DVCS ep updates

O. Jevons

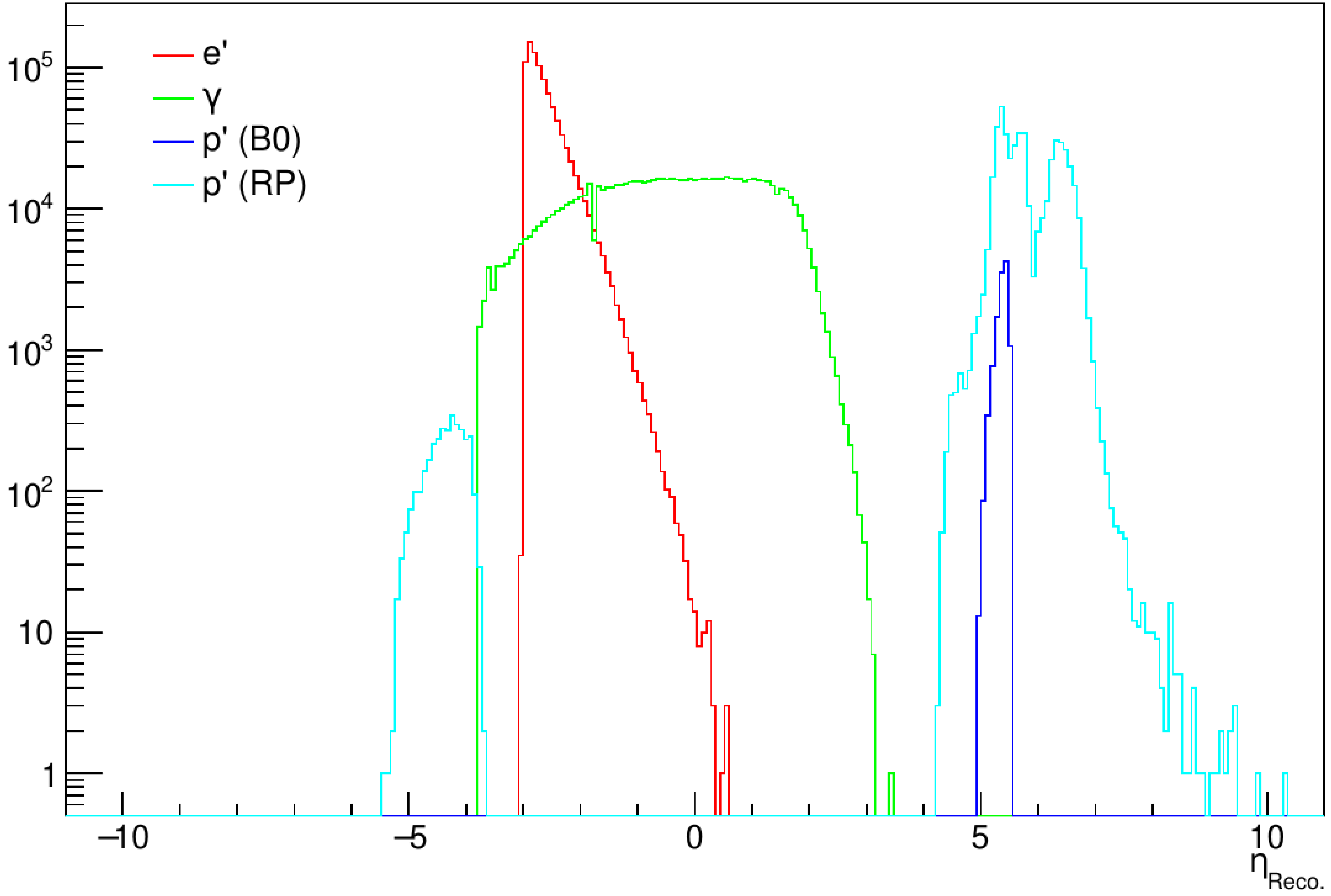
Since last meeting

- Testing following ePIC meeting in Frascati

- Checking need for proton theta cut in latest simulation campaigns.
- (Brief) Test of ePIC PID efficiency.
- Testing InclusiveKinematics branches against manual calculations.
  - Testing different methods to calculate kinematics.
- Good ol' Trento Phi

Need for proton theta cuts?

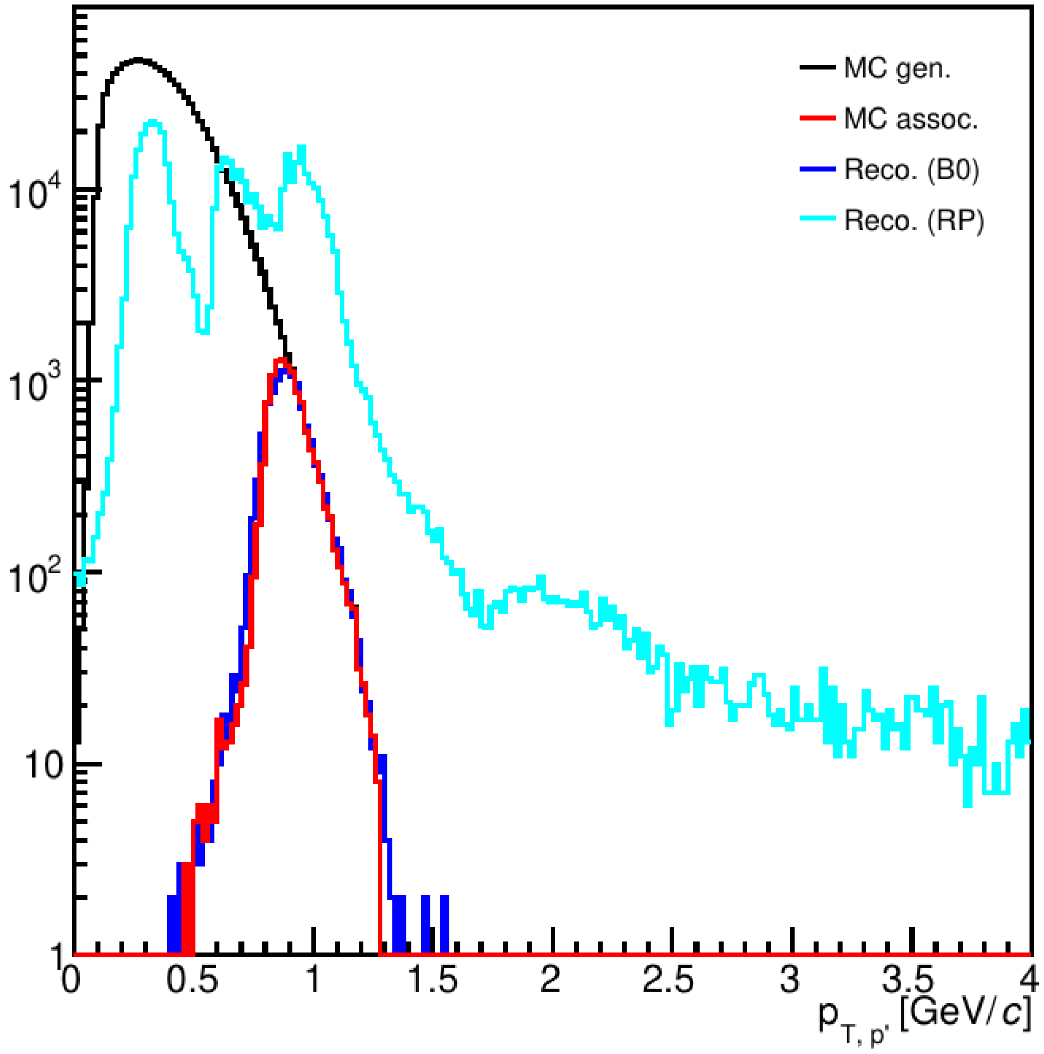
24.10.1 campaign w/out RP theta cuts



Assumption about poor Roman Pot reconstruction still made for 24.10.1 campaign.

Actually checking this confirms assumption – cut on RP track geometry still needed.

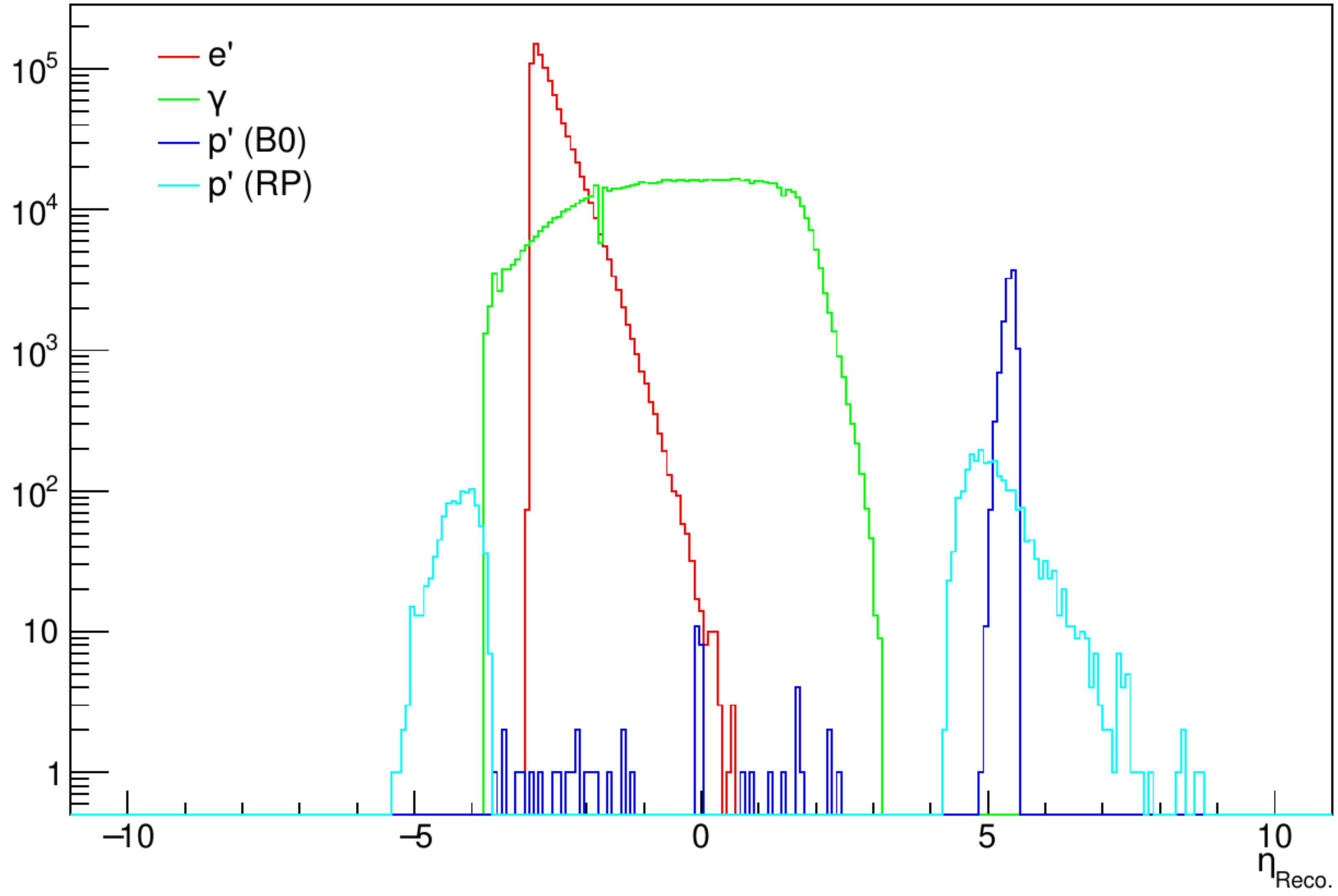
24.10.1 campaign w/out RP theta cuts



Reconstruction of p_T not enough to provide appropriate cut.

