TCS Update

Exclusive Diffractive and Tagging WG Meeting

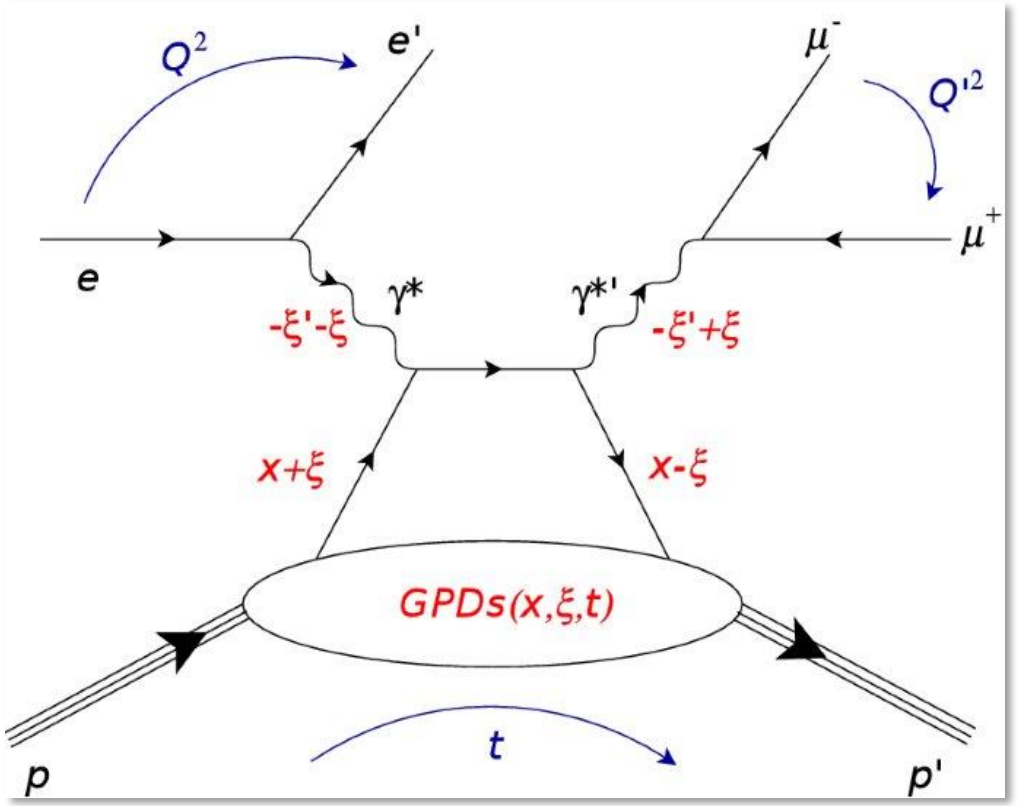
Gary Penman*
University of Glasgow
23rd February 2026

*gary.penman@glasgow.ac.uk

Introduction

Overview

- December preTDR 18x275
- February Campaign Plots
- May Early Science 10x130
- RAD->Combi RAD (combinatorial + background analysis)
- Amplitude Fitting Work – Physics extraction



Quasi-real (Q^2->0) l^+l^- photoproduction: Timelike Compton Scattering.

Recap December preTDR 18x275

Produced a first version of analysis macros, overleaf note and plots for Dec 1st deadline.

https://github.com/Garypenman/TCS-ePIC-preTDR/releases/tag/v1.0.0

Relevant Analysis repo(s) now tagged on github as "preTDR" version – reproducibility as code evolves.

https://github.com/Garypenman/rad/releases/tag/v1.0.0

https://github.com/Garypenman/epic-rad/releases/tag/v1.0.0

https://github.com/Garypenman/DDVCS-Analysis/releases/tag/v1.0.0

Code can be used and built upon to move into early science analysis.

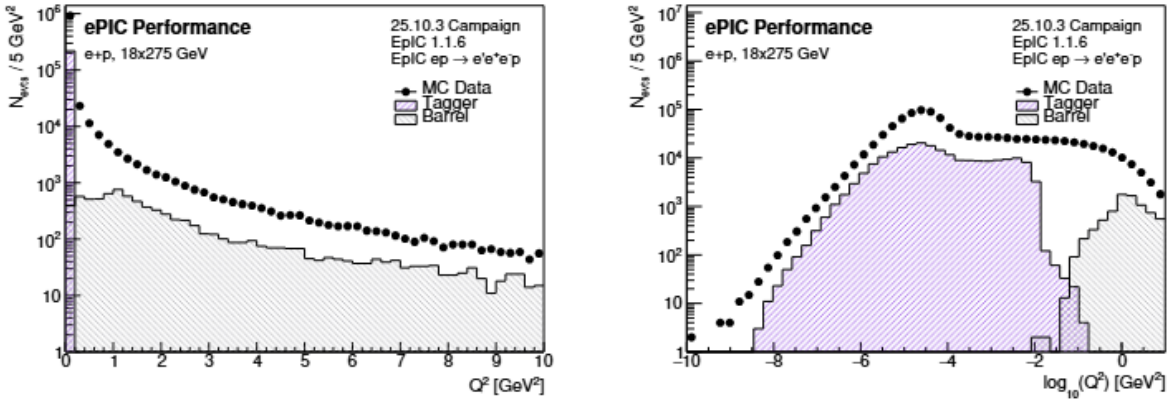


Figure 4.37: Number of generated events in (left) Q^2 and (right) $\log_{10}(Q^2)$ plotted with reconstructed values out of the tagger and central detector. Scale is not normalized to any luminosity. Analysis code used to analyze the reconstructed data can be found in this repository and scripts used to make this plot can be found in this repository.

