



Unity-based virtual reality for detector and event visualization in JUNO experiment

Kai-Xuan Huang¹ · Tian-Zi Song¹ · Yu-Ning Su¹ · Cheng-Xin Wu¹ · Xue-Sen Wang² · Yu-Mei Zhang² · Zheng-Yun You¹

Received: 5 May 2025 / Revised: 18 June 2025 / Accepted: 2 August 2025 / Published online: 5 February 2026

© The Author(s), under exclusive licence to China Science Publishing & Media Ltd. (Science Press), Shanghai Institute of Applied Physics, the Chinese Academy of Sciences, Chinese Nuclear Society 2026

Abstract

Detector and event visualization are crucial components of high-energy physics (HEP) experimental software. Virtual reality (VR) technologies and multimedia development platforms, such as Unity, offer enhanced display effects and flexible extensibility for visualization in HEP experiments. In this study, we present a VR-based method for detector and event displays in the Jiangmen Underground Neutrino Observatory (JUNO) experiment. This method shares the same detector geometry descriptions and event data model as those in the offline software and provides the necessary data conversion interfaces. The VR methodology facilitates an immersive exploration of the virtual environment in JUNO, enabling users to investigate the detector geometry, visualize event data, and tune the detector simulation and event reconstruction algorithms. Additionally, this approach supports applications in data monitoring, physics data analysis, and public outreach initiatives.

Keywords Virtual reality · Event display · Unity · Detector geometry · JUNO

Kai-Xuan Huang and Tian-Zi Song have contributed equally to this work.

This work was supported by the National Natural Science Foundation of China (Nos. 12175321, W2443004, 11975021, 11675275, and U1932101), Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA10010900), 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

✉ 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, Sun Yat-sen University, Zhuhai 519082, China

1 Introduction

Visualization techniques are essential in all aspects of modern High-Energy Physics (HEP) experiments. In the Roadmap for HEP Software and Computing R&D for the 2020s [1] and the HEP Software Foundation Community White Paper [2], recommendations and guidelines for visualization tools, such as virtual reality (VR) technologies [3] in future software development, are specifically discussed, particularly regarding interactivity, detector geometry visualization, and event display. Compared to traditional visualizations, VR techniques offer a truly immersive perspective, which enhances the interactive experience with a better understanding of the detector geometry and event information. In recent years, some HEP experiments have developed VR applications for event displays and outreach. These include the Belle2VR software [4, 5] for the BelleII experiment [6], the ATLASrft platform [7, 8] for the ATLAS experiment [9], the CMS VR application [10] for the CMS experiment [11], and the Super-KAVE program [12, 13] for the Super-K experiment [14].

The development of VR applications typically involves game engines, such as Unity [15] or Unreal Engine [16]. Unity is a cross-platform engine that supports the

development of games, videos, animations, and architectural visualizations. It has been employed for detector visualization and event display in various HEP experiments, including BelleII, BESIII [17], ALICE [18], ATLAS [19], JUNO [20], and the Total Event Visualizer (TEV) of the CERN Media Lab [21], all of which achieve excellent visualization effects.

The Jiangmen Underground Neutrino Observatory (JUNO) [22–24] is situated underground in southern China with a 650 m rock overburden. The primary scientific goal of JUNO is to determine the neutrino mass hierarchy. Over an approximately seven-year operational period, JUNO is expected to determine the neutrino mass hierarchy with a significance of 3σ [25], and to measure the oscillation parameters Δm_{31}^2 , Δm_{21}^2 , and $\sin^2 \theta_{12}$, achieving a precision of 0.2% for Δm_{31}^2 , 0.3% for Δm_{21}^2 , and 0.5% for $\sin^2 \theta_{12}$ [26, 27], respectively.

Additionally, the JUNO experiment is capable of investigating various types of neutrinos, including earth, atmospheric, solar, and supernova neutrinos [22]. Its excellent energy resolution and large fiducial volume provide promising opportunities for exploring numerous essential topics in neutrino physics.

In this study, we developed a VR-based event display tool using Unity for JUNO. This software is compatible with various platforms through Head-Mounted Displays (HMDs) [28] and offers functionalities including the VR-based visualization of the JUNO detector, event displays for different types of data, interfaces for reading and converting event data information, and Spatial User Interface (Spatial UI) control features.

The remainder of this paper is organized as follows. In Sect. 2, we introduce VR-based software for HEP experiments. In Sect. 3, the software methodologies are described, including the JUNO VR framework, the data flow of the detector geometry and event data conversion, as well as interaction methods with the Spatial UI. The visualization of detector units and event data in the VR-based tool is introduced in Sect. 4. The potential for further applications is discussed in Sect. 5. Finally, the performance of the software is introduced in Sect. 6.

2 Visualization and VR

2.1 Unity and VR

In HEP experiments, physicists typically develop detector descriptions and event visualization tools within offline software frameworks. These event display tools are usually built upon the widely used HEP software, such as Geant4 [29] or ROOT [30], which provide user-friendly

visualization capabilities that facilitate software development. With the upgrades to ROOT and its EVE package [31], the development of event display tools has become more efficient. Several recent HEP experiments, including ALICE, CMS [11], BESIII [32], JUNO [33, 34], and Mu2e [35], adopt ROOT EVE for developing event display software. However, owing to the limited visualization technique support in ROOT, its display capabilities do not fully meet the diverse requirements of physicists, and most ROOT applications remain confined to the Linux platform.

To enhance visualization quality, interactivity, and multi-platform support, several event display tools have been developed based on external visualization software. Unity is widely applied in the field of HEP and is used in projects including BelleII, BESIII, ALICE, ATLAS, and JUNO. Unity is a professional video and game development engine based on C#, and visualization software built on Unity offers several advantages.

- *Impressive visualization quality.* Unity, a widely adopted professional 3D engine in the industry, offers advanced visual capabilities that surpass those of the traditional software used in HEP, such as ROOT. Additionally, its continuous updates enable HEP visualizations to remain aligned with cutting-edge developments in graphics technology.
- *Cross-platform support.* The comprehensive multi-platform support of Unity enables seamless export and deployment of projections across a range of operating systems, including Windows, Linux, macOS, iOS, Android, and web browsers. This functionality ensures that the same visualization project can be accessed across various platforms, thereby minimizing the development effort and streamlining maintenance tasks.
- *High-quality VR rendering and performance optimization.* Unity supports modern graphics technologies, such as real-time lighting, global illumination, and physics-based rendering. Light behaves according to the principles of physics, including energy conservation and Fresnel reflections [36], resulting in more realistic and immersive graphical effects in VR. These features are crucial for enhancing details such as lighting, shadows, textures, and environmental interactions, significantly improving the user's sense of immersion. Additionally, Unity optimizes the VR performance by rendering separate images for each eye, providing a dual-eye perspective while maintaining smooth rendering and minimizing motion blur and latency.
- *VR HMDs compatibility.* Unity supports most popular VR HMDs, including Meta Quest 2 and Quest 3 [37], HTC Vive [38], Valve Index [39], and Vision Pro [40]. Using the extended reality interaction toolkit in Unity,

developers can easily create interactive applications for various devices without device-specific coding.

