PROCEEDINGS OF SCIENCE

Trigger-Level Event Reconstruction for Neutrino Telescopes Using Sparse Submanifold Convolutional Neural Networks

Felix J. Yu,[a,*] **Jeffrey Lazar**[a,b] **and Carlos A. Argüelles**[a]

[a] *Department of Physics and Laboratory for Particle Physics and Cosmology, Harvard University, Cambridge, MA 02138, US*

[b] *Department of Physics and Wisconsin IceCube Particle Astrophysics Center, University of Wisconsin–Madison, Madison, WI 53706, USA*

E-mail: felixyu@g.harvard.edu, jlazar@icecube.wisc.edu, carguelles@g.harvard.edu

Convolutional neural networks (CNNs) have seen extensive applications in scientific data analysis, including in neutrino telescopes. However, the data from these experiments present numerous challenges to CNNs, such as non-regular geometry, sparsity, and high dimensionality. Consequently, CNNs are highly inefficient on neutrino telescope data, and require significant pre-processing that results in information loss. We propose sparse submanifold convolutions (SSCNNs) as a solution to these issues and show that the SSCNN event reconstruction performance is comparable to or better than traditional and machine learning algorithms. Additionally, our SSCNN runs approximately 16 times faster than a traditional CNN on a GPU. As a result of this speedup, it is expected to be capable of handling the trigger-level event rate of IceCube-scale neutrino telescopes. These networks could be used to improve the first estimation of the neutrino energy and direction to seed more advanced reconstructions, or to provide this information to an alert-sending system to quickly follow-up interesting events.

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

ICRC2023
38th International Cosmic Ray Conference
The Astroparticle Physics Conference

*Speaker

PoS(ICRC2023)1004

1. Introduction

Gigaton-scale neutrino telescopes have ushered in a fresh perspective of the Universe, enabling us to examine the highest energy neutrinos. While there are a variety of proposed designs, many follow the detection principle outlined by the DUMAND project [20] and consist of an array of optical modules (OMs) deployed in liquid or solid water. This detector paradigm shows great promise, and analyses by these experiments have already provided the first evidence of astrophysical neutrino sources [4, 6]. However, prior to analysis, it's crucial to separate high-energy neutrinos from the overwhelming background induced by cosmic-ray muons. While a high-energy neutrino may trigger a detector once every few minutes, cosmic-ray muons typically induce a trigger rate on the order of kHz.

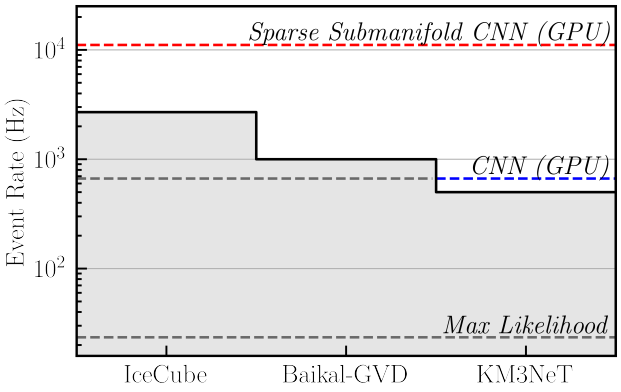

Figure 1: ***Event rates of triggers in different neutrino telescopes [1, 10, 12] compared to the run-times of various reconstruction methods.*** Sparse submanifold CNNs and their performance are detailed in this article. The CNN and maximum likelihood method run-times are taken from [16]. Notably, sparse submanifold CNNs can process events well above standard trigger rates in both ice- and water-based experiments.

Given their inability to cross a significant part of the Earth without halting, cosmic-ray muons exhibit a unique zenith dependency. This characteristic allows for their elimination by applying a cut on the reconstructed direction of an event. Consequently, a dependable reconstruction capable of managing the ~kHz background frequency is the initial stage in distinguishing neutrinos. Moreover, a rapid reconstruction method could serve as part of an alert system that notifies researchers of events that are highly likely to be astrophysical neutrinos.

