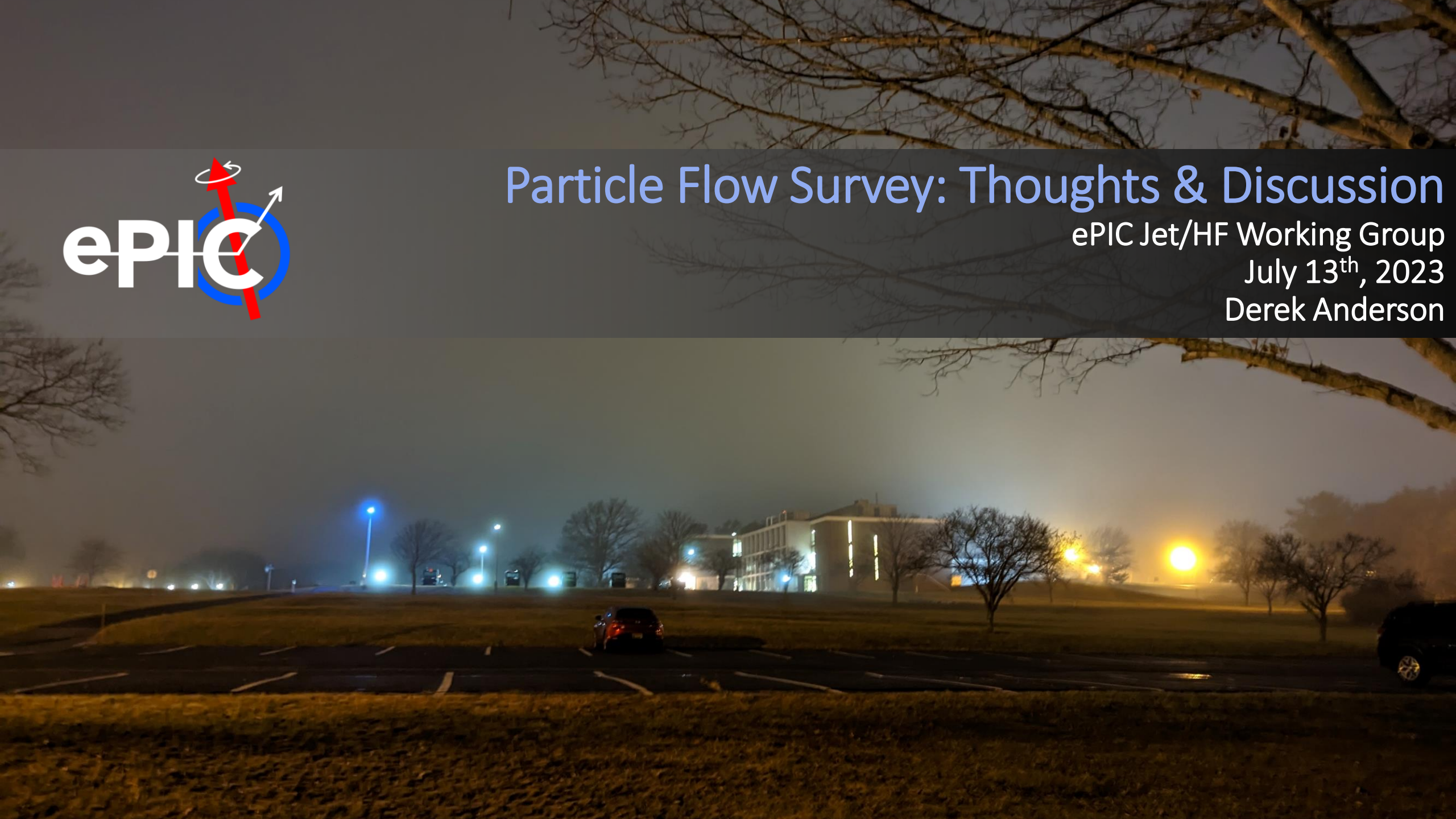


# Particle Flow Survey: Thoughts & Discussion

ePIC Jet/HF Working Group

July 13th, 2023

Derek Anderson

# Particle Flow Squad | Introduction

- Physics Analysis and C/S Coordinators identified 4 priorities for reconstruction software:
  - Vertexing and PID
  - Low Q2 Tagger
  - Electron Finder
  - **Particle Flow**

