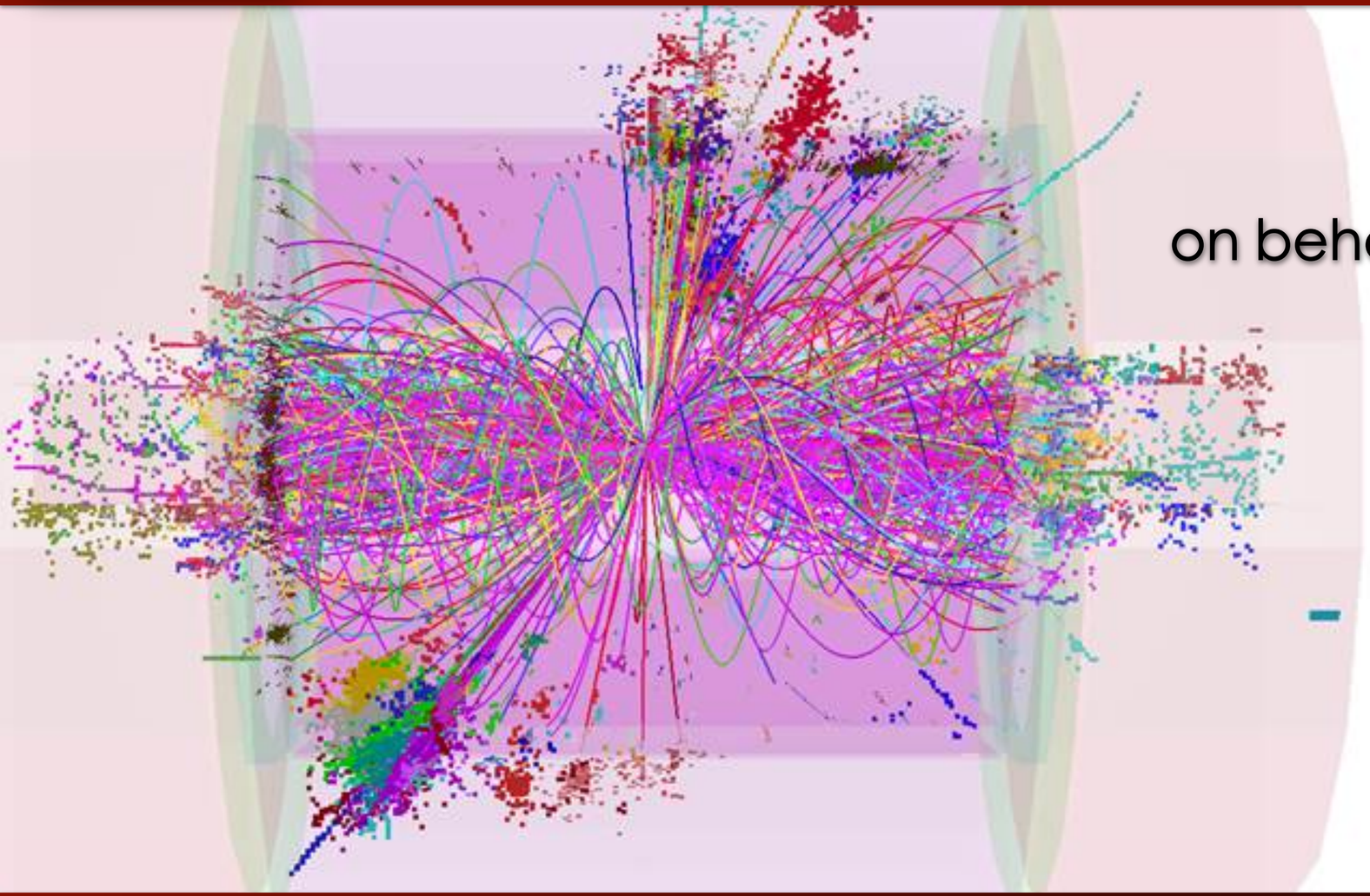
Conformal tracking for the CLIC detector



Emilia Leogrande (CERN),
on behalf of the CLICdp Collaboration

emilia.leogrande@cern.ch

Connecting the Dots 2018
Seattle - 22/03/2018

In this talk

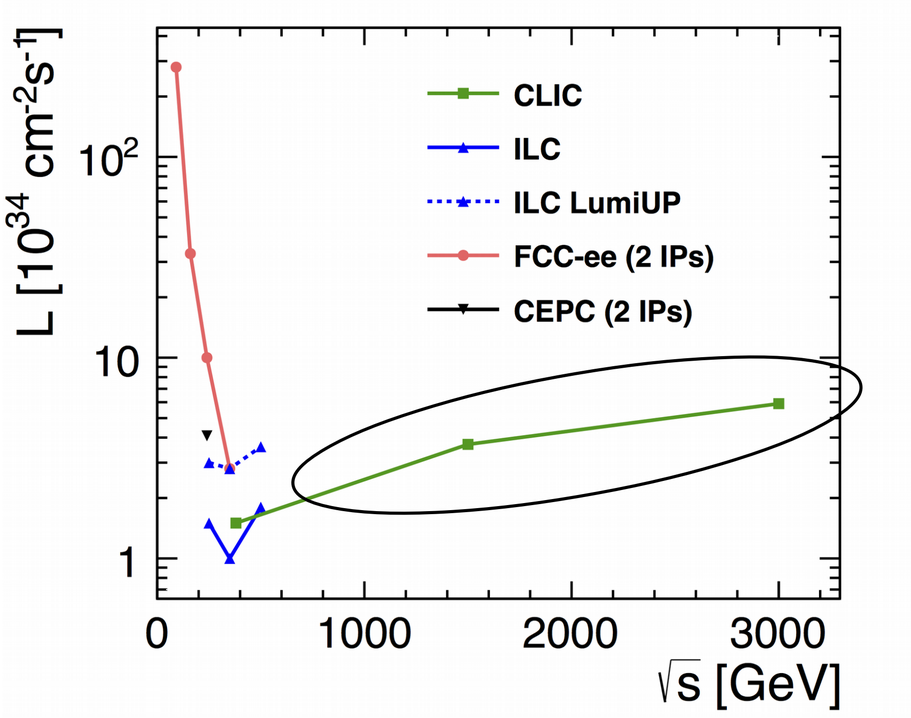
- ☆ Introduction
  - ✦ CLIC, detector requirements from physics and experiment, detector design

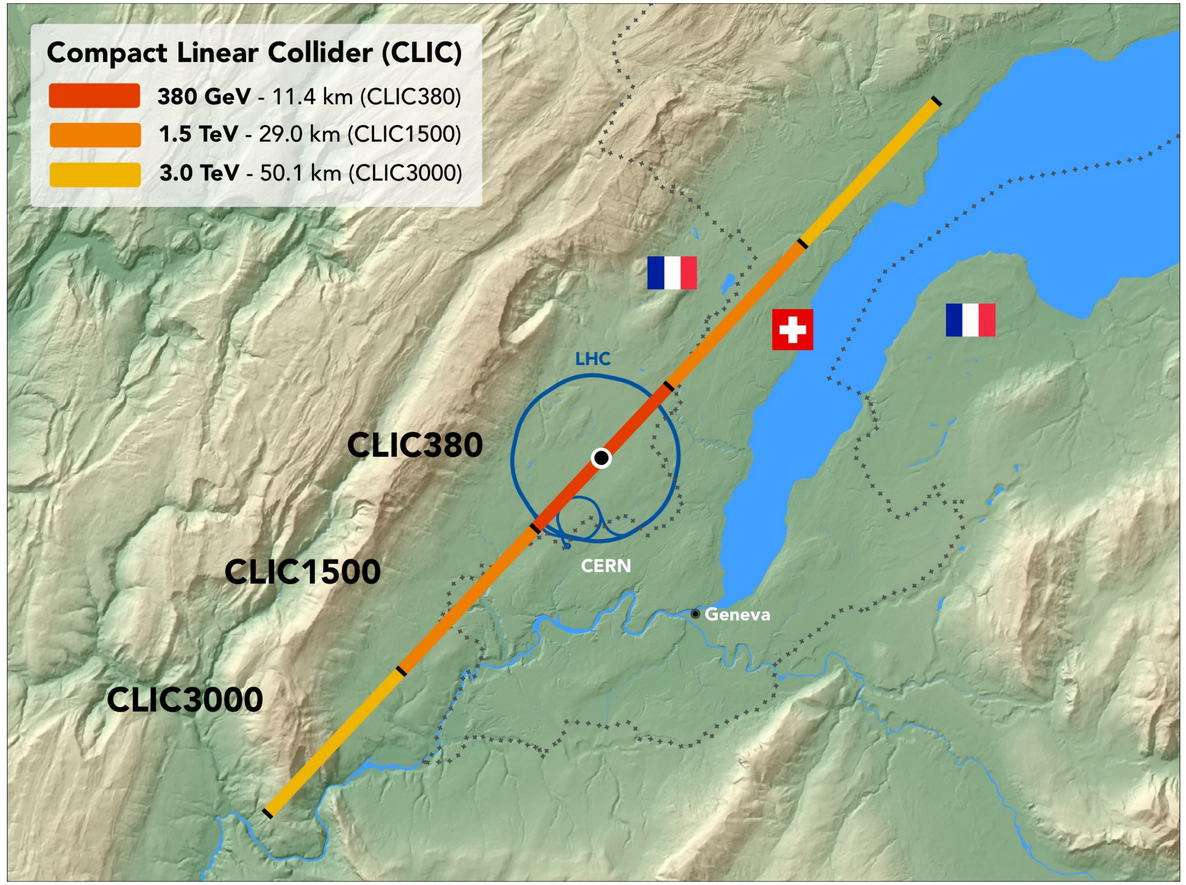
- ☆ Simulation and reconstruction software
  - ✦ detector model in full simulation
  - ✦ from simulated to reconstructed hits
  - ✦ pattern recognition in conformal space
  - ✦ track fit

- ☆ Tracking performances
  - ✦ resolution, efficiencies, fake rates
  - ✦ further improvements and open points

Compact Linear Collider

- High-luminosity linear e^+e^- collider
- Three energy stages up to 3 TeV
  - adaptable to LHC discoveries at 13/14 TeV
- The multi-TeV option for future lepton colliders





Physics goals

- ☆ Standard model physics
  - ☆ Higgs and top precision measurements
  - ✦ cross-section of interesting processes is 8 orders of magnitude lower than total pp cross-section

- ☆ BSM physics
  - ☆ masses, mixing angles, couplings, spins of new sparticles
  - ✦ very rare processes accessible due to low backgrounds (e.g. no QCD)

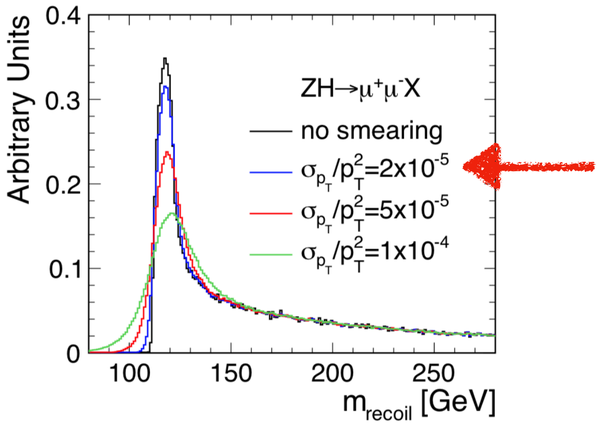
CLIC physics and tracking performance requirements

Physics goals

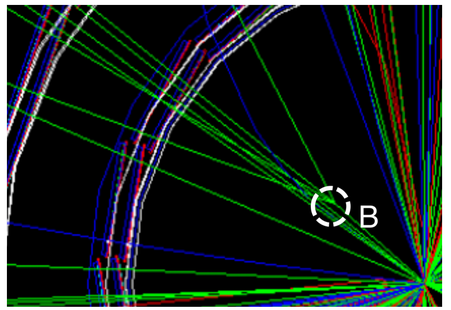
- Standard model physics
  - Higgs and top precision measurements
  - cross-section of interesting processes is 8 orders of magnitude lower than total pp cross-section
- BSM physics
  - masses, mixing angles, couplings, spins of new sparticles
  - very rare processes accessible due to low backgrounds (e.g. no QCD)

=> Tracking performance requirements

- Momentum resolution



- Impact parameter resolution

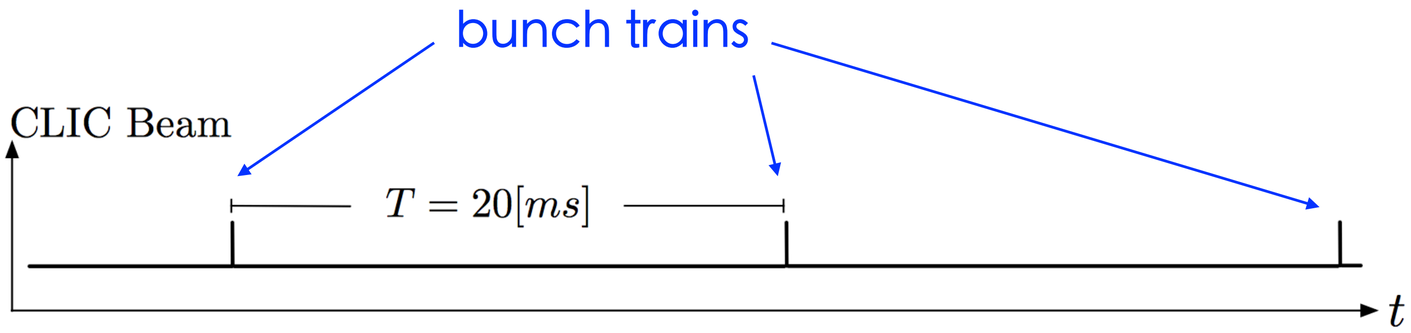


$$\sigma_{d_0}^2 = a^2 + \frac{b^2}{p^2 \sin^3\theta}$$

$a \lesssim 5\ \mu\text{m}$ and $b \lesssim 15\ \mu\text{m GeV}$

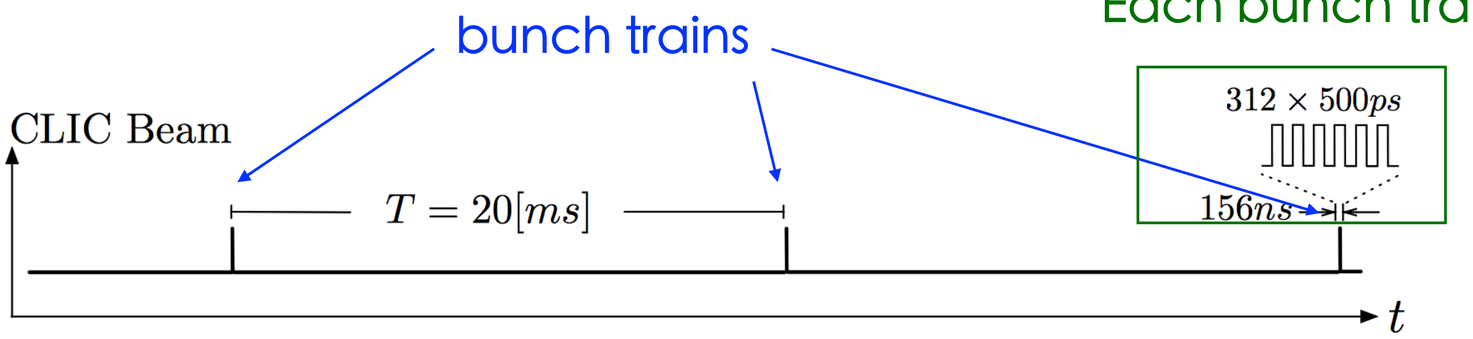
- Best possible angular coverage

CLIC beam structure and beam-induced backgrounds

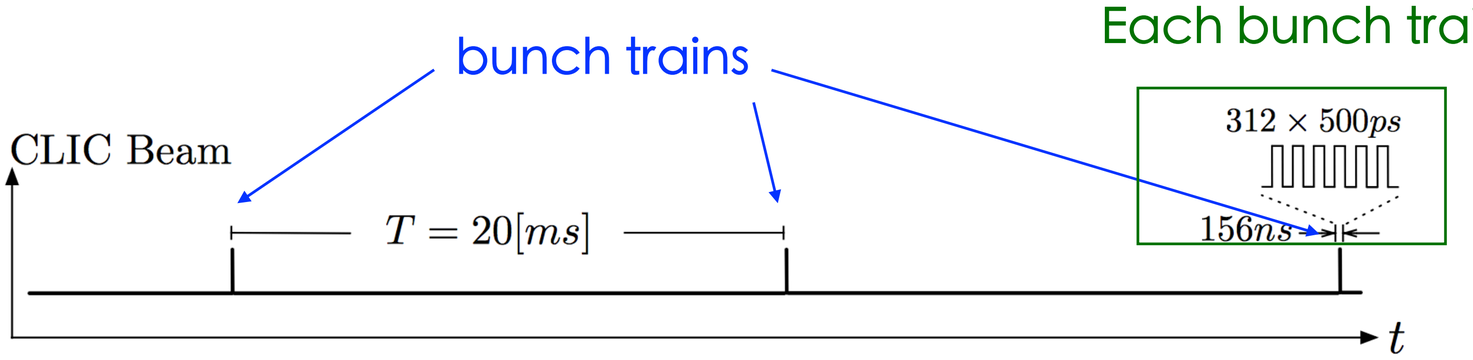


CLIC beam structure and beam-induced backgrounds

Each bunch train = 312 bunches, distant 0.5ns

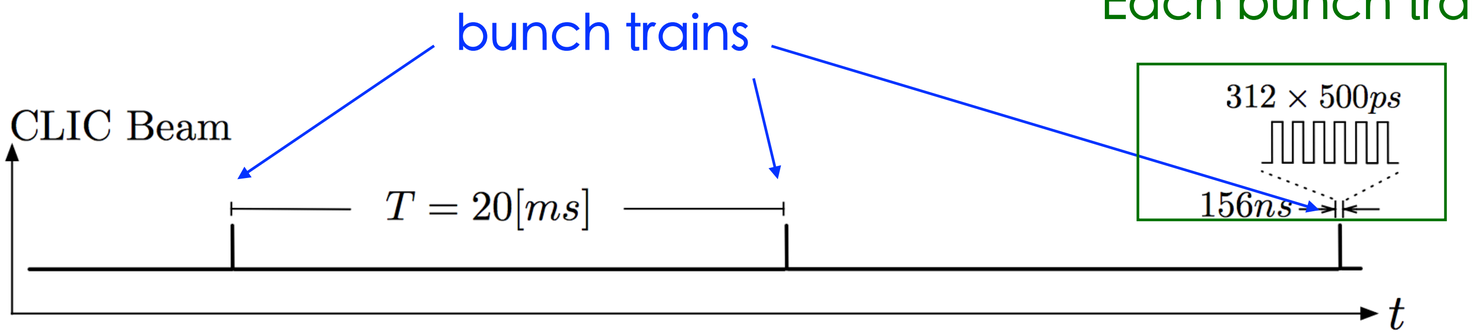


CLIC beam structure and beam-induced backgrounds



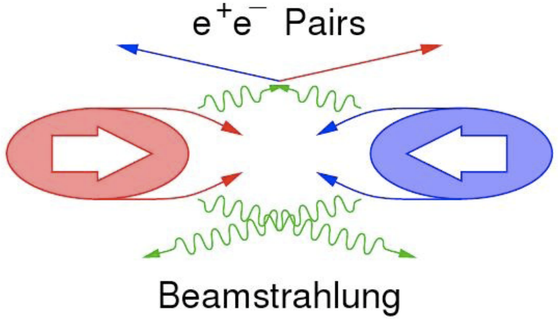
- entire bunch train available for offline reconstruction
- not all bunches contain a "hard" interaction

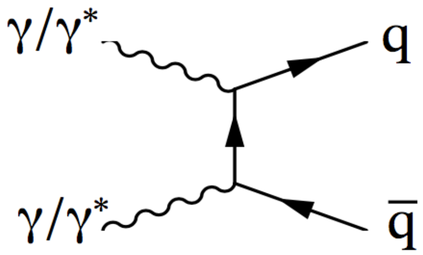
CLIC beam structure and beam-induced backgrounds



