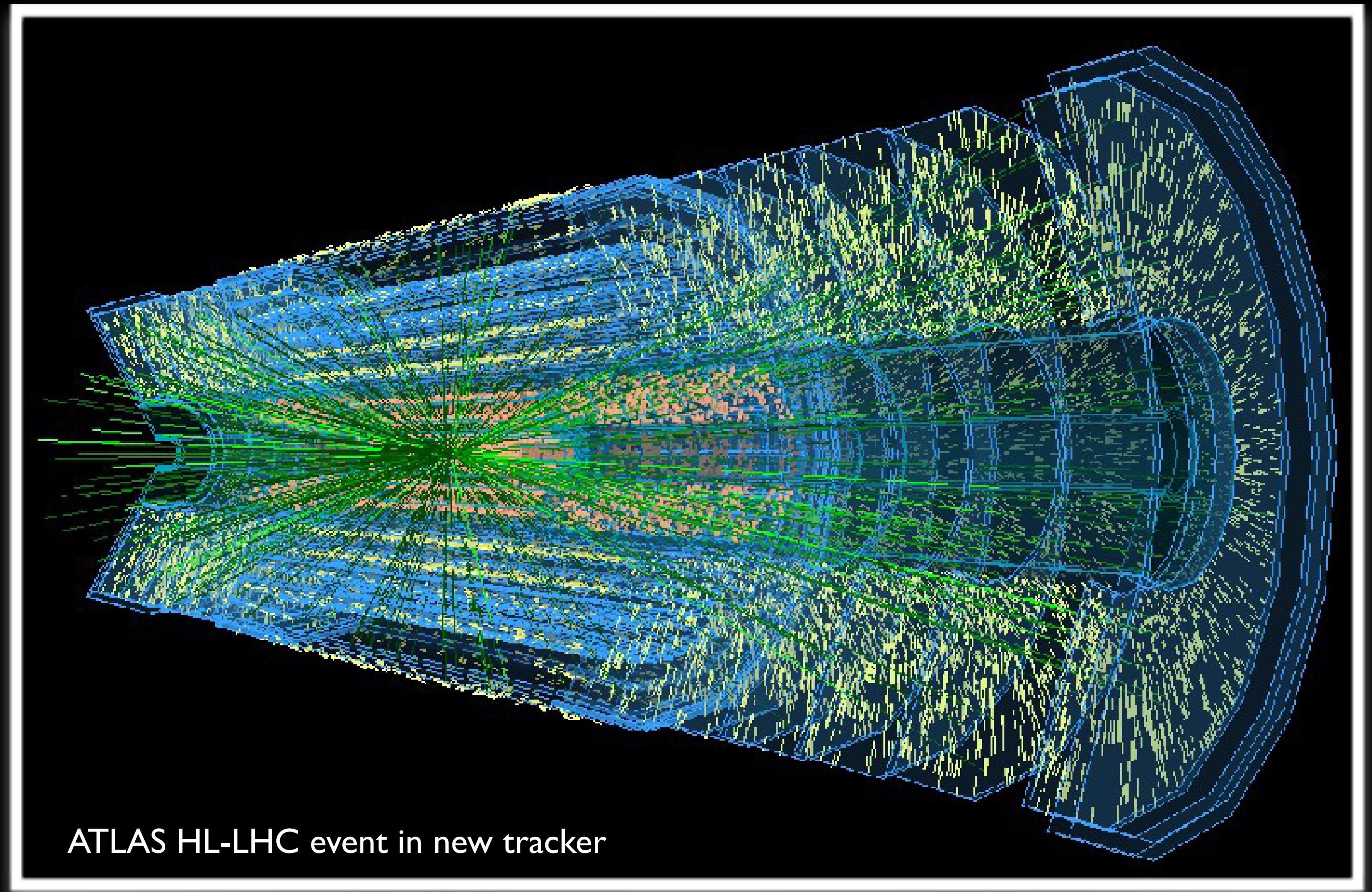
Pattern Recognition in HEP (Track Reconstruction)

Lecture given at the Institute Pascal, Orsay

Markus Elsing, October 14th 2019



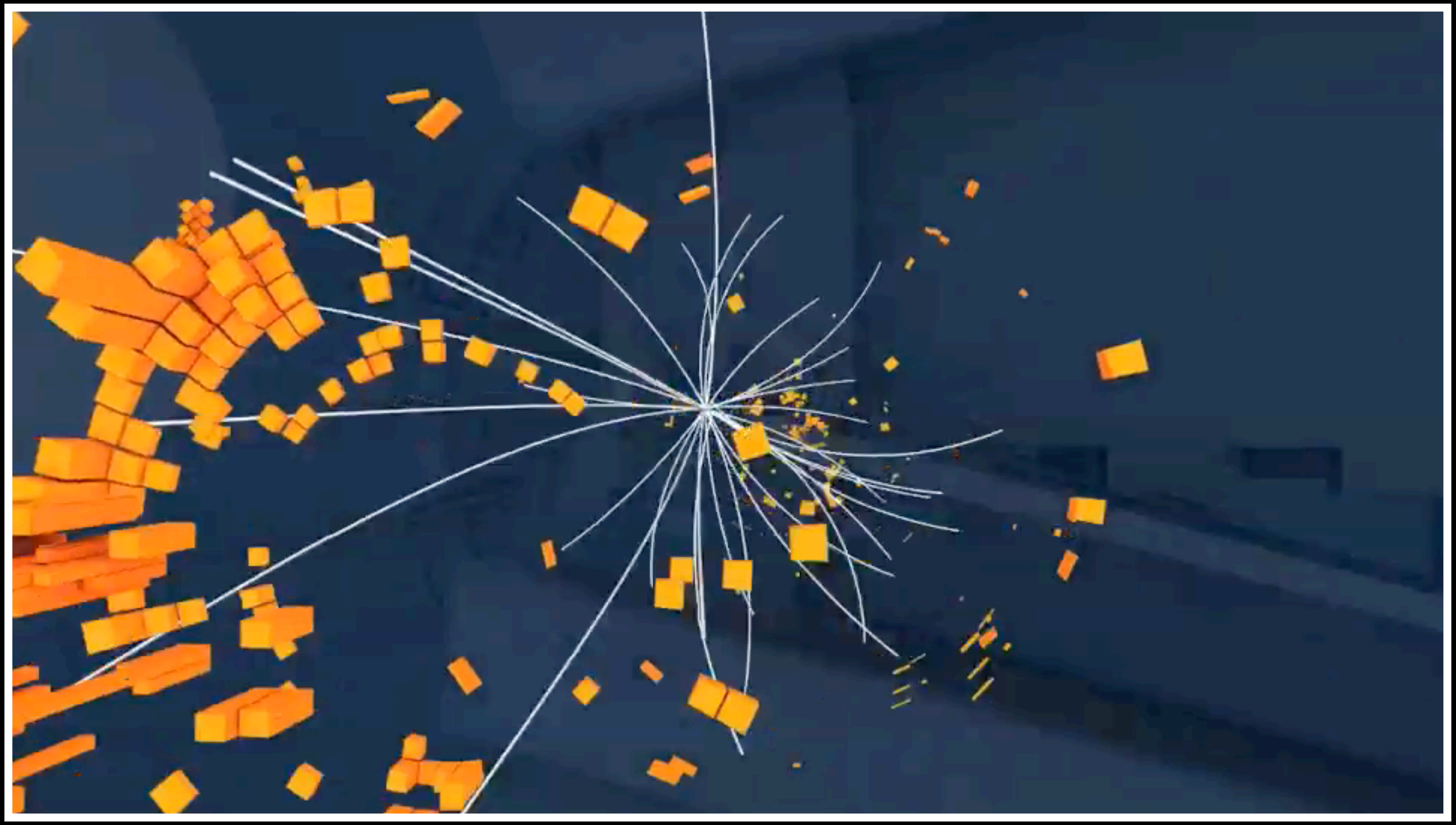
ATLAS HL-LHC event in new tracker

About this Lecture

- This is the first presentation of the Institut Pascal on Learning to Discover : Advanced Pattern Recognition
  - ➡ organisers asked me to give an introduction to classical pattern recognition to set
    the scene for the following in the coming two weeks
- physics of particle detection is well understood
  - ➡ classical reconstruction techniques explicitly explore this knowledge, unlike Machine Learning inspired approaches that deduce it from data
  - ➡ analytical models and sophisticated numerical techniques, developed over ~50 years
- presentation in a style of an introductory lecture
  - ➡ starting from detection principles and how they are exploited by the classical pattern recognition techniques
  - ➡ focus on track reconstruction, following the example of the offline software of the ATLAS experiment (personal bias)

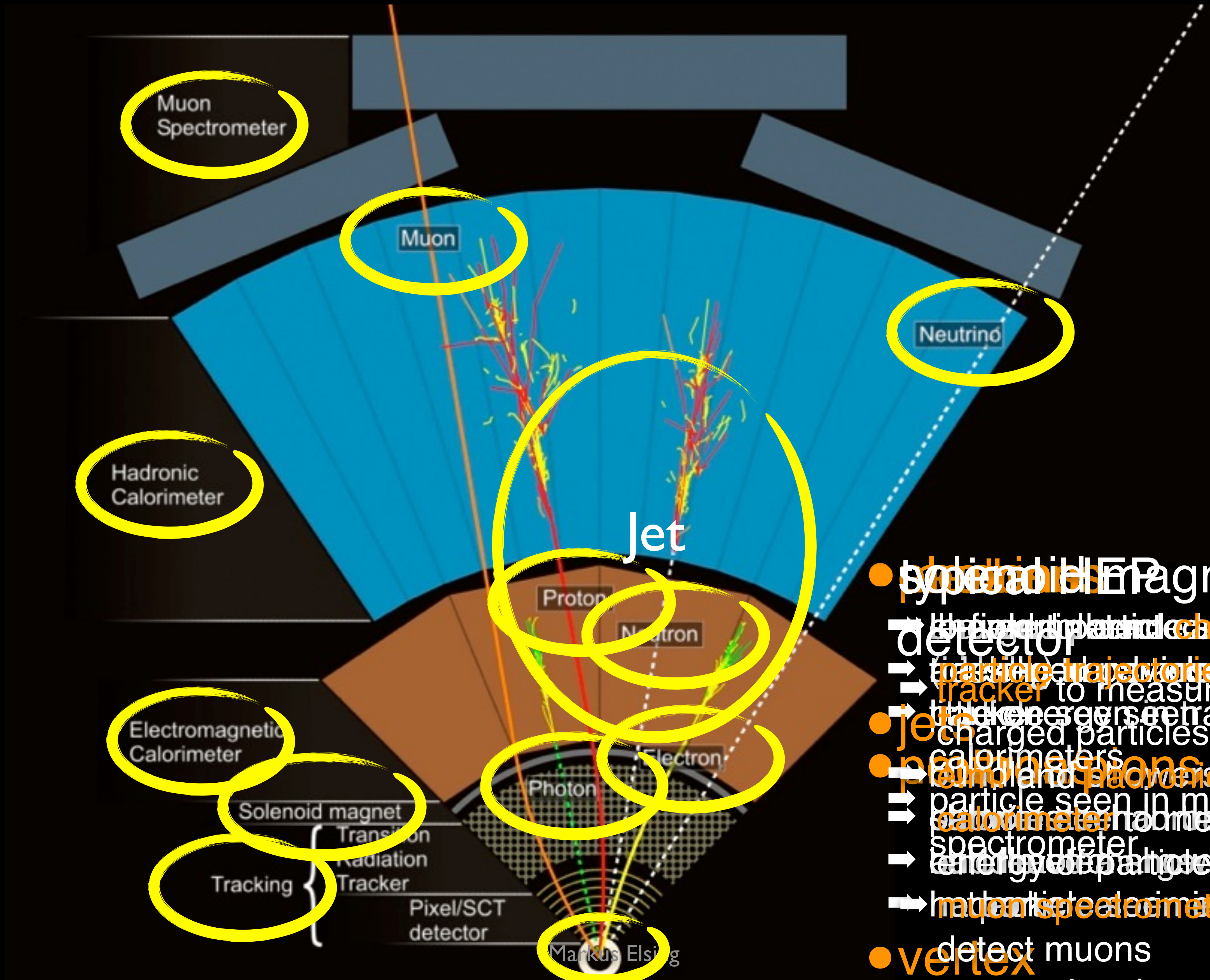
Introduction

Event Detection and Reconstruction



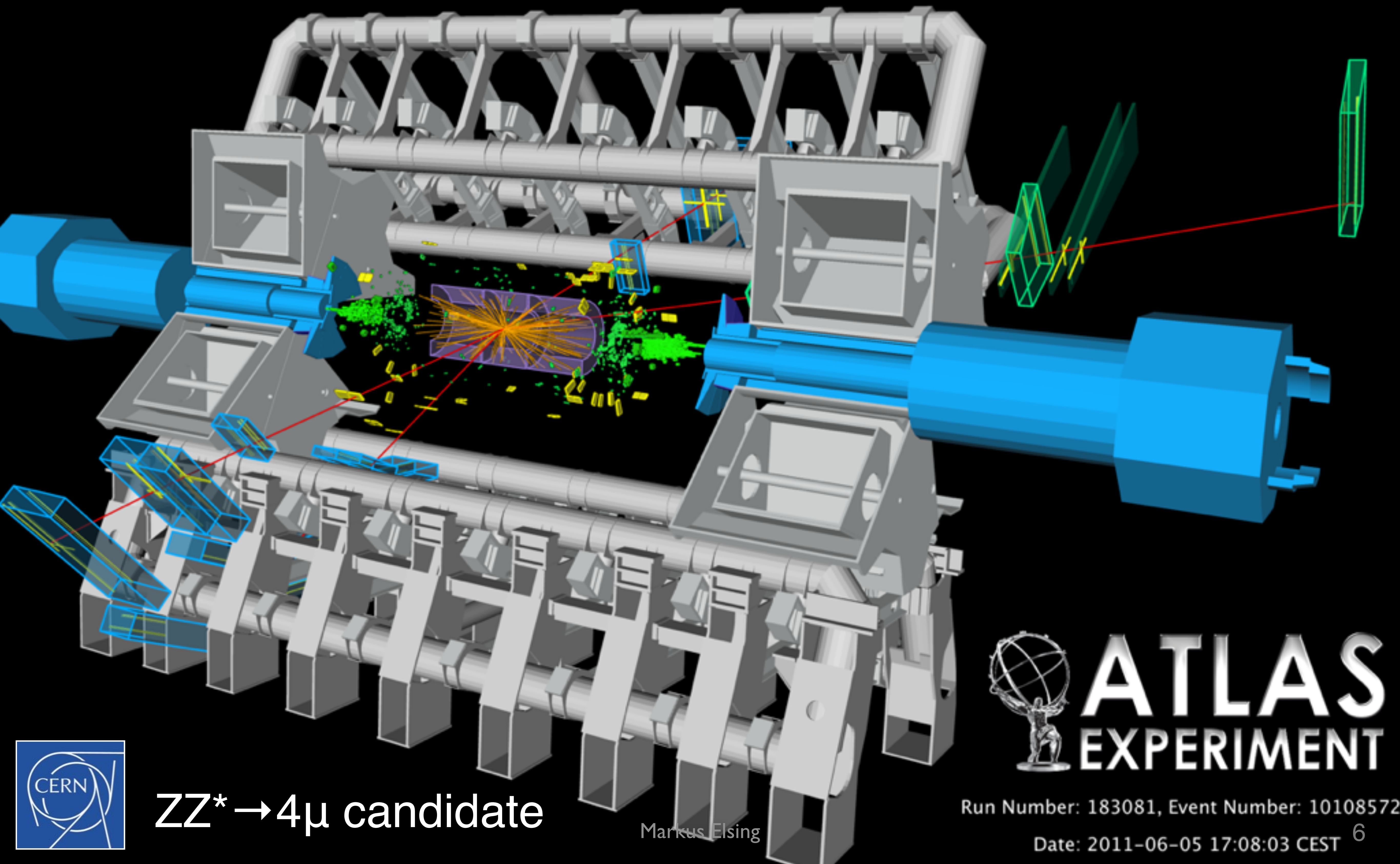
- ➡ LHC experiments are giant "cameras" to take "pictures" of p-p collisions
  - taking a picture every 25 nsec (40 MHz) with 100 million channels
- ➡ task of the reconstruction is the interpretation of the picture !
  - answer the question: which particles were produced ?

Event Reconstruction “in a Nutshell”



In Reality ?

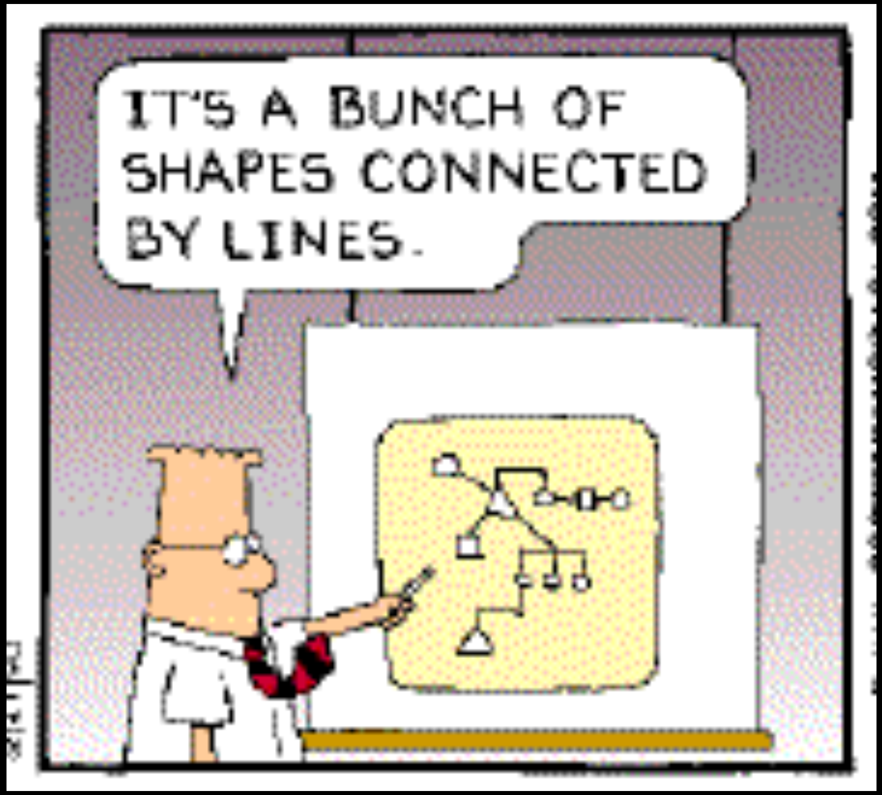
... a bit more complicated



ZZ*→4μ candidate

Introduction to Track Reconstruction

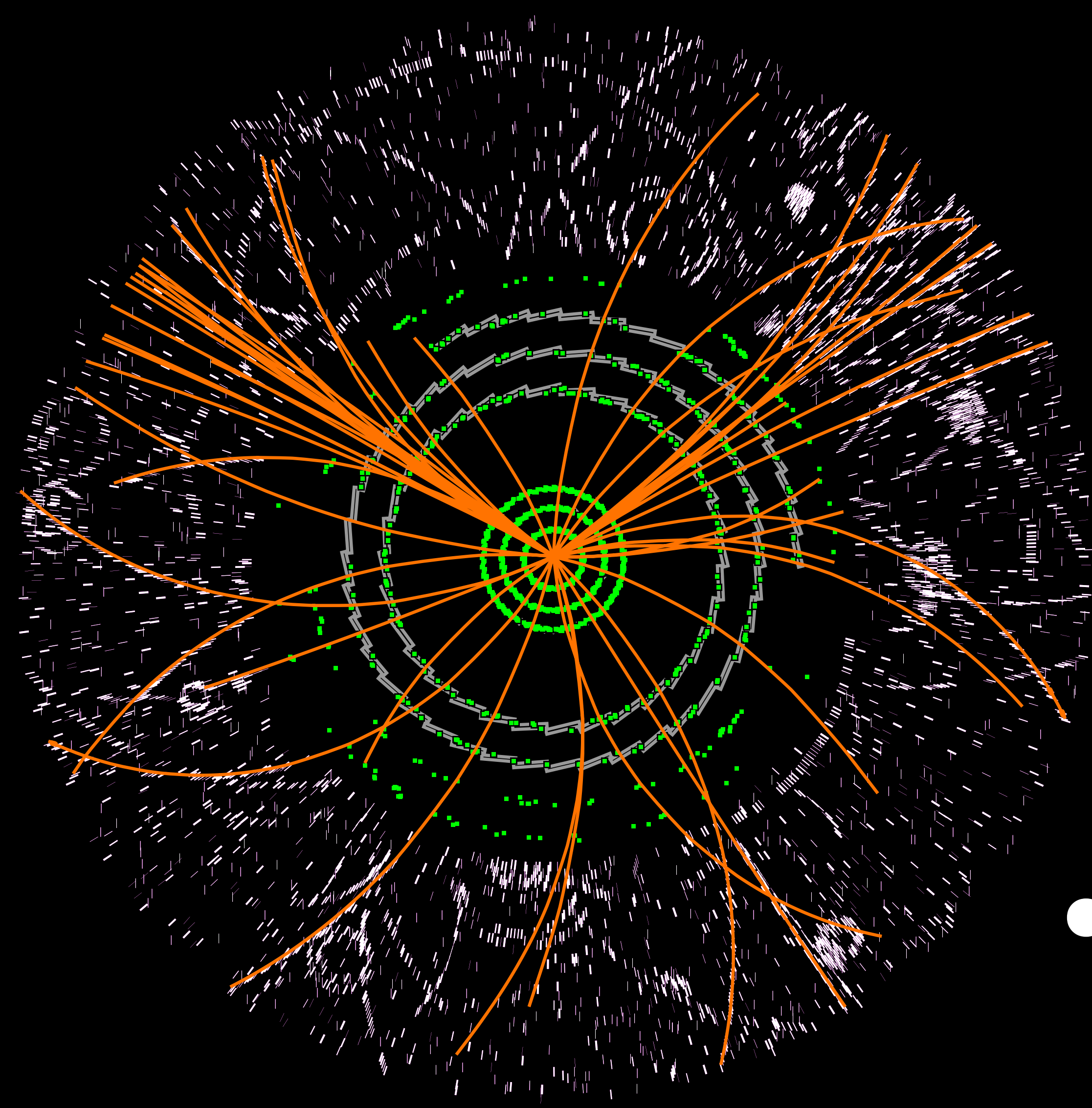
- in this lecture I will discuss the most complex and CPU consuming aspect of event reconstruction at the LHC
  - ➡ finding trajectories (tracks) of charged particles produced in p-p collisions
- will have to introduce various techniques for
  - ➡ pattern recognition, detector geometry, track fitting, extrapolation ...
  - ➡ including mathematical concepts and aspects of software technology



... so why does it matter ?

The Tracking Problem

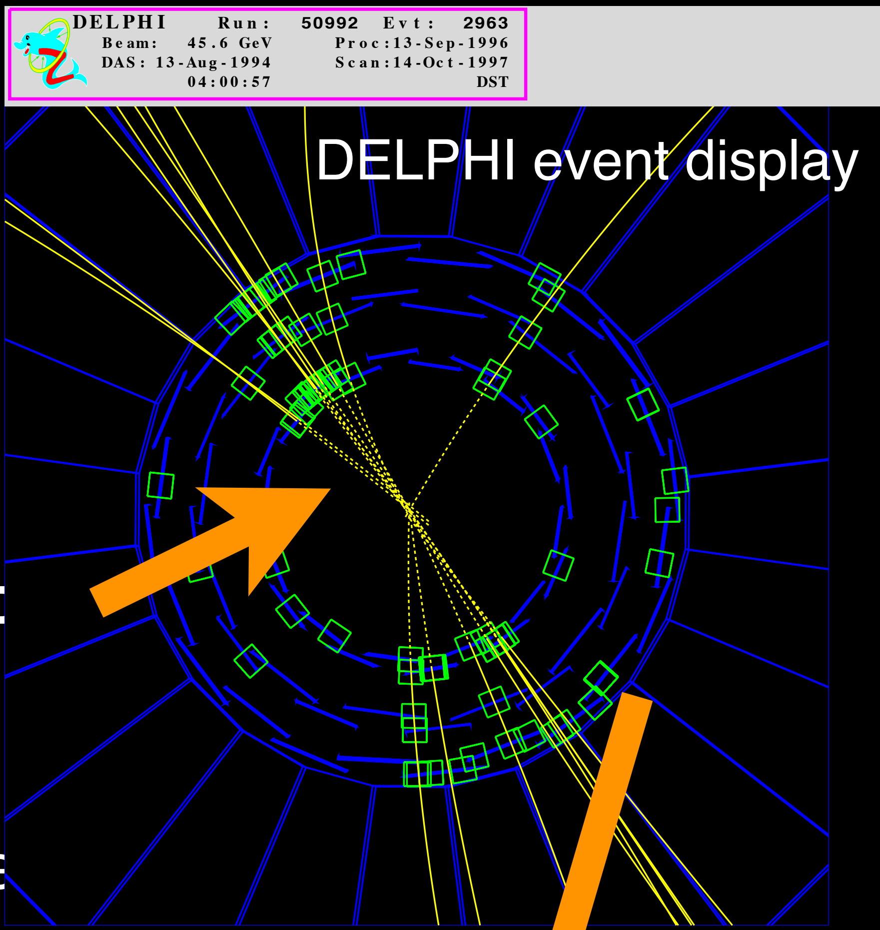
- particles produce in a p-p interaction leave a cloud of hits in the detector



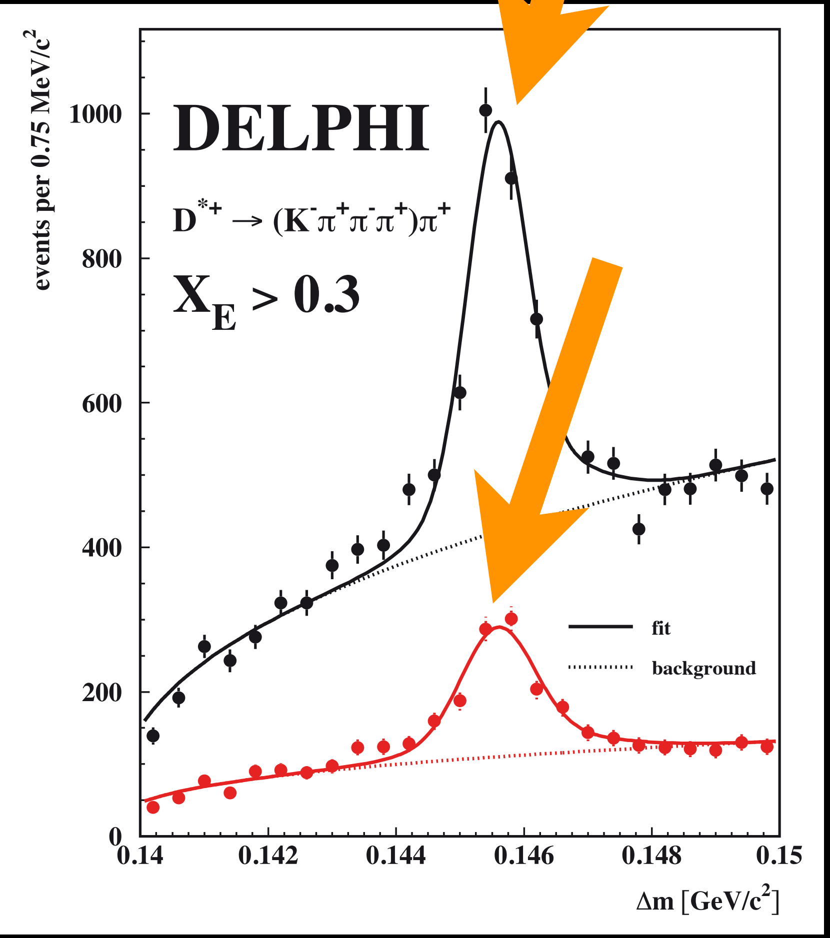
- tracking software is used to reconstruct their trajectories

Role of Tracking Software

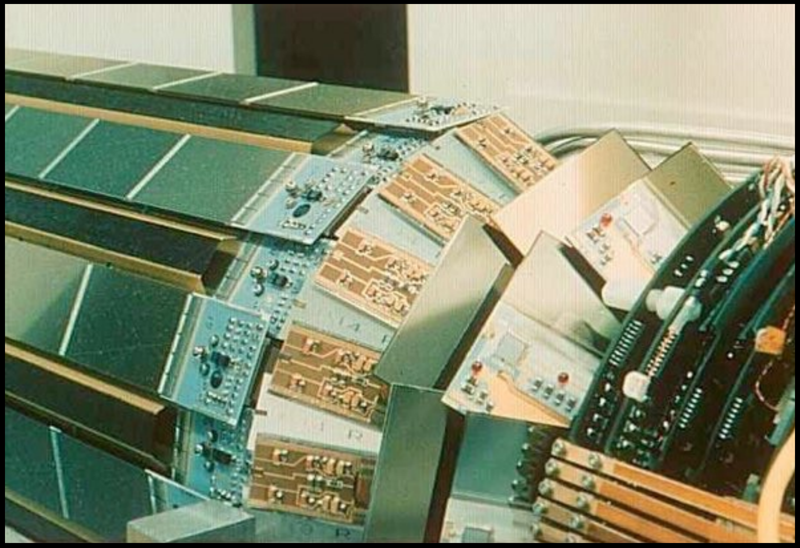
- optimal tracking software
  - ➡ required to fully explore performance of detector
- example: DELPHI Experiment at LEP
  - ➡ silicon vertex detector upgrade
    - initially not used in tracking to resolve dense jets
    - pattern mistakes in jet-chamber limit performance
  - ➡ 1994: redesign of tracking software
    - start track finding in vertex detector
  - ➡ factor ~ 2.5 more D* signal after reprocessing

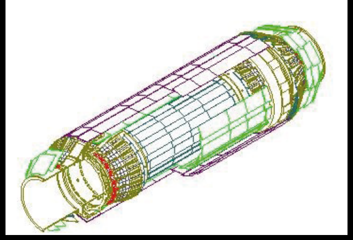


DELPHI event display



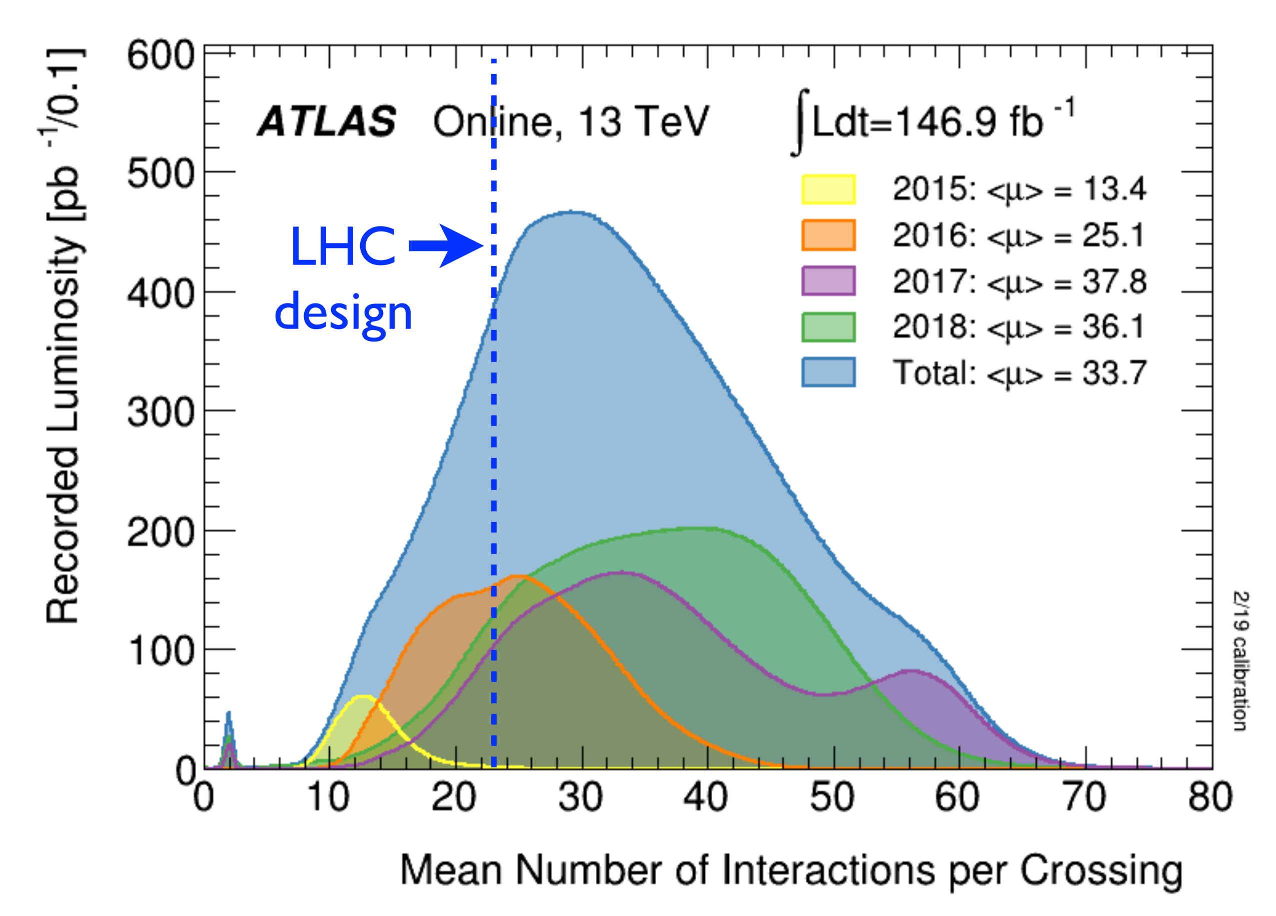
(M.E. et al)

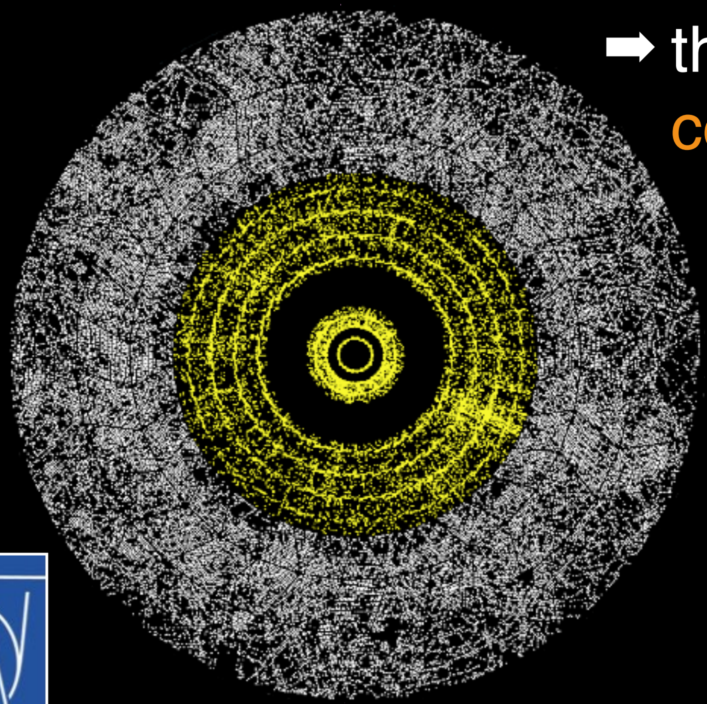

DELPHI vertex detector



Tracking at the LHC ?

