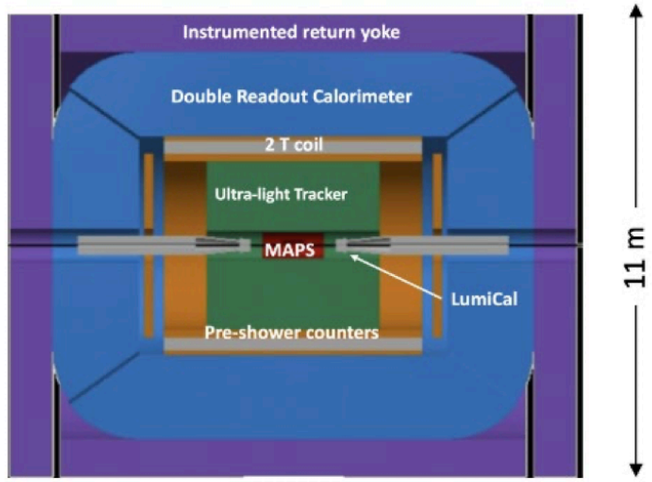
CLD



- Well established design
  - ILC -> CLIC detector -> CLD
- Full Si vtx + tracker
- CALICE-like calorimetry;
- Large coil, muon system
- Engineering still needed for operation with continuous beam (no power pulsing)
  - Cooling of Si-sensors & calorimeters
- Possible detector optimizations
  - σ_p/p, σ_E/E
  - PID ($\mathcal{O}$(10 ps) timing and/or RICH)?
  - …

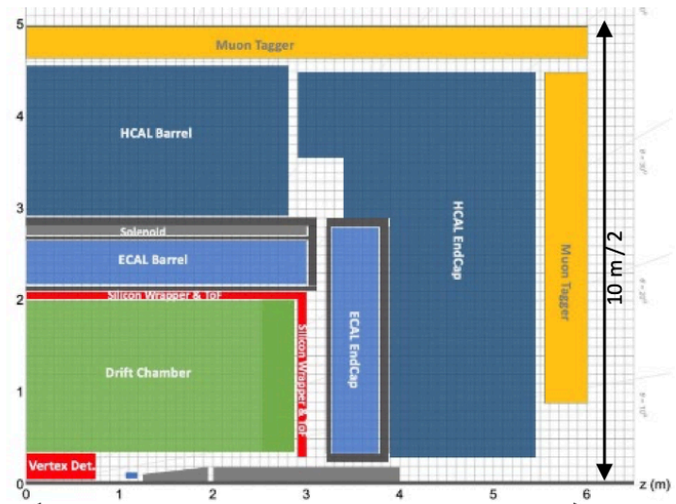
CDR

IDEA



- A bit less established design
  - But still ~15y history
- Si vtx detector; ultra light drift chamber with powerful PID; compact, light coil;
- Monolithic dual readout calorimeter;
  - Possibly augmented by crystal ECAL
- Muon system
- Very active community
  - Prototype designs, test beam campaigns, …

ALLEGRO

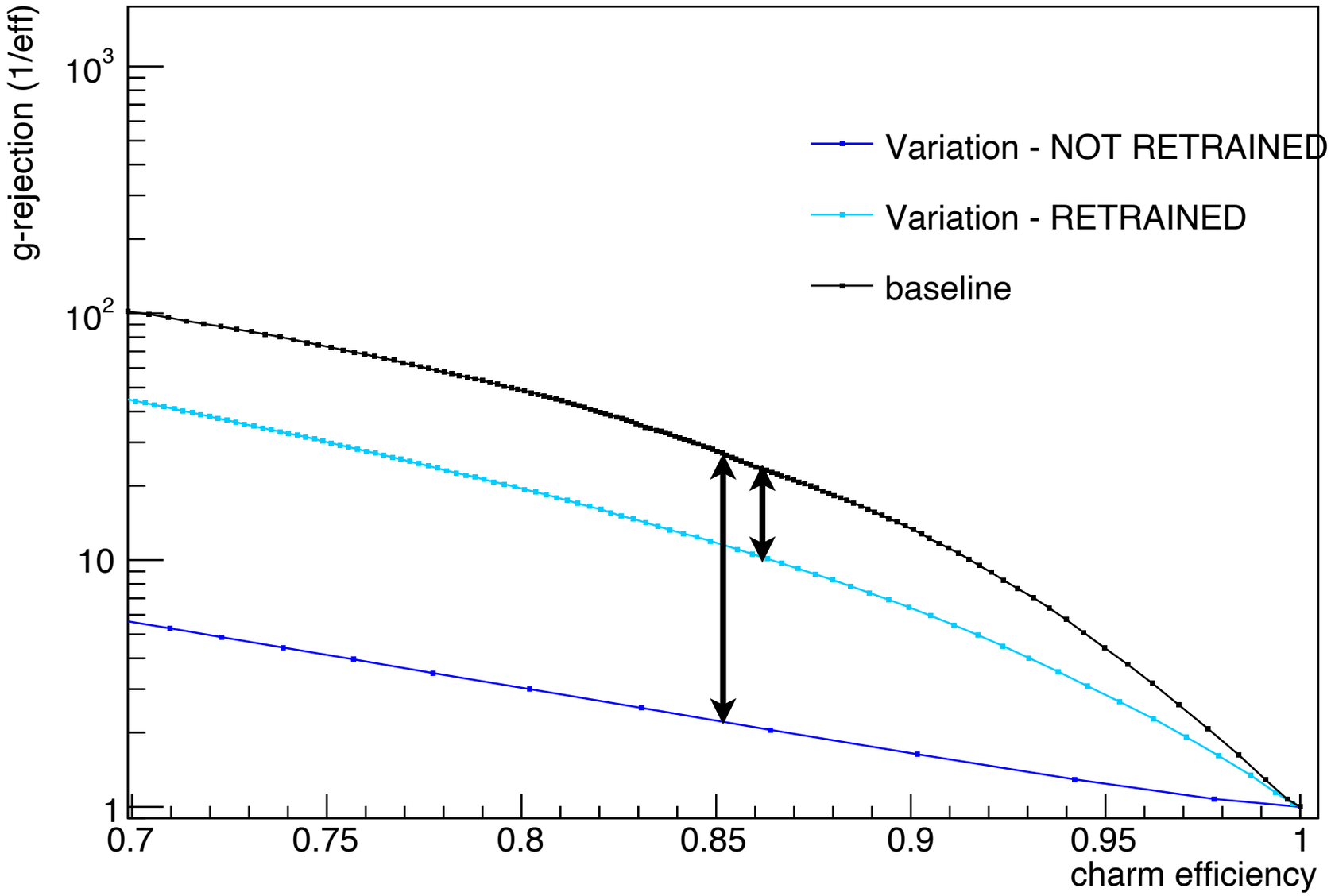


- The "new kid on the block"
- Si vtx det., ultra light drift chamber (or Si)
- High granularity Noble Liquid ECAL as core
  - Pb/W+LAr (or denser W+LKr)
- CALICE-like or TileCal-like HCAL;
- Coil inside same cryostat as LAr, outside ECAL
- Muon system.
- Very active Noble Liquid R&D team
  - Readout electrodes, feed-throughs, electronics, light cryostat, …
  - Software & performance studies

FCC-ee CDR: https://link.springer.com/article/10.1140/epjst/e2019-900045-4

Brookhaven National Laboratory

Why is Retraining Necessary?



- Obviously, given a detector configuration, ParticleNet would be trained against it
- Re-training allows recovering of (a significant) part of drop in performance
  - **Need re-training for fair & meaningful performance assessment of each point in the detector-configuration space**