BB/SBS GEM Tracking Software—A Brief Overview

Andrew Puckett
University of Connecticut

Hall A Winter Meeting

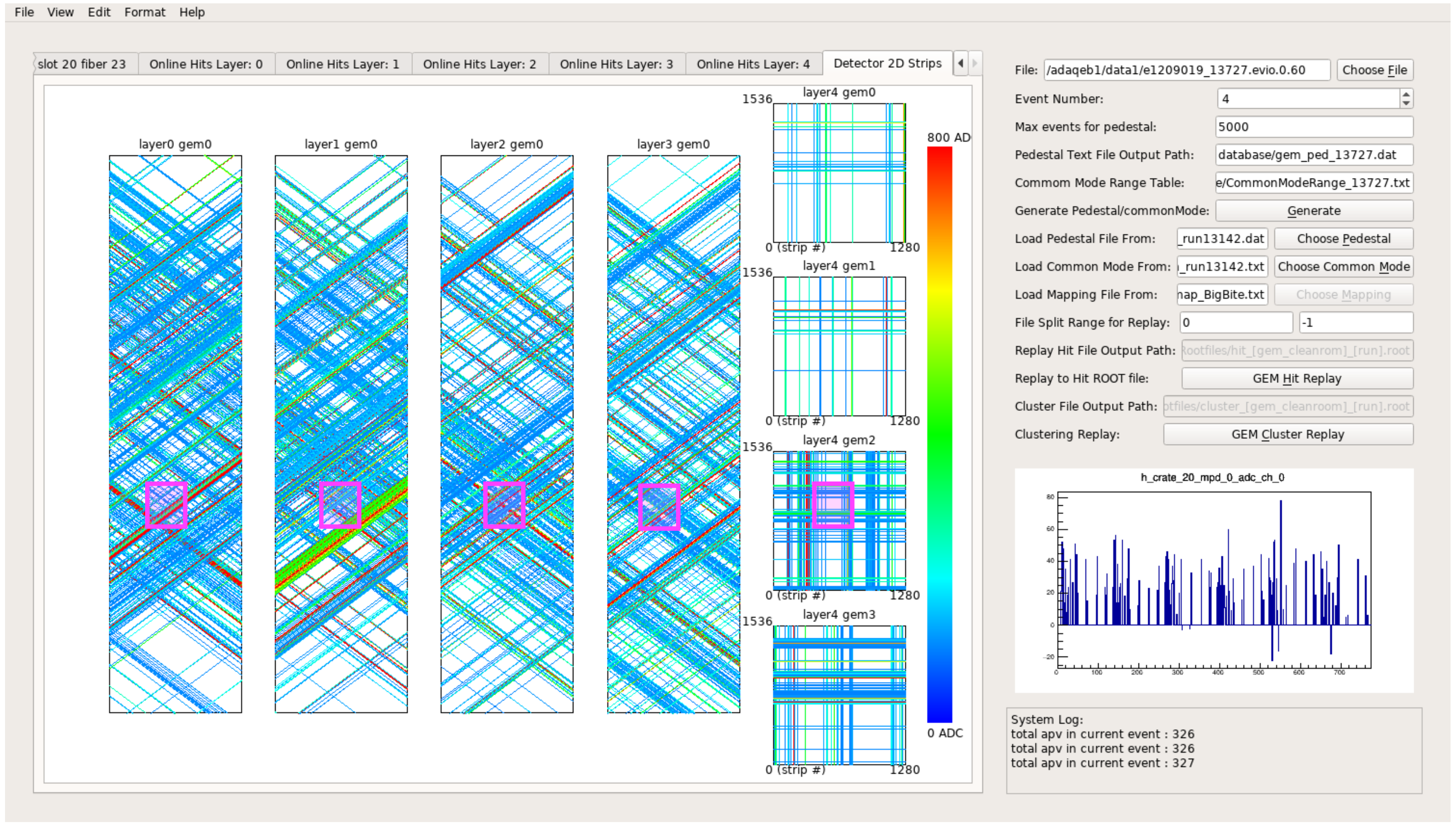
Jan. 17, 2024

UCONN

SBS GEM Software Design Principles

- Main GEM-related classes in SBS-offline:
  - MPDModule: Raw data decoder; unpacks the GEM binary data and "loads" it into Podd's "*THaSlotData*" structures that are used by the higher-level GEM decoding routines
  - SBSGEMModule: Main class for individual GEM detectors. Contains GEM decoding, clustering, and hit-finding routines. Inherits Podd's *THaSubDetector*, meaning it can't be constructed on its own, it has to be contained in some "parent" detector (e.g., *SBSGEMSpectrometerTracker*)
  - SBSGEMTrackerBase: Abstract base class containing main track-finding algorithms and geometry information, only derived classes inheriting Podd's "THaTrackingDetector" or "THaNonTrackingDetector" can actually be constructed
  - SBSGEMSpectrometerTracker: Main GEM "tracker" class for GEMs used in "spectrometers" to reconstruct particle kinematics back to the target (angles, vertex, momentum, etc). Inherits *SBSGEMTrackerBase* and *THaTrackingDetector*
  - SBSGEMPolarimeterTracker (forthcoming): class for GEMs to be used for nucleon polarimetry in GEN-RP/GEP. Inherits *SBSGEMTrackerBase* and *THaNonTrackingDetector*
- SBS-offline GEM framework can handle arbitrary layouts of planar GEMs with exactly two non-parallel readout strip orientations, generically labeled "U" and "V". In other words, all actual and potential layouts of GEMs that are contemplated for the SBS program.
- For now only straight-line tracking is supported (again, with SBS requirements in mind).
- Rectangular geometry of active area is implicitly assumed in calculating position of allowed 2D hit candidates and checking whether hits/ projected tracks are in GEM active area. Non-rectangular geometries (e.g., SOLID) could be accommodated with straightforward modifications, but curved tracking in a magnetic field is a heavier lift.
- Only "2D" hit combinations can currently be used in tracking. This does not result in a significant loss of efficiency due to the charge correlation between "X" and "Y" (or "U" and "V") strips.
- GEM "Modules" (individual detectors) are grouped into "layers" for tracking purposes.
- Tracking proceeds in three dimensions from the start; each GEM layer can provide exactly one point in 3D space along the track; tracking algorithm finds hit combinations mainly based on the χ^2 of a straight-line fit in 3D space, augmented by other criteria such as timing of hits, time and charge correlation between "U" and "V" strips, target reconstruction, etc.
- "Region of interest" for tracking is defined by front and back "constraint points" and widths. These have to be provided by external detectors. Tracking without external constraints is only viable under low-occupancy conditions.

What we're up against (GMN run 13727, 12 uA LD2, $Q^2 = 4.5\ GeV^2, E = 4\ GeV$)



File: /adaqeb1/data1/e1209019_13727.evio.0.60

Event Number: 4

Max events for pedestal: 5000

Pedestal Text File Output Path: database/gem_ped_13727.dat

Commom Mode Range Table: e/CommonModeRange_13727.txt

Generate Pedestal/commonMode:

Load Pedestal File From: _run13142.dat

Load Common Mode From: _run13142.txt

Load Mapping File From: nap_BigBite.txt

File Split Range for Replay: 0 -1

Replay Hit File Output Path: rootfiles/hit_[gem_cleanrom]_[run].root

Replay to Hit ROOT file:

Cluster File Output Path: otfiles/cluster_[gem_cleanroom]_[run].root

Clustering Replay:

```
System Log:
total apv in current event : 326
total apv in current event : 326
total apv in current event : 327
```

Event display credit: Xinzhan Bai

- The fundamental ambiguity of position reconstruction is the association of 1D clusters into 2D "hit candidates".
- The number of 2D "hit candidates" is proportional to the *square* of the number of hits in the active area in the event.
- While timing and charge correlation provide some handle, for efficient track-finding we need to consider almost all possible combinations in practice under high-rate conditions due to small signals and signal/noise ratios
- At high rates/occupancies, tracking only becomes feasible with constraints from external detectors to reduce the combinatorics that must be considered.

- Single event display for BigBite GEMs; all fired strips color-coded by ADC values

🟪 = approximate size of calorimeter-constrained track search region at each layer

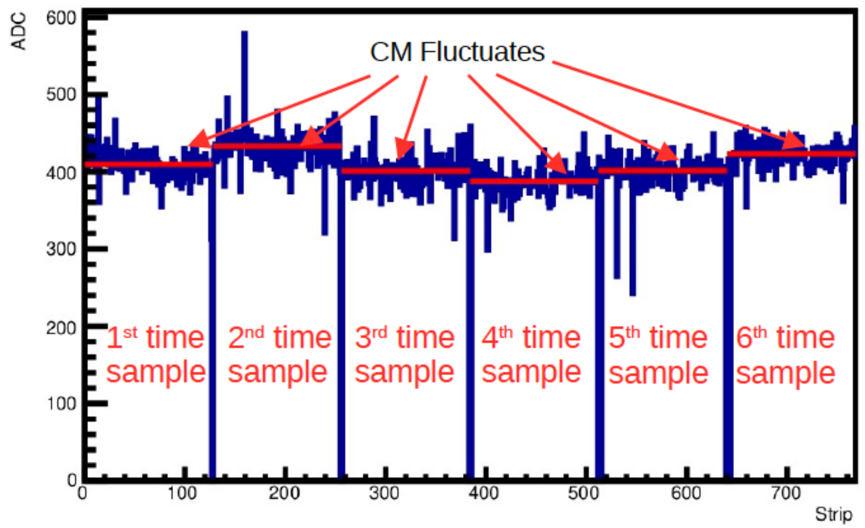
SBS/BigBite GEM decoding

APV Raw Data

- For every event and for every APV time sample the common mode (CM) value fluctuates.
  - This value must be calculated in real time and subtracted to recover the real signals.
- A pedestal run is first analyzed and records the following information is extracted.
  - Mean CM value for each APV
  - RMS CM value for each APV
  - Mean pedestal value for each strip
  - RMS pedestal value for each strip

From Sean Jeffas

One APV Event



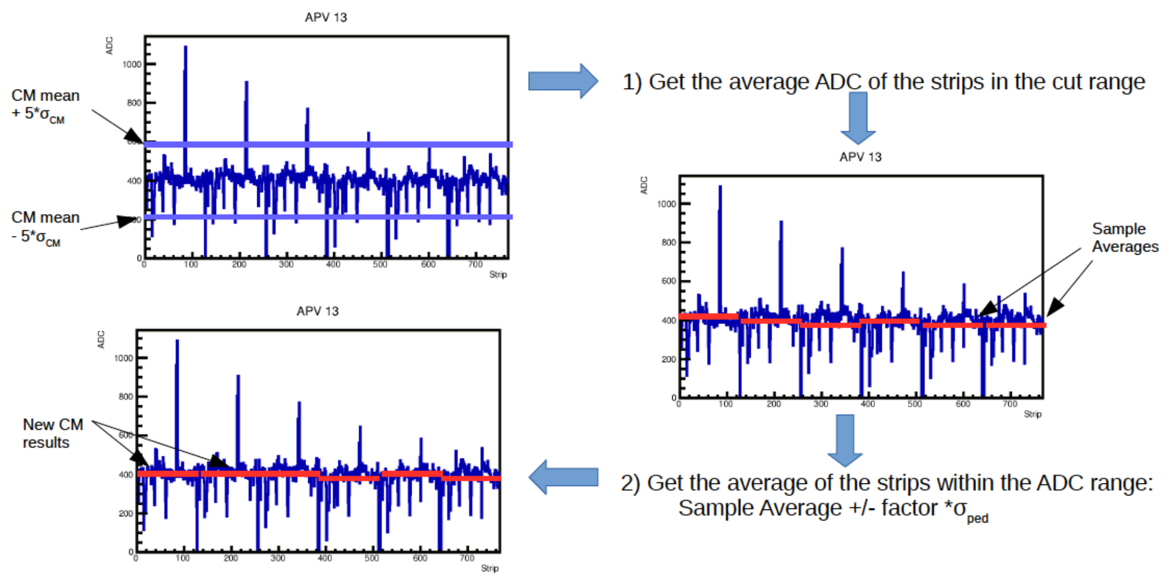
2

- Raw data format: 6 12-bit APV25 ADC samples per strip per event; MPD readout
  - Efficient binary encoding (B. Raydo *et al.*) 2 ADC samples per 32-bit word
- Each strip has a constant offset ("pedestal") and both "random" and "common mode" noise (random global fluctuation of APV25 baseline from sample to sample, common to all 128 channels on that chip)
- Online zero-suppression using "Danning" algorithm during GMN, and "enhanced Danning" algorithm during GEN
- 1/100 events full readout for debugging/data quality monitoring/sanity-checking/corrections
- Unanticipated negative "sag" of common-mode baseline during GMN necessitated the development of offline corrections which make heavy use of the full-readout events!
- Three common-mode algorithms are routinely in use offline (each has its pros and cons)

Online and "offline" common-mode algorithms: From Sean Jeffas

Danning Algorithm

- CM mean, CM RMS, and pedestal RMS values from a pedestal run are loaded to the DAQ.



- Step 2 can be repeated for multiple iterations.

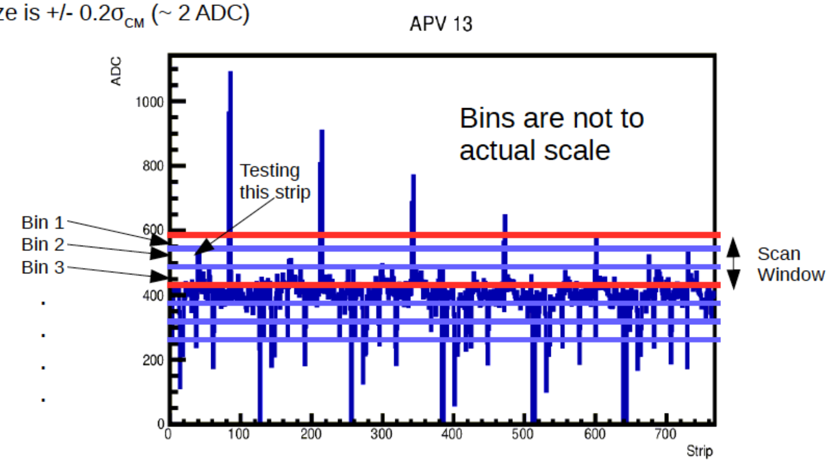
4

Histogramming Method

- Developed by Andrew Puckett.
- Slice the ADC range into bins, and count how many entries are in each bin.
- Low ADC noise should group around the baseline.
  - Extremely unlikely for several strips with signal to group into the same bin.

