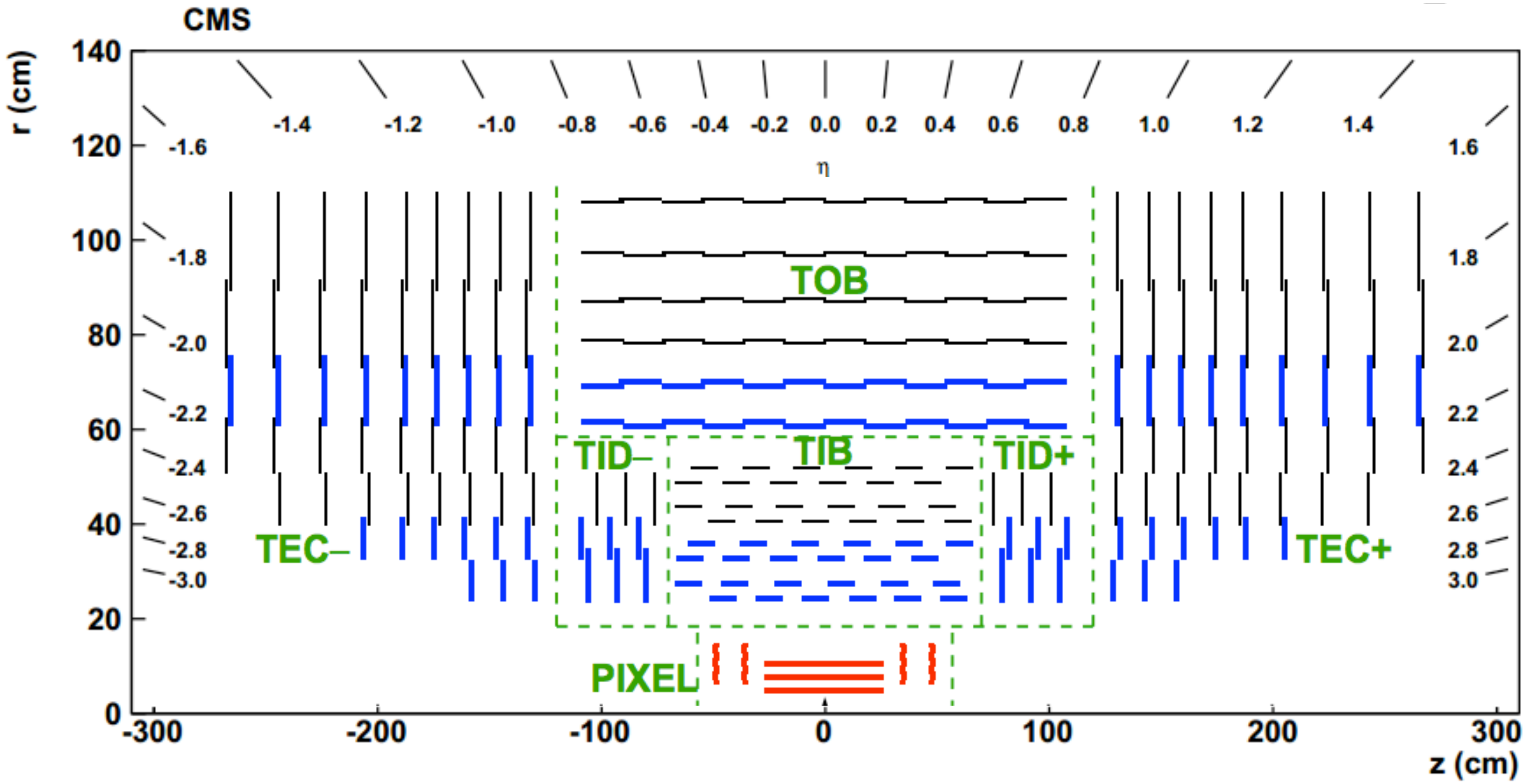
Tracking and vertexing algorithms at high pileup

Vertex 2014 – Sept 18, 2014

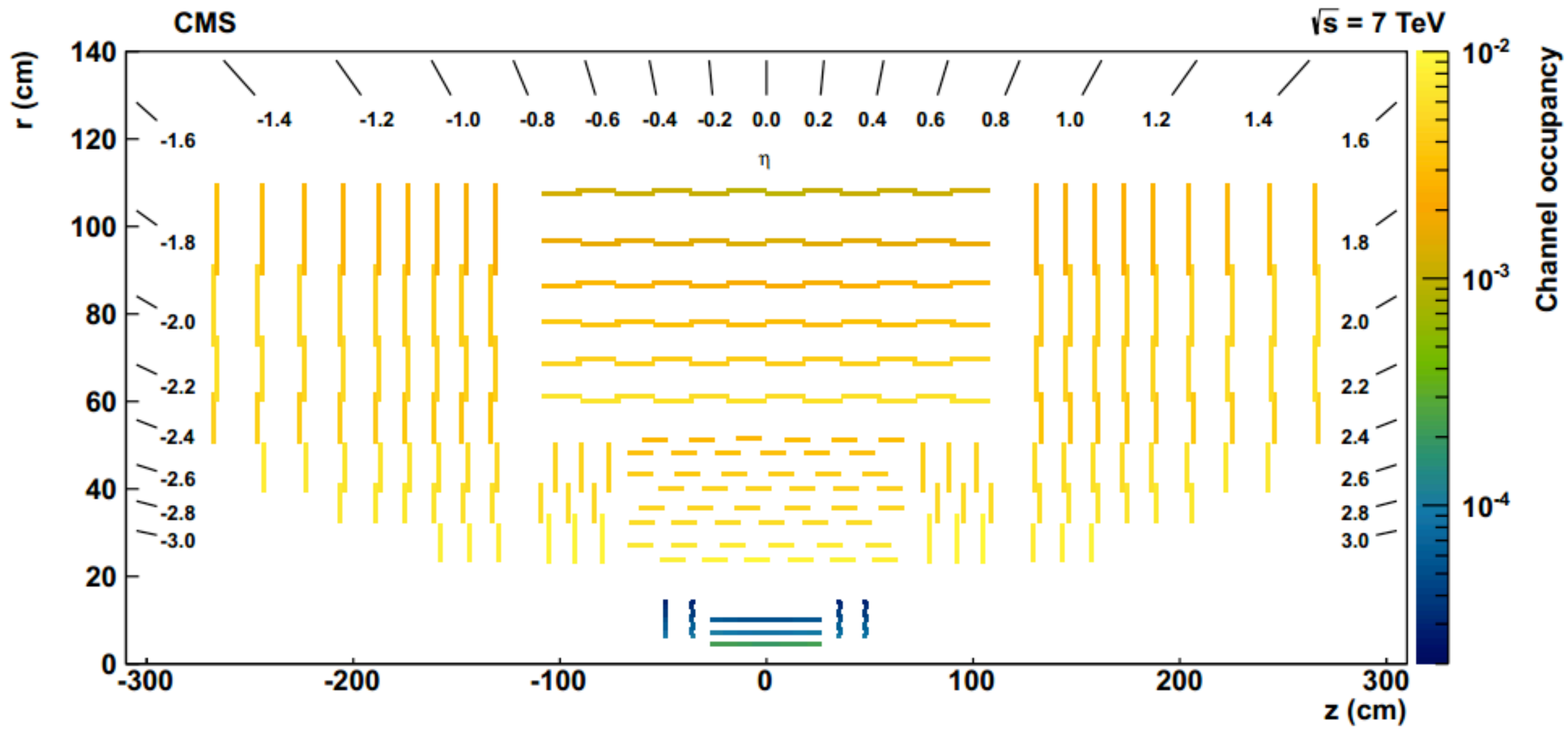
G.Cerati (UCSD)
on behalf of the CMS Collaboration

Introduction



- CMS Tracker:
  - Pixel Detector: 66M channels, **100x150 μm²** pixel
  - **Si-Strip** detector: 9.6M channels, **80-180 μm** pitch, 10-20 cm long
    - Double-sided modules glued with 100 mrad angle provide 3D position (global coordinates)

Introduction

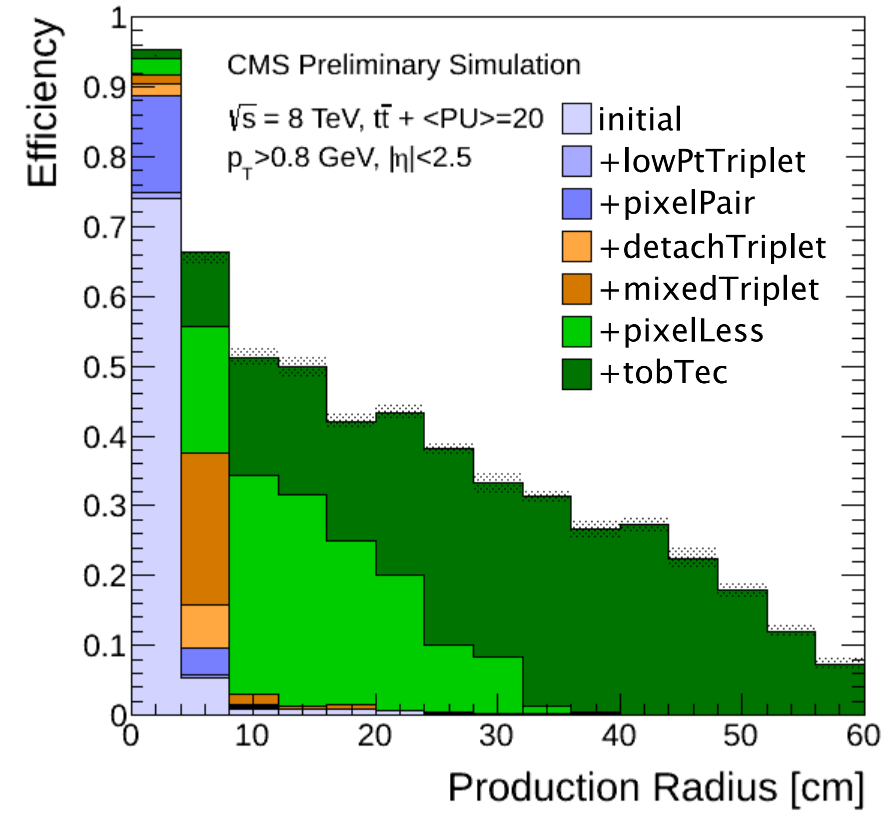


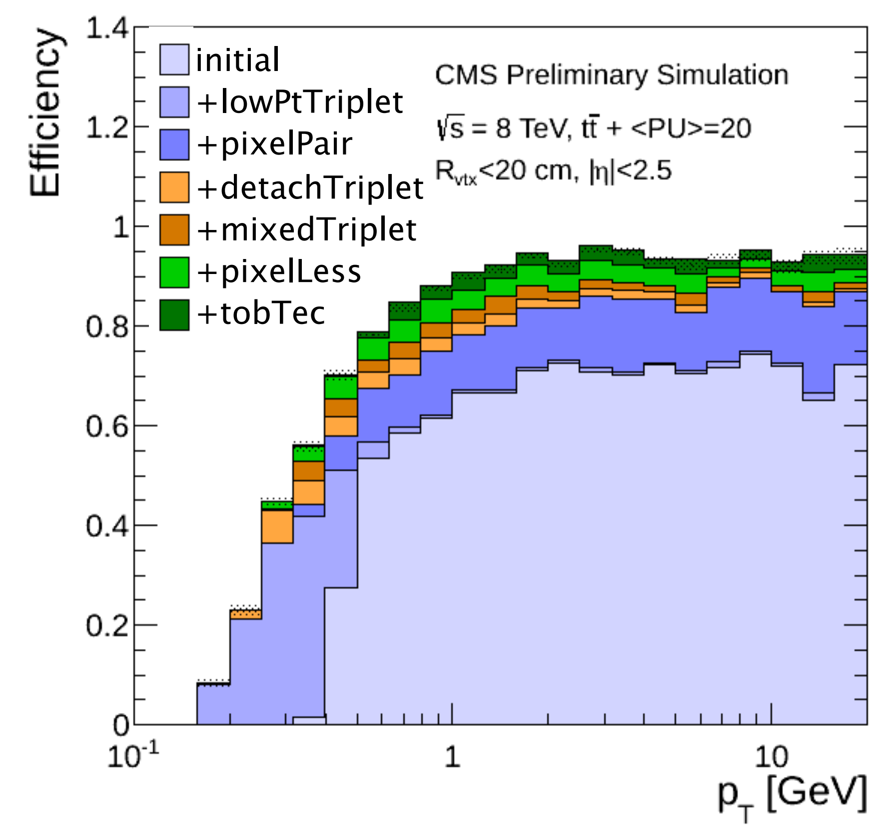
- CMS Tracker:
  - **Pixel** Detector: 66M channels, **100x150 μm²** pixel
  - **Si-Strip** detector: 9.6M channels, **80-180 μm** pitch, 10-20 cm long
    - **Double-sided modules** glued with 100 mrad angle provide 3D position (global coordinates)
  - **Occupancy** 10-100x **lower in pixel** region due to high granularity
    - inside out tracking from pixel layers
    - complemented with seeding from double sided strip layers

CMS Track and Vertex Reconstruction

- Tracking based on Kalman Filter and divided in 4 main steps:
  - seeding, pattern recognition, fitting and selection
- Procedure repeated iteratively, removing hits associated to high quality tracks ("High Purity") to reduce combinatorics
- Seven iterations used in Run1:

N	Step Name	Seeding	Target Track
0	Initial	pixel triplets	prompt, high p_T
1	LowPtTriplet	pixel triplets	prompt, low p_T
2	PixelPair	pixel pairs	high p_T recovery
3	DetachTriplet	pixel triplets	displaced--
4	MixedTriplet	pixel+strip triplets	displaced-
5	PixelLess	inner strip pairs	displaced+
6	TobTec	outer strip pairs	displaced++



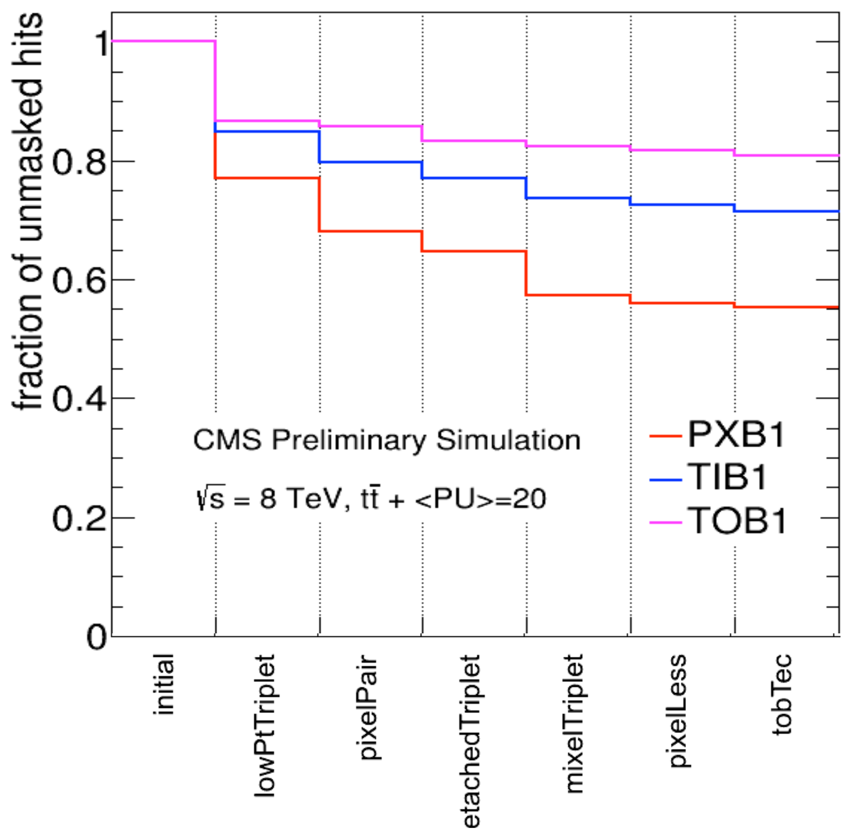


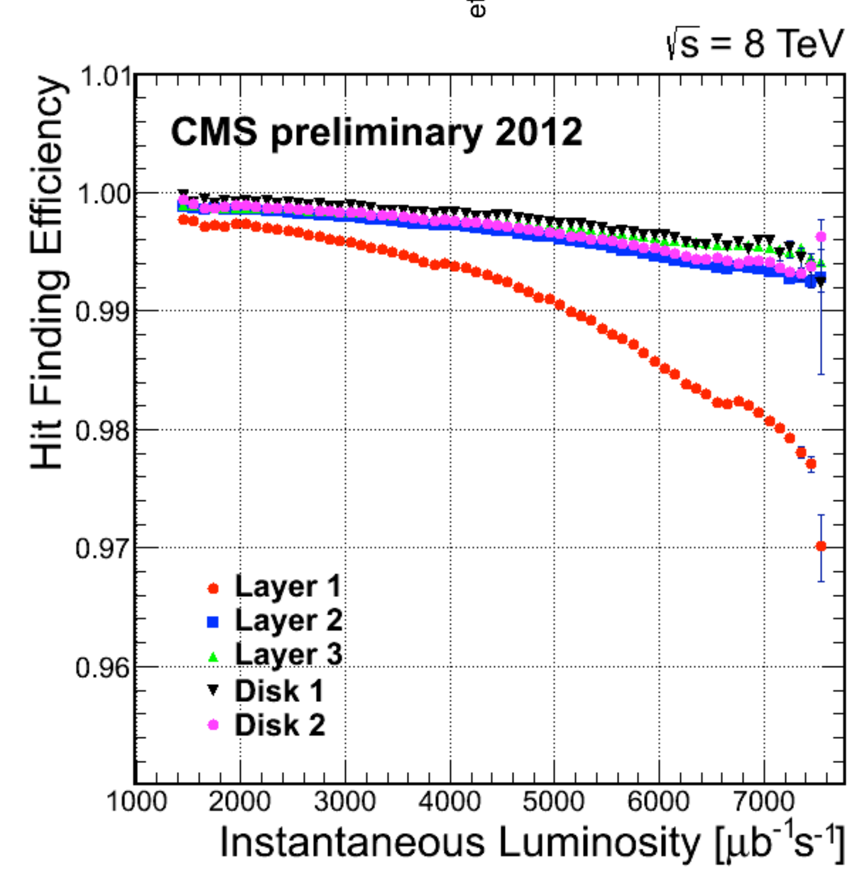
Vertex reconstruction:

- Track clustering with Deterministic Annealing
  - Resolving power Δz ~1 mm
- Adaptive Vertex fit
  - Resolution O(10 μm) both in x-y and z
- Reconstructed vertices sorted according to higher Σp_T^2

Tracking challenges at high pile-up (I)

- Tracking in high pile-up is a **challenge** due to increasing **tracker occupancy**
- Pile-up **scenarios for Run2**:
  - Startup at **50ns** bunch crossing (bx), collect up to ~1/fb, **PU~25**
  - Ramp up at **25ns** bx, collect up to ~9/fb with **PU~25**
  - Stable running at **25ns** bx, up to ~9/fb with **PU~40**
- **Iterative tracking is not the definitive solution**, tracker is far from being empty after all iterations
- **Pixels** are affected by **dynamic inefficiency**, mainly due to saturation of chip readout buffer.