At the trigger-level, a simple but fast reconstruction is typically done by solving a least squares problem via matrix inversion, as is the case for `LineFit` [2] in IceCube or `QFit` in ANTARES [8]. Machine learning has demonstrated potential by offering a reconstruction of similar quality with fewer runtime demands [14, 16]; however, the fastest convolutional neural network (CNN) developed for high-energy neutrinos is not able to keep pace with a kHz-scale trigger-level rate. In this proceeding, we present a reconstruction technique employing a sparse submanifold CNN (SSCNN), which resolves this runtime challenge. We will illustrate our method by focusing on solid-water detectors, but our results and conclusions readily generalize to water-based detectors. In this context, our SSCNN achieves better angular resolutions than methods such as `Linefit` while requiring a comparable run-time, enabling improved trigger-level cuts and serving as a better seed for the

likelihood-based reconstruction. Fig. 1 summarizes typical event rates found in neutrino telescopes and compares these to the execution rate of various reconstructions. At the same time, SSCNN is also able to reconstruct the neutrino energy, a task which has not been done at trigger-level.

The rest of this article is organized as follows. In Sec. 2 we motivate and introduce sparse submanifold convolutions; in Sec. 3 we describe the data sets used for training and testing; in Sec. 4 we evaluate the performance of the network. Finally, in Sec. 5 we conclude with some parting words. The code detailing our implementation of SSCNN has been made available at Ref. [21].

2. Methods

Convolutional neural networks (CNNs) have become the staple architecture for image-like data, and have achieved great success in a wide range of applications, including neutrino physics [5, 19]. However, data from neutrino telescopes presents inherent challenges to CNNs. In particular:

- *Non-regular geometry*: CNNs are designed to operate on images, which are arranged on Cartesian grids. Neutrino telescope sensors are typically spaced irregularly [9], with varying distances and arrangements in between each sensor.
- *Sparsity*: Traditional CNNs use convolutions which operate on all points in the given input data. This leads to computational inefficiencies when the data is sparse.
- *High dimensionality*: Events occur in large spatial and temporal scales. This makes using traditional CNNs computationally unfeasible on raw 4D data (three spatial and one time) without information loss or significant pre-processing.

In this proceeding, we present a solution to address these challenges by incorporating sparse submanifold convolutions into our approach [15]. Previous studies have demonstrated the effectiveness of this strategy in neutrino experiments utilizing liquid argon time projection chambers [7, 13]. The utilization of sparse submanifold convolutions in our network offers a natural resolution to the challenges outlined earlier. Sparsity and high dimensionality are no longer a concern, as the number of computations performed will depend only on the number of OM hits. With this improved computational efficiency, we can also handle non-regular geometries more smoothly by using the spatial coordinates of each OM hit (in meters from the center of the detector). This allows us to consider data of any shape or arrangement, without restricting ourselves to a Cartesian grid.

Our SSCNN replaces traditional convolutions with sparse submanifold convolutions. Unlike traditional convolutions that apply a learned kernel over the entire input data, sparse submanifold convolutions exclusively operate on the non-zero elements. This approach effectively eliminates the inefficiency associated with employing CNNs on sparse data, where the majority of operations involve multiplying zeros together. Furthermore, to preserve the sparsity of the data after applying multiple layers in succession, sparse submanifold convolutions enforces that the coordinates and number of output activations matches those of the input. In other words, the features do not spread layer after layer. This is crucial for the efficiency of SSCNNs that are very deep, as the data would otherwise become progressively less sparse throughout the network. The lack of feature spreading will have a minimal impact on performance as long as the network can rely on local information.

It should be noted that SSCNNs still compute over a grid-like structure, but this structure can be arbitrarily large because the network only operates on a submanifold of it.

