ESCalate India

IIT Indore Group: Hasan Mustafa and Vineet Tripathi, Dr. Ankhi Roy

RKMRC Kolkata Group: Sayantan Neogi, Dr. Amal Sarkar

with

Dr. Dmitry Romanov (JLAB)

ATHENA India Meeting
26th August 2021

e - π separation using machine learning

Possible Group
DWG-Calorimetry/PID

G4E

- Using G4E(Part of Escalate Framework) simulated hits in 3x3 calorimeter.
- Simulated 100,000 events.

ROOT FILES

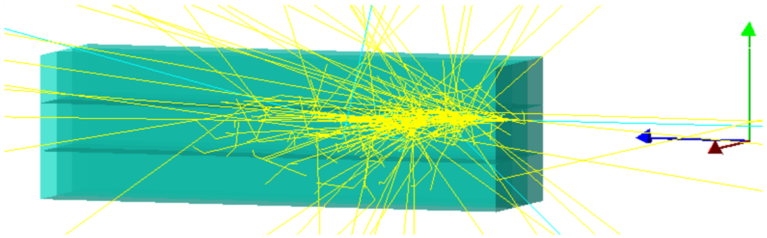
Preprocessing and Data Augmentation

- Read the root file generated using uproot and convert data to numpy array.
- ONE Feature containing 3x3 adc responses was extracted and recorded from each event.
- Normalize the data and perform logarithmic transformation.
- Same data set used in Energy Extraction(Next Slide)

Numpy Array

Keras Model

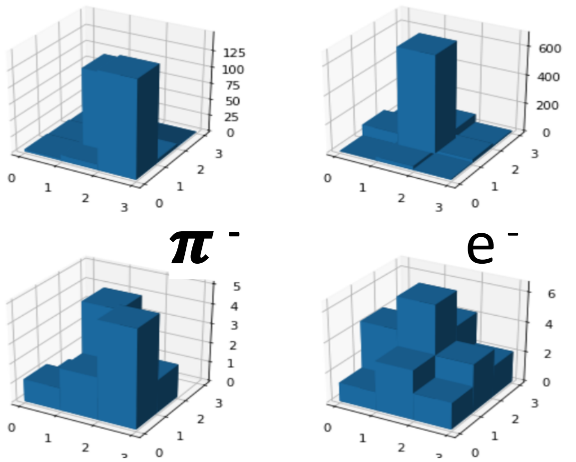
- Neural network was trained separately; first on original data and then augmented data.
- Categorical Cross Entropy loss was minimized in the training
- Model's predictions were tested with test data



Sample event: Electron creating an EM shower in the calorimeter.