Some minor typos in axis labels – now fixed.

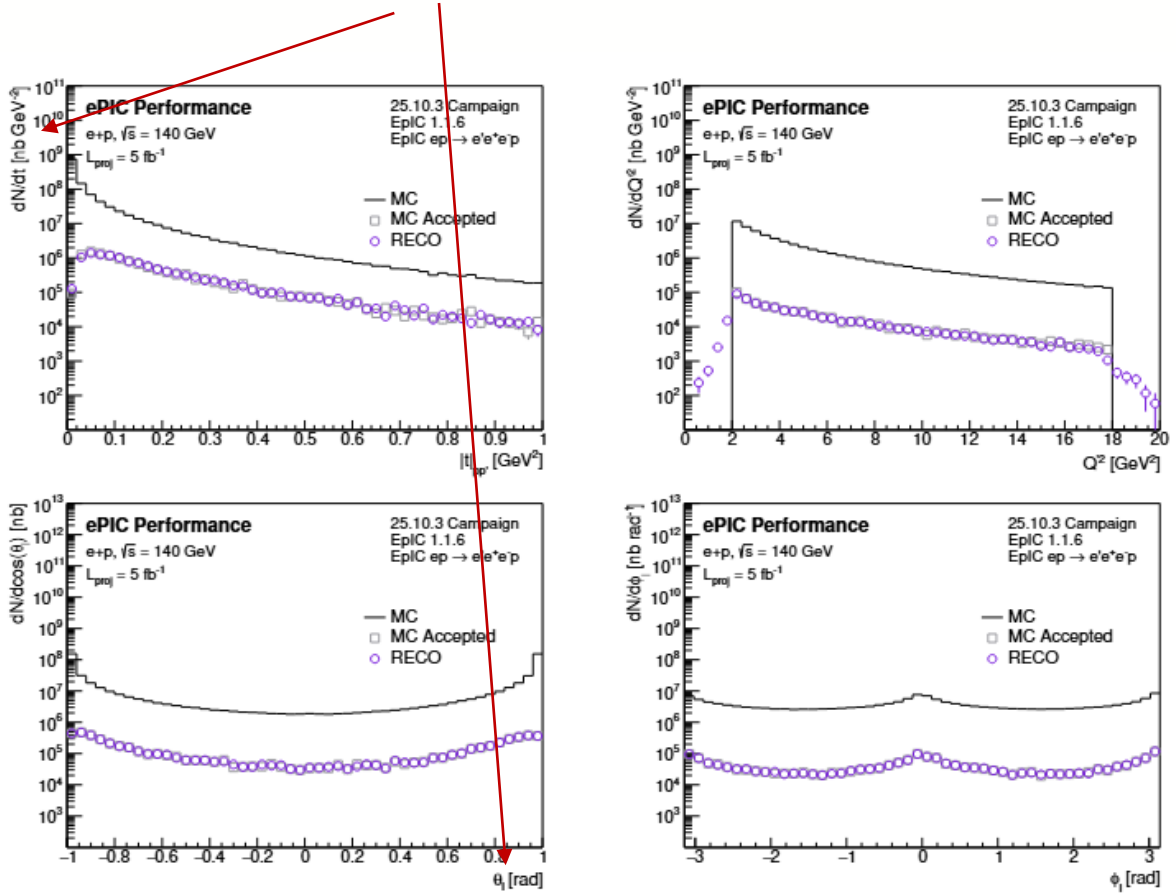
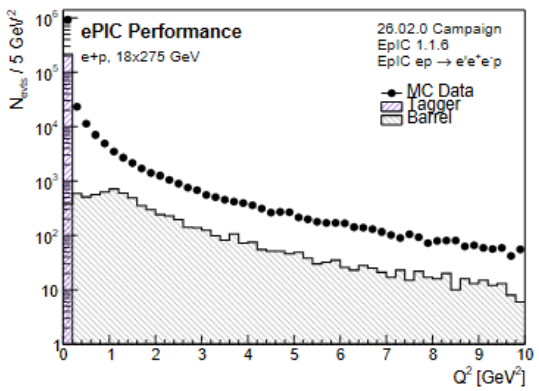
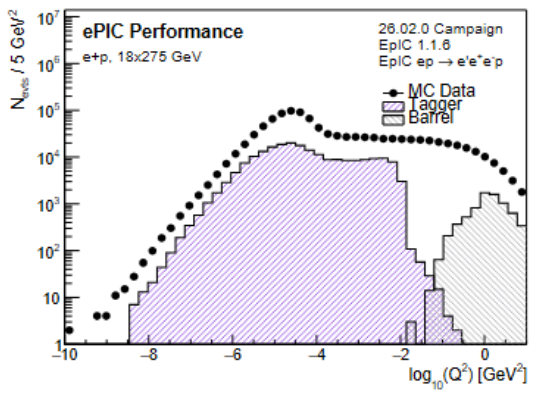
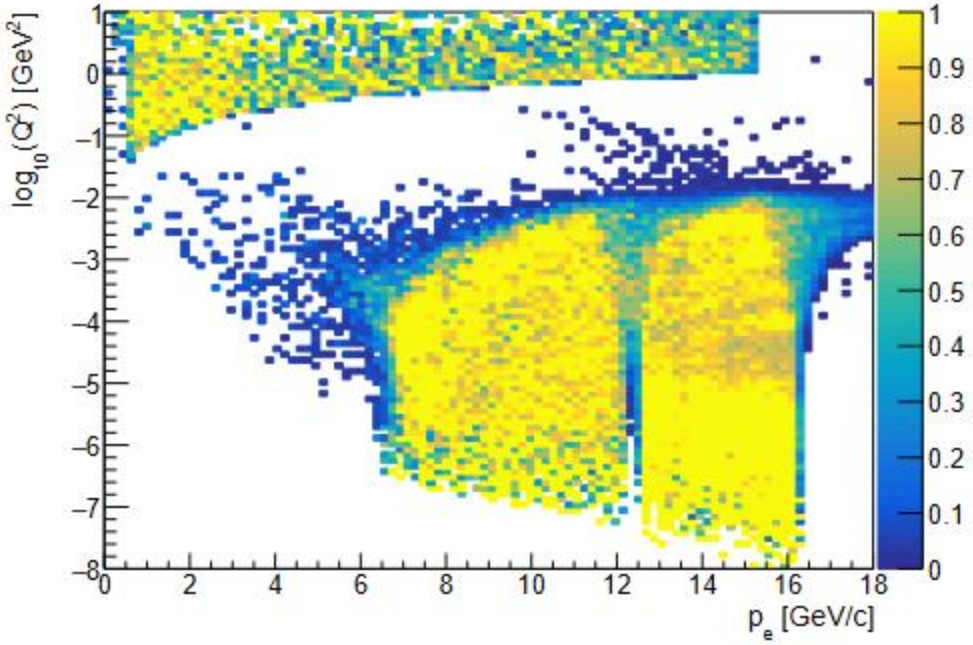