Additionally, Unity provides a fast turnaround during the development cycle. Projects can be executed immediately, running quickly on VR devices for easier debugging without the need to compile and link executable files [41].

Compared to 3D-based event visualization software, VR technology significantly enhances the user's visual experience. VR applications are typically conducted using HMDs. According to Steam VR hardware statistics [42], more than half of the users utilize Meta Quest 2 and Quest 3. These devices, based on the Android operating system, offer sufficient immersion and are widely used in various fields, including gaming, social interaction, and education. Equipped with accelerometers, gyroscopes, and cameras, these devices can track the user's head and hand movements, enabling interaction and navigation within virtual environments. Additionally, the controllers facilitate interaction with the Spatial UI in the virtual environment. VR technology provides synthesized sensory feedback, creating a strong sense of immersion and presence in a simulated environment.

Most HEP experiments are typically conducted in underground or restricted areas that are inaccessible during data collection. VR technology enables the public to explore these experiments in an immersive environment and observe detector operations and event data collection. This offers a fundamental understanding of the types of scientific research conducted in HEP, which is highly beneficial for both educational and outreach purposes.

Furthermore, by simulating particle emissions and their interactions with detectors, VR provides physicists with an immersive platform for refining offline simulations and reconstruction software [43–46]. It can also enhance the simulation accuracy. For JUNO, considering the deformation of the stainless-steel truss, offsets need to be applied to the PMT positions based on limited survey data [47–49]. Overlap checks and position tuning using the VR event display tool are particularly helpful. Additionally, VR enables physicists to analyze rare events as though they are physically present within the inner detector environment, providing an alternative approach for data analysis and inspiring creativity.

2.2 VR application in HEP

In recent years, VR applications have been developed for several HEP experiments, event visualization, and outreach. These software include Belle2VR [5] for the BelleII experiment, ATLASrift [7, 8] for the ATLAS experiment, and Super-KAVE [12, 13] for the Super-K experiment.

Belle2VR is an interactive VR visualization tool developed using Unity, which is designed to represent subatomic particle physics. This application allows users to explore the BelleII detector and observe particle jets generated in high-energy e^+e^- collisions. The Super-KAVE application immerses the user in a scaled representation of the Super-K detector, allowing them to explore the virtual space, switch between event datasets, and change the visualization modes [12, 13]. In addition to providing VR modes for exploring the detector and standard event displays, the application features a supernova event visualization technique that simulates the conversion of a star into a supernova. This leads to thousands of neutrino events within approximately ten seconds. It serves as a valuable outreach tool, offering a new example of visualization techniques for various applications in neutrino particle physics. ATLASrift, a VR application developed for the ATLAS experiment, is primarily used for data visualization and outreach [9]. Users move around and inside the detector, as well as explore the entire underground experimental cavern and its associated facilities, including shafts, service halls, passageways, and scaffolds.

3 Methodologies

VR technology provides an immersive experience for users. However, the development of comprehensive event display software utilizing VR for HEP experiments still involves significant challenges.

The first challenge is to convert the detector geometry, typically based on Geant4 simulations, into a format such as FBX [50] which can be imported into Unity. Given that detectors usually consist of tens of thousands of components, manually creating the geometry imposes a significant workload. Another significant challenge is extracting and converting event information into a structure that is compatible with Unity. In HEP experiments, the fundamental information for event display is typically defined by offline software and stored in the ROOT format. However, because Unity does not support the direct reading of ROOT files, a dedicated conversion process is required. Additionally, a bijective mapping was established to link the detector unit identifiers used in the offline software [51] with the names assigned to the corresponding geometries in Unity.

This section introduces the software architecture and data flow of the JUNO VR program. We describe the process of detector geometry conversion, exchange of essential event information from offline software to Unity, and strategy for matching detector units. Additionally, we discuss the construction of the Spatial UI and provide an overview of its functionality.

3.1 Software structure and data flow

The event display software should provide visualization capabilities, including detector geometry, event data information at different levels, and interactive controls. For JUNO VR software visualization, the first step involves converting and importing the detector geometry and event data information into Unity for display, followed by the development of the interactive controls. As shown in Fig. 1, the JUNO event display software consists of four components.

- *Detector geometry conversion.* The geometric models of the detector were constructed using Geant4 in the detector simulation and initially stored in a Geometry Description Markup Language (GDML) file [52]. The GDML file is then automatically converted to the FBX format using the GDML-FBX conversion tool [17, 53], which is compatible with import into Unity.
- *Event data conversion.* The Event Data Model (EDM) [54] encompasses various types of event information exchanged between different components of the JUNO online and offline software, including data acquisition, simulation, calibration, and reconstruction. The event information for the JUNO VR event display was extracted from the offline software EDM [55]. By combining the detector identifier and Unity geometry name matching rules, the detector information is remapped, generating event information that Unity can directly import and conform to the geometry hierarchy in Unity.
- *Detector and event information visualization.* The detector geometry, simulation, and reconstruction information, as well as the hit information and their associations, were visualized in Unity. By adjusting the material properties and combining Unity's layers, lighting, and rendering

effects, an immersive and outstanding visualization experience in the VR mode was achieved.

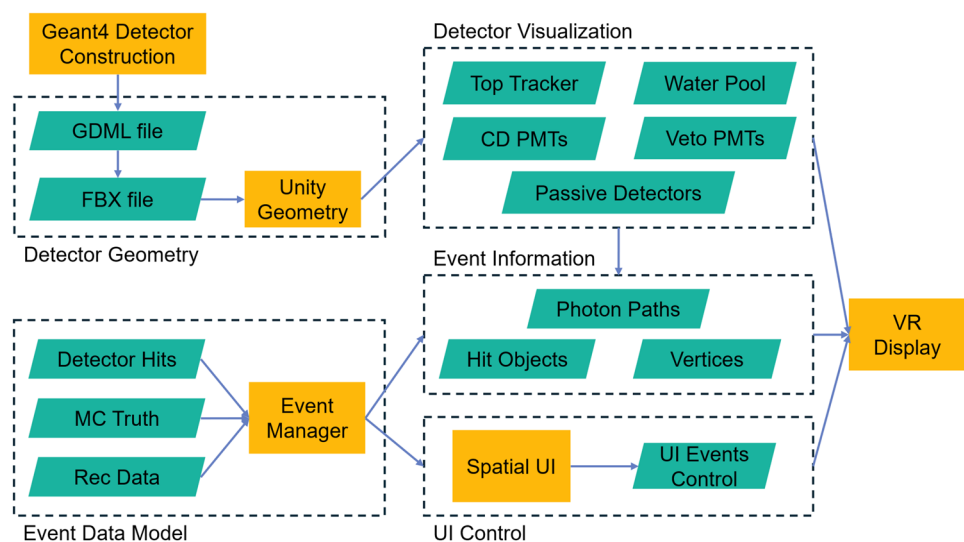
- *Spatial UI and interactive control.* The Spatial UI is designed to facilitate visualization and interaction with the detector and event information. It includes the sub-detector geometry panel and the event display panel, which allow users to control the display of subdetectors, switch between event types, and manage the event display process. Interactive control was enabled through the Meta Quest 3 controller, with distinct functions assigned to the joystick and various buttons. These functions include controlling the visibility of each panel, navigating within the 3D virtual detector environment, and switching perspectives.

