Prepared for submission to JHEP

Classifying neutrinos in a long-baseline neutrino oscillation experiment

R. Kalra

University College London,
London, UK

Abstract: Accurate identification of neutrino properties (flavour, interaction, scattering type etc.) is crucial to the study of neutrino oscillations in the NoVA experiment. This study describes an application and comparison of a MVCNN (Multi-View Convolutional Neural Network) algorithm to to classify muon neutrino (ν_μ) charged-current events based solely on photo-detector images, resulting in satisfactory performance (F1 score: 0.82). Adaptations of the MVCNN were applied to regression of neutrino energy (MAE: 6.12) and lepton to neutrino energy ratio (MAE: 0.16), as well as classification of neutrino flavour (F1 score: 0.83) and scattering type (F1 score: 0.48).

Contents

1 Introduction

The categorisation of particle interactions is a prevalent issue in the field of high-energy physics (HEP) [1–3]. Previous methods of classification involved reconstructing high-level features such as clusters, tracks, showers, jets, and rings that are associated with particle interactions recorded by the detector and summarising the energies, directions, and shapes of these objects [4]. This study examines the effectiveness of using convolutional neural networks (CNN) to classify neutrino events based on simulated NoVA-like neutrino detector images. The objective is apply a CNN model to accurately classify the images muon neutrino (ν_μ) charged-current events.

Neutrino Physics and The NoVA Experiment

Neutrinos are one of the fundamental particles in the universe. They are leptons, have a neutral electric charge, and are considered almost massless [5]. Currently, three types (flavours) of neutrinos are known to exist: the electron neutrino (ν_e), muon neutrino (ν_μ), and tau neutrino (ν_τ). These pair with charged lepton partners: the electron (e^-), muon (μ^-), and tau (τ^-), respectively. Neutrinos interact extremely rarely via two categories of weak processes [6]. The first is the charged-current (CC) interaction where a neutrino transforms into its charged partner i.e. $\nu_\mu \rightarrow \mu^-$. The second is the neutral-current (NC) interaction, where a neutrino scatters and does not change flavour. Neutrinos can interact with nucleons, which are made up of quarks and gluons, in different ways. A CC interaction has sub-types of interaction with distinct features. For instance, quasi-elastic (QE) events are considered 'clean' events, usually producing two tracks [7], while deep inelastic scattering (DIS) events [8] are 'messy' events that may produce many tracks and showers. Resonant (RES) events possess a mix of features from both QE and DIS events.

Neutrino oscillation is a phenomenon where a neutrino produced in one flavour can change

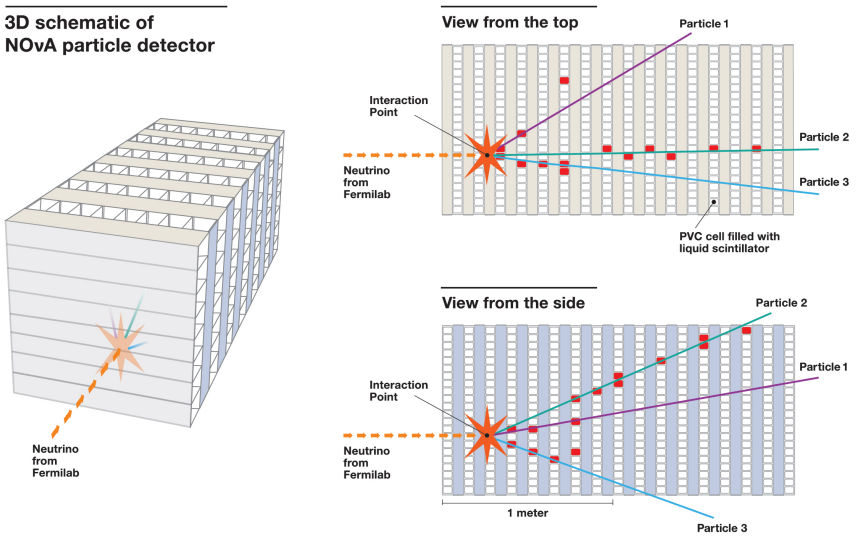


Figure 1: Schematic of the NOvA detector with top and side view neutrino tracks detailed

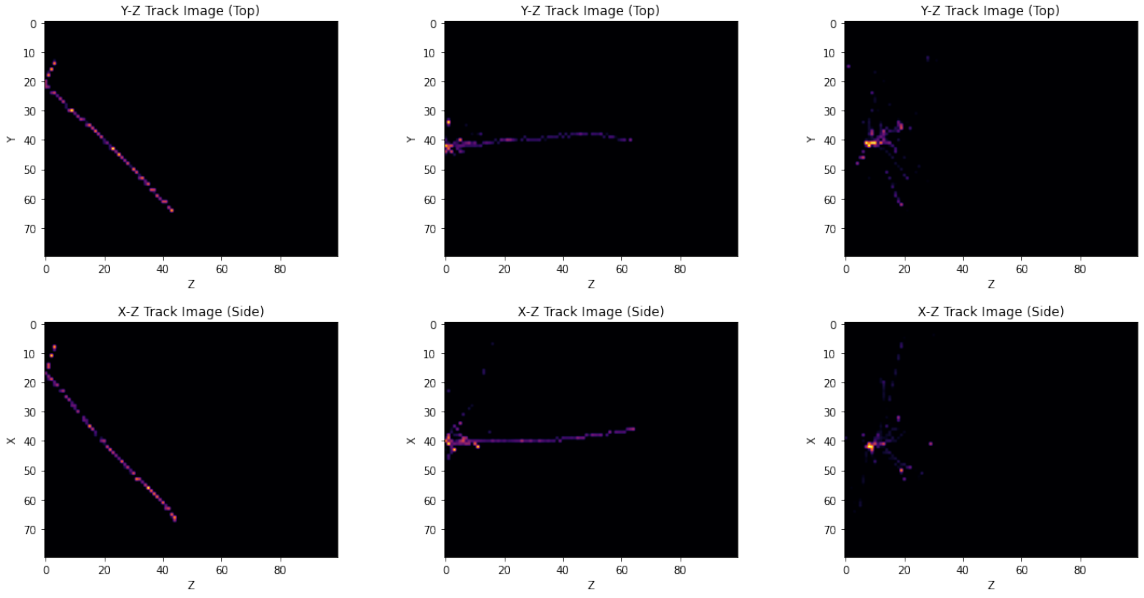


Figure 2: Side (x-z) and top (y-z) views of photo-detector images, including quasi-elastic (left), resonant (centre) and deep-inelastic (right) scattering

