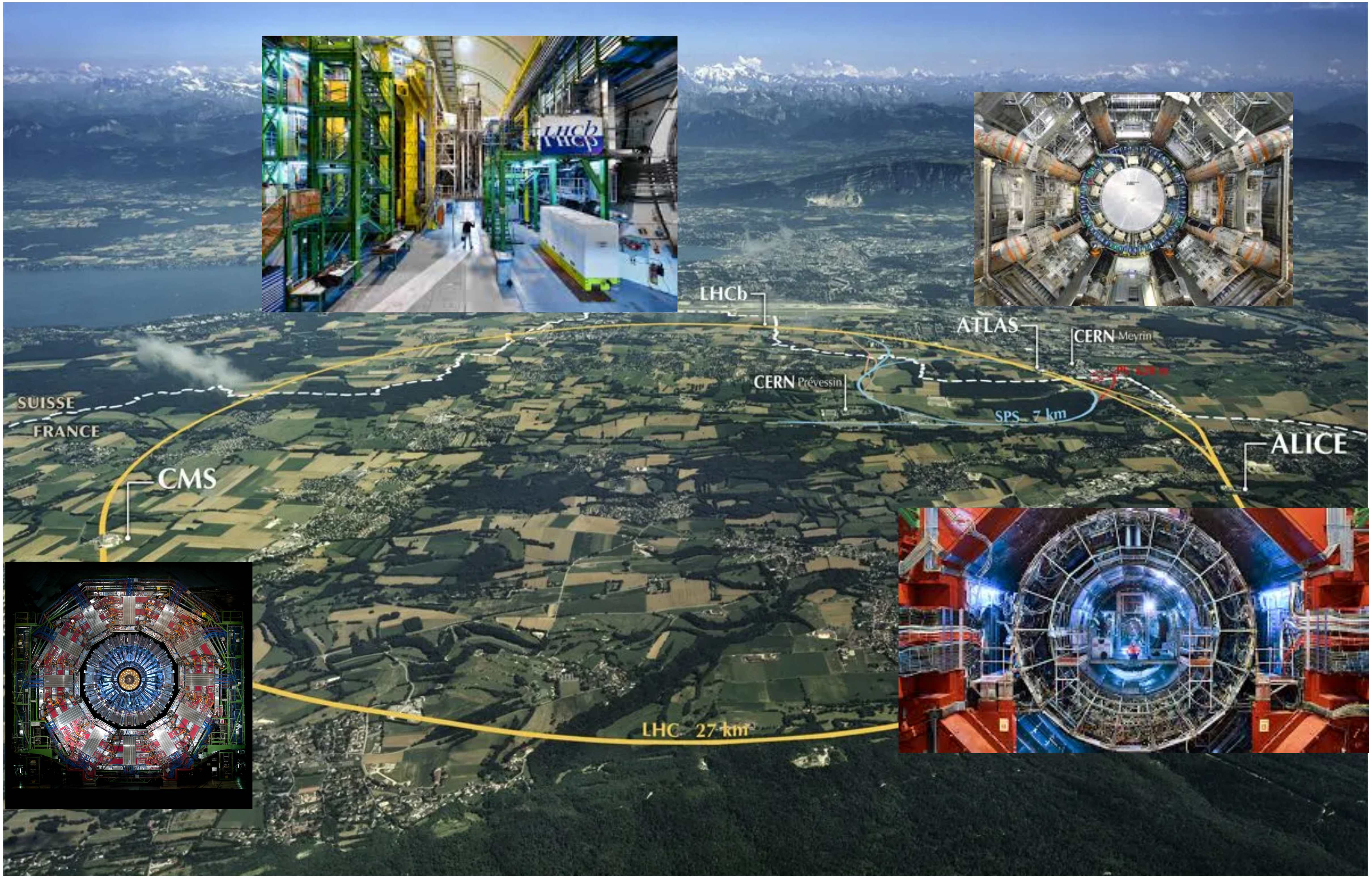
LHCb data analysis hands-on

IDPASC 2025 - Orsay, France

Christina Agapopoulou (IJCLab/CNRS)