- ➡ LHC is a high luminosity machine
  - proton bunches collide every 25 nsec in experiments
  - each time > 20 p-p interactions are observed ! (event pileup)
- ➡ our detectors see hits from particles produced by all > 20 p-p interactions
  - ~100 particles per p-p interaction
  - each charged particle leaves ~50 hits





- ➡ this is how 1 pp collisions looks like
  - now imagine 50 of them overlapping
  - task of tracking software is to resolve the mess ...

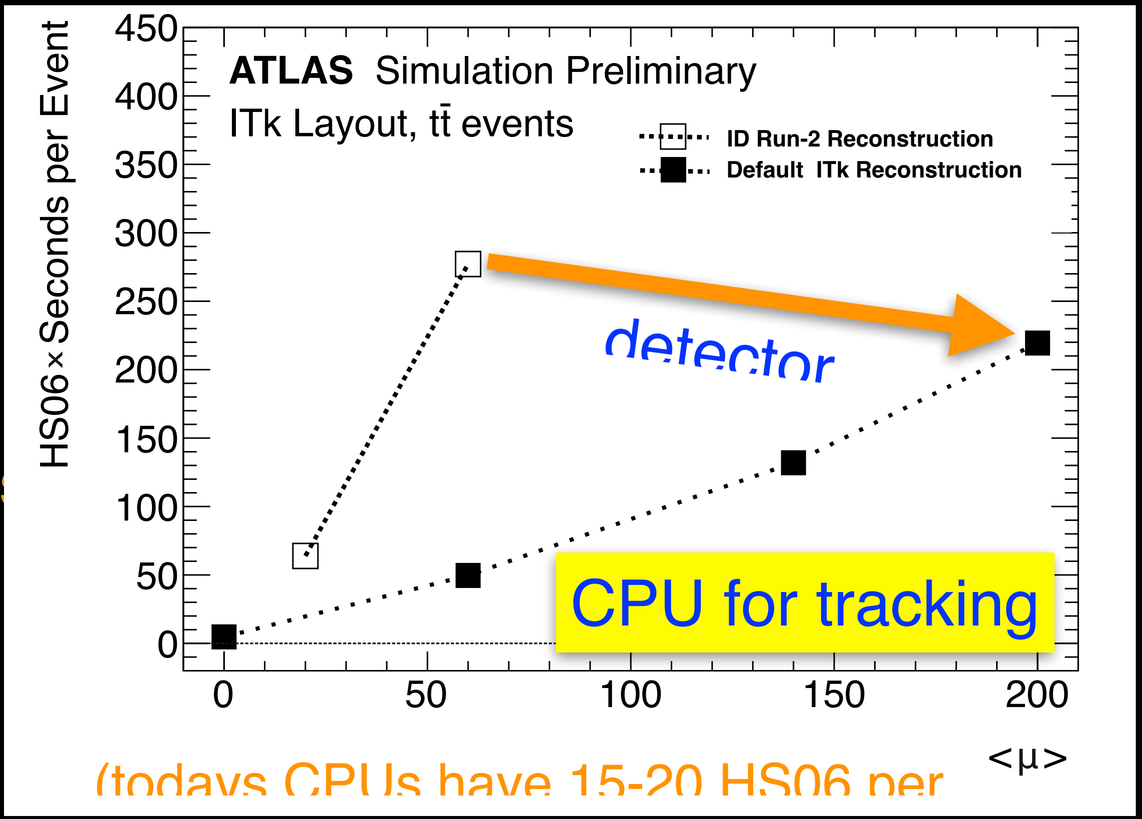
And in the Future (HL-LHC) ?

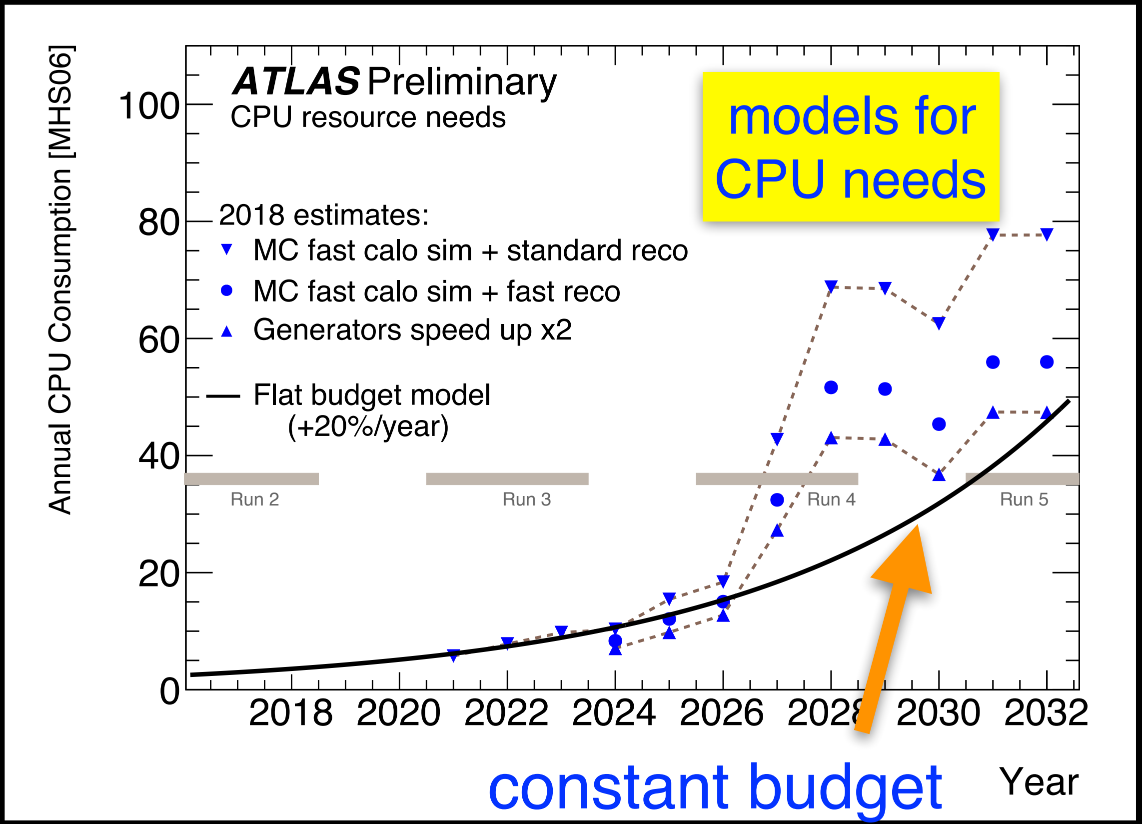
➡ At HL-LHC we expect up to 200 pile-up
- ATLAS+CMS upgrade the tracking systems
- new systems optimised for 200 pile-up

➡ CPU for reconstruction still a major concern !
- requirements may exceed computing budgets
- CPU for tracking is one of the cost drivers !

➡ how to tackle the challenge ?
- better software technology (ACTS projects)
- better algorithmic approaches (incl. ML)

➡ in addition, scientific computing is moving to heterogeneous processing technologies, with more and more HPCs (software challenge)





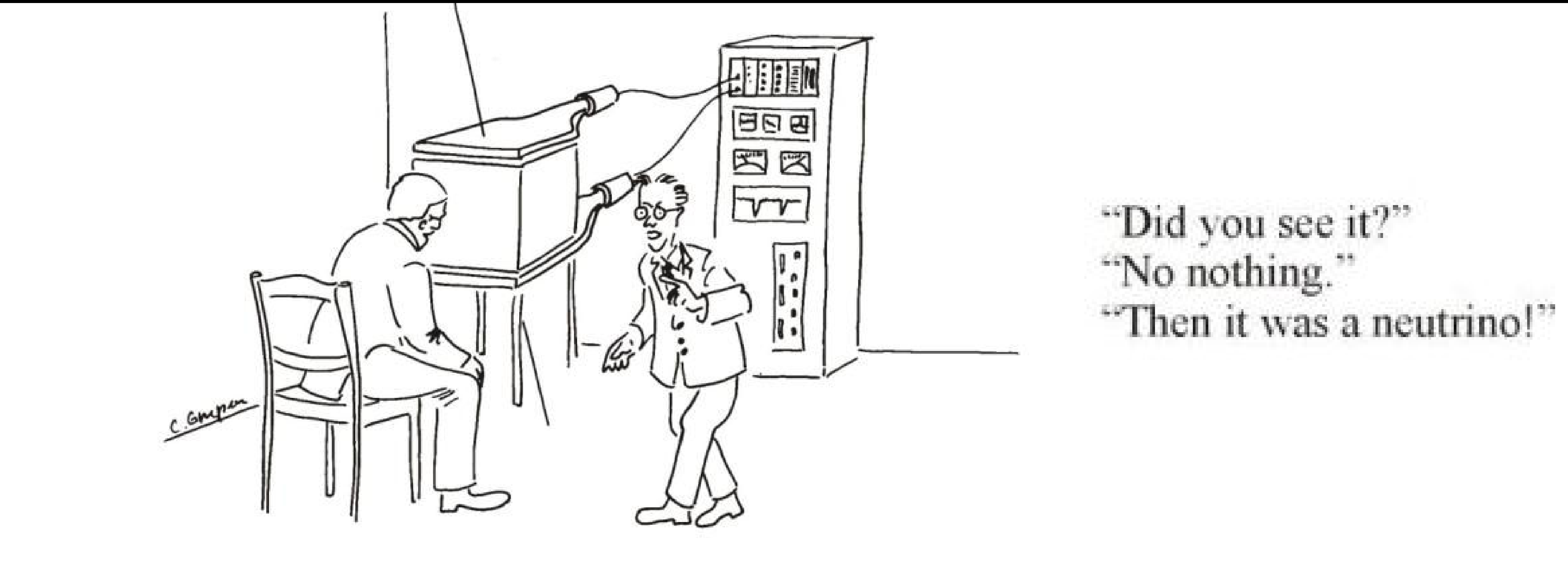
Outline of this Lecture

- Tracking Detectors
  - ➡ principles of semiconductor tracker and drift tubes
- Charged Particle Trajectories and Extrapolation
  - ➡ from trajectory representations to extrapolation toolkits
- Track Fitting
  - ➡ classical least square, Kalman filter and examples for advanced techniques
- Track Finding
  - ➡ search strategies, Hough transforms, progressive track finding, ambiguity solution
- ATLAS Track Reconstruction as an real-life example
  - ➡ putting it all together
- Bonus: Towards HL-LHC

Tracking Detectors

Passage of Particles through Matter

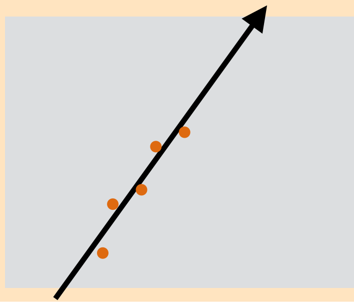
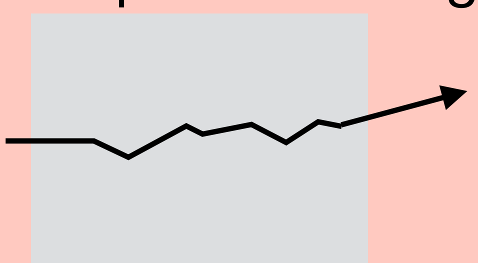
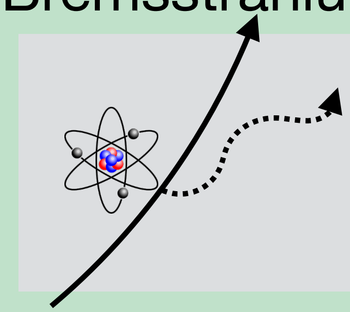
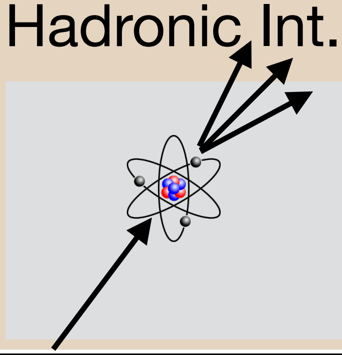
- any device that is to detect a particle must interact with it in some way
  - ➡ well, almost...



Claus Grupen, Particle Detectors, Cambridge University Press, Cambridge 1996 (455 pp. ISBN 0-521-55216-8)

  - ➡ in many experiments neutrinos are detected by missing transverse momentum for completeness

Interactions most relevant to Tracking

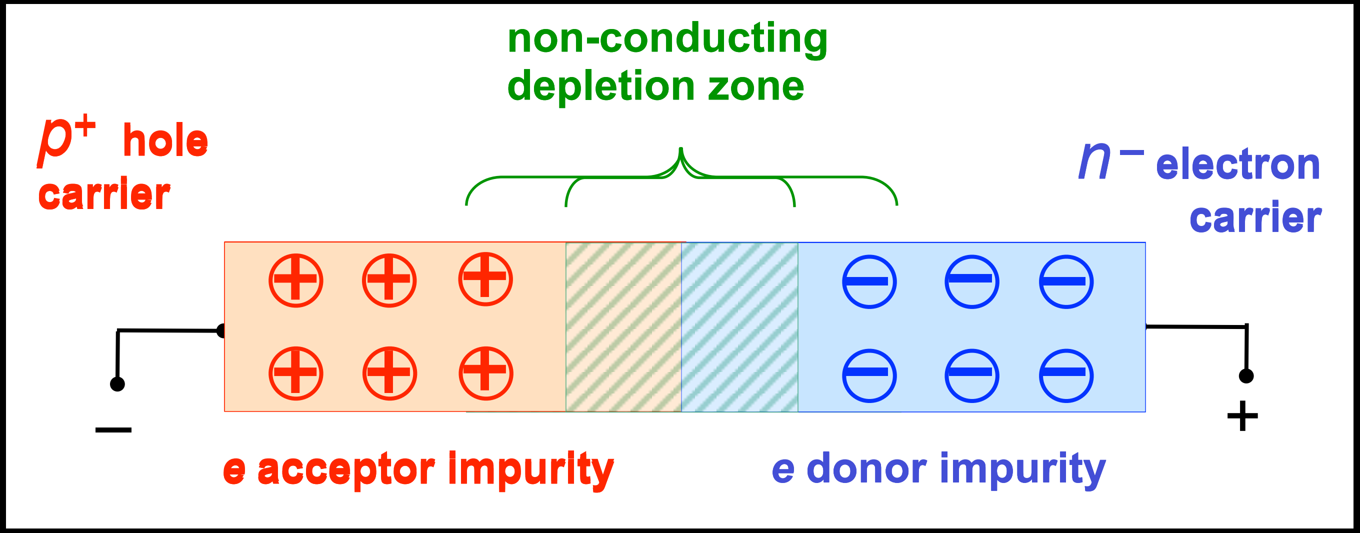
Type	particles	parameter	characteristics	effect
Ionisation loss 	all charged particle	effective density $A/Z * \rho$	small effect in tracker, small dependence on p	increases momentum uncertainty
Multiple Scattering 	all charged particle	radiation length X_0	almost gaussian average effect 0, depends $\sim 1/p$	deflects particles, increases measurement uncertainty
Bremsstrahlung 	all charged particle, dominant for e	radiation length X_0	energy loss proportional $\sim E$, highly non-gaussian, depends $\sim 1/m^2$	introduces measurement bias and inefficiency
Hadronic Int. 	all hadronic particles	nuclear interaction length Λ_0	incoming particle lost, rather constant effect in p	main source of track reconstruction inefficiency

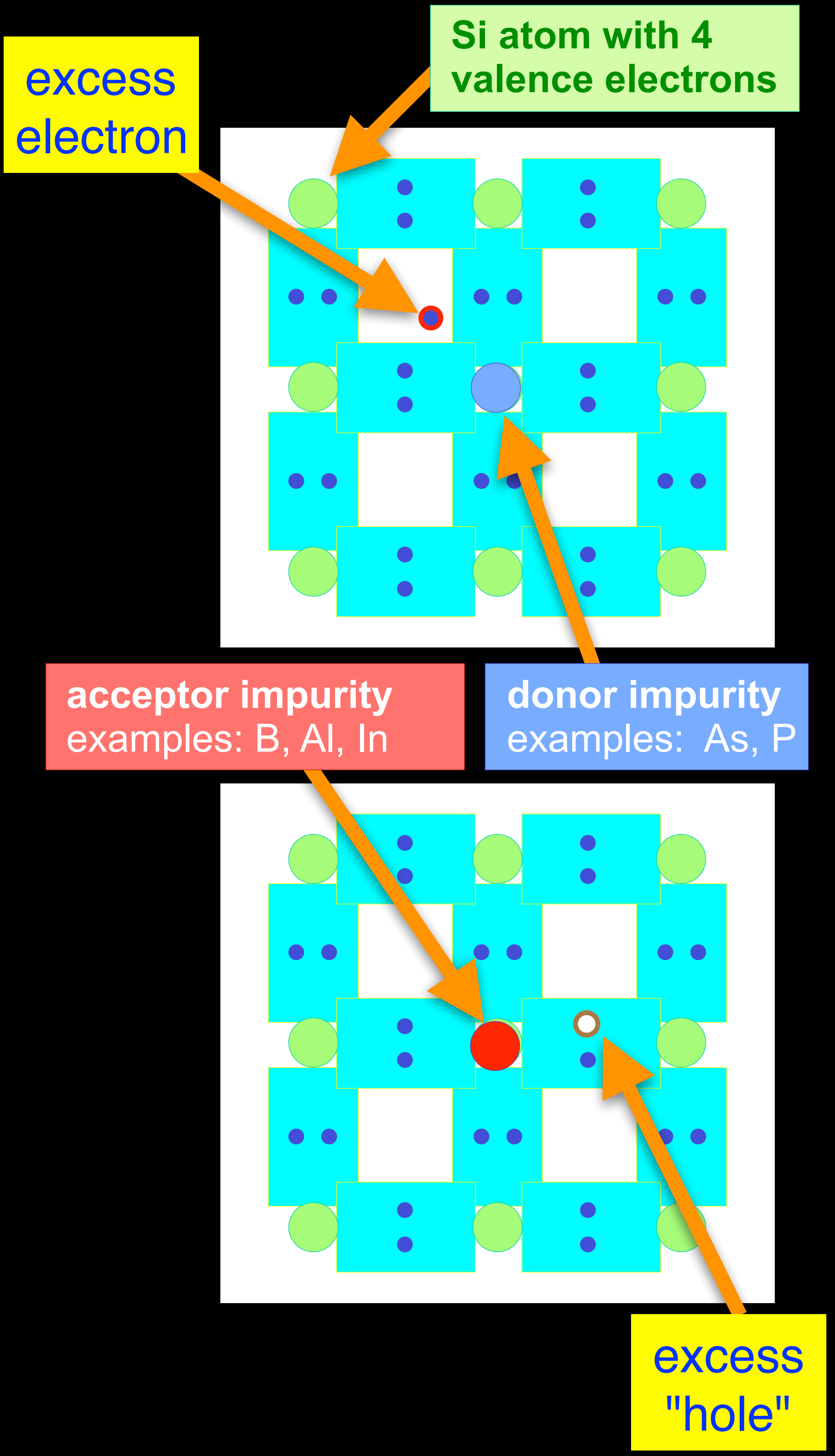
➡ tracking detectors explore effects like ionisation to measure charged particles

- let's discuss the basic principles of semiconductor trackers and drift tubes

Semiconductors as Particle Detectors

- schema of a silicon diode (p-n junction)
  - ➡ doping silicon cristal semiconductor to implant excess electrons or "holes"
    - n doping adds electro-phile atoms
    - p doping adds electro-phobe atoms
  - ➡ both materials together form a diode

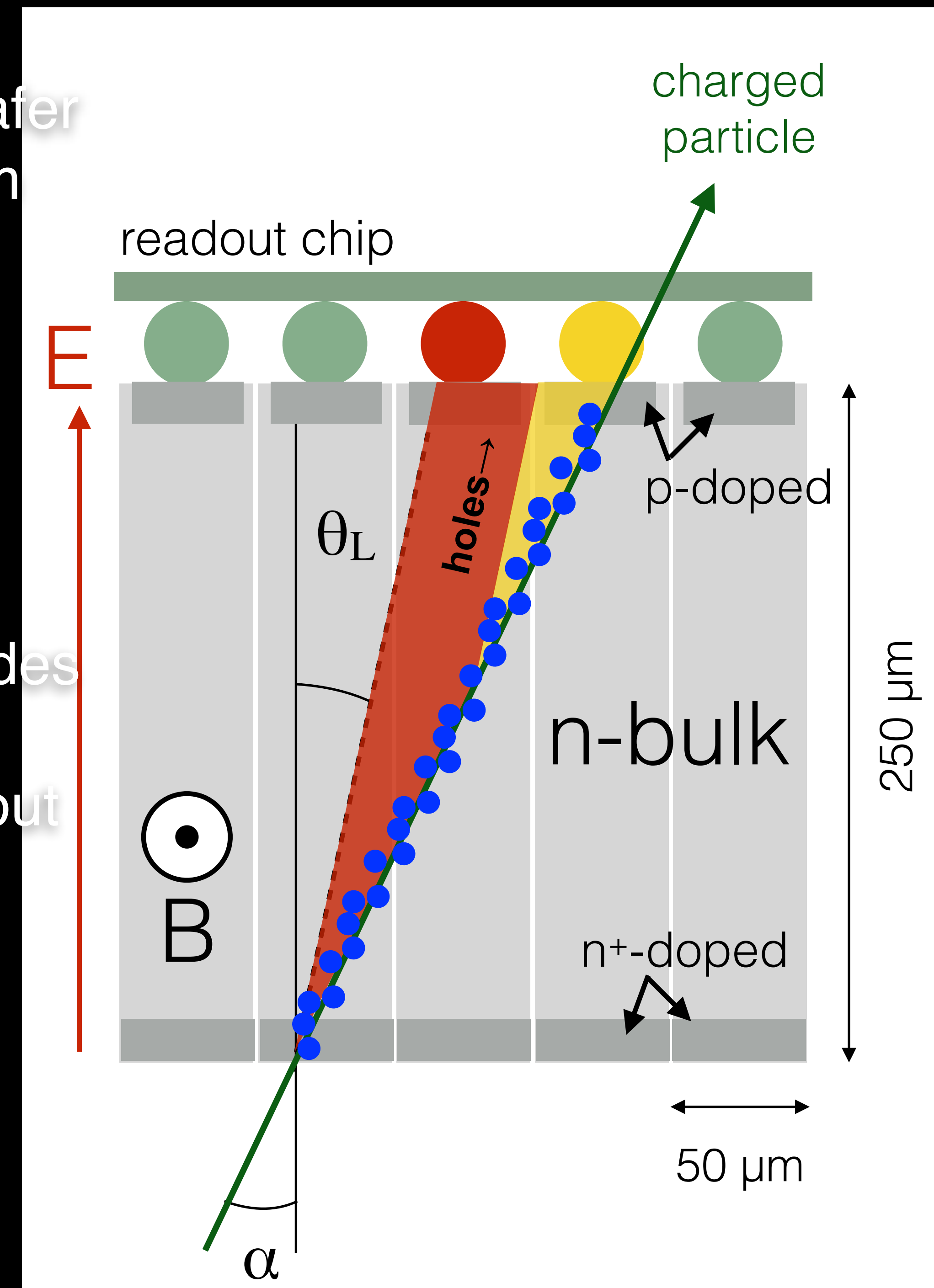


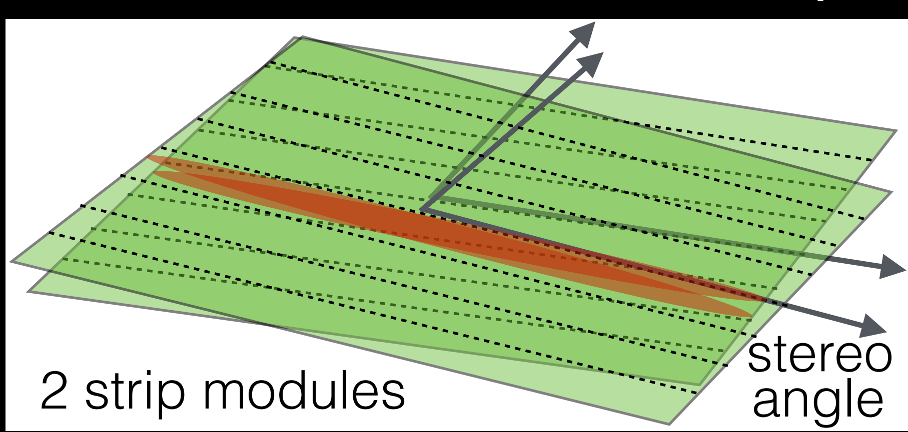


- recombination in junction creates depletion zone, acts as potential barrier against doping potential
- apply reverse bias voltage to enlarge potential barrier in depletion zone, increases its resistance further

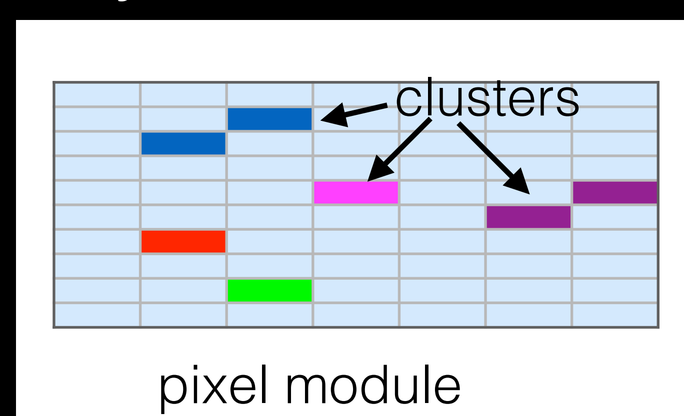
Semiconductors as Particle Detectors

- basic schema of a silicon detector
  - ➡ many reverse biased large diodes on a silicon wafer
    - allows for small structures, typical pitch is 50 µm
  - ➡ traversing charged particle ionises silicon
    - creates electron-hole pairs, drifting in E-field to electrodes leading to measurable signals in
    - diodes
    - Lorentz angle θ_L deflection in presence of B-field
- 2 types: silicon strips and pixels
  - ➡ strip module: 50 µm pitch, wafers with ~6 cm diodes
    - needs 2 modules to measure both coordinates
  - ➡ pixel module: e.g. 50x400 µm pixel, analog readout
    - clusters measures precisely both coordinates





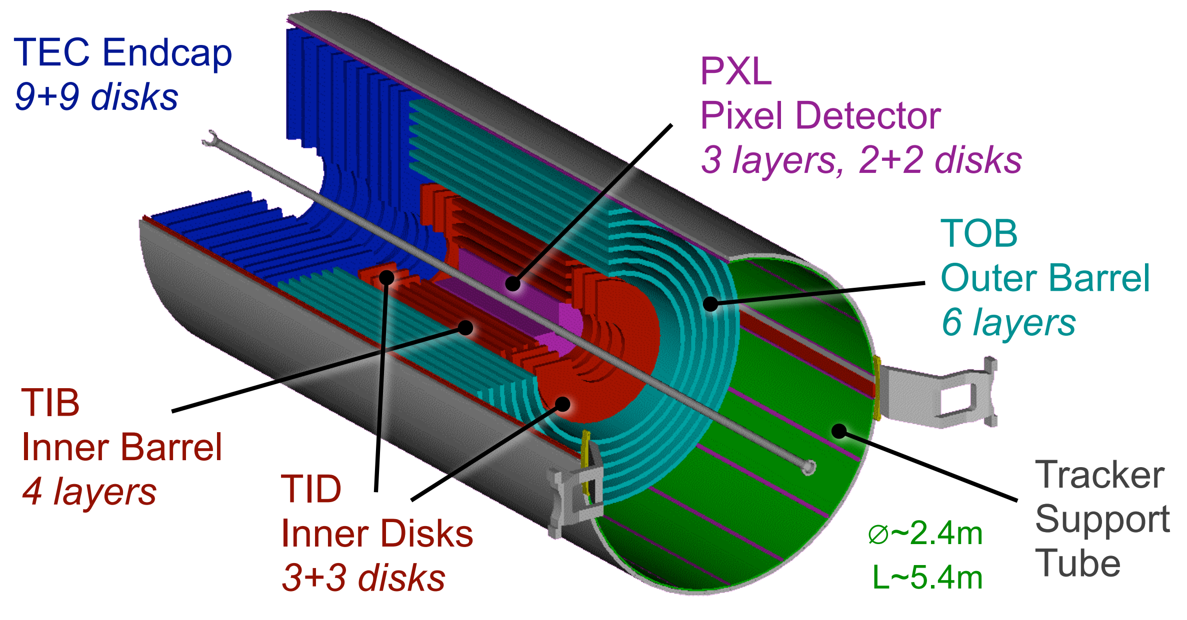
2 strip modules



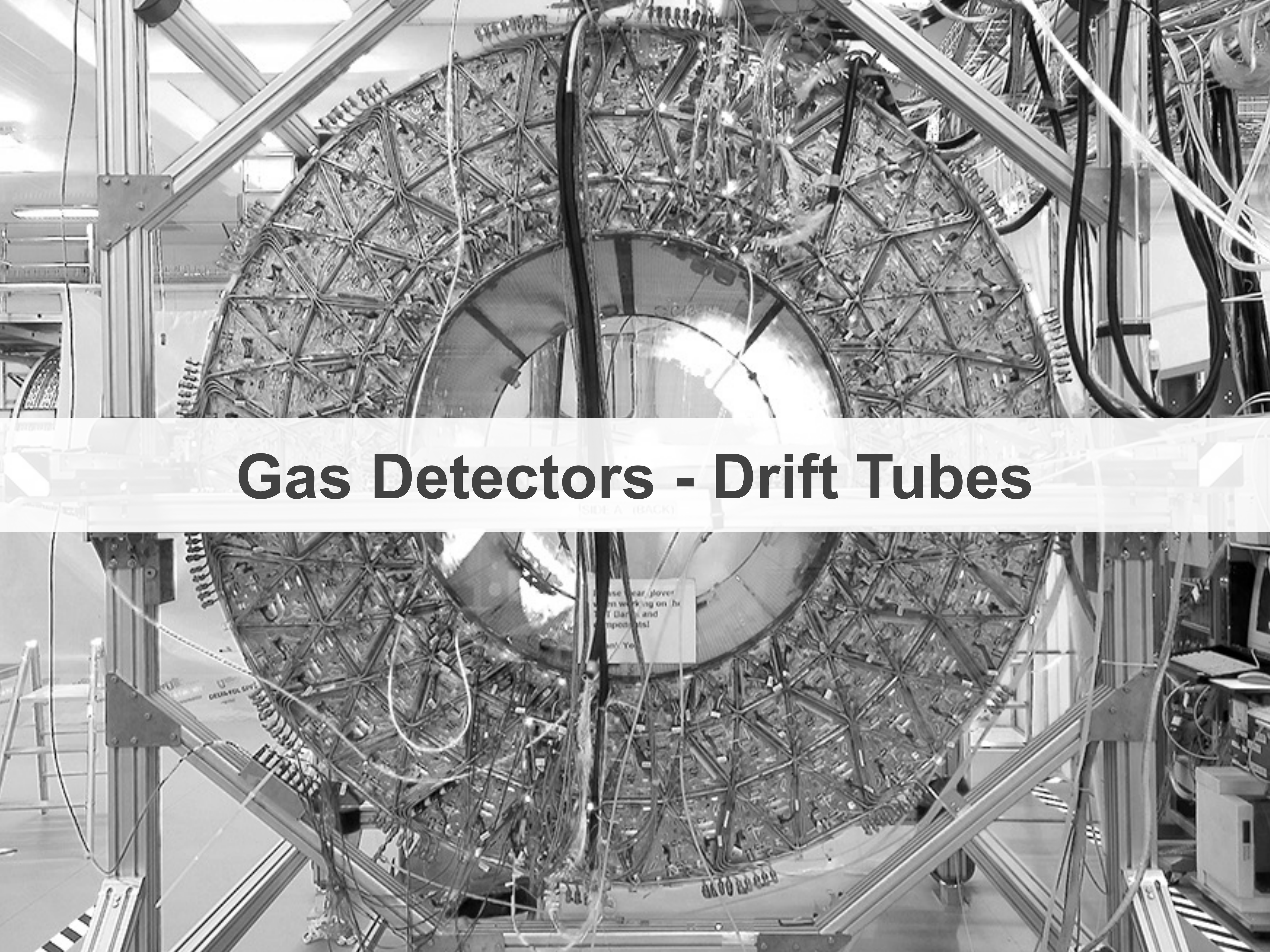
pixel module

CMS Tracker

- largest silicon tracker today
  - ➡ Pixels: 66M channels, 100x150 μm² Pixe
  - ➡ strip detector: ~23m³, 210m² of Si area, 10.7M channels

