Muon reconstruction with a convolutional neural network in the JUNO detector

Yan Liu[1,2] · Wei-Dong Li[1,2] · Tao Lin[2] · Wen-Xing Fang[2] · Simon C. Blyth[2] · Ji-Lei Xu[2] · Miao He[2] · Kun Zhang[2]

Abstract

Purpose The Jiangmen Underground Neutrino Observatory (JUNO) is designed to determine the neutrino mass ordering and measure neutrino oscillation parameters. A precise muon reconstruction is crucial to reduce one of the major backgrounds induced by cosmic muons.
Methods This article proposes a novel muon reconstruction method based on convolutional neural network (CNN) models. In this method, the track information reconstructed by the top tracker is used for network training. The training dataset is augmented by applying a rotation to muon tracks to compensate for the limited angular coverage of the top tracker.
Result The muon reconstruction with the CNN model can produce unbiased tracks with performance that spatial resolution is better than 10 cm and angular resolution is better than 0.6 °. By using a GPU-accelerated implementation, a speedup factor of 100 compared to existing CPU techniques has been demonstrated.

Keywords JUNO · Muon reconstruction · Convolutional neural networks · GPU

Introduction

Jiangmen Underground Neutrino Observatory (JUNO) [1] is a liquid scintillator detector that is designed to determine the neutrino mass ordering and to precisely measure neutrino oscillation parameters. Its location 53 km away from both Yangjiang and Taishan nuclear power plants in Southern China is chosen to optimize the oscillation measurement sensitivity. A schematic view of the JUNO detector is shown in Fig. 1. The innermost central detector (CD) is 20 kiloton of liquid scintillator contained by an acrylic sphere and instrumented with ~ 18,000 20-inch and 25,600 3-inch photomultiplier tubes (PMTs). The CD is submerged in a water pool which is instrumented with a further 2400 PMTs forming the water Cherenkov detector (WCD). The top tracker (TT), located above the CD and WCD, is a 3-layer muon

tracker detector with size 47 m × 20 m × 3 m providing precise atmospheric muon tracking over about 60% of the water pool. Ionizing particles cause excitations within the scintillator which subsequently de-excite resulting in the emission of scintillation photons. These photons propagate through the detector undergoing scattering, absorption and re-emission as well as refraction and reflection at material boundaries before some of them reach the PMTs. The charge and time information measured by the PMTs enables the originating vertex position and energy to be reconstructed.

The primary background for neutrino detection arises from ^{9}Li/^{8}He induced by muon spallation in the liquid scintillator or muon shower particles. This background can be reduced by excluding a veto volume along the muon track within a time window surrounding the time at which the muon passes through the detector. A precise muon reconstruction directly allows the veto volume and time windows to be reduced which increases the livetime achievable by the JUNO detector.

The vital importance of muon reconstruction for JUNO has led to the development of several muon reconstruction methods including:
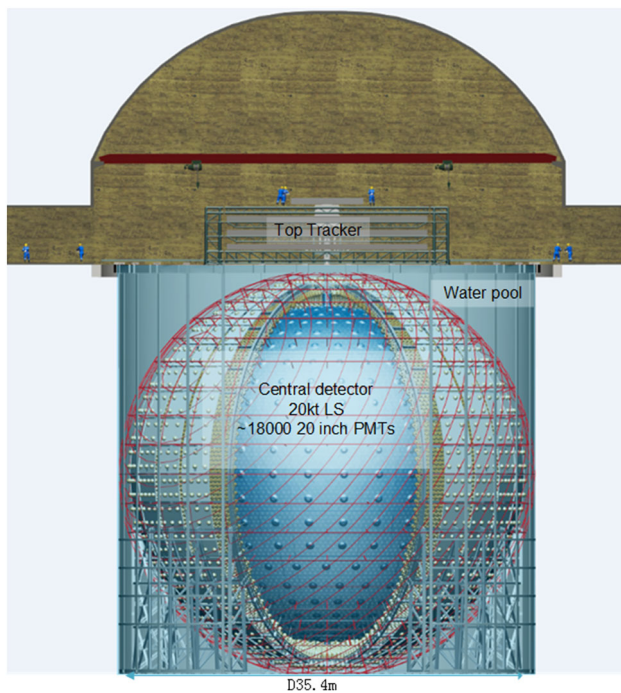
– Muon reconstruction with a geometrical model [2] which is based on the geometrical shape of the first light and

✉ Tao Lin
lintao@ihep.ac.cn

1 University of Chinese Academy of Sciences, Beijing 100049, China

2 Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China

Fig. 1 A schematic view of the JUNO detector

provides a spatial resolution of 20 cm and an angular resolution of 1.6 ° over the whole detector.

– Muon tracking with the fastest light [3] which uses the least squares fit to the PMT first hit time (FHT) measurements. After correction of the FHT for each PMT, the reconstruction yields spatial resolution less than 3 cm and angular resolution of less than 0.4 °.

This article reports a novel alternative technique for muon reconstruction that applies deep learning techniques and benefits from GPU acceleration. This technique has the advantages of speed and avoids the need to develop detailed optical models or to interpret calibration data. All studies have been based on JUNO Monte Carlo simulation data.

