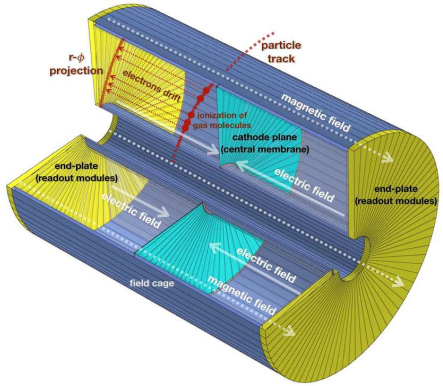
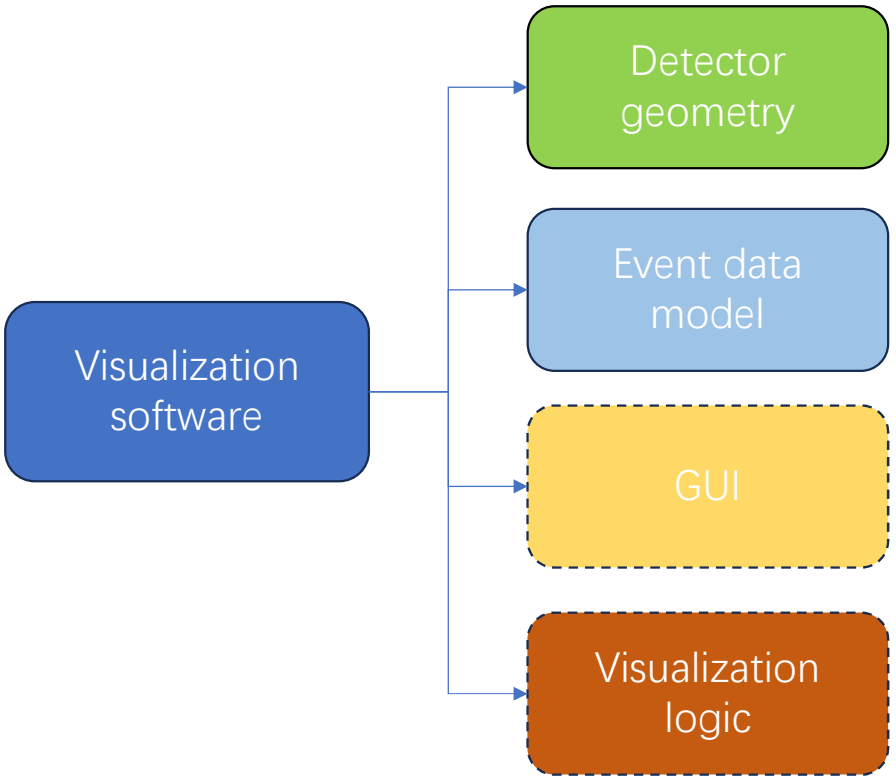
Event display and Interactive Visualization Tools for the CEPC

Yujie Zeng（曾宇杰）, Tianzi Song（宋天资）, Zhaoyang Yuan（袁朝阳）,
Yumei Zhang（张玉美）, Zhengyun You（尤郑昀）
Sun Yat-sen University（中山大学）

The 2025 International Workshop on the High Energy Circular Electron Positron Collider
2025.11.10

