ACTS Tracking For Muon Collider

Karol Krizka, Simone Pagan Griso

on behalf of the Muon Collider Detector and Physics Group

April 20, 2021

APS April Meeting

Current Tracking Implementation

- **Designed for the e^+e^- environment**
  - Inherited as part of the ILC software framework
- **Implements conformal tracking (1908.00256)**
  - Transform circular tracks into straight lines using conformal map
  - Use *cellular automata* to look for lines, allowing for deviations
- **Problem: $\mu^+\mu^-$ collider is much busier due to BIB**
  - Heavy pre-filtering of hits is necessary for track finding to work

Show events display of ee vs mumu

Add link to Simone's (detector description) and Massimo's (tracking) talks.

Why ACTS for MCC?

ACTS GitHub

ACTS is a generic library for track fitting at collider experiments.

- **Dedicated team working on advancing tracking algorithms**
  - Tracking is hard!
- **Allows us explore alternate algorithms**
  - Triplet-based seeding optimized for high multiplicity environments
- **Code optimization come for free**
  - Also explores modern computing architectures (ie: CUDA)

HOWTO: Tracking

1)Pattern recognition

- Create collection of hits corresponding to track candidates
- MCC: Conformal tracking
- ACTS: Triplet-based seeding

2)Track fit

- Kalman Filter to obtain track parameters
- Material description of detector required
- MCC: Straight Kalman Filter
- ACTS: Combinatorial Kalman Filter looks for hits compatible with seed

Truth Tracking

Validates detector description in ACTS.

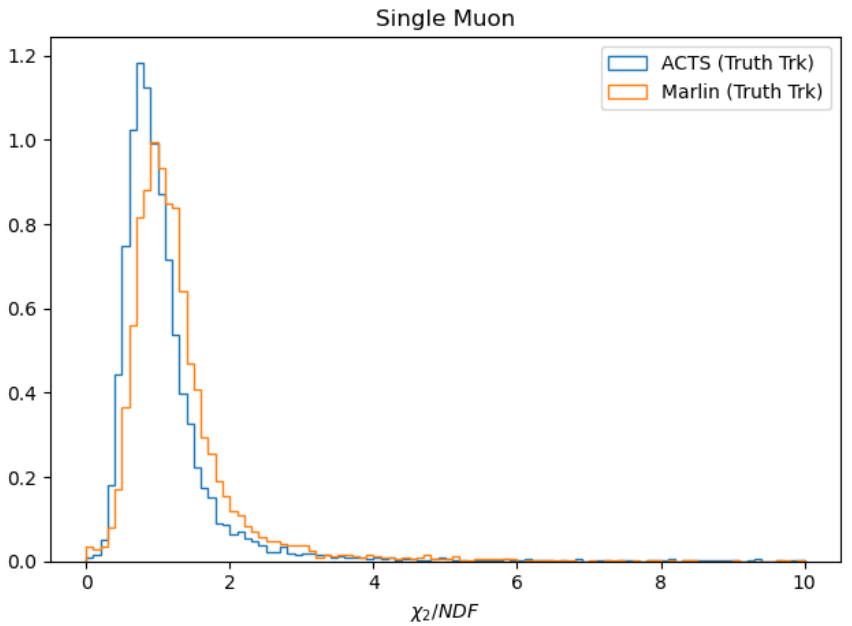
Pattern Recognition

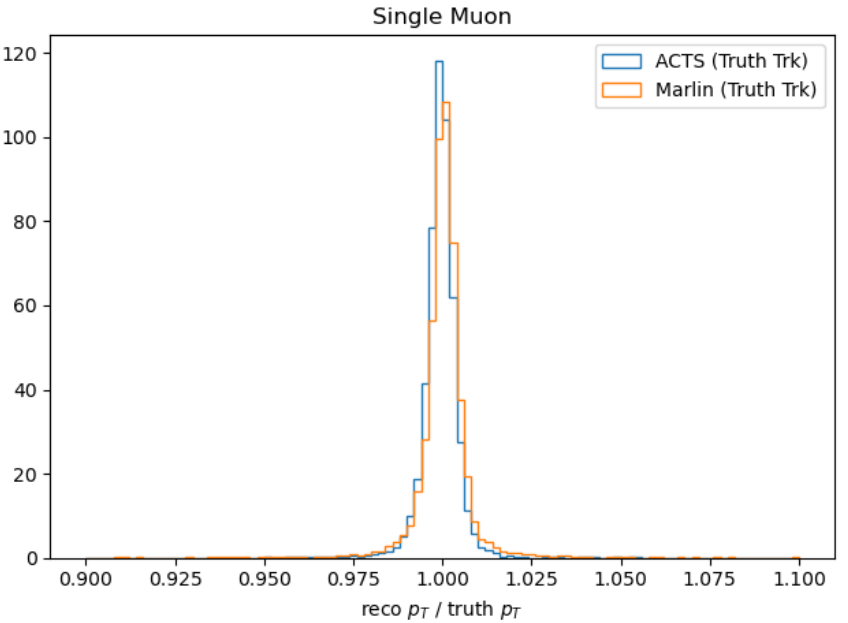
- Use hits associated to MC particle (100% efficiency)
- Same code for MCC and ACTS

Fit Library	Execution Time
ACTS	0.5 ms / evt
MCC	100 ms / evt

Track Fit

- Kalman Filter, but ACTS vs MCC implementation





Triplet Seeding

ACTS Documentation

1) Choose *N* layers for seeding

- 4 in our case

2) Form seeds containing three hits

- All possible combinations in *N* layers

3) Remove bad seeds based on estimated track parameters

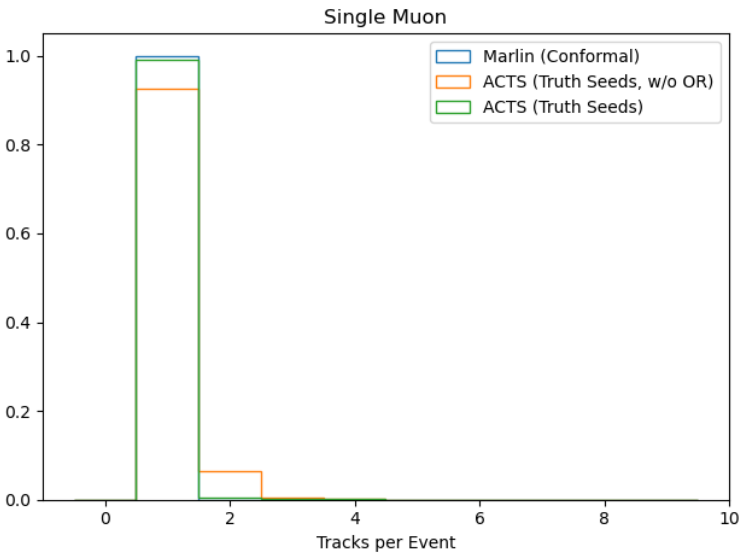
4) Remove overlap between seeds sharing the same middle hit

5) Use estimated track parameters as input to CKF

Add picture explaining this.

Description of CKF

Truth CKF Tracking



Seeding (the truth part)

- Use MC particle kinematics

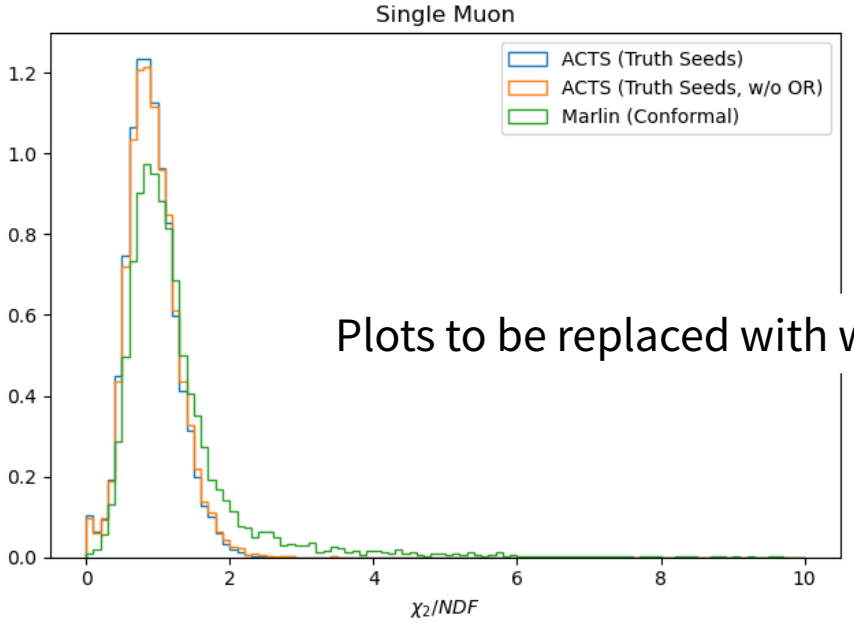
Track Fit

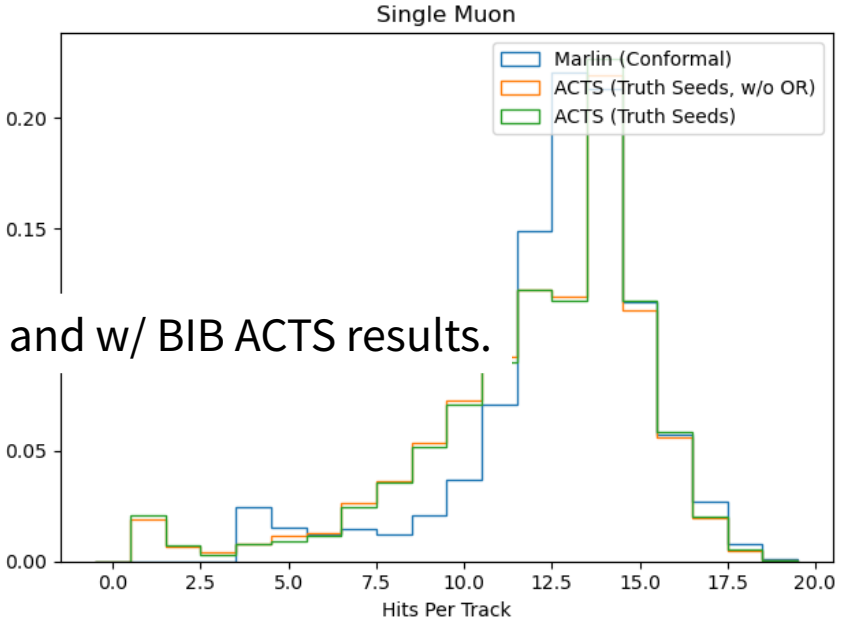
- Combinatorial Kalman Filter in ACTS

Fit Library	Execution Time
ACTS	0.5 ms / evt
Conformal	120 ms / evt

Overlap Removal

- Group by tracks sharing 50% of the hits, pick one with most (or highest χ^2)

