TECHNICAL REPORT

Vertex finding in neutrino-nucleus interaction: a model architecture comparison

Published by IOP Publishing for Sissa Medialab

Received: *January 10, 2022*
Accepted: *August 3, 2022*
Published: *August 31, 2022*

TECHNICAL REPORT

Vertex finding in neutrino-nucleus interaction: a model architecture comparison

The MINERνA Collaboration

F. Akbar,[a,*] **A. Ghosh,**[b,c] **S. Young,**[d] **S. Akhter,**[a] **Z. Ahmad Dar,**[e,a] **V. Ansari,**[a] **M.V. Ascencio,**[f] **M. Sajjad Athar,**[a] **A. Bodek,**[g] **J.L. Bonilla,**[h] **A. Bravar,**[i] **H. Budd,**[g] **G. Caceres,**[c] **T. Cai,**[g] **M.F. Carneiro,**[j,c] **G.A. Díaz,**[g] **J. Felix,**[h] **L. Fields,**[k] **A. Filkins,**[e] **R. Fine,**[g] **P.K. Gaur,**[a] **R. Gran,**[l] **D.A. Harris,**[m,n] **D. Jena,**[n] **S. Jena,**[o] **J. Kleykamp,**[g] **A. Klustová,**[p] **D. Last,**[q] **A. Lozano,**[c] **X.-G. Lu,**[r,s] **E. Maher,**[t] **S. Manly,**[g] **W.A. Mann,**[u] **K.S. McFarland,**[g] **B. Messerly,**[v] **J. Miller,**[b] **O. Moreno,**[e,h] **J.G. Morfín,**[n] **J.K. Nelson,**[e] **C. Nguyen,**[w] **A. Olivier,**[g] **V. Paolone,**[v] **G.N. Perdue,**[n,g] **K.-J. Plows,**[s] **M.A. Ramírez,**[q,h] **D. Ruterbories,**[g] **H. Su,**[v] **V.S. Syrotenko,**[u] **A.V. Waldron,**[p] **B. Yaeggy**[b] **and L. Zazueta**[e]

[a] *AMU Campus, Aligarh, Uttar Pradesh 202001, India*
[b] *Departamento de Física, Universidad Técnica Federico Santa María, Avenida España 1680 Casilla 110-V, Valparaíso, Chile*
[c] *Centro Brasileiro de Pesquisas Físicas, Rua Dr. Xavier Sigaud 150, Urca, Rio de Janeiro, Rio de Janeiro, 22290-180, Brazil*
[d] *Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, U.S.A.*
[e] *Department of Physics, William & Mary, Williamsburg, VA 23187, U.S.A.*
[f] *Sección Física, Departamento de Ciencias, Pontificia Universidad Católica del Perú, Apartado 1761, Lima, Perú*
[g] *Department of Physics and Astronomy, University of Rochester, Rochester, NY 14627, U.S.A.*
[h] *Campus León y Campus Guanajuato, Universidad de Guanajuato, Lascurain de Retana No. 5, Colonia Centro, Guanajuato 36000, Guanajuato, México*
[i] *University of Geneva, 1211 Geneva 4, Switzerland*
[j] *Department of Physics, Oregon State University, Corvallis, OR 97331, U.S.A.*
[k] *Department of Physics, University of Notre Dame, Notre Dame, IN 46556, U.S.A.*
[l] *Department of Physics, University of Minnesota — Duluth, Duluth, MN 55812, U.S.A.*
[m] *York University, Department of Physics and Astronomy, Toronto, Ontario, M3J 1P3, Canada*
[n] *Fermi National Accelerator Laboratory, Batavia, Illinois 60510, U.S.A.*
[o] *Department of Physical Sciences, IISER Mohali, Knowledge City, SAS Nagar, Mohali — 140306, Punjab, India*

[*] Corresponding author.

https://doi.org/10.1088/1748-0221/17/08/T08013

2022 JINST 17 T08013

[p] The Blackett Laboratory, Imperial College London, London SW7 2BW, U.K.
[q] Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA 19104, U.S.A.
[r] Coventry CV4 7AL, U.K.
[s] Oxford University, Department of Physics, Oxford, OX1 3PJ, U.K.
[t] Massachusetts College of Liberal Arts, 375 Church Street, North Adams, MA 01247, U.S.A.
[u] Physics Department, Tufts University, Medford, MA 02155, U.S.A.
[v] Department of Physics and Astronomy, University of Pittsburgh, Pittsburgh, PA 15260, U.S.A.
[w] University of Florida, Department of Physics, Gainesville, FL 32611, U.S.A.

E-mail: fakbar@ur.rochester.edu

Abstract: We compare different neural network architectures for machine learning algorithms designed to identify the neutrino interaction vertex position in the MINERvA detector. The architectures developed and optimized by hand are compared with the architectures developed in an automated way using the package "Multi-node Evolutionary Neural Networks for Deep Learning" (MENNDL), developed at Oak Ridge National Laboratory. While the domain-expert hand-tuned network was the best performer, the differences were negligible and the auto-generated networks performed as well. There is always a trade-off between human, and computer resources for network optimization and this work suggests that automated optimization, assuming resources are available, provides a compelling way to save significant expert time.

Keywords: Analysis and statistical methods; Data processing methods; Simulation methods and programs; Software architectures (event data models, frameworks and databases)

ArXiv ePrint: 2201.02523

2022 JINST 17 T08013

Contents

1 Introduction 1

2 MINERvA dectector 2

3 Analysis challenges in the MINERvA Experiment 4

4 Network topology of the domain-expert designed ML model 4

5 Neural architecture search (NAS) 5

6 MENNDL 5

7 Details of simulation 7

8 Results 7

9 Conclusion 13

2022 JINST 17 T08013

1 Introduction

Particle physics experiments are increasing their use of Machine Learning (ML) algorithms at reconstruction level to maximize their physics output. These algorithms currently tend to use one specific architecture, which means coming up with values for the number of layers of each type and the number of nodes in each of these layers. The architecture is chosen "by hand", and then the algorithms are trained on both labeled simulated and unlabeled real particle physics data in the cases of supervised learning and unsupervised learning, respectively. The process of choosing a specific architecture is lengthy and often takes months of human effort. Though deep learning has been able to bypass the process of manual feature engineering by learning representations in conjunction with statistical models in an end-to-end fashion, neural network architectures themselves are typically designed by experts in a painstaking, ad hoc manner. Neural architecture search (NAS) has been boosted as the path forward for alleviating this pain by automatically identifying architectures that are as good as hand-designed ones.

A group from Oak Ridge National Laboratory (ORNL) developed a package named Multi-node Evolutionary Neural Networks for Deep Learning (MENNDL) [1, 2] which addresses the model selection problem and eases the demands on data researchers. MENNDL leverages a large number of compute nodes, which communicate over Message Passing Interface (MPI) to distribute the task of finding the optimal architecture across the nodes of a supercomputer. In our previous paper [3], we applied ML for improving neutrino interaction point (or "vertex") reconstruction. We developed the Convolutional Neural network (CNN) based ML model inside the collaboration

for the neutrino events using both TensorFlow and Caffe APIs. We showed that the ML-based reconstruction improved the neutrino vertex reconstruction significantly compared to traditional cut-based method. Moreover, we explored the Domain Adversarial Neural Network (DANN) based ML model developed using Caffe API, where we used both the labeled simulated events and the unlabelled real data to train the model. Following this procedure, we were able to remove the possible biases coming from large simulated training sets.

