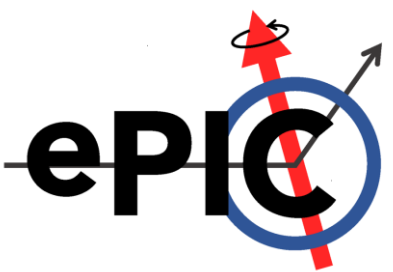
Physics Benchmarks for Low-Q2

Derek Glazier, Simon Gardner
University of Glasgow



Intro - Types of LowQ2 Benchmarks

1) Detector Benchmarks https://github.com/eic/detector_benchmarks

   A) Beamline Validation Benchmarks
      - use Geant4 particle gun
   B) Detector Response and Reconstruction Benchmarks
      - mixed quasi-real photoproduction electrons
         + Bremsstrahlung + backgrounds.
         * Not presented here

2) Physics Benchmarks : https://github.com/eic/physics_benchmarks

   A) Small sample to validate pull request
      - use 1000 ρ production events (elSpectro)

   B) Simulation campaign data
      - use EPIC/RECO/*/q2_0to1/pythia_ep_noradcor_*

Detector Benchmarks

Far Backward Steering and Acceptance

Beamline Benchmarks: [benchmarks/beamline]

Validate Geometry updates in CI to ensure changes are understood.

Run locally:

In top level benchmark dir with sourced ePIC geometry by calling:

snakemake –cores 4 results/beamline/test/

Runs 2 simulations:

Beam Core

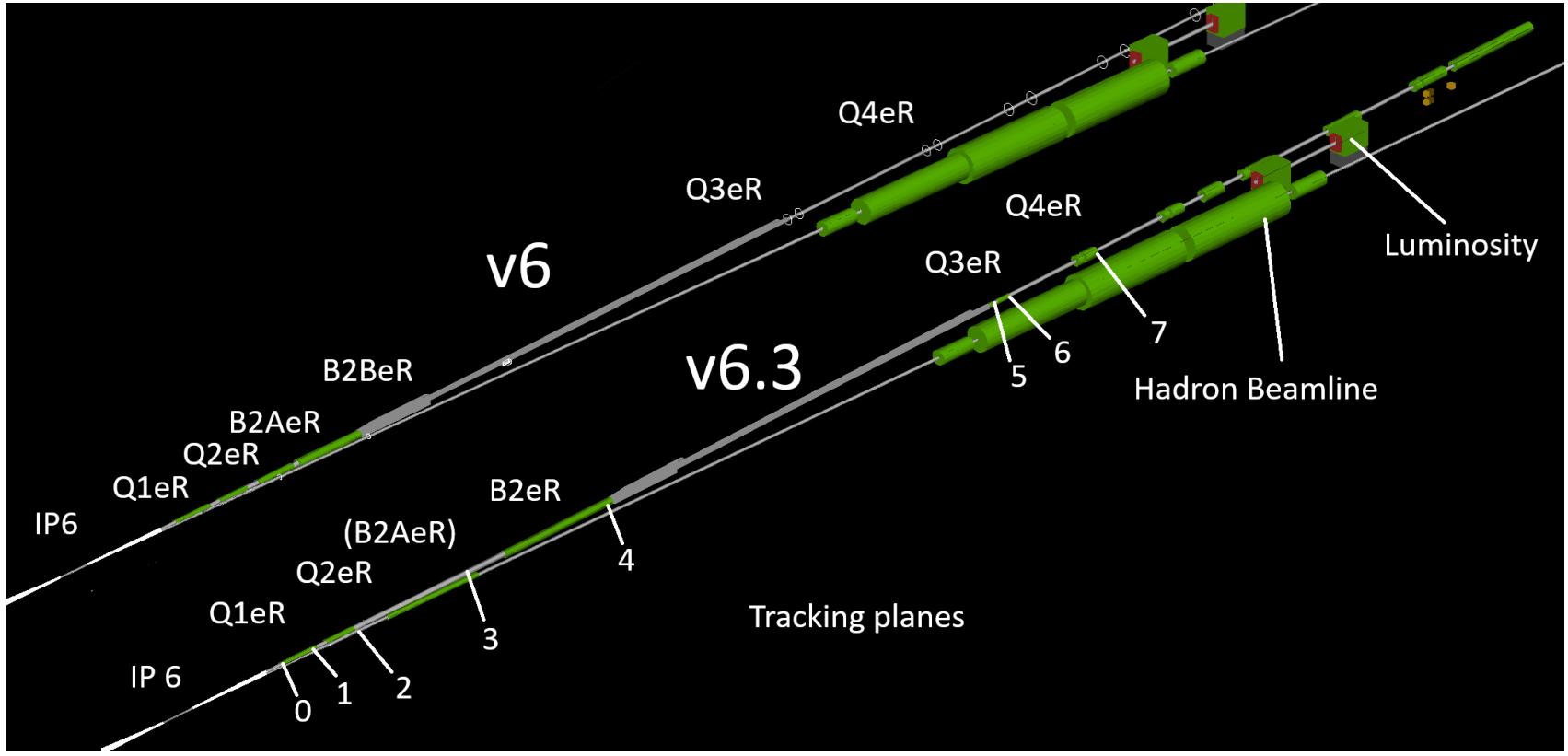
```
/gps/ang/type beam2d
/gps/ang/sigma_x 0.0002017 rad # 201.7 urad
/gps/ang/sigma_y 0.0001873 rad # 187.3 urad
/gps/pos/type Beam
/gps/pos/sigma_x 0.119 mm
/gps/pos/sigma_y 0.0107 mm
#/gps/energy 17.846 GeV
/gps/energy 18 GeV
/gps/particle e-
```

Far-Backward Acceptance

```
/gps/ang/type beam2d
/gps/ang/sigma_x 0.011 rad # 11 mrad
/gps/ang/sigma_y 0.011 rad # 11 mrad
/gps/pos/type Beam
/gps/pos/sigma_x 0.119 mm
/gps/pos/sigma_y 0.0107 mm
/gps/particle e-
/gps/ene/type Lin
/gps/ene/gradient 0.00001
/gps/ene/min 6 GeV
/gps/ene/max 18 GeV
```

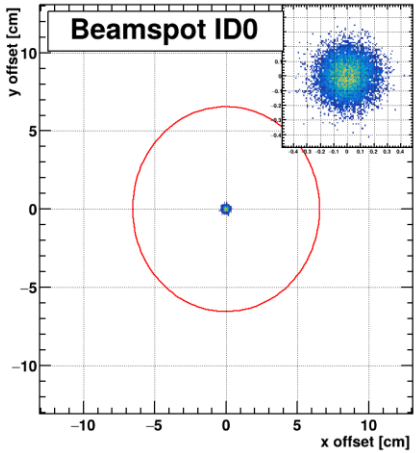
Far Backward Steering and Acceptance

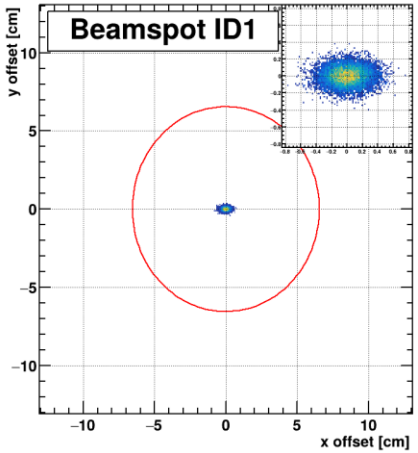
Update geometry to match project and all future changes
Moving B2eR out of cryostat
Virtual tracking detectors placed in beampipe.