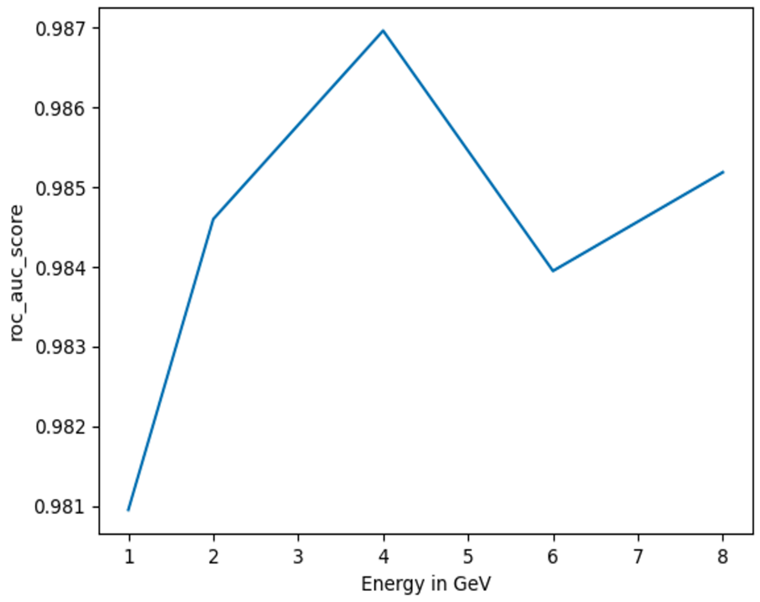
Original Data

Log Transformed Data



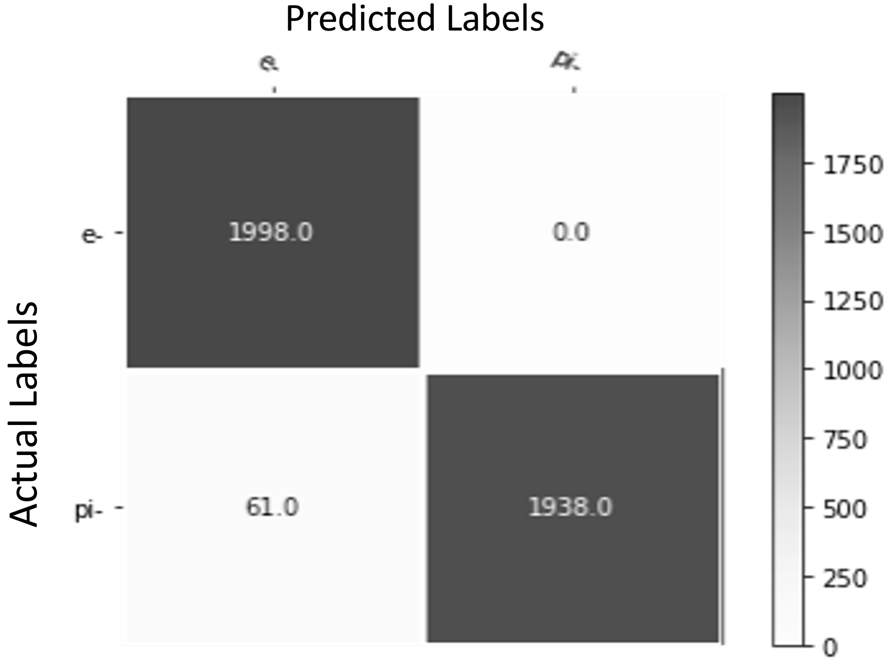
Results

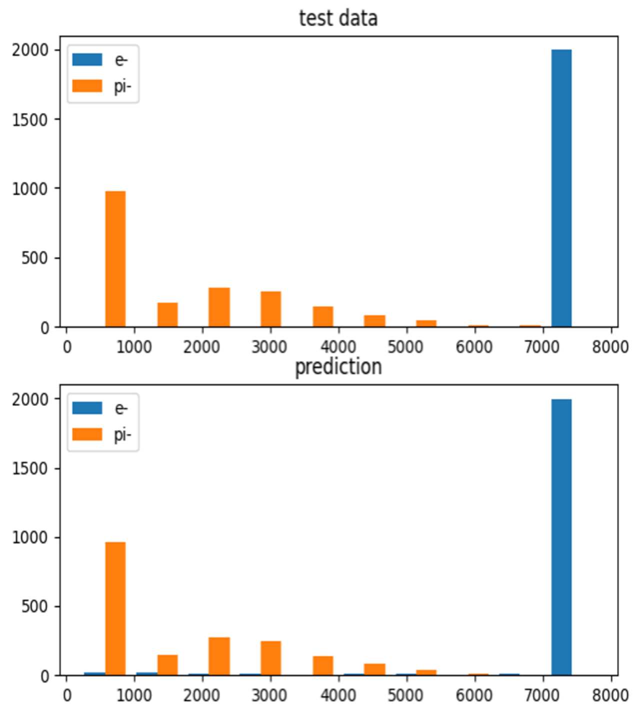
Possible Group
DWG-Calorimetry/PID



Accuracy vs Energy

Confusion matrix for 8 GeV particles





Total ADC-value distribution

Neural Network with augmented data

- Neural Network Trained on Augmented Data
- Accuracy Independent of Energy (~99%)
- Considerably less number of misclassified pions compared to the baseline.

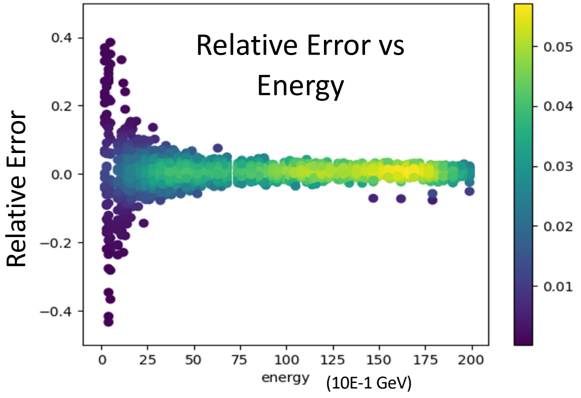
Energy Extraction using machine learning

