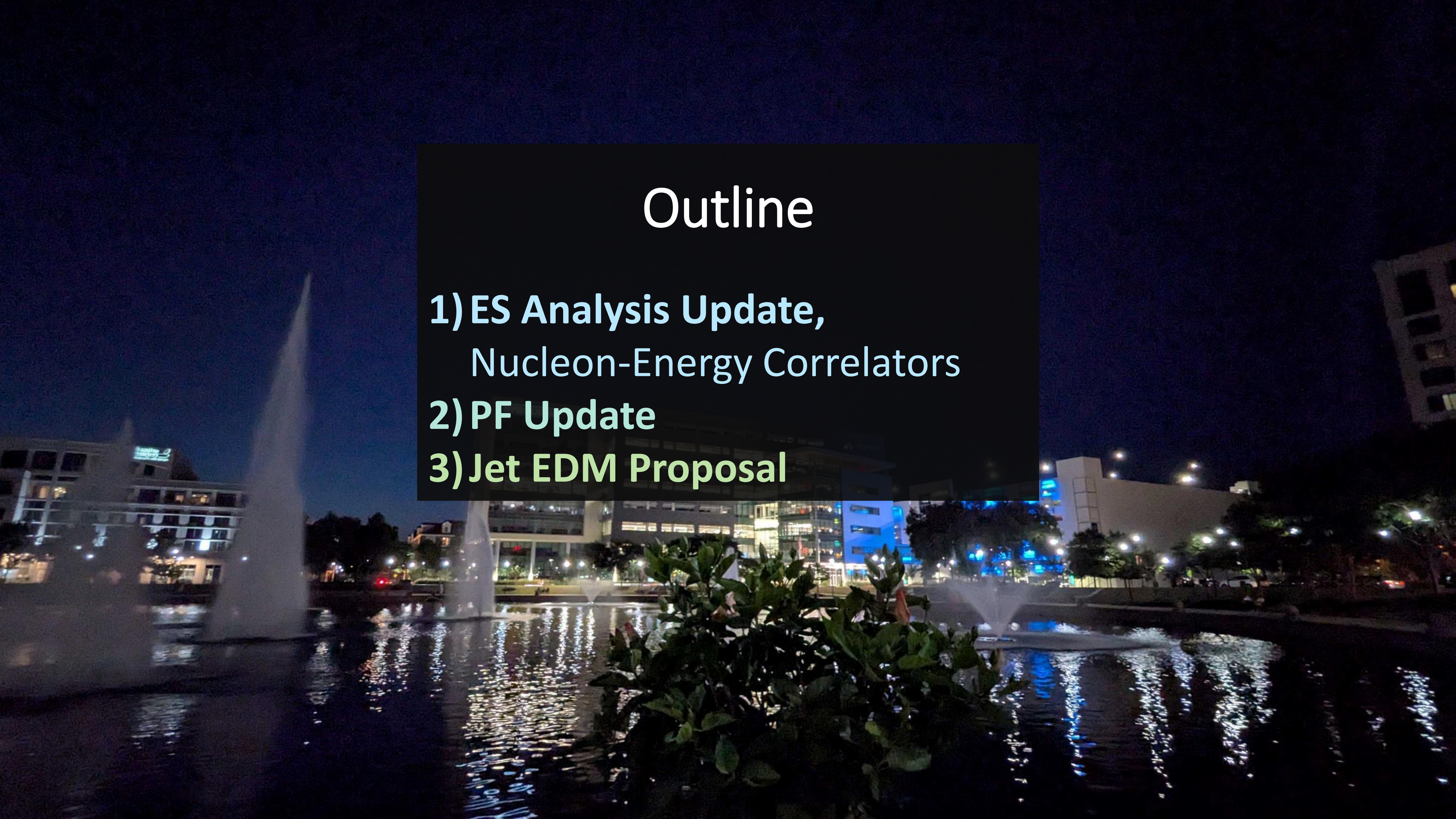
ES Analysis & Software Updates

EIC UGM 2025, Jet/HF Workfest

Derek Anderson

Jefferson Lab



Outline

1) **ES Analysis Update,**
   Nucleon-Energy Correlators
2) **PF Update**
3) **Jet EDM Proposal**

ES Analysis Update | Energy-Energy Correlators

2-Point Energy Correlators (EEC)

$$\mathrm{EEC}(\theta) = \frac{1}{\sigma} \sum_{(i,j)\in \mathrm{jet}} \int d\sigma \frac{E_i E_j}{E_{\mathrm{jet}}^2} \delta(\theta_{ij} - \Delta\theta)$$

- E_i= energy of i$^{\mathrm{th}}$ jet constituent
- θ = distance between pairs of jet constituents

○ **Energy-Energy Correlators:** measure statistical correlations in energy flux

  ☞ 2-point allows for clean imaging of jet structure as function of angular scale

  – Very effective tools at hadron colliders, and potentially at EIC!

