Jet flavor identification for FCCee

Franco Bedeschi, LG, Michele Selvaggi
[EPJ C 82 646 (2022) link]

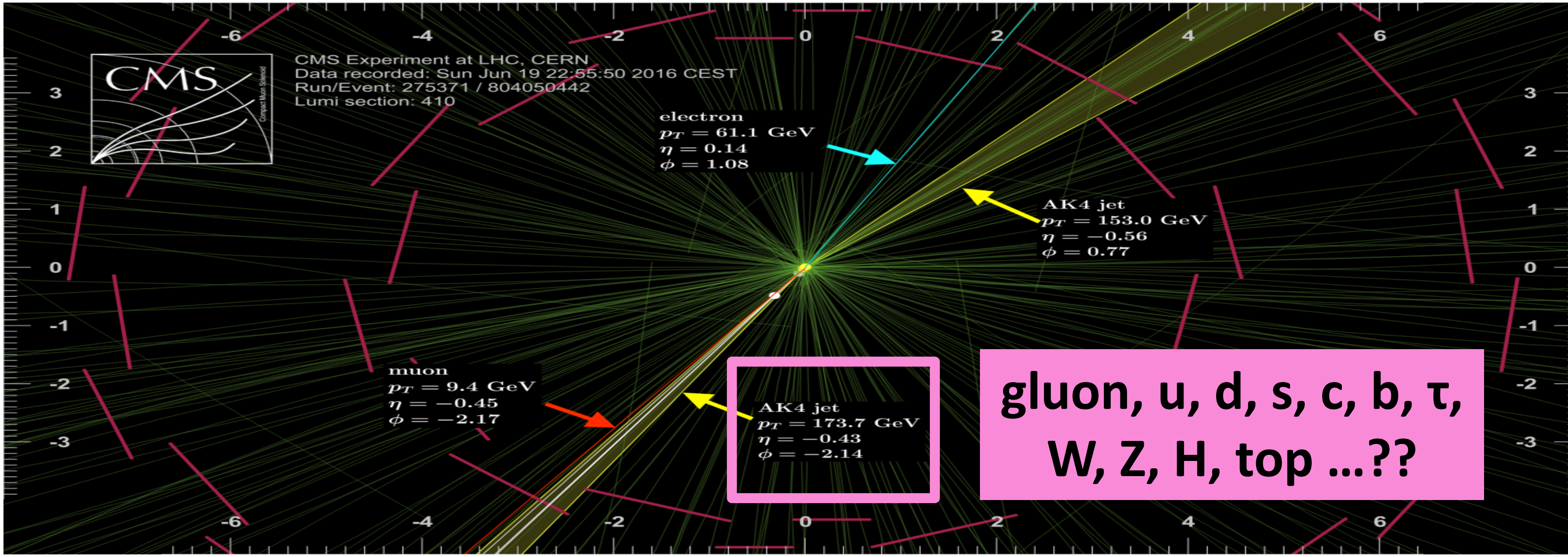
New members: Andrea Del Vecchio, Laurent Forthomme

FCC-ee QCD Physics: Topical Meeting on Jet flavor and Tagging
Dec 13, 2022

Outline

- Review a decade in jet identification @ LHC
- Jet flavour tagging developments for FCCee
- Technical details/ How(Where) to start

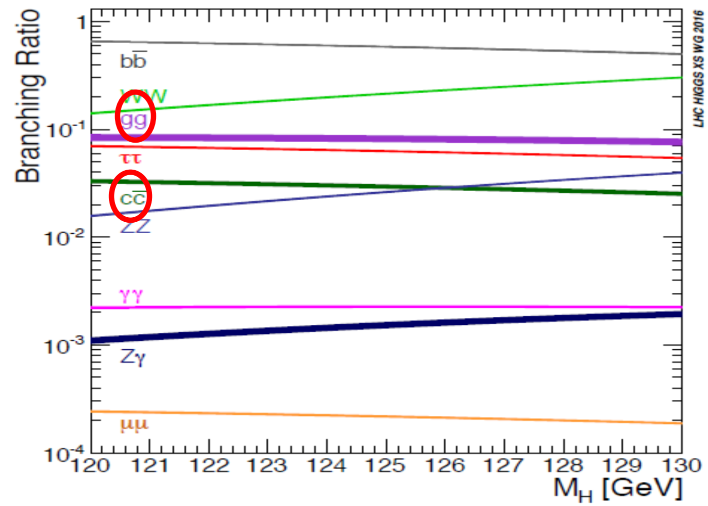
Jet flavor identification ("tagging")

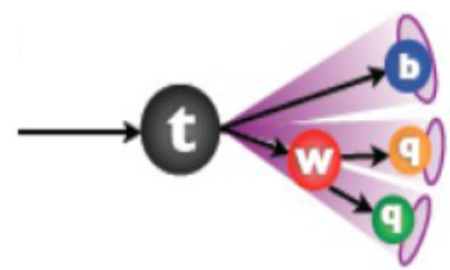


- A topic of high interest in both TH and EXP communities
  - More than 30 years at colliders
    - b-jets at LEP and Tevatron; W, Z, H,.. at the LHC
- Recently: much more powerful algorithms w/ multi-object tagging capabilities
  - opened-up unchartered territory

Physics motivation

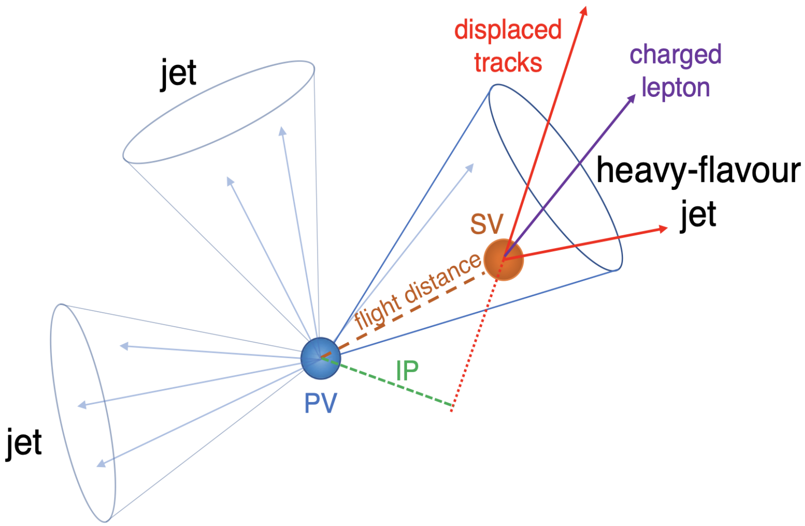
- Flavour tagging essential for the e^+e^- program, e.g.:
  - Higgs Sector:
    - (HL-)LHC can access 3rd gen. couplings and a few of 2nd generation
    - Future e^+e^-: Measure Higgs particle properties and interactions in challenging decay modes
      - E.g. cc, 1st gen quarks/fermions, gg [?]
  - Top quark physics [if E_{CM} sufficient]
    - Precise determination of top properties [mass, width, Yukawa]
  - QCD Physics
    - strong couplin (aS), event shapes ..





Basics for jet flavor identification

bottom/charm-tagging

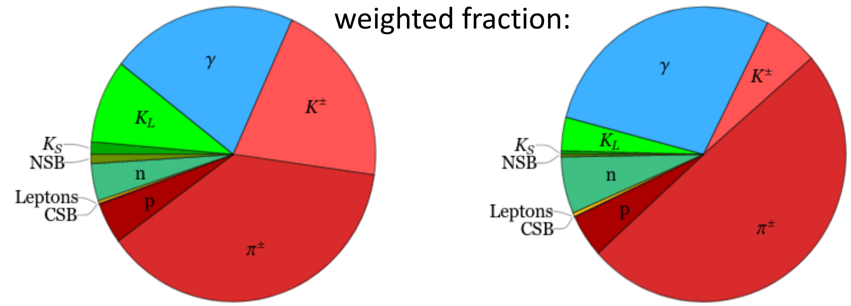


- Large lifetime
- Displaced vertices/tracks
- Large track multiplicity
- non-isolated e/μ

strange-tagging

[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
  - Neutral Kaons:
    - $K_S \to \pi\pi$, K_L

Detector constraints:

Pixel/tracking detectors

- Little material, spatial resolution, precise track alignment

PID detectors:

- timing capabilities, energy loss (gas/silicon)

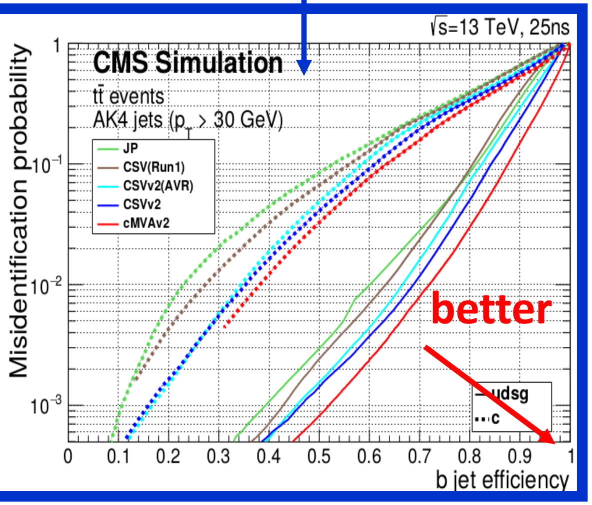
Review of a decade [CMS]

- Enormous progress over the last few years:

e.g., b-tagging

The early days:
Human – inspired
high-level variables
+
Cut-based selection

Run 1 **Run 2** **Run 3**



[Disclaimer: Focus on CMS results; similar methods developed by the other LHC experiments]

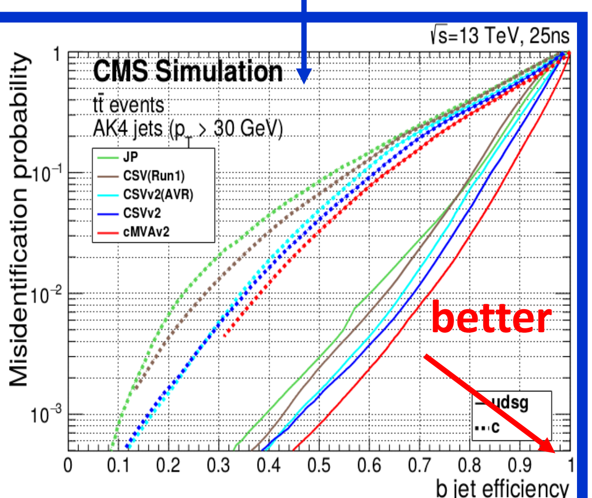
Review of a decade [CMS]

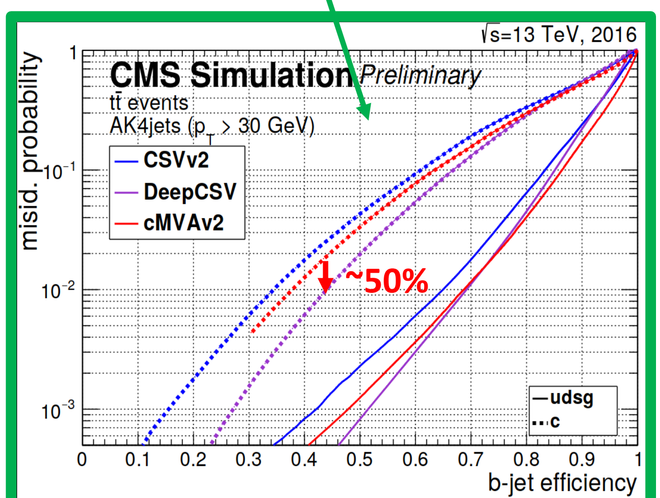
- Enormous progress over the last few years:

e.g., b-tagging

The early days:
Human – inspired
high-level variables
+
Cut-based selection

Early Run 2:
Human – inspired
high-level variables
+
Simple ML





[Disclaimer: Focus on CMS results; similar methods developed by the other LHC experiments]