First, a bit of intro

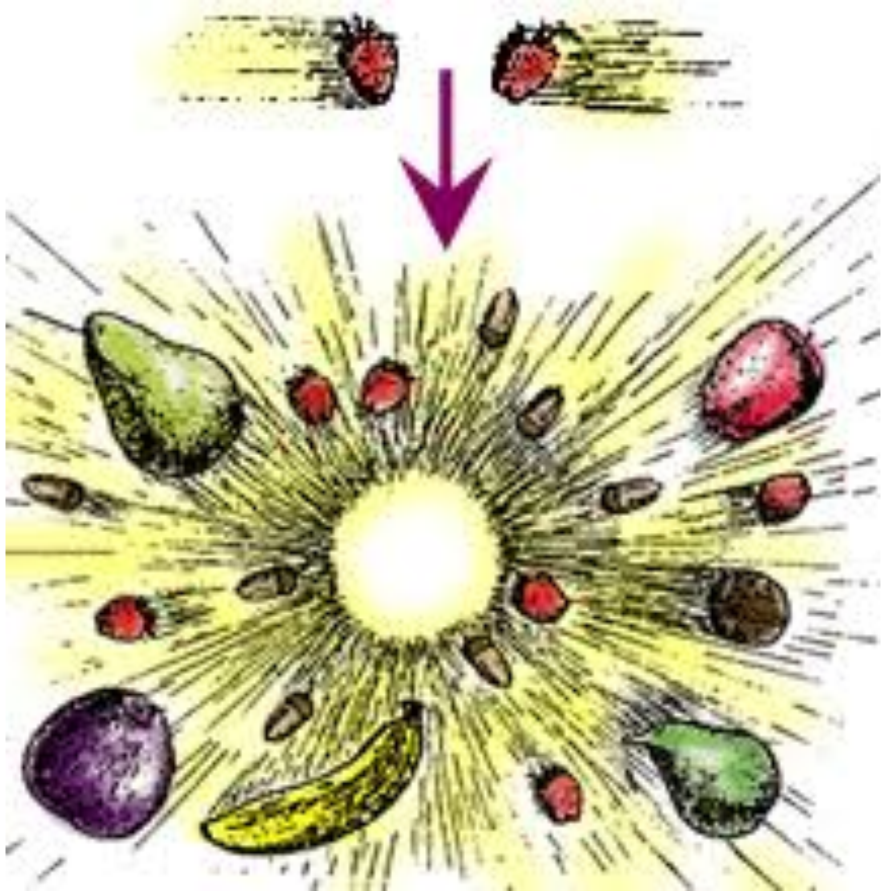
The Large Hadron Collider

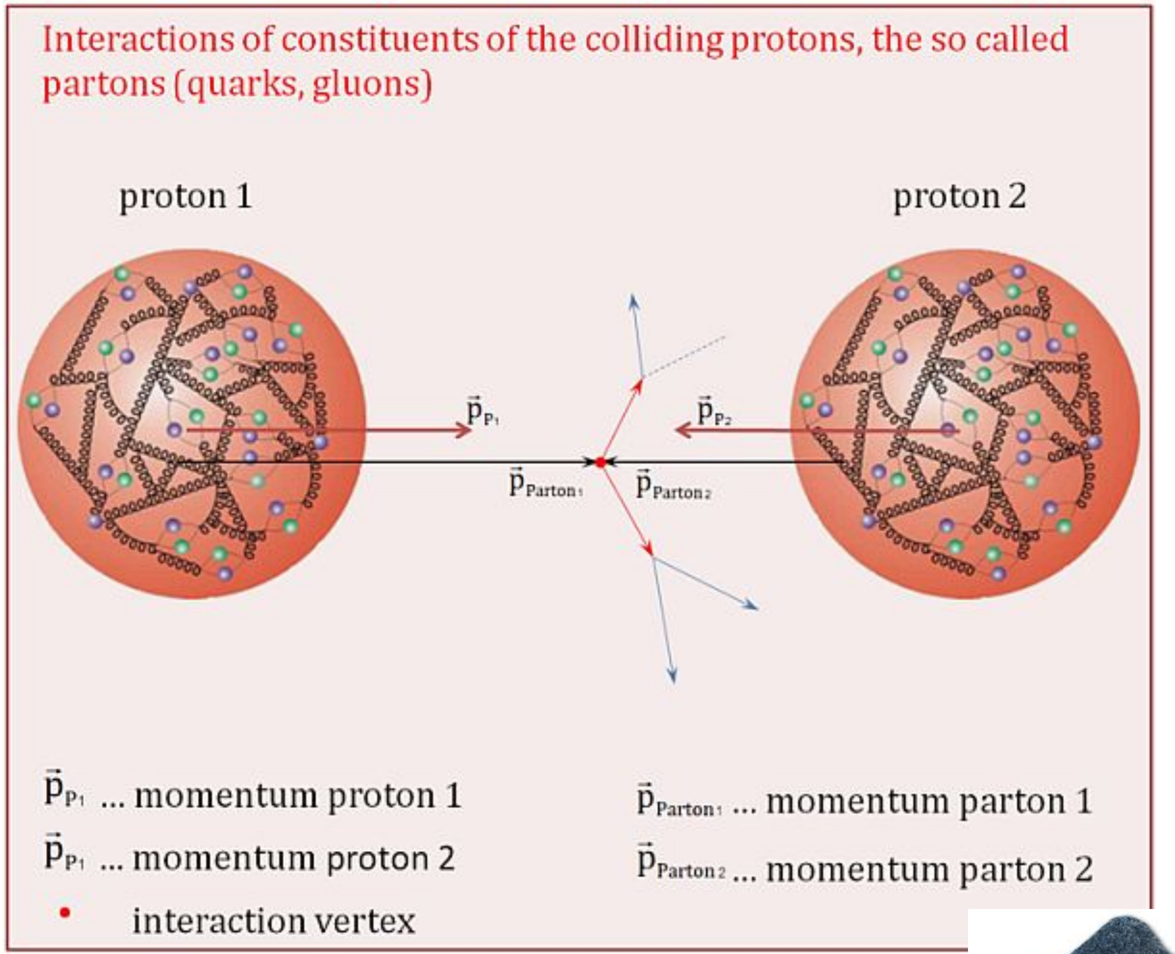


A 27km ring accelerating billions of protons to 99.9999991% the speed of light, and colliding them at TeV energies

What happens when two particles collide

- Or more specifically: what happens when trillions of highly energetic beams of particles collide
- **E=mc²**
- Energy gets converted to mass… and vice versa





Interactions of constituents of the colliding protons, the so called partons (quarks, gluons)

$\vec{p}_{P_1}$ ... momentum proton 1

$\vec{p}_{P_1}$ ... momentum proton 2

• interaction vertex

$\vec{p}_{Parton_1}$ ... momentum parton 1

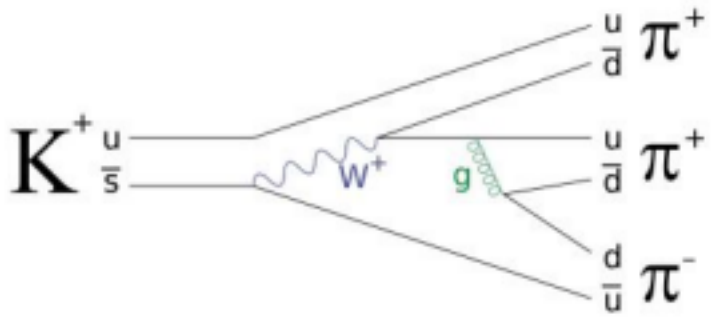
$\vec{p}_{Parton_2}$ ... momentum parton 2

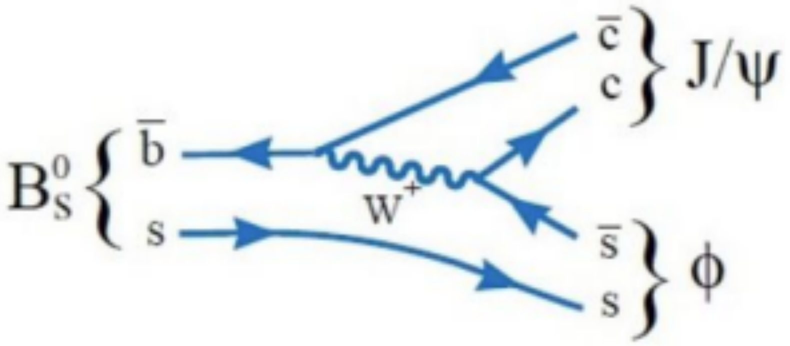
- When ultra-energetic quarks and gluons interact with each other, **new particles can be created**

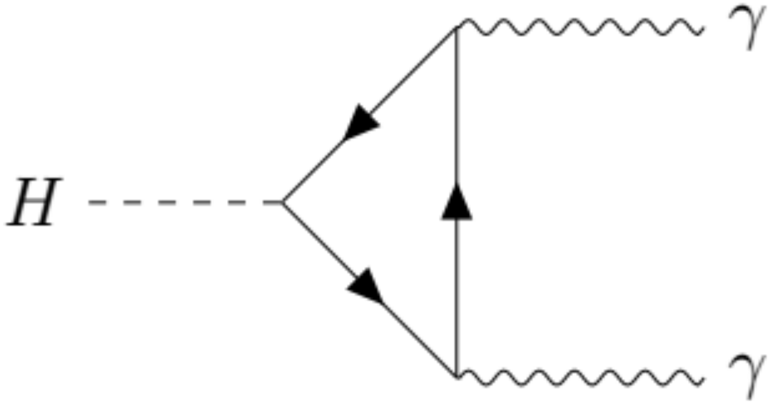
Particle decays

- Not all particles are stable - most of them are actually unstable and will at some point "**decay**" into smaller, lighter particles, conserving energy and momentum

- They will do this in sequence until they reach what we know as **stable particles** (particles which we have never observed decay without external force)







*Typical particle decay times are **extremely small**: 10^{-10} to 10^{-25} seconds. There's very few particles that can travel enough distance for us to detect them!*

- Does this mean we cannot observe the rest?

- **NO! Conservation of momentum and energy + Einstein comes to the rescue**

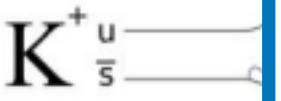
$$E = \sqrt{m^2c^4 + pc^2} \Longrightarrow mc^2 = \sqrt{(E_1 + E_2 + E_3 + ...)^2 - (p_1 + p_2 + p_3 + ...)^2c^4}$$

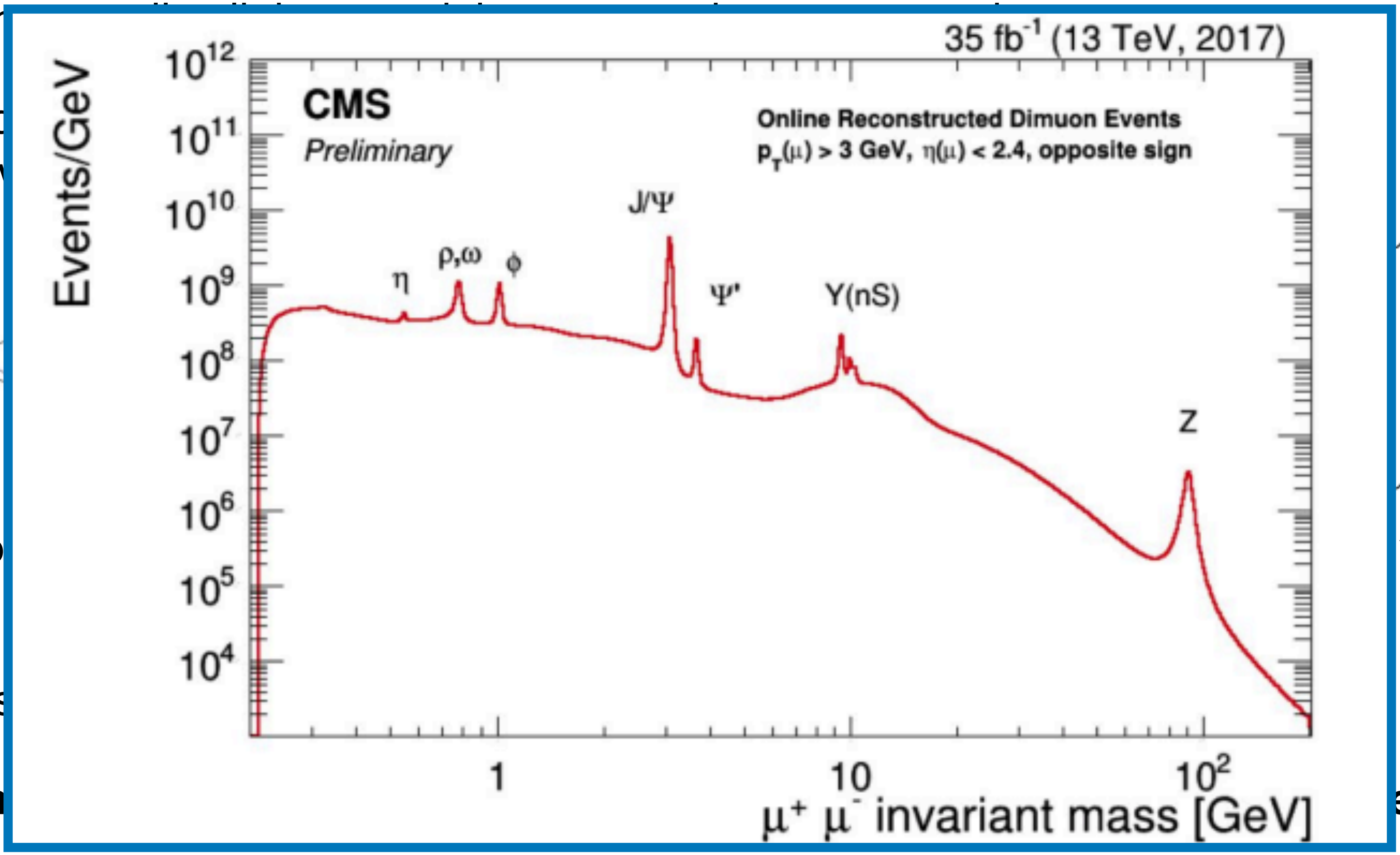
We can get back the original particle by measuring the energy and momentum of it's decay products (or children)

