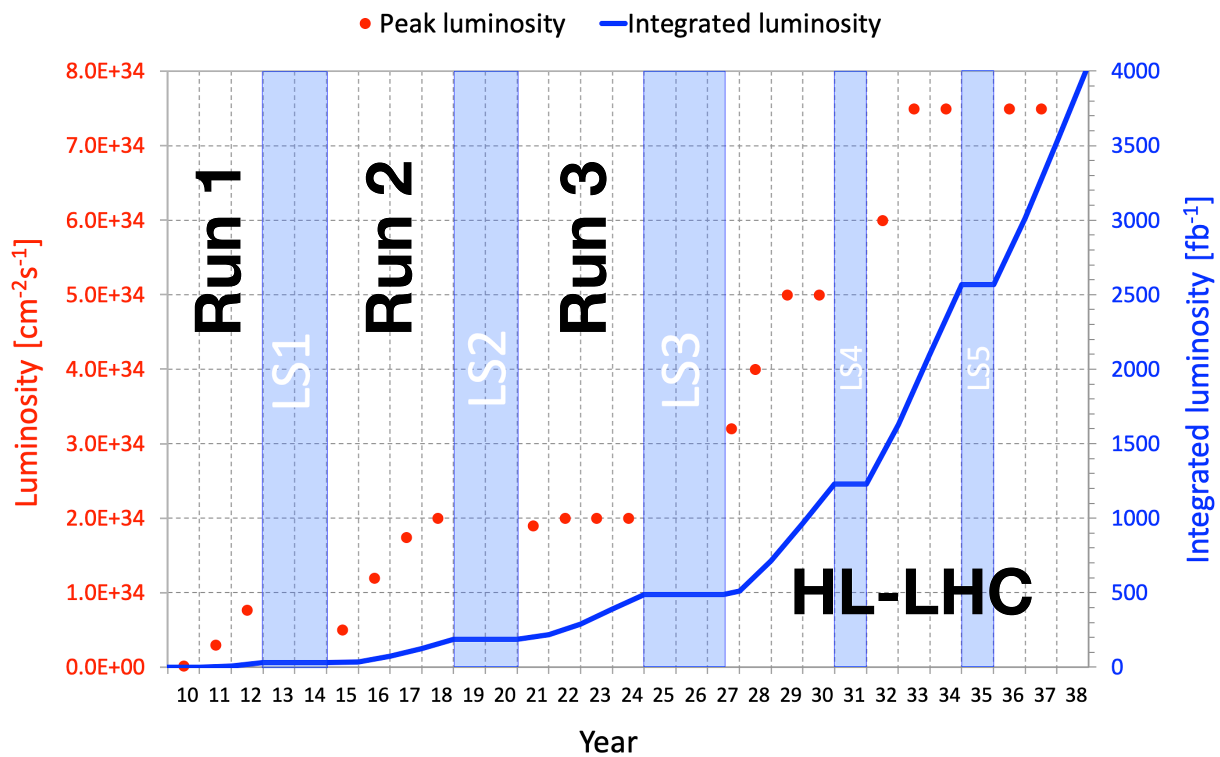
Segment Linking: A Highly Parallelizable Track Reconstruction Algorithm for HL-LHC

HEP 2022

Philip Chang, Peter Elmer, Yanxi Gu,
Slava Krutelyov, Balaji Venkat Sathia Narayanan,
Gavin Niendorf, Tres Reid, Matevž Tadel, Manos Vourliotis,
Bei Wang, Peter Wittich, Avi Yagil

Intro to HL-LHC Challenges

- High Luminosity LHC (**HL-LHC**): Planned LHC upgrade with up to ×**8 current luminosity**.



- Higher luminosity ⇒

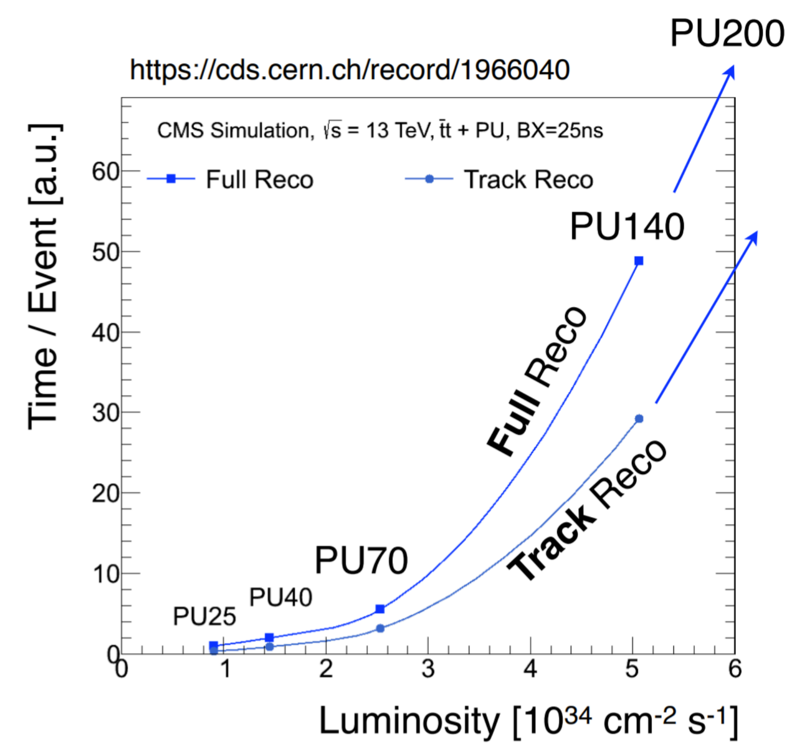
  More simultaneous interactions (PU): $\langle PU \rangle_{LHC} \lesssim 70 \rightarrow$ $\boldsymbol{\langle PU \rangle_{HL\text{-}LHC} \approx 200}$

- More PU ⇒

  Increased computational complexity ⇒

  - Increased timing.
  - Increased cost.

https://cds.cern.ch/record/1966040



Heterogeneous Computing

- Run 2 computational resources dominated by CPU ⇒

  Exploit alternative processing units: FPGA, **GPU**, etc.

- Application of GPU computing in CMS:
  - Pixel track & vertex reconstruction (patatrack).
  - Outer tracker strip local reconstruction.
  - Electromagnetic calorimeter (ECAL) reconstruction.
  - Hadronic calorimeter (HCAL) reconstruction.

Expected to be large fraction of High Performance Computing

- Significant **reductions in timing** (~25%) for Run 3 CMS High Level Trigger (HLT)[*].
- Alternative processing units: Cheaper per nominal compute operation ⇒

  Significant **cost reduction** (~35–75%) projected for HL-LHC computational resources[*].

