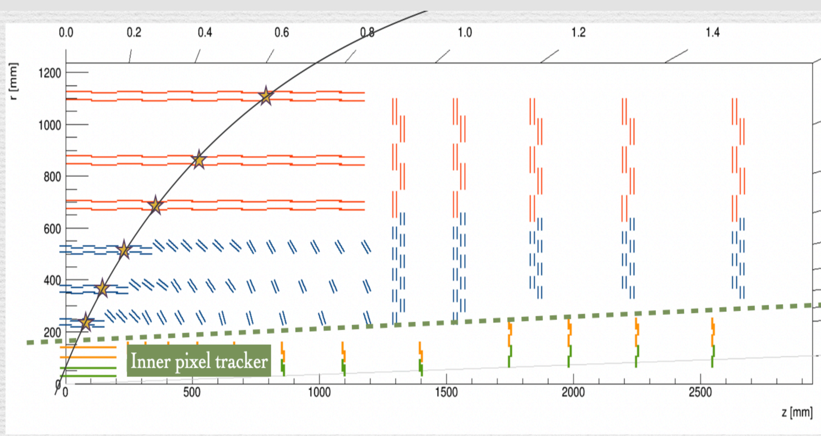
Line Segment Tracking in Phase 2 software upgrade in CMS

Yanxi Gu et al.* on behalf of the CMS collaboration

Tracking challenges and GPU solutions

- Tracking is a very time-consuming component of the CMS event reconstruction today (~40%)
- Due to the massive number of collisions and tracks in the upcoming HL-LHC, traditional Kalman Filter based method will take a very large amount of time. Novel GPU solutions are essential for outer track finding with competitive timing
- Line Segment Tracking(LST) is a highly parallelizable track building algorithm designed to exploit GPU capabilities
- The algorithm starts from tracker hits and connects neighboring objects to build short tracks. Then it links short tracks to get longer tracks. Each linking step is parallelizable on GPU

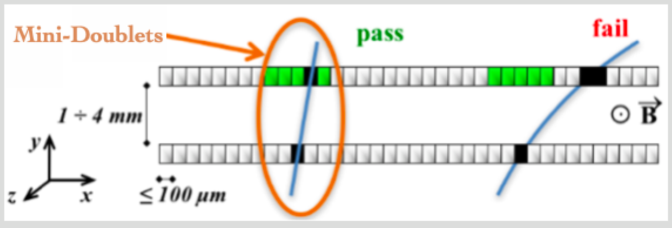


Algorithm logic

From hits to Mini-Doublets

Each layer has modules built out of a stack of 2 sensors. so this step is combining the 2 sensor hits in 1 layer

- p_T threshold set to 0.8 GeV
- Calculate a particle trajectory with the minimum p_T to define a window (green)
- Combine hits to build a Mini-Doublet(MD)

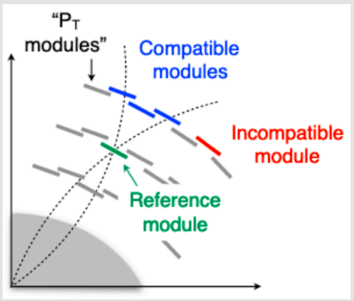


From hits to Mini-Doublets

From Mini-Doublets to Segments

This step links between 2 layers

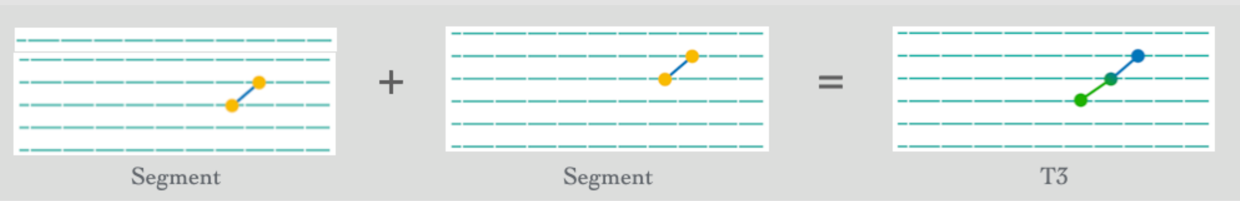
- Construct compatible module maps between layers, with positive/negative charge, p_T threshold
- Link MDs to Segments
- Each Segment has 4 hits (2 MDs)