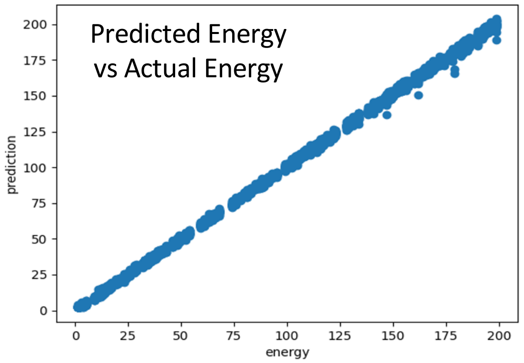
Possible Group
DWG-Calorimetry/PID

Keras Model

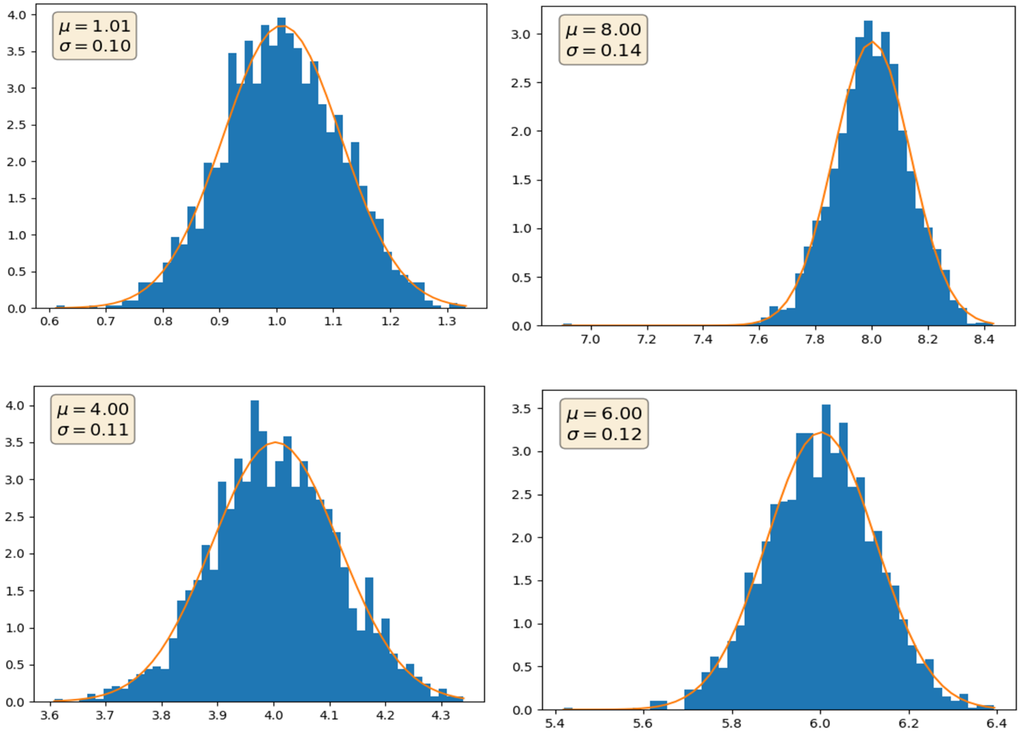
- Neural network was trained on the data.
- A combination of mean square relative error and mean absolute error was minimized in the training.
- Model's predictions were tested with following test data.
  - Multiple events of same energy
  - Random Energy Events

RESULTS





Distribution of Predicted Energy for events of Known Energy



Loss Function Used: `tf.square((y_true - y_pred)/(y_true))+tf.abs(y_true-y_pred)`

IITI Group

Beampipe construction from CAD files

Possible Group
DWG-Calorimetry/PID

Visualization and preprocessing

- Obtained CAD files (.stl) for different parts of beampipe from the repository.
- Used FreeCAD to visualize the obtained beampipe chambers.

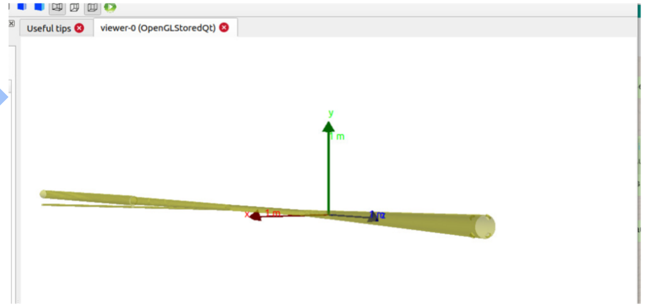
CAD files

Modifying G4E Source Code

- Modified the beampipe construction file to add beampipe elements from CAD.
- Assembled the beampipe elements using the dimensions and parameters specified.