Will maintain current status: track theta

24.12.0 campaign w/out RP theta cuts

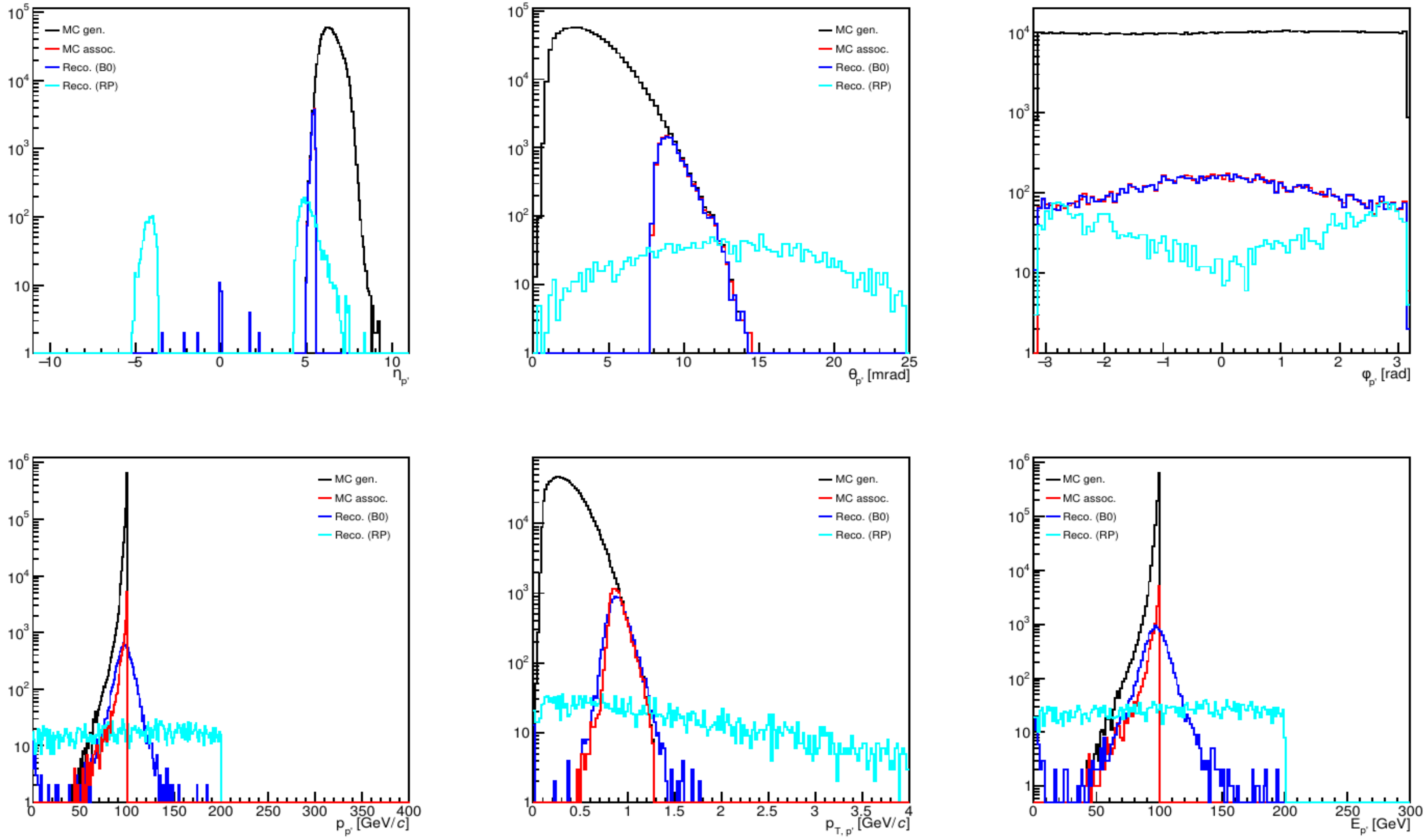


Still poor RP reconstruction in December campaign.

Now, MANY PROTONS MISSING?

Also, some B0 proton tracks are reconstructed w/in the barrel?

24.12.0 campaign w/out RP cuts



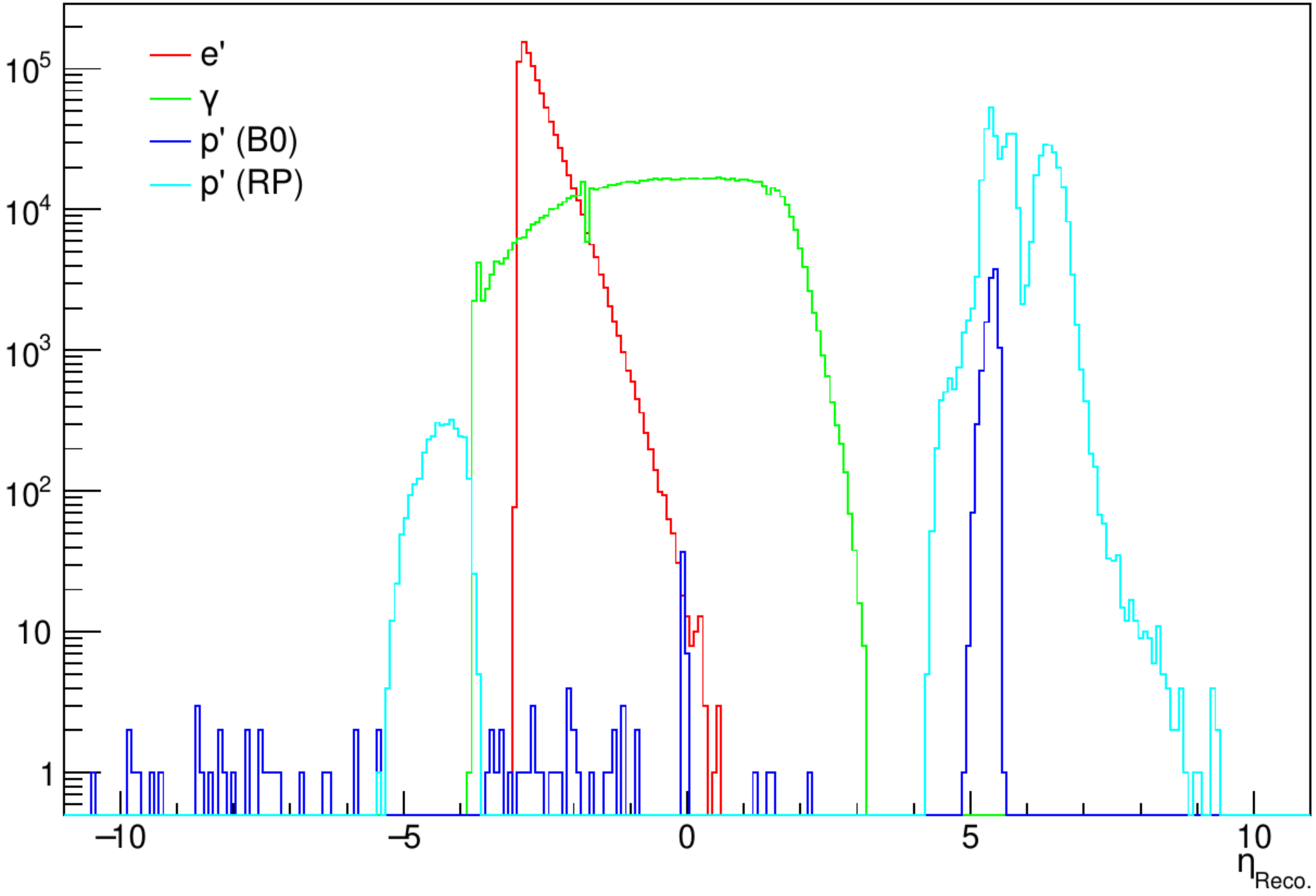
RP reconstruction now worse than before.

Flat energy distribution?

Theta distribution?

Need to check 25.01.1 campaign.

25.01.1 campaign w/out RP theta cuts



Need to maintain proton theta cuts.

B0 tracks seem to show outliers outside B0 coverage.
-> Change between 24.10.1 and 24.12.0 campaigns?

Need to plot kinematics for events outside expected coverage.

Discussion following presentation

- Spoke to A. Jentsch after presenting to PWG.
- Problem identified:
  - EICrecon using the wrong magnet geometry file for RP reconstructions.
  - Had already been raised by Alex back in October, but hadn't been acted on for the campaigns.
- Fix should now be in place ready for February campaign.

InclusiveKinematics testing

Background and formulae

Test of InclusiveKinematics branches

- InclusiveKinematics branches automatically calculated by EICrecon.
- Methods used for calculations:
  - "Electron" method (only uses scattered electron information)
  - Jaquet-Blondel (JB) method (only uses information about hadronic final state)
  - Double Angle (DA) method
  - Sigma method.
  - e-Sigma method.
- Last 3 use a combination of all final state particles for kinematic calculations.
- (All methods use the beam particles).

Inclusive kinematics: electron method

- Only uses information from the scattered electron.
- "Standard" formulae for Q^2, x and y

$$Q^2 = -(k - k')^2 \qquad x = \frac{Q^2}{2q.p} \qquad y = \frac{q.p}{k.p}$$

Inclusive kinematics: JB method

- Only uses information from the hadronic final state.
  - Everything *except* the scattered electron.
- Formulae rely on combined quantities from HFS: $p_{T,h}$ and Σ_h

$$p_{T,h}^2 = (\Sigma p_x)^2 + (\Sigma p_y)^2 \qquad \Sigma_h = \Sigma E - \Sigma p_z$$

$$y = \frac{\Sigma_h}{2E_e} \qquad Q^2 = \frac{p_{T,h}^2}{1-y} \qquad x = \frac{Q^2}{4E_e E_p y}$$

Beam electron energy

Beam proton energy

Inclusive kinematics: DA method

- Uses a combination of scattered electron and HFS information.
  - Only cares about the track angle of the scattered electron.
- Good if you don't have good energy resolution for the e^-.
- ($p_{T,h}$ and Σ_h definitions as previously shown).

$$\gamma = \cos^{-1}\left(\frac{p_{T,h}^2 - \Sigma_h^2}{p_{T,h}^2 + \Sigma_h^2}\right)$$

$$y = \frac{\tan(\theta_e/2)}{\tan(\theta_e/2) + \tan(\gamma/2)} \qquad Q^2 = \frac{4E_e^2}{\tan(\theta_e/2)}\frac{1}{\tan(\theta_e/2) + \tan(\gamma/2)} \qquad x = \frac{Q^2}{4E_e E_p y}$$

Inclusive kinematics: Σ method

- Uses a combination of information from all final state particles.
- Introduces an equivalent for $p_{T,h}$ and Σ_h, but for the scattered electron.
- The afterburner procedure needs to be undone before this method is used.

p_T of the scattered electron

$$\Sigma_{tot} = \Sigma_e + \Sigma_h$$

(E – p_z) of the scattered electron

$$y = \frac{\Sigma_h}{\Sigma_{tot}}$$

$$Q^2 = \frac{p_{T,e}^2}{1-y}$$

$$x = \frac{Q^2}{4E_eE_py}$$

Inclusive kinematics: eΣ method

- Uses a mix of the Σ method and just the scattered electron information to calculate different quantities.

$$y_\Sigma = \frac{\Sigma_h}{\Sigma_{tot}}$$

$$y_e = \frac{\Sigma_e}{2E_e}$$

$$Q_\Sigma^2 = \frac{p_{T,e}^2}{1 - y_\Sigma}$$

$$y_{e\Sigma} = \frac{Q_e^2}{4E_e E_p x_\Sigma}$$

$$Q_e^2 = \frac{p_{T,e}^2}{1 - y_e}$$

$$x_\Sigma = \frac{Q_\Sigma^2}{4E_e E_p y_\Sigma}$$

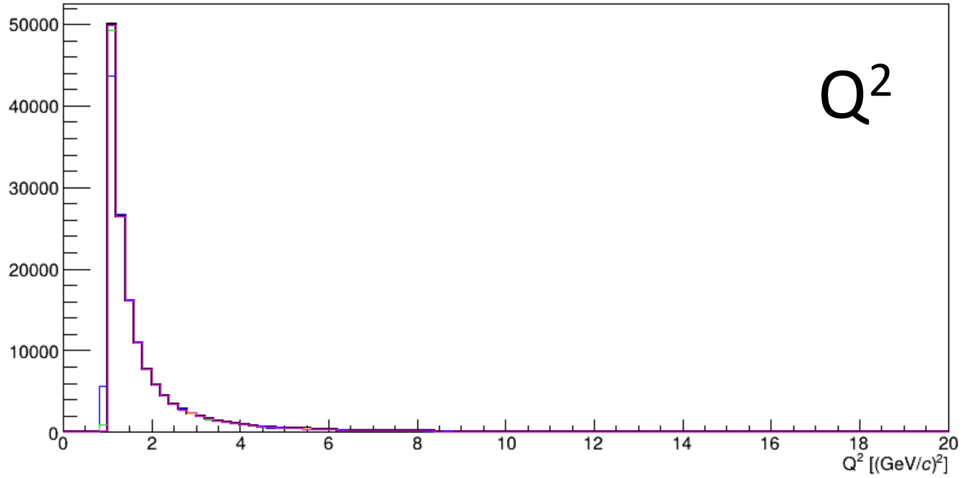
Notes on the InclusiveKinematics branches

