

Adversarial training to improve robustness of deep neutrino classifiers in the NOvA experiment

Kevin Mulder

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London

Department of Physics and Astronomy
University College London

December 18, 2022

I, Kevin Mulder, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

The NOvA experiment is a long-baseline neutrino oscillation experiment. Consisting of two functionally identical detectors situated off-axis in Fermilab's NuMI neutrino beam. The Near Detector observes the unoscillated beam at Fermilab, while the Far Detector observes the oscillated beam 810 km away. This allows for measurements of the oscillation probabilities for multiple oscillation channels, $\nu_\mu \rightarrow \nu_\mu$, $\bar{\nu}_\mu \rightarrow \bar{\nu}_\mu$, $\nu_\mu \rightarrow \nu_e$ and $\bar{\nu}_\mu \rightarrow \bar{\nu}_e$, leading to measurements of the neutrino oscillation parameters, $\sin \theta_{23}$, Δm_{32}^2 and δ_{CP} .

These measurements are produced from an extensive analysis of the recorded data. Deep neural networks are deployed at multiple stages of this analysis. The Event CVN network is deployed for the purposes of identifying and classifying the interaction types of selected neutrino events. The effects of systematic uncertainties present in the measurements on the network performance are investigated and are found to cause negligible variations. The robustness of these network trainings is therefore demonstrated which further justifies their current usage in the analysis beyond the standard validation.

The effects on the network performance for larger systematic alterations to the training datasets beyond the systematic uncertainties, such as an exchange of the neutrino event generators, are investigated. The differences in network performance corresponding to the introduced variations are found to be minimal.

Domain adaptation techniques are implemented in the AdCVN framework. These methods are deployed for the purpose of improving the Event CVN robustness for scenarios with systematic variations in the underlying data.

Impact Statement

The NOvA experiment is situated at the leading edge of particle physics, taking data and enabling higher precision measurements of neutrino oscillation parameters. The work undertaken as part of this PhD has supported these efforts in multiple ways. Some examples include the detector calibration, optimization of data structures, support of the NOvA Testbeam experiment and direct data taking. As the measurements rely on analyses of the accumulated neutrino oscillation data, work on the deep neural networks which are deployed in these analyses has a direct impact on the produced results. This work includes training, optimization, evaluation and validation of various deep neural networks as well as education on the available computational resources and networks themselves.

The robustness studies provide increased confidence in the justified usage of the deep neural networks in this, and other, physics analyses. The developed framework, AdCVN, enables easily accessible tools for future robustness evaluations. Additionally it contains the means necessary to deploy several domain adaptation techniques. While the environment of the NOvA experiment with small domain shifts has been found to be limited for these techniques, this is not expected to be the case for other physics experiments. Likewise the robustness evaluation and domain adaptation techniques are not limited to one or several physics experiments, they can be applied in any industry which relies in some form on Machine Learning.

As this doctoral work is part of the Centre for Doctoral Training in Data Intensive Science it has been presented, discussed and disseminated across an

audience far extending beyond the field of particle physics. This has generated interest through many different outreach engagements such as presentations, projects, placements, seminars and other industry events.

Acknowledgements

The journey that was this PhD has been made possible and supported by many. Firstly I'd like to thank my family for their continual support and unyielding resilience to put up with my unending ramblings about physics, especially during the long lock-downs of the COVID-19 pandemic. Special thanks to my supervisor Ryan Nichol for stepping up during challenging times.

During my time at Fermilab I had the opportunity to get to know my fellow collaborators on the NOvA and NOvA testbeam experiment. In particular Alex Himmel, Micheal Baird, Karl Warburton, Teresa Lackey and Anne Norrick were both equally helpful with the professional and social aspects of collaboration and life at Fermilab. Additionally I'd like to thank Ablilash Yallappa Dombara whose friendship and support has been enriching to say the least.

At UCL, there are several people who have helped me in various ways throughout the PhD. For this they all have my gratitude, however I'd like to especially thank Davide Piras, Simeon Bash, Fern Pannell, Veera Mikola and Mustafa Özkaynak.

Contents

1	Neutrino interactions and oscillations	21
1.1	A brief history of the neutrino	21
1.2	The Standard Model	22
1.2.1	Electromagnetic interactions	23
1.2.2	Weak interactions	23
1.2.3	Strong interactions	24
1.2.4	Fermions in the Standard Model	24
1.3	Neutrinos in the Standard Model	25
1.4	Neutrino interactions	27
1.5	Neutrino mass	32
1.5.1	Majorana neutrinos	33
1.5.2	Seesaw Mechanism	35
1.6	Neutrino oscillations in vacuum	36
1.6.1	Muon neutrino disappearance	39
1.6.2	Electron neutrino appearance	41
1.7	Neutrino oscillations in matter	43
1.7.1	Mass hierarchy	44
2	Experimental status of neutrino oscillation measurements	46
2.1	Solar neutrinos	46
2.1.1	Solar neutrino experiments	48
2.2	Atmospheric neutrinos	52
2.2.1	Atmospheric neutrino experiments	54

	2.3 Reactor neutrinos	57
	2.3.1 Reactor neutrino experiments	58
	2.4 Accelerator neutrinos	60
	2.4.1 Long baseline experiments	62
	2.4.2 Short-baseline experiments	64
	2.5 Experimental values of the oscillation parameters	66
	2.5.1 Measurements of the NOvA experiment	69
3	The NOvA Experiment 71	
	3.1 Fermilab and the NuMI beamline	71
	3.2 Off-axis design	76
	3.3 Delivered beam	77
	3.3.1 Data selection	78
	3.4 Detector design	80
	3.4.1 The NOvA detector cell	81
	3.4.2 Liquid scintillator	84
	3.4.3 Wavelength Shifting Fibre	85
	3.4.4 Cell optical readout	86
	3.5 DAQ system	87
	3.5.1 Data concentrator module	89
	3.5.2 Timing system	91
	3.5.3 Triggers	94
4	Simulation and Calibration 96	
	4.1 NuMI Beamline simulation	96
	4.1.1 Neutrino interaction simulation	97
	4.1.2 Cosmic ray simulation	100
	4.2 Detector simulation	100
	4.2.1 GEANT4 detector geometry	101
	4.2.2 Photon generation and transport simulation	102
	4.2.3 Detector response	104

4.3	Calibration	109
4.3.1	Relative calibration	111
4.3.2	Absolute calibration	116
4.3.3	Transposer	118
5	Reconstruction	121
5.1	Clustering	121
5.1.1	4DSlicer	121
5.1.2	TDSlicer	124
5.2	Track reconstruction	127
5.2.1	Elastic arms vertex reconstruction	130
5.2.2	Fuzzy-K vertex reconstruction	132
5.2.3	Prong matching	136
5.3	Particle tracking	137
5.3.1	KalmanTrack reconstruction	139
5.3.2	BreakPointFitter	141
6	Machine Learning in NOvA	146
6.1	Artificial neural networks	147
6.1.1	Neurons	148
6.1.2	Activation function	149
6.2	Training	152
6.2.1	Loss functions	153
6.2.2	Backpropagation	154
6.2.3	Gradient descent and optimizers	156
6.3	Convolutional neural networks	159
6.3.1	The convolutional layer	162
6.3.2	The pooling layer	162
6.3.3	Batch normalization	163
6.3.4	Connected classifier	165
6.4	CVN architectures	165

	Contents	10
6.4.1	ResNet	166
6.4.2	Modified MobileNet	168
6.5	ML datasets	172
6.5.1	Generalization	172
6.6	Domain adaptation	173
6.6.1	Adversarial domain adaptation	175
6.6.2	Gradient reversal	178
6.6.3	Loss penalization	180
7	Domain shifts and adaptation in NOvA	182
7.1	Data selection	182
7.1.1	Data quality	183
7.1.2	Event quality	184
7.1.3	Containment	185
7.1.4	Cosmic rejection	186
7.2	Datafile formatting and processing	187
7.2.1	Data preprocessing	189
7.2.2	Dynamical data processing	190
7.3	Domain datasets	192
7.3.1	Neutrino event generator exchange	193
7.3.2	Systematic uncertainties	195
7.4	Computational environment	199
7.4.1	Keras and TensorFlow	201
7.4.2	The AdCVN framework	203
7.4.3	Custom components	207
7.4.4	Network evaluation	213
8	Validation of domain adaptation techniques on toy scenarios	228
8.1	Binary classification	229
8.1.1	Data generation	229
8.1.2	Network architectures	230

Contents	11
8.1.3 Training and evaluation	232
8.2 Binary classification with gradient reversal	236
8.3 Binary classification with loss penalization	240
8.4 Multi-class with loss penalization	244
8.4.1 Batchsize reduction	247
9 NOvA domain shifts impact and mitigation	252
9.1 AdCVN training procedure	252
9.2 Single domain trainings	255
9.3 Domain adaptation trainings	259
9.3.1 Domain adversarial	259
9.3.2 Gradient reversal	260
9.3.3 Robustness evaluation	263
9.3.4 Layer analysis	279
9.3.5 Loss penalization with memory	282
9.4 Discussion and future work	296
10 Conclusions	299
Appendices	300
A W boson jet binary classification	300
Bibliography	304