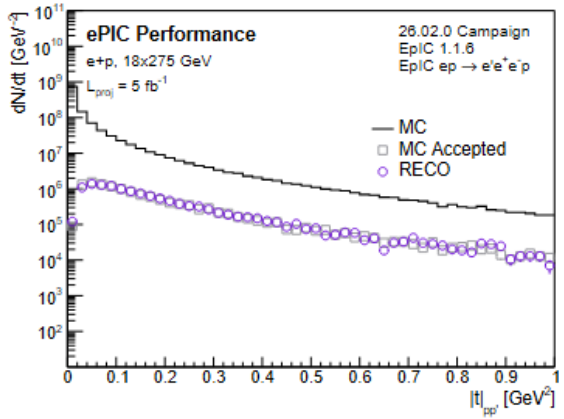
Figure 4.38: Projected rates of the full exclusive reconstruction of the observables which enter the $d^4\left(\sigma_{BH}+\sigma_{TCS}+\sigma_{INT}\right)$ differential cross section at an integrated luminosity of 5 fb^{-1}. (Top Left) $|t|$, (Top Right) Q'^2, (Bottom Left) θ_l and (Top Right) ϕ_l, the polar and azimuthal decay angles in the TCS photon rest frame, respectively. Analysis code used to analyze the reconstructed data can be found in this repository, and scripts used to make this plot can be found in this repository.

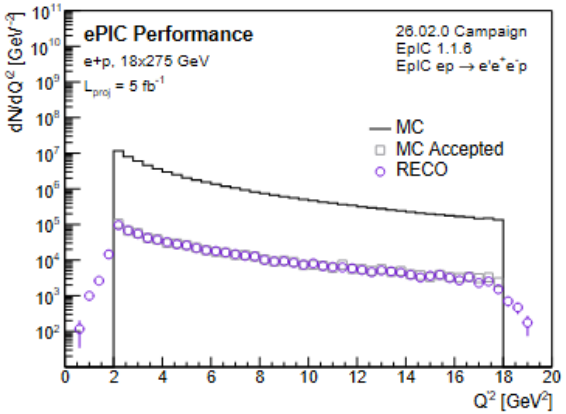
February Campaign

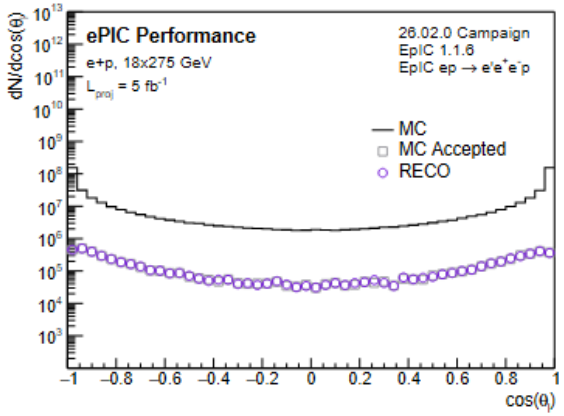


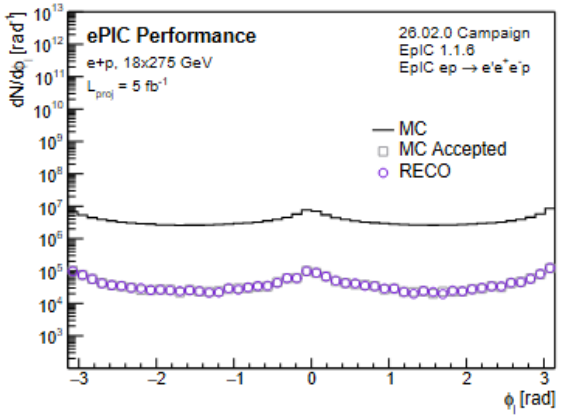












25.10.3

```
Tagger Acceptance: 0.239951
Photon Acceptance: 0.206065
Proton Acceptance: 0.223837
Folded Acceptance: 0.0110677
Physics Cut Acceptance: 0.916363
Events passing final cuts: 10573
Scaled up by xsec*Lum/Ngen: 247408
```

