

# Simulation towards ECCE proposal

Jin Huang

Brookhaven National Lab

Cameron Dean

Los Alamos National Lab

# ECCE simulation

Concept from detector team  
(an example with 1.5T magnet shown here)

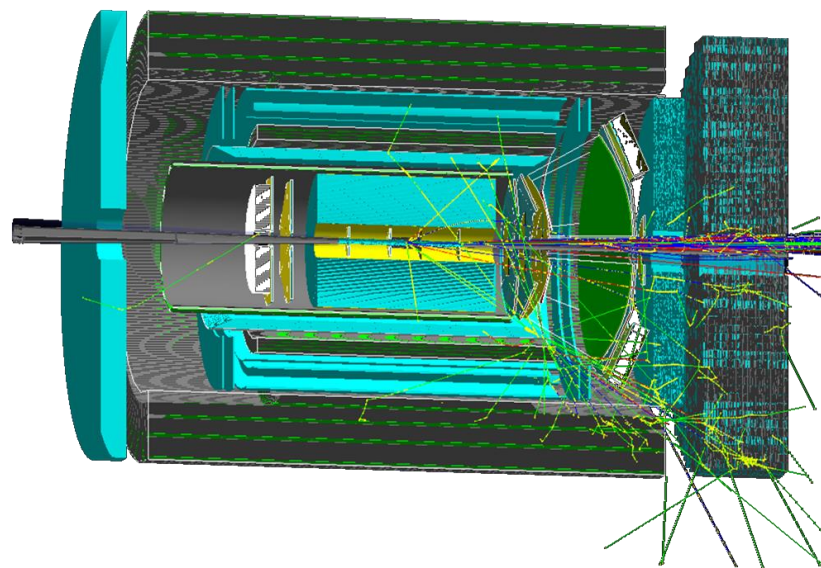


Fun4All sim. and reconstruction

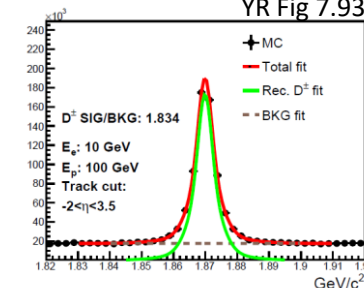


Validating performance

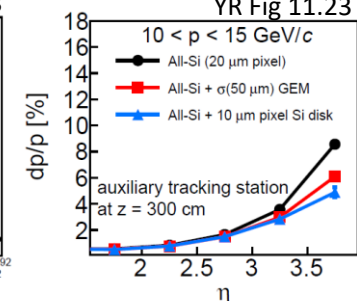
Yellow Report (YR) Fig 11.65



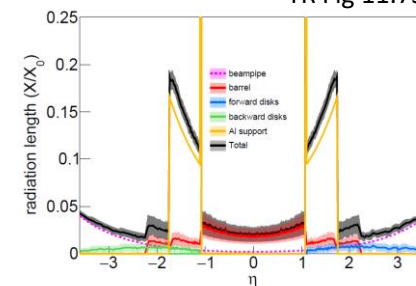
YR Fig 7.93



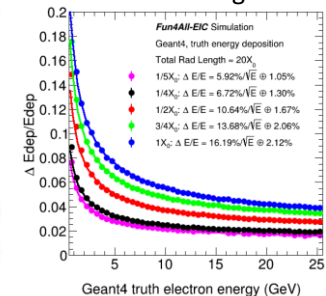
YR Fig 11.23



YR Fig 11.73

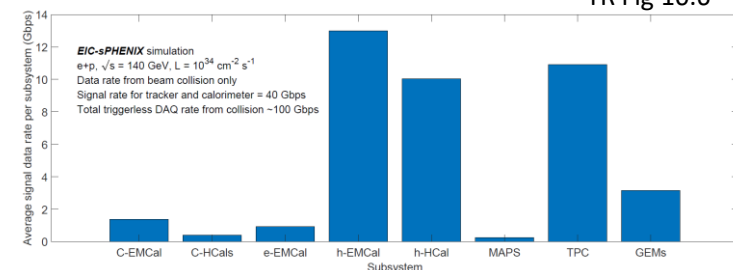


YR Fig 11.55




- Actual ECCE concept and study to be defined by the proto-collaboration. Figures in this slide are used for illustration purpose
- Fun4All-ECCE framework will be used as the simulation work in proposal stage
- ECCE simulation will be open to new software development/adoption/improvement for use in after the proposal stage

YR Fig 10.6



# Simulation coordination towards proposal

- ▶ Time is short
- ▶ Full G4 simulation can be time-consuming. Simulation team plan to help WGs plot most direct path for delivering plots in the proposal
- ▶ Simulation campaign will be planned around major plot deliverables for the proposal
- ▶ Parallelize detector definition and exercise of reco. tools using existing configurations
- ▶ For large sample, we will organize coordinated simulation production, making result DST files available to analyzers


- 
- Dec 1<sup>st</sup> : proposal submission
  - Nov 1<sup>st</sup>: final proposal for editing
  - Sept 1<sup>st</sup> : all major simulation plot done. After this date, we just do polishing, composing narratives around the figures in the performance chapter of the proposal.
  - Aug 1<sup>st</sup> : Final simulation production done
  - July 1<sup>st</sup> : Final simulation production start
  - May 1<sup>st</sup>: First simulation campaign, followed by first round of analysis. From May – July, many studies probably need another iteration of simulation-analysis to advance detector design.
  - Entire Apr: develop simulation setup to run.
  - Today: 1<sup>st</sup> simulation workshop

# Simulation Software Infrastructure



# Simulation Software Infrastructure

- ▶ <https://github.com/ECCE-EIC/> is born this week!
  - Open-source software repository
  - Independent fork from EICUG and sPHENIX
- ▶ Please email your github username to [jhuang@bnl.gov](mailto:jhuang@bnl.gov)
- ▶ At the ECCE proposal stage we will avoid detach from sPHENIX core-software
  - Sync fork of Fun4All **core-software** to sPHENIX
  - Will implement CI gateway check before each syncing
- ▶ We will build our own detector and analysis on top of the Fun4All core:
  - **ecce-detectors** [daily build, require pull request]:  
G4 detector description, reco. Module
  - **macros** [daily build, require pull request]:  
ECCE detector description and common macros
  - **Analysis** [collaboration-wide writable]:  
Analysis module, macro, notebook



## EIC Comprehensive Chromodynamics Experiment

Develop an EIC detector around an existing 1.5T solenoid and is envisioned to offer full en detection region

<https://www.ecce-eic.org/>

Repositories 13 Packages People 6 Teams 3 Projects Settings

Find a repository... Type Language Sort

### utilities

Forked from eic/utilities  
the Fun4All bag of utilities

Perl 12 0 0 0 Updated 1 hour ago

### doxygen

Forked from eic/doxygen  
Software reference <https://ECCE-EIC.github.io/doxygen> via JenkinsCI

HTML 2 0 0 0 Updated 2 hours ago

### analysis

Generic analysis repo belong to all members. Post your analysis modules and macros here.

