Particle Flow Status

Derek Anderson (ISU)

ePIC Collaboration Meeting

January 21st, 2025

Introduction | Some Context (1/3)

- Particle flow identified as a priority reco. task during the 1st joint Physics-S&C meeting
  - Has been on the task list since May 2023!
  - **Left:** Sal & Rosi's AC update at the 2023 EIC UGM
    - ☞ **Shows specific charge of task**

- **July 13th, 2023 Jet/HF meeting:** completed lit review and established plan to implement ***PFAlpha:***
  - ☞ A simple, bare bones PF algorithm to provide a baseline and spur further development
  - See slides here!

S&C Coordinators + Analysis Coordinators meeting
May 17th

… All right, we need to come together!

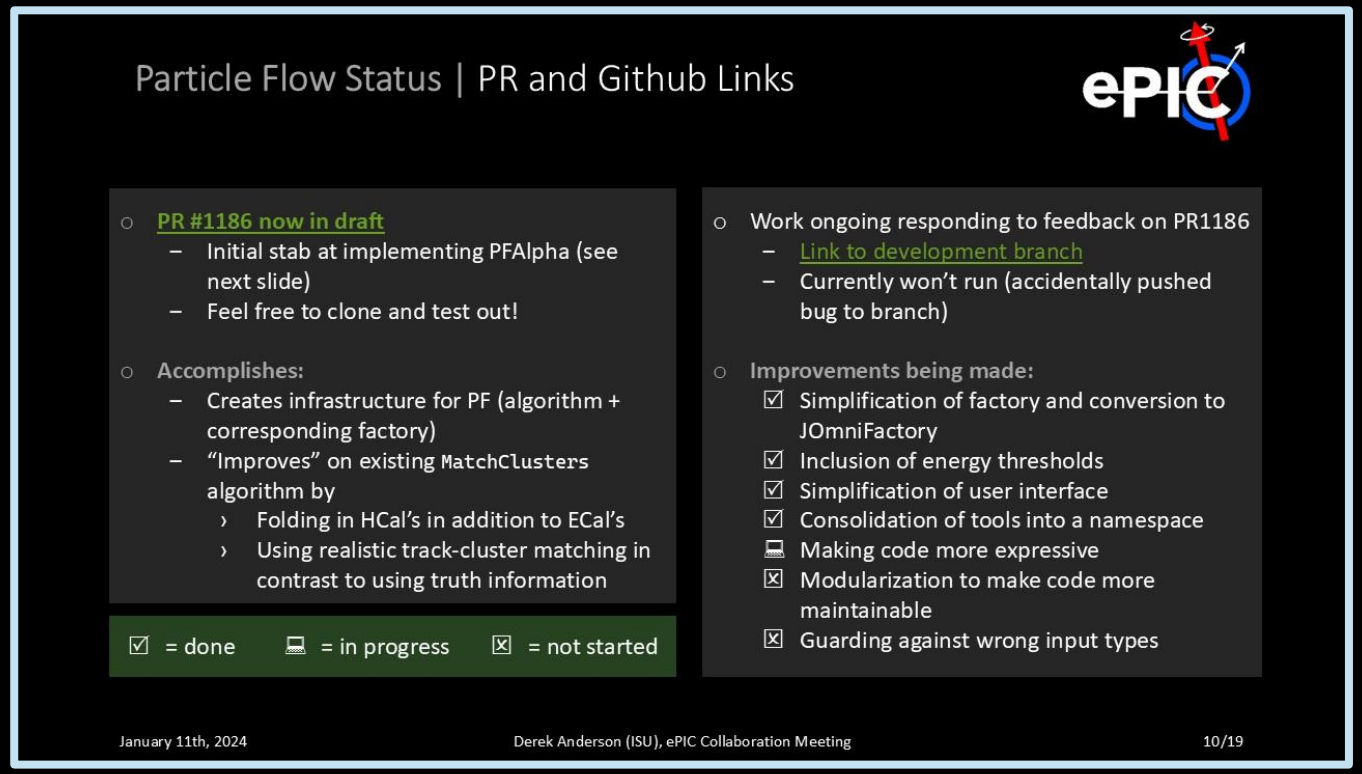
- Indico at: https://indico.bnl.gov/event/19473/
- Live notes at: Live Notes
- Identified **4 priorities**:
  - **Electron Finder**: Developing an efficient and accurate algorithm for identifying electrons and identifying the scattered electron of the DIS process
  - **Vertexing and PID**: Enhancing the vertexing capabilities and particle identification techniques to study heavy flavor physics
  - **Particle Flow**: Improving the jet reconstruction using particle flow information
  - **Low-Q^2**: Integration of the low-Q^2 tagger into the reconstruction framework for precise measurements of photo production and vector mesons

19

Introduction | Some Context (2/3)

- This time last year: had a functioning prototype in EICrecon#1186 (now closed)
  - **Left:** slide from my summary during Jet/HF workfest summarizing status & to-do's
  - Will discuss algorithm itself later

- **At that CM:** we decided that PF was ***not*** a priority for the pTDR
  - ∴ Task put on the back-burner until further notice
  - But still made progress on PF-related items while addressing pTDR needs (more later!)