Histogramming Procedure:

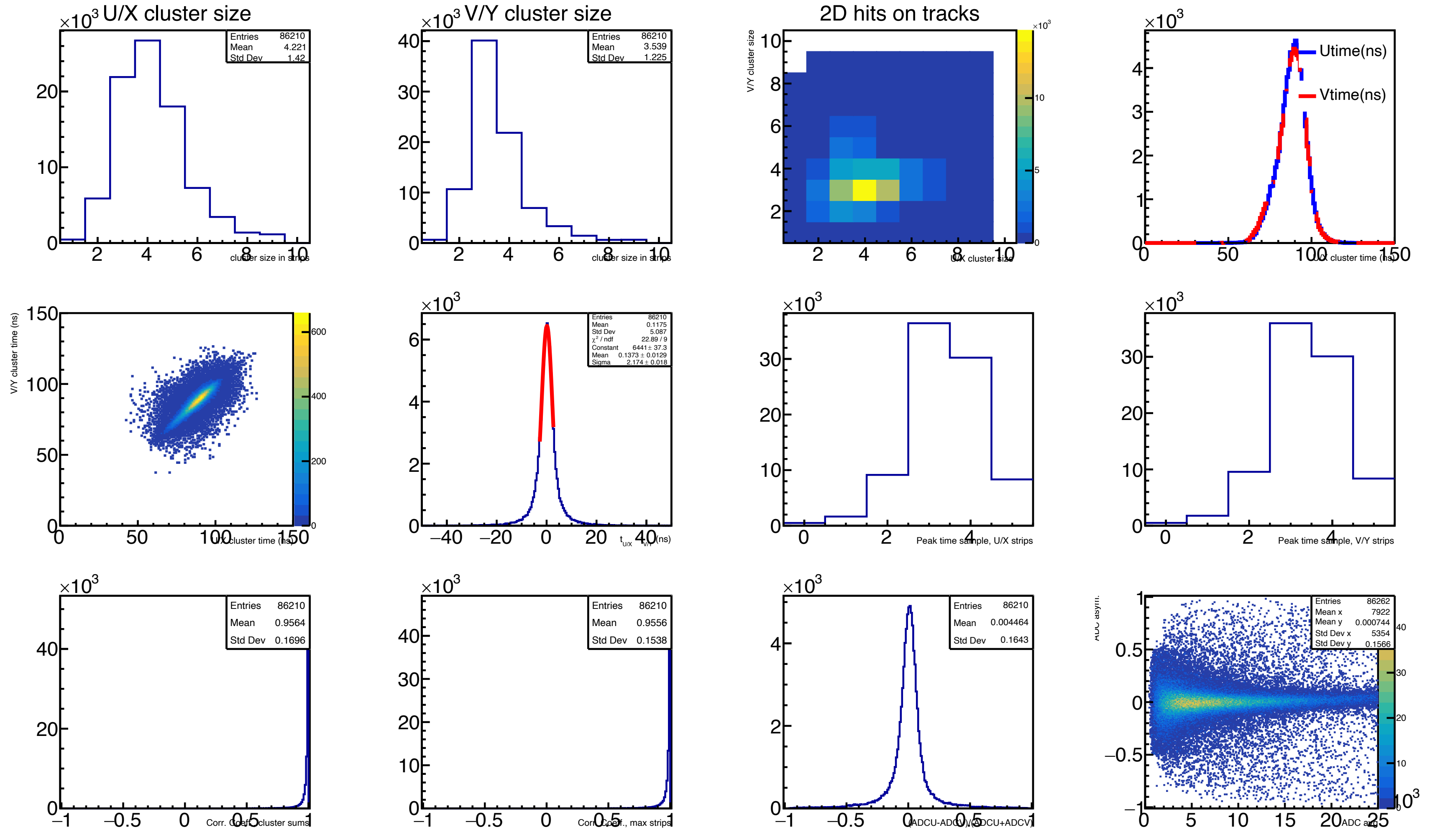
- Add each strip to all bins in the scan window
  - This is necessary for proper statistics, since there are only 128 strips.
- The bin with the maximum number of counts is averaged
- 10 bins inside each scan window
  - Scan window is +/- 2σ_{CM} (~ 20 ADC)
  - Bin size is +/- 0.2σ_{CM} (~ 2 ADC)



5

- The "Danning algorithm" averages all strips within a defined window on a first iteration, and then shrinks the upper (and lower) limits of the window and calculates 2nd and subsequent averages using the results of the previous iteration.
- "Histogramming" or "scanning window" method counts ADC hits in a series of overlapping bins of user-configurable width and step size (small compared to width), and averages ADC values in the bin with the largest number of hits

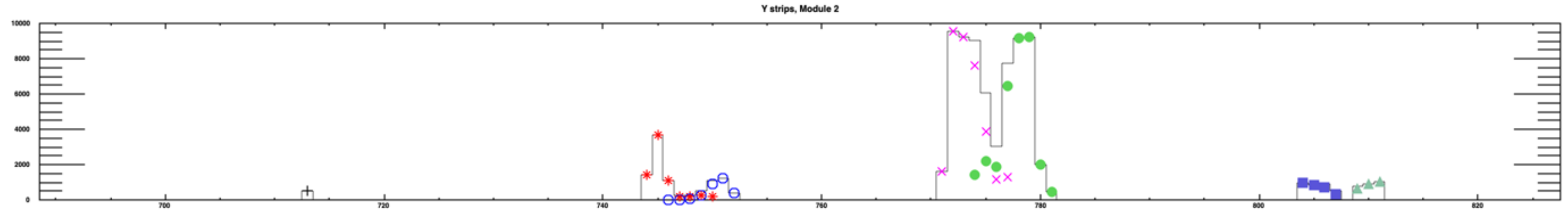
1D Clustering and 2D Hit Association

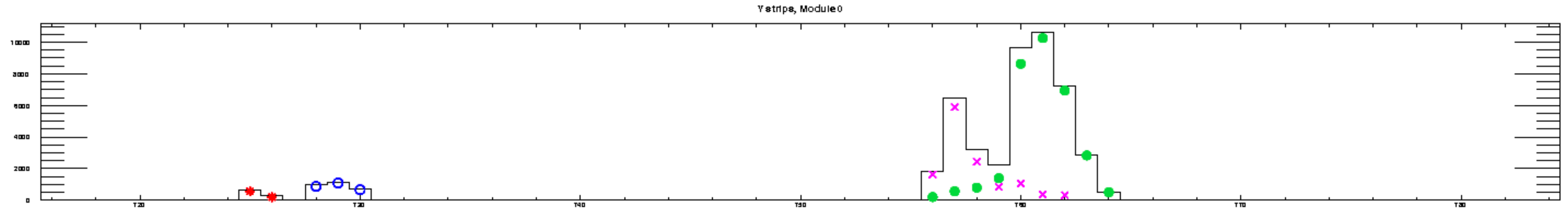


Above: Standard "online" display of BigBite GEM clustering results

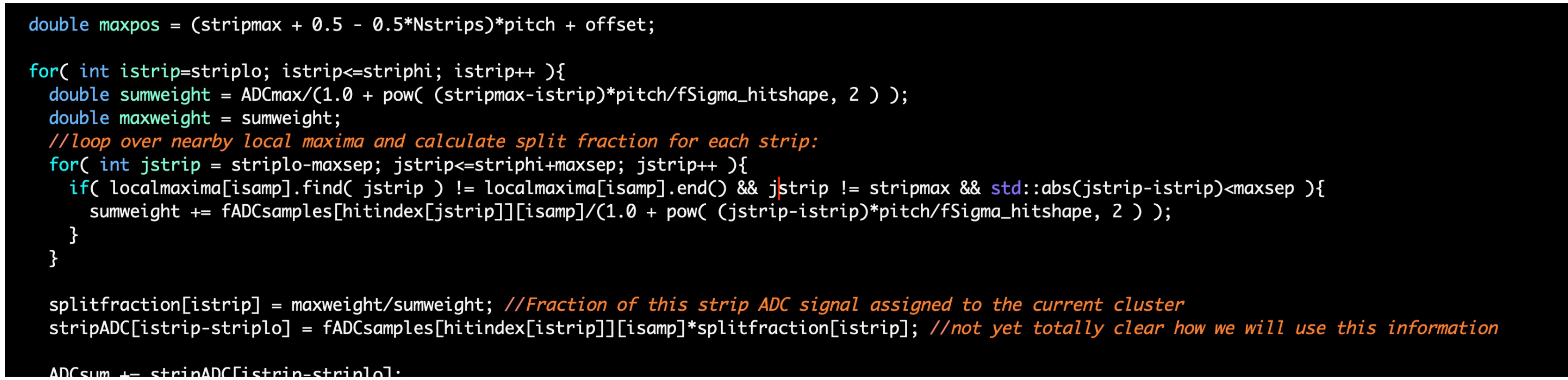
- 1D clustering is simple and fast; loop on all fired strips and find local maxima of ADC value (passing some timing cuts)
- Add all neighboring fired strips to the left and the right that are consistent in time with the central maximum.
- Check for overlapping clusters in contiguous groupings of strips, calculate "peak prominence" and erase "insignificant" maxima.
- Split overlapping clusters (next slide)
- Form all possible combinations of 2D hits, with possible filtering based on ADC/time correlation, etc.

Splitting Overlapping Clusters-–Event Display Examples





- These examples are from cosmic data. In cases of “peak-valley-peak” in a contiguous group of strips, we calculate “peak prominence”, discard local maxima that are insufficiently “prominent”, and split overlapping clusters by assigning a weight estimating the contribution of each ”hit” (local maximum) to the signal on a given strip, according to an assumed Lorentzian shape with user-configurable width



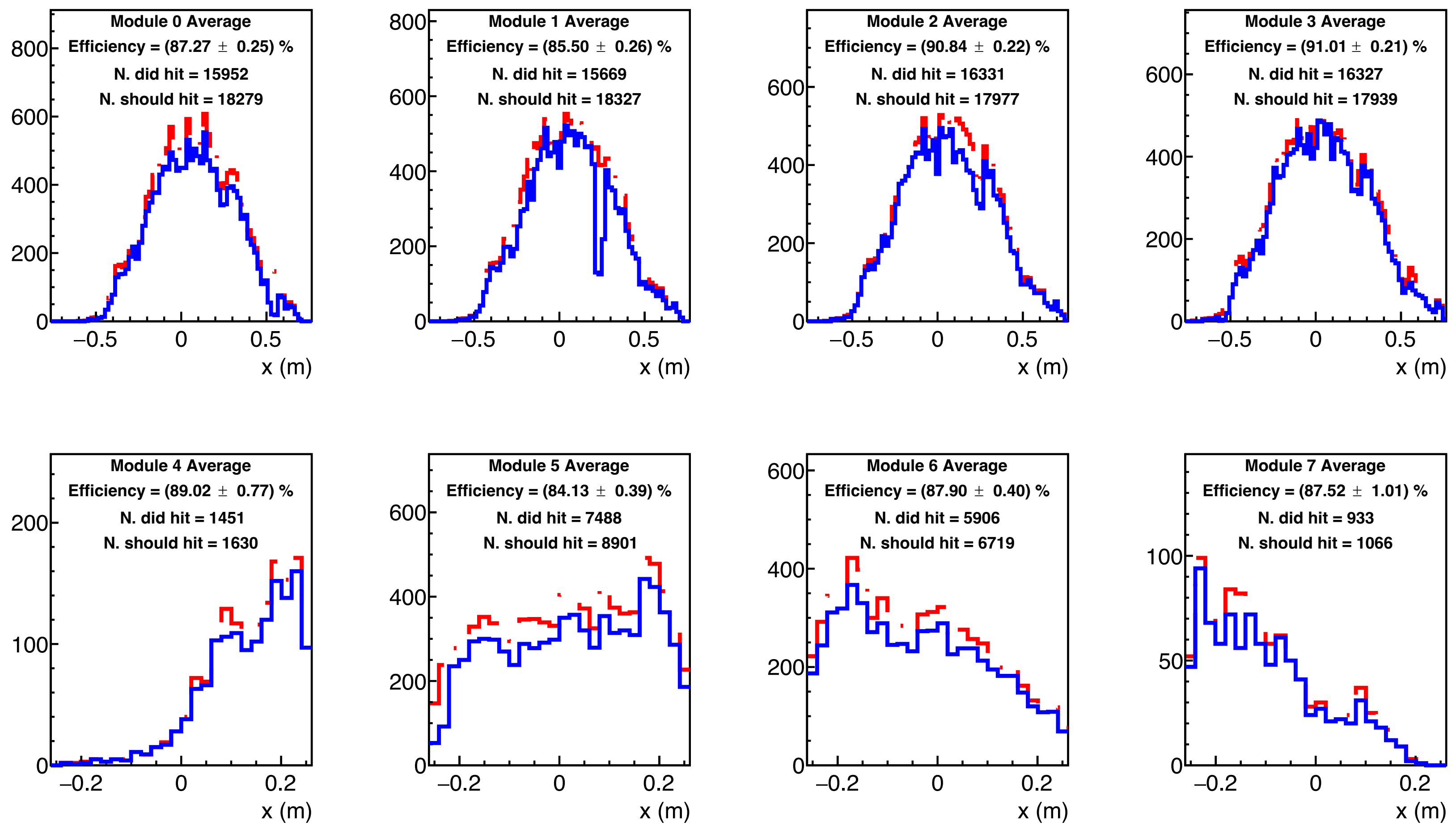
```
double maxpos = (stripmax + 0.5 - 0.5*Nstrips)*pitch + offset;

for( int istrip=striplo; istrip<=striphi; istrip++ ){
  double sumweight = ADCmax/(1.0 + pow( (stripmax-istrip)*pitch/fSigma_hitshape, 2 ) );
  double maxweight = sumweight;
  //loop over nearby local maxima and calculate split fraction for each strip:
  for( int jstrip = striplo-maxsep; jstrip<=striphi+maxsep; jstrip++ ){
    if( localmaxima[isamp].find( jstrip ) != localmaxima[isamp].end() && jstrip != stripmax && std::abs(jstrip-istrip)<maxsep ){
      sumweight += fADCsamples[hitindex[jstrip]][isamp]/(1.0 + pow( (jstrip-istrip)*pitch/fSigma_hitshape, 2 ) );
    }
  }

  splitfraction[istrip] = maxweight/sumweight; //Fraction of this strip ADC signal assigned to the current cluster
  stripADC[istrip-striplo] = fADCsamples[hitindex[istrip]][isamp]*splitfraction[istrip]; //not yet totally clear how we will use this information

  ADCsum += stripADC[istrip-striplo];
```

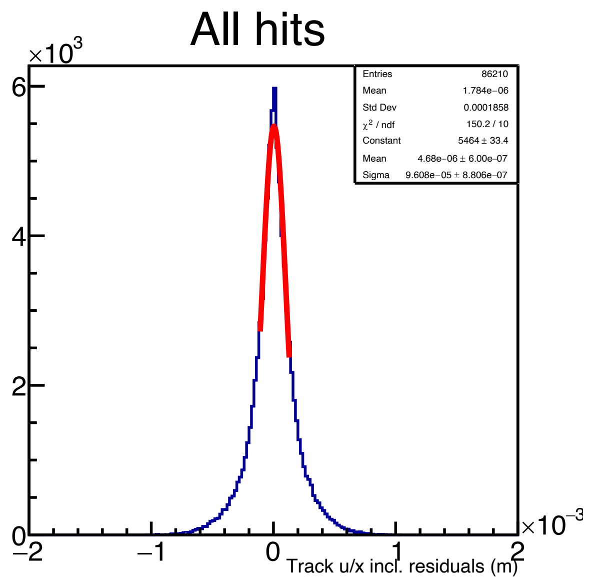
Weightings are normalized so that the sum of all “split fractions” is 1. The assumed width is $\sigma = 0.4\ mm$ (an educated guess). No systematic studies done yet on clustering performance with varying width, or with splitting enabled/disabled

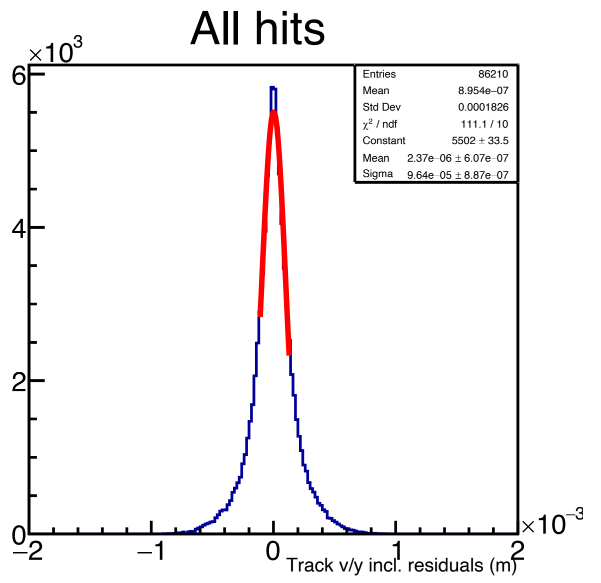
Track-finding algorithm (simplified, basic overview)

- First, we form all possible 2D hit combinations from 1D clustering results, consistent with active area and external detector constraints. Optional "filtering" based on ADC/time correlations, hit timing, and other "hit quality" parameters.
- Divide each tracking layer into a 2D uniform rectangular "grid" (currently $5x5\ mm^2$ with 1.5 mm "edge tolerance" seems to give best performance) and populate a list of 2D hit candidates in each grid bin.
- Loop over all grid bins with hits in the "front" layer. Form straight line from grid bin center to "back constraint point" (usually calorimeter cluster).
    - NEW: calculate error matrix from grid bin width and back constraint width and then project to back GEM layer to define range of grid bins that must be considered in the back layer. Very large speed boost for SBS tracking in particular, where the only constraint is a high-energy cluster in HCAL and the "front constraint" is the entire active area! Reduction of combinatorics by up to a factor of 48,000 in the worst case! This is a significant breakthrough for future "inclusive" experiments like A1n/SIDIS and also pion physics like ALL/KLL where calorimeter energy constraint cannot be as tight.
- Loop over grid bins with hits in region of interest in back layer defined by front grid bin center and back constraint point.
- For each "allowed" combination of grid bins in front and back layers, and for all combinations of one GEM hit in front and back layers within these allowed bin combinations, we form a straight line and project to all intermediate layers, with spatial resolution more or less equivalent to the intrinsic GEM resolution. If straight-line projection is within "edge tolerance" (1.5 mm) of a neighboring bin, we consider hits in neighboring bins as well.
- Within bins of interest in intermediate layers, loop on all possible combinations of one hit per layer, find the combination with best chi-squared per dof of straight-line fit in 3D, consistent with some other cuts on "hit quality" (timing, ADC correlation, optics, etc).
- Because chi-squared is always biased toward smaller numbers of hits, we start by requiring all N layers, then N-1, N-2, down to a minimum of 3 hits on the track, treating all possible combinations of M<N layers on an equal footing on each track-finding iteration.
- **Credit to Weizhi Xiong for many of the ideas that went into this track-finding algorithm and Eric Fuchey for the testing using simulation!**
- (well-configured) track-finding is now quite fast, to the point that GEM decoding is the main speed bottleneck at low occupancy, and nearly as much of a speed bottleneck as tracking even under the most extreme conditions encountered thus far.

