Update on Long-lived Particle Study – Tracking Differences Between MuCol Software Versions

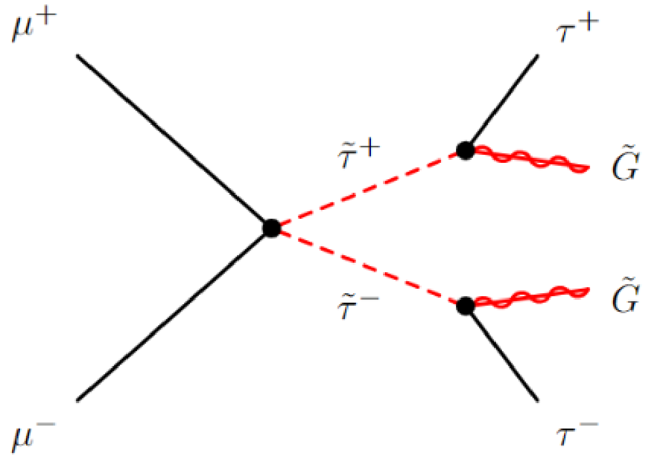
Mark Larson, Tate Flicker, Kane Huang, Leo Rozanov, Ben Rosser, Karri DiPetrillo

The University of Chicago

October 15, 2024

Motivation & Project Goals

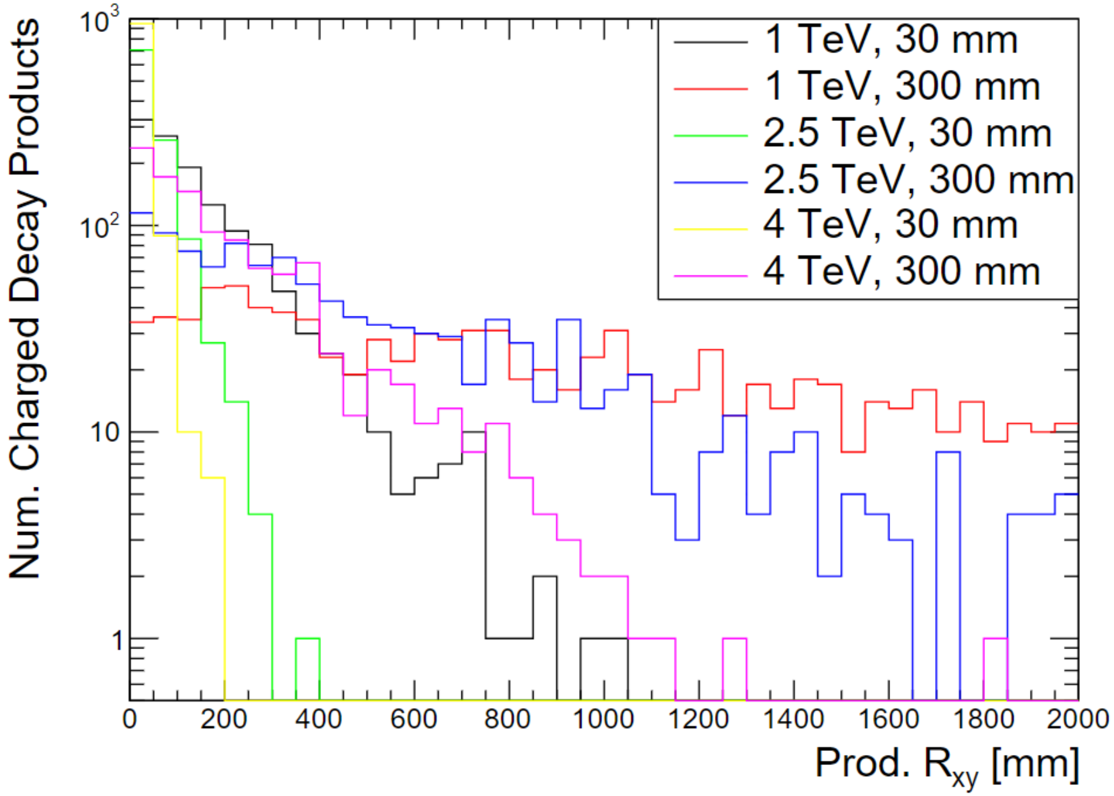
- Want to demonstrate feasibility of **direct** and **indirect detection** of **long-lived particles** at a 10 TeV muon collider, especially **in presence of BIB**
  - Karri's talk about experimental challenges of LLPs at muon collider
  - Use baseline GMSB model to study $\tilde{\tau}'s$, allowing for indirect & direct detection
- Develop reconstruction methods to **maximize efficiency reconstructing $\tilde{\tau}$ and displaced tracks**, while **minimizing hits & tracks from BIB**
  - Provide efficiencies/sensitivities given varying levels of signal acceptance & BIB background rejection based on **timing windows, track requirements**
- This talk is an update w.r.t. previous talk given by Leo Rozanov on June 4th



Aim to reconstruct $\tilde{\tau}$ & τ decay products (e, μ, π)

Signal Generation & Simulation

- Generate signal points with $m_{\tilde{\tau}}$ between 1, 4.5 TeV, and mean decay time between 0.05 and 10 ns ($c\tau$ between 15, 3000 mm) using MadGraph & Pythia



Displacement distributions of charged τ decay products given mass and mean decay length of $\tilde{\tau}$

- Then use DD4HEP & GEANT4 to simulate detector response, Marlin & Gaudi to digitize detector response, reconstruct physics objects

Extend tracker hit timing acceptance windows

Use modified ACTS track reconstruction algorithm to reconstruct displaced tracks

Note: using 3 TeV detector design

Timing Windows

- For higher mass signal points ($m_{\tilde{\tau}} \geq 2.5$ TeV) $\tilde{\tau}$ slowly moving
- Tracker hits from $\tilde{\tau}$, τ decay products often fall outside nominal timing windows, resulting in worsened tracking efficiency

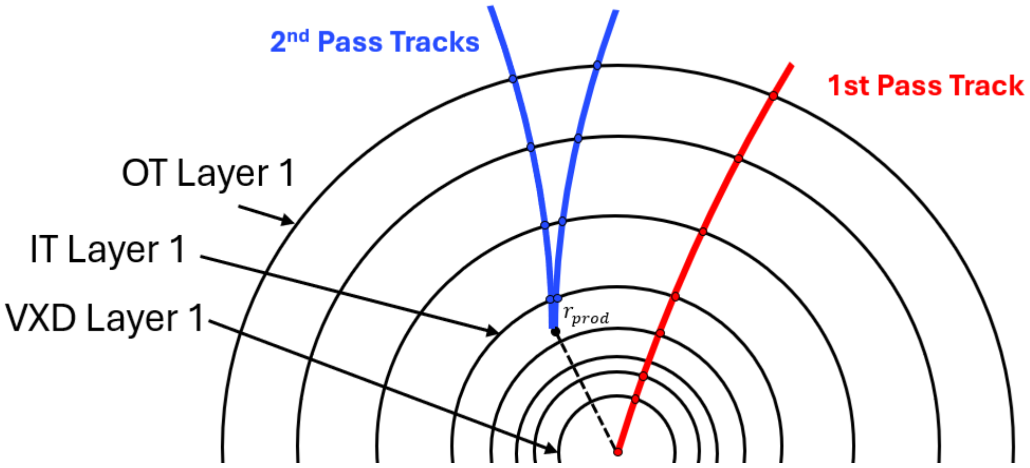
Detector	Min. t_{hit} [ns]	Max t_{hit} [ns] (nominal)	Max t_{hit} [ns] (extended)
Vertex Detector	-0.09	0.15	0.32
Inner Tracker	-0.18	0.30	0.64
Outer Tracker	-0.18	0.30	0.64

- Have defined tight, medium, and loose timing windows based on acceptance of $\tilde{\tau}$, τ decay products

Displaced Tracking Configuration

- To allow for reconstruction of displaced tracks, run two passes of ACTS track reconstruction based off methodology from Federico Meloni

Parameter	1st Tracking Pass	2nd Tracking Pass
Layers for Seed Finding	Vertex Detector Barrel & Endcap	OT Barrel & Endcap, 3rd IT Layer
Collision Region	1 mm	100 mm
Maximum Track d_0	5 mm	150 mm
Seed Finding r_{max}	150 mm	1500 mm
Minimum p_T	0.5 GeV	1 GeV



- Will later incorporate max. χ^2_{red} and min. N_{hits} for both passes

Different Software Versions

- Originally setup modified tracking configuration and obtained results using v2.8 of the muon collider software stack
- In August migrated to v2.9, integrating changes to k4run Python steering files
- Resulted in differences in tracking efficiency, despite using the same parameters in the steering file
  - Would like to understand these differences and whether they could be mitigated
- To process BIB overlayed samples (for which reconstruction takes especially long) developed method of job submission on Open Science Grid (OSG)
  - OSG has dedicated resources for muon collider studies

Analysis Setup

- pyLCIO analysis script & hit-based matching using lcRelation matches reconstructed tracks to $\tilde{\tau}, \tau$ decay products (displaced tracks)
- Define acceptance as:
  - **Prompt:** Decays past last VXD layer
  - **Displaced:** Produced within final IT layer, decays outside final OT layer
  - Note: acceptance not dependent on timing information
- Analysis level selections:
  - **Prompt:** ≥ 7 VXD barrel hits
  - **Displaced:** ≥ 4 IT hits + OT hits
- Fake tracks defined by:
  - Multiple unrelated Monte Carlo particles (MCPs) matched to same track
  - Track w/o any related MCPs

Changes in Displaced Tracking Eff.

Note: all results shown **w/o BIB overlay**, for 1000 events

Sample	Eff. (v2.8)	Eff. (v2.9)	AxE. (v2.8)	AxE. (v2.9)	Fake Rate (v2.8)	Fake Rate (v2.9)
1 TeV, 0.1 ns	61.5%	59.8%	49.3%	47.8%	0.086	0.116
1 TeV, 1 ns	48.0%	40.5%	11.2%	9.7%	0.198	0.178
2.5 TeV, 0.1 ns	63.3%	62.1%	53.2%	52.3%	0.081	0.100
2.5 TeV, 1 ns	38.3%	30.1%	19.4%	15.4%	0.054	0.048
4 TeV, 0.1 ns	63.3%	64.2%	51.3%	52.1%	0.095	0.118
4 TeV, 1 ns	33.5%	31.8%	24.6%	23.4%	0.019	0.028

Tracking efficiency generally worsened, especially for higher displacement samples

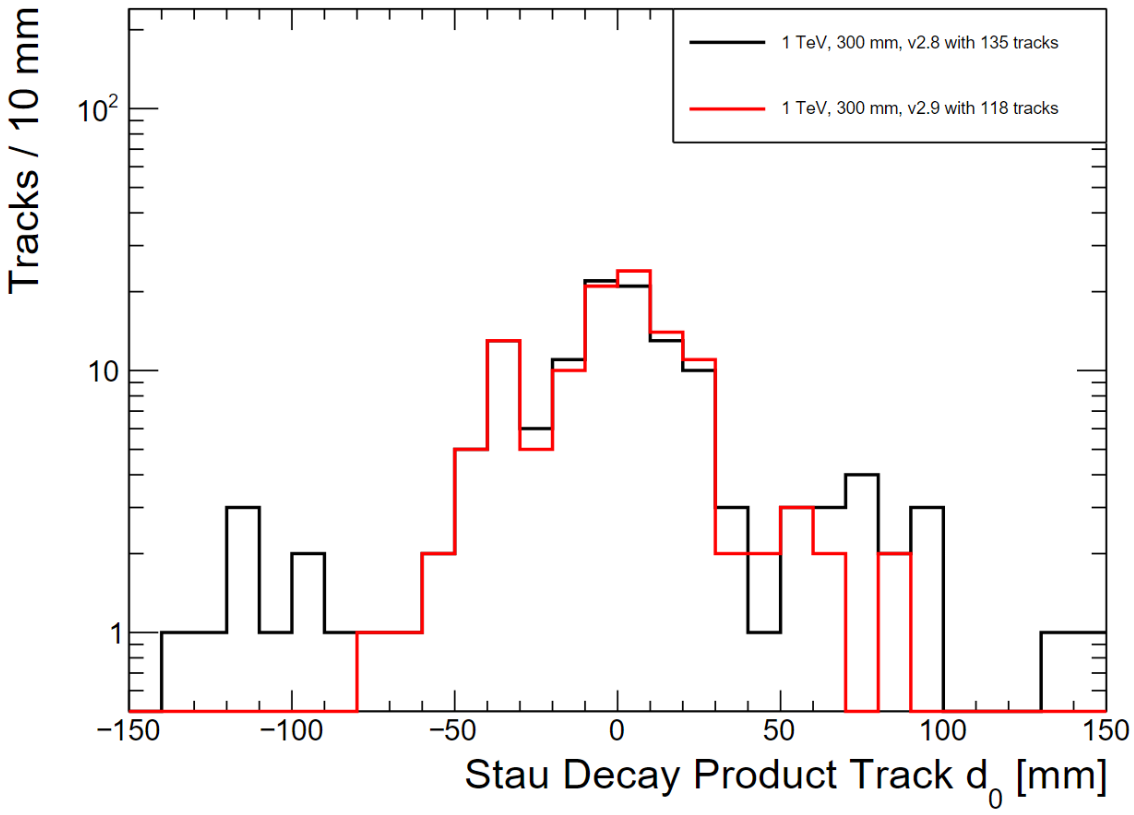
Changes in $\tilde{\tau}$ Tracking Eff.