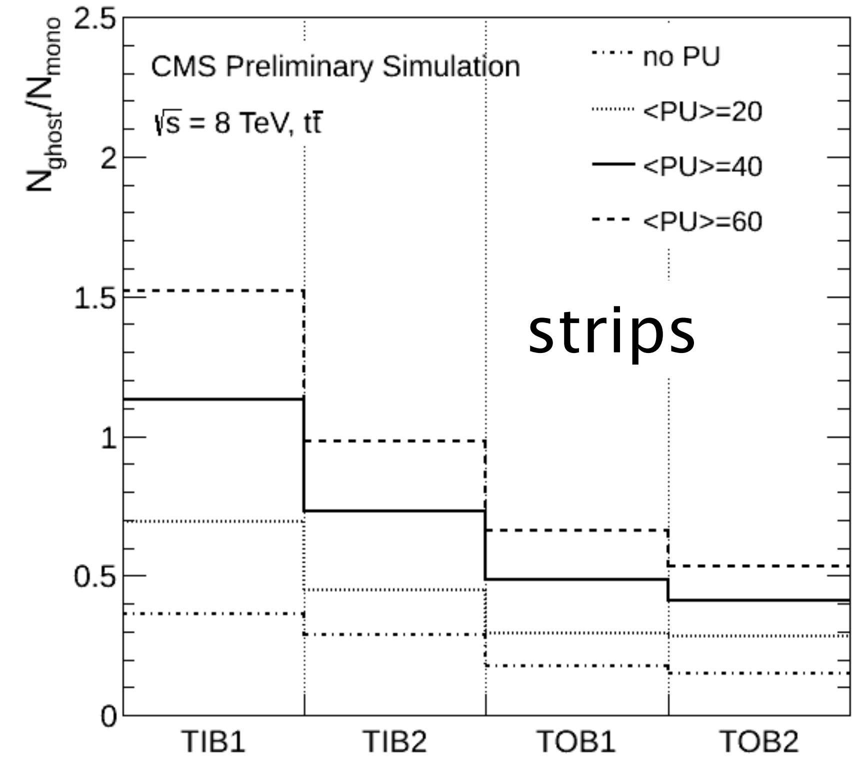




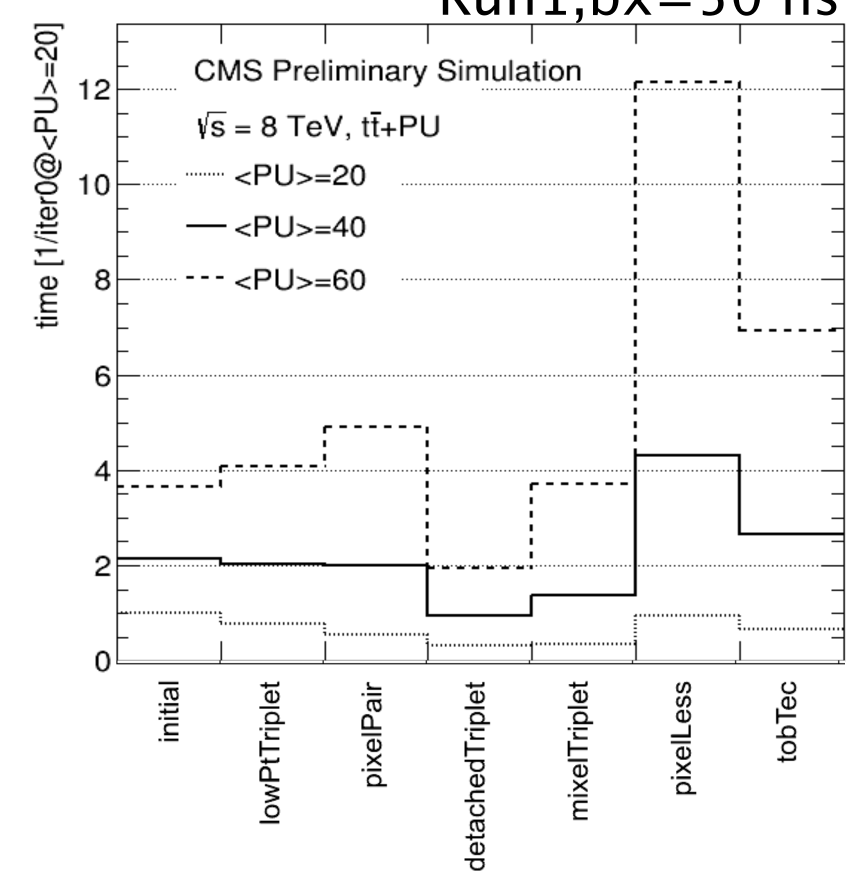
Tracking challenges at high pile-up (II)

- With 25 ns bunch crossing, out of time pile-up **increases the occupancy of the strip detector by ~45%**
  - only ~5% for pixels
- Even worse, on double-sided strip layers the number of **ghost hits** increases and in TIB1 becomes larger than true hits at <PU>=40
  - ghost hits are due to ambiguities when more than one track crosses a glued detector
- As a consequence, the effect of pile-up is dramatic on **iterations seeded by pairs of strip matched hits** (PixelLess and TobTec)
  - still problematic for steps seeded by PixelPairs and MixedTriplets
  - **pixel triplet seeded steps are linear** (Initial) or close to linear (LowPtTriplet, DetachedTriplet) with respect to pile-up

bx=50 ns

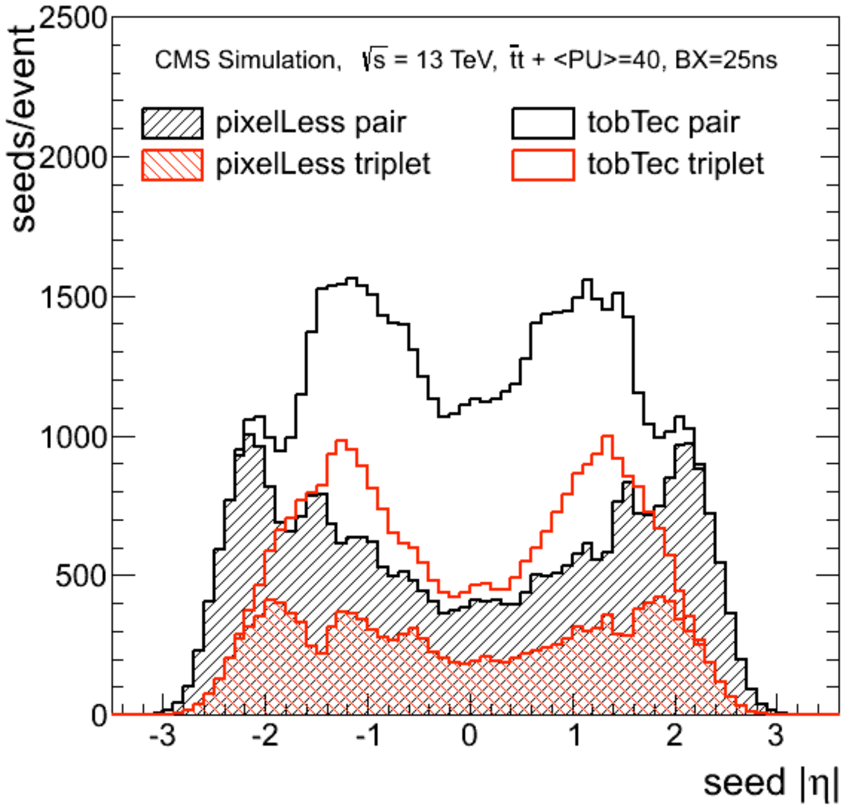


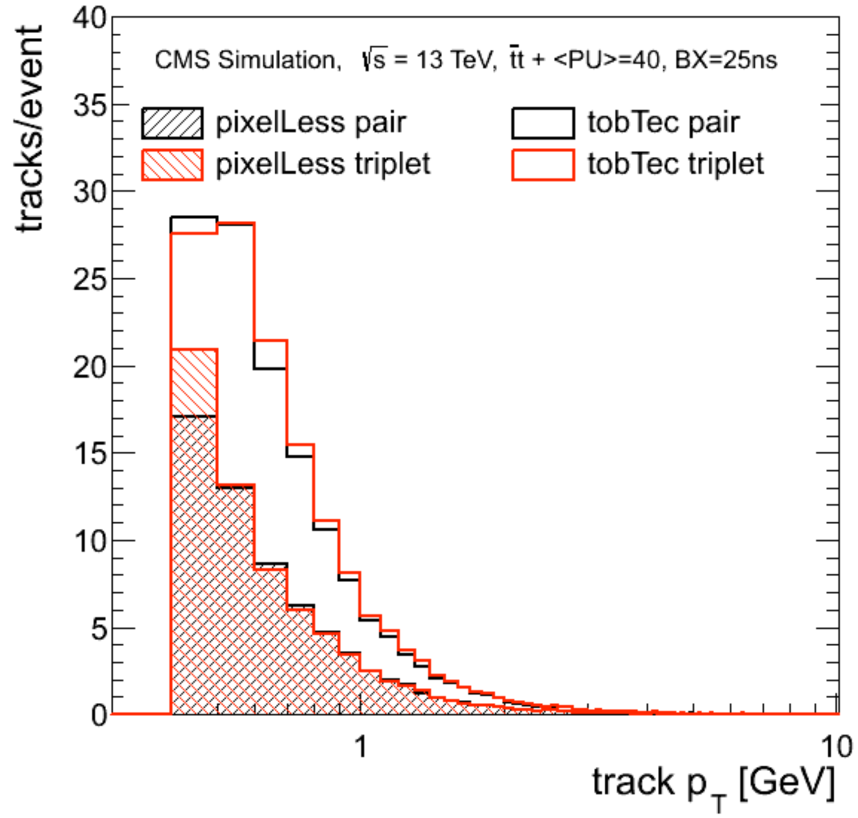
Run1,bx=50 ns



Tracking Developments for Run2

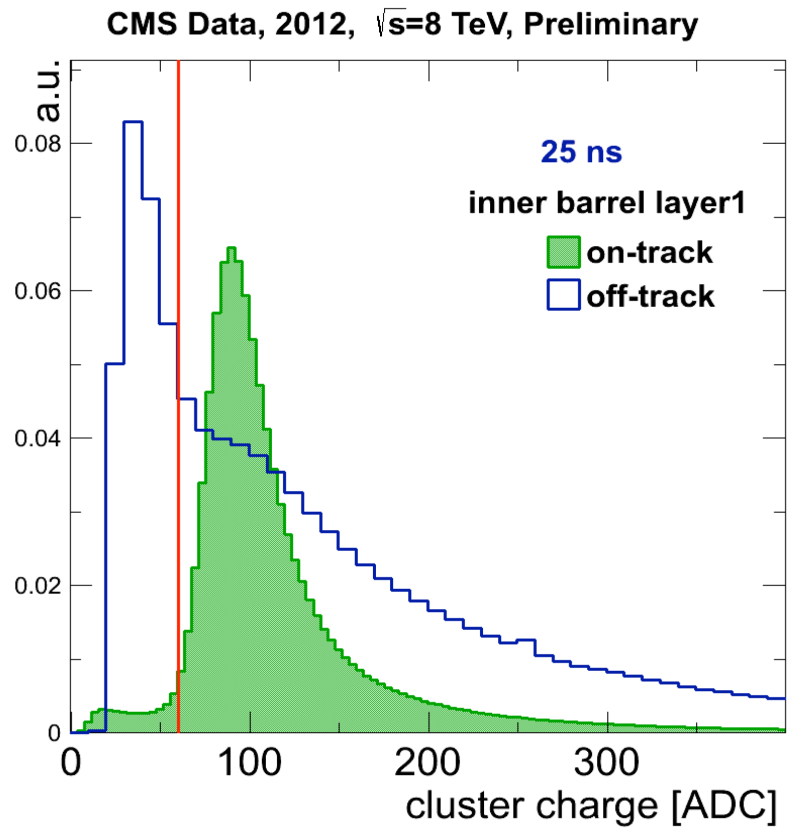
Triplet-based Seeding from Strips

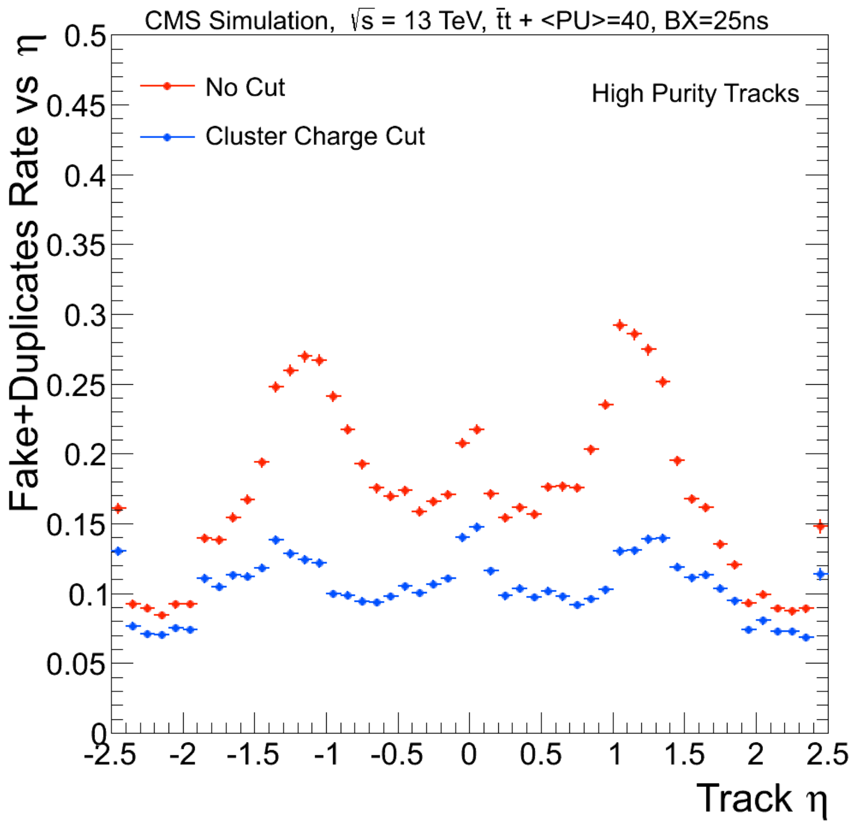




- Seeding is key for robust tracking against PU: new algorithm for strip-seeded steps
- Main feature is the **χ2 cut from straight line fit** of 3 points in the RZ plane
- From each pair at most one triplet can be produced:
  - if only one matching 3rd hit is found, the triplet is used as seed
  - if more than one matching 3rd hit is found on the same compatible layer, the pair is used as seed
  - if more than one matching 3rd hit is found in different layers, the triplet with best chi2 is used
  - if no third hit can be found, the pair is discarded
- **Tighten beam spot constraints** as much as possible with no efficiency loss
- Effective in **rejecting half of the seeds**, reconstructing the **same number of tracks**

