Strange Tagging using CNN

...

Freya Blekman (DESY+UHH), Florencia Canelli (UZH), Kunal Gautam (VUB+UZH)
Eduardo Plörer (VUB+UZH), A.R. Sahasransu (VUB), Lode Vanhecke (VUB)

Why s-tagging?

- Separating s- and u/d-jets is one of hardest problems in jet flavour tagging
- $Z \rightarrow ss$ decay width measurement, potential Higgs$\rightarrow$strange studies, tool for various BSM studies (FCNC top, etc)
- Improvement in s-tagging with the use of ML
- Good detector performance test

Related Work

- There have been similar studies for hadron colliders, and ongoing studies for e+e- colliders.
  - 2003.09517 (Nakai et. al.)
  - 1811.09636 (Duarte-Campderros et. al.)
  - 2011.10736 (Erdmann et. al.)
  - And others

Development and analysis of a strange quark tagger for future electron-positron colliders

Lode Vanhecke

June 4, 2021

*Lode Vanhecke's Bachelor Thesis at VUB

Developing on the work by Lode Vanhecke and AR Sahasransu:

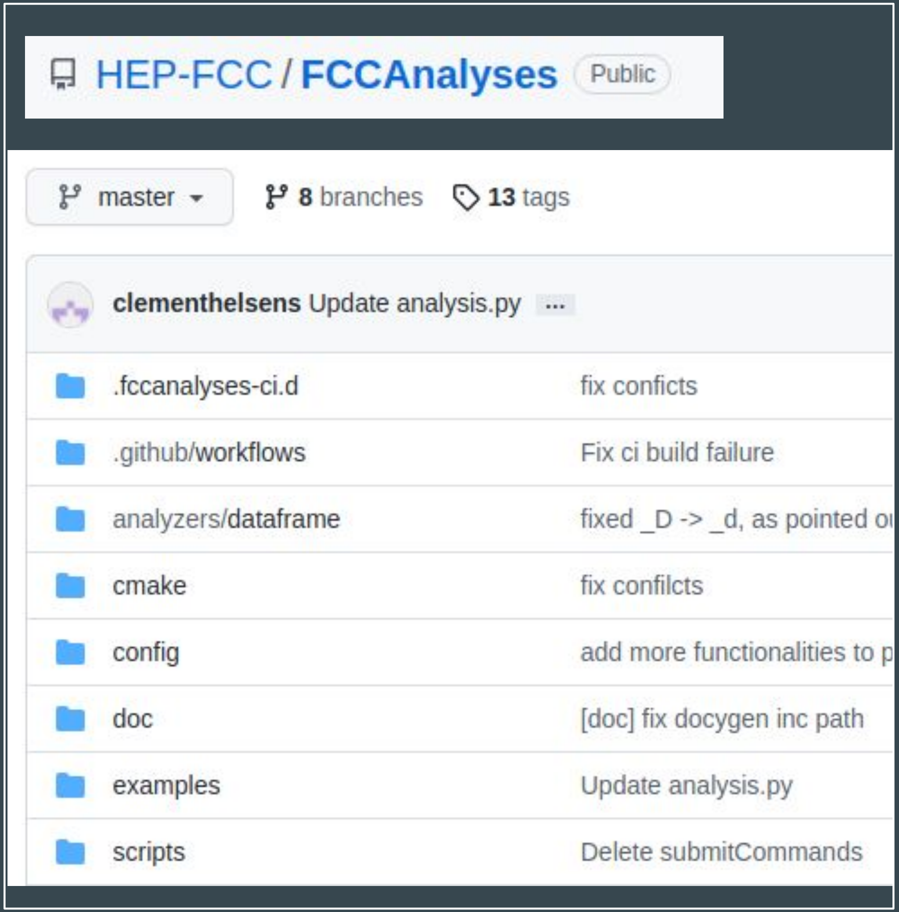
- Event generation using MadGraph(v2.6.6) and Pythia(v8.243)
  - Now using FCCee samples
- Using **gen level** information only
- Here jet clustering was done using anti-kt algorithm with a radius of 0.4
  - Upgraded in the study presented today to eekt (Durham) algorithm
- 2D Angular distribution of jet constituents around the jet axis as the distinguishing variable
  - Particle ID assumptions
- **No decay lengths/lifetime info used**
- Strange jet tagging using a CNN

Strategy/This talk: To confirm/reproduce the results in that thesis using Spring2021 FCCee IDEA samples

Spring2021 IDEA Samples

3	p8_ee_Zuds_ecm91	1,000,000,000
4	p8_ee_Zcc_ecm91	1,000,000,000
5	p8_ee_Zbb_ecm91	1,000,000,000

- Preselection in FCCAnalyses to make ntuples with 600,000 Zuds events (only Gen-level information):
  - Analysed 100,000 events out of these to study jet behaviour
  - Trained the CNN model with these 100,000 events
  - Tested the model on the next 200,000 events



Analysis Frameworks

coffea 0.7.8

```
pip install coffea
```

Tools for doing Collider HEP style analysis with columnar operations

coffea makes use of uproot and awkward-array to provide an array-based syntax for manipulating HEP event data in an efficient and numpythonic way.

