Advanced Methods for Roman Pots Reconstruction at the EIC

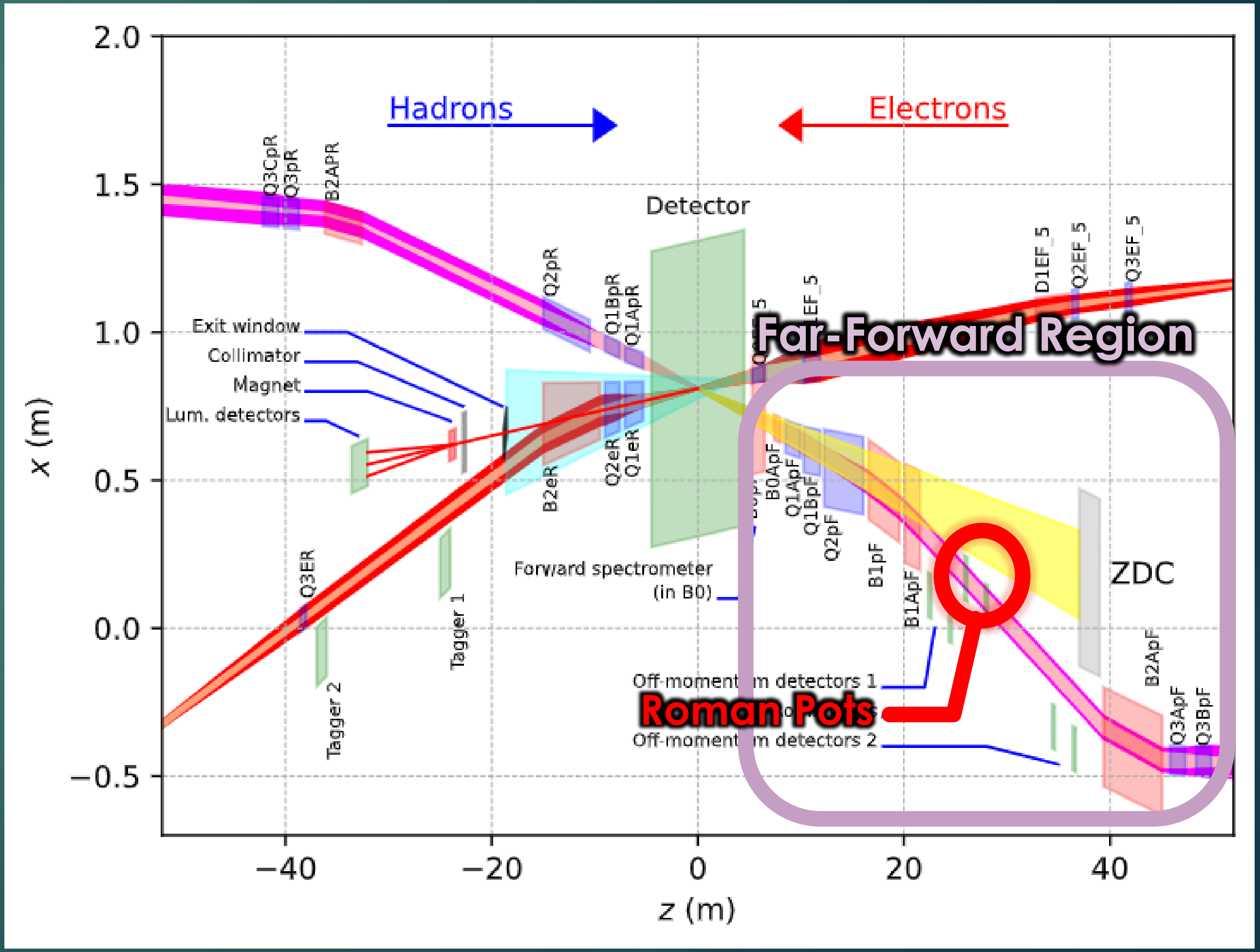
David Ruth, Alex Jentsch, Michael Murray
03/28/2023 – DIS2023

Acknowledgements

- A big thanks to all our collaborators on the EIC generic R&D Team!
- See Alex Jentsch's talk at the end of the session for more details on the Far-Forward detectors!
- Thank you to Jefferson Lab and JSA for providing travel funds to help me be at this conference

The EIC Physics Program Requires Detection and Reconstruction of Very Far-Forward Protons

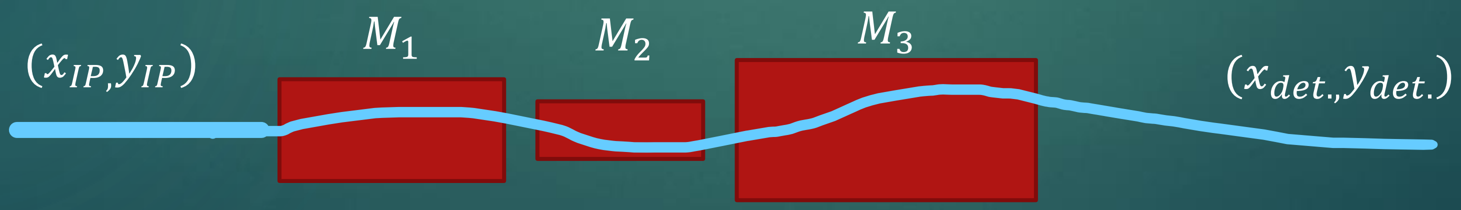
- Final State from many possible interactions:
  - e+p
  - e+d
  - e+He_3
  - e+A
- Roman Pots used to measure protons with momentum between 65% and 100% of the beam momentum
- Momentum is altered by the magnets between the main detector and the Roman Pots
- Necessary to reconstruct the scattered kinematics from the measured kinematics in the detector



