Fast muon simulation in the JUNO experiment with neural networks

Wenxing Fang[1,*], *Weidong* Li[1], and *Tao* Lin[1]

[1]Institute of High Energy Physics, Beijing 100049, People's Republic of China

Abstract. The Jiangmen Underground Neutrino Observatory (JUNO) experiment is set to begin data taking in 2024 with the aim of determining the neutrino mass ordering (NMO) to a significance of 3 σ after 6 years of data taking. Achieving this goal requires effective background suppression, with the background induced by cosmic-ray muons being one of the most significant sources of interference in the NMO study. Accurately simulating the cosmic-ray muon background is crucial for the success of the experiment, but the sheer number of optical photons produced by the muon makes this detector simulation process extremely time-consuming using traditional methods such as Geant4. This paper presents a fast muon simulation method that employs neural networks to expedite the simulation process. Our approach achieves an order-of-magnitude speed-up in simulation time compared to Geant4, while still producing accurate results.

1 Introduction

The JUNO experiment is a multipurpose neutrino experiment [1–3] with the main goal of determining the neutrino mass ordering (NMO) using the inverse beta decay process (IBD). In this process, an anti-neutrino from the reactor interacts with a proton, producing a positron and a neutron. The positron quickly annihilates with an electron, producing two 511 keV gamma rays, while the neutron travels a certain distance in the liquid scintillator (LS) before being captured by a hydrogen nucleus and producing a 2.2 MeV gamma ray. A schematic of the IBD process is shown on the left-hand side of Figure 1. Other future neutrino experiments, such as Hyper-K [4] and DUNE [5], also aim to determine the NMO. However, the JUNO experiment aims to achieve this goal with a 3 σ significance after 6 years of data-taking, which is expected to be around 2030.

The careful design of the JUNO detector is crucial for achieving the experiment's goal of determining the NMO. The overall layout of the detector is shown in the right-hand side of Figure 1. At the top is the top tracker (TT), which is used to detect and reconstruct cosmic-ray muons coming from the top. With the help of the TT, muon-induced IBD-like events can be efficiently removed from the data. Below the TT is the water pool Cherenkov detector, which can detect and reconstruct cosmic-ray muons from other directions and also protect the central detector (CD) from natural radioactivity in the surrounding rocks. However, the most critical part of the detector is the CD itself, which is fully filled with a 20-thousand-ton liquid scintillator. The CD is used to reconstruct the energy and position of particles, as well

*e-mail: fangwx@ihep.ac.cn

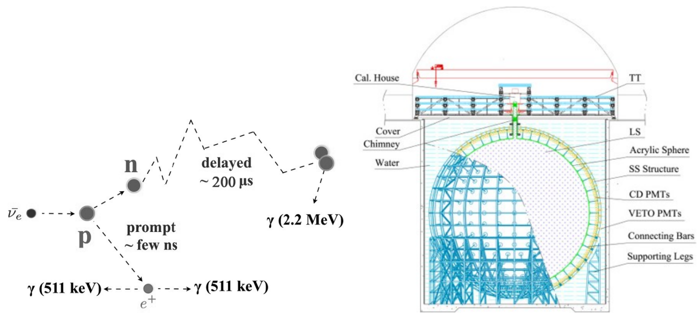

Figure 1. Schematic view of the IBD process (left) and the JUNO detector (right).

as perform particle identification. The CD is surrounded by a large number of photomultiplier tubes (PMTs), including ~18,000 20-inch large PMTs and ~25,000 3-inch small PMTs, which are used to detect the optical photons produced in the LS. Together, these PMTs can detect the faint light signals produced by particles interacting in the LS, which is critical for reconstructing the events accurately.

Given that IBD events are rare, it is crucial to minimize the background events as much as possible to study the NMO. After optimizing the selection criteria through a series of cuts, the IBD signal event rate is 60 per day, while the background event rate is 3.8 per day [3]. However, even with these optimized cuts, residual backgrounds can still be present, particularly those arising from the ^{9}Li/^{8}He nucleus produced by the interaction between cosmic-ray muons and the LS. To study muon-induced backgrounds in detail, a full detector simulation using Geant4 [6] is typically necessary. However, when a cosmic-ray muon (with an average energy of 215 GeV) passes through the CD, it produces millions of optical photons, and a Geant4 simulation can take several hours to complete. Therefore, it is crucial to develop fast simulation methods to speed up the detector simulation process and enable efficient background studies.