As input, the SSCNN takes in a list of coordinates representing the space-time coordinates of OMs in which there were a nonzero number of photon hits, and an associated feature vector containing the number of photon hits which occurred within a 1 ns time window on that OM, starting from the time indicated in the coordinate tensor. Notably, each coordinate can correspond to any feature vector, allowing for flexibility in encoding neutrino telescope data configurations. While a 3-dimensional approach focusing solely on spatial positions and using timing information as features is possible, we chose to treat timing as coordinates. This decision enables us to capitalize on the intrinsic time structure inherent in neutrino telescope data.

3. Event Simulation

Our benchmark case follows an ice-embedded IceCube-like geometry, where the OMs are spaced our approximately 125 meters horizontally and 17 meters vertically. The events used in this work are μ^- from ν_μ charged-current interactions. The initial neutrino sampling, charged lepton propagation, and photon propagation were simulated using the `Prometheus` package [18]. The incident neutrinos have energies between 10^2 GeV and 10^6 GeV sampled from a power-law with a spectral index of -1. Since most of the events that trigger neutrino telescopes are downward-going cosmic-ray muons, we generated a down-going dataset. Specifically, the initial momenta have zenith angles between 80° and 180°. It is worth noting that this definition of zenith angle is different from the convention which is typically used by neutrino telescopes, which take 0° to be downgoing. In summary, the data consists of the module ID, module position and time of arrival of all simulated photons which reached an OM. Please see [18] for additional details on the internal simulation process. Importantly, the photons generated in `Prometheus` are only tracked to the surface of the OM.

We then add noise in the style of [17] to the resulting photon distributions. Once all photons have been added, we then implement a trigger criteria similar to the one described in [3], based off pairs of coincident light in neighboring OMs. After this cut, we are left with 462892 events from 3 million simulated events, which we split between the training and test data sets of 412892 and 50000 sizes respectively. One can see distributions of the events which pass this trigger as a function of true energy, zenith, and azimuth in Fig. 2.

Beyond the trigger-level dataset, we assess the network on a dataset with additional quality cuts, enabling performance evaluation on events more likely for final analysis. This involves three quantities: N_{OM}, r_{COG}, and R_{ell}. The initial two variables—the count of unique OMs detecting light and the distance from the charge-weighted center of gravity to the detector's center—are simple. However, R_{ell} requires fitting a two-parameter ellipsoid to all light-detecting OMs, and then determining the long-axis to short-axis ratio—closer to one suggests spherical events, higher ratios imply longer, track-like events. Straight cuts on these variables demand events to meet $N_{\mathrm{OM}} > 11$, $r_{\mathrm{COG}} < 400$ m, and $2 < R_{\mathrm{ell}} < 8$. The first cut eliminates low-charge, hard-to-reconstruct events, the second removes corner clipper events due to μ^- passing near the detector's edge, and the final cut on R_{ell} ensures events have a long lever-arm for reconstruction. These cuts decrease the training and testing dataset sizes to 108585 and 13183 events, respectively.