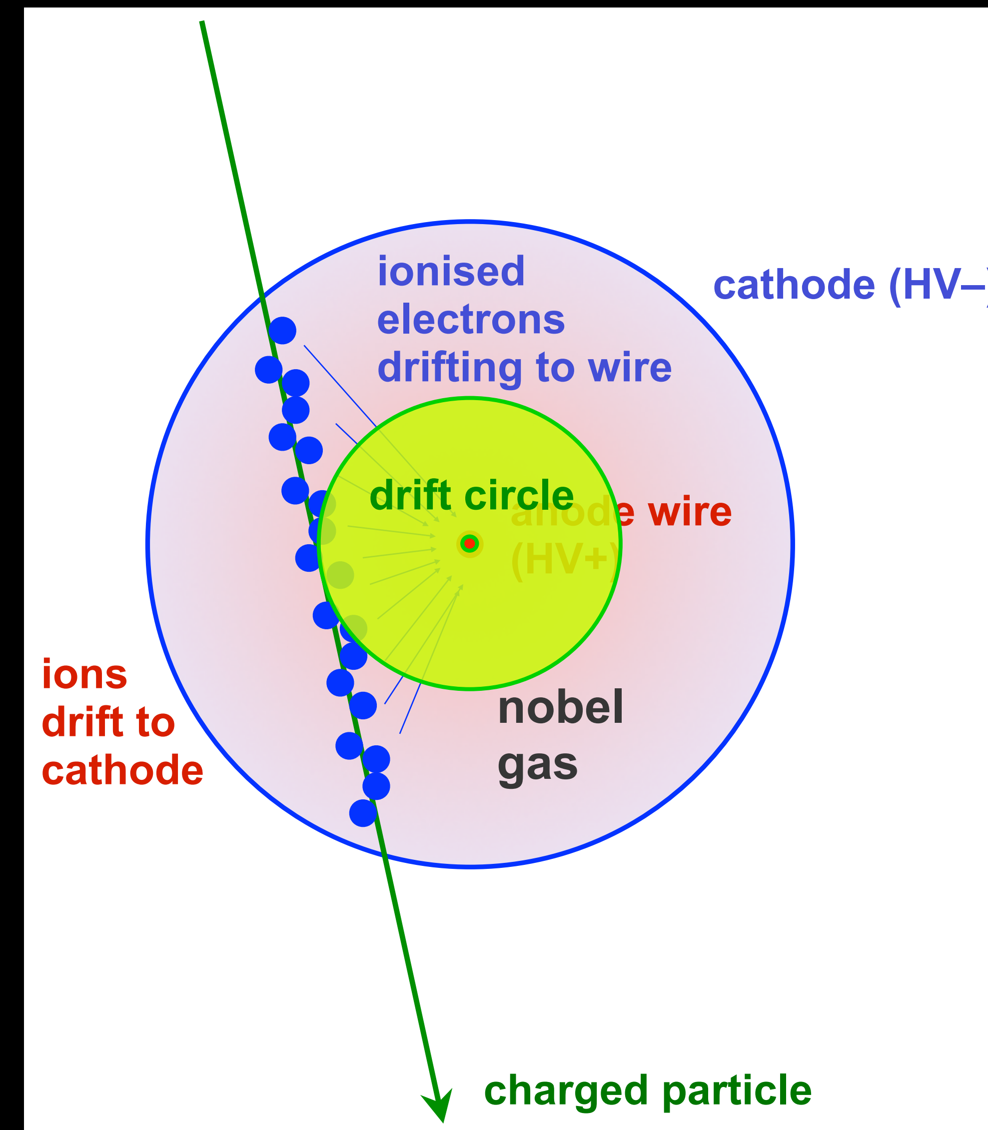


Gas Detectors - Drift Tubes



Classical Gas Detectors - Drift Tubes

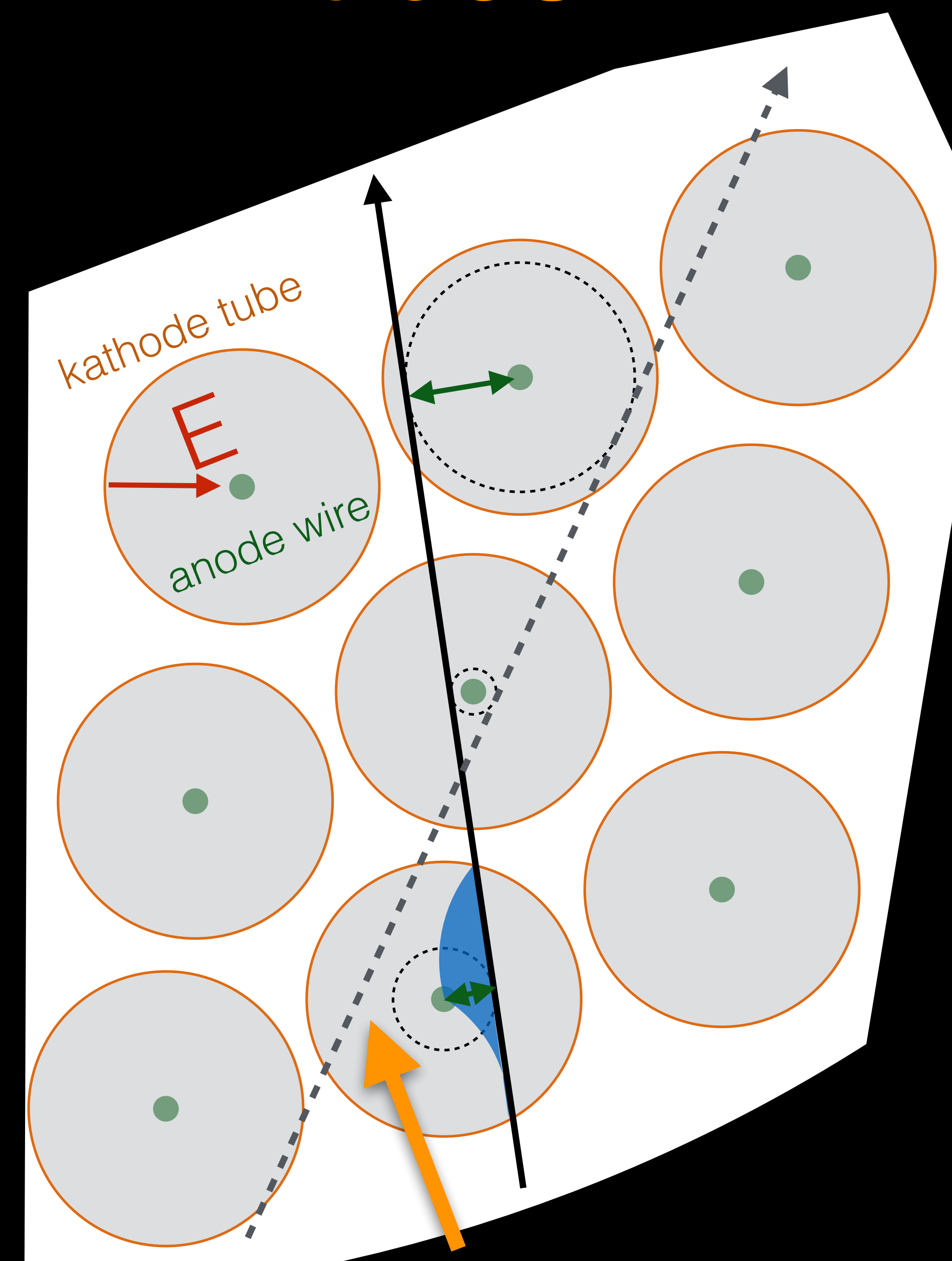
- detection technique for charged particles
  - ➡ used in muon systems and ATLAS TRT
- particles traversing tube ionises the gas
  - ➡ deposited charge drifts to anode wire in electric (E) field
    - charge amplification in high E-field in vicinity of wire leads to large signal pulse
  - ➡ measure time of signal pulse to determine drift circle
    - Lorentz angle deflection in B-field (not shown)
    - fast signal detection (v_D~30 ns/mm)
    - resolution of O(100 μm) on measured radius



TRT: Kapton tubes, ∅ = 4 mm
MDT: Aluminium tubes, ∅ = 30 mm

Classical Gas Detectors - Drift Tubes

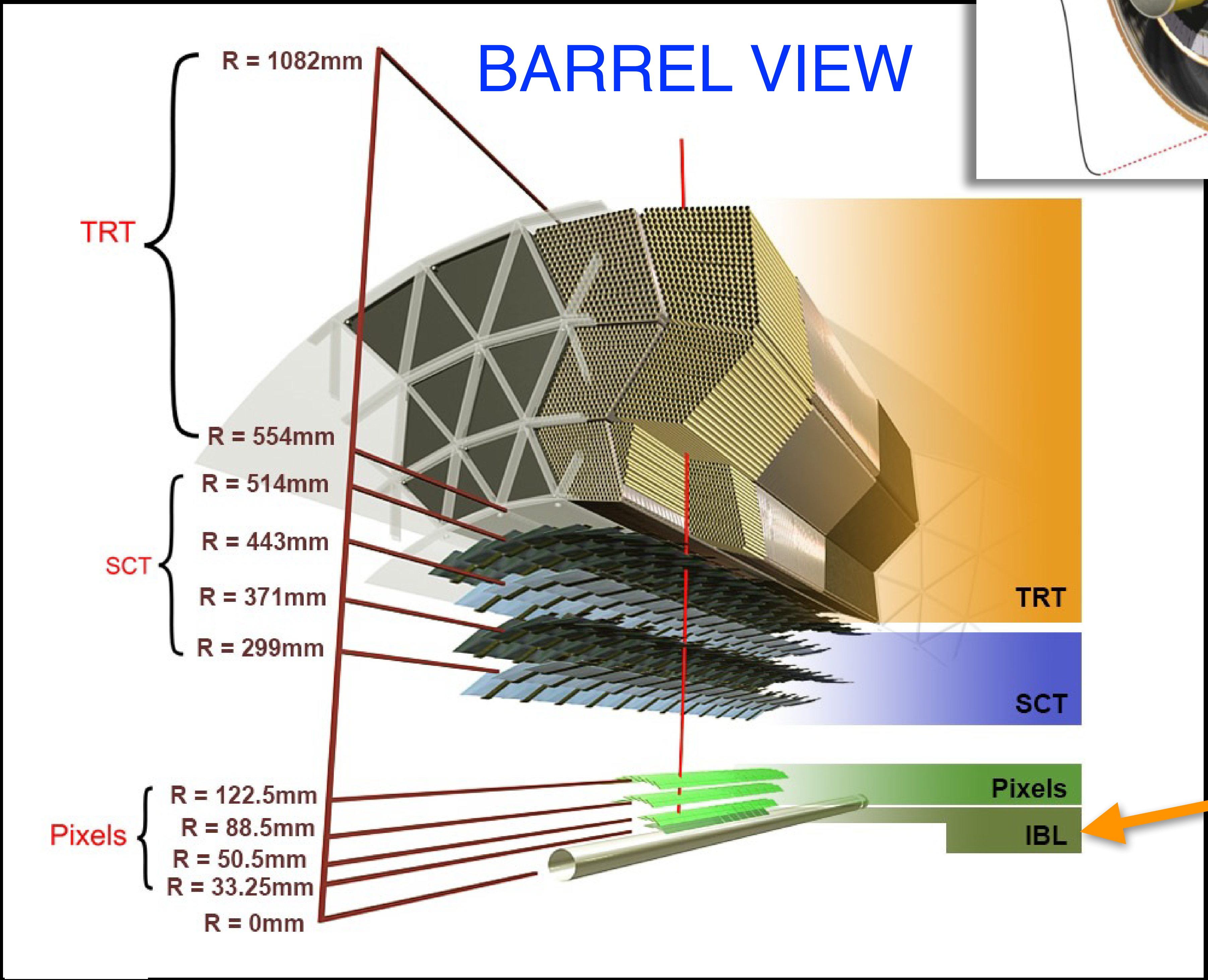
- detection technique for charged particles
- particles traversing tube ionises the gas
  - ➡ deposited charge drifts to anode wire in electric (E) field
    - charge amplification in high E-field in vicinity of wire leads to large signal pulse
  - ➡ measure time of signal pulse to determine drift circle
  - Lorentz angle deflection in B-field (not shown)
    - fast signal detection (v_D~30 ns/mm)
    - resolution of O(100 µm) on measured radius
- track reconstruction from drift circles
  - ➡ obtain drift radii from measured times
  - ➡ combined several measurements to find track
    - resolve left-right ambiguity (dotted line)
  - ➡ ATLAS TRT: as well electron identification using transition radiation

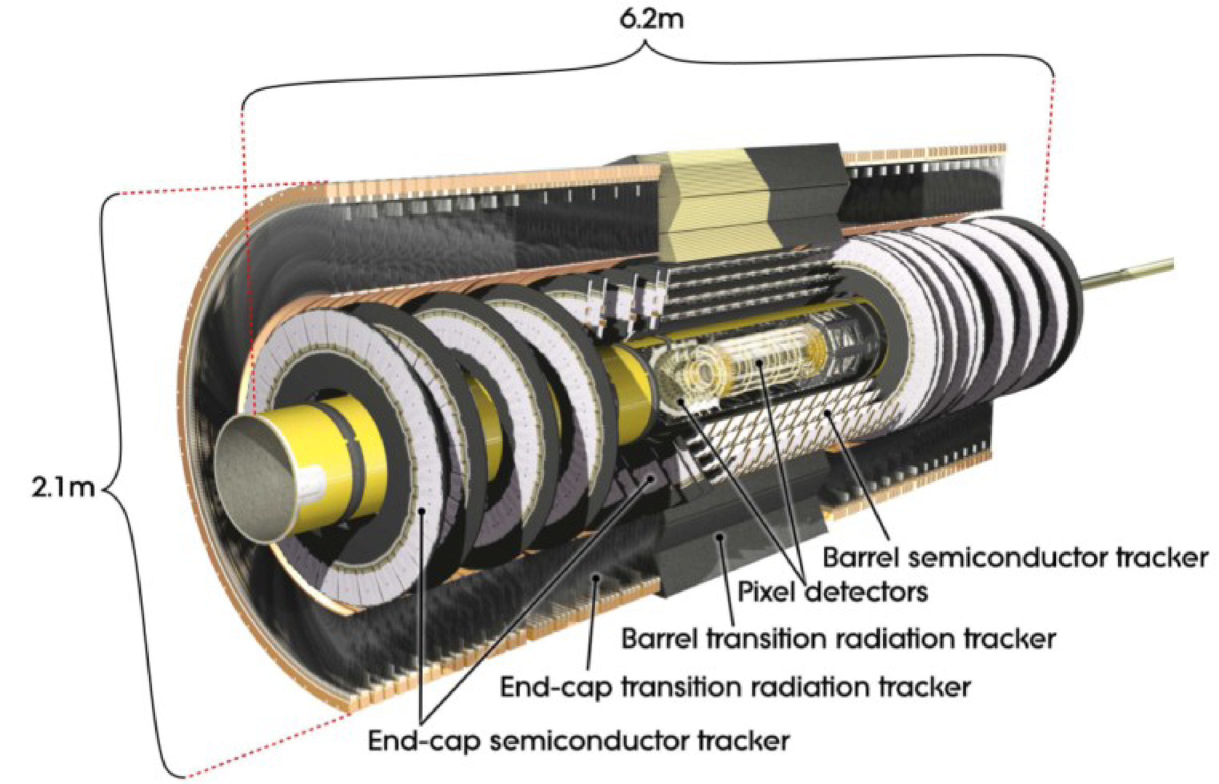


right side of ambiguity has large residual

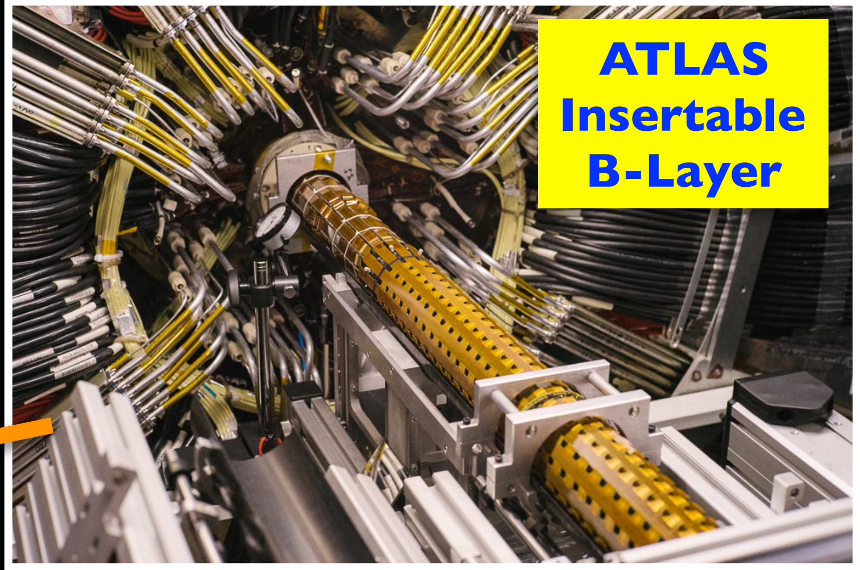
ATLAS Inner Detector

➡ combines semiconductor trackers and drift tubes



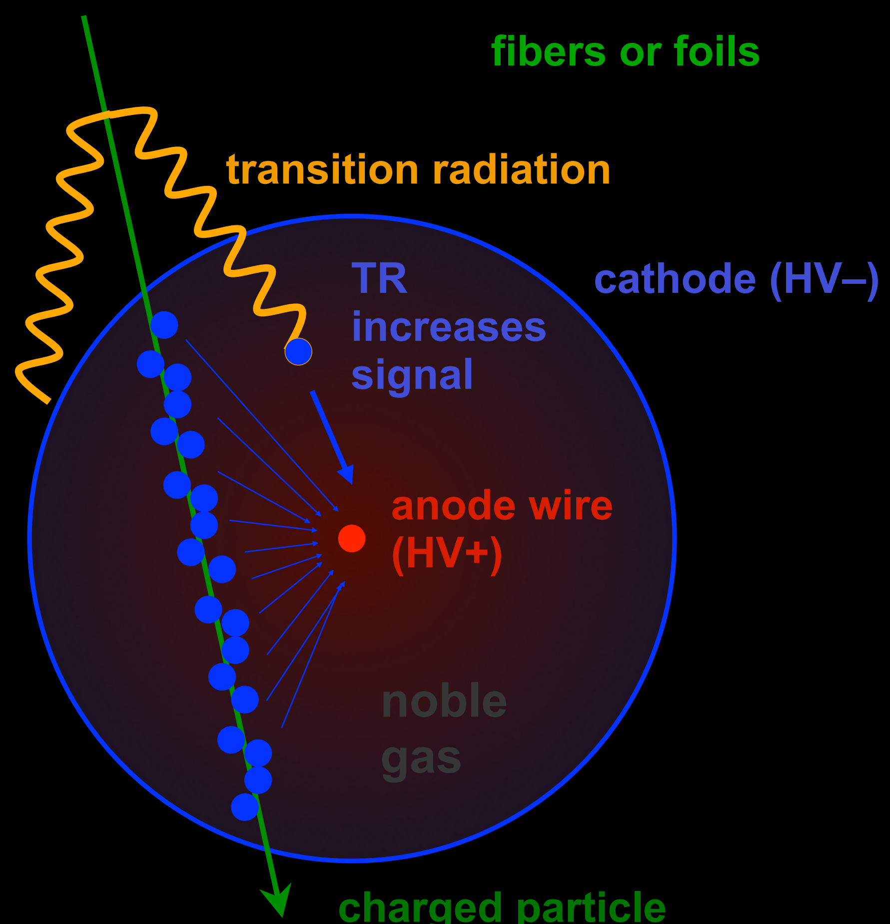


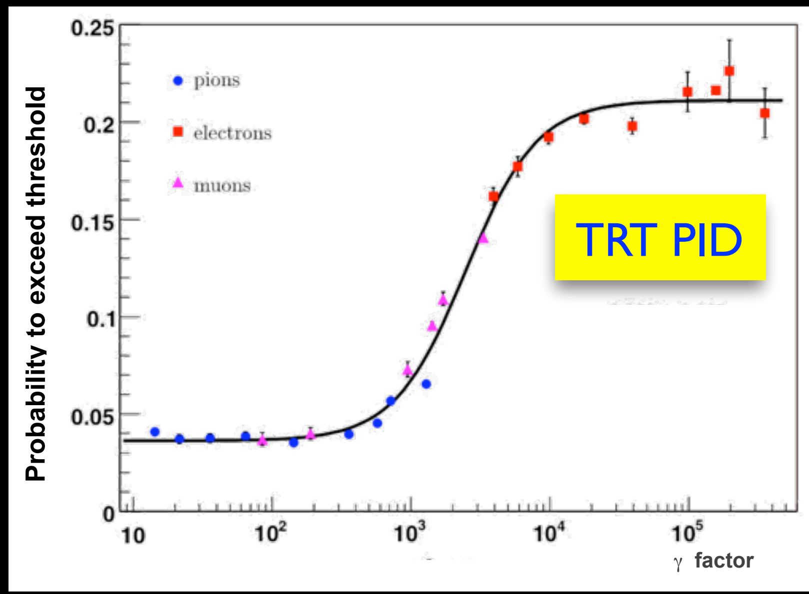
- barrel track passes:
  - ➡ 4 Pixel layers
  - ➡ 4x2 silicon Strips on stereo modules

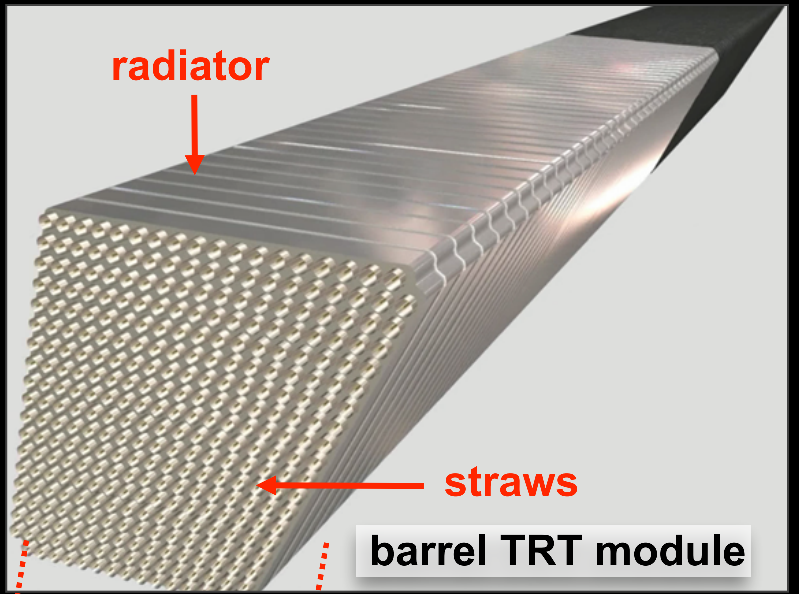


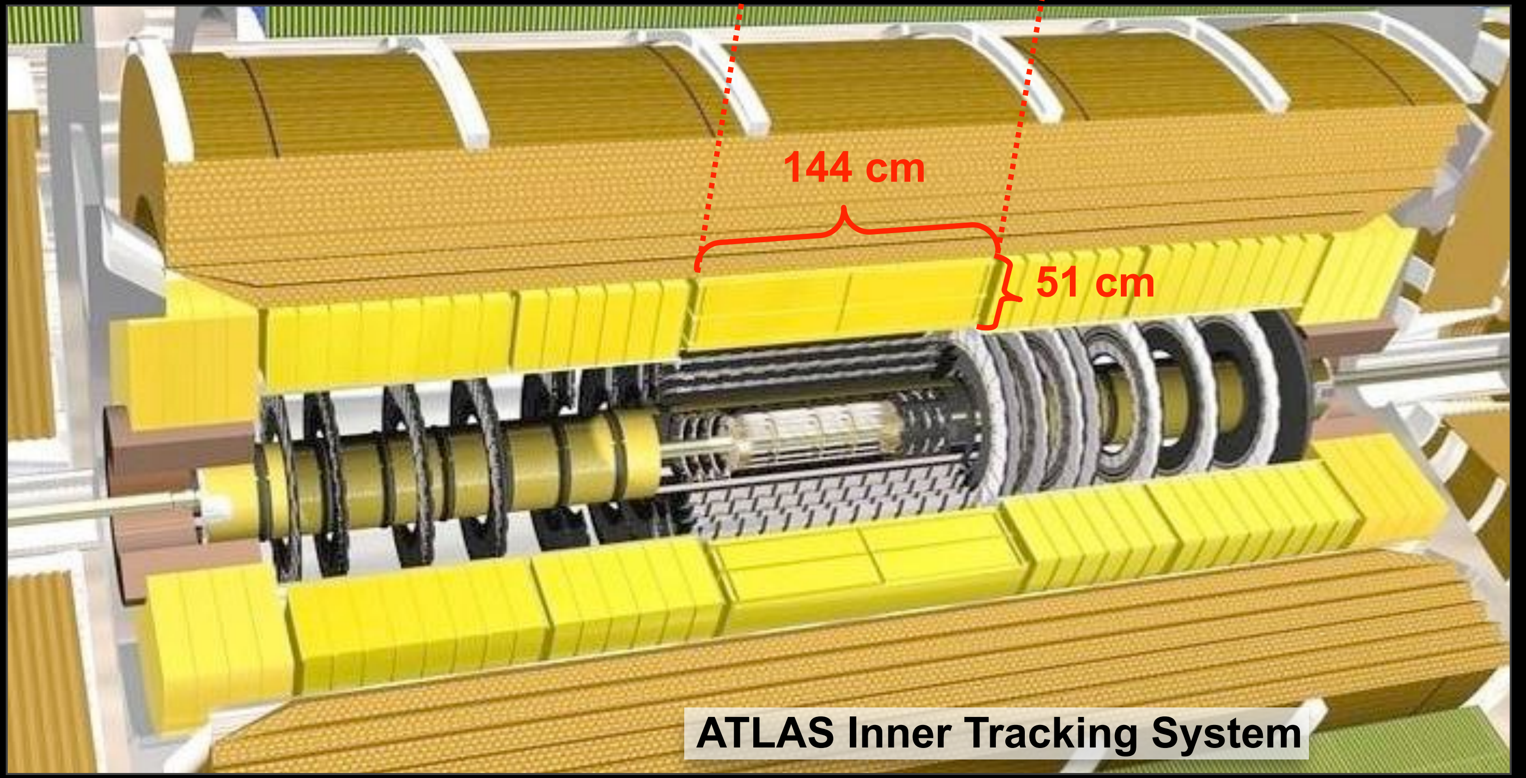
Electron Identification in the ATLAS

➡ e/π separation via transition radiation: polymer (PP) fibers/foils interleaved with drift tubes









➡ electrons radiate → higher signal

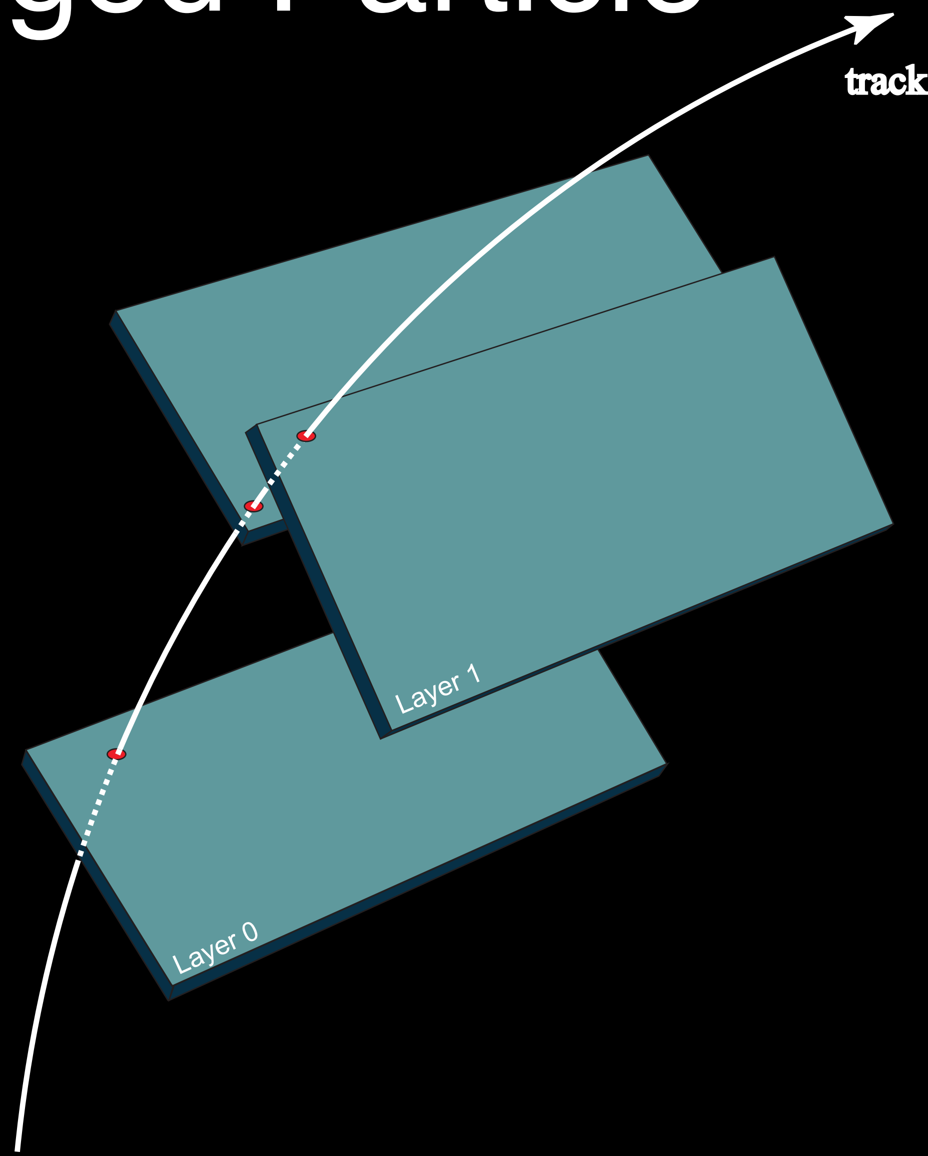
➡ PID info by counting high-threshold hits component

