

Graph Neural Networks for low-energy event classification & reconstruction in IceCube

The IceCube collaboration

R. Abbasi,¹⁶ M. Ackermann,⁶² J. Adams,¹⁷ N. Aggarwal,²⁴ J.A. Aguilar,¹¹ M. Ahlers,²¹ M. Ahrens,⁵² J.M. Alameddine,²² A.A. Alves Jr.,³⁰ N.M. Amin,⁴² K. Andeen,⁴⁰ T. Anderson,^{58,59} G. Anton,²⁵ C. Argüelles,¹³ Y. Ashida,³⁸ S. Athanasiadou,⁶² S. Axani,¹⁴ X. Bai,⁴⁸ A. Balagopal V.,³⁸ M. Baricevic,³⁸ S.W. Barwick,²⁹ V. Basu,³⁸ R. Bay,⁷ J.J. Beatty,^{19,20} K.-H. Becker,⁶¹ J. Becker Tjus,¹⁰ J. Beise,⁶⁰ C. Bellenghi,²⁶ S. Benda,³⁸ S. BenZvi,⁵⁰ D. Berley,¹⁸ E. Bernardini,⁴⁶ D.Z. Besson,³³ G. Binder,^{7,8} D. Bindig,⁶¹ E. Blaufuss,¹⁸ S. Blot,⁶² F. Bontempo,³⁰ J.Y. Book,¹³ J. Borowka,⁶³ C. Boscolo Meneguolo,⁴⁶ S. Böser,³⁹ O. Botner,⁶⁰ J. Böttcher,⁶³ E. Bourbeau,²¹ J. Braun,³⁸ B. Brinson,⁵ J. Brostean-Kaiser,⁶² R.T. Burley,¹ R.S. Busse,⁴¹ M.A. Campana,⁴⁷ E.G. Carnie-Bronca,¹ C. Chen,⁵ Z. Chen,⁵³ D. Chirkin,³⁸ K. Choi,⁵⁴ B.A. Clark,²³ L. Classen,⁴¹ A. Coleman,⁴² G.H. Collin,¹⁴ A. Connolly,^{19,20} J.M. Conrad,¹⁴ P. Coppin,¹² P. Correa,¹² S. Countryman,⁴⁴ D.F. Cowen,^{58,59} R. Cross,⁵⁰ C. Dappen,⁶³ P. Dave,⁵ C. De Clercq,¹² J.J. DeLaunay,⁵⁷ D. Delgado López,¹³ H. Dembinski,⁴² K. Deoskar,⁵² A. Desai,³⁸ P. Desiati,³⁸ K.D. de Vries,¹² G. de Wasseige,³⁵ T. DeYoung,²³ A. Diaz,¹⁴ J.C. Díaz-Vélez,³⁸ M. Dittmer,⁴¹ H. Dujmovic,³⁰ M.A. DuVernois,³⁸ T. Ehrhardt,³⁹ P. Eller,²⁶ R. Engel,^{30,31} H. Erpenbeck,⁶³ J. Evans,¹⁸ P.A. Evenson,⁴² K.L. Fan,¹⁸ A.R. Fazely,⁶ A. Fedynitch,⁵⁶ N. Feigl,⁹ S. Fiedlschuster,²⁵ A.T. Fienberg,⁵⁹ C. Finley,⁵² L. Fischer,⁶² D. Fox,⁵⁸ A. Franckowiak,¹⁰ E. Friedman,¹⁸ A. Fritz,³⁹ P. Fürst,⁶³ T.K. Gaisser,⁴² J. Gallagher,³⁷ E. Ganster,⁶³ A. Garcia,¹³ S. Garrappa,⁶² L. Gerhardt,⁸ A. Ghadimi,⁵⁷ C. Glaser,⁶⁰ T. Glauch,²⁶ T. Glüsenkamp,²⁵ N. Goehlike,³¹ J.G. Gonzalez,⁴² S. Goswami,⁵⁷ D. Grant,²³ S.J. Gray,¹⁸ T. Grégoire,⁵⁹ S. Griswold,⁵⁰ C. Günther,⁶³ P. Gutjahr,²² C. Haack,²⁶ A. Hallgren,⁶⁰ R. Halliday,²³ L. Halve,⁶³ F. Halzen,³⁸ H. Hamdaoui,⁵³ M. Ha Minh,²⁶ K. Hanson,³⁸ J. Hardin,^{14,38} A.A. Harnisch,²³ P. Hatch,³² A. Haungs,³⁰ K. Helbing,⁶¹ J. Hellrung,⁶³ F. Henningsen,²⁶ L. Heuermann,⁶³ S. Hickford,⁶¹ C. Hill,¹⁵ G.C. Hill,¹ K.D. Hoffman,¹⁸ K. Hoshina,^{38,a} W. Hou,³⁰ T. Huber,³⁰ K. Hultqvist,⁵² M. Hünnefeld,²² R. Hussain,³⁸ K. Hyman,²² S. In,⁵⁴ N. Iovine,¹¹ A. Ishihara,¹⁵ M. Jansson,⁵² G.S. Japaridze,⁴ M. Jeong,⁵⁴ M. Jin,¹³ B.J.P. Jones,³ D. Kang,³⁰ W. Kang,⁵⁴ X. Kang,⁴⁷ A. Kappes,⁴¹ D. Kappesser,³⁹ L. Kardum,²² T. Karg,⁶² M. Karl,²⁶ A. Karle,³⁸ U. Katz,²⁵ M. Kauer,³⁸ J.L. Kelley,³⁸ A. Kheirandish,⁵⁹ K. Kin,¹⁵ J. Kiryluk,⁵³ S.R. Klein,^{7,8} A. Kochocki,²³ R. Koirala,⁴² H. Kolanoski,⁹ T. Kontrimas,²⁶ L. Köpke,³⁹ C. Kopper,²³ D.J. Koskinen,²¹ P. Koundal,³⁰ M. Kovacevich,⁴⁷ M. Kowalski,^{9,62} T. Kozynets,²¹ E. Krupczak,²³ E. Kun,¹⁰ N. Kurahashi,⁴⁷ N. Lad,⁶² C. Lagunas Gualda,⁶² M.J. Larson,¹⁸ F. Lauber,⁶¹ J.P. Lazar,^{13,38} J.W. Lee,⁵⁴ K. Leonard,³⁸ A. Leszczyńska,⁴² M. Lincetto,¹⁰

^aAlso at Earthquake Research Institute, University of Tokyo, Bunkyo, Tokyo 113-0032, Japan.



Q.R. Liu,³⁸ M. Liubarska,²⁴ E. Lohfink,³⁹ C. Love,⁴⁷ C.J. Lozano Mariscal,⁴¹ L. Lu,³⁸
 F. Lucarelli,²⁷ A. Ludwig,^{23,34} W. Luszczak,³⁸ Y. Lyu,^{7,8} W.Y. Ma,⁶² J. Madsen,³⁸
 K.B.M. Mahn,²³ Y. Makino,³⁸ S. Mancina,³⁸ W. Marie Sainte,³⁸ I.C. Mariş,¹¹ S. Marka,⁴⁴
 Z. Marka,⁴⁴ M. Marsee,⁵⁷ I. Martinez-Soler,¹³ R. Maruyama,⁴³ T. McElroy,²⁴ F. McNally,³⁶
 J.V. Mead,²¹ K. Meagher,³⁸ S. Mechbal,⁶² A. Medina,²⁰ M. Meier,¹⁵ S. Meighen-Berger,²⁶
 Y. Merckx,¹² J. Micallef,²³ D. Mockler,¹¹ T. Montaruli,²⁷ R.W. Moore,²⁴ R. Morse,³⁸ M. Moulai,³⁸
 T. Mukherjee,³⁰ R. Naab,⁶² R. Nagai,¹⁵ U. Naumann,⁶¹ A. Nayerhoda,⁴⁶ J. Necker,⁶²
 M. Neumann,⁴¹ H. Niederhausen,²³ M.U. Nisa,²³ S.C. Nowicki,²³ A. Obertacke Pollmann,⁶¹
 M. Oehler,³⁰ B. Oeyen,²⁸ A. Olivas,¹⁸ R. Orsoe,²⁶ J. Osborn,³⁸ E. O'Sullivan,⁶⁰ H. Pandya,⁴²
 D.V. Pankova,⁵⁹ N. Park,³² G.K. Parker,³ E.N. Paudel,⁴² L. Paul,⁴⁰ C. Pérez de los Heros,⁶⁰
 L. Peters,⁶³ T.C. Petersen,²¹ J. Peterson,³⁸ S. Philippen,⁶³ S. Pieper,⁶¹ A. Pizzuto,³⁸
 M. Plum,⁴⁸ Y. Popovych,³⁹ A. Porcelli,²⁸ M. Prado Rodriguez,³⁸ B. Pries,²³
 R. Procter-Murphy,¹⁸ G.T. Przybylski,⁸ C. Raab,¹¹ J. Rack-Helleis,³⁹ M. Rameez,²¹
 K. Rawlins,² Z. Rechav,³⁸ A. Rehman,⁴² P. Reichherzer,¹⁰ G. Renzi,¹¹ E. Resconi,²⁶
 S. Reusch,⁶² W. Rhode,²² M. Richman,⁴⁷ B. Riedel,³⁸ E.J. Roberts,¹ S. Robertson,^{7,8}
 S. Rodan,⁵⁴ G. Roellinghoff,⁵⁴ M. Rongen,³⁹ C. Rott,^{51,54} T. Ruhe,²² L. Ruohan,²⁶
 D. Ryckbosch,²⁸ D. Rysewyk Cantu,²³ I. Safa,^{13,38} J. Saffer,³¹ D. Salazar-Gallegos,²³
 P. Sampathkumar,³⁰ S.E. Sanchez Herrera,²³ A. Sandroock,²² M. Santander,⁵⁷ S. Sarkar,²⁴
 S. Sarkar,⁴⁵ M. Schaufel,⁶³ H. Schieler,³⁰ S. Schindler,²⁵ B. Schlueter,⁴¹ T. Schmidt,¹⁸
 J. Schneider,²⁵ F.G. Schröder,^{30,42} L. Schumacher,²⁶ G. Schwefer,⁶³ S. Sclafani,⁴⁷
 D. Seckel,⁴² S. Seunarine,⁴⁹ A. Sharma,⁶⁰ S. Shefali,³¹ N. Shimizu,¹⁵ M. Silva,³⁸
 B. Skrzypek,¹³ B. Smithers,³ R. Snihur,³⁸ J. Soedingrekso,²² A. Søgaard,²¹ D. Soldin,³¹
 C. Spannfellner,²⁶ G.M. Spiczak,⁴⁹ C. Spiering,⁶² M. Stamatikos,²⁰ T. Stanev,⁴² R. Stein,⁶²
 T. Stezelberger,⁸ T. Stürwald,⁶¹ T. Stuttard,²¹ G.W. Sullivan,¹⁸ I. Taboada,⁵ S. Ter-Antonyan,⁶
 W.G. Thompson,¹³ J. Thwaites,³⁸ S. Tilav,⁴² K. Tollefson,²³ C. Tönnis,⁵⁵ S. Toscano,¹¹
 D. Tosi,³⁸ A. Trettin,⁶² C.F. Tung,⁵ R. Turcotte,³⁰ J.P. Twagirayezu,²³ B. Ty,³⁸
 M.A. Unland Elorrieta,⁴¹ K. Upshaw,⁶ N. Valtonen-Mattila,⁶⁰ J. Vandenbroucke,³⁸
 N. van Eijndhoven,¹² D. Vannerom,¹⁴ J. van Santen,⁶² J. Vara,⁴¹ J. Veitch-Michaelis,³⁸
 S. Verpoest,²⁸ D. Veske,⁴⁴ C. Walck,⁵² W. Wang,³⁸ T.B. Watson,³ C. Weaver,²³ P. Weigel,¹⁴
 A. Weindl,³⁰ J. Weldert,³⁹ C. Wendt,³⁸ J. Werthebach,²² M. Weyrauch,³⁰ N. Whitehorn,^{23,34}
 C.H. Wiebusch,⁶³ N. Willey,²³ D.R. Williams,⁵⁷ M. Wolf,³⁸ G. Wrede,²⁵ J. Wulff,¹⁰ X.W. Xu,⁶
 J.P. Yanez,²⁴ E. Yildizci,³⁸ S. Yoshida,¹⁵ S. Yu,²³ T. Yuan,³⁸ Z. Zhang,⁵³ and P. Zhelnin¹³

¹Department of Physics, University of Adelaide, Adelaide, 5005, Australia

²Department of Physics and Astronomy, University of Alaska Anchorage, 3211 Providence Dr., Anchorage, AK 99508, U.S.A.

³Department of Physics, University of Texas at Arlington, 502 Yates St., Science Hall Rm 108, Box 19059, Arlington, TX 76019, U.S.A.

⁴CTSPS, Clark-Atlanta University, Atlanta, GA 30314, U.S.A.

⁵School of Physics and Center for Relativistic Astrophysics, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.

⁶Department of Physics, Southern University, Baton Rouge, LA 70813, U.S.A.

⁷Department of Physics, University of California, Berkeley, CA 94720, U.S.A.

