Nuclear Science and Techniques (2026) 37:113
https://doi.org/10.1007/s41365-026-01932-3

An event visualization software based on Phoenix for the CEPC experiment

Yu-Jie Zeng[1] · Tian-Zi Song[1] · Xue-Sen Wang[2] · Yu-Mei Zhang[2] · Tao Lin[3] · Zheng-Yun You[1]

Received: 30 May 2025 / Revised: 11 August 2025 / Accepted: 13 August 2025 / Published online: 10 April 2026

Abstract
In high-energy physics (HEP) experiments, visualization software plays a pivotal role in detector design, offline software development, and event data analysis. The visualization tools integrate detailed detector geometries with complex event data models, providing researchers with invaluable insights into experimental results. Phoenix is an emerging general-purpose visualization platform for current and next-generation HEP experiments. In this study, we developed an event display software based on Phoenix for the CEPC experiment. It offers the necessary functionalities for visualizing detector geometries and displaying event data, allowing researchers to optimize detector design, test simulation and reconstruction algorithms, and analyze event data in a visualized manner. Additionally, we discuss the future applications of the event display software, including its use in online monitoring and the potential to build virtual reality projects for enhanced data visualization.

Keywords Visualization · Event display · Phoenix · CEPC

1 Introduction

The circular electron positron collider (CEPC) is an international scientific facility proposed by the Chinese particle physics community in 2012 [1]. Designed to explore fundamental physics, particularly as a Higgs factory, the CEPC will be situated in China within a circular underground tunnel approximately 100 km in circumference. This double-ring collider will circulate electron and positron beams in opposite directions through separate beam pipes, with detectors installed at two interaction points.

The CEPC is designed to operate in three distinct modes: H mode ($e^+e^- \to ZH$), Z mode ($e^+e^- \to Z$), and W mode ($e^+e^- \to W^+W^-$). These modes correspond to center-of-mass energies of 240 GeV, 91 GeV, and 160 GeV, respectively [2, 3]. It will be an advanced factory of Higgs, Z, W bosons, and top quarks, enabling precision measurements of the Higgs boson properties [4], along with those of the mediators of the weak interaction, the W and Z bosons [5, 6]. It also provides an opportunity to search for physics beyond the standard model (BSM), allowing scientists to discover new physics and unknown phenomena [7].

As the CEPC project progresses [8], a series of software needs to be developed, and event display software is an indispensable part [9]. In modern high-energy physics (HEP) experiments, the structures of detectors are becoming increasingly complex to achieve higher measurement accuracy and detection efficiency. This necessitates sophisticated visualization tools to aid scientists in learning about the structure of detectors, analyzing physical events, and optimizing simulation and reconstruction algorithms [10–12].

This work was supported by the National Natural Science Foundation of China (Nos. 12175321, W2443004, 11975021, 11675275, and U1932101), National Key Research and Development Program of China (Nos. 2023YFA1606000 and 2020YFA0406400), National College Students Science and Technology Innovation Project, and Undergraduate Base Scientific Research Project of Sun Yat-sen University.

✉ Yu-Mei Zhang
zhangym26@mail.sysu.edu.cn

✉ Tao Lin
lintao@ihep.ac.cn

✉ Zheng-Yun You
youzhy5@mail.sysu.edu.cn

[1] School of Physics, Sun Yat-sen University, Guangzhou 510275, China

[2] Sino-French Institute of Nuclear Engineering and Technology, Zhuhai 519082, China

[3] Institute of High Energy Physics, Beijing 100049, China

The ROOT software [13] and its EVE package [14] have been widely used to develop event visualization tools for HEP experiments. However, as a native C++ application, it lacks advanced web-based features, such as cross-platform compatibility and ease of deployment. The JSROOT [15] package, a web-based extension of ROOT, addresses many of these limitations by enabling browser-side visualization through WebGL and JSON data formats. However, there still lacks a unified framework that integrates modern 3D rendering capabilities with user-friendly interfaces. Phoenix is a web-based event display framework specifically designed for HEP experiments. It is lightweight and highly cross-platform, built on top of JSROOT and three.js libraries [16]. With the aid of these two components, Phoenix can provide a detailed 3D display of the detector geometry and events along with a user-friendly interface, making it a good choice for developing visualization tools for the CEPC. Therefore, in this study, we developed an event display software for the CEPC using the Phoenix framework. The software not only provides basic visualization functions for the detector and events, but also has a variety of extra features, allowing users to change the visualization effects through a graphical user interface (GUI) and check tracks, hits, and Monte Carlo (MC) information. Hence, the Phoenix-based event display software will be an effective tool for detector design, offline software development, and physics analysis in the CEPC experiment.

The remainder of this paper is structured as follows. In Sect. 2, we introduce the Phoenix framework for visualization in HEP experiments. In Sect. 3, the structure and data flow of the event display for the CEPC based on Phoenix are described. In Sect. 4, the visualization features of the CEPC event display are described. The applications are discussed in Sect. 5. Finally, Sect. 6 provides a summary.

2 Phoenix framework

Phoenix is a web-based event visualization framework for HEP experiments [17]. It is based on the TypeScript language and Angular.js framework [18] and uses the popular three.js [16] library for 3D rendering. In 2017, the HEP Software Foundation (HSF) visualization white paper called for a common event format and common tools to aid visualization [19], and the Phoenix framework was adopted as an extensive framework for event and geometry display. The Phoenix framework currently supports built-in geometries and event data for ATLAS [20], CMS [21], LHCb [22], and TrackML [23]. Several demonstrations have also been created for the FASER [24], FCC-ee, and FCC-hh [25].

Compared with native applications built on the ROOT software and its EVE package, the Phoenix framework has the following advantages.

