Reconstruction of Track-like Event in TRIDENT Project Based on Graph Neural Network

Cen Mo,[a,*] Fan Hu,[b] Liang Li[a] and Donglian Xu[a,c] for the TRIDENT collaboration

[a] *School of Physics and Astronomy, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai, China*

[b] *Department of Astronomy, Peking University, 5 Yiheyuan Rd, Beijing, China*

[c] *Tsung-Dao Lee Institute, Shanghai Jiao Tong University, 520 Shengrong Rd, Shanghai, China*

E-mail: mo_cen@sjtu.edu.cn

TRoplcal DEep-sea Neutrino Telescope (TRIDENT) is a next-generation neutrino detector to be located in the South China Sea. Muon track events are the primary channel for the discovery of potential astrophysical neutrino sources. In a typical track-like event, less than 1% of photosensors are hit, making Graph Neural Networks particularly well-suited for their reconstruction. In this study, we have trained Graph Neural Networks with simulated track-like events to reconstruct the direction and energy of incoming muons with high resolution. We present the accuracy and speed of the machine learning based method and compare it with classic reconstruction methods.

38th International Cosmic Ray Conference (ICRC2023)
26 July - 3 August, 2023
Nagoya, Japan

ICRC2023
38th International Cosmic Ray Conference
The Astroparticle Physics Conference

*Speaker

1. Introduction

Machine learning techniques have already been widely used in high-energy physics. They provide a powerful handle for analysing data with high dimensionality and volume. Various attempts have been made to investigate the power of machine learning approaches in neutrino telescopes in recent years. Deep neural networks and Boosted Decision Trees (BDTs) are used to classify cascade events in IceCube [1]. 3D convolutional neural networks (CNNs) are applied to reconstruct various neutrino events in KM3NeT/ORCA [2]. CNNs with hexagonally shaped kernel are utilized to reconstruct cascade events in IceCube's hexagonal geometry [3]. Sparse Submanifold Convolutional Neural Networks (SSCNNs) are trained to overcome sparsity of input data and works as a trigger-level event reconstruction for IceCube [4]. Graph Neural Networks (GNNs) are developed for low-energy event reconstruction and classification in IceCube [5]. Reconstruction approaches based on machine learning have been shown to be faster and easier to develop compared to the traditional likelihood based reconstruction method.

TRoplcal DEep-sea Neutrino Telescope (TRIDENT) aims to identify and study astrophysical neutrino sources [6]. There are mainly three kinds of neutrino events: track-like events from muon neutrinos in charged-current interactions, shower-like events from neutrinos in neutral-current interactions and electron neutrinos in charged-current interactions, and finally double-cascade events from tau neutrinos in charged-current interactions. In this study, we introduce a GNN-based method to reconstruct muon track events simulated using TRIDENT's simulation framework. We have found that trained GNN models can achieve high resolution and rapid processing speed when reconstructing the direction and energy of the muons.

2. Track-like Event Simulation

We have simulated ν_μ charge current interactions events from 1 TeV to 1 PeV. The ν_μ energy distribution takes the form of E^{-2}. The simulated events are generated using the TRIDENT simulation framework consisting of two components: the event generator and the detector response simulator.

The event generator simulates the deep inelastic scattering (DIS) process for ν_μ based on the CORSIKA8 simulation framework [7]. The TRIDENT detector is represented by a cylinder with 2500m radius and 1000m height located 2900m below the sea level. The energy of the neutrino E_ν is sampled from a E^{-2} energy spectrum. The direction of the neutrino is uniformly sampled from a 4π solid angle. The DIS process is simulated with PYTHIA8 program [8] as shown in Figure 1. $\hat{n}_\nu$ is the neutrino direction and $\vec{x}_f$ is the DIS vertex.

Particles decay and travel through the TRIDENT detector until they reach the edge of the detector. The interaction processes between particles and the detector are simulated using the detector response simulator.

