

Graph Neural Networks for Low-Energy Event Classification & Reconstruction in IceCube

R. Ørsøe,¹ M. Ha Minh,¹ P. Eller,¹ A. Søgaard,² T. Petersen,² The IceCube collaboration

¹*Department of Physics, Technical University Munich,
James-Frank-Str. 1 85748 Garching, Germany*

²*Niels Bohr Institute, University of Copenhagen, Blegdamsvej 17 2100 Copenhagen, Denmark*

E-mail: first@one.univ

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 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. We propose to represent IceCube events as point cloud graphs and use a Graph Neural Network (GNN) as the classification and reconstruction method. We demonstrate the capability of the GNN approach in 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 1–1000 GeV energy range to the current state-of-the-art maximum likelihood techniques used in current IceCube analyses. We include the effects of known systematic uncertainties. For neutrino event classification, we find that the GNN provides a 15% increase in signal efficiency at the same background rate, compared to current IceCube methods. Alternatively, the GNN offers a reduction of the background (i.e. false positive) rate by a factor 5-6 (to below 1%) at the same signal efficiency. For the reconstruction of energy, direction, and interaction vertex, we find an average improvement of 13–20% in resolution compared to current maximum likelihood techniques in the energy range of 1–30 GeV. Finally, 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 2.7kHz, which opens the possibility of using low energy neutrinos in online searches for transient events.

KEYWORDS: Only keywords from JINST's keywords list please

ARXIV EPRINT: [we.need.one](https://arxiv.org/abs/1908.01312)

Contents

	1 Introduction	2
	1.1 The IceCube Detector	2
35	1.2 The Challenge of IceCube Event Classification and Reconstruction	2
	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
40	2.2 Model Architecture	6
	2.3 Event Classification and Reconstruction	8
	3 Performance on Low Energy Neutrino Events	9
	3.1 Selected Datasets	10
	3.2 Event Classification Performance	10
45	3.3 Event Reconstruction Performance of DYNEDGE	11
	3.4 Runtime Performance	14
	4 Robustness Test	14
	4.1 Perturbation of Input Variables	15
	4.2 Variations in Ice Properties and Module Acceptance	16
50	4.2.1 Classification	17
	4.2.2 Reconstruction	18
	5 Summary and Conclusions	21

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) on 86 strings, placed at depths between 1450 m and 2450 m. The main detector array consists of 78 strings arranged on a roughly hexagonal array, with an average horizontal distance of 125 m between neighboring strings [1] (see Fig. 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 area, named “DeepCore” [2], 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].

Charged particles resulting from neutrino interactions in the ice can emit faint traces of Cherenkov light. This near-UV light is eventually registered as signals in the DOMs. This signal, however, is interspersed with noise stemming from radioactive decays in the DOMs and intrinsic noise of the readout electronics. The IceCube trigger aims to reduce the noise component, and event candidates are read out from the detector at a rate of around 2.7 kHz [5]. These recorded events are dominated by random triggers caused by random triggers caused by noise hits, followed by triggers due to downgoing atmospheric muons, followed by atmospheric neutrinos at a rate several orders of magnitude below.

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 two morphological classes that serve as proxies for the underlying neutrino flavour and interaction type. “Track-like” 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. Aside from a large background of atmospheric muons produced by cosmic rays, this class of events is dominated by ν_μ charged-current (CC) interactions from atmospheric and astrophysical neutrinos. However, 17% of ν_τ CC interactions produce τ leptons that decay into muons, and these are also observed as track events [6]. The other class comprises “cascade-like”

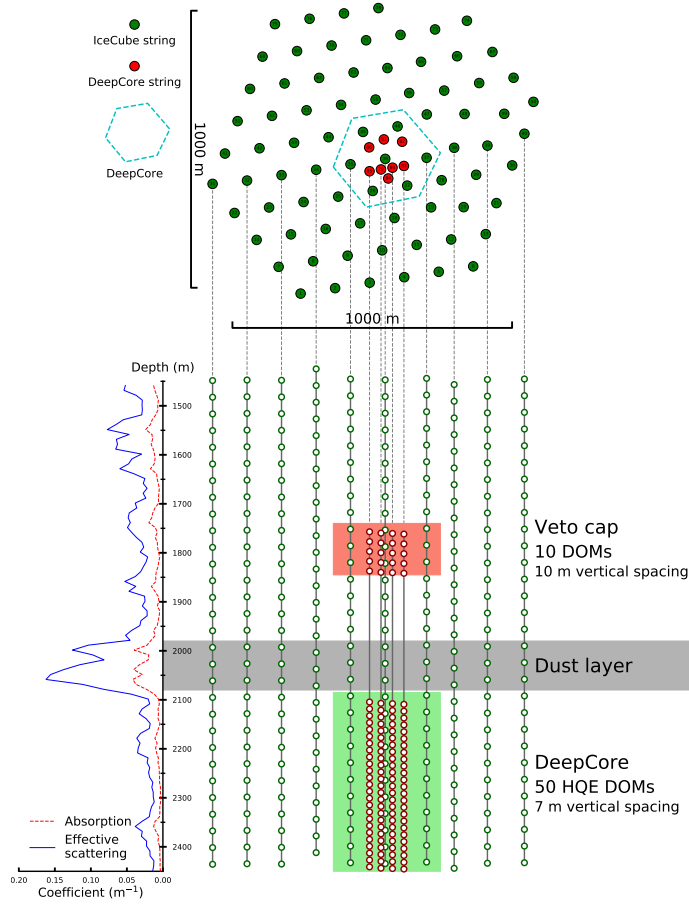


Figure 1: Top and side view of IceCube detector [2]. Ice properties as a function of depth is shown on the left. The in-ice dust layer is marked in grey. The dust layer is a layer in the ice with dust impurities and therefore reduced optical qualities. DeepCore is placed under the dust layer and shown in green.

90 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), ν_e CC, and 83% of ν_τ CC interactions. For brevity, we will refer to the classification of track
(T) and cascade (C) morphologies as “ \mathcal{T}/C ”. Another classification task is the discrimination of
signal events (neutrinos) from background events (atmospheric muons), which will be referred to
95 as “ ν/μ ”.

Input Data The input data to IceCube reconstruction algorithms are a time series of "hits", or
PMT voltage pulses, per event. Each element in this sequence corresponds to the time t at which
the PMT readout indicates a measured pulse with charge q . The PMTs themselves are located at

fixed DOM positions (D_{xyz}) and have a certain quantum efficiency (QE). These six variables of the time sequence are summarized in Table 1 and form the input data to the reconstruction.