List of Figures

1.1	Beta decay spectrum.	22
1.2	The elementary particles of the Standard Model.	26
1.3	Feynman diagrams of generalized weak interactions.	29
1.4	Feynman diagrams of CC interactions.	31
1.5	The CC (anti) neutrino cross sections as a function of energy.	32
1.6	Neutrino oscillation spectrum of the 2020 NOvA ν_μ disappearance analysis	41
1.7	Feynman diagrams of coherent forward scattering on electrons	43
1.8	The two possible mass orderings.	45
2.1	The solar neutrino flux decomposed into contributions of neutrino producing solar processes.	47
2.2	Schematic overview of several solar neutrino experiments.	52
2.3	Neutrino production originating in the Earth's atmosphere.	54
2.4	SK measurement of upwards and downwards going neutrino events.	56
2.5	SK measurement of flux-averaged charged-current tau neutrino cross section.	57
2.6	KamLAND ratio $\bar{\nu}_e$ spectrum.	59
2.7	Schematic overview of several reactor neutrino experiments.	61
2.8	Reconstructed FD neutrino energy spectra.	69
2.9	Confidence level contours for the oscillation parameters.	70
3.1	Map of the off-axis beamline originating at Fermilab.	72

List of Figures	13
3.2 Schematic overview of the Fermilab accelerator complex. . . .	73
3.3 Expected number of neutrino events at both detectors for both modes.	75
3.4 Schematic overview of the NuMI beamline.	76
3.5 Off-axis energy spectra and expected event rates as a function of energy.	78
3.6 Time series data of the delivered NuMI beam up to March 2020	79
3.7 The bare PVC cells stacked in alternating planes to construct the frame of the NOvA detectors.	82
3.8 The NOvA near and far detectors.	83
3.9 Normalized emission spectra for the NOvA scintillator.	85
3.10 Schematic overview of a detector cell and plane structures. . .	86
3.11 Optical connector with WLS fibres.	88
3.12 The 32 pixel APD.	88
3.13 The front end board.	90
3.14 A schematic overview of the FD DAQ system.	91
3.15 Schematic overview of the NOvA timing system.	92
3.16 Illustration of the DCS algorithm.	94
4.1 Data-MC comparisons of the NOvA genie tune.	99
4.2 APD response distributions for various excess noise models. .	106
4.3 CR-RC circuit.	106
4.4 Positive muon stopping power in pure copper.	110
4.5 An illustration of the tri-cell calibration criteria.	111
4.6 Average thresholding and shielding correction factors.	113
4.7 Post TS correction photo-electrons per pathlength as a function of W	114
4.8 Attenuation fits for both views and detectors.	115
4.9 Profiles of the mean reconstructed to true energy ratios as a function W	116

4.10 The deposited energy of stopping muons as a function of distance from the track endpoint for both detectors. 117

4.11 Flow diagram showing the order of modules required to be run for the completion of the relative calibration. 119

5.1 Diagram illustrating clustering with the DBSCAN algorithm. . 122

5.2 Demonstration of the centroid clustering algorithm. 126

5.3 Comparison of the 4DSlicer and TDSlicer on an event. 126

5.4 Performance comparison of the 4DSlicer and TDSlicer. 128

5.5 Example FD slice with elastic arms and multi-hough lines reconstruction applied. 130

5.6 Goodness of fit for the elastic arms vertex reconstruction method. 132

5.7 The event from Figure 5.5 reconstructed using FuzzyK. 135

5.8 Completeness of FuzzyK produced prongs as a measure of performance. 136

5.9 Prong matching of a ν_e CC QE event. 138

5.10 Schematic overview of the KalmanTrack reconstruction method. 140

5.11 Example of KalmanTrack reconstructed tracks for a FD event. 141

5.12 Completeness of KalmanTrack produced tracks as a measure of performance. 142

5.13 Schematic overview of the BreakPointFitter. 143

5.14 Distribution of BPF scattering surfaces. 144

5.15 Example of a BPF reconstructed neutrino candidate event in the FD. 145

6.1 Schematic overview of simple MLP neural networks. 148

6.2 Schematic overview of a neuron. 149

6.3 Overview of single fold activation functions which depend singularly on x 150

6.4 Schematic of the final two layers ($L-1, L$) of a neural network. 155

6.5 Gradient descent along a non-convex surface. 157

6.6	Schematic overview of the two branch Siamese network architecture.	160
6.7	Examples of typical pixelmap images which form the CVN input.	161
6.8	Schematic overview of a convolutional neural network.	163
6.9	Schematic overview of the operations a convolutional and pooling layer perform in a neural network.	164
6.10	Schematic overview of the AlexNet convolutional architecture.	166
6.11	The inception module and GoogLeNet architecture.	167
6.12	Example of a residual learning block.	167
6.13	Overview of various length ResNet architectures.	168
6.14	Depthwise separable convolutions.	171
6.15	Under, over and appropriate fitting examples.	173
6.16	Overview of GANs for the purpose of generating images.	176
6.17	Ensemble adversarial domain adaptation training procedure.	177
6.18	Schematic overview of adversarial domain adaptation in a single network architecture.	179
7.1	GENIE dataset broken down by interaction classes	194
7.2	GiBUU dataset broken down by interaction classes	194
7.3	Impact of the flux corrections on the GENIE and GiBUU datasets	195
7.4	Summary of the systematic uncertainties in the 2020 analysis	196
7.5	Domain shifted calibration dataset broken down by interaction classes	198
7.6	Domain shifted light level dataset broken down by interaction classes	199
7.7	Schematic overview of the WC-IC	200
7.8	Example of a simple computational graph	202
7.9	Schematic overview of a network architecture with eventweighting	209
7.10	Schematic overview of a network architecture with loss penalization and memory layers.	214
7.11	Example training curves	216

List of Figures	16
7.12 Schematic overview of a network output spectrum	219
7.13 Schematic overview of a confusion matrix	220
7.14 Two-sample KS and Kuiper statistics	226
8.1 Schematic overview of the binary classification toy scenario. .	230
8.2 Schematic overview of multiple network configurations	233
8.3 Pretrained evaluation on the binary classification toy scenario	234
8.4 Loss curve domain adversarial training	235
8.5 Evaluation on the binary classification toy scenario post adver- sarial training	236
8.6 Loss curves for training with and without the gradient reversal layer on the binary classification scenario	239
8.7 Evaluation of the gradient reversal technique on the binary classification toy scenario	240
8.8 Decision surfaces of the loss penalization technique on the binary classification scenario	244
8.9 Schematic overview of the multiclass classification toy scenario.	245
8.10 PID spectra of the loss penalization technique on the multiclass classification scenario	246
8.11 Decision surfaces of the loss penalization technique on the mul- ticlass classification scenario	248
8.12 Impact of batchsize reduction on the standard model training on the multiclass classification scenario	249
8.13 Impact of batchsize reduction on the loss penalized training on the multiclass classification scenario	249
8.14 Impact of batchsize reduction on the loss penalized training with memory on the multiclass classification scenario	250
9.1 Training curves for domain classifiers	257
9.2 PID spectra of the domain classifiers	258

9.3	End of training losses and accuracies for ResNet18 networks with and without GR layers	262
9.4	Training accuracy curves for a suite of networks trained with and without GR layers on the GENIE vs GiBUU domain shift	264
9.5	Training accuracy curves for a suite of networks trained with and without GR layers on the light level domain shift	265
9.6	Training accuracy curves for a suite of networks trained with and without GR layers on the calibration domain shift	266
9.7	PID spectra of the ν_μ class for a suite of modified mobinenets trained with and without the GR layer on the light level domain shift	267
9.8	PID spectra of the domains for a suite of modified mobinenets trained with and without the GR layer on the GENIE vs GiBUU domain shift	268
9.9	PID spectra of the domains for a suite of modified mobinenets trained with and without the GR layer on the light level domain shift	269
9.10	PID spectra of the domains for a suite of modified mobinenets trained with and without the GR layer on the calibration domain shift	270
9.11	PR curves of the domains for a suite of modified mobinenets trained with and without the GR layer on the light level domain shift	271
9.12	ROC curves of the domains for a suite of modified mobinenets trained with and without the GR layer on the light level domain shift	272
9.13	Summary statistics of the domain PID spectra for a suite of modified mobinenets trained with and without the GR layer on the domain shifts	273