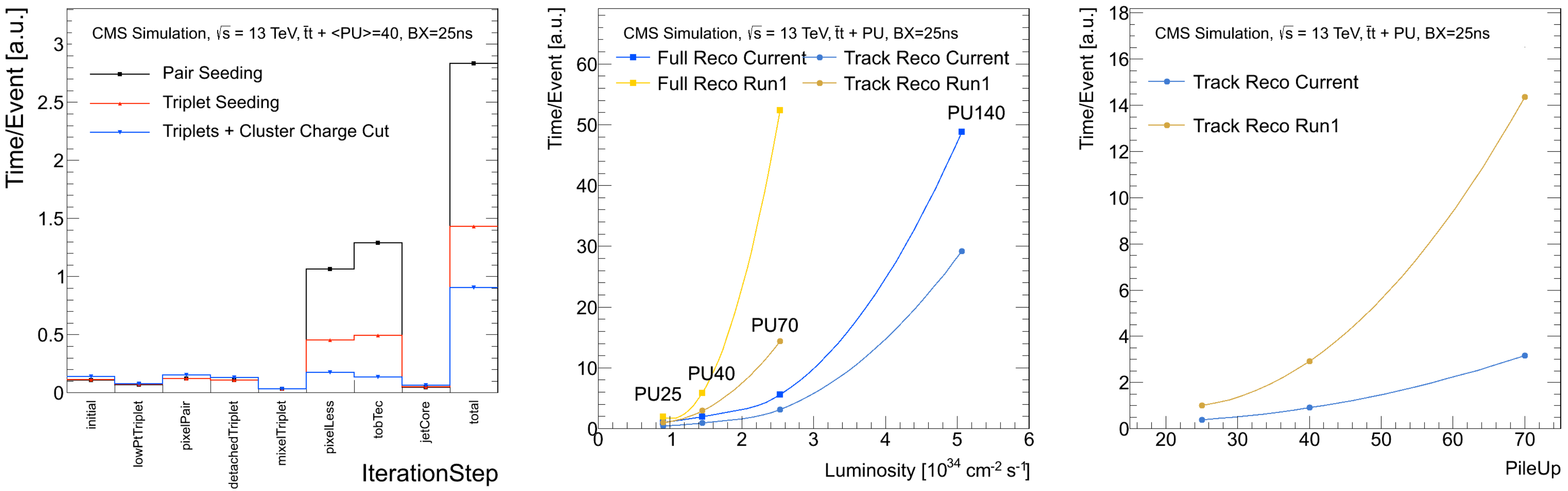
Strip Cluster Charge Cut





- With **25 ns bx**, the increase in occupancy for the strip detector induce an increase by **2x both on timing and fake rate**
- Clusters from out of time pile-up are characterized by low collected charge
  - loopers arrive at random time with respect to bx (or tail effects on charge collection)
- **Cutting on the cluster charge suppresses the effect**
  - can be applied upfront track reconstruction, during seeding or during pattern recognition
  - accounts for **sensor thickness** and **trajectory crossing angle**
    - p_T dependent cut to preserve potential signal from fractional charge particles
- Stable performance ensured by **gain calibration** in Prompt Calibration Loop
    - data vs MC and data vs time within 5% during Run1

Timing



- Both new seeding and cluster charge cut **reduce timing** of PixelLess and TobTec iterations by a **factor 2x**

- **Benchmark timing and physics performance** across releases and for different pile-up
  - TTbar samples with realistic alignment and calibration conditions
- PU scenarios:
  - BX=25 ns, <PU>=25, 40, 70, 140
  - BX=50 ns, <PU>=25
- Iterative tracking **time reduction** (for BX=25 ns):
  - **2x at PU=25, 3x at PU=40, 4x at PU=70**

More Developments

- Further **time gain** from tracking **optimizations**:
  - Global optimization of iterative tracking:
    - Reorder of iterations: moved iter3 right after iter0 – faster first!
    - Remove redundancy: reduce layer combinations for seeding, harmonize selection criteria
    - Stable performance, about 10% overall gain
  - Speedup from code optimizations

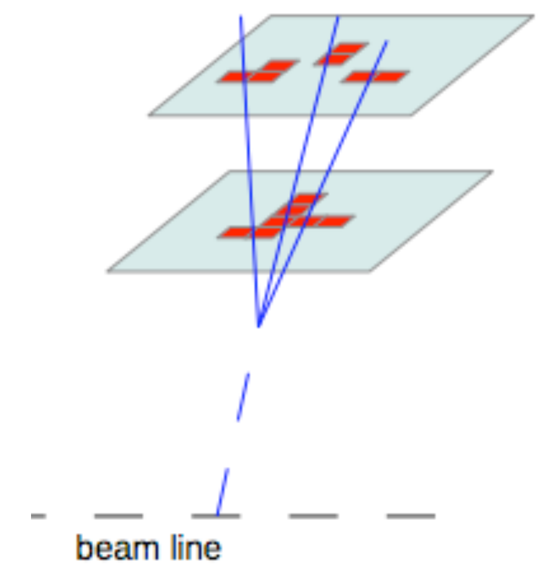
- Pixel **dynamic inefficiency** recently included in **simulation**
  - At PU40, **efficiency reduction** is at (a few) **percent level**
    - Almost no difference in fake rate
  - Iterative tracking is **robust** against detector inefficiencies:
    - **less** tracks are reconstructed with steps seeded by **pixel-triplets**
    - partially **recovered by other iterations**

- Use a **multivariate** approach for **track selection**
  - Variables: Number of layers, lost hits, $\chi 2$, η, relative p_T error, number of hits, ...
  - Higher efficiency for **low p_T and displaced tracks**
  - **Reduce fake rate** in transition and forward region
  - First version already in production, can be further improved before data taking

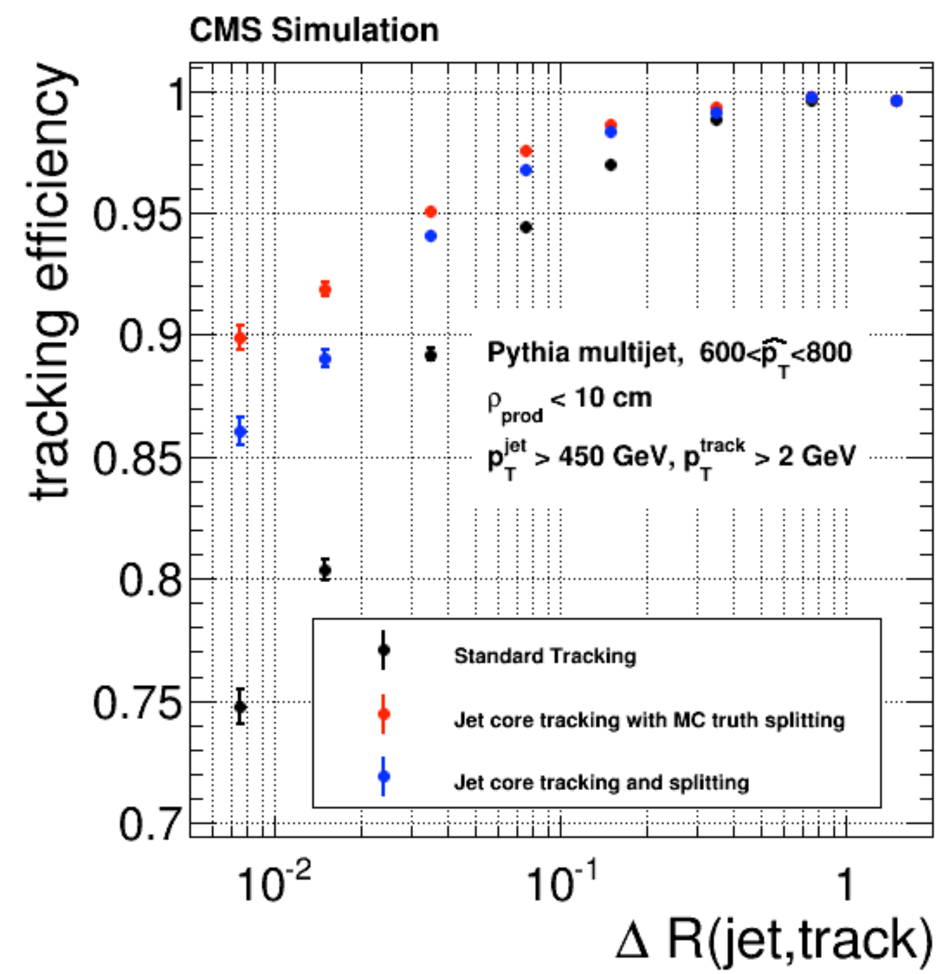
Tracking in High p_T Jets

- Tracking timing solved, focus on improving **physics performance** with tracking iterations dedicated to **specific objects**

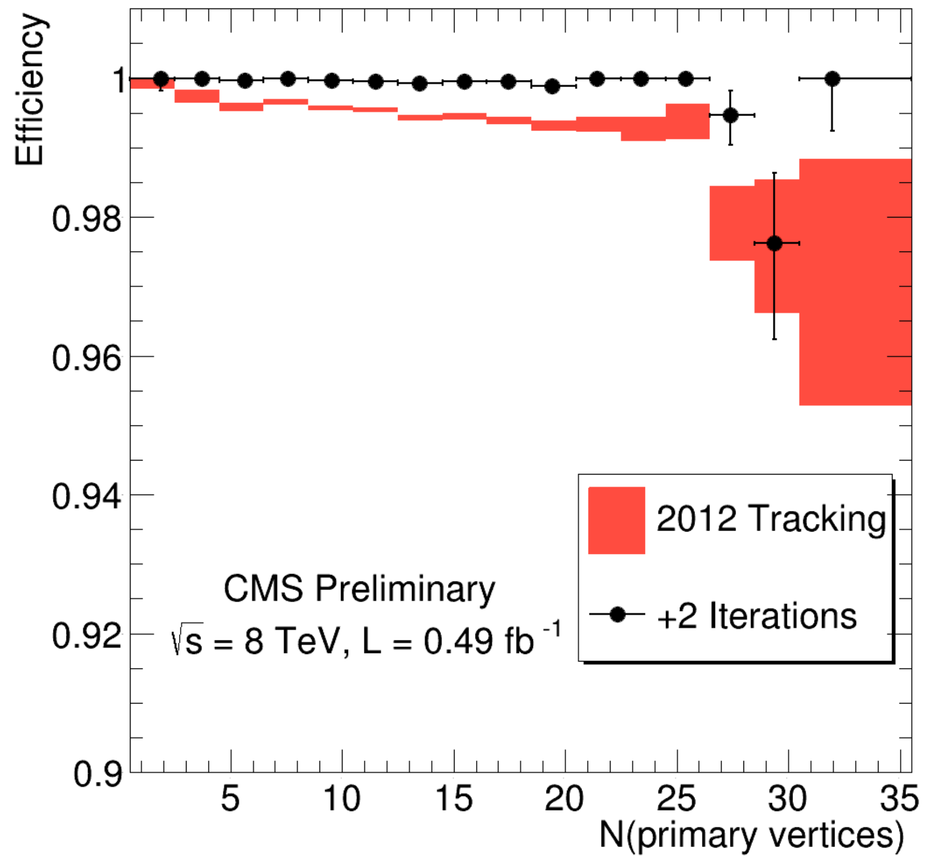
Merged clusters in b-jets

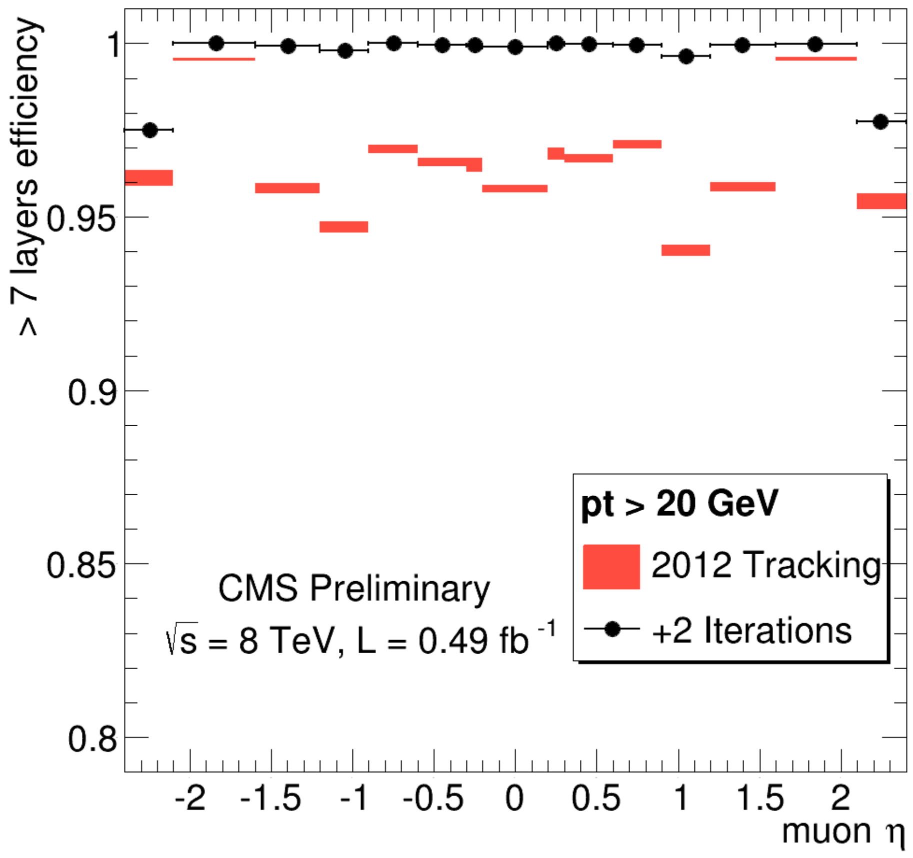


- Tracking in high p_T jets is crucial to keep high **b- and τ-tagging** efficiency
- Dense environment:
  - **small two-track separation**
  - **merged clusters**
    - only one hit with badly estimated position and uncertainty
- A **new dedicated iteration** has been developed
  - **regional**, along high p_T calo jets
    - threshold needs to be a trade-off between timing and physics
  - **looser** tracking **cuts** to follow combinatorial expansion
  - **cluster splitting** (using the jet direction as guidance)
  - **improved efficiency at small dR**
- We plan to include also new features:
  - **deterministic annealing filter**, fits tracks re-weighting close-by hits



