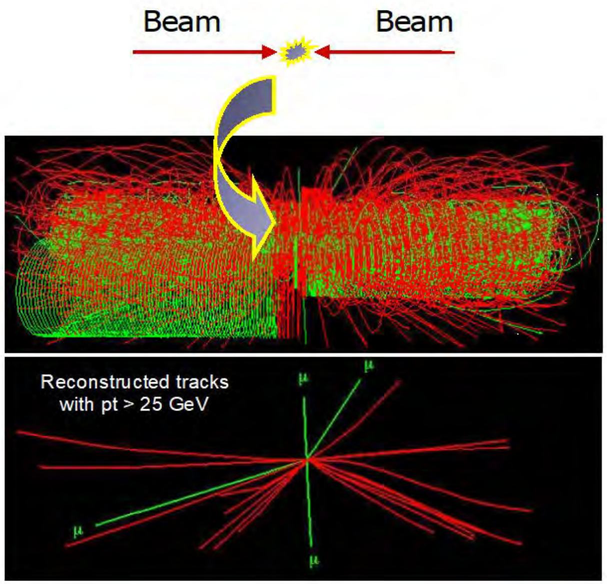
Event Reconstruction in High-Energy Physics Experiments

S. Shmatov, **N. Voytishin**

20–25 Oct 2024
Yerevan, Armenia
Asia/Yerevan timezone

MMCP

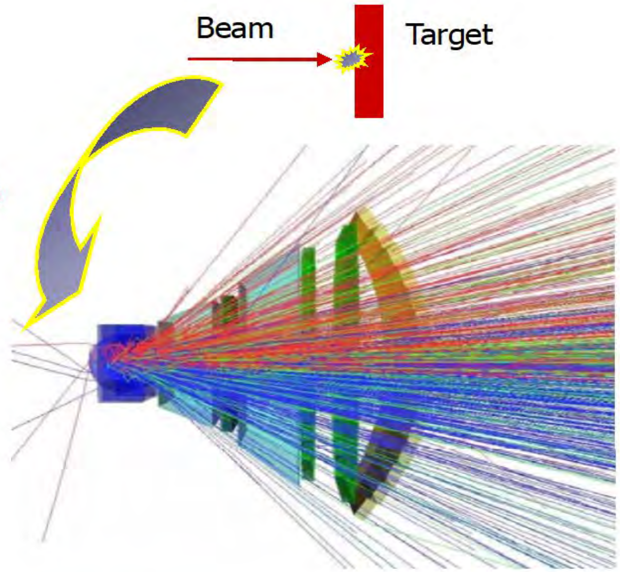
HEP Experiments: Collider and Fixed-Target



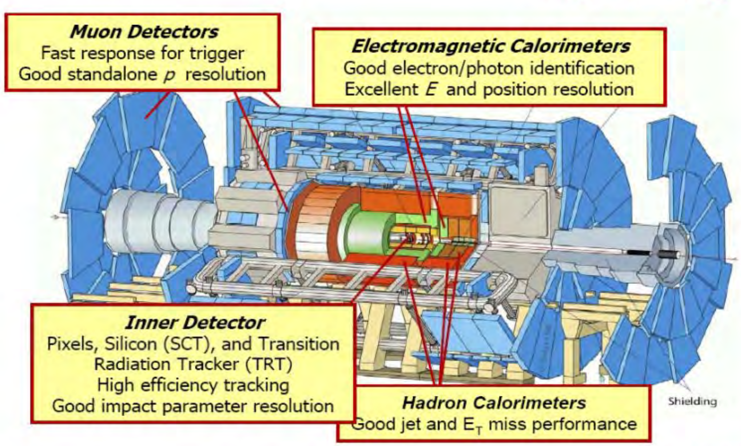
Inelastic collisions
$10^7 - 10^9$

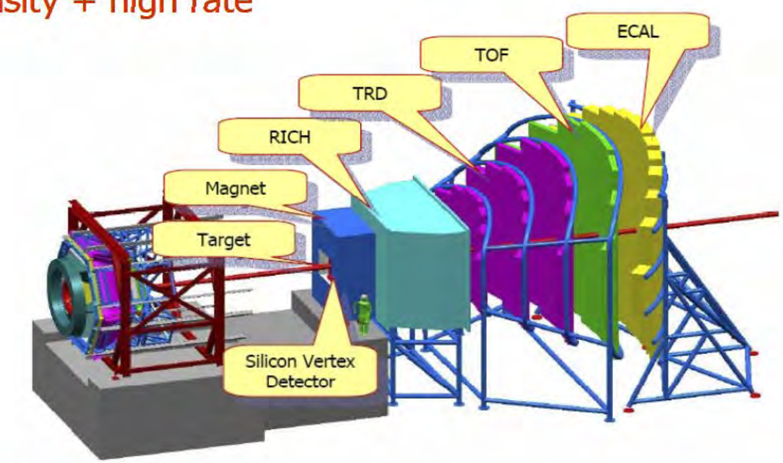
10^{11}

Signal events
$10^2 - 10^{-2}$



High energy = high density + high rate



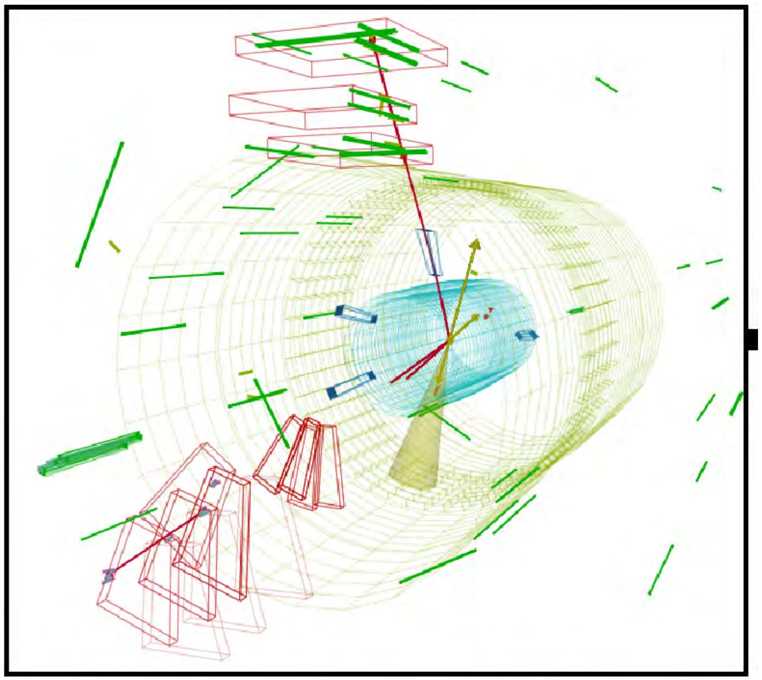


ATLAS (CERN)

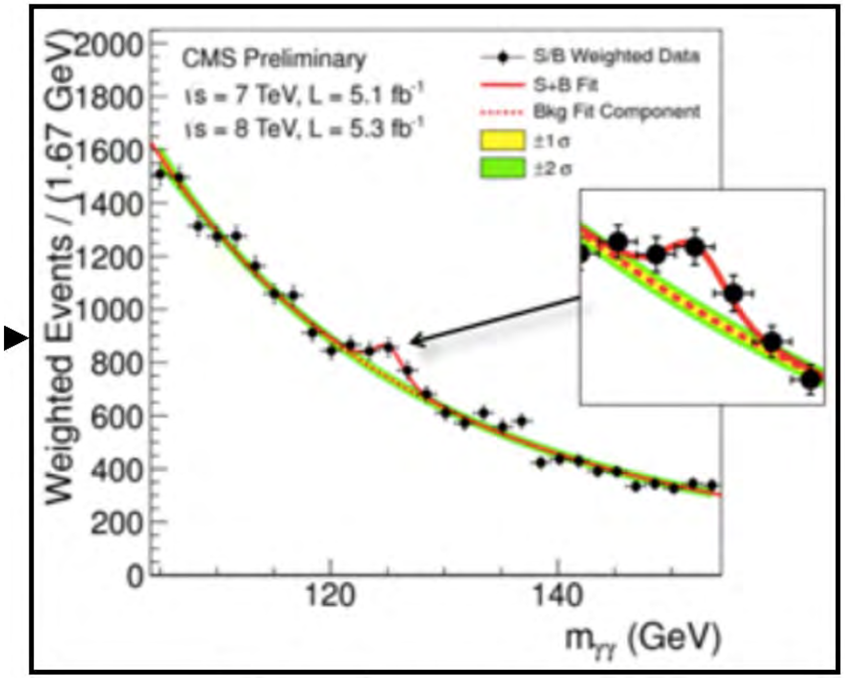
CBM (FAIR/GSI)

What just happened?