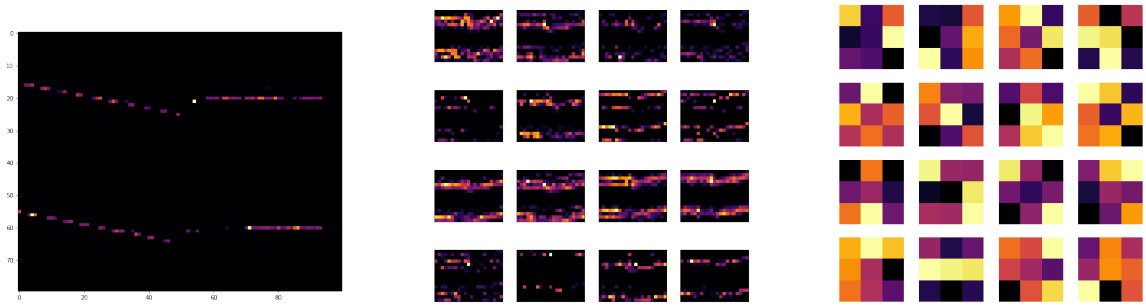
into a different flavour as it travels through space, due to quantum mechanical mixing of its mass eigenstates. [9, 10]. It is one of the only known processes observed to break the Standard Model [11]. The NuMI Off-axis ν_e Appearance (NOvA) experiment is a long-baseline neutrino oscillation experiment [12]. It uses a beam of muon neutrinos with energies of a few GeV, and consists of two detectors: a 300 metric-ton near detector at Fermilab and a larger 14 metric-kiloton far detector in Minnesota [12–14]. The far detector consists of 344,000 extruded plastic PVC cells filled with liquid scintillator [14]. When a

neutrino interacts with the scintillator, charged particles are released and their energy collected by photo-detectors connected to wavelength-shifting fibers. Photo-detector images are collected in an x-y and x-z view, as described in figure 1. Resulting image examples are shown in figure 2. The experiment aims to study three flavor neutrino oscillations by analysing the charged particles produced when neutrinos interact with the detectors, providing insight into the flavor, interaction type, and energy of the neutrinos. Particularly, the proportion (ν_μ) events at the far detector can be used to determine whether oscillation has occurred.

Neural Networks

An artificial neural network (ANN) is a collection of interconnected artificial neurons that process information using a connected graph approach. Connections between neurons are modelled using numerical weights and biases [15, 16]. In high-energy physics, the multilayer perceptron (MLP) is a commonly used ANN. It consists of an input layer, one or more hidden layers, and an output layer, where all layers are fully connected [17]. A non-linear function is applied to the weighted sum of the outputs of nodes in the previous layer, adding a bias term. Supervised learning is typically used to determine the weights and biases of the MLP by providing it with inputs and known labels. The MLP's output is compared to the truth values, and the error, or loss, is calculated using backpropagation, which computes the gradient of the loss function with respect to the weights and biases. The MLP has found success, however the full connectivity of the MLP can lead to overfitting and difficulties in adapting to variations in data [17].

Deep learning is a class of machine learning algorithms using multiple layers to extract progressively higher level features from the input [17, 18]. Convolutional neural networks (CNNs) are a type of deep neural network specifically designed to work with image data. CNNs are inspired by the organisation and functioning of the visual cortex of animals, which uses hierarchical processing of visual stimuli to identify objects and their features [19, 20]. They use several convolutional layers to identify different image features such as edges in lower layers and digits, letters or faces in higher layers [21, 22]. The layers apply a set of learnable filters [23] (typically small in size compared to the input image) and are convolved with the input to produce feature maps. The output of the convolutional layers is passed through a series of non-linear activation functions, such as ReLU [24], to introduce non-linearity and enhance the network's representational power. The feature maps are processed by pooling layers, which downsample the feature maps and reduce the spatial dimensionality of the input while preserving the most important information [23]. An example of an image, convolutional outputs and filters is shown in figure 3. CNNs may also include regularisation techniques such as dropout [25], batch normalisation [26] and L1 or L2 regularisation to prevent overfitting. The cross-entropy (binary and categorical) loss function is commonly used in classification tasks. The gradient of the loss function as a function of the weights and biases is calculated via backpropagation, and stochastic gradient descent or its variants are typically used to update the weights and biases [17]. Standard CNNs have certain limitations, including inability to effectively handle high-



(a) Original image (b) Convolutional output (c) Convolutional filters

Figure 3: Example of a convolutional layer output and a 3x3 convolutional filter

resolution images and the limited ability to model long-range dependencies in sequential data [27, 28]. In some tasks, CNNs struggle with capturing both local and global features simultaneously, leading to suboptimal performance [27]. To address this, modifications to the CNN architecture have been proposed, such as the inception module [29] and the multi-view CNN [30].

2 Model Architecture

Standard CNN Model

First, a simple convolutional neural network was used. Image pairs were fed into the network together, with each view as a separate channel. 4 convolutional layers were used, each with 3x3 filters, and 32 neurons in the first layer and 64 neurons in all other layers. Max pooling (2D) with a pool size of 2x2 was used [22, 23]. The output layer used the binary cross-entropy loss function, for binary classification. To avoid overfitting, L2 regularisation was used on each layer, with a regularisation parameter of 0.001 [25]. The CNN model and component layers are shown in figure 4.

