AFB studies at 500 GeV

LCFI+ Flavour Tag Optimization

ILD Top/HF group meeting
3/02/23

Jesús P. Márquez Hernández

January updates

1. Particle Swarm Optimization: Final results.
   - ROC plots, ROC Integrals and efficiency-purity plots.

2. Introducing dEdx to LCFI+.
   - Implementation finished.
     - Need to be reviewed.

3. Re-training of flavour tag weights using dEdx.
   - In *preliminary phase*.

Particle Swarm Optimization

PSO – Performance plots

- On the next slides:
  - Plots for b-tag and c-tag:
    - ROC, considering the desired flavour as signal and the others as background.
      - Also approximated AUC (ROC Integral) values to compare
    - Purity vs Efficiency.
  - Luminosities:

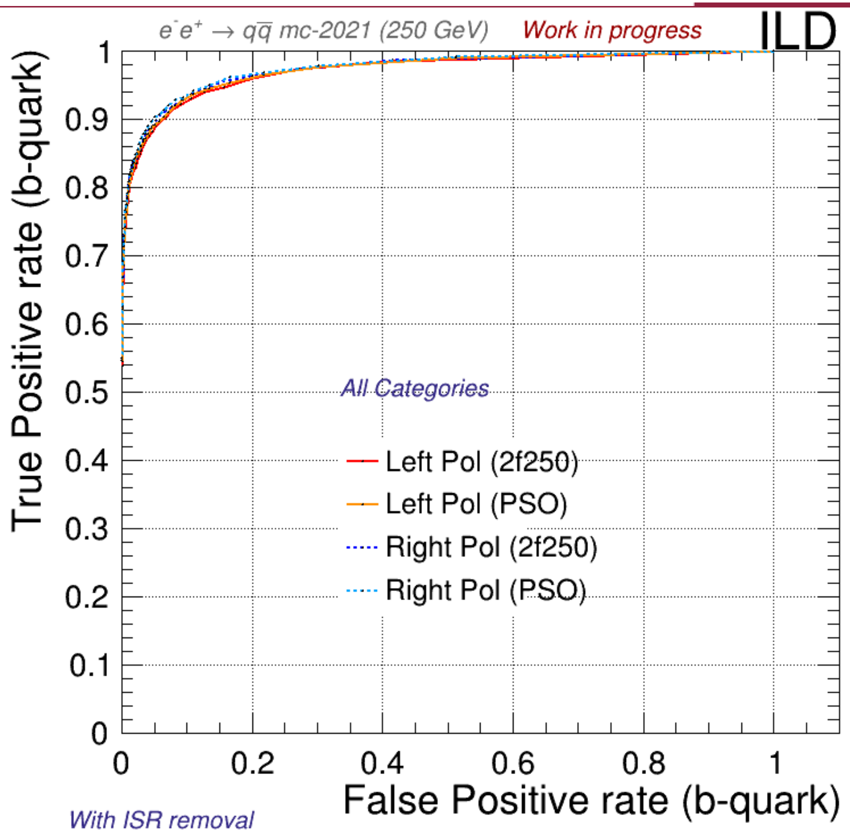
PSO
PSO (pol. test)

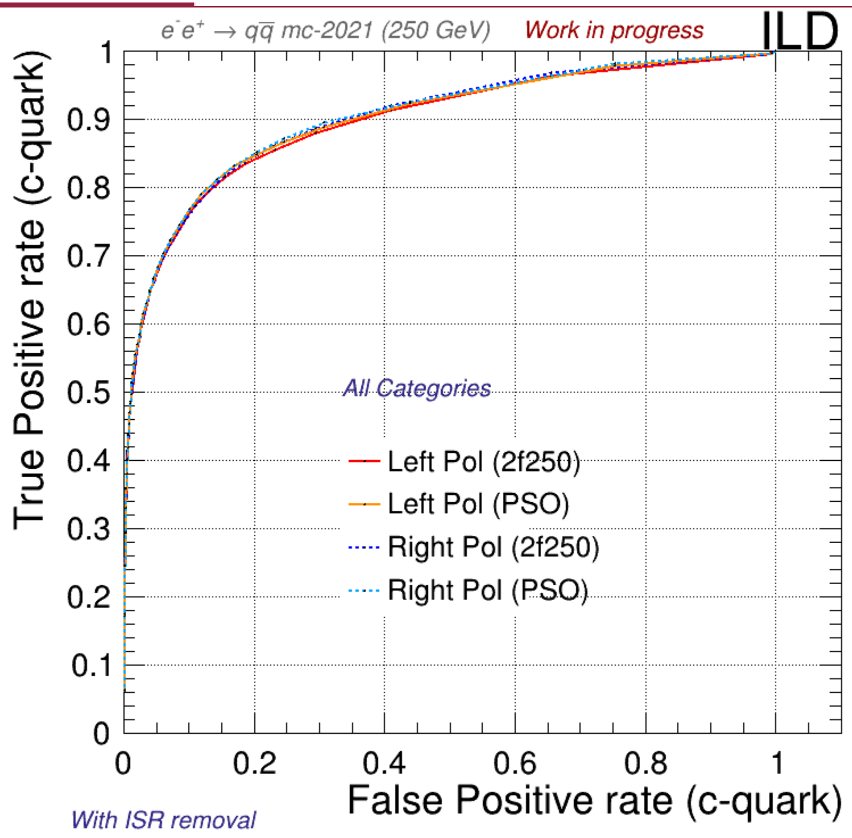
	Test + Training (50/50)	Sample for plots / Independent sample for plots
250 GeV, $e^-_L e^+_R$	20 fb^{-1} (2.5 M events)	1.6 fb^{-1} (200 K events)
250 GeV, $e^+_R e^+_L$	36 fb^{-1} (2.5M events)	2.8 fb^{-1} (200 K events)
500 GeV, $e^-_L e^+_R$	250 fb^{-1} (1.2M events)	1430 fb^{-1} (7.0 M events)
500 GeV, $e^+_R e^+_L$	480 fb^{-1} (1.5M events)	1060 fb^{-1} (3.2 M events)

250 GeV: Test + Training (50/50) | Sample for plots

500 GeV: Test + Training (50/50) | Independent sample for plots

PSO Performance (250 GeV)