- *Highly cross-platform* Unlike other event display software built as native C++ applications, Phoenix is a web-based framework and can run on any modern web browser. This framework was built as a web application and can be accessed by simply visiting a URL.
- *High-level support for HEP experiments* As an event visualization framework for HEP experiments, Phoenix provides high-level support for visualizing events. It supports the visualization of various types of physics objects, including tracks, hits, calorimeter cells, vertices, compound objects, and missing energy. Phoenix has also integrated the JSROOT [15] package, which is part of the ROOT framework, to provide support for visualizing 3D ROOT geometry. In addition to the ROOT format, 3D geometries such as OBJ [26], glTF [27], and JSON are also supported.
- *Extensive* Designed as experiment agnostic, Phoenix aims to support as many experiments as possible. It provides a standard interface to create a visualization application; thus, adding a new experiment demonstration only requires the developer to create a simple extension.
- *User interface and features* Phoenix offers rich user interface options, including labeling functionality for physics objects, shareable URLs for predefined events and configurations, useful outreach activities or physics briefings, and built-in animations that make event displays interactive and engaging. All these configurations can be saved, reloaded, applied by default, or passed through the URL parameters.

Therefore, the Phoenix framework is a competitive choice for developing event visualization tools. In fact, Phoenix has been the official web event display for ATLAS and is also used by FCC, LHCb, and Belle II experiments [28].

3 Methodologies

Event display software for HEP experiments should provide a detailed visualization of the detector structure and event data, featuring important information such as detector hits, showers, reconstructed tracks, event information, and a GUI to control the display. To achieve this, the CEPC event display software adopts the Phoenix framework and uses the CEPC offline software [29] to provide the detector geometry and event data.

The structure of the Phoenix-based CEPC event display software is shown in Fig. 1. The workflow of the CEPC event display software can be divided into the following steps. First, the CEPC offline software provides detector geometries in DD4hep [30, 31] compact XML format. It also generates event data through simulation or reconstruction and stores them in the EDM4hep [32, 33] format. DD4hep and EDM4hep are

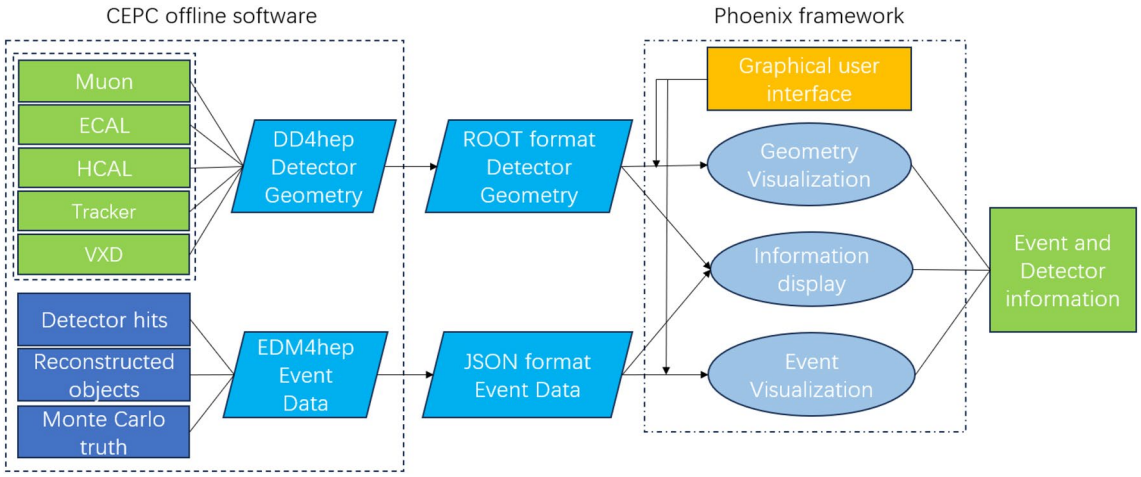


Fig. 1 The structure and data flows of CEPC event display software based on Phoenix. The left side shows the CEPC offline software, which provides the detector geometry in the DD4hep compact XML format and event data in the EDM4hep format. The geometry and event data are converted to ROOT format and JSON format, respectively, serving as the input of the Phoenix framework on the right. With a graphical user interface, Phoenix provides a detailed 3D visualization and information display

both parts of Key4hep, a framework that provides all the necessary ingredients for future HEP experiments [34, 35]. Second, because the Phoenix framework does not directly support DD4hep XML format geometries and EDM4hep format event data, a series of conversions are required [36, 37]. The geometries were converted to ROOT format using the ROOT software, and the event data were converted to JSON format while preserving the essential information for visualization. Finally, the geometry files and event data are loaded into the web display tool, rendering the detector and events, while essential information for analysis, such as particle energy or tracker hits, is also displayed in the user interface (UI).

A simple but practical GUI is available for controlling the workflow. Figure 2 shows the user interface of the event display software. The right corner of the figure shows the Phoenix menu, which can be used to control the visibility of each detector object; the bottom of the figure shows the Phoenix icon bar, which can be used to upload or export files, change display modes, and enable or disable other high-level features. Users can also use the GUI control to upload geometry and event data files for visualization, change the display effects, and choose which part of the event or detector to visualize. The CEPC and Phoenix logos are displayed on the left and the top, and users can visit the CEPC and Phoenix homepage by clicking the logos.

