

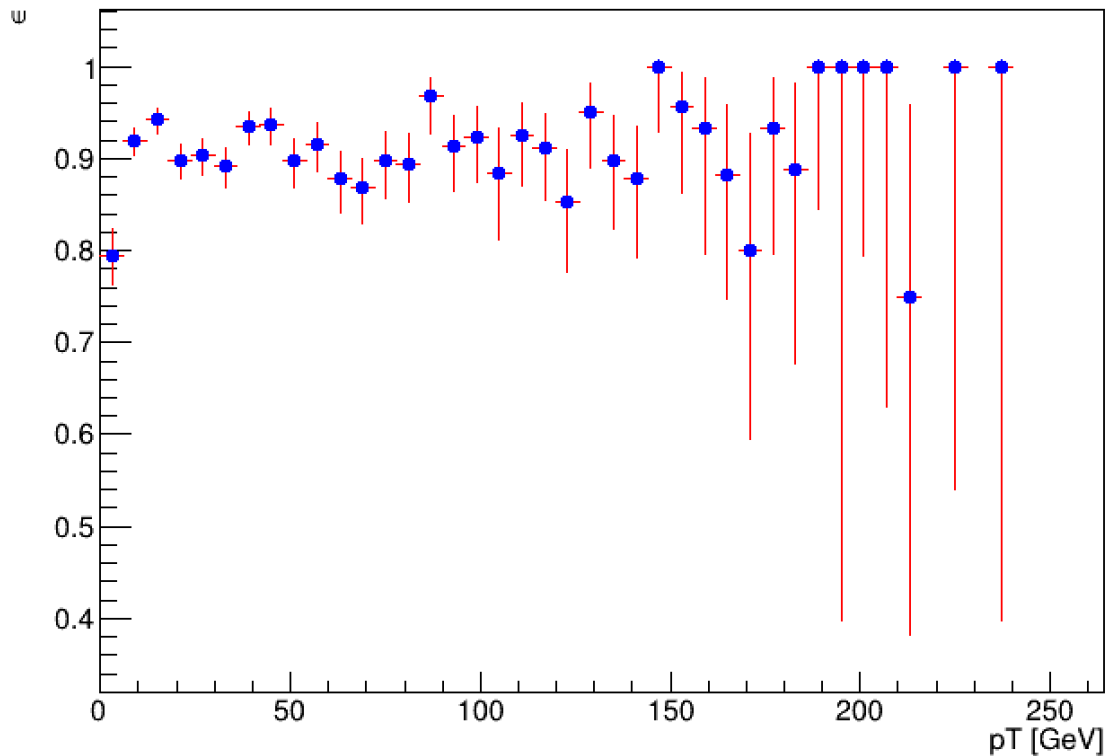
Trk-Cluster Matching Updates (10/19)

- Tracking issues resolved with the new sample
- Digging into the Pandora TrackClusterAssociation algorithm
- Examined cuts that may be causing our problem

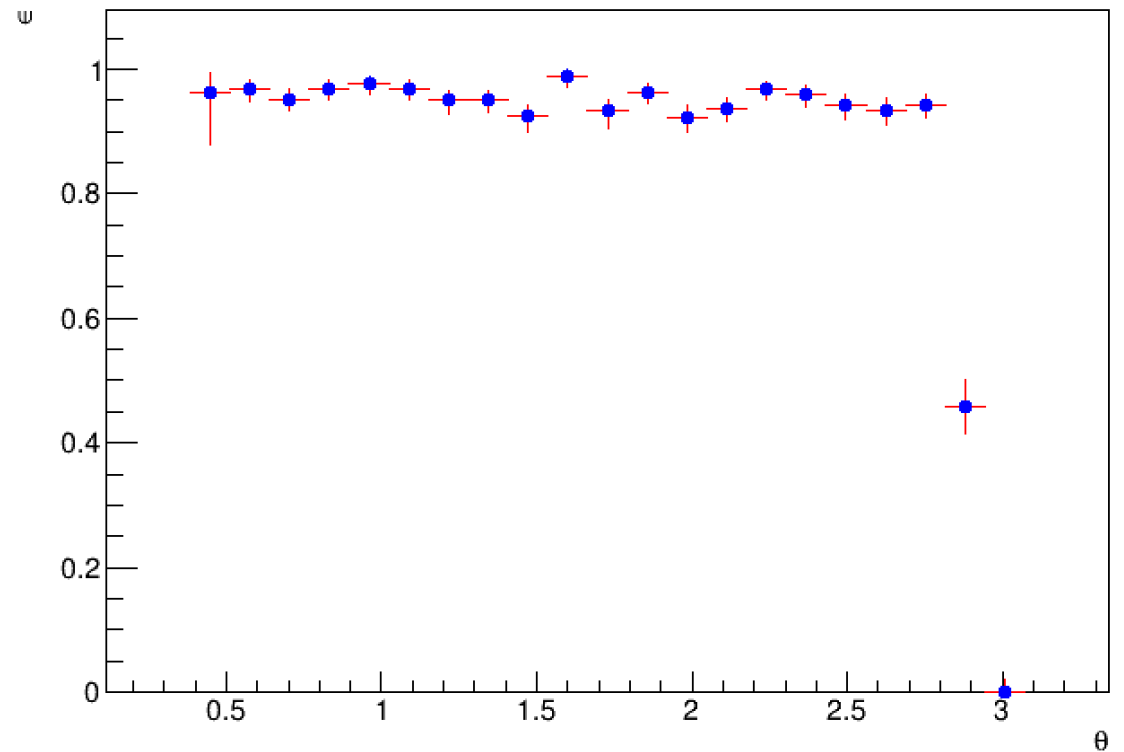
Tracking is back up!

