Jet flavor identification for FCCee

ECFA Higgs Factories: Reconstructions, 2023

Dolores Garcia Marti
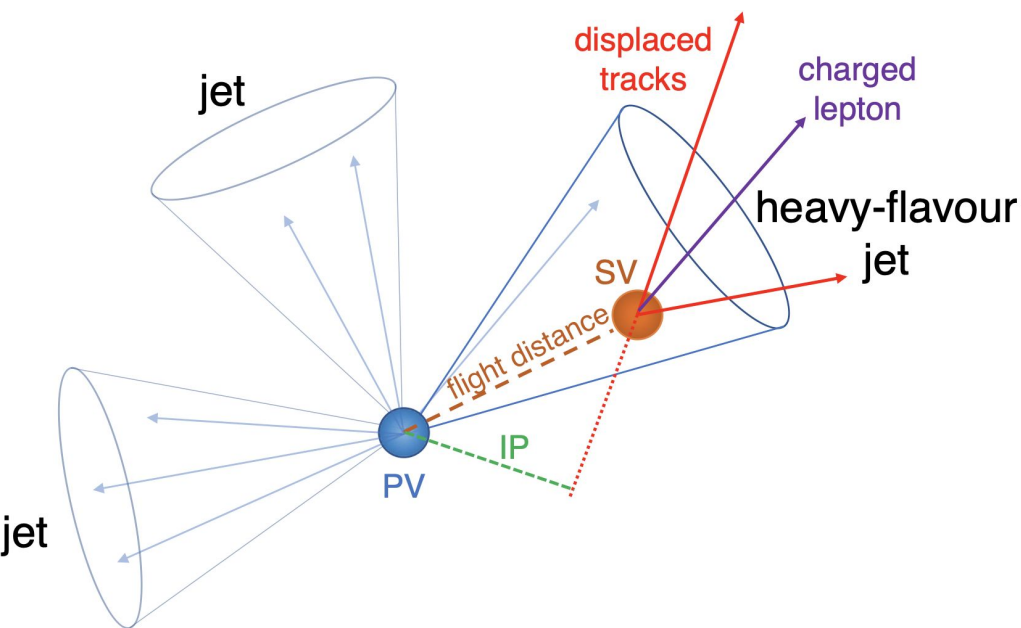
Physics motivation

- Flavour tagging essential for the e^+e^- program, e.g.:

  - Higgs Sector:
    - (HL-)LHC can access 3rd gen. couplings and a few of 2nd generation
    - Future e^+e^-: Measure Higgs particle properties and interactions in challenging decay modes
      - E.g. cc, 1st gen quarks/fermions, gg [?]

  - Top quark physics [if E_{CM} sufficient]
    - Precise determination of top properties [mass, width, Yukawa]

  - QCD Physics
    - strong coupling (a_s), event shapes ..
    - modelling of hadronization, MC tuning, …

  - ….
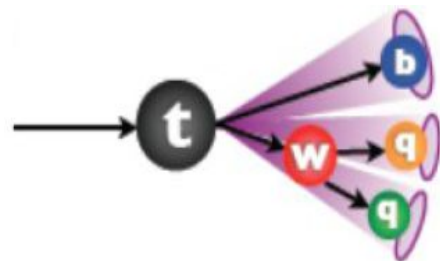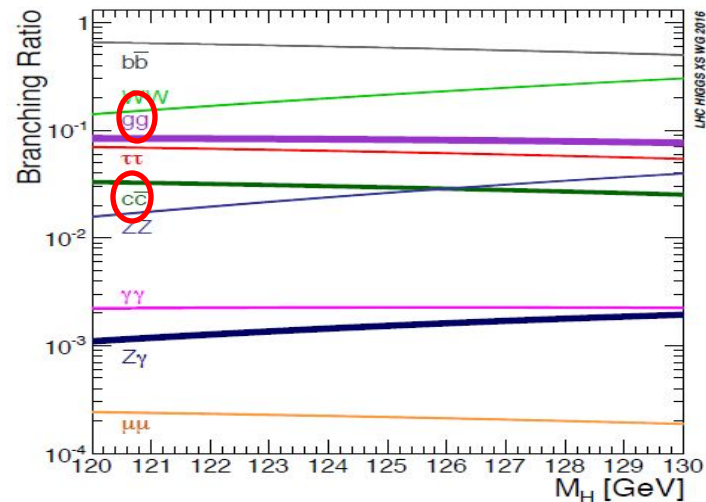
Basics of flavour tagging (b/c)

Detector constraints:
Need power pixel/tracking detectors
- Good spatial resolution
- As little material as possible
- Precise track alignment

- Large lifetime
  - b (c) lifetime ~1 ps (~0.1ps)
  - b **(c)** decay length: ~500 μm
  - (2-3) mm for ~50 GeV boost

- Displaced vertices/tracks
  - Large impact parameters
  - Tertiary vertices when B hadron decays to C hadron

- Large track multiplicity
  - ~5 (~2) charged tracks/decay

- Presence of non-isolated e/μ
  - ~20 (10)% in B (C) decays

Input variables

- Comparison of input distributions for different jet flavors

Constituent relative energy
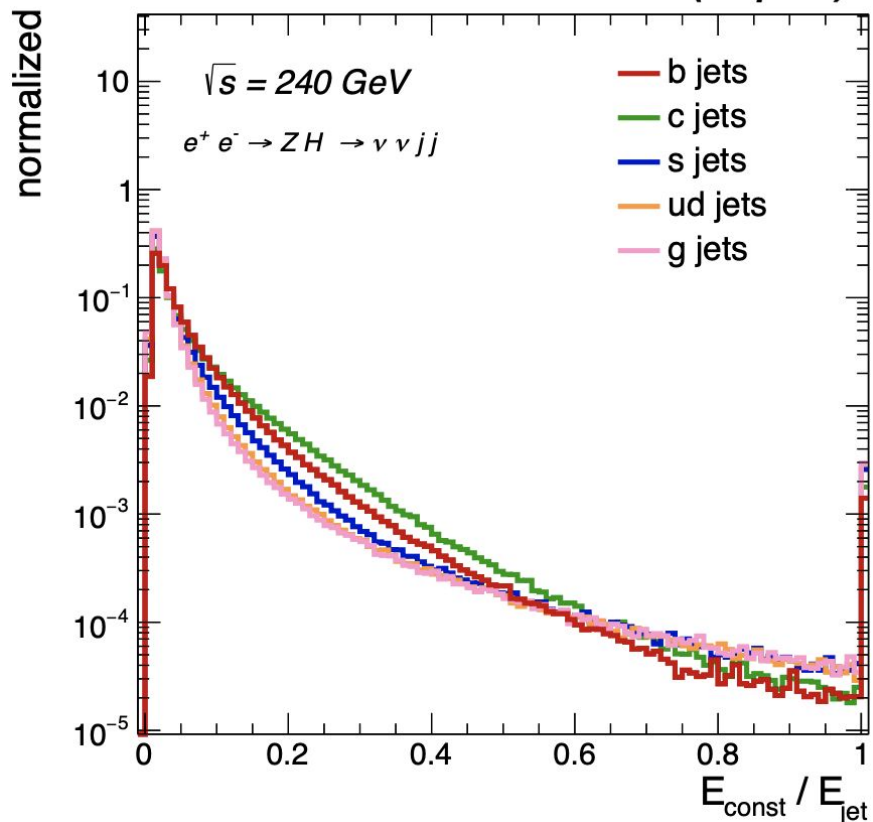
Impact parameter (d_0)

FCC-ee simulation (Delphes)

$\sqrt{s} = 240 \; GeV$

$e^+ e^- \to ZH \to \nu\nu jj$

— b jets
— c jets
— s jets
— ud jets
— g jets

E_{const} / E_{jet}

FCC-ee simulation (Delphes)

$\sqrt{s} = 240 \; GeV$

$e^+ e^- \to ZH \to \nu\nu jj$

— b jets
— c jets
— s jets
— ud jets
— g jets

2D signed IP (mm)
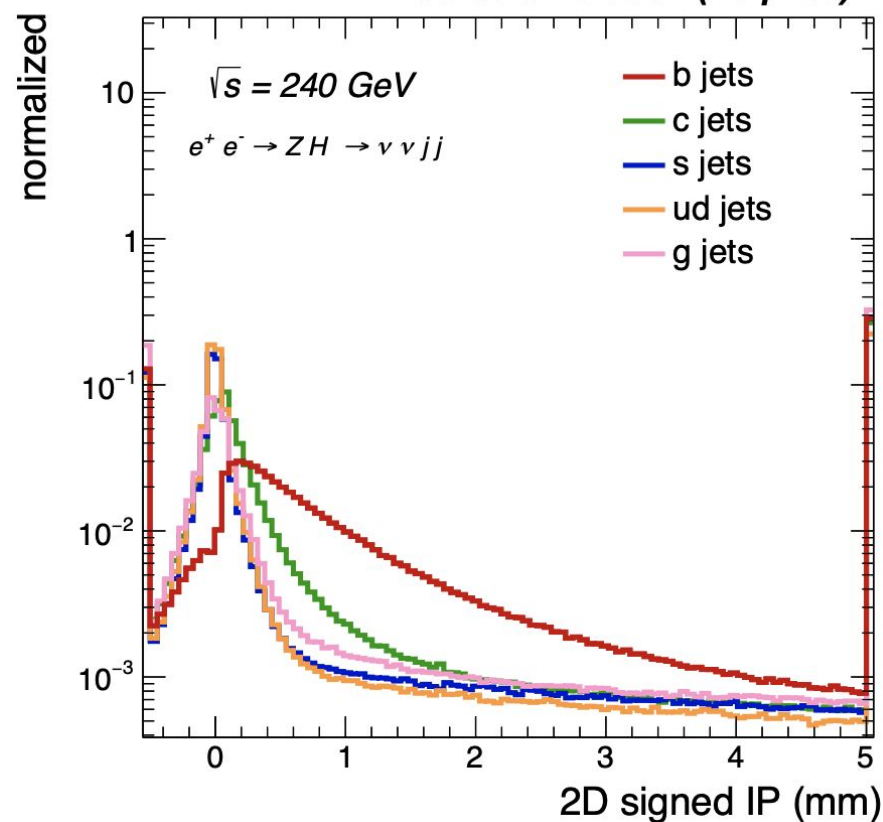
Basics of flavour tagging (strange)