The detector response simulator is built based on Geant4 software framework [9, 10] . The detector is made of 1200 vertical strings in a Penrose tiling pattern with a radius of 2000m, as shown in Figure 2. For each string, there are 20 digital optical modules (DOMs) with a vertical spacing of 30m. In the detector response simulator, we use Geant4 to simulate the propagation of charged particles and the emission of Cherenkov photons. The cascade of electrons is described by

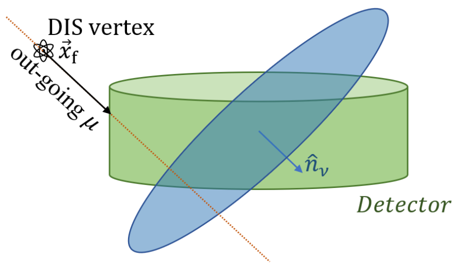

Figure 1: The DIS vertex sampling.

parameterization functions to speed up particle-by-particle simulation of the cascade by $\mathcal{O}(1000)$ times. The OptiX ray tracing framework [11] is utilized to accelerate the propagation of Cherenkov photons tracked up to the surface of DOMs.

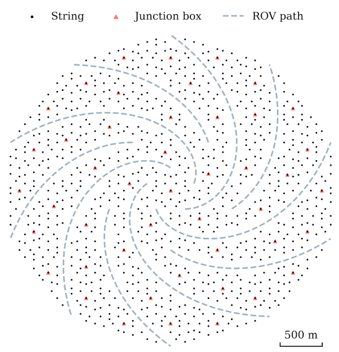

Figure 2: Top view of TRIDENT detectors.

To achieve a large photon detection area and measure precise photon hit times, new type of DOMs called hybrid DOMs (hDOMs) [12] which contain multiple Photomultiplier Tubes (PMTs) and Silicon Photomultipliers (SiPMs) are used in the TRIDENT detector. Inside the hDOMs, optical processes such as the refraction, absorption, and reflection of Cherenkov photons are simulated.

3. Network Architecture

To fully utilize the topological features of track-like events in neutrino telescope, we evaluated performances of various neural networks. We adopted the Submanifold Sparse Convolutional Network (SSCNN)[13] for the task of direction reconstruction. SSCNN demonstrated the ability to effectively handle high-dimensional data in neutrino telescope. However, it consistently undeperformed the GNN in terms of angular resolution for track-like events. Therefore, we choose to use GNN for track-like event reconstruction.

Each neutrino event in TRIDENT can be represented as a point cloud, where the point cloud is an edge-less graph comprised by a collection of nodes positioned within a standard 3D coordinate system with XYZ axes. In this context, a neutrino event is denoted as $G = \{u, x_i, pos_i\}$. Here, u represents the global feature of the event, capturing overall event characteristics, while x_i represents the feature associated with the i-th DOM. Additionally, pos_i signifies the location of the i-th DOM. The construction of u and x_i varies depending on the specific task at hand.

As depicted in Figure 3, our GNN architecture incorporates a fundamental building block known as the EdgeConv block, modified from the EdgeConv block used in ParticleNet [14]. The EdgeConv block, initially introduced in [15], functions as a convolution-like operation designed to capture local geometric attributes within point clouds. Given a point cloud G, the EdgeConv block begins by establishing edges for the point cloud. Each node is connected to its k nearest neighboring points through edges, where k is a user-defined hyperparameter. For each edge, a feature vector is defined by $e_{ij} = \phi_\theta(u, x_i, x_j - x_i)$, where ϕ is a multilayer perceptron (MLP) with θ denoting its parameters. The EdgeConv block then performs an aggregation operation to gather information from the connected nodes. The feature vector of the i-th node is updated as $x'_i = (\underset{j=1,\ldots k}{Max}\{e_{ij}\} + x_i)$, where Max is the aggregation operation and x_i is added as a shortcut connection, as introduced in [16]. The output of the EdgeConv block is passed through a Rectified Linear Unit (ReLU) activation function [17] to introduce non-linearity.