⁸Lawrence Berkeley National Laboratory, Berkeley, CA 94720, U.S.A.

- ⁹*Institut für Physik, Humboldt-Universität zu Berlin, D-12489 Berlin, Germany*
- ¹⁰*Fakultät für Physik & Astronomie, Ruhr-Universität Bochum, D-44780 Bochum, Germany*
- ¹¹*Université Libre de Bruxelles, Science Faculty CP230, B-1050 Brussels, Belgium*
- ¹²*Vrije Universiteit Brussel (VUB), Dienst ELEM, B-1050 Brussels, Belgium*
- ¹³*Department of Physics and Laboratory for Particle Physics and Cosmology, Harvard University, Cambridge, MA 02138, U.S.A.*
- ¹⁴*Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.*
- ¹⁵*Department of Physics and The International Center for Hadron Astrophysics, Chiba University, Chiba 263-8522, Japan*
- ¹⁶*Department of Physics, Loyola University Chicago, Chicago, IL 60660, U.S.A.*
- ¹⁷*Department of Physics and Astronomy, University of Canterbury, Private Bag 4800, Christchurch, New Zealand*
- ¹⁸*Department of Physics, University of Maryland, College Park, MD 20742, U.S.A.*
- ¹⁹*Department of Astronomy, Ohio State University, Columbus, OH 43210, U.S.A.*
- ²⁰*Department of Physics and Center for Cosmology and Astro-Particle Physics, Ohio State University, Columbus, OH 43210, U.S.A.*
- ²¹*Niels Bohr Institute, University of Copenhagen, DK-2100 Copenhagen, Denmark*
- ²²*Department of Physics, TU Dortmund University, D-44221 Dortmund, Germany*
- ²³*Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, U.S.A.*
- ²⁴*Department of Physics, University of Alberta, Edmonton, Alberta, Canada T6G 2E1*
- ²⁵*Erlangen Centre for Astroparticle Physics, Friedrich-Alexander-Universität Erlangen-Nürnberg, D-91058 Erlangen, Germany*
- ²⁶*Physik-department, Technische Universität München, D-85748 Garching, Germany*
- ²⁷*Département de physique nucléaire et corpusculaire, Université de Genève, CH-1211 Genève, Switzerland*
- ²⁸*Department of Physics and Astronomy, University of Gent, B-9000 Gent, Belgium*
- ²⁹*Department of Physics and Astronomy, University of California, Irvine, CA 92697, U.S.A.*
- ³⁰*Karlsruhe Institute of Technology, Institute for Astroparticle Physics, D-76021 Karlsruhe, Germany*
- ³¹*Karlsruhe Institute of Technology, Institute of Experimental Particle Physics, D-76021 Karlsruhe, Germany*
- ³²*Department of Physics, Engineering Physics, and Astronomy, Queen's University, Kingston, ON K7L 3N6, Canada*
- ³³*Department of Physics and Astronomy, University of Kansas, Lawrence, KS 66045, U.S.A.*
- ³⁴*Department of Physics and Astronomy, UCLA, Los Angeles, CA 90095, U.S.A.*
- ³⁵*Centre for Cosmology, Particle Physics and Phenomenology - CP3, Université catholique de Louvain, Louvain-la-Neuve, Belgium*
- ³⁶*Department of Physics, Mercer University, Macon, GA 31207-0001, U.S.A.*
- ³⁷*Department of Astronomy, University of Wisconsin — Madison, Madison, WI 53706, U.S.A.*
- ³⁸*Department of Physics and Wisconsin IceCube Particle Astrophysics Center, University of Wisconsin — Madison, Madison, WI 53706, U.S.A.*
- ³⁹*Institute of Physics, University of Mainz, Staudinger Weg 7, D-55099 Mainz, Germany*
- ⁴⁰*Department of Physics, Marquette University, Milwaukee, WI, 53201, U.S.A.*
- ⁴¹*Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, D-48149 Münster, Germany*
- ⁴²*Bartol Research Institute and Department of Physics and Astronomy, University of Delaware, Newark, DE 19716, U.S.A.*
- ⁴³*Department of Physics, Yale University, New Haven, CT 06520, U.S.A.*
- ⁴⁴*Columbia Astrophysics and Nevis Laboratories, Columbia University, New York, NY 10027, U.S.A.*

- ⁴⁵*Department of Physics, University of Oxford, Parks Road, Oxford OX1 3PU, U.K.*
- ⁴⁶*Dipartimento di Fisica e Astronomia Galileo Galilei, Università Degli Studi di Padova, 35122 Padova PD, Italy*
- ⁴⁷*Department of Physics, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, U.S.A.*
- ⁴⁸*Physics Department, South Dakota School of Mines and Technology, Rapid City, SD 57701, U.S.A.*
- ⁴⁹*Department of Physics, University of Wisconsin, River Falls, WI 54022, U.S.A.*
- ⁵⁰*Department of Physics and Astronomy, University of Rochester, Rochester, NY 14627, U.S.A.*
- ⁵¹*Department of Physics and Astronomy, University of Utah, Salt Lake City, UT 84112, U.S.A.*
- ⁵²*Oskar Klein Centre and Department of Physics, Stockholm University, SE-10691 Stockholm, Sweden*
- ⁵³*Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11794-3800, U.S.A.*
- ⁵⁴*Department of Physics, Sungkyunkwan University, Suwon 16419, Korea*
- ⁵⁵*Institute of Basic Science, Sungkyunkwan University, Suwon 16419, Korea*
- ⁵⁶*Institute of Physics, Academia Sinica, Taipei, 11529, Taiwan*
- ⁵⁷*Department of Physics and Astronomy, University of Alabama, Tuscaloosa, AL 35487, U.S.A.*
- ⁵⁸*Department of Astronomy and Astrophysics, Pennsylvania State University, University Park, PA 16802, U.S.A.*
- ⁵⁹*Department of Physics, Pennsylvania State University, University Park, PA 16802, U.S.A.*
- ⁶⁰*Department of Physics and Astronomy, Uppsala University, Box 516, S-75120 Uppsala, Sweden*
- ⁶¹*Department of Physics, University of Wuppertal, D-42119 Wuppertal, Germany*
- ⁶²*DESY, D-15738 Zeuthen, Germany*
- ⁶³*III. Physikalisches Institut, RWTH Aachen University, D-52056 Aachen, Germany*

E-mail: analysis@icecube.wisc.edu

ABSTRACT: IceCube, a cubic-kilometer array of optical sensors built to detect atmospheric and astrophysical neutrinos between 1 GeV and 1 PeV, is deployed 1.45 km to 2.45 km below the surface of the ice sheet at the South Pole. The classification and reconstruction of events from the in-ice detectors play a central role in the analysis of data from IceCube. Reconstructing and classifying events is a challenge due to the irregular detector geometry, inhomogeneous scattering and absorption of light in the ice and, below 100 GeV, the relatively low number of signal photons produced per event. To address this challenge, it is possible to represent IceCube events as point cloud graphs and use a Graph Neural Network (GNN) as the classification and reconstruction method. The GNN is capable of distinguishing neutrino events from cosmic-ray backgrounds, classifying different neutrino event types, and reconstructing the deposited energy, direction and interaction vertex. Based on simulation, we provide a comparison in the 1 GeV–100 GeV energy range to the current state-of-the-art maximum likelihood techniques used in current IceCube analyses, including the effects of known systematic uncertainties. For neutrino event classification, the GNN increases the signal efficiency by 18% at a fixed background rate, compared to current IceCube methods. Alternatively, the GNN offers a reduction of the background (i.e. false positive) rate by over a factor 8 (to below half a percent) at a fixed signal efficiency. For the reconstruction of energy, direction, and interaction vertex, the resolution improves by an average of 13%–20% compared to current maximum likelihood techniques in the energy range of 1 GeV–30 GeV. The GNN, when run on a GPU, is capable of processing IceCube events at a rate nearly double of the median IceCube trigger rate of 2.7 kHz, which opens the possibility of using low energy neutrinos in online searches for transient events.

KEYWORDS: Analysis and statistical methods; Data analysis; Neutrino detectors; Particle identification methods

ARXIV EPRINT: [2209.03042](https://arxiv.org/abs/2209.03042)

2022 JINST 17 P11003

Contents

1	Introduction	1
1.1	The IceCube detector	1
1.2	The challenge of IceCube event classification and reconstruction	3
1.2.1	Traditional reconstruction methods	4
1.2.2	Machine-learning-based reconstruction methods	5
2	Graph Neural Networks applied to IceCube data	5
2.1	Preprocessing of IceCube events	6
2.2	Model architecture	6
2.3	Training configuration and loss functions	8
3	Performance on low-energy neutrino events	9
3.1	Selected datasets	10
3.2	Event classification performance	11
3.3	Event reconstruction performance of DYNEDGE	11
3.4	Runtime performance	15
4	Robustness test	15
4.1	Perturbation of input variables	15
4.2	Variations in ice properties and module acceptance	17
4.2.1	Classification	18
4.2.2	Reconstruction	18
5	Summary and conclusions	22

1 Introduction

1.1 The IceCube detector

The IceCube Neutrino Observatory, located at the geographic South Pole, consists of a cubic kilometer of ice instrumented with 5,160 Digital Optical Modules (DOMs) [1] on 86 strings, placed at depths between 1450 m and 2450 m. The main detector array consists of 78 strings arranged in a roughly hexagonal array, with an average horizontal distance of 125 m between neighboring strings [2] (see figure 1). Each string supports 60 DOMs separated vertically by 17 m. Each DOM contains a 10'' Photomultiplier Tube (PMT) facing downwards. Around the center string in the deepest part of the array where the optical transparency of the ice is highest, modules on 8 additional strings have been installed. This volume, named “DeepCore” [3], has an increased spatial density of DOMs and features PMTs with an enhanced quantum efficiency (QE) compared to the main array. The IceCube Observatory is constructed to detect neutrino interactions spanning the energy

range of a few GeV to several PeV, with the purpose of exploring properties of both the cosmos and fundamental properties of neutrinos [3, 4].

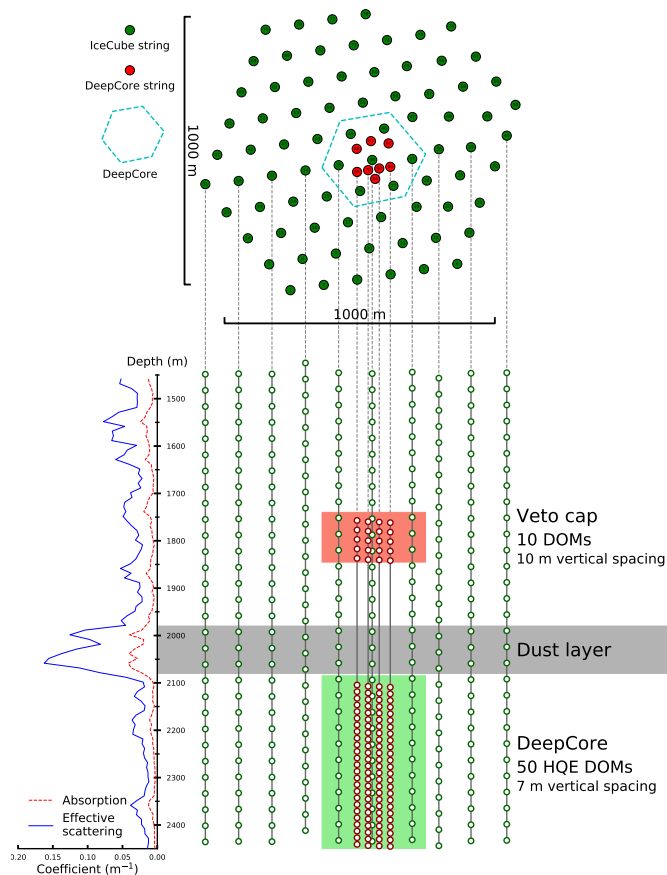


Figure 1. Top and side view of IceCube detector [3]. Ice properties as a function of depth is shown on the left. The in-ice dust layer is marked in gray. The dust layer is a layer in the ice with dust impurities and therefore reduced optical qualities. DeepCore is placed under the dust layer and shown in green. High Quantum Efficiency (HQE) DOMs are placed under the dust layer in the DeepCore region.

Charged particles resulting from neutrino interactions in the ice will emit Cherenkov radiation detectable by IceCube. The number of detected photons is many orders of magnitude lower than those emitted. For the neutrinos detectable by IceCube, the signal in a neutrino event may range from a low of a few to a high of order 100,000 detected photoelectrons. In the energy range this work is focusing on — from a few GeV up to around 100 GeV — a typical interaction deposits about 49 GeV of energy and has just 17 detected [5] photoelectrons. This signal, however, is interspersed with noise stemming from, for example, radioactive decays in the DOMs. Event candidates are read out from the detector at a trigger rate of around 2.7 kHz [6]. These recorded events are dominated by triggers due to downgoing atmospheric muons, followed by first random triggers caused by noise pulses, second by atmospheric neutrinos at a rate less than 10 mHz [3].