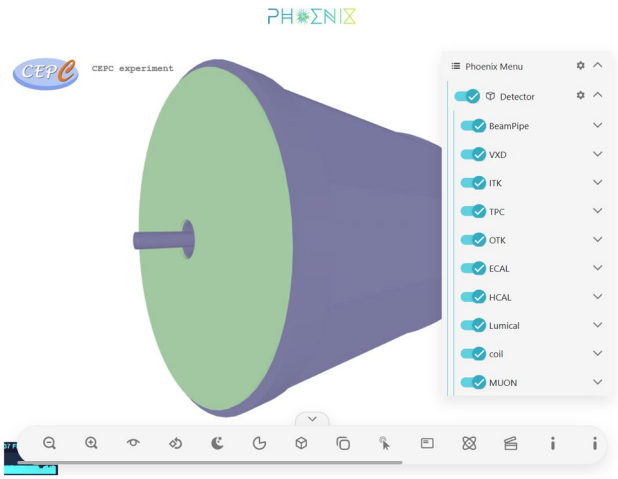


Fig. 2 (Color online) The GUI of CEPC event display software based on Phoenix. The detector is shown at the center, with the UI components in the surrounding area. The panel on the right is the Phoenix menu, and the tool bar on the bottom is the Phoenix icon bar

4 Visualization

4.1 Visualization of detector

To enable the visualization of the CEPC detector within the Phoenix framework, a series of conversion steps were undertaken [38]. Initially, the geometries of the detector were stored in the DD4hep XML description within the CEPC offline software. By running Geant4 [39] simulations, the detector geometries were loaded to perform simulations, and with the G4GDMLParser module provided by the Geant4 toolkit, the geometries were exported from the simulation in Geometry Description Markup Language (GDML) format [40, 41]. This step ensured that all geometric details of the detector components were accurately captured and preserved.

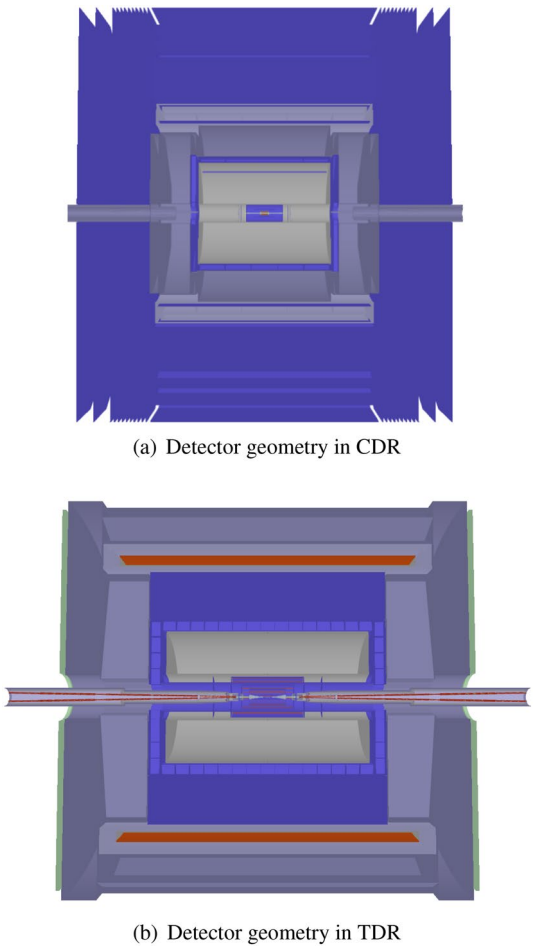
Subsequently, these GDML geometries were converted into the ROOT format using the geometry export

functionalities provided by the ROOT framework. This conversion is necessary because Phoenix does not natively support GDML as a detector-input format. The conversion process from the DD4hep XML format to the ROOT format retains the full geometric complexity of the detector, with no explicit simplification applied to the original design. Although the ROOT format internally represents geometry using Constructive Solid Geometry (CSG), the JSROOT package automatically converts this CSG-based geometry into surface-based representations for rendering. To manage performance during 3D visualization, JSROOT also allows the setting of upper limits on the number of visible nodes and surfaces, effectively mitigating potential rendering bottlenecks. Consequently, even highly complex detector models can be visualized interactively without significant performance degradation.

In the context of the CEPC detector visualization within the event display software, the detector is composed of several key components, including but not limited to the Vertex Detector (VXD) [42], the Tracker [43, 44], the Electromagnetic Calorimeter (ECAL) [45], the Hadronic Calorimeter (HCAL) [46], and the Muon Detector (Muon). The geometry of each component was individually generated and loaded separately into Phoenix, allowing users to selectively visualize specific parts of the detector using the Phoenix menu.

Once the geometries are loaded into the Phoenix framework, they are immediately visualized in the software, and the users can freely change their perspective of view, zoom in or zoom out the 3D model, and clip the geometries to observe the details inside the detector. As the CEPC detector is still under design, multiple versions of the geometry are available in the software for comparative analysis. Figure 3 illustrates the visualization of several versions of the CEPC detector within Phoenix. Figure 3a shows the conceptual design report (CDR) version of detector geometry. Figure 3b displays the latest technical design report (TDR) version of detector geometry under development [47]. In addition, Fig. 3 presents detailed section views of CEPC detector geometries, sequentially showing the VXD, Tracker, ECAL, HCAL, Electromagnetic Coil [48], and Muon detector from the innermost to the outermost layers [3].

Additionally, the Phoenix framework offers users the flexibility to select specific sections of the detector for visualization. For instance, Fig. 4 illustrates the display of CEPC sub-detectors. Owing to performance constraints, many detailed aspects of the detector geometries may be omitted in the default view. However, users can manually change the opacity of the 3D model or activate the "Wireframe" mode through the user interface, as shown in Fig. 5. This mode reveals the mesh structures of the detector geometries, providing a detailed view of the geometric configurations while rendering the 3D models transparent. This feature is particularly useful for gaining deeper insights into the internal structure and layout of complex detector components.