Treating the readout of the whole detector as an image composed of pixel values from PMT charge and time measurements allows image classification techniques to be applied to track finding and fitting. For image processing tasks, convolutional neural networks (CNN) have been widely used in many areas such as image classification [4], object detection [5] and image segmentation [6]. Within high energy physics, image processing with CNN has been used in jet classification [7,8] and event classification [9,10]. In this paper, the CNN is adopted to perform muon track reconstruction in the CD detector.

This paper is organized into a description of how the training and testing datasets were prepared in section 2 followed by an explanation of the chosen deep neural network architecture and training procedure in section 3. Performance metrics

of the muon reconstruction are presented in section 4, and a discussion of possible challenges for the application of this technique to experimental data is given in section 5, followed by the conclusions of this study summarized in section 6.

Dataset and muon reconstruction model

Simulation of single muon tracks

In this study, all Monte Carlo (MC) simulated muon events used for network training and testing are produced by the official JUNO simulation framework [11,12] which is based on the Geant4 toolkit [13]. The underground muon sample before input into detector is generated by the muon transportation software MUSIC [14] with JUNO mountain profile and the parameterized Gaisser formula of muon [15] at the mountain surface. The energy range of cosmic muon reaching the JUNO detector is from 0.1 GeV to 10 TeV in simulation, and the average energy is about 207 GeV. The flux is about 4 $\times 10^{-3}$ Hz / m^2. There are about 90% of muon events with single track and about 10% with multiple tracks. Only the events with 200 GeV muon track are selected for this study, so that the muons will go through the whole detector and exit at the bottom.

Along the muon trajectory through the materials of the detector geometry, the Geant4 toolkit simulates physics processes including ionization and nuclear interactions. Muons passing through the scintillator result in excitations of the molecules of the scintillating medium which subsequently de-excite producing photons. In addition, as muons travel through both the water pool and the scintillator, their charged nature results in the generation of photons from Cherenkov radiation. Both scintillation photons and Cherenkov photons are propagated through the detector taking into account optical processes such as scattering, absorption and re-emission. A fraction of the propagated photons reach PMTs and eject a photo-electron from the photocathode resulting in electronic signals for the charge and arrival time. PMT characteristics including the quantum efficiency and collection efficiency are measured and used as inputs to the simulation.

Figure 2 illustrates a muon track passing through the liquid scintillator (LS), together with the positions of PMTs that detect optical photons emitted along the muon trajectory.

Comparisons of simulated and real experimental data often reveal some differences. This presents a problem as neural networks trained with simulated samples may learn from features that differ from those present in the real data causing the model to be invalid. As real JUNO experimental data are not yet available, a comparison between independent aspects of the simulated data is used as a cross-check. The muon track parameters obtained from the top tracker (TT) reconstruction [1] are used as an independent alterna-
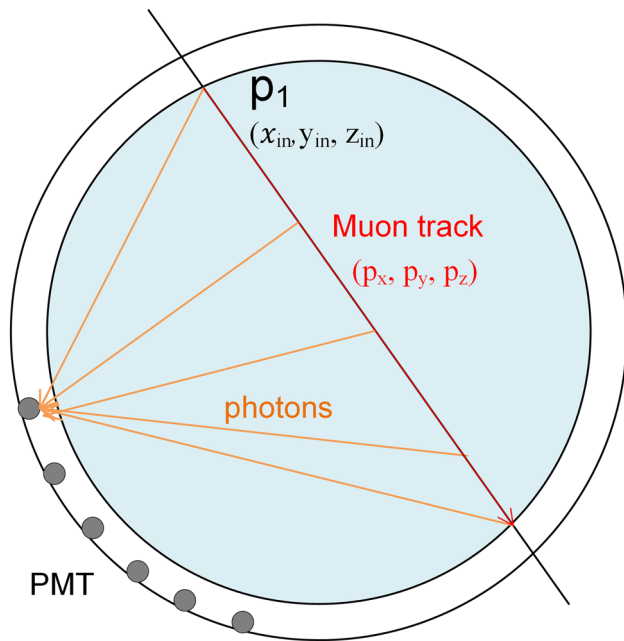
Fig. 2 Schematic of muon track, optical photon paths and PMT positions. The shaded area represents the LS

tive for training the neural network and compared with results obtained by training based on the MC truth information.

Modeling of muon reconstruction

Muon reconstruction uses the first light arrival times and number of photoelectrons detected by all the PMTs as well as the positions of the PMTs. The muon track is modeled using its direction vector (p_x, p_y, p_z) and the intersection position of the track with surface of the LS sphere $\mathbf{r}_{in} = (x_{in}, y_{in}, z_{in})$ using a coordinate system which origin at the center of the LS sphere.

$$\mathbf{track} = (x_{in}, y_{in}, z_{in}, p_x, p_y, p_z). \tag{1}$$

Muon reconstruction is the estimation of the 6 independent track parameters $\mathbf{track}^{rec}$ from PMT measurements of the numbers of photoelectrons ($\mathbf{Q}$), fastest light arrival times ($\mathbf{T}$) as well as the positions of all the PMTs ($\mathbf{R}$).

$$\mathbf{track}^{rec} = f(\mathbf{Q}, \mathbf{T}, \mathbf{R}). \tag{2}$$

Reconstruction performance is assessed using the angle α between the true track and the reconstructed track and the difference between the distance to CD center ΔD between the true and reconstructed track.

