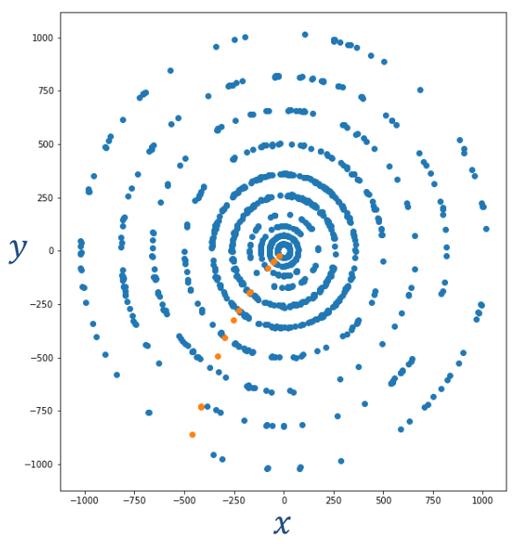
Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

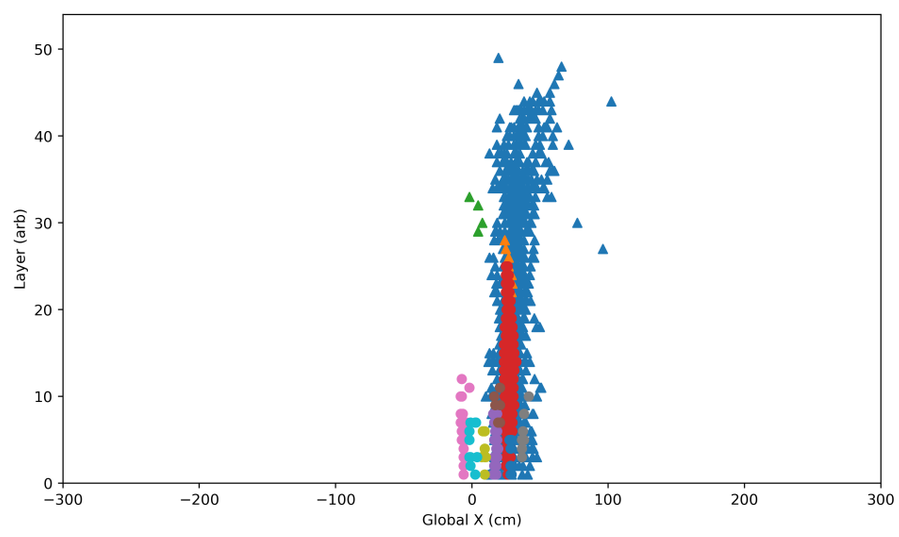
Introduction to Pattern Recognition in HEP

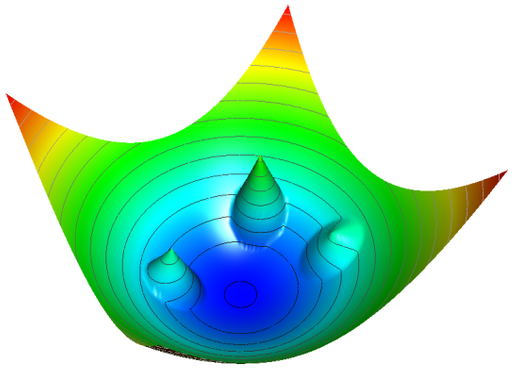
Lindsey Gray

HCPSS 2022

22 August 2022







Who am I and why am I talking about this?

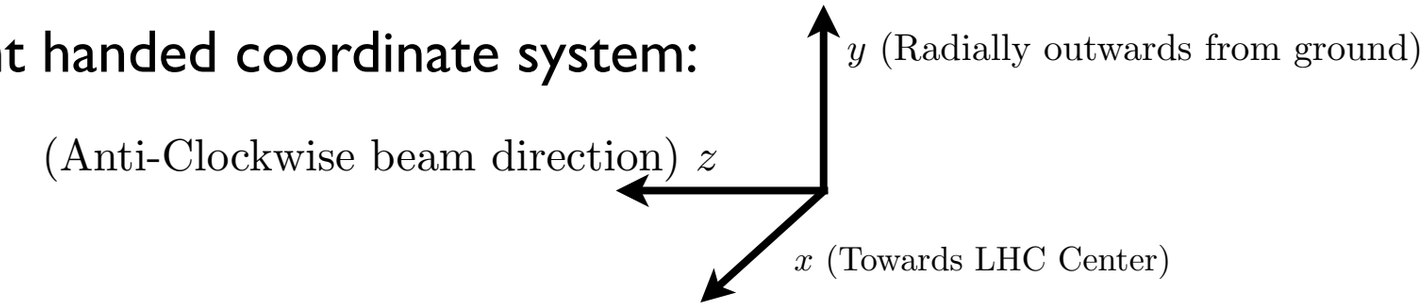
- I am a staff scientist at Fermilab working on the CMS experiment
  - I've worked in Electroweak and Higgs physics, with a theme of final states with photons
    - Largely related to backgrounds - this work required fairly detailed understanding of how we tag the signatures of interesting particles in our detector
  - This led me to becoming interested in understanding how we reliably find the patterns of all the kinds of pertinent signatures in our detector, and in turn to making contributions in a variety of areas across CMS such as:
    - Tracking charged particles and finding collision vertices
    - Energy deposit identification in the CMS electromagnetic calorimeter
    - Developing algorithms, use cases for precision timing detectors, imaging calorimeters
    - Utilizing machine learning in these contexts to build new algorithms and problem solving tools
  - I also work on analysis software design predominantly focusing on python-based tools and modern compute infrastructure

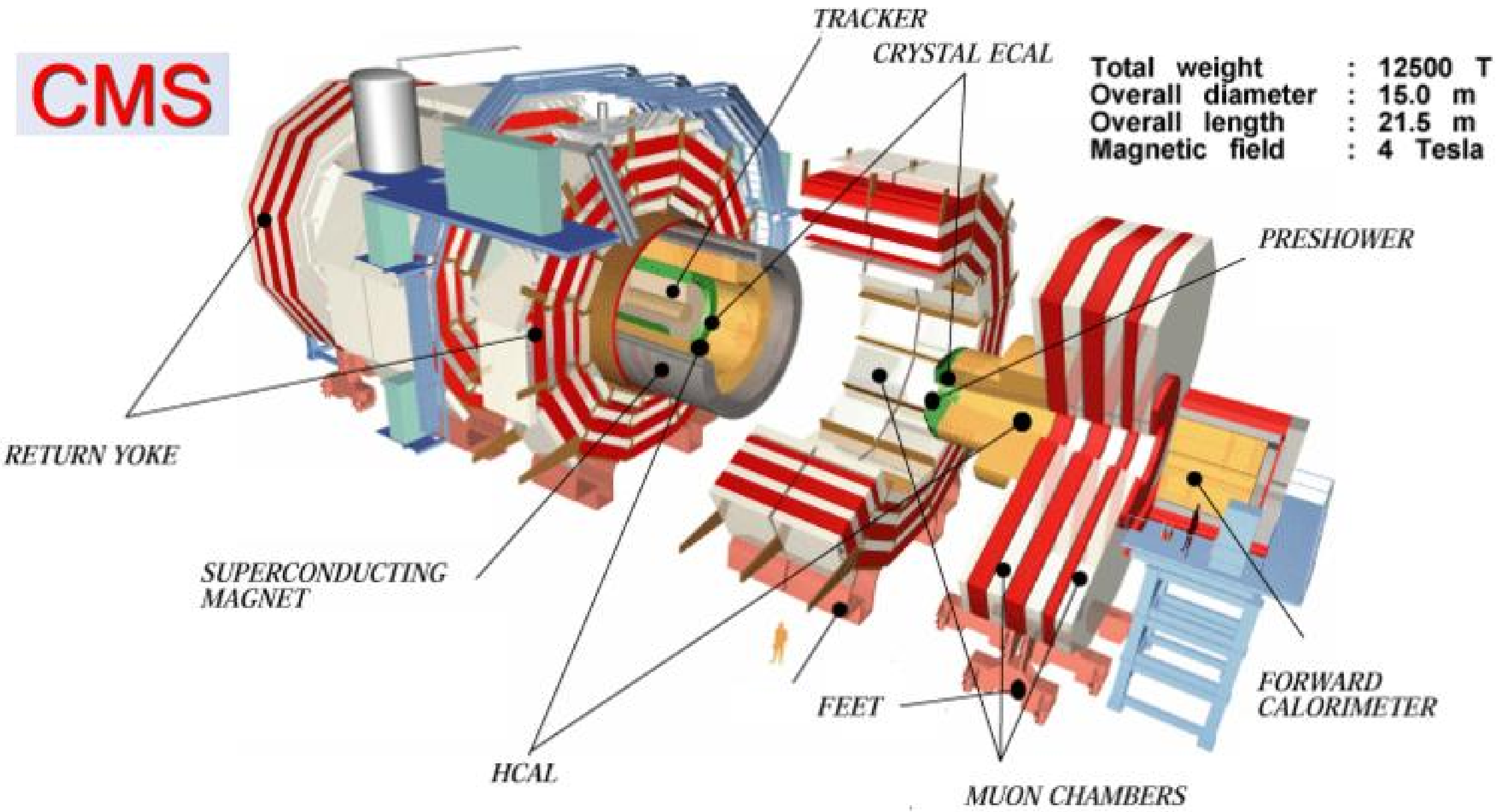
- This will be a fairly CMS/LHC-focused and practical discussion of pattern recognition and ML, though ~all of this is applicable elsewhere
  - The first two lectures will be the essential ingredients of Pattern Recognition and ML
  - The third lecture will be more on how these domains are evolving and merging together

Quick Reminder: The CMS Detector

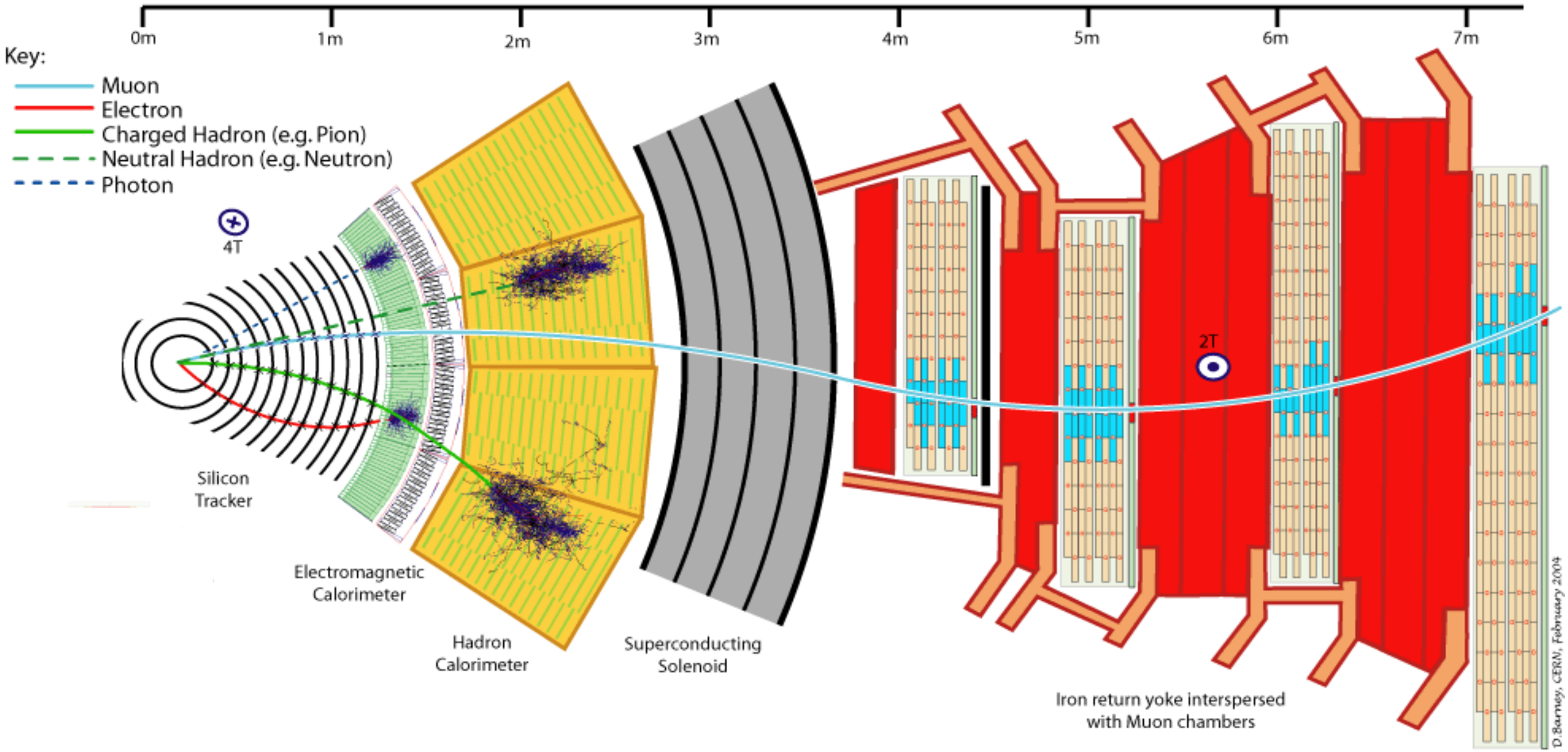
$$\eta = -\ln\tan\frac{\vartheta}{2}$$

Right handed coordinate system:



