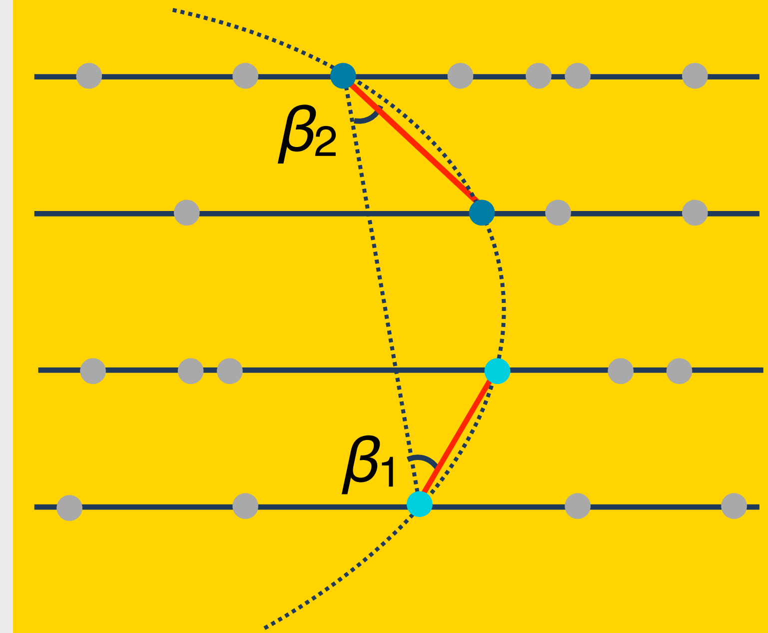
Parallelizable Track Pattern Recognition in High-Luminosity LHC

CTD 2020
April 20, 2020



Philip Chang,
V. Krutelyov, M. Masciovecchio,
B. Venkat Sathia Narayanan,
M. Tadel, A. Yagil

Univ. of California
San Diego

Outline

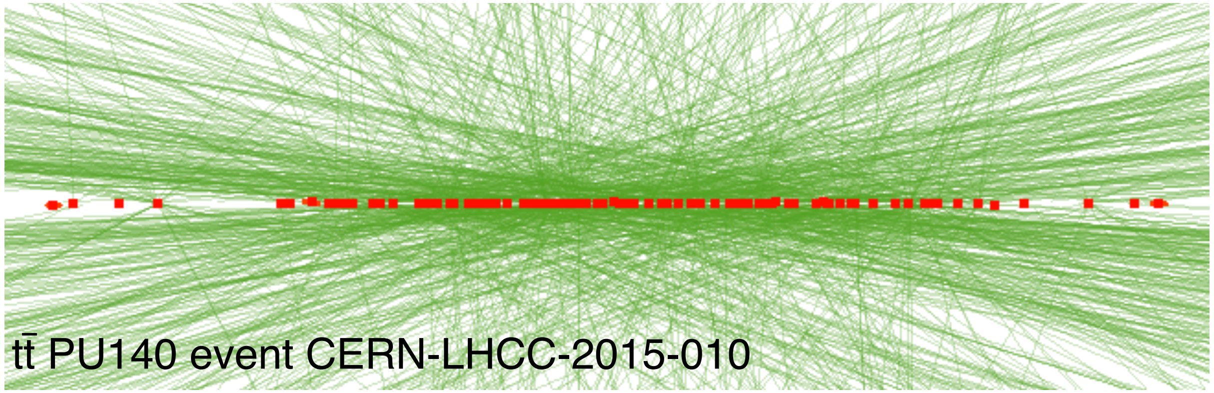
- HL-LHC challenges
- Importance of investigating novel algorithms
- Parallel track building in CMS Phase 2 outer tracker
- Preliminary results
- Outlook

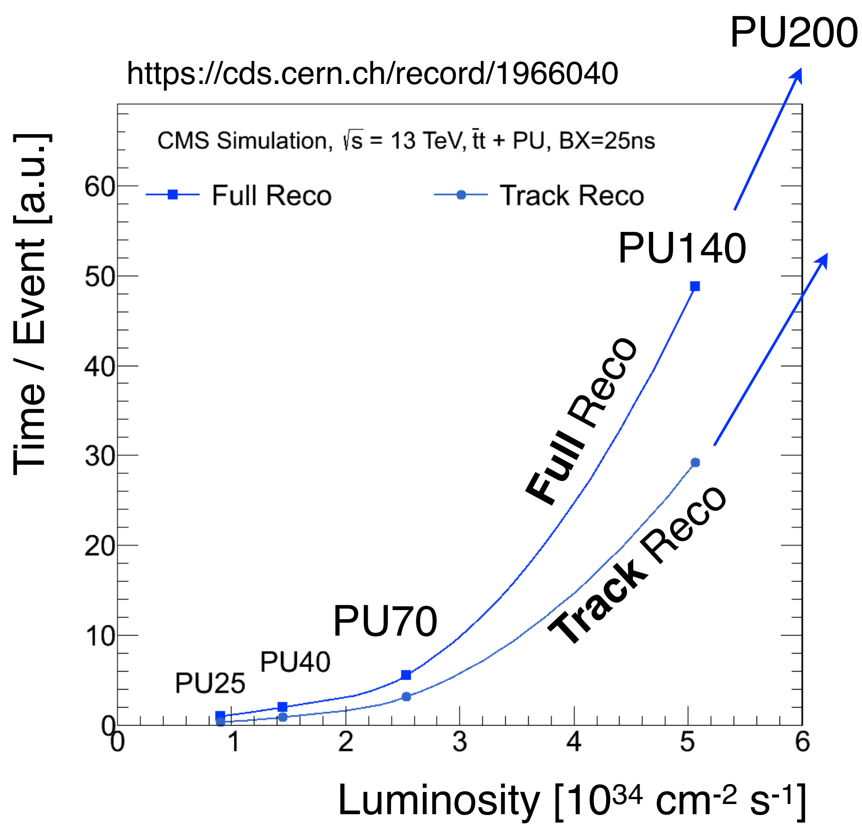
Challenges of tracking in HL-LHC

Chang
UCSD

- Track reconstruction is crucial for physics
- Track reconstruction is time consuming
- For HL-LHC the problem is only getting worse
- New (multiple) solutions must be explored

  ⇒ Most important thing to keep in mind is the impact to physics performance





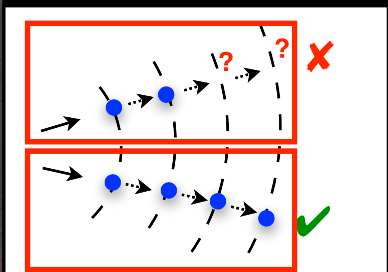
For HL-LHC need to make choices / improvements in order not to compromise physics due to large pileup

Novel algorithms on the horizon

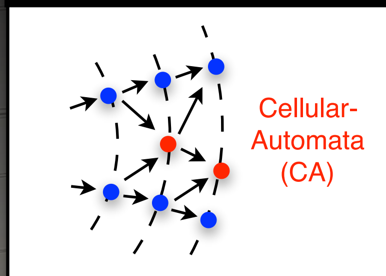
Chang
UCSD

Parallel in seeds/evt./region and vectorize calculation (SIMD)

mkFit



Patatrack

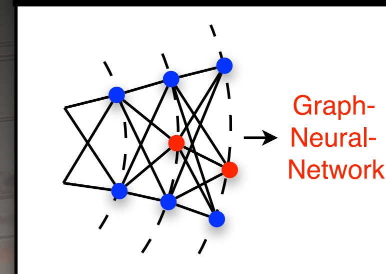


only need to know near neighbor ⇒ parallelizable

GPUs

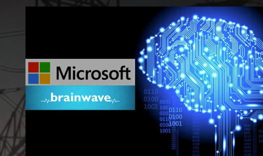
Multi-core

Exa.TrkX



Bunch of matrix multiplication ⇒ can be parallelized

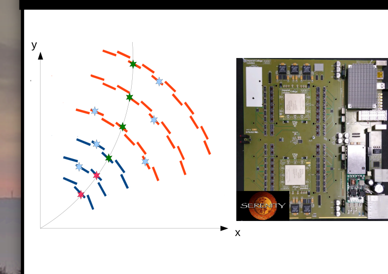
Cloud?



