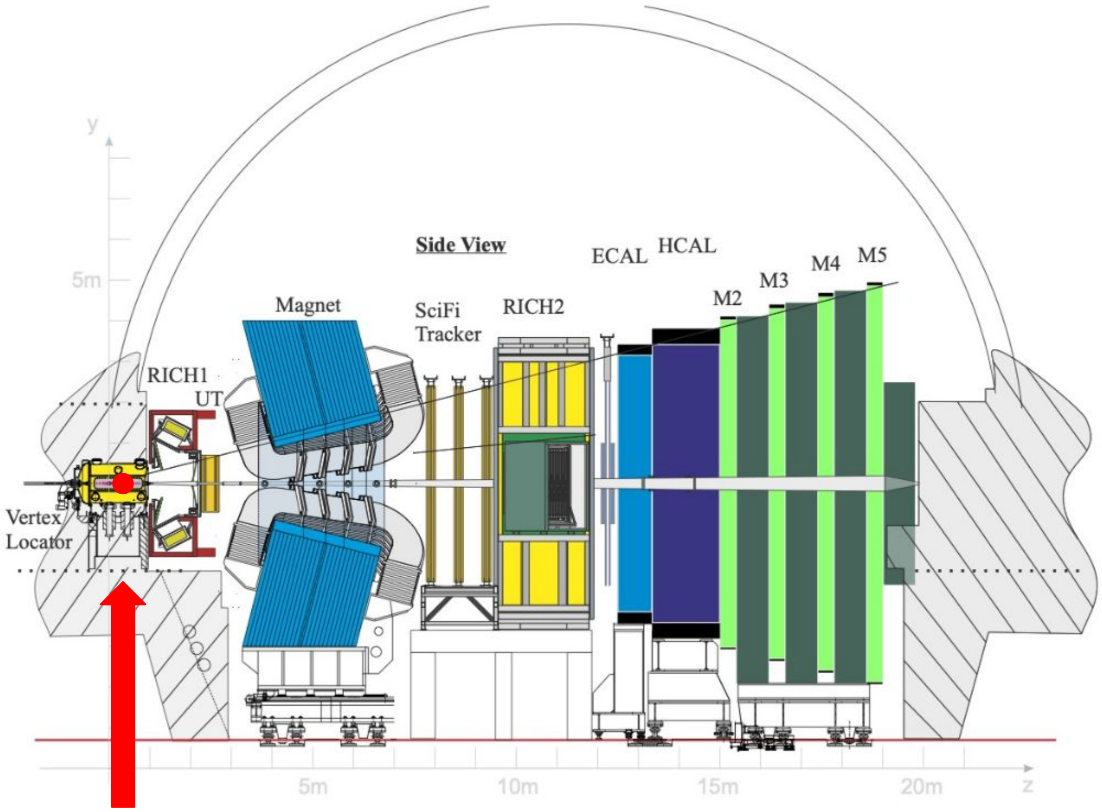

BACK
TO THE DAVINCI
PART II

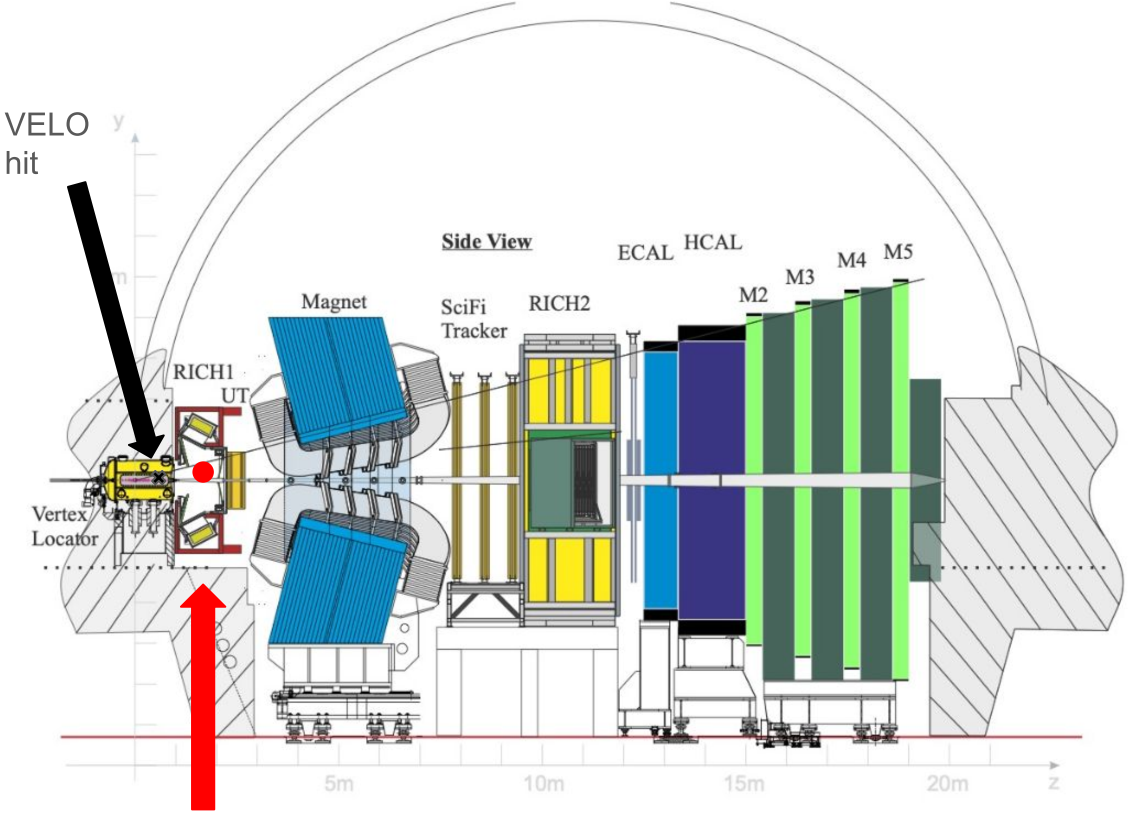
Recap

- We can now run a basic DaVinci script on MC with our python file and yaml file!!
- But just looking at M, P and PT isn't much, we probably want some more variables to help us study this decay in more detail…
- We need more FUNCTORS.
- But first a lightning fast recap on the LHCb detector to help explain where some functors come from!

