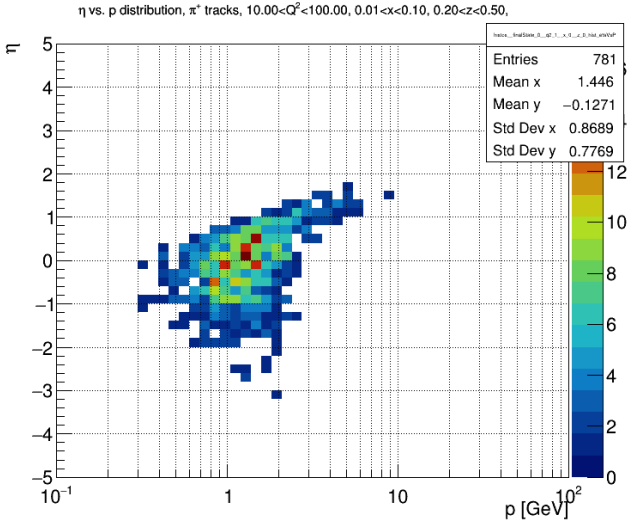
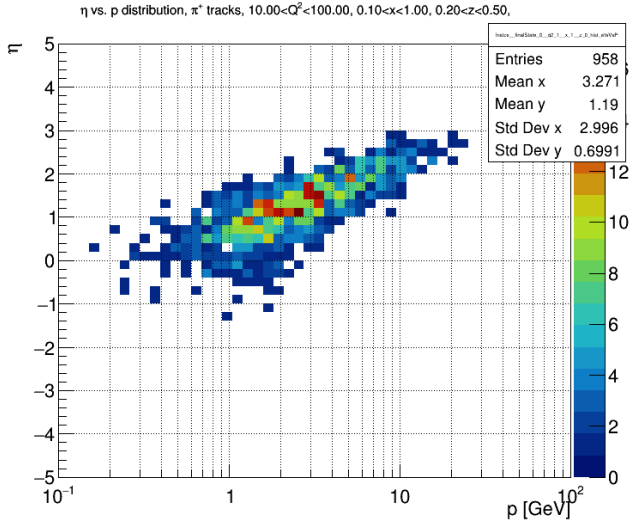
Common Software Status

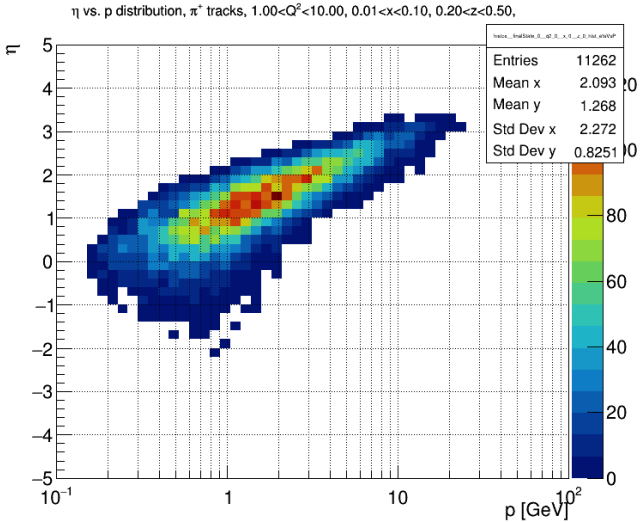
https://github.com/c-dilks/largex-eic

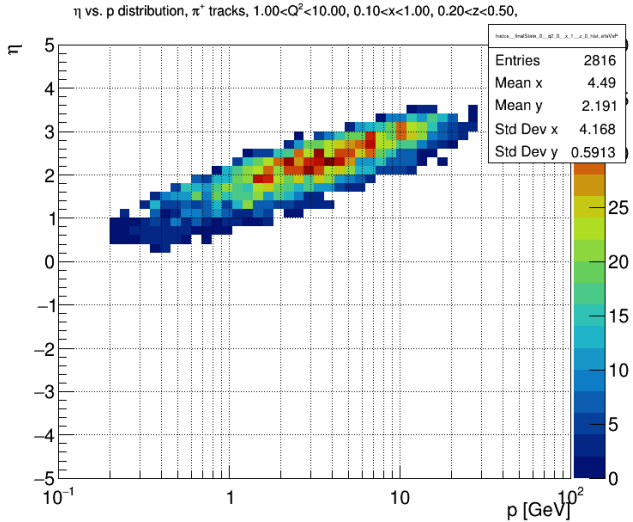
- **Updates**
  - Full simulations implementation nearing completion, and undergoing testing (update from Sanghwa)
  - Asymmetry fit code included as a submodule; validated with “fake” asymmetry injection
  - Fully arbitrary multidimensional binning is possible (on a dev branch)
  - Single particle (pions, kaons, etc.) and jet final states
  - 3D and 4D histogram support (Duane)
- **Plans**
  - Add depolarization factors
  - Asymmetry injection with grids, interpolation (Duane)
  - Resolution studies (IIT Bombay?)

