Particle Flow and Finders: ElCrecon Update

2026 EIC UGM, Tracking Workfest

Derek Anderson

07.16.2026

Overview | This Talk and Summary

- **Intended to complement Alex & Stephen's talks on muID & eID**
  - Will be focusing on what's currently in (or close to being in) EICrecon and near-term plans
  - Hope to spur discussion on what's needed for next steps
- **Will divide talks into 3 Components:**
  1) Track-Cluster Matching
  2) Particle Flow
  3) Electron Finding
- **High-level status overview shown right**
  - ☞ Will go into (gory) detail in following sides

Track-Cluster Matching
- Matching parameters need tuning (Mounika)
- Algorithm needs improvements (Alexander, Sasha, Wouter)

Particle Flow
- EICrecon implementation approaching completion (Subhadip, Esteban, Derek)
- PFA0, PFA1 tuning ongoing (Mounika, Maciej)

Electron Finding
- Improvements ongoing (Diego, Barak, Stephen, Win)
- MatchClusters algorithm deprecation becoming critical

Track-Cluster Matching | Algorithm (1/2)

- **Track-Cluster Matching:** matches tracks and clusters in (η, φ) space based on track projections
  - Implemented in EICrecon#1768

- **Algorithm:**
  1) For each cluster:
     a) Set $\Delta r = \Delta r_{match}$
     b) For each point in each projection:
        i. Skip if not in I_{calo} at S_{use}
        ii. Compute distance d in (η, φ) space
        iii. If $d < \Delta r$
           - Set $\Delta r = d$ and match to track
     c) If match found,
        - Create association between track and matched cluster

Inputs:
- Clusters
- Track projections

Outputs:
- Track-cluster matches

Parameters:
- Δr_{match}: max distance between projection and cluster to consider
  - Default = 0.5 for all calorimeters
- I_{calo}: ID of calorimeter to match to
- S_{use}: surface of projection to use
  - Note: set to 1 right now

Track-Cluster Matching | Algorithm (2/2)

- **Track-Cluster Matching:** matches tracks and clusters in (η, φ) space based on track projections
  - Implemented in EICrecon#1768

- **Algorithm:**
  1) For each cluster:
     a) Set $\Delta r = \Delta r_{match}$
     b) For each point in each projection:
        i. Skip if not in I_{calo} at S_{use}
        ii. Compute distance d in (η, φ) space
        iii. If $d < \Delta r$
           - Set $\Delta r = d$ and match to track
     c) If match found,
        - Create association between track and matched cluster

- **Subtle bug:** Cluster order matters! Which clusters get matched is first-come-first-served
  - ☞ Alexander & Sasha (CTU), Wouter (UM) working on EICrecon#2498

- **Needs tuning!** Δr_{match} set to 0.5 for all calorimeters
  - Mounika (IISER-Tirupati) working on this while working on tuning PFA0

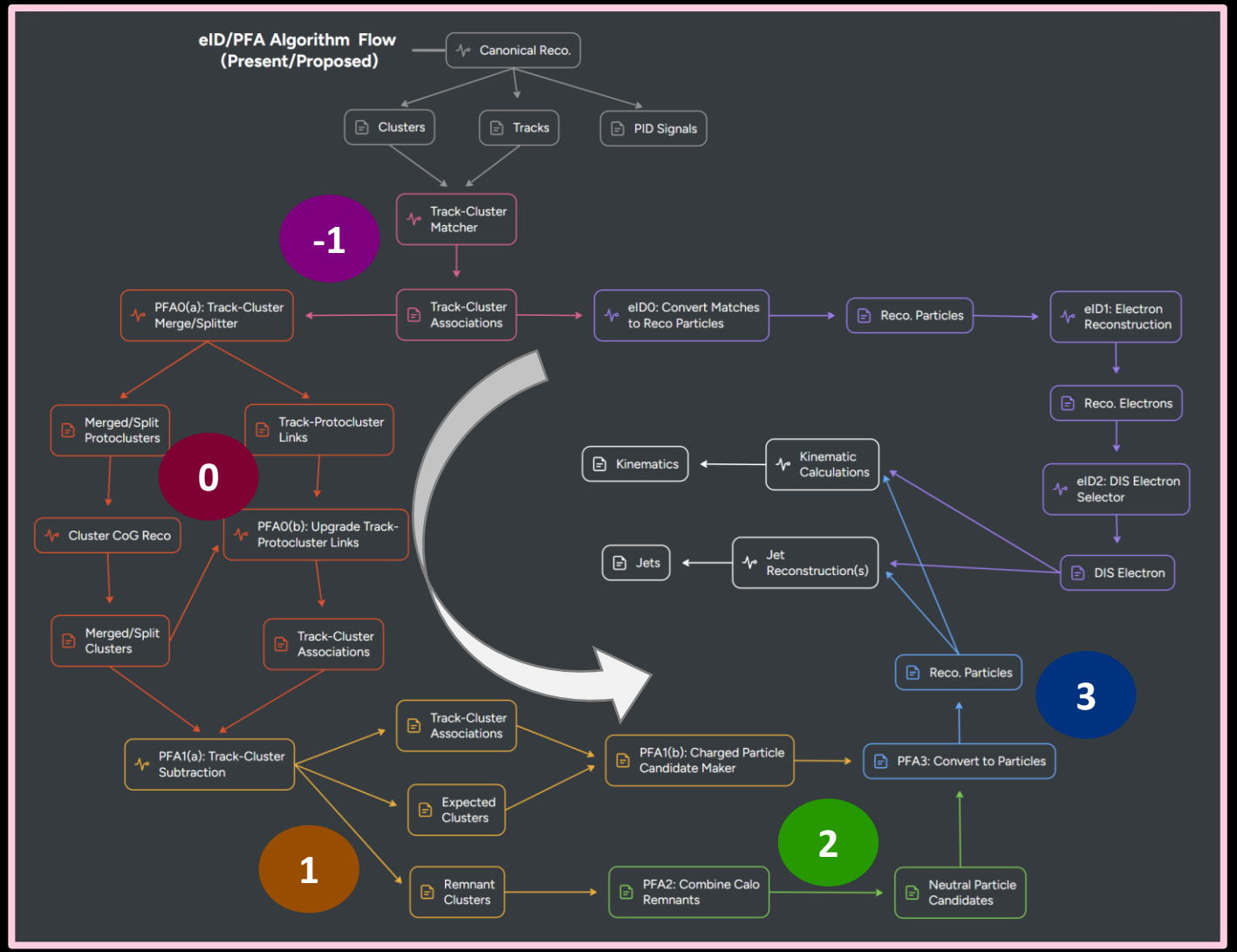
- **Other to-do's:**
  - Upgrade from associations to links
  - Enable S_{use} to be adjusted
  - Enable distance metrics other than (η, φ)