- HEP experiments can produce O(10) petabytes of data per year (LHC case).
- Data is processed to the stage of physics papers → measurements and discoveries.



Example collision event from CMS

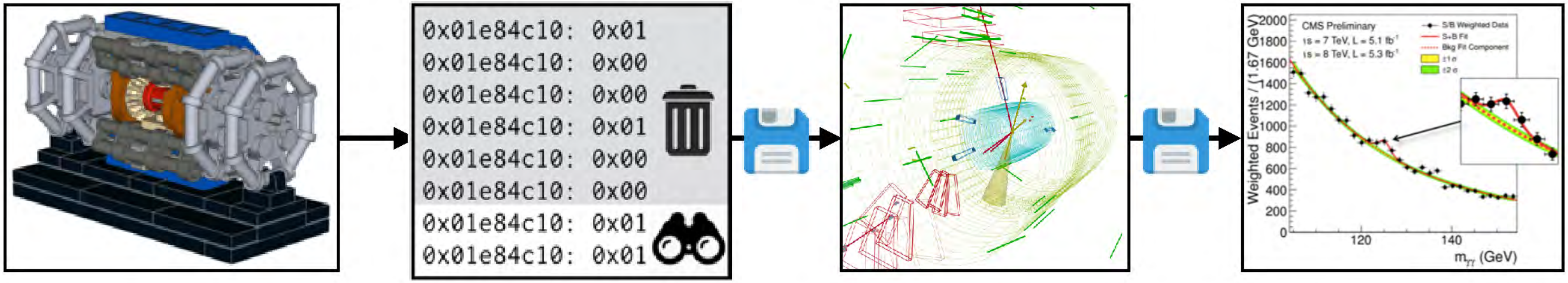


Higgs discovery at CMS

- Many steps involved
- Each step has computing costs, varying inefficiencies, often in large backgrounds.

Data Flow

Data reconstruction generally involves several steps of processing and reduction:



Stage	Trigger	Event Reconstruction	Stripping (AKA Skimming)
Description	Initial selection for finding interesting events.	Reconstruct triggered data into list of particles.	Signature selection trained by prior physics knowledge.
Hardware Implemented	Local electronics or CPU/ GPU processing farm.	Inside trigger and/or the Grid (see later).	The Grid.
Timescale	Live.	Almost live (requires detector calibration). Repeated ~yearly.	Any point, ~monthly turn around.
Data reduction factor	10^{6*} (permanent loss).	10x (used for Physics).	Analysis dependant.

This talk

Event Reconstruction

- Triggered detector collision data → particle interactions.
- Seek the following information as input for physics analysis
  - What particles were created?
  - Where were they produced?
  - What were the parent particles?

To find this, perform

- **Tracking**: Reconstruct particle trajectories into tracks.
- **Vertexing**: Group particles into vertices.
- **Particle ID**: Find the particle identification of each track (e.g. a muon, electron etc.).

Requirements for reconstruction algos:

- Fast
- Good quality (enough for physics analysis)

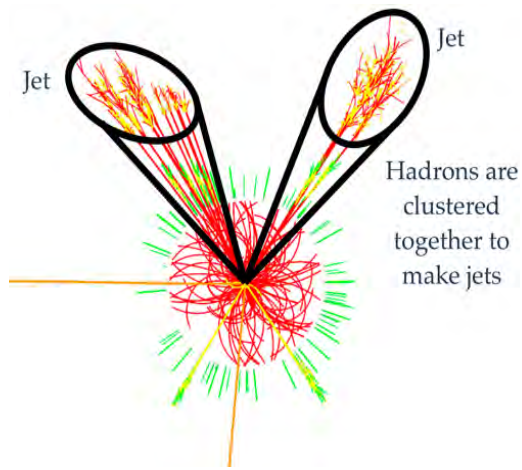
Usually anti correlated - a fast algorithm often leads to inefficiency and impurities (see later).

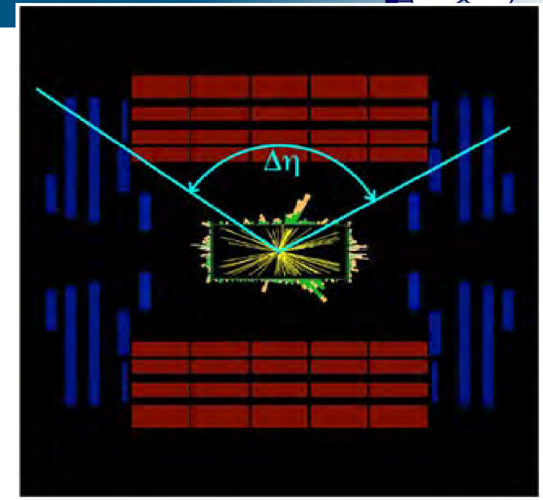
Trigger Bias (not everything depends from reco-algo)

- Data sets from triggers inevitably biased by trigger. E.g. experiment finds deficit Higgs candidates with ET < 5 GeV (unsurprising if ETTrig = 5 GeV).
- Can be accounted for:
  - ✓ Comparisons with simulation, many factors (detector performance, collider conditions).
  - ✓ Comparison with non-triggered data: Far lower rate! Have to extrapolate.

Physics Objects

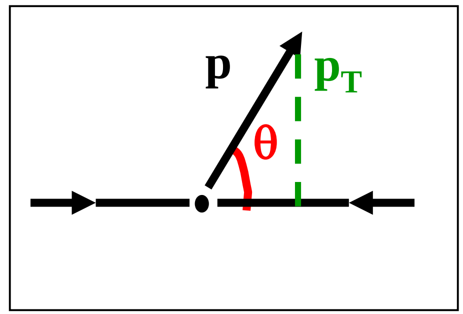
- Muons (transverse momentum p_T)
- Electrons (energy and tr. momentum p_T)
- Photons (energy)
- Jets (energy and coordinates)
- Unstable Particles
- Missing energy and p_T
  - vectorial sum of all transverse momentum

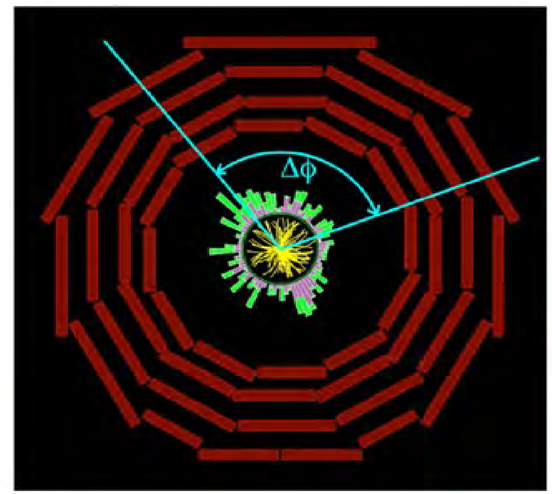




Kinematic Variables

- Transverse momentum p_T (energy)
  - particles that escape detection have p_T=0
  - total visible p_T = 0
- Longitudinal momentum p_Z and energy E_Z
  - particles that escape detection have p_T=0
  - visible p_Z is not conserved (not so usefull variable)