Kinematic Coverage: η vs. p, pions, 0.2<z<0.5

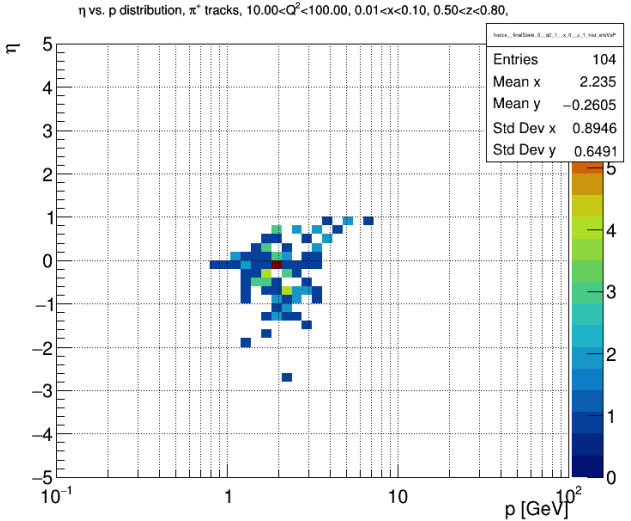


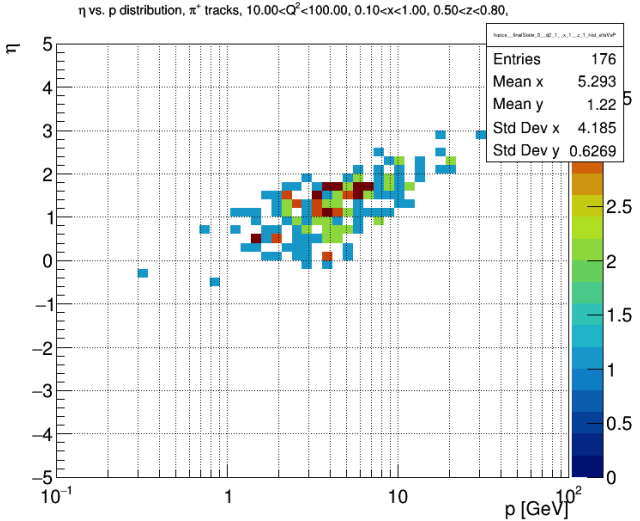


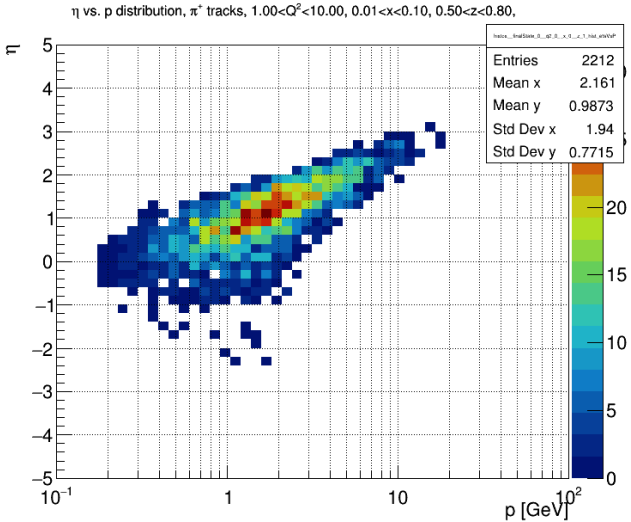


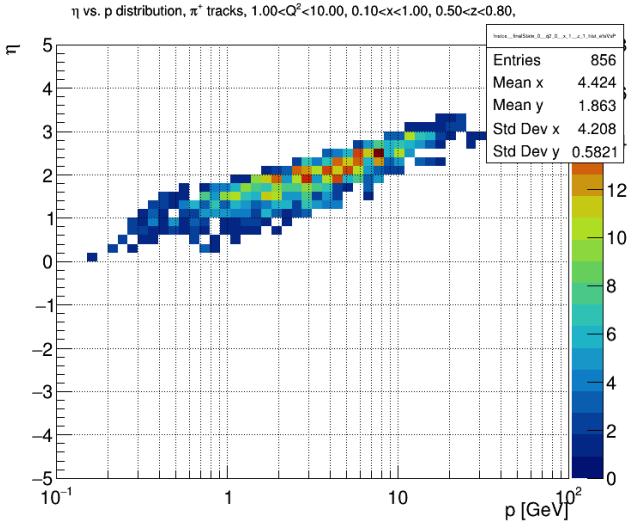


Kinematic Coverage: η vs. p, pions, 0.5<z<0.8

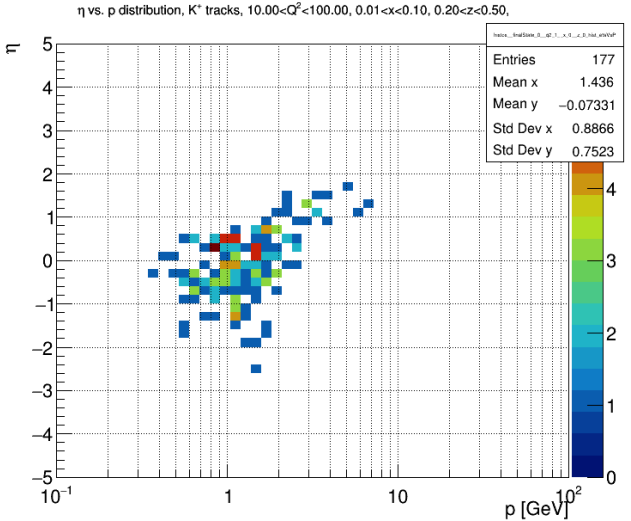


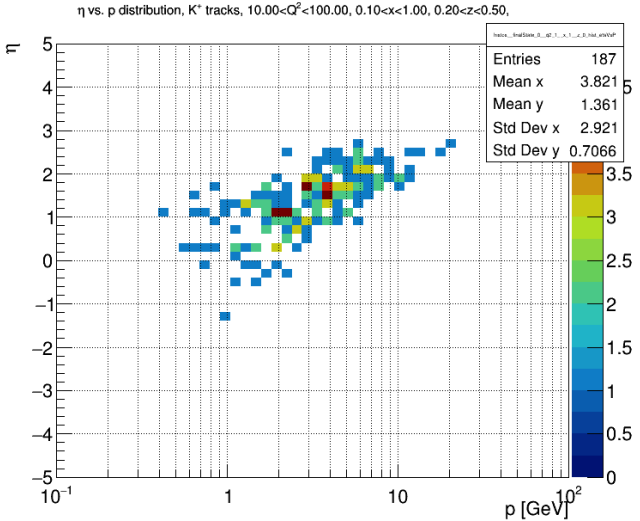


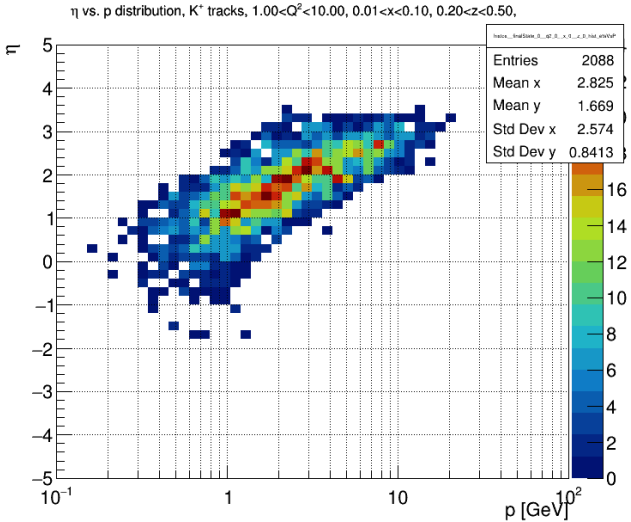


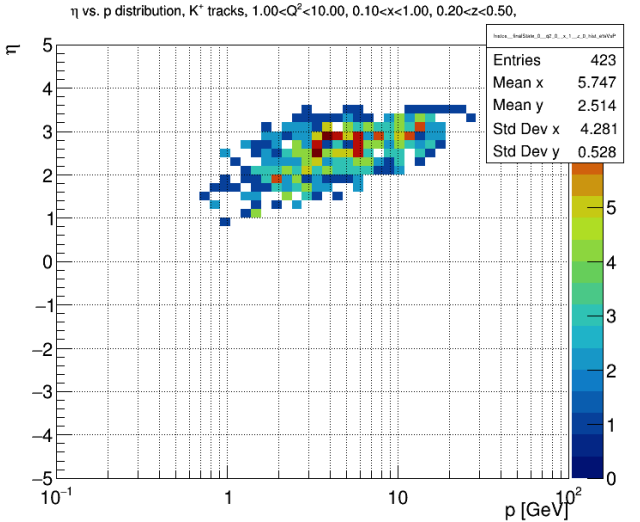


Kinematic Coverage: η vs. p, kaons, 0.2<z<0.5

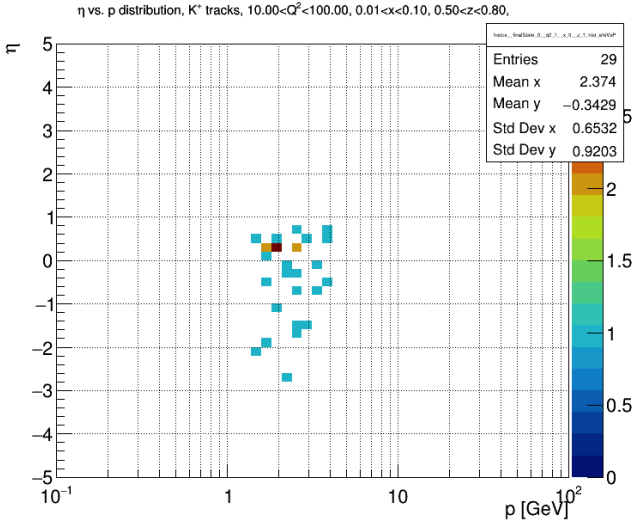


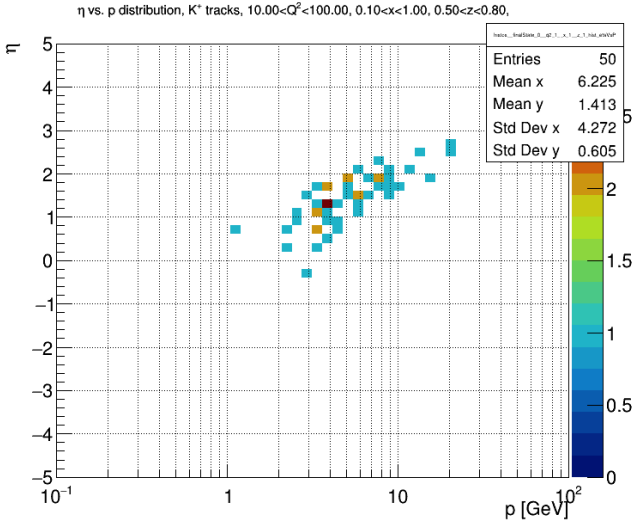


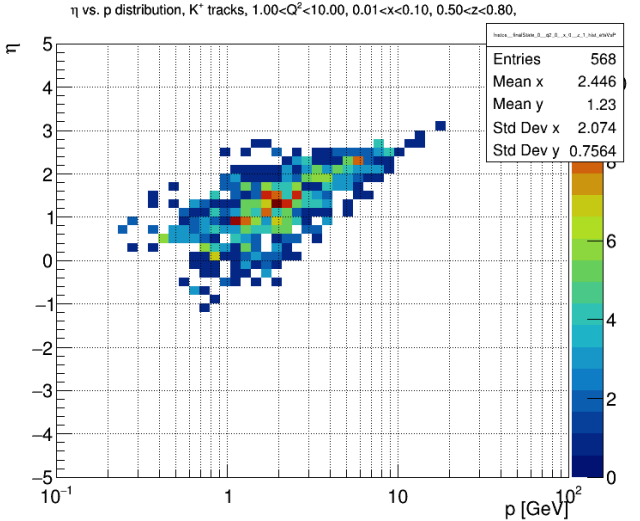


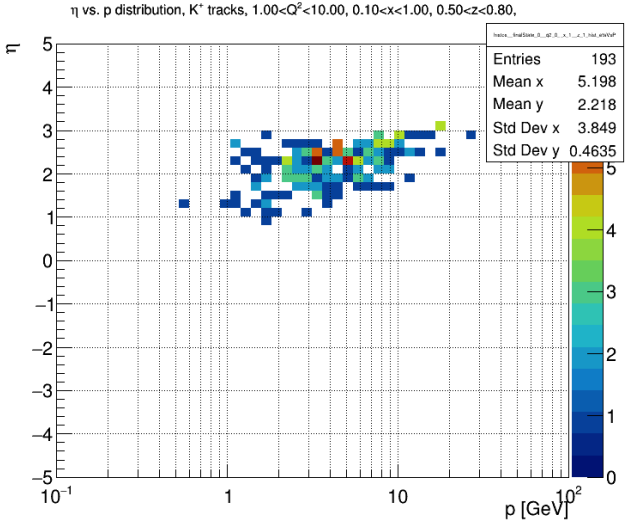


Kinematic Coverage: η vs. p, kaons, 0.5<z<0.8

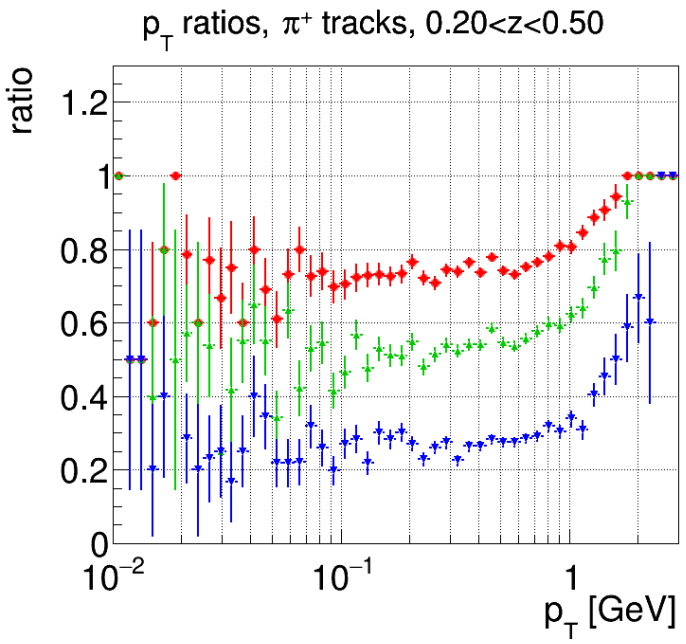


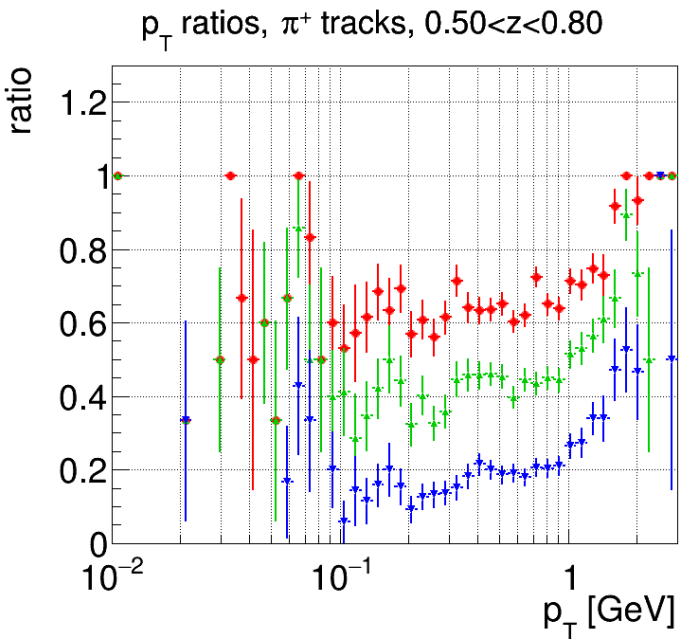