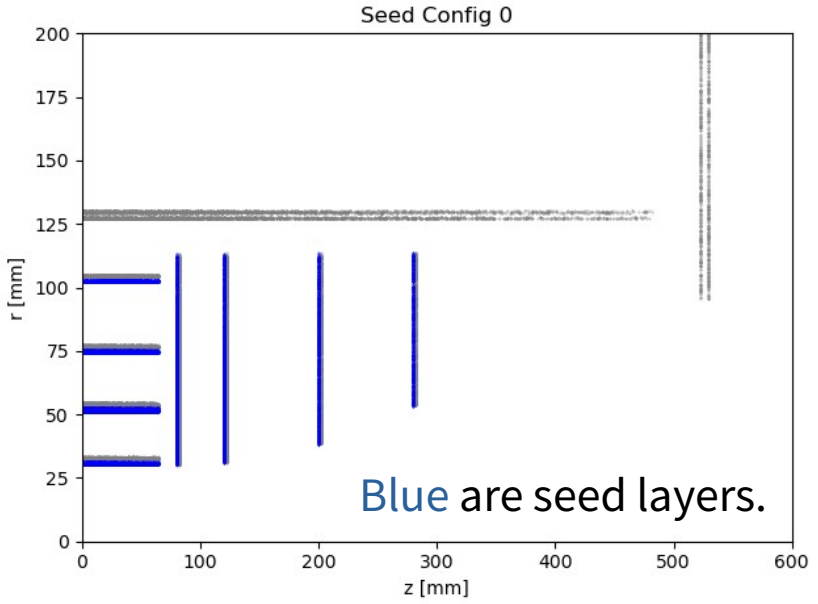


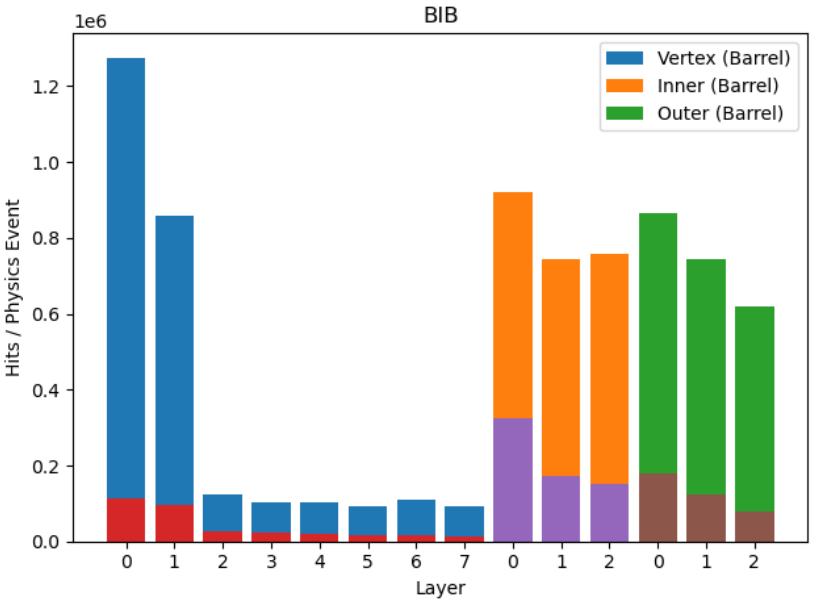


Plots to be replaced with w/o BIB and w/ BIB ACTS results.

Seeding Layers

All hits , second colors after timing cuts. Add time window in overlay



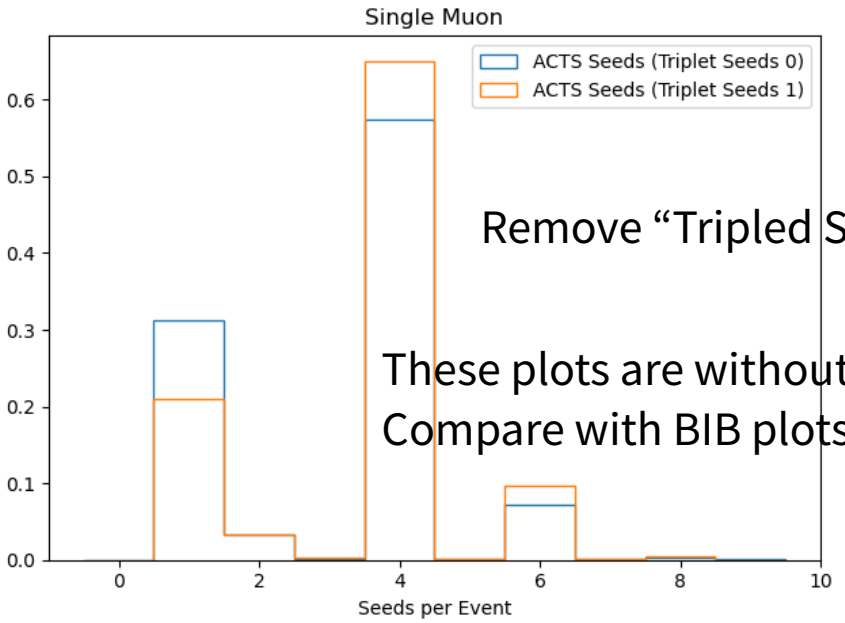


- **Using only inner part of the Vertex doubles**
  - Prevents redundant “too close together” combinations
  - Future: Reduce hits with doublet requirements in double layer?
- **~300k seeds per event**
  - 200 ms / seed x 300k seeds / event = ~16 hours / event

	Combinations
All Triplets	700B
Seeds	2000

Per region, x144 regions

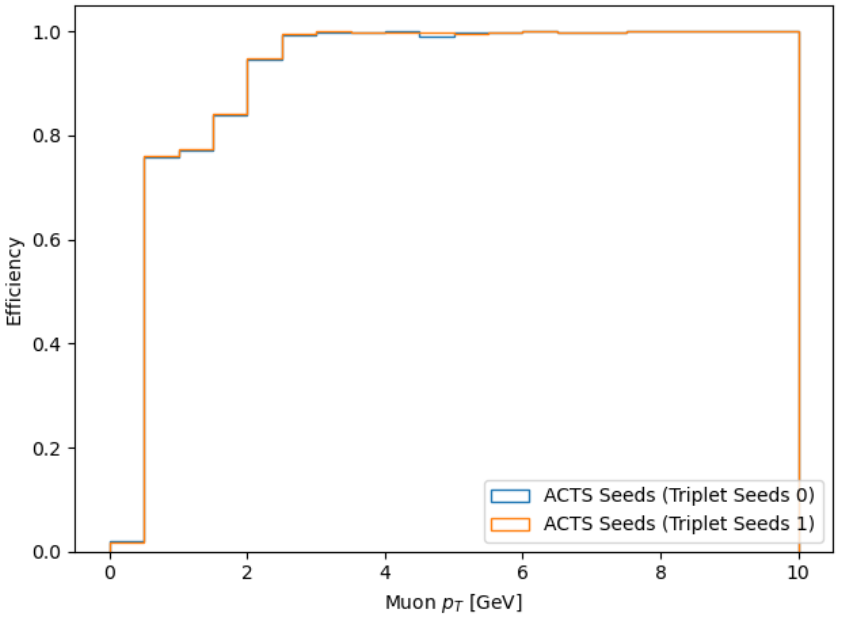
Found Seeds in Full BIB

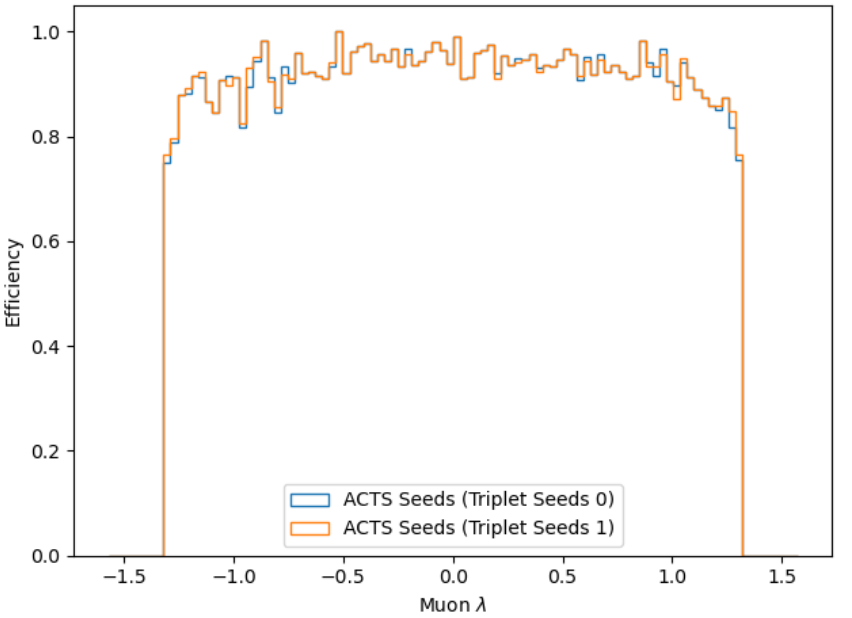


Remove “Tripled Seeds 1”

These plots are without BIB, Compare with BIB plots.