Header and source code

- Rendered the beampipe from macro file.

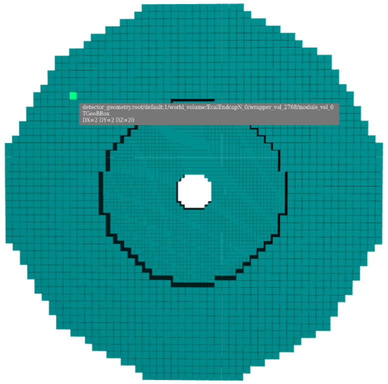


Energy Extraction for Hybrid Calorimeter(In Progress)

Athena

- Generate events with hits near the transition region by isolating the calorimeter from Athena Detector.

Fig: Hybrid Calorimeter having two distinct regions (Concentric Circles) with different resolutions.



Uproot

- Parse the root files using Uproot and obtain a data set where energy deposited in each cell is recorded.

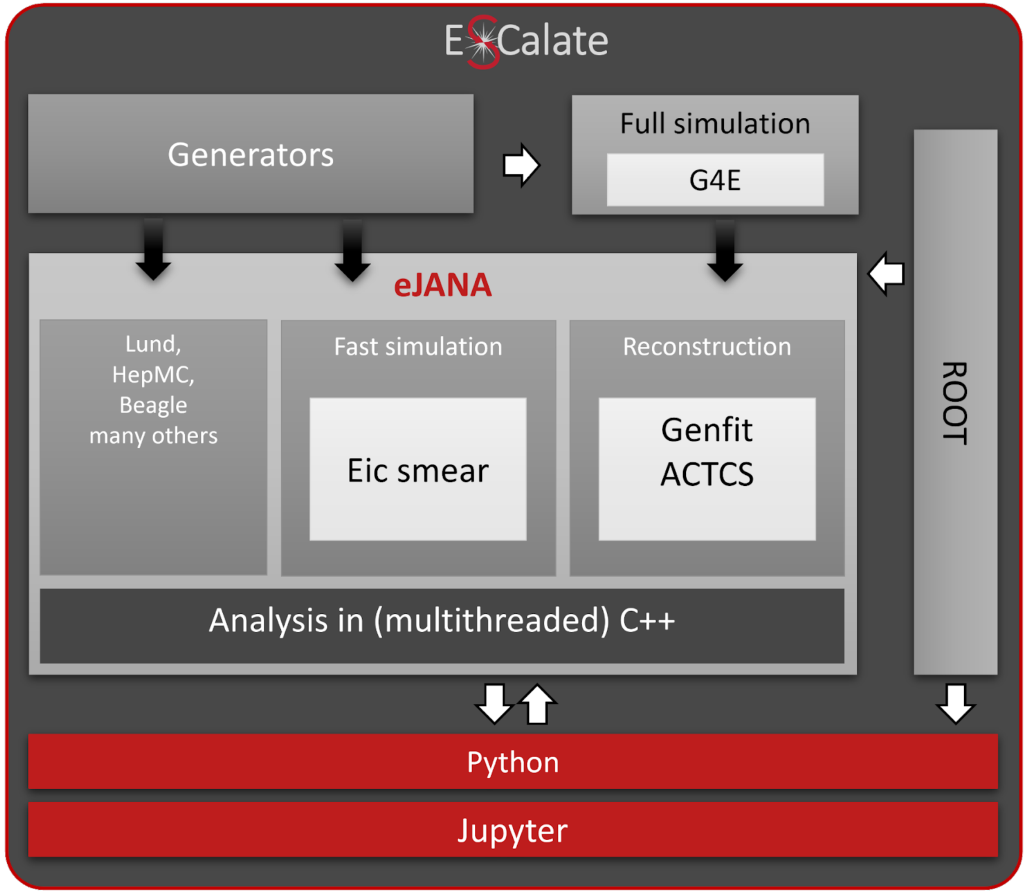
Tensorflow/Keras

- Train a Deep Neural Network to predict energy of particles.
- Use the neural network to predict energy of the particle hitting the calorimeter.
- Task is especially challenging when there are two nearby hits in the calorimeter.

RKMRC Group

Full Simulation with EScalate

Possible Group
ATHENA Software



EScalate is a platform which can perform full and fast simulations

Why EScalate?

