Status of STCF Software

Xingtao Huang (SDU)

On behalf of STCF software group

The 2021 Workshop on future Super c-tau factories

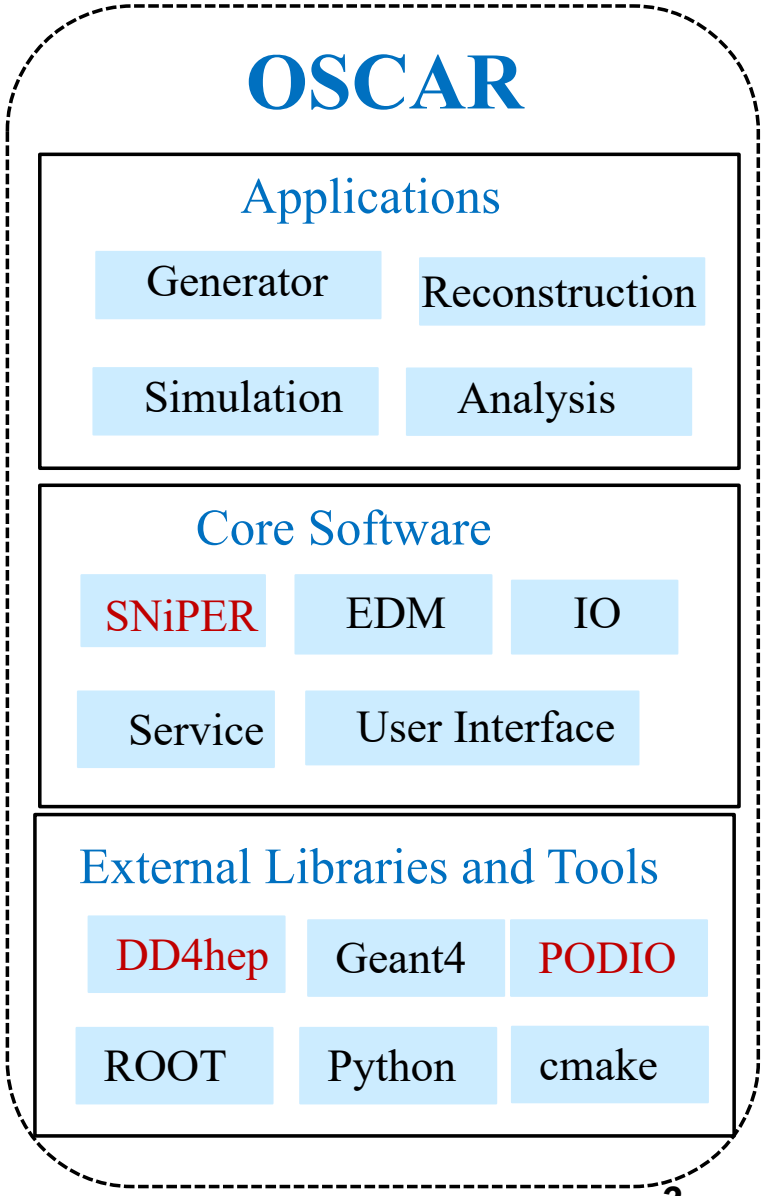
November 15-17, 2021 (online)

Outline

- Introduction
- Core software (EDM)
- Detector Geometry and Simulation
- Reconstruction Algorithms
- Validation System
- Visualization Tool
- Summary and Future Plan

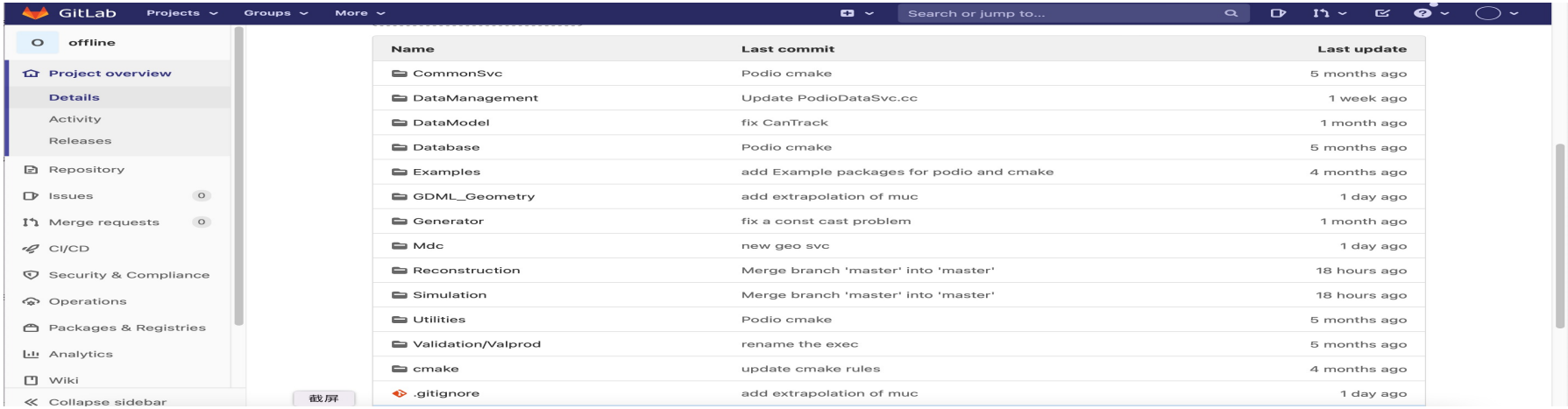
Introduction

- The Offline Software of Super Tau-Charm Facility （OSCAR） was developed based on SNiPER framework since Oct. 2018.
  - Core software
  - Applications
  - External libraries and tools
- SNiPER
  - a light-weight framework
  - support both collider and non-collider HEP Exp.
  - adopted by JUNO, LHAASO, nEXO
  - define interfaces to all software components and controls their execution.
- Partially based on Key4hep
  - PODIO: a generic event data model toolkit
  - DD4hep: a common geometry description toolkit