Sample	Eff. (v2.8)	Eff. (v2.9)	AxE. (v2.8)	AxE. (v2.9)
1 TeV, 1 ns	82.2%	82.1%	76.9%	76.7%
2.5 TeV, 1 ns	89.0%	88.9%	72.2%	72.3%
2.5 TeV, 10 ns	88.2%	87.6%	85.4%	85.2%
4 TeV, 1 ns	84.0%	83.3%	50.5%	49.6%
4 TeV, 10 ns	84.8%	84.0%	80.3%	79.5%
4.5 TeV, 10 ns	11.2%	11.1%	10.4%	10.3%

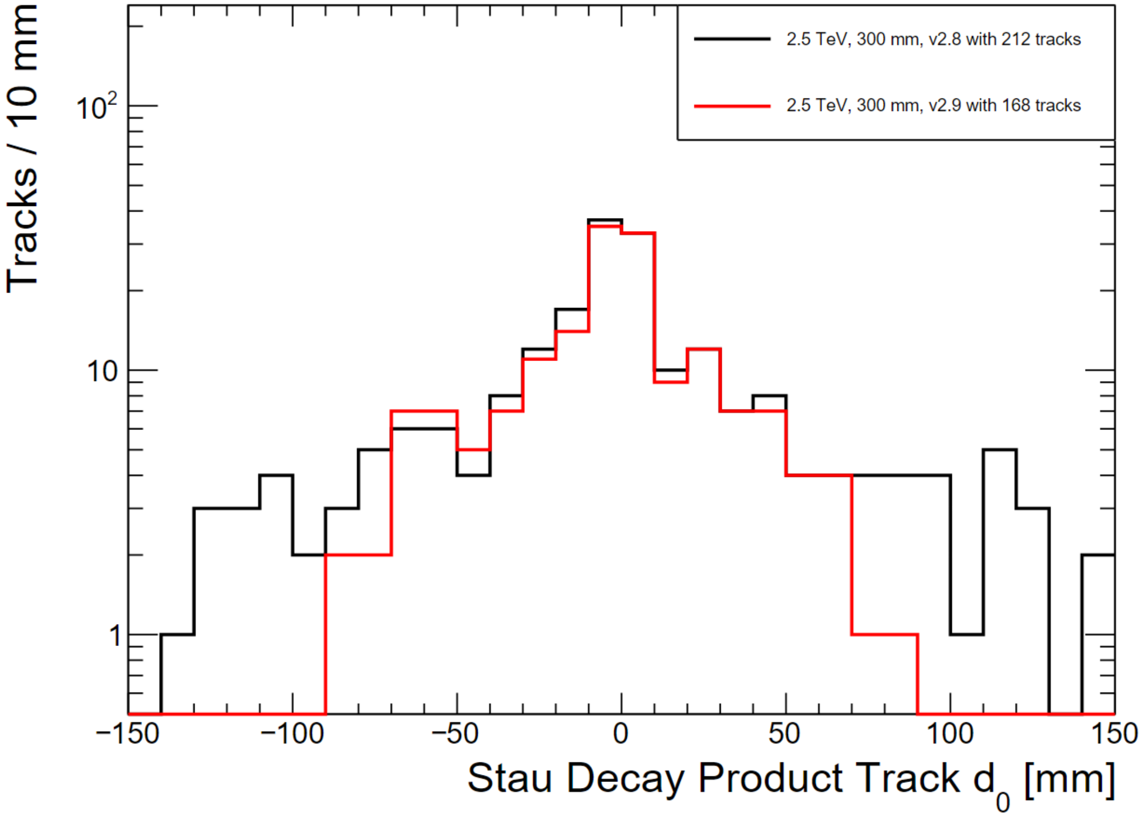
Efficiencies largely unchanged for prompt $\tilde{\tau}$ tracks