3.2 Detector geometry conversion

The detector geometry in HEP experiments is typically complex, consisting of up to millions of detector units. The description of these detectors is commonly developed using specialized geometric languages, such as GDML and Detector Description for High-Energy Physics (DD4hep) [56, 57]. The JUNO experiment, along with BESIII, PHENIX [58], and LHCb [59], uses GDML to describe and optimize the geometry of detectors for conceptual design and offline software development. GDML is a detector description language based on Extensible Markup Language (XML) [60] that describes detector information through a set of textual tags and attributes, providing a persistent description of the detector. The geometry description files of detectors typically include essential information about the detector model, such as lists of materials, positions, rotations, solids, and hierarchical structures of the detector.

Because the GDML format does not directly support import into Unity, some previous HEP applications

Fig. 1 (Color online) The software framework and data flow in JUNO VR



involving Unity typically required the manual construction of geometric models. Given that HEP detectors are usually highly complex, creating 3D detector models in Unity is particularly challenging. However, Unity supports the direct import of several 3D file formats, including FBX, DAE [61], DXF [62], and OBJ [63]. Among these, FBX stands out as a widely used 3D asset format because of its ability to handle intricate scene structures. This includes not only geometry but also animations, materials, textures, lighting, and cameras, making it a highly suitable choice for HEP applications involving complex 3D models.

A method that can automatically convert GDML or DD4hep to the FBX format is essential for detector construction in Unity. Several studies have proposed automated methods for converting GDML files into FBX files, significantly facilitating Unity-based development. For instance, the BESIII collaboration group suggested using FreeCAD [64], a 3D CAD and modeling software, in conjunction with the CAD data optimization software Pixyz [65], with the STEP [66] format as an intermediate conversion format [17]. The CMS collaboration group employs SketchUp software for auxiliary data conversion [67].

Recently, methods have been proposed to directly convert GDML files to FBX files [53]. This research, based on the above method, enables a fast and automatic conversion process from GDML to FBX, which can be completed in just a few minutes and saves significant time in the conversion process. This approach is particularly beneficial during the recent geometric updates of the JUNO detector at the commissioning stage, enabling the swift conversion of the updated FBX file, which includes the latest geometry model of the real detector units after installation.

3.3 Event data conversion

In HEP experiments, event data are typically stored in files with binary raw data format or ROOT format. ROOT, an efficient data analysis framework, is widely adopted for high-performance data input and output operations. However, because Unity cannot directly read ROOT files, it is necessary to extract the required event information based on the EDM and convert it into a text format that Unity can process.

The essential information for event display comprises three main components: detector unit hits, Monte Carlo (MC) truth, and reconstruction data. The detector unit hits include the hit time and hit charge for each detector unit, such as a PMT. MC truth provides detailed truth information, such as simulated vertices and photon trajectories (including 3D coordinates and propagation with time), which facilitates a deeper analysis of the particle direction and relative velocity. Reconstruction data typically contain the reconstructed vertex positions, energy information, and

additional track information for muon events like direction. Together, this information serves as the foundation for developing event display functionalities and interactive control modules based on the Spatial UI.

Furthermore, the identifiers used for the detector units in offline software may differ from the names of the geometric objects in Unity. In HEP experiments, the detector identifier system assigns a unique ID to each detector unit and plays a critical role in various applications, including data acquisition, simulation, reconstruction, and analysis. Therefore, establishing an accurate mapping between the detector identifiers in offline software and geometric objects, such as PMT in Unity, is essential to ensure the accurate display of an event. Based on the EDM readout rules and leveraging the mapping between the identifier module and geometric objects in Unity, an automated readout and conversion interface was developed to export event display information.

For the JUNO VR software, multiple types of datasets are provided, including radioactive background, Inverse Beta Decay (IBD) [68], cosmic-ray muons, and other types of events. The event display dataset was designed to encompass both simulated and real-data event types. Simulated events were produced using the JUNO offline software to facilitate detector simulation, commissioning, and optimization of reconstruction algorithms. Because JUNO has not yet commenced formal data acquisition, real-data events are obtained from the Data Challenge dataset [69], which has data structures identical to those expected during actual operation. With the event data conversion interface, datasets with various types of data are ready to be displayed in the Unity-based visualization and VR software.

3.4 Spatial UI and interactive control

The Spatial UI serves as an interface that facilitates interaction between the user and the VR application. For the JUNO VR project, we developed two Spatial UIs: the subdetector geometry control panel and the event display control panel, as shown in Fig. 2.

The subdetector geometry panel primarily controls the visualization attributes of the geometries of various subdetectors, including the Central Detector (CD) large PMTs, CD small PMTs, Top Tracker, and water pool PMTs. Detailed information about the subdetectors of JUNO is provided in Sect. 4.1. In addition to the sensitive detectors such as PMTs, an “Other Structure” toggle controls the display of passive structures, such as the steel structure, acrylic ball, PMT support structures, and liquid filling pipelines. Additionally, the “Data Type” drop-down is used to switch between different types of events collected during real-data acquisition or from simulation. The “Photon Trail Mode” toggle enables the switching of display modes for photon paths, either



Fig. 2 (Color online) The Spatial UI in JUNO VR. On the left is the JUNO VR event display control panel, and on the right is the sub-detector geometry control panel

represented by green lines or in a manner closely resembling the particle motion.

The event display panel is designed to implement the core functionality for event visualization, which includes a toggle for switching the display mode between simulation and data types, a slider for controlling the display of an event with its timeline evolution, a drop-down menu for selecting different types of events, and a button to play the event animation. A “Draw Hit” button initiates the animation of the full event hit process, which plays within a period of time window, with the time slider moving in sync with the event timeline, enabling the user to track the current time of the event.

Interactive control is achieved using controllers, gesture operations, eye tracking, and other input methods in HMDs. The following discussion focuses on testing the interactive control for the Meta Quest 3. For other HMDs, the cross-platform support provided by the extended reality interaction toolkit in Unity minimizes the development differences between various devices. Simple adaptations based on the specific features of HMDs are sufficient for operation.