26.02.0

```
Tagger Acceptance: 0.23597
Photon Acceptance: 0.21913
Proton Acceptance: 0.224096
Folded Acceptance: 0.0115876
Physics Cut Acceptance: 0.92135
Events passing final cuts: 11082
Scaled up by xsec*Lum/Ngen: 259319
```

May Early Science 10x130

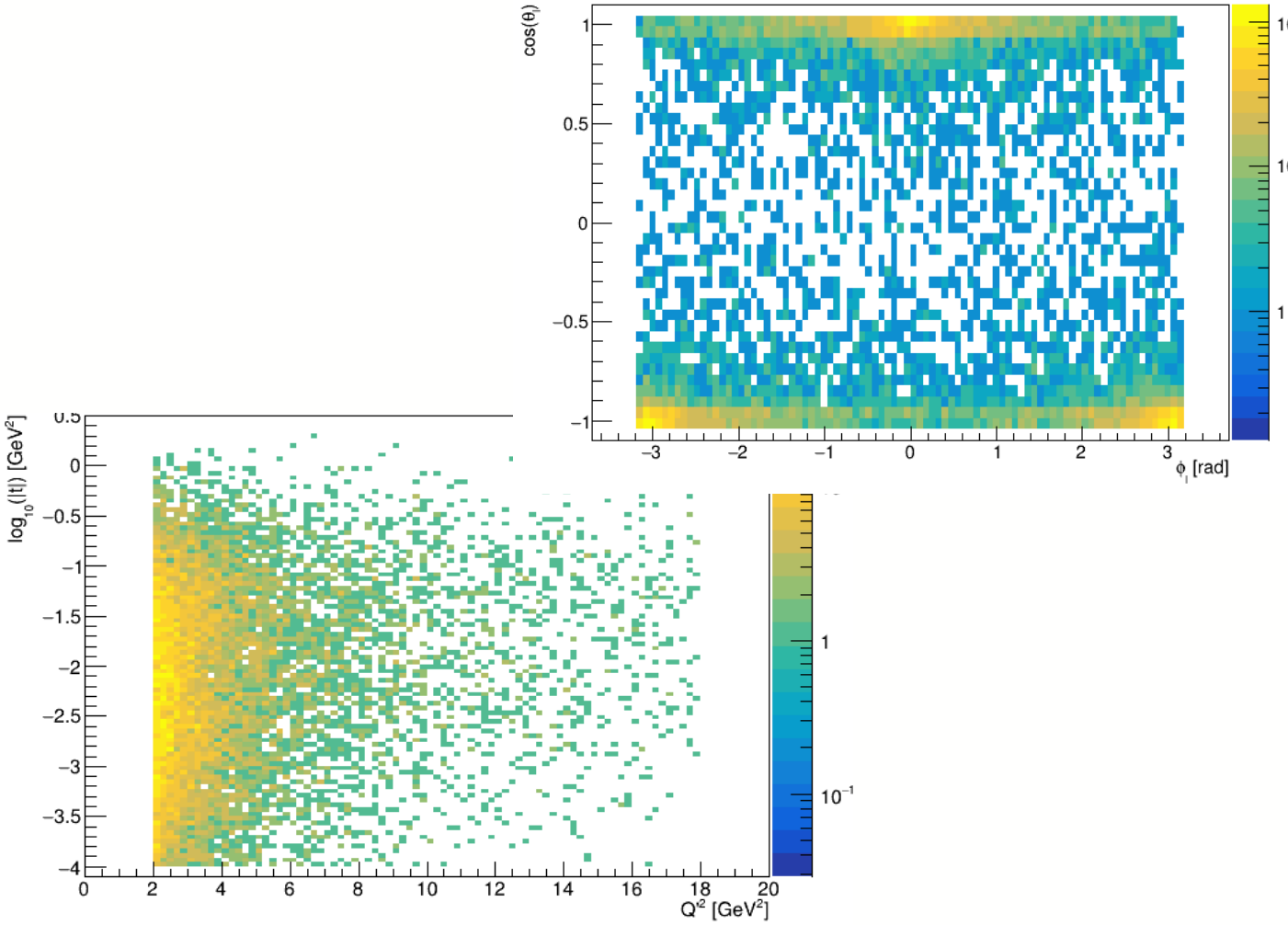
Generated 1M h+, 1M h- 10x130 events with EpIC

Steering cards and metadata has been updated to DDVCSdataset github.
https://github.com/Garypenman/rad/releases/tag/v1.0.0
Production request has been made.

Plan to continue building out analysis code on 18x275 data, should run out the box on march campaign 10x130 data.