(a) Detector geometry in CDR

(b) Detector geometry in TDR

Fig. 3 (Color online) The section views of different versions of CEPC detector geometries. **a** shows the Conceptual Design Report (CDR) version of detector geometry, while **b** presents the latest Technical Design Report (TDR) version of detector geometry

4.2 Visualization of events

In HEP experiments, it is standard practice to store event data in the ROOT format because of its efficiency and flexibility. However, because Phoenix does not natively support the ROOT event format for visualization, an additional step involving the conversion of event data formats is required before visualization can occur. The Phoenix framework provides support for loading event data from JSON format files. To convert the ROOT event format to JSON format, the tool EDM4hep2json, which is provided by the EDM4hep [49] toolkit, was utilized. This tool can convert EDM4hep event data to JSON format while preserving the essential event information required for visualization.

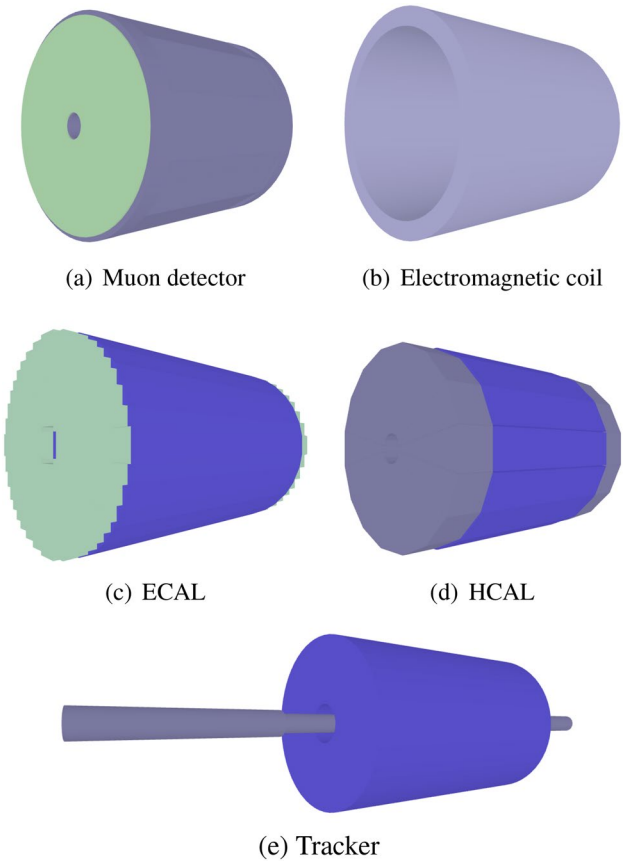


Fig. 4 (Color online) Visualization of various sub-detectors in CEPC event display software

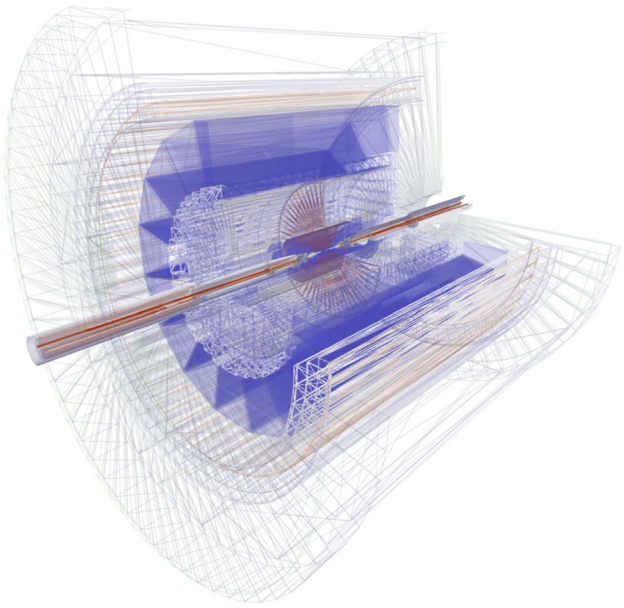


Fig. 5 (Color online) Visualization of CEPC detector in the wire frame mode

Specifically, the EDM4hep2json toolkit effectively "dumps" the EDM4hep event data into JSON format, ensuring that all relevant event information is retained and usable within Phoenix. This conversion process bridges the gap between the event data storage format and the visualization requirements of the Phoenix framework, enabling comprehensive and interactive visualizations of the events in the CEPC detector.

Once converted to the JSON format, the user can manually upload the event file to the event display software, and the first event in the event data file is immediately visualized. The user can also choose which event to visualize through the user interface. Figure 6 provides an illustrative example of the CEPC detector along with a single event, as visualized by the Phoenix event display software. The figure highlights three primary physical event objects:

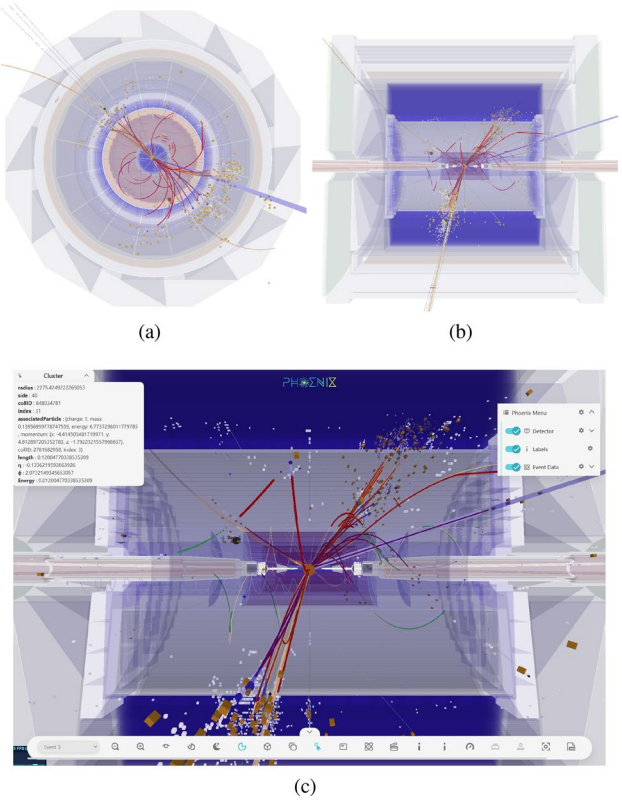


Fig. 6 (Color online) The display of an $e^+e^- \to ZH, Z \to \mu^+\mu^-, H \to b\bar{b}$ event in CEPC event display. The detector is shown here to illustrate the interaction between particles and detectors. **a** and **b** plot the detector and event in X-Y and Z-R section view, respectively. **c** enlarges the details of **b** to show the event objects, where the red lines are tracks, the small and white cubes are calorimeter cells, and the yellow pointing cuboids are clusters