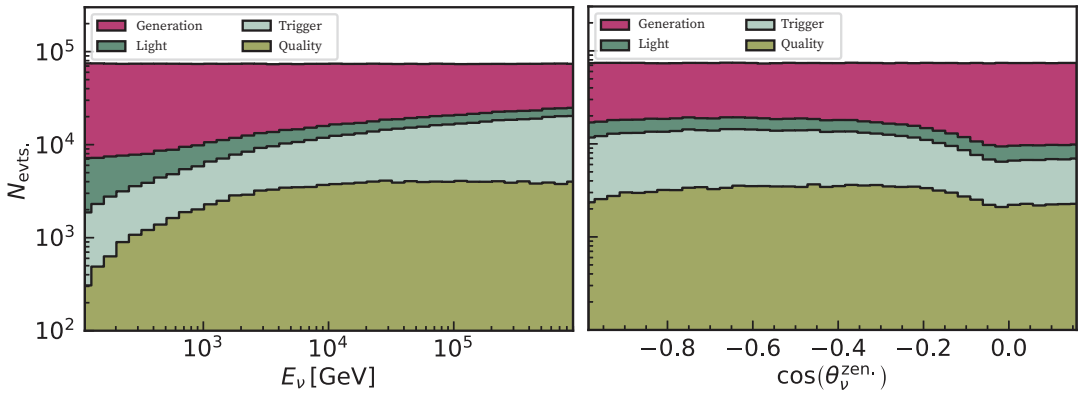

Figure 2: ***Distributions of events in true energy and zenith.*** *Left:* The distribution of events used as a function of true neutrino energy. As expected, the generated distribution is flat when binned logarithmically since the generation was sampled according to a E_ν^{-1} distribution. *Right:* The same distribution as a function of true neutrino zenith angle. Once again the generated distribution is flat in the cosine of this angle, which is proportional to the differential solid angle. nearly flat, with slightly lower efficiency near the horizon.

4. Performance

We utilize a ResNet-based architecture, taking advantage of residual connections between layers to promote robust learning for deeper networks. A typical block of the network consists of a sparse submanifold convolution, followed by batch normalization and the parametric rectified linear unit (PReLU) activation function. We use the `PyTorch` deep learning framework and the `MinkowskiEngine` [11] library to implement the network. The network was trained on each dataset (trigger and quality) for 25 epochs using a batch size of 128 and the `AdamW` optimizer. The initial learning rate was set at 0.001 and was dropped periodically during training.

In this study, we focus on training the network to estimate two key parameters: the primary neutrino energy, denoted as E_ν, and the directional pointing vector composed of its three components (X_ν, Y_ν, Z_ν). To address complexities arising from azimuthal periodicity and undesirable boundary condition behavior at extreme angles, we opt to learn the directional vector directly instead of relying on zenith and azimuth angles.

To accommodate the wide range of magnitudes that these parameters can exhibit, the network is trained to predict the logarithmic energy, $\log_{10}(E_\nu)$, and the normalized directional vectors. This normalization is crucial for handling variations across different scales effectively.

For the energy reconstruction training, the `LogCosh` loss function is used, as it is more robust to outliers than the standard `MSE` loss. The loss function is defined as follows,

$$\mathcal{L}_E = \frac{1}{N}\sum_i^N \log\left(\cosh\left(x_i - y_i\right)\right), \tag{1}$$

where N is the number of events in the batch, x_i are the predictions, and y_i are the labels. For the angular reconstruction, an angular distance loss function is used, namely,

$$\mathcal{L}_A = \frac{1}{N}\sum_i^N \arccos\left(\frac{\vec{X}_i \cdot \vec{Y}_i}{||X_i||\,||Y_i||}\right), \tag{2}$$

SSCNN (GPU)	**0.090 ± 0.007 ms**
SSCNN (CPU)	**65.22 ± 117.04 ms**
Max Likelihood (CPU)	42.6 ± 175 ms

Table 1: ***Per-event average run-time performance.*** The forward pass run-times (mean ± STD) for SSCNN was evaluated on trigger-level events. A likelihood-based method for energy and angular reconstruction was included for reference [16].

where $\vec{X}_i$ and $\vec{Y}_i$ are the predicted and true directional vectors, respectively. The total loss is given by the sum of these two separate losses, with an additional weighting factor on each loss to ensure balanced learning.

We evaluate the run-time performance of the SSCNN in terms of the forward pass duration on both CPU and GPU hardware. The CPU benchmark is performed on a single core of an Intel Xeon Platinum 8358 CPU, while the GPU benchmark uses a 40 GB NVIDIA A100. As is generally the case for neural networks, running on GPU is preferred due to its superior parallel computation capabilities. Additionally, the use of sparse submanifold convolutions has greatly enhanced our GPU memory efficiency, enabling us to run larger batch sizes during inference. The SSCNN can process events at a rate of 11,098 Hz on a 40 GB NVIDIA A100 GPU, while handling a batch size of 12,288 events simultaneously. This is fast enough to handle the expected ~kHz current and planned large neutrino telescopes. The run-time results on both GPU and CPU are summarized in Table 1.

