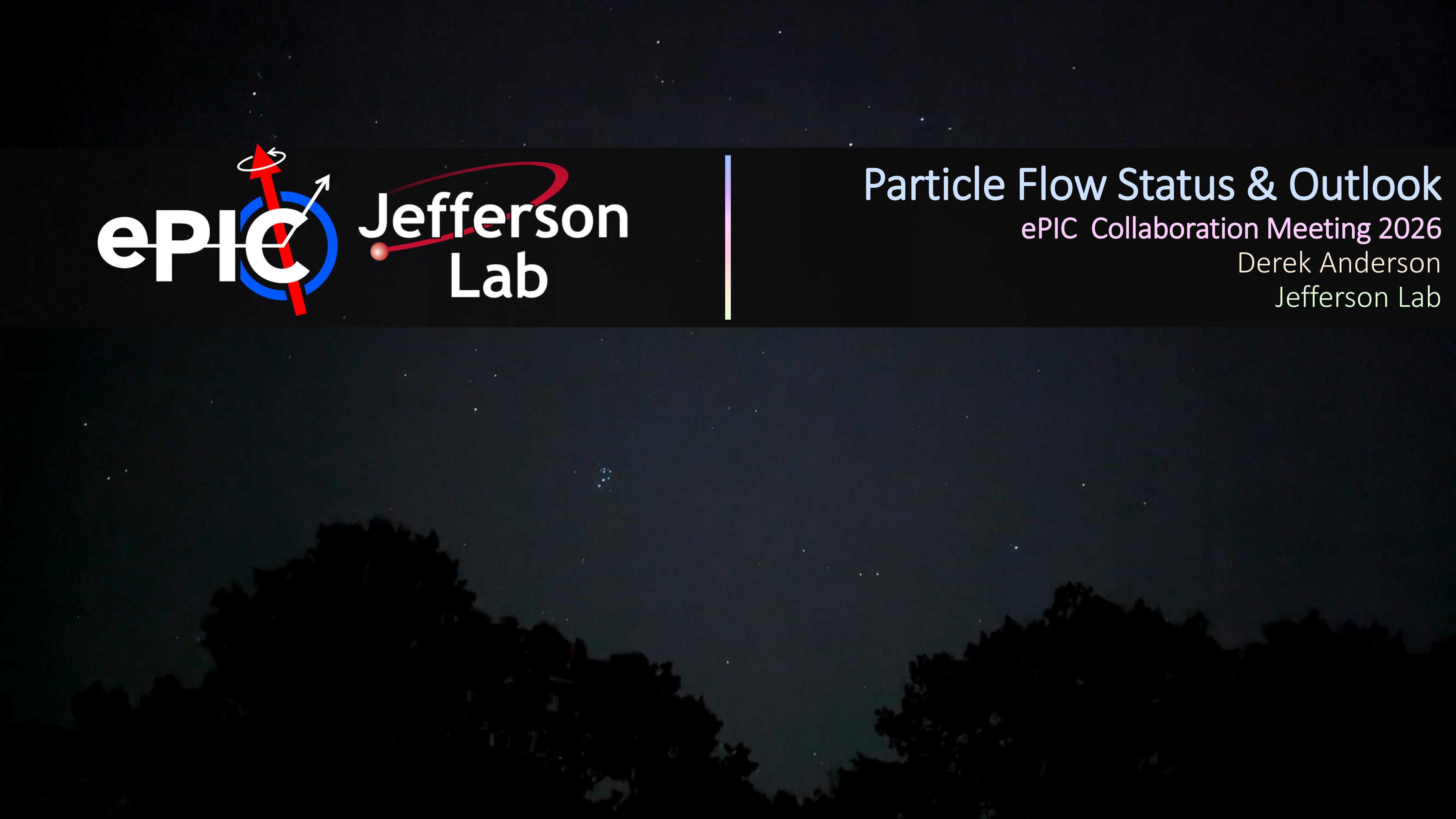


Particle Flow Status & Outlook

ePIC Collaboration Meeting 2026

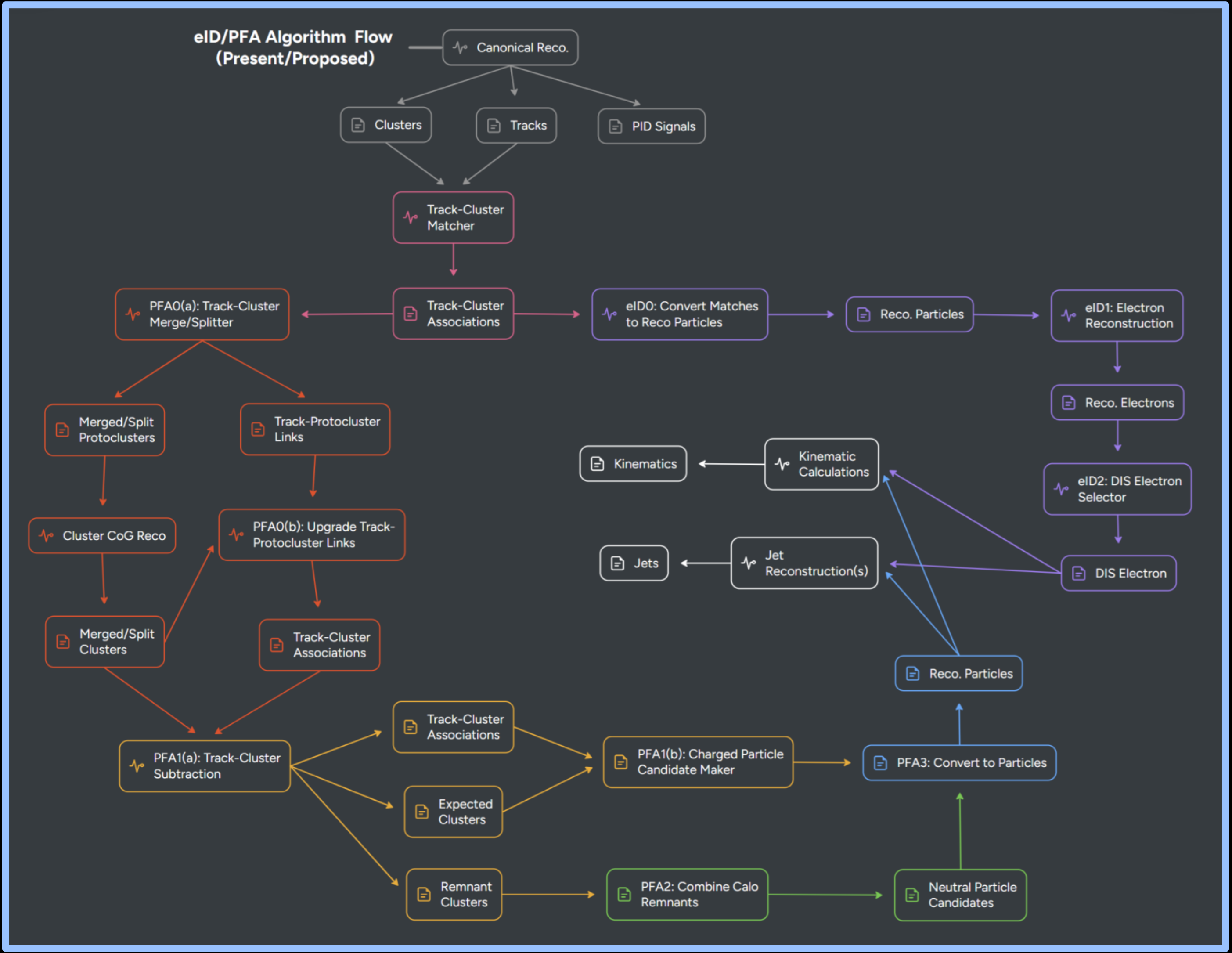
Derek Anderson
Jefferson Lab

Introduction | Particle Flow Priority Task

○ **PF Reconstruction Task:** improve jet reconstruction using particle flow (PF) info
- Also touches many aspects of holistic reco beyond jets
- eg. Neutral reco benefits greatly from PF techniques

⇒ **Task Goal:** implementation of ***PFAlpha***, a (relatively) simple PF baseline to gauge further developments
- **Right:** schematic of algorithm flow of PFAlpha
- Aiming to ensure modularity of overall algorithm

○ **Note:** PFAX = "PFAlpha stage X"
- Flow is split up into major stages of the overall algorithm



eID/PFA Algorithm Flow (Present/Proposed)

Canonical Reco.