- *Tracks* Representing the trajectories of charged particles, depicted as red lines.
- *Calorimeter cells* Indicating energy deposits within a calorimeter, shown as small and transparent cubes.
- *Clusters* Groupings of energy deposits in a calorimeter, represented by yellow pointing cuboids.

In addition, the software supports the visualization of several other physics objects, including:

- *Jets* Simplified geometric cones indicating the direction of energy flow.
- *Hits* Individual detector measurements, illustrated as points in space.
- *Vertices* Origins of tracks, displayed as dots.

The software further enhances the user experience by enabling the selective visualization of event objects through an intuitive interface. Users can disable certain object types, such as calorimeter cells, to focus on higher-level features like clusters. For instance, if the clusters within the ECAL detector are stored as a collection in the EDM4hep event data file, users have the option to enable or disable their visualization, allowing for a focused analysis of specific aspects of the event.

Moreover, the software facilitates a detailed analysis by providing information about the selected objects. When the "Object selecting" mode is activated, users can select objects to view detailed information in the info panel located in the top left corner. For example, selecting a cluster reveals its position, energy, and associated reconstructed particle properties (including charge, energy, mass, and momentum), as shown in the left-top corner of Fig. 6c. This functionality is invaluable for a thorough event analysis.

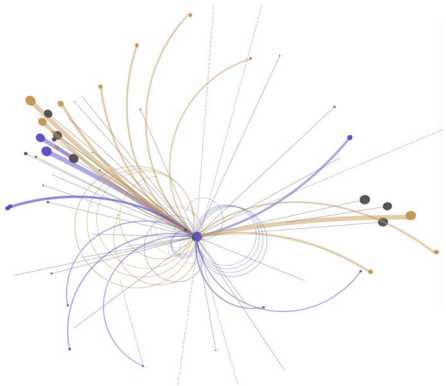
4.3 Visualization of simulation information

Monte Carlo information, which is generated from simulated particle interactions, provides ground-truth data for validating reconstructed objects (e.g., matching tracks to MC particles). In modern HEP experiments, MC information also plays an important role in event analysis, detector design, and algorithm optimization [11, 50]. It can help physicists evaluate the performance of the detector, test the reconstruction and analysis processes, and understand the signal event and source of the backgrounds. Because the CEPC experiment is still under planning, MC simulation is the primary source of event data, and utilizing MC information is the key to optimizing the detector design and reconstruction algorithm [12].

Based on the Phoenix framework, we extended the original set of visualization primitives to display particles. Our event display software now supports a detailed visualization of the MC information. The EDM4hep event data model stores MC information as an individual collection, and the software displays it along with other event objects. For collider experiments such as the CEPC, the MC information includes detailed properties of MC simulated particles, enabling physicists to validate reconstruction processes by comparing reconstructed tracks with the MC truth information of particles. The event display software extracts information about the MC particles, such as their vertex, momentum, charge, and endpoint, to calculate and draw their trajectories within the detector. As an example, Fig. 7 displays an event with its reconstructed information and particle MC truth information. The similarity between the two subfigures distinctively validates the accuracy of the reconstruction based on visual comparison.

In the event display software, the MC truth information is visualized as a set of tracks. To obtain a better visualization effect, the software divides the MC particles into three categories.

(a) Reconstructed information



(b) MC truth information of particles

Fig. 7 (Color online) Display of the reconstructed information and MC truth of a simulated event $e^+e^- \to ZH, Z \to \nu\bar{\nu}, H \to gg$, **a** plots the reconstructed tracks, calorimeter cells and hits in the event, while **b** displays the MC truth information of all particles in this event

- *Charged particles* The charged particles will interact with the magnetic field and be detected by the Tracker. They are drawn as blue and yellow curves to show their possible tracks according to the MC information. The curves were interpolated to precisely show their trajectories in the magnetic field. The colors of the curves represent the charge; negatively charged particles are represented by yellow curves, whereas positively charged particles are represented by blue curves.
- *Neutral particles (except for neutrinos)* The neutral particles may interact with calorimeters, leaving energy deposits in the detector. These are plotted as straight black lines.
- *Neutrinos* In most cases, neutrinos will not interact with the detector, and they are always seen as the missing energy and missing momentum in event analysis. The software uses black dashed lines to represent the neutrinos.

In addition, the tracks visualized the complete spatial information of the MC particles, including their vertices and endpoints. The endpoint of each track is marked by a sphere, which can help physicists determine the moving direction of the MC particles. Furthermore, similar to other event objects, the display of MC information can be enabled or disabled through the user interface, and the tracks of MC particles can also be selected to provide more detailed information.

5 Features and applications

5.1 Features compared to other event display software

The CEPC event display software is forked from the original Phoenix framework and is developed as an independent branch, which not only inherits the advantages and features of the Phoenix framework but also introduces some new functionalities to meet the requirements for detector design and event analysis. Compared with the original Phoenix framework, the CEPC event display software provides full support for visualizing detector geometries and events. It also has some extra features to provide more diverse visualization and information, such as the display of MC truth information and enhanced geometry-clipping capabilities. Compared to the ROOT-based [51–53] or Unity-based [54–56] event display software, its advantages become even more obvious, making it the prime choice for CEPC event display in specific situations. Generally, the CEPC event display software has the following prominent features.