1.2 The challenge of IceCube event classification and reconstruction

Event reconstruction can be framed as a problem of parameter inference. Given a set of detector observations, the reconstruction aims to infer the physics properties of a (neutrino) interaction. A reconstruction algorithm defines and implements this process of taking input data and returning parameter estimators.

Parameters of interest and categorization of events. The parameters of interest estimated in the reconstruction vary by application. The deposited energy, the direction of the neutrino candidate, and the interaction vertex are of central relevance to many IceCube physics analyses. Events are also categorized into multiple morphological classes, of which only two are relevant for this work because the vast majority of the sample is below 100 GeV. These two event classes serve as proxies for the underlying neutrino flavor and interaction type. “Track-like” events are proxies for ν_μ charged-current (CC) interactions from atmospheric and astrophysical neutrinos. These events contain a muon that can travel a long distance inside the detector while emitting Cherenkov radiation, producing a signature that looks like a track going through the ice. However, track-like signatures can also be produced by atmospheric muons from cosmic rays, and from the 17% of ν_τ CC interactions that produce τ leptons that decay into muons [7], but these are not considered true tracks for the purposes of this work. The other class comprises “cascade-like” events, containing everything that is not described by the track-like class. These events consist of electromagnetic and hadronic particle showers, such as those produced in $\nu_{e,\tau,\mu}$ neutral-current (NC) and $\nu_{e,\tau}$ CC interactions. For brevity, we will refer to the classification of track (\mathcal{T}) and cascade (\mathcal{C}) morphologies as “ \mathcal{T}/\mathcal{C} ”. Another classification task is the discrimination of neutrino events from atmospheric muons events, which will be referred to as “ ν/μ ”.

Input Data. Data from the IceCube PMTs is digitized with waveform digitizers which record the majority of the PMT signals from neutrino interactions. These waveforms undergo an unfolding process [8] that attempts to extract estimated photon arrival times and the corresponding charge of individual photoelectrons each of which is a so-called “pulse”. The pulses form the detector response to an interaction and make up a time series. Each element in this sequence corresponds to the time t at which the PMT readout indicates a measured pulse with charge q . Most pulses in the energy regime considered in this work stem from single photons. These result in pulse charges close to one photoelectron, with variation in the charge due to the stochastic nature of the PMT amplification process. The detection of multiple, nearly instantaneous photons results in higher per-pulse charges. The PMTs themselves are located at fixed DOM positions (D_{xyz}) and have an empirically determined QE. The DOMs deployed in DeepCore have an efficiency that is roughly a factor of 1.35 times the QE of standard DOMs [3]. These six variables of the time sequence are summarized in table 1 and form the input data for the reconstruction.

The amount of Cherenkov radiation produced in an event depends primarily on the energy of the interacting particle, and less energetic neutrinos often lead to a reduced amount of Cherenkov light. Consequently, the number of pulses for an event, n_{pulses} , is highly dependent on the event itself, and makes low energy neutrinos particularly difficult to identify and reconstruct. In this study, we focus on neutrinos with energy less than 1 TeV. The range of 1–30 GeV is of particular importance to the study of atmospheric neutrino oscillations [9, 10]. For typical events in this energy range,

Table 1. Node features in graph representation of neutrino events. The units shown here are before the preprocessing mentioned in section 2.1.

Feature	Description	Unit
D_{xyz}	Position of DOMs in IceCube coordinates	m
t	Pulse time relative to trigger time	ns
q	Charge of a pulse	P.E.
QE	Quantum efficiency of PMT	-

n_{pulses} can range from below ten to several hundred after noise hit removal. Because some pulses in an event are due to noise pulses, low-energy neutrino events often suffer from a relatively poor signal-to-noise ratio. In addition, at low energies the track events can be so short that they cannot be easily distinguished from cascade-like events. While the irregular geometry of IceCube help distinguish events that would otherwise leave identical detector responses in a regular geometry configuration, it provides an additional layer of complexity to the development of reconstruction algorithms. In addition, reconstruction is further complicated by varying ice properties as a function of x, y, z , position in the IceCube array [11]. The z -dependence of the ice absorption and effective scattering is shown to the left in figure 1.

1.2.1 Traditional reconstruction methods

Maximum likelihood estimation is a standard technique for parameter inference. The key ingredients are the likelihood itself and a maximization strategy. The exact reconstruction likelihood for IceCube events is, however, intractable and can only be approximated. In IceCube, event reconstructions range in complexity from simple analytical approximations of likelihoods [12] to highly sophisticated detector response functions based on photon ray tracing [5].

A major challenge in modeling the detector response is including the right amount of detail of an event’s Cherenkov light emission profile, as well as the inhomogeneous optical properties of the ice, which is visualized on the left side of figure 1. Both affect the expected photon pattern in the PMTs. Analytic approximations, which are fast but inaccurate, are predominantly used as first guesses for both online and offline processing [12]. More sophisticated reconstruction techniques come with a higher computational cost and can only be applied to a subset of the events. For the neutrino energies considered in this work, (i.e. low-energy events between 1 GeV–1000 GeV) such likelihood-based reconstructions were used, for example, in the analyses of refs. [9, 10]. We will use the state-of-the-art IceCube low-energy reconstruction algorithm `RETRO` [5] as a benchmark for our work. On average, `RETRO` requires around 40 seconds to reconstruct one event on a single CPU core, which is one of the main shortcomings of the method. Another limitation is its use of simplifying approximations, such as the assumed uniformity in the azimuthal response of the IceCube DOMs, as well as the assumption that ice properties change as a function of depth only.

The IceCube Upgrade [13] will augment IceCube’s detection capabilities by adding additional detector strings featuring new DOM types: the “mDOM” and “DEgg”. The mDOMs carry 24 3” PMTs providing almost homogeneous angular coverage, while the DEggs carry two 8” PMTs, one facing up and one down [14]. Adjusting `RETRO` to work with the IceCube Upgrade is difficult

while keeping memory usage and computing time at a reasonable level, due to assumed symmetries being broken and the increase of different module types.

1.2.2 Machine-learning-based reconstruction methods

An alternative method to maximum likelihood estimation is regression with Neural Networks (NNs) [15]. Instead of approximating the likelihood and traversing its parameter space with an optimization algorithm, a regression algorithm returns parameter estimates directly. These regression models are trained by minimizing a defined loss function. Since an optimal regression algorithm may be highly nonlinear, artificial NNs can offer a viable solution. A regression algorithm needs to map the ragged input data of shape $[n_{\text{pulses}}, 6]$, where the number of columns refers to the six node features (D_x, D_y, D_z, t, q, QE) described in table 1, onto an output of shape $[1, D]$ estimating event level truth information, where D is the number of parameters we are interested in. The spatio-temporal nature of IceCube data, with the addition of varying sequence lengths, makes it difficult to map event reconstruction in IceCube to common machine learning techniques. Previously published IceCube machine learning methods embed events into pseudo-images and perform regression using a Convolutional Neural Network (CNN) [16]. This class of algorithms is effective at identifying local features in data — i.e., CNNs efficiently group similar data points in the compressed “latent space” representation of the input images. However, embedding IceCube events into images is a lossy operation aggregating pulse information into per-DOM summary statistics, a process that severely degrades the information available in low-energy events. Due to this reason, the CNN reconstruction method is mainly used at energies higher than what is considered in our work, but an adaptation for our energy range exists [17] [18]. An alternative NN architecture employing a Graph Neural Network (GNN) [19] approach is used in [20], but focuses solely on ν/μ classification.

We propose a general reconstruction method based on GNNs that can be applied to the entire energy range of IceCube, is compatible with the IceCube Upgrade, and can reconstruct events studied in this work at speeds fast enough to run in real time online processing at the South Pole. We consider a set of reconstruction attributes: ν/μ classification, \mathcal{T}/C classification, deposited energy E , zenith and azimuth angles (θ, ϕ) , and x, y , and z coordinates of the interaction vertex, denoted by V_{xyz} . We will benchmark our reconstruction method on a simulated low-energy IceCube sample used for atmospheric neutrino oscillation studies.

2 Graph Neural Networks applied to IceCube data

Generally, GNNs are NNs that work on graph representations of data. A graph consists of *nodes* interconnected by *edges*. Nodes are associated with data, and the edges specify the relationship among the nodes. By adopting graphs as the input data structure, the idea of convolution generalizes from the application of filters on the rigid structure of grids to abstract mathematical operators that utilize the interconnection of nodes in its computation. This allows such models to naturally incorporate an irregular geometry directly in the edge specification of the graph, without imposing artificial constraints. For this reason, GNNs are a natural choice of machine learning paradigm for problems with irregular geometry, such as IceCube event reconstruction.

We choose to represent each event by a single graph. Each observed pulse is represented by a node in the graph and contains the per-DOM information shown in table 1. Each node in the graph is

connected to its 8 nearest neighbors based on the Euclidean distance, and for this reason we consider the interconnectivity of the nodes in the graphs to be spatial [21].

2.1 Preprocessing of IceCube events

The variables listed in table 1 come in different units and can span over different ranges, for example the relative positions of triggered DOMs can take values between tens of meters to > 1 km; or the times spanned by pulses may range from tens of nanoseconds to several microseconds. While neural networks can in theory process data in any range of real numbers, the complexity of the loss landscape that a model navigates during training is highly dependent on the relative scale of the input data. In addition, distributions not centered around zero can lead to a slower convergence time [22]. Each input variable x is therefore transformed into \bar{x} using

$$\bar{x} = \frac{x - P_{50\text{th}}(x)}{P_{75\text{th}}(x) - P_{25\text{th}}(x)}, \quad (2.1)$$

where $P_{i\text{th}}(x)$ is the i -th percentile of the distribution of input feature x . This transformation brings the input variables into roughly similar orders of magnitude, gives a median of zero and makes them unitless.

2.2 Model architecture

Our GNN implementation, henceforth referred to as `DYNEDGE`, is a general purpose reconstruction and classification algorithm that is applied directly to the event pulses from the IceCube detector. Therefore, the algorithm does not rely on the pulses to be aggregated into summary statistics like [16]. Instead, `DYNEDGE` uses graph learning to extract features from the pulses, making the mapping of measurements to prediction a fully learnable task. Additionally, `DYNEDGE` learns the optimal edges between nodes and can be applied to events with any number of pulses. `DYNEDGE` is implemented using GraphNeT [23], a software framework for graph neural networks in neutrino telescopes built using PyTorch Geometric [24].

`DYNEDGE` uses a convolutional operator *EdgeConv* [25], developed to act on point cloud graphs in computer vision segmentation analysis. For every node n_j with node features x_j , the operator convolves x_j via local neighborhood of n_j as

$$\tilde{x}_j = \sum_{i=1}^{N_{\text{neighbors}}} \text{MLP}(x_j, x_j - x_i), \quad (2.2)$$

where \tilde{x}_j denotes the convolved node features of n_j , and the Multilayer Perceptron (MLP) takes as input the unconvolved node features of n_j and the pairwise difference between the unconvolved node features of n_j and its i -th neighbor. Thus, the connectivity of the node in question serves as a specification of which neighboring nodes contribute to the convolution operation. A full convolution pass of the input graph consists of repeating Equation (2.2) for every node in the graph. When compared to convolutions as understood from CNNs, Equation (2.2) corresponds to a convolutional operator where the kernel size is analogous to $N_{\text{neighbors}}$ and the ‘‘pixels’’ on which the kernel acts are defined by the edges, instead of the position of the kernel.

The full architecture of `DYNEDGE` is shown in figure 2. First, the following 5 global statistics are calculated from the input graph: node homophily ratio [26] of D_{xyz} and t , and number of pulses

in the graph. The homophily ratio is the ratio of connected node pairs that share the same node feature, and thus a number between 0 and 1. The homophily ratio of D_{xyz} indicates what fraction of connected pulses originate from the same PMT. For a graph where all nodes are connected to each other, a value of 0.5 indicates that half the pulses in the event are same-PMT pulses. As such, the homophily ratio quantifies how many of the pulses originate from the same PMT and how many of the pulses happened at the same time. Second, the input graph is propagated through 4 different EdgeConv blocks such that the output from one flows into the next. As illustrated in the bottom right corner of figure 2, the EdgeConv block is initialized with a k -nearest neighbors (k -nn) computation that redefines the edges of the input graph. This step lets DYNEDGE calculate the 8 nearest Euclidean neighbors and means that the first convolution block connects the nodes in true xyz -space and the subsequent convolution blocks connects the nodes in an increasingly abstract latent space. By letting the subsequent convolutional blocks interpret the convolved xyz -coordinates from the past block, DYNEDGE is allowed to connect the nodes arbitrarily in each convolutional step, which effectively lets DYNEDGE learn the optimal neighborhoods of the event for a specific task given the 8-nearest-neighbors constraint.