From MDs to Segments

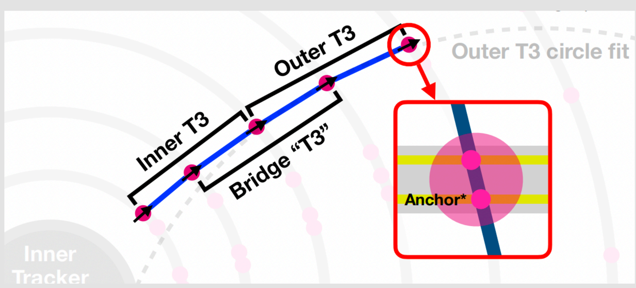
From Segments to T3s

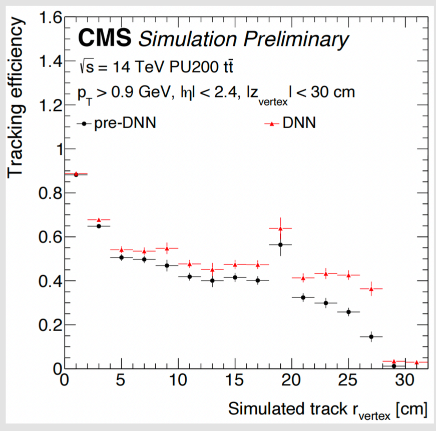
- Require Segments to be in adjacent layers and share a common MD
- Geometric selection: alignment check, circle fit criteria in r-phi plane, etc
- Link compatible segments to build T3s. Each T3 has 6 hits (in 3 layers)

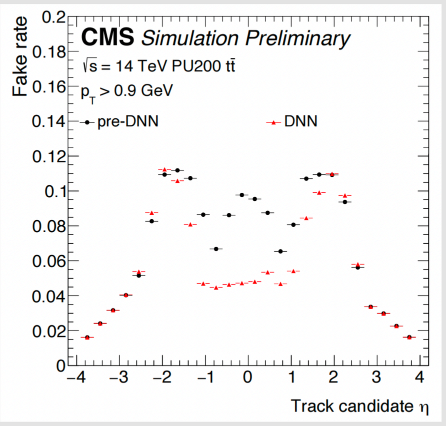


From T3s to T5s

- Two T3s in adjacent layers sharing a common MD can be linked as one T5.
- Apply loose orthogonal quality cuts based on geometrics
- Train a 2-layer DNN, with each layer 32 nodes. Then apply cut on DNN scores
- From the comparison with the original cut based selections, displaced tracks efficiency increases while the fake rate drops.

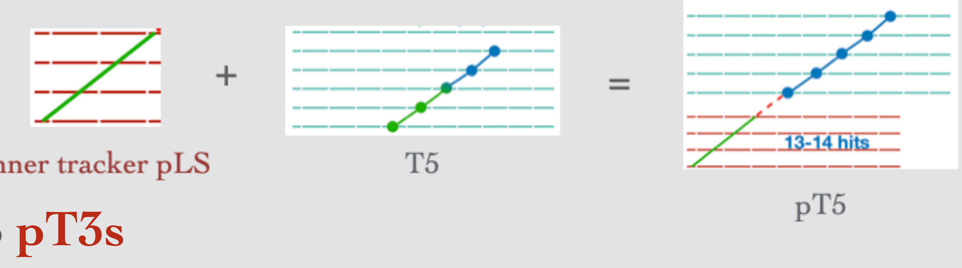






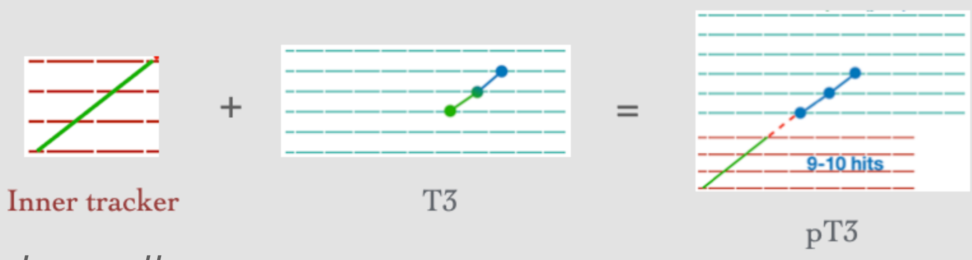
From T5s to pT5s

- Accept 3 hits or 4 hits pixel seeds as input pixel Line Segments(pLS)
- Link the pLSs with T5s. Require them to pass geometric constraints.