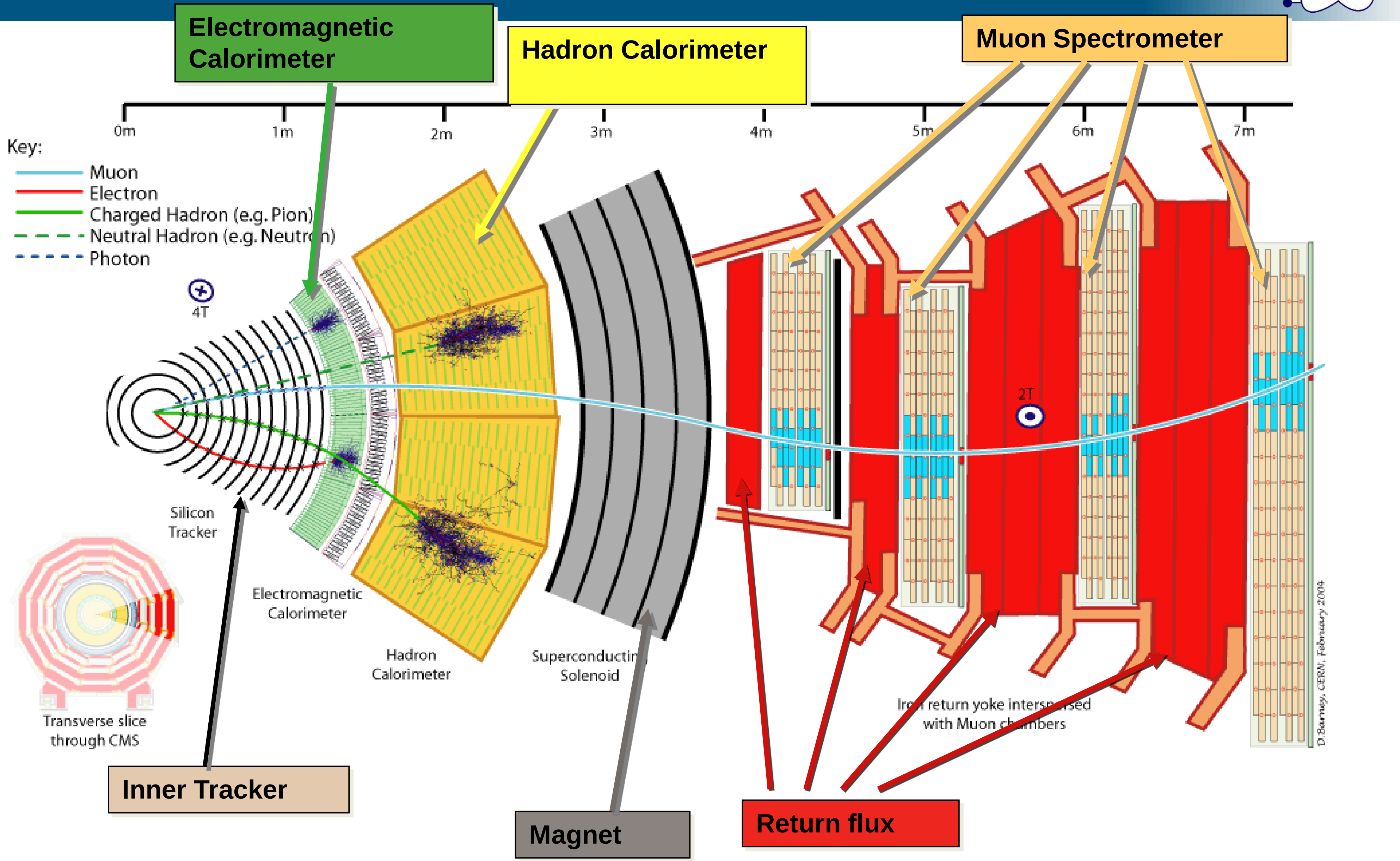
- Angles
  - azimuthal and polar angles
  - polar angle θ is not Lorenz invariant ⇒
  - rapidity y
  - or (or m=0) pseudorapidity η

$$y \equiv \frac{1}{2}\ln\left(\frac{E+p_z}{E-p_z}\right)$$

$$\eta = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$$

4π-experiments cover 360^0 over ϕ and large pseudorapidity range, |η| ≤ 5.0 (0.8^0)

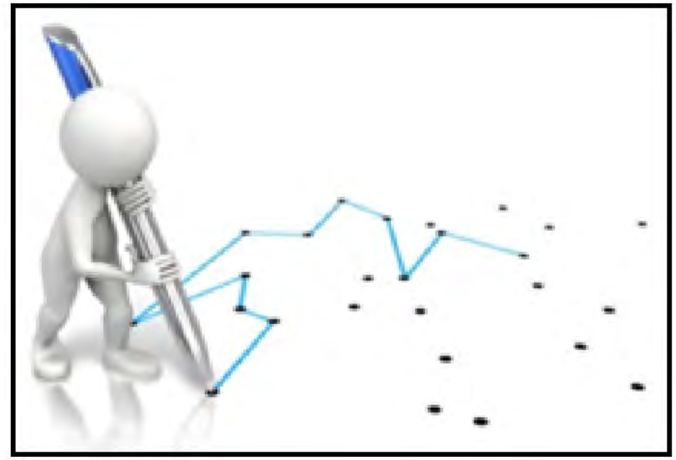
Particles in Detectors



Tracking Algorithms

Tracking particles through detectors involves two step

- Pattern recognition: identifying which detector hits for a track.
- Track fit: approximate the path of the particle with an equatior

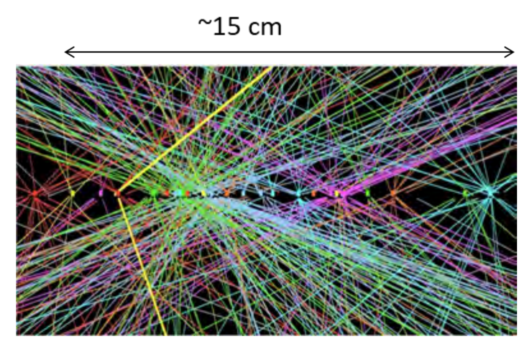


- No one size fits all solution.
- Many detectors use different combinations of algorithms (e.g. LHCb uses 4 different algorithms for different combinations of sub detectors, but basic ideas are the same). Usually a trade off between
  - ✓ Efficiency: fraction of real tracks found
  - ✓ Purity: fraction of tracks that are real
  - ✓ Computational speed.

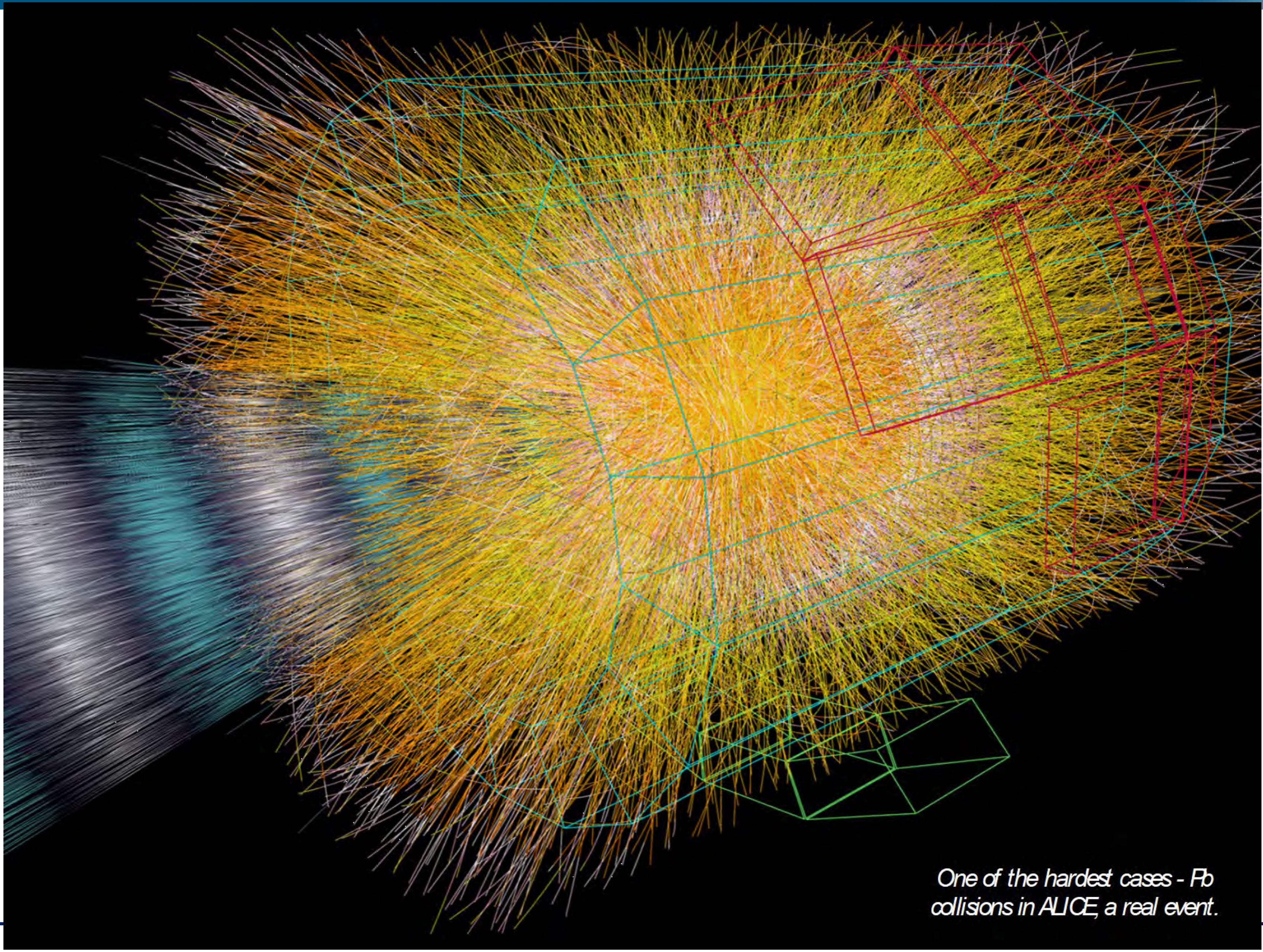
Typically these two are anti correlated: a good efficiency typically has a bad purity, and vice versa. Both good efficiency and purity is usually computationally expensive - see later.