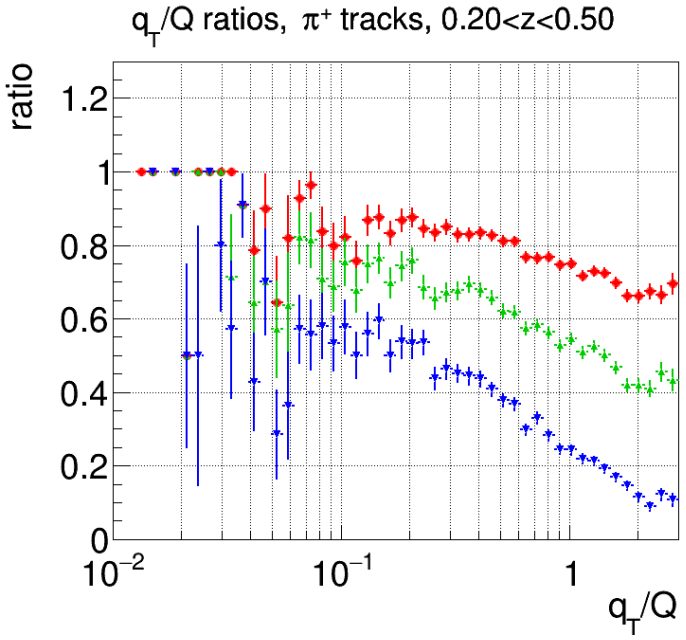


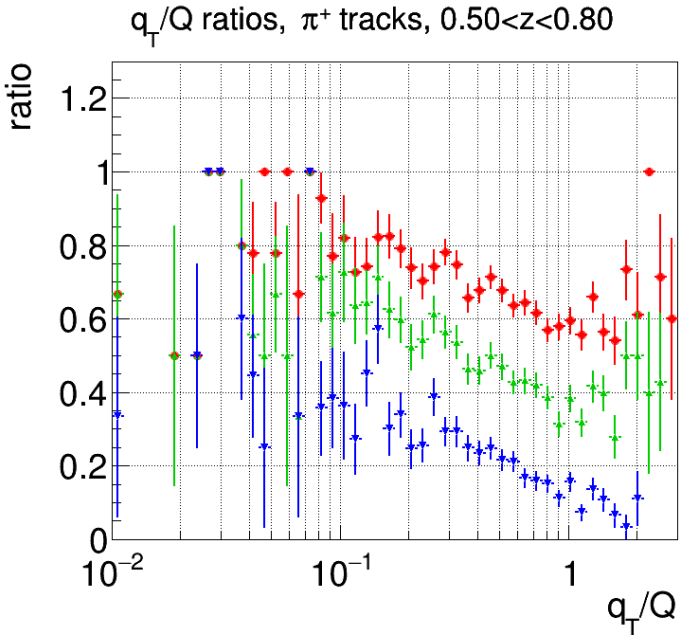
$y>y_{min}$ / $y>0$


p_T ratios, π⁺ tracks, 0.20<z<0.50
ratio
1.2
1
0.8
0.6
0.4
0.2
0
10^-2
10^-1
1
p_T [GeV]

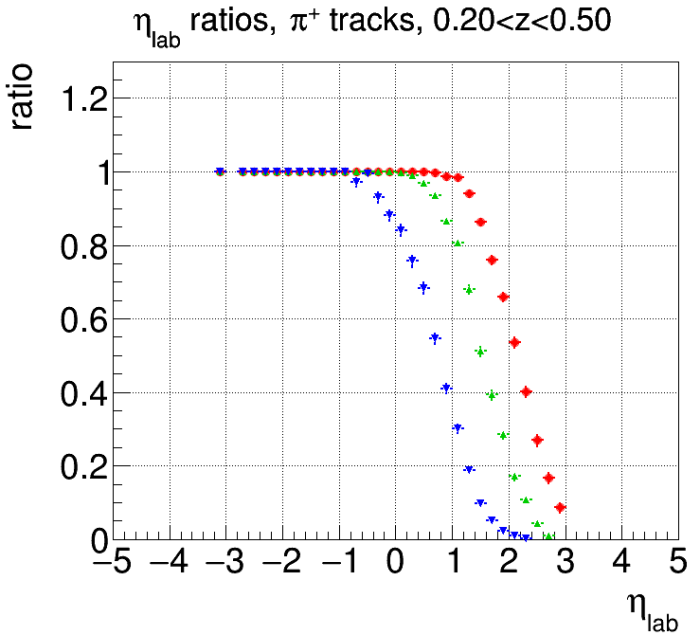

p_T ratios, π⁺ tracks, 0.50<z<0.80
ratio
1.2
1
0.8
0.6
0.4
0.2
0
10^-2
10^-1
1
p_T [GeV]

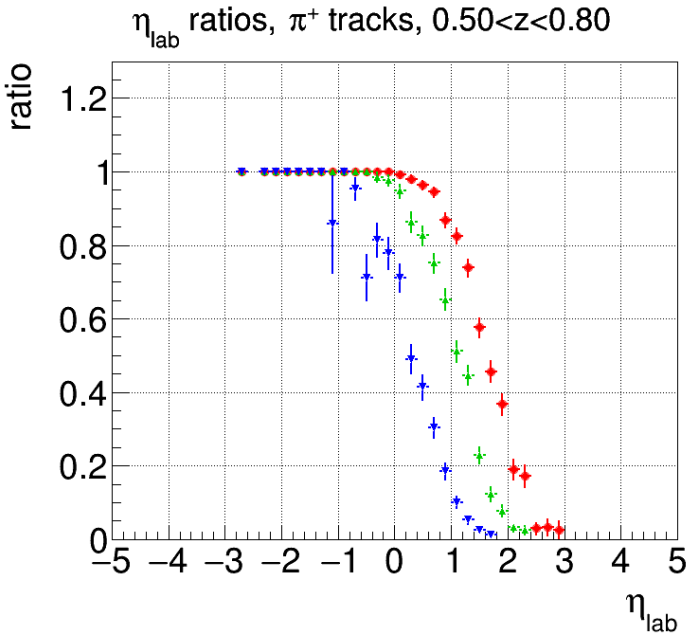
$y>y_{min}$ / y>0


q_T/Q ratios, π^+ tracks, 0.20<z<0.50
ratio
1.2
1
0.8
0.6
0.4
0.2
0
10^{-2}
10^{-1}
1
q_T/Q


q_T/Q ratios, π^+ tracks, 0.50<z<0.80
ratio
1.2
1
0.8
0.6
0.4
0.2
0
10^{-2}
10^{-1}
1
q_T/Q

$y>y_{min}$ / $y>0$


η_{lab} ratios, π^+ tracks, 0.20<z<0.50
ratio
1.2
1
0.8
0.6
0.4
0.2
0
−5 −4 −3 −2 −1 0 1 2 3 4 5
η_{lab}


η_{lab} ratios, π^+ tracks, 0.50<z<0.80
ratio
1.2
1
0.8
0.6
0.4
0.2
0
−5 −4 −3 −2 −1 0 1 2 3 4 5
η_{lab}