The successful application of ML brings the following questions; is the developed algorithm the optimal solution? Can the same architecture be used for similar kinds of datasets but generated from antineutrino events? Do we need a systematic to take into account the possible bias coming from the choice of the architecture? In order to investigate these questions, we performed a thorough comparison between an algorithm that was tuned by hand over several months of researcher's time given an intimate knowledge of the detector's performance and an algorithm that was tuned using MENNDL over a few hours with a supercomputer at Oak Ridge. Exploring many topologies for a specific physics problem may help us to understand the systematic related questions mentioned above. The performances were compared using simulated antineutrino interactions in the MINERvA detector. These interactions provide a useful arena for a model comparison because there is a detailed hit-level simulation of the detector and an important output that is needed for future physics analyses: the spatial location of the neutrino interaction vertex in a complex detector geometry. The comparison involves testing several different models using the MENNDL package for vertex finding, where those models were optimized over different architectures. Although automated neural architecture searches are widely used in industry, to the best of our knowledge this is the first time one is being used to analyze neutrino/antineutrino physics data.

The paper is organized as follows: in section 2, we describe the MINERvA detector. Next in section 3, we discuss the challenges in reconstructing the vertex of the neutrino interaction. In section 4, we summarize the network topology of the domain-expert-designed model developed by the MINERvA collaboration. In section 5, we discuss about NAS, followed by section 6 where details of MENNDL are described. In section 7 we discuss the details of the simulation, and in section 8 we discuss the results. Section 9 provides the conclusion.

2 MINERvA dectector

The MINERvA detector [4] consists of a nuclear target region where most of the interactions considered in this paper take place, followed by a fine-grained tracker which measures the products from the upstream interactions and also serves as an active target. There is electromagnetic and hadronic calorimetery surrounding the nuclear targets and the tracking region, and a muon spectrometer (the MINOS near detector [5]) is located downstream of the hadronic calorimeter region. A right-handed Cartesian coordinate system is used in the MINERvA experiment. In this coordinate system, the origin is located at the center of inner detector. Z-axis is along the beam line parallel to the surface of the earth. X-Y plane being orthogonal to the beam line with Y-axis pointing vertically upwards and X-axis, perpendicular to both of the other axes, horizontal pointing to beam left. In this system the beam central axis is in the Y-Z plane and points slightly downward at 3.34°. The Z-axis is chosen in a way that $Z = 1200$ cm at the front face of the MINOS near detector.

2022 JINST 17 T08013

The nuclear target region of the detector is complex and consists of both active and passive elements. The region is designed to have the same target material (iron or lead) located in several different regions relative to the center of the neutrino beam and relative to the muon spectrometer which is itself highly non-symmetric with respect to the neutrino beam axis. The nuclear target design allows cross-checks of the cross section measurements between regions that have different geometric acceptances. The longitudinal segmentation of the nuclear target region is shown in figure 1(a), where the neutrino beam is incident from the left of the figure. The transverse segmentation of the different passive nuclear targets is shown in figure 1(b).

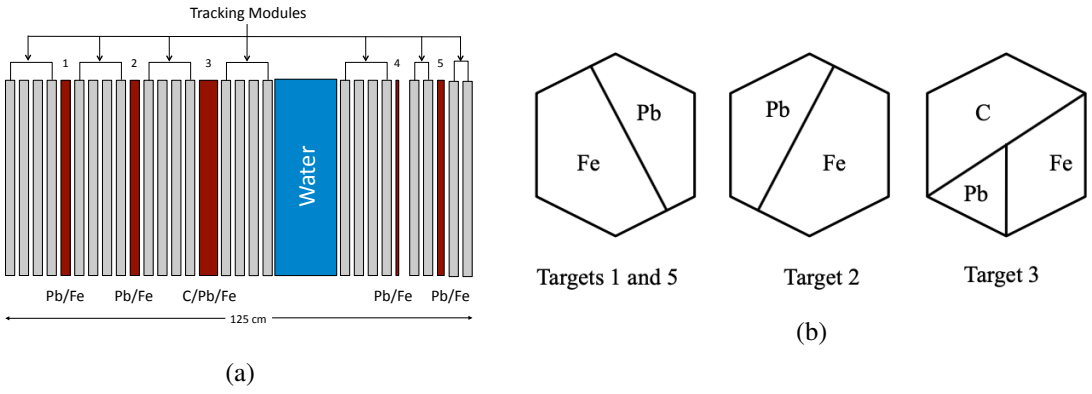


Figure 1. (a) Longitudinal segmentation. (b) Transverse segmentation: targets 1, 2, 3, and 5 are shown, and target 4 consists entirely of lead. Target 3 is indicated by the figure at the right, and targets labeled 1, 2, and 5 on the left alternate between the configurations in the left and middle of the diagram at the right (figures taken from [4]).

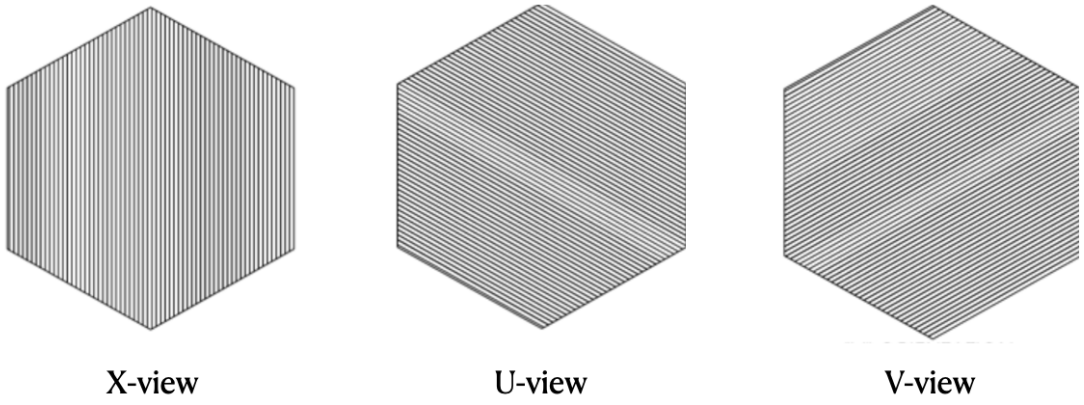


Figure 2. Three orientation/views of the scintillator planes.

The active scintillator planes that provide the input to the ML algorithm consist of nested bars with a triangular cross section whose base is 3.3 cm and height is 1.7 cm. The transverse cross section of the detector is hexagonal in shape, and the planes have three different orientations to allow for stereoscopic track reconstruction: the scintillator bars are either vertical or they are oriented $\pm 60^\circ$ with respect to vertical. Two successive scintillator planes have strips oriented vertically followed by one of either positive or negative 60° (figure 2). Each grey rectangle in figure 1(a) corresponds to two scintillator planes where the sequential non-vertical bar directions alternate between $+60^\circ$ and -60°.

2022 JINST 17 T08013

3 Analysis challenges in the MINERvA Experiment

The MINERvA experiment has a broad physics program designed to measure a wide range of neutrino interaction channels. One of the primary themes in MINERvA's physics program is to measure how different neutrino interaction channels are modified by the nuclear environment where an interaction happens. In order to do that, we use a detector that has several different thin (passive) targets made of the nuclei we want to study (for example carbon, iron, lead), and then we need to determine the target in which neutrino interaction took place. In order to determine the interaction location, segmented scintillator planes are located both upstream and downstream of the thin targets. The signals from the scintillator planes are then used to predict the start of the interaction. Neutrinos are neutral particles and leave no trace in the scintillator. Nevertheless, charged particles produced by the neutrino interactions do leave traces, emerging from the interaction vertex in all directions. In general the higher the neutrino energy, the more energy can be transferred to the nucleus, and the more energy that is transferred, the more final state charged particles can be produced. This sounds like a straightforward problem that might not require a ML algorithm, but for high-momentum transfers, there can be several charged particles leaving the interaction point in many directions.