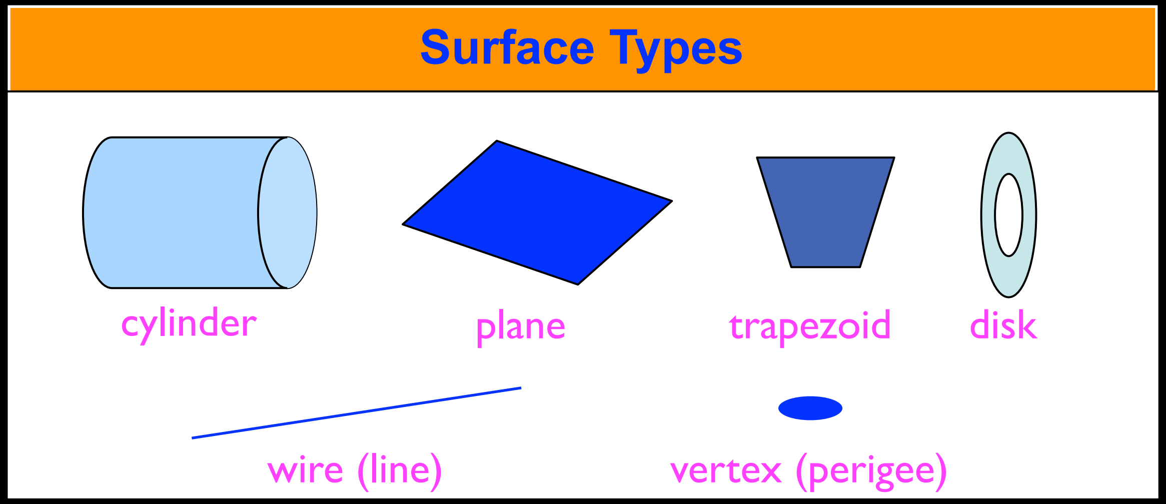
Trajectories and Extrapolation

A Trajectory of a Charged Particle

- ➡ in a solenoid B-field a charged particle trajectory is describing a helix
  - a circle in the plane perpendicular to the field (Rϕ)
  - a path (not a line) at constant polar angle (θ) in the Rz plane
- ➡ a trajectory in space is defined by 5 parameters
  - the local position (l_1,l_2) on a plane, a cylinder, ..., on the surface defining
    a reference system
  - the direction in θ and ϕ plus the curvature Q/P_T
- ➡ ATLAS choice:

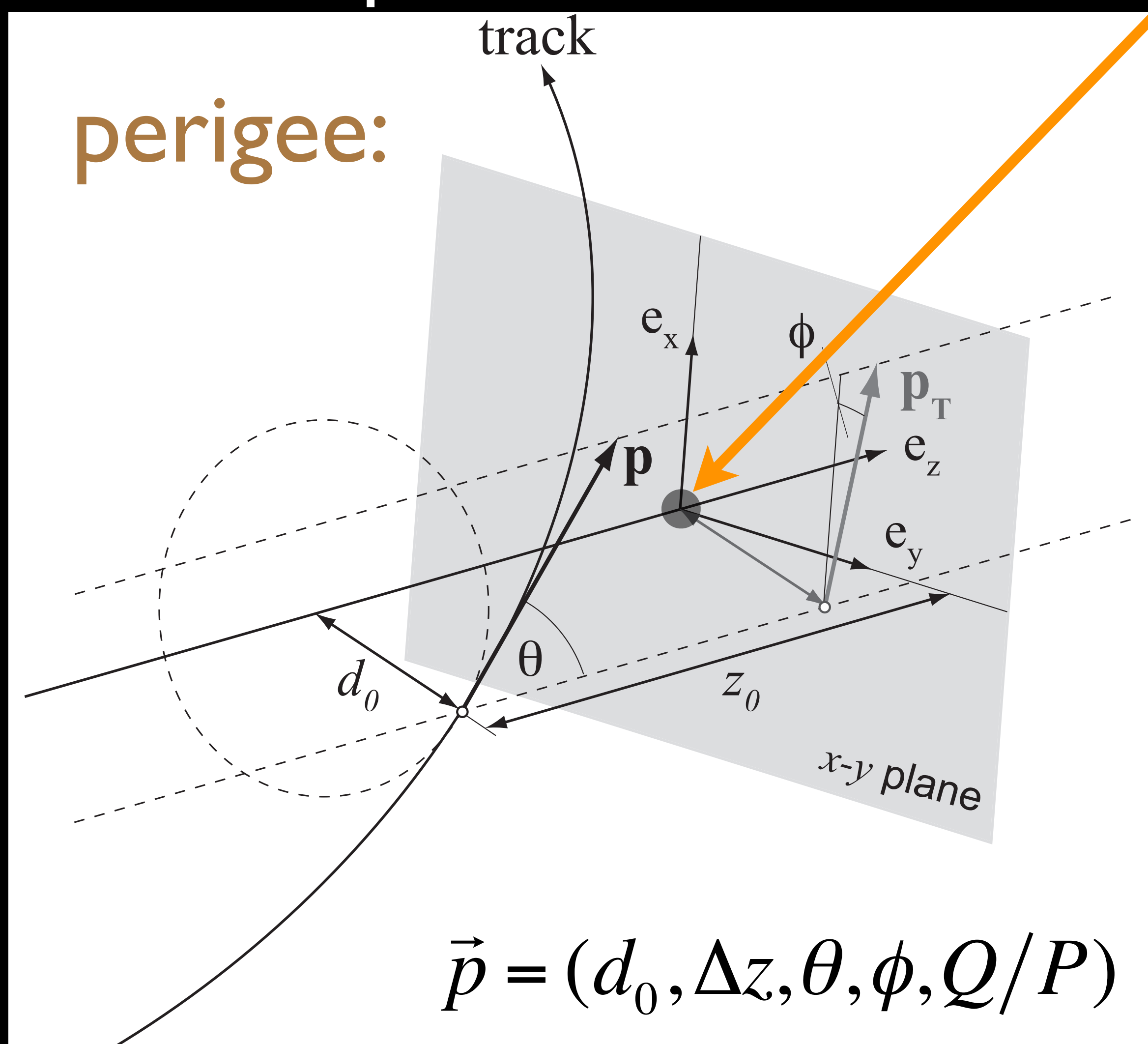
$$\vec{p} = (l_1, l_2, \theta, \phi, Q/P)$$

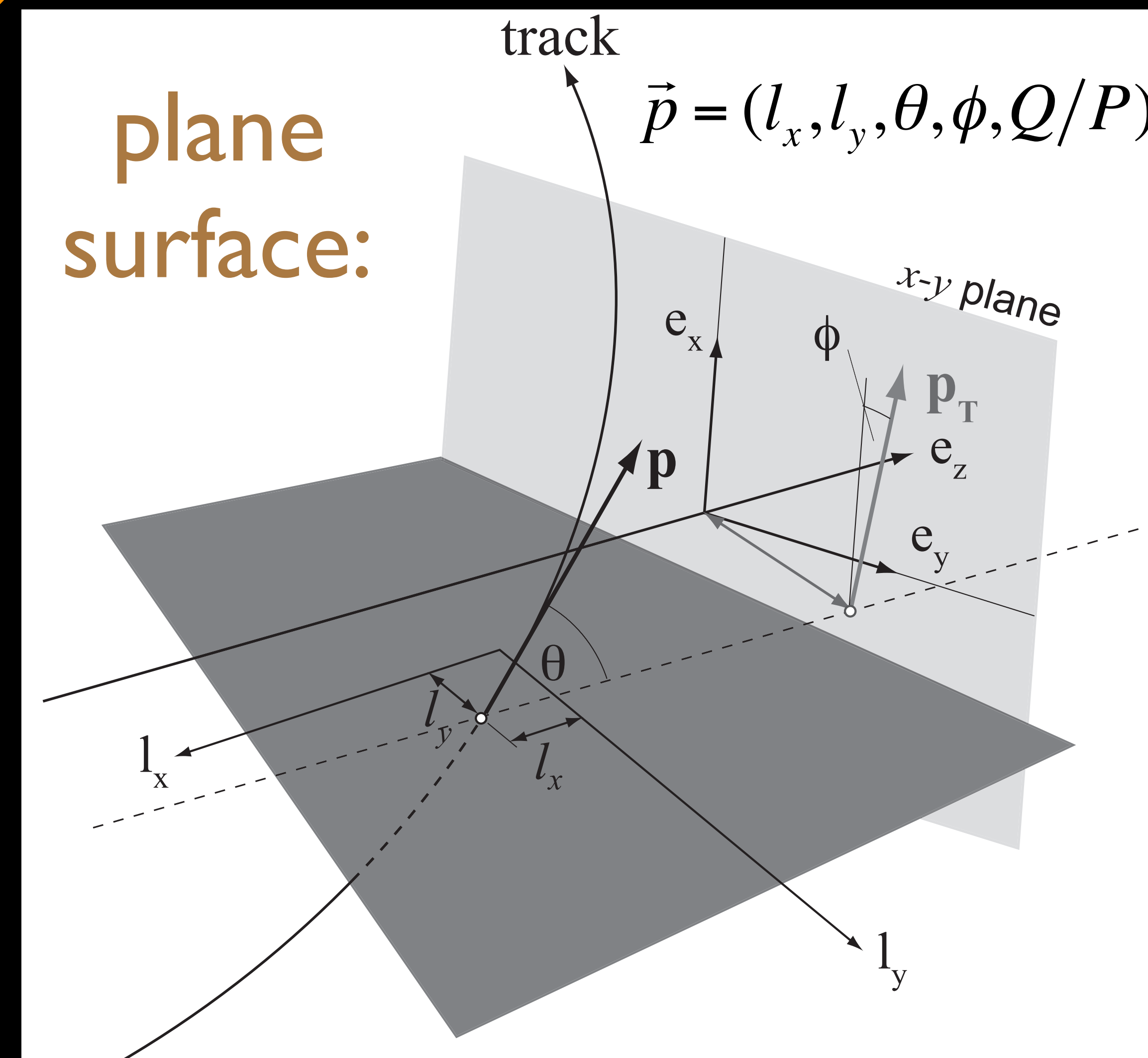




The Perigee Parameterisation

- helix representation w.r.t. a vertex

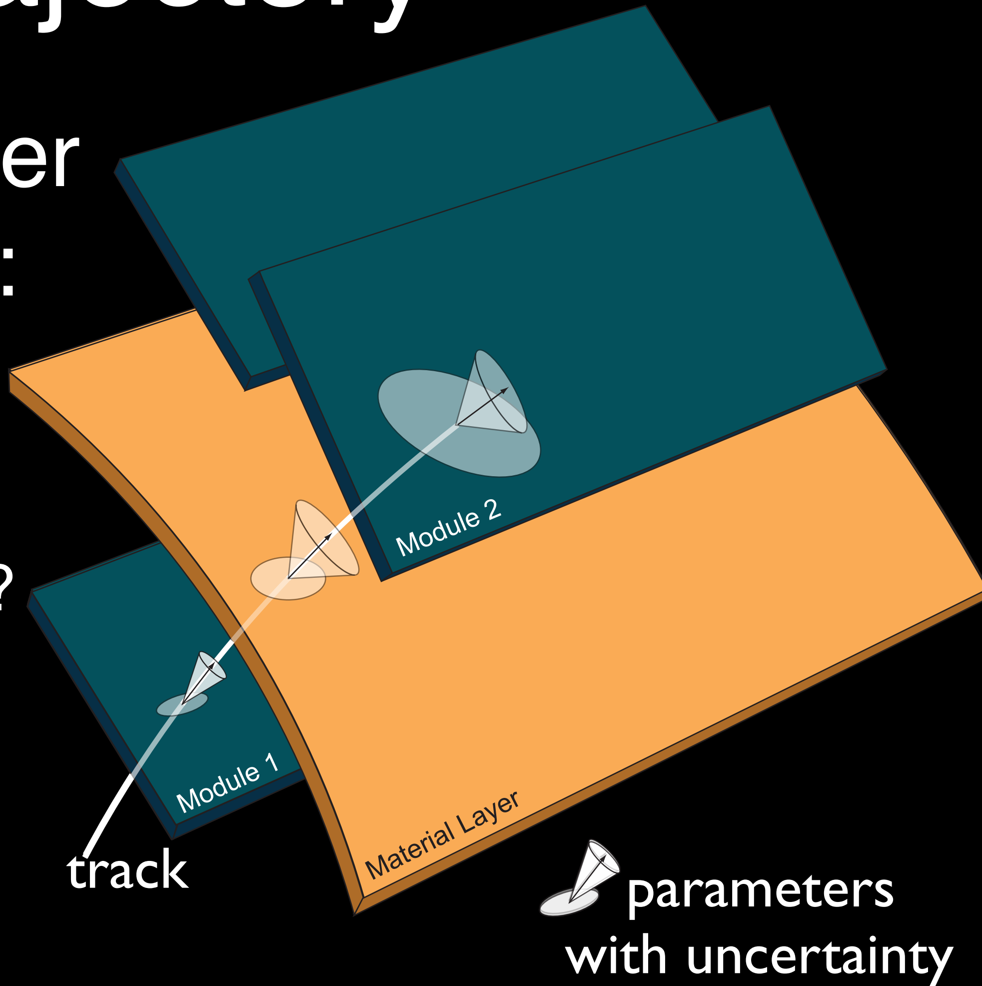




- commonly used
  - ➡ e.g. to express track parameters near the production vertex
  - ➡ alternative: e.g. on plane surface

Following the Particle Trajectory

- basic problems to be solved in order to follow a track through a detector:
  - ➡ next detector module that it intersects ?
  - ➡ what are its parameters on this surface ?
    - what is the uncertainty of those parameters ?
  - ➡ for how much material do I have to correct for ?



- requires:
  - ➡ a detector geometry
    - surfaces for active detectors
    - passive material layers
  - ➡ a method to discover which is the next surface (navigation)
  - ➡ a propagator to calculate the new parameters and its errors
    - often referred to as "track model"

- for a constant B-field (or no field)
  - ➡ an analytical formula can be calculated for an intersection of a helix (or a straight line) on simple surfaces (plane, cylinder, vertex,...)

Track Propagation in realistic B-Field

- for inhomogeneous B-field there is no analytical solution
  - ➡ start from equation of motion for a particle with charge q in magnetic field B:

$$\frac{d\vec{p}}{dt} = q\vec{v} \times \vec{B}$$

  - ➡ can be written as set of differential equations for motion along z with x(z) and y(z)

$$\frac{d^2x}{dz^2} = \frac{q}{p}R\left[\frac{dx}{dz}\frac{dy}{dz}B_x - \left(1+\left(\frac{dx}{dz}\right)^2\right)B_y + \frac{dy}{dz}B_z\right]$$

$$\frac{d^2y}{dz^2} = \frac{q}{p}R\left[\left(1+\left(\frac{dy}{dz}\right)^2\right)B_x - \frac{dx}{dz}\frac{dy}{dz}B_y - \frac{dx}{dz}B_z\right]$$

with:

$$R = \frac{ds}{dz} = \sqrt{1+\left(\frac{dx}{dz}\right)^2+\left(\frac{dy}{dz}\right)^2}$$

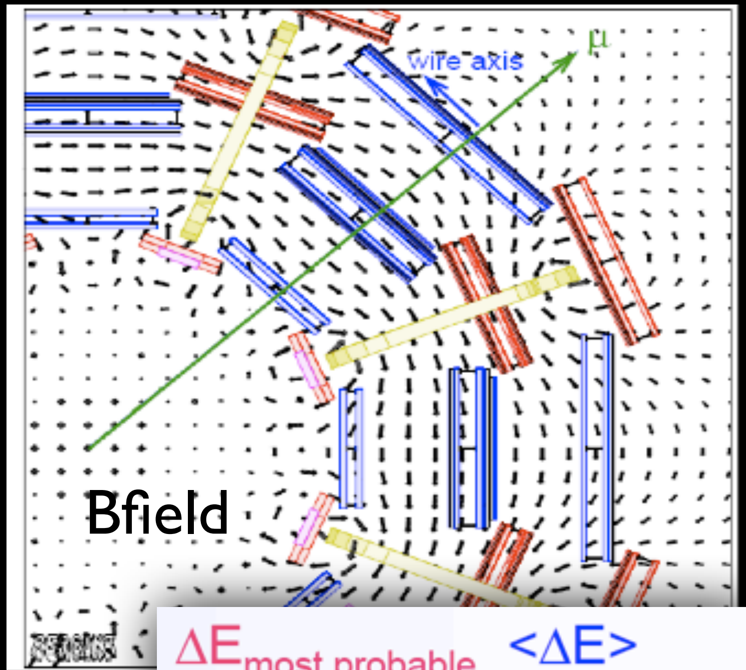
- no analytical solution for inhomogeneous B-field, requires numerical integration along the path of the trajectory

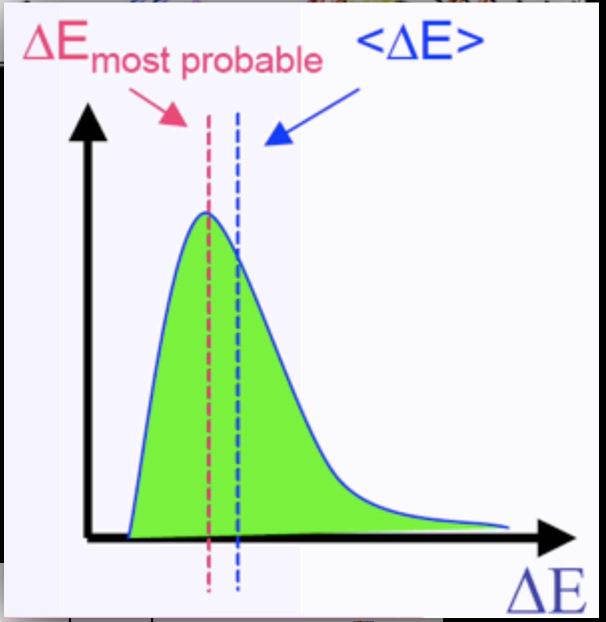
➡ numerical integration done using Runge-Kutta technique

- in ATLAS a 4th order adaptive Runge-Kutta-Nystrom approach is used, propagates covariance matrix in parallel (Bugge, Myrheim, 1981, NIM 179, p.365)

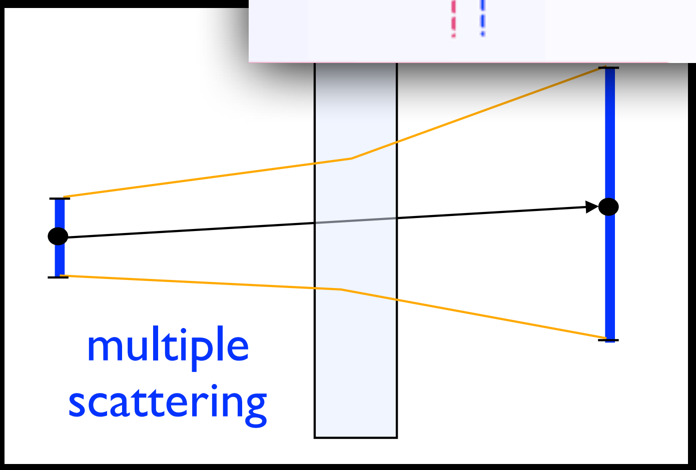
Track Propagation in realistic B-Field

- ATLAS Runge-Kutta propogator:
  - ➡ parameter propagation is 4th order
  - ➡ adaptive: use 3rd order result to monitor step precision and adapt step size (h)
  - ➡ monitor the remaining distance to the target surface, if a few µm, use Taylor approximation to reach surface
  - ➡ Nystrom technique: does as well numerical integration of Jacobian for error propagation (fast & precise)



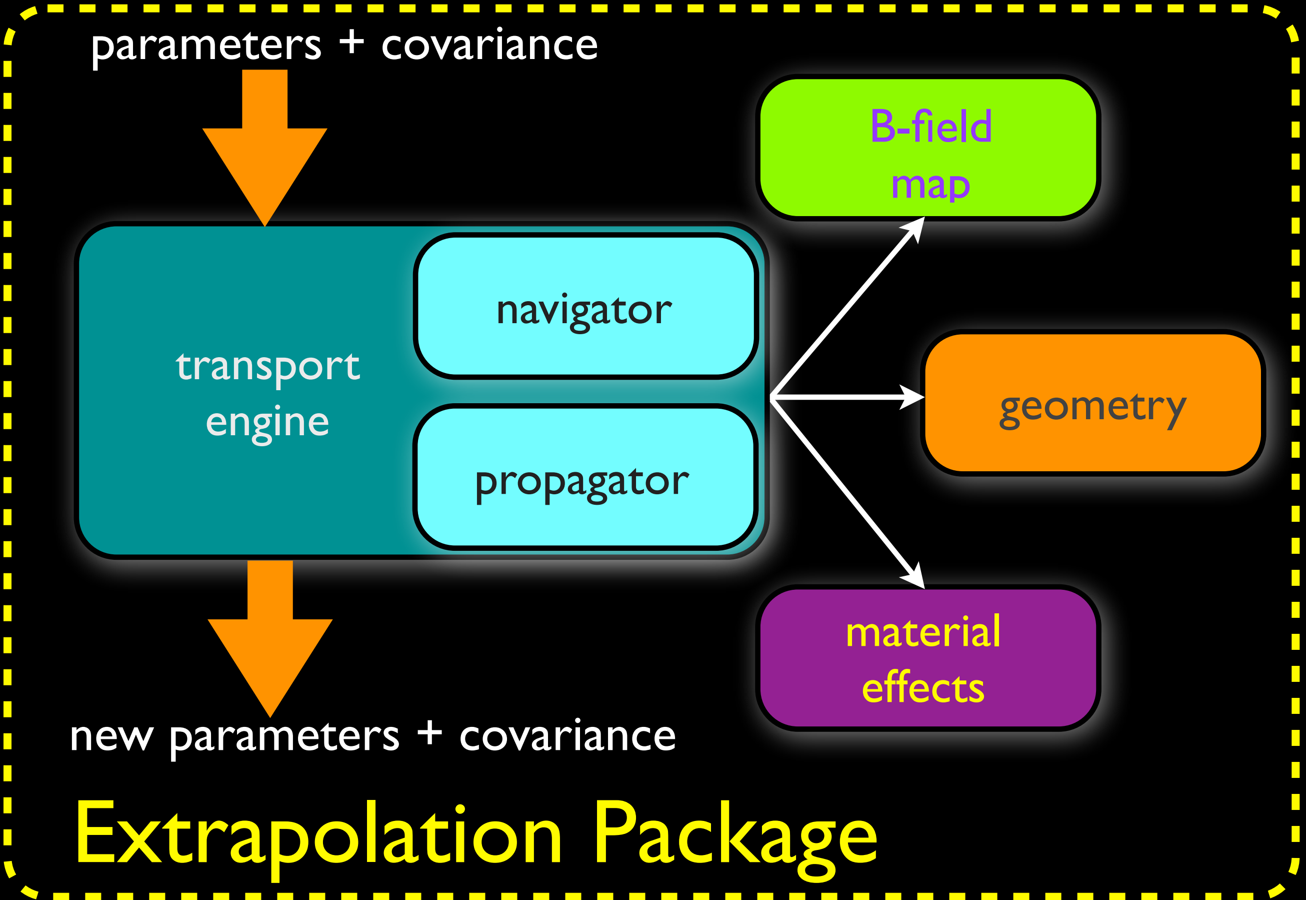


- need to allow for material effects
  - ➡ energy loss
    - use most probably energy loss for x/X_0
    - correct momentum (curvature) and its covariance
  - ➡ multiple scattering
    - increases uncertainty on direction of track
    - for given x/X_0 traversed add term to covariances of θ and ϕ on a material "layer"



The Track Extrapolation Package

- a transport engine used in tracking software
  - ➡ central tool for pattern recognition, track fitting, etc.
  - ➡ parameter transport from surface to surface, including covariance
  - ➡ encapsulates the track model, geometry and material corrections



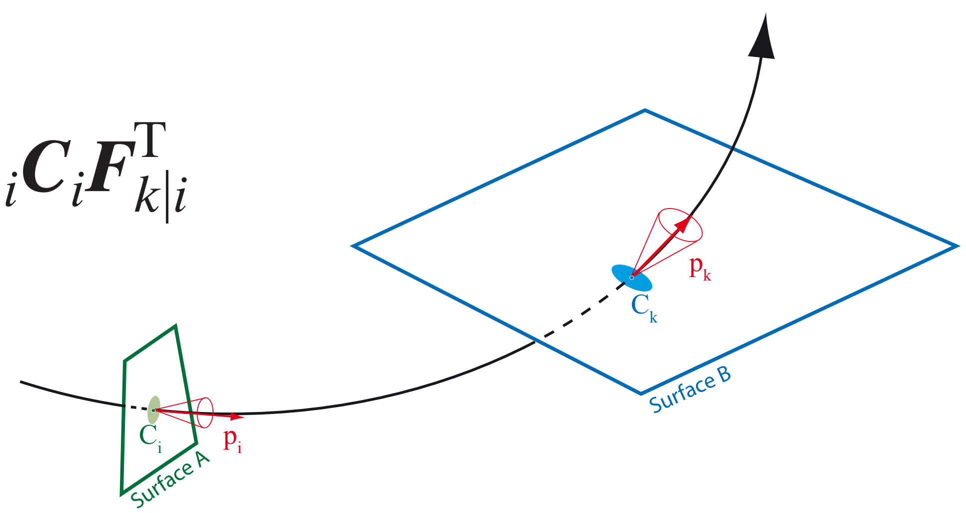
track following in mathematical terms:

$$q_k = f_{k|i}(q_i)$$

convariance: $C_k = F_{k|i} C_i F_{k|i}^T$

with: $f_{k|i}$ ~ track model

$$F_{k|i} = \frac{\partial q_k}{\partial q_i}$$ ~ Jacobi matrix



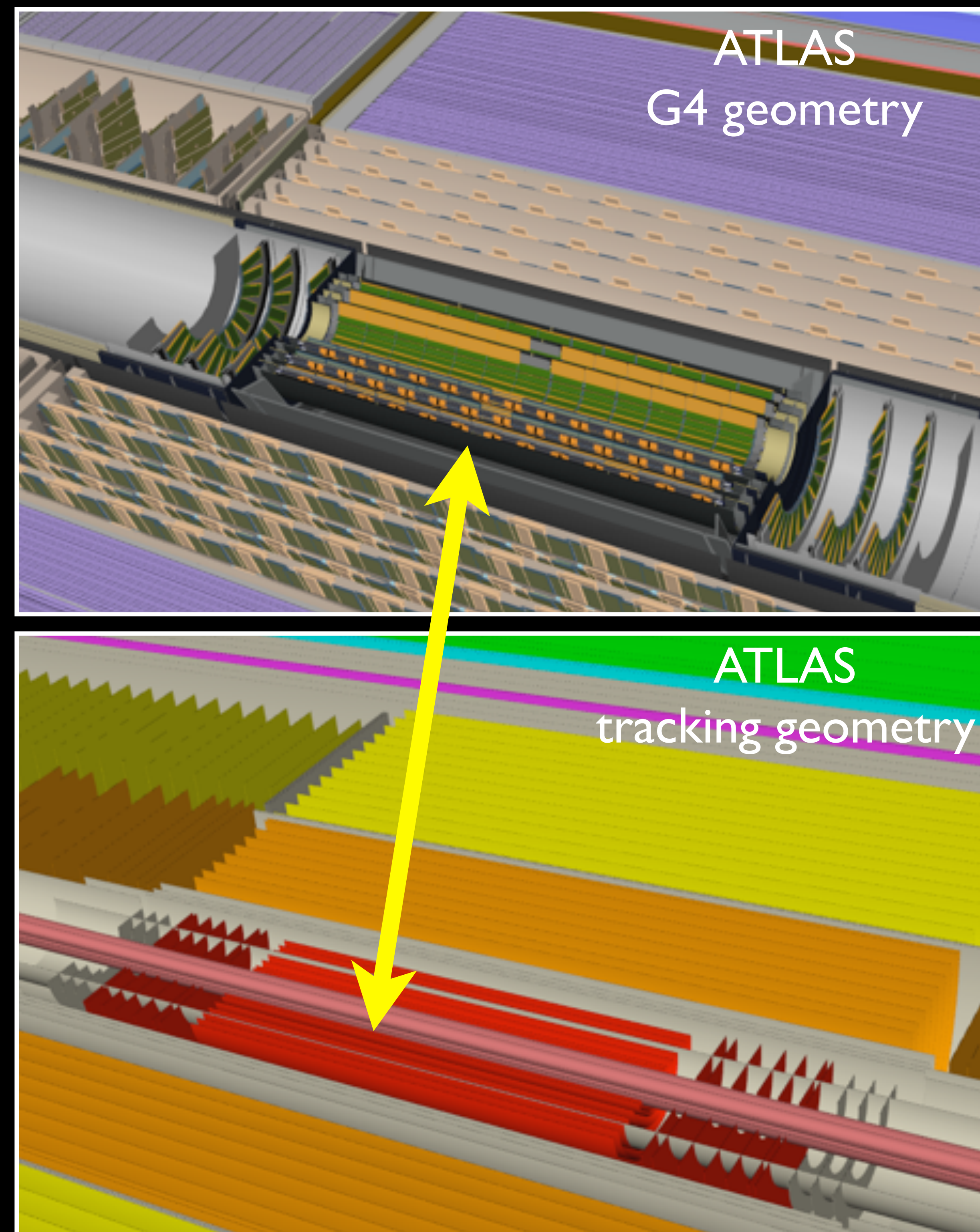
Full and Fast (Tracking) Geometries

- complex G4 geometries not optimal for reconstruction
  - ➡ simplified tracking geometries
  - ➡ material surfaces, field volumes

- reduced number of volumes for tracking
  - ➡ blending details of material onto simple surfaces/volumes
  - ➡ surfaces with 2D material density maps, templates per Si sensor...