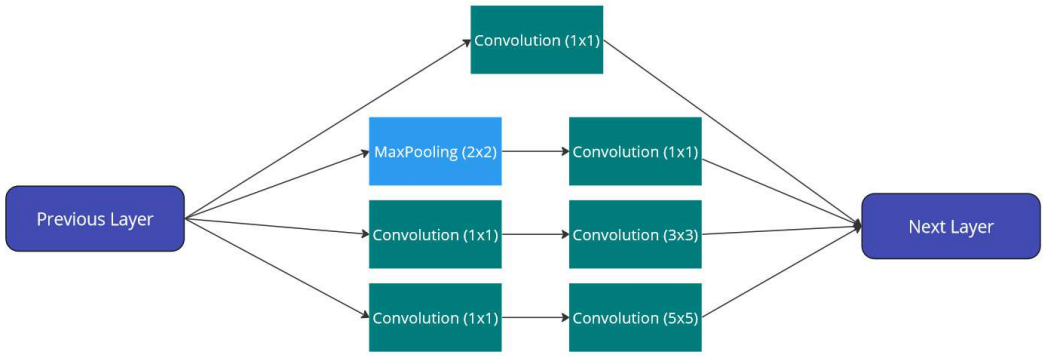
(a) Standard CNN model architecture

(b) Convolutional module architecture

Figure 4: Standard CNN model

Inception Model

Use of an inception module was tested. The module was proposed by C. Szegedy et al. [28] in 2015, to increase network depth and width and keeping computational budget constant. The module has been used in many CNN architectures, such as GoogLeNet, Inception-v3 [31], and Inception-ResNet [29]. It enables all 1x1, 3x3, and 5x5 filters, convolution and pooling to be performed in parallel on the output from previous layers. This allows capture of both local and global features, and different levels of abstraction [31]. The filter outputs are concatenated and inputted subsequent layer, as shown in figure 5a. To reduce the total parameters in the network, a 1x1 convolutional layer is used to reduce dimensionality before applying larger convolutions. The model architecture included the inception module between two 3x3 convolutional layers, as shown in figure 5b.



(a) Inception module architecture

(b) Inception model architecture

Figure 5: Inception model

Multi-View CNN Model

A Multi-view convolutional neural network (MVCNN) was implemented, based on the model proposed by Su et al. in 2015 [30]. The MVCNN utilised two branches, one for the side and one for the top view images, to avoid the model from considering the two views as color channels of the same image. Compared to a normal CNN architecture, which typically takes a single image or volume as input, an MVCNN is able to leverage multiple perspectives of an object or scene to improve performance. By incorporating information from multiple views, an MVCNN can learn more robust representations that are invariant to changes in viewpoint, lighting, and other factors that might affect a single view. Each branch used an identical architecture, as shown in figure 6, with 16 neurons for each convolutional layer, as determined through a manual grid search. Max pooling layers were used to reduce the dimensionality of the images, and dropout layers with a 30% rate were added after each convolutional layer to avoid overfitting. The final model

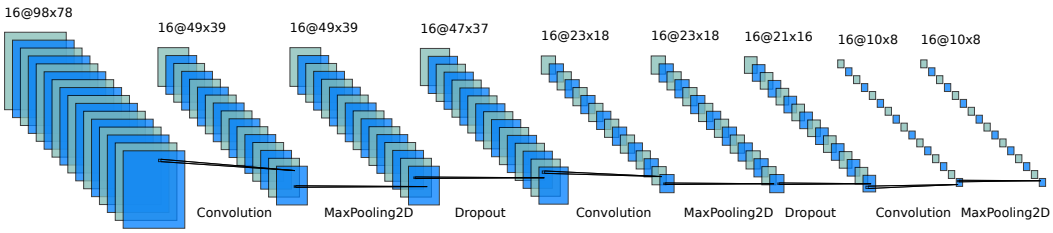


Figure 6: Diagram of the CNN architecture for a single branch of the Multi-View CNN model

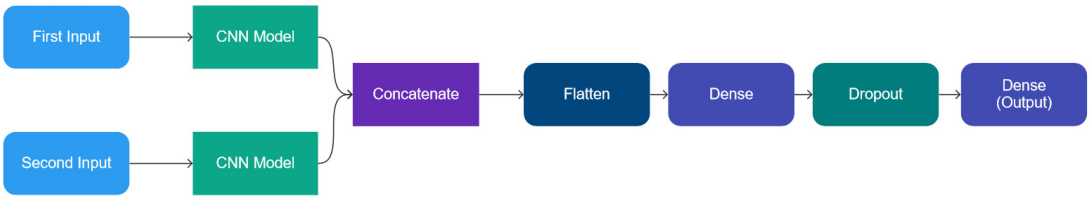
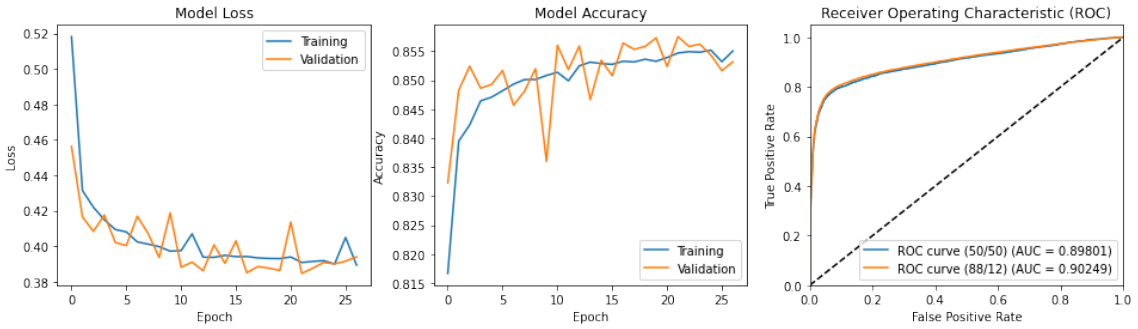


Figure 7: Multi-view CNN model architecture

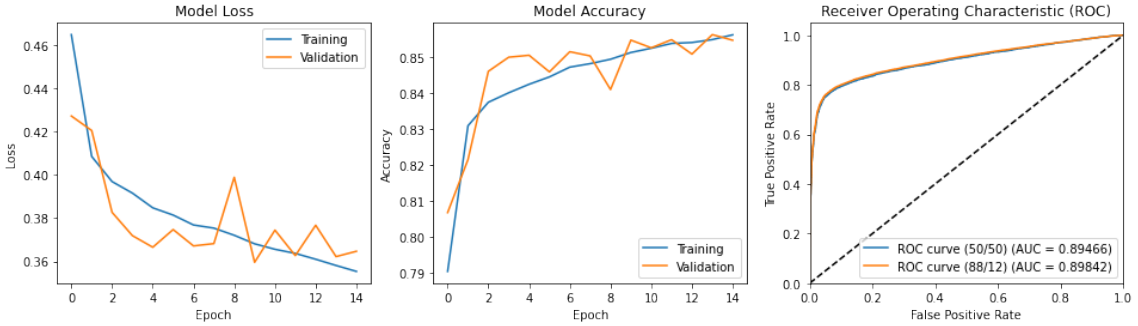
architecture is shown in figure 7. Testing different numbers of inception modules (up to 3) in each branch of the network showed no significant improvement in accuracy above the original convolutional layers.

