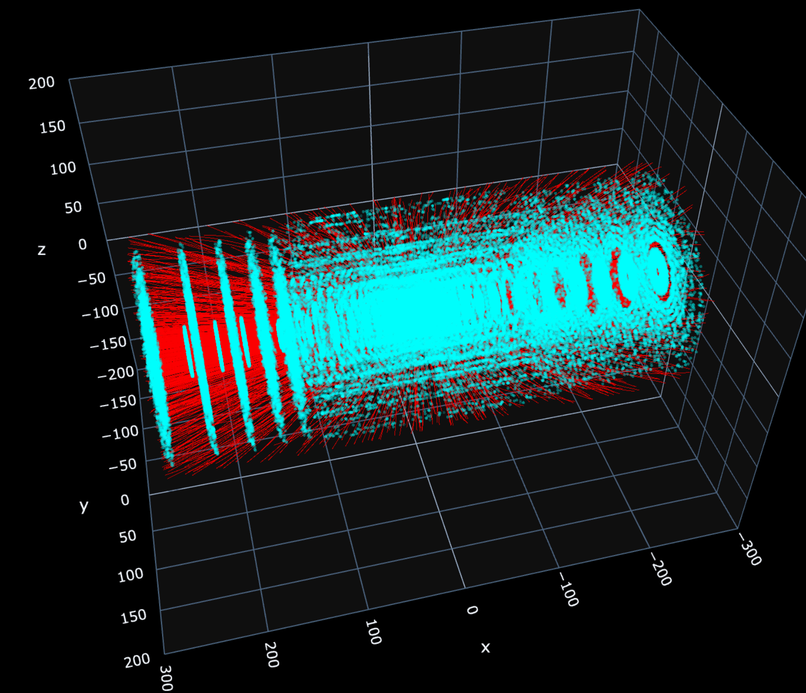
Line Segment Tracking at CMS

Computing in High Energy & Nuclear Physics (CHEP)
May 8, 2023



Philip Chang

Y. Gu, V. Krutelyov, B. Venkat Sathia Narayanan,
M. Tadel, E. Vourliotis, A. Yagil

B. Wang, P. Elmer

T. Reid, G. Niendorf, P. Wittich

UF UNIVERSITY of FLORIDA

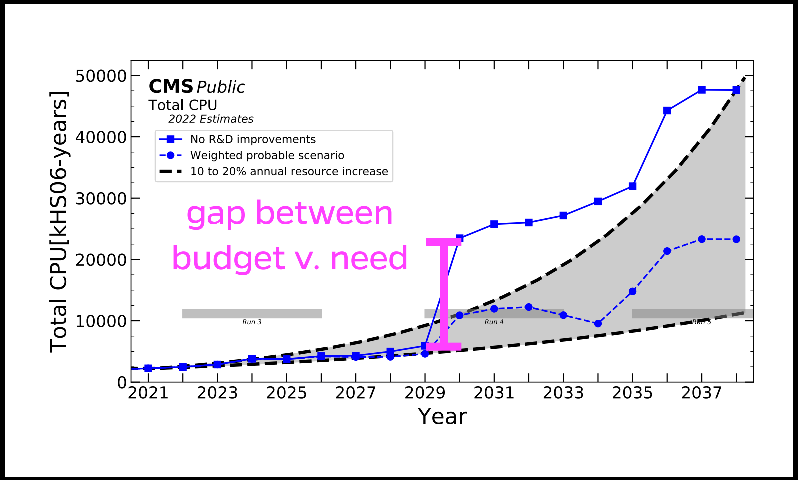
UC San Diego

PRINCETON UNIVERSITY

Cornell University

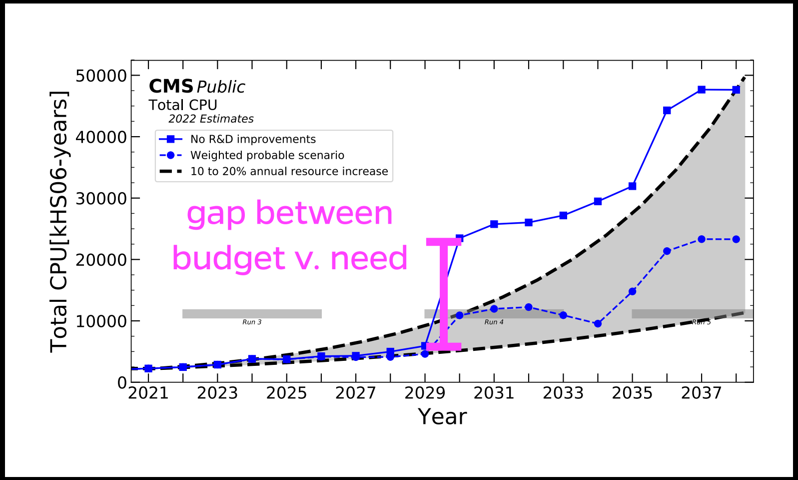
On behalf of CMS Collaboration

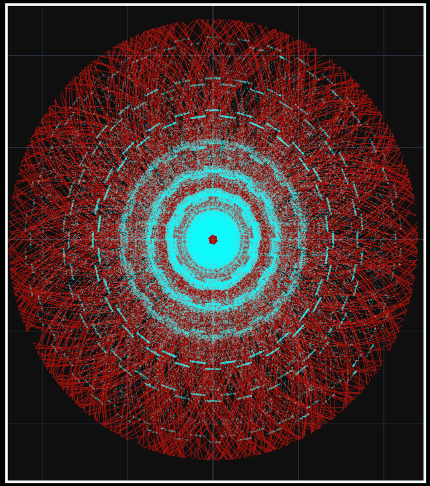
Outline



In 2029, new era of data taking starts and **we must R&D**

Outline

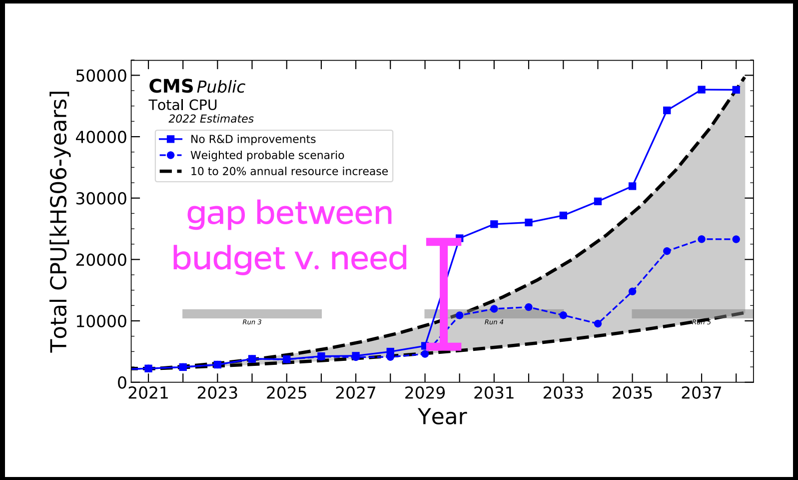




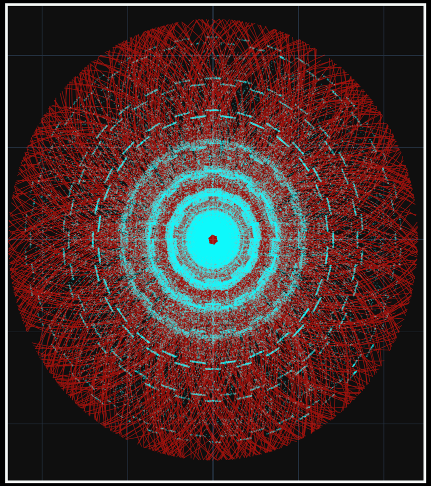
In 2029, new era of data taking starts and **we must R&D**

One of the main culprits is tracking due to **combinatorics**

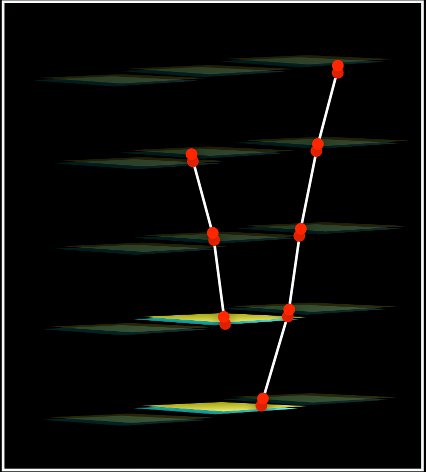
Outline



In 2029, new era of data taking starts and **we must R&D**

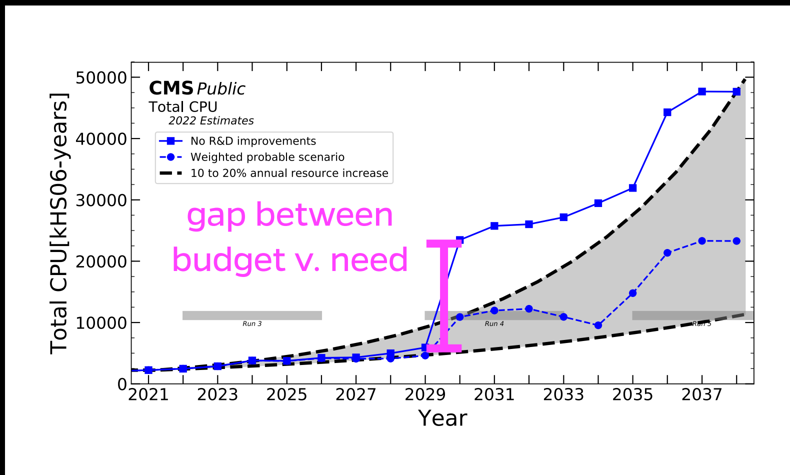


One of the main culprits is tracking due to **combinatorics**

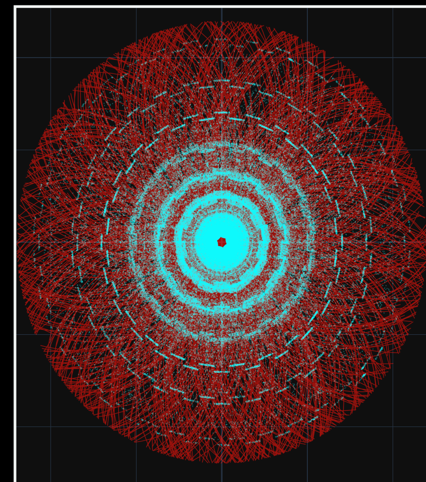


We present a new tracking algorithm called **“Line-Segment Tracking” (LST)**

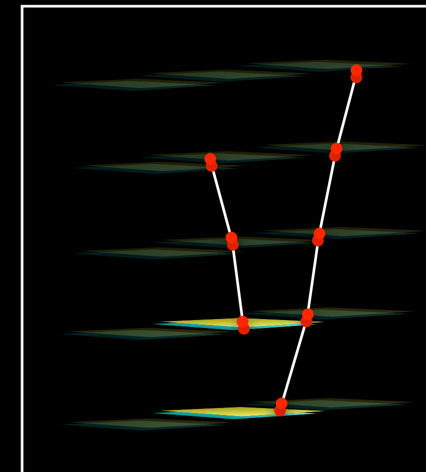
Outline



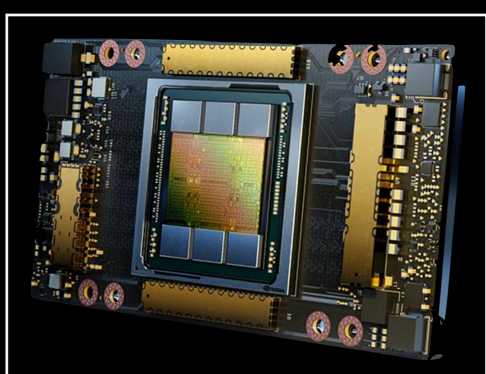
In 2029, new era of data taking starts and **we must R&D**



One of the main culprits is tracking due to **combinatorics**

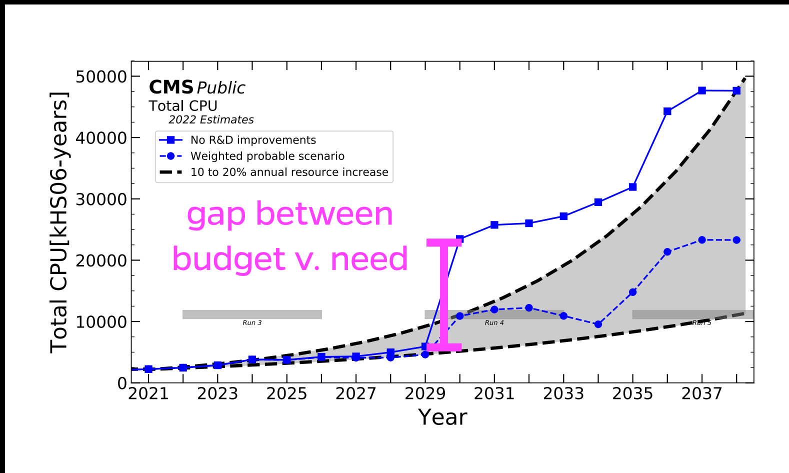


We present a new tracking algorithm called **"Line-Segment Tracking" (LST)**

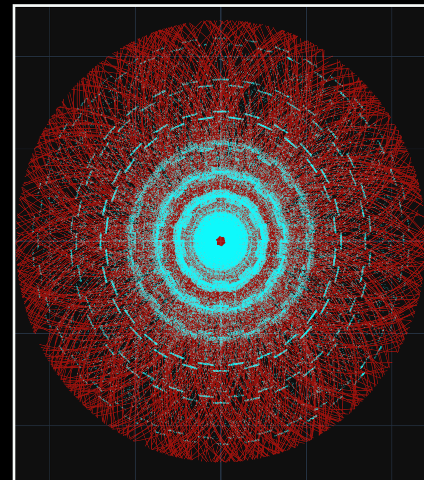


LST is implemented to be **accelerated on GPUs**

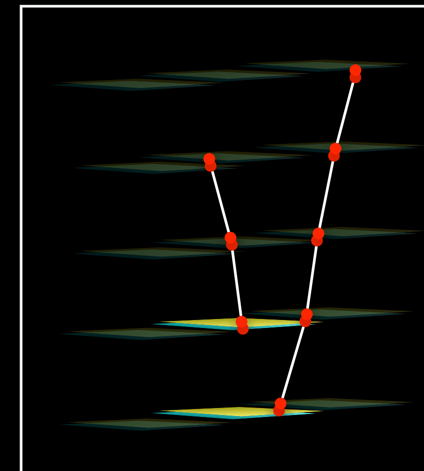
Outline



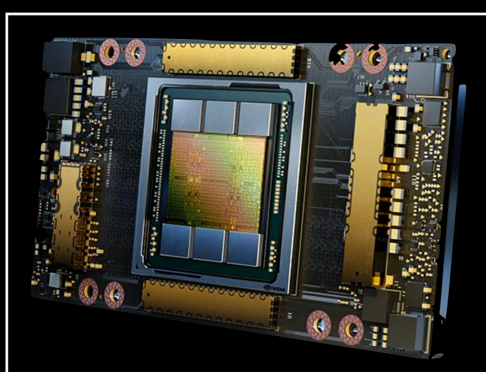
In 2029, new era of data taking starts and **we must R&D**



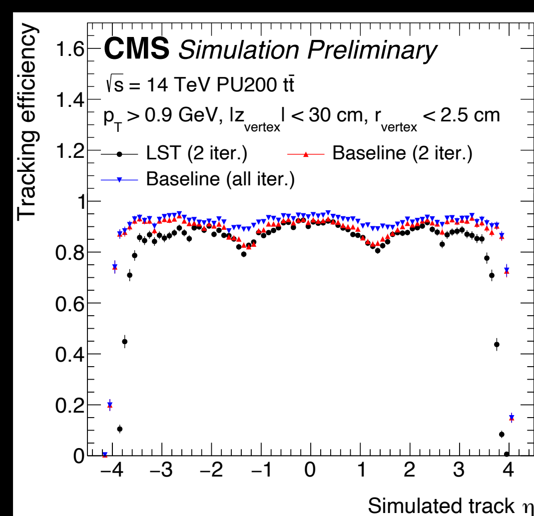
One of the main culprits is tracking due to **combinatorics**



We present a new tracking algorithm called **"Line-Segment Tracking" (LST)**



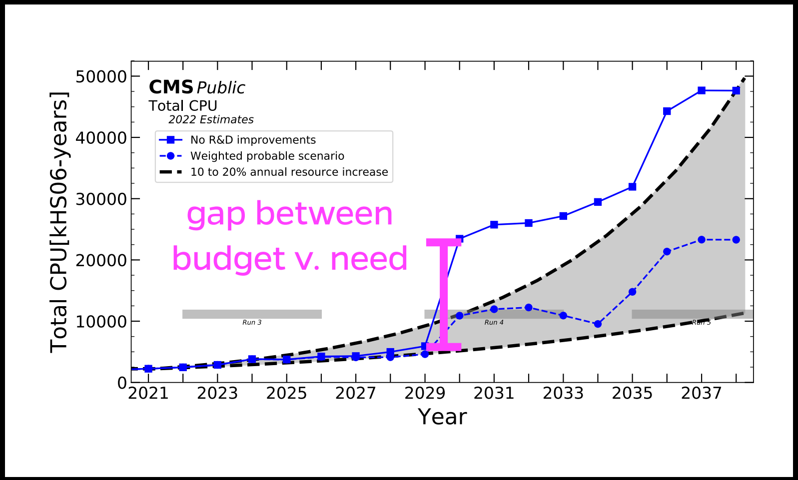
LST is implemented to be **accelerated on GPUs**



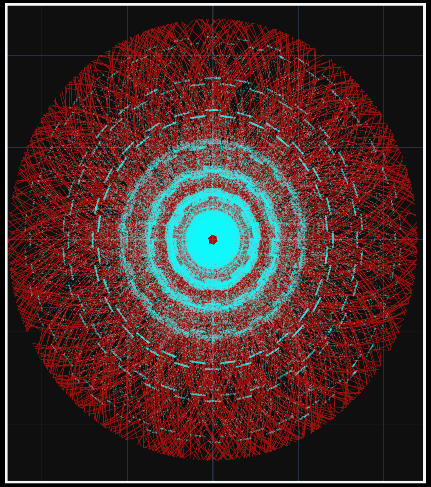
Provides **good performance**

UF
Chang
Florida

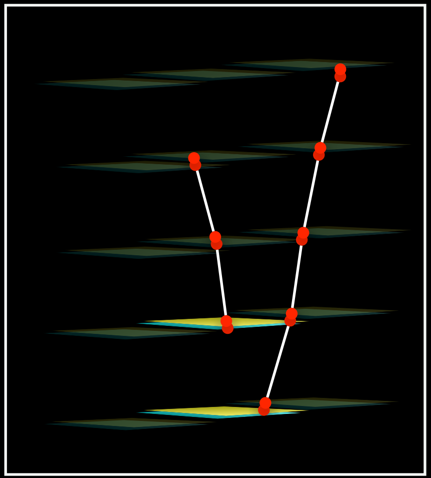
Outline



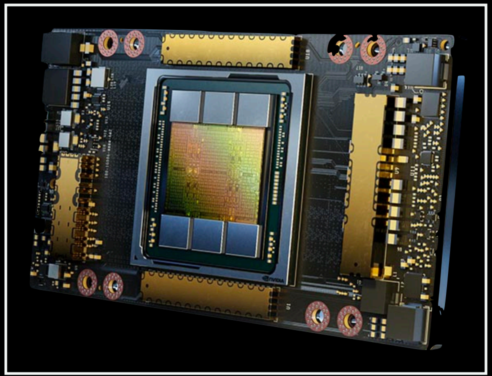
In 2029, new era of data taking starts and **<u>we must R&D</u>**



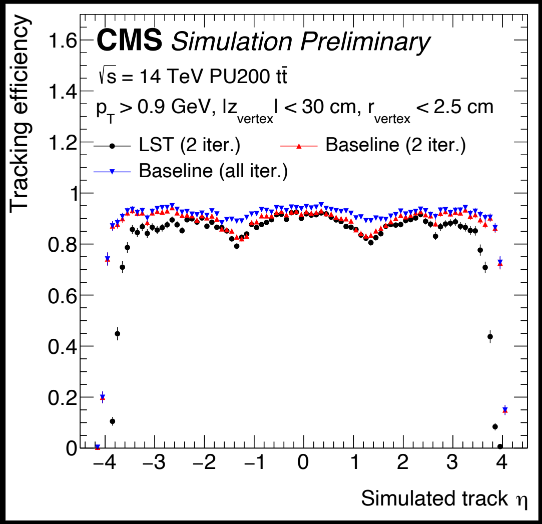
One of the main culprits is tracking due to **<u>combinatorics</u>**



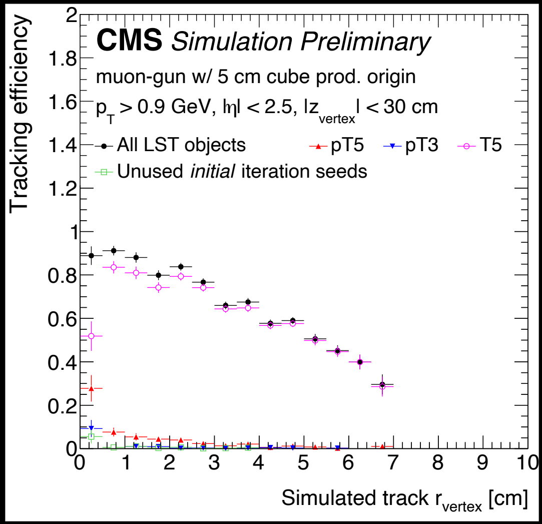
We present a new tracking algorithm called **<u>"Line-Segment Tracking" (LST)</u>**