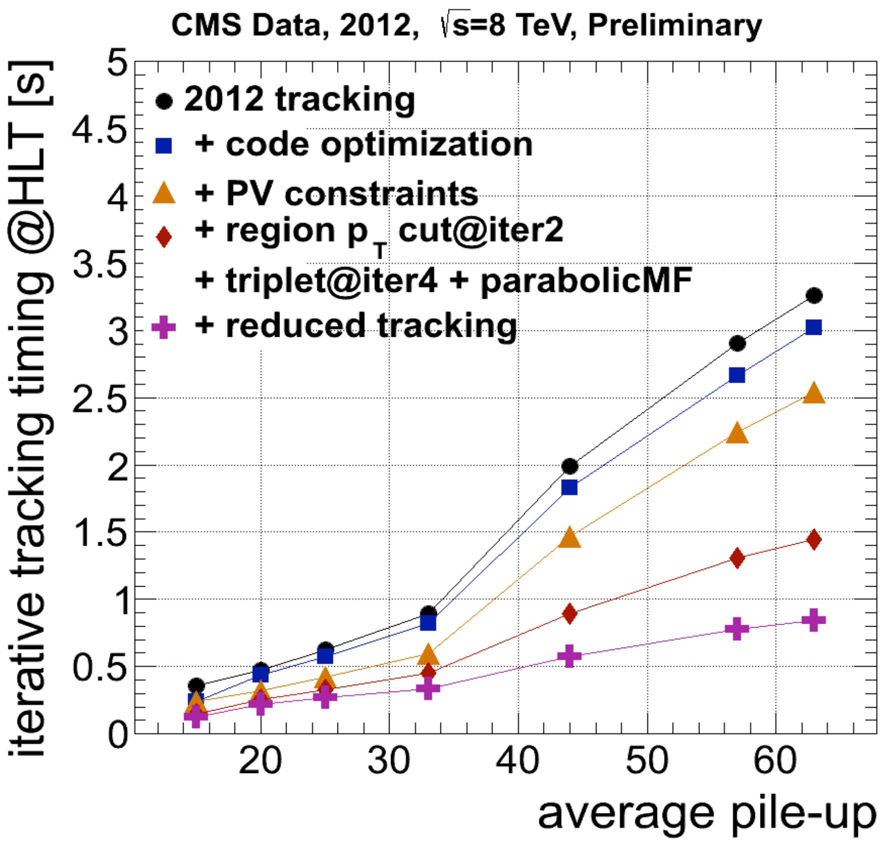
Muon Recovery Iterations

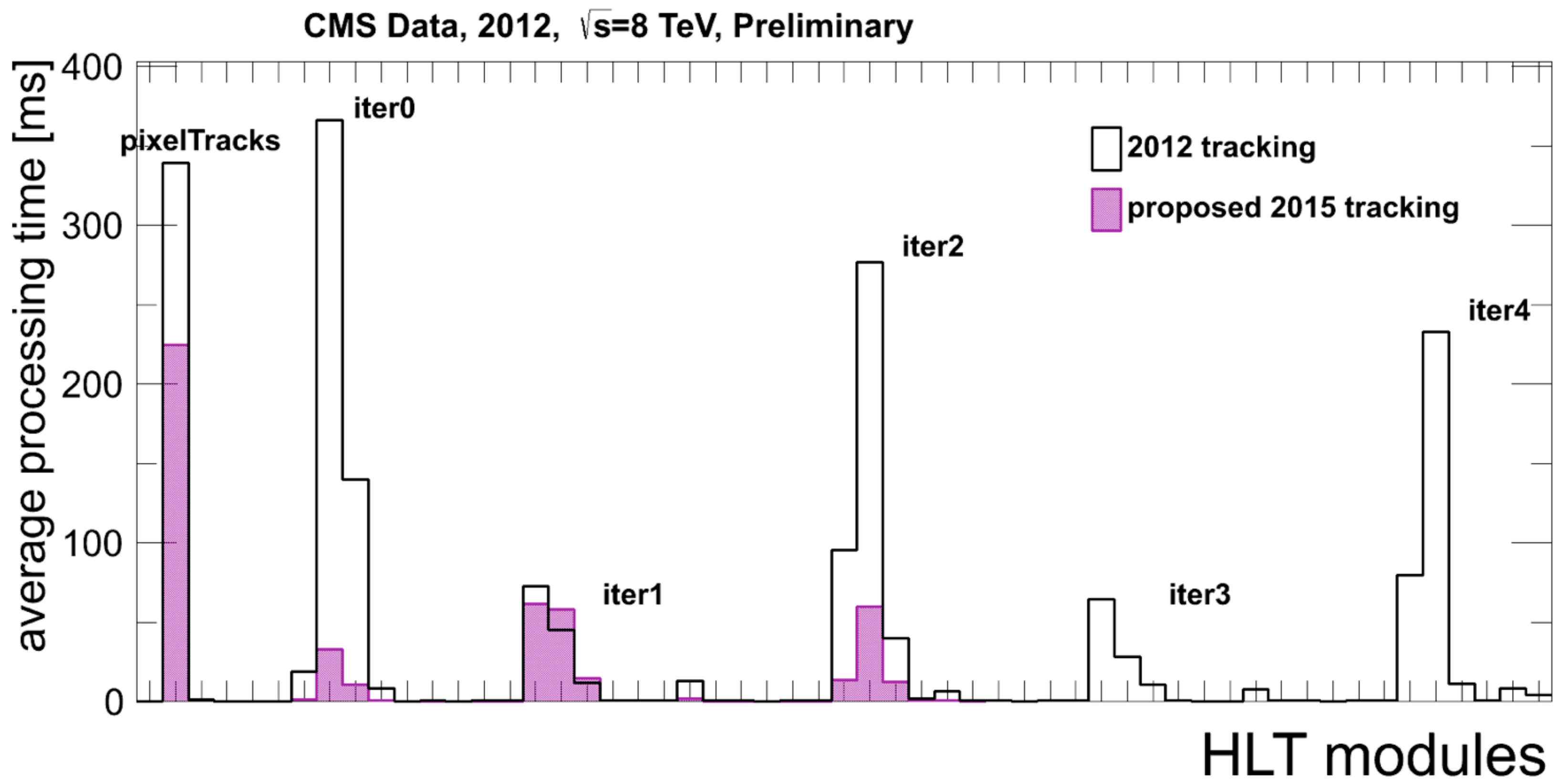




- In 2012 data it was noticed a **loss of muon reconstruction efficiency** in the tracker, increasing with pile-up.
- In order to recover it, **2 additional iterations** have been designed:
  - **Outside-in**: seeded from the muon system, recover the missing muon-track in the tracker
  - **Inside-Out**: re-reconstruct muon-tagged tracks with looser requirements to improve the hit-collection efficiency
- **Full efficiency recovered** with the new iterations

Tracking at HLT

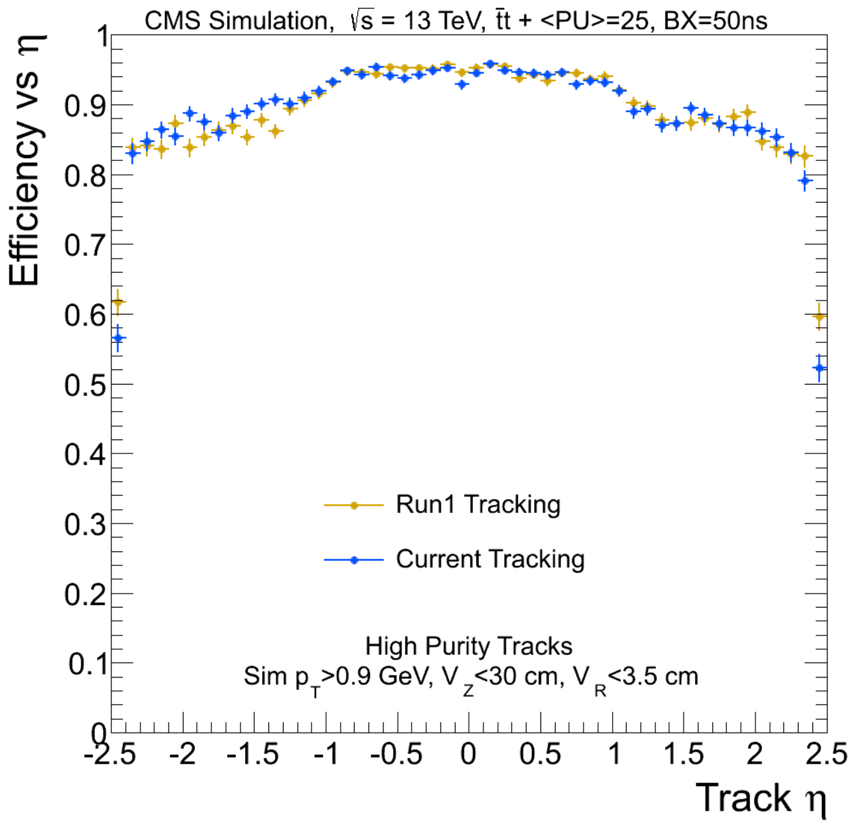




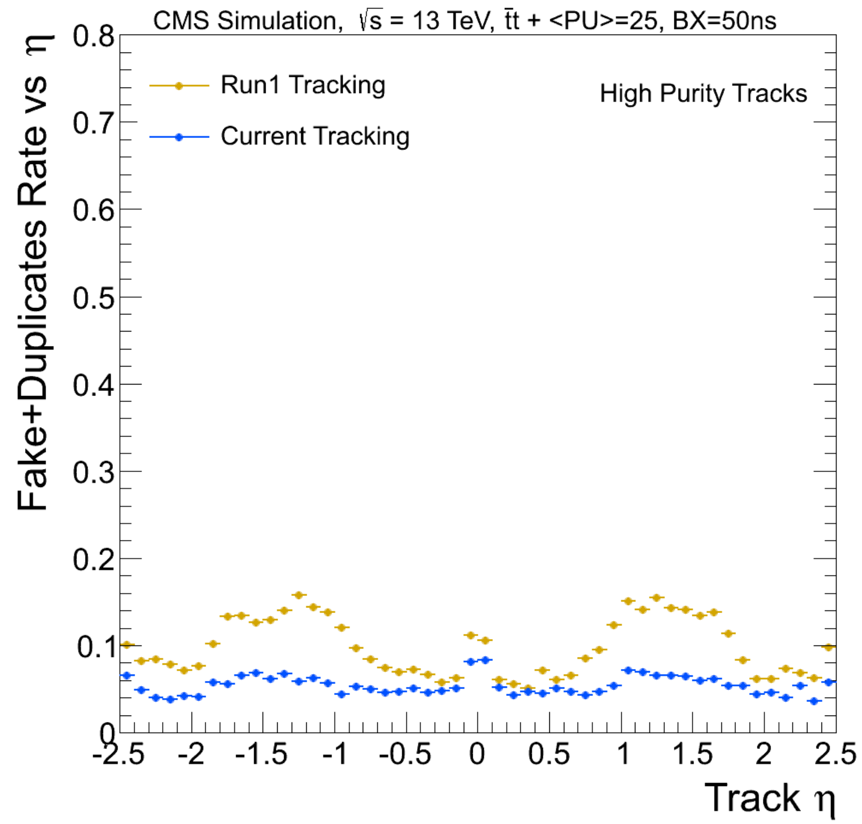
- Tracking **widely used at HLT**, crucial for keeping low rates and high efficiency wrt offline
- Timing is even more problematic at HLT: already in Run1, CMS developed a **dedicated, faster** tracking **configuration** for HLT
  - regional reconstruction around region of interest, simplified algorithms
- Run2 has higher energy, higher luminosity, higher PU: tracking needs large speed up
  - **PV constraints, higher p_T cuts, bring improvements from offline, prioritize reconstruction**
    - first reconstruct tracks with higher impact on physics (especially jets and MET)
    - drop iterations that are less important
- Achieved **4x time reduction** at PU~40 with same performance

Performance vs PU

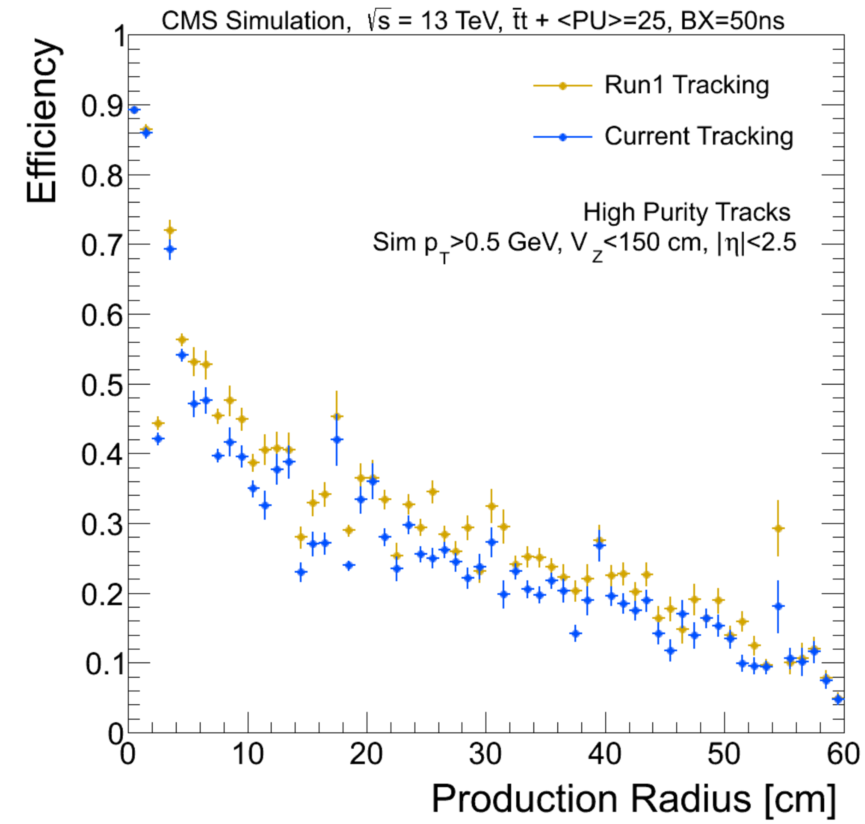
Performance Comparison with Run1-like PU



high p_T prompt tracks



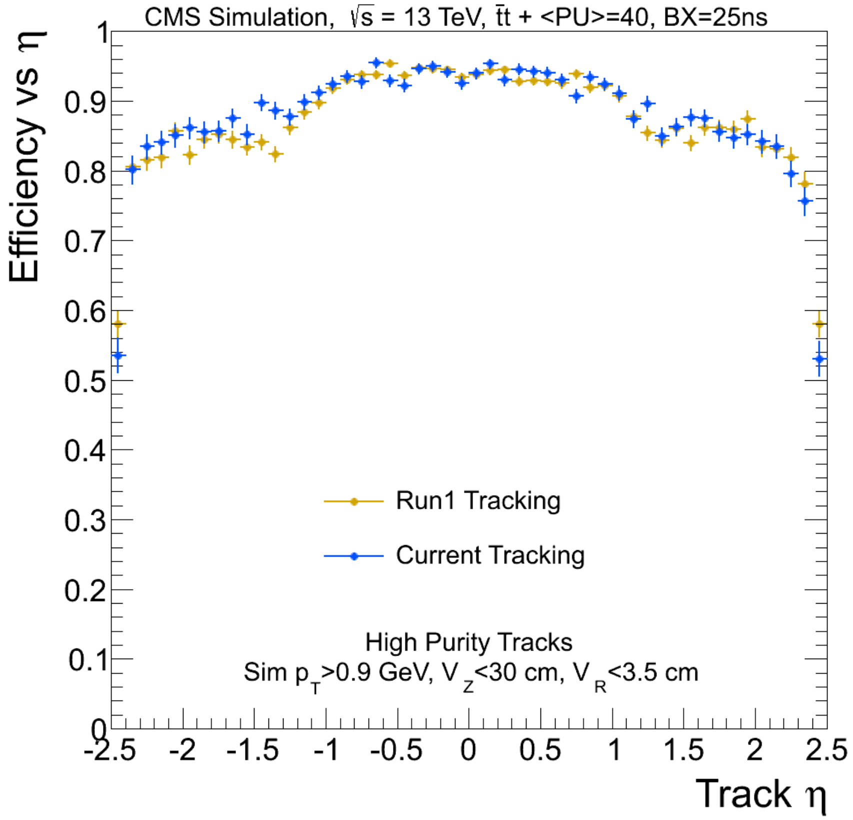
all tracks



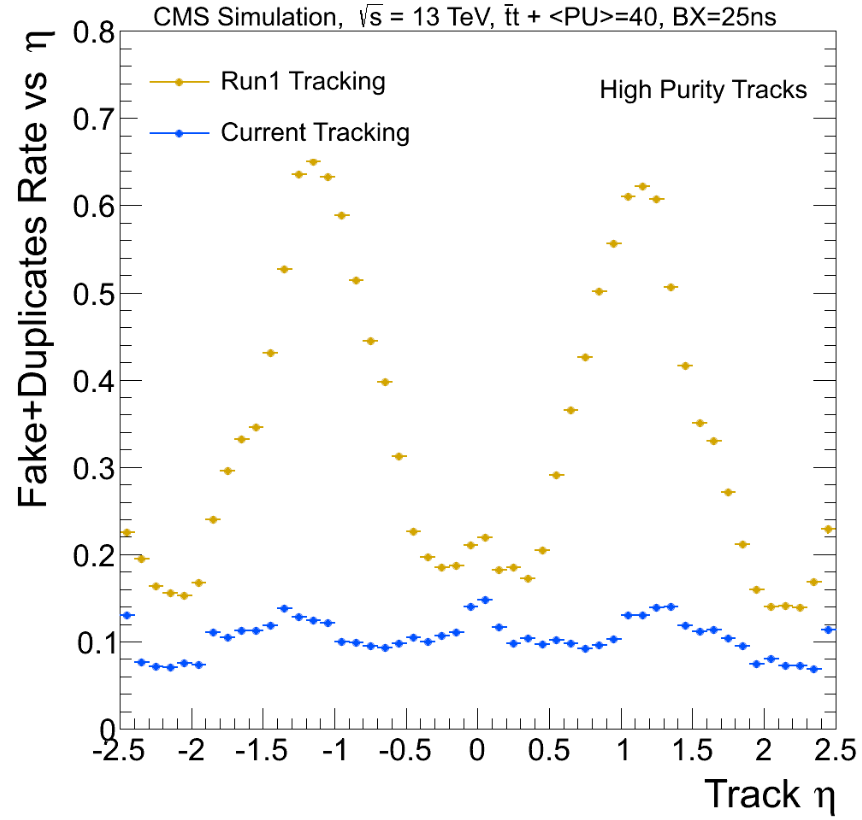
low p_T displaced tracks

- TTbar events with **<PU>=25, BX=50ns**
- Same or higher efficiency for prompt tracks
- Up to **2x** reduction in fake rate
- Slight efficiency loss for displaced tracks