Review of a decade [CMS]

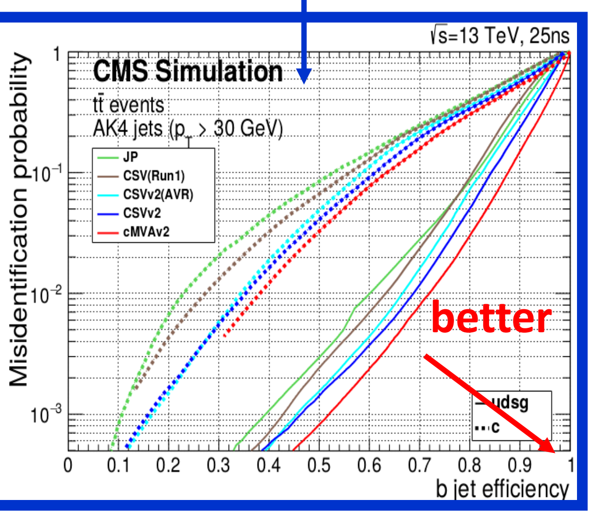
- Enormous progress over the last few years:

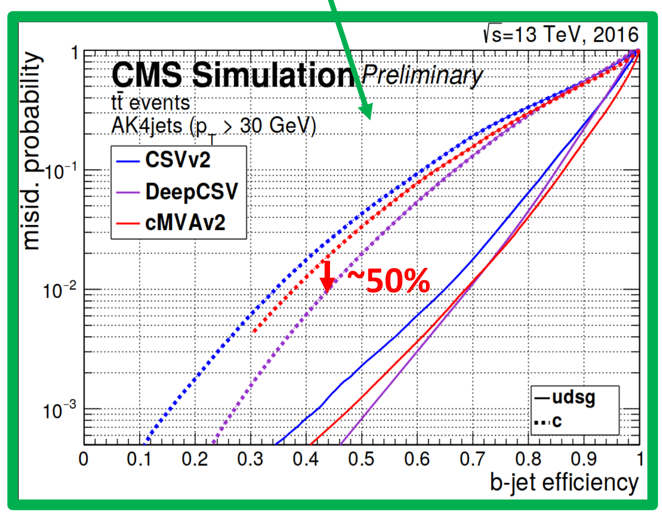
e.g., b-tagging

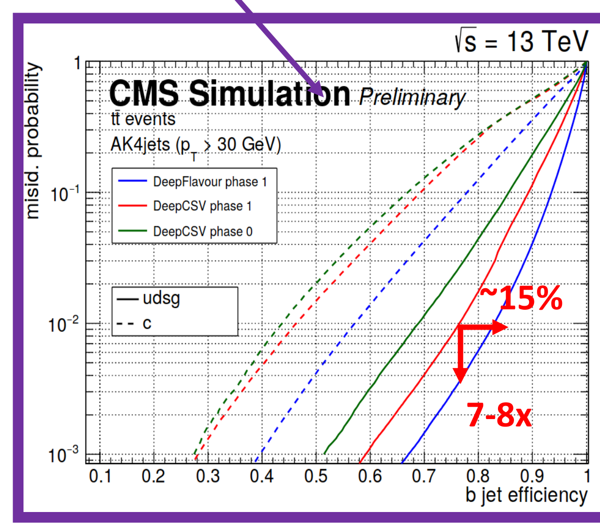
The early days:
Human – inspired
high-level variables
+
Cut-based selection

Early Run 2:
Human – inspired
high-level variables
+
Simple ML

The "Game changer":
Inputs: low-level info
[as particle sequences]
+
Advanced ML (CNN, RNN)







[Disclaimer: Focus on CMS results; similar methods developed by the other LHC experiments]

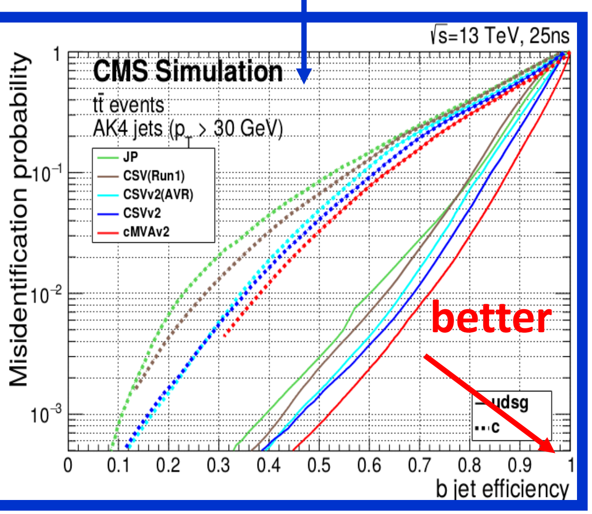
Review of a decade [CMS]

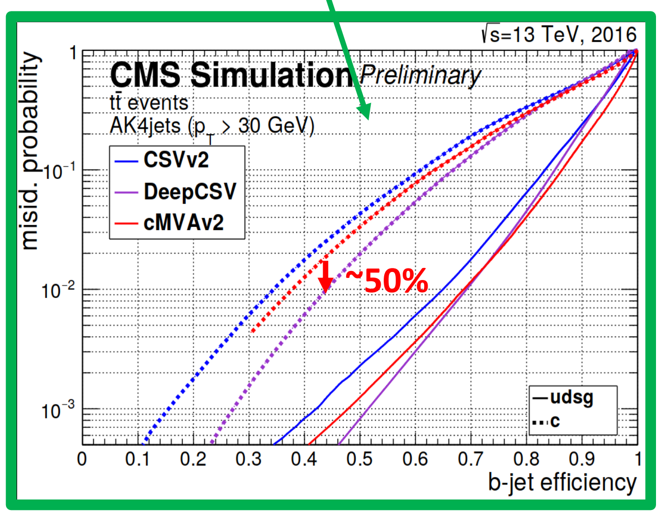
- Enormous progress over the last few years:

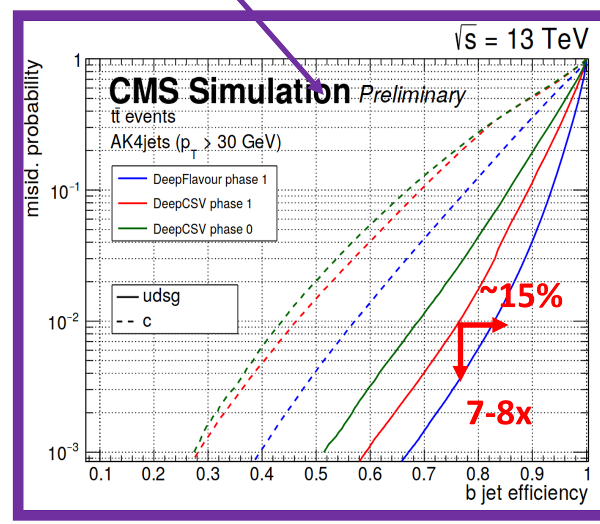
e.g., b-tagging

The early days:	**Early Run 2:**	**The "Game changer":**	**Pushing limits further:**
Human – inspired high-level variables + Cut-based selection	Human – inspired high-level variables + Simple ML	Inputs: low-level info [as particle sequences] + Advanced ML (CNN, RNN)	Inputs: low-level info [as unordered sets] + Graph Neural Nets

Run 1 **Run 2** **Run 3**







[Disclaimer: Focus on CMS results; similar methods developed by the other LHC experiments]

Review of a decade [CMS]

- Enormous progress over the last few years:

The early days:
Human – inspired high-level variables
+
Cut-based selection

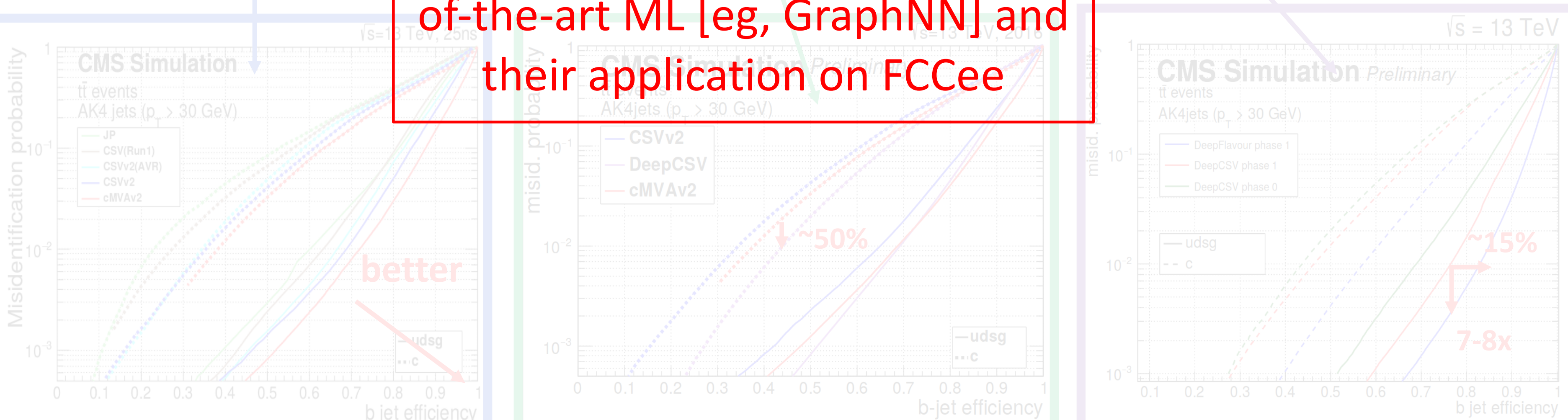
Early Run 2:
Human – inspired high-level variables
+
Simple ML

The "Game changer":
Inputs: low-level info
+
Advanced ML (CNN, RNN)

Pushing limits further:
Inputs: low-level info
[as unordered sets]
+
Graph Neural Nets

Run 1 | Run 2 | **Run 3**

Today: only discuss the very latest developments using state-of-the-art ML [eg, GraphNN] and their application on FCCee