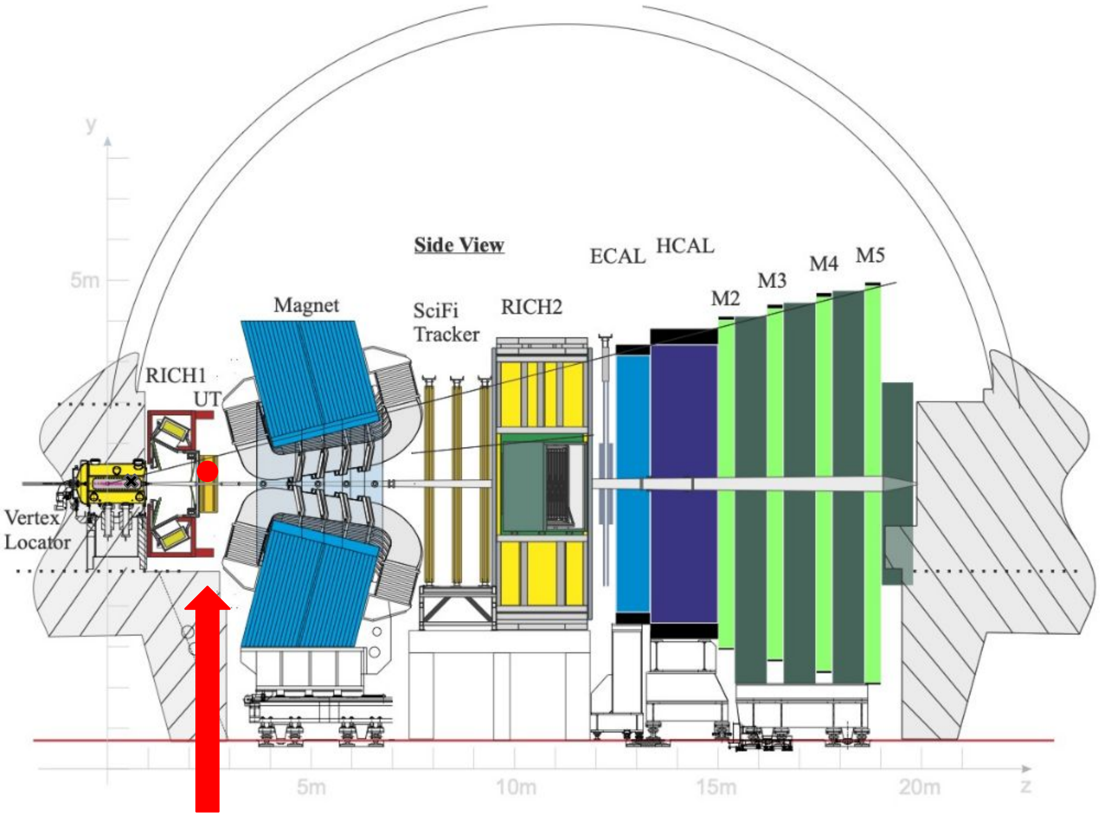
LHCb Basics



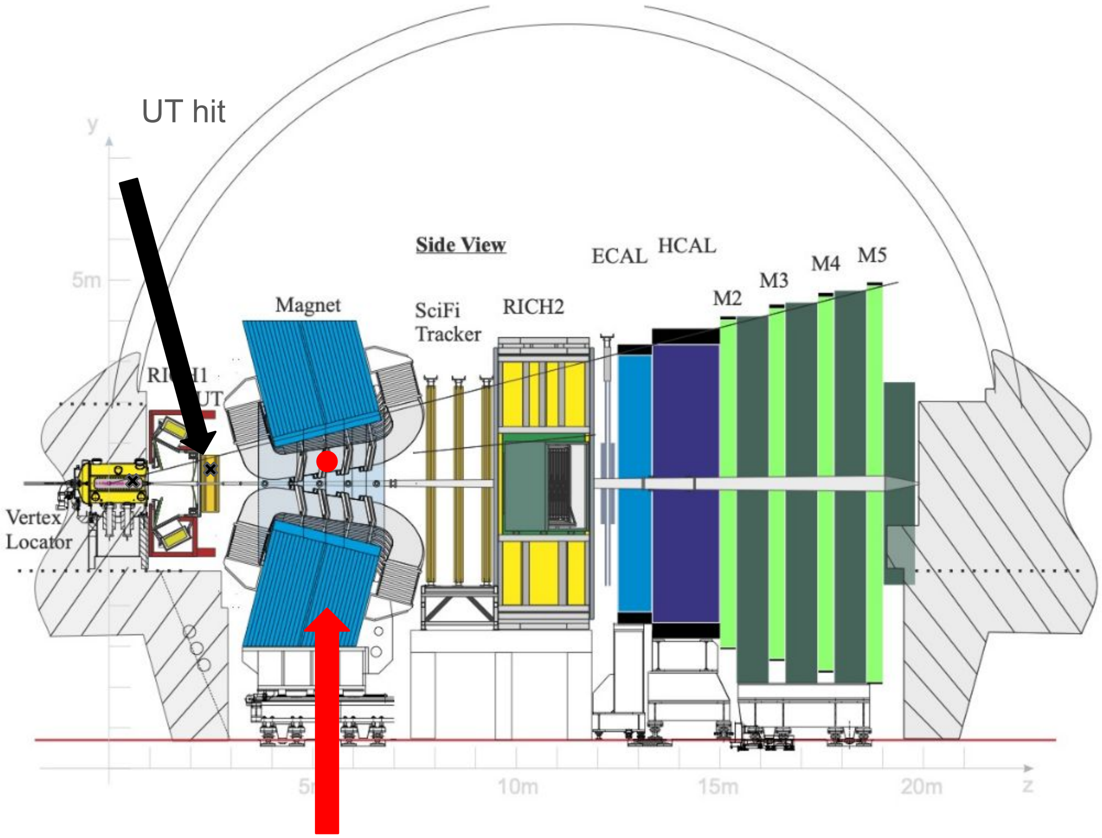
Imagine we are some charged particle that has just been created in a proton-proton (PP) collision in the VELO.



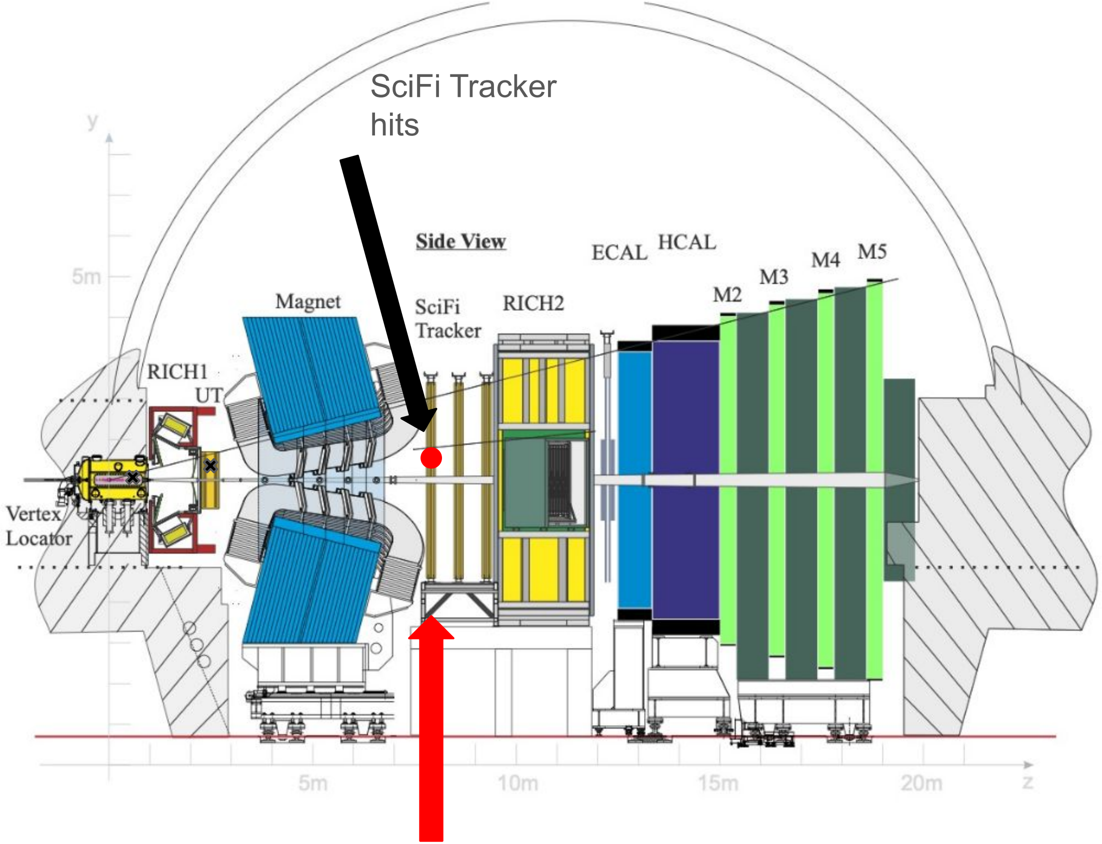
We travel through the RICH detector giving us some particle species identification (PID).