Particle decays

- Not all particles are stable - most of them are actually unstable and will at some point “**decay**” i

- They will c
  (particles v





- Typical p ... very few
  particles

- Does this

- **NO! Con** ... **e**

$$E = \sqrt{m^2c^4 + pc^2} \Longrightarrow mc^2 = \sqrt{(E_1 + E_2 + E_3 + ...)^2 - (p_1 + p_2 + p_3 + ...)^2c^4}$$

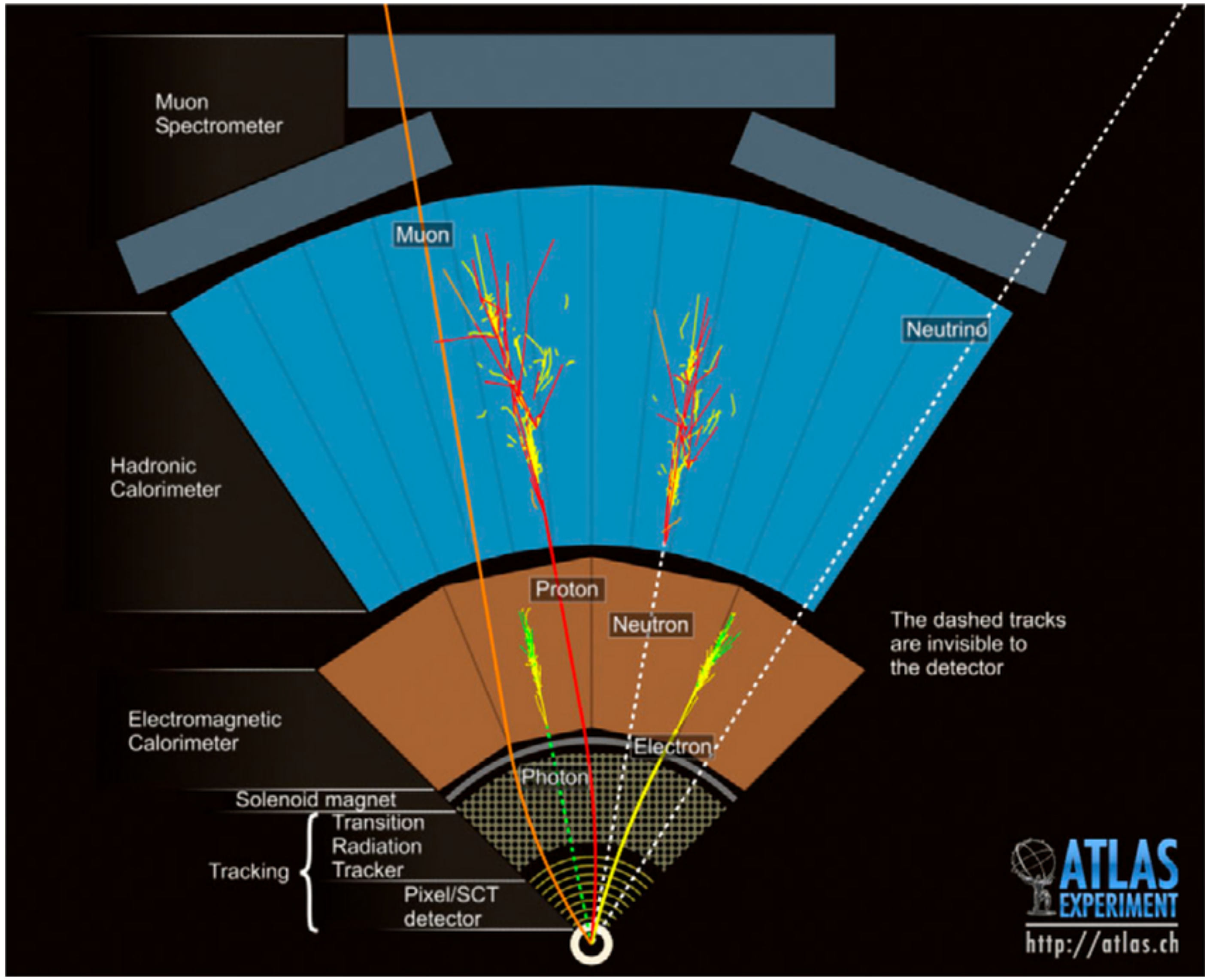
We can get back the original particle by measuring the energy and momentum of it’s decay products (or children)

How do the experiments work

Subatomic particles are **millions of times smaller** than what the world's strongest microscopes can detect… so how do we find them?

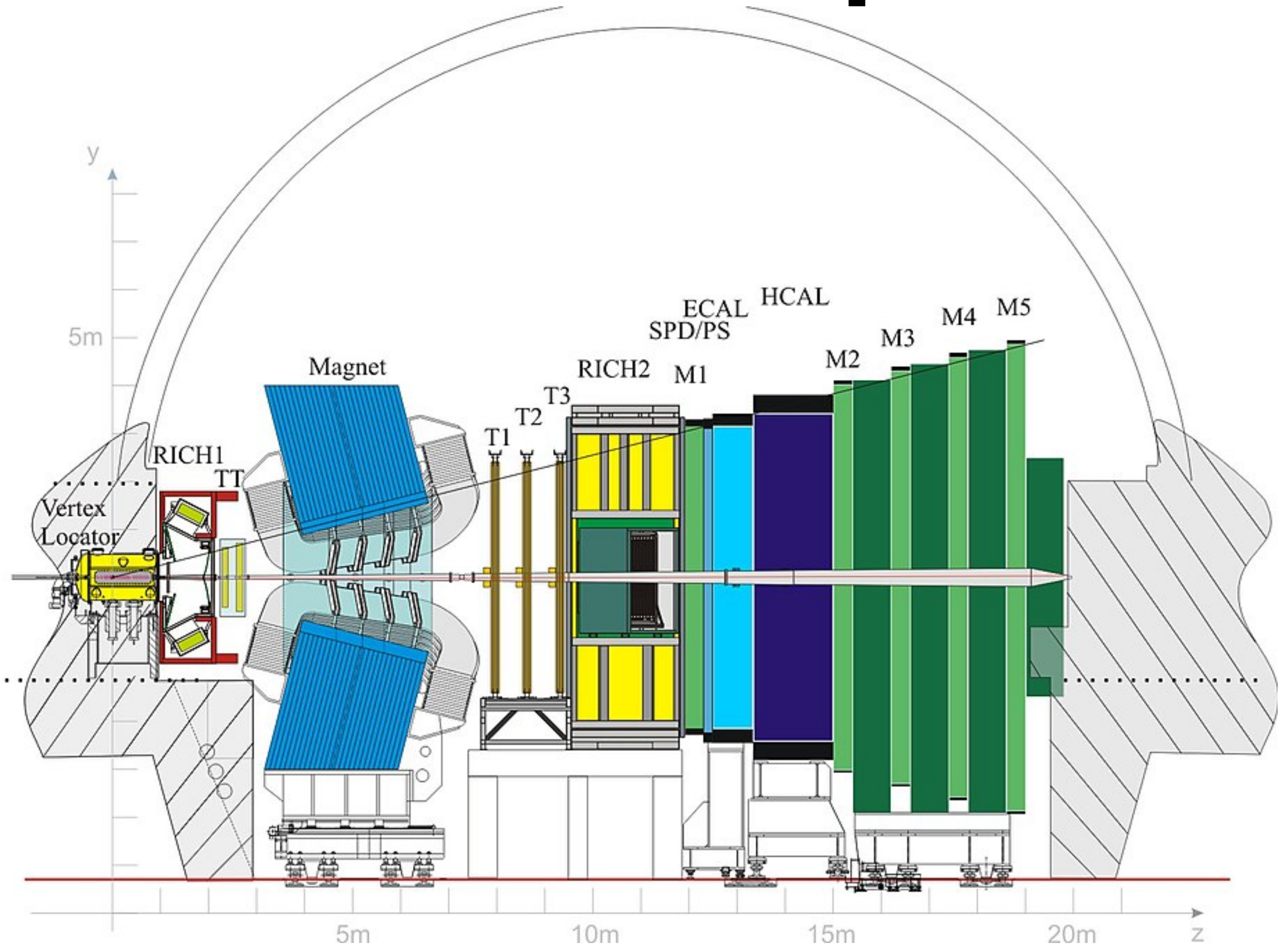
➡Make them interact with material and convert the interaction to electronic signals!

