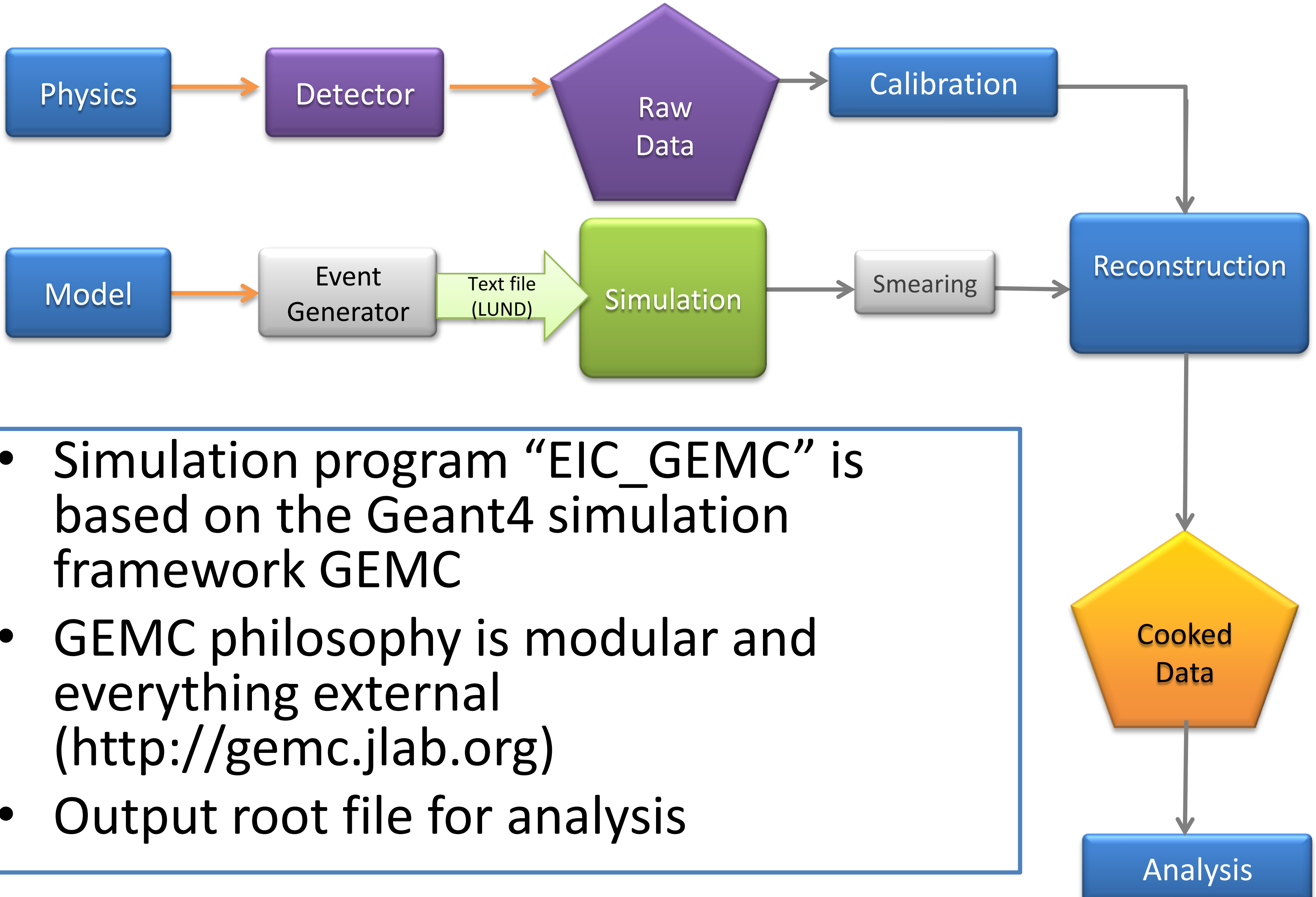
JLEIC Detector Simulation
Forward Ion Detection

Zhiwen Zhao

2016/10/28

Duke
UNIVERSITY

Data Flow



- Simulation program “EIC_GEMC” is based on the Geant4 simulation framework GEMC
- GEMC philosophy is modular and everything external (http://gemc.jlab.org)
- Output root file for analysis

Event Generation

- Independent
- Text file in LUND format, providing kinematics and particle 4 vectors
- an example