- Jupiter lab interface
- Can write programs in Python / c^{++}
- Modular in both ways
- Ensure data consistency
- Easy to understand

Our aim is to modified such a way that,

- Can write programs in Python / c^{++}
- Take any format of MC data as input
- Deploy like Docker, Spack, CVMS etc
- Easy to maintain and modify

Available simulation option

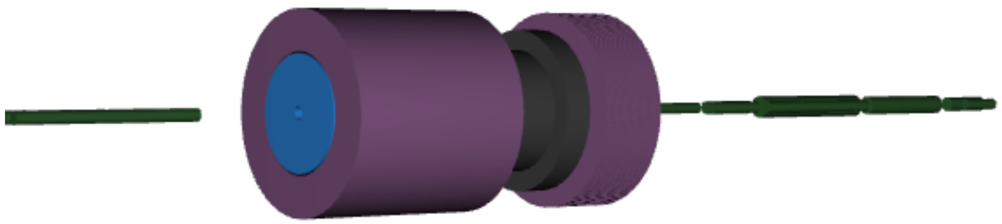
- **Fun4all** (originated from within (s)PHENIX, mature and centered around the use of ROOT macros)
- **G4E** (build up for the EIC (and therefore in a “younger” stage of development) constructed as a pure GEANT4 application (and integrated into JupyterLab environment))

Running EScalate

Easy to understand and modify

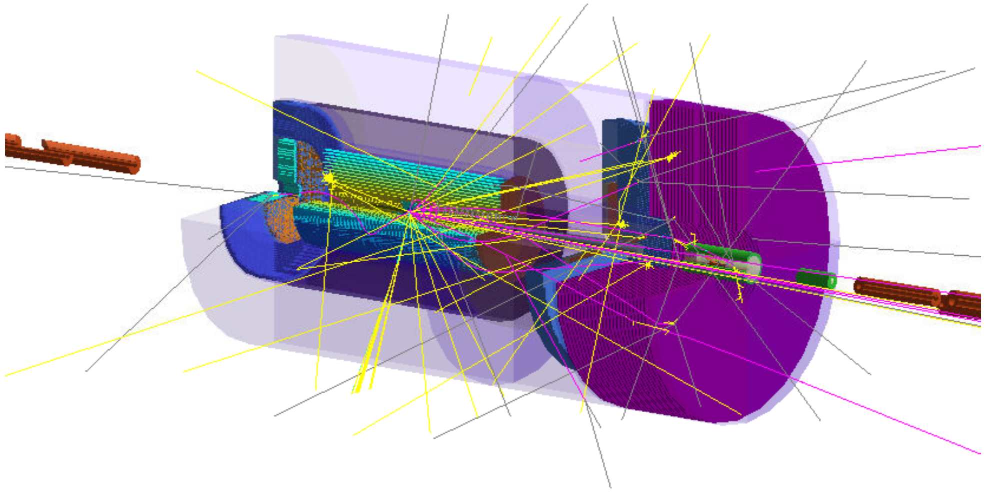
- ➢ Can specify which detector we want
- ➢ Can select MC data to simulate

On Jupiter Lab



- No detector internal layout
- Tracks can not be seen

On VNC Platform



- Detector internal lay out in detail.
- Tracks can be seen
- Improve performance with processor
- Track IDs, ongoing

Possible Group
ATHENA Software

Summary

Testing: 2 phase technic

- Phase 1 – test MC format data and test detector to identify all particle correctly. Used catch2 for testing.
- Phase 2 - check environment to test all framework. Used GitHub ci action to test continuous testing. If we change anything, ci automatically activate (DEVOPS technic).

Current work in progress

- ❖ Associating tracking IDs with tracks in simulated interface. Currently, it can be seen on the terminal
- ❖ A major part of Escalate are in process of implementation in ATHENA software

Future direction

- Improving multithreading.
- Improve simulation quality.
- Improve particle tracking id
- Add more flexibility.