Table 1 TT reconstruction performance for a sequence of selection criteria

Selection	1	2	3	4
Events	145K	130K	46K	40K
N_{point} Cuts	–	>2	>2	>2
χ^2 cuts	–	–	<0.2	<0.2
$N_{channel}$ Cuts	–	–	–	<8
α mean (°)	0.30	0.23	0.21	0.20
α RMS (°)	1.28	0.58	0.30	0.16
ΔD mean (mm)	1.1	0.8	0.6	0.3
ΔD RMS (mm)	321	157	91	75

Muon tracking and selection with top-tracker

Comparisons of TT reconstructed muon tracks with MC truth tracks show that it is necessary to apply track quality criteria to select well reconstructed muons. Table 1 shows the reconstruction performance for a series of selection criteria on the number of the hit TT layers (N_{point}), the least squares value of TT reconstruction (χ^2) and the number of activated scintillator bars in the X, Y directions of the layer ($N_{channel}$).

The reconstruction performance is substantially improved within the high quality selection. Further quality criteria tested did not provide significant performance gains. The highest quality selection 4 was used for the network training with TT tracks.

Rotation method-based data augmentation

As the TT partially covers the water pool, it is only able to reconstruct muons over a limited solid angle region. Figure 3 shows the distributions of CD sphere injection points for reconstructed TT muons in the upper plot and for the MC truth tracks in the lower plot. A rotation method has been developed that effectively extends the angular range of the training sample using a rotational symmetry assumption.

The rotation method is illustrated in Fig. 4 showing how an event is rotated and a new event is generated based on the reconstructed tracks. A randomized rotation matrix of each event is defined considering a new injection point and rotation angle ω of the track around the vector $\mathbf{v}$. The track and the positions of PMTs are rotated together using the same rotation matrix. After the randomized rotation, a rejection sampling is applied to get a new position that conforms to the expectation of the distribution.

For muon events reconstructed by TT, after using the rotation method, the distribution of the injection point is shown in Fig. 5. Comparing Fig. 3b and Fig. 5, it can be found that the injection point distribution after rotation is similar to the one from cosmic muon in CD which basically extends the ranges of injection points.

(a) Injection points distribution of TT reconstruction

(b) Injection points distribution of MC truth tracks
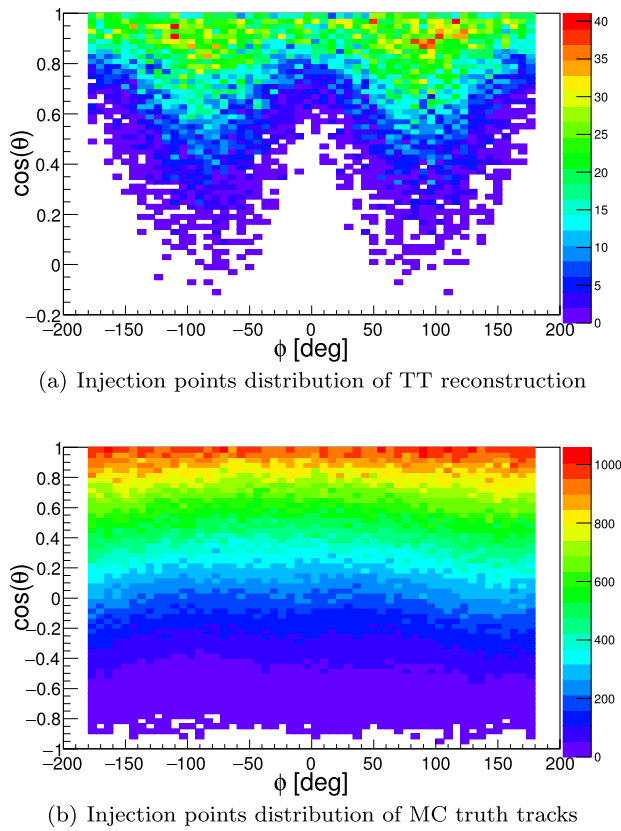
Fig. 3 The two-dimensional distribution of $(\cos\theta, \phi)$, where (θ, ϕ) are the injection positions at the surface of the LS sphere in spherical coordinates. The injection positions are at the north pole of LS sphere when $\cos(\theta) = 1$ ($\theta = 0$ deg). The distribution modulated with and without TT geometry and reconstruction is shown in (**a**) and (**b**), respectively. In **a**, the non-uniform distribution along ϕ is influenced by the TT geometry. In **b**, the distribution is not uniform along ϕ, because the underground muon distribution was modulated by the mountain profile above

Dataset composition

Application of CNN models to muon reconstruction requires the fixed PMT positions and measurements of PMT charge and times to be converted into image arrays. All PMT positions (x, y, z) are converted into two-dimensional spherical coordinates (θ, ϕ) using the equidistant cylindrical projection.