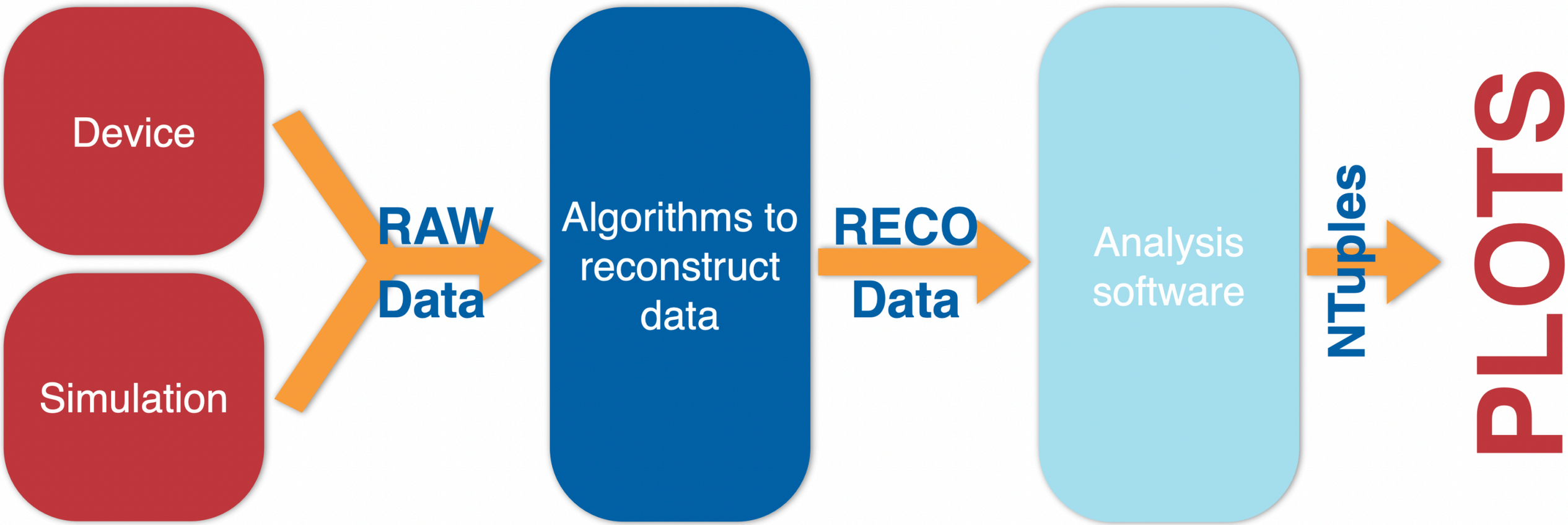


Pattern Recognition in a Nutshell



- A modern collider experiment like CMS is composed of multiple subsystems
  - Each designed to measure a specific suite of particles as well as possible
  - Pattern recognition is a suite of algorithms that tells us if signatures of interesting particles were found in a given detector, or even across multiple detectors

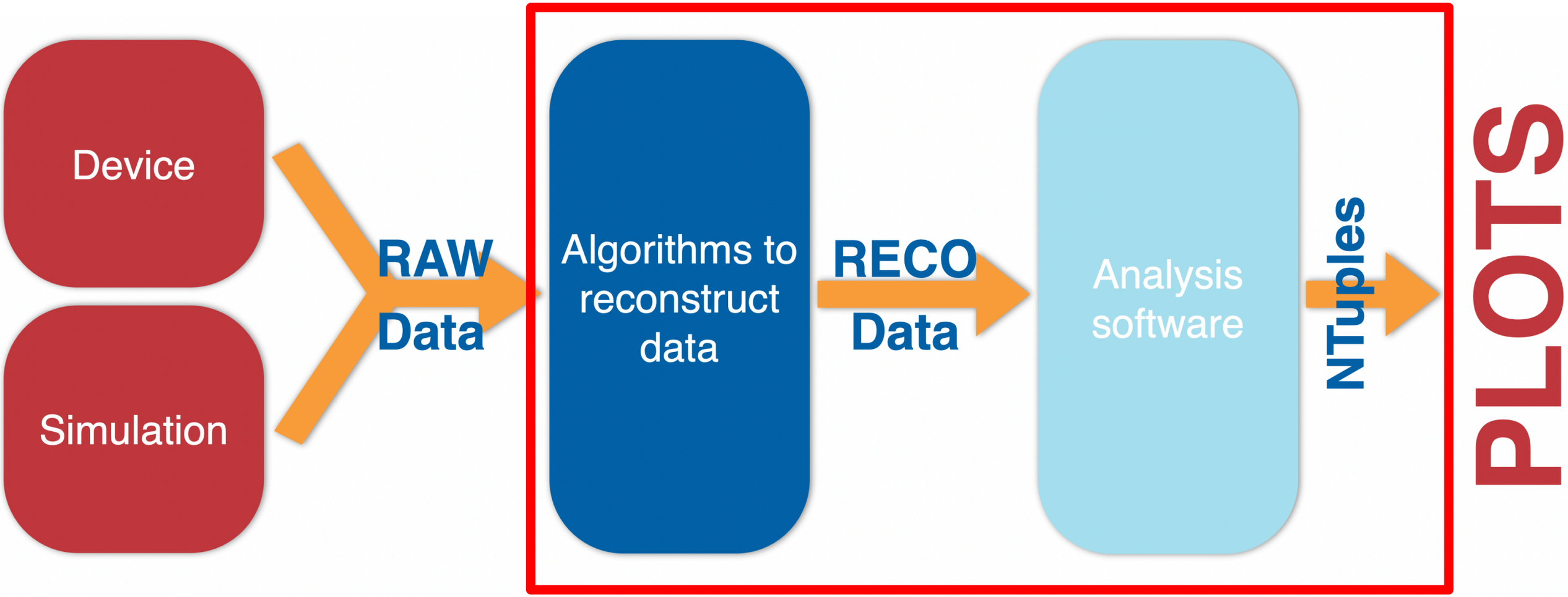
Where are pattern recognition algorithms used?



- **Software** is important for every step on the way to scientific results

(Oli. Gutsche)

Where are pattern recognition algorithms used?



- **Software** is important for every step on the way to scientific results

(Oli. Gutsche)

What do pattern recognition algorithms do?

- Primarily they serve as a way to detect the presence of an expected signature in our data
  - This could be an energy deposition that looks like a blob of active readouts next to each other
  - Or it could be looking for a large number of activated read outs along some expected trajectory, indicating they may come from the same particle

- This aids us greatly as physicists because each step of pattern recognition vastly reduces the complexity of our data so that we can analyze it efficiently and make ever-higher-level physics insights from the data we collect
  - “There was a charged particle” -> “the charged particle had this momentum” -> “it was probably an electron from the decay of a Z-boson”

Major Divisions of Pattern Recognition

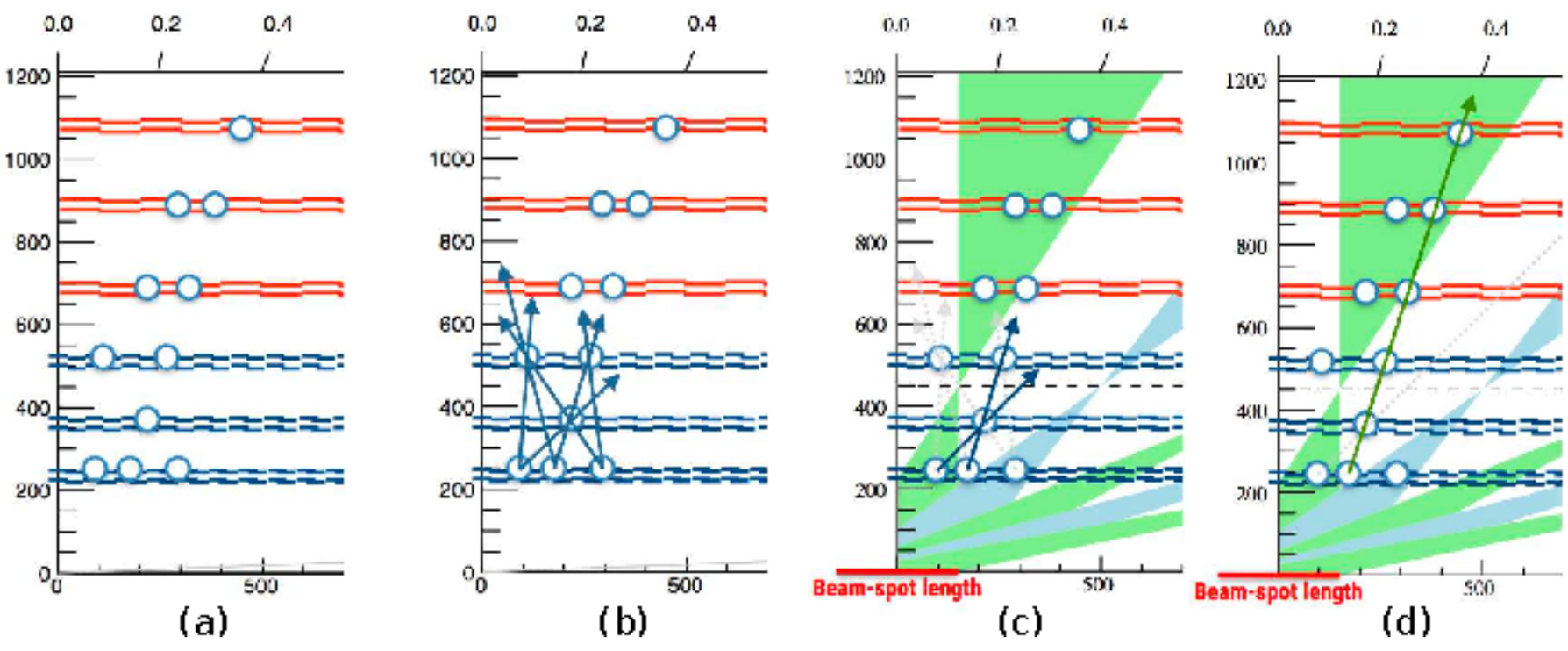
- The kind of algorithm used and how it is tuned depend very much on both the physics of the particle being sought after as well as the materials and readout electronics of the detector being used to find the particle

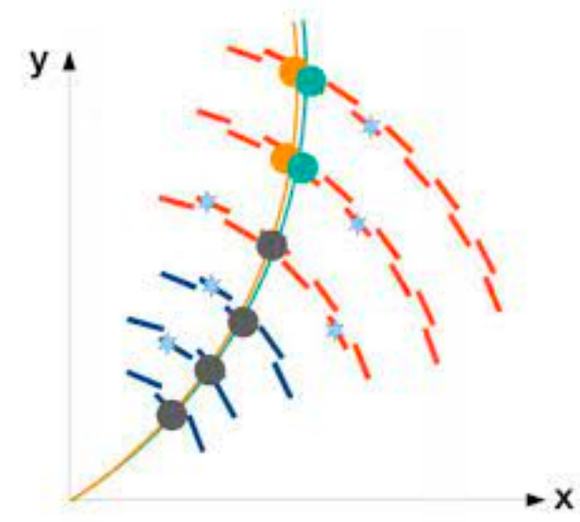
- In broad strokes, you've heard about tracking detectors and calorimeters
  - In both cases the objective for a pattern recognition algorithm is to group unlabeled data from these detectors together into an efficient and pure list of particle "candidates" for further analysis
  - I'm going to try to avoid math (lots of well trodden linear algebra and statistics) where it's not important and focus more on:
    - What these algorithms do
    - When you should think about using them

- This isn't going to cover all pattern recognition algorithms, but rather the ones that might be pertinent to your scientific interests
  - The way these algorithms work can hint at mechanisms for systematic uncertainties, as well as backgrounds in your analysis

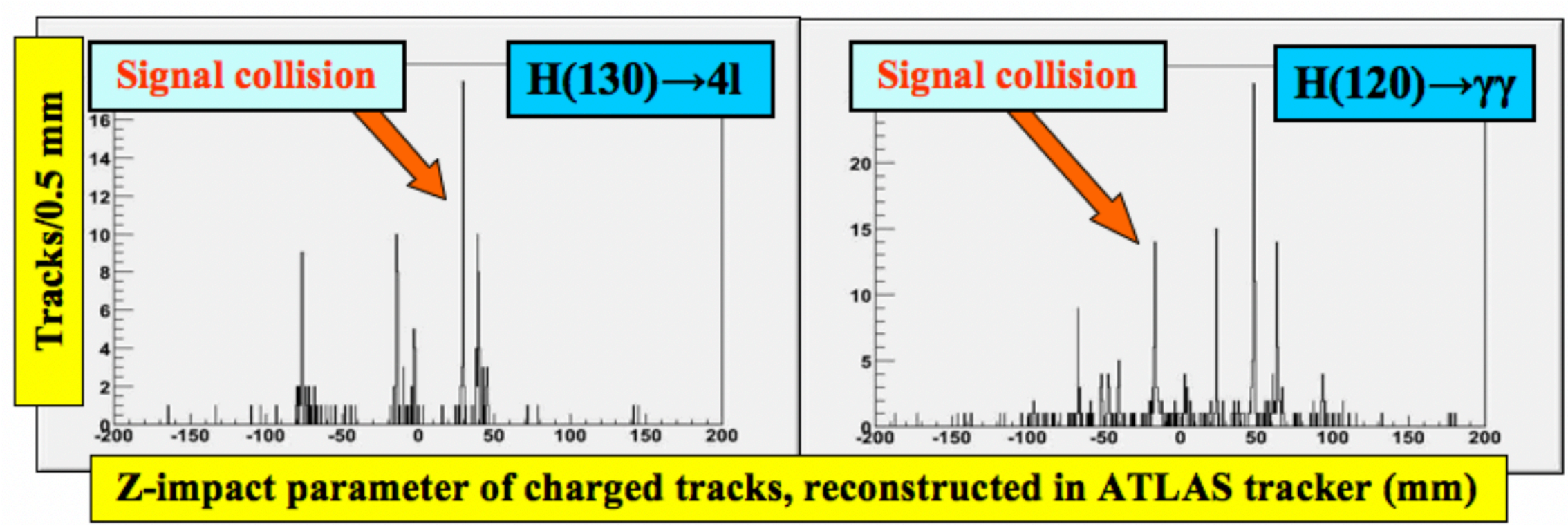
Major Divisions of Pattern Recognition: Track-finding

- Tracking detectors are low density, sparsely sampled, with fine granularity
  - Very little material as to not destroy the trajectory of the particle
  - Consequently, algorithms that automatically detect tracks have complicated models within them that allow a hypothesis for a particle's trajectory to be propagated between the sparse sampling layers
  - In order to build and then extract the physical parameters of a track hypothesis we have to make sure the data points collected by an algorithm are consistent with each other from layer to layer of the detector and then wholly consistent with our model of the particle's transport.