- **Assume hit in all 4 layers**
  - 3 choose 4 = 4
- **Missing seeds at low p_T**
- **Same efficiency in both**





Towards Seeded CKF

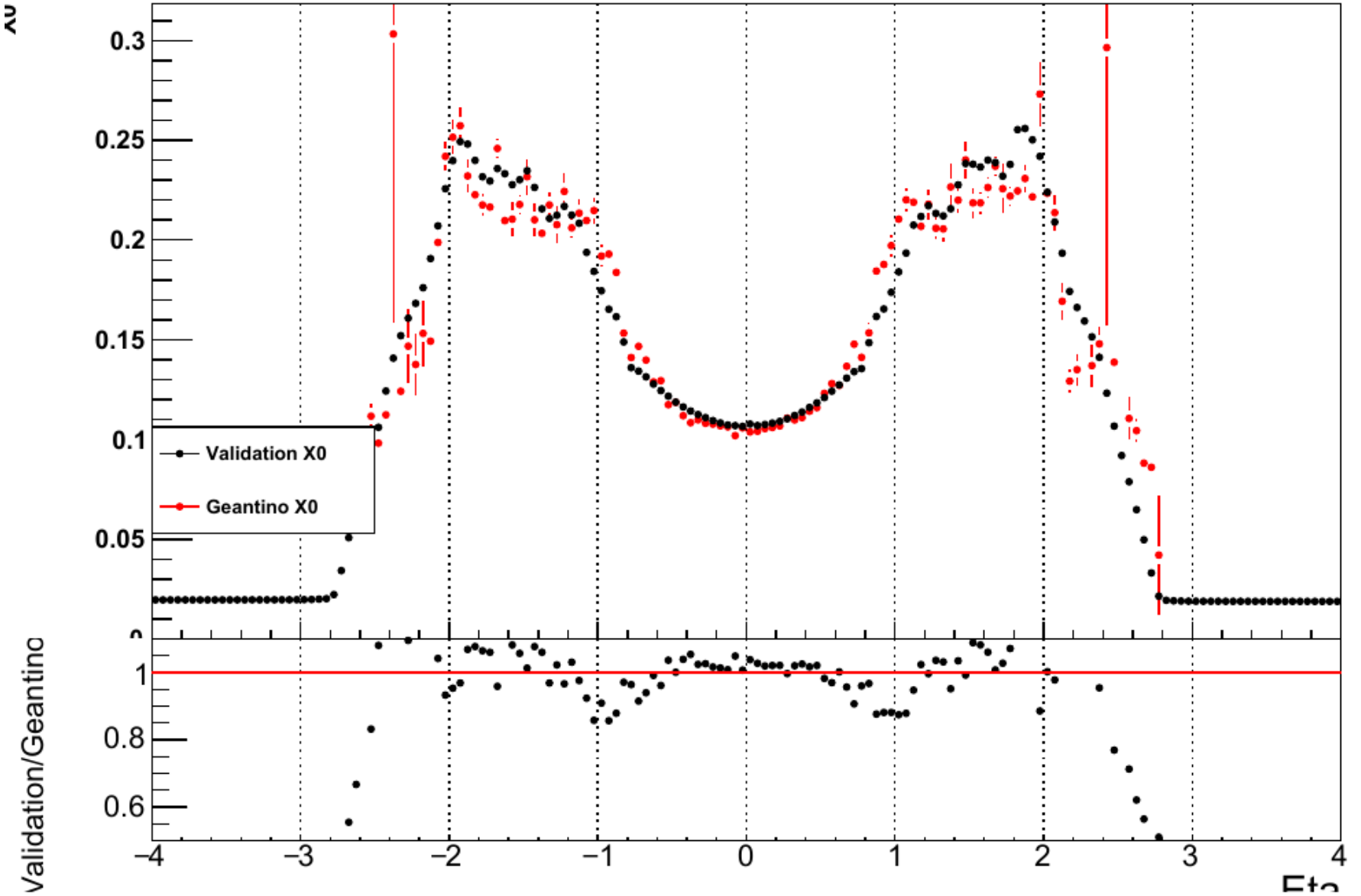
- **Need to reduce number of seeds by at least x10**
  - Reduce hits via more realistic digitization (Simone?)
  - Optimization of seed finding configuration
  - Consistent timing of hits within a triplet
  - Consistent hits within doublet layers
- **Need to recover seed efficiency at low p_T**
  - Optimization of seed finding configuration

Conclusions

- **Current baseline for tracking is conformal tracking**
  - Found to be sub-optimal in the μ+μ- environment
- **Tried to use algorithms from the LHC experiments**
  - Triplet-seeding + combinatorial kalman filter
  - Implemented using the ACTS library
- **BIB is too much even for triplet seeding out-of-the-box**
  - *X* seeds → 1 day / per event
  - Might need to exploit detector features for practical performance
- **ACTS implementation of common algorithms is faster**

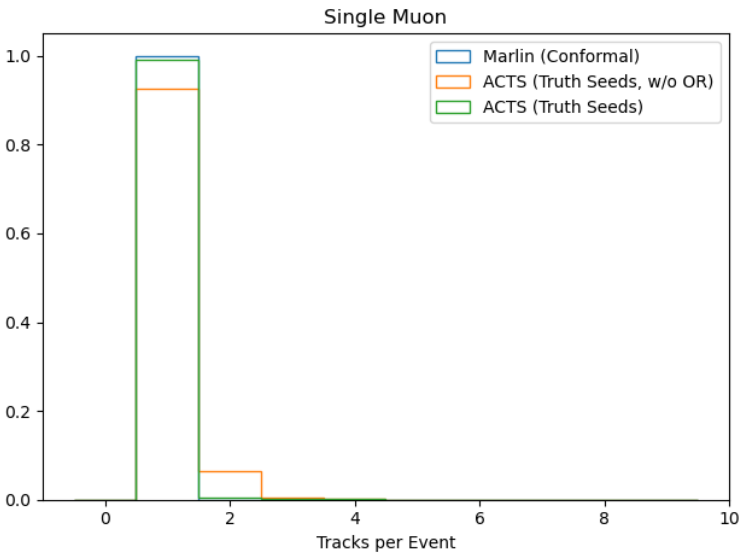
BACKUP

Material Validation



Add notes about importing MCC geometry into ACTS.

Truth CKF Tracking



Seeding (the truth part)