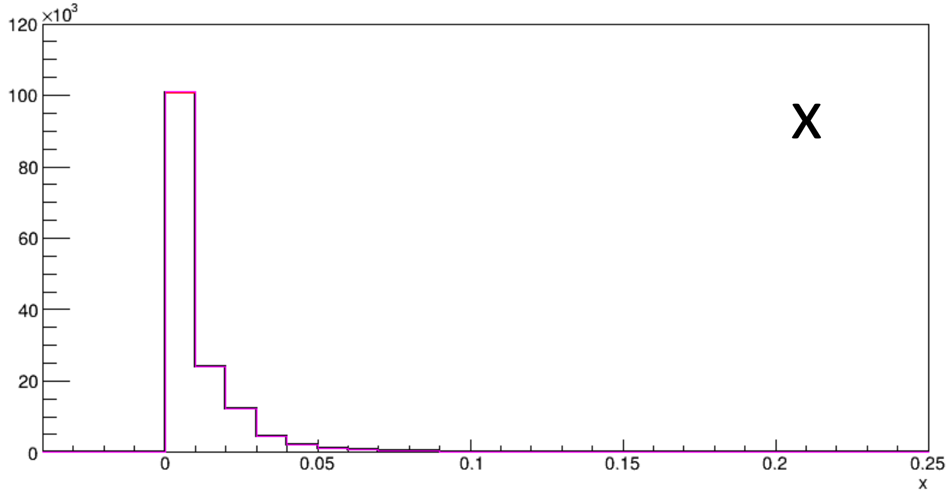
- The "Truth" branch uses the MC information and the electron method for its variables.
- All other branches use the reconstructed particle collections.
- Both the truth and reconstructed branches assume the scattered electron.
  - First non-beam electron in the collection.
- HadronicFinalState calculations only use the ReconstructedParticles branch.
  - Scattered electron identified from MCParticle associations.
  - Misses particles not present in RecoParticles (eg. Roman Pot tracks).

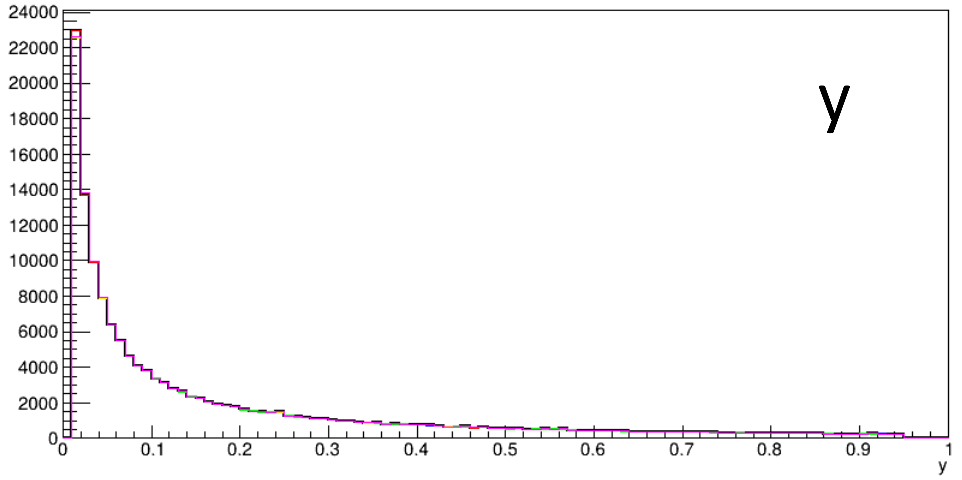
InclusiveKinematics testing

Results – Raw distributions

IncKinTruth vs manual calc. (MCParticles)



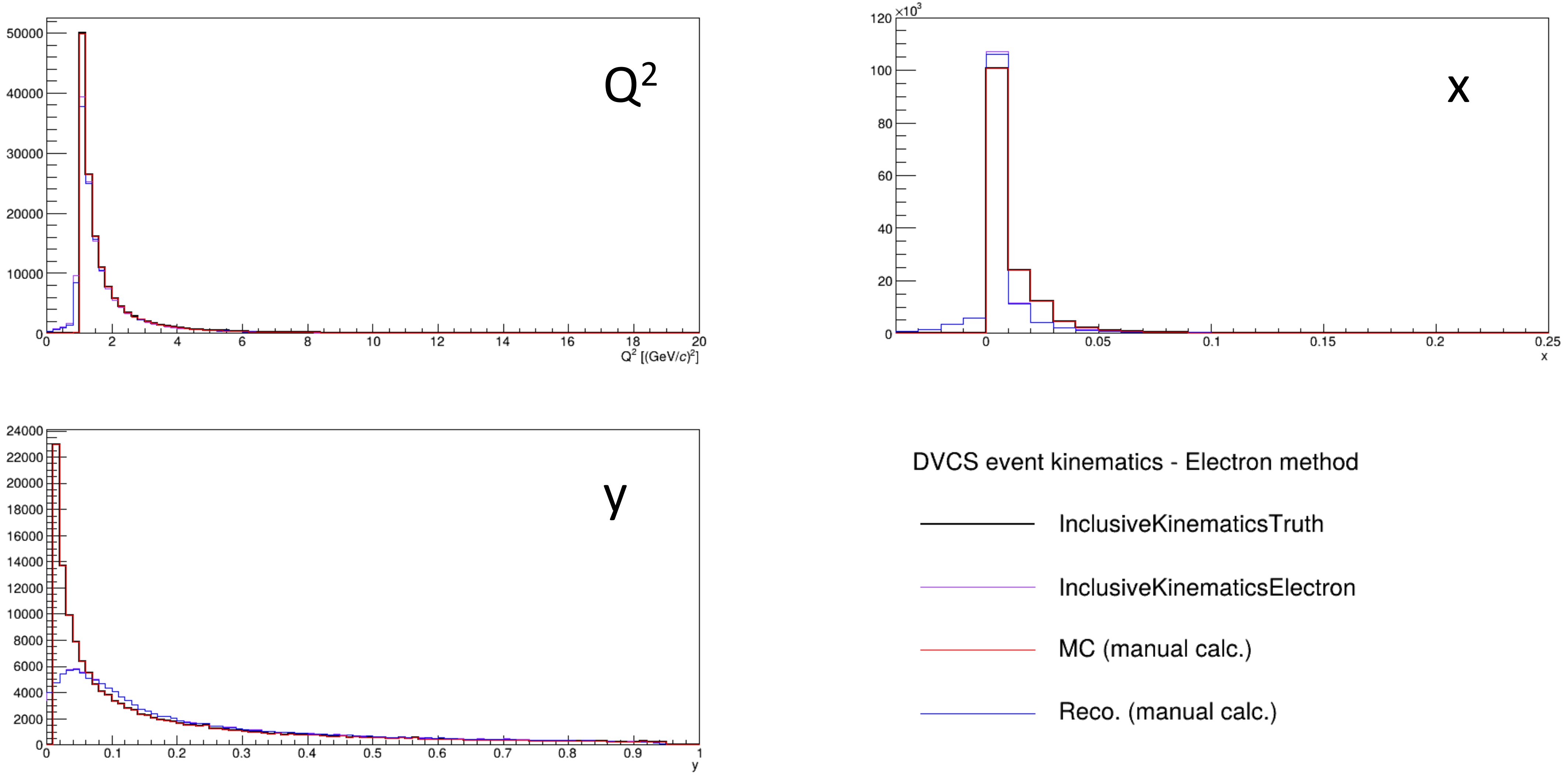




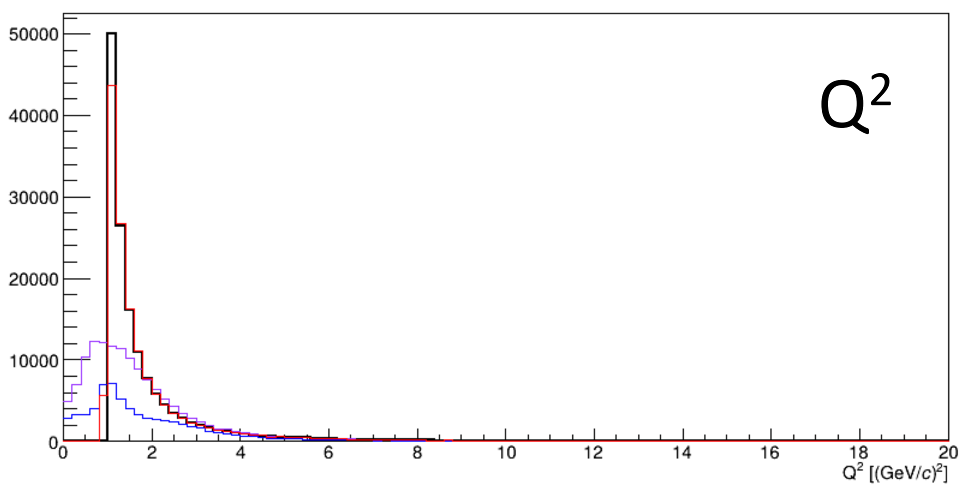
DVCS event kinematics - TRUTH

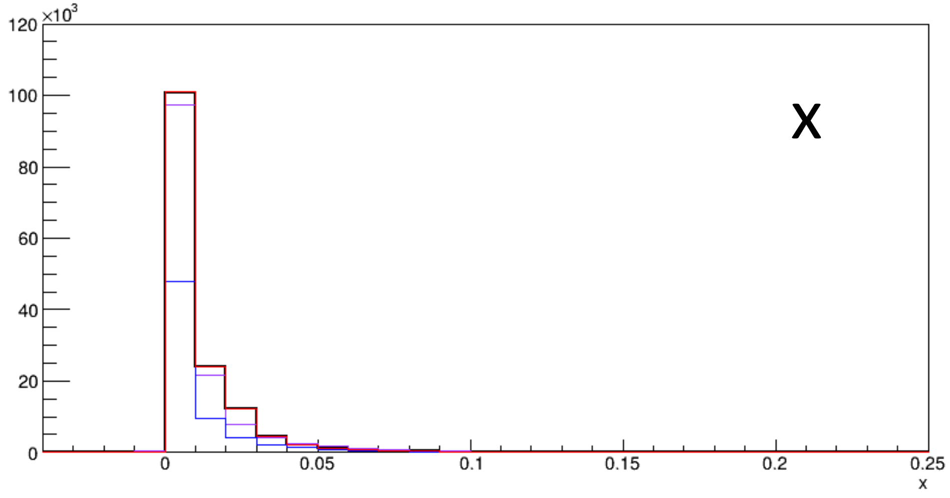
- InclusiveKinematicsTruth
- MC (manual) - e^- method
- MC (manual) - Double angle method
- MC (manual) - Jaquet-Blondel method
- MC (manual) - Σ method
- MC (Manual) - e-Σ method