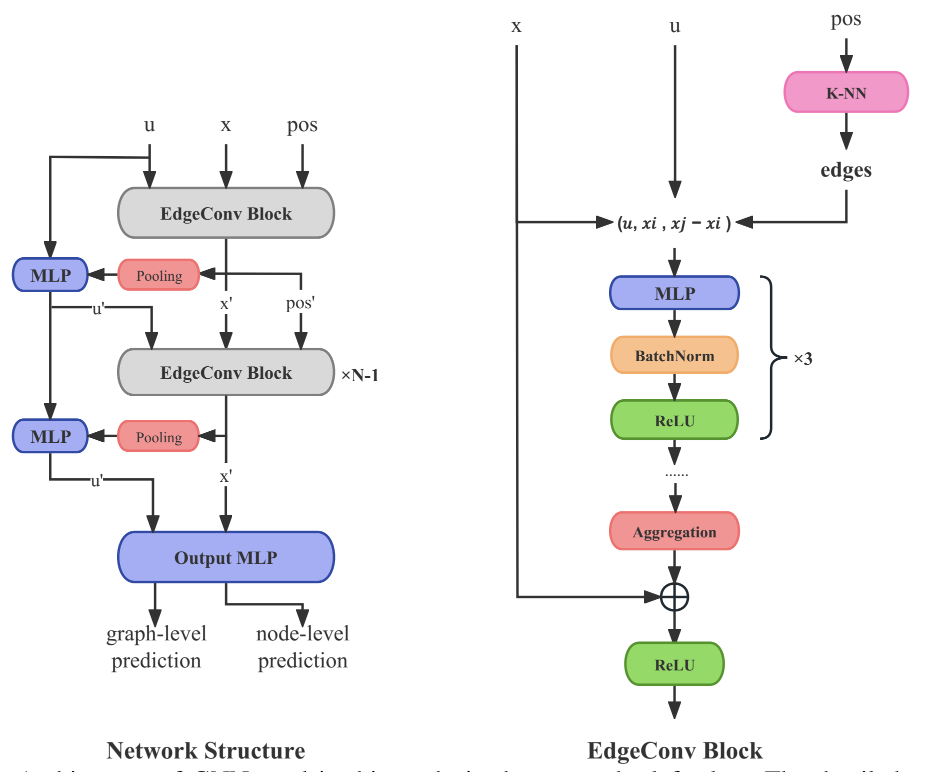

Figure 3: Architecture of GNN used in this study is shown on the left plot. The detailed structure of EdgeConv block is illustrated on the right plot.

The complete GNN architecture incorporates multiple iterations of the EdgeConv block, each with different parameters, to enrich the graph representation and capture high-level information. During each iteration, the EdgeConv block takes the input graph G and updates the node features to x'_i, resulting in a updated point cloud:

$$\begin{aligned} u' &= \Phi_\Theta(u, \text{Global\_Average\_Pooling}(\{x'_i\})) \\ x'_i &= x'_i \\ pos'_i &= x'_i \end{aligned}$$

where Φ is another MLP with parameters Θ. After the final iteration, the GNN produces an enriched graph with high-level information. The desired physical parameters can then be reconstructed using an output MLP layer, with the input being either the set of node features x_i for DOM-level reconstruction or the global feature vector u for event-level reconstruction.

Classic reconstruction methods for direction of muon tracks utilize photon residual times t_{res}, which quantify the difference between the time of photon hit and the expected time for photons based on their geometry. By constructing a probability density function (PDF) of t_{res}[12], the maximum likelihood methods achieve an angular resolution of approximately 0.1 degrees for high-energy ν_μ events[6]. Inspired by this approach, we design the input features to focus on the time and position of each hit. For a specific ν_μ event, we denote the position and time of the first photon hit as $D_0 = (x_0, y_0, z_0)$ and T_0, respectively. Each triggered DOM is represented as a node in the point cloud, with the position of the i-th node defined as the relative spatial vector $pos_i = (x_i, y_i, z_i) - D_0$. The feature vectors of the nodes consist of the concatenated position, time, and the number of recorded hits: $x_i = (pos_i, c(t_i - T_0), n_i)$, where t_i is the time of the first photon hit for the i-th DOM and c is the speed of light. The global feature vector is obtained by performing a Global Average Pooling operation on the set of node feature vectors: $u = \text{Global\_Average\_Pooling}(x_i)$. The specific feature vectors used for each graph are summarized in Table 1.