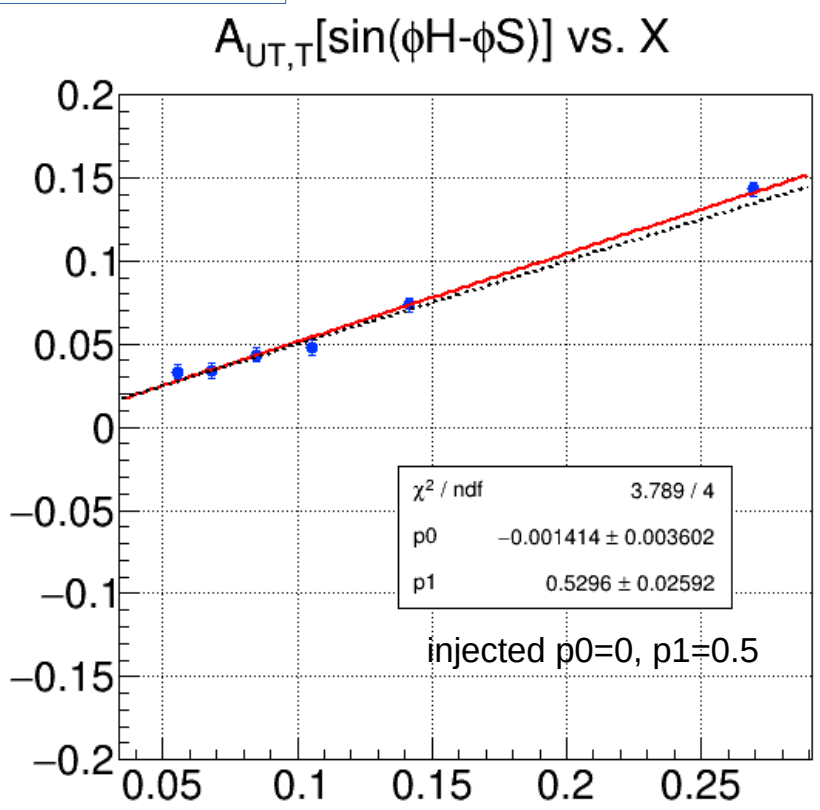
Asymmetry Injection

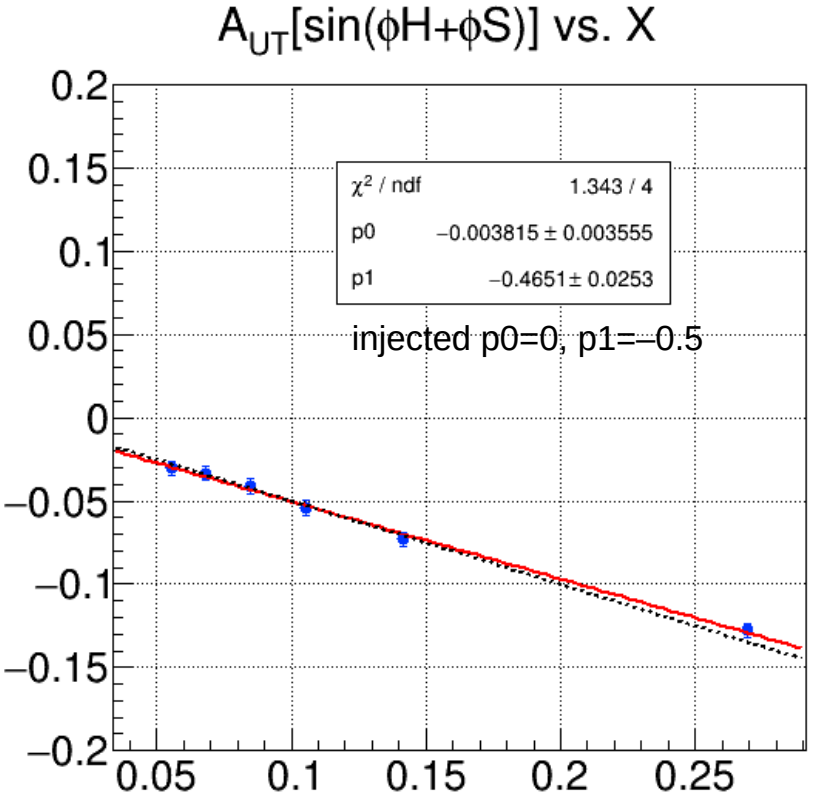
Inject linear functions of x for A_{UT} Sivers and Collins, based on **reconstructed $\{x,\phi_h,\phi_S\}$**

injected

linear fit

points = brufit





Asymmetry Injection

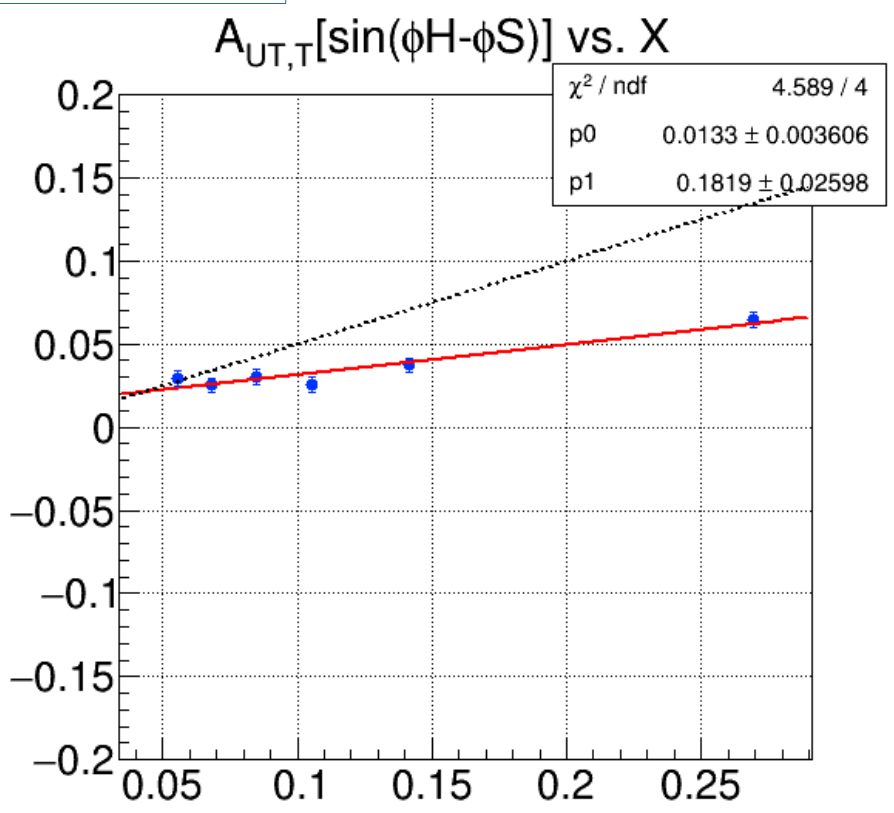
Inject linear functions of x for A_{UT} Sivers and Collins, based on **generated $\{x,\phi_h,\phi_S\}$**

