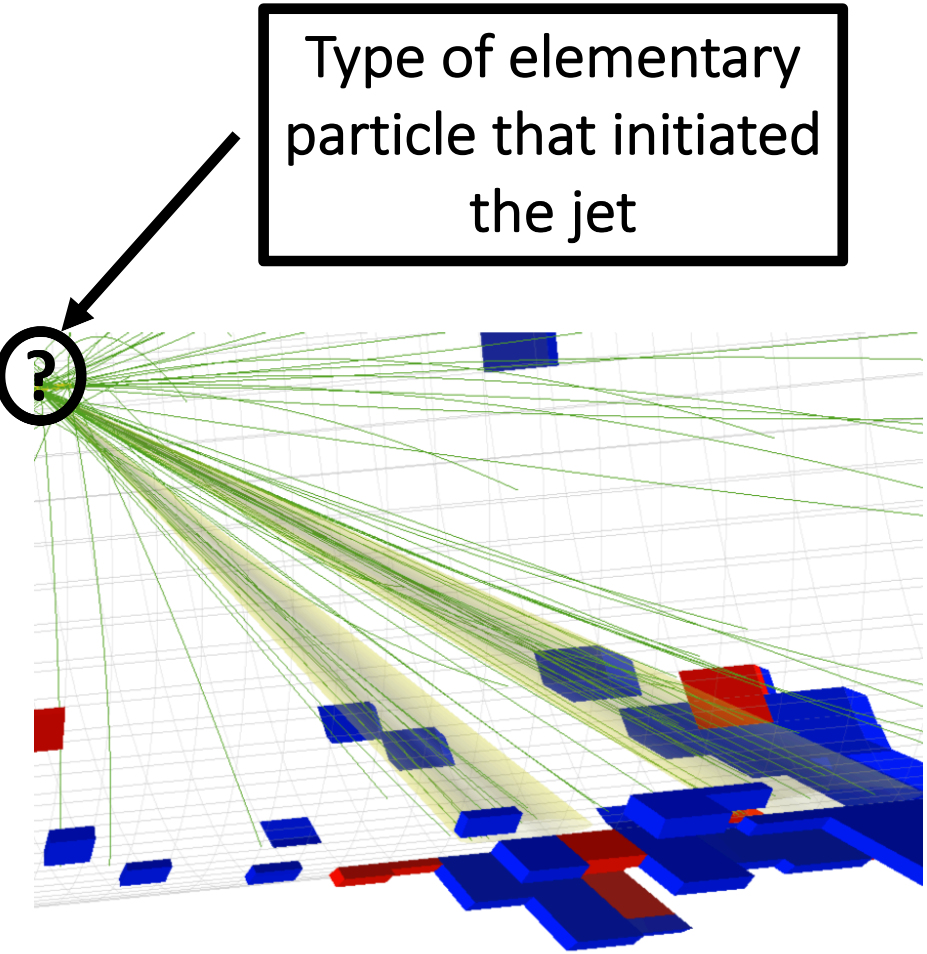
Jet flavor identification [in FCC-ee]

F.Bedeschi, A. Del Vecchio, L. Forthomme, D. Garcia,
Loukas Gouskos, M. Selvaggi,

International Workshop on Future Linear Colliders
SLAC, May 2023

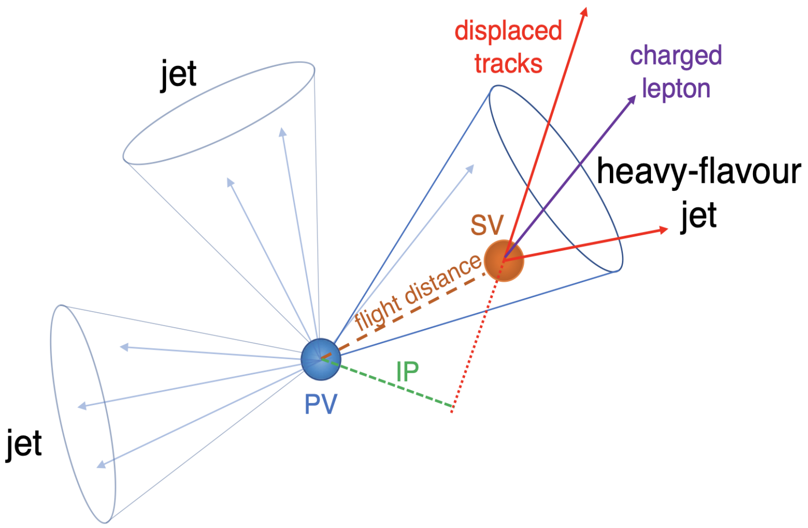
Introduction



- Flavour tagging essential for the e^+e^- physics program
  - Higgs sector
    - Measure couplings better than %-level
  - Top physics
    - precise determination of top quark properties [mass, width,..]
  - QCD physics
    - strong coupling (a_S), event shapes,..
    - modeling of hadronization, MC tuning
  - ...

Basics for jet flavor identification

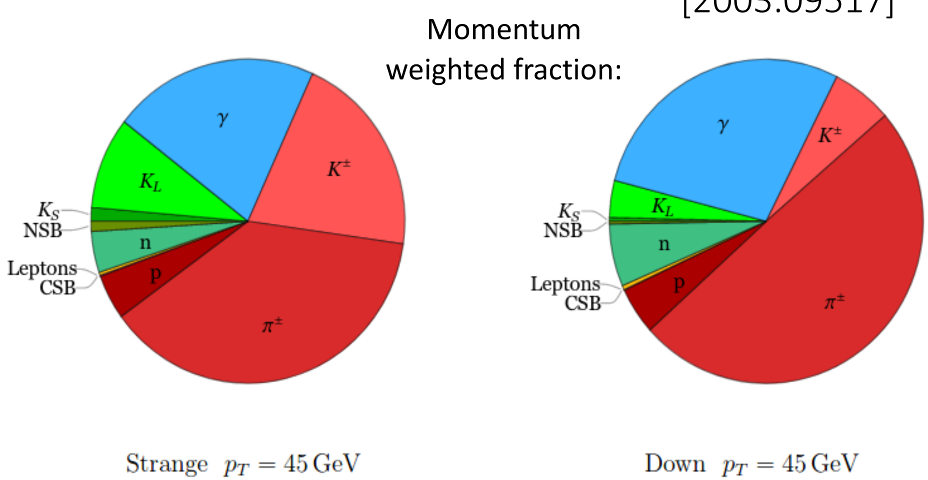
b/c-tagging



- Large lifetime
- Displaced tracks/vertices
- Fragmentation
- non-isolated e/mu

strange-tagging

[2003.09517]



- Large Kaon content
  - Charged Kaon as track
    - K/π separation
  - Neutral Kaons
    - KS→ππ, K_L

In the beginning: unclear what correlations existed among these

Ingredients for powerful jet taggers

- Detectors
  - Pixel/tracking systems: Little material, spatial resolution, precise track alignment
  - PID systems: timing capabilities, energy loss (gas/silicon)
- Algorithm design
  - optimal representation of jet
  - optimal processing of detector signal & evt reconstruction

- Scope of this work:
  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 different particle species
      - Results shown for FCC-ee & IDEA detector

Detectors characteristics in e^+e^-

- 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]

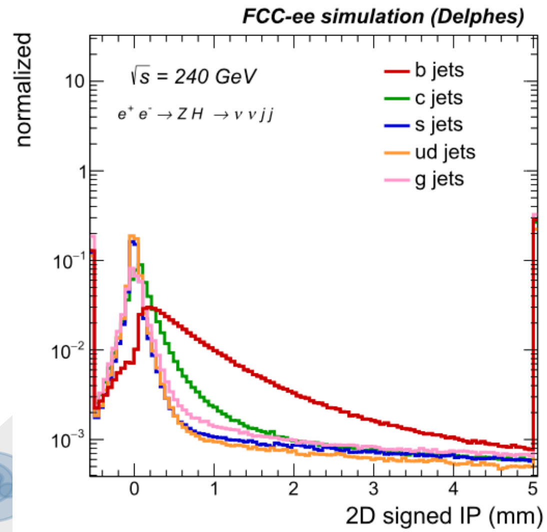
Numbers indicative
concepts evolve rapidly

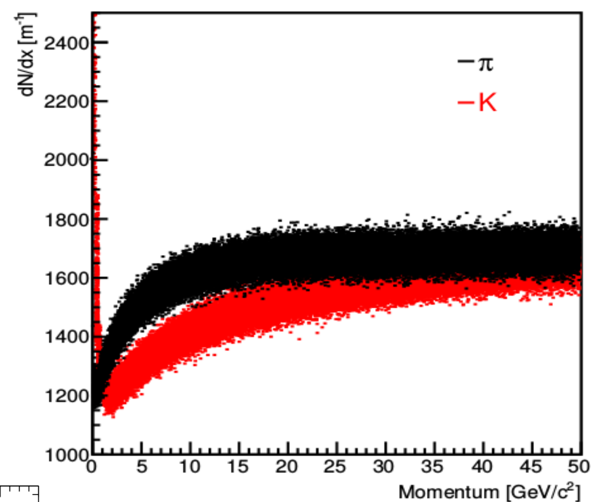
Natural place to explore potential of jet tagging algorithms using advanced ML

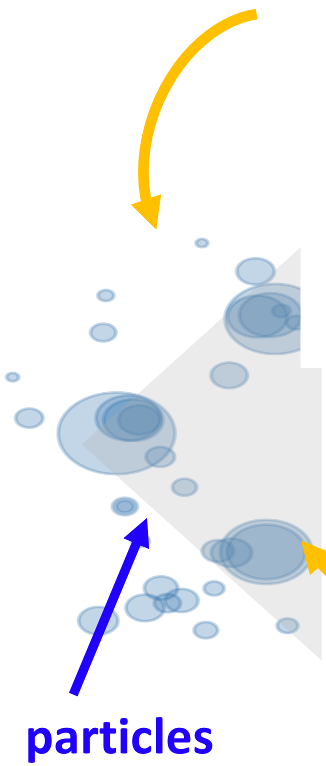
Jet identification via ParticleNet

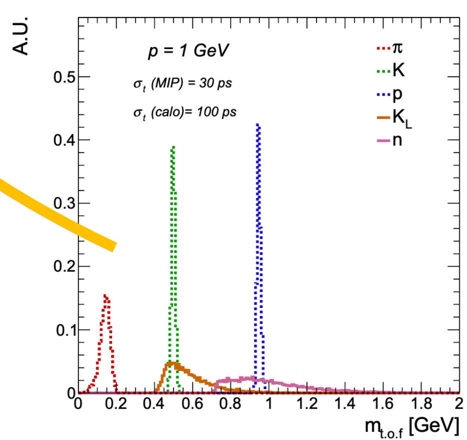
- Jet representation: Particle cloud
  - i.e. unordered set of particles