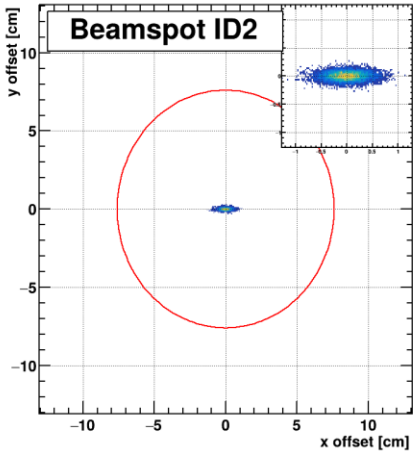


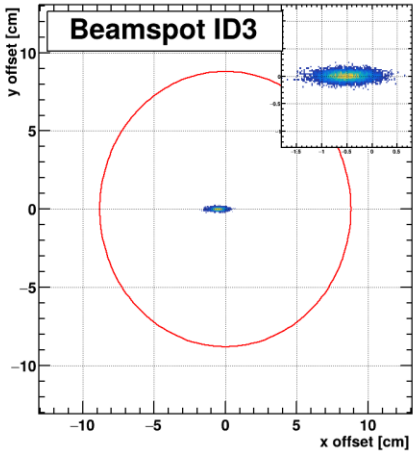
Far Backward Steering and Acceptance

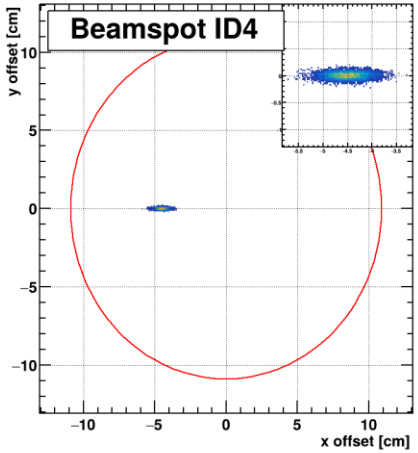
Current Beamline: Beamspot off center of Q3eR (ID5)

