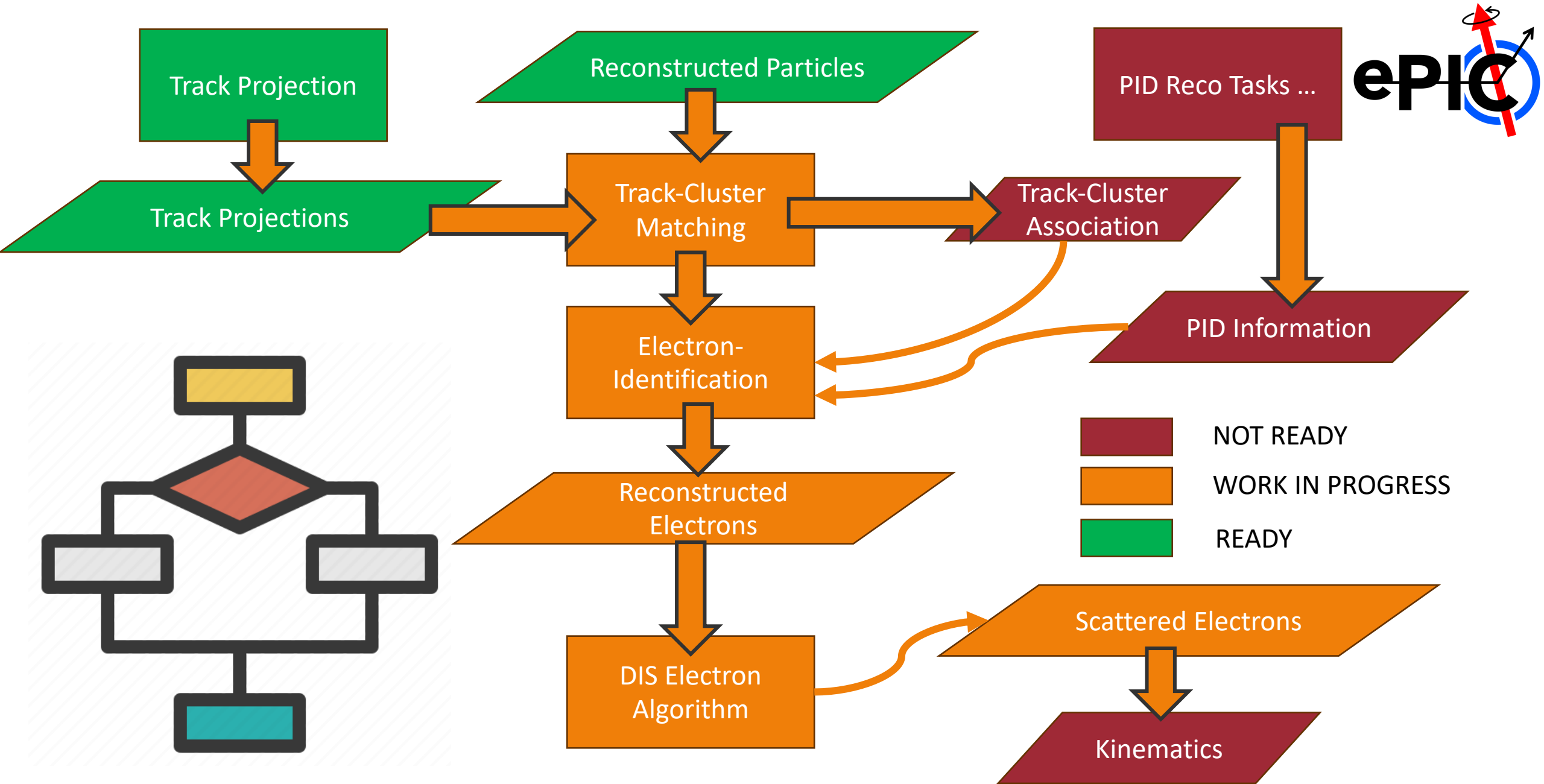


# DIS Electron Finding in ePIC TDR Readiness

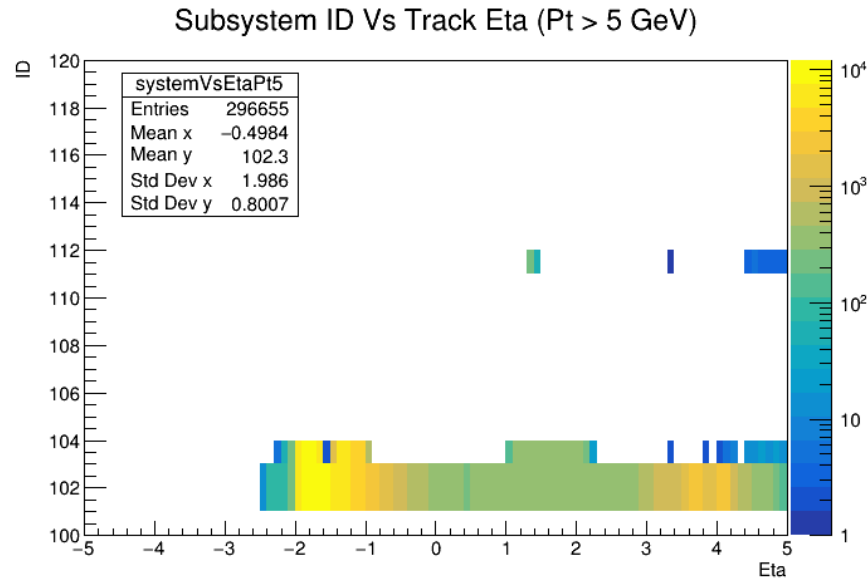
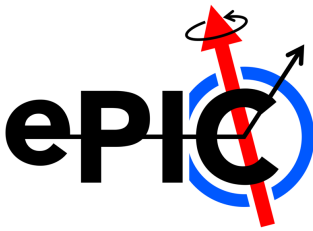
Daniel Brandenburg (point of contact)

BRANDENBURG.89@OSU.EDU

**SPECIAL THANKS:** Tyler Kutz, Brian Paige,  
Tristan Protzman, Temple U, Kong Tu, Andrii  
V, Markus D., Wouter D. Dmitry K. ....



# Track Projections - Resolved



Track projections show ID (detector id) at eta values that should not be possible

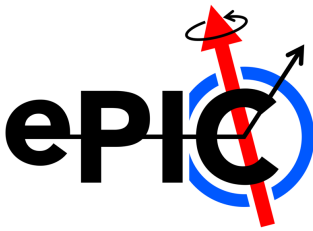
Thanks to Tyler K., Brian P. and OSU students for working on this.

Thanks to software experts for help

- TrackPropagation\_factory: resolve an indexing bug by using ACTS surface ID fields #1257
- Fixed and merged into eicrecon
- Brian confirmed it looks good except that there is a typo mixing up endcap CALOs
- PR hotfix in prep



# Track Cluster Matching



- Initial Track-Cluster Matching algorithm in development
- Association between Reco Cluster and ReconstructedParticle #52

Blocking Issue:

- Tracks are not filled
- Skip directly to Reconstructed Particles
- No relationship for associating projection<->track<->reconstructed particle

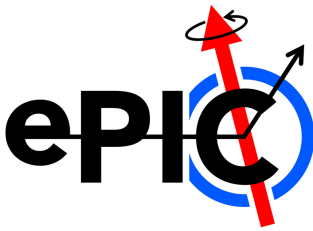
Solution:

Tracks as first-class citizen

Update to data-model

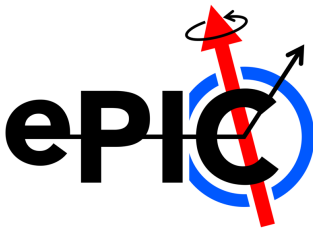
```
495 +  
496 +   edm4eic::TrackClusterMatch:  
497 +     Description: "Association between a Cluster and a ReconstructedParticle"  
498 +     Author : "D. Brandenburg"  
499 +     Members:  
500 +       - uint32_t          cluID          // Index of corresponding cluster (position  
         in cluster array)  
501 +       - uint32_t          plcID          // Index of corresponding  
         ReconstructedParticle (position in reco particle array)  
502 +       - float            weight         // weight of this association  
503 +     OneToOneRelations:  
504 +       - edm4eic::Cluster  clu           // reference to the cluster  
505 +       - edm4eic::ReconstructedParticle plc // reference to the Reco  
         particle  
⊖
```