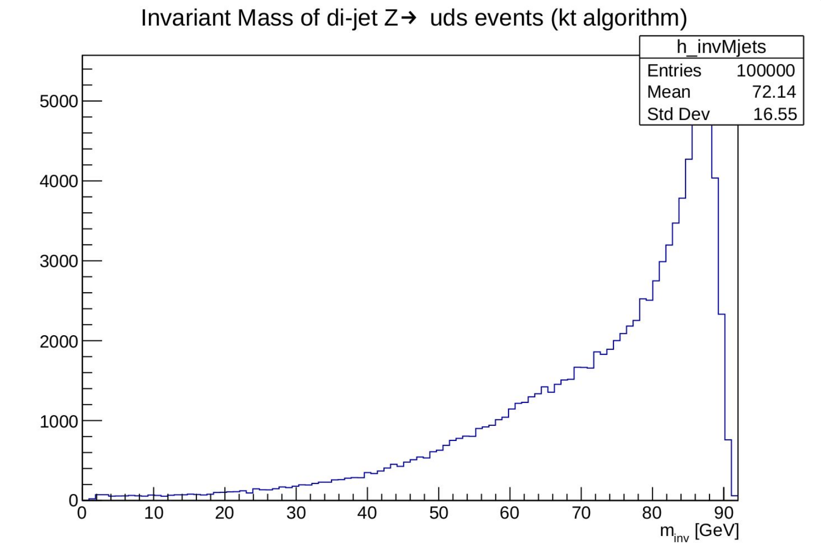
ROOT
Data Analysis Framework

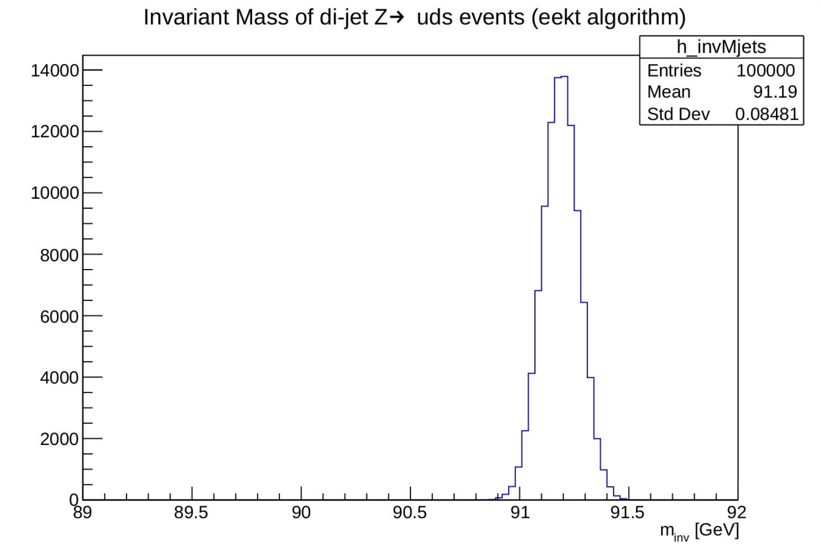
A parallel analysis on ROOT 6.24/04

Jet Clustering

Jet Clustering: Choice of Algorithm

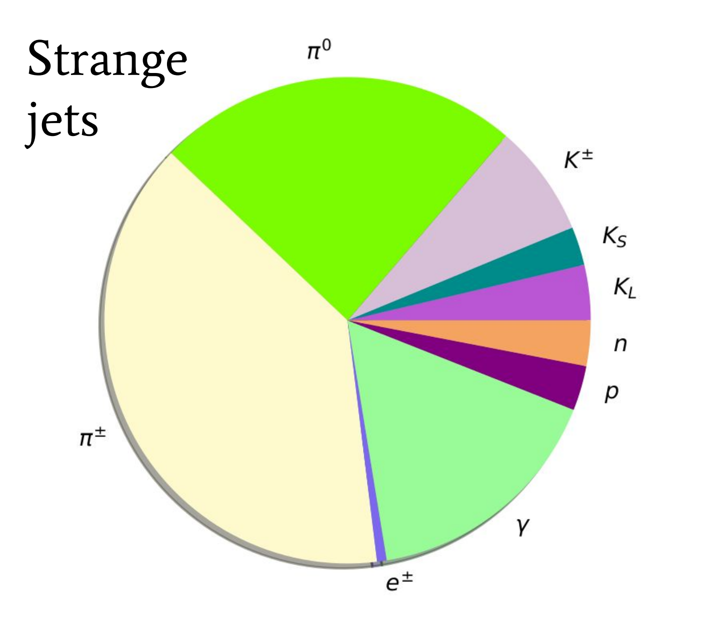
Exclusive clustering into exactly 2 jets using E-scheme for 100,000 Z⟶uds events

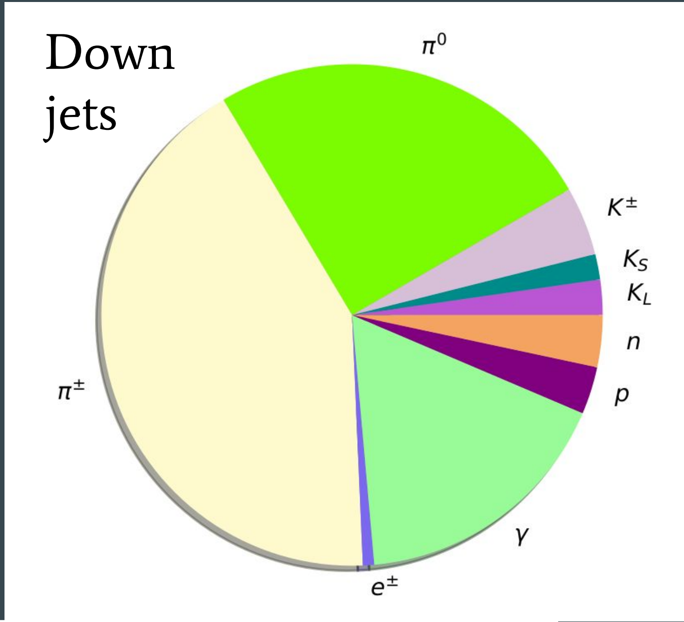




Due to absence of a distance parameter with respect to the beam in the *eekt-algorithm*, particles close to the beam are not thrown away but clustered into jets, unlike the *kt-algorithm*; therefore jet axis and jet cone spread depend on the choice of clustering algorithm.

