Analysis of ePIC Output With Afterburned Events

Alex Jentsch, Jihee Kim, Brian Page
Brookhaven National Lab
April 17th, 2024

Updated: May 20th, 2024

Introduction

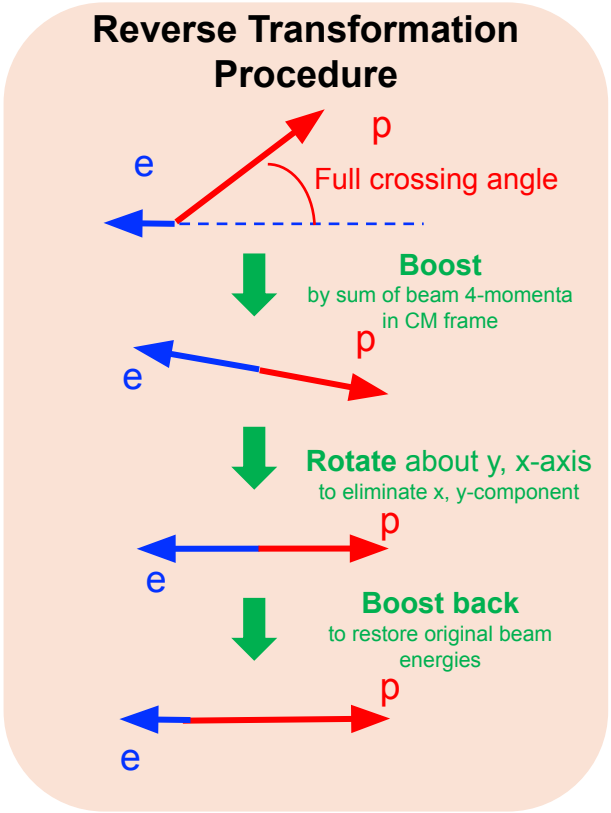
- **<u>What does the afterburner actually do?</u>**
  - Most event generator writes events in head-on frame (no crossing angle or beam effects)
  - “Afterburner” introduces **beam crossing angle** and **beam effects (angular divergence & momentum spread)** to events and changes from head-on frame to Lab frame
    - Lab frame: Electron beam is along z axis (solenoid axis is aligned with electron beam) and Hadron beam takes full crossing angle
  - That results in all particles being **boosted** and **rotated** including addition of beam effects and those are saved into MCParticles in “Afterburned” HepMC file

Introduction

- **What does the afterburner actually do?**
  - Most event generator writes events in head-on frame (no crossing angle or beam effects)
  - "Afterburner" introduces **beam crossing angle** and **beam effects (angular divergence & momentum spread)** to events and changes from head-on frame to Lab frame
    - Lab frame: Electron beam is along z axis (solenoid axis is aligned with electron beam) and Hadron beam takes full crossing angle
  - That results in all particles being **boosted** and **rotated** including addition of beam effects and those are saved into MCParticles in "Afterburned" HepMC file
- **True MC** info from event generator is *missing* in "Afterburned" HepMC file
- However, we can retrieve True MC info!
- To extract True MC information from "Afterburned" file
  - Need to do reverse transformation from Lab frame to Head-on frame (**"post-burn"**)

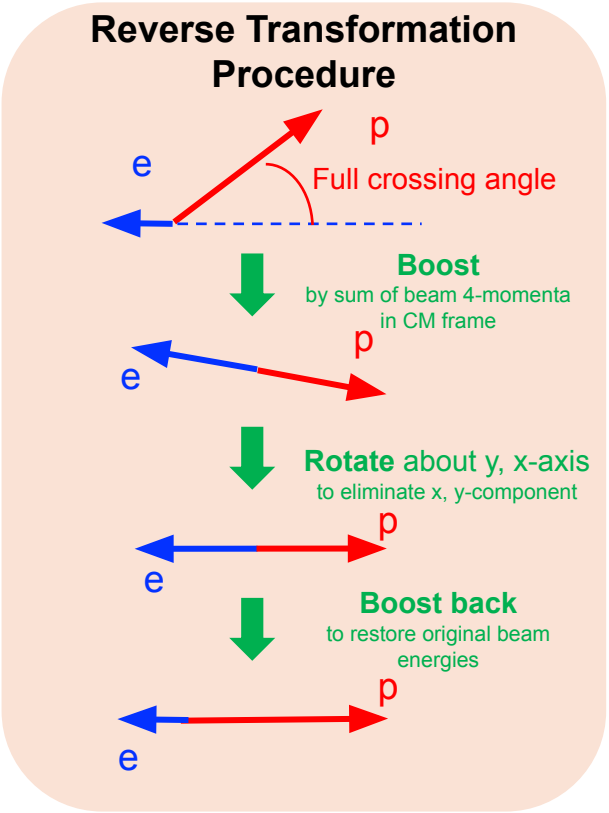
How to Get Back to True MC from Afterburned

Boost and rotation is determined by beam particles of event (event-by-event)



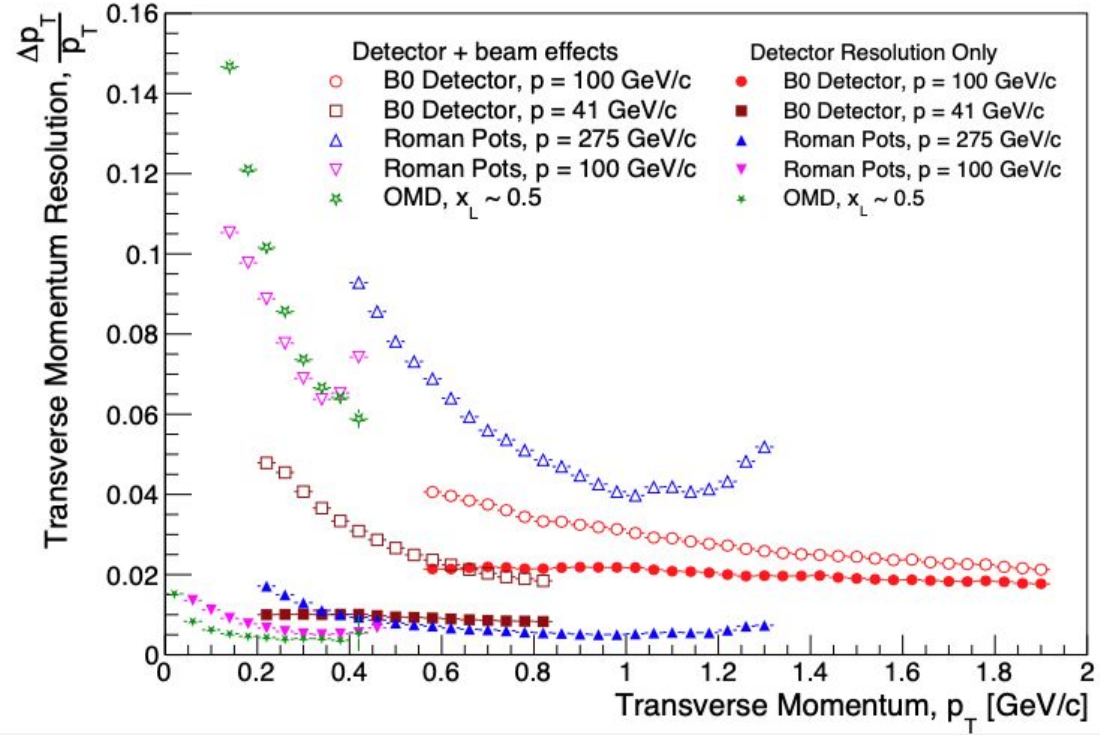
How to Get Back to True MC from Afterburned

Boost and rotation is determined by beam particles of event (event-by-event)



- **Two approaches**
  - **True MC events** for true MC to full-reco comparison
    - Remove ALL effects to get back generated MC information.
  - **Realistic events** remove only crossing angle, keep beam FX in events
    - This is an important distinction!! Crossing angle can be accounted-for in real events, but beam effects such as angular divergence are random and cannot be removed from a real event!