Clusters

Tracks

PID Signals

Track-Cluster Matcher

PFA0(a): Track-Cluster Merge/Splitter

Track-Cluster Associations

eID0: Convert Matches to Reco Particles

Reco. Particles

eID1: Electron Reconstruction

Merged/Split Protoclusters

Track-Protocluster Links

Reco. Electrons

Kinematics

Kinematic Calculations

eID2: DIS Electron Selector

Cluster CoG Reco

PFA0(b): Upgrade Track-Protocluster Links

Jets

Jet Reconstruction(s)

DIS Electron

Merged/Split Clusters

Track-Cluster Associations

Reco. Particles

Track-Cluster Associations

PFA1(a): Track-Cluster Subtraction

PFA1(b): Charged Particle Candidate Maker

PFA3: Convert to Particles

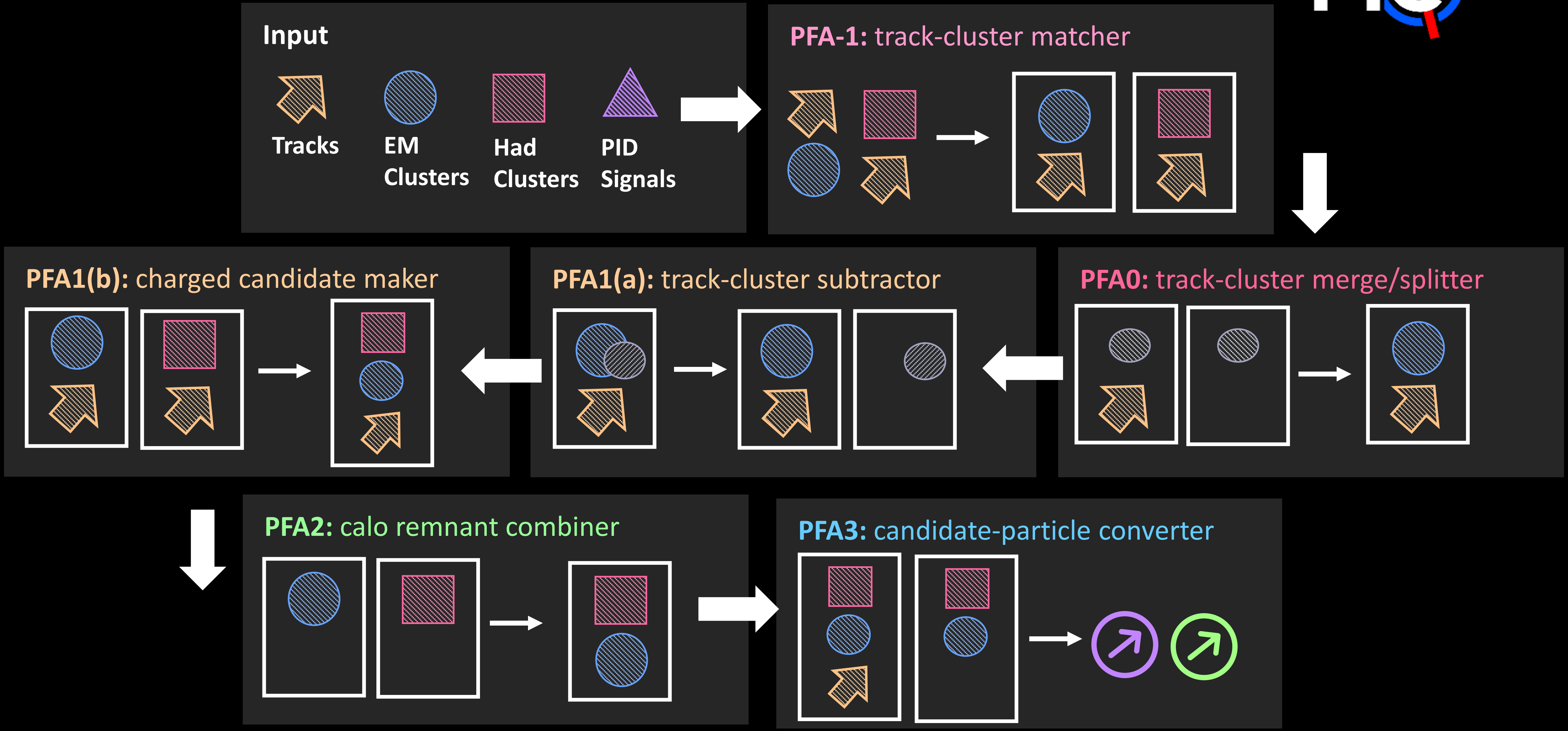
Expected Clusters

Remnant Clusters

PFA2: Combine Calo Remnants

Neutral Particle Candidates

Introduction | Baseline Overview



Input

Tracks

EM Clusters

Had Clusters

PID Signals

PFA-1: track-cluster matcher

PFA0: track-cluster merge/splitter

PFA1(a): track-cluster subtractor

PFA1(b): charged candidate maker

PFA2: calo remnant combiner

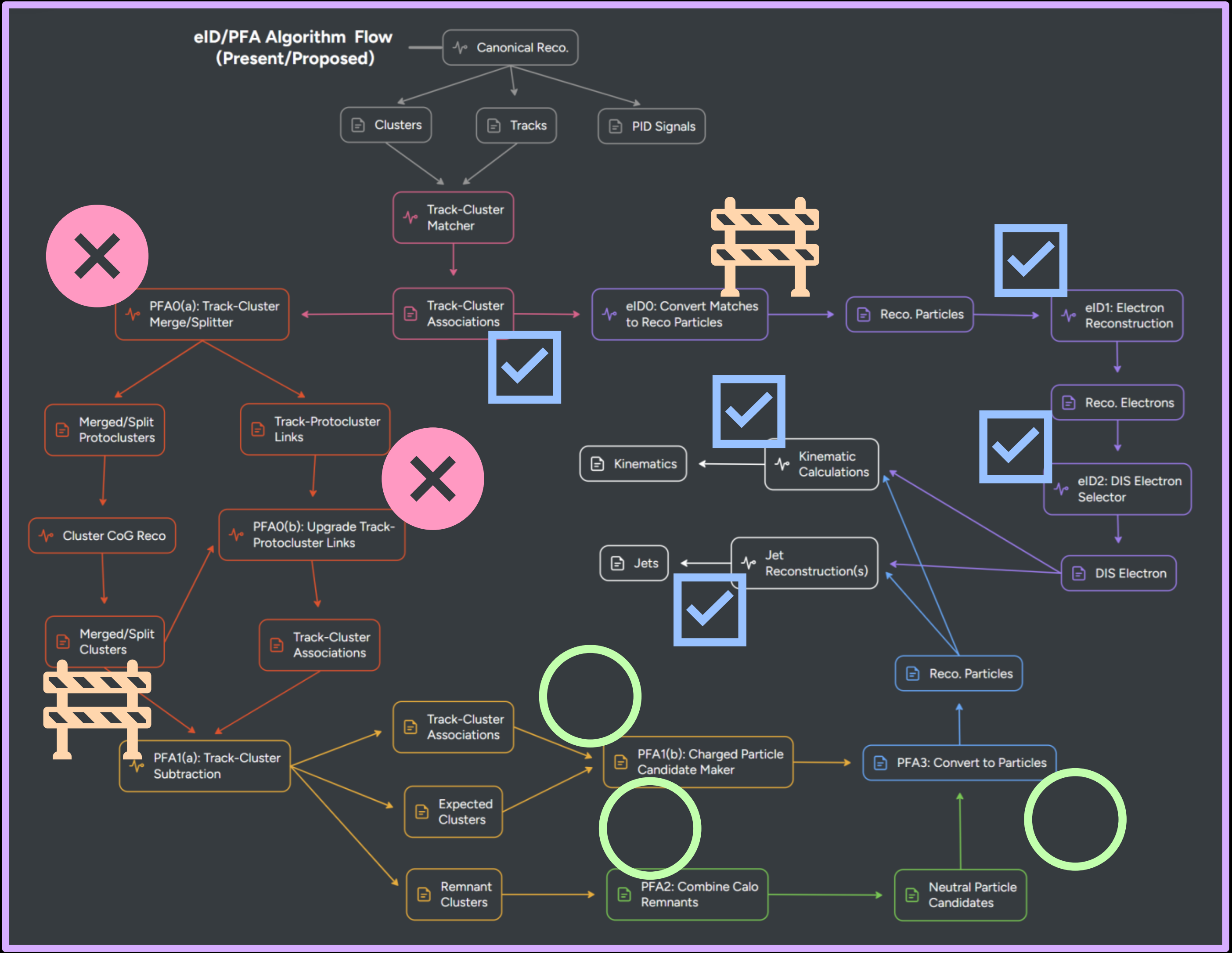
PFA3: candidate-particle converter

PF Status | Status as of Physics Readiness WS

- **PFA0 stages were blocked**
  - Introduction of track-protocluster links exposed bug in our JANA2 extensions
  - Required patch from JANA2 2.4.3 (which had breaking changes)
  - But substantial work done on PFA0(a) already…

- **PFA1(a) was in progress**
  - PFA1(b), PFA2, & PFA3 stages all were to-do...