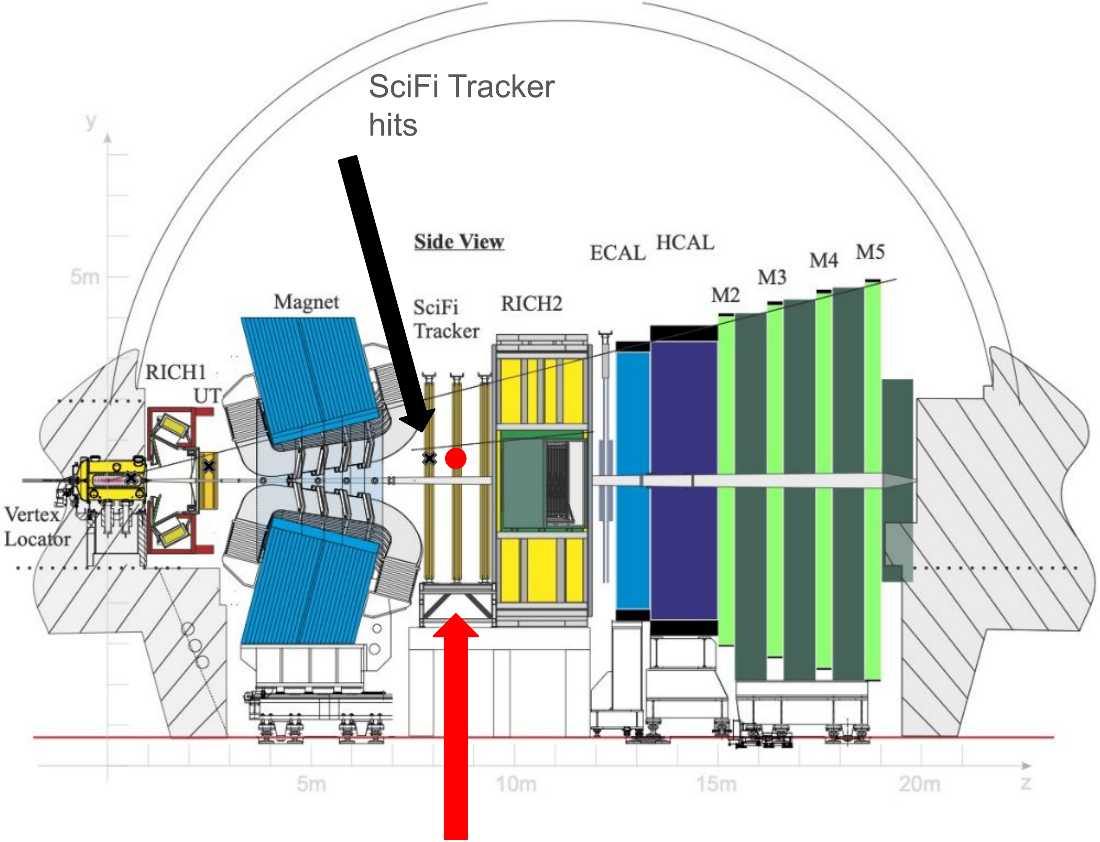
Then we leave some more hits in the upstream tracking station.



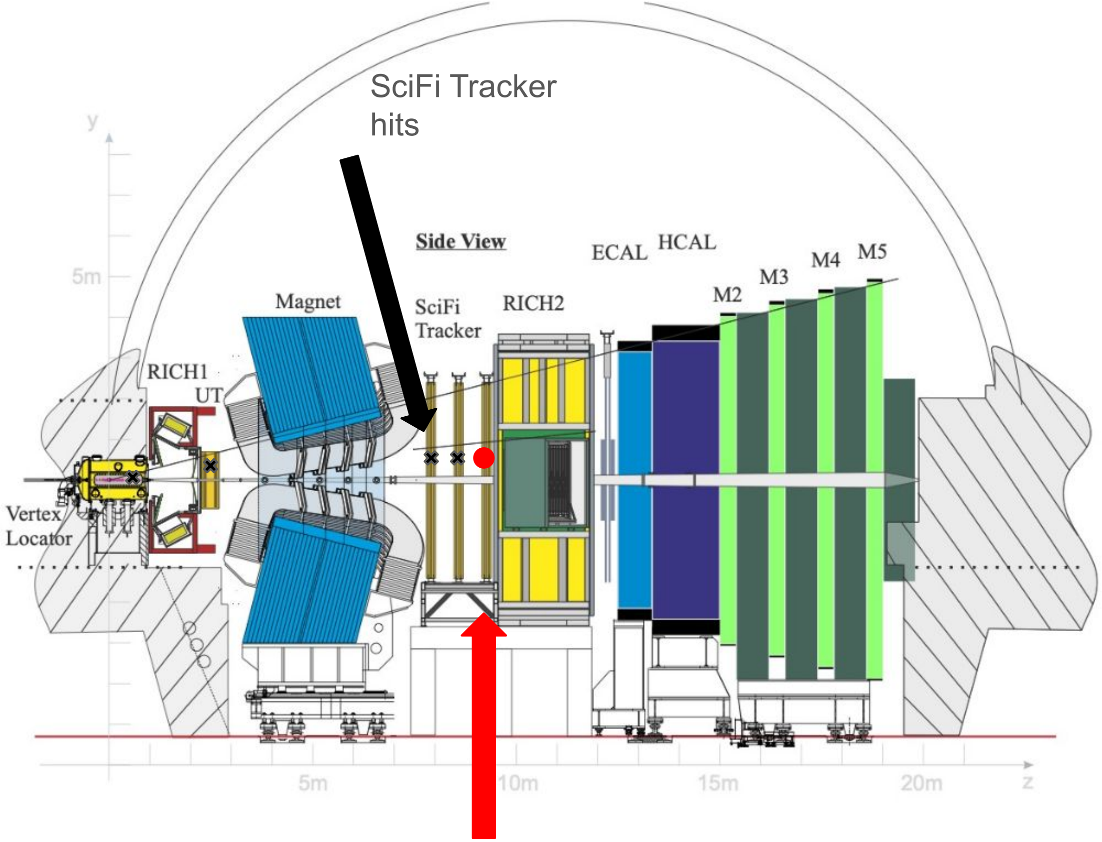
Bent by the magnet.