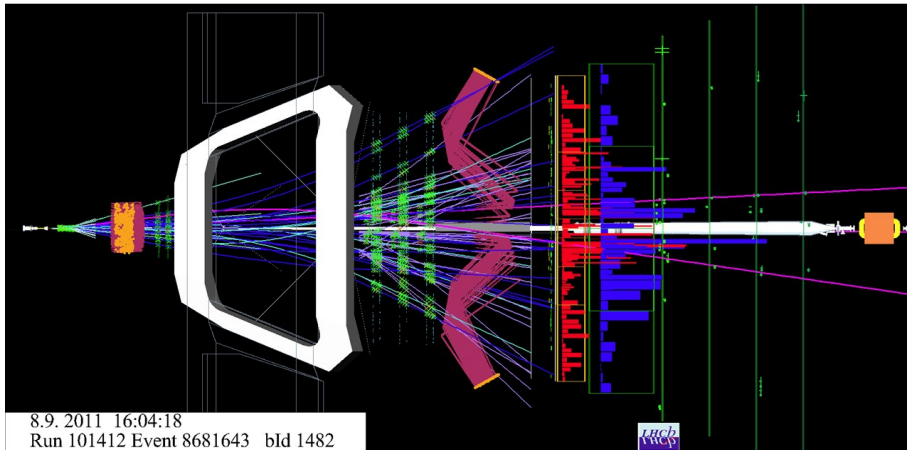


Contents

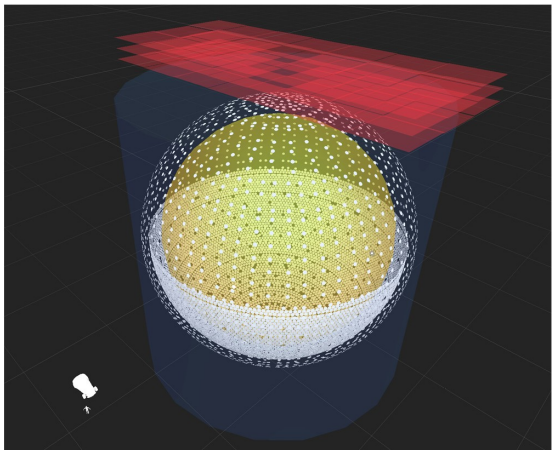


Event display software

- In modern High-Energy Physics (HEP) experiments, the structures of detectors and events are becoming increasingly complex.
- An event display software for HEP experiments is used to visualize detector structure and event information.
- Can greatly reduce the difficulties when analyzing detector geometries and event data.
- Many HEP experiments have already developed their own event display tools, to aid scientists learn about the structure of detectors, analyze physical events, optimize algorithms...



(a)

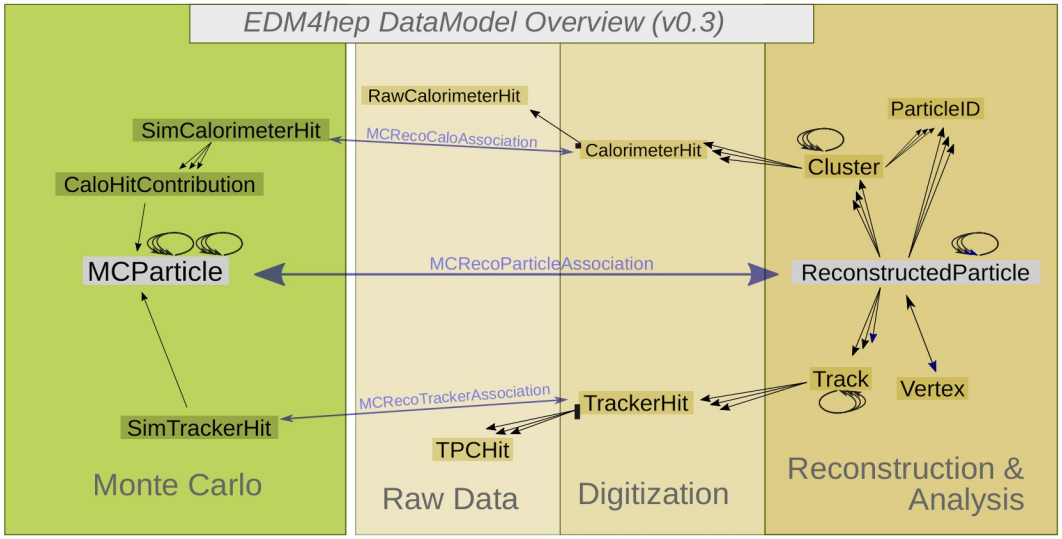


(b)

Fig.1 Some event display tools in HEP experiments. (a) is an event display tool based on Gaudi for LHCb, while (b) is an event display tool developed with Unity for JUNO