Easily accessible Based on three.js, the event display software can be either deployed locally or on a web server, and then the user can access it with any web browser that supports WebGL [57]. This includes most modern web browsers, such as Chrome, Firefox, Edge, Android browser, and iOS Safari. Similar to other software, browsers rely on GPUs to accelerate the rendering of 3D models; thus, event display software requires modern GPU support. However, with the aid of advancing front-end technologies, the need for powerful hardware is minimized, and it can operate efficiently even on mobile devices.

Lightweight One of the outstanding features of the Phoenix-based CEPC event display software is its lightweight nature. Unlike the ROOT framework, which heavily relies on third-party libraries, it can be easily bundled into a package and distributed to other devices. The software bundle occupies only 85 MB of storage space, enabling users to deploy the application on a personal computer without installing external software packages. The Unity editor also allows the source code to be built into a standalone application. However, the resulting applications are generally not cross-platform and usually occupy too much memory.

Fast The CEPC event display software also demonstrates great advantages on performance. To evaluate its performance, we deployed the software on a personal computer with an Intel i5-12400 processor and an Intel Iris Xe integrated graphics card and accessed it via the Chrome browser. The software renders the detector and event smoothly, achieving a refresh rate exceeding 30 frames per second (FPS), and the average memory usage is approximately 800 MB. During the test, we observed that the Iris Xe GPU utilization increased from 1% to approximately 18% during active visualization, indicating that the rendering process leverages both the CPU and GPU resources for efficient performance. Notably, the performance is closely related to the complexity of the detector geometry and event data. When rendering simpler geometries, the refresh rate can be even higher. In comparison, the typical refresh rate of ROOT-based event display software is approximately 10 FPS [52, 53], and a high latency between the users and computing servers can significantly reduce its performance.

Comprehensive visualization features Instead of writing basic logics for event display, the CEPC event display software inherits the functionalities of Phoenix framework, which provides complete features for detector and event visualization. Furthermore, the software extends the original set of functions of the Phoenix framework to fulfill the requirements for a deeper analysis, making it a practical choice for CEPC event display.

5.2 Potential applications

As previously introduced, the CEPC event display software provides a convenient and flexible way to visualize

the detector and events. It has great potential for application and further development. Some possible applications of the event display software are listed below.

Detector design and event analysis As an event display software, the primary goal of its development is to assist researchers in analyzing detector geometry and event data. The event display software provides the necessary features to display CEPC events and detectors, and it also has the flexibility to display other 3D geometries with EDM4hep format events, enabling users to gain insights into the detector structure and event signal. Researchers can receive feedback from the software and consequently optimize the detectors and algorithms.

Online monitoring Since CEPC event display software is developed using the web-based Phoenix framework, it is naturally suitable for online tasks. Instead of writing complex logic to fetch data from the Internet, the event display software can be easily extended to visit URLs to display online event data. Because the software can run in any modern web browser, users can check the experiment status using a smartphone. This will help the physicists instantly modify the status of the facility and enhance high-quality experimental data acquisition.

Further development As introduced before, the Phoenix framework is highly modular and extendable, the changes and extensions we develop for the event display software can be easily applied to other applications, even to other HEP experiments such as BESIII [58] and JUNO [59]. Moreover, the 3D library three.js [16] used by Phoenix has provided support for Virtual Reality (VR) [60] and Augmented Reality (AR) [61] features, and the software can be further developed to build VR/AR applications for the CEPC experiment.

6 Summary

With the aid of the Phoenix framework, we developed a web-based event visualization software for the CEPC experiment. It is purely based on front-end web technologies and can be easily deployed and run on any web server or personal computer. This software provides the necessary facilities for visualizing and displaying CEPC detectors and events, allowing researchers to gain deep insights into detector structures and event characteristics, thereby enhancing the continuous optimization of detectors, simulations, and reconstruction algorithms. It also has great potential for various applications, such as online monitoring and VR/AR programs.

Author contributions All authors contributed to the conception and design of the study. Material preparation, data collection and analysis were performed by Yu-Jie Zeng, Tian-Zi Song, Xue-Sen Wang, Yu-Mei Zhang, Tao Lin and Zheng-Yun You. The first draft of the manuscript was written by Yu-Jie Zeng and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data availability The data that support the findings of this study are openly available in Science Data Bank at https://cstr.cn/31253.11.sciencedb.j00186.00936 and https://doi.org/10.57760/sciencedb.j00186.00936.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