Particle Flow Status | PR and Github Links

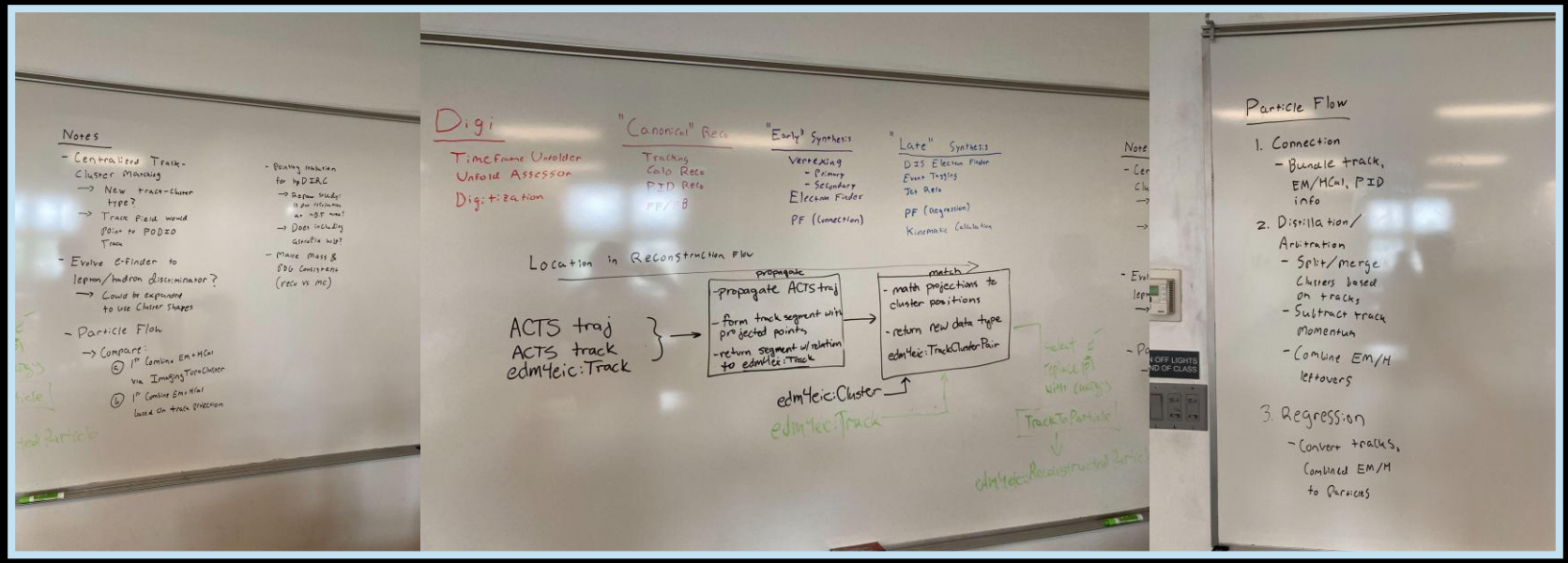
- PR #1186 now in draft
  - Initial stab at implementing PFAlpha (see next slide)
  - Feel free to clone and test out!
- Accomplishes:
  - Creates infrastructure for PF (algorithm + corresponding factory)
  - "Improves" on existing `MatchClusters` algorithm by
    - Folding in HCal's in addition to ECal's
    - Using realistic track-cluster matching in contrast to using truth information

☑ = done 💻 = in progress ☒ = not started

- Work ongoing responding to feedback on PR1186
  - Link to development branch
  - Currently won't run (accidentally pushed bug to branch)
- Improvements being made:
  - ☑ Simplification of factory and conversion to JOmniFactory
  - ☑ Inclusion of energy thresholds
  - ☑ Simplification of user interface
  - ☑ Consolidation of tools into a namespace
  - 💻 Making code more expressive
  - ☒ Modularization to make code more maintainable
  - ☒ Guarding against wrong input types

January 11th, 2024 Derek Anderson (ISU), ePIC Collaboration Meeting 10/19

Introduction | Some Context (3/3)



- **2024 EIC UGM:** very successful workfest focused on holistic reconstruction & eID!
  - Esp. productive discussion on ensuring modularity & synergy between eID and PF
    - ☞ Introduction of ***pseudoparticle*** concept (more later…)
  - See workfest slides and summary in links!
- Had several follow-up discussions in Reco WG & Weekly S&C meetings during fall semester
  - August 6th, 2024
  - August 7th, 2024
  - August 27th, 2024
  - September 16th, 2024
  - October 14th, 2024
  - October 28th, 2024
  - October 30th, 2024