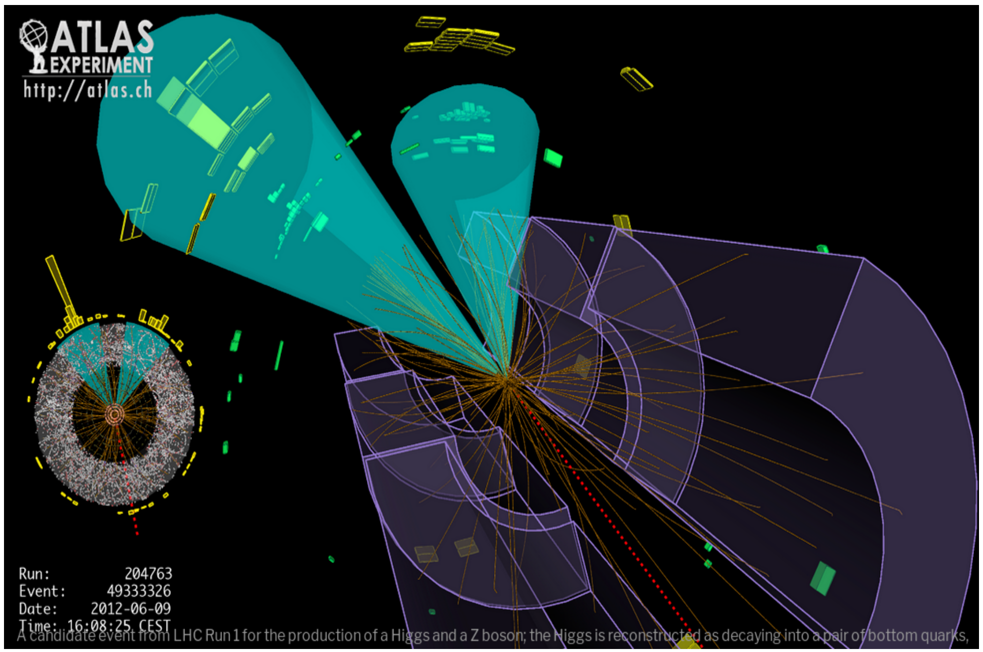
Part 2: Jet Flavor tagging @ FCCee

- **Scope:** Build a general framework for developing flavor tagging algorithms for future colliders [eg., e^+e^-]
  - Fast detector simulation
    - Understand detector requirements/ optimize design
      - eg., vertexing and PID capabilities of the FCCee detectors
  - Develop a versatile flavor tagger
    - Identify with high purity gluons and ud, strange, charm, bottom quarks
    - Baseline: ParticleNet jet tagging algorithm
      - modifications to meet e+e- motivations/challenges/goals
    - Results shown for FCCee & IDEA

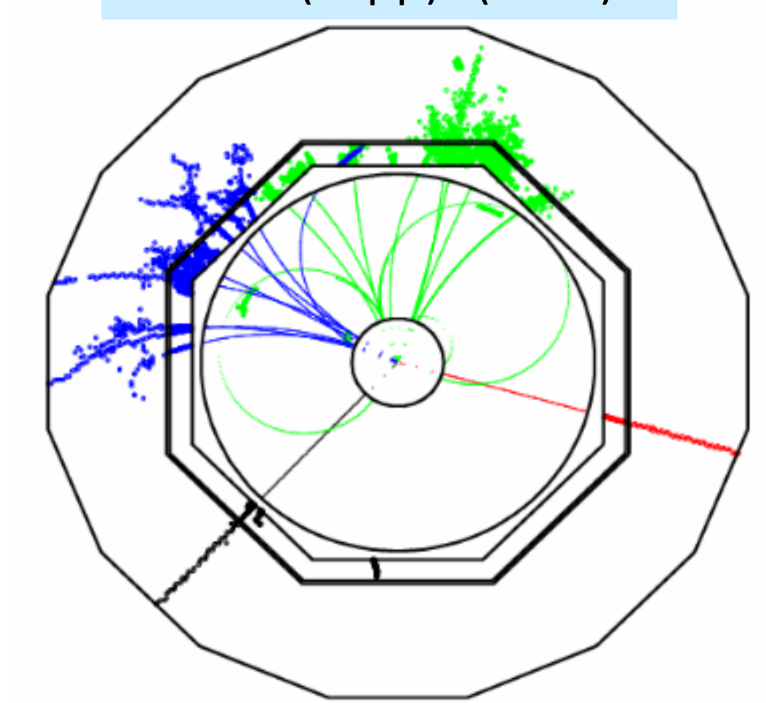
Jet tagging: from hh → e^+e^- colliders

- e^+e^- colliders provide a very clean environment
  - Lower occupancy , no pileup

LHC: Z(→νν)H(→bb)



e^+e^-: Z(→μμ)H(→bb)



Jet tagging: from hh → e^+e^- colliders

- e^+e^- colliders provide a very clean environment
  - Lower occupancy , no pileup
- Powerful detectors:
  - Pixel/tracking detectors tailored for b/c tagging
    - Higher granularity wrt to LHC detectors
      - ATLAS/CMS pixel size: O(~100x100 μm^2)
    - Less tracking material
      - ~0.4% X_0/layer CMS/ATLAS Pixel, ~0.15-0.2% X_0/layer in e^+e^- detectors
      - better impact parameter resolution/ less multiple scattering
      - CMS/ATLAS Pixel resolution: O(10) µm; ~2-5 µm in e^+e^-
  - PID capabilities
    - dE/dx (Si tracker), dN/dx (Drift)
    - Time-of-flight [timing layer]

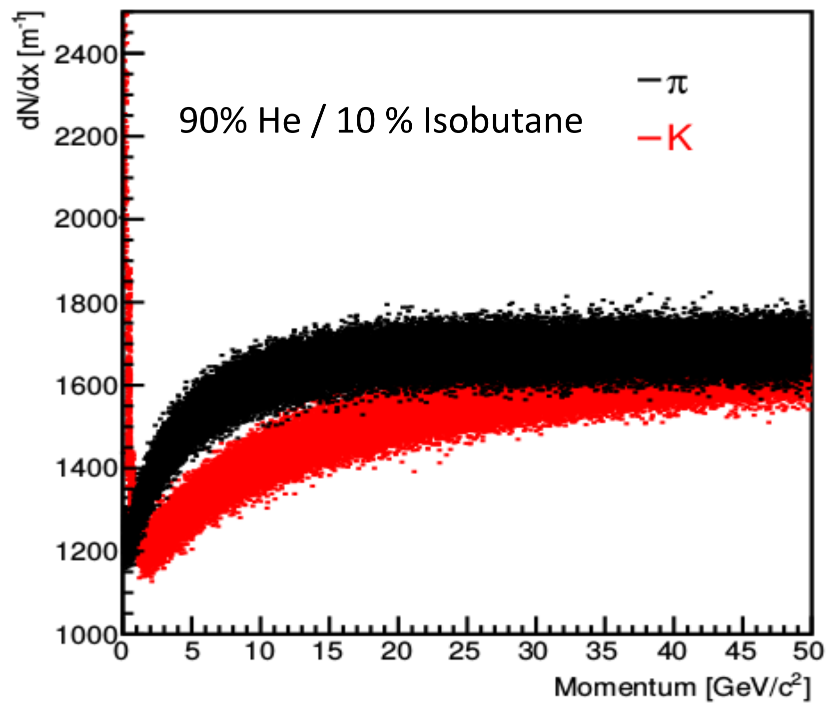
→ e^+e^-: Natural place to explore potential of jet tagging algorithms using advanced ML
→ A step further: Consider reconstructing the full event in e^+e^-

Particle ID: Cluster counting (dN/dx)

- Count number of **primary ionization** clusters along track path
- Avoids large Landau flukes
- Requires high granularity
- module added in Delphes

IDEA detector:

```
###################
# Cluster Counting
###################

module ClusterCounting ClusterCounting {

  add InputArray TrackSmearing/tracks
  set OutputArray tracks

  set Bz $B

  ## check that these are consistent with DCHCANI/DCHNANO parameters in TrackCovariance module
  set Rmin $DCHRMIN
  set Rmax $DCHRMAX
  set Zmin $DCHZMIN
  set Zmax $DCHZMAX

  # gas mix option:
  # 0:  Helium 90% - Isobutane 10%
  # 1:  Helium 100%
  # 2:  Argon 50% - Ethane 50%
  # 3:  Argon 100%

  set GasOption 0

}
```

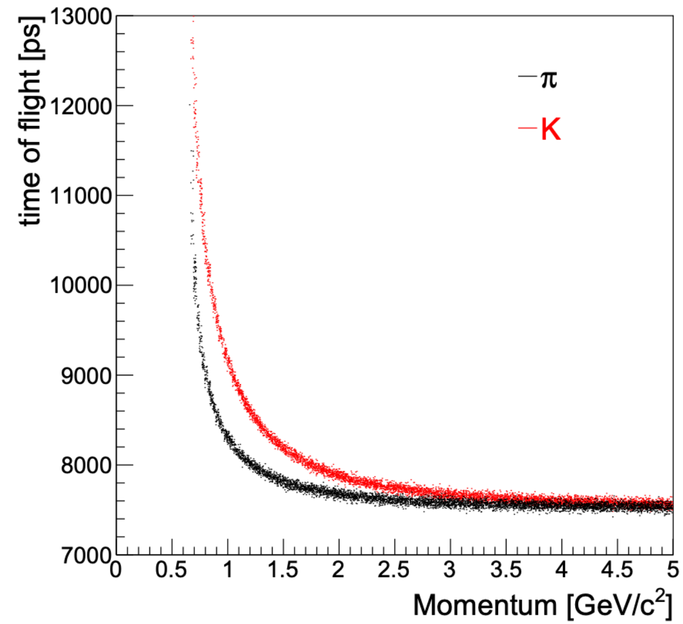


Particle ID: TOF

- Good K/π separation at low-momenta:

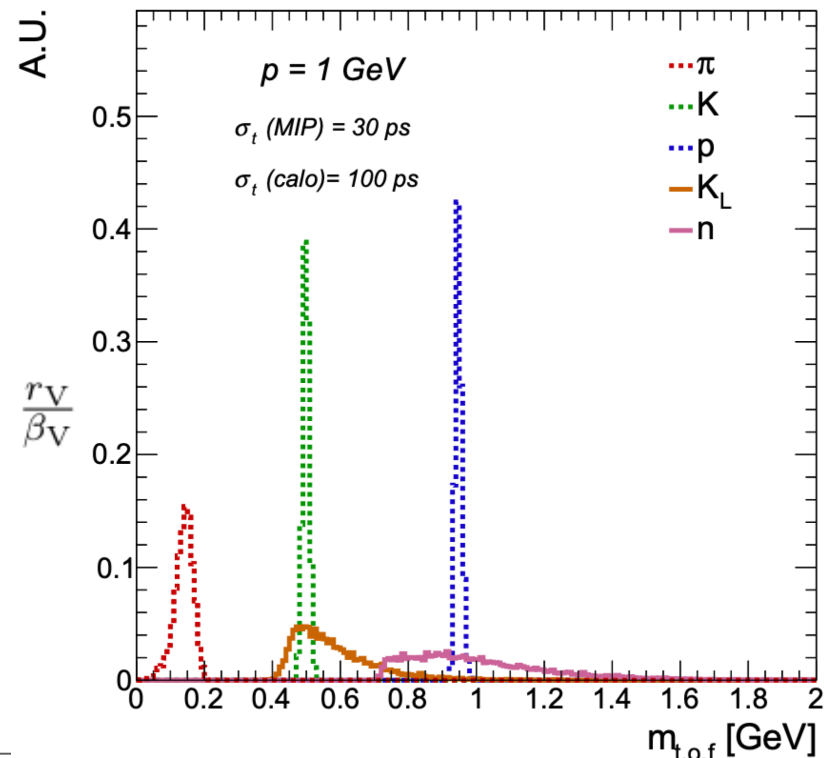
$$t_{\text{flight}} \equiv t_F - t_V = \frac{L}{\beta} = \frac{L\sqrt{p^2+m^2}}{p}$$

- Assumption on vertex time
  [crucial for highly displaced K_s]