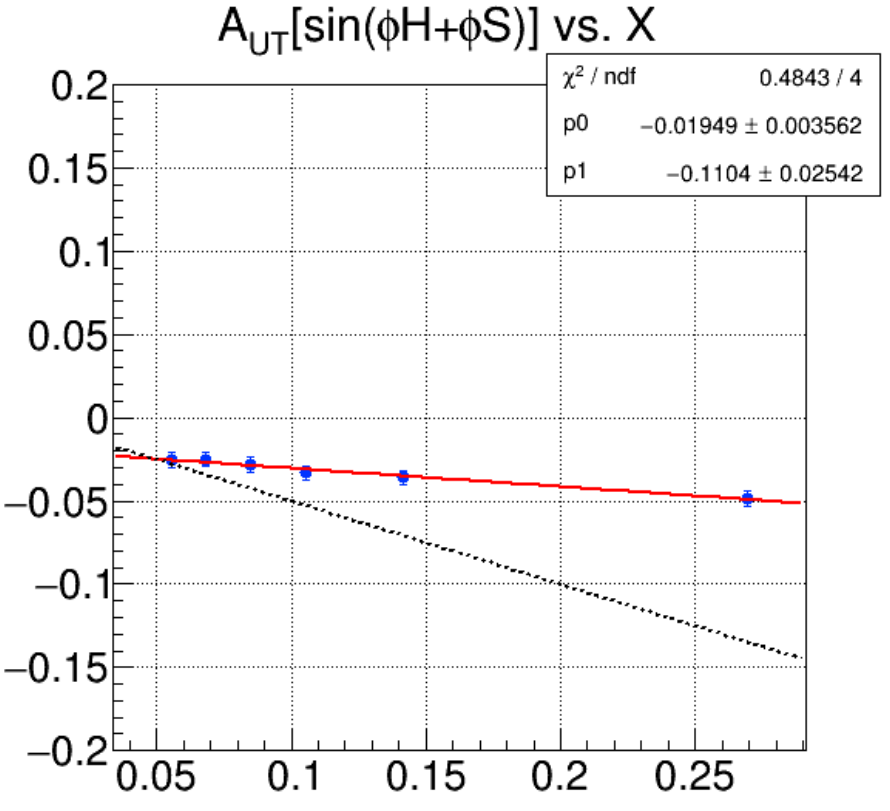
injected

linear fit

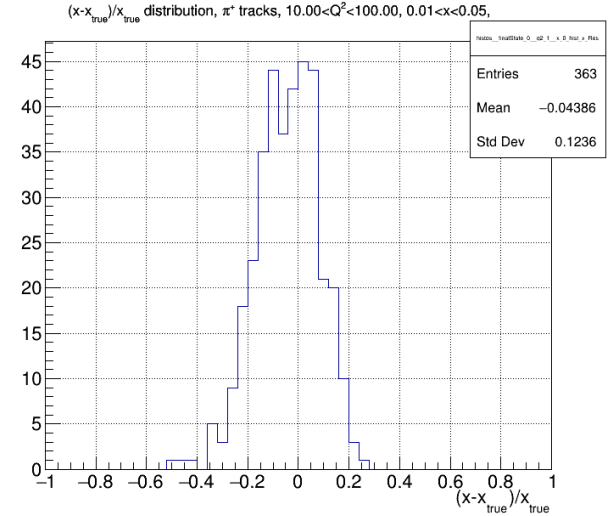
points = brufit

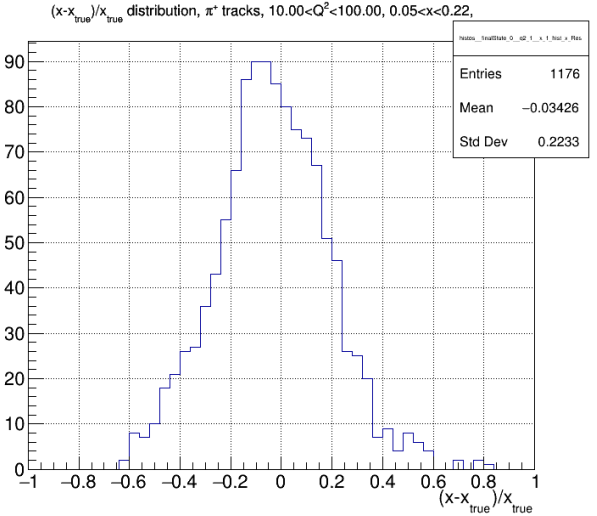
something may be wrong, need to check calculations of generated kinematics...