- With the resolution of the tau decay length and a wider Z spread in CKFT settings, charged pion tracking is looking good:

Tracking Efficiency, pions (tau gun)



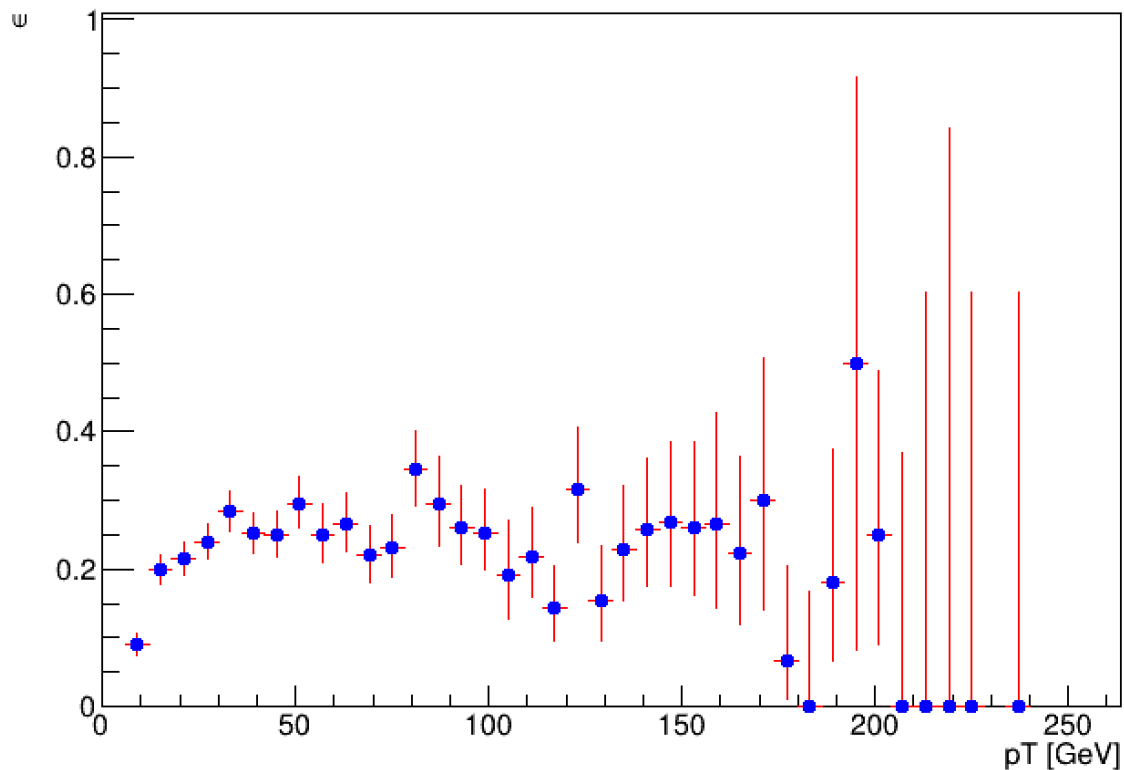
Tracking Efficiency, pions (tau gun)