# DIS Lepton Finder Algorithms

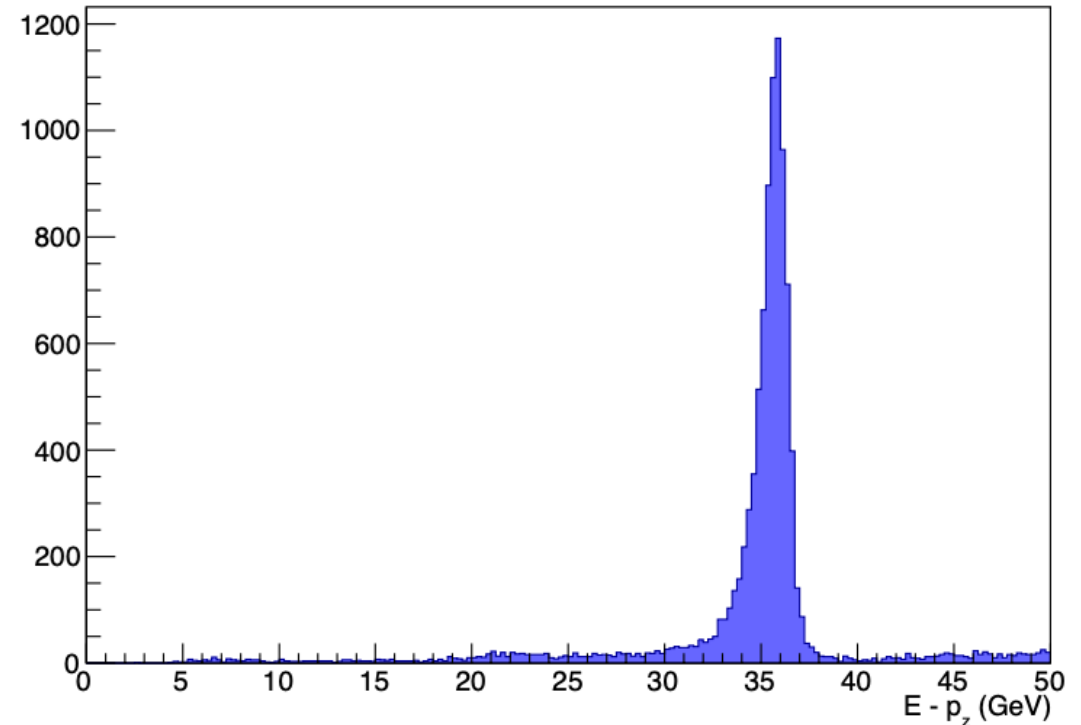


- So far two variants:
- Algorithms and factories for scattered electrons #1277
- `ScatteredElectronsTruth` – finds MC scattered Electron and get associated reconstructed particle – collection of length 1, always the true\* electron
- `ScatteredElectronsEMinusPz` – computes E-Pz for electron candidates and corresponding hadronic final state
- \*seems there is some issue in Pythia with the identification of the scattered electron from MC info...

# E-Pz algorithm

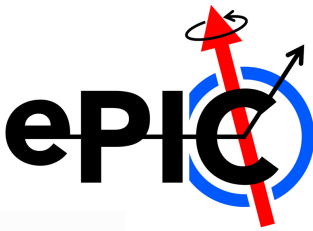


- Track matching next milestone for fully reconstruction-based  $E/p$  cut
- Further ePID refinement with addition of  $E - p_z$  cut (thanks to K. Tu for help on implementation)
- Implementation of EICrecon algorithm/factory underway
- Still considering best output format, how to handle multiple DIS electron candidates, etc.