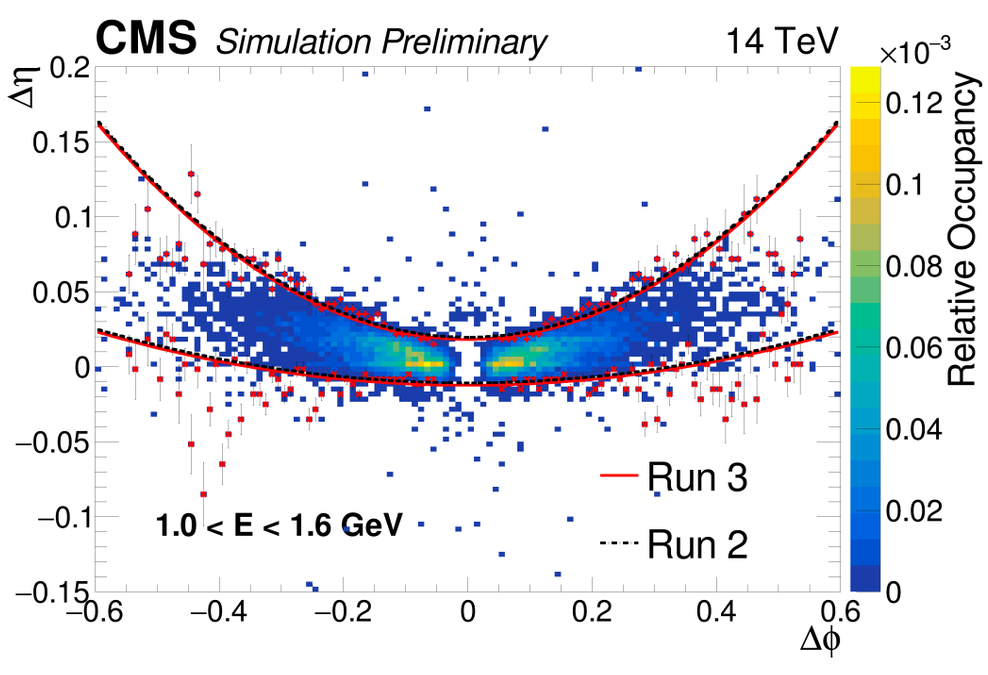
Major Divisions of Pattern Recognition: Vertexing

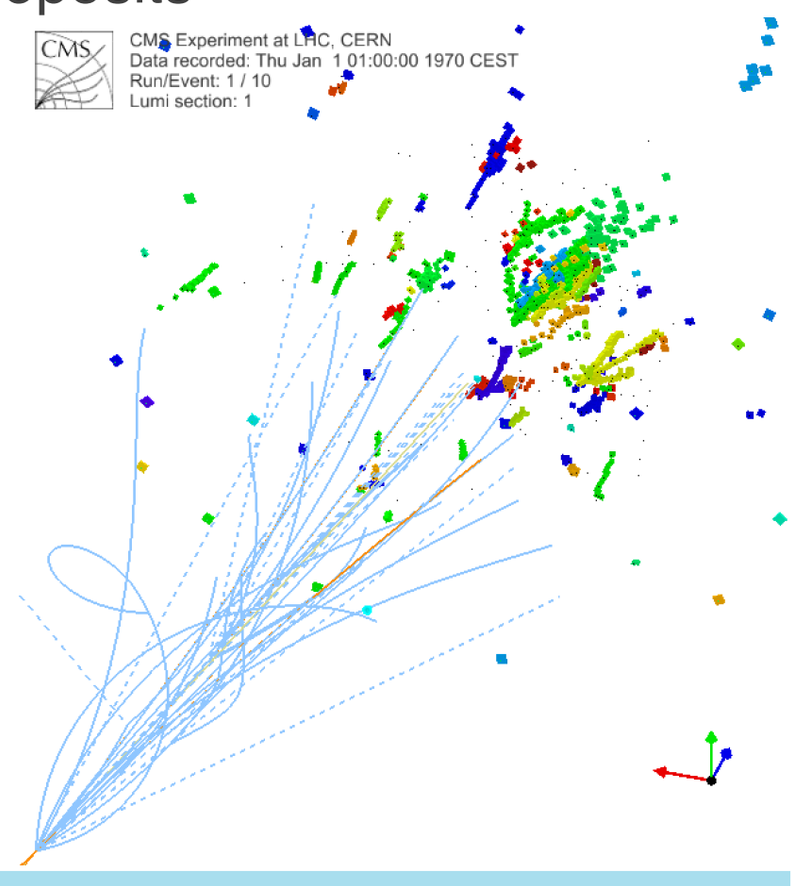


- Once you have a list of track hypotheses from track finding it is useful to determine where groups of tracks in that list have come from
  - This algorithm is critically important for b-tagging and pileup removal
    - The better you know a collision's position, the better your ability to compare what's around it
  - When analyzing a small number of charged tracks, often used to re-fit or re-build tracks that should be coming from a common point of origin to improve measurement quality
- Challenge at the (HL-)LHC: knowing which vertex is the most interesting one!

Major Divisions of Pattern Recognition: Clustering

- Calorimeters are dense, densely sampled, with coarser granularity than a tracker*
  - As you saw last week, these devices measure particles' energies by causing them to interact with material and measuring ionization deposits and/or Cherenkov
  - Pattern recognition in calorimetry is used to define what energy deposits in a calorimeter are relevant to each other, which are most often close-by energy deposits
  - The more finely-segmented your calorimeter is the more complex of a model is needed to determine meaningful groupings of energy deposits

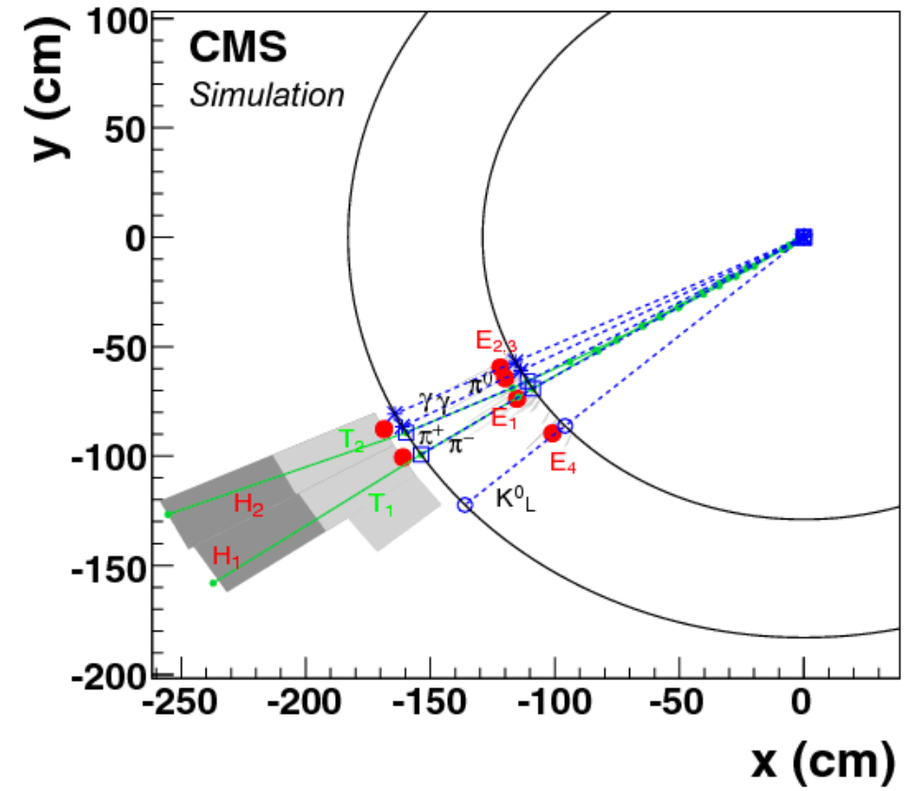


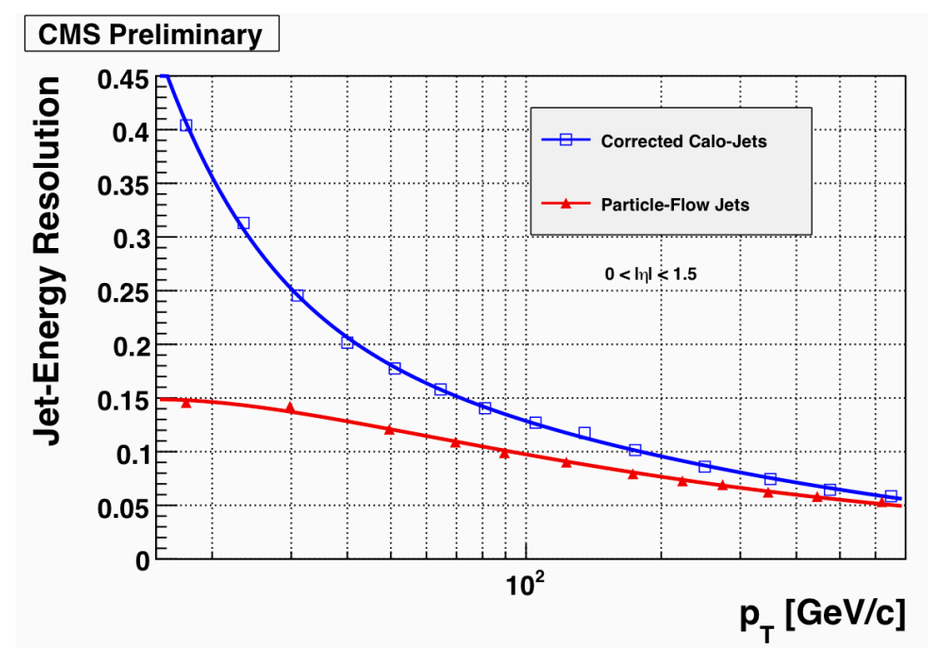


* - this is changing with modern technology but not yet broadly true

Major Divisions of Pattern Recognition: “Particle Flow”

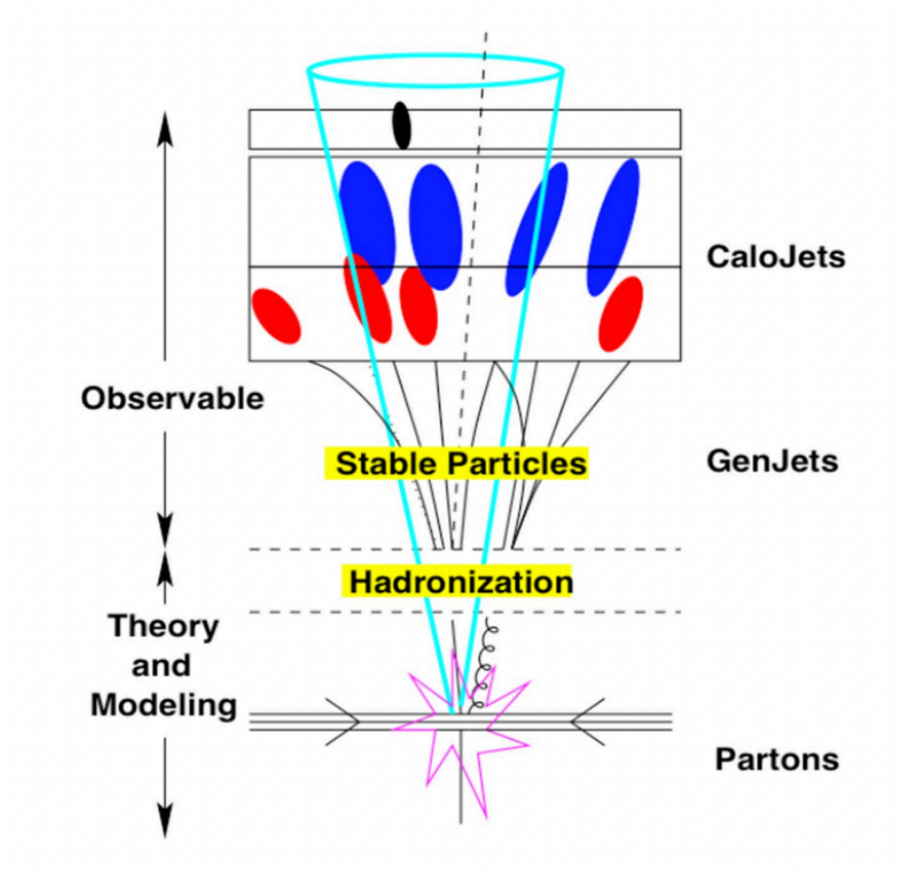
- Once you have tracks and clusters from the various components of a detector it’s nominally possible to do an analysis
  - However, we know that despite reconstructing the particles in different detectors these are measurements of the same particles
  - This can be used to our advantage to exploit the strengths of each sub-detector
- The overall goal of these kinds of algorithms is to cluster together signatures of the same particle across *different* detectors
  - Then cross-correlated that detector information to determine the most precise measurement
  - Or even determine the presence of overlapping particles
- This class of algorithms has be revolutionary for experimental jet physics