Vertex finding is made challenging by vertex occlusion in passive material, particle re-interactions, and hadronic shower cascades: these factors combine to significantly degrade vertex finding accuracy, requiring very strict sample selection cuts and background subtraction estimation procedures to accurately predict and subtract the large background. Supervised CNN could improve on the state of the art of reconstruction algorithms. In a previous publication [3] we applied CNN based supervised ML model to find the vertex. We considered specific events with high momentum transfers or in other words, events originating from deep inelastic scattering (DIS) events. Using the ML approach significantly improves the efficiency and accuracy of our vertex-finding algorithm. The CNN-based method was the top performer, including in comparisons with a tracking algorithm based on fitting clusters of hits with a Kalman filter [6]. In addition, we presented means to mitigate the possible model biases coming from the large labeled training sample using the Domain Adversarial Neural Network (DANN) [7] algorithm.

4 Network topology of the domain-expert designed ML model

Two CNN based supervised models using Caffe [8] and Tensorflow [9] application programming interfaces (APIs) were developed. The structure of the models is the same in both APIs. A description of the domain-expert-designed model (as it will be named throughout the paper) is given in [3]. The network is made of three separate towers for X, U and V views where each tower comprises four iterations of convolution and max pooling [10, 11] layers with ReLUs (Rectified Linear Unit) [12] acting as the non-linear activations. Each pooling layer consists of a kernel which decreases the dimension along the transverse axis by one. After the four iterations of convolution, ReLU and pooling, there is a fully connected layer with 196 semantic output features. The outputs for the three views are concatenated and fed to another fully connected layer with 98 outputs which in turn is input for a final fully connected layer with 67 outputs for the plane classifier, which is the input to a Softmax [13] layer. Fully connected layer allows non-linear combinations of the discovered features and operates at the end of the feature discovery layers to associate the discovered features to

2022 JINST 17 T08013

desired outputs. Finally, we assign the network cost using a cross-entropy function [14], which is subsequently minimized. Figure 3 shows the structure of the domain-expert-designed-Tensorflow model diagrammatically. Here, the plane classifier identifies the plane where the true vertex is located. To classify the data with the plane classifier, we use a total of 67 "planecodes". This includes two "overflow" classes — code 0 for events upstream of the detector, and code 66 for everything downstream of the last considered plane.

5 Neural architecture search (NAS)

NAS [15–18] describes a family of automated architecture optimization algorithms (i.e. optimizing the number and type of layers, hyperparameters of those layers, connectivity of layers, etc.). NAS is distinct from hyperparameter optimization which is generally restricted to optimizer algorithm choice and minor changes to layer hyperparameters of a fixed network architecture. In other words, NAS forms a scaffolding for a neural network architecture and hyperparameter search performs simple scalar adjustments or algorithm choices given an architecture. The idea of automatically learning and evolving network topologies was first explored in [19]. More recently, the pioneering works by [20] and [21] have led to a number of better, faster and cost-efficient NAS methods, thus attracting a lot of attention to this field.

There any many approaches to NAS, but they must all define a search space, search strategy, and a model evaluation strategy. The search space defines the components of the network and their possible combinations and connections. The search strategy determines the method of optimization in order to explore the search space, which significantly affects the efficiency of the search, as well as the effectiveness of the final proposed architecture. The choice of the method of optimization ensures the sufficient investigation of the search space and keeps the resulting architecture close to the optimum. Finally, the evaluation strategy compares the intermediate results, thereby helping the search strategy to choose the best option during the search process. Several techniques can be employed for the search strategies, e.g., Bayesian optimization [22], evolutionary algorithms [23], gradient descent [24], reinforcement learning [20], and meta-learning [25].

6 MENNDL

Multi-node Evolutionary Neural Networks for Deep Learning (MENNDL) [1, 2] is a NAS algorithm that utilizes evolutionary optimization for evolving the architecture and hyperparameters of convolutional neural networks on high performance computers.

MENNDL uses a master-worker paradigm such that a master process is used to perform the core evolutionary process and workers are used to evaluate the fitness of the generated networks. The evolutionary process is asynchronous. Once a sufficient number of individual networks has been evaluated, a new set of individuals are produced through selection, crossover, and mutation and then queued to be evaluated. This ensures the computational resources used by the workers are not left idle. MENNDL has previously shown success in automatically designing neural networks for several scientific datasets [1, 2]. It is particularly useful for datasets that have very different image characteristics than the photographic imagery typically used in machine learning literature (e.g. ImageNet [26]). A key difference between MENNDL and other NAS methods is that it focuses