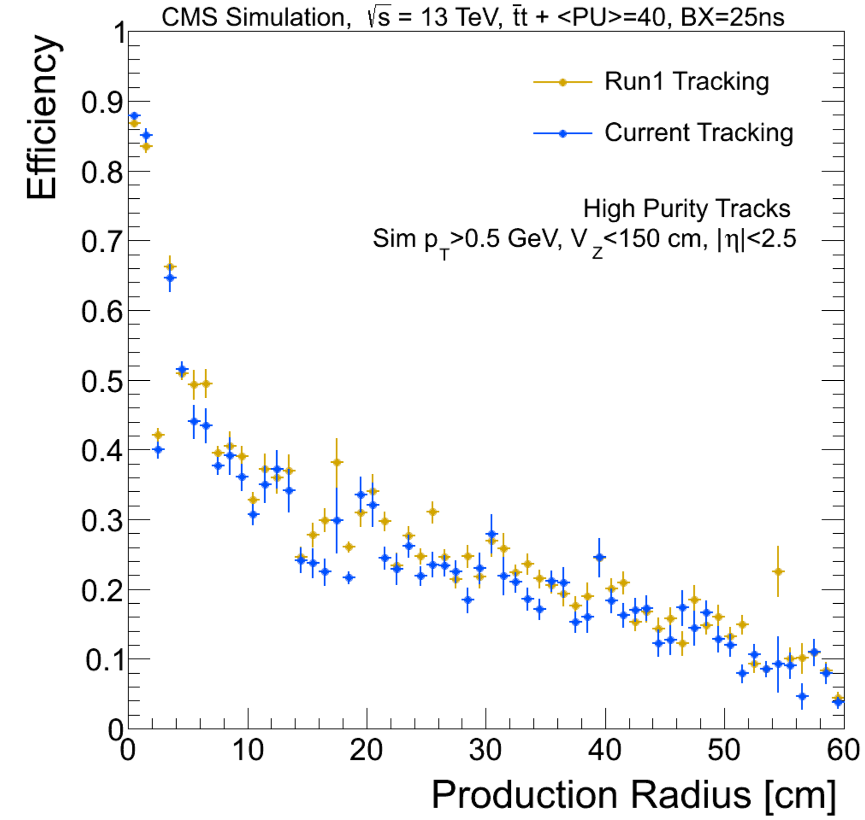
Performance Comparison with Run2-like PU



high p_T prompt tracks



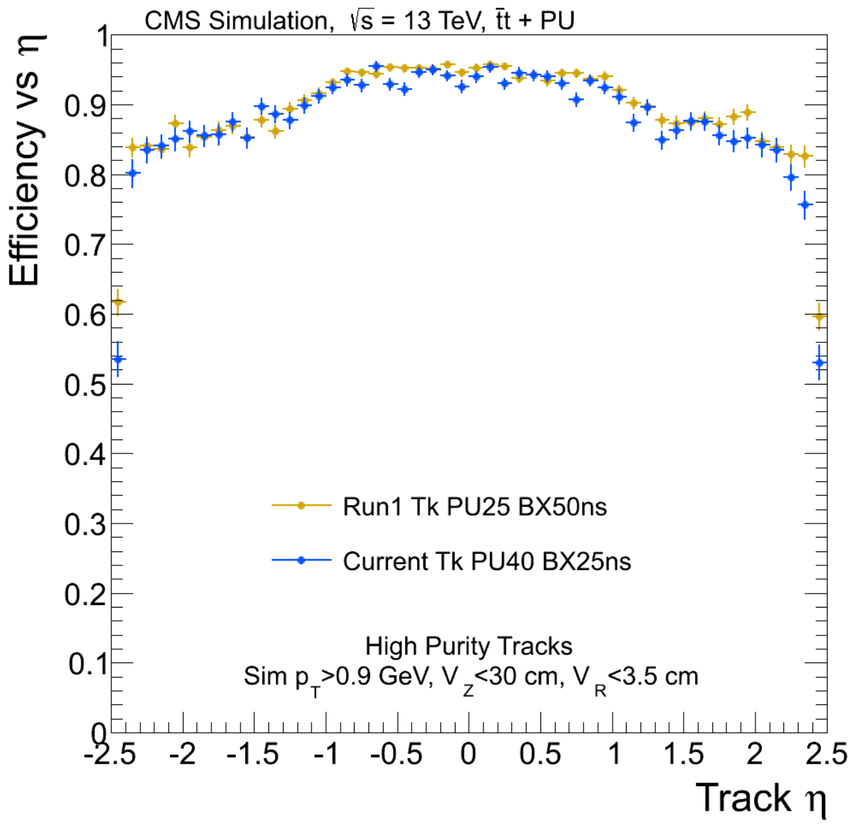
all tracks



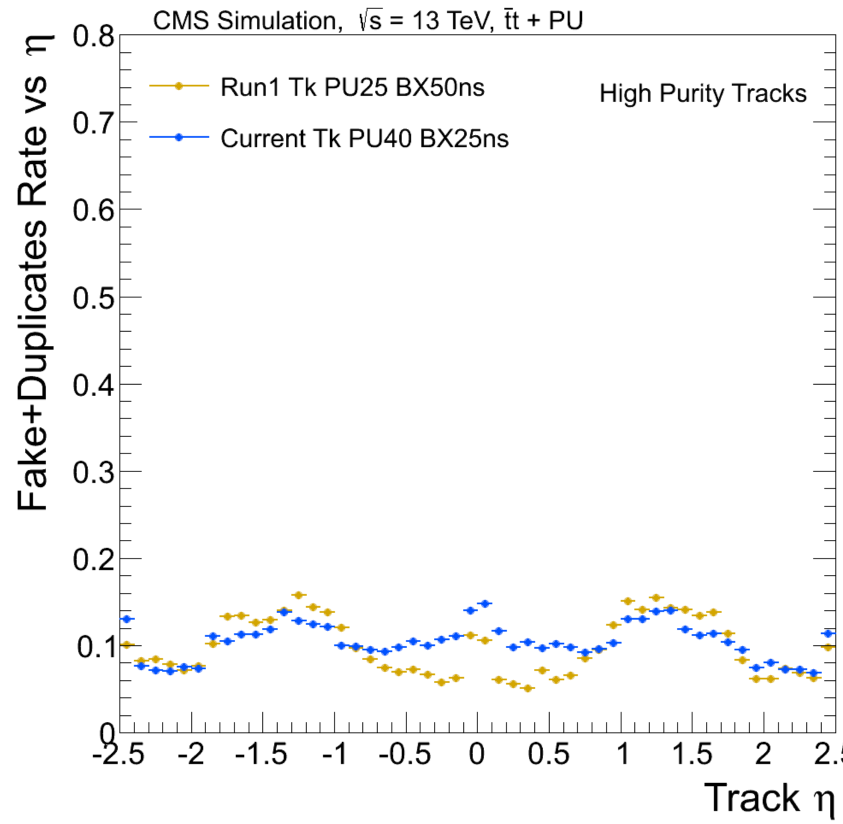
low p_T displaced tracks

- TTbar events with **<PU>=40, BX=25ns**
- Same or higher efficiency for prompt tracks
- Up to **6x** reduction in fake rate
- Slight efficiency loss for displaced tracks

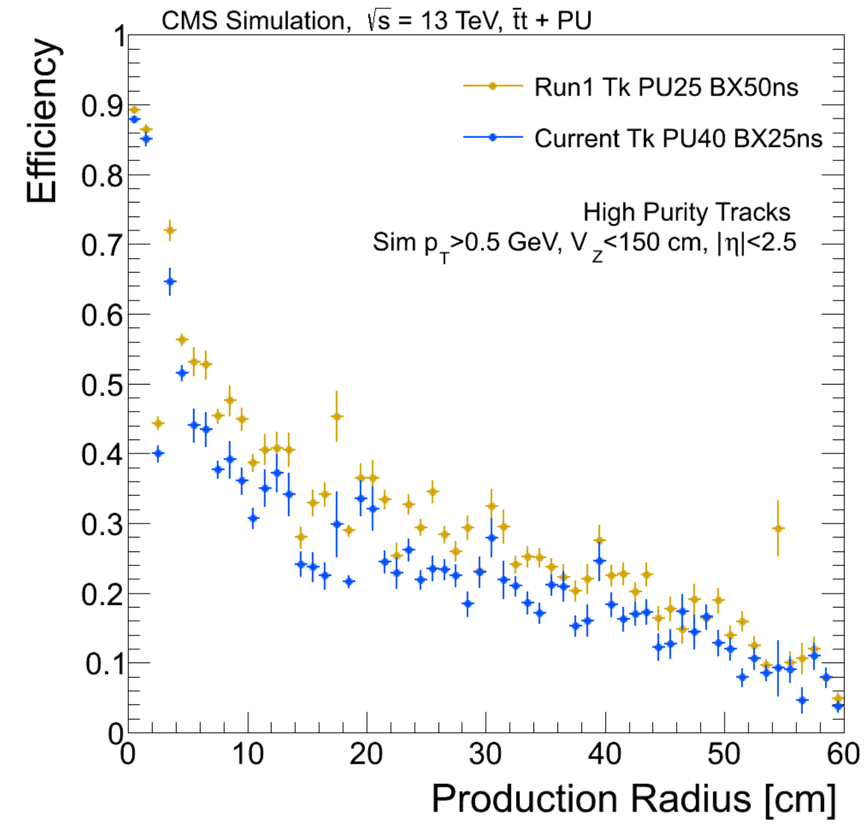
Run1 vs Run2 with Nominal PU Conditions



high p_T prompt tracks



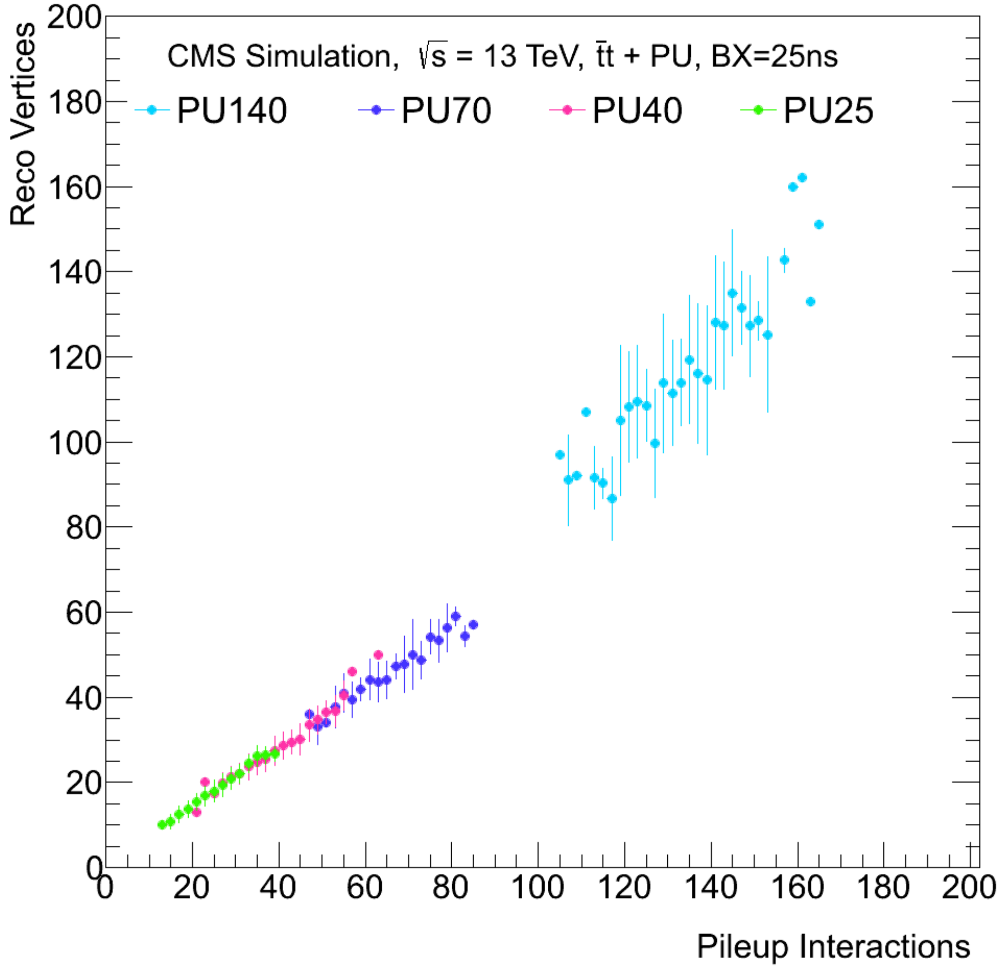
all tracks

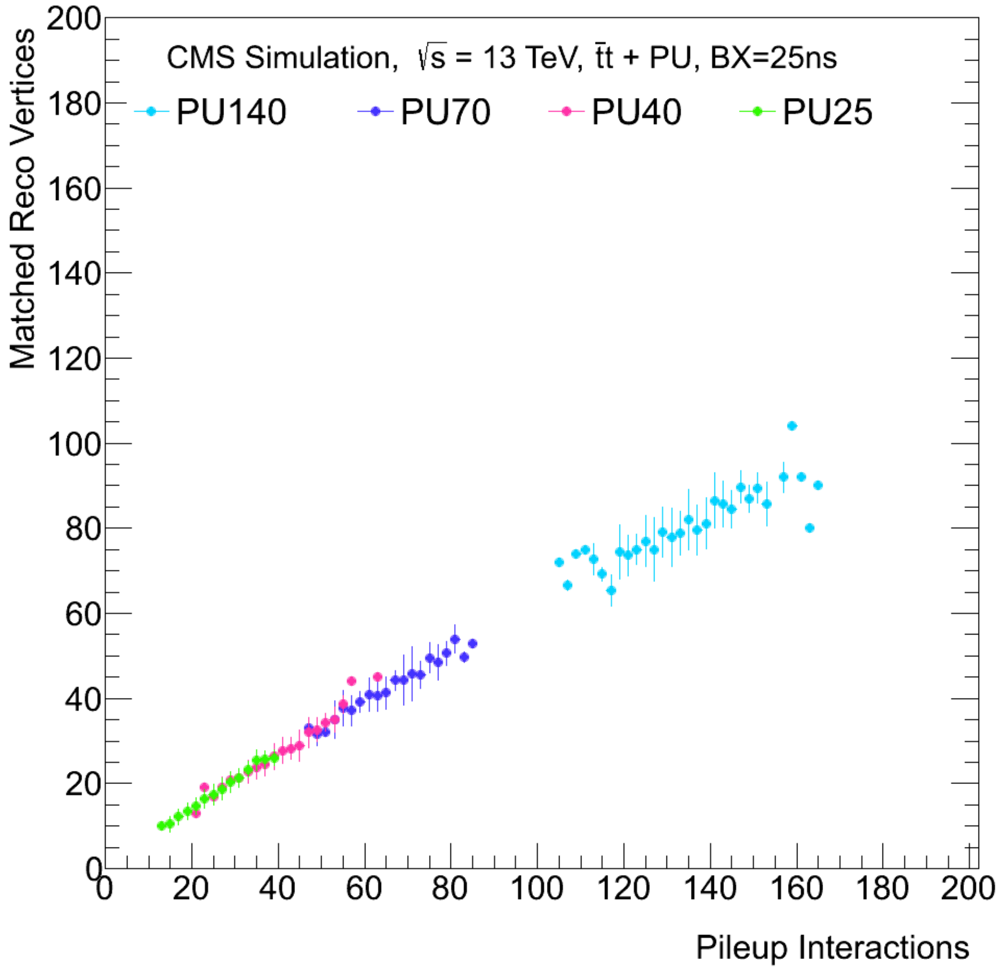


low p_T displaced tracks

- For physics analyses, the relevant comparison is with **nominal PU conditions**
  - Run1-like PU (<PU>=25, BX=50ns) for Run1 release
  - Run2-like PU (<PU>=40, BX=25ns) for Run2 release
- With much worse conditions, in Run2 we have same efficiency for prompt tracks, slightly higher fake rate, slightly lower efficiency for displaced tracks
- Run2 CMS **physics performance** to be the **same despite large PU** increase!
  - at least for reconstruction objects based on tracks