- Use MC particle kinematics

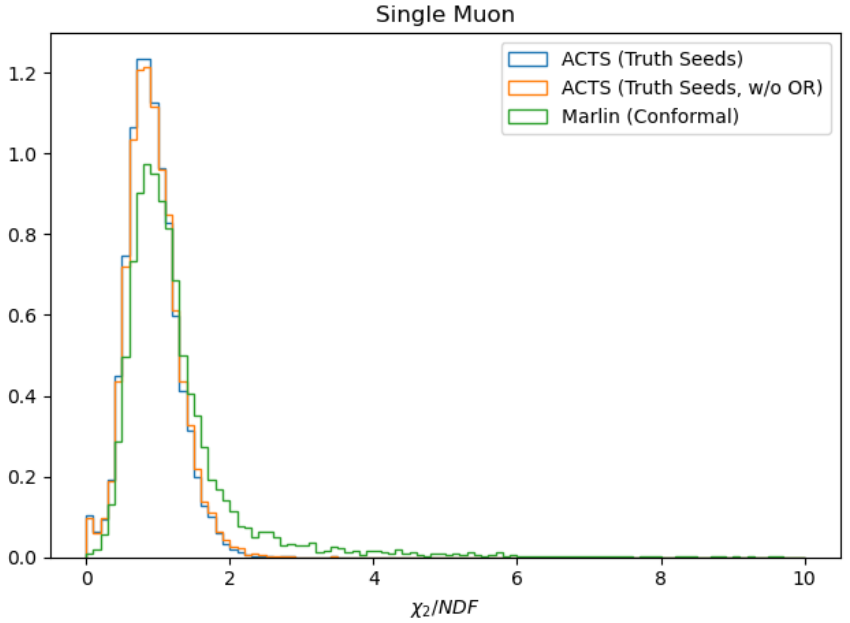
Track Fit

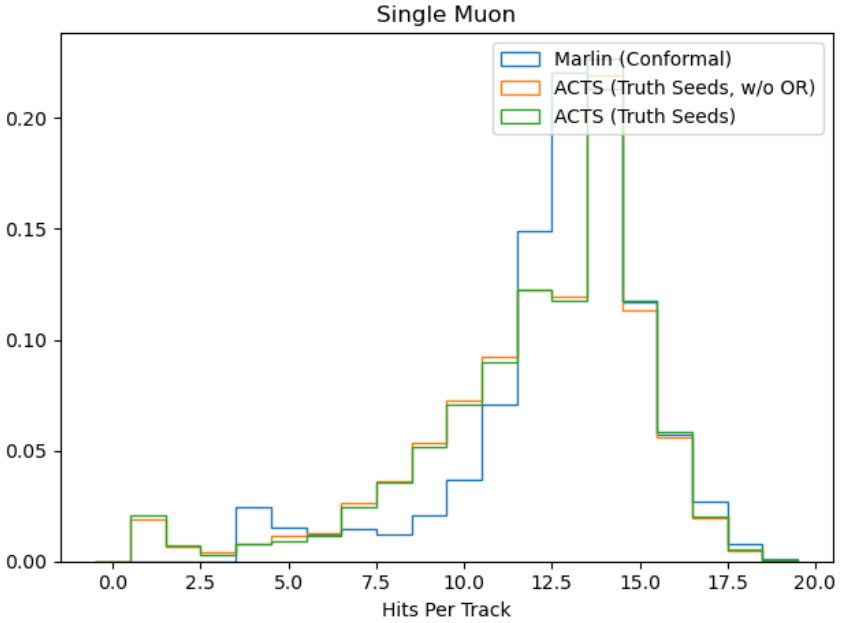
- Combinatorial Kalman Filter in ACTS

Fit Library	Execution Time
ACTS	0.5 ms / evt
Conformal	120 ms / evt

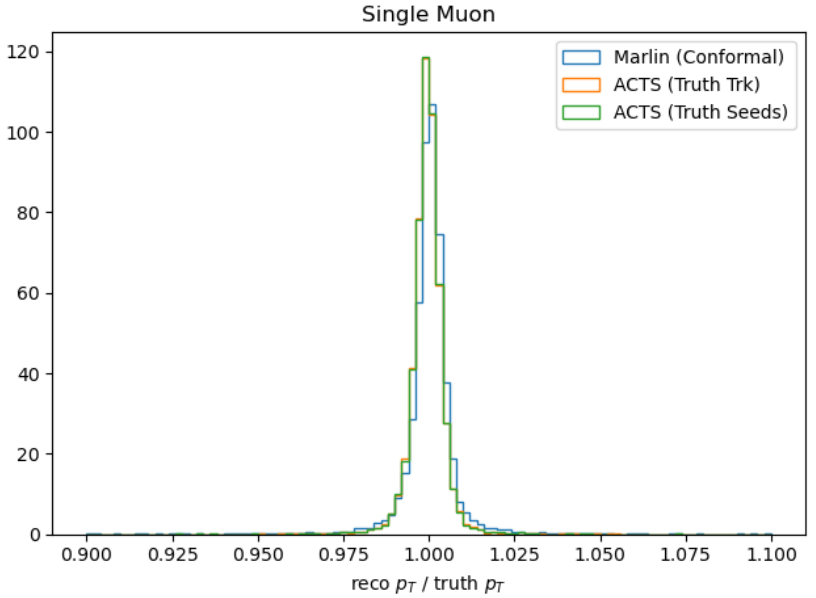
Overlap Removal

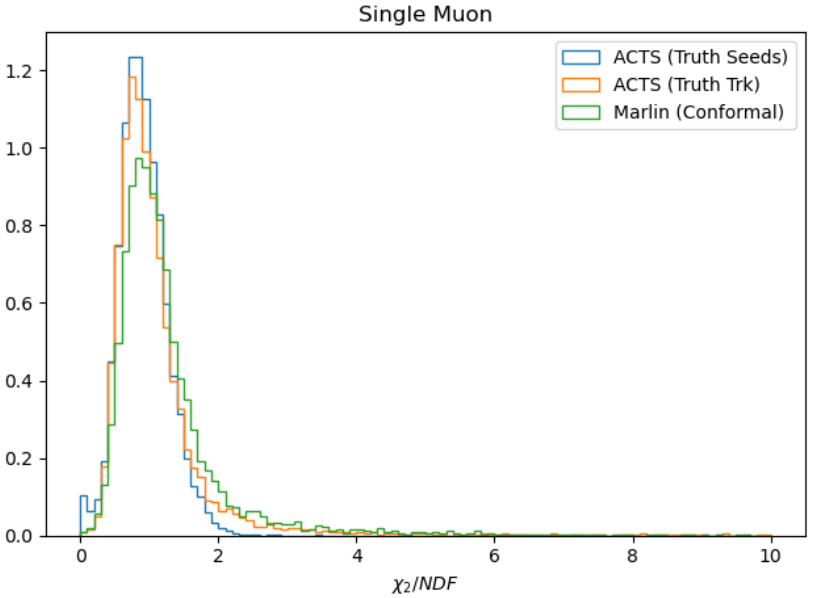
- Group by tracks sharing 50% of the hits, pick one with most (or highest χ^2)

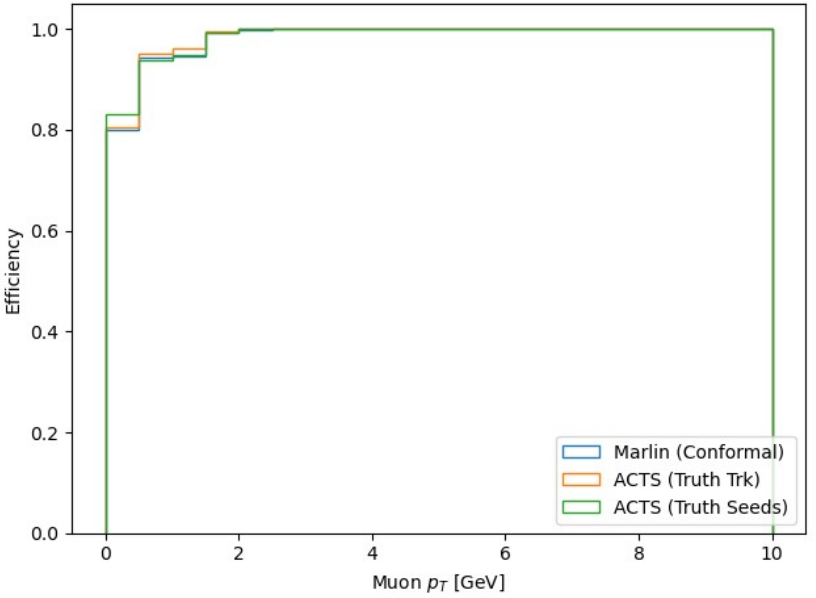


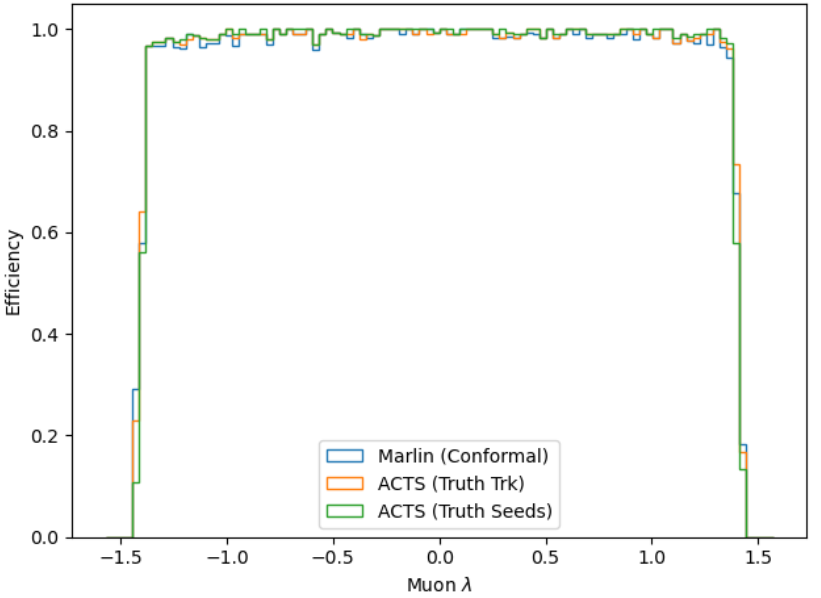


More Truth CKF







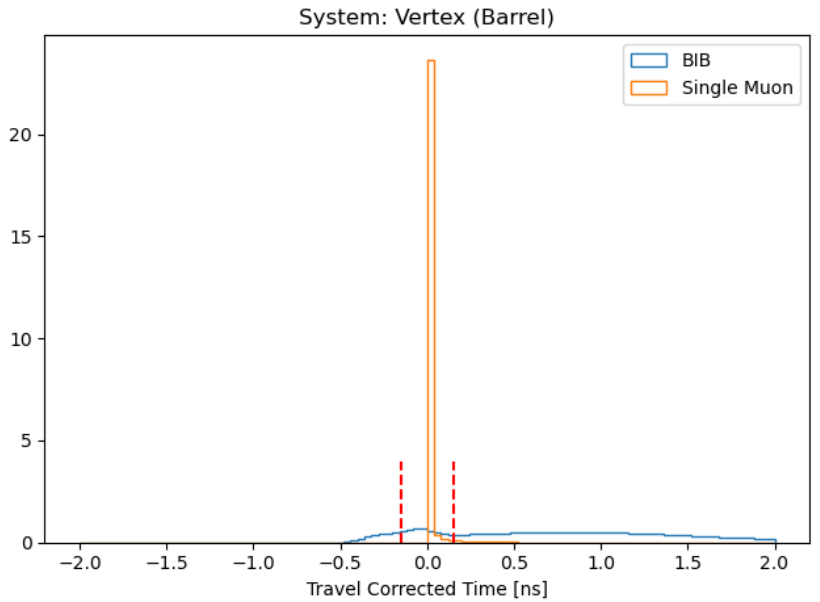


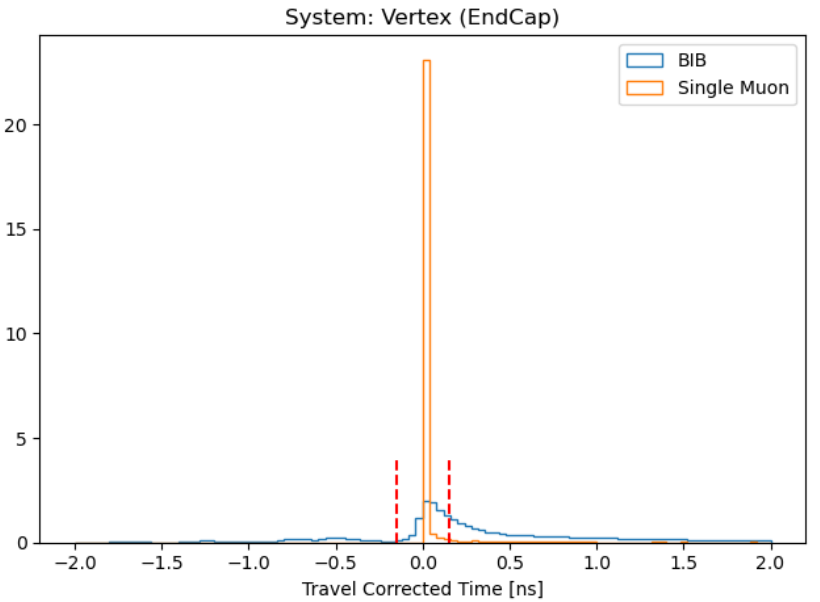
BIB Overlay Technical Notes

- **Using new BIB overlay files from Massimo**
  - Enough events to only overlay *five events* in "100% BIB"
- **Overlaying a single event takes ~*6 min***
- **Performing overlay is limited by disk reads**
  - Load 8GB worth of (compressed) data for each event…
  - `start-stop` timing shows most of the time is reading next BIB event
  - Callgrind also confirms this
- **Speed-up via pre-skimming MCParticle collection**
  - *Remove the MCParticle collection before using BIB files*
  - Drops size of single overlay to 4GB (*three times as fast*!)
  - Still need to fix some broken links…