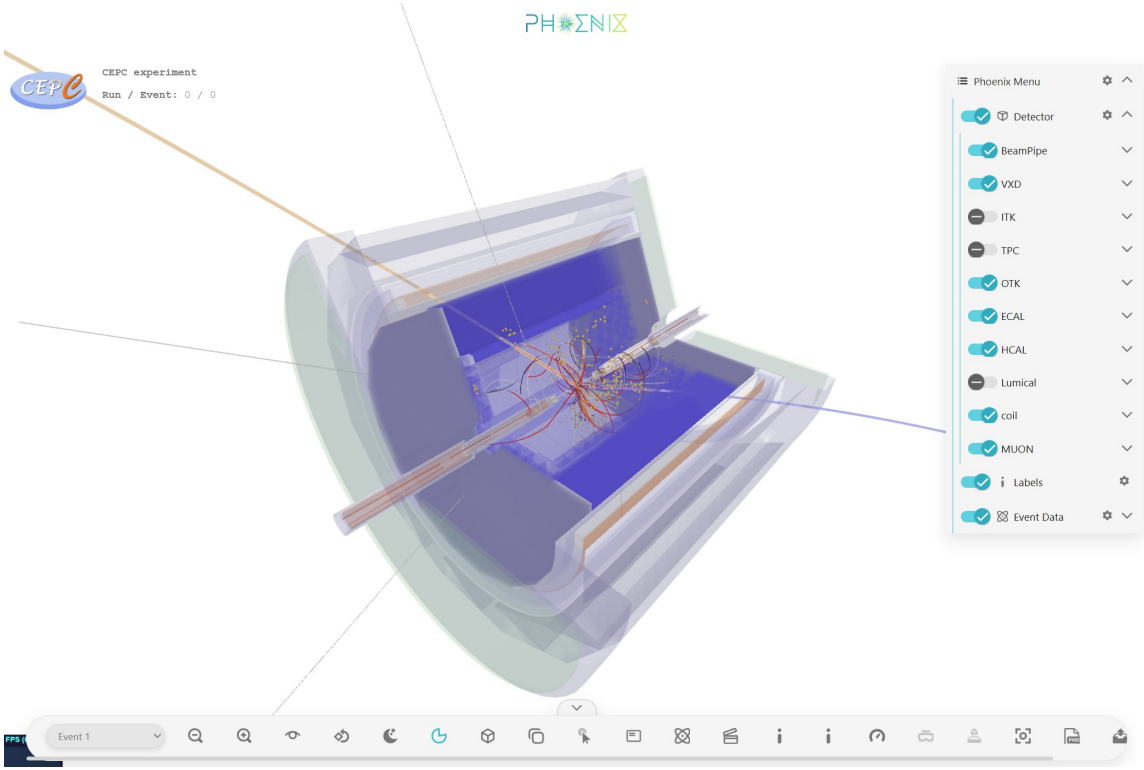
Event data model

- EDM4hep is the common event data model (EDM) that is used by communities in the Key4Hep project such as ILC, CLIC, FCC-ee & FCC-hh, CEPC, ...
- EDM4hep model efficiently keeps three stages of event information: Simulation, Calibration, and Reconstruction
- Since CEPC offline software uses EDM4hep to represent event information, our event display tools also accepts EDM4hep formatted event files for visualization.
- Before visualization a simple conversion from ROOT format to JSON format is taken, by using the tool **edm4hep2json**.



Contents

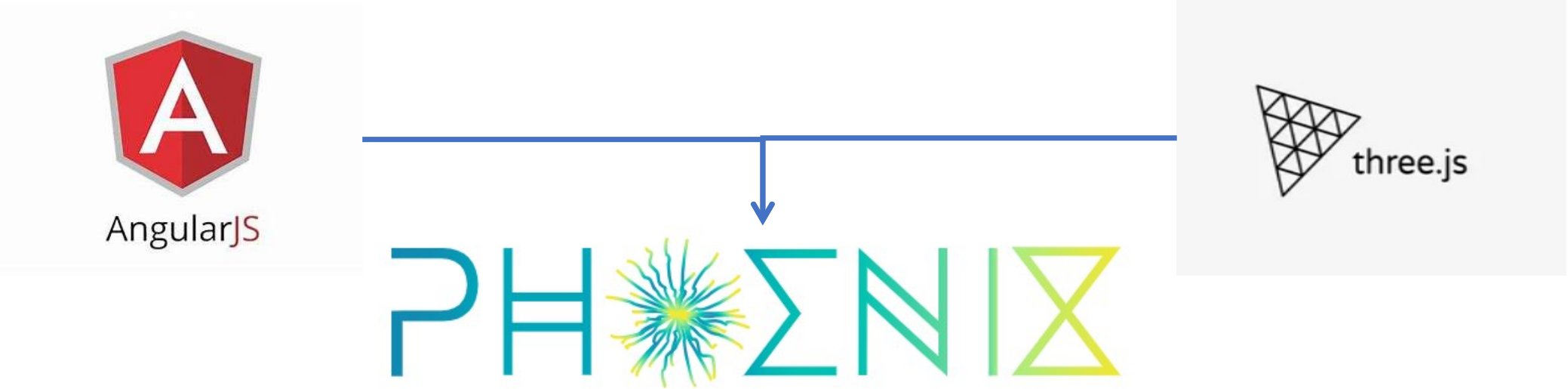
- Introduction
- Visualization in Phoenix
  - Detector visualization
  - Event display
- Visualization in Unity
  - Detector visualization
  - Event display
- Summary