LST is implemented to be **<u>accelerated on GPUs</u>**



Provides **<u>good performance</u>**



Naturally tracks **<u>displaced particles</u>**

Online vs. Offline / GPU v. xPU

UF
Chang
Florida

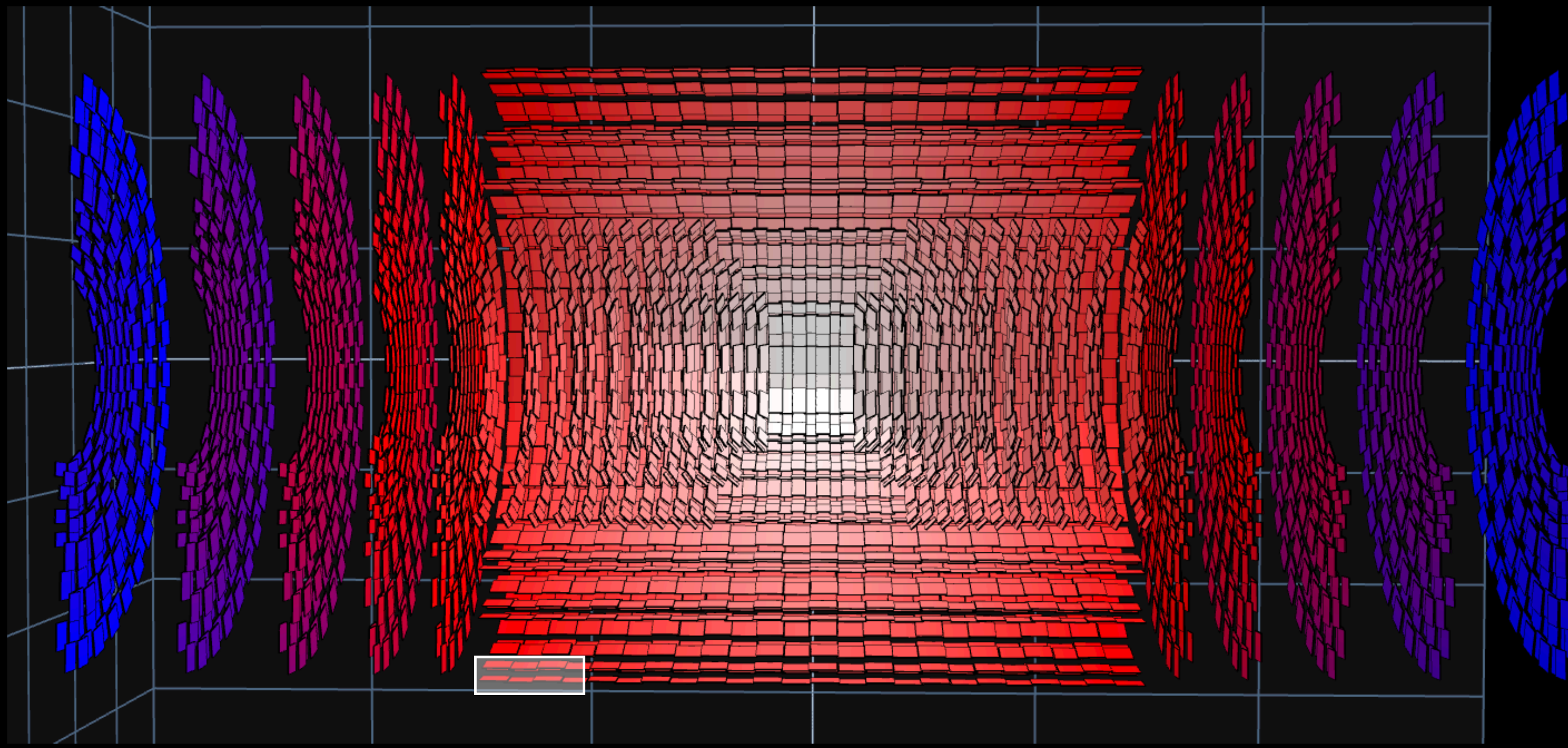
We did not box ourselves to one or the other

We primarily focused on developing algorithms that can run in parallel on processors capable of parallelism

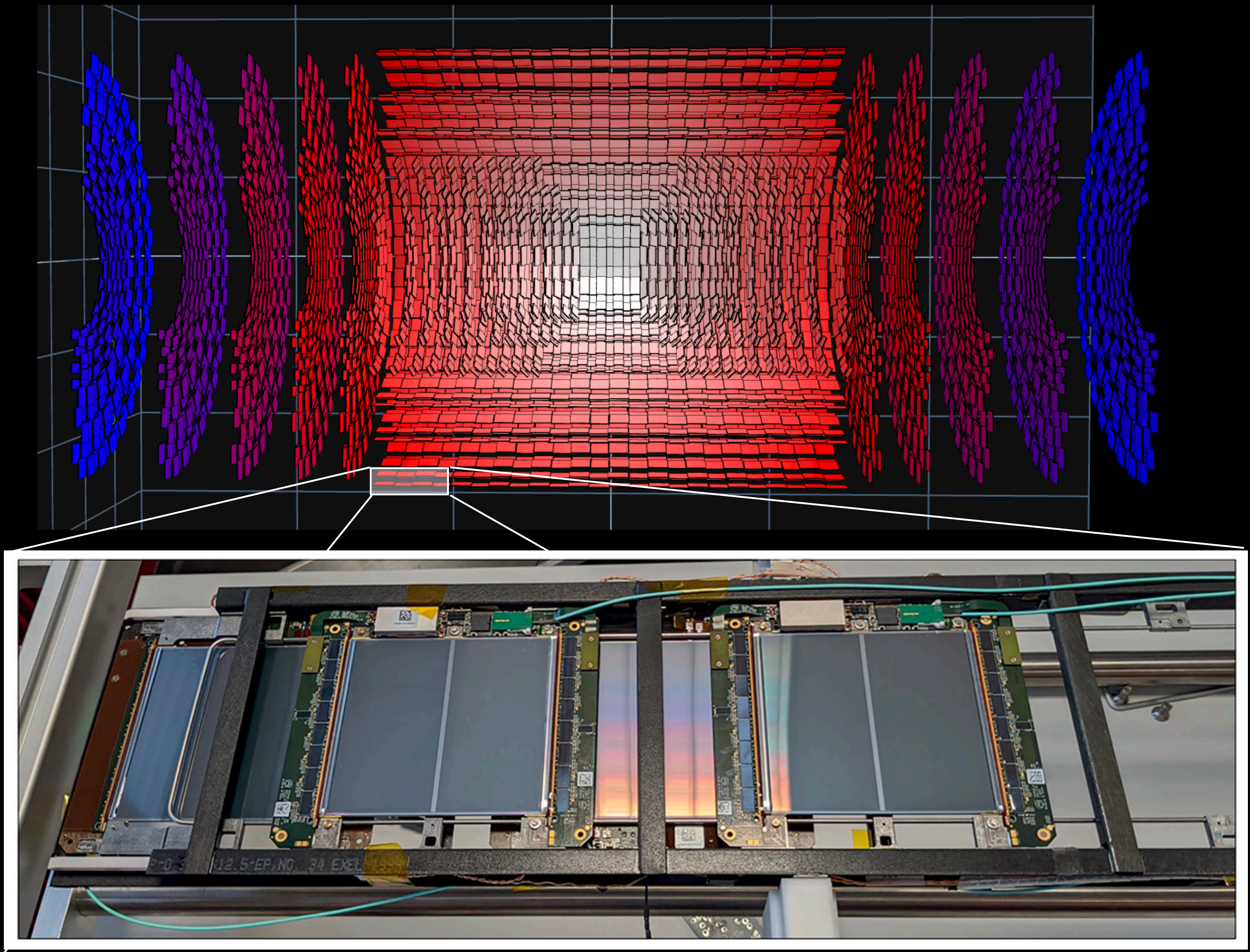
Once we developed the algorithm to a more mature state, we implemented our work on GPUs

But the algorithm itself is parallel in nature and potentially possible to exploit other architecture