Sigma 18x275 = ~ 6 nb
Sigma 10x130 = ~ 3 nb
Across the same phase space region

preTDR and early science projections at 5fb-1
-> half the cross section -> sqrt(2) impact on statistical uncertainties



10x130 4D phase space (EpIC generator)

RAD -> Combi RAD

ProcessHepMCTCSCombi.C - emacs@np-psi.physics.gla.ac.uk

File Edit Options Buffers Tools C++ Help

Save Undo

```
#include "CommonDefines.h"
#include "HepMCElectro.h"
#include "KinematicsProcElectro.h"
#include "KineCalculation.h"
#include "Indicing.h"
#include "ElectronScatterKinematics.h"
#include <TBenchmark.h>

/**
 * @brief Example Script: TCS Analysis with Combinatorics and Missing Mass.
 * Updated to use SetMesonParticles shortcut and CloneLinked.
 */
void ProcessHepMCTCSCombi(){

  using namespace rad;
  using namespace rad::consts::data_type;

  gBenchmark->Start("df");

  // ============================================================================
  // 1. SETUP & INJECTION
  // ============================================================================
  HepMCElectro hepmc{
      "hepmc3_tree",
      "/w/work5/home/garyp/eic/Farm/data/EpIC_DDVCS_ee_18x275/rootfiles/18x275_ddvcs_edecay_hplus.root"
  };
  hepmc.SetupMC();

  // ============================================================================
  // 2. PARTICLE DEFINITIONS
  // ============================================================================
  hepmc.SetBeamIonIndex(3);
  hepmc.SetBeamElectronIndex(0);
  hepmc.SetScatElectronCandidates({1, 6});
  hepmc.SetParticleCandidates("ele", {1, 6});
  hepmc.SetParticleIndex("pos", 7);
  hepmc.SetParticleIndex("pprime", 5);

  hepmc.MakeCombinations();

  // ============================================================================
  // 3. KINEMATICS PROCESSOR (Standard Topology)
  // ============================================================================
  KinematicsProcElectro kine{&hepmc, MC()};

  // A. Define Particles FIRST (Topology Construction)
  kine.Creator().Sum("gprime", {{"ele", "pos"}});

  // Missing Mass: n_calc = (Beam + Target) - (ScatEle + Jpsi + pi+)
  kine.Creator().Diff("pprime_calc",{
        {consts::BeamIon(), consts::BeamEle()},
        {"gprime", consts::ScatEle()}
```

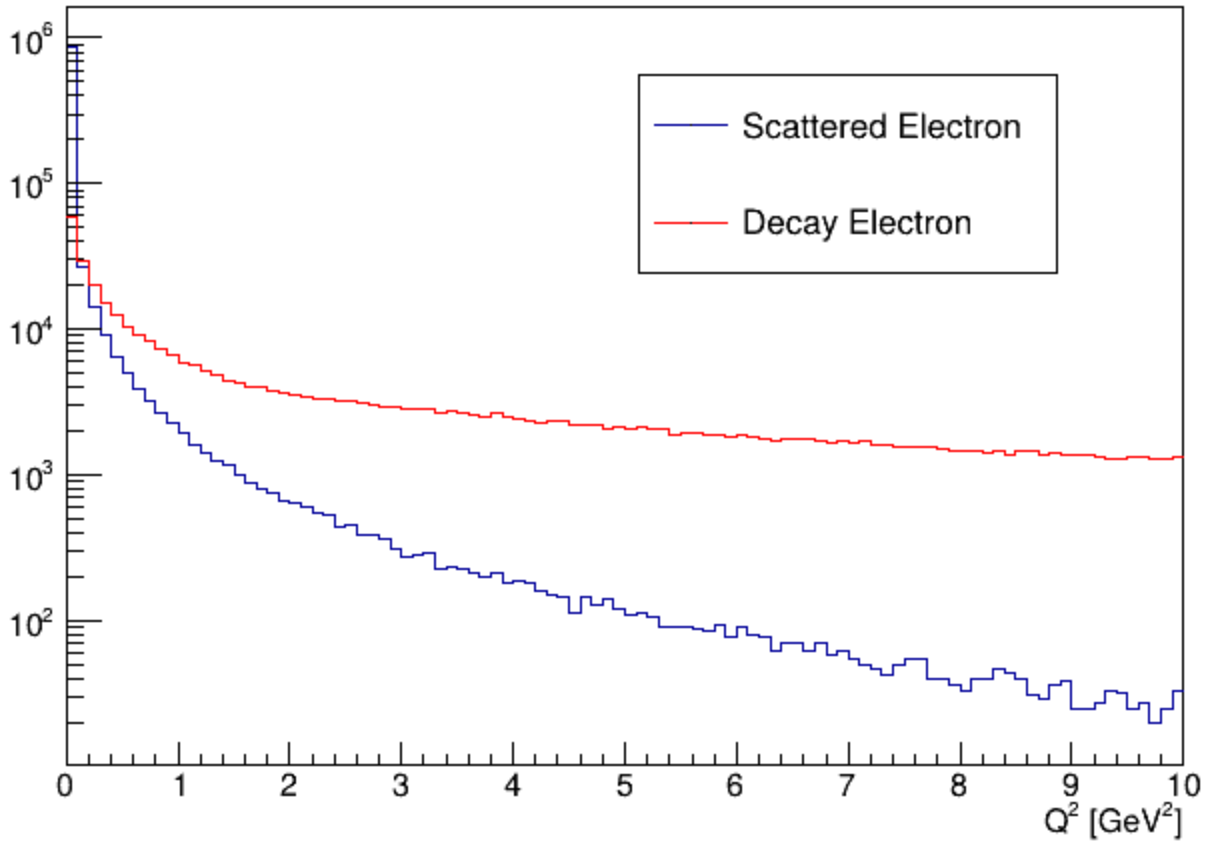
-:--- ProcessHepMCTCSCombi.C Top L51 (C++/l Abbrev)

Beginning of buffer

Example new TCS HepMC analysis macro for combi rad framework

Large rewrite of base RAD code over winter in order to facilitate "combinatorial analysis".