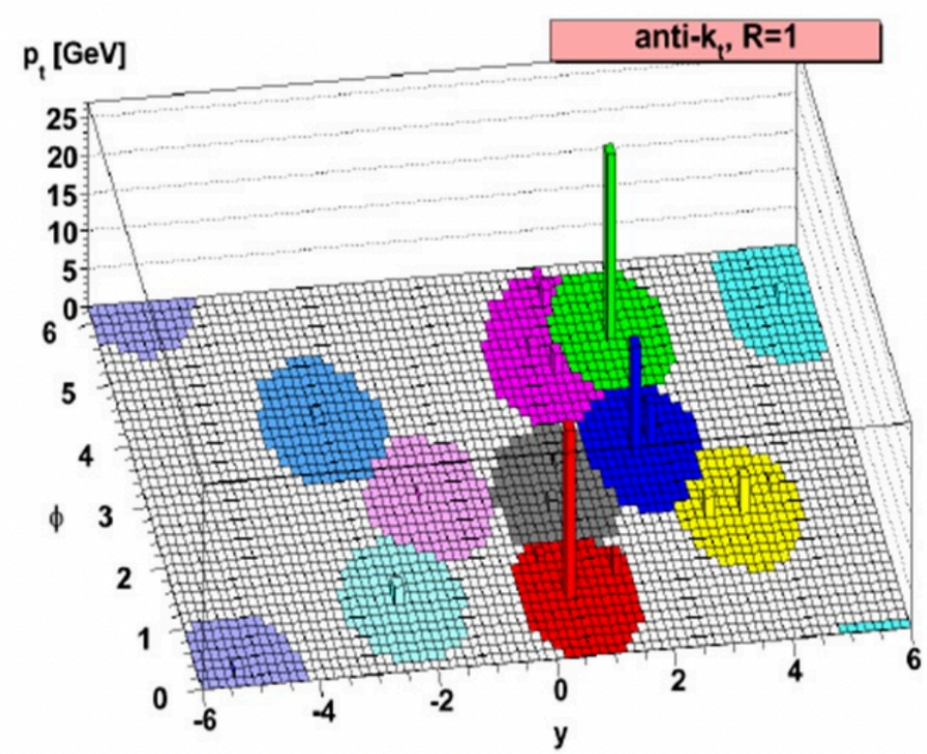




Major Divisions of Pattern Recognition: Jet Finding

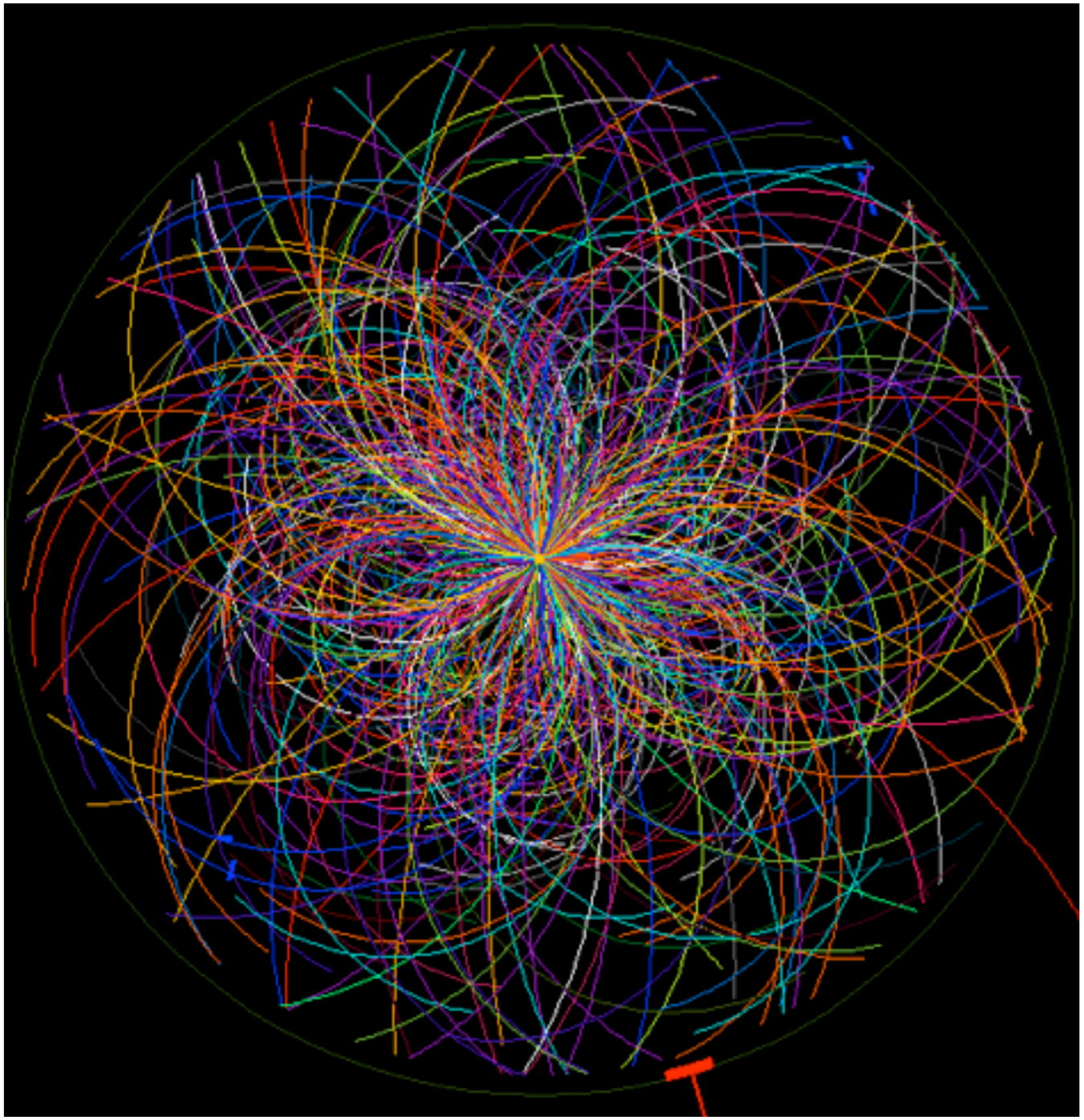
- Finally, once you have a list of particles, you'll need to be able to find not only distinct detector-based signatures like leptons and photons but also the diffuse signatures of hadronized color-charged initial states
  - Modern jet algorithms function by assigning particles to each other based on a transverse momentum weighted distance
  - Different strengths of the weighting give rise to different behaviors of the algorithm
    - This can be used to stabilize the algorithms in the presence of backgrounds





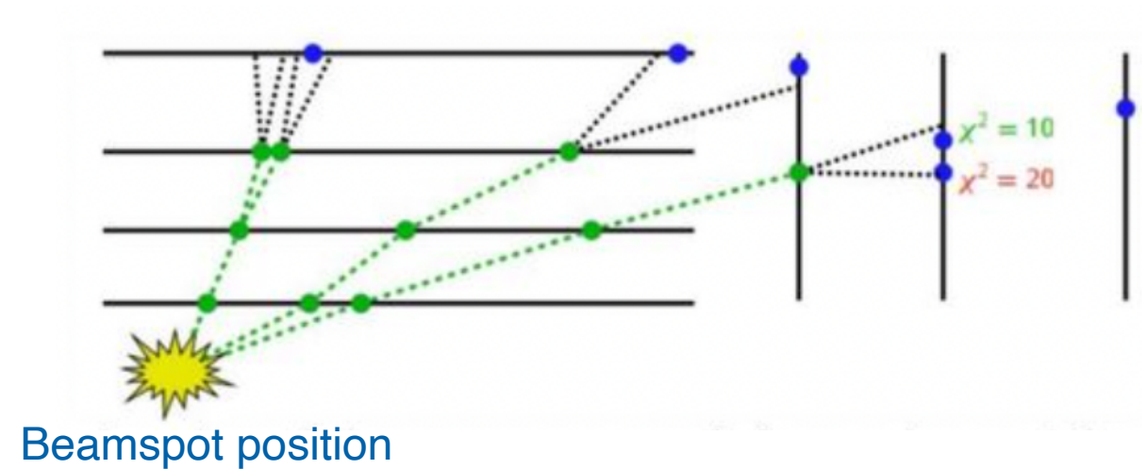
Now that we've done the "use survey" of pattern recognition algorithms, it's time to inspect them a little more closely.

Tracking - how to find charged particles?

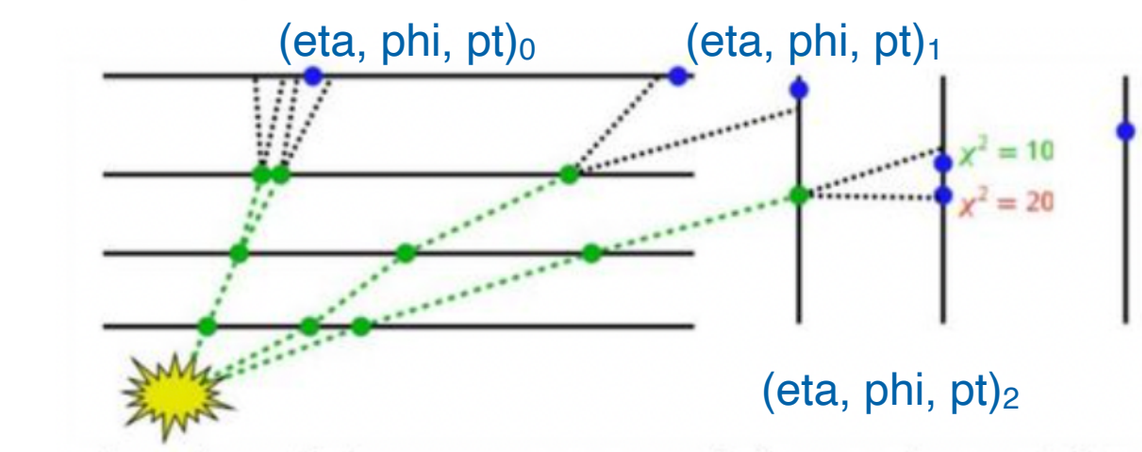


Tracking - Track Seeding

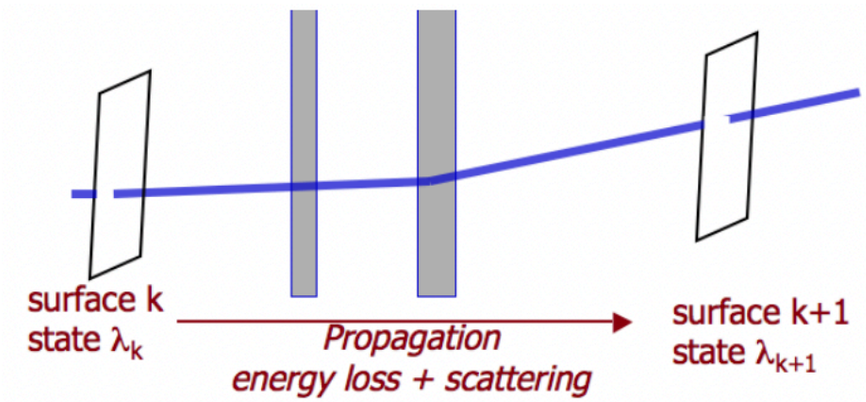
- In general the tracking algorithm requires a set of guesses to start from
  - At first the only information we have is the estimated size of the interaction region from machine parameters and confirmation with measurements
  - However, this is still enough to reduce from having to consider all possible combinations of detector readouts to only those that are consistent with a charged track emanating from the interaction region
- As more hits are accumulated we are better able to discriminate incorrect combinations
  - Similarly, the better we are able to guess the track's parameters



Beamspot constrained estimate



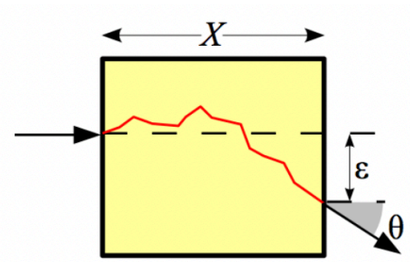
Tracking - The Kalman Filter for Track Finding and Fitting



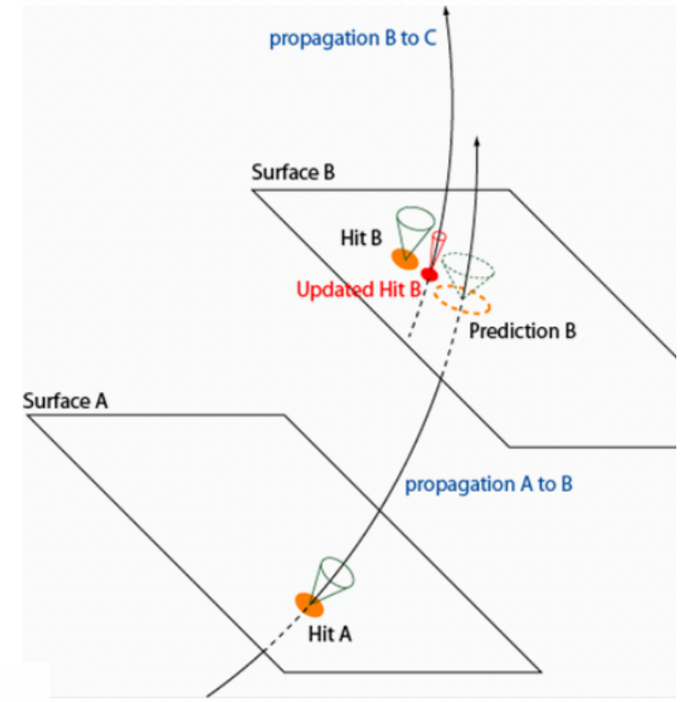
Ionization Losses

$$-\frac{dE}{dx} = Kz^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 T_{\max}}{I^2} - \beta^2 - \frac{\delta(\gamma\beta)}{2} \right]$$