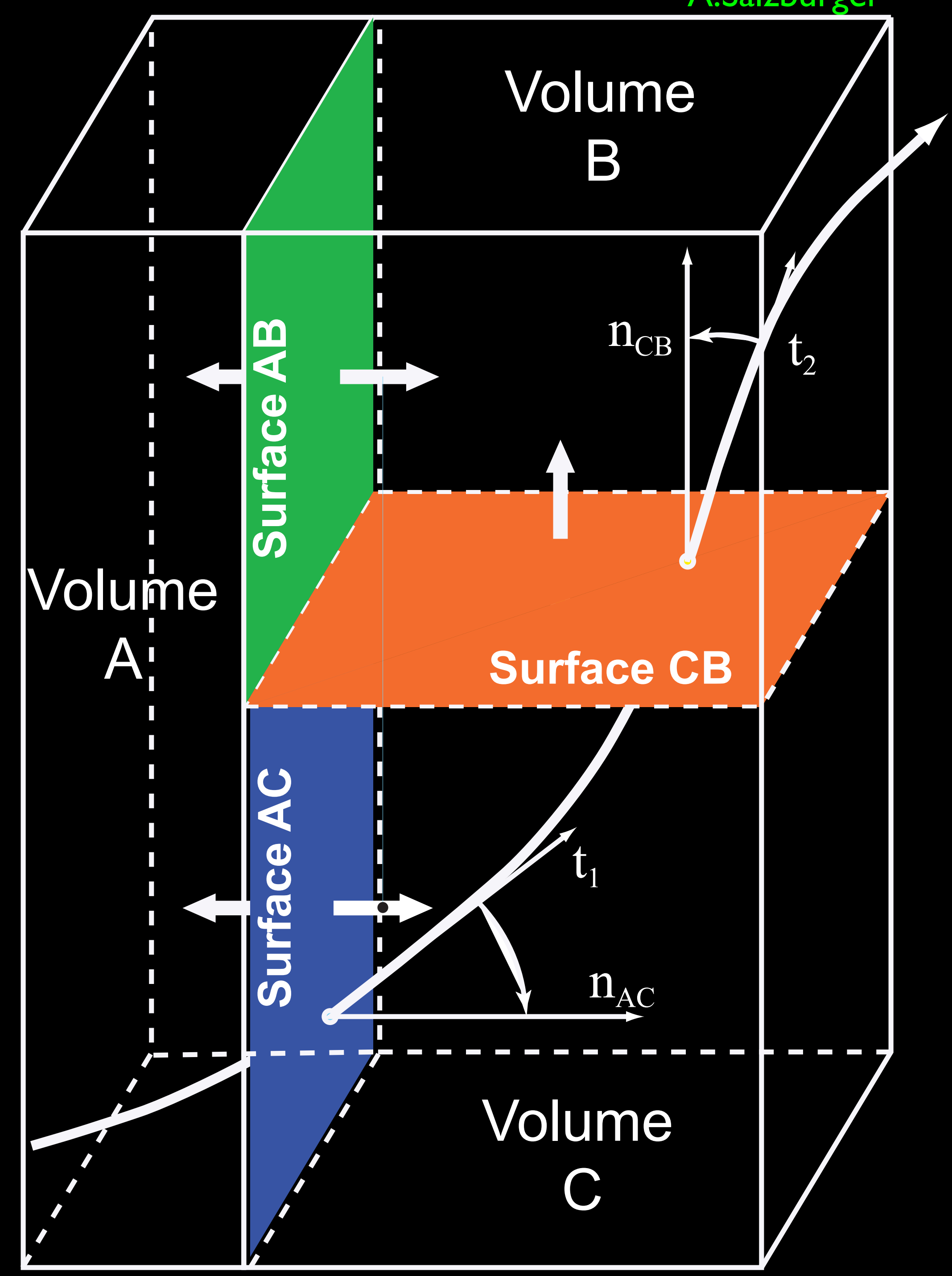
	G4	tracking
ALICE	4.3 M	same *1
ATLAS	4.8 M	10.2K *2
CMS	2.7 M	3.8K *2
LHCb	18.5 M	30

*1 ALICE uses full geometry (TGeo)
*2 plus a surface per Si sensor



Embedded Navigation Schemes

- **embedded navigation** scheme in tracking geometries
  - ➡ G4 navigation uses voxelisation as generic navigation mechanism
  - ➡ embedded navigation for simplified models
    - used in pattern recognition, extrapolation, track fitting and fast simulation



- **example**: ATLAS
  - ➡ developed geometry of connected volumes
  - ➡ boundary surfaces connect neighbouring volumes to predict next step

ATLAS	G4	tracking	ratio
crossed volumes in tracker	474	95	5
time in SI2K sec	19.1	2.3	8.4

(neutral geantinos, no field lookups)

Track Fitting

From Measurement Model to Track

- measurements m_k of a track
  - ➡ in mathematical terms a model:

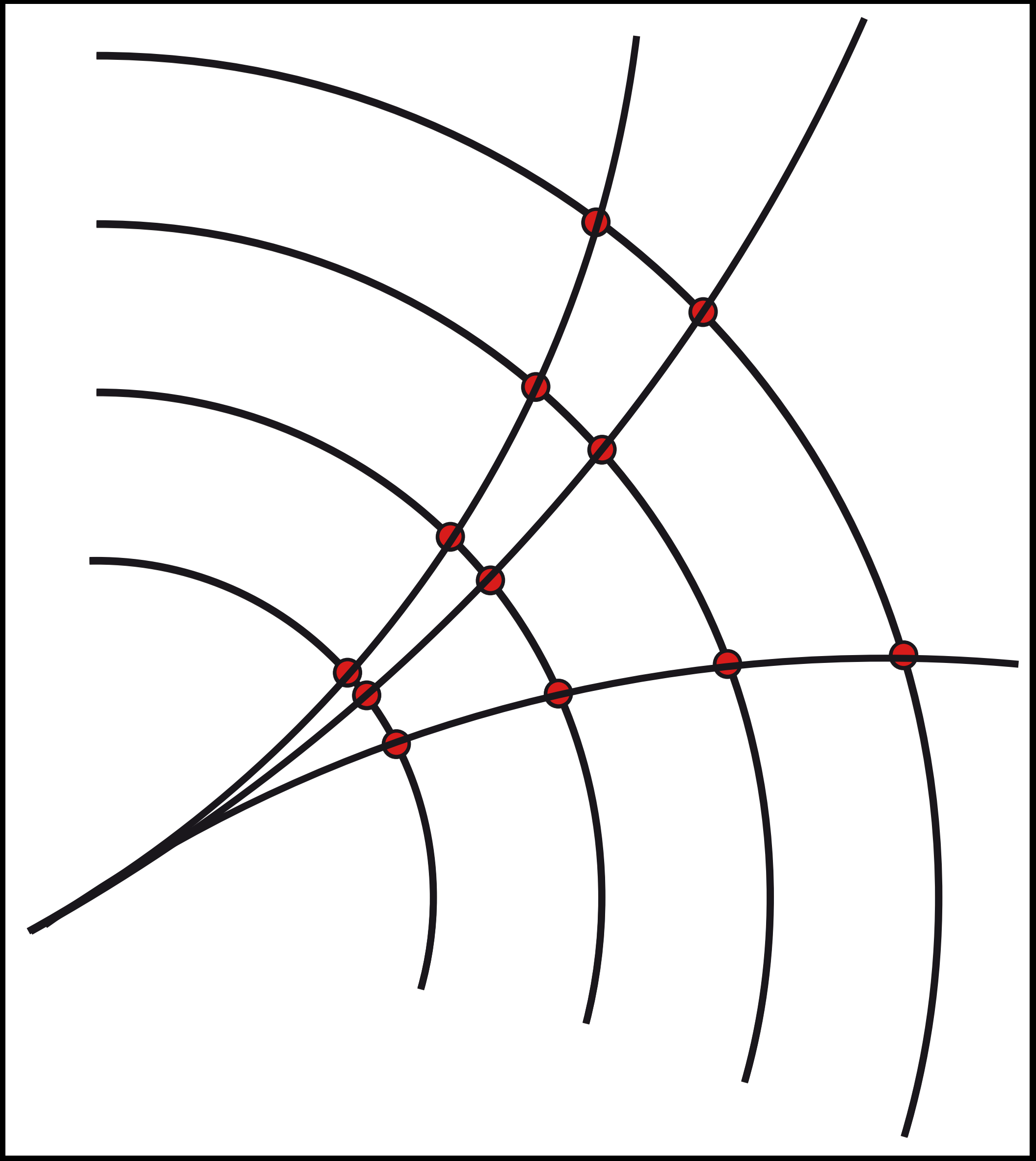
$$m_k = h_k(q_k) + \gamma_k$$

with: h_k ~ functional dependency of measurement on e.g. track angle

γ_k ~ error (noise term)

$$H_k = \frac{\partial m_k}{\partial q_k}$$ ~ Jacobian, often contains only rotations and projections

  - ➡ in practice those m_k are clusters, drift circles



- task of a track fit
  - ➡ estimate the track parameters and their uncertainties from a set of measurements
- examples for fitting techniques
  - ➡ Least Square track fit or Kalman Filter track fit
  - ➡ more specialised versions: Gaussian Sum Filter or Deterministic Annealing Filters

Classical Least Square Track Fit

- construct and minimise the χ^2 function:

Carl Friedrich Gauss is credited with developing the fundamentals of the basis for least-squares analysis in 1795 at the age of eighteen. **Legendre** was the first to publish the method, however.

➡ Write down Least Square function:

$$\chi^2 = \sum_k \Delta m_k^T G_K^{-1} \Delta m_k \quad \text{with:} \quad \Delta m_k = m_k - d_k(p)$$

d_k contains measurement model and propagation of the parameters p : $\boldsymbol{d}_k = \boldsymbol{h}_k \circ \boldsymbol{f}_{k|k-1} \circ \cdots \circ \boldsymbol{f}_{2|1} \circ \boldsymbol{f}_{1|0}$

G_k is the covariance matrix of m_k .

➡ Linearise the χ^2 with a Taylor expansion:

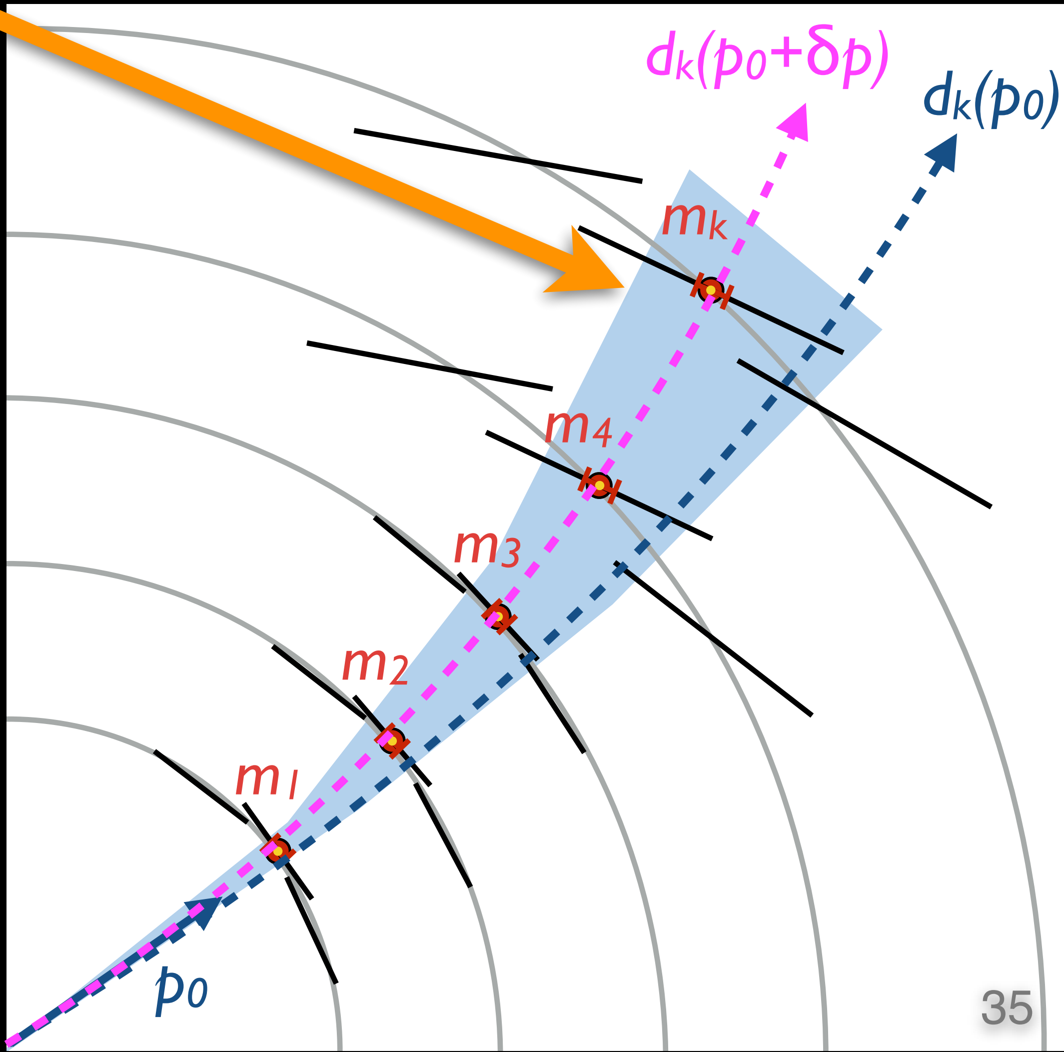
$$d_k(p_0 + \delta p) \cong d_k(p_0) + D_k \cdot \delta p \text{ + higher terms}$$

with Jacobian: $\boldsymbol{D}_k = \boldsymbol{H}_k \boldsymbol{F}_{k|k-1} \cdots \boldsymbol{F}_{2|1} \boldsymbol{F}_{1|0}$

➡ Minimising linearised χ^2 yields system of linear equations:

$$\frac{\partial \chi^2}{\partial p} = 0 \Rightarrow \delta p = \left(\sum_k D_k^T G_k^{-1} D_k \right)^{-1} \sum_k D_k^T G_k^{-1} (m_k - d_k(p_0))$$

and covariance of δp is: $C = \left(\sum_k D_k^T G_k^{-1} D_k \right)^{-1}$



Classical Least Square Track Fit

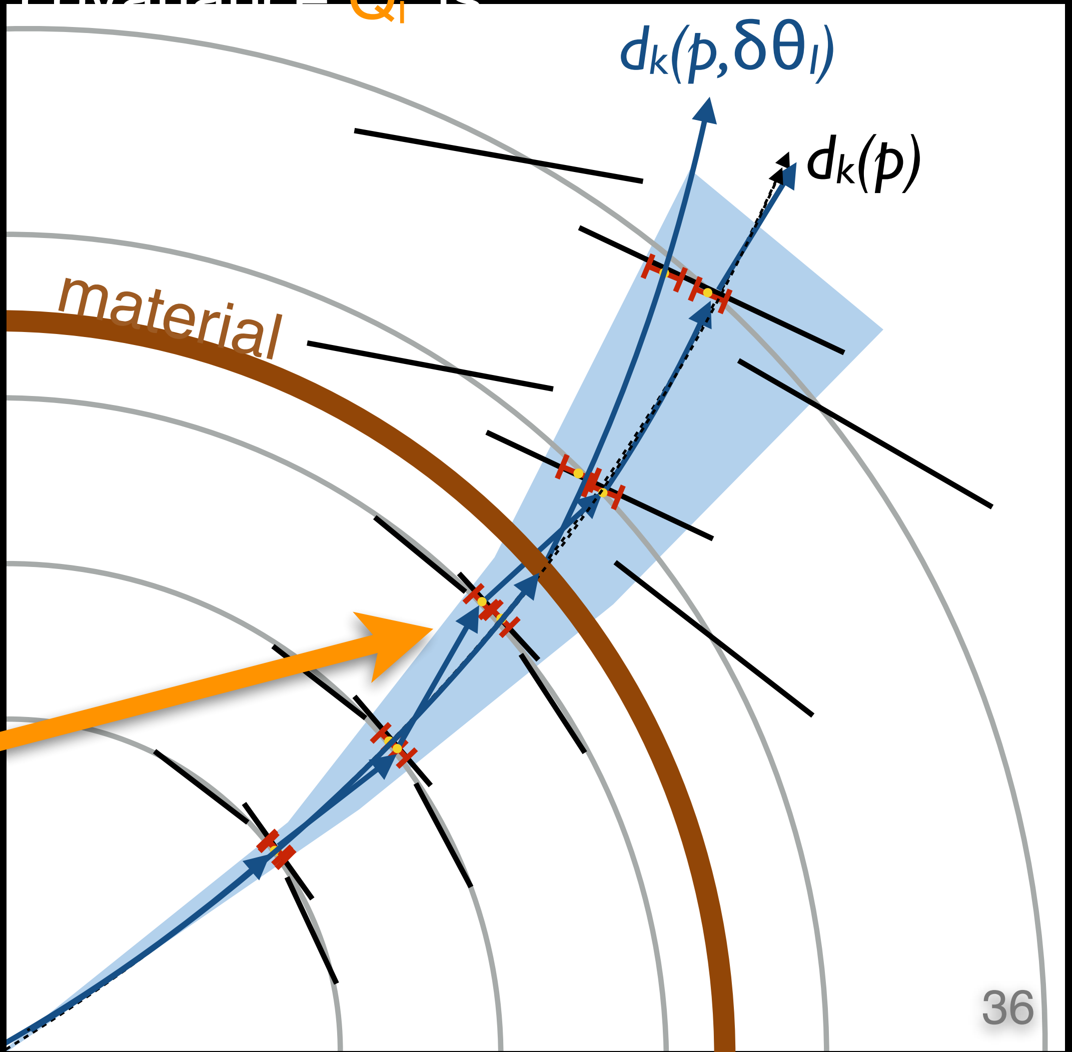
similar to Broken Lines fit

- allowing for material effects in fit:
  - ➡ can be absorbed in track model f_{kli} , provided effects are small
  - ➡ for substantial multiple scatting, allows for scattering angles in the fit
- introduce scattering angles on material surfaces
  - ➡ on each material surface, add 2 angles $\delta\theta_i$ as fee parameters to the fit
  - ➡ expected mean of those angles is 0 (!), their covariance Q_i is given by multiple scattering in x/X_0
- results in additional term in χ^2 equations:

$$\chi^2 = \sum_k \Delta m_k^T G_K^{-1} \Delta m_k + \sum_i \delta\theta_i^T Q_i^{-1} \delta\theta_i$$

with: $\Delta m_k = m_k - d_k\left(p, \delta\theta_i\right)$

  - ➡ computationally expensive (invert a dimension 5+2*n matrix)
  - ➡ advantage is that the fitted track follows precisely the particle trajectory (e.g. for ATLAS muon reconstruction)

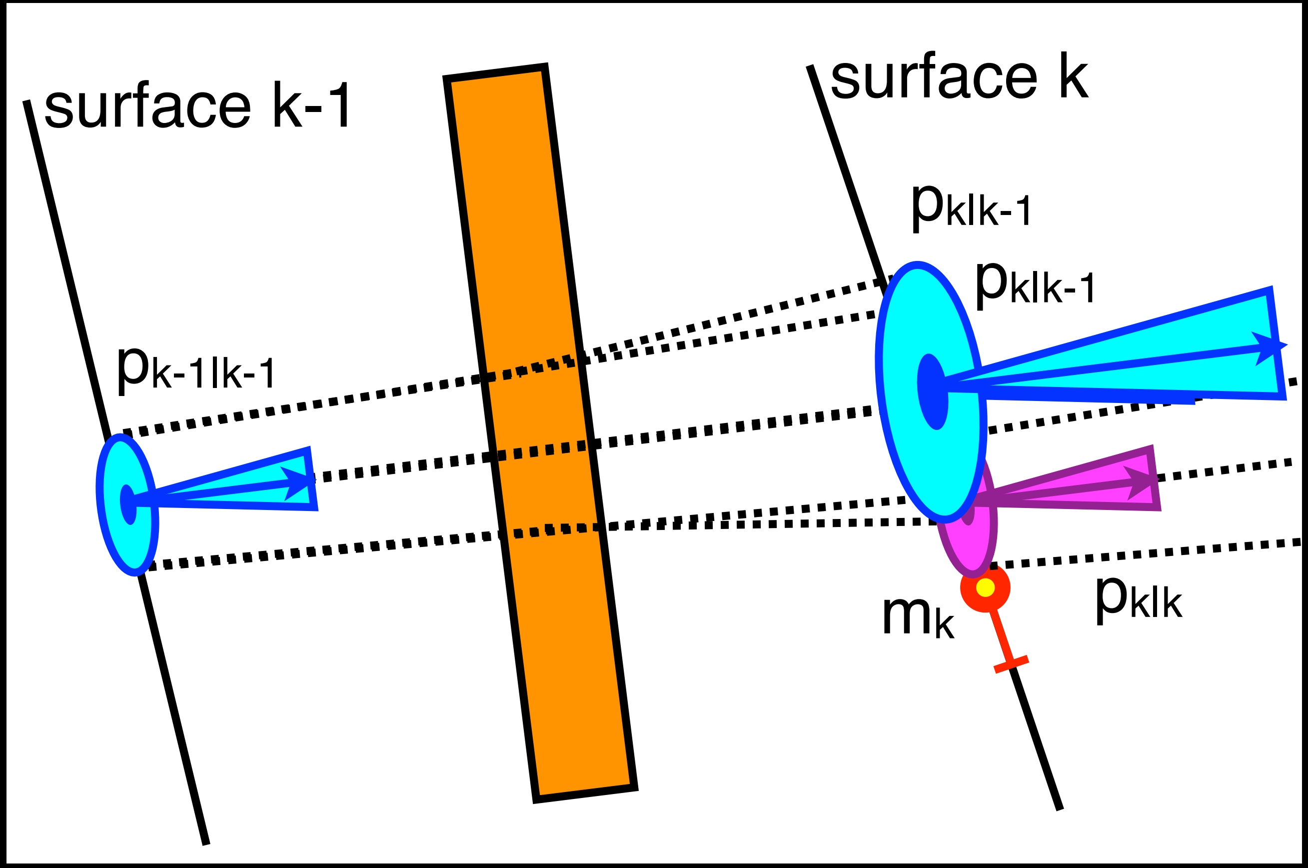


The Kalman Filter Track Fit

- a Kalman Filter is a progressive way of performing a least square fit
  - ➡ can be shown that it is mathematically equivalent

- how does the filter work ?
  - ➡ estimate starting parameters $p_{0|0}$
  - ➡ iterate over all hits 1..K:

1. take trajectory parameters $p_{k-1|k-1}$ at point k-1
2. propagate to point k to get predicted parameters $p_{k|k-1}$
3. update predicted parameters with measurement m_k to obtain $p_{k|k}$ (simple weighted mean or gain matrix update)
4. and start over with 1.



- material effects (multiple scattering and energy loss)
  - ➡ incorporated in the propagated parameters $p_{k|k-1}$ (extrapolated prediction)
  - ➡ and therefore enters automatically in the updated parameters $p_{k|k}$ at point k

The Kalman Filter Track Fit

- **forward filter**
  - ➡ in mathematical terms:

1. propagate p_{k-1} and its covariance C_{k-1} :

$$q_{k|k-1} = f_{k|k-1}(q_{k-1|k-1})$$
$$C_{k|k-1} = F_{k|k-1} C_{k-1|k-1} F^{T}_{k|k-1} + Q_k$$

with Q_k ~ noise term (M.S.)

2. update prediction to get $q_{k|k}$ and $C_{k|k}$:

$$q_{k|k} = q_{k|k-1} + K_k[m_k - h_k(q_{k|k-1})]$$
$$C_{k|k} = (I - K_k H_k) C_{k|k-1}$$

with K_k ~ gain matrix :

$$K_k = C_{k|k-1} H^{T}_{k} (G_k + H_k C_{k|k-1} H^{T}_{k})^{-1}$$

- ➡ precise fit result q_k at end of fit

- **Kalman Smoother:**
  - ➡ provides full information along track
  - ➡ equivalent: average forw./back. filter

- ➡ alternative to gain matrix approach is a weighted mean to obtian $p_{k|k}$
  - but requires to invert 5x5 matrix instead of a matrix of rank(G_k)

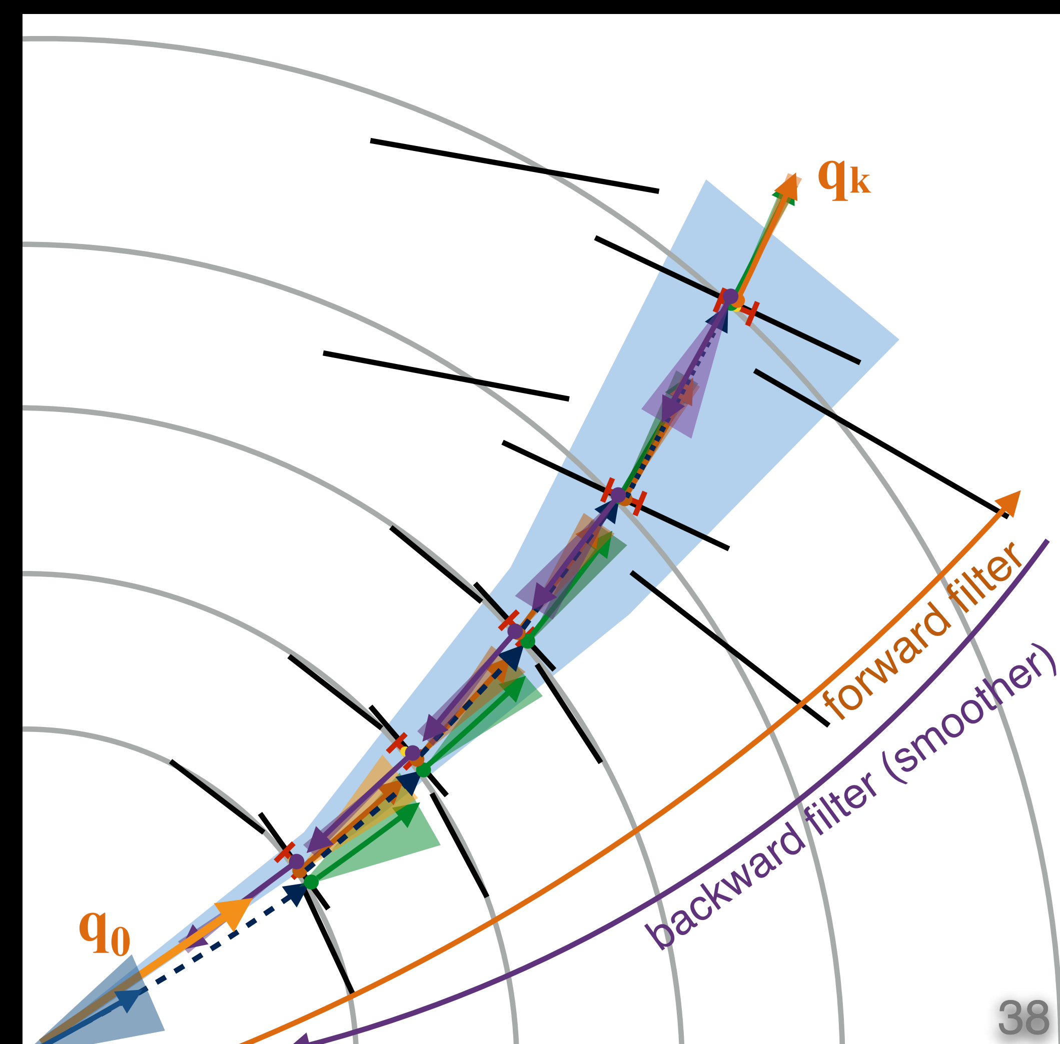
➡ Smoother in mathematical terms:

proceeds from layer $k+1$ to layer k :

$$q_{k|n} = q_{k|k} + A_k(q_{k+1|n} - q_{k+1|k})$$
$$C_{k|n} = C_{k|k} - A_k(C_{k+1|k} - C_{k+1|n})A^{T}_{k}$$

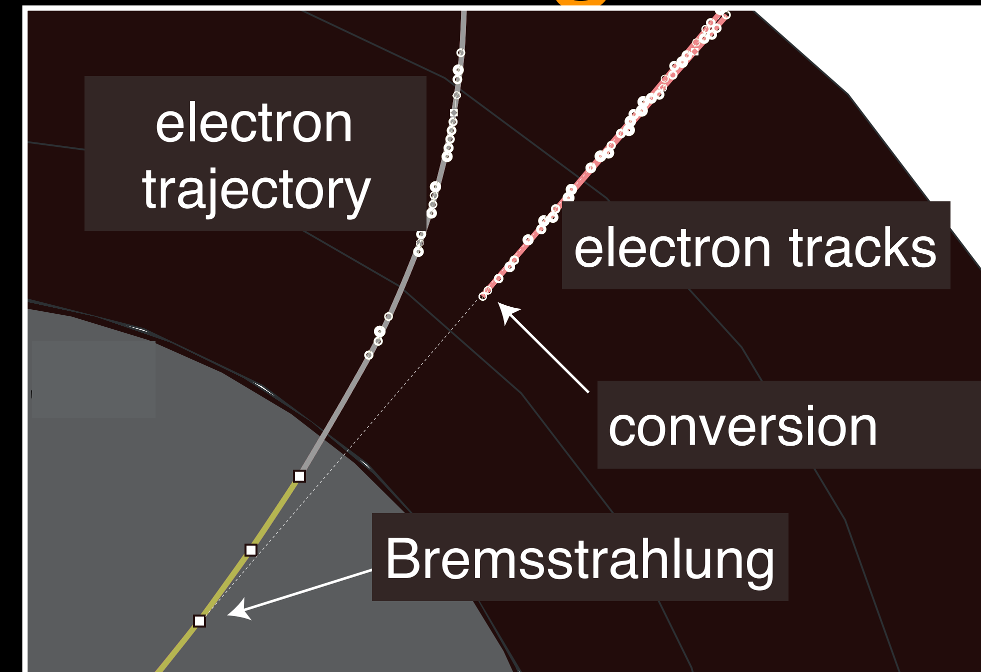
with A_k ~ smoother gain matrix :

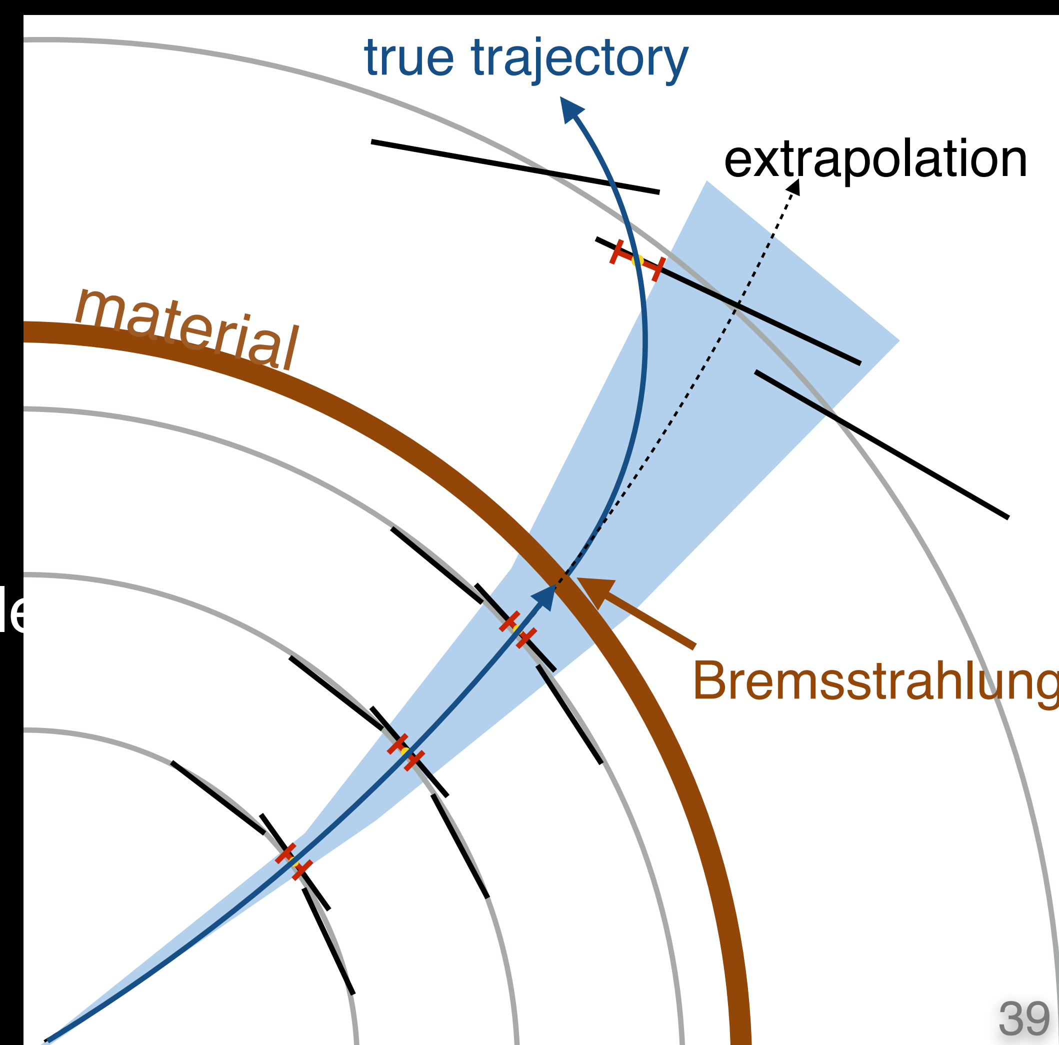
$$A_k = C_{k|k} F^{T}_{k+1|k} (C_{k+1|k})^{-1}$$



Fitting for Electron Bremsstrahlung

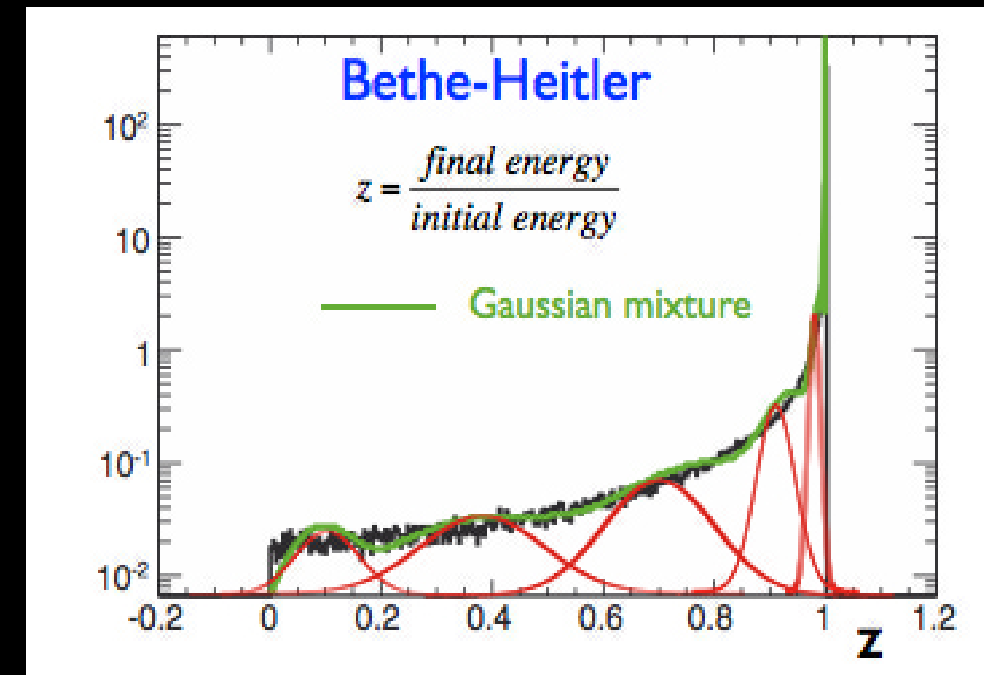
- material in tracker
  - ➡ e-Bremsstrahlung and γ-conversions
- electron efficiency limited
  - ➡ momentum loss due to Bremsstrahlung leads to sudden large changes in track curvature
  - ➡ loosing hits after Brem. leads to inefficiency
  - ➡ fit either biased towards small momenta or fails completely because of bad χ^2
- techniques to allow for Bremsstrahlung in track fitting
  - ➡ for Least Square track fit
    - allow Brem. effect to change curvature, additional term similar is to scattering angle
  - ➡ for Kalman Filter
    - increase correction for material effects in propagation to allow for Brem.
  - ➡ better: Gaussian Sum Filter