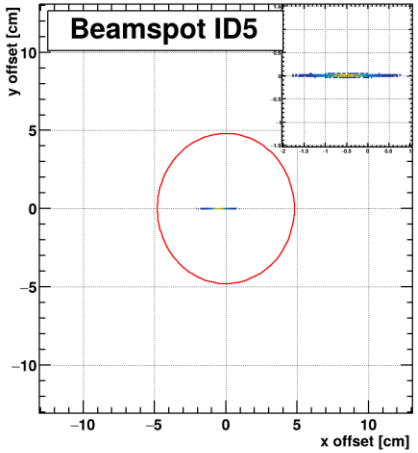


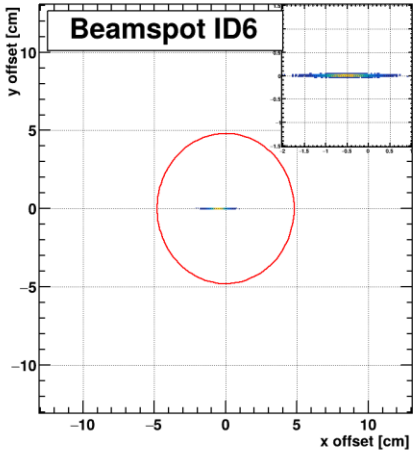


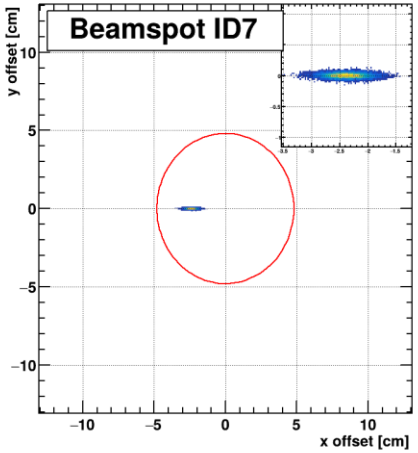






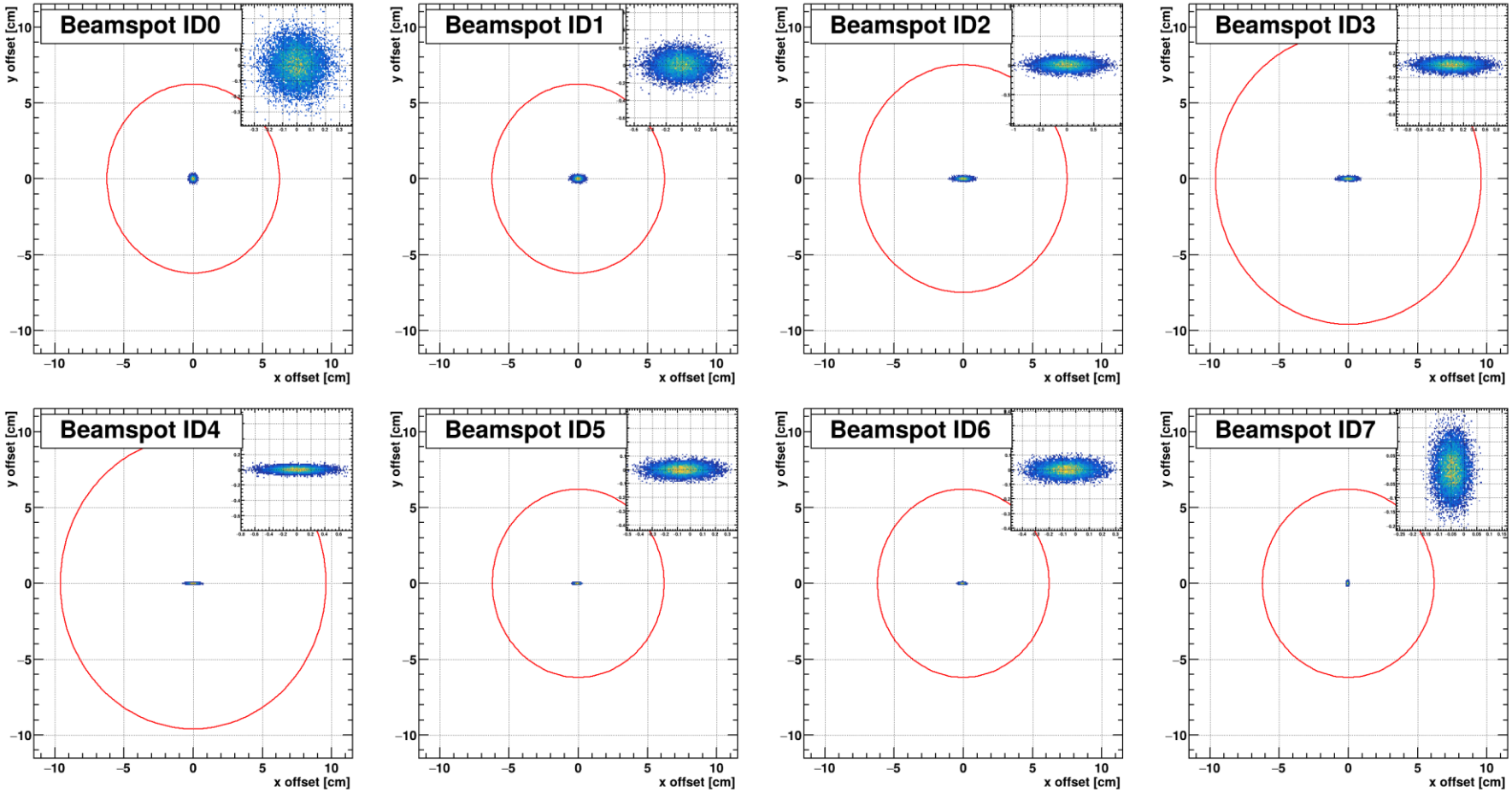






Far Backward Steering and Acceptance

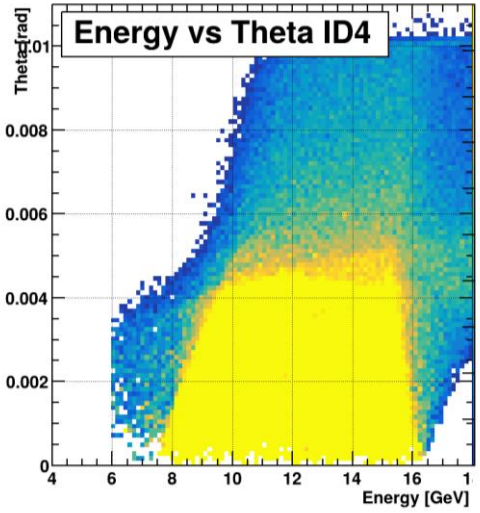
Updated Beamline: Nicely centered beam



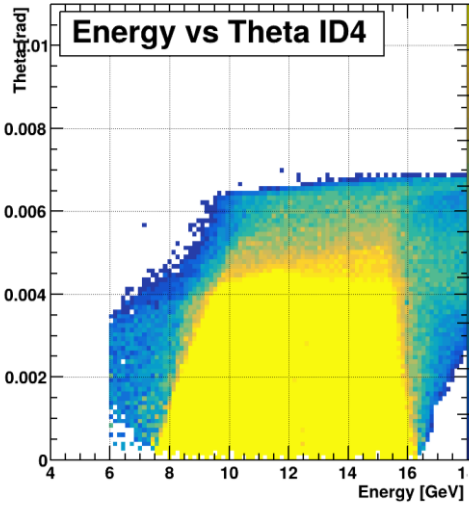
Far Backward Steering and Acceptance

	Radius	End	Opening
Original	47.60 mm	4560.17 mm	10.4 mrad
Current	31 mm	4544.49 mm	6.8 mrad

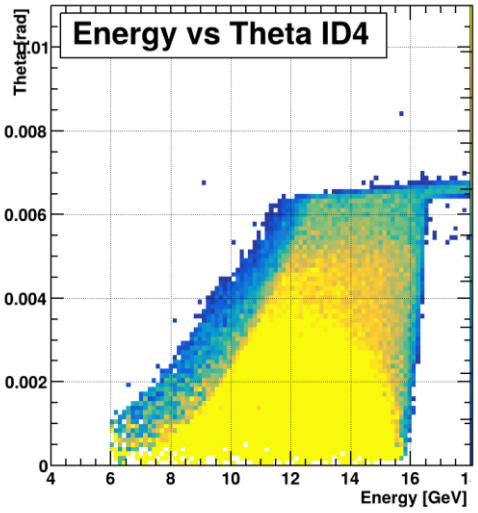
Without benchmark in place acceptance change from smaller central beampipe wasn't identified at the time



Old central beampipe



Current – cold B2eR



6.3 – warm B2eR

Real 6.3 – smaller diameter

Low-Q2 Detector Benchmark results

Flag Failure in Github CI by any step returning non 0 value

Steering fail on:

- Missing beamline electrons in any virtual plane
- Beam offset from center of magnets (not yet)

Acceptance failure on.

- Large fractional acceptance change in drift volume (not yet)

Artefacts from current benchmarks being run:

Beamline benchmark artefacts

Several improvements planned including better formatted plots, geometry pictures and explanation report.

Low-Q2 Electron Reconstruction

Check Neural Network reconstruction of Low-Q2 scattered electron for CI changes.

Reuse events from acceptance benchmark run.
Feed in position and momentum of virtual hits before drift volume.
Check reconstruction offsets and resolutions.

- If offset or resolution too high.
  - Run EICRecon on sample producing tagger tracks –
    (MC virtual tracker hits too perfect.)
  - Retrain network on position and momentum of tracks
- If pass after retraining
  - Create PR for epic-data
  - Create PR to branch triggering CI tests updating model

*Still under development.

Physics Benchmarks

Physics Benchmarks

Process kinematics - extract physics

Reaction in variant mass W
Invariant mass of meson particles
Invariant mass of baryon particles
Production t distribution
Centre-of-Mass production angles
Decay product angular distributions

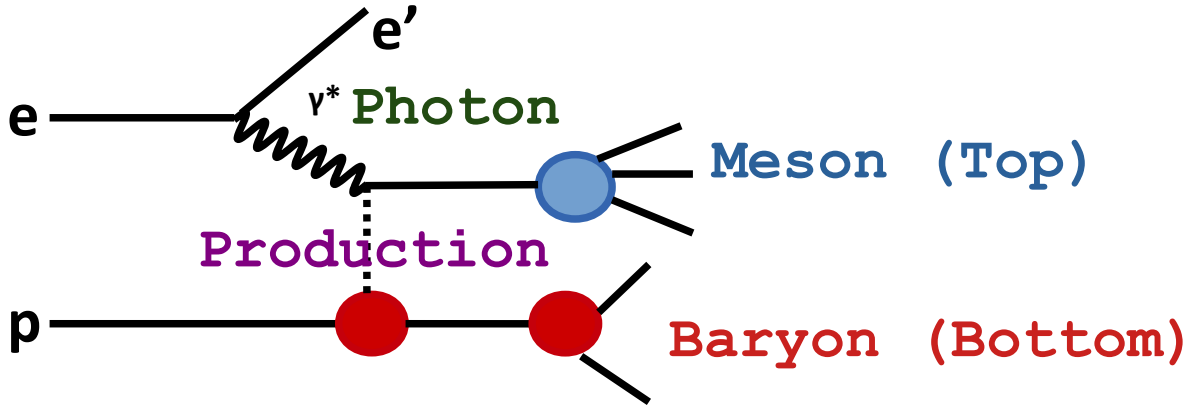
Exclusivity – reduce backgrounds

* Tagger + Central
=> detect tagger e- + meson decay particles
=> reconstruct missing baryon
=> missing mass, missing momentum
(as reactions generally small t, missing momentum ~= proton beam)

* Tagger + Central + FarForward
As above + proton
4 constraints from energy and momentum conservation
- can define many variables, but only 4 independent

RAD - Reaction Aware DataFrames

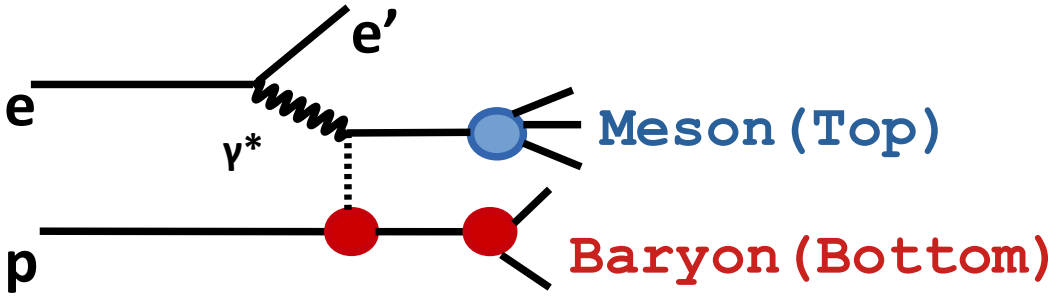
Extend RDataFrame to automate analysis of reactions
Only depends on ROOT and RAD
https://github.com/dglazier/epic-rad

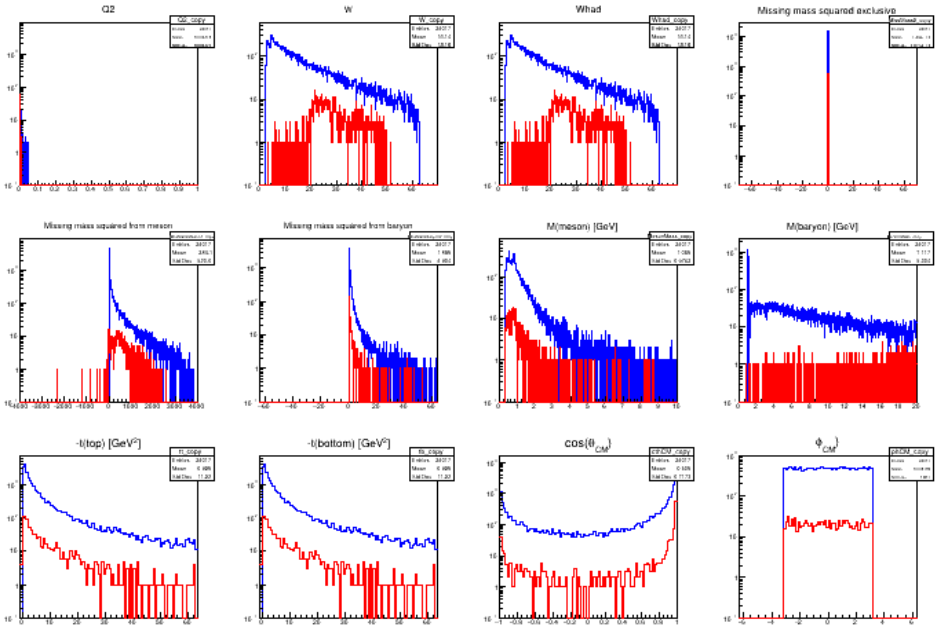


- Identify final state particles (Filter)
- Associate them with Top and Bottom vertices (Define)
- Calculate Photon Kinematics
- Calculate Production Kinematics
- Calculate Top/Bottom Intermediate states
- Calculate Top/Bottom Decay Kinematics