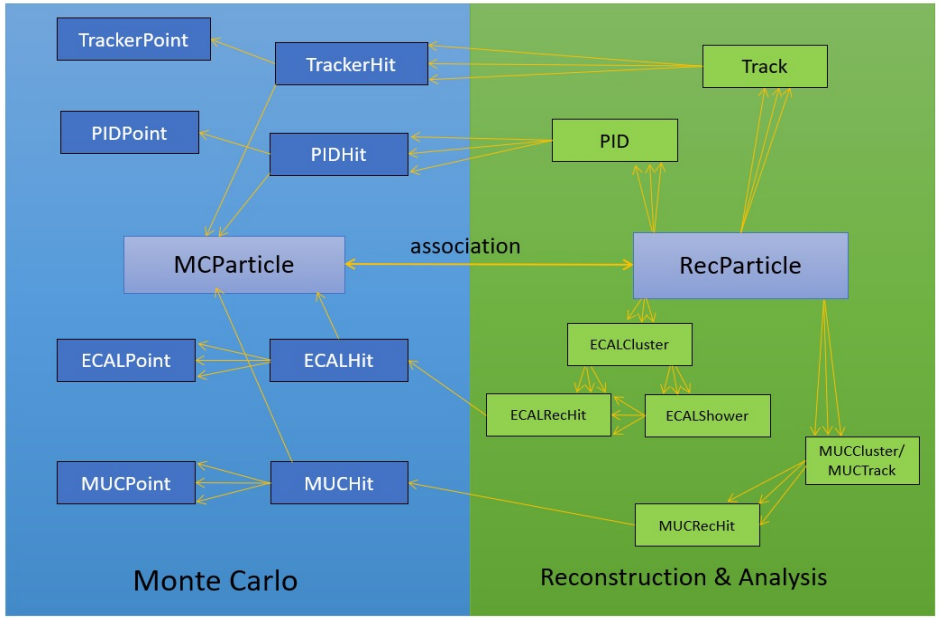
Development environment

- Supported Operating System: SLC 7 and CentOS 7
- Programming Language: C++ 11, Python 2.7
- Configuration Language：Python
- Software Management Tool：CMake
- Version Control Tool：Gitlab
  - The webpage: http://202.141.163.203:8009/oscar
  - 15 developers part-time working on development of OSCAR
  - The pull-request mechanism is used to synchronize all developers’ work
- Users’ Guide: http://202.141.163.203:8009/oscar/documents



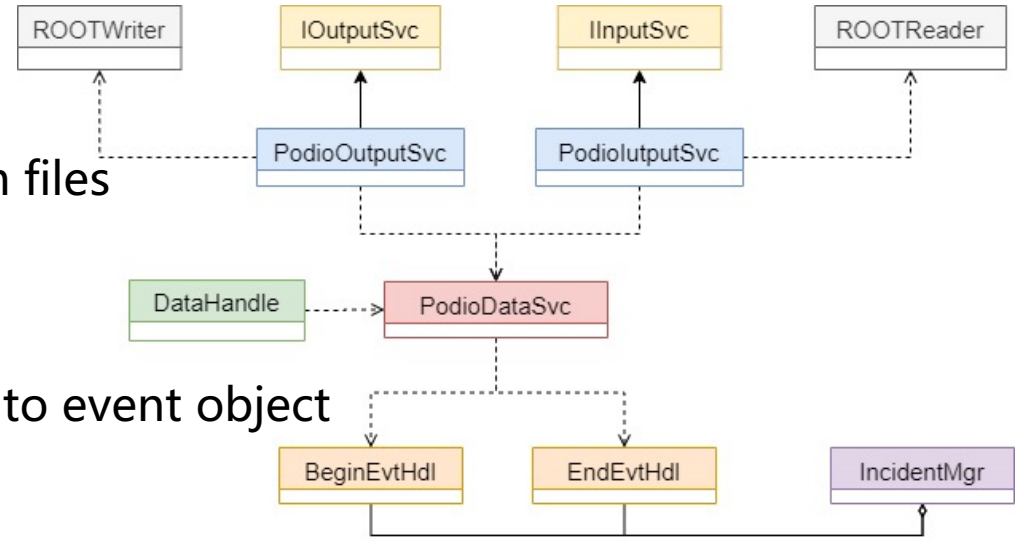
Event Data Model: updated from ROOT to PODIO

- PODIO: new Event Data Model toolkit developed by HEP communality, and used by FCC, CEPC, SCT…
  - Simple memory model
  - support concurrency when design
  - excellent I/O : ROOT, SIO , HDf5
- With PODIO, common core classes is described in YAML file and C++ code is automatically generated
- Re-designed EDM with PODIO for simulation and reconstruction
  - EDM for each sub-detectors is implemented separately with no inheritance relationship
  - Build up one to one/many relations between different POD objects



Event Data Management System

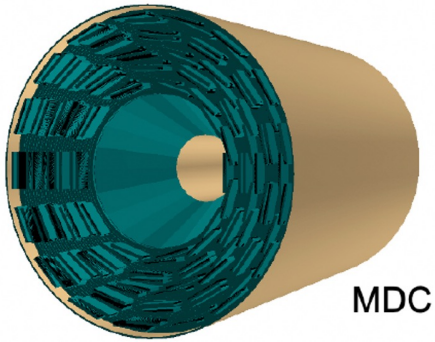
- Developed a new EDM system to integrate PODIO within OSCAR
  - PodioDataSvc
    - memory management
  - PodioInputSvc
    - read the PODIO objects from files
  - PodioOutputSvc
    - write PODIO objects to files
  - Two datahandles provide access to event object
    - getRWColl(…)
    - getROColl(…)

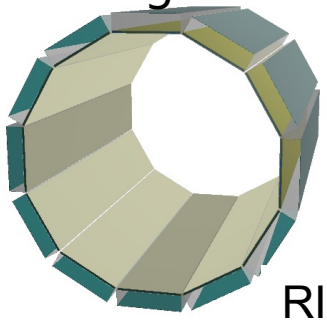


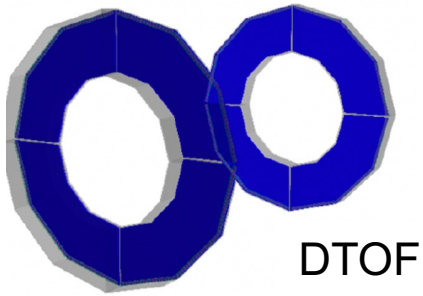

```
ITDHitCollection*       itdhits = getROColl(ITDHitCollection, "ITDHitColl");

ECALHitCollection*     ecalhits = getRWColl(ECALHitCollection, "ECALHitColl");
ECALPointCollection* ecalpoints = getRWColl(ECALPointCollection, "ECALPointColl");
```

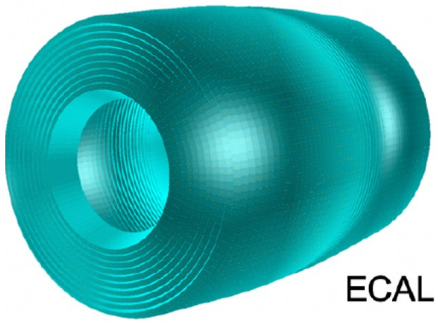
Detector Geometry Description

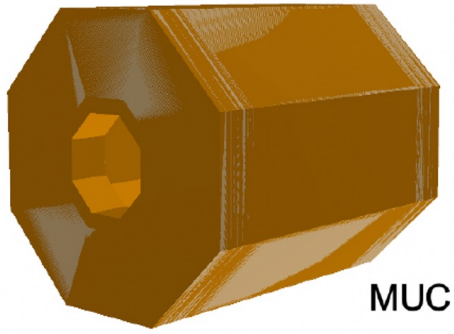
- The Full STCF Detector is described with DD4hep
  - Use a single source for detector simulation, reconstruction and visualization
  - Each sub-detector is implemented with a single compact file
  - The version number is used for different design options
  - Optimizing the detector geometry according to changes of the detector design