H. Qu and LG
PRD 101 056019 (2020)
F. Bedeschi, M. Selvaggi, LG
EPJ C 82 646 (2022)







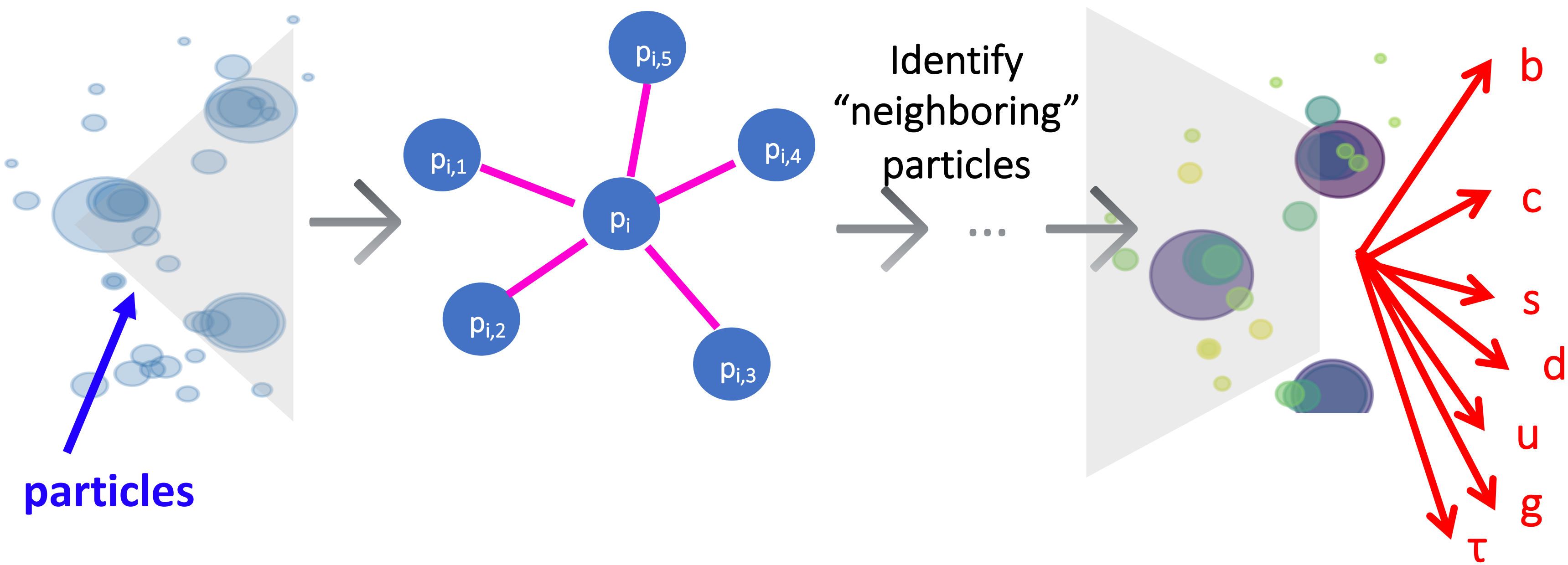


O(20) properties/particle

Jet identification via ParticleNet

H. Qu and LG
PRD 101 056019 (2020)
F. Bedeschi, M. Selvaggi, LG
EPJ C 82 646 (2022)

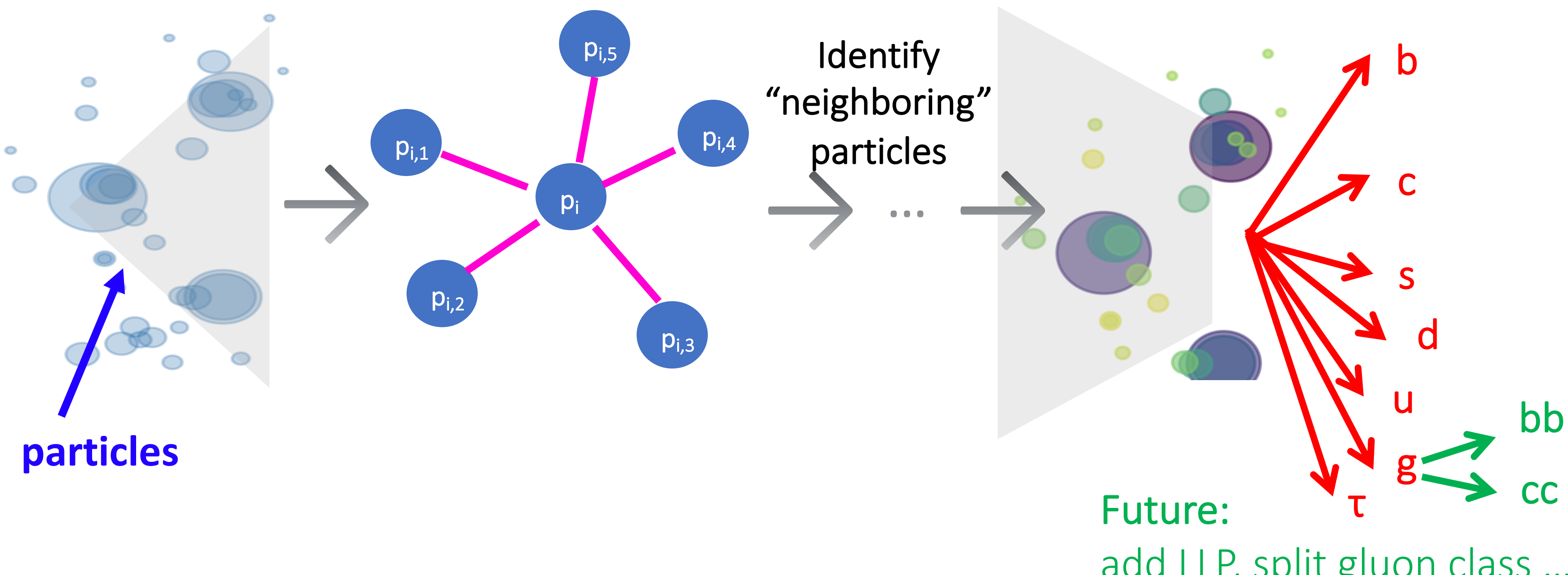
- Jet representation: Particle cloud
  - i.e. unordered set of particles
- Network architecture: Graph Neural Networks
  - Particle cloud represented as a graph
    - particles: vertices of graph; interactions b/w particles: edges of graph
- Hierarchical learning approach: local → global structures



Jet identification via ParticleNet

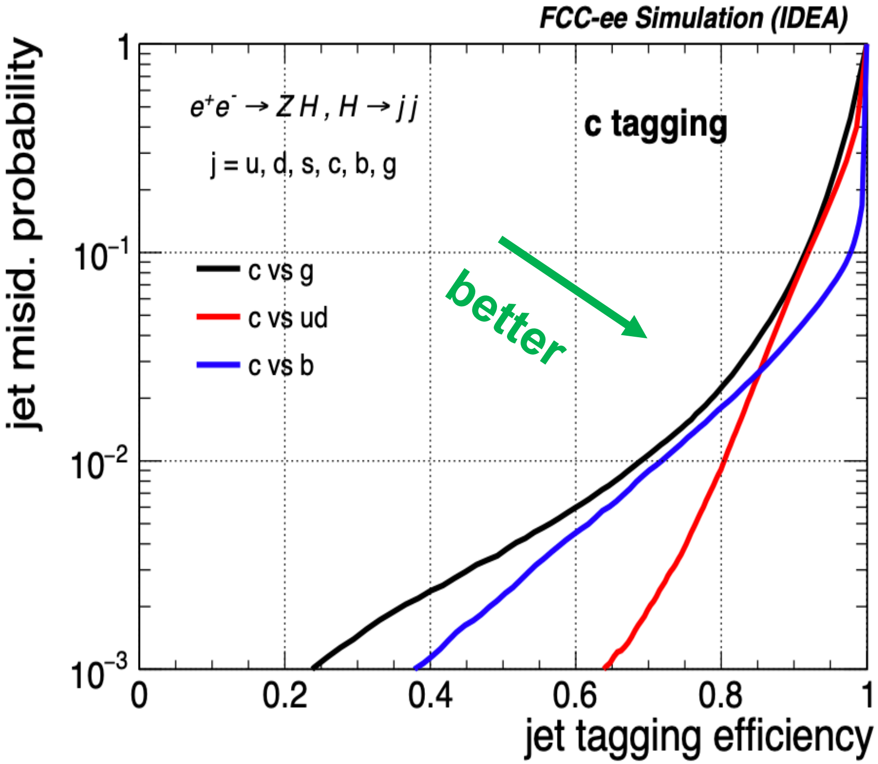
H. Qu and LG
PRD 101 056019 (2020)
F. Bedeschi, M. Selvaggi, LG
EPJ C 82 646 (2022)

- Jet representation: Particle cloud
  - i.e. unordered set of particles
- Network architecture: Graph Neural Networks
  - Particle cloud represented as a graph
    - particles: vertices of graph; interactions b/w particles: edges of graph
- Hierarchical learning approach: local → global structures