○ Commonly measured in jets today

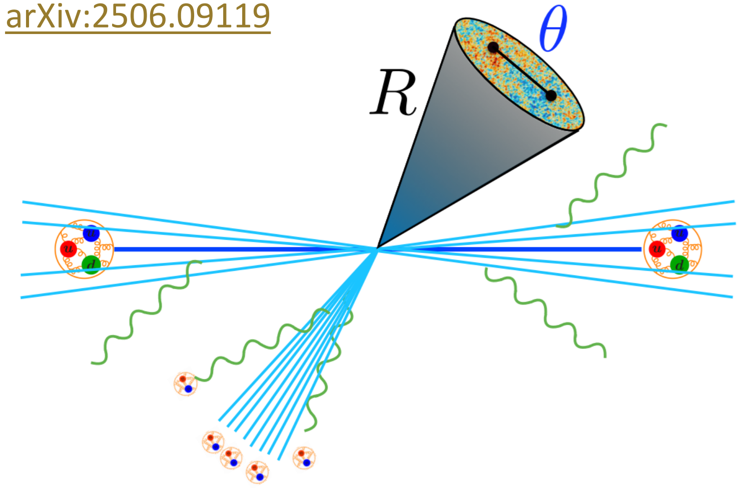
  – But historically was over all particles in event!

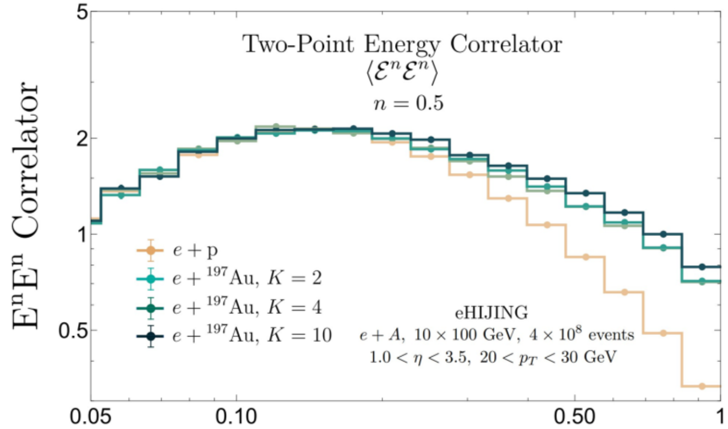
    › C.f. PRL 41 (1978) 1585

  – More event-shape approach being revisited today

    › E.g. aziumuthally-dependent EECs (arXiv:2310.1519), or **Nucleon-Energy Correlators**

arXiv:2506.09119





arXiv:2303.08143 θ

ES Analysis Update | Nucleon-Energy Correlators

Nucleon-Energy Correlators (NEC)

$$\text{NEEC} = \sum_i \int d\sigma(x_B, Q^2, p_i) x_B^{N-1} \frac{E_i}{E_p} \delta(\theta - \theta_i)$$

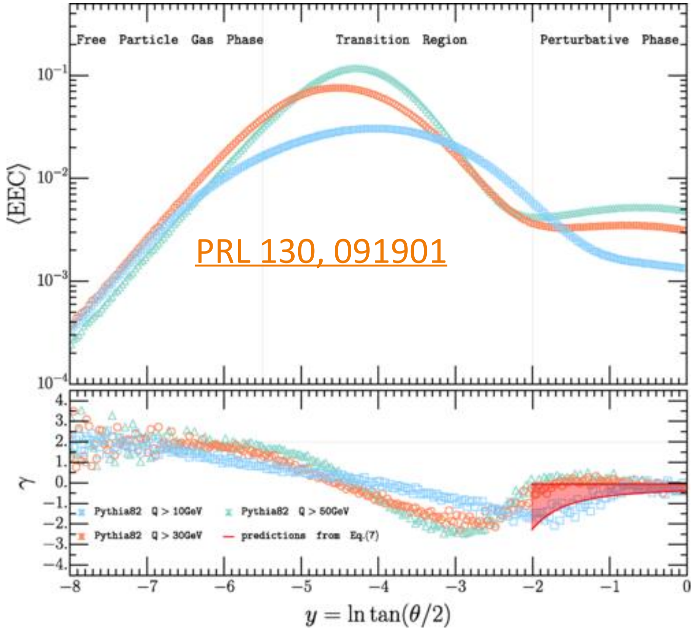
- E_i, θ_i = energy, Breit frame angle of ith particle
- E_p = energy of scattered proton

○ **Nucleon-Energy Correlators:** provide a variation of EECs applicable to both collider & fixed target kinematics!

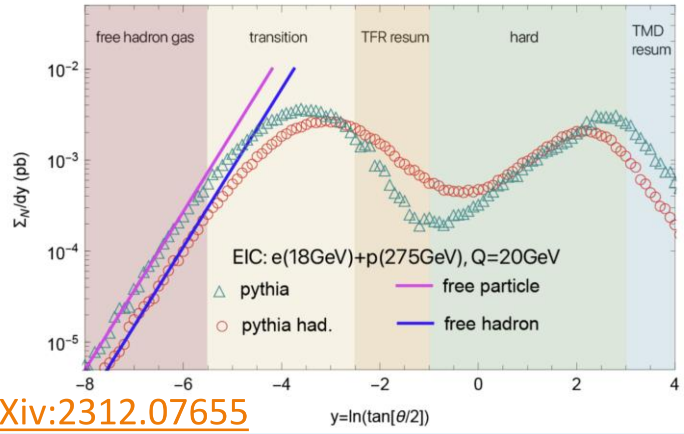
- **In essence:** $\theta_{ij} \rightarrow \theta_{breit}$
  - › Enables sensitivity to different physics processes by selecting different θ_{breit}
- ☞ See PRL 130, 091901, arXiv:2312.07655,