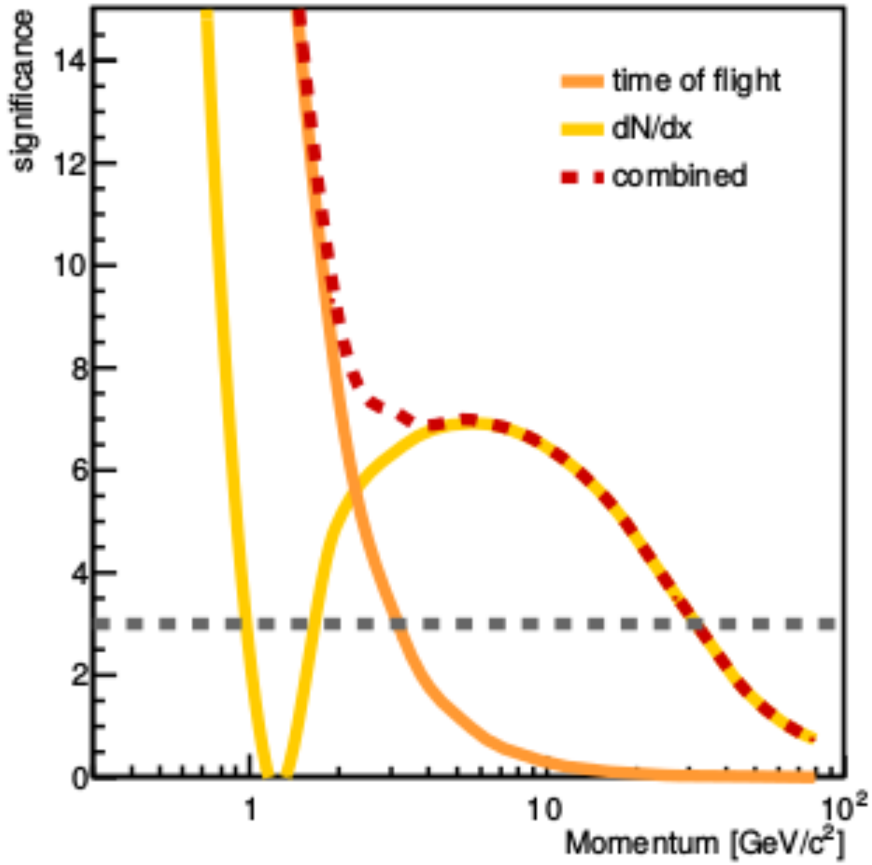



```
########################################
#   Time Of Flight Measurement
########################################

module TimeOfFlight TimeOfFlight {
  set TrackInputArray TimeSmearing/tracks
  set VertexInputArray TruthVertexFinder/vertices

  set OutputArray tracks

  # 0: assume vertex time tV from MC Truth (ideal case)
  # 1: assume vertex time tV = 0
  # 2: calculate vertex time as vertex TOF, assuming tPV=0

  set VertexTimeMode 2
}
```

$$t_V = \frac{r_V}{\beta_V}$$



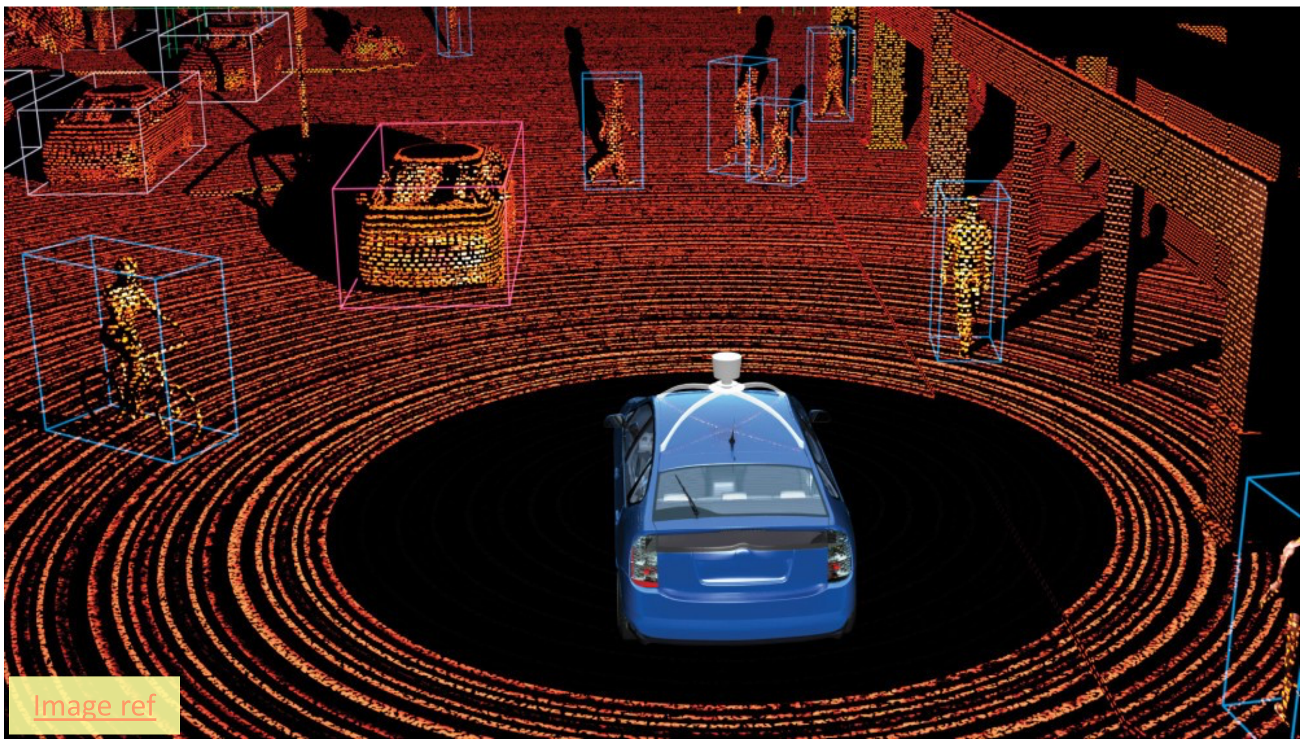
ParticleID: Combined



3σ K/π separation for tracks w/ p<30 GeV

Designing a Graph-based tagger

- **Jet as particle sequence:** striking improvement in jet tagging performance
  - **Important limitation:** Must impose a "human-chosen" ordering [in p_T, displacement, etc..]
  - **However**, a Jet is an intrinsically unordered set of particles with relationships between the particles
- Beyond sequences: **Point clouds**

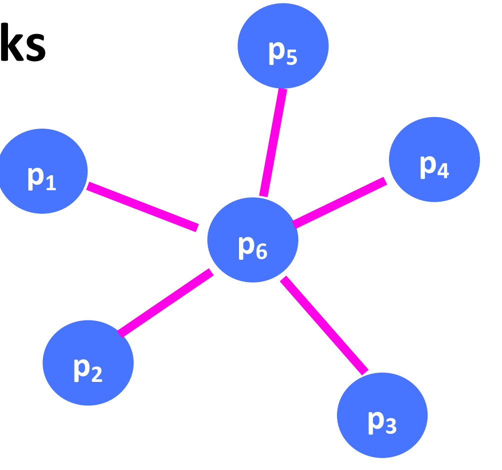


- A very active research area in ML community
- A set of unordered data points in space (x,y,z) with no fixed structure
- Points "close" in space represent physical objects

Designing a Graph-based tagger (II)

- Improve jet representation: "***Particle Sequences***" → "***Particle Clouds***"
  - Treat the jet as an unordered set of particles
  - Rich set of information per particle
    - can be "viewed" as the coordinates of each particle in an abstract space

- Improved Network architecture: **Graph Neural Networks**
  - Particle cloud represented as a graph
    - Each particle: **vertex** of the graph
    - Connections between particles: the **edges**



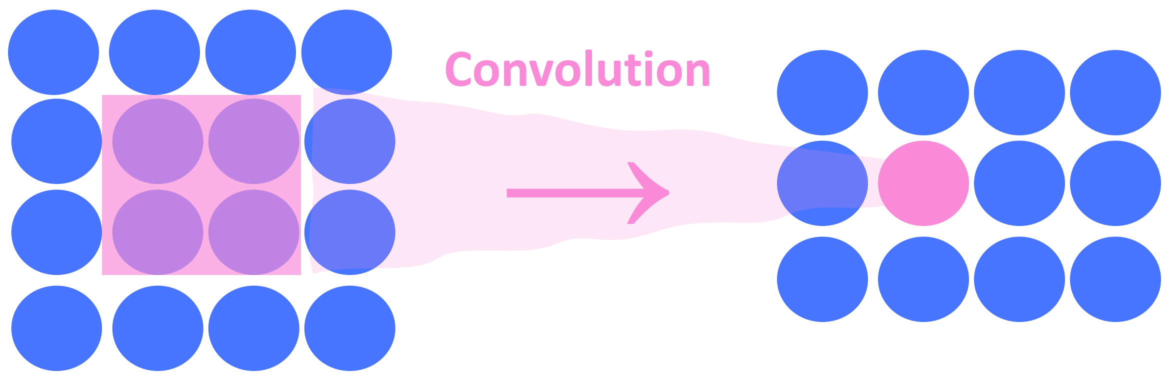
- **Build** the graph:
  - One approach: Fully connected Graph [but computationally very expensive]
  - Another possibility: apply some criteria
    - e.g., k-Nearest Neighbors (kNN)

Designing a Graph-based tagger (III)

- 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:

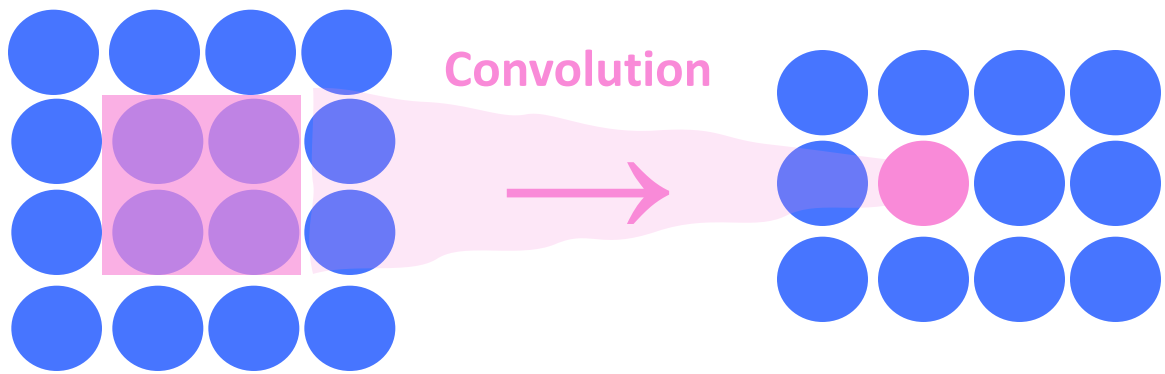


Designing a Graph-based tagger (IV)

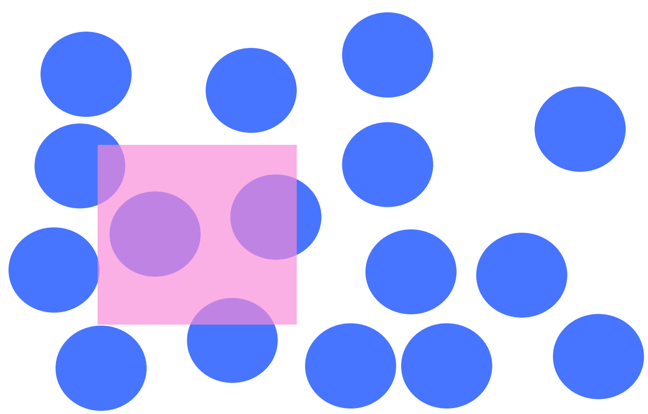
- Last step: **Learn** from the graphs
  - Follow a **hierarchical learning** approach:
    First learn **<u>local</u>** structures and **then more** **<u>global</u>** ones