9.14	Summary statistics of the class PID spectra for a suite of modified mobinenets trained with and without the GR layer on the GENIE vs GiBUU domain shift	276
9.15	Summary statistics of the class PID spectra for a suite of modified mobinenets trained with and without the GR layer on the light level domain shift	277
9.16	Summary statistics of the class PID spectra for a suite of modified mobinenets trained with and without the GR layer on the calibration domain shift	278
9.17	Schematic overview of the layer analysis locations in the network architecture	280
9.18	Node response summary distributions	281
9.19	Unweighted means of the summary statistics for the domain combinations of the network layers	283
9.20	Training accuracy curves for memory enhanced loss penalized network trainings	284
9.21	Training curves for memory enhanced loss penalized network trainings with relative strength scheduling	287
9.22	v_μ class PID spectra for memory enhanced loss penalized network trainings with relative strength scheduling	288
9.23	v_μ class ROC curves for memory enhanced loss penalized network trainings with relative strength scheduling	289
9.24	v_μ class PR curves for memory enhanced loss penalized network trainings with relative strength scheduling	290
9.25	v_μ class ROC curves for non domain adaptational trainings for the GENIE vs GiBUU domain shift	291
9.26	Summary statistics for memory enhanced loss penalized network trainings with relative strength scheduling	292
9.27	Summary statistics for memory enhanced loss penalized network trainings with relative strength scheduling	294

9.28	Summary statistics for memory enhanced loss penalized network trainings with relative strength scheduling	295
A.1	Approximate median significane as a function of relative strength	303

List of Tables

2.1	Overview of solar neutrino experiments	53
2.2	Overview of reactor neutrino experiments	60
2.3	Overview of long baseline accelerator neutrino experiments.	65
2.4	Overview of experiments contribution the current best determination of the neutrino oscillation parameters.	67
2.5	Three flavour oscillation parameters from the NuFIT fit to global data.	68
2.6	Best fit values of the oscillation parameters.	70
3.1	Purity of the FD selected samples.	79
3.2	Event counts at the NOvA FD.	80
4.1	Materials used in the GEANT4 detector geometry simulations	102
7.1	Cosmic rejection selection efficiency	187
9.1	CVN training hyperparameters	254
9.2	Range of relative strengths for the loss penalization method with memory	286

Chapter 1

Neutrino interactions and oscillations

1.1 A brief history of the neutrino

The history of the neutrino starts approximately one century ago with the energy spectrum of beta decay. Unlike the earlier measurements of the alpha and gamma spectra, this was found to be continuous [1]. This appeared to break the law of conservation of energy, as the total amount of energy before the decay was no longer equivalent to that after the decay had occurred. The continuous beta decay spectrum is shown in Figure 1.1.

Wolfgang Pauli wrote a letter in 1930 which offered a solution to this problem [3]. Three years later Enrico Fermi incorporated the ideas previously set out by Pauli and formally published a theory of beta decay. The described solution introduced a new particle, dubbed the neutrino by Fermi, which would balance out the description of beta decay. This new particle was to be the carrier of the missing energy such that the conservation of energy holds. The properties of the neutrino were thus inferred from the conservation of energy, momentum and angular momentum.

The existence of the neutrino would be experimentally confirmed in 1956 through the Cowan-Reines experiment [4]. The unique signature of beta capture was used for this purpose. Antineutrinos produced in a nuclear reactor interact with a proton to produce both a neutron and positron. Both of these products

In this thesis natural units are used $\hbar = c = 1$ unless explicitly stated otherwise

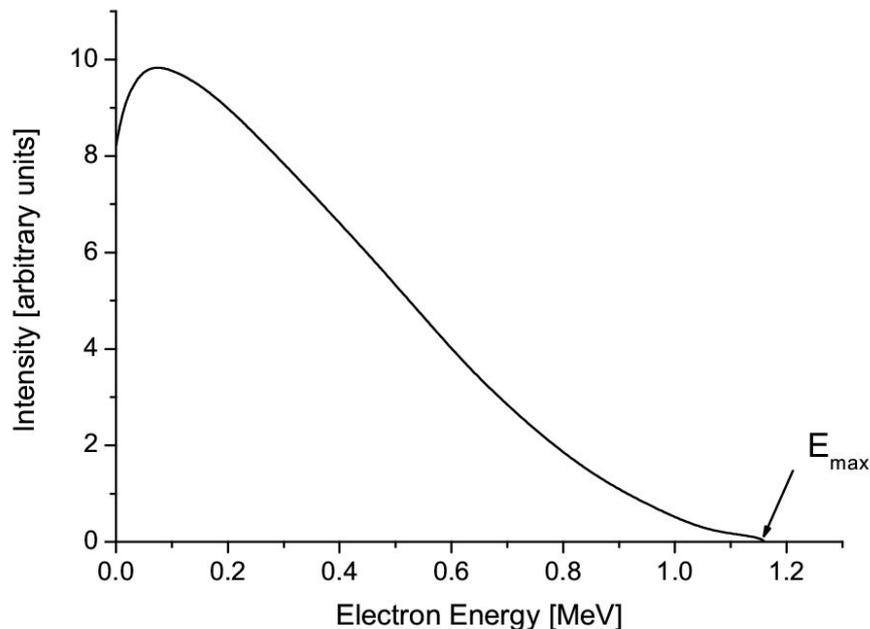


Figure 1.1: Beta decay spectrum of Bismuth-210. Courtesy of [2].

release gamma rays when they interact further, two from electron-positron annihilation and one from neutron capture in a nucleus.

Other postulated parts of the neutrino theory would later be experimentally confirmed as well. This was done through the examination of positively and negatively charged pion decays [5]. As was the case with the beta decay, the interaction is unbalanced due to the missing energy. However since muons and not electrons are the decay products here, a different type of neutrino was found. The different types of neutrinos are called flavours. Another neutrino flavour was found to be the tau neutrino [6].

1.2 The Standard Model

The Standard Model, also referred to as the Weinberg-Glashow-Salam Model appropriately named after Sheldon Glashow, Abdus Salam and Steven Weinberg [7], is a gauge quantum field theory describing elementary particles and most of their interactions. It obeys the global Poincaré symmetry, ensuring that the theory is compatible with special relativity and possesses inertial frame in-

variance. The internal symmetry is given by the local $SU(3) \times SU_L(2) \times U(1)_{em}$ gauge symmetry and is associated with the described interactions.

The elementary particles are split into two distinct groups, the fermions and bosons. The fermions possess half-integer spin ($\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \text{etc}$) and therefore obey Fermi-Dirac statistics. These specify that no two identical fermions can occupy the same quantum state which is referred to as the Pauli exclusion principle. When two fermions have the same values for their respective quantum numbers they occupy the same quantum state. In contrast to the fermions, bosons possess integer spin ($0, 1, 2, \text{etc}$) and obey Bose-Einstein statistics. Unlike the Fermi-Dirac statistics there is no exclusion requirement placed on the quantum states of bosons, meaning they can occupy the same quantum state.

The interactions described within the Standard Model are the electromagnetic, weak and strong interactions. These are all mediated through force-carrying particles, the Gauge Bosons.

1.2.1 Electromagnetic interactions

The first interaction corresponds to the local $U(1)_{em}$ gauge symmetry. A direct consequence of invariance under this symmetry is the coupling of matter and gauge fields leading to interaction terms between them. Following Noether's theorem [8] electromagnetic charge is found to be a conserved quantity. The accompanying force-carrying gauge boson is the massless photon.

1.2.2 Weak interactions

The second interaction corresponds to the local $SU(2)_L$ gauge symmetry. The L indicates that the symmetry acts differently on left- and right-handed fields. The left-handed fields transform as doublets, while the right-handed fields transform as singlets. Given the $SU(2)_L$ gauge symmetry, the theory describing this interaction is a Yang-Mills theory. Following Noether's theorem, the associated conserved quantity is the weak isospin. There are three accompanying force-carrying bosons, two electromagnetically charged ones W^\pm with weak isospins ± 1 and a single neutral one Z with weak isospin 0. All of these are massive

particles giving rise to a limited interaction range. The masses of the W and Z bosons are larger than the other force-carrying particles in the Standard Model. The range of this interaction is therefore smaller than those of the others described in the Standard Model. Additionally the apparent strength of the weak interaction at the energies of the (W,Z) boson masses is lower than that of the electro-magnetic and strong interactions. It is for these reasons the weak interaction is referred to as weak.

1.2.3 Strong interactions

The third and final interactions correspond to the local $SU(3)$ gauge symmetry. Like the previous symmetry, the Yang-Mills Theory can be used to describe this interaction. Following Noether's theorem the associated conserved quantity is the colour charge. There are eight force-carrying particles mediating this interaction, called gluons. The gluons are massless, however the range of this interaction is limited. Due to its high strength compared to the other interactions it is called the strong interaction. Besides the high strength it also has the ability to form flux tubes which occur when two colour charged particles are separated. This results in the creation of new particles through the process of hadronization. This leads to colour confinement, where isolation of colour charged particles is not possible as the newly created particles form into bound pairs.