Particle Flow | Algorithm Overview

Algorithm

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

- **Notes:**
  - PFAX = "PFAlpha Stage X"
  - Finer details of each stage in following slides/backup

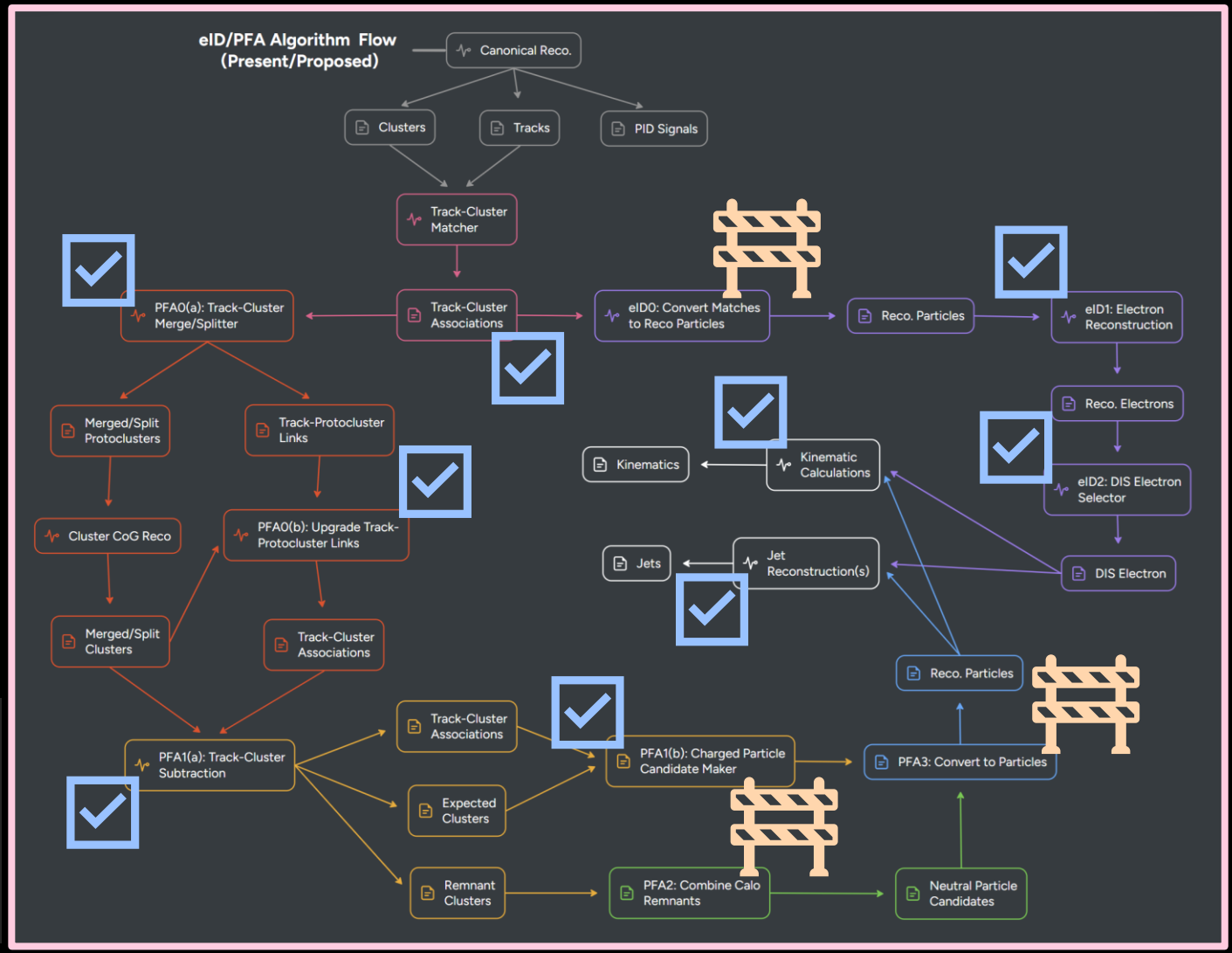


Particle Flow | Current Status

- **Last update:** 05.27 PAC/S&C Discussion
  - All of PFA0, PFA1 merged and available as of EICrecon v1.39.0 (26.07.0)
  - Focus now on PFA2 (Subhadip, CTU), PFA3 (Esteban, MSU)

- **Tuning:** in progress!
  - Mounika (IISER-Tirupati) tuning PFA0
    - ☞ Targeting update in Jet/HF PWG at end of month
  - Maciej (AGH) tuning PFA1
    - ☞ Update not long after PFA0

◯ = To-do
🚧 = In progress
☑ = Done/already in EICrecon
⊗ = Blocked



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

- **Track-Cluster Merging:** Algorithm based on ATLAS's split recovery procedure
  - ☞ c.f. Eur. Phys. J. C (2017) 77:466
  - – Implemented in EICrecon#1406, updated in EICrecon#1699

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

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

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

Inputs:
- Clusters
- Track-cluster matches
- Track projections

Outputs:
- Merged/split clusters
- Track-(merged/split) cluster matches

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

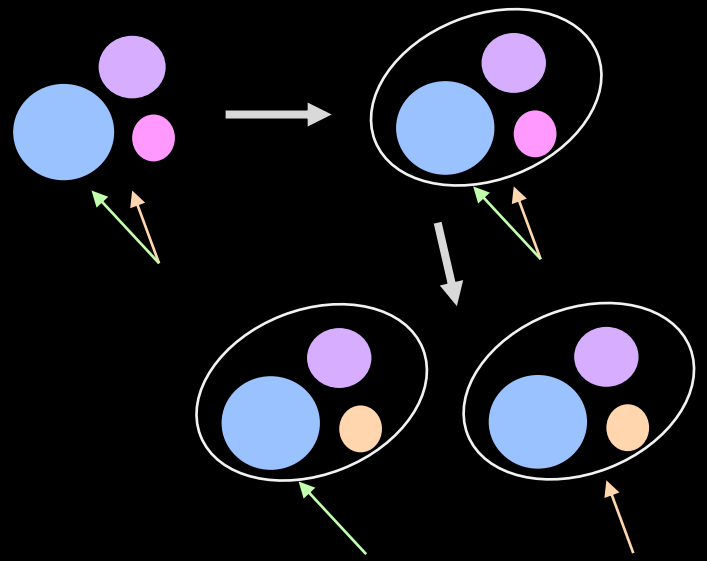
Particle Flow | PFA0 | Track-Cluster Merge/Splitter (2/2)

- **Track-Cluster Merging:** Algorithm based on ATLAS's split recovery procedure
  - ☞ c.f. Eur. Phys. J. C (2017) 77:466
  - – Implemented in EICrecon#1406, updated in EICrecon#1699

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

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

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



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

Particle Flow | PFA1(a) | Track-Cluster Subtractor (1/3)

- **Track-Cluster Subtractor:** subtracts momentum of matched track(s) from cluster
  - Implemented in EICrecon#1627

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

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

- **Track-Cluster Subtractor:** subtracts momentum of matched track(s) from cluster
  - Implemented in EICrecon#1627

- **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 $E_{clust} > f_{sub}E_{trk}$,
        - 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 $E_{clust} > f_{sub}E_{trk}$?