○ **Worthwhile observable to study because:**

- Potential connections
  - › To TMD PDFs (arXiv:2403.08874)
  - › To Fracture Functions (arXiv:2406.08559)
- Stress test of our PF algorithms



PRL 130, 091901



arXiv:2312.07655

ES Analysis Update | Current Status

- **Current status:** in code development!
  - Analysis code on GitHub at:
    - EPNucleonEnergyCorrelator.cxx
    - EPNucleonEnergyCorrelator.jl

- **EPNucleonEnergyCorrelator.cxx:** c++ implementation using RDataFrames
  - Prototyping code with a ROOT macro
    - ☞ **Planning to convert to physics benchmark**
  - Will evolve code into a compiled library for ease of deployment and extension when ready

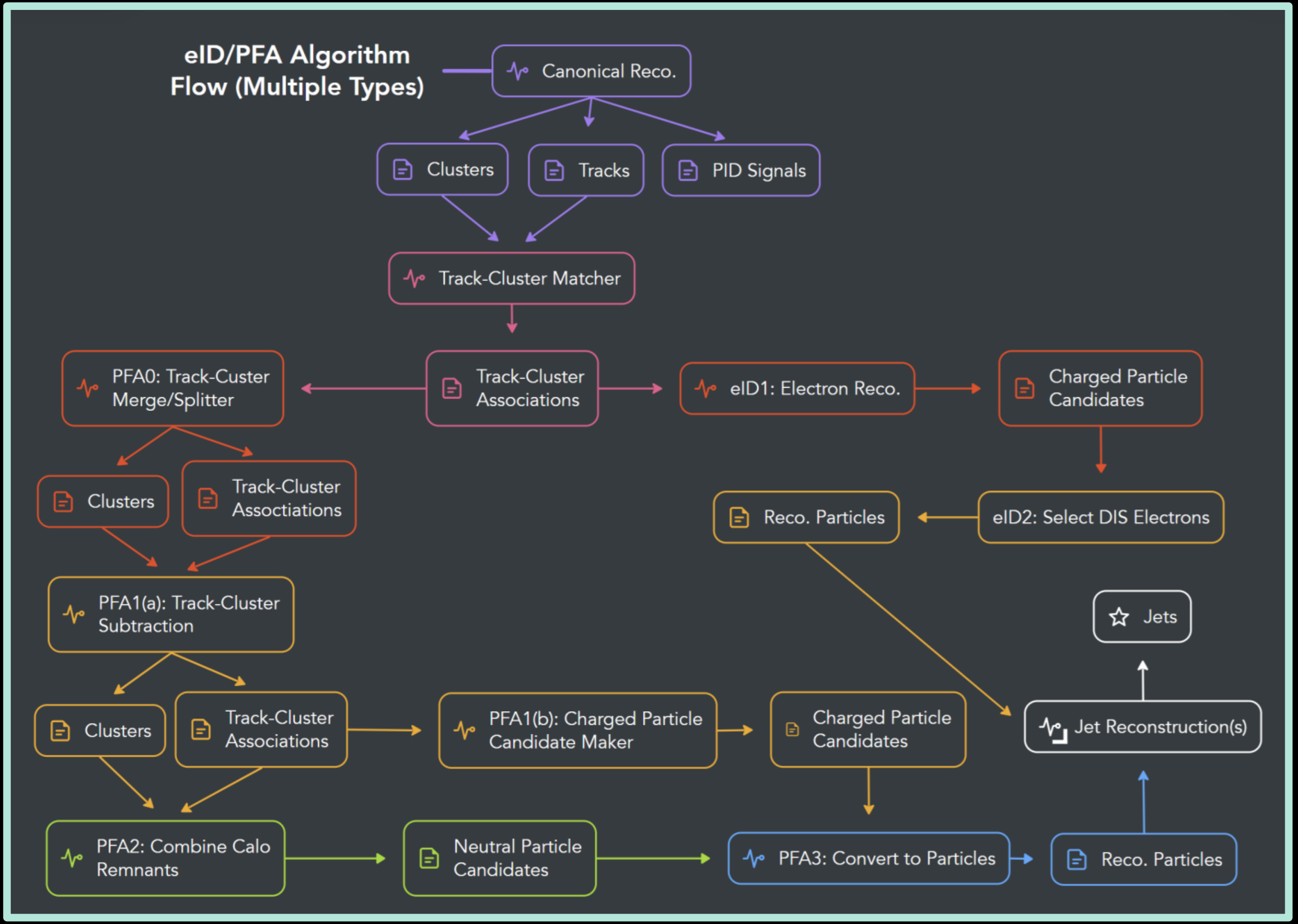
- **EPNucleonEnergyCorrelator.jl:** planned Julia implementation
  - ☞ Won't start on until c++ version is functioning and we have a solid set of Early Science plots

- **1st Round of Plots:** next meeting! (Apologies!)

