PF and eID Status Updates

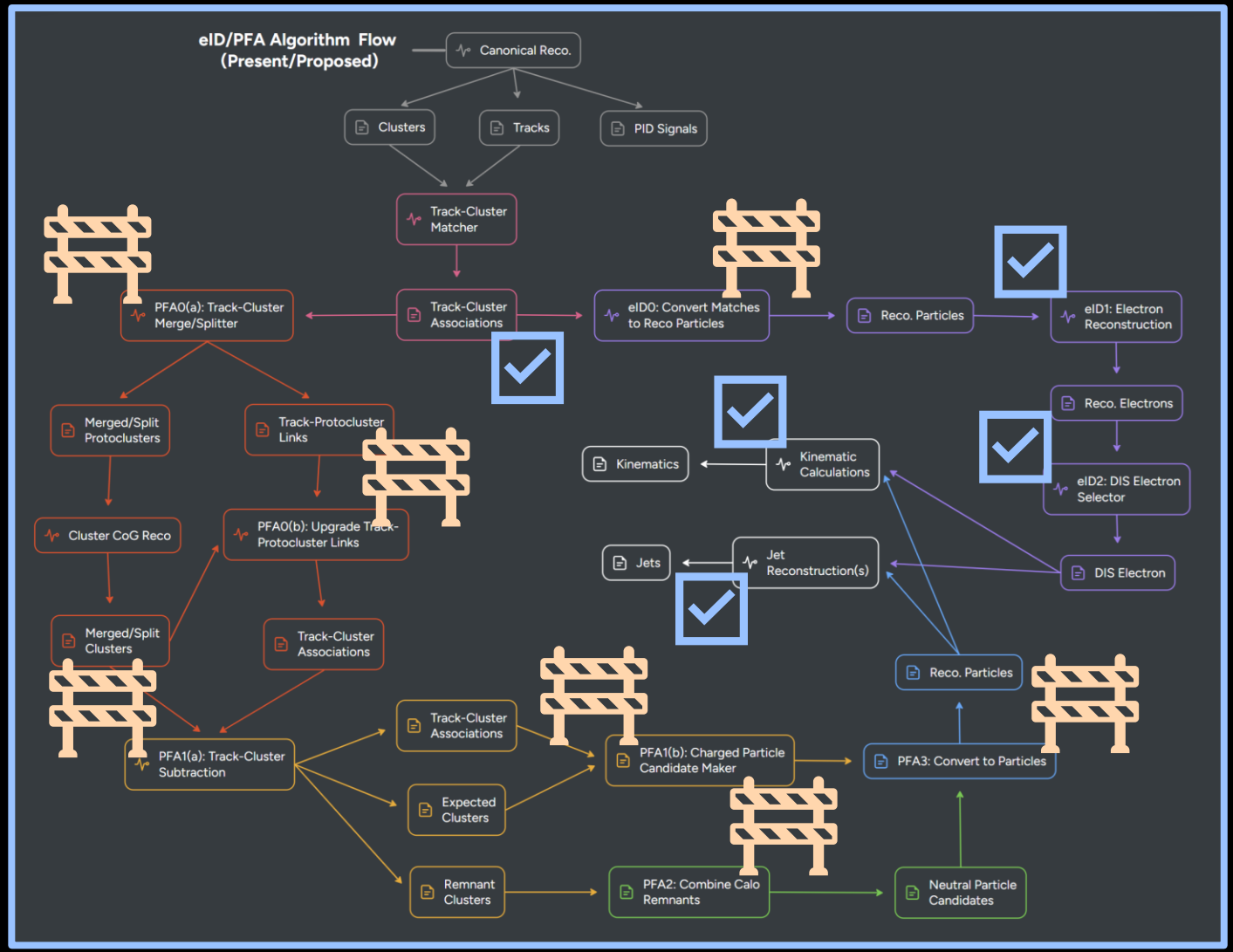
EIC/ePIC Physics Readiness Workshop

Win Lin, Stephen Maple, Derek Anderson

03.19.2026

PF Status | Recent Updates

- **Last update:** 02.11 Physics/S&C Discussion
  - Since then, PR reviews in progress
  - Still on track for 26.04 target
- Now ramping up benchmarking/tuning effort
  - Restarted PF Tech discussions on 03.03
  - **New workforce!**
    - Mounika (IISER-Tirupati)
    - Maciej (AGH University)



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

PF Status | Development Tasks

Task	Description	Issue/PR/Note	Exp Campaign	Assignees
PFA-1	Deprecate MatchClusters, replace w/ pure reco equivalent	EICrecon#1956	**26.05.0**	Tristan, **OPEN**
PFA0(a)	Complete merge/splitter update	EICrecon#1699	**26.04.0**	Derek
PFA0(b)	Implement track-protocluster link promotion algorithm	EICrecon#2293	**26.04.0**	Derek
PFA1(a)	Revive and finish track-cluster subtractor	EICrecon#1627	**26.04.0**	Derek
PFA1(b)	Track-cluster converter (synergy w/ PFA-1)	EICrecon#2124	**26.04.0**	Derek
EDM0	Flagging ecal vs. hcal clusters	epic#994	**26.04.0**	Derek
PFA2	Implement calo remnant combiner	EICrecon#2195	**26.04.0**	Subhadip
PFA3	Implement particle regressor/convertor	EICrecon#2399	**26.04.0**	Esteban
INT	Tie all stages into a single workflow	EICrecon#900	**26.04.0**	**OPEN**

- **Above:** development tasks
  - All critical-path stages have PRs open and are under review (except INT)
  - **Goal is still to merge by 26.04 campaign**

- **Note:** PFA-1 is not on critical path for PF
  - But is relevant for eID development, and benchmarking PF performance

PF Status | 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**	**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	**To-do**	**≤ 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	**To-do**	**≤ 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 | 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**	**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**	**≤ 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

