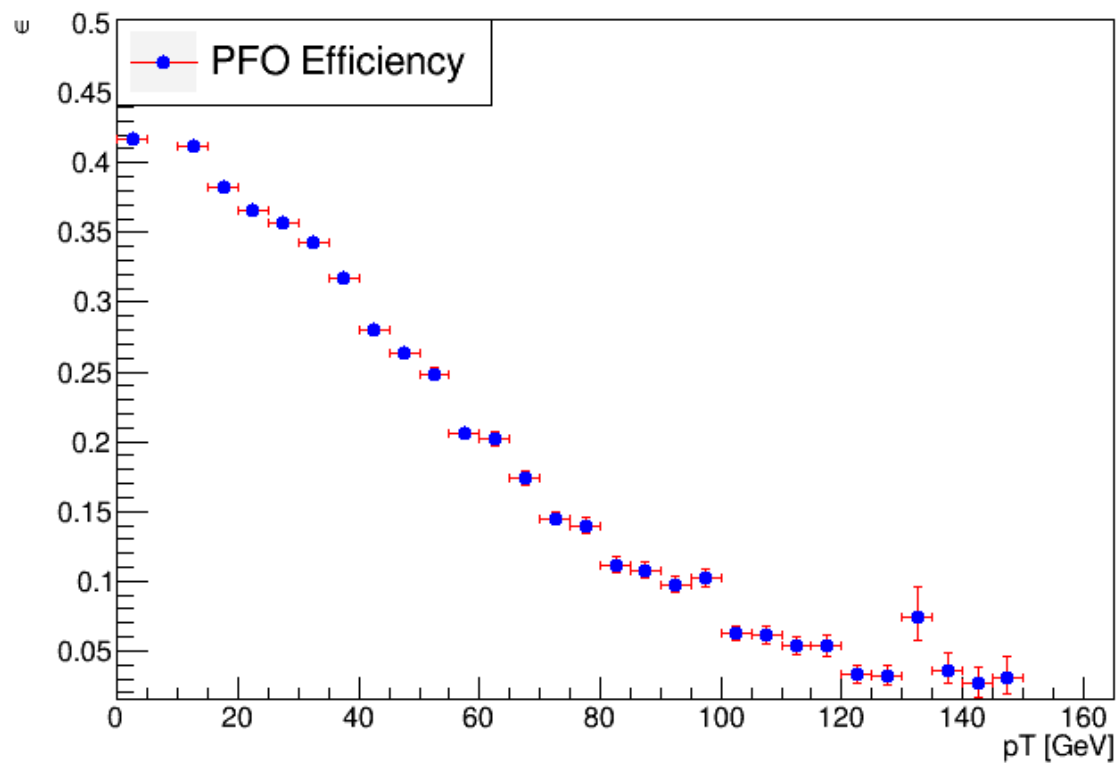# TAU RECO UPDATES – PION EFFICIENCY STUDIES
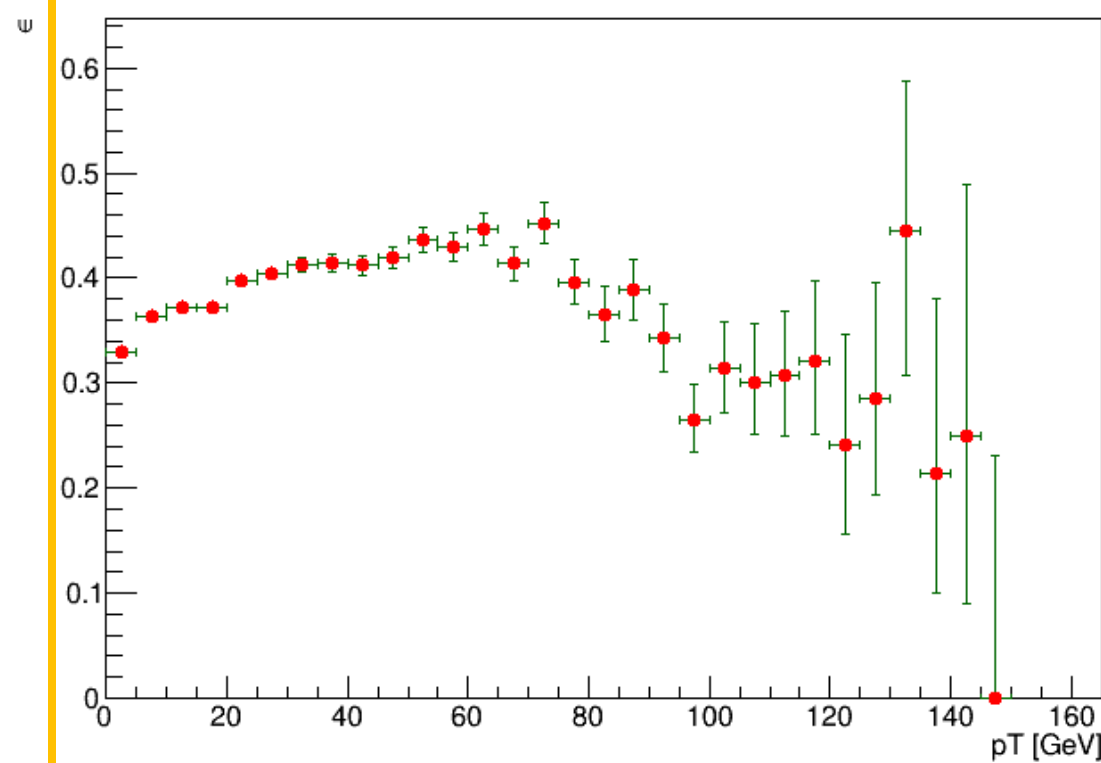
7/7/23