Example images of the charge Q and time T channels at the PMT pixel positions are shown in Fig. 6. The charge channel has two clusters corresponding to the muon entry and exit points at the CD sphere surface and the time T channel has a valley of low values in the region of the muon entry point. These two channels are combined together in an image pixel array, with each pixel having two channel values with the values of charge Q and time T, while the values for pixels that do not correspond to PMTs are set to zero. Then, the feature standardization [16] is applied to each channel of the image pixel array for the normalization of the image before
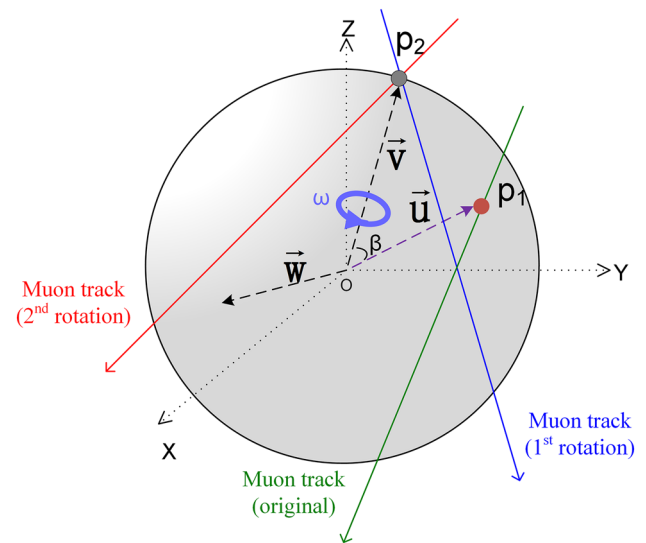
Fig. 4 Schematic diagram of the rotation method. The original muon is shown in green. First the track is rotated along the **w** with angle β, where $\mathbf{w} = \mathbf{u} \times \mathbf{v}$. After the first rotation, the track in blue is then rotated around **v** with angle ω. The final track is shown in red. The detector is also rotated using the same rotation matrix
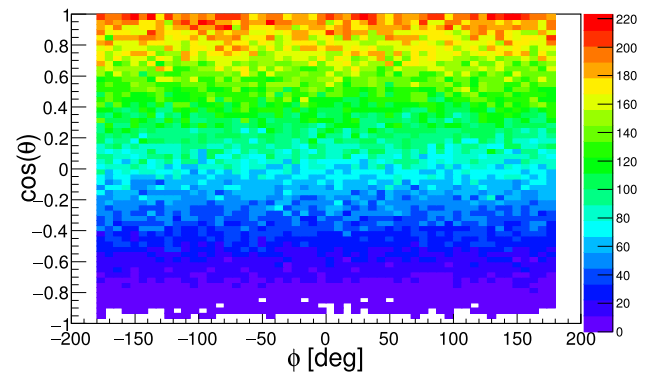
**Fig. 5** Injection point distribution after rotation

entering the neural network:

$$X' = (X - \mu)/\sigma \tag{3}$$

where X is the original data, X' is the normalized data, μ is the mean of X, and σ is the standard deviation of X.

In the training, the label of the direction vector (p_x, p_y, p_z) and intersection position vector (x_{in}, y_{in}, z_{in}) have been normalized to unit vectors, respectively.

The training and testing samples used in this study are shown in Table 2. The sample of directly simulated muons is split into training, validation and testing samples named "Training MC," "Validation MC" and "Testing MC" in the table. The training dataset obtained by TT reconstruction and rotation is called "training TT_rot."

(a) Image of channel Q

(b) Image of channel T
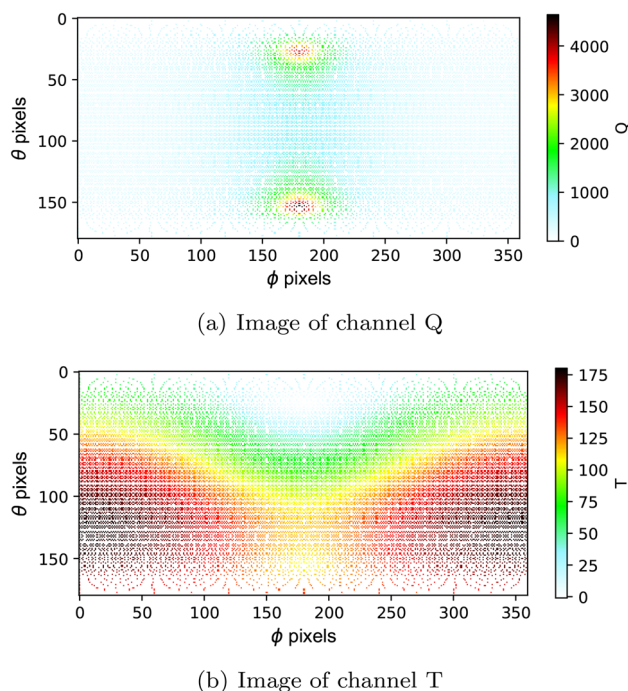
Fig. 6 Charge and time channel images for a vertical downward muon track with entry point at spherical coordinates (25,180) degrees on the CD sphere. The color scale from blue through to red represents increasing values

Table 2 The datasets used for the training, validation and testing

Dataset	Events	Image size	Label
Training MC	300K	$180 \times 360 \times 2$	MC truth
Training TT_rot	300K	$180 \times 360 \times 2$	Rotated TT rec
Validation MC	20K	$180 \times 360 \times 2$	MC truth
Testing MC	20K	$180 \times 360 \times 2$	MC truth

The first column is the dataset name. The second column is the number of events in each dataset. The third column is the size of each image. The last column indicates the datasets are generated from the full simulation or augmented from the rotated events after TT reconstruction

Network architecture

The use of artificial neural networks to solve complex problems has been explored since the 1940s. In recent years, with the dramatic increase in available computing power, it has become feasible to use computationally intensive neural networks with many inner layers. These neural networks with many computational layers are called deep neural networks (DNN), which can learn various potential features in large amounts of data. Convolutional neural network (CNN or ConvNet) is one form of deep neural network. CNN consists of one or more convolutional layers and a top fully connected layer, as well as associated weights and pooling layers. This structure makes convolutional neural networks suitable for

processing two-dimensional image data and extracting features from the data efficiently.