Features	Description
D_{xyz}	Position of DOMs in IceCube coordinates
t	Pulse time relative to trigger time
q	Charge of a pulse
QE	quantum efficiency of PMT

Table 1: Node features in graph representation of neutrino events. Some DOMs have an improved photocathode in their PMT and QE is the relative quantum efficiency measure between these and the ordinary DOMs [2].

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 dependant 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 [7, 8]. For typical events in this energy range, n_{pulses} can range from below ten to several hundred after cleaning. Because some of the pulses in an event are due to noise hits, low-energy neutrino events often suffer from a relatively poor signal-to-noise ratio. In addition, at low energies the track event can be so short that they cannot be easily distinguished from cascade-like events. Reconstruction, in general, is further complicated by the irregular DOM positioning and varying ice properties as a function of x,y,z, position in the IceCube array.

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 have ranged in complexity from simple straight-line least-squares fits to event geometries to highly sophisticated detector response functions based on photon ray tracing.

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. 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 [9]. 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 1–1000 GeV) such likelihood-based reconstructions were used, for example, in the analyses of Refs. [7, 8]. We will use the state-of-the-art IceCube low-energy reconstruction algorithm RETRO [10] as a benchmark for our work. On average, RETRO requires around 40 seconds

to reconstruct one event, which is one of the main shortcomings of the method. Another limitation is its use of approximate symmetries, such as the azimuthal symmetry of IceCube DOMs, as well as the assumption that ice properties change as a function of depth only.

The IceCube Upgrade [11] 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 [12]. Porting 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. Hence, it is currently not planned to use RETRO for the Upgrade detector.

1.2.2 Machine-Learning-Based Reconstruction Methods

An alternative method to maximum likelihood estimation is (logistic) regression. 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 Neural Networks (NNs) can offer a viable solution. A regression algorithm needs to map the ragged input data of shape $[n_{\text{pulses}}, 6]$ 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) [13]. Embedding events into images, however, 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 for cascade events at energies higher than what is considered in our work. An alternative NN architecture employing a Graph Neural Network (GNN) approach is used in [14], but focuses solely on ν/μ classification.

Our work proposes a general reconstruction method based on GNNs that scales to the entire energy range of IceCube, is compatible with the IceCube Upgrade, and can reconstruct at speeds fast enough to run in real time online processing at the South Pole. We consider a complete 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 generalises 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.

For our approach, we choose each event to be represented 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 neighbours based on the Euclidean distance, and for this reason we consider the interconnectivity of the nodes in the graphs to be spatial [15].

2.1 Preprocessing of IceCube Events

The variables listed in Table 1 span different units and orders of magnitudes. The positions of the DOMs vary over a kilometer, and the times that pulses span are several microseconds, while charge is measured in single photoelectrons. 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 will lead to a slower convergence time [16]. Each input variable x is therefore transformed into \tilde{x} using

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

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

2.2 Model Architecture

Our GNN implementation, referred to as DYNEDGE from here onwards, uses a convolutional operator *EdgeConv* [17], 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 neighbourhood of n_j as

$$\tilde{x}_j = \sum_{i=1}^{N_{\text{neighbors}}} \text{MLP}(x_j, x_j - x_i), \quad (2.1)$$

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 Eq. (2.1) for every node in the graph.

The full architecture of DYNEDGE is shown in Fig. 2. Firstly, the following global statistics are calculated from the input graph: node homophily ratio [18] 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 feature¹. Secondly, 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 Fig. 2, the EdgeConv block is initialized with a k -nearest neighbours (k -nn) computation that redefines the edges of the input graph. This step lets DYNEDGE calculate the 8 nearest Euclidean neighbours and effectively 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.

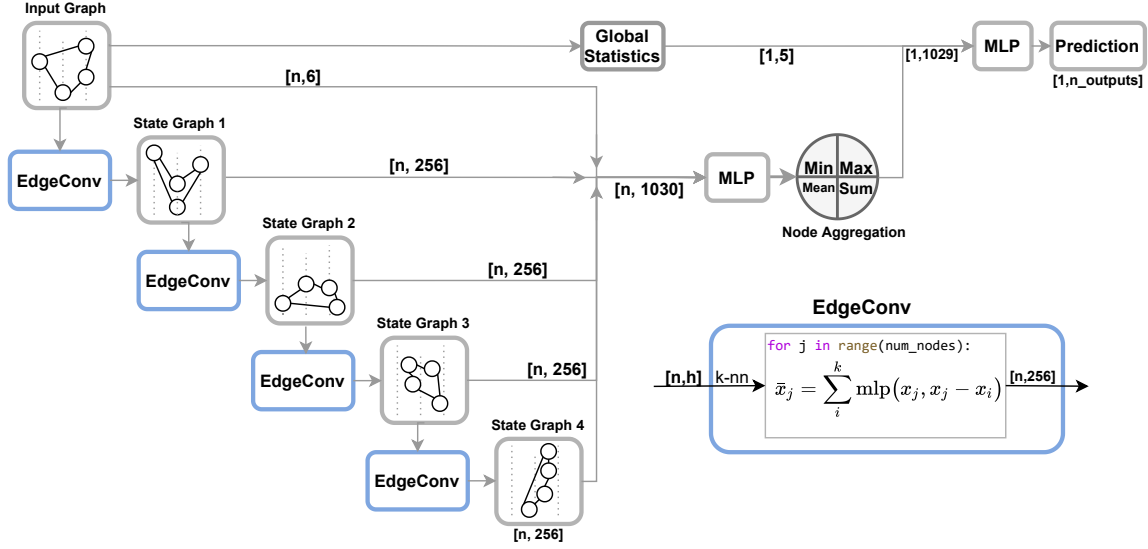


Figure 2: A diagram of the architecture of DYNEDGE. The "Input Graph" is a toy illustration of the input, and the subsequent "State Graphs" illustrates how the node position and connectivity change after convolutions by EdgeConv. An illustration of the inner workings of the EdgeConv block is provided to the right.

The input graph and each state graph are then concatenated together into an $[n, 1030]$ -dimensional array that is passed through a fully connected MLP block containing two layers. The block maps the $[n, 1030]$ -dimensional array into an $[n, 256]$ -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 on the form $f : [n, 256] \rightarrow [1, 256]$ ². Aggregations are concatenated together to minimize information loss, which produces an array with dimension $[1, 4 \times 256]$ that is concatenated together with the initially calculated global statistics, producing a $[1, 1029]$ -dimensional input array to the subsequent 2-layer MLP block that makes the final predic-

¹The homophily ratio of quantifies how many of the pulses originate from the same PMT and how many of the pulses happened at the same time.

²While more sophisticated pooling methods were tested, none improved upon this choice.

tion 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 Fig. 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 Fig. 2. Also, hyperparameters such as the number of nearest neighbours used in the k -nn 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.