Multiple Coulomb Scattering



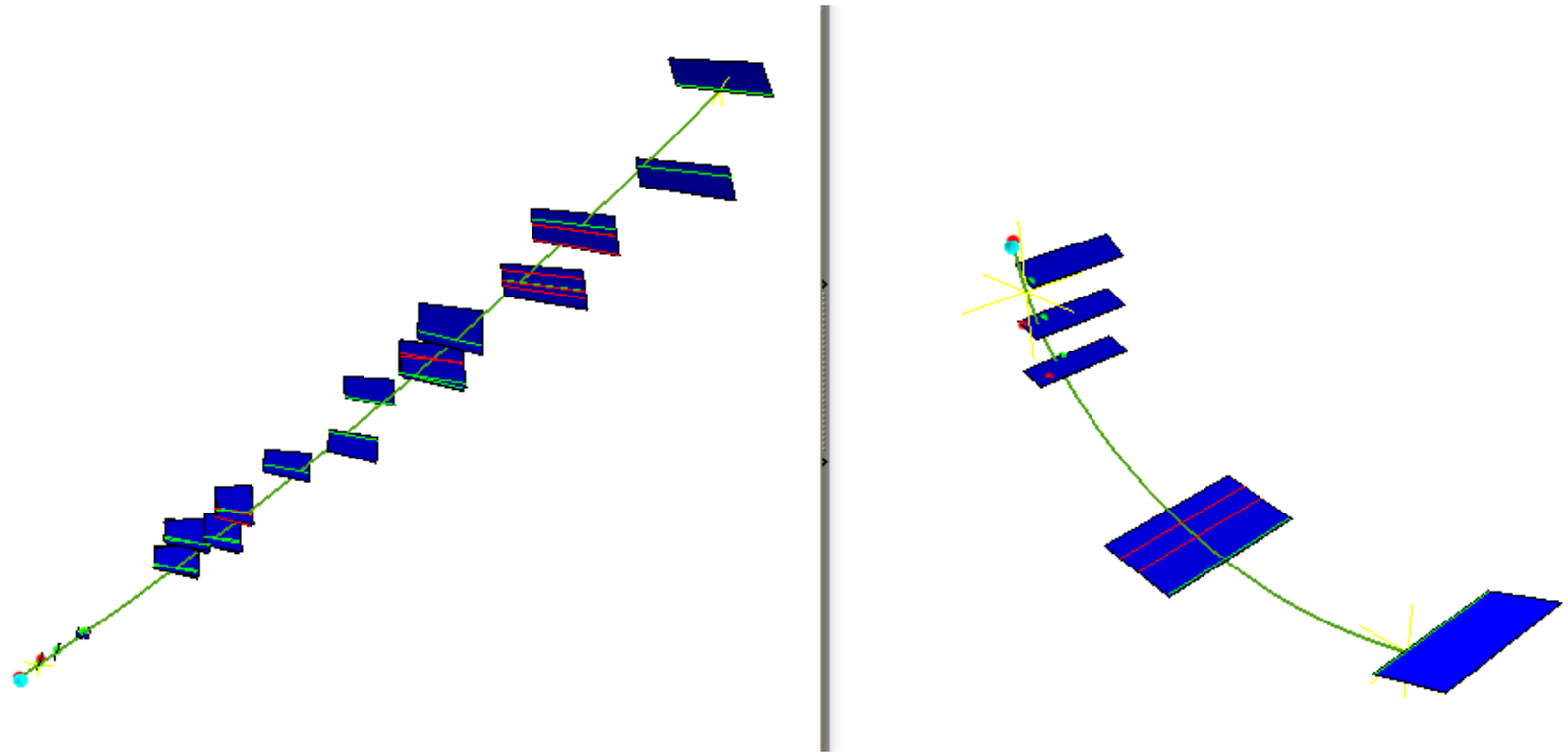
$$\sigma(\theta) = \frac{13.6\,\text{MeV}}{\beta c p} z \sqrt{x/X_0} \left(1 + 0.038 \ln (x/X_0)\right)$$



Kalman filter

- The Kalman filter is a linear fitting method that can be used in situations where new data is being added as a function of time
  - It originated in measuring dynamical systems like missile guidance and auto-pilots
  - In our case we have a dynamical system in the sense that for each track guess we have a model of the magnetic field and materials in the tracker and our charged particle is propagating through the detector subject to each of these
- The Kalman filter allows us to update our model with measurements such that further data improves our ability to predict the next measurement

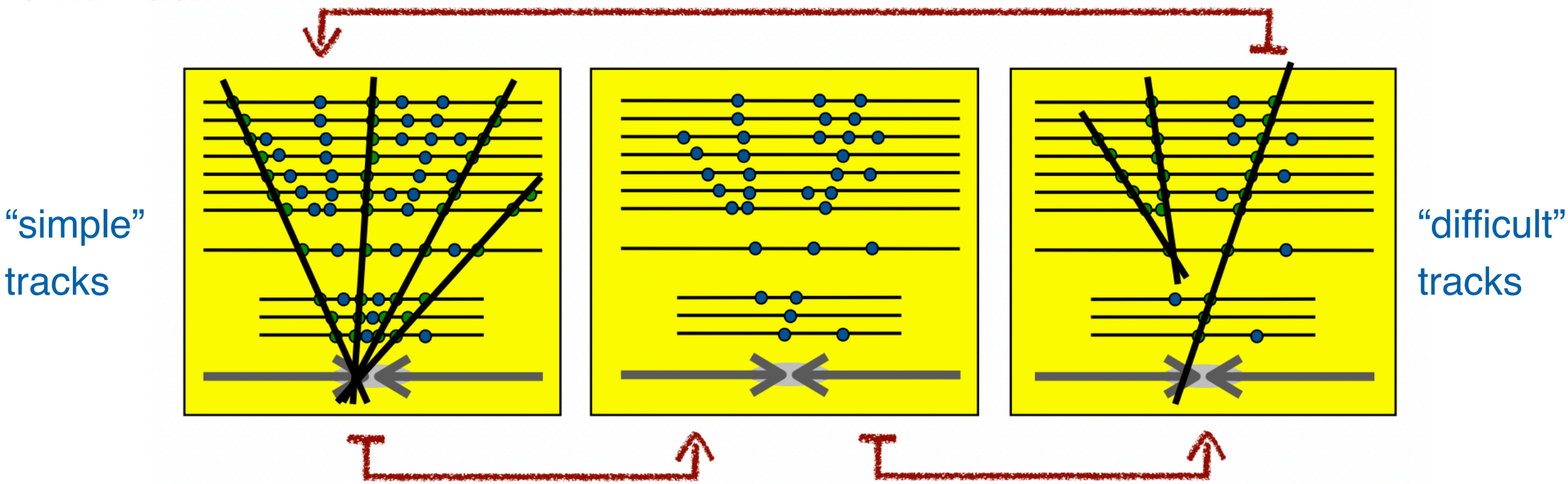
Tracking - The Kalman Filter for Track Finding and Fitting



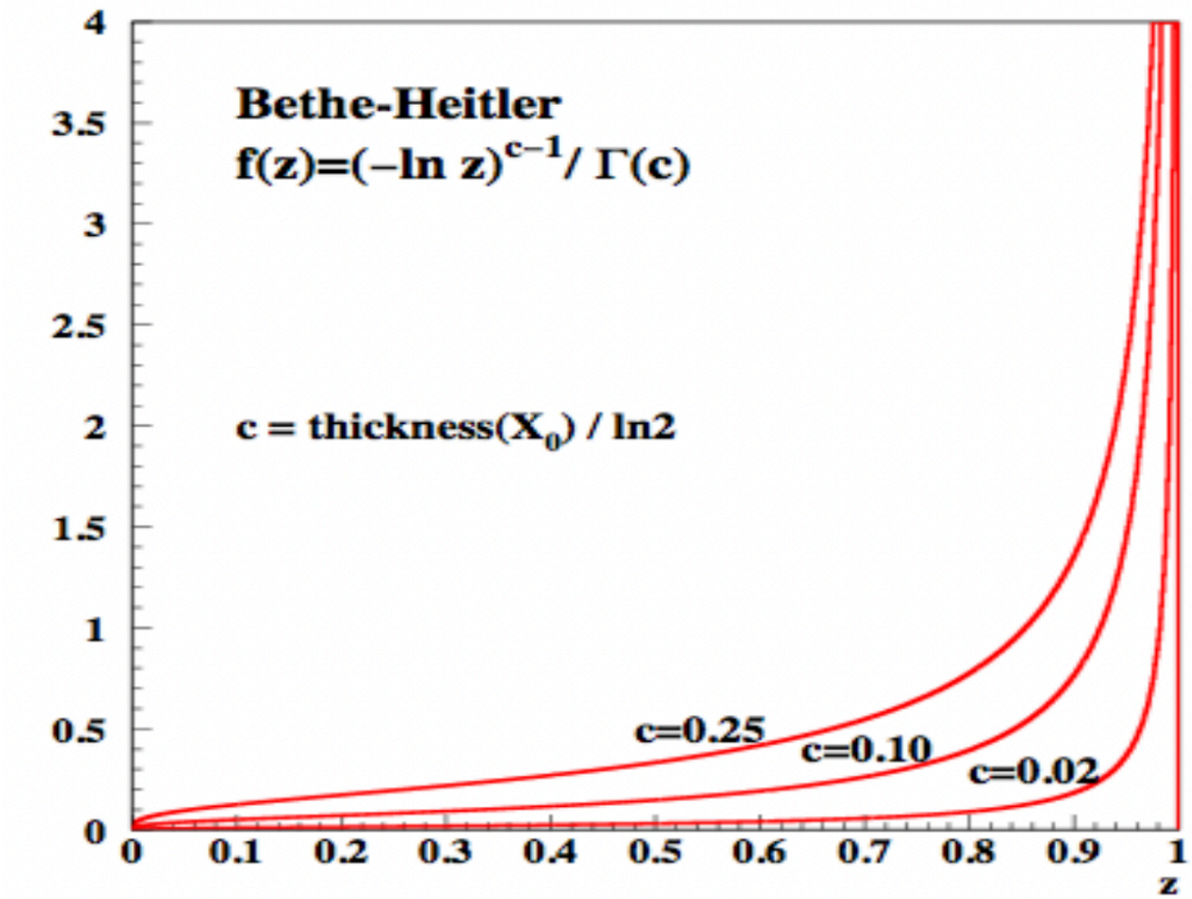
- Since the Kaman filter always updating an internal model we can take advantage of this once we have exhausted all of our input data
  - With the dataset fixed we no longer operate in a time-ordered mode and can consider the constraints of later data on earlier hypotheses of the track parameters
  - This generates a “global” fit of the track parameters and extracts the maximum amount of information about the track given
- This allows us to determine how consistent the track is with our model and make decisions on if we want to use this in our analysis

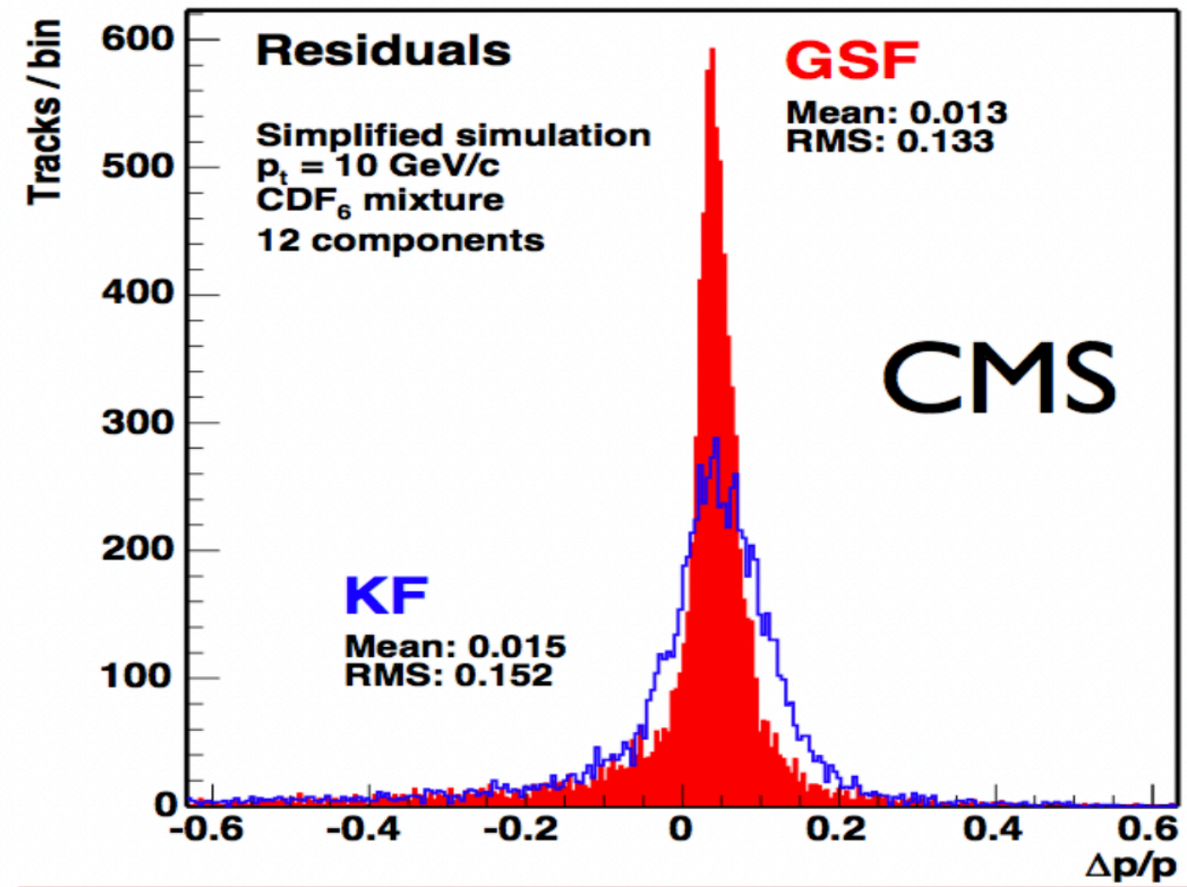
Tracking - Iterative Track Finding

- So far we've talked about finding single or few tracks
  - How do we find multiple tracks that might be overlapping?
- One approach is to try to find tracks sorting them by how "simple" they are to find
  - For instance, starting from long, straight tracks with well-measured initial states
  - Then moving on to tracks that show large curvature or start later in the detector
- After completing a step, remove all the found tracks and continue with the remainder