∴ Formed 4 squads inside reconstruction working group to address topics
  - 👉 **PF Squad Charge:** improve jet reconstruction using particle flow information

- 2 distinct regions for PF at ePIC
  - **Barrel/Backward:** JER set by tracker + EMCal
    - 👉 Need tracks to deconvolve clusters for neutrals
  - **Forward:** JER can be improved by combining track + calorimeter information
    - 👉 Need to separate overlapping clusters

- **2 initial tasks:**
  - 👍 Survey existing implementations
  - 👉 Explore necessity of custom approach in barrel/backwards

# Particle Flow Survey | General

- **Particle flow:** use appropriate detectors to measure appropriate particles
  - Avoid double-counting energy
  - a.k.a “Energy Flow”

- **At its most basic:**
  1) Match tracks to calorimeter clusters
  2) Subtract track energy from calorimeter energy

⇒ (Almost) all algorithms build on this in some way

- Some examples
  - CELLO (PETRA)
  - ALEPH (LEP)
  - DELPHI (LEP)
  - H1 (HERA)
  - D0 (Tevatron)
  - CDF (Tevatron)
  - Pandora (ILC/CLIC, MicroBooNE)
  - ATLAS (LHC)
  - CMS (LHC)
  - sPHENIX (RHIC)

# Particle Flow Survey | ALEPH & DELPHI

- ALEPH and DELPHI implemented the first "modern" PF algorithms
  - ☞ Both make use of PID capabilities

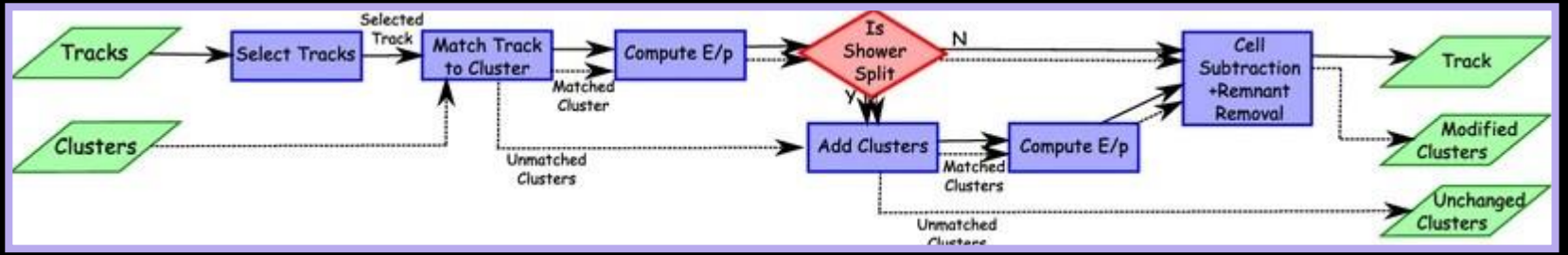
**ALEPH:**

- Associate tracks to clusters and iteratively clean track+cluster objects
- **Algorithm:**
  1) Project tracks & associate them to clusters
  2) Identify $e^{\pm}$ & remove them
  3) Identity $\pi^0/\gamma$ & remove them
  4) Identify $\mu^{\pm}$ & remove them
  5) Do track-cluster subtraction
  6) Any remaining calorimeter energies are flagged as $h^0$

**DELPHI:**

- Very similar, but slightly different order of operations
- **Algorithm:**
  1) Identify $e^{\pm}/\gamma$ & remove them
  2) Extrapolate tracks through HPC (EMCal) + HCal
  3) Any clusters "close" to extrapolated tracks are associated with track and removed
  4) Any remaining clusters are flagged as $h^0$

# Particle Flow Survey | ATLAS




ATLAS [arXiv:1703.10485]