Reconstruction conditions:

- high multiplicity and density of flying charged particles
- high collision rate
- high data flow density
- the presence of massive layers of matter – calorimeters, magnetic yoke...
- pile-up

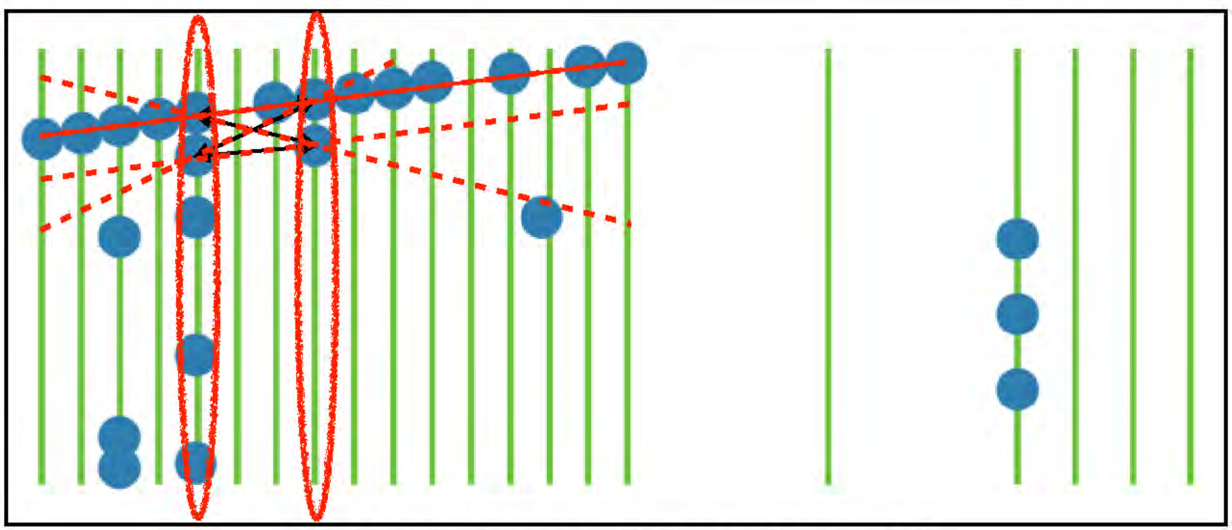


Challenge



Tracking - Pattern Recognition

Name	Description	Scalability
Combinatorial	• Form every track from each possible combination. • Access each track by quality (e.g. ! ^{2}) and tag.	$n_{Tracks}!$
Hough Transform	• Transform points into a system where clusters form. • E.g. for straight tracks, take the difference between consecutive hits. • Group (e.g. in a histogram) and tag peaks.	x
Seeding	• Form seeds from pairs of hits on a sub set of the detector. • Extrapolate the seed and count hits intercepted. • Tag if sufficient number of hits.	nlog(n)



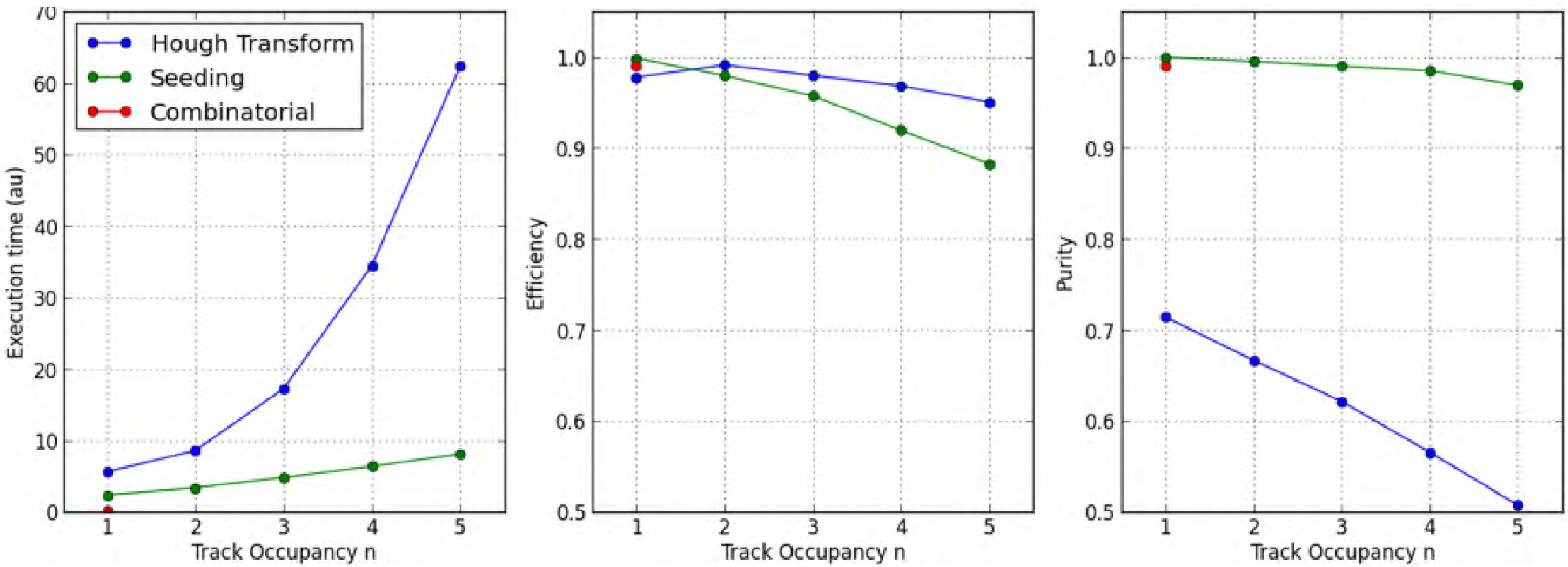
LHCb VELO data event (2d projection, top half)

Pattern Recognition Algorithms

Recall three main factors in choosing such algorithms:

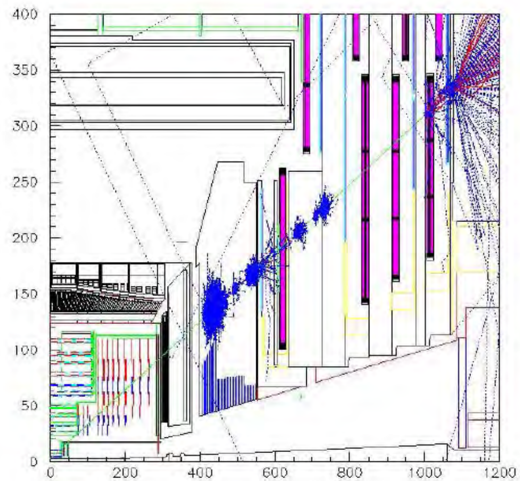
- Efficiency: fraction of real tracks found
- Purity: fraction of tracks that are real
- Computational speed

Toy simulation for LHCb VELO:



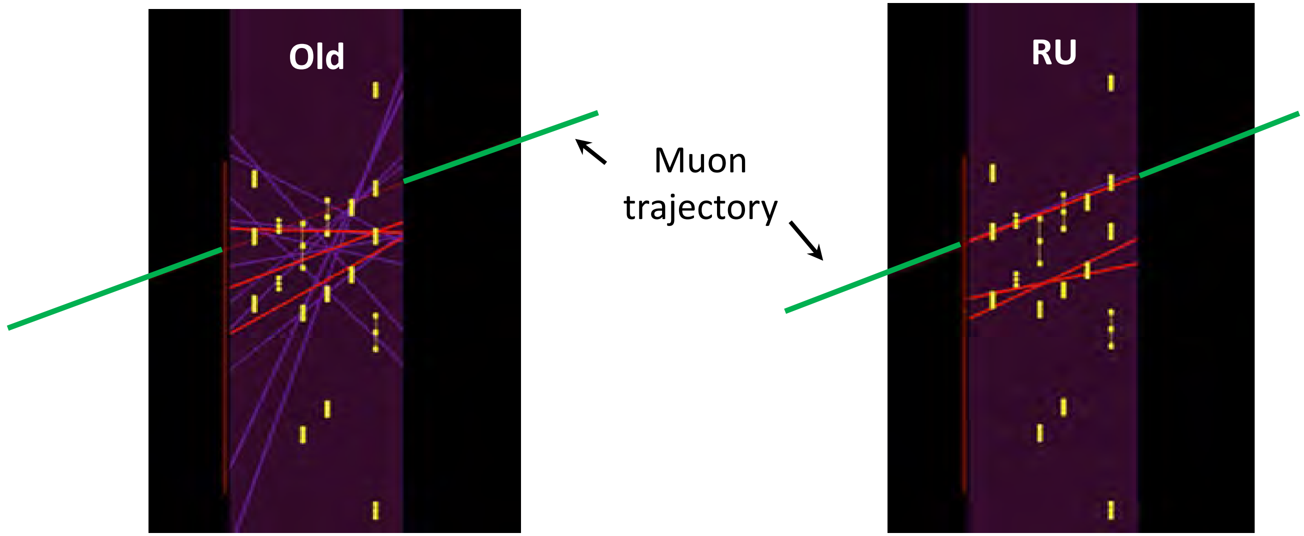
Typically use a combination of these algorithms