Particle Flow Update | In-Progress Baseline

PFAlpha: the proposed baseline, looks like:

1) [PFA-1] Match tracks to EMCal, HCal clusters
2) [PFA0] Merge clusters based on track E/p in a cone of size R_0
3) [PFA1a] Subtract expected track energy from merged clusters
   - Split into tracks + expected energy, and remnant clusters (leftover energy)
4) [PFA1b] Convert tracks + expected energy to particle candidates
5) [PFA2] Combine remnant EMCal, HCal clusters in a cone of size R_1, convert to particle candidate
6) [PFA3] Covert candidates to reconstructed particles



eID/PFA Algorithm Flow (Multiple Types)

Canonical Reco.

Clusters

Tracks

PID Signals

Track-Cluster Matcher

PFA0: Track-Custer Merge/Splitter

Track-Cluster Associations

eID1: Electron Reco.

Charged Particle Candidates

Clusters

Track-Cluster Associations

Reco. Particles

eID2: Select DIS Electrons

PFA1(a): Track-Cluster Subtraction

Jets

Clusters

Track-Cluster Associations

PFA1(b): Charged Particle Candidate Maker

Charged Particle Candidates

Jet Reconstruction(s)

PFA2: Combine Calo Remnants

Neutral Particle Candidates

PFA3: Convert to Particles

Reco. Particles

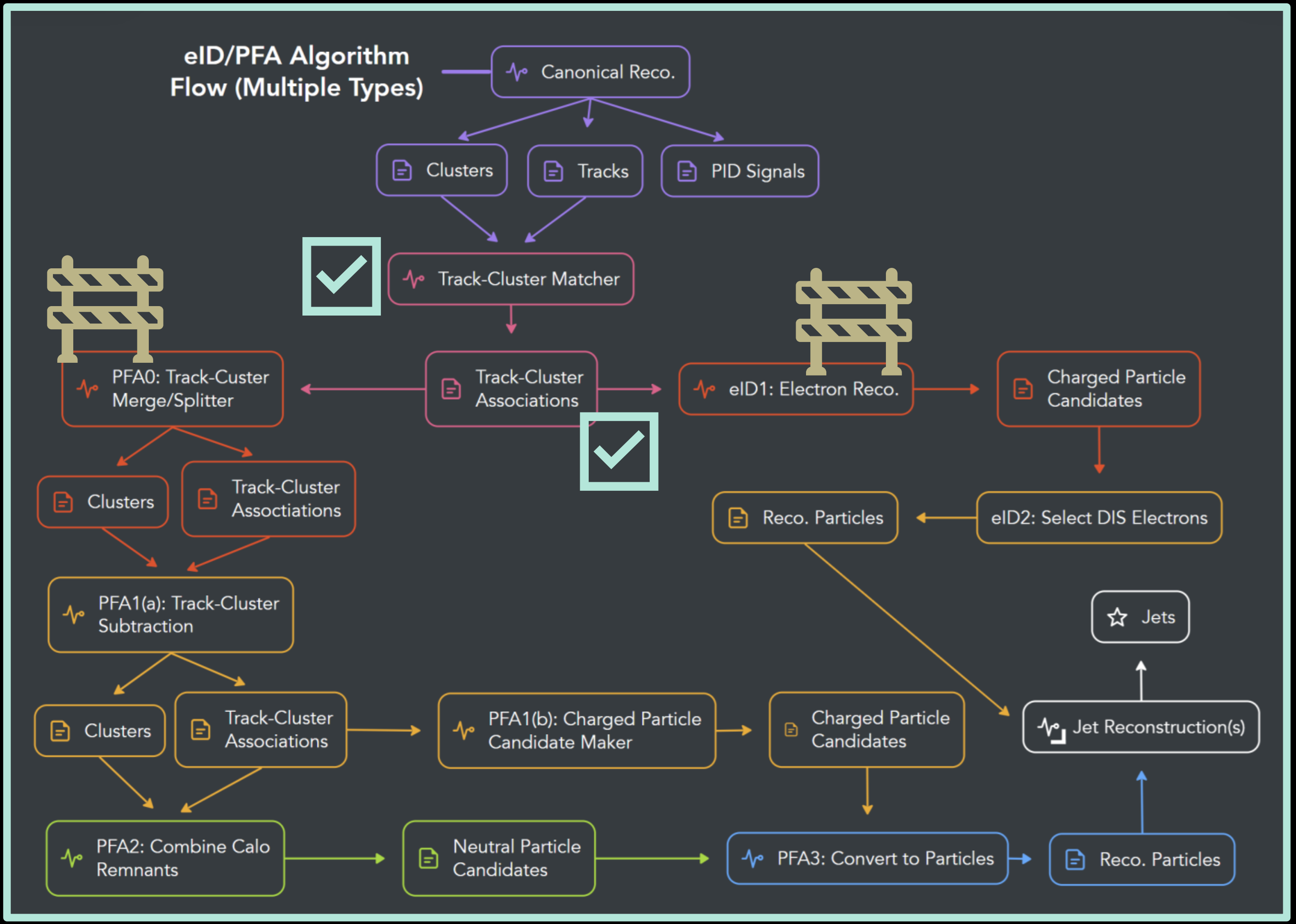
Particle Flow Update | Timeline Since Frascati

ePIC

Developments since Frascati CM:

1) [02.06.2025] PFA0 PR opened and in progress
   ☞ (See next slide)
2) [02.23.205] PFA1(a) PR ready to open after PFA0 merged
   ☞ (PR needs update)
3) [02.27.2025] Held 1st PF SubWG meeting
4) [03.11.2025] Draft PR for candidate types open
5) [03.28.2025] Initial track-cluster matcher merged
6) [05.06.2025] Track-Protocluster link merged
7) [07.06.2025] Multi-calo track-cluster matcher merged

☑ = Done
🚧 = In-progress



eID/PFA Algorithm Flow (Multiple Types)

Canonical Reco.