Interstitial Developments | Initial Stab

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

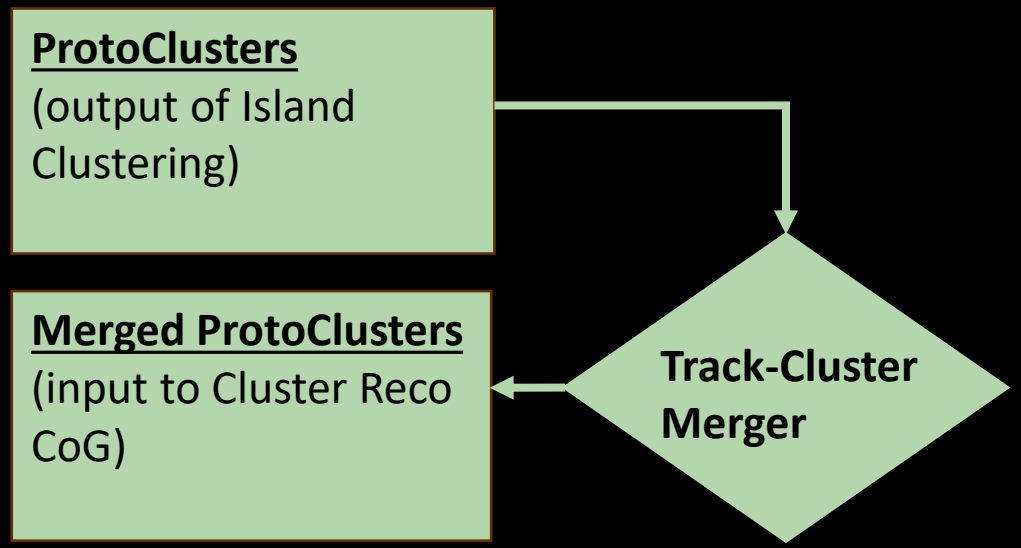
Interstitial Developments | Track-Cluster Merge/Splitter

- **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}:

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

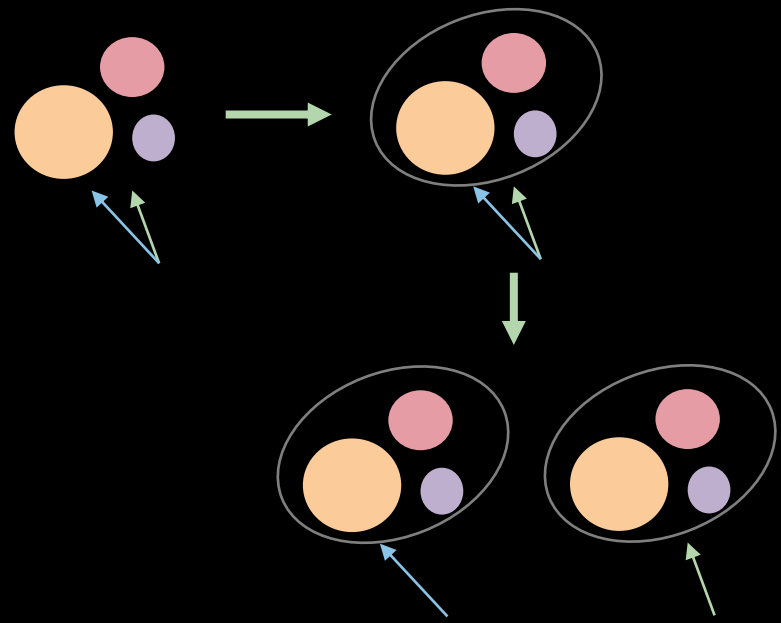
  3) If $S < S_{cut}$, add clusters inside Δr_{add}
  4) If multiple tracks pointing to merged cluster:
     3) Split into one cluster for each track & reweight transverse shape by p_{trk}, track projection



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

Interstitial Developments | Track-Cluster Merge/Splitter

- **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

Interstitial Developments | Candidate Particle Types

```
## A charged particle candidate
edm4eic::ChargedRecoParticleCandidate:
  Description: "Candidate charged reconstructed particle"
  Author: Tyler Kutz, Derek Anderson, Shujie Li
  OneToOneRelations:
    - edm4eic::Track track  // reconstructed track...other relations are matched to this
  OneToManyRelations:
    - edm4hep::ParticleID particleIDs  // associated particle IDs
    - edm4eic::Cluster    ecalClusters // ECAL clusters matched to this track
    - edm4eic::Cluster    hcalClusters // HCAL clusters matched to this track
  VectorMembers:
    - float ecalWeights  // weights of matched ecal clusters
    - float hcalWeights  // weights of matched hcal clusters
```