0 0 0 0 Updated 3 hours ago

### ecce-detectors

ECCE dedicated detector and reco repository

0 0 0 0 Updated 4 hours ago

### coresoftware

Forked from sPHENIX-Collaboration/coresoftware  
Our big core software repository

C++ 94 0 0 0 Updated 2 days ago


### opt\_sphenix\_scripts

Forked from sPHENIX-Collaboration/opt\_sphenix\_scripts  
Scripts we keep in /opt/sphenix

Shell 4 0 0 0 Updated 2 days ago

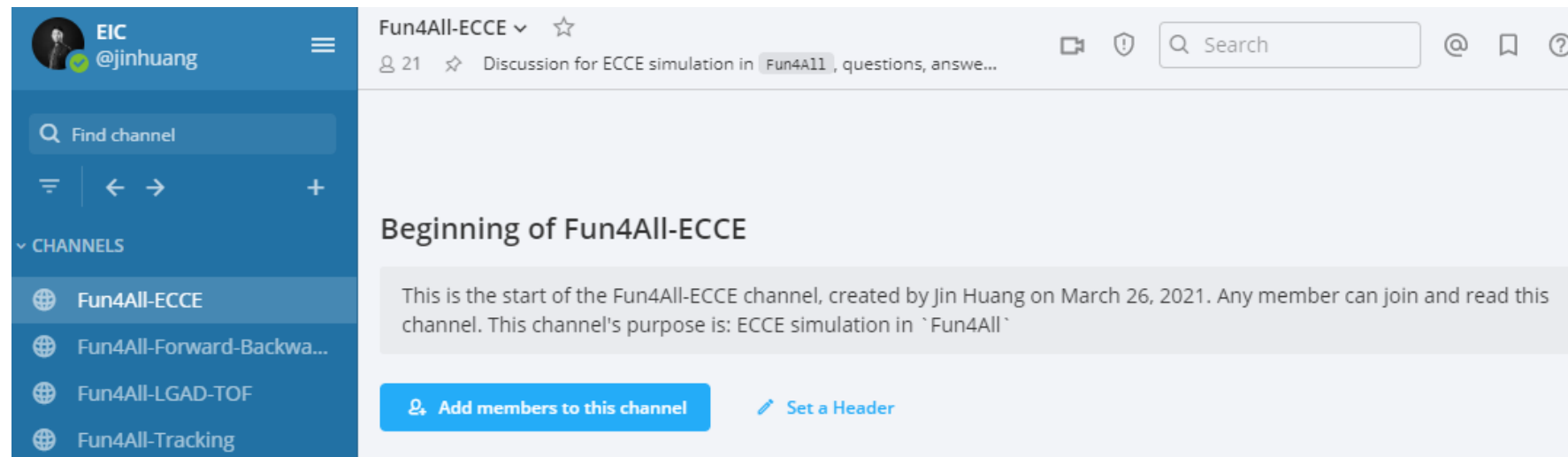
### macros

Forked from sPHENIX-Collaboration/macros  
Official macros to run sPHENIX coresoftware. Welcome to fork and to edit for your own studies.



# Simulation Software Infrastructure: chat

- ▶ As we are putting together software stack for ECCE, things will be broken and fixed.
- ▶ Don't hesitate to ask questions!
- ▶ Instant chat for help/discussions on ECCE simulations  
<https://chat.sdcc.bnl.gov/eic/channels/fun4all-ecce>
- ▶ Don't have BNL SDCC account? No problem: email [jhuang@bnl.gov](mailto:jhuang@bnl.gov) to get an invitation



# Simulation Software Infrastructure

- ▶ Daily build of ECCE software and distribute via OpenScienceGrid CVMFS
- ▶ In place where sourced `eic_setup.sh` or `sphenix_setup.sh`:  
`source /cvmfs/eic.opensciencegrid.org/ecce/gcc-8.3/opt/fun4all/core/bin/ecce_setup.sh -n`
- ▶ See talk: Chris Pinkenburg
- ▶ Use out of box in major NP/HEP scientific computing centers
- ▶ Install in your institution Linux sever or laptop via Linux VM:  
<https://github.com/ECCE-EIC/Singularity>
- ▶ Same container can be used for EIC-Smear fast simulation [talk: Kolja ]

<https://github.com/ECCE-EIC/Singularity>

## Singularity container for ECCE@EIC Fun4All

Singularity container for ECCE@EIC Fun4All allows any user to run the EIC RCF/SDCC environment with the nightly builds on your local computers or on external high-performance computing clusters.

This repository is aimed for users offsite to [the BNL RACF computer center](#). This repository includes the instruction and local update macro for this Singularity container, which ensures binary reproducible simulation and reconstruction.

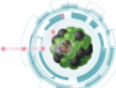
Option-2 CI validations: `updatebuild.sh --build=new` build running

standard macros git tutorials git code reference Doxygen last commit february

# How do I navigate around all this mountain of code?

Start w/ search  
anything here

- ▶ [ecce-eic.github.io/doxygen/](https://ecce-eic.github.io/doxygen/)
- ▶ All software on GitHub now digested in Doxygen via JenkinsCI
- ▶ Auto built at change of any repository
- ▶ Search anything in Doxygen site for [ECCE GitHub software](https://ecce-eic.github.io/doxygen/)

 **ECCE @ EIC Software**  
Reference for ECCE @ EIC simulation and reconstruction software on GitHub

Home page | Related Pages | Modules | Namespaces | **Classes** | Files | External Links | Search

Class List | Class Index | Class Hierarchy | Class Members

**ReadEICFiles Class Reference**

#include <coresoftware/blob/master/simulation/g4simulation/g4main/ReadEICFiles.h>

▼ Inheritance diagram for ReadEICFiles:

```
graph BT
    Fun4AllBase --> SubsysReco
    SubsysReco --> ReadEICFiles
    PHHepMCGenHelper --> ReadEICFiles
```

► Collaboration diagram for ReadEICFiles:

Public Member Functions

ReadEICFiles (const std::string &name="EICReader")
virtual ~ReadEICFiles ()
int Init (PHCompositeNode *topNode)
int process_event (PHCompositeNode *topNode)
bool OpenInputFile (const std::string &name)
void SetFirstEntry (int e)
void SetNodeName (const std::string &s)

► Public Member Functions inherited from SubsysReco

► Public Member Functions inherited from Fun4AllBase

► Public Member Functions inherited from PHHepMCGenHelper


# Fun4All Jenkins-CI Pipeline Workflow

- CI check in upstream sPHENIX-Fun4All repo
- Help needed to implement a CI QA pipeline in ECCE too