Momentum Reconstruction is performed with Transfer Matrices

$$\begin{pmatrix} x_{ip} \\ \theta_{x,ip} \\ y_{ip} \\ \theta_{y,ip} \\ z_{ip} \\ \Delta p/p \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \\ c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \\ d_0 & d_1 & d_2 & d_3 & d_4 & d_5 \\ e_0 & e_1 & e_2 & e_3 & e_4 & e_5 \\ f_0 & f_1 & f_2 & f_3 & f_4 & f_5 \end{pmatrix} \begin{pmatrix} x_{det.} \\ \theta_{x,det.} \\ y_{det.} \\ \theta_{y,det.} \\ z_{det.} \\ \Delta p/p \end{pmatrix}$$

- Map each variable measured at the detector back to the variables at the IP
- Unique for different positions along the beam axis

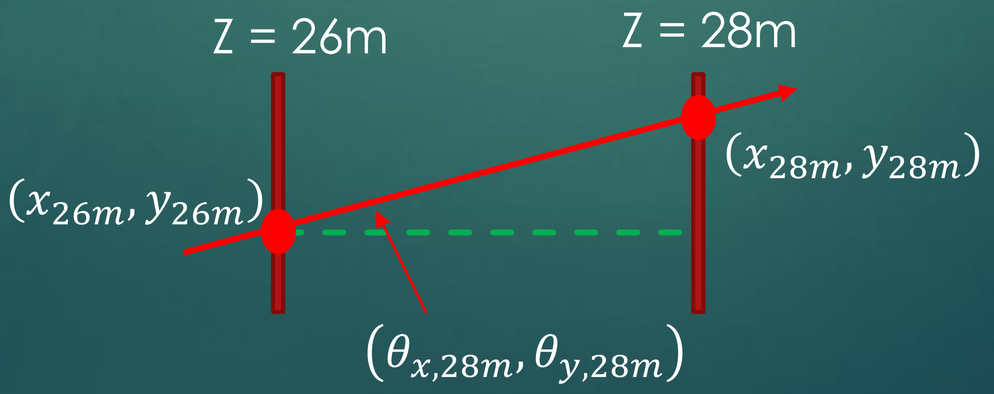


$$M_{transfer} = M_1 M_2 M_3 \ldots$$

For Central Momentum Track Protons, Matrix can be generated with BMAD

$$\begin{pmatrix} 1.88 & 28.97 & 0.0 & 0.0 & 0.0 & 0.25 \\ -0.0211 & 0.21 & 0.0 & 0.0 & 0.0 & -0.034 \\ 0.0 & 0.0 & -2.26 & 3.78 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.18 & -0.145 & 0.0 & 0.0 \\ 0.057 & 1.014 & 0.0 & 0.0 & 1.0 & 0.026 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix} \begin{pmatrix} x_{ip} \\ \theta_{xip} \\ y_{ip} \\ \theta_{yip} \\ z_{ip} \\ \Delta p/p \end{pmatrix} = \begin{pmatrix} x_{28m} \\ \theta_{x,28m} \\ y_{28m} \\ \theta_{y28m} \\ z_{28m} \\ \Delta p/p \end{pmatrix}$$

- For DVCS, this works well, but protons can have large deviations from central orbit momentum
- <u>Need different matrix for other momentum fractions!</u>