The graphical user interface of CEPC event display in Phoenix

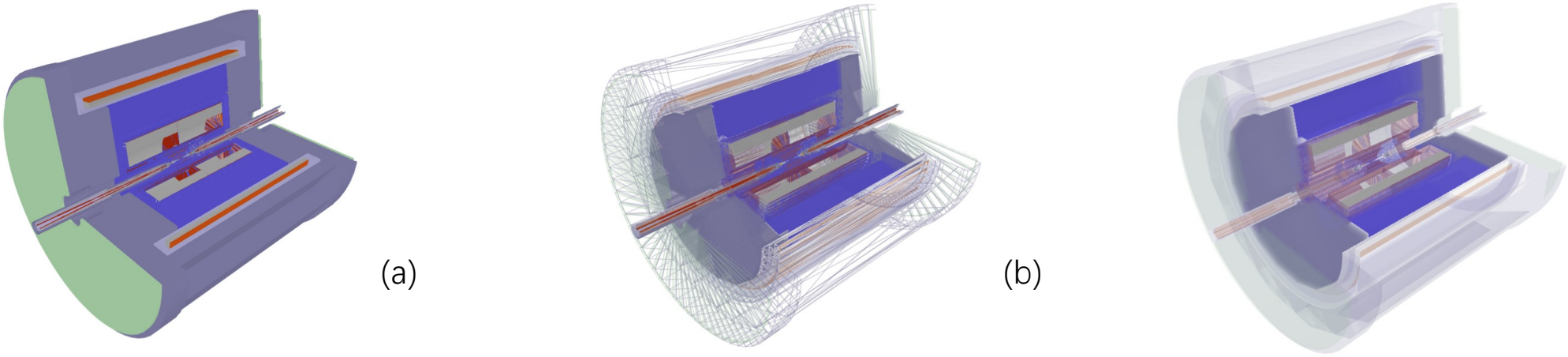
Phoenix, a web-based event display system

- Phoenix is an extensible, experiment-agnostic framework for event and geometry visualization.
- Can be used to visualize detector geometry in ROOT format, and uploaded events in JSON.
- Is a web-based framework. Can run on any modern browser that supports WebGL.
- Developed using Angular.js framework. Used Three.js and jsroot libraries to obtain 3D support and ROOT compatibility. Can also be easily deployed in any web server.



Detector visualization in Phoenix

- Phoenix loads ROOT formatted geometries to visualize the detector.
- Supports a variety of high level features:
  - Showing the geometry mesh
  - Adjusting the transparency of geometry
  - Show different components of geometry
  - Clipping the geometry in different angles

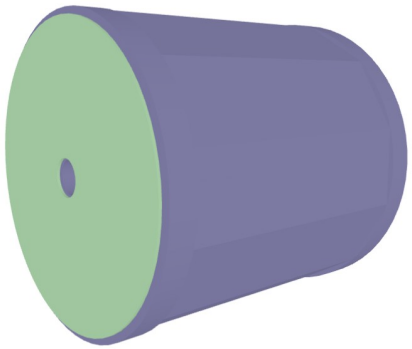


(a) (b)

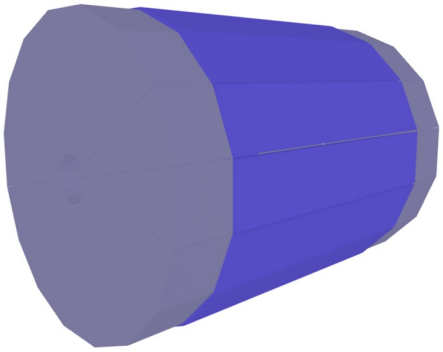
Fig.2 The CEPC detector in Phoenix. (b) and (c) shows its visualization in wireframe and transparent mode

Detector visualization in Phoenix

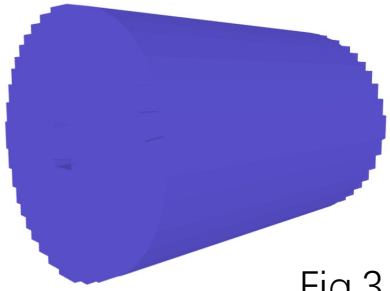
- Phoenix supports showing different components of CEPC detector alone, enabling the user to see its internal structure.