Clusters

Tracks

PID Signals

Track-Cluster Matcher

PFA0: Track-Custer Merge/Splitter

Track-Cluster Associations

eID1: Electron Reco.

Charged Particle Candidates

Clusters

Track-Cluster Associations

Reco. Particles

eID2: Select DIS Electrons

PFA1(a): Track-Cluster Subtraction

Jets

Clusters

Track-Cluster Associations

PFA1(b): Charged Particle Candidate Maker

Charged Particle Candidates

Jet Reconstruction(s)

PFA2: Combine Calo Remnants

Neutral Particle Candidates

PFA3: Convert to Particles

Reco. Particles

Particle Flow Update | Current Status and To-Do’s

- **Immediate task:** Complete PFA0 (EICrecon#1699)
  - Track-Protocluster Links now implemented (EDM4eic#108)
  - Began integrating into merge/splitter, but Link Collections seemingly don’t play nicely with Omnifactories...
  - Discussed with Nathan (now investigating), need to circle back

- **Next Steps:** after PFA0
  - Implement "promotion" algorithm to convert Track-Protocluster Links to Track-Cluster Matches
  - Revive and finish PFA1 (EICrecon#1627)
  - PFA2 (calo remnant combiner) on hold until PFA1 is merged

- **Current Status:** work paused for a while, now resuming
  - Pause due to professional transitions, conferences, etc.

- Want to restart semi-regular PF technical meetings after EIC UGM!
  - ☞ **Look for emails & time polls soon**

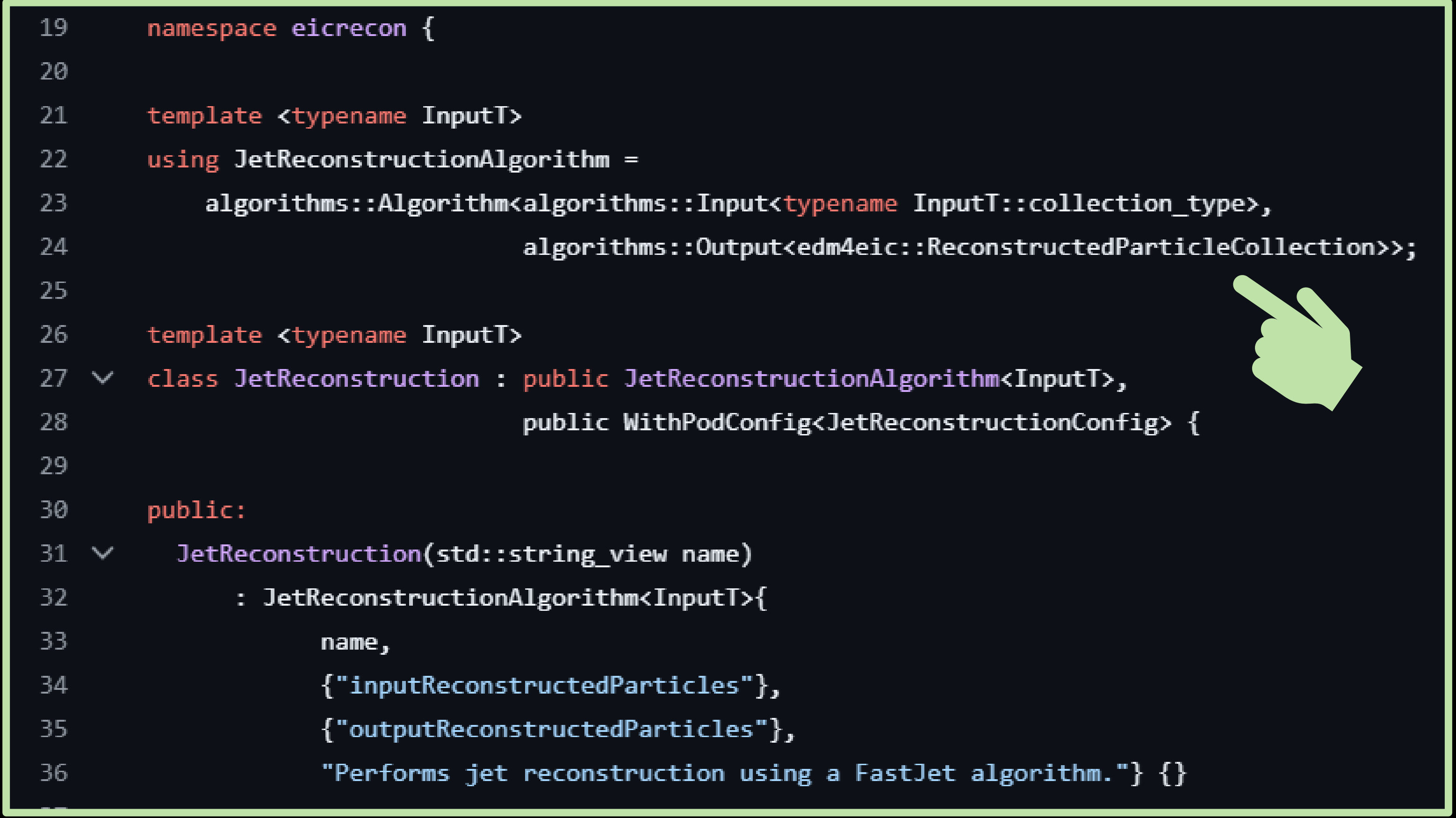
- **Related tasks**
  - Make sure all relevant algorithms are using track-cluster matches
  - Implement algorithm to convert track-cluster matches to reco particles (for eID)
  - Upgrade Track-Cluster Matches from *associations* to *links*