This paper is structured as follows: Section 2 will give a brief introduction to previous related work. Section 3 will describe the fast muon simulation using neural networks which only focus on the detector simulation stage, electronic simulation is not included. The results are presented in Section 4. A summary is given in Section 5.

2 Related work

In previous work [7], the voxel method was developed for fast muon simulation in the JUNO detector. The basic idea of the voxel method is to simulate the response of each PMT to the particle's energy deposition in a small voxel. An illustration of the voxel method can be seen in the left-hand plot of Figure 2. For each voxel, the method simulates the number of photon electrons (N p.e.) detected by each PMT, as well as the corresponding hit time. The voxel method assumes that the JUNO CD detector has spherical symmetry, such that the distribution of N p.e. and hit time for each PMT depends on the radius (R) of the voxel and the angle (θ) between the voxel and the PMT. Before performing the voxel simulation, the N p.e. and hit time distributions need to be saved in ROOT histograms according to different R and θ bins, using Geant4 simulated samples. During the voxel simulation, for each voxel with energy deposition, the method samples the N p.e. for each PMT according to the R and θ, and then samples the corresponding hit times from the histograms. The voxel method offers

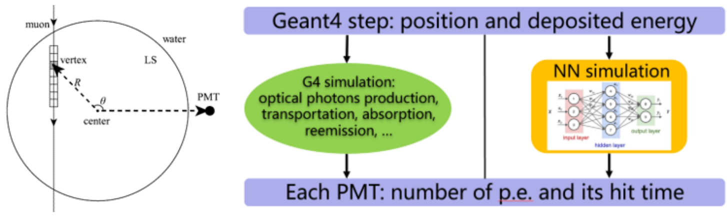

Figure 2. Schematic view of the voxel method (left) and the ML method (right).

significant advantages in terms of speed, as it avoids the need to simulate the huge number of optical photons produced by a muon passing through the detector. However, the method also has limitations. For instance, it requires a large amount of memory to store the histograms needed for N p.e. and hit time sampling. Additionally, the JUNO detector has a stainless steel support structure that breaks the spherical symmetry of the CD, meaning that more factors need to be considered in the voxel simulation to account for this effect.

3 Methodology

3.1 The neural network model

The use of neural networks for fast muon simulation is based on a similar idea to the voxel method described in Section 2. Instead of relying on Geant4 for simulating optical photons, neural networks are employed to simulate the number of p.e. detected by each PMT, as well as their corresponding hit times. The right-hand plot of Figure 2 shows a schematic comparison between the traditional full Geant4 simulation and the simulation using neural networks. The main difference between the two approaches is in the optical photon simulation part, which is the most time-consuming component. Otherwise, the two methods are similar in other respects.

To simulate the number of p.e. and hit times detected by each PMT using neural networks, two separate neural networks are trained for each type of large PMT (NNVT, HighQENNVT, and Hamamatsu). One neural network is used for simulating the number of p.e., while the other is used for simulating the corresponding hit times. The detailed structure of these two neural networks can be seen in Figure 3. Each neural network consists of a fully connected architecture, with input, hidden, and output layers. The ReLU activation function is used after each hidden layer to introduce nonlinearity into the network.

For the N p.e. simulation neural network, the inputs consist of the position of the energy deposition ($x_{edep}, y_{edep}, z_{edep}$), the position of the PMT ($x_{PMT}, y_{PMT}, z_{PMT}$), and the photon detection efficiency (PDE_{PMT}) of the PMT. Since the distribution of N p.e. follows the Poisson distribution, it has only one free parameter, which is the expectation value. Therefore, the output of the neural network represents the expected N p.e. value for a 1 MeV visible energy deposition. To perform the N p.e. simulation, one first obtains the expectation value from the neural network. This expectation value is then multiplied by the visible energy deposition, giving a corrected expectation value λ_{cor}. Finally, an N p.e. sampling is performed from a Poisson distribution whose expectation value is equal to λ_{cor}.

After performing the N p.e. simulation, a hit time simulation neural network is applied to simulate the hit time of each N p.e. detected by each PMT. The inputs of the hit time simulation network include the radius (R) and θ, which have the same definitions as described

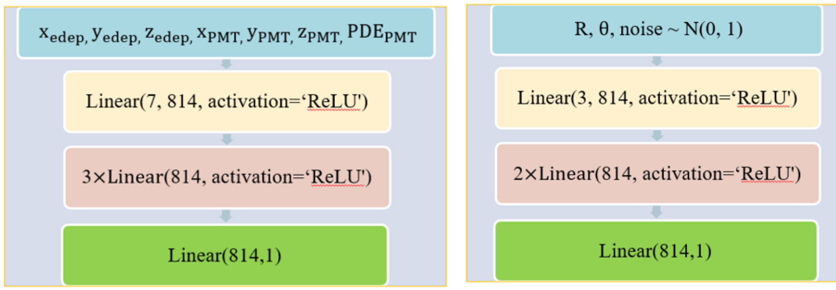

Figure 3. The structure of N p.e. (left) and hit time (right) simulation neural networks.