3 Model Training

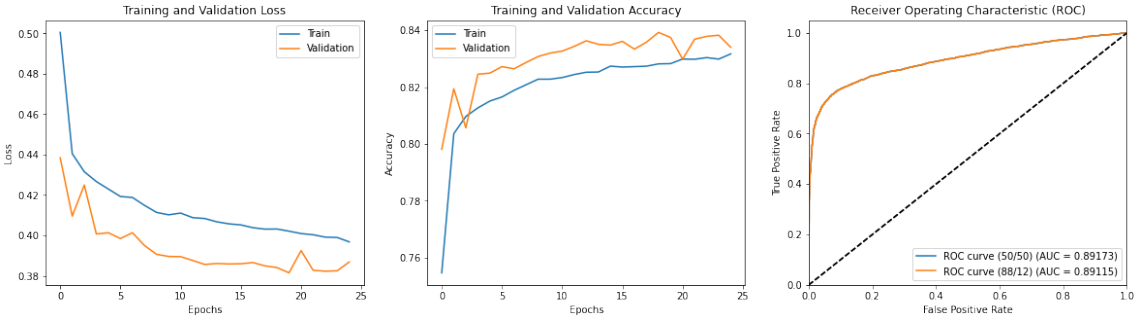
The original dataset consisted of 700,000 image pairs. The imbalanced data (88% ν_μ CC events) was addressed by undersampling excess data. Oversampling/data augmentation was not used as it resulted in overfitting for all tested models. Training images were normalised to the range 0 to 1 where computationally viable. This left 164,000 image pairs, with 80% used for training and 20% for validation. A separate dataset of 24,000 image pairs was used for testing. Early stopping callbacks with a patience of 5 epochs was used to avoid overtraining. Models were tested on both the rebalanced dataset and the unbalanced dataset. Each model's test loss, accuracy, and receiver operating characteristic curve are shown in figure 8a, 8b and 8c. All models had similar test loss and accuracy (within $\pm$ 0.02). The ROC curve is a plot of true positive rate (TPR) against false positive rate (FPR) [32, 33]. The diagonal ($y = x$) reference line represents false positive result being produced at the same rate as positive results [33]. A classifier with reasonable accuracy must have an ROC curve in the upper left above the reference line [32]. Performance can therefore be measured using the area under the curve (AUC) [34]. ROC curves for all models had AUCs > 0.89. However, the MVCNN model had a significantly fewer total



(a) Standard CNN model loss vs epochs (left), accuracy vs epochs (middle) and ROC curve for the balanced and original test dataset (right)



(b) Inception CNN model loss vs epochs (left), accuracy vs epochs (middle) and ROC curve for the balanced and original test dataset (right)



(c) Multi-view CNN model loss vs epochs (left), accuracy vs epochs (middle) and ROC curve for the balanced and original test dataset (right)

Figure 8: Comparison of the different model performance metrics

parameters (two orders of magnitude less) than other models. Therefore, the MVCNN was chosen as the optimal model. Test results for each model are presented in Table 1.

Model	Total Parameters	Test Loss	Test Accuracy	AUC
Standard CNN	2,141,089	0.3946	0.8439	0.8980
Inception CNN	5,424,315	0.3756	0.8418	0.8947
Multi-View CNN	30,097	0.3815	0.8391	0.8917

Table 1: Table of results for each of the tested models. Area under the ROC curve (AUC) is for the balanced test set. 24,000 events were used for testing.

4 Testing MVCNN Performance

Classifier performance may be measured using the F1 score, precision and recall [35]. Precision is the fraction of true positive predictions out of all positive predictions, allowing measurement of accuracy of positive predictions. Recall (sensitivity) is the fraction of true positive predictions out of all actual positives in the data, which measures the model's ability to identify all positive instances. F1 score is the harmonic mean of precision and recall and it is used as a combined measure of the two. It gives equal weight to precision and recall, and its value ranges from 0 to 1, with 1 being the best possible score. The precision-recall curve shows the trade-off between the two at different classification thresholds. Generally, an ideal model would have high precision and high recall across all threshold values, resulting in a curve that hugs the upper-right corner of the graph. The MVCNN achieved a precision of 0.92 and a recall of 0.74 for classifying muon neutrino CC events. The F1-score for muon neutrino CC events was 0.82, indicating a good balance between precision and recall. The macro avg F1-score was 0.84. The precision-recall curve is shown in figure 9 and shows stable precision until approximately 0.65 recall.

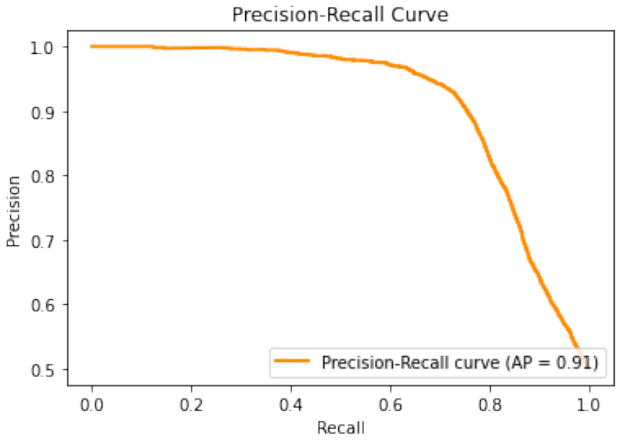


Figure 9: MVCNN classifier precision-recall curve

The classifier was tested on several datasets with different types of events. Firstly, it was tested on DIS and QE events. Performance was expected to be better on QE events due to the 'cleaner' tracks. The MVCNN was evaluated before and after re-training on each

DIS and QE datasets. Before training, accuracy on the DIS test set was 0.8360 with loss of 0.3856, while accuracy on the QE set was 0.9375 with loss of 0.2085. After re-training, there was no improvement in loss or accuracy for both QE and DIS sets. The model's 0.10 deviation in accuracy between QE and DIS data suggests it more easily recognises 'cleaner' QE event tracks.