FPGA deploy?

subregion/event parallelization

L1Track



Parallelizing Science
paralléliseur de science

Most algorithms currently used or used today are sequential

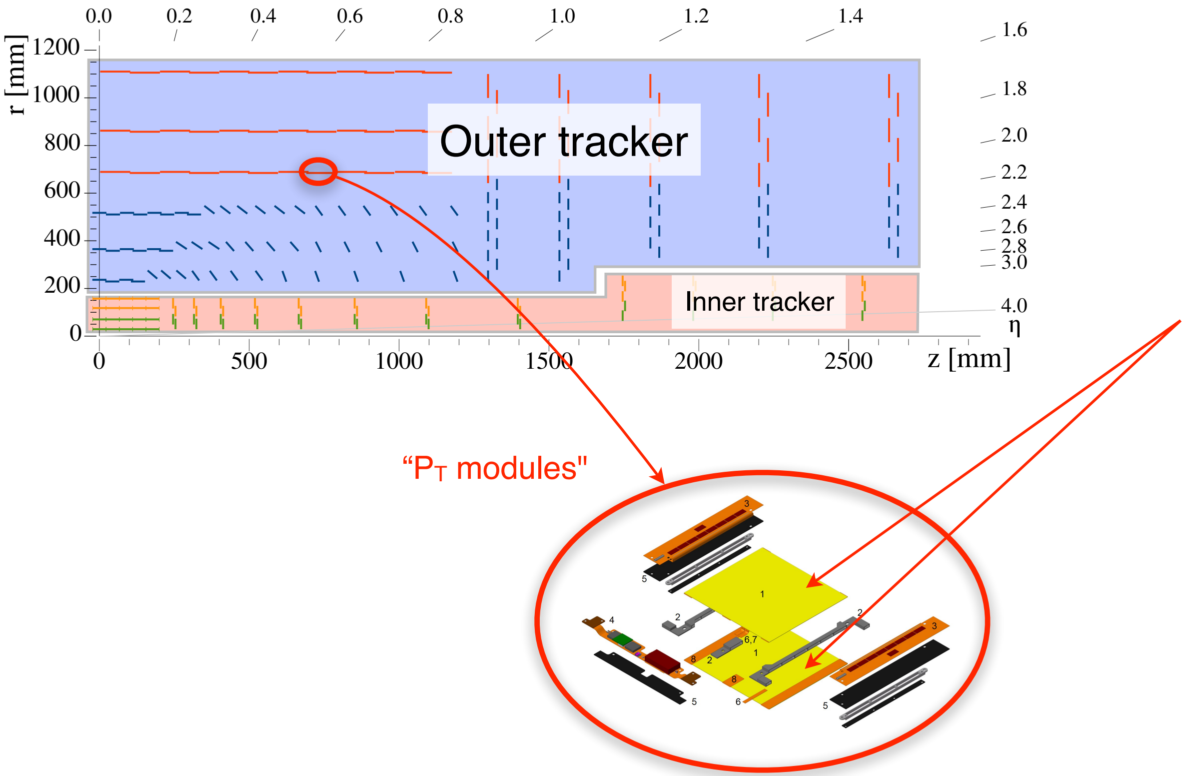
Many novel efforts are developing **parallelizable** algorithms

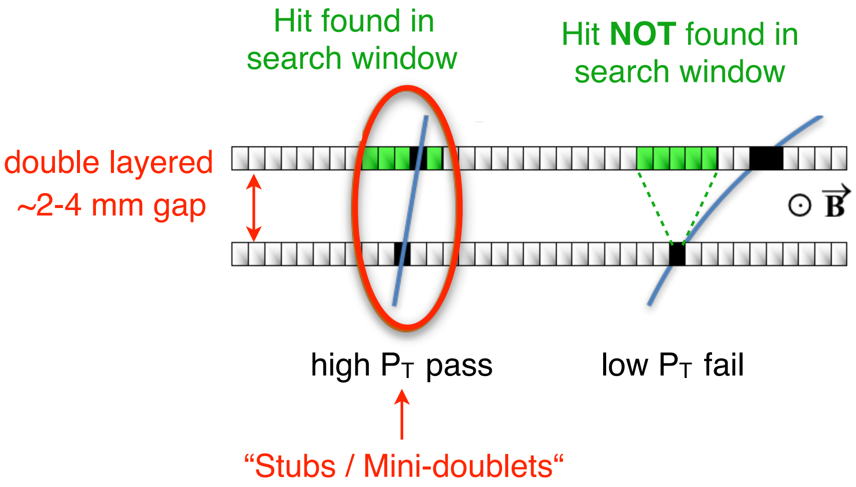
(See many contributions from the CTD2020)

CMS Tracker in HL-LHC

Chang
UCSD

- Outer tracker consists of "P_T *modules*" (two modules closely sandwiched)
- Choice driven by level-1 track trigger capability
- Forming stubs / **mini-doublets** is local and can be highly parallelized

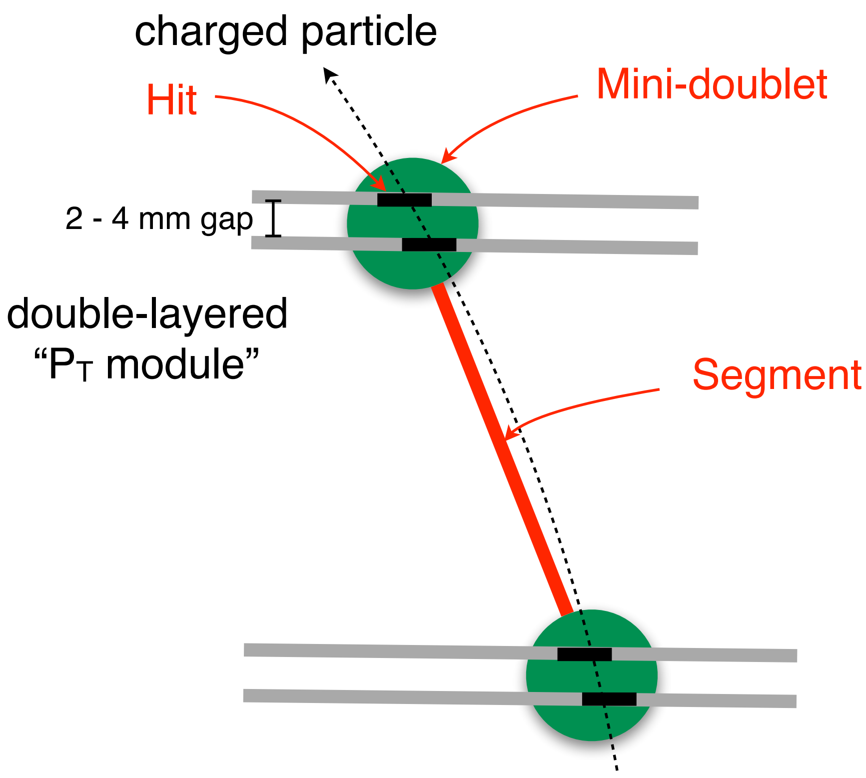


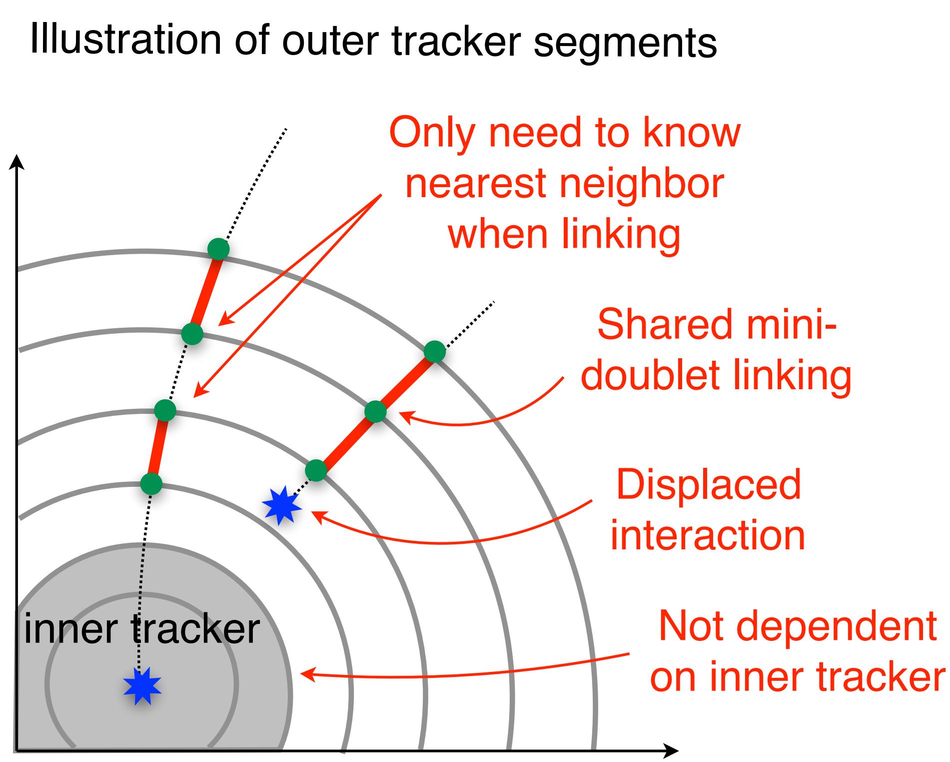


Enables parallel algorithm in outer tracker

Segment building and linking

- Two **mini-doublets** (•) together can form a **segment** (•——•)
- Segment building and linking† algorithm is also inherently parallel
- Outer tracker segment linking would have unique physics opportunity