Jet EDM Proposal | Context

- ○ **Jets produced in EICrecon:** currently stored as edm4eic:: ReconstructedParticle
  - – Workaround due to lack of dedicated jet type
  - – Not ideal for several reasons:
    - › Jets are extended objects
    - › Jets can pick up background
    - › Conceptual clarity (e.g. parton, jet not 1-to-1)

∴ Put together a proposal inspired by FastJet PseudoJet!

- – **PR:** EDM4eic#118
- – **Issue:** EDM4eic#88

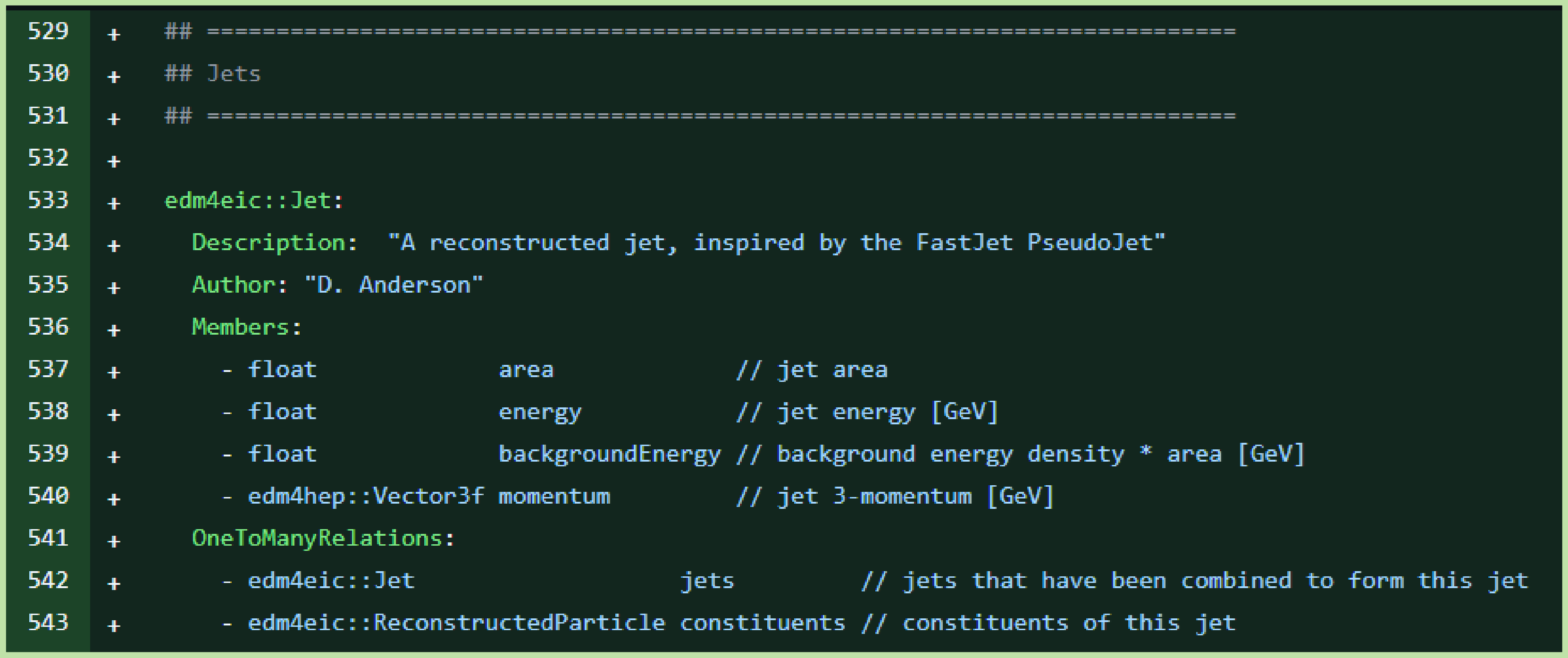


```
namespace eicrecon {

template <typename InputT>
using JetReconstructionAlgorithm =
    algorithms::Algorithm<algorithms::Input<typename InputT::collection_type>,
                          algorithms::Output<edm4eic::ReconstructedParticleCollection>>;

template <typename InputT>
class JetReconstruction : public JetReconstructionAlgorithm<InputT>,
                          public WithPodConfig<JetReconstructionConfig> {

public:
  JetReconstruction(std::string_view name)
      : JetReconstructionAlgorithm<InputT>{
            name,
            {"inputReconstructedParticles"},
            {"outputReconstructedParticles"},
            "Performs jet reconstruction using a FastJet algorithm."} {}
```