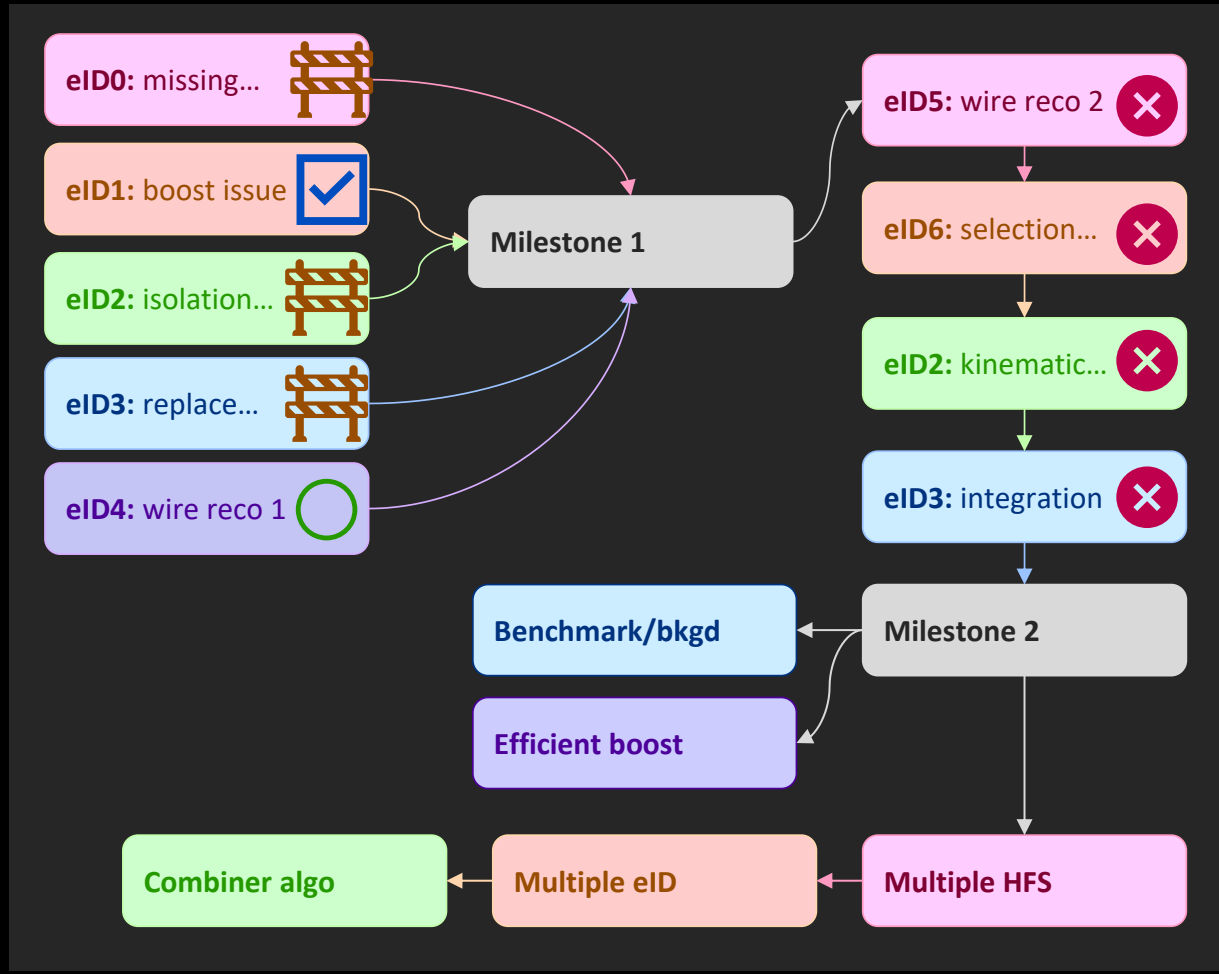
eID Status | eID Task Flowchart

- **Last update:** at 02.11 Physics/S&C Discussion
  - Making steady progress
  - **Right:** task flowcharts for eID development

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

  ☞ Visualizers in backup

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



eID Tasks | Toward Milestone 1

Task	Description	Issue/PR/Note	Exp Campaign	Assignees
eID0	Resolve missing EMCal-track associations	03.10 Inclusive Mtg	**26.05.0**	Diego, Barak
eID1	☑ Resolve issues with boost.h	PR Merged	**26.03.0**	Stephen
eID2	Add isolation cut to DIS electron selection	EICrecon#2404	**26.04.0**	Stephen
eID3/PFA-1	Replace MatchClusters, with pure reco equivalent	EICrecon#1956	**26.05.0**	Tristan, **OPEN**
eID4	Wire reco DIS electrons into kinematics (step 1)	EICrecon#2333	**26.05.0**	**OPEN**

Milestone 1:

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

ETA: Most should be ready by 26.05.0 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

eID Tasks | Toward Milestone 2

Task	Description	Issue/PR/Notes	Exp Campaign	Assignees
eID5	Wire reco DIS electrons into kinematics (step 2)	EICrecon#2333	**26.06.0**	**OPEN**
eID6	Improve DIS selection (adding p_T ranking, etc.)	**TBD**	**CY26.Q2**	**OPEN**
eID7	Improve kinematic calculations (different beams)	**TBD**	**CY26.Q3**	**OPEN**
eID8/PFA4	Integrating PFAlpha + eID	Contingent on PF	**CY26.Q3**	**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

eID Status | Longer Term eID/lepID Tasks

- **Below:** longer term eID/event reco topics to keep on our radar, 1st advertised at CM
  - Generalized lepton ID is new! Will discuss more in Stephen's talk

- **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

Thanks!

Questions/Comments?