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()

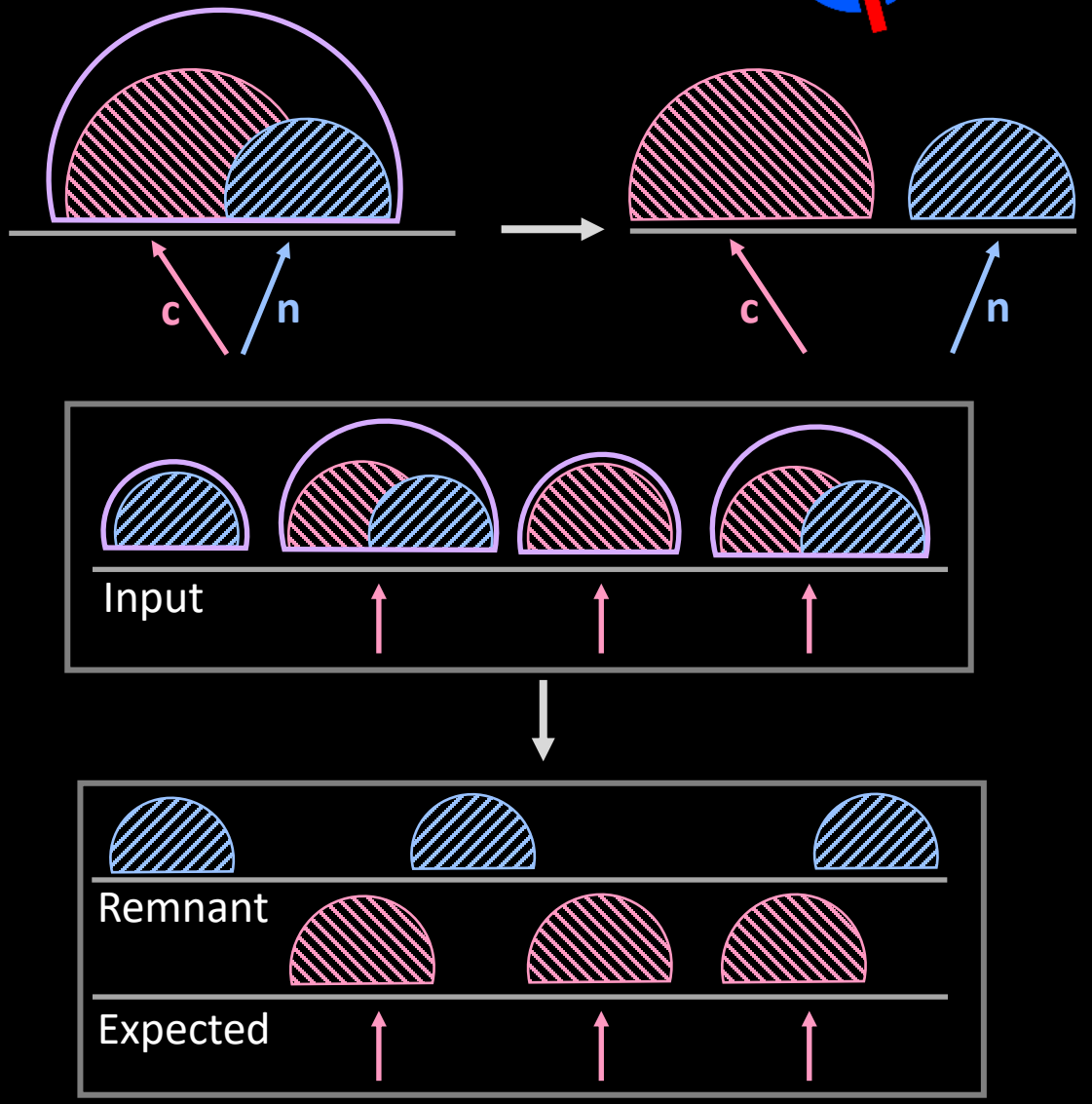
Particle Flow | PFA1(a) | Track-Cluster Subtractor (3/3)

- **Track-Cluster Subtractor:** subtracts momentum of matched track(s) from cluster
  - Implemented in EICrecon#1627

- **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 $E_{clust} > f_{sub}E_{trk}$,
        - 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



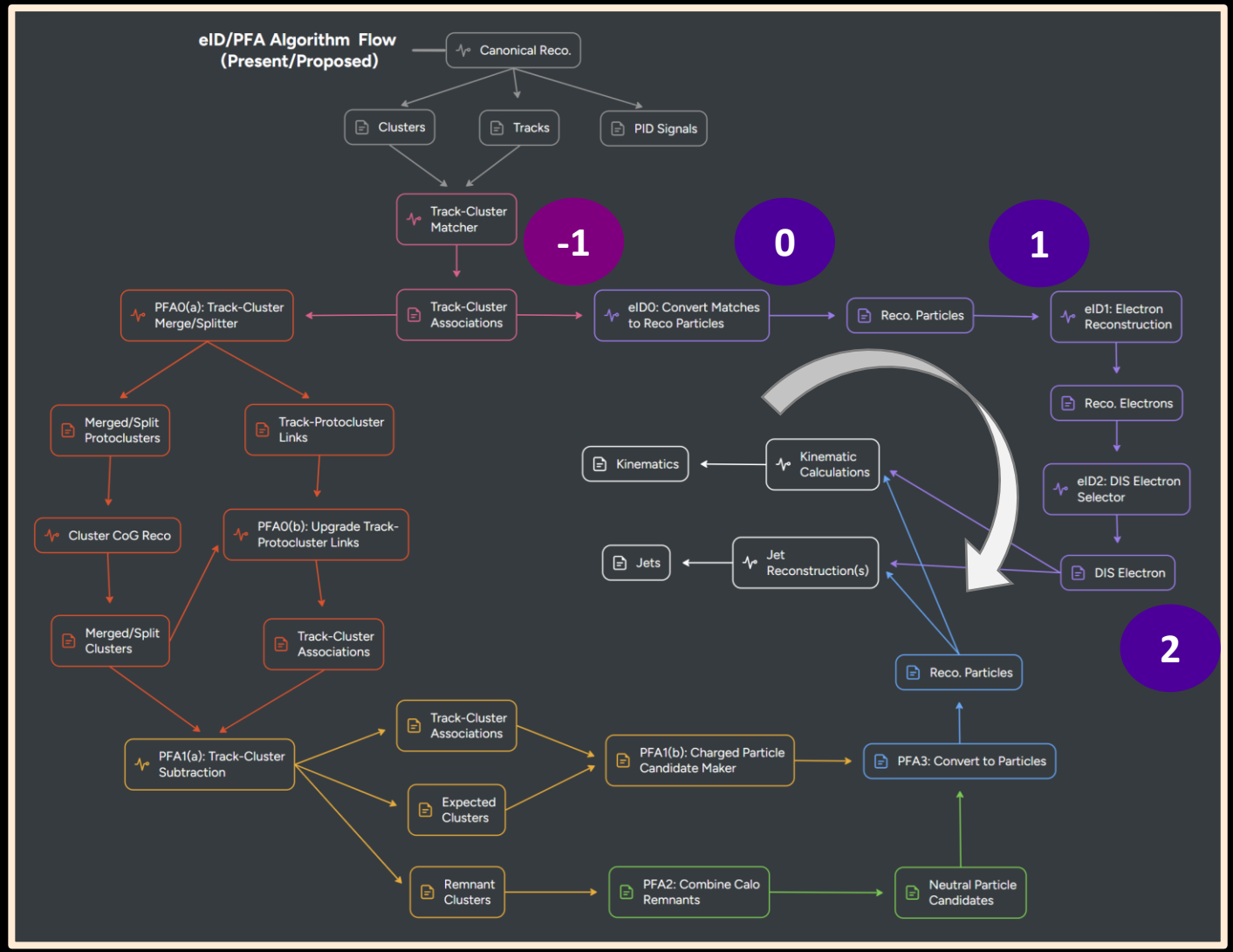
Electron ID | (EICrecon) Algorithm Overview