Model architecture

The network model architecture adopted in this study and illustrated in Fig. 7 is based on the VGGNet16 [17] configuration. It is composed of an input layer, convolutional layers, activation layers, pooling layers and fully connected layers. The input layer is the 2-channel (Q and T) image converted from the muon event data.

The principal elements in this model are blocks of three convolution layers with a 3×3 convolutional kernel followed by a pooling layer (max pooling) [18]. The convolutional layers are used to perform data feature extraction. The max pooling layer is used for signal downsampling, reducing the data size input to the next layer and also reducing the impact of local data on the result which makes the results more robust. As shown in the study [17], the use of three 3×3 convolutional layers has several advantages. Firstly, this architecture has the same large effective receptive field as a 7×7 convolutional layer, but decreases the number of learning parameters by almost a factor of two. Secondly, it can inject more nonlinearity operations between the two convolution layers which enhanced the ability to learn features.

The fully connected hidden layers convert the abstract feature signal into the output of the prediction task. After some tries, the number of neurons at each fully connected hidden layer is chosen as following: 1024, 512, 256. Compared with original VGGNet, this kind of setting not only gives better muon reconstruction performance but also reduces the number of parameters in fully connected hidden layers by 75%.

Finally, a fully connected output layer with 6 nodes corresponding to the muon track parameters is applied to obtain the predictions of the network. The six output values of the network are the predicted values of track entry position into CD (x_{in}, y_{in}, z_{in}) and track direction (p_x, p_y, p_z) of the muon.

Loss function

The loss function is an indicator used to measure the difference between the model prediction and the true values which is used for the optimization of the model. The loss function used in this study, shown in Eq. 4, consists of two terms. The first term represents the error between the true label and predicted label of the muon track. The second term provides a regularization penalty [19] which acts to avoid overfitting of the model and contains λ coefficient scale and a summation
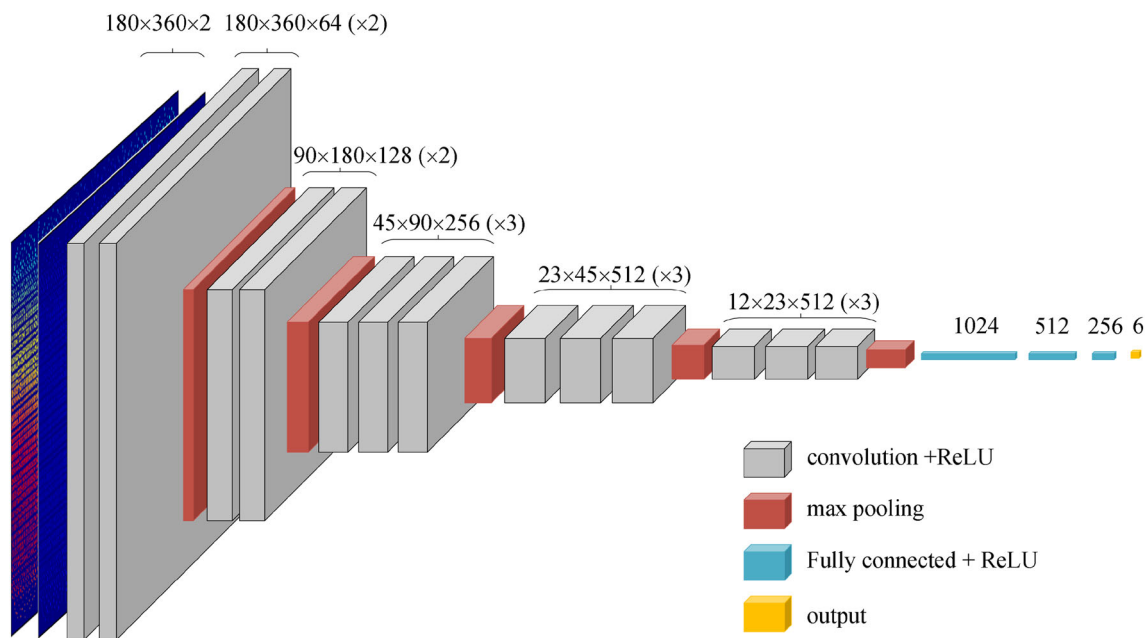
Fig. 7 Convolutional neural networks configuration adapted from VGGNet16, which is widely used to extract image features and image classification

over the weights of the model W_j.

$$Loss = \sum_i (||\mathbf{track}_i^{\text{predict}} - \mathbf{track}_i^{\text{truth}}||^2) + \sum_j \lambda ||W_j||^2 \tag{4}$$

The training process acts to reduce the loss function resulting in the predicted values approaching closer to the actual values. With muon reconstruction, this corresponds to improving the spatial and angular resolution of the reconstruction.

Implementation and training

The network used in this article is implemented using the TensorFlow [20] open-source framework and its Python interface. The hardware environment uses the NVIDIA Tesla V100 GPU. Network training uses 2 GPUs for parallel computing. Using Mini-Batch [21] training strategy, the size of batch is 256 (256 samples are trained at the same time in each iteration), which is divided into two smaller batches and computed in parallel on 2 GPUs. Two GPUs provide a speedup of 1.7 times, as compared to using a single GPU. On a system equipped with two NVIDIA Tesla V100 GPUs, training a single network took about 70 h depending on the architecture.