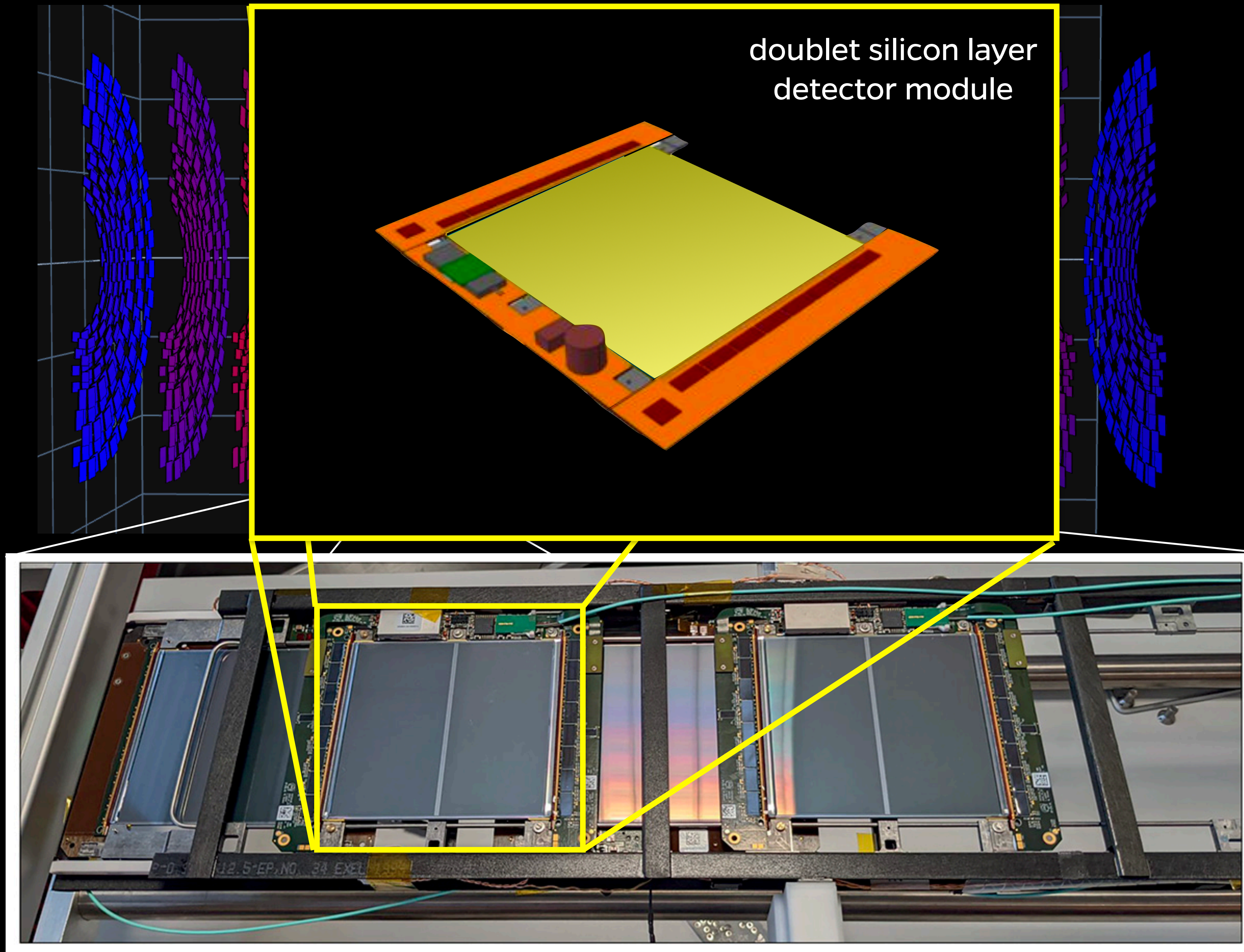
Opportunity at HL-LHC CMS Outer Tracker



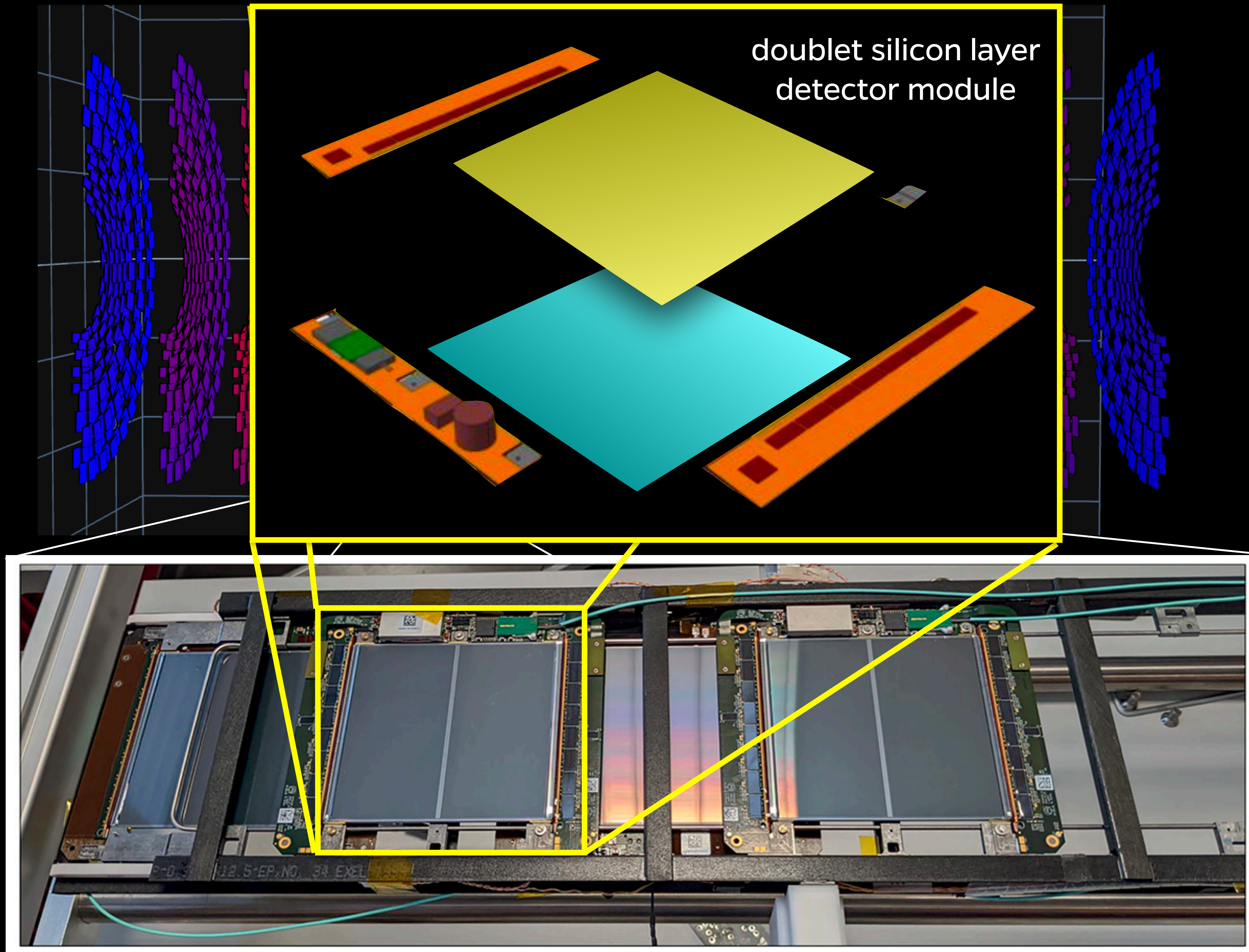
Opportunity at HL-LHC CMS Outer Tracker

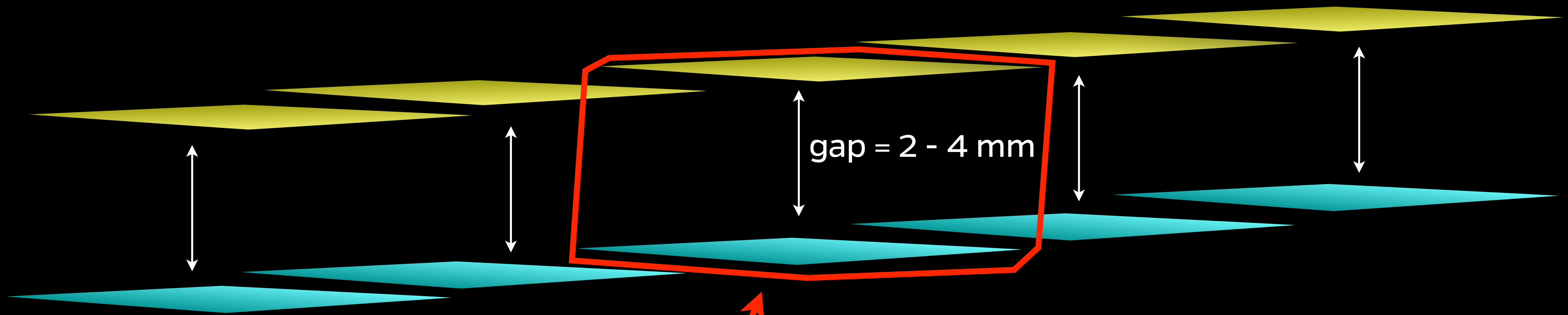


Opportunity at HL-LHC CMS Outer Tracker

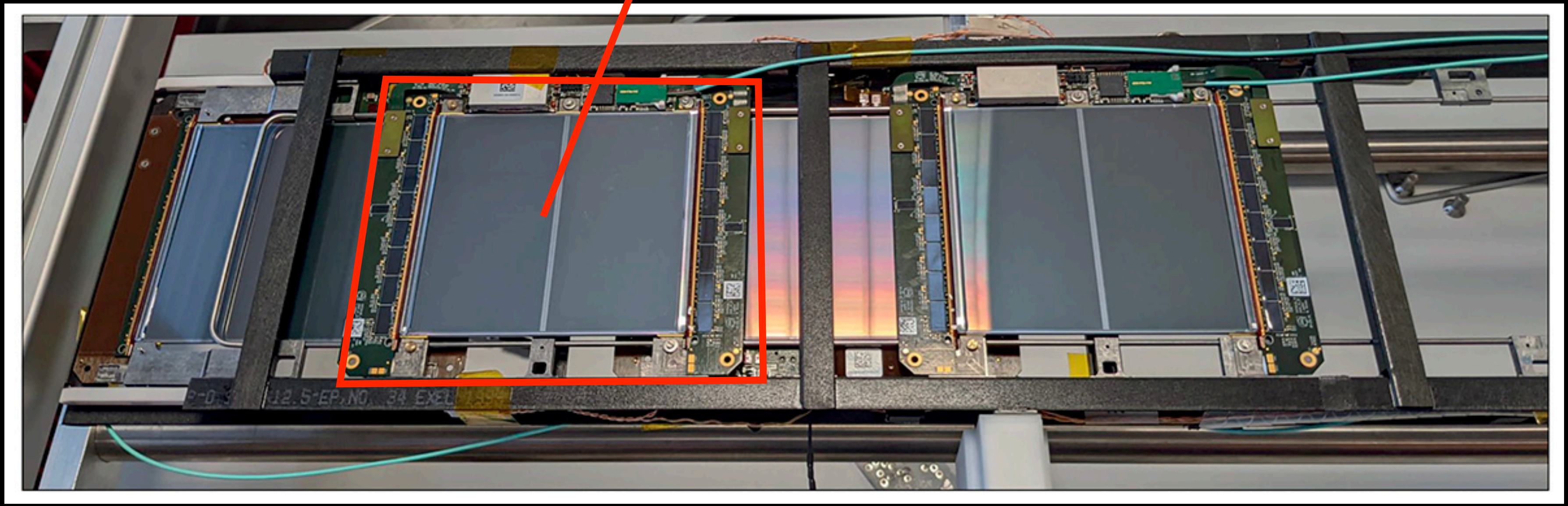


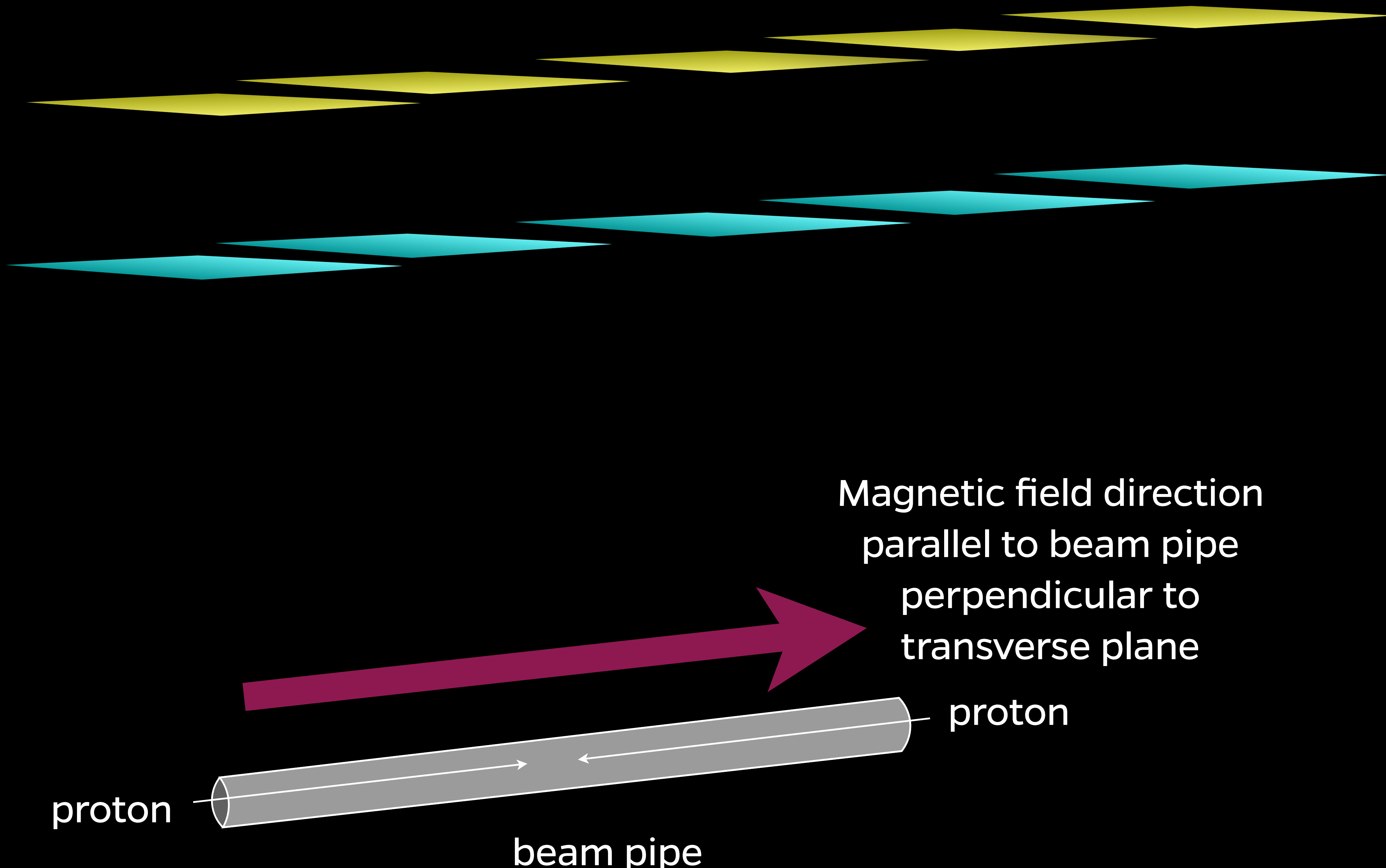
Opportunity at HL-LHC CMS Outer Tracker

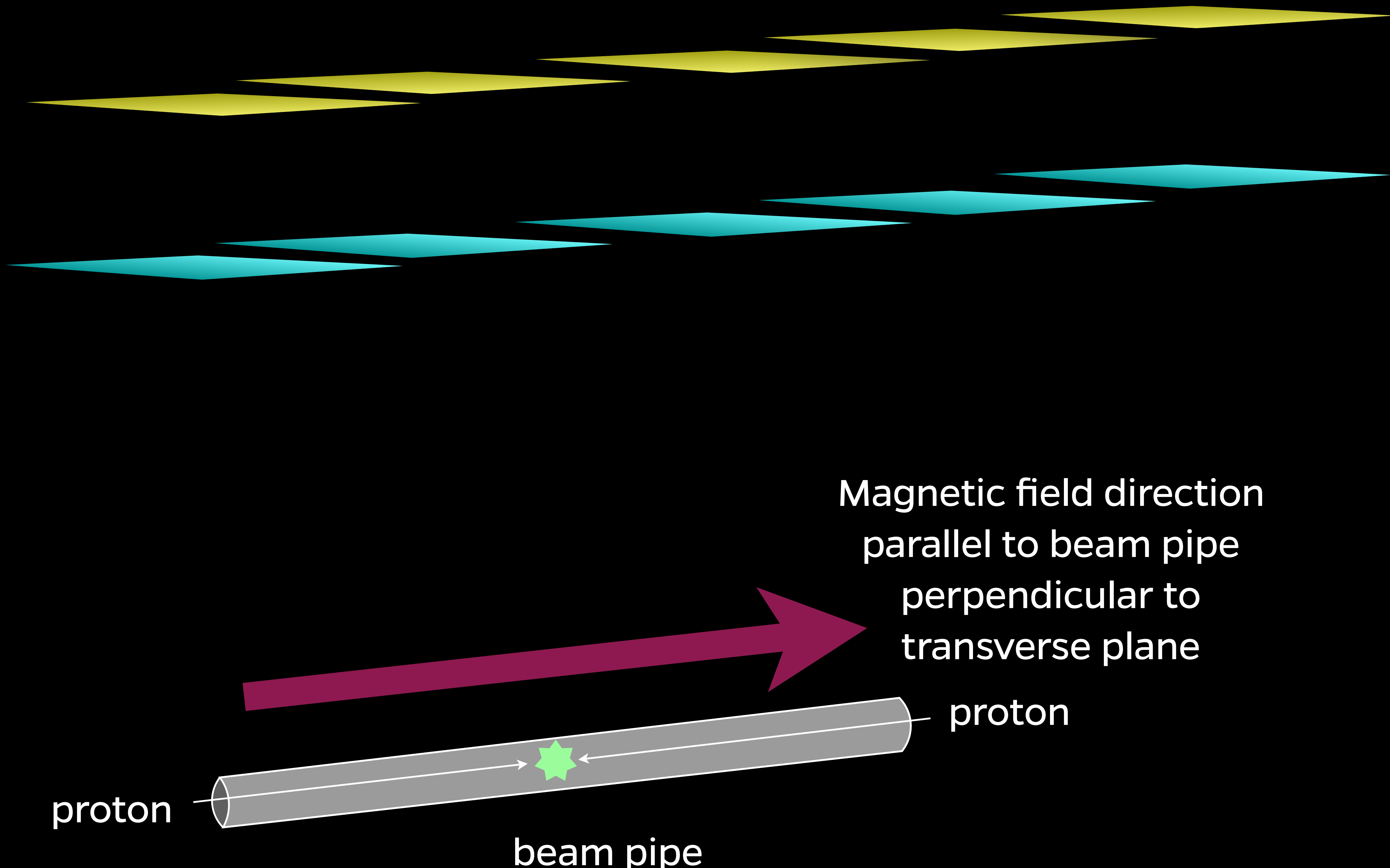


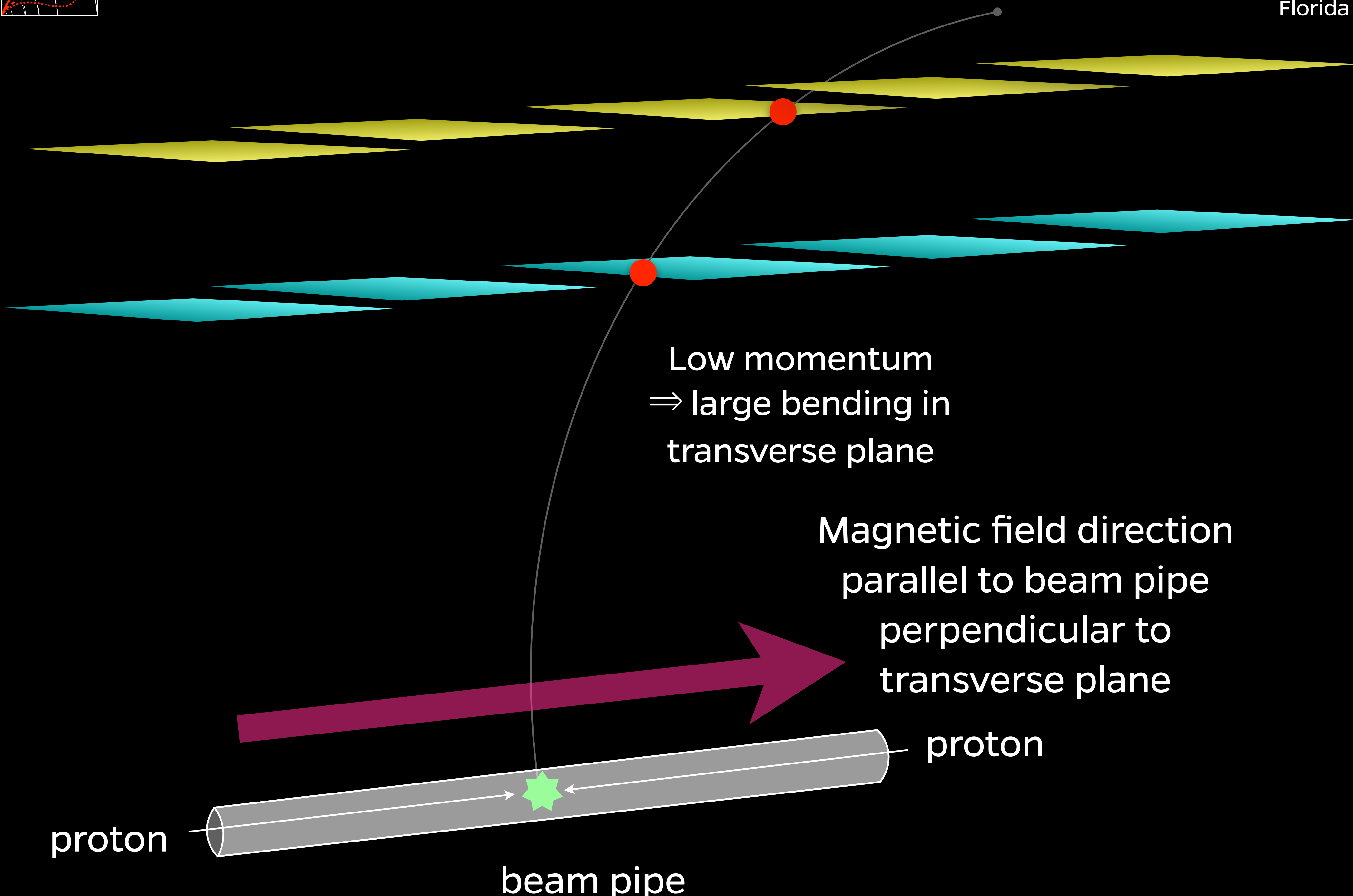


each pair is one module

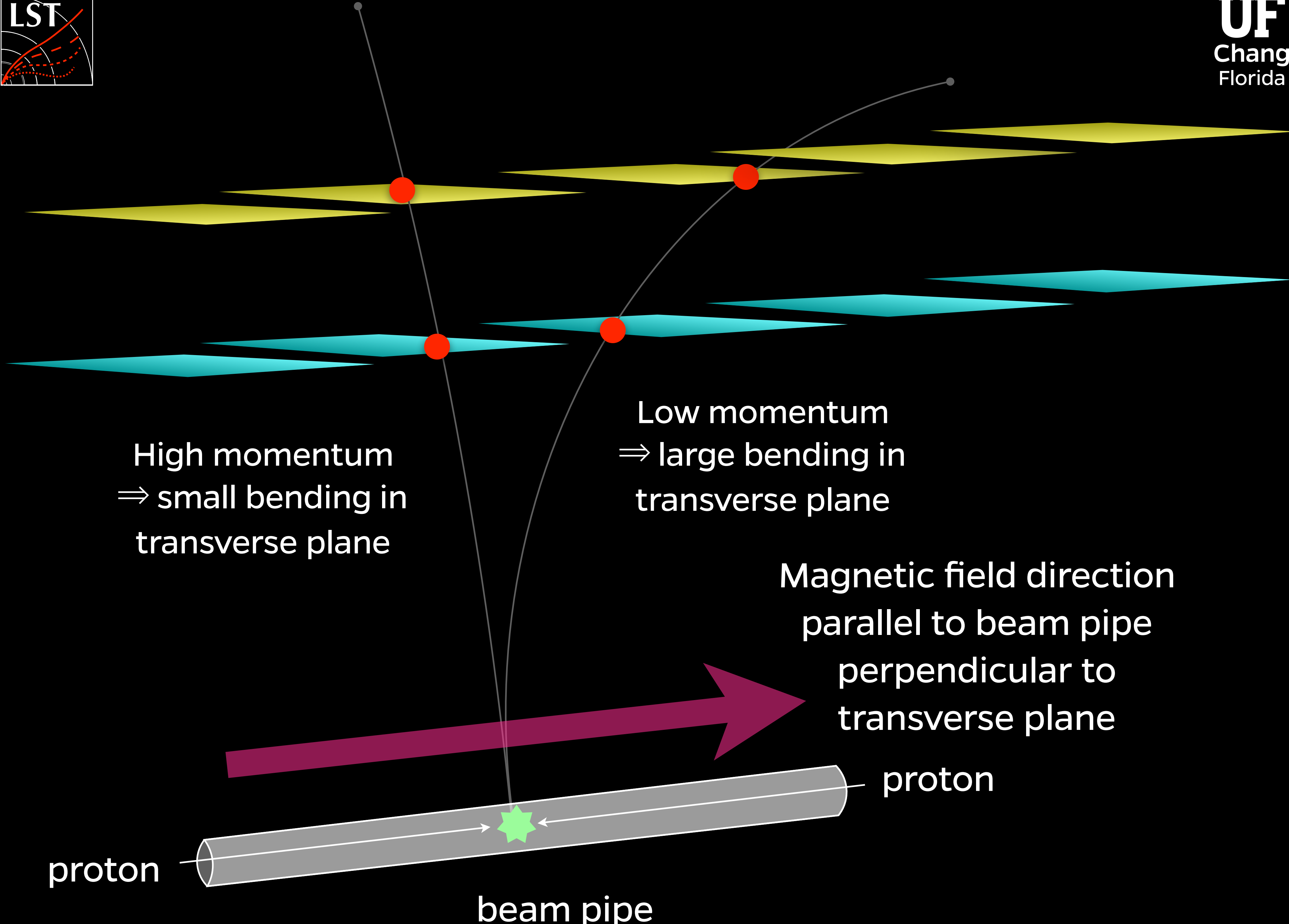



Magnetic field direction
parallel to beam pipe
perpendicular to
transverse plane
proton
proton
beam pipe

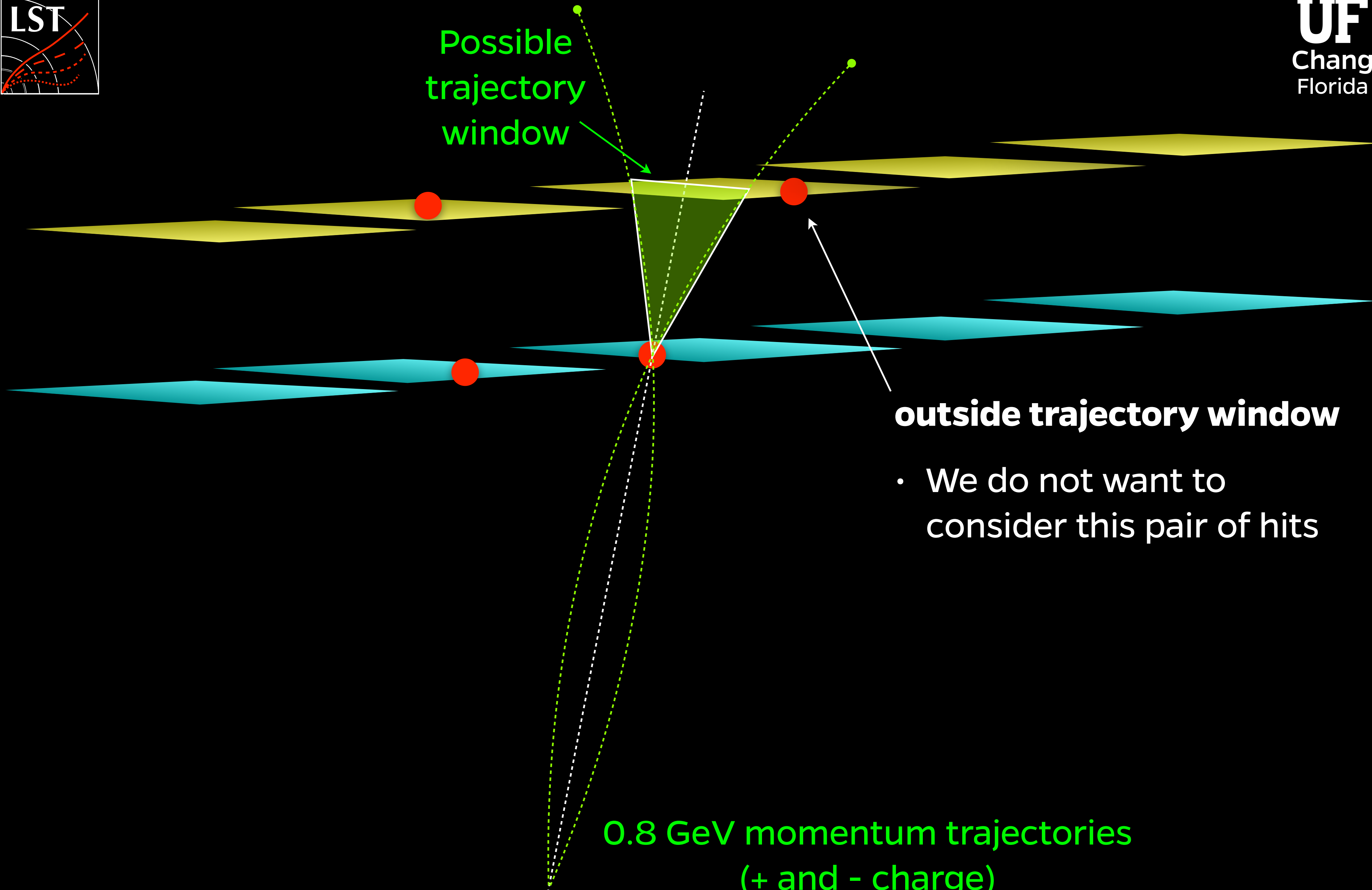

Magnetic field direction
parallel to beam pipe
perpendicular to
transverse plane
proton
proton
beam pipe


Low momentum
⇒ large bending in
transverse plane
Magnetic field direction
parallel to beam pipe
perpendicular to
transverse plane
proton
proton
beam pipe

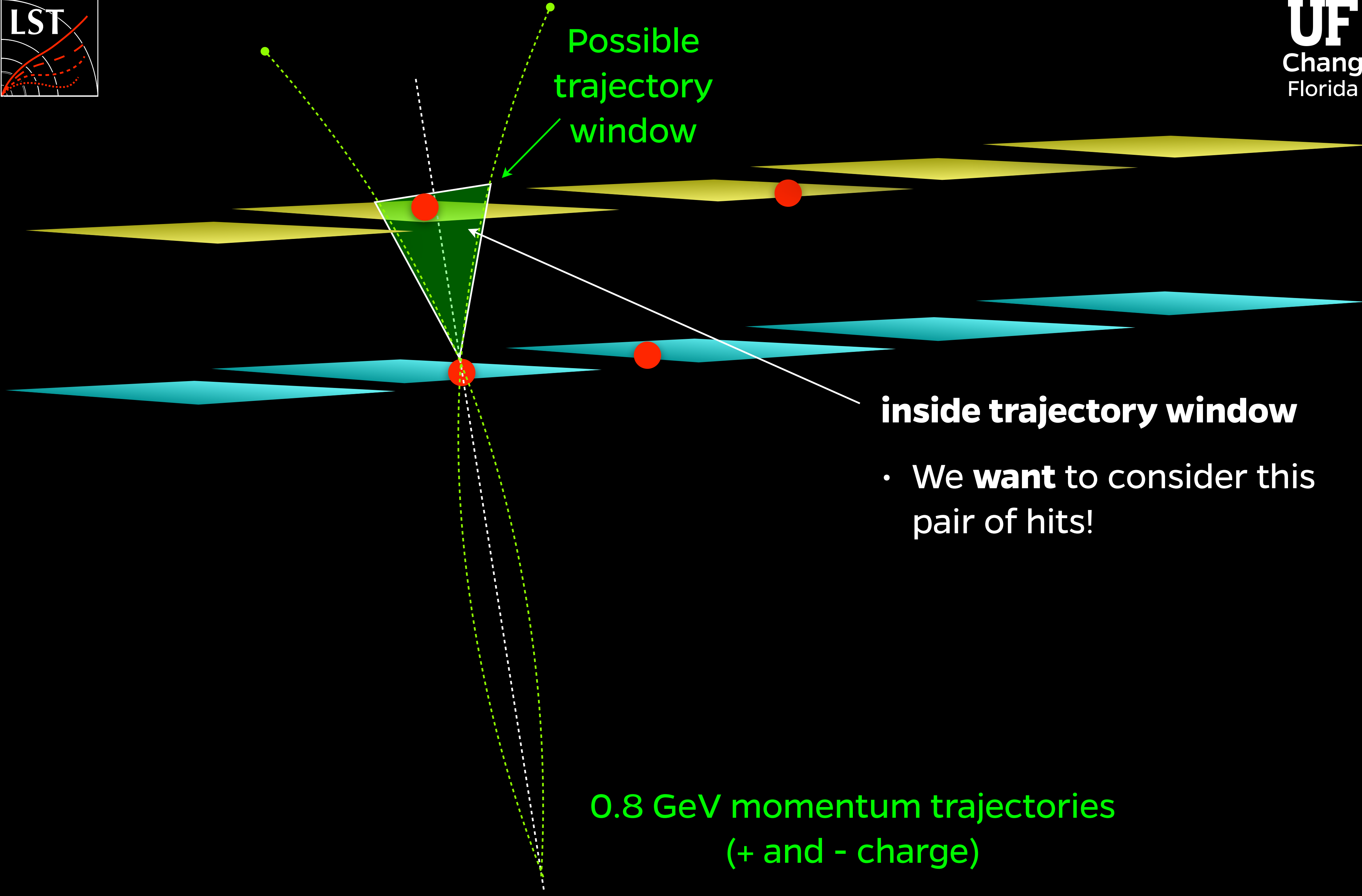
UF
Chang
Florida



Chang
Florida



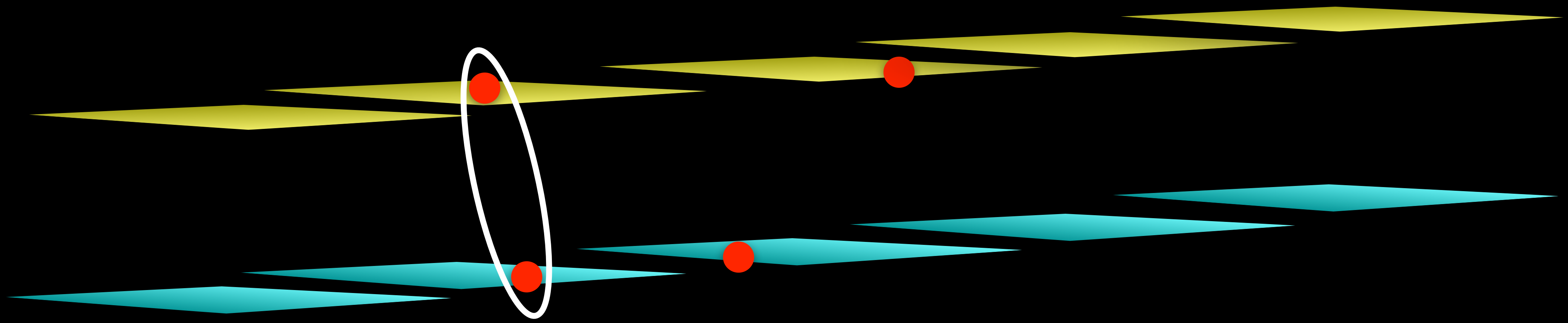
- We do not want to consider this pair of hits



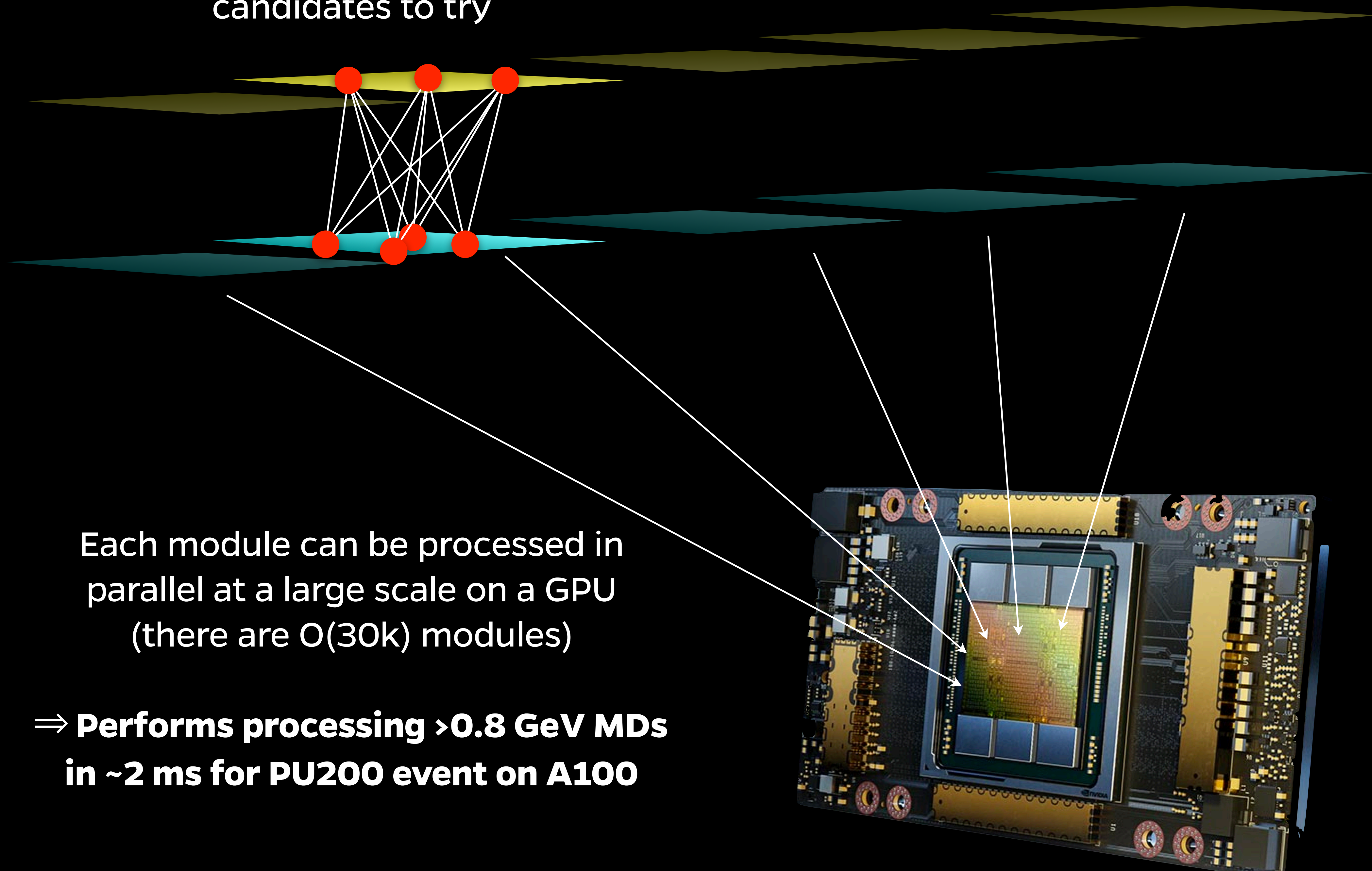
inside trajectory window

- We **want** to consider this pair of hits!