2022 JINST 17 T08013

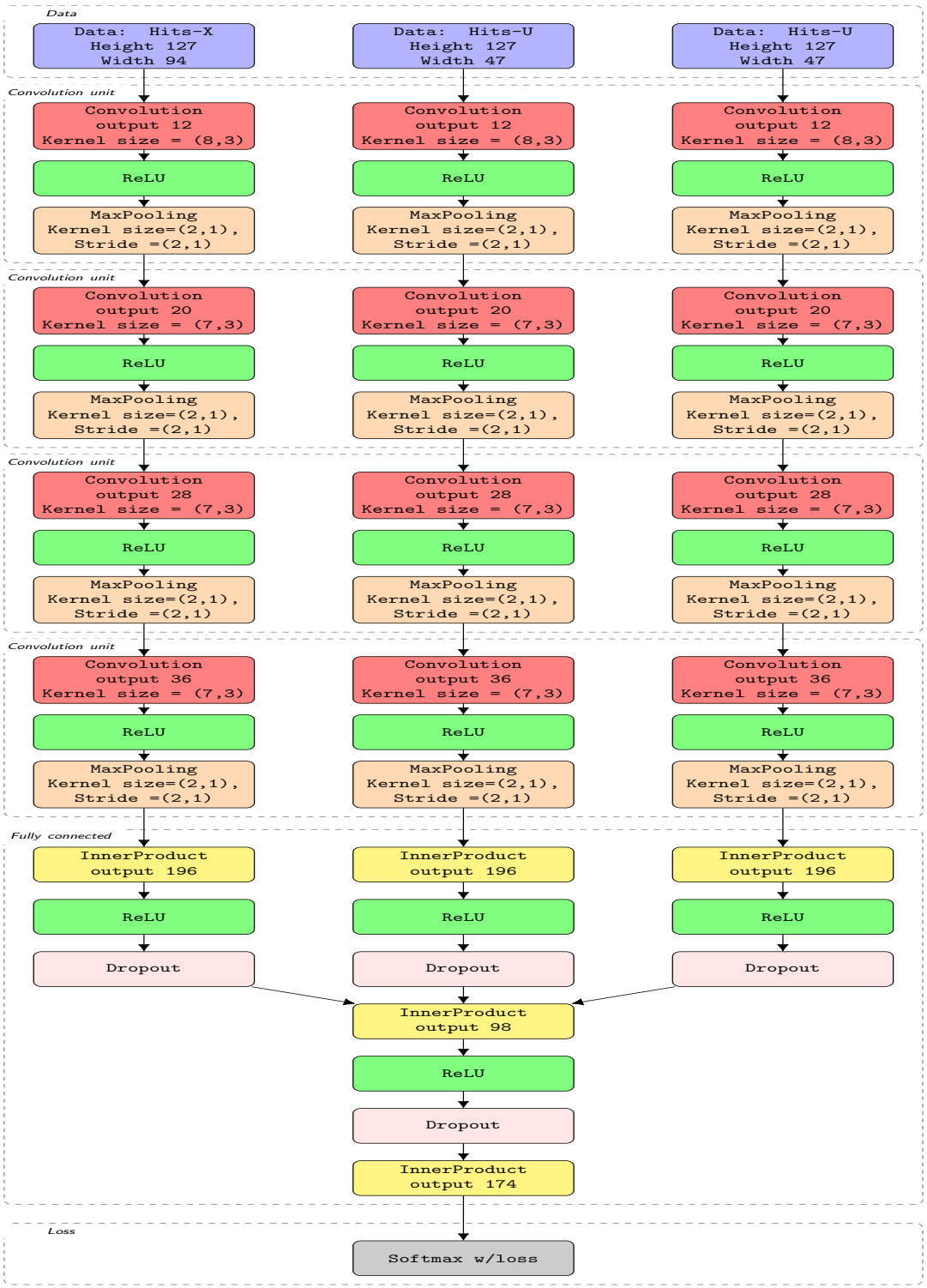


Figure 3. The flow chart of the network structure of the domain-expert-designed-Caffe model.

on exploring all possible hyperparameters for the potential layers along with the composition of those layers, as opposed to most other methods which limit the range of hyperparameters to a small set (e.g. DARTS only allows for a handful of potential kernel sizes [27]).

In this work, some modifications were made in order to handle the unique data used. In order to handle the three views, the same three tower structure was used for the MENNDL networks, with each tower having identical layer types and hyperparameters learned through the evolutionary process. The maximum size of hyperparameters related to the width of the input image was set to the maximum allowed by the two smaller views. MENNDL generated and evaluated approximately 10,000 networks (models in the following) on the Titan Supercomputer using 500 nodes for 12 hours. Example networks are shown in figure 4. From these networks, a subset was chosen for further training and evaluation based on their validation accuracy. The events used for training and developing the MENNDL networks are the simulated events generated using a neutrino flux peaked around 6 GeV.

It is noted that these models are primarily optimized over short run periods (only 5000 iterations) using the neural network software library Caffe. The version of MENNDL used in this work is defined in the 2017 paper [2], and thus the hyperparameters optimized by MENNDL in this work are limited to those defining the layers and the architecture of the neural network. We can optimize the number of layers and the type of each layer along with layer specific hyperparameters (e.g. kernel size, number of filters, dropout rate, etc.). Solver hyperparameters such as learning rate are not optimized by MENNDL.

7 Details of simulation

We choose 9 models out of the 10,000 generated ones, based on their validation accuracy. Validation loss and accuracy are the loss and accuracy on a validation set, which gives a measure of the quality of the model. Next, we consider a data sample of size 1.66 million events, and divide that sample in two: $\sim 93\%$ of the events are used for training whereas the remaining $\sim 7\%$ are used for validation of the models. These events are the antineutrino events coming from the NuMI (Neutrinos at the Main Injector) beam with a peak energy of around 6 GeV. We train the MENNDL models and the domain-expert-designed-Caffe model using the simulated antineutrino events up to 20 iterations (epochs). Note that, since we use a CNN-based model, we make images of the input events. We present the data as two channel images using deposited energy and hit time to train the network. Each event contains three images — one for each view (X, U, V) — that are fed into the separate convolutional towers. We pre-process the images by normalizing the values of the total deposited energy to unity on an event-by-event basis. Timing values are pre-processed by subtracting the event time extracted by fitting a linear model to the anchor track and scaled by the largest absolute value of all the relative times in the event. A detailed description of image processing is given in [3]. To study the impact of the models on a physics analysis, we use another simulated data sample containing two million events which is 10% of the total antineutrino statistics.

8 Results

The validation accuracy and loss curves of the 9 chosen MENNDL models are shown in figure 5. Table 1 shows the number of trainable parameters of each of the MENNDL models along with their corresponding time of execution. Among these 9 MENNDL models, we selected the model

2022 JINST 17 T08013

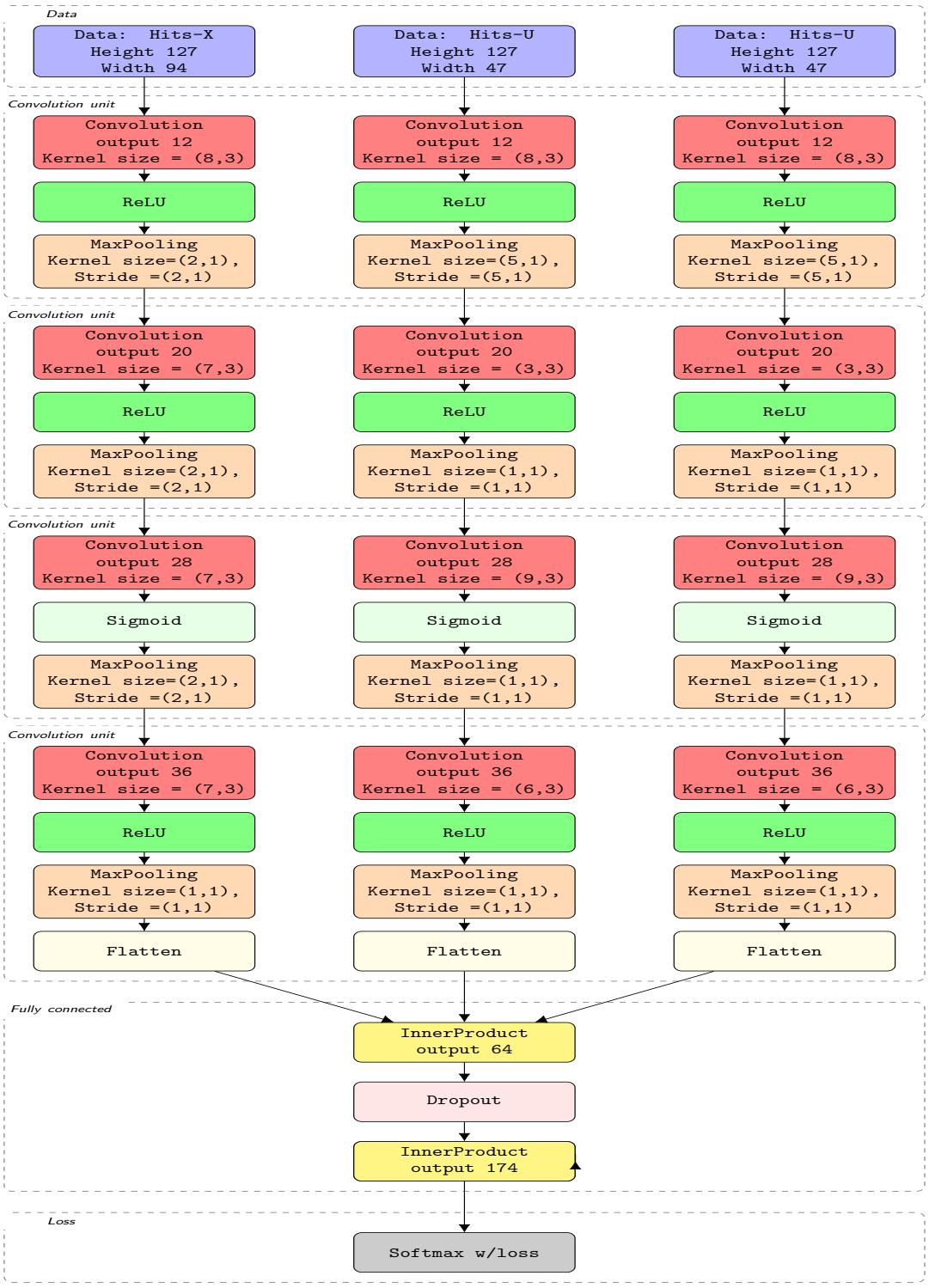


Figure 4. The flow chart of the network structure of the MENNDL model (MENNDL-5 whose performance is indicated by yellow line in figure 5). This model performed best among all the MENNDL models. The performance of this model is compared with the domain-expert-designed-Caffe model.