Typical GEM Tracking Performance: BigBite

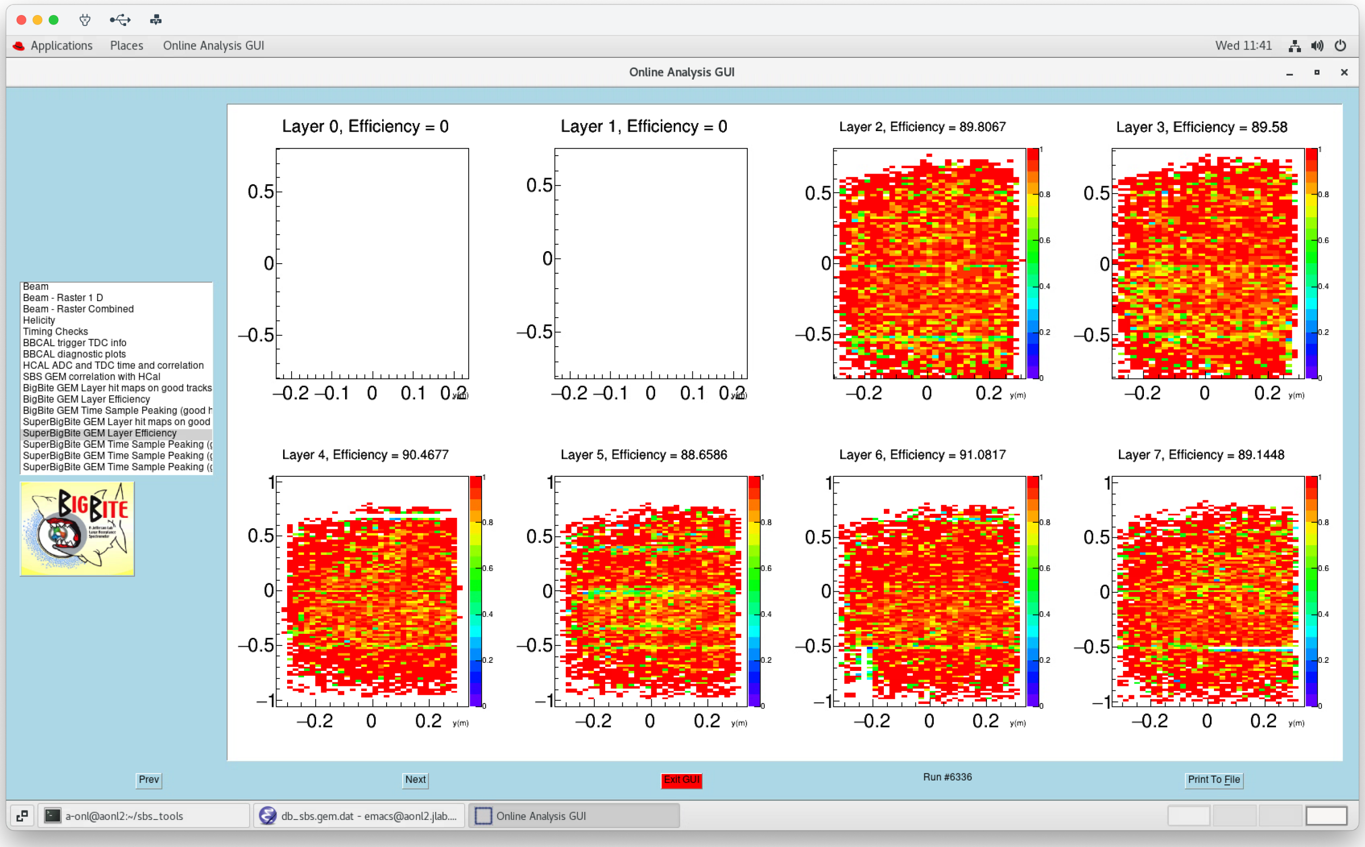


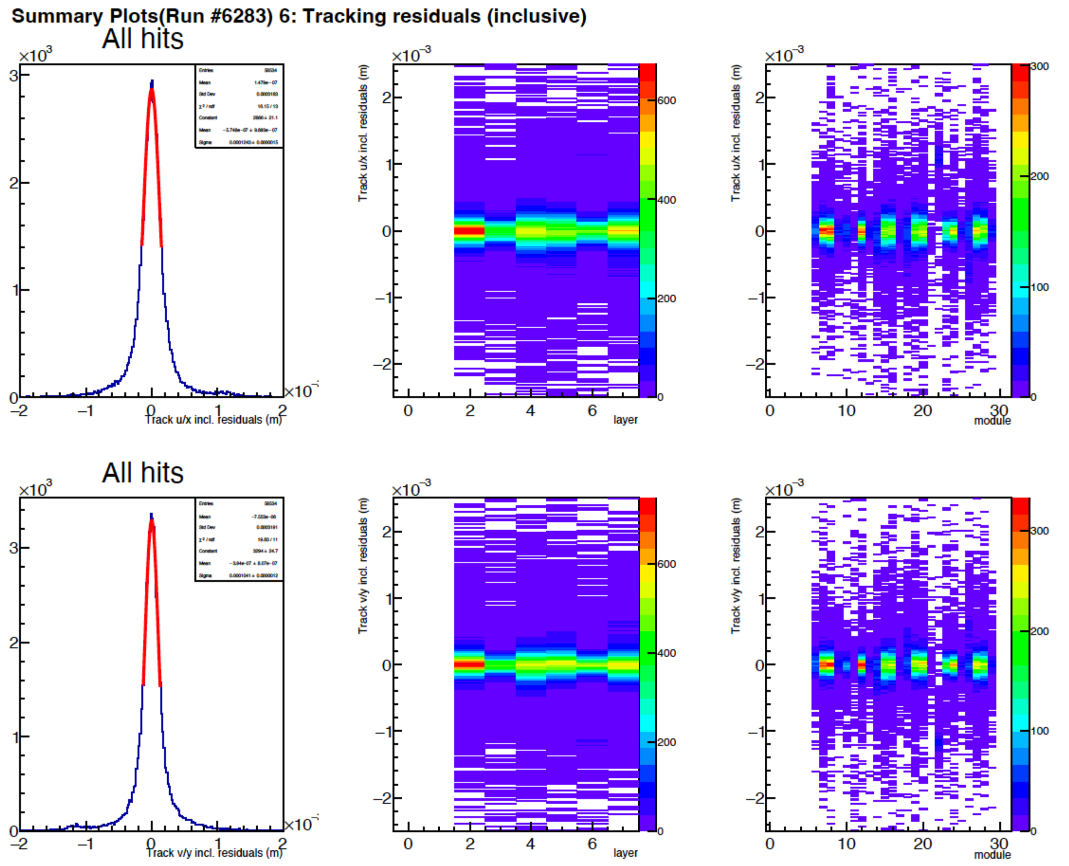




- Above: track-based efficiencies versus position, BigBite
- Right: “inclusive” tracking residuals, typical width ~100 μm

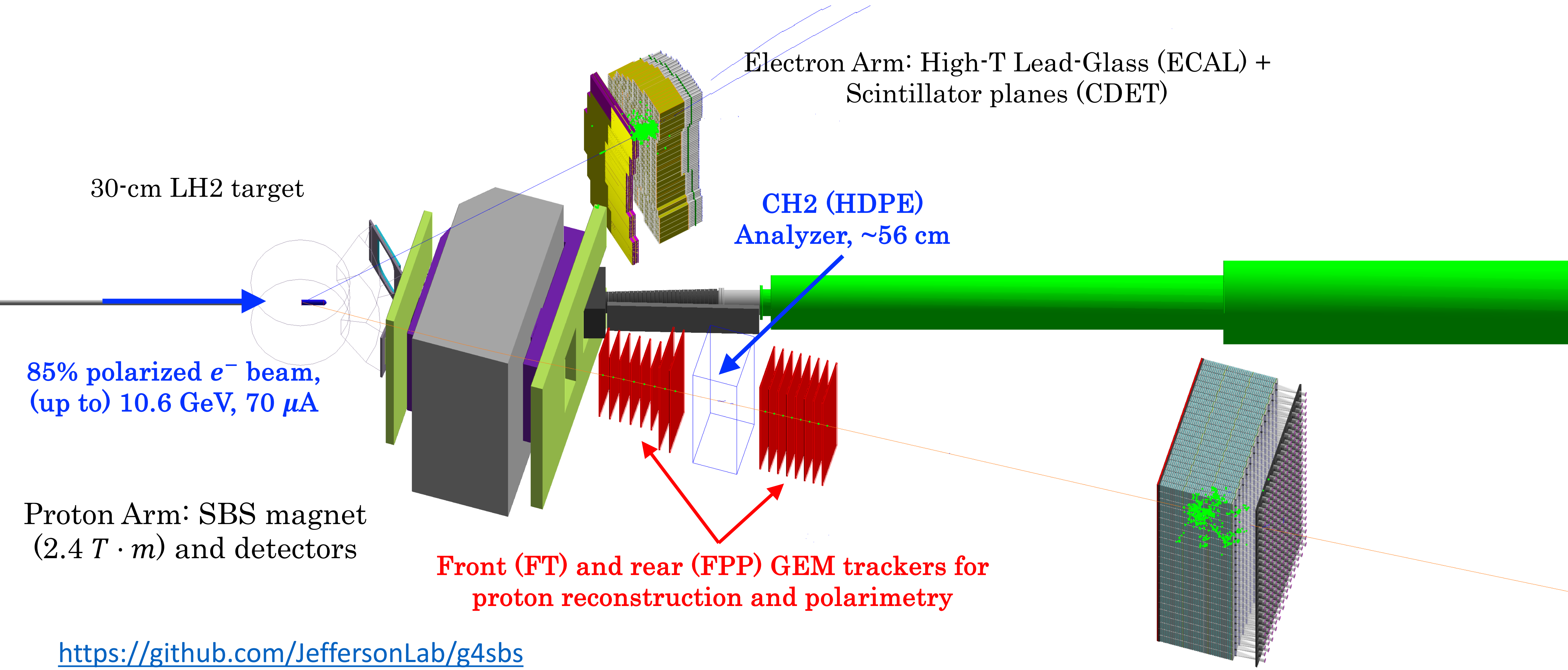
Typical GEM Tracking Performance: SBS


Applications
Places
Online Analysis GUI
Wed 11:41
Layer 0, Efficiency = 0
Layer 1, Efficiency = 0
Layer 2, Efficiency = 89.8067
Layer 3, Efficiency = 89.58
Layer 4, Efficiency = 90.4677
Layer 5, Efficiency = 88.6586
Layer 6, Efficiency = 91.0817
Layer 7, Efficiency = 89.1448
Beam
Beam - Raster 1 D
Beam - Raster Combined
Helicity
Timing Checks
BBCAL trigger TDC info
BBCAL diagnostic plots
HCAL ADC and TDC time and correlation
SBS GEM correlation with HCal
BigBite GEM Layer hit maps on good tracks
BigBite GEM Layer Efficiency
BigBite GEM Time Sample Peaking (good
SuperBigBite GEM Layer hit maps on good
SuperBigBite GEM Layer Efficiency
SuperBigBite GEM Time Sample Peaking
SuperBigBite GEM Time Sample Peaking
SuperBigBite GEM Time Sample Peaking
Prev
Next
Exit GUI
Run #6336
Print To File
a-onl@aonl2:~/sbs_tools
db_sbs.gem.dat - emacs@aonl2.jlab...
Online Analysis GUI



- Left: SBS GEM efficiency (six-layer) right before the end of (canceled) A_LL run
- Right: SBS GEM inclusive residuals (width ~= 100 um) from the same run

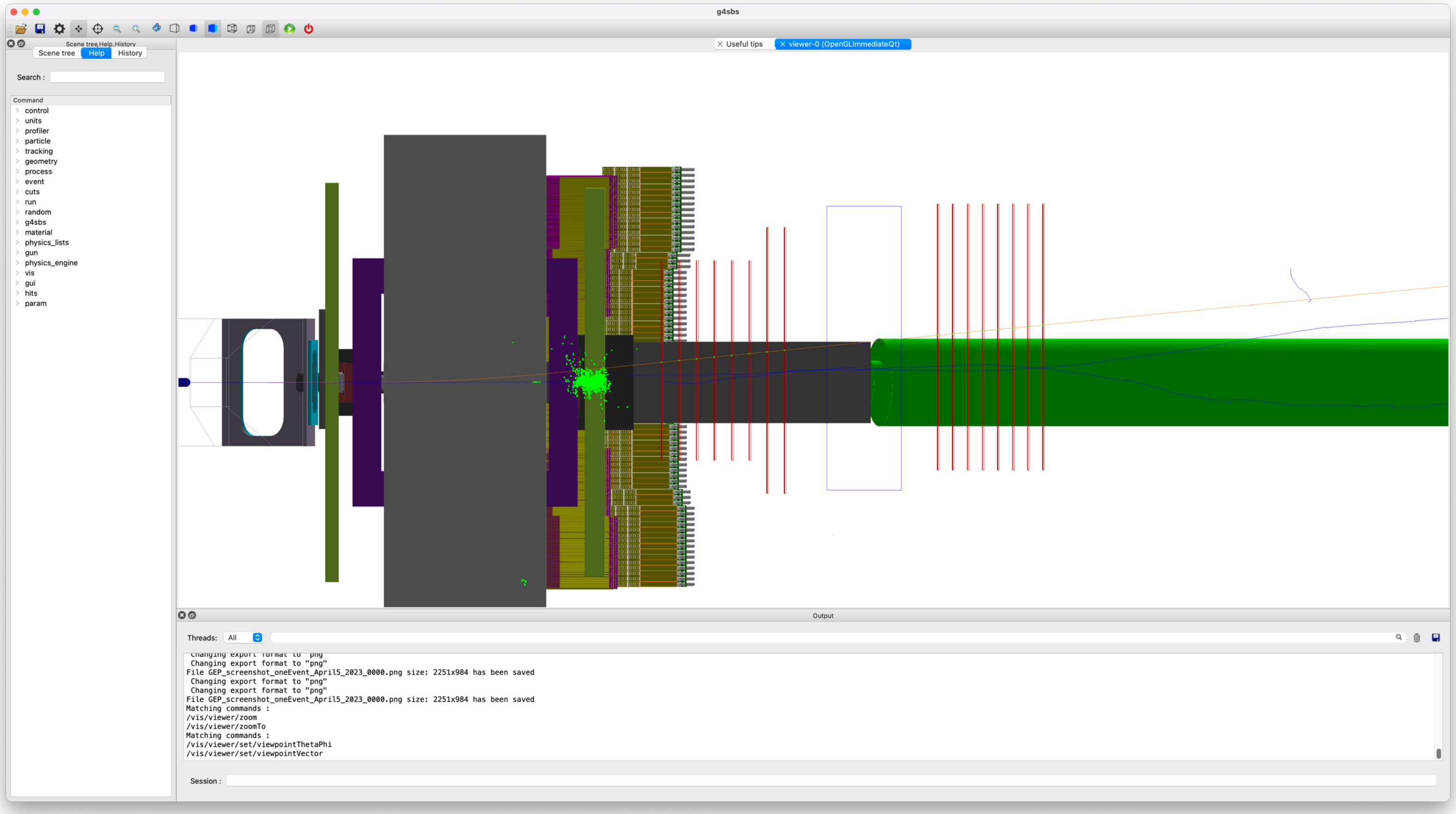
SBS-GEP in Monte Carlo


Electron Arm: High-T Lead-Glass (ECAL) +
Scintillator planes (CDET)
30-cm LH2 target
CH2 (HDPE)
Analyzer, ~56 cm
85% polarized e^- beam,
(up to) 10.6 GeV, 70 μA
Proton Arm: SBS magnet
($2.4\ T \cdot m$) and detectors
Front (FT) and rear (FPP) GEM trackers for
proton reconstruction and polarimetry

https://github.com/JeffersonLab/g4sbs

Hadron Calorimeter (HCAL)

The SBS proton polarimeter



```
 Changing export format to "png"
 Changing export format to "png"
File GEP_screenshot_oneEvent_April5_2023_0000.png size: 2251x984 has been saved
 Changing export format to "png"
 Changing export format to "png"
File GEP_screenshot_oneEvent_April5_2023_0000.png size: 2251x984 has been saved
Matching commands :
/vis/viewer/zoom
/vis/viewer/zoomTo
Matching commands :
/vis/viewer/set/viewpointThetaPhi
/vis/viewer/set/viewpointVector
```

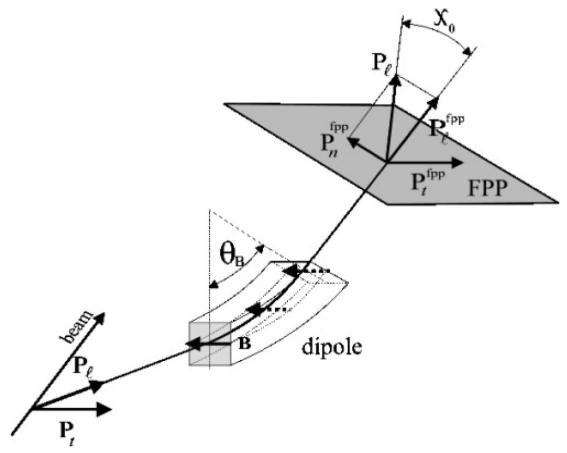


FIG. 15. Precession of the polarization component P_ℓ in the dipole of the HRS by an angle χ_θ.