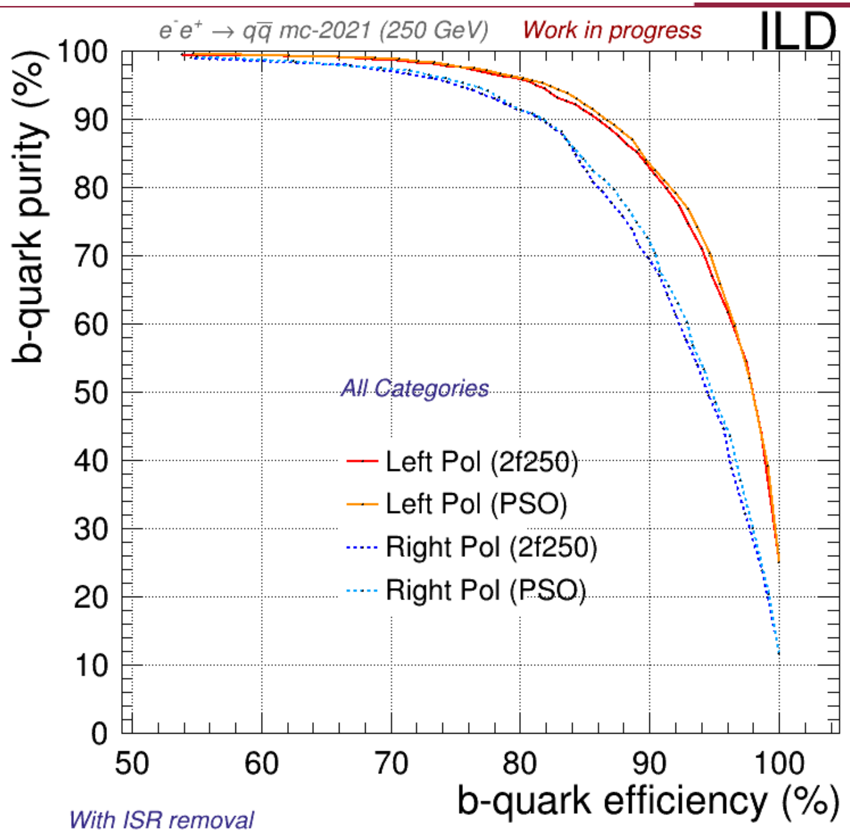


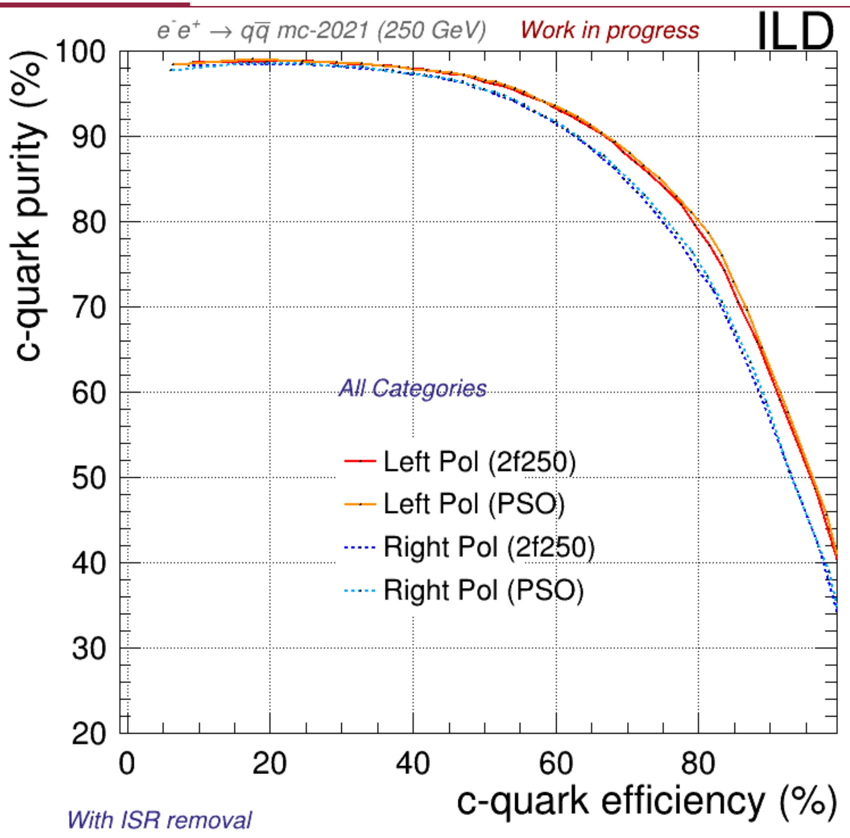


[Integrals 2f250] e^-_L: 0.971 | e^-_R: 0.973
[Integrals PSO] e^-_L: 0.973 | e^-_R: 0.976

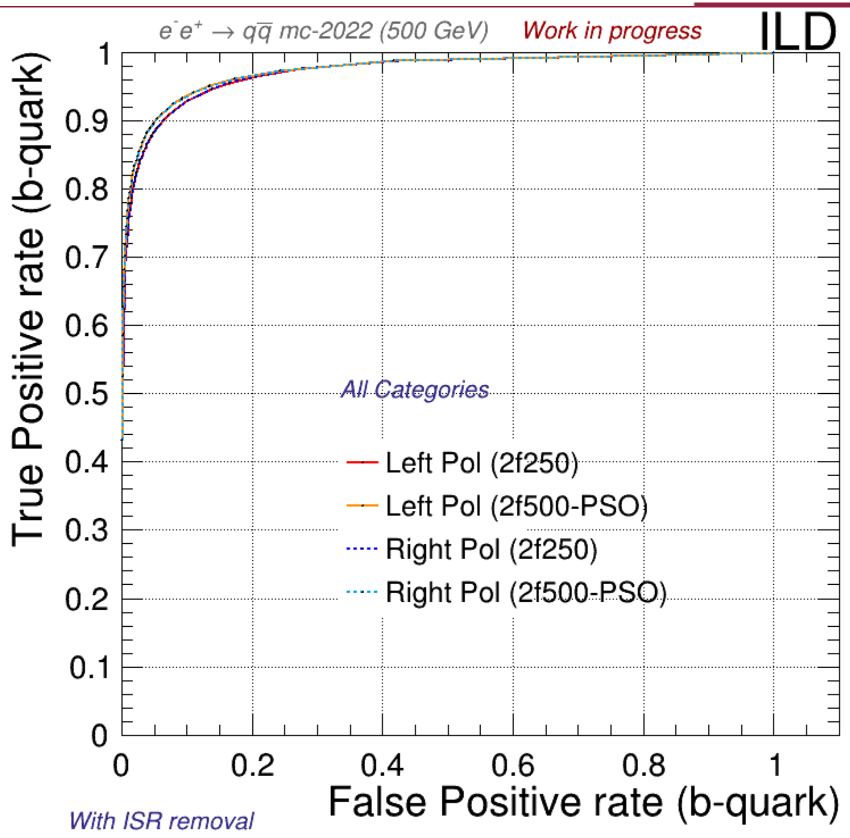
[Integrals 2f250] e^-_L: 0.898 | e^-_R: 0.902
[Integrals PSO] e^-_L: 0.901 | e^-_R: 0.904

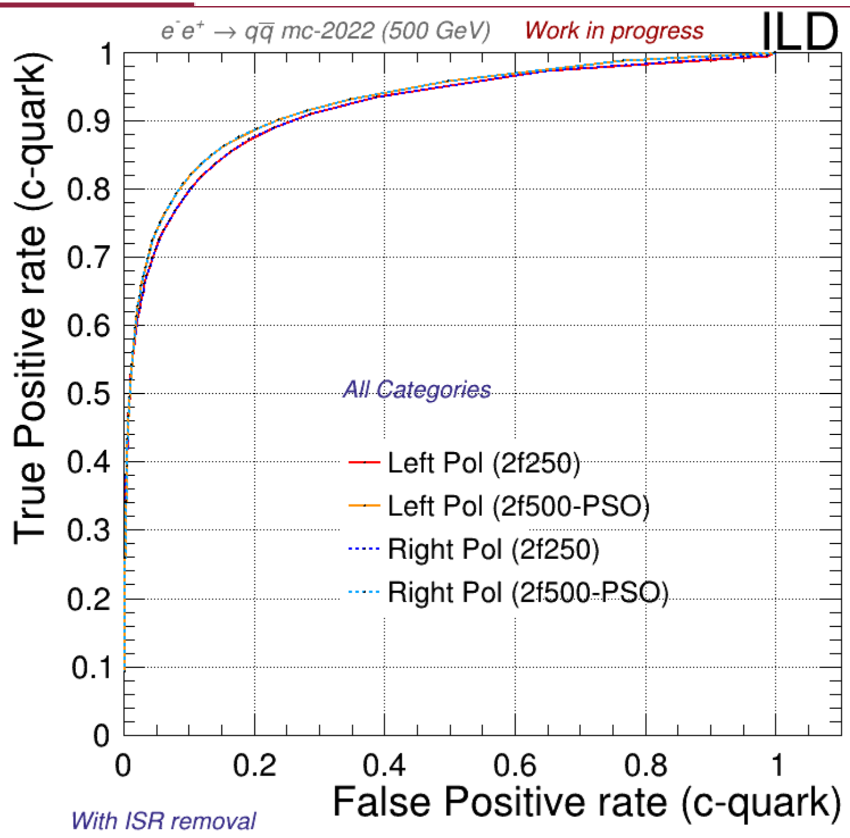
PSO Performance (250 GeV)





PSO Performance (500 GeV)