BIB Timing

SimTrackerHit (no smearing, usually 50 ps)

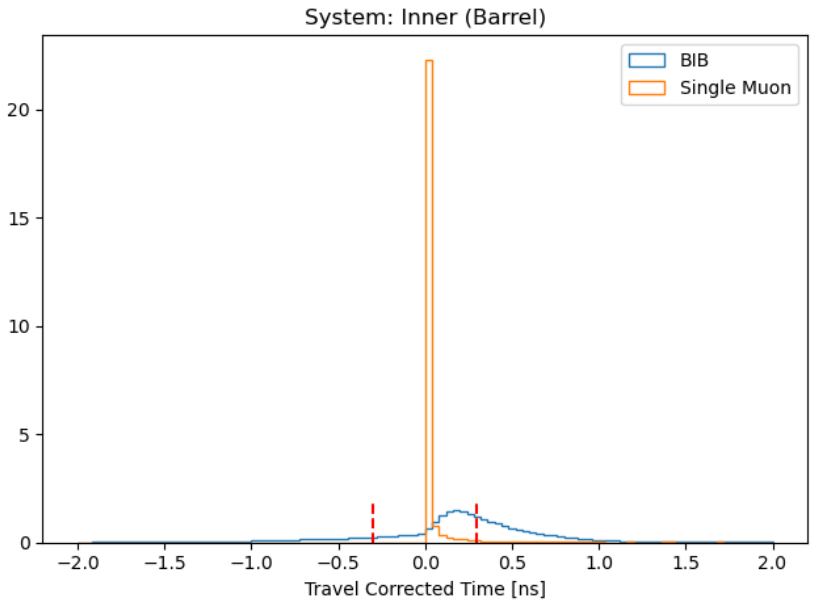


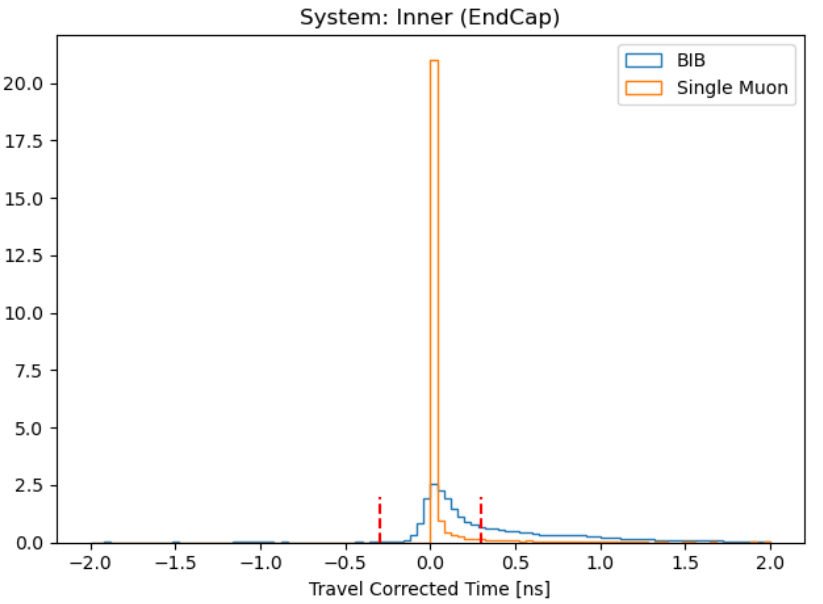


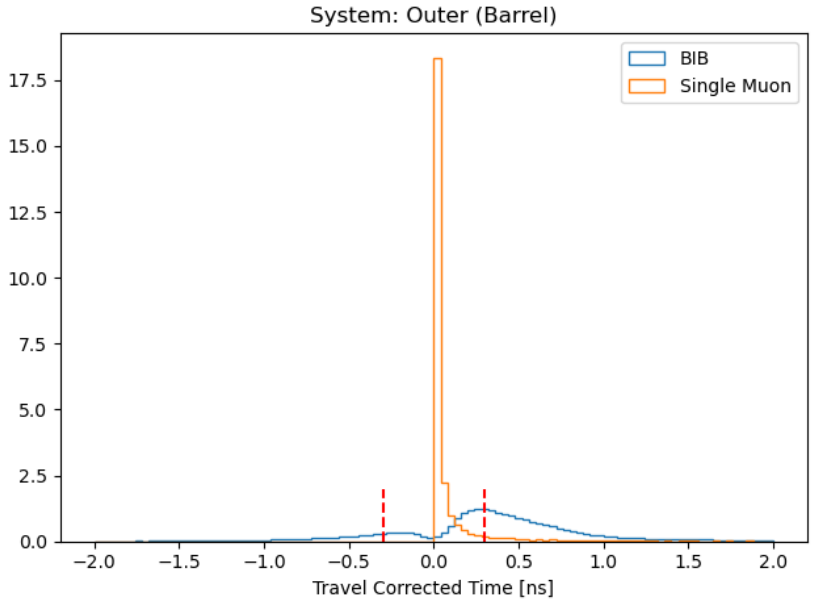
- **Based on SimTrackerHit (no smearing)**
  - Current default is 50 ps time resolution
- **Does not include cuts from Overlay processor**

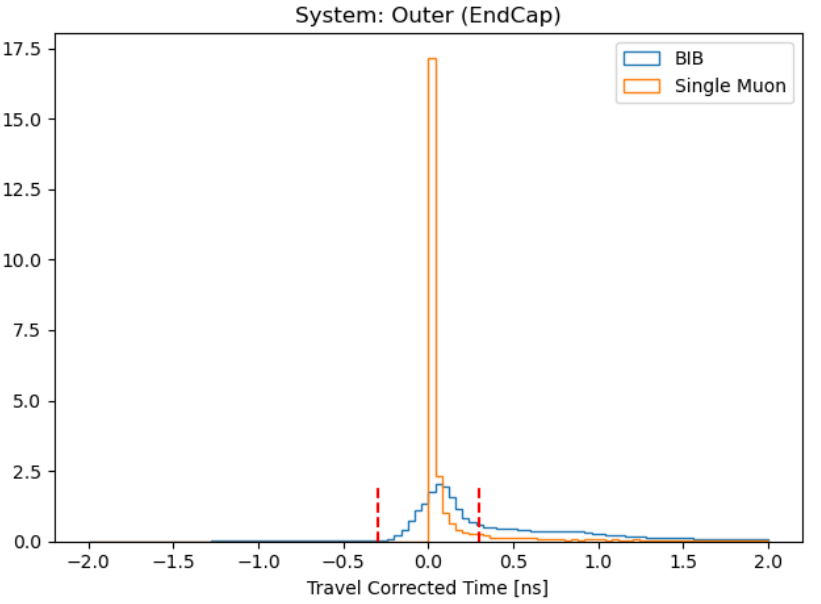
BIB Timing

SimTrackerHit (no smearing, usually 50 ps)