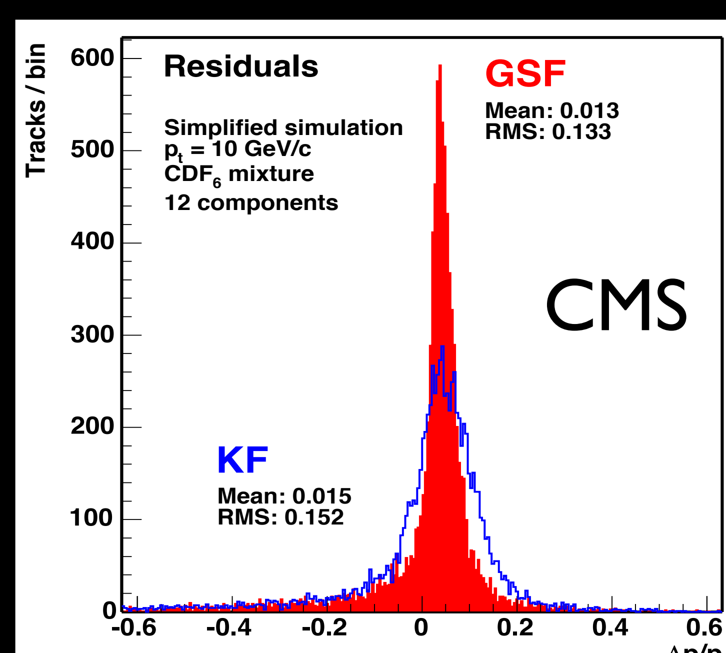


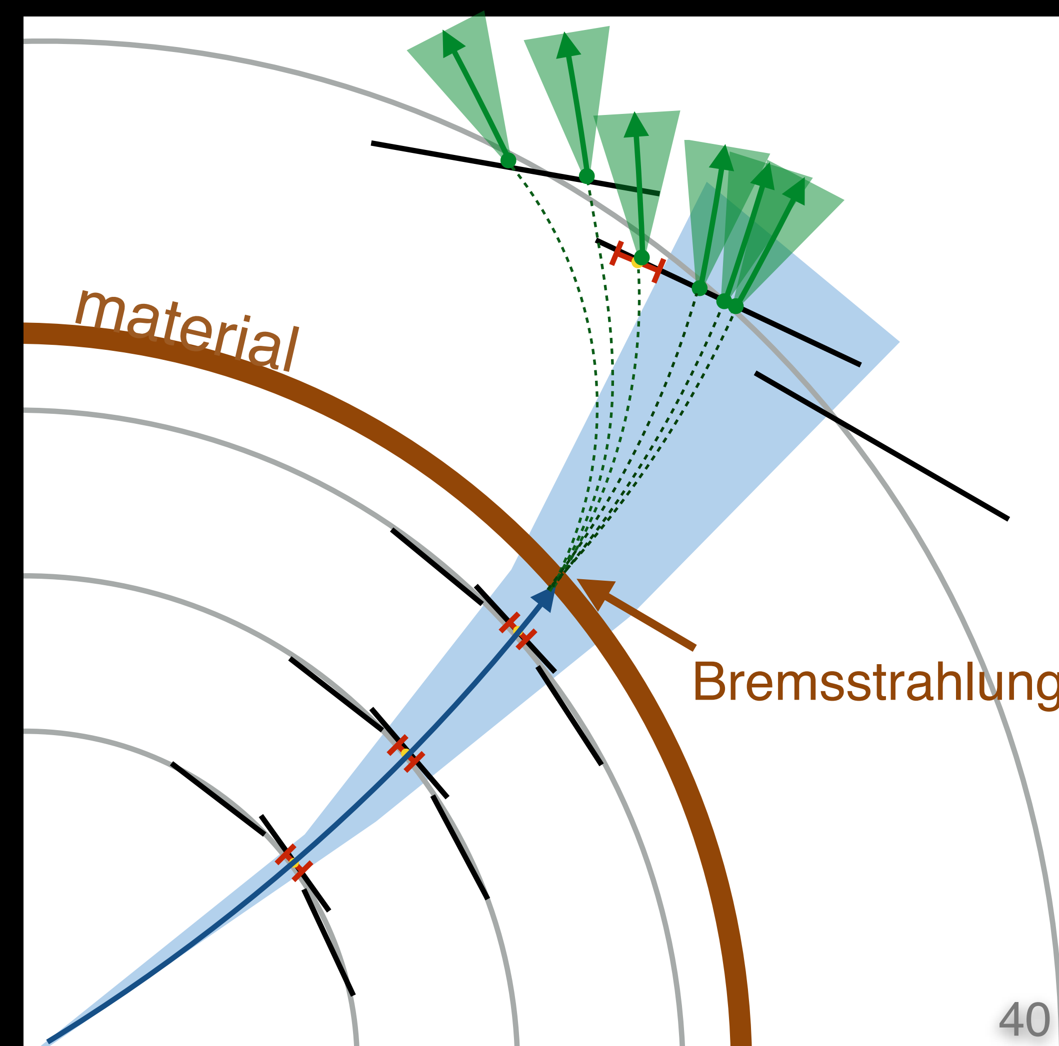


The Gaussian Sum Filter

- approximate Bethe-Heitler distribution as Gaussian mixture
  - ➡ state vector after material correction becomes sum of Gaussian components
    - relative weights from Bethe-Heitler distribution
    - GSF step resembles set of parallel Kalman Filters
    - computationally expensive!
  - ➡ computational reduction to avoid combinatorial explosion after several material layers
    - re-evaluate weights of components based on compatibility with hits
    - drop components with too low weights
  - ➡ GSF improves fit performance w.r.t. Kalman Filter

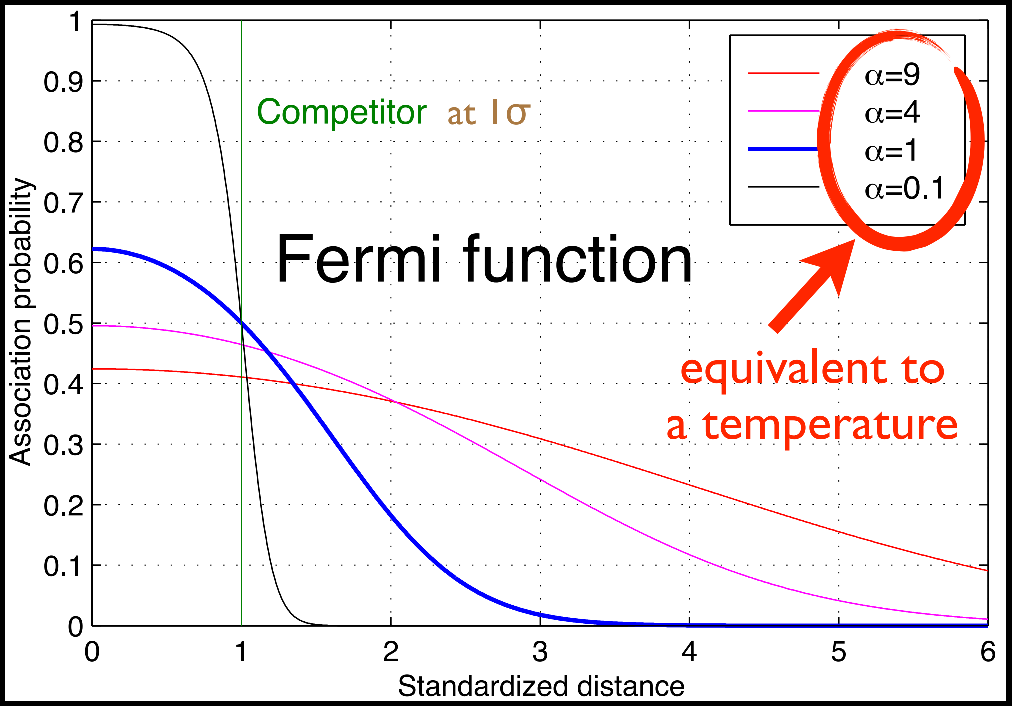






Deterministic Annealing Filters

- robust technique
  - ➡ developed for fitting with high occupancies
    - e.g. ATLAS TRT with high event pileup
    - reconstruction of 3-prong τ decays
  - ➡ can deal with several close by hits on a layer



- adaptive fit

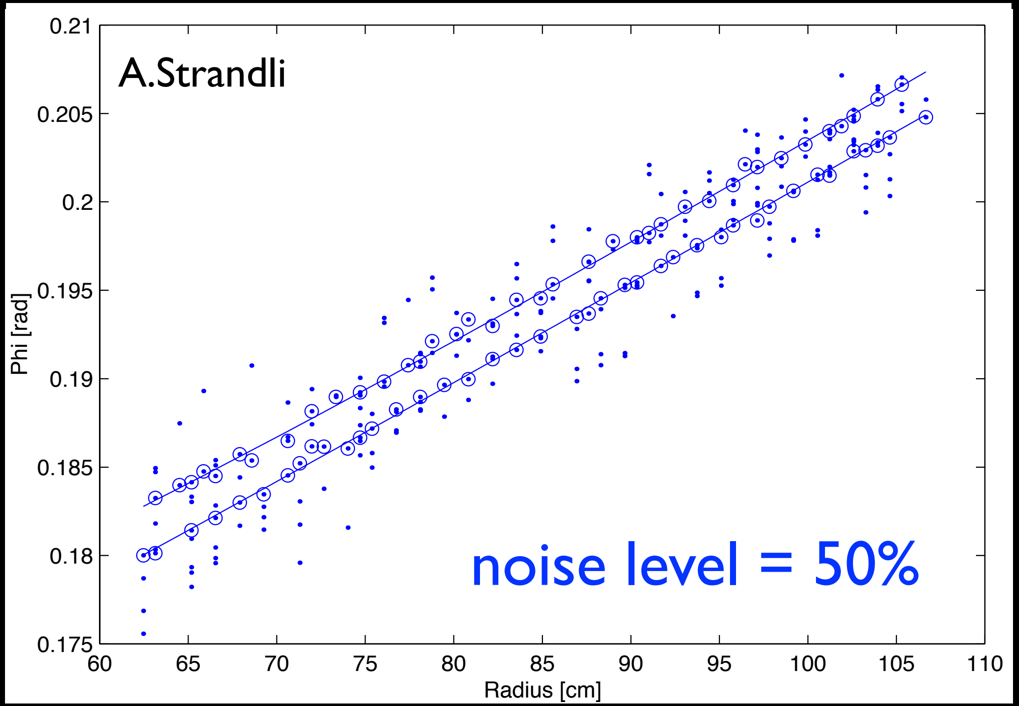
$$p_{ik} = \frac{\exp\left(-\hat{d}_{ik}^2/T\right)}{\sum_{j=1}^{n_k} \exp\left(-\hat{d}_{jk}^2/T\right)}$$

Boltzman factor

…ach hit in layer with …lity with: $\hat{d}_{ik} = d_{ik}/\sigma_k$

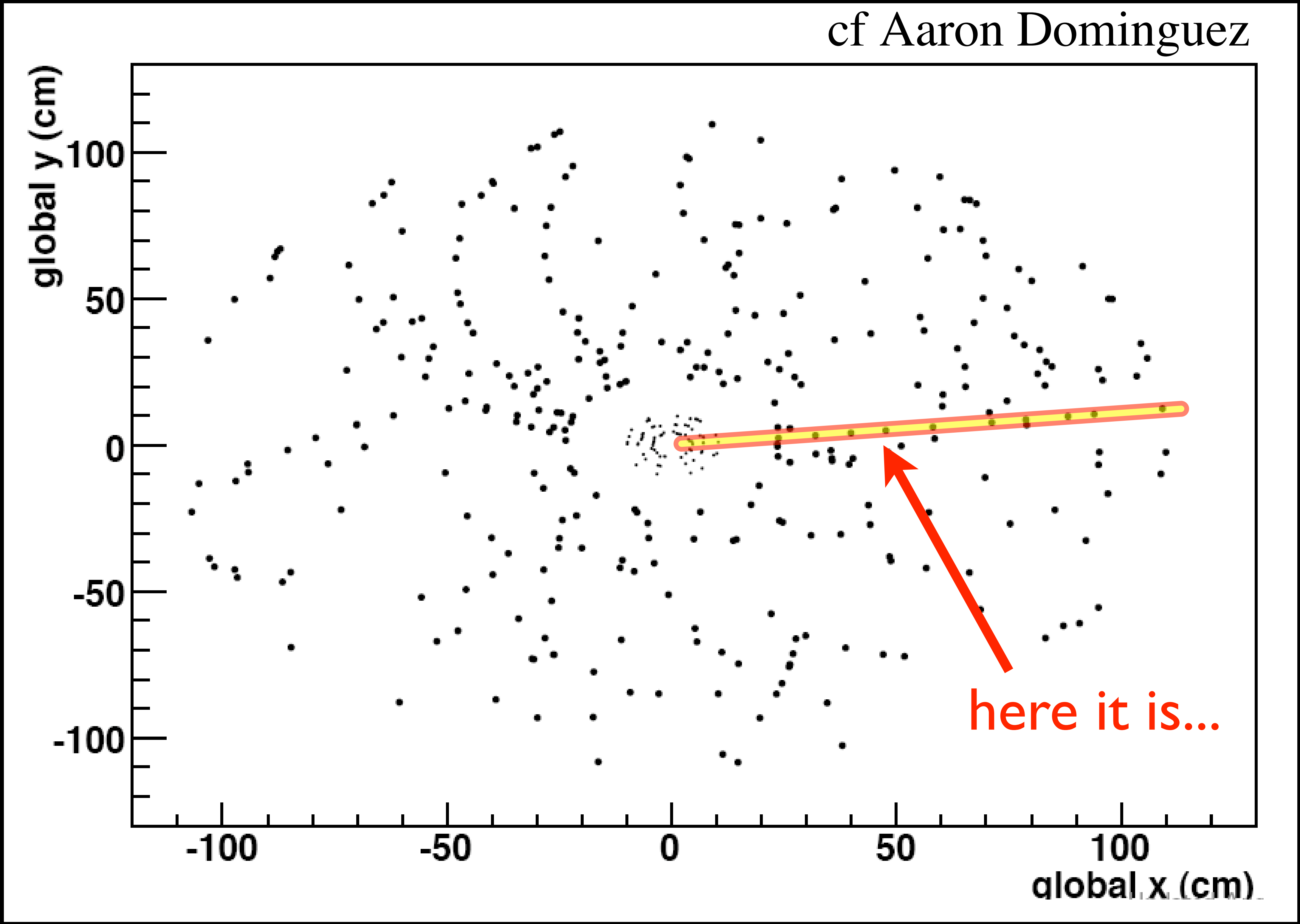
normalised distance

  - ➡ process decreasing temperature T is called annealing (iterative)
    - start at high T ~ all hits contribute same
    - at low T ~ close by hits remain
  - ➡ can be written as a Multi Track Filter



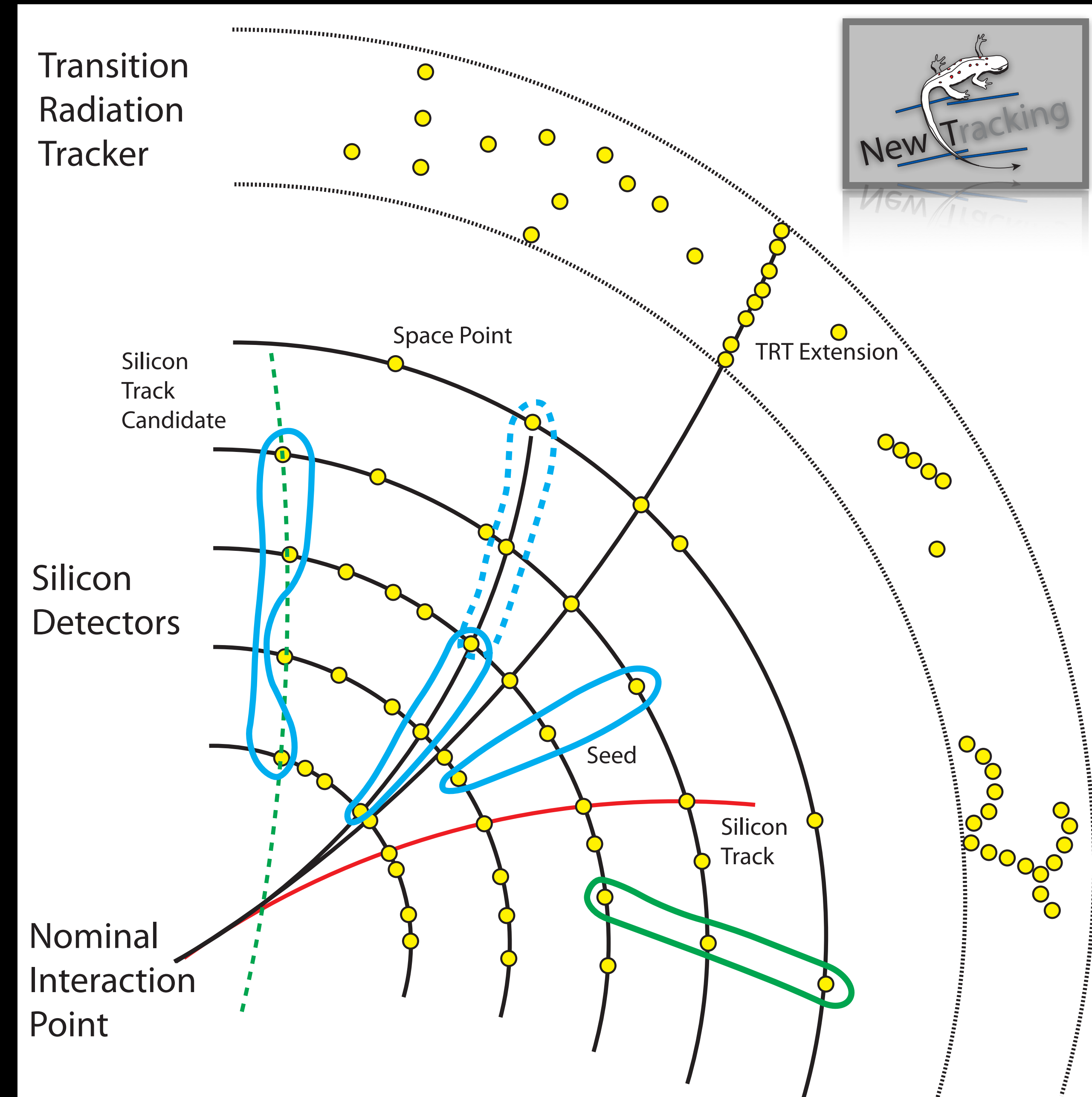
Track Finding

Track Finding: Can you find the 50 GeV



Track Finding

- the task of the track finding
  - ➡ identify track candidates in event
  - ➡ cope with combinatorial explosion of possible hit combinations
- different techniques
  - ➡ rough distinction: local/sequential and global/parallel methods
  - ➡ local method: generate seeds and complete them to track candidates
  - ➡ global method: simultaneous clustering of detector hits into track candidates
- some local methods
  - ➡ track road
  - ➡ track following
  - ➡ progressive track finding



- some global methods
  - ➡ conformal mapping
    - Hough and Legendre transform
  - ➡ adaptive methods
    - Elastic Net, Cellular Automaton ...
  - ➡ segment merging

    (will only discuss conformal mapping)

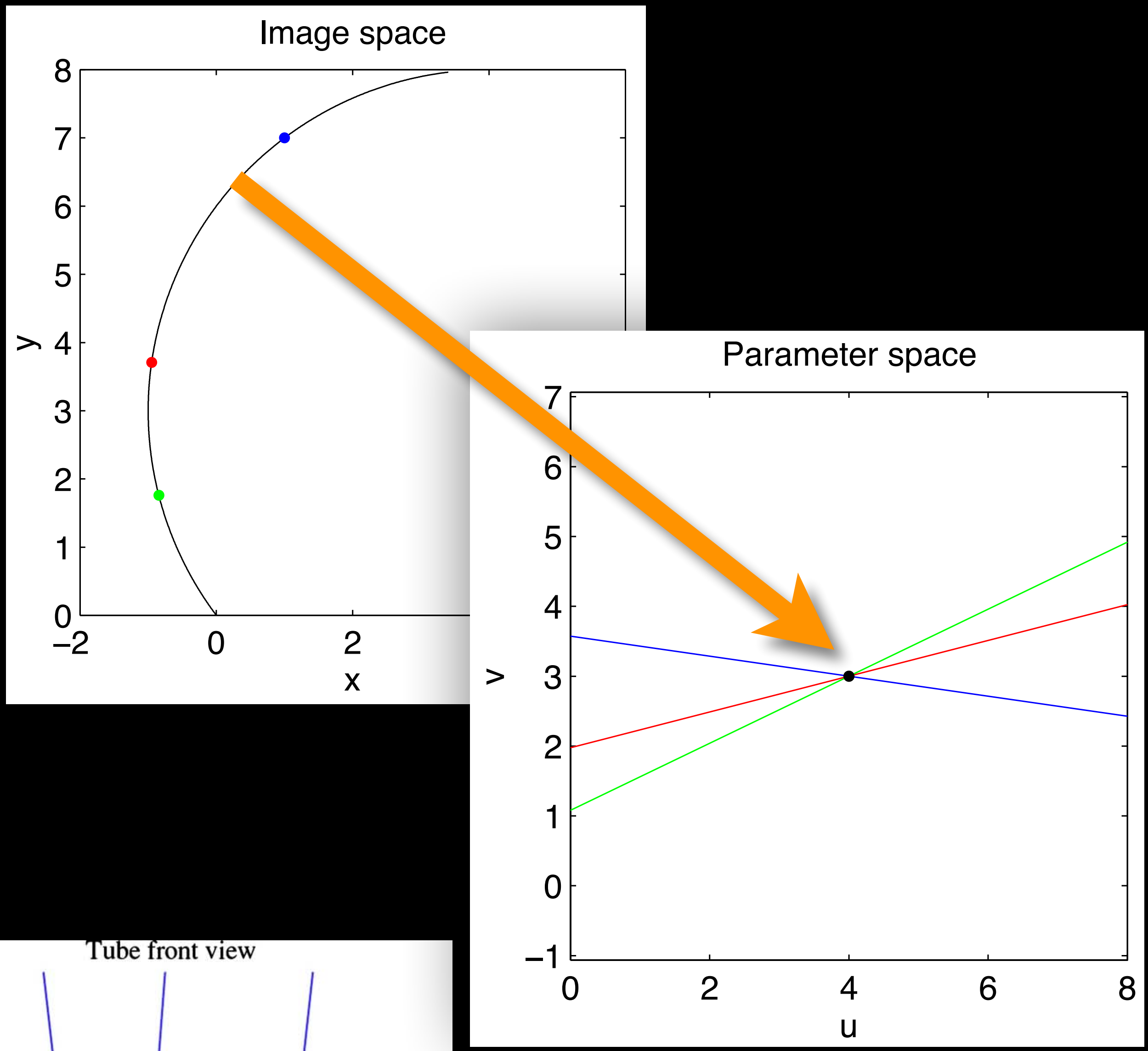
Conformal Mapping

- **Hough transform**
  - ➡ cycles through the origin in x-y transform into point in u-v

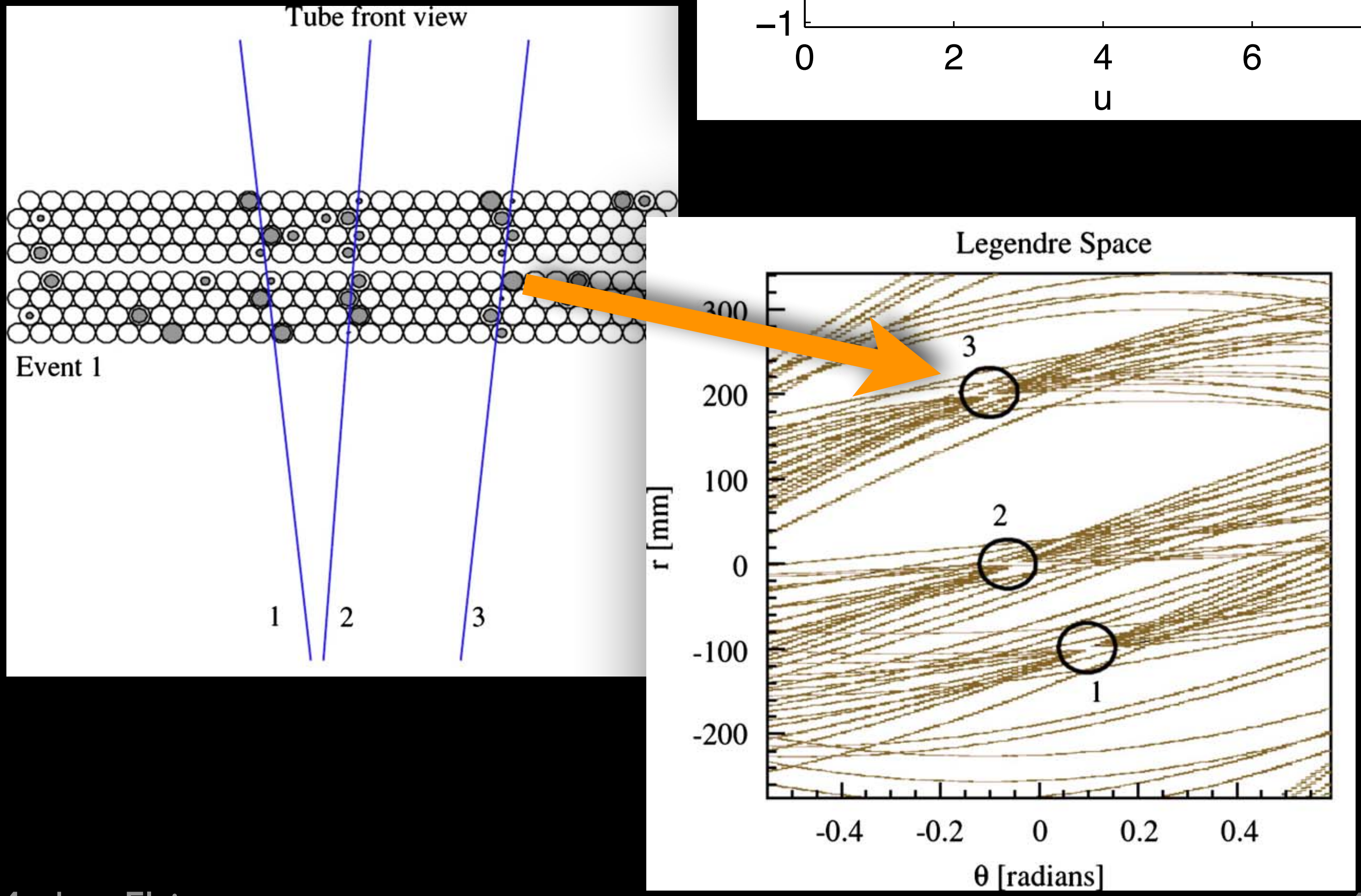
$$u = \frac{x}{x^2 + y^2}, \quad v = \frac{y}{x^2 + y^2}$$

$$\Rightarrow \quad v = -\frac{x}{y}u + \frac{x^2 + y^2}{2y}$$

  - • each hit becomes a straight line
  - ➡ search for maxima (histogram) in parameter space to find track candidates

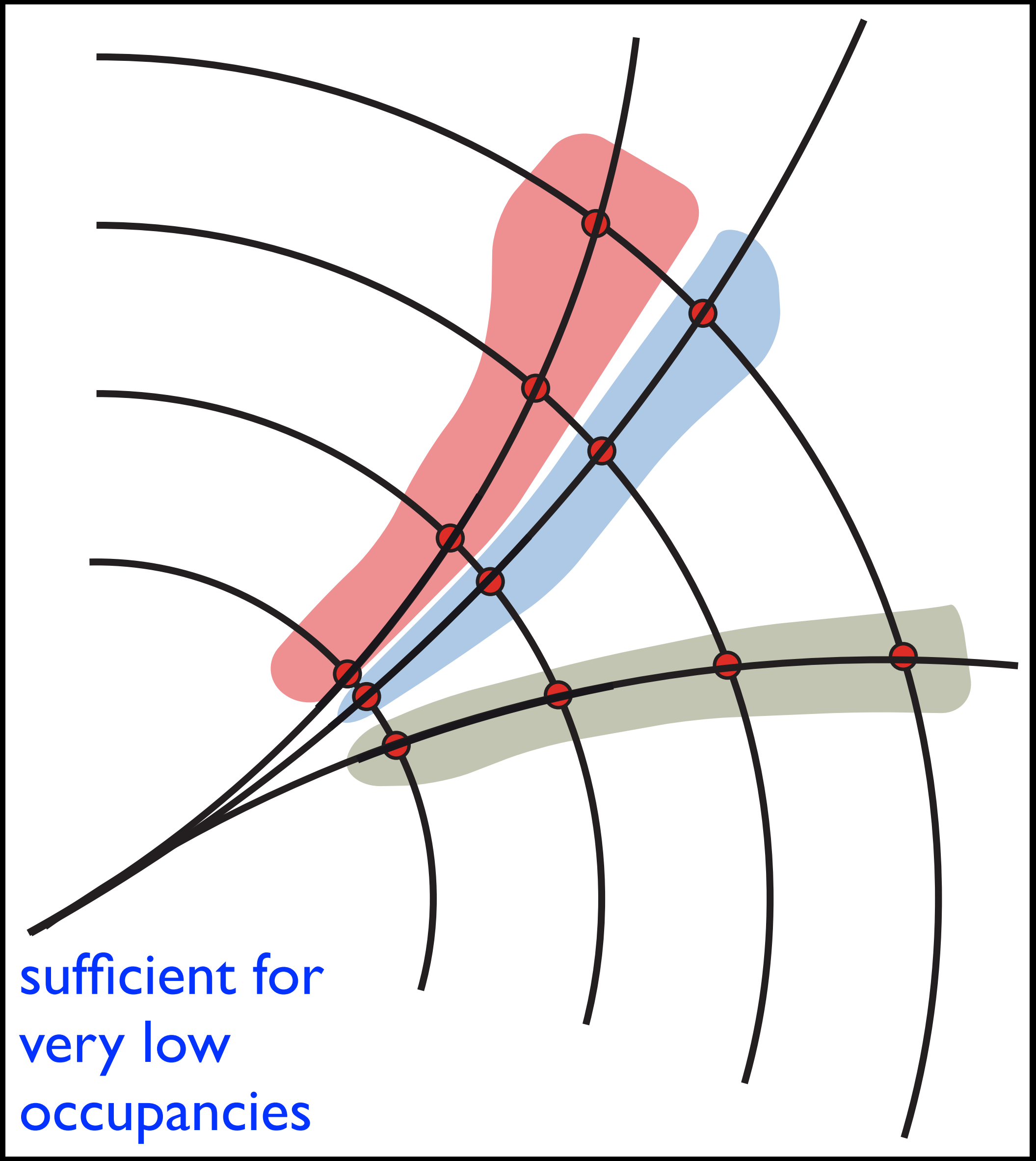


- **Legendre transform**
  - ➡ used for track finding in drift tubes
  - ➡ drift radius is transformed into sine-curves in Legendre space
  - ➡ solves as well L-R ambiguity



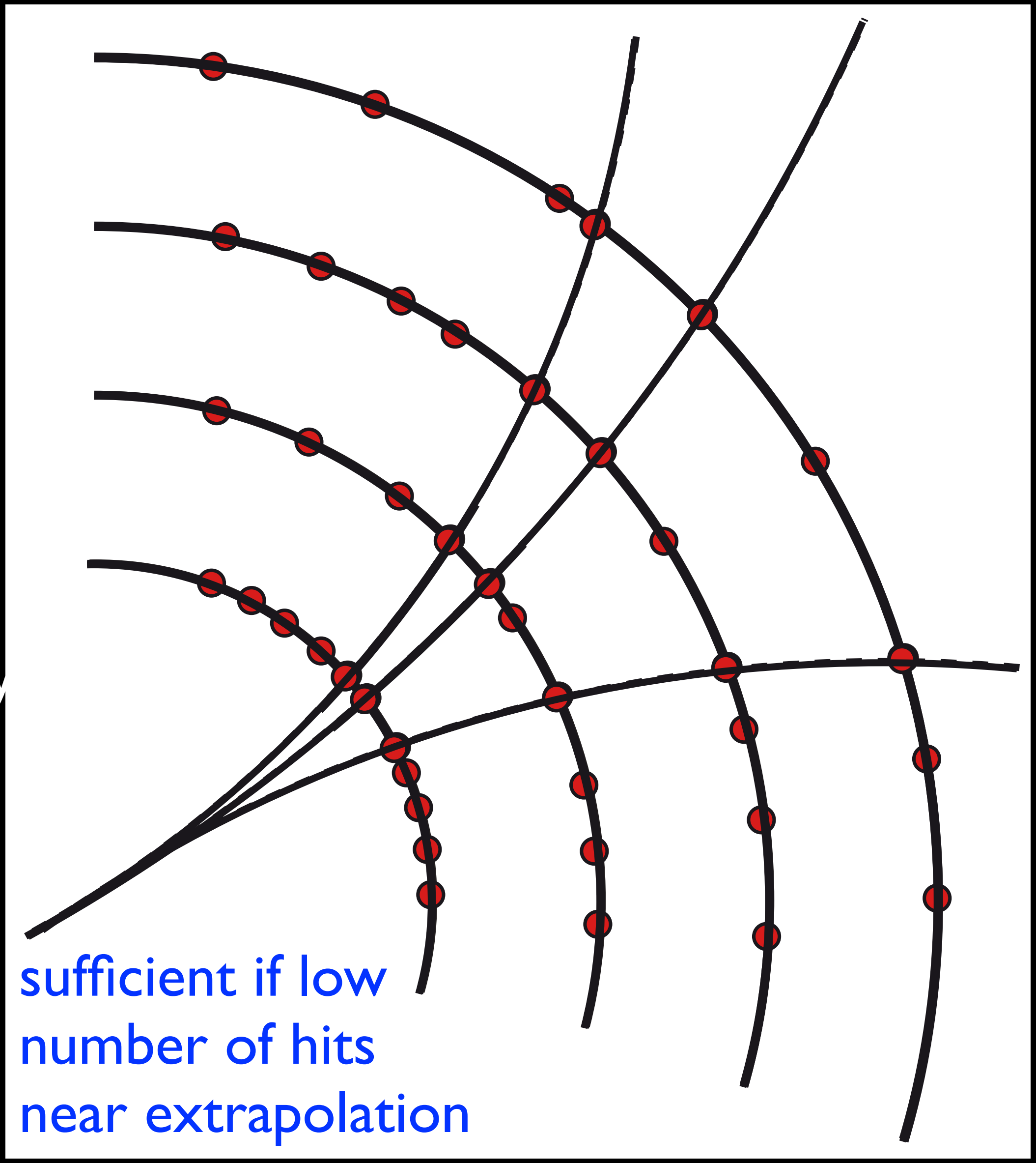
Local Track Finding

- Track Road algorithm
  - ➡ find seeds ~ combinations of 2-3 hits
  - ➡ build road along the likely trajectory
  - ➡ select hits on layers to obtain candidates