○ = To-do
🚧 = In progress
☑ = Done/already in EICrecon
☒ = Blocked



eID/PFA Algorithm Flow (Present/Proposed)

Canonical Reco.

Clusters

Tracks

PID Signals

Track-Cluster Matcher

Track-Cluster Associations

PFA0(a): Track-Cluster Merge/Splitter

eID0: Convert Matches to Reco Particles

Reco. Particles

eID1: Electron Reconstruction

Merged/Split Protoclusters

Track-Protocluster Links

Reco. Electrons

Kinematics

Kinematic Calculations

eID2: DIS Electron Selector

Cluster CoG Reco

PFA0(b): Upgrade Track-Protocluster Links

Jets

Jet Reconstruction(s)

DIS Electron

Merged/Split Clusters

Track-Cluster Associations

Reco. Particles

Track-Cluster Associations

PFA1(a): Track-Cluster Subtraction

PFA1(b): Charged Particle Candidate Maker

PFA3: Convert to Particles

Expected Clusters

Remnant Clusters

PFA2: Combine Calo Remnants

Neutral Particle Candidates

PF Status | As of Today

- **PFA0 now unblocked!**
  - Introduced track-cluster associations, EDM4eic#130
  - Resolved bug with patch for JANA2 bug, eic-spack#794

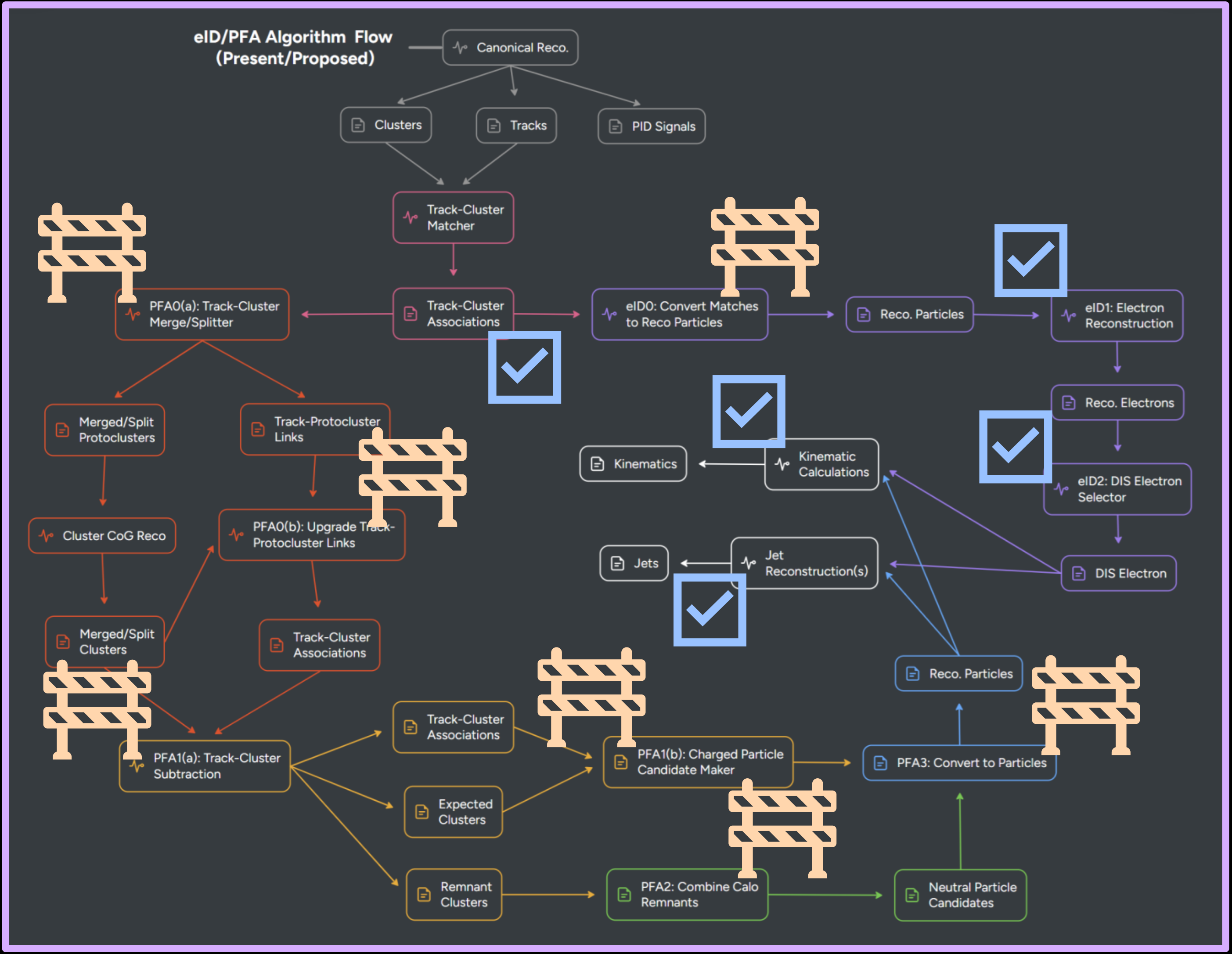
- Now ***ALL*** stages have PRs ready for review or are in progress!
  - ☞ Huge thanks to **Subhadip Pal (CTU)** & **Esteban Molina (UMich)** for helping out!