* RAD Core is experiment independent clas12-rad built on top of this too

Reaction Aware DataFrames






```
//////////////////////////
// Analyse e p->pi+ pi- p'
rad::config::ePICReaction epic{"tree","file"};

epic.setScatElectronIndex(useNthOccurance(1,11),{"tru_pid"});
epic.setParticleIndex("pip",useNthOccurance(1,211),{"tru_pid"});
epic.setParticleIndex("pim",useNthOccurance(1,-211),{"tru_pid"});
epic.setParticleIndex("prot",useNthOccurance(1,2212),{"tru_pid"});

epic.setBaryonParticles({"prot"});

epic.Particles().Sum("rho",{"pip","pim"});
epic.setMesonParticles({"pip","pim"});

//masses column name, {+ve particles}, {-ve particles}
rad::rdf::MissMass(epic,"W","{scat_ele}");
rad::rdf::MissMass(epic,"MissMass","{scat_ele,pip,pim,prot}");
rad::rdf::MissMass(epic,"MissMassRho","{scat_ele,pip,pim}");
rad::rdf::Mass(epic,"RhoMass","{pip,pim}");
rad::rdf::Mass(epic,"DppMass","{pip,prot}");
rad::rdf::Mass(epic,"D0Mass","{pim,prot}");

//t distribution, column name
rad::rdf::TTop(epic,"t_top");
rad::rdf::TBot(epic,"t_bot");
rad::rdf::TPrimeTop(epic,"tp_top");
rad::rdf::TPrimeBot(epic,"tp_bot");

//CM production angles
rad::rdf::CMAngles(epic,"CM");
rad::rdf::Q2(epic,"Q2");

//decay angles
rad::rdf::gn2s0s0s12::HelicityAngles(epic,"Heli");
```

Physics Benchmarks for pull request

[Physics Benchmarks - LowQ2 - Link]

Just use 1000 ρ production events (elSpectro)
Generic benchmarking framework based on RAD
Use truth matching of tracks – not testing PID, just best case
Can give pairs of meson and baryon PDGs (for different final states)
Currently just use recon. e-′ + π+π- and missing proton/baryon

```
ReactionBenchmarks excTaggHists{"ExcTagger",mesons[ipy],baryons[ipy]};
//Define each variable we wish to benchmark with histogram model
excTaggHists.AddVar("Q2",{"Q2","Q2",200,0,0.05},{200,-0.0005,0.0005});
excTaggHists.AddVar("W",{"W","W",200,0,1.1*WMax});
...
```

e.g. {221,-211},{""}

Define histogram models
And opt. resolution models
For each variable

ReactionBenchmarks will create :
basic distributions , truth/recon;
efficencies and resolutions for each Var

[Physics Benchmarks - LowQ2 pull request - Artifacts -Link]

Common Bench - test_against

EIC / benchmarks / **common_bench**

C **common_bench**

Develop new class from common_bench framework

Targets for current tests are taken from prior reference results file

Each variable added to ReactionBenchmark generates a resolution test

Results of Gaus fit used for current result

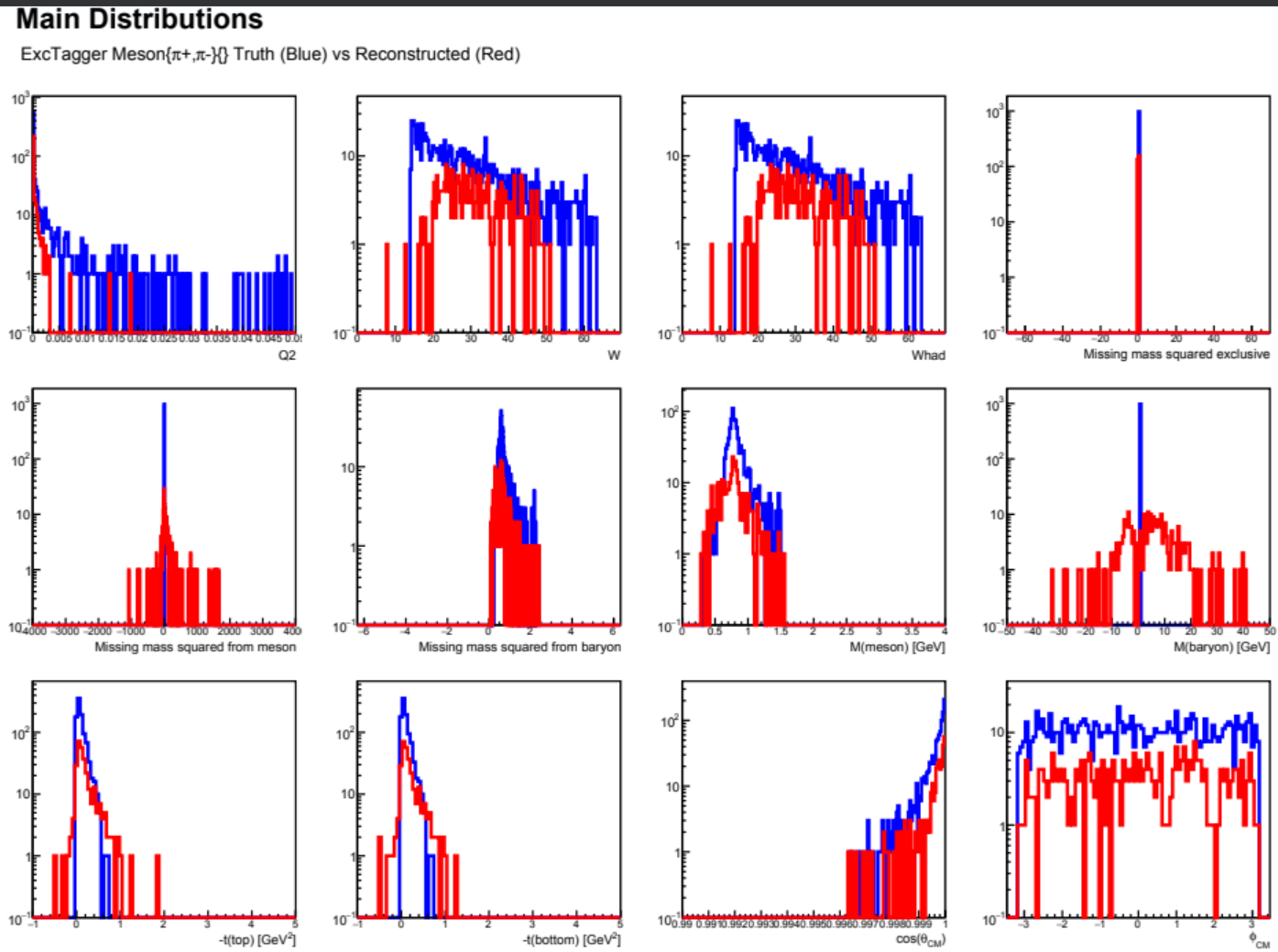
benchmark_against.h file Link

```
namespace common_bench {
  //want to create benchmark tests for each variable
  //either from scratch or using a previous test as target value

  class test_against {

  public:

  test_against(const std::string& oldfile,double tolerance=1.01 )
```

