The Journal of Open Source Software

GraphNeT: Graph neural networks for neutrino telescope event reconstruction

Andreas Søgaard [1¶], **Rasmus F. Ørsøe** [1,2], **Morten Holm** [1], **Leon Bozianu** [1], **Aske Rosted** [3], **Troels C. Petersen** [1], **Kaare Endrup Iversen** [1], **Andreas Hermansen** [1], **Tim Guggenmos** [2], **Peter Andresen** [1], **Martin Ha Minh** [2], **Ludwig Neste** [4], **Moust Holmes** [1], **Axel Pontén** [5], **Kayla Leonard DeHolton** [6], and **Philipp Eller** [2]

1 Niels Bohr Institute, University of Copenhagen, Denmark **2** Technical University of Munich, Germany **3** Chiba University, Japan **4** Technical University of Dortmund, Germany **5** Uppsala University, Sweden **6** Pennsylvania State University, USA ¶ Corresponding author

DOI: 10.21105/joss.04971

Software

- Review
- Repository
- Archive

Editor: Dan Foreman-Mackey

Reviewers:

- @JostMigenda
- @GageDeZoort

Submitted: 06 October 2022
Published: 12 May 2023

Summary

Neutrino telescopes, such as ANTARES (ANTARES Collaboration, 2011b), IceCube (IceCube Collaboration, 2012, 2017), KM3NeT (KM3NeT Collaboration, 2016), and Baikal-GVD (Baikal-GVD Collaboration, 2018) have the science goal of detecting neutrinos and measuring their properties and origins. Reconstruction at these experiments is concerned with classifying the type of event or estimating properties of the interaction.

`GraphNeT` (Søgaard et al., 2023) is an open-source Python framework aimed at providing high quality, user friendly, end-to-end functionality to perform reconstruction tasks at neutrino telescopes using graph neural networks (GNNs). `GraphNeT` makes it fast and easy to train complex models that can provide event reconstruction with state-of-the-art performance, for arbitrary detector configurations, with inference times that are orders of magnitude faster than traditional reconstruction techniques (IceCube Collaboration, 2022a).

GNNs from `GraphNeT` are flexible enough to be applied to data from all neutrino telescopes, including future projects such as IceCube extensions (IceCube-Gen2 Collaboration, 2017, 2021; IceCube-PINGU Collaboration, 2014) or P-ONE (P-ONE Collaboration, 2020). This means that GNN-based reconstruction can be used to provide state-of-the-art performance on most reconstruction tasks in neutrino telescopes, at real-time event rates, across experiments and physics analyses, with vast potential impact for neutrino and astro-particle physics.

Statement of need

Neutrino telescopes typically consist of thousands of optical modules (OMs) to detect the Cherenkov light produced from particle interactions in the detector medium. The number of photo-electrons recorded by the OMs in each event roughly scales with the energy of the incident particle, from a few photo-electrons and up to tens of thousands.

Reconstructing the particle type and parameters from individual recordings (called events) in these experiments is a challenge due to irregular detector geometry, inhomogeneous detector medium, sparsity of the data, the large variations of the amount of signal between different events, and the sheer number of events that need to be reconstructed.

Multiple approaches have been employed, including relatively simple methods (ANTARES Collaboration, 2011a; IceCube Collaboration, 2022b) that are robust but limited in precision and likelihood-based methods (Aartsen & others, 2014; Abbasi et al., 2013; AMANDA Collaboration, 2004; ANTARES Collaboration, 2017; Chirkin, 2013; IceCube Collaboration, 2014, 2021b, 2022b) that can attain a high accuracy at the price of high computational cost and detector specific assumptions.

Recently, machine learning (ML) methods have started to be used, such as convolutional neural networks (CNNs) (IceCube Collaboration, 2021a; KM3NeT Collaboration, 2020) that are comparatively fast, but require detector data being transformed into a regular pixel or voxel grid. Other approaches get around the geometric limitations, but increase the computational cost to a similar level as the traditional likelihood methods (Eller et al., 2022).

Instead, GNNs can be thought of as generalised CNNs that work on data with any geometry, making this paradigm a natural fit for neutrino telescope data.

The `GraphNeT` framework provides the end-to-end tools to train and deploy GNN reconstruction models. `GraphNeT` leverages industry-standard tools such as `pytorch` (Paszke et al., 2019), `PyG` (Fey & Lenssen, 2019), `lightning` (Falcon & The PyTorch Lightning team, 2019), and `wandb` (Biewald, 2020) for building and training GNNs as well as particle physics standard tools such as `awkward` (Pivarski et al., 2020) for handling the variable-size data representing particle interaction events in neutrino telescopes. The inference speed on a single GPU allows for processing the full online datastream of current neutrino telescopes in real-time.

Impact on physics

`GraphNeT` provides a common framework for ML developers and physicists that wish to use the state-of-the-art GNN tools in their research. By uniting both user groups, `GraphNeT` aims to increase the longevity and usability of individual code contributions from ML developers by building a general, reusable software package based on software engineering best practices, and lowers the technical threshold for physicists that wish to use the most performant tools for their scientific problems.