- Convolution operations proven to be very powerful

Fixed grid:



point/particle cloud:

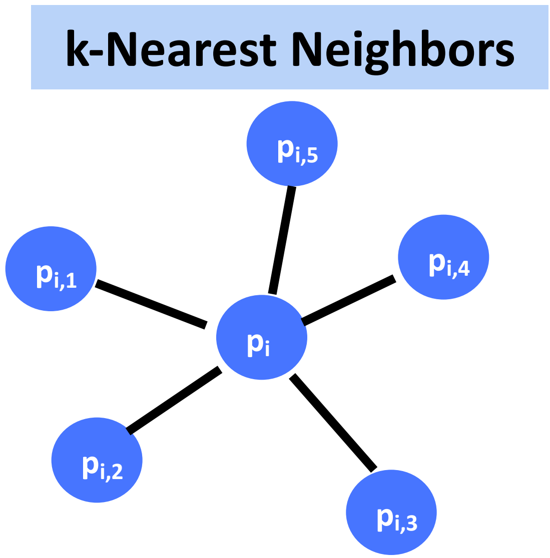


… but not straightforward on point/particle clouds
- Irregular and unordered sets
- Requires a permutation invariant convolutional operation

EdgeConv: Convolution on point clouds

Y. Wang et al.

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

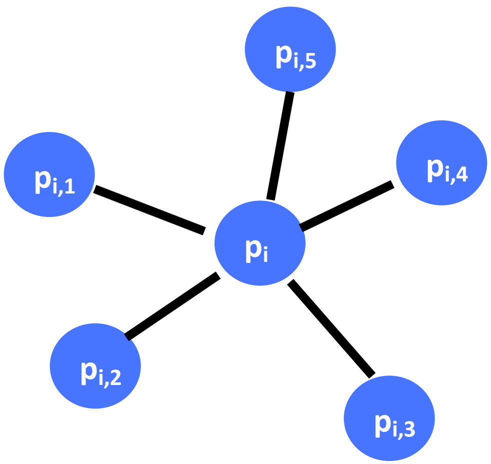


EdgeConv: Convolution on point clouds

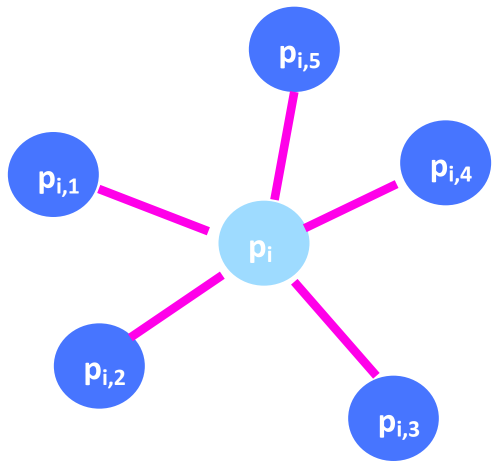
Y. Wang et al.

- 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



- In a nutshell:

$$p_i' = \mathop{\square}_{j=1}^{k} h_\Theta(p_i, p_{ij} - p_i)$$

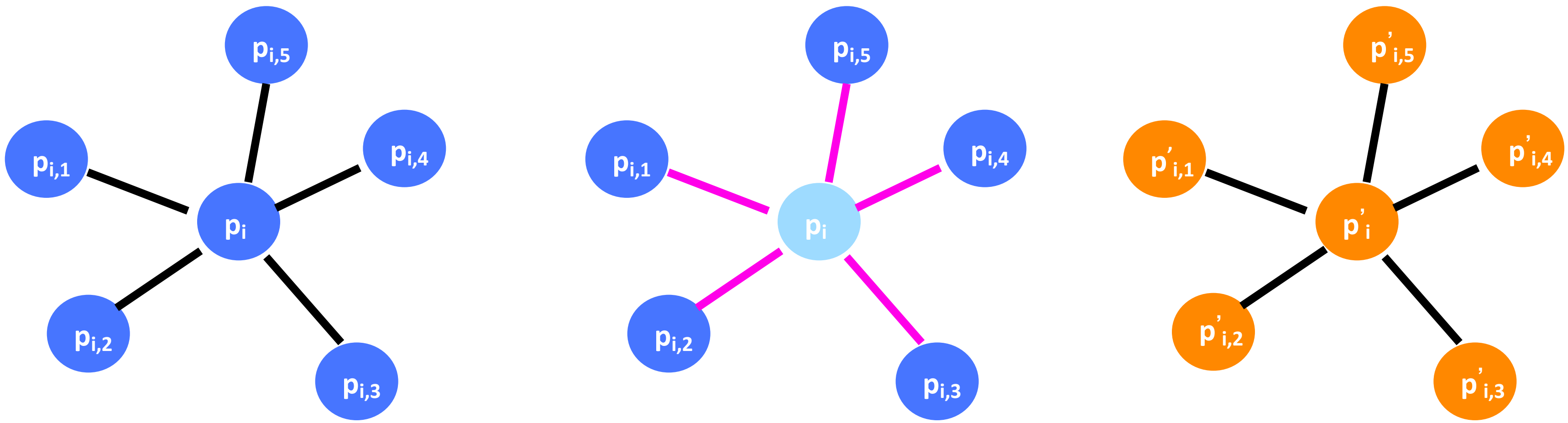
ParticleNet:
h_Θ: MLP [shared across edges]
□ : average over all *k*-NN

EdgeConv: Convolution on point clouds

Y. Wang et al.

- 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**



- In a nutshell:

$$p'_i = \square_{j=1}^{k} h_\Theta(p_i, p_{ij} - p_i)$$

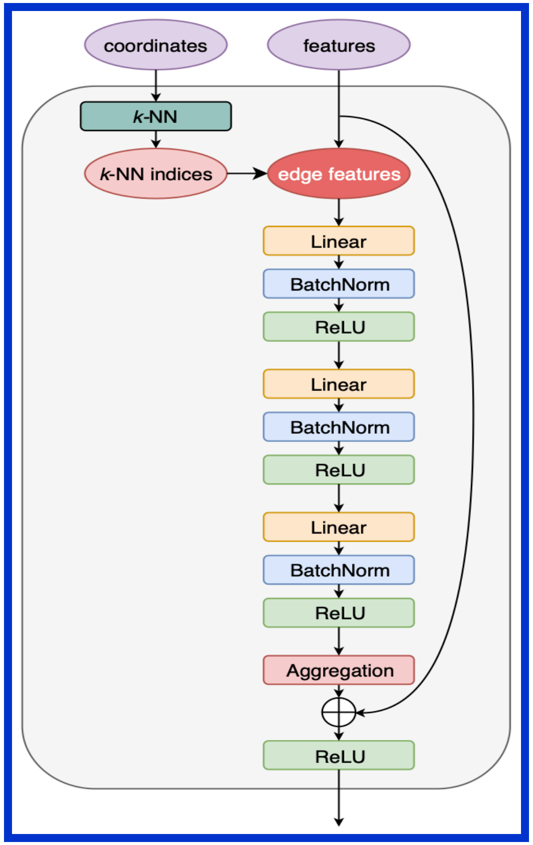
ParticleNet:
h_Θ: MLP [shared across edges]
□ **: average over all *k*-NN**

ParticleNet for jet tagging

H. Qu and LG
PRD 101 056019 (2020)

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

EdgeConv block



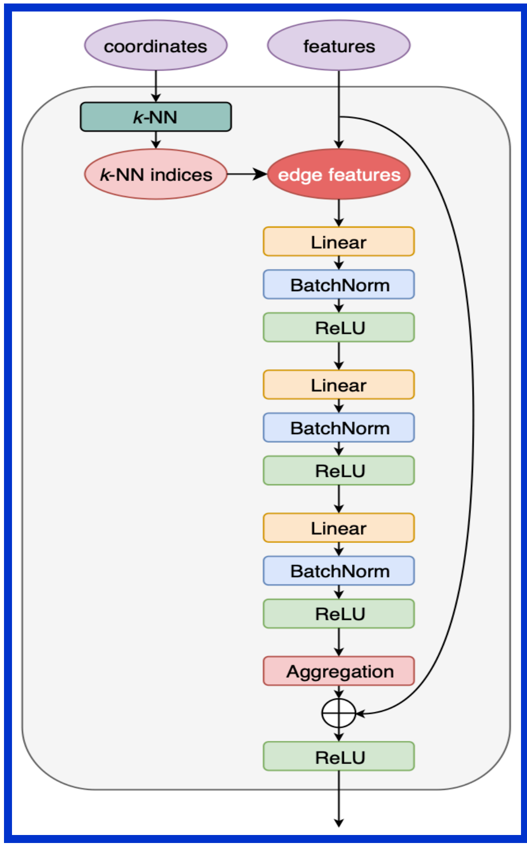
Introduced:
- features beyond spatial coordinates
- residual connections
- MLP conf.

ParticleNet for jet tagging (II)

H. Qu and LG
PRD 101 056019 (2020)

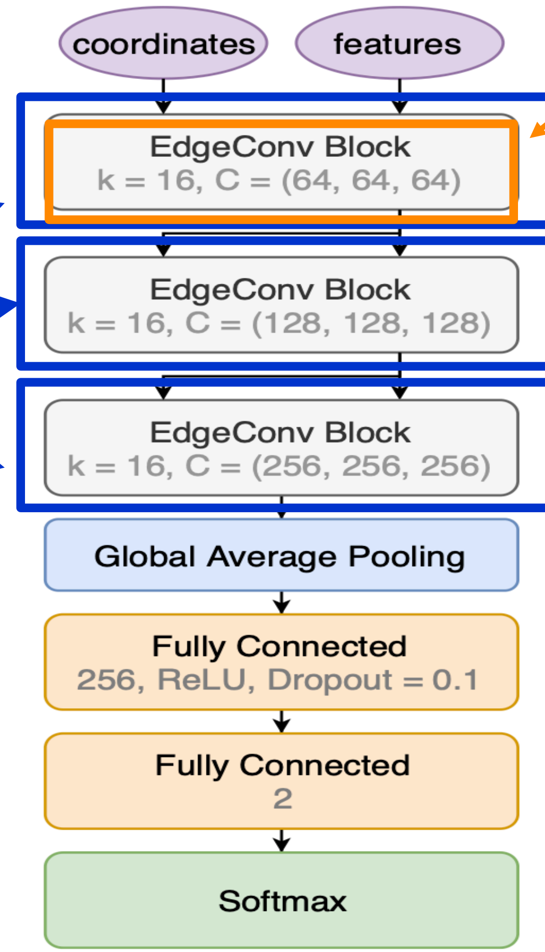
- Based on EdgeConv and DGCNN
  - but customized for the jet tagging task

EdgeConv block



Introduced:
- features beyond spatial coordinates
- residual connections
- MLP conf.