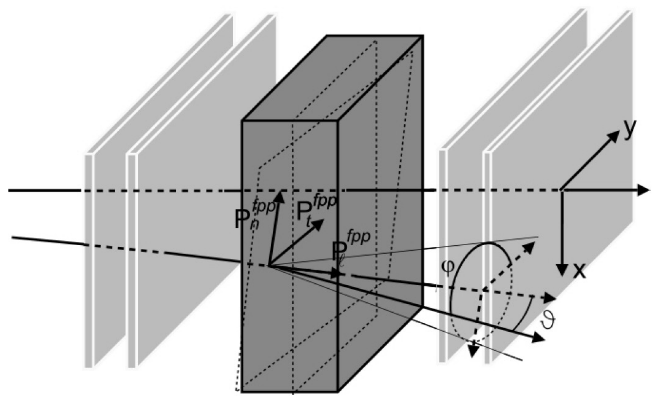
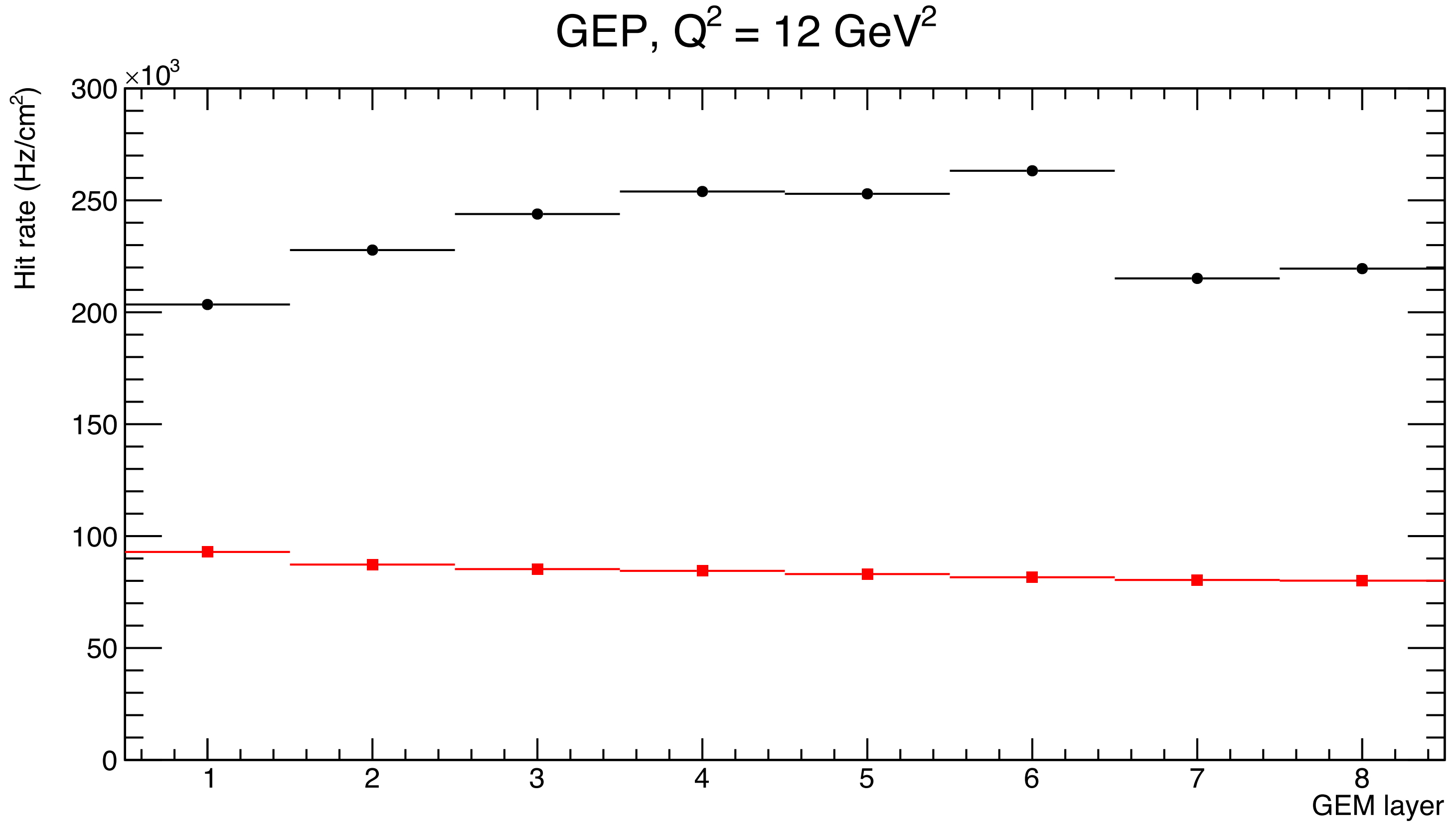


FIG. 9. Principle of the polarimeter, showing a noncentral trajectory through the front chambers, scattering in the analyzer, and a track through the back chambers; ϑ is the polar angle, and φ is the azimuthal angle from the y direction counterclockwise.

- SBS dipole field integral ~2.4 T*m rotates longitudinal polarization by approximately 80 degrees (6-degree central trajectory bend angle at highest Q, proton p = 7.3 GeV/c)
- SBS front tracker measures proton kinematics and defines incident trajectory for secondary scattering
- Azimuthal asymmetry of secondary scattering measures proton polarization; analyzing power is "self-calibrating"

GEP GEM background rates (latest estimates with new polarimeter layout)



- **Front tracker (Back tracker),** rates per unit area
- GEP background rates roughly 2-3X the worst-case from GMN
- ***g4sbs* reproduces observed BigBite GEM rate/occupancy at "low" beam currents for which the effect of gain reduction is not significant**

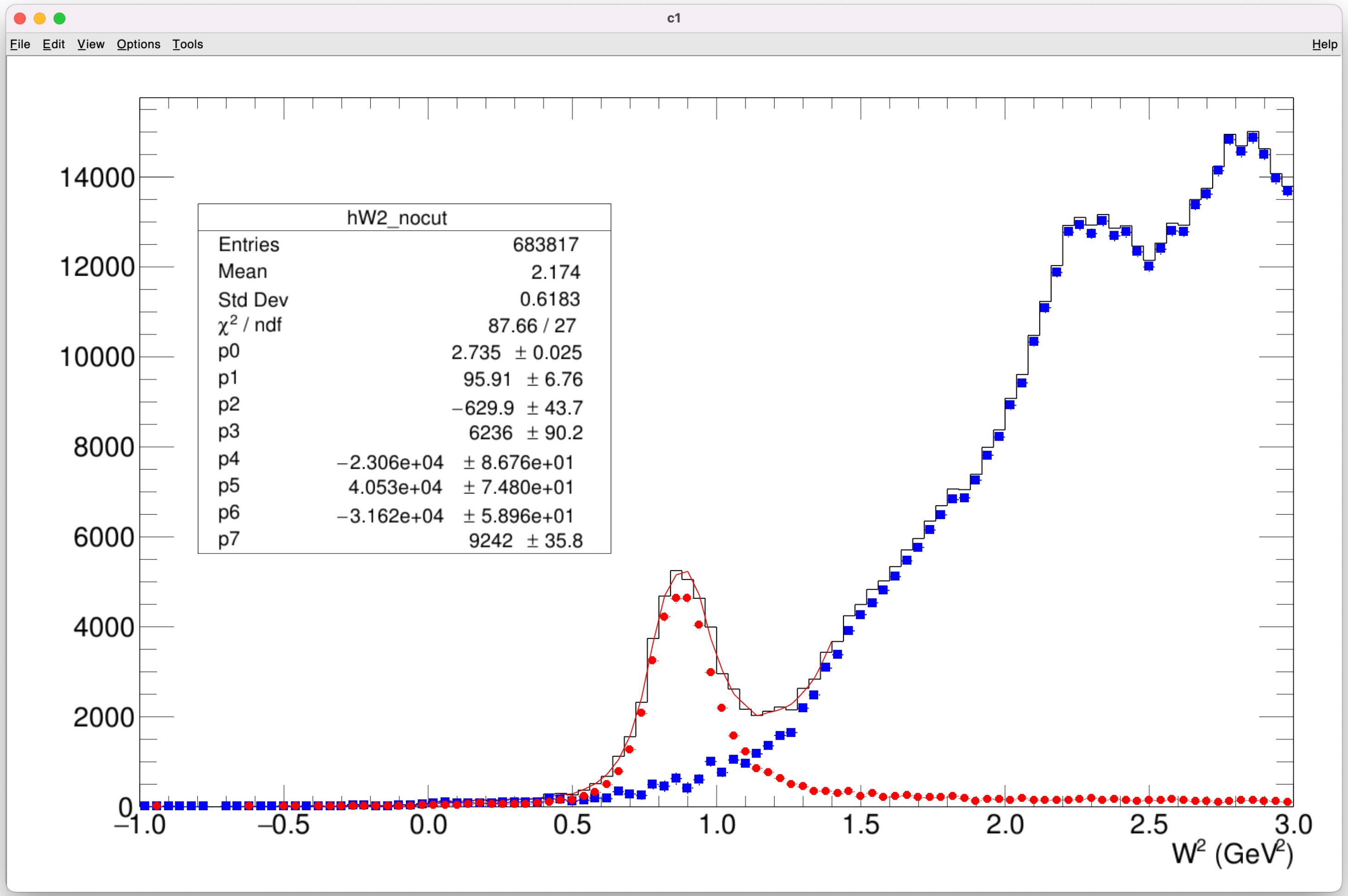
Notes

- Expected GEP rates are 2.5-3X lower than original proposal estimates (see, e.g., here and here), owing to:
  - Shorter target (30 vs 40 cm)
  - Lower beam current assumption (50 vs 75 uA)
  - GEMs slightly farther from target in final layout
- GEP tracking simulations by Weizhi Xiong for original SBS GEM layout showed good tracking performance under similar background conditions to these
- **If gain/efficiency drop is mitigated at the hardware level, then we should be in good shape**
- These background rates can be further reduced by up to ~25-40% with carefully placed/designed target shielding ("hadron filter") with acceptable impact on resolution

From GMN to GEN-RP/GEP

- Observed gain/efficiency drop during GMN led to rapid fall-off of tracking efficiency at high beam current.
- High current study at the end of GMN up to 34 uA on LH2/LD2 → GEM data taken with full readout to avoid bias from zero-suppression at high occupancy
- For GEP ERR we did simulated and real tracking efficiency studies, focused on GMN LH2 data obtained in the 4.5 GeV^2, low-ϵ kinematics.
  - LH2 data because hydrogen elastic yield is most reliable proxy for overall tracking efficiency
- GEP-equivalent beam current for this setting in terms of expected GEM background/occupancy would be ~120 uA

Elastic Yield Estimation from GMN data



```
hW2_nocut
Entries                 683817
Mean                     2.174
Std Dev                 0.6183
χ² / ndf           87.66 / 27
p0                2.735  ± 0.025
p1                95.91  ± 6.76
p2               −629.9  ± 43.7
p3                 6236  ± 90.2
p4    −2.306e+04  ± 8.676e+01
p5     4.053e+04  ± 7.480e+01
p6    −3.162e+04  ± 5.896e+01
p7                 9242  ± 35.8
```

"Yield" is charge-normalized, live-time corrected elastic peak integral → most reliable proxy for overall tracking efficiency

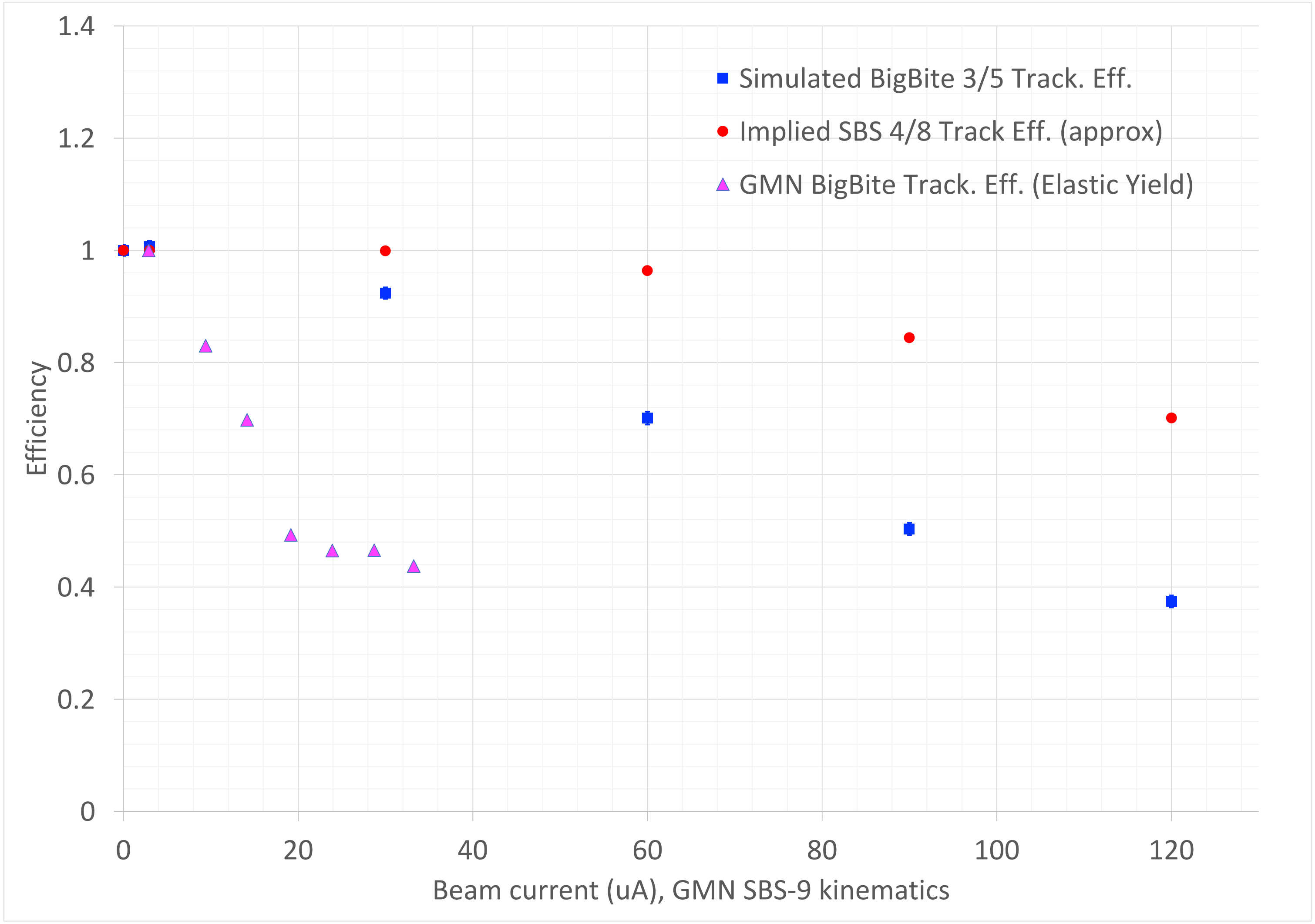
Elastic yield estimation from data

- **Black = all events**
- **Red = scaled elastic from coincidence with HCAL**
- **Blue = estimated background**

GMN SBS9 Kinematics:

$$\begin{aligned} E &= 4\ \mathrm{GeV} \\ \theta &= 49\ \mathrm{deg} \\ Q^2 &= 4.5\ \mathrm{GeV}^2 \\ \mathrm{Target} &= 15\text{-cm LH}_2 \end{aligned}$$

Real and Simulated BigBite Tracking Efficiency, GMN “SBS-9”



- *g4sbs* reproduces observed BigBite GEM rate/occupancy at “low” beam current (3 uA)
- “High current” study done at end of GMN went up to 34.5 uA on LH2, LD2 in previous slide’s kinematics.
- Relative efficiency from elastic yield drops rapidly with beam current (effect of GEM gain/efficiency drop)
- “Flattening” at high beam current not fully understood
- **Simulated BigBite tracking efficiency assuming stable gain/efficiency** shows much slower drop-off
- **GEP-equivalent** beam current for this configuration is ~120 (50) μA for Front Tracker (Back Tracker)
- Implied 4/8 efficiency for SBS FT in GEP is ~70%, consistent with assumption in PAC47 uncertainty projections (rough, preliminary)