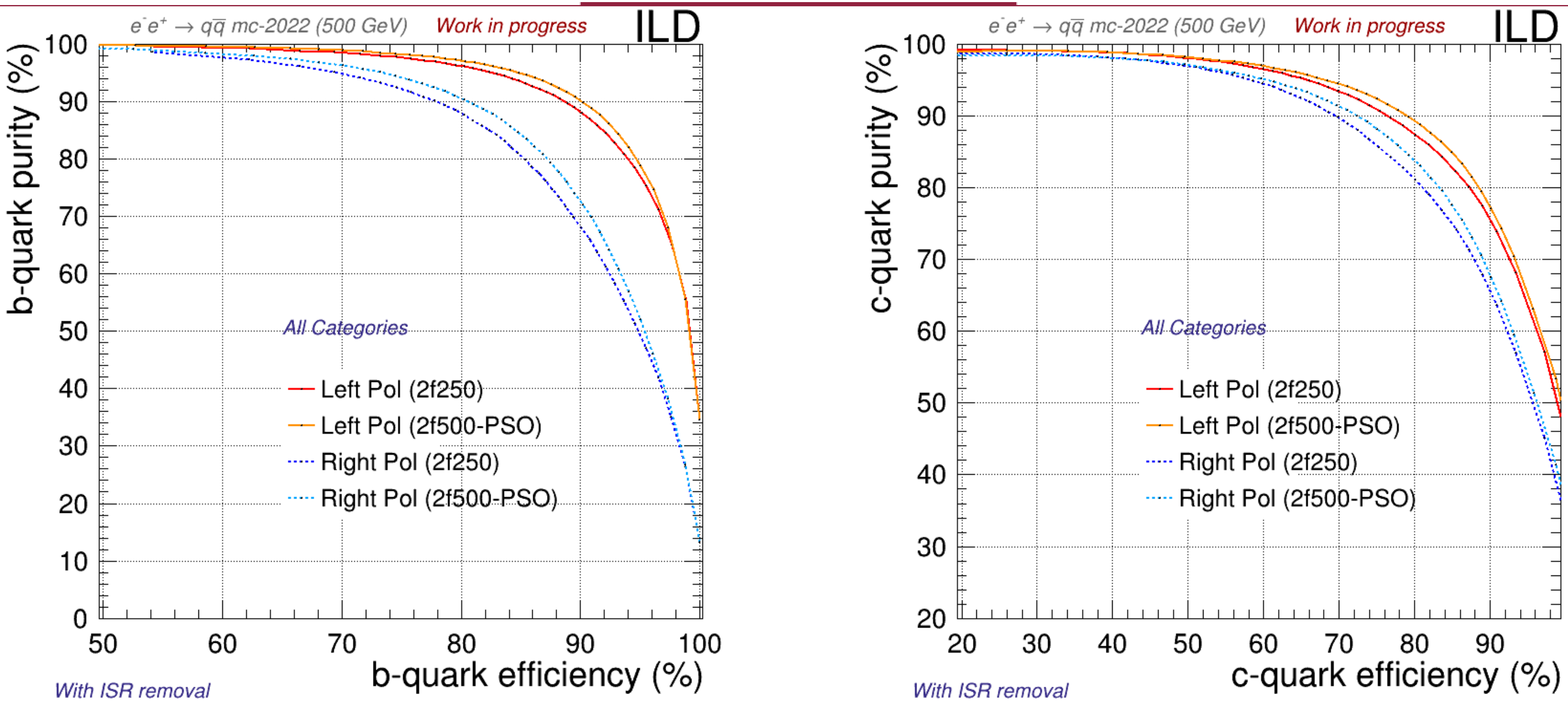




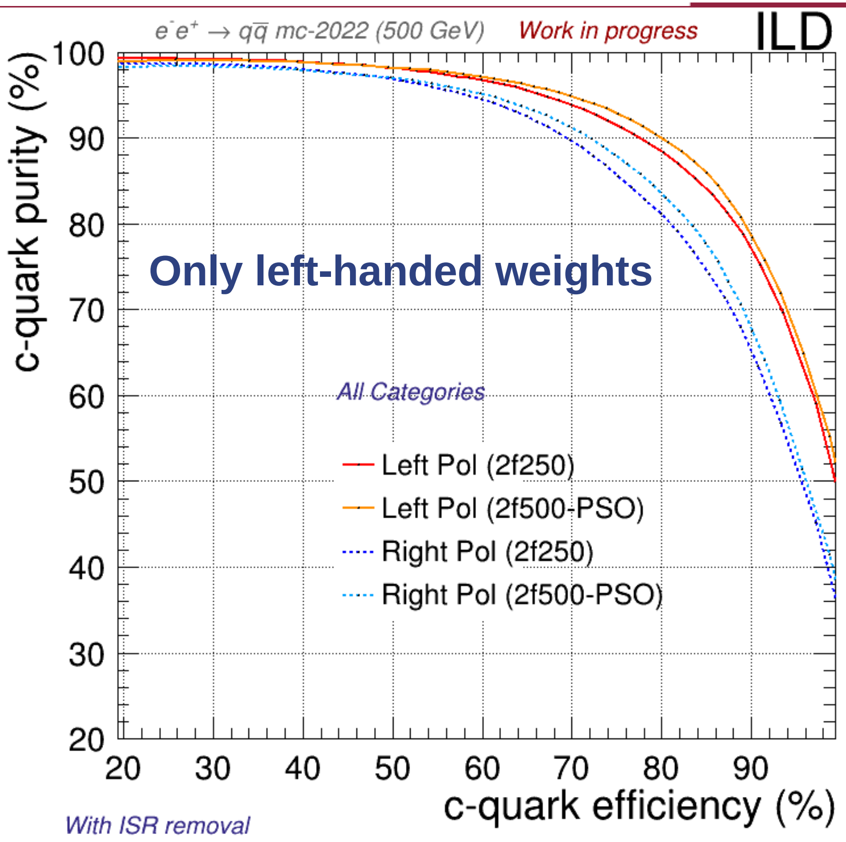
[Integrals 2f250] e^-_L: 0.972 | e^-_R: 0.971
[Integrals PSO] e^-_L: 0.974 | e^-_R: 0.974

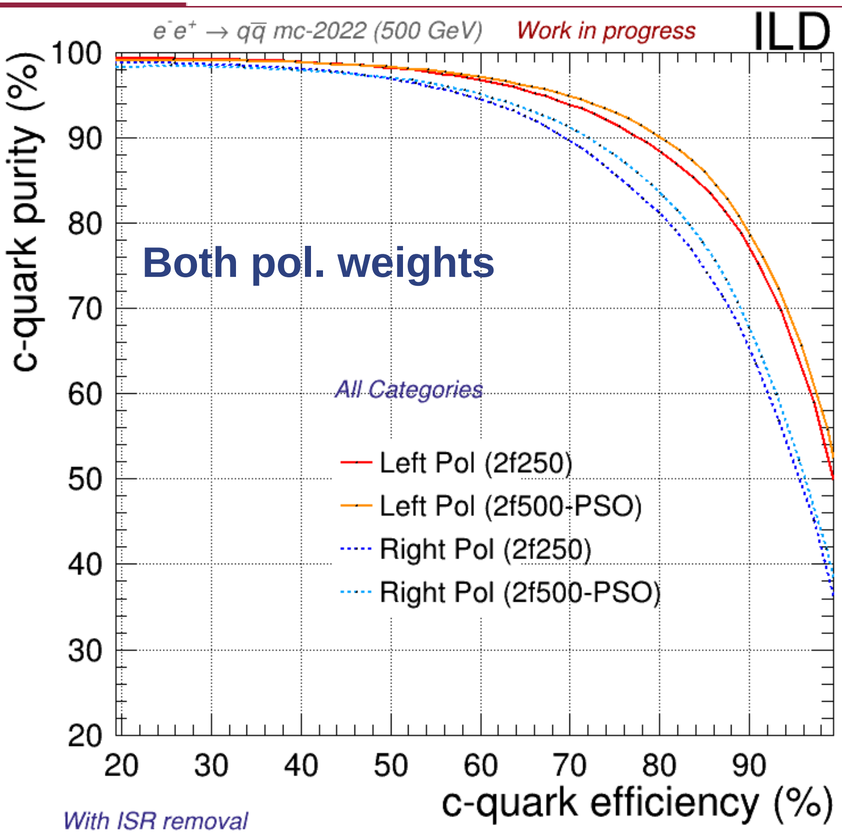
[Integrals 2f250] e^-_L: 0.917 | e^-_R: 0.917
[Integrals PSO] e^-_L: 0.925 | e^-_R: 0.925

PSO Performance (500 GeV)



Polarization dependence (500 GeV)