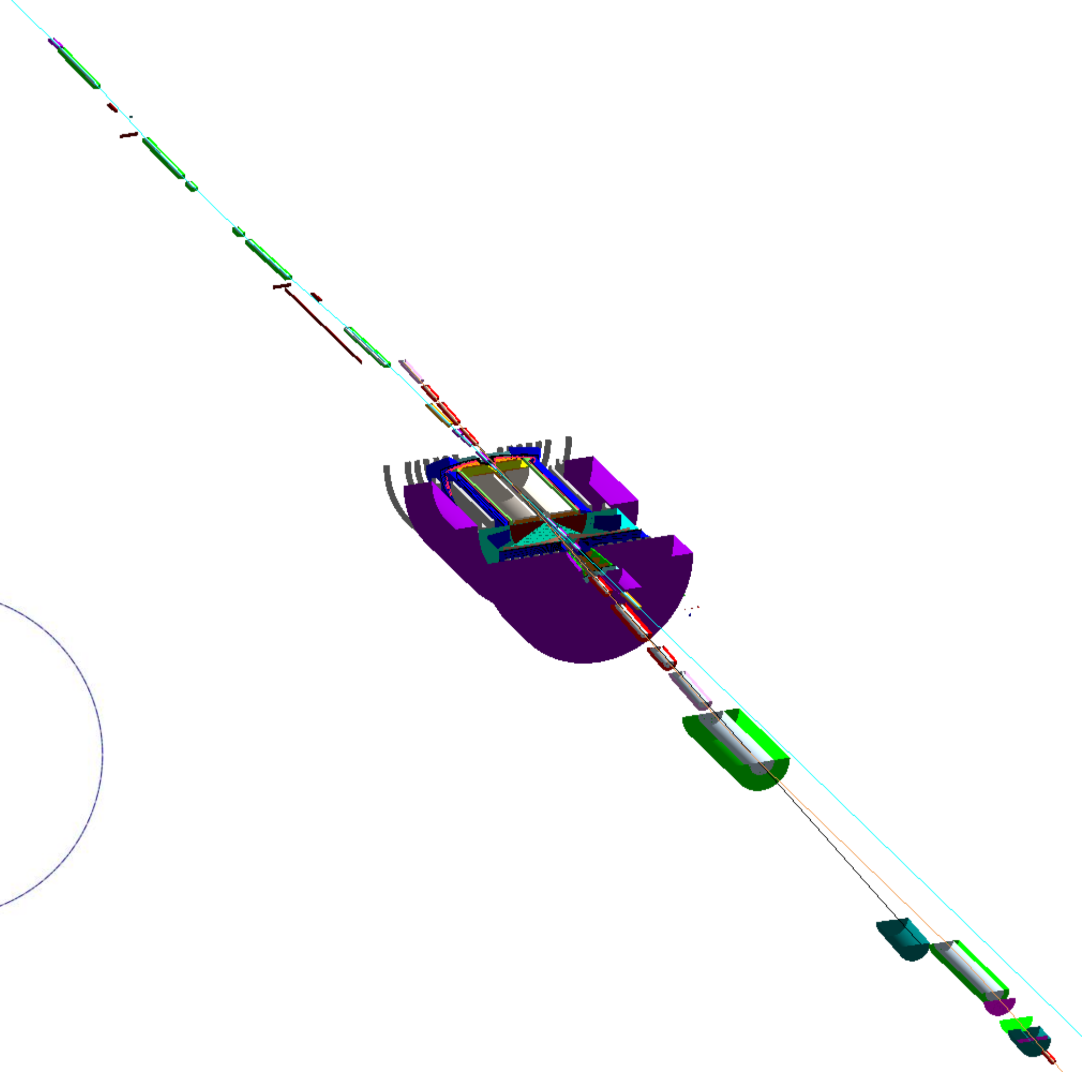
Header	2 1. 1. 0 -1 0.209 0.336 6.373 1.448 -1.000
Particle 1	1 -1. 1 11 0 0 0.9636 -0.1675 7.2357 7.3015 0.0005 0.0000 0.0000 0.0000
Particle 2	2 1. 1 2212 0 0 -0.6536 0.0604 0.3367 1.1935 0.9383 0.0000 0.0000 0.0000

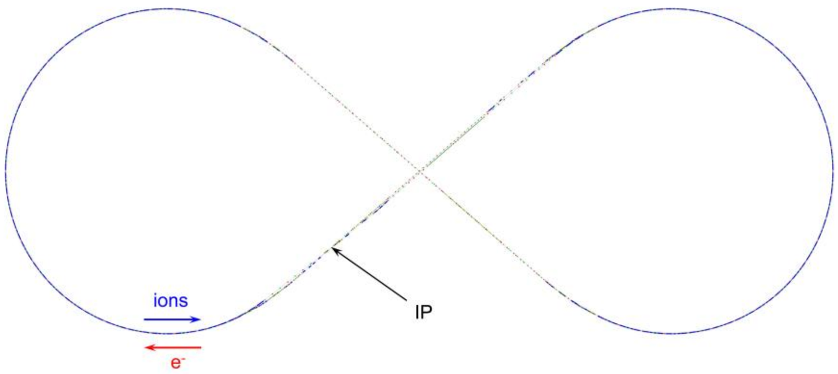
EIC event generators we have so far

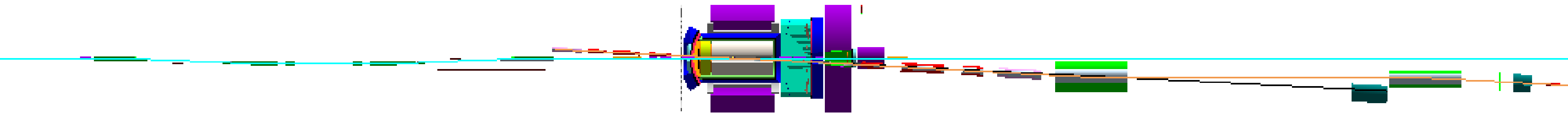
- DVCS
  "MILOU" from HERA/BNL
- Forward tagging generator
  developed by the LDRD
- SIDIS
  developed by Duke U for EIC whitepaper
- Inclusive
  "eicRate", based on CTEQ and Wiser fit
- more from BNL group

Detector Simulation (interaction region)