- ATLAS makes use of a sophisticated variation on the "cluster – track" idea
  - ☞ Very similar (but still distinct) to what's being utilized at sPHENIX

**ATLAS:**

- **Algorithm:**
  1) Match tracks to clusters
  2) Determine if cluster is split
     a) **If yes**, then add more clusters to track+cluster object
     b) **Otherwise** move on
  3) Subtract expected track energy cell-by-cell from clusters
  4) **Return:**
     - Tracks
     - Matched clusters w/ nonzero energy after subtraction
     - Unmatched clusters

# Particle Flow Survey | PandoraPFA

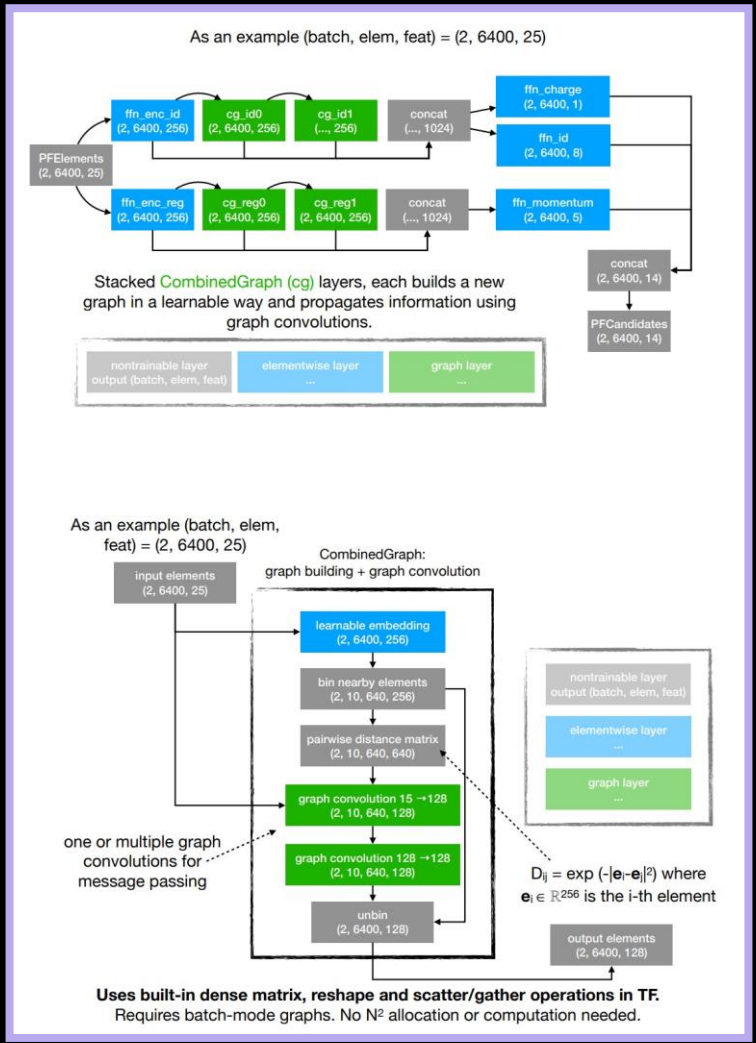
- ILC/CLIC (esp. CALICE) efforts have focused on PF
  - Produced many algorithms, e.g.
    - › Arbor
    - ☞ **PandoraPFA**

- **PandoraPFA:** a *very* sophisticated PF algorithm for high granularity calorimeters
  - Part of AIDAsoft
  - Has produced an extensive detector-agnostic implementation
  - Currently deployed at MicroBooNE

**Algorithm:** 8 stages in total

1) Select tracks for analysis
2) Select calorimeter cells & cluster based on geometry
3) Recluster cells into cones around track+EMCal projections:
   i. 1st identify and remove possible $\gamma$ clusters
   ii. Then cluster remaining cells
4) Recluster non-$\gamma$ clusters based on topology
5) Attempt to split overlapping clusters
6) Apply more sophisticated $\gamma$-ID algorithm to separate $\gamma$ from $h^0$
7) Neutral fragments from $h^{\pm}$ are identified and removed
8) **Return:** "PF Objects"
   - Matched track+cluster objects with rudimentary PID applied