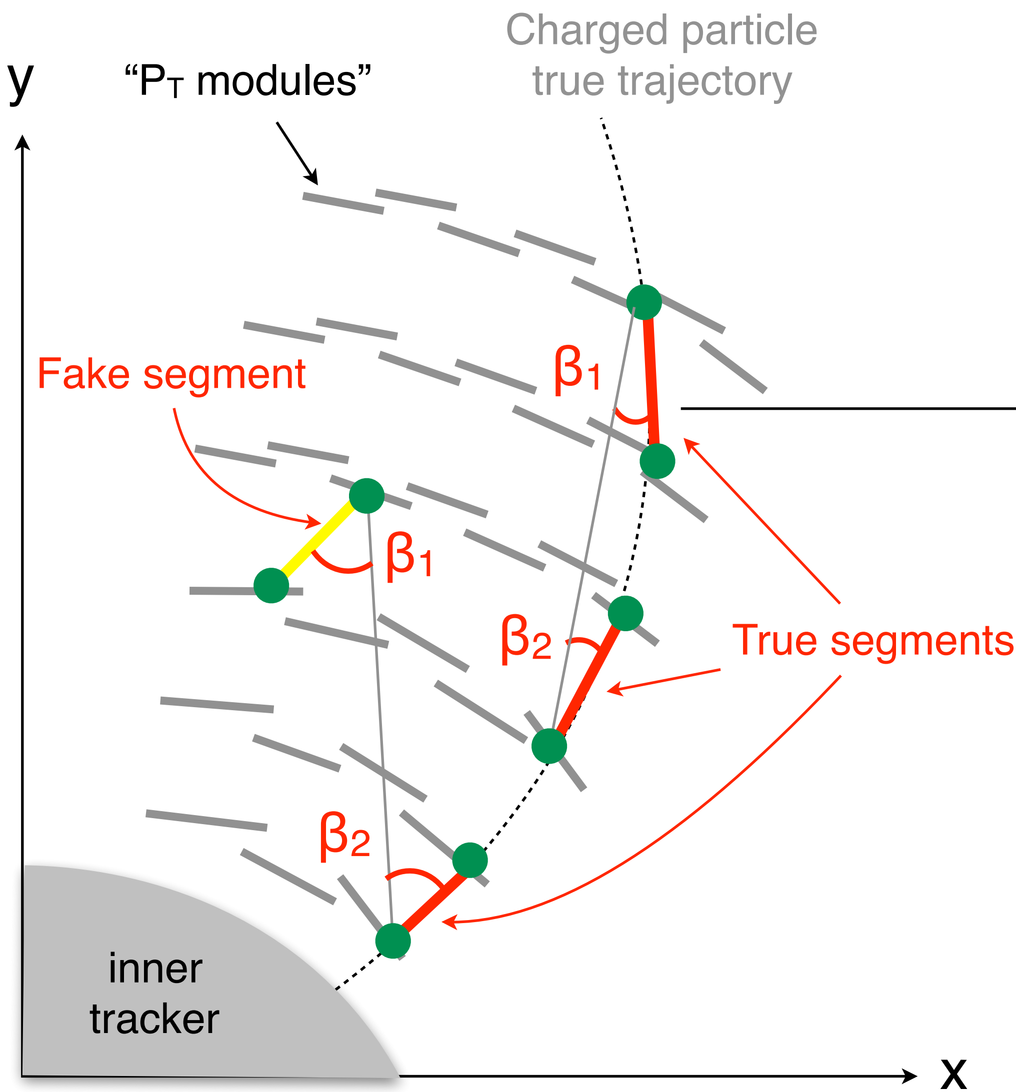


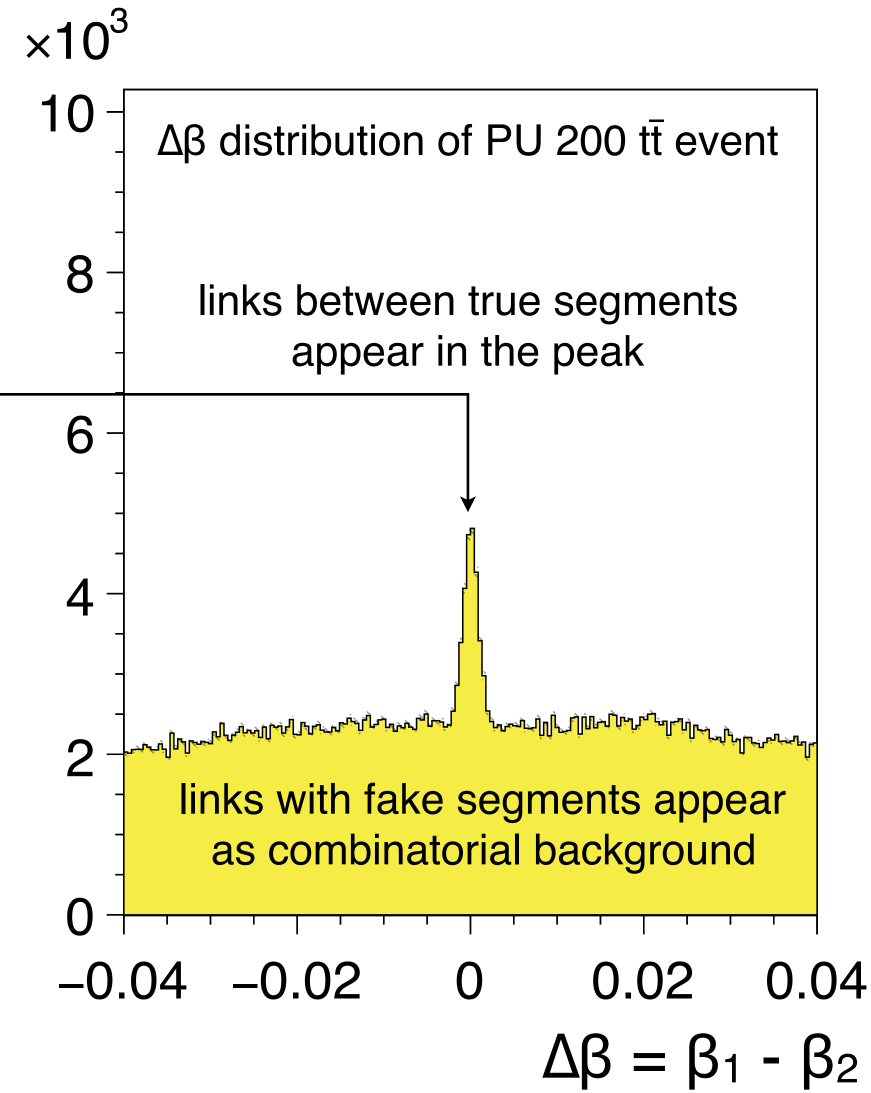


†Similarity to CDF central outer tracker (COT) segment linking, (cf. eXtremely Fast Tracker)

Track pattern recognition algorithm that is inherently parallel can be explored in the outer tracker of HL-LHC CMS tracker

Geometrical linking of segments





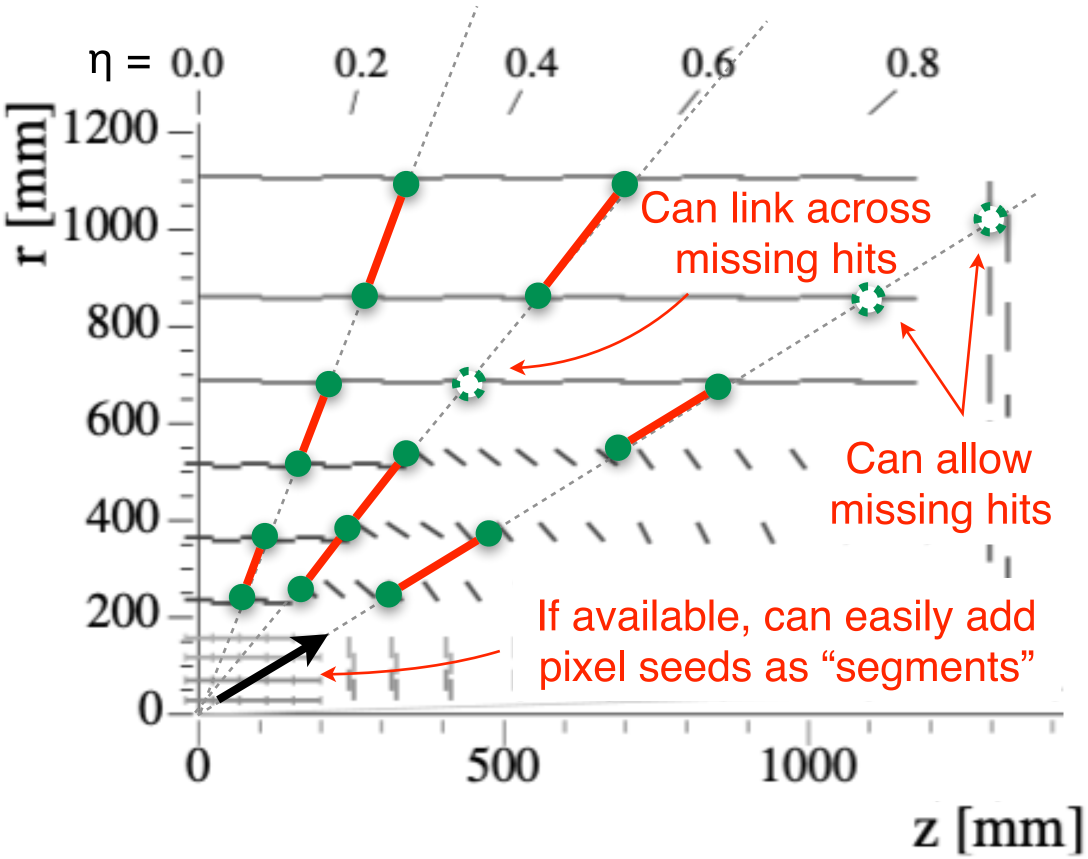
Method based on J.Phys.Conf.Ser. 898 (2017) 4, 042023

Segments can be linked geometrically

Robustness of the approach

non-exhaustive list of possible track candidates built shown below.