Thank you to Kong Tu and Tyler Kutz

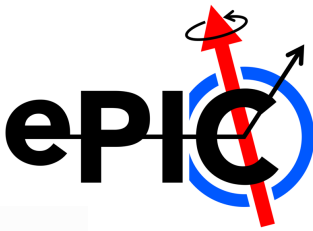
# ScatteredElectronsTruth



```
inline auto find_first_scattered_electron(const edm4hep::MCParticleCollection* mcparts) {  
    return find_first_with_status_pdg(mcparts, {1}, {11});  
}
```

```
template<class T>  
auto find_first_with_status_pdg(  
    const T* parts,  
    const std::set<int32_t>& status,  
    const std::set<int32_t>& pdg) {  
    T c;  
    c.setSubsetCollection();  
    for (const auto& p: *parts) {  
        if (status.count(p.getGeneratorStatus()) > 0 &&  
            pdg.count(p.getPDG()) > 0) {  
            c.push_back(p);  
            break;  
        }  
    }  
    return c;  
}
```

# ScatteredElectronsTruth

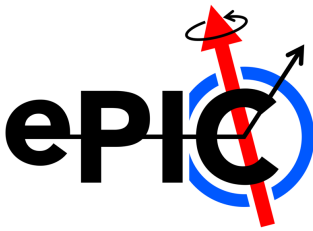


```
// Get first scattered electron
const auto ef_coll = find_first_scattered_electron(mcparts);
if (ef_coll.size() == 0) {
    m_log->trace("No truth scattered electron found");
    return;
}

// Associate first scattered electron
// with reconstructed electron
auto ef_assoc = rcassoc->begin();
for (; ef_assoc != rcassoc->end(); ++ef_assoc) {
    if (ef_assoc->getSimID() == (unsigned) ef_coll[0].getObjectID().index) {
        break;
    }
}
```



# ScatteredElectronsEMinusPz

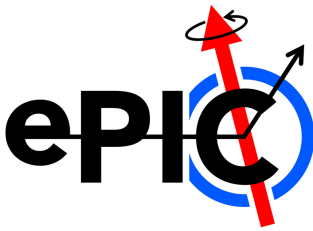


```
for (const auto& p: *rcparts) {  
    // this is a hack - getObjectID() only works within  
    // a collections, not unique across all collections.  
    // What we want is to add all reconstructed particles  
    // except the one we are currently considering as the  
    // (scattered) electron candidate.  
    // This does work though and in general it has only  
    // one match as I would hope (tested on pythia events)  
    if ( edm4hep::utils::magnitude(p.getMomentum())  
        != edm4hep::utils::magnitude(e.getMomentum()) ) {  
        vHadron.SetCoordinates(  
            p.getMomentum().x,  
            p.getMomentum().y,  
            p.getMomentum().z,  
            m_pion // Assume pion for hadronic state  
        );  
  
        // Sum hadronic final state  
        vHadronicFinalState += vHadron;  
    } else {  
        m_log->trace( "Skipping electron in hadronic final state" );  
    }  
}
```

Issue: getObjectID() only works  
within same collection

Solution: use Subset Collection (PR  
in progress)

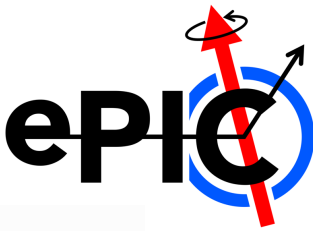
# ScatteredElectronsEMinusPz



```
// Calculate the E-Pz for this electron  
// + hadron final state combination  
// For now we keep all electron  
// candidates but we will rank them by their E-Pz  
double EPz=(vScatteredElectron+vHadronicFinalState).E()  
           - (vScatteredElectron+vHadronicFinalState).Pz();
```

Compute E-Pz for all (negative charge) electron  
candidates identified in ReconstructedElectron collection

# ScatteredElectronsEMinusPz



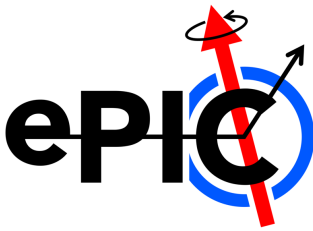
```
// map sorts in ascending order by default
// sort by descending
bool first = true;
// for (auto kv : scatteredElectronsMap) {
for (auto kv = scatteredElectronsMap.rbegin(); kv != scatteredElectronsMap.rend(); ++kv) {

    double EMinusPz = kv->first;
    // Do not save electron candidates that
    // are not within range
    if ( EMinusPz > m_cfg.maxEMinusPz
        || EMinusPz < m_cfg.minEMinusPz ){
        continue;
    }
}
```