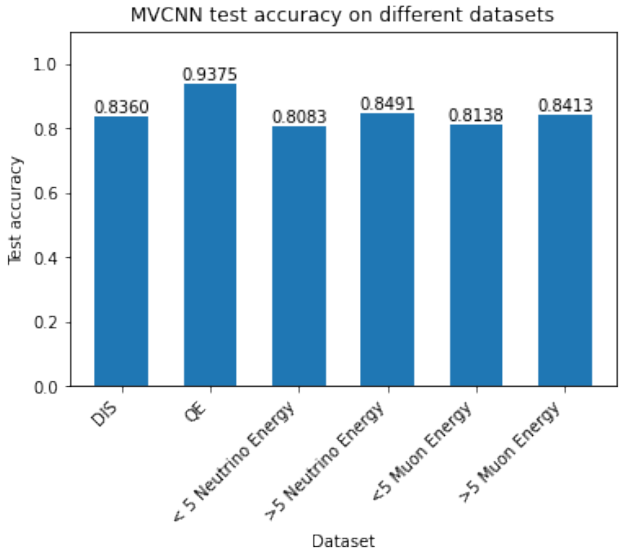


Figure 10: MVCNN classifier performance on different datasets

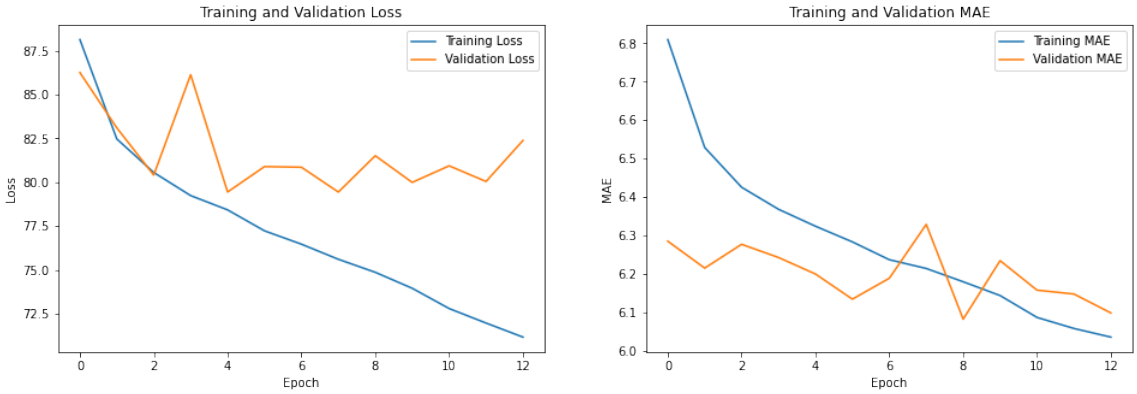
Performance was tested on high energy (> 5 GeV) and low energy (< 5 GeV) neutrinos. High energy neutrinos were expected to exhibit brighter tracks due to more intense scintillation light in the detector. On low energy events, loss was 0.4410 and accuracy 0.8083, while on high energy events, loss was 0.3694 and accuracy 0.8491. Re-training for 18 epochs (low energy) and 14 epochs (high energy) did not significantly improve performance. Performance was then tested on low (< 5 GeV) and high (> 5 GeV) muon energy data. Before re-training, test loss was 0.4326 and test accuracy was 0.8138 for the low energy set. Loss was 0.3870 and accuracy was 0.8413 for the high energy set. After re-training for 27 epochs, the low energy set improved to 0.3821 loss and 0.8370 accuracy, while the high energy set only slightly improved to 0.3726 loss and 0.8429 accuracy after 17 epochs. The MVCNN test accuracy for different datasets is shown in figure 10.

5 Further Investigation

Neutrino Energy Regression

The classifier was modified to estimate the energy of a neutrino from detector images. The number of filters per convolutional layer was changed to 7x7, 5x5, 3x3 for the first three layers (in that order). An inception module was added between the 2nd and 3rd convolutional layers. The number of neurons was changed to 128 for the first convolutional layer and 64 for the rest. The dense output layer was changed to 64 neurons. The output layer activation was linear, loss was mean squared error (MSE), and accuracy metric

was mean absolute error (MAE). 140,000 events were used for training, with an 80/20 training/validation split, and 35,000 events for testing. Dropout rates after convolutional layers were reduced to 20%, and before the output layer to 30% to retain more features. Validation ended with an MAE of 6.11 and MSE of 80.89, as shown in figure 11. Test loss (MSE) was 79.85 and test accuracy (MSE) was 6.12.

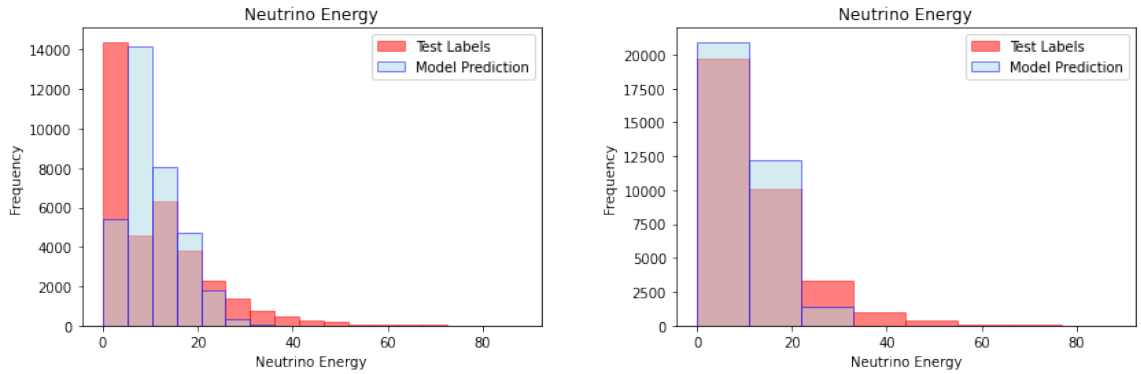


(a) Loss (mean squared error)

(b) Accuracy (mean absolute error)

Figure 11: Adapted MVCNN training metrics for neutrino energy regression

The distribution of predicted neutrino energies was compared to test labels and visualised in figure 12. Figure 12a and 12b demonstrate the significance of the mean absolute error (MAE), as the model tends to falsely predict low energy neutrinos (0-5 GeV). This issue is less apparent when the bin width is increased to 10, making the predictions appear satisfactory.