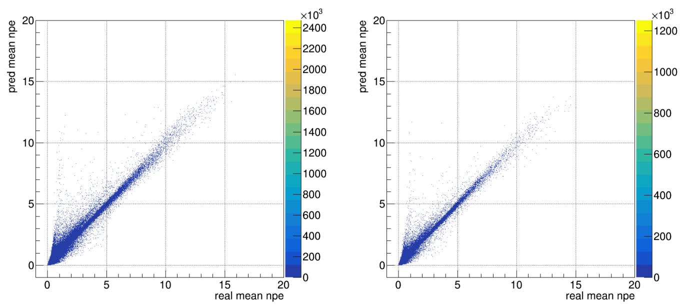

Figure 4. The 2D scatter plot shows the target (or real) expected N p.e. (in the X-axis) versus the predicted value (in the Y-axis) for HighQENNVT PMTs (left) and Hamamatsu PMTs (right)

in Section 2. In addition, random noise from a standard normal distribution is also inputted to simulate the stochastic fluctuation effect. The output of the hit time simulation network is the simulated hit time for each N p.e. detected by each PMT.

3.2 Training and performance

To train the N p.e. and hit time simulation neural networks for each type of PMT (NNVT, HighQENNVT, and Hamamatsu), laser events with 1 MeV visible energy deposited in various positions are generated. The mean absolute error (MAE) is used as the loss function for the N p.e. simulation network training, while the differentiable two-sample test statistics [8] is used as the loss function for the hit time simulation network training. The Adam optimizer [9] is used for both network training processes.

Figure 4 shows the performance of the N p.e. simulation neural networks, and overall, the predicted values agree well with the target (or real) values. However, there is still room for improvement in the region where the expected N p.e. is close to zero. Figure 5 provides examples of the performance of the hit time simulation neural networks, and in general, the simulated hit times from the neural networks show good agreement with those from the Geant4 simulation. This indicates that the hit time simulation neural networks are capable of accurately capturing the complex photon propagation and detection processes in the PMTs.

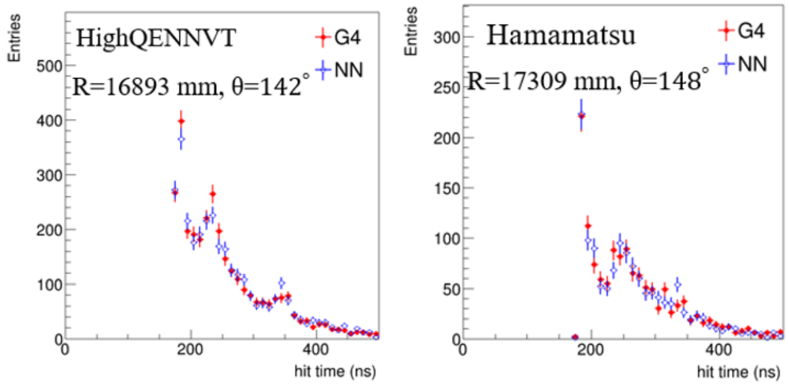

Figure 5. The hit time distribution comparison between Geant4-based (red) and neural network-based simulation (blue).

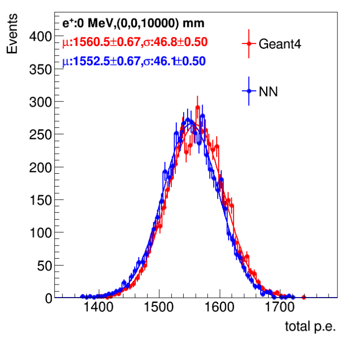

Figure 6. The comparison of total p.e. distribution between Geant4 (red) and neural network simulation (blue) for 0 MeV e^{+} at Z axis with z position equal 10 meters.