Jet Clustering: Constituent Distribution

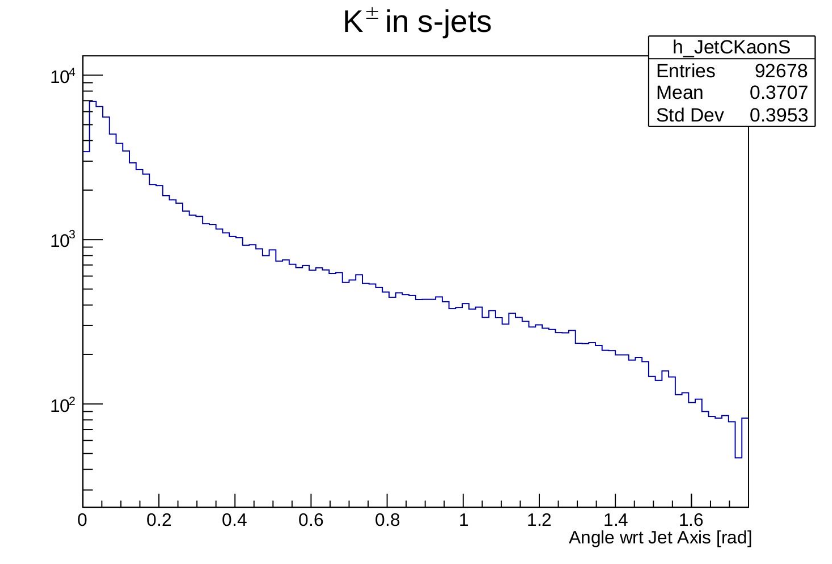


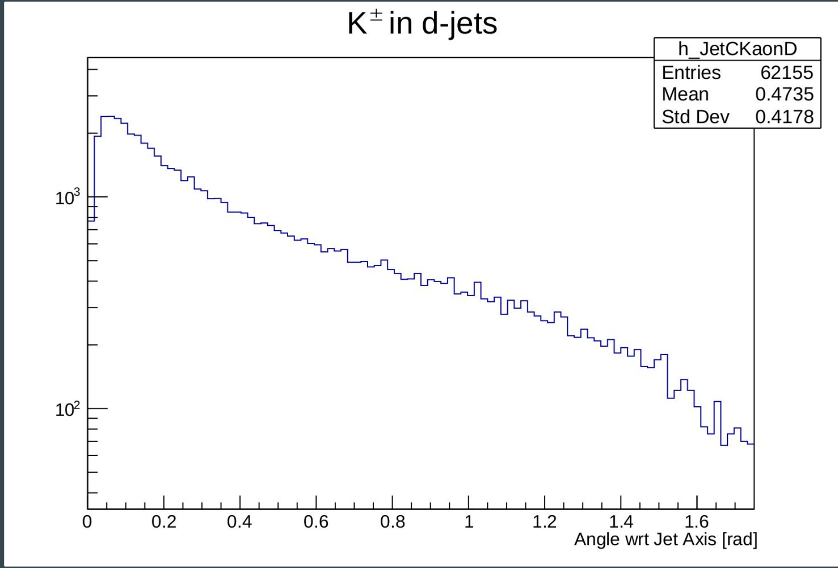


Distribution of jet constituents in jets from 100,000 Z⟶uds events, i.e. 200,000 jets

- Jet constituents have following basic cuts applied: pT> 500 MeV and |cos(θ)| < 0.97 (~=14°)
- s-jets tend to have more kaons, while the d-jets tend to have more pions.
- The most simple s-tagger would exploit the differences in multiplicity of jet constituents to distinguish these jets.

Jet Clustering: Angular Distribution of Constituents



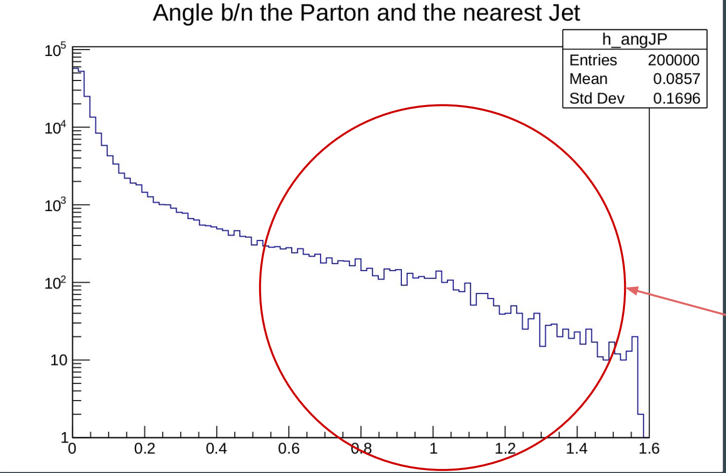


- Another potential distinguishing feature can be the angular distribution of these jet constituents around the jet axis.
- Kaons in the s-jets tend to be closer to the jet axis compared to those in d-jets.

Jet Clustering: MC truth Jet Flavour Assignment

FCCAnalyses: JetTaggingUtils.cc:get_flavour()

```
Float_t dot = p.px()*parton.momentum.x+p.py()*parton.momentum.y+p.pz()*parton.momentum.z;
Float_t lenSq1 = p.px()*p.px()+p.py()*p.py()+p.pz()*p.pz();
Float_t lenSq2 = parton.momentum.x*parton.momentum.x+parton.momentum.y*parton.momentum.y+parton.momentum.z*parton.momentum.z;
Float_t norm = sqrt(lenSq1*lenSq2);
Float_t angle = acos(dot/norm);
if (angle <= 0.3) result[j] = std::max(result[j], std::abs ( parton.PDG ));
```



Flavour assignment seems derived for cone(-style) algorithms, not clear how appropriate for the kT jets used here

These uds jets will either be inaccurately tagged or stay untagged with the current definition of flavour assignment.

Jet Clustering: MC truth Jet Flavour Assignment

	FCCAnalyses	MC Truth
s	61,732	72,044
d	56,872	72,016
u	46,840	55,940
c	1,006	-
b	94	-
untagged	33,456	-
In a sample of 100,000 $Z \longrightarrow$ uds events, i.e. 200,000 uds jets		

This is a potential area for improvement in FCCAnalyses, since with the present definition of flavour assignment, a significant number of jets are not assigned a flavour and there are others which are mistagged.