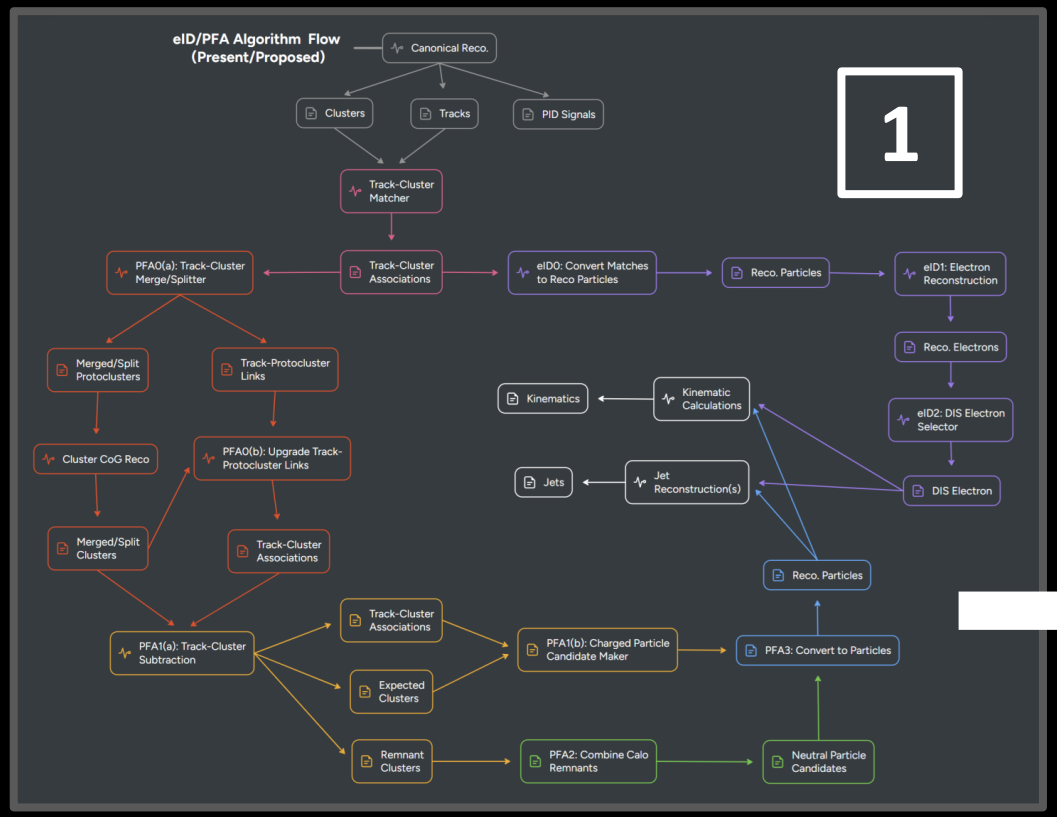
Summary

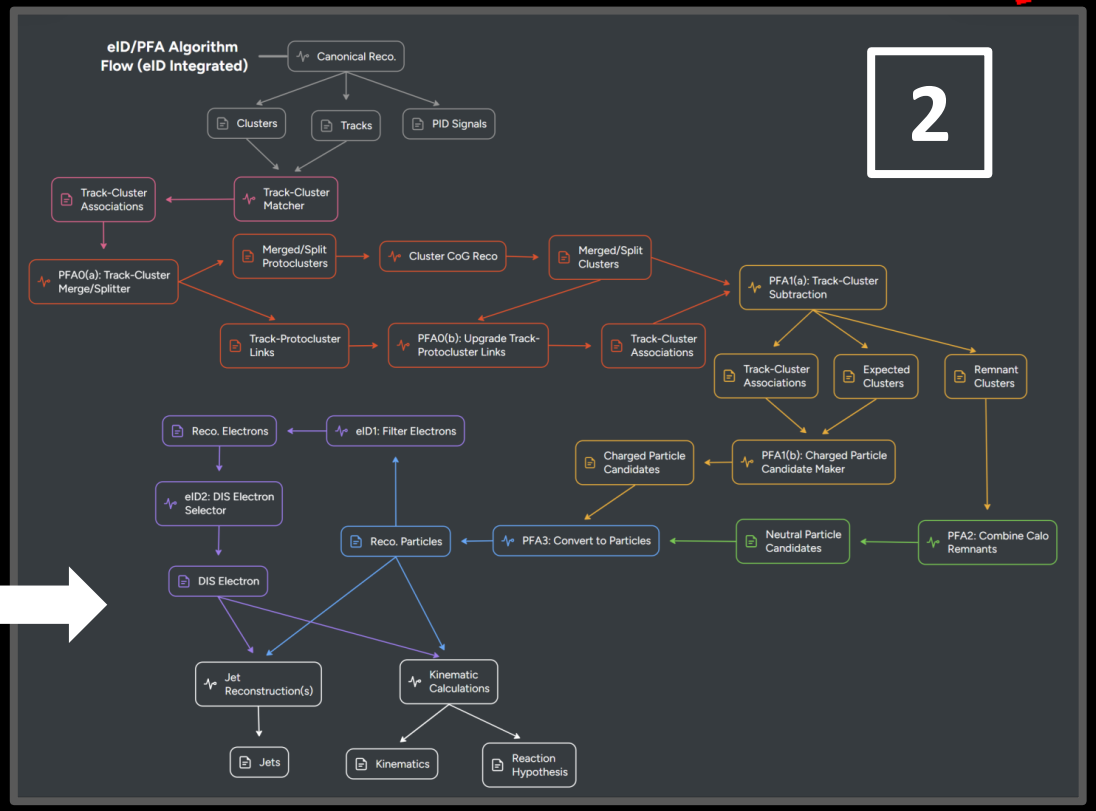
- PF development working towards 26.04 goal, benchmarking effort ramping up
- eID development is underway, lots of room to contribute!

PF/eID Contributors

- Maciej Blaut (AGH University)
- Daniel Brandenberg (OSU)
- Esteban Molina Cardenas (UMich)
- Mounika Choppali (IISER-Tirupati)
- Diego Cristancho (UH)
- Win Lin (SBU)
- Stephen Maple (Birmingham)
- Subhadip Pal (CTU)
- Tristan Protzman (LU)
- Dmitry Kalinkin (BNL)
- Tyler Kutz (Mainz)
- Barak Schmookler (UH)