The good candidate pair becomes a **Mini-Doublet (MD)**

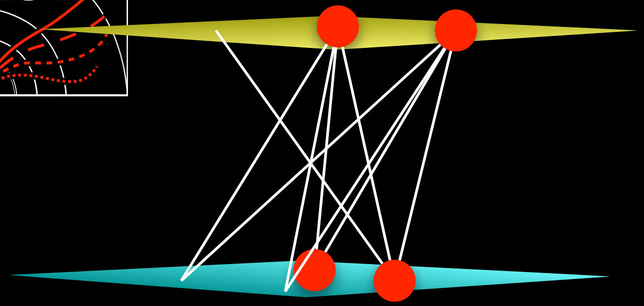


In a given module, there are multiple candidates to try

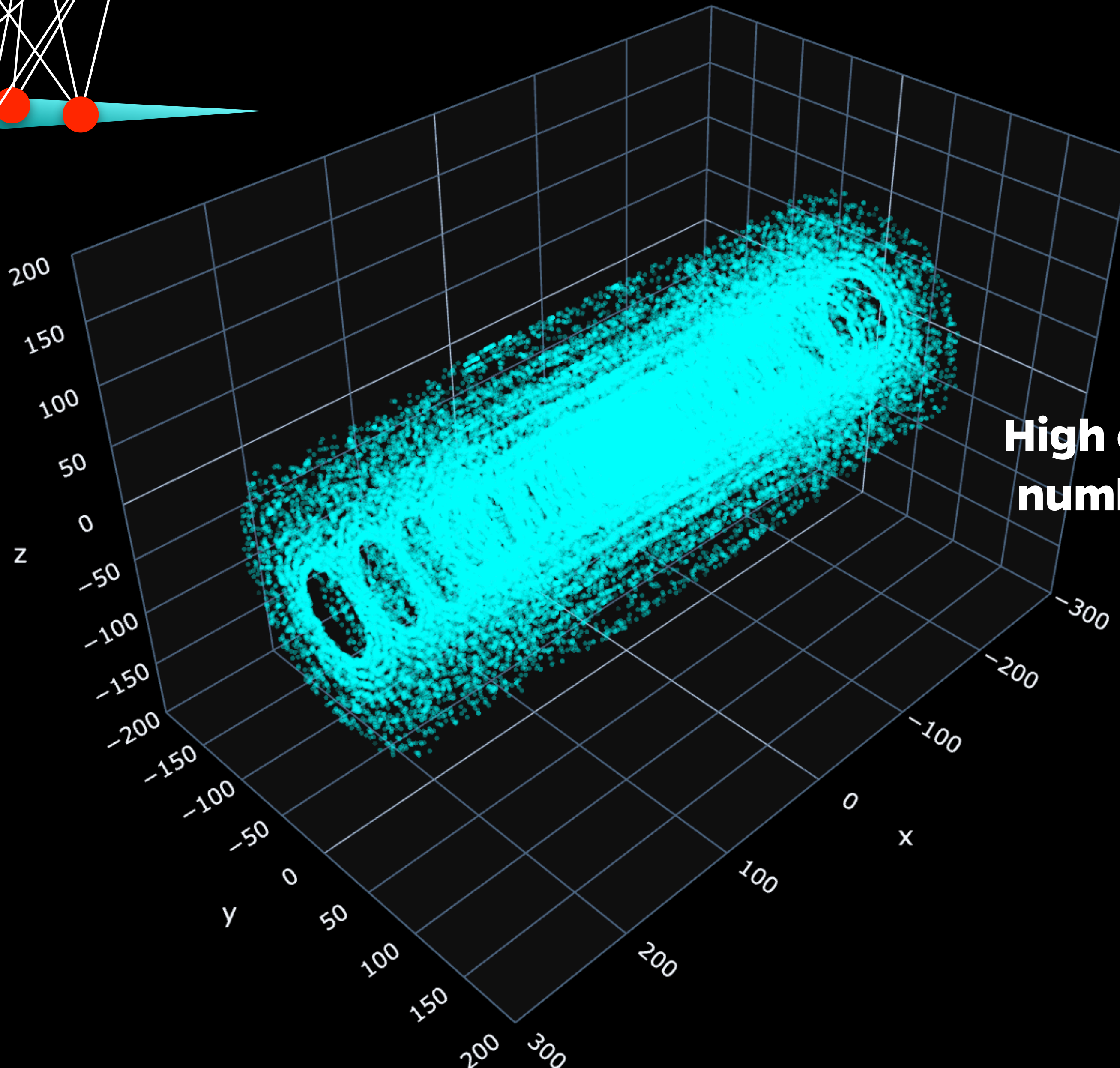


Each module can be processed in parallel at a large scale on a GPU (there are O(30k) modules)

⇒ **Performs processing >0.8 GeV MDs in ~2 ms for PU200 event on A100**

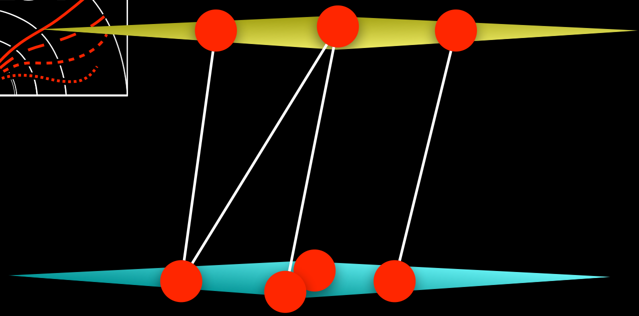


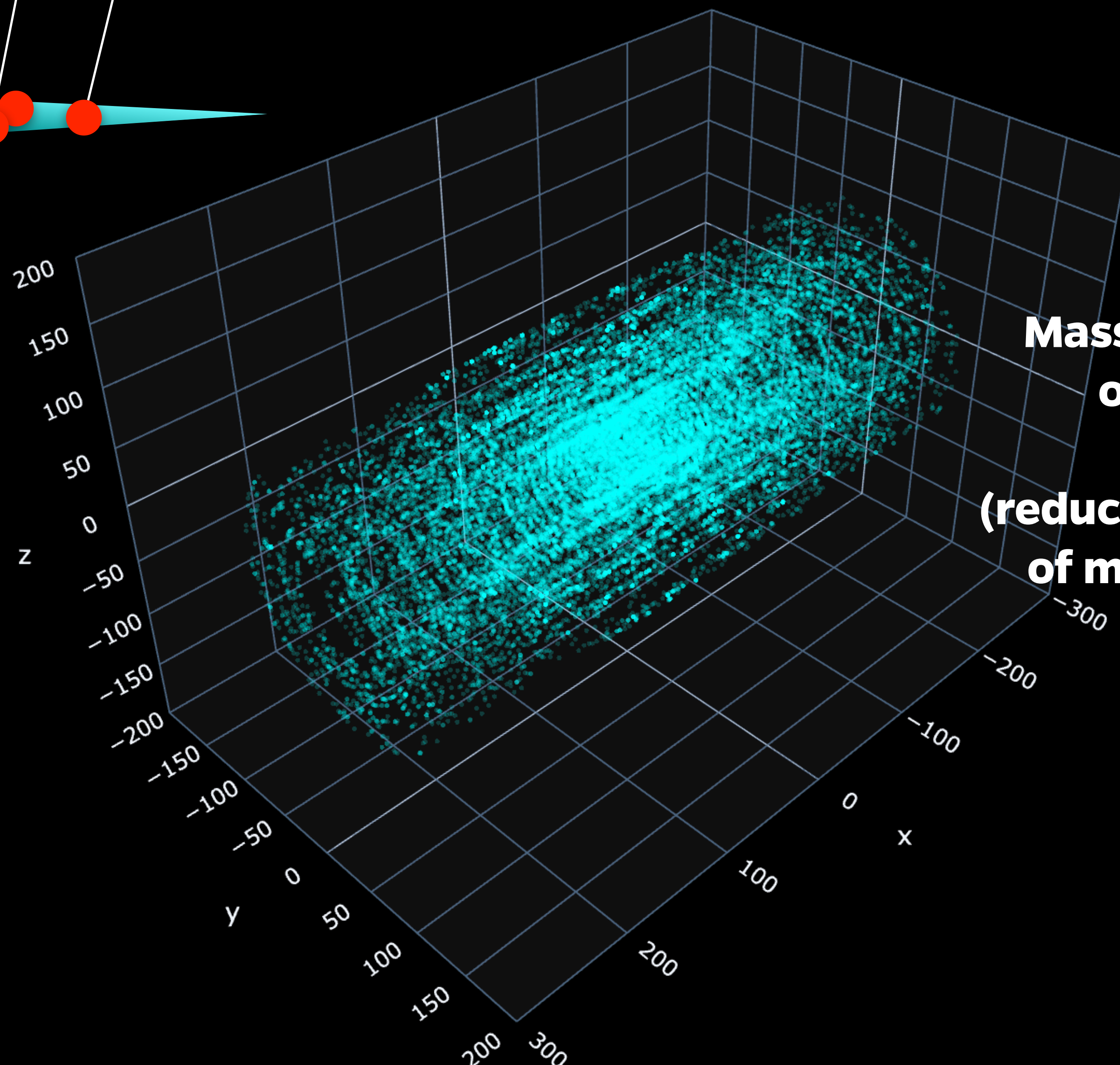
All hits in all modules



High occupancy number of hits

Mini-Doublets only

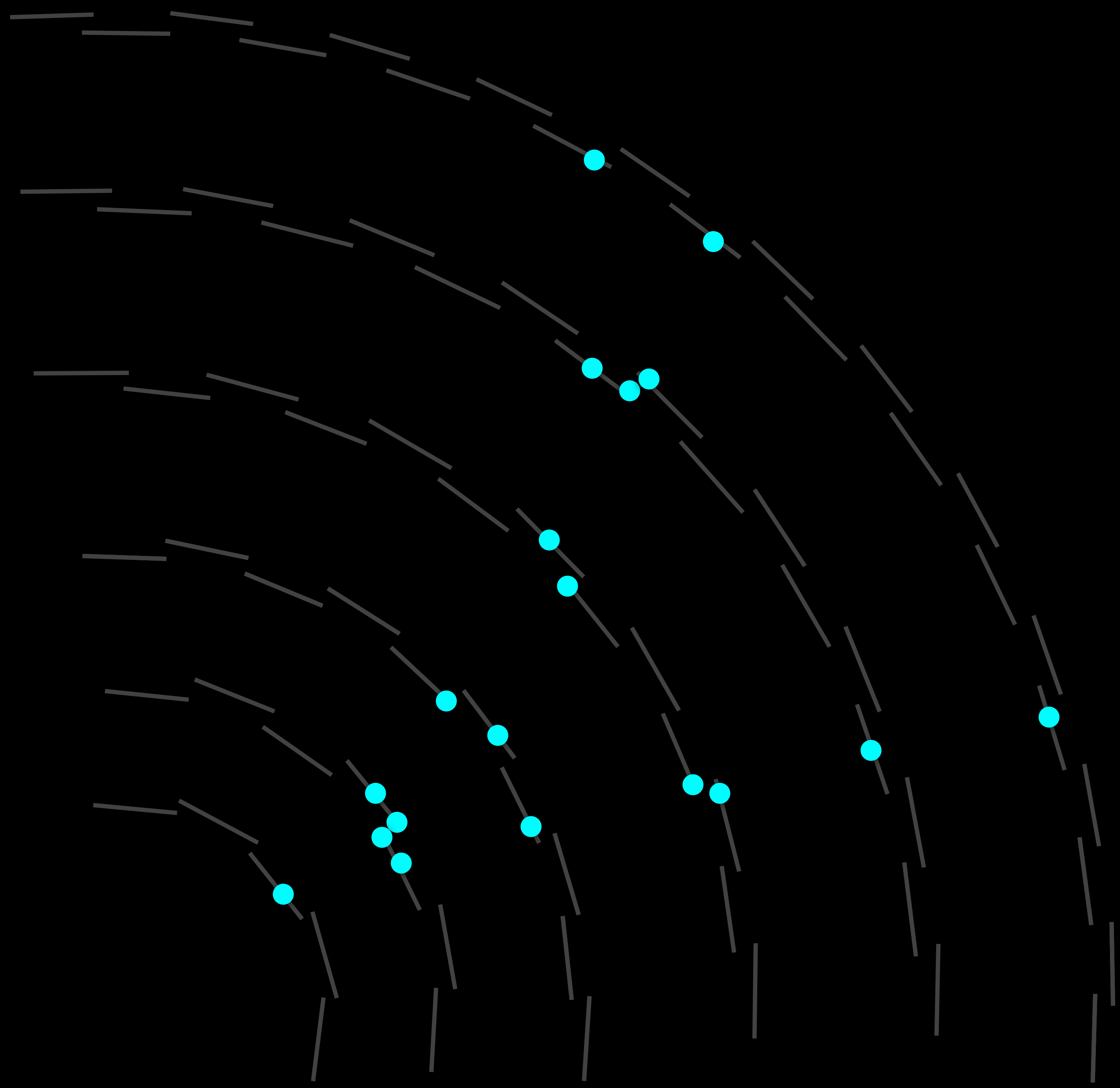




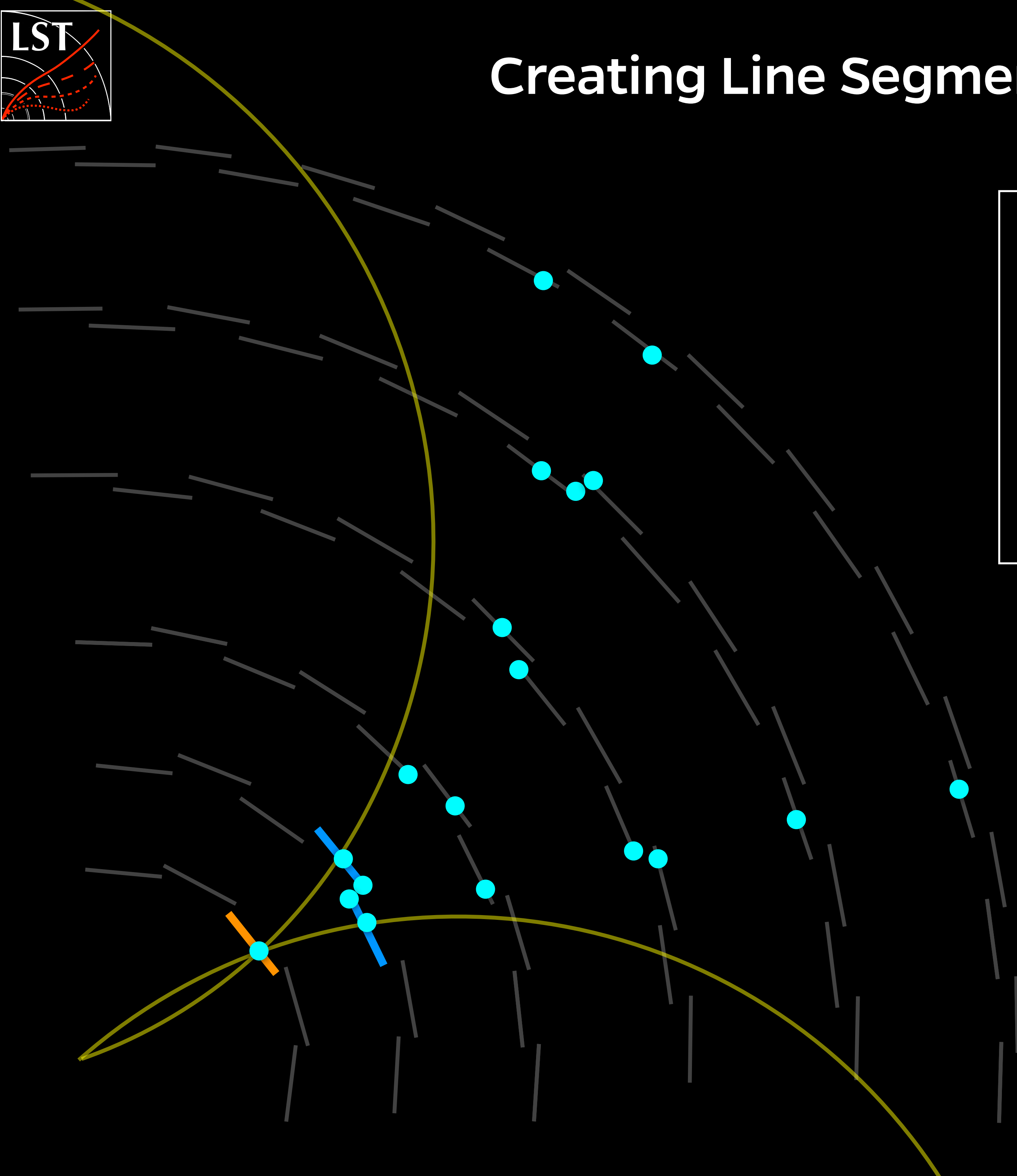
Mass reduction in occupancy

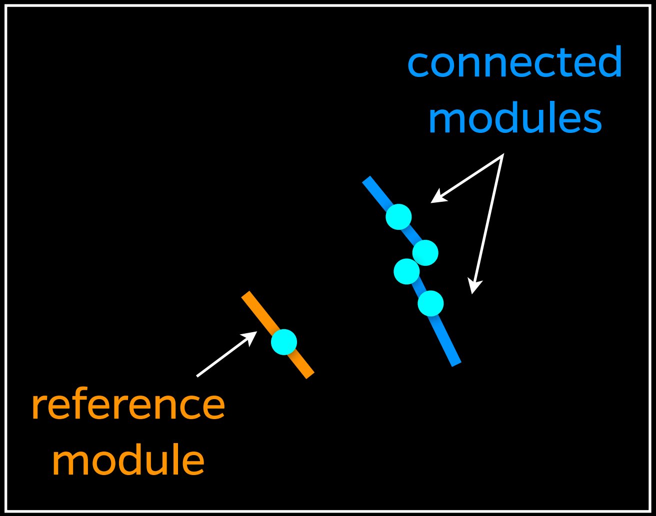
(reduction of ~1 order of magnitude per module)