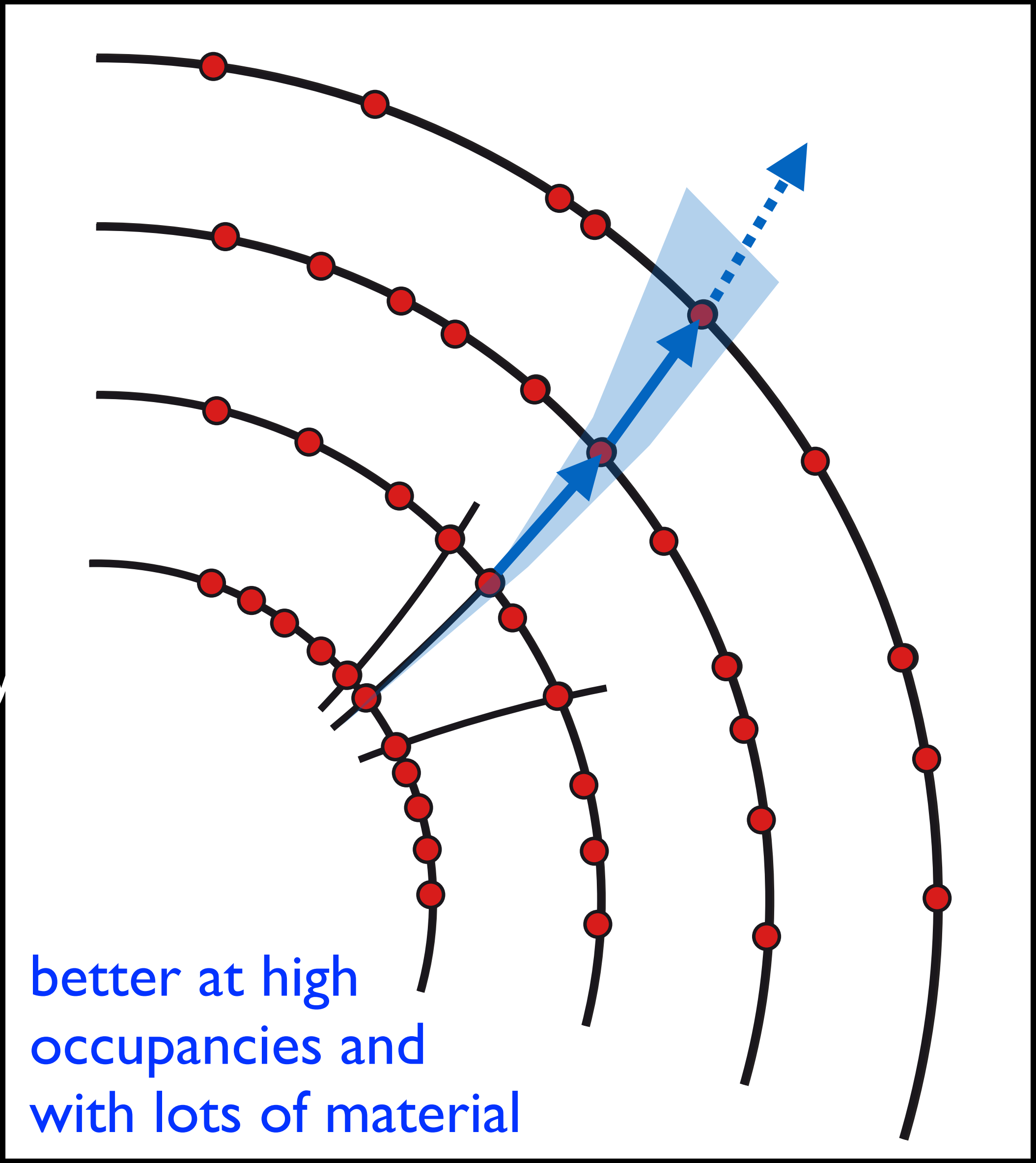
Local Track Finding

- Track Road algorithm
  - ➡ find seeds ~ combinations of 2-3 hits
  - ➡ build road along the likely trajectory
  - ➡ select hits on layers to obtain candidates
- Track Following
  - ➡ find seeds ~ combinations of 2-3 hits
  - ➡ extrapolate seed along the likely trajectory
  - ➡ select hits on layers to obtain candidates



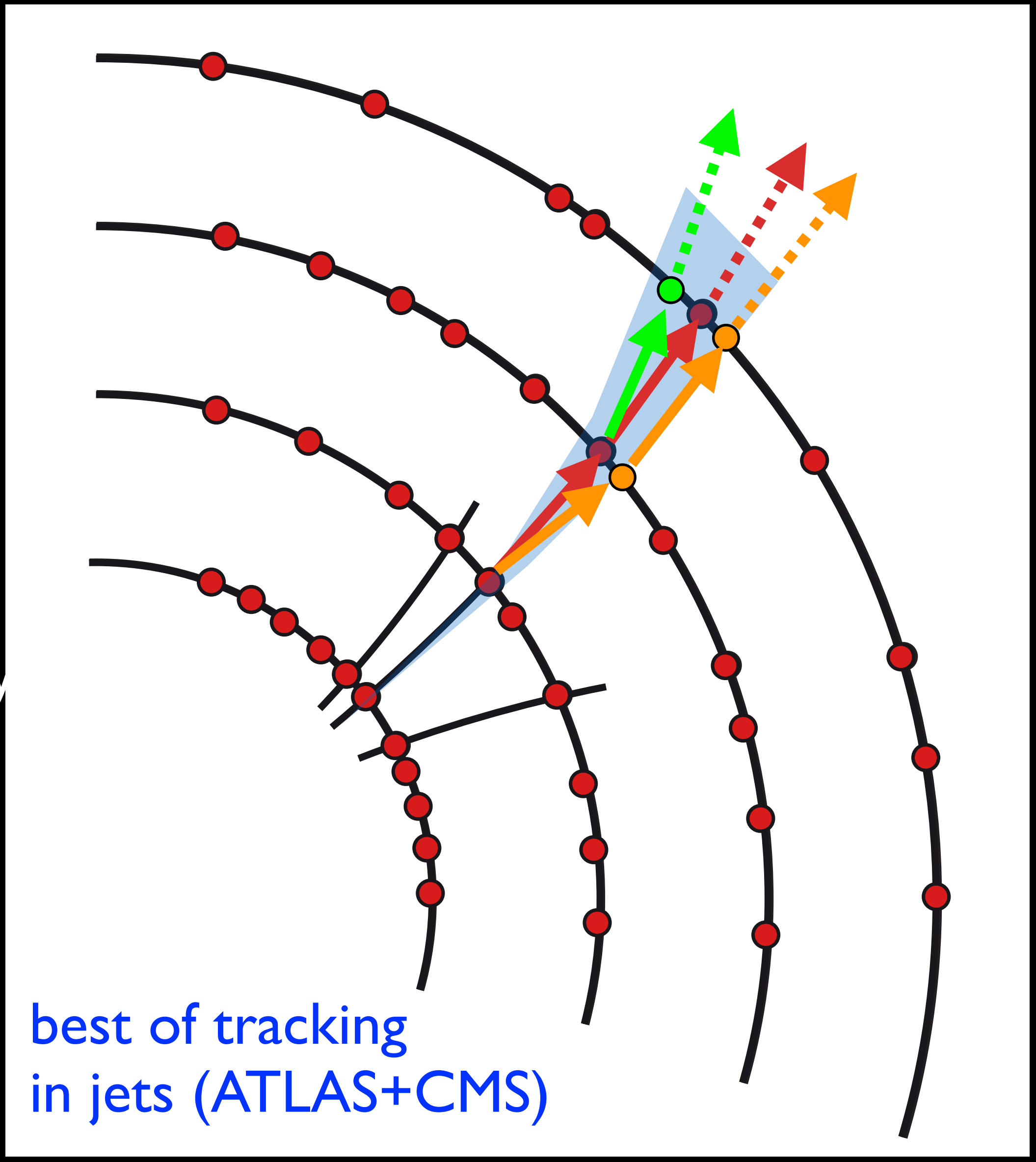
Local Track Finding

- Track Road algorithm
  - ➡ find seeds ~ combinations of 2-3 hits
  - ➡ build road along the likely trajectory
  - ➡ select hits on layers to obtain candidates
- Track Following
  - ➡ find seeds ~ combinations of 2-3 hits
  - ➡ extrapolate seed along the likely trajectory
  - ➡ select hits on layers to obtain candidates
- Progressive Track Finder
  - ➡ find seeds ~ combinations of 2-3 hits
  - ➡ extrapolate seed to next layer, find best hit and update trajectory
  - ➡ repeat until last layers to obtain candidates



Local Track Finding

- Track Road algorithm
  - ➡ find seeds ~ combinations of 2-3 hits
  - ➡ build road along the likely trajectory
  - ➡ select hits on layers to obtain candidates
- Track Following
  - ➡ find seeds ~ combinations of 2-3 hits
  - ➡ extrapolate seed along the likely trajectory
  - ➡ select hits on layers to obtain candidates
- Progressive Track Finder
  - ➡ find seeds ~ combinations of 2-3 hits
  - ➡ extrapolate seed to next layer, find best hit and update trajectory
  - ➡ repeat until last layers to obtain candidates
- Combinatorial Kalman Filter
  - ➡ extension of a Progressive Track Finder for dense environments
  - ➡ full combinatorial exploration, follow all hits to find all possible track candidates



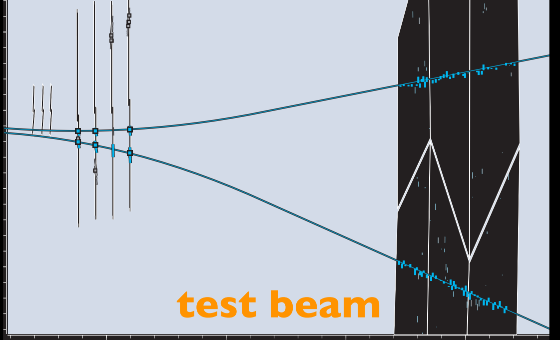
Track Reconstruction in ATLAS

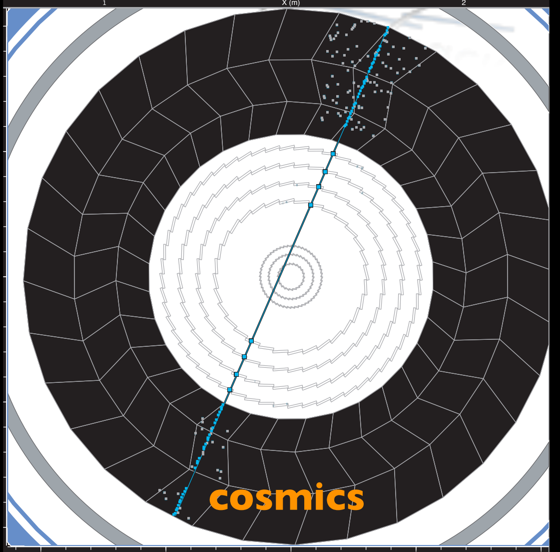
... and in Practice ?

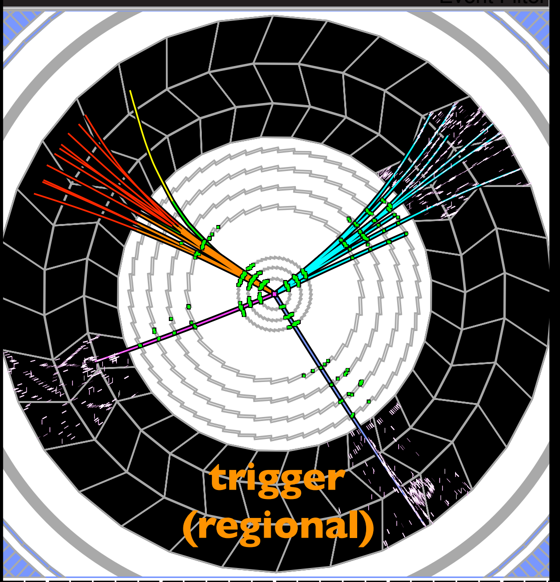
- choice of reconstruction strategy depends on:
  - ➡ detector technologies
  - ➡ physics/performance requirements
  - ➡ occupancy and backgrounds
  - ➡ technical constraints (CPU, memory)

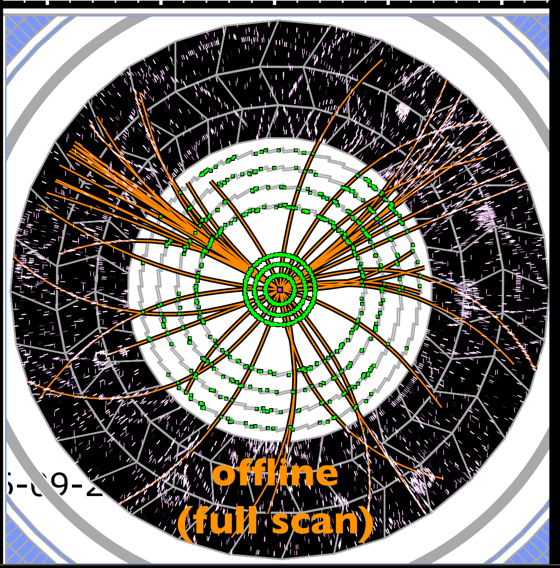
- even for same detector setup one looks at different types of events:
  - ➡ test beam
  - ➡ cosmics
  - ➡ trigger (regional)
  - ➡ offline (full scan)

- track reconstruction in use by experiments
  - ➡ many apply a combination of different techniques
  - ➡ often iterative ~ different strategies run one after the other to obtain best possible performance within resource constraints



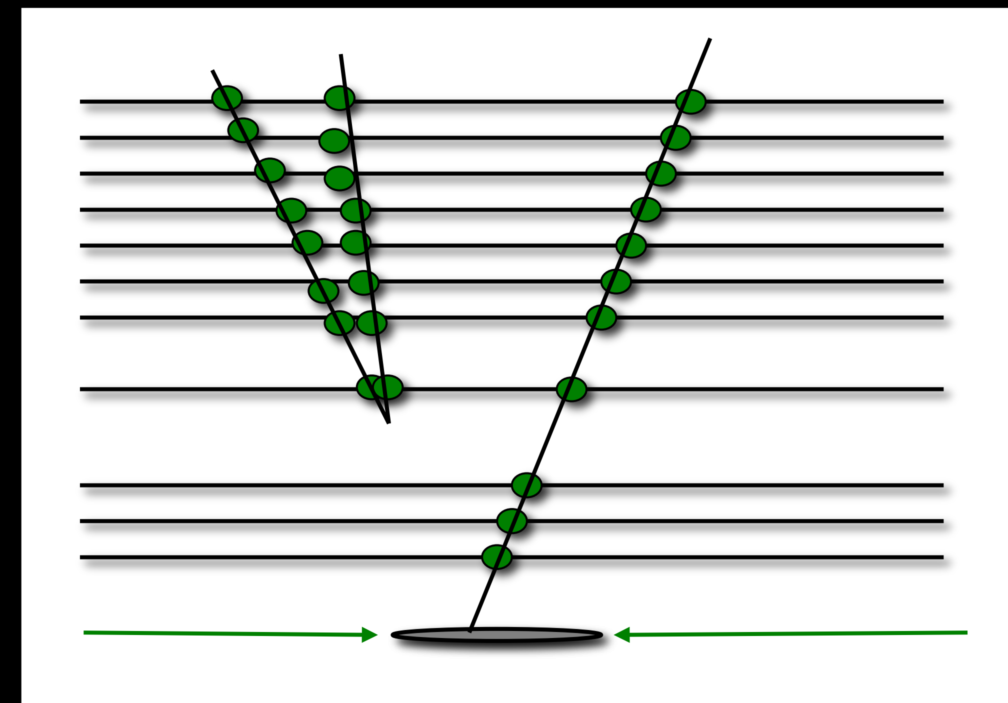


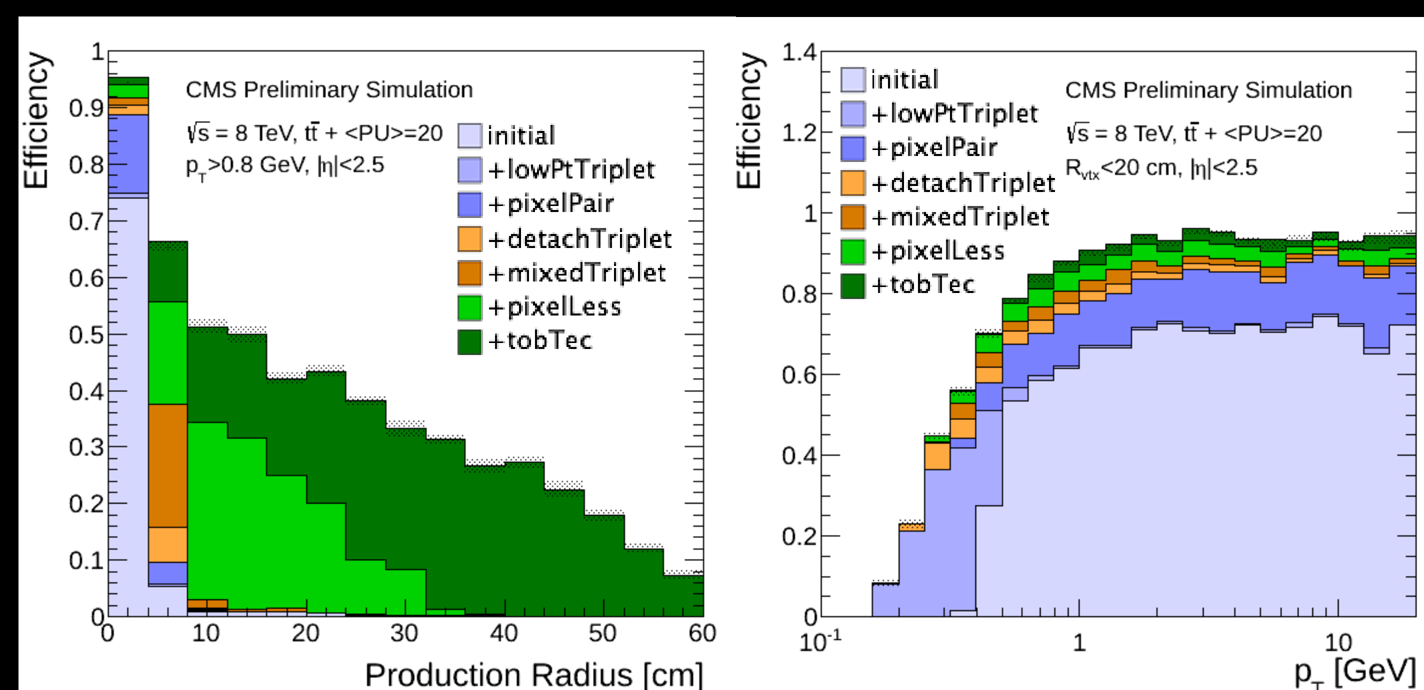




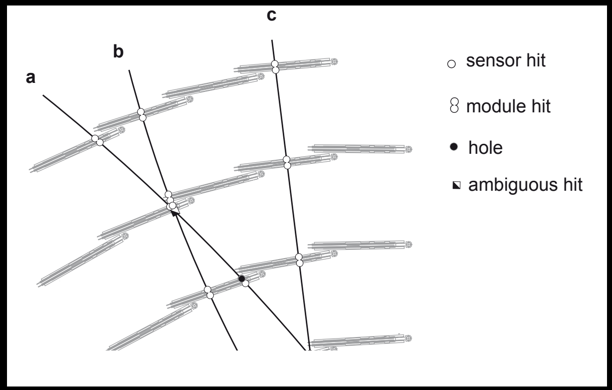
The Iterative Tracking Strategy

- task of track finding step is to find all track candidates
  - ➡ at the same time, minimise combinatorial overhead !
  - ➡ restrict seeding for combinatorial Kalman Filter to set of layers
- iterative seeding approach:
  - ➡ find initial set of track candidates
  - ➡ remove used hits from event
  - ➡ seed tracking from different set of layers to find more tracks
  - ➡ ... etc.
  - ➡ optimal choice of iterative seeding strategy is matter of tuning (e.g. CMS did 7 iterations in Run-1)

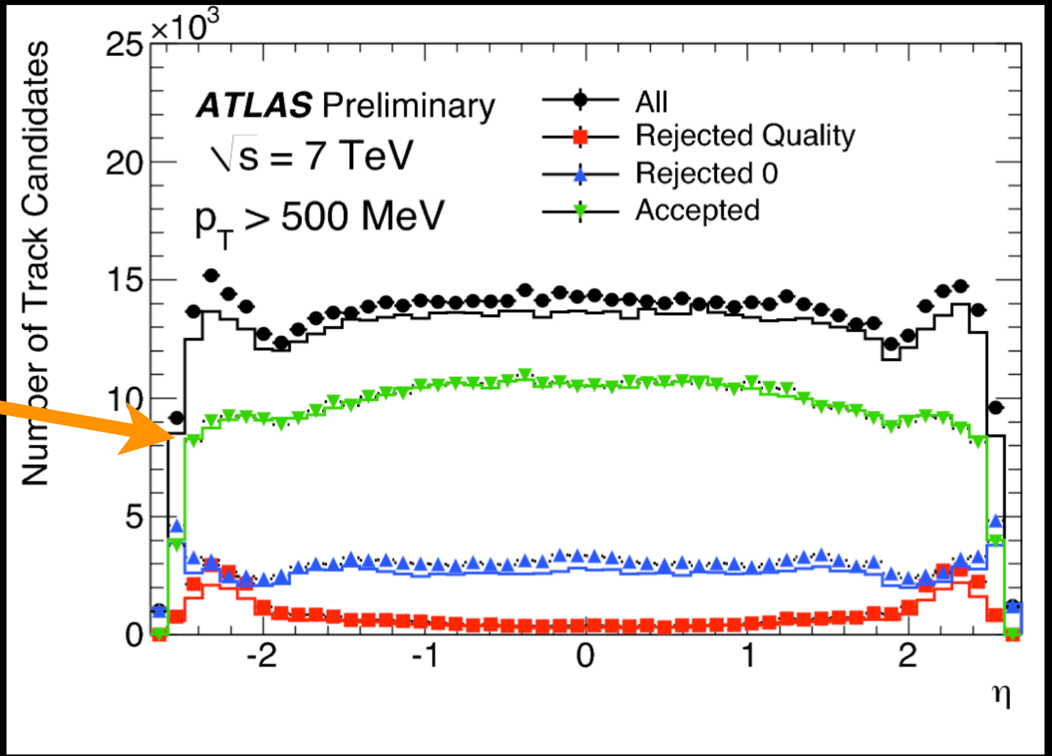




The Ambiguity Solution

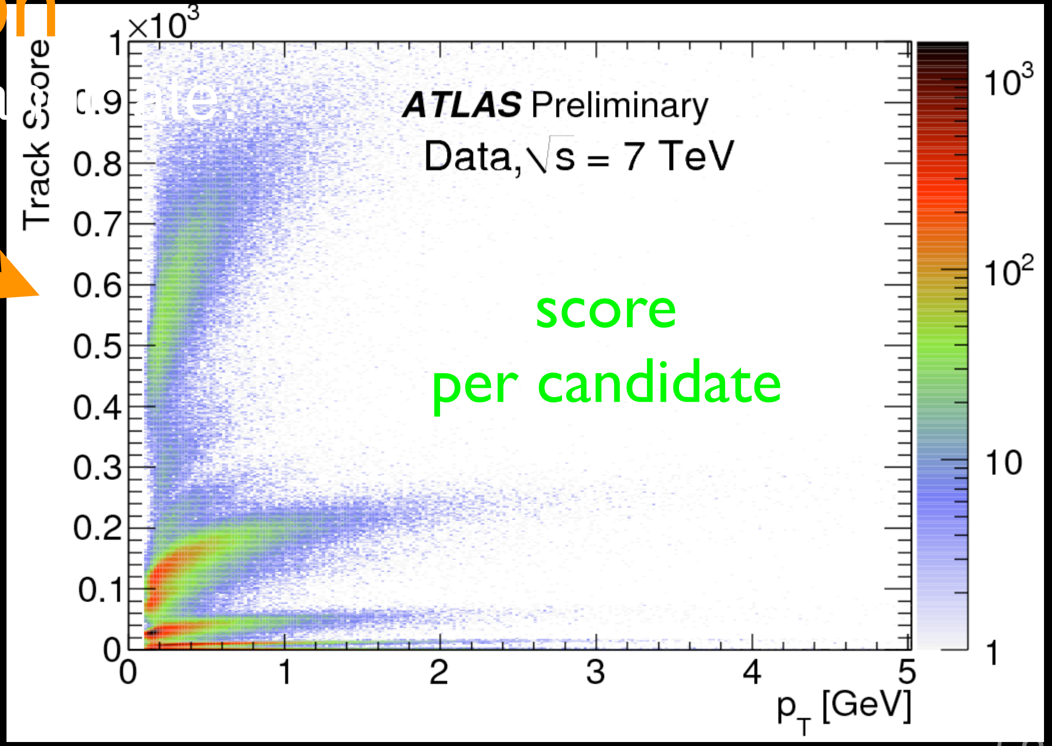


- track selection cuts
  - ➡ applied at every stage in reconstruction
  - ➡ still more candidates than final tracks and too high rate of fakes



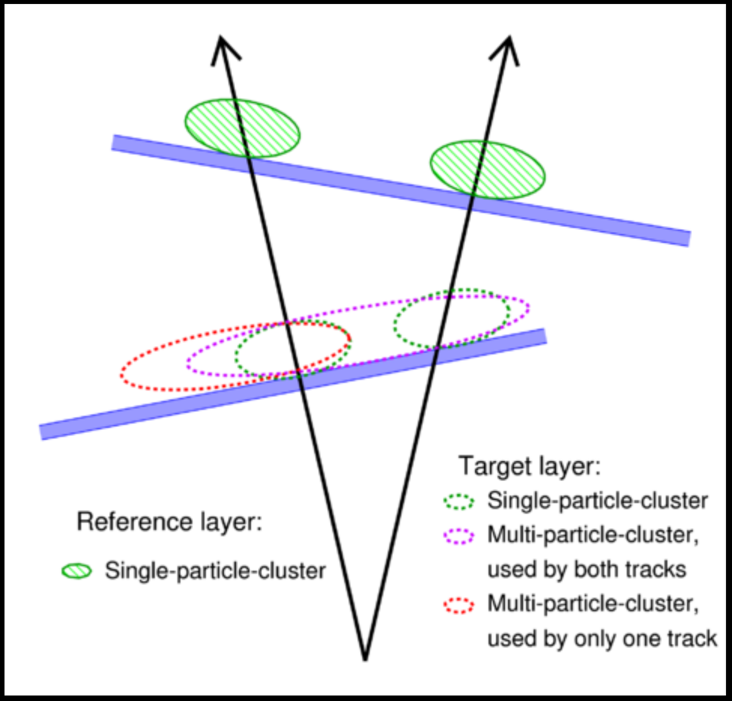
- task of ambiguity solution
  - ➡ select good tracks and reject fakes
  - ➡ ATLAS: precise fit with outlier removal, NN cluster splitting and Brem. recovery

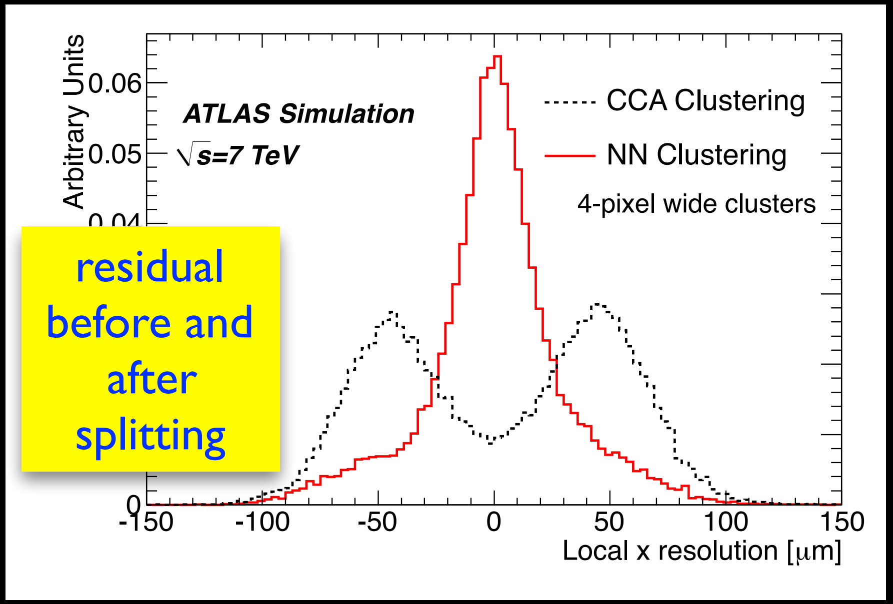
- ATLAS: ordered iterative selection
  1. evaluate quality function ("score") for each candidate
     - hit content, holes
     - number of shared hits
     - fit quality...
  2. candidate with best score wins
  3. if too many shared hits, create sub-track if candidate with remaining hits passes cuts

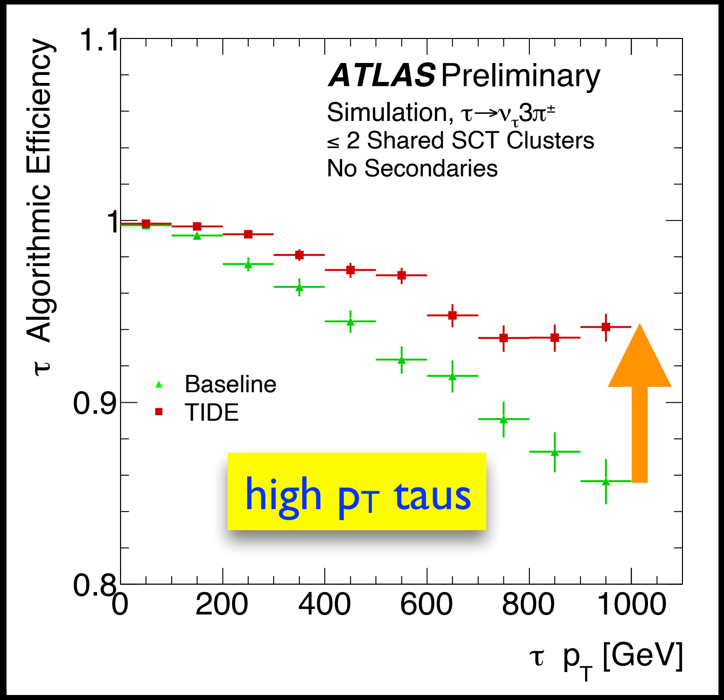


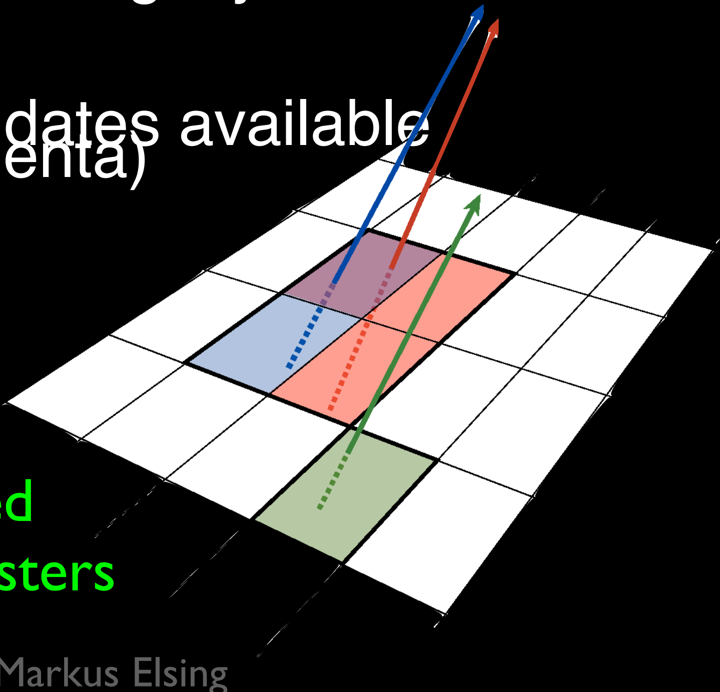
Resolving dense Jets

- problem of cluster merging
  - ➡ merging when track separation reaches single Pixel size
  - ➡ during track reconstruction shared clusters are penalised to reduce fakes and duplicate tracks
- NN cluster splitting in Pixels
  - ➡ identify merged clusters and splitting them
    - identify merge clusters, split them and correct positions
  - ➡ splitting/sharing decision done in ambiguity processing
    - full track information for all candidates available
- crucial in many areas:
  - ➡ b-tagging (especially at high momenta)
  - ➡ jet calibration and particle flow
  - ➡ 3-prong τ identification









shared
Pixel clusters

ATLAS NewTracking Software Chain

pre-processing
- ➡ Pixel+SCT clustering
- ➡ TRT drift circle formation
- ➡ space points formation

combinatorial track finder
- ➡ iterative :
  1. Pixel seeds
  2. Pixel+SCT seeds
  3. SCT seeds
- ➡ restricted to roads
- ➡ Brem.recovery in EM Regions-of-Interest

ambiguity solution
- ➡ runs hole search
- ➡ scores tracks according to quality
- ➡ NN cluster splitting in jets
- ➡ precise least square fit with Brem.recovery
- ➡ final selection cuts

extension into TRT
- ➡ progressive finder
- ➡ refit of tracks with Brem.
- ➡ scoring and selection