- Segments (or linked ones) are used to build track candidates
- There are multiple ways to build, which can all be done in parallel
- Pixel seeds can be easily added as a “segment” if available

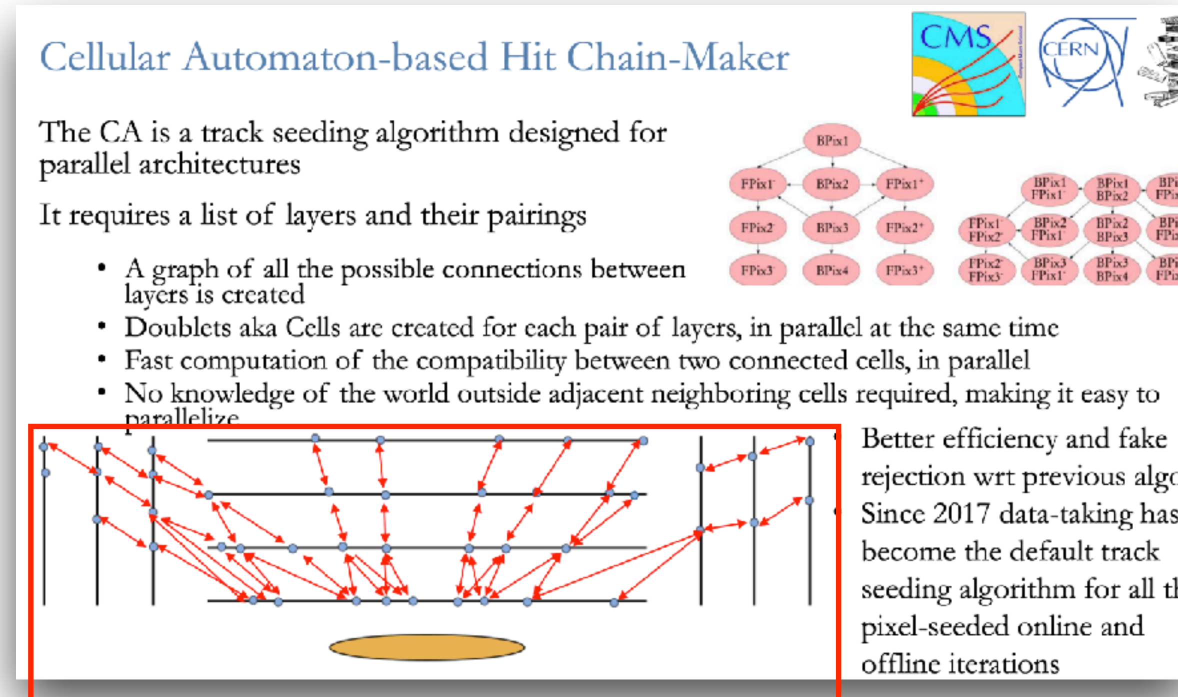


Track candidates can be formed in **multiple** ways in **parallel**

Connection to other parallel algorithms

- Byproduct of the method is a full graph connection between mini-doublets
- Possible to explore different methods at this stage
  - e.g. CA, GNN

CTD2019 Patatrack F. Pantaleo



Equivalent graph obtained

The algorithm being developed has many legs

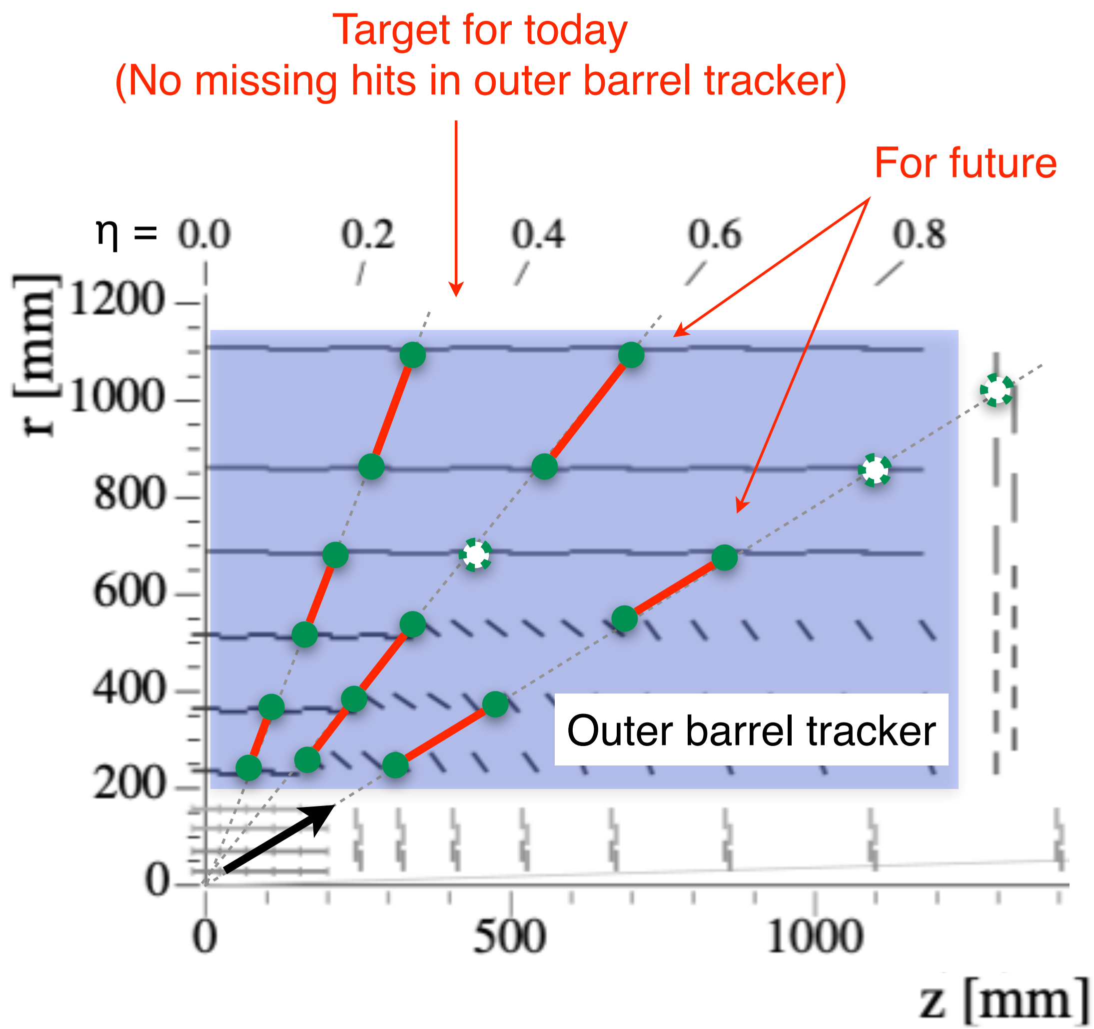
Benefits of OT parallel track building algo

- Benefits from parallel algorithm:
  - Possible speed up if combinatorial/fakes are under control
  - Traditional Kalman-Filter based algorithm is serial and sequential
    - (Efforts on-going to deploy on modern architecture. e.g. mkFit)
  - Proposed track building algo is naturally parallelizable and vectorizable
- Outer tracker benefits:
  - Complementarity w/ pixel based seeding
  - CMS tracking is critically dependent on pixel health and performance
  - Affordable iterations are all pixel-based
  - Single largest exposure to physics program
  - Potential for displaced vertex tracking
- Could also potentially benefit at HLT for more extensive tracking

Outer tracker parallel track building algo can be fast and help physics program of CMS