Algorithm

1) **[eID-1]** Match tracks to EMCal, HCal clusters
2) **[eID0]** Convert track-cluster matches to particles
   - Does ***NOT*** exist yet!
3) **[eID1]** Identify electrons based on E/p
4) **[eID2a]** (Reco) Rank electrons based on E-p_z
5) **[eID2b]** (Truth) Find DIS electron based on truth info

- **Current input to algorithm:** output of MatchClusters!
  - Clusters are matched to tracks based on truth info
  - 👉 **This is a *BIG* leak of truth info!**

- **Notes:**
  - eIDX = "eID Stage X"
  - Finer details of each stage in following slides

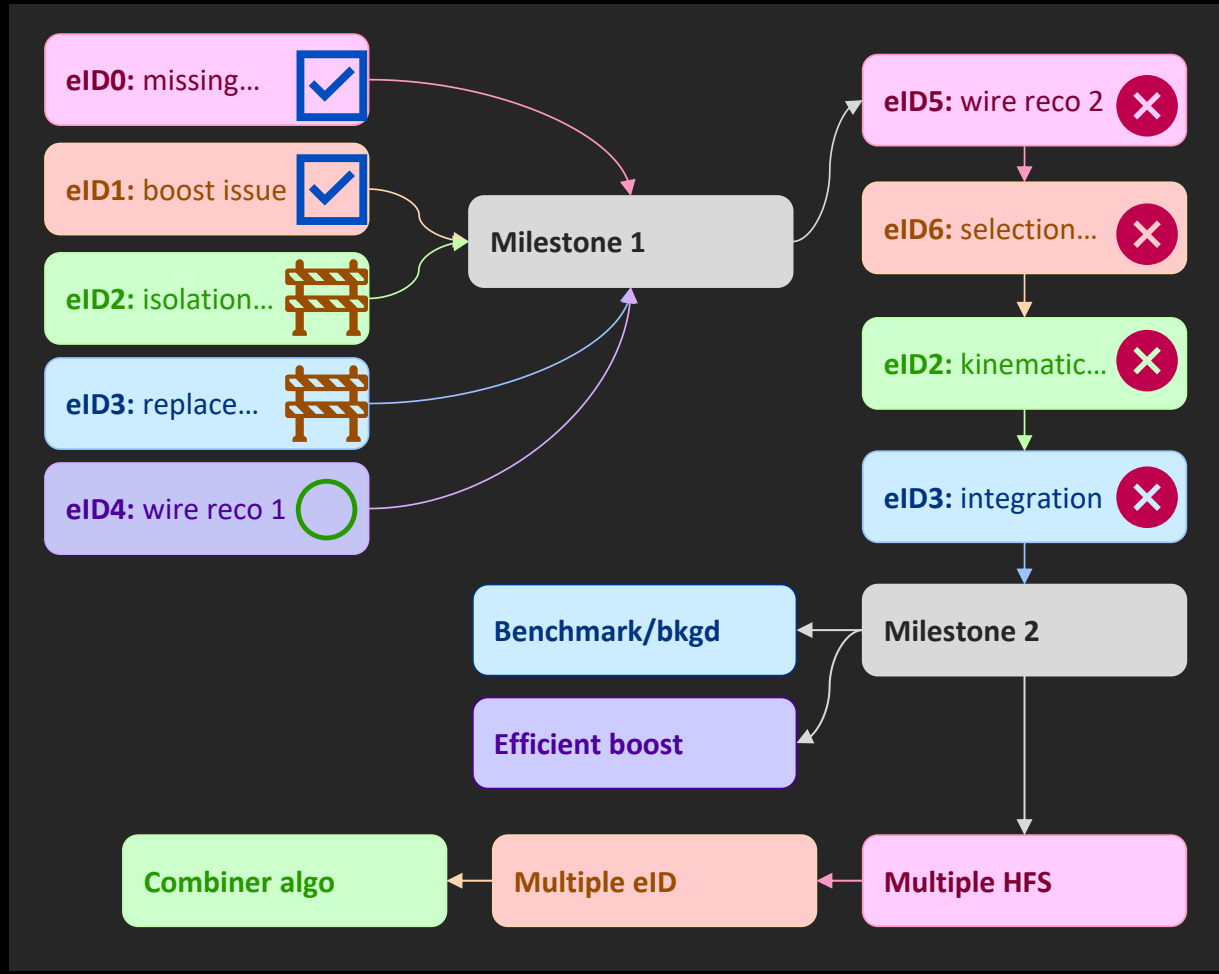


Electron ID | eID Task Flowchart

- **Milestones:** 2 key ones along EICrecon eID development
  1) Truth info removed, standalone improvements integrated
  2) Further improvements to DIS selection, integration with PF

- **Last update:** 05.27 PAC/S&C Discussion
  - **Right:** task flowcharts for eID development
  - ☞ eID3 (MatchClusters deprecation) moving onto critical path for PF…

◯ = To-do
▦ = In progress
☑ = Done/already in EICrecon
⊗ = Blocked



Electron ID | eID0 | Track-Cluster Match Converter

- **Track-Cluster Match Converter:** MatchClusters (current source of ReconstructedParticles) is one of biggest truth info leaks in reconstruction (see EICrecon#1956)
  - **Note:** MatchClusters looks identical to this *except* it uses MC-reco associations instead of matches
  - Should supersede MatchClusters while PFAlpha develops and serve as benchmark

- **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
- EMCal Clusters
  - › Intended to parallel MatchClusters as much as possible for benchmarking

Outputs:
- Reconstructed particles

Parameters:
- None!