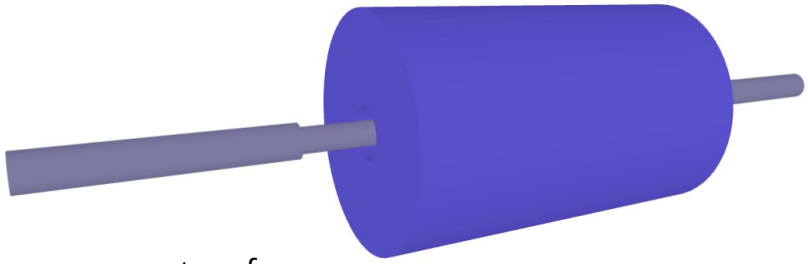
(a) Muon detector



(b) HCAL



(c) ECAL

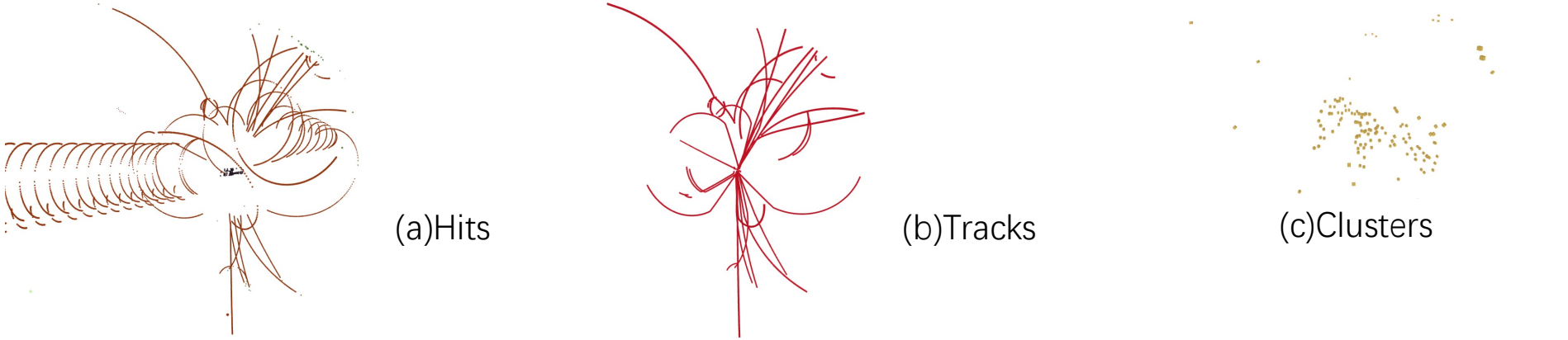


(d) Tracker

Fig.3 The different components of CEPC detector in Phoenix

Event display in Phoenix

- To visualize CEPC events, convert the event file to JSON format firstly.
  - Setup EDM4hep toolkit, and then use command $edm4hep2json events.root
- An event is composed of several components in EDM4hep data model, including tracks, clusters, calorimeter cells, hits and jets.
  - Tracks: the trajectories of charged particles
  - Calorimeter cells: Energy deposits within a calorimeter
  - Clusters: Groupings of energy deposits in a calorimeter
  - Hits: Individual detector measurements