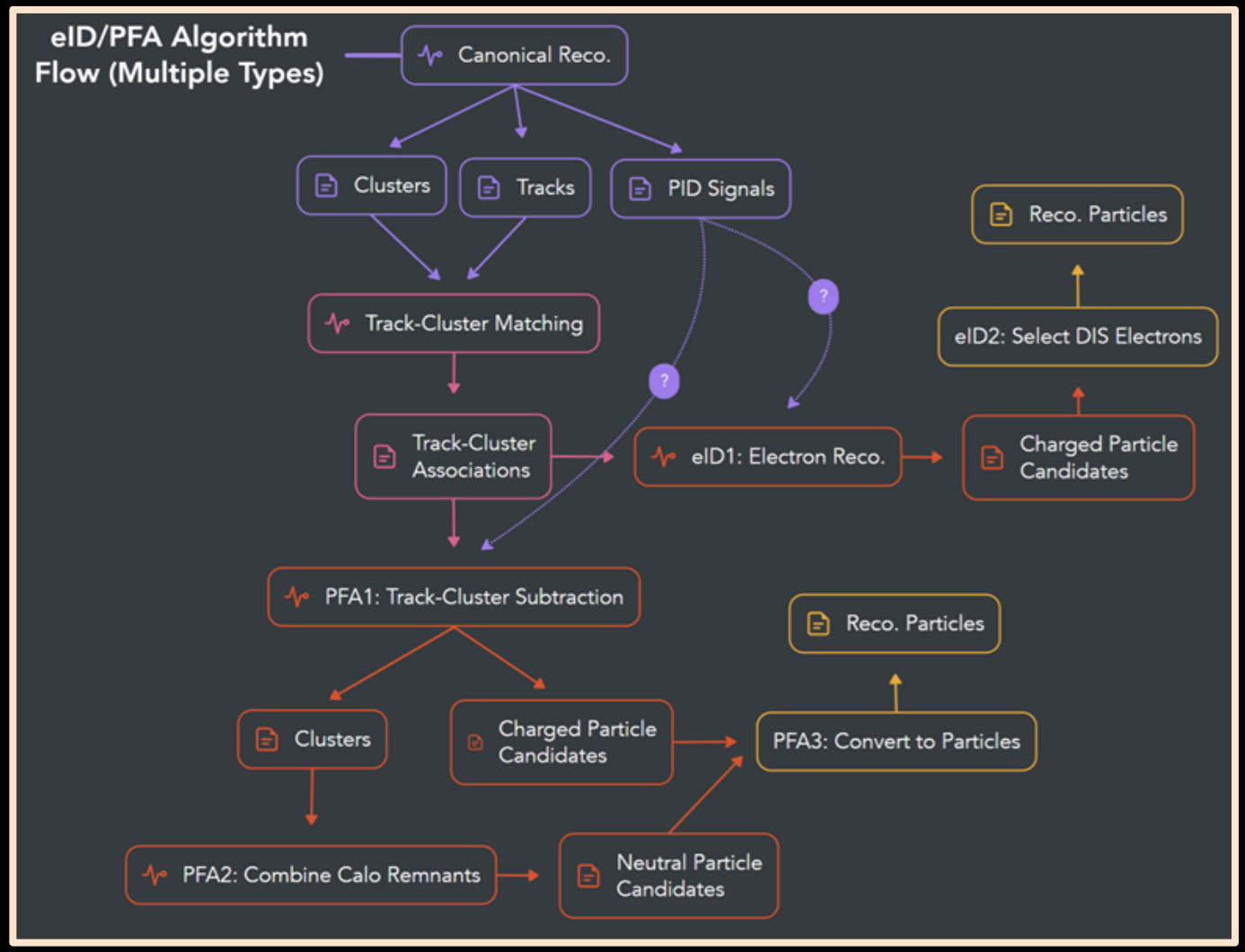
```
## A neutral particle candidate
edm4eic::NeutralRecoParticleCandidate:
  Description: "Candidate neutral reconstructed particle"
  Author: Tyler Kutz, Derek Anderson, Shujie Li
  OneToManyRelations:
    - edm4hep::ParticleID particleIDs  // associated particle IDs
    - edm4eic::Cluster    ecalClusters // associated ECAL clusters
    - edm4eic::Cluster    hcalClusters // associated HCAL clusters
  VectorMembers:
    - float ecalWeights  // weights of associated ecal clusters
    - float hcalWeights  // weights of associated hcal clusters
```