1.2.4 Fermions in the Standard Model

The fermions constitute most of the recognizable matter. Within the Standard Model further classification is possible. These divisions of fermions are made based on the differing values of their quantum numbers. The first group of fermions are the quarks. These possess mass and have a colour, electric and weak charge. Therefore the quarks can participate in the electromagnetic, weak and strong interactions. Six separate quark flavours can be distinguished, the up (u), down (d), charm (c), strange (s), top (t) and bottom (b) quarks. Each of these has an associated antiparticle, which are defined as being equivalent

to their counterparts except for having negated values of their charges. All the quarks can be further subdivided based on their mass and electric charge. For the masses the six quarks are split into three pairs (u, d) , (c, s) , (t, b) . Each pair corresponds to a mass generation. For the electric charge, (u, c, t) have charge $+\frac{2}{3}$ and (d, s, b) have charge $-\frac{1}{3}$. Due to the properties of the strong interaction, quarks form colourless bound states called hadrons, which can be further split into the mesons and baryons.

The second group of fermions are the leptons. These possess no colour charge, which means they do not interact via the strong force. The six leptons are the electron (e), electron neutrino (ν_e), muon (μ), muon neutrino (ν_μ), tau (τ) and tau neutrino (ν_τ). As with the quarks there are multiple ways of splitting the six leptons (and their antiparticles). The e, μ, τ leptons are electrically charged with a value of -1 , the neutrinos are neutral with a charge of 0 . Three flavour pairs can be distinguished, (e, ν_e) , (μ, ν_μ) , (τ, ν_τ) .

A full overview of the Standard Model contents can be seen in Figure 1.2.

1.3 Neutrinos in the Standard Model

As neutrinos are fermions and follow Fermi-Dirac statistics the general form of a Lagrangian describing free fermions can be used for free neutrinos. It takes the following form,

$$L = \bar{\psi}(i\gamma^\mu \partial_\mu - m)\psi \quad (1.1)$$

where the neutrino fields ψ are represented as 4-component Dirac spinors and γ^μ are the gamma matrices. The Dirac equation and its conjugate form can be found through the Euler-Lagrange equations,

$$\frac{\partial L}{\partial \bar{\psi}} - \frac{\partial}{\partial x_\mu} \frac{\partial L}{\partial (\partial_\mu \bar{\psi})} = (\partial_\mu \gamma^\mu - m)\psi = 0 \quad (1.2)$$

These fields are required to be invariant under the global spacetime symmetries of the proper orthochronous Lorentz group. This is a subgroup of the Poincaré group. Where the Poincaré group covers spacetime transforma-

Standard Model of Elementary Particles

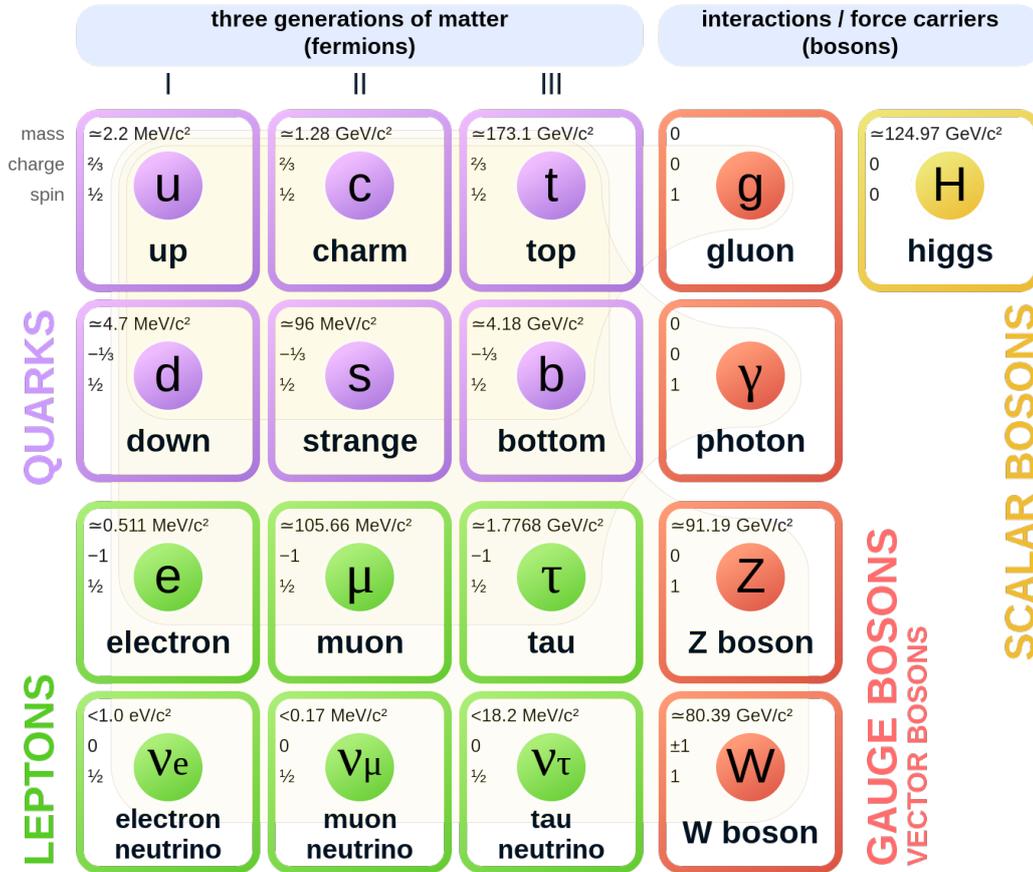


Figure 1.2: The elementary particles of the Standard Model. Courtesy of [9].

tions in the forms of boosts, rotations, translations and reflections, the proper orthochronous Lorentz group $SO^+(1,3)$ only covers the boosts and rotations. The Lorentz group decomposes into the following irreducible spin $\frac{1}{2}$ representations, $SU_L(2) \oplus SU_R(2)$, resulting in chiral Weyl representations which are left- and right-handed. The Dirac spinors can be decomposed into their left- and right-handed components,

$$\psi = \begin{pmatrix} \psi_L \\ \psi_R \end{pmatrix} \quad (1.3)$$

1.4 Neutrino interactions

Given that the neutrinos are electrically neutral fermions they only interact via the weak force. As mentioned previously the weak interaction does not couple to right- and left-handed chiral states equally. The difference stems from the form of the weak currents. The inclusion of both a vector and axial component ($V - A$) in the weak current ensures that the resulting interactions allow for parity violation. Parity transformation is the inversion of the spatial coordinate(s). Under parity transformations an axial vector remains unchanged while a vector becomes negative. While originally thought to be conserved, Lee and Yang proposed the possibility of parity violation in 1956 [10] and Wu et al. experimentally confirmed it in 1957 [11].

The weak current for interactions mediated via the charged W^\pm bosons is given by,

$$J^\mu = \frac{1}{2} \bar{\psi} \gamma^\mu (1 - \gamma^5) \psi + h.c. \quad (1.4)$$

The form of the weak neutral current differs by the value of its ($V - A$) coupling strengths g_V and g_A ,

$$J^\mu = \frac{1}{2} \bar{\psi} \gamma^\mu (g_V - g_A \gamma^5) \psi + h.c. \quad (1.5)$$

Using the chiral projection operator on the Dirac spinors, they can be decomposed into their left- and right-handed components,

$$\psi_L = \frac{(1 - \gamma^5)}{2} \psi, \quad \psi_R = \frac{(1 + \gamma^5)}{2} \psi \quad (1.6)$$

Alternatively for the dirac adjoint,

$$\bar{\psi}_L = \bar{\psi} \frac{(1 + \gamma^5)}{2}, \quad \bar{\psi}_R = \bar{\psi} \frac{(1 - \gamma^5)}{2} \quad (1.7)$$

Applying this decomposition on the weak charged and neutral currents, the coupling of the weak interaction to the left- and right-handed states can be examined separately. Using the properties of the gamma matrices to simplify

the form of the current,

$$J^\mu = \frac{1}{2}\bar{\psi}\gamma^\mu(1-\gamma^5)\psi = \frac{1}{4}\bar{\psi}\gamma^\mu(1-\gamma^5)^2\psi = \frac{1}{4}\bar{\psi}(1+\gamma^5)\gamma^\mu(1-\gamma^5)\psi = \bar{\psi}_L\gamma^\mu\psi_L \quad (1.8)$$

Here the weak interaction clearly couples to either the left-handed chiral particle states or the right-handed antiparticle states.

The left-handed fermions possess a non-zero weak isospin and can be grouped into weak isospin doublets whereas the right-handed fermions are consigned to singlets. While the right-handed fermions cannot interact via the weak charged current, they can couple to the weak neutral current. This holds only for the case of electrically charged right-handed fermions, as this charge contributes to the total weak hypercharge. The contribution is given by,

$$Y_W = 2(Q - T_3) \quad (1.9)$$

where Q represent the electric charge, T_3 the third component of the weak isospin and Y_W the total weak hypercharge.

From experimental observations [4] neutrino interactions appear¹ to conserve lepton flavour number. This indicates that the number of leptons from each generation before and after the interaction should be equivalent, resulting in lepton flavour conservation. Consequently either the neutrino interacts with a charged lepton from the same generation, or with a neutrino having the same flavour as itself. The lepton flavour numbers are unique to each generation of lepton, the sum (or total) lepton number is conserved. The antiparticles carry a negative lepton number to their particle counterpart.