(a)Hits (b)Tracks (c)Clusters

Fig.4 The visualization of Hits, Tracks, Clusters of an event in Phoenix

Event display in Phoenix

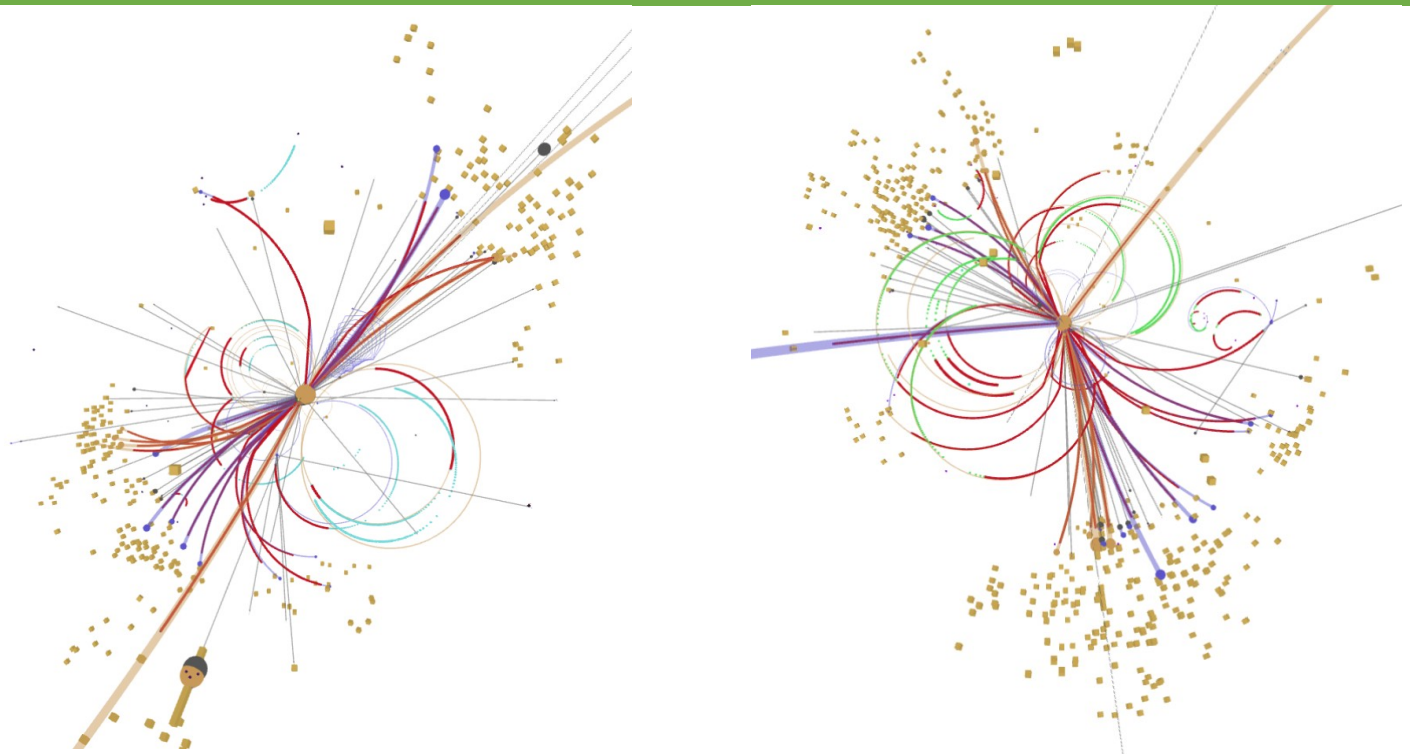
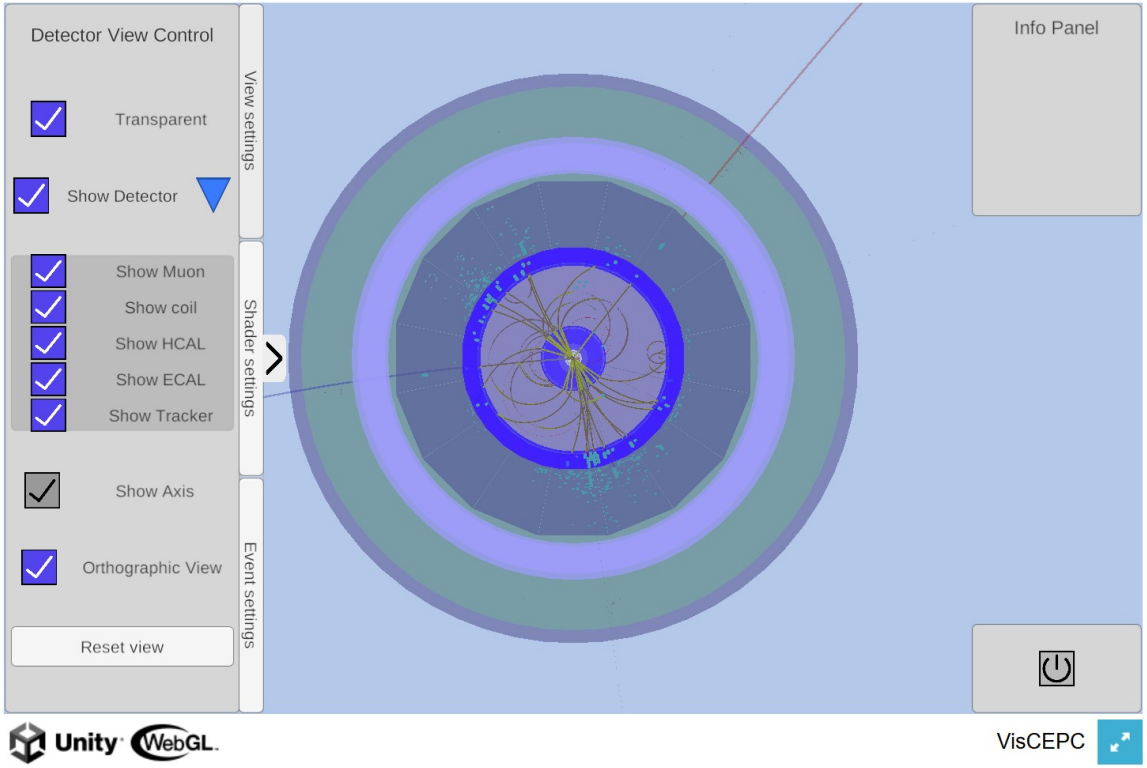


Fig.5 The visualization of two $e^+e^- \to ZH, Z \to \mu^+\mu^-, H \to b\bar{b}$ events in Phoenix

Contents

- Introduction
- Visualization in Phoenix
  - Detector visualization
  - Event display
- Visualization in Unity
  - Detector visualization
  - Event display
- Summary



The graphical user interface of CEPC event display in Unity. Build on WebGL platform

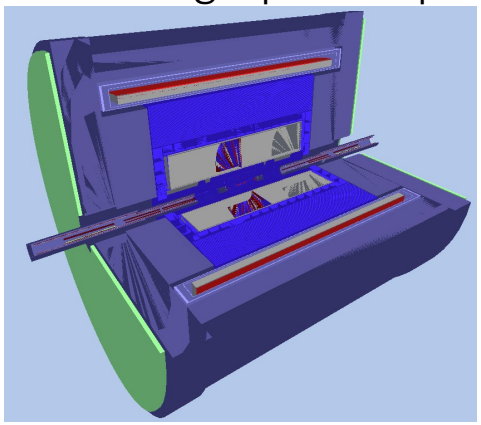
Unity engine

- **Unity** is the world's most popular development platform for creating multiplatform games and interactive experiences.
- Unity has the following advantages:
  - **Easy to use:** Unity engine has a intuitive interface for beginners to get started quickly. Additionally, it provides comprehensive documentation for users.
  - **High-Quality Graphics:** Unity supports advanced graphics technologies like physically-based rendering.
  - **Cross Platform:** With Unity, we can develop once and deploy the application across multiple platforms (Windows, Android, Linux, MacOS, ···)
- Is a more sophisticated choice for building event display tools from zero.