Electron ID | eID1 | Electron Reconstruction

- **Electron Finder:** identifies electrons based on E/p
  - Longer term, this is where we would make use of PID subsystem info
  - Nearer term, could add in HCal info as a veto or factor into a likelihood

- **Algorithm:**
  1) For each reconstructed particle
     b) Compute E/p
     c) If $E/p \in \left((E/p)_{min}, (E/p)_{max}\right)$
        - Add reconstructed particle to output

Inputs:
- Reconstructed particles

Outputs:
- Reconstructed electrons

Parameters:
- $(E/p)_{min}$: minimum E/p value
  - Defaults = 0.9 for all e^-, 0.7 for DIS e^-
- $(E/p)_{max}$: maximum E/p value
  - Defaults = 1.2 for all e^-, 1.3 for DIS e^-

Electron ID | eID2a | Reco DIS Electron Selection

- **Reco DIS Electron Selector:** creates list of DIS e^- candidates ranked by E-p_z
  - Should refactor to take advantage of HadronicFinalState algorithm
  - Working towards adding in isolation cut in EICrecon#2404

- **Algorithm:**
  1) Define map of $E - p_z$ onto reconstructed electron, sorted in decreasing $E - p_z$
  2) For each reconstructed electron
     a) For each reconstructed particle
        i. Skip if same particle as electron
        ii. Add p_μ to Σp_μ
     b) Compute from Σp_μ
     c) If $E - p_z \in \left((E - p_z)_{min}, (E - p_z)_{max}\right)$
        – Add electron to map
  3) For each entry in map
     – Add to electron to output

Inputs:
- Reconstructed electrons
- Reconstructed particles

Outputs:
- Ranked list of DIS e^- candidates

Parameters:
- $(E - p_z)_{min}$: minimum E-p_z value
  - Default = 0 GeV
- $(E - p_z)_{max}$: maximum E-p_z value
  - Default = 10^7 GeV

Electron ID | eID2b | Truth DIS Electron Selection

- **Truth DIS Electron Selector:** identifies DIS e^- based on truth info
  - Should also refactor to take advantage of HadronicFinalState algorithm

- **Algorithm:**
  1) Find 1st e^- , $e^-_{DIS\ MC}$, in event record (MC Particles)
  2) Identify reconstructed association of $e^-_{DIS\ Rec}$
  3) Add $e^-_{DIS\ Rec}$ to output

Inputs:
- MC Particles
- MC-Reco particle associations
- Reconstructed particles

Outputs:
- DIS e^-

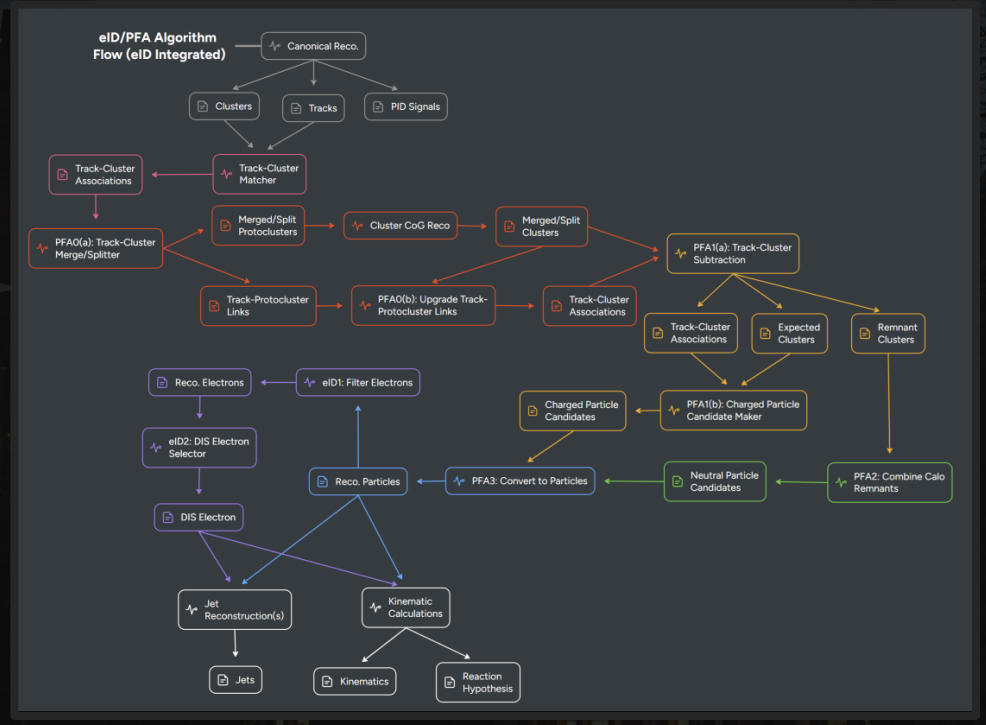
Parameters:
- None!

Thanks!