[2003.09517]

Momentum weighted fraction:
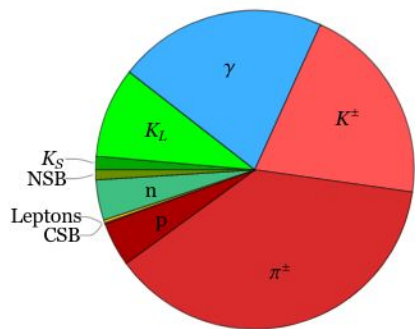
Strange $p_T = 45\,\mathrm{GeV}$
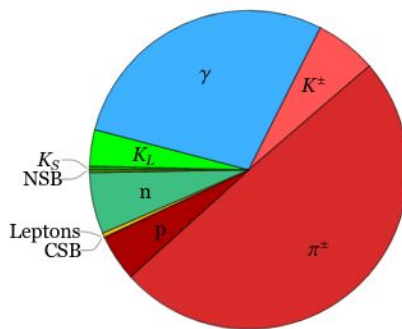
Down $p_T = 45\,\mathrm{GeV}$
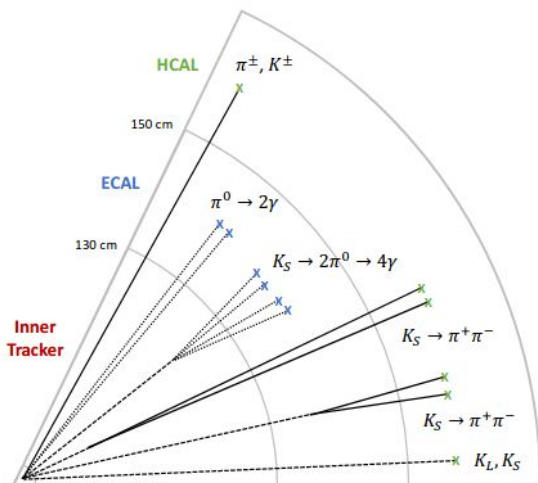
- Large Kaon content
  - Charged Kaon as track:
    - K/pi separation
      - TOF
      - dEdx/dNdx
  - Neutral Kaons:
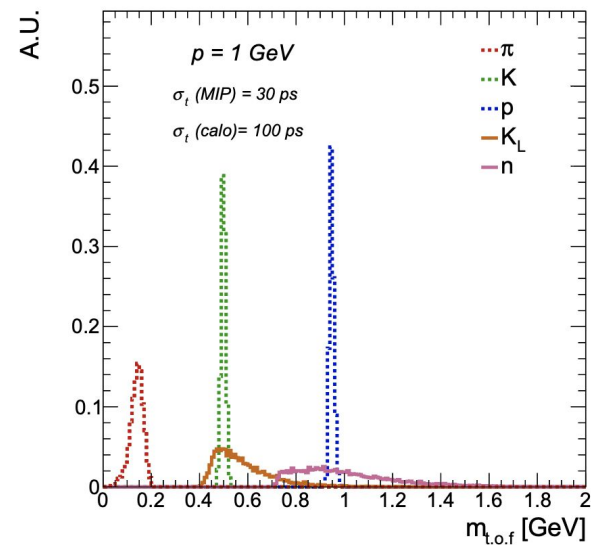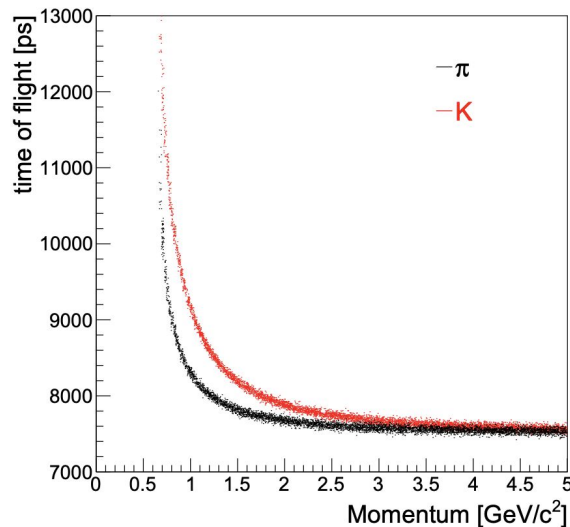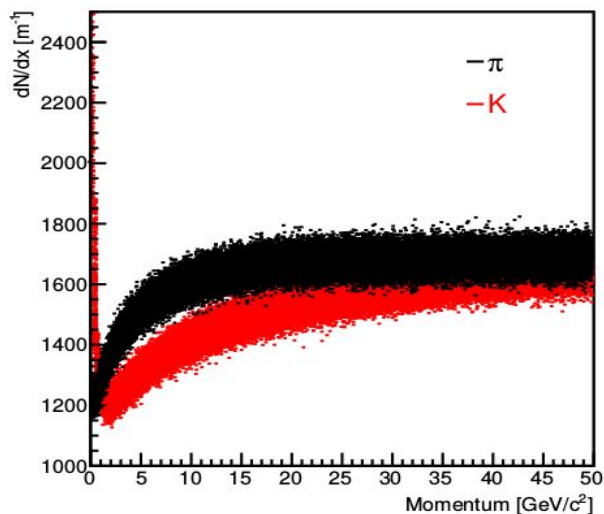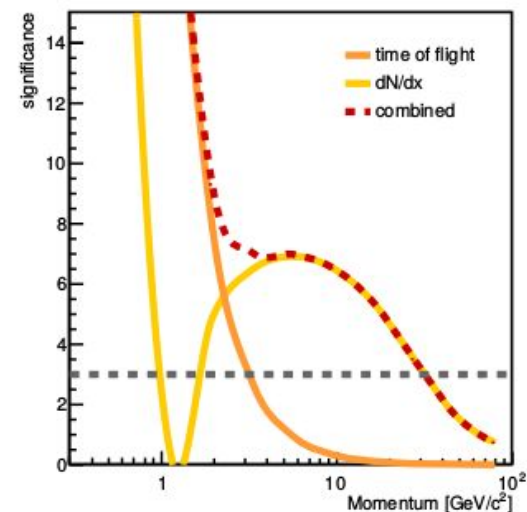   - $K_S \rightarrow \pi\pi$
      - Displaced 2 track vertex
      - 4 photons
    - K_L
      - TOF vs n ?

Detector constraints:

- timing detectors
- charged energy loss (gas/silicon)
- cherenkov detectors

Particle ID: dN/dx and ToF

- Count number of **primary ionization** clusters along track path
- ToF results in good K/π separation at low-momenta
- Modules added in Delphes

Designing a Graph-based tagger

- **Jet representation:** critical for powerful jet tagging algorithms

  - **In theory:** A spray of particles produced by the hadronization of q and g

  - **Experimentally:** A cone of reconstructed particles in the detector

- Reminder: Current and future experiments have / will have a **PF-based** event reconstruction

  - **Output:** mutually exclusive list of particles

    - Rich set of info/particle

      o Energy/momentum, position

      o Displacement, particle type

      o timing

      o …

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

- **Until recently:** Jet taggers based on human-inspired higher-level observables

  - Inputs to cut-based or simple ML-based algorithms

- Move to **particle-based jet tagging**: i.e. exploit directly the full list of jet constituents (ReconstructedParticles) and **new advances in ML**

Full list of input variables

Variable	Description
Kinematics	
$E_{\mathrm{const}}/E_{\mathrm{jet}}$	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
$\mathrm{SIP}_{2\mathrm{D}}$	signed 2D impact parameter of the track
$\mathrm{SIP}_{2\mathrm{D}}/\sigma_{2\mathrm{D}}$	signed 2D impact parameter significance of the track
$\mathrm{SIP}_{3\mathrm{D}}$	signed 3D impact parameter of the track
$\mathrm{SIP}_{3\mathrm{D}}/\sigma_{3\mathrm{D}}$	signed 3D impact parameter significance of the track
$d_{3\mathrm{D}}$	jet track distance at their point of closest approach
$d_{3\mathrm{D}}/\sigma_{d_{3\mathrm{D}}}$	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
$m_{\mathrm{t.o.f.}}$	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

Designing a Graph-based tagger

- **Jet:** intrinsically <u>unordered set</u> of particles with relationships b/w the particles
  - i.e. human-chosen ordering not optimal

- A very active research area in ML community: **Point clouds**

Represent the object as a set of "points"