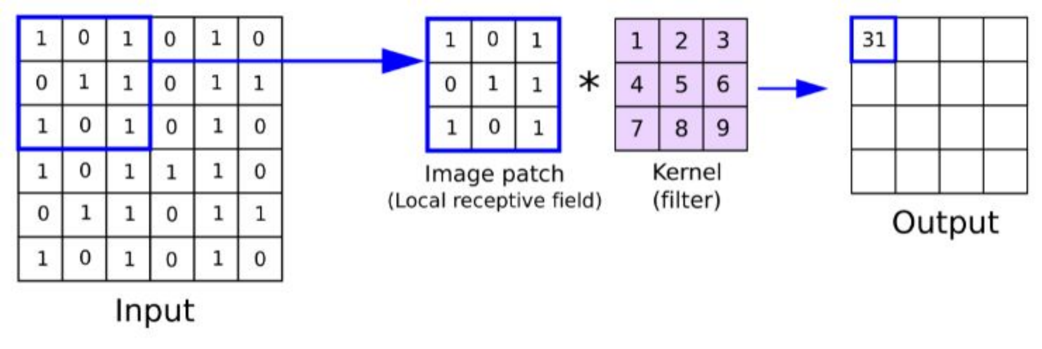
Clustering the jets inclusively may reduce this effect. Though it hasn't been checked yet.

Jet Tagging with a CNN

Convolutional Neural Networks: Basics behind Image Recognition

Convolutional Layer

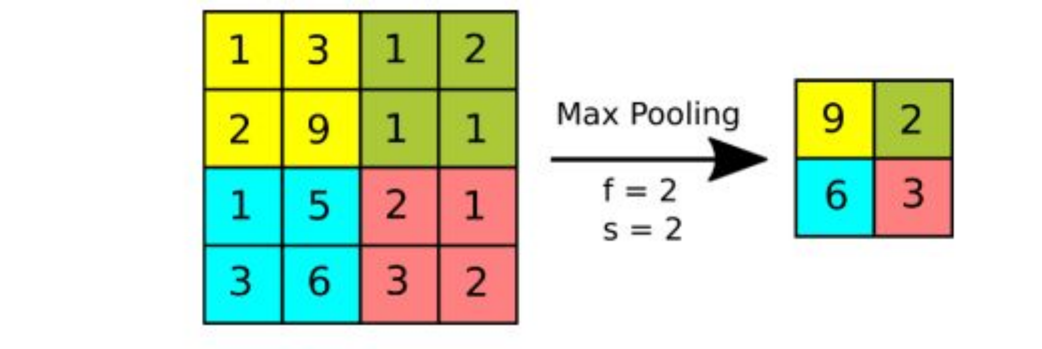
- Slide the filter across the image, one column at a time; when at the end of the row, move down a row



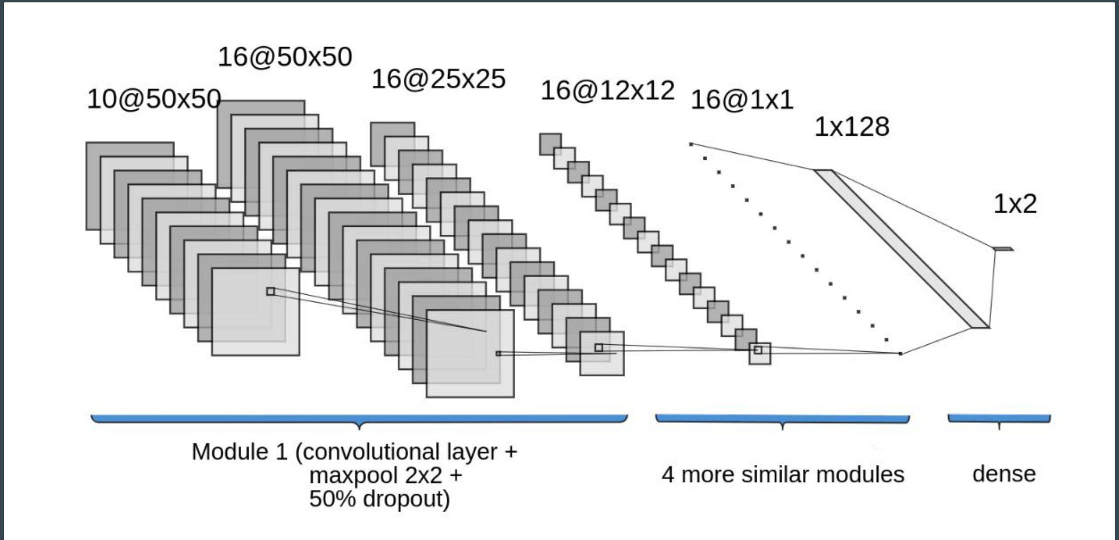
*https://anhreynolds.com/blogs/cnn.html

Max Pooling Layer

- Reduce sub-matrices to a single value - the highest value within the sub-matrix