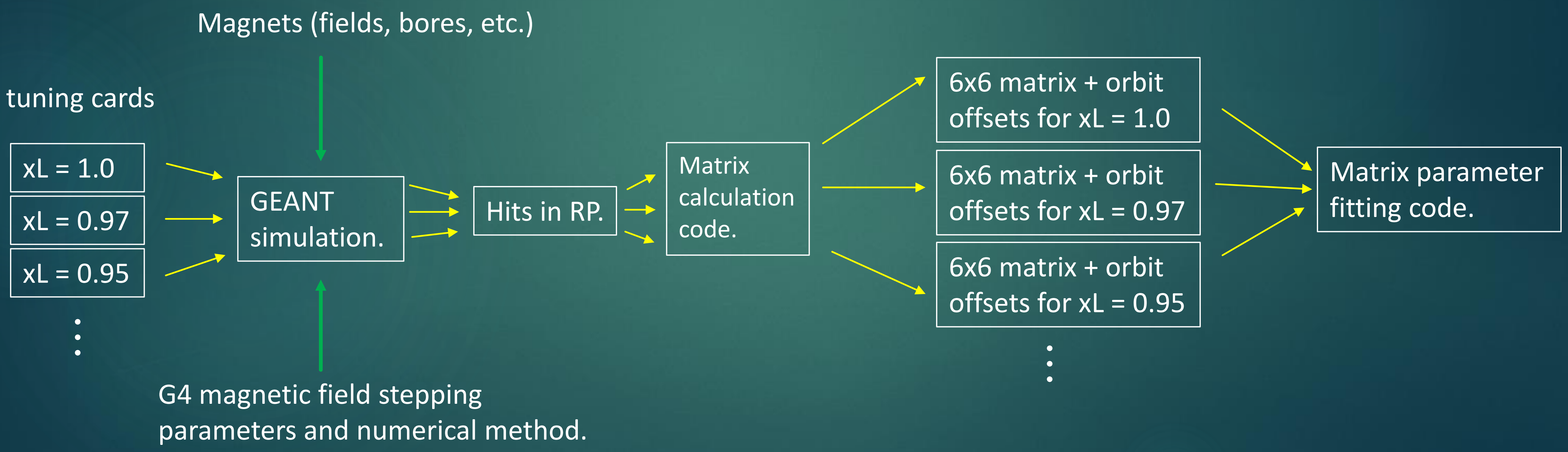


Current Solution: Use an input 'tuning card' for each Momentum Fraction

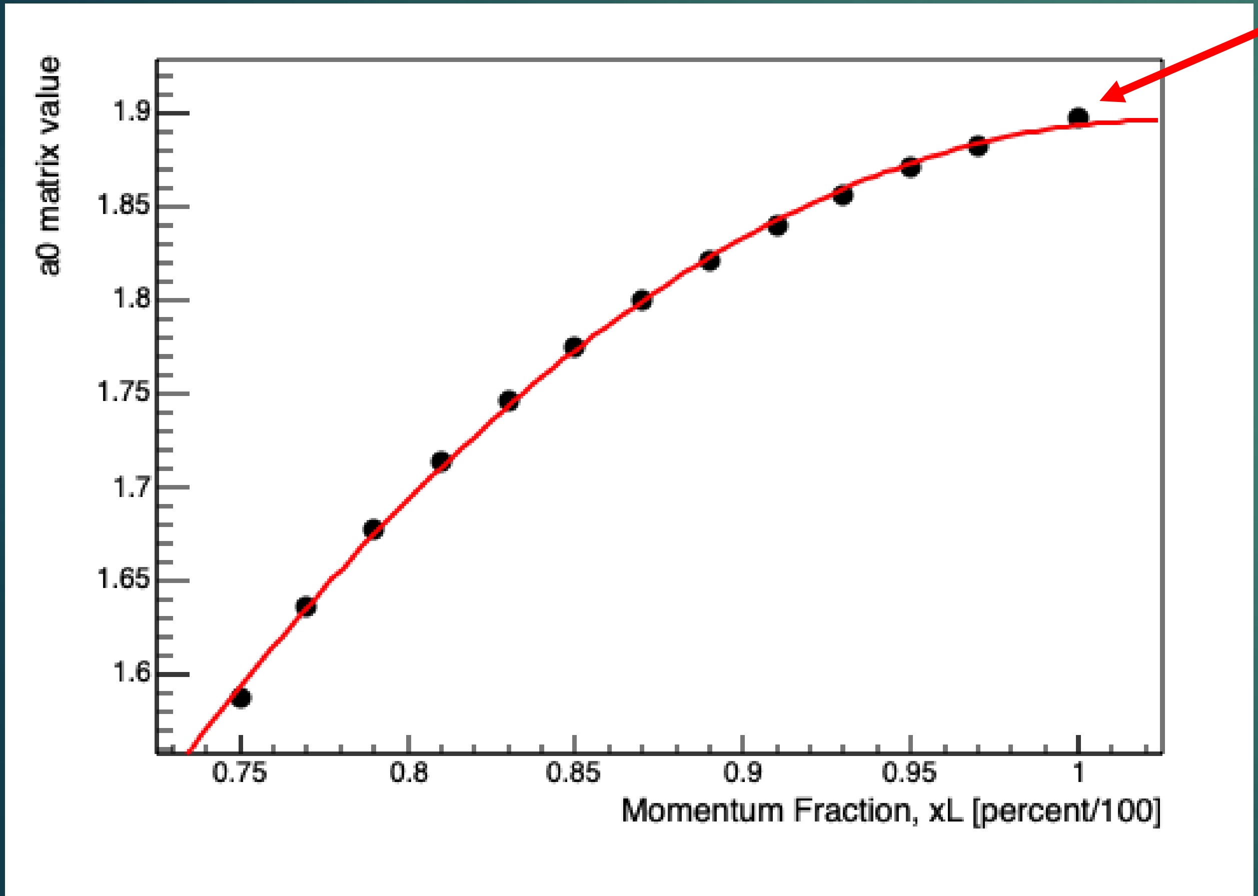
longitudinal momentum fraction

$$x_L = \frac{p_{z,proton}}{p_{z,beam}}$$

(For a 275 GeV beam, a 270 GeV proton has an x_L of 0.98.)