Reconstruction of high pT muon trajectories



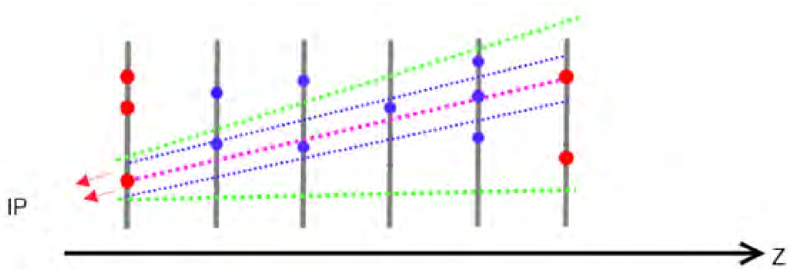
High pT muon passing through the CMS endcap muon system

84 hits in chamber

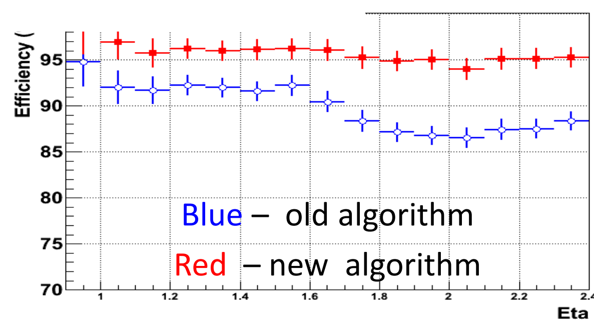


15 track-segments

4 track-segments



RoadUsage(RU) segment reconstruction algorithm

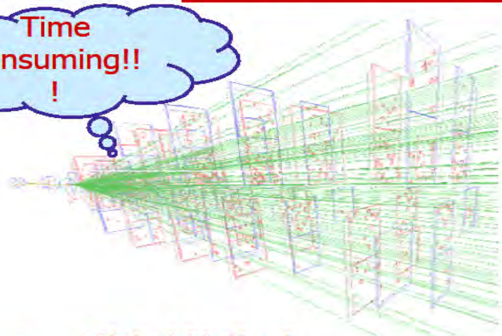


Reconstruction efficiency vs. pseudorapidity

Methods for Track Finding

1 Track finding

Time consuming!!!

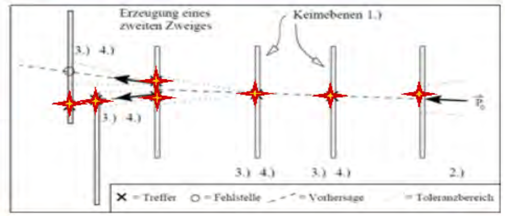


Combinatorics
+ Precision
= Speed ?

- Global Methods
  - all hits are treated equivalently
  - typical methods:
    - Conformal Mapping
    - Histogramming
    - Hough Transformation
- Local Methods
  - sequential selection of candidates
  - typical methods:
    - Track following
    - Kalman Filter
- Neural Networks
  - combine local and global relations
  - typical methods:
    - Perceptron
    - Hopfield network
    - Cellular Automaton
    - Elastic Net

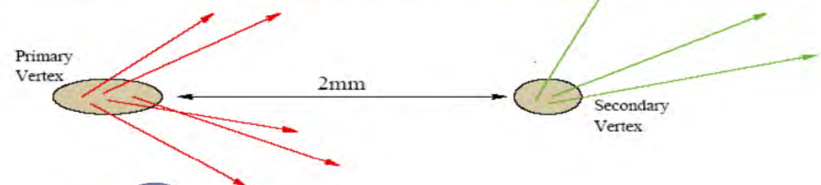
2 Track fitting

Kalman Filter



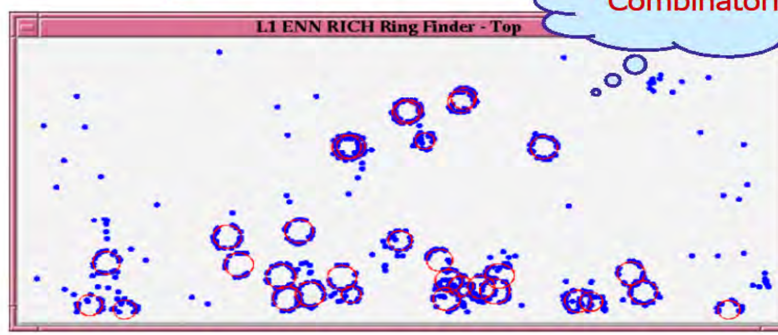
3 Vertex finding/ fitting

Kalman Filter



4 PID: Ring finding

Combinatorics

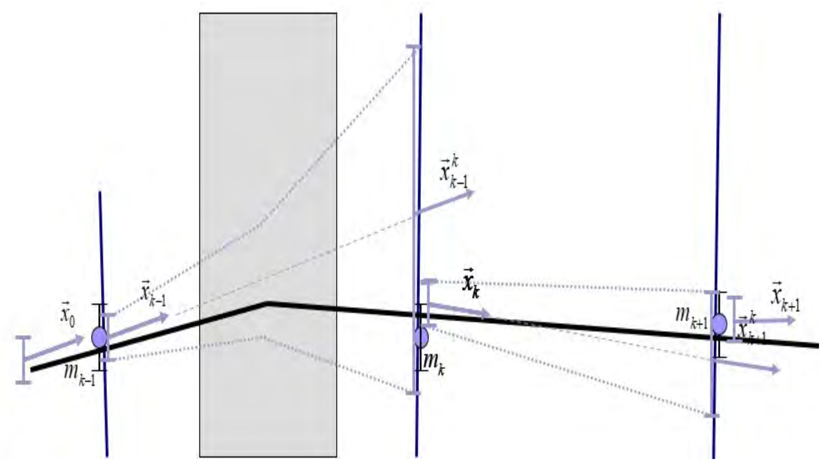


In details in the next talk by G. Ososkov

Kalman filter

Among the many tracking methods, the most effective was the method using the **Kalman filter**, since it allows one to easily take into account the non-uniformity of the magnetic field, multiple scattering and energy losses.

Kalman Filter (KF) – an efficient recursive filter that estimates the state of a **linear dynamic system** using a series of imprecise measurements.

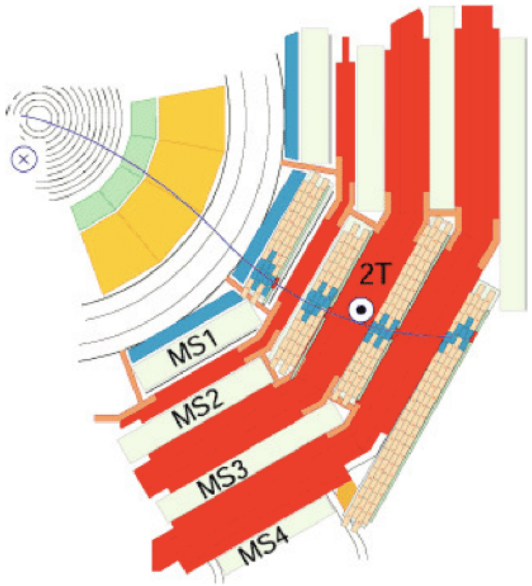


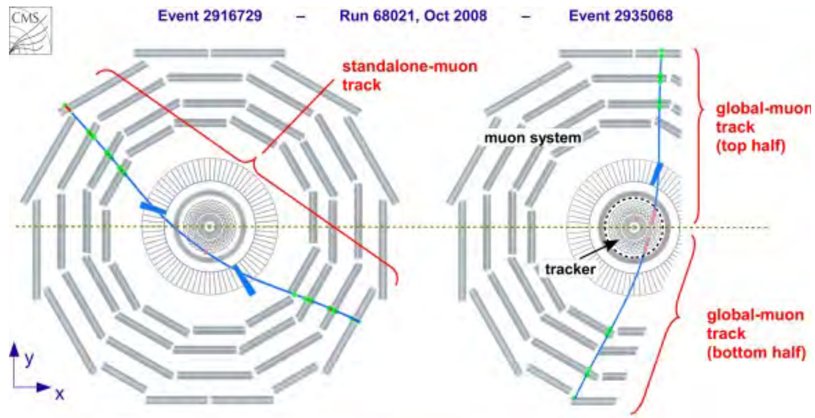
State vector $\vec{x} = (x, y, t_x, t_y, q/p)^T$ is iteratively evaluated to predict the track position on the next coordinate plane, taking into account the change in the covariance matrix and error corridors.