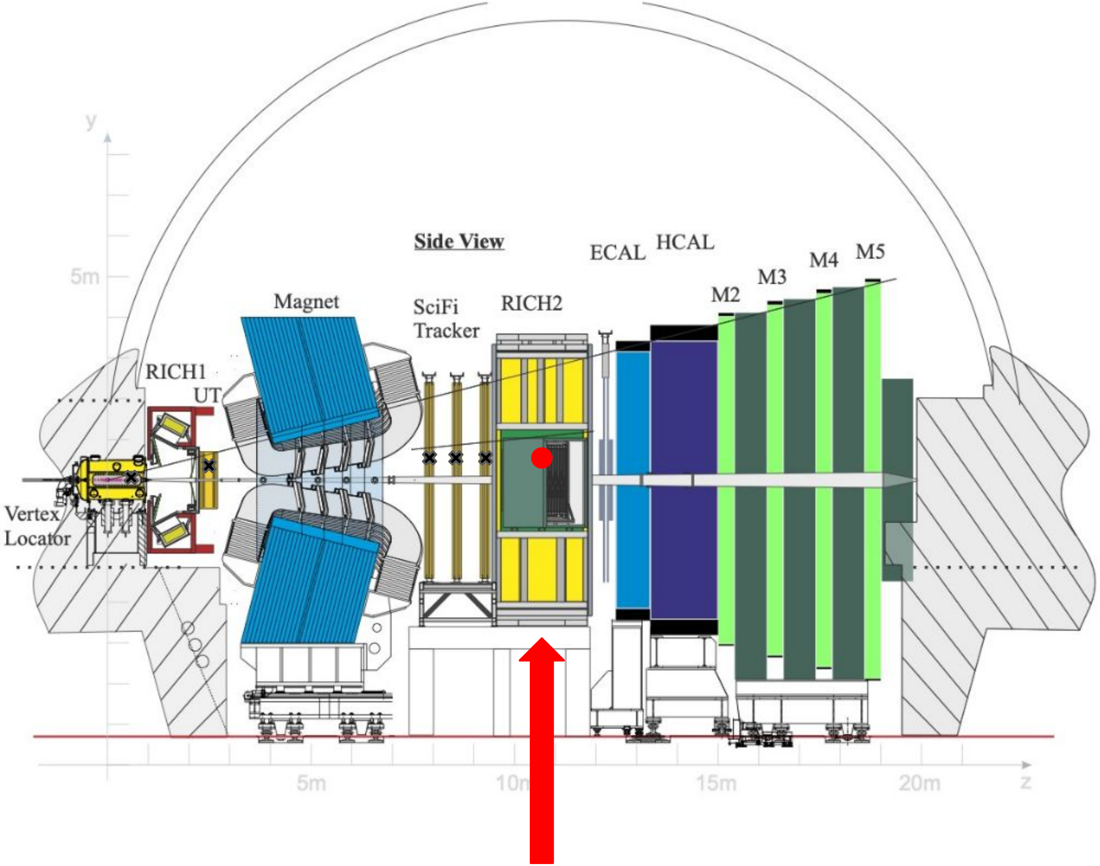
Leave hits in the next tracking stations.



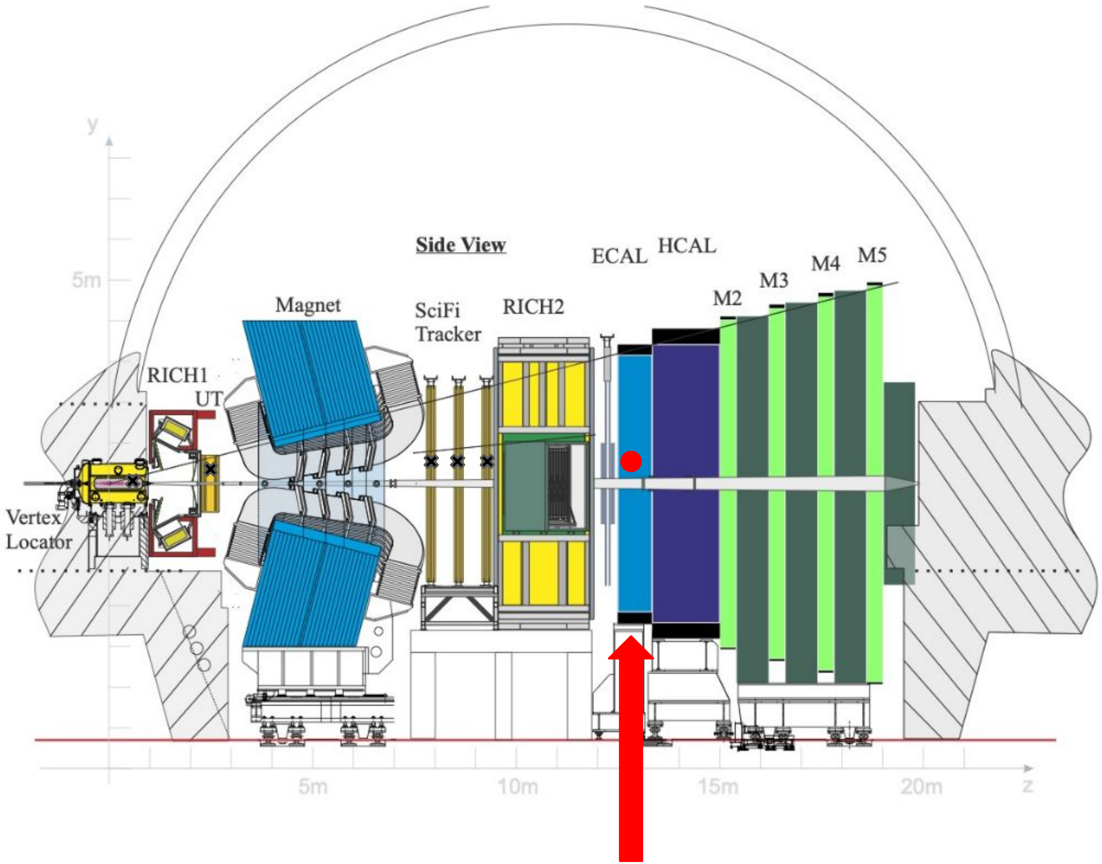
Leave hits in the next tracking stations.