Fit Each Matrix value to interpolate to other Momentum Fractions

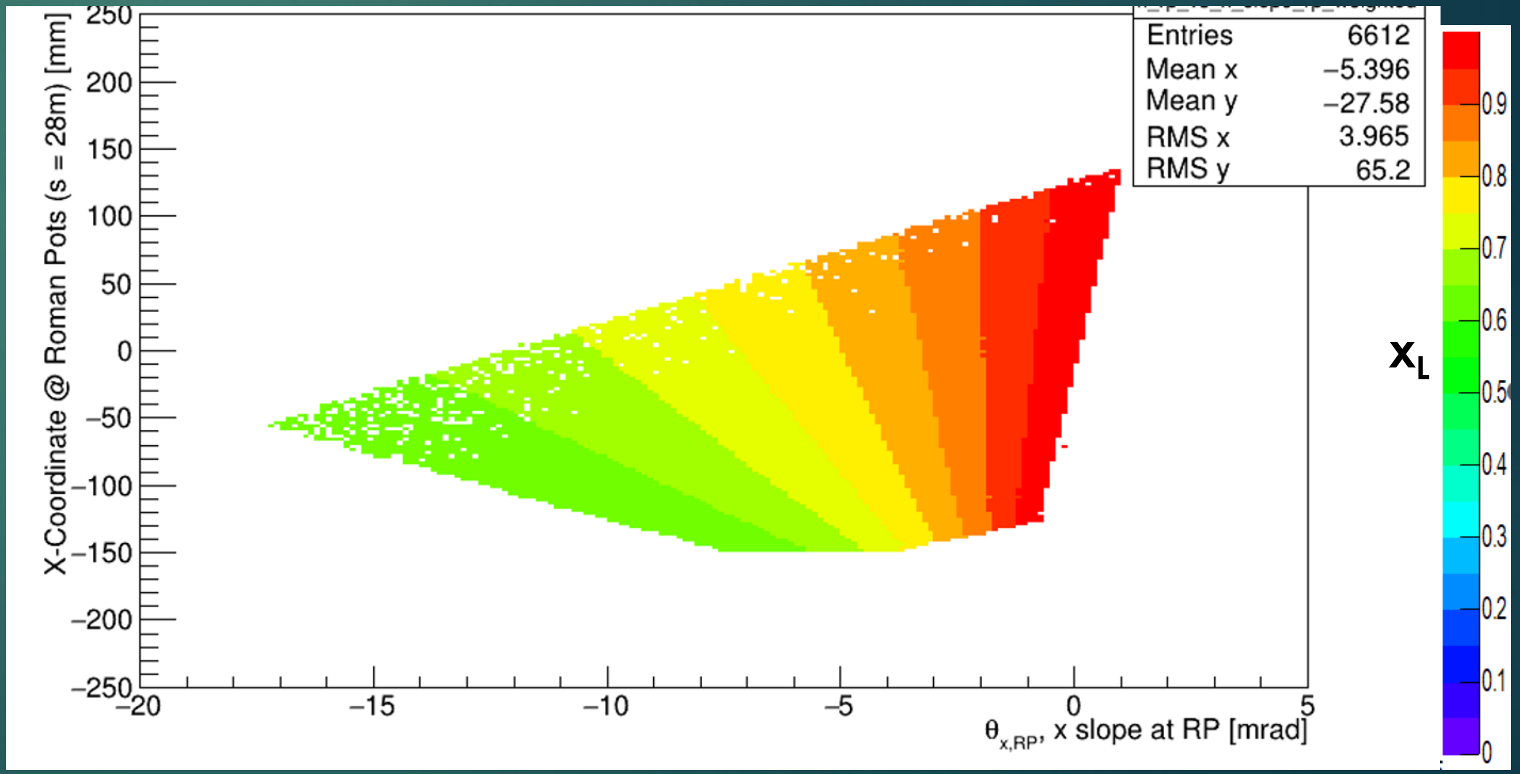


$$\begin{pmatrix} 1.88481537 & 28.96766544 & 0.0000 & 0.0000 & 0.0000 & 0.24906255 \\ -0.02114673 & 0.20555261 & 0.0000 & 0.0000 & 0.0000 & -0.03322467 \\ 0.0000 & 0.0000 & -2.25541901 & 3.78031509 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.17782524 & -0.14532313 & 0.0000 & 0.0000 \\ 0.05735551 & 1.01363652 & 0.0000 & 0.0000 & 1.0000 & 0.02568709 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{pmatrix}$$

- Plot each of the 36 matrix values as a function of xL
- Fit each of these plots with a quadratic polynomial to interpolate
- Now we just need to extract xL to generate the reconstruction matrix

Lookup x_L at z = 28m on the beamline with (θ_x, x) for the Roman Pots

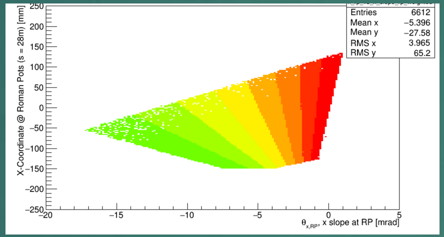
- "Chromaticity Plot" used as a lookup-table to extract x_L for a given set of detector x and θ_x
- Produced by running the simulation with no reconstruction matrix
- We can then use x_L with the fits to generate the matrix entries



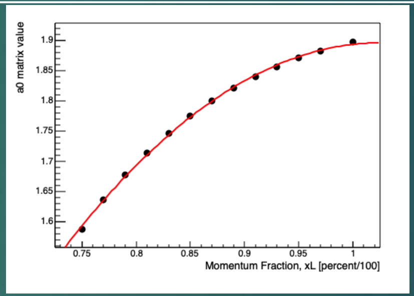
Now we can use the coordinates measured at the Roman Pots to reconstruct the scattered momentum vector