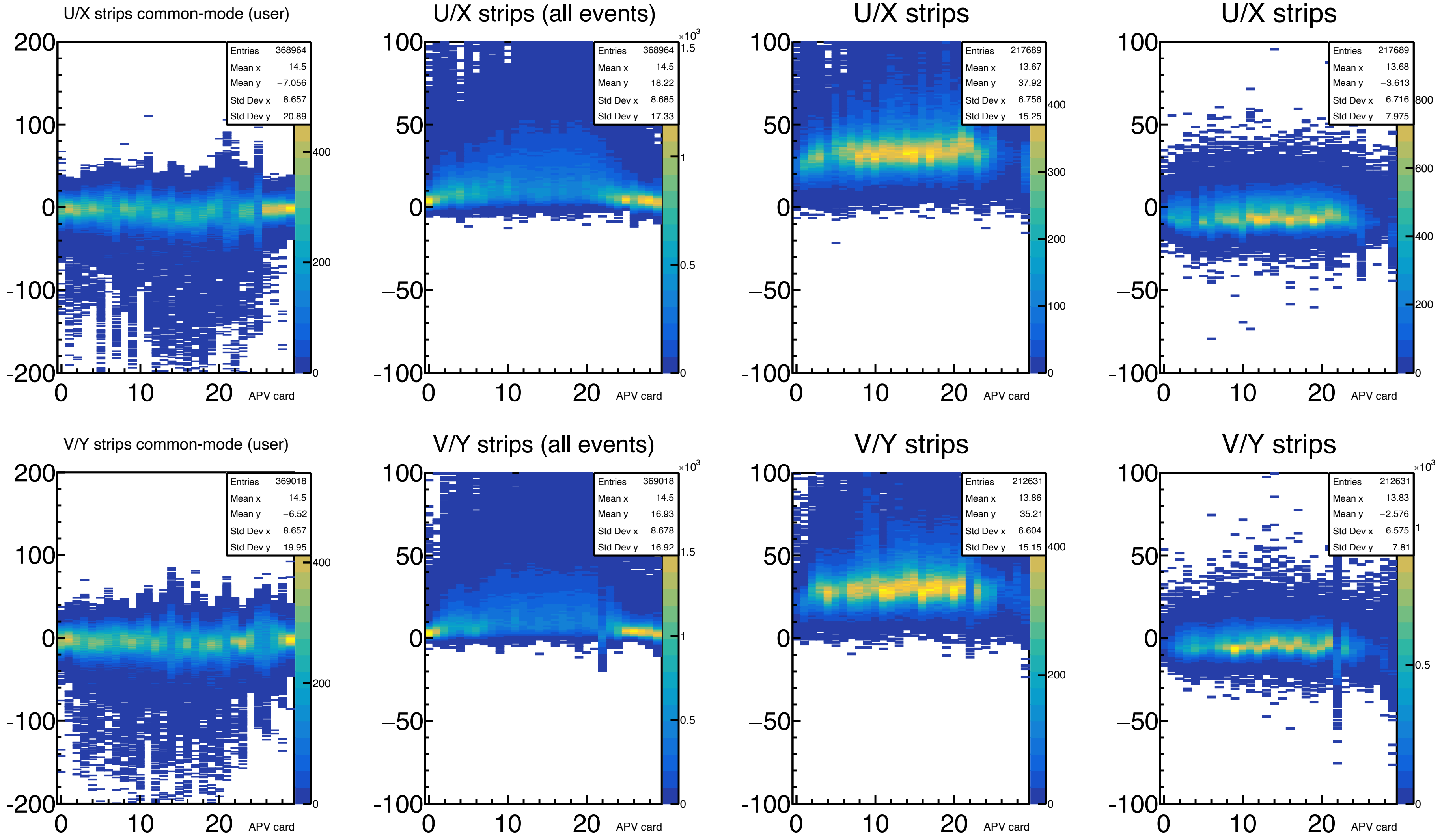
PRELIMINARY conclusions on GEM gain/efficiency during GMN

- The observed deviations from linearity of the excess divider current and the hit rate/occupancy are qualitatively consistent with the observed reductions in tracking efficiency*
- BigBite tracking was/is particularly vulnerable to this issue due to using only 5 GEM layers in tracker → gain/efficiency drop required us to run GMN/nTPE significantly below proposal luminosity (but we still got the physics!)
- Simulated BigBite tracking at various background levels (under stable gain assumption) shows that the efficiency reduction seen during GMN comes mostly (but not entirely) from the hardware.
- "Parallel divider" concept will mitigate this issue for GEP front tracker
- **GEP polarimeter redesign with more redundant 8-layer tracking assemblies in front and back trackers will make the overall tracking far more robust against individual layer inefficiencies; small dead areas, intermittent localized hardware/electronics issues, etc.**
- Hardware fix for gain drop will be verified during upcoming GEN-RP run
- **as measured by elastic yield, requiring 3/5 layers to form a track*

Summary/conclusions

- There is MUCH more to discuss, but I don't have time in this talk.
- SBS GEM software effort was a smashing success (we were reconstructing particle tracks and kinematics and reconstructing sieve hole patterns and elastic peaks and rapidly measuring GEM HV/efficiency plateaus on day one of GMN)
  - **This effort could not have been accomplished without enormous support from GEM hardware AND software and JLab/SBS DAQ teams and students/postdocs in particular, too many to name, from INFN/UVA/W&M/JLab/etc**
- Readiness of tracking software led to rapid commissioning/understanding of other BigBite/SBS detectors
- GEN-RP/GEP are big steps up in background/occupancy—hardware and software challenges abound!
- If you (or your student/PD) have an affinity for hardcore C++ programming, fast event reconstruction/tracking algorithms in a high-background environment, and/or ideas for more advanced AI/ML approaches to noise filtering/etc that can be practically implemented, let's talk!

Backups



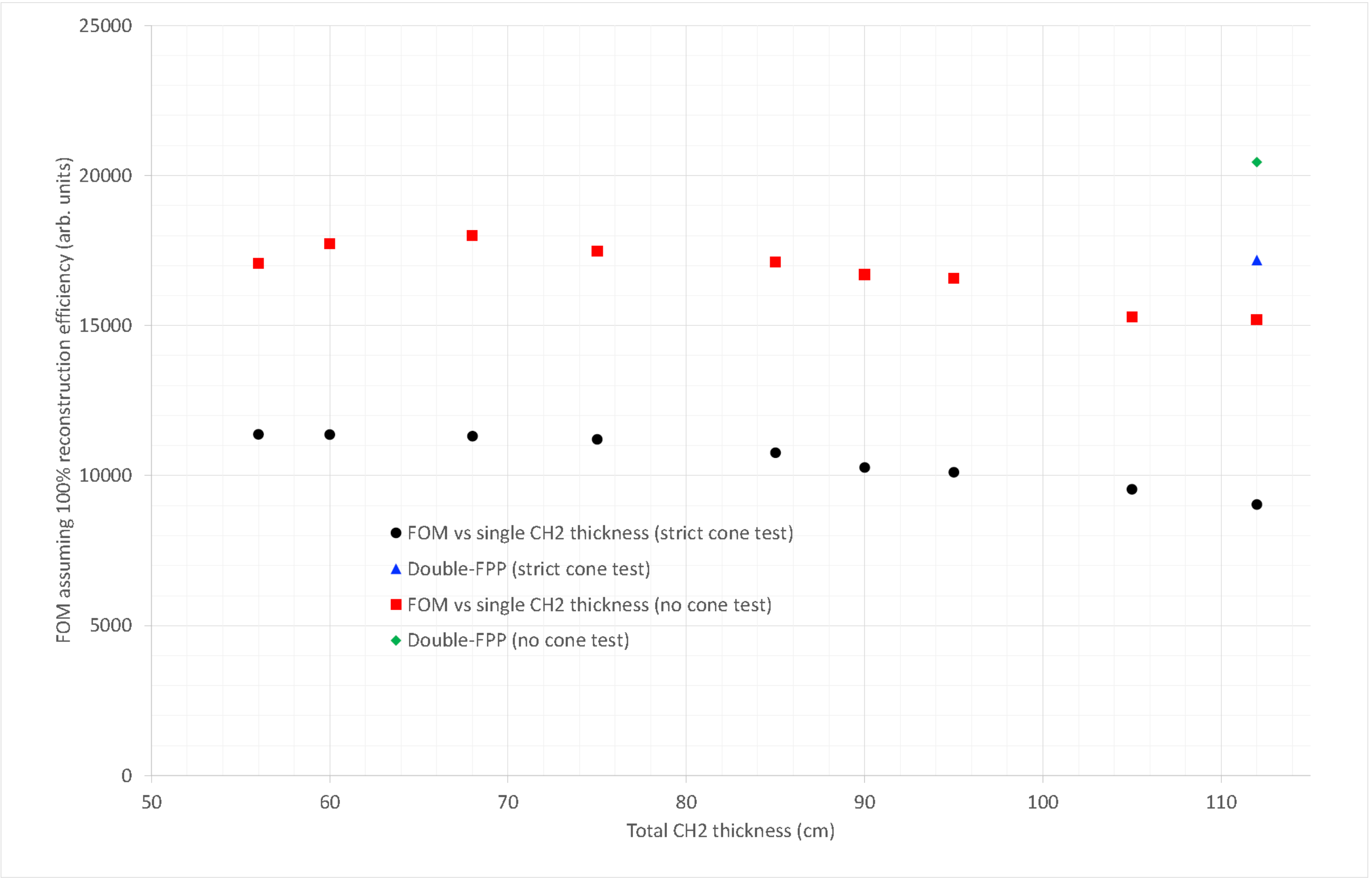
- Full readout events let us study performance of different common-mode algorithms and the biases introduced under different beam conditions
- These plots are from a relatively low-current run on LH2 (GMN, $Q^2 = 4.5$ GeV^2)

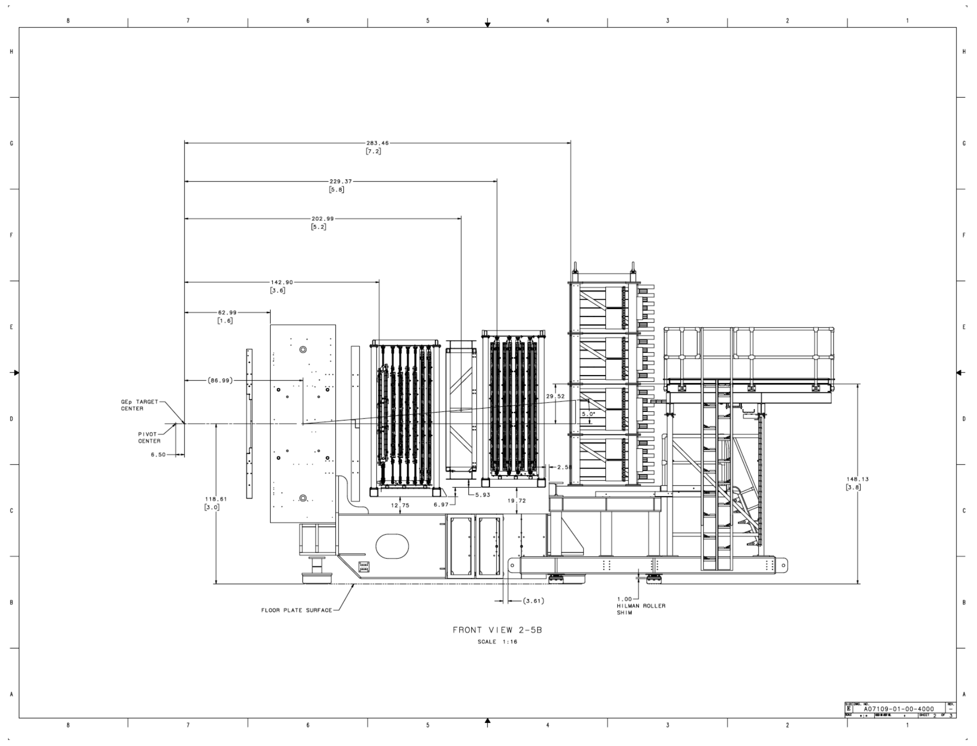
- Far left: “True” common mode from histogramming method minus mean from pedestal run by APV card
- Mid left: “True” common mode from histogramming method minus ”online Danning” algorithm
- Mid right: Applied offline common-mode correction from rolling average “bias” measured over previous N full readout events (GMN only)
- Far right: residual “bias” of corrected common-mode after corrections

NEW--SBS optics calibrations from GEN data w/SBS GEMs

- At GEN Kin-2 we have H2 reference cell data at 30% and 100% SBS field with SBS GEMs in the data stream
- Q^2 is relatively low, elastic event selection with W^2 and HCAL is extremely clean
- Can calibrate angle and vertex reconstruction extremely well
- Momentum calibration still a work in progress due to “overfitting” issues (too few independent constraints)—formalism needs some development

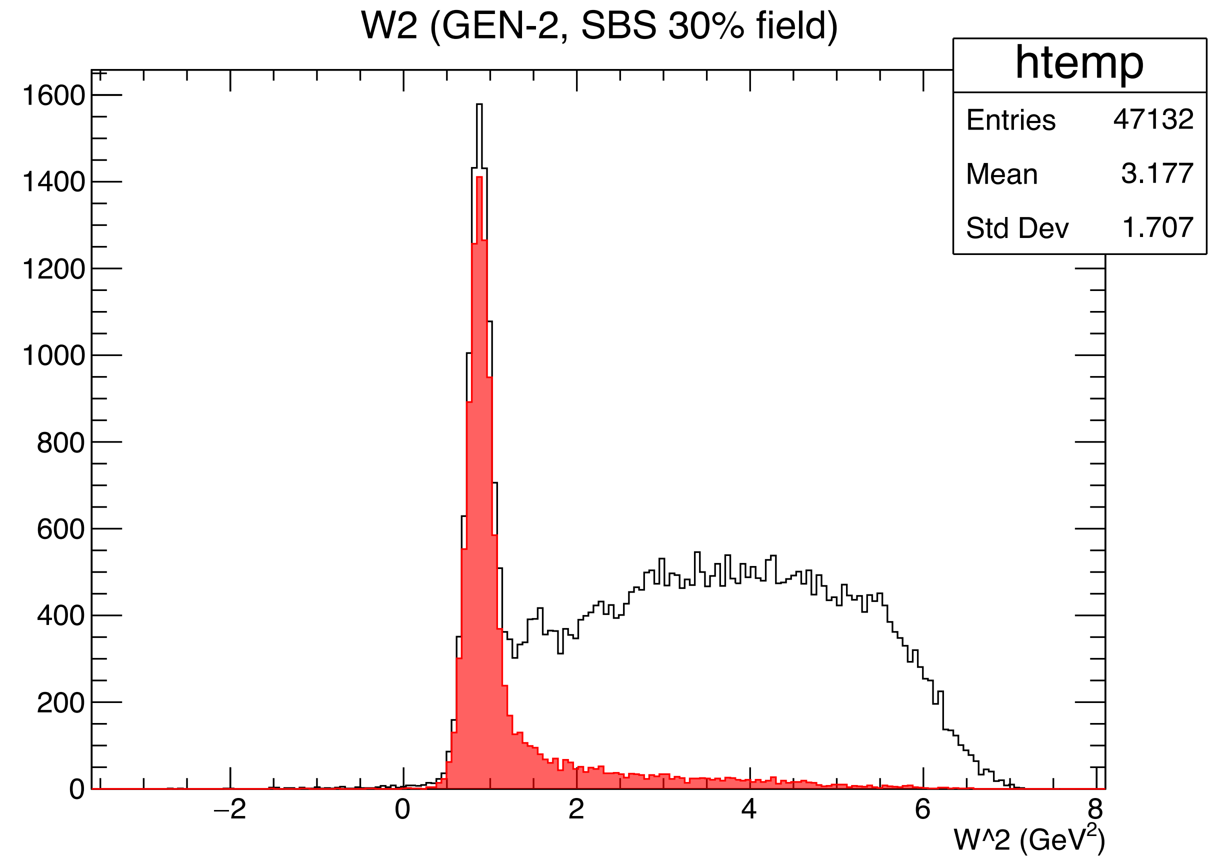
Figure-Of-Merit assessment for single-analyzer polarimeter