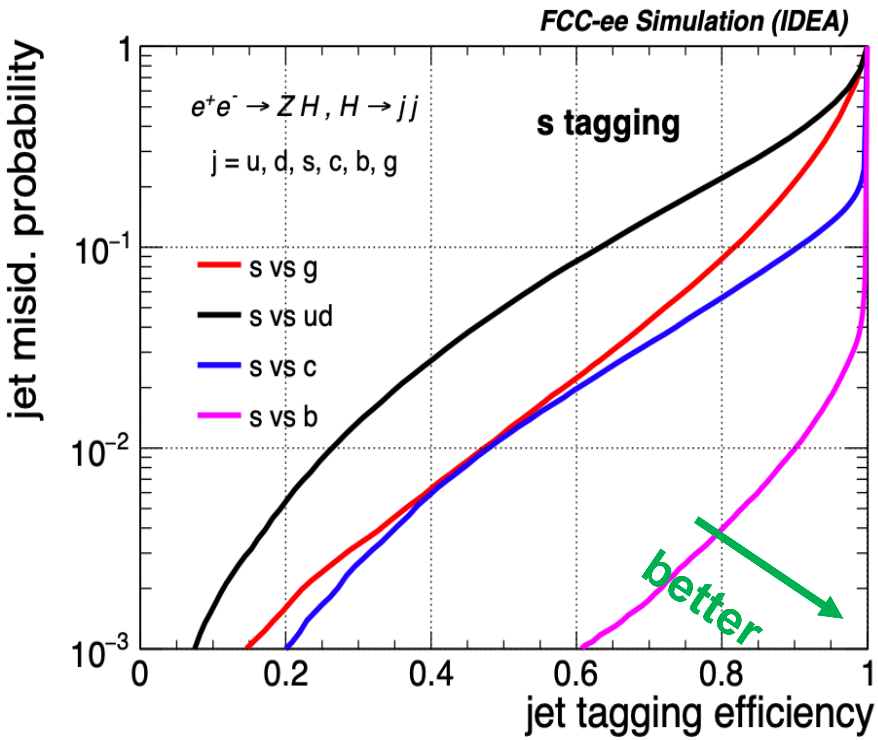


Performance

c-tagging



strange-tagging



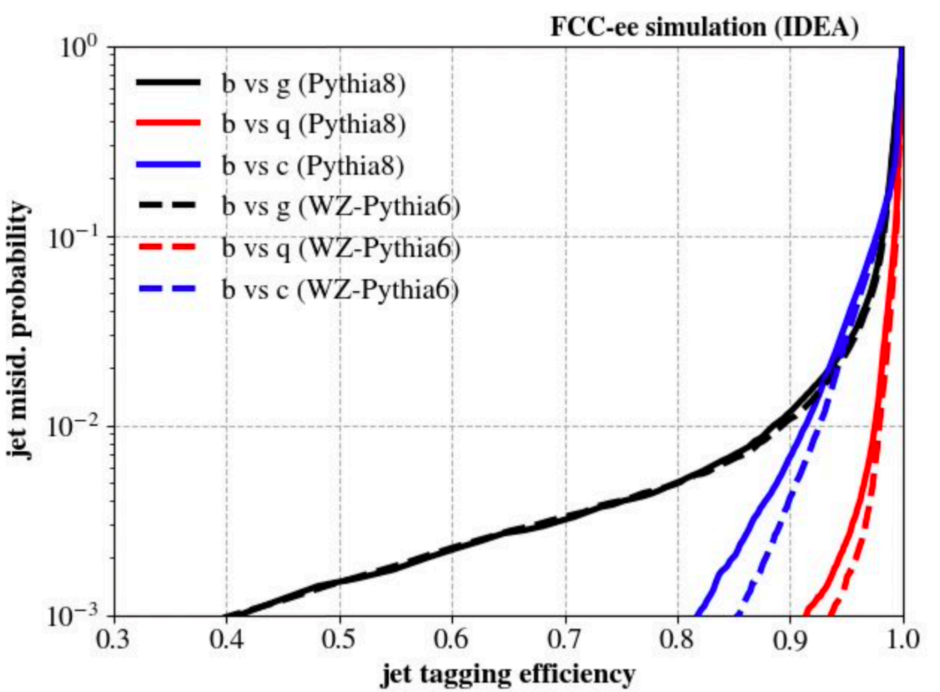
WP	Eff (c)	Mistag (g)	Mistag (ud)	Mistag (b)
Loose	90%	7%	7%	4%
Medium	80%	2%	0.8%	2%

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

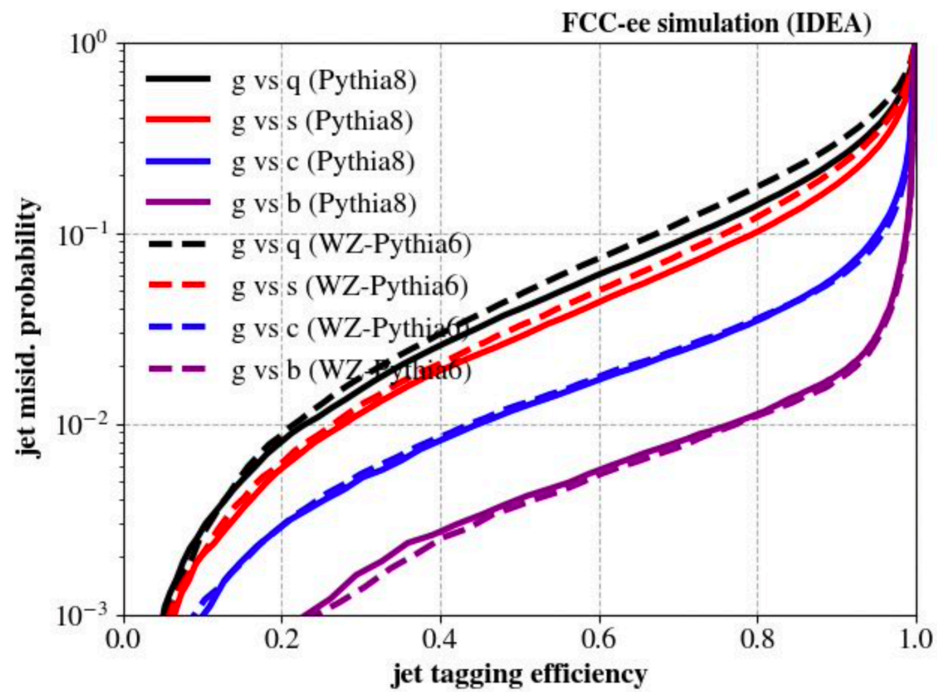
Robustness

- ParticleNet-ee: trained using Pythia8 samples
  - tested on Pythia 8 [solid lines]
  - tested on WZ-Pythia 6 [dashed lines]

b-tagging



gluon -tagging

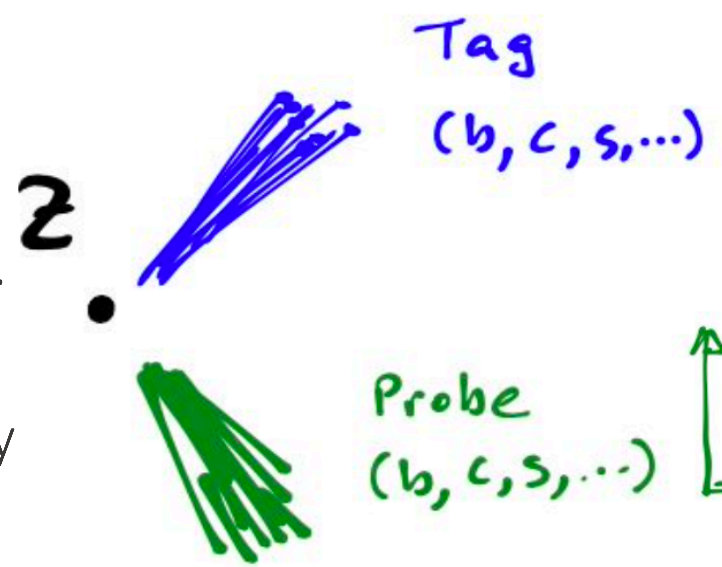


Modest dependence

[still many tricks in the bag to reduce the dependence]

Improving robustness

- Current development relies solely on MC
  - Full control of class definition, lot's of [MC] data [~2M jets/ jet flavor]
    - but: MC != Data; potentially lead to large uncertainties
    - NB: it's also not Full SIM ..
- Another route: Use data
  - [Obvious] advantage: much smaller syst unc.
- How: Tag-and-probe @ Z pole
  - First: Tag one of the two jets with high purity
    - e.g. by using a pretrained MC-based algo
  - Then: create a training sample using the 2nd jet (probe).



FCC-ee @ Zpole

Z→hadrons	~70%	0.7×10^{6} M
→ uu/cc	~12%/flavor	8.4×10^{4} M/ flavor
→ dd/ss/bb	~15%/flavor	**1.1×10^{5} M/ flavor**

Improving robustness (II)

- Take into account tagging performance [& mistag rates]
  - NB: Each class does not have to be 100% pure on specific jet flavor or have the same population