(a) Histogram with bin width of 5

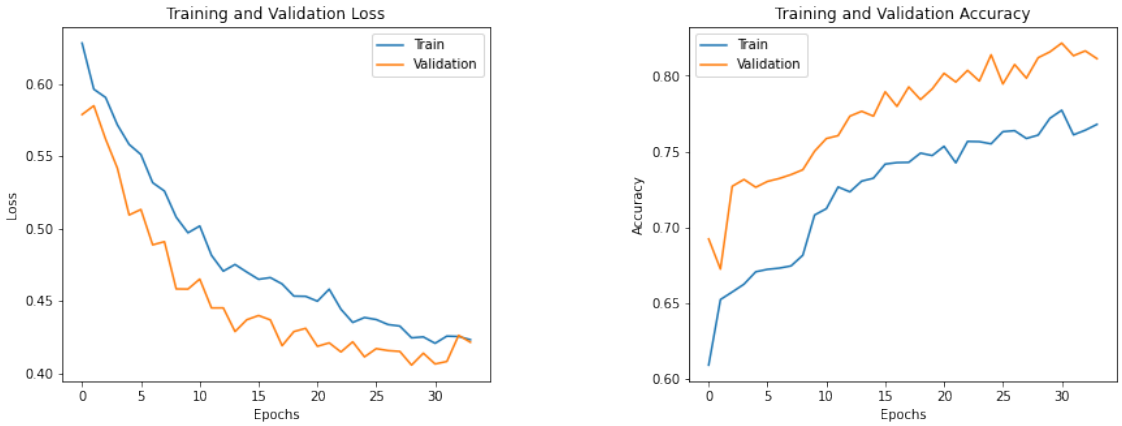
(b) Histogram with bin width of 10

Figure 12: Histograms showing the predicted and actual neutrino energy distributions

Neutrino Flavour Classification

The multi-view model was adapted to classify neutrino flavours using event images. Due to the absence of tau-neutrinos in the dataset and the small number of events that did not correspond to a specific flavour, only muon-neutrinos (0) and electron-neutrinos (1)

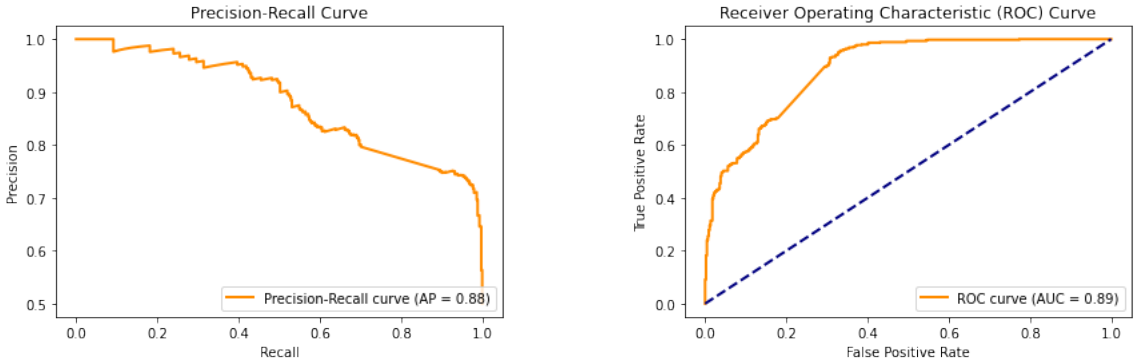
were used as classes. The binary classifier retained the same loss function, activation, and number of output neurons. The model was trained on a balanced dataset of 7,700 muon-neutrinos and electron-neutrinos for 34 epochs. figure 13 displays the training and validation loss and accuracy. On a test set of 1,500 images, the model achieved a test



(a) Loss (b) Accuracy

Figure 13: MVCNN training metrics for binary classification of neutrino flavours

loss and accuracy of 0.3919 and 0.8083, respectively. The model achieved a precision of 0.92 and 0.74 for muon neutrino and electron neutrino, respectively. Recall for muon neutrino and electron neutrino was 0.67 and 0.94, respectively, meaning that the model was able to identify 67% and 94% of actual examples for these classes. The F1-score was higher for electron neutrino (0.83) than for muon neutrino (0.78), indicating better performance on the former. The macro and weighted average scores of 0.83 suggest that the model performed consistently across both classes, with slightly more weight given to the larger class. Figure 14a shows the precision-recall curve, which indicates that the classifier's precision declines sharply when recall is increased beyond approximately 0.97. This suggests that the classifier struggles when it tries to recall all positive instances. The



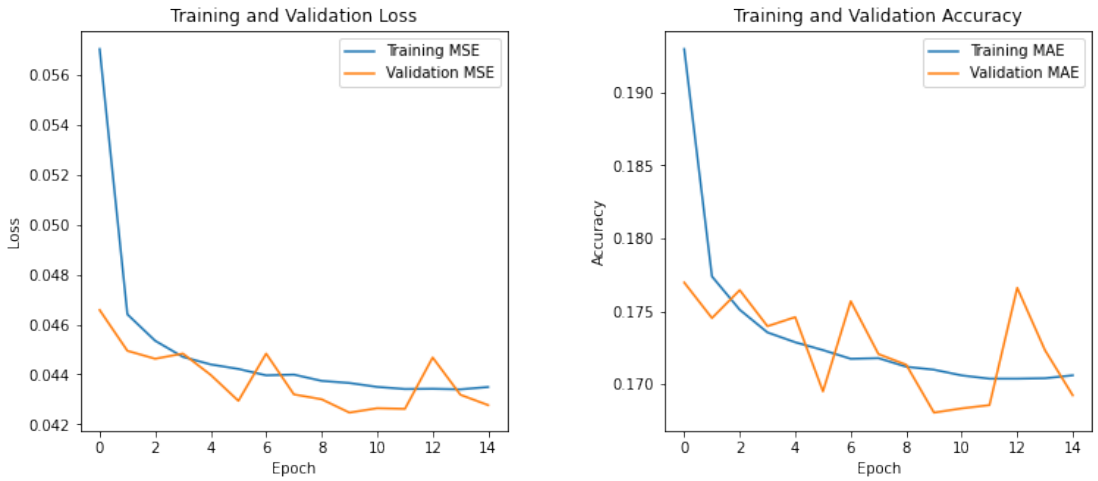
(a) Precision-recall curve (b) ROC curve