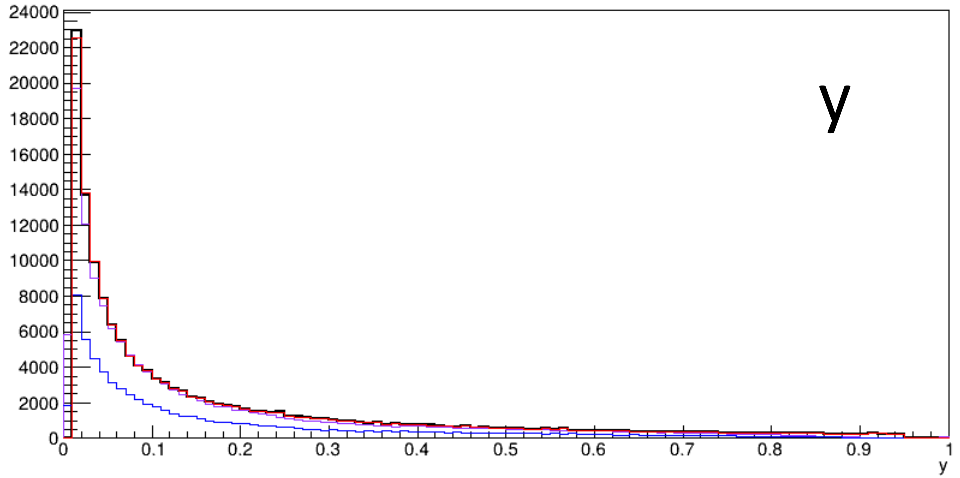
IncKin vs manual calc. - Electron method



IncKin vs manual calc. - JB method



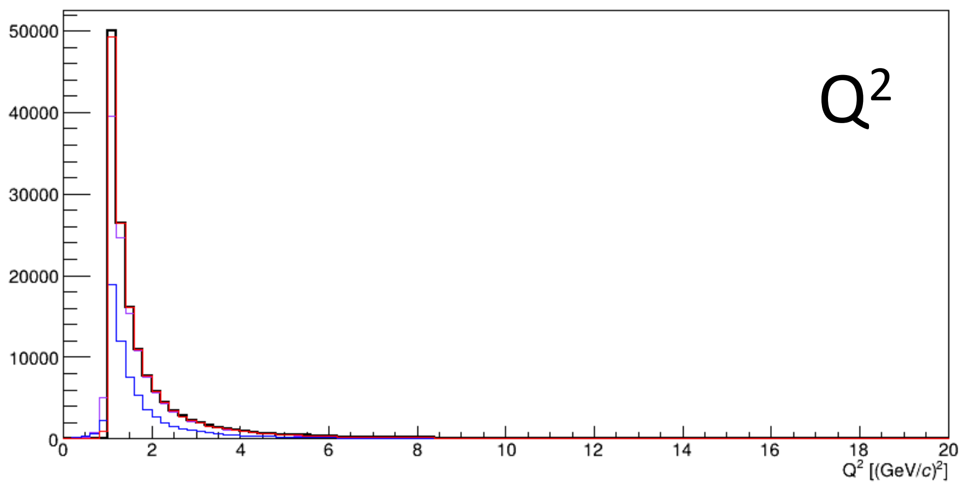


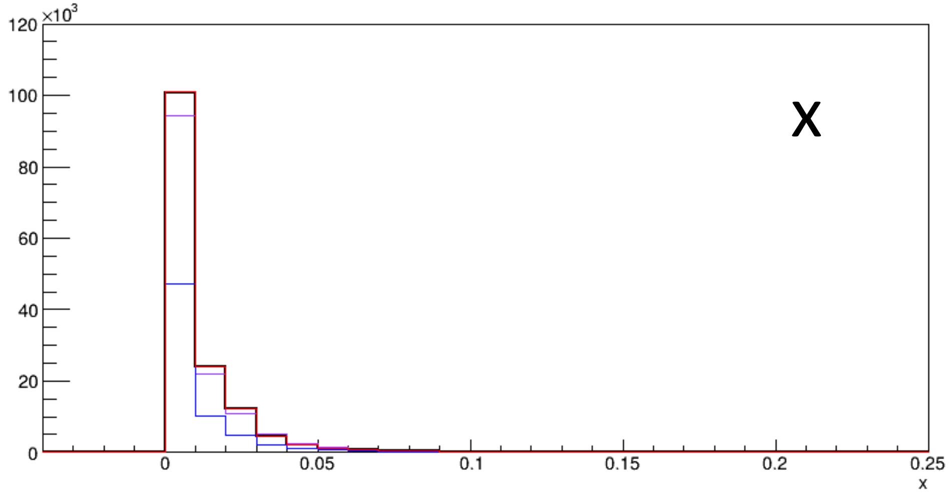


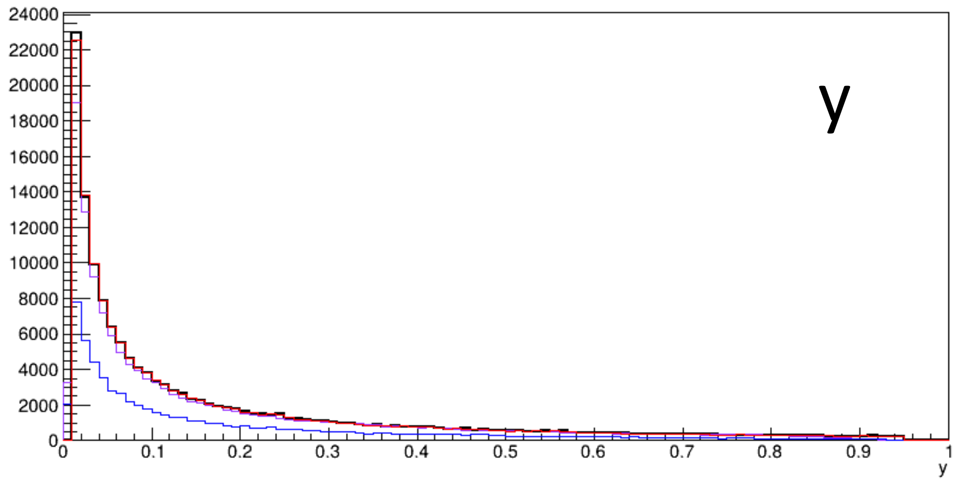
DVCS event kinematics - JB method

- InclusiveKinematicsTruth
- InclusiveKinematicsJB
- MC (manual calc.)
- Reco. (manual calc.)

IncKin vs manual calc. - DA method



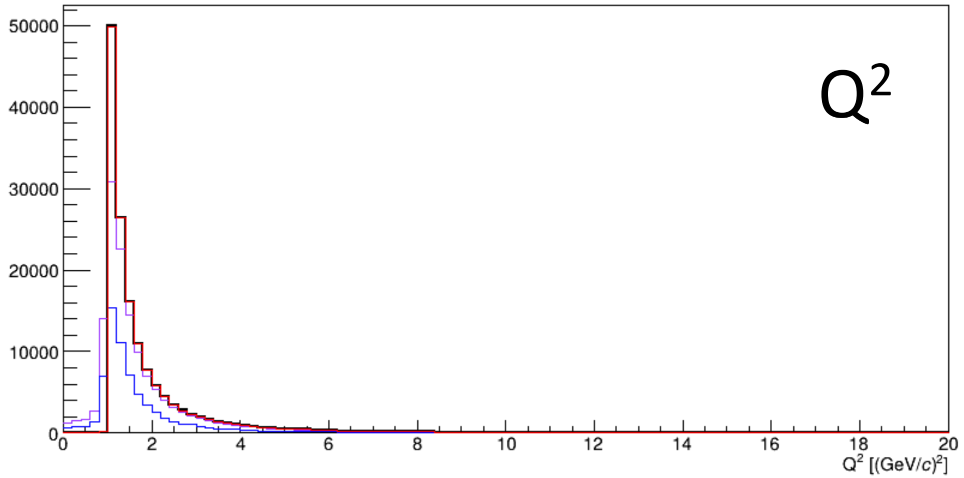


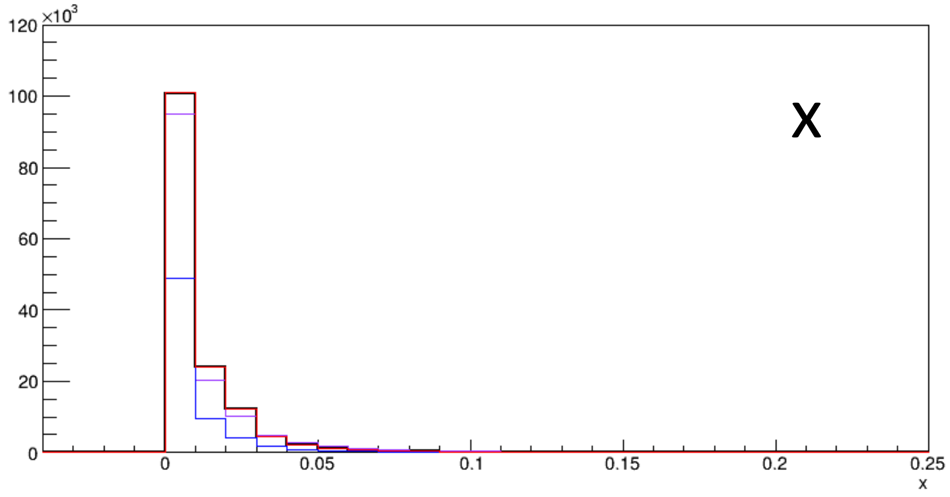


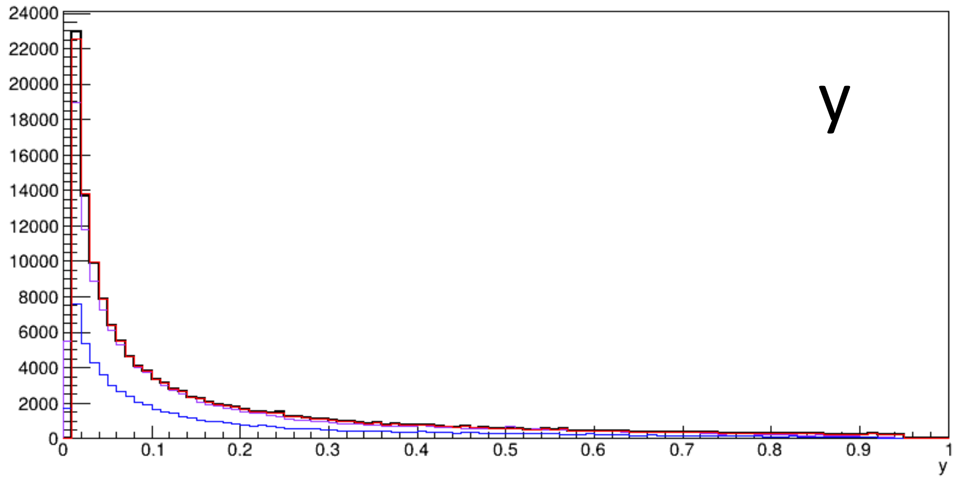
DVCS event kinematics - DA method

- InclusiveKinematicsTruth
- InclusiveKinematicsDA
- MC (manual calc.)
- Reco. (manual calc.)