The `GraphNeT` models can improve event classification and yield very accurate reconstruction, e.g., for low energy neutrinos observed in IceCube. Here, a GNN implemented in `GraphNeT` was applied to the problem of neutrino oscillations in IceCube, leading to significant improvements in both energy and angular reconstruction in the energy range relevant to oscillation studies (IceCube Collaboration, 2022a). Furthermore, it was shown that the GNN could improve muon vs. neutrino classification and thereby the efficiency and purity of a neutrino sample for such an analysis.

Similarly, improved angular reconstruction has a great impact on, e.g., neutrino point source analyses.

Finally, the fast (order millisecond) reconstruction allows for a whole new type of cosmic alerts at lower energies, which were previously unfeasible. GNN-based reconstruction makes it possible to identify low energy ($<$ 10 TeV) neutrinos and monitor their rate, direction, and energy in real-time. This will enable cosmic neutrino alerts based on such neutrinos for the first time ever, despite a large background of neutrinos that are not of cosmic origin.

Usage

`GraphNeT` comprises a number of modules providing the necessary tools to build workflows from ingesting raw training data in domain-specific formats to deploying trained models in domain-specific reconstruction chains, as illustrated in Figure 1.

The Journal of Open Source Software

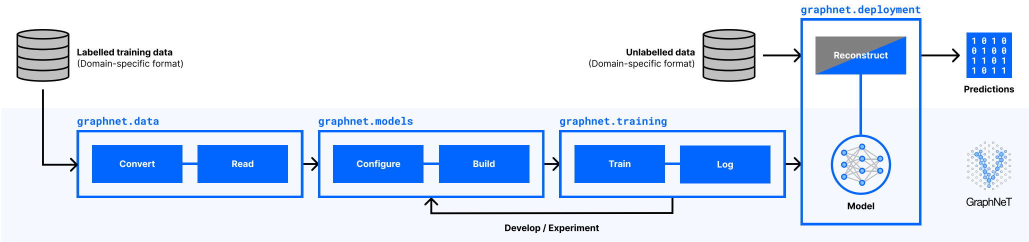

Figure 1: High-level overview of a typical workflow using `GraphNeT`: `graphnet.data` enables converting domain-specific data to industry-standard, intermediate file formats and reading this data; `graphnet.models` allows for configuring and building complex GNN models using simple, physics-oriented components; `graphnet.training` manages model training and experiment logging; and finally, `graphnet.deployment` allows for using trained models for inference in domain-specific reconstruction chains.

`graphnet.models` provides modular components subclassing `torch.nn.Module`, meaning that users only need to import a few existing, purpose-built components and chain them together to form a complete GNN. ML developers can contribute to `GraphNeT` by extending this suite of model components — through new layer types, physics tasks, graph connectivities, etc. — and experiment with optimising these for different reconstruction tasks using experiment tracking.

These models are trained using `graphnet.training` on data prepared using `graphnet.data`, to satisfy the high I/O loads required when training ML models on large batches of events, which domain-specific neutrino physics data formats typically do not allow.

Trained models are deployed to a domain-specific reconstruction chain, yielding predictions, using the components in `graphnet.deployment`. This can either be through model files or container images, making deployment as portable and dependency-free as possible.

By splitting up the GNN development as in Figure 1, `GraphNeT` allows physics users to interface only with high-level building blocks or pre-trained models that can be used directly in their reconstruction chains, while allowing ML developers to continuously improve and expand the framework's capabilities.

Acknowledgements

Andreas Søgaard has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 890778. The work of Rasmus Ørsøe was partly performed in the framework of the PUNCH4NFDI consortium supported by DFG fund "NFDI 39/1", Germany.

References

Aartsen, M. G., & others. (2014). Improvement in Fast Particle Track Reconstruction with Robust Statistics. *Nucl. Instrum. Meth. A*, *736*, 143–149. https://doi.org/10.1016/j.nima.2013.10.074

Abbasi, R., Abdou, Y., Ackermann, M., Adams, J., Aguilar, J. A., Ahlers, M., Altmann, D., Andeen, K., Auffenberg, J., Bai, X., & al., et. (2013). An improved method for measuring muon energy using the truncated mean of dE/dx. *Nucl. Instum. Meth. A*, *703*, 190–198.