From T3s to pT3s

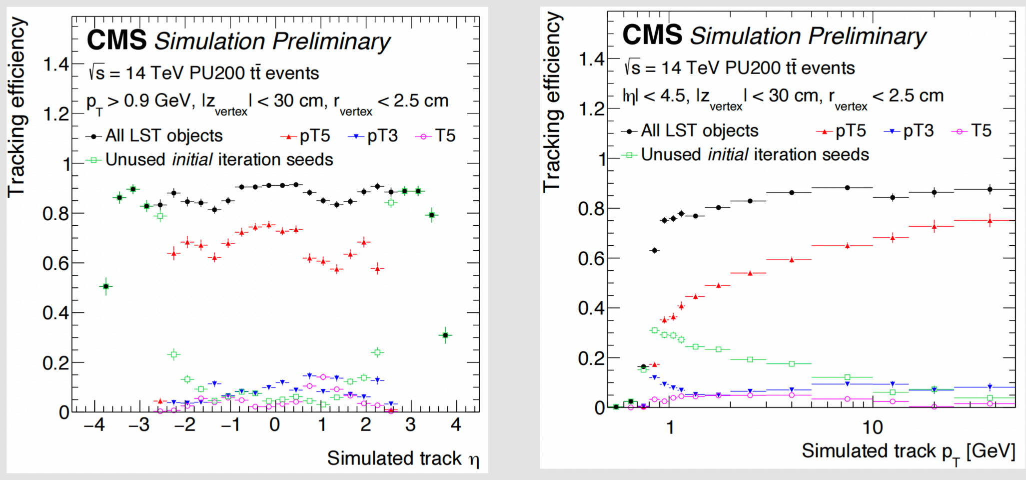
- Use the remaining T3s and pLSs to build pT3s. Each pT3 has 9-10 hits

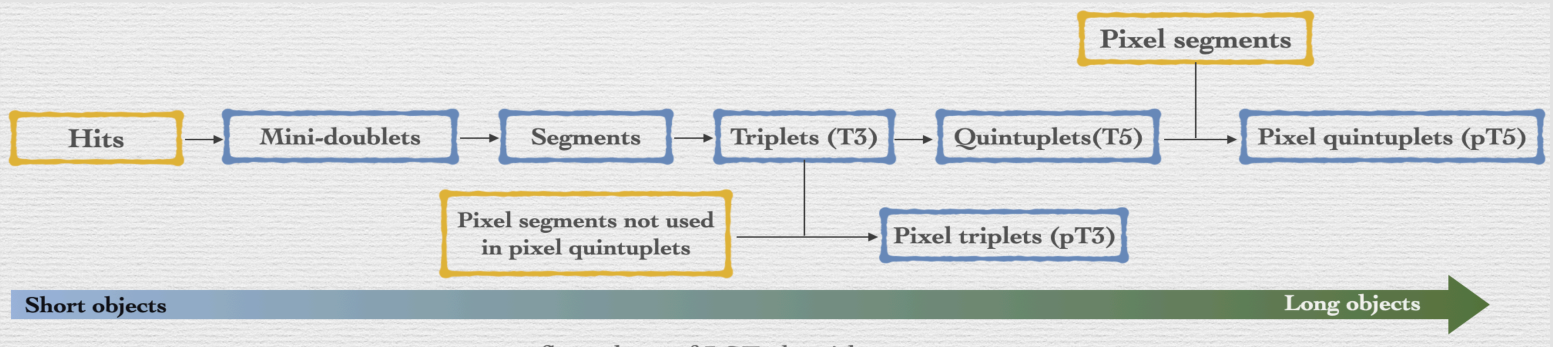


Track candidate collection

Duplicate removal (within each collection) and cross cleaning (across different collections) are implemented

- pT5 : purest object with most hits
- pT3 : low p_T tracks missed by pT5
- T5 : displaced tracks which escape the detection in the pixel detector
- 4-hits pLS: low p_T and high $|\eta|$ tracks