Beam effects vs. detector effects



- Beam effects are the dominant source of smearing for most of the FF detectors.
- Even in the main detector, the will have an impact on analysis, and users should have the tools to separate various smearing contributions.

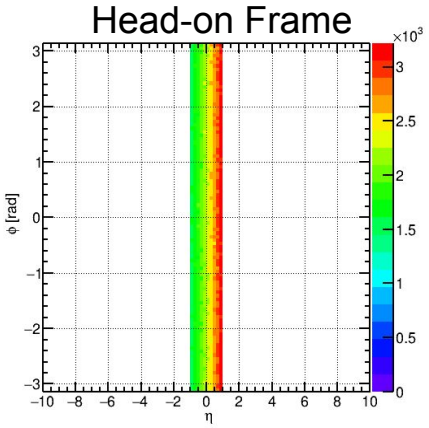
PYTHIA ep 18×275 GeV^2 Sample

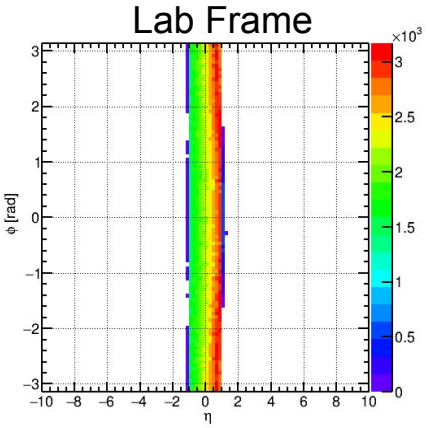
Look into different pseudo-rapidity region

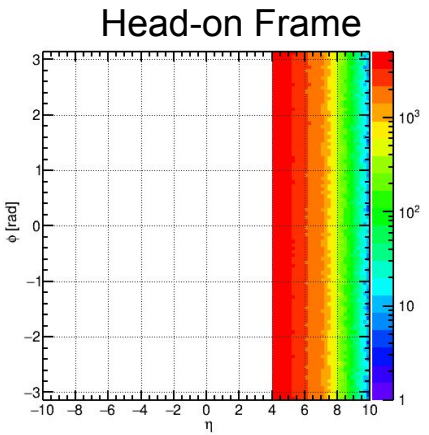
- **Top panel:** Mid-rapidity ($|\eta| < 1$)
- **Bottom panel:** Far-forward ($\eta > 4$)

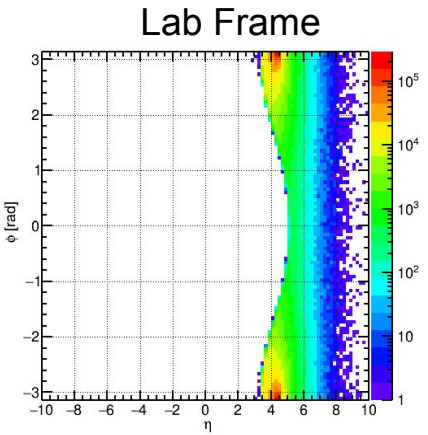
Particles are selected by pseudo-rapidity cut in Head-on frame (Left column). And then those particles in Lab frame is drawn (Right column).

In Mid-rapidity region, boost-rotation doesn't affect much, but in far-forward region, it does, to make particles direct to hadron beam line.









PYTHIA ep 18×275 GeV^2 Sample

Look into momentum components for different particles

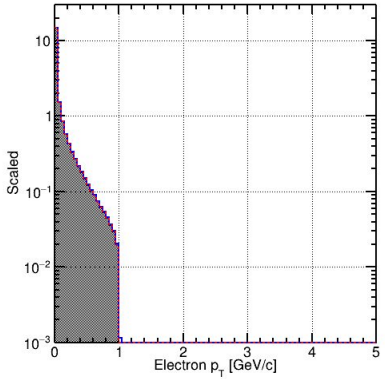
- Electron (top): mostly go to backward
- Proton (bottom): mostly go to forward

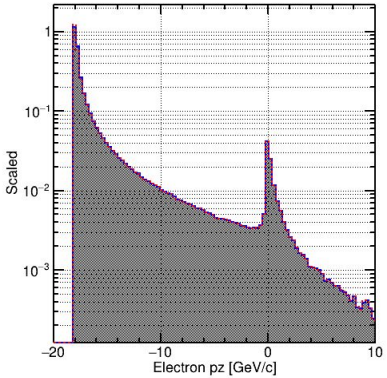
Here, all final-state electrons and all final-state protons are shown in p_T and p_z histograms.

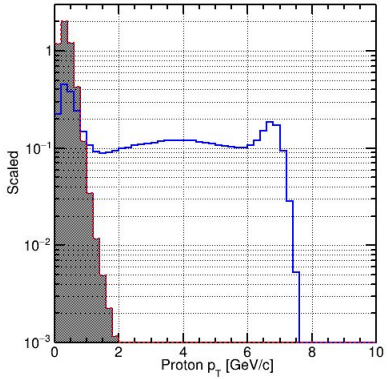
Color scheme: Nominal, Afterburned, and Afterburner effects removed

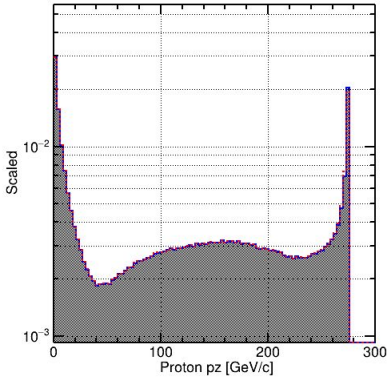
Impact of crossing angle (θ_{CA}) in p_T

- 275 GeV proton expects ~ 6.8 GeV ($p_T = P_z * \sin(\theta_{CA})$)