- entire bunch train available for offline reconstruction
- not all bunches contain a "hard" interaction

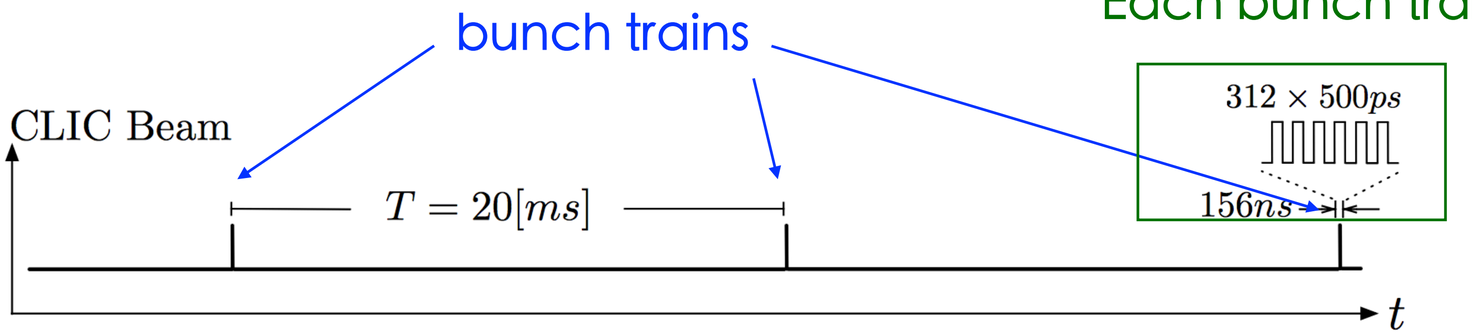
Two main sources of background





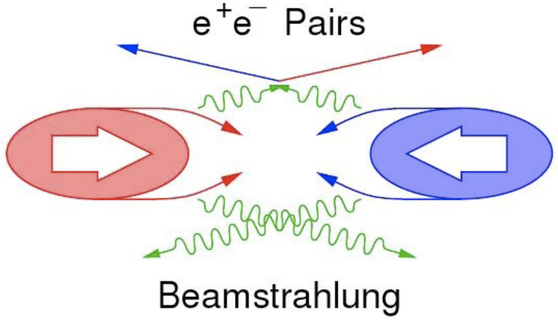
- 19k particles/bunch train at 3TeV
- 17k particles/bunch train at 3TeV

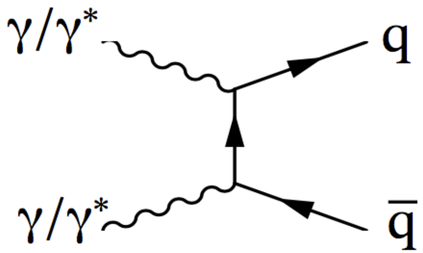
CLIC beam structure and beam-induced backgrounds



- entire bunch train available for offline reconstruction
- not all bunches contain a "hard" interaction

Two main sources of background



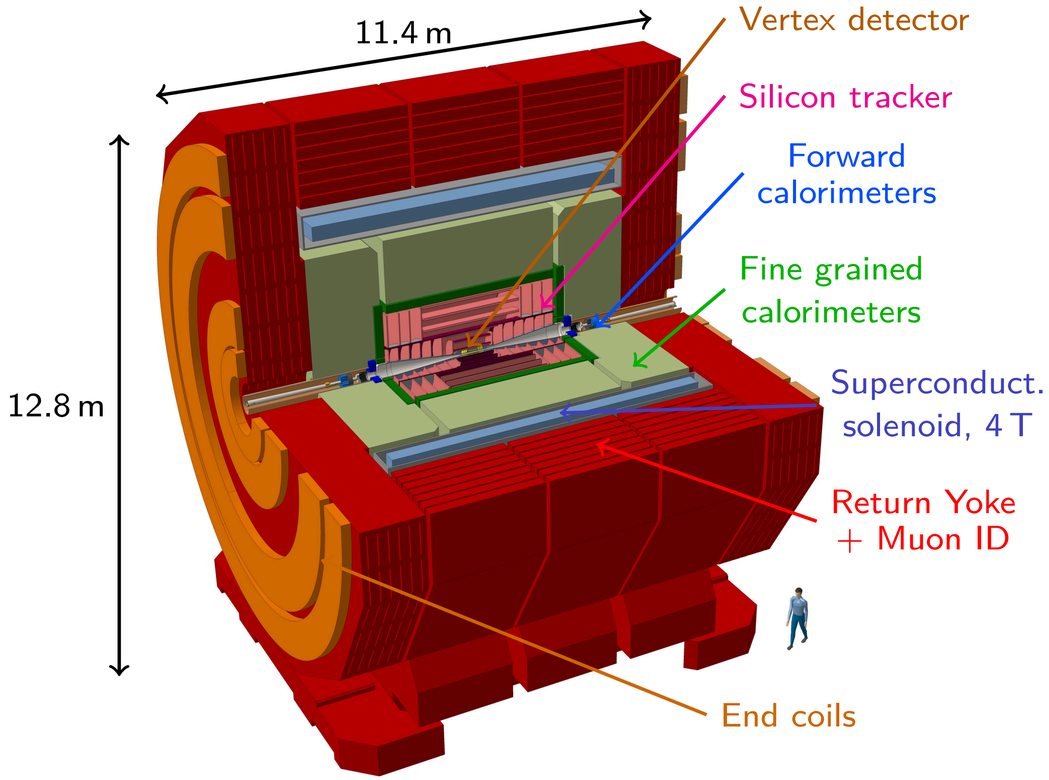


- 19k particles/bunch train at 3TeV
- 17k particles/bunch train at 3TeV

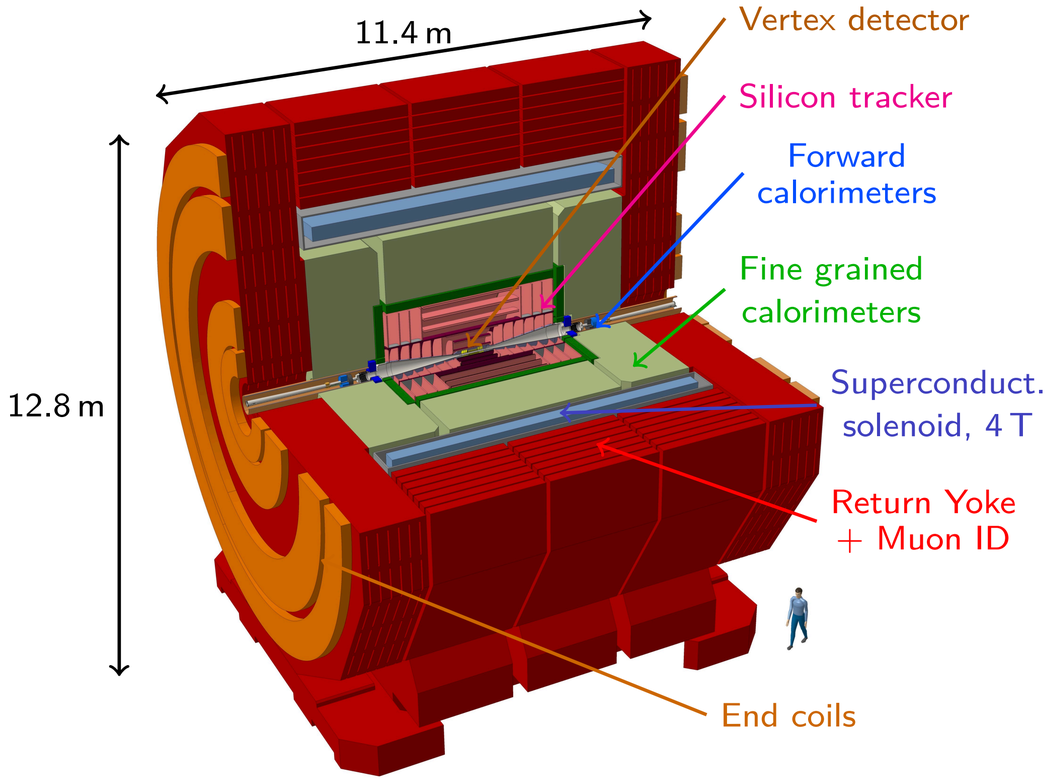
=> Tracking timing requirements

- Time stamp hits from the detectors (central det <=> physics event)
- Impose timing cuts
  - 15ns integration time (30BX)
  - 15ns/√2 hit resolution

The CLIC detector - focus on the tracking system

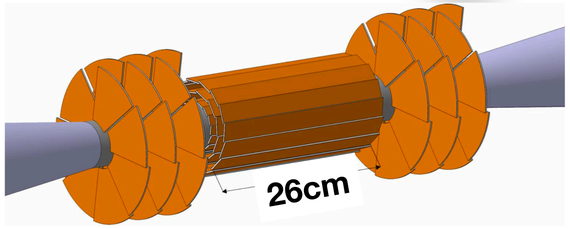


The CLIC detector - focus on the tracking system

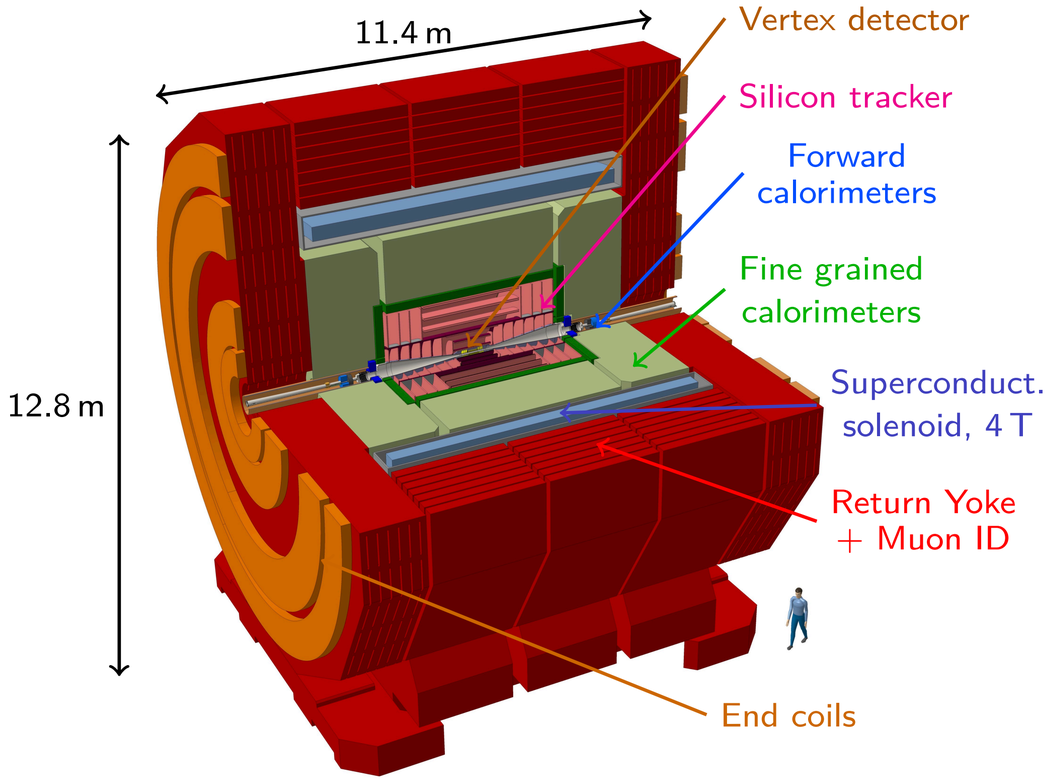


Vertex detector

- ☆ Silicon pixels 25x25 μm^2
- ☆ single point resolution = 3 μm
- ☆ material budget: 0.2%X0 per layer

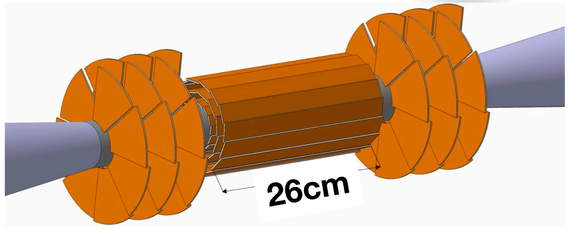


The CLIC detector - focus on the tracking system



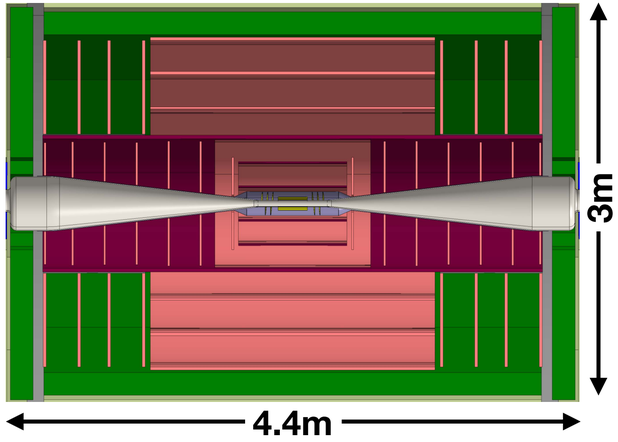
Vertex detector

- ☆ Silicon pixels 25x25 μm^2
- ☆ single point resolution = 3 μm
- ☆ material budget: 0.2%X0 per layer

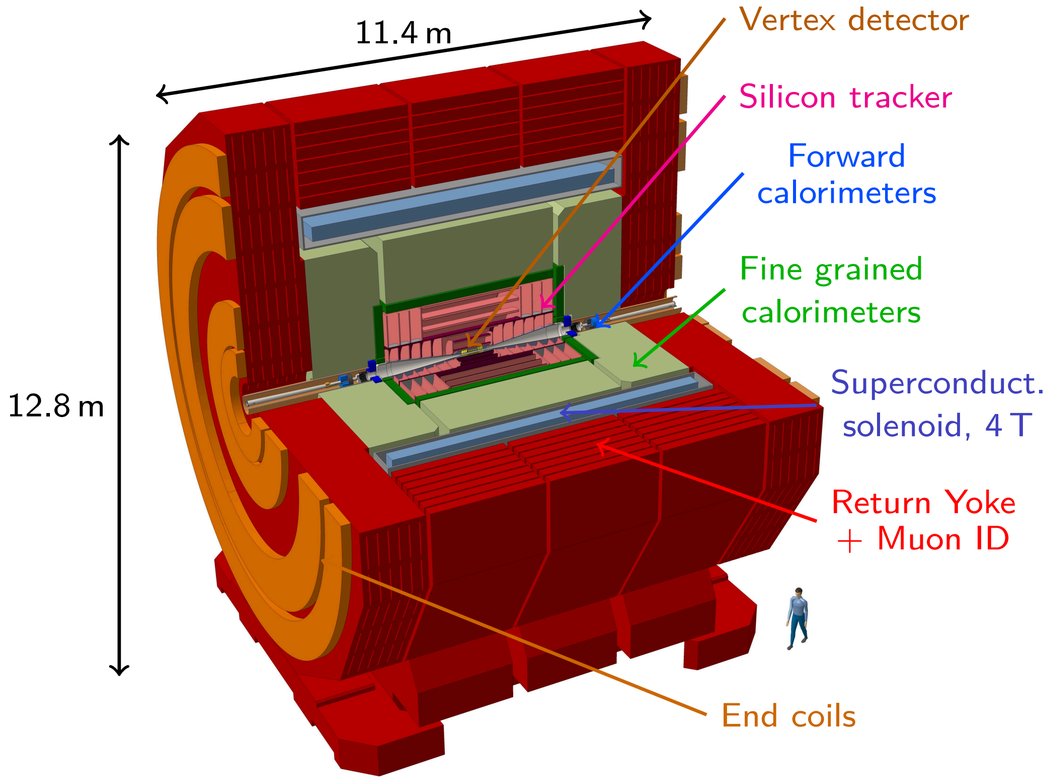


Tracker detector

- ☆ Silicon microstrips
- ☆ single point resolution = 7 x 90 μm^2
- ☆ material budget:
  - ✦ detector: ~1%X_0 per layer
  - ✦ support&cables: ~2.5%X_0

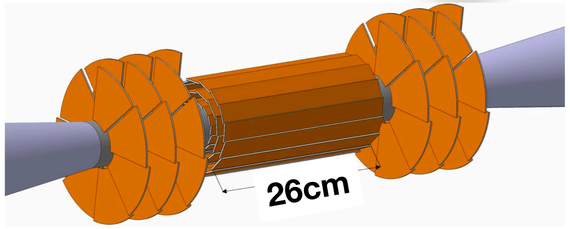


The CLIC detector - focus on the tracking system



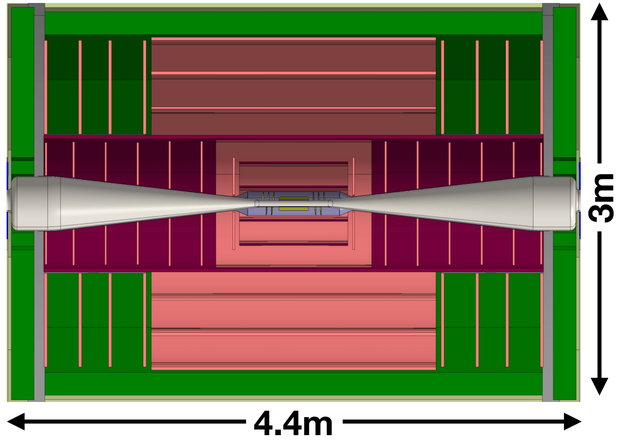
Vertex detector

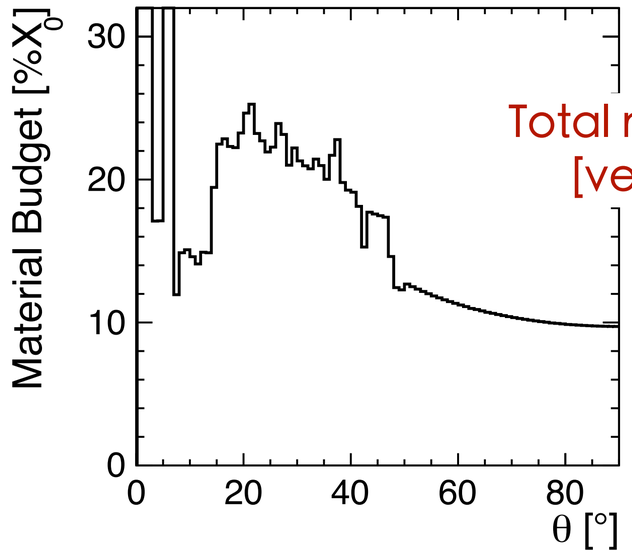
- ☆ Silicon pixels 25x25 μm²
- ☆ single point resolution = 3 μm
- ☆ material budget: 0.2%X0 per layer



Tracker detector

- ☆ Silicon microstrips
- ☆ single point resolution = 7 x 90 μm²
- ☆ material budget:
  - ✦ detector: ~1%X_0 per layer
  - ✦ support&cables: ~2.5%X_0





Total material budget [vertex+tracker]

Low-mass detector perfectly suited for a conformal-mapping based pattern recognition

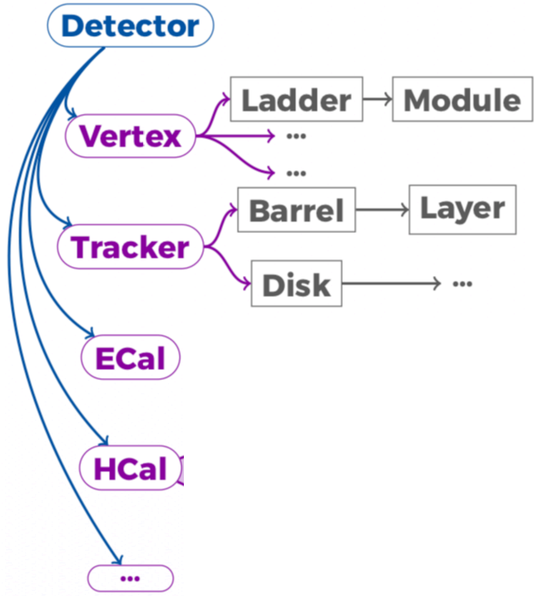
Detector implementation in full simulation