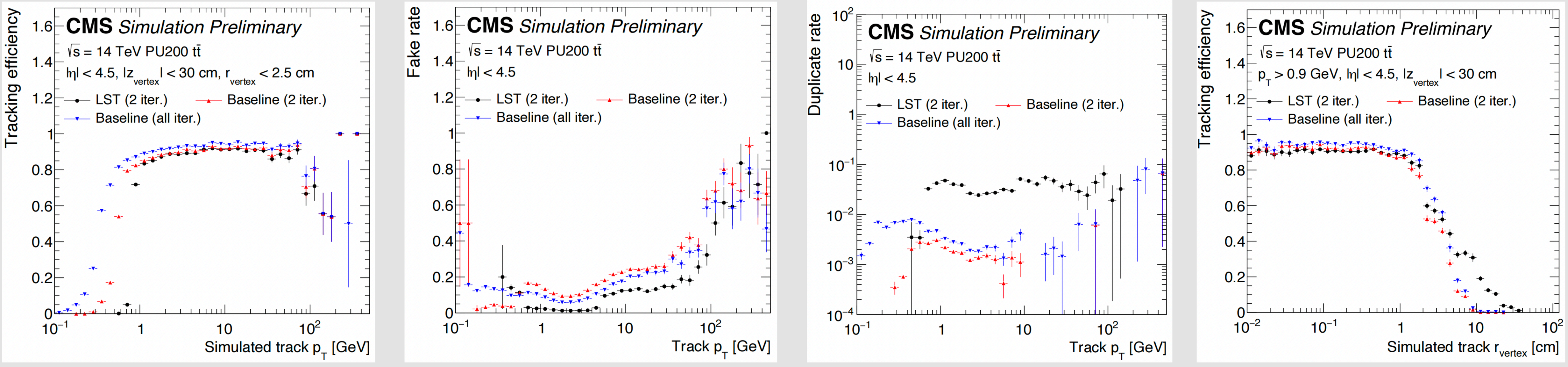




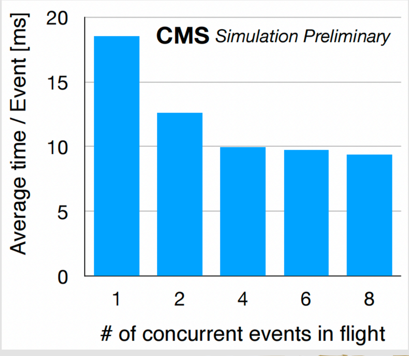
flow chart of LST algorithm

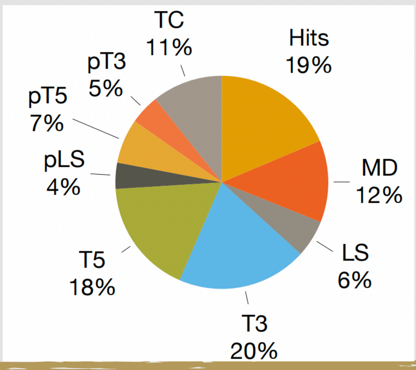
Physics and timing performance

- The black and red curves show the comparison between LST and CMSSW benchmark performance (CKF algorithm, ref. 4)
- LST has achieved comparable efficiency with CMSSW tracking benchmark performance above targeted $p_T > 0.8$ GeV
- LST reduces the track fake rate greatly, but has higher duplicate rate than the CMSSW baseline
- LST at high r_{vertex} shows significantly better tracking efficiency even compared to baseline tracking with all iterations



- Timing performance is measured on GPU (NVIDIA A100). Here is the average time to process one event
- Timing performance improves when going from single stream to multi-stream
- Does not include data transfer time from host to device and vice versa





Status and plans

- Good performance LST algorithm with high efficiency and low fake rate
- Test setup with CMSSW(offline and HLT), got comparable physics performance and potential to greatly reduced the timing
- Standalone code fully migrated to alpaka(ref. 6) to support both CPU and GPU backend
- Integrate LST as an official CMSSW external package to allow further testing centrally. Full integration into CMSSW package can follow
- Extension to broader phase space: T4(displaced tracks), pT2(low pT tracks)
- ML explorations on different LST objects, GNN possibility
- Synergies with other phase 2 algorithms, e.g. mkFit(ref. 5), etc

Authors and references

Balaji Venkat Sathia Narayanan, Jonathan Guiang, Manos Vourliotis, **Yanxi Gu**, Slava Krutelyov, Matevž Tadel, Avi Yagil[UCSD]
Tres Reid, Gavin Niendorf, Peter Wittich[Cornell]
Mathew Dittrich, Mayra Silva, Philip Chang[U of Florida]
Andres Rios Tascon, Peter Elmer [Princeton]

- Performance of Line Segment Tracking algorithm at HL-LHC, CMS Public Note, CMS DP-2023/019
- Improved Performance of Line Segment Tracking Using Machine Learning, CMS Public Note, CMS DP-2023/075
- Improving tracking algorithms with machine learning a case for line-segment tracking at the High Luminosity LHC, CMS CR-2023/075
- Description and performance of track and primary-vertex reconstruction with the CMS tracker, CMS-TRK-11-011
- Speeding up Particle Track Reconstruction using a Parallel Kalman Filter Algorithm, Journal of Instrumentation, arXiv:2006.00071
- Tuning and optimization for a variety of many-core architectures without changing a single line of implementation code using the Alpaka library, arXiv 1706.10086