Backup | eID Milestones

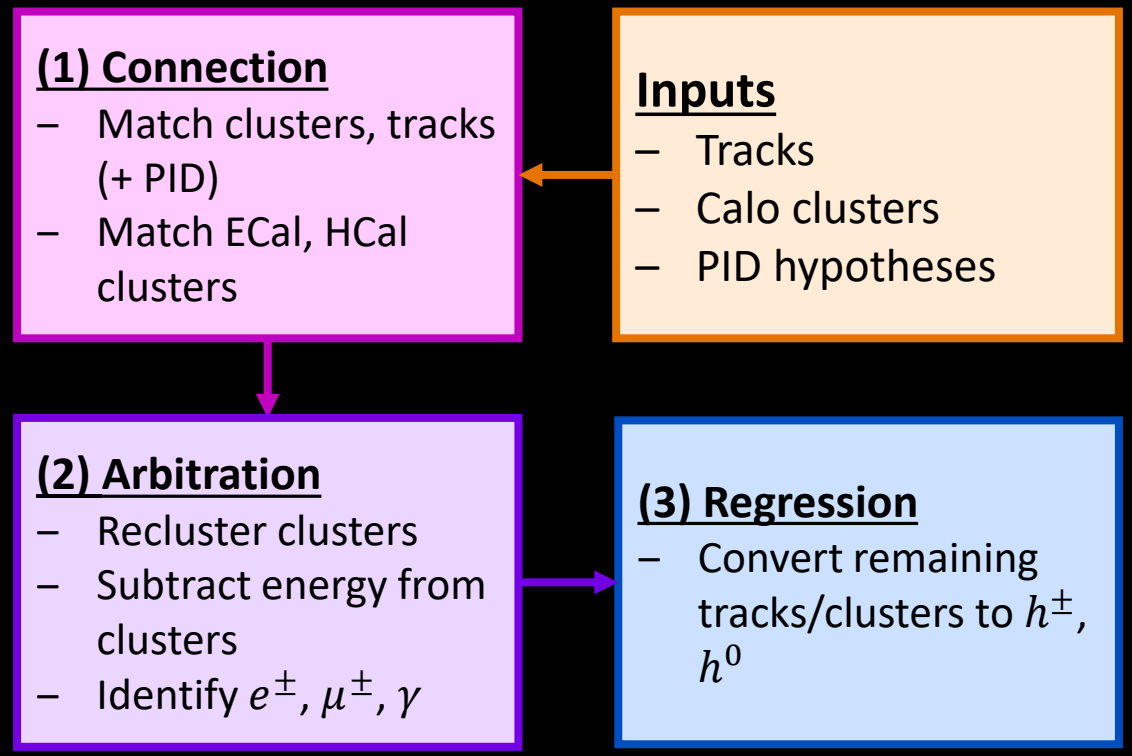




- **Above:** flowcharts to illustrate PF + eID workflows after each milestone
  1) Existing eID workflow w/o truth info + prototype developments
  2) Integration w/ PF

Backup | Particle Flow Alpha (1/2)

- **Right:** schematic representation of a general EF/PF-like workflow
  - Summarizes key stages of such a workflow
  - Lists illustrate typical tasks in each stage
- Started working towards a simplistic "bare bones" PFA in 2023 which will
  a) Function as baseline to compare against more sophisticated PFAs
  b) Improve our full (charge + neutral) jet reconstruction
  c) Address gaps in ePIC's *holistic* reconstruction
- **When starting:** several of these operations didn't exist in our reconstruction software!



Backup | Particle Flow Alpha (1/2)

- **Particle Flow Alpha (PFAlpha):** name given to our baseline algorithm
  - Aiming at being (relatively) simple EF-like algorithm, *but* still fills in critical gaps in reco
  - And **critically** is designed to be *modular*

- **Initial conception:**
  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 reconstructed particles**
     - Tracks
     - Subtracted, merged clusters