With these constraints the neutrino interactions can take several generalized forms. The simplest of these are the neutral-current (*NC*) interactions. Here the interaction is mediated through the exchange of the neutral Z boson. The incoming neutrino scatters off a nucleon and maintains its lepton flavour. As the Z boson couples to each lepton flavour equally the flavour of the incoming

¹Neutrino oscillations explicitly violate lepton number conservation.

(and outgoing) neutrino is not generally restricted to a specific lepton flavour. The general form for NC interactions can be represented as,

$$\nu_x + N \rightarrow \nu_x + X \quad (1.10)$$

where N represents the nucleon on which the incident neutrino scatters and X represents the ensemble of particles present in the final state. The exact form of X depend on the specific circumstances of the interaction. Antineutrinos can scatter off nucleons in a similar way to the scattering shown in equation (1.10).

The remainder of the neutrino interactions can be categorized as charged current (CC) interactions. These are mediated through the exchange of the charged W^\pm bosons. The incoming neutrino scatters off a nucleon and transforms into the associated lepton of the same generation. The general form for CC interactions can be represented as,

$$\nu_l + N \rightarrow l + X \quad (1.11)$$

In this expression the l indicates one of the three lepton flavours. The general form for the antineutrinos would be similar to equation (1.11), although the lepton should be converted into an antilepton as well. Several examples of the generalized NC and CC interactions are shown in Figure 1.3, utilizing the formalism of Feynman diagrams.

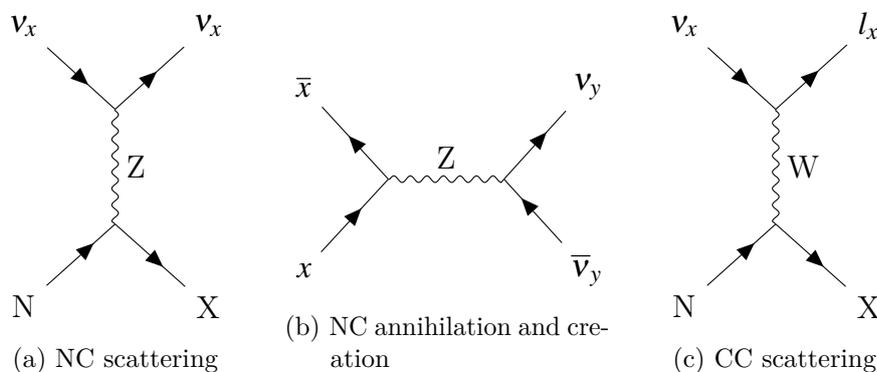


Figure 1.3: Feynman diagrams of generalized weak interactions, where x and y indicate particles originating from the same lepton generation

The charged current interactions can be further broken down by the effects on the nucleus or the resulting final state. Some of the relevant types are quasi-elastic (*QE*) scattering, resonant pion production (*RES*), coherent scattering (*COH*), meson exchange current (*MEC*) and deep inelastic scattering (*DIS*). In the case of the incident neutrino scattering on a nucleon in the nucleus it would be a quasi-elastic interaction. Otherwise if it would have scattered on a pair of nucleons in the nucleus it would be a meson exchange current interaction. It can also scatter from the entire nucleus as a whole, in which case it would be a coherent scattering interaction. Another type of scattering which can occur is the deep inelastic scattering. Here the neutrino scatters on a single quark inside a nucleon and knocks it out of the nucleus. This typically results in hadronization in the final state. Resonant pion production interactions occur when the nucleon is excited into an intermediate resonant baryon state which then decays into a nucleon and pion. In Figure 1.4 the previously introduced charged current interactions are shown, with the exception of the coherent scattering as this interaction is equivalent to the quasi-elastic scattering under the exchange of a single nucleon to the entire nucleus. There are scattering types beyond those previously discussed, such as fragmentation in which the nucleus is broken up into multiple fragments or final state interactions in which the interacting particles can undergo multiple separate interactions with the particles within the nucleus before exiting.

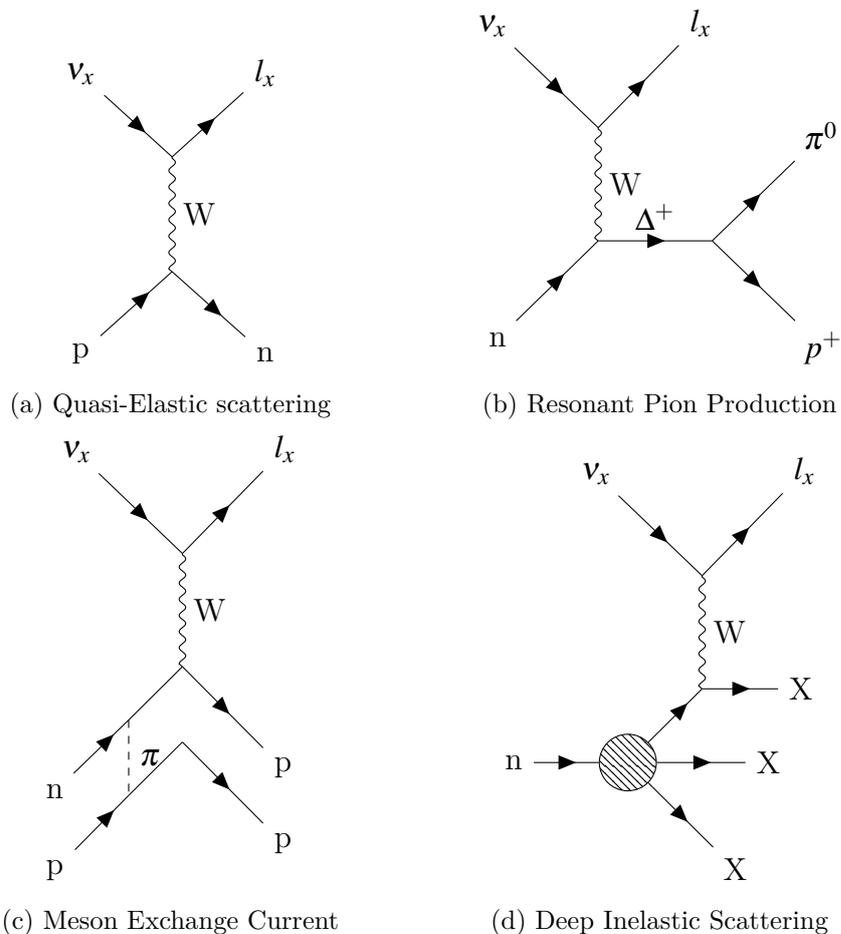


Figure 1.4: Feynman diagrams of a number of charged current interaction subtypes.

The probability that each of these processes occur depends on many factors [12], but is measured and expressed by the cross section. The total cross section for the neutrino interaction is the sum of the cross sections of all the possible interaction types. As the energy dependence is not uniform between the different interaction types, individual contributions to the total cross section vary with neutrino energy. That dependence is shown in Figure 1.5.

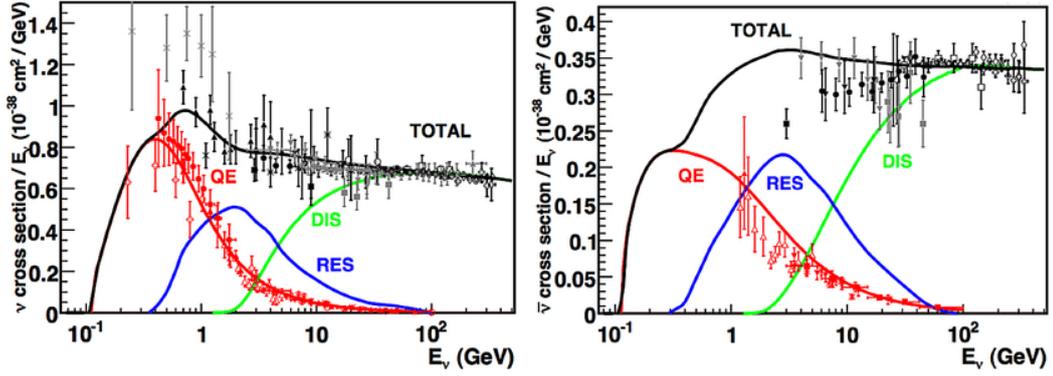


Figure 1.5: Total neutrino (left) and antineutrino (right) CC cross sections per nucleon divided by neutrino energy and plotted as a function of energy. Adapted from Figure 9, courtesy of [12].