[*]: CMS-TDR-022

Segment Linking Overview

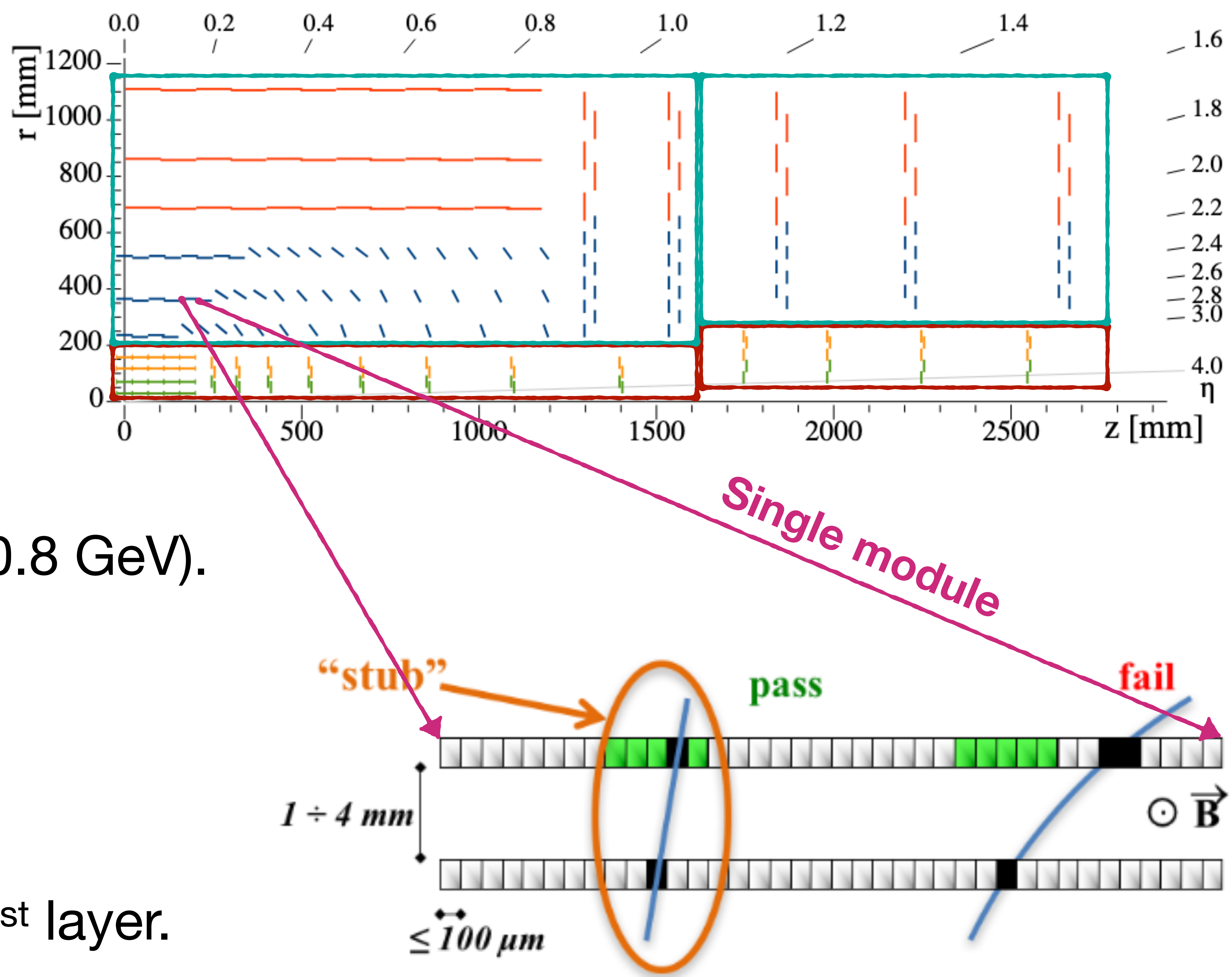
- Leverage GPU performance in track reconstruction → Algorithm architecture suitable for **parallelization**.
- **Segment Linking:**
  A highly parallelizable track reconstruction algorithm
  - Moves away from sequential Kalman filter based algorithms → mkFit project tries to make the most out of that.
  - Relies on **local hits** in the tracker to build short tracks.
  - Short tracks linked to form **progressively longer tracks**.
  - Selects objects from intermediate collections to create a **Track Candidate** collection with **high efficiency** and **low fake rate**.
- Inspired by XFT algorithm in the CDF at the Tevatron.
- Prototype presented in ICHEP 2016.

CMS Outer Tracker @ HL-LHC

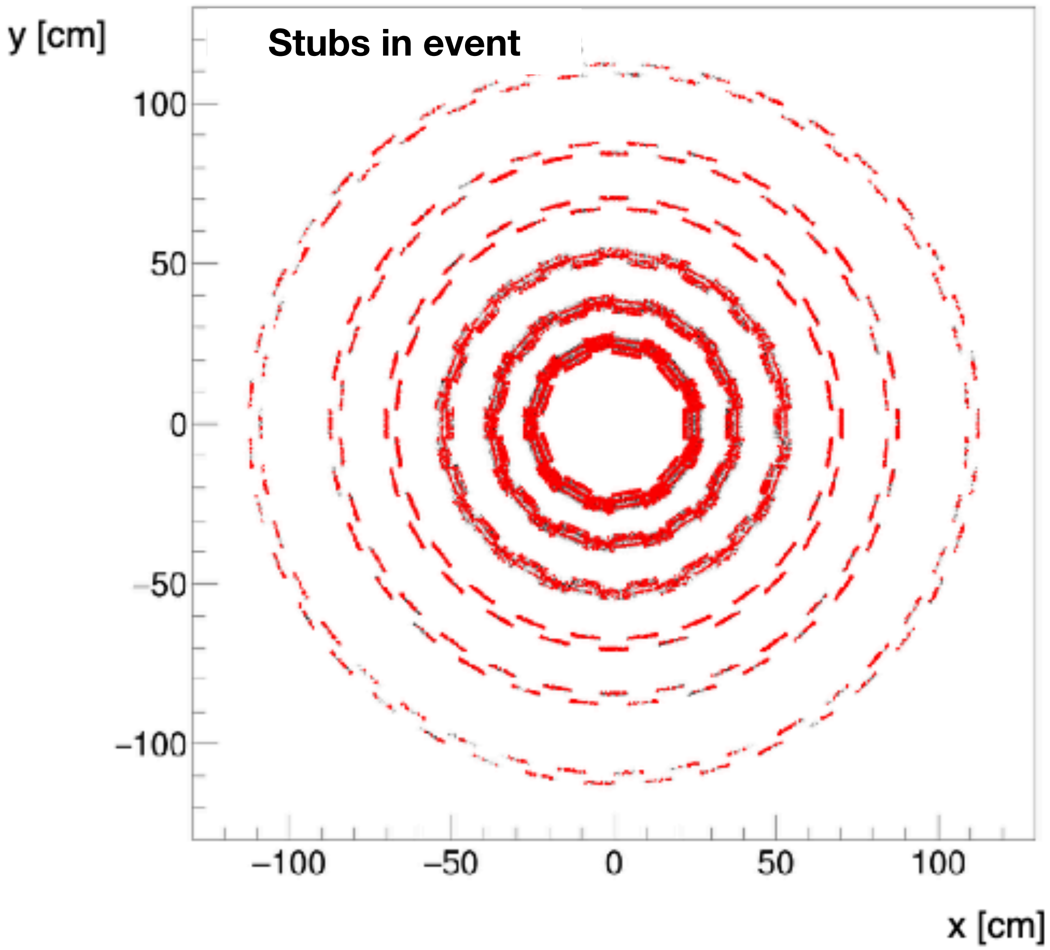
- CMS HL-LHC **outer tracker** ideal for algorithm application:
  - Closely-spaced sensors ⇒

    Local hits (**stubs**) above p_T threshold (0.8 GeV).
- Stubs instead of hits ⇒

  Up to ×7.5 reduction of combinatorics:
  - E.g. 5.9k vs. 36k in 1st layer.