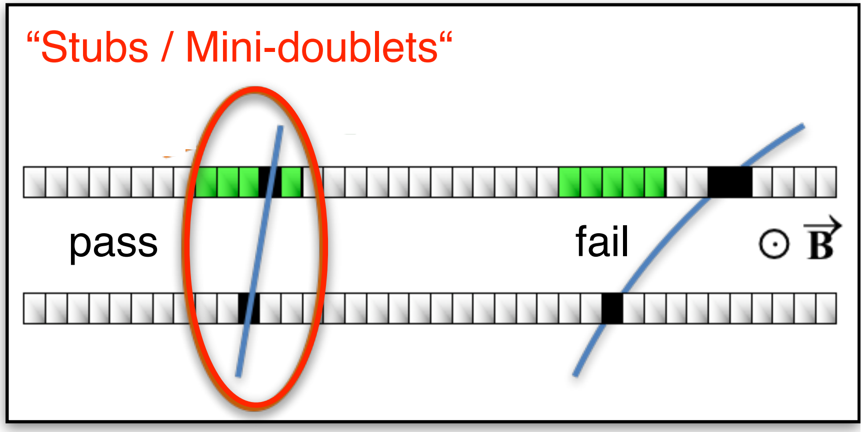
Case study for this talk

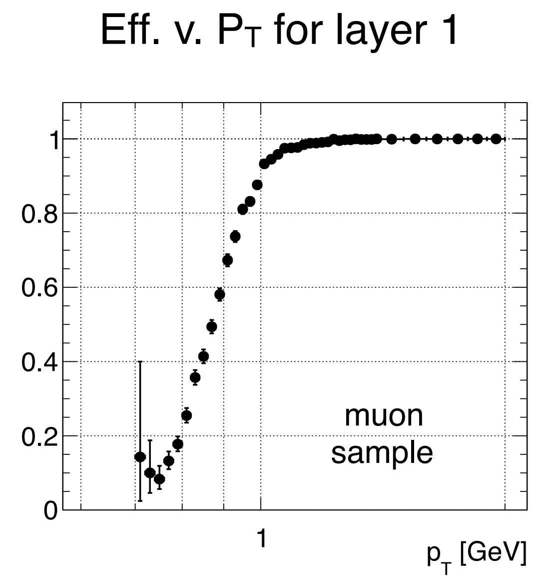
- As a start we focus on no missing hits tracks in barrel outer tracker (12 hits in total)
- This kind of track candidate has the benefit of not relying on the pixel track seeds
- Efficiencies reported today will be for these tracks unless otherwise noted

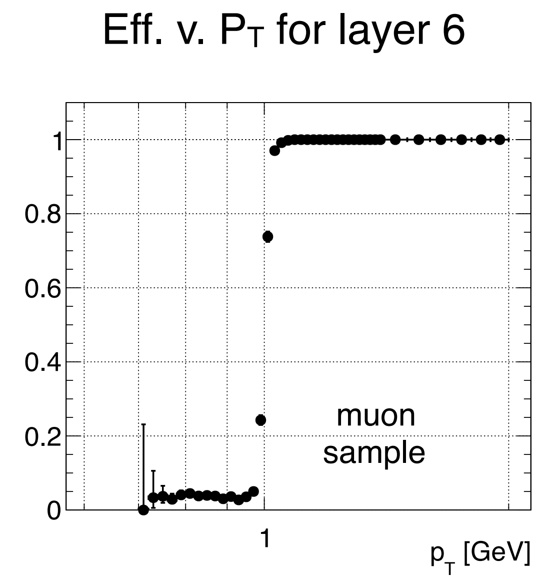


Today we target tracks with no missing hits in outer tracker barrel

Mini-doublet building and efficiency



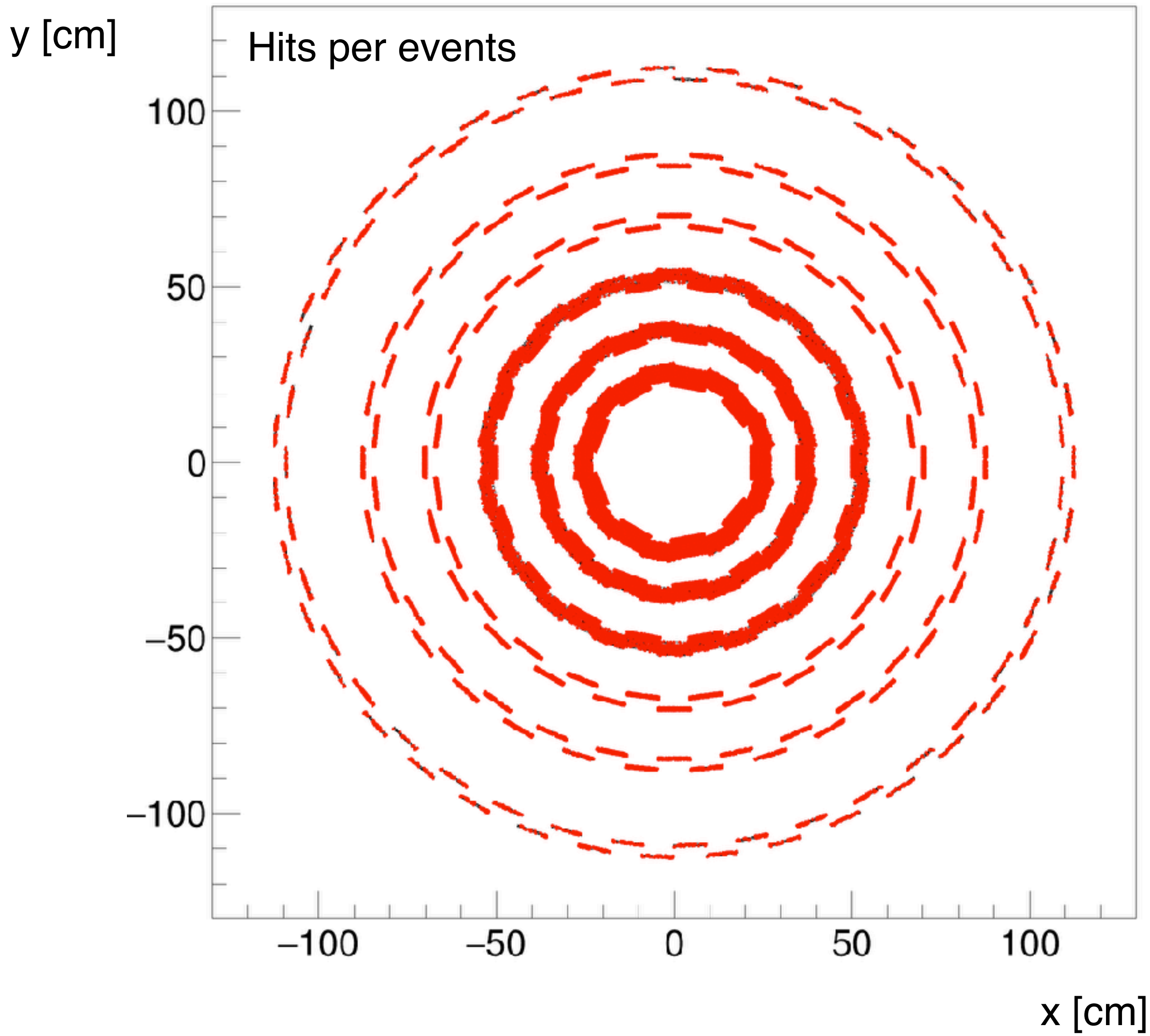




- Utilize P_T module and build mini-doublets out of individual hits
- Target P_T threshold is 1 GeV (cf. L1Track targets > 2 GeV)
- Plateaus soon after targeted 1 GeV threshold
- Only requires information on its own module ⇒ suitable for parallelization

Good mini-doublet efficiency achieved soon after targeted threshold of 1 GeV in barrel region