○ = To-do

🚧 = In progress

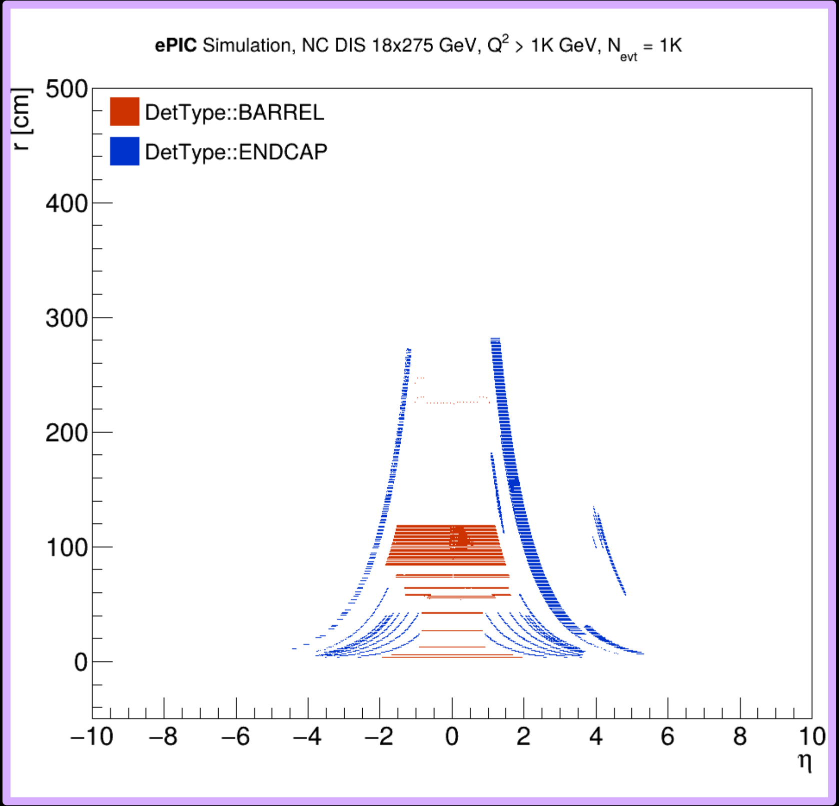
☑ = Done/already in EICrecon

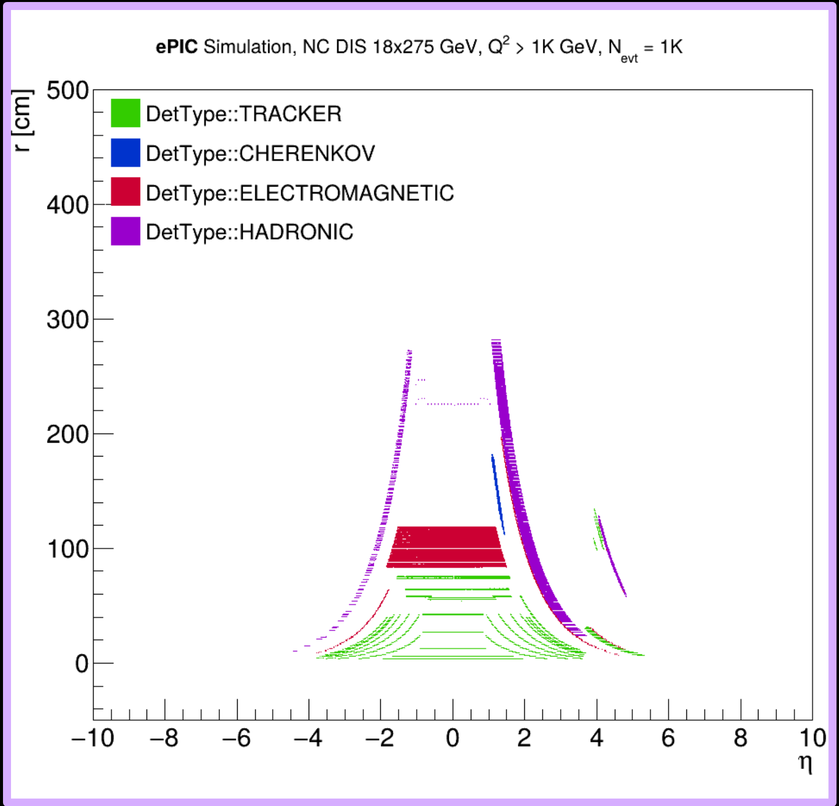
⊗ = Blocked



eID/PFA Algorithm Flow (Present/Proposed)
Canonical Reco.
Clusters
Tracks
PID Signals
Track-Cluster Matcher
Track-Cluster Associations
PFA0(a): Track-Cluster Merge/Splitter
eID0: Convert Matches to Reco Particles
Reco. Particles
eID1: Electron Reconstruction
Reco. Electrons
eID2: DIS Electron Selector
DIS Electron
Merged/Split Protoclusters
Track-Protocluster Links
Kinematics
Kinematic Calculations
Cluster CoG Reco
PFA0(b): Upgrade Track-Protocluster Links
Jets
Jet Reconstruction(s)
Merged/Split Clusters
Track-Cluster Associations
Reco. Particles
PFA1(a): Track-Cluster Subtraction
Track-Cluster Associations
PFA1(b): Charged Particle Candidate Maker
PFA3: Convert to Particles
Expected Clusters
Remnant Clusters
PFA2: Combine Calo Remnants
Neutral Particle Candidates

PF Status | EMCal vs. HCal





- **Not listed on previous slide:** need to distinguish EMCal vs. HCal clusters *generically*
  - DD4hep has tools to do this, via DetTypes
  - epic#994 applies missing types to (most) subsystems

- Example of how to extract DetTypes in analysis found here in snippets

- **Above:** DetTypes for hits in 18x275 NC DIS events

PF Status | Development Task Details

Task	Description	Issue/PR	Notes	Assignees
PFA-1	Deprecate MatchClusters, replace w/ pure reco equivalent	EICrecon#1956		Tristan
PFA0(a)	Complete merge/splitter update	EICrecon#1699	PR open	Derek
PFA0(b)	Implement track-protocluster link promotion algorithm	EICrecon#2293	Needs unit test	Derek
PFA1(a)	Revive and finish track-cluster subtractor	EICrecon#1627	PR open	Derek
PFA1(b)	Track-cluster converter (synergy w/ PFA-1)	EICrecon#2124	PR open	Derek
EDM0	Flagging ecal vs. hcal clusters	epic#994	PR open	Derek
PFA2	Implement calo remnant combiner	EICrecon#2195	PR open	Subhadip
PFA3	Implement particle regressor/convertor	EICrecon#2130		Esteban

- ○ Missed goal to have implementation in by CM. But on track to have in by end of **CY26.Q1**
  - – After individual stages merged, need PR to tie all together into 1 workflow

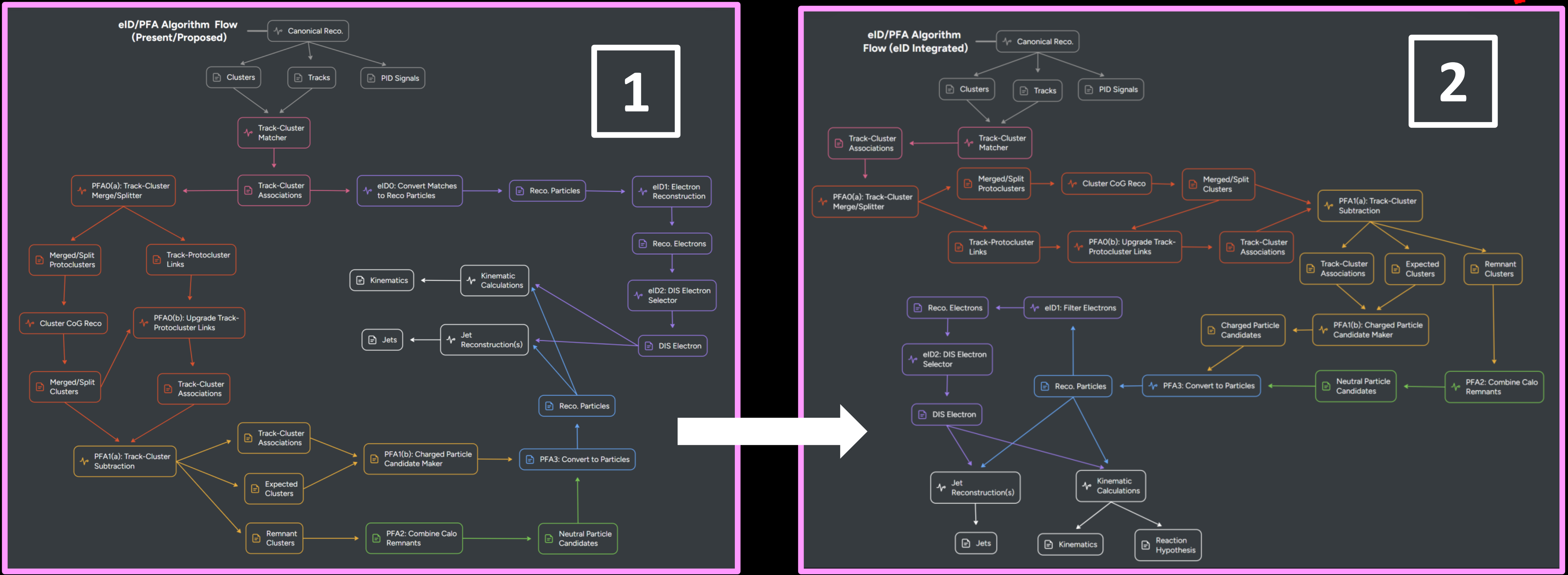
- ○ **But:** tuning & benchmarking *still needed for each stage*
  - ☞ This is something **WE STILL NEED HELP WITH**
  - – (Task details in backup)

Electron Finder: Proposed Development Plan

ePIC Collaboration Meeting 2026

(Inclusive PWG) Stephen, Win
(Reco WG) Chandra, Derek, Shujie

eID Tasks | Context and Milestones


eID/PFA Algorithm Flow (Present/Proposed)
1
Canonical Reco.
Clusters
Tracks
PID Signals
Track-Cluster Matcher
PFA0(a): Track-Cluster Merge/Splitter
Track-Cluster Associations
eID0: Convert Matches to Reco Particles
Reco. Particles
eID1: Electron Reconstruction
Reco. Electrons
Merged/Split Protoclusters
Track-Protocluster Links
Kinematics
Kinematic Calculations
eID2: DIS Electron Selector
Cluster CoG Reco
PFA0(b): Upgrade Track-Protocluster Links
Jets
Jet Reconstruction(s)
DIS Electron
Merged/Split Clusters
Track-Cluster Associations
Reco. Particles
Track-Cluster Associations
PFA1(a): Track-Cluster Subtraction
PFA1(b): Charged Particle Candidate Maker
PFA3: Convert to Particles
Expected Clusters
Remnant Clusters
PFA2: Combine Calo Remnants
Neutral Particle Candidates
eID/PFA Algorithm Flow (eID Integrated)
2
Canonical Reco.
Clusters
Tracks
PID Signals
Track-Cluster Associations
Track-Cluster Matcher
Merged/Split Protoclusters
Cluster CoG Reco
Merged/Split Clusters
PFA0(a): Track-Cluster Merge/Splitter
PFA1(a): Track-Cluster Subtraction
Track-Protocluster Links
PFA0(b): Upgrade Track-Protocluster Links
Track-Cluster Associations
Track-Cluster Associations
Expected Clusters
Remnant Clusters
Reco. Electrons
eID1: Filter Electrons
Charged Particle Candidates
PFA1(b): Charged Particle Candidate Maker
eID2: DIS Electron Selector
Reco. Particles
PFA3: Convert to Particles
Neutral Particle Candidates
PFA2: Combine Calo Remnants
DIS Electron
Jet Reconstruction(s)
Kinematic Calculations
Jets
Kinematics
Reaction Hypothesis