A (basic) example: two electrons are reconstructed in the final state of an event -> there are now two choices of scattered electron and therefore Q2.



TCS Combi-RAD Setup (Example)

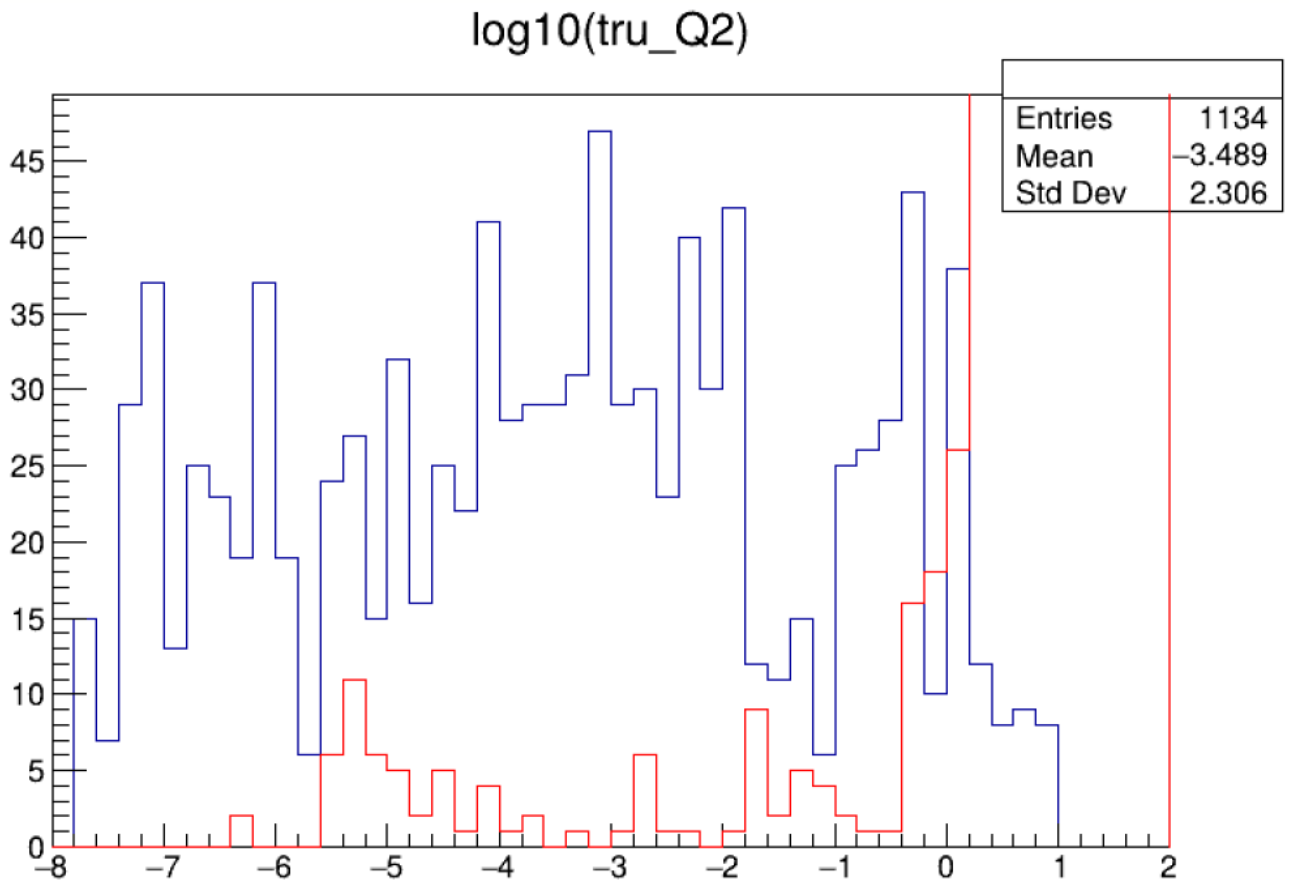
```
root [6] events->Scan("MCParticles.PDG:MCParticles.momentum.z:MCParticles.generatorStatus")
************************************************************
*    Row   * Instance * MCParticl * MCParticl * MCParticl *
************************************************************
*        0 *        0 *       11 * -17.99854 *        4 *
*        0 *        1 *       11 * -17.63491 *        1 *
*        0 *        2 *       22 * -0.363626 *       13 *
*        0 *        3 *     2212 * 274.97614 *        4 *
*        0 *        4 *       22 * 1.0722021 *        3 *
*        0 *        5 *     2212 * 273.54031 *        1 *
*        0 *        6 *       11 * 1.4346587 *        1 *
*        0 *        7 *      -11 * -0.362456 *        1 *
*        0 *        8 *       11 * 0.0014734 *        0 *
*        0 *        9 *       22 * 0.0418103 *        0 *
*        0 *       10 *       22 * 0.2465751 *        0 *
*        0 *       11 *       22 * 0.0260029 *        0 *
*        0 *       12 *       22 * 0.1950669 *        0 *
*        0 *       13 *       11 * 0.0042578 *        0 *
*        0 *       14 *       11 * 0.0266722 *        0 *
*        0 *       15 *       22 * 0.0160718 *        0 *
*        0 *       16 *       22 * 0.0497269 *        0 *
*        0 *       17 *       22 * 0.0078861 *        0 *
*        0 *       18 *       22 * 0.0825953 *        0 *
*        0 *       19 *       22 * 0.0038061 *        0 *
*        0 *       20 *       22 * 0.0052239 *        0 *
*        0 *       21 *       11 * 0.0024149 *        0 *
*        0 *       22 *       22 * 0.0044093 *        0 *
*        0 *       23 *       22 * 0.0032838 *        0 *
*        0 *       24 *       22 * 0.1450339 *        0 *
```

```
rad_df.SetBeamsFromMC(0, 3);
```

```
const int Role_ScatEle  = 1,
const int Role_Recoil   = 5,
const int Role_DecayEle = 6,
const int Role_DecayPos = 7,
const int lep_PDG = 11)
```

```
rad_df.SetParticleCandidates(consts::ScatEle(), Role_ScatEle, rad::index::FilterIndices(11), {"rec_true_pid"});
rad_df.SetParticleCandidates("ele", Role_DecayEle, rad::index::FilterIndices(11),  {"rec_true_pid"});
rad_df.SetParticleCandidates("pos", Role_DecayPos, rad::index::FilterIndices(-11), {"rec_true_pid"});
rad_df.SetParticleCandidates("pprime", Role_Recoil, rad::index::FilterIndices(2212), {"rec_true_pid"});
```