- Different types of particles have different interactions with material



typical experiment layout in layers of specialised detectors

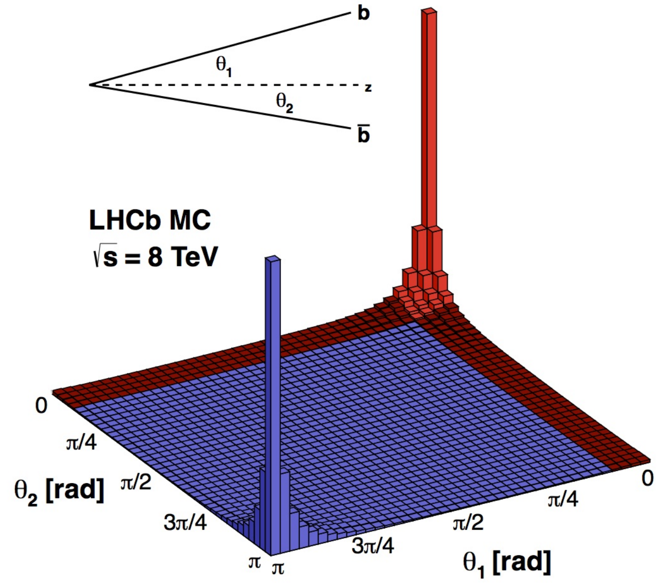
The LHCb experiment



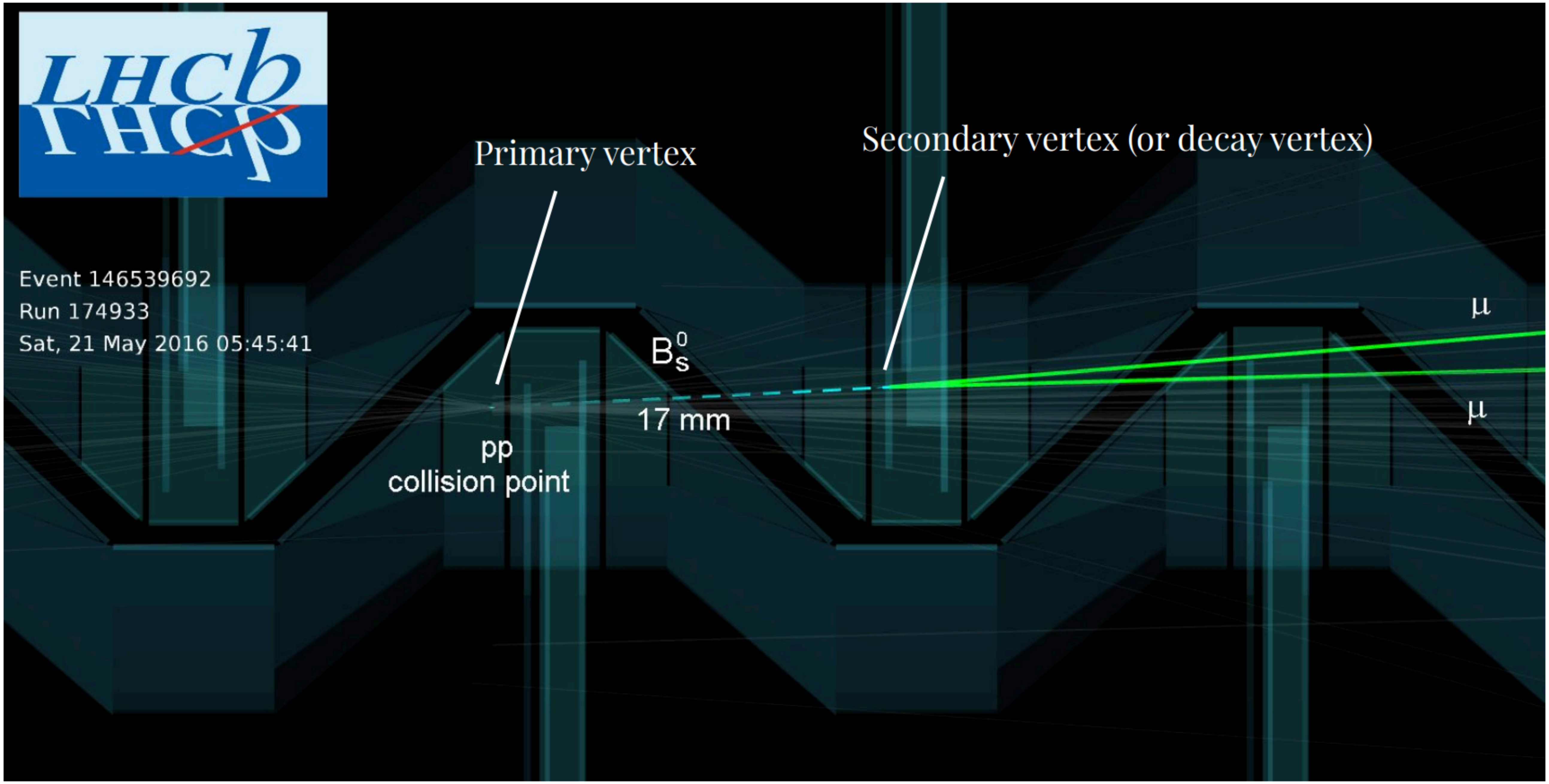
We saw in Yasmine's course why flavour physics is interesting… and especially why we care about the b quark

LHCb is a forward spectrometer in the LHC, specialised in the measurements of b- and c-quarks!

- High precision vertexing & tracking systems
- Complemented with excellent PID
- Collected 9 fb^{-1} of physics-worthy data in Run 1 (2011-2012) and Run 2 (2015-2018)