*https://anhreynolds.com/blogs/cnn.html



*Lode Vanhecke's Bachelor Thesis at VUB

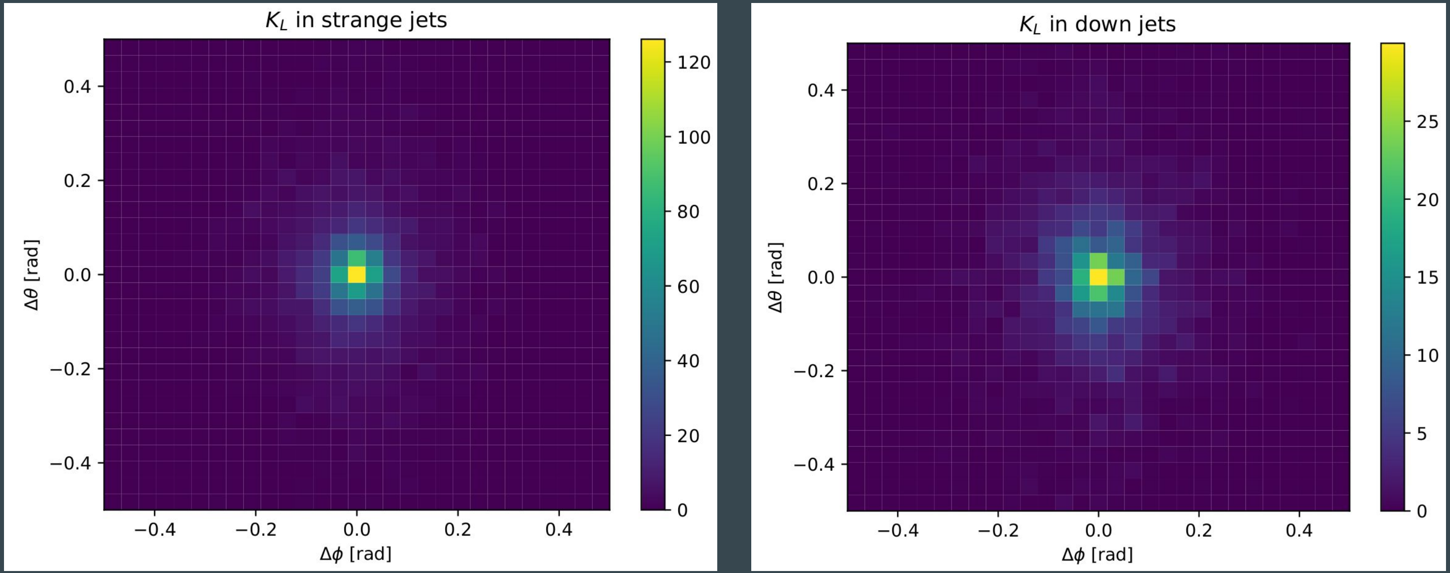
Strategy:

- 2D histograms of uds jet constituents in the θ-ϕ space centered around the jet-axis
- Pass through a CNN classifier
- Choose working points with required fake rates

Network Structure:

- Similar structure as above, except the image resolution changed from 50x50 pixels to 29x29 pixels and one fewer module
  - 10 inputs
  - 4 modules with a convolutional layer (16 filters), a maxpool layer, and a 50% dropout layer
  - Flattened to a dense layer
  - Fully connected to the output layer (3 nodes) with softmax activation function

Jet Images



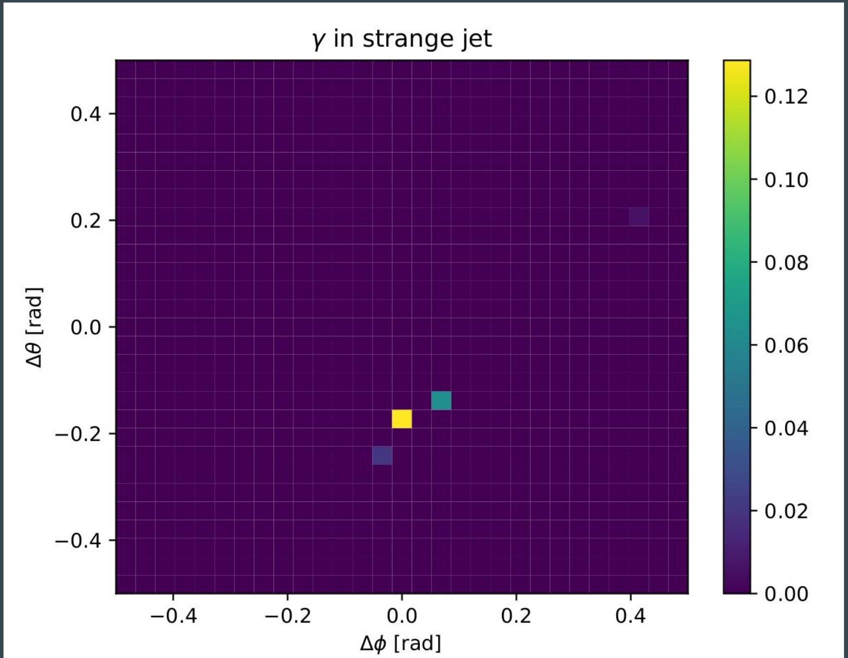
- Images are made for different constituent types (next slide)
- θ-ϕ distribution is weighted by a normalisation factor = |p| (constituent)/ |p| (jet)
    - To make the tagger less dependent on momentum (we still need to check if this is true)

CNN Input

Jet Images from 10 Categories (inspired by assumptions of particle ID, no fake rate etc included):

1. $K^{\pm}$
2. $\pi^{\pm}$
3. K_L
4. $e^{\pm}$
5. $\mu^{\pm}$
6. γ
7. p
8. n
9. $K_s \rightarrow \pi^+\pi^-$
10. $\pi^0 \rightarrow \gamma\gamma$

Each image is 29x29 pixels, encompassing the range (-0.5, 0.5) radians in the θ-ϕ space centered around the jet axis.