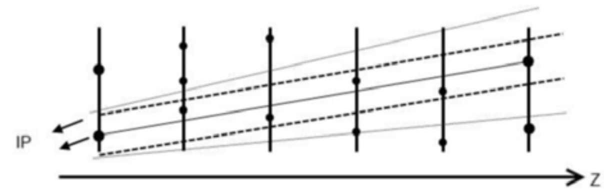
The main flaw of KF – the need to know the initial value of the state vector $\vec{x}$, seeding

Muon Track and Dimuons Reconstruction

CMS Muon System shows a excellent performance to detect different resonances

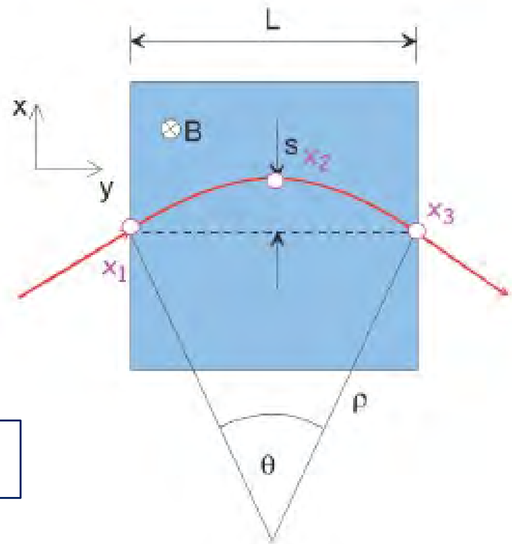


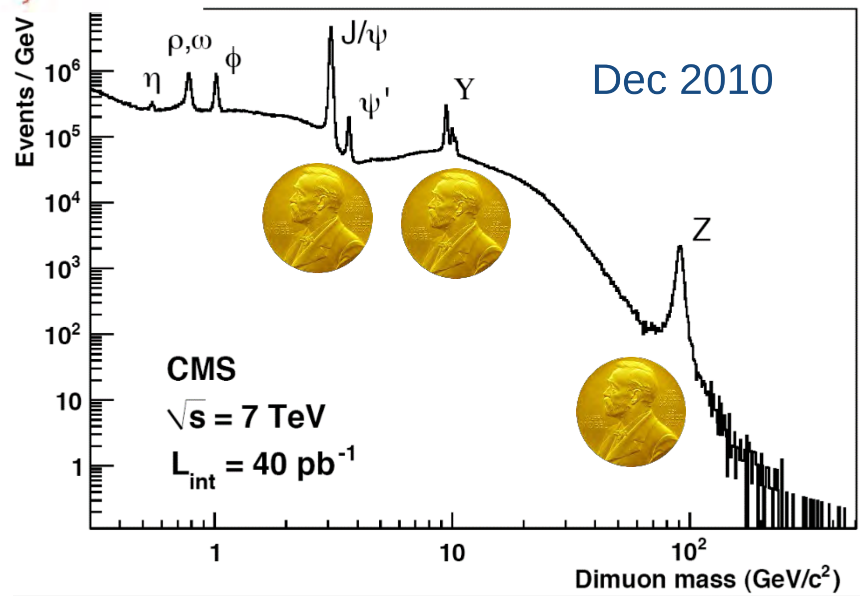




$$\frac{\sigma(p_T)}{p_T} = \frac{\sigma_x p_T}{0.3BL^2}\sqrt{\frac{720}{N+4}}$$

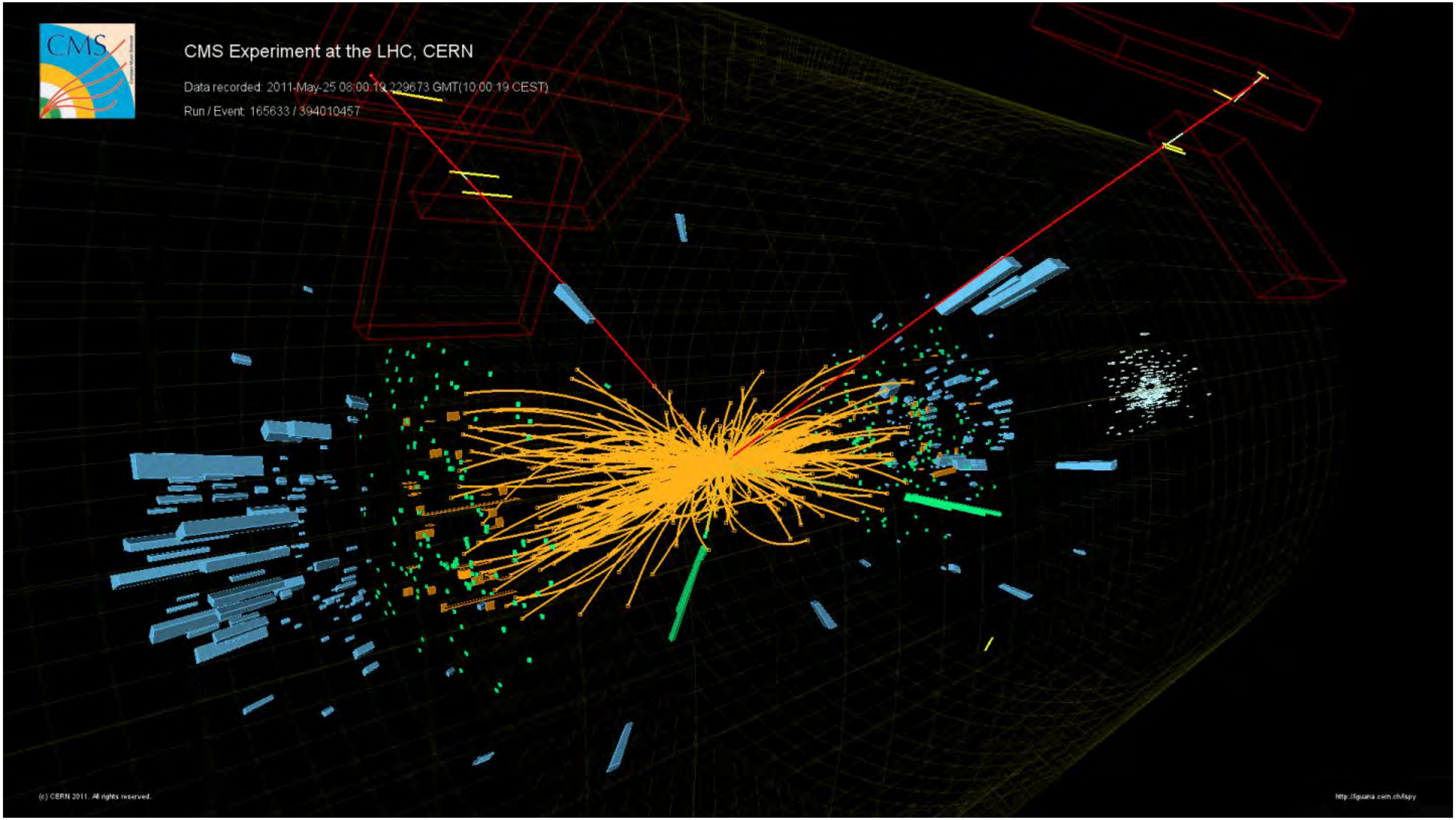
sagitta calculation



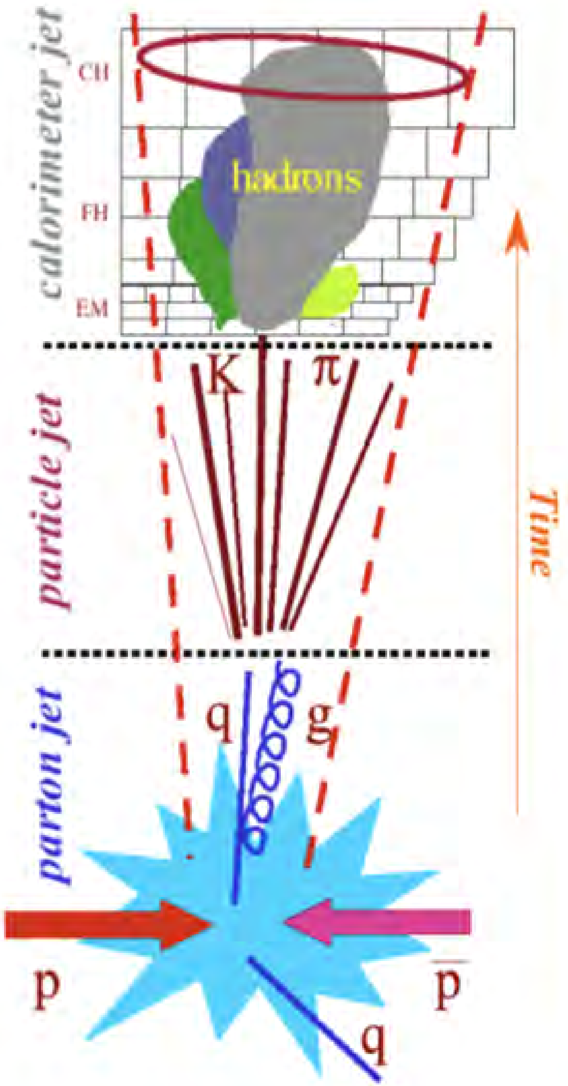


https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhysicsResultsMUO

Example of h → ZZ → 2e 2μ



Jet Finding



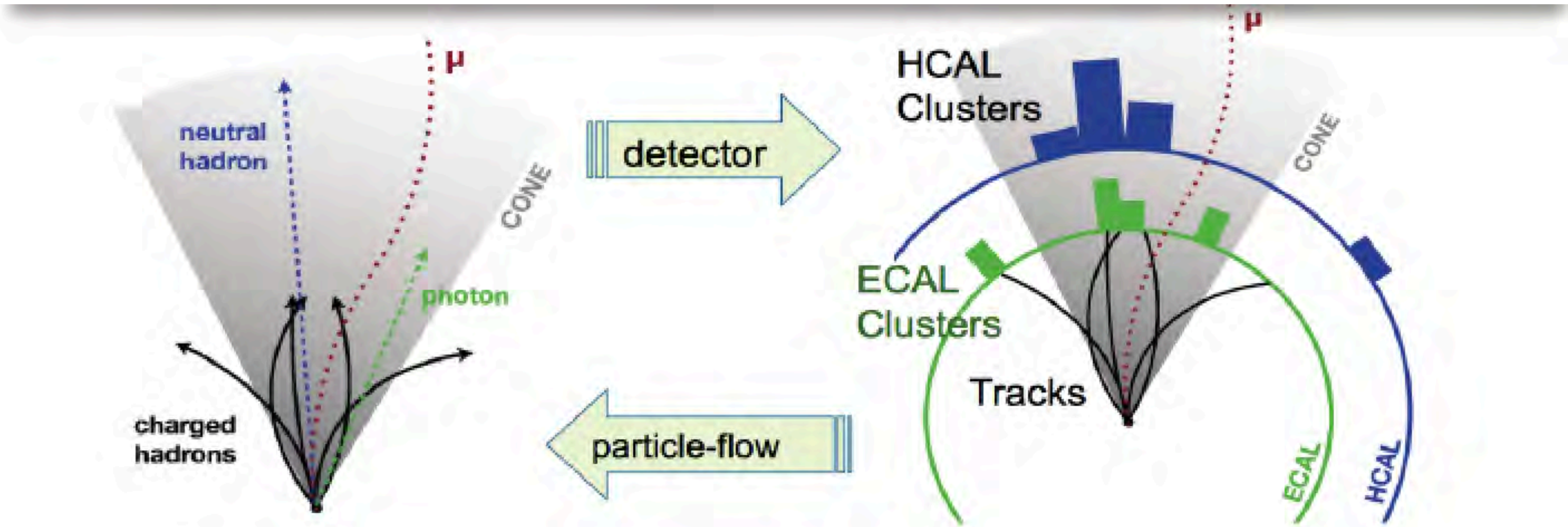
- **Calorimeter jet (cone)**
  - jet is a collection of energy deposits with a given cone *R*: $R = \sqrt{\Delta\varphi^2 + \Delta\eta^2}$
  - cone direction maximizes the total E_T of the jet
  - various clustering algorithms

→ correct for finite energy resolution
→ subtract underlying event
→ add out of cone energy

- **Particle jet**
  - a spread of particles running roughly in the same direction as the parton after hadronization