Best case: 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%

More “challenging”: s-tagging

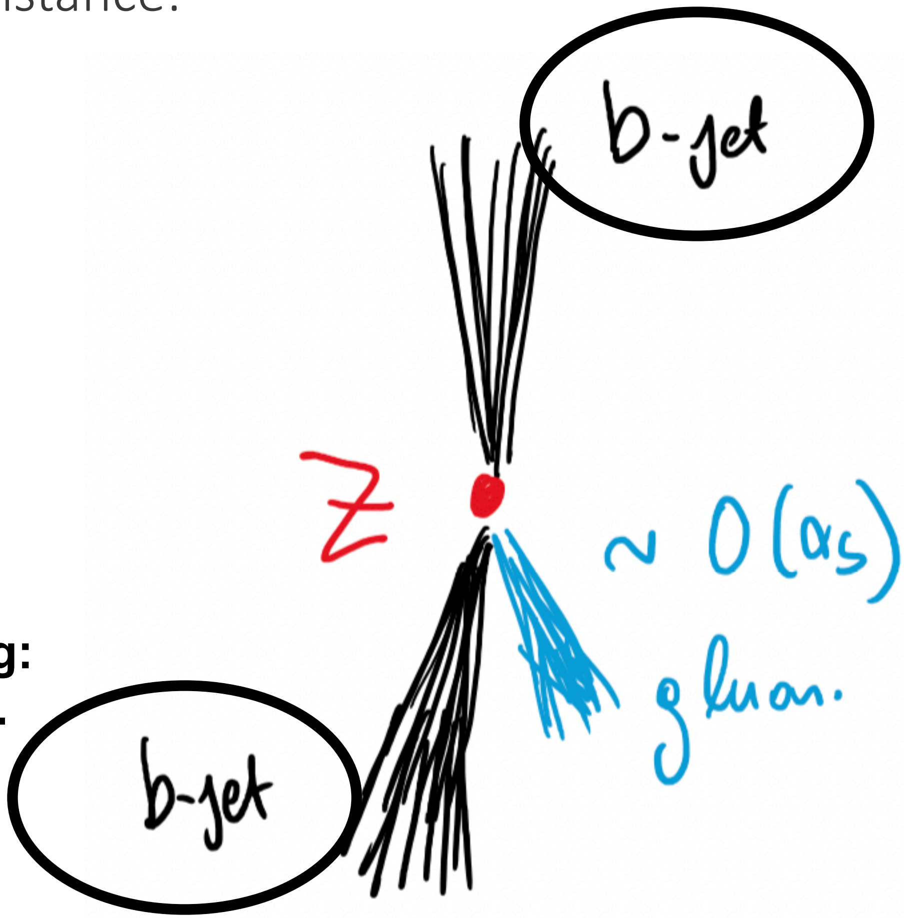
WP	Eff (s)	Mistag (g)	Mistag (ud)	Mistag (c)	Mistag (b)
Loose	90%	20%	40%	10%	1%
Medium	80%	9%	20%	6%	0.4%

- Back-of-the-envelope: Training sample @ Zpole
  - bottom jets: ~1×10^5 M, strange jets: ~8.8×10^4 M
    - all other jet flavors in between

Much larger training sample than what used for the MC-based training sample

Gluon tagging using data?

- Challenging… topic for discussion and brainstorming
  - For instance:



b-tagging:
~90% eff.

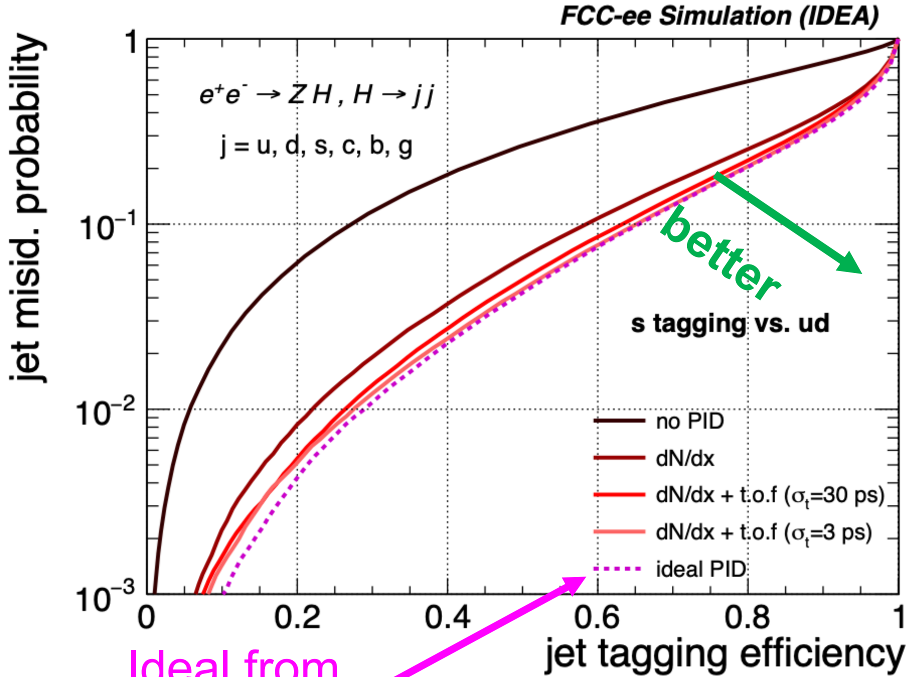
b-tagging:
~90% eff.

- **3rd jet is a gluon: O(1-10%) depending on momentum, angle**
- **Still more than enough for developing the gluon-category**

To be tested

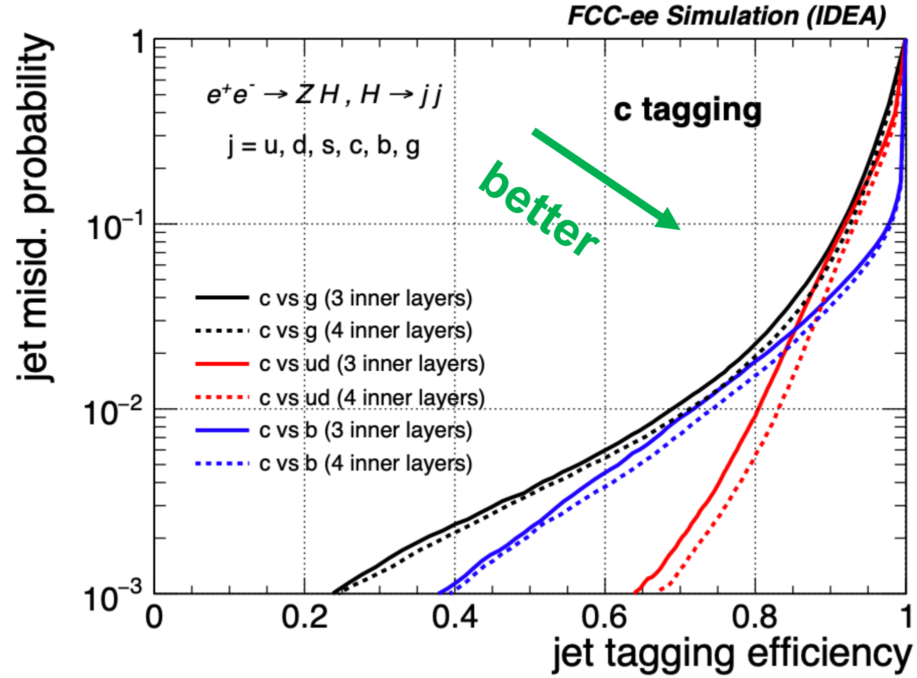
Impact of detector configurations

Strange tagging [PID]



Ideal from MC

c-tagging [PIX layers]

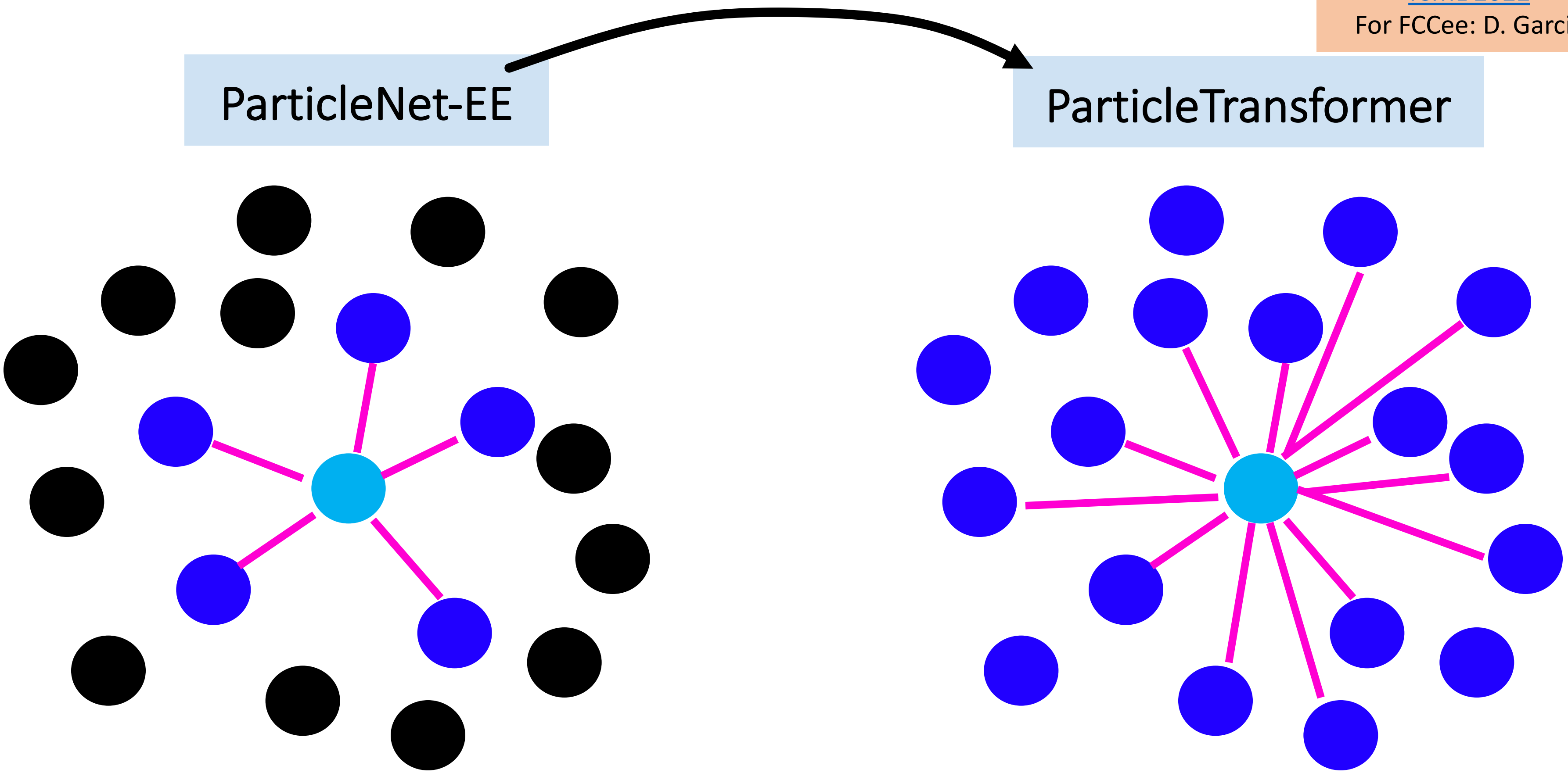


- dN/dX brings most of the gain
  - TOF (3ps): marginal improvement
  - dN/dX + TOF (30ps): ~ perfect PID