Group points based on similarity [usually using ML]
e.g. Identify parts
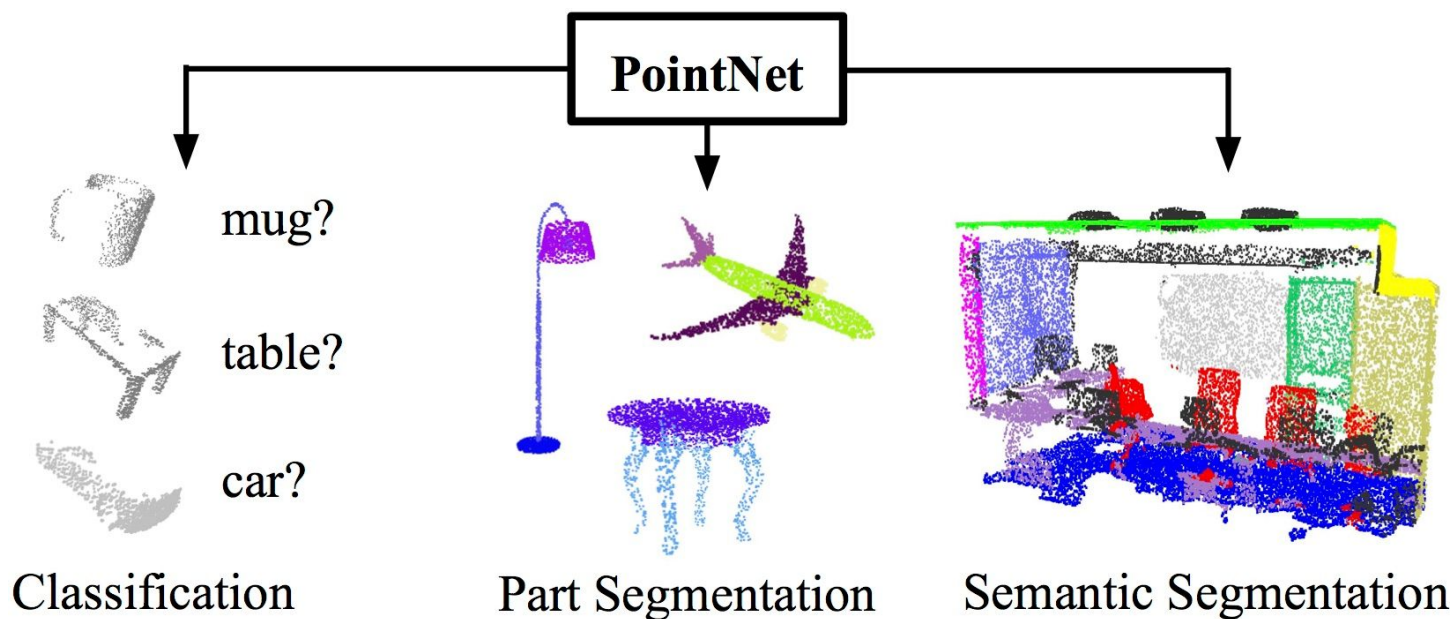
[Ref] PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, C.R.Qi et all, 2017

ParticleNet(-ee)

- Jet representation: *"Point Cloud"→"Particle Clouds"*

  - Treat the jet as an <u>unordered set of particles</u>

- Algorithm design: Graph Neural Networks

  - Particle cloud represented as a graph

    - Each particle: **node** of the graph; Connections between particles: the **edges**

- Follow a hierarchical learning approach

  - First learn local structures → then move to more global ones
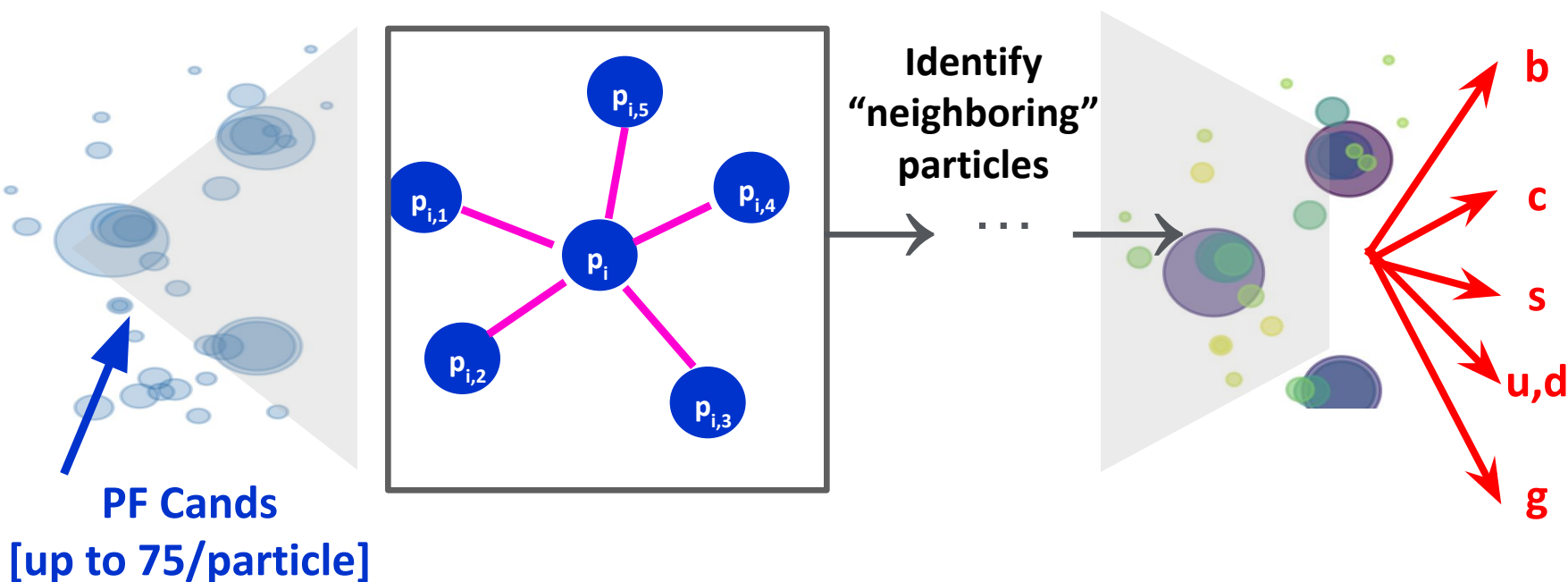
Identify "neighboring" particles

PF Cands [up to 75/particle]

b

c

s

u,d

g

Designing a Graph-based tagger

- Last step: **Learn** from the graphs

  - Follow a **hierarchical learning** approach:
    First learn <u>local</u> structures and then more <u>global</u> ones

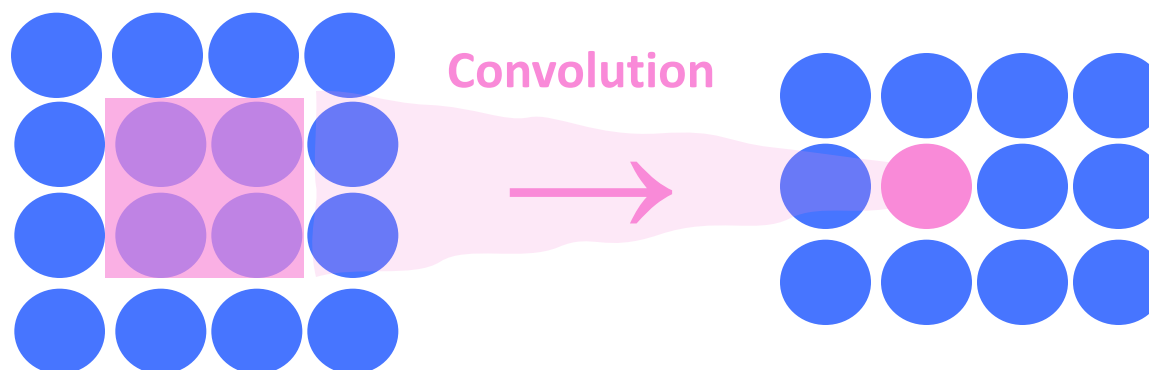- Convolution operations proven to be very powerful

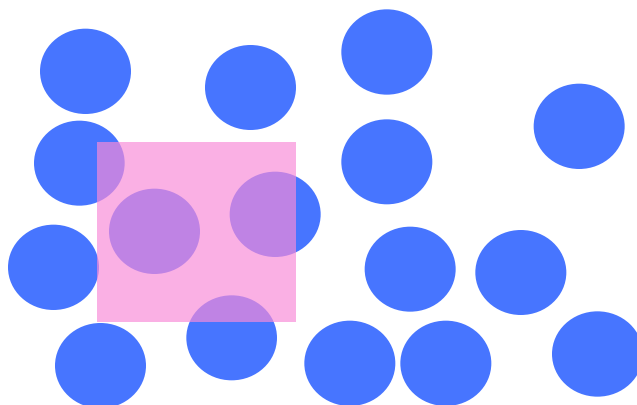**Fixed grid:**

Convolution
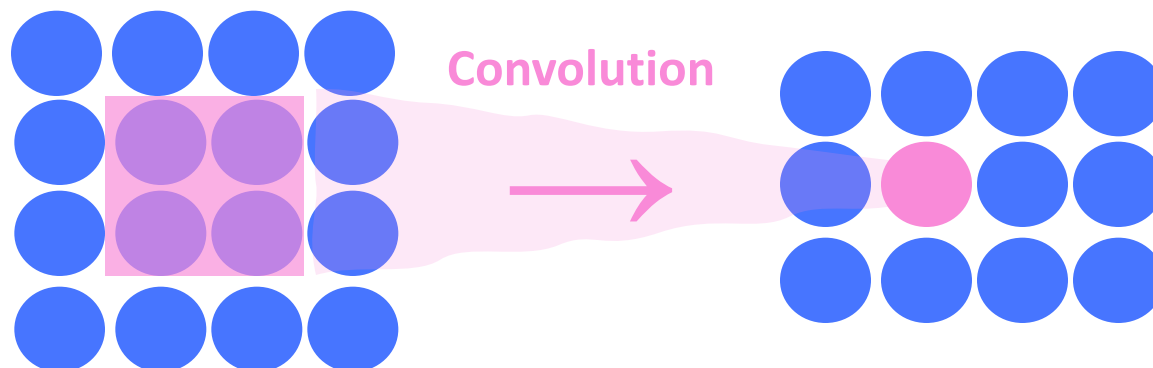
Designing a Graph-based tagger

- Last step: **Learn** from the graphs

  - Follow a **hierarchical learning** approach:
    First learn <u>local</u> structures and then more <u>global</u> ones

- Convolution operations proven to be very powerful

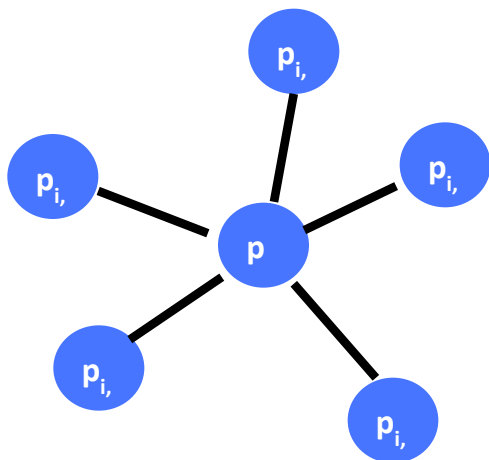**Fixed grid:**

Convolution

point/particle cloud:

… but not straightforward on point/particle clouds
- Irregular and unordered sets
- Requires a permutation invariant convolutional operation

EdgeConv: Convolution on point clouds

- Find the *k-nearest neighbors* of each point
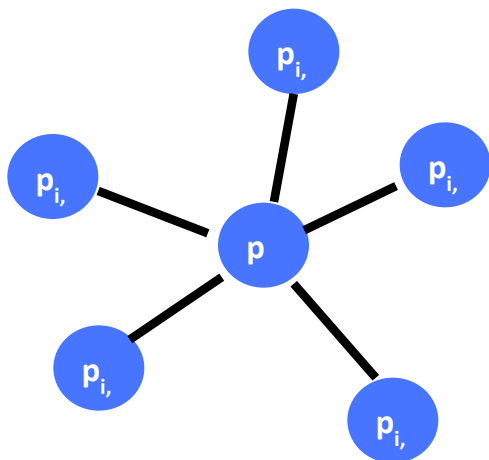
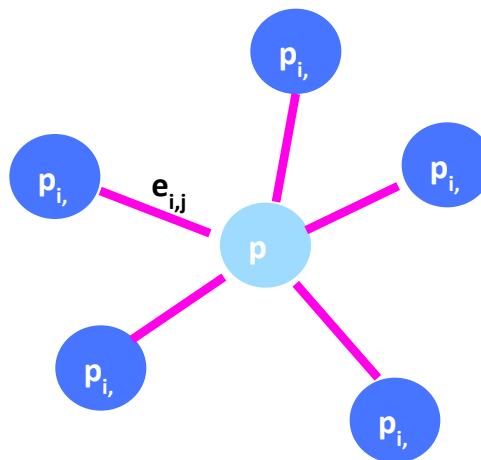**k-Nearest Neighbors**

EdgeConv: Convolution on point clouds

- Find the *k*-nearest neighbors of each point

- Design a permutation invariant **convolution operation**

  - Define an **edge feature** function → **aggregate** edge features w/ a symmetric func.

k-Nearest Neighbors

Convolution operation

ParticleNet:
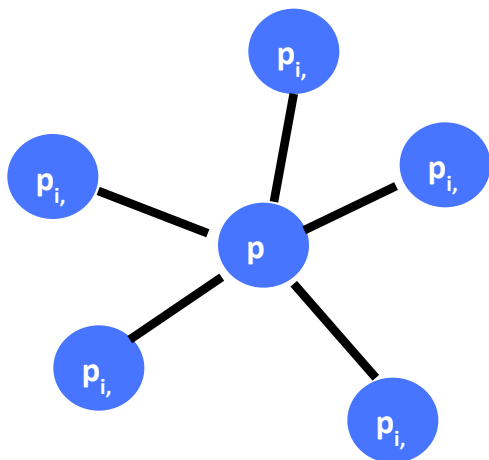U_Θ: MLP [shared across edges]
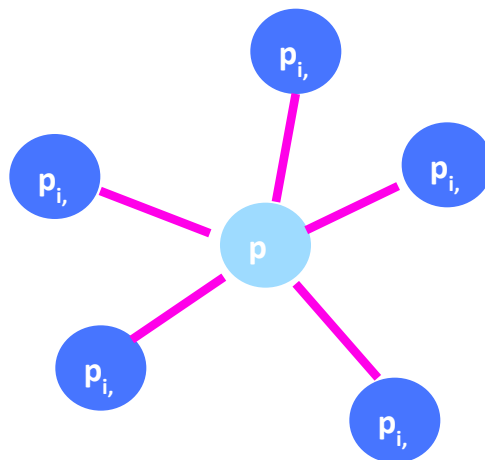□ : average over all *k*-NN

EdgeConv: Convolution on point clouds

- Find the **k-nearest neighbors** of each point

- Design a permutation invariant **convolution operation**

  - Define an **edge feature** function → **aggregate** edge features w/ a symmetric func.

- **Update Graph (ie Dynamic Graph CNN, DGCNN):**
  Using *k*NN in the feature space produced after EdgeConv

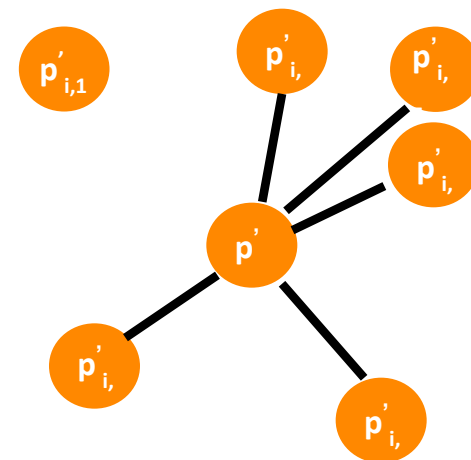 - Can be viewed as a mapping from one particle cloud to another

k-Nearest Neighbors

Convolution operation

Update Graph

ParticleNet:

h_Θ: MLP [shared across edges]

□ : **average over all *k*-NN**

ParticleNet for jet tagging (II)

- Based on EdgeConv and DGCNN
  - but customized for the jet tagging task

EdgeConv block

ParticleNet Architecture
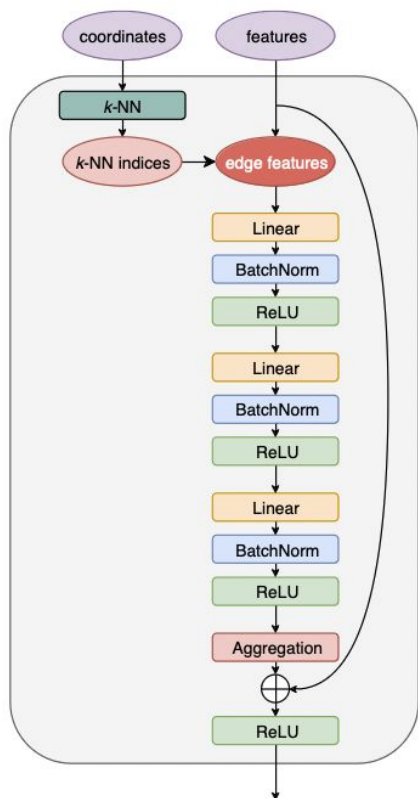
Introduced:
- features beyond spatial coordinates
- residual connections
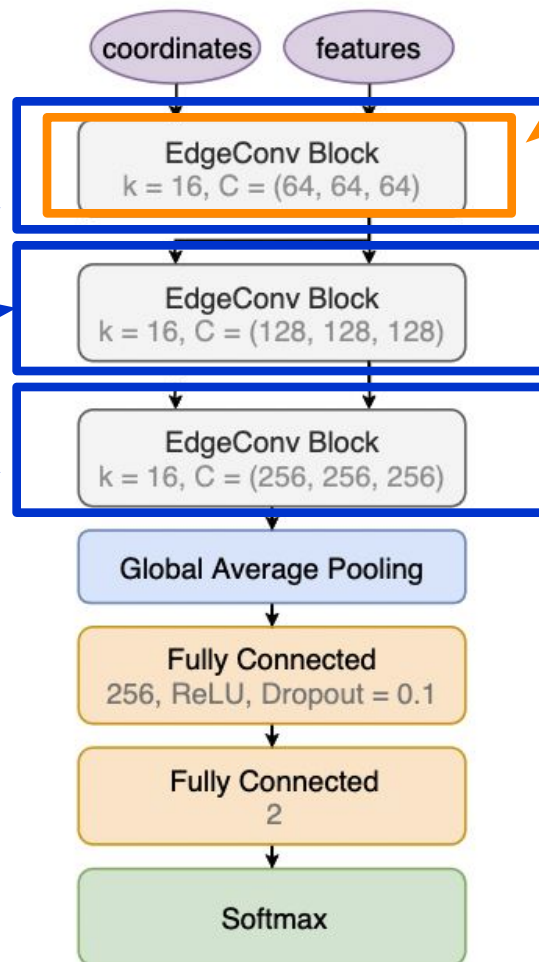
particles distributed in η-φ

From **local** to more **global** structures

ParticleNet in FCCSW

Sample Generation for training

- Generation of the samples in EDM4hep (whole event reconstructions, features for training not explicit)
- FCCAnalyses (wrapper RDataFrame)
  - Per-event → per-jet structure
  - 2 stages. 1: read edm4hep and extract features. 2: produce n-tuples one per class.
  - final dataset: 5/7 classes and 10^6 events per class
  - trained on gpus (A100)

Inference

- Inference in FCCAnalyses:
  - load ONNX training files
  - Extract hard vertex and perform jet clustering
  - Extract jet constituents and compute observables
  - Evaluate NN → output: **one probability per category**

Training the model

EDM4Hep

FCCAnalyses (RDF)

Jet Clustering (JetClusteringHelper)

Jets

Jet Constituent Data (JetFlavourHelper)

Jet Constituents

PyROOT

Event ROOT Tree

Jet ROOT Tree

Weaver (PyTORCH)