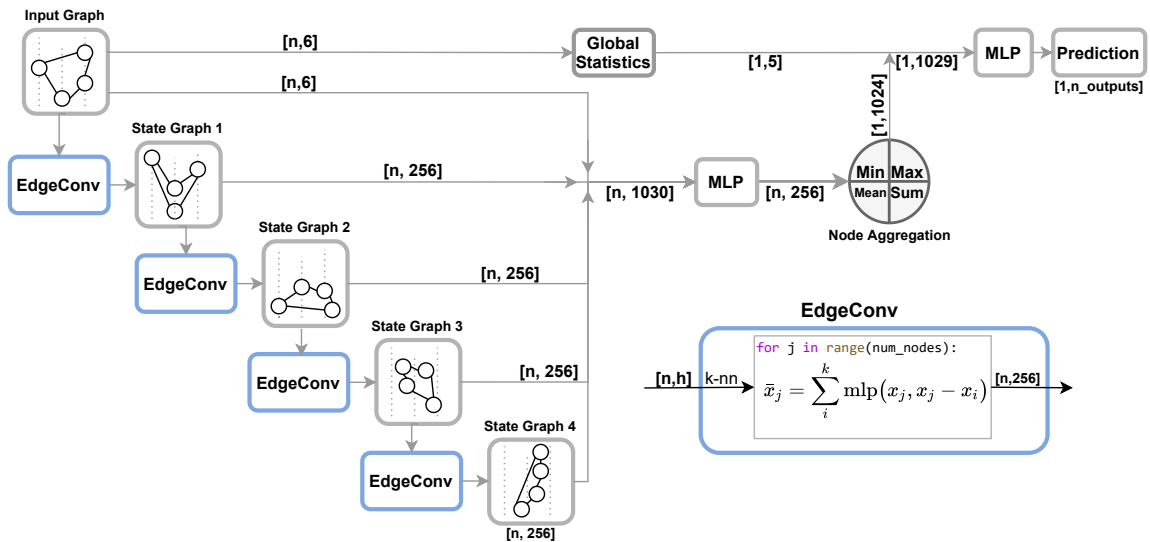


Figure 2. A diagram of the architecture of DYNEDGE. The “Input Graph” is a toy illustration of the input, and the subsequent “State Graphs” illustrates how the node position and connectivity change after convolutions by EdgeConv. An illustration of the inner workings of the EdgeConv block is provided to the right, where h represents an arbitrary number of columns. The number of pulses is represented by n .

Third, the input graph and each state graph are concatenated together into a $[n, 1030]$ -dimensional array that is passed through a fully connected MLP block containing two layers. The block maps the $[n, 1030]$ -dimensional array into a $[n, 256]$ -dimensional array. This array is aggregated node-wise into summary statistics in four parallel ways; *mean*, *min*, *max*, and *sum*, each corresponding to a many-to-one projection of the form $f : [n, 256] \rightarrow [1, 256]$.¹ Aggregations are concatenated together to minimize information loss, which produces an array with dimension $[1, 4 \cdot 256]$ that is concatenated

¹While more sophisticated pooling methods were tested, none improved upon this choice.

together with the initially calculated 5 global statistics, producing a $[1, 1029]$ -dimensional input array to the subsequent 2-layer MLP block that makes the final prediction by mapping the $[1, 1029]$ -dimensional array to a $[1, n_{\text{outputs}}]$ -dimensional output. This node aggregation scheme removes the need for zero-padding of the input data, and allows the model to function on any number of pulses.

The architecture shown in figure 2 is the result of multiple iterations of architecture tests. We find that increasing the number of convolutional layers leads to identical performance but at increased training time, whereas a decrease in the number of convolutional layers leads to a noticeably worse performance. In addition, a wide variety of data-driven pooling operations have been tested, but none improved upon the many-to-one projections shown in figure 2. Also, hyperparameters such as the number of nearest neighbors used in the k-nearest-neighbors computation have been subject to tuning. We find that for k larger than 8, the convolutions become too coarse to learn local features, whereas for k smaller than 8 the convolutions become too fine to be globally descriptive. Also, the internal dimensions of the data referenced both in text and in figure 2 have been subject to hyperparameter optimization. These dimensions effectively set the number of learnable parameters in the model. We find that a significant reduction in learnable parameters yields an under-parameterized model that cannot learn the complicated relationships between data and task. A significant increase in learnable parameters increases training time and yields similar performance as reported for the current configuration.

2.3 Training configuration and loss functions

A DYNEDGE network is trained for each of the reconstruction variables: deposited energy E , zenith and azimuth angles (θ , ϕ), the interaction vertex V_{xyz} , ν/μ classification and \mathcal{T}/C classification. This totals 6 independently trained models. The difference between each model is the choice of loss function, number of outputs, and training selection.

Each model is trained with a batch size of 1024, using the ADAM [27] optimizer. The batch size indicates the number of events in each sub sample of the training set, on which the gradients of the loss are calculated. A custom piece-wise-linear implementation of a One-Cycle [28] learning rate schedule with a warm-up period [29] is used. The scheduler increases the global learning rate linearly from 10^{-5} to 10^{-3} during the first 50% of iterations in the first epoch, and thereafter linearly decreases the learning rate to 10^{-5} during the remaining iterations in the 30 epoch training budget. To counteract overfitting, early stopping [30] is implemented with a patience of 5 epochs.

For all classification tasks, the Binary Cross Entropy [31] loss is used, since we only consider two categories: neutrino or muon events, and tracks or cascades. However, for each regression task, a specific loss function is chosen.

For regression of deposited energy, a LogCosh [32] function is used

$$\text{Loss}(E) = \ln(\cosh(R_E)), \quad (2.3)$$

applied to the residual $R_E = \log_{10}(E_{\text{reco}}/\text{GeV}) - \log_{10}(E_{\text{true}}/\text{GeV})$. The embedding $f : E_{\text{true}} \rightarrow \log_{10}(E_{\text{true}}/\text{GeV})$ is added to account for the large range that the deposited energy spans, and $\log_{10}(E_{\text{reco}}/\text{GeV})$ is the model's prediction of the embedded, deposited energy. LogCosh is symmetric around zero and, therefore, punishes over- and under-estimation equally. When compared to more conventional choices such as mean-squared error (MSE), LogCosh offers a steadier gradient around zero and is approximately linear for large residuals, both beneficial to the training process.

Table 2. Targets of the GNN-based classification and reconstruction algorithm, based on the truth values of the event simulation. Includes definitions of residual distributions for regression targets.

Targets	Description	Residual Definition
ν/μ	Classification of neutrino vs. muon events	–
E	Deposited energy of neutrino interaction	$R_E = \log_{10}(E_{\text{reco}}) - \log_{10}(E_{\text{true}})$
θ, ϕ	Zenith and azimuth angles of neutrino	$R_{\text{angle}} = \text{angle}_{\text{reco}} - \text{angle}_{\text{true}}$
\vec{r}	Direction vector of neutrino	$R_{\vec{r}} = \arccos \frac{\vec{r}_{\text{reco}} \cdot \vec{r}_{\text{true}}}{ \vec{r}_{\text{reco}} \vec{r}_{\text{true}} }$
V_{xyz}	Vertex position of neutrino interaction	$R_{V_{xyz}} = \vec{P}_{\text{reco}} - \vec{P}_{\text{true}} $
\mathcal{T}/C	Classification into tracks and cascades	–

For angular regression, a von Mises-Fisher [33] Sine-Cosine loss is used, where the true angle ϕ is embedded into a 2D vector space using $f : \phi \rightarrow [\sin \phi \cos \phi]$. DYNEDGE is then tasked with predicting this embedded vector together with an uncertainty k . The $d = 2$ von Mises-Fisher distribution, where d is the dimension of the unit vectors, is given by $p(\bar{x}|\bar{u}, k) = C_2(k) \exp(k\bar{u} \cdot \bar{x}) = C_2 \exp(k \cos \Delta\phi)$ where \bar{x} is the predicted embedded vector; \bar{u} is the embedding of the true angle; k resembles $1/\sigma^2$ of a normal distribution; and C_2 is a normalization constant written in terms of modified Bessel functions. The $d = 2$ von Mises-Fisher distribution describes a probability distribution on a 1-sphere embedded in \mathbb{R}^2 and bears resemblance to loss functions such as $1 - \cos(\Delta\phi)$ but with the added functionality of uncertainty estimation via k . Note that $\Delta\phi$ is chosen to be the angle between the predicted and the true embedding vector.² The loss function is created by taking the negative log of $p(\bar{x}|\bar{u}, k)$:

$$\text{Loss}(\phi) = -\ln p(\bar{x}|\bar{u}, k) = -\ln(C_2(k)) - \ln(k/4\pi) + k + \ln(1 - e^{-2k}) - k \cos(\Delta\phi). \quad (2.4)$$

After zenith and azimuth angles are regressed individually, the direction is produced by transforming zenith and azimuth into a direction vector $\vec{r}_{\text{reco}} \in \mathbb{R}^3$.

The true interaction vertex is embedded using $f : (x, y, z) \rightarrow \left(\frac{x}{|\max(x)|}, \frac{y}{|\max(y)|}, \frac{z}{|\max(z)|} \right)$ for the same reasons mentioned in section 2.1. DYNEDGE then predicts the embedded interaction vertex vector, and the loss function is the Euclidean distance between the true (\vec{P}_{true}) and reconstructed (\vec{P}_{reco}) embedding vectors.

3 Performance on low-energy neutrino events

In this section, we quantify the performance of our proposed algorithm based on simulated data in the energy range of 1 GeV–1000 GeV, where the majority of selected events are below 100 GeV (see figure 4), and compare it with the state-of-the-art RETRO algorithm. When possible, we provide comparisons for tracks ($\nu_\mu CC$) and cascades (all other neutrino interactions) separately. At the end of the section, a short examination of the runtime performance will be provided.

²A more intuitive approach such as a $d=3$ von Mises-Fisher, where $\Delta\phi$ is chosen to be the angular difference between predicted and true direction vectors in \mathbb{R}^3 , was found to lead to suboptimal results because the azimuthal component of the loss is too dominant at lower energies. Substantial improvement in zenith reconstruction is gained by estimating the two angles separately.

3.1 Selected datasets

The IceCube simulation used for training and testing the `DYNEDGE` classifications and reconstructions is borrowed from the collaboration’s simulation for neutrino oscillation analyses. The dataset and selection process is similar to the ones described in [10], which is also used in [5]. The interactions were simulated with GENIE [34], and simulations of the propagation of secondaries, particle showers, and the propagation of Cherenkov light are the same as used in [10]. The event selection aims to provide a clean neutrino sample in the DeepCore region of IceCube by removing pure noise events, atmospheric muon events and by applying containment cuts. `RETRO` is run at the late stages of the event selection (level 7 in [10]) because of its high computational cost. We have chosen this late stage of the event selection as our dataset because it allows for a direct comparison with `RETRO`.

The resulting selection totals approximately 8.3 million simulated neutrino and muon events, and because the event selection removes virtually all pure noise events, these have been omitted from this work. Once weighted to a physical spectrum, the data sample contains less than 5% atmospheric muons [10] and the neutrino sample consists of mostly contained interactions below 100 GeV. Distributions of a few event level variables of the selected events are shown in figure 4. From the 8.3 million event sample, we construct three task specific datasets, summarized in table 3.

Table 3. Selected datasets for the reconstruction and classification tasks for `DYNEDGE`. Testing sets are defined as the data on which no back-propagation has been made, and therefore includes the validation set.

Task	Total	ν_e^{NC} (%)	ν_e^{CC} (%)	ν_μ^{NC} (%)	ν_μ^{CC} (%)	ν_τ^{NC} (%)	ν_τ^{CC} (%)	μ (%)
ν/μ (Train)	215k	1.6	15.0	1.6	15.0	4.4	12.3	50.1
ν/μ (Test)	106k	1.6	15.1	1.5	15.3	4.5	12.2	49.8
Reconstruction (Train)	3.76M	3.2	30.0	3.1	30.3	9.0	24.4	
Reconstruction (Test)	4.37M	1.4	12.8	6.6	64.3	4.0	11.0	
\mathcal{T}/C (Train)	731k	16.7	16.7	16.7	49.9			
\mathcal{T}/C (Test)	7.4M	0.8	21.2	3.9	48.2	7.0	18.9	

Because the event selection aims to produce a pure neutrino sample, the raw count of muon events in the 8.2 million sample is less than 200k. In addition, there are significantly more cascade events than tracks (ν_μ CC), making the full data set class-imbalanced. This work under-samples [35] the full data set into three task-specific training sets with corresponding test sets to counteract the imbalance between the three classes: tracks, cascades and muons.

The \mathcal{T}/C training data set contains an even amount of ν_μ CC events (considered “tracks”) and CC ν_e + NC events (considered “cascades”), and due to the high number of cascade events, the training sample consists of only 731k events. The ν_τ events are omitted from the training sample (but included in the test sample), since 17% of ν_τ^{CC} events produce track-like signatures that may confuse the model during training with a track-like signature but a cascade-like label. The corresponding test set for \mathcal{T}/C constitutes the remaining neutrino events, mostly consisting of cascades. For the ν/μ classification task, the training dataset is chosen such that there are an equal amount of neutrino and muons events. Because of the few muon events, this dataset is the smallest, and the neutrinos have been sampled such that there is an even amount of each flavor.

For reconstruction, a dataset with equal amounts of neutrino flavors has been chosen. On this dataset, energy, zenith, azimuth, and interaction vertex are regressed. In order to keep the balance between tracks and cascades, one may most naturally choose the dataset from the \mathcal{T}/C task, because it is balanced between the two event classes, but this selection yields significantly lower statistics. It was found that the high statistics choice provided general improvements compared to reconstructions from `DYNEDGE` trained purely on the balanced \mathcal{T}/C set.