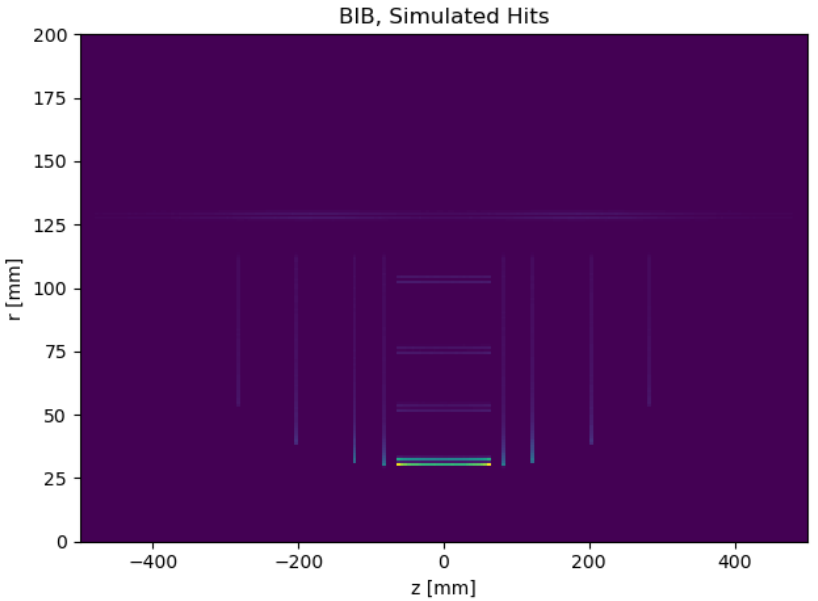


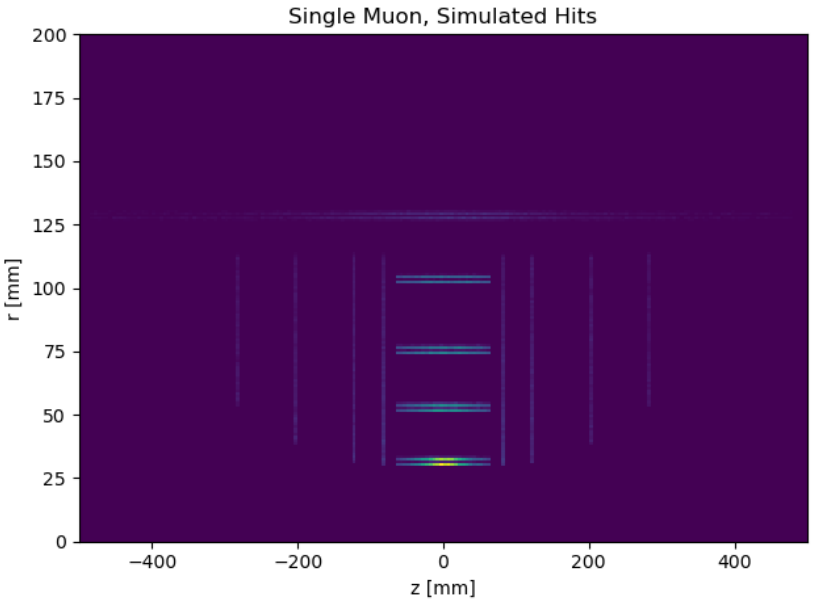


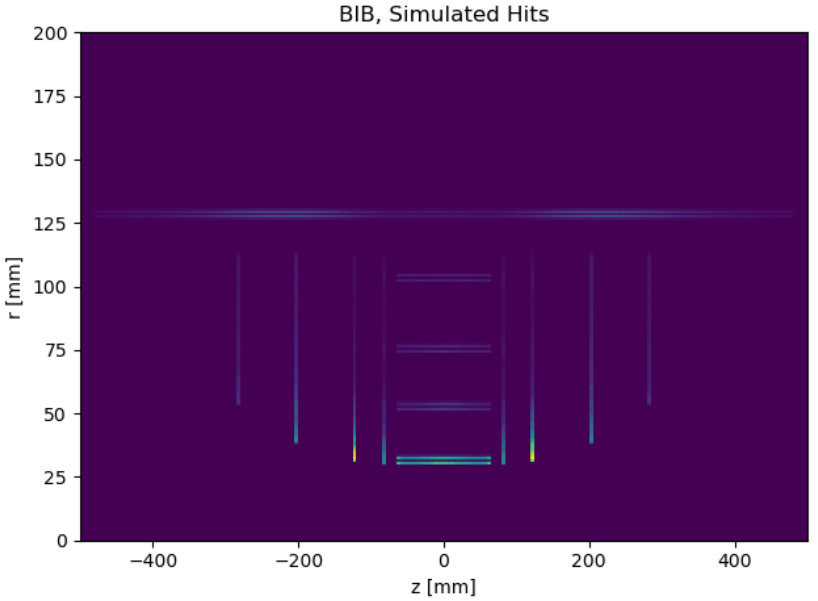


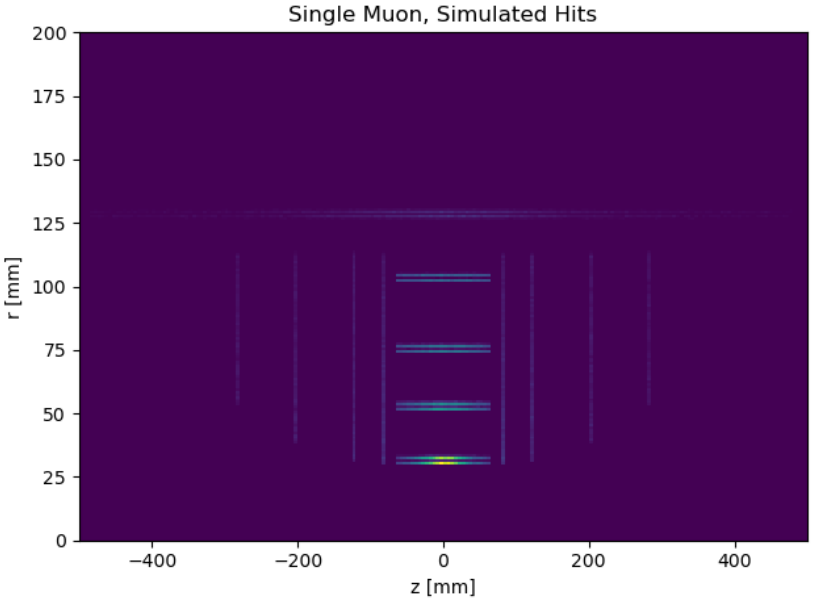


BIB Distribution



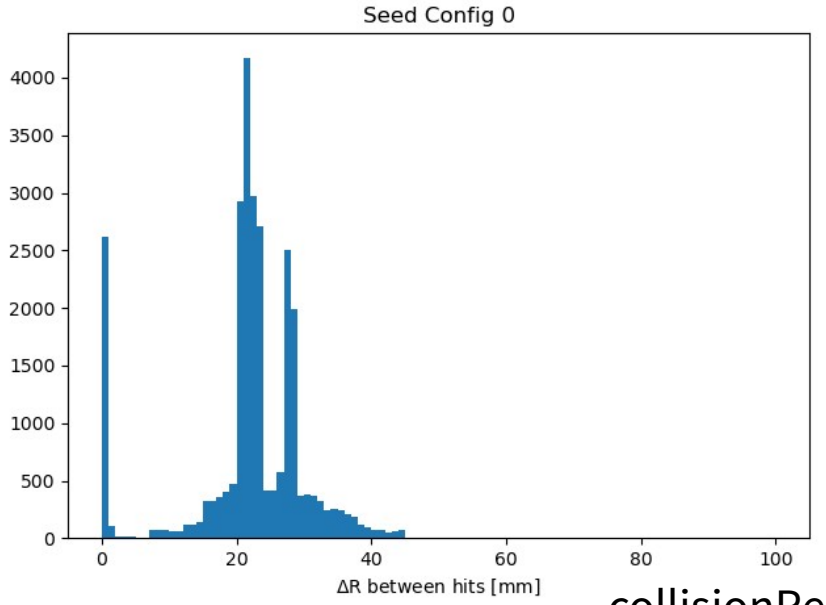




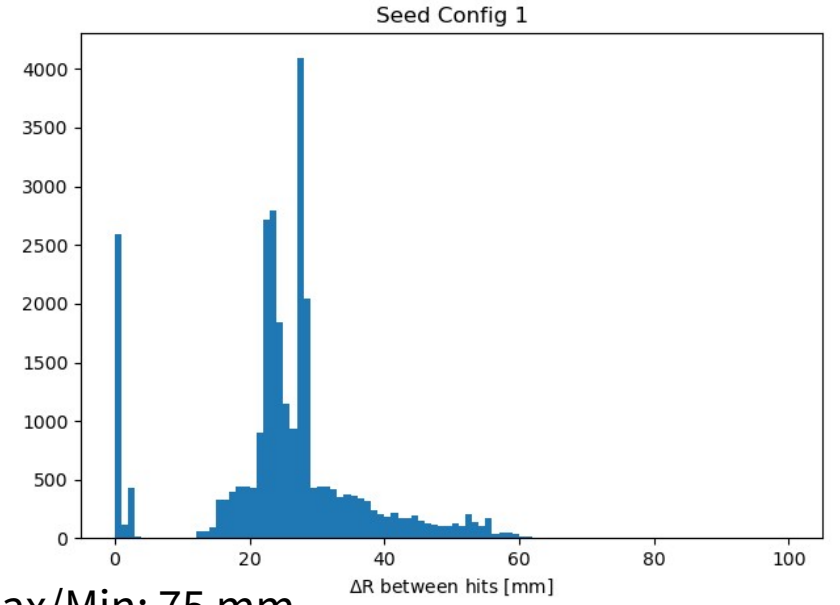


Optimizing Seeding Settings

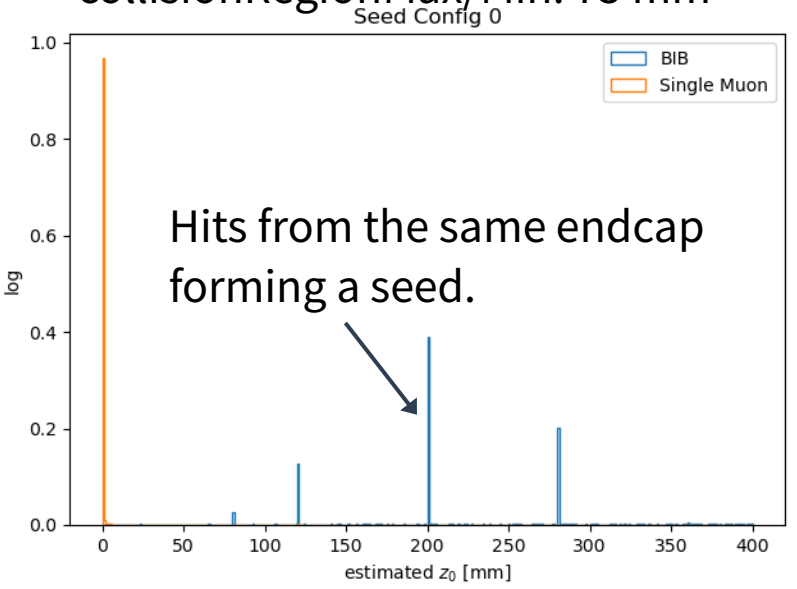
deltaRMin: 5 mm to remove same layer



deltaRMax: 80 mm



collisionRegionMax/Min: 75 mm



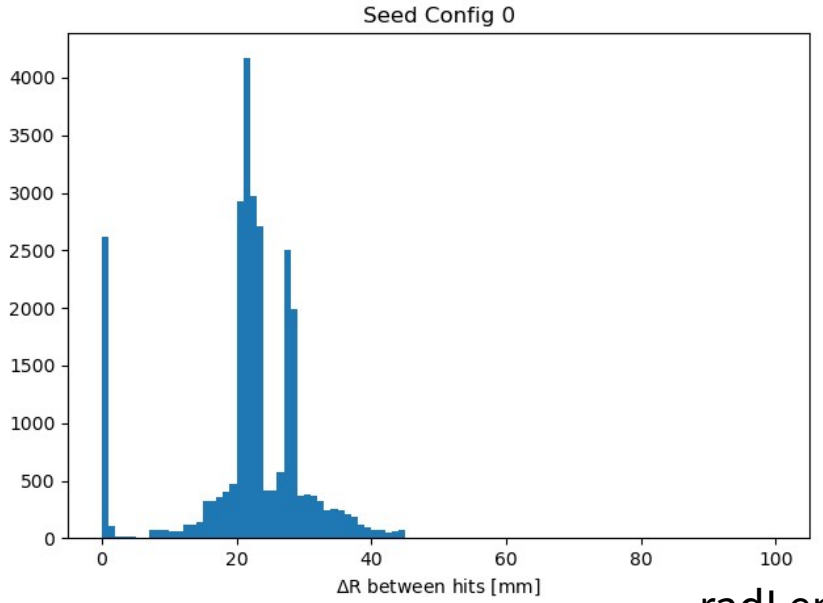
Hits from the same endcap forming a seed.