Creating Line Segments



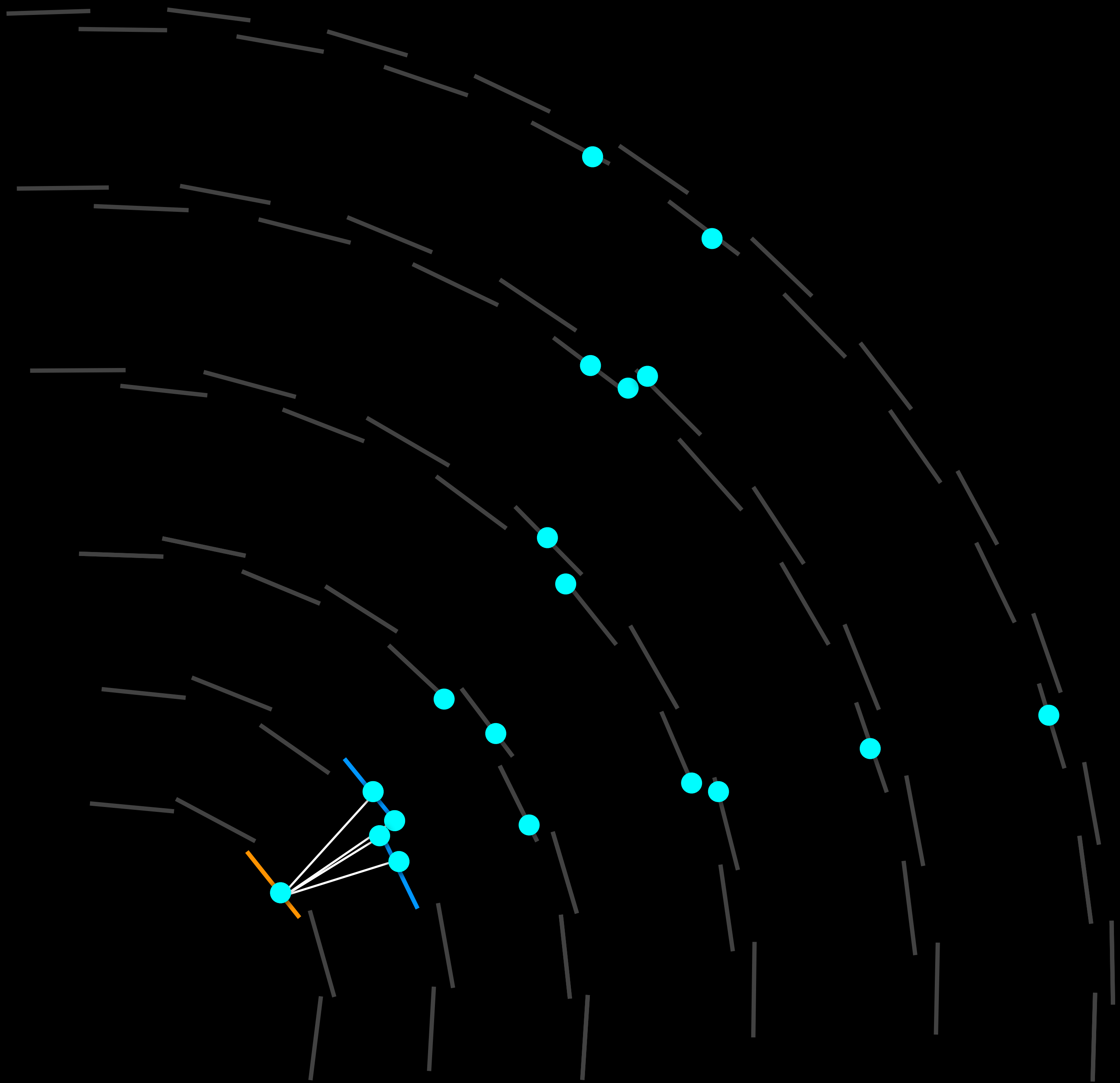
Creating Line Segments



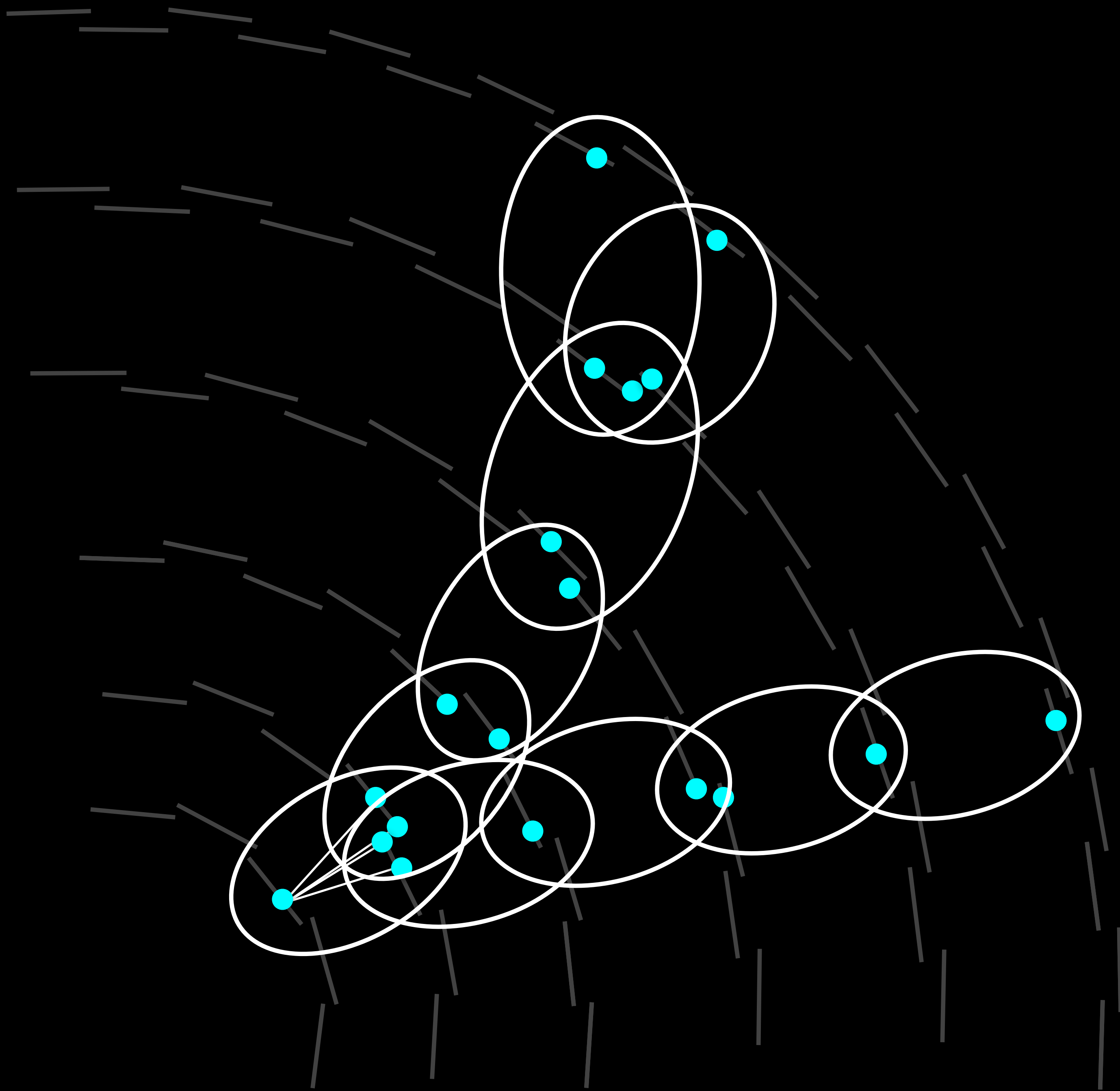


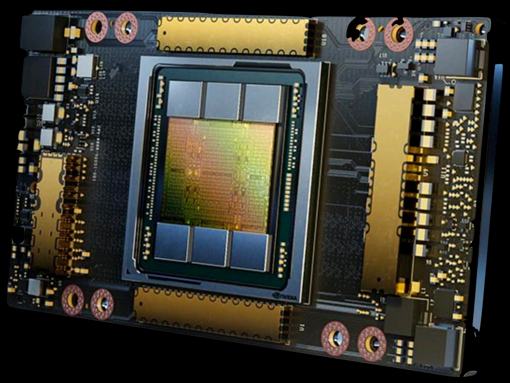
Create Module Map based on helices and simulation

Creating Line Segments



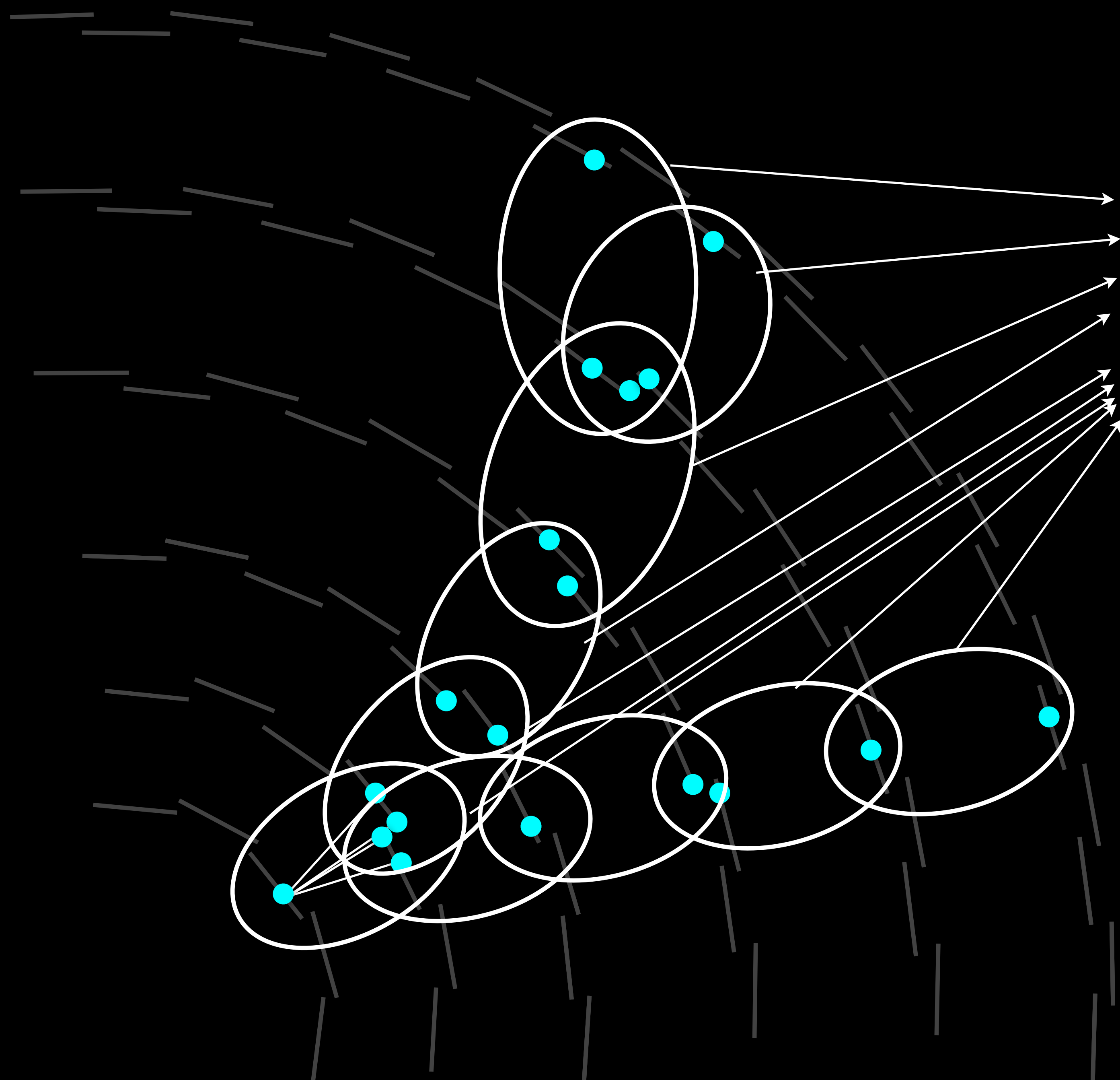
Creating Line Segments

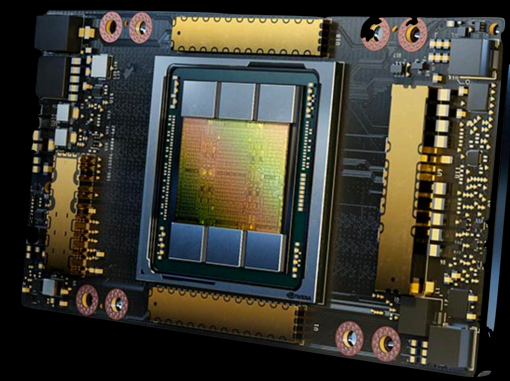




Each pair of modules (depicted by ovals) can process line segment creation in parallel

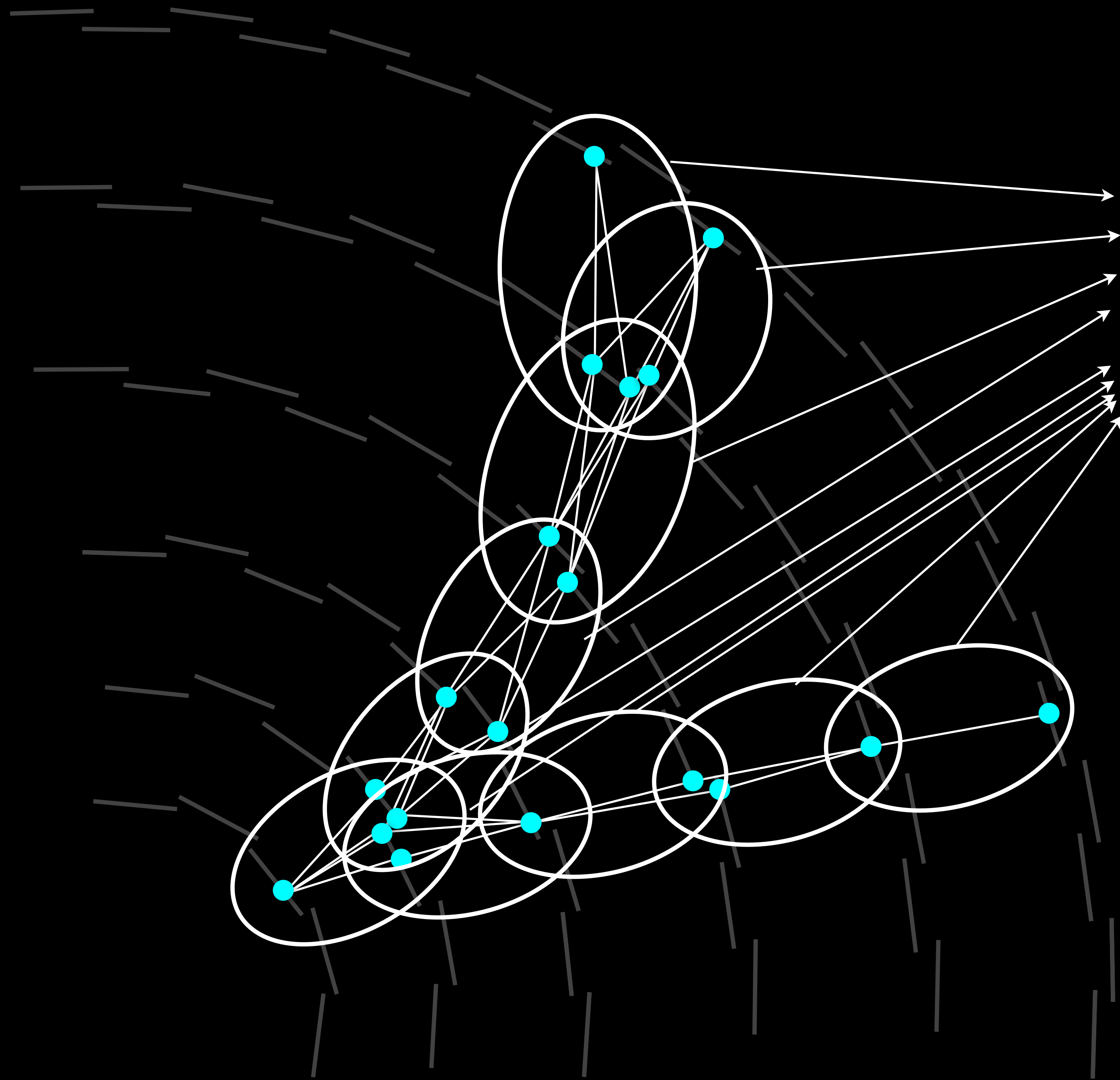
Creating Line Segments

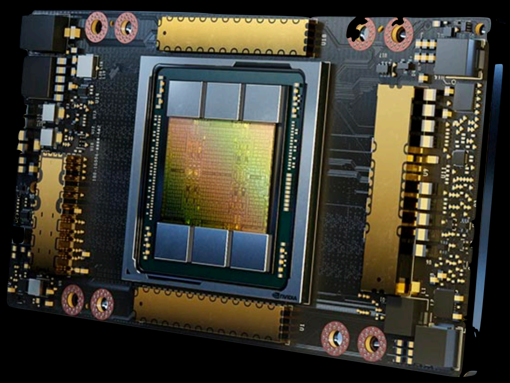




Each pair of modules (depicted by ovals) can process line segment creation in parallel

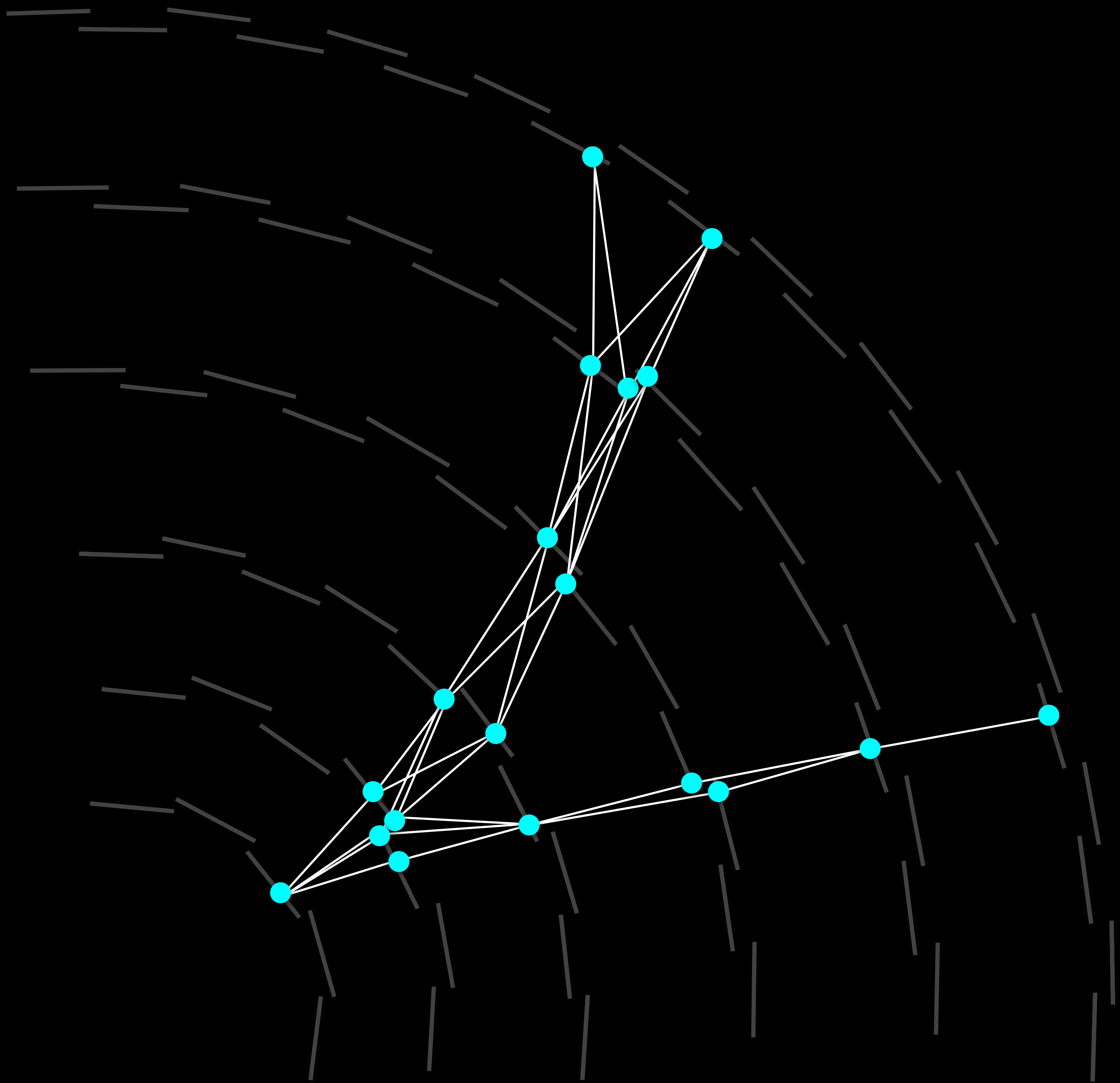
Creating Line Segments

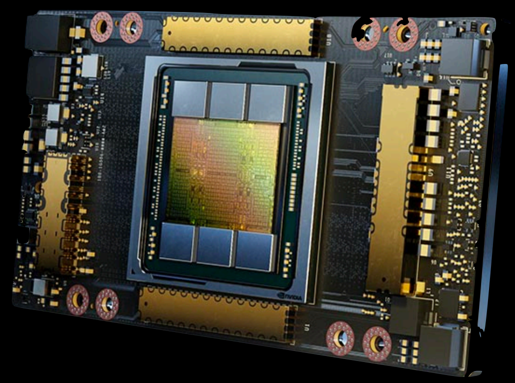




Each pair of modules (depicted by ovals) can process line segment creation in parallel

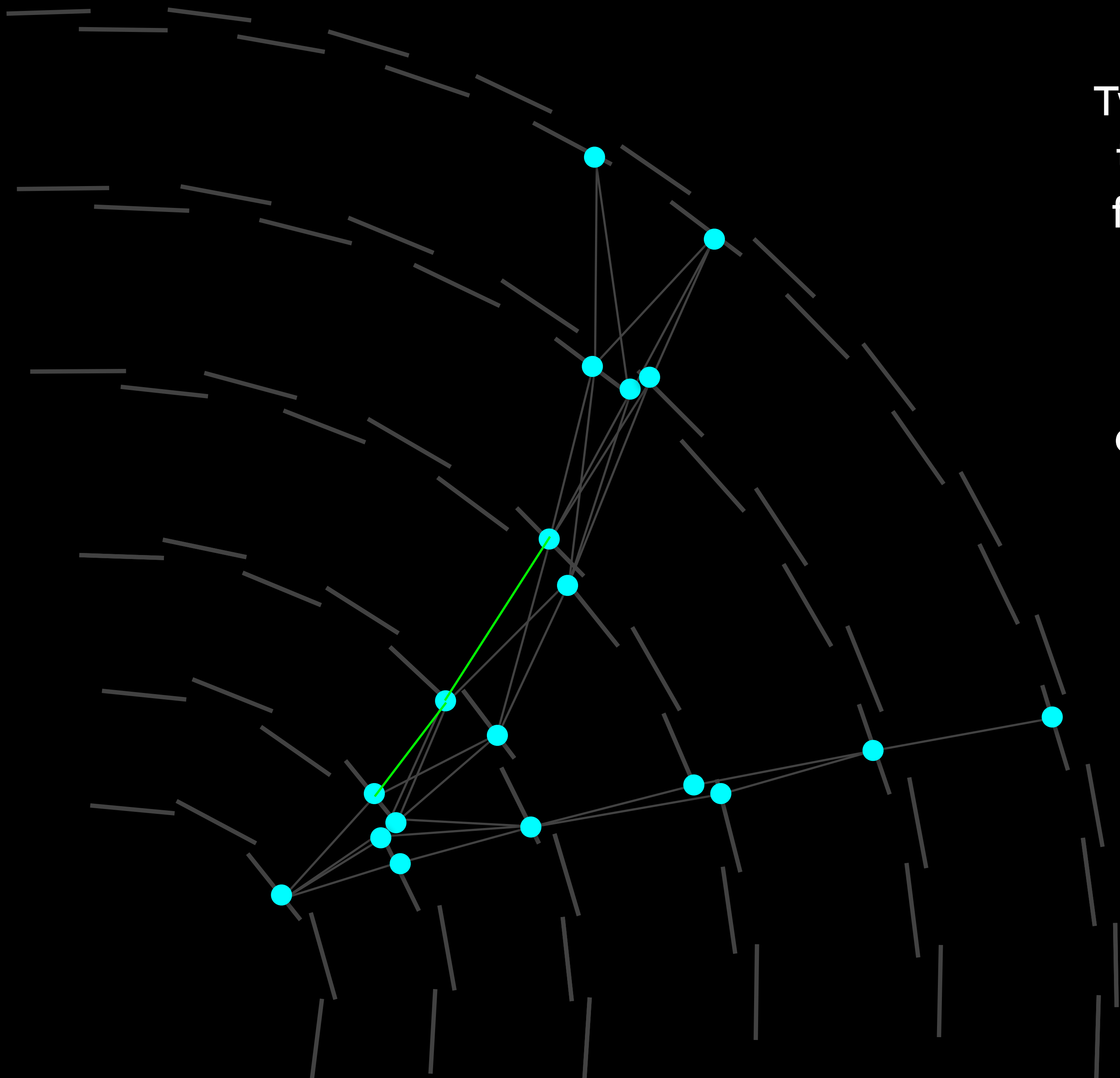
Creating Line Segments





Each pair of modules (depicted by ovals) can process line segment creation in parallel

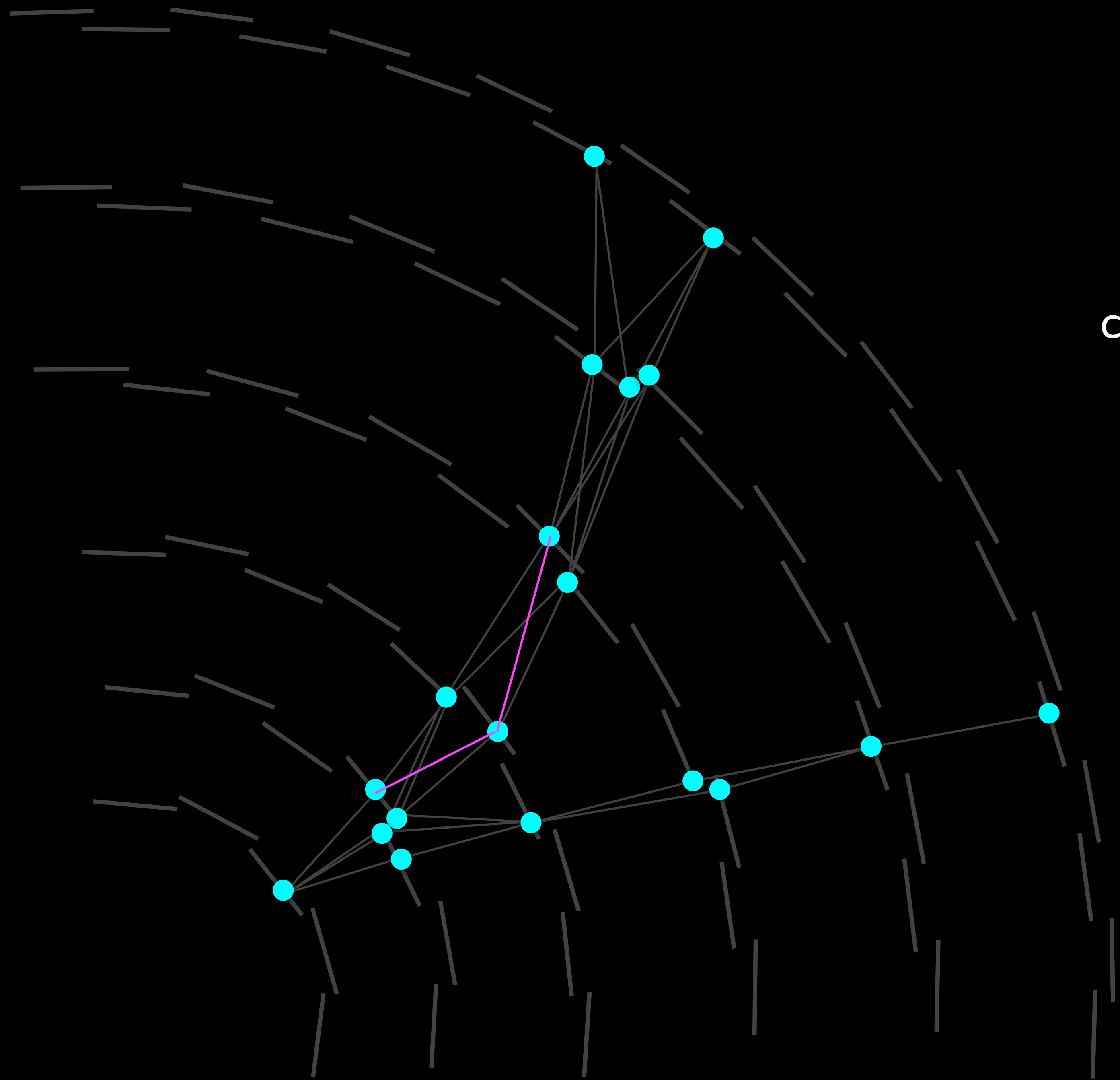
Connecting Line Segments



Two Line Segments can then be connected to form a Triplet of mini-doublets (T3)