Single source of geometry information for simulation, reconstruction and analysis: DD4hep

Detector implementation in full simulation

Single source of geometry information for simulation, reconstruction and analysis: DD4hep

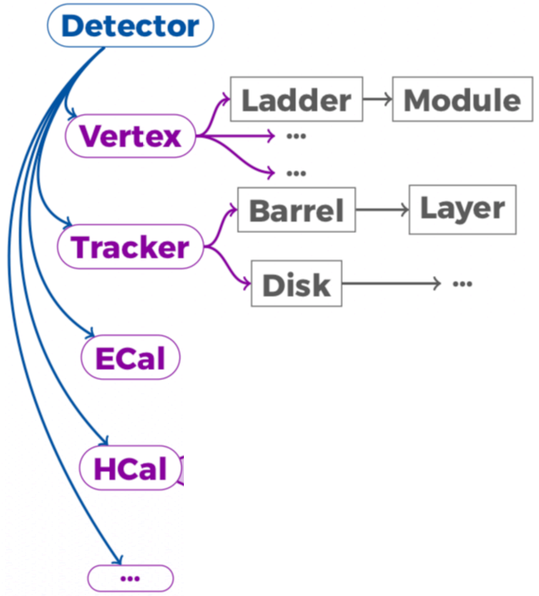
- ☆ Detector description = tree-like hierarchy of DetElements
  - ✦ detector constructors (drivers) [C++]
  - ✦ compact description [xml]



Detector implementation in full simulation

Single source of geometry information for simulation, reconstruction and analysis: DD4hep

- Detector description = tree-like hierarchy of DetElements
  - detector constructors (drivers) [C++]
  - compact description [xml]



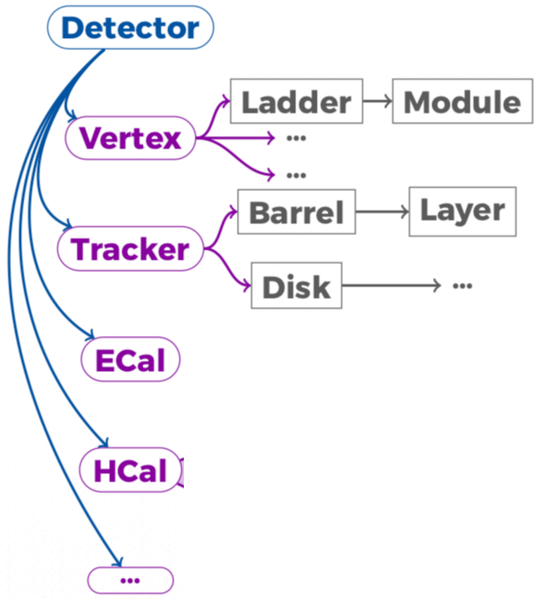
- Reconstruction extension = high-level view of detector
  - DDRec (subpackage of DD4hep) attaches Data Structures to DetElements
  - Data Structures contain physical properties (layers, cell sizes, point resolutions,..)

```
//Adding data structure to a DetElement
DetElement caloDet("myCalo", detID);
DDRec::LayeredCalorimeterData* caloData = new DDRec::LayeredCalorimeterData;
// create detector, fill the caloData
caloDet.addExtension< DDRec::LayeredCalorimeterData >( caloData );
```

Detector implementation in full simulation

Single source of geometry information for simulation, reconstruction and analysis: DD4hep

- Detector description = tree-like hierarchy of DetElements
  - detector constructors (drivers) [C++]
  - compact description [xml]



- Reconstruction extension = high-level view of detector
  - DDRec (subpackage of DD4hep) attaches Data Structures to DetElements
  - Data Structures contain physical properties (layers, cell sizes, point resolutions,..)

```
//Adding data structure to a DetElement
DetElement caloDet("myCalo", detID);
DDRec::LayeredCalorimeterData* caloData = new DDRec::LayeredCalorimeterData;
// create detector, fill the caloData
caloDet.addExtension< DDRec::LayeredCalorimeterData >( caloData );
```

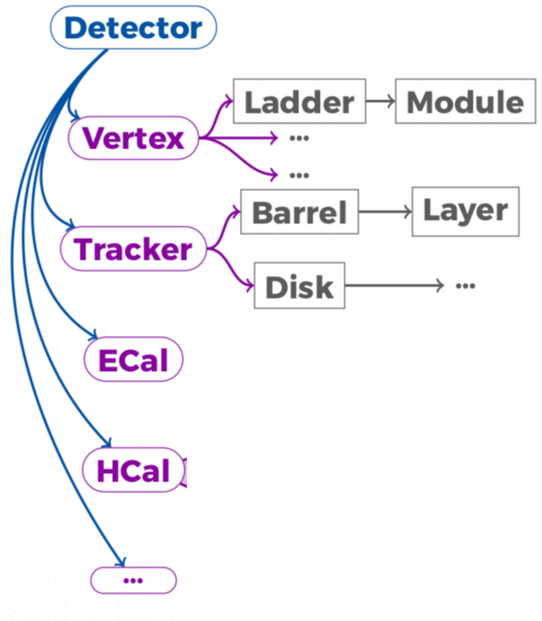
- Surface extension = condensed info for reconstruction
  - measurement directions of hits, local-to-global, material
  - placed in the middle of sensitive material
  - properties from averaging total material

sensitive
support 1
support 2
support 3

Detector implementation in full simulation

Single source of geometry information for simulation, reconstruction and analysis: DD4hep

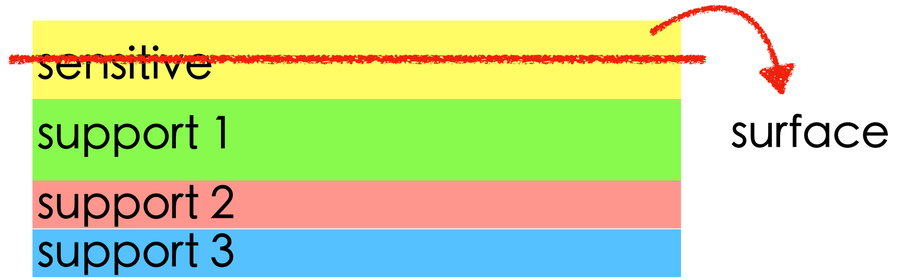
- Detector description = tree-like hierarchy of DetElements
  - detector constructors (drivers) [C++]
  - compact description [xml]



- Reconstruction extension = high-level view of detector
  - DDRec (subpackage of DD4hep) attaches Data Structures to DetElements
  - Data Structures contain physical properties (layers, cell sizes, point resolutions,..)

```
//Adding data structure to a DetElement
DetElement caloDet("myCalo", detID);
DDRec::LayeredCalorimeterData* caloData = new DDRec::LayeredCalorimeterData;
// create detector, fill the caloData
caloDet.addExtension< DDRec::LayeredCalorimeterData >( caloData );
```

- Surface extension = condensed info for reconstruction
  - measurement directions of hits, local-to-global, material
  - placed in the middle of sensitive material
  - properties from averaging total material



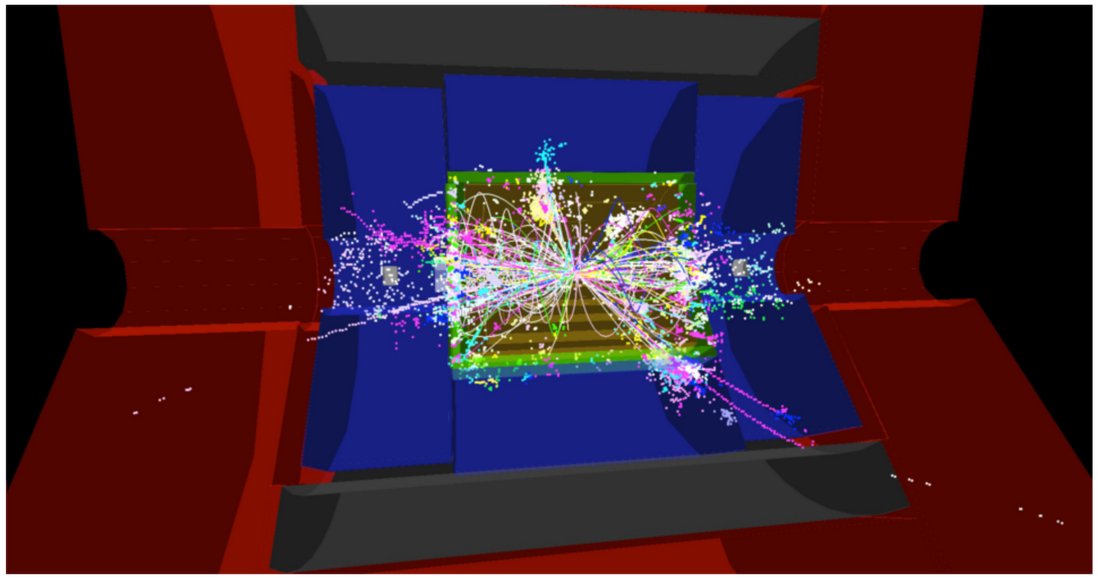
From simulated hits to reconstructed hits

Simulation with DDSim

- geometry, magnetic field, sensitive detectors
- MonteCarlo events or particle guns
- physics list (Geant4)
- MC-truth linking (hits and particles that produced them)

Overlay background

- simulated physics events + $\gamma\gamma$->hadrons
  - timing cuts to reduce background



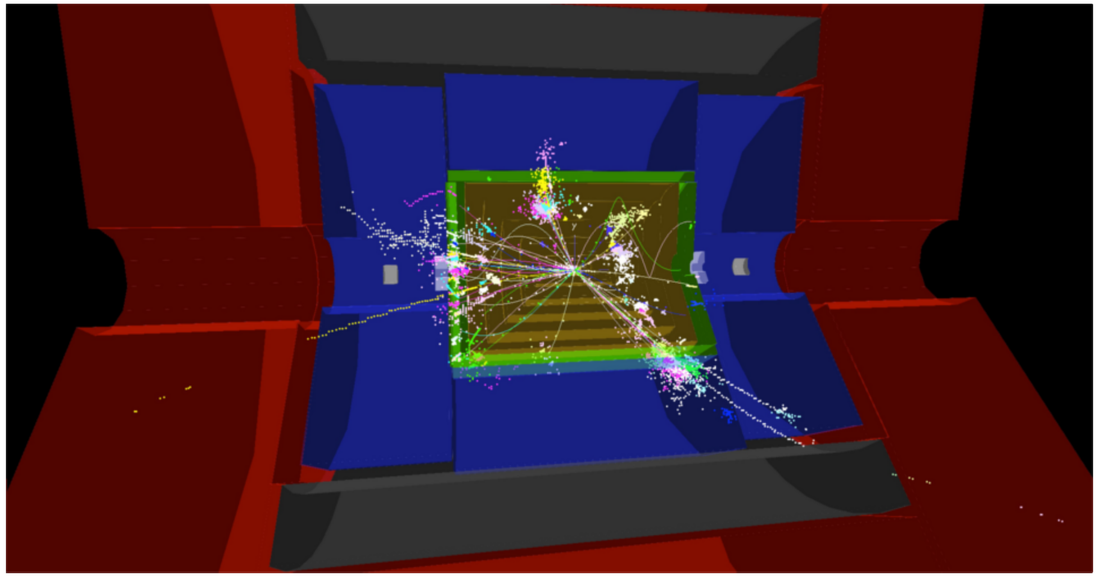
From simulated hits to reconstructed hits

Simulation with DDSim

- geometry, magnetic field, sensitive detectors
- MonteCarlo events or particle guns
- physics list (Geant4)
- MC-truth linking (hits and particles that produced them)

Overlay background

- simulated physics events + $\gamma\gamma$->hadrons
  - timing cuts to reduce background



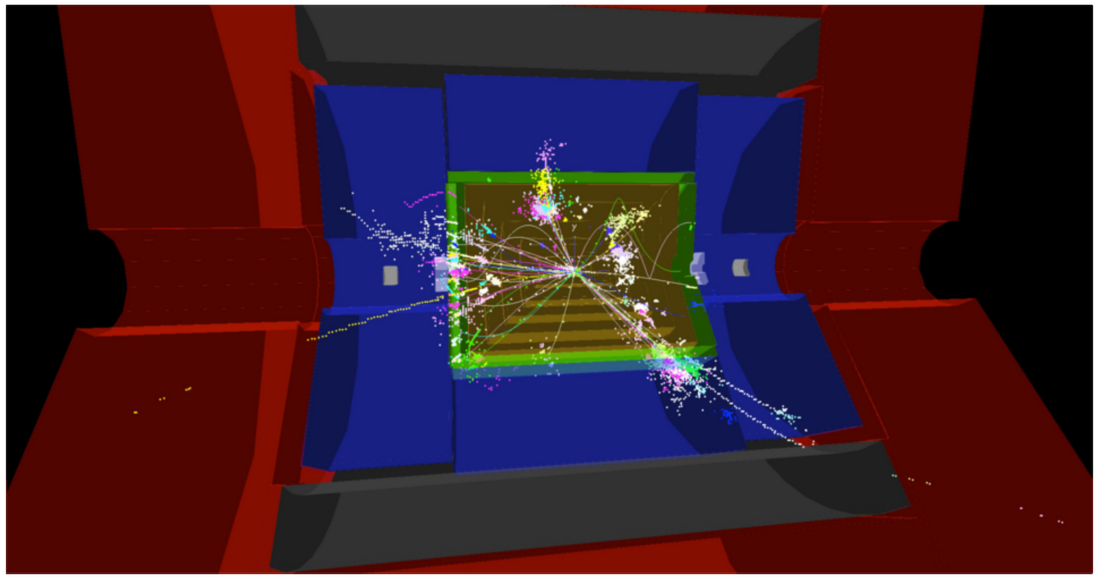
From simulated hits to reconstructed hits

Simulation with DDSim

- geometry, magnetic field, sensitive detectors
- MonteCarlo events or particle guns
- physics list (Geant4)
- MC-truth linking (hits and particles that produced them)

Overlay background

- simulated physics events + γγ->hadrons
  - timing cuts to reduce background



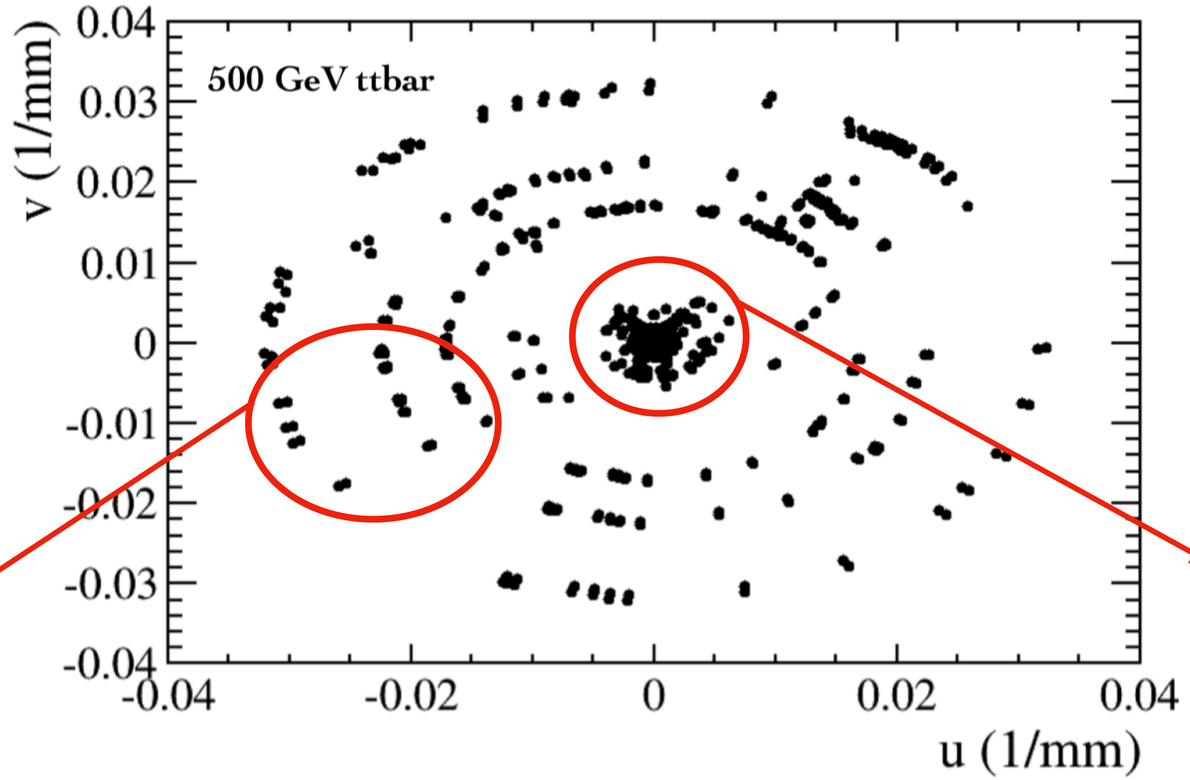
Digitization

- To simulate the effect of pixels (strips), the simulated hits are smeared with a Gaussian distribution
  - σ = single point resolution of the sub detector
  - smearing in local coordinate system

Tracks in conformal space

- Conformal mapping applies a geometry transform that maps circles in the x,y plane passing through the origin into straight lines in the u,v plane

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



hits from the vertex

hits from the tracker

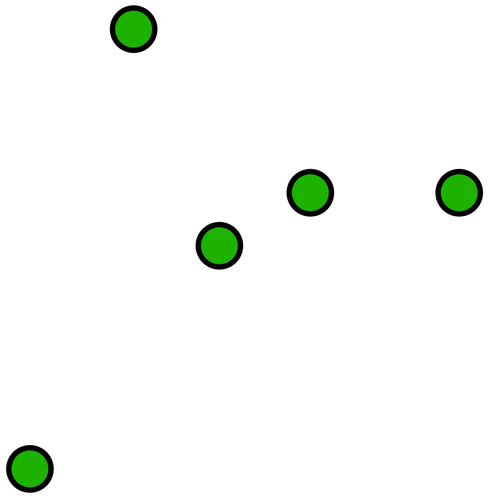
- Pattern recognition in conformal space via cellular automaton, used to perform straight line search

Pattern recognition - full chain

Build tracks	Vertex barrel	Standard cuts
Extend tracks	Vertex endcap	Standard cuts
Build tracks	Vertex b + e	Standard cuts
Build tracks	Vertex b + e	Looser cuts (angle x 5)
Build tracks	Vertex b + e	Looser cuts (angle x 10)
Build tracks	Vertex b + e	Looser cuts (angle x 10; χ^2 x 20)
Extend tracks	Tracker collections	Looser cuts (angle x 10; χ^2 x 20)
Build tracks	All collections	Displaced cuts

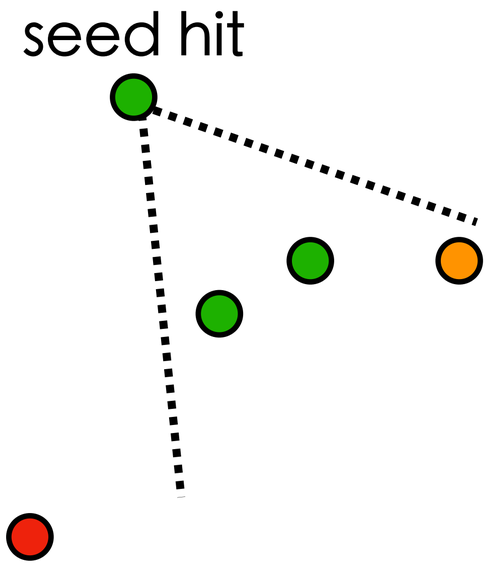
- This pattern recognition is geometry-agnostic:
  - based on the hits position in global space
  - no constrain on the hit position in the subdetectors