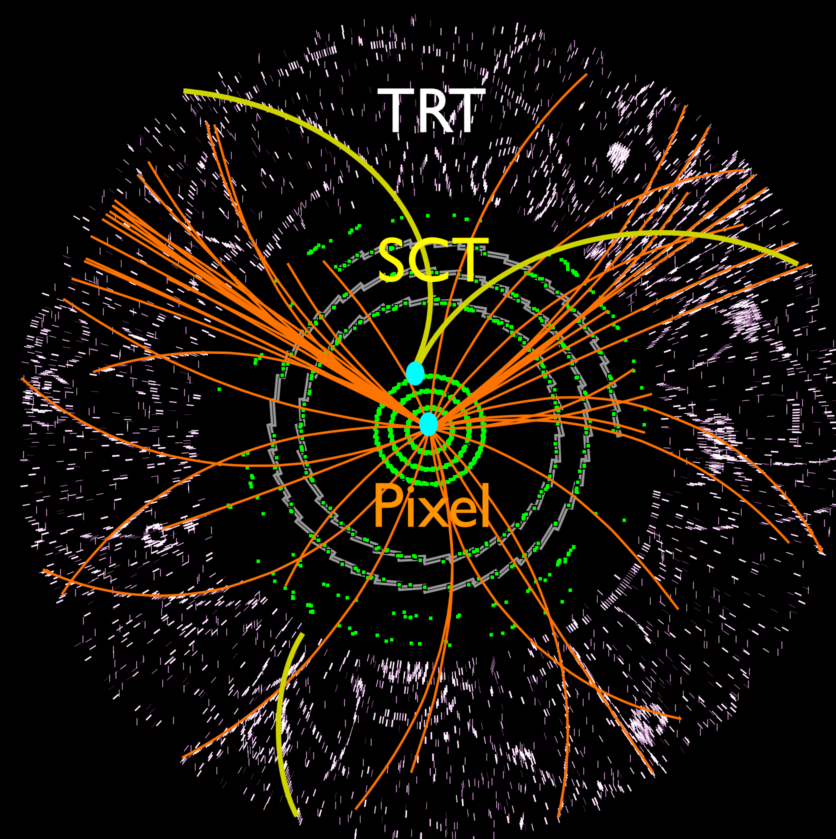
TRT segment finder
- ➡ in EM Regions-of-Interest
- ➡ on remaining drift circles

TRT seeded finder
- ➡ from TRT into SCT+Pixels
- ➡ combinatorial finder

ambiguity solution
- ➡ precise fit and selection
- ➡ TRT seeded tracks

standalone TRT
- ➡ unused TRT segments

vertexing
- ➡ primary vertexing
- ➡ conversion and V0 search



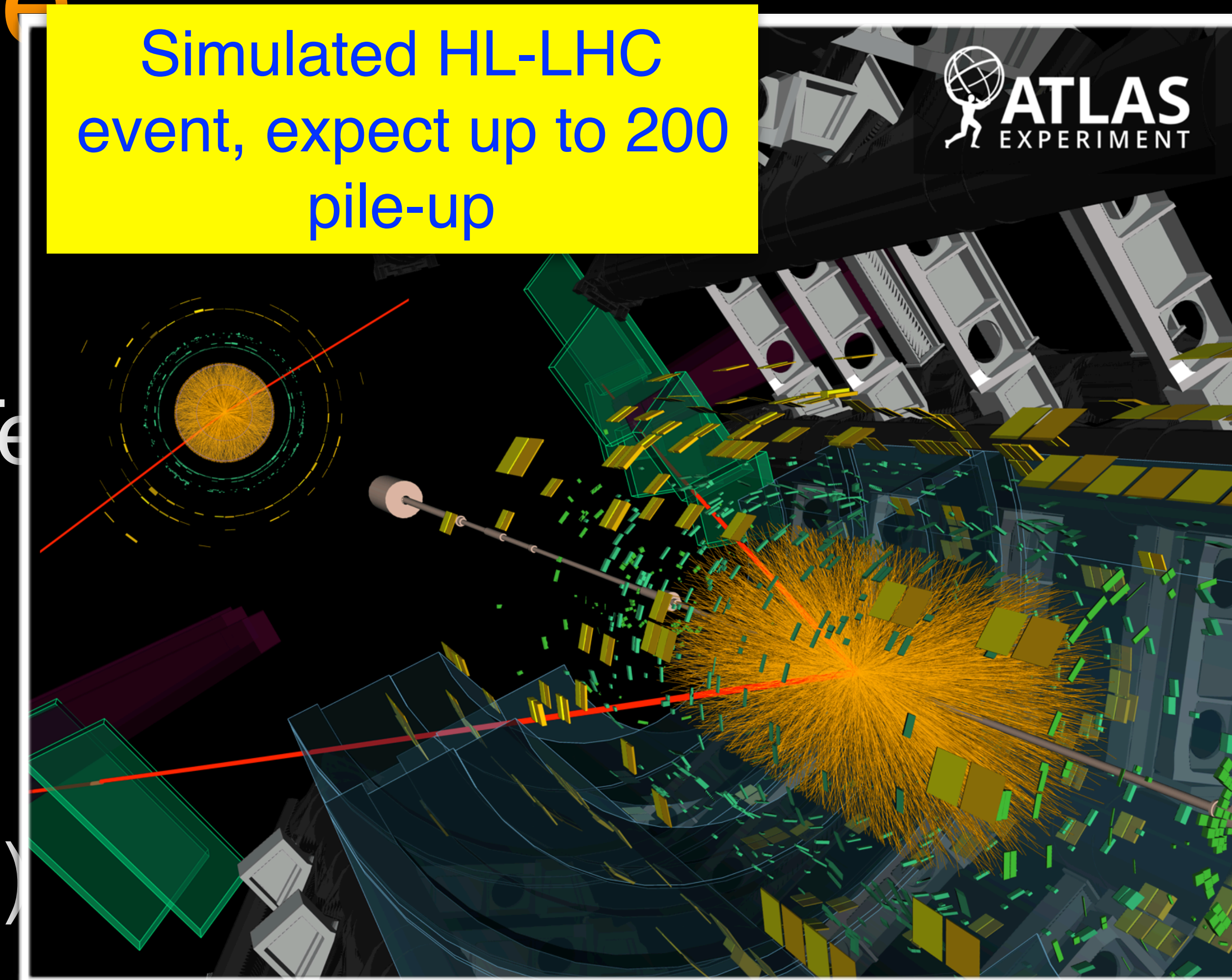
Let's Summarize...

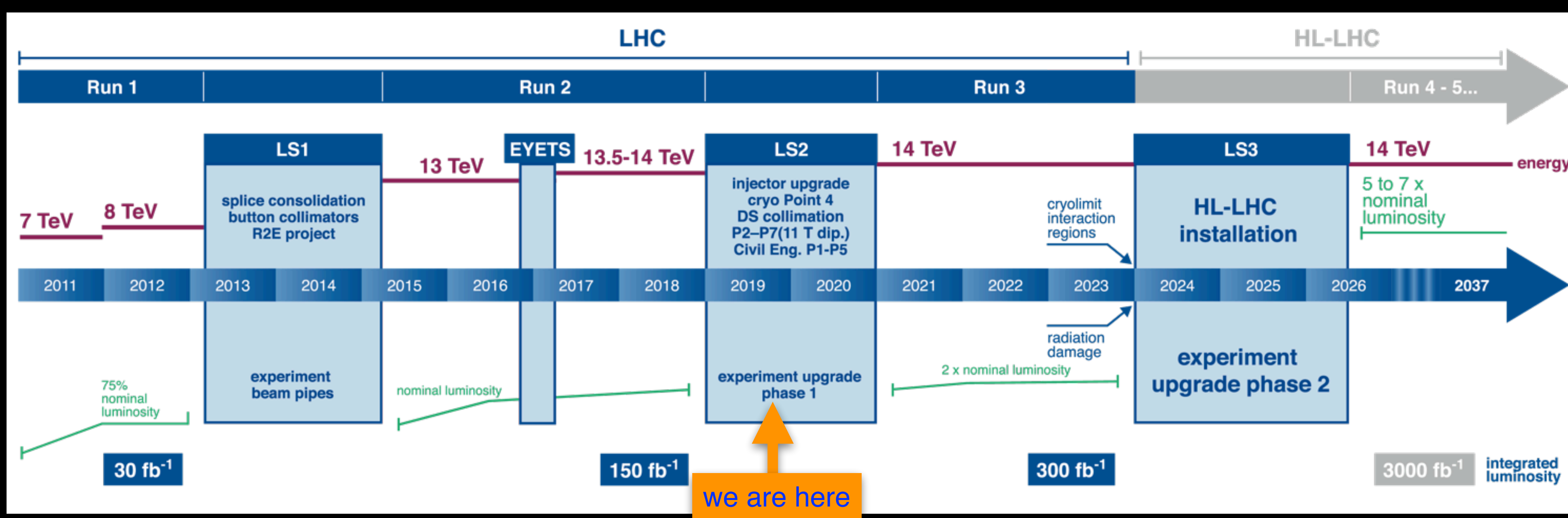
- I introduced the reconstruction in a nutshell and why tracking is important for HEP computing
- I discussed briefly the principles of semiconductor trackers and drift tubes
- we went over concepts and techniques for track extrapolation, fitting and finding
- we saw how to put things together to implement the ATLAS Track Reconstruction

Bonus: Towards HL-LHC

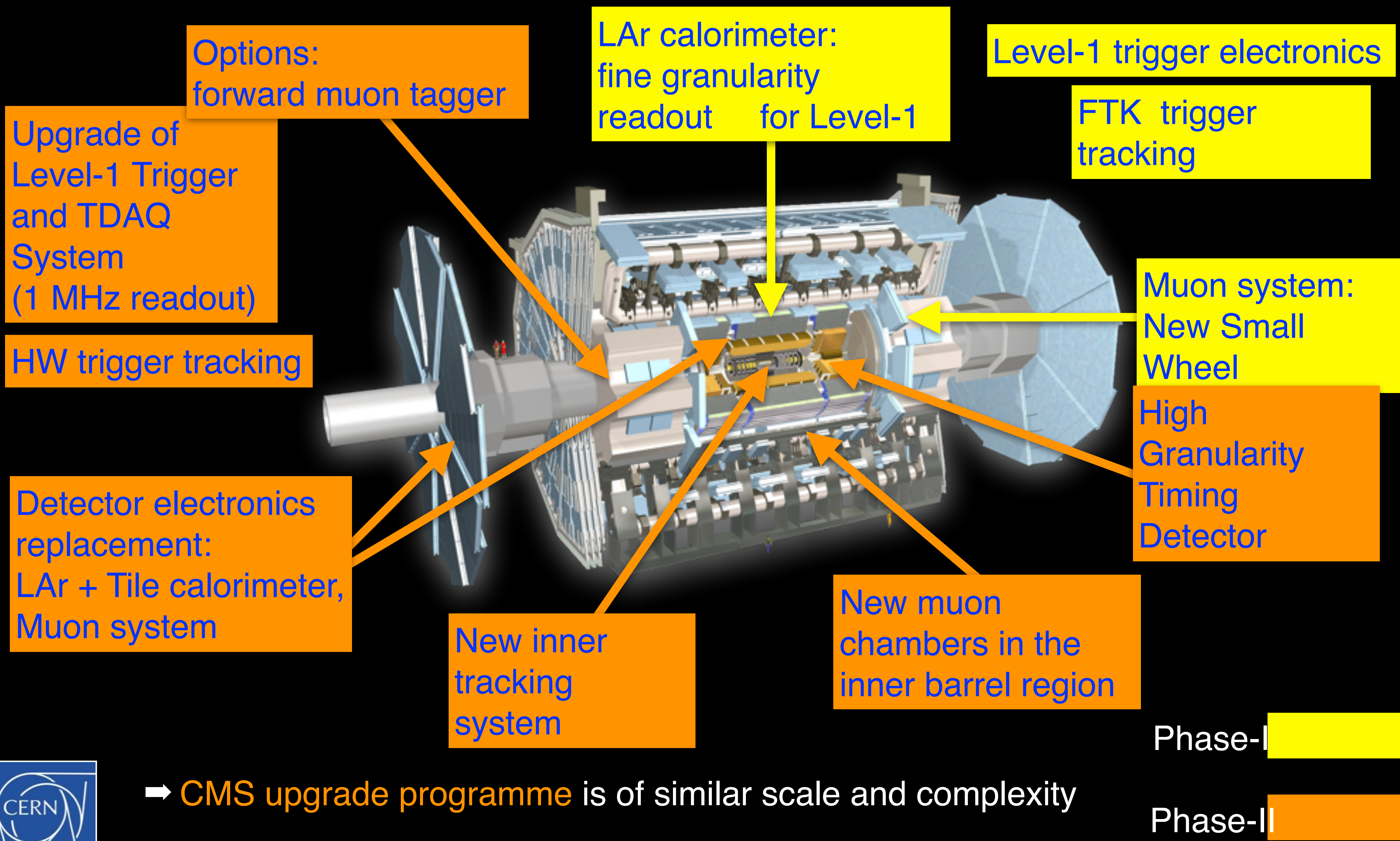
Preparing for the Future

- current Long-Shutdown 2
  - ➡ Phase-1 upgrade
  - ➡ first set of upgrades for ATLAS+CMS
- Run-3 to collect 300 fb^{-1} at 14 Te
- Long-Shutdown 3
  - ➡ Phase-2 upgrade
  - ➡ major upgrade of ATLA+CMS experiment
- High Luminosity LHC (3000 fb^{-1})





ATLAS Phase-I and Phase-II Upgrades



➡ CMS upgrade programme is of similar scale and complexity

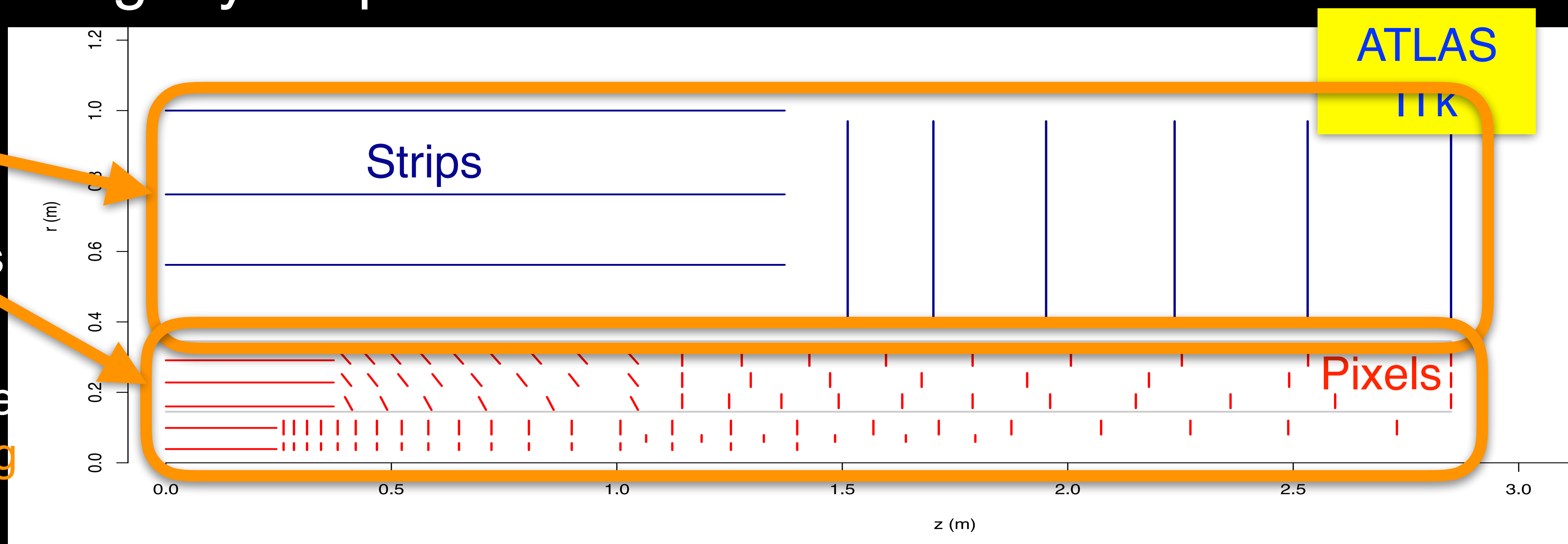
Comparison of new ATLAS and CMS

- tried scaling layout plots to match dimensions...

➡ ATLAS ITk:
- 4 double-strip
- 5 pixel layers
- total 9 layers
- total 13 hits
- eta < 4

➡ classical strip layout:

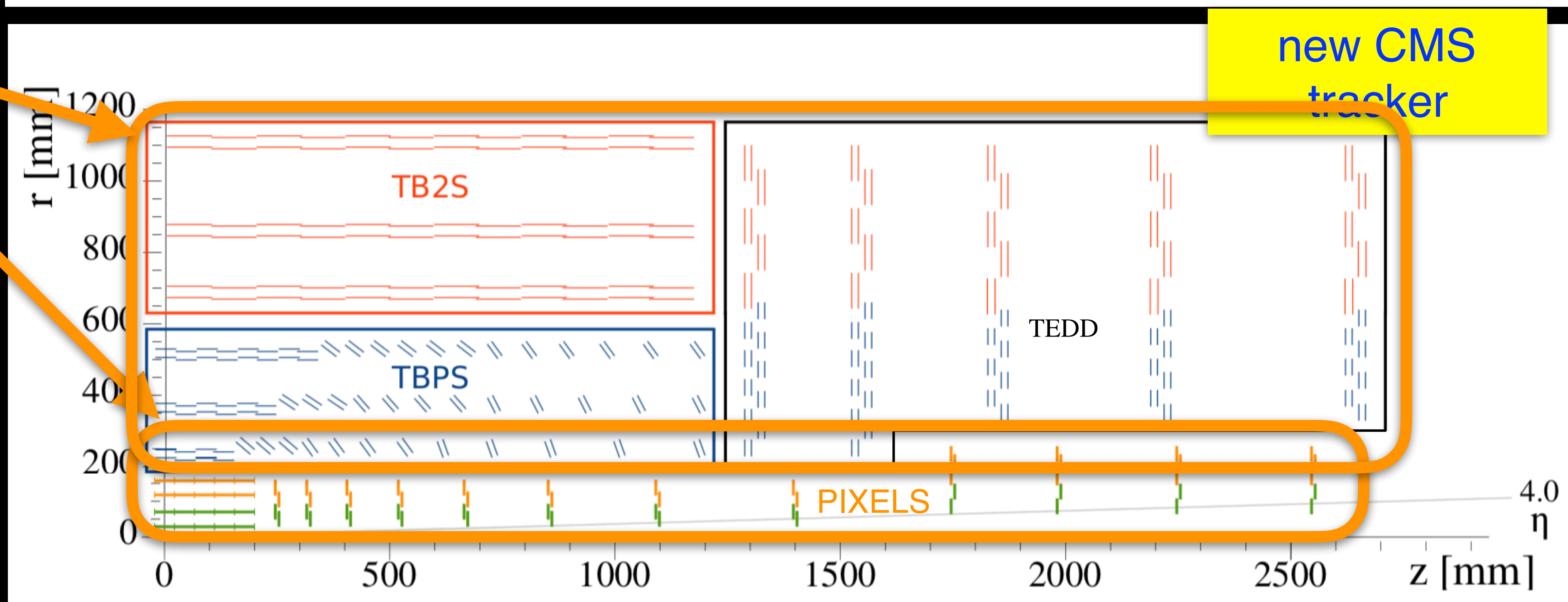
➡ novel pixel layout:
- barrel cylinders and disks in the end-caps
- optimised for fast and precise forward tracking



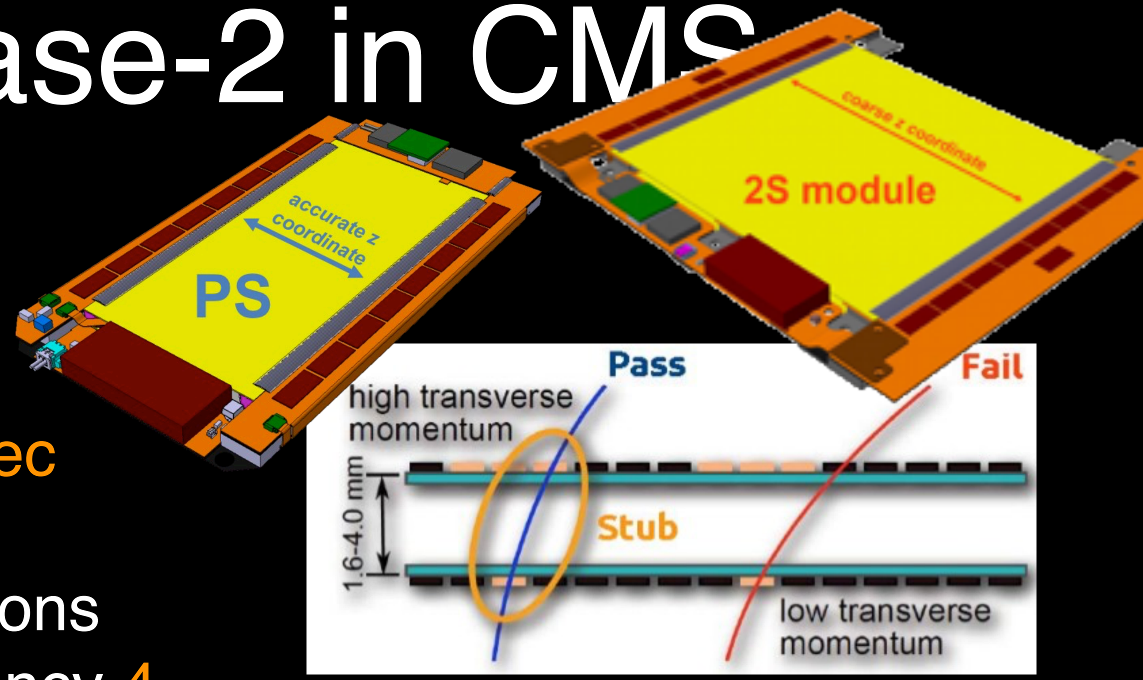
➡ CMS tracker:
- 3 double-strip
- 3 strip+pixel
- 4 pixel layers
- total 10 layers
- total 16 hits
- eta < 4

➡ novel outer layout:
- double-layer stubs for fast hardware trigger tracking

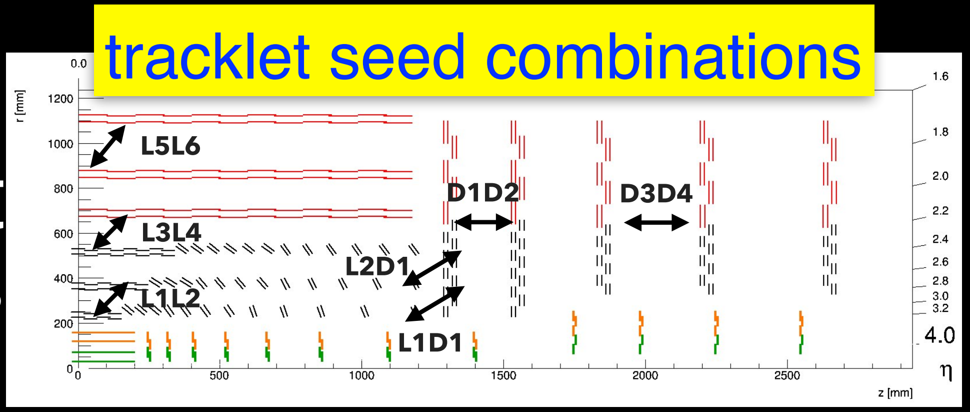
➡ classical pixel layout:
- barrel cylinders and disks in the end-caps



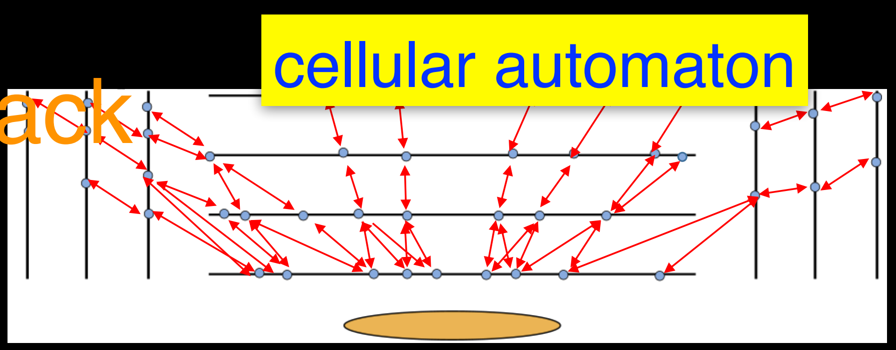
Trigger Tracking for Phase-2 in CMS



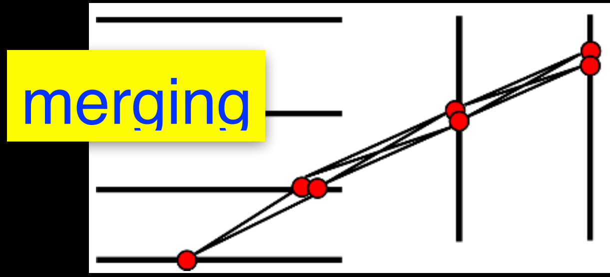
- high luminosity is a challenge for fast online trigger event selection
  - ➡ latency for level-1 trigger decision is 12.5 µsec
  - ➡ plan to use tracking information in level-1, in particular for keeping p_T thresholds for muons
  - ➡ requires high-p_T track finding at 40 MHz, latency 4 µsec



- idea: track-stubs for high-p_T candidat
  - ➡ coincidences in electronics of PS and 2S modules
  - ➡ use FPGAs to merge stubs into tracklet seeds, to extend seeds and for Kalman Filter track fit

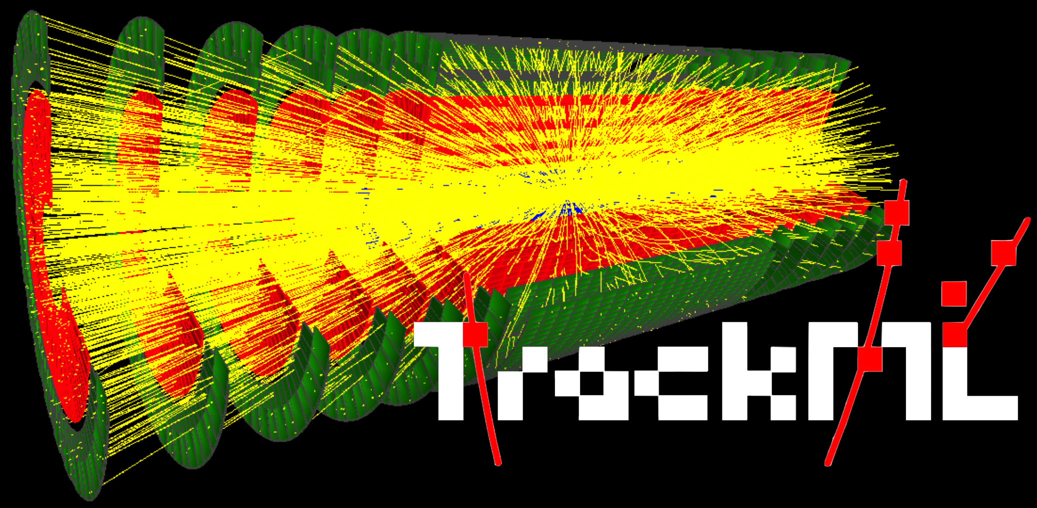


- (HLT) pixel tracking on GPUs - Patatrack
  - ➡ strategy:
    - parallelised cellular automaton for seed finding
    - merge overlapping candidates and apply simple (Riemann) or Broken Line fit
  - ➡ CMS announced to equip the HLT farm with GPUs already for Run-3



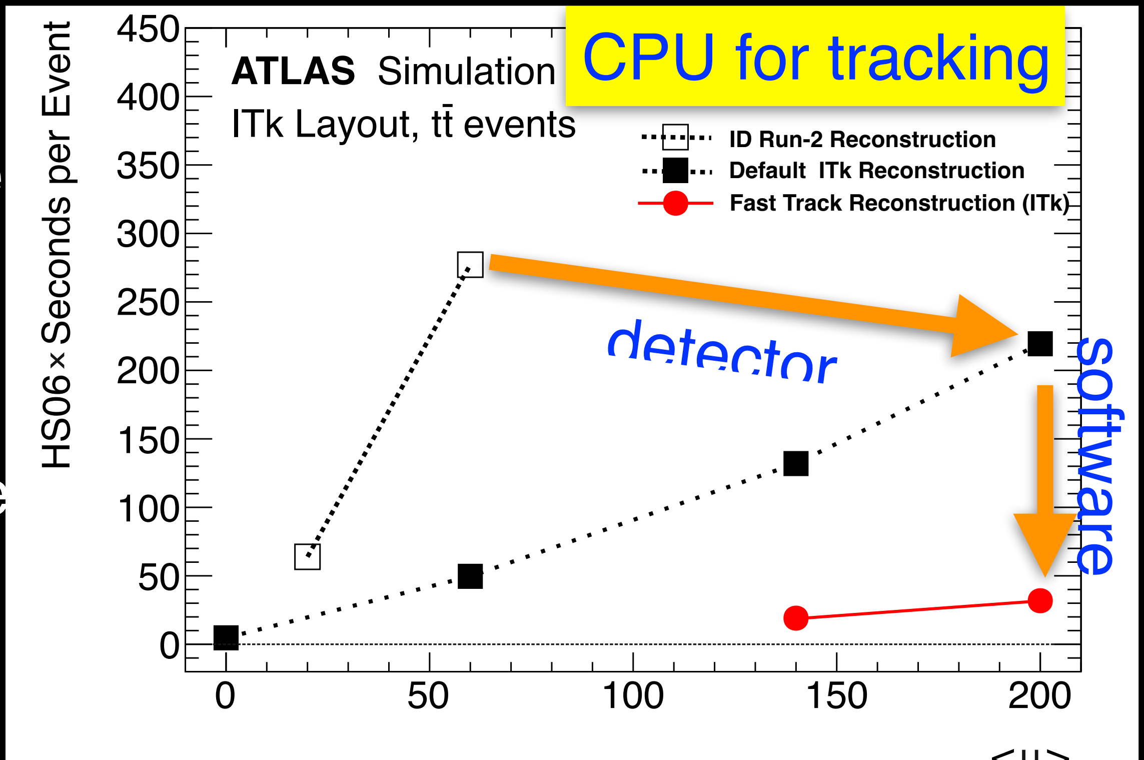
Fast Offline Tracking for Phase-2

- Intensive R&D on tracking software
  - ➡ ACTS as an open source tracking project, "community" project ATLAS/Belle-II/FCC ...
  - ➡ tracking community workshops (CTD)
    - and of course, this Institut Pascal
  - ➡ R&D on support for GPUs and other co-processors (online and offline)



- TrackingML Challenge
  - ➡ reaching out to data science community
  - ➡ open data detector based on ACTS software

- ATLAS also invests in optimising its classical tracking chain
  - ➡ adapt strategy to fully explore new detector
    - seeding in new 5 layer pixel detector
  - ➡ optimise track selection for physics use-case
    - high purity working point
  - ➡ extremely encouraging results !
    - R&D continues to maximise physics



(todays CPUs have 15-20 HS06 per

Discussion...