Tracking - Gaussian Sum Filter for Electrons



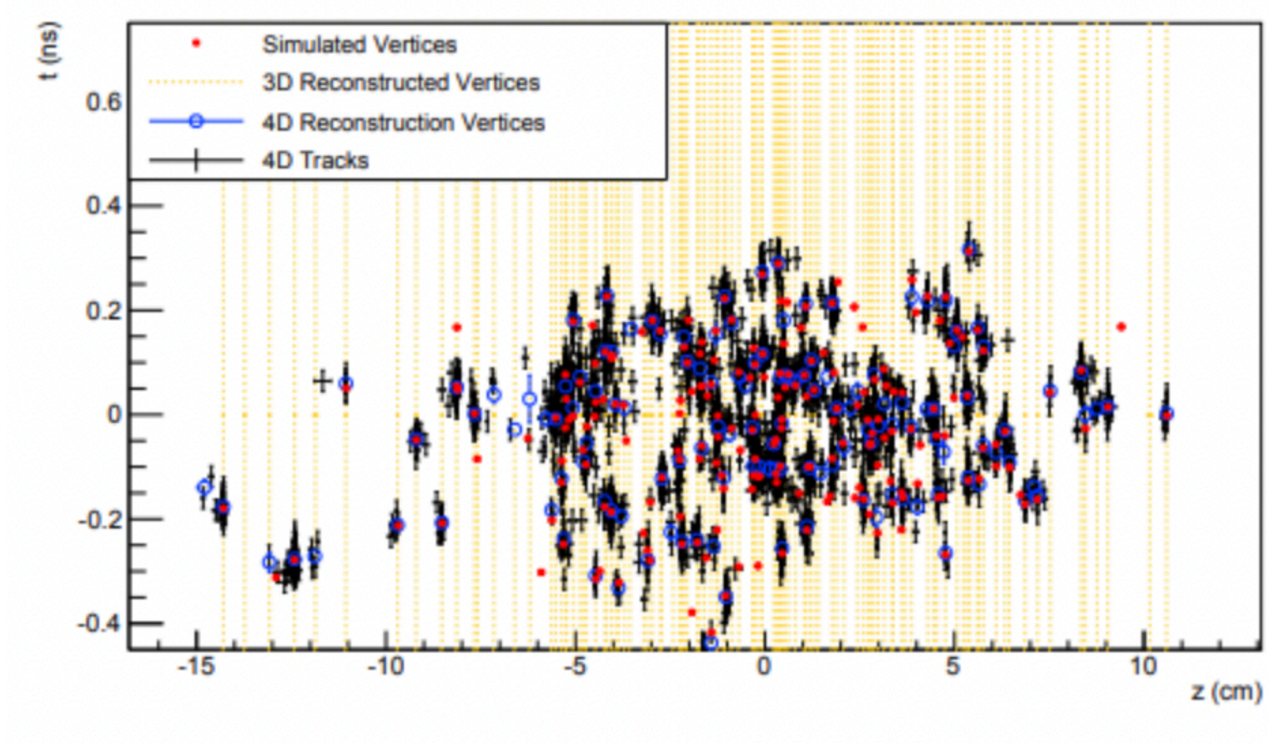


- Electrons can be a special case in tracking since the particle is so light
  - Depending on energy, more multiple scattering and/or a significantly increased chance to have a large non-gaussian energy loss (bremsstrahlung)
  - In order to model this, approximate the energy loss probabilities with a sum of gaussians as opposed to a single one (as in the Kaman filter)
  - Minimize the number of necessary losses according to the expected distribution
- Given that this requires iterated integration, this is very expensive but worth it for cases like electrons

Vertexing - Simulated Annealing

Algorithm 4: Simulated annealing

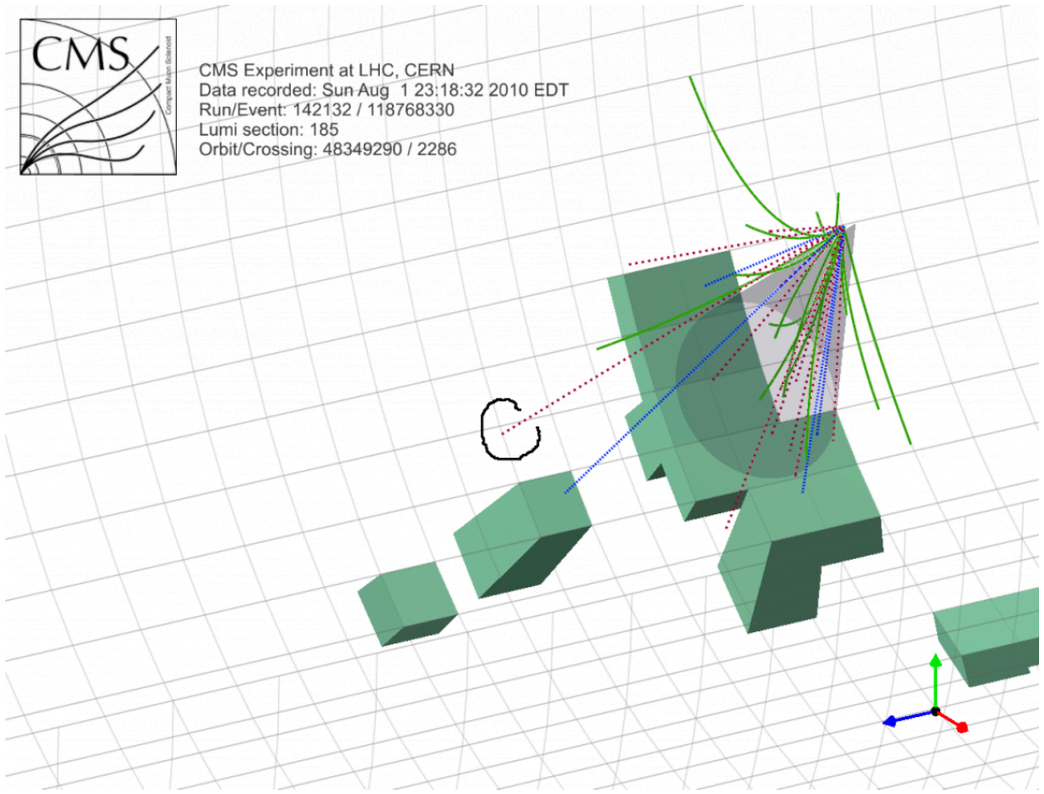
```
begin
    S_0 ← Construct initial solution;
    if S_0 ≠ ∅ then
        S* ← S_0;
        for i = 1, ..., n_sol do  S_i ← S_0 ;
        Temp ← Temp_0;
        while stop criterion is not met do
            for I = 1, ..., maxI do
                for i = 1, ..., n_sol do
                    S'_i ← randomNeighbor(S_i);
                    Δ ← f(S'_i) − f(S_i);
                    if Δ < 0 then  S_i ← S'_i ;
                    else
                        r ~ U[0, 1];
                        if r < exp(−Δ/Temp) then  S_i ← S'_i ;
                    end
                end
            end
            for i = 1, ..., n_sol do
                S_i ← multi-RVND(S_i);
                if f(S_i) < f(S*) then  S* ← S_i ;
            end
            for i = 1, ..., n_sol do  S_i ← S* ;
            Temp ← αTemp;
        end
    end
end
```

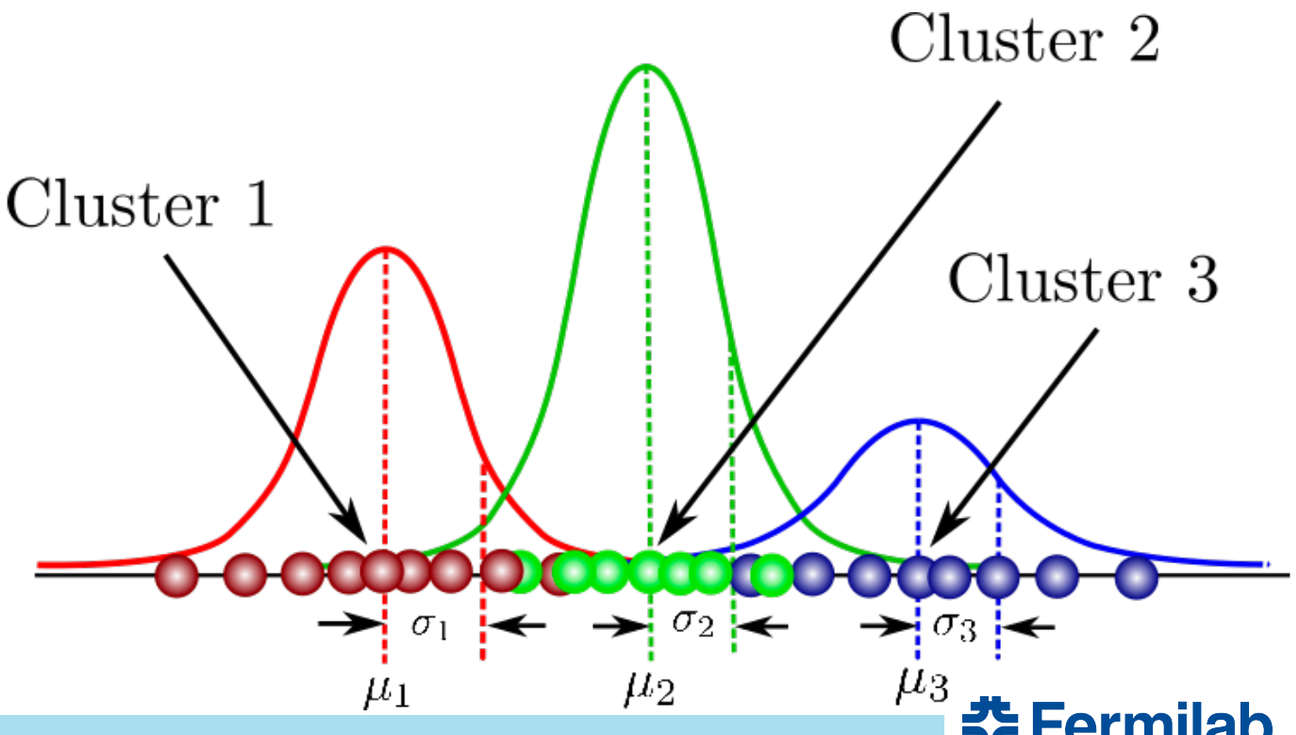


- From track finding we know the best estimates of the track parameters at any point along the estimated trajectory, as well as the errors on those parameters
- Physically, we can group the tracks together by looking collectively at their closest approaches to the interaction region
  - Using the annealing algorithm above we start from loose associations of tracks and consider with tighter and tighter tolerance how consistent their errors are

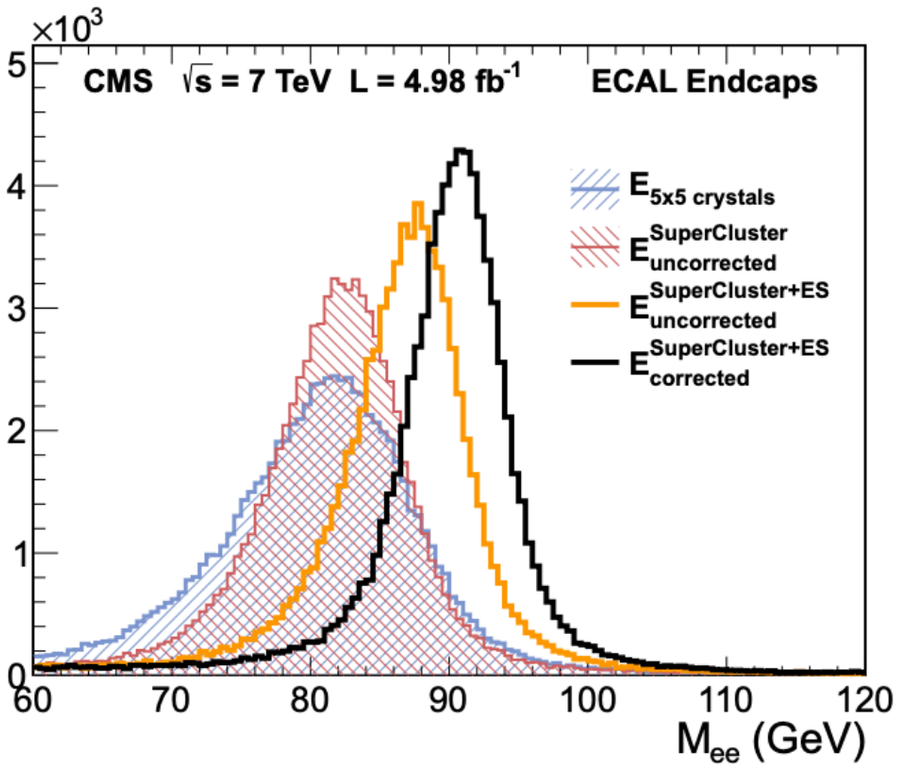
Calorimeter Clustering - Gaussian Mixture Models