The controller buttons resemble a typical gamepad with the addition of side buttons. The X&Y buttons on the left controller are used to control the visibility of the sub-detector geometry panel. When displayed, the position of this panel is based on the user’s orientation and appears at the front left of the user’s view. Users can drag or hide the panel to avoid obstructing their view when visualizing events. The A&B buttons on the right controller are used to control the visibility of the event display panel. When displayed, the panel appeared at the front right of the user’s view. Based

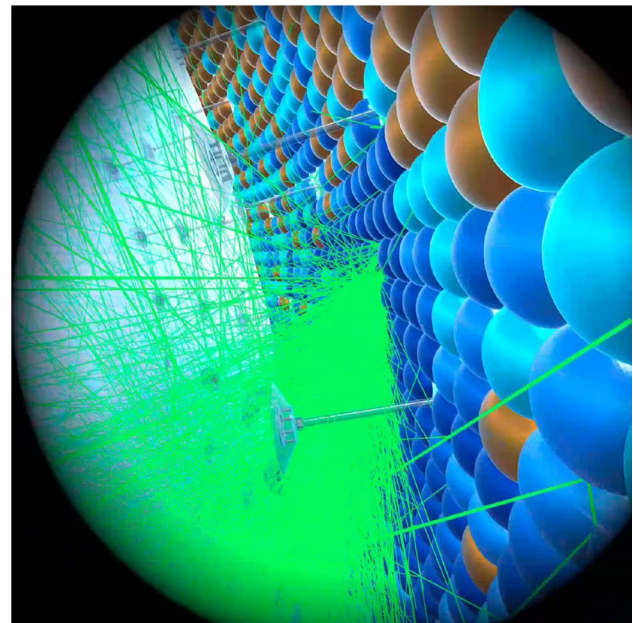


Fig. 3 (Color online) User perspective during motion while checking the display information of a simulated muon event in the JUNO VR application. The CD small PMTs are not shown. Detailed information about the subdetectors of JUNO is provided in Sect. 4.1

on the gyroscope and accelerometer hardware of the Meta Quest 3, these planes were always oriented perpendicular to the user’s view orientation.

The joystick on the left controller controls the user’s 3D movement, based on both the controller input and the user’s view orientation. For example, when the user’s head orientation is directed toward the upper right, pushing the joystick upwards moves the user in the virtual space toward that direction. Figure 3 illustrates the user’s viewpoint during motion in the JUNO VR application. The event depicted is a simulated muon event. Additional details presented in the figure are described in Sect. 4. The joystick on the right controls the user’s viewpoint direction. Additionally, the user can change their head orientation to switch perspectives. The side button was used for interaction confirmation. Furthermore, when interacting with the Spatial UI, if the controller’s laser pointer touches the corresponding component, audio and highlight feedback are provided, making the interaction smoother for the user.

4 Visualization in JUNO

This section introduces the visualization effects in JUNO VR, including detector geometry, hit distribution for different types of events, MC true information, and display of event reconstruction outputs.

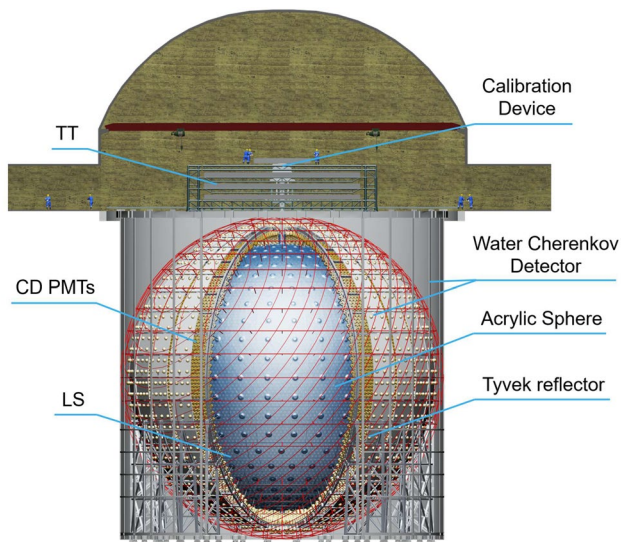


Fig. 4 (Color online) Schematic view of the JUNO detector

4.1 Detector units

The schematic design of the JUNO detector is illustrated in Fig. 4 [23]. The detector includes a water pool, CD [47], and Top Tracker [70]. The CD is the heart of the JUNO experiment and is filled with 20 kt of liquid scintillator [71, 72] to serve as the target for neutrino detection. The liquid scintillator is housed within a spherical acrylic vessel with a thickness of 120 mm and an inner diameter of 35.4 m. This vessel is supported by a spherical stainless-steel structure with an inner diameter of 40.1 m. To detect photons, the CD is equipped with 17,612 20-inch PMTs and 25,600 3-inch PMTs. The CD is surrounded by a water pool containing 35 kt of highly purified water, which effectively shields the detector from external radioactivity originating from the surrounding rocks. The water pool is also instrumental in vetoing cosmic-ray muons, with 2,400 20-inch PMTs deployed as part of the water Cherenkov detector. The Top Tracker, located at the top of the water pool, plays a key role in measuring and vetoing muon tracks [73, 74].

As described in Sect. 3.2, the JUNO detector geometry was converted from the GDML file and matched between the identifier module and Unity geometry for each detector unit. The visualization effects of the entire JUNO detector in the VR application are shown in Fig. 5.

The light blue cylindrical structure represents the water pool, with the water pool PMTs positioned outward, as indicated by the yellow portion of the spherical structure. At the top of the water pool, the reddish-brown structure represents the Top Tracker detector. From the interior view in the JUNO VR, the spherical acrylic vessel is shown in light gray, as depicted in Fig. 2, although it is almost fully transparent in reality to allow more photons to pass through.

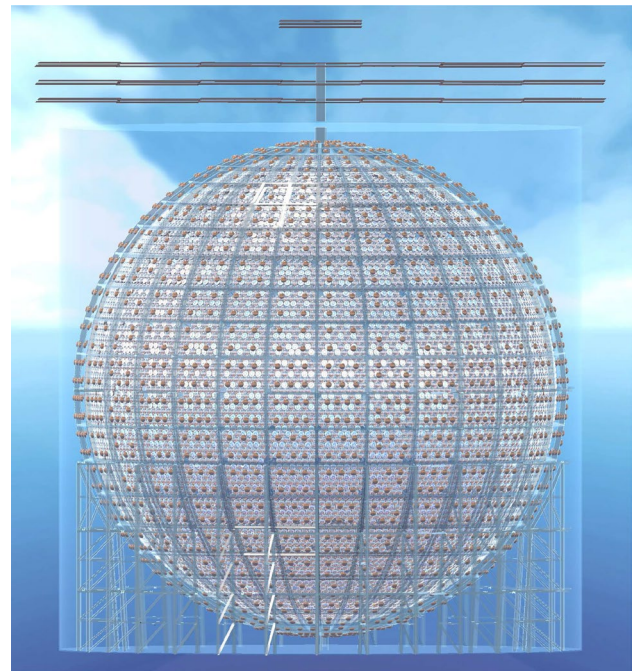


Fig. 5 (Color online) JUNO detector in the VR application

Surrounding this vessel is a stainless-steel structure, shown in dark gray in Fig. 5. The CD detector PMTs, oriented toward the center of the sphere, are designed to receive photons with their photocathodes, so that only the white tail structures of every PMTs are visible in Fig. 5.