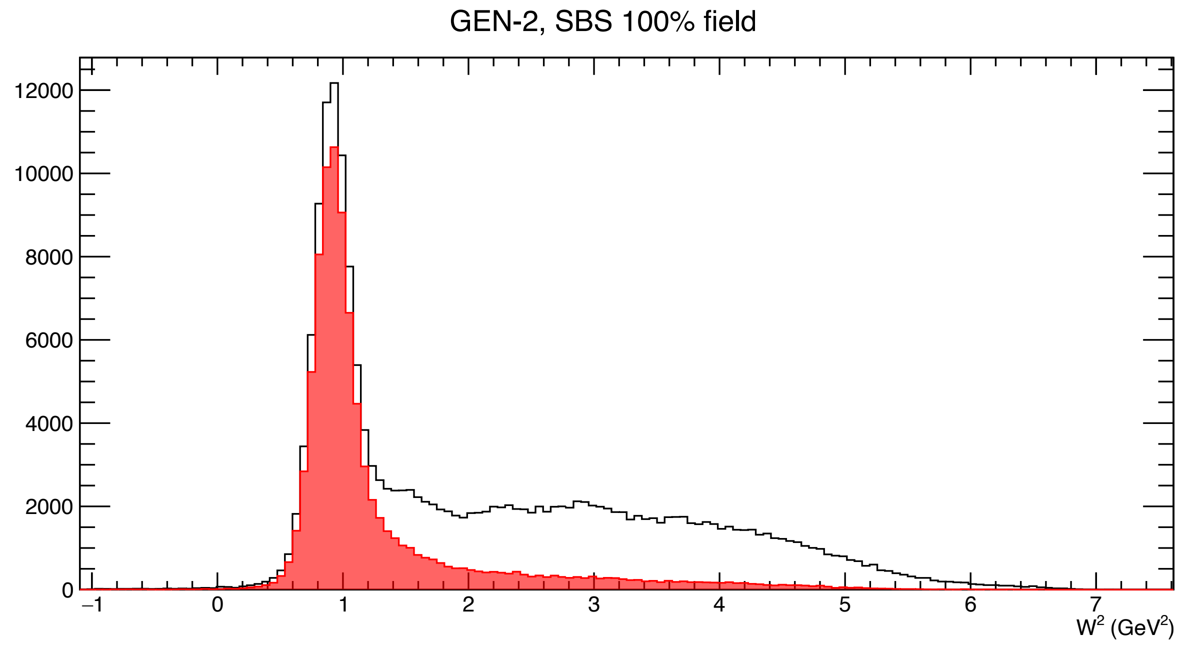




- Assuming we do NOT apply a strict cone test in the analysis, **new design has only ~15% lower FOM than original design (8% increase in error bar),** *assuming* 100% tracking efficiency in both cases. This includes acceptance differences between the two designs.
- The extra redundancy in tracking and the simplicity of tracking geometry, event topology and external detector constraints should more than offset the loss in nominal FOM by improving reconstruction efficiency

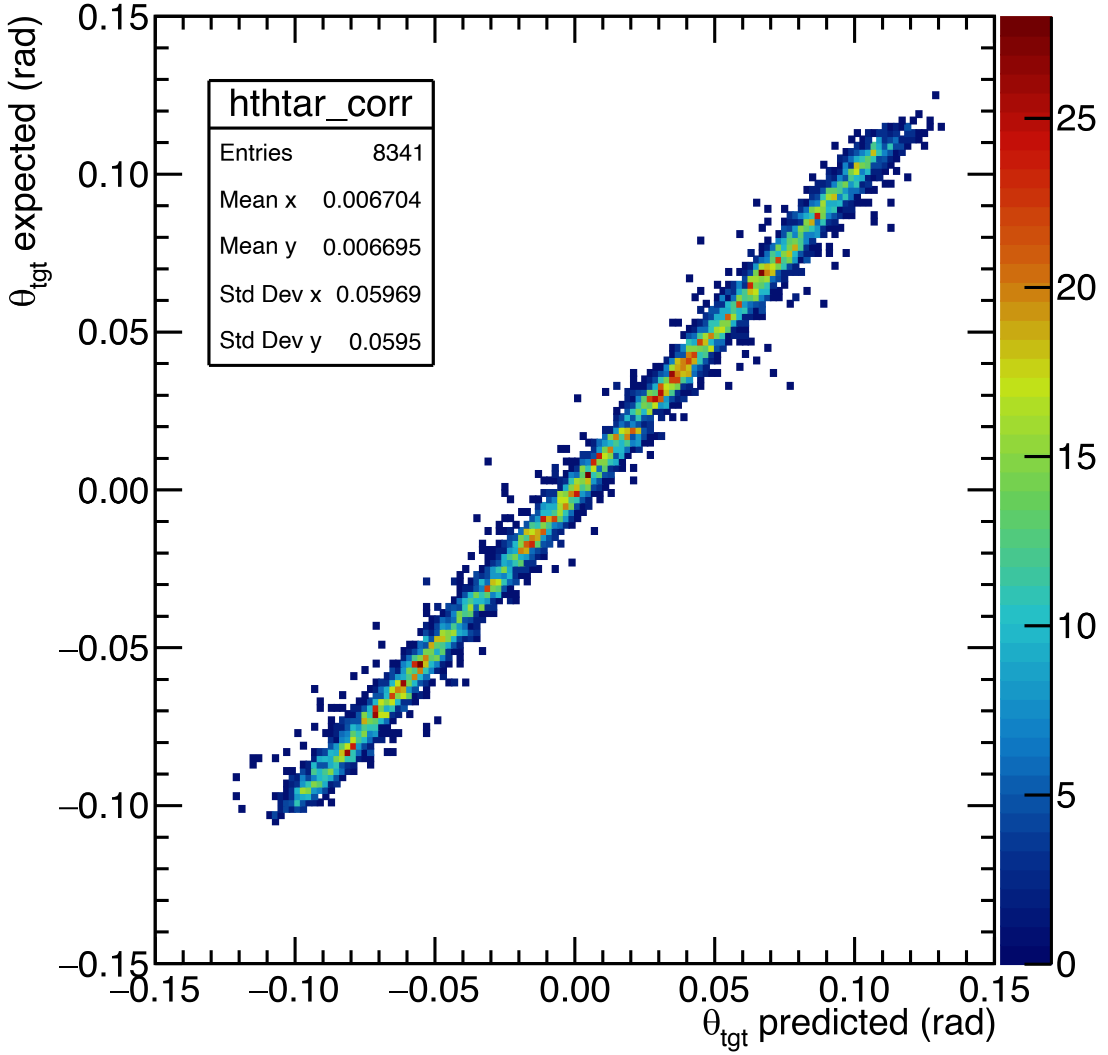
GEN-2 Invariant Mass (H2 Ref. Cell, SBS 30% and 100% field)

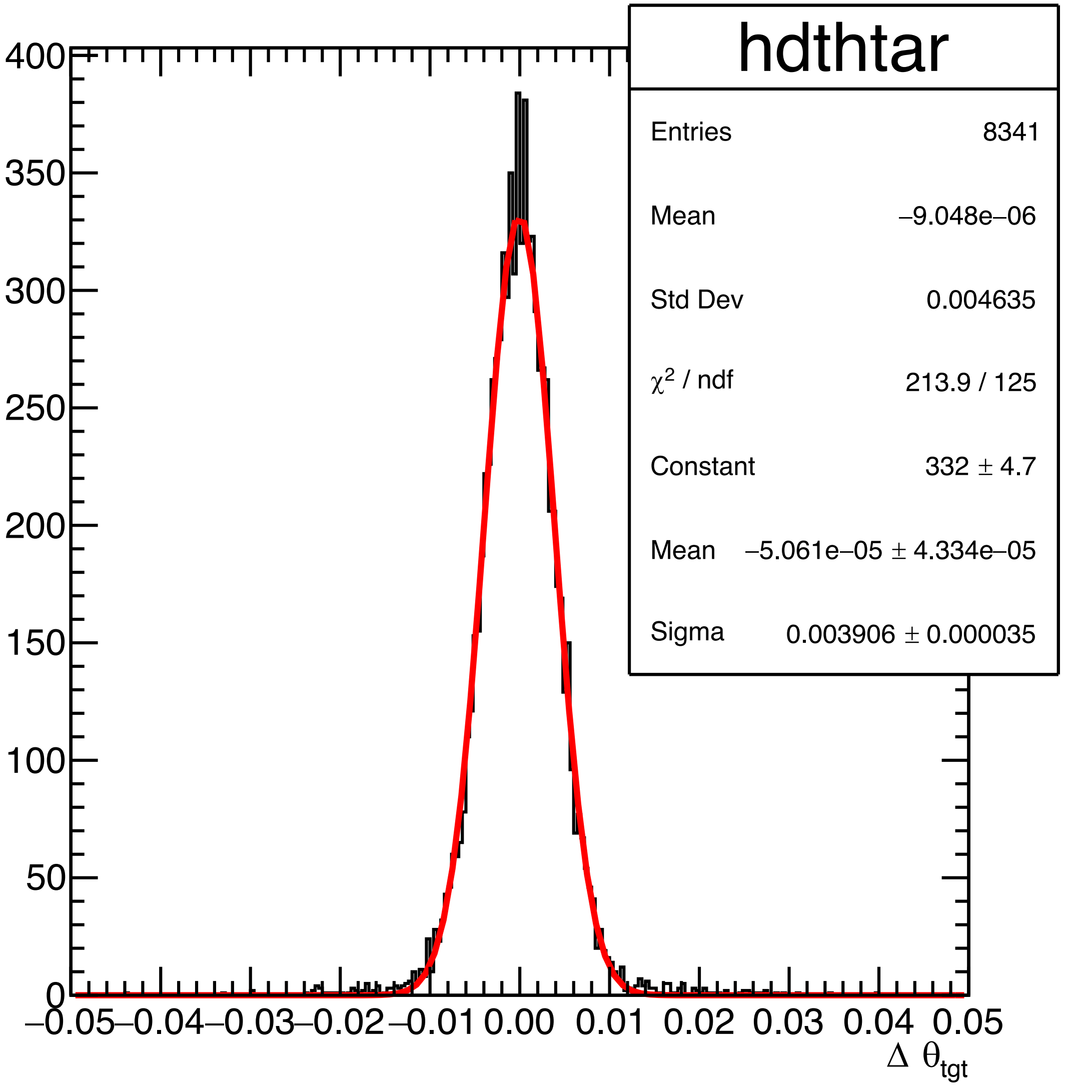




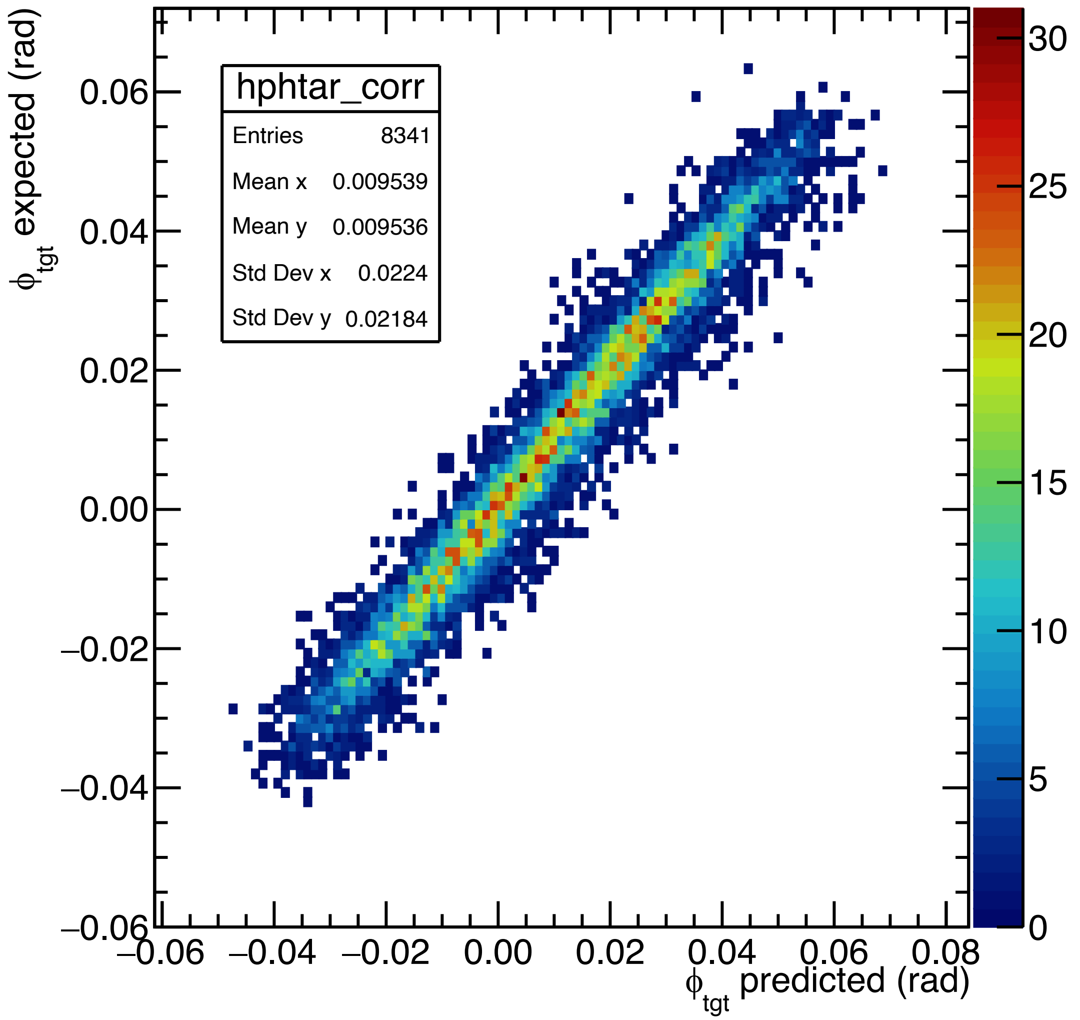
Background lower with 100% field due to more sweeping of inelastic particles

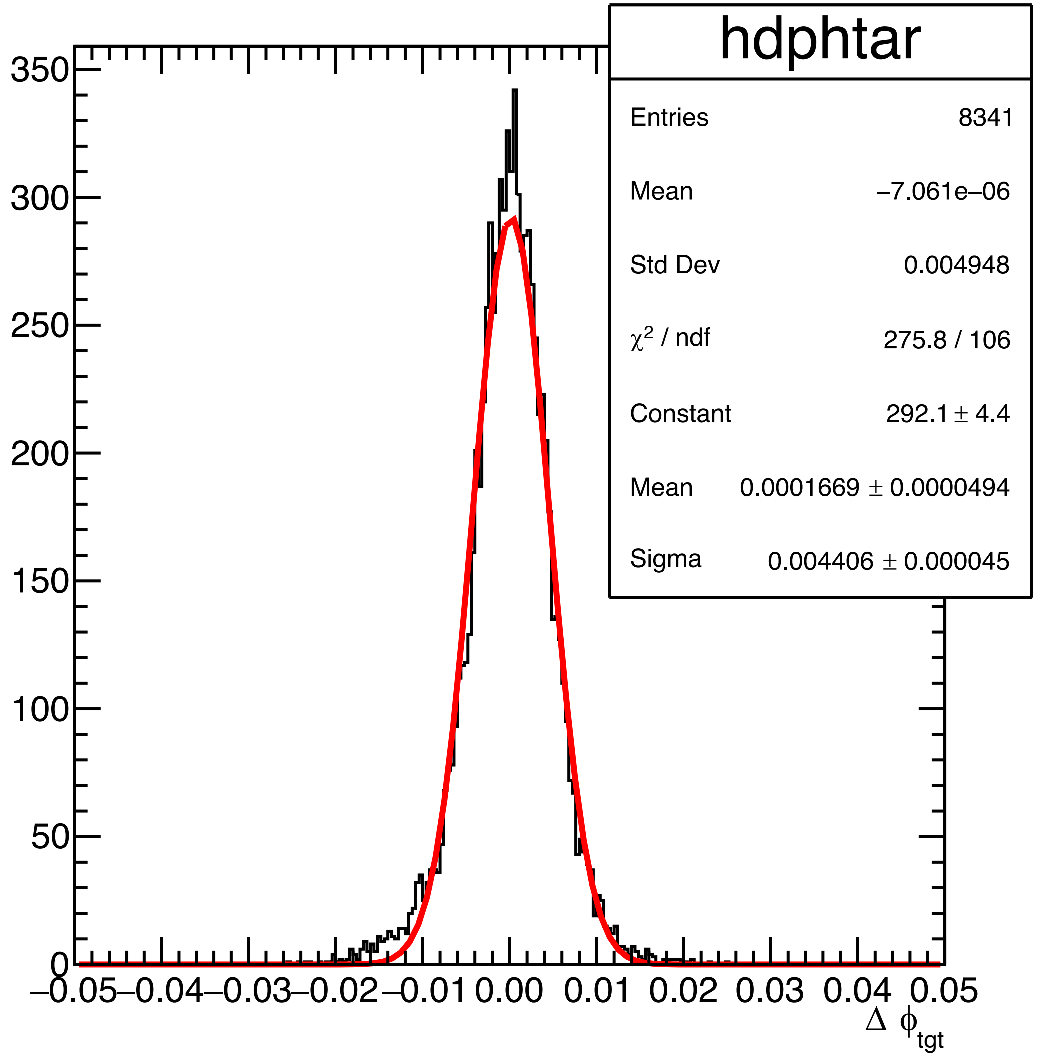
Target θ reconstruction (SBS 30% field)