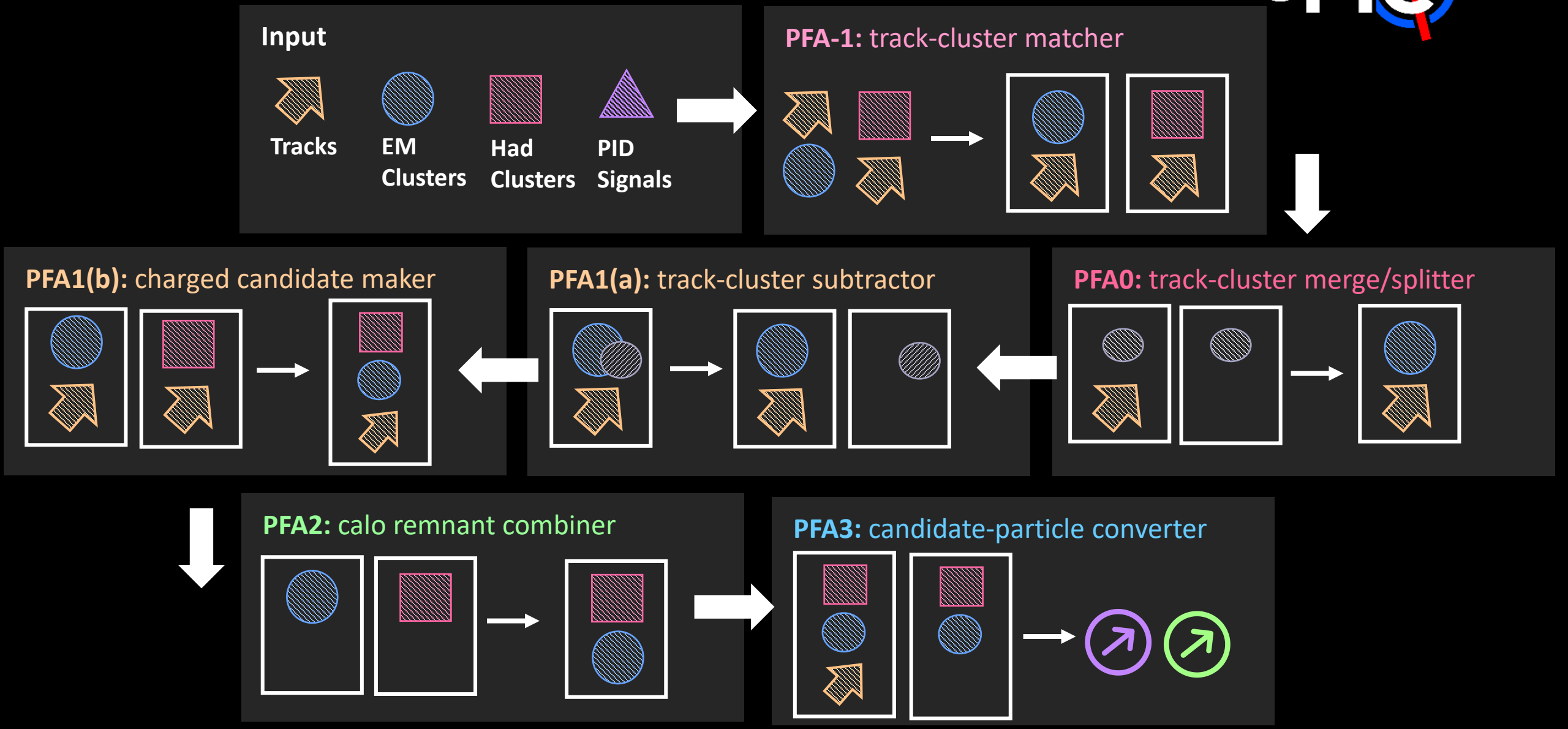
Questions/Comments?

Outlook

- As noted in previous talks, ultimate goal is to treat PF, eID, muID, and beyond on equal footing
  - ☞ **Right:** Possible workflow uniting PFAlpha and current eID routine
- Ongoing (and to-do) tasks moving us towards that goal
- **Note:** all 3 groups (PF, eID, muID) need to check impact of Barak's fixes in [EICrecon#1921]



Backup | PF Algorithm Schematic



Backup | PF Development Tasks

Task	Description	Issue/PR/Note	Exp Campaign	Assignees
PFA-1	Deprecate MatchClusters, replace w/ pure reco equivalent	EICrecon#1956	**26.09.0**	Tristan, **OPEN**
PFA0(a)	☑ Complete merge/splitter update	EICrecon#1699	**DONE**	Derek
PFA0(b)	☑ Implement track-protocluster link promotion	EICrecon#2293	**DONE**	Derek
PFA1(a)	☑ Revive and finish track-cluster subtractor	EICrecon#1627	**DONE**	Derek
PFA1(b)	☑ Track-cluster converter (synergy w/ PFA-1)	EICrecon#2124	**DONE**	Derek
EDM0	☑ Flagging ecal vs. hcal clusters	epic#994	**DONE**	Derek
PFA2	Implement calo remnant combiner	EICrecon#2195	**26.08.0**	Subhadip
PFA3	Implement particle regressor/convertor	EICrecon#2399	**26.08.0**	Esteban
INT	Tie all stages into a single workflow	EICrecon#900	**26.09.0**	**OPEN**

- **Above:** development tasks
  - All critical-path stages have PRs open and are under review (except INT)
- **Note:** PFA-1 is not (yet) on critical path for PF
  - ☞ But is relevant for eID development, and benchmarking PF performance

Backup | PF Benchmark Tasks (1/2)

Tasks	Issue/PR/Note	Exp Campaign	Assignees
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**	**≤ 26.10.0**	Mounika, **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	**In progress**	**≤ 26.10.0**	Mounika
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	**In progress**	**≤ 26.10.0**	Maciej

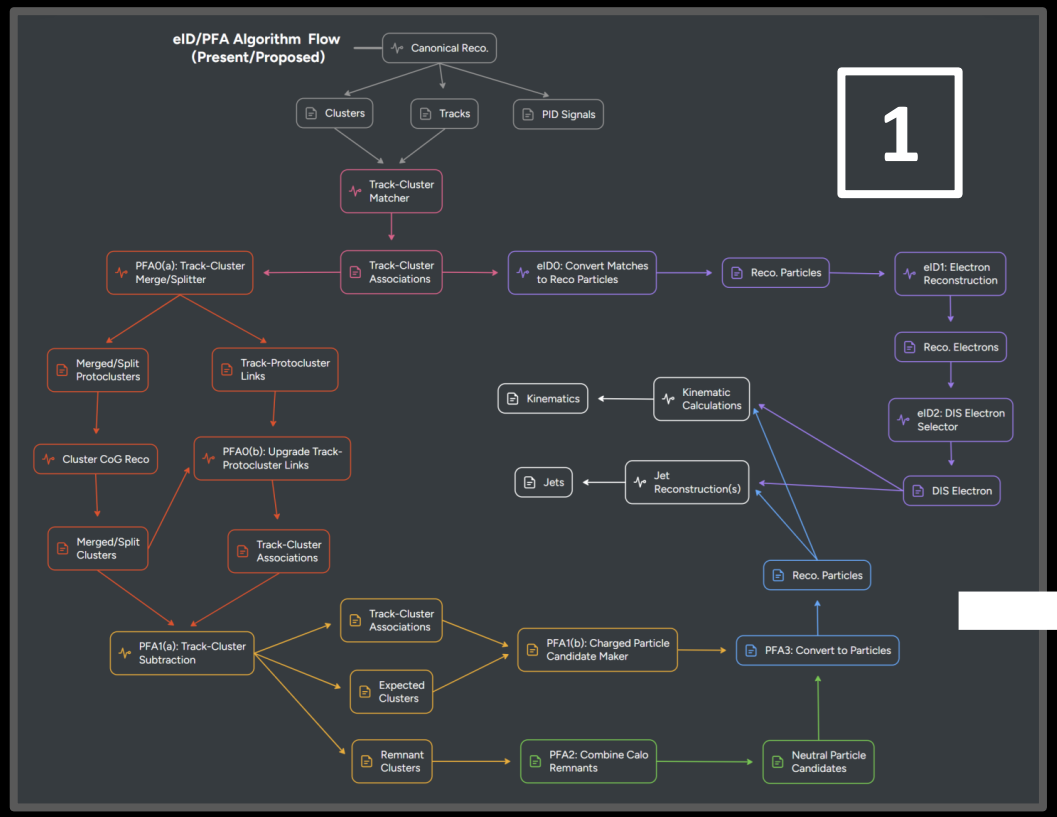
- **Above:** benchmarking/tuning tasks
  - New workforce getting ready to work on benchmarking once algos merged
  - **Still room to contribute!**
- **Notes:**
  - PES/R = Particle Energy Scale/Resolution
  - SM = Split/Merge, EX = Expected, RE = Remnant
  - dRct = distance b/n cluster & matched track