CNN Training

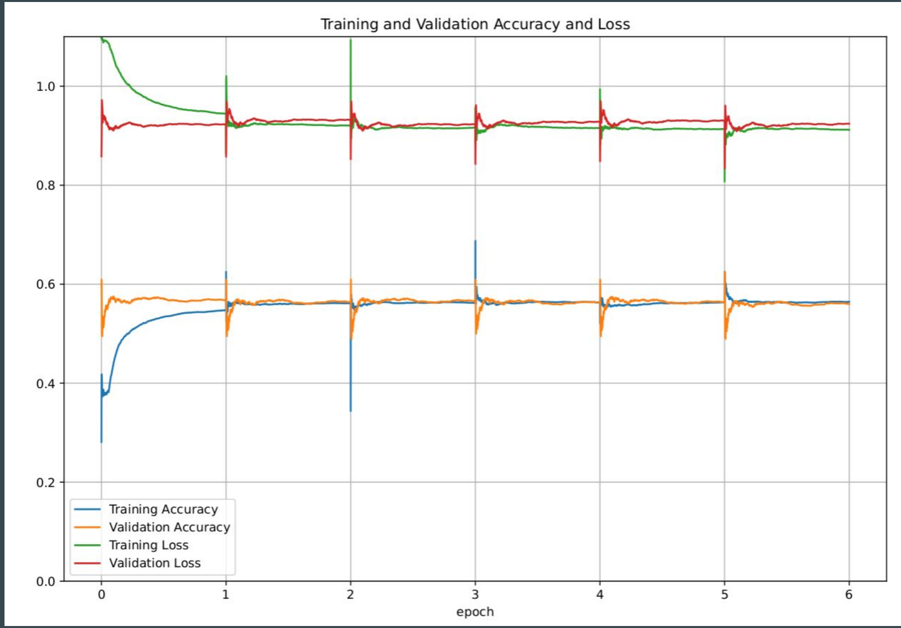
- Trained the CNN with 100,000 di-jet $Z \rightarrow uds$ events, i.e. 200,000 jets.
- Implemented in Tensorflow
- ~14k trainable parameters (lightweight)

  -> No obvious overfitting/overtraining

- Categorical cross entropy as loss function

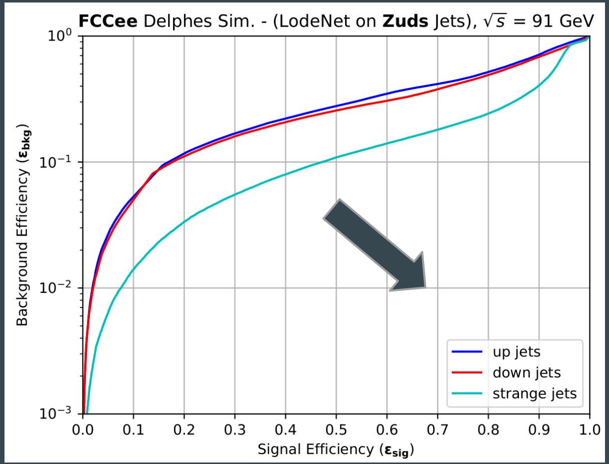
$$L(\mathbf{y}, \mathbf{p}) = -\Sigma_i^C y_i log(p_i)$$

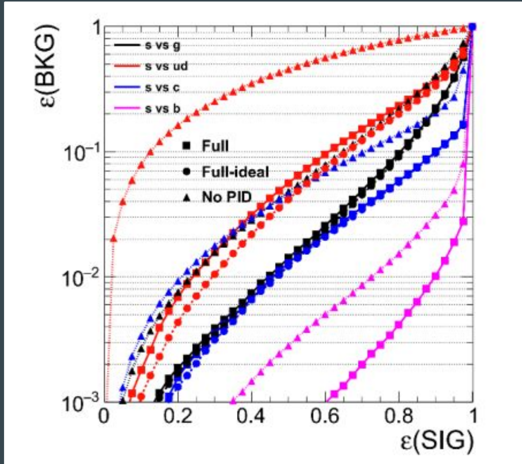
(**y** is true class label vector, and **p** is network prediction vector)



Performance

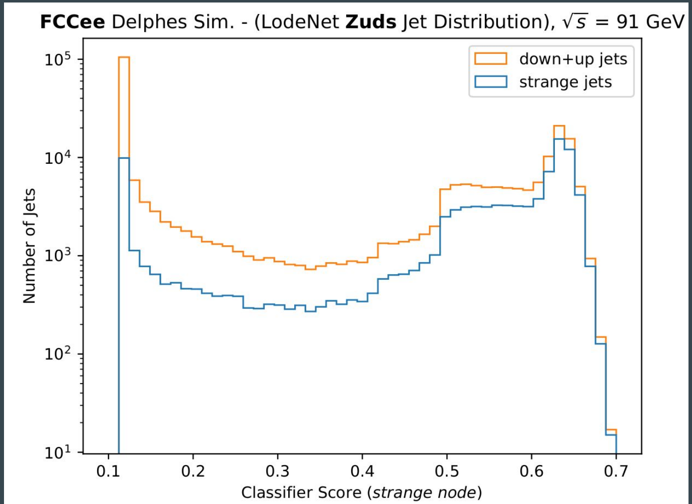
While testing the model on 200,000 events





*compare to Michele Selvaggi (TOP2021)