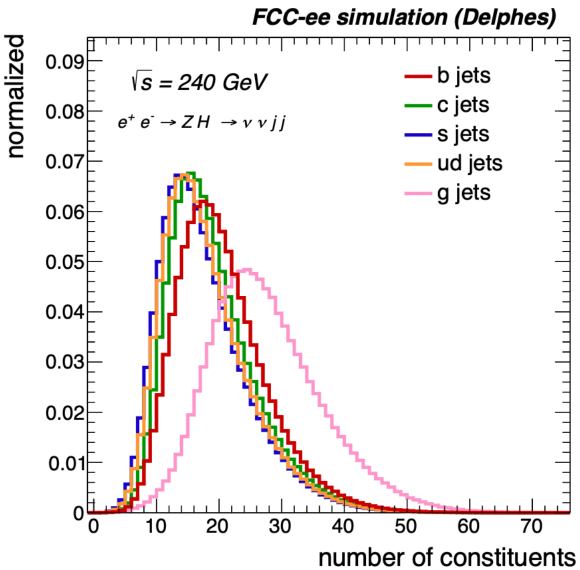
ParticleNet Architecture



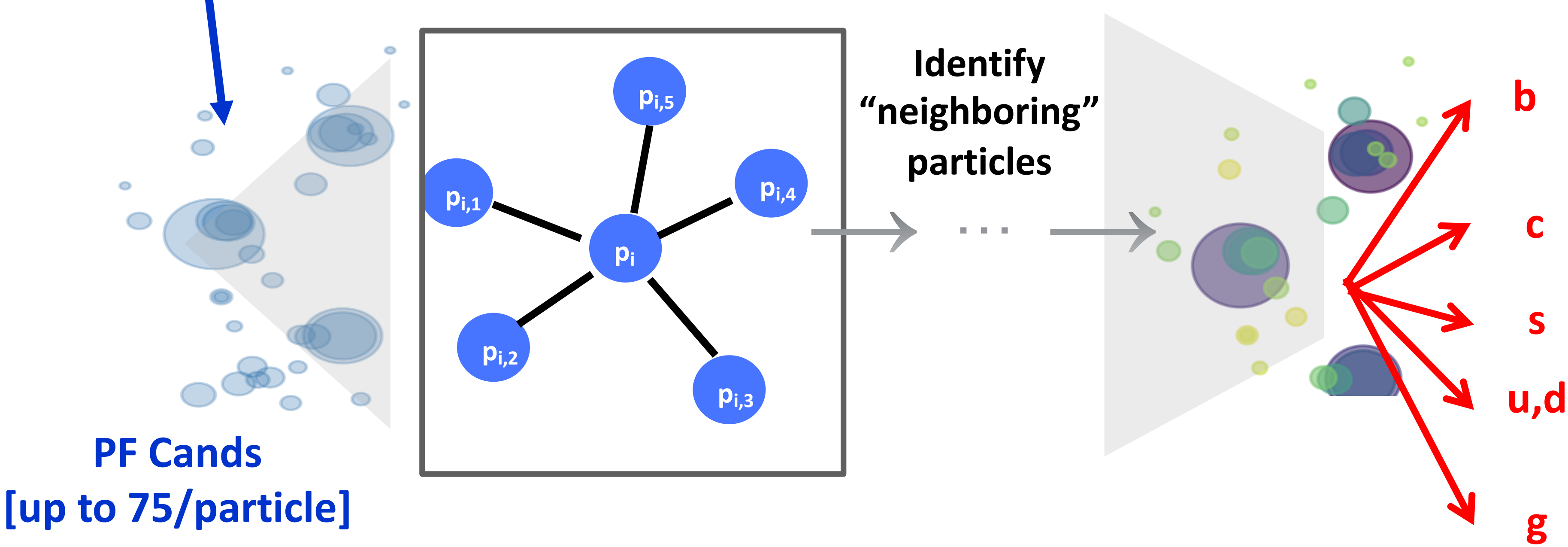
particles distributed in η-ϕ

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

ParticleNet-ee

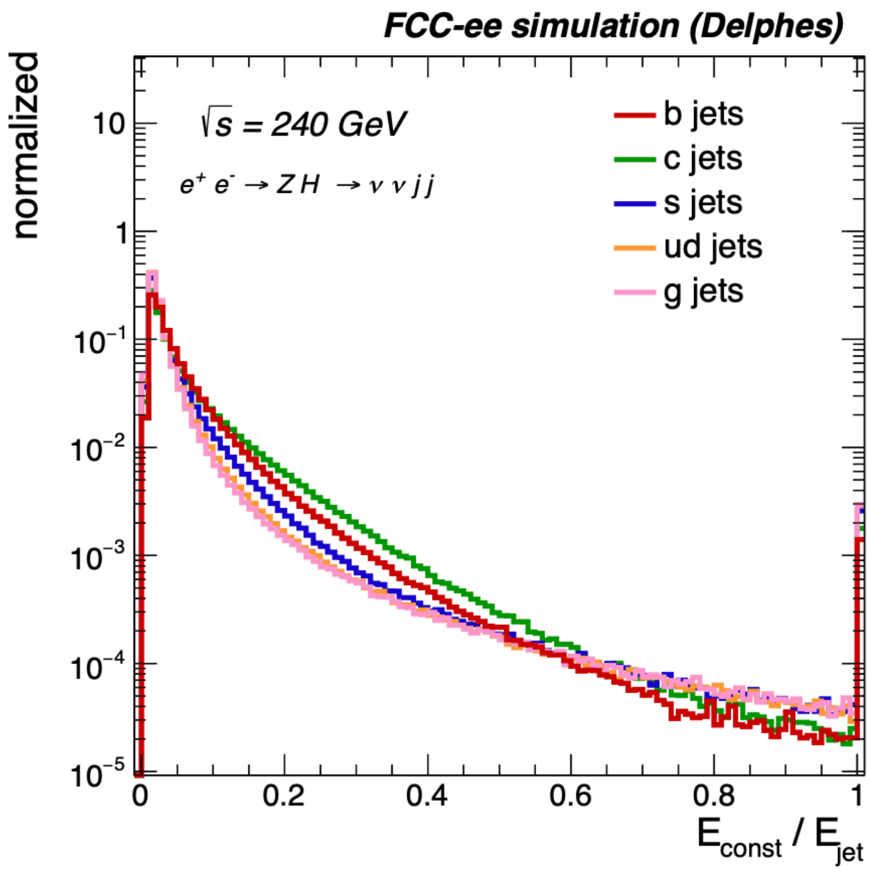


Particle features: O(20)/particle — **Particle kinematics, particle charge, Impact parameter (d0, dz) and significance, particle type (el, mu, γ,...)**

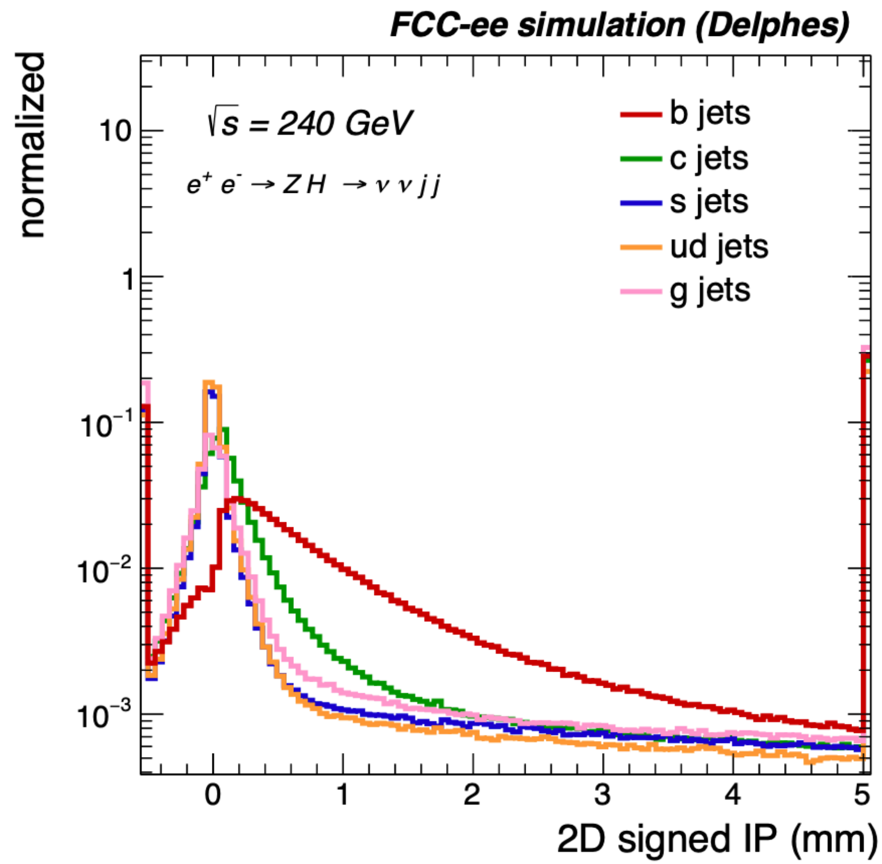


Example of input features

Constituent relative energy

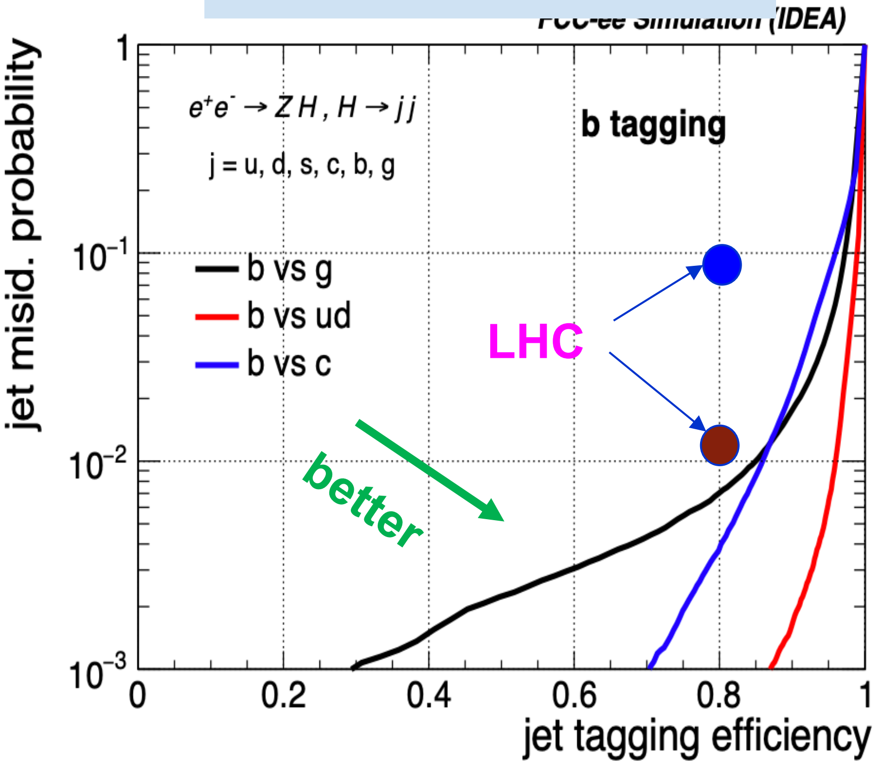


Impact parameter (d_0)