Figure 14: Flavour binary classifier performance curves

classifier can correctly identify all positive instances with high recall, but it also identifies a large number of false positives, resulting in low precision. This may be attributed to the significant class imbalance favouring muon neutrinos (97% of the data) and the consequent reduction of data to balance the dataset during training. This imbalance may have resulted in limited positive examples, causing the classifier to overgeneralise from the data available. The ROC curve in figure 14b shows that the classifier identifies false positives at a low cost in terms of missed true positives, with a sharp increase in the curve up to a FPR of 0.05. From a FPR of 0.05 to 0.37, the curve increases at a steady rate, indicating a good balance between TPR and FPR. During this range, the precision increases from 0.5 to 0.95, suggesting that the classifier is becoming more selective in its predictions. The high TPR from a FPR of 0.35 onwards indicates that the classifier can still identify most true positives while being more selective. Overall, the classifier has a good trade-off between sensitivity and specificity, with high TPR and relatively low FPR.

Lepton to Neutrino Energy Ratio Regression

The adapted MVCNN model used for neutrino energy regression was again used for regression of $\frac{E_\ell}{E_\nu}$. The model trained for 14 epochs on a dataset of 138,000 events, and the test set comprised 34,000 events after removing events with zero neutrino energy to avoid invalid values during training. Loss was measured using mean squared error and accuracy using mean absolute error. Training metrics are shown in figure 15. Test loss was 0.042 and accuracy was 0.160.

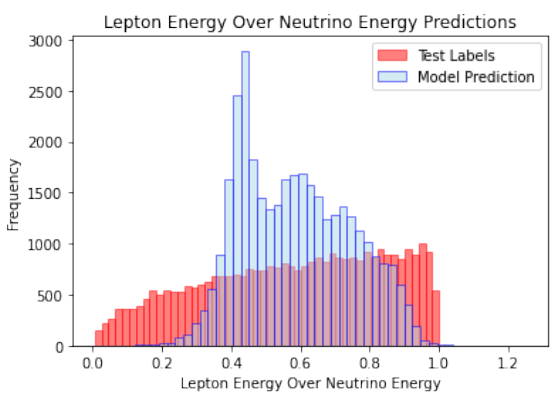


(a) Loss (mean squared error)

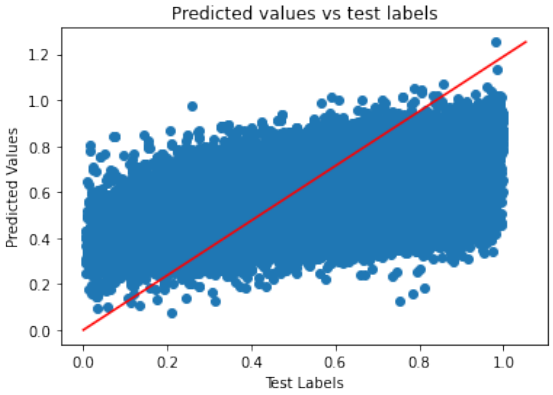
(b) Accuracy (mean absolute error)

Figure 15: MVCNN training metrics for regression of $\frac{\ell_E}{\nu_E}$

The distribution of model predictions and test labels is shown in figure 16a. It is clear that the model tends to over-predict the frequency of energies in certain regions, with peaks at 0.4 and 0.7. There is also a seemingly anomalous peak at 0.6. It also under-predicts in particularly high or low energy ratios (e.g. < 0.4 and > 0.75). This is shown by the scatter plot of predictions against test labels in figure 16b



(a) Histogram of model predictions and corresponding test labels



(b) Scatter plot of predictions against test labels. The red line indicates optimal trend ($y = x$)

Figure 16: Comparison of model predictions and test labels

Interaction/Scattering Type Classification

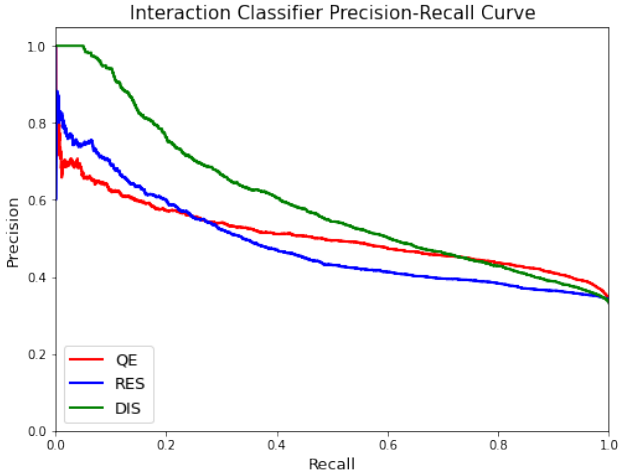


Figure 17: Interaction classifier precision-recall curves

Classification of multiple interaction types (QE, DIS and RES) was attempted via multi-class classification. Since flavour classification was already done, the classes were defined as QE (0), RES (1), DIS (2). The model was an adaptation of the 'CVN model' proposed by A. Aurisano et al [4]. It consists of several convolutional layers with different filter sizes, max pooling layers, and inception modules, designed to learn increasingly complex features from the input. Local response normalisation (LRN) layers were used to reduce overfitting. The softmax activation function was used for multi-class classification. Precision was best on the DIS class (0.55) and was similar for QE (0.46) and RES (0.47). Recall was highest for the QE class (0.64), followed by DIS (0.44) and RES (0.38). F1-score was highest for QE (0.54), followed by DIS (0.49) and RES (0.42). The precision-recall curve is shown in

figure 17. Test accuracy was 0.50.

6 Conclusion

Three models were tested on muon neutrino CC event classification. The multi-view CNN model was determined as most efficient (accuracy: 0.84, F1-score: 0.82). Performance was tested on different datasets e.g. DIS, QE, high and low neutrino energy and high and low muon energy, resulting in satisfactory performance (> 0.80 accuracy). The MVCNN was applied to: regression of neutrino energy (MAE: 6.12), regression of the lepton to neutrino energy ratio (MAE: 0.16), binary classification of neutrino flavour (F1 score: 0.83) and a multi-classification of interaction type (F1 score: 0.48).