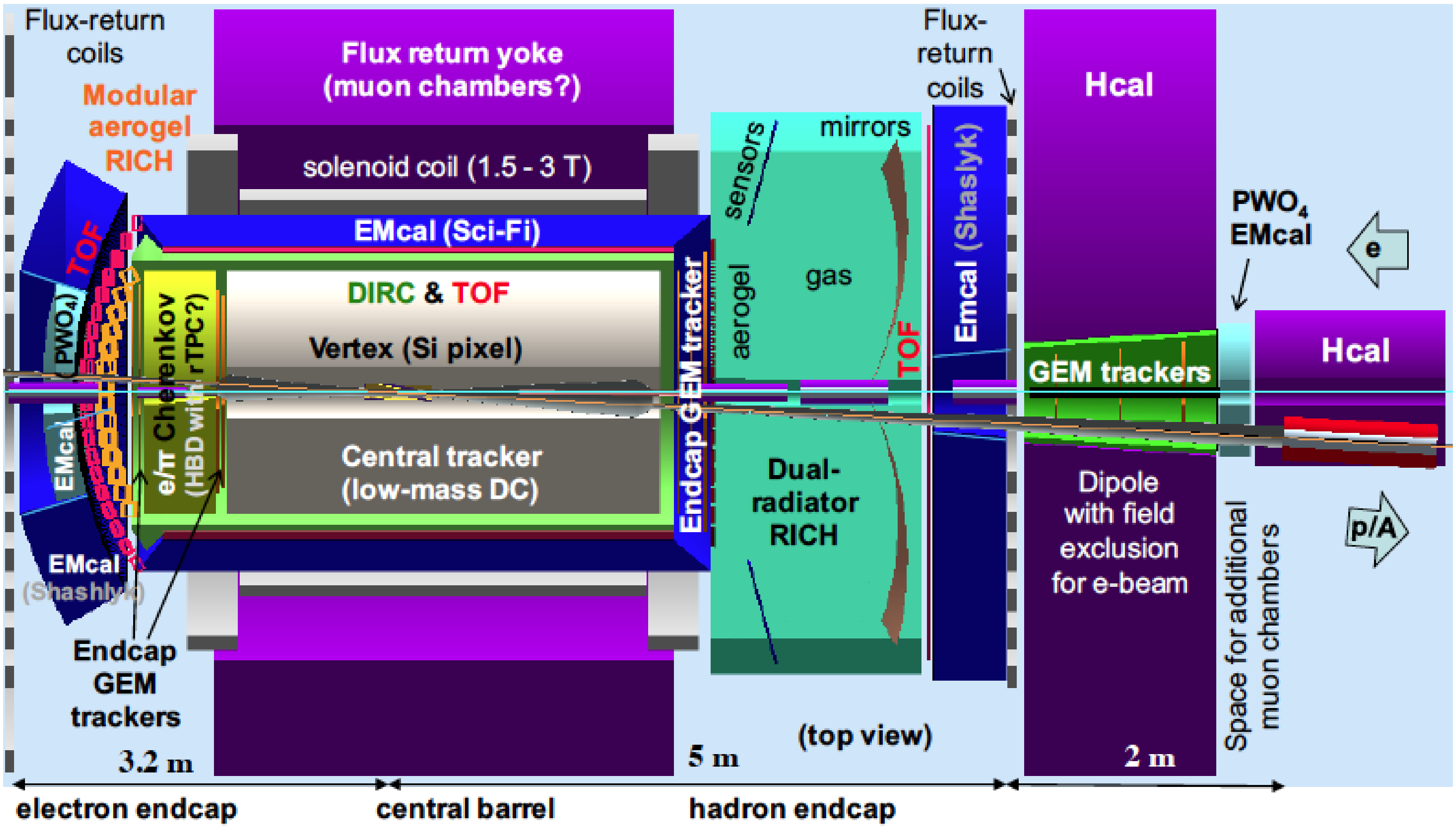
- Fully-integrated central detector and beamline with forward detection of e- and ion