- **Critical Idea from 2024 UGM Workfest:** a *pseudoparticle/candidate particle* type
  - In spirit, similar to a protocluster *but* for reco. particles
  - Brings together needed track + clusters with weights ahead of final reconstruction step
- This interface will help keep PFAs and eID modular
  - And -- down the road -- facilitate more tightly integrating the two workflows
  - e.g. both utilizing the same candidate → reco particle algorithm

Modular Approach | Overview

- **Left:** diagram to illustrate topology of PF and eID algorithms *as of October 30th, 2024*
  - It integrates discussions had during 2024 EIC UGM and fall Reco WG meetings

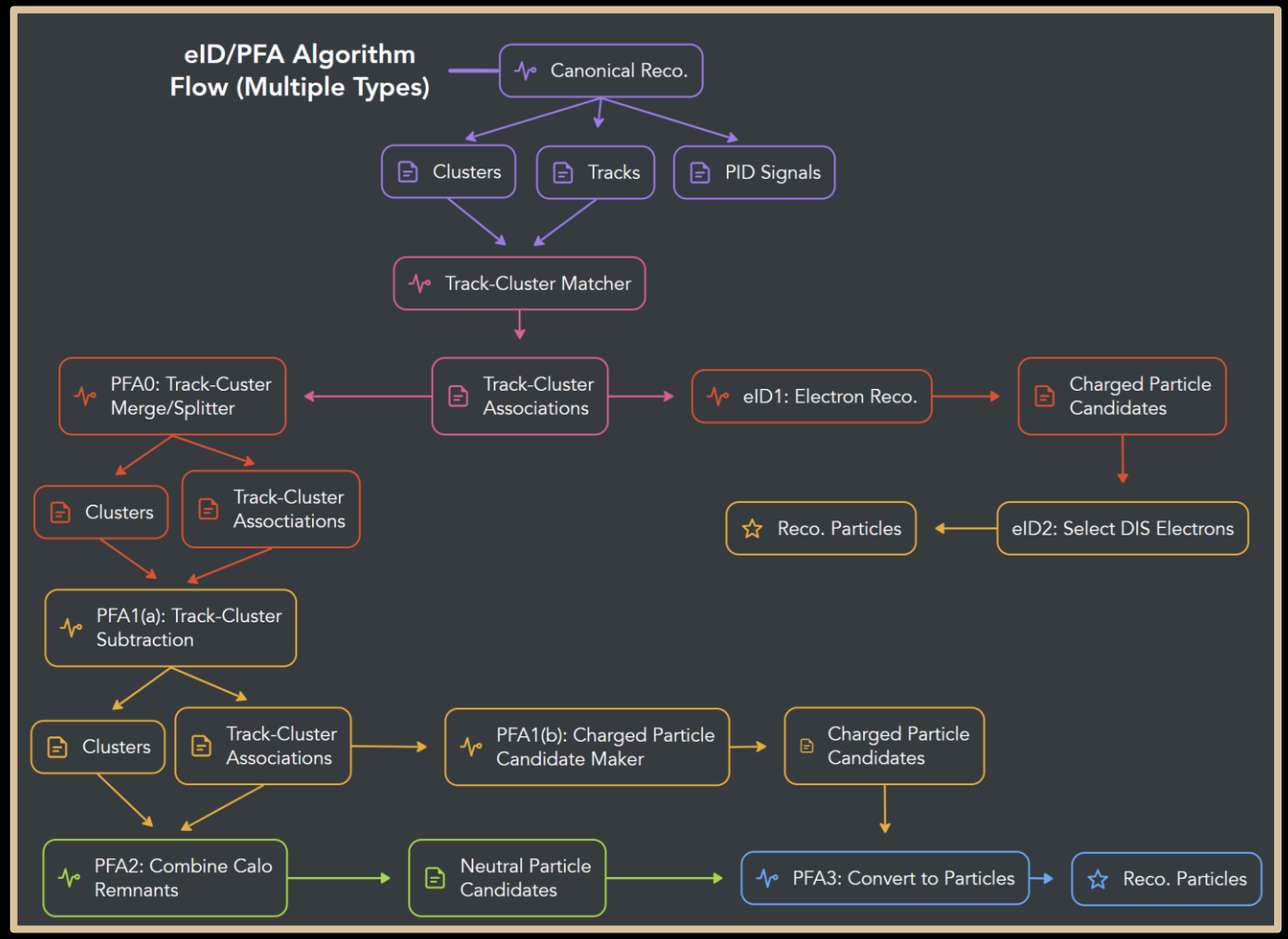
- **This approach helps keeps over PFA modular!**
  - ☞ Each step of old/monolithic PFAlpha is now separated into its own algorithm



Modular Approach | Recent Changes

- **Plan has evolved since then!**
  - 2 new changes:
    1) Merge/Splitter now integrated into topology
    2) PFA1 split into 2 algorithms
  - Latter change provides a track-cluster subtraction algorithm for use everywhere!