IITI Group

RKMRC Group

Group activities and availability

Ankhi Roy and the group (IIT Indore)

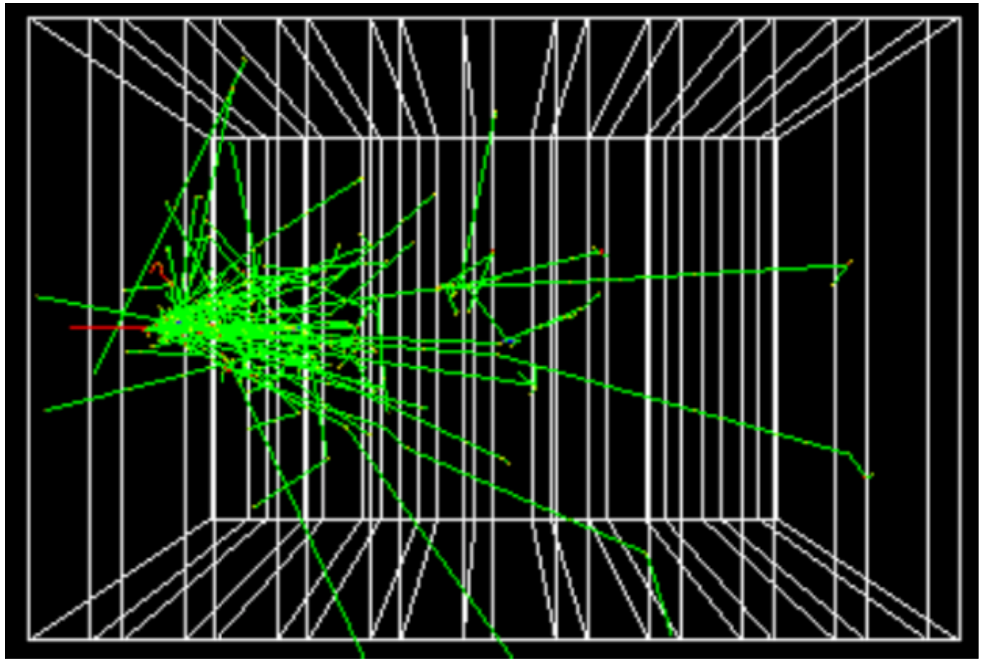
Name of the Students	Activities / interest	Availability
Sagar and Siddhant	EMCal, HCal resolution with Fun4All Framework	Working with Chris
Hasan and Vineet	Escalate Machine Learning with Dmitry	Working for ATHENA
Diksha (M.Sc, Phy) & Hridey	PWG exclusive, interested in XYZ spectroscopy	Available for next one year
Nilay (M.Sc, Phy)	ML for PID in TRD with Yulia and Dmitry	Working for ATHENA

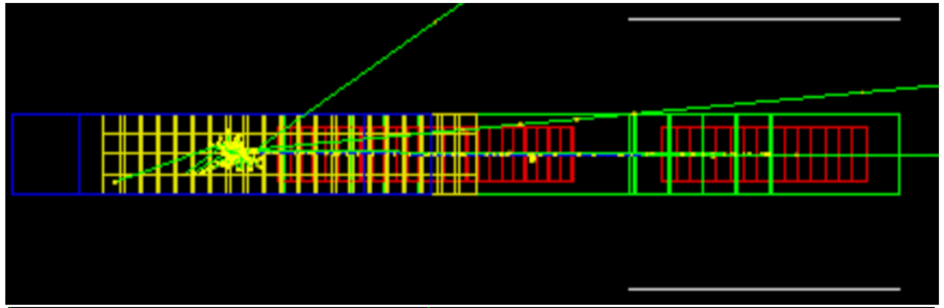
Amal Sarkar and the group (RMKRC, KOL)

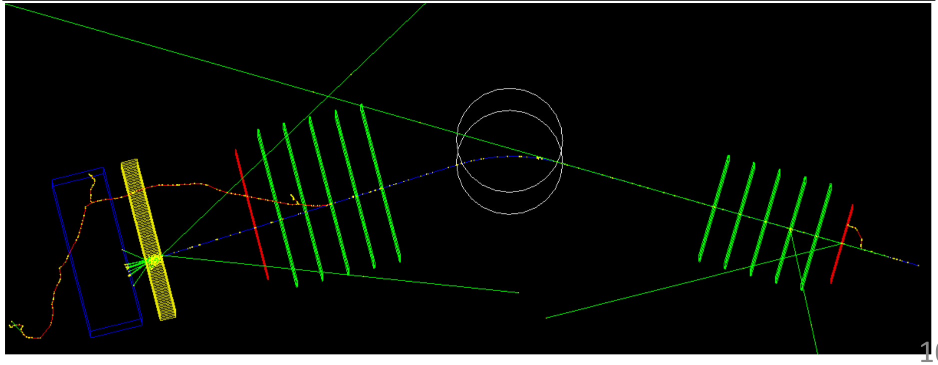
Name of the students	Activities / interest	Availability
Sayantan Neogi (MSc, CS)	Continue working on Escalate inputs to ATHENA software with Dr. Dmitry	Working in ATHENA
Swapnamoy, Kaosor Ali (MSc, Phy)	Will work on the Physics part of tracking [GEM Tracking] in contact with ERD6 group.	Available for next one year
Bappadittya (M.Sc, Phy) Payal (M.Sc, Phy)	PWG exclusive, interested in XYZ spectroscopy	Available for next one year [in ATHENA]
Two M.Sc, CS, will be available from September 2021.	Will work on the software part of tracking with Dr. Dmitry's group	Available for next one year [in ATHENA]