Minimal impact when training with sets of specific polarization

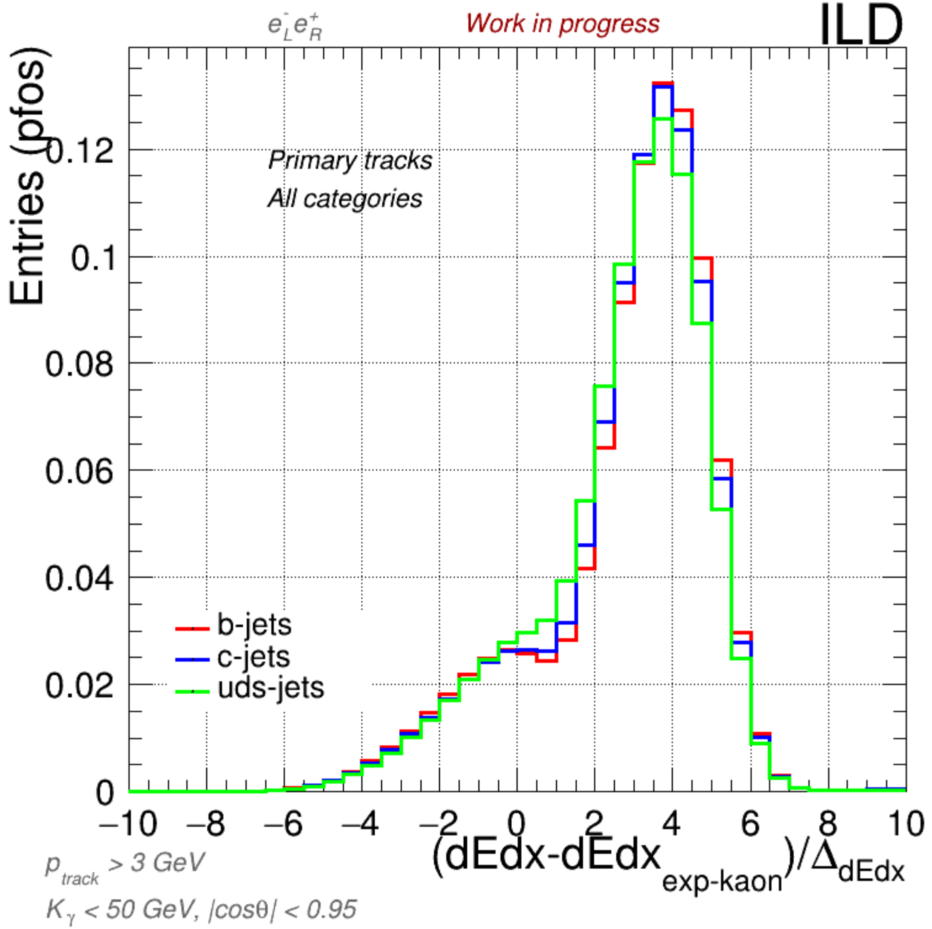
PSO Performance - Conclusions

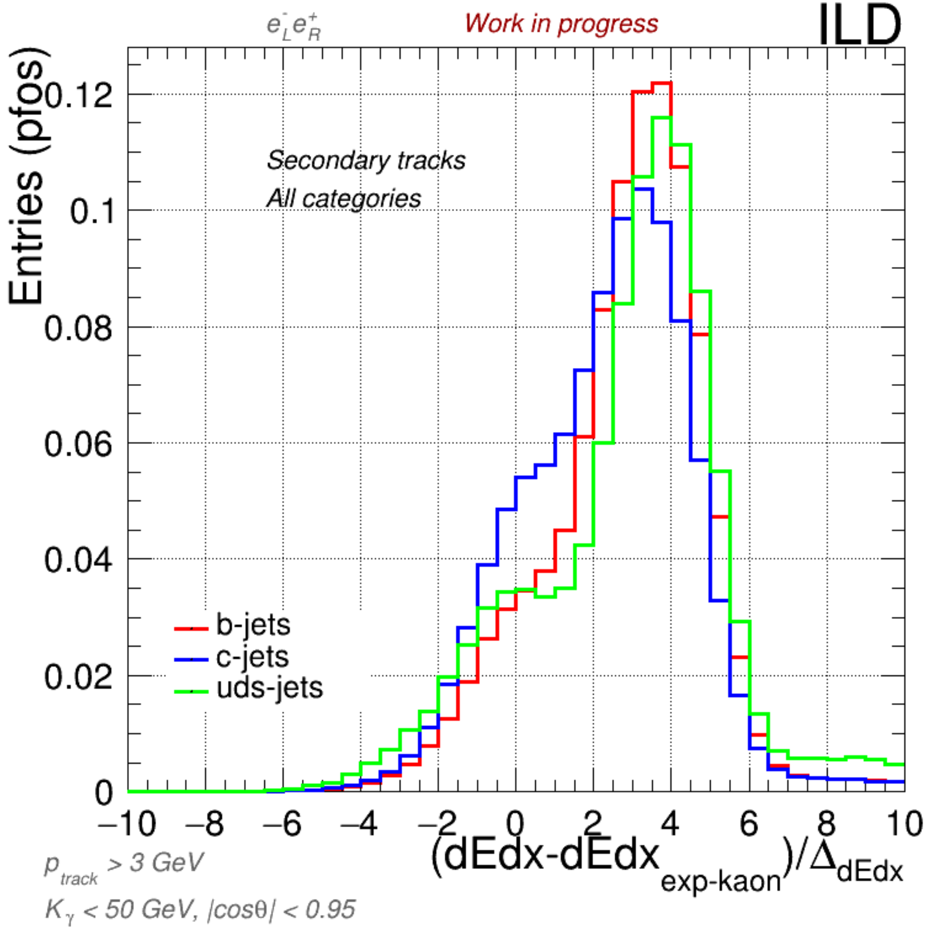
- We can notice:
  - An **improvement** in b-tag and c-tag.
    - At 250 GeV and 500 GeV.
  - The impact in efficiency if we train for different polarizations is **negligible**.

- Other observations (*behind the scenes*):
  - The new weights produce more stable results:
    - Performed as good with the data used for training and the new and bigger samples used for the final results.

Introducing dEdx to LCFI+

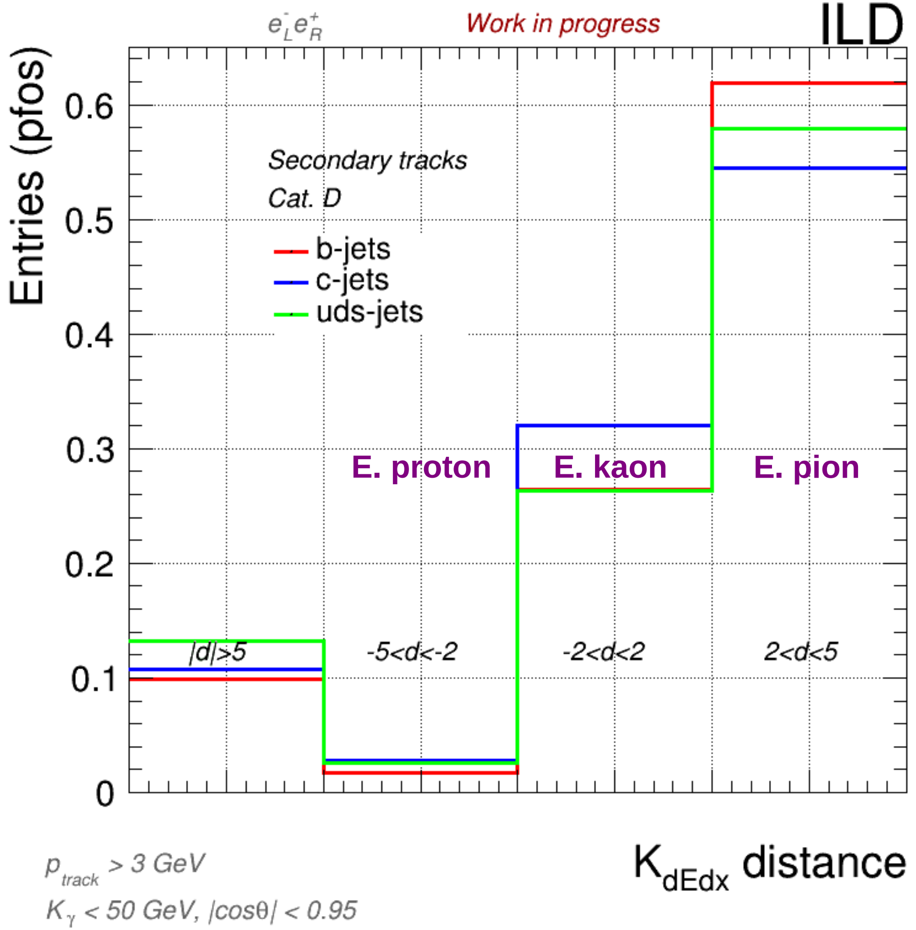
dEdx – KDS for different quark flavours

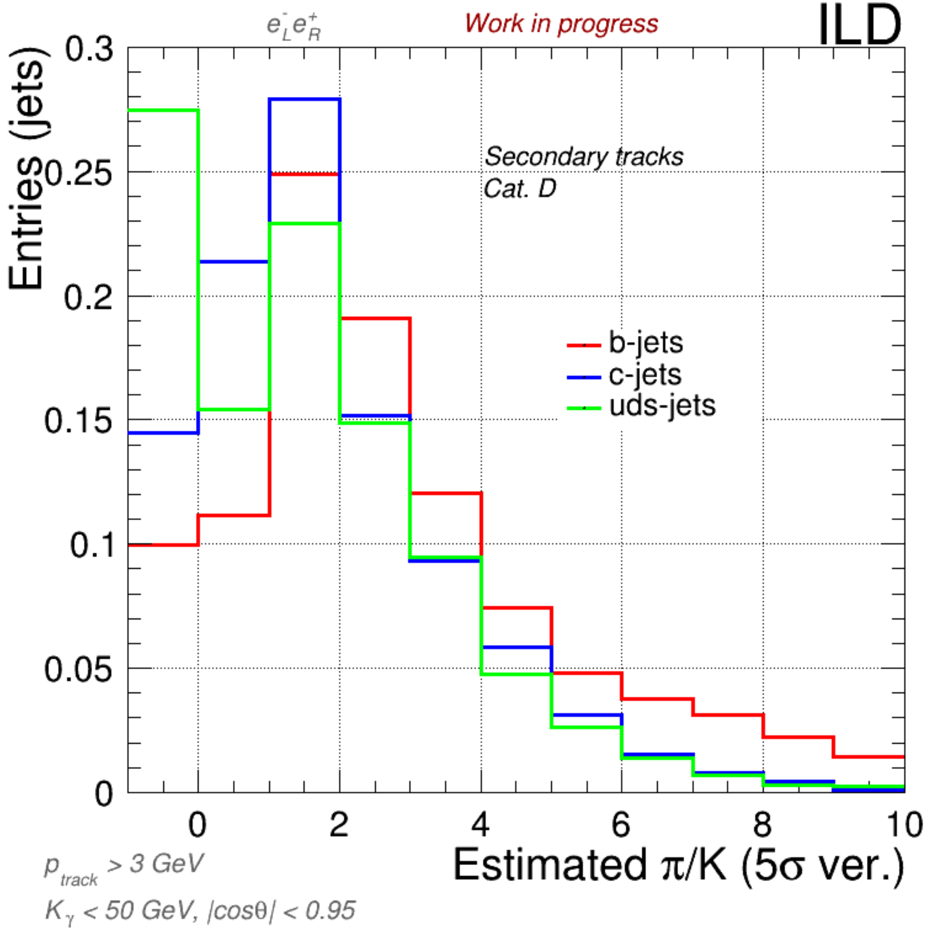




We classify the pfos depending on the Kaon Distance Significance (loaded in our .lcio files)

dEdx – KDS classification comparison





We build estimated π/K, π/p and K/p. If the denominator is 0, we set the ratio to -1.