- **Working out details in a dedicated development branch:**
  - ☞ EICrecon branch here, edm4eic branch here
  - Will roll-out each algorithm in a series of PRs



Modular Approach | 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
   b) Sum E_{trk} of all projections in R_{sum}^{ECal}, R_{sum}^{HCal} of seed — **PFA0**
   c) Sum E_{clust} of all ECal, HCal clusters in R_{sum}^{ECal}, R_{sum}^{HCal} respectively
   d) If $\Sigma E_{trk}^{ECal,HCal} < \Sigma E_{clust}^{ECal,HCal}$ — **PFA1(a)**
      i. Subtract $f_{trk}^{ECal,HCal} \times E_{trk}^{ECal,HCal}$ 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_{sum}^{ECal}, R_{sum}^{HCal} 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_{sum}^{HCal} 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

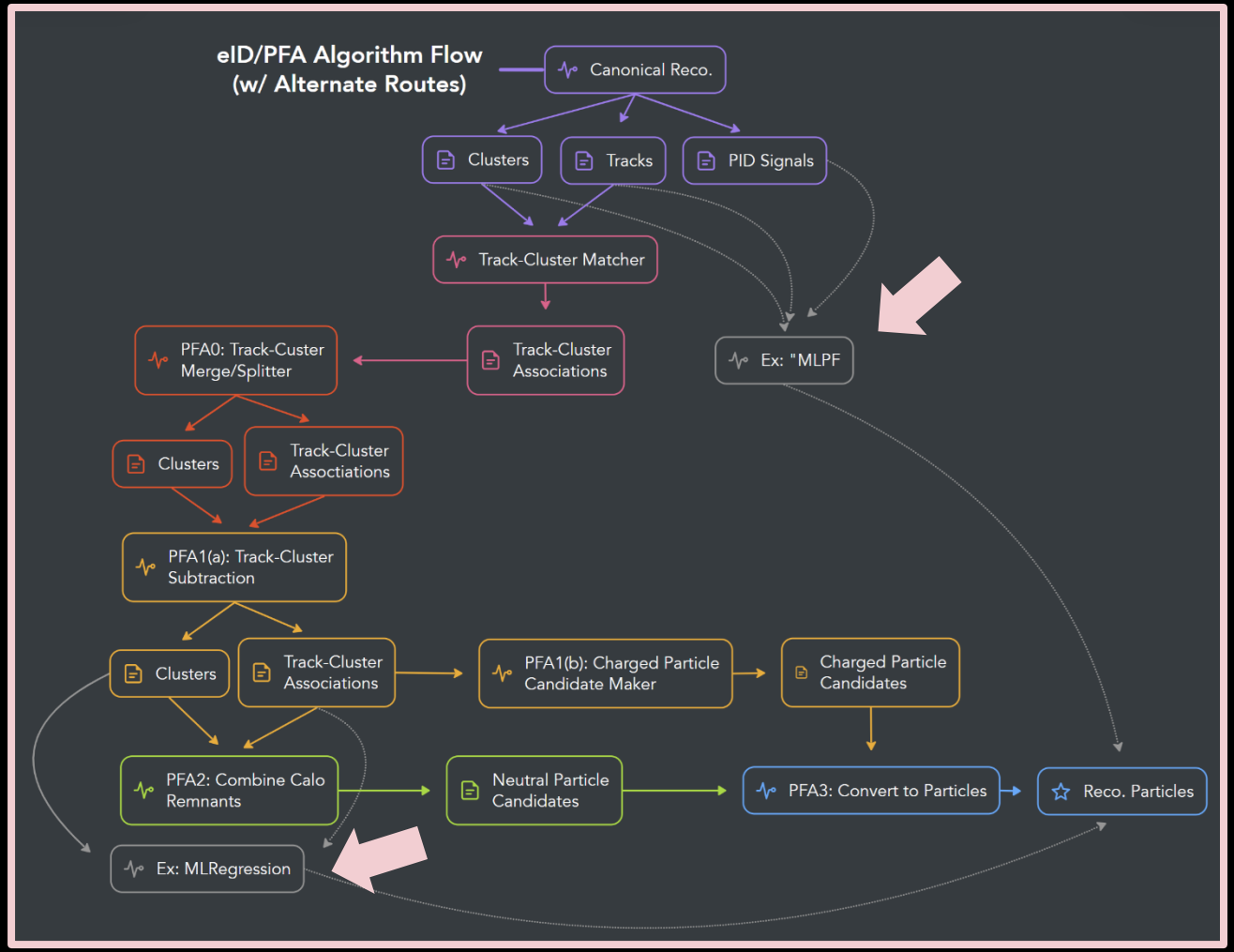
Modular Approach | PRs, Issues, and More