2.3 Event Classification and Reconstruction

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 [19] optimizer and a custom piece-wise-linear implementation of One-Cycle [20] learning rate schedule with a warm-up period [21]. The scheduler increases linearly from 10^{-5} to 10^{-3} during the first 50% of iterations in the first epoch and thereafter linearly decreases to 10^{-5} during the remaining iterations in the 30 epoch training budget. Early stopping [22] is implemented with a patience of 5 epochs.

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_{\text{xyz}}} = \vec{P}_{\text{reco}} - \vec{P}_{\text{true}} $
\mathcal{T}/C	Classification into tracks and cascades	–

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.

For classification, the Binary Cross Entropy (BCE) loss is used since we only ever consider two categories: neutrino or muon events, and tracks or cascades. For each regression task a specific

loss function is chosen.

For regression of deposited energy, a LogCosh function is used

$$\text{Loss}(E) = \ln \left(\cosh \left(\log_{10} E_{\text{reco}} - \log_{10} E_{\text{true}} \right) \right), \quad (2.2)$$

where the embedding $f : E \rightarrow \log_{10} E$ is added to account for many orders of magnitude that the deposited energy spans, and $\log_{10} E_{\text{reco}}$ 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 MSE, LogCosh offers a steadier gradient around zero and is approximately linear for large residuals.

For angular regression, a 2D von Mises–Fisher [23] 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 2D von Mises-Fisher distribution is given by $p_2(\bar{x}|\bar{u}, k) = C_2(k) \exp(k \bar{u} \cdot \bar{x}) = C_2 \exp(k \cos \Delta\phi)$ 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 2D 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 vector³. The loss function is created by taking the negative log of $p_2(\bar{x}|\bar{u}, k)$:

$$\text{Loss}(\phi) = -\ln p_2(\bar{x}|\bar{u}, k) = -\ln(C_2(k)) - \ln(k/4\pi) + k + \ln(1 - e^{-2k}) - k \cos(\Delta\phi). \quad (2.3)$$

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 $f : (x, y, z) \rightarrow \left(\frac{x}{|\max(x)|}, \frac{y}{|\max(y)|}, \frac{z}{|\max(z)|} \right)$. DYNEDGE then predicts the embedded interaction vertex vector, and the loss function used is the Euclidean distance between the true and reconstructed 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 1–1000 GeV. At the end of the section, a short examination of the runtime performance will be provided.

³A more intuitive approach such as a 3D 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 sub-optimal results because the azimuthal component of the loss is too dominant at lower energies. Substantial improvement in zenith reconstruction is gained by keeping the two angles apart.

3.1 Selected Datasets

The IceCube simulation used for training and testing the DYNEDGE reconstructions and classifications is borrowed from the collaboration’s official simulation for neutrino oscillation analyses. The dataset and selection process is similar to the ones described in Ref. [8] and is also the same dataset as used in Ref. [10].

To ensure a fair comparison between DYNEDGE and RETRO, only the events that have passed to the last step in the event selection are considered, to ensure that any corrections to the reconstructions from RETRO have been applied. This totals approximately 8.3 million neutrino and muon events that are further divided into three task specific datasets summarized in Table 3.

Task	ν_e^{NC}	ν_e^{CC}	ν_μ^{NC}	ν_μ^{CC}	ν_τ^{NC}	ν_τ^{CC}	μ
ν/μ (Train)	3k	32k	3k	32k	9k	1k	108k
ν/μ (Test)	2k	16k	2k	16k	5k	13k	53k
Reconstruction (Train)	121k	1.13M	117k	1.14M	339k	917k	
Reconstruction (Test)	60k	558k	289k	2.81M	176k	479k	
\mathcal{T}/C (Train)	122k	122k	122k	365k			
\mathcal{T}/C (Test)	60k	1.57M	285k	3.57M	515k	1.40M	

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.

As seen from Table 3 the \mathcal{T}/C training dataset contains an even amount of tracks and cascades. ν_τ is 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. For the ν/μ classification task, the training dataset is chosen such that there is an equal amount of signal and background events.

For reconstruction, a data selection with equal amounts of neutrino flavours have been chosen to maximize statistics. A more balanced set between tracks and cascades may be beneficial and will be explored in the future. On this dataset, energy, zenith, azimuth, and interaction vertex are regressed.

The normalized distributions of regression targets on test and training set from Table 2 are plotted in Fig. 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) [24] curves from which the Area Under the Curve (AUC) [25] is calculated. A typical threshold in classification score is indicated in red in the left panel of Fig. 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 14.9% at the same FPR, or alternatively, the FPR can be reduced by close to a factor of 6 from 5.0% to 0.85% at the same TPR.

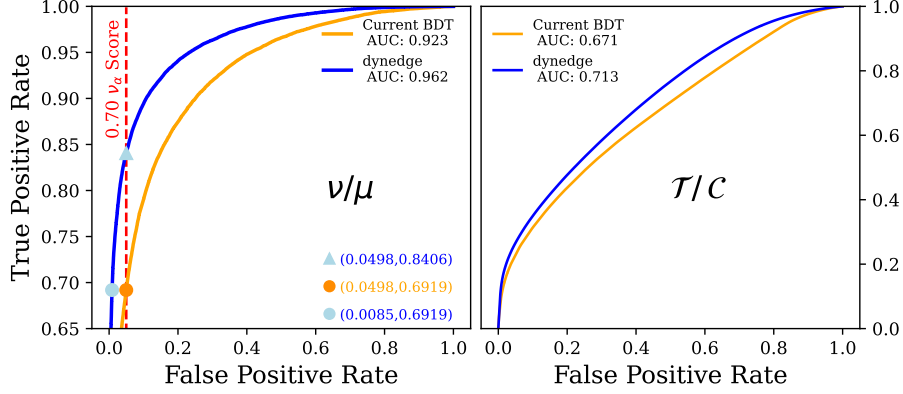


Figure 3: **Left:** ROC curves for DYNEDGE and current BDT ν/μ classification models. **Right:** ROC curve for DYNEDGE and current BDT T/C classification models.

The large difference in classification performance shown in Fig. 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, which reduces the information available for the BDTs. In the right panel of Fig. 3 it can be seen that DYNEDGE offers improvement in the T/C classification task with respect to the current BDT classifier, although with smaller margins than for ν/μ .

3.3 Event Reconstruction Performance of DYNEDGE

The event reconstruction performance for deposited energy, zenith, direction, and interaction vertex are shown in Fig. 5. The performance measure is the resolution, which we quantify here as the width W of the residual distribution R . For the deposited energy and zenith it 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 is more robust to outliers in the residual distribution than the standard deviation of the sample.