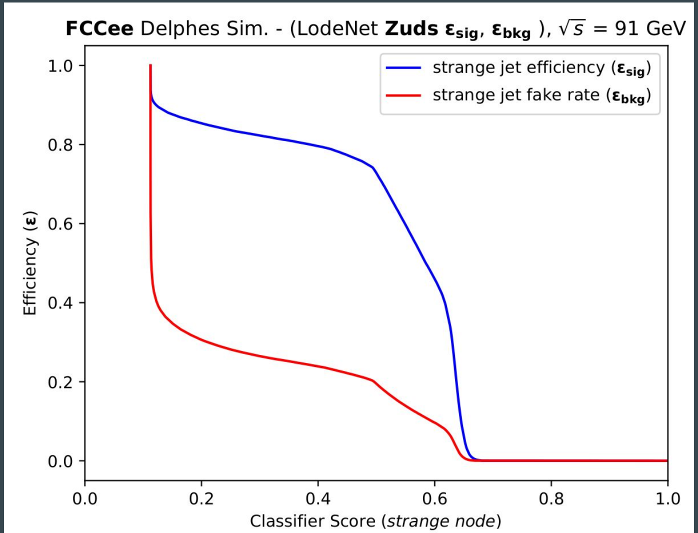
Performance at 10% fake rate is ~50%



Classifier score distribution that leads to these ROC curves

Working points

- Three working points defined at fake rates of 10%, 5%, and 1%, respectively



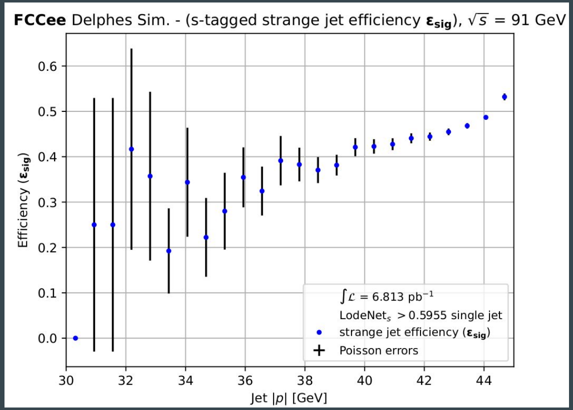
	10% fake rate	5% fake rate	1% fake rate
Classifier Score (strange node)	0.5955	0.6320	0.6480
Signal Efficiency	47.2%	27.7%	7.5%

Signal Efficiency vs |p|

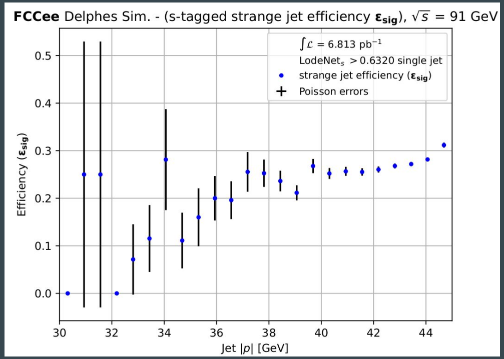
Increase in signal efficiency wrt |p| in 10% and 5% working points