The training neural network uses the stochastic gradient descent method to calculate the gradient of the loss function in the weight parameter space and then advances the weight parameters by a certain step in the direction of the negative gradient. After reaching the set number of iterations or the set error accuracy, the weight parameter is considered to have reached the optimum position where the loss function is the smallest. The step size in the gradient descent process is the learning rate. After some studies, the following optimized learning rate schedule is used: the learning rate is initialized to be 0.1 at the beginning of training and is decreased by a factor of 10 after each 50,000 training steps.

The loss values from training dataset and validation dataset are monitored during the network training. It is found that after 200,000 training steps, the loss values from training dataset and validation dataset are almost stable which means the network has reached the optimized status. Besides, during the training the loss from validation dataset decreases smoothly which indicates the training process does not suffer from overfitting (a phenomenon that a model has overlearned features specific to the training dataset that are not present in general datasets, which is harmful to the model).

Performance study

Reconstruction resolution and efficiency

The network proposed in this paper is separately trained on dataset "Training TT_rot" and "Training MC" which is used for verifying the rotation method. After training, they are tested by dataset "Testing MC" for performance analysis. After trying some fitting functions for the distribution of α, it is found that the fitting effect is good using the improved
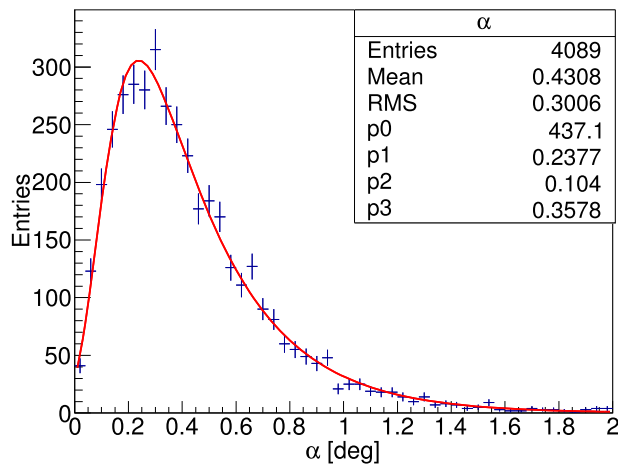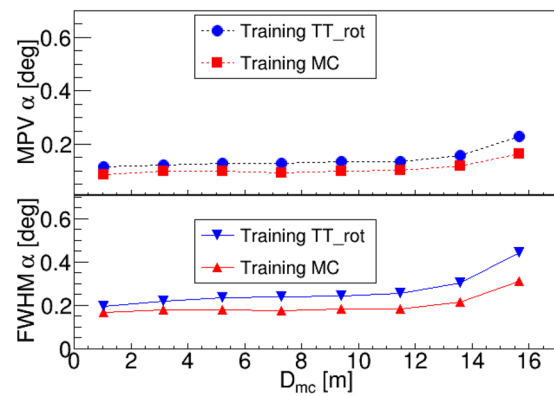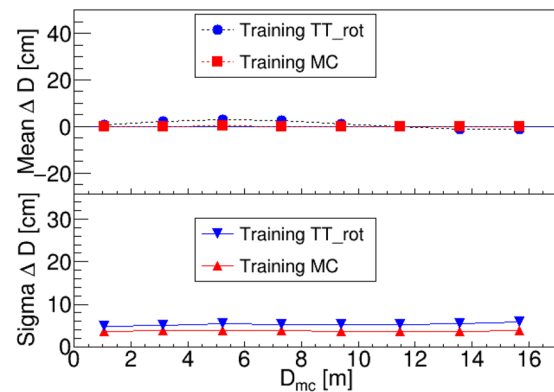
(a) α distribution fitted by improved Landau function
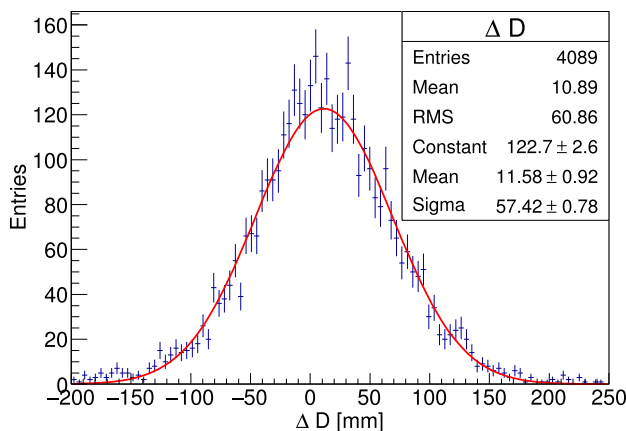
(b) ΔD distribution fitted by Gaussian function

Fig. 8 α and ΔD reconstructed from the events where the track is 14.7–16.7 m from the center of CD

(a) Most probable value(MPV) and full width at half maxima(FWHM) value of α versus D_{mc}

(b) Mean and sigma value of ΔD versus D_{mc}

Fig. 9 Performances of reconstruction with various training data

Landau function [22] $f(x) = p_0 \exp(-p_3(\lambda + \exp(-\lambda)))$, where $\lambda = \frac{x-p_1}{p_2}$. The full width at half maximum (FWHM) of the fitted distribution is defined as the resolution of α. The distribution of reconstructed ΔD is fitted to a Gaussian function, and its resolution is obtained from the sigma of the Gaussian.