.

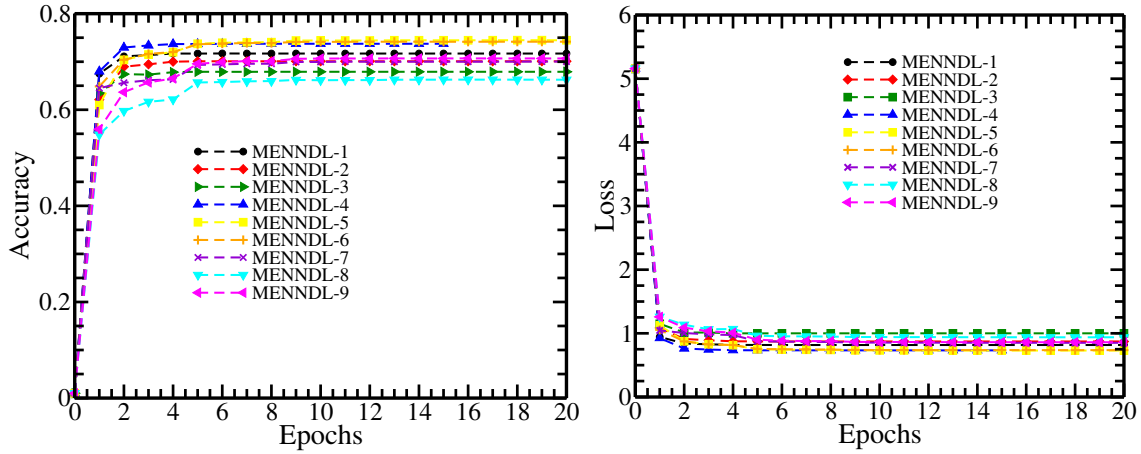
2022 JINST 17 T08013

MENNDL Models	Total number of adjustable parameters	Time of execution (gpu-hour)	Validation loss after 20 epochs	Validation accuracy after 20 epochs (%)
MENNDL-1	415104	8*15.5	0.82	71.7
MENNDL-2	949248	8*23	0.87	70.1
MENNDL-3	2857568	8*16.75	1.00	67.9
MENNDL-4	2857920	8*20.5	0.85	68.5
MENNDL-5	1637712	8*20.75	0.73	74.4
MENNDL-6	1637712	8*20.75	0.74	74.2
MENNDL-7	2014224	8*20.5	0.85	70.0
MENNDL-8	3549240	8*20.5	0.94	66.2
MENNDL-9	2691616	8*20.5	0.86	70.6

Table 1. Total number of adjustable parameters and time of execution, validation loss and validation accuracy for the nine MENNDL models are presented here. Validation loss, and accuracy for those nine MENNDL models are also shown in figure 5. Time of execution has been reported in terms of gpu-hour. We used 8 gpus (Tesla P100) and we used the same setup for all the models.

Models	Validation loss after 20 epochs	Validation accuracy after 20 epochs (%)	Total number of parameters
DED-Caffe	0.78	72.6	12115716
MENNDL-5	0.73	74.5	1637712

Table 2. The comparison between domain-expert-designed-Caffe and MENNDL models in terms of validation loss, validation accuracy, and the total number of adjustable parameters.



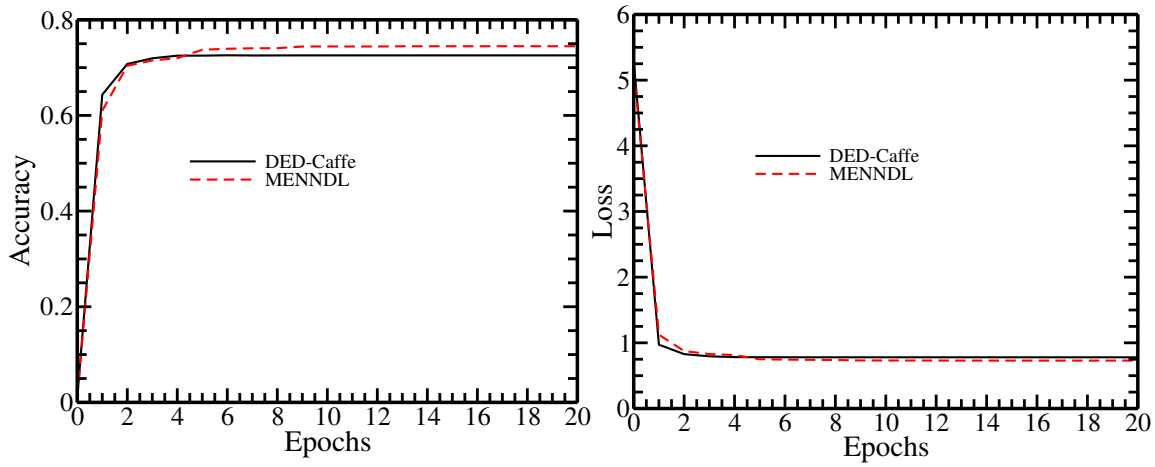
(a) Validation accuracy

(b) Validation loss

Figure 5. Validation accuracy and validation loss for different MENNDL models. Among these nine models, model *MENNDL-5* (yellow coloured square-dashed line) has the highest accuracy (74.5%) and lowest loss (0.73) values.

.

2022 JINST 17 T08013



(a) Validation accuracy

(b) Validation loss

Figure 6. Validation accuracy and validation loss plots for domain-expert-designed(DED)-Caffe model (solid line) and MENNDL-5 model (dashed line).