Returns 'ScatteredElectronsEMinusPz' collection – ranked with the highest E-Pz first, then in descending order

Even without strict electron PID – right “most” of the time

# Checking the Scattered Electron

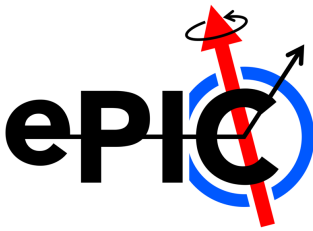


- Independent use of ScatteredElectronsTruth\_factory and ScatteredElectronsEMinusPz\_factory allows us to ‘check’ the algorithm’s performance

```
[reco:ScatteredElectronsEMinusPz] [trace] We have 3 candidate electrons
[reco:ScatteredElectronsEMinusPz] [trace] Skipping electron in hadronic final state
[reco:ScatteredElectronsEMinusPz] [trace] E-Pz=17.044048885959747
[reco:ScatteredElectronsEMinusPz] [trace] ScatteredElectron has Pxyz=( 1.0658381, 2.5452766, -8.107633 )
[reco:ScatteredElectronsEMinusPz] [trace] skipping positron
[reco:ScatteredElectronsEMinusPz] [trace] skipping positron
[reco:ScatteredElectronsEMinusPz] [trace] Selecting candidates with 0 < E-Pz < 10000000
[reco:ScatteredElectronsEMinusPz] [trace] Max E-Pz Candidate:
[reco:ScatteredElectronsEMinusPz] [trace] E-Pz=17.044048885959747
[reco:ScatteredElectronsEMinusPz] [trace] ScatteredElectron has Pxyz=( 1.0658381, 2.5452766, -8.107633 )
[reco:ScatteredElectronsTruth] [trace] We found 1 scattered electrons using Truth assocaiton
[reco:ScatteredElectronsTruth] [trace] TRUTH scattered electron has E-Pz = 17.044048885959747
[reco:ScatteredElectronsTruth] [trace] TRUTH scattered electron has Pxyz=( 1.065838098526001, 2.545276641845703, -8.107632637023926 )
and E/p = 1.0000000507582056
```

These allow ‘baseline’ algorithms to compare against as we develop more sophisticated (ML) algorithms

# Summary



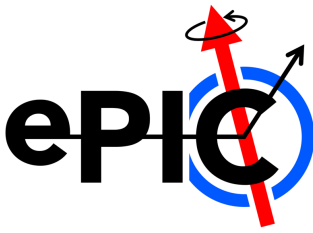
- Next Major Milestones

- Proper use of `Tracks` - how can eFinder group help? (I have a potential post-doc who would like to help)
- Fully integrate Track-Cluster Matching
- Incorporate PID information
- Setup for advanced lepton finder algorithms
- Data format updates needed
  - Track-Cluster Match
  - Generic ScatteredElectron cases (see next slides)
  - Kinematics (inclusive, semi-inclusive, exclusive, etc.)

**SPECIAL THANKS:** Tyler Kutz, Brian Paige, Tristan Protzman, Temple U, Kong Tu, Andrii V, Markus D., Wouter D. Dmitry K. ....

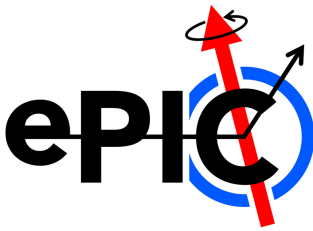


# Data format updates



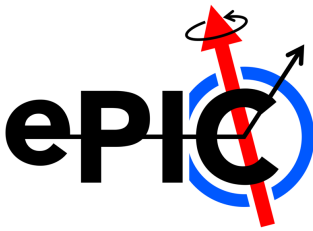
- Track-Cluster Match
- Collection for ranked scattered electrons
  - Something like an object + a likelihood
  - Would be nice to also somehow associate an “algorithm data” object, e.g. E-Pz info for the ScatteredElectronsEMinusPz
- Kinematics formats for:
  - Inclusive
  - Exclusive
  - Etc.