Jet EDM Proposal | Proposal

- **Right:** the proposed jet type
  - Energy, momentum fields identical to Reconstructed Particle
  - Adds field for area, background energy density

- **Note 3 design choices:**
  1) Only relations to reconstructed particles allowed
     - ☞ Places jet reco at the final stages of reconstruction
  2) Jet substructure *not* included
     - ☞ Deferred to analysis
  3) Relation to other jets allows for indication of sub-jets, seed jets, or background jets



```
+   ## ===========================================================================
+   ## Jets
+   ## ===========================================================================
+
+   edm4eic::Jet:
+     Description:  "A reconstructed jet, inspired by the FastJet PseudoJet"
+     Author: "D. Anderson"
+     Members:
+       - float             area             // jet area
+       - float             energy           // jet energy [GeV]
+       - float             backgroundEnergy // background energy density * area [GeV]
+       - edm4hep::Vector3f momentum         // jet 3-momentum [GeV]
+     OneToManyRelations:
+       - edm4eic::Jet                   jets         // jets that have been combined to form this jet
+       - edm4eic::ReconstructedParticle constituents // constituents of this jet
```

Jet EDM Proposal | Downstream Changes

What components would need to be updated?

1) **[EICrecon]** jet reconstruction algorithm
2) **[physics_benchmarks]** jet benchmark
3) **[snippets]** TTree reader jet example
   - ☞ Not required, but good for documentation!
4) All user analysis scripts/macros

Any others?

What changes would we need to make?

1) Change edm4eic::ReconstructedParticle → edm4eic::Jet
2) Change getParticles() → getConstituents()
3) Remove any references to getPDG(), getCharge()

But that's it!

- ☞ Otherwise new type would behave identical to Reconstructed Particle

Thanks!

Questions, Discussion?

Backup | Initial PFA Attempt

- ○ **PFAlpha:** initial stab in EICrecon#1186 (now closed)
  - Initial implementation aimed for just a single algorithm
  - Initially even aimed to handle all 3 regions of central detector in one algorithm…

- ○ **The gist:**
  1) Project tracks through calos
  2) Associate all calo clusters in cone of size R around track
  3) Sum all calo energy in cone and subtract expected track energy from sum
  4) Merge leftover clusters in cones of size R
  5) **Return PFObjects (reco. particles)**
     - Tracks
     - Subtracted, merged clusters

  ☞ (Details in backup)

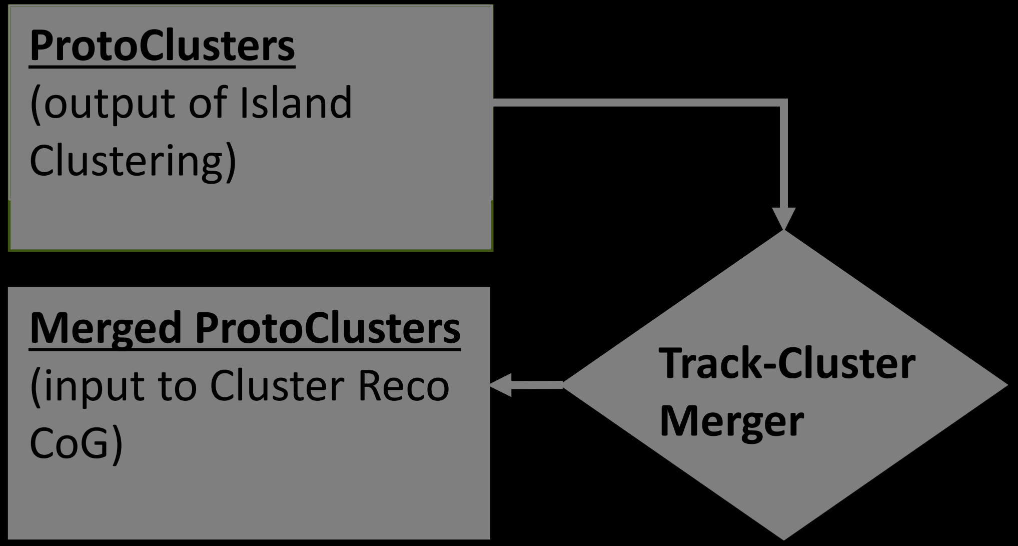
- ○ **Clear Drawbacks!**
  - ☒ Monolithic by definition
  - ☒ Hard to maintain, evolve
  - ☒ Wiring in new PF algorithms means rewriting lots of code

Parameters

- R^{ECal}_{sum}: radius in (η, φ) in which to combine ECal clusters
- R^{HCal}_{sum}: same but for HCal
- f^{ECal}_{sub}: fraction of track energy to subtract from ECal clusters
- f^{ECal}_{sub}: same but for HCal

Backup | Track-Cluster Merge/Splitter (1/2)

- **Track-Cluster Merging:** implemented to address in pTDR need (cluster merging)
  - Algorithm outine based on ATLAS's split recovery procedure
    - c.f. Eur. Phys. J. C (2017) 77:466
  - Implemented in EICrecon#1406



ProtoClusters (output of Island Clustering)

Track-Cluster Merger

Merged ProtoClusters (input to Cluster Reco CoG)