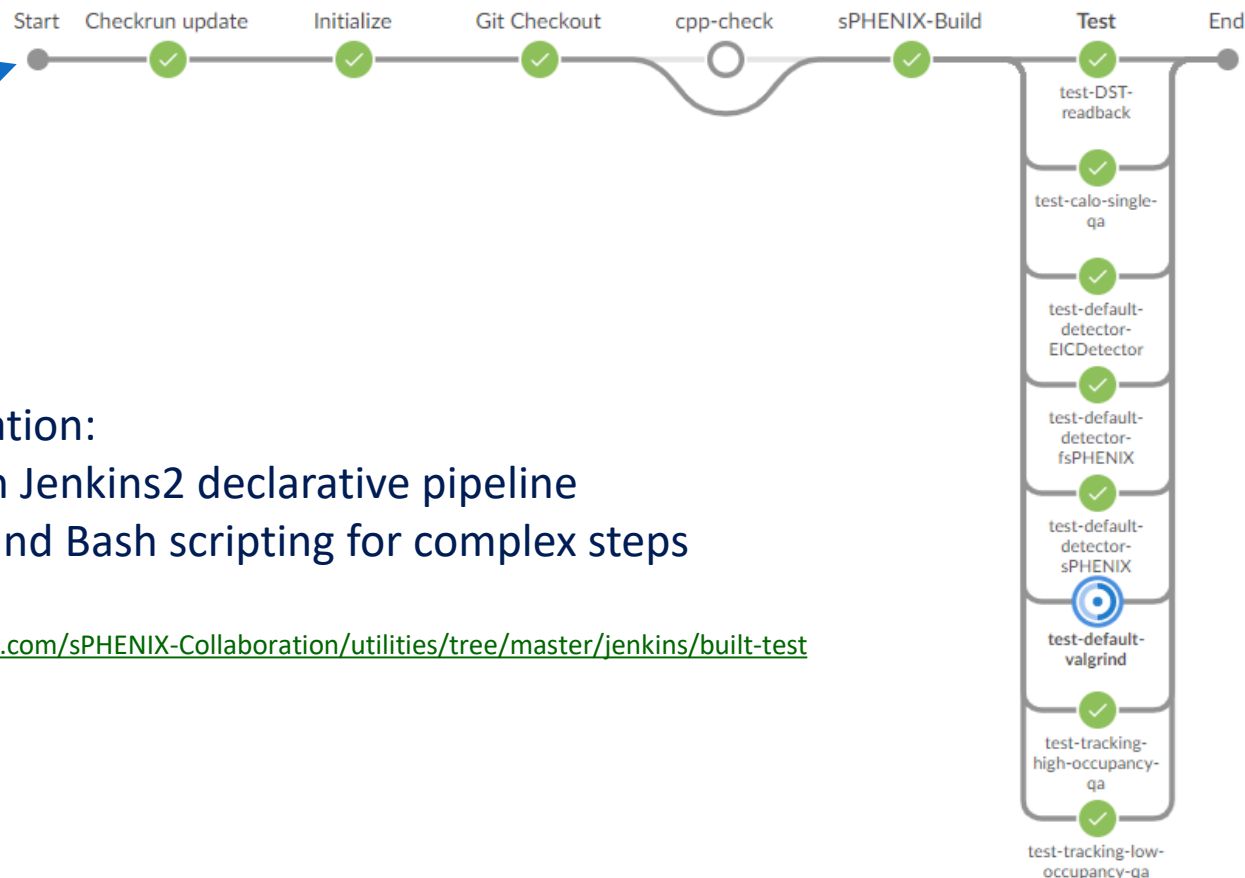
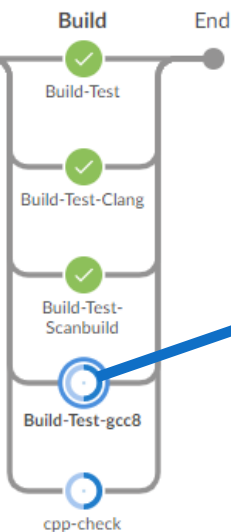
sphenix-jenkins-ci bot commented 9 days ago

### Build & test report

Report for commit `150f6adc21b60a2d67f59fc797e78700337b`:



- **build** passing builds and tests overall are SUCCESS.
- **build** passing Build with configuration of `gcc-8.3 / new` is SUCCESS. Compiler report (full)/(new), build log
  - **build** passing Generating DST and readback: build is SUCCESS
  - **build** passing Calorimeter QA: build is SUCCESS
    - QA-calorimeter for e- at  $p_T=4\text{GeV}$ : combined  $\chi^2/\text{nDoF} = -0 / 72$ , and combined p-Value = 1
    - QA-calorimeter for pi+ at  $p_T=30\text{GeV}$ : combined  $\chi^2/\text{nDoF} = -0 / 72$ , and combined p-Value = 1
    - QA-calorimetric-jet for e- at  $p_T=4\text{GeV}$ : combined  $\chi^2/\text{nDoF} = -0 / 42$ , and combined p-Value = 1
    - QA-calorimetric-jet for pi+ at  $p_T=30\text{GeV}$ : combined  $\chi^2/\text{nDoF} = -0 / 42$ , and combined p-Value = 1
  - **build** passing Tracking QA at high occupancy: build is SUCCESS
    - QA-Intt: combined  $\chi^2/\text{nDoF} = -0 / 72$ , and combined p-Value = 1
    - QA-Mvtx: combined  $\chi^2/\text{nDoF} = -0 / 54$ , and combined p-Value = 1
    - QA-Tpc: combined  $\chi^2/\text{nDoF} = -0 / 56$ , and combined p-Value = 1
    - QA-tracking: combined  $\chi^2/\text{nDoF} = 3.09879 / 38$ , and combined p-Value = 1
    - QA-vertexing: combined  $\chi^2/\text{nDoF} = 0.000285769 / 98$ , and combined p-Value = 1
  - **build** passing Tracking QA at low occupancy: build is SUCCESS
    - QA-Intt: combined  $\chi^2/\text{nDoF} = -0 / 72$ , and combined p-Value = 1
    - QA-Mvtx: combined  $\chi^2/\text{nDoF} = -0 / 54$ , and combined p-Value = 1
    - QA-Tpc: combined  $\chi^2/\text{nDoF} = -0 / 56$ , and combined p-Value = 1
    - QA-tracking: combined  $\chi^2/\text{nDoF} = -0 / 42$ , and combined p-Value = 1
    - QA-vertexing: combined  $\chi^2/\text{nDoF} = -0 / 98$ , and combined p-Value = 1
  - **build** passing system `gcc-8.3`, build `new`: run the default EICDetector macro: build is SUCCESS, output
  - **build** passing system `gcc-8.3`, build `new`: run the default sPHENIX macro: build is SUCCESS, output
  - **build** passing system `gcc-8.3`, build `new`: run the default sPHENIX macro: build is SUCCESS, output
  - **build** unstable system `gcc-8.3`, build `new`: Valgrind test: build is UNSTABLE, valgrind report
- **build** passing Build with configuration of `x8664_s17 / clang` is SUCCESS. clang report (full)/(new), build log
- **build** passing Build with configuration of `x8664_s17 / new` is SUCCESS. Compiler report (full)/(new), build log
- **build** passing Build with configuration of `x8664_s17 / scan` is SUCCESS. scan-build report (full)/(new), build log
- **build** passing `cpp-check`: is SUCCESS, cppcheck report (full)/(new)



## Implementation:

- Based on Jenkins2 declarative pipeline
- Python and Bash scripting for complex steps
- Git SCM:

<https://github.com/sPHENIX-Collaboration/utilities/tree/master/jenkins/built-test>

e.g. <https://github.com/sPHENIX-Collaboration/coresoftware/pull/1117#issuecomment-805131915>

# Software modules



# Event generators

- ▶ Running example given in sample EIC sim macro:  
[https://github.com/ECCE-EIC/macros/blob/master/detectors/EICDetector/Fun4All\\_G4\\_EICDetector.C](https://github.com/ECCE-EIC/macros/blob/master/detectors/EICDetector/Fun4All_G4_EICDetector.C)
- ▶ Arbitrary set-of-particle generators for testing
- ▶ Main event generator input format: [HepMC files](#)
- ▶ Built-in generators : [PYTHIA8.3](#), [PYTHIA6](#), [SARTRE](#)
- ▶ Reads [events generated](#) in [EIC-Smear package](#) [talk: Kolja]
  - Share the same Singularity image and CVMFS vol. Source different setup macros.
  - Allow comparison of same event sample processed by EIC-Smear and by the Fun4All simulation-reco.
- ▶ Support multiple [background pile ups](#)