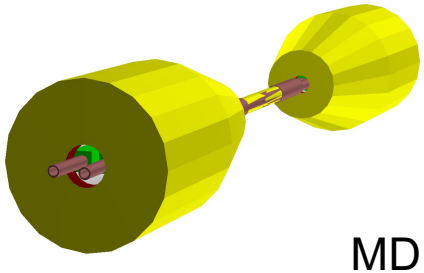


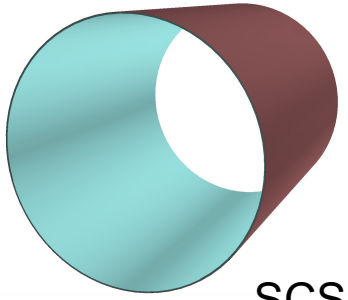








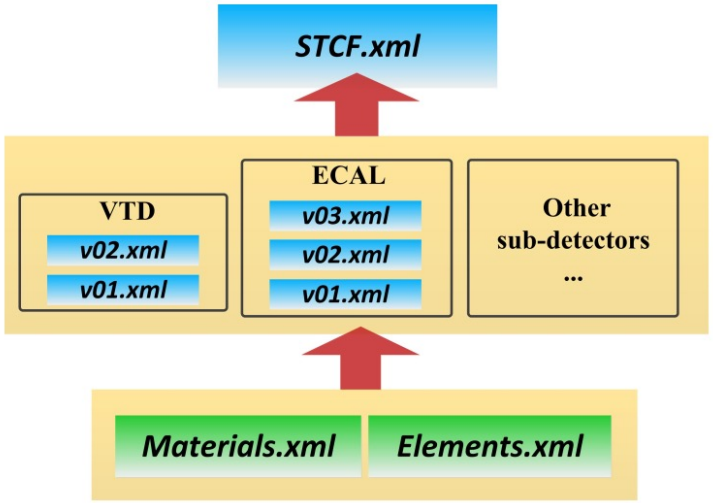




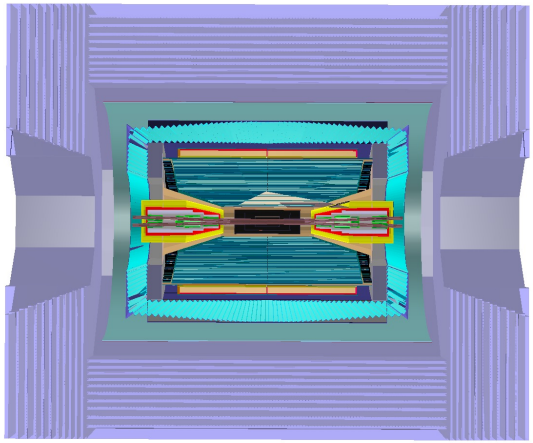
Detector Geometry Managment

- Developed the detector geometry management system (GMS)*
  - The full detector can be easily build up with different sub-detector design options
  - Support single sub-detector simulation and the full detector simulation

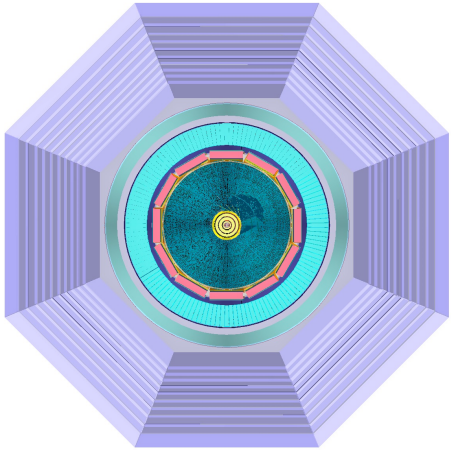
More details in next talk by He



Structure of the geometry parameters repository



cross view in the r-z plane

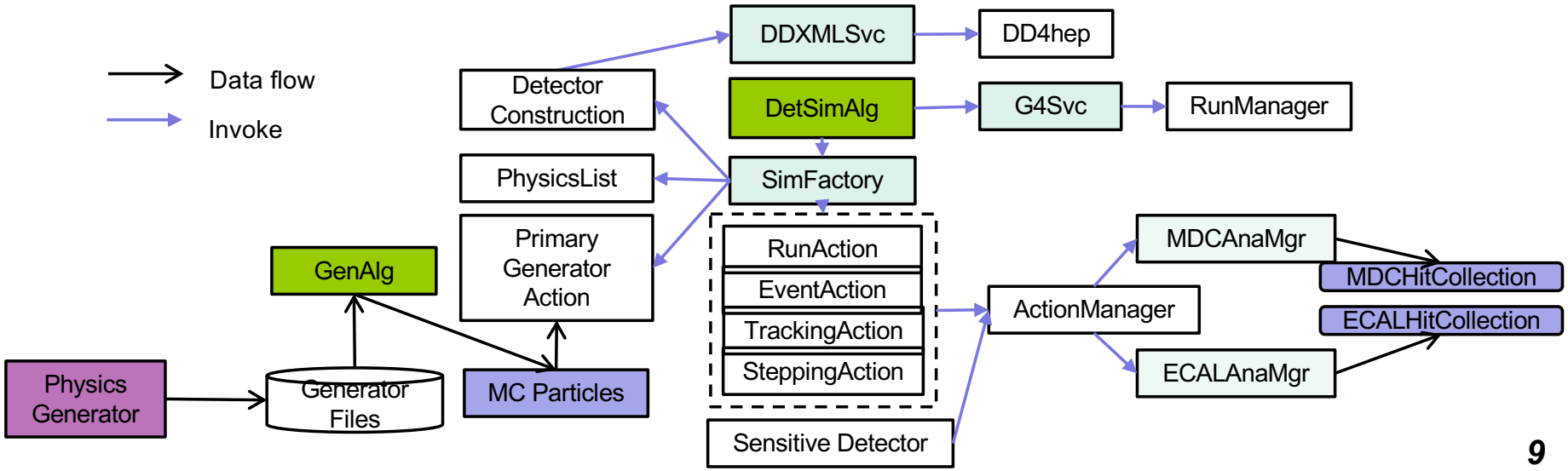


cross view in the x-y plane

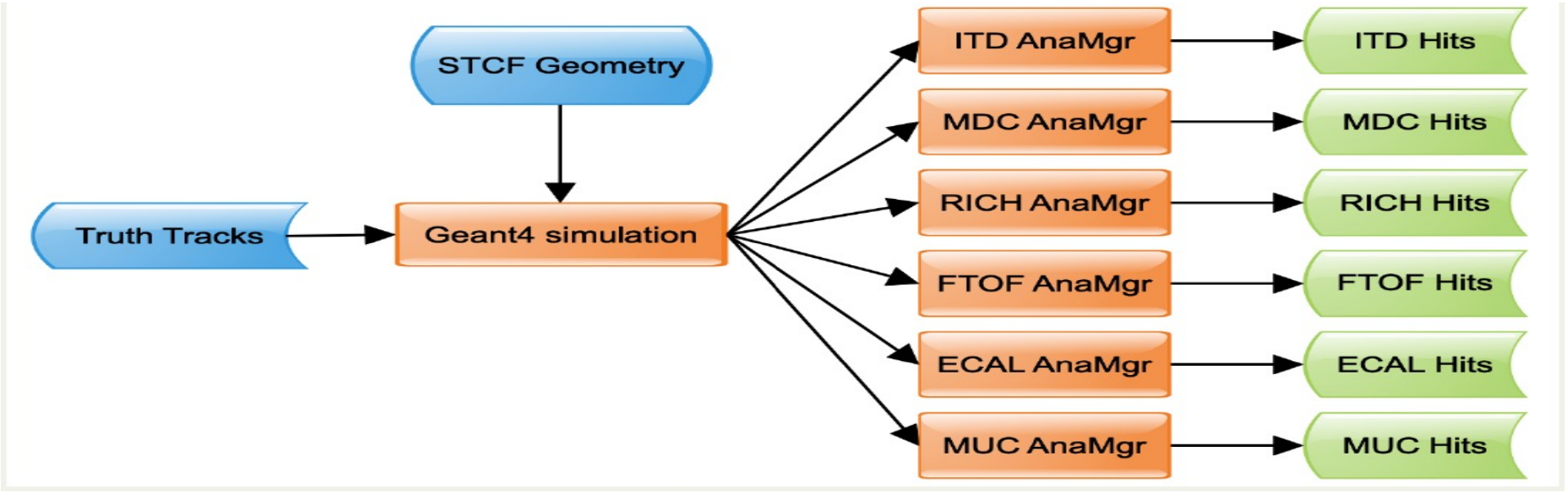
* "*Detector geometry management system designed for Super Tau Charm Facility offline software*", published on JINST 2021_JINST_16_T04004