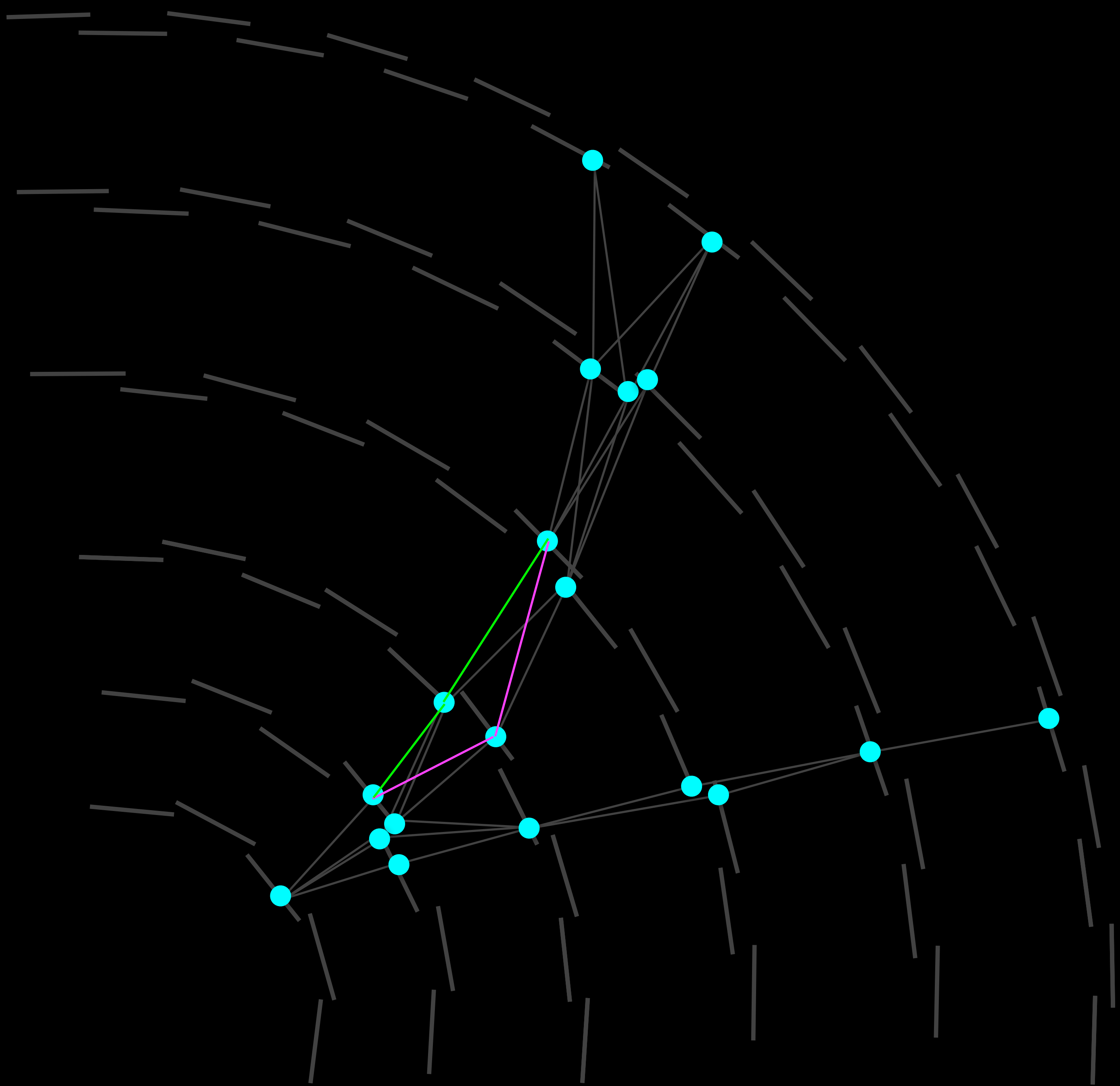
We apply geometric constraints to require them to be helix-like

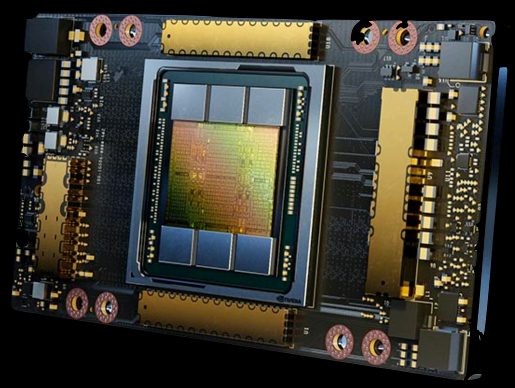
Connecting Line Segments



If the triplets do not look like a good candidate they are not created

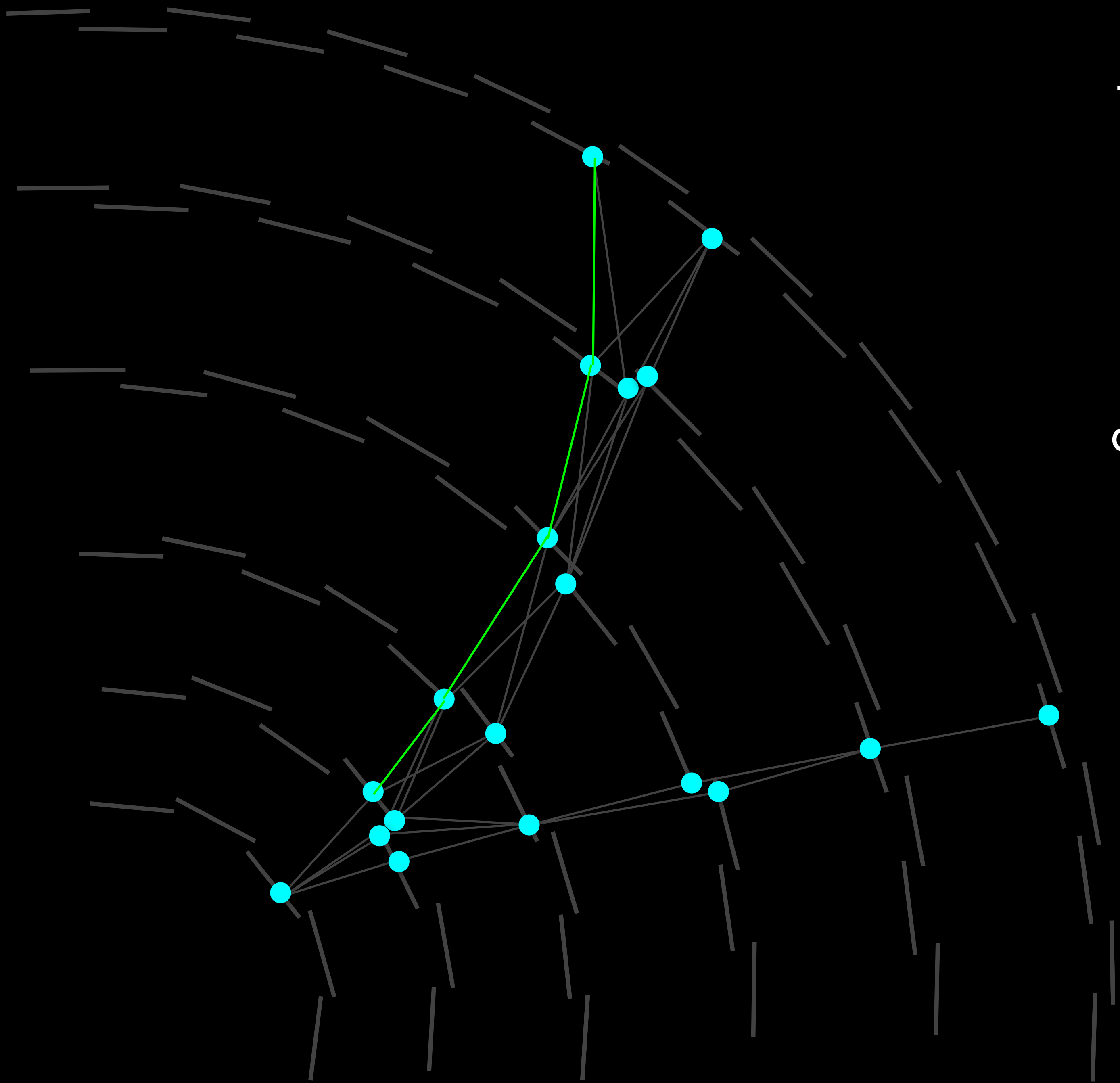
Connecting Line Segments





Triplet formation can be performed in parallel on GPUs

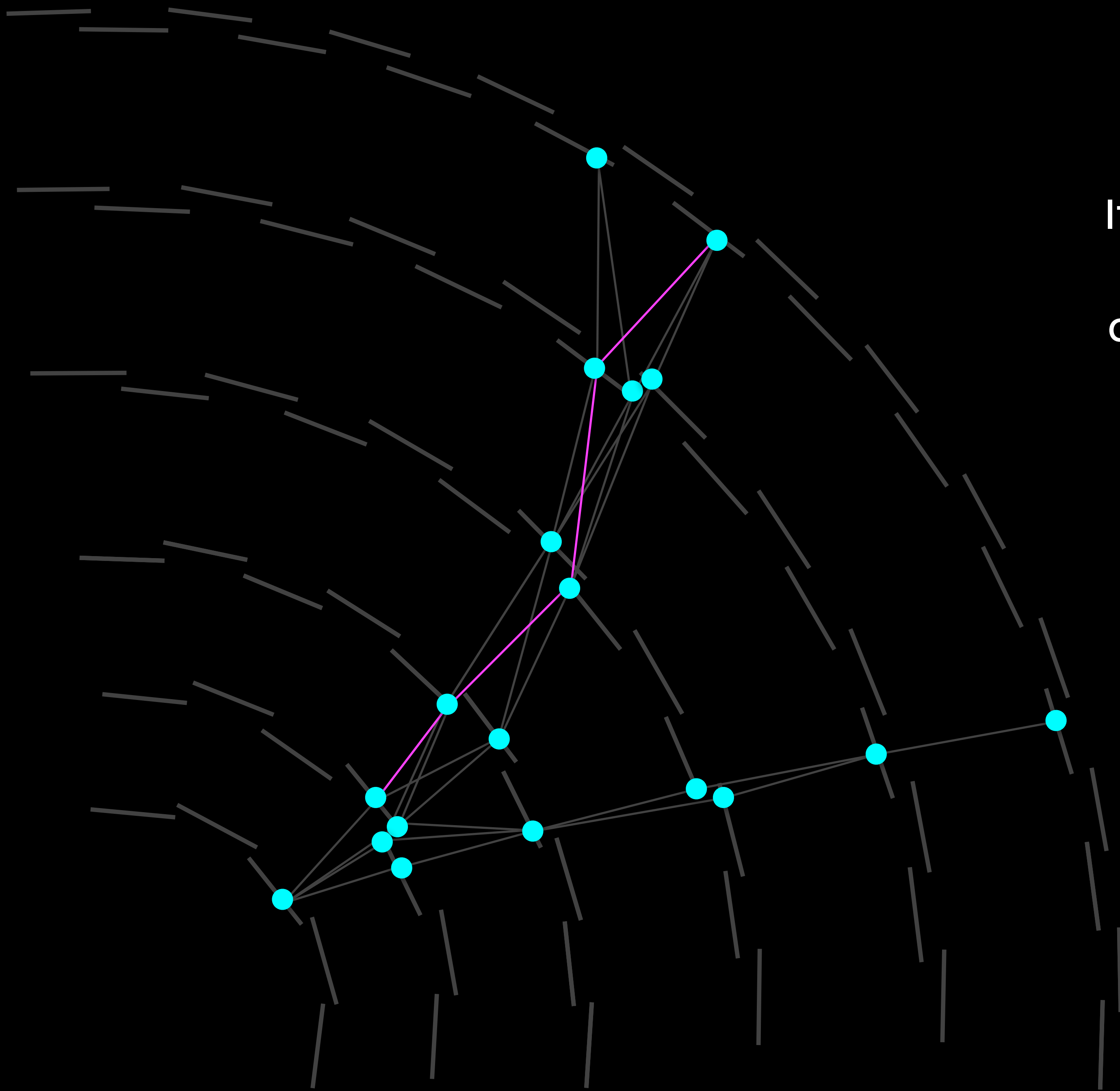
Creating Quintuplets



Two T3's can then be connected to form a Quintuplet of mini-doublets (T5)

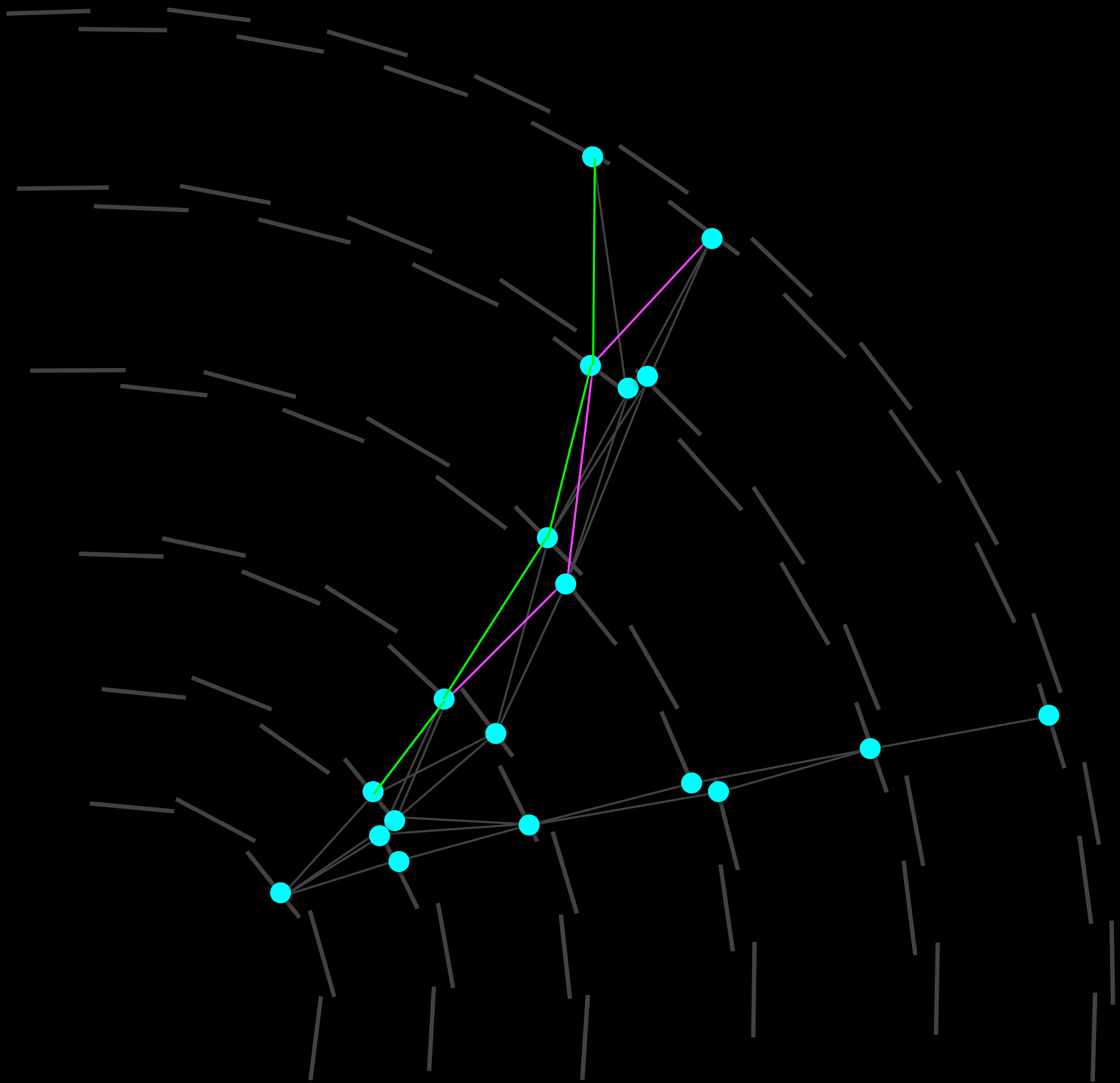
We apply geometric constraints to require them to be helix-like

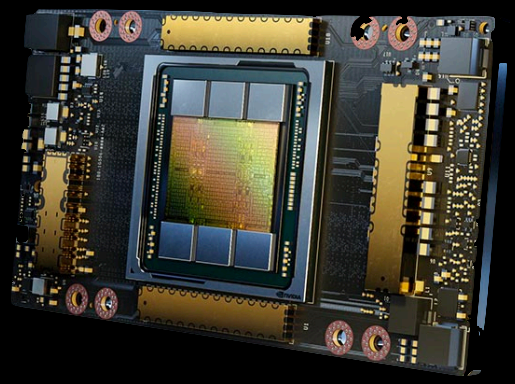
Creating Quintuplets



If the quituplets do not look like a good track candidate they are not created

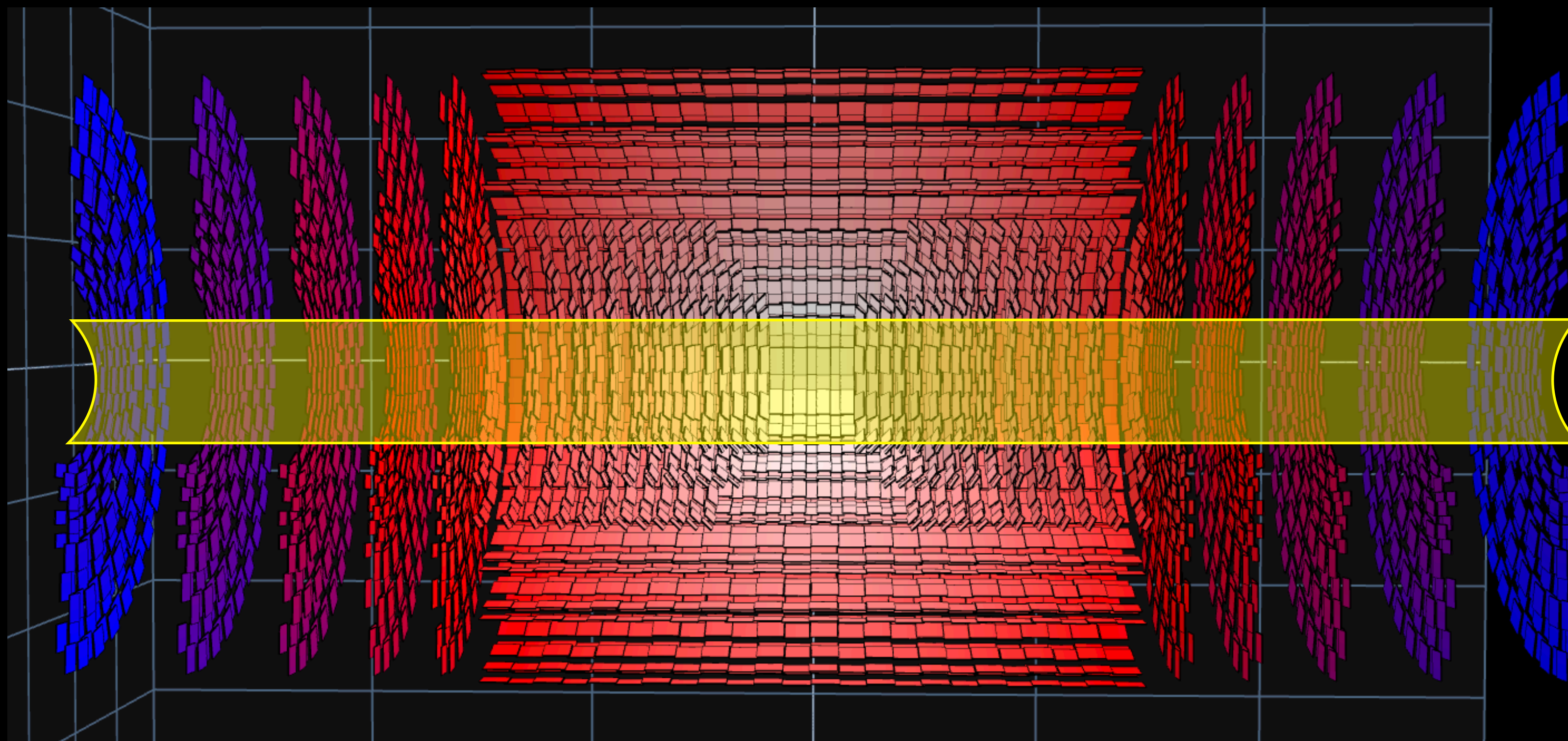
Creating Quintuplets





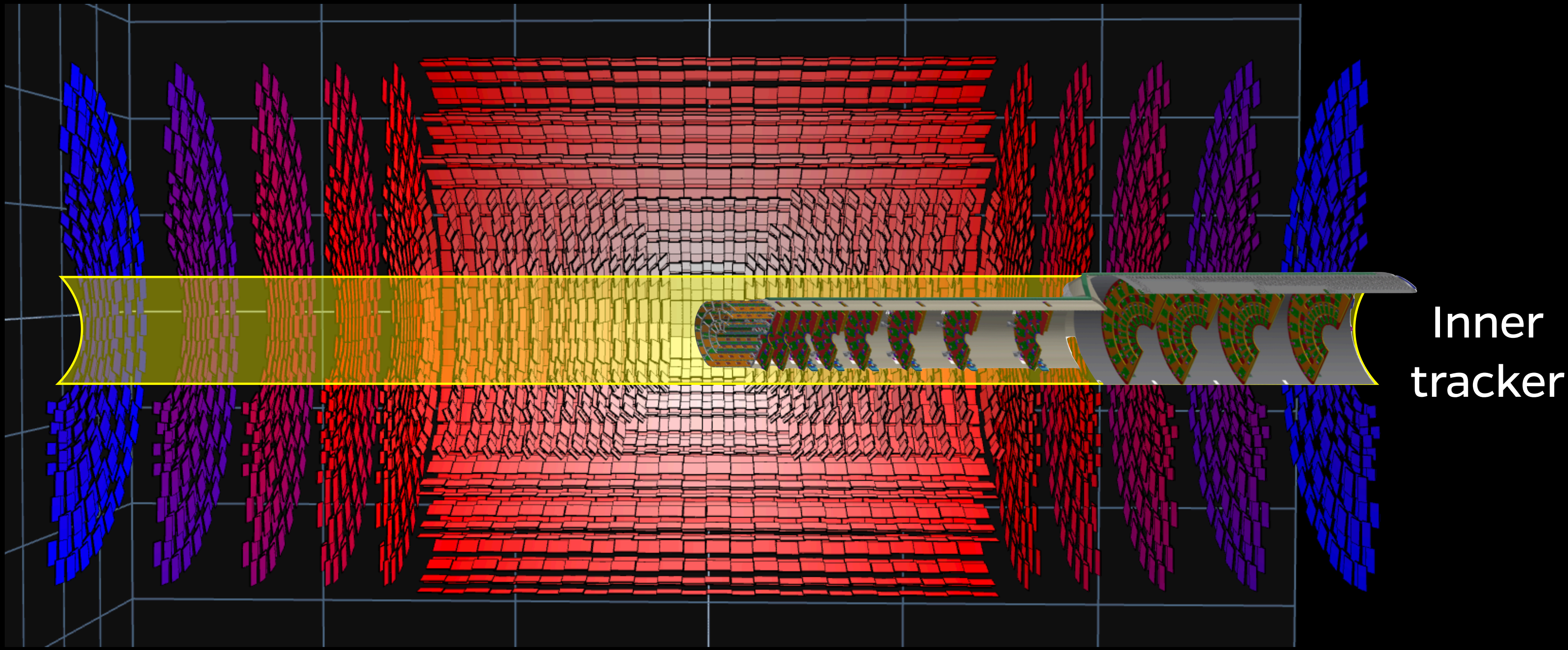
Quintuplet formation can be performed in parallel on GPUs