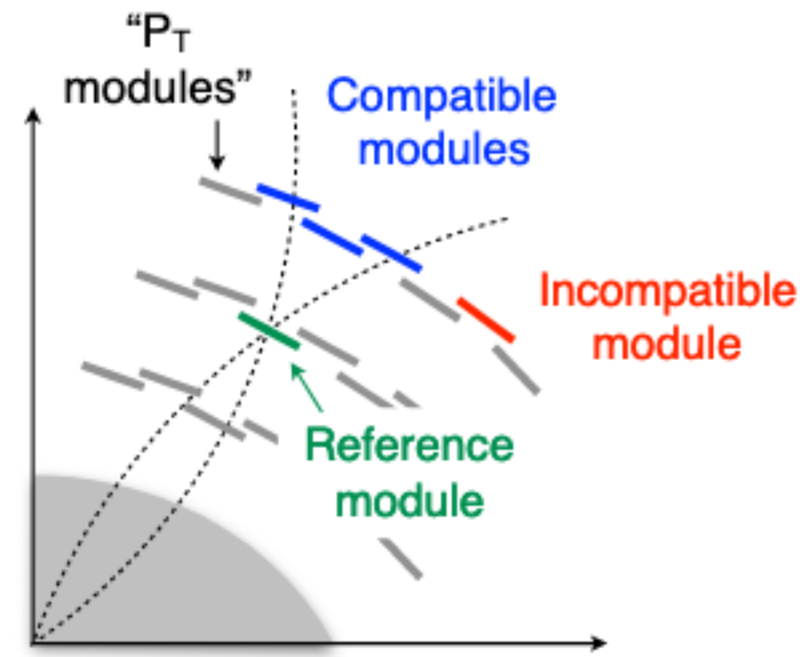


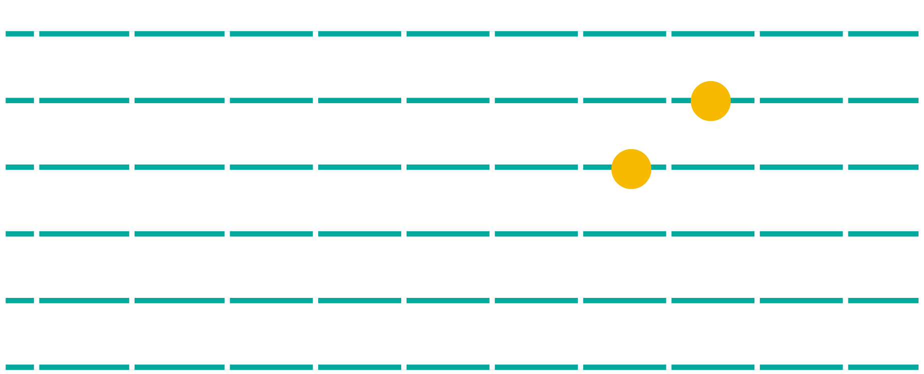
Linking Stubs



Consider **helices** with
- ± **charge**
- $\mathbf{p_T}$ = 0.8 GeV ⇒

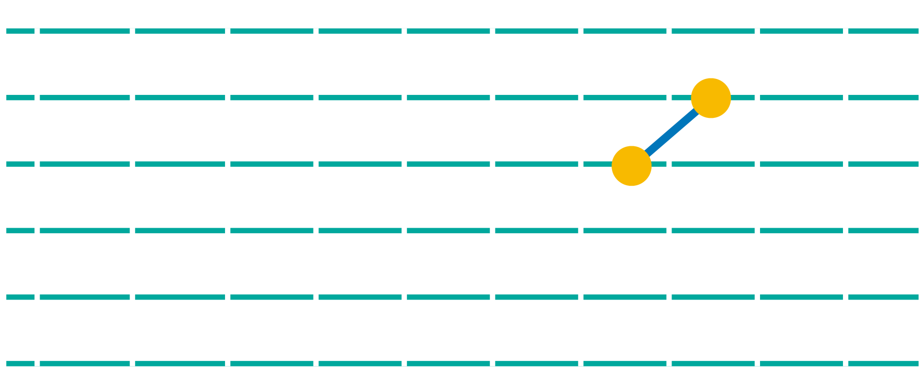
Geometrical requirements on modules in different layers.