Item	PR/Issue	Assignee	Target	Notes
Track-Cluster Matcher	EICrecon#1694	Tristan P. (Lehigh)	TBD	
PFA 0: Update Merge/Splitter	EICrecon#1699	Derek A. (ISU)	Mid-Feb.	In dev branch
PFA 1(a): Track-Cluster Subtractor	EICrecon#1627	Derek A. (ISU)	Mid-Feb.	In dev branch
Charged/Neutral Particles	EDM4eic#97	Derek A. (ISU)	March	In dev branch
PFA 1(b): Charged Candidate Converter	To-Do	Derek A. (ISU)	March	
PFA 2: Remnant Combiner	To-Do	Derek A. (ISU)	April	
PFA 3: Particle Converter	To-Do	Derek A. (ISU)	April	
Cross-Calo Topocluster Maker	EICrecon#1561	Tristan P. (Lehigh)	TBD	

Looking Forward | How About Other Algorithms?

- Straightforward to hire in alternate/additional PFAs
  - For example:
    1) add an ML model to do regression from calo remnants to reco. particles
    2) Or try more “end-to-end”” ML algorithm similar to MLPF (EPJ C 81, 381 (2021))
  - Examples are in light grey boxes

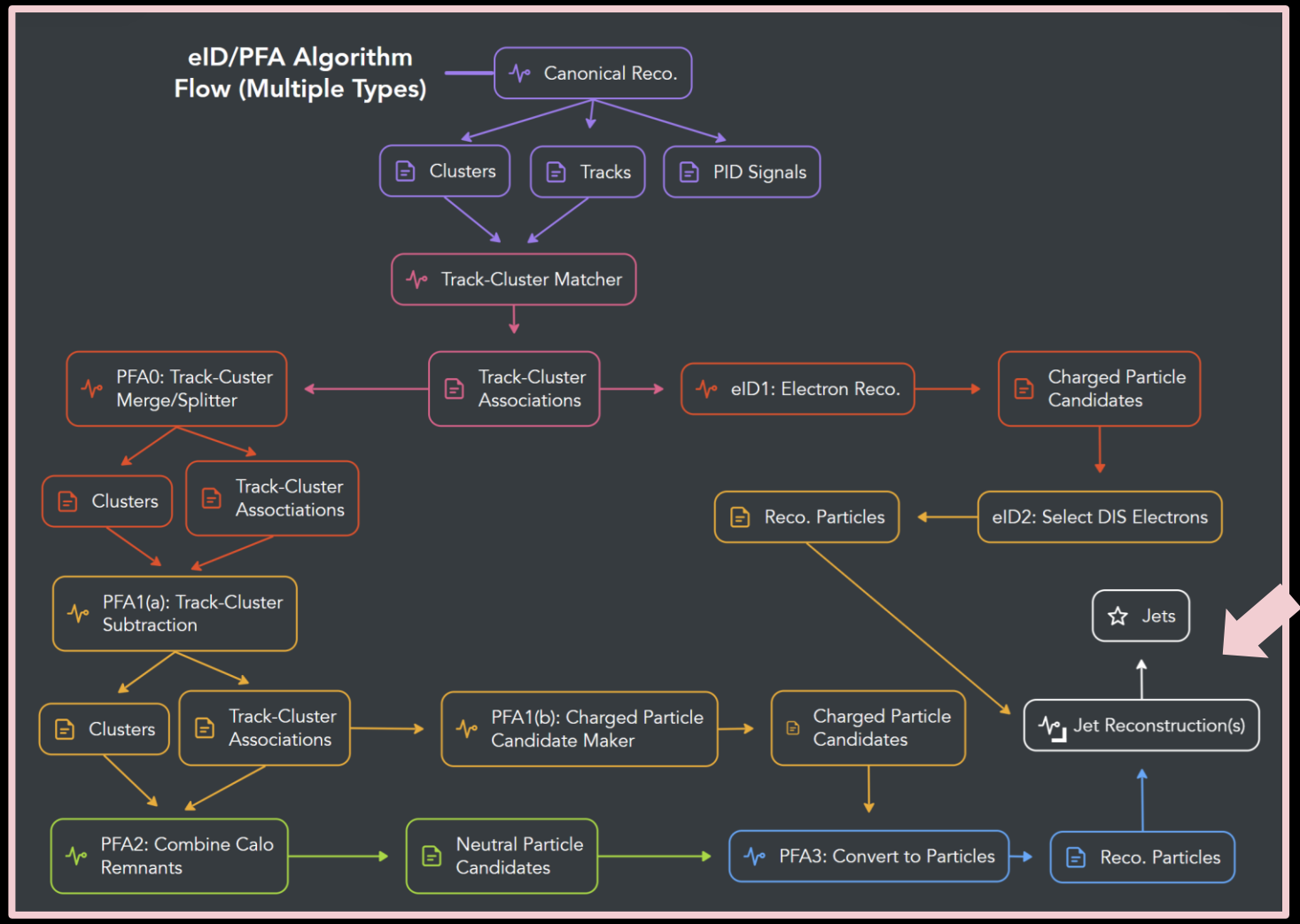
☞ (OFC, can also do things *other* than ML algorithms!)