Owing to the hardware capabilities of Meta Quest 3, there is no need to optimize the grid of detector units or replace them with simplified geometric shapes. Most of the geometric details of the detector units are preserved, achieving effects that are difficult to accomplish in event displays based on the ROOT. Additionally, for the detector units, to more closely replicate the effect of real PMTs, we assigned different material properties to the detector units, including visualization attributes such as color, reflectivity, and metallicity, to achieve the best display effect.

4.2 MC simulation event display

MC simulation is crucial for detector design and assists physicists in evaluating the performance of the detector and tuning the reconstruction algorithms. There are various types of signal and background events in JUNO, and we currently focus primarily on radioactive backgrounds, IBD signals, and muon events.

The IBD event, $\nu_e + p \rightarrow e^+ + n$, is the major signal event for detecting electron anti-neutrinos in the JUNO experiment [22, 23]. JUNO identifies and reconstructs IBD events by detecting positron and neutron capture signals.

This dual-signal characteristic helps to effectively identify antineutrino signal events while suppressing the large background events.

For the IBD event, there are both positron and neutron signals, whose photon paths are displayed in green and red, respectively, as shown in Fig. 6. The triggered detector units are color-coded from cyan to dark blue based on the number of hits in the event, with bluer colors indicating a higher number of hits. PMTs that were not triggered are displayed in yellow by default. Furthermore, in the time evolution of an event, the color of the fired PMTs changes with time according to the associated timing information. The neutron-induced photon paths are delayed by approximately $170 \mu\text{s}$ relative to those from the positron, and this delay can be visualized using the time slider in the JUNO VR environment.

One major background event type is cosmic-ray muon events. Muons are secondary particles produced by high-energy cosmic rays in the earth's atmosphere and possess strong penetrating power. Despite JUNO being located 650 m deep underground, a small fraction of muons can still penetrate the overlying shielding and enter the detector, generating muon events.

Figure 7 presents the event information for the simulated muon events. The photon trajectories are represented by light green lines. These paths gradually extended over time, depicting the propagation of photons. In the simulated event shown, the directions of these photon paths may change, indicating their interactions with the detector materials. For a muon event, as a muon penetrates the detector, it continuously produces photons while depositing its energy in the liquid scintillator.

Event reconstruction plays a key role in JUNO data processing, reconstructing the vertex and energy of an event,

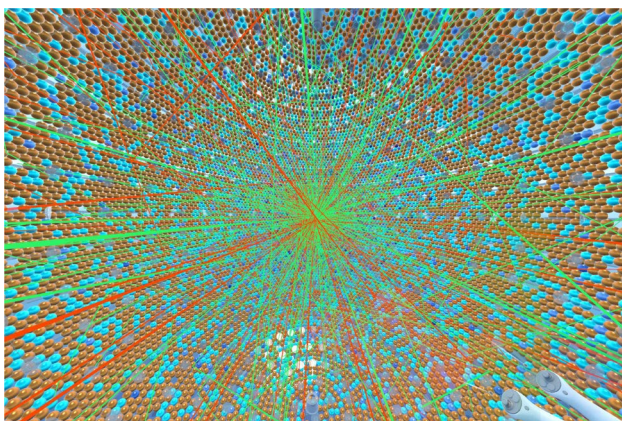


Fig. 6 (Color online) Event display for a simulated IBD event in the JUNO VR application. The green lines represent the photon paths of the positron, and the red lines indicate the photon paths of the neutron. The yellow spheres represent PMTs that are not triggered, whereas the spheres with a color gradient from light blue to blue indicate PMTs with an increasing number of hits

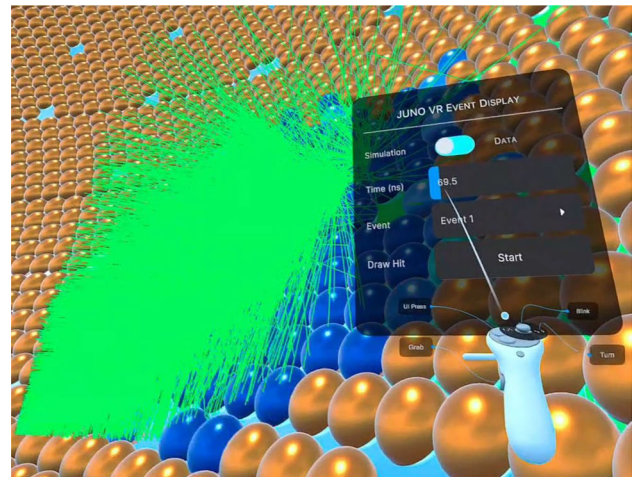


Fig. 7 (Color online) Event display for a simulated muon event in the JUNO VR application. The green lines represent the photons generated along the path of the muon that penetrated the detector. The controllers and lasers emitted from the controllers represent the user's interactive control

which is essential for determining the neutrino mass hierarchy. For point-like events, such as IBD signals, almost all photon paths originate from the same event vertex. Figure 8 shows the reconstructed vertex and MC truth. The initial particle production vertex (red sphere), derived from the MC truth, indicates where the positron is created. The weighted energy deposit vertex (green sphere) marks the positron's annihilation point in the liquid scintillator. The reconstructed vertex (purple sphere) is produced using the event reconstruction algorithm. The reconstruction bias (light yellow line) represents the discrepancy between the reconstructed vertex and the energy deposit vertex. A

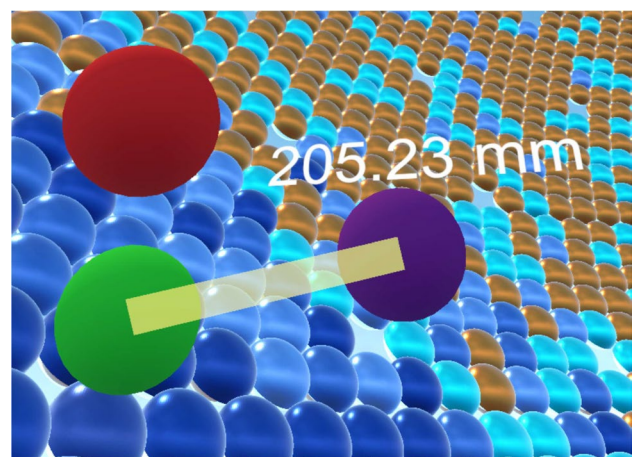


Fig. 8 (Color online) Comparison of the reconstructed vertex (purple) with the weighted energy deposit vertex (green) and the particle production vertex (red) from MC truth in a simulated event. The yellow line indicates the reconstruction bias

shorter distance indicates a more accurately reconstructed vertex. In an ideal scenario, the reconstructed vertex converges to the true vertex.