- **The gist:**
  1) Match track projection to cluster
  2) If matched, calculate significance b/n E_{clust} energy & expected E_{dep}:

$$S(E_{clust}) = \frac{E_{clust} - (p_{proj} \times \langle E/p \rangle)}{\sigma(E_{dep})}$$

  3) If $S < S_{cut}$, add clusters inside Δr_{add}
  4) If multiple tracks pointing to merged cluster:
     3) Split into one cluster for each track & reweight transverse shape by p_{trk}, track projection

Parameters:

- $\langle E/p \rangle$: Average E/p
- $\sigma(E_{dep})$: Spread of dep. energy
- S_{cut}: Threshold to run split-recovery
- Δr_{add}: Window to add clusters
- σ_{trk}: scale for transverse shape reweighting

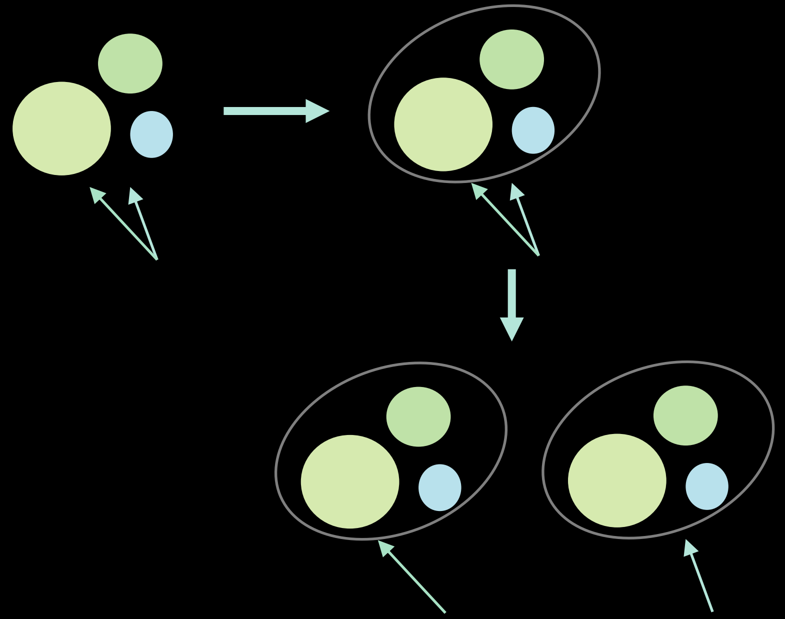
Backup | Track-Cluster Merge/Splitter (1/2)

- **Track-Cluster Merging:** implemented to address in pTDR need (cluster merging)
  - Algorithm outine based on ATLAS's split recovery procedure
    - c.f. Eur. Phys. J. C (2017) 77:466
  - Implemented in EICrecon#1406

- **The gist:**
  1) Match track projection to cluster
  2) If matched, calculate significance b/n E_{clust} energy & expected E_{dep}:

$$S(E_{clust}) = \frac{E_{clust} - (p_{proj} \times \langle E/p \rangle)}{\sigma(E_{dep})}$$

  3) If $S < S_{cut}$, add clusters inside Δr_{add}
  4) If multiple tracks pointing to merged cluster:
     3) Split into one cluster for each track & reweight transverse shape by p_{trk}, track projection



Parameters:
- $\langle E/p \rangle$: Average E/p
- $\sigma(E_{dep})$: Spread of dep. energy
- S_{cut}: Threshold to run split-recovery
- Δr_{add}: Window to add clusters
- σ_{trk}: scale for transverse shape reweighting

Backup | Mapping Old Onto New

Track-Cluster Matcher

1) Subtract projected E_{trk} from ECal, HCal clusters
   a) Identify seed (highest p_{trk}) track projection at inner face of ECal

PFA0

   b) Sum E_{trk} of all projections in R^{ECal}_{sum}, R^{HCal}_{sum} of seed
   c) Sum E_{clust} of all ECal, HCal clusters in R^{ECal}_{sum}, R^{HCal}_{sum} respectively

PFA1(a)

   d) If $\Sigma E^{ECal,HCal}_{trk} < \Sigma E^{ECal,HCal}_{clust}$
      i. Subtract $f^{ECal,HCal}_{trk} \times E^{ECal,HCal}_{trk}$ of nearest projection from each cluster
      ii. Pass subtracted clusters onto step 2
   e) Repeat 1(a) – 1(d)(ii) until all projections have been used

PFA2

2) Combine remaining ECal, HCal clusters into topo-clusters
   a) Combine nearby ECal, HCal clusters
      i. Identify seed (highest E_{clust}) ECal cluster
      ii. Merge all ECal, HCal clusters in R^{ECal}_{sum}, R^{HCal}_{sum} of seed
      iii. Repeat 2(a)(i) – 2(a)(iii) until no ECal clusters are left
   b) Combine remaining HCal clusters
      i. Identify seed HCal cluster
      ii. Add all HCal clusters in R^{HCal}_{sum} of seed
      iii. Repeat 2(b)(i) – 2(b)(iii) until no HCal clusters left
3) Return PFObjects

PFA1(b)/PFA3

○ **Note:** new approach *also* splits up PFA0 - 2 into separate calorimeters/eta regions