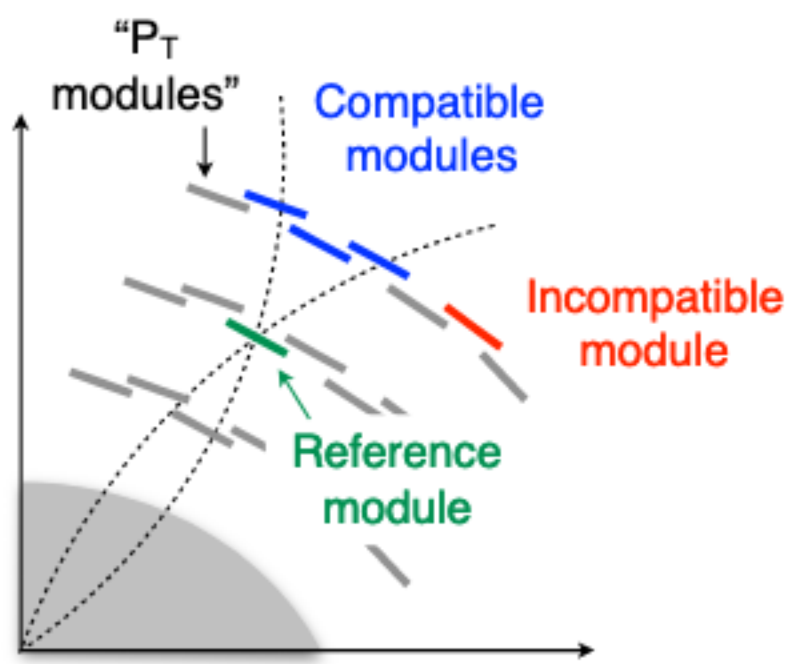


Stub + Stub

= **Segment**

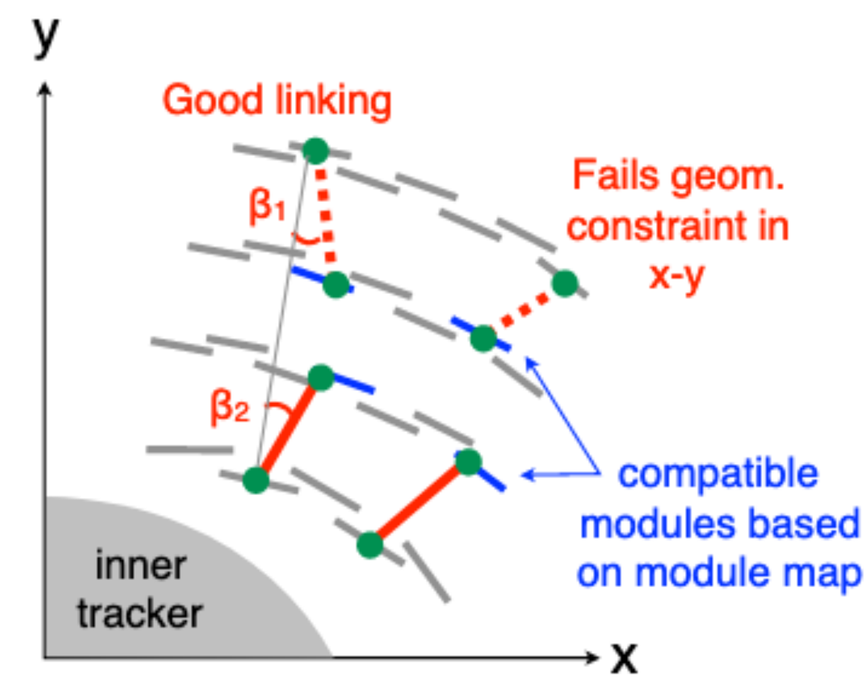


Linking Segments

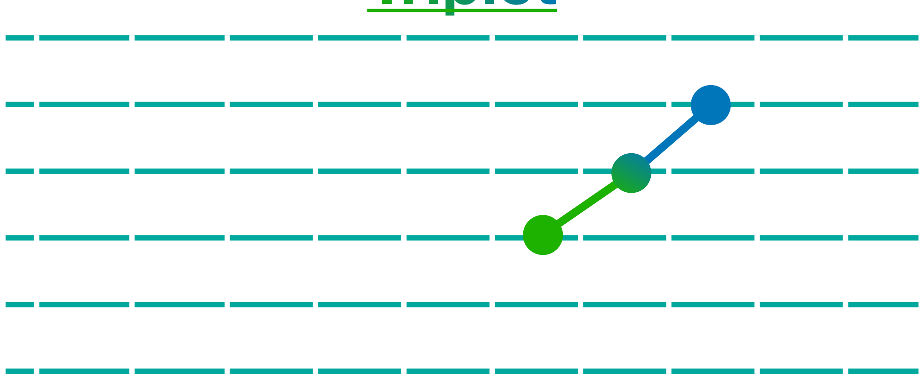


Combine:
- **Compatible modules** info
- **Geometrical requirements**

to link segments into **longer objects** with **common stub.**

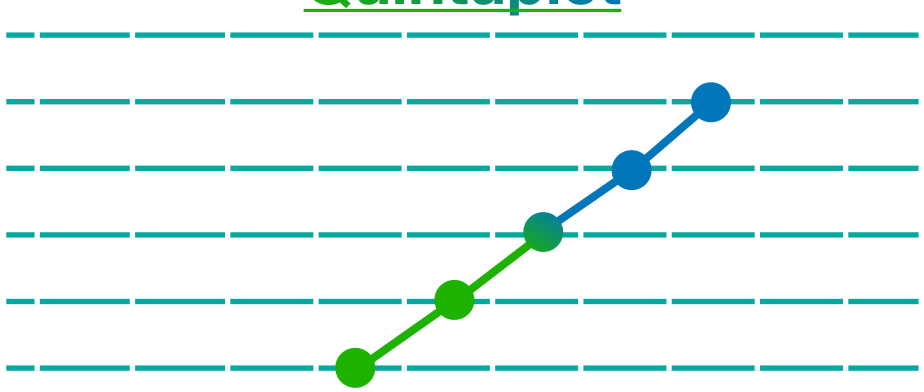


Segment + Segment = **Triplet**

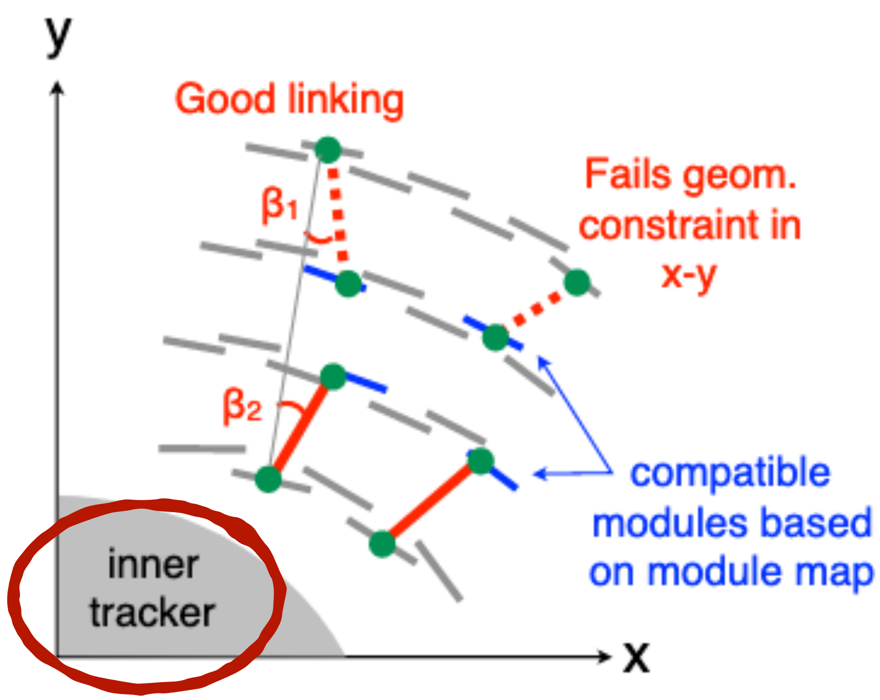


Outer Tracker

Triplet + Triplet = **Quintuplet**



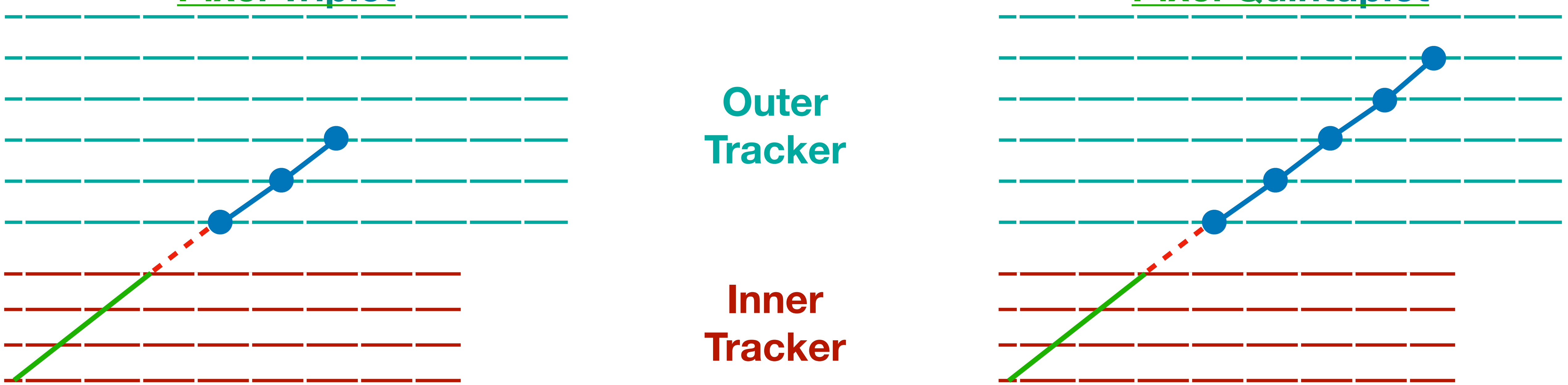
Linking Pixel Segments