Displaced Track d_0

Note: all plots using 1000 signal events without BIB overlay

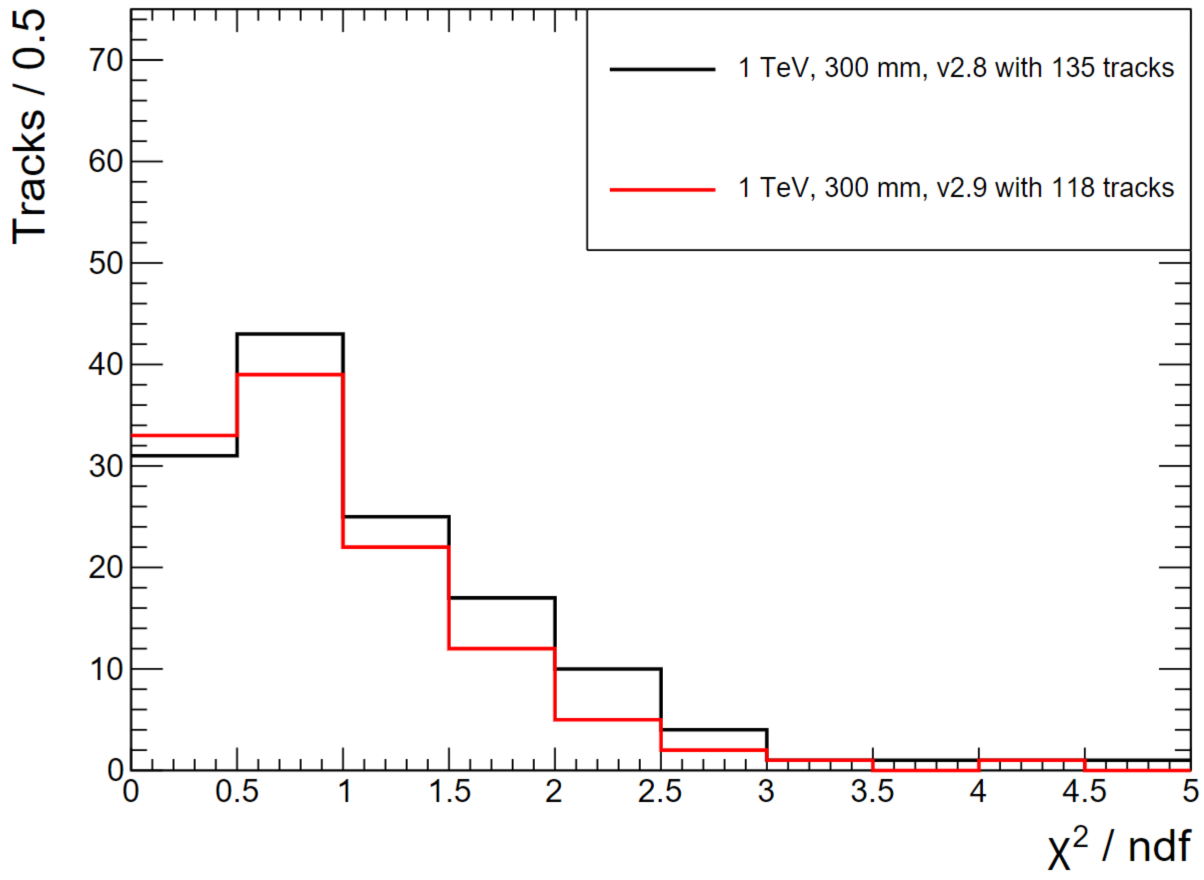


1 TeV, 1 ns

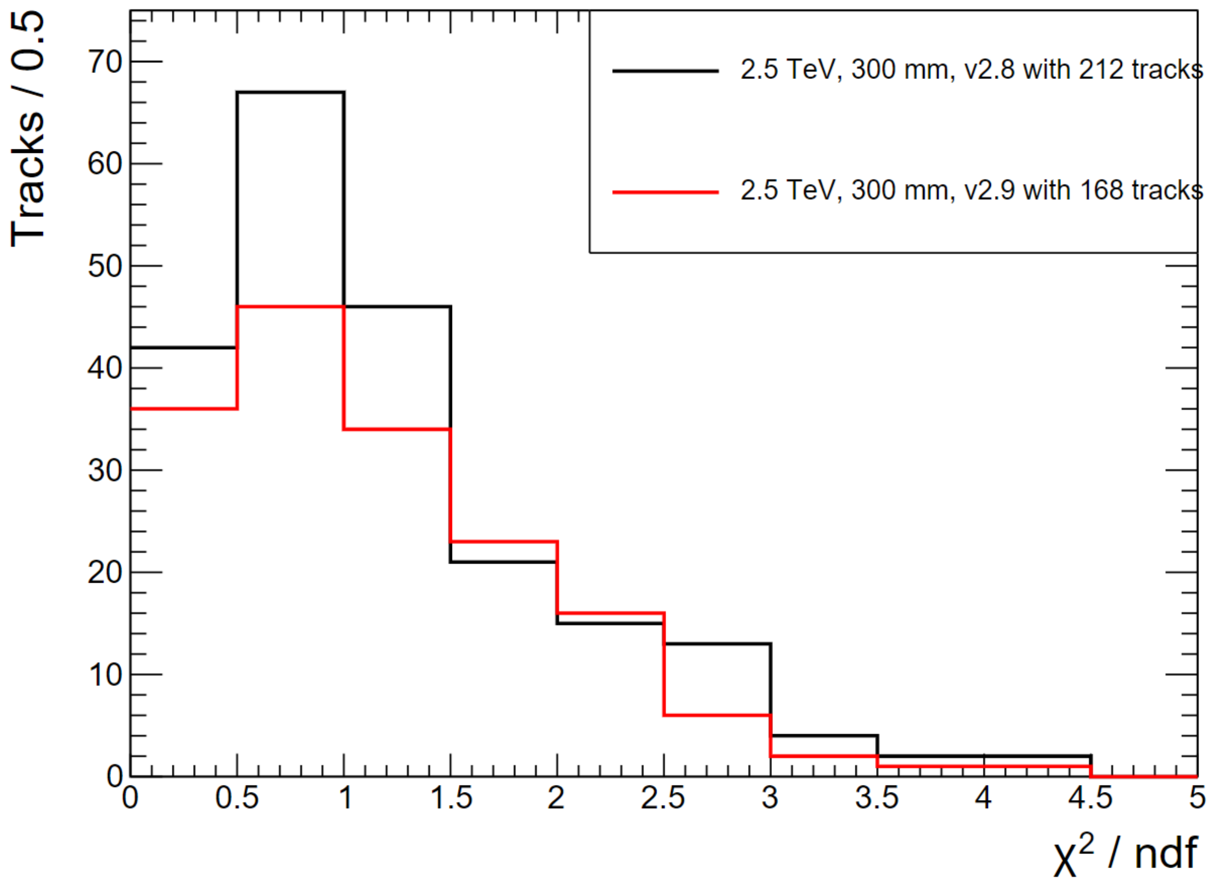


2.5 TeV, 1 ns

Displaced Track χ^2_{red}

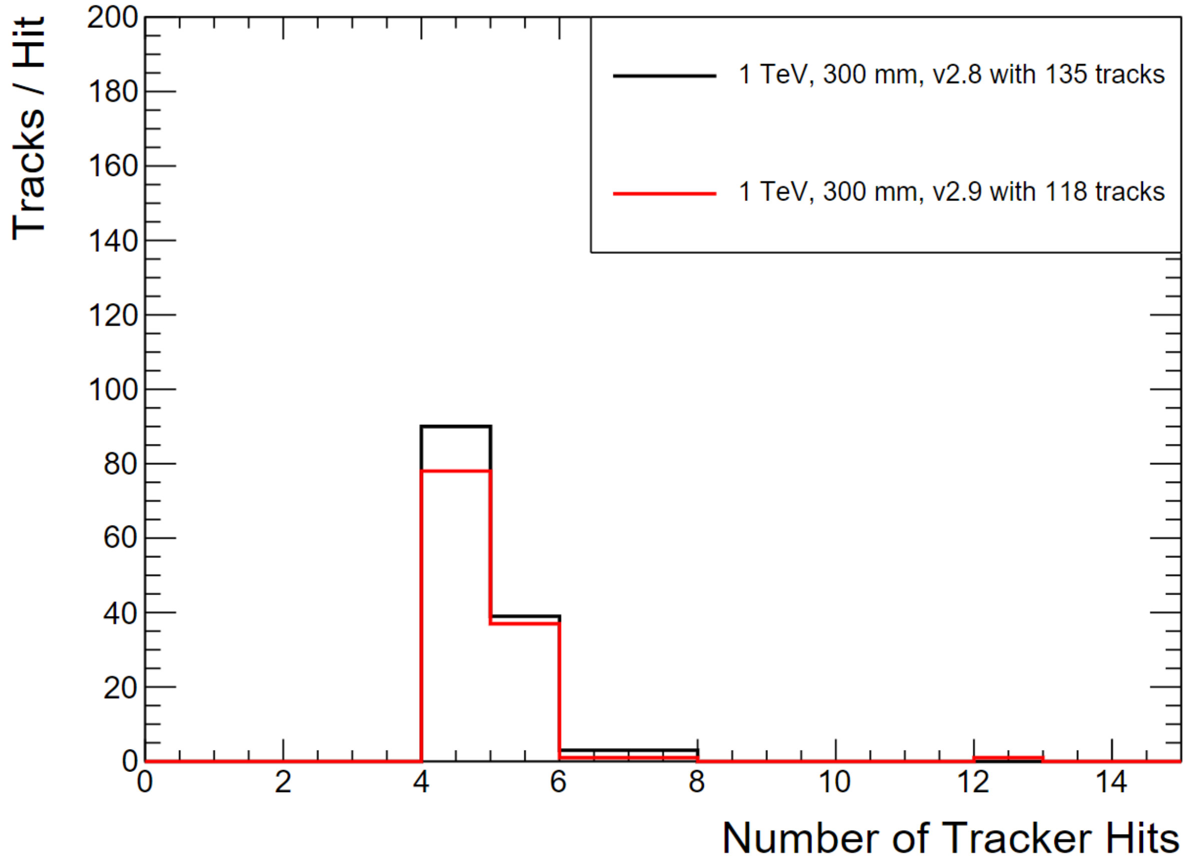


1 TeV, 1 ns

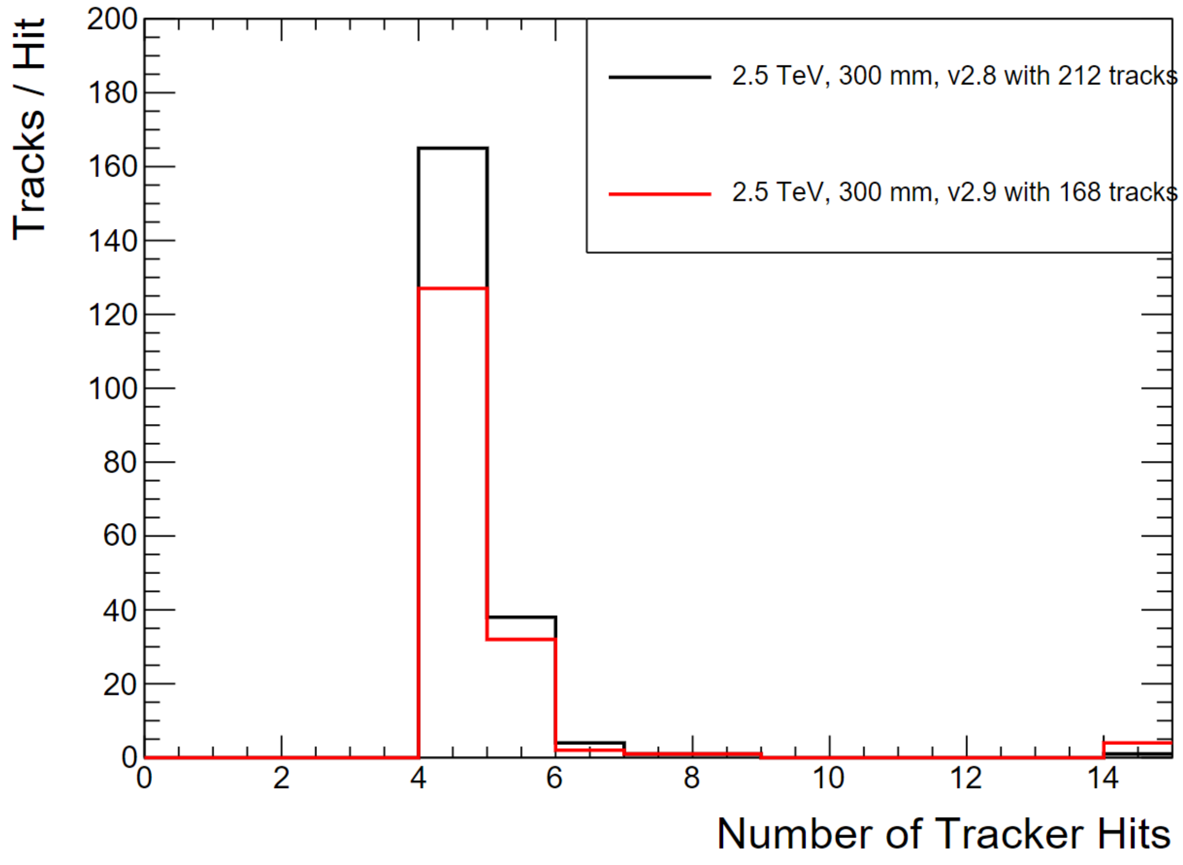


2.5 TeV, 1 ns

Displaced Track N_{hits}

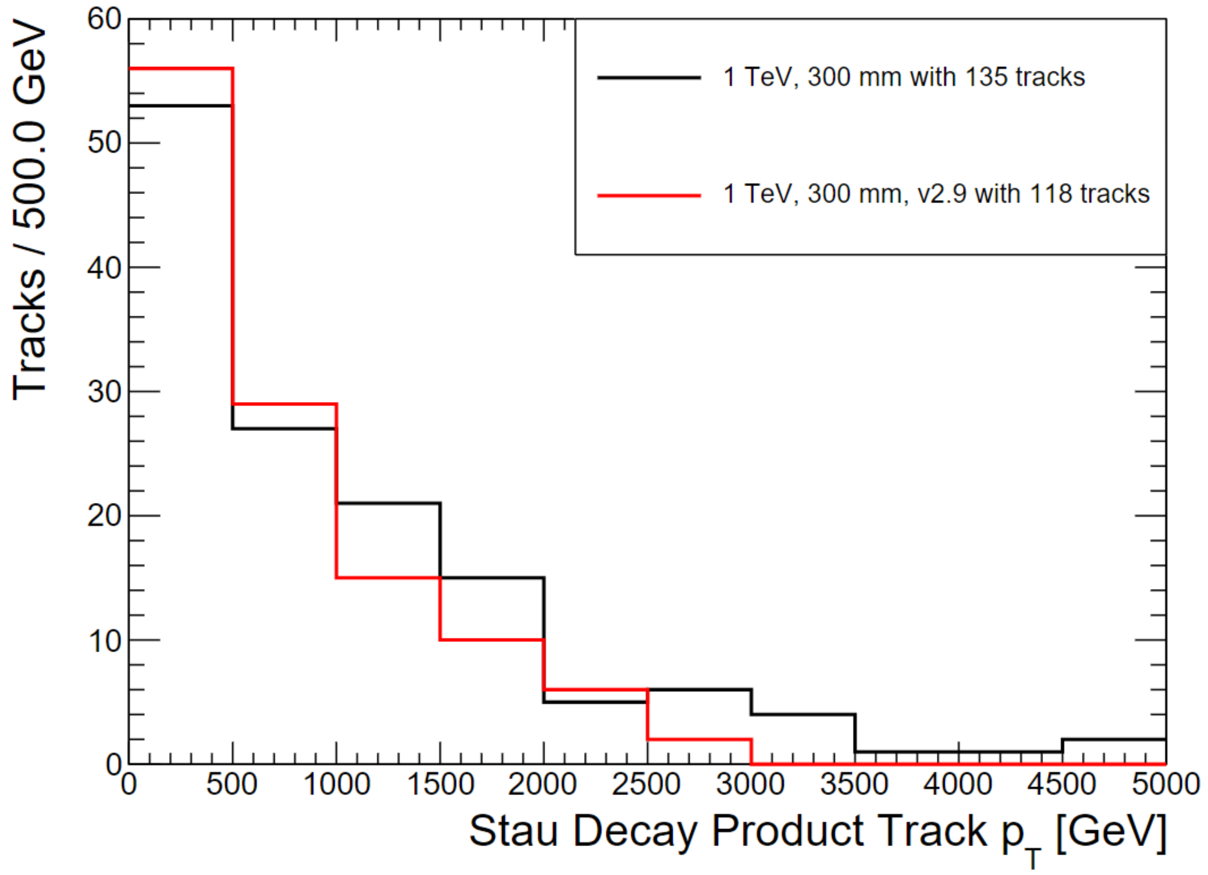


1 TeV, 1 ns

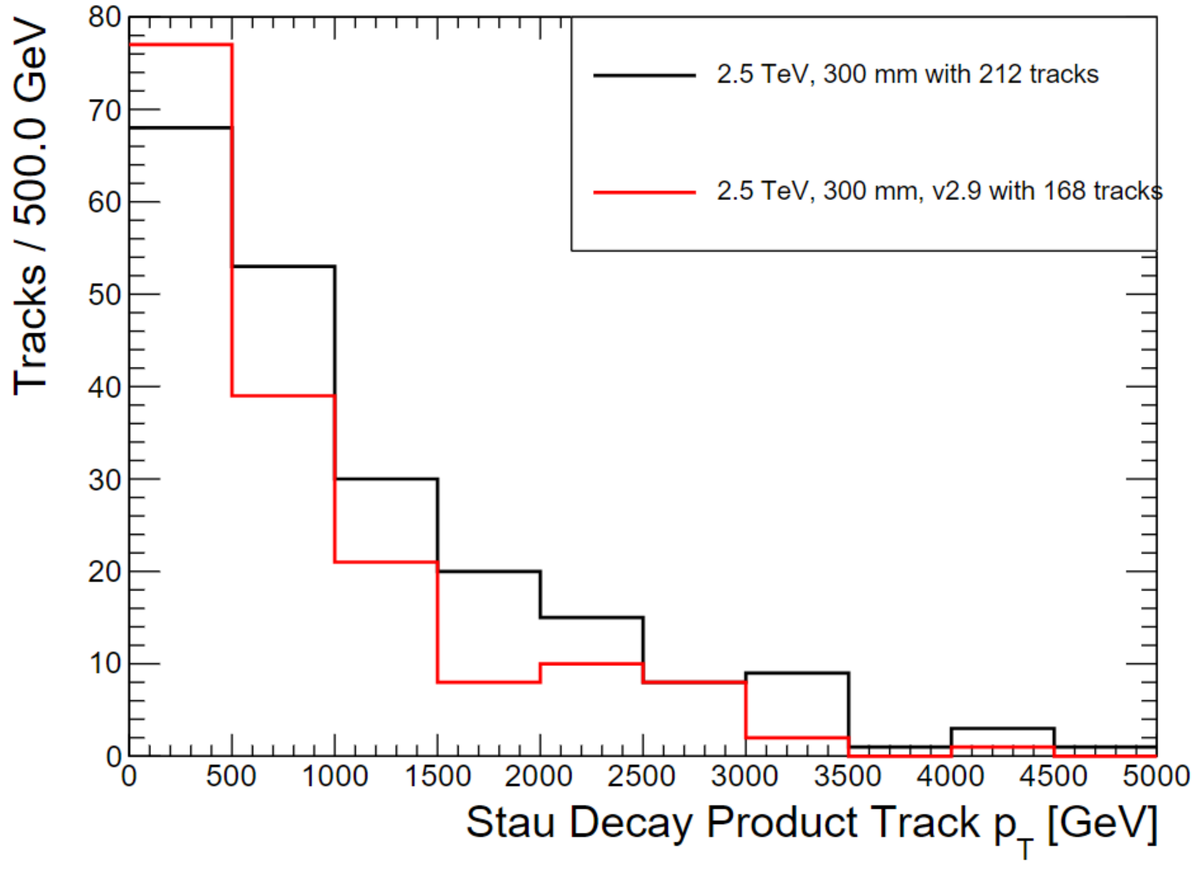