AMANDA Collaboration. (2004). Muon track reconstruction and data selection techniques in AMANDA. *Nucl. Instrum. Meth. A*, *524*, 169–194. https://doi.org/10.1016/j.nima.2004.01.065

ANTARES Collaboration. (2011a). A fast algorithm for muon track reconstruction and its application to the ANTARES neutrino telescope. *Astropart. Phys.*, *34*, 652–662. https://doi.org/10.1016/j.astropartphys.2011.01.003

ANTARES Collaboration. (2011b). ANTARES: the first undersea neutrino telescope. *Nucl. Instrum. Meth. A*, *656*, 11–38. https://doi.org/10.1016/j.nima.2011.06.103

ANTARES Collaboration. (2017). An algorithm for the reconstruction of neutrino-induced showers in the ANTARES neutrino telescope. *Astron. J.*, *154*(6), 275. https://doi.org/10.3847/1538-3881/aa9709

Baikal-GVD Collaboration. (2018). Baikal-GVD: status and prospects. *EPJ Web Conf.*, *191*, 01006. https://doi.org/10.1051/epjconf/201819101006

Biewald, L. (2020). *Experiment tracking with weights and biases*. https://www.wandb.com/

Chirkin, D. (2013). Event reconstruction in IceCube based on direct event re-simulation. *33rd International Cosmic Ray Conference*.

Eller, P., Fienberg, A., Weldert, J., Wendel, G., Böser, S., & Cowen, D. F. (2022). *A flexible event reconstruction based on machine learning and likelihood principles*. https://arxiv.org/abs/2208.10166

Falcon, W., & The PyTorch Lightning team. (2019). *PyTorch Lightning* (Version 1.4). https://doi.org/10.5281/zenodo.3828935

Fey, M., & Lenssen, J. E. (2019). *Fast Graph Representation Learning with PyTorch Geometric*. https://github.com/pyg-team/pytorch_geometric

IceCube Collaboration. (2012). The design and performance of IceCube DeepCore. *Astropart. Phys.*, *35*(10), 615–624. https://doi.org/10.1016/j.astropartphys.2012.01.004

IceCube Collaboration. (2014). Energy Reconstruction Methods in the IceCube Neutrino Telescope. *JINST*, *9*, P03009. https://doi.org/10.1088/1748-0221/9/03/P03009

IceCube Collaboration. (2017). The IceCube Neutrino Observatory: Instrumentation and Online Systems. *JINST*, *12*(03), P03012. https://doi.org/10.1088/1748-0221/12/03/P03012

IceCube Collaboration. (2021a). A Convolutional Neural Network based Cascade Reconstruction for the IceCube Neutrino Observatory. *JINST*, *16*, P07041. https://doi.org/10.1088/1748-0221/16/07/P07041

IceCube Collaboration. (2021b). A muon-track reconstruction exploiting stochastic losses for large-scale Cherenkov detectors. *JINST*, *16*(08), P08034. https://doi.org/10.1088/1748-0221/16/08/P08034

IceCube Collaboration. (2022a). *Graph Neural Networks for Low-Energy Event Classification & Reconstruction in IceCube*. https://arxiv.org/abs/2209.03042

IceCube Collaboration. (2022b). *Low Energy Event Reconstruction in IceCube DeepCore*. https://arxiv.org/abs/2203.02303

IceCube-Gen2 Collaboration. (2017). PINGU: A Vision for Neutrino and Particle Physics at the South Pole. *J. Phys. G*, *44*(5), 054006. https://doi.org/10.1088/1361-6471/44/5/054006

IceCube-Gen2 Collaboration. (2021). IceCube-Gen2: the window to the extreme Universe. *J. Phys. G*, *48*(6), 060501. https://doi.org/10.1088/1361-6471/abbd48

IceCube-PINGU Collaboration. (2014). *Letter of Intent: The Precision IceCube Next Generation Upgrade (PINGU)*. https://arxiv.org/abs/1401.2046

KM3NeT Collaboration. (2016). Letter of intent for KM3NeT 2.0. *J. Phys. G*, *43*(8), 084001. https://doi.org/10.1088/0954-3899/43/8/084001

KM3NeT Collaboration. (2020). Event reconstruction for KM3NeT/ORCA using convolutional neural networks. *JINST*, *15*(10), P10005. https://doi.org/10.1088/1748-0221/15/10/P10005

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. https://doi.org/10.48550/arXiv.1912.01703

Pivarski, J., Escott, C., Smith, N., Hedges, M., Proffitt, M., Escott, C., Nandi, J., Rembser, J., bfis, benkrikler, Gray, L., Davis, D., Schreiner, H., Nollde, Fackeldey, P., & Das, P. (2020). *Scikit-hep/awkward-array: 0.13.0* (Version 0.13.0). Zenodo. https://doi.org/10.5281/zenodo.3952674

P-ONE Collaboration. (2020). The Pacific Ocean Neutrino Experiment. *Nature Astron.*, *4*(10), 913–915. https://doi.org/10.1038/s41550-020-1182-4

Søgaard, A., F. Ørsøe, R., Holm, M., Bozianu, L., Rosted, A., C. Petersen, T., Endrup Iversen, K., Hermansen, A., Guggenmos, T., Andresen, P., Ha Minh, M., Neste, L., Holmes, M., Pontén, A., Leonard DeHolton, K., & Eller, P. (2023). *GraphNeT* (Version 1.0.0). https://doi.org/10.5281/zenodo.6720188