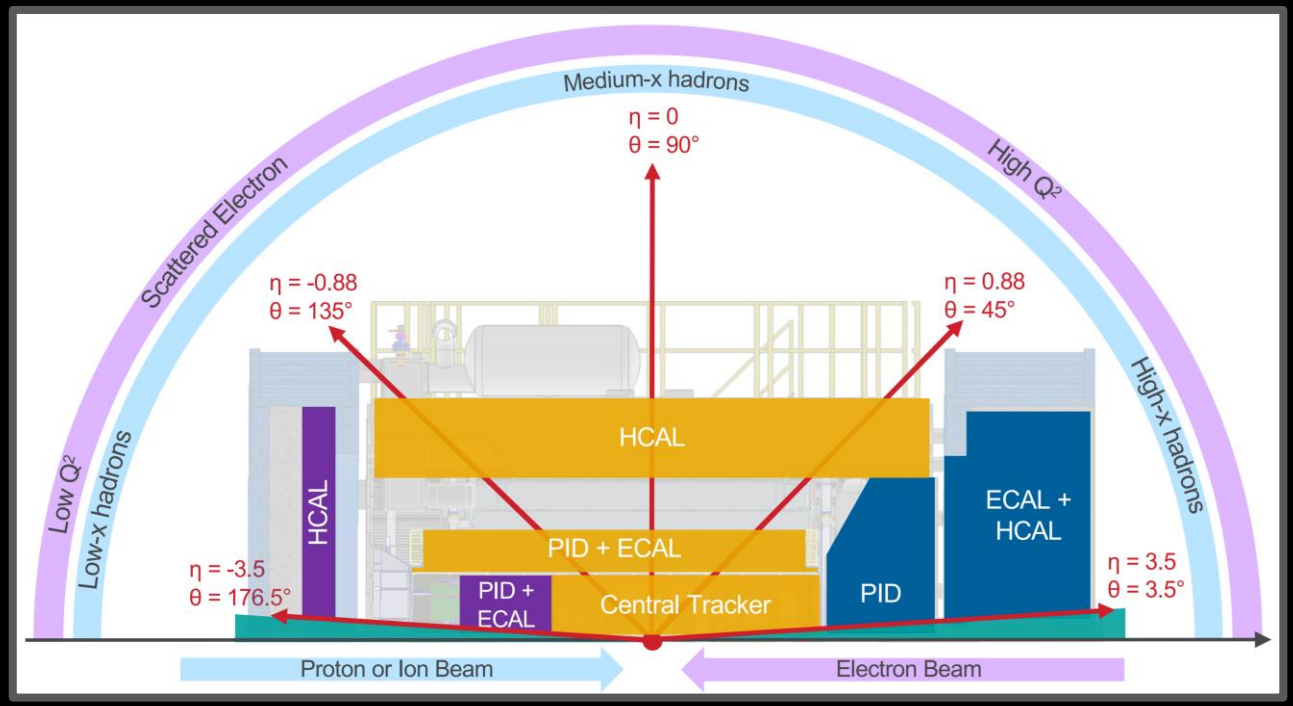
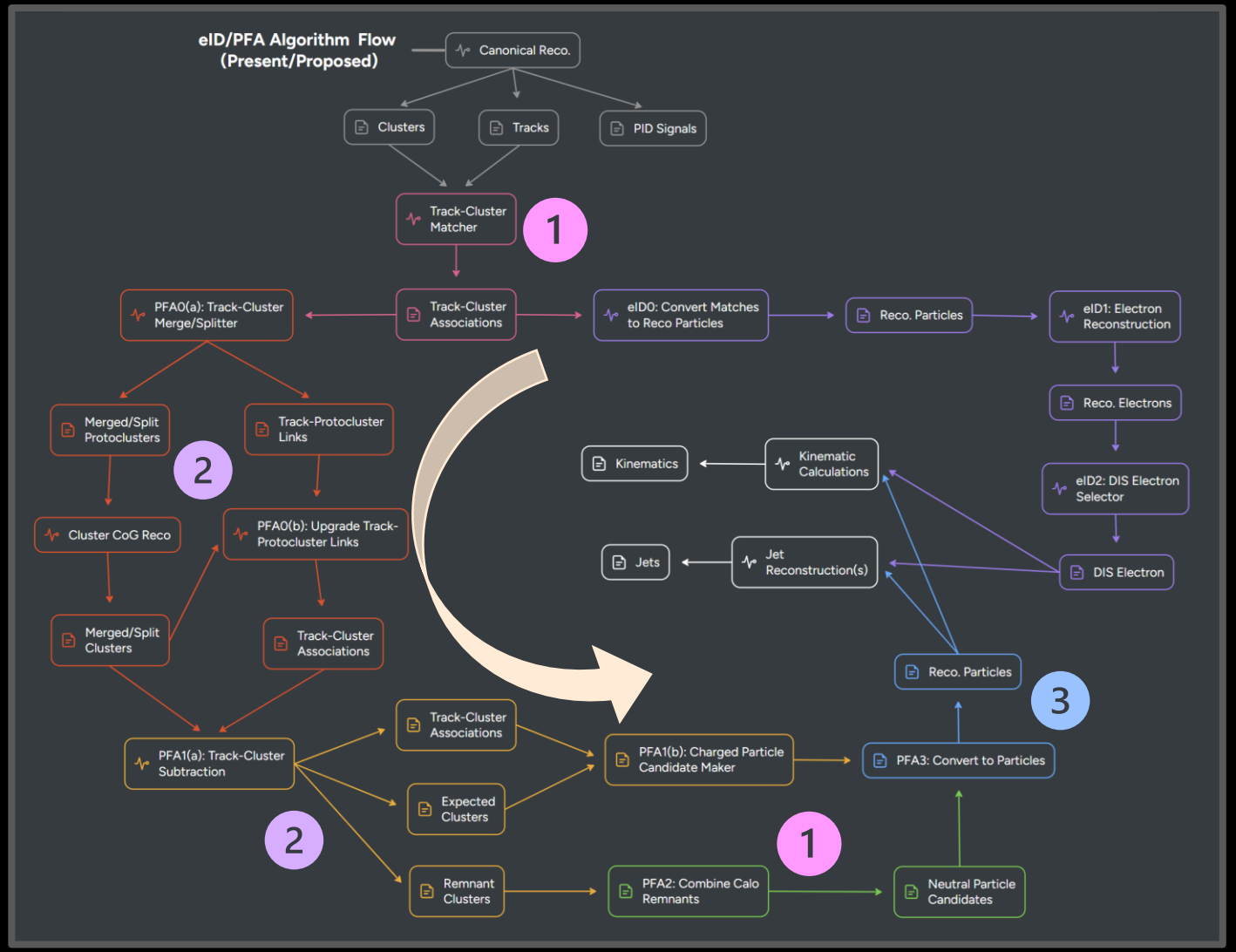