Note: All EIC full event generator has EIC beam parameters applied by default: collision crossing angle, beam divergence, vertex smear, crab cavity kick [[ref](#)]. Also can switch off in the main macro

<https://eic.github.io/software/mcgen.html>

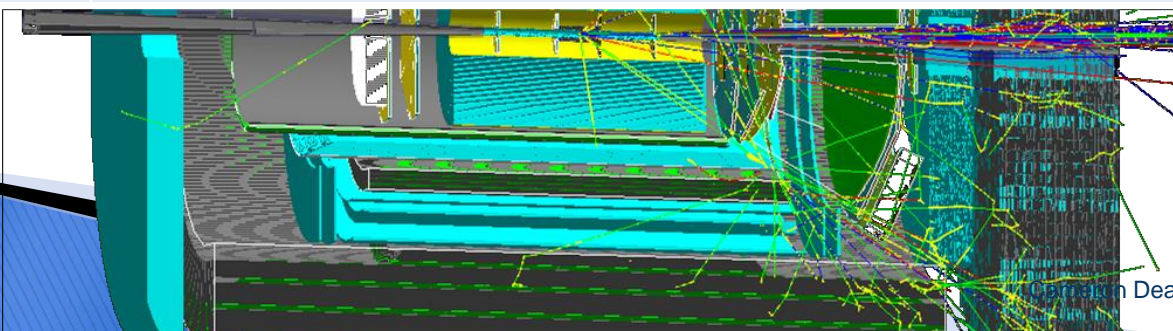
## Monte Carlo Event Generators

- PYTHIA6
- BeAGLE
- DJANGO
- MILOU
- RAPGAP
- PEPSI
- eSTARlight (external link)
- Sartre (external link)

# Status of detector descriptions

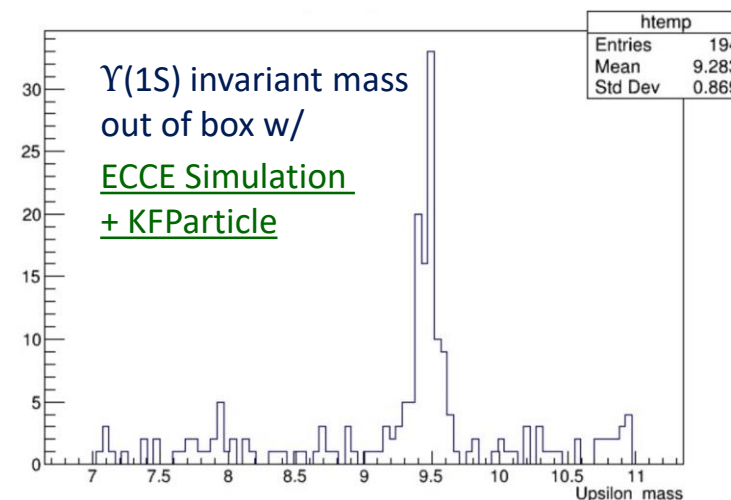
- ▶ Full detector model provides digitization and various level of reconstruction chain
- ▶ For fast prototyping, fast detector model can be built with simple shapes on macro level
- ▶ Adding detectors: see talk Friedericke Bock & Nicolas Schmidt

Detector	Status / link to code	Help needed
Silicon trackers	<a href="#">Full model for ALPIDE</a> , fast model for <a href="#">ITS3</a> , <a href="#">FST</a>	Update ECCE setup
TPC	<a href="#">Full model</a> , <a href="#">fast model</a>	EIC setup, dE/dx
MPGD tracker	<a href="#">Fast model</a>	Update ECCE setup
Barrel Calorimeter	Full model for <a href="#">SPACAL</a> , <a href="#">sPHENIX HCals</a> , fast model for <a href="#">Shashlyke</a>	Update ECCE setup
Forward calorimeter	Full model for <a href="#">Shashlyke</a> , <a href="#">PbScifi</a> , fast model for <a href="#">crystal calorimeter</a>	Light collection uniformity
PID / TOF	Full model for <a href="#">MRPC</a> and Fast model for <a href="#">LGAD</a>	Update ECCE setup
PID / RICH	Full model for <a href="#">mRICH</a> , <a href="#">Gas RICH</a> ; missing DIRC and dual RICH (material placeholder)	dRICH, RICH reco
Far forward	Fast model [Talk: Ciprian]	Beamline material