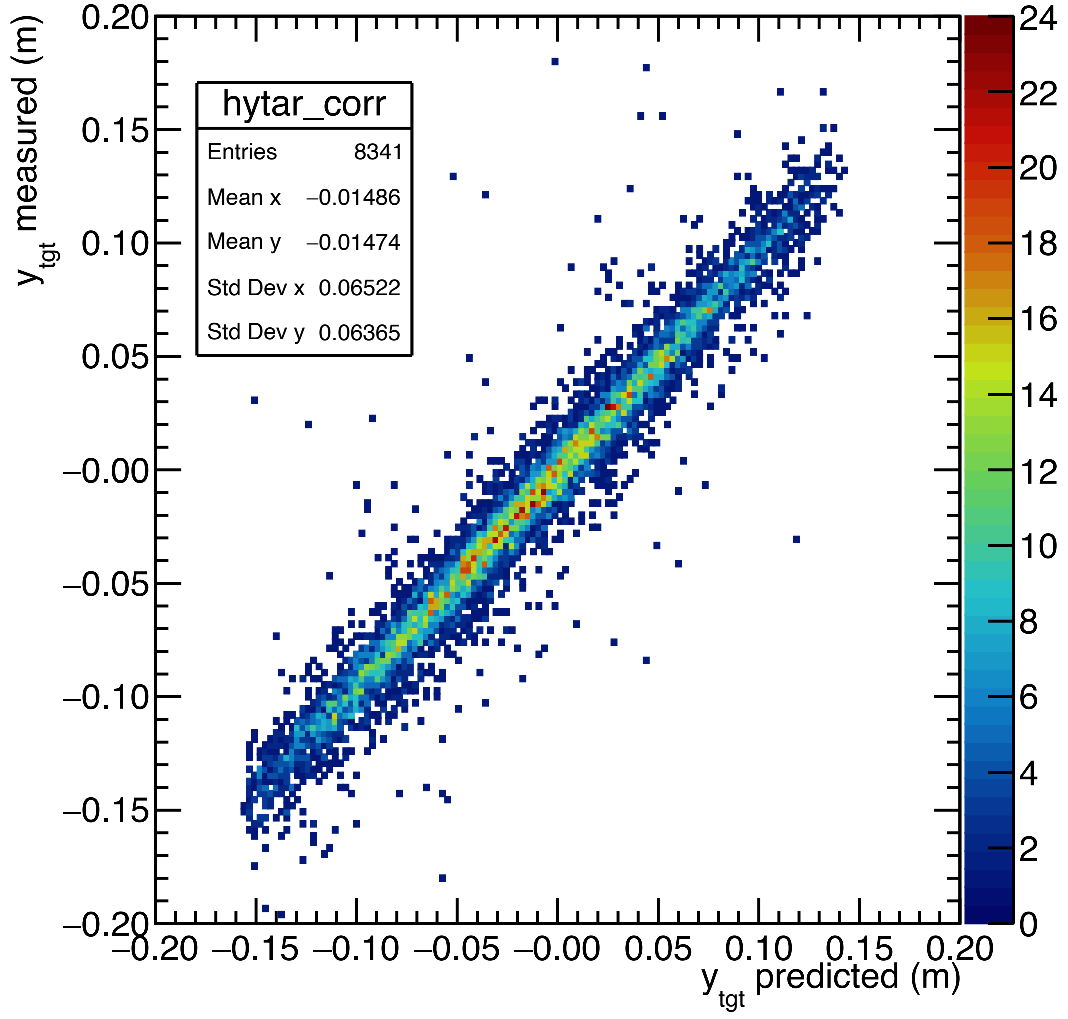


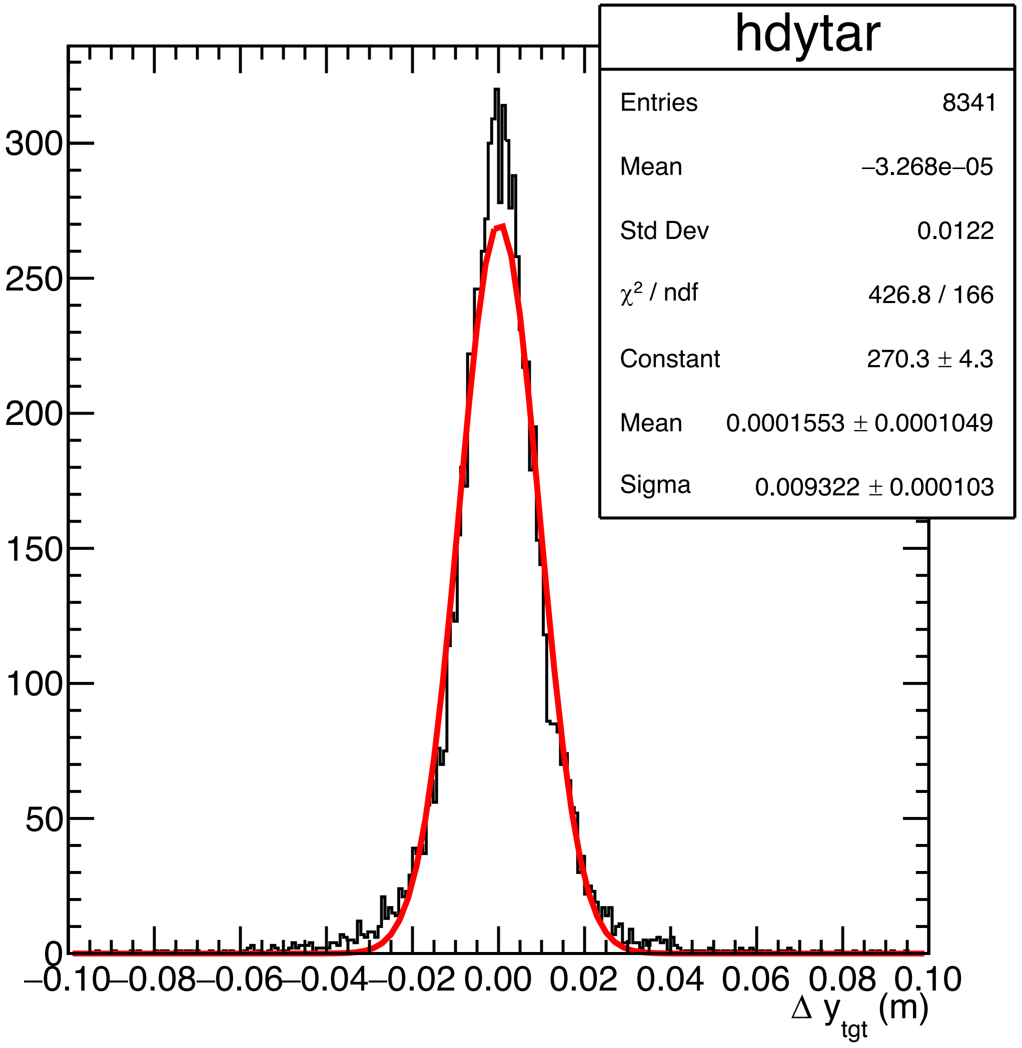
Target ϕ reconstruction (SBS 30% field)



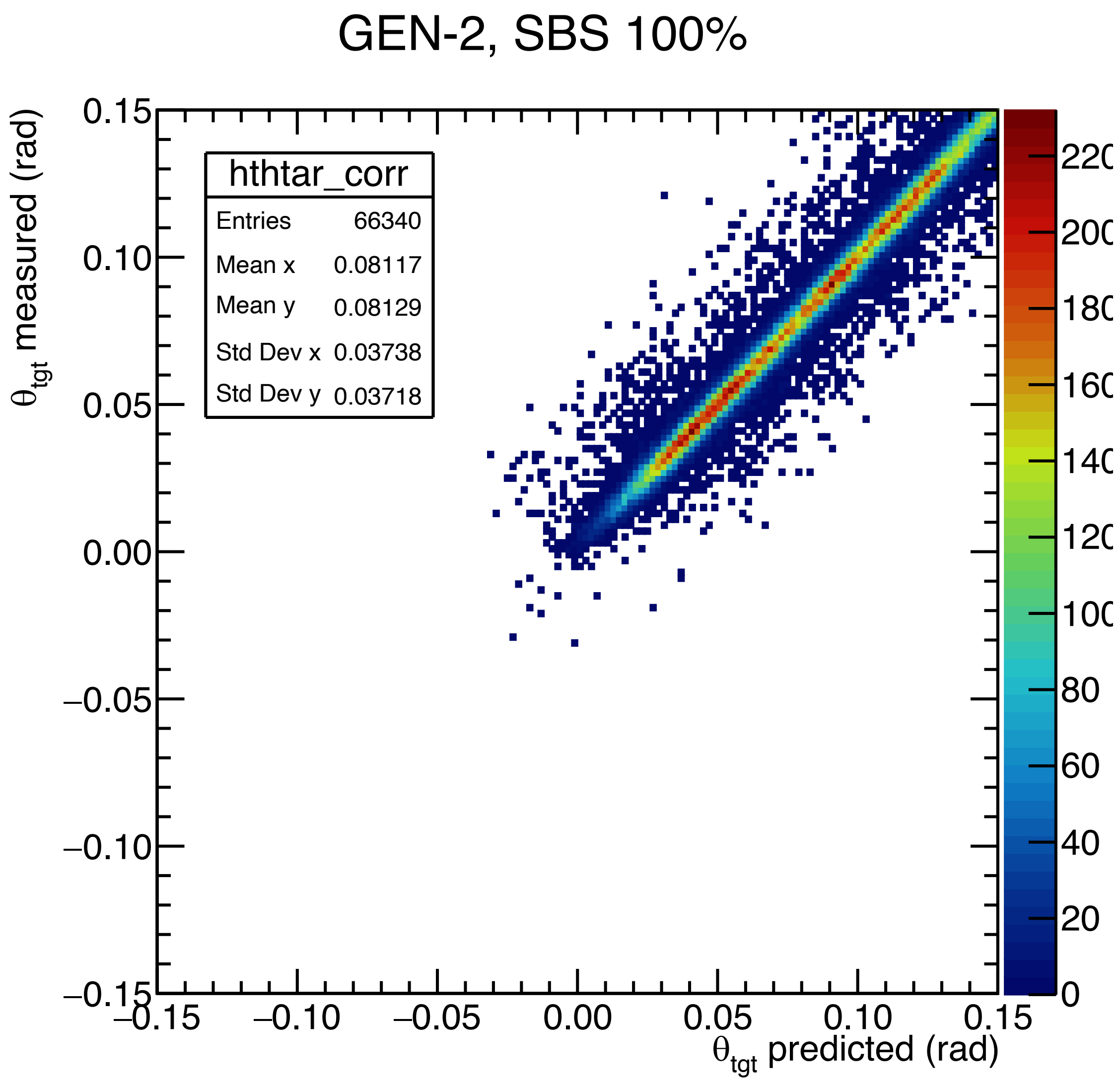


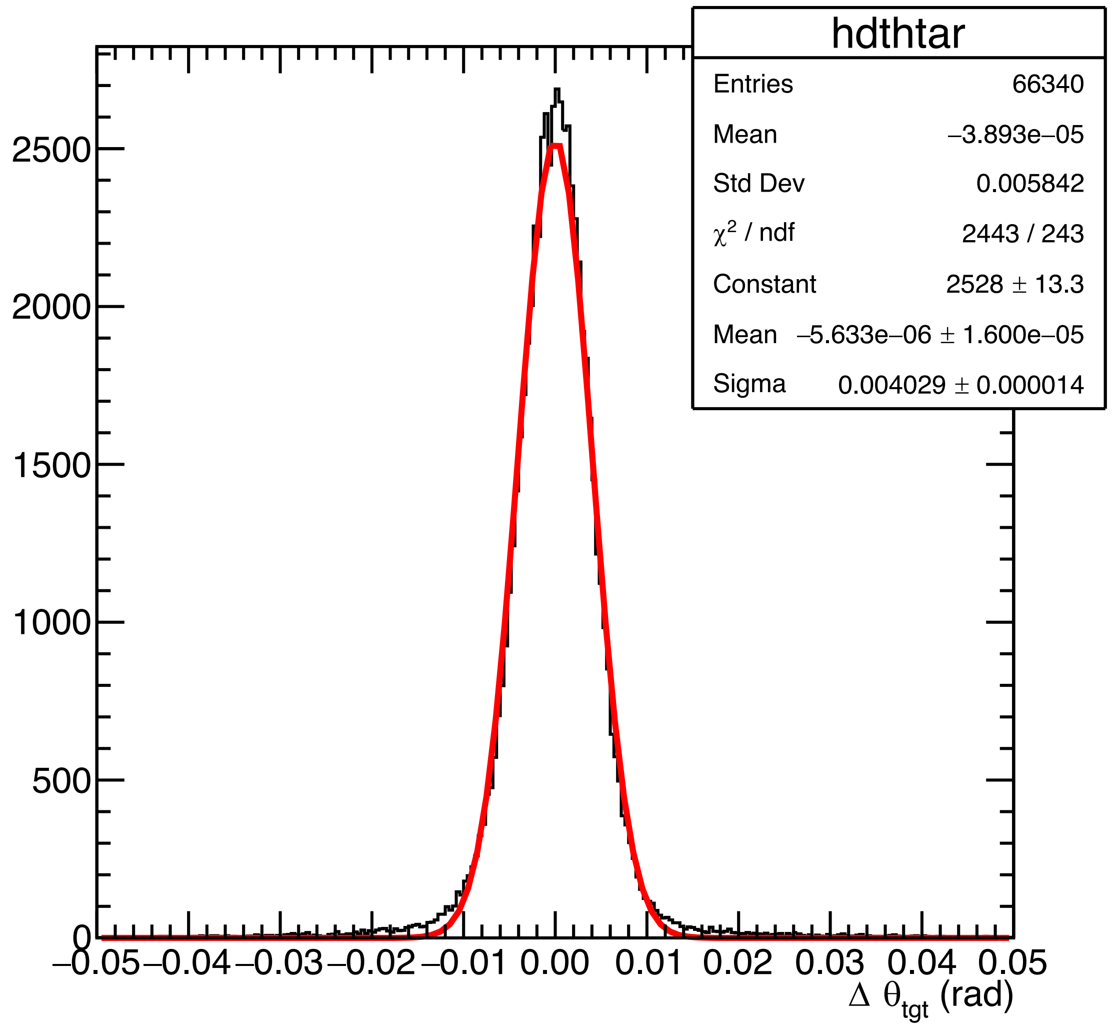
y target reconstruction (SBS 30% field)