IncKin vs manual calc. - Σ method







DVCS event kinematics - Σ method

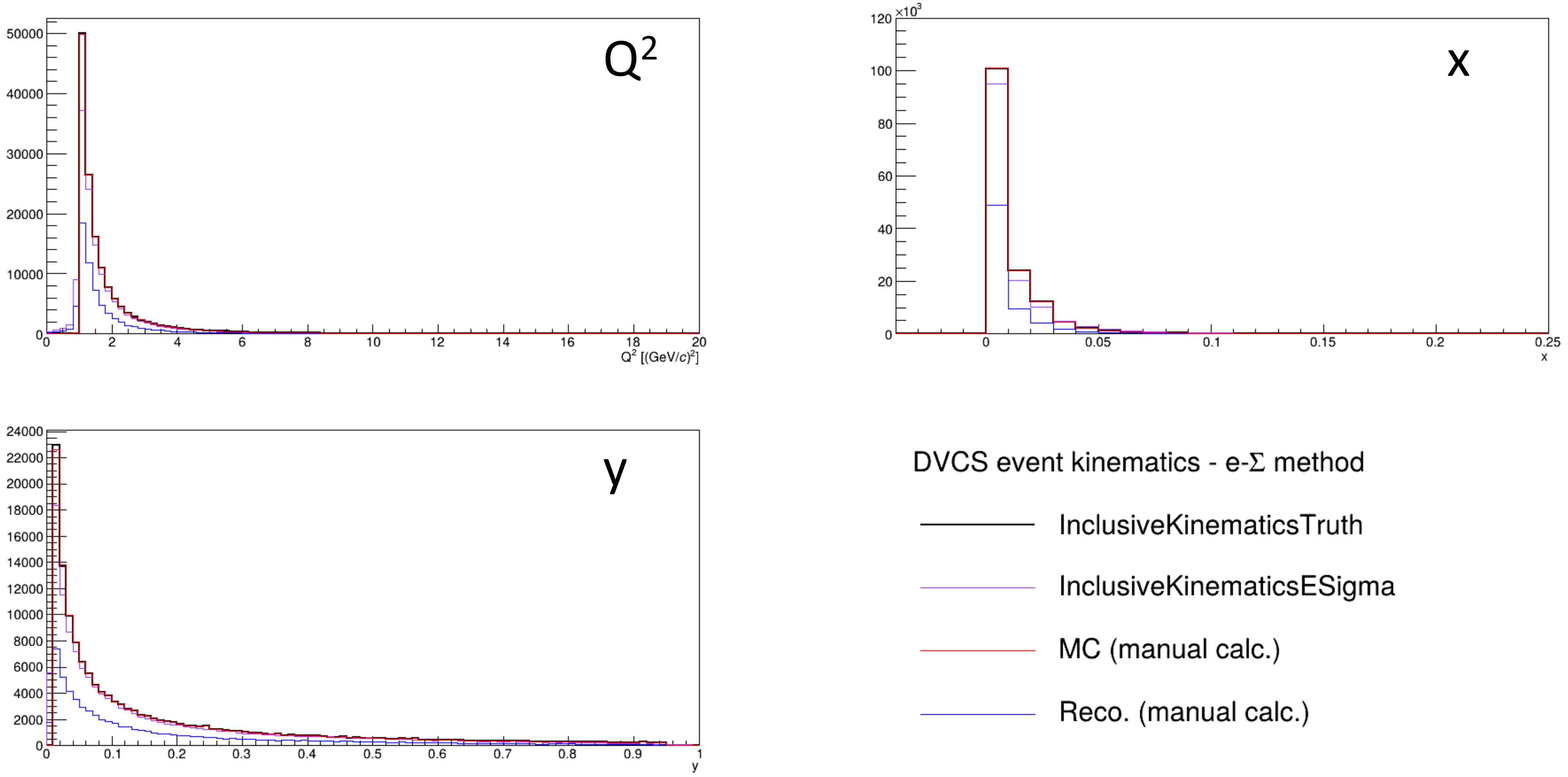
InclusiveKinematicsTruth

InclusiveKinematicsSigma

MC (manual calc.)

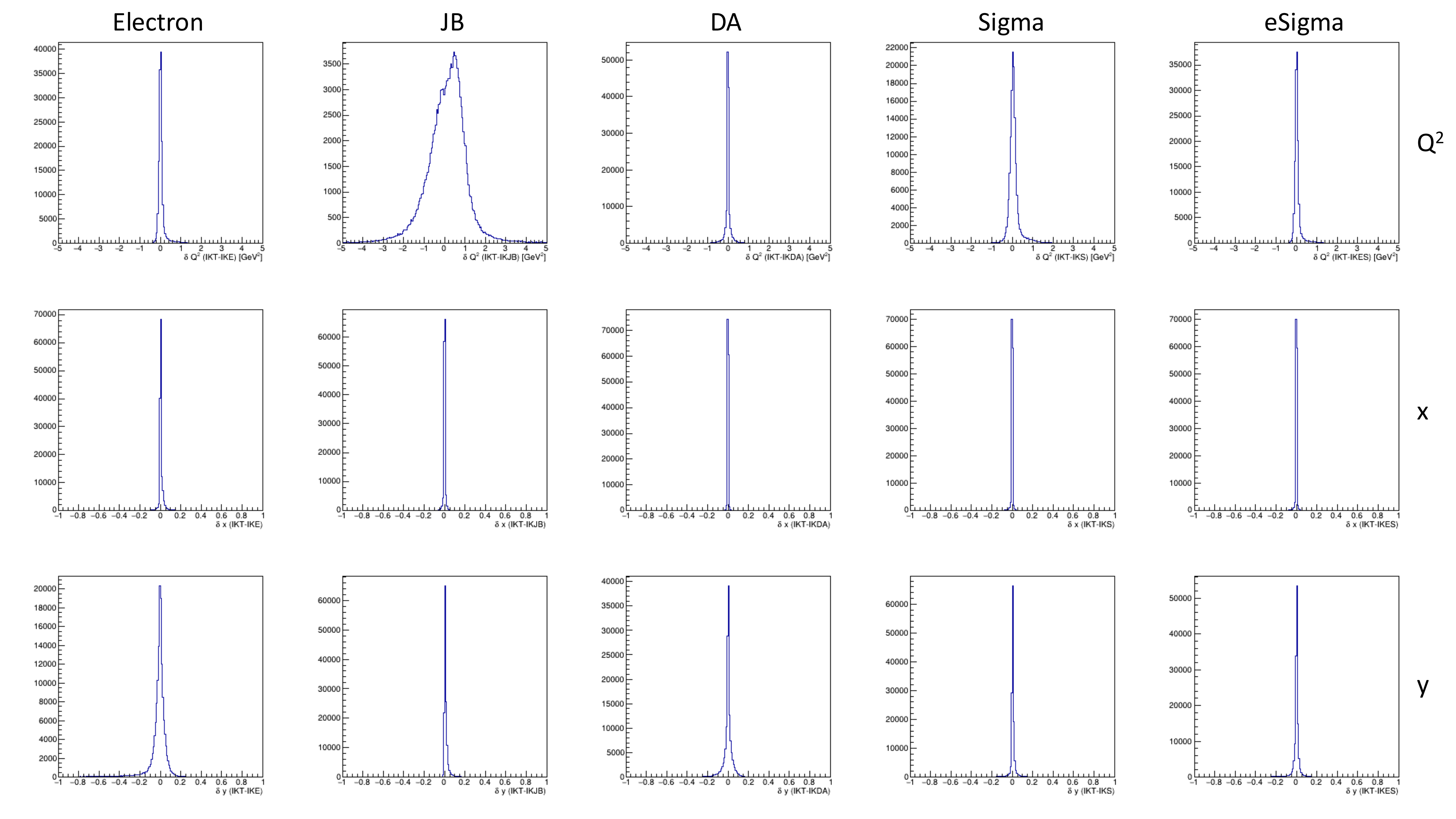
Reco. (manual calc.)

IncKin vs manual calc. - eΣ method

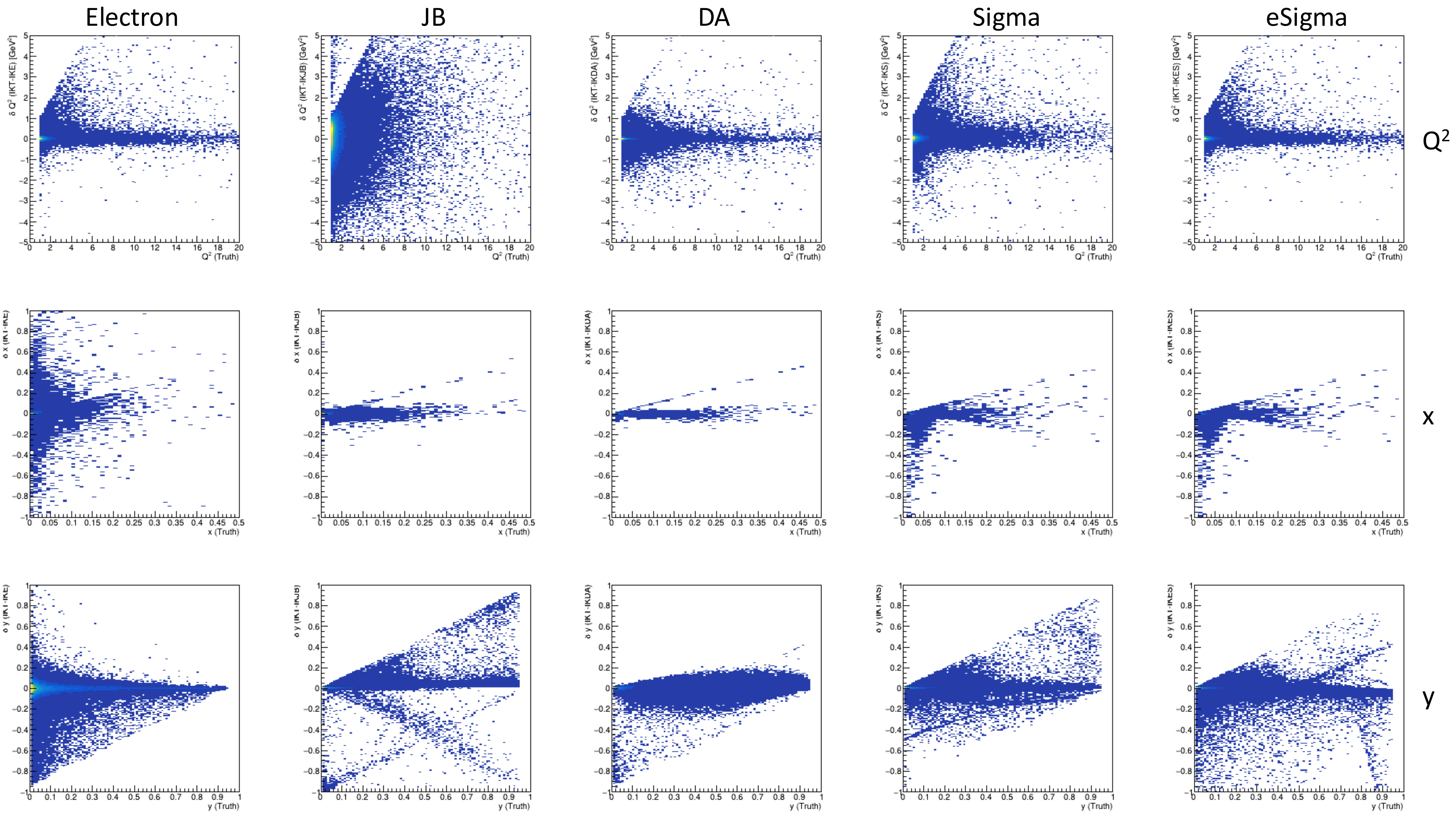


InclusiveKinematics testing

Results – differences between InclusiveKinematicsTruth and different InclusiveKinematicsXYZ branches