1. X. Lou, The circular electron positron collider. Nat. Rev. Phys. **1**, 232–234 (2019). https://doi.org/10.1038/s42254-019-0047-1
2. The CEPC Study Group, CEPC conceptual design report: Volume 1 - accelerator. arXiv:1809.00285 (2018). https://doi.org/10.48550/arXiv.1809.00285
3. The CEPC Study Group, CEPC conceptual design report: Volume 2 - physics & detector. arXiv:1811.10545 (2018). https://doi.org/10.48550/arXiv.1811.10545
4. J. de Blas, M. Cepeda, J. D'Hondt et al., Higgs boson studies at future particle colliders. J High Ener Phy. **2020** 139 (2020). https://doi.org/10.1007/JHEP01(2020)139
5. M. Farina, G. Panico, D. Pappadopulo, et al., Energy helps accuracy: electroweak precision tests at hadron colliders. Phys. Lett. B **772**, 210–215 (2017). https://doi.org/10.1016/j.physletb.2017.06.043
6. J. Peng, M. Zhao, X. Wang, et al., Prospect for measurement of *CP* -violating observables in $B_s^0 \to D_s^{\mp} K^{\pm}$ decays at a future *Z* factory. J. High Energ. Phys. 2025, 212 (2025). https://doi.org/10.48550/arXiv.2502.11172
7. Y.Q. Fang, S.T. Xin, Physics highlights at CEPC. Nucl. Part. Phys. Proc. **330–335**, 15–19 (2023). https://doi.org/10.1016/j.nuclphysbps.2023.04.011
8. J. Gao, CEPC project status. arXiv:2505.04663 (2025). https://doi.org/10.48550/arXiv.2505.04663
9. The HEP Software Foundation, J. Albrecht, A.A. Alves et al., A roadmap for HEP software and computing R&D for the 2020s. Comput. Softw. Big Sci. **3**, 7 (2019). https://doi.org/10.1007/s41781-018-0018-8
10. M. Ruan, H. Zhao, G. Li et al., Reconstruction of physics objects at the circular electron positron collider with arbor. Europ. Phys. J. C **78**, 426 (2018). https://doi.org/10.1140/epjc/s10052-018-5876-z
11. M. Liu, W. Li, X. Huang et al., Simulation and reconstruction of particle trajectories in the CEPC drift chamber. Nucl. Sci. Tech. **35**, 128 (2024). https://doi.org/10.1007/s41365-024-01497-z
12. X. Ai, Acts: A common tracking software. arXiv:1910.03128 (2019). https://doi.org/10.48550/arXiv.1910.03128
13. R. Brun, F. Rademakers, ROOT – an object oriented data analysis framework. Nucl. Instrum. Meth. A **389**, 81–86 (1997). https://doi.org/10.1016/S0168-9002(97)00048-X
14. M. Tadel, Overview of eve – the event visualization environment of root. J. Phys: Conf. Ser. **219**, 042055 (2010). https://doi.org/10.1088/1742-6596/219/4/042055
15. Javascript root. https://root.cern/js/
16. Mrdoob, Javascript 3d library three.js. https://threejs.org/
17. E. Moyse, F. Ali, E. Cortina et al., The phoenix event display framework. EPJ Web of Conf. **251**, 01007 (2021). https://doi.org/10.1051/epjconf/202125101007