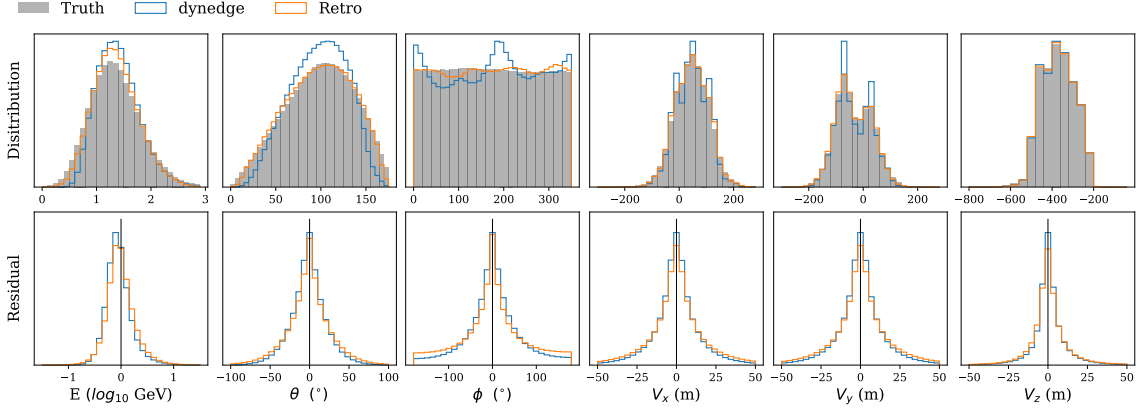


Figure 4: Area normalized distributions of predicted and true target variables on the reconstruction test set and their corresponding residual distributions. Residuals shown here follows the definitions shown in Table 2 with the exception of vertex coordinates.

The residual distributions for deposited energy (R_E), angles (R_{angle}), interaction vertex (R_{xyz}), and direction (R_{dir}) are defined in Table 2 and some are shown in Fig. 4. From Fig. 4 it can be 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. On average, it will be optimal for a machine learning model to estimate a value close to the mean of the true distribution for difficult examples. This behavior is evident for DYNEDGE in the reconstructions of zenith and azimuth, where azimuth picks up multimodal artifacts due to its the cyclic nature.

For the bounded residual distributions for direction and vertex, 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 bin the residual distributions in energy. Additionally, we separate the residual distributions in track and cascade events to examine the performance of the algorithms in detail.

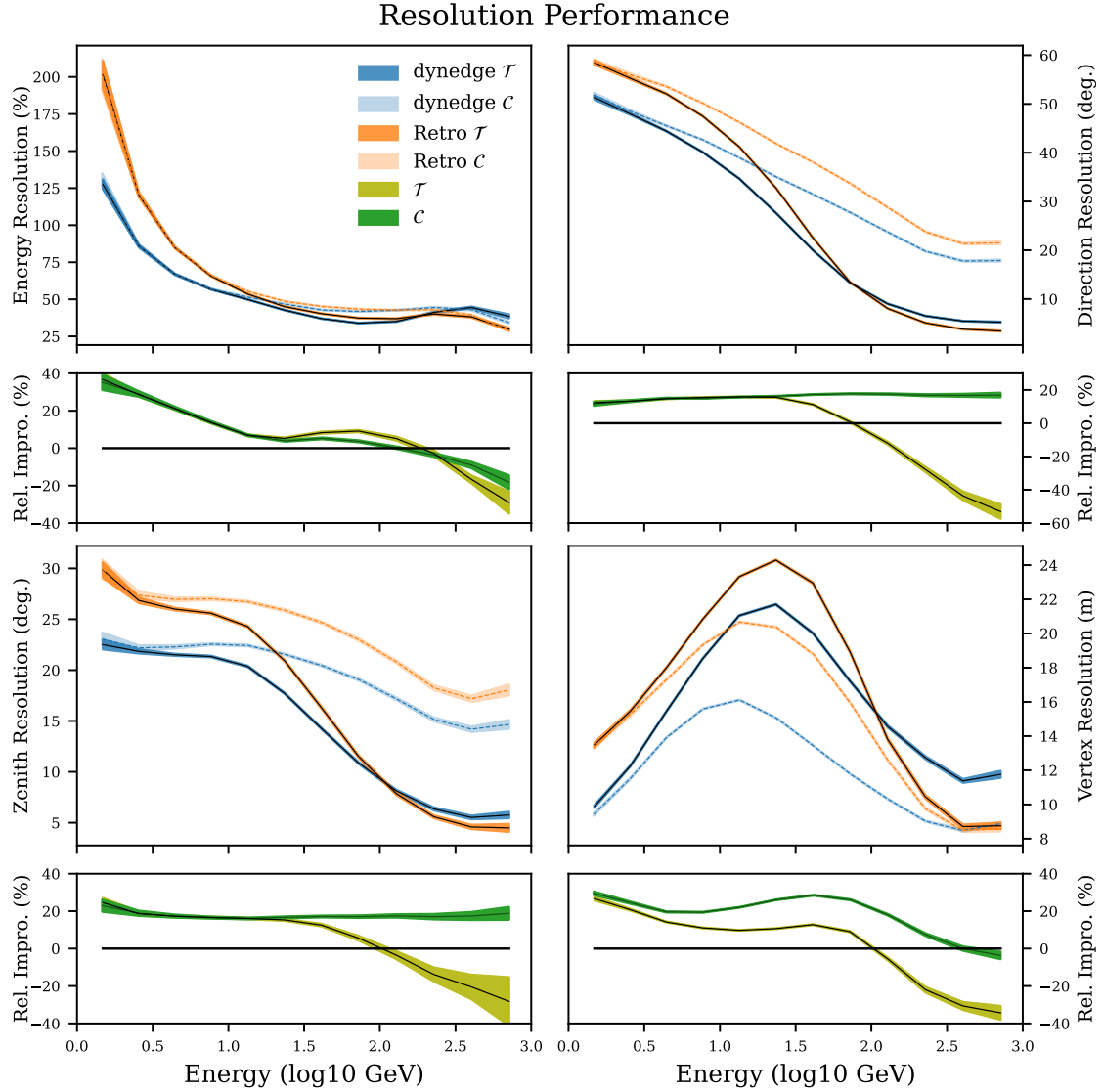


Figure 5: Event reconstruction performance for DYNEDGE, estimating deposited energy (top left), zenith angle (top right), direction (bottom left), and interaction vertex (bottom right). In all four cases, the performance is compared to RETRO, and the ratio below the plots shows relative improvement of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

As can be seen from Fig. 5, DYNEDGE performance improves on the existing RETRO reconstruction in all variables and across all signatures between 1 to 30 GeV. This energy range is of particular importance to neutrino oscillation analyses, where flavour oscillations of atmospheric neutrinos start to appear at 25 GeV and below [8]. 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 Fig. 4. As mentioned in Section 3.1, the reconstruction event selection is chosen to maximize available statistics with an equal amount of neutrino flavors, as opposed to an optimized selection. This choice improves on the overall performance under the given circumstances, but at the same time leads to and underrepresentation of track-like events which further lower the available statistics for this event type at higher energies.