Detector Simulation

- The whole detector simulation chain is completed
  - Developed a unified generator interface to configure different physics generators
    - Babayaga, Phokhara, KKMC, EvtGen, DIAG 36, etc.
  - developed a new service (DDXMLSvc)
    - Deliver detector geometry from DD4hep to both Geant4 and recon. algorithms
    - Provide the user interface to configure Sensitive Detector
  - One analysis manager is mandatory for each sub-detector to retrieve Geant4 simulation information and save them into the PODIO objects



Single Sub-detector or Full Detector Simulation



- Only record simulation information from ECAL

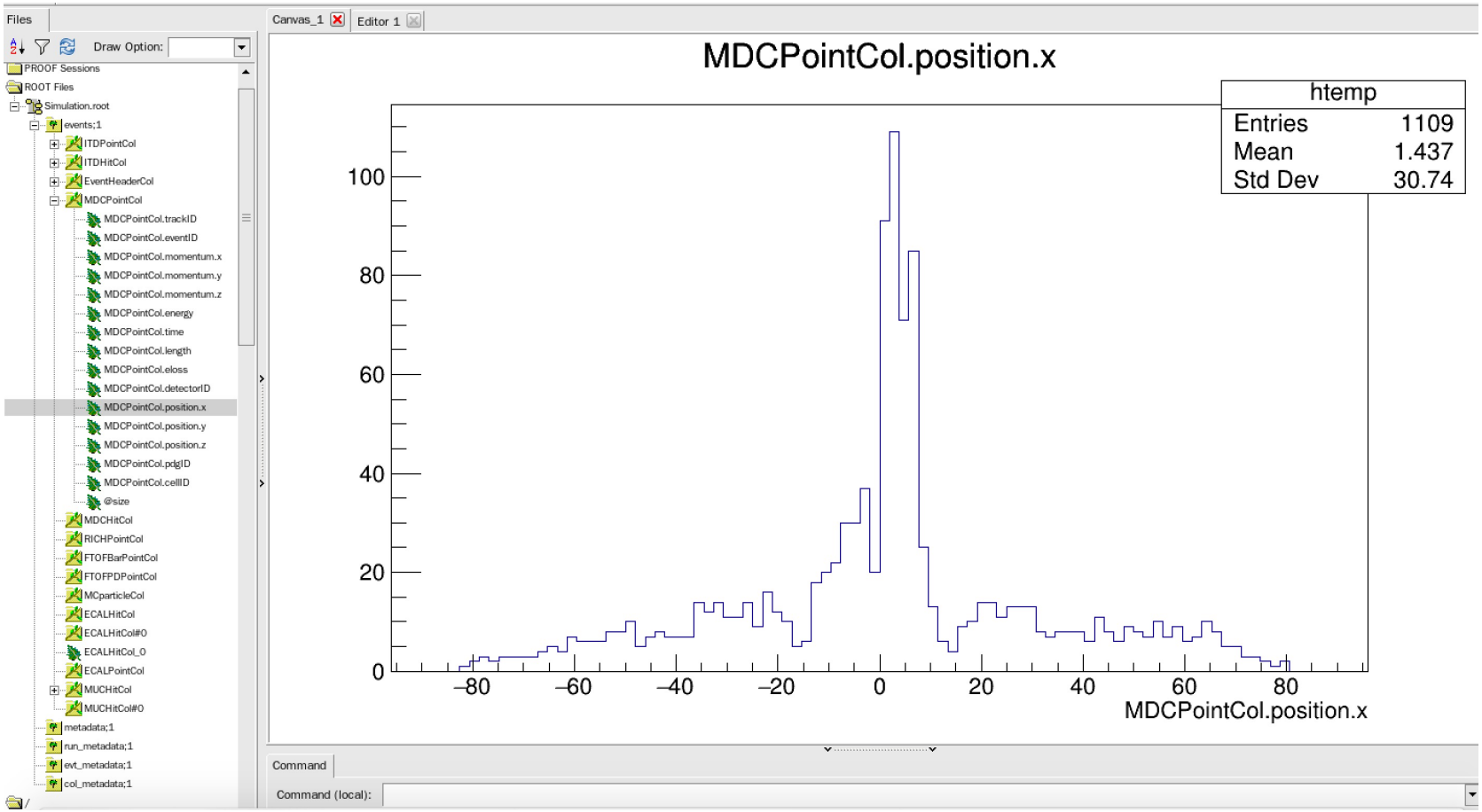
```
factory = task.createSvc("FullSimFactory/FullFacory")
factory.property("AnaMgrList").set(["GeneratorMgr","ECALAnaMgr"])
```

- Record simulation information from Full detector

```
factory = task.createSvc("FullSimFactory/FullFacory")
factory.property("AnaMgrList").set(["GeneratorMgr","ITDAnaMgr","MDCAnaMgr","FTOFA
naMgr", "RICHAnaMgr") "ECALAnaMgr","MUCAnaMgr")
```

Finished Transition from ROOT EDM to PODIO

- All simulation algorithms have been updated
- New simulated results are consistent with old ones
- Found one bug of PODIO and asked developers to fix it