Effects of the P_T module on combinatorics

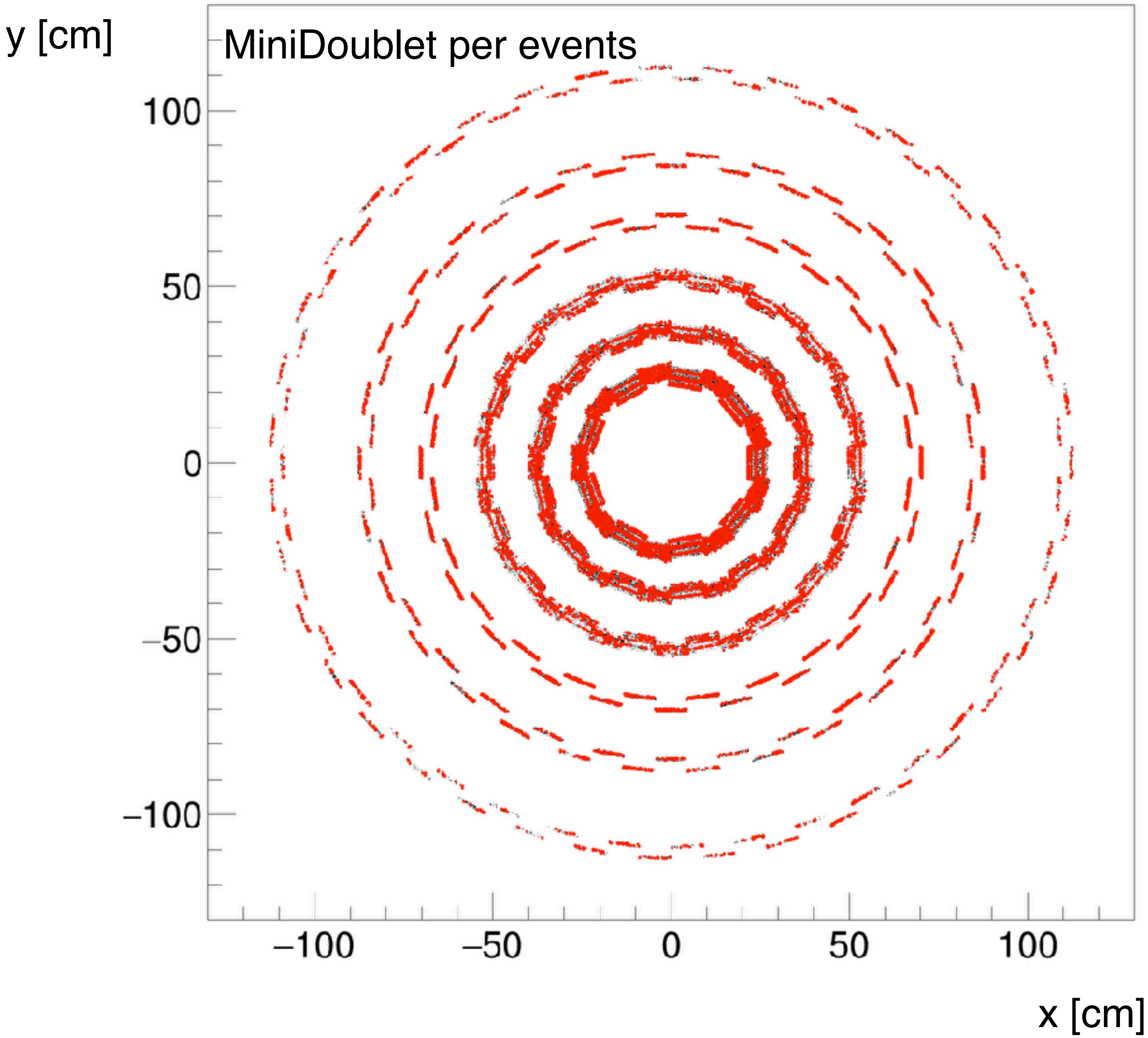


- PU200 $t\bar{t}$ event
- All hits in the barrel detector plotted
- Large number of hits per layer
- Naive combinations will lead to combinatorics explosion

Layer	1	2	3	4	5	6
# hits	36K	28K	21K	17K	12K	6K

Very large hit multiplicity for PU 200 event

Effects of the P_T module on combinatorics



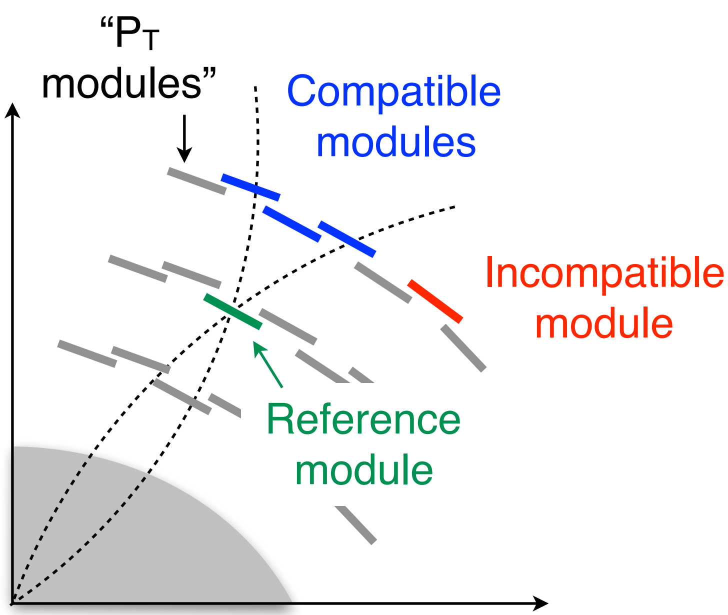
- PU200 $t\bar{t}$ event
- All mini-doublets (MDs) in the barrel detector plotted
- 3x to 7x reduction in multiplicity per layer
- ⇒ massive reduction in combinatorics in parallel

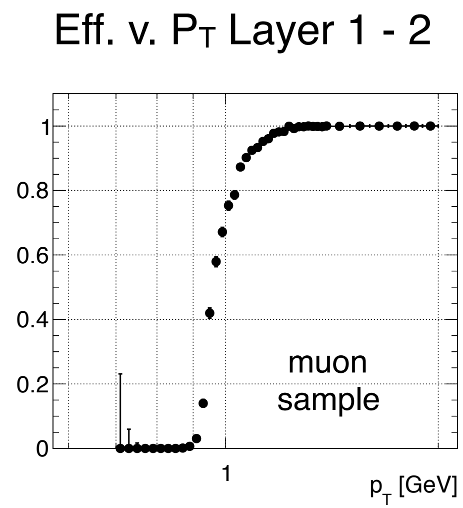
Layer	1	2	3	4	5	6
# hits	36K	28K	21K	17K	12K	6K
# MD	5.9K	3.8K	3.1K	3.7K	3.3K	2.2K
Ratio	6.1	7.3	6.8	4.6	3.5	2.7

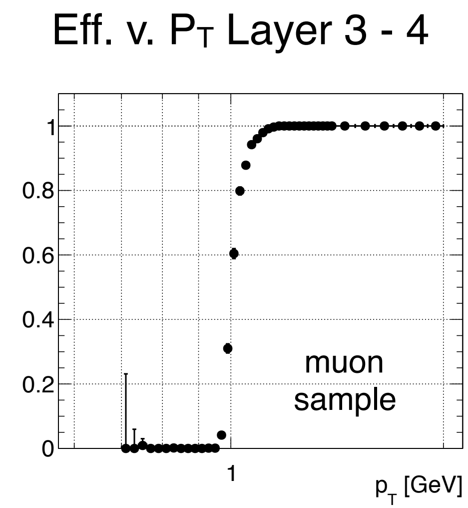
Mini-Doublet formation much reduces combinatorics

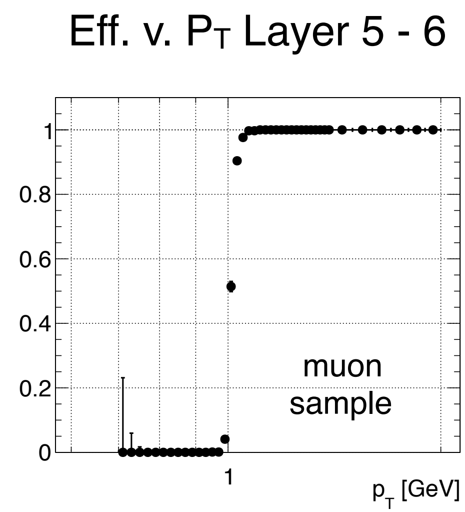
Segment building and efficiency

Chang
UCSD







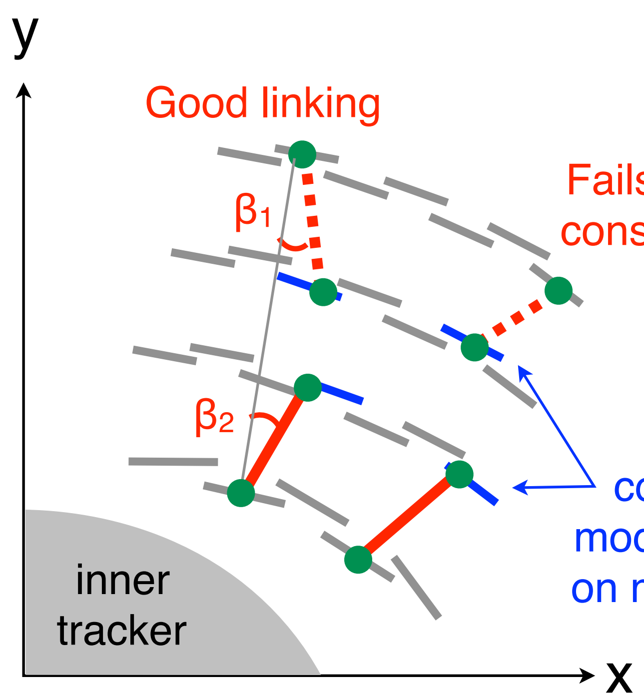


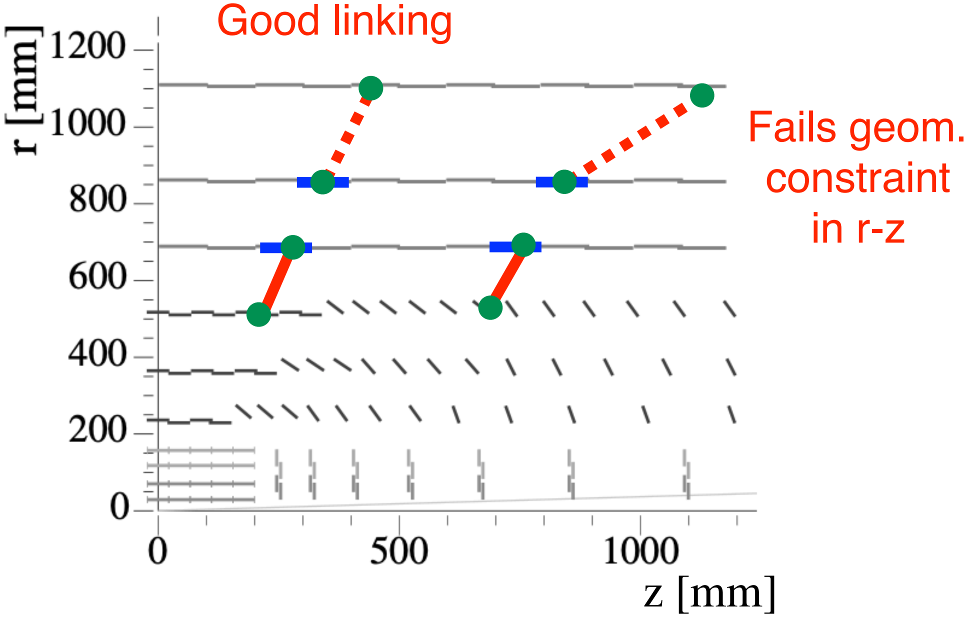
- Utilize module map built from muon gun simulations in building segments
- Compatibility between Mini-Doublet angles and segment angle required
- Same 1 GeV threshold targeted