2.5 TeV, 1 ns

Displaced Track p_T

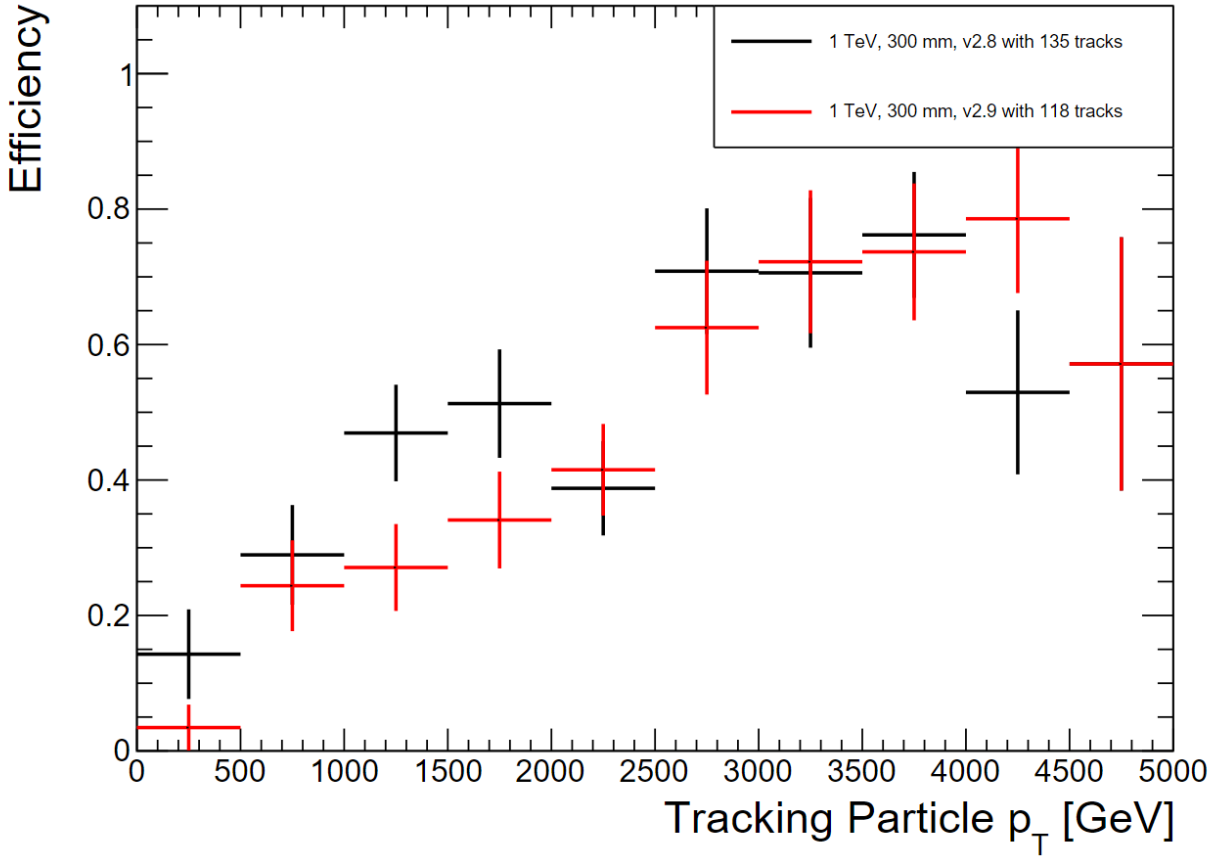


1 TeV, 1 ns

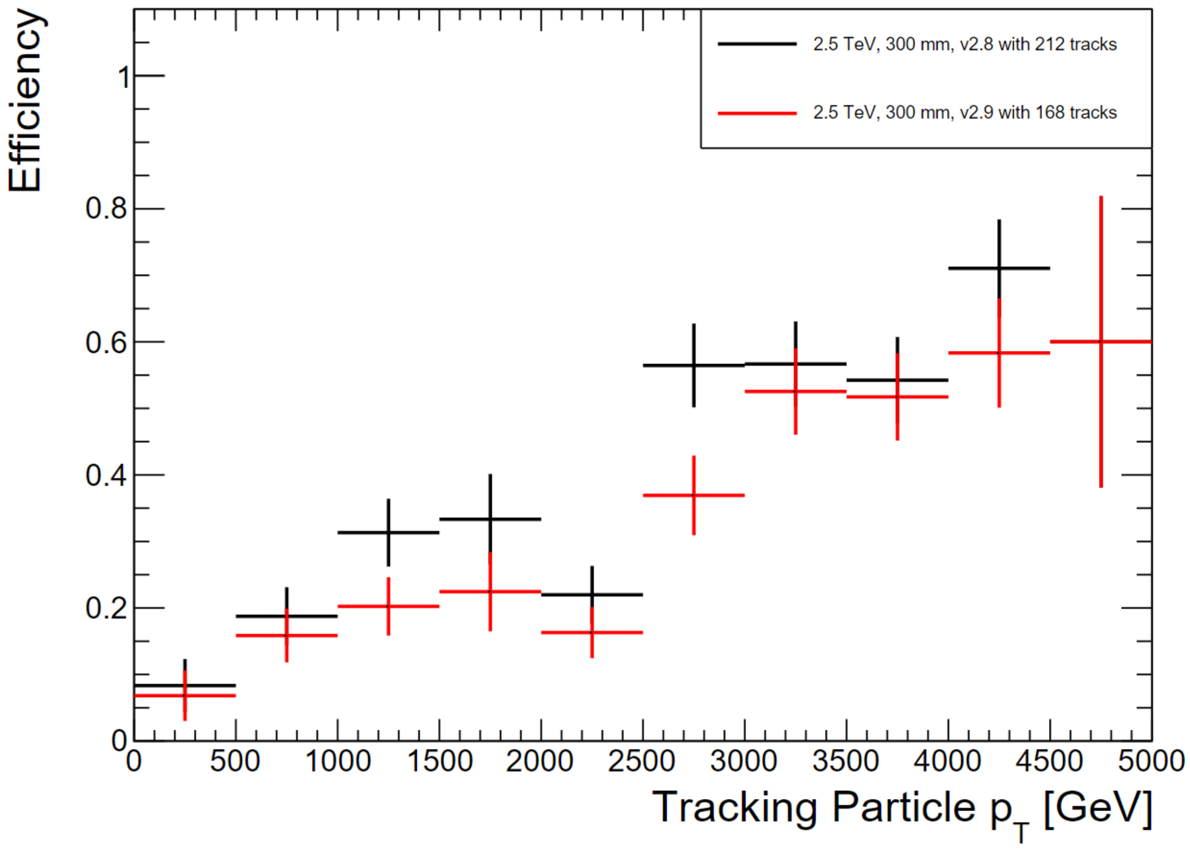


2.5 TeV, 1 ns

Displaced Tracking Efficiency vs. p_T

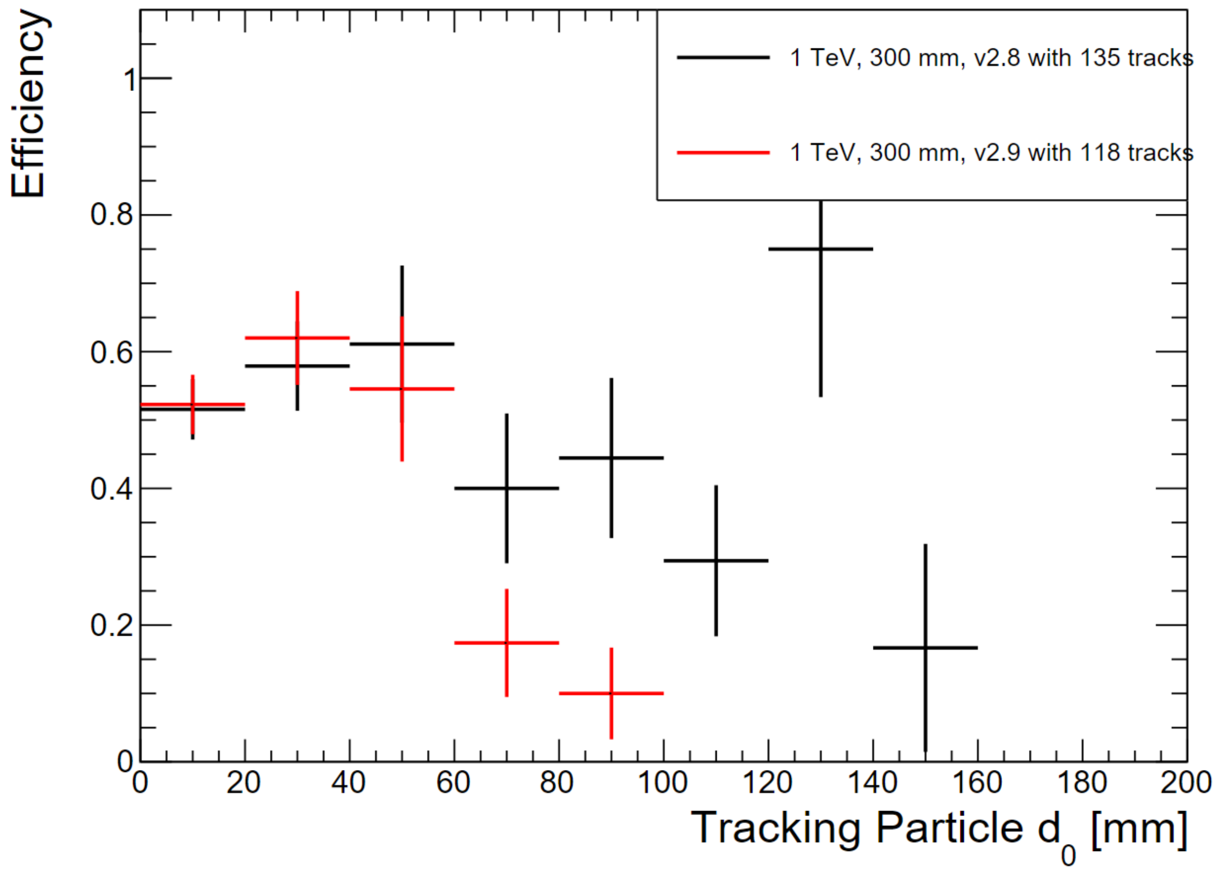


1 TeV, 1 ns

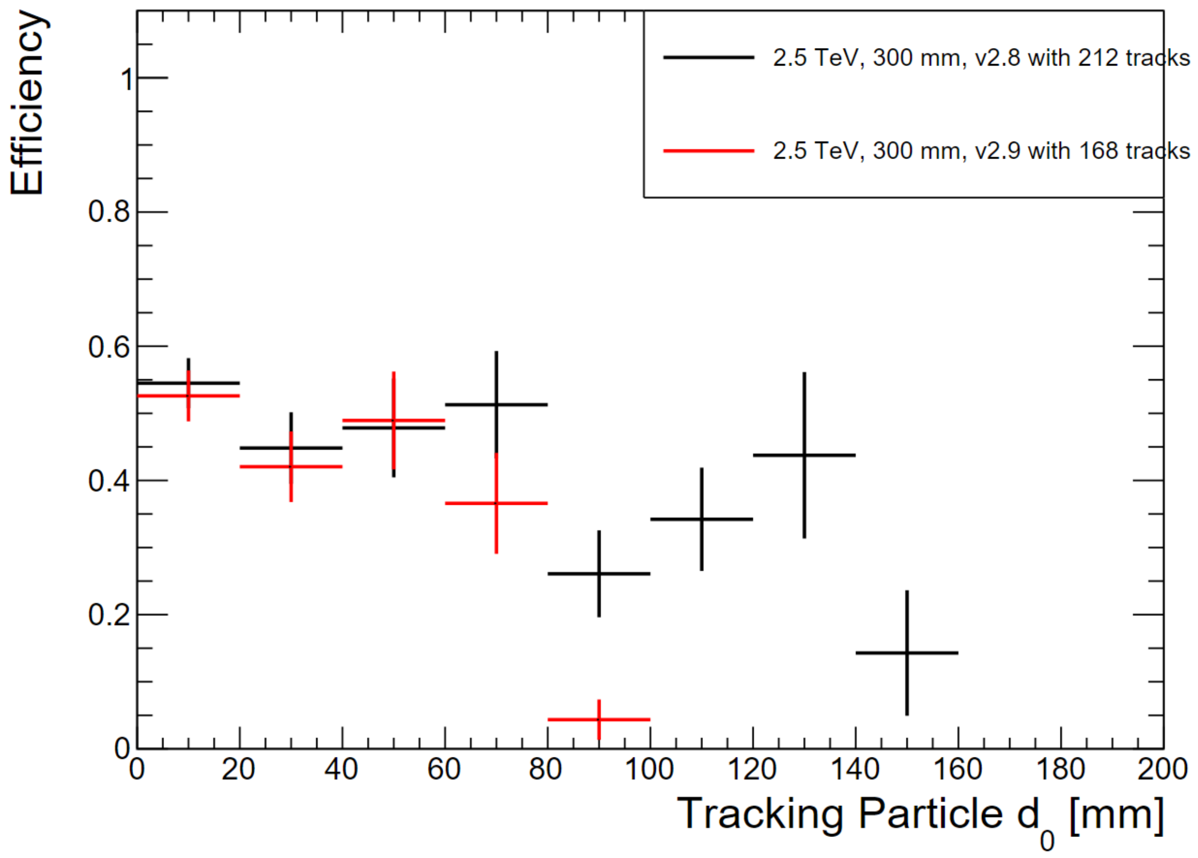


2.5 TeV, 1 ns

Displaced Tracking Efficiency vs. d_0

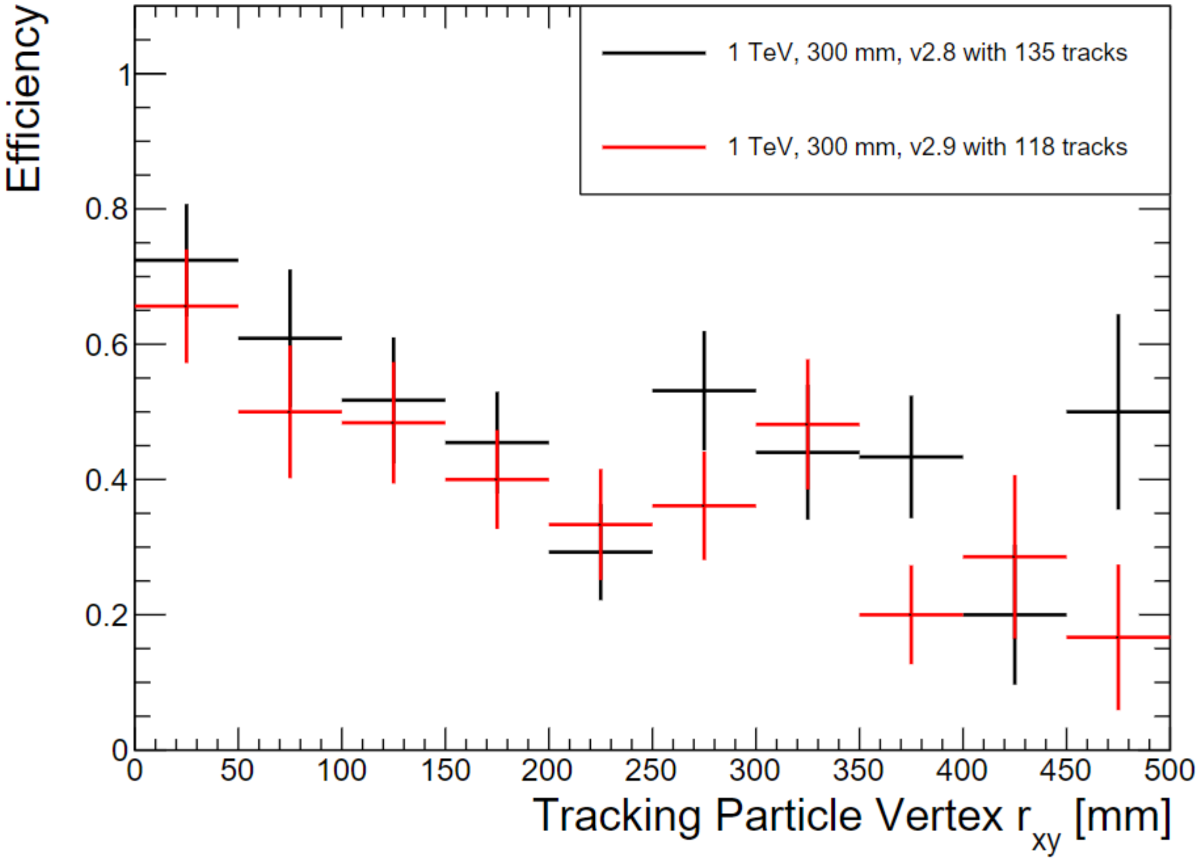


1 TeV, 1 ns

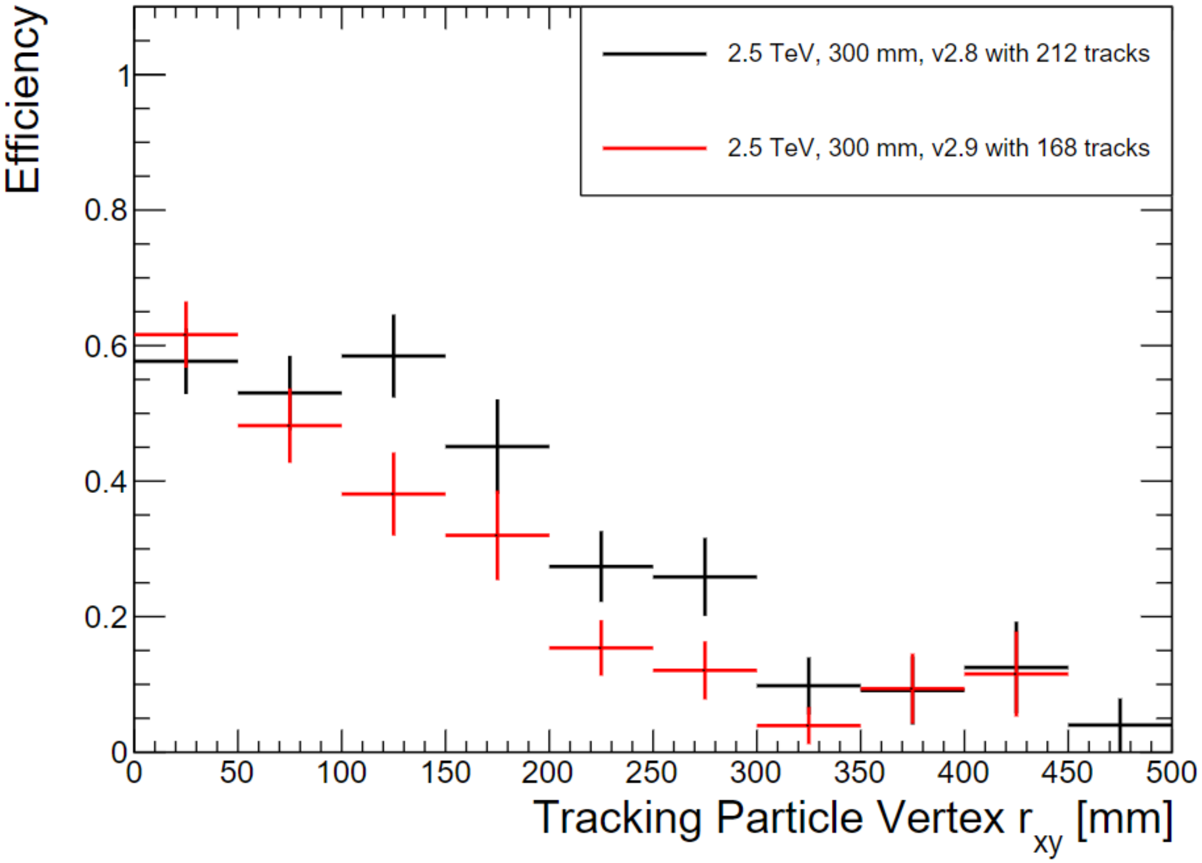