The classification results are available in figure 3 and reconstruction results are shown in figure 5. The normalized distributions of regression targets on test and training sets from table 2 are plotted in figure 4 for each event selection in table 3. The \mathcal{T}/C and Reconstruction selections are similar in target distributions, whereas the ν/μ event selection have additional artifacts due to the presence of muons.

3.2 Event classification performance

The performance of the `DYNEDGE` classifiers and the currently-used Boosted Decision Tree (BDT) methods is characterized using Receiver Operating Characteristics (ROC) [36] curves, from which the Area Under the Curve (AUC) [37] is calculated. The ν/μ classification task is used in event selection, and for this reason a threshold for the classification scores, shown in the bottom left plot of figure 3, is used to make a decision on whether an event is labeled a neutrino or a muon. A typical threshold in classification score is indicated in red in the left panel of figure 3. Intersection points between the red line and the ROC curves represents the corresponding False Positive Rate (FPR) and True Positive Rate (TPR) at this choice of selection.

By comparing the TPR of intersection points, one can deduce that `DYNEDGE` offers increased signal efficiency by 18% relative to the BDT at a FPR of 0.025. Alternatively, the FPR can be reduced by over a factor of 8 from 2.5% to 0.3% relative to the BDT at a fixed TPR of 0.6 for the ν/μ classification task. From the ROC curve for the \mathcal{T}/C classification task, `DYNEDGE` offers an increased AUC score of about 6%. A threshold for the \mathcal{T}/C classification task is not shown because the \mathcal{T}/C classification scores, shown in the bottom right plot, are typically binned when used in an analysis. When inspecting the \mathcal{T}/C classification score distributions in the bottom right of figure 3 it can be seen that `DYNEDGE` offers a better separation of \mathcal{T}/C events than the BDT.

The large difference in classification performance shown in figure 3 between the BDT and `DYNEDGE` can be explained by the fact that classical BDTs require sequential input data, such as IceCube events, to be aggregated from $[n_{\text{pulses}}, n_{\text{features}}]$ into $[1, d]$ -dimensional arrays as a preprocessing step, where d is the number of event variables for the BDTs to act on. This preprocessing step reduces the information available to the BDTs.

3.3 Event reconstruction performance of `DYNEDGE`

The event reconstruction performance for deposited energy, zenith, direction, and interaction vertex is shown in figure 5. The performance metric is the resolution, which we quantify here as the width W of the residual distribution R . For the deposited energy and zenith reconstructions, W is defined as

$$W = \frac{p_{84\text{th}}(R) - p_{16\text{th}}(R)}{2}, \quad (3.1)$$

where $p_{16\text{th}}(x)$ and $p_{84\text{th}}(x)$ correspond to the 16th and 84th percentile. For a normal distribution, the quantile W corresponds to the standard deviation σ , but it is more robust to outliers. For direction and

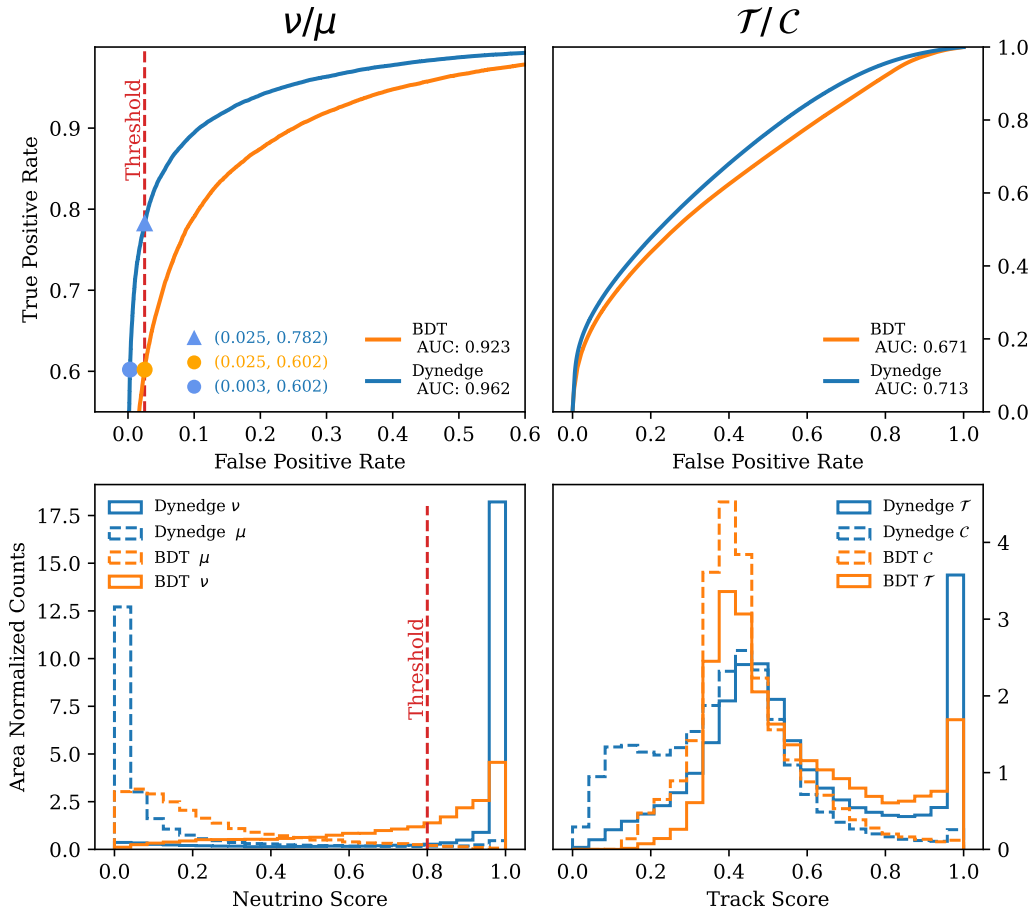


Figure 3. Left: ROC curves for DYNEDGE and current BDT ν/μ classification models and below are the distribution of classification scores shown with the threshold labeled in red. **Right:** ROC curve for DYNEDGE and current BDT T/C classification models. Distributions in prediction scores are shown below.

interaction vertex reconstructions, (which have strictly positive residual distributions), the resolution W is defined as the 50th percentile $p_{50\text{th}}(x)$. Since reconstruction resolutions are highly dependent on the energy of the neutrino, we provide the resolutions binned in energy. Additionally, we separate the residual distributions in true track (ν_μ CC) and cascade events to examine the performance of the algorithms in detail.

For comparison between DYNEDGE and RETRO, the relative improvement defined by

$$\text{relative improvement} = \left(1 - \frac{W_{\text{DYNEDGE}}}{W_{\text{RETRO}}}\right) \cdot 100 \quad (3.2)$$

is used. Here W_{DYNEDGE} and W_{RETRO} corresponds to the resolution of DYNEDGE and RETRO for a given reconstruction task. The relative improvement provides a percentage comparison of the resolutions.

The residual distributions for deposited energy (R_E)³, angles (R_{angle}), interaction vertex (R_{xyz}), and direction (R_{dir}) are defined in table 2 and some are shown in figure 4. From figure 4 it can be

³Notice that the residual distribution for energy is changed to $R_E = \frac{E_{\text{reco}} - E_{\text{true}}}{E_{\text{true}}} \cdot 100$ for the calculation of resolutions.

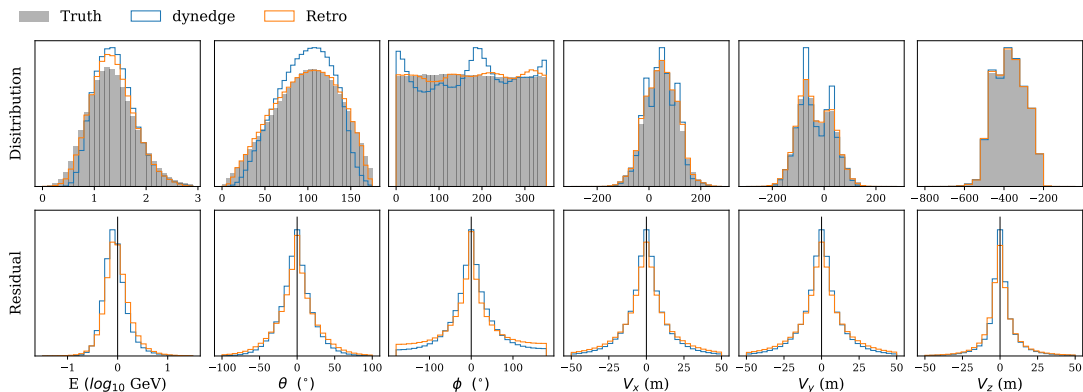


Figure 4. Area normalized distributions of predicted and true target variables on the reconstruction test set and their corresponding residual distributions. Residuals shown here follows the definitions shown in table 2 except for vertex coordinates, which show the relative difference between reconstructed and true coordinate.

seen that DYNEDGE tends to over-estimate the deposited energy for very low-energetic events. This is partly due to the lower statistics and the relatively poor signal-to-noise ratio below 30 GeV. On average, it will be optimal for a machine learning model to estimate a value close to the mean of the true distribution for examples where it has not yet learned a more optimal solution. Such examples can be considered to be difficult for the method, and by predicting a value close to the mean of the true distribution, the model minimizes the loss. This behavior is evident for DYNEDGE in the reconstructions of zenith and azimuth, where events with a relatively high estimated uncertainty have a preferred solution that minimizes the loss function. For the regression of azimuth, this behavior yields multimodal artifacts due to the cyclic nature of the variable. As seen in figure 1, DeepCore have more strings in the north/south than east/west. The events with high estimated azimuthal uncertainty piled around π and 2π are generally low energetic track events with a detector response similar to a neutrino traveling east-west, but are in fact traveling north or south. If there isn't enough information to pinpoint whether the event belongs in either north or south, the loss can be optimized by picking a value close to the observed locations of the pileups, as they correspond to a “mean” between north and south. The timing information of the event can be used to pick the optimal pile. We emphasize that while the distribution of predictions give the impression that RETRO is closer to the true distribution than DYNEDGE, the correct assessment of the quality of reconstructions comes from interpretations of the residual distributions. From the bottom panel of figure 4 it's evident that DYNEDGE produces reconstructions with narrower residual distributions and therefore superior resolutions.

As can be seen from figure 5, DYNEDGE performance improves upon the existing RETRO reconstruction in all 6 reconstruction variables for both track and cascade events between 1 GeV–30 GeV. This energy range is of particular importance to neutrino oscillation analyses, where flavor oscillations of atmospheric neutrinos appear below 25 GeV [10]. The typical reconstruction improvement in this range is at the level of 15%–20%. For zenith and direction reconstruction, the improvement is constant with energy for cascade-like events, while for track-like events the improvements ends around 100 GeV.

DYNEDGE performance relative to RETRO generally decreases at higher energies, and is possibly due to the lower number of available training samples at these energies, as shown in figure 4. As

mentioned in section 3.1, the reconstruction event selection is chosen to maximize available statistics with an equal amount of neutrino flavors, rather than optimizing the balance of track and cascade events. This choice improves on the overall performance under the given circumstances, but at the same time leads to an underrepresentation of track-like events, which further lowers the available statistics for this event type at higher energies.