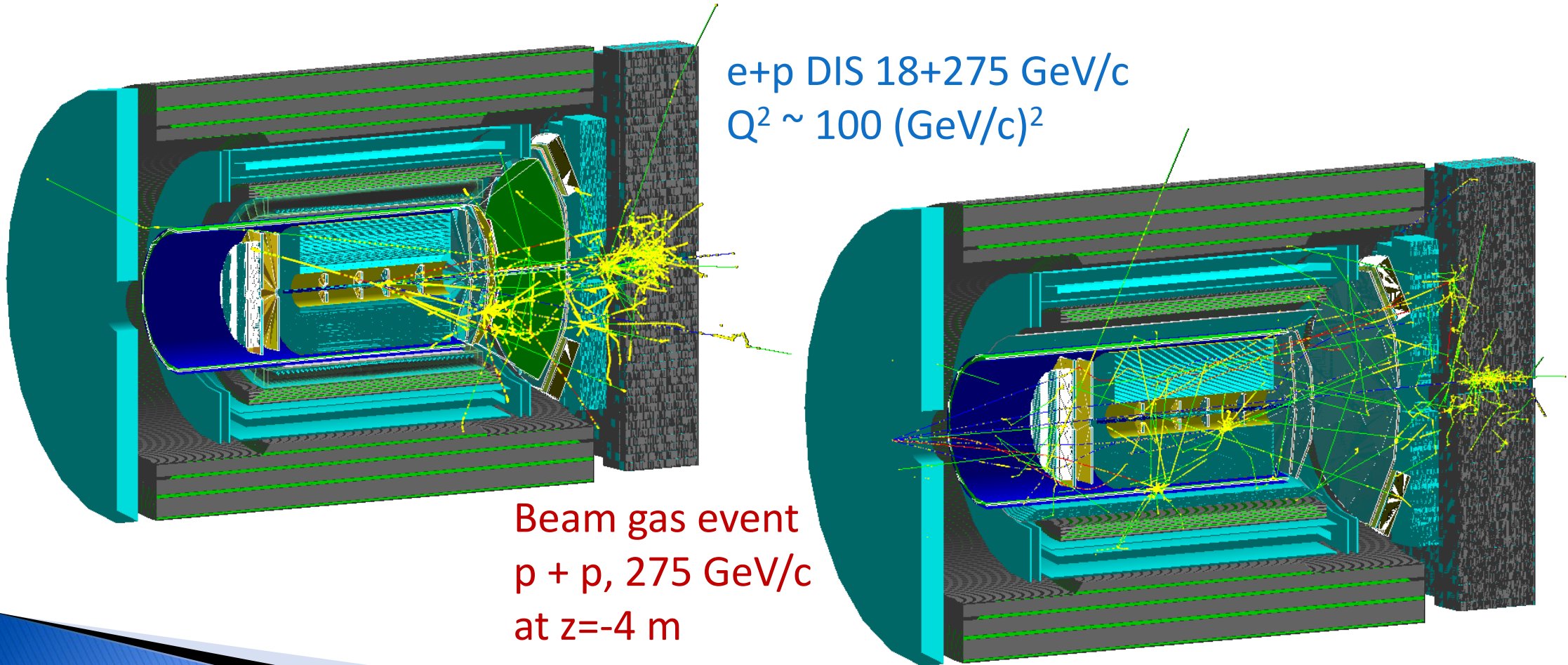
# Reconstruction

- ▶ Tracking:
  - Use GenFit2 for fast prototyping (PHG4TrackFastSim), widely used in YR tracking studies
  - sPHENIX switched to ACTS: fast to fit but long development time for adopting new tracker set **Not** suited for ECCE at this stage [help needed: pattern reco. with ACTS]
- ▶ Calo reco:
  - Clusterizers, FastJet, Particle flow jet (prototype, need volunteer)
- ▶ PID reco:
  - GenFit2 for TOF, e.g. track length and timing smearing
  - RICH has to rely on fast smearing (help needed: full reco of RICH need major development)
- ▶ Resonance search/HF reconstruction:
  - KFParticle
- ▶ Detailed truth tracing evaluation chain
  - e.g. what portion of reco jet is from a truth jet



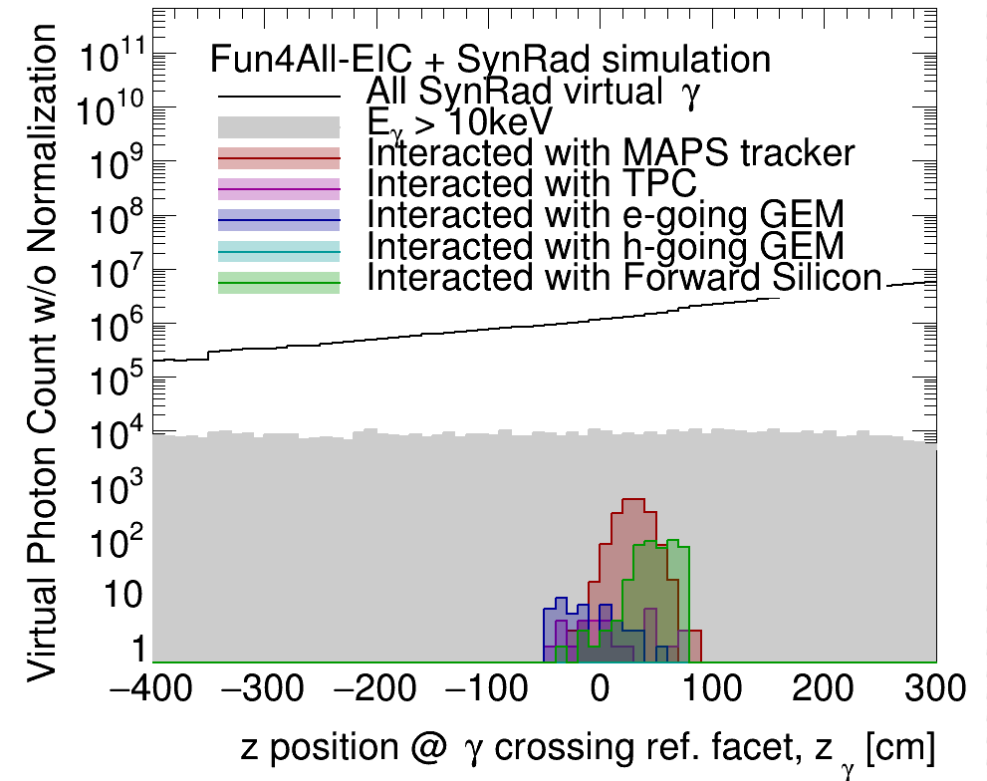
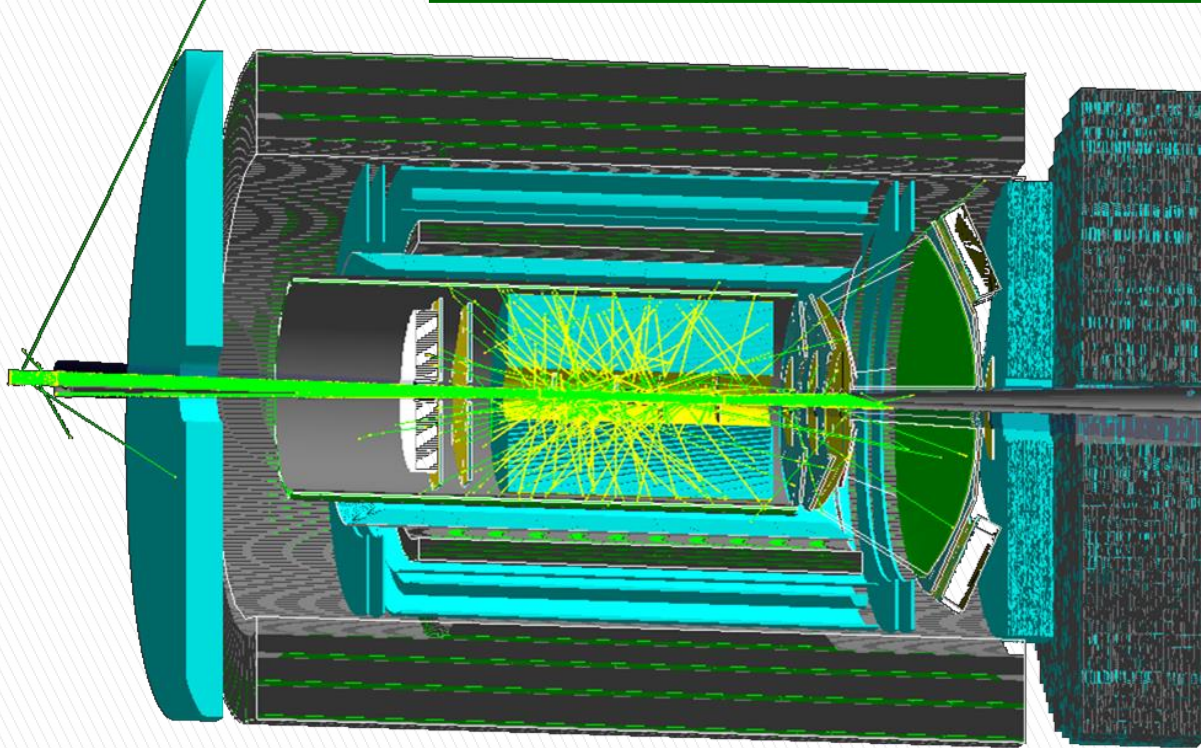
# Background embedding: beam gas interaction

- ▶ Built-in Pythia8 for beam gas interaction background generation



# Background embedding : Synchrotron radiation

- Fun4All has [interface to input Synchrotron Photon simulation](#), used in Synchrotron study leading to CD-1 review




100k Synchrotron photon in full detector simulation [YR Fig 10.12]