ONNX Model
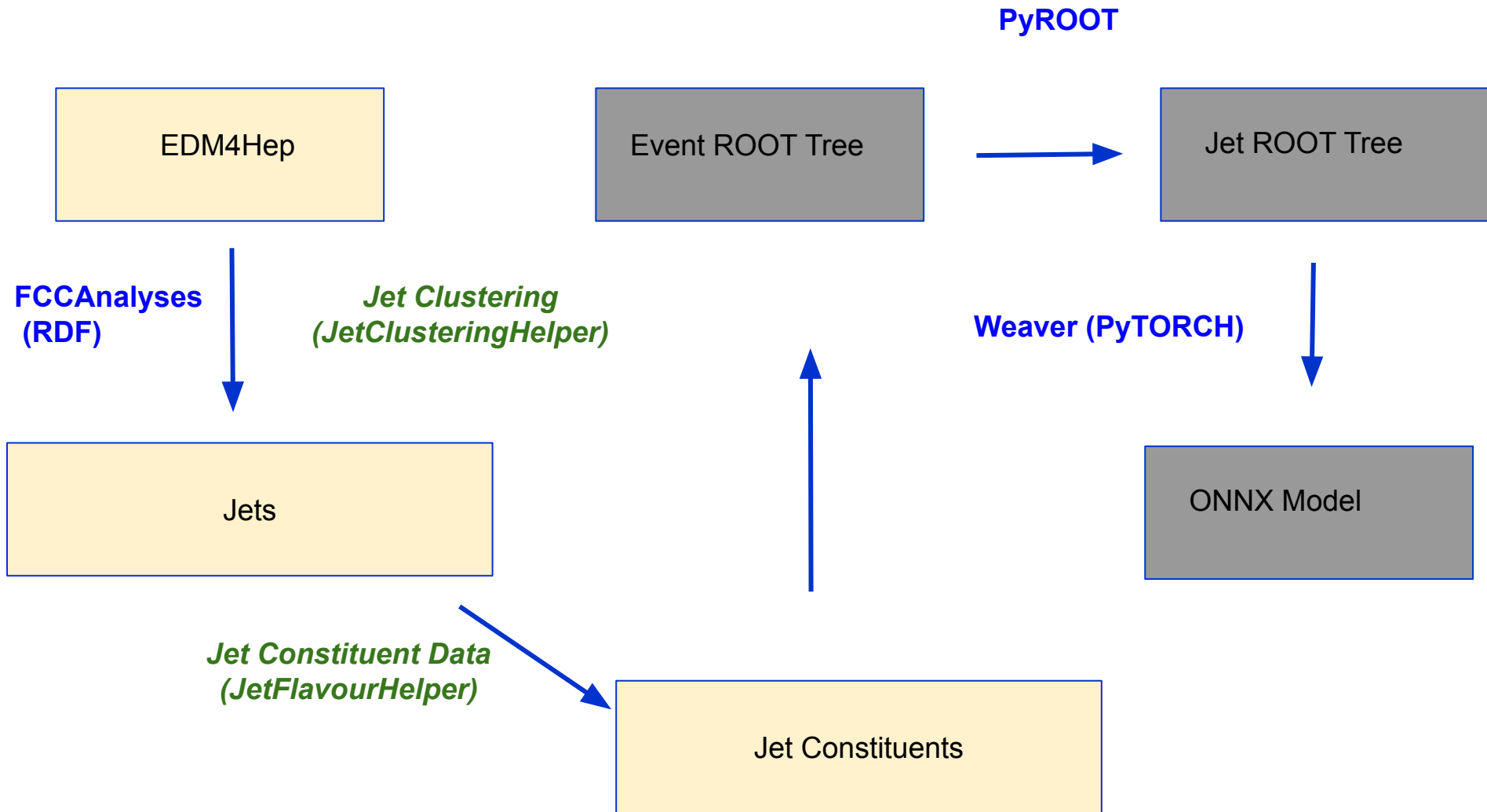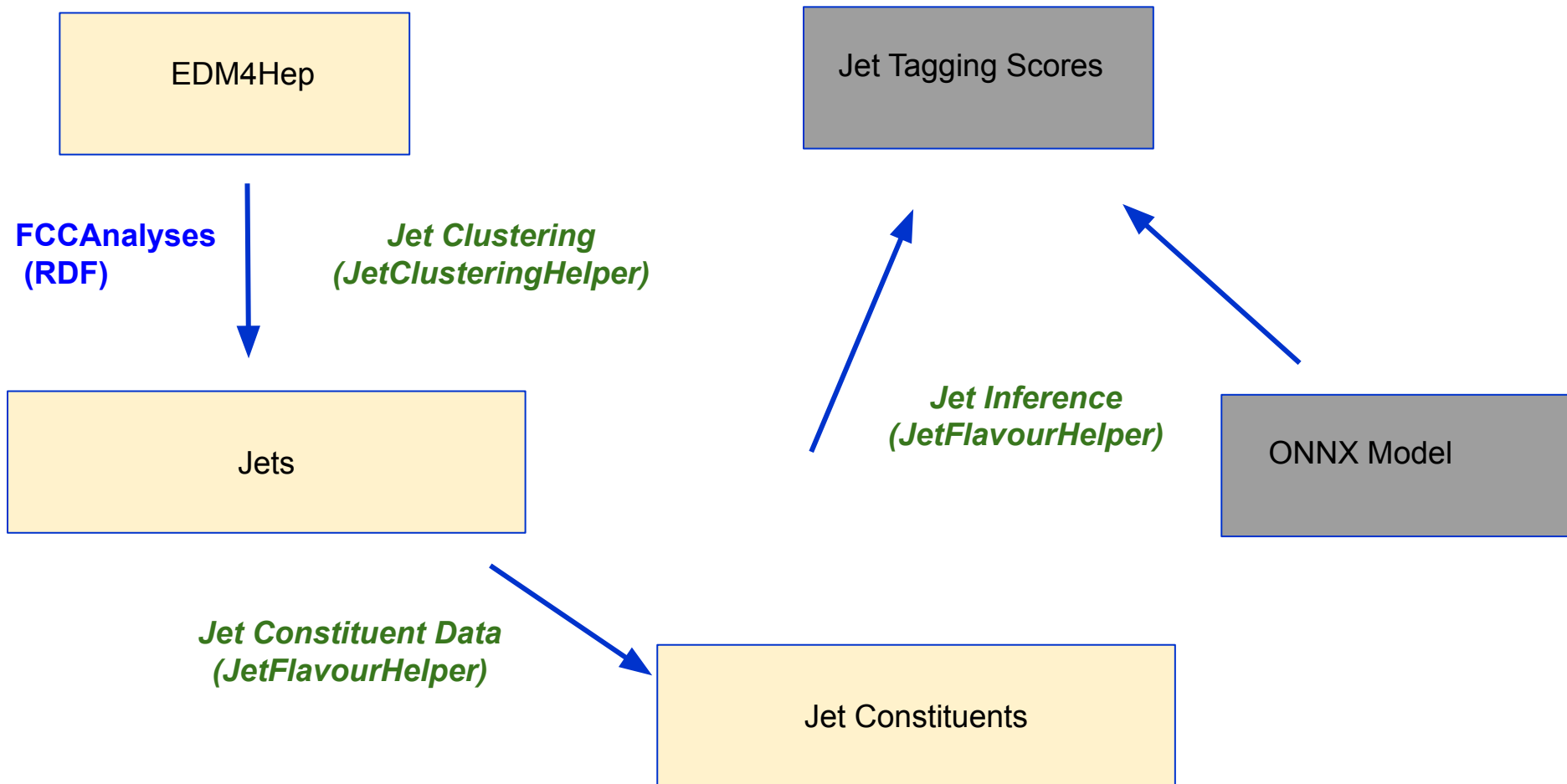
Inference

Inference

```python
## get local file, else download from url
weaver_preproc = get_file_path(url_preproc, local_preproc)
weaver_model = get_file_path(url_model, local_model)

from addons.ONNXRuntime.python.jetFlavourHelper import JetFlavourHelper
from addons.FastJet.python.jetClusteringHelper import ExclusiveJetClusteringHelper
```

```python
## define jet clustering parameters
jetClusteringHelper = ExclusiveJetClusteringHelper(collections["PFParticles"], njets, tag)

## run jet clustering
df = jetClusteringHelper.define(df)

## define jet flavour tagging parameters
```

JetClusteringHelper

```python
jetFlavourHelper = JetFlavourHelper(
    collections,
    jetClusteringHelper.jets,
    jetClusteringHelper.constituents,
    tag,
)

## define observables for tagger
df = jetFlavourHelper.define(df)

## tagger inference
df = jetFlavourHelper.inference(weaver_preproc, weaver_model, df)

return df
```
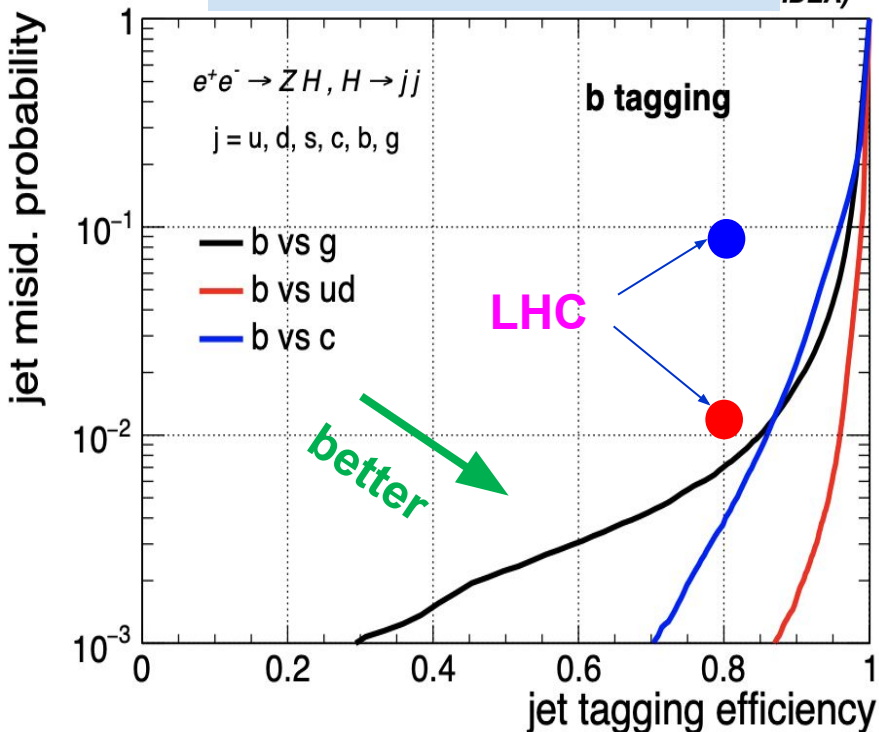
JetFlavourHelper

Loading model parameters

Jet clustering

Obtain input parameters

Run inference

ParticleNet@FCCee: b/c tagging

b-tagging

(IDEA)