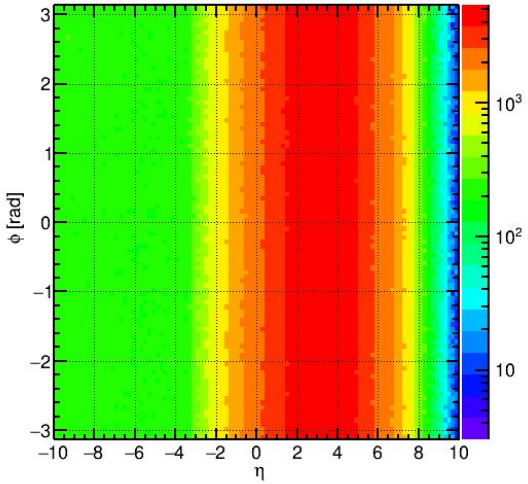




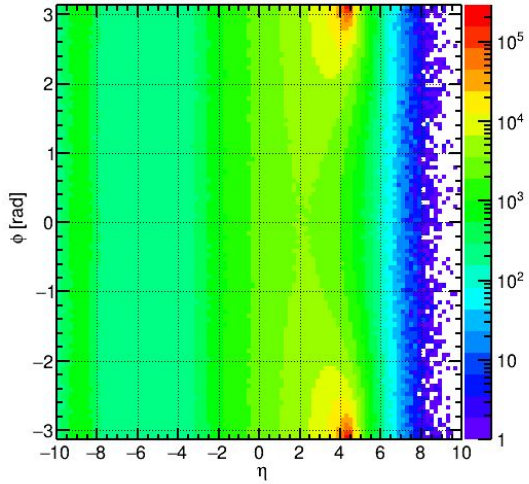


PYTHIA ep 18×275 GeV^2 Sample

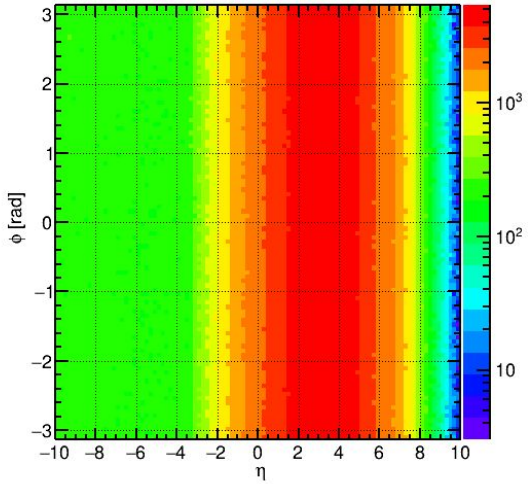
Nominal from MC Generator



Afterburned



Afterburner removed



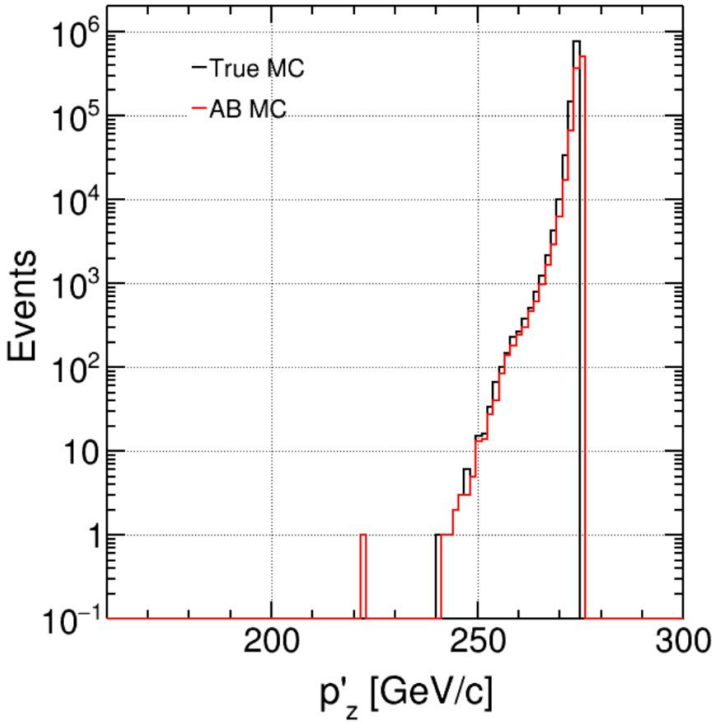
All final-state particles are shown in eta-phi plane.

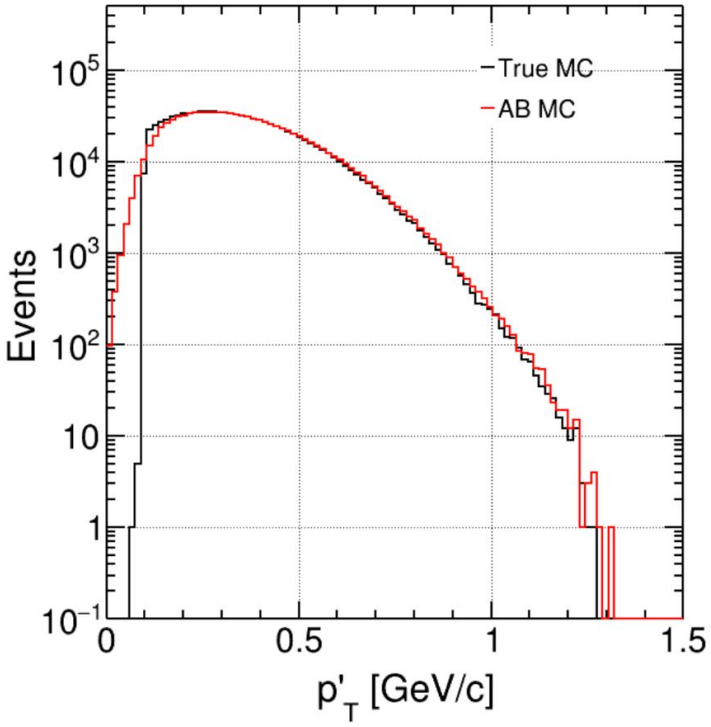
Left figure: Nominal distribution from MC event generator

Middle figure: Afterburned sample introduces a hot spot at η ~ 4.3 (crossing angle = 25 mrad) where IP-6 hadron beam is aligned

Right figure: Take Afterburned sample (middle) and apply reverse transformation and resulting is in a good agreement with nominal