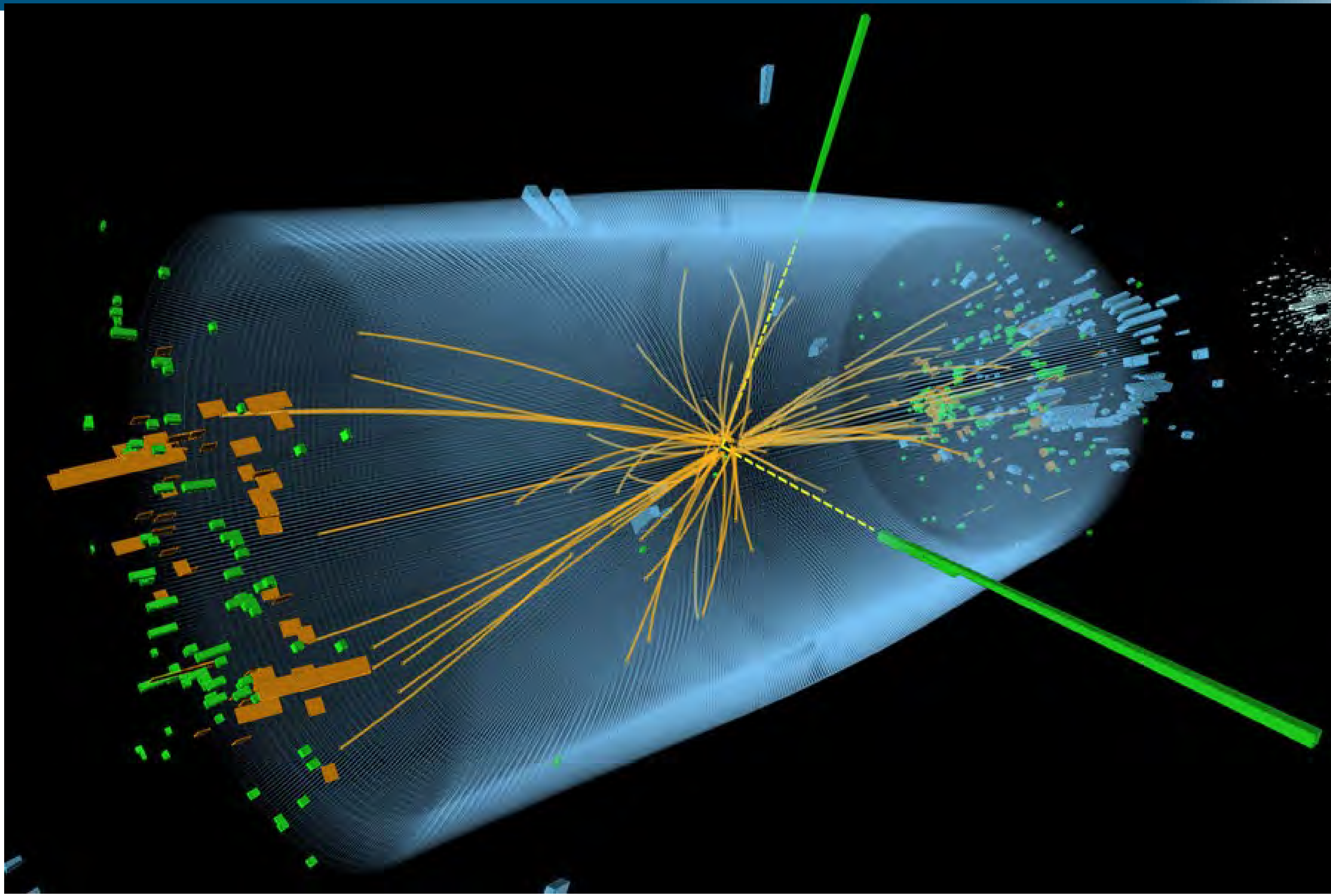
Global Event Reconstruction

Using all information of the detector together for optimal measurement



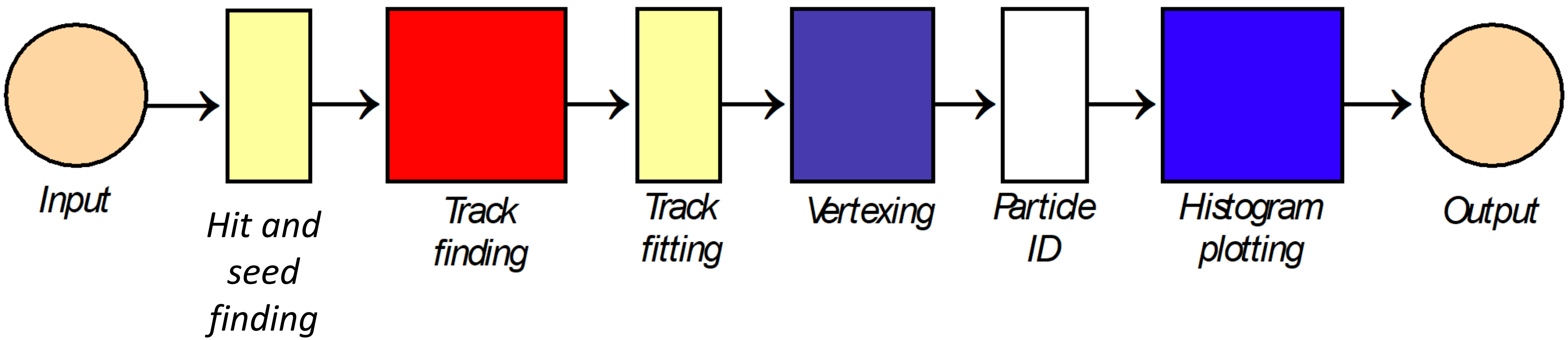
- Optimal combination of information from all subdetectors
- Returns a list of reconstructed particles
  - e,μ,γ, charged and neutral hadrons
    - Used in the analysis as if it came from a list of generated particles
    - Used as building blocks for jets, taus , missing transverse energy , isolation and PU particle identification

Example of $h \to 2\gamma$



Event Reconstruction Implementation

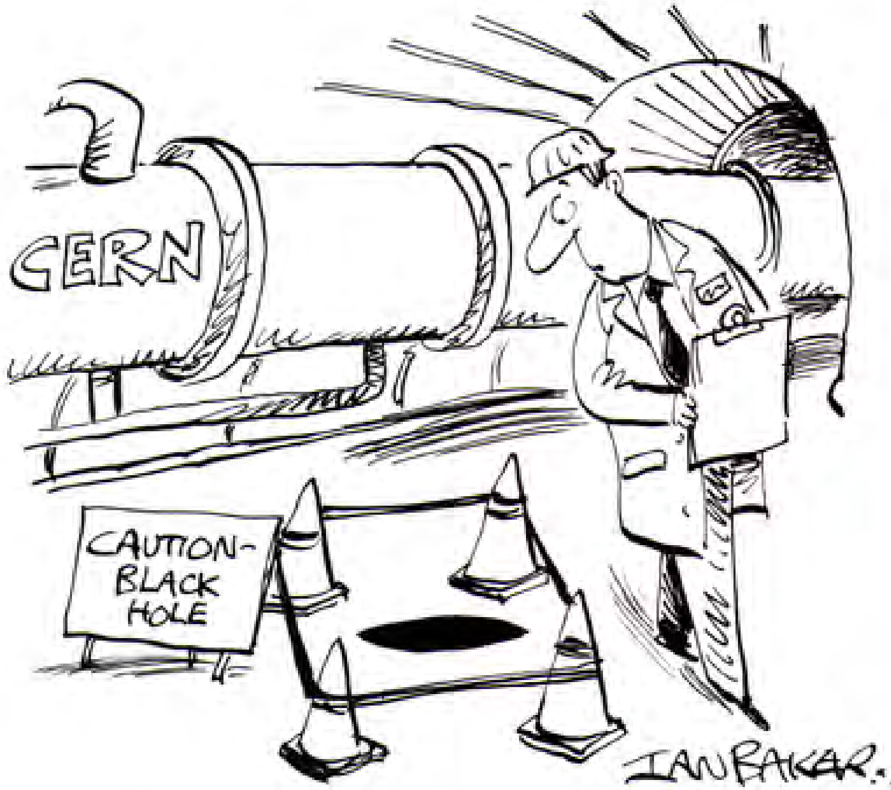
- Each reconstruction stage typically (sometimes by necessity) follows sequentially, e.g:



- Such a chain can be performed for a single event, or large set of events.
  - Reminder: each event is (usually) statistically independent of each-other.

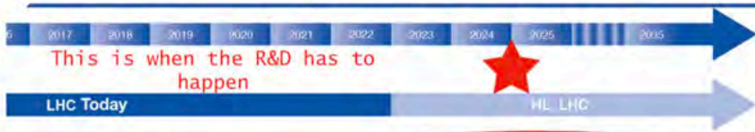
- Strategy for single core is obvious, but for multi core, not so much.
- Nowadays, reconstruction involves tens of thousands of CPUs worldwide - need efficient strategy.
- Currently limited by memory:
  - E.g. CMS end of 2011 could only 6 out of 8 cores on average.

Thank you for your attention!



Machine Learning