Project eAST (IITB, IITM, Goa University)

- **Present task:** Work on interface to MCEGs (HepMC3) and help with validation of test-beam data.
- Vashishtha Kochar, Aryan Borker, Pranjal Verma, Chinmay Seth, Suvarna Patil, and other colleagues are getting familiar with HepMC3 and Geant4 tutorials.
- They will test the interface to MCEGs for the supported formats.
- Thanks to Makoto Asai for introductory lecture with some excellent hands-on exercises to get familiar with Geant4. This would help in development of eAST (following pictures are from some basic examples)







Thank you ….

Design we need

- Modular
- Provide interfaces to internal layers.
- Interaction between layers must be clear.
- Each layer must be replaceable.

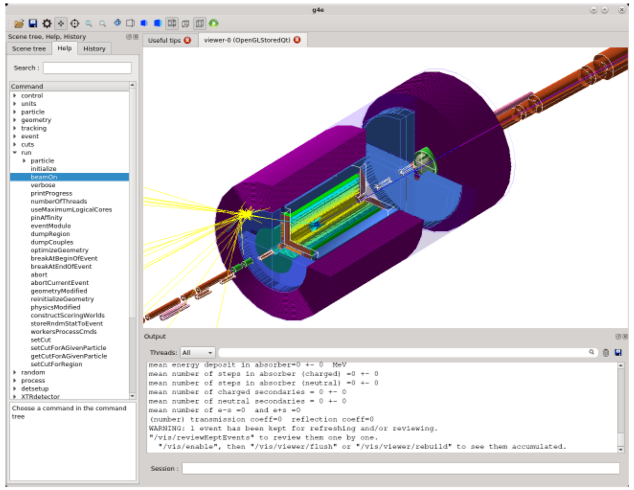
simple	JupyterLab web interface
moderate	analysis scripts, python
complex	eJANA, plugins, C++
expert	JANA, eic-smear, *fun4all*, ROOT, Geant4

Available simulation option

Fun4all (originated from within (s)PHENIX, mature and centered around the use of ROOT macros)

G4E (build up for the EIC (and therefore in a “younger” stage of development) constructed as a pure GEANT4 application (and integrated into JupyterLab environment))

Simulation – G4E/Geant 4



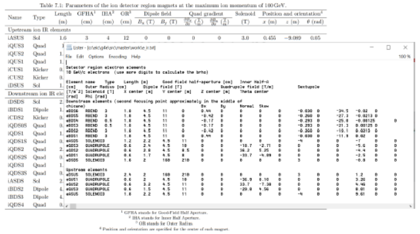
Standalone C++ Geant4 application

Various EIC MC file formats
Beagle, Pythia6, HEPMC - Pythia8, Herwig and others

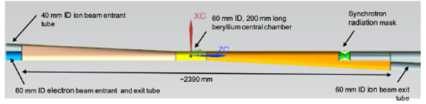
Integration with accelerator elements

Infrastructure to import Geant4 detector geometry and simulation code

Accelerator design (beam elements)

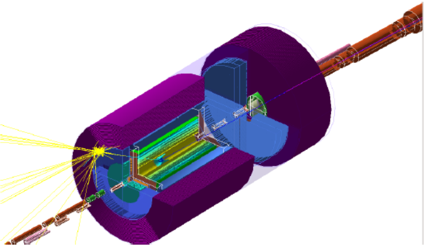


Engineering Design (CAD)