2.5 TeV, 1 ns

Displaced Tracking Efficiency vs. r_{xy}

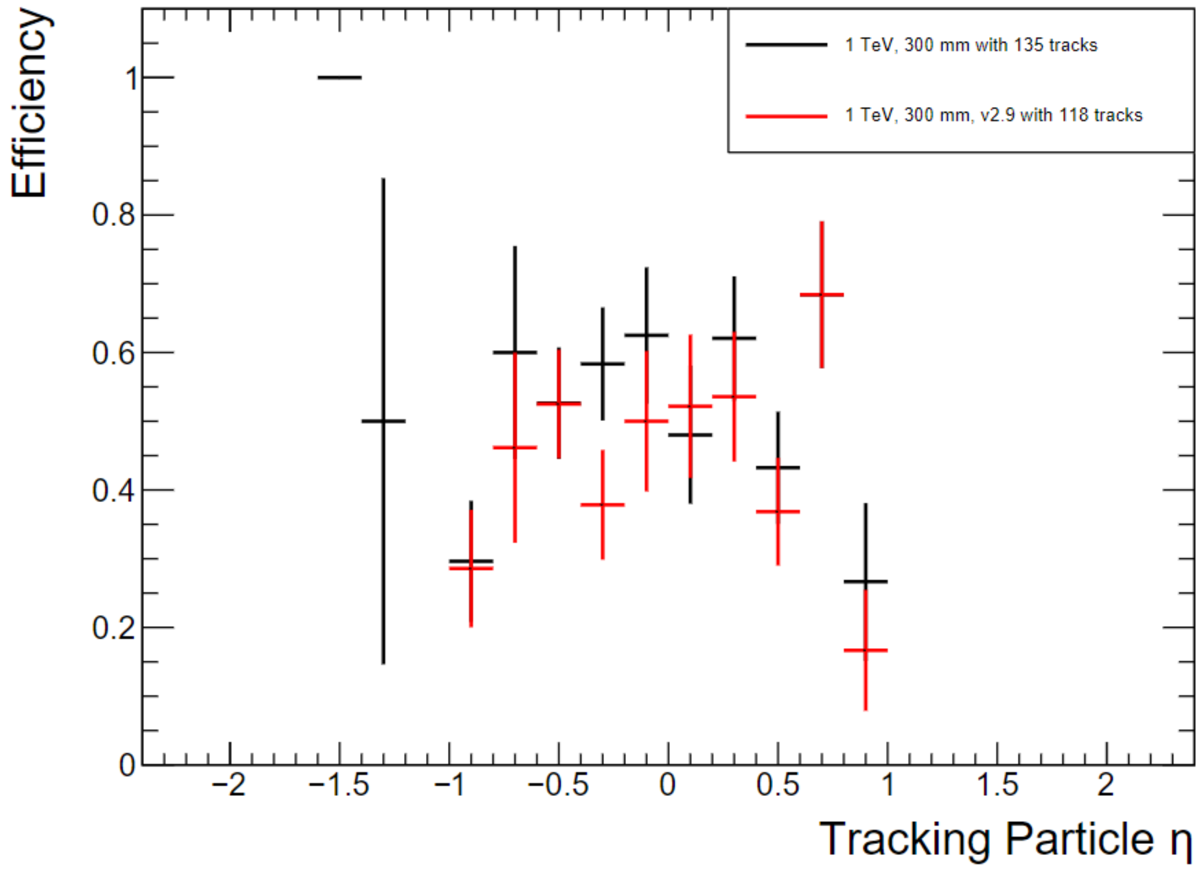


1 TeV, 1 ns

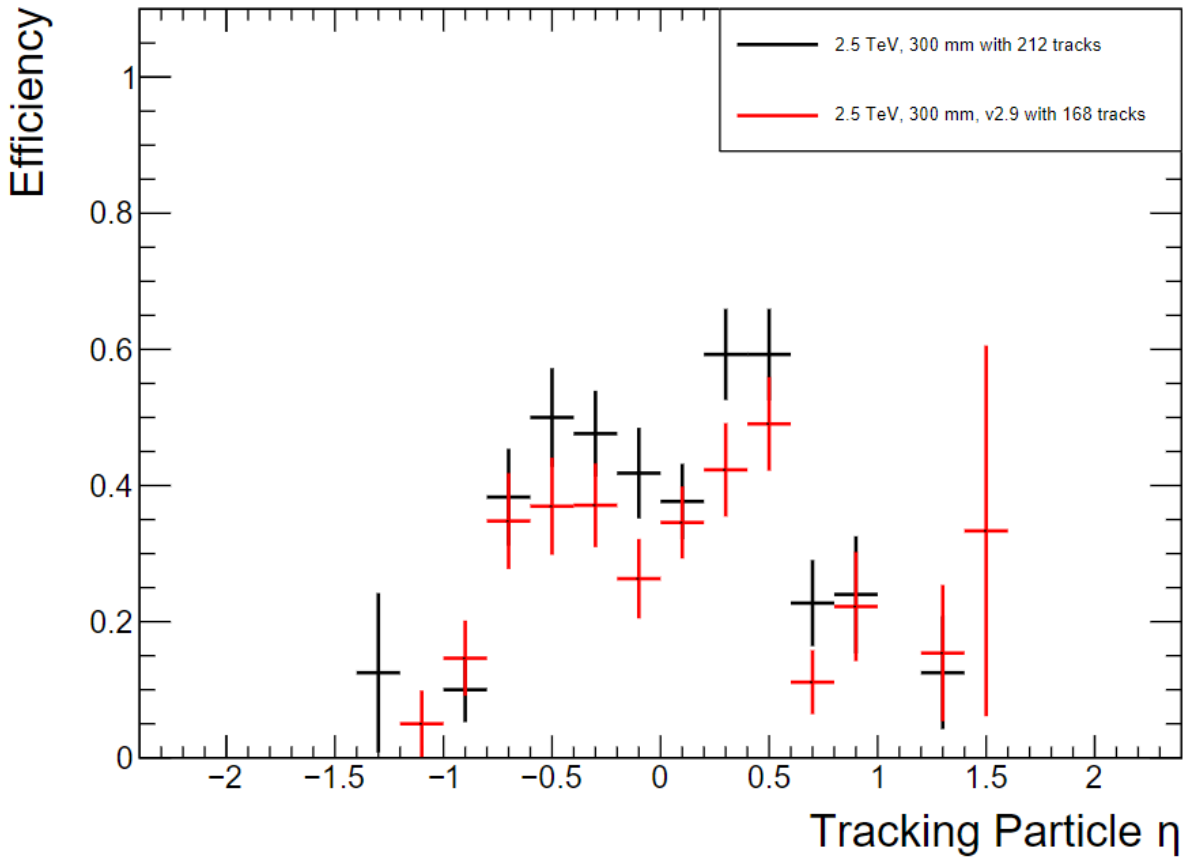


2.5 TeV, 1 ns

Displaced Tracking Efficiency vs. η



1 TeV, 1 ns



2.5 TeV, 1 ns

Prelim. BIB Results

- Using job submission setup, processed 100 event 10% BIB overlayed samples
- Tested reco. level cuts for 2nd tracking pass:
  - $\chi^2_{red} < 3,\ p_T > 1\ GeV,$ ≥ 1 IT hits & ≥ 3 OT hits