Inner tracker seeds as Line Segments

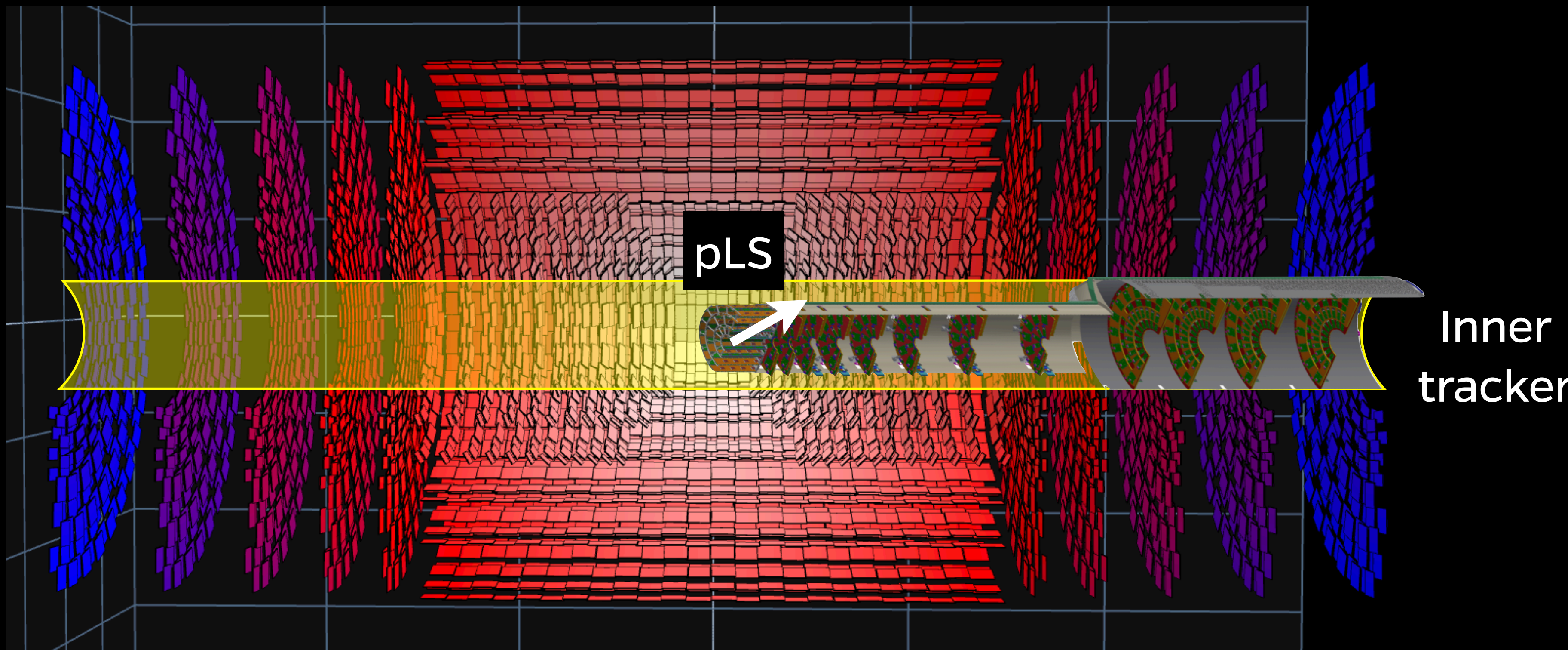


Inner
tracker

Inner tracker seeds as Line Segments

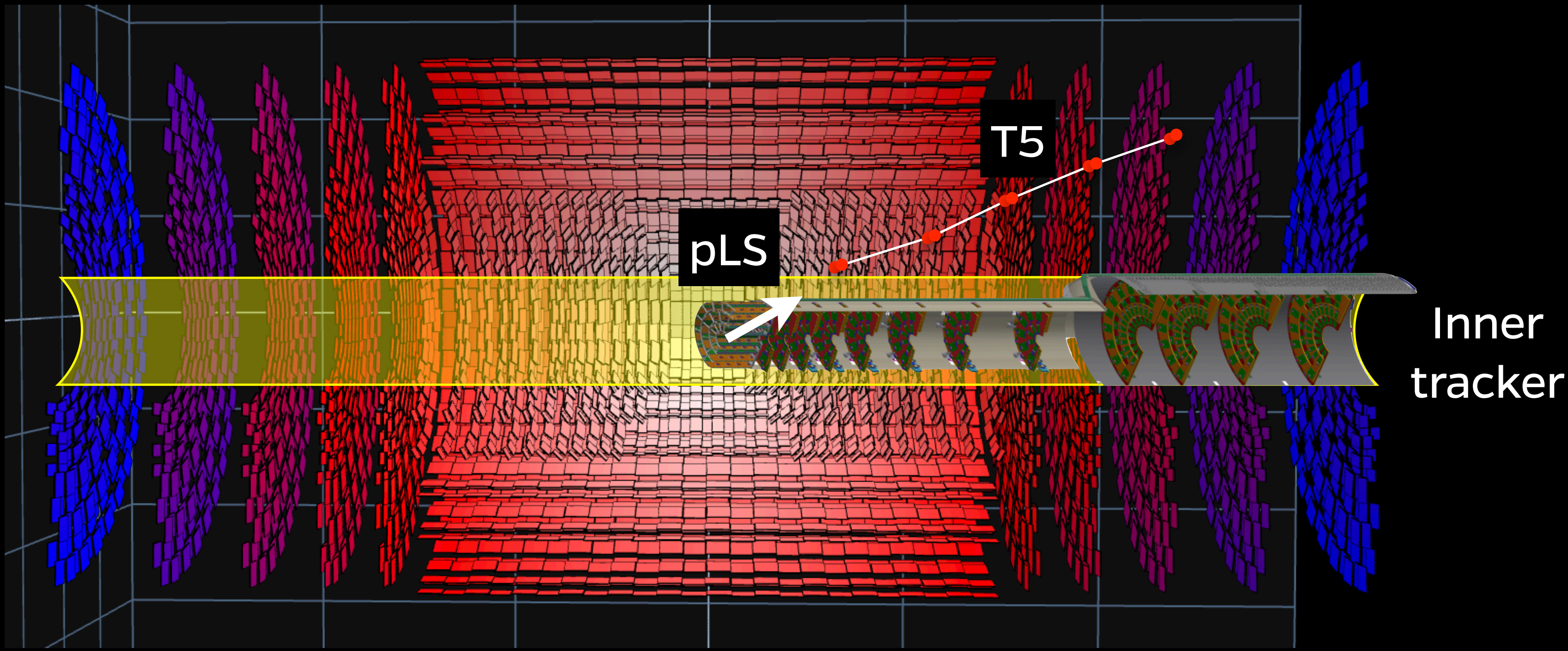


Inner tracker seeds as Line Segments



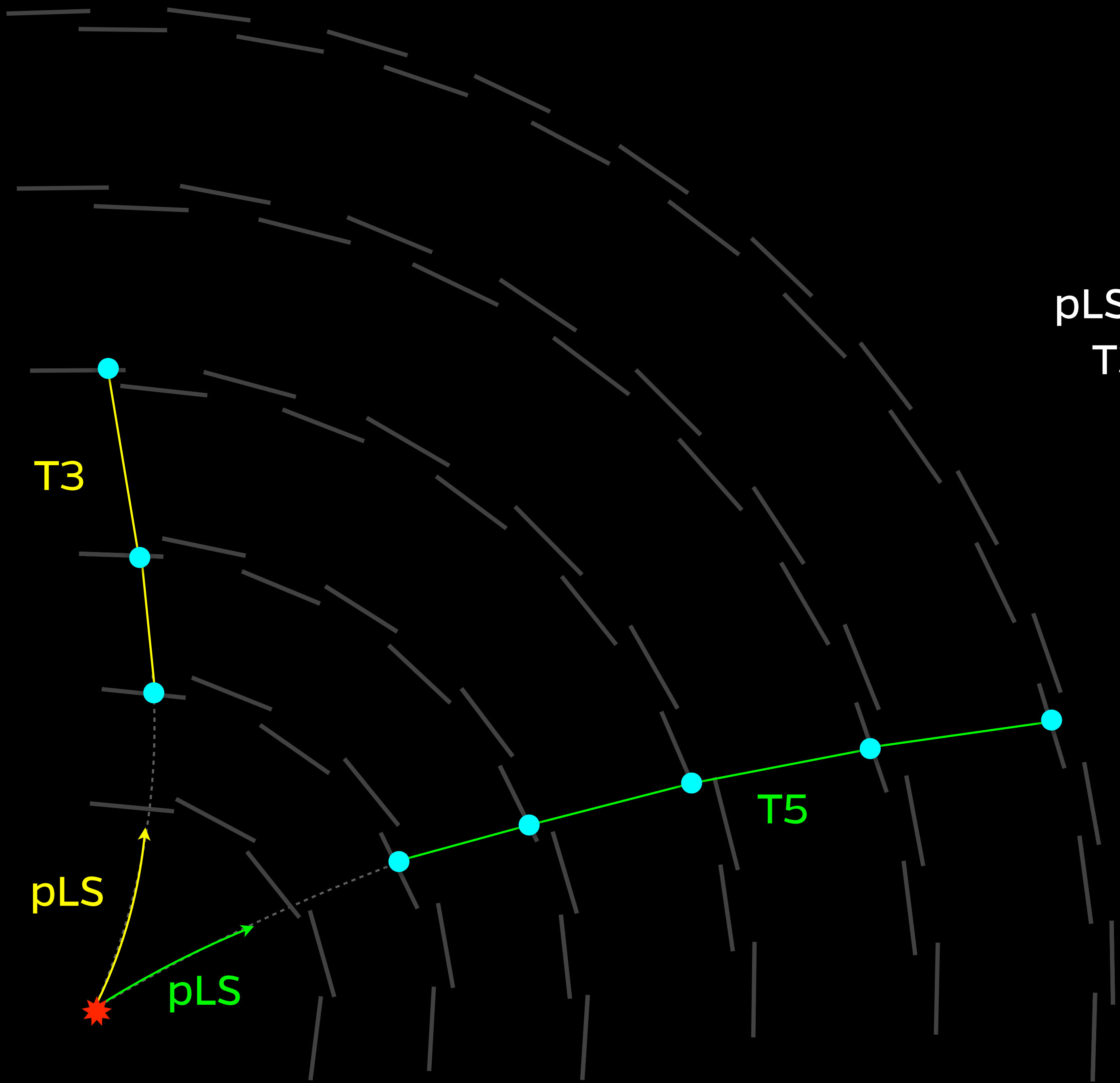
Inner tracker provides “mini”-tracks with three to four hits
⇒ use these “seeds” as equivalent as a Line-Segment called “pixel LS” (pLS)

Inner tracker seeds as Line Segments



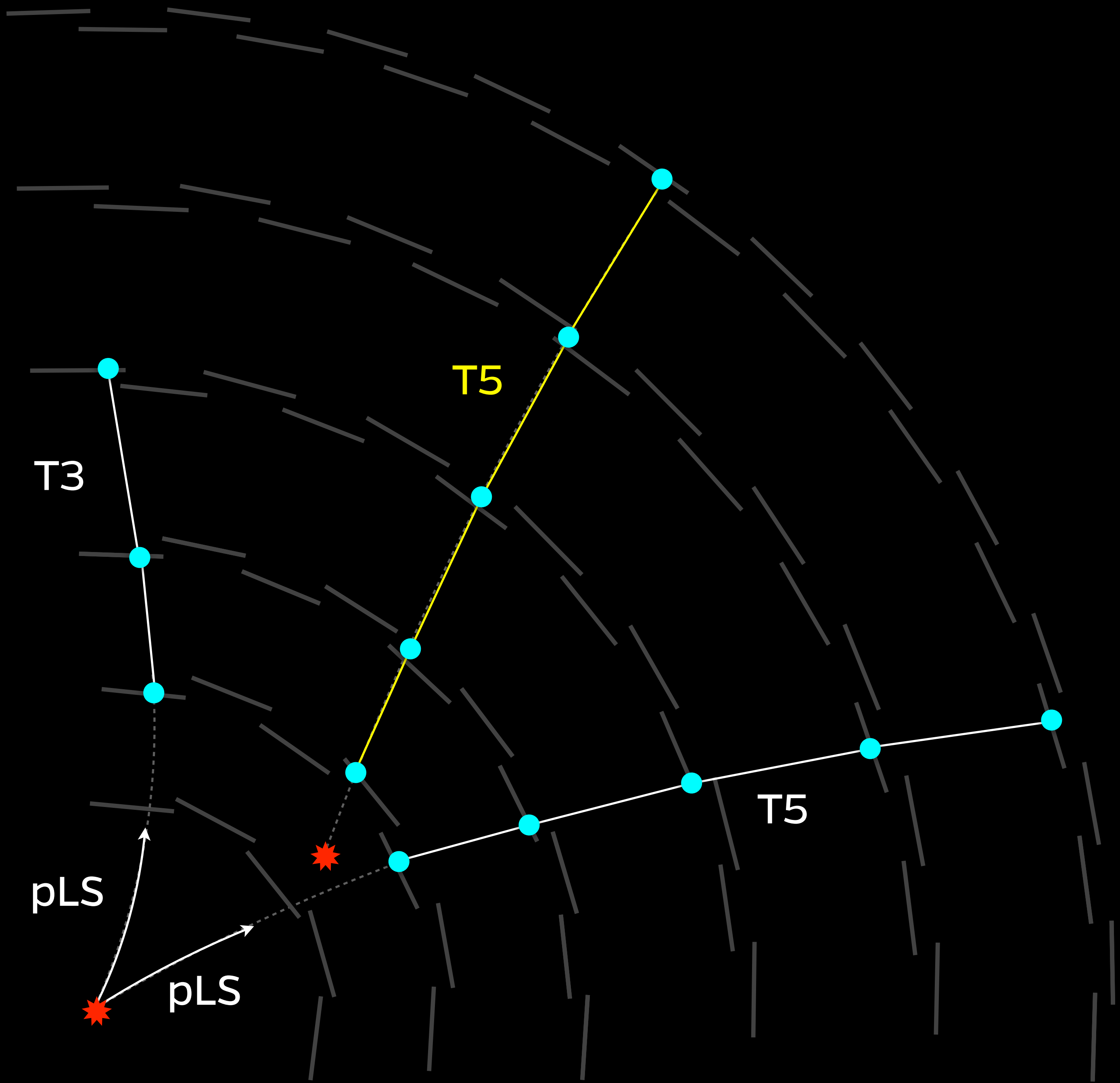
Inner tracker provides “mini”-tracks with three to four hits
⇒ use these “seeds” as equivalent as a Line-Segment called “pixel LS” (pLS)

Types of track candidates built



pLSs are then connected with T3's and T5's to form pT3, and pT5

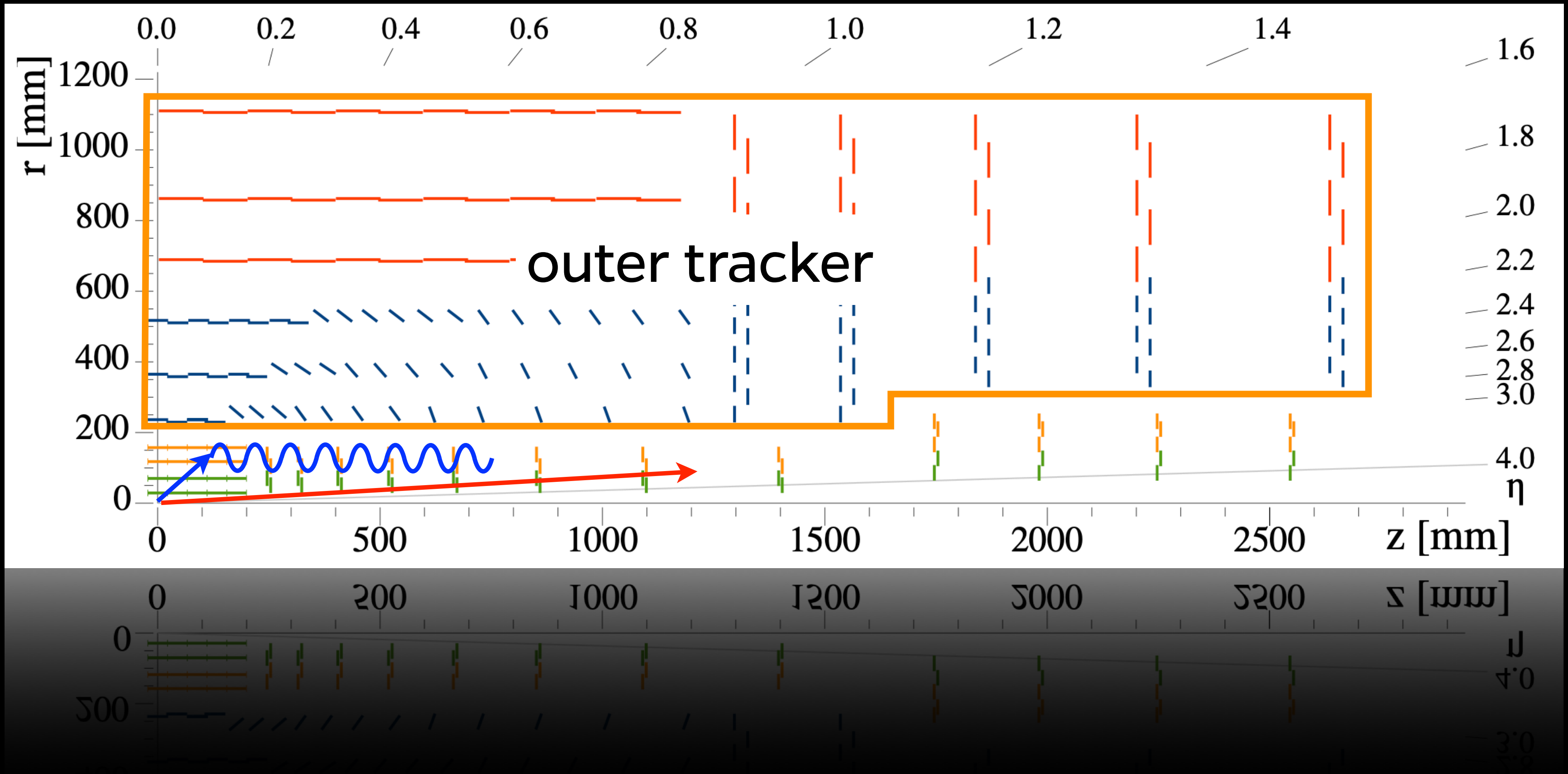
Types of track candidates built



T5's can be standalones even if no matching pLSs are found

(cf. T5's have 10 hits, and can naturally track displaced tracks)

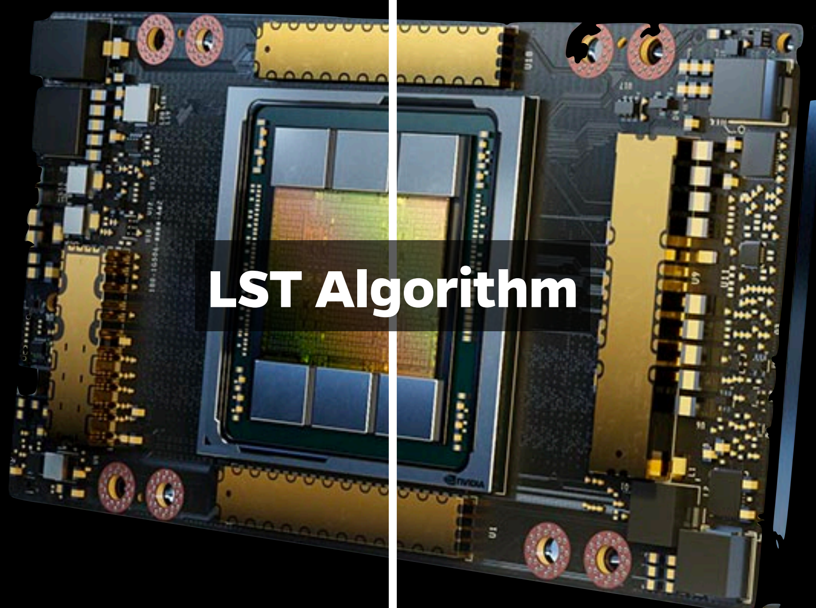
Adding unused pLS



Not every pLSs are consumed by outer tracker and unused ones are useful for short or forward tracks