# Particle Flow Survey | MLPF at CMS




CMS [JP:CS **2438**, 012100 (2023)]

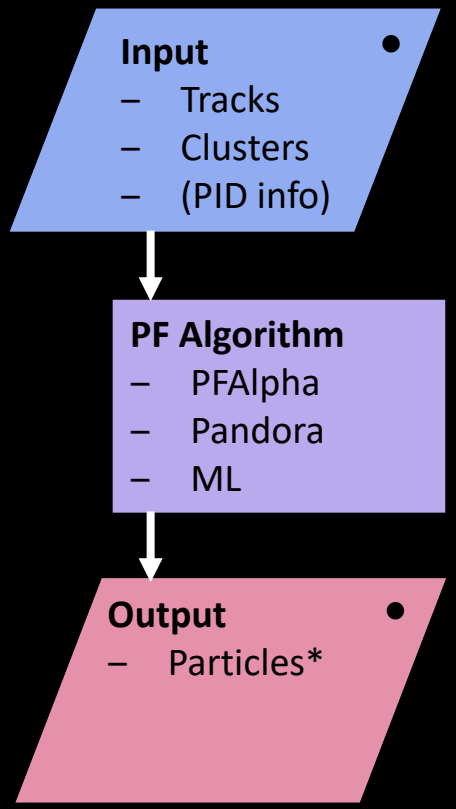
- An ML alternative to existing algorithm is being explored at CMS
  - Ref.s:
    - EPJC **81**, 381 (2021)
    - JP:CS **2438**, 012100 (2023)
  - Could ease computational requirements

**Implementation:**

1) Extensive track and calorimeter information is fed to a GNN model
2) GNN converts track/calorimeter hits/cells into connected graphs
3) Graphs are then regressed to particles

# Particle Flow Discussion | Algorithm Strawman

**General Reconstruction Flow:**




* = ReconstructedParticle object
● = fixed

- Infrastructure to do "bare-bones" PF *largely* exists
  - ⇒ Could implement a basic "alpha" **(PFAlpha)** algorithm
- **Rationale:**
  - Motivate and test development of necessary software
  - Serve as baseline to compare refinements against
  - Allow analyzers to quickly start working w/ output
- Development then proceeds with testing more refined approaches, e.g.
  - Such as PandoraPFA
  - ML-based models
  - Etc.

# Particle Flow Discussion | Algorithm Strawman

**PFAlpha:**

1) Project tracks through calos
2) Associate all calo clusters within a cone of size R around the track
3) Sum all calo energy in cone and subtract expected track energy from sum
4) **Return**
   - Tracks
   - Subtracted clusters
   - Unassociated clusters

- Infrastructure to do "bare-bones" PF *largely* exists
  - ⇒ Could implement a basic "alpha" **(PFAlpha)** algorithm
- **Rationale:**
  - Motivate and test development of necessary software
  - Serve as baseline to compare refinements against
  - Allow analyzers to quickly start working w/ output
- Development then proceeds with testing more refined approaches, e.g.
  - Such as PandoraPFA
  - ML-based models
  - Etc.

# Particle Flow Discussion | To-Do

**Missing Infrastructure (Major):**
- PF Framework
  - Factories
  - Algorithm + configuration files
- Improved track-cluster associator
  - Extend to include Hcals
  - However, truth-based implementation may work for interim