Detector background as function of beam-pipe exit-location

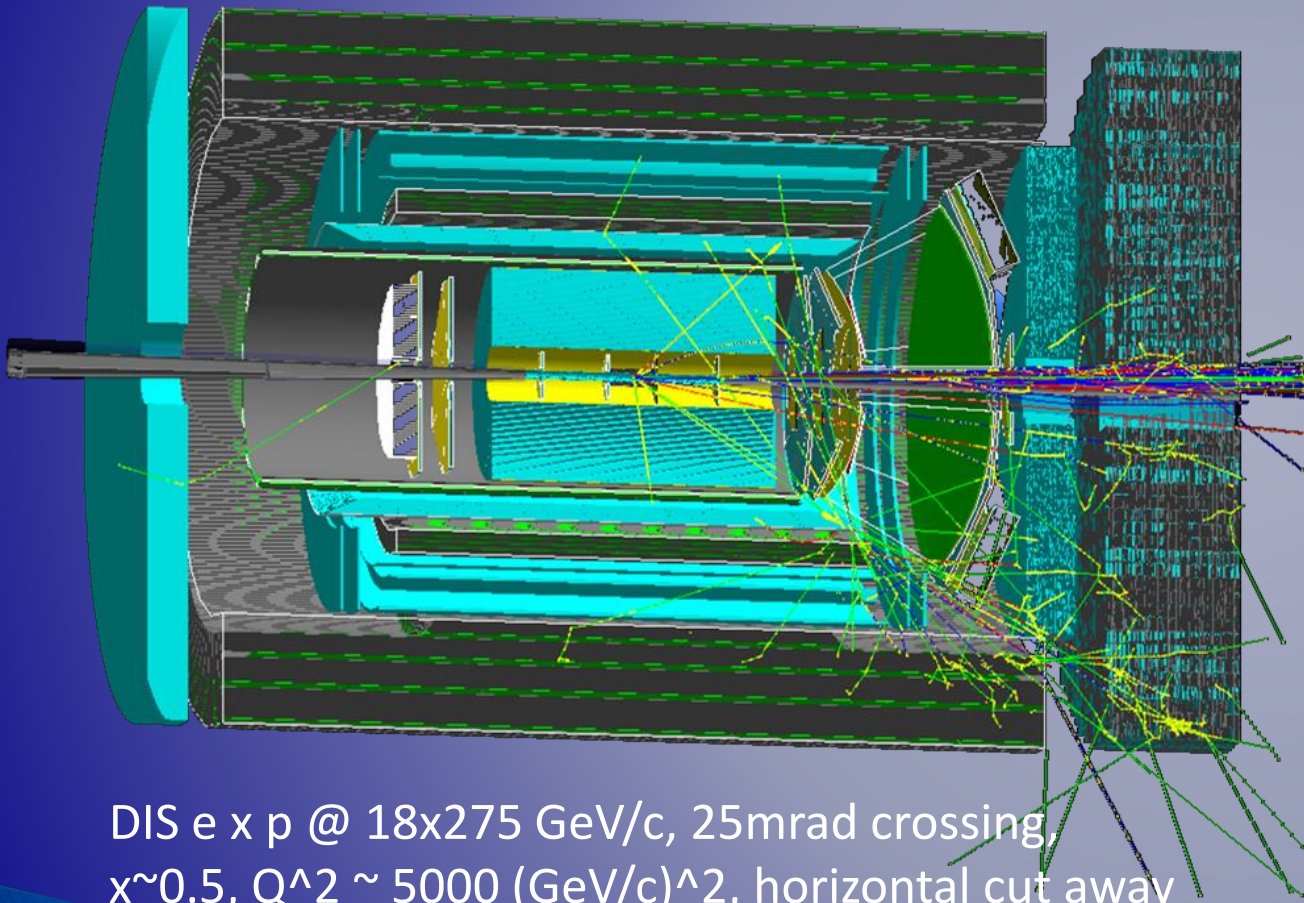
Note: all photons simulated for detector interaction, without cuts on z or energy. EIC/July-2020 lattice & chamber

# Summary

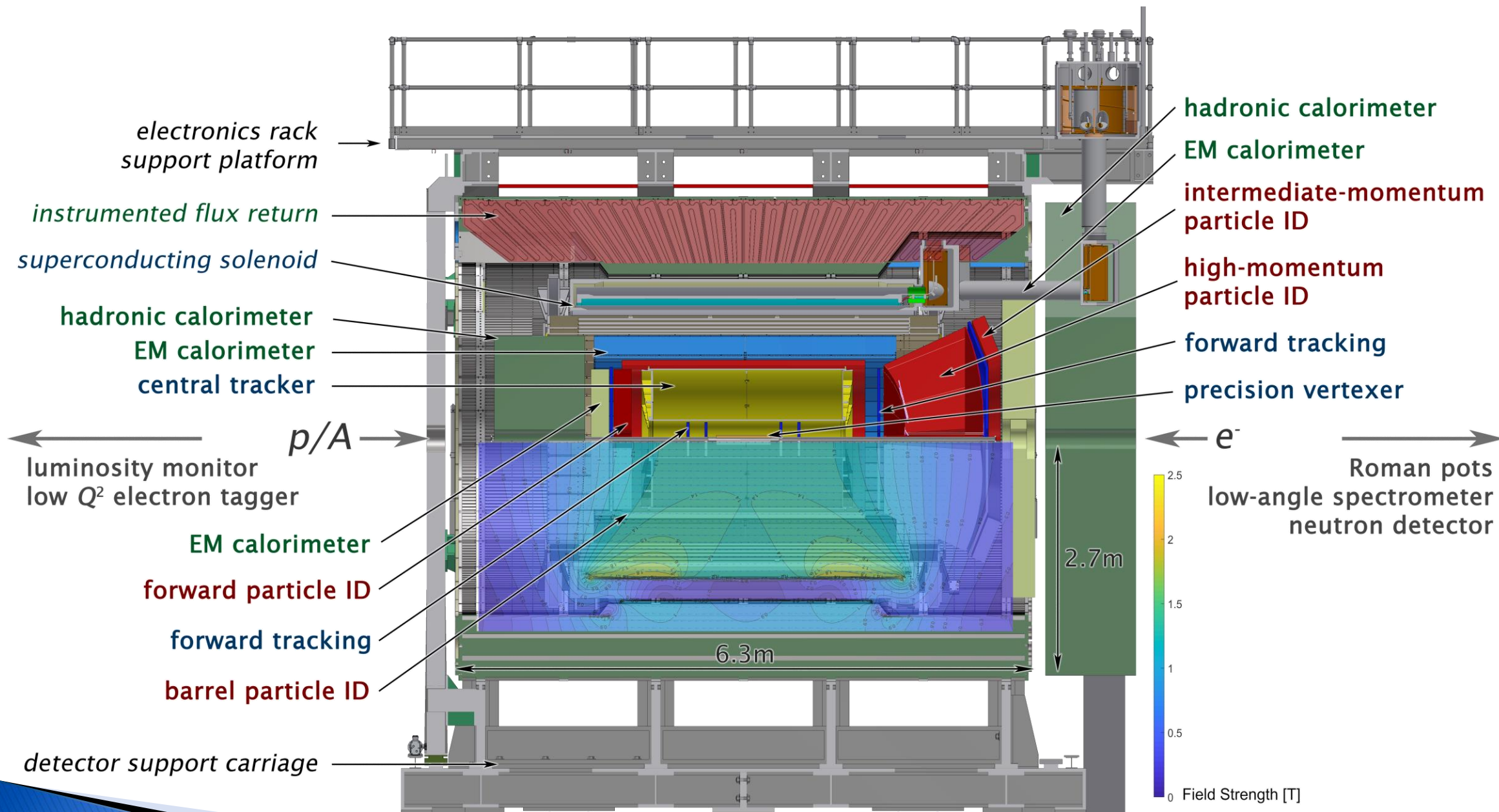
- ▶ Time is short
- ▶ Good foundation for ECCE simulation based on Fun4All
- ▶ Will help WGs design simulation path for each specific study
- ▶ Help needed in many area
- ▶ Don't hesitate to ask questions
- ▶ Send [jhuang@bnl.gov](mailto:jhuang@bnl.gov)
  - Your GitHub username to join <http://github.com/ECCE-EIC>
  - If no BNL account, email to enter [MatterMost chat](#)