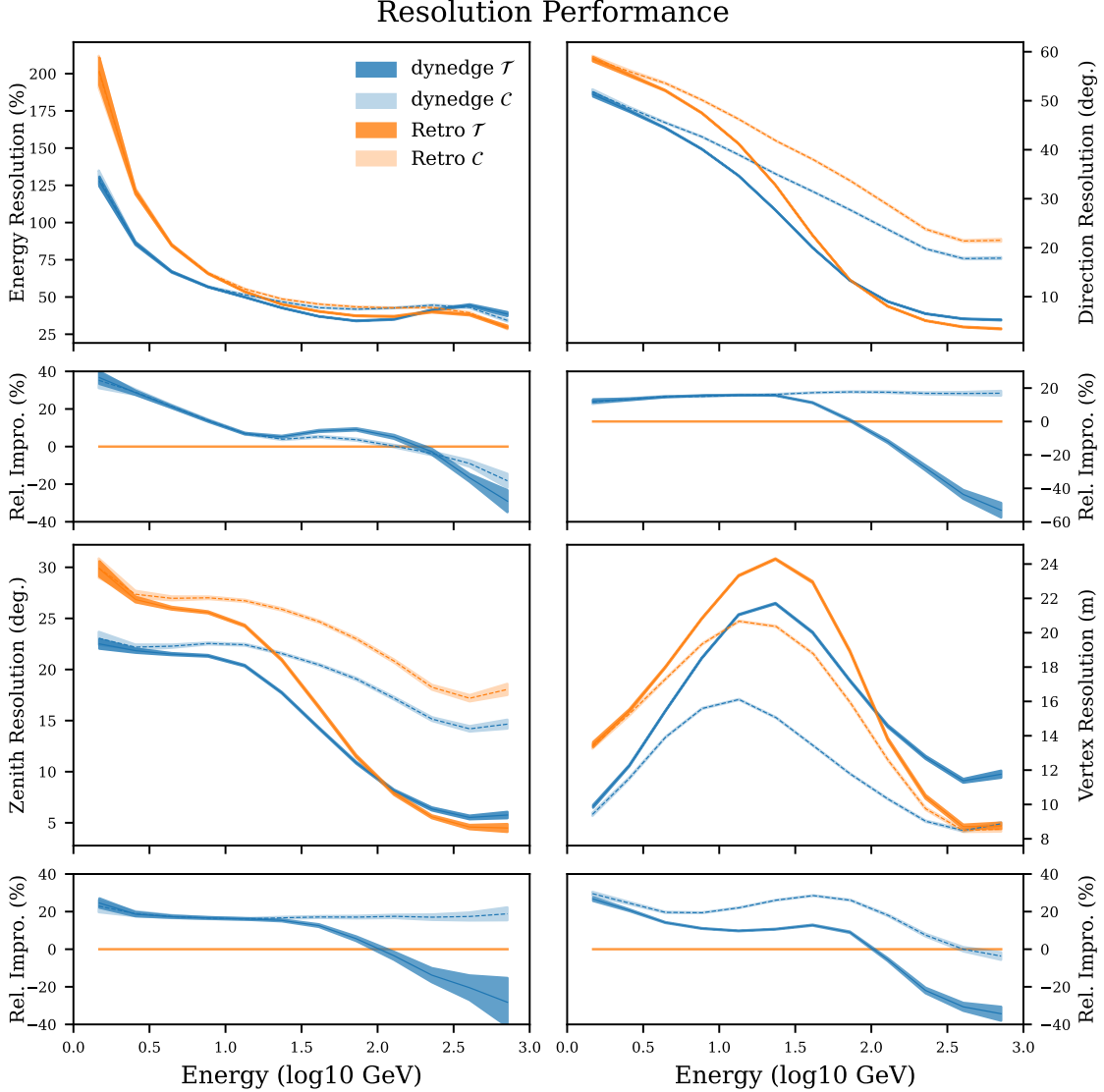


Figure 5. Event reconstruction performance for DYNEDGE, estimating deposited energy (top left), zenith angle (top right), direction (bottom left), and interaction vertex (bottom right). In all four cases, the performance is compared to RETRO, and the ratio below the plots shows relative improvement of DYNEDGE w.r.t. RETRO, where the light blue and dark blue curves represents the relative improvement for cascades and tracks, respectively. Positive values indicate an improvement in resolution. Lines represent reconstruction resolution, and the bands cover 1σ resolution uncertainty.

3.4 Runtime performance

The inference speed test measures the wall-time required to reconstruct the zenith angle on 1 million neutrino events in batches of 7168 events.⁴ The test includes the time required to load the batch into memory, apply re-scaling, convert the batch into graphs, move data to GPU memory, and pass them through DYNEDGE. A budget of 40 parallel workers is allocated to feed the model with batches. The test does not include overhead associated with writing the predictions to disk. The test was repeated 5 times on both GPU and CPU. The average speed was found to be 30.6 ± 2.0 kHz on GPU, and 0.221 ± 0.004 kHz for a single CPU core. The inference speed was tested for other variables considered in this work and was found to be within the uncertainties of the above values. Since a separate DYNEDGE model is trained for each variable, the individual reconstructions can be run in parallel, in which case the reconstruction rate of an event is equal to the inference speed if fully parallelized. If the reconstruction of each variable is not run in parallel, the reconstruction rate of an event is approximately the inference speed divided by the number of desired variables. For a full reconstruction of energy, zenith, azimuth, interaction vertex, ν/μ , and \mathcal{T}/C classification, the reconstruction rate is approximately 5.1 kHz on the GPU, or 37 Hz for the single CPU core, respectively. With classification and reconstruction speed of nearly double of the median IceCube trigger rate (2.7 kHz), DYNEDGE is in principle capable of real time reconstruction of IceCube events using a single GPU.

4 Robustness test

The results presented in section 3 show that DYNEDGE is suitable for both classification and reconstruction tasks in the low-energy range of IceCube, specifically for the neutrino-oscillation-relevant energy range. However, these results are computed on simulation based on a nominal configuration of the detector. In section 4.1, we investigate the robustness of DYNEDGE to perturbations of input variables D_{xyz} , t and q . These variables constitute the node features, and as such the perturbation test probes the robustness of DYNEDGE to systematic variations in the node features themselves. In section 4.2, we investigate the robustness of DYNEDGE to changes in systematic uncertainties associated with the detector medium and the angular acceptance of the DOMs. Variations in such assumptions lead not to variations in the node features, but to the topology of the events. The test therefore probes the robustness of DYNEDGE to variations in the connectivity of the graph representations of the events.

4.1 Perturbation of input variables

There are systematic uncertainties associated with the input variables shown in table 1. The position of each string is only known to a precision of a few meters horizontally, and the vertical position is known with a precision better than one meter [6].⁵ In this section, we investigate the variation in resolution and AUC from perturbations of input variables by $x \rightarrow x + \epsilon$, where x is an input variable and ϵ is a random number from a normal distribution with standard deviation σ_x .

⁴The inference speed was tested on a server running Ubuntu 20.04.3 LTS with a 64-core @2.00 GHz AMD EPYC 7662 using a single NVIDIA A100 SXM4 40 GB VRAM, 1TB of RAM, and 5TB of NVME disk space.

⁵Strings are also not perfectly vertical, which results in a non-uniform uncertainty on the horizontal and vertical position, but this effect is neglected in this test.

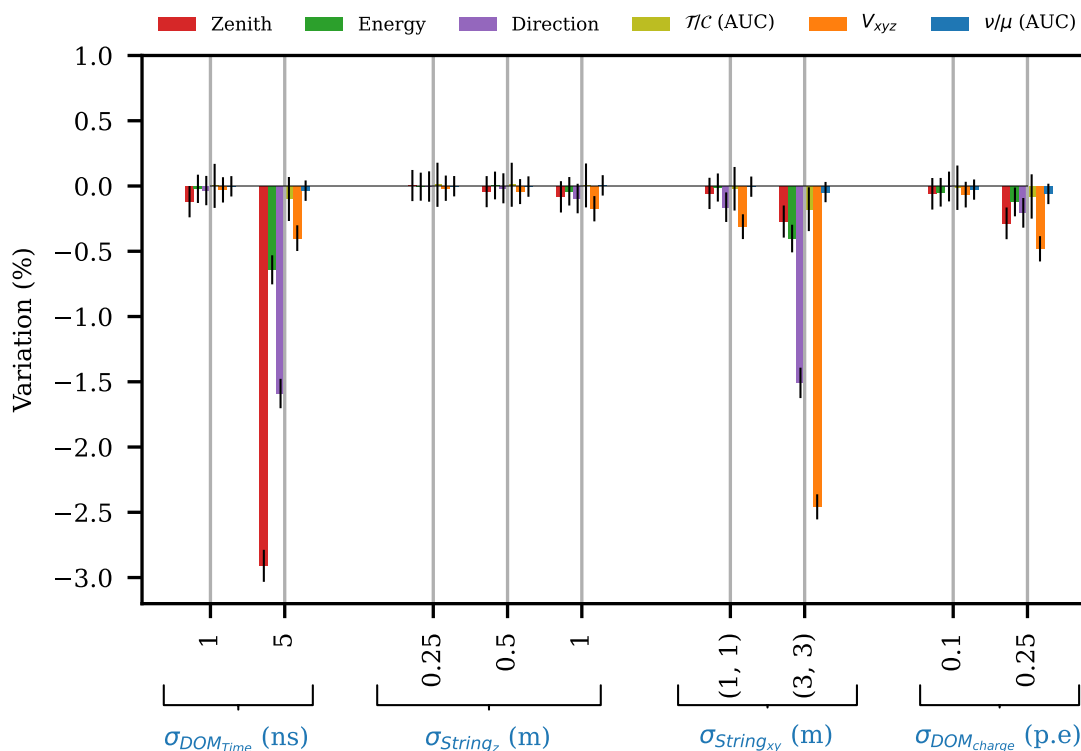


Figure 6. The variation in resolution and AUC induced by perturbation of input variables for DYNEDGE. The standard deviations of the normal distributions from which the perturbations are drawn are shown in the x -axis, and the 4 different perturbation tests are labeled in blue at the bottom. Thin vertical lines centered on each bar are error bars.

For the DOM positions, one ϵ is drawn for each string, resulting in correlated shifts for all DOMs in one string, and uncorrelated shifts between strings. The horizontal and vertical positions of the DOMs are also independently perturbed. Time and charge are perturbed pulse-wise, meaning that each pulse is perturbed independently. DYNEDGE is not re-trained, i.e. we use the networks trained on nominal detector assumptions, and run on perturbed input datasets. We perturb time, z -coordinate, xy -coordinates, and charge separately, for a few choices of σ . We calculate the percentage variation with respect to the nominal resolution and AUC score. Our test addresses two questions. First, it gives an indication of the expected change in reconstruction performance due to a systematic shift in the input variables. Second, it serves as a gauge on feature importance for the different reconstruction tasks. In figure 6, variation in resolution and AUC is shown as a function of the perturbation width σ . A negative value indicates a *worsening* with respect to the nominal performance. For example, -5% means a worsened resolution with a width W that is 5% larger. For AUC scores, a variation by -5% means a decrease of 5%. As seen in figure 6, zenith is the most sensitive to perturbations of the time, as expected, since time perturbations of the pulses may reverse the direction. Perturbations of vertical and horizontal positions of strings have little effect on energy and direction, but impact the interaction vertex resolution.

4.2 Variations in ice properties and module acceptance

The South Pole ice — IceCube’s detector medium — is a glacier with an intricate structure of layers with varying optical properties, and the refrozen boreholes where strings were deployed are understood to have different optical properties than the bulk ice. These sources of systematic uncertainty are constrained from fits to calibration data, but only to a finite precision of about $\pm 10\%$ of the nominal value [9]. In the tests presented in this section, we quantify the robustness of our reconstruction to changes in the ice properties allowed by calibration data, using a five parameter model covering the most important sources of systematic detector uncertainties in IceCube. We

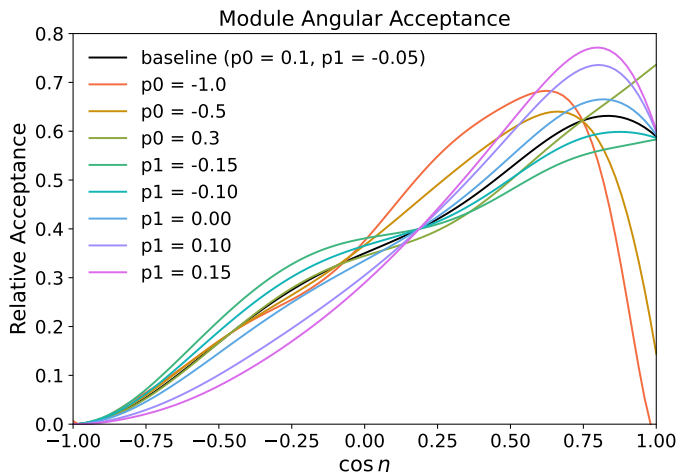


Figure 7. Variations in the DOM angular acceptance used for the robustness tests. The angle η is the photon arrival angle at the Module, with $\cos \eta = 1$ being head-on towards the face of the PMT and $\cos \eta = -1$ towards the backside of the PMT where generally the acceptance drops to zero.

consider 20 additional simulation sets with altered ice properties and module acceptance. Specifically, there are 8 perturbed simulation sets varying the scattering and absorption coefficients of the South Pole bulk ice by $(-30\%, -10\%, +10\%, +30\%)$ independently, four sets with variations in the overall optical efficiency of DOMs $(-10\%, -5\%, +5\%, +10\%)$, and 9 sets with independent variations in two key parameters p_0 and p_1 altering the angular acceptance of the DOMs. These last two parameters are the principal components of a unification of different approaches in IceCube that model changes in the borehole ice via the angular photon acceptance.⁶ As seen in figure 7, an increase in p_0 leads to an increase in directly up-going photon acceptance (face of the PMT), whereas changes in p_1 predominantly affect the photon acceptance around the waist of the PMT.

In this subsection, we seek to quantify the robustness of `DYNEDGE` to these variations by letting the model predict on the systematic sets and comparing the bias, resolution, and AUC to that recorded on the nominal data set used in section 3. For reference, the variation in the predictions of `DYNEDGE` is compared with the variations in predictions from the current method `RETRO`. For each individual systematic data set, only the events that are present in both the nominal set and the given systematic set are considered. The constraint to overlapping events is applied because the event selection

⁶https://github.com/icecube/angular_acceptance.

process is itself sensitive to systematic uncertainties and can therefore lead to different distributions in parameters such as deposited energy, impacting event selection.