		1 TeV 1 ns	2.5 TeV 1 ns
w/o BIB	Disp. Tracking Eff.	40.5%	30.1%
	$\tilde{\tau}$ Tracking Eff.	82.1%	88.9%
	Fake Trk. / Ev.	0.178	0.048
10% BIB	Disp. Tracking Eff.	22.2%	5.7%
	$\tilde{\tau}$ Tracking Eff.	67.0%	67.8%
	Fake Trk. / Ev.	726.1	740.1
10% BIB + Cuts	Disp. Tracking Eff.	19.2%	3.3%
	$\tilde{\tau}$ Tracking Eff.	59.1%	59.4%
	Fake Trk. / Ev.	1.43	1.22

Looking for Feedback

- Could changes to ACTS in v2.9 be causing drop in displaced tracking efficiency?
  - Ideas to mitigate this?
- Suggestions for improving displaced track reconstruction in presence of BIB?
  - Trying ΔR-based matching to see if this improves matching to τ decay products
- Have observed poor p_T resolution for prompt & displaced tracks
  - Could potentially limit effectiveness of p_T as BIB rejection handle
- Link to reco. and digi. steering files used

```
MyCKFTracking_LLP.Parameters = {
    "CKF_Chi2CutOff": ["10"],
    "CKF_NumMeasurementsCutOff": ["1"],
    "MatFile": [the_args.MatFile],
    "PropagateBackward": ["False"],
    "RunCKF": ["True"],
    "SeedFinding_CollisionRegion": ["100"],
    "SeedFinding_DeltaRMax": ["350"],
    "SeedFinding_DeltaRMin": ["5"],
    "SeedFinding_ImpactMax": ["150"],
    "SeedFinding_MinPt": ["1000"],
    "SeedFinding_RMax": ["1500"],
    "SeedFinding_ZMax": ["2200"],
    "SeedFinding_RadLengthPerSeed": ["0.1"],
    "SeedFinding_SigmaScattering": ["50"],
    "SeedingLayers": [
        "23", "2", "23", "4", "23", "6", "23", "8",
        "20", "2",
        "24", "2", "24", "4", "24", "6",
        "25", "2", "25", "4", "25", "6", "25", "8",
        ],
    "TGeoFile": [the_args.TGeoFile],
    "SeedCollectionName": ["SeedTracks_LLP"],
    "TrackCollectionName": ["AllTracks_LLP"],
    "TrackerHitCollectionNames": ["SlimmedHitsCollection"]
}
```

Parameters used for 2^{nd} tracking pass

Conclusions

- Demonstrated high reconstruction efficiency for $\tilde{\tau}'s$, displaced tracks without BIB overlay using modified track reconstruction, extended timing windows
  - Efficiency for more displaced samples decreased migrating from v2.8 → v2.9 of software stack, uncertain of exact causes
  - Demonstrated complementarity of direct and indirect detection methods
- Developed new computing setup on OSG cluster to process BIB overlayed samples in parallel
- Identified reconstruction level cuts for rejecting most BIB fake tracks while accepting most displaced tracks
  - Found significantly reduced displaced tracking efficiency in presence of BIB
  - Potential for tighter analysis level cuts to reject most of remaining BIB fake tracks

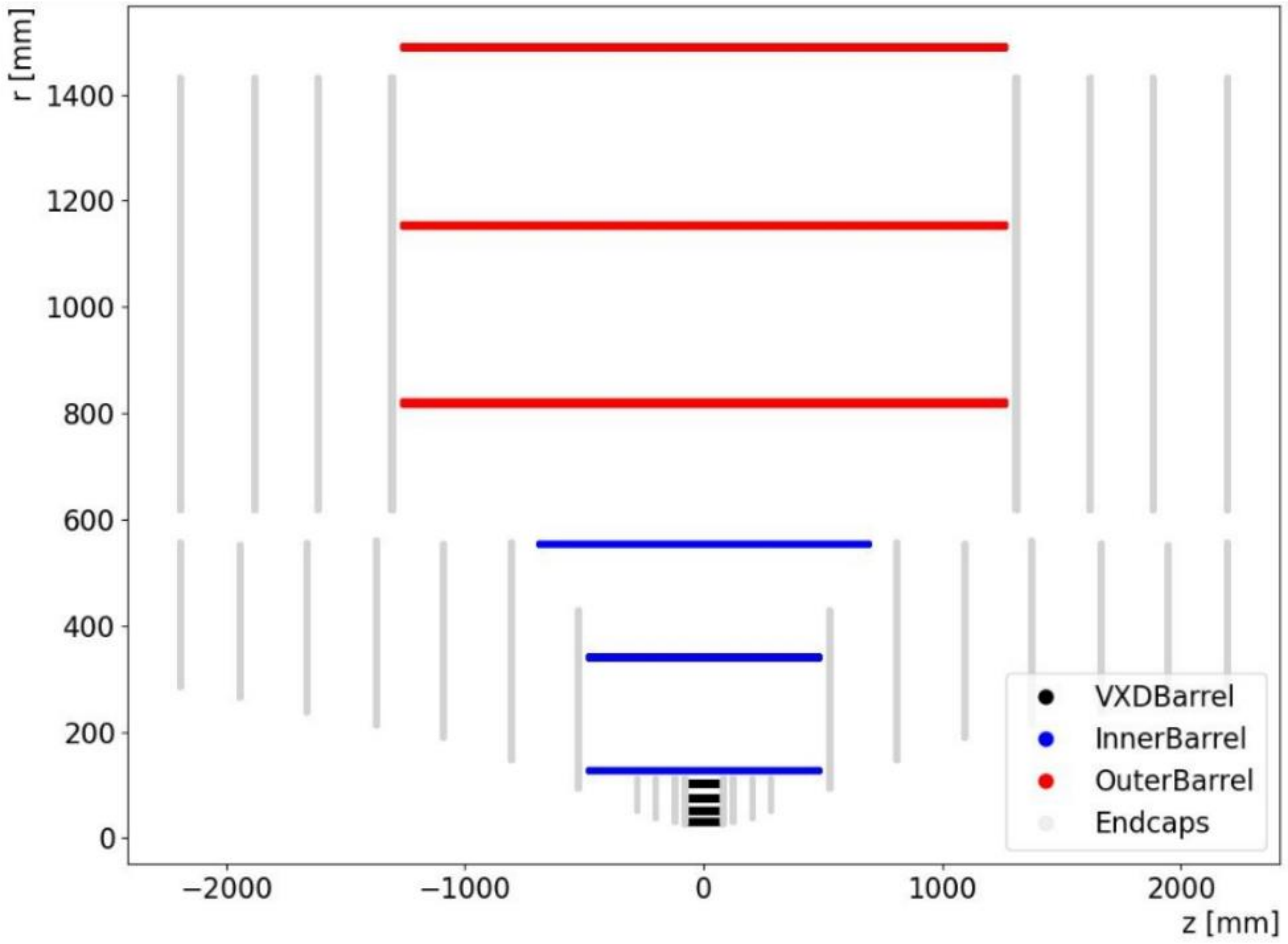
Future Work

- Make any modifications to track reconstruction algo. to mitigate decreases in tracking efficiency after version change & BIB overlay
- Producing BIB overlayed results for tight, medium, loose timing windows
  - Continue optimizing reco. level cuts
- Will later give another presentation detailing our physics results:
  - **Prompt $\tilde{\tau}'s$:** extract time-of-flight, mass information
  - **Displaced tracks:** indicate any efficiency with strong BIB rejection, report on any additional computing challenges

Backup

Tracker Geometry

- Barrel tracker consists of:
  - Pixel Vertex Detector with 4 doublet layers
  - 3 Inner Tracker layers
  - 3 Outer Tracker layers
- Provides coverage for displaced tracks up to $R_{xy} \approx 550$ mm for central processes

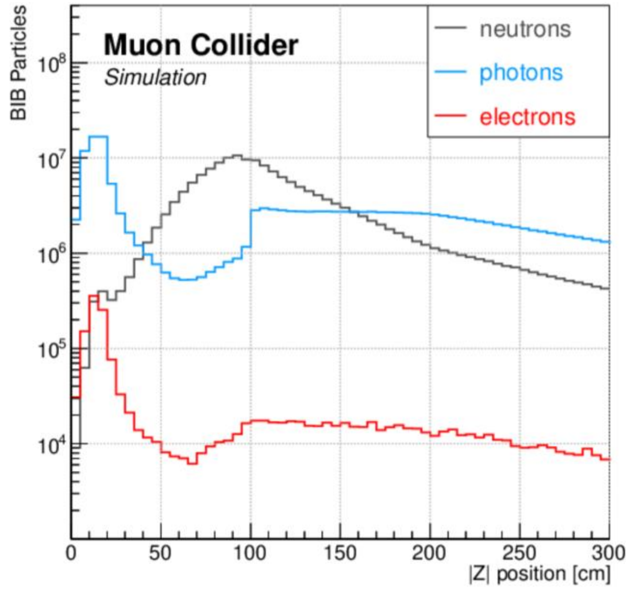


BIB Challenges

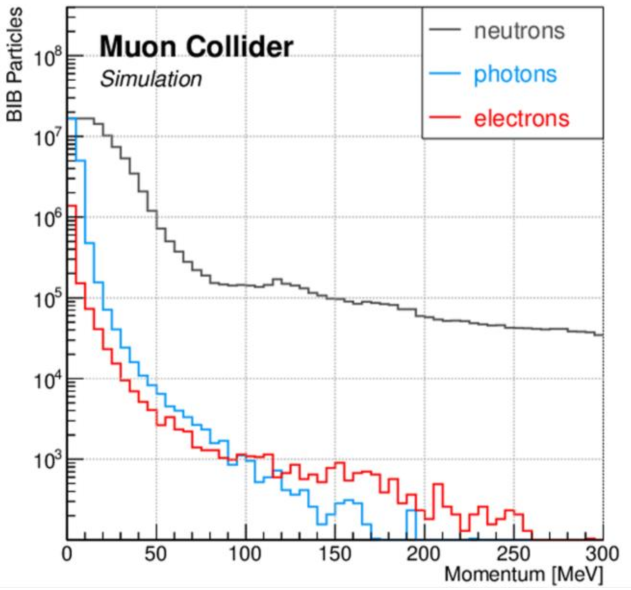
- Presence of BIB results in an average of 500k hits in innermost tracker layer per event, this necessitates using powerful rejection handles to prevent formation of fake tracks

BIB Properties:

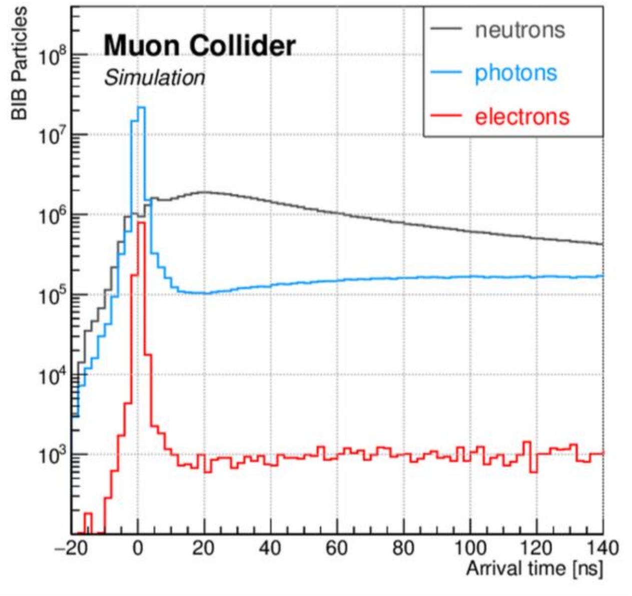
Arrives through side of detector



BIB particles have low momentum



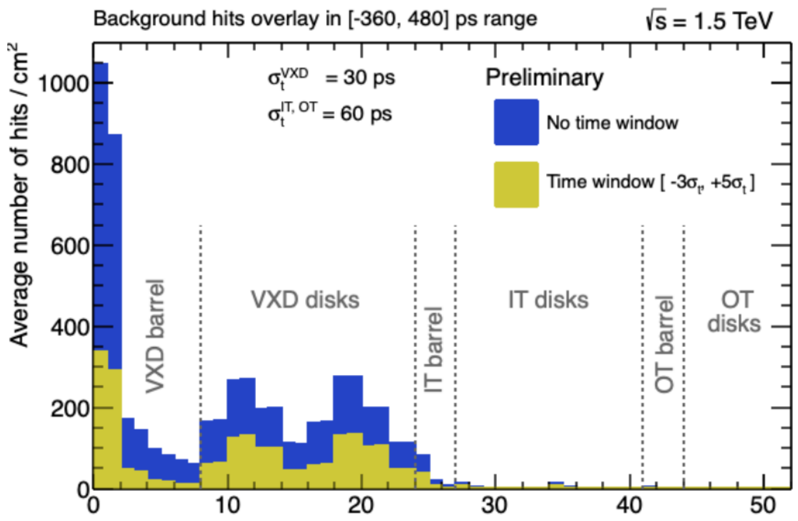
BIB particles arrive out of time w.r.t. bunch crossing