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 events⁴. 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 0.22 ± 0.01 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 online reconstruction of IceCube events using a single GPU.

4 Robustness Test

The results presented in Section 3 show that DYNEDGE is a strong contender 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, *Perturbation of Input Variables*, we investigate the robustness of DYNEDGE to perturbations of input variables D_{xyz} , t and q . These variables constitutes 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, *Variations in Ice Properties and Module Acceptance*, we investigate the robustness of DYNEDGE to changes in systematic uncertainties associated with the detector medium and the angular photon acceptance of the DOMs. Variations

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.

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 errors 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 meter⁵. 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 . For the DOM positions, one ϵ is drawn for each string, resulting in correlated shifts among a string, and uncorrelated shifts between strings. The horizontal and vertical positions are also independently perturbed. Time and charge are perturbed pulse-wise, meaning that each pulse is perturbed independently.

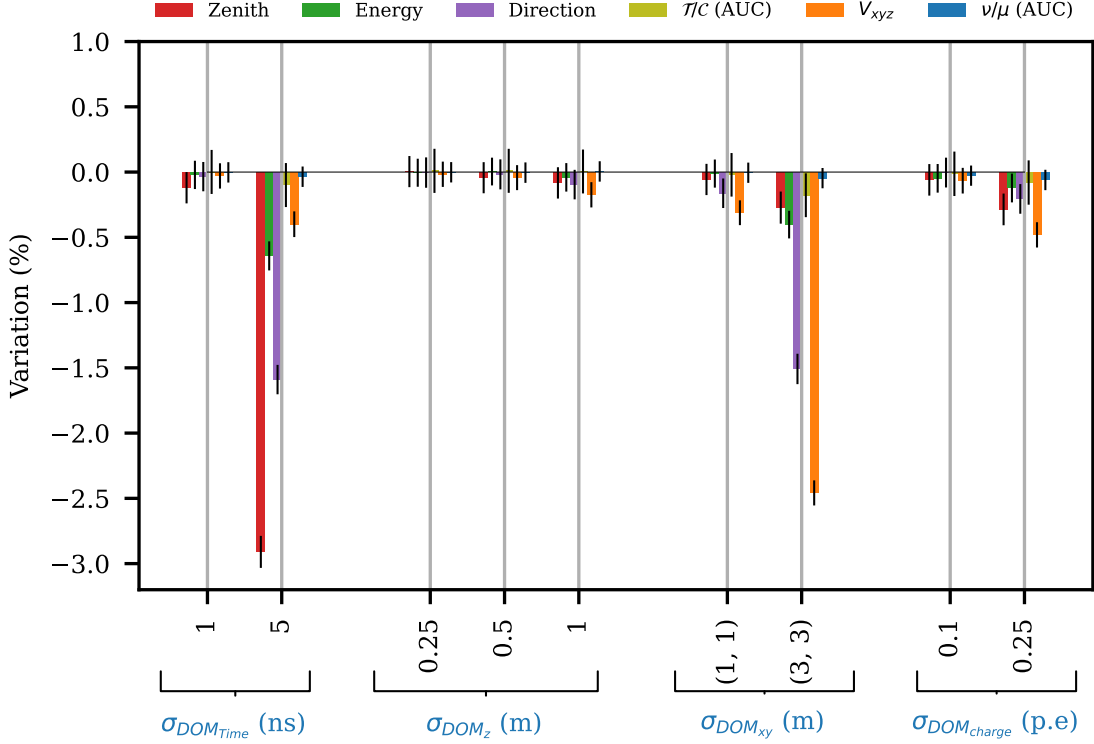


Figure 6: The variation in resolution and AUC induced by perturbation of input variables for DYNEDGE. The standard deviations of the normal distributions from which the perturbations are drawn are shown in the x-axis, and the 4 different perturbation tests are marked in blue.

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.

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 Fig. 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, a resolutions variation by -5% means a worsened resolution with a width W that is 5% larger. For AUC scores, a variation by -5% means a decrease by 5%. As seen in Fig. 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 altered optical properties with respect to the bulk ice. These sources of systematic uncertainty are constrained from fits to calibration data, but only to a finite precision. 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 consider 20 additional simulation sets with altered ice properties and module acceptance [oscNext]. 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 module’s angular acceptance.

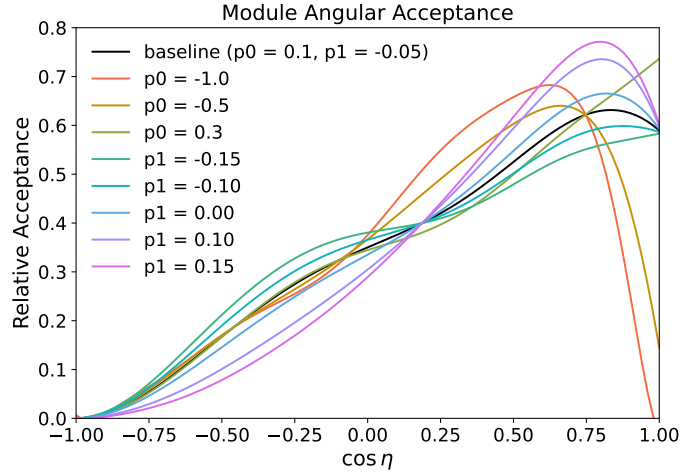


Figure 7: Variations in the DOM angular acceptance used for the robustness tests. The angle η is the photon arrival angle at the Module, with $\cos \eta = 1$ being head-on towards the face of the PMT and $\cos \eta = 0$ towards the backside of the PMT where generally the acceptance drops to zero.

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 acceptance⁶. As seen in Fig. 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 set 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 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 current BDT. 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

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

410 results are shown in Fig. 8.