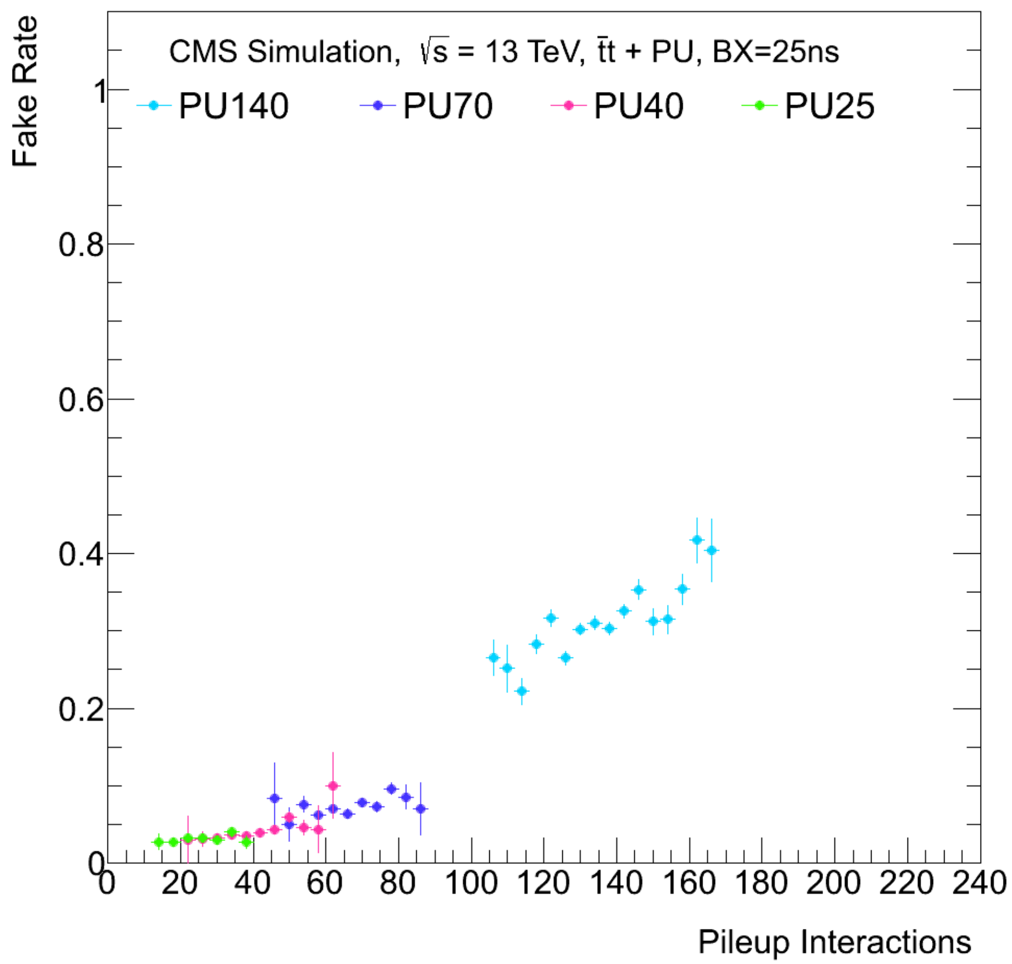
Primary Vertex Performance vs PU

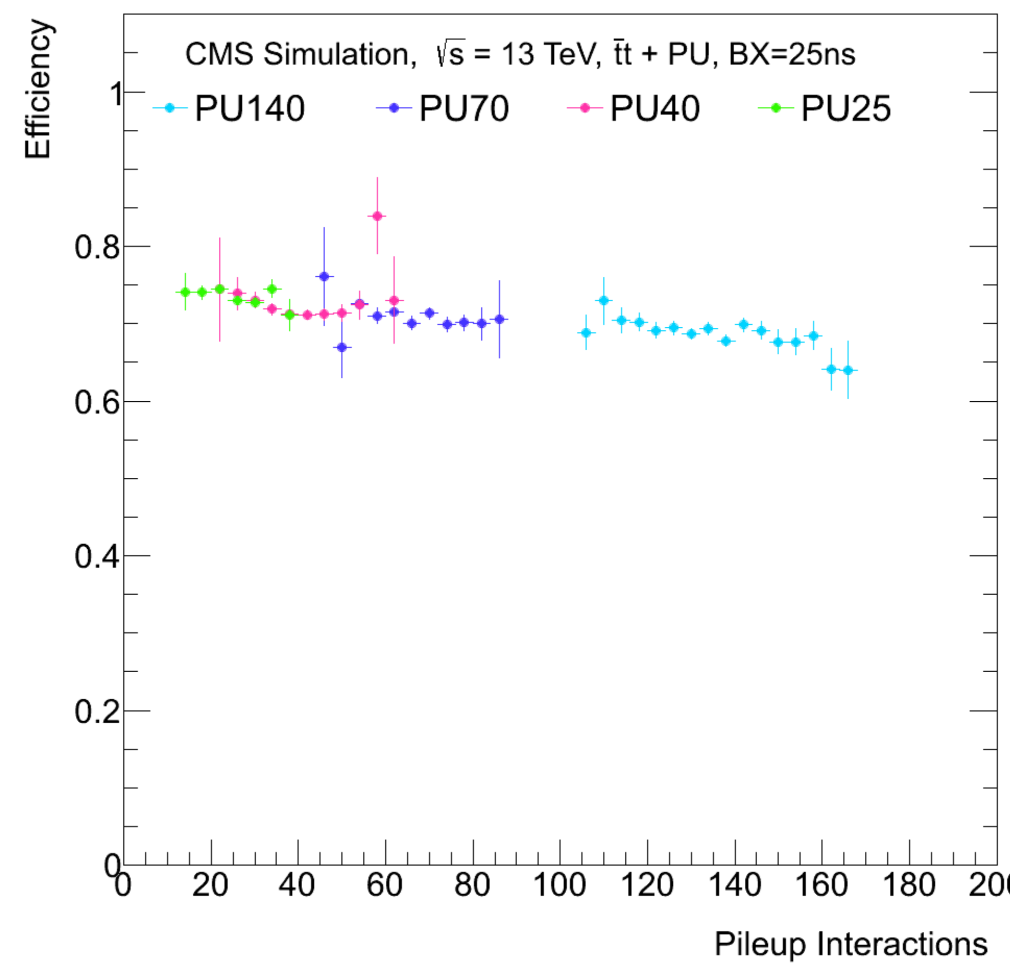




- **Number of reconstructed** vertices vs PU shows a **linear** trend with slope ~0.7 up to PU70. **Excess** of reconstructed vertices for **PU140**
- **Number of matched** vertices has **linear** trend **over all range**
  - A reconstructed vertex matches a simulated if $\Delta z<1$ mm and $\Delta z<3\sigma_Z$

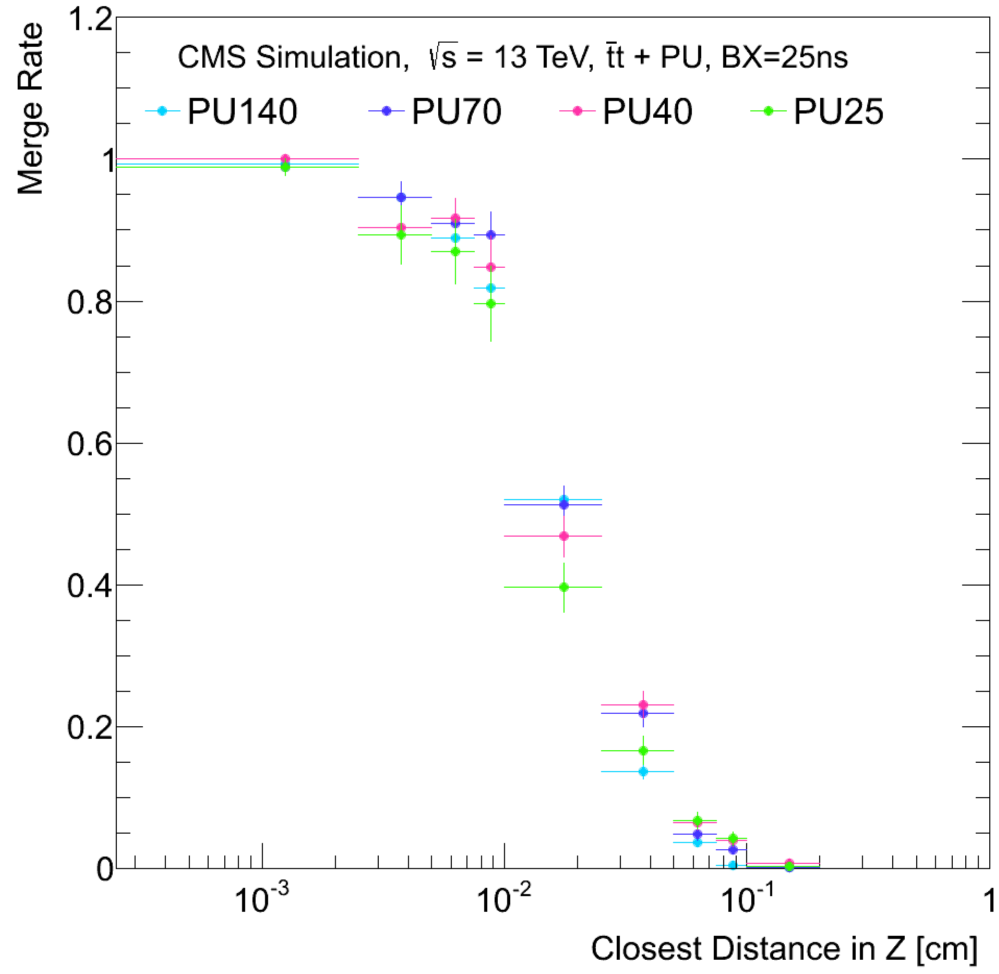
Primary Vertex Performance vs PU

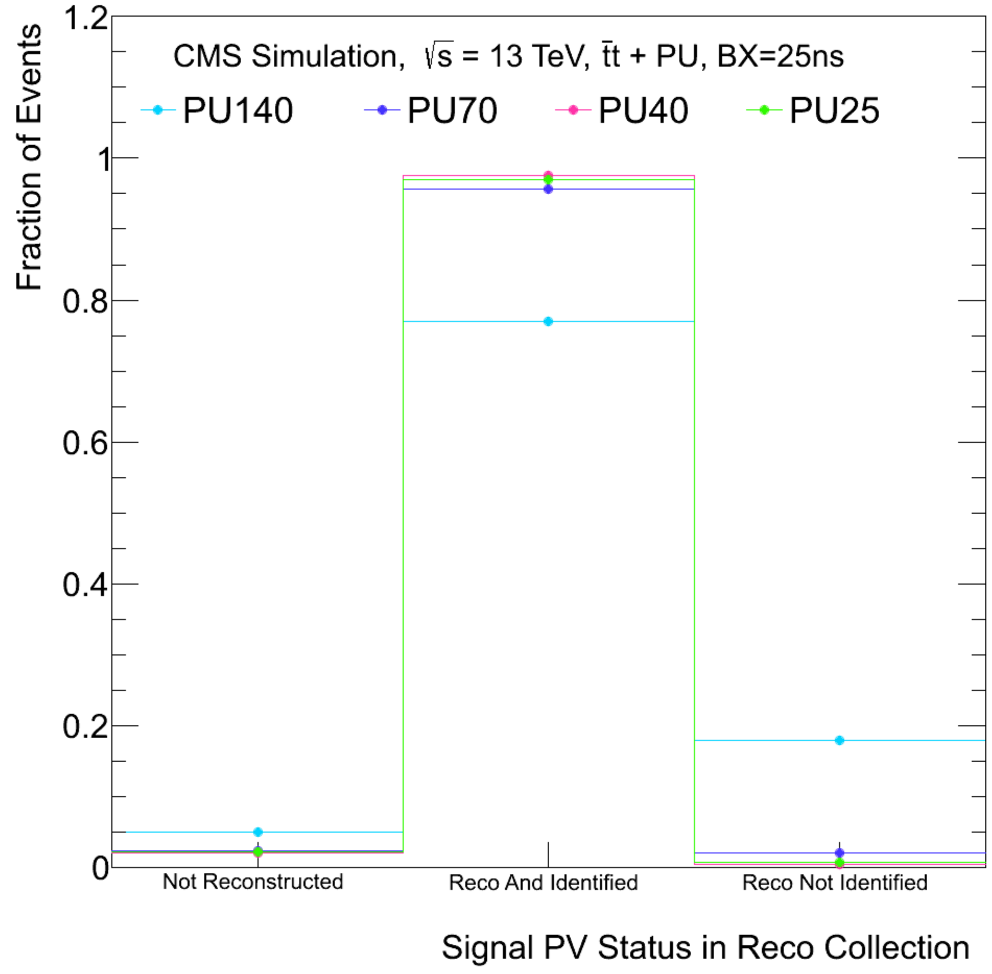




- **Number of reconstructed** vertices vs PU shows a **linear** trend with slope ~0.7 up to PU70. **Excess** of reconstructed vertices for **PU140**
- **Number of matched** vertices has **linear** trend **over all range**
  - A reconstructed vertex matches a simulated if $\Delta z < 1$ mm and $\Delta z < 3\sigma_z$
- These results are the effect of a **faster than linear** increase in **fake rate** and a **linear** decrease in **efficiency**

Merge Rate and PV Identification

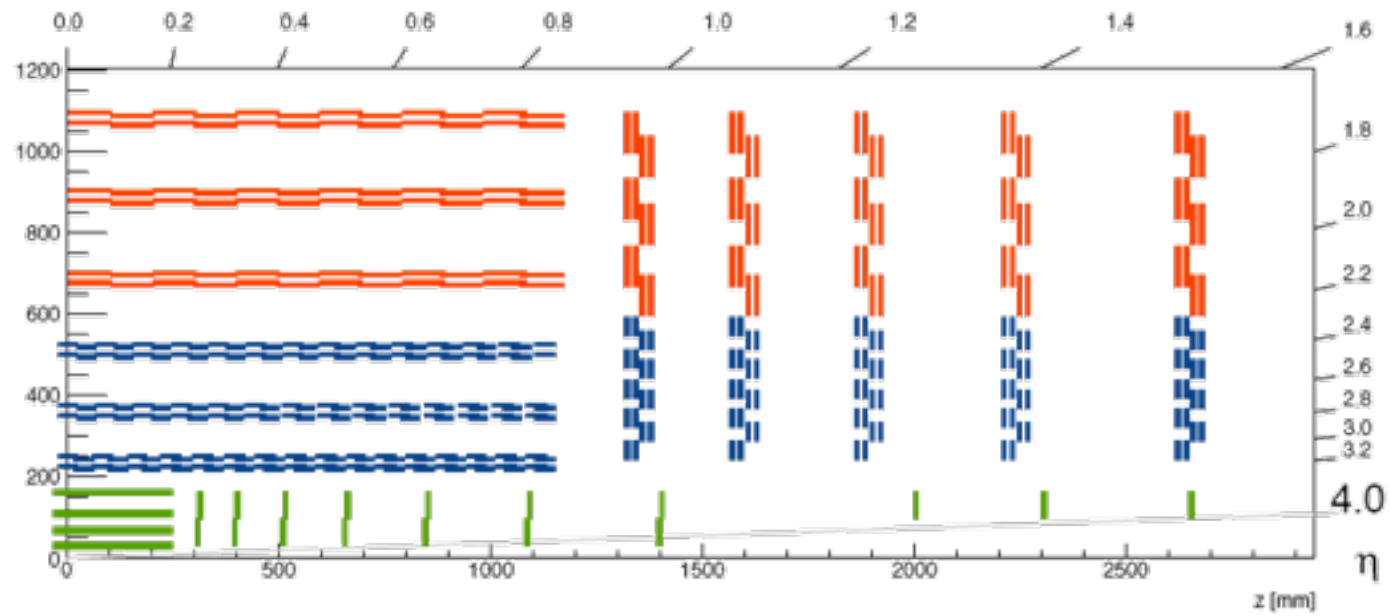




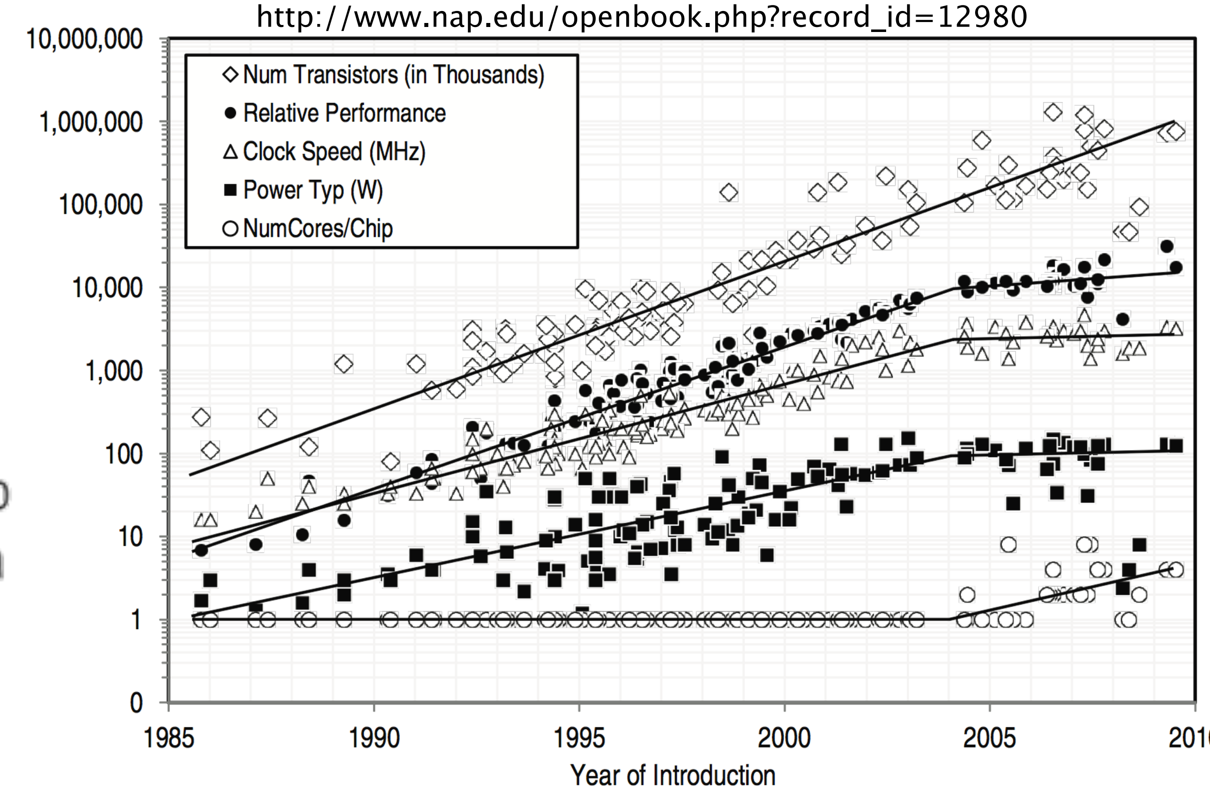
- **Merged** rate curve starting **below 1 mm separation** and ~PU independent
- On ttbar events, signal **vertex ID** based on highest Σp_T^2 has **stable** performance **up to PU70**
  - **reconstruction and identification OK for Run2 PU conditions!**