Detector "hit" coordinates

Lookup x_L



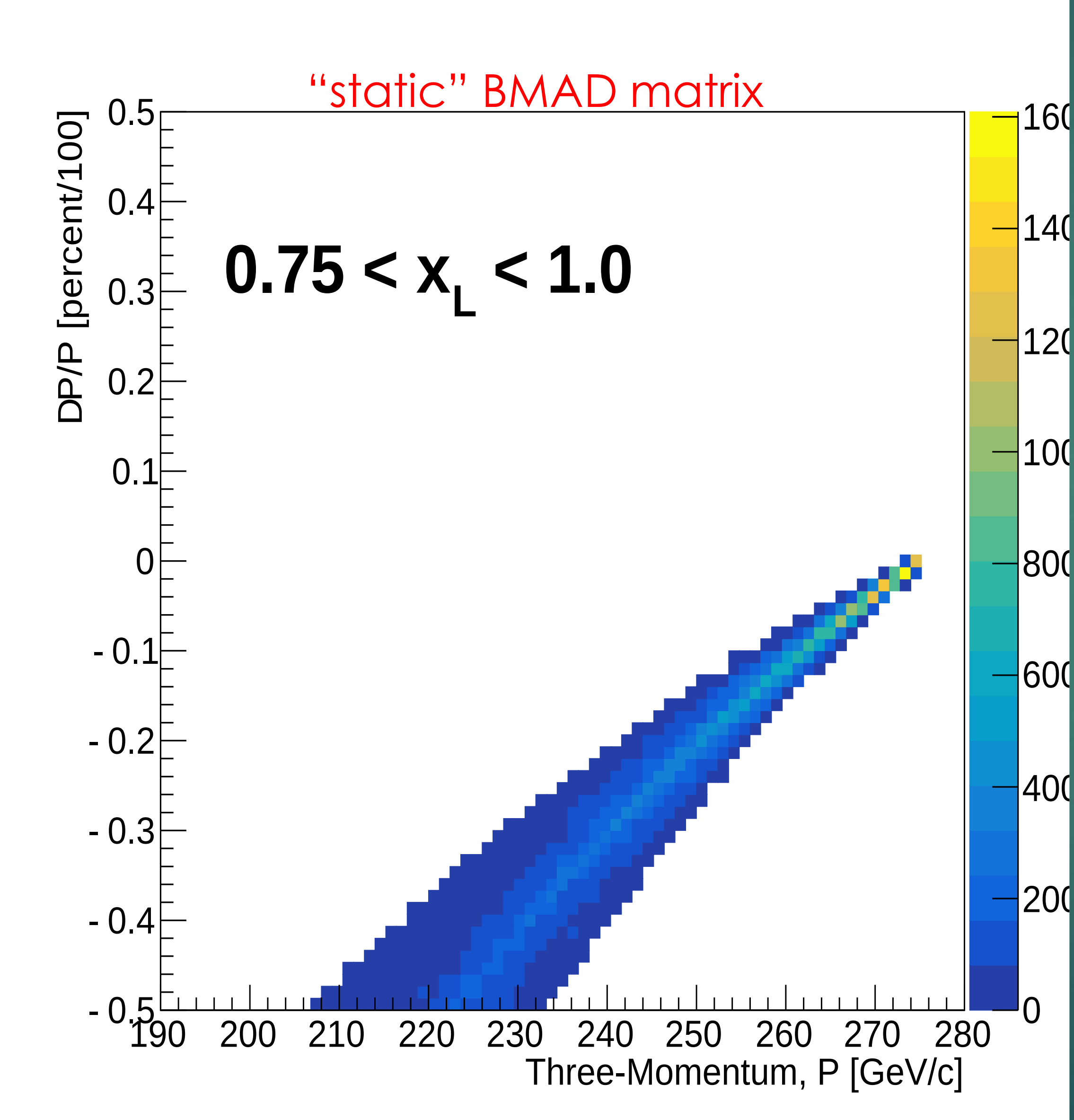
Calculate matrix parameters and offsets from fit equations.