Fig.6 The platforms that Unity engine supports. The red ticks marks the platforms that our event display tool fully supports

Detector visualization in Unity

- The detector geometries are converted to FBX format, and then be loaded as assets in Unity.
- Currently supported features:
  - Showing the detector transparently
  - Show different components of geometry
  - Clipping the geometry horizonally or vertically
  - Switch to orthographic or perspective view



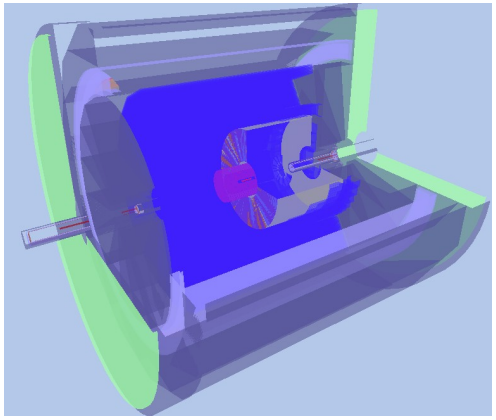


Fig.7 The visualization of CEPC detector in Unity

Event display in Unity

- The CEPC event display in Unity also requires the user to convert the event file to JSON format
- Also fully supports visualizing components in EDM4hep data model, including tracks, clusters, calorimeter cells, hits and Jets

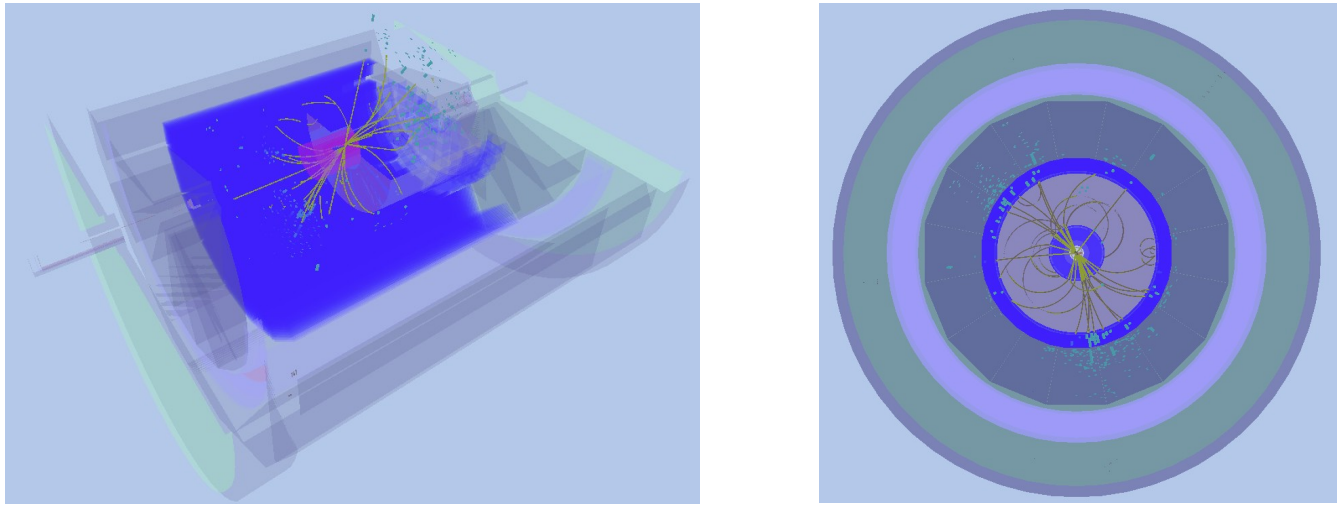


Fig.8 The visualization of an $e^+e^- \to ZH, Z \to \mu^+\mu^-, H \to b\bar{b}$ event in Unity, together with the CEPC detector

Event display in VR devices with Unity

- With the strong support of Unity XR plugin, the CEPC event display can even work in VR devices, providing an immersive experience.
- CEPC event display in VR platforms is still under development.