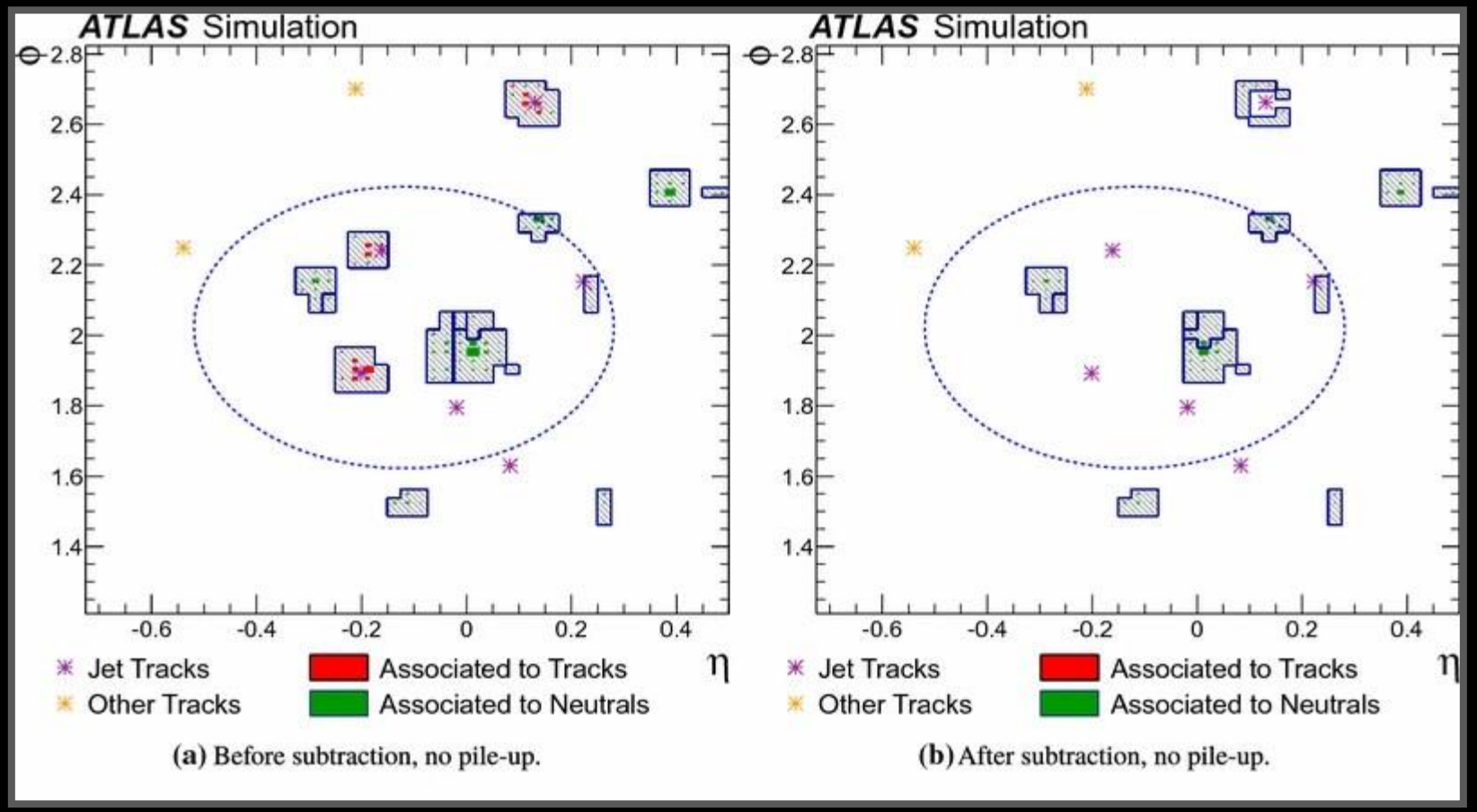
**Missing Infrastructure (Minor):**
- PFObject Visualizer:
  - Plugin (or service?) to visualize clusters, tracks, etc.
  - Crucial for debugging
- Downstream analysis:
  - Code to look at impact of changes
  - Existing jet benchmarks are good starting place

**Open Questions:**
- Does implemented cluster splitting work in non-enabled* detectors?
- How well do existing MC-cluster associations work?
  - Currently handled by MatchClusters algorithm
  - Would a separate MC-cluster associator be better?

**Major =** necessary for implementation
**Minor =** can be pursued in parallel with implementation
**Yellow =** connection with other groups
* = existing implementation enabled for central ECals and ECalLumiSpec (not enabled for Imaging/SciFi)


Thank you!

# Backup | Example PFObject Visualizer




ATLAS [arXiv:1703.10485]