Good segment efficiency achieved soon after targeted threshold of 1 GeV in barrel region

Segment linking (a.k.a. tracklet building)

- Consider potential segments (●····●) to link utilizing module map
- Use geometrical arguments to link segments (same 1 GeV threshold targeted)
- Various angle compatibility between segments and linked segments used

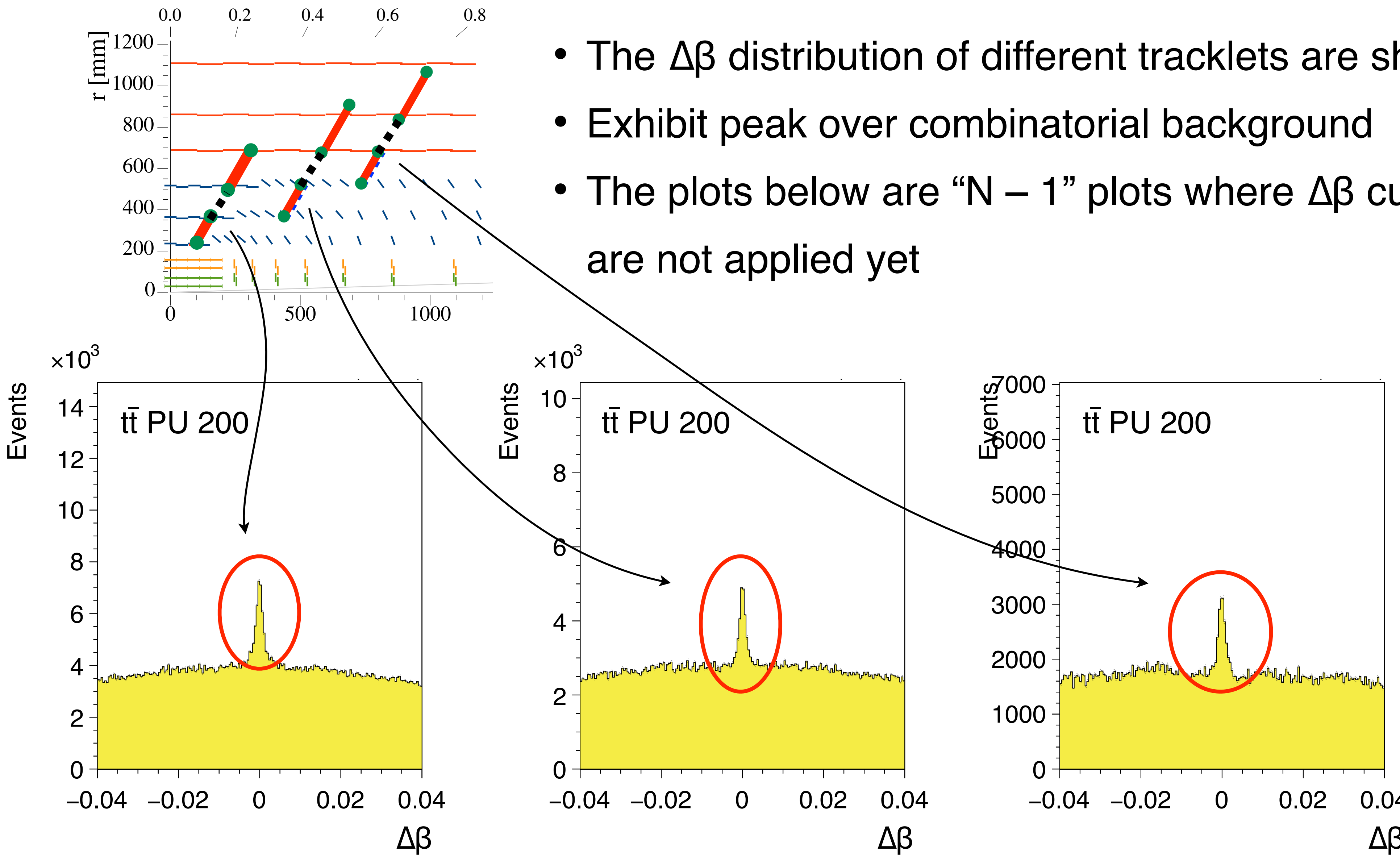




Segments are linked based on geometrical argument

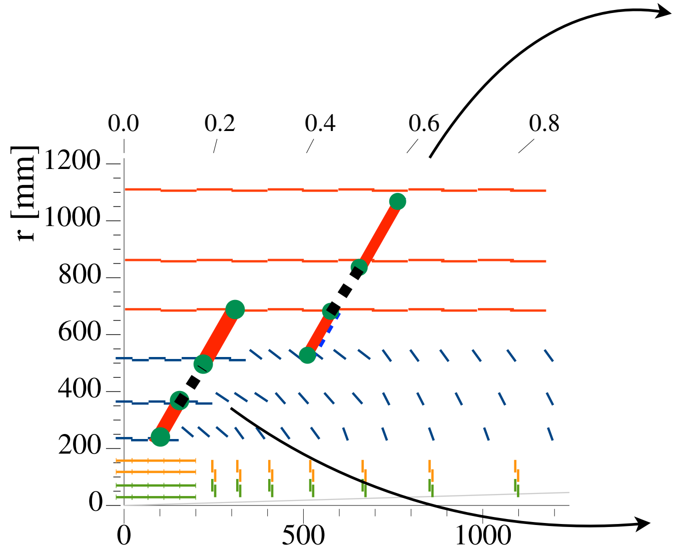
Δβ distributions

- The Δβ distribution of different tracklets are shown
- Exhibit peak over combinatorial background
- The plots below are "N – 1" plots where Δβ cuts are not applied yet