4.2.1 Classification

To quantify the robustness of the \mathcal{T}/C and ν/μ tasks, the AUC is computed from each ROC curve on each systematic set for both DYNEDGE and the BDTs. The robustness of classification is then defined as the relative improvement in AUC as compared to the nominal AUC:

$$\text{AUC variation} = \left(1 - \frac{\text{AUC}_{\text{sys}}}{\text{AUC}_{\text{nom}}}\right).$$

Currently, muon simulation is only available for the sets with variation in optical efficiency, limiting the test of ν/μ AUC variation to these sets. The test of \mathcal{T}/C variation extends to all sets, and the results are shown in figure 8. As seen in figure 8, the error on AUC variation for ν/μ is much higher

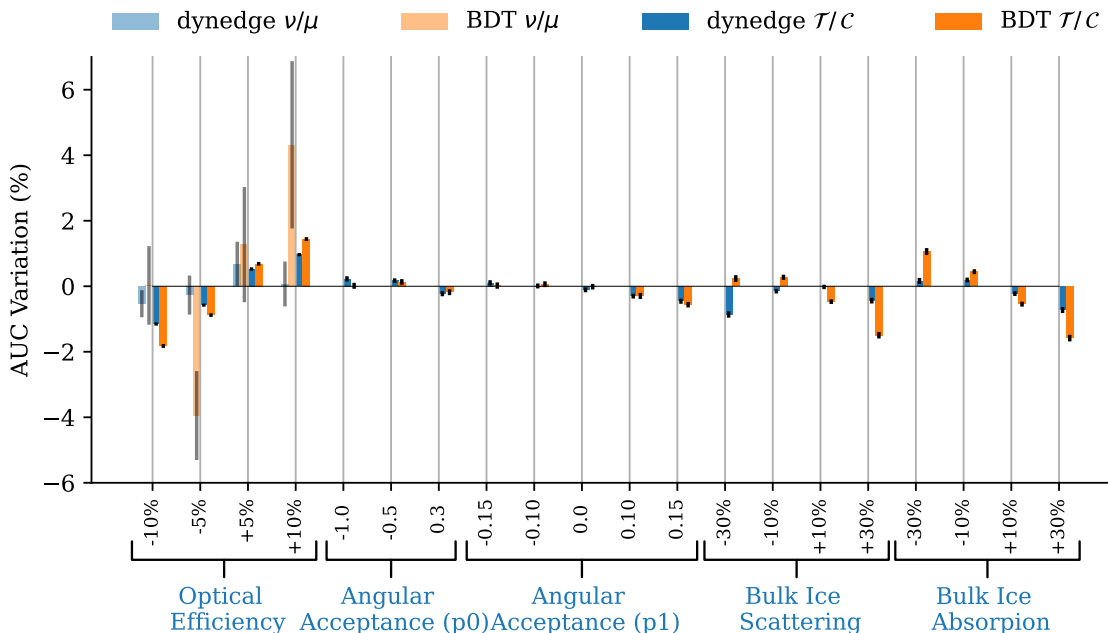


Figure 8. The variation in AUC for DYNEDGE and the current BDT on the 20 different systematic sets. Track/cascade classification is denoted by \mathcal{T}/C and neutrino/muon classification is denoted by ν/μ . Variation in ν/μ AUC is only shown on the first 4 samples, as the rest of the sets have no available muons.

than for \mathcal{T}/C . This is due to the lower statistics of the data selection shown in table 3. In general, we observe that \mathcal{T}/C classification has a variation well below $\pm 2\%$ of the nominally expected AUC for both methods, DYNEDGE and RETRO. The ν/μ classification task, however, indicates a higher robustness of our proposed DYNEDGE model than the current BDT classifier.

4.2.2 Reconstruction

The robustness of the reconstruction is reported in two main metrics; bias variation (BV) and resolution variation (RV), defined as $BV_{\text{sys}} = p_{50\text{th}}(R_{\text{sys}}) - p_{50\text{th}}(R_{\text{nom}})$ and $RV_{\text{sys}} = (1 - W_{\text{sys}}/W_{\text{nom}})$ where

W and R are as defined in section 3.3. These measures quantify the change relative to the nominal quantities. Since bias for the direction reconstruction is ambiguous, the bias variations of the zenith and azimuth reconstructions are shown instead.

Figure 9 shows the bias change for energy, zenith, and azimuth for the different simulation variations, while figure 10 reports the change in resolution. The biases induced by changes in systematic uncertainties is nearly identical between DYNEDGE and RETRO for all variables considered, while a slight overall increase in magnitude for DYNEDGE can be observed. Zenith bias is very sensitive to ice properties, whereas the induced bias in reconstructed energy is strongly affected by the optical efficiency and scattering/absorption of light, but less sensitive to angular acceptance of the DOMs. Such an effect is expected as the deposited energy is correlated with the number of measured pulses. Also, a linear change in DOM efficiency leads to an approximately linear change in energy bias for both methods. As seen in figure 10, the overall effect on the resolution of the reconstruction is nearly identical for DYNEDGE and RETRO. While the zenith bias is unaffected by optical efficiency variations, the zenith resolution changes for both DYNEDGE and RETRO as a function of the optical efficiency. A decrease/increase in optical efficiency leads to a decrease/increase in available information for the methods, which, as expected, leads to a widening/narrowing of the residual distribution.

To summarize the difference in robustness between DYNEDGE and RETRO of the current BDT classifiers, the RMS of the variations shown in figure 9, 10, 8 is presented in table 4.

Table 4. Variation in bias, resolution, and AUC of reconstruction across all 20 systematic samples for DYNEDGE and RETRO for track and cascade events. (Results for ν/μ classification are based only on the sets with varying DOM optical efficiency.)

Target	DYNEDGE (RMS)		RETRO/BDT (RMS)	
	Tracks	Cascades	Tracks	Cascades
Energy Bias (%)	9.1	6.4	7.2	5.6
Zenith Bias ($^{\circ}$)	1.4	3.2	1.0	2.4
Azimuth Bias ($^{\circ}$)	0.07	0.09	0.04	0.19
Energy Resolution (%)	4.8	7.7	4.9	7.1
Zenith Resolution (%)	3.1	3.2	4.2	3.6
Direction Resolution (%)	4.5	4.9	4.9	4.5
Vertex Resolution (%)	1.2	1.2	3.2	5.3
ν/μ AUC (%)	0.5		3	
\mathcal{T}/\mathcal{C} AUC (%)	0.5		0.8	

The RMS values for ν/μ AUC are based only on the variations seen on 4 sets varying the optical efficiency of the DOMs. When considering the RMS values for the regression tasks, DYNEDGE is generally affected by higher RMS values for both track and cascade events, indicating a higher variation induced by the systematic uncertainties compared to RETRO. The single exception is the resolution in the reconstructed vertex, where DYNEDGE exhibits a clear advantage over RETRO.

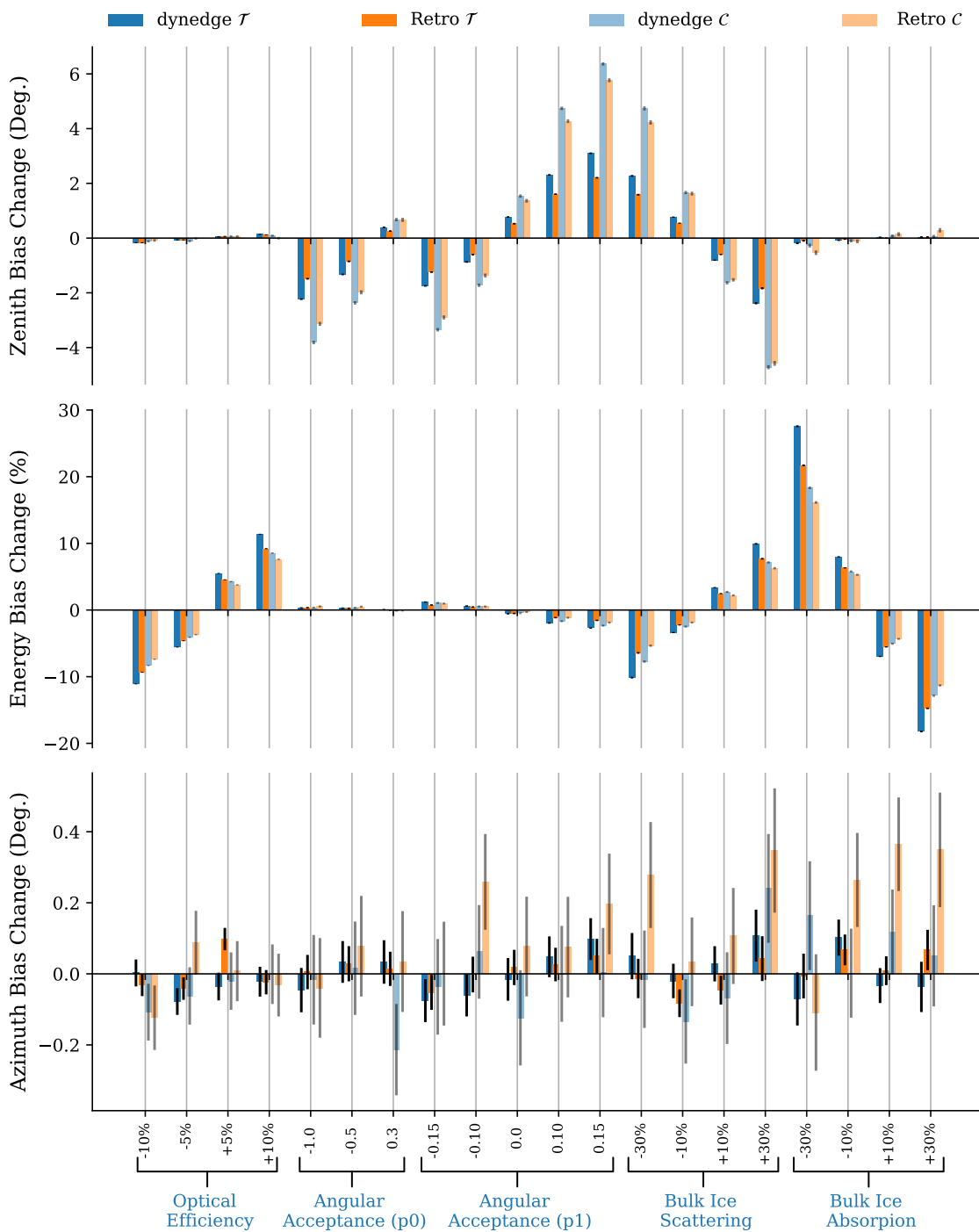


Figure 9. The variation in bias for DYNEDGE and RETRO on the 20 different systematic sets for reconstruction targets energy, zenith, and direction. Please notice that the y-axis values for azimuth is much smaller than for zenith.

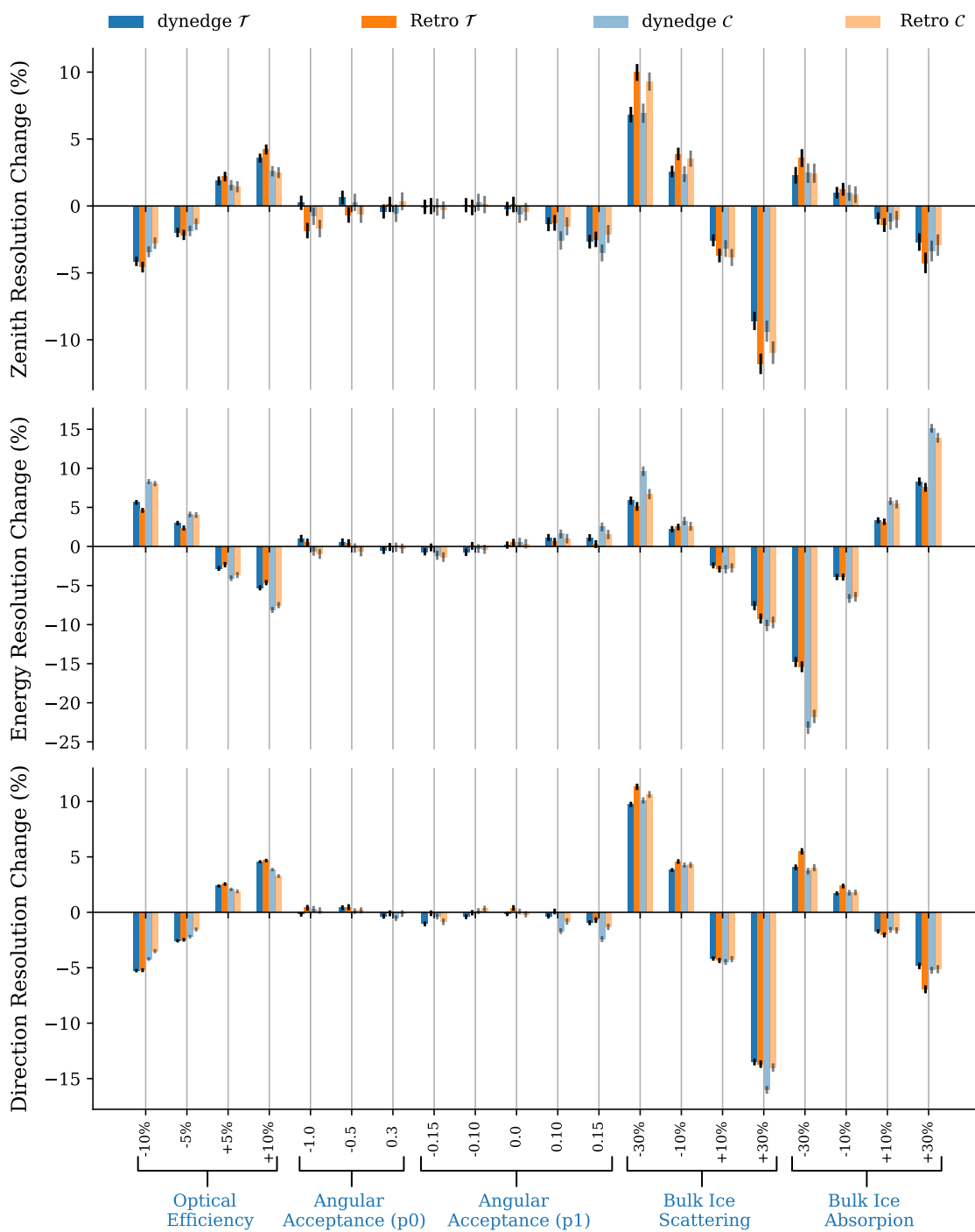


Figure 10. The variation in resolution for DYNEDGE and RETRO on the 20 different systematic sets for reconstruction targets energy, zenith, and direction.