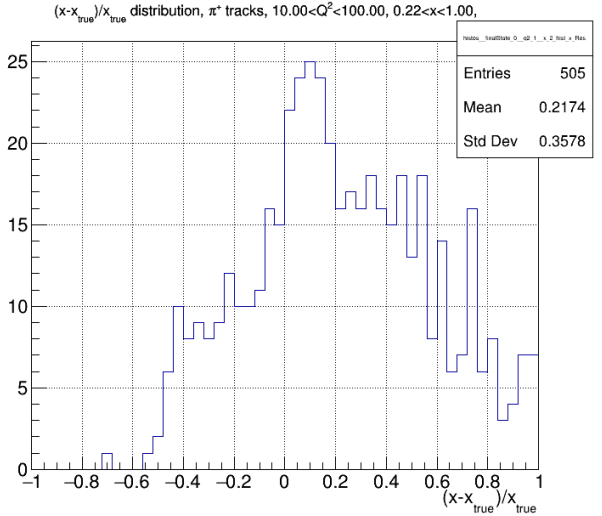


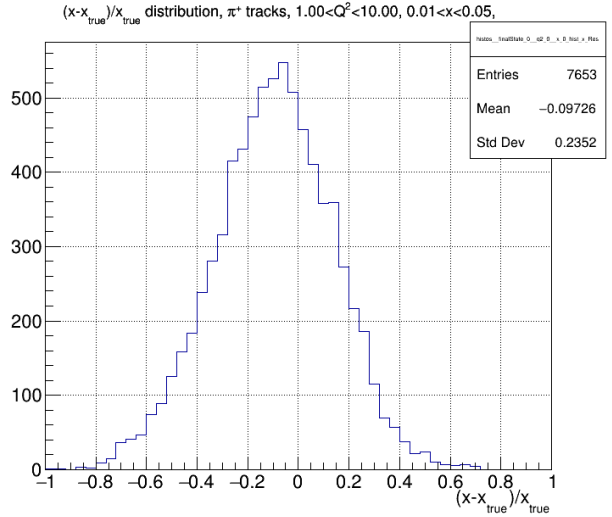


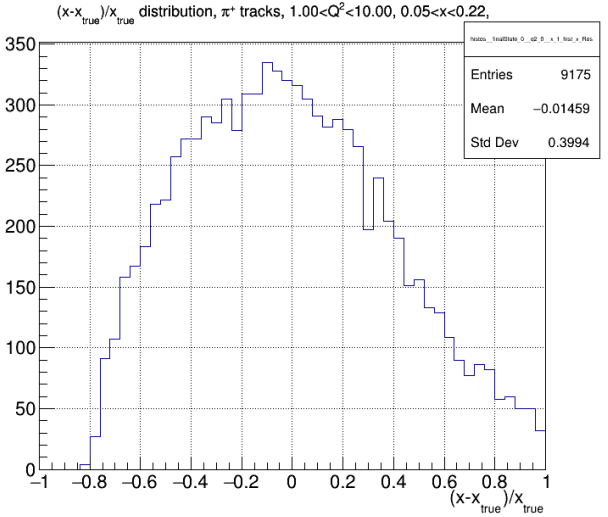
x Resolutions in 6 (x,Q2) bins

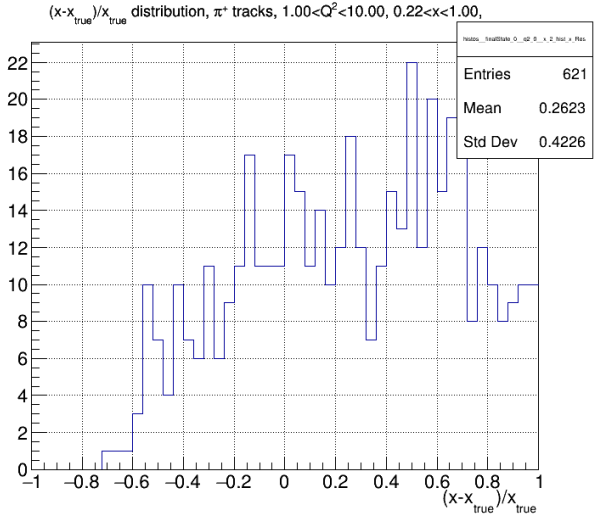




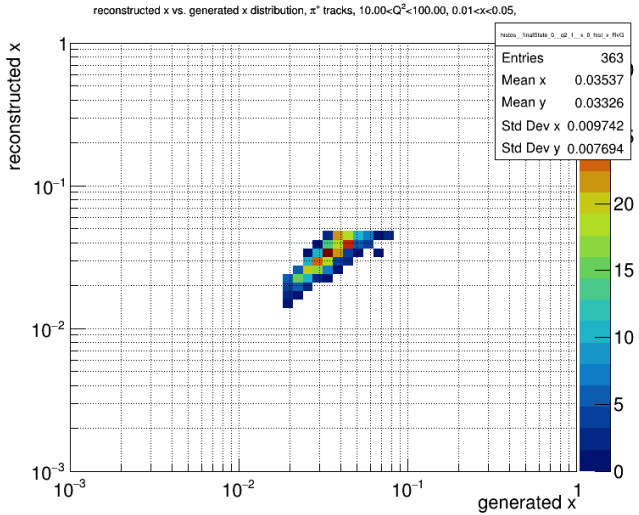


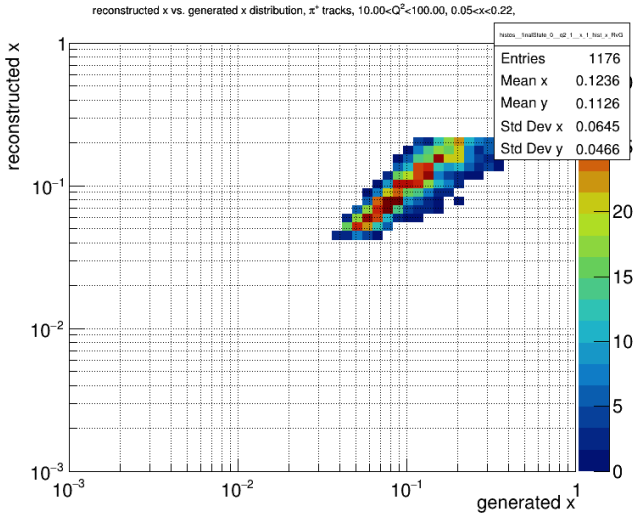


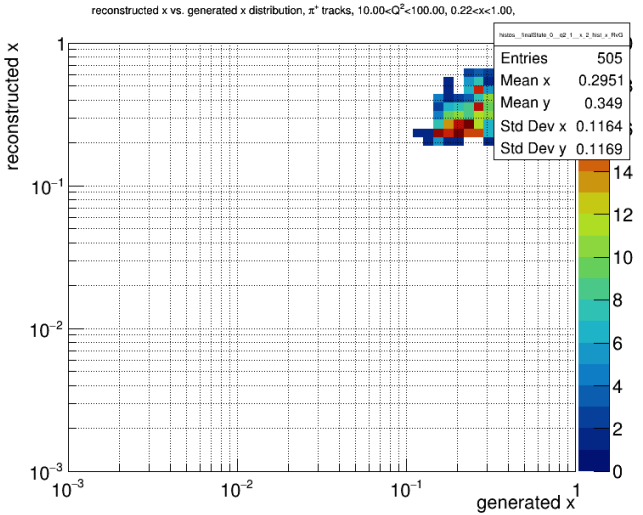


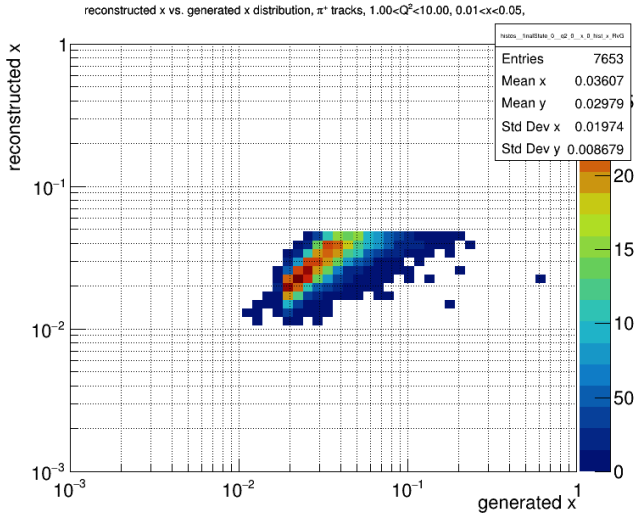


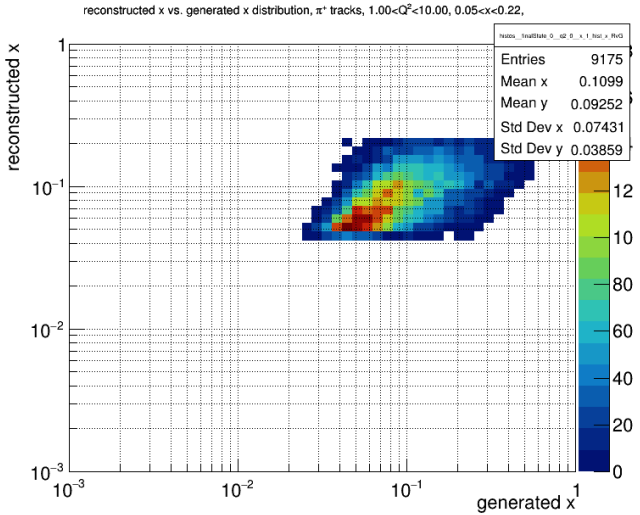
x reconstructed vs. generated

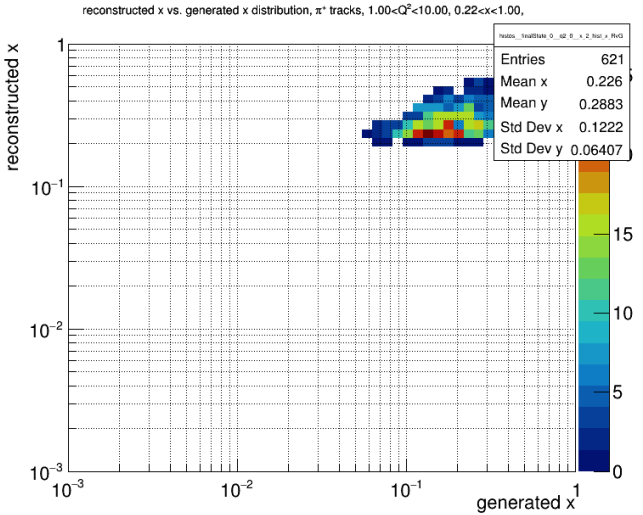




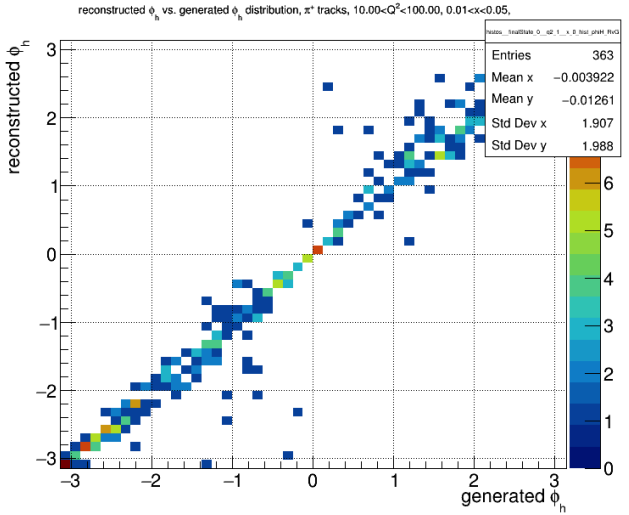


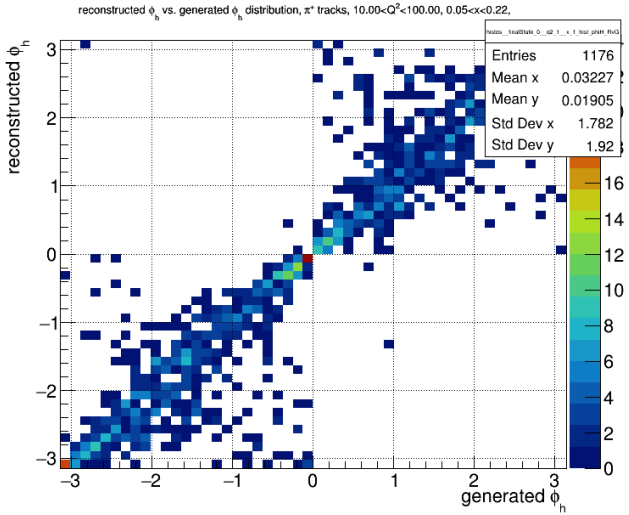


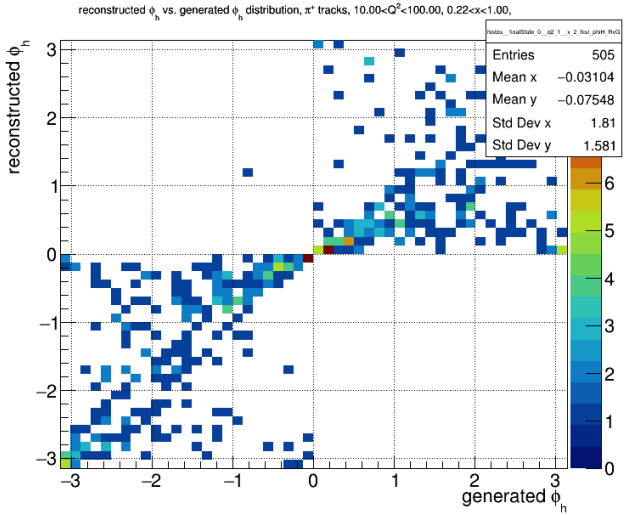


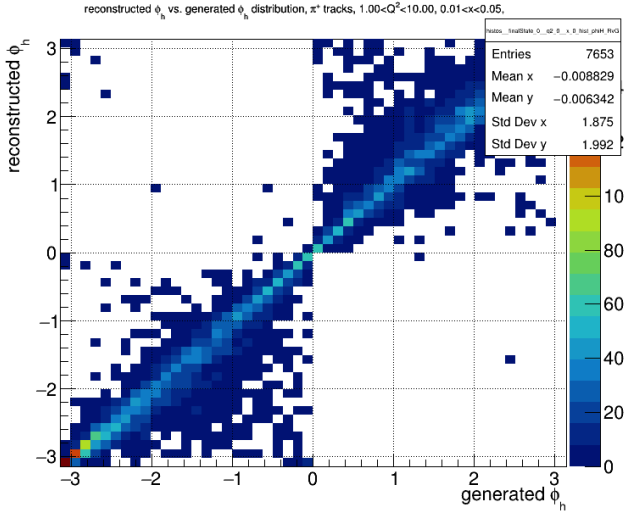


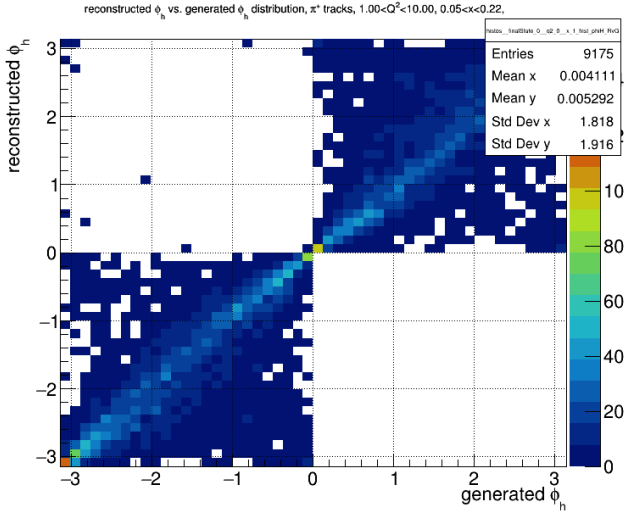
ϕh reconstructed vs. generated

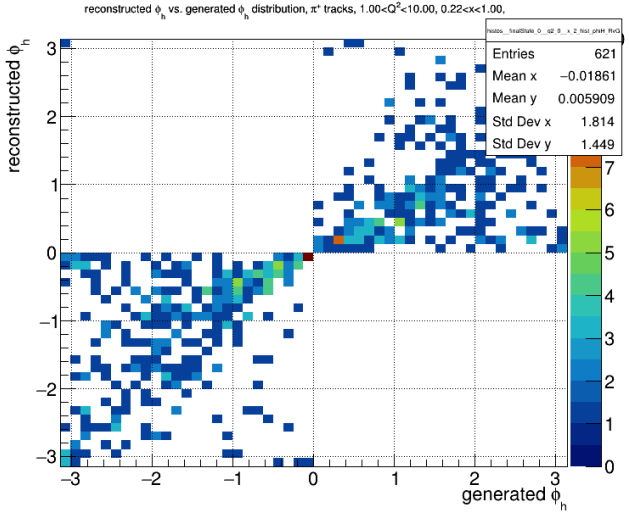




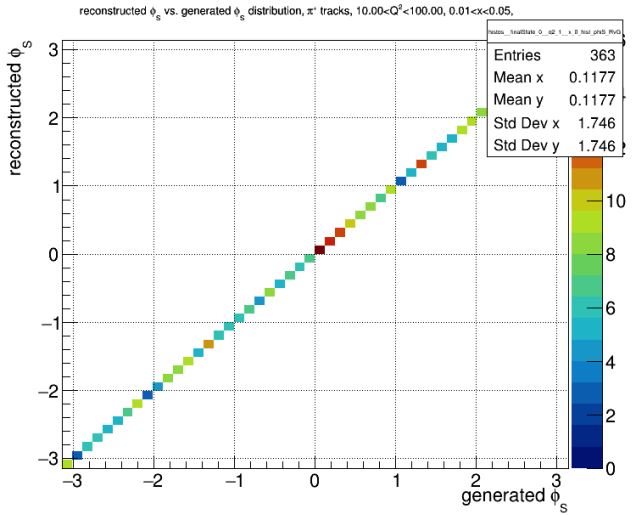


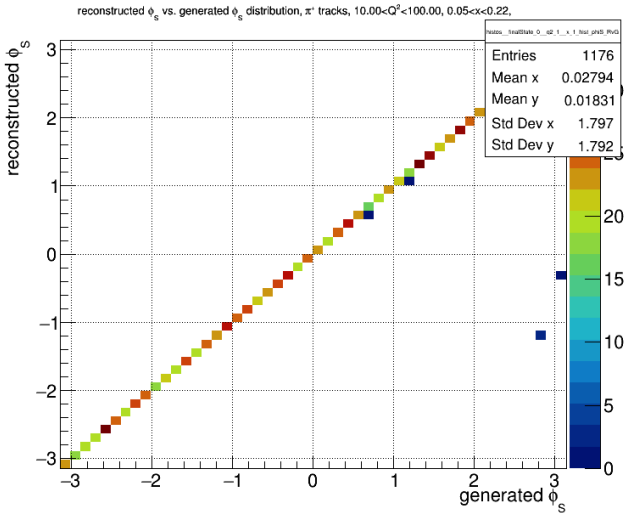


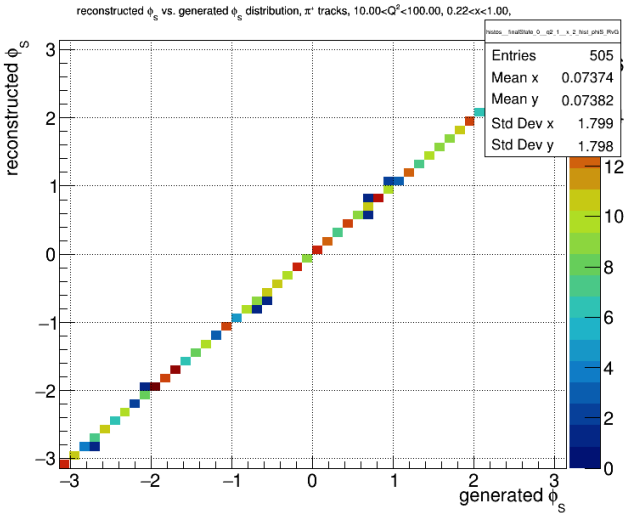


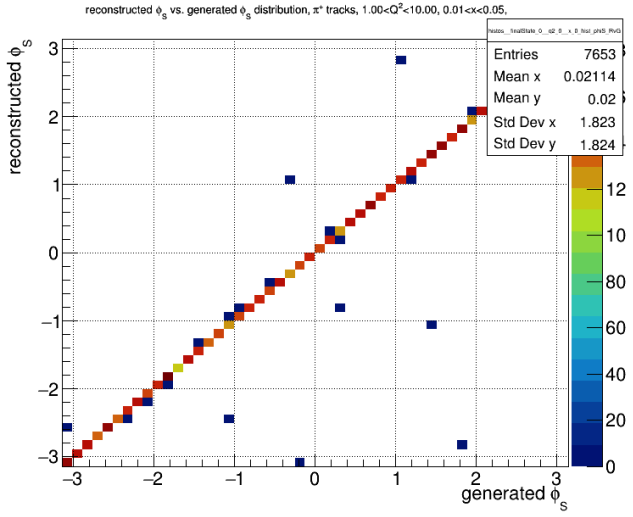


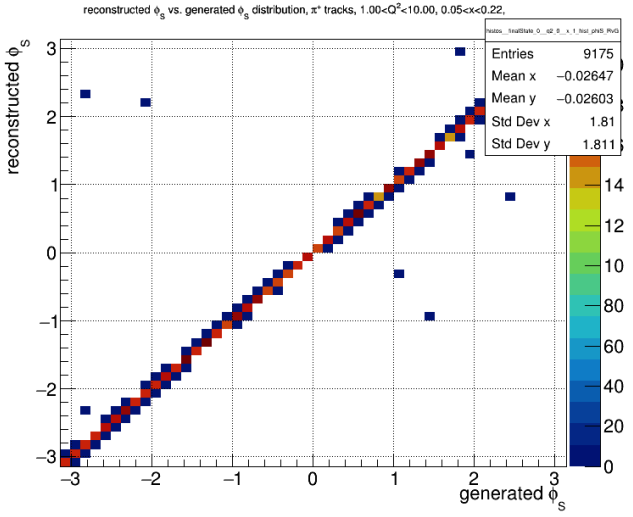
ϕS reconstructed vs. generated

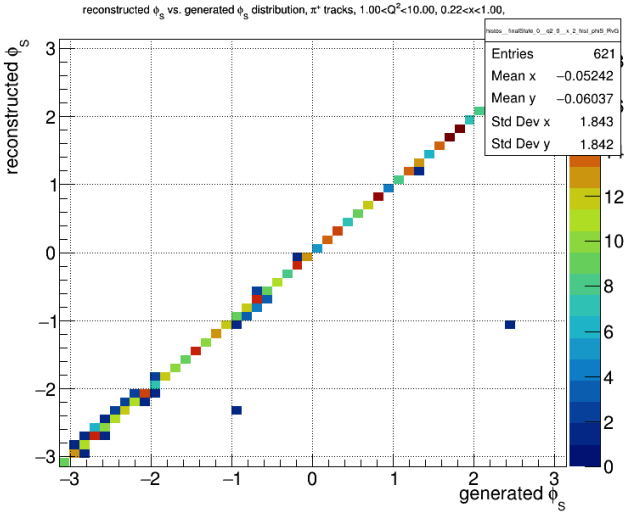












Software technical discussion

Software technical discussion

- **Updates**
  - Fake asymmetry injection is now possible; **TODO: realistic injection models**
  - Asymmetry fit code included as a submodule
  - 3&4D histogram support (Duane)