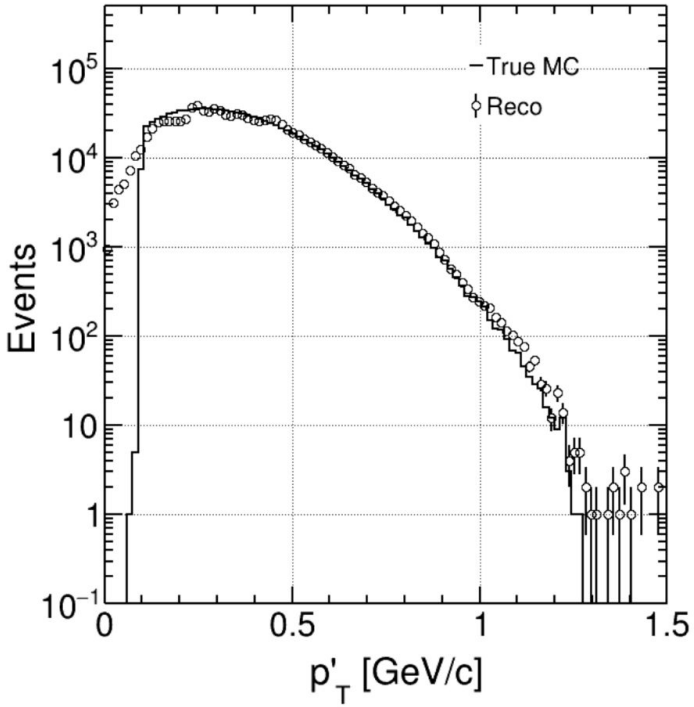
DVCS 18x275 Sample with IP8 - protons

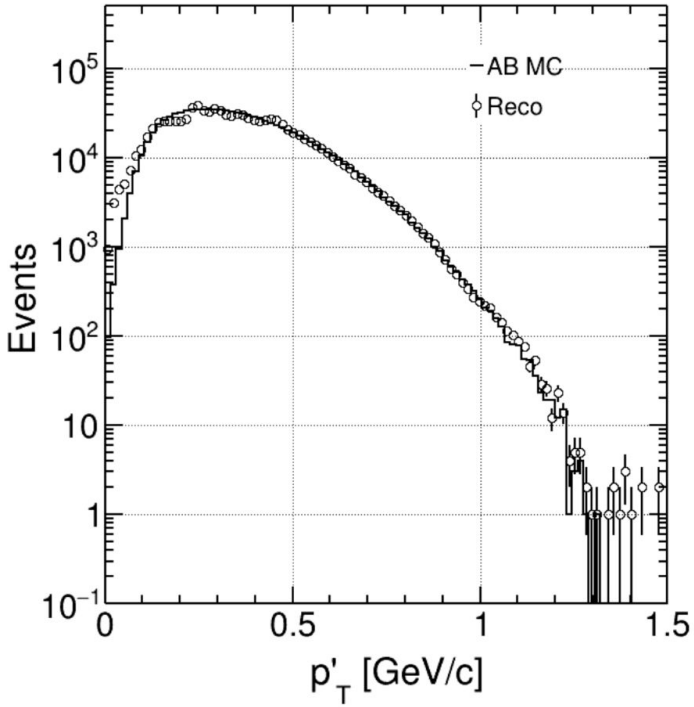




- MCTruth vs. afterburned MC, with crossing angle removed

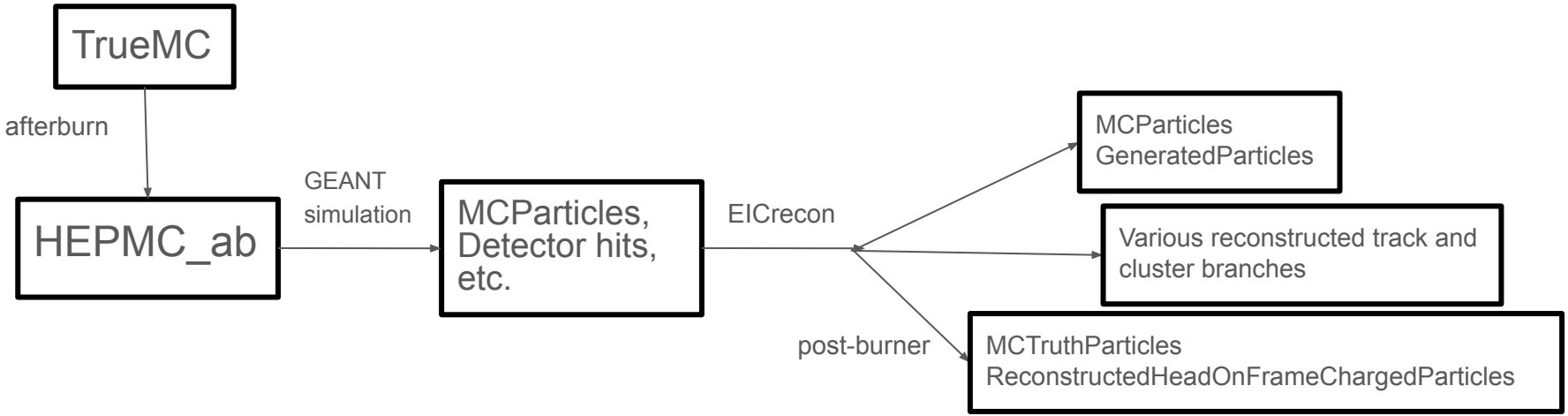
DVCS 18x275 Sample with IP8 - protons





- MCTruth vs. full reconstructed (left) and afterburned MC vs. full reconstructed (right), with crossing angle removed.

How an analysis should work for a user



- Right now, EICrecon by default spits out “afterburned” MC information, which has all beam effects and the crossing angle, and spits out reconstructed particles in the crossing angle frame.
  - **The post-burner will remove the full set of “afterburned” modifications from MCParticles and store it in a new collection.**
  - **This gives you back the TrueMC information your generator produced.**
- It will also remove *only the crossing angle* from the ReconstructedChargedParticle (and others) branch(s) so the head-on frame is consistently used.

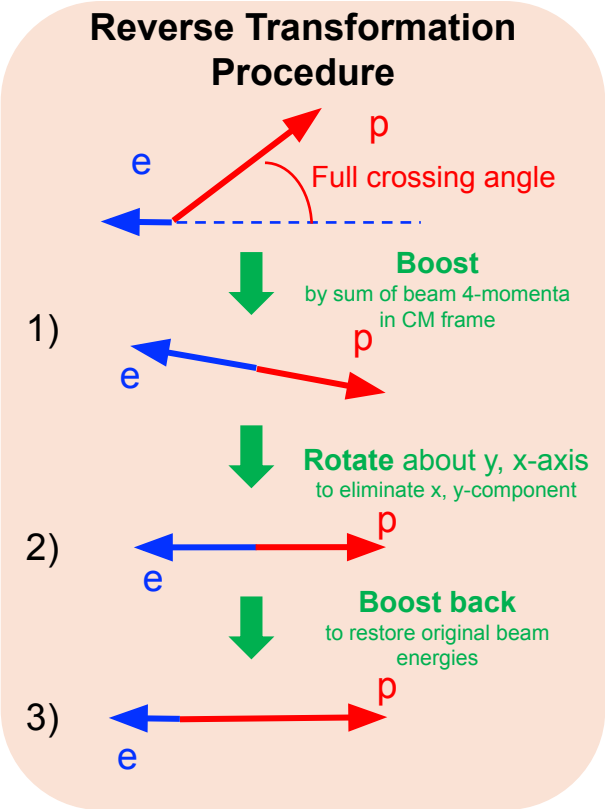
Summary

- Final physics analyses will require proper handling of the crossing angle and beam effects.
  - Comparisons between output from npsim/EICrecon and MCTruth require these effects are properly removed from MCParticles and stored in a new branch to ensure resolutions for observables are accurate.
- A preliminary solution is already in place in EICrecon and we are testing it now.
  - We do not propose removing any information from EICrecon at this time - simply adding new output branches which contains the “post-burned” information.
  - We will need some technical help on outputting the result of the algorithm into the relevant data structure (e.g. “MCParticleCollection” vs “ReconstructedParticleCollection”).
- More discussion to come with software and physics working groups.
  - Especially on technical implementation, and user awareness (making sure users understand what the quantities mean, where to find detailed explanations, etc.).

Now for some additional comments and things to consider…

How to Get Back to True MC from Afterburned

Boost and rotation is determined by beam particles of event (event-by-event)




```
// Defining beam particles
TLorentzVector e_beam(0.,0.,0.,0.);
TLorentzVector h_beam(0.,0.,0.,0.);
for(const auto& v : evt.vertices())
{
  for(const auto& p : v->particles_in())
  {
    TLorentzVector mc(p->momentum().px(), p->momentum().py(), p->momentum().pz(), p->momentum().e());
    if(p->pid() == 11 && p->status() == 4)
      e_beam.SetPxPyPzE(mc.Px(), mc.Py(), mc.Pz(), mc.E());
    if((p->pid() == 2112 || p->pid() == 2212) && p->status() == 4)
      h_beam.SetPxPyPzE(mc.Px(), mc.Py(), mc.Pz(), mc.E());
  }
}

// Defining boost, rotation, and boost-back
// Boost
TLorentzVector cmBoost = e_beam + h_beam;
TLorentzVector boost(-cmBoost[0], -cmBoost[1], -cmBoost[2], cmBoost[3]);
TVector3 b(0., 0., 0.);
b = boost.BoostVector();

e_beam.Boost(b);
h_beam.Boost(b);

// Rotate in Y and X
double rotAboutY = -1.0*TMath::ATan2(h_beam.Px(), h_beam.Pz());
double rotAboutX =  1.0*TMath::ATan2(h_beam.Py(), h_beam.Pz());

e_beam.RotateY(rotAboutY);
h_beam.RotateY(rotAboutY);
e_beam.RotateX(rotAboutX);
h_beam.RotateX(rotAboutX);

// Boost-back
TLorentzVector boostBack(0., 0., cmBoost[2], cmBoost[3]);
TVector3 bb(0., 0., 0.);
bb = boostBack.BoostVector();

e_beam.Boost(bb);
h_beam.Boost(bb);
```

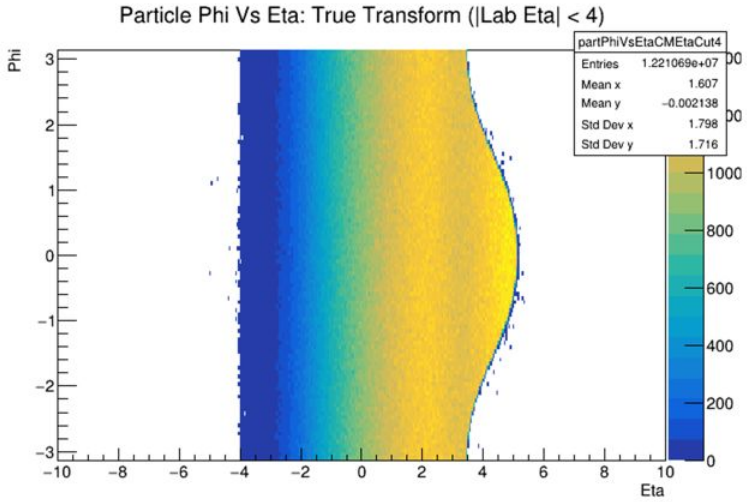
How to Get Back to True MC from Afterburned