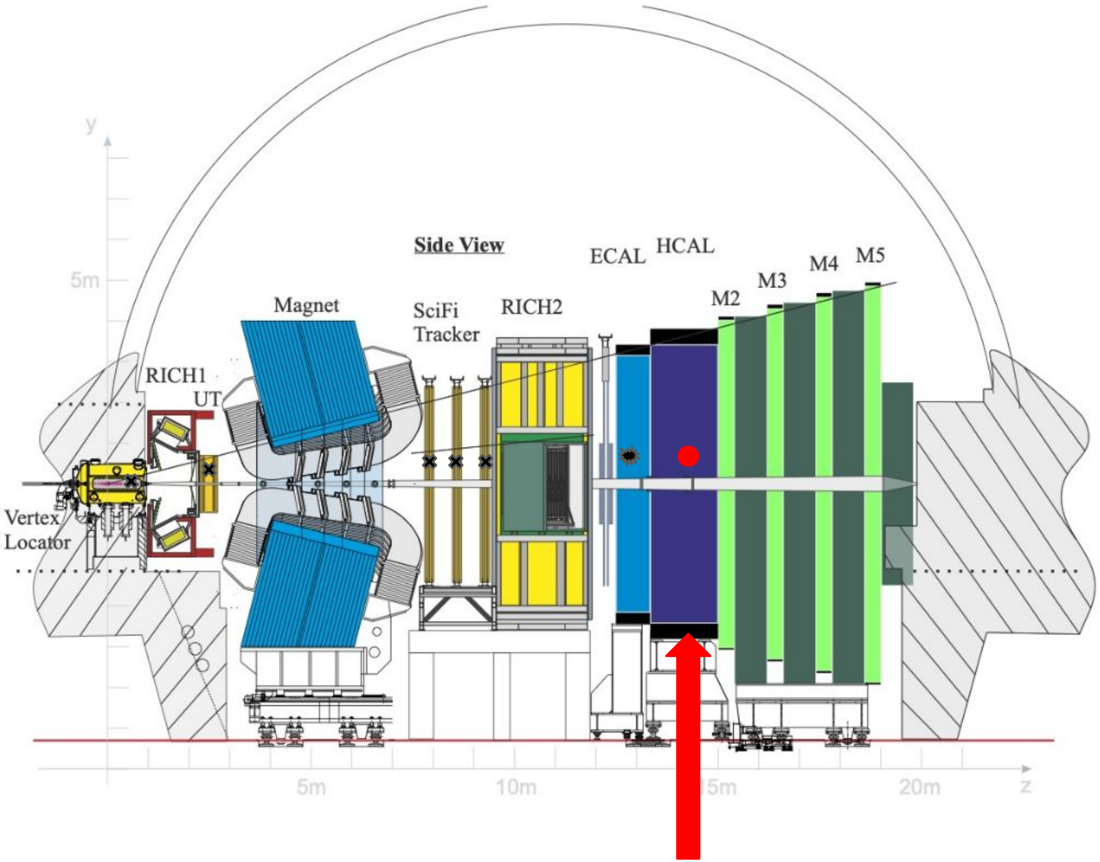
Leave hits in the next tracking stations.



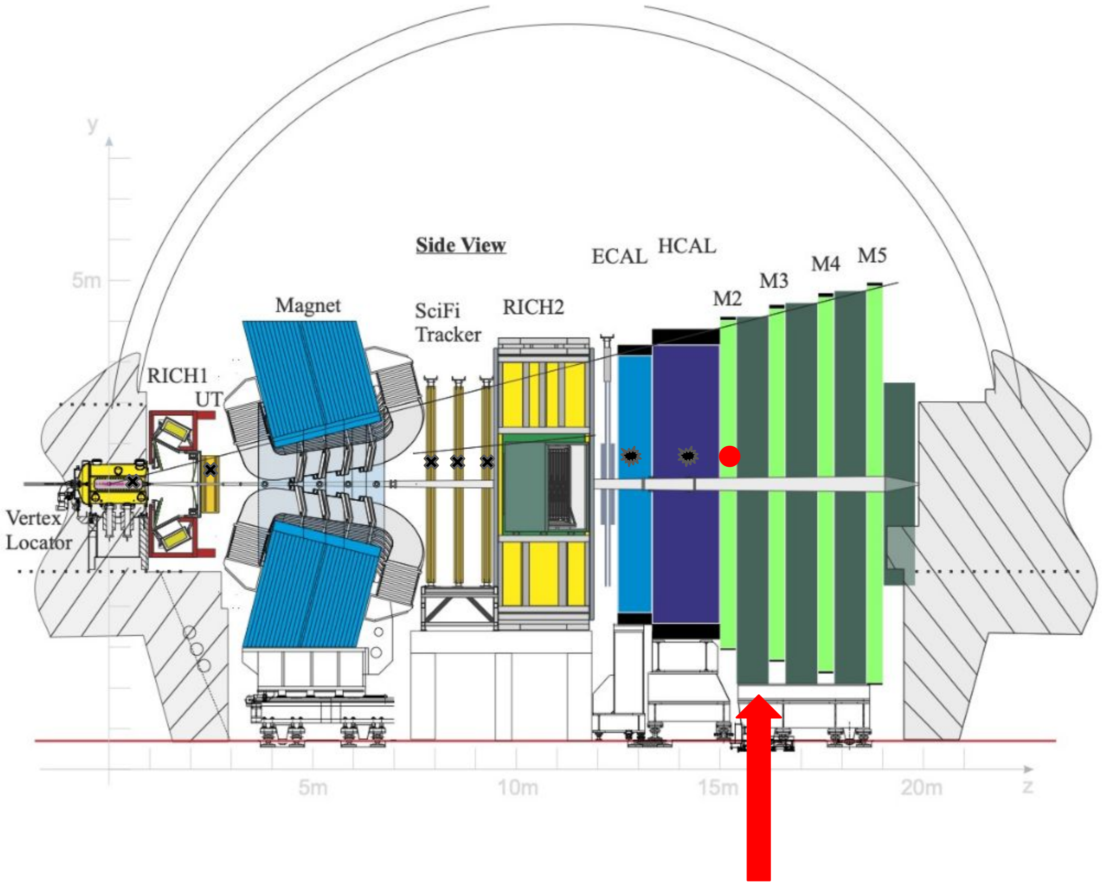
Further PID in RICH2



Then electrons and photons deposit their energy into the ECAL (Electromagnetic Calorimeter).



Hadrons will leave hadronic showers in the HCAL (Hadronic Calorimeter).



Then only very high energy particles e.g. muons will get this far and interact with the muon chambers.

Functors

There are loads!

- There are lots of different variables we can put in our root file, things like the mass, momentum and transverse momentum are fairly self explanatory.
- But what if I asked you for the IPCHI2, the DOCA's or the ghost probability…
- These can at first appear a bit more abstract or just confusing acronyms.
- In this lesson I will try to explain the most common functors and what they actually mean and then together we will adapt our basic script to include these in our decay!