4 Results

This section presents performance checks at the event level for different types of particles. The first check involves comparing the total p.e. distributions obtained from Geant4 simulation and neural network simulation. An example is shown in Figure 6, which compares the total p.e. distributions for 0 MeV e^{+} at the Z axis, with the z position equal to 10 meters. Each distribution is fitted with a Gaussian function, and the mean and σ can be obtained from the fit. Figure 7 shows a comparison of the fitted mean and resolution (i.e., the fitted σ divided by the fitted mean) between the Geant4 simulation and neural network simulation for different particles in different positions. As can be seen, the Geant4 simulation and neural network simulation exhibit good agreement, with similar means and resolutions across all particle types and positions.

To check the performance of the hit time simulation, Figure 8 compares the hit time and first hit time distributions obtained from the Geant4 simulation and neural network simulation

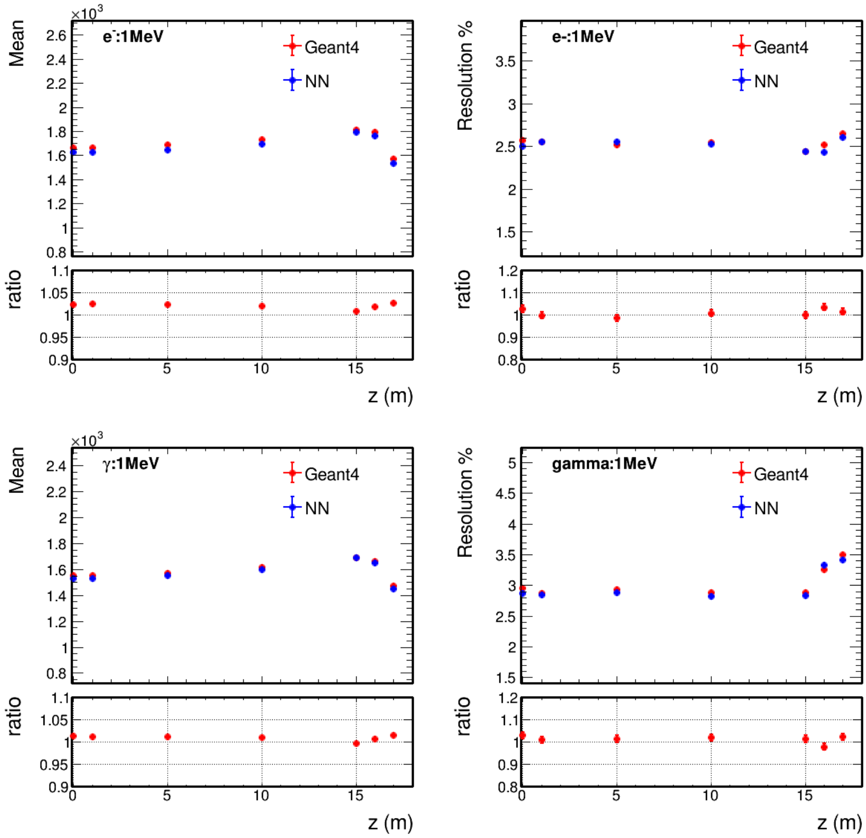

Figure 7. The comparison of the mean (left) and resolution (right) of total p.e. between Geant4 and neural network simulation for 1 MeV e^- (top) and 1 MeV γ (bottom) with different positions at Z axis.

for different particle types. Overall, the two simulation methods show good agreement, indicating that the neural network simulation is able to accurately capture the hit time distribution for different types of particles. However, in regions where hit times (or first hit times) exceed 200 ns, further improvement in agreement is still required. It is possible that incorporating additional training samples could be beneficial in addressing this issue.

To evaluate the speedup achieved by the neural network simulation compared to the traditional Geant4 simulation, Figure 9 shows a comparison of the simulation time for muon events with the same start position (0,0,19 m) and direction (0,0,-1) along the negative Z axis. The neural network simulation is performed on a single GPU (V100) card. As can be seen from Figure 9, the neural network simulation achieves a speedup of more than one order of magnitude over the Geant4 simulation for a wide range of muon energies.

5 Summary

Simulating cosmic-ray muons is an essential aspect of the JUNO experiment, but traditional muon simulation methods based on Geant4 can be extremely time-consuming. This paper proposes a fast and efficient method for simulating cosmic-ray muons using neural networks.