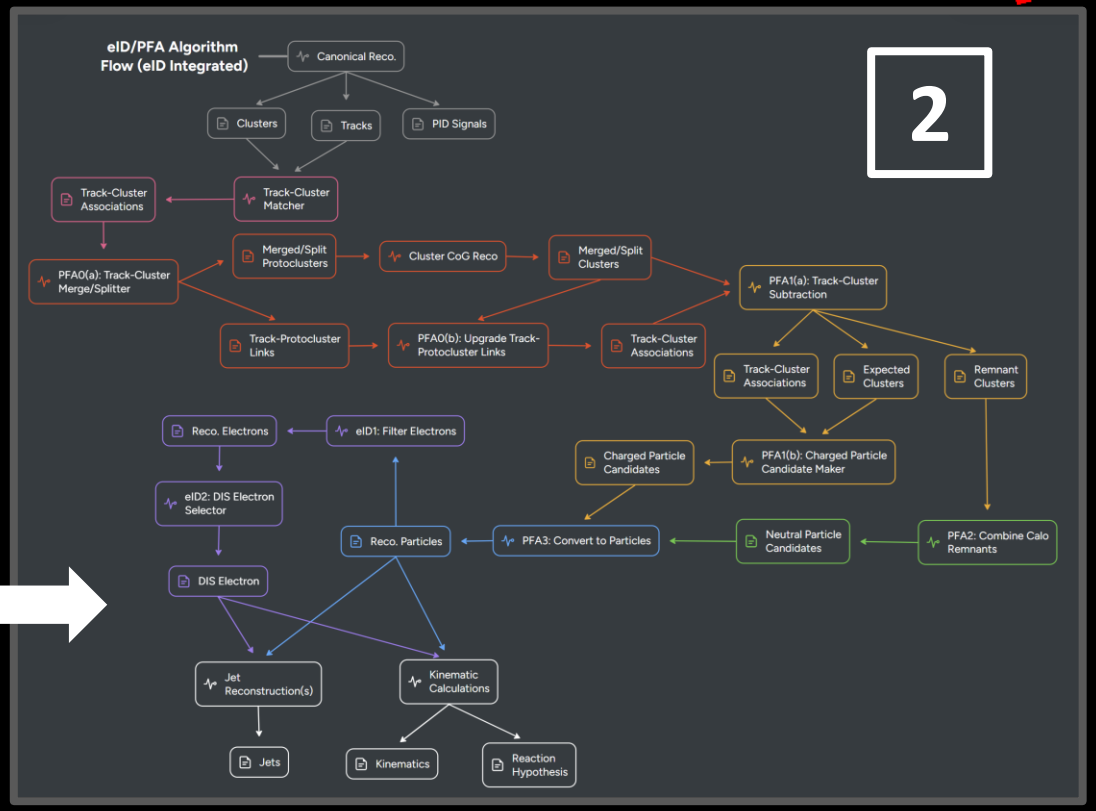
PF Status | PF Benchmark Tasks (2/2)

Tasks	Issue/PR/Note	Exp Campaign	Assignees
PFA2 Benchmark **- input:** sum eREClust (EM, H), eREClust (EM, H), nREClust (EM, H), sum eNeuPar, eNeuPar, nNeuPar **- output:** sum eRecPar, nRecPar	**To-do**	**≤ 26.10.0**	Subhadip
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**	**≤ 26.10.0**	Esteban
PHYS Benchmark: JES/R	**To-do**	**≤ 26.10.0**	Dener, **OPEN**
PHYS Benchmark: Jets - E, mass, FFs (jt, z), Substructure (dRcst, angularity, EECs)	**To-do**	**TBD**	Dener, **OPEN**
PHYS Benchmark: Events - TEECs, NECs	**Some work done**	**TBD**	Derek, **OPEN**

- **Above:** benchmarking/tuning tasks (cont.)
  - **Still room to contribute!**
  - Incl. on physics side (not on critical path)
- **Notes:**
  - EM = “Electromagnetic”, H = “Hadronic
  - dRcst = constituent delta-R

Backup | eID Milestones





- **Above:** flowcharts to illustrate PF + eID workflows after each milestone
  1) Existing eID workflow w/o truth info + prototype developments
  2) Integration w/ PF

Backup | Milestone 1 eID Tasks

Task	Description	Issue/PR/Note	Exp Campaign	Assignees
eID0	☑ Resolve missing EMCal-track associations	03.10 Inclusive Mtg	**DONE**	Diego, Barak
eID1	☑ Resolve issues with boost.h	PR Merged	**DONE**	Stephen
eID2	Add isolation cut to DIS electron selection	EICrecon#2404	**26.09.0**	Stephen
eID3/PFA-1	Replace MatchClusters, with pure reco equivalent	EICrecon#1956	**26.09.0**	Tristan, **OPEN**
eID4	Wire reco DIS electrons into kinematics (step 1)	EICrecon#2333	**26.08.0**	**OPEN**

Milestone 1:

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

ETA: Final tasks should be done by 26.09.0 campaign

- **Notes:**
  - eID4 very straightforward, but will be crucial for benchmarking milestone 2 developments
  - Reco kinematics in eID4 *do not* have to be made default until collaboration is ready

Backup | Toward Milestone 2 eID Status

Task	Description	Issue/PR/Notes	Exp Campaign	Assignees
eID5	Wire reco DIS electrons into kinematics (step 2)	EICrecon#2333	**26.10.0**	**OPEN**
eID6	Improve DIS selection (adding p_T ranking, etc.)	**TBD**	**CY26.Q3**	**OPEN**
eID7	Improve kinematic calculations (different beams)	**TBD**	**CY26.Q4**	**OPEN**
eID8/PFA4	Integrating PFAlpha + eID	Contingent on PF	**CY26.Q4**	**OPEN**

Milestone 2: overlaps with event reco priority to some extent

1. Further improvements to DIS selection
2. Integration with PF

ETA: still needs more planning to determine scope of work of a few tasks

- **Notes:**
  - These tasks might be better done sequentially to assess impact
  - Tough to assign specific campaigns for eID6 – eID8 before having clearer picture of work scope