Link here for a list of all functors

Link here for an example using all functors

Some simple ones

- F.PARTICLE_ID -> Gives you the PDG (Particle Data Group) assigned particle ID.
- F.CHARGE -> Gives you the charge…
- F.ETA -> Eta angle also known as Pseudorapidity
- F.PHI -> Phi angle
- F.ISMUON -> Is it a muon ? (0 or 1)
- F.END_VX/VY/VZ -> gives you the end vertex position
- F.PX/PY/PZ -> Gives you the momentum in each direction
- F.ENERGY -> The energy

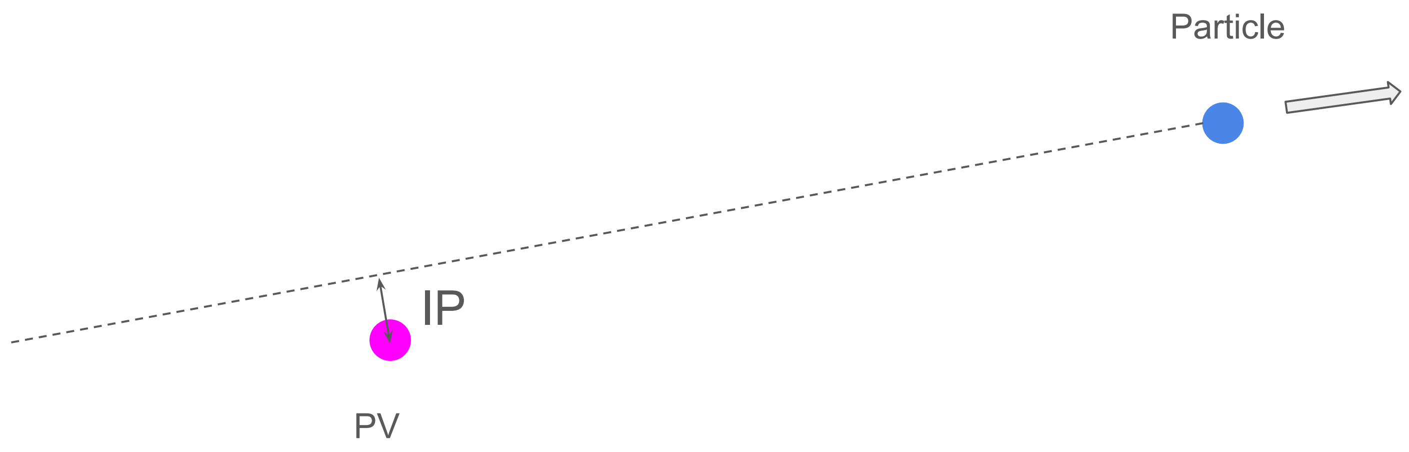
OWNPV

- The OWNPV prefix just means it calculates this value with respect to its vertex. So calculations where the particles value varies depending where the vertex is in space.
- OWNPV_X/Y/Z ->The position in X, Y or Z of the PV
- OWNPV_FD -> The flight distance of the particle from its vertex
- OWNPV_LTIME -> The lifetime from its vertex

For the next few we will zoom into the LHCb!

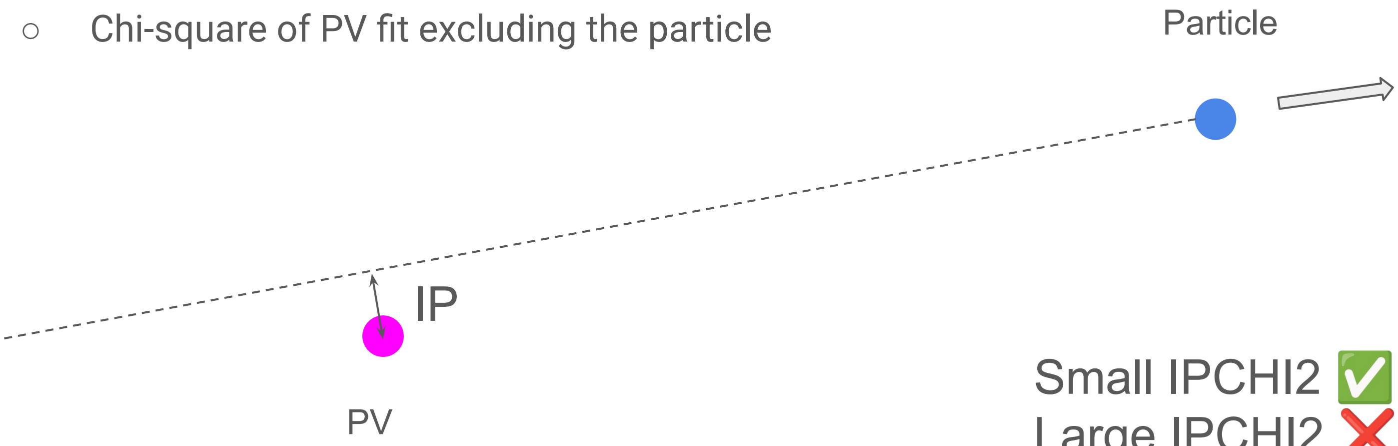
IP - Impact Parameter

- The IP is the shortest distance between a track and some vertex (e.g. PV).



IPCHI2 - Impact Parameter Chi Squared

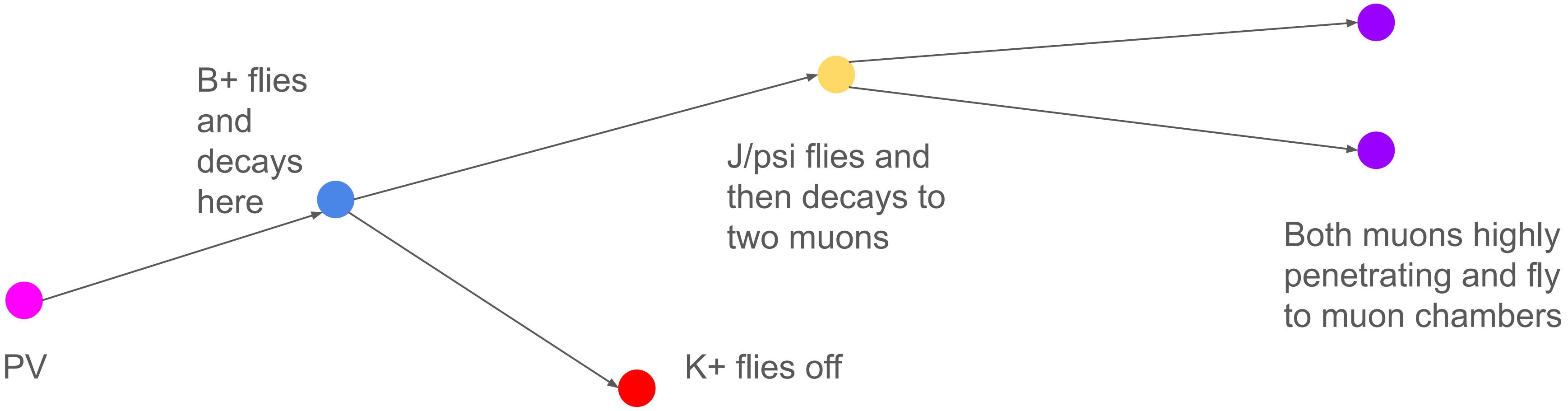
- It gives a measure of how likely the track originated from that vertex.
- The Chi-square is computed as the difference between the:
  - Chi-square of PV fit including the particle
  - Chi-square of PV fit excluding the particle