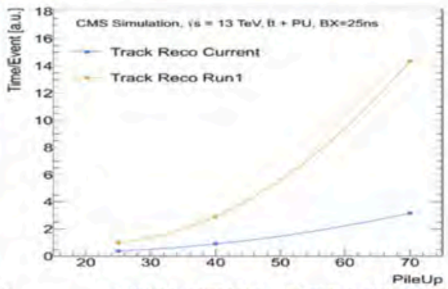
HL-LHC: elephant in the room



- ~40 collisions/event
- ~10 sec/event processing time
- (at best)Same computing resources as today

- ~200 collisions/event
- ~minute/event processing time(*)
- (at best)Same computing resources as

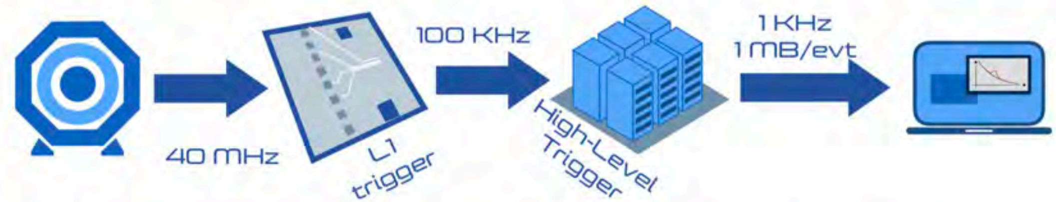
- *Flat budget vs. more needs =* ***current rule-based reconstruction algorithms will not be sustainable***
- ***Adopted solution:*** *more granular and complex detectors → more computing resources needed → more problems*
- ***Modern Machine Learning might be the way out***



(*)With nowadays software development

The LHC Big Data Problem

- *Too many data, too large data -> need to filter online*

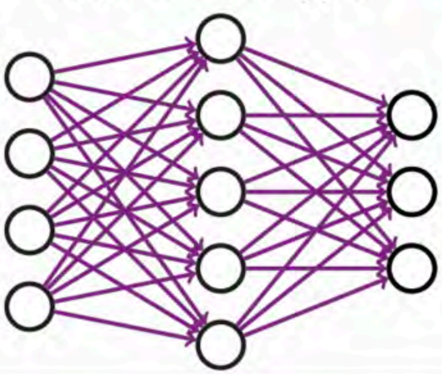


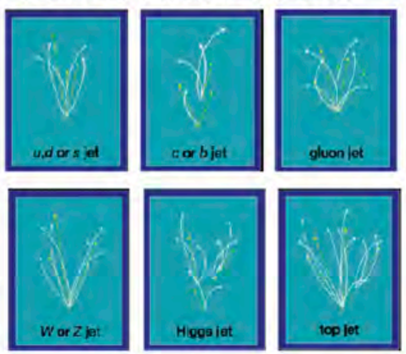
- *The solution to the HL-LHC problem: modern Machine Learning as a fast shortcut between the data and the right answer (the outcome of our traditional & slow algorithms)*

DEEP LEARNING TECHNIQUES

Deep neural networks

based on many low-level features with large training data sets to classify jets

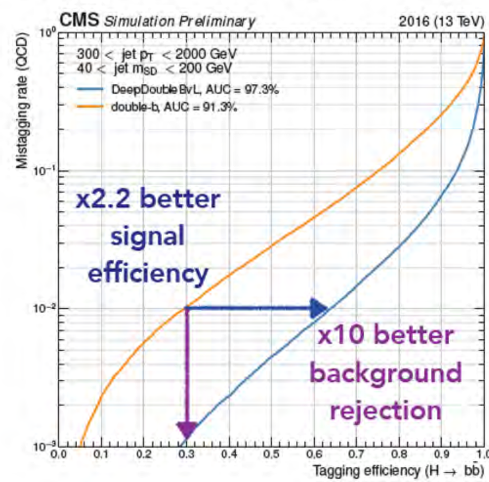




DP-2018/033

DEEP DOUBLE-B TAGGER

- Large performance gain over previous algorithm



What just happened?