- Additional pixel layer
  - c-tagging: 2x improved BKG rejection
  - marginal/no improvement in b-tagging

Pushing the limits further

based on:
H. Qu, C. Li, S. Qian
ICML 2022
For FCCee: D. Garcia

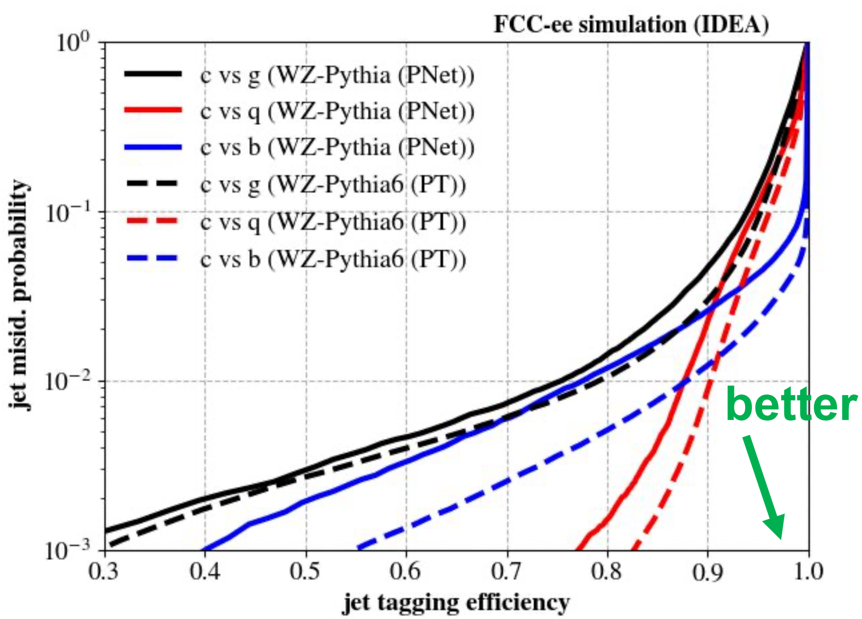


Use the *k*-nearest particles
[*k*=8 for ParticleNet-EE]

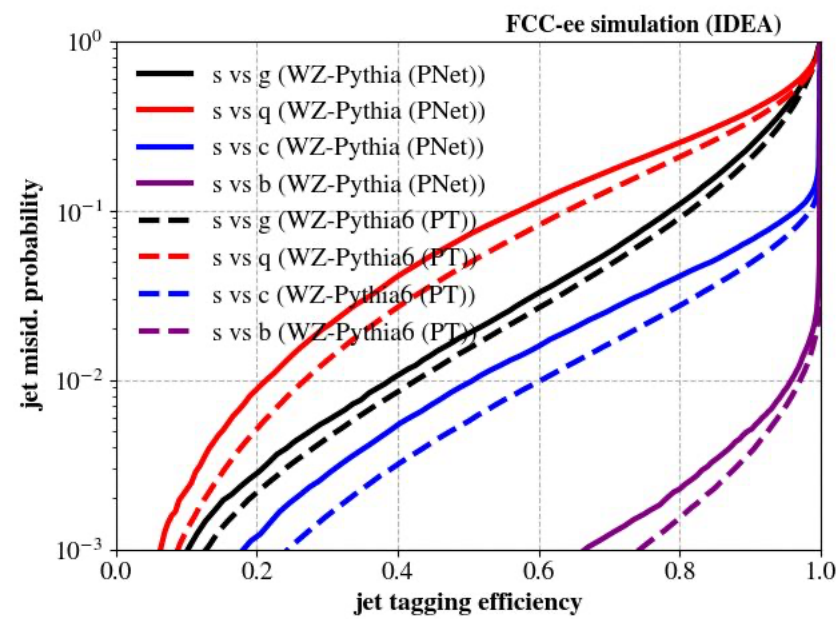
- **Fully connected graph**
- **Include per-particle-pair properties more directly**

Pushing the limits further (II)

c-tagging



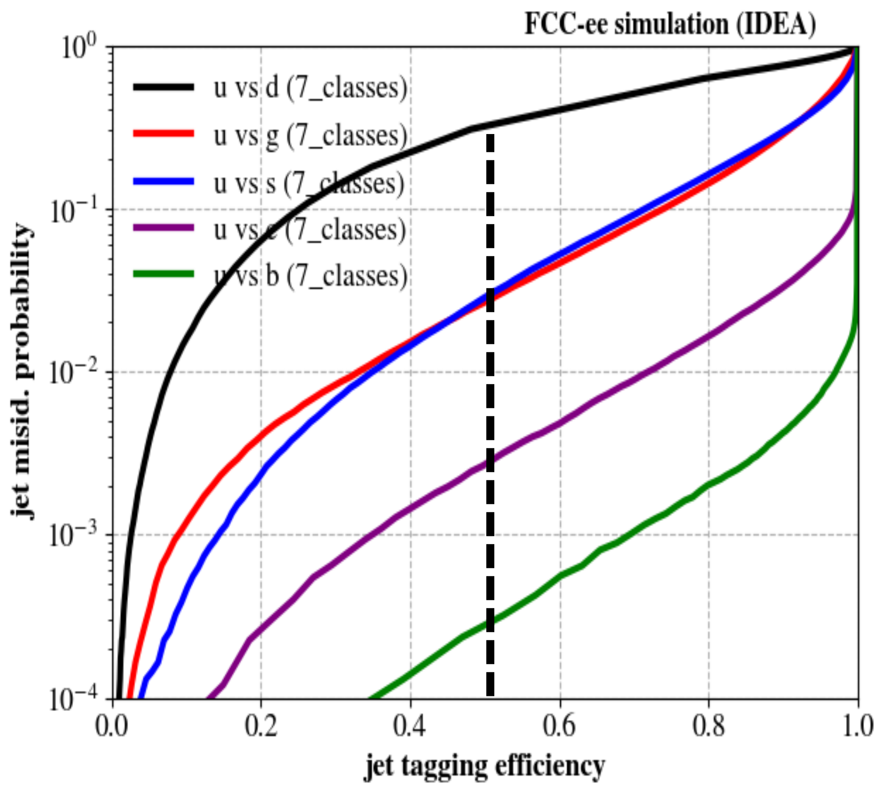
strange-tagging



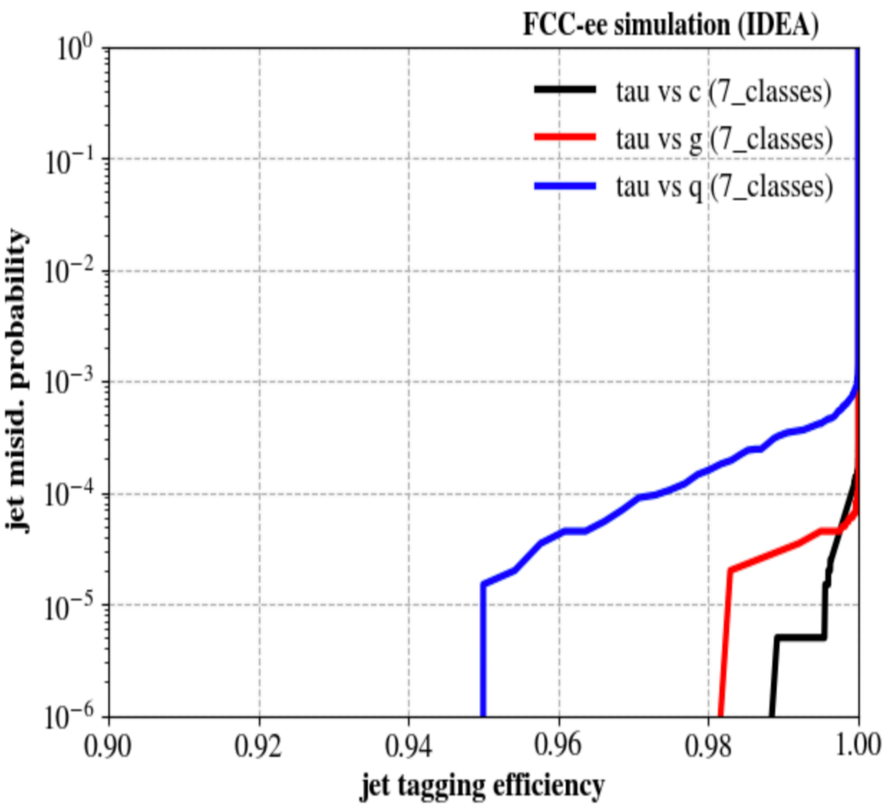
Improvement: up to 2x in BKG rejection

Also: more versatile

up-tagging



tau-tagging



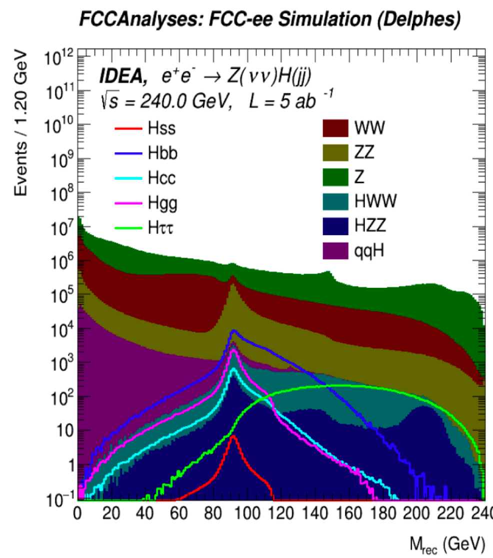
- up/down: better than random guess [thanks to jet charge]
- Tau identification: effectively no signal loss up to 0.1% fake rate