OVERVIEW

- Recall, overarching motivation: trace tau reconstruction inefficiency upstream and locate its origin

- Assessed PFO efficiency for all particle IDs

- Assessed mis-identification, both in general and with pions

- Looked under the hood of the Pandora ParticleID algorithms

- Traced problem further upstream, looked at more tracking efficiencies
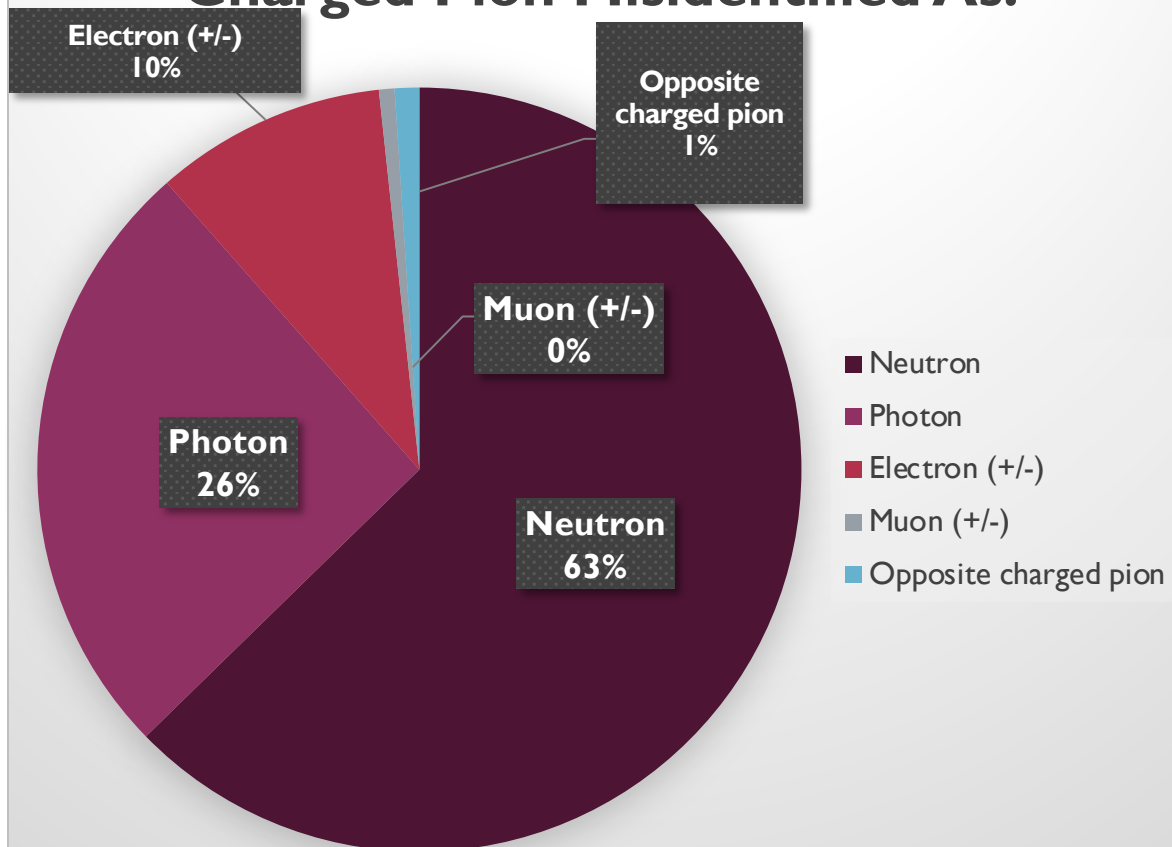
PFO EFFICIENCY (ALL PARTICLES)