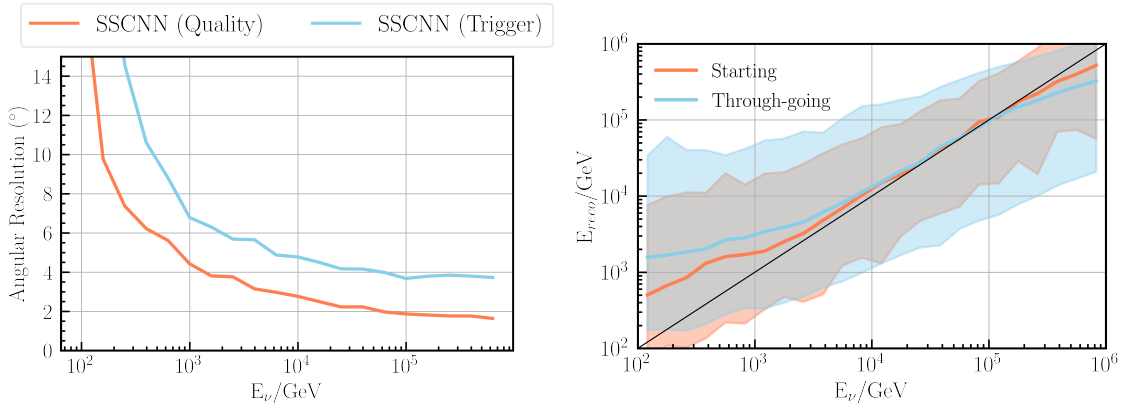

Figure 3: ***Reconstruction performance of SSCNN.*** *Left:* Angular reconstruction results. *Right:* Energy reconstruction results. Both results are binned by the true neutrino energy. Solid lines indicate the median of each bin, while the shaded regions in of the energy reconstruction results are the 5% to 95% confidence level bands.

We first test the network on reconstructing the direction of the primary neutrino. We measure performance using the angular resolution metric, which is calculated by taking the angular difference between the predicted and true directional vectors. Fig. 3 shows the angular resolutions as a function of the true neutrino energy. Lower-energy events generally produce less photon hits, leading to a shorter lever-arm and, consequently, worse resolution. As expected, the trigger-level events are generally harder to reconstruct due to the lower light yield and the presence of corner-clipper

events. This is especially true for angular reconstruction, where the SSCNN is able to reach under 4° median angular resolution on the highest-energy events. Enforcing the previously described quality cuts improves the results of the SSCNN by roughly 2° across the entire energy range. This performance is comparable to or better than current trigger-level reconstruction methods used in neutrino telescopes. For example, the current trigger-level direction reconstruction at IceCube is done using the traditional `Linefit` algorithm [2], which has a median angular resolution of approximately 10° on raw data.

We also test the networks by reconstructing the energy of the primary neutrino. Fig. 3 summarizes the energy reconstruction results. Events where the interaction point of the neutrino occurs outside the detector, known as through-going events, make up the majority of our dataset. As a result, predicting the neutrino energy has an inherent, irreducible uncertainty produced by the unknown interaction vertex and the muon losses outside of the detector. This missing-information problem leads to an intrinsic uncertainty in the logarithmic neutrino energy of approximately 0.4 for a through-going event. Additionally, the network performs noticeably worse at the lowest and highest energies, with a tendency to overpredict at low energies and underpredict at high energies. This behavior can be attributed to the artificial energy bounds on the simulated training dataset.