Teaser from analysis front

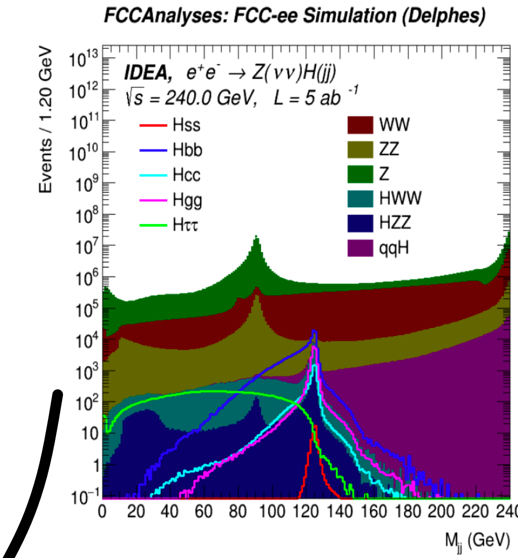
- Tools fully incorporated in FCCSW
  - Example: Z(→vv)H(→qq)

Signal extraction: 2D fit

m(rec)

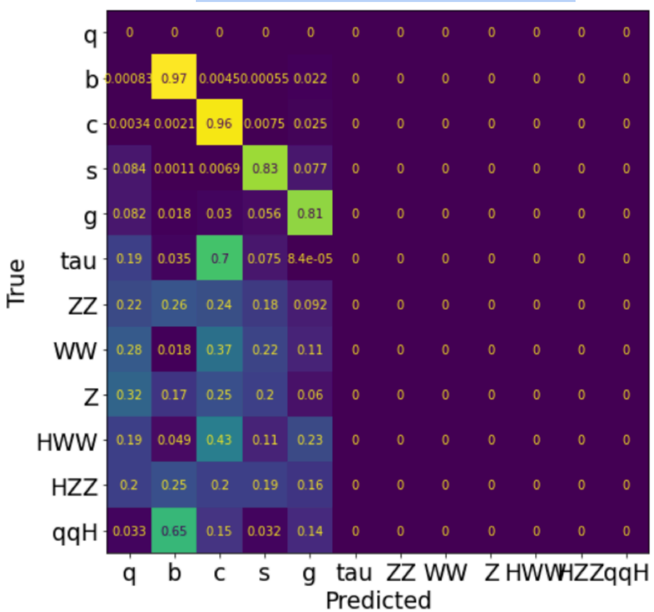


m(jj)



Categorize events: bb, cc, ss, gg
Sub-categories w/ different S/B

ParticleNet-ee



Results @ 5ab^{-1}
(syst: 5% BKG, 0.1% SIG)

Z(→vv) H(→qq)	bb	cc	ss	gg
δμ/μ (%)	0.4	2.9	140	1.2

Summary

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

- Sophisticated jet tagging algorithm developed for FCC-ee
  - Striking improvement in tagging performance compared to previous tools
    - allows us to explore more of the detector and event reconstruction potential
  - Fully integrated in FCCSW [data preparation, training, validation, inference, analysis]
  - Exploration in [FCC-ee] physics analyses started
  - Still room for improvement / other ideas to try
    - Strong interest by the theory and experiment communities

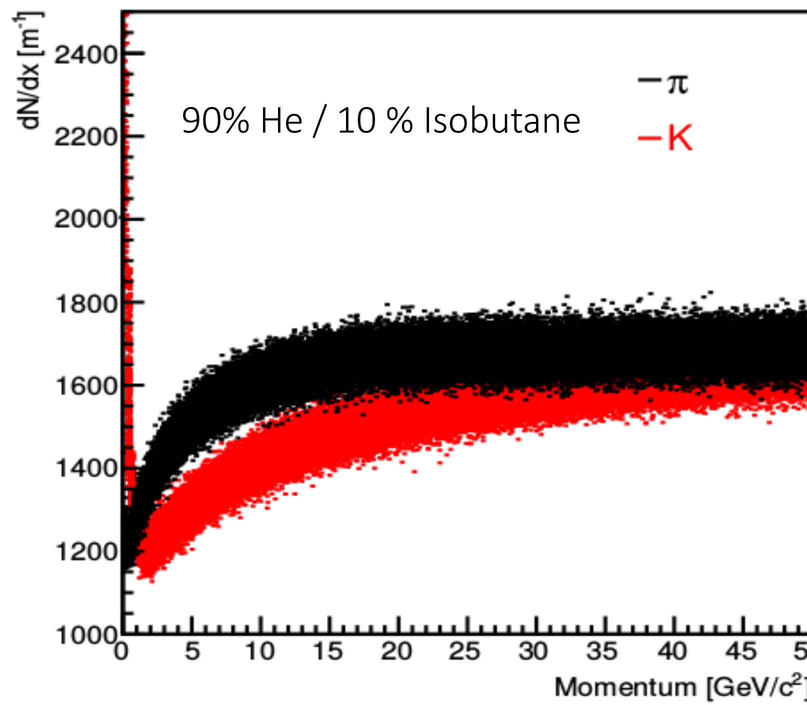
- An obvious area of synergy between the communities of the proposed experiments

Additional material

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:

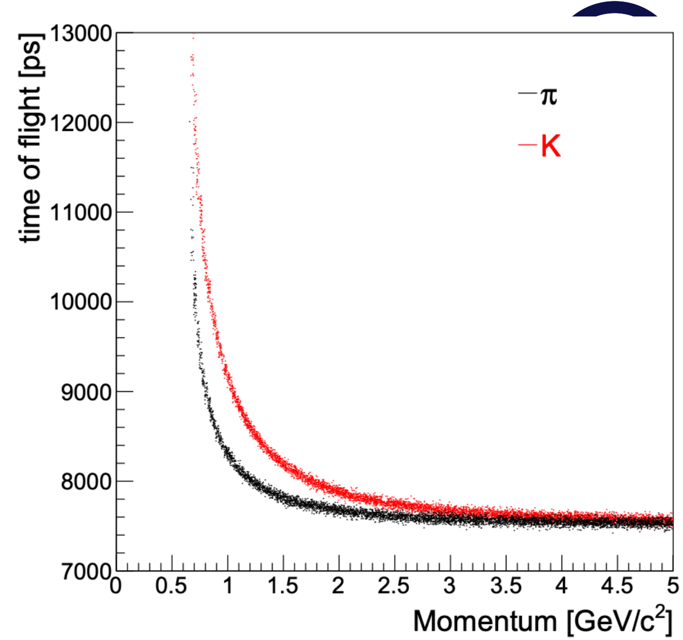



```
###################
# Cluster Counting
###################

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:

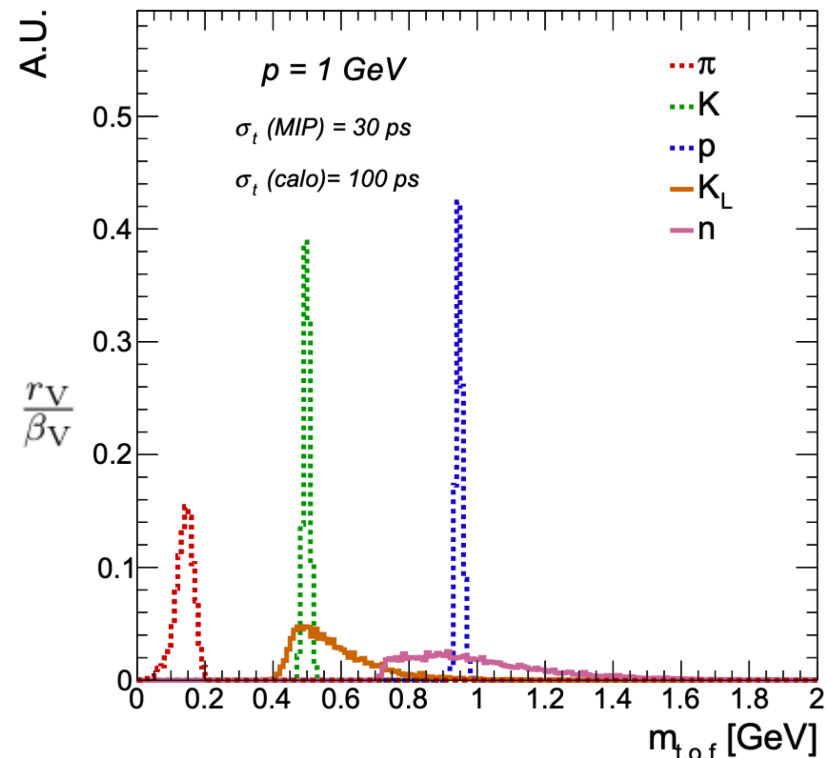
$$t_{\text{flight}} \equiv t_{\text{F}} - t_{\text{V}} = \frac{L}{\beta} = \frac{L\sqrt{p^2+m^2}}{p}$$



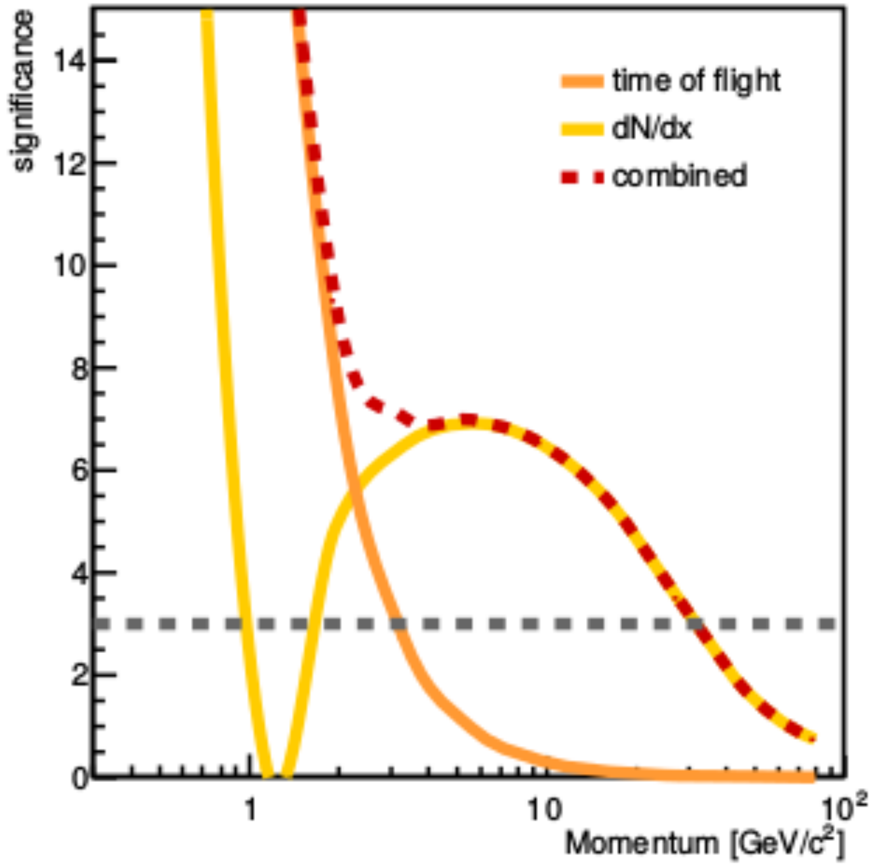
- Assumption on vertex time
  [crucial for highly displaced K_s]