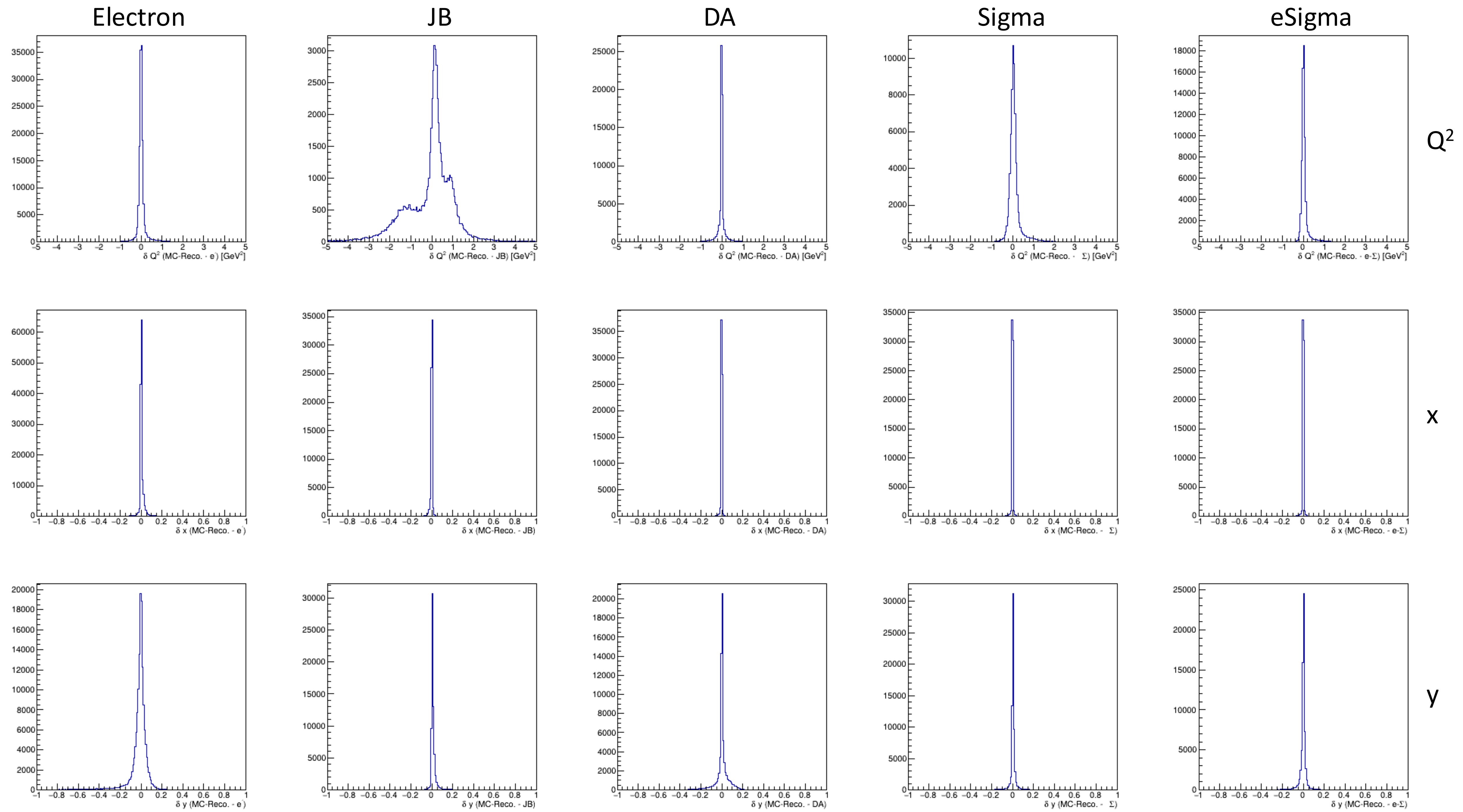


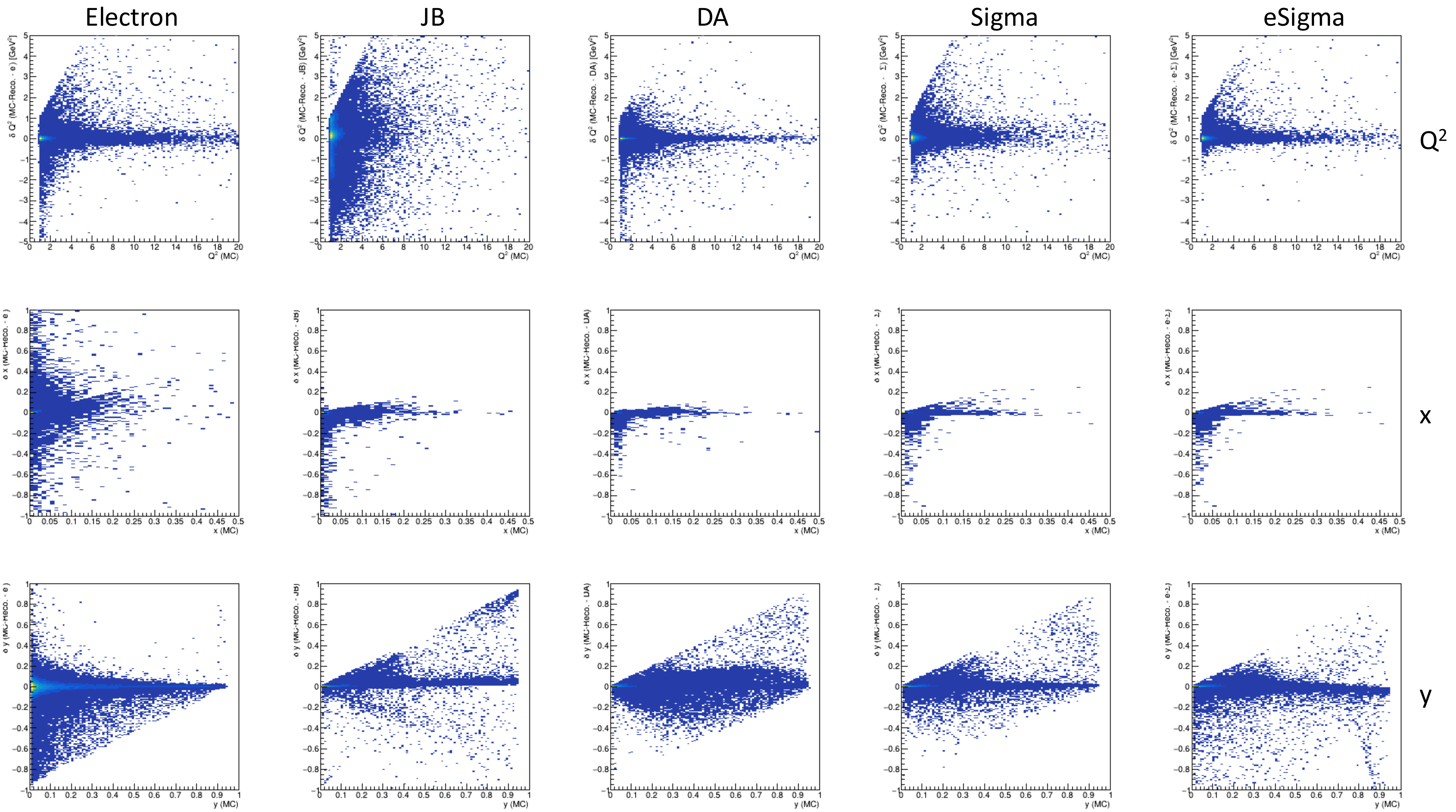
Electron
JB
DA
Sigma
eSigma
Q2
x
y
δ Q2 (IKT-IKE) [GeV2]
δ Q2 (IKT-IKJB) [GeV2]
δ Q2 (IKT-IKDA) [GeV2]
δ Q2 (IKT-IKS) [GeV2]
δ Q2 (IKT-IKES) [GeV2]
δ x (IKT-IKE)
δ x (IKT-IKJB)
δ x (IKT-IKDA)
δ x (IKT-IKS)
δ x (IKT-IKES)
δ y (IKT-IKE)
δ y (IKT-IKJB)
δ y (IKT-IKDA)
δ y (IKT-IKS)
δ y (IKT-IKES)


Electron
JB
DA
Sigma
eSigma
Q²
x
y
Q² (Truth)
x (Truth)
y (Truth)

InclusiveKinematics testing

Results – differences between manual calculations using MCParticles and ReconstructedParticles

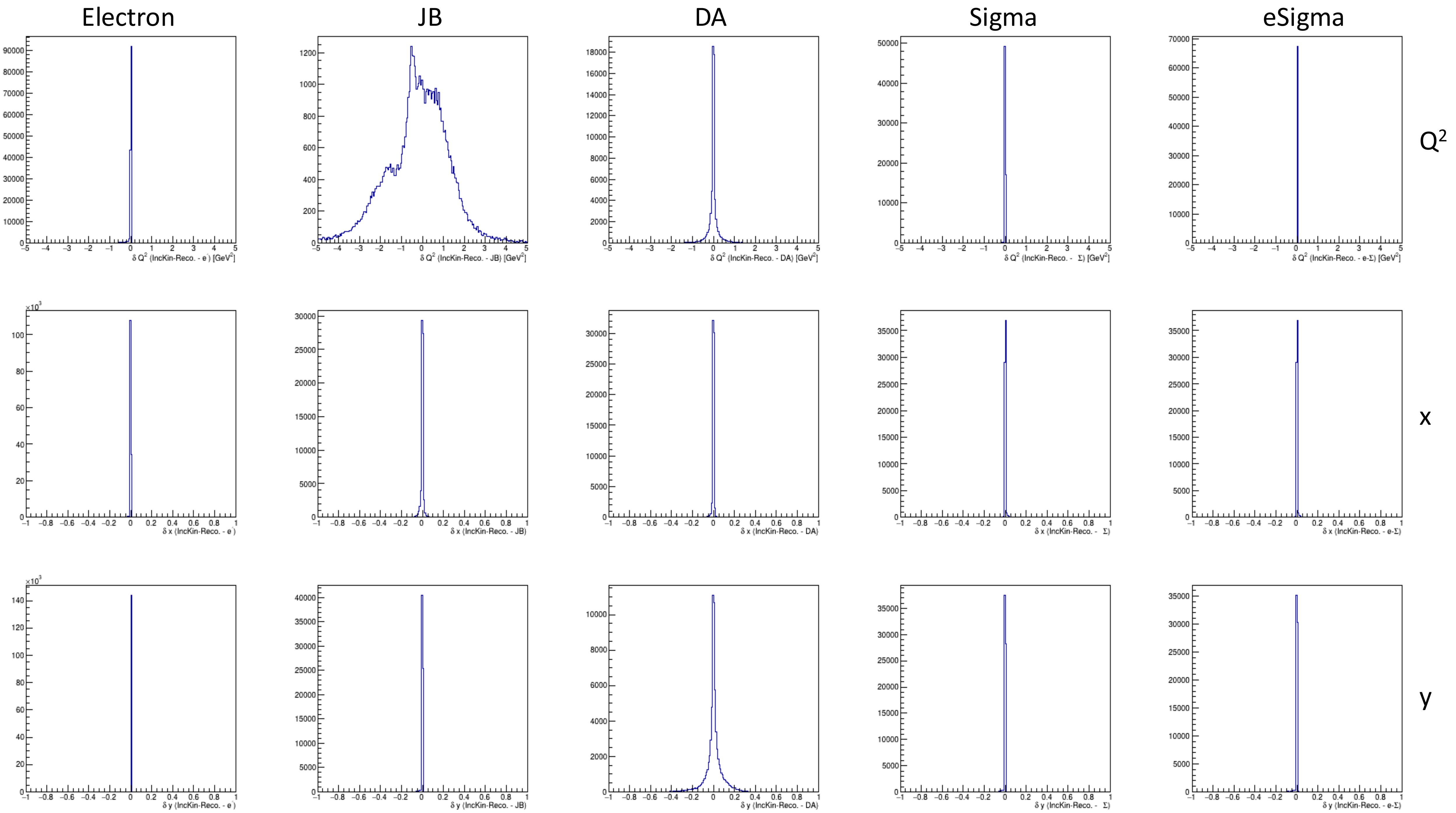

Electron
JB
DA
Sigma
eSigma
Q2
x
y
δ Q2 (MC-Reco. - e') [GeV2]
δ Q2 (MC-Reco. - JB) [GeV2]
δ Q2 (MC-Reco. - DA) [GeV2]
δ Q2 (MC-Reco. - Σ) [GeV2]
δ Q2 (MC-Reco. - e-Σ) [GeV2]
δ x (MC-Reco. - e')
δ x (MC-Reco. - JB)
δ x (MC-Reco. - DA)
δ x (MC-Reco. - Σ)
δ x (MC-Reco. - e-Σ)
δ y (MC-Reco. - e')
δ y (MC-Reco. - JB)
δ y (MC-Reco. - DA)
δ y (MC-Reco. - Σ)
δ y (MC-Reco. - e-Σ)