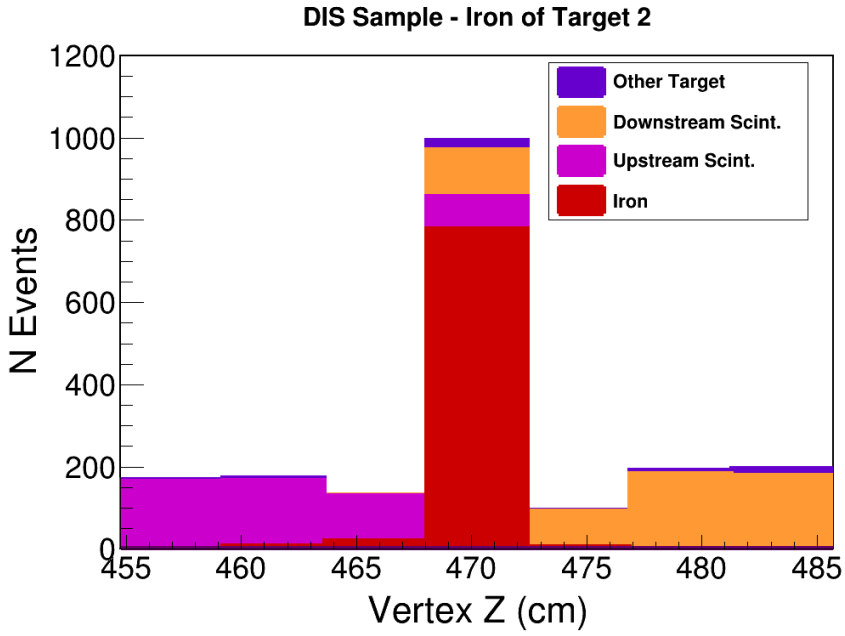
(MENNDL-5) with the highest validation accuracy for the further comparison with the domain-expert-designed-Caffe model.

Figure 6 shows the comparison of the validation accuracy and validation loss between domain-expert-designed-Caffe, and MENNDL-5 (indicated by the yellow line in figure 5) models. The red dashed line corresponds to the MENNDL-5 model (henceforth named as MENNDL model) whereas the black solid line corresponds to the domain-expert-designed-Caffe model. Figures 6(a) and 6(b) show the loss and accuracy as the number of epochs increases. The MENNDL model took $\sim$ 20 hours to train whereas the time taken by the domain-expert-designed-Caffe model for the training is more than 24 hours. MENNDL model was automated while the domain-expert-designed-Caffe model required human intervention. We can see from the figure 6 that both models start to saturate at the same number of epochs, and after 10 epochs of training they show similar accuracy and loss. Table 2 lists the validation loss and accuracy achieved by these two models after reaching the saturation with the total number of adjustable parameters in each model. We also compared the number of parameters of each model, which is shown in the third column of table 2. We see that model found by MENNDL have almost 10 times fewer parameters than the model designed by hand, thus having less expressivity than the more complex domain-expert-designed-Caffe model. This is expected due to the process used by MENNDL, where the fitness of the model is evaluated after a limited number of iterations. However, domain-expert-designed-Caffe model, and MENNDL model demonstrate similar performance, validating the approach used by MENNDL [2].

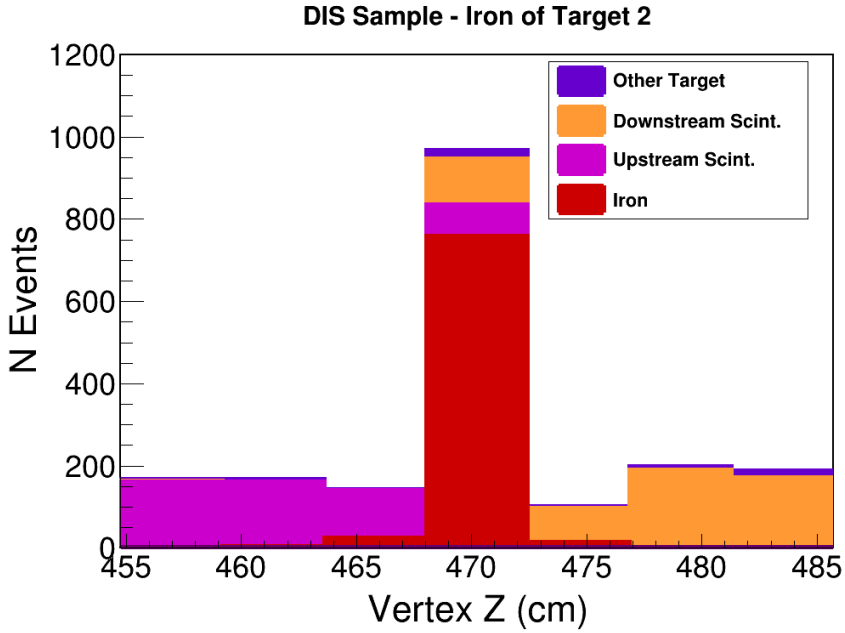
Next, we test our models where we generate the ML-based predictions of the neutrino interaction vertex of each event using these two models. The predictions generated from both the models are incorporated in the MINERvA analysis framework.

Figure 7 shows the event distribution at target 2 of the MINERvA detector with respect to vertex Z using antineutrino simulated data set in deep inelastic scattering (DIS) region. The variable vertex Z is closely related to the actual position of the nuclear targets in the MINERvA detector. We use the ML-based prediction for the reconstructed vertex position, where figures 7(a), and 7(b) represent

2022 JINST 17 T08013



(a) Domain-expert-designed(DED)-Caffe



(b) MENNDL

Figure 7. The event distributions for two different machine learning models using target 2 of the MINERvA detector. Three nuclear targets (iron, lead and carbon) from the MINERvA detector were used in this study. Events reconstructed from the source nuclear target (iron) are represented in red and other targets (lead and carbon combined) are presented in purple. The number of events coming from the plastic scintillator upstream (in this case the six planes between targets 1 and 2 which are closest to target 2) and downstream (in this case the six planes between targets 2 and 3 which are closest to target 2) of the nuclear target are presented in orange and magenta, respectively.

2022 JINST 17 T08013

the domain-expert-designed-Caffe and MENNDL models, respectively. As target 2 is composed of iron and lead, the plots are made for the number of events in iron and lead separately. Here we show the event distribution generated only in iron. In addition to the event distributions coming from the nuclear target (iron), the number of events coming from the plastic scintillator upstream (in this case the six planes between targets 1 and 2, which are closest to target 2) and downstream (in this case the six planes between targets 2 and 3, which are closest to target 2) of the nuclear target are also presented. The two event distributions are quite similar to each other, which signifies that both models provided similar predictions for the interaction vertex. In figure 8, there is the ratio of the two distributions in figure 7b and 7a, ratio which is always compatible with 1 given the error bars.