Combine **outer tracker** with **inner tracker** objects ⇒ **Cleaner tracks**

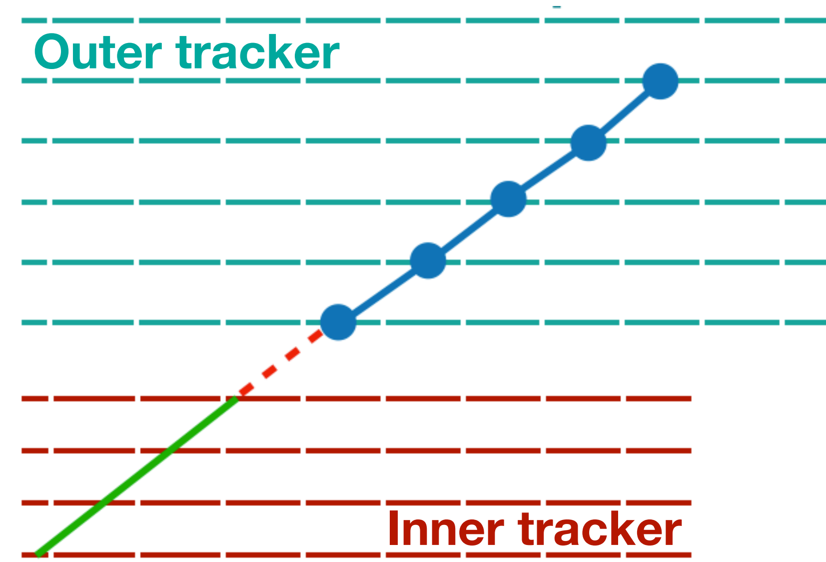
Pixel Segment + Triplet =
Pixel Triplet

Pixel Segment + Quintuplet =
Pixel Quintuplet



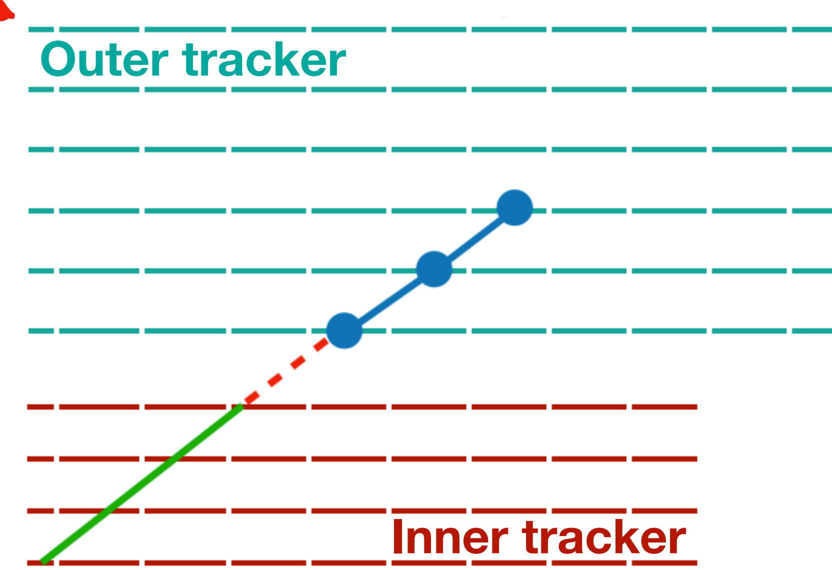
Track Candidate Collection

- Final **Track Candidate (TC)** collection includes:
  - **Pixel Quintuplets (pT5)**
    → **Performance driver**.

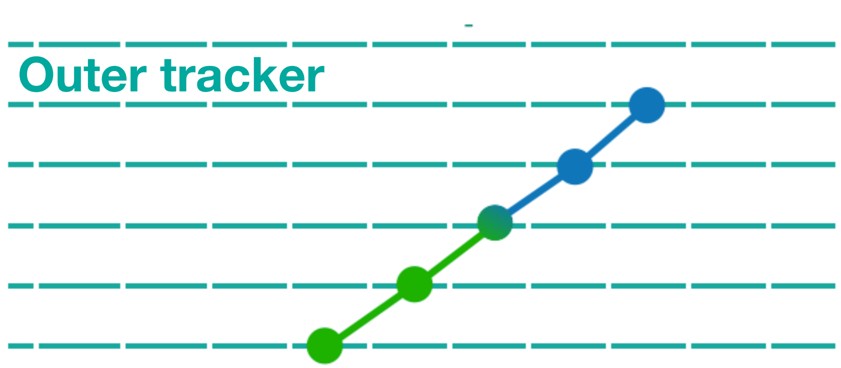


  - Unlinked **Pixel Segments (pLS)**
    → Important at low p_T and **high |η|**.

  - **Pixel triplets (pT3)**
    → Regain **low p_T** efficiency.