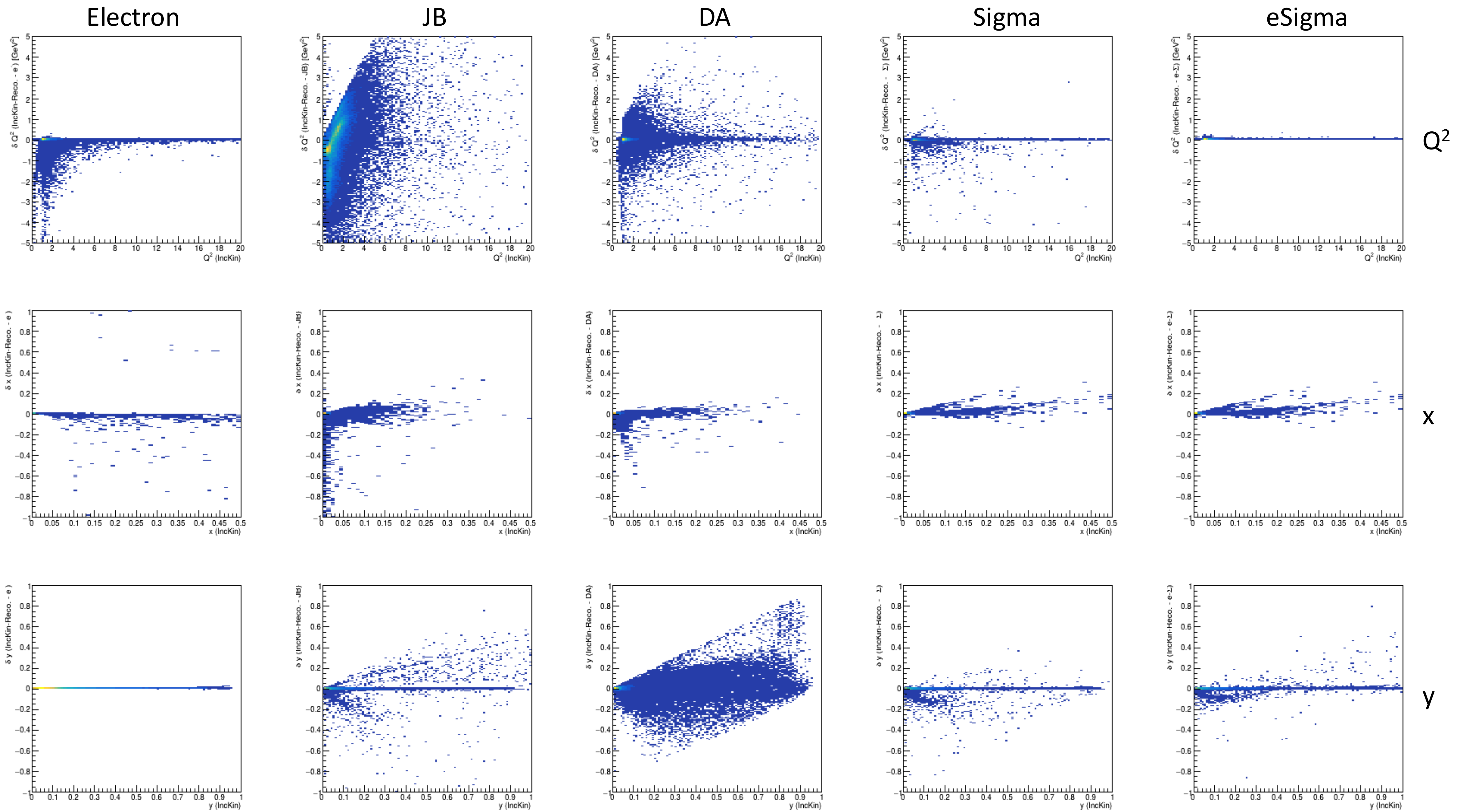


Electron
JB
DA
Sigma
eSigma
Q2
x
y
Q2 (MC)
x (MC)
y (MC)

InclusiveKinematics testing

Results – differences between InclusiveKinematicsXYZ branches and manual calculations using ReconstructedParticles


Electron
JB
DA
Sigma
eSigma
Q2
x
y
δ Q2 (IncKin-Reco. - e') [GeV2]
δ Q2 (IncKin-Reco. - JB) [GeV2]
δ Q2 (IncKin-Reco. - DA) [GeV2]
δ Q2 (IncKin-Reco. - Σ) [GeV2]
δ Q2 (IncKin-Reco. - e-Σ) [GeV2]
δ x (IncKin-Reco. - e')
δ x (IncKin-Reco. - JB)
δ x (IncKin-Reco. - DA)
δ x (IncKin-Reco. - Σ)
δ x (IncKin-Reco. - e-Σ)
δ y (IncKin-Reco. - e')
δ y (IncKin-Reco. - JB)
δ y (IncKin-Reco. - DA)
δ y (IncKin-Reco. - Σ)
δ y (IncKin-Reco. - e-Σ)



Electron
JB
DA
Sigma
eSigma
Q²
x
y
Q² (IncKin)
x (IncKin)
y (IncKin)

Brief aside: ePIC PID performance

ePIC PID test

- Using 100 files from the 25.01.1 simulation campaign to test the efficiency of ePIC PID.

	Generated	No. reconstructed		No. Missed (PID code 0)	
		#	%	#	%
Electrons	147847	84542	57.2	2568	1.74
Photons	147847	140847	95.3	0	0
Protons	147847	146265(*)	98.9(*)	1582(*)	1.07(*)

(*) - Proton PID comes from MC seeding, so should be 100%. "Missed" protons indicate a discrepancy with this process.

ePIC PID test

- The ePIC PID systems still need significant work to improve their electron reconstruction.
- Efficiency of 57% seen in 10x100 beam setting – from previous campaigns, other settings won't be much better.
  - 5x41: similar efficiency seen (using 24.10.1 campaign)
  - 18x275: significantly worse efficiency seen in 24.10.1 campaign.
    - For higher beam energy, missed electrons dominate mis-identified particles.

Trento Phi calculation

Trento Phi

- Have mentioned this in previous meetings.
  - Recommendation: use a full 4-vector calculation for Lorentz invariance of result.
- 2 papers found, from (some of) the same authors:
  - Single-spin asymmetries, the Trento conventions (Dec. 2004)[1]
  - Semi-inclusive deep inelastic scattering at small transverse momentum (Feb. 2007)[2]
- Give *slightly* different formulae for phi but look almost identical.

Phi_h (1)

- Paper 1 gives a formula for calculating phi using 3-vectors
  - Given in **target rest frame** (or any other frame reached by boosting from this along **q**.

$$\cos\phi_h = \frac{(\hat{\boldsymbol{q}} \times \boldsymbol{l})}{|\hat{\boldsymbol{q}} \times \boldsymbol{l}|} \cdot \frac{(\hat{\boldsymbol{q}} \times \boldsymbol{P}_h)}{|\hat{\boldsymbol{q}} \times \boldsymbol{P}_h|},$$

- Also gives a 4-vector method

$$\cos\phi_h = -\frac{g_\perp^{\mu\nu} l_\mu P_{h\nu}}{|l_\perp|\,|P_{h\perp}|},$$

$$|l_\perp| = \sqrt{-g_\perp^{\mu\nu} l_\mu l_\nu} \text{ and } |P_{h\perp}| = \sqrt{-g_\perp^{\mu\nu} P_{h\mu} P_{h\nu}}.$$

$$g_\perp^{\mu\nu} = g^{\mu\nu} - \frac{q^\mu P^\nu + P^\mu q^\nu}{P\cdot q\,(1+\gamma^2)} + \frac{\gamma^2}{1+\gamma^2}\left(\frac{q^\mu q^\nu}{Q^2} - \frac{P^\mu P^\nu}{M^2}\right),$$

$$\gamma = 2xM/Q,$$

Phi_h (2)

- Paper 2 gives the 4-vector calculation slightly differently:

$$\cos \phi_h = -\frac{l_\mu P_{h\nu}\, g_\perp^{\mu\nu}}{\sqrt{l_\perp^2\, P_{h\perp}^2}} ,$$

$$l_\perp^\mu = g_\perp^{\mu\nu} l_\nu \text{ and } P_{h\perp}^\mu = g_\perp^{\mu\nu} P_{h\nu}$$

Phi_h (3)

- Have tried both of these 4-vector methods, as well as the 3-vector calculation given.
- Testing using raw HEPMC data:
  - The 2x 4-vector methods do not agree, regardless of the boost applied.
  - When boosting to the target rest frame, the method in paper [2] agrees (at least to within 1 part in 10^4) with the 3-vector calculation.
    - The method in [1] is also close; ~1% off.
  - As soon as any other boost (or no boost/in lab frame) is applied, these methods diverge.
- (2 events shown on next 2 slides to demonstrate)

Phi$_h$ (4)

```
P3EVector k(0.0000000000000000e+00, 0.0000000000000000e+00, -9.9999999869440064e+00, 1.0000000000000000e+01);
P3EVector p(0.0000000000000000e+00, 0.0000000000000000e+00, 9.9995598131265865e+01, 1.0000000000000001e+02);
P3EVector kp(-8.8050940609571138e-01, -7.0570191241249569e-01, -8.4172495289531071e+00, 8.4925497877530507e+00);
P3EVector pp(-3.9743891263112607e-01, 1.0121128687525965e-01, 9.9634057254544004e+01, 9.9639319150554201e+01);
```

```
Target rest frame phi_h = -0.0171958 rad
[DEBUG] TPhi4 (no boost - HEPMC)        = [DEBUG] num = 0.0113858
[DEBUG] den (sq.) = 0.011659     den (mag) = -nan
[DEBUG] cPhi (sq.) = -0.976563   cPhi (mag) = -nan
[DEBUG] Phi (sq.) = 2.92466      Phi (mag) = -nan
2.92466 rad
[DEBUG] TPhi4 (Tgt rest frame - HEPMC)  = [DEBUG] num = -1.8048
[DEBUG] den (sq.) = 2.99658      den (mag) = 2.98172
[DEBUG] cPhi (sq.) = 0.602286    cPhi (mag) = 0.605287
[DEBUG] Phi (sq.) = 0.924435     Phi (mag) = 0.920669
0.924435 rad
[DEBUG] TPhi4 (epCoM - HEPMC)           = [DEBUG] num = -113113
[DEBUG] den (sq.) = 113116       den (mag) = 113116
[DEBUG] cPhi (sq.) = 0.999977    cPhi (mag) = 0.999977
[DEBUG] Phi (sq.) = 0.00683595   Phi (mag) = 0.00674819
0.00683595 rad
[DEBUG] Beam p; pre-CoM boost: (0,0,99.9956,100)
[DEBUG]          post-CoM boost: (0,0,-1.81899e-12,0.938272)
[DEBUG] TPhi3 (Tgt rest frame - HEPMC)  = 0.924375 rad
```