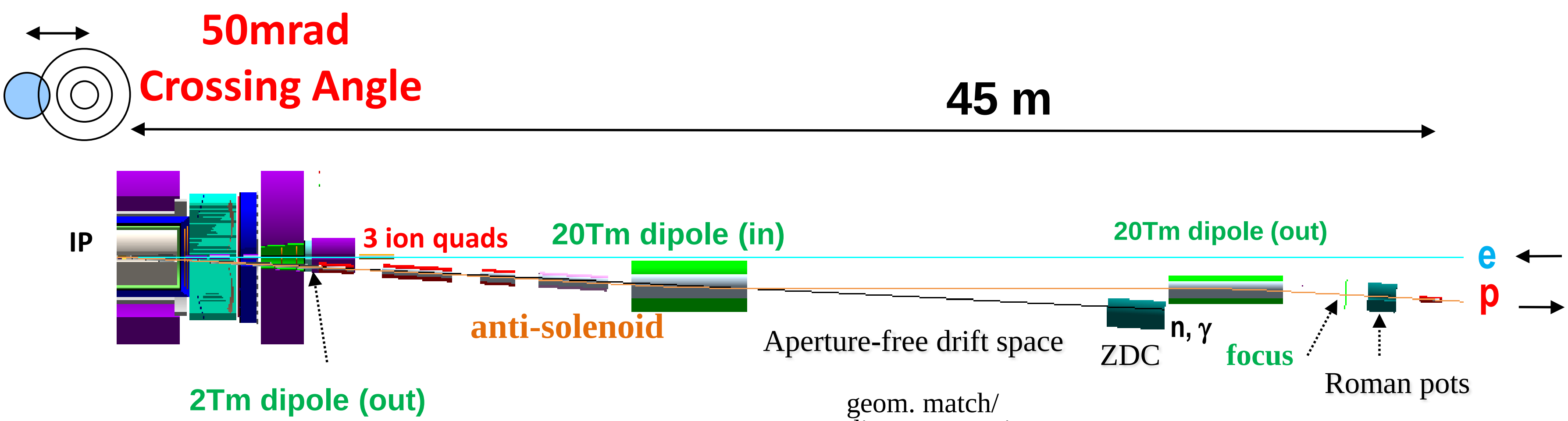


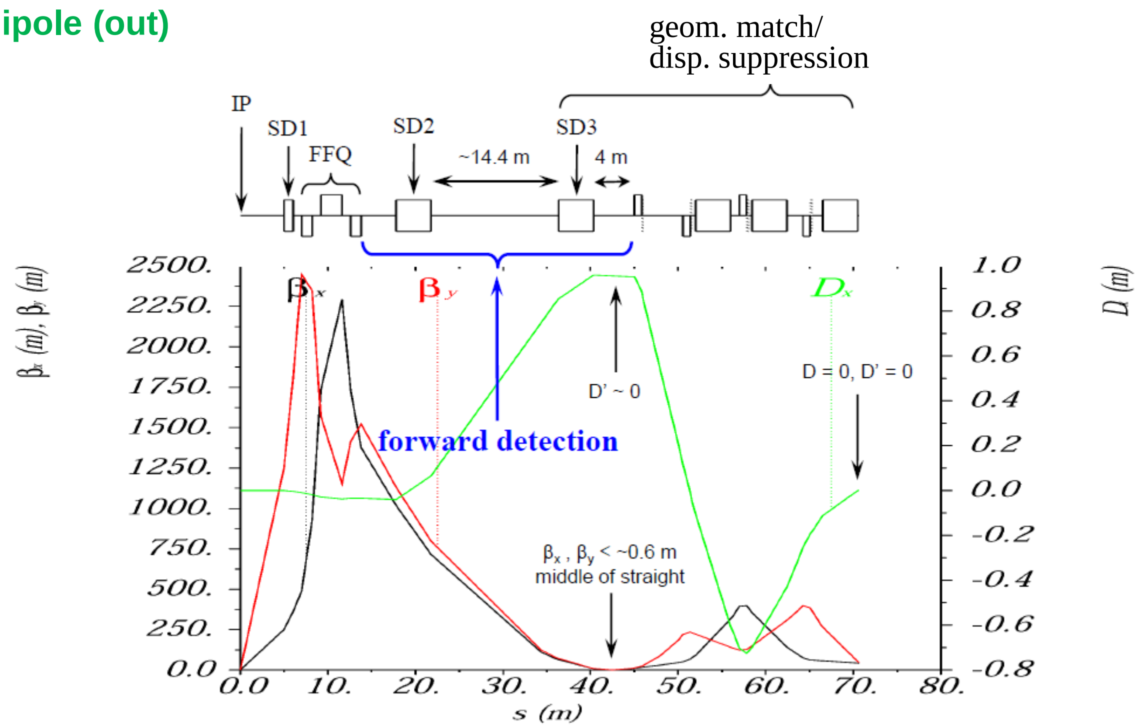


Detector Simulation (Central Detector)



Detector Simulation (Ion Forward)





Beamline Optics
from ACC design to detector design

- A simple text file to record the essential from ACC design
- A script to auto generate type, geometry, position, strength of all magnets in simulation
- It ensures a coherent IR design between accelerator or detector

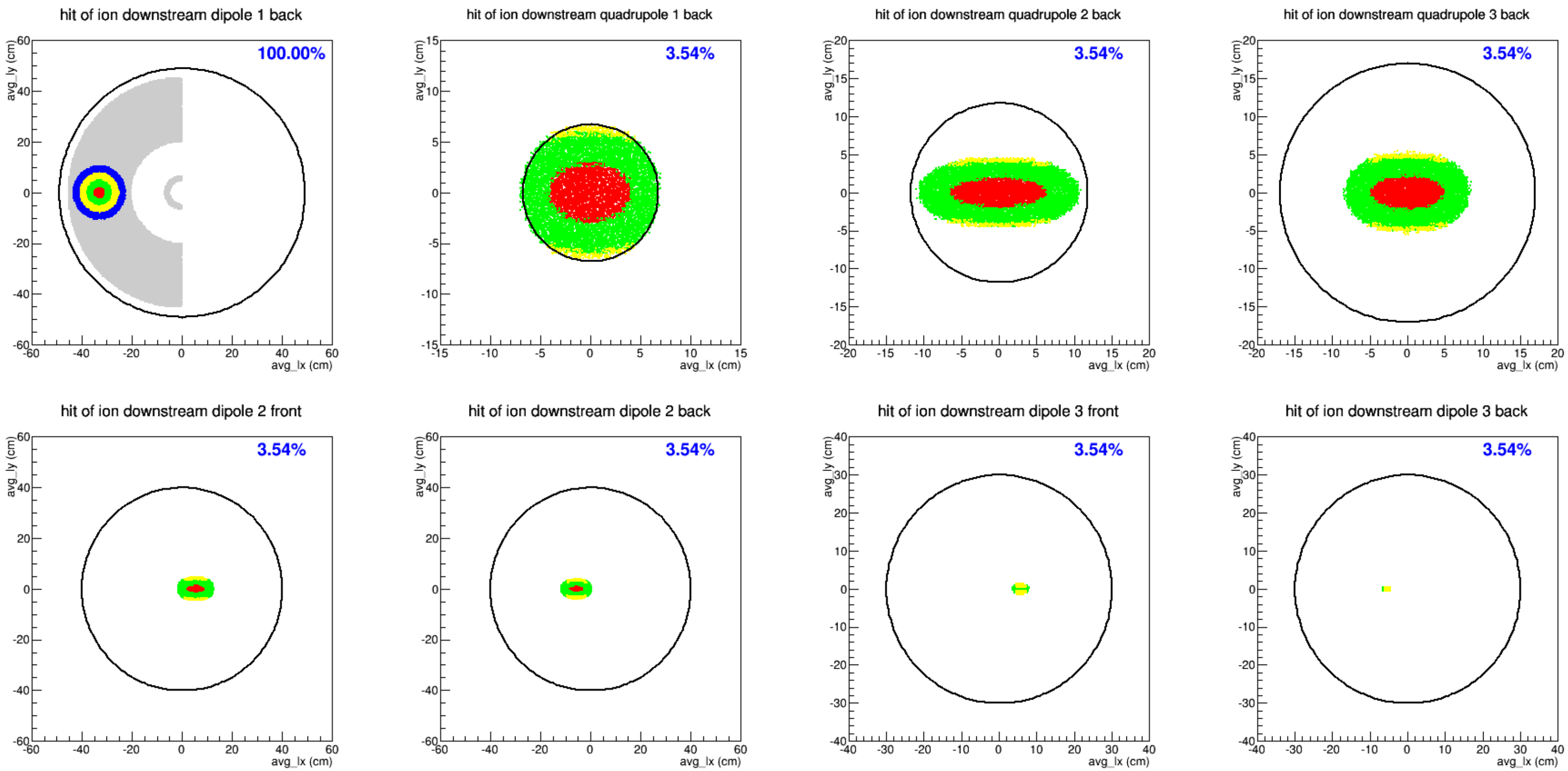
```
i_ds_dipole1_length | 1.497893902 | m | ion downstream dipole1 length | - | - | - | - | - | -
i_ds_dipole1_innerhalfaperture | 45 | cm | ion downstream dipole1 innerhalfaperture | - | - | - | - | - | -
i_ds_dipole1_outerhalfaperture | 49 | cm | ion downstream dipole1 outerhalfaperture | - | - | - | - | - | -
i_ds_dipole1_strength | 1.333366896 | T | ion downstream dipole1 strength | - | - | - | - | - | -
i_ds_dipole1_X | 0 | m | ion downstream dipole1 X center | - | - | - | - | - | -
i_ds_dipole1_Z | 5.742698253 | m | ion downstream dipole1 Z center | - | - | - | - | - | -
i_ds_quad1_length | 1.2 | m | ion downstream quad1 length | - | - | - | - | - | -
i_ds_quad1_innerhalfaperture | 6.75 | cm | ion downstream quad1 innerhalfaperture | - | - | - | - | - | -
i_ds_quad1_outerhalfaperture | 17.1 | cm | ion downstream quad1 outerhalfaperture | - | - | - | - | - | -
 i_ds_quad1_strength |-87.02661678 | T/m | ion downstream quad1 strength | - | - | - | - | - | -
i_ds_quad1_X |-0.390919069 | m | ion downstream quad1 X center | - | - | - | - | - | -
i_ds_quad1_Z | 7.5899211 | m | ion downstream quad1 Z center | - | - | - | - | - | -
i_ds_quad1_Theta |-0.05599600898 | rad | ion downstream quad1 Theta | - | - | - | - | - | -
```