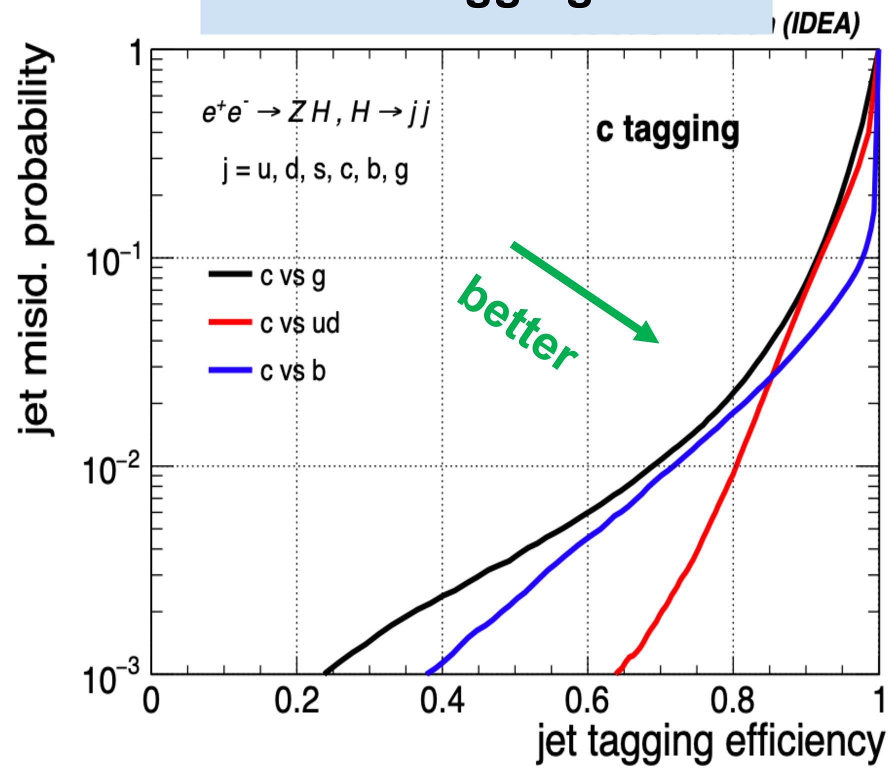
ParticleNet@FCCee: b/c tagging

b-tagging



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

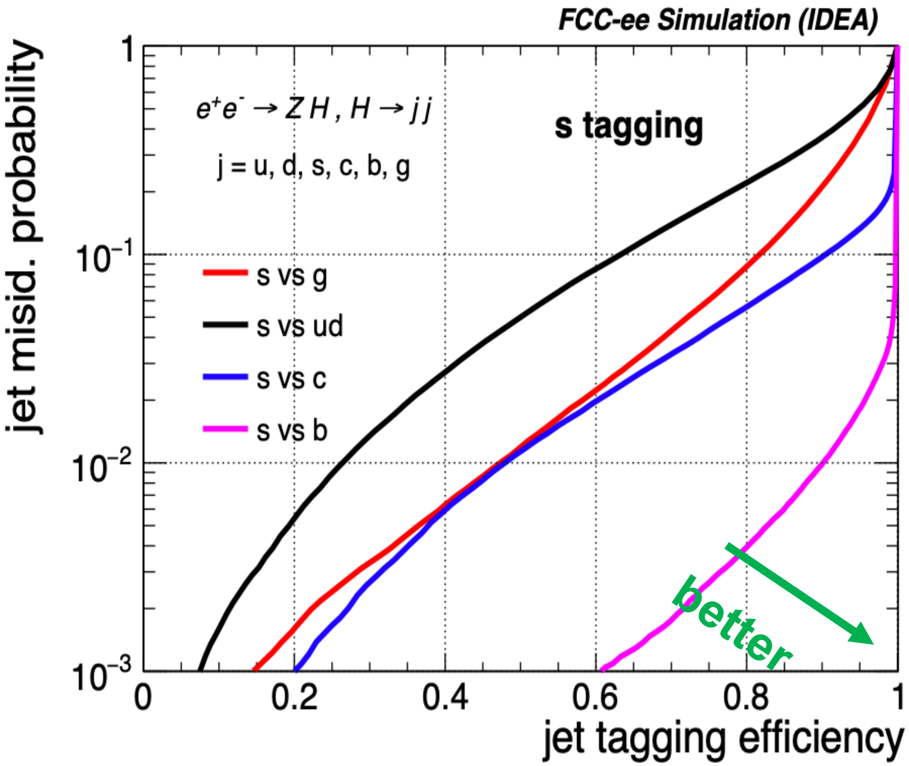
c-tagging



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

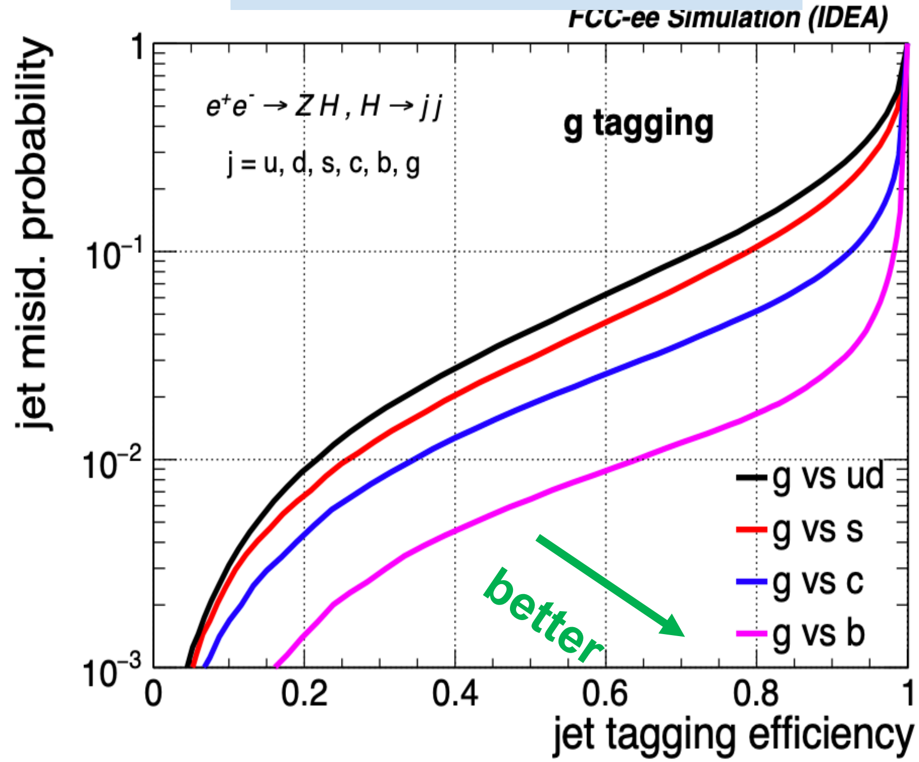
ParticleNet@FCCee: s/g tagging

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%

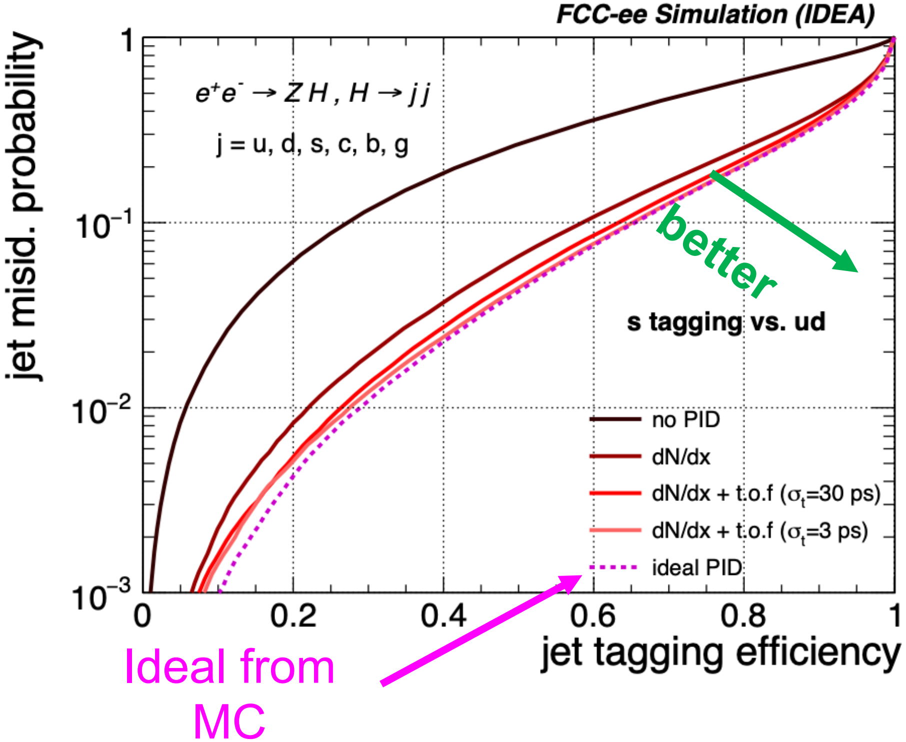
gluon -tagging



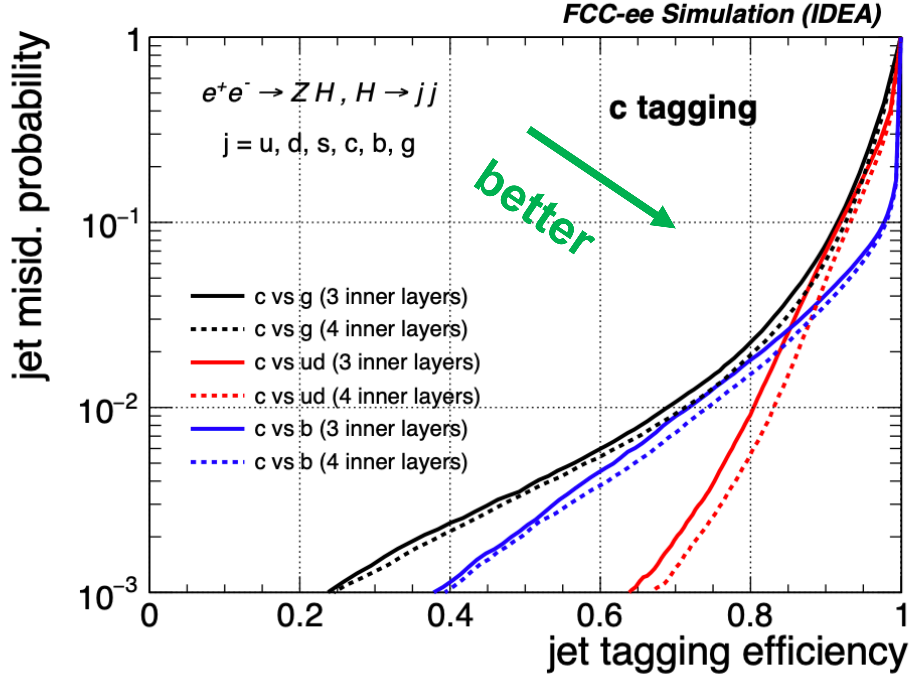
WP	Eff (g)	Mistag (ud)	Mistag (c)	Mistag (b)
Loose	90%	25%	7%	2.5%
Medium	80%	15%	5%	2%

Impact of detector configurations

Strange tagging [PID]



c-tagging [PIX layers]