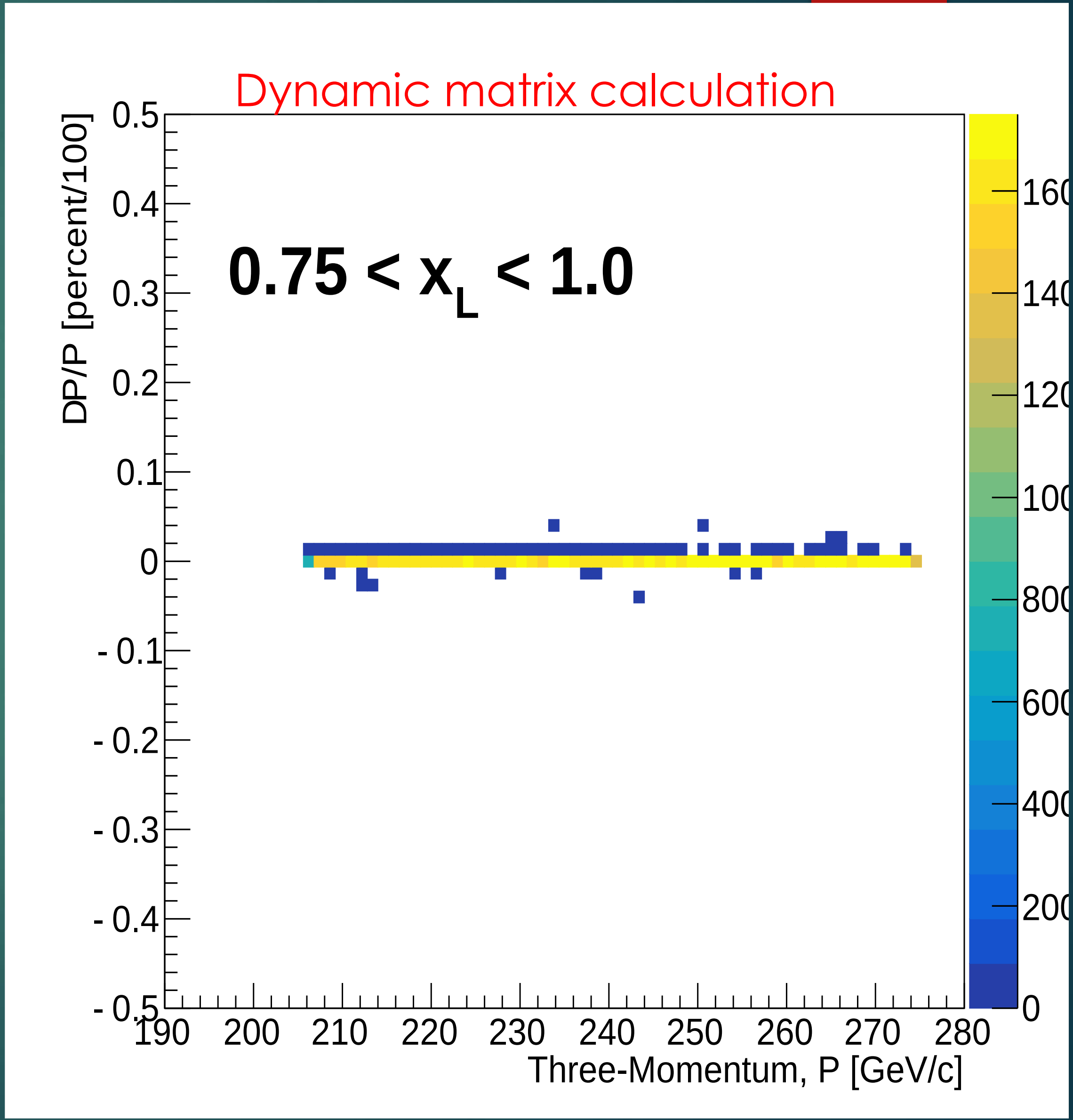


$$\begin{pmatrix} 1.88481537 & 28.96766544 & 0.0000 & 0.0000 & 0.0000 & 0.24906255 \\ -0.02114673 & 0.20555261 & 0.0000 & 0.0000 & 0.0000 & -0.03322467 \\ 0.0000 & 0.0000 & -2.25541901 & 3.78031509 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.17782524 & -0.14532313 & 0.0000 & 0.0000 \\ 0.05735551 & 1.01363652 & 0.0000 & 0.0000 & 1.0000 & 0.02568709 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{pmatrix}$$

Reconstructed momentum vector.