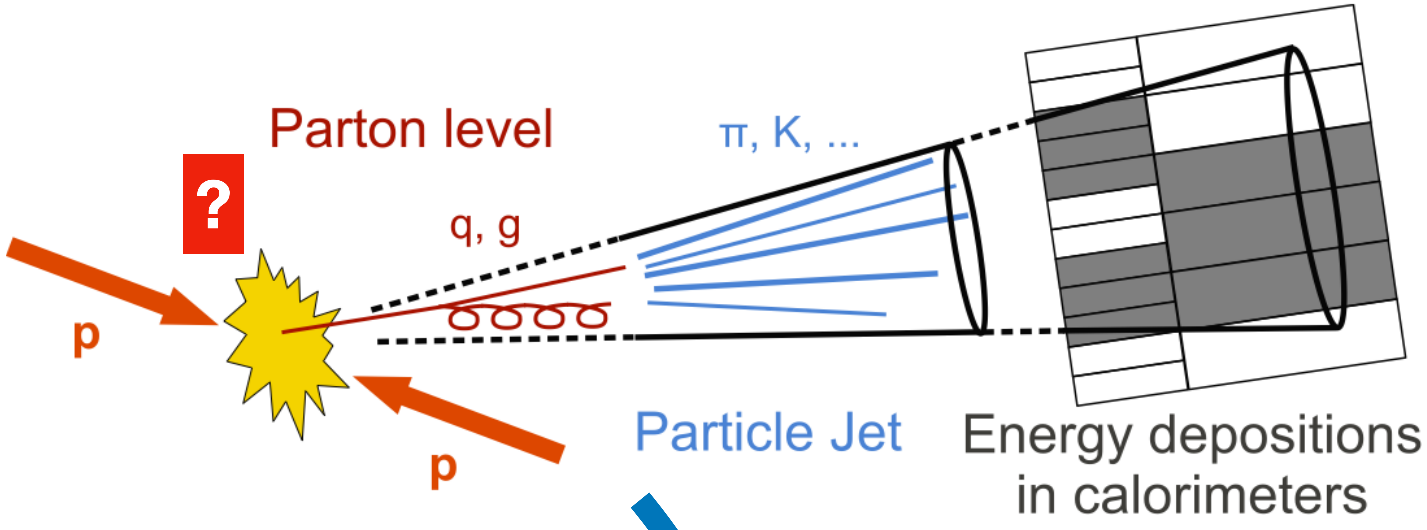


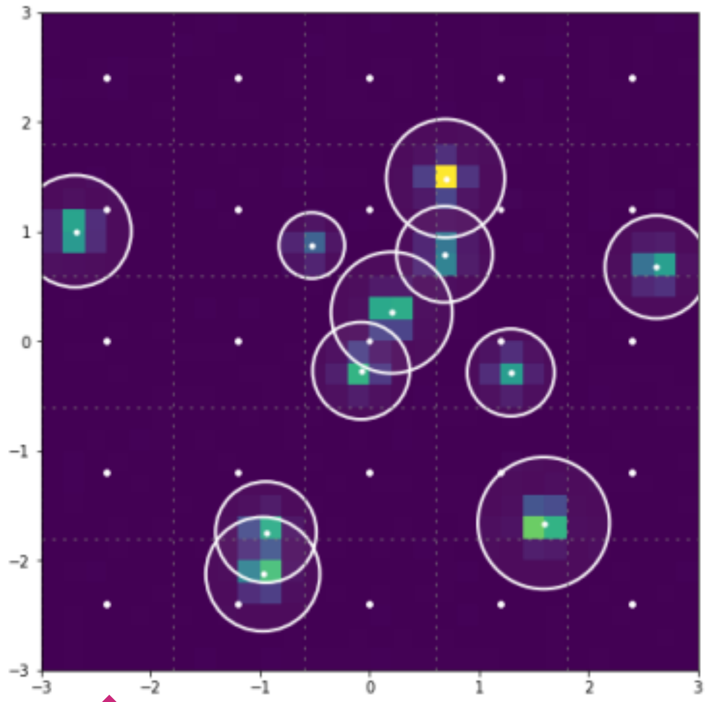
A typical LHCb event

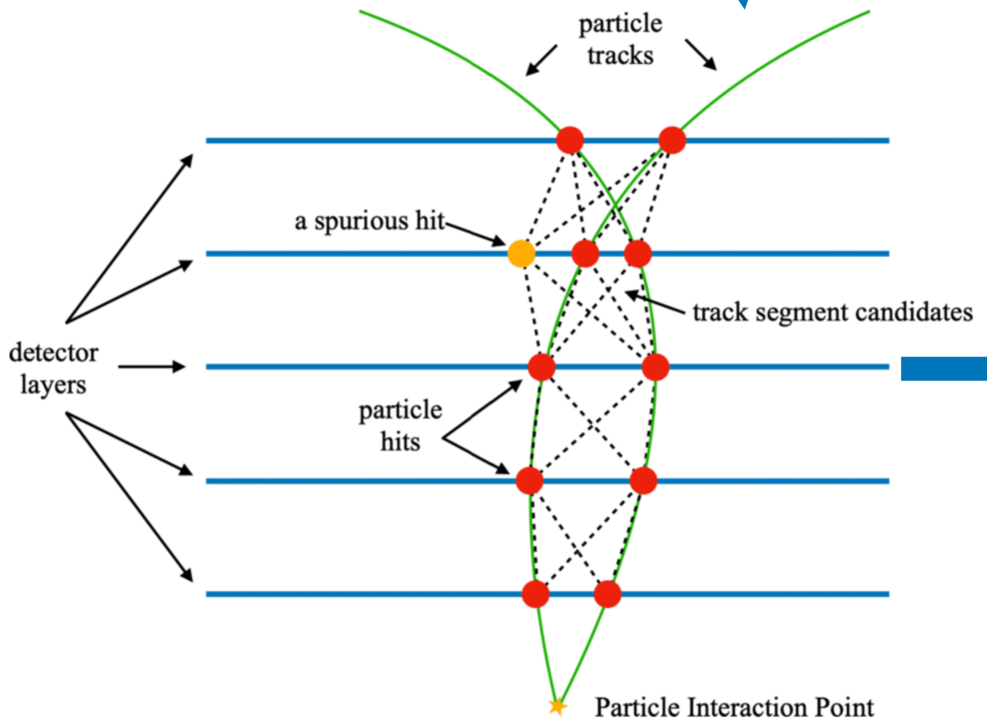


From detector signals to particles

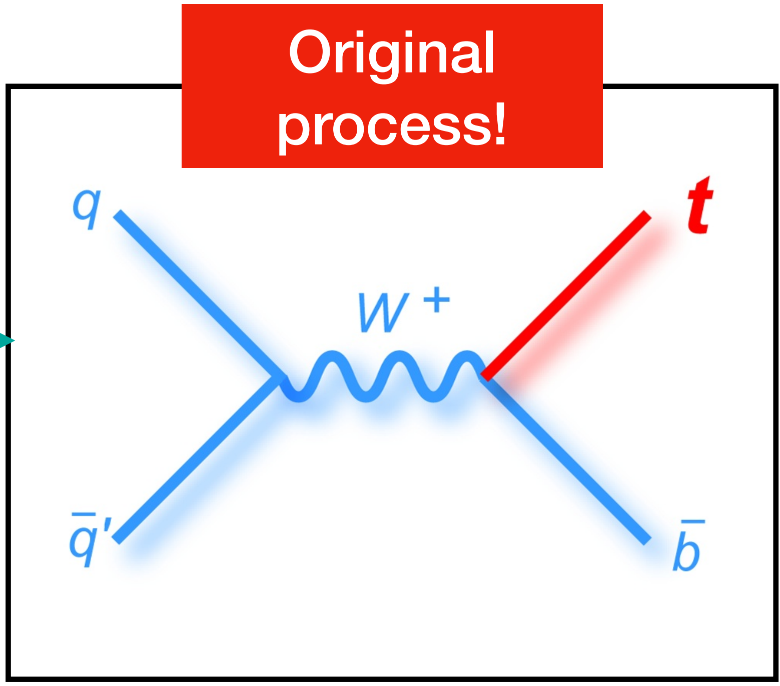
We collect raw detector signals… how do we get to particles?







Particle reconstruction


Original process!

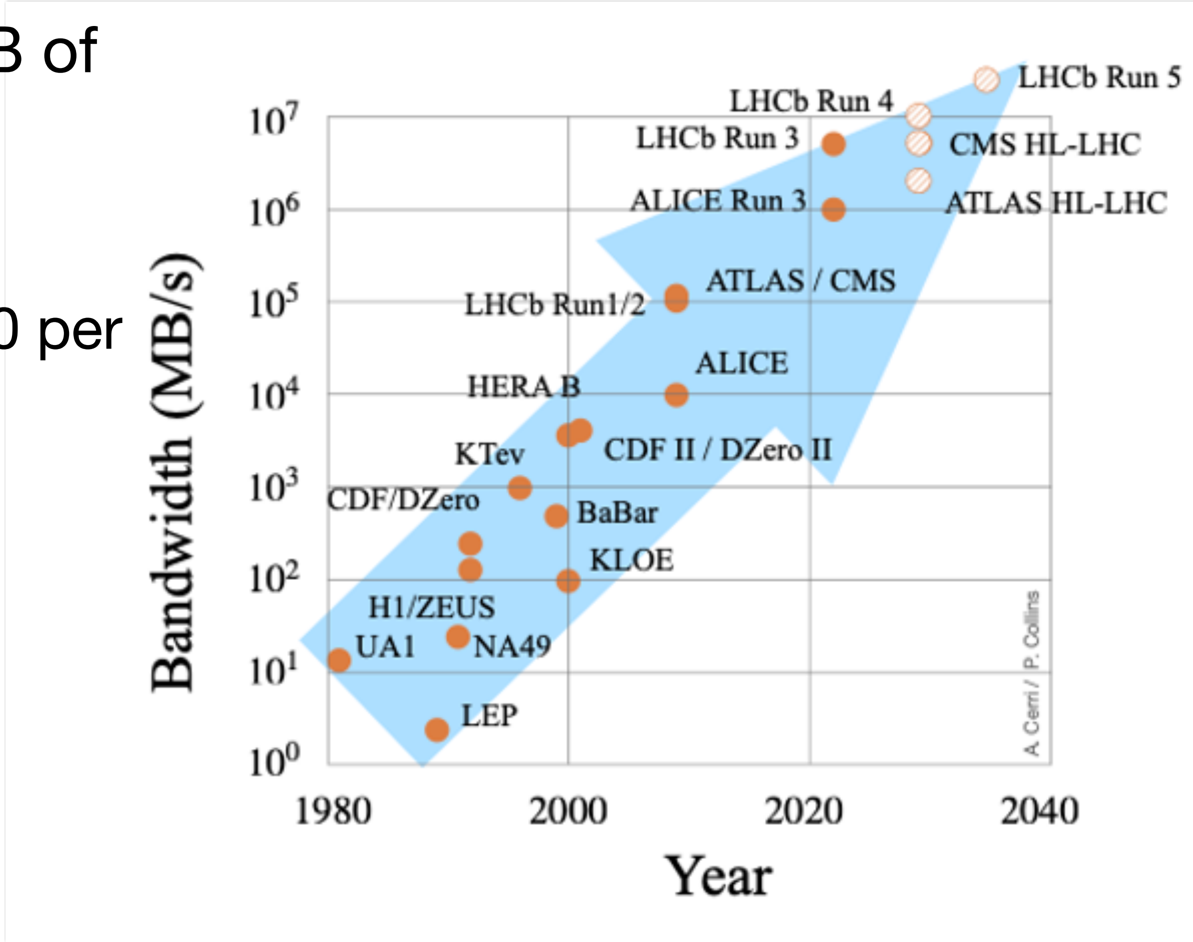
Particle reconstruction via pattern recognition algorithms!