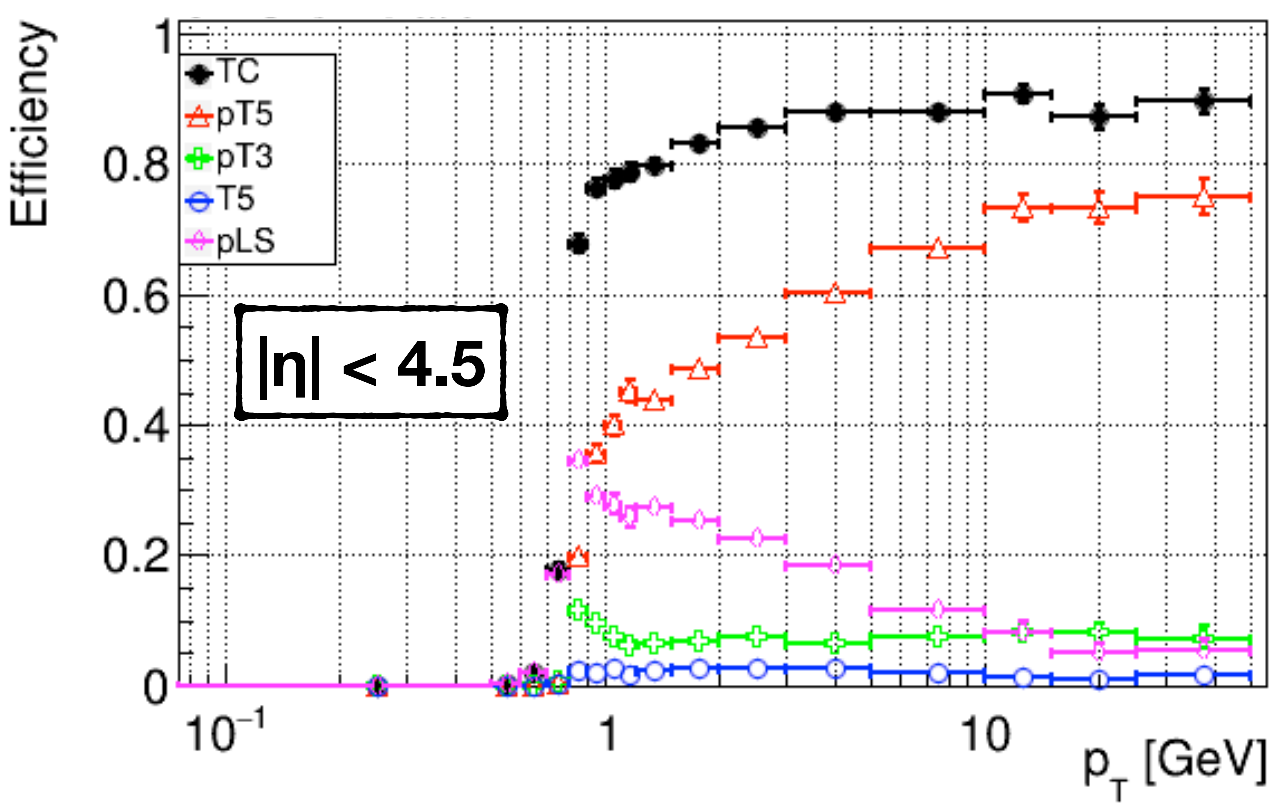


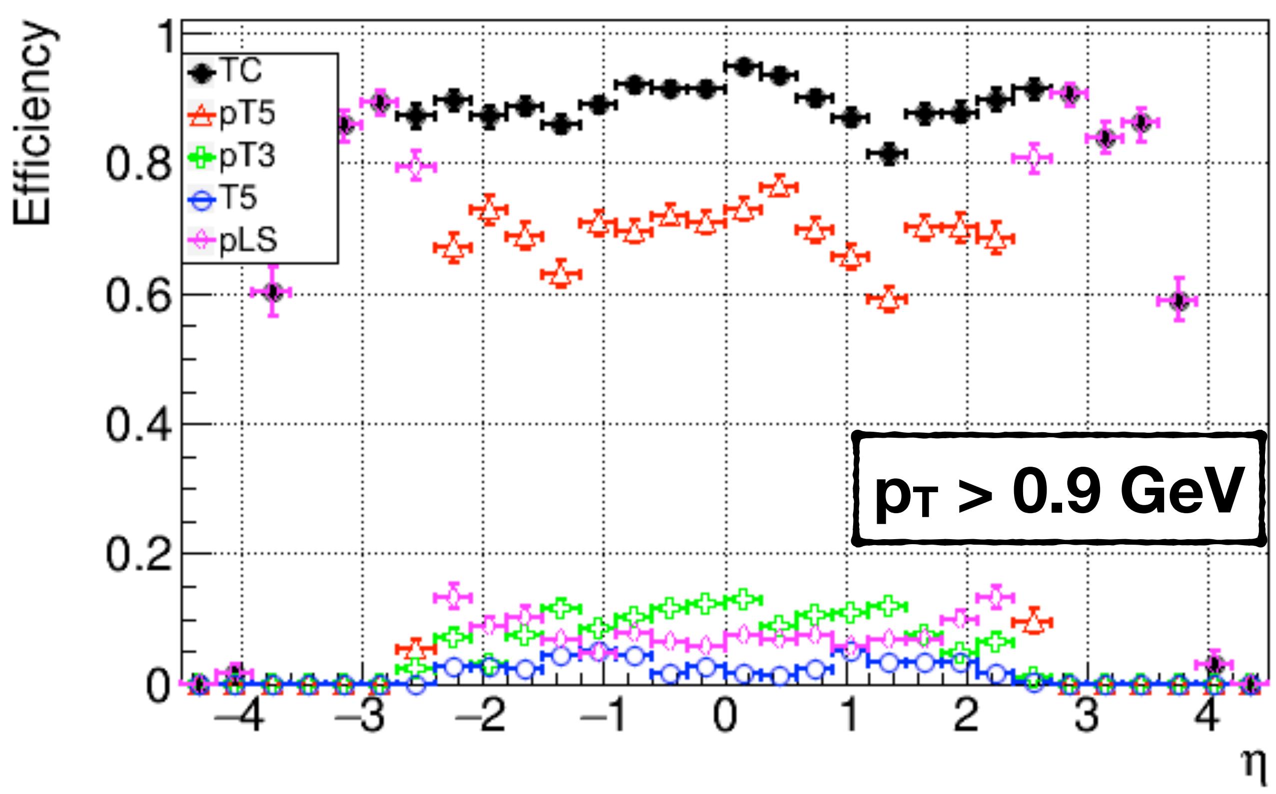
  - **Quintuplets (T5)**
    → Important for **displaced** tracks.



Physics Performance: Efficiency

- Efficiency vs. p_T & η checked in $t\bar{t}$ sample in PU200.
- Denominator: All simulated tracks passing:
  - $|d_z| < 30$ cm
  - $|d_{xy}| < 2.5$ cm
- Numerator: Simulated tracks matched to TC track (>75% hits).

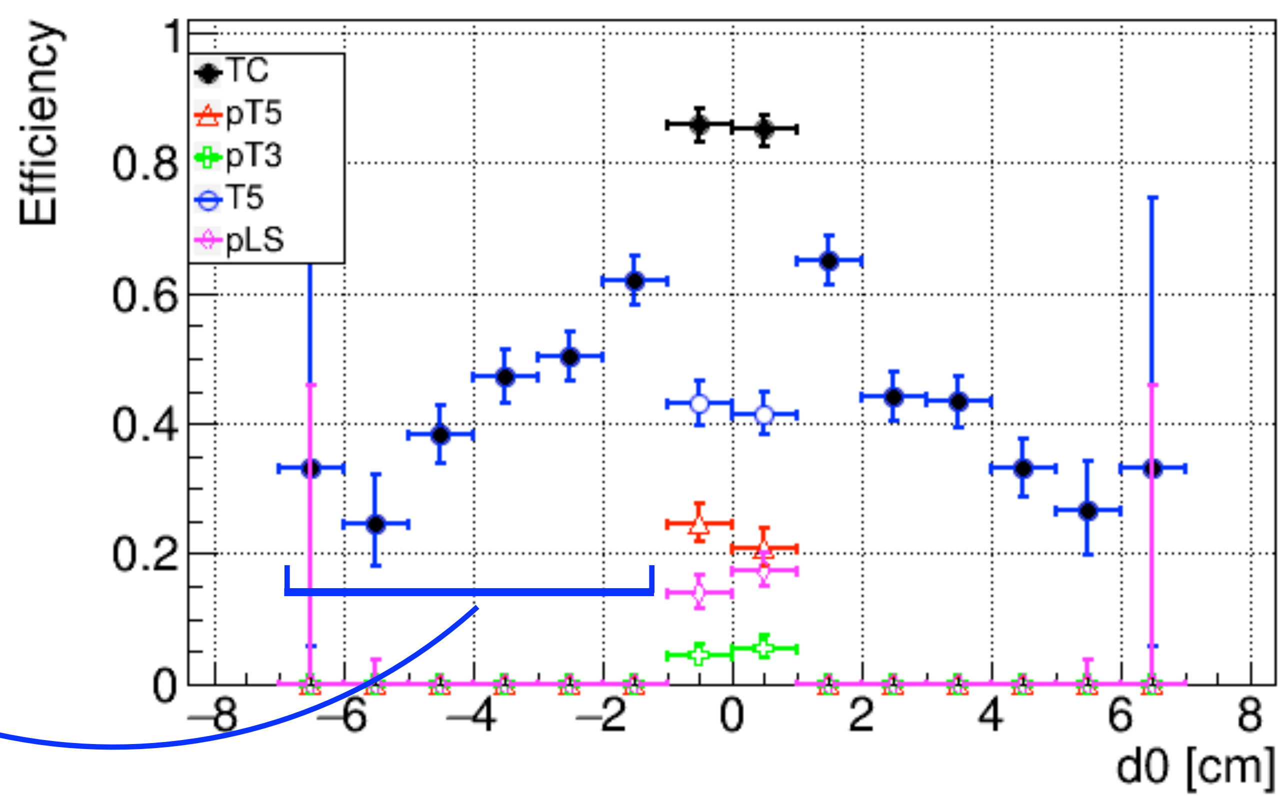




Physics Performance: Efficiency

- Efficiency vs. d_{xy} checked in muon gun sample with displaced vertex uniformly distributed in a (5 cm)^3 cube around the interaction point.
- Denominator: All simulated tracks passing:
  - $|d_z| < 30$ cm
- Numerator: Simulated tracks matched to TC track (>75% hits).