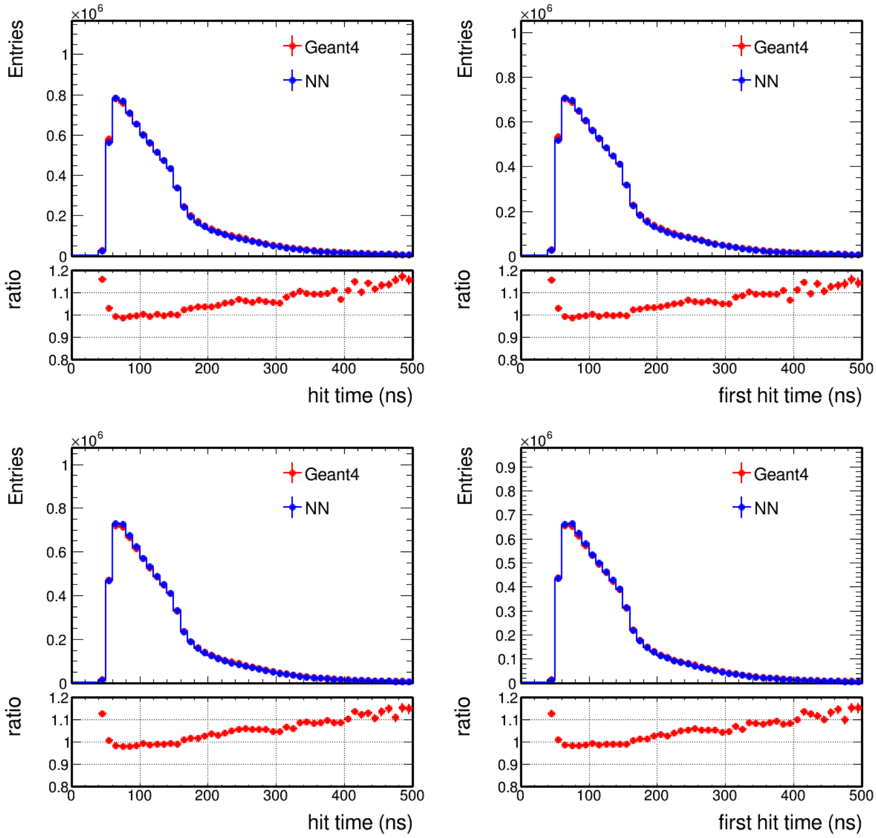

Figure 8. The distribution of the hit time (left) and first hit time (right) of PMTs for 1 MeV e^{-} (top) and 1 MeV γ (bottom) at Z axis with z position equal 10 meters.

The basic idea of this method is to use neural networks for simulating the optical photon detection process, which is the most time-consuming part of the Geant4 simulation. The results of the neural network simulation show good agreement with those obtained from the Geant4 simulation and more than one magnitude speed-up can be achieved for the muon simulation.

Acknowledgments

This work is supported by the Innovation Project of the Institute of High Energy Physics under Grant No.E15455U210, CAS Center for Excellence in Particle Physics.

References

[1] E. Ciuffoli, J. Evslin, X. Zhang, Physical Review D **88** (2013)

[2] Y.F. Li, J. Cao, Y. Wang, L. Zhan, Physical Review D **88** (2013)

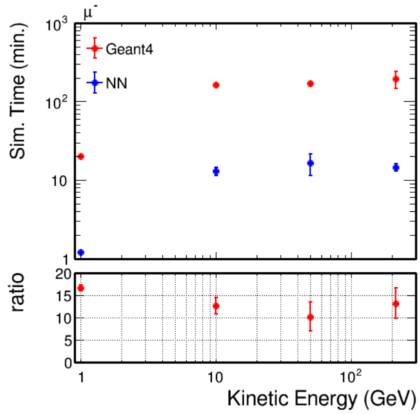

Figure 9. Muon simulation time comparison between Geant4 simulation and neural network simulation versus the energy of muon.

[3] F. An et al., Journal of Physics G: Nuclear and Particle Physics **43**, 030401 (2016)

[4] Y. Itow, PoS **ICRC2021**, 1192 (2021)

[5] R. Acciarri et al., *Long-baseline neutrino facility (lbnf) and deep underground neutrino experiment (dune) conceptual design report volume 1: The lbnf and dune projects* (2016), `1601.05471`

[6] *Geant4, a simulation toolkit*, https://geant4.web.cern.ch/node/1

[7] T. Lin, Z.Y. Deng, W.D. Li, G.F. Cao, Z.Y. You, X.Y. Li, Chinese Physics C **40**, 086201 (2016)

[8] J. Djolonga, A. Krause, *Learning Implicit Generative Models Using Differentiable Graph Tests* (2017), `1709.01006`

[9] D.P. Kingma, J. Ba, *Adam: A method for stochastic optimization* (2017), `1412.6980`