UF
Chang
Florida

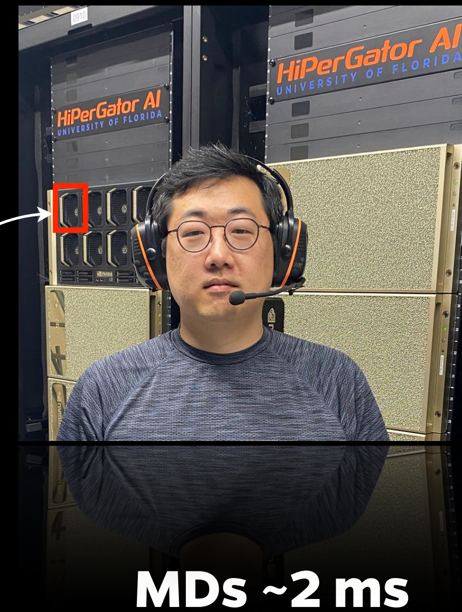
Outer Tracker Hits, Inner Tracker Seeds



LST Algorithm

Track Candidate Collection

{pT5s, ..., pT3s, ..., T5s, ..., pLSs, ..., }



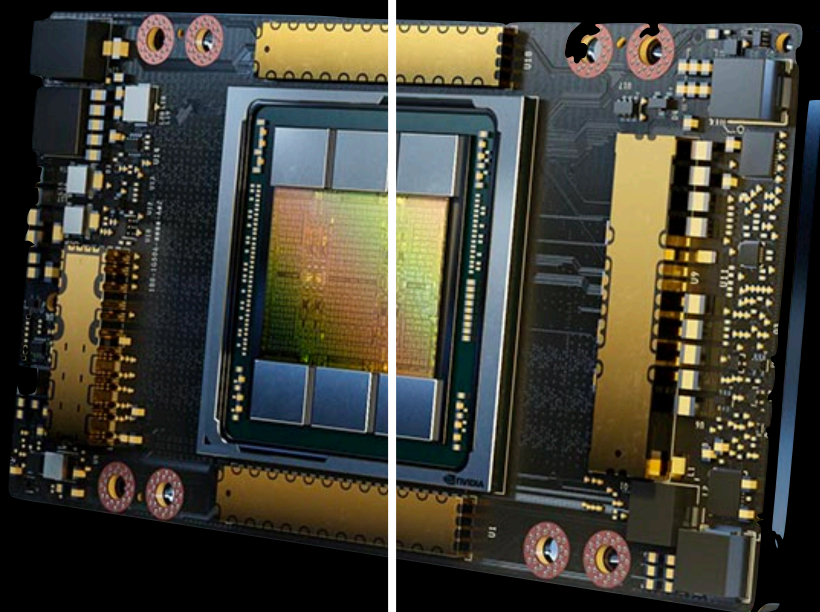
MDs ~2 ms
LSs ~1 ms
T3s ~3-4 ms
T5s ~3-4 ms
pT3s ~2 ms
pT5s ~2 ms
TCs ~3 ms

Total ~ 17 ms / evt

Measured on **single A100** at HiPerGator AI of University of Florida

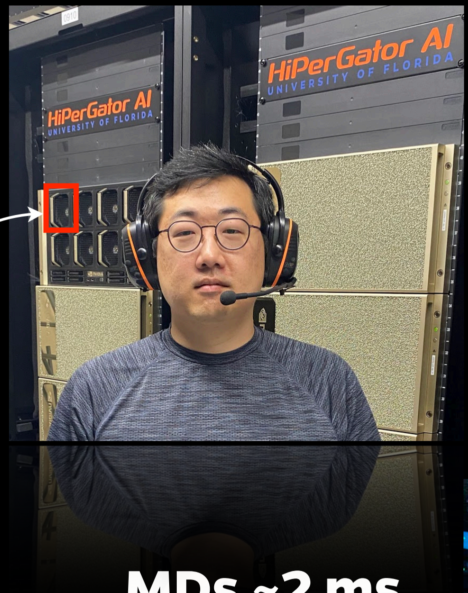
UF
Chang
Florida

Outer Tracker Hits, Inner Tracker Seeds

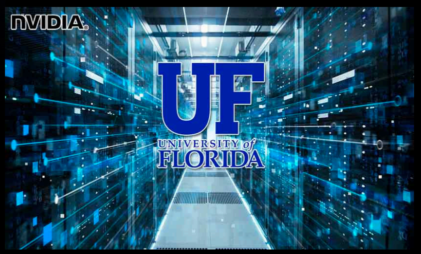


Track Candidate Collection

{pT5s, ..., pT3s, ..., T5s, ..., pLSs, ..., }



MDs ~2 ms
LSs ~1 ms
T3s ~3-4 ms
T5s ~3-4 ms
pT3s ~2 ms
pT5s ~2 ms
TCs ~3 ms



Measured on **single A100** at HiPerGator AI of University of Florida

Total ~ 17 ms / evt

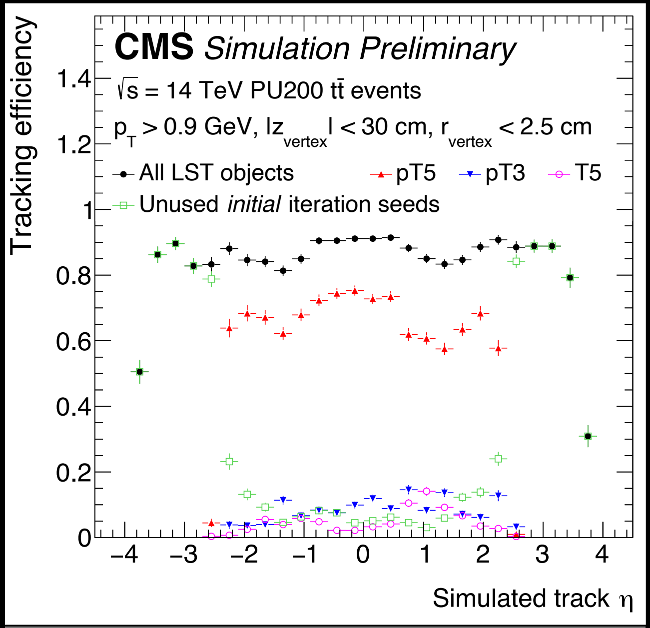
Total ~ 9.6 ms / evt

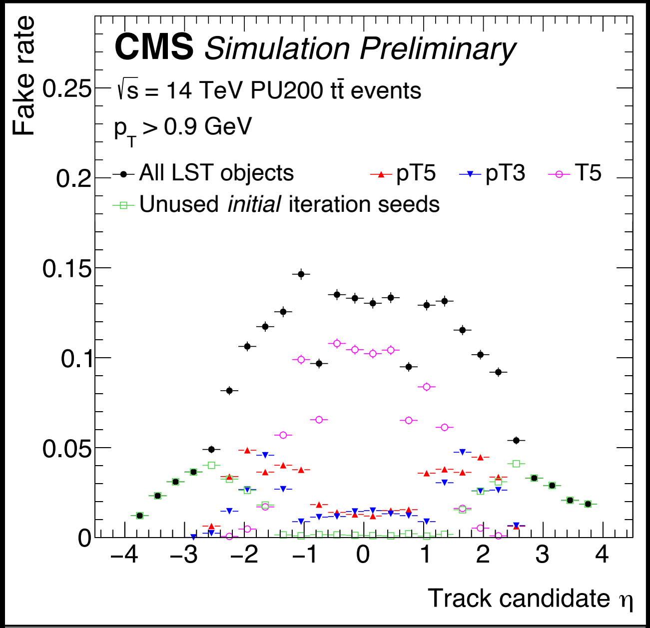
If multi-streaming to have multiple events in flight

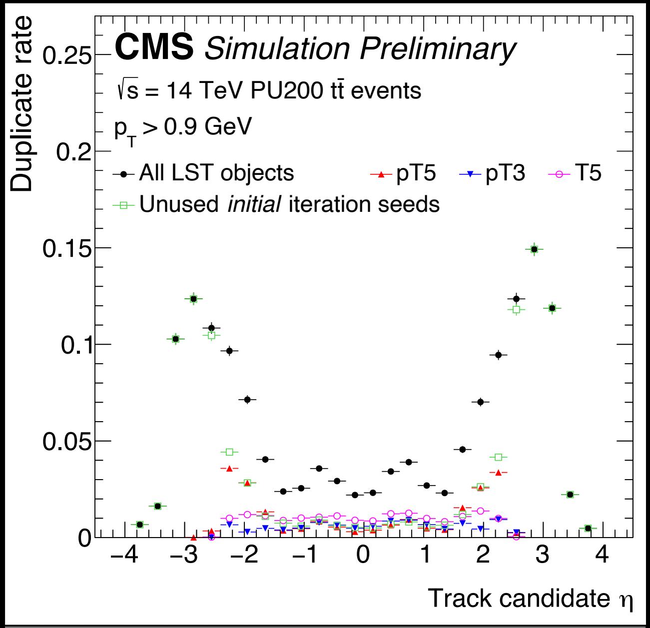
Performance on $t\bar{t}$ Pileup 200 events

First we use standalone setup with rough estimate of track parameters to assess performance.
(Full result with integration to CMS Software featured in later slides.)

Efficiency of Track Candidates	Fake Rate of Track Candidates	Duplicate Rate of Track Candidates



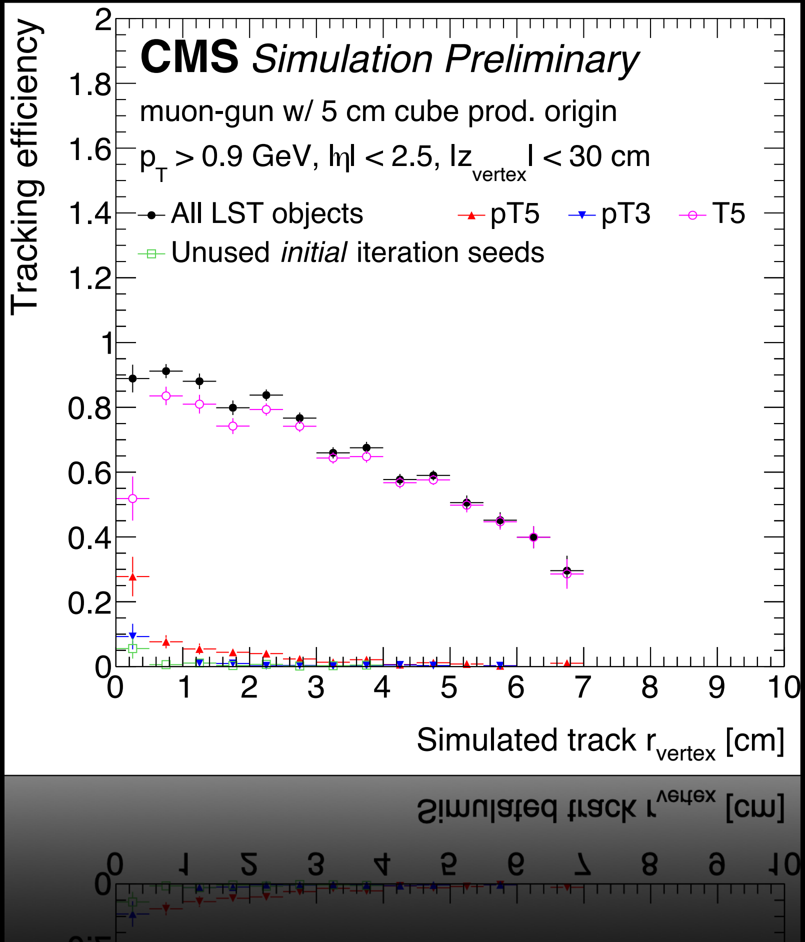




Performance for displaced tracks

First we use standalone setup with rough estimate of track parameters to assess performance.
(Full result with integration to CMS Software featured in later slides.)

Work in progress



Out-of-the-box (with no configuration change) LST covers good amount of displaced tracks
Can be improved further

displaced muon gun sample

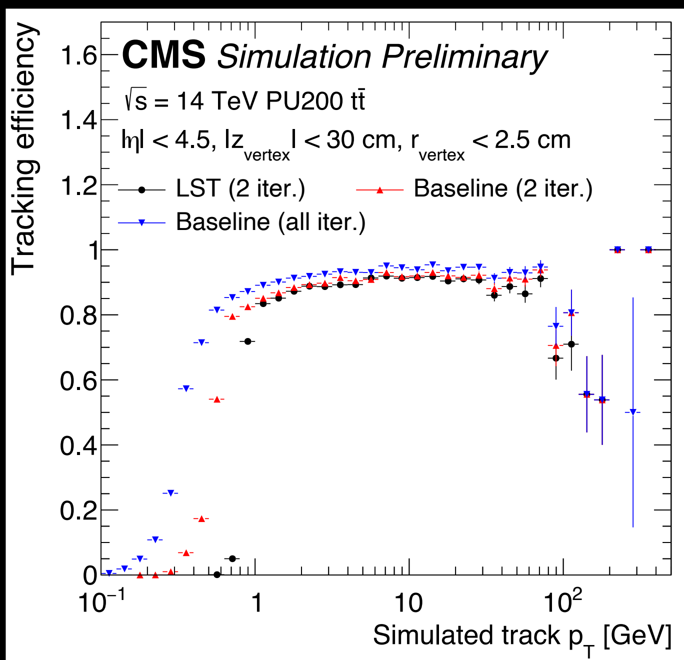
Preliminary LST result from CMSSW

Chang
Florida

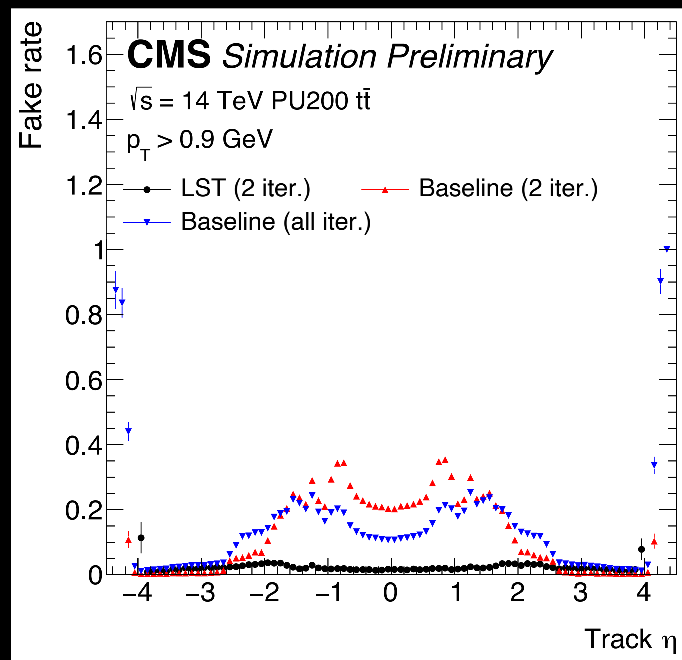
Work-in-progress

Next, we use integrate the LST algorithm to CMS Software (CMSSW) workflow. Track fitting is performed and “loose” track selection applied.

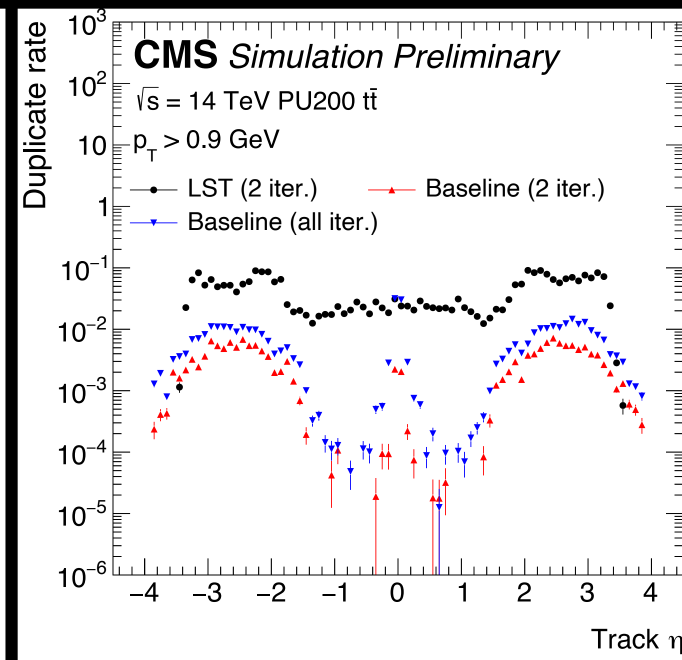
Efficiency of Track Candidates



Fake Rate of Track Candidates



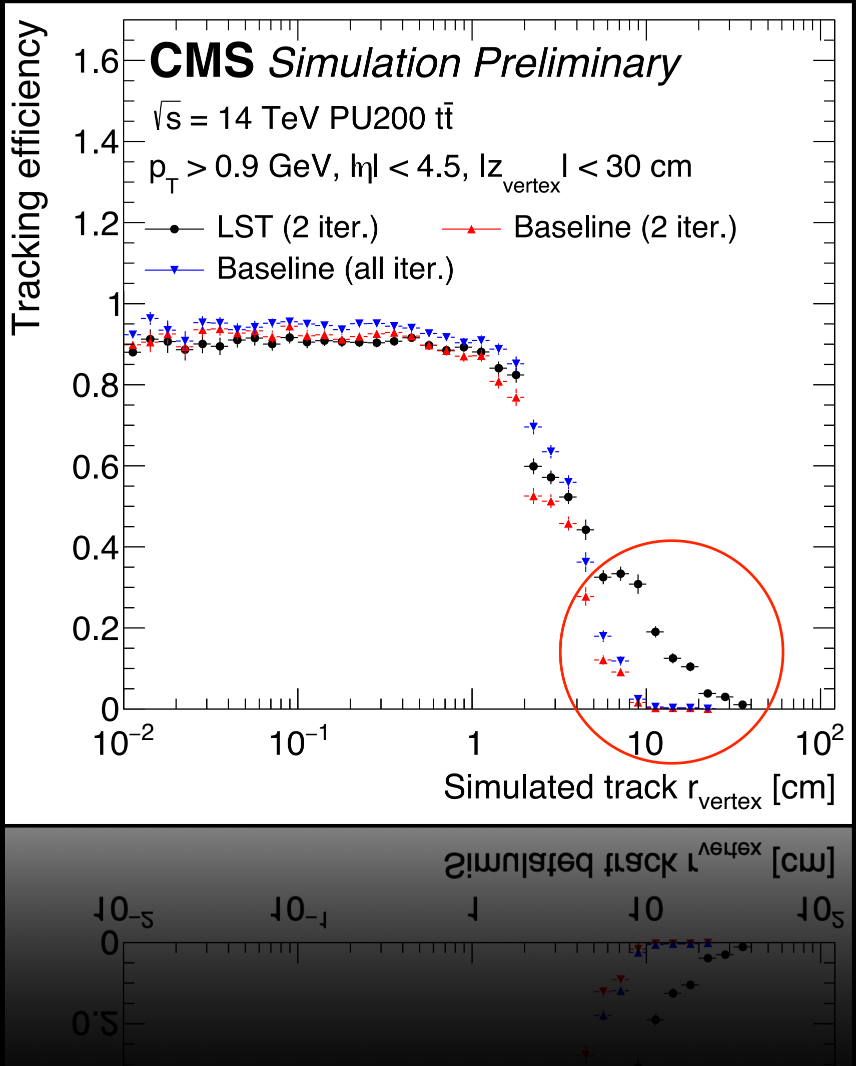
Duplicate Rate of Track Candidates



Performance for displaced tracks

Work-in-progress

Next, we use integrate the LST algorithm to CMS Software (CMSSW) workflow. Track fitting is performed and “loose” track selection applied.



Out-of-the-box (with no configuration change) LST covers good amount of displaced tracks
Can be improved further

Summary

Presented LST algorithm for HL-LHC using GPUs

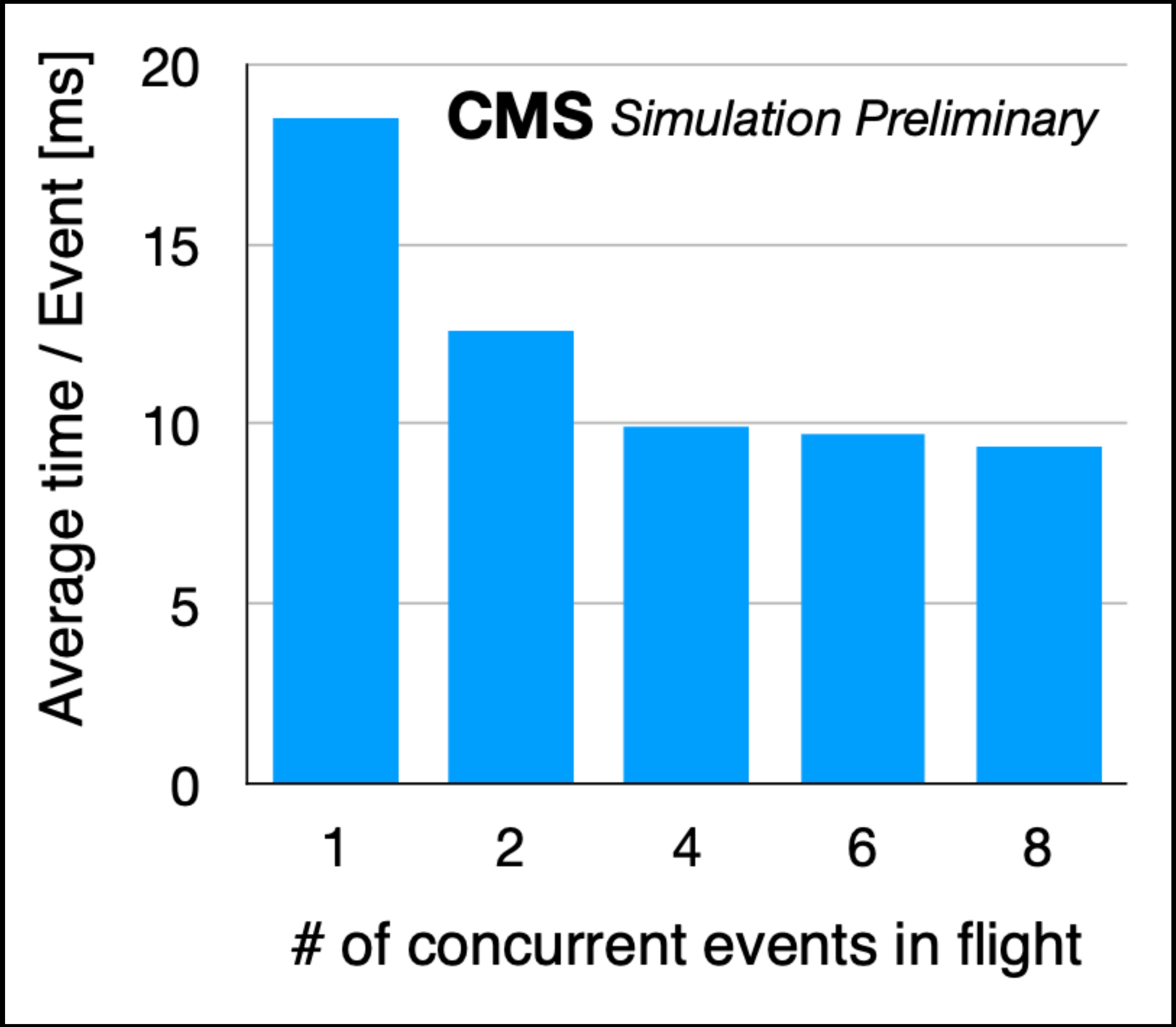
Shows good performance on par w/ default tracking

Naturally extends to displaced tracks

Integration / tuning / optimization on-going

Backup

Timing vs. N-streaming



Timing breakdown pie-chart