```
########################################
#   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
}
```

$$t_{\text{V}} = \frac{r_{\text{V}}}{\beta_{\text{V}}}$$



ParticleID: Combined



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

Designing a Graph-based tagger

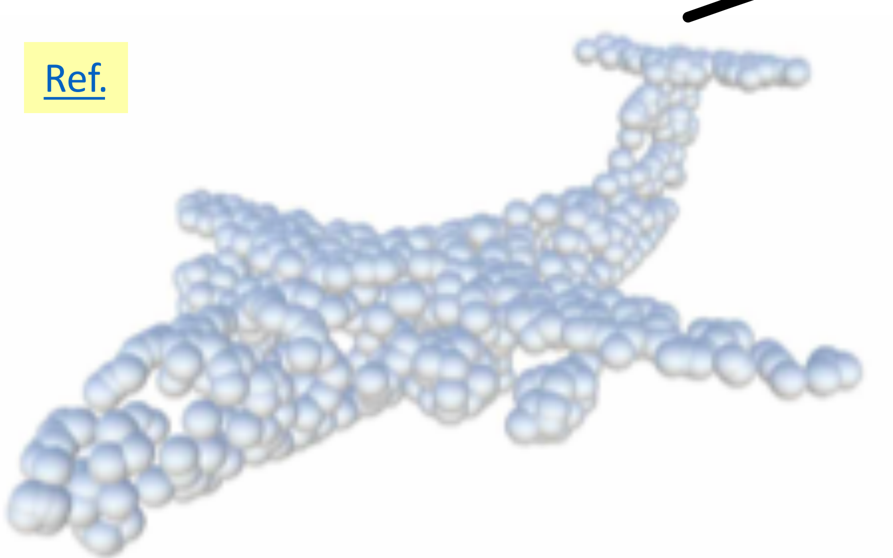
- Jet: intrinsically <u>unordered set</u> of particles with relationships b/w the particles
  - i.e. human-chosen ordering not optimal

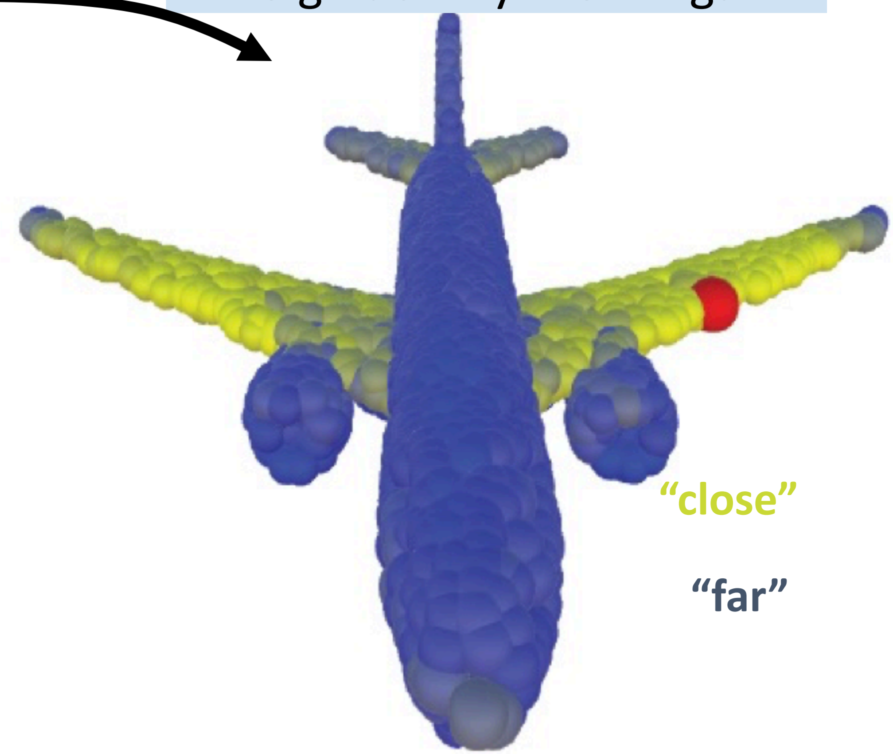
- A v… a in ML communit…

Represent the object as a set of "points"

Group points based on similarity [usually using ML]
e.g. Identify the wings

[Ref.]

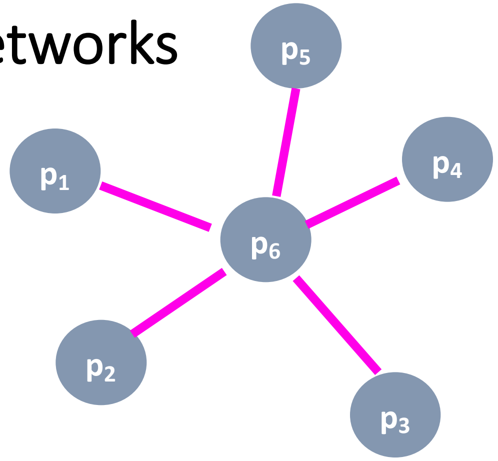




Designing a Graph-based tagger (II)

- 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



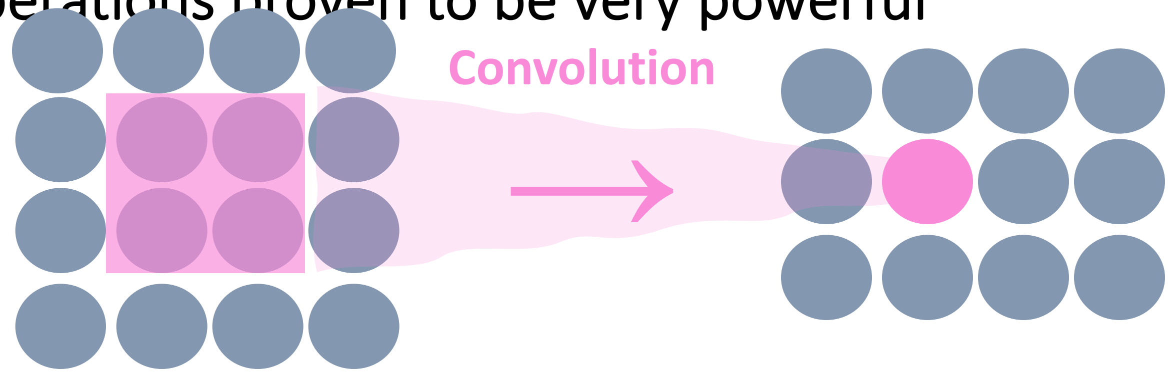
- Build the graph:
  - One approach: Fully connected Graph [but computationally very expensive]
  - Another possibility: apply some criteria
    - e.g., *k*-Nearest Neighbors (*k*NN)

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:

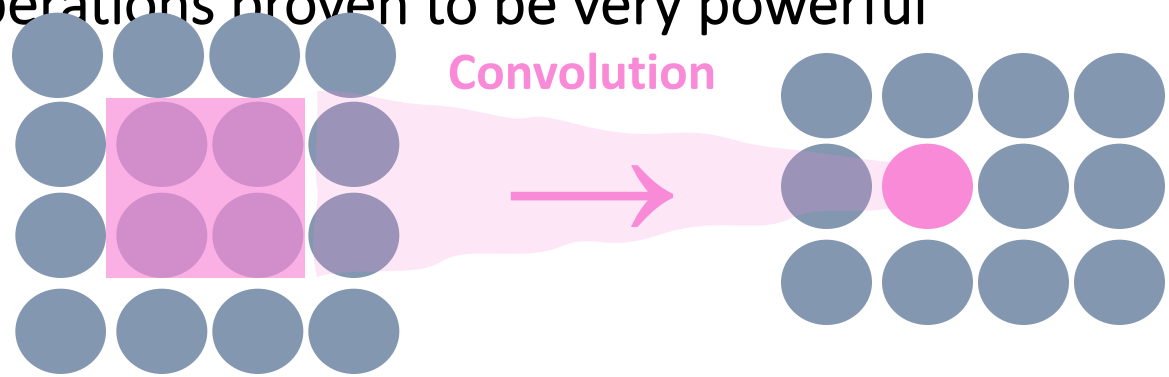


Designing a Graph-based tagger (IV)

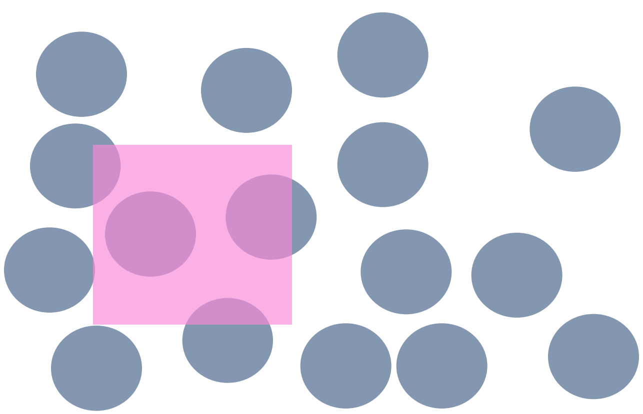
- Last step: Learn from the graphs
  - Follow a **hierarchical learning** approach:
    First learn <u>local</u> structures and **then more <u>global</u>** ones

- Convolution operations proven to be very powerful

Fixed grid:



point/particle cloud:

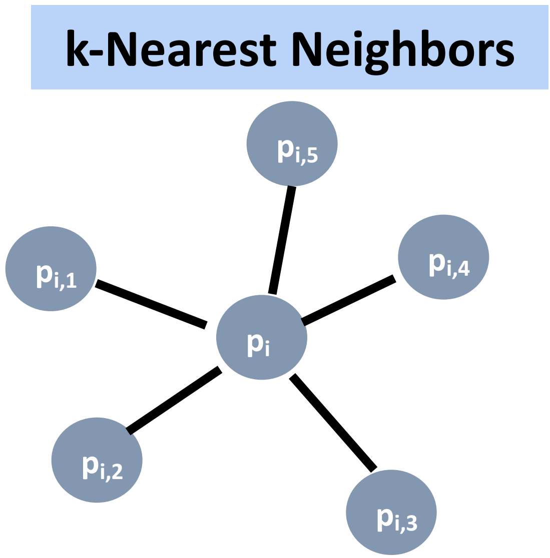