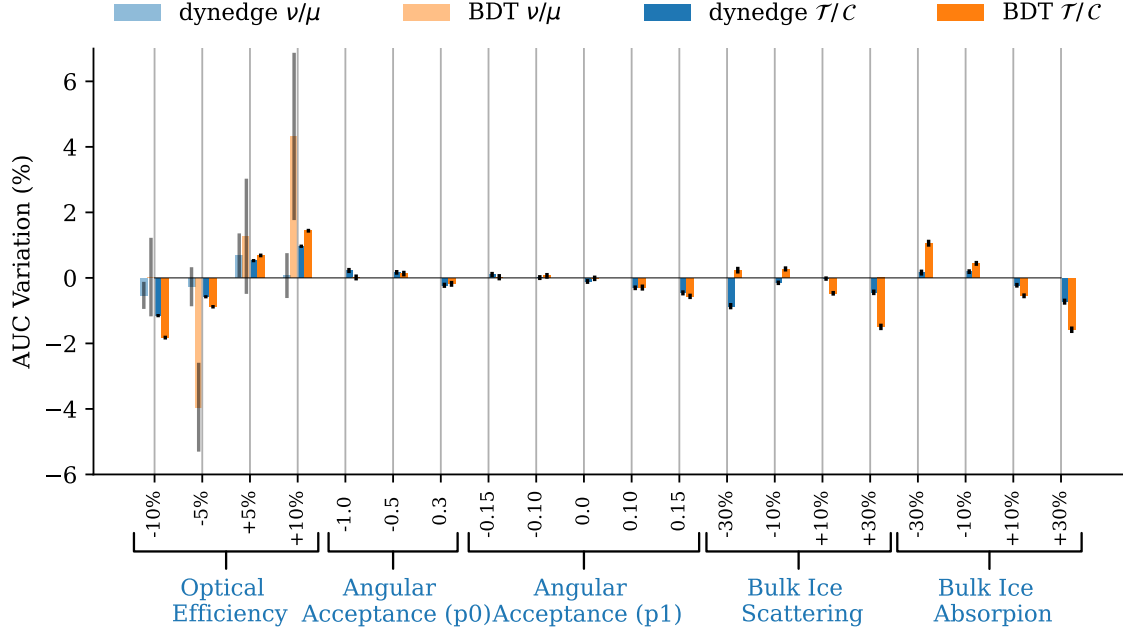


Figure 8: The variation in AUC for DYNEDGE and the current BDT on the 20 different systematic sets. Track/cascade classification is denoted by \mathcal{T}/C and neutrino/muon classification is denoted by ν/μ . Please note that variation in ν/μ AUC is only shown on the first 4 samples, as the rest of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

As seen from Fig. 8, the error on AUC variation for ν/μ is much higher than for \mathcal{T}/C . This is due to the lower statistics on 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 state-of-the-art.

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.

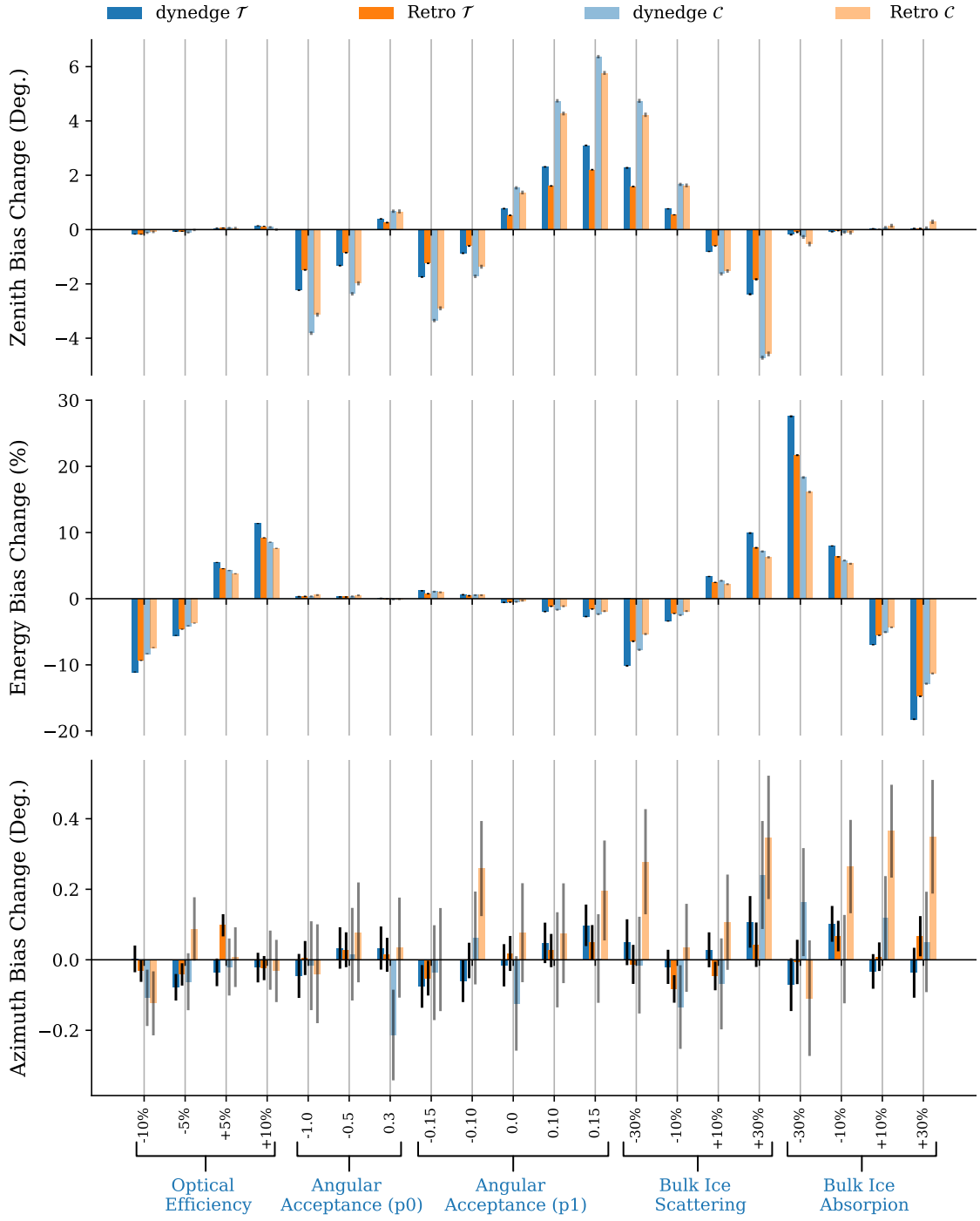


Figure 9: The variation in bias for DYNEDGE and RETRO on the 20 different systematic sets for reconstruction targets energy, zenith, and direction. Please notice that the y-axis values for azimuth is much smaller than for zenith.