- 
- Dec 1<sup>st</sup> : proposal submission
  - Nov 1<sup>st</sup>: final proposal for editing
  - Sept 1<sup>st</sup> : all major simulation plot done. After this date, we just do polishing, composing narratives around the figures in the performance chapter of the proposal.
  - Aug 1<sup>st</sup> : Final simulation production done
  - July 1<sup>st</sup> : Final simulation production start
  - May 1<sup>st</sup>: First simulation campaign, followed by first round of analysis. From May – July, many studies probably need another iteration of simulation-analysis to advance detector design.
  - Entire Apr: develop simulation setup to run.
  - Today: 1<sup>st</sup> simulation workshop

# Extra information



DIS e x p @ 18x275 GeV/c, 25mrad crossing,  
 $x \sim 0.5$ ,  $Q^2 \sim 5000 \text{ (GeV/c)}^2$ , horizontal cut away



# EIC x-sec : further quantification [Courtesy E. Aschenauer]

- ▶ Inelastic e+p scattering x-sec:
  - For a luminosity of  $10^{34} \text{ cm}^{-2}\text{s}^{-1}$  50ub corresponds to **500 kHz**
- ▶ Elastic e+p cross-section:
  - For EIC central barrel, elastic cross section is **small** comparing to the inclusive QCD processes
- ▶ Beam gas interaction:
  - Beam proton – beam gas fix target inelastic interactions. The pp elastic cross section is smaller ( $\sim 7 \text{ mb}$ )
  - For a vacuum of  $10^{-9} \text{ mbar}$  in the detector volume (10m) this gives a rate of **14 kHz**

Beam [GeV]	HERA	5 x 50	10 x 100	18 x 275
$Q^2 > 10^{-9} \text{ GeV}$	65.6	<b>29.9</b>	<b>41.4</b>	<b>54.3 ub</b>
$Q^2 > 1 \text{ GeV}$	1.29	0.45	0.65	0.94 ub

Beam [GeV]	HERA	5 x 50	10 x 100	18 x 275
$\sigma [y_{\text{Exp}} > -4]$	5 pb	<b>5 ub</b>	<b>0.7 ub</b>	<b>0.06 ub</b>
$\sigma [y_{\text{Exp}} > -6]$	11 ub	420 ub	100 ub	29 ub

$E_p$ :	50 GeV	100 GeV	275 GeV	920 GeV
	<b>38.4 mb</b>	<b>38.4 mb</b>	<b>39.4 mb</b>	41.8 mb

# Data Rate

## MAPS silicon tracker

## TPC

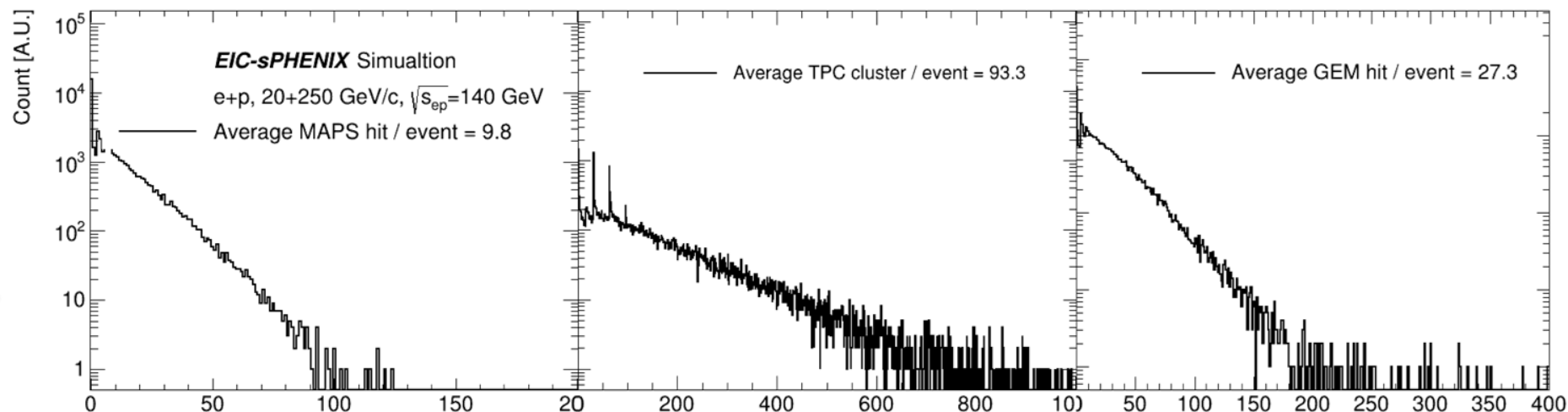
## Forward/backward GEM

Raw data: 16-24 bit / MAPS hit  
(3-layer ALPIDE model)

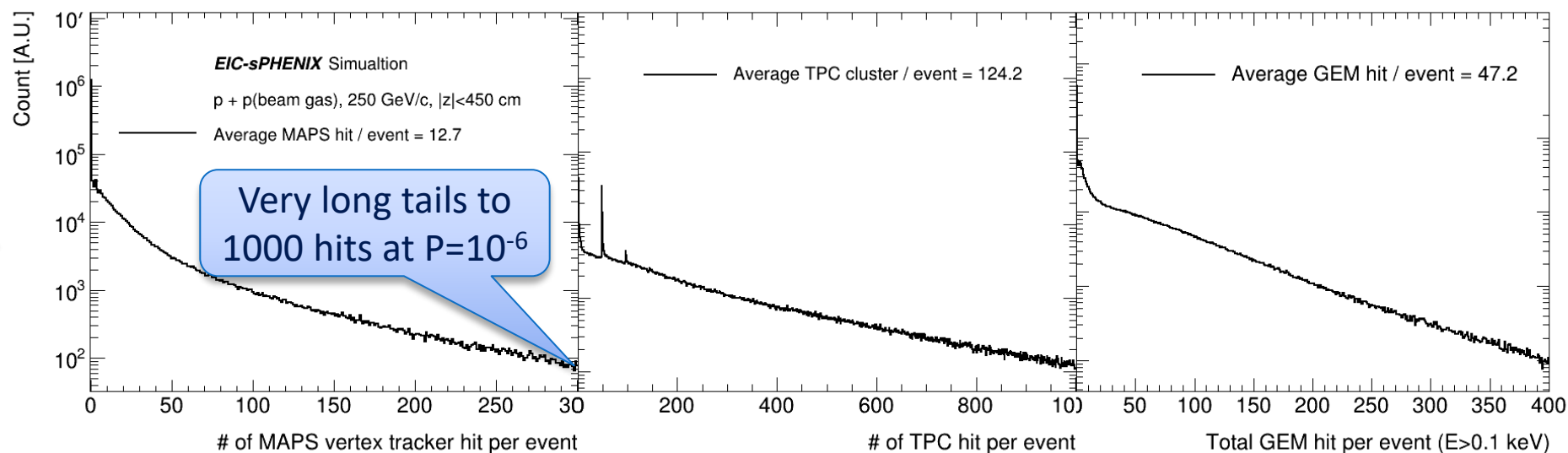
Raw data: 3x5 10 bit / TPC hit  
+ headers (60 bits)