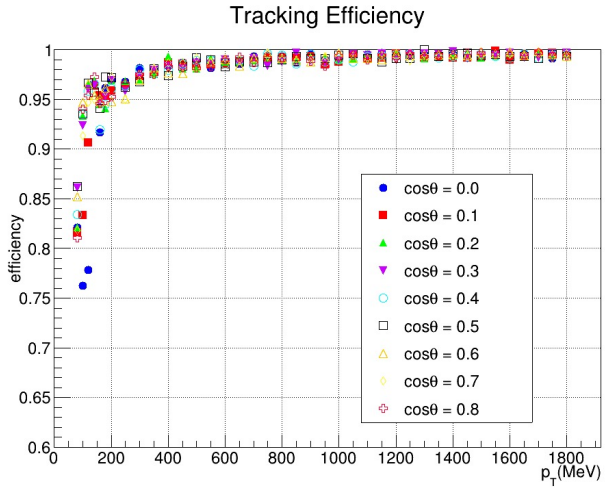


Tracker Reconstruction

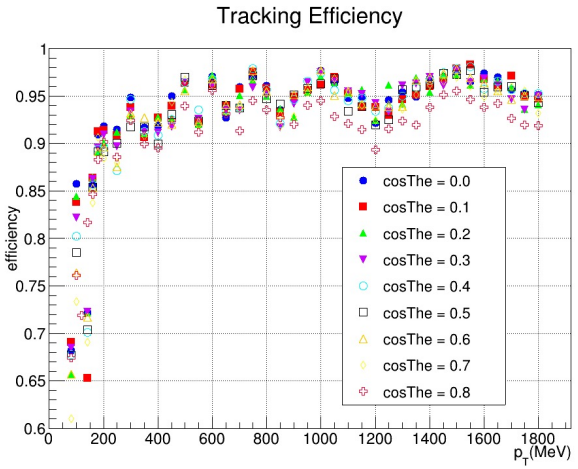
- Developed Track finding algorithm based on Hough transformation

MDC hits
ITD hits → Conformal transformation → Hough(Legendre) transformation → Histogram filling Peak finding → Selection → Least Squares Fitting

- Track fitting algorithm is performed by the Deterministic Annealing Filter (DAF) method, an extension of Kalman Filter , in GENFIT2
- Performance study with/without noise



No noise

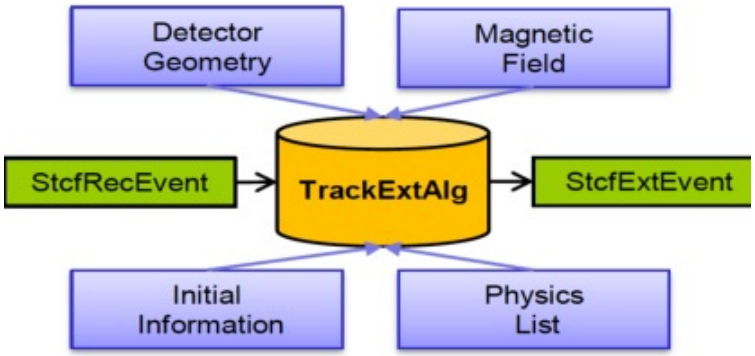


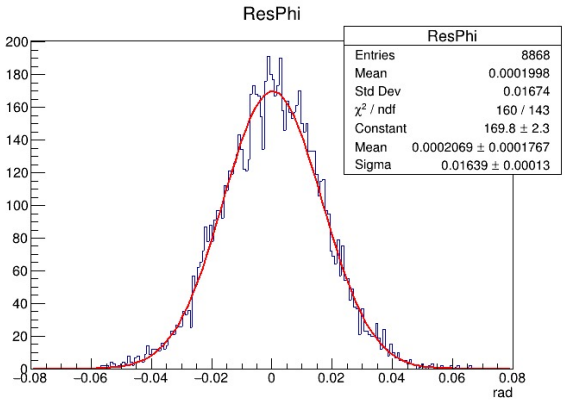
Approximately 800 MDC noise hits

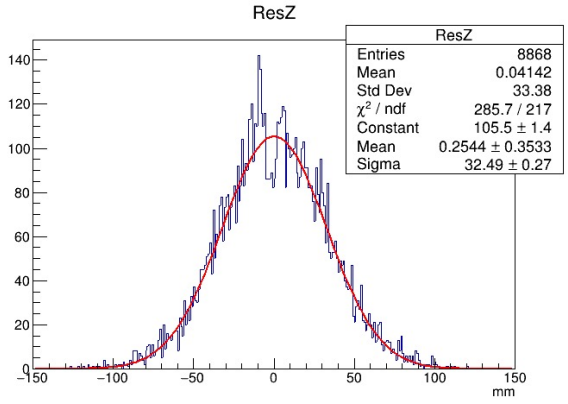
Tracking efficiency for single π+

Track Extrapolation

- A Geant4-based track extrapolation algorithm is imported from BESIII
  - Extrapolating the MDC fitting track into outer sub-detectors
  - Support 5 hypotheses: e, mu, pi, K , p
  - Has been used by other sub-detectors







Pt=1.0GeV mu-,Theta=90 extrapolation position vs truth position

Extrapolate to MUC

RICH Reconstruction

- A likelihood-based PID method is studied
  - The photon collected in each anode pad follows the Poisson distribution:

$$pdf_{i,h} = Poisson\left(N_i + 10^{-3}, avg_{i,h} + 10^{-3}\right),$$

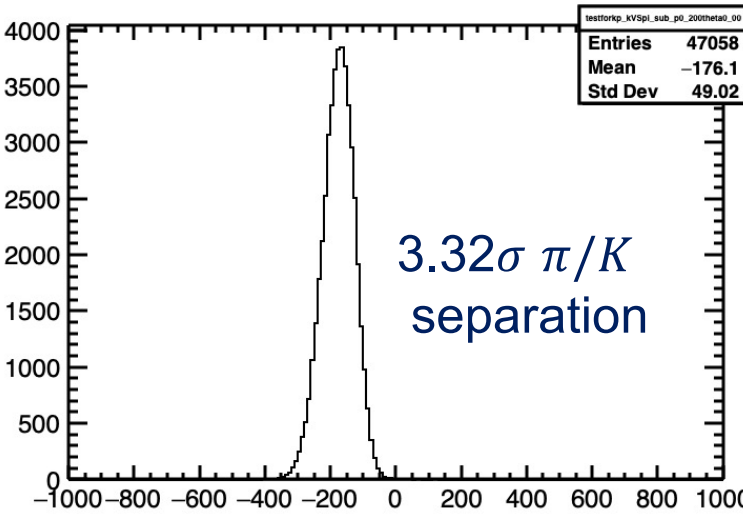
  - The Likelihood of h hypothesis:

$$\ln \mathcal{L}_h = \sum_i^{npads} \ln pdf_{i,h} \quad (\text{h: pi, K, P})$$

  - The Difference in log-likelihood (DLL) between two hypothesis

$$DLL = \sum_i^{npads} \ln \frac{pdf_{i,\pi}}{pdf_{i,K}}$$

More details in Qian's talk



- The implementation within OSCAR is on-going now

DTOF Reconstruction

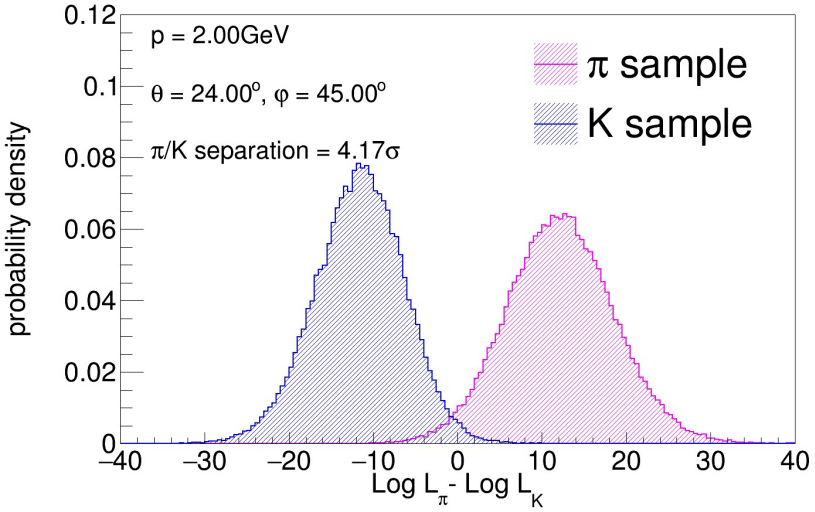
- A likelihood-based PID method is studied for DTOF
  - use time information of the photo electrons.