Re-training flavor tagging (coding)

TrackNtuple.cc+TrackProb.C ← Prepare track info: D0/Z0/vertexing tolerance, etc.

↓

MakeNtuple.cc ← Creates the Ntuples used for the training process. (It needs the track info for many of them.)

↓

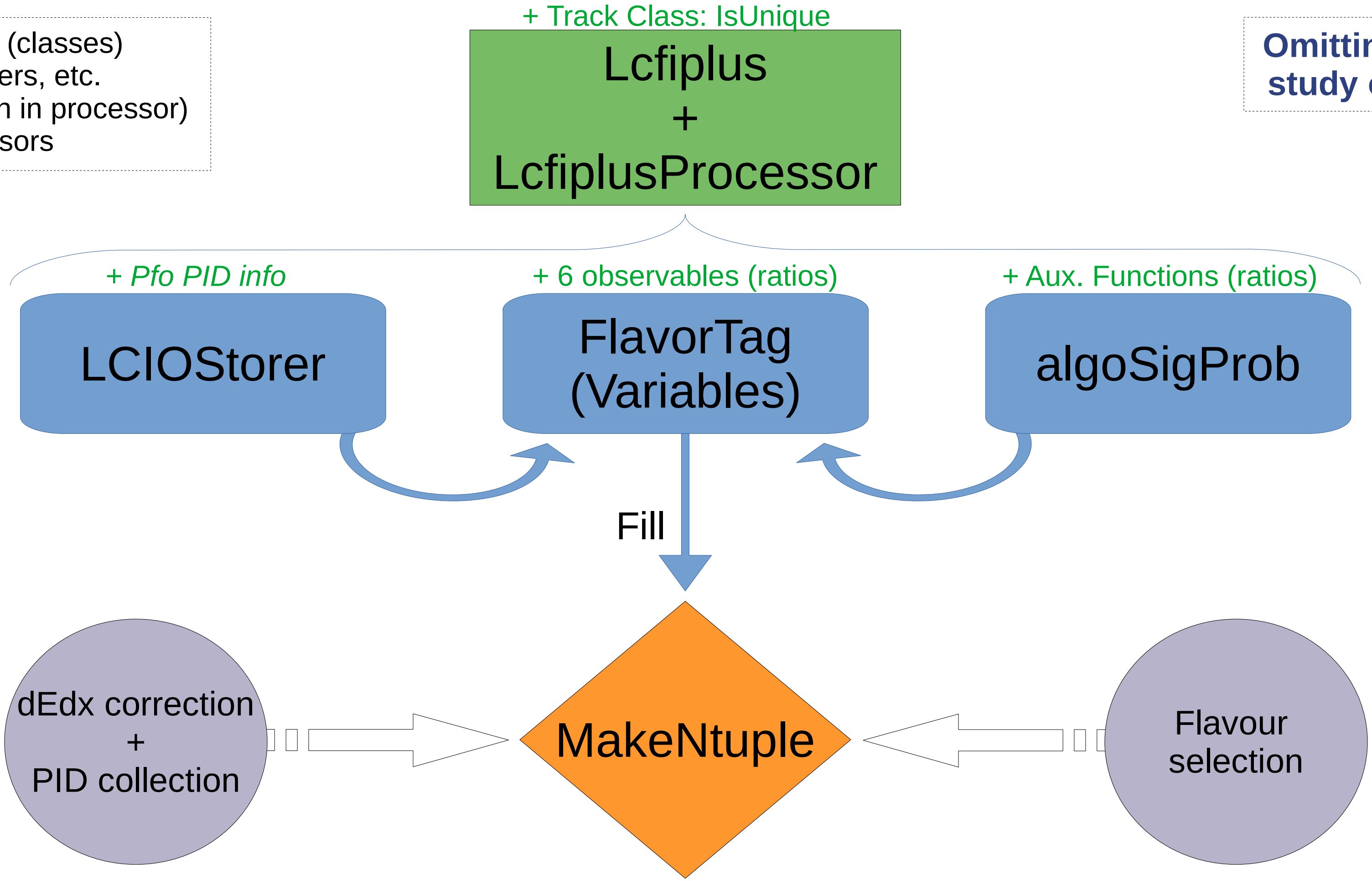
TrainMVA.cc ← Run the TMVA with the BDTs for each category. Creates the weight files for b/c-tag.

To introduce dEdx in the training process we need to load it into the Ntuples when we run MakeNtuple.cc

LCFI+ MakeNtuple Workflow (+dEdx)

Main definitions (classes)
Functions, readers, etc.
Algorithm (to run in processor)
External processors

Omitting parts I didn't study or interact with



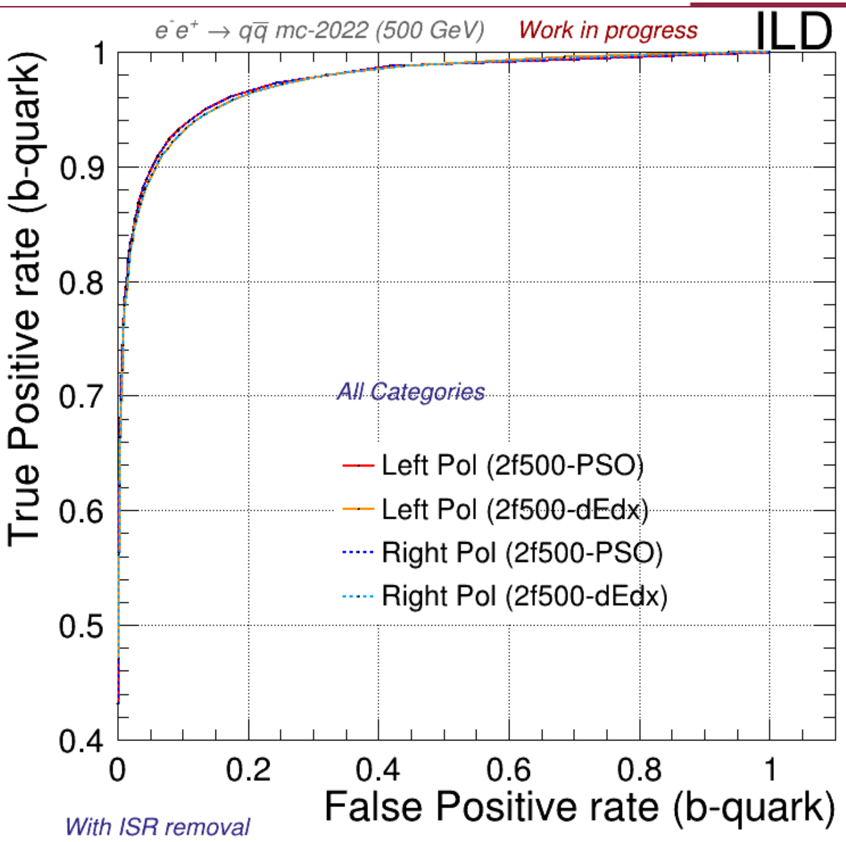
Retraining of flavour tagging using dEdx

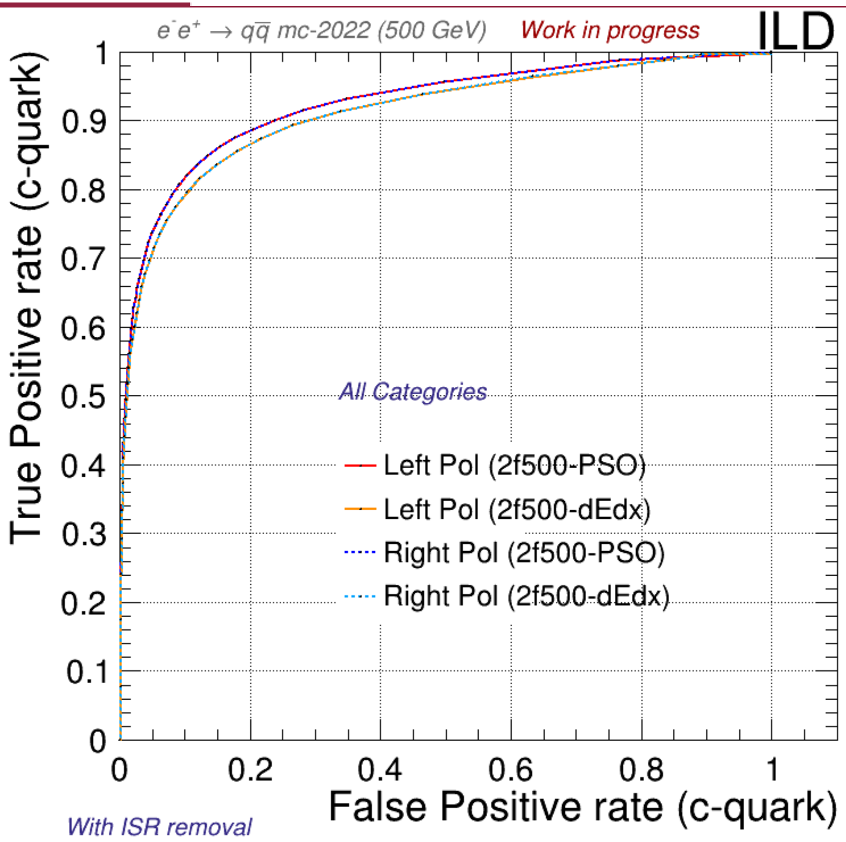
First test using dEdx

- I ran a first test using the same samples than the 500 GeV PSO studies.
  - Using the same configurations for the BDTs but with the 6 extra variables in the Ntuples.