- **Recent issues**
  - Jet loop memory leak (fixed)
  - Generated kinematics not being filled (fixed)
  - Enhancement: add depolarization factors to Kinematics **(open)**
- **Active branches**
  - *fullsim*: would be good to merge ASAP
  - *coverage-studies*: status update? plan to merge everything except Analysis
  - *dag*: main data structure refactor, requesting feedback, plan to merge soon
- **Miscellaneous**
  - Move macros to subdirectory macro/, or you can create your own subdir

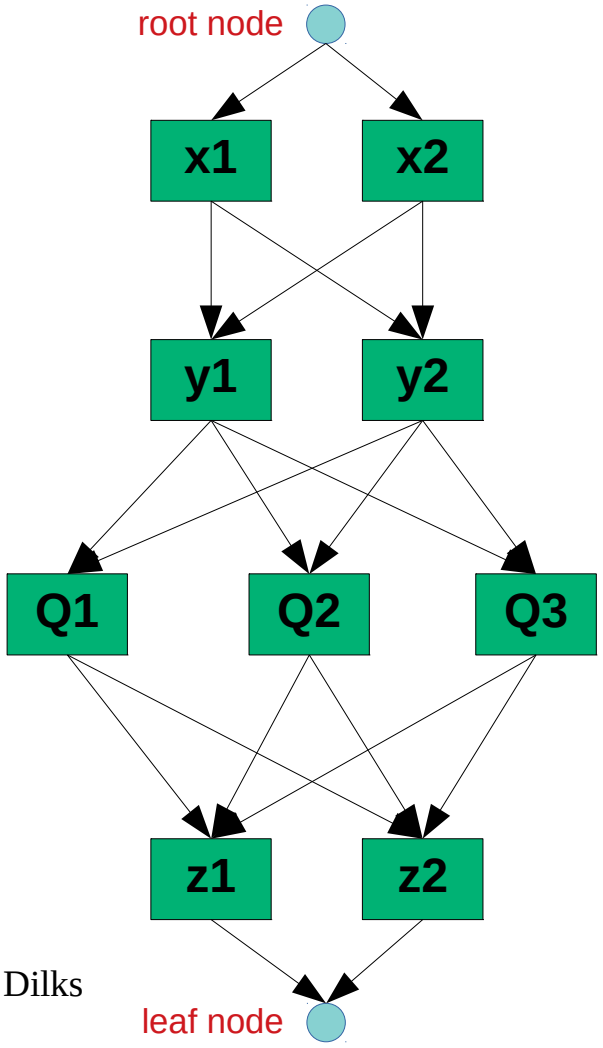
Nested For Loops

~6D arrays are not sustainable!
→ replaced with a graph data structure

```
Histos *histSet[NptBins][NxBins][NzBins][NqBins][NyBins][NfinalStateBins];
Histos *histSetCoverage[NpBins][NxBins][NetaBins][NqBins][NyBins][NfinalStateBins];
Histos *histSetJets[NptjetBins][NzjetBins][NxBins][NqBins][NyBins];
Histos *histSetBreitJets[NptjetBins][NzjetBins][NxBins][NqBins][NyBins][NrecMethodBins];
```

```
// build list of histogram sets to fill
histSetFillList.clear();
for(int bpt : v_pt) {
  for(int bx : v_x) {
    for(int bz : v_z) {
      for(int bq : v_q) {
        for(int by : v_y) {
          if(!CheckDiagonal(bpt,bx,bz,bq)) {
            histSetFillList.push_back(histSet[bpt][bx][bz][bq][by][bFinalState]);
          };
        };
      };
    };
  };
};
```