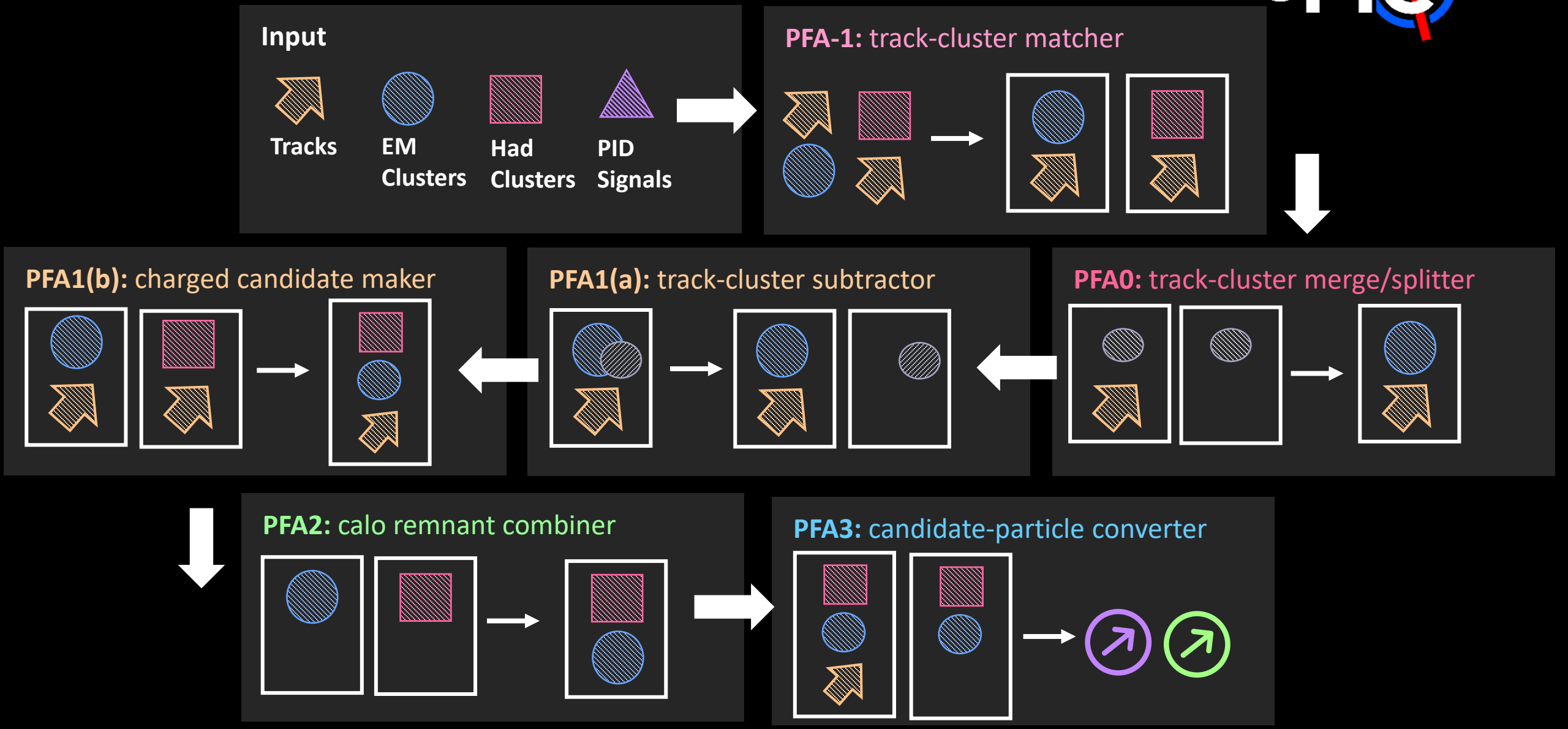
Figure by Sylvester Joosten

Backup | Algorithm Overview (1/2)

- **Since kickoff:** actual algorithm significantly evolved
  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 backup



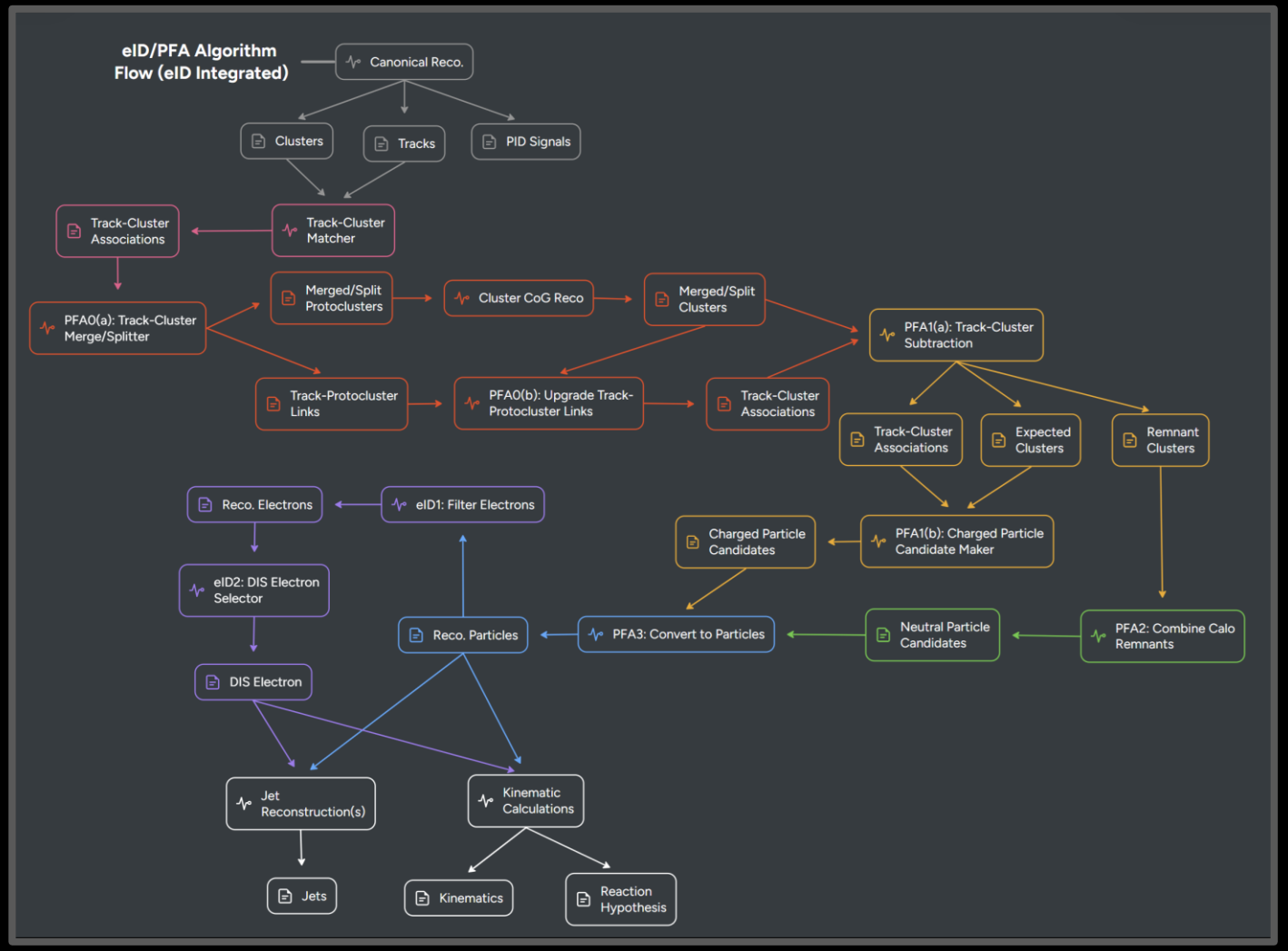
Backup | Algorithm Overview (2/2)



Backup | Future Developments

- **After PFAlpha:** critical next step is integrating PF baseline with existing electron ID (eID)
  - Will be 1st step towards more rigorous lepton/hadron separation
  - **Right:** possible algorithm flow after integration w/ eID

- **Other key next steps:** which will define evolution towards **PFBeta**
  - Integration of PID detectors into algorithm
    - ☞ Holistic PID is another reco priority this year!
  - Integration of additional lepton ID, *especially* muon ID
  - More principled topoclustering (cross-calorimeter) algorithms
  - Exploration of ML techniques



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

- **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)