- On the next slides:
  - ROC plots.
    - ROC Integral values.
  - Efficiency vs purity plots.

PSO vs dEdx Performance (500 GeV)

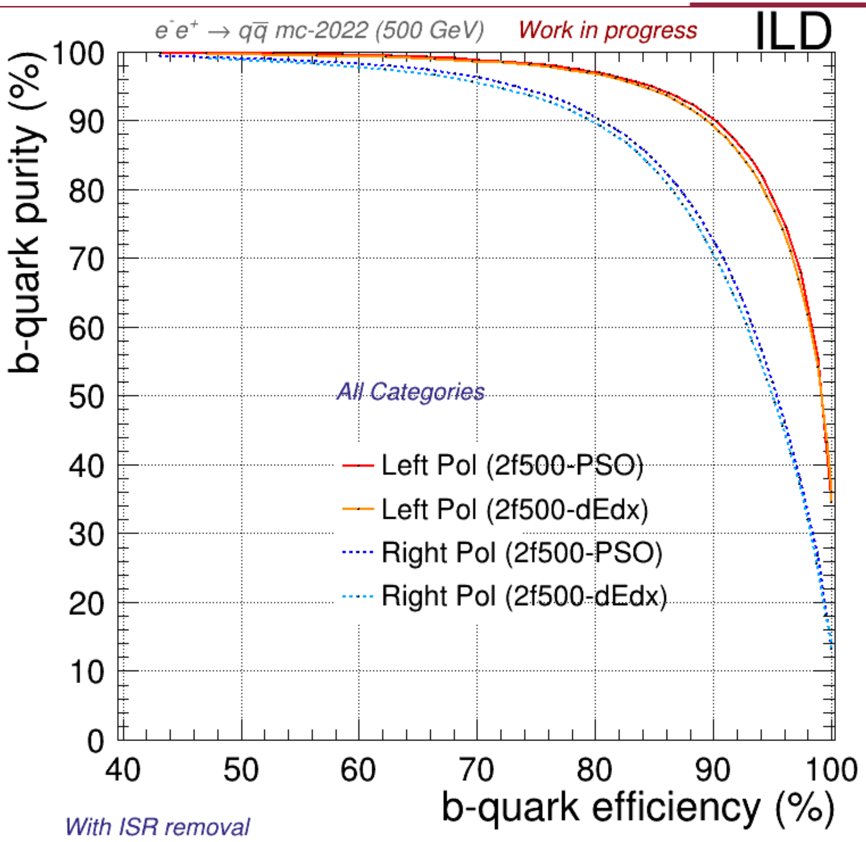


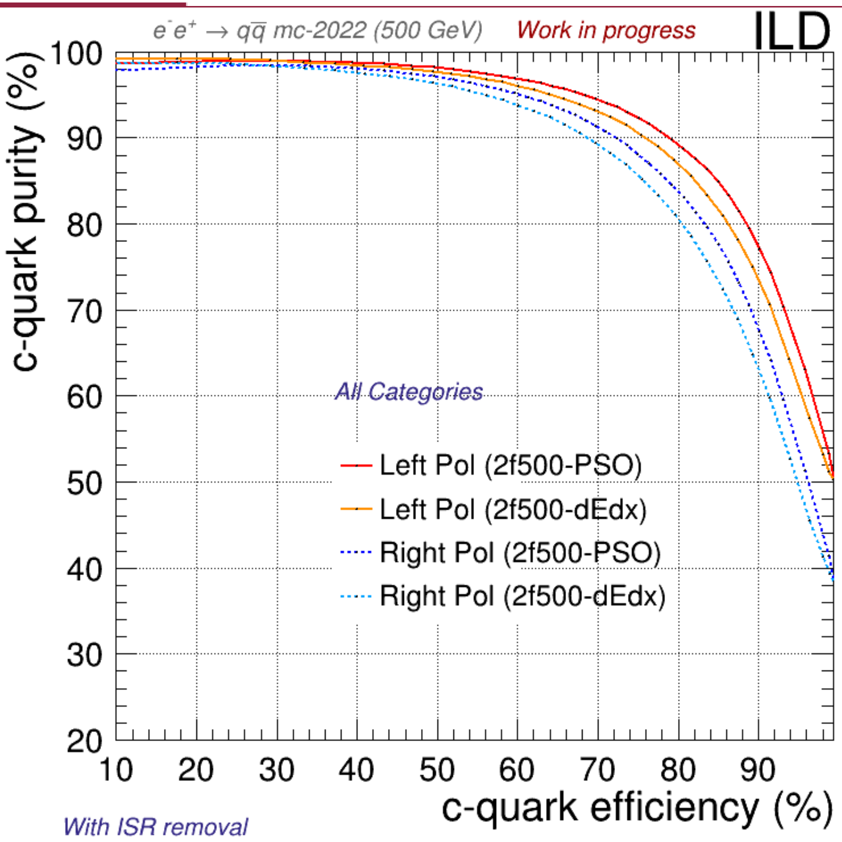


[Integrals PSO] e^-_L: 0.974 | e^-_R: 0.974
[Integrals dEdx] e^-_L: 0.973 | e^-_R: 0.973

[Integrals PSO] e^-_L: 0.925 | e^-_R: 0.925
[Integrals dEdx] e^-_L: 0.912 | e^-_R: 0.912

PSO vs dEdx Performance (500 GeV)





Now we have worst performance! :(

Summary & Prospects

- Now, dEdx can be implemented in LCFI+ and used for Flavour Tagging.
    - Needs to be reviewed.
- We have more information available but *worse performance*?!
    - Since we are using BDTs highly optimized (PSO) to a specific set of variables:
        - We were ***underfitting*** the new set.

- We have to fix the underfitting problem and check the real impact of dEdx.
    - Run the PSO again, using all variables.
        - Run for 250 GeV as well.
- After it, select the final set of weights.
    - Then, move to **physical studies** (R_q & A_{FB})!

Thanks for your attention

Back-up

Events for each category

With ISR removal

Z-Pole (LCFI+ paper[1])

Events (%)			
Cat.	b jets	c jets	uds jets
A	22.9	59.5	98.1
B	39.7	39.8	1.80
C	13.5	0.54	0.02
D	23.8	0.19	0.04

1. LCFIPlus: A Framework for Jet Analysis in Linear Collider Studies

250 GeV samples

Events (%)			
Cat.	b jets	c jets	uds jets
A	13.9	46.2	98.2
B	30.5	51.0	1.59
C	23.9	2.29	0.11
D	31.7	0.55	0.14

500 GeV samples

Events (%)			
Cat.	b jets	c jets	uds jets
A	11.2	35.8	96.7
B	28.6	58.3	2.64
C	22.9	4.65	0.26
D	37.3	1.27	0.42

Category	A	B	C	D
Number of vertices	0	1	1	2
Number of single-track pseudovertices	0-2	0	1	0

Boosted Decision Trees (BDT) - TMVA

- We are already working with these Gradient Boosted Decision trees using ROOT's Toolkit for MultiVariate data Analysis (TMVA). We use the following parameters:
  - BoostType=Grad.
  - NTrees.
  - Shrinkage.
  - UseBaggedBoost:BaggedSampleFraction.
    - Bagging: A new sampling is performed before each step (removes biases).
  - NCuts (binning used when sampling).
  - MaxDepth (Nº of leaves).

The Particle Swarm Algorithm optimizes the use of *these parameters*

We used all but the orange ones, which are method definitions

PSO - Overview

- Particle Swarm Optimization is a Gradient-free, bio-inspired, stochastic, population-based algorithm to optimize any kind of process towards a certain goal:
  - No maths involved in the optimization (no gradients or loss functions!).
  - It just keeps trying configurations and saves the best-performing one.
    - It mimics how animals look for resources, by trial and error.
- How it works:
  - We have N “particles” (in our case: configurations of the BDT). Then:
    1) The BDT runs with the configuration of the particle.
    2) When finished, each particle gets a performance score.
       -We define a Function Of Merit (FOM) for this scoring
    3) We track each particle’s best configuration and the best global one.
    4) The particles moves to a new configuration (next slide).