Final remarks

Tracking Challenges Beyond Run2



The Future of Computing Performance: Game Over or Next Level?
Samuel H. Fuller and Lynette I. Millett
http://www.nap.edu/openbook.php?record_id=12980



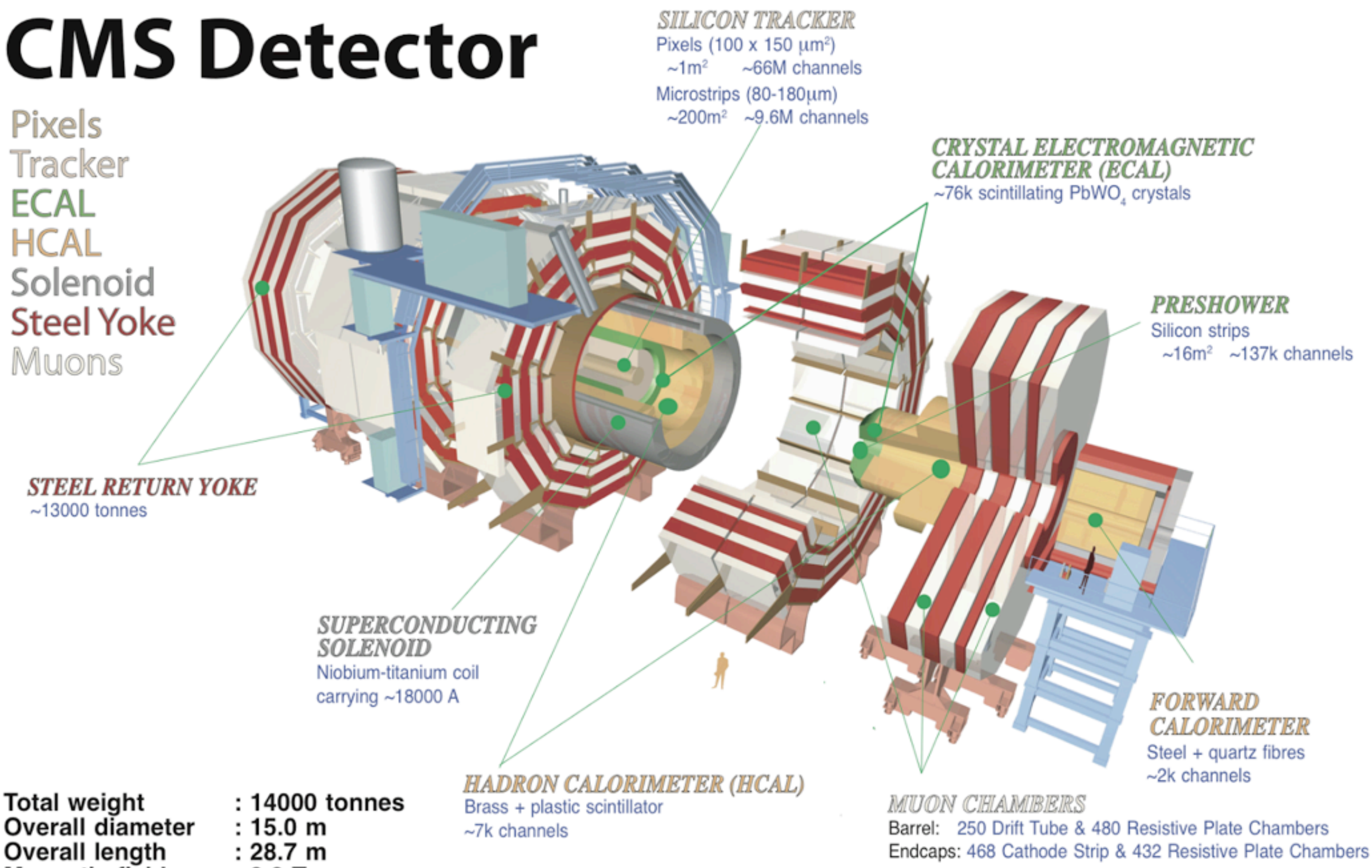
- New detectors:
  - Tracking with **upgrade geometries** is functional
    - **Phase1@50PU** and **Phase2@140PU** give comparable result
    - Phase2 geometry includes extension **up to |η|~4**
  - Fast and reduced readout of Phase2 outer tracker: **L1 track trigger** under development
- New computing technologies:
  - **Many cores**: parallelization, large **vector units**: same-instruction-multiple-data
    - Large effort made CMS tracking thread safe in production release
  - **Algorithms need to be developed/adapted** to work efficiently on new hardware
    - Hough Transform, Cellular Automata, Kalman Filter
- Time for R&D is now!

Conclusions

- High PU is a challenge for tracking
- Timing is under control
- Run2 performance comparable or better than Run1
- Stable primary vertex reconstruction performance with Run2 conditions
- Work for Run2 is not over, further improvements on the way (offline and HLT)
- SLHC is coming, tracking is getting ready for it

Backup

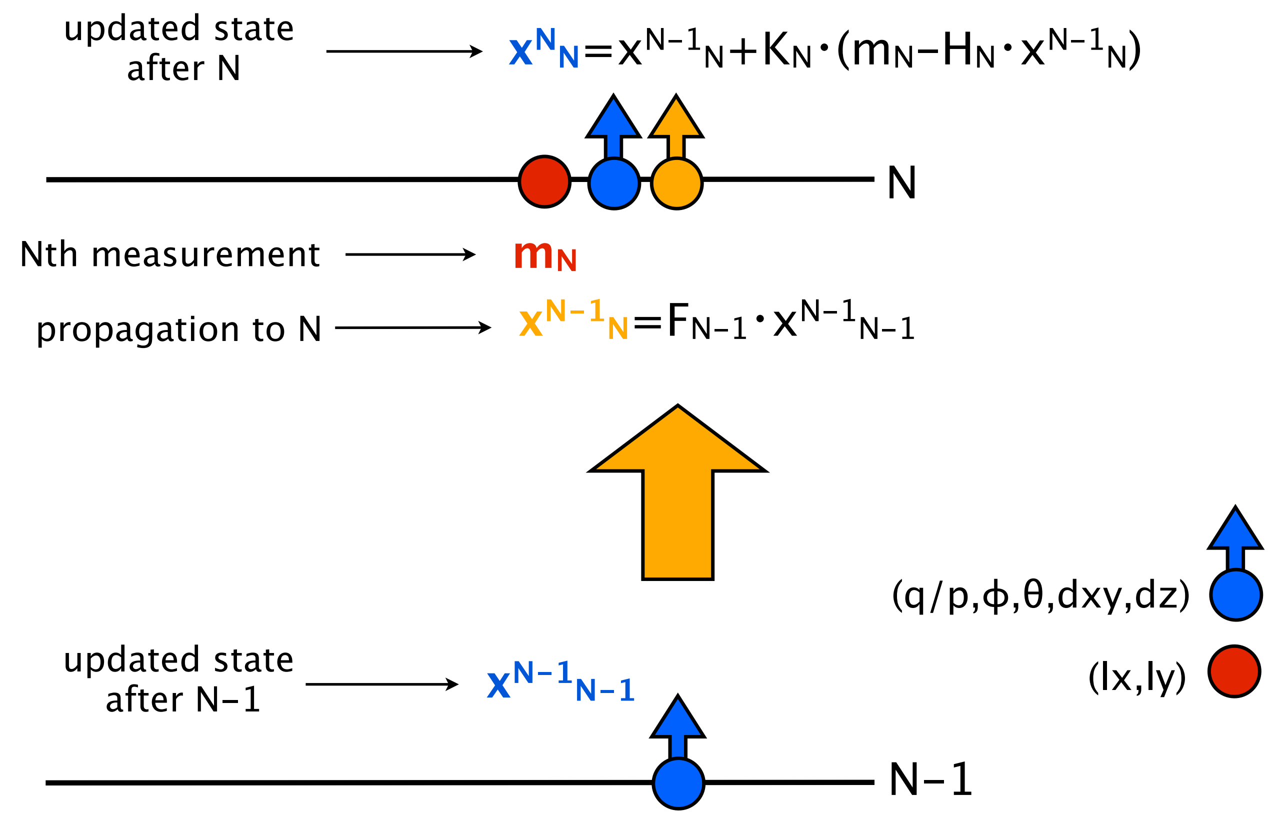
CMS Detector

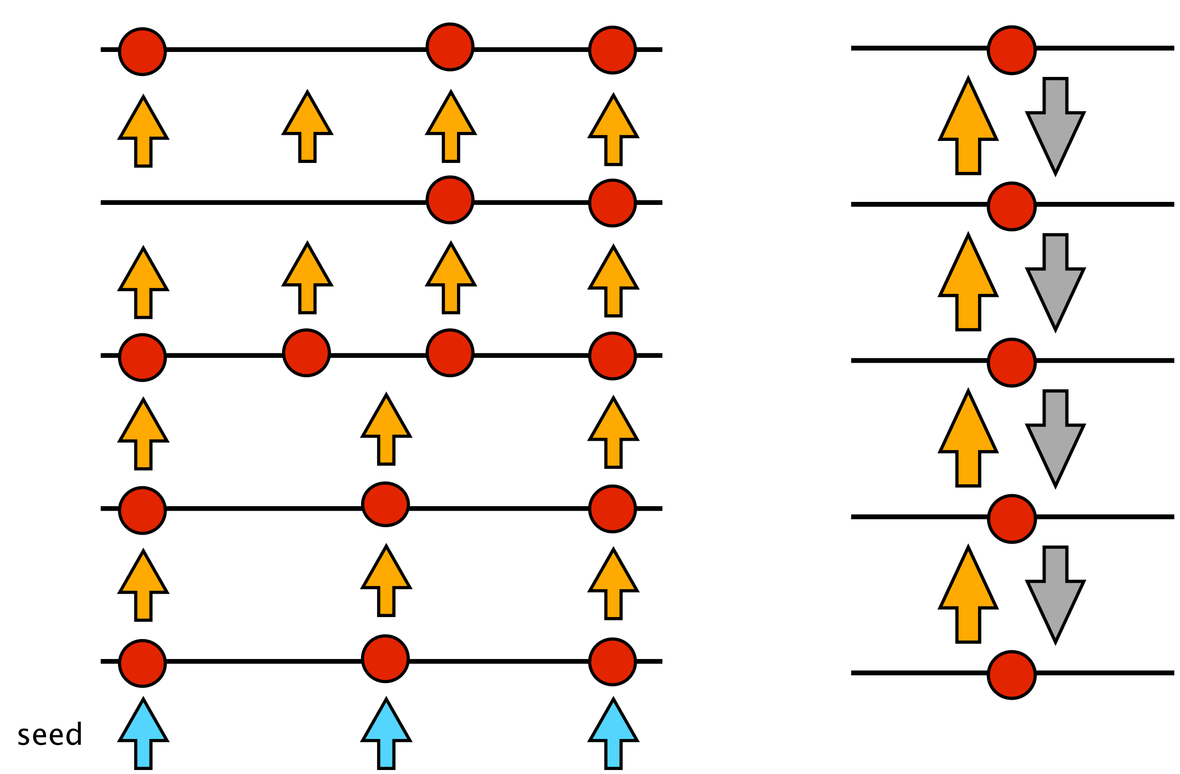


Total weight	**: 14000 tonnes**
Overall diameter	**: 15.0 m**
Overall length	**: 28.7 m**
Magnetic field	**: 3.8 T**

Track Reconstruction Steps

- Tracking based on Kalman Filter
  - add hits following the track direction, measurement (and thus extrapolation) precision improved at each layer
- Seeding
  - proto-tracks made of hit pairs or triplets with direction compatible with beam spot or vertices
- Pattern recognition
  - starting from the seed, search for compatible hits along the track; N+1 combinations considered:
    - best N based on χ2, +1 accounts for missing hit
- Fitting
  - estimation of track parameters; combining forward and backward fit yields best measurement at each point along the track
- Selection
  - assign quality flags based on N_{hits}, χ^2 and beam spot compatibility
    - tighter (looser) cuts for tracks with small (large) number of layers with hits
    - poor tracks discarded, best quality: “High Purity”





Primary Vertex Reconstruction

- Track Selection
  - Tight track quality selection: χ^2/ndof<20, nLayers≥5, nPixelLayers≥2, $d0/\sigma_{d0}<5$
  - No p_T selection to reconstruct also soft pile-up vertices
- Vertex reconstruction based on Deterministic Annealing
  - T→∞: all tracks associated to a unique vertex (beam spot)
  - T>Tmin: vertex identification by dynamic splitting (soft track assignment)
  - T<Tmin: unique assignment of tracks to a vertex
  - Resulting resolving power ~1 mm
- Adaptive Vertex fit
  - Iterative re-weighted Kalman Filter
  - With and without beam-spot constraint
  - Asymptotic resolution: 10 μm in x (y) and 12 μm in z