How do we get the data

Not every proton-proton collision is the same: **different interactions & decays have different frequencies**

- At LHC, we get p-p interactions every **25 nano-seconds** - this allows us to probe some of the rarest interactions in nature

- But it also means **A LOT OF DATA** : around one PB of collision data per second
  - Comparable to the **traffic of a streaming platform**
  - My laptop has 1 TB of storage - I would need x1000 per second to store this information
  - **Not all the collision data are interesting!**



- Experiments put in place a fast filtering system to discard non-interesting data: **trigger**
  - we typically keep only 0.001-0.01% of all data

We still generate around 20 PB/year that we need to store and analyse

Experimental particle physics is a big-data domain

Lifetime of a typical (LHCb) analysis

Find an interesting channel (and its normalisation if needed)

Study how you can find it in your favorite detector/ experiment

Make trigger lines*

Collect some data!

Lifetime of a typical (LHCb) analysis

Find an interesting channel (and its normalisation if needed)

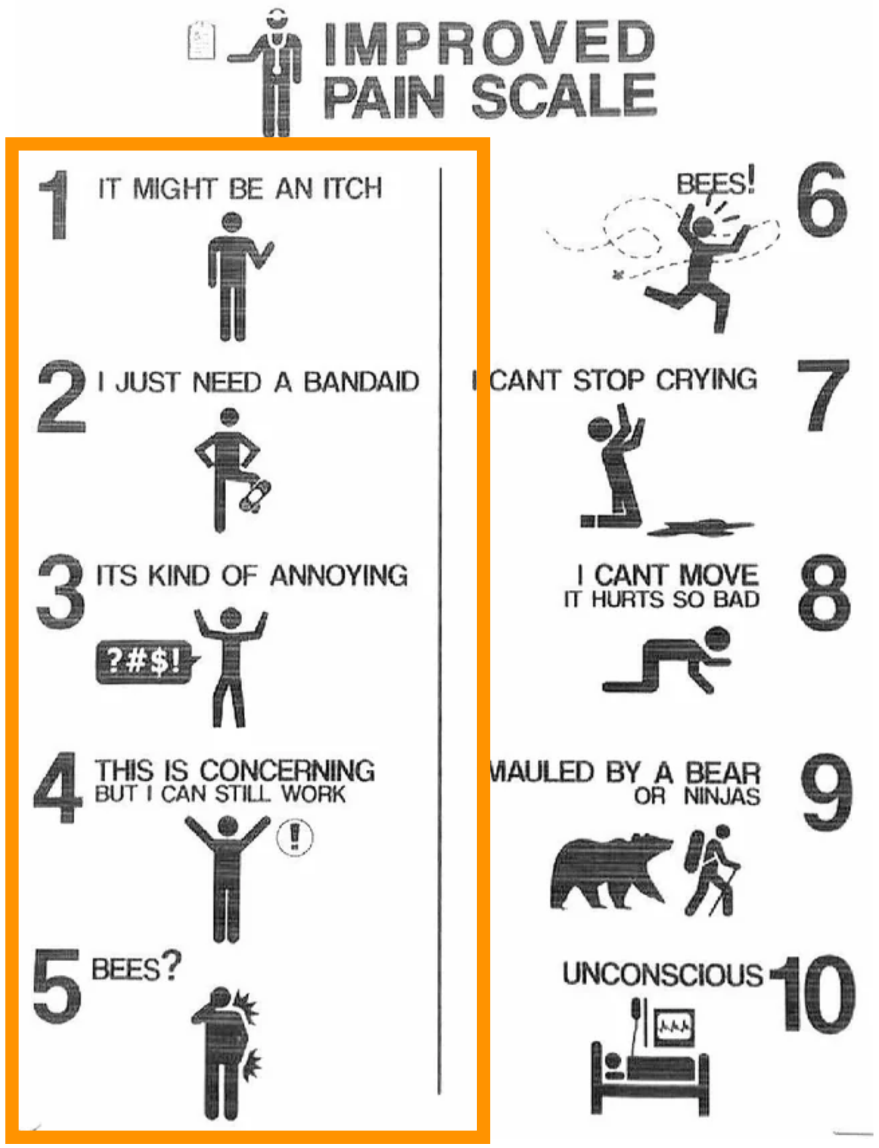
Study how you can find it in your favorite detector/ experiment

Make trigger lines*

Collect some data!

Offline selection

Fit



Lifetime of a typical (LHCb) analysis

Find an interesting channel (and its normalisation if needed)

Study how you can find it in your favorite detector/ experiment

Make trigger lines*

Collect some data!

Offline selection

Fit

Data/MC reweighing

Cross-checks

Toys

Efficiency calculations

Systematic uncertainties

Unblinding

Lifetime of a typical (LHCb) analysis

Find an interesting channel (and its normalisation if needed)

Study how you can find it in your favorite detector/ experiment

Make trigger lines*

Collect some data!

Offline selection

Fit

Data/MC reweighing

Cross-checks

Toys

Efficiency calculations

Systematic uncertainties

Unblinding

Extract results & publish

What we will cover in this hands-on

We will study CP asymmetry in the B meson decays - in particular, our channel will be the $B^0_{(s)} \to K^+\pi^-$ **and** $\overline{B^0}_{(s)} \to K^-\pi^+$

We will use real LHCb data taken from Run 1 (2011 - 2012) - sorry they couldn’t be more recent, but latest data are not public yet, since the collaboration is still analysing them! We will focus on:

1) **Cleaning up the data (i.e. selection) -> Today**

2) **Fitting the data and extracting results -> Tomorrow**

We will use typical tools used in experimental particle physics, such as (py)ROOT, TMVA & RooFit - instructions to setup the appropriate environment can be found here

* **Disclaimer:** there are many more steps in a particle physics data analysis, such as data/mc weighting, efficiency calculation, systematic uncertainties etc - making a full analysis takes more than 4 hours ;) but I hope we all learn something new that we can apply to our work!

The material of these lectures is adapted from the "Lectures on common analysis techniques in flavour physics" course given at the IPHC Strasbourg by L. Capriotti, J. Cerasoli, G. Dujany, V. Lisovskyi and myself.