CHARGED PION MIS-IDENTIFICATION

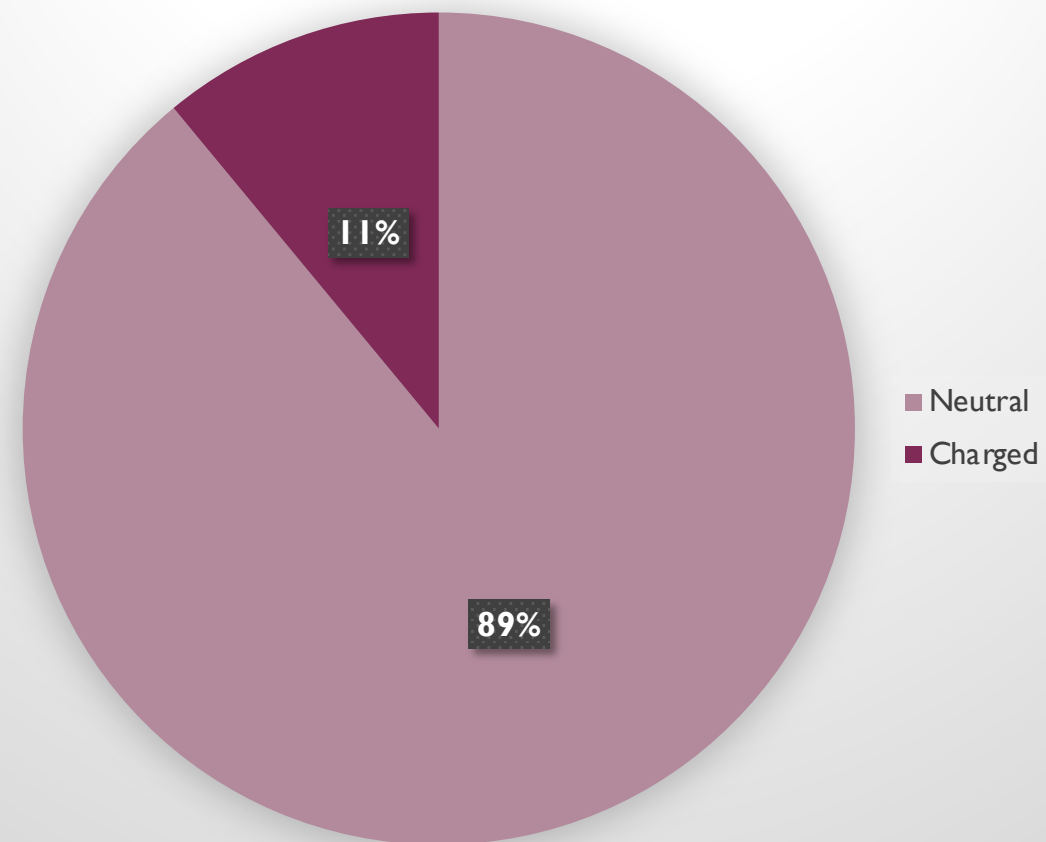Charged Pion Misidentified As:

- Neutron
- Photon
- Electron (+/-)
- Muon (+/-)
- Opposite charged pion

Electron (+/-) 10%
Opposite charged pion 1%
Muon (+/-) 0%
Photon 26%
Neutron 63%

Charge of MisID

- Neutral
- Charged

11%
89%

BREAKDOWN OF PANDORA ALGORITHMS

Input

Refitted, Deduped Tracks

Hit Bucket:
ECal Collections
HCal Collections

Event Preparation/ Hit Selection

Selected Trks, Hits

Muon Clustering
Photon Clustering
Track Association
Reclustering
Fragment Removal

Clusters

PFO Creation
Particle ID
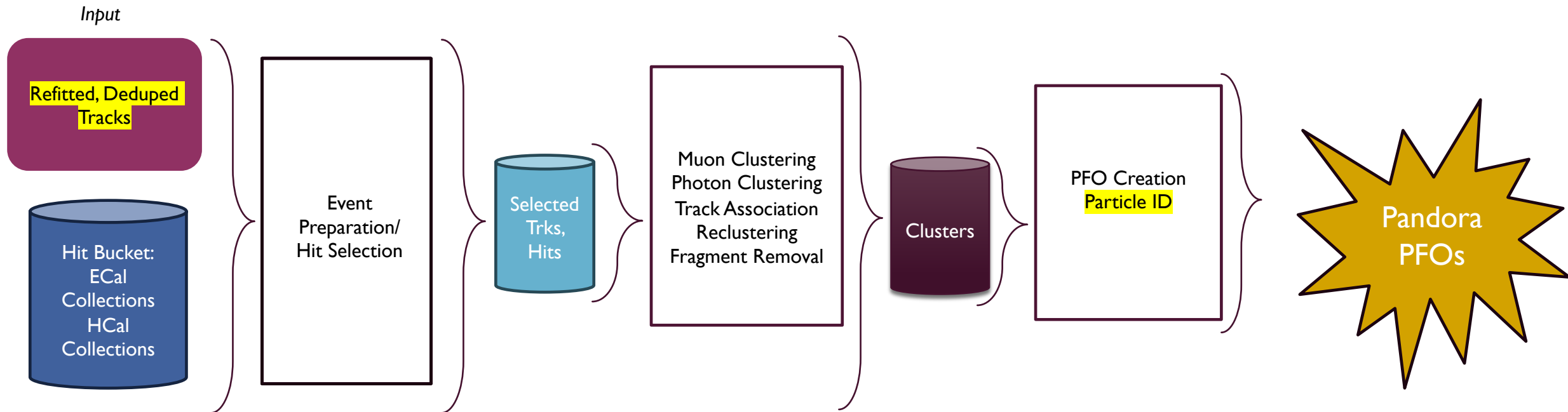
Pandora PFOs

# POOR TRACK ASSOCIATION→MIS-IDENTIFICATION OF CHARGED PARTICLES

- In the Pandora PFO Creation alg, particles w/o associated tracks are assigned as neutral candidates

- In order to address so many spurious neutral particles, need to address our poor track association efficiency

- Fig. at right: what fraction of PFO from charged MCP have one or more associated track?