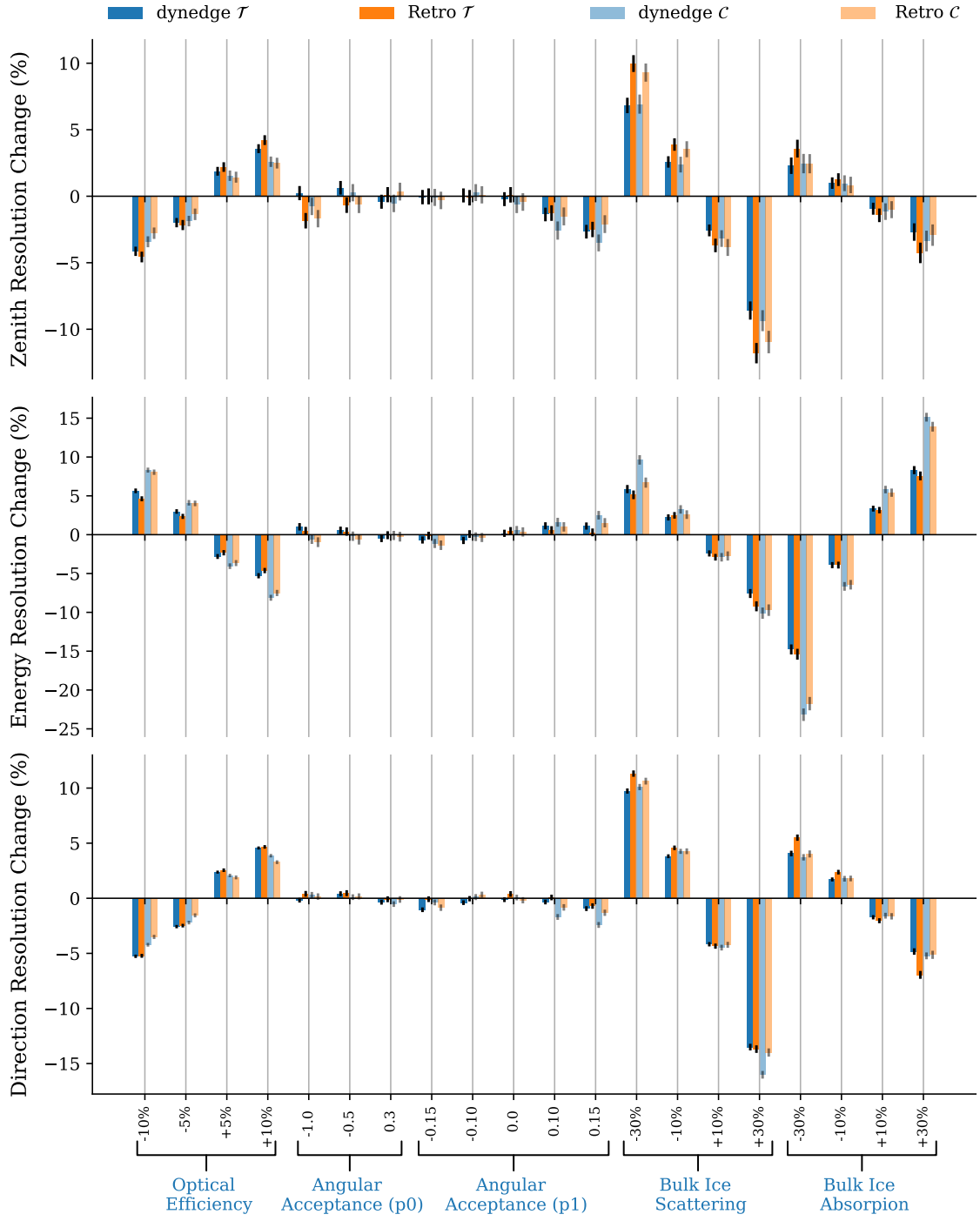


Figure 10: The variation in resolution for DYNEDGE and RETRO on the 20 different systematic sets for reconstruction targets energy, zenith, and direction.

Fig. 9 shows the bias change for energy, zenith, and azimuth for the different simulation variations, while Fig. 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 for energy regression is highly affected by the optical efficiency and scattering/absorption of light, but less sensitive to angular photon acceptance. Such an effect is expected as the deposited energy is correlated with the number of measured pulses. Specifically, it is heartening to see that a linear change in DOM efficiency leads to an approximately linear change in energy bias for both methods. As seen in Fig. 10, the overall effect on reconstruction resolution is nearly identical for between 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/the current BDT classifiers, the RMS of the variations shown in Figs. 8 to 10 is presented in Table 4.

Target	DYNEDGE (RMS)		RETRO/BDT (RMS)	
	Tracks	Cascades	Tracks	Cascades
Energy Bias (%)	9.1	6.4	7.2	5.6
Zenith Bias (°)	1.4	3.2	1.0	2.4
Azimuth Bias (°)	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}/C AUC (%)	0.5		0.8	

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.)

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 it can be observed that 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, with the only clear exception being the resolution of the vertex regression.

5 Summary and Conclusions

We propose a GNN-based reconstruction algorithm for IceCube events named DYNEDGE, that applies to any IceCube event. We have selected simulated low-energy data used for studies of

atmospheric neutrino oscillation 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.

We show that DYNEDGE offers substantial improvements to both \mathcal{T}/C and ν/μ classifications in the entire low energy range. In the oscillation relevant energy range of 1–30 GeV we have demonstrated a 15–20 % improvement in reconstruction resolution of energy, zenith, direction, and interaction vertex. While reconstruction resolutions see substantial improvements in the oscillation-relevant energy range, the improvement decreases at higher energies, and in some cases 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 at this region.

To study the robustness of DYNEDGE to variations in known systematic detector uncertainties, we compared performance to the currently-in-use reconstruction algorithms under variations of certain key systematics. We show that the overall robustness to variations in optical efficiency, angular acceptance and bulk ice scattering and absorption is similar between DYNEDGE and RETRO. Furthermore, a study with the random perturbation of DYNEDGE’s inputs underlies the robustness of the algorithm.

Finally, we provide an examination of the reconstruction speed and show that DYNEDGE is in principle capable of online reconstruction of events 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 and the proposed IceCube Gen2.

References

- [1] J. Kiryluk. *Neutrino Physics with the IceCube Detector*. 2008. arXiv: [0806.1717 \[astro-ph\]](#) (cit. on p. 2).
- [2] R. Abbasi et al. “The design and performance of IceCube DeepCore”. In: *Astroparticle Physics* 35.10 (May 2012), pp. 615–624. ISSN: 0927-6505. DOI: [10.1016/j.astropartphys.2012.01.004](#). URL: <http://dx.doi.org/10.1016/j.astropartphys.2012.01.004> (cit. on pp. 2–4).
- [3] Francis Halzen and Spencer R. Klein. “IceCube: An Instrument for Neutrino Astronomy”. In: *Rev. Sci. Instrum.* 81 (2010), p. 081101. DOI: [10.1063/1.3480478](#). arXiv: [1007.1247 \[astro-ph.HE\]](#) (cit. on p. 2).
- [4] R. Abbasi et al. “The Design and Performance of IceCube DeepCore”. In: *Astropart. Phys.* 35 (2012), pp. 615–624. DOI: [10.1016/j.astropartphys.2012.01.004](#). arXiv: [1109.6096 \[astro-ph.IM\]](#) (cit. on p. 2).