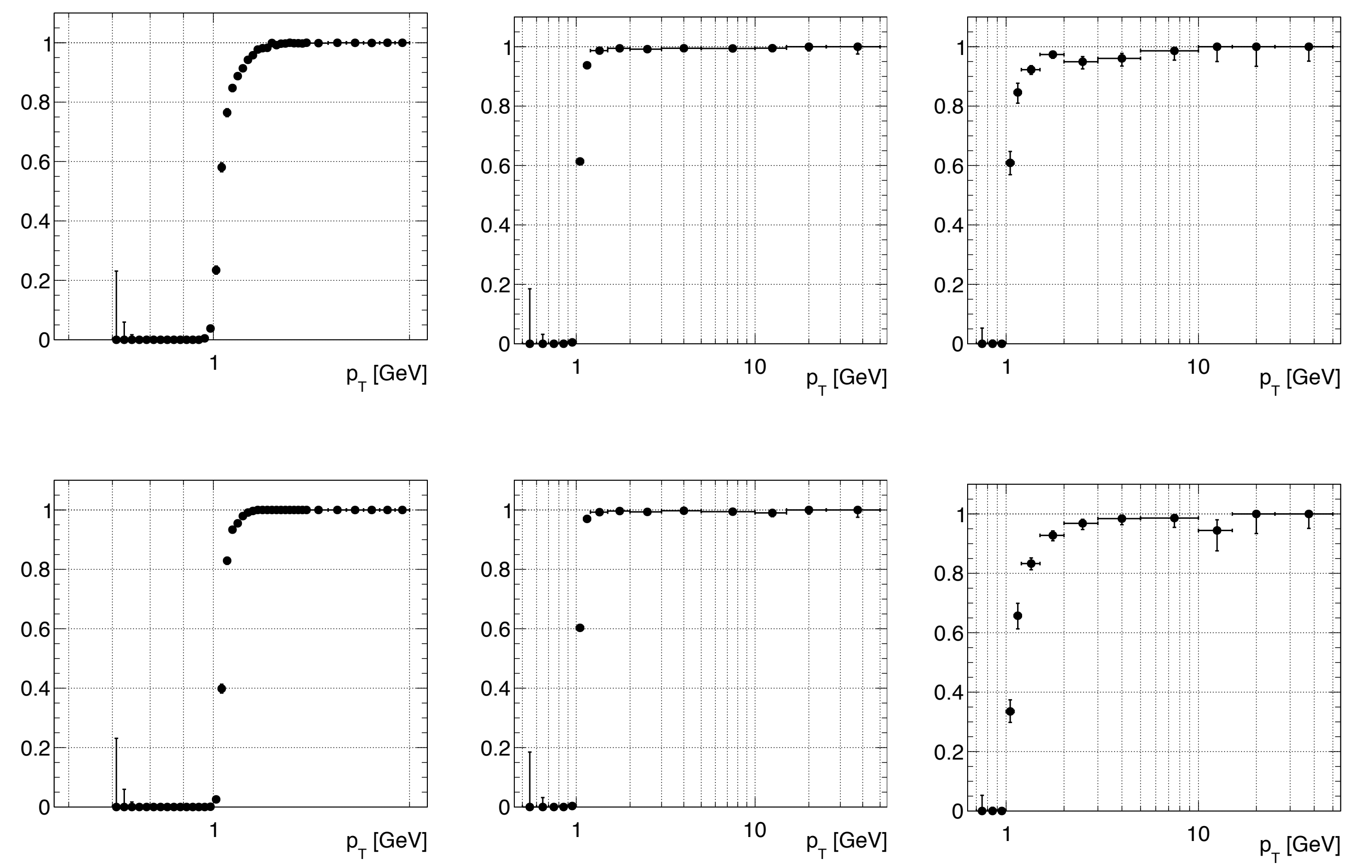


Δβ distribution shows peak above combinatorial background

Tracklet (linked segment) efficiency



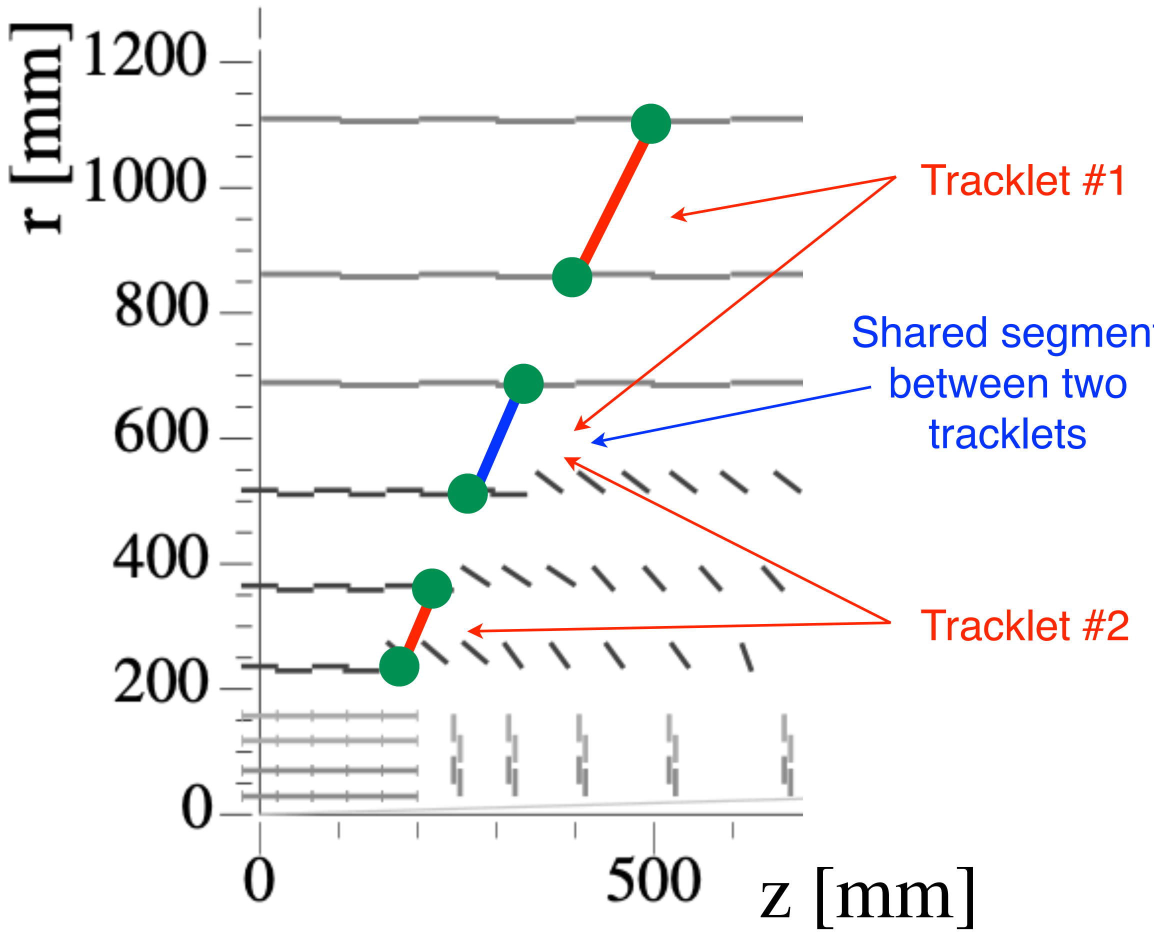
μ-Gun events | $\pi^{\pm}$ from $t\bar{t}$ events | *e* from $t\bar{t}$ events



Good segment linking efficiency of μ/e/π achieved soon after targeted threshold of 1 GeV in barrel region

Track candidate building

- Two tracklets can be put together to build a track candidate by requiring a share segment
- More ways of building track candidates will be explored for future studies

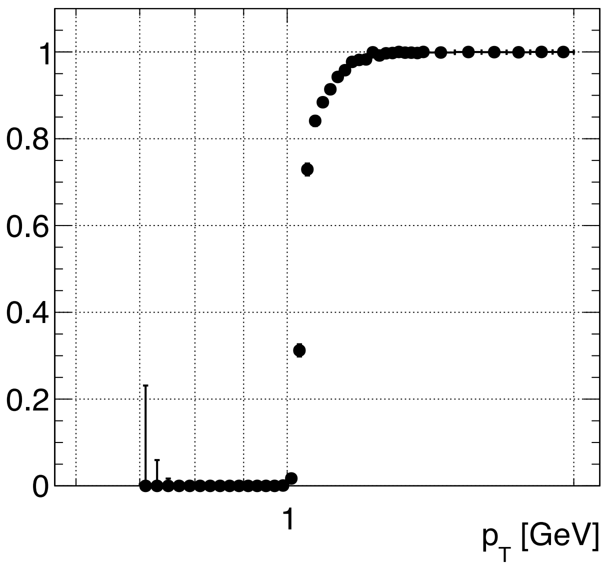


A track candidate is built from putting together two tracklets that share a common segment

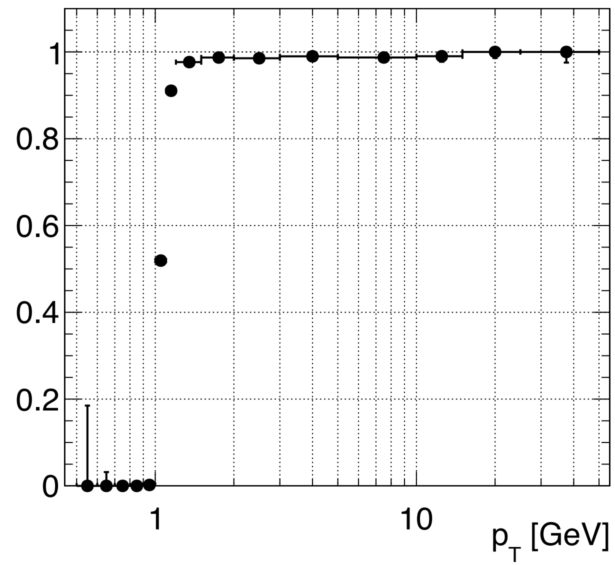
Track candidate algorithmic efficiency

- Track candidate efficiency shown
- Track candidates in barrel are composed of 12 hits from the outer tracker
- Denominator <u>only includes tracks with no missing hits present</u> in barrel

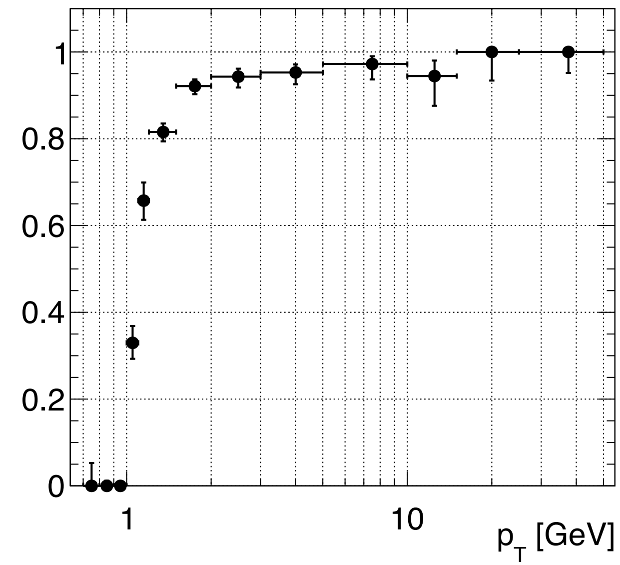
μ-Gun



π± from $t\bar{t}$ events



e from $t\bar{t}$ events



Good track candidate efficiency achieved for *μ*/*e*/π soon after targeted threshold of 1 GeV in barrel region

Summary

- As we enter HL-LHC Phase 2, reconstruction challenges are getting tougher
- We must explore new algorithms to not cost physics output
- Many promising novel algorithms are parallel in nature
- Here we studied a parallel track building algorithm for outer tracker geometry
- Benefits of the algorithms are:
    - Natural parallelization and vectorization
    - Complementarity w/ pixel based seeding (potential physics gain)
    - Also could potentially benefit software trigger for more extensive tracking
- Preliminary barrel region segment linking algorithm has been developed and created preliminary track candidates using 12 hits in the outer tracker
- Algorithm can be sped up as each step of the algorithm can be massively parallelized