Pattern recognition - build tracks in vertex barrel



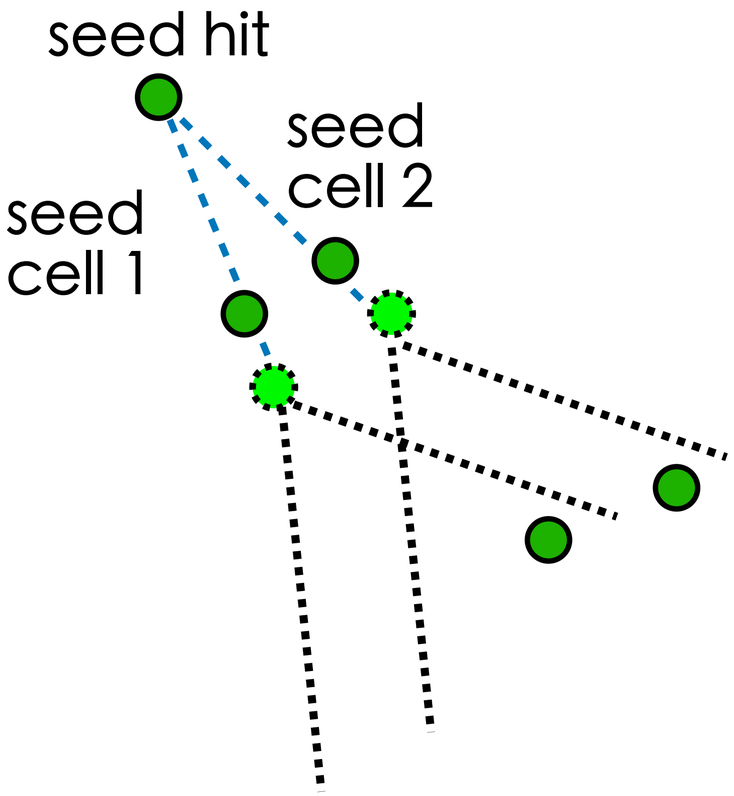
- Search for a pattern starts in the vertex barrel hits

Pattern recognition - build tracks in vertex barrel



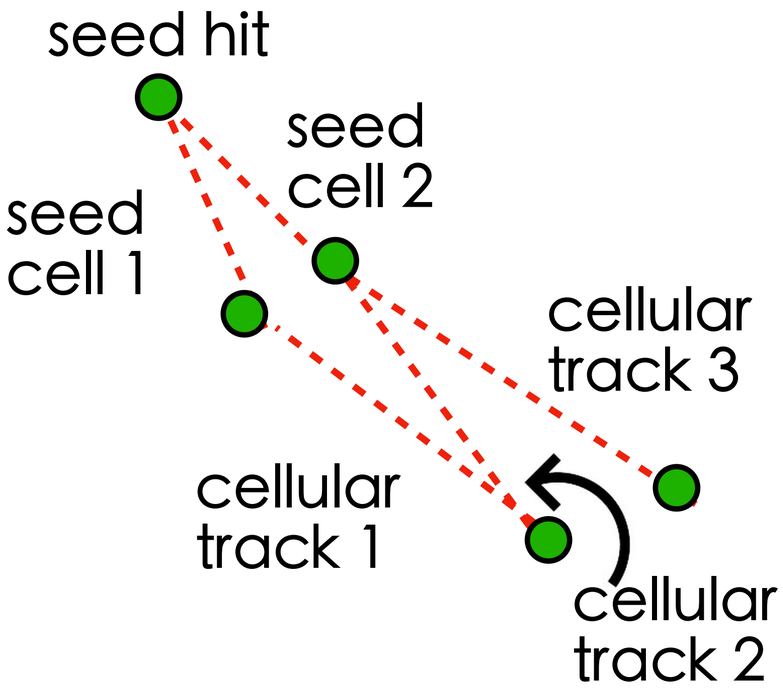
- Search for a pattern starts in the vertex barrel hits
- Each hit is used as seed to look for neighbors, i.e.
  - hits not outside the search cone
  - hits not too far from the seed hit

Pattern recognition - build tracks in vertex barrel



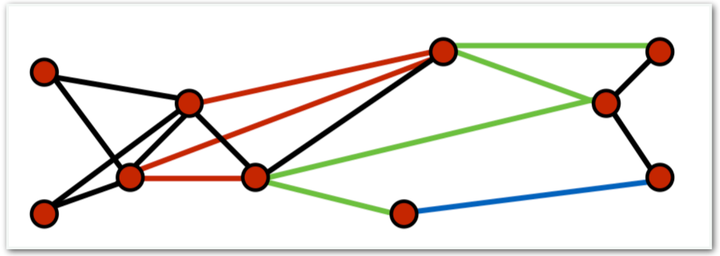
- Search for a pattern starts in the vertex barrel hits
- Each hit is used as seed to look for neighbors, i.e.
  - hits not outside the search cone
  - hits not too far from the seed hit
- Seed cells are created and extended to fake hits from which the look for neighbors is repeated

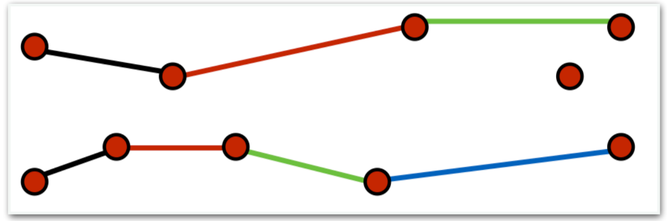
Pattern recognition - build tracks in vertex barrel



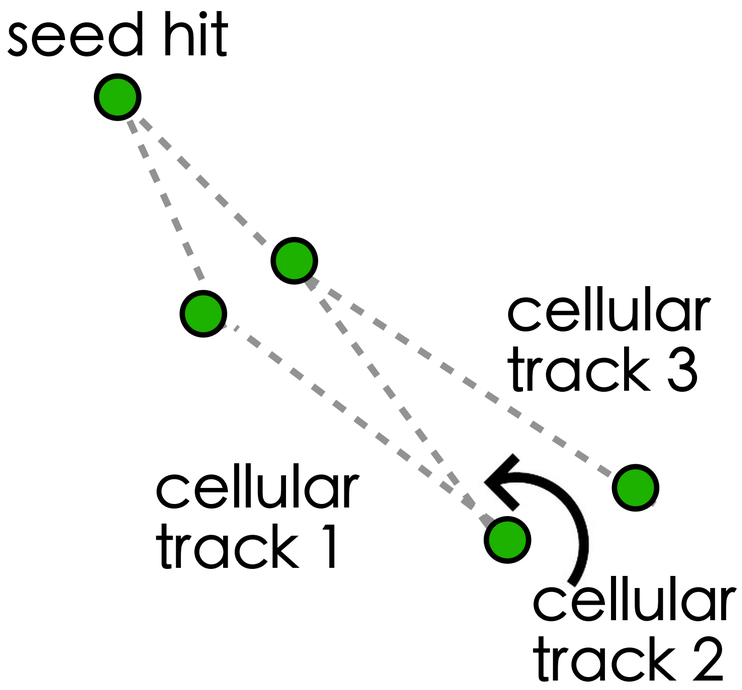
- Search for a pattern starts in the vertex barrel hits
- Each hit is used as seed to look for neighbors, i.e.
  - hits not outside the search cone
  - hits not too far from the seed hit
- Seed cells are created and extended to fake hits from which the look for neighbors is repeated
- Cellular tracks are vectors of cells
  - each cell includes a weight
  - each subsequent link increments the weight by 1
  - create valid tracks starting from higher weight back to the seed cell
    - if more paths available, branch the track

w = 0
w = 1
w = 2
w = 3



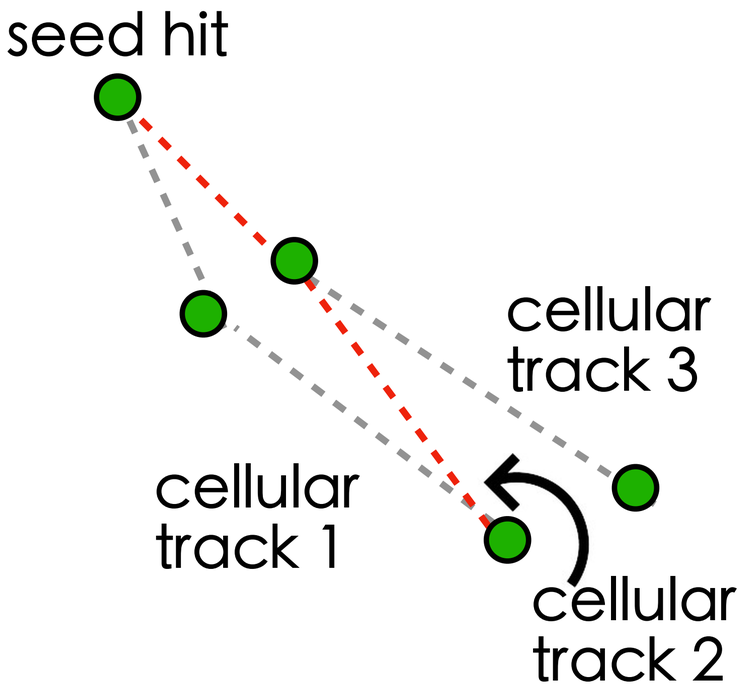


Pattern recognition - build tracks in vertex barrel



- For all cellular tracks stemming from the seed hit:
  - linear regression in (u,v) —> χ^2/ndf
  - linear regression in (s,z) —> χ^2_{sz}/ndf
    - s: arc segment in the transverse plane
  - hits progressively removed one by one to refit and compare χ^2

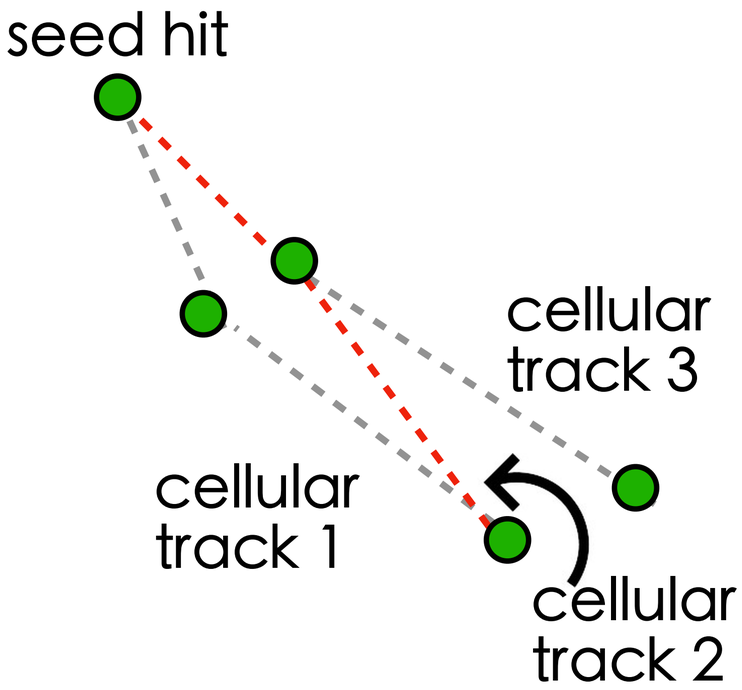
Pattern recognition - build tracks in vertex barrel



- For all cellular tracks stemming from the seed hit:
  - linear regression in (u,v) —> χ^2/ndf
  - linear regression in (s,z) —> χ^2_{sz}/ndf
    - s: arc segment in the transverse plane
  - hits progressively removed one by one to refit and compare χ^2

- Best track is the one with lowest χ^2
  - similar χ^2 are kept
  - clones (overlapping hits >= 2) are skimmed
    - longest usually preferred if χ^2 not too large

Pattern recognition - build tracks in vertex barrel

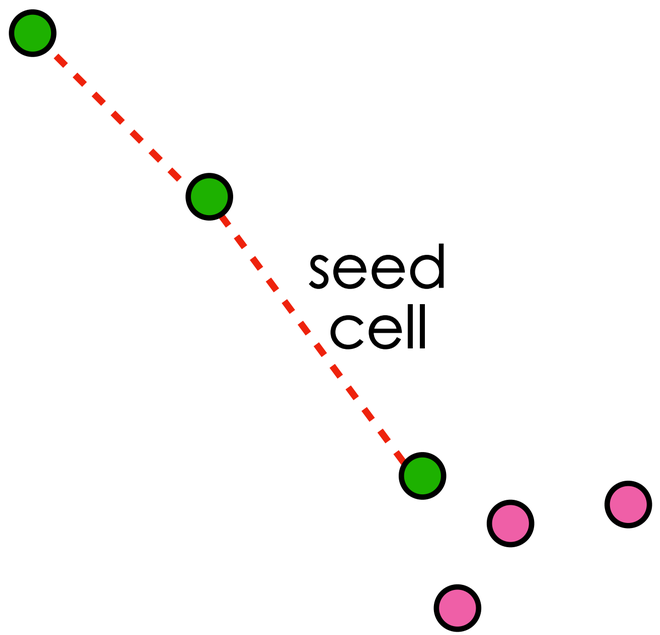


- For all cellular tracks stemming from the seed hit:
  - linear regression in (u,v) —> χ^2/ndf
  - linear regression in (s,z) —> χ^2_{sz}/ndf
    - s: arc segment in the transverse plane
  - hits progressively removed one by one to refit and compare χ^2

- Best track is the one with lowest χ^2
  - similar χ^2 are kept
  - clones (overlapping hits >= 2) are skimmed
    - longest usually preferred if χ^2 not too large

- Hits are marked as used and the search continues with the unused ones in the collection

Pattern recognition - extend tracks in vertex endcaps



- Tracks made in the vertex barrel are extended in the vertex endcaps, starting from the seed cell between the last two hits of the track
- Hit candidates with which to extend the track are picked based on the nearest neighbors search
- Each hit candidate is progressively added to the track. The track is refitted and the hit is accepted/ rejected based on a χ^2 cut

Pattern recognition - build tracks with leftover vertex hits and extend in tracker

- Once all possible extensions are made, try to build new tracks with leftover hits from combined collection vertex barrel + endcap
  - standard cuts
  - looser angle cuts
  - looser χ^2 cuts

Pattern recognition - build tracks with leftover vertex hits and extend in tracker

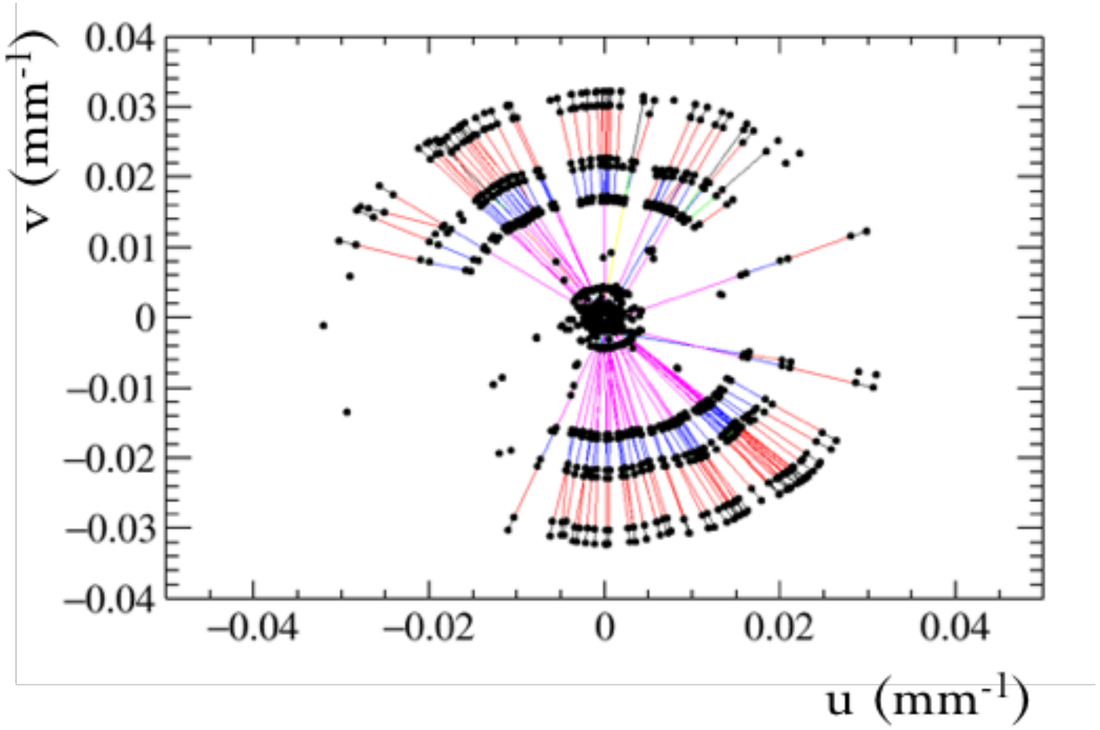
- Once all possible extensions are made, try to build new tracks with leftover hits from combined collection vertex barrel + endcap
  - standard cuts
  - looser angle cuts
  - looser χ^2 cuts

- Extend all tracks made in the vertex (barrel, endcap, combined) with all the tracker collections

Pattern recognition - build tracks with leftover vertex hits and extend in tracker

- Once all possible extensions are made, try to build new tracks with leftover hits from combined collection vertex barrel + endcap
  - standard cuts
  - looser angle cuts
  - looser χ^2 cuts

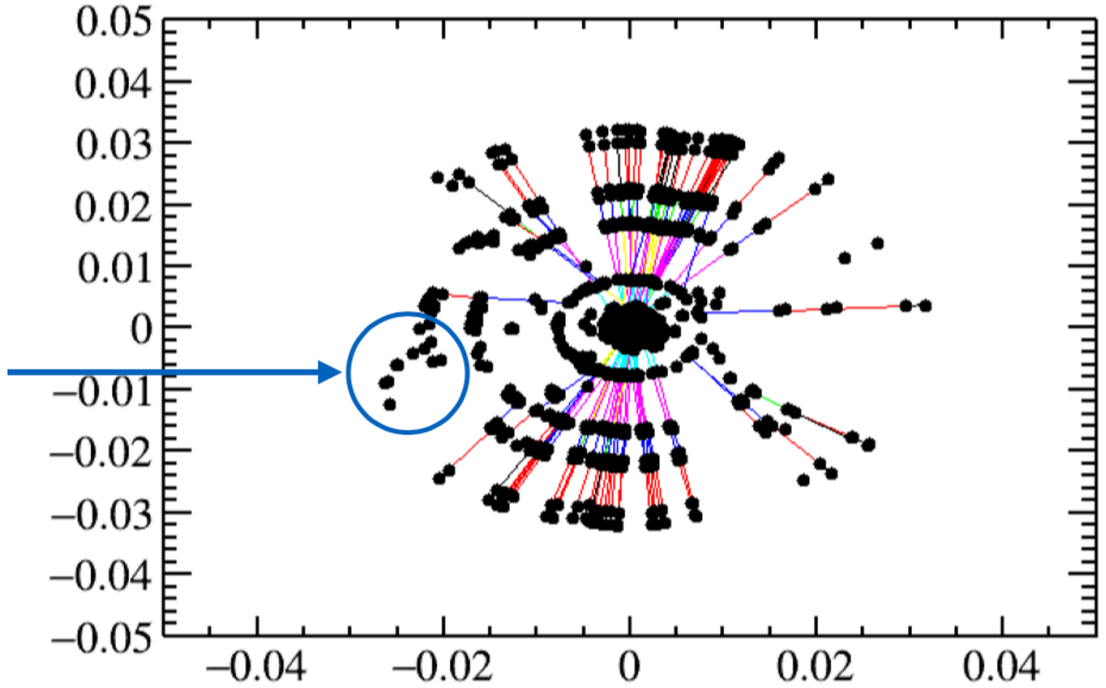
- Extend all tracks made in the vertex (barrel, endcap, combined) with all the tracker collections



Min number of hits to make a track: 4

Last step of the chain: displaced tracks in conformal space

- Conformal mapping turns circles through the origin in (x,y) into straight lines in (u,v)
  => quadratic terms to include displaced tracks, but eventually χ^2 breaks down