- Hovering around 50% for reconstructed particles with truth charge = +/-1

```
StatusCode PfoCreationAlgorithm::CreateNeutralPfos() const
{
    const ClusterList *pClusterList = NULL;
    PANDORA_RETURN_RESULT_IF(STATUS_CODE_SUCCESS, !=, PandoraContentApi::GetCurrentList(*this, pClusterList));

    // Examine clusters with no associated tracks to form neutral pfos
    for (ClusterList::const_iterator iter = pClusterList->begin(), iterEnd = pClusterList->end(); iter != iterEnd;
    {
        const Cluster *const pCluster = *iter;

        if (!pCluster->GetAssociatedTrackList().empty())
            continue;
```
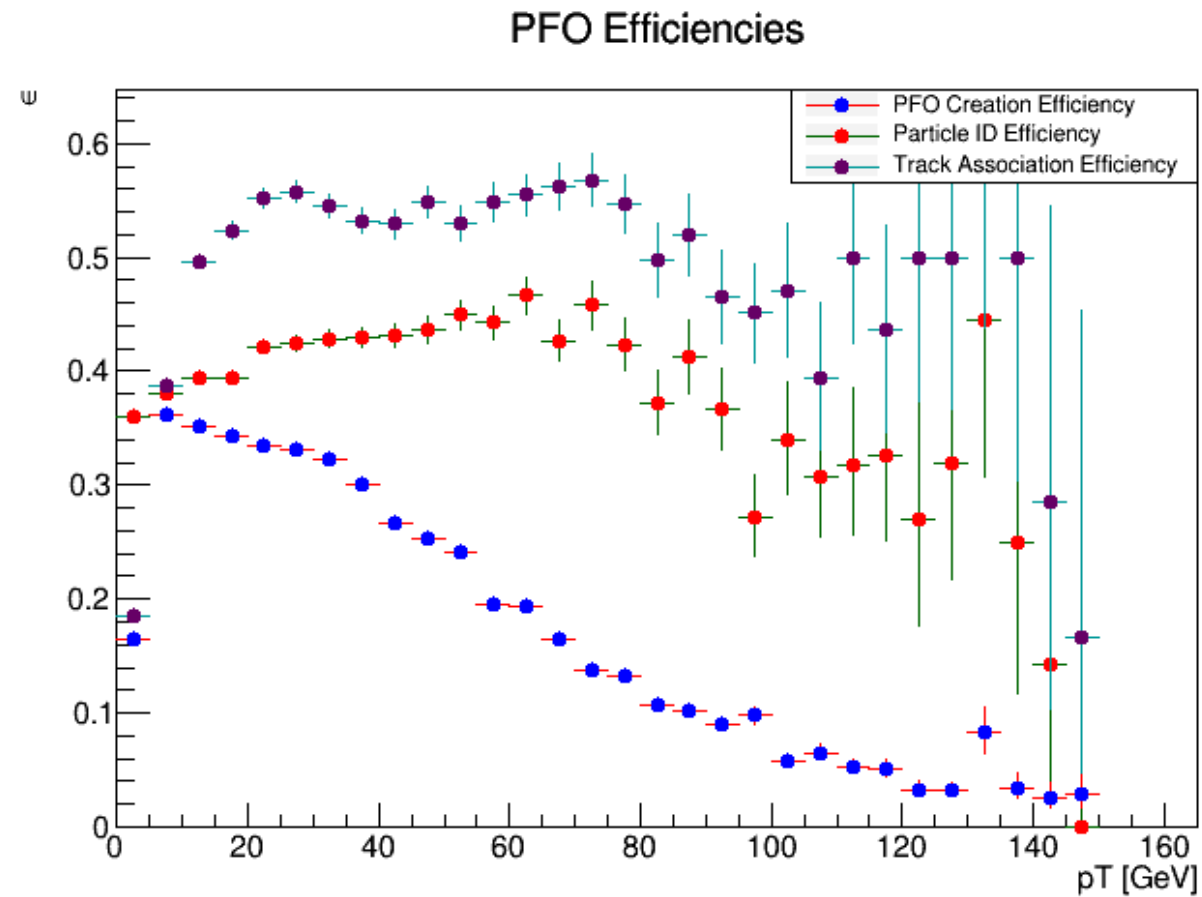