Example from CI job

```
Result of test Q2_res"pass" 1.29131e-05 < 1.87258e-05
Result of test W_res"fail" 0.749117 < 0.52525
Result of test Whad_res"fail" 0.749117 < 0.52525
Result of test MissMass2_res"fail" 1.74777e-13 < 1.7088e-13
Result of test MissMassMeson2_res"fail" 36.1481 < 28.4504
Result of test MissMassBaryon2_res"fail" 0.0119207 < 0.00583133
Result of test MesonMass_res"fail" 0.00690449 < 0.00435625
Result of test BaryonMass_res"pass" 8.35832 < 9.56259
Result of test tt_res"fail" 0.0902027 < 0.064122
Result of test tb_res"fail" 0.092183 < 0.0664826
```

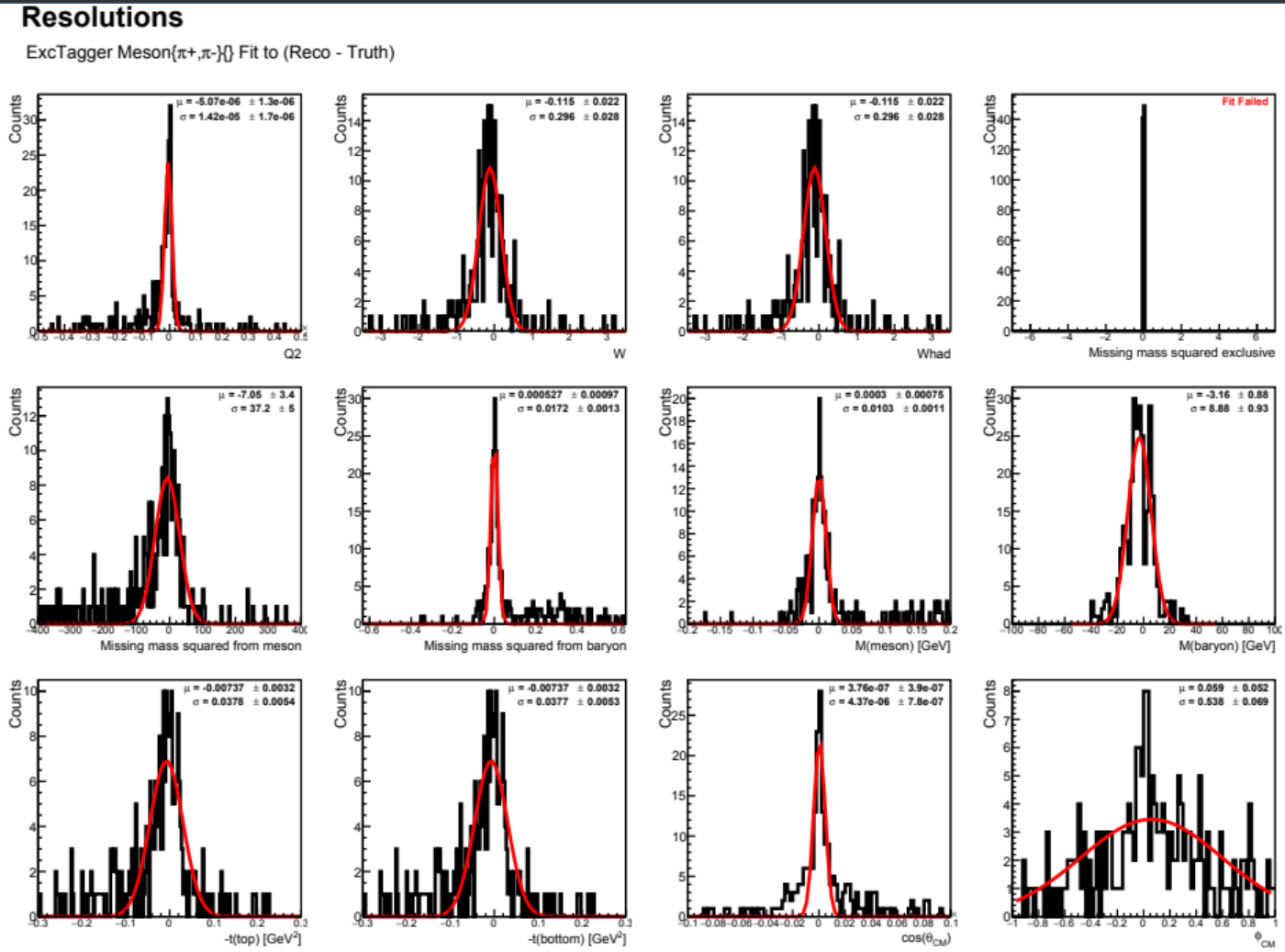
Benchmark Artifacts



Basic distributions :
Invariant masses
Production kinematics

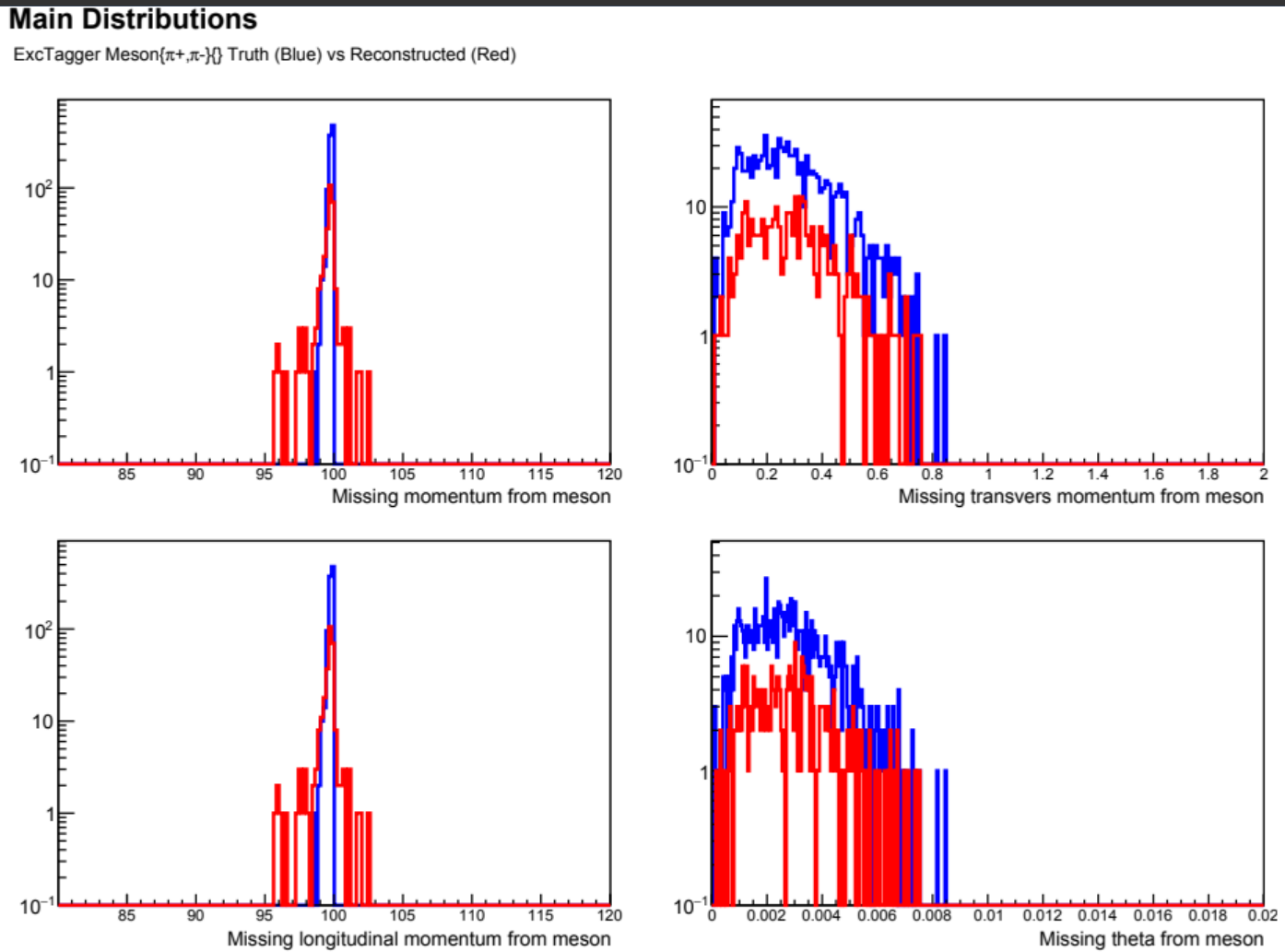
Truth
Recon

Benchmark Artifacts



Resolutions :
Invariant masses
Production kinematics

Benchmark Artifacts

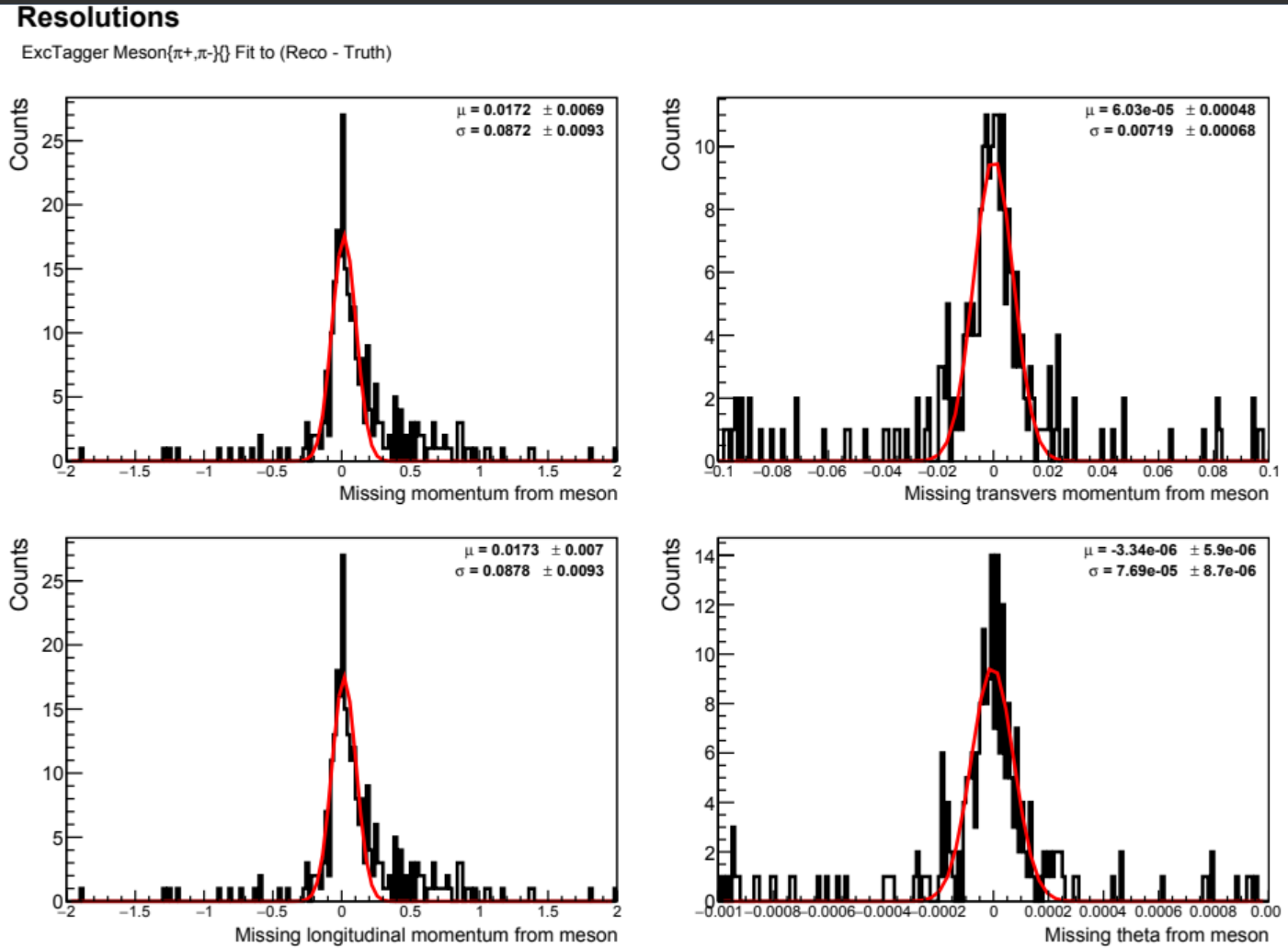


Basic distributions : Exclusivity

Truth

Recon

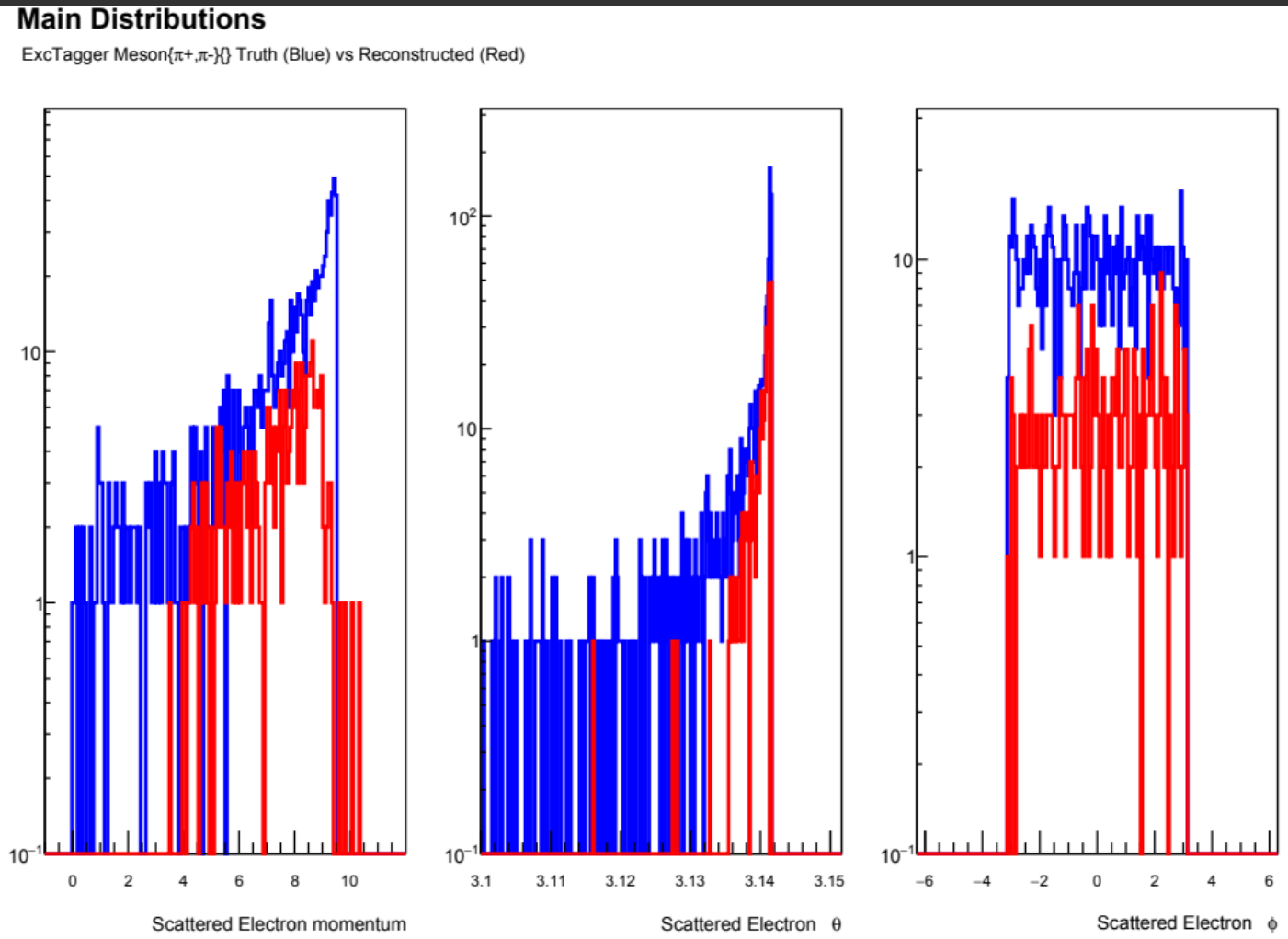
Benchmark Artifacts



Resolutions :