Fresh Combi-RAD Output



Using 26.02.0 campaign files to now build out combi-rad software.

Some "truth" info is being output incorrectly -> possible internal indexing bug. Investigating.

"Losing" events, see 4k sample -> 1.1k output.

Clearly, still fixing teething issues with new framework. Soon this will be a powerful tool for folding in background merged and DIS background event samples to the main physics events, and performing unbiased cut based analysis as well as fits and beyond.

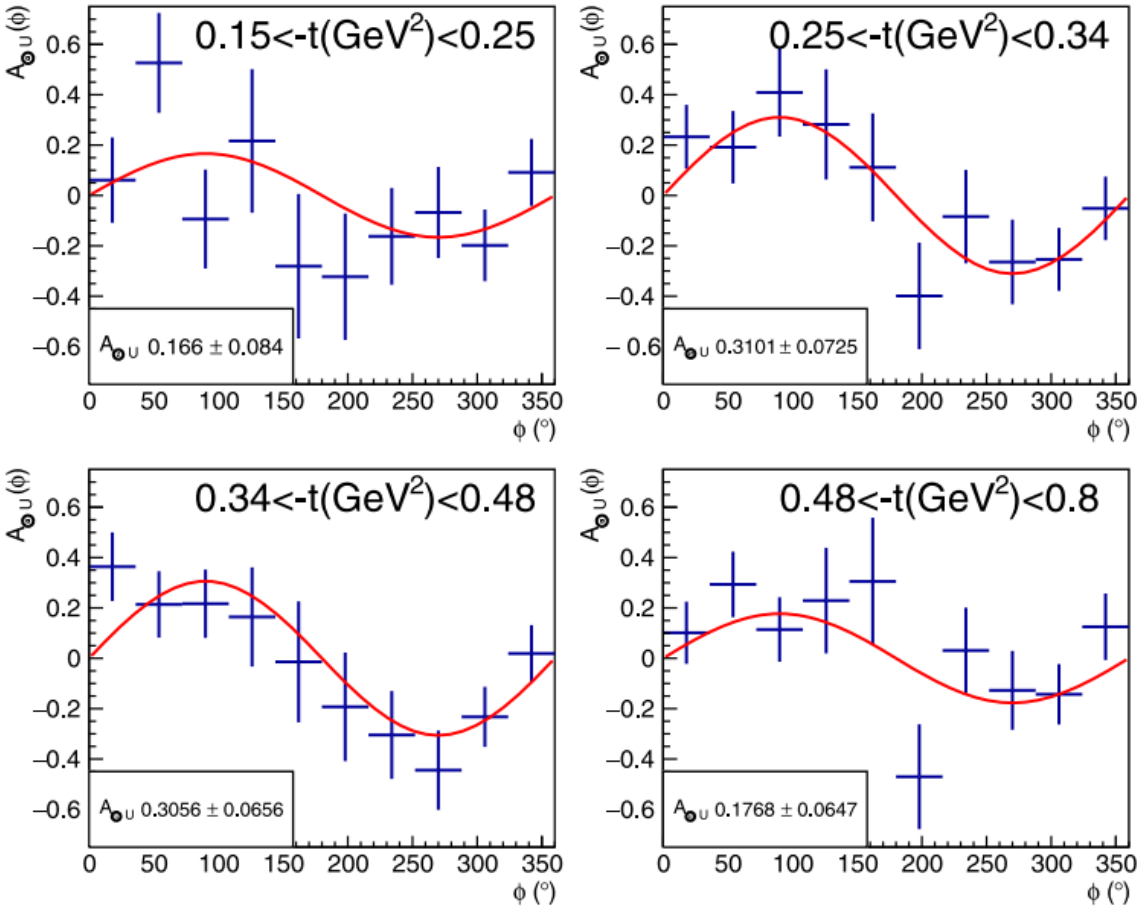
Physics Extraction – Amplitude Fitting

Currently only one published TCS measurement so far:

Phi Asymmetry integrated over Q'^2 and theta!