4.3 Real-data event display

For the real-data event, we utilize the Data Challenge dataset [69], whose data structures and processing pipeline are identical to those employed during data acquisition. This ensures that the software functions seamlessly once the experiment enters formal operation. The event composition in this dataset is the same as that in the MC simulation, encompassing radioactive background events, IBD signals, and muon events.

Figure 9 presents the event information for a muon event derived from real-data events. The reconstructed muon travels through the detector along the magenta line. The left and right sides represent the reconstructed incident and exit points of the muons, respectively. A time offset was established by dividing the track distance by the speed of light. Users can observe the trajectory of the muon using the Spatial UI. Because the exact point of photon emission along the path cannot be determined, the photon information is not displayed in this mode. Using the reconstructed hit time, the corresponding point on the trajectory was linked to the relevant PMT unit. Once the photon particles arrive at the PMT units, the triggered PMTs will change color accordingly.

Moreover, by exploiting Unity's robust visualization capabilities, a specialized mode was developed to simulate photon paths using particle-like effects instead of simple line trajectories to display the propagation of particles more realistically.

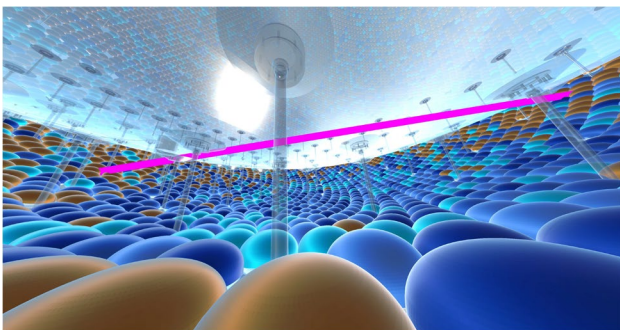


Fig. 9 (Color online) Display of a reconstructed muon event from the datasets in the JUNO VR application. The translucent part represents the CD acrylic sphere and its supporting components. The magenta line indicates the reconstructed muon track by connecting the points where the muon enters and exits the JUNO detector

5 Applications

The JUNO VR software provides an immersive interactive experience, allowing users to intuitively understand the detector structure and event information. Some features and applications of the visualization software are listed as follows.

Data quality monitoring. The data quality monitoring system [75–78] is designed to identify data issues promptly, ensuring the acquisition of high-quality data. During the future data-acquisition phase, event information can be extracted in real time and automatically from the reconstructed files of the data quality monitoring system. Based on Unity-supported databases such as SQLite, event information can be transmitted from the data quality monitoring server to the JUNO VR software. This enables immersive visualization of the detector operation status and event information during the data-acquisition phase. For example, an animation of a real-time data-acquisition event is automatically played every 30 s. Through immersive visualization, shifters can easily monitor anomalies such as hot or dead PMT channels.

Physics analysis. Physical analysis involves in-depth research of neutrino events on extract physical parameters, validate theoretical models, search for rare signals, and uncover new phenomena. This requires a detailed analysis of large volumes of complex data. Through the VR interface, researchers can reconstruct an immersive view of the event in three-dimensional space, allowing them to freely explore the data, observe event details from multiple perspectives and identify potential patterns and anomalies.

Outreach. HEP experiments are usually difficult for the public and students to understand because of their complex theoretical and experimental contents. Based on the VR application, students can understand the structure of the JUNO detector and the processing of signal and background events through interactive operations, thereby enhancing their engagement and understanding of the physics and principles of the HEP experiments. Visualization programs, including VR, stand out in the fields of education and public outreach. Owing to Unity's cross-platform support and compatibility with various HMDs, the completed project can be exported to different platforms and utilized with different HMDs, meeting the requirements of various outreach scenarios.

6 Performance

In experimental evaluations conducted on the mainstream VR device, the Meta Quest 3, the JUNO VR application was capable of processing a variety of event types and demonstrated sufficient computational performance. During testing, the CPU utilization of the device remained below 70%, GPU utilization remained below 40%, and the display maintained a stable refresh rate of 72 frames per second. The software's interactive response primarily depends on the event type. For muon events, which contain a larger volume of hit information, the latency when switching between events is approximately 3 s; for IBD and radioactive background events, it is approximately 1 s.

The event display of the JUNO VR application underwent rigorous testing, and the application was capable of processing both simulated and real-data events.

7 Summary

VR technology significantly enhances the visualization effects of HEP experiments. A JUNO VR application for detector and event visualization was developed using Unity. By converting GDML to the FBX format, an efficient construction of the complex detector geometry in Unity was achieved. An event data conversion interface was created by matching the detector identifier module and detector geometry hierarchy in Unity. Through Spatial UIs, users can easily control the display of various subsystems for detector and event visualization.

With the ongoing construction of the JUNO experiment, the VR event display software was successfully developed, and more features are expected to be added in future updates. VR technology offers an immersive, interactive experience and holds great potential in areas such as offline software development, data acquisition, physics analysis, education, and public outreach.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Kai-Xuan Huang, Tian-Zi Song, Yu-Ning Su, Cheng-Xin Wu, Xue-Sen Wang, Yu-Mei Zhang, and Zheng-Yun You. The first draft of the manuscript was written by Kai-Xuan Huang, 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.j00186.00915> and <https://doi.org/10.57760/sciencedb.j00186.00915>.