Looking Forward | Interface to Jets

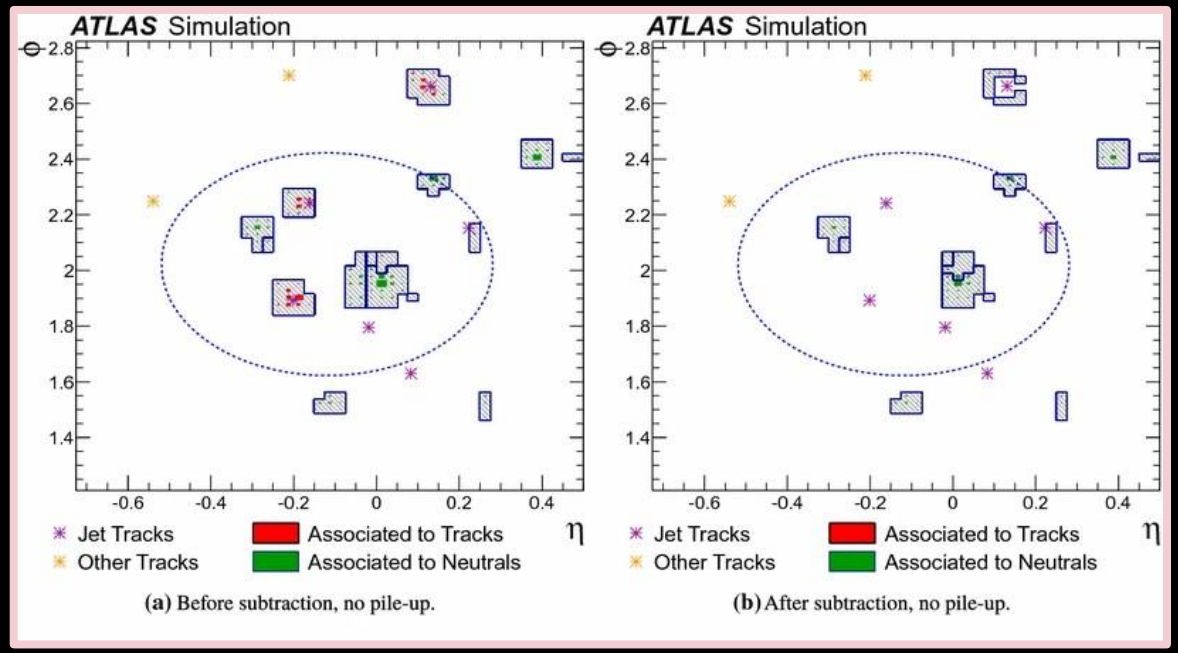
- **Also straightforward to interface PFAlpha with jet reconstruction!**
  - Jet reco already ingests Reco Particles
  - And planned jet type intentionally only has relations to Reco Particles
    - ☞ (See here)

- **Note:** also included lines going from eID into jet reco
  - ☞ Will need its output for both Centauro and normal jet algorithms at the reco level



Looking Forward | Longer-Term To-Do's

- **To-Do:** some utility macros to generate to generate plots like from ATLAS
  - Would be HUGE help in debugging
- **To-Do:** PF benchmarks
  - Particle energy-scale and resolution (PES/R) natural choices
  - Maybe "misidentification rate"?
- **To-Do:** benchmarks also could be valuable in optimizing PF parameters and other reconstruction parameters…



ATLAS [arXiv:1703.10485]

Backup

Backup | Detailed Breakdown of Algorithm

1) **Subtract projected E_{trk} from ECal, HCal clusters**
   a) Identify seed (highest p_{trk}) track projection at inner face of ECal
   b) Sum E_{trk} of all projections in R_{sum}^{ECal}, R_{sum}^{HCal} of seed
   c) Sum E_{clust} of all ECal, HCal clusters in R_{sum}^{ECal}, R_{sum}^{HCal} respectively
   d) If $\Sigma E_{trk}^{ECal,HCal} < \Sigma E_{clust}^{ECal,HCal}$
      i. Subtract $f_{trk}^{ECal,HCal} \times E_{trk}^{ECal,HCal}$ 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

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_{sum}^{ECal}, R_{sum}^{HCal} 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_{sum}^{HCal} of seed
      iii. Repeat 2(b)(i) – 2(b)(iii) until no HCal clusters left
3) **Return PFObjects**