- **Building on Physics Readiness WS discussion:** inclusive+reco convenors met to start planning work on EICrecon e$^-$ finder
  - ☞ **Goal:** propagate vetted prototype developments, integrate w/ PF

- Identified initial task list, broad strokes of program, and 2 milestones:
  1) Existing eID workflow w/o truth info + prototptype developments
  2) Integration w/ PF

eID Tasks | Toward Milestone 1

Task	Description	Notes	Assignees
eID0	Resolve missing EMCal-track associations		Barak, Win, **Help**
eID1	Resolve issues with boost.h	EICrecon#2331	Win
eID2	Add isolation cut to ElectronReconstruction	EICrecon#2332	**Help**
eID3/PFA-1	Deprecate MatchClusters, replace with pure reco equivalent	EICrecon#1956	Tristan
eID4	Wire reco DIS electrons into kinematic calculations (step 1)	EICrecon#2333	**Help**

Milestone 1: there are well-defined steps to bring EICrecon e$^-$-finder to next level, meaning that

1. Updates are integrated from Inclusive PWG's prototype code, and
2. Use of truth information is removed.

ETA: end of CY26.Q1 ***(cond. on workforce)***

- **Help** indicates where additional workforce is needed
- Tasks are decoupled and can proceed independently
- **eID4** has synergy with **Event Kinematics** reconstruction priority
- **Note:** reco kinematics *do not* have to be made default until collaboration is ready

eID Tasks | Toward Milestone 2

Task	Description	Notes	Assignees
eID5	Wire reco DIS electrons into kinematic calculations (step 2)	EICrecon#2333	**Help**
eID6	Improve DIS selection (adding p_T ranking, etc.)		TBD
eID7	Improve kinematic calculations (handling different beams)		TBD
eID8/PFA4	Integrating PFAlpha + eID		TBD

Milestone 2: tasks still require elaboration. But can identify high-level steps towards next major milestone, which will be partly defined by integrating PF and eID.

ETA: end of CY26.Q3 ***(again cond. on workforce)***

- **TBD** indicates task needs elaboration
- **eID7** has synergy with **Event Kinematics** reconstruction priority
- These tasks might be better done sequentially to assess impact
- **Note:** PFAlpha aiming for implementation + tuning by end of CY26.Q1

eID Tasks | Longer Term

- Several longer term topics to think about:
  - **Handling multiple HFS***
    - Multiple e^- candidates means multiple HFS need to be checked
  - **Handling multiple eID algorithms***
    - Each will have a HFS + kinematic set attached
    - Doing calculations in EICrecon can help with combinatorics downstream
  - **Algorithm to combine multiple kinematics, HFS***
    - Ideally would have algorithm to integrate over combinatorics, provide default values
  - **Standardized benchmarks + assessing backgrounds**
    - Critical for performance eval.

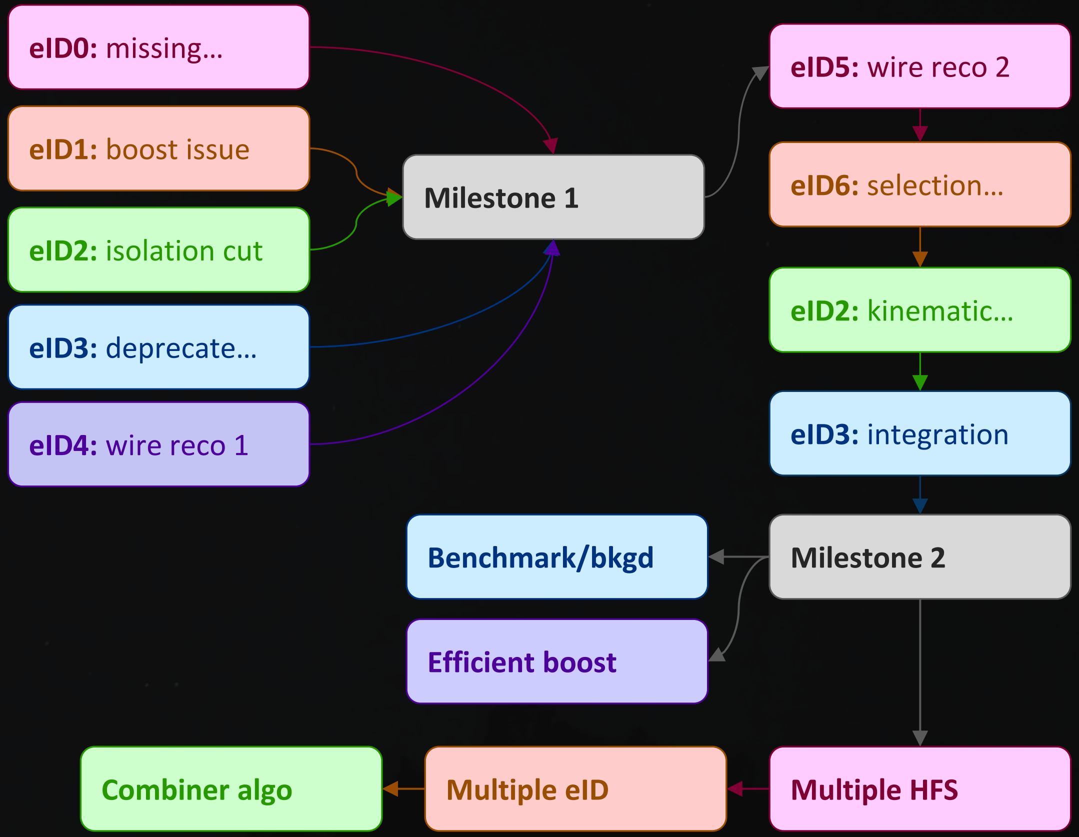
- Long term (cont.)
  - **More efficient boost calculation**
    - Boost to CoM really only needs to be calculated once per beam setting
  - **Settling boundary between reconstruction, analysis**
    - Broader question which touches on every PWG
    - Where do analysis tools like RAD fit in?

- **Lastly:** e^- finder is important *for everyone,* so *everyone* is open to help here!
  - These tasks are a great way to learn about both DIS physics and our software framework!

* **Note:** synergy w/ event reco priority

Summary

- ○ **Particle Flow**
  - – Made strong progress between Physics Readiness WS and now
    - › Implementation by end CY26.Q1 feasible
  - – But tuning + benchmarking still to be done
    - › ***We need help with this!!***
- ○ **e^- Finder Development**
  - – Planning started to develop EICrecon eID
    - › **Help needed here, too!**
    - › Several defined tasks to contribute to
  - – **Right:** flowchart of tasks for EICrecon eID development