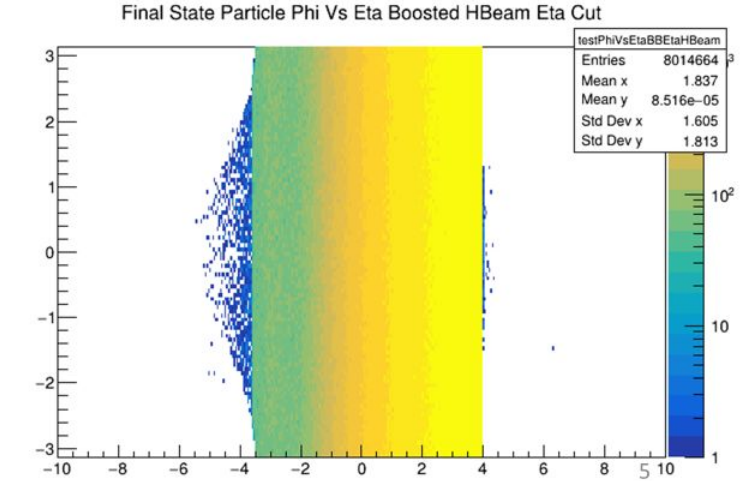
```
// Accessing final-state particles
for(const auto& v : evt.vertices())
{
  for(const auto& p : v->particles_out())
  {
    TLorentzVector mc(p->momentum().px(), p->momentum().py(), p->momentum().pz(), p->momentum().e());
    mc.Boost(b);
    mc.RotateY(rotAboutY);
    mc.RotateX(rotAboutX);
    mc.Boost(bb);

    // After this, all particles are back to true MC level

  }
}
```

- **Two approaches**
  - **True MC events** for true MC to full-reco comparison
    - Remove ALL effects to get back generated MC information.
  - **Realistic events** remove only crossing angle, keep beam FX in events
    - This is an important distinction!! Crossing angle can be accounted-for in real events, but beam effects such as angular divergence are random and cannot be removed from a real event!

How to Define Acceptance Cuts

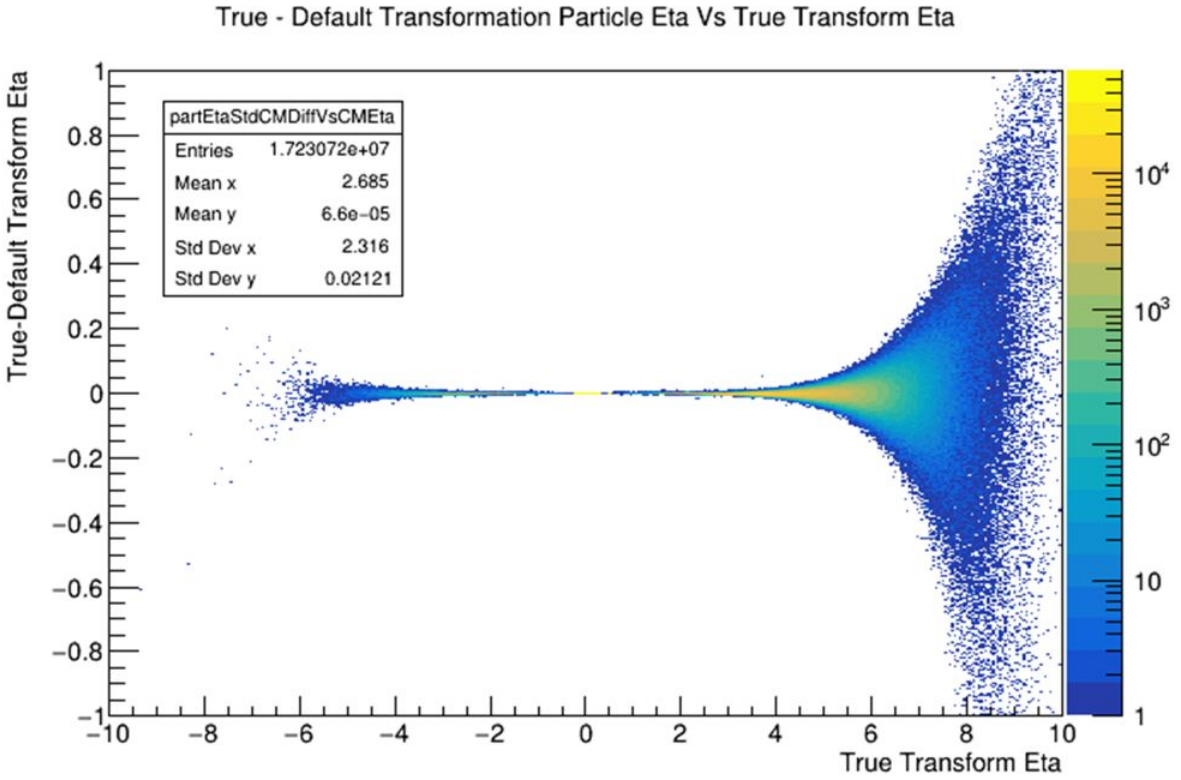
- Because it has a single meaningful axis, the head-on frame is still likely the best frame for defining our physics quantities
- However, will our detector acceptance (the limit of HCal or tracking coverage) be symmetric around the electron beam or the hadron beam? If it is symmetric about the hadron beam, we should use eta as defined with respect to the hadron beam to place our cuts
- Both plots show the phi vs eta distribution where these quantities are defined in the head-on frame
  - Top plot applies a cut for |eta| < 4 where eta is defined relative to the electron beam
  - Bottom plot applies a cut for |eta| < 4 where eta is defined relative to the hadron beam
  - At negative eta, can define acceptance relative to the electron beam – this will eliminate distortions in electron going direction