1.5 Neutrino mass

Returning to the Lagrangian for massive free fermions (1.1), a Lagrangian for neutrinos can be constructed,

$$L = \bar{\psi}_\nu (i\gamma^\mu \partial_\mu) \psi_\nu - m \bar{\psi}_\nu \psi_\nu \quad (1.12)$$

Extending the mass term of equation (1.12) and decomposing into left- and right-handed neutrino states,

$$m \bar{\psi}_\nu \psi_\nu = m (\bar{\psi}_{\nu,R} \psi_{\nu,L} + \bar{\psi}_{\nu,L} \psi_{\nu,R}) \quad (1.13)$$

where $\bar{\psi}_{\nu,L} \psi_{\nu,L} = 0$ and $\bar{\psi}_{\nu,R} \psi_{\nu,R} = 0$ from the previously introduced projection operators.

The difference in weak isospin for the left-handed ($+\frac{1}{2}$) and right-handed (0) neutrino fields, coupled with their neutral electric charge, results in the surviving mass terms being charged and thus not gauge invariant. This invariance issue is not unique to the neutrino fields. The left-handed leptons with a weak isospin of $-\frac{1}{2}$ couple to the right-handed leptons with 0 weak isospin. Using the electric charge, the total weak hypercharge of the left- and right-handed lepton fields are $Y_W = -1$ and $Y_W = -2$ respectively. For these leptons the invariance issue is resolved through coupling to a field with the required properties. This field

would need to be electrically neutral ($Q = 0$) and form part of a weak doublet with $T_3 = +\frac{1}{2}$. The weak hypercharge then becomes $Y_W = -1$. Because taking the adjoint of a field negates its charges, the total hypercharge of the coupled term becomes 0. Therefore coupling of this field to the leptonic mass term would restore the neutrality and thus the gauge invariance of the mass terms in the Lagrangian. The field with these properties is the Higgs boson.

Analogous to the leptons, coupling to an additional field could restore gauge invariance to the neutrino mass terms. Assuming the sterile right-handed neutrino fields exists, a field with different quantum numbers is required. Here a field which is electrically neutral $Q = 0$ and has a third component of the weak isospin $T_3 = -1$ would suffice. This field could not form part of a doublet $(-\frac{1}{2}, +\frac{1}{2})$ with this specific isospin, instead it would be part of a Higgs triplet $(-1, 0, 1)$. Coupling to this Higgs field would result in a total weak hypercharge $Y_W = 0$ and would therefore restore gauge invariance for the neutrino fields. The Higgs triplet is not included in the Standard Model and none of its theorized effects have been experimentally observed [13].

1.5.1 Majorana neutrinos

The generation of neutrino mass is not confined to the same mechanism as the massive leptons, another mechanism has been proposed by Majorana [14]. This mechanism relies on the neutrino being its own antiparticle and thus being a Majorana particle. To see how this could be correct another symmetry operator is needed, one which allows for the transformation of a particle to its antiparticle counterpart and vice versa. This is the Charge Conjugation operation. For the Dirac spinors, in the chiral Weyl representation, it can be defined as,

$$\psi^c = \eta C \bar{\psi}^T \quad (1.14)$$

where η is a phase factor taken to be 1. If the charge conjugate of a chirally decomposed spinor is taken its chirality is flipped,

$$\psi_R = C \bar{\psi}_L^T \quad (1.15)$$

This can be derived by using the Euler-Lagrange equation (1.2) on the chirally decomposed Lagrangian for free fermions. This results in two coupled equations, which can be transformed into each other through consecutive applications of the hermitian conjugate operation and (anti-)commutation of the gamma matrices. Furthermore a cross-check is possible through the application of the left-handed chiral projection operator on the right-handed field of equation (1.15), which returns a value of zero as expected for a right-handed field.

Equation (1.15) allows the expression of the chiral decomposed fermion fields in only left-handed components,

$$\psi = \psi_L + \psi_R = \psi_L + C\bar{\psi}_L^T = \psi_L + \psi_L^c \quad (1.16)$$

where ψ_L^c is defined as the charge conjugated field. Taking the charge conjugation of the fields expressed in this way returns itself, suggesting that this fermion satisfies the Majorana condition and is its own antiparticle. Due to the involvement of the charge conjugation operator this expression only holds for neutral particles, such as the neutrinos. Being electrically neutral, its weak charge is not conserved due to the broken symmetry.

1.5.1.1 Majorana lepton number violation

A separate consequence of neutrinos possibly being Majorana particles would be their potential for lepton number violation. In the Dirac description neutrinos consist of both particles and antiparticles with lepton number of $L = +1$, $L = -1$ respectively. Assigning these numbers to the Majorana neutrinos would result in frequent violations on the order of $\Delta L = \pm 2$. An example of this is actively being searched for and is known as neutrinoless double beta decay [15] [16].

With the definition of the Majorana fields in equation (1.16) a new mass term can be constructed which couples the left to the right-handed fermion field, where the right-handed field is now expressed in terms of the left-handed field,

$$L_M = -m\bar{\psi}_L^c\psi_L \quad (1.17)$$

The resulting mass term is now purely dependant on the left-handed field components, suggesting no need for additional sterile right-handed fields to be introduced into the Standard Model. However this doesn't hold due to the gauge invariance of this new term. Unlike in the Dirac description, the right-handed field now has a nonzero isospin component as it is the transformed left-handed field. To restore gauge invariance coupling to a Higgs triplet would once more be required, as described in section 1.5.

1.5.2 Seesaw Mechanism

Utilizing elements from both the Dirac and Majorana descriptions a third mechanism can be defined. The mass term is described in its most general form from the left- and right-handed chiral decompositions and the Majorana conjugate fields,

$$\begin{aligned}\bar{\Psi}\Psi &= \frac{1}{2}(\bar{\Psi}_L\Psi_R + \bar{\Psi}_L(\Psi^c)_R + (\bar{\Psi}^c)_L\Psi_R + (\bar{\Psi}^c)_L(\Psi^c)_R) \\ &\quad + \frac{1}{2}(\bar{\Psi}_R\Psi_L + \bar{\Psi}_R(\Psi^c)_L + (\bar{\Psi}^c)_R\Psi_L + (\bar{\Psi}^c)_R(\Psi^c)_L) \\ &= (\bar{\Psi}_L\Psi_R + \bar{\Psi}_R\Psi_L) + \frac{1}{2}(\bar{\Psi}_L(\Psi_L)^c + \bar{\Psi}_L^c\Psi_L) + \frac{1}{2}(\bar{\Psi}_R(\Psi_R)^c + \bar{\Psi}_R^c\Psi_R)\end{aligned}\tag{1.18}$$

where the chirality flipping characteristic of the charge conjugation operator is expressed as $(\Psi_L)^c = (\Psi^c)_R$.

This can be rewritten in Hermitian terms,

$$\bar{\Psi}M\Psi = m_D\bar{\Psi}_L\Psi_R + m_L\bar{\Psi}_L(\Psi_L)^c + m_R\bar{\Psi}_R(\Psi_R)^c + h.c.\tag{1.19}$$

and further expressed in matrix form,

$$\bar{\Psi}M\Psi = \left((\bar{\Psi}_L)^c \quad \bar{\Psi}_R \right) \begin{pmatrix} m_L & m_D \\ m_D & m_R \end{pmatrix} \begin{pmatrix} \Psi_L & (\Psi_R)^c \end{pmatrix} + h.c.\tag{1.20}$$

As the mass matrix M contains non-zero off-diagonal terms the chiral neutrino fields cannot be mass eigenstates. They therefore do not have a definite mass. Similar to the quark mixing matrix, a rotation applied to the fields

could be used to produce the mass eigenstates. Using matrix diagonalization techniques [17] the following eigenvalues are found,

$$m_{1,2} = \frac{1}{2} \left[(m_L + m_R) \pm \sqrt{(m_L - m_R)^2 + 4m_D^2} \right] \quad (1.21)$$

These mass eigenvalues now correspond to the real physical particle states, which are superpositions of the flavour eigenstates.

Considering that the left-handed Majorana field would require coupling to the Higgs triplet, $m_L = 0$ in the Standard Model. The right-handed Majorana field has no such limitations as it is a weak isospin singlet. The m_R term can therefore take any value. For the Seesaw mechanism (type 1) this parameter is chosen to be $m_R \gg m_D$ as this will have a notable effect on the neutrino mass states,

$$m_1 = \frac{m_D^2}{m_R}, \quad m_2 = m_R \left(1 + \frac{m_D^2}{m_R^2} \right) \quad (1.22)$$

The resulting neutrinos would have either a very small ($\frac{1}{m_R}$) or very large mass $\approx m_R$. The light neutrino would correspond to the superposition state with left-handed fields, and the heavy neutrino to the right-handed fields. If the Dirac mass term would be of a similar size 10^6 eV as the charged leptons and the term m_R would be of the order $10^{15} - 10^{16}$ eV, the resulting light neutrino mass would be in the 10^{-3} eV range. This would be consistent with current experimental limits of the neutrino mass [18].

1.6 Neutrino oscillations in vacuum

Neutrinos have been found to be capable of changing lepton flavour through repeated observations. As this occurs with a degree of periodicity this behaviour is said to be oscillating. It was first suggested by Pontecorvo in 1957 [19].