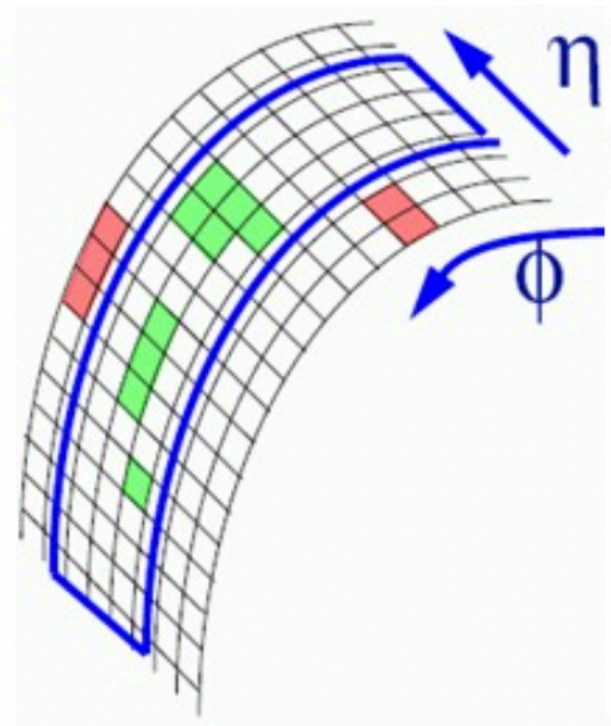
- The granularity of Calorimeters in CMS is often more fine than the actual spread of the energy deposit left by a particle
  - For most of the CMS calorimeters the size of a cell is about 60-90% of the expected size of the shower *if* the shower is in the center of the cell
- Hadron collider environments are also high-multiplicity and particles can be near each other!
  - So - we have to deal with cases where not only energy is spread over multiple cells but that energy deposit is shared between multiple particles!
- A typical approach in CMS is to assume the particles have a gaussian profile

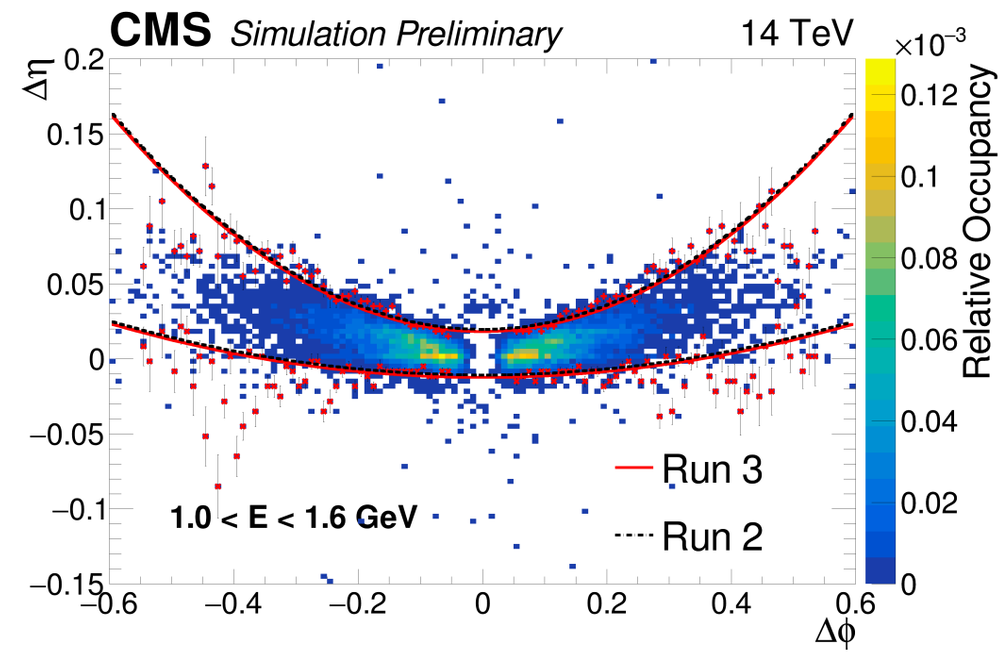




Calorimeter Clustering - Heuristic Rule Based Groupings

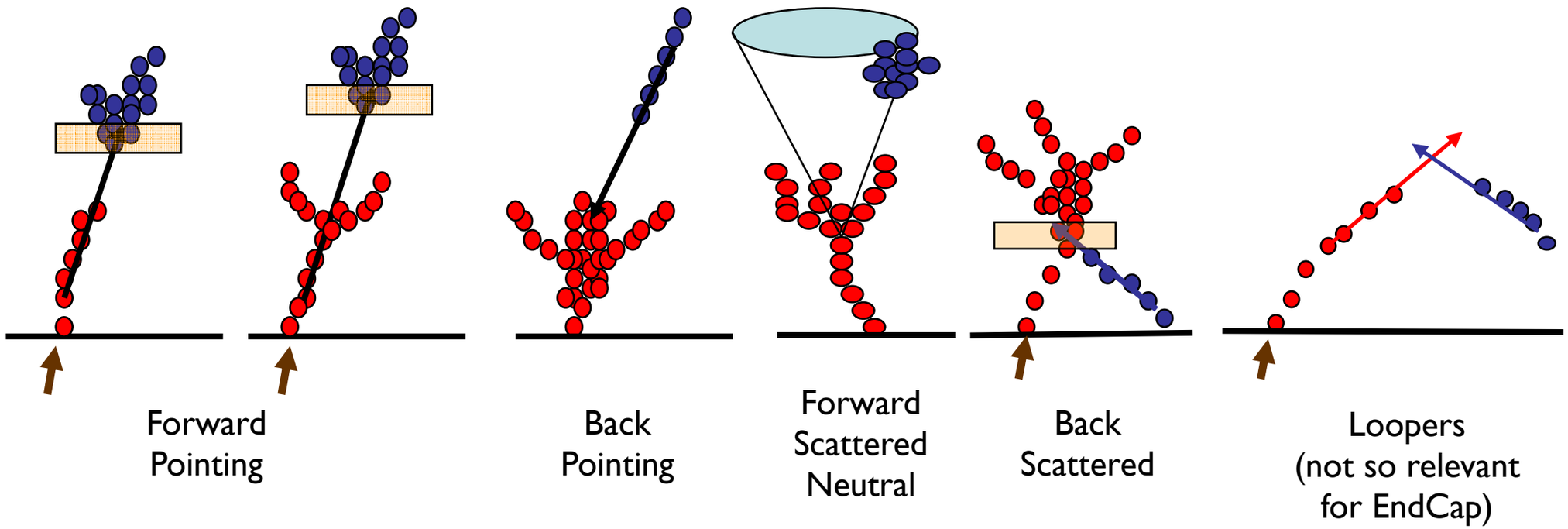




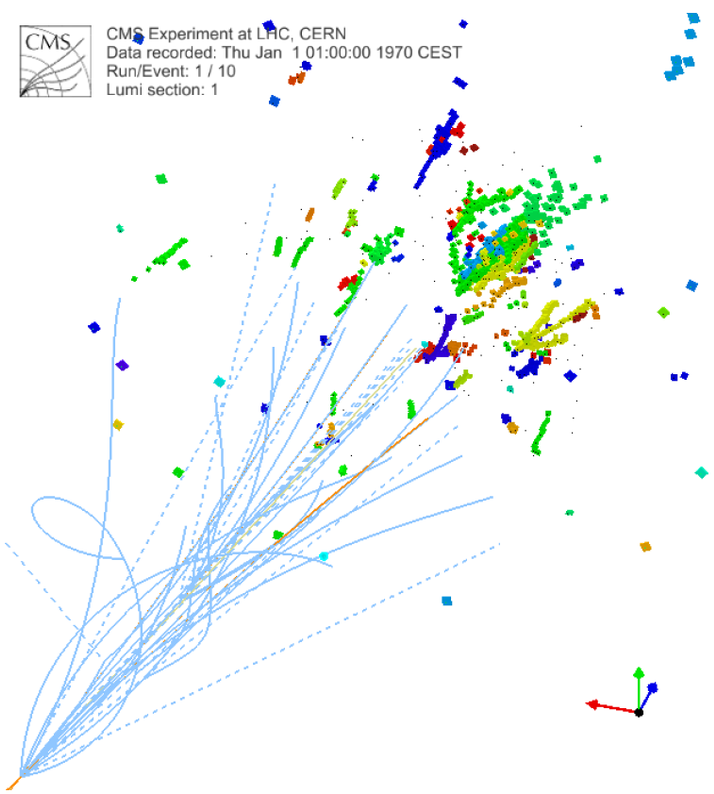


- Since we know that electrons will brem and photons will convert reconstructing a single cluster is not sufficient to precisely or accurately estimate the energies of particles in the CMS calorimeter
  - In order to collect these clusters together into “super clusters” we developed a number of heuristic rules over the years that take into account the spread of charged particles in the CMS magnetic field that must be accounted for when trying to determine a complete representation of a particle

Calorimeter Clustering - Heuristic Rule Based Groupings



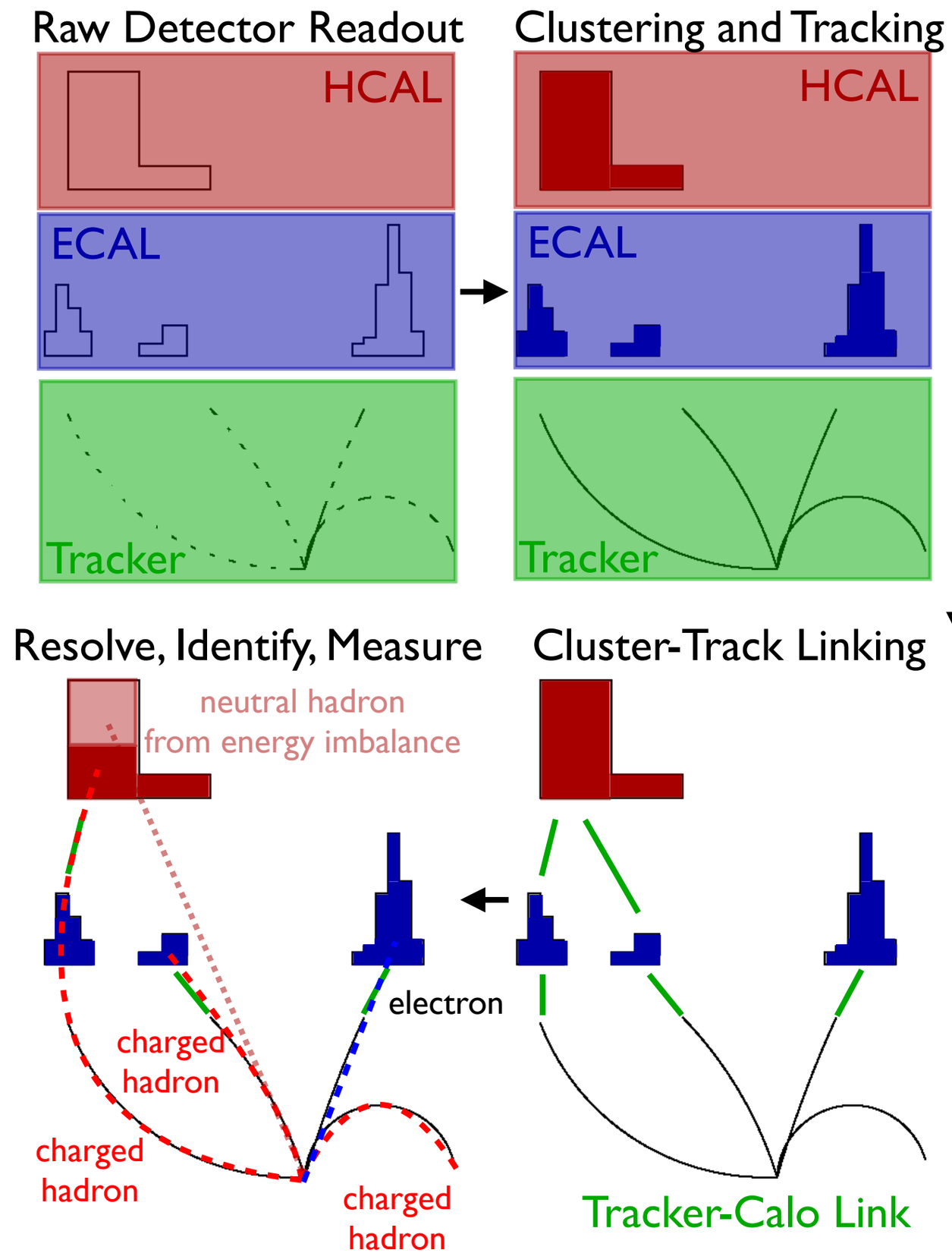
- In very finely grained calorimeters, significantly more shower structure is visible and a much higher degree of accuracy is possible when collecting energy
- However, the array of patterns that are possible is incredibly large and generally only describable by probability distributions or general rules
- We'll get into this in more detail in the third lecture