Figure 8a and b shows the distributions and fitting results of α and ΔD obtained by using "Training TT_rot" as training data when the distance of muon track from the center of CD (D_{mc}) is 14.7–16.7 m. Good fits have been achieved for both the angular and spatial resolutions.

Figure 9 shows how the reconstructed α and ΔD vary with D_{mc}. Comparing the performance obtained when training with the "Training TT_rot" and "Training MC" datasets, we observe:

– For the "Training TT_rot" dataset, the angular bias of α is less than 0.4 °, and the spatial bias of ΔD is less than

3 cm. Also, the results from the "Training TT_rot" and "Training MC" are very similar suggesting that using TT reconstructed tracks with experiment data will provide a realistic way to train the network that does not depend on the simulation.

– The sigma (or resolution) of α and ΔD increases with the D_{mc}. When the muon track is close to the edge of CD, the position of the incident point and the exit point are very close to each other which makes an accurate reconstruction of the track direction difficult. This degradation of angular reconstruction at large distances from the CD center is also seen with the existing muon reconstruction methods [2,3]. Overall, the angular resolution is better than 0.6 ° and the spatial resolution is better than 10 cm.

In order to understand our muon reconstruction method better, Fig. 10 provides another check of the muon reconstruction performance obtained with the dataset "Training TT_rot." Here, the Δr resolution is expressed as FWHM of $\Delta r = |\mathbf{r}_{in}^{predict} - \mathbf{r}_{in}^{truth}|$ which means the difference between the true and predicted positions of the muon injection point at LS surface. The direction resolution is shown by the sigma of
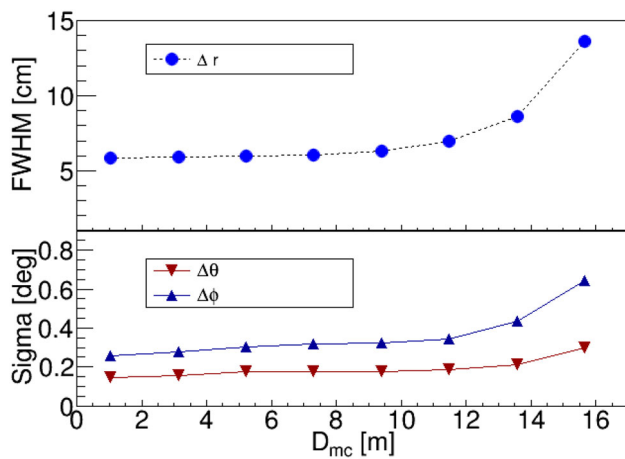
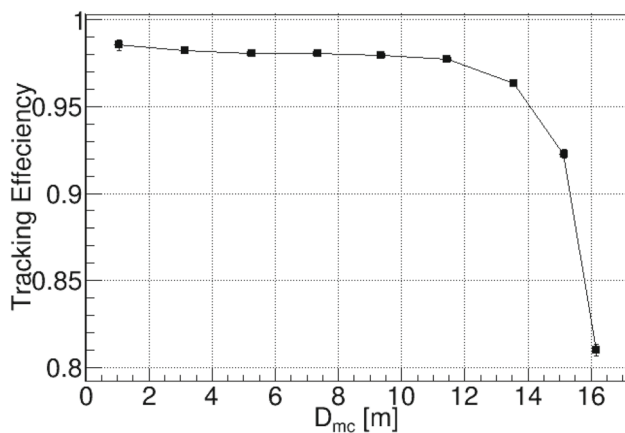**Fig. 10** Reconstruction spatial resolution and angular resolution

Fig. 11 Reconstruction efficiency for various muon track distances to the CD center

the distribution of $\Delta\theta$ and $\Delta\phi$ which express the differences of θ angles and ϕ angle between the reconstructed and true direction of muon track, respectively. From Fig. 10, one can see the results are normal and there are no anomalies.

The track reconstruction efficiency is defined as the ratio of well reconstructed tracks to all tracks, with tracks regarded as well reconstructed when the deviations of all parameters are less than five times their standard deviations. Figure 11 presents the reconstruction efficiency for a range of track distances from the CD center. For distances of less than 14 m, the tracking efficiency is greater than 95% and dropped to 86% for tracks at the edge of the CD.

Reconstruction time cost

The time cost of muon reconstruction using three different approaches is presented in Table 3. The time for the CPU implementation of the neural network method presented in this paper is found to be improved by more than a factor of two compared to the fastest light method. The GPU implementation of the neural network method provides a huge improvement in processing speed by a factor of one hundred compared to the CPU implementation.

Discussion

The muon reconstruction accuracy of the deep learning model largely relies on the amount and the diversity of data available during training. Applying rotation to injected muon tracks effectively augments the training data. However, the disadvantage with the rotation method is that it could lead to small changes in geometry that can cause a muon track represented by an image to lose part of its original features, which might possibly degrade the performance of reconstruction. Studies show that the performance discrepancy between the rotation method and the MC truth method is at the level of a few percent, which is still in an acceptable range. In order to further improve the performance of the CNN model, more sophisticated data augmentation method, such as data augmentation using GANs [23], can be investigated and potentially used to generate artificial training data in the future.

The muon reconstruction results presented in the previous section are obtained in an ideal scenario. With the real experimental data, there are many factors that could possibly degrade the performance of the model. The possible influence factors are as follows: the dark noise created by the PMT, electronics saturation, the deformation of acrylic sphere, etc.