For each iteration

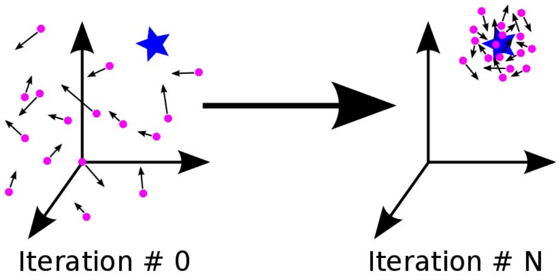
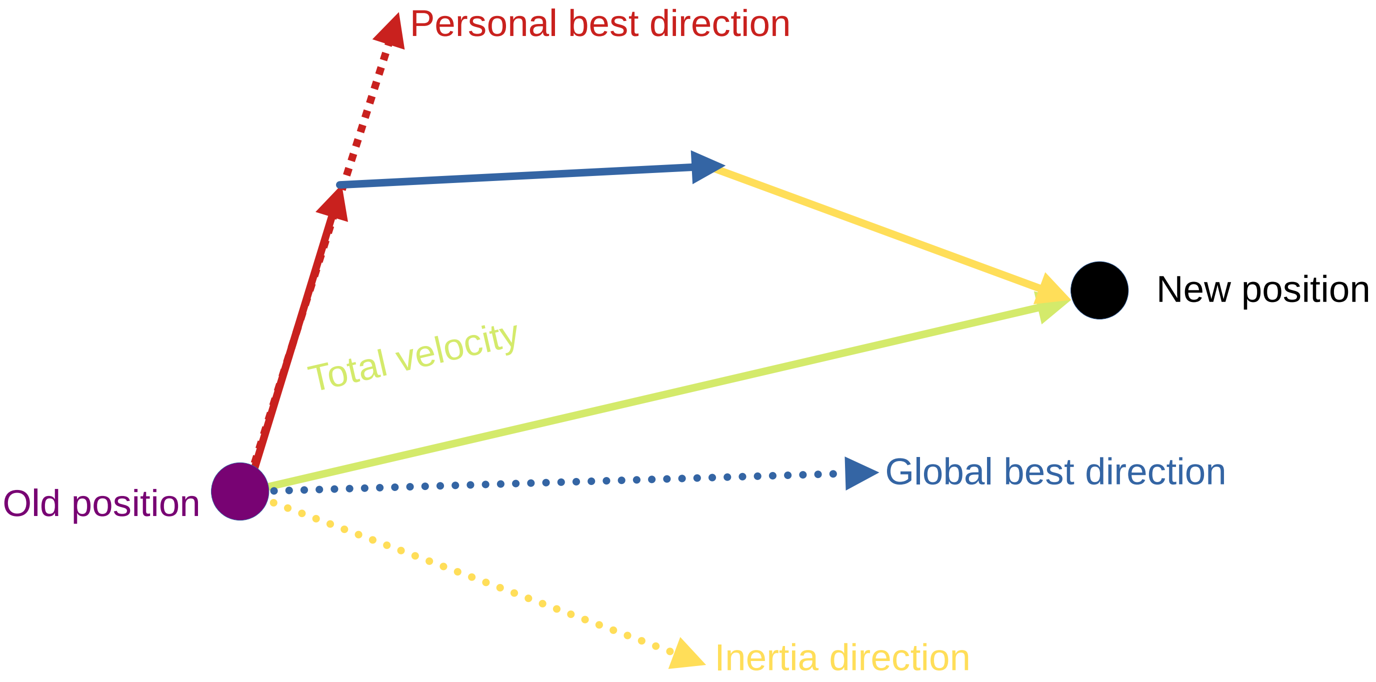


Image taken from a website

PSO - Overview

Position: $\overrightarrow{X_i^{t+1}} = \overrightarrow{X_i^t} + \overrightarrow{V_i^{t+1}}$

Velocity: $\overrightarrow{V_i^{t+1}} = w\overrightarrow{V_i^t} + c_1 r_1 \left(\overrightarrow{P_i^t} - \overrightarrow{X_i^t}\right) + c_2 r_2 \left(\overrightarrow{G^t} - \overrightarrow{X_i^t}\right)$

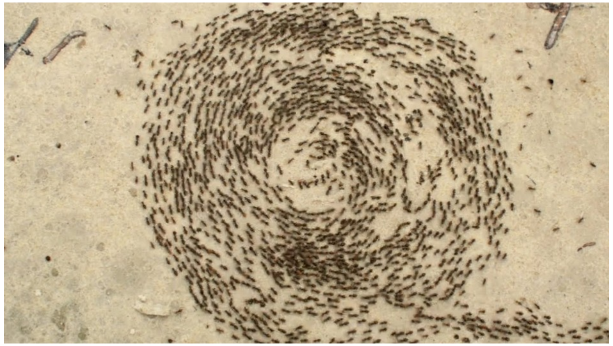


PSO – Adaptation to FT

- We need:
  - A 3-class classifier (b quarks, c quarks, uds quarks).
  - We also want to avoid overfitting:
    - Kolmogorov-Smirnov test
    - Anderson-Darling test

Control biased test scores. (more info in back-up)
Each of them have flaws, so using both is a safer way to go!

  - We need a FOM adapted to 3 different classes.
  - A final check is **always needed**:



Trial and error can go wrong sometimes!

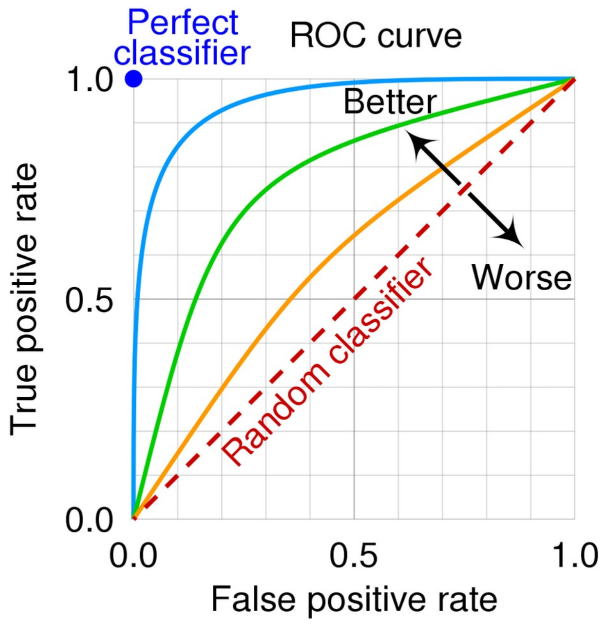
PSO - FOM

- The FOM being used is the averaged value of the Integral of the Receiver Operating Characteristic curve for each of the 3 data classes.
    - Considering the target class as signal and the others as background.

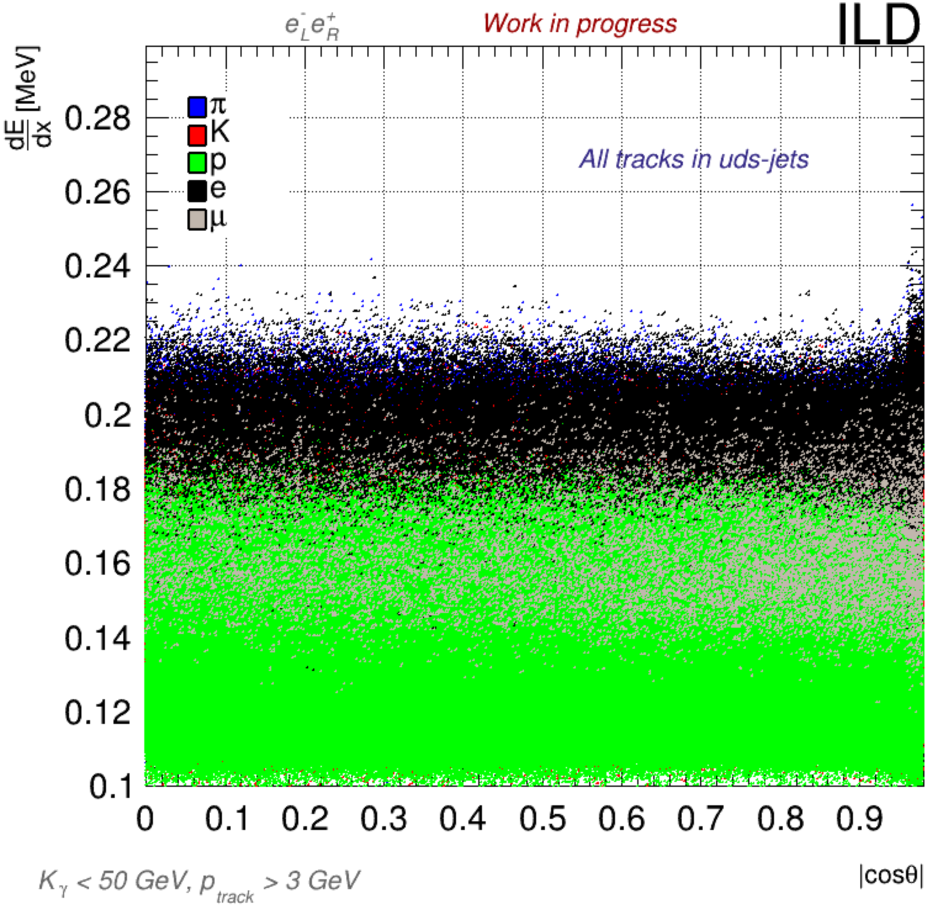
- The FOM is simply:

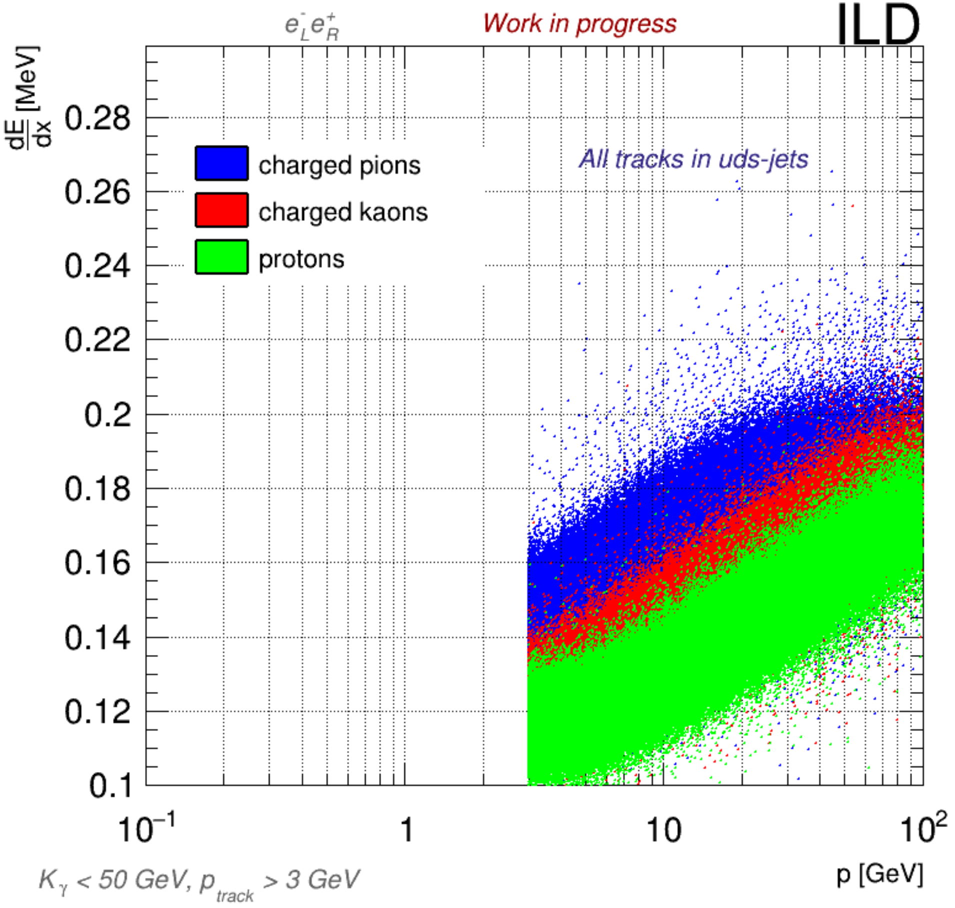
$$FOM = (AUC[b_{quark}] + AUC[c_{quark}] + AUC[uds_{quarks}]) / 3,$$

where AUC = "Area Under Curve" (ROC Integral).



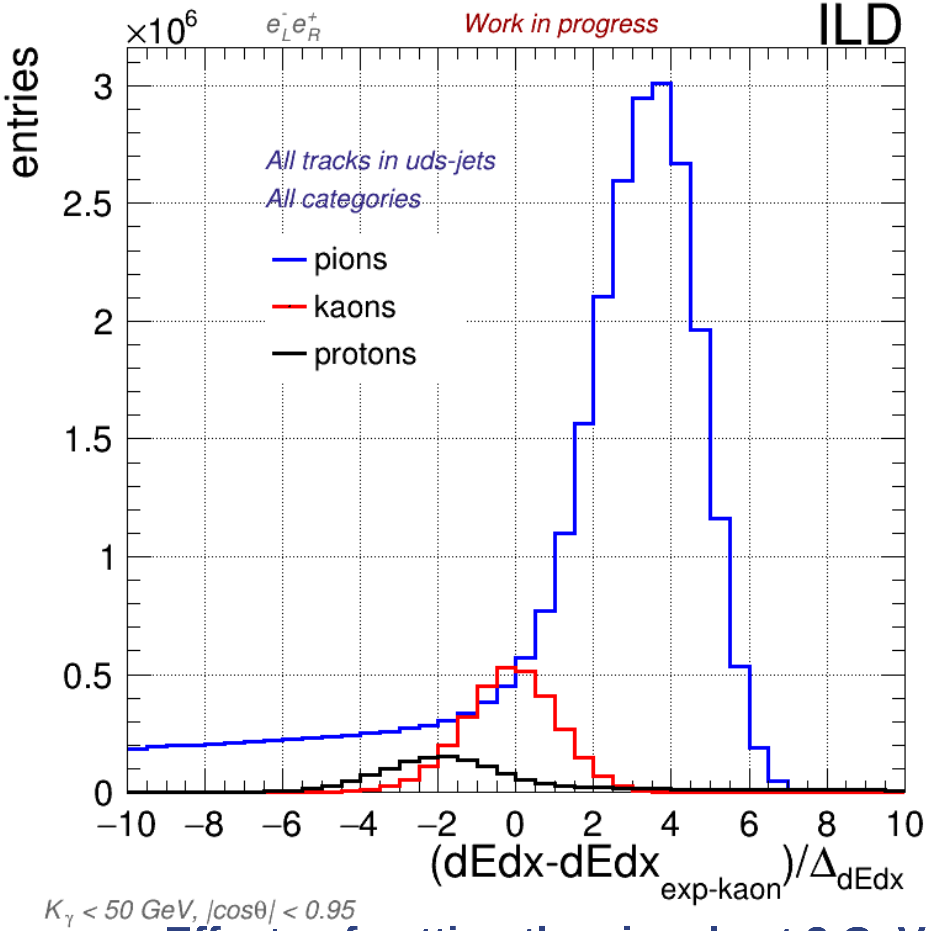
dEdx – Cut in tracks' momentum

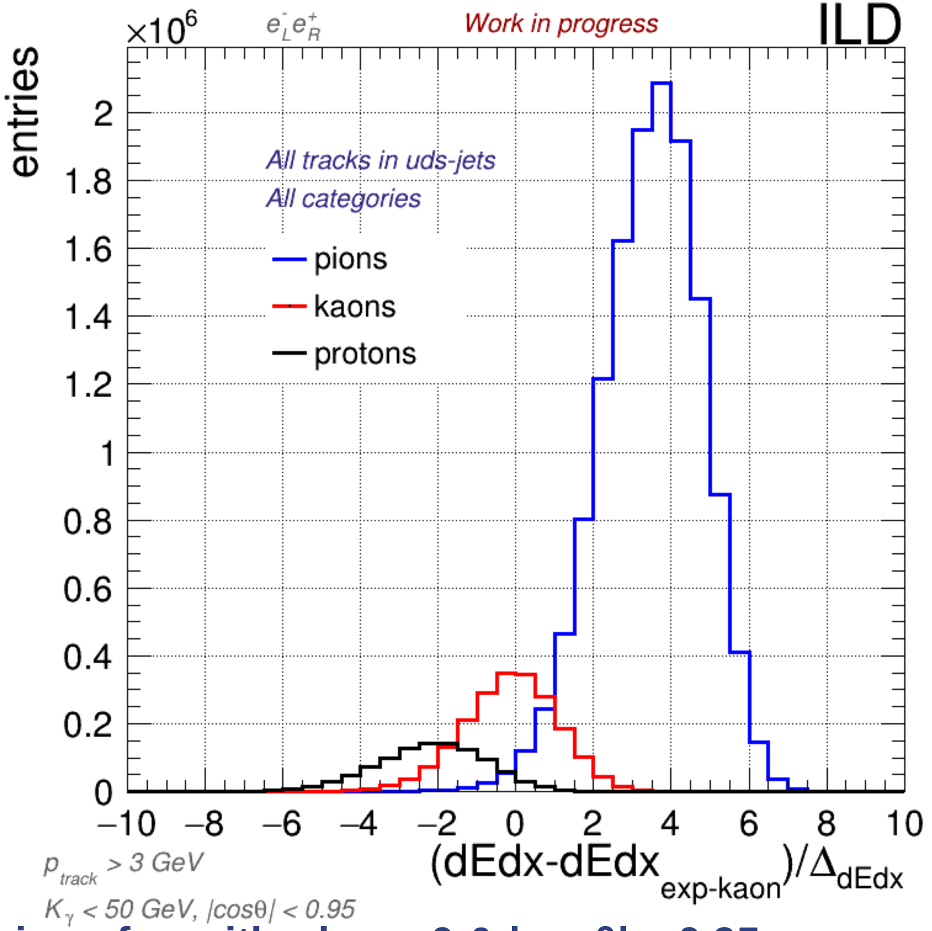




Effects of cutting the signals at 3 GeV. This behavior is similar to b and c jets.

dEdx – Kaon Distance Significance (KDS)





Effects of cutting the signals at 3 GeV, removing pfos with $_1d_{kaon} = 0$ & $|\cos\theta| < 0.95$

1. A peak appeared at d=0. Probably an initialization issue in *dEdxKaonDistance. Plots in back-up slides.*

dEdx – KDS for different quark flavours