Little improvement in 1% working point

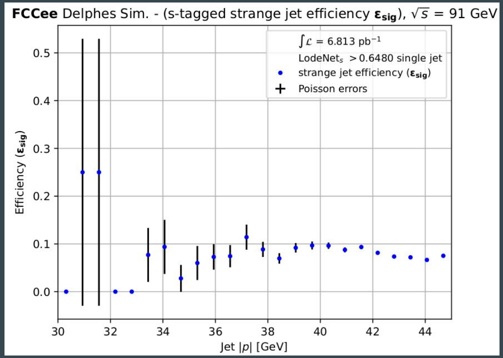
10%



5%



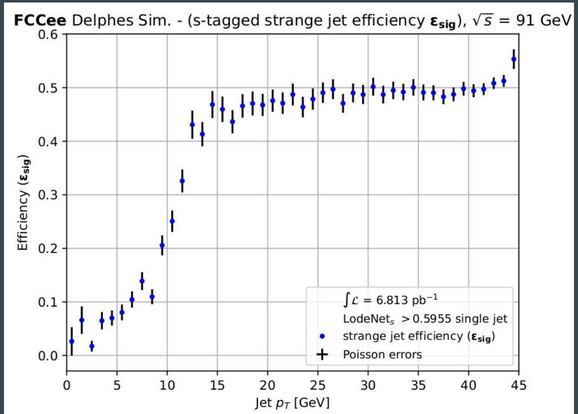
1%



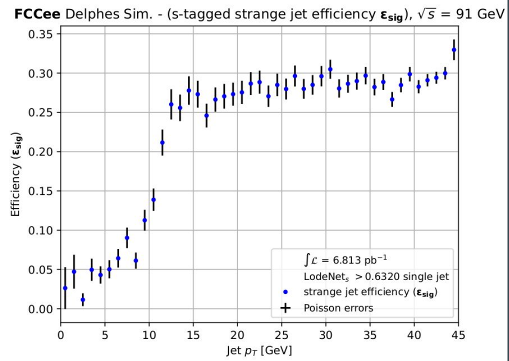
Signal Efficiency vs pT

Increase in signal efficiency up to a threshold visible at all working points

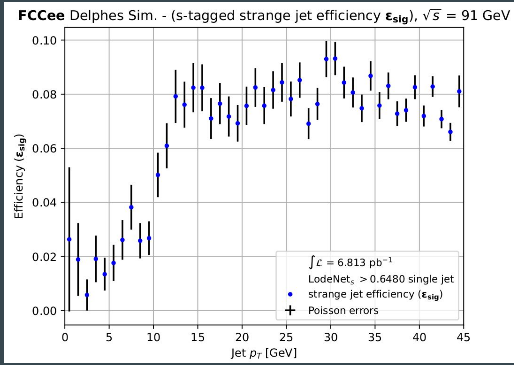
10%



5%



1%

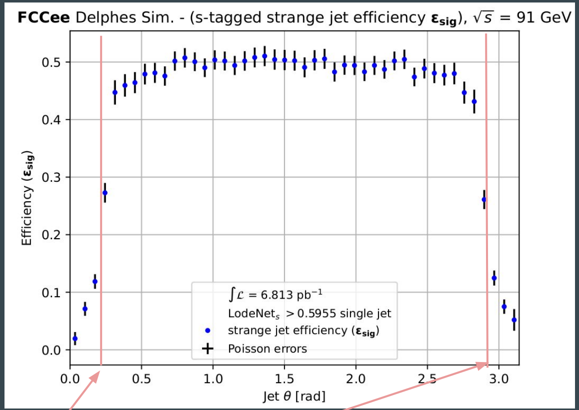


Tagging below 10 GeV jet pT might challenging

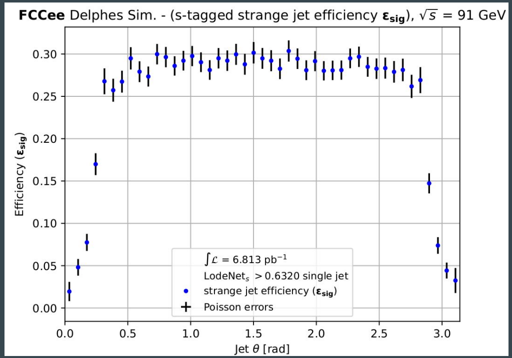
Signal Efficiency vs θ

No signal efficiency dependence on the polar angle, up to forward/backward jets

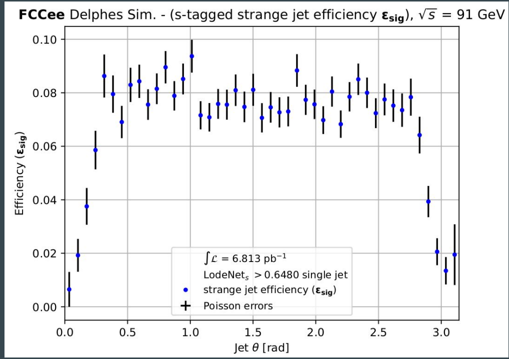
10%



5%



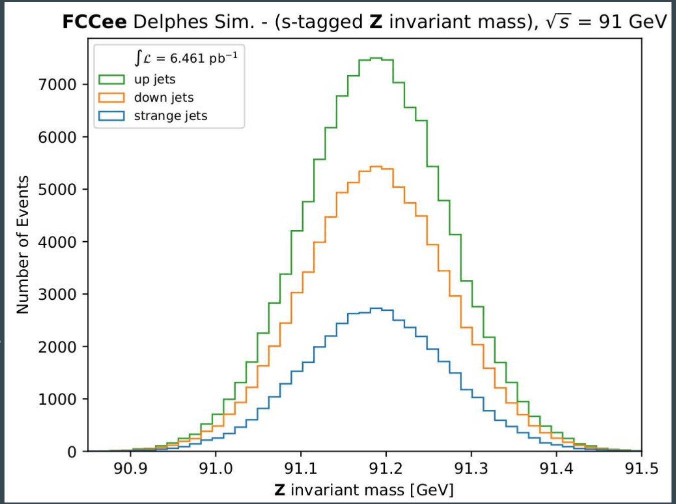
1%



Drop in efficiency consistent with IDEA acceptance as cuts were introduced on particles before making the jet images

Z Peak Reconstruction

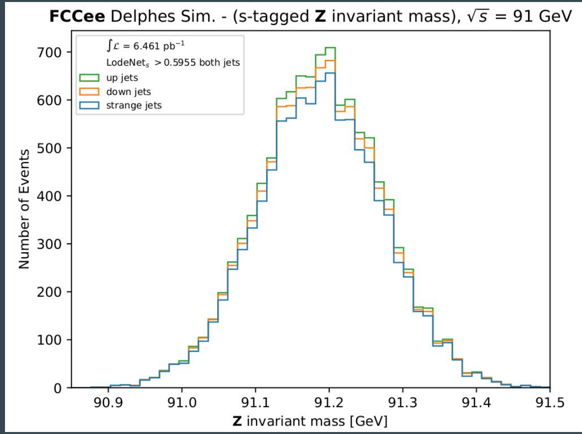
- Z peak before tagging vs after tagging both jets
- Ignored background and b/c quark decays (for now)

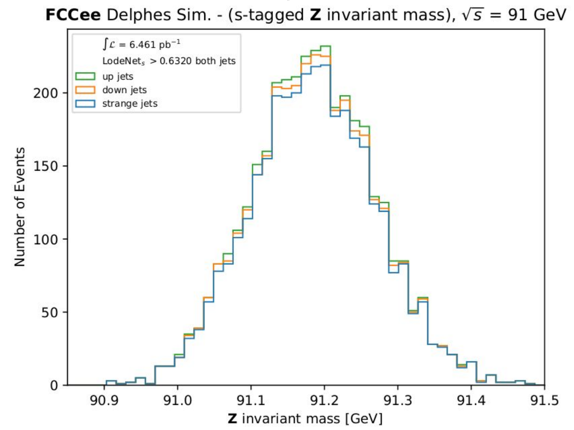


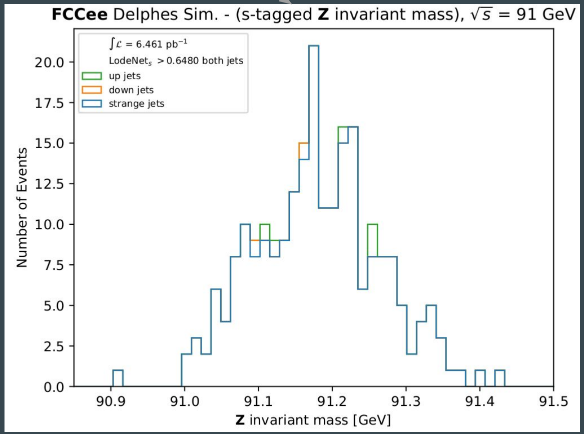
10%

5%

1%







Summary

- First look at a CNN based s-tagger on Spring2021 IDEA event samples
  - Studied jet constituent multiplicity and angular distribution around the jet axis as potential distinguishing variables
  - Implemented a CNN model in Tensorflow and trained with jet images from 100,000 events
  - Evaluated this trained network on the jets from 200,000 events to review its performance
  - 3 working points at fake rates of 10%, 5%, and 1%, with signal efficiencies of 47%, 28%, and 8% respectively
- Performed in FCCAnalyses, in parallel with stand-alone ROOT and coffea
- Study performed to familiarise ourselves with FCC software and samples

Potential Improvements

- Study the contribution of each category in the classifier
- Background samples not included
- Retrain with reconstructed particles
- Introduce Kinematic Cuts on input information

Outlook

- K. Gautam and E. Plörer are full-time on FCC-ee
- Looking forward to contributing to FCC-ee centrally coordinated tagging effort
- Looking for input on potential to include CNNs in flavour tagging if complementary.
  - Suggestions?