5. Conclusions

In this proceeding, we have utilized an SSCNN for event reconstruction in neutrino telescopes. It has been demonstrated that these networks maintain competitive performance in energy and angular reconstruction tasks, operating at the μs time scale. This acceleration allows the SSCNN to process events at a significantly higher rate than the trigger rate of current neutrino telescopes, a measure anticipated to be analogous for other active or under-construction neutrino telescopes such as IceCube, KM3NeT, P-ONE, and Baikal-GVD. Reaching this threshold makes the SSCNN a feasible option for online reconstruction at the detector site where resources are limited and where first guesses of the energy and direction of the neutrino are made. As highlighted in the introduction, this can have a substantial impact on current real-time analyses, where our first estimations can also be utilized in an alert-sending system, which will notify collaborators if the detector sees an interesting event. Additionally, these reconstructions can serve as seeds for more time-consuming reconstructions, and thus improving these first estimations will be beneficial to all subsequent analyses.

References

[1] Very high-energy gamma-ray follow-up program using neutrino triggers from icecube. *Journal of Instrumentation*, 11(11):P11009, nov 2016.

[2] M. Aartsen et al. Improvement in fast particle track reconstruction with robust statistics. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 736:143–149, feb 2014.

[3] M. G. Aartsen et al. The IceCube Neutrino Observatory: Instrumentation and Online Systems. *JINST*, 12(03):P03012, 2017.

[4] M. G. Aartsen et al. Neutrino emission from the direction of the blazar TXS 0506+056 prior to the IceCube-170922A alert. *Science*, 361(6398):147–151, 2018.

[5] R. Abbasi et al. A Convolutional Neural Network based Cascade Reconstruction for the IceCube Neutrino Observatory. *JINST*, 16:P07041, 2021.

[6] R. Abbasi et al. Evidence for neutrino emission from the nearby active galaxy NGC 1068. *Science*, 378(6619):538–543, 2022.

[7] P. Abratenko et al. Semantic segmentation with a sparse convolutional neural network for event reconstruction in MicroBooNE. *Physical Review D*, 103(5), mar 2021.

[8] J. A. Aguilar et al. A fast algorithm for muon track reconstruction and its application to the ANTARES neutrino telescope. *Astropart. Phys.*, 34:652–662, 2011.

[9] J. Ahrens et al. IceCube Preliminary Design Document. 2001.

[10] B. Bakker. Trigger studies for the Antares and KM3NeT neutrino telescopes, July 2011.

[11] C. Choy, J. Gwak, and S. Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, jun 2019.

[12] B.-G. Collaboration. BAIKAL-GVD: Gigaton Volume Detector in Lake Baikal. 2012.

[13] L. Dominé and K. Terao. Scalable deep convolutional neural networks for sparse, locally dense liquid argon time projection chamber data. *Phys. Rev. D*, 102:012005, Jul 2020.

[14] J. García-Méndez, N. Geißelbrecht, T. Eberl, M. Ardid, and S. Ardid. Deep learning reconstruction in ANTARES. *JINST*, 16(09):C09018, 2021.

[15] B. Graham and L. van der Maaten. Submanifold sparse convolutional networks, 2017.

[16] M. Hünnefeld. Online Reconstruction of Muon-Neutrino Events in IceCube using Deep Learning Techniques, 2017.

[17] M. J. Larson. *Simulation and identification of non-Poissonian noise triggers in the IceCube neutrino detector*. PhD thesis, Alabama U., Alabama U., 2013.

[18] J. Lazar, S. Meighen-Berger, C. Haack, D. Kim, S. Giner, and C. A. Argüelles. Prometheus: An Open-Source Neutrino Telescope Simulation. 4 2023.

[19] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, and T. Wongjirad. Machine learning at the energy and intensity frontiers of particle physics. *Nature*, 560(7716):41–48, 2018.

[20] A. Roberts. The birth of high-energy neutrino astronomy: A personal history of the dumand project. *Rev. Mod. Phys.*, 64:259–312, Jan 1992.

[21] F. Yu and J. Lazar. https://github.com/felixyu7/nt_ssnet, 2022.