5 Summary and conclusions

We propose a GNN-based reconstruction algorithm for IceCube events named `DYNEDGE`, that can be applied to any IceCube event. We have selected simulated low-energy data used for studies of atmospheric neutrino oscillations in IceCube as our dataset, and in this energy range, we have benchmarked `DYNEDGE` against the state-of-the-art reconstruction and classification algorithms as a proof of concept.

`DYNEDGE` offers substantial improvements to both \mathcal{T}/C and ν/μ classifications in the entire low energy range. In the energy range 1 GeV–30 GeV, relevant to standard atmospheric $\nu_{\mu}-\nu_{\tau}$ oscillation studies, `DYNEDGE` exhibits a 15–20% improvement in reconstruction of energy, zenith, direction, and interaction vertex. Outside the energy range relevant to neutrino oscillations, the improvement in event reconstruction decreases, and at high energies becomes worse than `RETRO`. This worsening effect is ascribed to lower statistics in the training set at these energies and a disproportional amount of cascades compared to more energetic (i.e. longer) tracks. Future studies will focus on improving performance in this region.

`DYNEDGE` and `RETRO` are both robust against systematic uncertainties in DOM optical efficiency and angular acceptance, as well as the scattering and absorption properties of the bulk ice. `DYNEDGE` is also robust against random perturbations to inputs such as DOM position, timing, and charge.

Based on tests of reconstruction speed, `DYNEDGE` could reconstruct events online at the South Pole. `DYNEDGE` is also flexible enough to be compatible with the planned IceCube Upgrade featuring new DOM types on new strings, as well as other neutrino detectors with arbitrary geometries. This feature may make `DYNEDGE` particularly useful for the next generation of large neutrino detectors, such as KM3Net [38] and the proposed IceCube Gen2 [39]. `DYNEDGE` is implemented using GraphNeT [23].

Acknowledgments

The IceCube collaboration acknowledges the significant contributions to this manuscript from Rasmus Ørsøe. The authors gratefully acknowledge the support from the following agencies and institutions: U.S.A. — U.S. National Science Foundation-Office of Polar Programs, U.S. National Science Foundation-Physics Division, U.S. National Science Foundation-EPSCoR, Wisconsin Alumni Research Foundation, Center for High Throughput Computing (CHTC) at the University of Wisconsin — Madison, Open Science Grid (OSG), Extreme Science and Engineering Discovery Environment (XSEDE), Frontera computing project at the Texas Advanced Computing Center, U.S. Department of Energy-National Energy Research Scientific Computing Center, Particle astrophysics research computing center at the University of Maryland, Institute for Cyber-Enabled Research at Michigan State University, and Astroparticle physics computational facility at Marquette University; Belgium — Funds for Scientific Research (FRS-FNRS and FWO), FWO Odysseus and Big Science programmes, and Belgian Federal Science Policy Office (Belspo); Germany — Bundesministerium für Bildung und Forschung (BMBF), Deutsche Forschungsgemeinschaft (DFG), Helmholtz Alliance for Astroparticle Physics (HAP), Initiative and Networking Fund of the Helmholtz Association, Deutsches Elektronen Synchrotron (DESY), and High Performance Computing cluster of the RWTH Aachen; Sweden — Swedish Research Council, Swedish Polar Research Secretariat, Swedish National Infrastructure for Computing (SNIC), and Knut and Alice Wallenberg Foundation; Australia — Australian Research Council; Canada — Natural Sciences and Engineering Research Council

of Canada, Calcul Québec, Compute Ontario, Canada Foundation for Innovation, WestGrid, and Compute Canada; Denmark — Villum Fonden, Carlsberg Foundation, and European Commission; New Zealand — Marsden Fund; Japan — Japan Society for Promotion of Science (JSPS) and Institute for Global Prominent Research (IGPR) of Chiba University; Korea — National Research Foundation of Korea (NRF); Switzerland — Swiss National Science Foundation (SNSF); United Kingdom — Department of Physics, University of Oxford.

References

- [1] ICECUBE collaboration, *The IceCube Data Acquisition System: Signal Capture, Digitization, and Timestamping*, *Nucl. Instrum. Meth. A* **601** (2009) 294 [[arXiv:0810.4930](#)].
- [2] ICECUBE collaboration, *Neutrino Physics with the IceCube Detector*, in proceedings of the 23rd Lake Louise Winter Institute: Fundamental Interactions, Lake Louise, Alberta, Canada, 18–23 February 2008, pp. 283–287 [[arXiv:0806.1717](#)].
- [3] ICECUBE collaboration, *The Design and Performance of IceCube DeepCore*, *Astropart. Phys.* **35** (2012) 615 [[arXiv:1109.6096](#)].
- [4] F. Halzen and S.R. Klein, *IceCube: An Instrument for Neutrino Astronomy*, *Rev. Sci. Instrum.* **81** (2010) 081101 [[arXiv:1007.1247](#)].
- [5] ICECUBE collaboration, *Low energy event reconstruction in IceCube DeepCore*, *Eur. Phys. J. C* **82** (2022) 807 [[arXiv:2203.02303](#)].
- [6] ICECUBE collaboration, *The IceCube Neutrino Observatory: Instrumentation and Online Systems*, 2017 *JINST* **12** P03012 [[arXiv:1612.05093](#)].
- [7] PARTICLE DATA GROUP collaboration, *Review of Particle Physics*, *PTEP* **2020** (2020) 083C01.
- [8] ICECUBE collaboration, *Energy Reconstruction Methods in the IceCube Neutrino Telescope*, 2014 *JINST* **9** P03009 [[arXiv:1311.4767](#)].
- [9] ICECUBE collaboration, *Measurement of Atmospheric Neutrino Oscillations at 6–56 GeV with IceCube DeepCore*, *Phys. Rev. Lett.* **120** (2018) 071801 [[arXiv:1707.07081](#)].
- [10] ICECUBE collaboration, *Measurement of Atmospheric Tau Neutrino Appearance with IceCube DeepCore*, *Phys. Rev. D* **99** (2019) 032007 [[arXiv:1901.05366](#)].
- [11] ICECUBE collaboration, *Measurement of South Pole ice transparency with the IceCube LED calibration system*, *Nucl. Instrum. Meth. A* **711** (2013) 73 [[arXiv:1301.5361](#)].
- [12] AMANDA collaboration, *Muon track reconstruction and data selection techniques in AMANDA*, *Nucl. Instrum. Meth. A* **524** (2004) 169 [[astro-ph/0407044](#)].
- [13] ICECUBE collaboration, *The IceCube Upgrade — Design and Science Goals*, *PoS ICRC2019* (2021) 1031 [[arXiv:1908.09441](#)].
- [14] ICECUBE-GEN2 collaboration, *A next-generation optical sensor for IceCube-Gen2*, *PoS ICRC2021* (2021) 1062 [[arXiv:2107.08837](#)].
- [15] T. Hastie, R. Tibshirani and J. Friedman, *Neural networks*, in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer New York, New York, NY, U.S.A. (2009), pp. 389–416.
- [16] R. Abbasi et al., *A Convolutional Neural Network based Cascade Reconstruction for the IceCube Neutrino Observatory*, 2021 *JINST* **16** P07041 [[arXiv:2101.11589](#)].

- [17] ICECUBE collaboration, *Using convolutional neural networks to reconstruct energy of GeV scale IceCube neutrinos*, 2021 *JINST* **16** C09019 [[arXiv:2109.08152](https://arxiv.org/abs/2109.08152)].
- [18] ICECUBE collaboration, *Direction reconstruction using a CNN for GeV-scale neutrinos in IceCube*, 2021 *JINST* **16** C11001.
- [19] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner and G. Monfardini, *The graph neural network model*, *IEEE Trans. Neural Networks* **20** (2009) 61.
- [20] N. Choma, F. Monti, L. Gerhardt, T. Palczewski, Z. Ronaghi, P. Prabhat et al., *Graph neural networks for icecube signal classification*, in proceedings of the 17th *IEEE International Conference on Machine Learning and Applications*, Orlando, FL, U.S.A., 17–20 December 2018, pp. 386–391.
- [21] Z. Chen, F. Chen, L. Zhang, T. Ji, K. Fu, L. Zhao et al., *Bridging the gap between spatial and spectral domains: A survey on graph neural networks*, [arXiv:2002.11867](https://arxiv.org/abs/2002.11867).
- [22] Y.A. LeCun, L. Bottou, G.B. Orr and K.-R. Müller, *Efficient backprop*, in *Neural Networks: Tricks of the Trade: Second Edition*, G. Montavon, G.B. Orr and K.-R. Müller, eds., Springer Berlin Heidelberg, Berlin, Heidelberg, Germany (2012), pp. 9–48.
- [23] A. Søgaard, R.F. Ørsøe, L. Bozianu, M. Holm, K.E. Iversen, T. Guggenmos et al., *Graphnet*, Zenodo, June, 2022, DOI: [10.5281/zenodo.6720189](https://doi.org/10.5281/zenodo.6720189).
- [24] M. Fey and J.E. Lenssen, *Fast Graph Representation Learning with PyTorch Geometric*, 2019 [[arXiv:1903.02428](https://arxiv.org/abs/1903.02428)].
- [25] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein and J.M. Solomon, *Dynamic graph cnn for learning on point clouds*, *ACM Trans. Graph.* **38** (2019) 146.
- [26] H. Pei, B. Wei, K.C.-C. Chang, Y. Lei and B. Yang, *Geom-gcn: Geometric graph convolutional networks*, in proceedings of the *International Conference on Learning Representations*, 26 April–1 May 2020, <https://openreview.net/forum?id=S1e2agrFvS>.
- [27] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, 2014 [[arXiv:1412.6980](https://arxiv.org/abs/1412.6980)].
- [28] L.N. Smith and N. Topin, *Super-convergence: very fast training of neural networks using large learning rates*, *Proc. SPIE* **11006** (2019) 1100612.
- [29] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao et al., *On the variance of the adaptive learning rate and beyond*, in proceedings of the *International Conference on Learning Representations*, 26 April–1 May 2020, <https://openreview.net/forum?id=rkgz2aEKDr>.
- [30] G. Raskutti, M.J. Wainwright and B. Yu, *Early stopping for non-parametric regression: An optimal data-dependent stopping rule*, in proceedings of the 49th *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, U.S.A., 28–30 September 2011, pp. 1318–1325.
- [31] C.M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg, Germany (2006).
- [32] S. Jadon, *A survey of loss functions for semantic segmentation*, in proceedings of the *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*, Via del Mar, Chile, 27–29 October 2020, pp. 1–7.
- [33] S. Kumar and Y. Tsvetkov, *Von mises-fisher loss for training sequence to sequence models with continuous outputs*, in proceedings of the *International Conference on Learning Representations*, New Orleans, LA, U.S.A., 6–9 May 2019, <https://openreview.net/forum?id=rJIDnoA5Y7>.

- [34] C. Andreopoulos et al., *The GENIE Neutrino Monte Carlo Generator*, *Nucl. Instrum. Meth. A* **614** (2010) 87 [[arXiv:0905.2517](#)].
- [35] H. He and E.A. Garcia, *Learning from imbalanced data*, *IEEE Trans. Knowledge Data Eng.* **21** (2009) 1263.
- [36] T. Fawcett, *An introduction to roc analysis*, *Pattern Recogn. Lett.* **27** (2006) 861.
- [37] D. McClish, *Analyzing a portion of the roc curve*, *Med. Decis. Making* **9** (1989) 190.
- [38] KM3NET collaboration, *Deep-sea deployment of the KM3NeT neutrino telescope detection units by self-unrolling*, *2020 JINST* **15** P11027 [[arXiv:2007.16090](#)].
- [39] ICECUBE-GEN2 collaboration, *IceCube-Gen2: the window to the extreme Universe*, *J. Phys. G* **48** (2021) 060501 [[arXiv:2008.04323](#)].