# Proposal



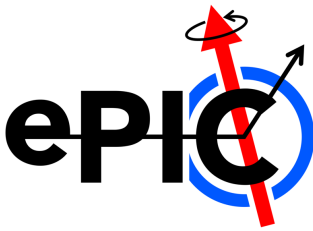
- Instead of having ScatteredElectrons (of type ReconstructedParticleCollection)
- Instead, new structure (simplest case):
- ScatteredElectronHypothesis:
  - EMinusPz
  - RankEMinusP
  - isTruthScatteredElectron
  - EOverP
  - HasLeadingCluster
  - ...
  - OneToOne Relation
    - recoParticle (points to the electron candidate)
  - OneToOne relation:
    - InclusiveKinematics
    - SemiInclusiveKinematics
    - ExclusiveKinematics
    - ...
- Also OneToMany relation on ReconstructedParticle (like PID)

# Proposal (more general)



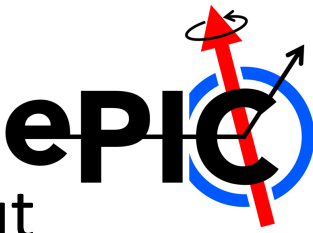
- ScatteredElectronHypothesis:
  - Algorithm Identifier
  - Ranking – single value which tells end user the relative RANK according to this algorithm
  - DATA member based on algorithm
  - OneToOne Relation
    - recoParticle (points to the electron candidate)
  - OneToOne relation:
    - InclusiveKinematics
    - SemilnclusiveKinematics
    - ExclusiveKinematics
    - ...
- ScatteredElectronHypothesis can then be a single collection (for all algorithms) or new collections can be added for new algorithms (using the same type)

# Electron Finder 07-10-2023



- Progress in June (for July Sim Campaign)
  - PR #666: Provides association containers + truth associations merged into main on June 24
  - PR #751: Implement basic electron finding with truth cluster matching
  - Work in parallel with the RECO approach (see tasks below)
- Major Tasks (July):
  - Utilize the July sim campaign output:
    - Implement a processor to test DIS lepton finder
    - Check purity of selected electrons
  - Track Projection Factory: provide track projections at relevant detectors (Tyler Kutz)
    - ✓ Needed data structure identified
    - ✓ Prototype factory in progress
  - Track Match Factory: Matching of projecting tracks to clusters (volunteer?)
    - Nicholas Schmidt already has some code (processor) to study track matching
    - Provides a starting point for factory
  - Study of E/p cuts to implement (volunteer potentially identified, discussing next steps)
  - Study HCAL info for hadron rejection / electron id
- Plans for July sim campaign
  - Utilize “ReconstructedElectrons” to test-drive DIS lepton finder (should be in EICRecon for Aug)
  - Continue work towards towards fully RECO level (complete track matching / compare to truth level matching)

# Truth approach



- PR #751 Add reconstructed electron factory, algorithm utilizing E/p cut

- <https://github.com/eic/ElCrecon/pull/751>

- ReconstructedElectrons Factory

- Input:

```
75 | {"MCParticles", "ReconstructedChargedParticles", "ReconstructedChargedParticleAssociations",  
76 | "EcalBarrelScFiClusterAssociations",  
77 | "EcalEndcapNClusterAssociations",  
78 | "EcalEndcapPClusterAssociations",  
79 | "EcalEndcapPInsertClusterAssociations",  
80 | "EcalLumiSpecClusterAssociations",  
81 | },
```

- Output: “ReconstructedElectrons”

- Utilizes the ElectronReconstruction Algorithm

- Any track with an ECAL match
  - Accept if  $0.9 < E/p < 1.2$  (needs to be studied and optimized)
  - TODO: use HCAL
  - TODO: handle multiple matches

- This is meant to be initial skeleton – keep same structure for RECO approach