18. Google, Angularjs – superheroic javascript mvw framework. https://angularjs.org
19. M. Bellis, R.M. Bianchi, S. Binet, et al., Hep software foundation community white paper working group – visualization. arXiv:1811.10309 (2018). https://doi.org/10.48550/arXiv.1811.10309
20. G. Aad, B. Abbott, D. Abbott et al., The ATLAS experiment at the CERN Large Hadron Collider: a description of the detector configuration for Run **3**. J. Instrum. **19**, P05063 (2024). https://doi.org/10.1088/1748-0221/19/05/p05063
21. The CMS Collaboration, S. Chatrchyan, G. Hmayakyan et al., The CMS experiment at the CERN LHC. Journal of Instrumentation **3**, S08004 (2008). https://doi.org/10.1088/1748-0221/3/08/S08004
22. P. Owen, N. Serra, Status and prospects of the LHCB experiment. Europ. Phys. J. Spec. Top. **233**, 225–240 (2024). https://doi.org/10.1140/epjs/s11734-023-01010-4
23. P. Calafiura, S. Farrell, H. Gray et al., Trackml: a high energy physics particle tracking challenge. 2018 IEEE 14th International Conference on e-Science (e-Science), Amsterdam, Netherlands, 2018, pp. 344-344 (2018). https://doi.org/10.1109/eScience.2018.00088
24. FASER Collaboration, Akitaka Ariga et al., Faser: Forward search experiment at the LHC. arXiv:1901.04468 (2019). https://doi.org/10.48550/arXiv.1901.04468
25. I. Agapov, M. Benedikt, A. Blondel et al., Future circular lepton collider FCC-ee: Overview and status. arXiv:2203.08310 (2022). https://doi.org/10.48550/arXiv.2203.08310
26. Adobe, Obj files: Uses & 3D applications. https://www.adobe.com/products/substance3d/discover/what-are-obj-files.html
27. T.K.G. Inc, glTF overview. https://www.khronos.org/gltf/
28. A. Pappas, B. Couturier, S. Ponce, New web based event data and geometry visualization for LHCB. J. Phys: Conf. Ser. **2438**, 012109 (2023). https://doi.org/10.1088/1742-6596/2438/1/012109
29. CEPC offline software prototype. https://code.ihep.ac.cn/cepc/CEPCSW
30. M. Frank, F. Gaede, M. Petric, et al., AIDASoft/DD4hep: v01–25-01. (2023). https://doi.org/10.5281/zenodo.7673417
31. M. Frank, F. Gaede, C. Grefe et al., DD4hep: A detector description toolkit for high energy physics experiments. J. Phys: Conf. Ser. **513**, 022010 (2014). https://doi.org/10.1088/1742-6596/513/2/022010
32. B.H. Frank Gaede1, Gerardo Ganis, Edm4hep and podio - the event data model of the key4hep project and its implementation. EPJ Web of Conferences. https://doi.org/10.1051/epjconf/202125103026
33. F. Gaede, T. Madlener, P. Declara Fernandez, et al., EDM4hep - a common event data model for HEPexperiments. In *41st* International Conference on High Energy physics - ICHEP2022, vol. 414, no.1237, Bologna, Italy, 6-13 July 2022. https://doi.org/10.22323/1.414.1237
34. G. Ganis, C. Helsens, V. Völkl, Key4hep, a framework for future HEP experiments and its use in FCC. Eur. Phys. J. Plus **137**, 149 (2022). arXiv:2111.09874, https://doi.org/10.1140/epjp/s13360-021-02213-1
35. T. Li, X. Huang, W. Huang et al., Core software for the super tau-charm facility. Mod. Phys. Lett. A **39**, 2440012 (2024). https://doi.org/10.1142/S0217732324400121
36. T.Z. Song, K.X. Huang, Y.J. Zeng et al., Detector description conversion and visualization in unity for high energy physics experiments. Front. Phys. **21**, 026201 (2026). https://doi.org/10.15302/frontphys.2026.026201
37. Z.Y. Yuan, T.Z. Song, Y.J. Zeng et al., Method for detector description conversion from DD4hep to Filmbox. Nucl. Sci. Tech. **35**, 146 (2024). https://doi.org/10.1007/s41365-024-01506-1
38. Z.Y. You, Y.T. Liang, Y.J. Mao, A method for detector description exchange among Root Geant4 and Geant3. Chin. Phys. C **32**, 572 (2008). https://doi.org/10.1088/1674-1137/32/7/012
39. S. Agostinelli, J. Allison, K. Amako et al., Geant4–a simulation toolkit. Nucl. Instrum. Meth. A **506**, 250–303 (2003). https://doi.org/10.1016/S0168-9002(03)01368-8
40. R. Chytracek, J. Mccormick, W. Pokorski et al., Geometry description markup language for physics simulation and analysis applications. IEEE Trans. Nucl. Sci. **53**, 2892–2896 (2006). https://doi.org/10.1109/TNS.2006.881062
41. Y.T. Liang et al., A uniform geometry description for simulation, reconstruction and visualization in the BESIII experiment. Nucl. Instrum. Meth. A **603**, 325–327 (2009). https://doi.org/10.1016/j.nima.2009.02.036
42. W.J. Dai, L. Xiao, G.X. Zhang et al., A Monolithic Active Pixel Sensor with a novel readout architecture for vertex detector in particle physics. JINST **20**, C03019 (2025). https://doi.org/10.1088/1748-0221/20/03/C03019
43. Y. Chang, H.R. Qi, X. She et al., Design and commission of TPC prototype integrated with 266 nm UV laser. JINST **20**, P02025 (2025). https://doi.org/10.1088/1748-0221/20/02/P02025
44. W. Pan, M. Dong, L. Wu et al., Drift chamber with cluster counting technique for CEPC. In *42nd International Conference on High Energy Physics (ICHEP2024)*, vol. 476, no.1101, Prague, Czech Republic, 18–24 July 2024. https://doi.org/10.22323/1.476.1101
45. H. Zhang, X. Zhao, C. Yuan et al., Design and simulation of a stereo crystal electromagnetic calorimeter for the CEPC. EPJ Web Conf. **320**, 00059 (2025). https://doi.org/10.1051/epjconf/202532000059
46. Y. Shi, Y. Zhang, M. Ruan et al., Design and optimization of the CEPC scintillator hadronic calorimeter. J. Instrum **17**, P11034 (2022). https://doi.org/10.1088/1748-0221/17/11/P11034
47. J. Gao, CEPC technical design report: accelerator. Radiat. Detect. Technol. Methods **8**, 1–1105 (2024). https://doi.org/10.1007/s41605-024-00463-y
48. M. Wang, F. Ning, L. Zhao et al., Conceptual Design of CEPC Detector Magnet. IEEE Trans. Appl. Supercond. **33**, 4500608 (2023). https://doi.org/10.1109/TASC.2023.3275213
49. Edm4hep webpage. https://github.com/key4hep/EDM4hep
50. W. Fang, X. Huang, W. Li et al., Refined drift chamber simulation in the CEPC experiment. EPJ Web Conf. **295**, 03034 (2024). https://doi.org/10.1051/epjconf/202429503034
51. Z.J. Li, M.K. Yuan, Y.X. Song et al., Visualization for physics analysis improvement and applications in BESIII. Front. Phys. **19**, 64201 (2024). https://doi.org/10.1007/s11467-024-1422-7
52. Z. You, K. Li, Y. Zhang et al., A ROOT based event display software for JUNO. J. Instrum. **13**, T02002–T02002 (2018). https://doi.org/10.1088/1748-0221/13/02/t02002
53. M.H. Liao, K.X. Huang, Y.M. Zhang et al., A root-based detector geometry and event visualization system for JUNO-TAO. Nucl. Sci. Tech. **36**, 39 (2025). https://doi.org/10.1007/s41365-024-01604-0
54. Unity real-time development platform. https://unity.com/
55. J. Zhu, Z. You, Y. Zhang et al., A method of detector and event visualization with unity in JUNO. J. Instrum. **14**, T01007 (2019). https://doi.org/10.1088/1748-0221/14/01/t01007
56. J. Li, Z. You, Event visualization with unity in BESIII experiment. In*42nd International Conference onHigh Energy Physics (ICHEP2024)*, vol. 476, no. 1047, Prague, Czech Republic, 18-24 July 2024. https://doi.org/10.22323/1.476.1047
57. Mozilla, Webgl: 2D and 3D graphics for the web. https://developer.mozilla.org/en-US/docs/Web/API/WebGL
58. M. Ablikim, Z.H. An, J.Z. Bai et al., Design and construction of the BESIII detector. *Nuclear Instruments and Methods* A 614, 345–399 (2010). https://doi.org/10.1016/j.nima.2009.12.050

59. A. Serafini, The JUNO experiment: status and physics potential. In *12th Neutrino Oscillation Workshop (NOW2024)*, vol. 473, no. 0004, Otranto, Lecce, Italy, 2-8 September 2024. https://doi.org/10.22323/1.473.0004
60. J. Zheng, K. Chan, I. Gibson, Virtual reality. IEEE Potentials **17**, 20–23 (1998). https://doi.org/10.1109/45.666641
61. J. Carmigniani, B. Furht, *Augmented reality: an overview*, (Springer, New York, New York, NY, 2011), pp. 3–46. https://doi.org/10.1007/978-1-4614-0064-6_1