eID0: missing…

eID1: boost issue

eID2: isolation cut

eID3: deprecate…

eID4: wire reco 1

Milestone 1

eID5: wire reco 2

eID6: selection…

eID2: kinematic…

eID3: integration

Milestone 2

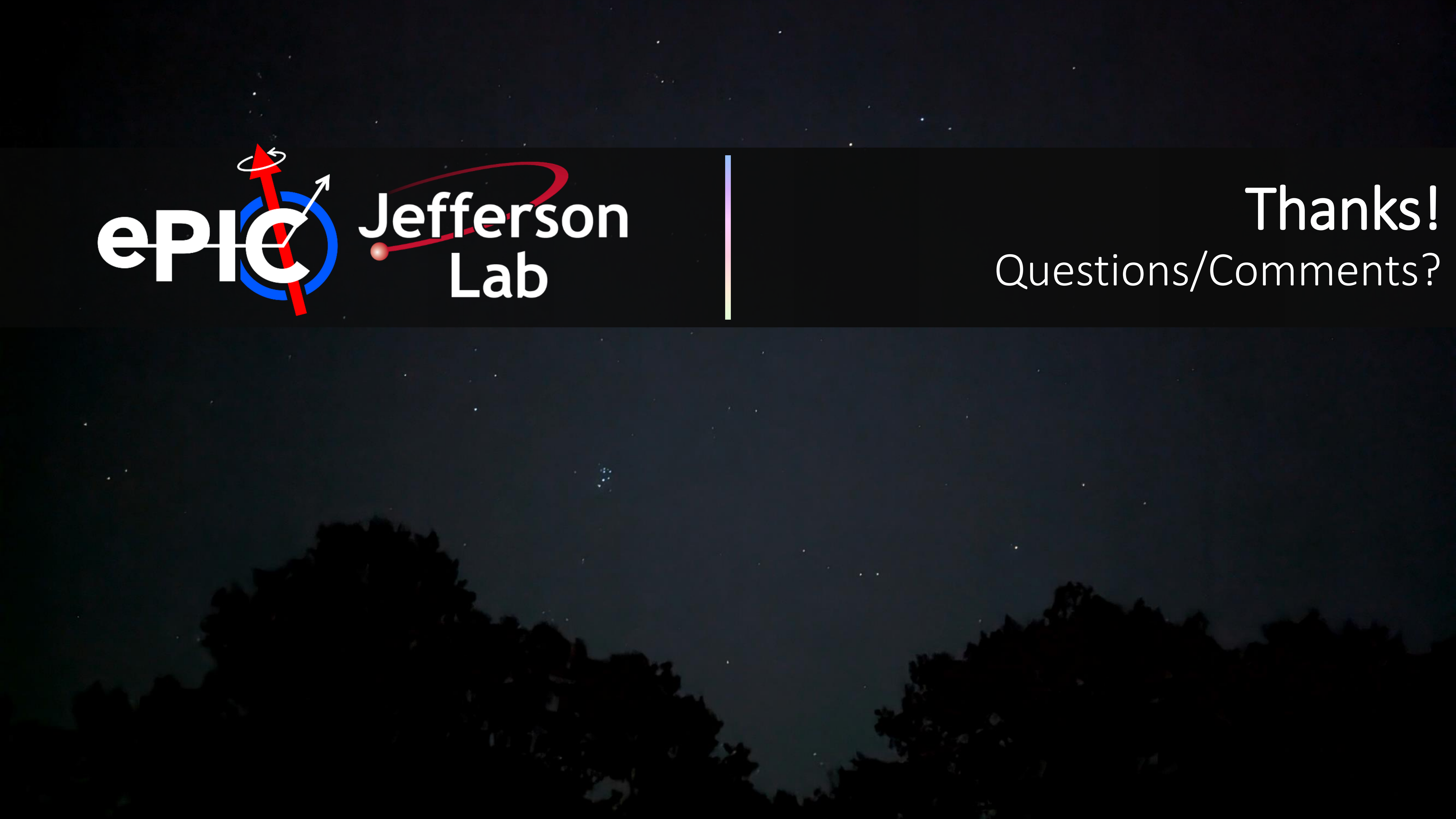
Benchmark/bkgd

Efficient boost

Multiple HFS

Multiple eID

Combiner algo



Thanks!

Questions/Comments?

Backup | Benchmark Task List (1/2)

Tasks	Issue/PR/Note	Est. labor time*	Assignee
PFA-1 Benchmark **- input:** Sum eClust, sum pTrk, nClust, nTrk, E/p matched clusters, sum eGenPar, eGenPar, nGenPar **- output:** Sum eRecPar, eRecPar, ePar, nRecPar, nPar, PES/R of reco pars	To-do	1 week	**OPEN**
PFA0 Benchmark **- input:** Sum eClust, eClust, pTrk, nTrk, nClust, E/p matched clusters **- output:** Sum eSMClust, eSMClust, nSMClust, E/p SM clust, dRct SM	Some work done	1 week	**BLOCKED**
PFA1 Benchmark **- input:** Sum eClust, eClust, sum pTrk, pTrk, nTrk, nClust, E/p matched clusters, sum pChrgPar, pChrgPar, nChrgPar **- output (expected):** sum eEXClust, eEXClust, nEXClust, E/p EX clust, dRct EX **- output (remnant):** sum eREClust, eREClust, nREClust **- output:** sum eEXClust + eREClust	To-do	1 week	**OPEN**

* Assuming 50% FTE, including code review time

○ **Notes:**

- PES/R = Particle Energy Scale/Resolution
- SM = Split/Merge, EX = Expected, RE = Remnant
- dRct = distance b/n cluster & matched track

Backup | Benchmark Task List (2/2)

Tasks	Issue/PR/Note	Est. labor time*	Assignee
PFA2 Benchmark **- input:** sum eREClust (EM, H), eREClust (EM, H), nREClust (EM, H), sum eNeuPar, eNeuPar, nNeuPar **- output:** sum eRecPar, nRecPar	To-do	1 week	**OPEN**
PFA3: **- input:** Sum eClust, sum pTrk, nClust, nTrk, E/p matched clusters, sum eGenPar, eGenPar, nGenPar **- output:** Sum eRecPar, eRecPar, ePar, nRecPar, nPar, PES/R of reco pars	To-do	1 week	**OPEN**
PHYS Benchmark: JES/R	To-do (just need wiring)	3 days	Dener
PHYS Benchmark[a,b]: Jets - E, mass, FFs (jt, z), Substructure (dRcst, angularity, EECs)	To-do	1 week	Dener
PHYS Benchmark[a]: Events - TEECs, NECs	NECs in progress (see here)	3 weeks	Derek (NECs)

* Assuming 50% FTE, including code review time

a) Desirable, but not required

b) Could do inclusive, HF-tagged, etc.

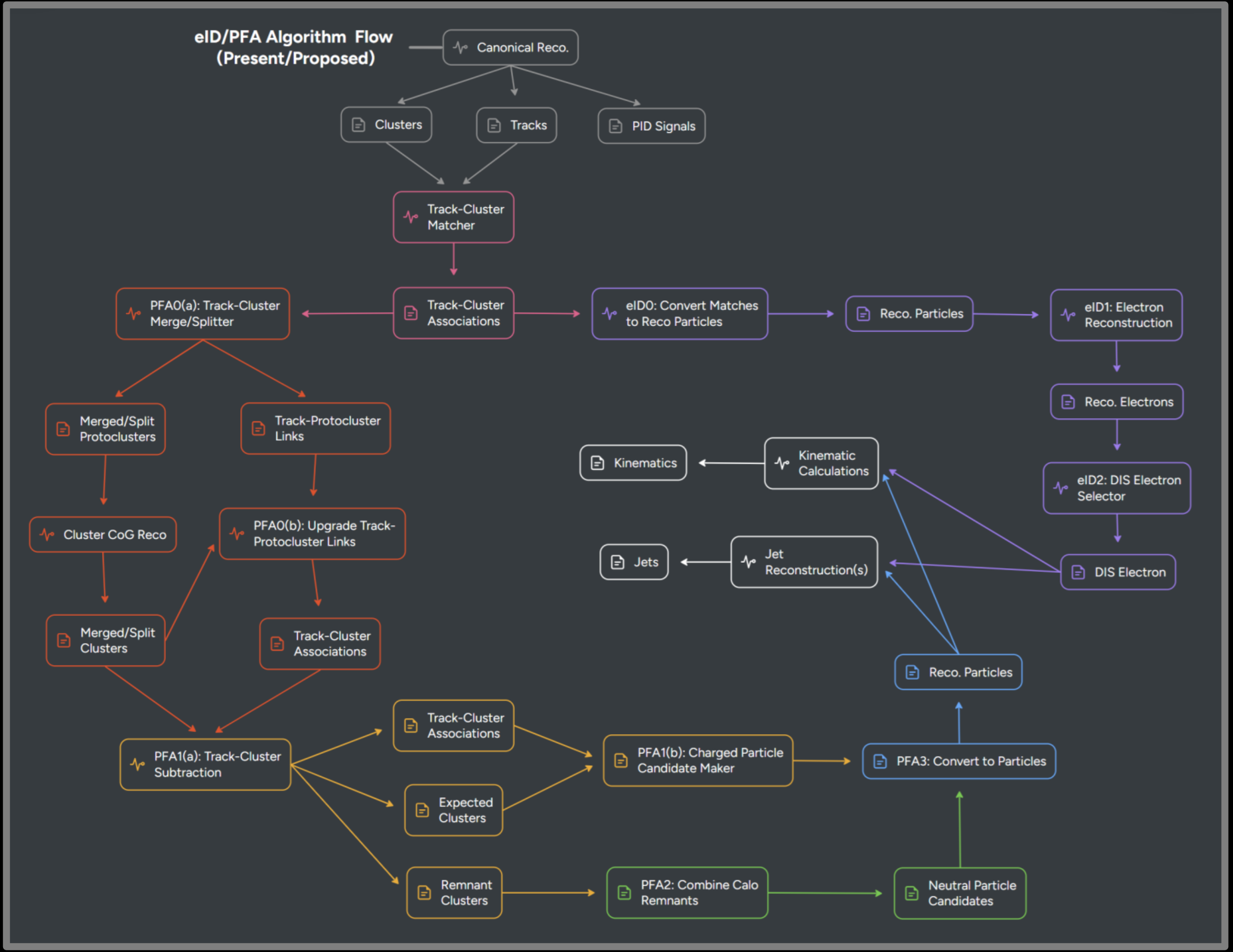
○ **Notes:**

- EM = “Electromagnetic”, H = “Hadronic
- dRcst = constituent delta-R

Backup | Baseline Overview

- **In broad strokes:** the overall algorithm is

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
   - Split merged clusters between matched tracks
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]** Convert candidates to reconstructed particles



eID/PFA Algorithm Flow (Present/Proposed)

Canonical Reco.