Table 1: Input Feature Description

Feature	Description
x, y, z	Relative position for DOM
ct	Relative time of first hit for DOM
n	Number of hits for DOM
u	Global features for graph

4. Reconstruction Results

In this study, we utilize the aforementioned GNN architecture to reconstruct the direction and energy of ν_μ events. The angular resolution is presented and compared with results using the maximum likelihood method employed in TRIDENT. Following this, we introduce another model that predicts the energy of muons resulting from ν_μ charge current interactions.

4.1 Direction Reconstruction

Instead of directly reconstructing the direction vector $\hat{n}_\mu = (n_x, n_y, n_z)$ of the muon, our network is trained to reconstruct the emission locations of Cherenkov photons. For each triggered DOM located at $\vec{D}_i$, the earliest trigger time T_i is known. By leveraging the truth information $(\vec{x}_\mu, t_\mu, \hat{n}_\mu)$ of the muon, we can determine the theoretical photon emission position, $\vec{r}_i$, as is illustrated in Figure 4. The network is trained to predict $\vec{r}_i$ for each triggered DOM, and the muon track is then reconstructed using a simple linear fit based on the predicted $\vec{r}_i$. Our findings indicate that the combination of GNN and the least squares method results in a smaller angular resolution compared to directly predicting the muon direction $\hat{n}_\mu$ using GNN.

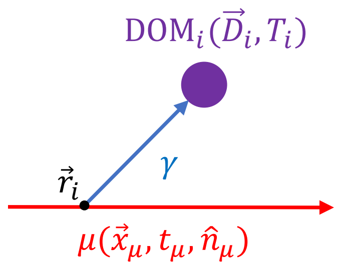
$\mathrm{DOM}_i(\vec{D}_i, T_i)$
γ
$\vec{r}_i$
$\mu(\vec{x}_\mu, t_\mu, \hat{n}_\mu)$

Figure 4: Muon emits Cherenkov photon at $\vec{r}_i$ and triggers DOM_i.

To reconstruct the emission position, we constructed a network comprising 6 EdgeConv blocks followed by 2 layers of MLP as the output block. In each EdgeConv block, the parameter k was set to 16. Since our target is to reconstructing the physical parameters of individual DOMs, the network is configured in the node-level prediction mode.

During the training process, a set of event selection criteria is applied to the dataset, enabling the network to effectively capture the topological features of ν_μ events. To highlight the unique characteristics of track-like events, training samples are filtered to include only those with a minimum muon track length of 500m. Furthermore, training samples are further constrained to contain more than 8 triggered DOMs. The loss function used in training is mean square error (MSE) and weights proportional to the number of photon hits, n_i, are applied:

$$Loss = \sum_i n_i \times |\vec{r}_i^{\,\mathrm{pred}} - \vec{r}_i|^2 / \sum_i n_i \tag{1}$$

To evaluate the performance of the model, the training samples are divided into two sets: the training set and the test set. The model with the minimum loss on the test set is selected as the final model.

The result is obtained by applying the trained model to a separate dataset consisting of 130,000 ν_μ events. For this evaluation, the criteria for event selection are relaxed compared to the strict criteria mentioned earlier. Here, the samples are required to trigger at least 2 DOMs with more than 10 hits, which is the minimum requirement for identifying a track-like event. Figure 5 presents the angular resolution achieved by the GNN model as a function of neutrino energy in the left plot, while the right plot displays the results of the traditional likelihood method. Both methods exhibit an angular resolution at the sub-degree level across most energy regions, demonstrating the effectiveness of the GNN model in reconstructing the neutrino direction.

The GNN reconstruction process is performed on a 32GB NVIDIA V100 GPU, with each batch consisting of approximately 2500 events. The average time required to reconstruct a single event is approximately 2ms. In contrast, the likelihood method spends roughly 350ms per event. This significant reduction in processing time highlights the computational advantages of the GNN approach in comparison to the traditional likelihood method.

We have found that the likelihood method outperforms the GNN method in terms of angular resolution. This outcome possibly can be attributed to the fact that the neural network is designed with a general architecture without prior neutrino-related knowledge. Incorporating physics constraints into the architecture may lead to improved performance.