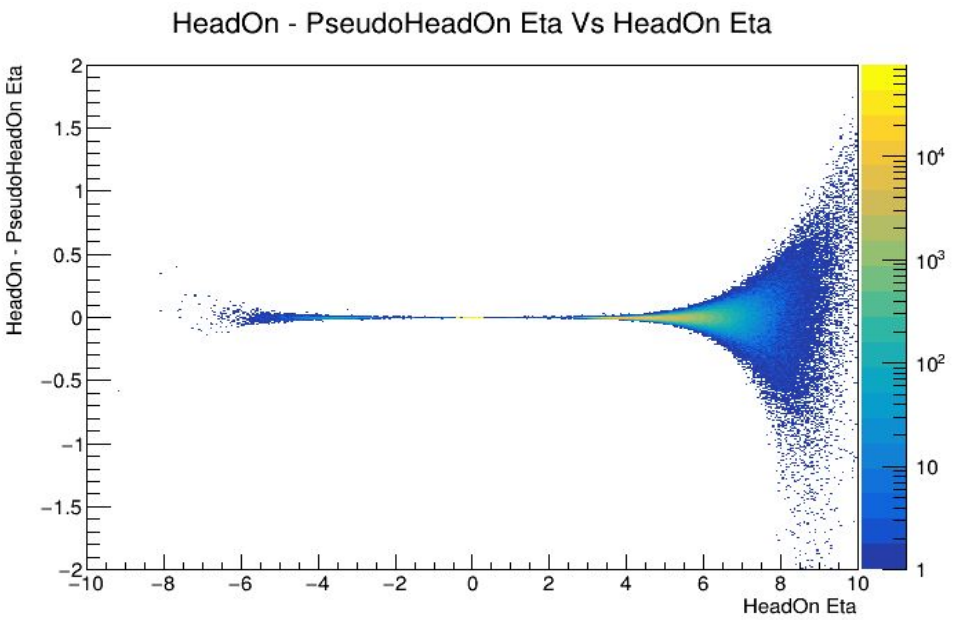


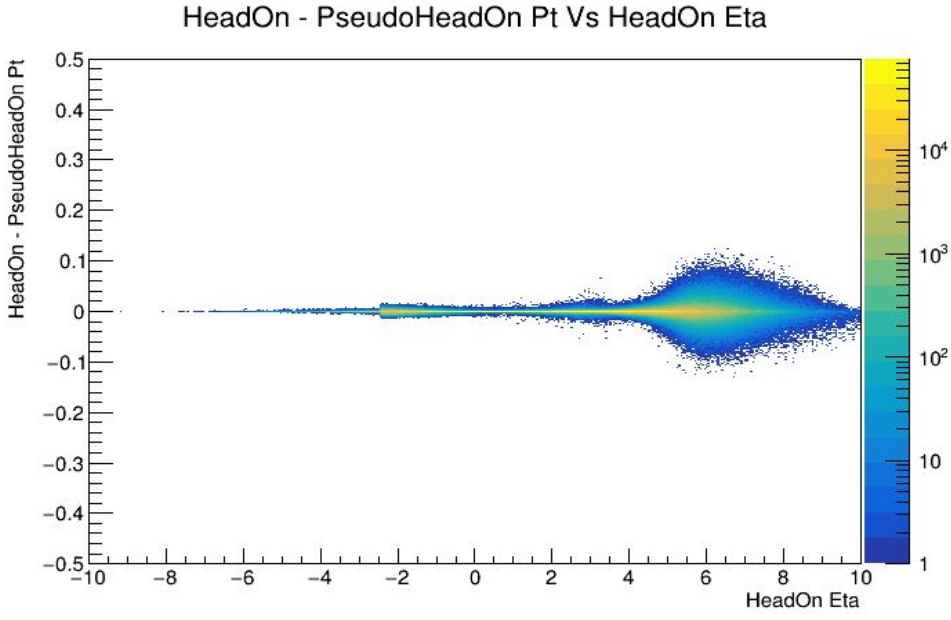
Beam smearing effects in head-on frame

- All plots above used transformation constructed assuming perfect knowledge of incoming beams
- Divergence, beam energy spread, etc introduce momentum deviations which cannot be measured event by event
- Compare true transformation (assuming perfect knowledge) with default transformation (assume nominal beam energies and crossing angle)
- Deviations are small at mid-rapidity but grow sizable as you move forward



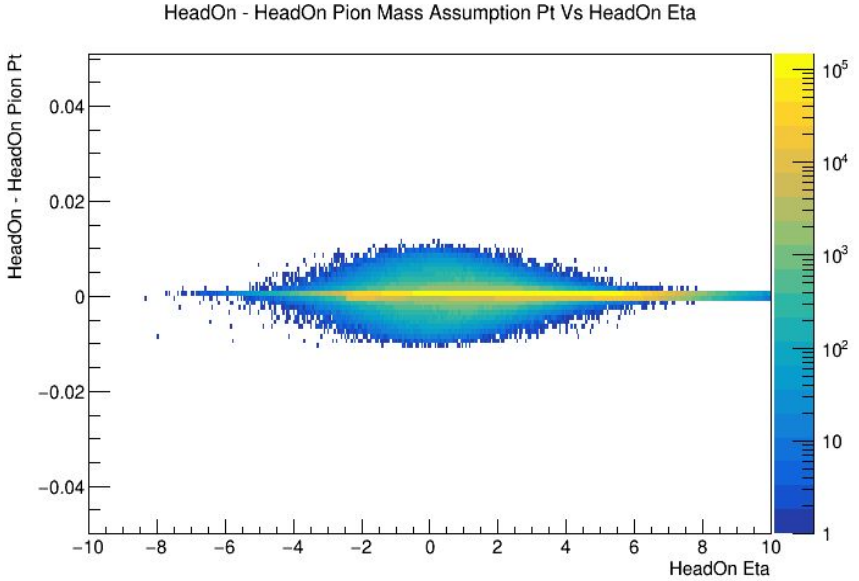
Comparison of head-on and crossing angle frame (no beam FX)





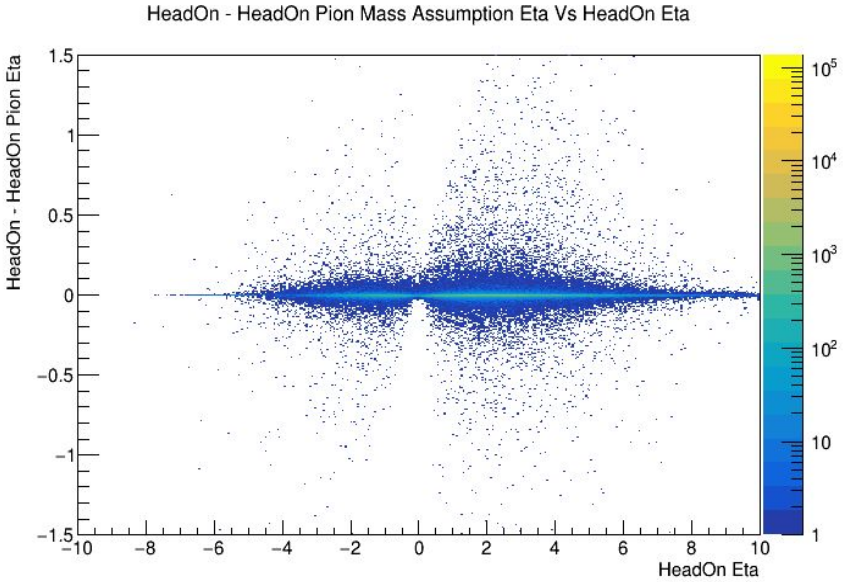
- Both plots show the difference between the head-on and pseudo head-on (crossing angle) frame for eta (left) and pT (right), both as a function of eta.
- The effect is largest in the forward direction.

Head-On Frame: Pion Mass Assumption for boost

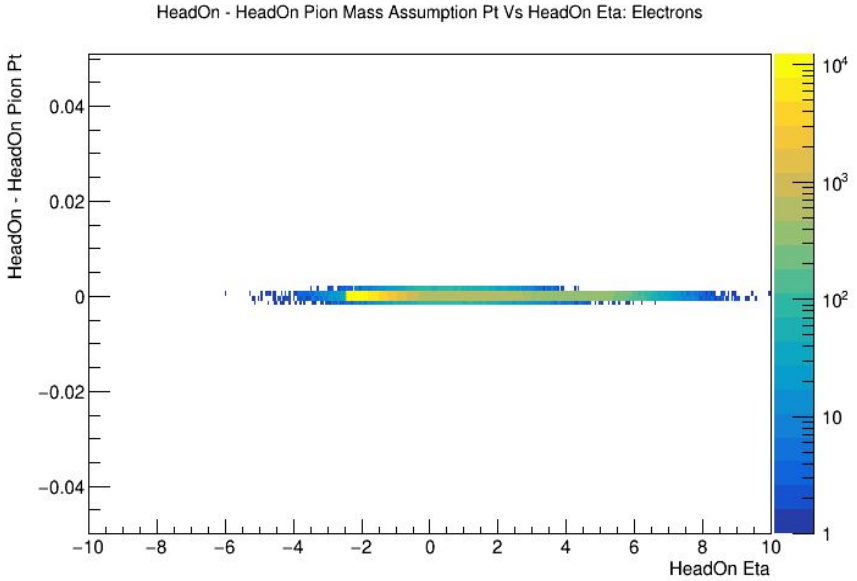


- Find ‘maximal’ deviation in boost to head-on frame from mis-PID
- Assume all tracks are pions and all neutrals are massless and and calc difference between true boost and boost with these mass assumptions