Raw data: 3x5 10 bit / GEM hit  
+ headers (60 bits)

e+p, Pythia6 Q2>0



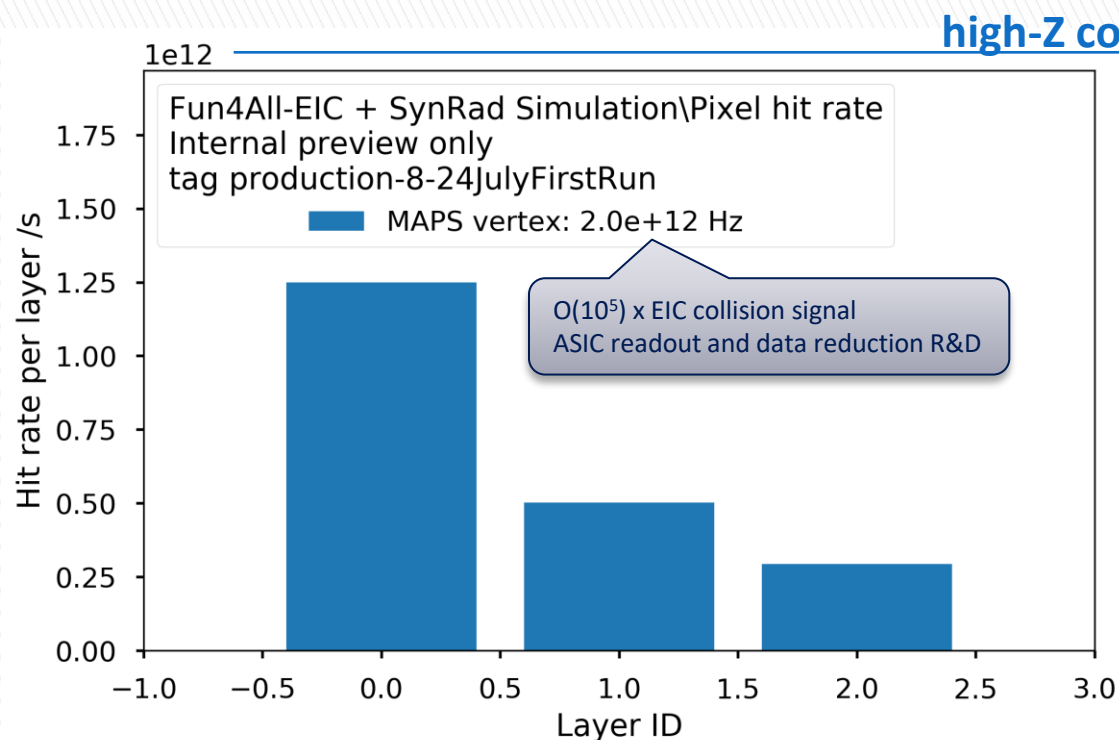
p+p(gas) Pythia8



Refs: sPH-cQCD-2018-001: <https://indico.bnl.gov/event/5283/>

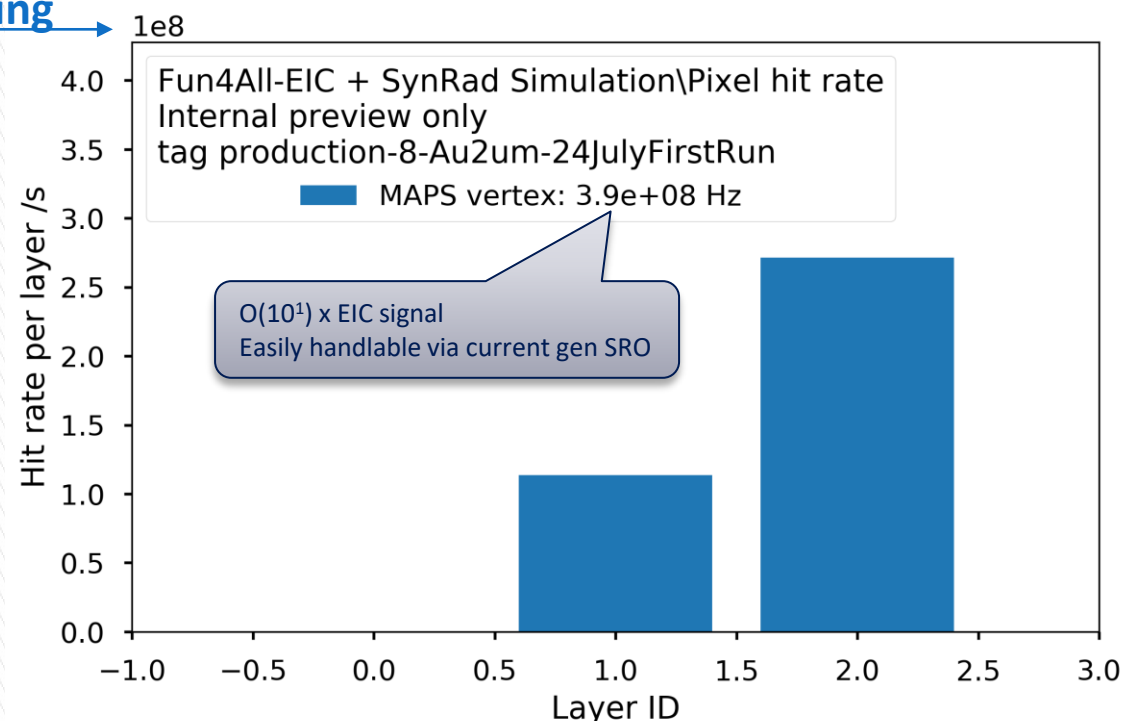
# Synchrotron background: detector response

- In the most recent lattice + beam chamber geometry, there is a known issue with main dipole fan reflect over far upstream beam chamber to Be-beam pipe section.
- Beam chamber tuning on-going, expect to reduce by orders of magnitude [DO NOT QUOTE THIS RATE]
- The reflected dipole fan induce high hit rate in barrel detectors prior to photon shield tuning, but high-Z coating on chamber, e.g. 2- $\mu\text{m}$  Au coating ( $0.06 X_0$ ) on Be pipe significantly reduces the synchrotron rate



Default 760 $\mu\text{m}$ -Be beam pipe

Dominated by dipole fan reflection. Expected to reduce with tuning



High-Z-coated beam pipe (+2 $\mu\text{m}$  Au)

Dominated by dipole fan reflection. Expected to reduce with tuning