Clusters

Tracks

PID Signals

Track-Cluster Matcher

Track-Cluster Associations

PFA0(a): Track-Cluster Merge/Splitter

eID0: Convert Matches to Reco Particles

Reco. Particles

eID1: Electron Reconstruction

Reco. Electrons

eID2: DIS Electron Selector

DIS Electron

Merged/Split Protoclusters

Track-Protocluster Links

Cluster CoG Reco

PFA0(b): Upgrade Track-Protocluster Links

Merged/Split Clusters

Track-Cluster Associations

Kinematics

Kinematic Calculations

Jets

Jet Reconstruction(s)

Reco. Particles

PFA1(a): Track-Cluster Subtraction

Track-Cluster Associations

Expected Clusters

Remnant Clusters

PFA1(b): Charged Particle Candidate Maker

PFA3: Convert to Particles

PFA2: Combine Calo Remnants

Neutral Particle Candidates

Backup | PFA-1/eID3 | Track-Cluster Match Converter

- **Track-Cluster Match Converter:** MatchClusters (current source of ReconstructedParticles) is one of biggest truth info leaks in reconstruction
  - But with track-cluster matches, we can now patch it (see EICrecon#1956)
  - Intended to supersede MatchClusters while PFAlpha develops

- **The algorithm:** is identical to MatchClusters, but with track-cluster matches rather than truth info
  1) Build map of tracks onto matched clusters, and set of clusters
  2) For each track:
     a) Remove matched cluster from cluster set
     b) Copy associated charged particle of track into output
  3) For each remaining cluster in set:
     - Create a reconstructed particle with mass and PDG of 0

Inputs:
- Track-cluster matches
- Reconstructed charged particles
- Clusters
  - › Planned on only using EMCal clusters (like MatchClusters)
  - › But *could* add in HCal's

Outputs:
- Reconstructed particles

Parameters:
- None (if using only EMCal clusters)

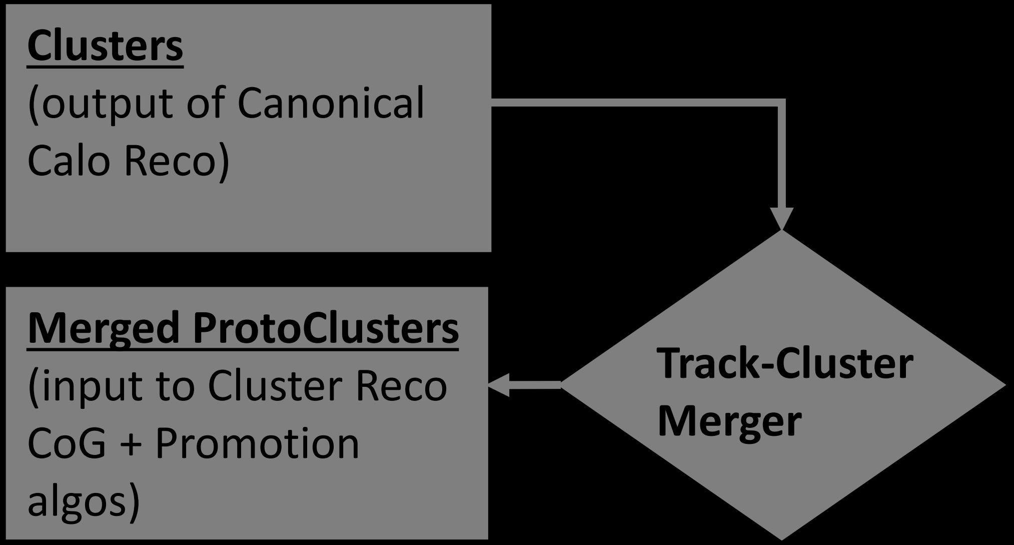
Backup | PFA0 | Track-Cluster Merge/Splitter (1/2)

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

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

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

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



Clusters
(output of Canonical Calo Reco)

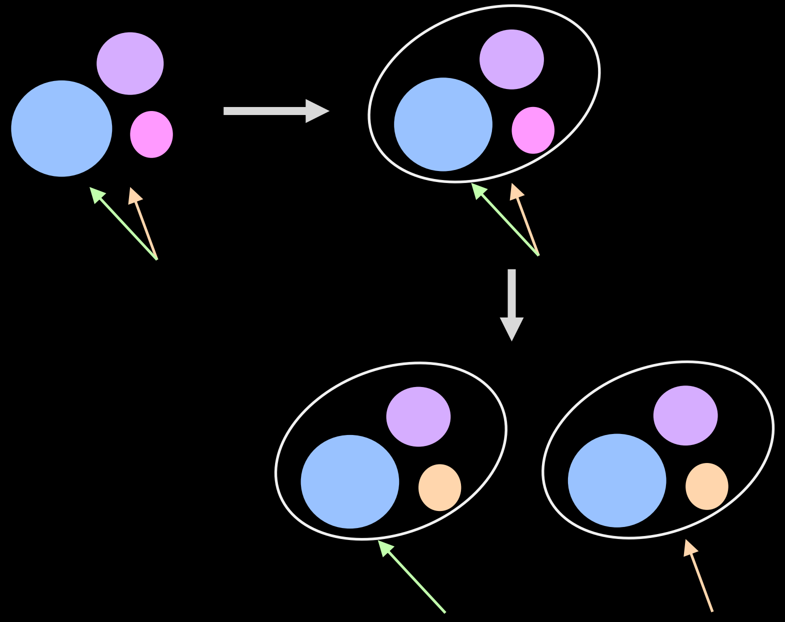
Track-Cluster Merger

Merged ProtoClusters
(input to Cluster Reco CoG + Promotion algos)

Parameters:
- $\langle E/p \rangle$: Average E/p
- $\sigma\left(E_{dep}\right)$: 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 | PFA0 | Track-Cluster Merge/Splitter (2/2)

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



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

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

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

Parameters:
- $\langle E/p \rangle$: Average E/p
- $\sigma\left(E_{dep}\right)$: 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 | PFA1(a) | Track-Cluster Subtractor (1/3)

- **Track-Cluster Subtractor:** subtracts momentum of matched track(s) from cluster
  - In progress at EICrecon#1627

- **The algorithm:**
  1) Build map of clusters onto *all* matched tracks
  2) For each cluster:
     a) Sum energy of matched tracks:

        $$E_{trk} = \sum p_{trk}(S_{use}) \oplus m_{trk}$$

     b) Subtract sum: $E_{sub} = E_{clust} - f_{sub}E_{trk}$
     c) If NOT consistent w/ 0,
        - Create remnant cluster w/ E_{sub}
        - Set expected cluster energy to $E_{clust} - E_{sub}$
     d) Create an association for each track matched to expected cluster

Inputs:

- Track-cluster matches
- Clusters
- Track projections

Outputs:

- Remnant clusters ($E_{sub} = E_{clust} - E_{trk}$)
- Expected clusters ($E_{clust} - E_{sub}$)
- Track-expected cluster association

Parameters:

- $\boldsymbol{f_{sub}}$: fraction of track energy to subtract
- $\boldsymbol{m_{default}}$: default mass to use for track energy
- $\boldsymbol{S_{use}}$: surface to evaluate track momentum at
- $\boldsymbol{k_{do\ n\sigma?}}$: turn on/off checking against resolutions
- $\boldsymbol{n\sigma_{cut}}$: max no. of sigmas to be consistent w/ 0
- $\boldsymbol{\sigma_{trk}}$: tracking resolution to use in n-sigma cut
- $\boldsymbol{\sigma_{cal}}$: calo resolution not use in n-sigma cut

Backup | PFA1(a) | Track-Cluster Subtractor (2/3)

- **Track-Cluster Subtractor:** subtracts momentum of matched track(s) from cluster
  - In progress at EICrecon#1627

- **The algorithm:**
  1) Build map of clusters onto *all* matched tracks
  2) For each cluster:
     a) Sum energy of matched tracks:

     $$E_{trk} = \sum p_{trk}(S_{use}) \oplus m_{trk}$$

     b) Subtract sum: $E_{sub} = E_{clust} - f_{sub} E_{trk}$
     c) If NOT consistent w/ 0,
        - Create remnant cluster w/ E_{sub}
        - Set expected cluster energy to $E_{clust} - E_{sub}$
     d) Create an association for each track matched to expected cluster

Sub-routine: is E_{sub} consistent w/ zero?