PFO Track-Association Efficiency
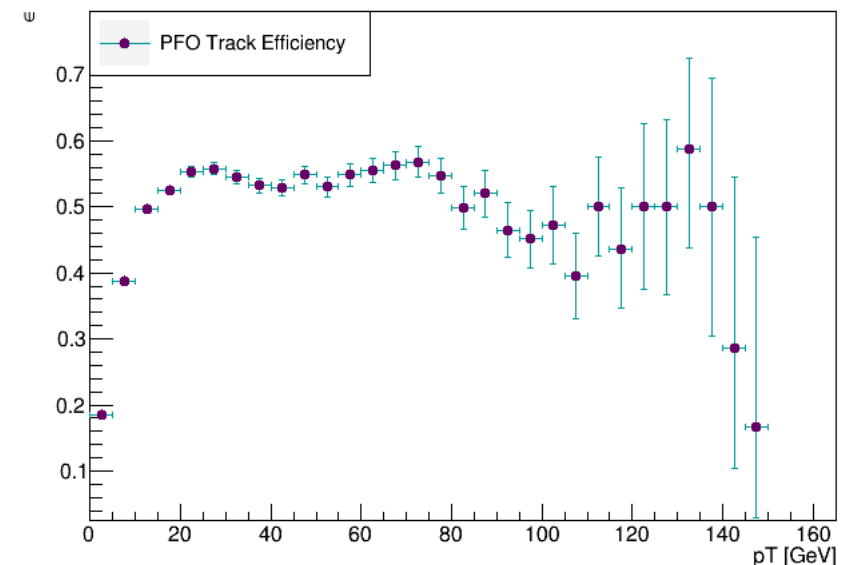
ALL 3 EFFICIENCIES OVERLAID

PFO Efficiencies

# CHECK: HOW IS OUR PID EFFICIENCY FOR PFO W/ TRKS?

- Selected only those PFOs with associated tracks

- Our particle identification efficiency for these particles is good (make allowances for low statistics)

- Therefore, we can conclude that **Pandora particle ID is functioning, and the problem is upstream**

ID Efficiency for PFO with Associated Tracks
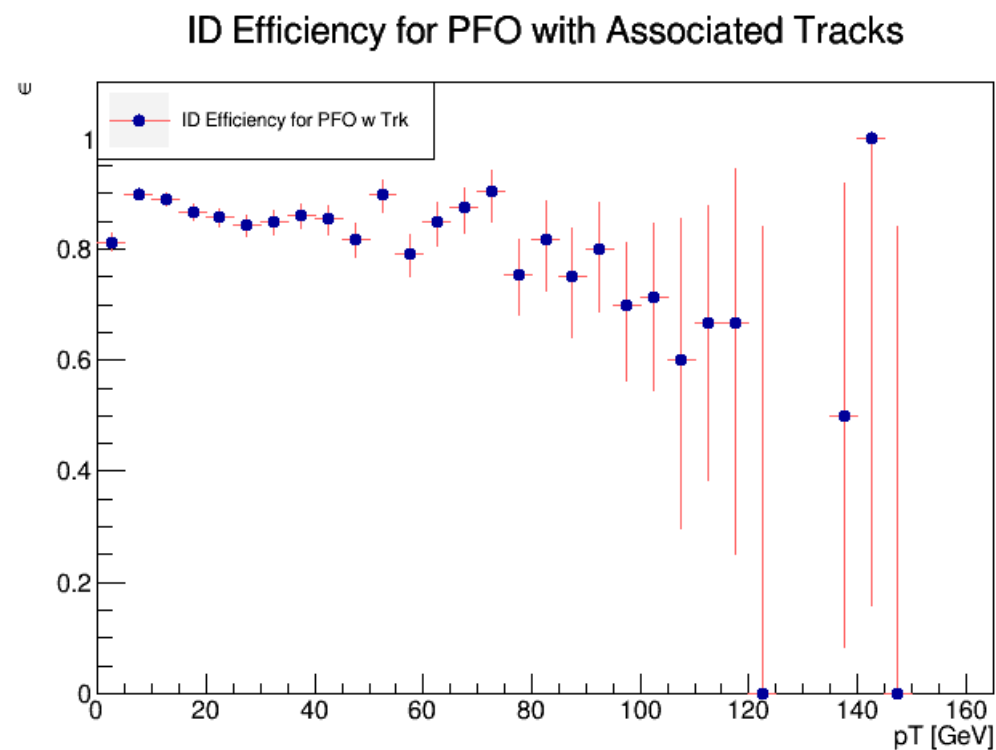
TRACKING EFFICIENCY ON A PID-SPECIFIC LEVEL