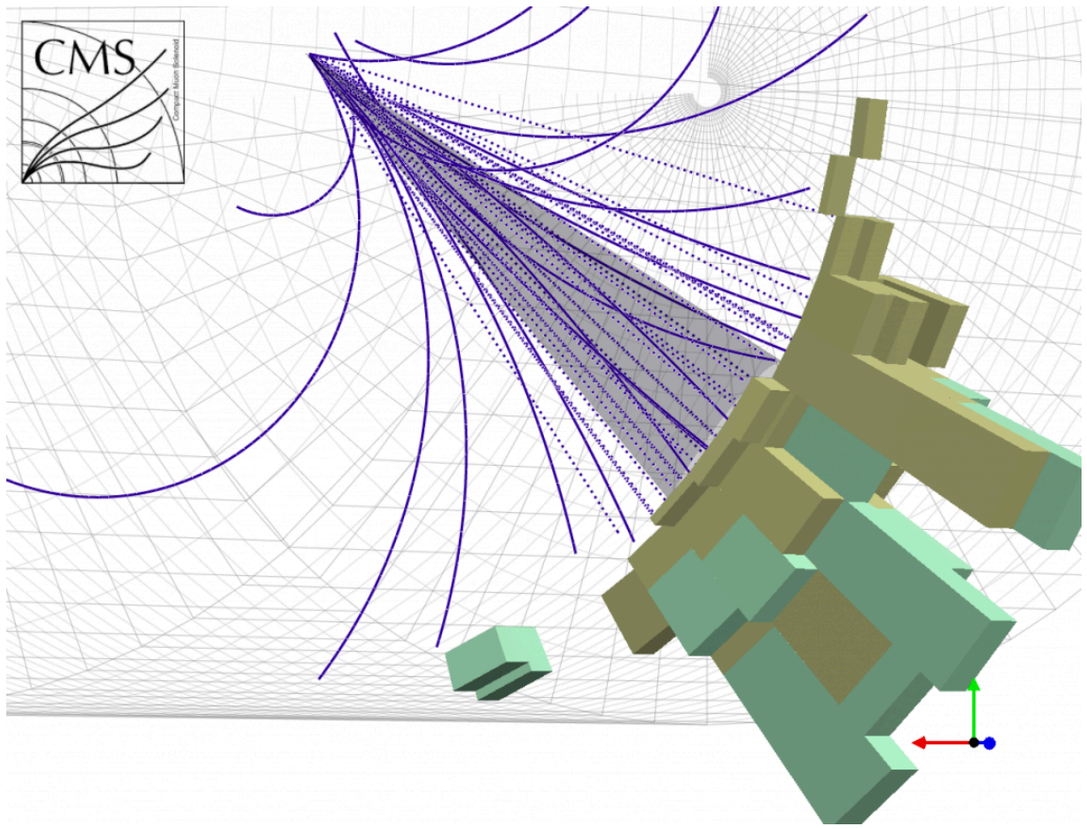
A Word on Particle Flow

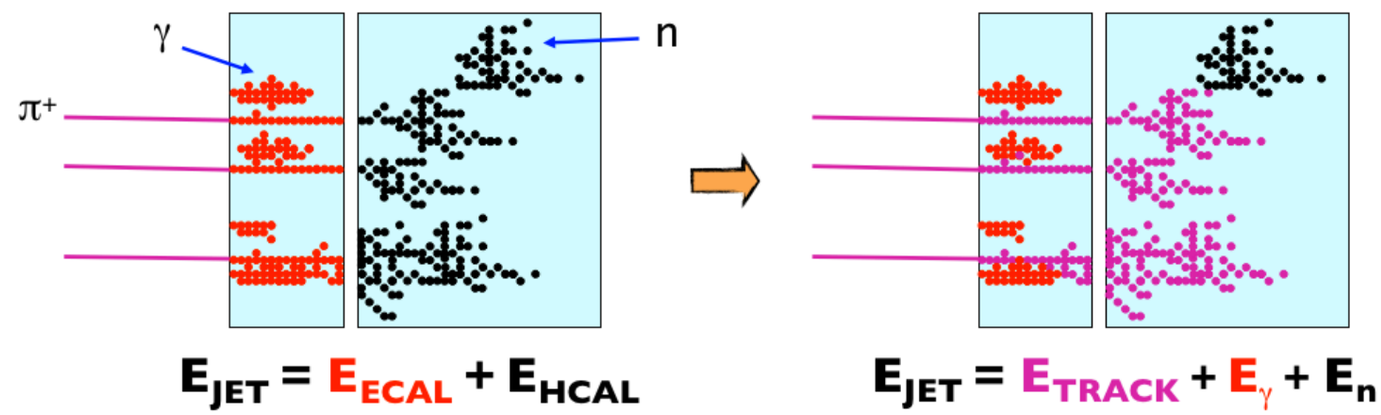
- A reconstruction that yields an unambiguous list of identified final state particles
  - Cluster detector hits together in each detector
  - Link tracking data to calorimetric deposits
    - ~60% of particles in jets are charged hadrons
    - 30% γ, 10% neutral hadrons
    - Augment calorimeter response with tracking
  - Use of all detector information to measure and identify all particles in a collision
    - Optimized use of all information critical to performance for jets and missing energy

- The way this is actually done is too much for this presentation, but I'm happy to answer questions later



Where we are headed with pattern recognition





- Finer granularity yields more precise shower identification
  - At the cost of easily conceptualized energy summing rules (e.g. summing towers top left)
  - Now need algorithms to define bounding volumes, etc…
- Particle Flow algorithms help by associating tracking with calorimetry
  - Can use tracking information to bring additional topological information to clustering
  - Further identification of particles allows precise calibrations to be applied (top right)

Jet Finding - The kT Algorithms

- Jets are created from the hadronization of colored particles produced in collisions
  - Typically there will be one hadron that contains most of the energy from the original quark or gluon, but the species of that hadron is up to random chance
  - Jets are inherently a culmination of the detectors and pattern recognition tools we've discussed so far and this is why particle flow brings an improvement!

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p})\frac{\Delta R^2}{R^2}\begin{cases} p=1 & \text{k}_\text{T} \text{ algorithm} \\ p=0 & \text{Cambridge/Aachen algorithm} \\ p=-1 & \text{anti-k}_\text{T} \text{ algorithm} \end{cases}$$

Procedure of Jet Finding

Calculate particle distance : d_{ij}

Calculate Beam distance : $d_{iB}=k_{ti}^{2p}$

Find smallest distance (d_{ij} or d_{iB})

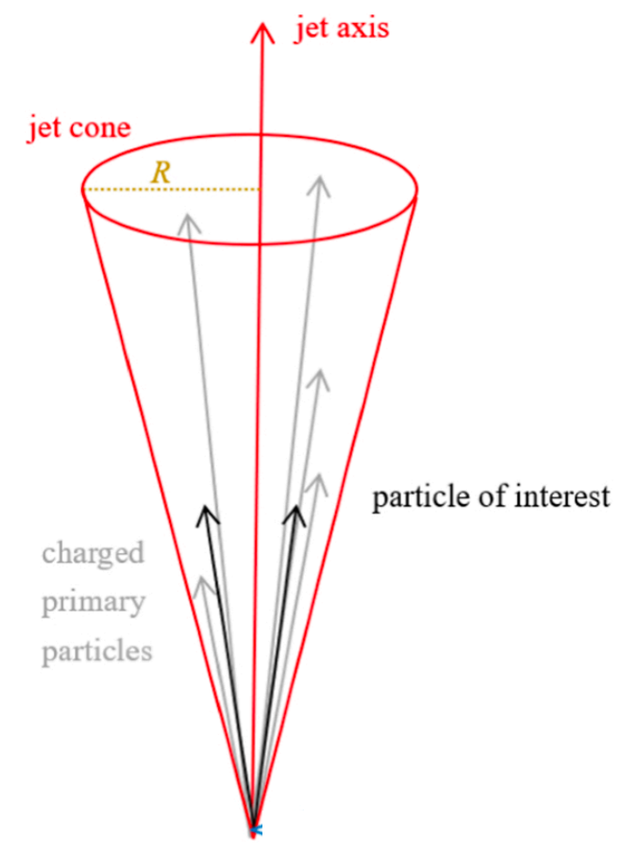
If d_{ij} is smallest combine particles

If d_{iB} is smallest

and the cluster momentum

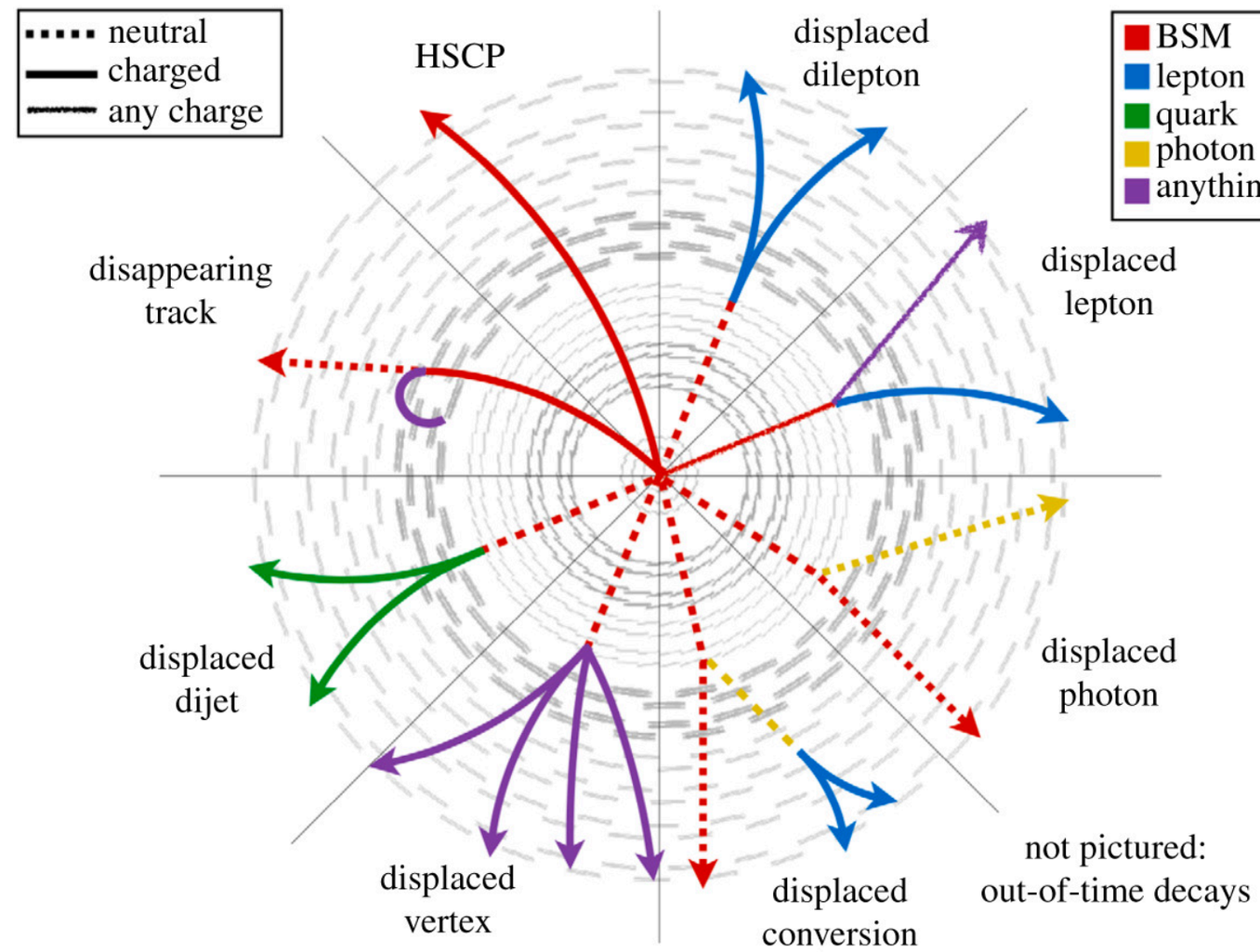
larger than threshold

call the cluster a Jet.



Pattern Recognition Failures and New Physics

- Often our pattern recognition algorithms are made assuming we know what we're trying to find!
  - This is where the line between analysis and "reconstruction" blurs
- Taking the example of long lived particles the tracking algorithms we've talked about can hamper our ability to search for new physics
  - Heavily displaced vertices are hard to find in iterative tracking algorithms
  - Tracks that "stop" are often thrown away
  - Highly ionizing particles can cause problems in detector readouts and spoil the pattern!



Closing Remarks

- This is really a survey of all the tools that are commonly used for reconstruction in particle physics
  - For tracking - the Kaman filter and Gaussian Sum Filter
  - Vertexing - simulated annealing
  - Calorimeter clustering - gaussian mixture modelling and various heuristic algorithms
  - Particle Flow - a collection of heuristics for considering the whole event
  - Jet Finding - Infrared / collinear safe algorithms for grouping hadronized particles

- These are the essential tools for making high-level physics sense of detector readouts so that we can make inference about the theories we want to study
  - These tools can be applied at any level of analysis and what is considered “reconstruction” now was at one point “analysis”

- We’ll find in the next lectures there are a large number of conceptual and practical overlaps with machine learning
  - Eventually we will see how these can be used together to create new tools!

- Thanks for your attention! Questions?