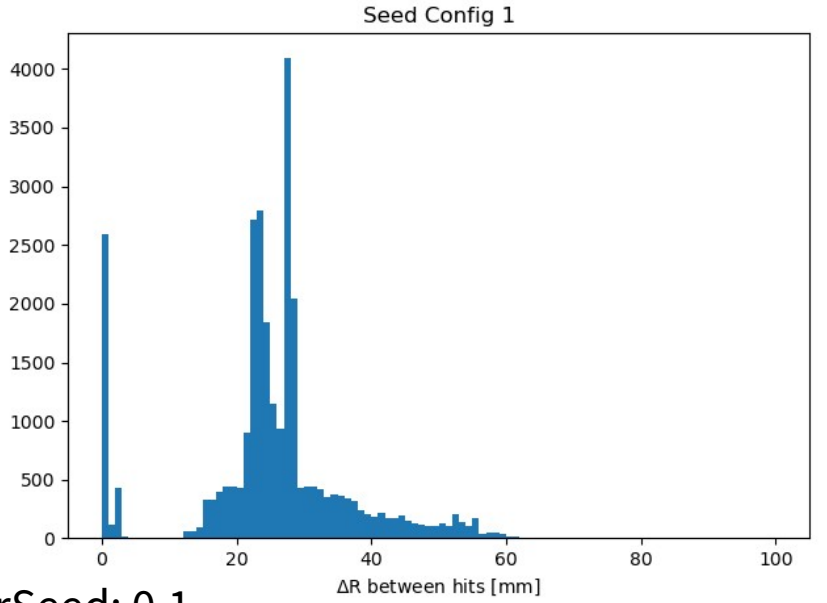
Bunch length:
5 mm to 10 mm
Maybe try 30 mm?

Need to keep collisionRegion cuts loose to allow for displaced tracks

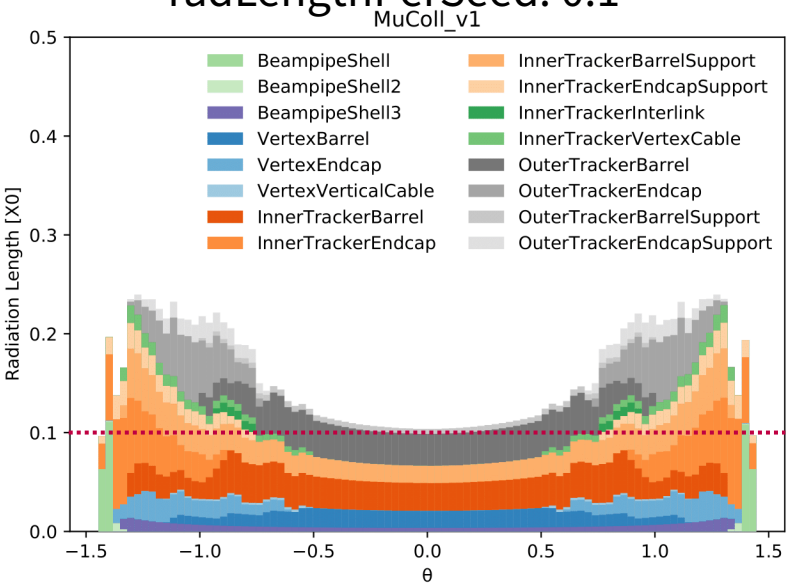
Optimizing Seeding Settings

deltaRMax: use 80 for both

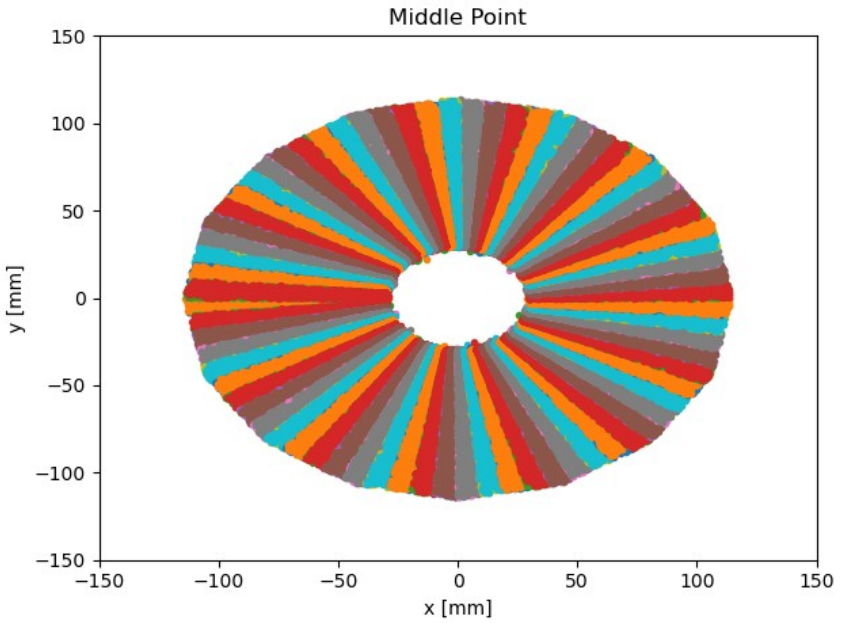


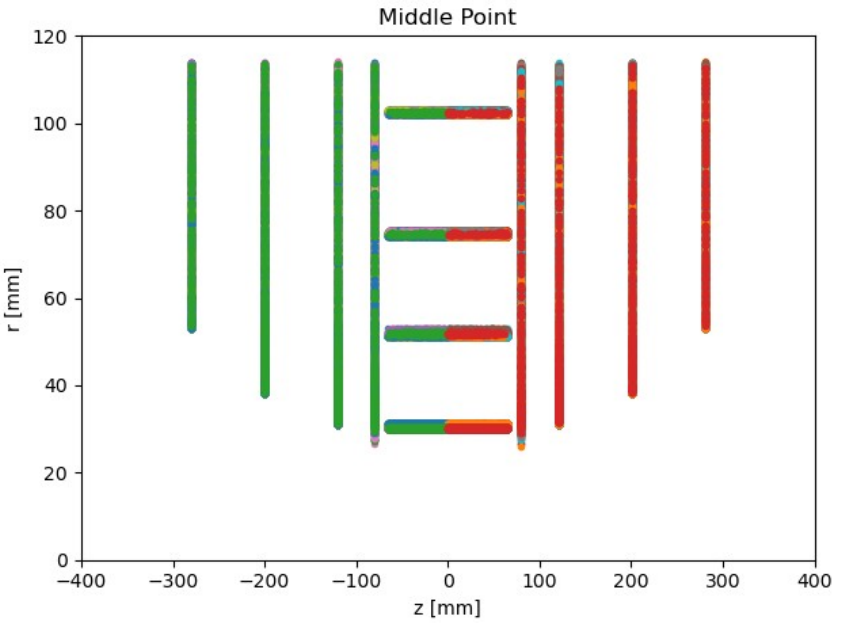


radLengthPerSeed: 0.1

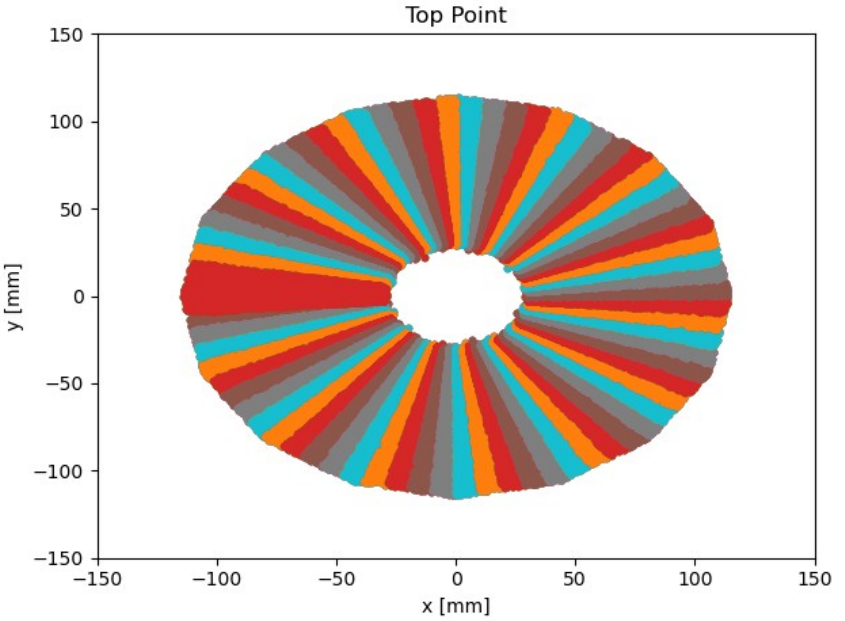


Seed Groups (Cfg 0)





- **ACTS looks for seeds in overlapping groups (binning)**
  - Middle point is binned in z (2) and ϕ (72)
  - Top/Bottom points are binned more coarsely (and overlap) in ϕ only
  - Top/Bottom bins seem to be identical
- **How is the size of top/bottom bins set?**



Combinations in Each Group (with BIB)

	Config 0	Config 1
Top	16278	25536
Middle	2745	4227
Bottom	16278	25536
Comb	700B	3T
Rd Comb	800M	1.1B
Seeds	2000	2000