This phenomenon follows naturally from the misaligned flavour and mass states. When a neutrino participates in an interaction mediated by the weak force, it does so as a flavour eigenstate. This state does not fully correspond with the free mass states of the neutrino, therefore allowing the possibility of

flavour changing as the particle travels through space.

The mixing of the neutrino flavour and mass states can be defined as a unitary transformation between the orthonormal flavour and mass eigenbases,

$$|\nu_\alpha\rangle = \sum_i^3 U_{\alpha i}^* |\nu_i\rangle, \quad |\nu_i\rangle = \sum_\alpha^3 U_{\alpha i} |\nu_\alpha\rangle \quad (1.23)$$

where the flavour eigenstates (e, μ, τ) are represented by the Greek subscripts and the mass eigenstates are denoted by the Latin subscripts. The individual transformation terms can be found through projection of a flavour(mass) onto a mass(flavour) eigenstate,

$$U_{\alpha i}^* = \langle \nu_i | \nu_\alpha \rangle, \quad (1.24)$$

The transformation terms are elements of the Pontecorvo-Maki-Nagakawa-Sakata (PMNS) matrix. This matrix is typically rewritten in the form of the mixing angles,

$$\begin{aligned} U &= \begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu1} & U_{\mu2} & U_{\mu3} \\ U_{\tau1} & U_{\tau2} & U_{\tau3} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & 0 & s_{13}e^{-i\delta} \\ 0 & 1 & 0 \\ -s_{13}e^{i\delta} & 0 & c_{13} \end{pmatrix} \begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (1.25)$$

Additional phases such as Majorana phases have been left out of the PMNS matrix as these do not have an effect on the oscillations. c_{ij} and s_{ij} represent $\cos(\theta_{ij})$ and $\sin(\theta_{ij})$ respectively. The remaining phase in the PMNS matrix is the CP-violating phase (δ), which as the name suggests influences the amount of charge parity violation that can occur in neutrino oscillations.

To further examine neutrino oscillations it is helpful to take a closer look at the free neutrino travelling through empty space. It can be defined in terms of the mass eigenbasis, and so its time evolution is governed by the Schrödinger

equation,

$$H|\mathbf{v}_i(t)\rangle = i\partial_t|\mathbf{v}_i(t)\rangle = E|\mathbf{v}_i(t)\rangle \quad (1.26)$$

Here H is the unperturbed Hamiltonian for the free particle in vacuum. This equation has well known solutions in the form of complex plane waves,

$$|\mathbf{v}_i(t)\rangle = e^{-i(E_i t - \vec{p}_i \cdot \vec{x}_i)} |\mathbf{v}_i(0)\rangle \quad (1.27)$$

This can be further simplified through treating the neutrinos as ultra relativistic particles ($|\vec{p}_i| = p_i \gg m_i$). This assumption can be justified from experiment observations which place strong upper bounds on the neutrino mass [18]. In the ultra-relativistic limit ($E \gg m$), the energy term can be approximated using a Taylor expansion,

$$E_i = \sqrt{p_i^2 + m_i^2} \approx p_i + \frac{m_i^2}{2p_i} \approx E + \frac{m_i^2}{2E} \quad (1.28)$$

Using this approximation and the natural units which give $t \approx L$ the full simplification becomes,

$$|\mathbf{v}_i(t)\rangle = e^{-iL(E-p)} |\mathbf{v}_i(0)\rangle = e^{-\left(\frac{im^2L}{2E}\right)} |\mathbf{v}_i(0)\rangle \quad (1.29)$$

With this expression for the free mass eigenstates, the previously defined flavour eigenstates take the following form,

$$|\mathbf{v}_\alpha(L)\rangle = \sum_i^3 U_{\alpha i}^* e^{-\left(\frac{im^2L}{2E}\right)} |\mathbf{v}_i(0)\rangle \quad (1.30)$$

This is the form describing the evolution of the flavour eigenstates as the neutrino travels through vacuum. Taking the projection of the flavour state that has travelled from its original form and squaring it returns the transition

(or oscillation) probability,

$$\begin{aligned}
 P_{\alpha \rightarrow \beta} &= |\langle \nu_{\beta}(L) | \nu_{\alpha} \rangle|^2 = \left| \sum_i^3 U_{\alpha i}^* U_{\beta i} e^{-\left(\frac{im^2 L}{2E}\right)} \right|^2 \\
 &= \sum_i^3 \sum_j^3 U_{\alpha i} U_{\beta i}^* U_{\alpha j}^* U_{\beta j} e^{-\left(\frac{i\Delta m_{ij}^2 L}{2E}\right)}
 \end{aligned} \tag{1.31}$$

where the orthonormality of the mass eigenstates is used to reduce the bra-kets and the mass squared difference is defined as $\Delta m_{ij}^2 = m_i^2 - m_j^2$.

Expanding this expression results in the following equation for the oscillation probability,

$$\begin{aligned}
 P_{\alpha \rightarrow \beta} &= \delta_{\alpha\beta} - 4 \sum_{i>j} \Re(U_{\alpha i} U_{\beta i}^* U_{\alpha j}^* U_{\beta j}) \sin^2 \left(\frac{\Delta m_{ij}^2 L}{4E} \right) \\
 &\quad + 2 \sum_{i>j} \Im(U_{\alpha i} U_{\beta i}^* U_{\alpha j}^* U_{\beta j}) \sin \left(\frac{\Delta m_{ij}^2 L}{2E} \right)
 \end{aligned} \tag{1.32}$$

The oscillation probability is dependent on several parameters namely the distance travelled L , the squared mass difference Δm_{ij}^2 , the neutrino energy E and the specific values of the mixing angles θ_{ij} . If the argument inside the sine function in this expression becomes equal to an odd multiple of π the oscillation probability reaches its maximum value. From this the expression for the oscillation length can be found,

$$L_{osc} = \frac{2\pi E}{\Delta m_{ij}^2} \tag{1.33}$$

1.6.1 Muon neutrino disappearance

From the general expression for the neutrino oscillation probabilities two relevant channels for the NOvA experiment can be identified. These are the muon neutrino disappearance mode and the electron neutrino appearance mode. The first of these specifically looks for the amount of muon neutrinos which do not oscillate away to a different neutrino flavour $\nu_{\mu} \rightarrow \nu_{\mu}$. The oscillation

probability given in (1.32) can be simplified significantly upon substitution of $(\alpha, \beta) \rightarrow (\mu, \mu)$. A further approximation can be made when the relative size of Δm_{21}^2 compared to Δm_{31}^2 is taken into account ($\Delta m_{31}^2 \gg \Delta m_{21}^2$) allowing for $\Delta m_{31} \approx \Delta m_{32}$. Consequently the oscillation probability for this channel then takes the following form,

$$P_{\mu \rightarrow \mu} = 1 - \sin^2 2\theta_{23} \sin^2 \left(\frac{\Delta m_{31}^2 L}{4E} \right) + 4 \sin^2 \theta_{13} \sin^2 \theta_{23} \cos 2\theta_{23} \sin^2 \left(\frac{\Delta m_{31}^2 L}{4E} \right) \quad (1.34)$$

Using the difference in relative size of the mixing angles θ_{13} and θ_{23} where ($\theta_{23} \gg \theta_{13}$) a final reduction of the expression is possible,

$$P_{\mu \rightarrow \mu} = 1 - \sin^2 2\theta_{23} \sin^2 \left(\frac{\Delta m_{31}^2 L}{4E} \right) \quad (1.35)$$

The individual components of equation (1.35) determine the size and shape of the neutrino oscillation spectrum. The first minimum occurs in the survival probability when the argument of the second squared sine function becomes $\left\| \frac{\Delta m_{31}^2 L}{4E} \right\| = \frac{\pi}{2}$. Through the ratio of $\frac{L}{E}$ it is easy to see that Δm_{32}^2 relates to the oscillation frequencies. The amplitude of the oscillation peak, in this case a minimum, is determined by the size of $\sin^2 2\theta_{23}$ and thus the value of θ_{23} . This can be seen in Figure 1.6.

For neutrino experiments typically L is fixed and well known whereas E can be measured or specifically sampled depending on the experimental setup. This leaves Δm_{32}^2 and θ_{23} as free parameters which can be determined through a measurement of the oscillation spectrum. The sign of the mass splitting cannot be measured in this simplified case as the squared sine function is even. Both a positive and negative value of Δm_{32}^2 will yield the exact same oscillation spectrum. Recalling that the oscillation probabilities are calculated for free neutrinos travelling in vacuum, this degeneracy can be broken with the introduction of matter effects later in this chapter.