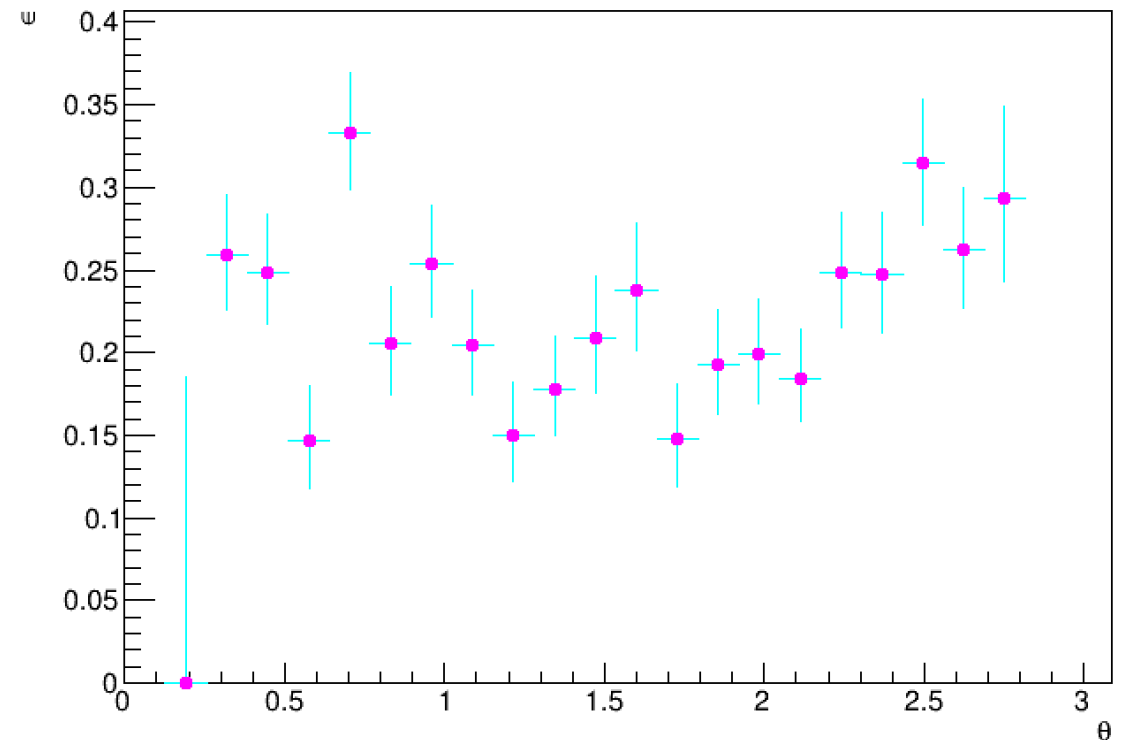
Now, back to the PFO problem

- Our PFO efficiency is abysmal for charged pions – as before, track-cluster matching seems to be the culprit

PFO Efficiency, π^+/π^- (tau gun)



PFO Efficiency, π^+/π^- (tau gun)

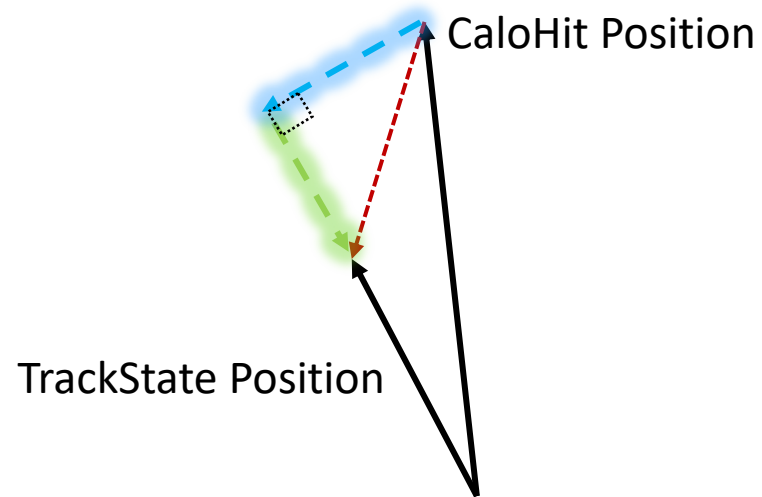


Pandora TrackClusterAssociationAlgorithm

- <https://github.com/PandoraPFA/LCContent/blob/master/src/LCTrackClusterAssociation/TrackClusterAssociationAlgorithm.cc>
- This is where we associate tracks to clusters
- Matching is done geometrically, via a helper function called GetTrackClusterDistance defined here:
<https://github.com/PandoraPFA/LCContent/blob/aabb475ab87a538826befb36e79e27bc977e321f/src/LCHelpers/ClusterHelper.cc>