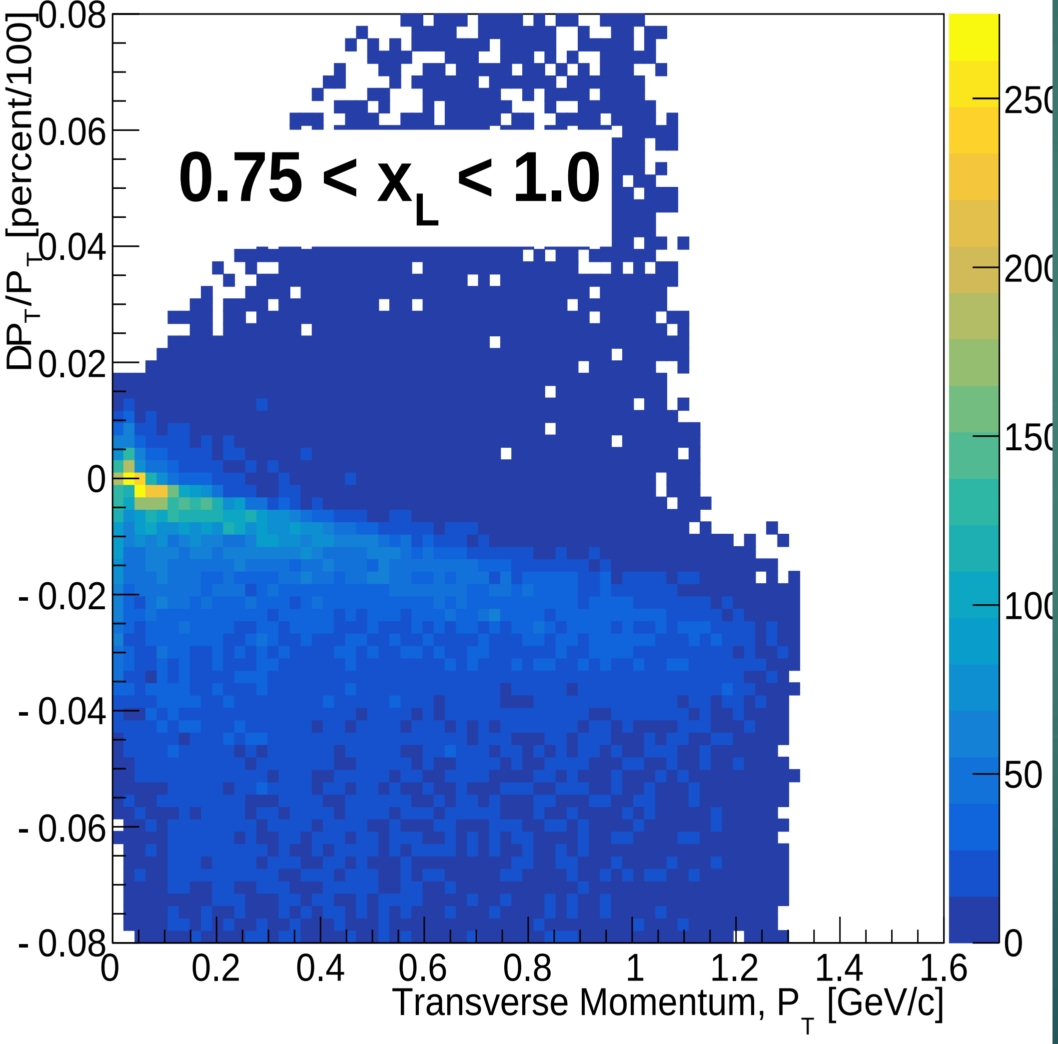
Results – Longitudinal Momentum



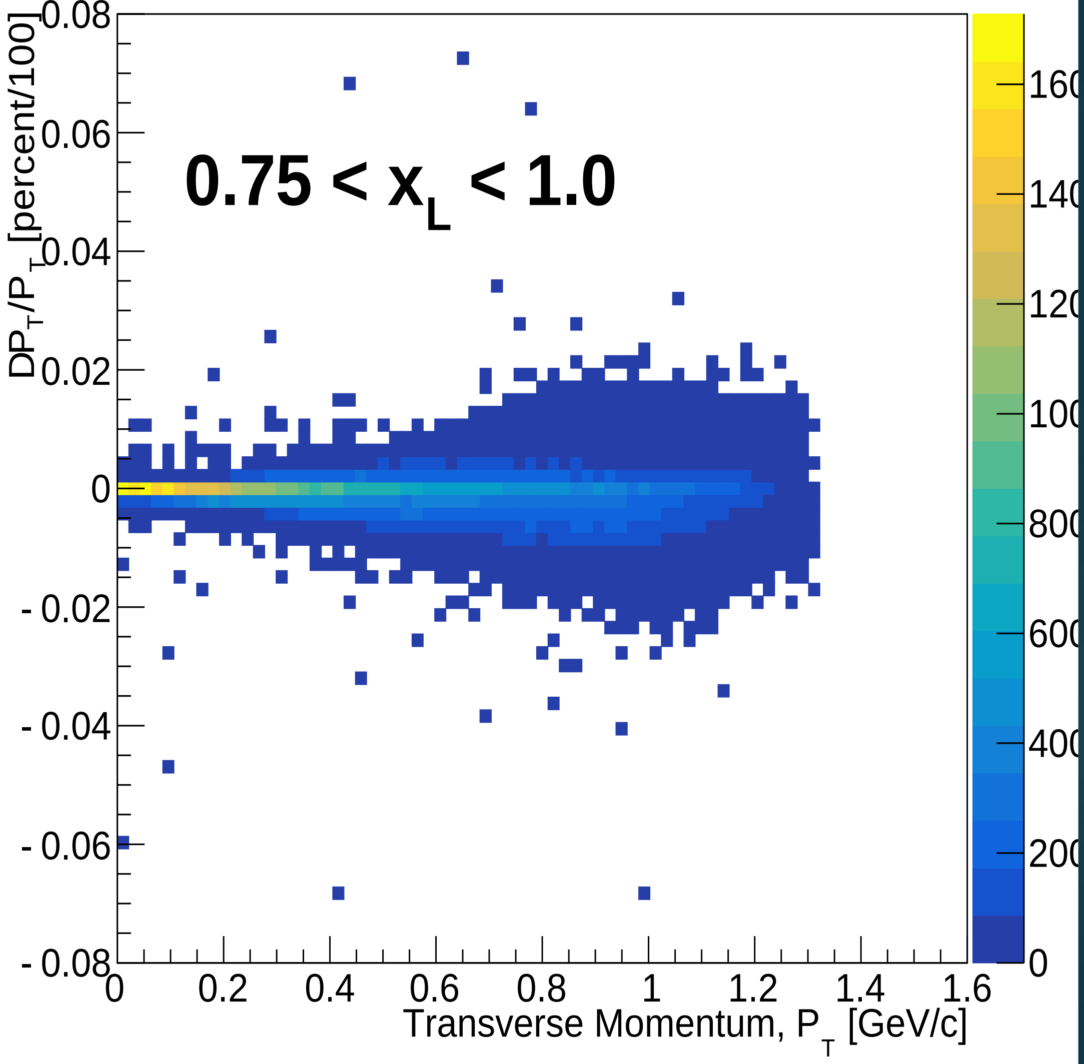


Results – Transverse Momentum

"static" BMAD matrix



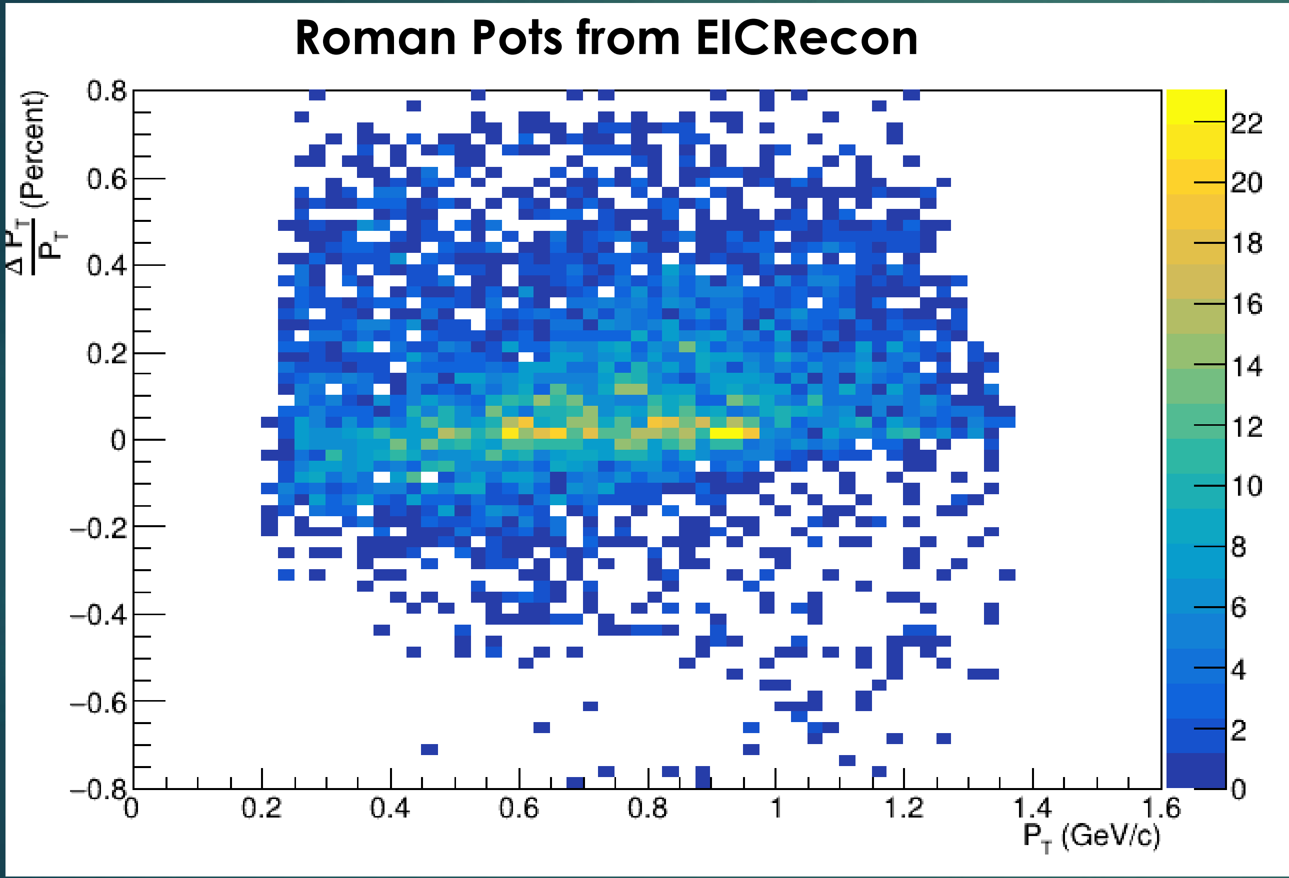
Dynamic matrix calculation



Current method has a number of drawbacks:

- Final solution dependent on initial choice of test trajectories (or 'tuning cards')
- Current study only goes down to x_L of around 0.75, doesn't capture momenta further from central orbit
- Assumption of linearity necessary for current approach
- Current approach will fail for more complicated interactions where the tagged particle is not coming directly from IP
- Needs to be completely redone for detectors at different points on the beamline
- Study would need to be completely redone in response to updates in the detectors or magnets

Current work: integration with EICRecon



- Static Matrix method recently added to EICrecon!
- Currently, work is being done on adding the dynamic method into EICrecon
- Investigating ongoing magnetic field transport issue in DD4HEP which greatly affects proton's motion through the magnets

Currently developing a more modern approach using Machine-Learning

- "Train" a machine learning model with a similar method to the dynamic approach
- ML Algorithm creates reconstruction matrix for any Roman Pots momentum vector
- Neural Network / Graph Neural Network

PyTorch

- Already included in ePIC detector framework
- Potentially quicker to train and retrain models than other frameworks
- C++ API ideal for working with ROOT and ePIC simulations
- PyTorch 2.0 just released with major performance improvements
- Large support ecosystem
- Ideal for quick prototyping of models, "research" focused option

Next Steps: Finish integrating existing method with ePIC framework, develop ML Algorithm

- Ongoing work to integrate the dynamic method described above with EICRecon and the ePIC framework
- Further development beginning now to try and develop a graph neural network with PyTorch
- We hope this machine-learning algorithm will compensate for the current issues with the dynamic method!

Backup Slide: Off Momentum-Detectors EICRecon Integration