$e^+ e^- \rightarrow ZH, H \rightarrow jj$

$j = u, d, s, c, b, g$

b tagging

- b vs g
- b vs ud
- b vs c

LHC

better

c-tagging

(IDEA)

$e^+ e^- \rightarrow ZH, H \rightarrow jj$

$j = u, d, s, c, b, g$

c tagging

- c vs g
- c vs ud
- c vs b

better

WP	Eff (b)	Mistag (g)	Mistag (ud)	Mistag (c)
Loose	90%	2%	0.1%	2%
Medium	80%	0.7%	<0.1%	0.3%
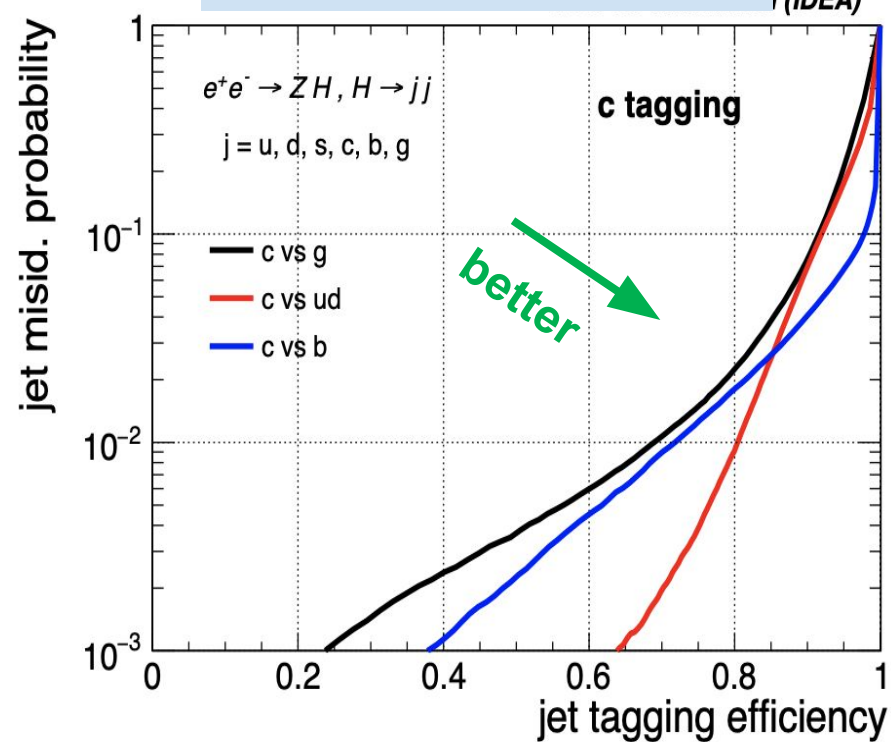
WP	Eff (c)	Mistag (g)	Mistag (ud)	Mistag (b)
Loose	90%	7%	7%	4%
Medium	80%	2%	0.8%	2%

ParticleNet@FCCee: s/g tagging

strange-tagging

FCC-ee Simulation (IDEA)

$e^+e^- \rightarrow ZH, H \rightarrow jj$

$j = u, d, s, c, b, g$

s tagging

- s vs g
- s vs ud
- s vs c
- s vs b

better

jet misid. probability

jet tagging efficiency

gluon -tagging

FCC-ee Simulation (IDEA)

$e^+e^- \rightarrow ZH, H \rightarrow jj$

$j = u, d, s, c, b, g$

g tagging

- g vs ud
- g vs s
- g vs c
- g vs b

better

jet misid. probability

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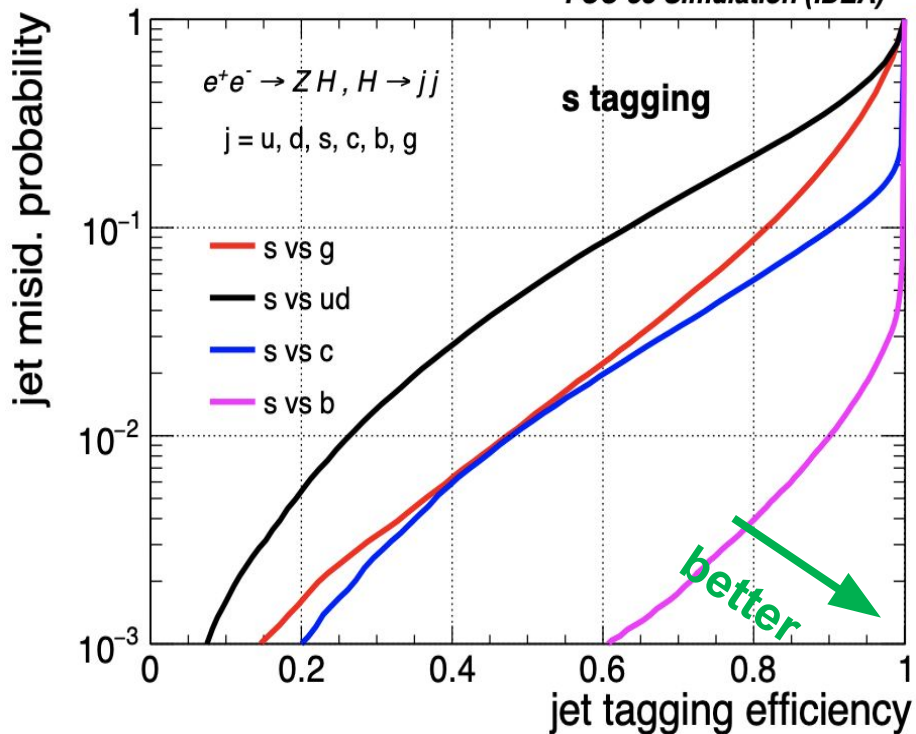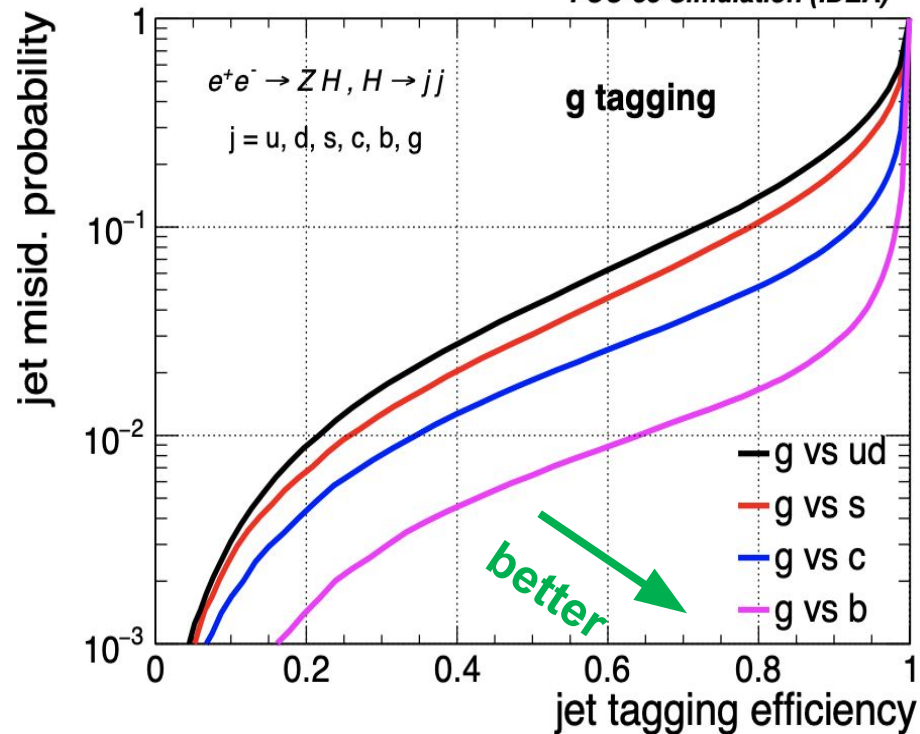
WP	Eff (s)	Mistag (g)	Mistag (ud)	Mistag (c)	Mistag (b)
Loose	90%	20%	40%	10%	1%
Medium	80%	9%	20%	6%	0.4%

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

Impact of detector configurations

FCC-ee Simulation (IDEA)

$e^+e^- \rightarrow ZH, H \rightarrow jj$

$j = u, d, s, c, b, g$

s tagging vs. ud

- no PID
- dN/dx
- dN/dx + t.o.f (σ_t=30 ps)
- dN/dx + t.o.f (σ_t=3 ps)
- ideal PID

better

Ideal from MC

jet misid. probability

jet tagging efficiency

FCC-ee Simulation (IDEA)

$e^+e^- \rightarrow ZH, H \rightarrow jj$

$j = u, d, s, c, b, g$

c tagging

better

- c vs g (3 inner layers)
- c vs g (4 inner layers)
- c vs ud (3 inner layers)
- c vs ud (4 inner layers)
- c vs b (3 inner layers)
- c vs b (4 inner layers)

jet misid. probability

jet tagging efficiency
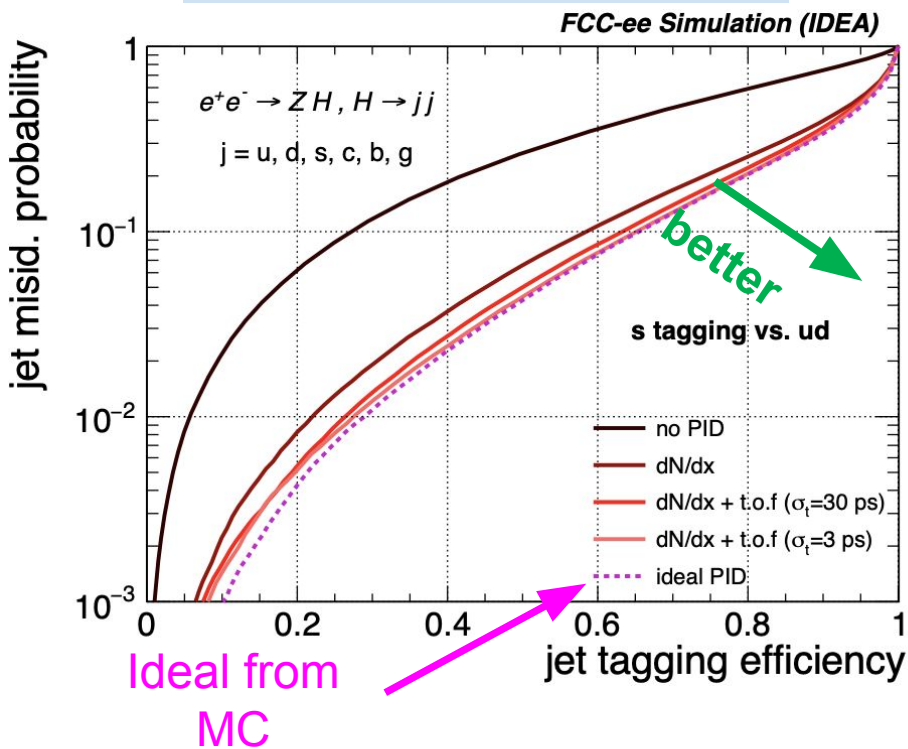
- dN/dx brings most of the gain additional gain w/ TOF (30ps)