When the experiment starts data taking, it takes time to complete the tuning of Monte Carlo simulation to make the simulation consistent with the data. During this period, deep learning model can be obtained by training with the augmented data sample. As soon as the reliable simulation data are available, the training can move to use the full simulation data.

Table 3 Reconstruction speed	Method	Hardware	Time per event (ms)
	Fastest light	CPU E5-2650 v4	5000
	Neutral network (batch size = 1)	CPU E5-2650 v4	2000
	Neutral network (batch size = 1)	GPU Tesla V100	20

Conclusion

In this study, a deep learning convolutional neural network approach is applied to muon track reconstruction in the JUNO central detector. Achieving good agreement between the real JUNO experimental data and the simulated MC data is expected to require extensive efforts to develop a detailed understanding of many small effects corresponding to improvements to the models of the scintillation, re-emission, photon propagation into PMTs and the electronics. Reconstruction techniques which are not dependent on the availability of finely tuned simulation data are highly beneficial. In this study, the track information reconstructed by the top tracker is used for network training instead of relying on Monte Carlo truth information. Due to the limited angular coverage of the top tracker a randomized event rotation technique has been developed which succeeds to generate sufficient training data by making the assumption of rotational symmetry. This dataset was used to train a convolutional neural network with promising reconstruction performance. The reconstruction efficiency is higher than 95% with the distance less than 14 meters of the track to CD center. The angular resolution is better than 0.6 ° and the spatial resolution is better than 10 cm, and these results demonstrate that muon reconstruction a convolutional neural network is able to provide an performance that meets the requirements. Compared to existing methods [3], the CNN model does not need to introduce additional real data correction. In addition, the GPU implementation of the technique has the advantage of a processing speed more than 100 times faster than conventional techniques.

References

1. Z. Djurcic et al., (JUNO Collaboration), arXiv:1508.07166
2. C. Genster, M. Schever, L. Ludhova, M. Soiron, A. Stahl, C.Wiebusch, https://doi.org/10.1088/1748-0221/13/03/T03003. arXiv:1906.01912
3. K. Zhang, M. He, W. Li, J. Xu, https://doi.org/10.1007/s41605-018-0040-8. arXiv:1803.10407
4. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma et al., arXiv:1409.0575
5. R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, in *CVPR* (2014)
6. J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in *CVPR* (2015)
7. G. Kasieczka, T. Plehn, M. Russell, T. Schell, https://doi.org/10.1007/JHEP05(2017)006. arXiv:1701.08784
8. P. Baldi, K. Bauer, C. Eng, P. Sadowski, D. Whiteson, https://doi.org/10.1103/PhysRevD.93.094034. arXiv:1603.09349
9. W. Bhimji, S.A. Farrell, T. Kurth, M. Paganini, Prabhat, E. Racah, https://doi.org/10.1088/1742-6596/1085/4/042034. arXiv:1711.03573
10. A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M.D. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, P. Vahle, https://doi.org/10.1088/1748-0221/11/09/P09001. arXiv:1604.01444
11. X. Huang et al., offline data processing software for the JUNO experiment, in *PoS ICHEP2016* (2017), p. 1051. https://doi.org/10.22323/1.282.1051
12. T. Lin et al., [JUNO], J. Phys. Conf. Ser. **898**(4), 042029 (2017). https://doi.org/10.1088/1742-6596/898/4/042029. arXiv:1702.05275 [physics.ins-det]
13. J. Allison et al., Recent developments in geant4. NIM A **835**, 186–225 (2016). https://doi.org/10.1016/j.nima.2016.06.125
14. V.A. Kudryavtsev, Comput. Phys. Commun. **180**, 339–346 (2009) https://doi.org/10.1016/j.cpc.2008.10.013. arXiv:0810.4635 [physics.comp-ph]
15. M. Guan, M.C. Chu, J. Cao, K.B. Luk, C. Yang, arXiv:1509.06176
16. K.K. Pal, K.S. Sudeep, Preprocessing for image classification by convolutional neural networks (IEEE, 2016), pp. 1778–1781. https://doi.org/10.1109/RTEICT.2016.7808140
17. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. ICLR **1**, 5 (2015)
18. D. Scherer, A. Muller, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition, in *Proceedings of the International Conference on Artificial Neural Networks (ICANN), Thessaloniki, Greece, 15–18 September 2010* (Springer, 2010)
19. E. Phaisangittisagul, An analysis of the regularization between L2 and dropout in single hidden layer neural network (IEEE, 2016), pp. 174–179
20. M. Abadi, et al., Tensorflow: large-scale machine learning on heterogeneous distributed systems (2015). arXiv:1603.04467
21. Y. LeCun, L. Bottou, G.B. Orr, K.R. Müller, *Neural Networks: Tricks of the Trade* (Springer, Berlin, 1998)
22. X.X. Cao, W.D. Li, R.A. Briere, C.L. Liu, Z.P. Mao, S.J. Chen, Z.Y. Deng, K.L. He, X.T. Huang, B. Huang et al., Chin. Phys. C **34**, 1852–1859 (2010). https://doi.org/10.1088/1674-1137/34/12/012
23. F.H.K. dos Santos Tanaka, C. Aranha. Data augmentation using GANs (2019). arXiv:abs/1904.09135