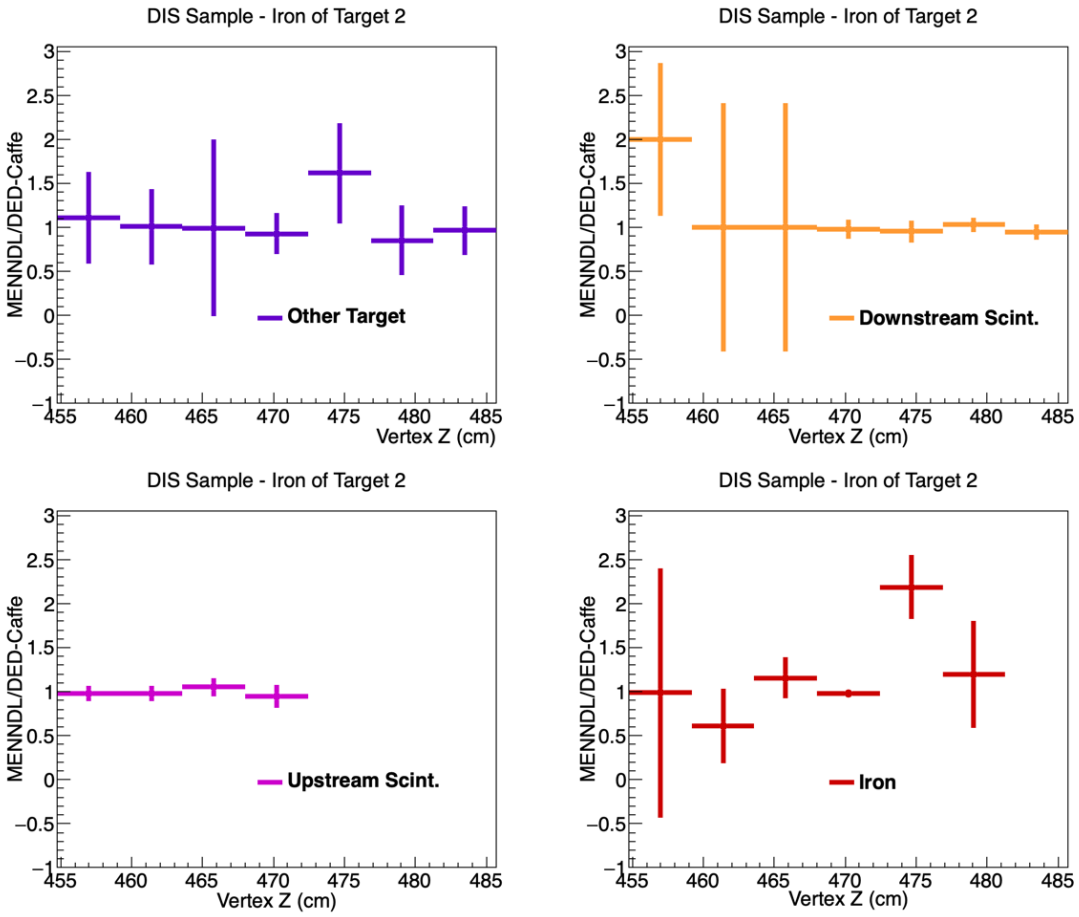


Figure 8. Ratio of the different components from sub-figures 7(a) and 7(b).

We calculate the efficiency and the purity of the sample, where efficiency is the fraction of true events reconstructed, and purity is the fraction of events where the reconstructed value was equal to the underlying true generated value. For the calculation of the efficiency, the numerator is the number of events in the source nuclear target passing the reconstructed sample selection cuts along with the DIS cut (momentum transferred, $Q^2 > 1\,\mathrm{GeV}^2$, and invariant mass of the hadronic state, $W > 2\,\mathrm{GeV}$). The denominator is all true generated charged-current DIS events in the true fiducial volume with true $\theta_\mu < 17°$. Purity is related to the efficiency. Purity is defined with the same

2022 JINST 17 T08013

numerator as for the efficiency which is divided by the number of events passing the reconstructed sample selection cuts and reconstructed DIS cut.

Figure 9 shows the efficiency and purity versus true target number. We compared the domain-expert-designed-Caffe and MENNDL models, and the difference between them varies between 0.8% to 3.4% and from 1.0% to 2.1% for efficiency and purity, respectively.

9 Conclusion

In this paper, we explored the performance of a ML model obtained from the MENNDL package developed by the ORNL group using the MINERvA simulated neutrino data. We further compared the MENNDL model with a successful model which was manually developed by the MINERvA collaboration for improving the vertex reconstruction precision for neutrino interactions. In this work, we trained those models using antineutrino DIS events, generated predictions for the vertex location of the antineutrino DIS events, and compared the performance of the two models in terms of vertex position accuracy at the nuclear target region, efficiency, and purity. In terms of the efficiency with respect to the true target location of DIS events, the differences between the two models ranged between 0.8% to 3.4%, and for the purity those differences varied between 1.0% to 2.1%. Therefore, the MENNDL generated model performs similarly to the domain-expert-designed model. We also see that the model found by MENNDL have almost 10 times fewer parameters than the domain-expert-designed model. Hence, MENNDL model have lower capacity than the comparatively more complex domain-expert-designed model. However, both models can achieve similar performance. From this, we observe that MENNDL favors lower capacity models over more complex ones. The domain-expert-designed model was carefully developed, but it was done manually, and therefore it required a significant amount of time of persons with intimate knowledge of the detector's performance. On the other hand, the MENNDL models were generated and evaluated using the package MENNDL where the models are trained over fewer iterations, therefore significantly reducing the amount of specialized human input and time required. This technique is quite common in industry for use in automated neural architectures but to the best of our knowledge, this is the first time it has been used to analyze neutrino/antineutrino physics data. In the coming years, the current running and future neutrino experiments will collect an unprecedented amount of information-dense data. As a result, the implementation of a ML algorithm will become necessary to extract the physics embedded in these data, and the MENNDL package will be of immense help in significantly reducing the required human time and associated time needed to produce an optimized ML model.

Acknowledgments

This document was prepared by members of the MINERvA Collaboration using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359. These resources included support for the MINERvA construction project, and support for construction also was granted by the United States National Science Foundation under Award No. PHY-0619727 and by the University of Rochester. Support for participating scientists was provided by NSF and DOE (U.S.A.); by CAPES and CNPq (Brazil);