Small IPCHI2 ✅
Large IPCHI2 ❌

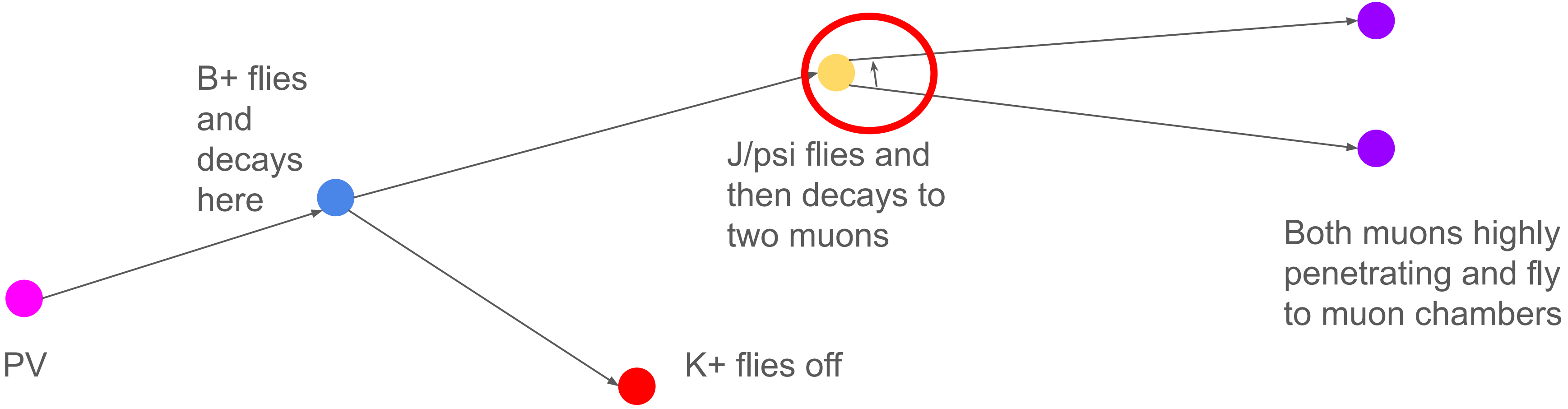
DOCA - Distance Of Closest Approach

- The shortest distance between a pair of tracks.
- So for our B+ ->J/psi (mu+ mu-) K+ channel we will probably want a small DOCA value between the two muons (since we say they came from the same vertex where the J/psi decayed).



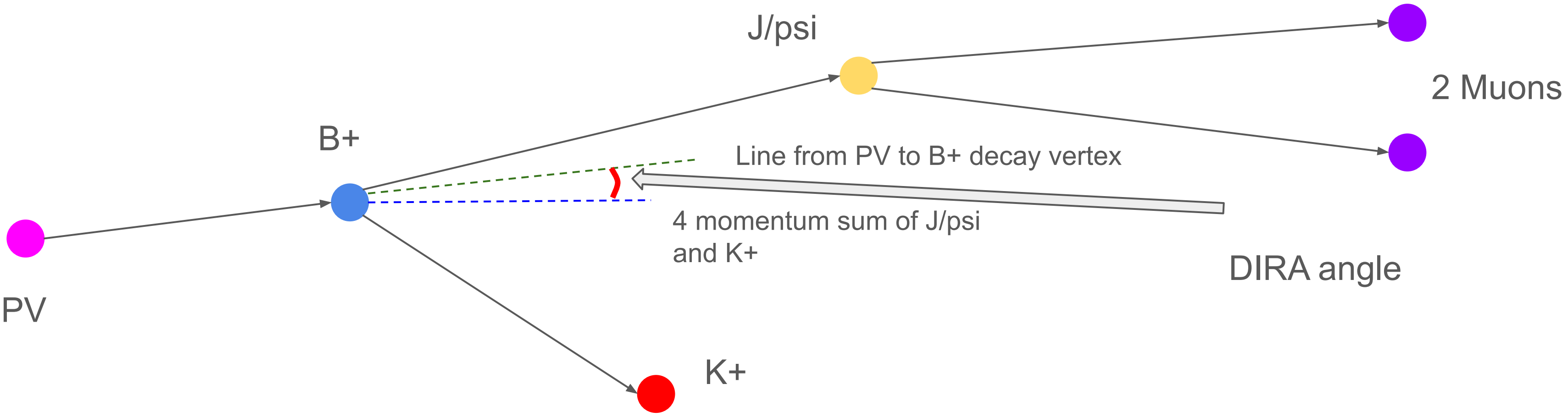
DOCA - Distance Of Closest Approach

- The shortest distance between a pair of tracks.
- So for our B+ ->J/psi (mu+ mu-) K+ channel we will probably want a small DOCA value between the two muons (since we say they came from the same vertex where the J/psi decayed).



DIRA - DIRection Angle

- The angle between a line drawn from the PV to the decay vertex and the sum of the 4 momentum of its decay products. Often take cos(DIRA), if ~ 1 then better pointing of momentum back to PV -> More likely to be a real B candidate.



PID - Particle IDentification

- The RICH detectors give us our PID values in the form of
- F.PID_P -> Proton identification
- F.PID_K -> Kaon identification
- F.PID_E -> Electron identification
- F.PID_MU -> Muon identification
- F.PID_PI -> Pion identification

These give the measure of the (proton, kaon etc) mass hypothesis relative to the pion hypothesis. More info on PID here. These use a log likelihood method.

PROBNN

- To expand further than the simpler log likelihood approach of PID variables PROBNN were developed.
- These also take into account correlations between detector subsystems and tracking.
- An MVA is trained with these features to produce probabilities for each particle.
- e.g. PROBNN_K, PROBNN_PI, PROBNN_P etc
- Also PROBNN_GHOST …

PROBNN_GHOST

Real Track:

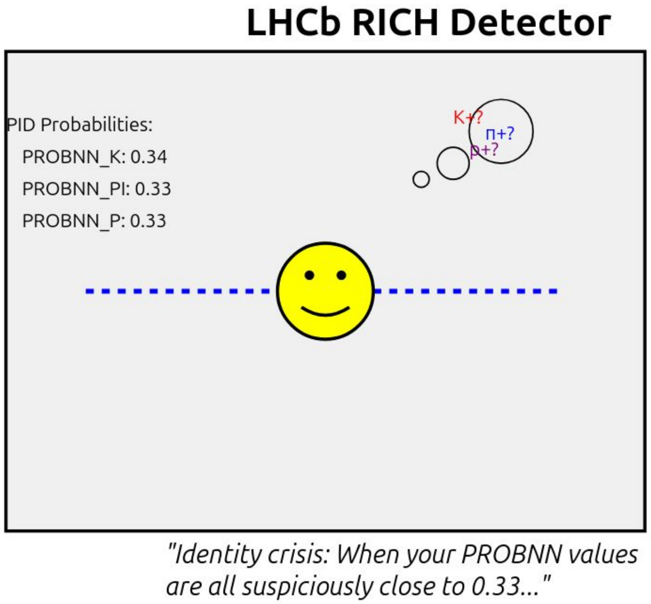
- Corresponds to actual particle
- Hits align well
- Points to physics vertex
- Consistent momentum

Ghost Track:

- Random combination of hits
- May use hits from multiple particles
- Could be noise hits
- Doesn't correspond to real particle

AI Funny?