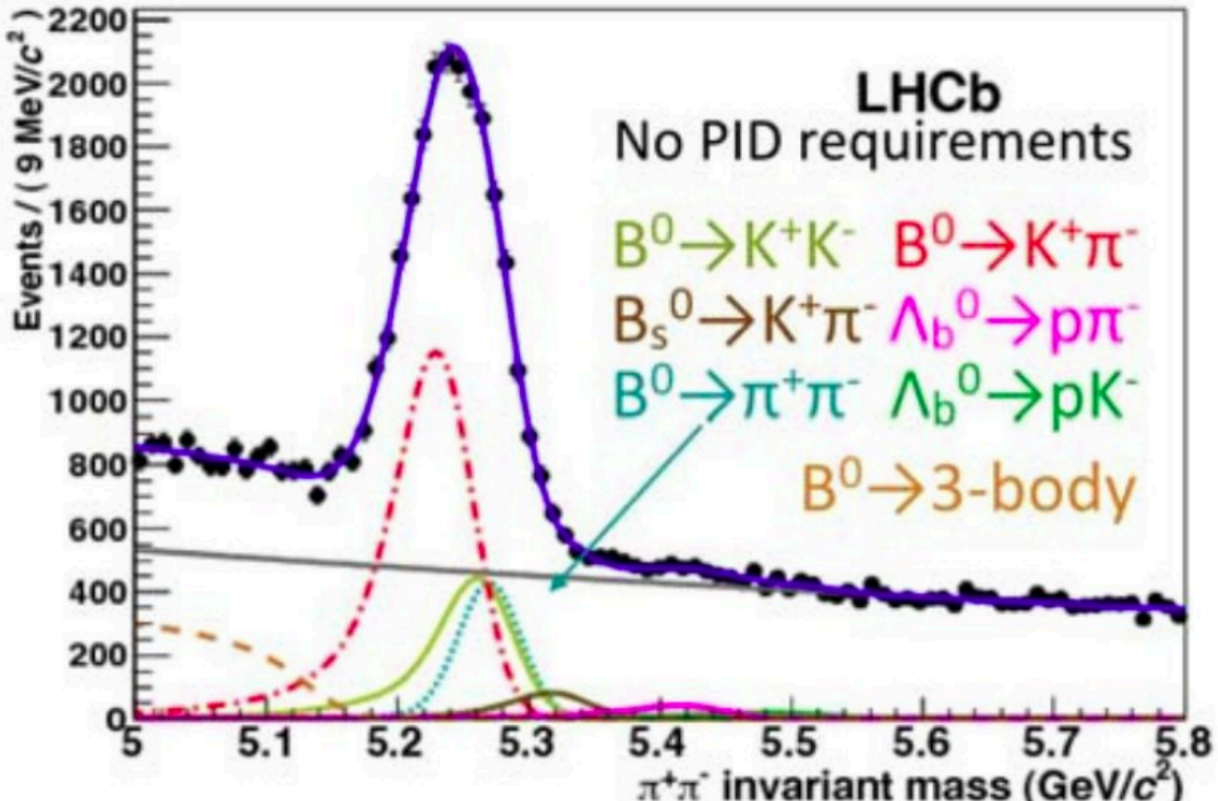
Cleaning up the data

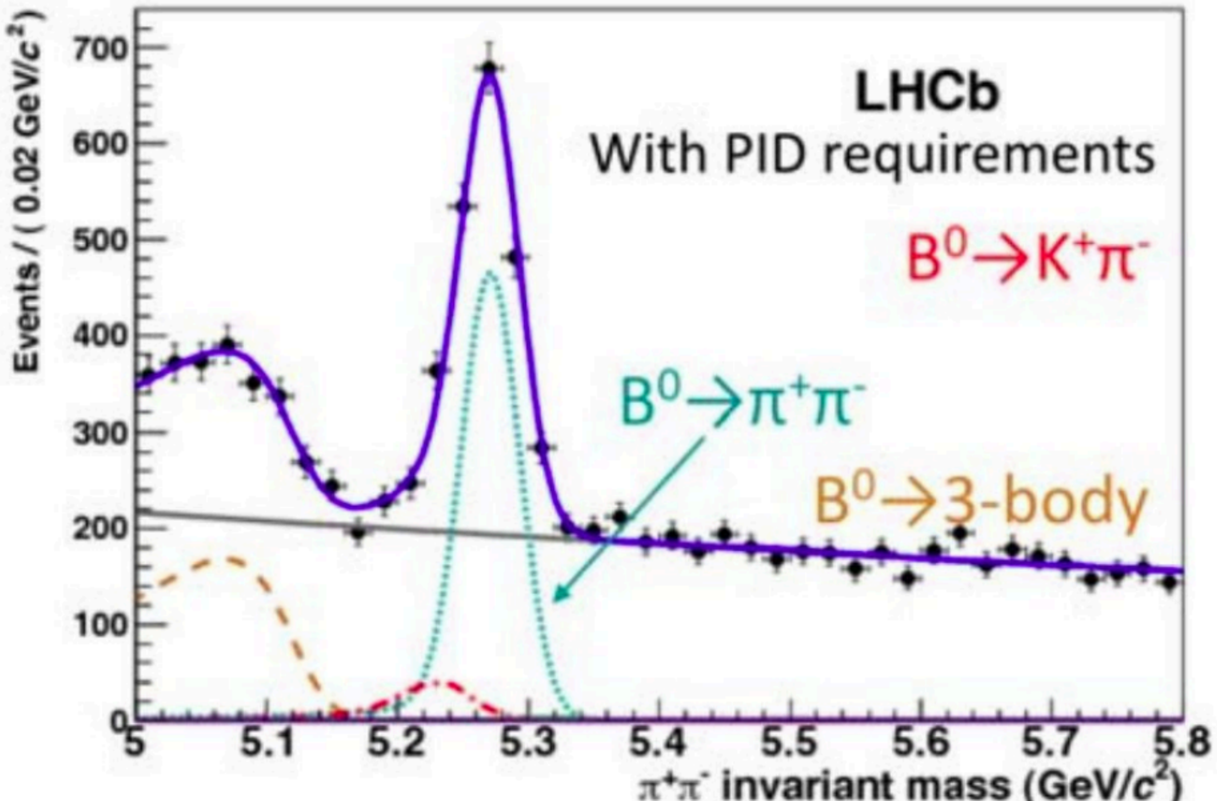
A p-p collider environment is quite busy… there are many processes that can fake our signal. In LHCb some typical ones are:

- **Combinatorial background:** random combinations of reconstructed objects with similar kinematics to our decay of interest. This background is usually non-peaking

- **Mis-identified background:** our particle identification algorithms make mistakes sometimes… in that case, the invariant mass combination will be off, and we may have some resonances from similar processes near our signal

- **Partially reconstructed background:** our tracks may be part of a more complex decay. If we miss the rest of the decay, we get this type of background, which typically gives a broad structure on the left-hand side of our signal (since we're missing some energy)