Declarations

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

References

1. HEP Software Foundation Collaboration, 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>
2. M. Bellis, R.M. Bianchi, S. Binet et al., HEP software foundation community white paper working group—Visualization (2018). <https://doi.org/10.48550/ARXIV.1811.10309>
3. I. Wohlgenannt, A. Simons, S. Stieglitz, Virtual reality. *Bus. Inf. Syst. Eng.* **62**, 455–461 (2020). <https://doi.org/10.1007/s12599-020-00658-9>
4. M. Bender, T. Kuhr, L. Piilonen, Belle II virtual reality projects. *EPJ Web Conf.* **214**, 02028 (2019). <https://doi.org/10.1051/epjconf/201921402028>
5. Z. Duer, L. Piilonen, G. Glasson, Belle2vr: a virtual-reality visualization of subatomic particle physics in the Belle II experiment. *IEEE Comput. Graphics Appl.* **38**, 33–43 (2018). <https://doi.org/10.1109/MCG.2018.032421652>
6. Belle-II Collaboration, Belle II technical design report. arXiv:1011.0352 (2010). <https://doi.org/10.48550/arXiv.1011.0352>
7. ATLAS Collaboration, Virtual reality and game engines for interactive data visualization and event displays in HEP, an example from the ATLAS experiment. *EPJ Web Conf.* **214**, 02013 (2019). <https://doi.org/10.1051/epjconf/201921402013>
8. I. Vukotic, E. Moyse, R.M. Bianchi, ATLASrft—a virtual reality application. in *Meeting of the APS Division of Particles and Fields*, arXiv:1511.00047 (2015). <https://doi.org/10.48550/arXiv.1511.00047>
9. ATLAS Collaboration, The ATLAS experiment at the CERN Large Hadron Collider. *J. Instrum.* **3**, S08003 (2008). <https://doi.org/10.1088/1748-0221/3/08/S08003>
10. CMS Collaboration, Leveraging virtual reality for visualising the CMS detector. PoS (ICHEP2024) 1171, <https://pos.sissa.it/476/1171/>. Accessed on 16 June 2025
11. CMS Collaboration, The CMS experiment at the CERN LHC. *J. Instrum.* **3**, S08004 (2008). <https://doi.org/10.1088/1748-0221/3/08/S08004>
12. B. Izatt, K. Scholberq, R.P. McMahan, Super-kave: an immersive visualization tool for neutrino physics. in *2013 IEEE Virtual Reality (VR), 2013*, pp. 75–76 (2013). <https://doi.org/10.1109/VR.2013.6549370>
13. E. Izatt, K. Scholberg, R. Kopper, Neutrino-kave: an immersive visualization and fitting tool for neutrino physics education. in *2014 IEEE Virtual Reality (VR), 2014*, 83–84 (2014). <https://doi.org/10.1109/VR.2014.6802062>
14. Y. Suzuki, The Super-Kamiokande experiment. *Eur. Phys. J. C* **79**, 298 (2019). <https://doi.org/10.1140/epjc/s10052-019-6796-2>
15. W. Goldstone, *Unity Game Development Essentials* (Packt Publishing Ltd, 2009)
16. A. Sanders, *An Introduction to Unreal Engine 4* (AK Peters/CRC Press, 2016)
17. K.-X. Huang, Z.-J. Li, Z. Qian et al., Method for detector description transformation to Unity and application in BESIII. *Nucl. Sci. Tech.* **33**, 142 (2022). <https://doi.org/10.1007/s41365-022-01133-8>
18. ALICE Collaboration, ALICE: Physics performance report, volume I. *J. Phys. G* **30**, 1517–1763 (2004). <https://doi.org/10.1088/0954-3899/30/11/001>

19. J. Pequeno, Camelia webpage. <https://pdgusers.lbl.gov/~pequeno/camelia>. Accessed on 15 March 2025
20. J. Zhu, Z.-Y. You, Y.-M. 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>
21. CM Lab, CERN TeV visualization framework webpage. <https://gitlab.cern.ch/CERNMediaLab/>. Accessed on 15 Mar 2025
22. JUNO Collaboration, JUNO physics and detector. *Prog. Part. Nucl. Phys.* **123**, 103927 (2022). <https://doi.org/10.1016/j.pnpnp.2021.103927>
23. JUNO Collaboration, JUNO conceptual design report. arXiv:1508.07166 (2015). <https://doi.org/10.48550/arXiv.1508.07166>
24. F.P. An, G.P. An, Q. An et al., Neutrino physics with JUNO. *J. Phys. G* **43**, 030401 (2016). <https://doi.org/10.1088/0954-3899/43/3/030401>
25. A. Abusleme, T. Adam, S. Ahmad et al., Potential to identify neutrino mass ordering with reactor antineutrinos at JUNO. *Chin. Phys. C* **49**, 033104 (2025). <https://doi.org/10.1088/1674-1137/ad7f3e>
26. JUNO Collaboration, Sub-percent precision measurement of neutrino oscillation parameters with JUNO. *Chin. Phys. C* **46**, 123001 (2022). <https://doi.org/10.1088/1674-1137/ac8bc9>
27. J.P. Athayde Marcondes de André, N. Chau, M. Dracos et al., Neutrino mass ordering determination through combined analysis with JUNO and KM3NeT/ORCA. *Nucl. Instrum. Meth. A* **1055**, 168438 (2023). <https://doi.org/10.1016/j.nima.2023.168438>
28. J. E. Melzer, K. Moffitt, Head mounted displays (1997)
29. GEANT4 Collaboration, GEANT4-a simulation toolkit. *Nucl. Instrum. Meth. A* **506**, 250–303 (2003). [https://doi.org/10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8)
30. R. Brun, A. Gheata, M. Gheata, The ROOT geometry package. *Nucl. Instrum. Meth. A* **502**, 676–680 (2003). [https://doi.org/10.1016/S0168-9002\(03\)00541-2](https://doi.org/10.1016/S0168-9002(03)00541-2)
31. 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>
32. Z.-J. Li, M.-K. Yuan, Y.-X. Song et al., Visualization for physics analysis improvement and applications in BESIII. *Front. Phys. (Beijing)* **19**, 64201 (2024). <https://doi.org/10.1007/s11467-024-1422-7>
33. Z.-Y. You, K.-J. Li, Y.-M. Zhang et al., A ROOT based event display software for JUNO. *J. Instrum.* **13**, T02002 (2018). <https://doi.org/10.1088/1748-0221/13/02/T02002>
34. 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>
35. Mu2e Collaboration, Mu2e technical design report. arXiv:1501.05241 (2014) <https://doi.org/10.48550/arXiv.1501.05241>
36. Unity Technologies, Standard shader. <https://docs.unity3d.com/2023.2/Documentation/Manual/shader-StandardShader.html>. Accessed on 15 Mar 2025
37. M. Aros, C.L. Tyger, B.S. Chaparro, Unraveling the meta quest 3: An out-of-box experience of the future of mixed reality headsets, in: *HCI International 2024 Posters*, Springer Nature Switzerland, Cham, 2024, 3–8 (2024). https://doi.org/10.1007/978-3-031-61950-2_1
38. HTC Corporation, HTC vive official website. <https://www.vive.com>. Accessed on 15 Mar 2025
39. V. Corporation, Valve index official website. <https://www.valvesoftware.com/en/index>. Accessed on 15 Mar 2025
40. R.-Z. Cheng, N. Wu, M. Varvello et al., A first look at immersive telepresence on apple vision pro, in *Proceedings of the 2024 ACM on Internet Measurement Conference, IMC '24*, Association for Computing Machinery, New York, NY, USA, 2024, pp. 555–562 (2024). <https://doi.org/10.1145/3646547.3689006>
41. Unity Technologies, Unity user manual. <https://docs.unity3d.com/Manual/index.html>. Accessed on 15 Mar 2025
42. V. Corporation, Steam hardware & software survey. <https://store.steampowered.com/hwsurvey>. Accessed on 15 Mar 2025
43. G.H. Huang, Y.F. Wang, W.M. Luo et al., Improving the energy uniformity for large liquid scintillator detectors. *Nucl. Instrum. Meth. A* **1001**, 165287 (2021). <https://doi.org/10.1016/j.nima.2021.165287>
44. Z.Y. Li, Y.M. Zhang, G.F. Cao et al., Event vertex and time reconstruction in large-volume liquid scintillator detectors. *Nucl. Sci. Tech.* **32**, 49 (2021). <https://doi.org/10.1007/s41365-021-00885-z>
45. Z. Qian, V. Belavin, V. Bokov et al., Vertex and energy reconstruction in JUNO with machine learning methods. *Nucl. Instrum. Meth. A* **1010**, 165527 (2021). <https://doi.org/10.1016/j.nima.2021.165527>
46. Z.-Y. Li, Z. Qian, J.-H. He et al., Improvement of machine learning-based vertex reconstruction for large liquid scintillator detectors with multiple types of PMTs. *Nucl. Sci. Tech.* **33**, 93 (2022). <https://doi.org/10.1007/s41365-022-01078-y>
47. JUNO Collaboration, The design and technology development of the JUNO central detector. *Eur. Phys. J. Plus* **139**, 1128 (2024). <https://doi.org/10.1140/epjp/s13360-024-05830-8>
48. T. Lin, Y.X. Hu, M. Yu et al., Simulation software of the JUNO experiment. *Eur. Phys. J. C* **83**, 382 (2023). <https://doi.org/10.1140/epjc/s10052-023-11514-x>
49. Z. Deng, Status of JUNO simulation software. *EPJ Web Conf.* **245**, 02022 (2020). <https://doi.org/10.1051/epjconf/202024502022>
50. Autodesk, Fbx webpage. <https://www.autodesk.com/products/fbx/overview>. Accessed on 15 Mar 2025
51. C.-X. Wu, Z.-Y. You, Detector identifier and geometry management system in JUNO experiment. *PoS ICHEP2024*, 1049 (2025). <https://doi.org/10.22323/1.476.1049>
52. R. Chytracek, J. McCormick, W. Pokorski et al., Geometry description markup language for physics simulation and analysis applications. *IEEE Trans. Nucl. Sci.* **53**, 2892 (2006). <https://doi.org/10.1109/TNS.2006.881062>
53. A. Iusupova, S. Nemnyugin, Geometry import into virtual reality visualization engine for HEP experiments at BM@N. *Nucl. Instrum. Meth. A* **1067**, 169619 (2024). <https://doi.org/10.1016/j.nima.2024.169619>
54. T. Li, X. Xia, X.-T. Huang et al., Design and development of JUNO event data model. *Chin. Phys. C* **41**, 066201 (2017). <https://doi.org/10.1088/1674-1137/41/6/066201>
55. JUNO Collaboration, Modern software development for JUNO offline software. *EPJ Web Conf.* **295**, 05015 (2024). <https://doi.org/10.1051/epjconf/202429505015>
56. 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>
57. 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>
58. PHENIX Collaboration, PHENIX detector overview. *Nucl. Instrum. Meth. A* **499**, 469–479 (2003). [https://doi.org/10.1016/S0168-9002\(02\)01950-2](https://doi.org/10.1016/S0168-9002(02)01950-2)
59. LHCb Collaboration, The LHCb detector at the LHC. *J. Instrum.* **3**, S08005 (2008). <https://doi.org/10.1088/1748-0221/3/08/S08005>
60. T. Bray, J. Paoli, C. Sperberg-McQueen, Extensible markup language (xml) 1.0. <http://www.w3.org/XML/1998/06/xmlspec-report-19980910.htm>. Accessed on 15 Mar 2025