I asked AI Chatbot Claude if he could generate a joke about PID...



You can decide for yourself …

Endless Functors

- There are of course many more!
- Can use things like F.MAX, F.MIN, F.SUM etc
- Will demonstrate some more in the code.
- As well as things called FunctorCollections that group together commonly used variables e.g. FC.Kinematics groups M, P, PX, PY, PZ, PT and Energy.

Really useful LHCb glossary I really recommend you check out here!

Intermediate Script

Your Turn - Digest the Info

- Please go to the gitlab and copy across the intermediate script, link here
- There is too much to write this by hand live - key changes:

  - We have used these functor collections. FC.Kinematics, FC.RecSummary etc. I have defined them in the script but feel free to check more out here and try and add your own extra ones!

  - Created an all_vars category for values we want for all, composite variables for the B+ and J/psi and track variables for muons and K+. This means we had to edit the variables dictionary:

  - Included this event_variables argument in funtuple to attach calculations that apply to the overall event e.g. the number of long tracks.

- The logic is exactly the same so take 20-30 mins now to copy across, read through the comments, check your script still runs and check the output has these extra variables!
- If you are stuck ask the person next to you and chat through it together! Or ask us!!

```
variables = {
    "ALL": all_vars,
    "Bplus": composite_variables,
    "Jpsi": composite_variables,
    "muplus": track_variables,
    "muminus": track_variables,
    "Kplus": track_variables
}
```

🍲🥘🥫😋

- Lunch time!
- After lunch I’m afraid you still have me…
- We will be having our final lesson on some more advanced topics!
- Maybe a small quiz to see what you have remembered + a prize!

BACKUP

OWNPV vs BPV(pvs)

New: OWNPV functors

- for every ‘BPV’ functor, there now is an equivalent ‘OWNPV’ functor, e.g.

```
F.OWNPVFDCHI2 > bpvvdchi2_min,
```

is equivalent to

```
F.BPVFDCHI2(defaultpvlist) > bpvvdchi2_min,
```

- pros/cons
  - easier configuration, less error prone
  - to be proven: save CPU by using the OWNPV functors
  - less flexibility

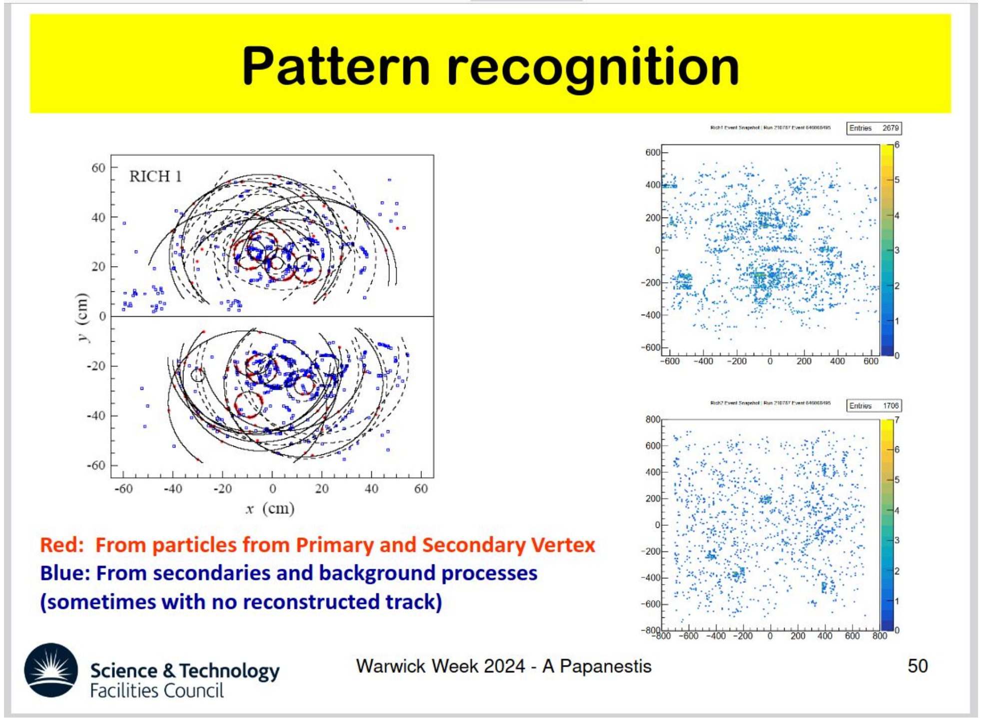
BPV Definition

-

```python
def BPV(Vertices: DataHandle = None):
    """ Determine Best Primary Vertex of a particle

    Functor's call operator expects a particle-like object.

    Note:
        "Best" is chosen as the PV for which the particle has the smallest impact parameter (IP).
        Note that this is different to Run 1+2, where "best" was chosen based on IP chi^2.
        See also :py:func:`~BEST_PV` functor for links to where this is defined in the code.
        See also final slide of final talk in https://indico.cern.ch/event/1361613/ for more details.

    Args:
        Vertices: DataHandle of the vertices - if not provided will use :py:func:`~OWNPV`

    """
    if Vertices is None: return OWNPV.bind(FORWARDARGS)
    return BEST_PV.bind(TES(Vertices), FORWARDARGS)
```

PID at LHCb

- Full credit to Antonis Papanestis!



PID at LHCb

- Full credit to Antonis Papanestis!

Pattern Recognition in Accelerator based Cherenkov Detector

- Events with large number of charged tracks giving rise to several overlapping Cherenkov Rings on the Photo detector plane.
  Problem: To identify which tracks correspond to which hits and then identify the type (e, π, p etc.) of the particle which created the tracks.

- Hough Transform: (used by ALICE at CERN)
  - Project the particle direction on to the detector plane
  - Accumulate the distance of each hit from these projection points in case of circular rings.
  - Collect the peaks in the accumulated set and associate the corresponding hits to the tracks.

- Likelihood Method: (used by LHCb at CERN)
  - For each of the track in the event, for a given mass hypothesis, create photons and project them to the detector plane using the knowledge of the geometry of the detector and its optical properties. Repeat this for all the other tracks.
  - From this calculate the probability that a signal would be seen in each pixel of the detector from all tracks.
  - Compare this with the observed set of photoelectron signal on the pixels, by creating a likelihood.
  - Repeat all the above after changing the set of mass hypothesis of the tracks. Find the set of mass hypothesis, which maximize the likelihood.

(Ref: R.Forty: Nucl. Inst. Mech. A 433 (1999) 257-261) 51

Science & Technology Facilities Council

Warwick Week 2024 - A Papanestis 51