quad 1,2,3 with 6,6,6T tip field

Hit pattern at ion downstream magnet

Angle		
Red	0.00-0.25	deg
Green	0.25-0.50	deg
Yellow	0.50-0.75	deg
Blue	0.75-1.00	deg
Gray	1.00 -any	deg

Proton 99-100GeV

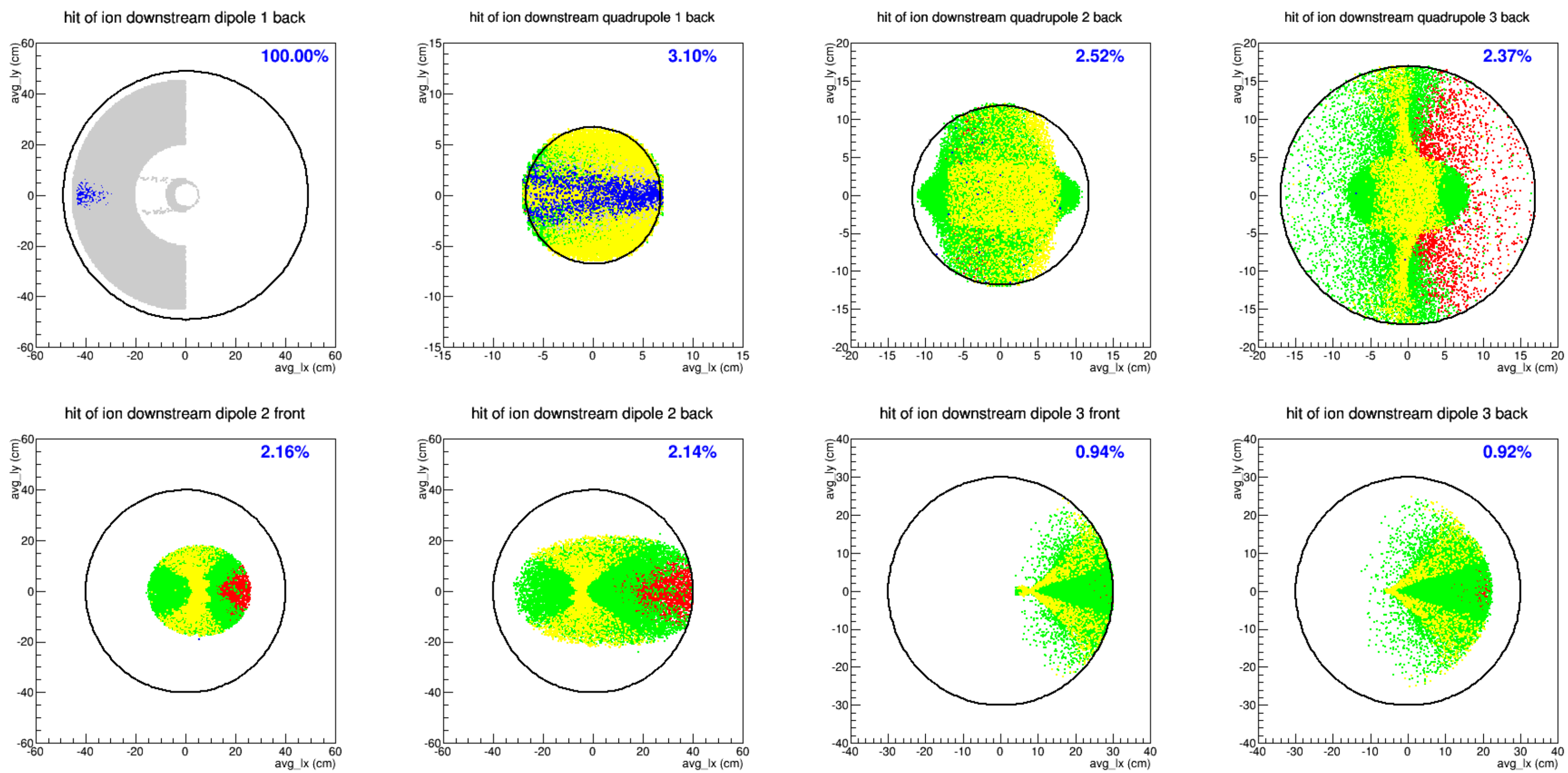


quad 1,2,3 with 6,6,6T tip field

Hit pattern at ion downstream magnet

Angle		
Red	0.00-0.25	deg
Green	0.25-0.50	deg
Yellow	0.50-0.75	deg
Blue	0.75-1.00	deg
Gray	1.00 -any	deg