  - TOF (3ps): marginal improvement
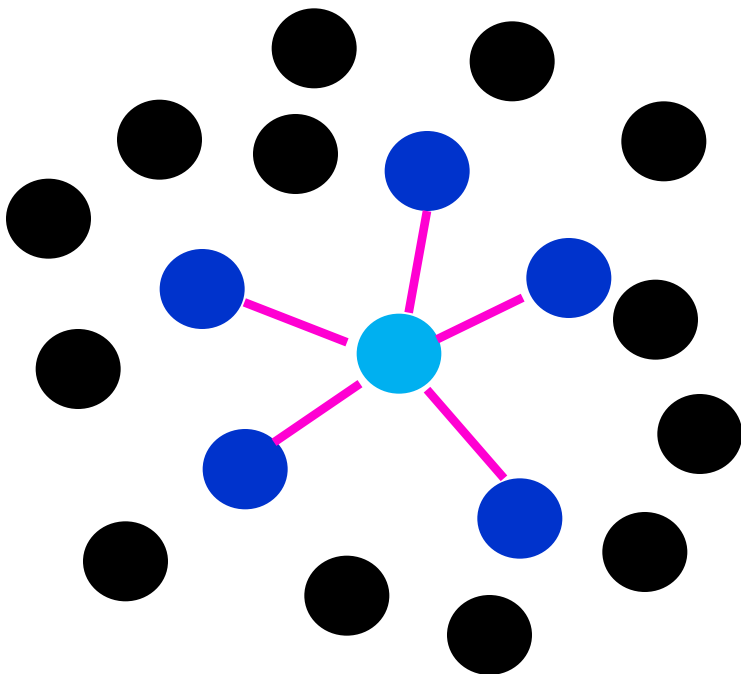 - dN/dX + TOF(30ps) ~ perfect PID
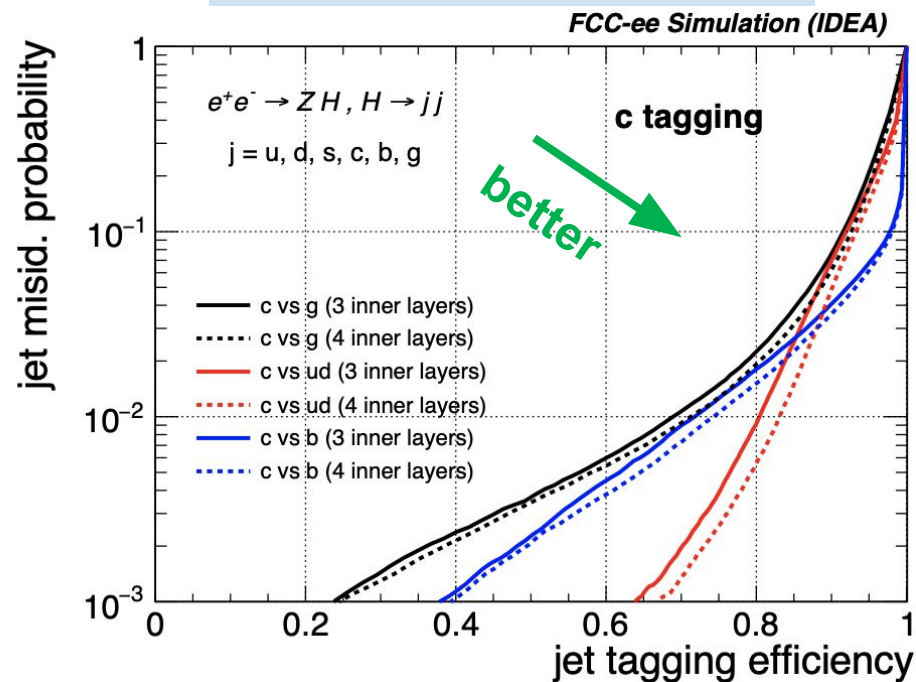
- Additional pixel layer 1 cm from beam pipe vs 1.5 cm:

  - improved BKG rejection in c-tagging
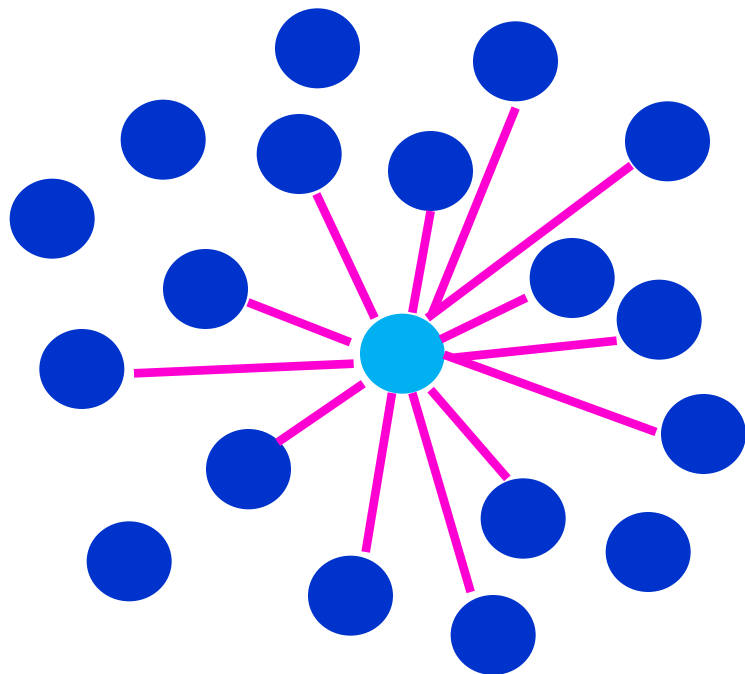 - marginal/no improvement in b-tagging

Pushing the limits further

based on:
H. Qu, C. Li, S. Qian
ICML 2022

ParticleNet-EE

ParticleTransformer

Use the *k*-nearest particles
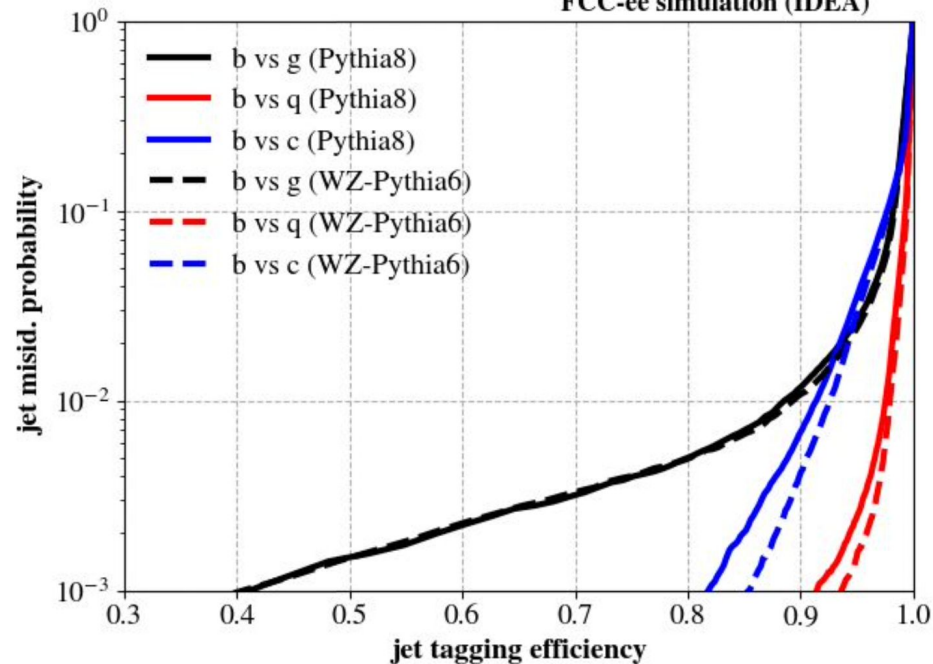[*k*=8 for ParticleNet-EE]

- **Fully connected graph**
- **Include per-particle-pair properties more directly**