$$\mathcal{L}_h = p_h(N_{p.e.}) \prod_{i=0}^{N_{p.e.}} f_h(TOF_i) \qquad f_h(t) = \begin{cases} gaus + 0.05, & signal\ and\ bkg \\ 0.05, & bkg \end{cases}$$

$$\Delta\ell = Log\,\mathcal{L}_\pi - Log\mathcal{L}_K \qquad p_h(N_{p.e.}) = crystalball$$

More details in Ming's talk

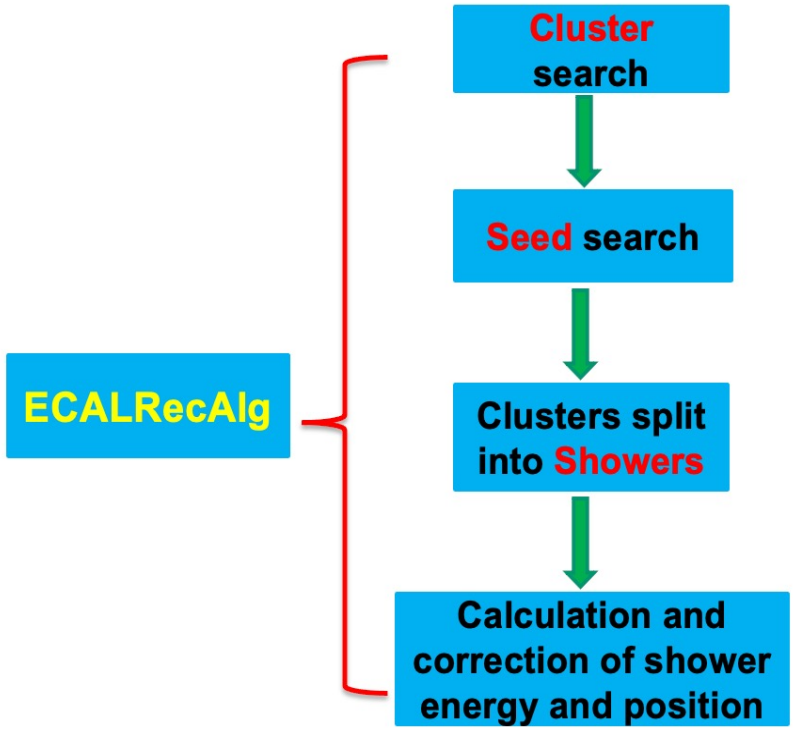
- separation power reaches 4.17σ
- The migration of the method to OSCAR is on-going now

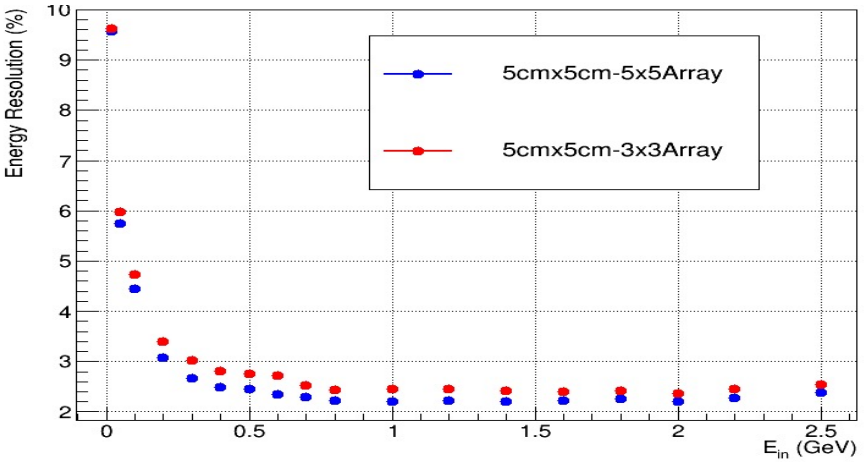


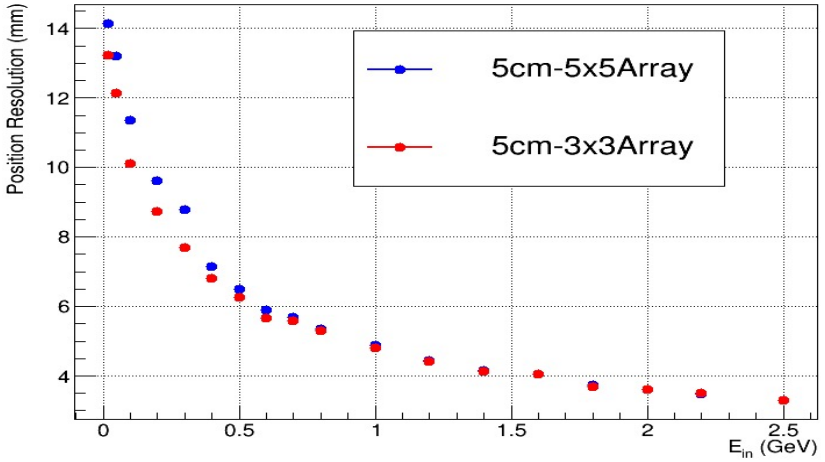
ECAL Reconstruction

- The algorithms have been developed and well-tested
  - the energy of the shower
  - the position of the photo