… but not straightforward on point/particle clouds
- Irregular and unordered sets
- Requires a permutation invariant convolutional operation

EdgeConv: Conv. on point clouds

[Y. Wang et al.](#)

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

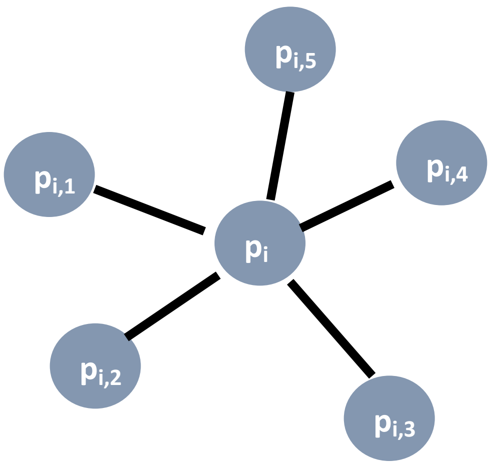


EdgeConv: Conv. on point clouds

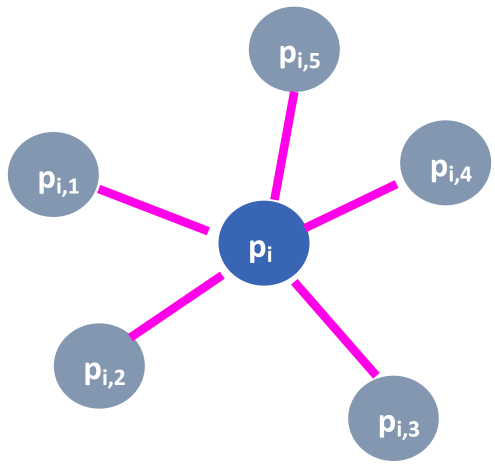
Y. Wang et al.

- 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:

$$p_i' = \underset{j=1}{\overset{k}{\square}} h_\Theta(p_i, p_{ij} - p_i)$$

ParticleNet:

h_Θ: MLP [shared across edges]

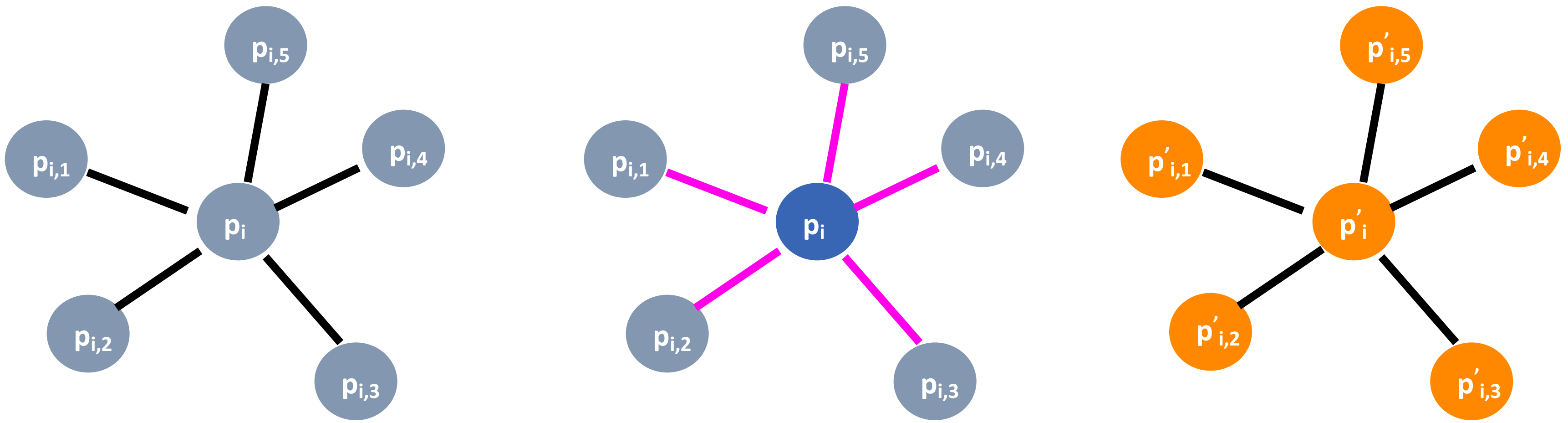
$\square$: average over all *k*-NN

EdgeConv: Conv. 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:

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

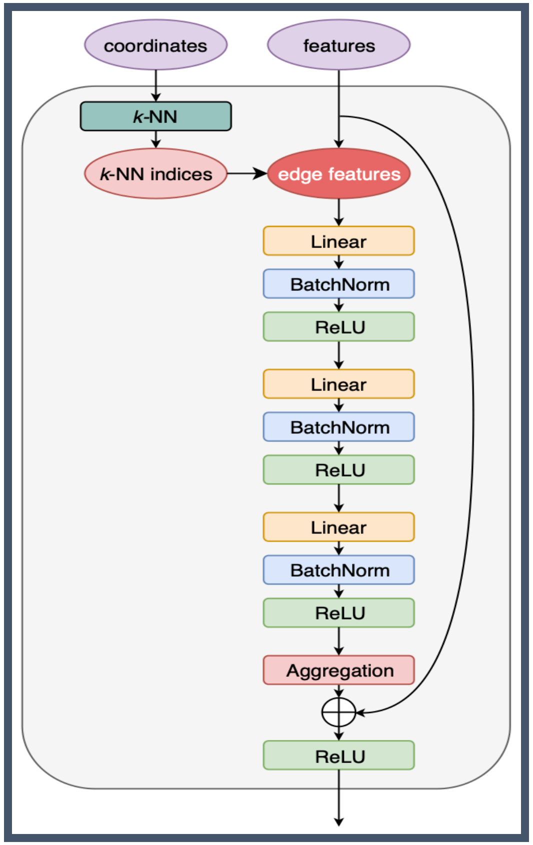
ParticleNet:
h_Θ: MLP [shared across edges]
$\boxed{}$ **: average over all *k*-NN**

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



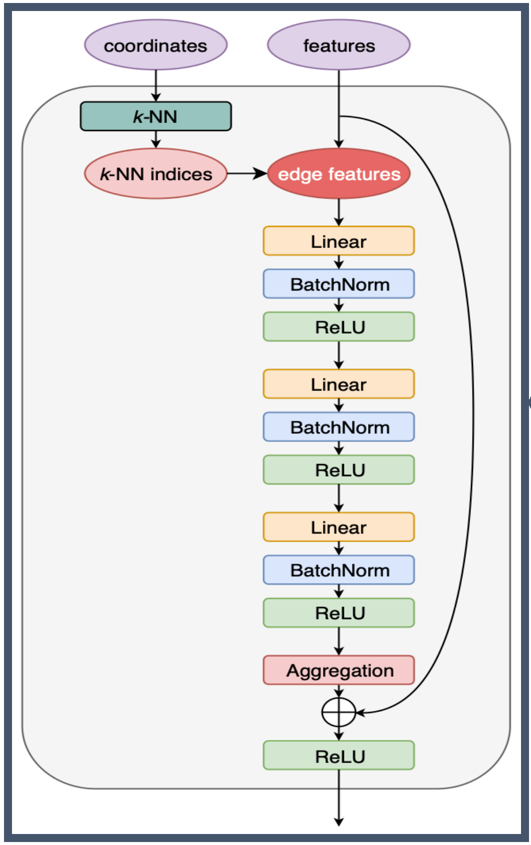
Introduced:
- features beyond spatial coordinates
- residual connections
- MLP conf.

ParticleNet for jet tagging (II)

H. Qu and LG
PRD 101 056019 (2020)

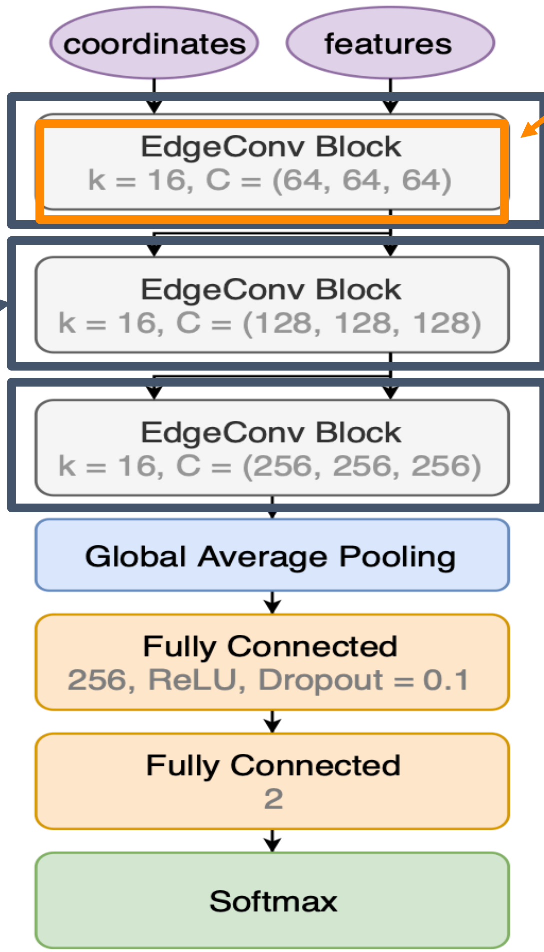
- Based on EdgeConv and DGCNN
  - but customized for the jet tagging task

EdgeConv block



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

ParticleNet Architecture



particles distributed in η-ϕ

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