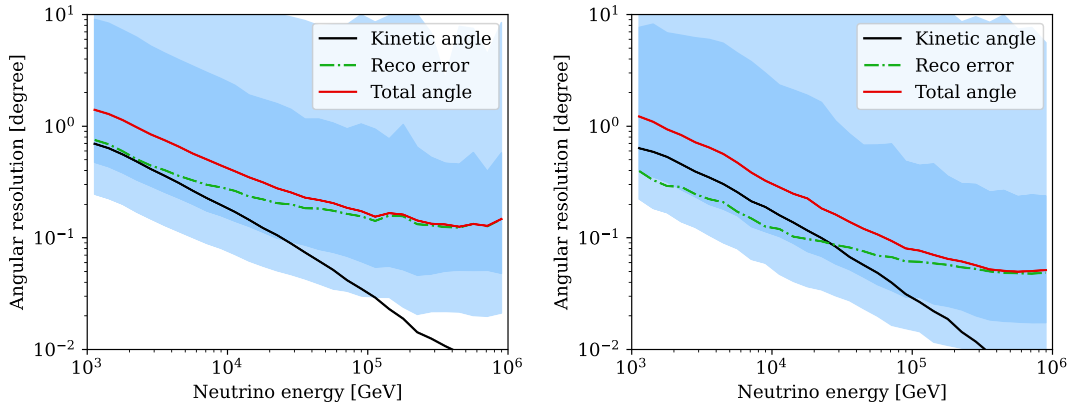

Figure 5: Median angular error of GNN (left) and likelihood method (right) depend on energy of ν_μ. The median angle between the reconstructed track and the true direction of μ and ν_μ is visualized by the green and red lines, respectively. Color bands exhibits the 68% and 90% quantiles. Black lines are the median angle between direction of μ and ν_μ.

4.2 Energy Reconstruction

In the generation of ν_μ charge current events, the DIS interaction vertex is likely to be generated outside of the detector region. In such cases, the energy loss of the muon before it reaches detector region is irreducible. Therefore, in this section we focus on the reconstruction of muon energy upon its arrival within the detector region using the GNN method.

For the task of reconstructing muon energy, we constructed a network comprising 4 EdgeConv blocks followed by 2 layers of MLP as the output block. In each EdgeConv block, the parameter k was set to 10. The smaller value of k makes the network to focus on capturing local information such as dE/dx, which is a valuable indicator for inferring the muon energy.

Considering that the value of E_μ spans several orders of magnitude, the network is trained to predict $log_{10}E_\mu$. The loss function is set to be MSE between the predicted and truth values of $log_{10}E_\mu$. Additionally, weights $w = log_{10}E_\mu - 2.5$ are applied to each sample during training. Denoting $log_{10}E_\mu$ for the n-th sample as y_n^{truth}, we have:

$$Loss = \sum_n w_n \times |y_n^{\text{pred}} - y_n^{\text{truth}}|^2 / \sum_n w_n \tag{2}$$

Weights are required here since the energy spectrum for training sample is soft, which leads to an imbalance in the distribution. By applying weights, the imbalance is mitigated, allowing the network to effectively learn from events across the energy spectrum.

In order to align with the logarithmic scale used in the target variable, the n_i term in the input feature vectors is transformed to $log(n_i)$. The trained model tends to predict a lower energy than truth E_μ. To address this, a shift term, $b = 0.15$, is added to each predicted value. The reconstructed energy is then defined as $E_{\text{recon}} = 10^{y^{\text{pred}}+b}$. The results of the energy reconstruction is illustrated in Figure 6. The median of $|log_{10}(E_{\text{recon}}/E_\mu)|$ is smaller than 0.2 for E_μ >1TeV.