More appropriate to fit both angles in t Qp2 bins.

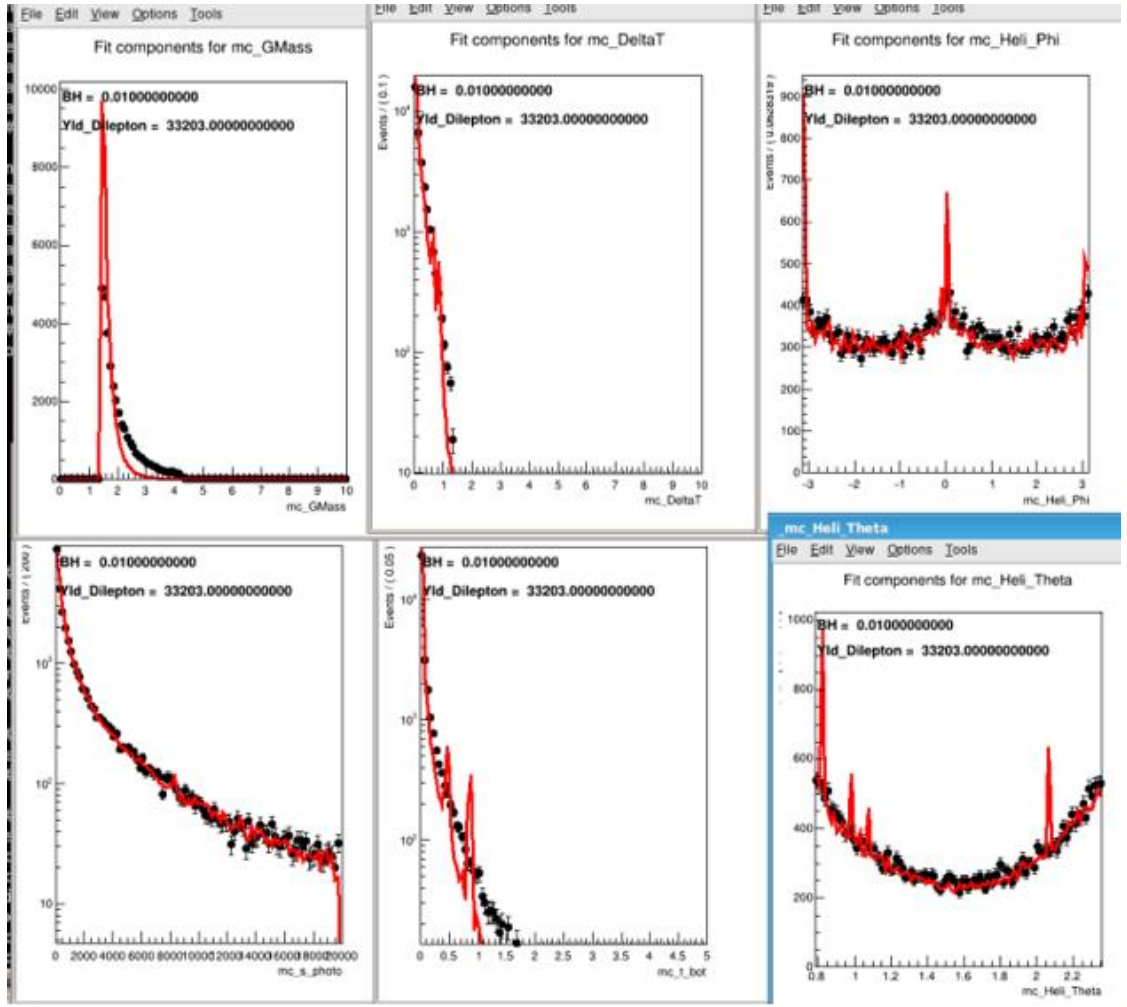
Currently building out the phenomenology for this using fitting tool https://github.com/dglazier/brufit



BSA from the first measurements of TCS, P. Chatagnon et al (2021) PRL.127.262501

Physics Extraction – Amplitude Fitting

Brufit Work



$$\frac{d\sigma_{BH}}{dQ^2\,dt\,d(\cos\theta)\,d\phi} = \frac{\alpha_{em}^3}{4\pi(s-M^2)^2}\frac{\beta}{-tL}\left[\left(F_1^2 - \frac{t}{4M^2}F_2^2\right)\frac{A}{-t} + (F_1+F_2)^2\frac{B}{2}\right]$$

$$A = (s-M^2)^2\Delta_T^2 - t\,a(a+b) - M^2b^2 - t(4M^2-t)Q^2 + \frac{m_\ell^2}{L}\left\{\left[(Q^2-t)(a+b) - (s-M^2)b\right]^2 + t(4M^2-t)(Q^2-t)^2\right\}$$

$$B = (Q^2+t)^2 + b^2 + 8m_\ell^2Q^2 - \frac{4m_\ell(t+2m_\ell^2)}{L}(Q^2-t)^2$$

$$L(\theta,\phi) = \frac{(Q^2-t)^2 - b^2(\theta,\phi)}{4}.$$

TCS Cross section is 4fold in t Qp2 theta phi

Physics is encoded in the amplitudes controlled by the CFFs.

Left: Proof of concept - Fitting the (full) BH amplitude for EpIC data with brufit.

Next: - extend to Interference term -> Re and Im parts give the CFFs

- Bin in Qp2 and t
- Fold in acceptances for statistical projections on Observables


University of Glasgow
Thank You!