- Additionally, BIB tracks formed will often be formed with low number of hits, poor fit quality, low p_T

BIB Rejection & Displaced Tracking

- Pointing requirement, track p_T requirements, timing hit acceptance windows, and number of hits to form track requirements all provide powerful BIB rejection
  - Concern is that BIB rejection handles – especially pointing requirement, cut on number of hits, and timing windows – may also reject long-lived particle signatures



Effect on hit density when applying nominal timing windows

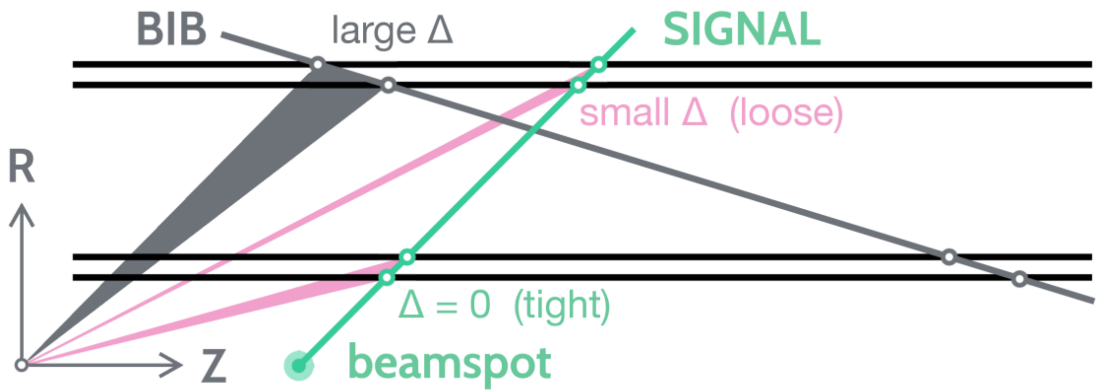
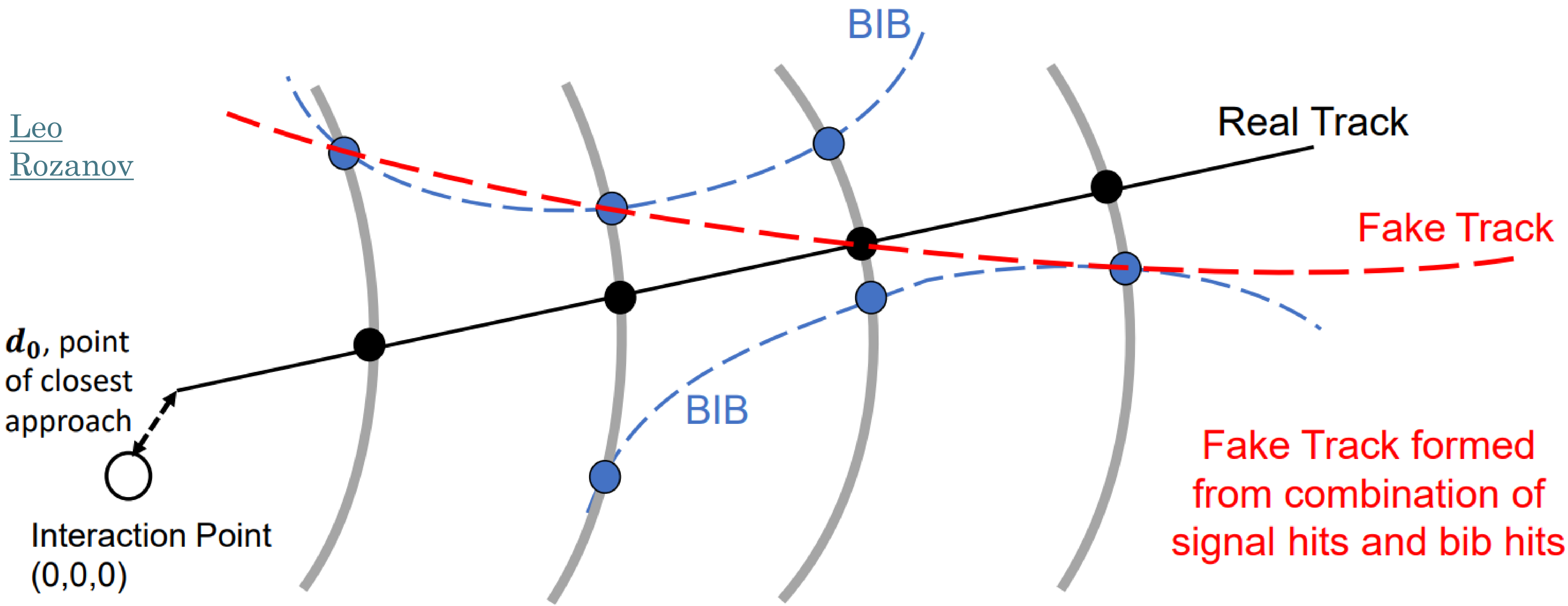


Illustration of how pointing requirement (using doublet layers) rejects BIB

Fake Track Formation

[Leo Rozanov]



Computing Setup

- Another challenge introduced by BIB is large file size and long run time when processing BIB events
  - Using 100% BIB yields file sizes of 1 GB / event, reconstruction time of multiple hours per event when keeping / using only tracker information!
  - Necessitates use of running processes in parallel
- To deal with these challenges, migrated setup from local workstation to larger computing cluster, allowing submission of condor jobs
  - Greatly reduces computing time, allowing full study of displaced tracking with signal in presence of 100% BIB

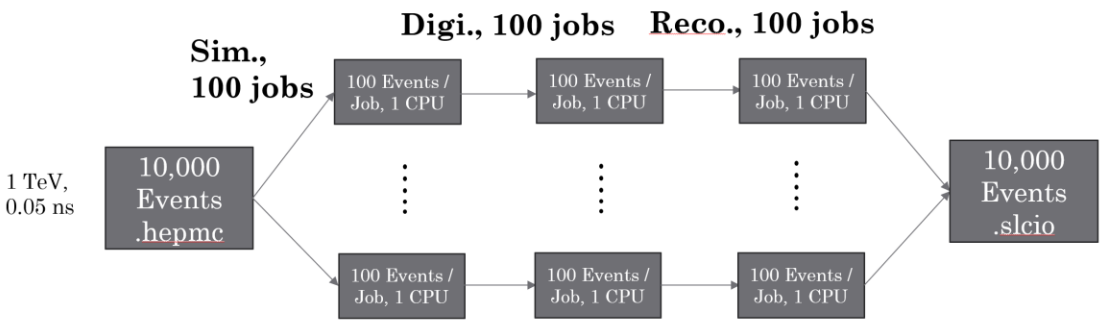
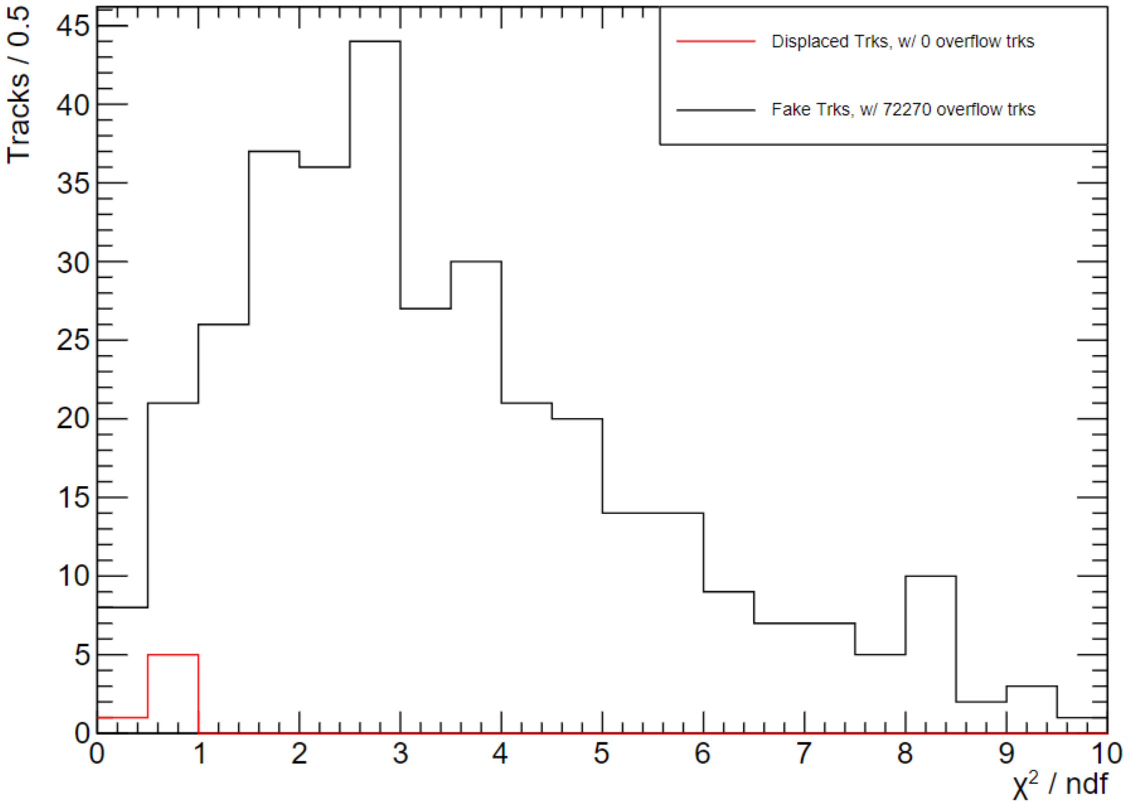


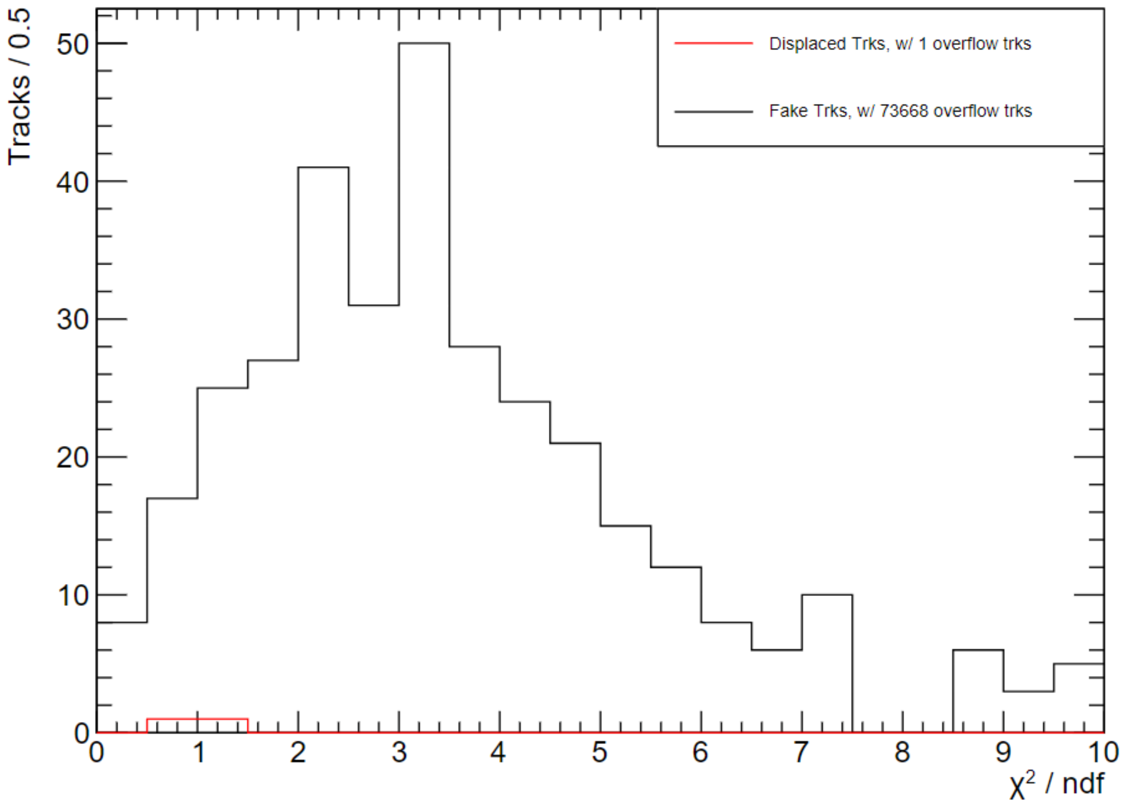
Illustration of job submission for one sample