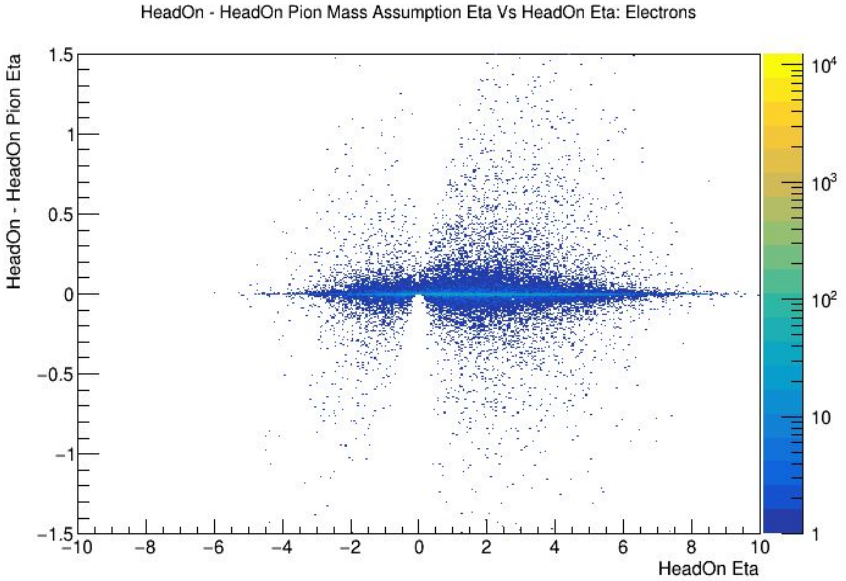
- The boost to the (pseudo)head-on frame operates on a 4-vector - particle energy (mass) needed
- What about tracks for which we don’t have PID?



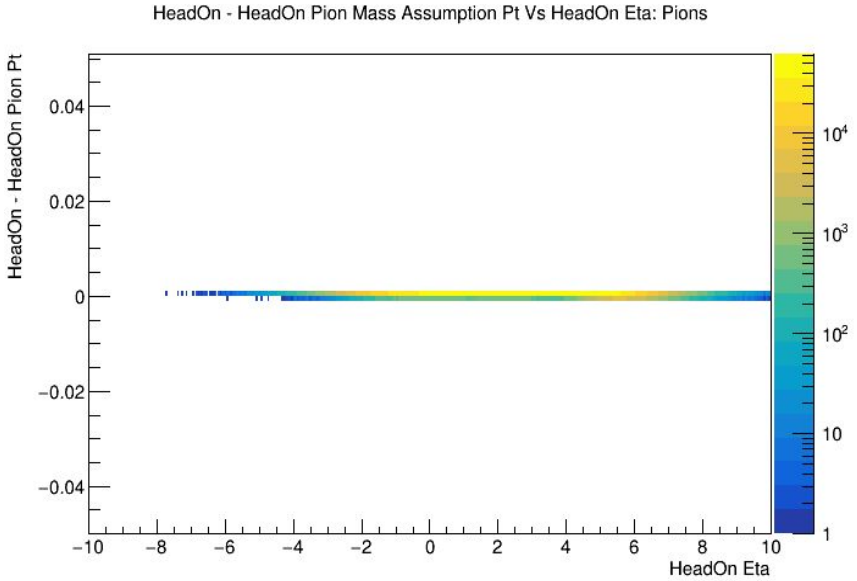
Head-On Frame: Pion Mass Assumption (Electrons)



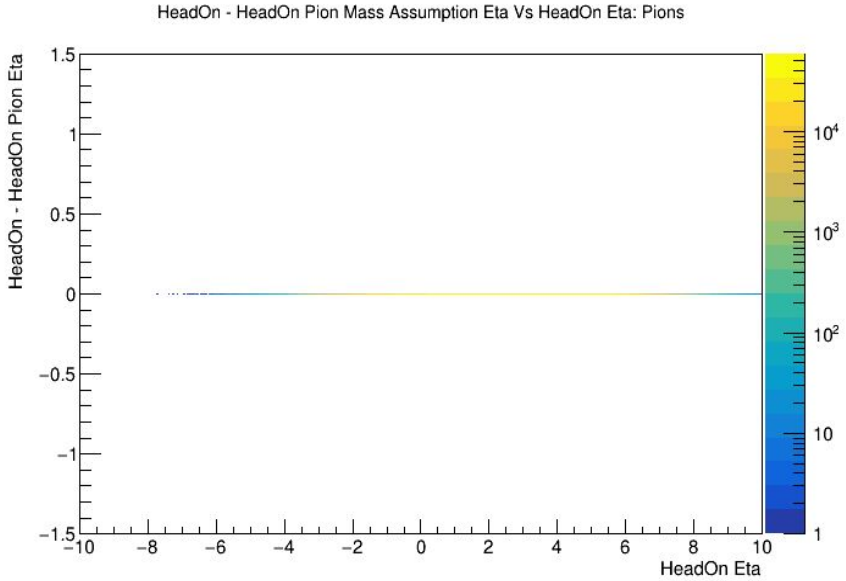
- Look at contributions from individual particles
- Electrons show largest deviation in eta if assumed to have pion mass



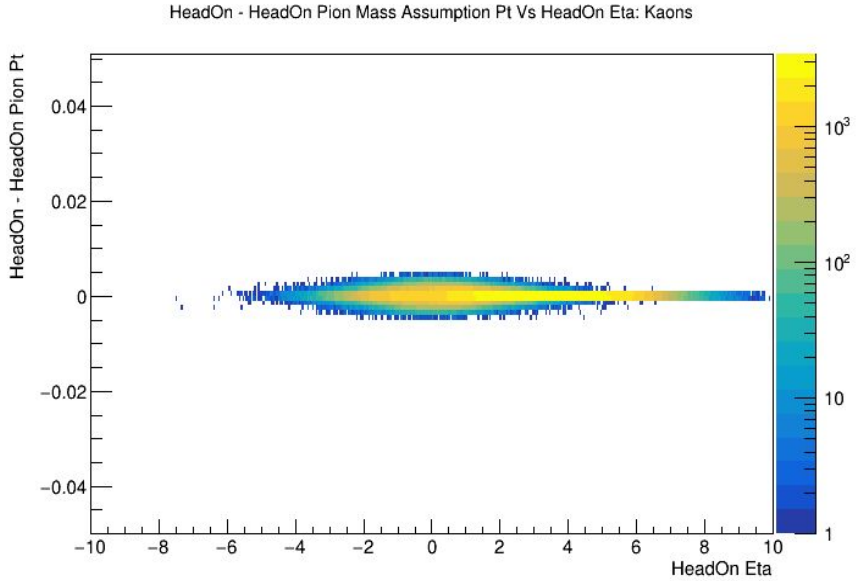
Head-On Frame: Pion Mass Assumption (Pions)