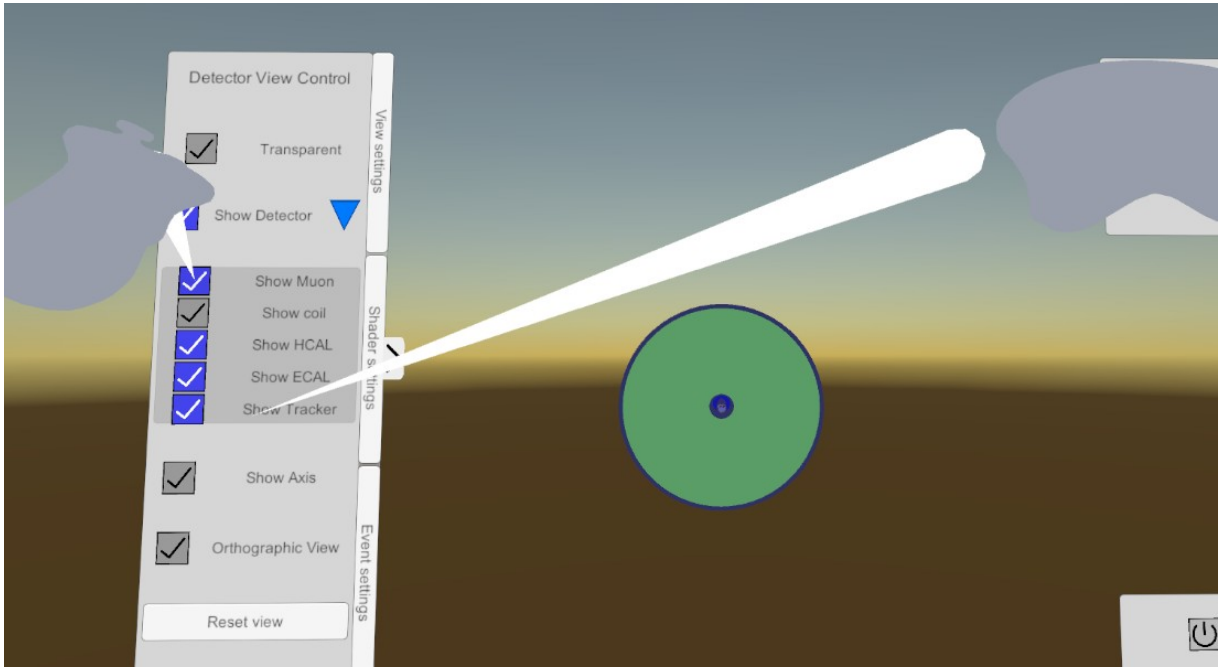
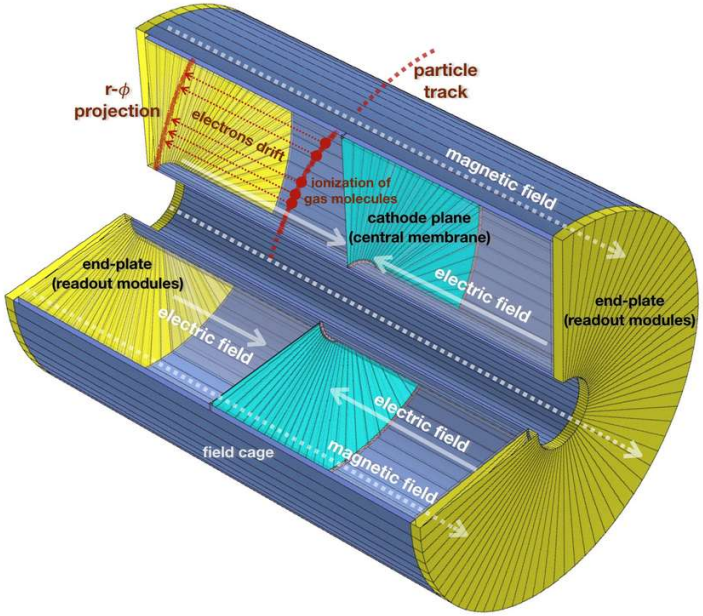


Fig.9 A screenshoot of CEPC event display in oculus quest2.

Contents

- Introduction
- Visualization in Phoenix
  - Detector visualization
  - Event display
- Visualization in Unity
  - Detector visualization
  - Event display
- Summary



Summary

- We have developed CEPC event display softwares, with basic functionalities, based on two frameworks
  - Phoenix: Web-based framework, already has some useful features at the beginning, suitable for developing event display software for HEP experiments
  - Unity: Popular development platform for creating 3D softwares. Provides a variety of high-level features for 3D rendering. good choice for building more sophisticated 3D applications
- Future prospects:
  - Continuous development focusing on expanding features such as VR support.
  - Potential enhancements include ROOT formatted event file support, real-time data streaming, event filtering and more.

Thank you for your attention!