2022 JINST 17 T08013

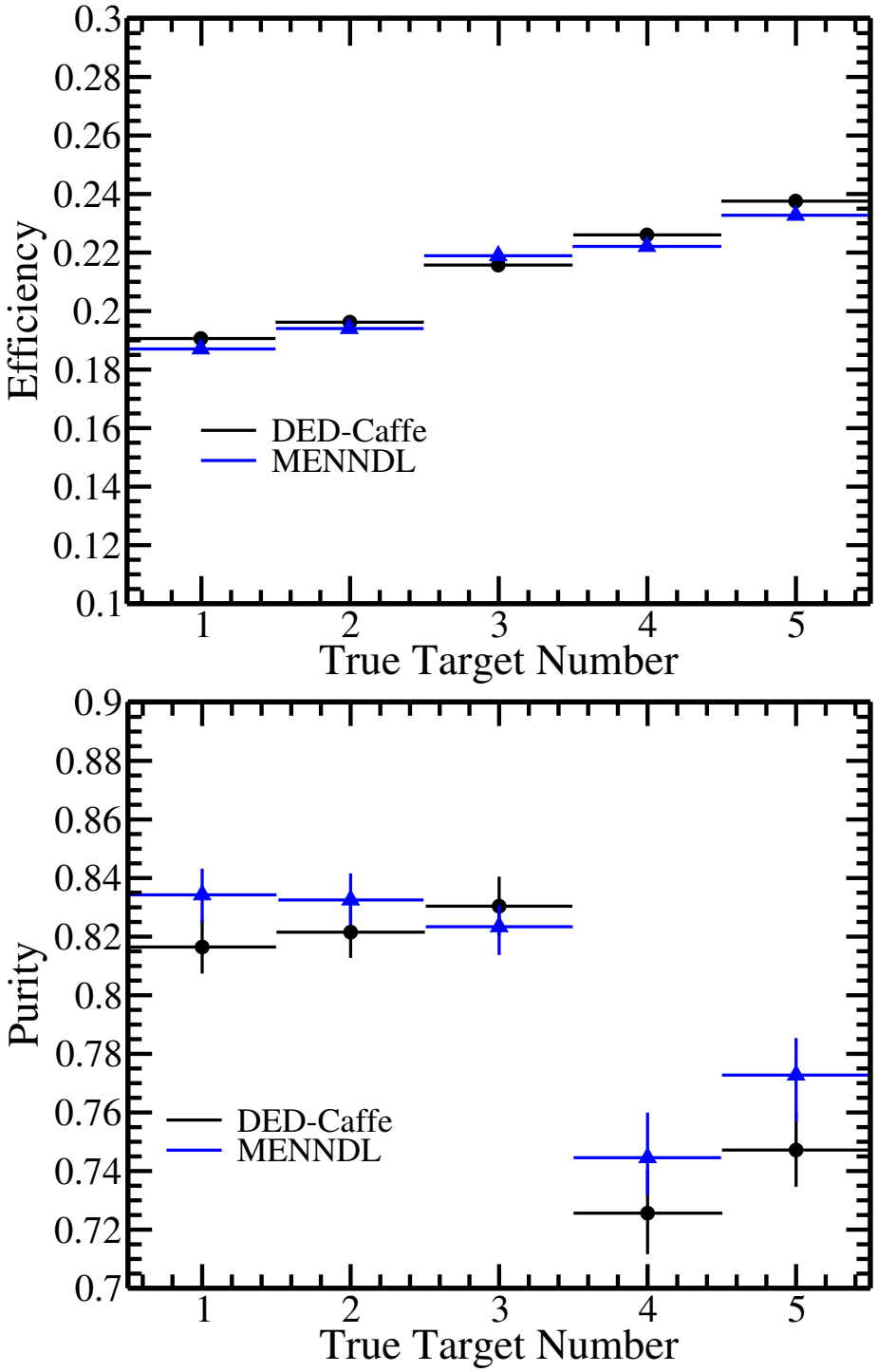


Figure 9. The results for efficiency and purity with respect to five different nuclear targets in the MINERvA detector. Two different machine learning models have been used to make efficiency plots for DIS sample: the domain-expert-designed(DED)-Caffe (black/circle) and MENNDL (blue/triangle).

by CoNaCyT (Mexico); by Proyecto Basal FB 0821, CONICYT PIA ACT1413, and Fondecyt 3170845 and 11130133 (Chile); by CONCYTEC (Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica), DGI-PUCP (Dirección de Gestión de la Investigación — Pontificia Universidad Católica del Peru), and VRI-UNI (Vice-Rectorate for Research of National University of Engineering) (Peru); NCN Opus Grant No. 2016/21/B/ST2/01092 (Poland); by Science and Technology Facilities Council (U.K.); by EU Horizon 2020 Marie Skłodowska-Curie Action; by a Cottrell Postdoctoral Fellowship from the Research Corporation for Scientific Advancement; by an Imperial College London President's PhD Scholarship. We thank the MINOS Collaboration for use of its near detector data. Finally, we thank the staff of Fermilab for support of the beam line, the detector, and computing infrastructure. The research here was also sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U.S. Department of Energy. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

[1] S.R. Young, D.C. Rose, T.P. Karnowski, S.-H. Lim and R.M. Patton, *Optimizing deep learning hyper-parameters through an evolutionary algorithm*, in *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments*, Austin, TX, U.S.A., 15 November 2015, pp. 1–5.

[2] S.R. Young, D.C. Rose, T.Johnston, W.T. Heller, T.P. Karnowski, T.E. Potok et al., *Evolving deep networks using HPC*, in *Proceedings of the Machine Learning on HPC Environments*, Denver, CO, U.S.A., 12–17 November 2017, pp. 1–7.

[3] MINERvA collaboration, *Reducing model bias in a deep learning classifier using domain adversarial neural networks in the MINERvA experiment*, 2018 *JINST* **13** P11020 [arXiv:1808.08332].

[4] MINERvA collaboration, *Design, Calibration, and Performance of the MINERvA Detector*, *Nucl. Instrum. Meth. A* **743** (2014) 130 [arXiv:1305.5199].

[5] MINOS collaboration, *The Magnetized steel and scintillator calorimeters of the MINOS experiment*, *Nucl. Instrum. Meth. A* **596** (2008) 190 [arXiv:0805.3170].

[6] G. Welch and G. Bishop, *An introduction to the Kalman filter*, Tech. Rep., University of North Carolina at Chapel Hill, U.S.A., 1995.

[7] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette et al., *Domain-adversarial training of neural networks*, *J. Mach. Learn. Res.* **17** (2016) 1.

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick et al., *Caffe: Convolutional Architecture for Fast Feature Embedding*, arXiv:1408.5093.

[9] M. Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*, arXiv:1603.04467

[10] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard et al., *Handwritten digit recognition with a back-propagation network*, in *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann, San Francisco, CA, U.S.A. (1989).

[11] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, *Gradient-based learning applied to document recognition*, *Proc. IEEE* **86** (1998) 2278.

2022 JINST 17 T08013

[12] V. Nair and G.E. Hinton, *Rectified linear units improve restricted Boltzmann machines*, in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Haifa, Israel, 21–24 June 2010, p. 807–814.

[13] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, U.S.A. (2016).

[14] K.P. Murphy, *Probabilistic Machine Learning: An introduction*, MIT Press, MA, U.S.A. (2022).

[15] T. Elsken, J. H. Metzen, F. Hutter et al., *Neural architecture search: A survey*, *J. Mach. Learn. Res.* **20** (2019) 1.

[16] L. Li and A. Talwalkar, *Random search and reproducibility for neural architecture search*, in *Proceedings of Conference on Uncertainty in Artificial Intelligence*, Tel Aviv, Israel, 22–25 July 2019, paper 129 [arXiv:1902.07638].

[17] M.F. Kasim, D. Watson-Parris, L. Deaconu, S. Oliver, P. Hatfield, D.H. Froula et al., *Building high accuracy emulators for scientific simulations with deep neural architecture search*, *Mach. Learn. Sci. Technol.* **3** (2021) 015013.

[18] P. Ren, Y. Xiao, X. Chang, P. yao Huang, Z. Li, X. Chen et al., *A comprehensive survey of neural architecture search*, *ACM Comput. Surv.* **54** (2022) 1.

[19] K.O. Stanley and R. Miikkulainen, *Evolving neural networks through augmenting topologies*, *Evolut. Comput.* **10** (2002) 99.

[20] B. Zoph and Q.V. Le, *Neural architecture search with reinforcement learning*, arXiv:1611.01578, 2016.

[21] B. Baker, O. Gupta, N. Naik and R. Raskar, *Designing neural network architectures using reinforcement learning*, arXiv:1611.02167, 2016.

[22] B. Ru, X. Wan, X. Dong and M. Osborne, *Interpretable neural architecture search via bayesian optimisation with weisfeiler-lehman kernels*, in *Proceedings of the International Conference on Learning Representations*, 3–7 May 2021 [arXiv:2006.07556].

[23] H. Liu, K. Simonyan, O. Vinyals, C. Fernando and K. Kavukcuoglu, *Hierarchical representations for efficient architecture search*, in *Proceedings of the International Conference on Learning Representations*, Vancouver, BC, Canada, 30 April–3 May 2018 [arXiv:1711.00436].

[24] S. Santra, J.-W. Hsieh and C.-F. Lin, *Gradient descent effects on differential neural architecture search: A survey*, *IEEE Access* **9** (2021) 89602.

[25] D. Lian, Y. Zheng, Y. Xu, Y. Lu, L. Lin, P. Zhao et al., *Towards fast adaptation of neural architectures with meta learning*, in *Proceedings of the International Conference on Learning Representations*, 27–30 April 2020.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, *ImageNet: A large-scale hierarchical image database*, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, U.S.A., 20–25 June 2009, pp. 248–255.

[27] H. Liu, K. Simonyan and Y. Yang, *Darts: Differentiable architecture search*, arXiv:1806.09055, 2018.

2022 JINST 17 T08013