1) **If** $E_{sub} < 0$, **YES**
2) **Else if** $k_{do\ n\sigma?}$
   a) Calculate $n\sigma$

   $$n\sigma = \frac{E_{sub}}{\sigma_{trk} \oplus \sigma_{cal}}$$

   b) **If** $n\sigma < n\sigma_{cut}$, **YES**
3) **Else**
   a) **If** $E_{sub} < \epsilon$, **YES**

Note: epsilon here is std::numeric_limits<double>::epsilon()

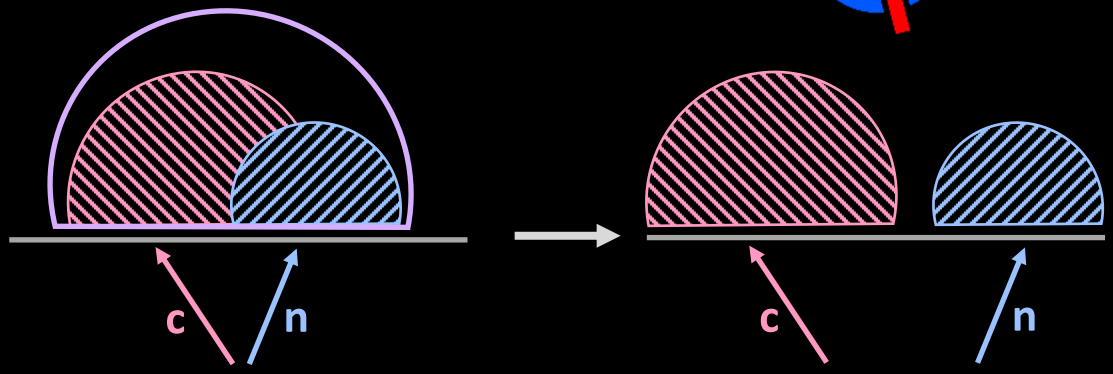
Backup | PFA1(a) | Track-Cluster Subtractor (3/3)

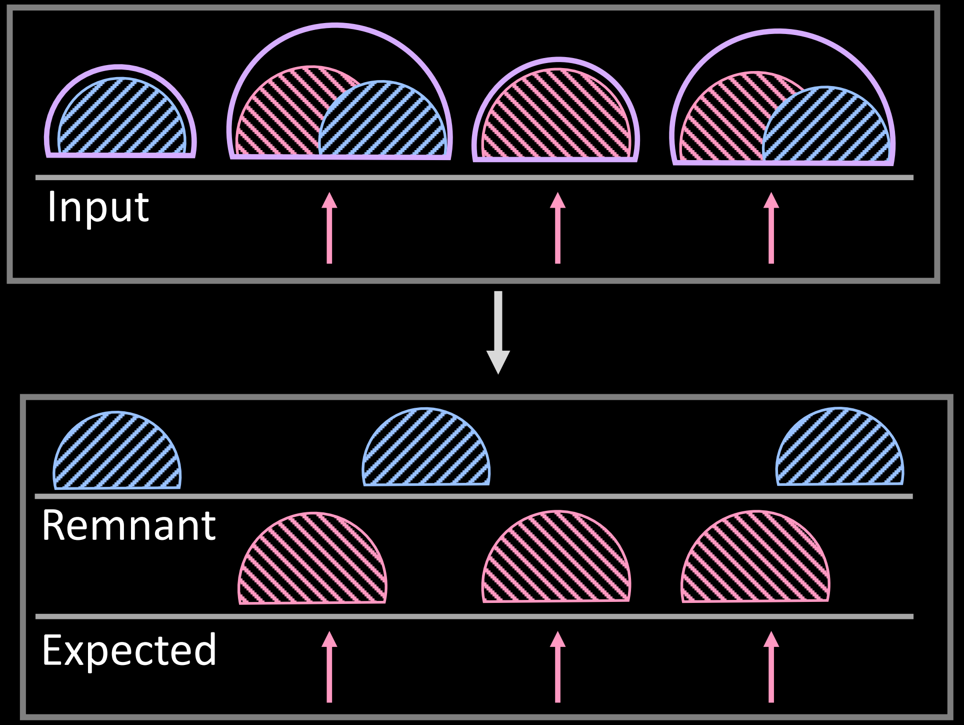
- **Track-Cluster Subtractor:** subtracts momentum of matched track(s) from cluster
  - In progress at EICrecon#1627

- **The algorithm:**
  1) Build map of clusters onto *all* matched tracks
  2) For each cluster:
     a) Sum energy of matched tracks:

        $$E_{trk} = \sum p_{trk}(S_{use}) \oplus m_{trk}$$

     b) Subtract sum: $E_{sub} = E_{clust} - f_{sub}E_{trk}$
     c) If NOT consistent w/ 0,
        - Create remnant cluster w/ E_{sub}
        - Set expected cluster energy to $E_{clust} - E_{sub}$
     d) Create an association for each track matched to expected cluster





Backup | PFA1(b) | Charged Particle Candidate Maker

- **Charged Candidate Maker:** forms track-cluster matches into a charged particle candidate
  - To-do

- **The algorithm:**
  1) Build map of tracks onto *all matched clusters*
  2) For each track:
     a) For each matched cluster:
        i. Identify if in an ECal or an HCal by checking system ID
        ii. Select relevant weight
        iii. Add to relevant members
     b) Add to relevant member

Inputs:
- Track-cluster matches

Outputs:
- Charged particle candidates

Parameters:
- **$\{ID_{ecal}\}$:** IDs of ECals to look for
- **$\{ID_{hcal}\}$:** IDs of HCals to look for
- **$\{w_{em}\}$:** weights of ECals to use (runs parallel to $\{Id_{ecal}\}$)
- **$\{w_{h}\}$:** weights of HCals to use (runs parallel to $\{Id_{hcal}\}$)

Backup | PFA2 | Calo Remnant Combiner

- **Calo Remnant Combiner:** combines remnant clusters from subtractor into neutral particle candidates
  - Still to-do!

- **The algorithm:**
  1) Combine nearby ECal, HCal clusters
     a) Identify seed ECal cluster
     b) Merge all ECal, HCal clusters in Δr^{em}_{add}, Δr^{h}_{add} of seed and create neutral candidate
     c) Repeat until no ECal clusters are left
  2) Combine remaining HCal clusters
     a) Identify seed HCal cluster
     b) Add all HCal clusters in Δr^{h}_{add} of seed and create neutral candidate
     c) Repeat until no HCal clusters are left

Inputs:
- Remnant ECal clusters
- Remnant HCal clusters

Outputs:
- Neutral particle candidates

Note: maybe make inputs vectors of collections?

Parameters:
- $\mathbf{\Delta r^{em}_{add}}$: window to add ECal clusters
- $\mathbf{\Delta r^{h}_{add}}$: window to add HCal clusters
- $\{w_{em}\}$: weights (or weight) of ECals to use
- $\{w_{h}\}$: weights (or weight) of HCals to use

Backup | PFA3 | Candidate-to-Particle Converter

- **Particle Converter:** takes candidate particles and turns them into reconstructed particles
  - Still to-do!

- **The algorithm:**
  1) Assign preliminary PID based on what info is available (e.g. no hcal clusters → electron, photon, or pi0)
  2) Calculate track energy

$$E_{trk} = p_{trk} \oplus m_{pid}$$

  3) Calculate calorimeter energy

$$E_{cal} = N_{cal}\left(\sum w_{em}E_{em} + \sum w_h E_h\right)$$

  4) If charged particle and $k_{use\ \sigma?}$, calculate resolution-weighted average of E_{cal} and E_{trk}
  5) Calculate remaining kinematics and create reconstructed particle

Inputs:
- Candidate charged/neutral particles
- Primary vertices (for neutral candidates)

Outputs:
- Reconstructed particles

Parameters:
- $k_{use\ \sigma?}$: turn on/off using resolution in energy calculation for charged candidates
- N_{cal}: normalization of calo energy
- σ_{trk}: tracking resolution to use in energy calc
- σ_{cal}: calo resolution to use in energy calc

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 cluster

- **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 | Mapping Initial PFA Attempt Onto Current

PFA-1

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_{sum}^{ECal}, R_{sum}^{HCal} of seed
   c) Sum E_{clust} of all ECal, HCal clusters in R_{sum}^{ECal}, R_{sum}^{HCal} respectively

PFA1(a)

   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

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