Robustness

- ParticleNet-ee trained using *Pythia 8* samples
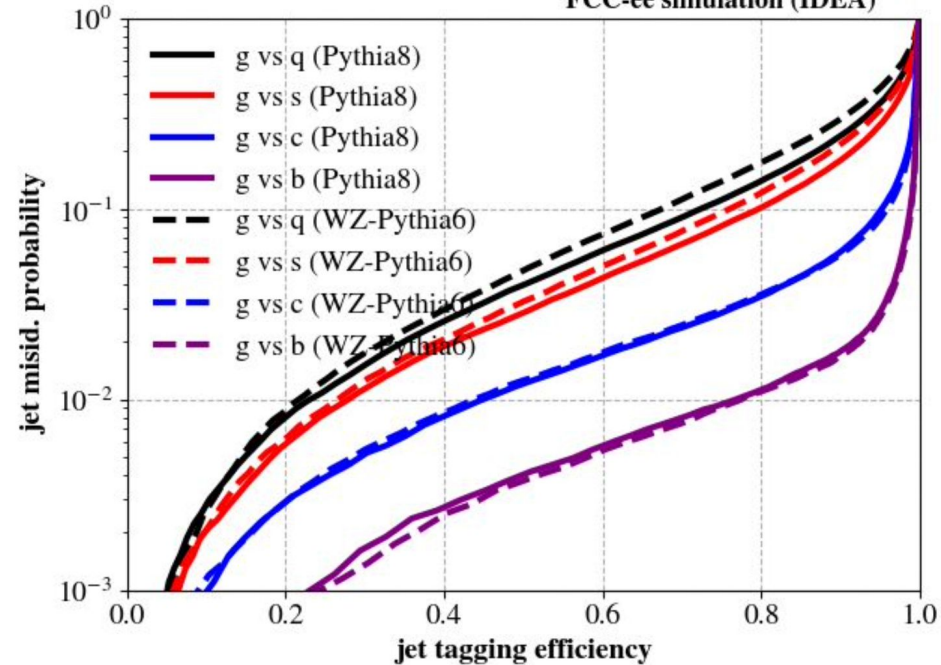 - tested on *Pythia 8* [solid lines]
  - tested on *WZ-Pythia6* [dashed lines]
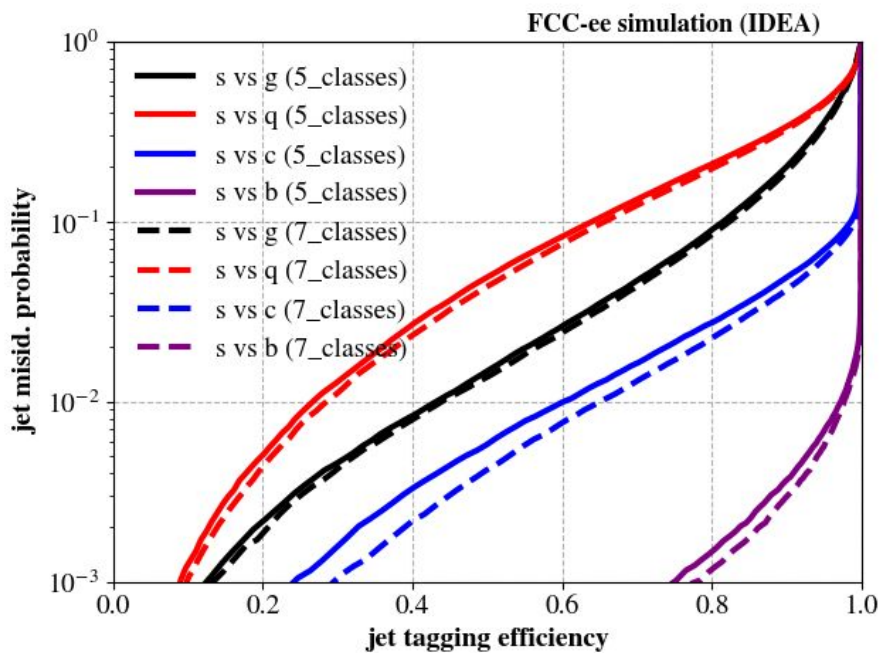
b-tagging

gluon -tagging

- Modest dependence on choice of generator
- More parton showers coming up (Herwig, Sherpa…)

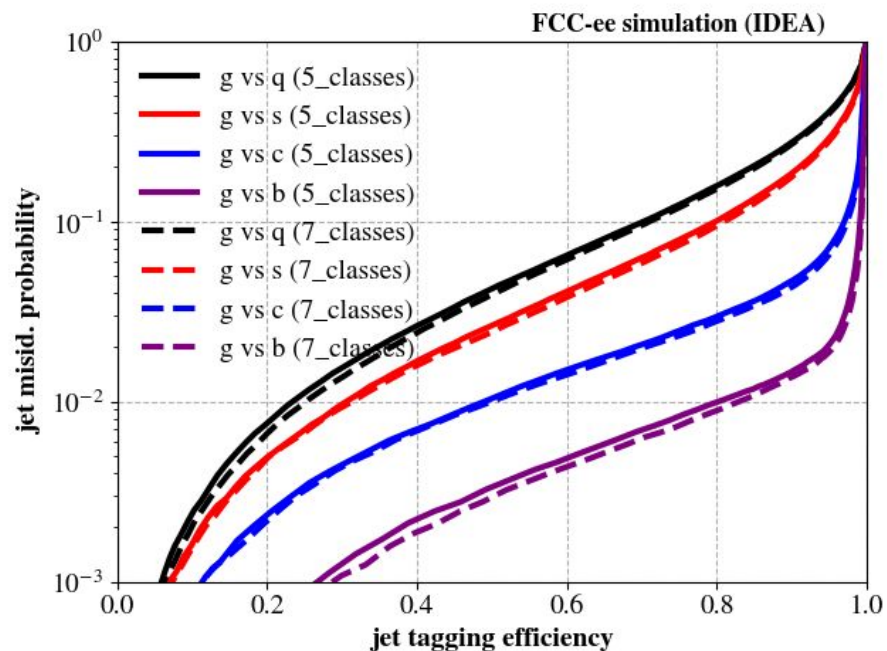# Extension

- ParticleNet-ee trained 7 classes
    - 5 classes, B, C, S, G, Q [solid lines]
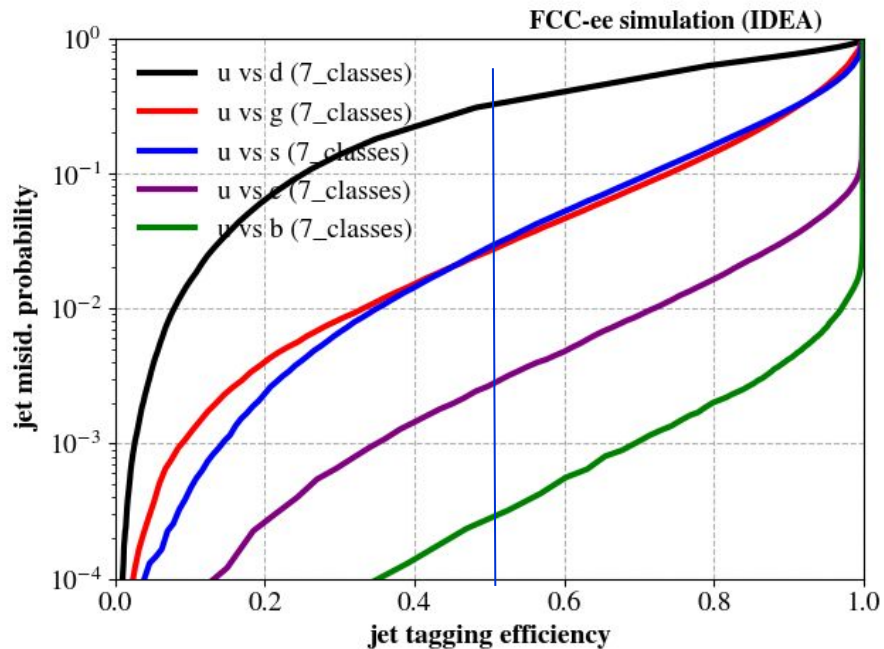   - 7 classes, B, C, S, G, U, D, TAU [dashed lines]
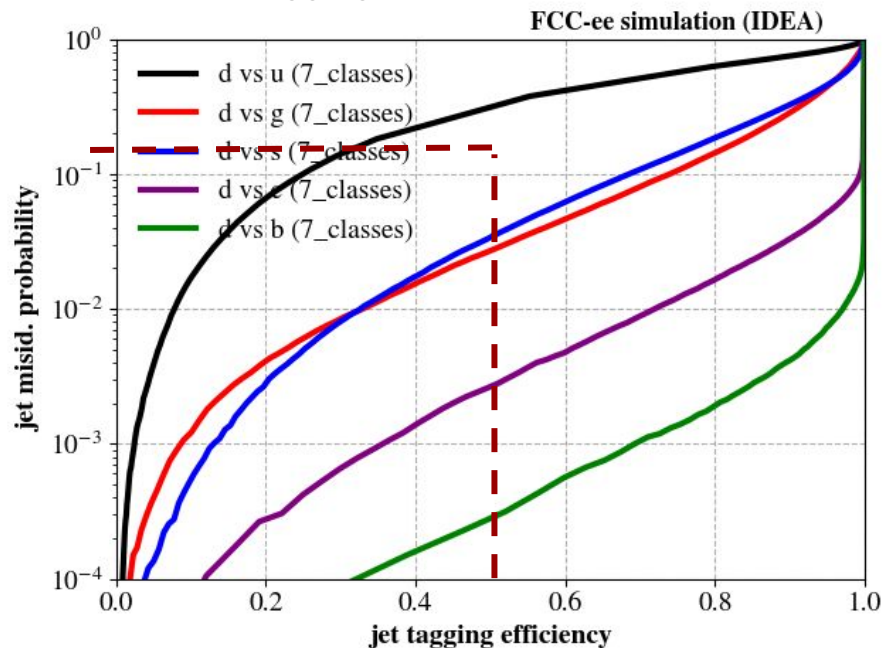
s-tagging

gluon -tagging

Tagger update(up and down)

Up -tagging

Down-tagging

- ● Up vs Down discrimination seems possible thanks to jet charge
- ● 30% bkg eff at 50% signal (better than random coin toss)

Summary

- Powerful jet flavour identification important for the e^+e^- physics program

- Sophisticated jet tagging algorithms developed for e^+e^- experiments
  - Striking improvement in tagging performance compared to previous tools
    - allows us to explore more of the detector and event reconstruction potential
  - Integrated in FCCSW [data preparation, training, validation, inference, analysis] and used in FCCee physics analyses

- Developments in ML will give insights (Rate distortion method, layer wise relevance propagation method (LRP))

- Still room for improvement / other ideas to try:
  - secondary tasks, secondary vertexing regression
  - New higher order graph architectures