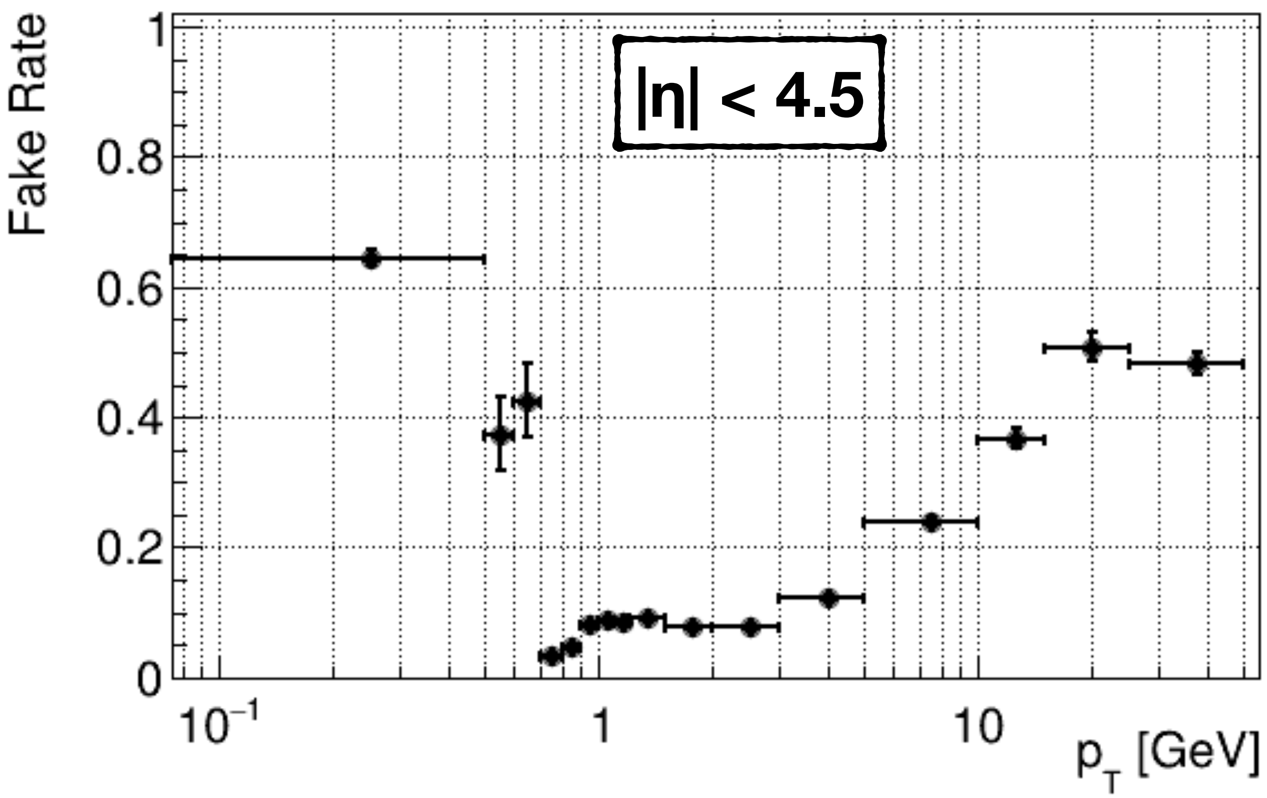
Displaced → T5

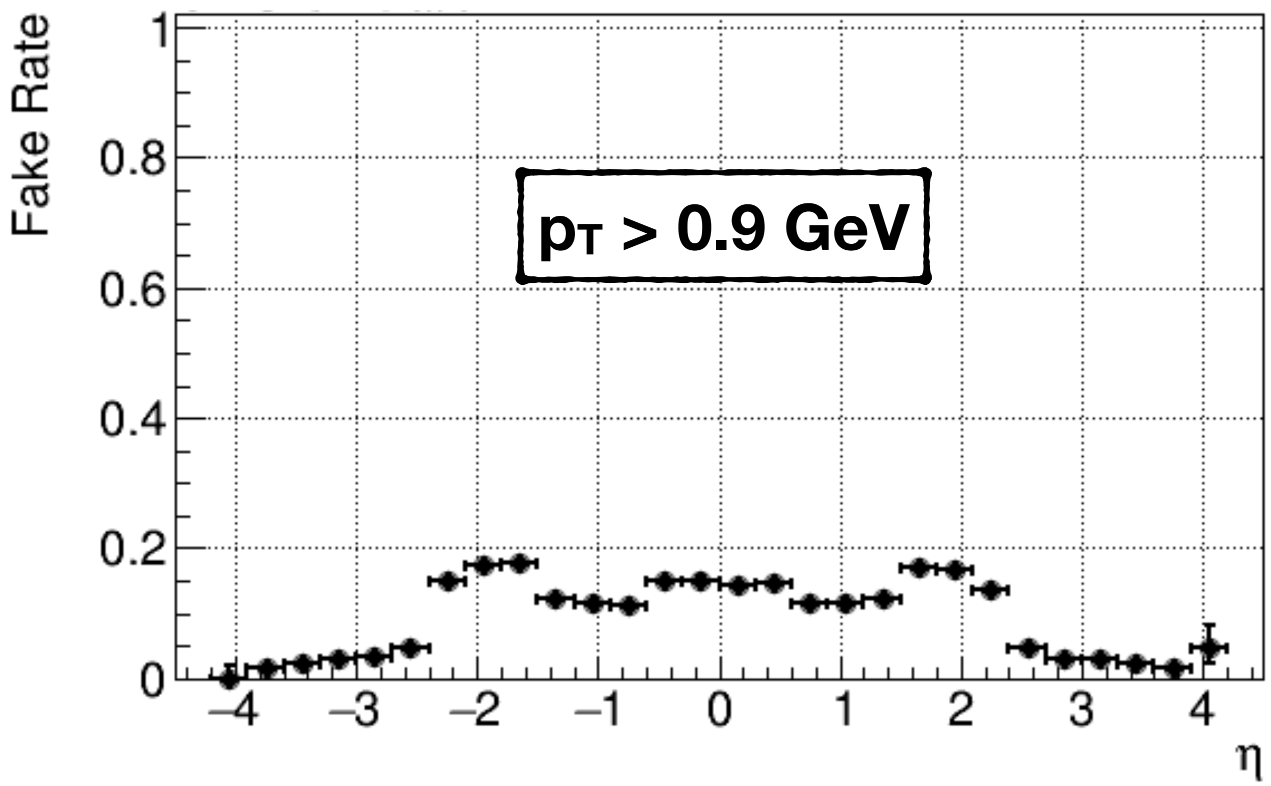


$p_T > 0.9$ GeV
$|\eta| < 4.5$

Physics Performance: Fake Rate

- Fake rate vs. p_T & η checked in $t\bar{t}$ sample in PU200.
- Denominator: All TC tracks.
- Numerator: TC tracks NOT matched to simulated track (>75% hits).