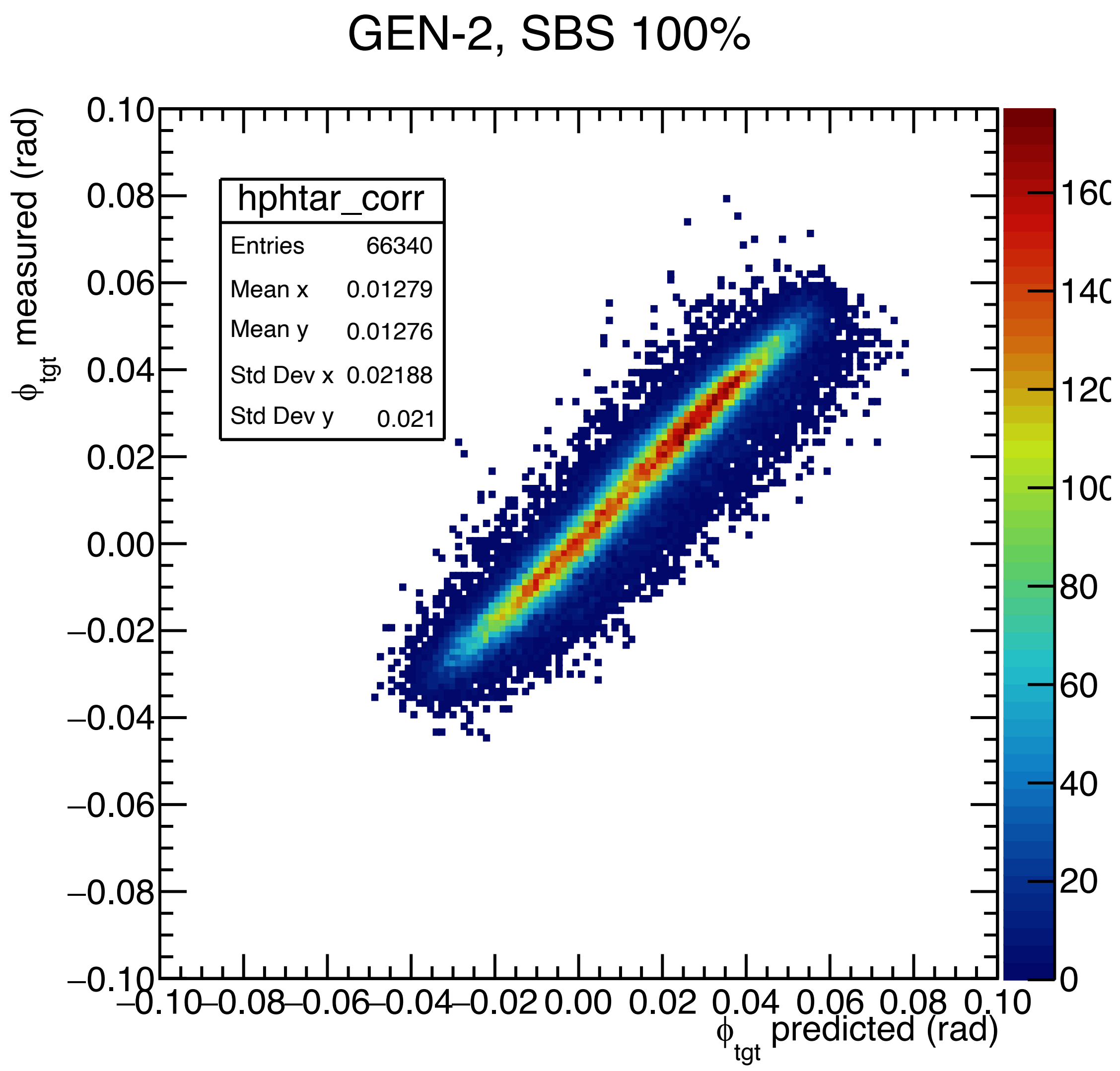


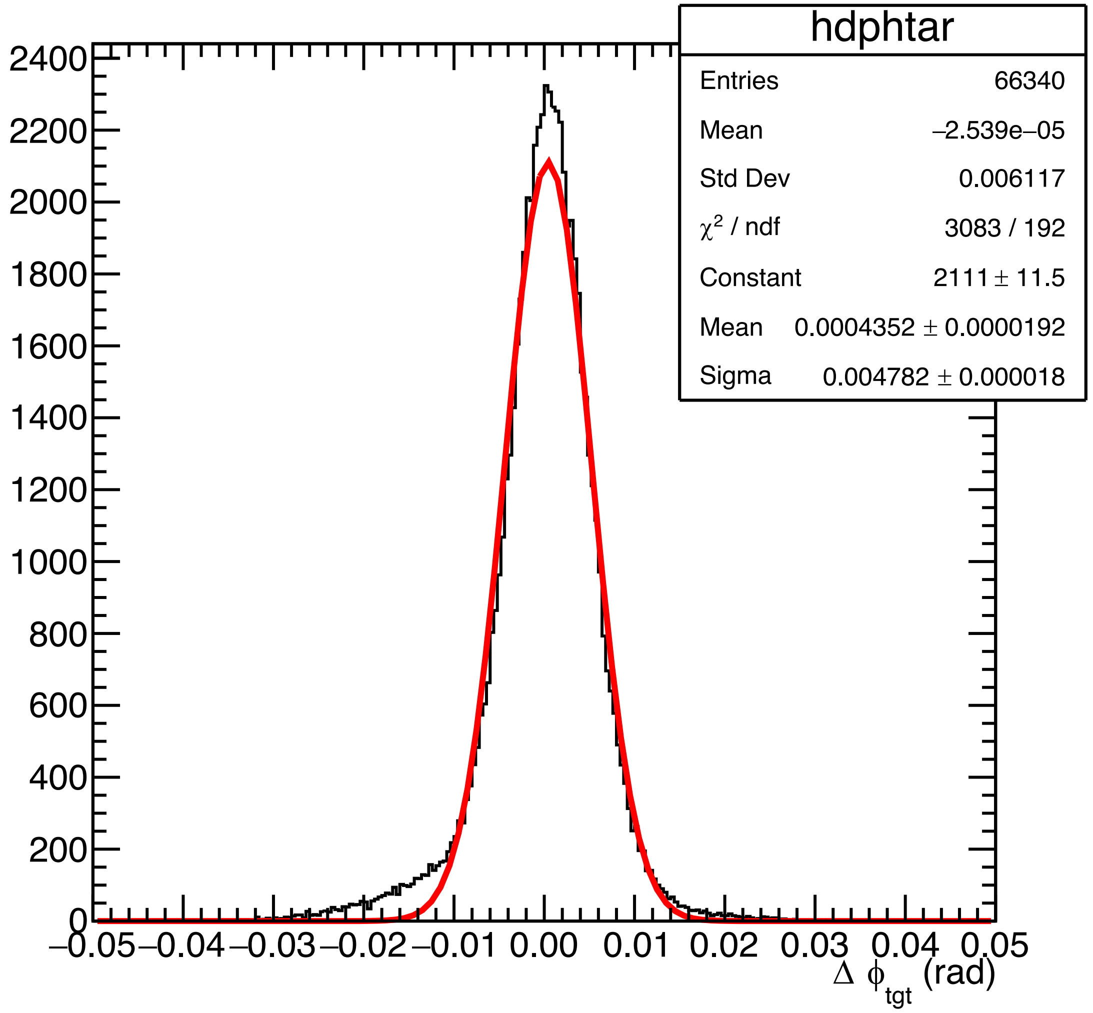
Target θ reconstruction (SBS 100% field)



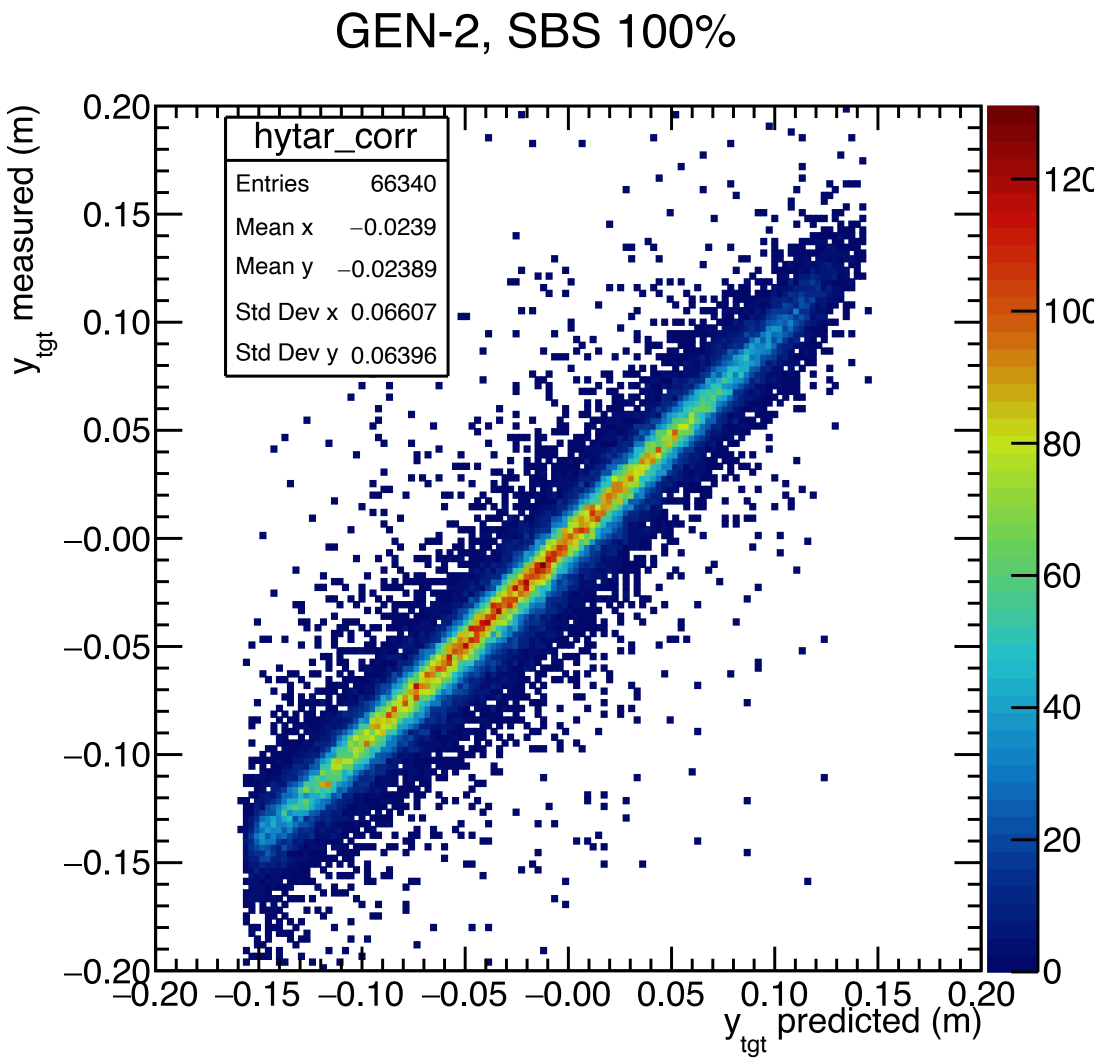


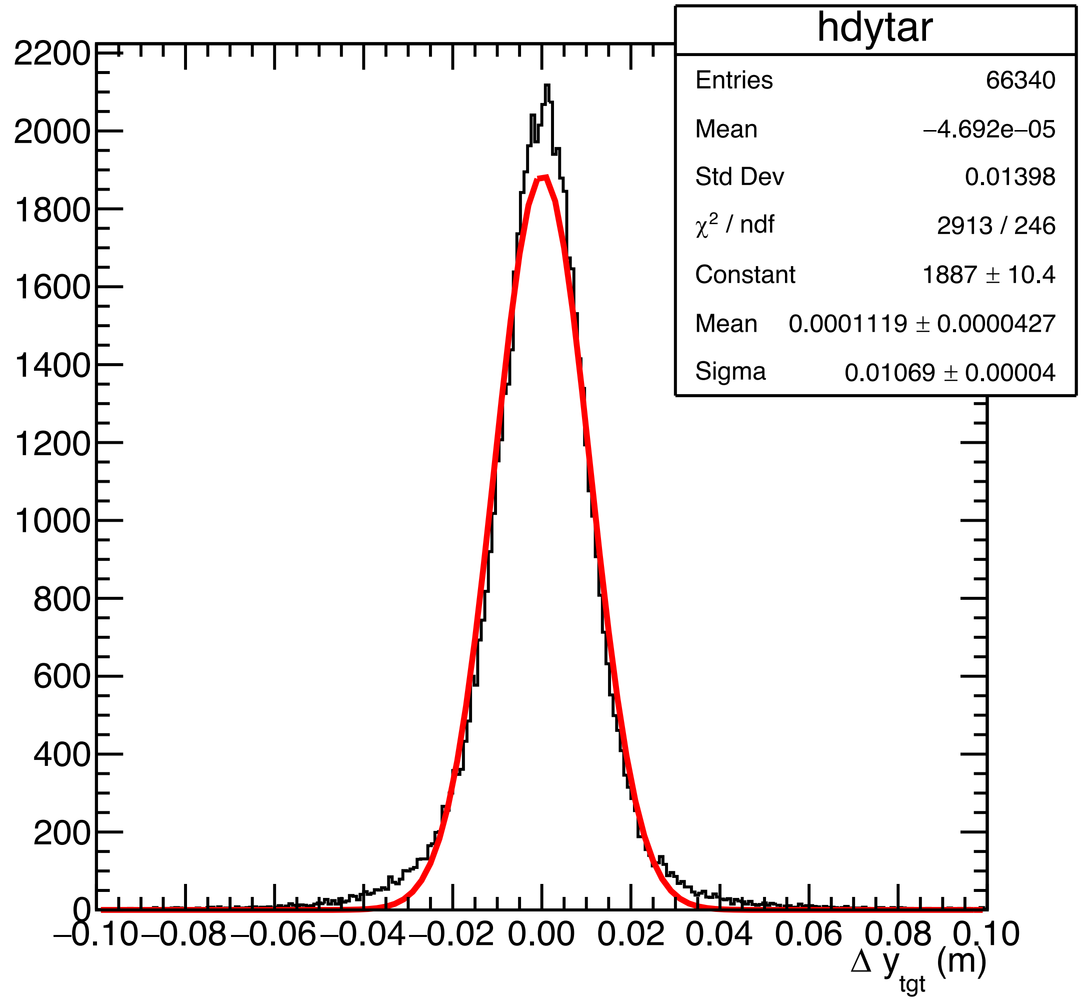
Target ϕ reconstruction (SBS 100% field)



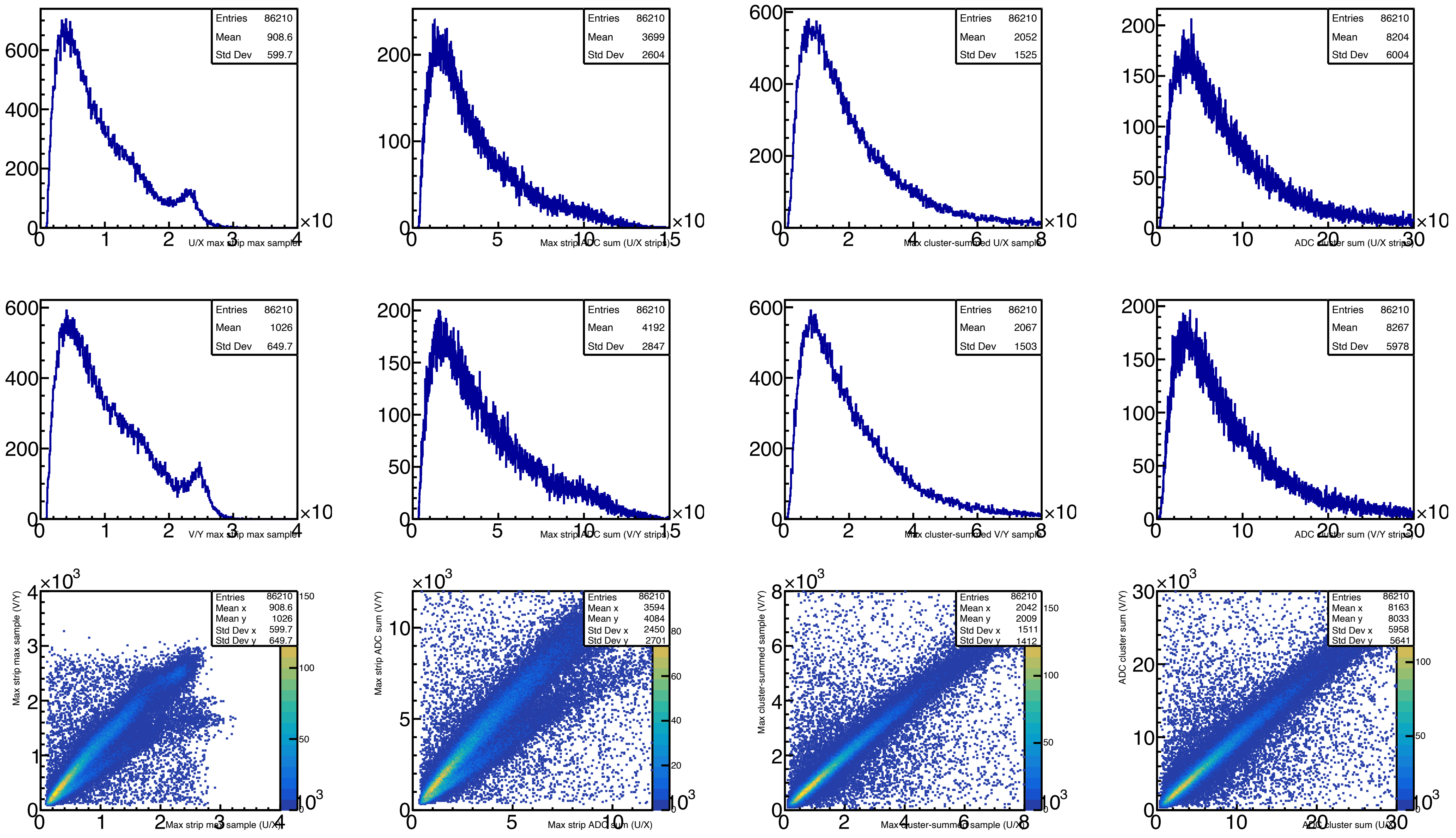


y target reconstruction (SBS 100% field)





Cluster ADC distributions and correlations: max sample, strip sum, max cluster-summed sample, total cluster sum


Entries 86210
Mean 908.6
Std Dev 599.7
U/X max strip max sample
Entries 86210
Mean 3699
Std Dev 2604
Max strip ADC sum (U/X strips)
Entries 86210
Mean 2052
Std Dev 1525
Max cluster-summed U/X sample
Entries 86210
Mean 8204
Std Dev 6004
ADC cluster sum (U/X strips)
Entries 86210
Mean 1026
Std Dev 649.7
V/Y max strip max sample
Entries 86210
Mean 4192
Std Dev 2847
Max strip ADC sum (V/Y strips)
Entries 86210
Mean 2067
Std Dev 1503
Max cluster-summed V/Y sample
Entries 86210
Mean 8267
Std Dev 5978
ADC cluster sum (V/Y strips)
Entries 86210
Mean x 908.6
Mean y 1026
Std Dev x 599.7
Std Dev y 649.7
Max strip max sample (V/Y)
Max strip max sample (U/X)
Entries 86210
Mean x 3594
Mean y 4084
Std Dev x 2450
Std Dev y 2701
Max strip ADC sum (V/Y)
Max strip ADC sum (U/X)
Entries 86210
Mean x 2042
Mean y 2009
Std Dev x 1511
Std Dev y 1412
Max cluster-summed sample (V/Y)
Max cluster-summed sample (U/X)
Entries 86210
Mean x 8163
Mean y 8033
Std Dev x 5958
Std Dev y 5641
ADC cluster sum (V/Y)
ADC cluster sum (U/X)