1) O(trillion) combinations in each group

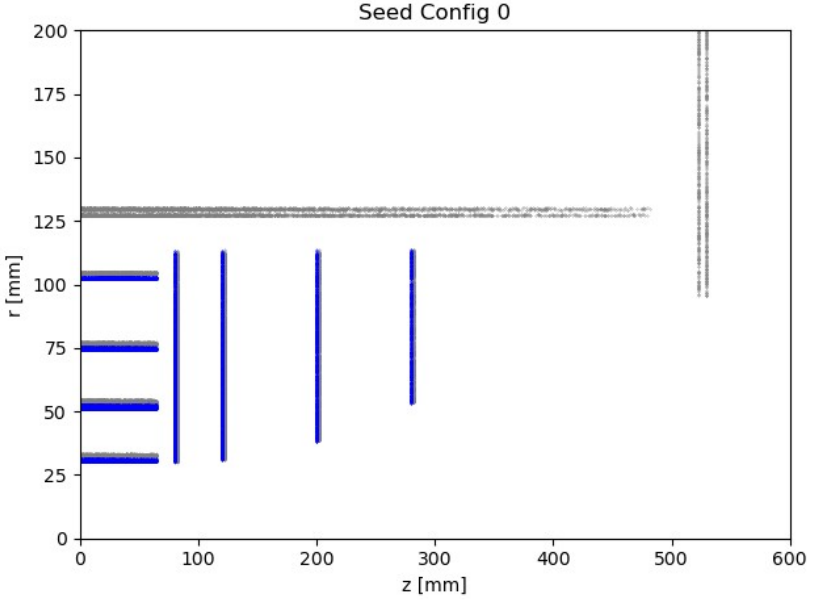
2) O(1 billion) possible seeds after initial geometry cuts

3) O(1000) final seeds after helix estimate and overlap removal

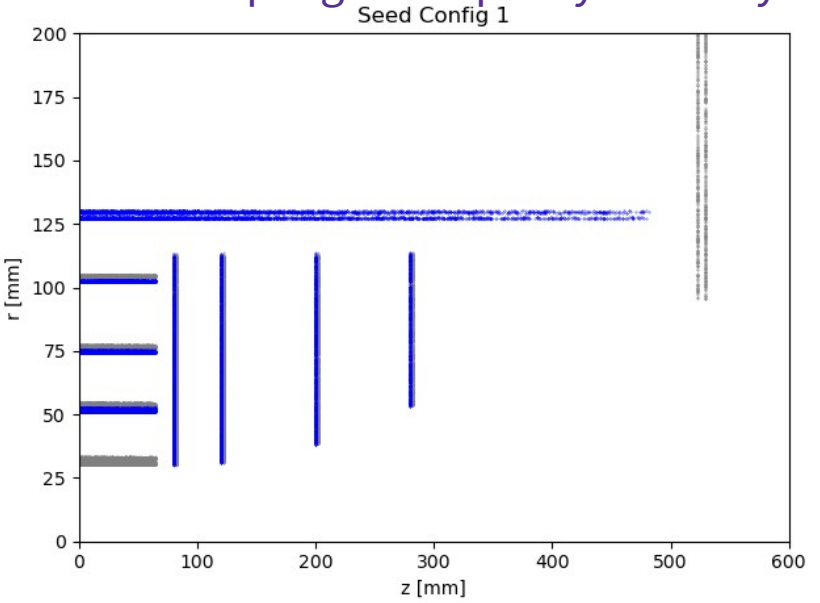
- This is the slowest step

Seeding Layers

Seed 0: Vertex detector



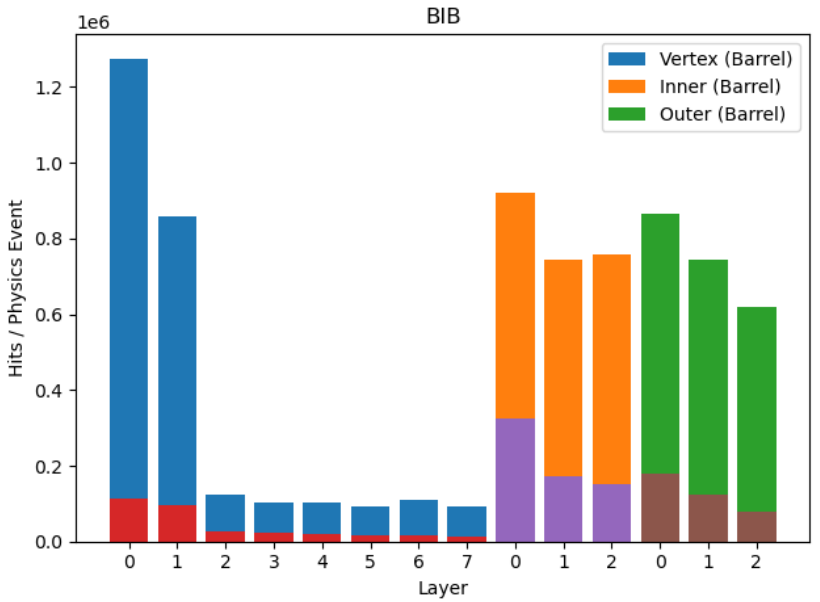
Seed 1: Skip high occupancy inner layer

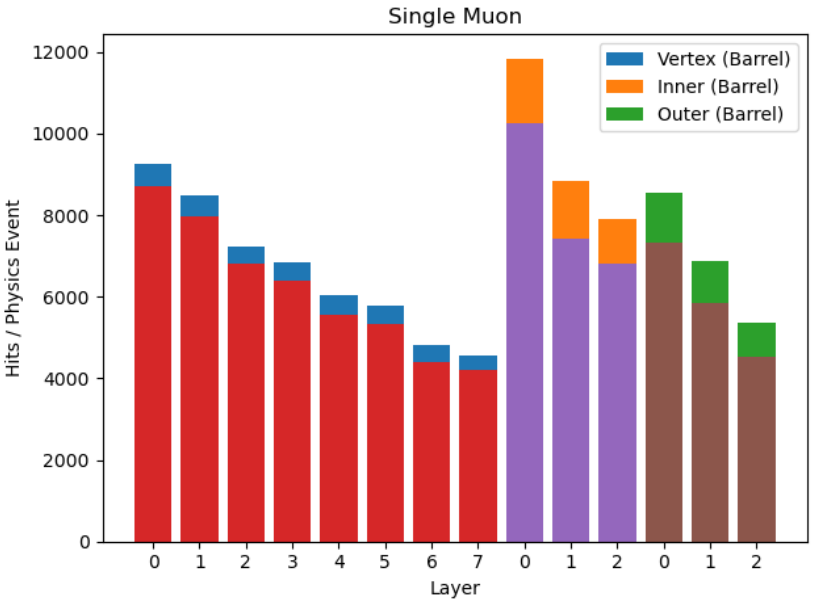


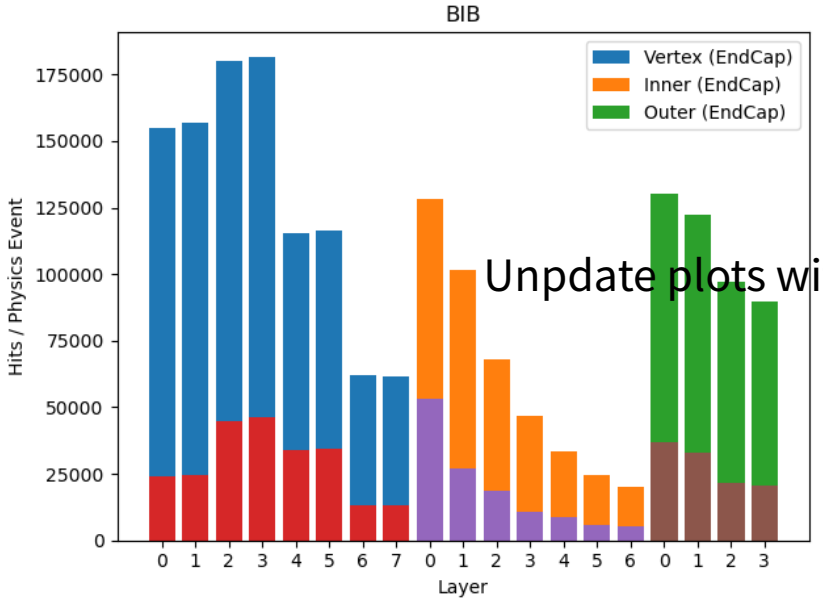
- **Using only inner part of the Vertex doubles**
  - Prevents redundant “too close together” combinations
  - Future: Reduce hits with doublet requirements in double layer?
- **Seed 1 reduces combinations by avoiding innermost layer**
  - Keeps inner endcap for coverage, occupancy high only at small R

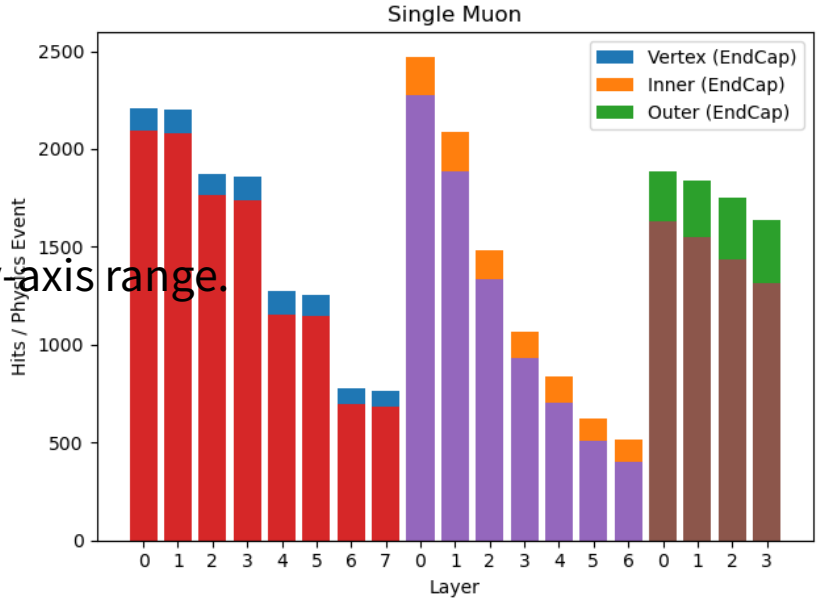
BIB Distribution

Second color is number of hits after timing cuts.









Unpdate plots with same y-axis range.