- Major strategy change
  - broader search angle than for prompt tracks
  - min number of hits: 5
  - inverted order search: from tracker to vertex hits

Pattern recognition - timing performance

Build tracks	Vertex barrel	Standard cuts	A
Extend tracks	Vertex endcap	Standard cuts	B
Build tracks	Vertex b + e	Standard cuts	C
Build tracks	Vertex b + e	Looser cuts (angle x 5)	D
Build tracks	Vertex b + e	Looser cuts (angle x 10)	E
Build tracks	Vertex b + e	Looser cuts (angle x 10; chi^2 x 20)	F
Extend tracks	Tracker collections	Looser cuts (angle x 10; chi^2 x 20)	G
Build tracks	All collections	Displaced cuts	H

	1 ttbar event @3TeV (66 tracks)
A	1.15s
B	0.0007s
C	0.07s
D	0.28s
E	0.21s
F	0.20s
G	0.05s
H	10s

- ☆ Too many tracks ($5x10^5$) exception allows to skip events for which the combinatorics would explode

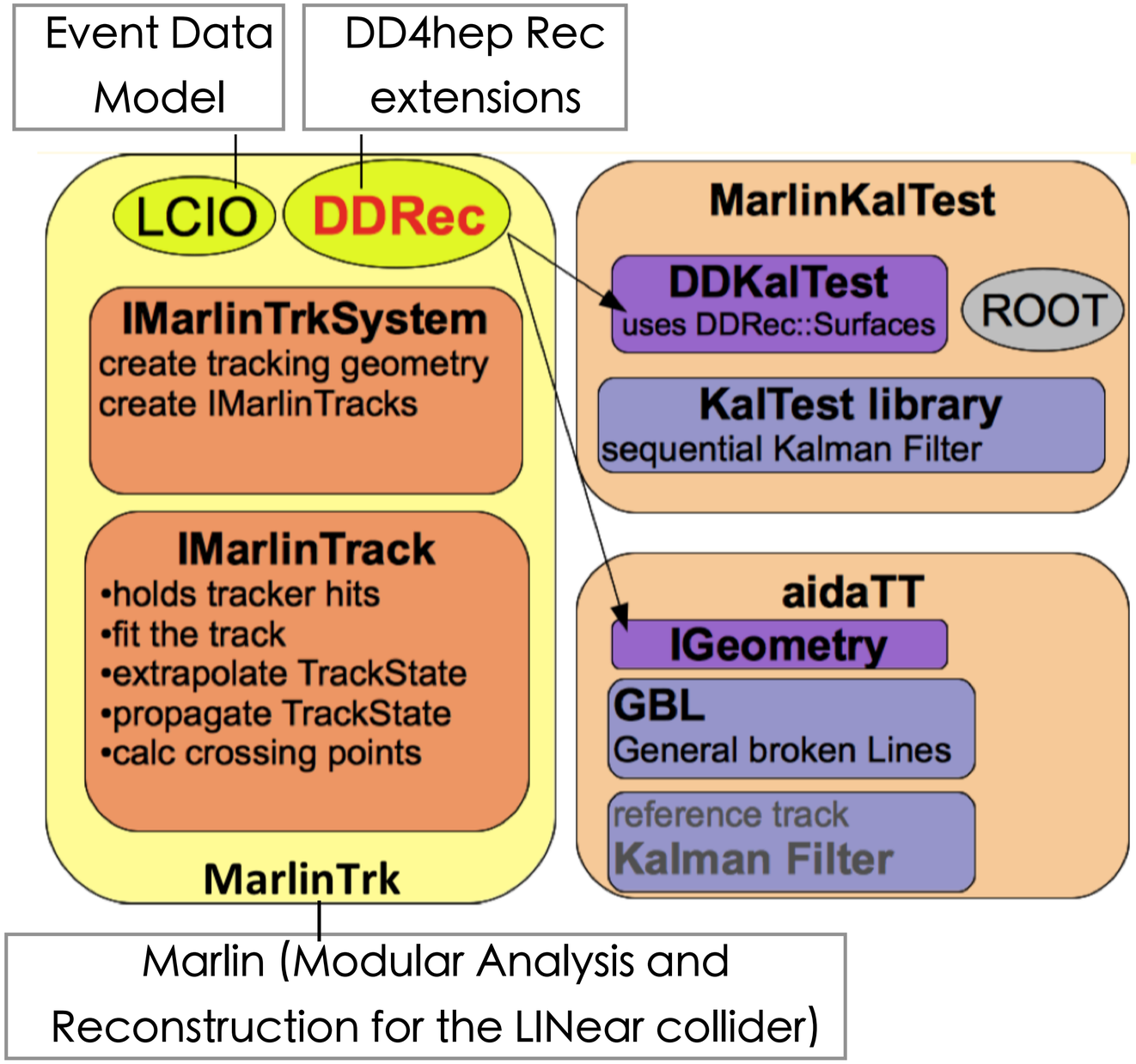
Pattern recognition - timing performance

Build tracks	Vertex barrel	Standard cuts	A
Extend tracks	Vertex endcap	Standard cuts	B
Build tracks	Vertex b + e	Standard cuts	C
Build tracks	Vertex b + e	Looser cuts (angle x 5)	D
Build tracks	Vertex b + e	Looser cuts (angle x 10)	E
Build tracks	Vertex b + e	Looser cuts (angle x 10; chi^2 x 20)	F
Extend tracks	Tracker collections	Looser cuts (angle x 10; chi^2 x 20)	G
Build tracks	All collections	Displaced cuts	H

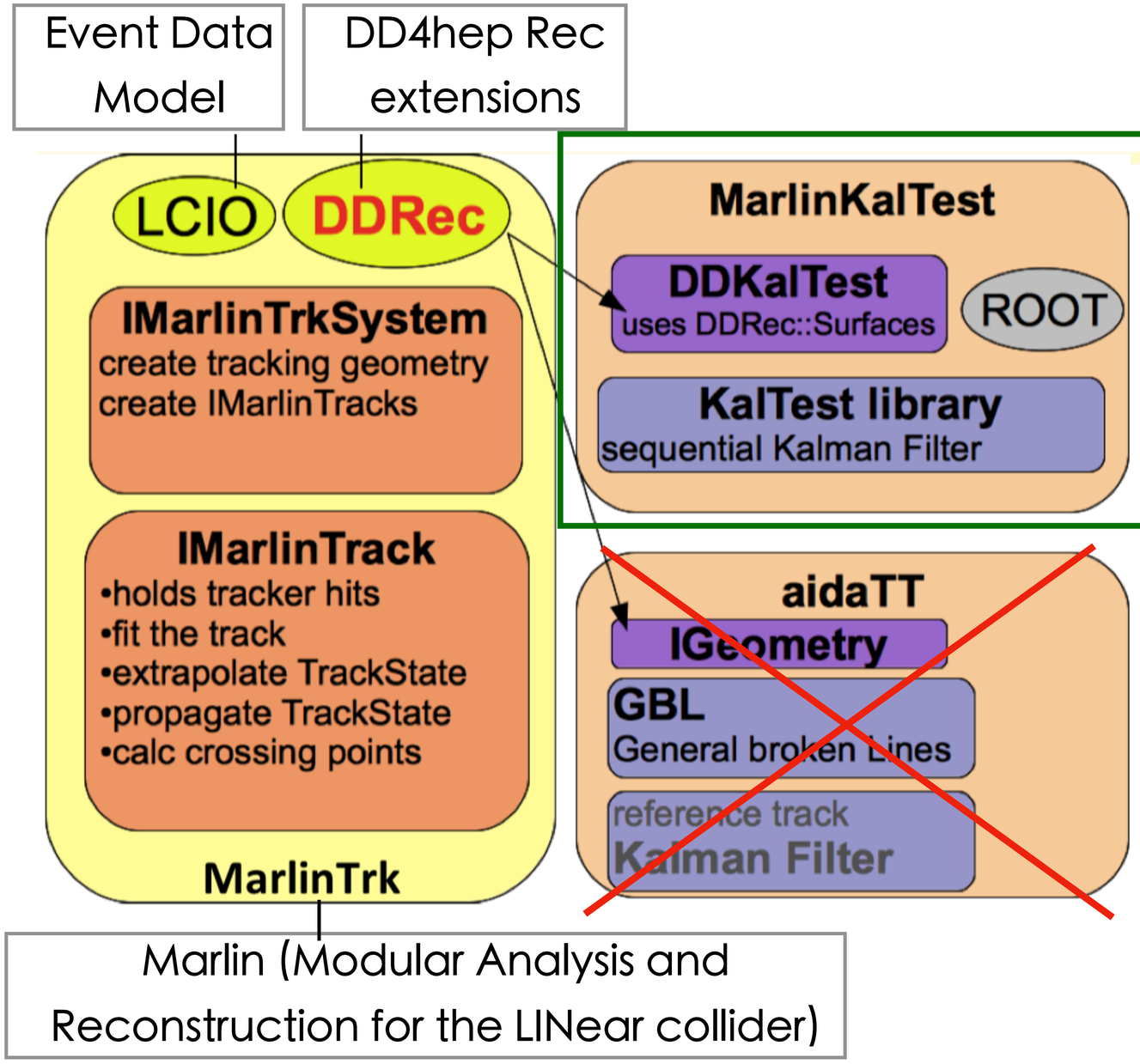
	1 ttbar event @3TeV (66 tracks)	1 ttbar event @3TeV with 30BX γγ->hadrons (680 tracks)
A	1.15s	1.55s
B	0.0007s	0.026s
C	0.07s	2.3s
D	0.28s	30s
E	0.21s	26.3s
F	0.20s	20s
G	0.05s	0.9s
H	10s	335s (=6min)

- ☆ Too many tracks ($5x10^5$) exception allows to skip events for which the combinatorics would explode

Fitting the tracks

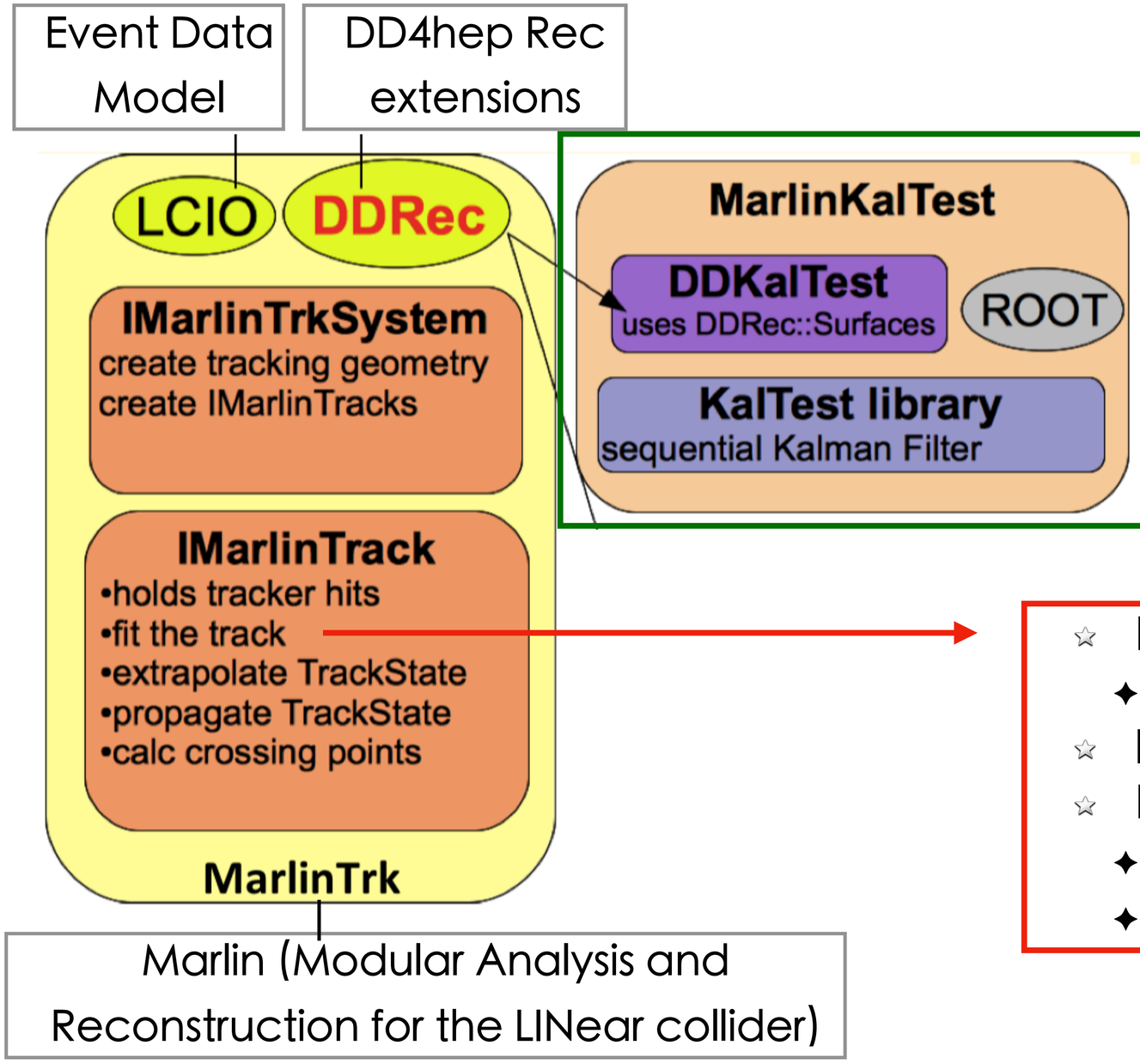


Fitting the tracks



- ☆ KalTest: iterative Kalman filter
- ☆ DDKalTest: DD4hep - KalTest = interface to provide surfaces

Fitting the tracks

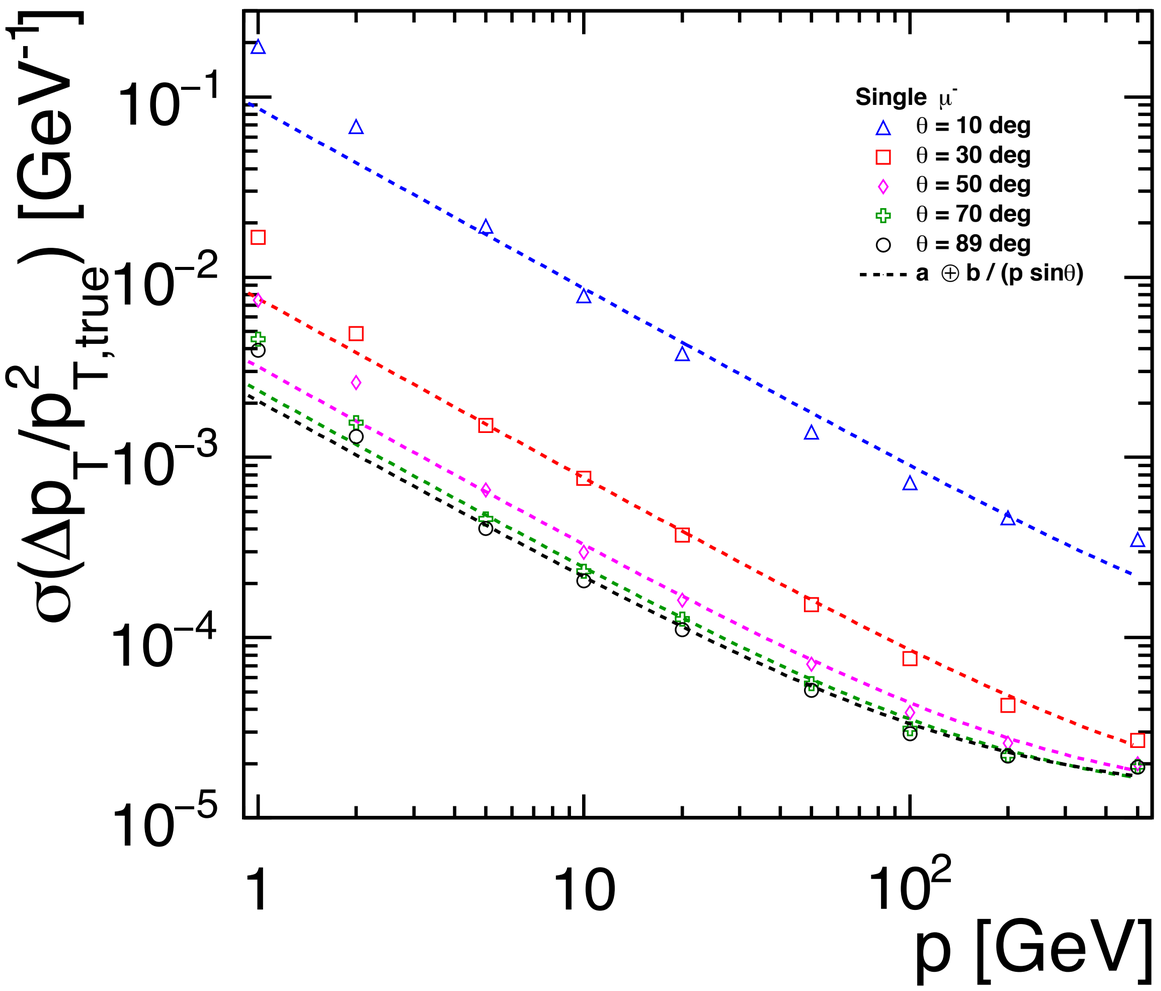


- KalTest: iterative Kalman filter
- DDKalTest: DD4hep - KalTest = interface to provide surfaces

- Helix prefit with 3 hits gives track state
  - first, middle, last
- Initialise fit with prefit parameters
- Kalman filter
  - hits added one by one
  - accepted/rejected based on a χ^2 cut

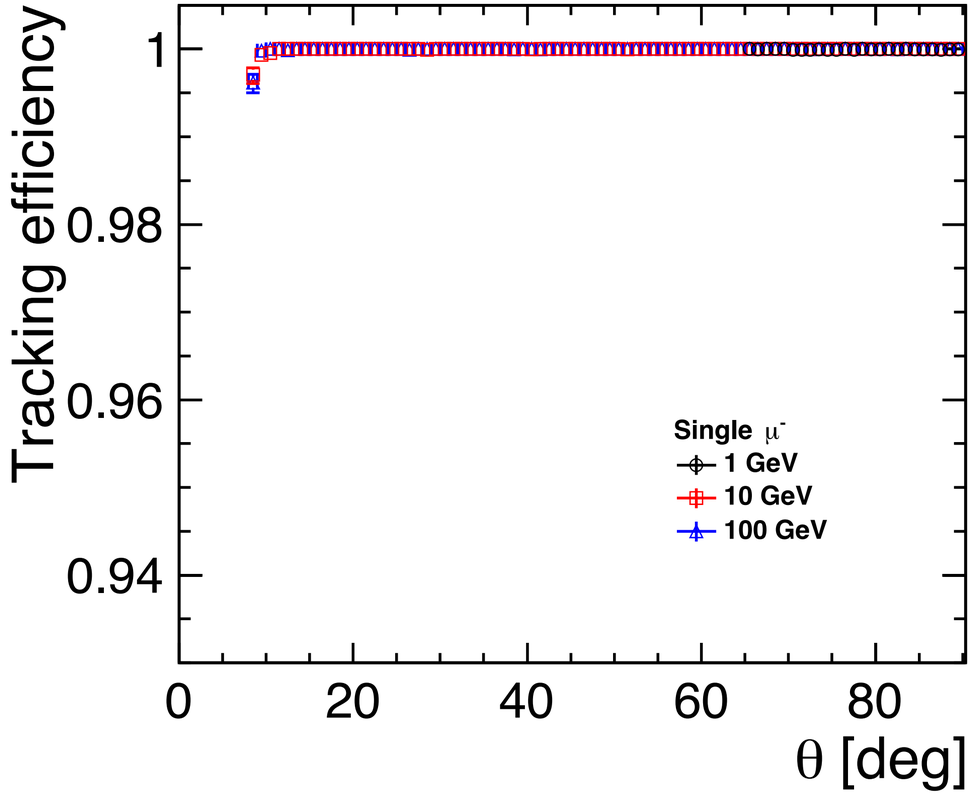
Performances 1. single particles

- Momentum resolution = σ of the Gaussian fit of the distribution $(p_{T\_reco} - p_{T\_true})/p^2_{T,true}$