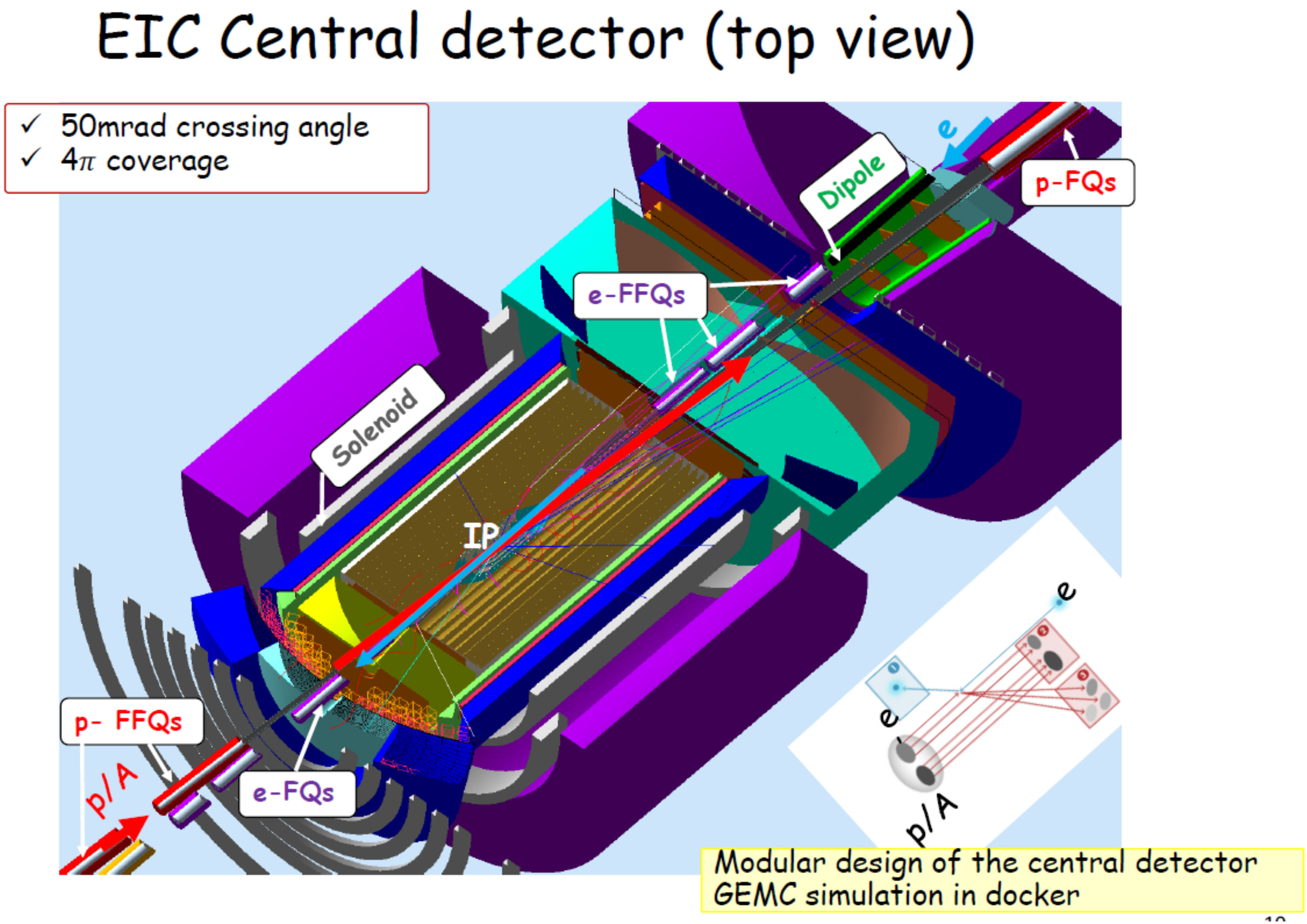


Tuning

Detector Simulations
(Geant4)



JLEIC


EIC Central detector (top view)
✓ 50mrad crossing angle
✓ 4π coverage
e
Dipole
p-FQs
e-FFQs
Solenoid
IP
p- FFQs
p/A
e-FQs
e
e
p/A
Modular design of the central detector
GEMC simulation in docker

Running On Jupiter La

Easy to understand and modify

Example:

g4epy is a python library to simplify configuration of running g4e

```
[ ]: from g4epy import Geant4Eic

[ ]: g4e = Geant4Eic(detector='jleic', beamline='erhic')\
              .source('../data/herwig6_20k.hepmc')\
              .output('hello')\
              .beam_on(200)

[ ]: g4e

[ ]: g4e.run()
```

- Can specify which detector we want
- Can select MC data to simulate

On Jupiter Lab