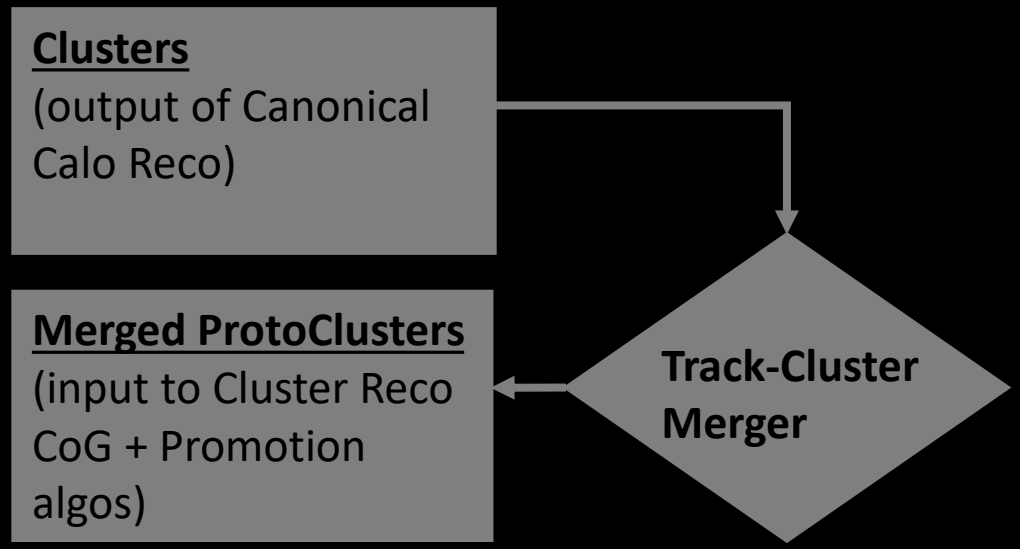
Backup | PFA0 | Track-Cluster Merge/Splitter (1/2)

- **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}:

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

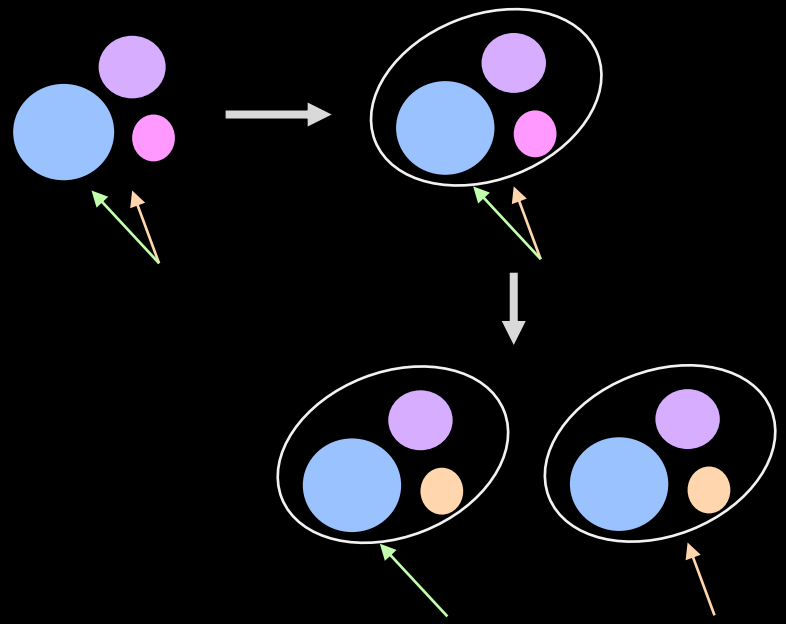
  3) If $S < S_{cut}$, add clusters inside Δr_{add}
  4) If multiple tracks pointing to merged cluster:
     - Split into one cluster for each track & reweight transverse shape by p_{trk}



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

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}:

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

  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

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:
- 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

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()

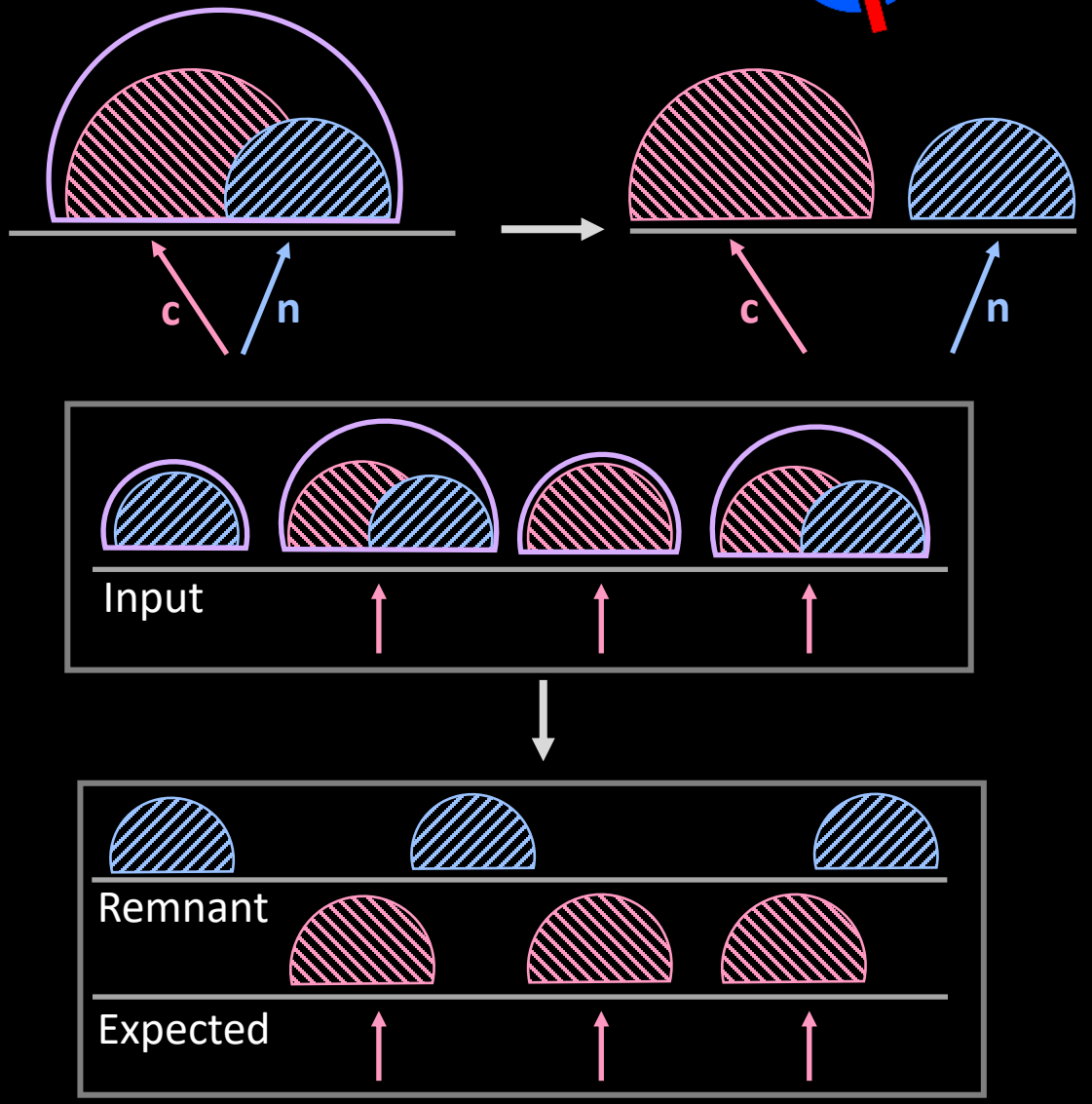
Backup | PFA1(a) | Track-Cluster Subtractor (3/3)