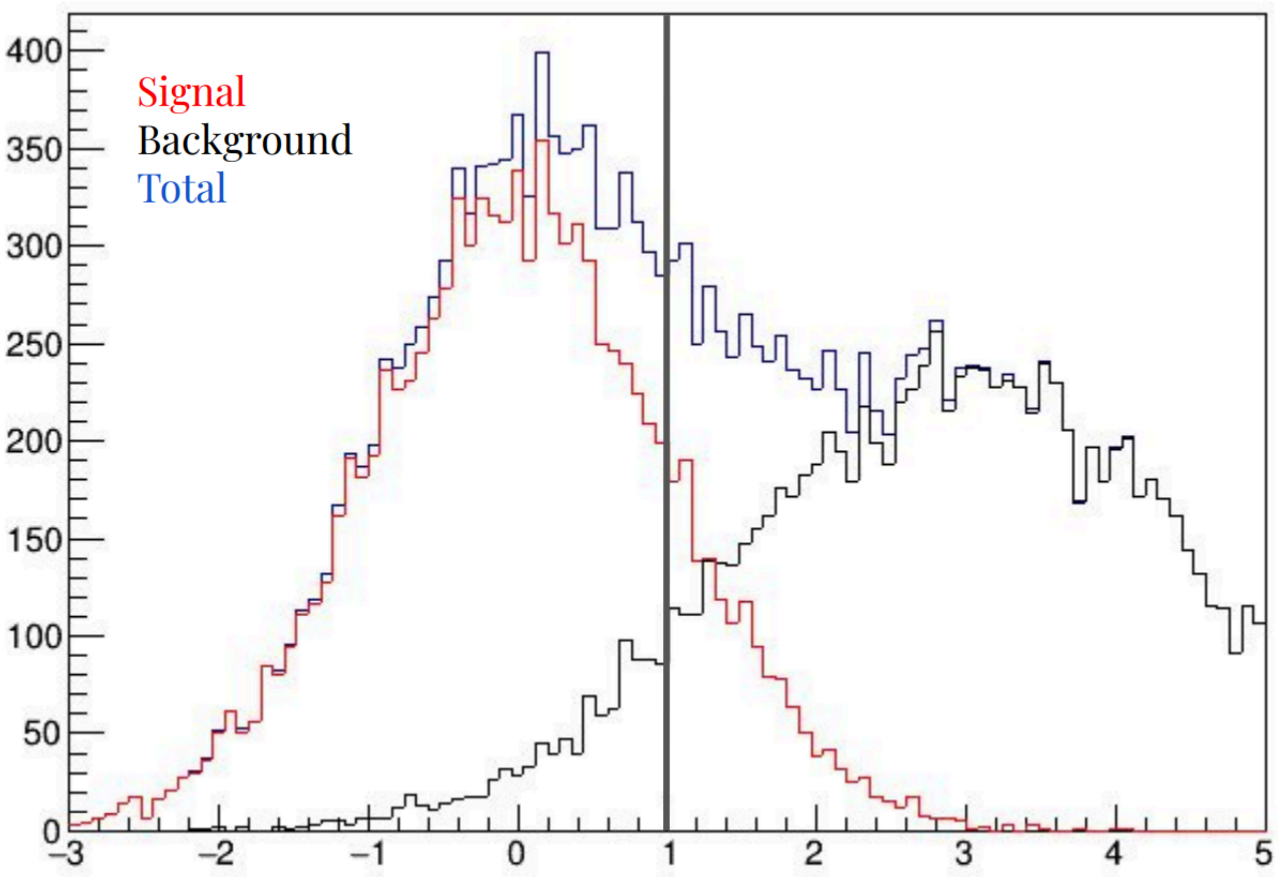


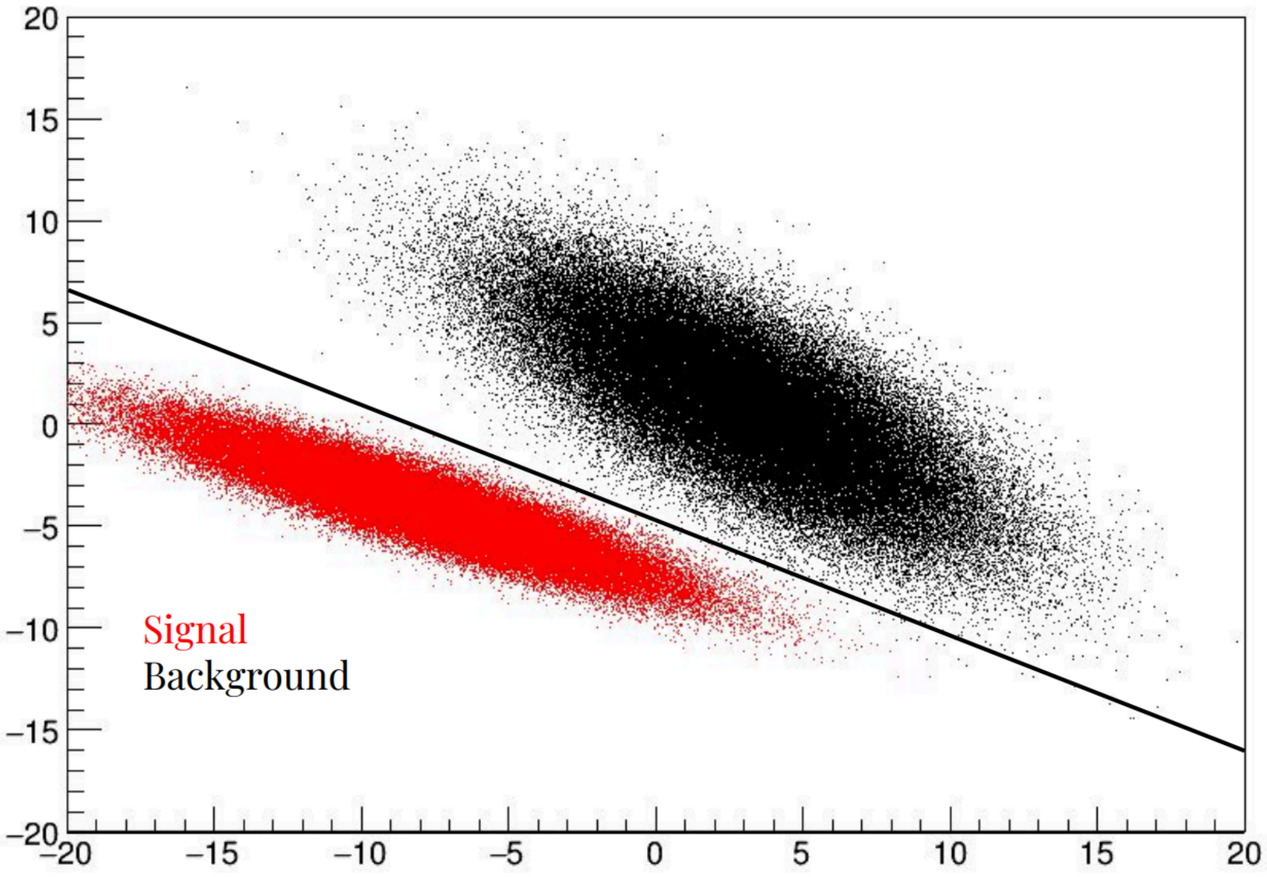


Cleaning up the data

We want to clean up our data, to remove as much background as possible, while maintaining enough signal statistics… we typically do this in two ways

- Cut-based selection: find variables that give discriminating power between our signal and background and apply cuts to remove the background. The cuts can be 1-D but also multi-dimensional



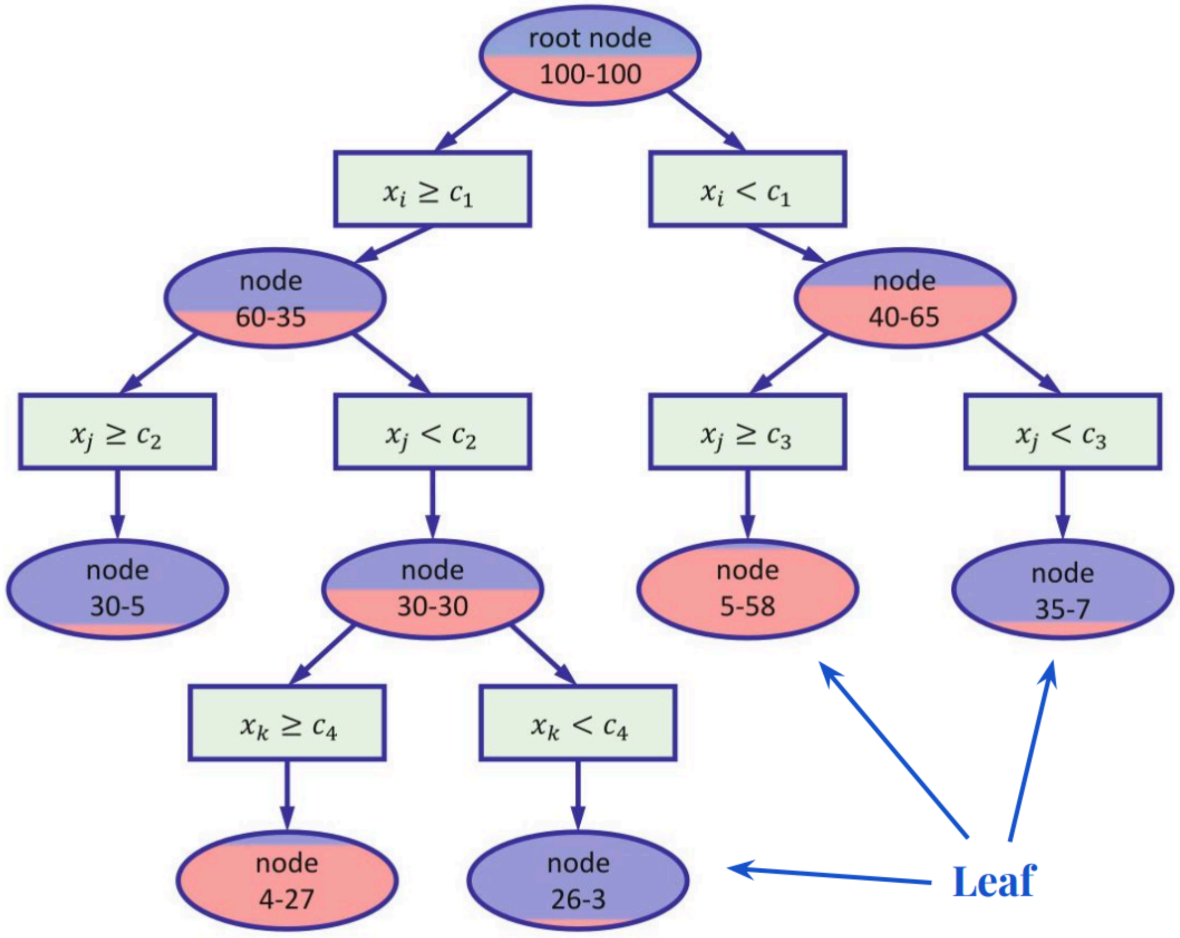


Cleaning up the data

- But we have a lot of variables! It can quickly become very complex and time consuming to study the relations between all the different variables…
- This is where ML can help us: we can train algorithms to learn complex multivariate patterns in our signal, and use them to discriminate against background
- We will use one such approach in this hands-on: the **Boosted Decision Tree**
  - One of the oldest, and most commonly used ML methods in particle physics

Main principles of the BDT

- It is based on decision trees, which is a series of cuts on randomly selected variables. The cuts are chosen to optimise a FoM
- One tree is only series of cuts… but what if we have many? We can boost the performance by creating a **Random Forest of many decision trees**
- If we use trees iteratively which are optimised based on the decisions obtained in the previous iteration, we **boost** our algorithm
- And this is how we end up with the **Boosted Decision Tree**



Enough talking, let’s get coding!