More details in Yunlong's talk

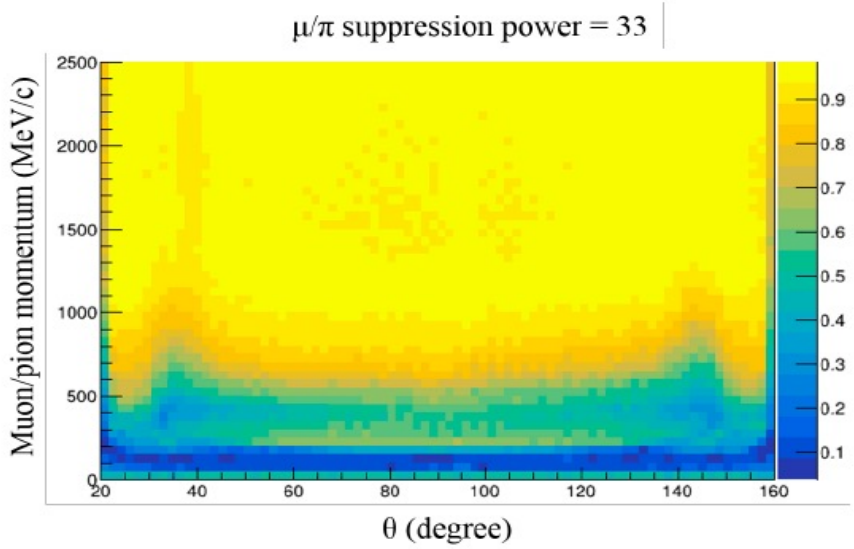


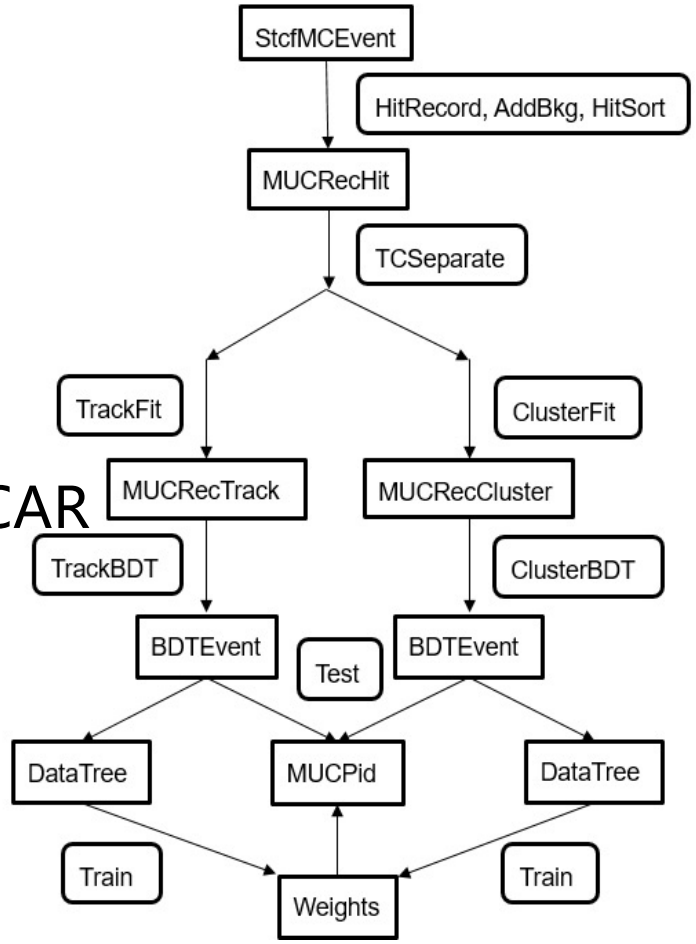




MUD Reconstruction

- The reconstruction algorithms are developed based on BDT method
  - Charged tracks
    - Use MDC，EMC，MUD reconstruction information
    - 14 BDT variables
  - Neutral particles
    - Use EMC ,MUD reconstruction information
    - 23 BDT variables
- All algorithms has been migrated into OSCAR

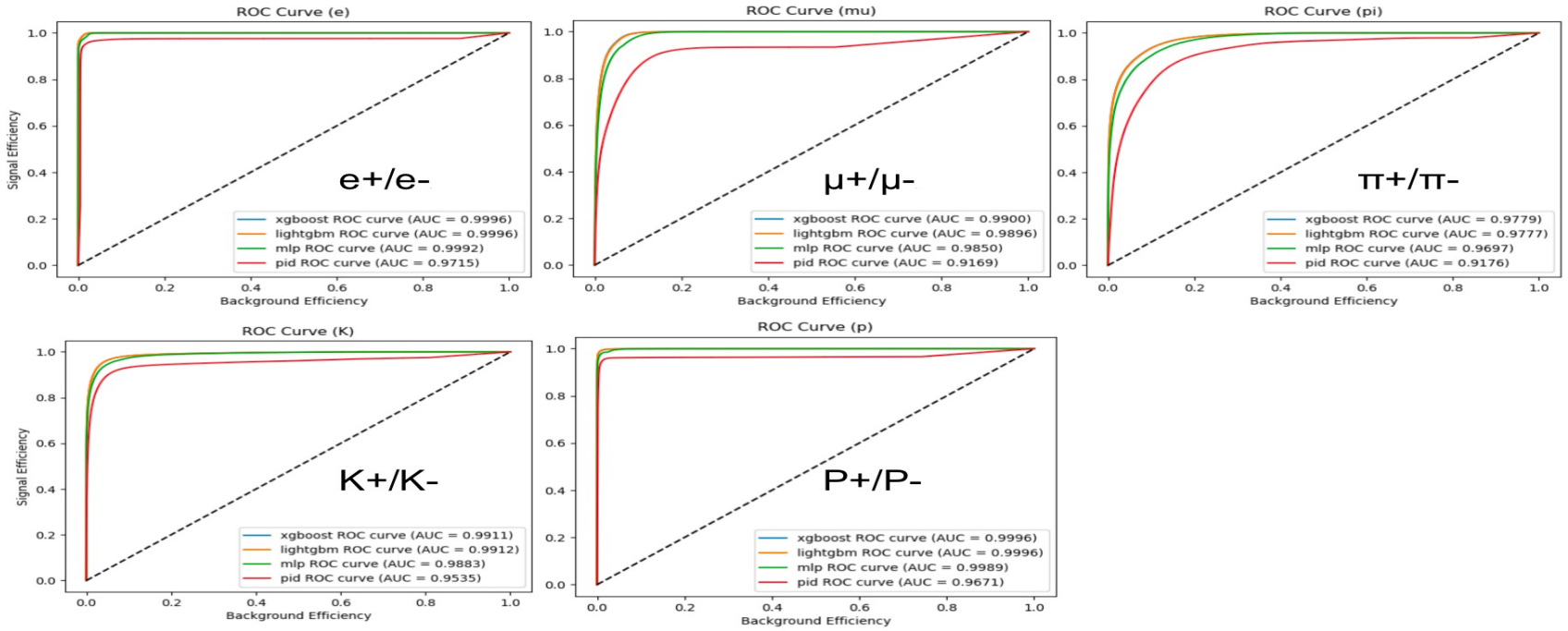




More details in Zhujun's talk

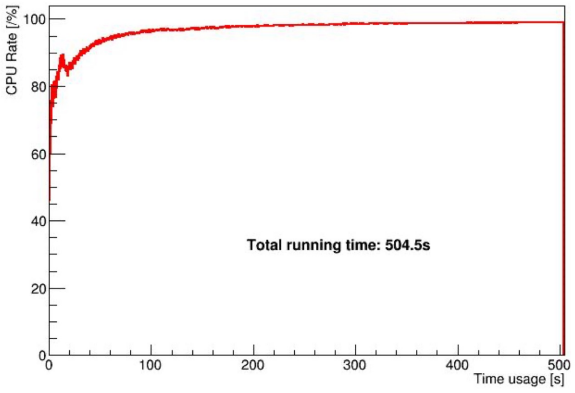
Global PID Algorithm Based on Deep Learning