Proton 0-100GeV

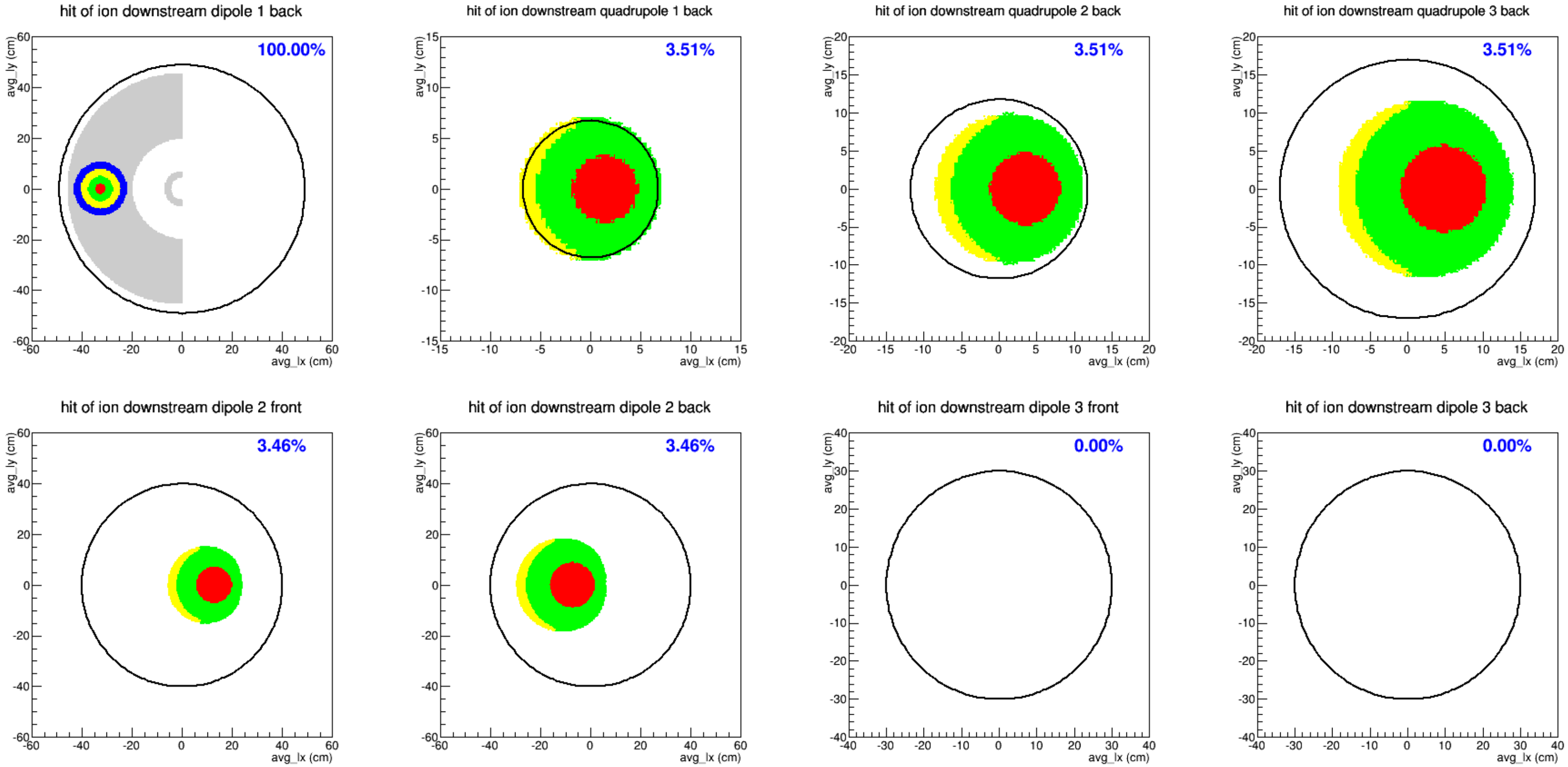


quad 1,2,3 with 6,6,6T tip field

Angle		
Red	0.00-0.25	deg
Green	0.25-0.50	deg
Yellow	0.50-0.75	deg
Blue	0.75-1.00	deg
Gray	1.00 -any	deg

Hit pattern at ion downstream magnet

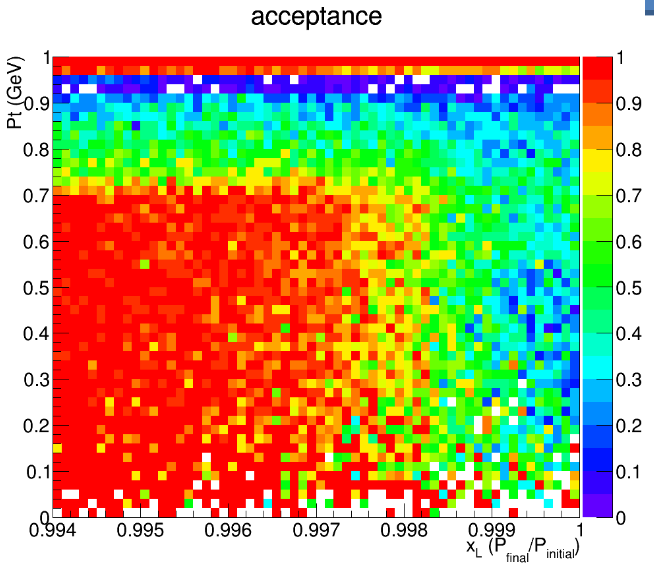
Neutron 0-100GeV

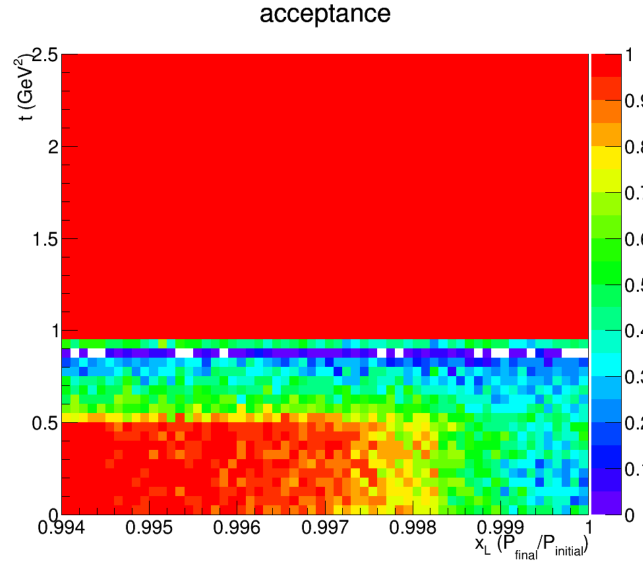


Acceptance (with cooling)