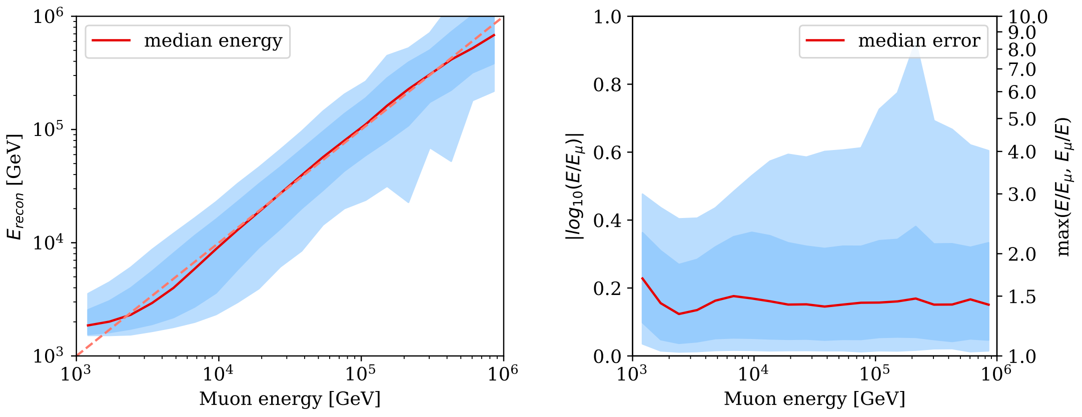

Figure 6: Energy reconstruction result of GNN. Left plot is the comparison between truth E_μ and energy predicted by GNN, and right plot illustrates the ratio between two quantity. Solid line is the median value and color bands corresponds to 68% and 90% quantiles.

5. Conclusion

This study presents a graph neural network architecture for the reconstruction of neutrino events in TRIDENT. The neural network is trained to accurately reconstruct the direction of ν_μ from track-like events. The model achieves an angular resolution a little worse than that of the traditional likelihood method but with a speed improvement of approximately 100 times. Another model is trained to reconstruct the muon energy, achieving a median value smaller than 0.2 in $|log_{10}(E_{\text{recon}}/E_\mu)|$. With room to improve, the architecture aims to be used to separate neutrino signal from the atmospheric muon background.

References

[1] S. Sclafani and M. Hünnefeld, "A search for neutrino sources with cascade events in IceCube." arXiv:2107.09103.

[2] S. Aiello *et al.*, "Event reconstruction for KM3net/ORCA using convolutional neural networks," *Journal of Instrumentation*, vol. 15, pp. P10005–P10005, oct 2020.

[3] R. Abbasi *et al.*, "A convolutional neural network based cascade reconstruction for the IceCube neutrino observatory," *Journal of Instrumentation*, vol. 16, no. 7, p. P07041.

[4] F. J. Yu, J. Lazar, and C. A. Argüelles, "Trigger-level event reconstruction for neutrino telescopes using sparse submanifold convolutional neural networks," 2023. arXiv:2303.08812.

[5] R. Abbasi *et al.*, "Graph neural networks for low-energy event classification &amp reconstruction in IceCube," *Journal of Instrumentation*, vol. 17, p. P11003, nov 2022.

[6] Z. P. Ye *et al.*, "Proposal for a neutrino telescope in South China Sea," 7 2022. arXiv:2207.04519.

[7] T. Huege, "Corsika 8 – the next-generation air shower simulation framework," 2022. arXiv:2208.14240.

[8] C. Bierlich *et al.*, "A comprehensive guide to the physics and usage of pythia 8.3," 2022. arXiv:2203.11601.

[9] S. A. *et al.* (GEANT4 Collaboration) *Nucl. Instrum. Meth. A*, vol. 506, no. 3, pp. 250–303, 2003.

[10] A. *et al.* (GEANT4 Collaboration) *IEEE Transactions on Nuclear Science*, vol. 53, no. 1, pp. 270–278, 2006.

[11] S. Blyth, "Opticks : GPU Optical Photon Simulation for Particle Physics using NVIDIA® OptiXTM," *EPJ Web Conf.*, vol. 214, p. 02027, 2019.

[12] F. Hu, Z. Li, and D. Xu, "Exploring a PMT+SiPM hybrid optical module for next generation neutrino telescopes," *PoS*, vol. ICRC2021, p. 1043, 2021.

[13] B. Graham and L. van der Maaten, "Submanifold sparse convolutional networks," 2017. arXiv:1706.01307.

[14] H. Qu and L. Gouskos, "Jet tagging via particle clouds," *Physical Review D*, vol. 101, mar 2020.

[15] Y. Wang *et al.*, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, vol. 38, oct 2019.

[16] K. He *et al.*, "Deep residual learning for image recognition," 2015. arXiv:1512.03385.

[17] A. F. Agarap, "Deep learning using rectified linear units (relu)," 2019. arXiv:1803.08375.