Backup | Longer Term eID/lepID Tasks

- **Below:** longer term eID/event reco topics to keep on our radar

- **Longer Term Topics:**
  - **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

- Longer term topics (cont.)
  - **Standardized benchmarks + assessing backgrounds**
    - Critical for performance eval.
  - **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?
  - **Generalized lepton ID**
    - IDing e^-, μ^-, τ^- crucial to many EIC physics programs
    - Should take principled, unified approach to lepton ID

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

Backup | PFA-1 | 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

- **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(b) | Track-Protocluster Match Promoter

- **Track-Protocluster Match Promoter:** turns track-protocluster matches into track-cluster matches
  - Implemented in EICrecon#2293

- **Algorithm:**
  1) For each track-protocluster match, i:
     a) Form track-cluster match between track and cluster i

Inputs:
- Track-protocluster matches
- Protoclusters
- Clusters

Outputs:
- Track-cluster matches

Parameters:
- None!

Backup | PFA1(b) | Charged Particle Candidate Maker

- **Charged Candidate Maker:** forms track-cluster matches into a charged particle candidate
  - Implemented in EICrecon#2124

- **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:
- None!

Backup | PFA2 | Calo Remnant Combiner

- **Calo Remnant Combiner:** combines remnant clusters from subtractor into neutral particle candidates
  - In progress at EICrecon#2195

- **Algorithm:**
  1) Combine nearby ECal, HCal clusters
     a) Identify seed ECal cluster
     b) Merge all ECal, HCal clusters in Δr_{add}^{em}, Δr_{add}^{h} 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_{add}^{h} 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

Parameters:
- $\mathbf{\Delta r_{add}^{em}}$: window to add ECal clusters
- $\mathbf{\Delta r_{add}^{h}}$: 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
  - In progress at EICrecon#2399

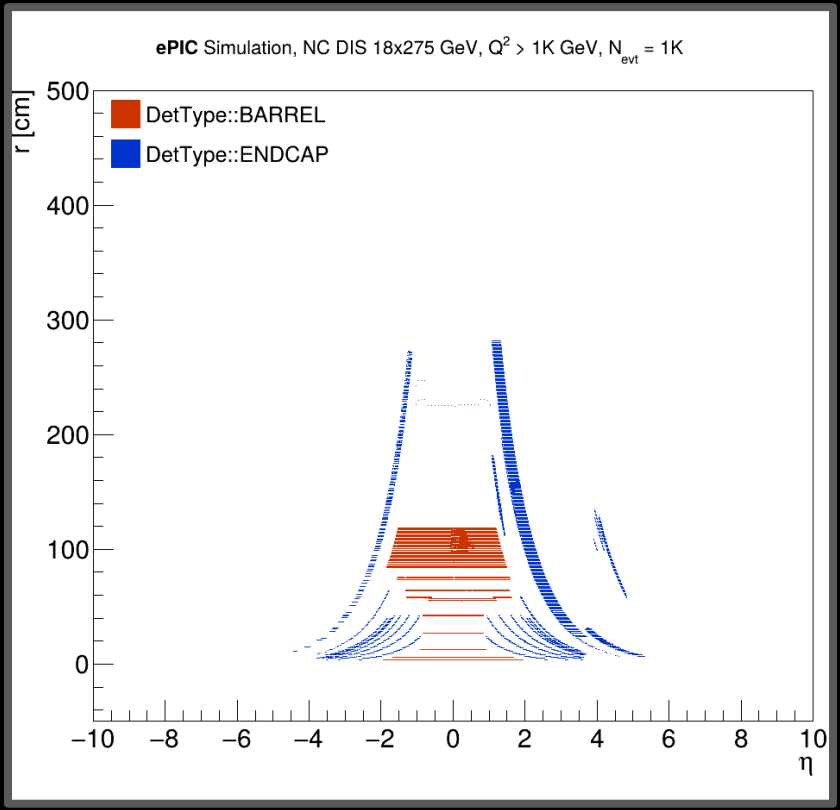
- **Algorithm:**
  1) Assign preliminary PID based on what info is available (e.g. no hcal clusters → electron, photon, or pi0)
  2) Calculate track energy
  $$E_{trk} = p_{trk} \oplus m_{pid}$$
  3) Calculate calorimeter energy
  $$E_{cal} = N_{cal}\left(\sum w_{em}E_{em} + \sum w_h E_h\right)$$
  4) 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

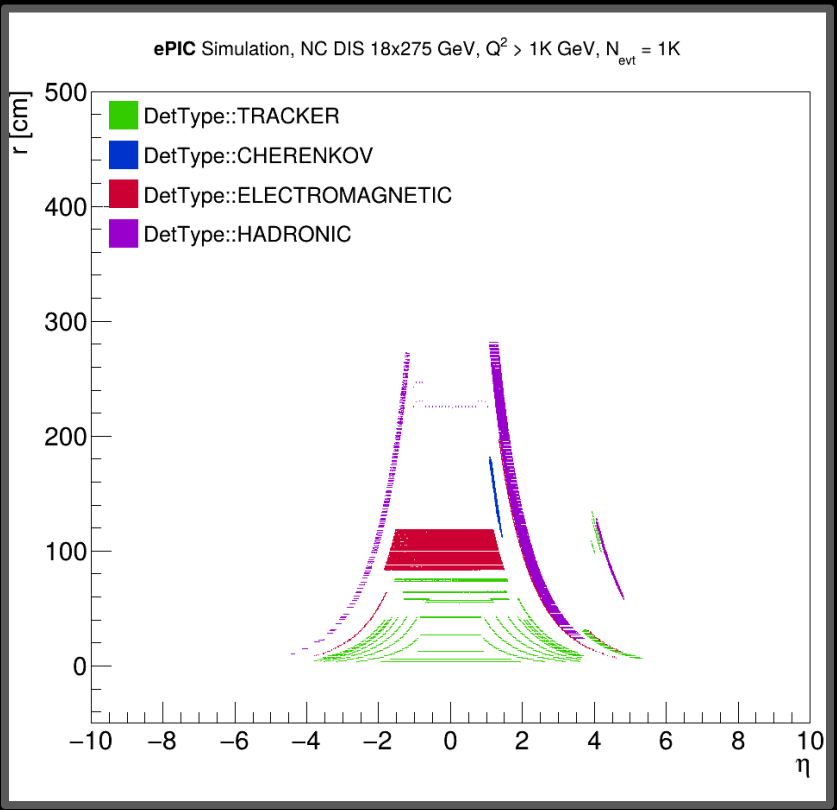
Inputs:
- Candidate charged/neutral particles
- Primary vertices (for neutral candidates)

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 | EMCal vs. HCal





- 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