Exclusivity

Benchmark Artifacts

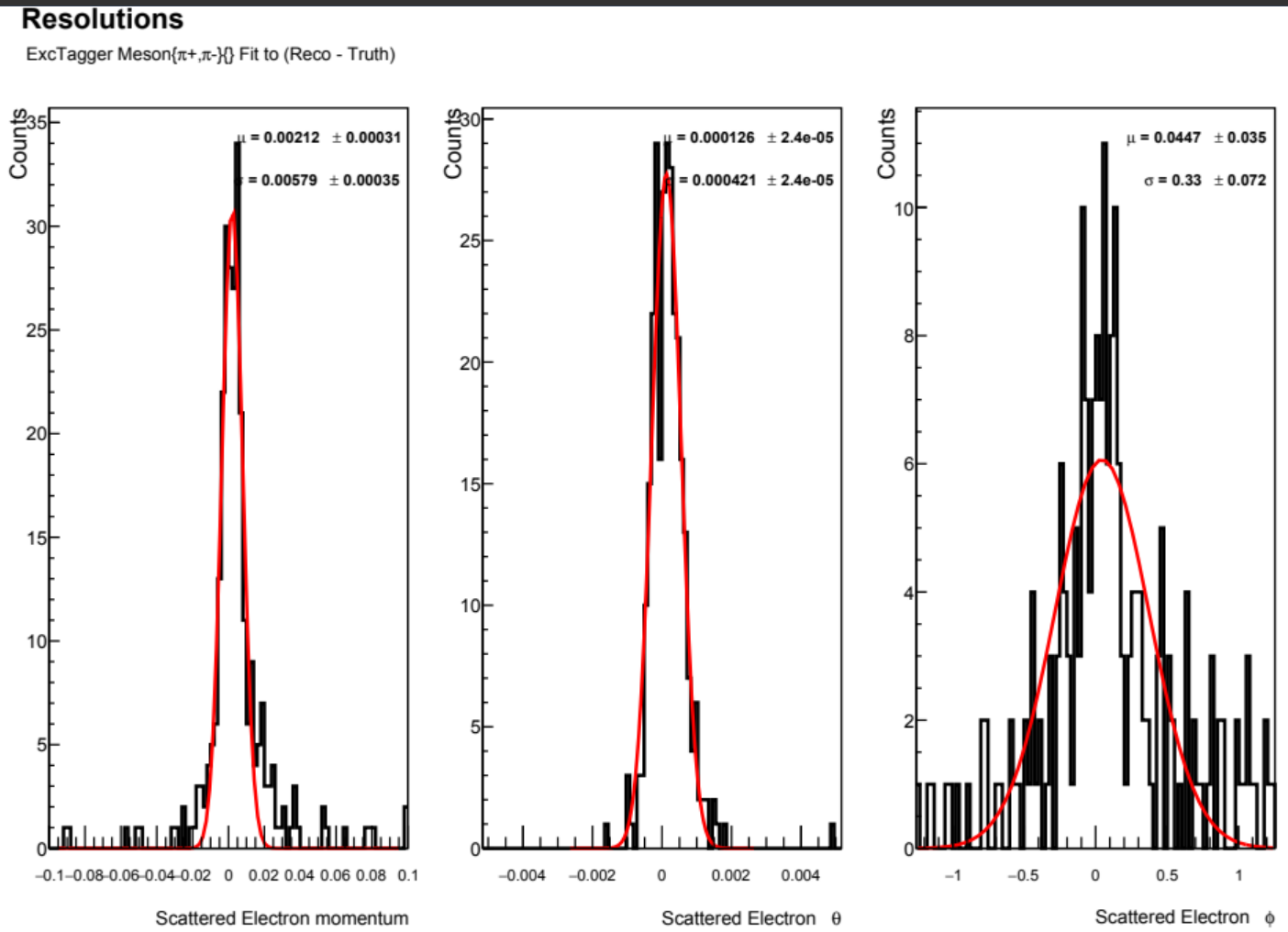


Basic distributions : e- momentum

Truth

Recon

Benchmark Artifacts

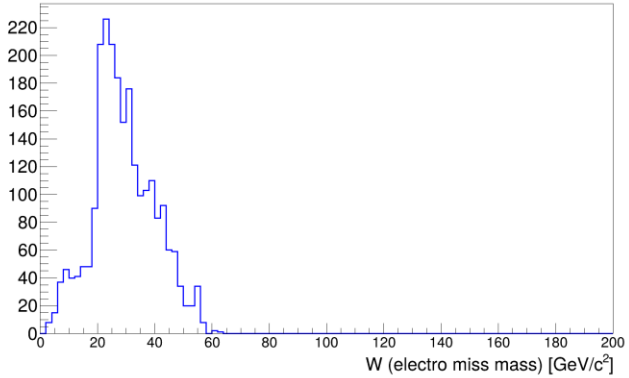


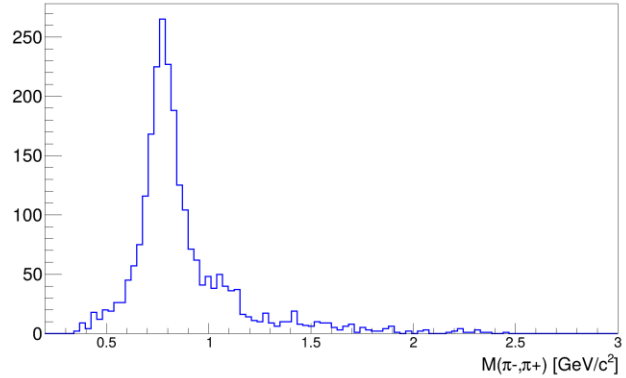
Resolutions :

Electron Momentum

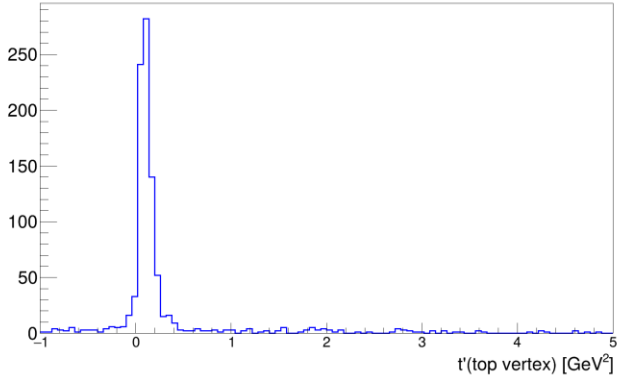
Physics Benchmarks for campaigns

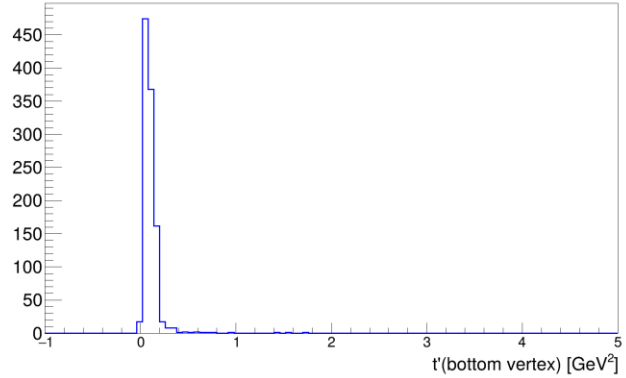
Same RAD and eic/physics_benchmarks framework applies
Run on lowQ2 Pythia data from simulation campaigns (25.05.0) 1/50 runs
Select e-′ + π+π- + p′ via
MCParticles – isolate exclusive and inclusive 2π reactions from Pythia