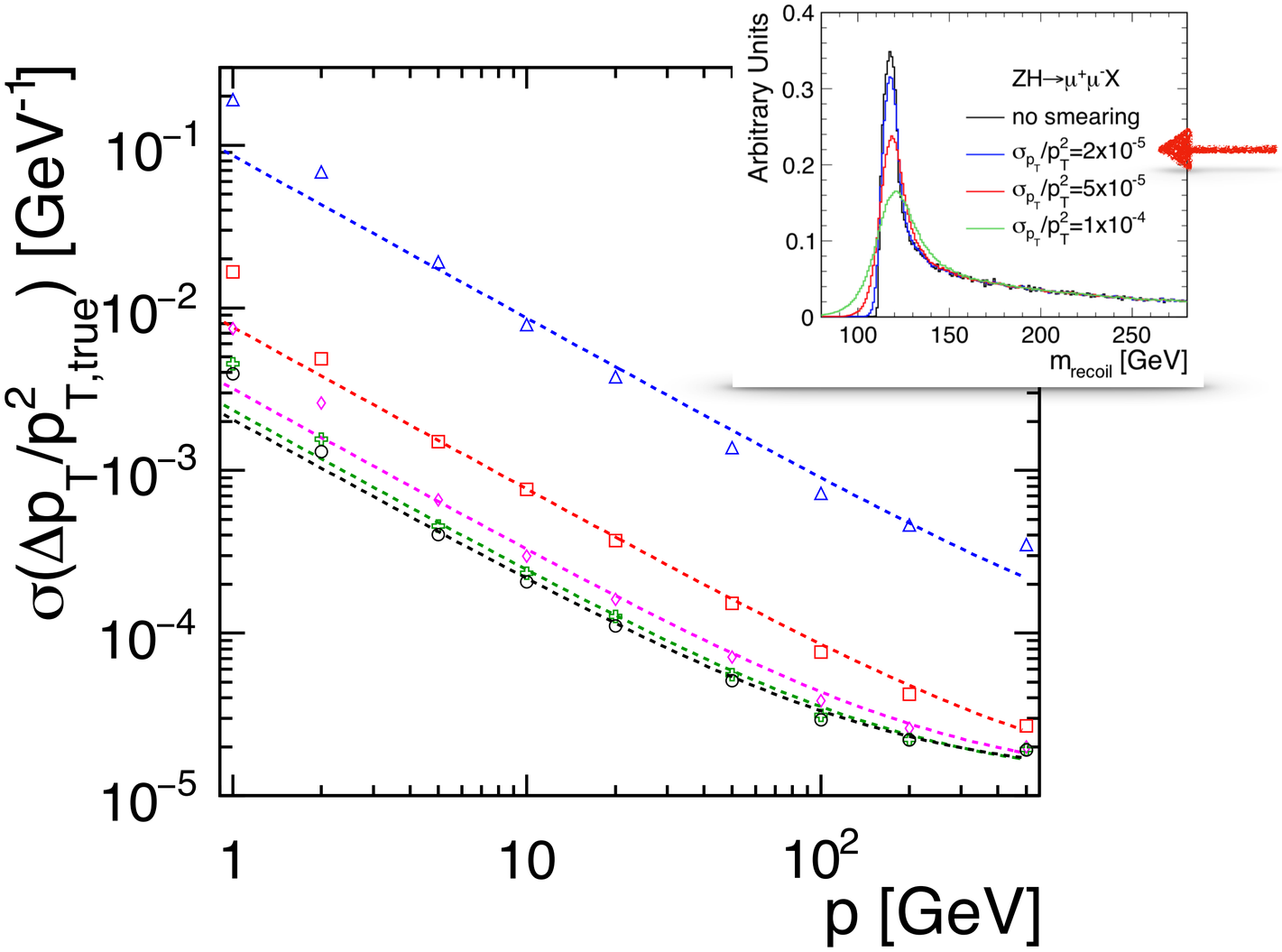
- Efficiency = fraction of reconstructed particles out of the reconstructable

  - stable
  - pT > 0.1 GeV/c
  - |cosθ| < 0.99
  - N unique hits >=4



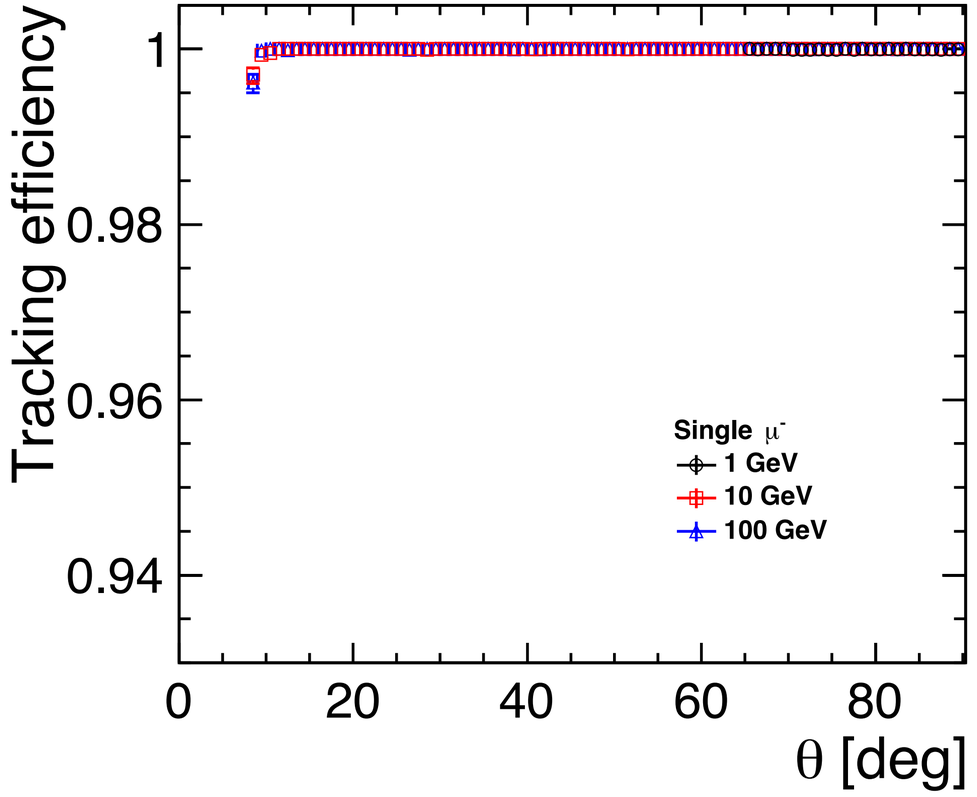
Performances 1. single particles

- Momentum resolution = σ of the Gaussian fit of the distribution $(p_{T\_reco} - p_{T\_true})/p^2_{T,true}$

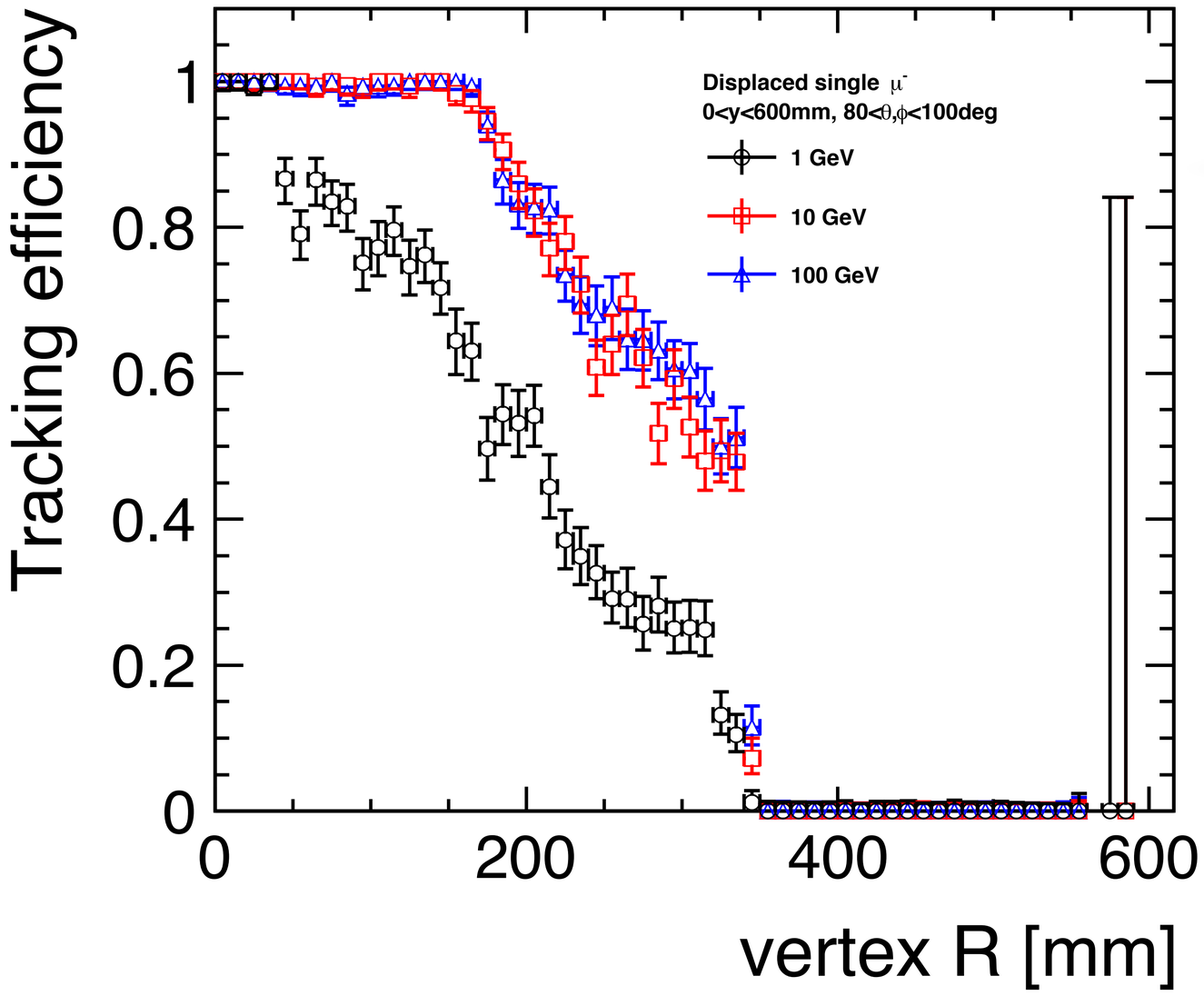


- Efficiency = fraction of reconstructed particles out of the reconstructable

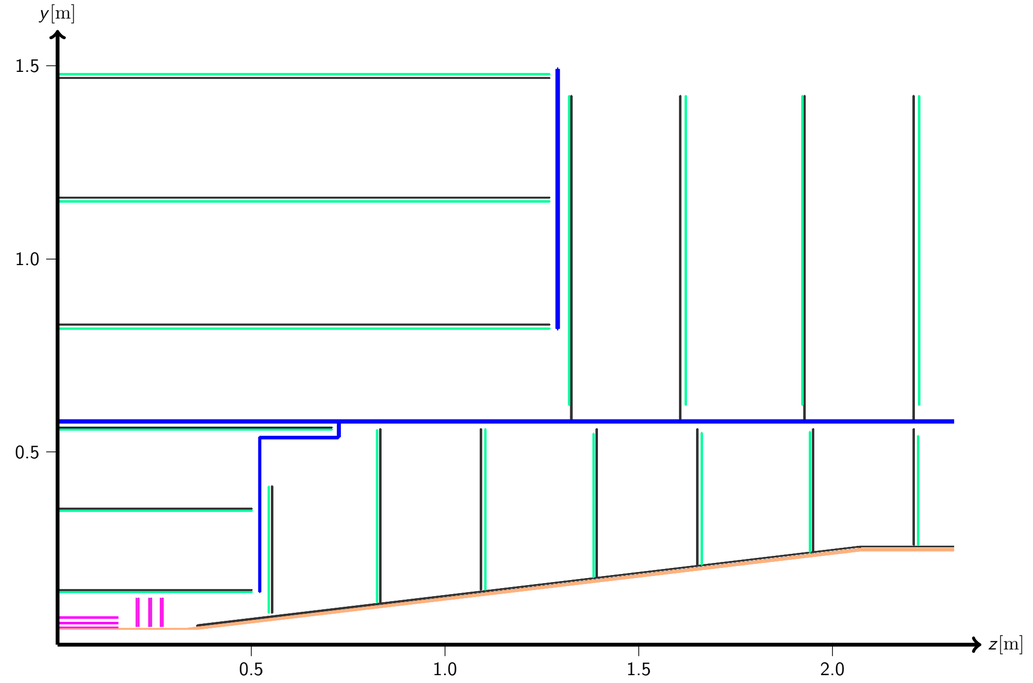
  - stable
  - pT > 0.1 GeV/c
  - |cosθ| < 0.99
  - N unique hits >=4



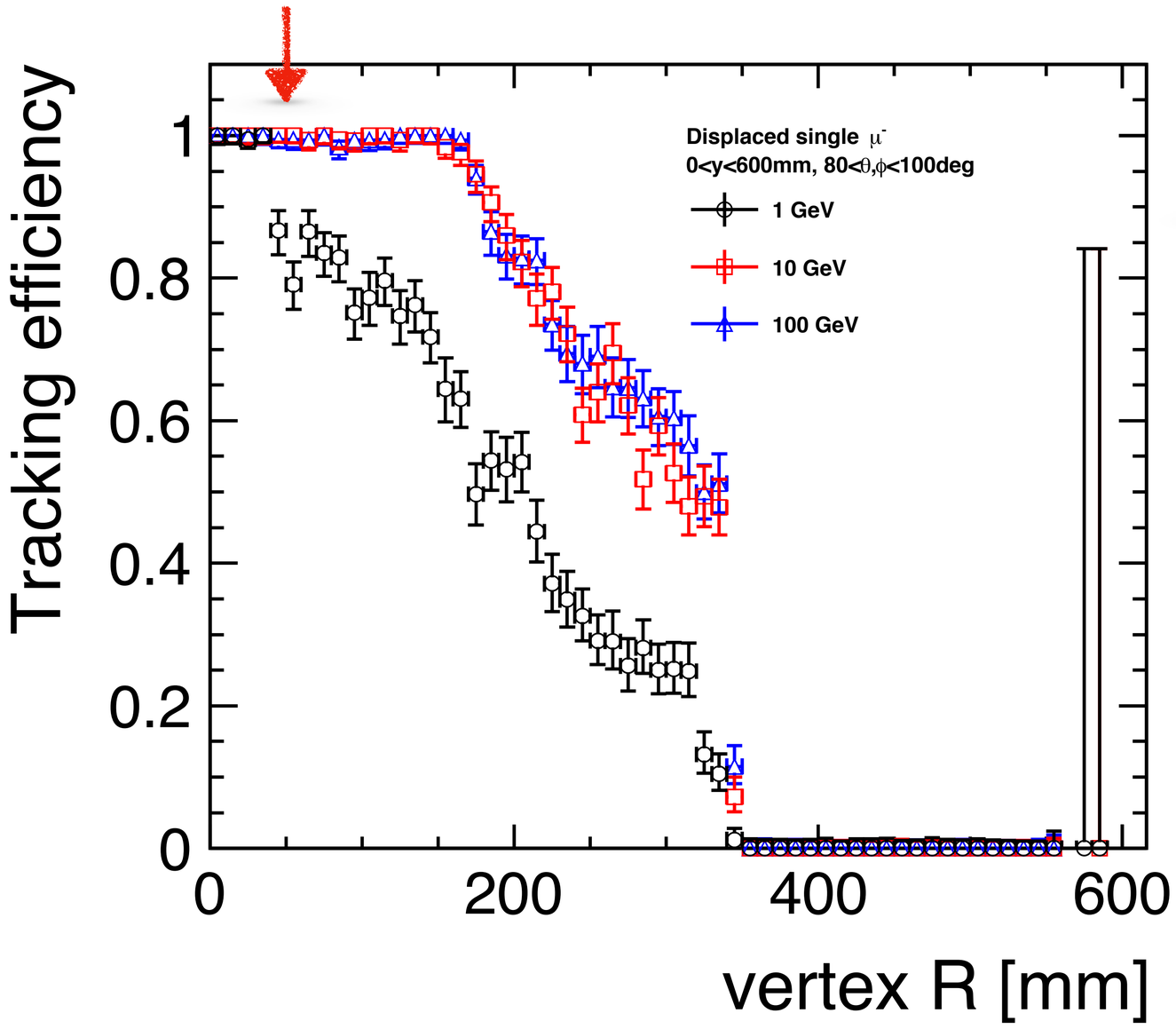
Performances 2. single particle efficiency for displaced tracks



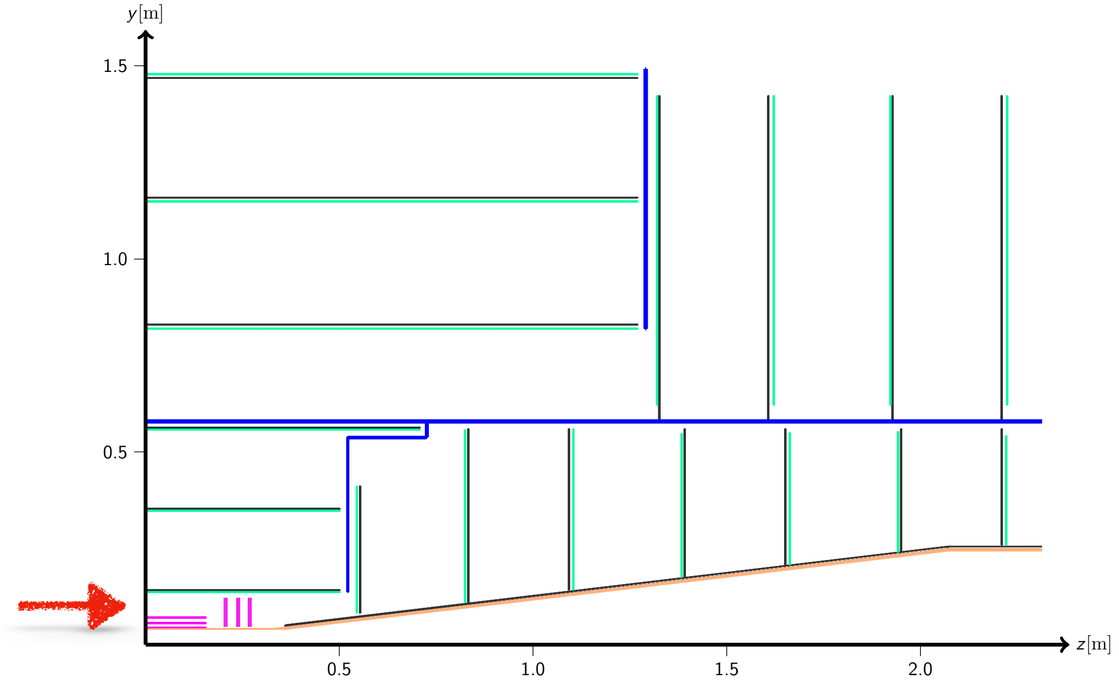
- broader search angle than for prompt tracks
- min number of hits: 5
- from tracker to vertex hits