- Overall tracking efficiency (assessed by making 1-1 matches between tracks and charged MCP) is not necessarily indicative of the performance we care about

- Manual debugging revealed that most of the charged MCPs without tracks were either taus (PDG=15) or particles with unusual ID that I could not find in the PDG database (e.g. 1000551250)

- Isolated tracking efficiency for pions individually might be more insightful (see at right for 20k events)

  - No Pandora reco, just ACTSTracking

Tracking Efficiency (pions)
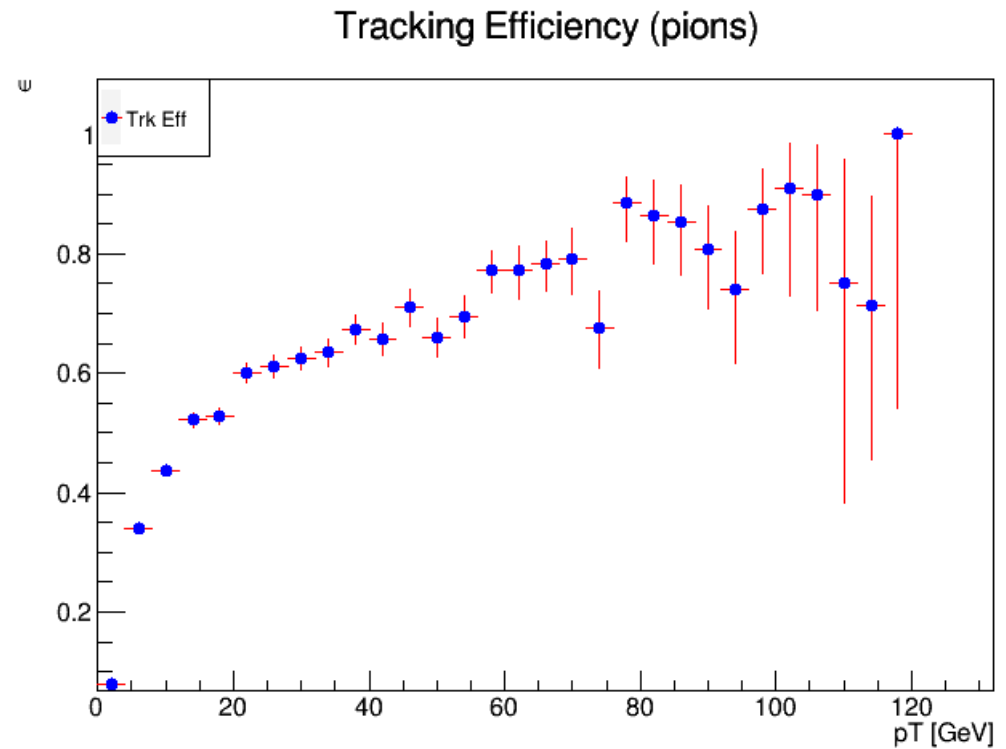
TAKEAWAYS/NEXT STEPS

- Huge mis-identification problem due to PFO without associated tracks

- Tracking efficiency for pions, while not perfect, is not as bad as expected

- Looking back at the Pandora structure, there is a track selection step between the tracking and the particle identification where more tracking cuts happen…perhaps these are not optimized?

- Next: examine more carefully this step, look at the tracking cuts, and see if this is where we are losing efficiency