61. K. Group, COLLADA document schema and reference (Version 1.5). <https://www.khronos.org/collada/>. Accessed on 15 Mar (2025)
62. Autodesk, Drawing exchange format (DXF) reference. <https://archive.ph/20121206003818>. Accessed on 15 Mar 2025
63. M. Reddy, Wavefront OBJ file format. <http://www.martinreddy.net/gfx/3d/OBJ.spec>. Accessed on 15 Mar 2025
64. F. Developers, Freecad webpage. <https://www.freecadweb.org>. Accessed on 15 Mar 2025
65. P. Software, Pixyz studio software. <https://www.pixyz-software.com/studio>. Accessed on 15 Mar 2025
66. S. Kemmerer, Step: the grand experience (1999). <https://doi.org/10.6028/NIST.SP.939>
67. T. Sakuma, T. McCauley, Detector and event visualization with sketchup at the CMS experiment. *J. Phys. Conf. Ser.* **513**, 022032 (2014). <https://doi.org/10.1088/1742-6596/513/2/022032>
68. P. Vogel, J.F. Beacom, Angular distribution of neutron inverse beta decay, $\bar{\nu}_e + \bar{p}e^+ + n$. *Phys. Rev. D* **60**, 053003 (1999). <https://doi.org/10.1103/PhysRevD.60.053003>
69. T. Lin, W.-Q. Yin, Offline data processing in the first JUNO data challenge. arXiv:2408.00959 (2024). <https://doi.org/10.48550/arXiv.2408.00959>
70. JUNO Collaboration, The JUNO experiment top tracker. *Nucl. Instrum. Meth. A* **1057**, 168680 (2023). <https://doi.org/10.1016/j.nima.2023.168680>
71. M. Yu, W.-J. Wu, Y.-Y. Ding et al., A Monte Carlo method for Rayleigh scattering in liquid detectors. *Rev. Sci. Instrum.* **93**, 113102 (2022). <https://doi.org/10.1063/5.0119224>
72. M. Yu, W.-J. Wu, N. Peng et al., Measurements of Rayleigh ratios in linear alkylbenzene. *Rev. Sci. Instrum.* **93**, 063106 (2022). <https://doi.org/10.1063/5.0091847>
73. K. Li, Z. You, Y. Zhang et al., GDML based geometry management system for offline software in JUNO. *Nucl. Instrum. Meth. A* **908**, 43–48 (2018). <https://doi.org/10.1016/j.nima.2018.08.008>
74. S. Zhang, J.-S. Li, Y.-J. Su et al., A method for sharing dynamic geometry information in studies on liquid-based detectors. *Nucl. Sci. Tech.* **32**, 21 (2021). <https://doi.org/10.1007/s41365-021-00852-8>
75. ATLAS Collaboration, ATLAS offline data quality monitoring. *J. Phys. Conf. Ser.* **219**, 042018 (2010). <https://doi.org/10.1088/1742-6596/219/4/042018>
76. CMS Collaboration, CMS data quality monitoring: Systems and experiences. *J. Phys. Conf. Ser.* **219**, 072020 (2010). <https://doi.org/10.1088/1742-6596/219/7/072020>
77. J.-F. Hu, Y.-H. Zheng, X.-D. Sun et al., A data quality monitoring software framework for the BESIII experiment. *Chin. Phys. C* **36**, 62–66 (2012). <https://doi.org/10.1088/1674-1137/36/1/010>
78. Daya Bay Collaboration, Onsite data processing and monitoring for the Daya Bay Experiment. *Chin. Phys. C* **38**, 086001 (2014). <https://doi.org/10.1088/1674-1137/38/8/086001>

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.