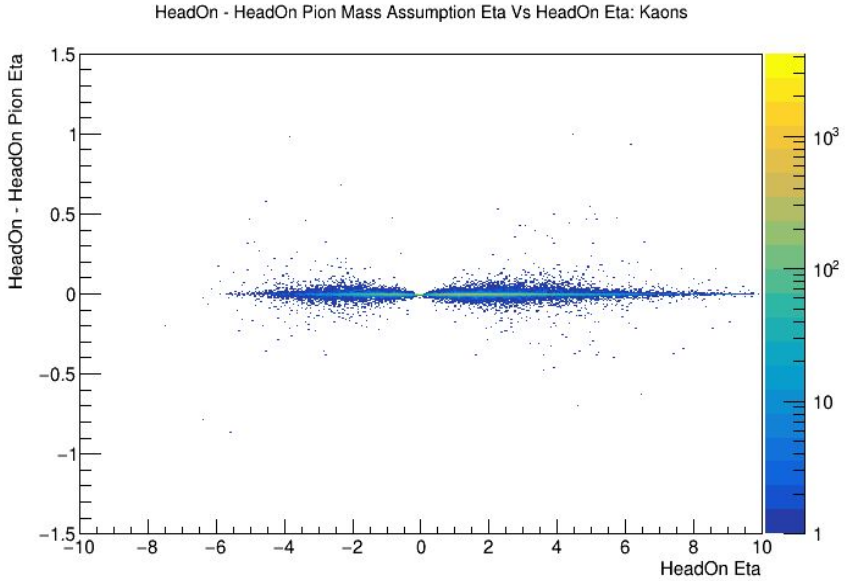


- Pions should exhibit no deviation as they are by definition using the correct mass

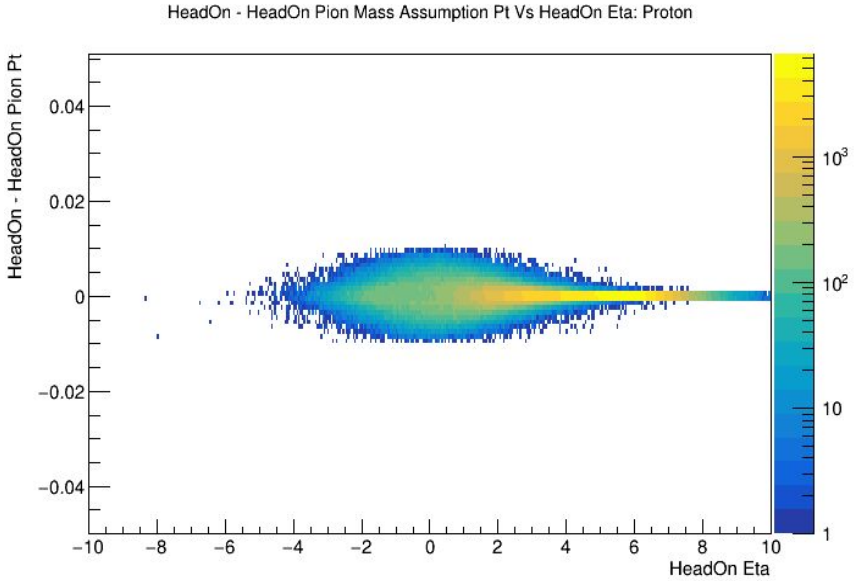


Head-On Frame: Pion Mass Assumption (Kaons)

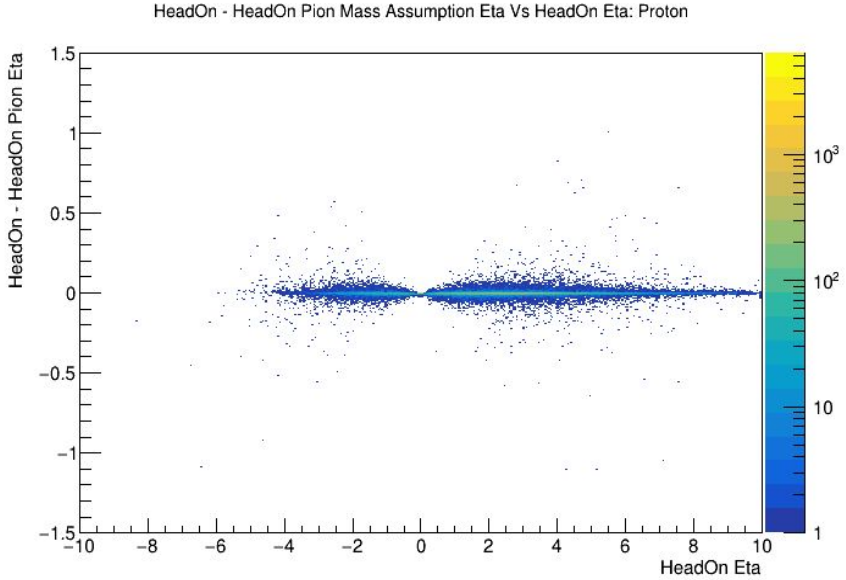




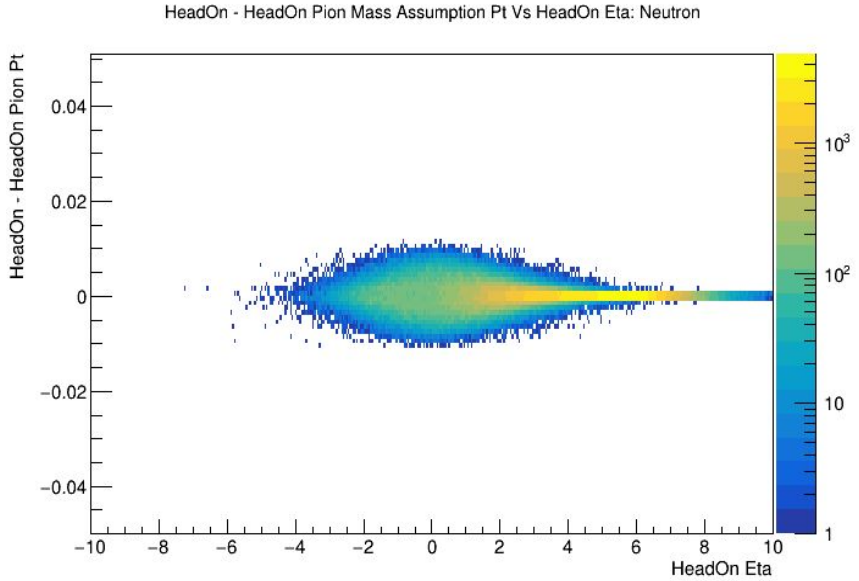
Head-On Frame: Pion Mass Assumption (Protons)

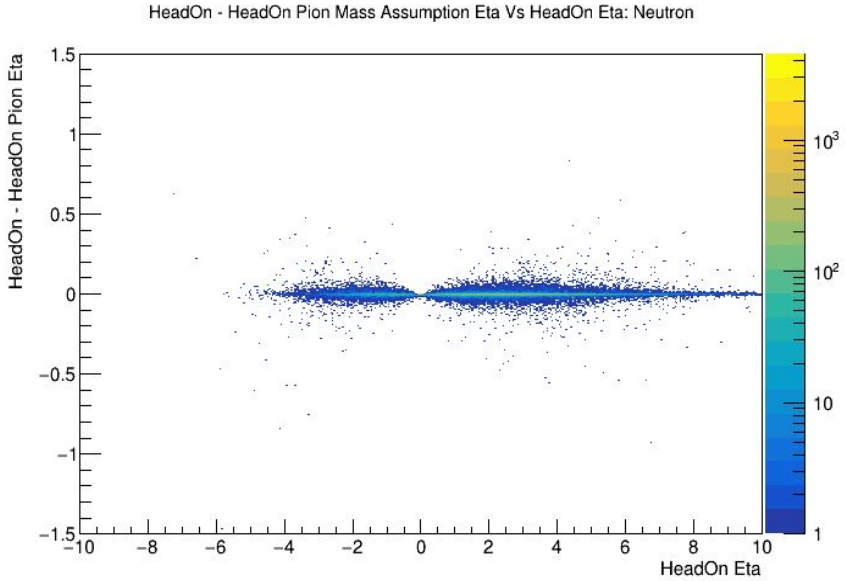


- Protons (and neutrons) show maximal deviation in transverse momentum between true head-on boost and head-on boost assuming pion mass



Head-On Frame: Pion Mass Assumption (Neutrons)





Head-On Frame: Pion Mass Assumption (KLongs)