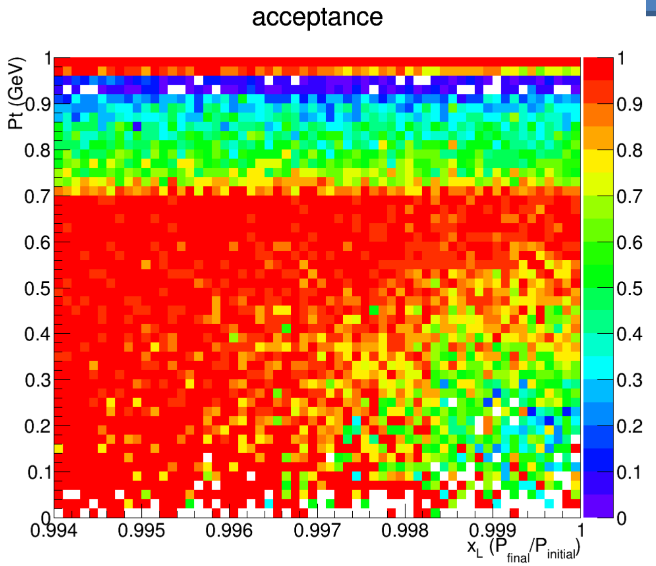
Extended vertex

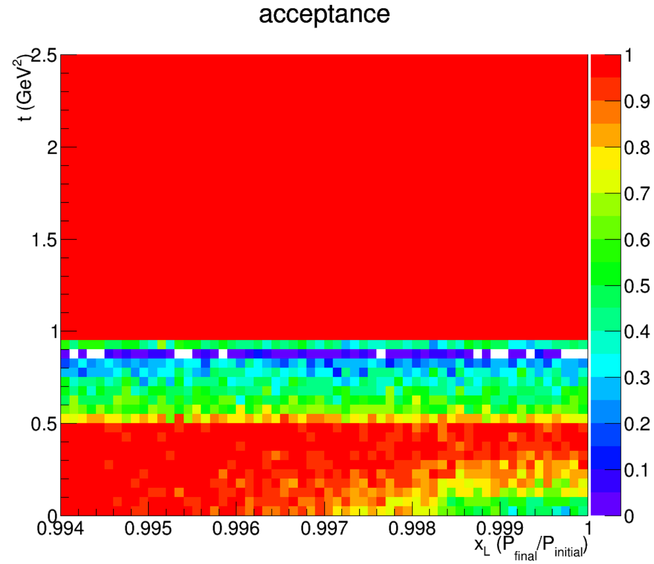
Cut on 2nd focus





Add these detected by two planes





Eic_fastMC

- Study acceptance of primary particles in eic_gemc and at various detector region with field effect, and store it in 2D/3D histogram
- For different detector region, use file/database to store detector features like track resolution and pid probability
- Front end is a function with stable input and output format as only interface to users
  - input: particle type and 4-mom
  - Return: acceptance at a particular region and its resolution and pid performance, also smeared 4 vector

How to use

https://jlabsvn.jlab.org/svnroot/eic/study/eic_fastmc/
(work in progress)

- #include "eic_fastmc.h"
- eic_fastmc(const char detector[],const char version[],const char ebeam[],const char ibeam[],const char solenoid[],bool debug=false);
- int get_eic_fastmc(const int pid,const double * kin,int *region,double *acc,double *res,double *kin_smeared,double *pident,bool debug=false);

- It takes input

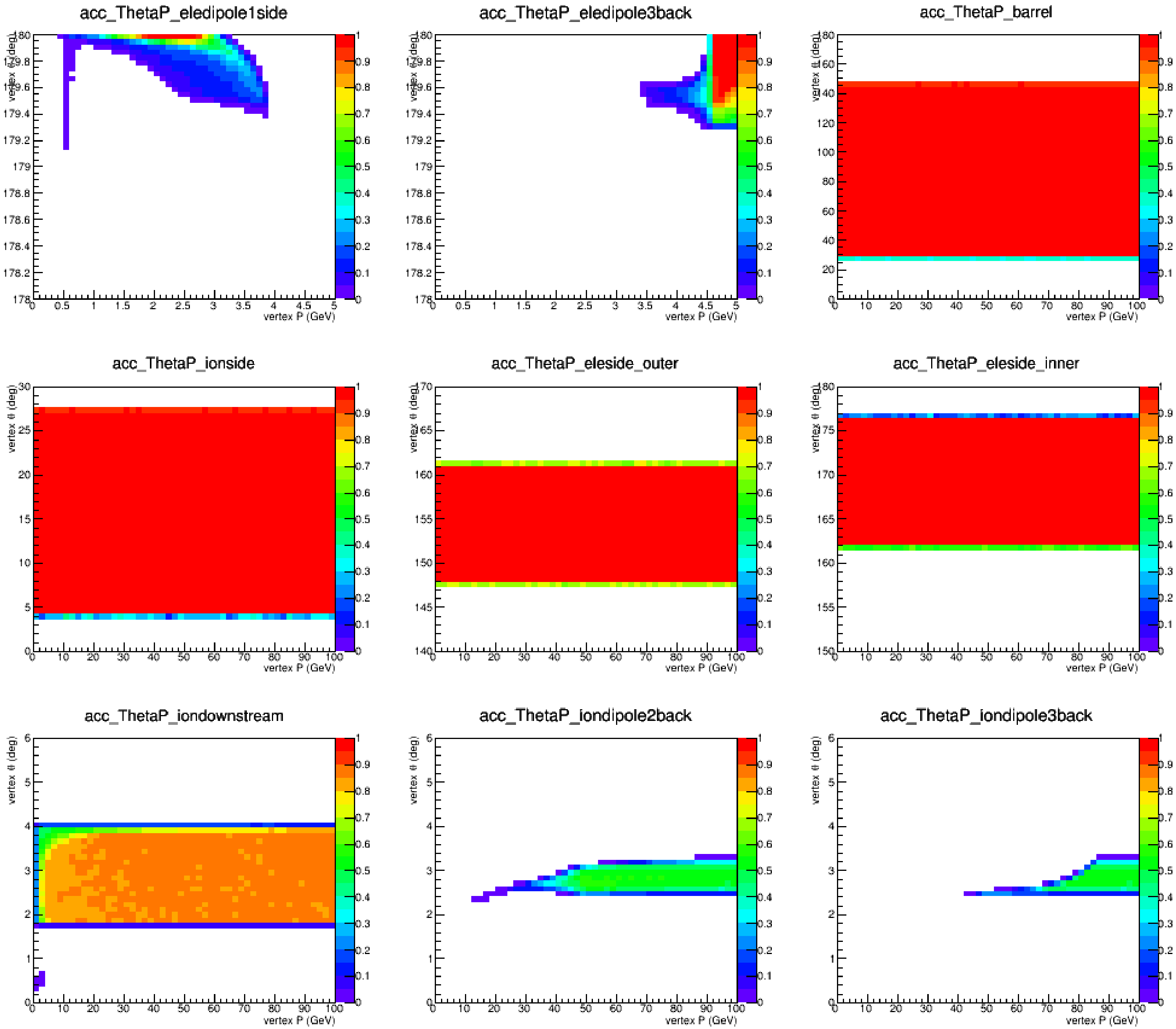
1. pid in PDG format (not implemented yet)
2. vertex P (GeV)
3. vertex Theta (deg)
4. vertex Phi (deg)
5. vertex E (GeV)