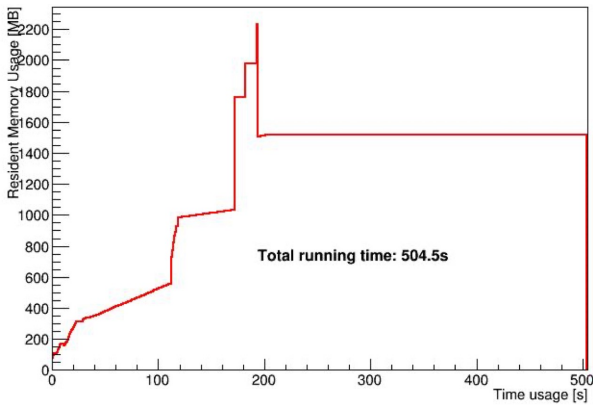
- Combine information of multiple sub-detectors
  - MDC, TOF, RICH, EMC, MUC
- Study the performance of different Models with BESIII Data
  - GBDT, MLP, SVM, CNN, GRU, LSTM
  - GBDT currently outperforms better than other models
- Plan to use OSCAR MC Data to study PID performance

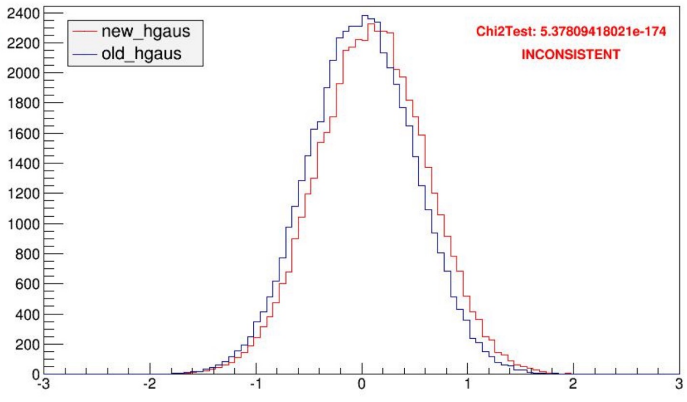


Automated Validation System

- An automated validation system is being developed for software validation at different levels
  - Unit test, integrated test, performance test, physical validation etc.
- A powerful toolkit is developed for building software validation workflow
  - Provide interfaces to define and run unit tests
  - Support various detectable failures (log errors, memory leaking, ...)
  - Support performance profiling
  - Support results validation based on statistical methods

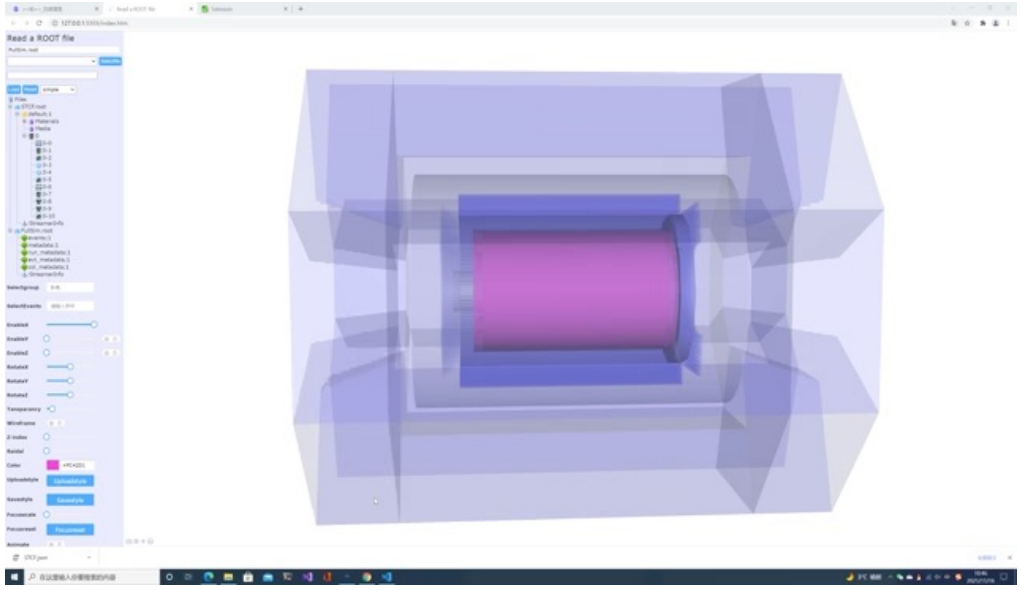


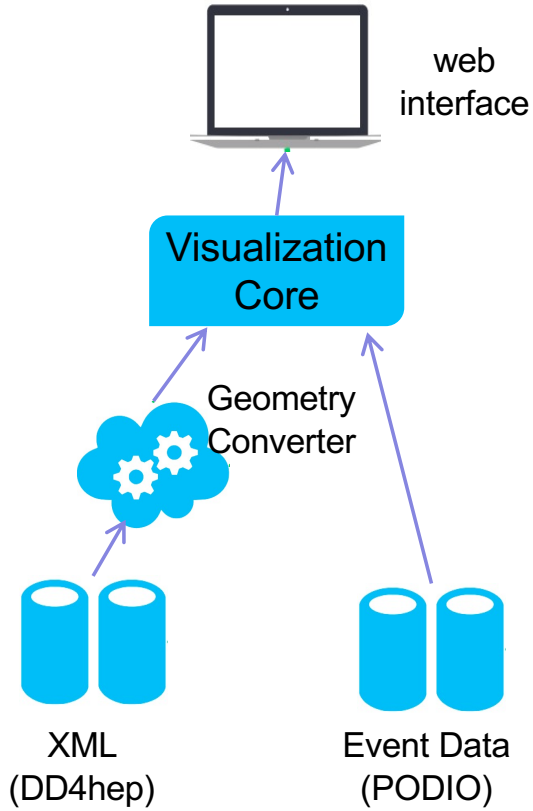




Visulization Toolkit

- A common detector visualization and event display toolkit is being developed based on:
  - WebGL: high-performance interactive 3D and 2D visualization
  - ThreeJS: 3D mesh renderer
  - DD4hep: detector description
  - PODIO: event data model
- Initially developed for HERD, optimizing for STCF now





Software Release

- OSCAR_1.0.0 (2020-06)
  - Phokhara
  - StcfEvtGen
  - DIAG36
  - MDI geometry
  - Magnetic Field
  - DataBuffer
  - MemMonitor
  - DetGeoConSvc
  - Qt5
  - Oscar command
- OSCAR_2.0.0 (2021-01)
  - RootIOSvc
  - TClonesArray
  - CommonSvc
  - Accessing EDM in muti-task
  - Shared ExternalLibs
  - Detector Cell Id
  - RICH Geometry update
  - RecAlg
  - RecEvent
- OSCAR_2.1.0 (2021-12)
  - EDM with PODIO
  - podio00-11
  - root 6.20.04
  - CMT->cmake
  - Validation package
  - DDXMLSvc

OSCAR_2.1.0 will be released soon after migration of EDM to PODIO and validation of reconstruction algorithms.

Summary and Future plan

- Lots of Progress has been made since last workshop
  - Redesign EDM with PODIO and developed new EDM system
  - Optimized the full detector simulation chain
  - Recon. algorithms for Tracker, ECAL and MUC are implemented with OSCAR
  - PID/Global PID algorithms are under developing
  - Developed prototype of the Validation and visualization tools
  - Software Management updates: SVN->Gitlab, CMT->Cmake
- Future plan for next year
  - Update geometry information according to the latest detector designs
  - Implement more realistic simulation of digitization process
  - Finish migration of reconstruction algorithms
  - Optimize reconstruction performance with backgrounds

Thanks for your attention!