The MVCNN may be improved by applying different strategies of fusing the visual features from each branch. For instance, fusing convolutional feature maps at different network depths, fusion of bottleneck latent representations prior to classification and score fusion may be considered, as proposed by M. Seeland et al. [36]. Transfer learning [37] (e.g. a fine-tuned ResNet18 model [4]) may offer accuracy improvements. Improved classification of muon neutrino charged-current events allows more precise measurement of neutrino oscillation occurrences. Improved neutrino energy identification allows for more precise neutrino oscillation probabilities for certain flavours. This has significance in our understanding of beyond Standard Model theories and matter-antimatter asymmetry [38].

Future research may wish to consider the application of the presented MVCNN model on similar neutrino image datasets e.g. for supernova neutrino classification. Adaptations of the model could be used to determine number of protons or pions and specific interaction mode.

References

[1] DUNE collaboration, *Neutrino interaction classification with a convolutional neural network in the DUNE far detector*, *Phys. Rev. D* **102** (2020) .

[2] M. Kubu and P. Bour, *CNN with residual learning extensions in neutrino high energy physics*, *Journal of Physics: Conference Series* **1730** (2021) .

[3] M. Andrews, M. Paulini, S. Gleyzer and B. Poczos, *End-to-end event classification of high-energy physics data*, *Journal of Physics: Conference Series* **1085** (2018) .

[4] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M.D. Messier, E. Niner et al., *A Convolutional Neural Network Neutrino Event Classifier*, *JINST* **11** (2016) P09001 [1604.01444].

[5] F. Reines and C.L. Cowan, *The neutrino*, *Nature* **178** (1956) 446.

[6] E.K. Akhmedov, *Neutrino physics*, in *ICTP Summer School in Particle Physics*, pp. 103–164, 6, 1999.

[7] R.L. Kustom, D.E. Lundquist, T.B. Novey, A. Yokosawa and F. Chilton, *Quasielastic neutrino scattering*, *Phys. Rev. Lett.* **22** (1969) 1014.

[8] R.J. Oakes and W.-K. Tung, *Deep Inelastic Muon Scattering in DUMAND*, in *DUMAND Summer Workshop*, 12, 1978.

[9] T.K. Kuo and J. Pantaleone, *Neutrino oscillations in matter*, *Rev. Mod. Phys.* **61** (1989) 937.

[10] S.M. Bilenky, C. Giunti and W. Grimus, *Phenomenology of neutrino oscillations*, *Prog. Part. Nucl. Phys.* **43** (1999) 1 [hep-ph/9812360].

[11] T2K collaboration, *T2K results on long-baseline oscillations*, *PoS* **NOW2022** (2023) 002.

[12] M.J. Frank, *The NOvA Experiment*, *AIP Conf. Proc.* **1560** (2013) 223.

[13] R. Patterson, *The NOvA experiment: status and outlook*, *Nuclear Physics B - Proceedings Supplements* **235-236** (2013) 151.

[14] NOvA collaboration, *The NOvA Technical Design Report*, Tech. Rep. FERMILAB-DESIGN-2007-01 (10, 2007).

[15] F. Rosenblatt, *Principles of neurodynamics: Perceptrons and the Theory of Brain Mechanisms* (1961).

[16] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Harvard University (1975).

[17] Y. LeCun, Y. Bengio and G. Hinton, *Deep learning*, *Nature* **521** (2015) 436.

[18] G.E. Hinton, *Deep belief networks*, *Scholarpedia* **4** (2009) .

[19] K. Fukushima, *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*, *Biological cybernetics* **36** (1980) 193.

[20] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard et al., *Backpropagation applied to handwritten zip code recognition*, *Neural computation* **1** (1989) 541.

[21] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard et al., *Handwritten digit recognition with a back-propagation network*, *Advances in neural information processing systems* **2** (1989) 396–404.

[22] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, *Gradient-based learning applied to document recognition*, *Proceedings of the IEEE* **86** (1998) 2278.

[23] A. Krizhevsky, I. Sutskever and G.E. Hinton, *Imagenet classification with deep convolutional neural networks*, *Communications of the ACM* **60** (2017) 84.

[24] V. Nair and G.E. Hinton, *Rectified linear units improve restricted boltzmann machines*, in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *Dropout: a simple way to prevent neural networks from overfitting*, *The Journal of Machine Learning Research* **15** (2014) 1929.

[26] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, in *International Conference on Machine Learning*, pp. 448–456, 2015.

[27] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*, MIT press (2016).

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov et al., *Going deeper with convolutions*, pp. 1–9, 06, 2015, DOI.

[29] C. Szegedy, S. Ioffe and V. Vanhoucke, *Inception-v4, inception-resnet and the impact of residual connections on learning*, *CoRR* (2016) .

[30] H. Su, S. Maji, E. Kalogerakis and E. Learned-Miller, *Multi-view Convolutional Neural Networks for 3D Shape Recognition*, *CoRR* (2015) .

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, *Rethinking the Inception Architecture for Computer Vision*, vol. 7, pp. 2818–2826, 2015, DOI.

[32] Z.H. Hoo, J. Candlish and D. Teare, *What is an ROC curve?*, *Emergency Medicine Journal* **34** (2017) 357.

[33] A.P. Bradley, *The use of the area under the roc curve in the evaluation of machine learning algorithms*, *Pattern Recognition* **30** (1997) 1145.

[34] C. Marzban, *The roc curve and the area under it as performance measures*, *Weather and Forecasting* **19** (2004) 1106 .

[35] P. Flach and M. Kull, *Precision-Recall-Gain Curves: PR Analysis Done Right*, in *Advances in Neural Information Processing Systems*, vol. 28, Curran Associates, Inc., 2015.

[36] M. Seeland and P. Mäder, *Multi-view classification with convolutional neural networks*, *PLOS ONE* **16** (2021) 1.

[37] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu et al., *A Comprehensive Survey on Transfer Learning*, *CoRR* **109** (2019) 43.

[38] T2K collaboration, *Constraint on the matter–antimatter symmetry-violating phase in neutrino oscillations*, *Nature* **580** (2020) 339.