Using acceptance determines region, then we find PID and resolution accordingly

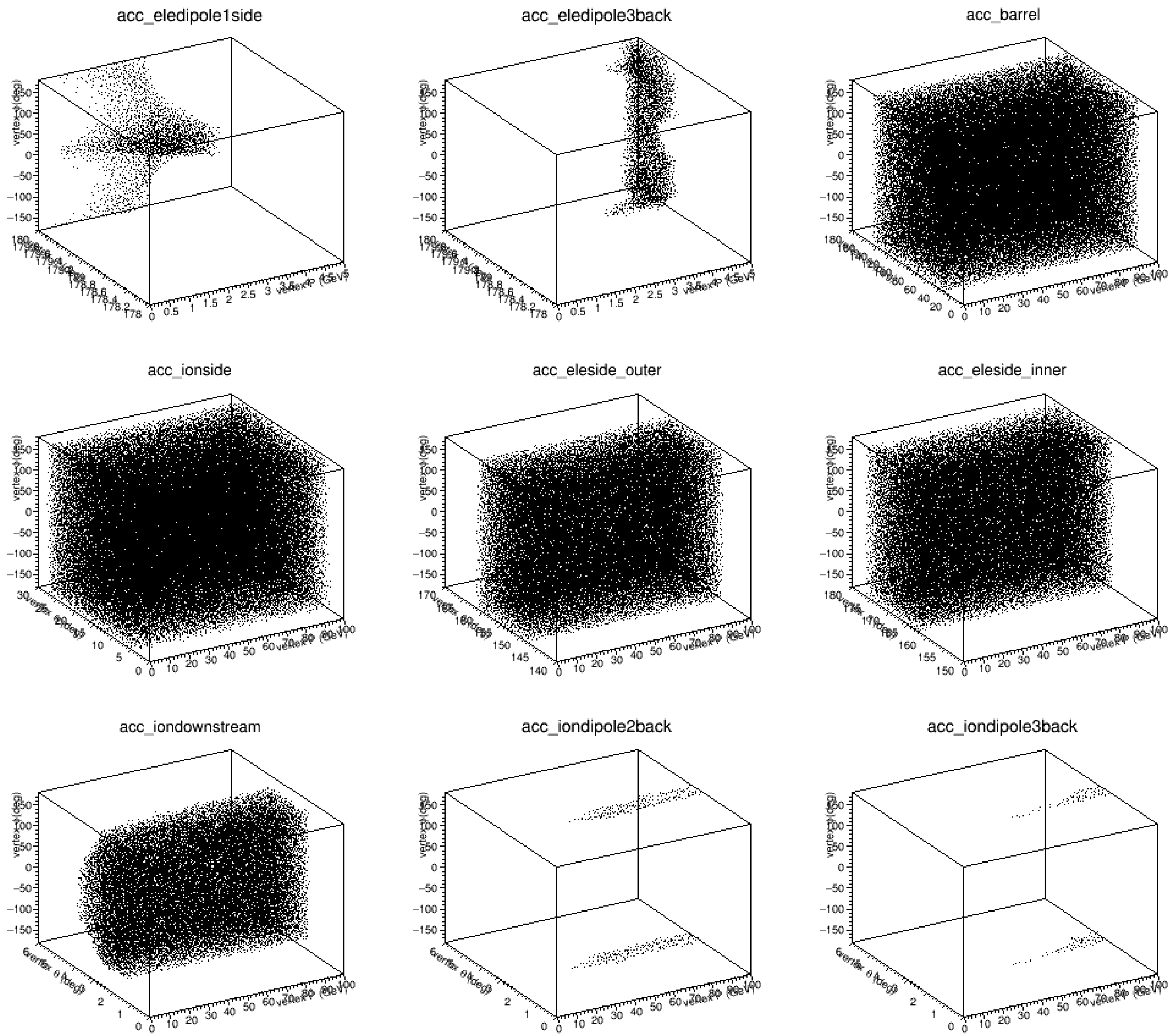
- and return output

1. region accepted (0-8) or not accepted in any region (-1)
2. acceptance value (0-1)
3. resolution of dP/P(%), dTheta(deg), dPhi(deg), dz (cm)
4. Gaussian smeared P(GeV),Theta(deg),Phi(deg),E(GeV)
5. probability of identifying this particle (0-1)

Acceptance 2D



Acceptance 3D



Summary

- JLEIC detector simulation “eic_gemc” has the capability to support different sub-systems development and combine them together effortless into full simulation
- We have established an easy way to translate ACC design to detector simulation
- eic_fastMC is under work for common user
- Things will evolve, in particular when we are searching for a complete software framework