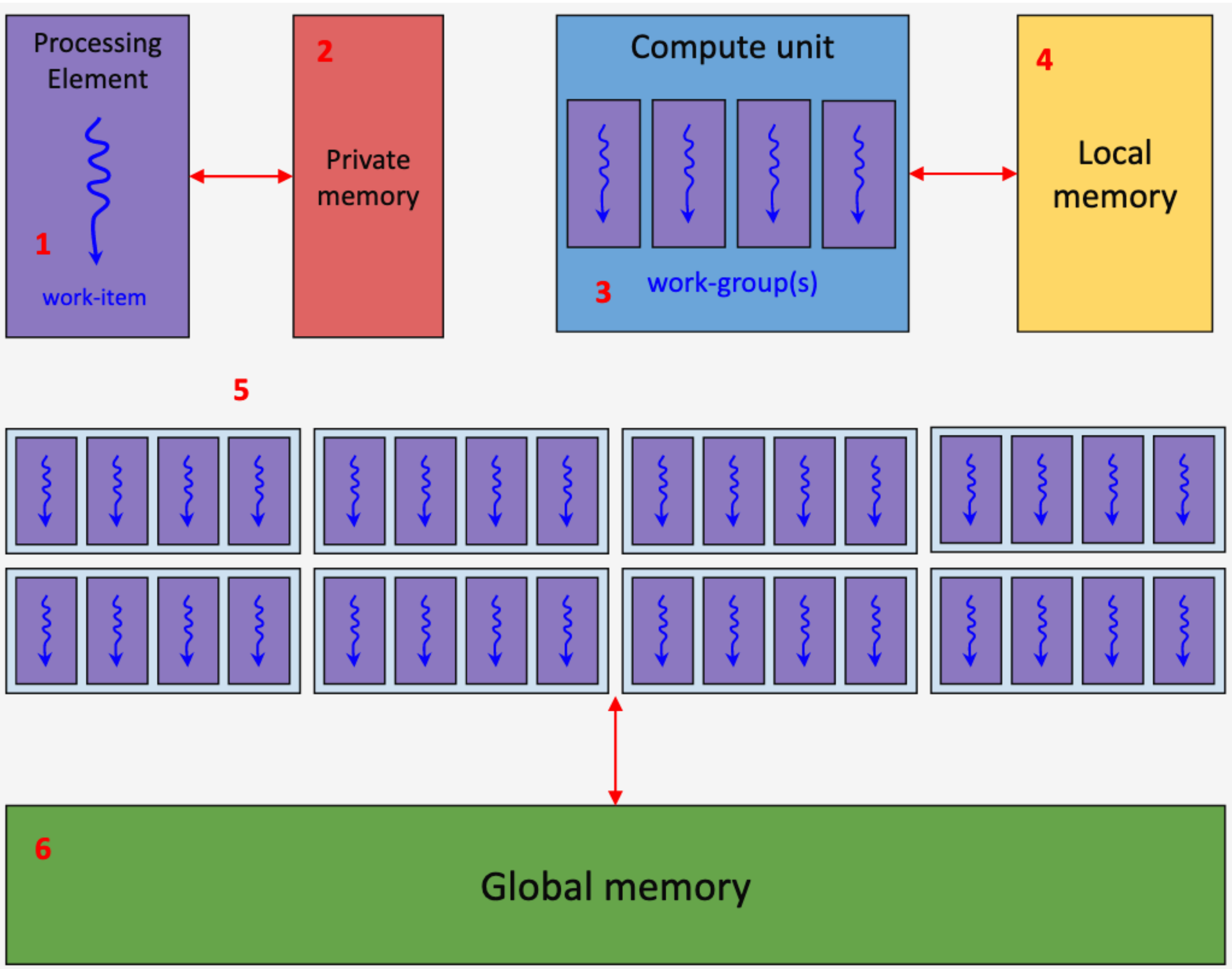
GPU Architecture

-

CPU	GPU
$\mathcal{O}(64)$ cores	$\mathcal{O}(100)-\mathcal{O}(1000)$ cores
~2 Flops (Intel Xeon)	15 Flops (NVIDIA Tesla V100)
Low latency	High latency (device-host memory transfers)
Good for serial processing	Good for parallel processing

⇒ GPU programming requires:

- Suitable algorithm (=parallelizable).
- Programming tricks (≈clever memory usage).



- Efficiently take advantage of:
  - Multitude of computing units.
  - Different memory levels (size vs. speed trade-off).

GPU Implementation

- Data stored in Structure of Arrays (**SoA**) format ⇒ Appropriate for parallelized access.

- **Custom cache allocation** for optimized memory assignment.

- Truncate number of objects based on final occupancy ⇒ **Minimize memory footprint.**

- **Full implementation** on NVIDIA Tesla V100: https://github.com/SegmentLinking/TrackLooper

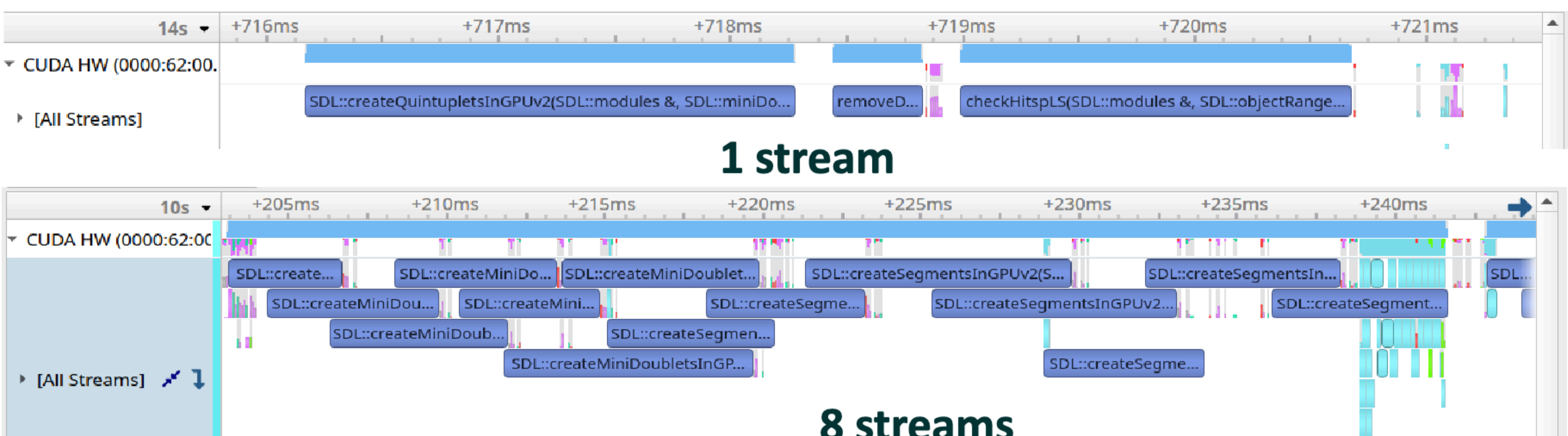
GPU Implementation

- **Kernels**: Independent set operations on separate processing element.

Stubs → Segments → Triplets → Quintuplets → Pixel Segments → Pixel Quintuplets → Pixel Triplets → Track Candidates

- Each kernel = Single Instruction, Multiple Data (**SIMD**) → Same steps applied to different objects → **Local nature of Segment Linking**.

- Speed up by **multi-streaming**: Run parts of kernels in different streams.
  - Memory transfers also "hidden" by concurrent computations.



Timing

- Timing measurements in $t\bar{t}$ sample in PU200.
  - Only algorithm execution →
    Final device to host memory transfer excluded.
- 1-stream: t ≈ 32 ms
- 8-stream: t ≈ 26 ms
  - ~20% timing improvement by exploiting multi-streaming.
- Comparable with latest CMS tracking (~40 ms) on CPU.
- Comparable also price-wise →
  2× 32-core Skylake Gold Xeon processors ≈ NVIDIA Tesla V100.

Future Updates

- Physics-wise:
  - Stretch performance to its limits →
    Further selection optimization for **better efficiency** and **lower fake rate**.
  - Improved **displaced tracks** reconstruction.
  - **Incorporate** latest developments (e.g. **patatrack** pixel tracks)
- Computing-wise:
  - **Optimize** algorithms for complex physics **parameter calculations**.
  - Extend usage of alternate data types → **Half precision**.
  - Improve **memory coalescence** ⇒ Timing reduction.
- The Great Design:
  CMS software integration for **HLT and offline usage**.

Summary

- Segment Linking:
  - A highly **parallelizable track reconstruction** algorithm.
  - Aimed at **facing HL-LHC challenges**.
  - Successfully **implemented on GPU**.
  - **Competitive physics performance** (efficiency and fake rate).
  - **Timing comparable** with current CMS HLT tracking algorithms.
  - More improvements and CMS integration planned for the future.