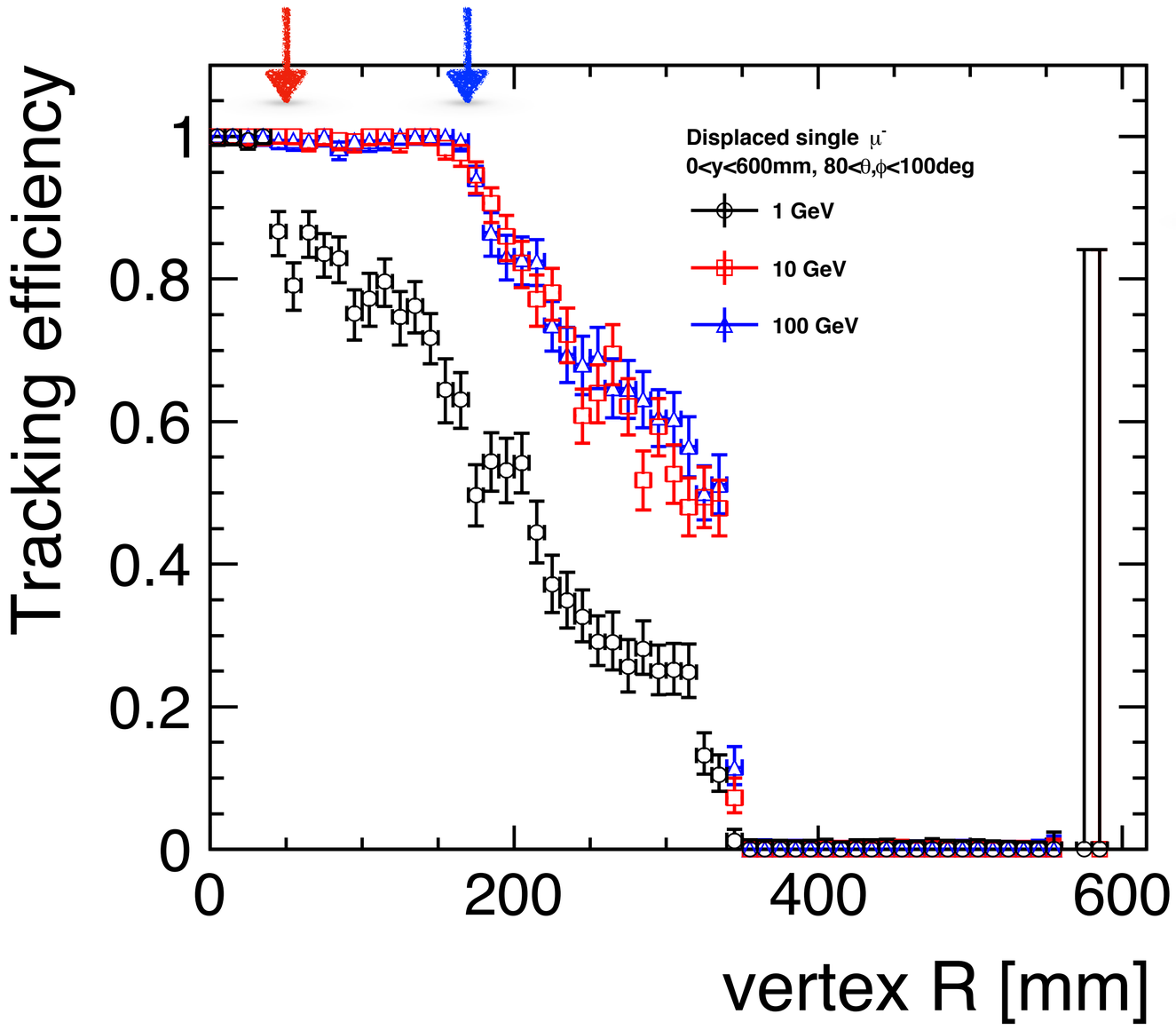
Performances 2. single particle efficiency for displaced tracks



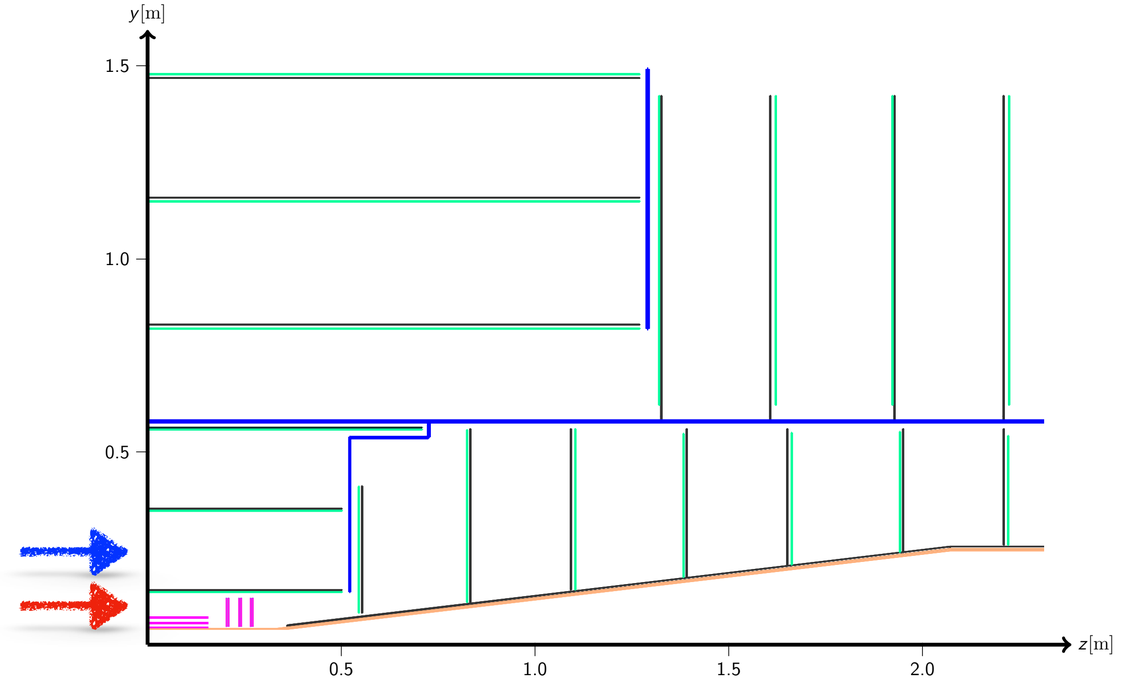
- broader search angle than for prompt tracks
- min number of hits: 5
- from tracker to vertex hits



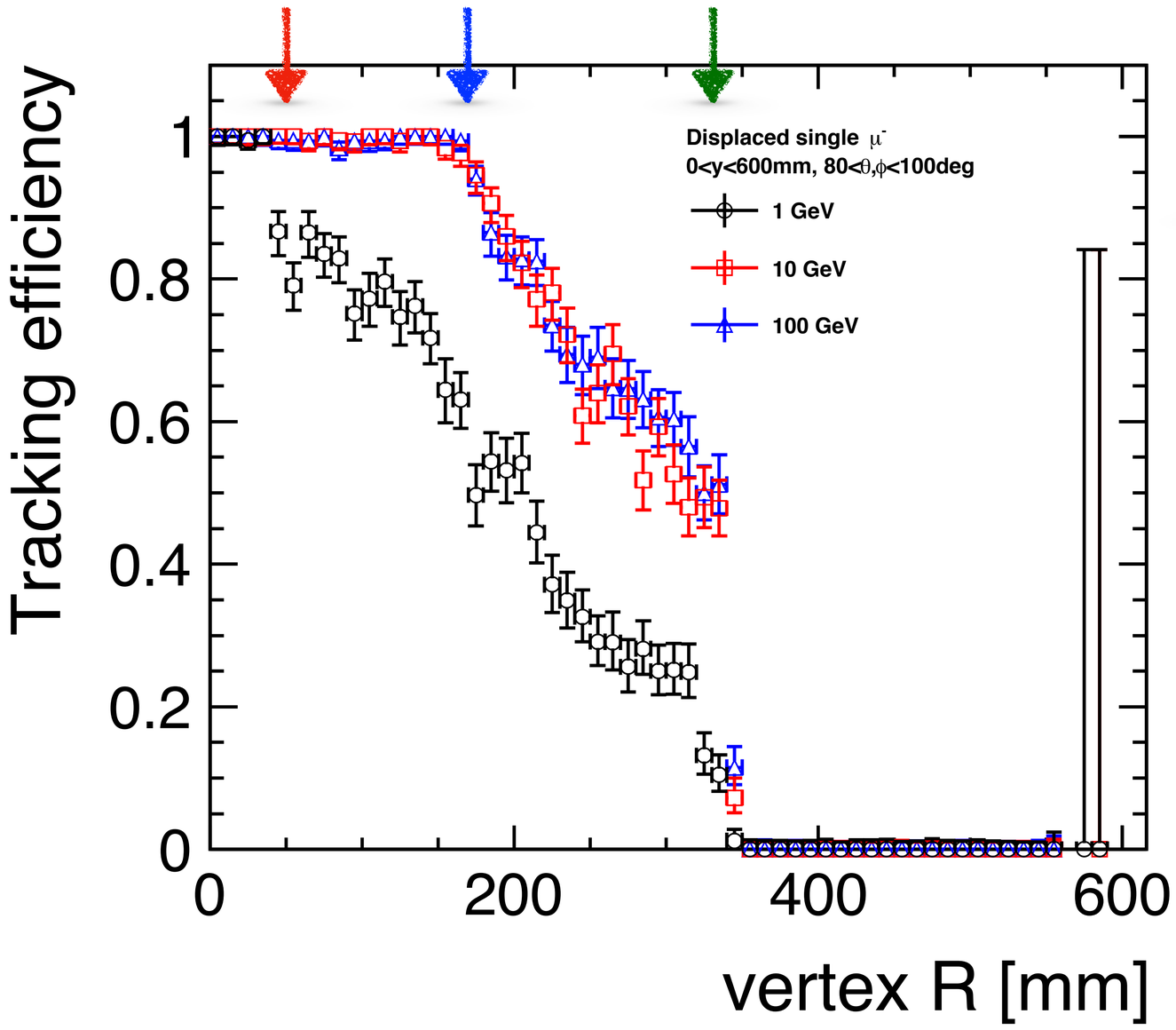
Performances 2. single particle efficiency for displaced tracks



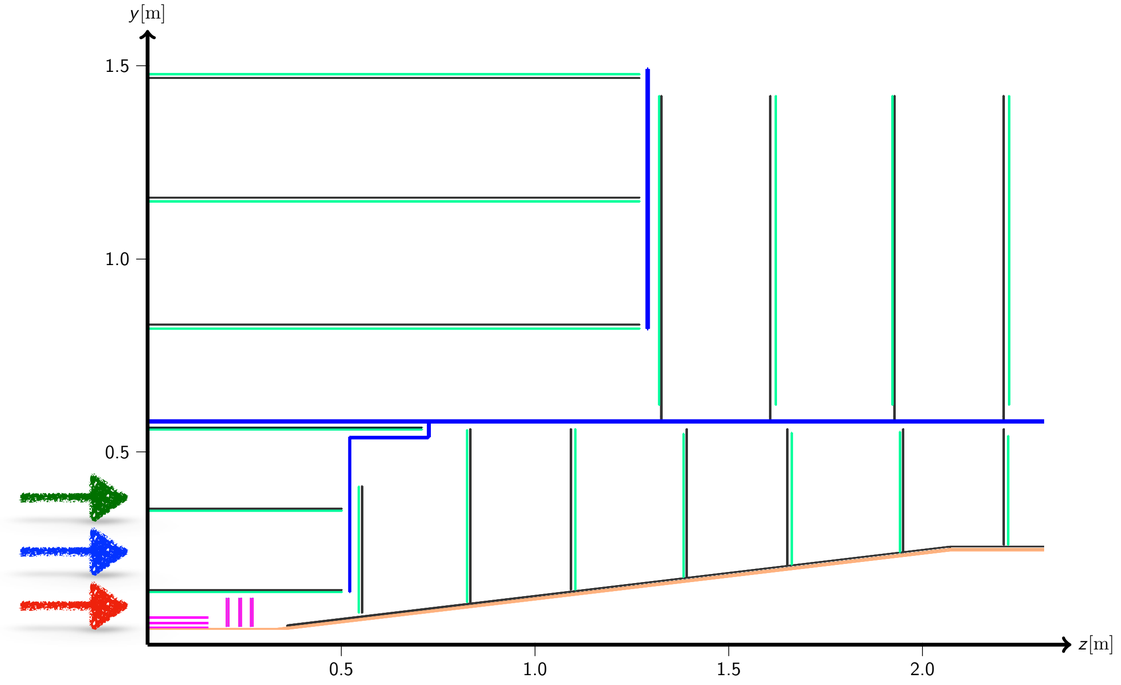
- broader search angle than for prompt tracks
- min number of hits: 5
- from tracker to vertex hits



Performances 2. single particle efficiency for displaced tracks



- broader search angle than for prompt tracks
- min number of hits: 5
- from tracker to vertex hits

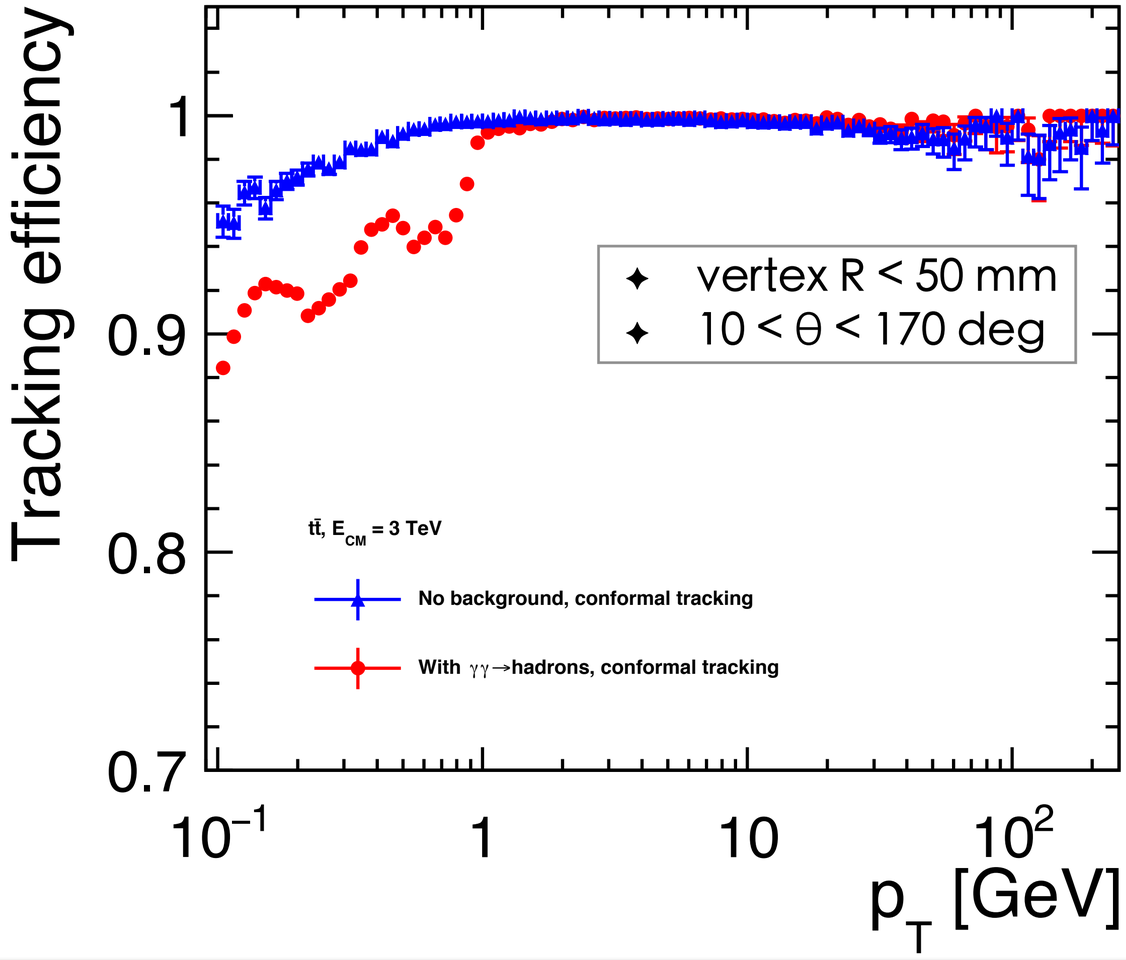


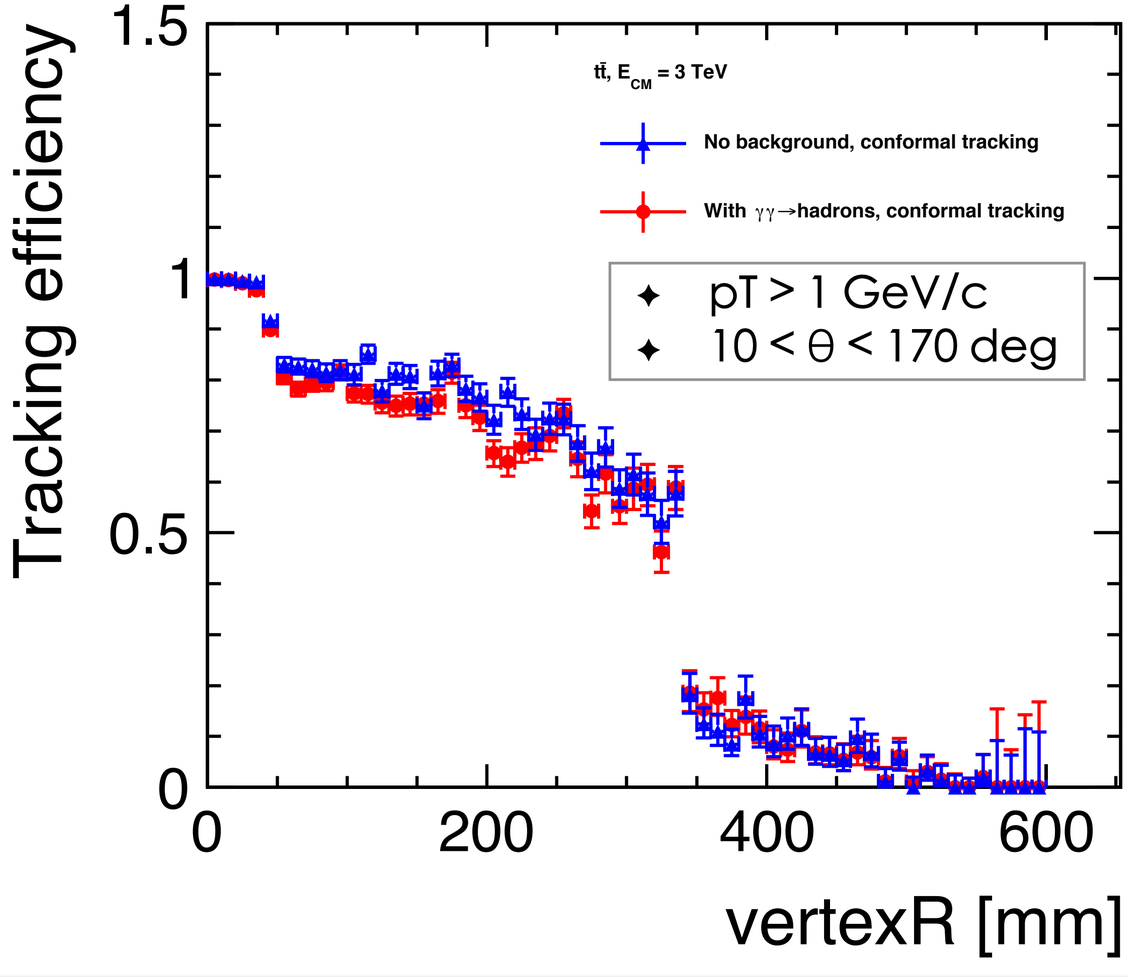
Performances 3. efficiency for ttbar events at 3TeV

☆ Efficiency = fraction of pure reconstructed particles out of the reconstructable

- purity >= 75%
- purity = Nhits belonging to the associated MC Particle / total Nhits