- [5] M.G. Aartsen et al. “The IceCube Neutrino Observatory: instrumentation and online systems”. In: *Journal of Instrumentation* 12.03 (Mar. 2017), P03012–P03012. ISSN: 1748-0221. DOI: [10.1088/1748-0221/12/03/p03012](https://doi.org/10.1088/1748-0221/12/03/p03012). URL: <http://dx.doi.org/10.1088/1748-0221/12/03/p03012> (cit. on p. 2).
- [6] P.A. Zyla et al. “Review of Particle Physics”. In: *PTEP* 2020.8 (2020), p. 083C01. DOI: [10.1093/ptep/ptaa104](https://doi.org/10.1093/ptep/ptaa104) (cit. on p. 2).
- [7] M. G. Aartsen et al. “Measurement of Atmospheric Neutrino Oscillations at 6–56 GeV with IceCube DeepCore”. In: *Physical Review Letters* 120.7 (Feb. 2018). ISSN: 1079-7114. DOI: [10.1103/PhysRevLett.120.071801](https://doi.org/10.1103/PhysRevLett.120.071801). URL: <http://dx.doi.org/10.1103/PhysRevLett.120.071801> (cit. on p. 4).
- [8] M. G. Aartsen et al. “Measurement of atmospheric tau neutrino appearance with IceCube DeepCore”. In: *Physical Review D* 99.3 (Feb. 2019). ISSN: 2470-0029. DOI: [10.1103/PhysRevD.99.032007](https://doi.org/10.1103/PhysRevD.99.032007). URL: <http://dx.doi.org/10.1103/PhysRevD.99.032007> (cit. on pp. 4, 10, 13).
- [9] J. Ahrens et al. “Muon track reconstruction and data selection techniques in AMANDA”. In: *Nucl. Instrum. Meth. A* 524 (2004), pp. 169–194. DOI: [10.1016/j.nima.2004.01.065](https://doi.org/10.1016/j.nima.2004.01.065). arXiv: [astro-ph/0407044](https://arxiv.org/abs/astro-ph/0407044) (cit. on p. 4).
- [10] R. Abbasi et al. “Low Energy Event Reconstruction in IceCube DeepCore”. In: (Mar. 2022). arXiv: [2203.02303](https://arxiv.org/abs/2203.02303) [[hep-ex](#)] (cit. on pp. 4, 10).
- [11] Aya Ishihara. *The IceCube Upgrade – Design and Science Goals*. 2019. arXiv: [1908.09441](https://arxiv.org/abs/1908.09441) [[astro-ph.HE](#)] (cit. on p. 5).
- [12] Vedant Basu et al. “A next-generation optical sensor for IceCube-Gen2”. In: *Proceedings of 37th International Cosmic Ray Conference — PoS(ICRC2021)* (July 2021). DOI: [10.22323/1.395.1062](https://doi.org/10.22323/1.395.1062). URL: <http://dx.doi.org/10.22323/1.395.1062> (cit. on p. 5).
- [13] R. Abbasi et al. “A convolutional neural network based cascade reconstruction for the IceCube Neutrino Observatory”. In: *Journal of Instrumentation* 16.07 (July 2021), P07041. ISSN: 1748-0221. DOI: [10.1088/1748-0221/16/07/p07041](https://doi.org/10.1088/1748-0221/16/07/p07041). URL: <http://dx.doi.org/10.1088/1748-0221/16/07/p07041> (cit. on p. 5).
- [14] Nicholas Choma et al. *Graph Neural Networks for IceCube Signal Classification*. 2018. arXiv: [1809.06166](https://arxiv.org/abs/1809.06166) [[cs.LG](#)] (cit. on p. 5).
- [15] Zhiqian Chen et al. *Bridging the Gap between Spatial and Spectral Domains: A Survey on Graph Neural Networks*. 2020. arXiv: [2002.11867](https://arxiv.org/abs/2002.11867) [[cs.LG](#)] (cit. on p. 6).
- [16] Yann A. LeCun et al. “Efficient BackProp”. In: *Neural Networks: Tricks of the Trade: Second Edition*. Ed. by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–48. ISBN: 978-3-642-35289-8. DOI: [10.1007/978-3-642-35289-8_3](https://doi.org/10.1007/978-3-642-35289-8_3). URL: https://doi.org/10.1007/978-3-642-35289-8_3 (cit. on p. 6).

- [17] Yue Wang et al. *Dynamic Graph CNN for Learning on Point Clouds*. 2019. arXiv: [1801.07829 \[cs.CV\]](#) (cit. on p. 6).
- [18] Hongbin Pei et al. *Geom-GCN: Geometric Graph Convolutional Networks*. 2020. arXiv: [2002.05287 \[cs.LG\]](#) (cit. on p. 6).
- [19] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: [1412.6980 \[cs.LG\]](#) (cit. on p. 8).
- [20] Leslie N. Smith and Nicholay Topin. *Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates*. 2018. arXiv: [1708.07120 \[cs.LG\]](#) (cit. on p. 8).
- [21] Liyuan Liu et al. *On the Variance of the Adaptive Learning Rate and Beyond*. 2021. arXiv: [1908.03265 \[cs.LG\]](#) (cit. on p. 8).
- [22] Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. “Early stopping for non-parametric regression: An optimal data-dependent stopping rule”. In: *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2011, pp. 1318–1325. DOI: [10.1109/Allerton.2011.6120320](#) (cit. on p. 8).
- [23] Sachin Kumar and Yulia Tsvetkov. *Von Mises-Fisher Loss for Training Sequence to Sequence Models with Continuous Outputs*. 2019. arXiv: [1812.04616 \[cs.CL\]](#) (cit. on p. 9).
- [24] Tom Fawcett. “An introduction to ROC analysis”. In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861–874. ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2005.10.010>. URL: <https://www.sciencedirect.com/science/article/pii/S016786550500303X> (cit. on p. 10).
- [25] Donna McClish. “Analyzing a portion of the ROC Curve”. In: *Medical decision making : an international journal of the Society for Medical Decision Making* 9 (Aug. 1989), pp. 190–5. DOI: [10.1177/0272989X8900900307](#) (cit. on p. 10).