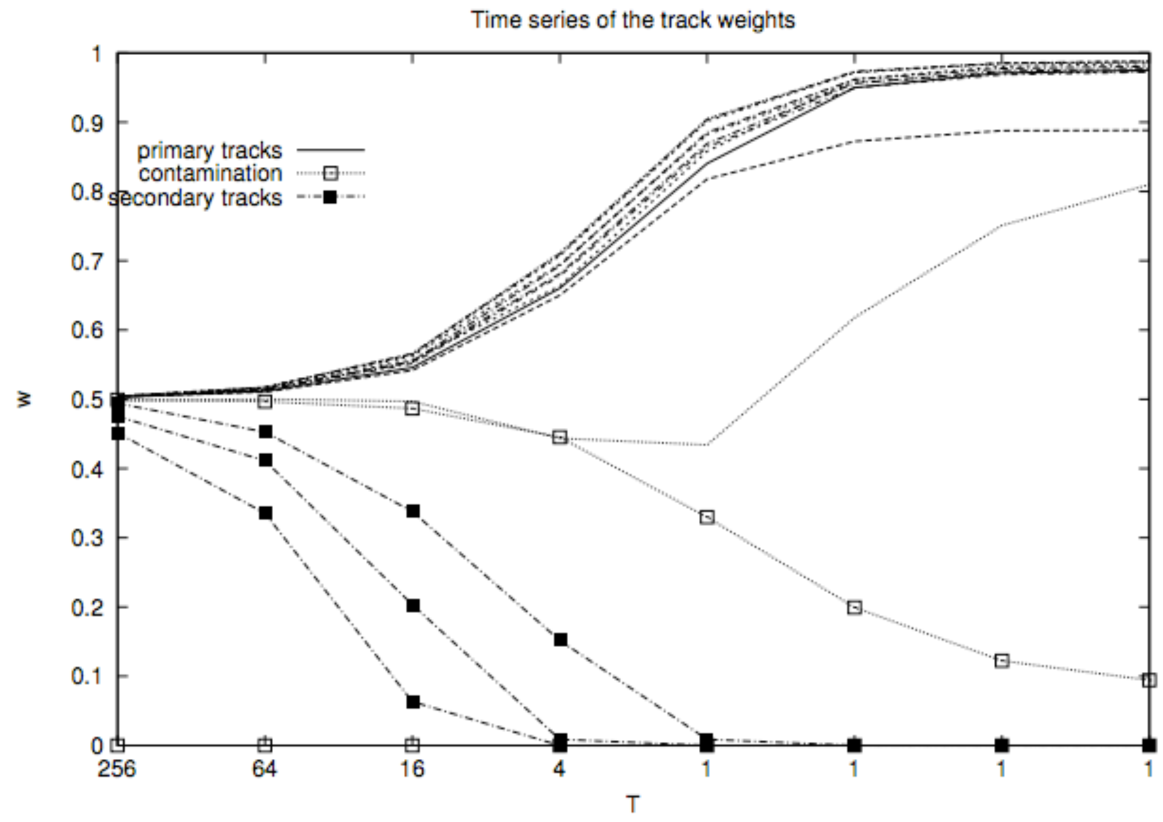
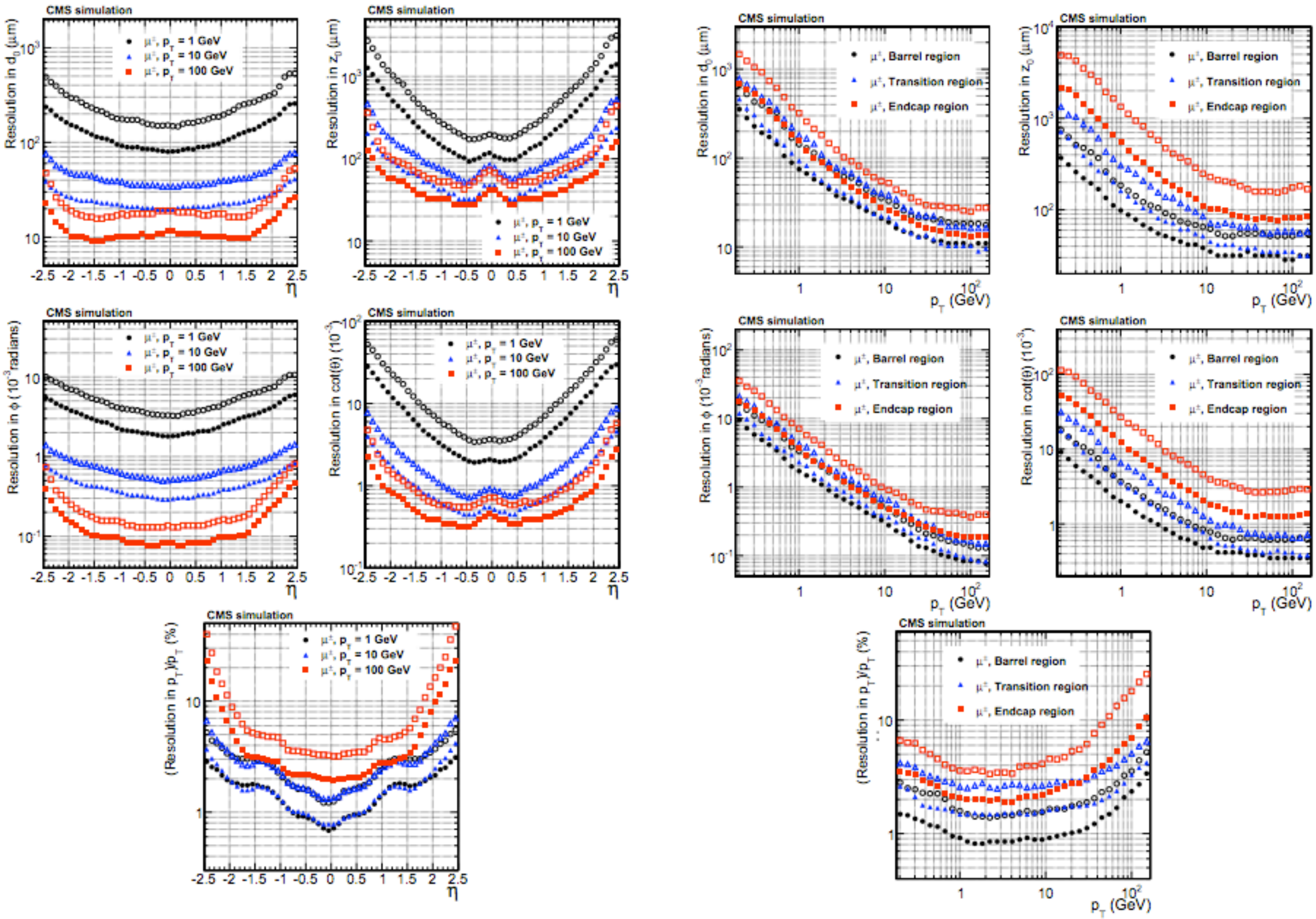
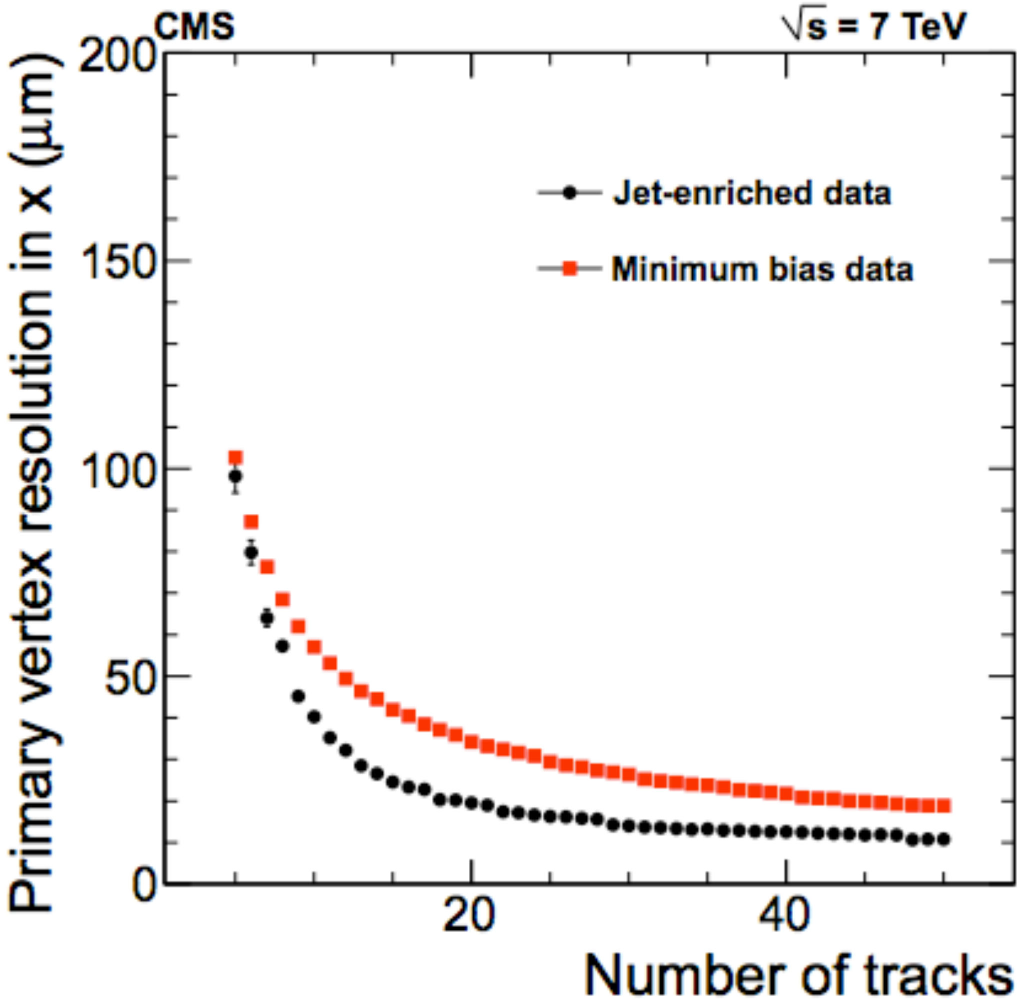


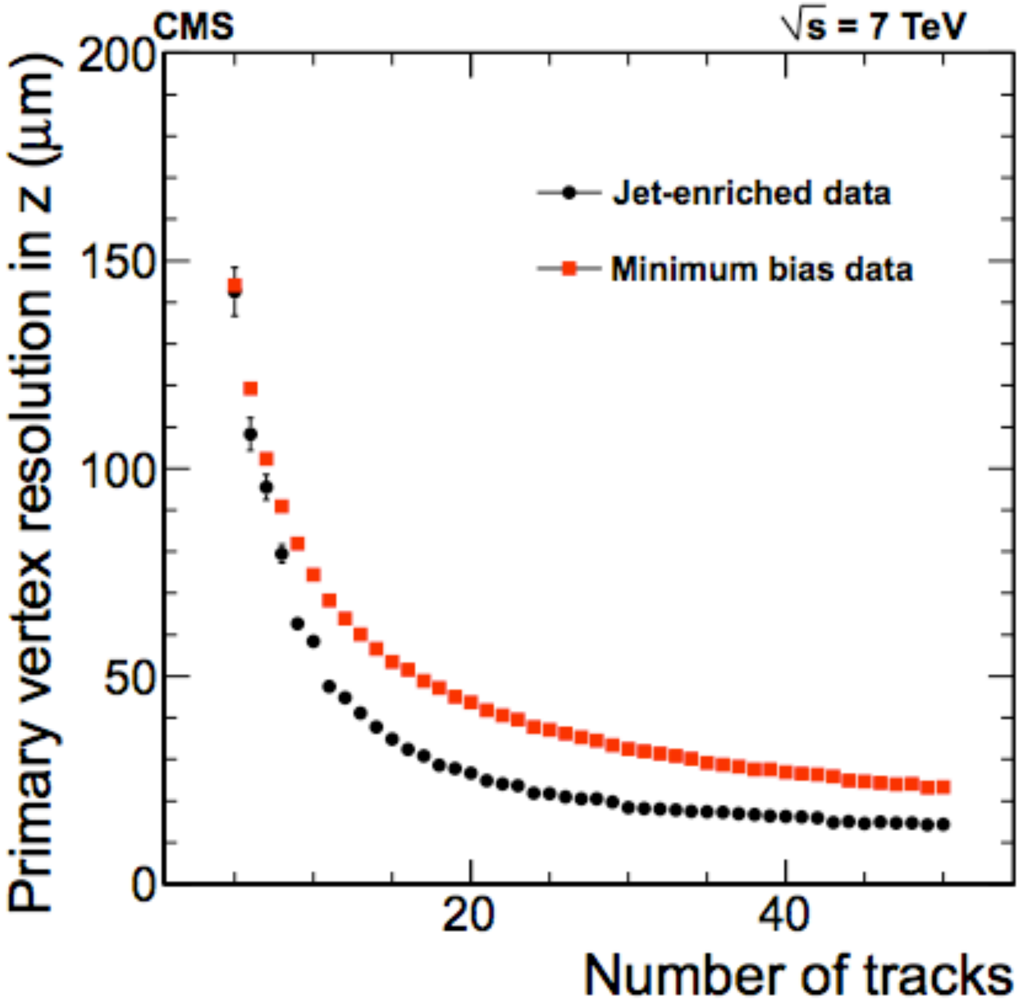
Figure 15: Evolution of the track weights of a $c\bar{c}$ primary vertex fit.

Track Parameter Resolutions



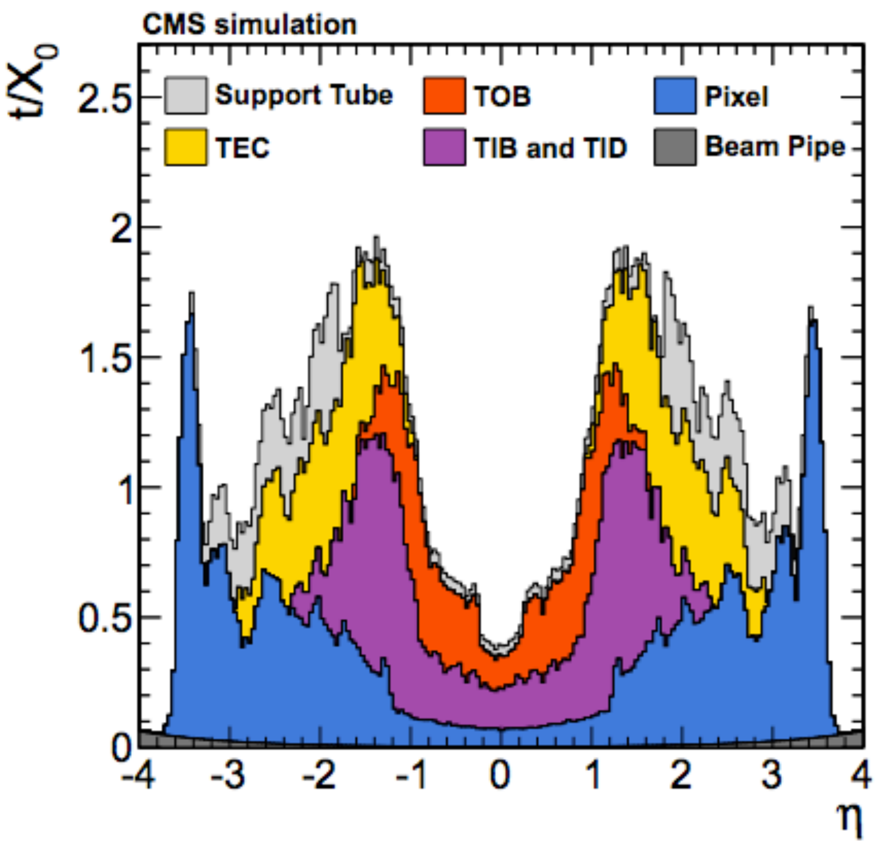
Vertex Resolution

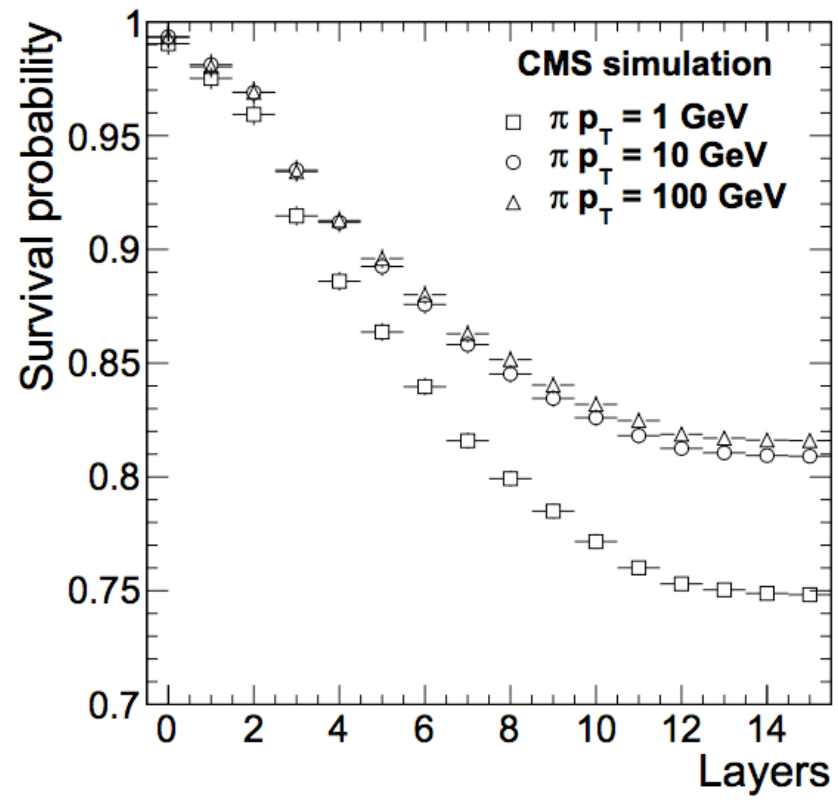


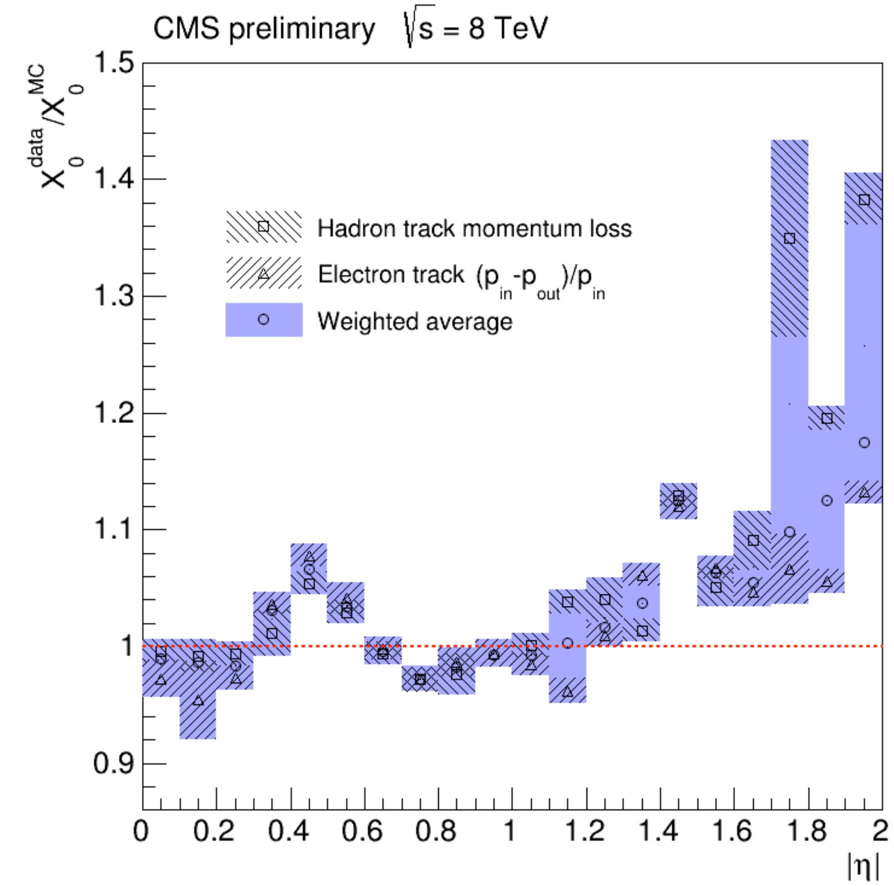


- For minimum-bias events, the resolutions in x and z are, respectively, less than 20 μm and 25 μm, for primary vertices with at least 50 tracks.
- Due to harder track momentum spectrum, the resolution is better for the jet-enriched sample across the full range of the number of tracks used to fit the vertex, approaching 10 μm in x and 12 μm in z for primary vertices using at least 50 tracks

Measurement of Tracker Material







- Different methods are used to estimate the tracker material:
- Hadron track momentum loss:
  - the momentum loss by hadron tracks in the tracker volume is proportional to the amount of material traversed.
  - The track momentum loss is estimated (on low pT tracks, 0.9<pT<1GeV) integrating it along the track trajectory.
- Electron track (pin-pout)/pin:
  - the electron momentum pin(pout) is obtained from the forward(backward) track fit.
  - The fractional difference between the two depends on the amount of energy radiated by bremsstrahlung and it is used as an estimator of the material traversed by the track.
    - Electrons are selected from Z→e+e- decays.
- The deviation from unity of the ratio X0data/X0MC is interpreted in terms of material mis-modeling and used as input to improve the tracker description in the simulation.