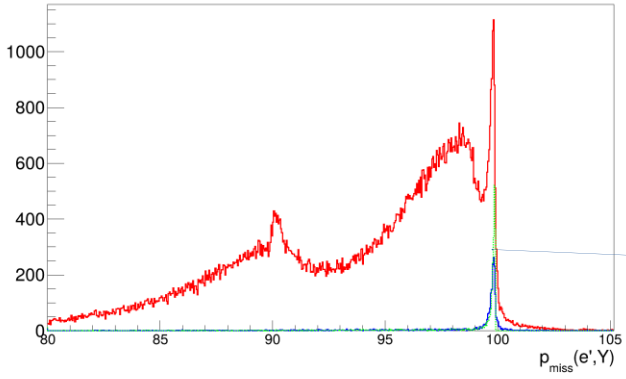
Basic distributions of reconstructed variables

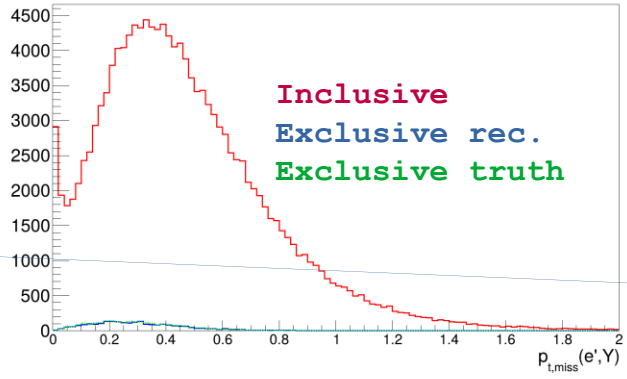


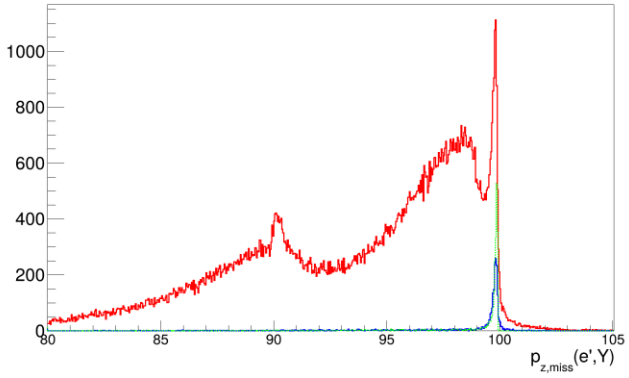


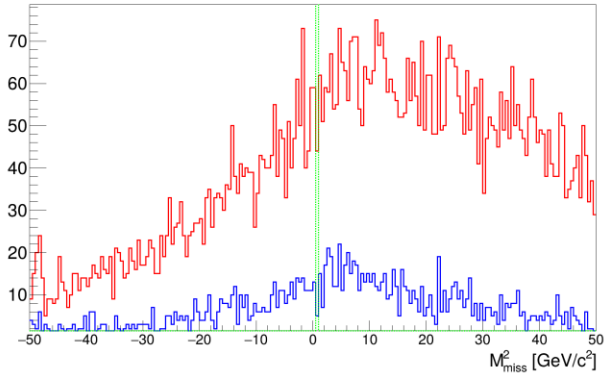
Link to example:
[epic-rad/examples/PythiaRho.C at master · dglazier/epic-rad]

LowQ2 Tagger - Exclusivity







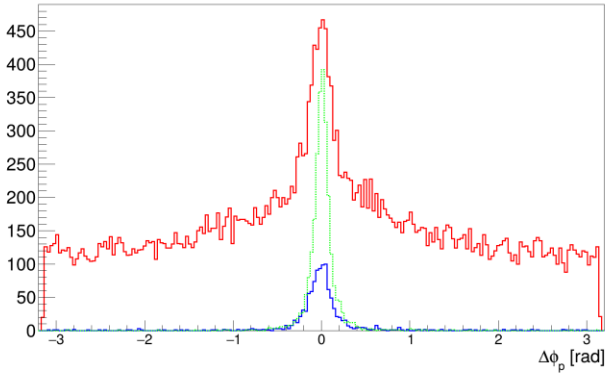


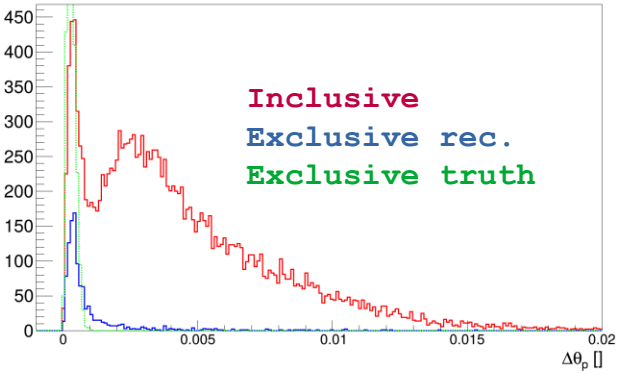
For small t production
Missing momentum gives handle on exclusivity
Significant peak
Requires sufficient resol.

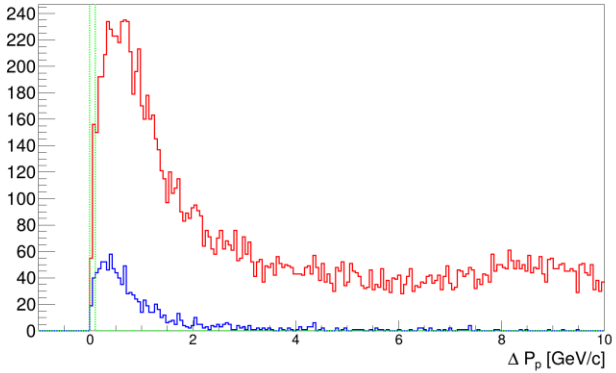
Missing mass gives some limited discrimination

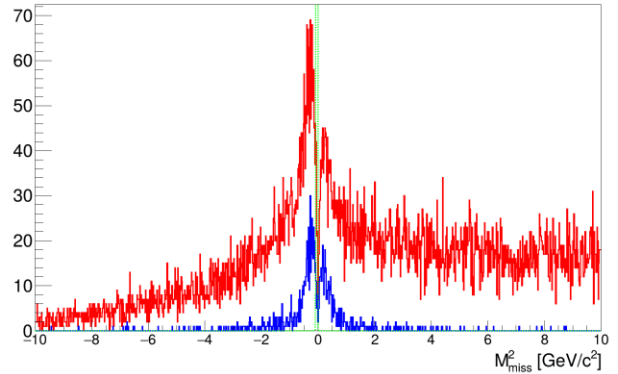
* exclusive/inclusive defined on truth info.

LowQ2 Tagger + Far Forward









Take difference p,θ,φ in calc. Proton(e′π+π-) and measured in FF

Good discrimination of exclusive to inclusive

Total missing mass also useful

* exclusive/inclusive defined on truth info.

Summary

- Low-Q2 Tagger and Physics benchmarks advanced and making good progress.
- Detector benchmarks in place to track changes.
- Nearly ready to run and replace reconstruction model on geometry change.
- Physics benchmark tracking rho resolutions in branch.
- Demonstration of exclusivity separation from Low-Q2 Pythia samples.
- RAD new powerful analysis tool.
  - Simple to run multiple physics analysis simultaneously.
  - More features than demonstrated here.
  - Encourage new users to carry out analysis and request features.

Thank You