GetTrackClusterDistance

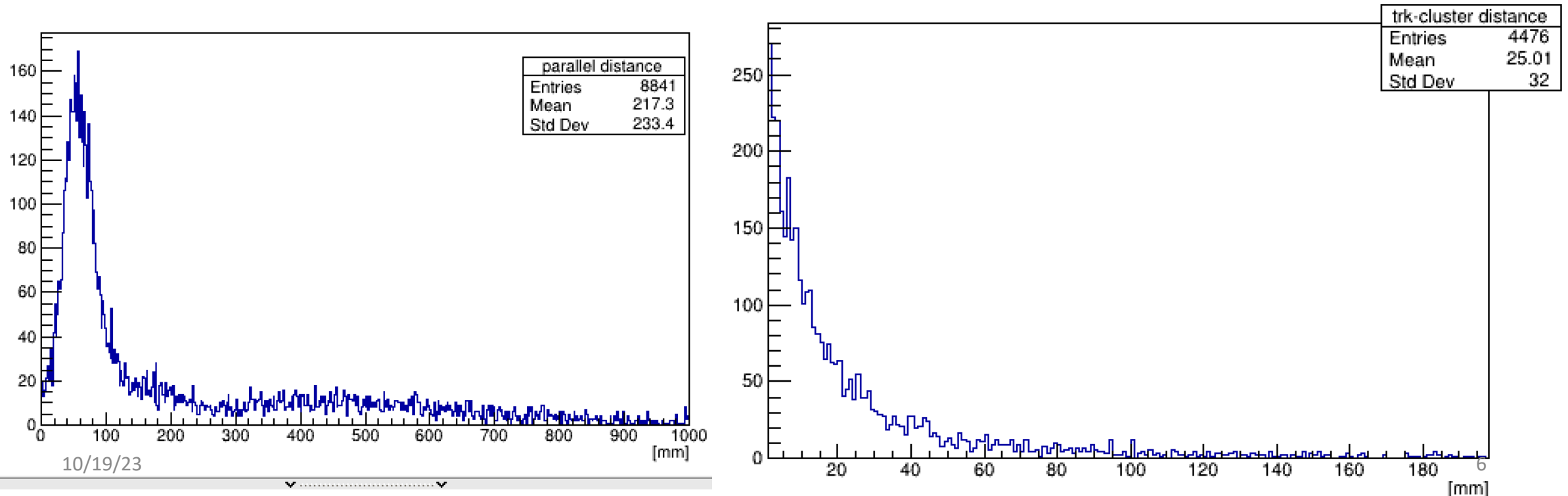
- For each calohit in the cluster, assess the distance from the track state
- Just geometric:



- First, we cut on **parallel distance** (max 100 mm)
- Then we cut on **perpendicular distance** (max 10 mm)

Parallel/Perpendicular Distances in Sample

- I replicated this method of finding track-cluster distance for the taugun sample and found the following distributions:
- (I chose the minimum distance in events with more than one track or cluster for perpendicular but not for parallel, which is why we have more entries for parallel)
- Parallel distance peaking a bit before it should be given the depth of the HCAL – will take a more careful look and compare with a charged pion gun sample



Conclusions...

- We might be losing some track/cluster pairs with the parallel distance cut, but I am more concerned about the aggressive **perpendicular** cut
- Perhaps we could try to generate a sample (doesn't have to be a full sample, just a few thousand events) with the cut increased to see if we get different results
- Can just insert the following line in PandoraSettingsDefault.xml to specify the cut
- By integrating the distribution below, I will find the perpendicular cut that would theoretically correspond to a 9-% efficiency level and we can try that value
- Also, once we establish that cut, we should start running over the BIB to see if raising the cut dramatically increases fake rate

```
<!-- Prepare particle flow objects -->
<algorithm type = "TrackPreparation">
  <CandidateListNames>Input</CandidateListNames>
  <MergedCandidateListName>PfoCandidates</MergedCandidateListName>
  <PfoTrackListName>PfoCreation</PfoTrackListName>
  <trackClusterAssociationAlgorithms>
    <algorithm type = "TrackClusterAssociation"/>
    <!-- Set cut on perpendicular distance - raise cut? -->
    <MaxTrackClusterDistance> 10. </MaxTrackClusterDistance>
    <algorithm type = "LoopingTrackAssociation"/>
    <algorithm type = "TrackRecovery"/>
    <algorithm type = "TrackRecoveryHelix"/>
    <algorithm type = "TrackRecoveryInteractions"/>
  </trackClusterAssociationAlgorithms>
</algorithm>
```