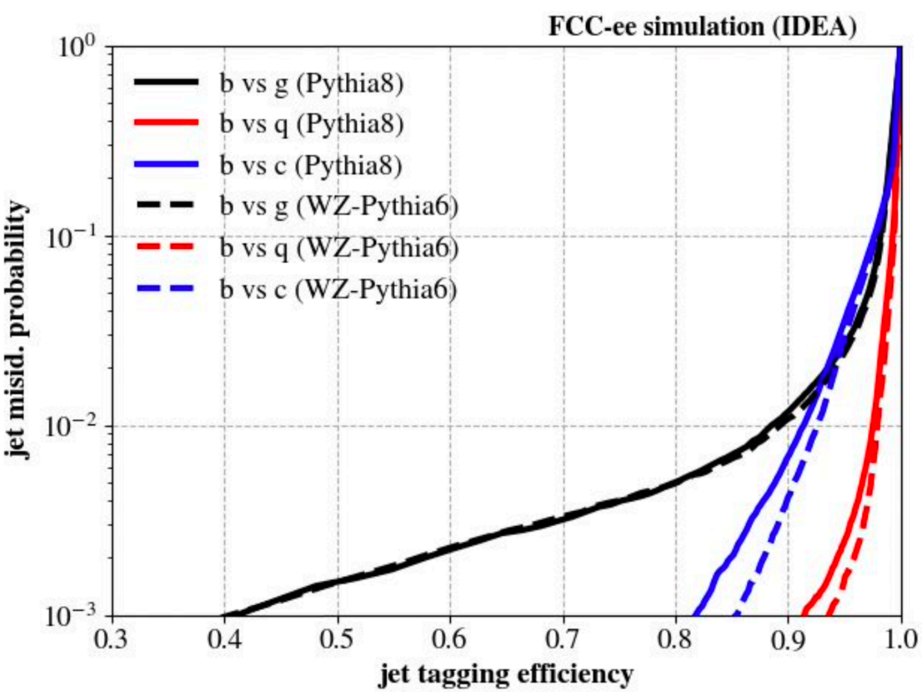


- dN/dx brings most of the gain additional gain w/ TOF (30ps)
  - TOF (3ps): marginal improvement
  - dN/dX + TOF(30ps) ~ perfect PID
- Additional pixel layer:
  - 2x improved BKG rejection in c-tagging
  - marginal/no improvement in b-tagging

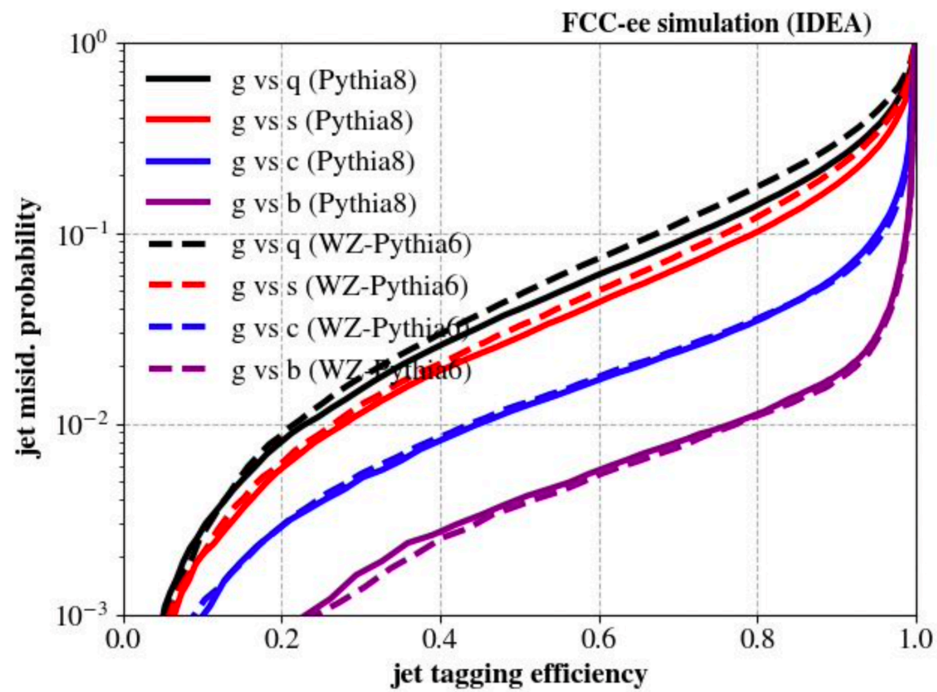
Comparison b/w different generators

- 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
  - still many tricks in our bag to further reduce the dependence

Part 3: Implementation in FCCSW

Mainly for reference
More details: M. Selvaggi's talk on FCCee-Higgs meeting [Dec. 6, 2022]

Executive summary:

- All steps from data preparation, training, and evaluation: validated against original paper, implemented in FCCSW, and documented [link]
- Significant code-cleanup; sequences and selection controlled via configuration files
- Example analysis is place; tests show expected performance

ParticleNet-ee in FCCSW

Example: prepare training dataset

https://github.com/selvaggi/FCCAnalyses/tree/master/examples/FCCee/weaver

Quick tour

stage 1: produce event based tree (Whizard or Pythia8)

```
### test produce input files for training (WHIZARD)
fccanalysis run examples/FCCee/weaver/stage1.py --output test_Hbb_wz.root --files-list /eos/experiment/fcc/ee/generation/Delp
```

```
### test produce input files for training (PYTHIA8)
fccanalysis run examples/FCCee/weaver/stage1.py --output test_Hbb_wz.root --files-list /eos/experiment/fcc/ee/generation/Delp
```

stage 2: produce jet based tree

```
python examples/FCCee/weaver/stage2.py  test_Hbb_wz.root out_Hbb_wz.root 0 100
python examples/FCCee/weaver/stage2.py  test_Hbb_py8.root out_Hbb_py8.root 0 100
```

31 October 2022 Andrea Del Vecchio | ParticleNet 3

ParticleNet-ee in FCCSW (II)

Example: run inference

https://github.com/selvaggi/FCCAnalyses/tree/master/examples/FCCee/weaver

test inference

```
### bb
fccanalysis run examples/FCCee/weaver/analysis_inference.py --output inference_bb.root --files-list /eos/experiment/fcc/ee/ge

### cc
fccanalysis run examples/FCCee/weaver/analysis_inference.py --output inference_cc.root --files-list /eos/experiment/fcc/ee/ge

### ss
fccanalysis run examples/FCCee/weaver/analysis_inference.py --output inference_ss.root --files-list /eos/experiment/fcc/ee/ge

### gg
fccanalysis run examples/FCCee/weaver/analysis_inference.py --output inference_gg.root --files-list /eos/experiment/fcc/ee/ge
```

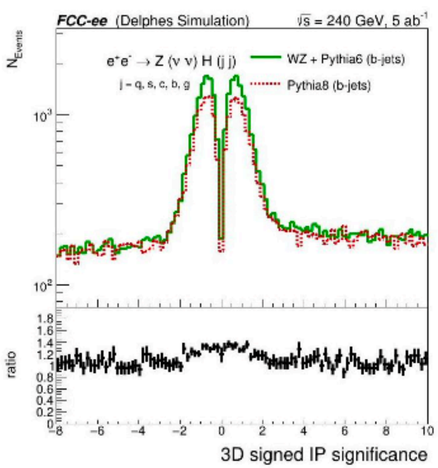
31 October 2022 Andrea Del Vecchio | ParticleNet 4

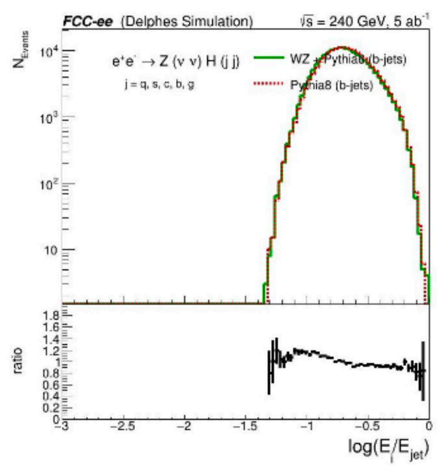
ParticleNet-ee in FCCSW (III)

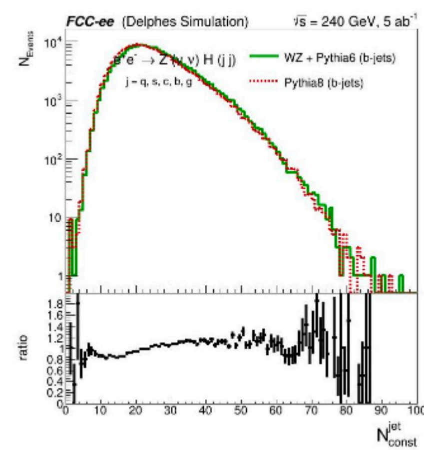
Status update II (validation)

https://github.com/selvaggi/FCCAnalyses/tree/master/examples/FCCee/weaver

- included validation scripts stage_plots.py:
  - to validate all input variables to the tagger
  - run on jet trees used for training (output of stage2.py)







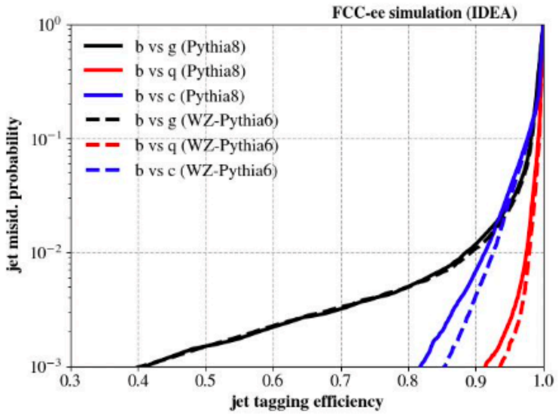
ParticleNet-ee in FCCSW (IV)

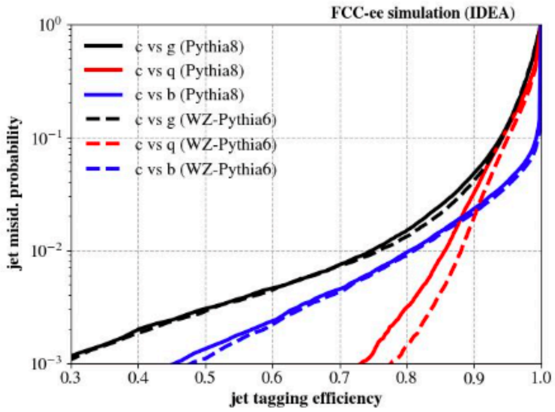
Status update II (rocs)

https://github.com/selvaggi/FCCAnalyses/tree/master/examples/FCCee/weaver

- included validation scripts plot_rocs.py:
  - to validate all input variables to the tagger
  - run on event trees used for physics analysis (output of analysis_inference.py)

```
python examples/FCCee/weaver/plot_rocs.py --indir /eos/experiment/fcc/ee/analyses/case-studies/higgs/flat_trees/zh_vvss_test/
```





Summary

- Precise jet flavour identification is essential for the success of the e^+e^- physics program

- Large effort in both TH and EXP communities to improve existing / develop new jet tools
  - Key player in these developments: Advanced ML algorithms
    - Allows us see much more of the true detector potential
  - Still room for improvement / other ideas to try
    - Strong interest by the theory and experiment communities

- Effort pays off: Large gain in performance wrt traditional approaches
  - Results in collision data look encouraging
  - Lots of effort to better understand what the DNN learns

- Jet tagging methods developed at the LHC can be explored in e^+e^- experiments and potentially enhance the e^+e^- physics program
  - And/or motivate the design of future detectors