- **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!

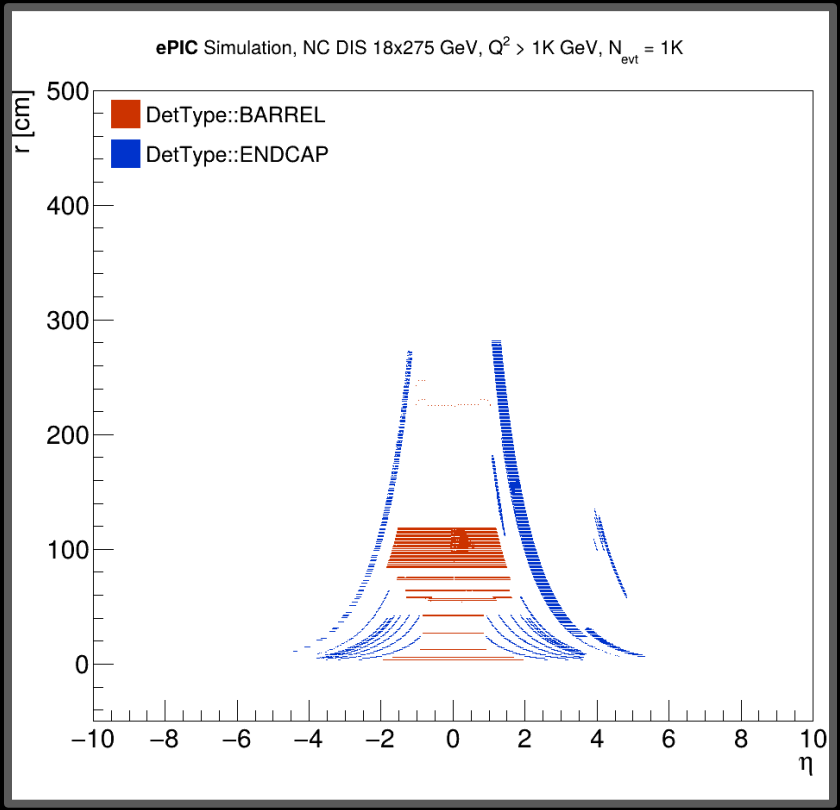
- **The algorithm:**
  1) Assign preliminary PID based on what info is available (e.g. no hcal clusters → electron, photon, or pi0)
  2) Calculate track energy
  $$E_{trk} = p_{trk} \oplus m_{pid}$$
  3) Calculate calorimeter energy
  $$E_{cal} = N_{cal}\left(\sum w_{em}E_{em} + \sum w_{h}E_{h}\right)$$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

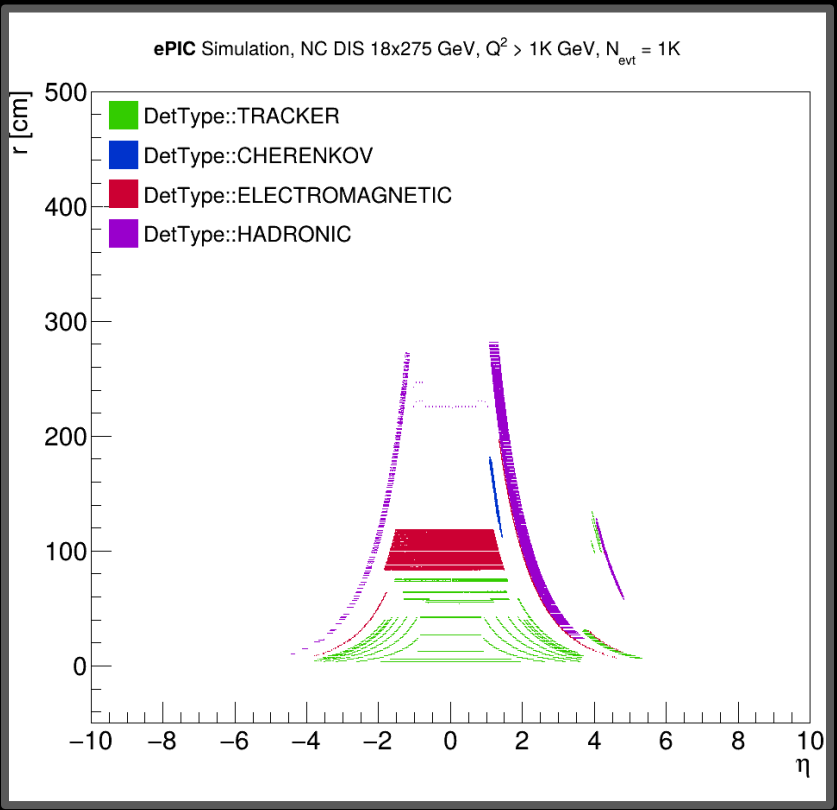
Inputs:
- Candidate charged/neutral particles
- Primary vertices (for neutral candidates)

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 | 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