Track χ^2_{red}(10% bib)

1 TeV 1 ns, 100 10% BIB events



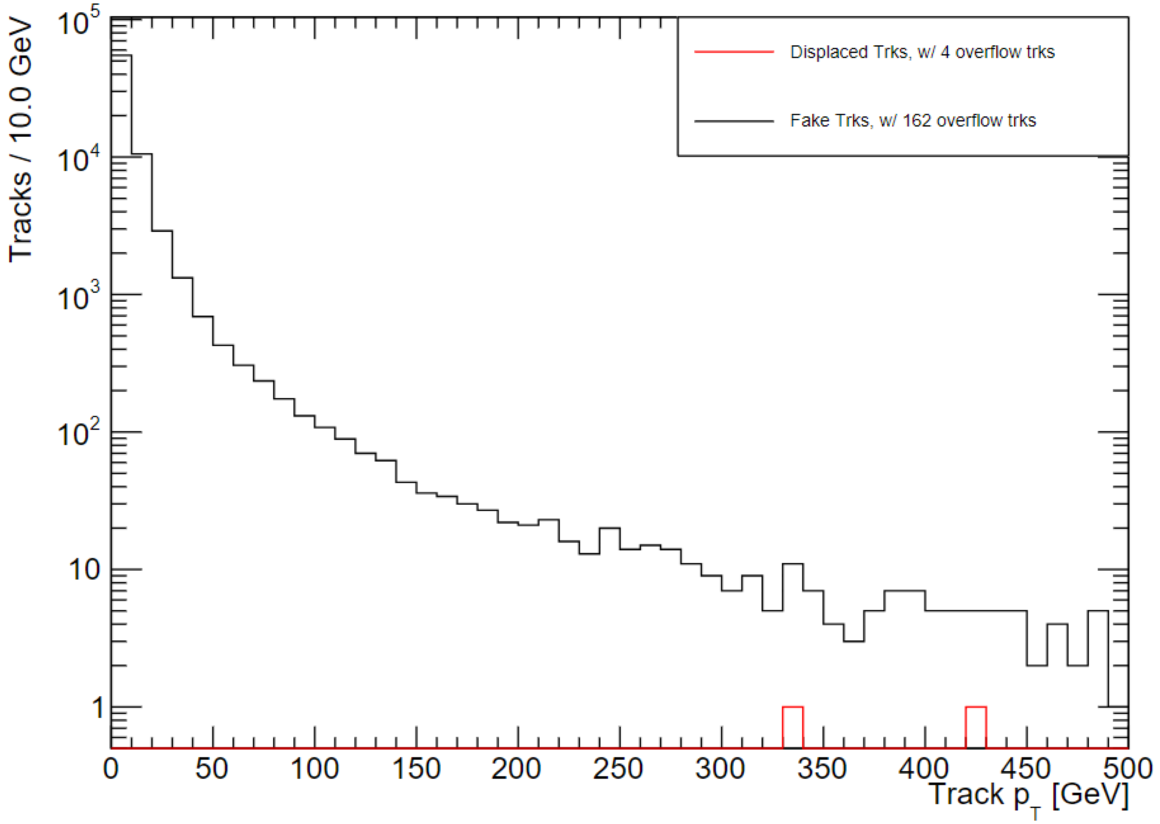
2.5 TeV 1 ns, 100 10% BIB events



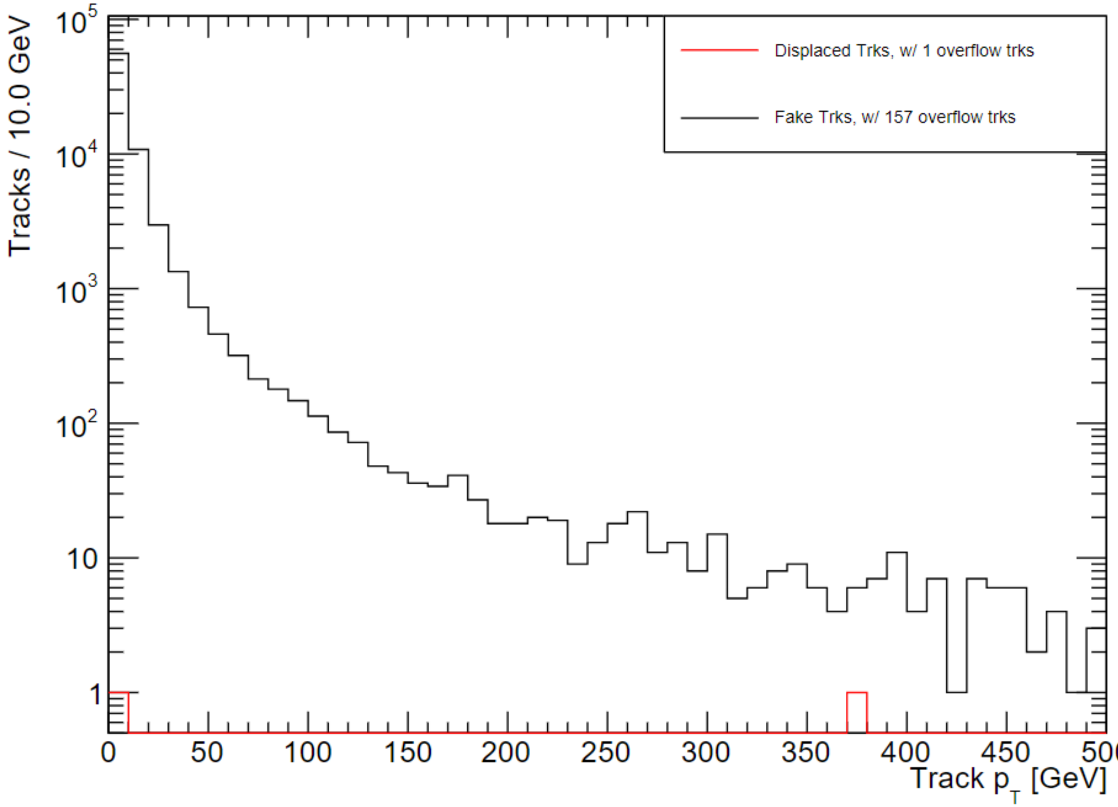
χ^2_{red} would provide most powerful rejection handle!

Track p_T (10% bib)

1 TeV 1 ns, 100 10% BIB events



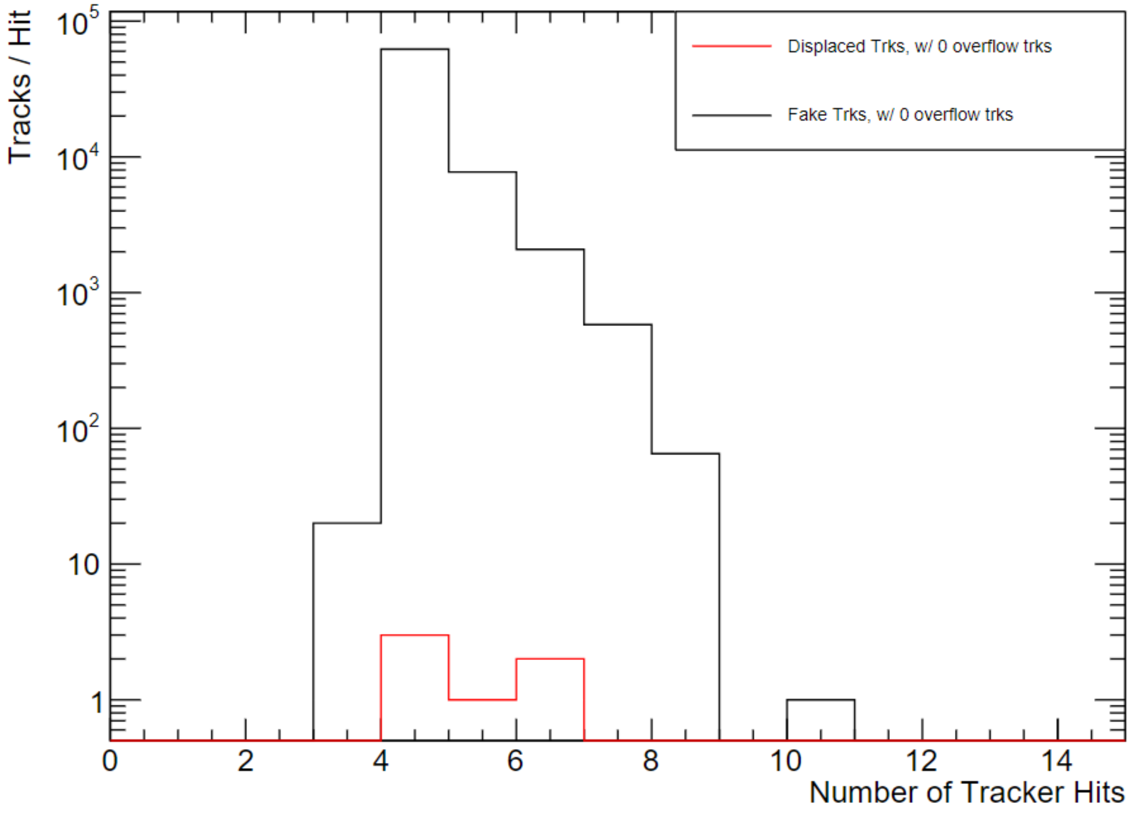
2.5 TeV 1 ns, 100 10% BIB events



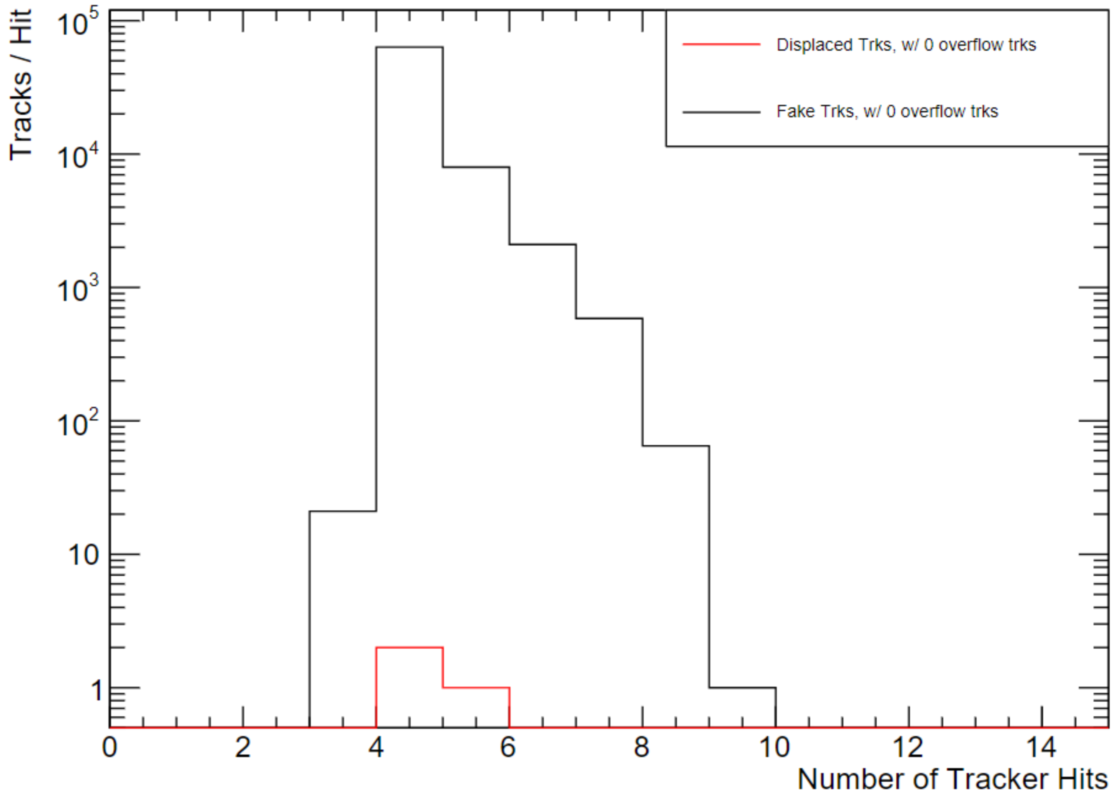
p_T would provide powerful secondary rejection handle, but should understand calculation of track p_T…

Track N_{hits}(10% bib)

1 TeV 1 ns, 100 10% BIB events



2.5 TeV 1 ns, 100 10% BIB events



N_{hits} cut likely should be left as ≥ 4 (need to fix for fake tracks in analysis script)