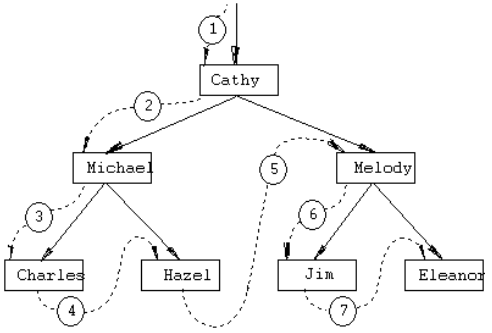
HistosDAG (Directed Acyclic Graph)



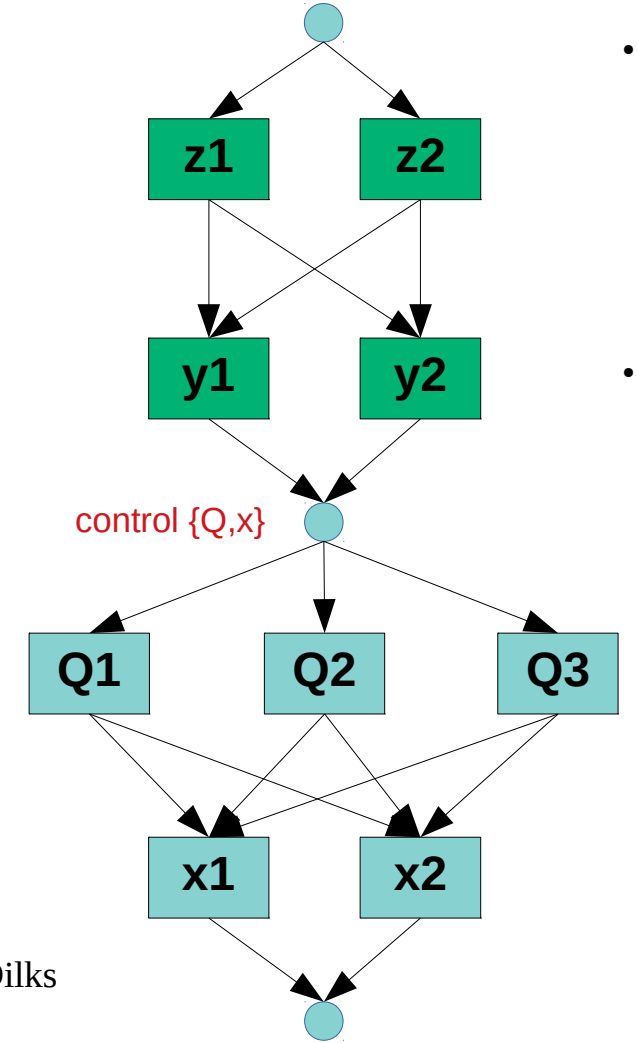
- Construction is controlled by specified binning scheme at the macro level
- Two node types:
  - Bin nodes hold 1-D bin specification, denoted by 
  - Control nodes hold *lambdas*, executable during DAG depth-first traversal, denoted by 
- Each “layer” of bins for a given variable is fully connected to adjacent layers
- Each unique path from root to leaf specifies a single multi-dimensional bin
- A lambda in the leaf node is called the “payload”, since it acts on all multidimensional bins

```
For {x} bins {
  For {y} bins {
    For {z} bins {
      For {Q} bins {
        PayloadOp();
      }
    }
  }
}
```

depth-first traversal



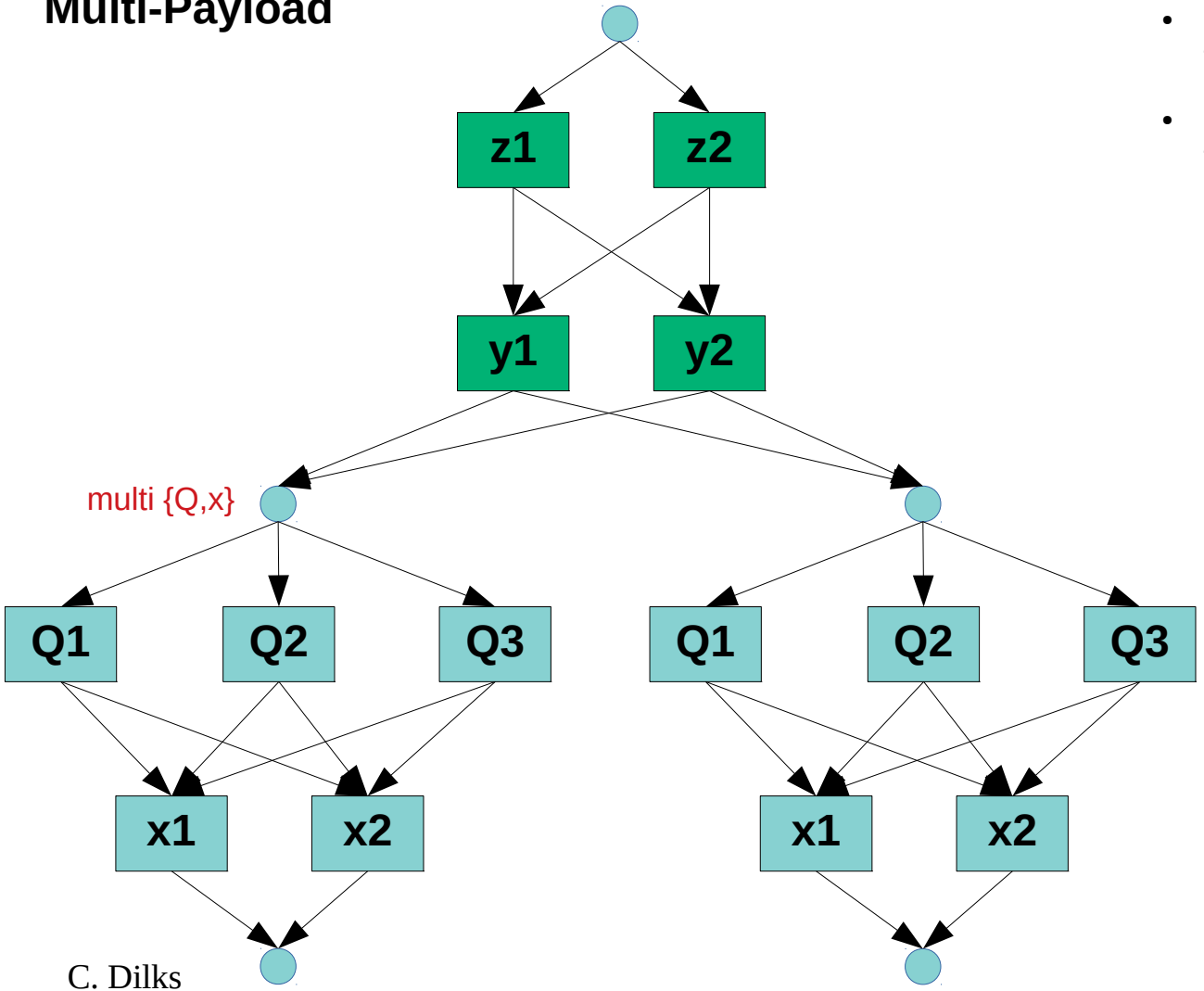
Subloops



- Layers can be rearranged in any desired order
- Control nodes can be inserted anywhere, to create “Subloops”, holding two lambdas:
  - Inbound lambda: executed while depth-first traversal passes through the control node, when descending toward the leaf node
  - Outbound lambda: executed when backtracking through the control node (ascending)
- No need to specify for loops; only need to specify lambdas (operators) and which subloop they act on (red lines)

```
For {z} bins {
  For {y} bins {
    InboundOp({Q,x},f(z,y)); // before subloop
    For {Q} bins {
      For {x} bins {
        PayloadOp();
      }
    }
    OutboundOp({Q,x},f(z,y)); // after subloop
  }
}
```

Multi-Payload



- It is possible to run a subloop as many times as desired, with differing payloads
- Use lambda captures to help pass info from one payload to the next

```
For {z} bins {
  For {y} bins {
    BeforeSubloop1({Q,x});
    For {Q} bins {
      For {x} bins {
        Payload1();
      }
    }
    AfterSubloop1({Q,x});
    BeforeSubloop2({Q,x});
    For {Q} bins {
      For {x} bins {
        Payload2();
      }
    }
    AfterSubloop2({Q,x});
  }
}
```

Status

- Implemented on ‘dag’ branch
- Delphes ‘Analysis’ class has been updated
  - No more nasty for loops anywhere!
  - ~15 binning schemes available
  - Final states include pions, kaons, jets
  - Reconstruction method switch
- Plan to update `AnalysisDD4hep` similarly
- **Tutorials have been updated**

General To Do List

- add depolarization factors (Chris)
- asymmetry injection/interpolation from grids (Duane)
- kinematic coverage plots (Connor)
- full simulation (Sanghwa)
- resolution studies (IIT Bombay)
- weights for Q2-min bins (_____)