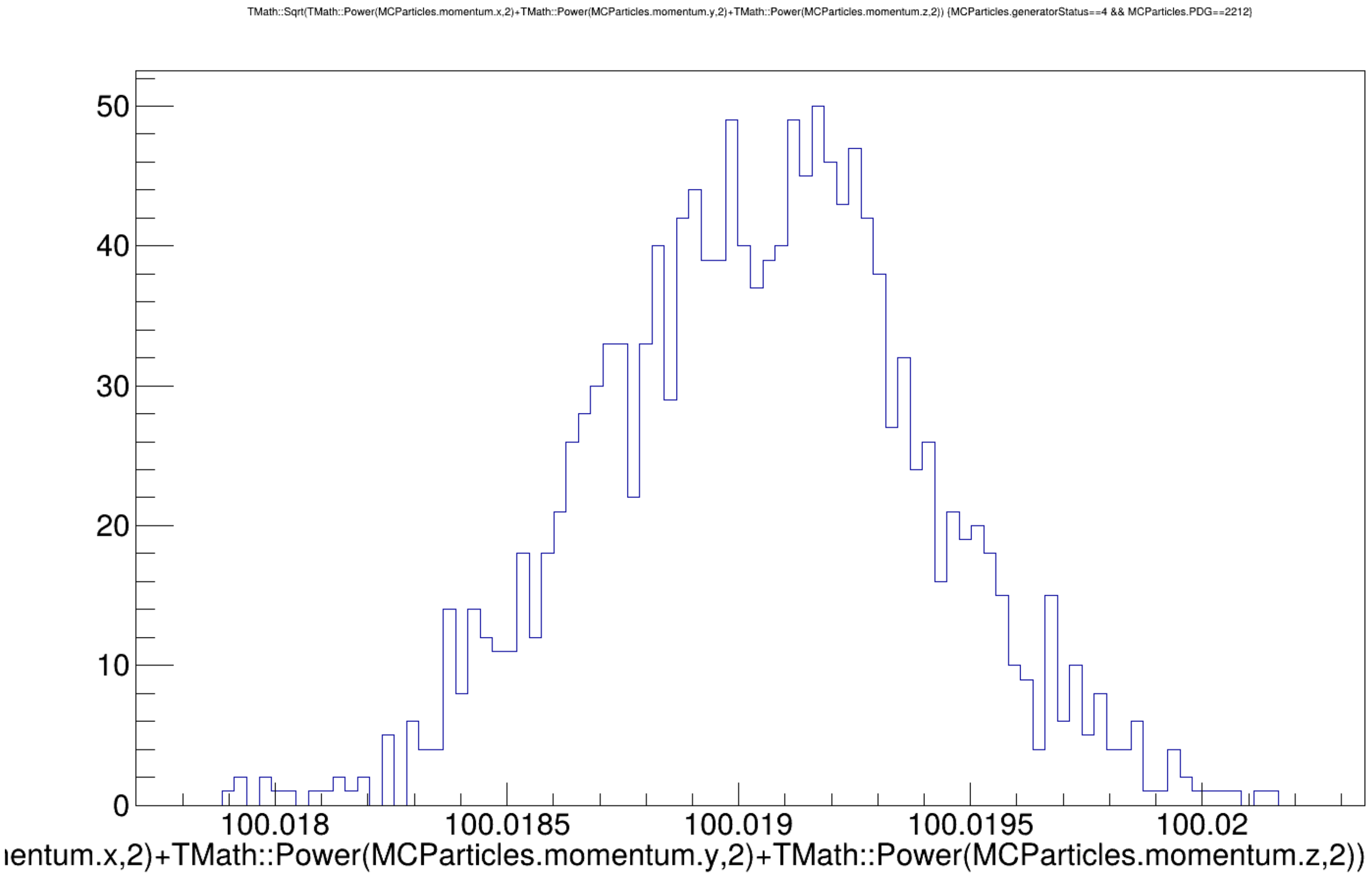
Phi_h (5)

```
P3EVector k(0.0000000000000000e+00,0.0000000000000000e+00,-9.9999999869440064e+00,1.0000000000000000e+01);
P3EVector p(0.0000000000000000e+00,0.0000000000000000e+00,9.9995598131265865e+01,1.0000000000000001e+02);
P3EVector kp(-5.2112621352190280e-01, 7.1975545871925573e-01, -7.8680413323163521e+00, 7.9180613232438688e+00);
P3EVector pp(-9.4227166382196192e-02, -3.9233430742232822e-01, 9.9912517523977399e+01, 9.9917737751197777e+01);
```

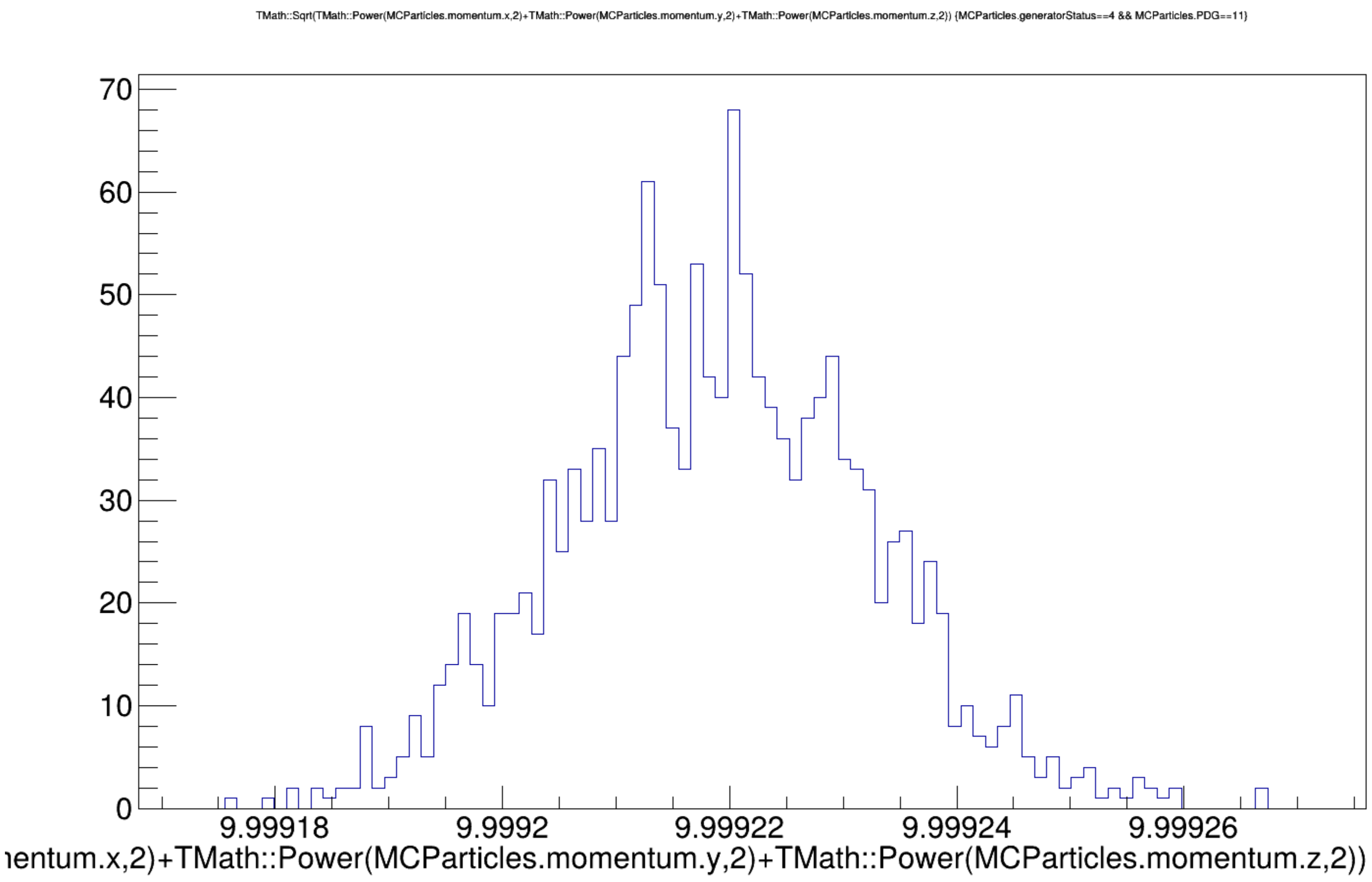
```
Target rest frame phi_h = -0.0171958 rad
[DEBUG] TPhi4 (no boost - HEPMC)        = [DEBUG] num = 0.00390626
[DEBUG] den (sq.) = 0.00394553   den (mag) = 0.124335
[DEBUG] cPhi (sq.) = -0.990046   cPhi (mag) = -0.0314173
[DEBUG] Phi (sq.) = 3.00038      Phi (mag) = 1.60222
3.00038 rad
[DEBUG] TPhi4 (Tgt rest frame - HEPMC)  = [DEBUG] num = 1.10659
[DEBUG] den (sq.) = 1.70124      den (mag) = 1.69239
[DEBUG] cPhi (sq.) = -0.650458   cPhi (mag) = -0.653861
[DEBUG] Phi (sq.) = 2.27898      Phi (mag) = 2.28347
2.27898 rad
[DEBUG] TPhi4 (epCoM - HEPMC)           = [DEBUG] num = -95149.5
[DEBUG] den (sq.) = 95189.2      den (mag) = 95189.2
[DEBUG] cPhi (sq.) = 0.999583    cPhi (mag) = 0.999582
[DEBUG] Phi (sq.) = 0.0288922    Phi (mag) = 0.0289005
0.0288922 rad
[DEBUG] Beam p; pre-CoM boost: (0,0,99.9956,100)
[DEBUG]         post-CoM boost: (0,0,-1.81899e-12,0.938272)
[DEBUG] TPhi3 (Tgt rest frame - HEPMC)  = -2.27889 rad
```

Backup

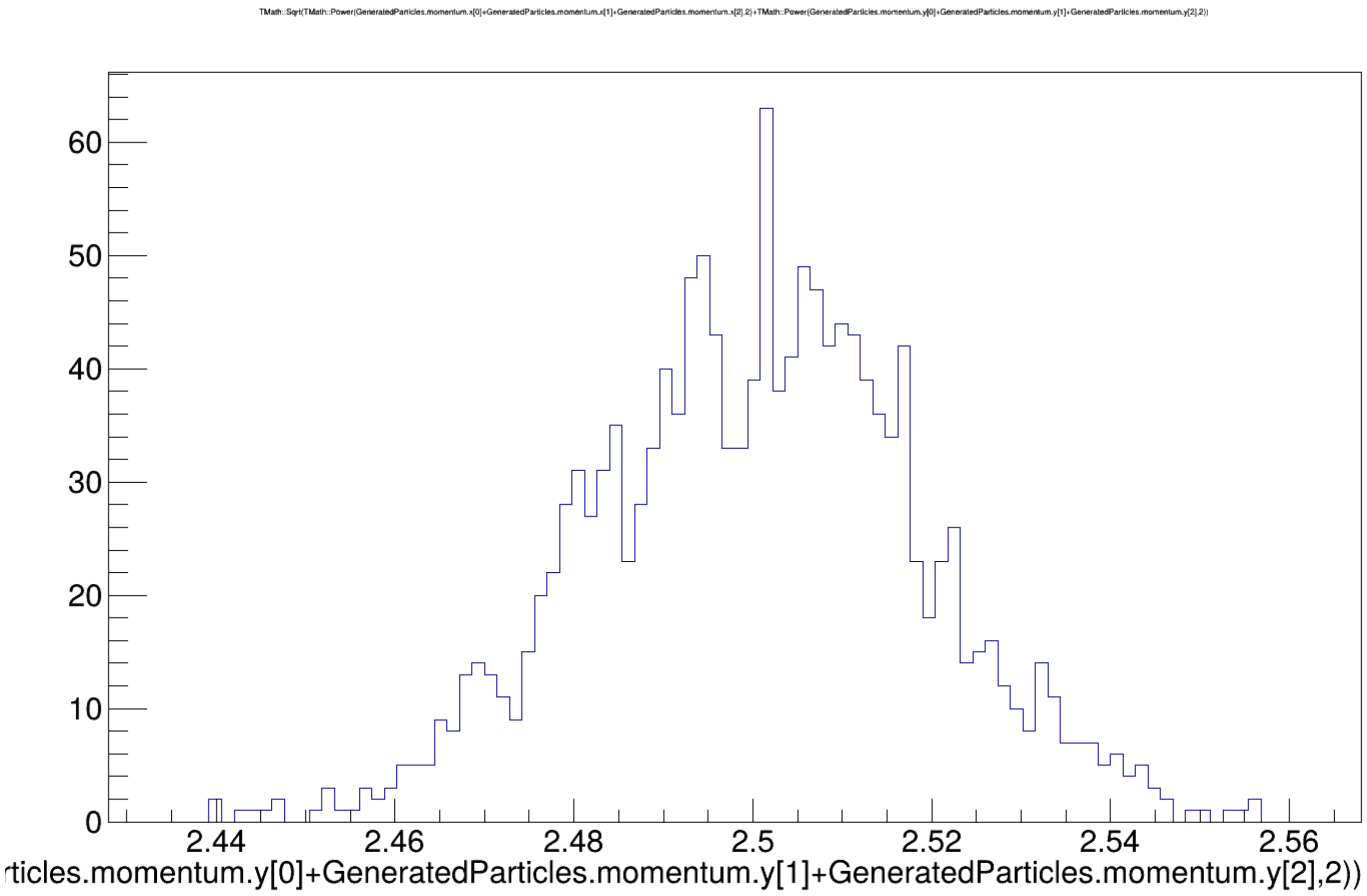
MCParticles.P() for beam p



MCParticles.P() for beam e



GeneratedParticles.Pt() sum



GeneratedParticles.P() sum