- stable
- pT > 0.1 GeV/c
- |cosθ| < 0.99
- N unique hits >=4



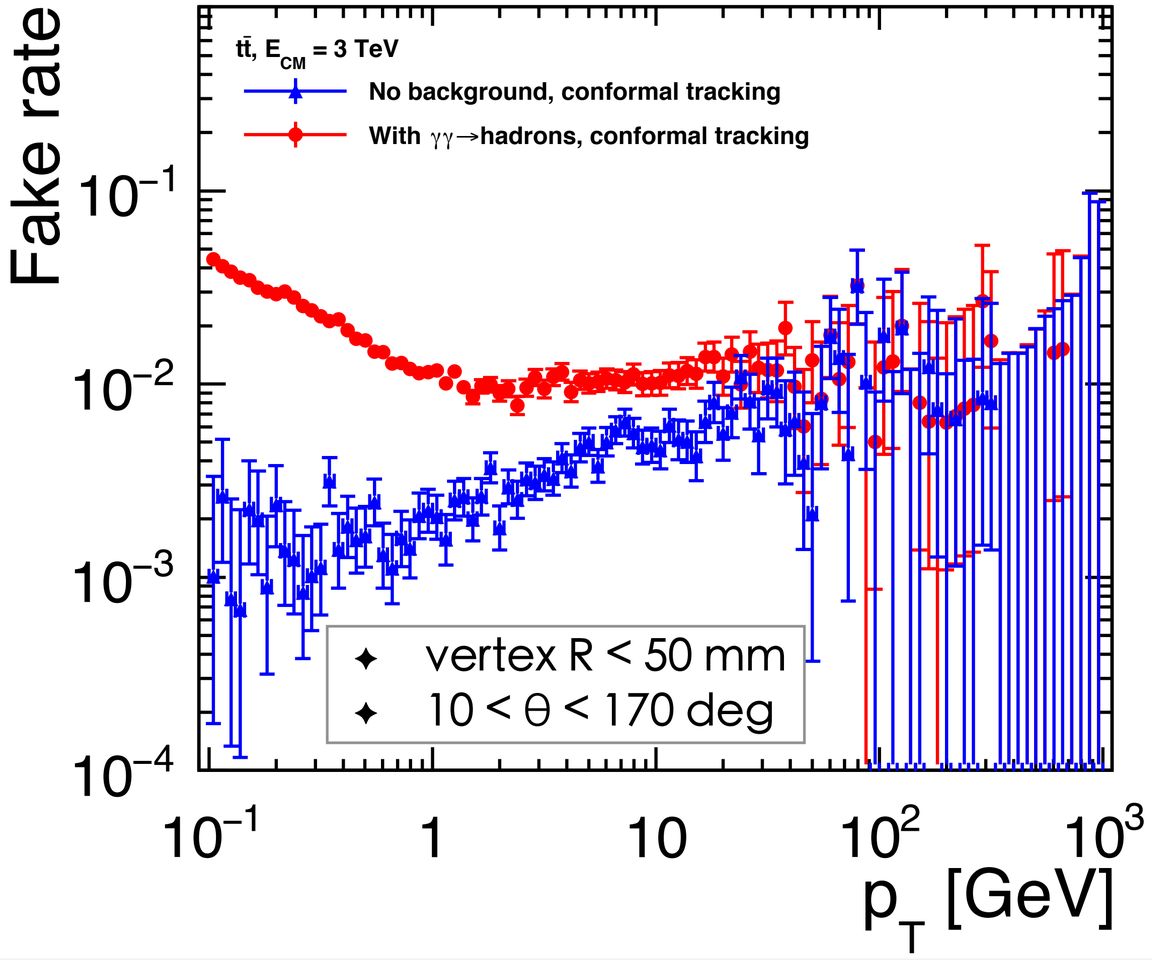


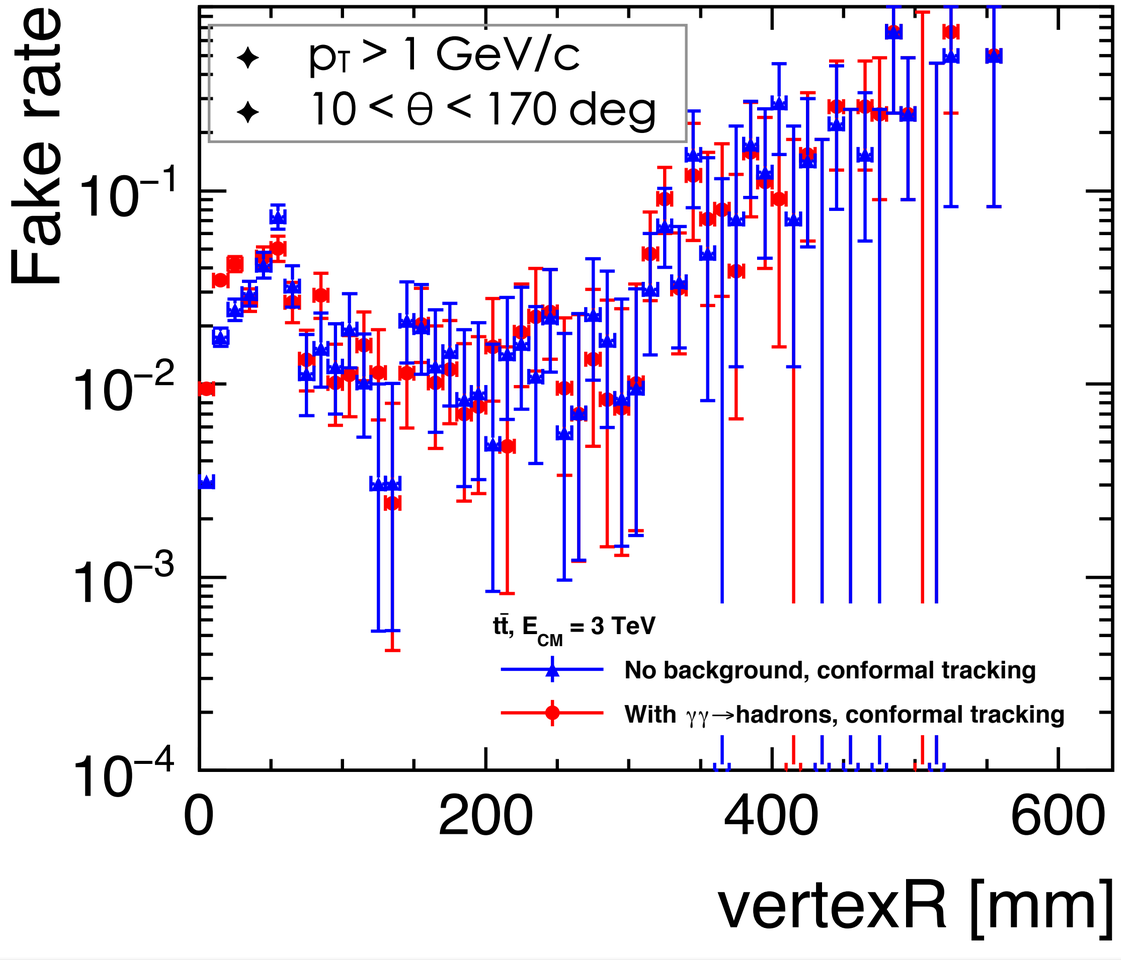
Performances 4. fake rate for ttbar events at 3TeV

☆ Fake rate = fraction of impure reconstructed particles out of the reconstructable

- purity < 75%
- purity = Nhits belonging to the associated MC Particle / total Nhits

- stable
- pT > 0.1 GeV/c
- |cosθ| < 0.99
- N unique hits >=4



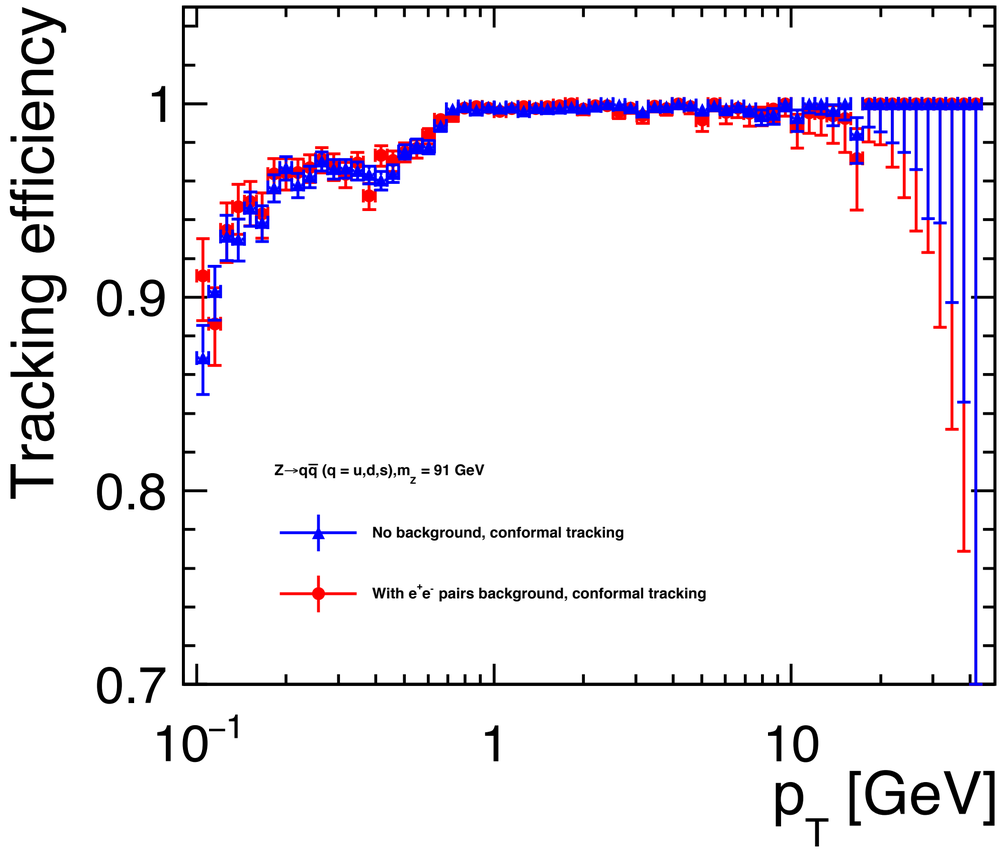


Performances 5. tracking efficiency for the FCC-ee CLIC-Like Detector (CLD)

Main (but not all) differences between CLICdet and CLD

- ☆ Detector solenoidal field 2T (4T for CLIC)
  - ✦ Outer tracker radius increased to 2.15m (1.5m)
- ☆ Beampipe radius 15mm (29mm)
  - ✦ Inner vertex radius decreased to 17mm (31mm)

→ Tuning of pattern recognition parameters (search angles, distances)

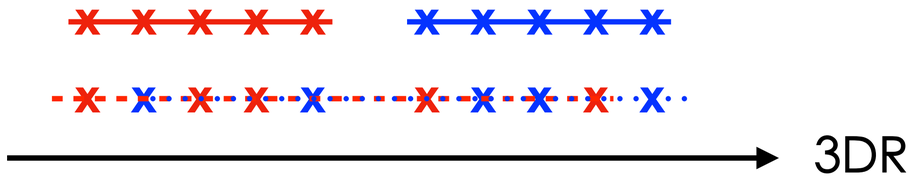


- ✦ Z-like boson events decaying at rest into light quarks
- ✦ + background from e+e- pairs

Further improvements and open points

- Currently implementing a check for clones at the end of the pattern recognition chain
  - longest track normally preferred, if the χ^2 is not too much larger

- Using MonteCarlo information for debugging purposes, it was realized that some particles are reconstructed by multiple tracks
  - split tracks
  - tracks picking complementary hits



- At the moment, artificially merging the hits from the two tracks and fit
  - angular and p_T matching
- But: problem of overmerging

Summary

- ☆ CLIC is the only multi-TeV option on the market for future lepton-collider
- ☆ A detector for CLIC has been designed, optimized and validated
- ☆ Its low-mass tracking system allows for conformal tracking algorithm

- ☆ Detector is available in full simulation
- ☆ Software reconstruction chain is fully implemented
  - ✦ simulated —> reconstructed hits
  - ✦ hits in conformal space
  - ✦ pattern recognition via cellular automaton
  - ✦ track fit

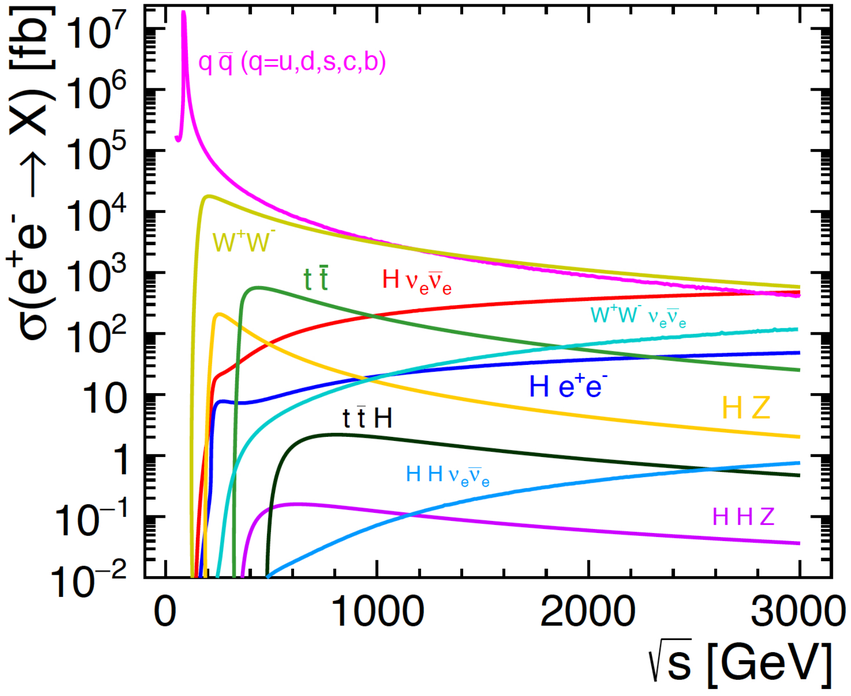
- ☆ Conformal tracking performs in reconstructing single particle and complex events
  - ✦ special effort has been spent lately for displaced tracks reconstruction
  - ✦ ongoing/foreseen improvements, cut optimizations, speed gain, solving open points

Special thanks to A. Sailer, D. Hynds, R. Simoniello
for their past and present work

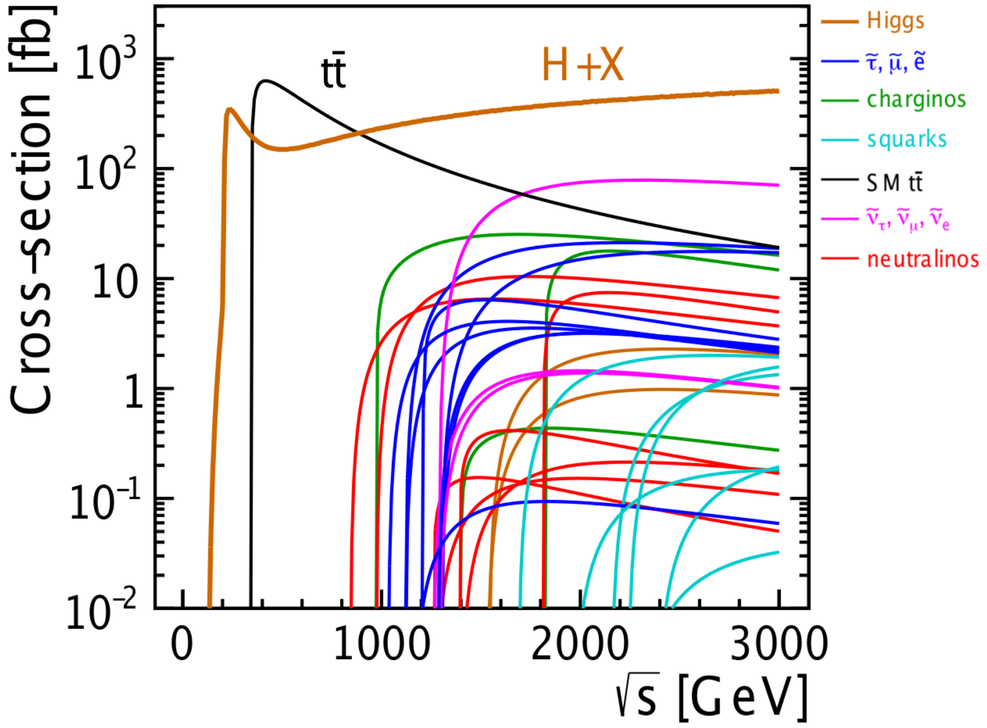
BACKUP SLIDES

CLIC physics

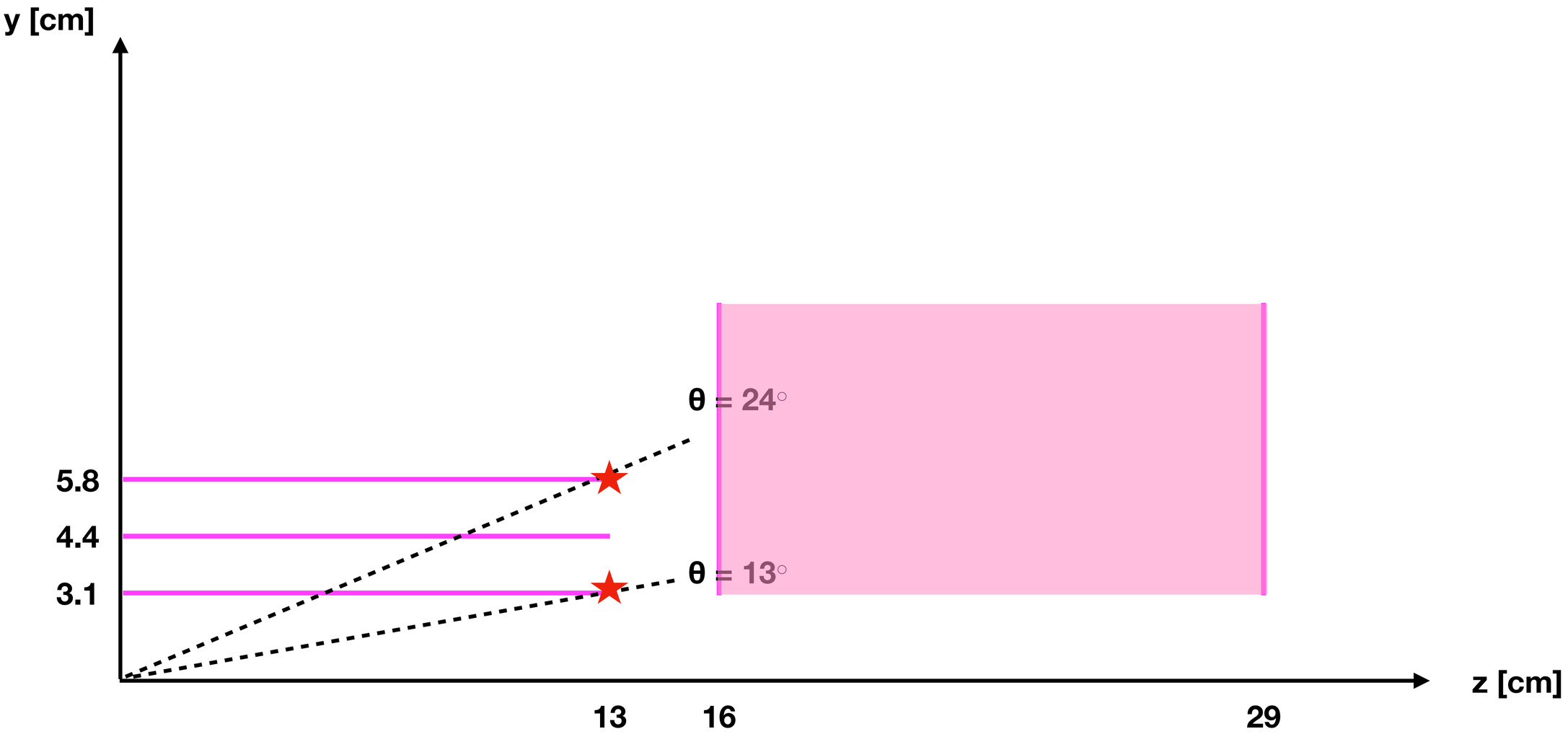
- Standard model physics
  - Higgs and top precision measurements



- BSM physics
  - masses, mixing angles, coupling, spins of the new sparticles



CLICdet vertex layers - radii and angular coverage



CLICdet tracker layers - radii and angular coverage