A measurement of the amount of CP violation (δ_{CP}) is not possible in this oscillation channel. The initial substitution in equation (1.32) results

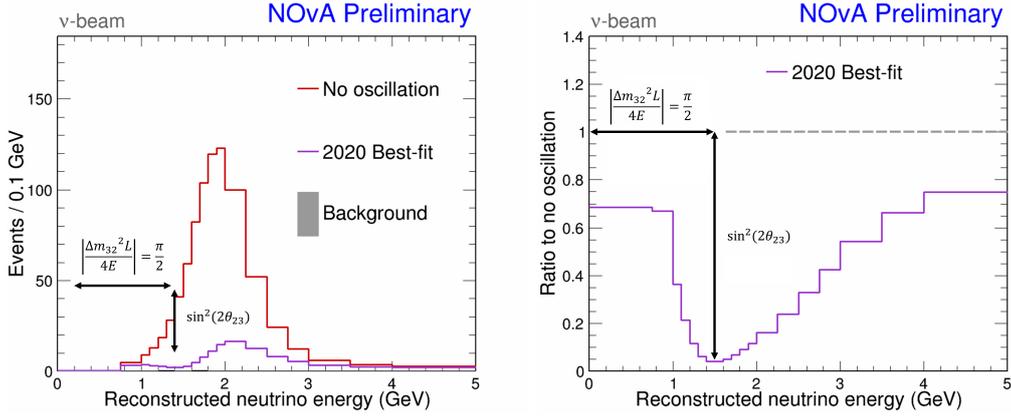


Figure 1.6: Neutrino oscillation spectra (left) and ratio (right) of the 2020 NOvA analysis for ν_μ disappearance. On the left both the unoscillated spectrum (red) and the oscillated spectrum (purple) at the NOvA FD are shown. On the right the ratio between the two is shown. The magnitude of the oscillation maximum is dependent on $\sin^2 \theta_{23}$ and the location is dependent on Δm_{32}^2 .

in the annihilation of the imaginary product terms of the PMNS matrix. The complex phase is part of these product terms. This conforms to the expectation under Lorentz invariance where CPT is a conserved symmetry. Neutrino oscillation under a CP transformation exchanges the neutrinos with their antiparticle counterparts, such as $(\nu_\alpha \rightarrow \nu_\beta) \xrightarrow{CP} (\bar{\nu}_\alpha \rightarrow \bar{\nu}_\beta)$. The T transformation results in a time reflection and exchanges the final and initial states, $(\nu_\alpha \rightarrow \nu_\beta) \xrightarrow{T} (\nu_\beta \rightarrow \nu_\alpha)$. Combination into the full CPT transformation for the survival probability where $(\alpha = \beta)$ yields $P(\nu_\alpha \rightarrow \nu_\alpha) = P(\bar{\nu}_\alpha \rightarrow \bar{\nu}_\alpha)$.

1.6.2 Electron neutrino appearance

The other relevant oscillation channel is the electron appearance. In this case muon neutrinos oscillate to electron neutrinos $\nu_\mu \rightarrow \nu_e$. As with the previous oscillation channel (α, β) can be substituted for (μ, e) . Applying a similar

approximation the resulting expression becomes,

$$\begin{aligned}
P_{\mu \rightarrow e} = & \sin^2 \theta_{23} \sin^2 2\theta_{13} \sin^2 \left(\frac{\Delta m_{31}^2 L}{4E} \right) \\
& + \cos^2 \theta_{13} \cos^2 \theta_{23} \sin^2 2\theta_{12} \sin^2 \left(\frac{\Delta m_{21}^2 L}{4E} \right) \\
& + \cos \theta_{13} \sin \theta_{13} \sin \theta_{23} \sin \left(\frac{\Delta m_{31}^2 L}{4E} \right) \sin \left(\frac{\Delta m_{21}^2 L}{4E} \right) \\
& \times \left(\cos \delta_{CP} \cos \left(\frac{\Delta m_{32}^2 L}{4E} \right) \mp \sin \delta_{CP} \sin \left(\frac{\Delta m_{32}^2 L}{4E} \right) \right)
\end{aligned} \tag{1.36}$$

This can be rewritten to a simpler form,

$$P_{\mu \rightarrow e} = P_{atm} + P_{sol} + 2\sqrt{P_{atm}P_{sol}} \left(\cos \delta_{CP} \cos \left(\frac{\Delta m_{32}^2 L}{4E} \right) \mp \sin \delta_{CP} \sin \left(\frac{\Delta m_{32}^2 L}{4E} \right) \right) \tag{1.37}$$

The components of this equation can be associated with atmospheric P_{atm} and solar P_{sol} neutrinos respectively.

$$P_{atm} = \sin^2 \theta_{23} \sin^2 2\theta_{13} \sin^2 \left(\frac{\Delta m_{31}^2 L}{4E} \right) \tag{1.38}$$

$$P_{sol} = \cos^2 \theta_{13} \cos^2 \theta_{23} \sin^2 2\theta_{12} \sin^2 \left(\frac{\Delta m_{21}^2 L}{4E} \right) \tag{1.39}$$

The oscillation parameters contained in equation (1.38) have been primarily experimentally determined by oscillation experiments measuring neutrinos produced in the Earth's atmosphere, likewise for (1.39) the neutrinos originate from processes occurring in the Sun. Given that $\Delta m_{31}^2 \gg \Delta m_{21}^2$ the atmospheric terms form the leading terms in the oscillation probability for this channel. The last term in equation (1.37) is the interference term [20]. The destructive (−) part of this term is for neutrinos whereas the constructive (+) part is for antineutrinos.

1.7 Neutrino oscillations in matter

The previous treatment of neutrino oscillations is only valid for the case of free neutrinos travelling through a vacuum. A modification is required to extend the validity of these solutions for neutrinos travelling through matter. This effect is referred to as the matter effect, or the Mikheyev-Smirnov-Wolfenstein effect [21]. As the neutrino travels through matter, multiple coherent forward scattering interactions are possible. A subset of these are the neutral current weak interactions which can occur with any of the neutrino flavours and are equally likely. They therefore affect all neutrinos flavour states equally and thus have no net effect on the oscillation probabilities. This does not hold for the charged current forward scattering interactions. Due to the presence of a single flavour of leptons in the material, only the corresponding neutrino flavour can participate in the CC interaction. The relevant lepton and neutrino flavour here is the electron. Figure 1.7 shows the Feynman diagrams for these CC and NC interactions.

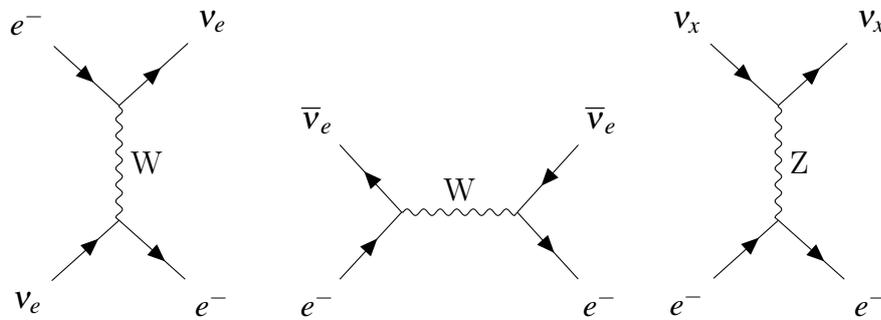


Figure 1.7: Feynman diagrams of coherent forward scattering on electrons.

These additional interaction modes which are only available to the electron neutrino flavour state result in an effective potential for the electron flavour component of the neutrino mass eigenstates. The consequence is a change to the neutrino's effective mass and by extension a change to its time evolution. This effective potential can be used to extend the Hamiltonian seen in equation

(1.26),

$$H = U \begin{pmatrix} \frac{m_1^2}{2E} & 0 & 0 \\ 0 & \frac{m_2^2}{2E} & 0 \\ 0 & 0 & \frac{m_3^2}{2E} \end{pmatrix} U^\dagger + \begin{pmatrix} \pm\sqrt{2}G_F N_e & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (1.40)$$

where the potential term is composed of the Fermi constant G_F and N_e is the number density of electrons in the material. Neutrinos (+) and antineutrinos (−) are distinguished by the sign of the potential term.

Repeating the process of diagonalisation with the Hamiltonian, new expressions for the mass states of neutrinos can be found. The differences due to the matter effects can be summarized as a set of modifications [22] to the oscillation probabilities in vacuum (1.32),

$$\sin\left(\frac{\Delta m_{ij}^2 L}{4E}\right) \xrightarrow{\text{matter effects}} \frac{\left(\frac{\Delta m_{ij}^2 L}{4E}\right)}{\left(\frac{\Delta m_{ij}^2 L}{4E} \mp \frac{G_F N_e L}{\sqrt{2}}\right)} \sin\left(\frac{\Delta m_{ij}^2 L}{4E} \pm \frac{G_F N_e L}{\sqrt{2}}\right) \quad (1.41)$$

The magnitude of the matter effects on the individual oscillation probabilities varies based on the specific values of the parameters described in equation (1.32). It also affects the oscillation probabilities for neutrinos and antineutrinos oppositely due to the sign difference in equation (1.41). Based on the sign of the mass squared difference the oscillation probability for muon to electron (anti)neutrinos $P(\nu_\mu \rightarrow \nu_e)$ is enhanced(suppressed). If the sign is negative the enhancement and suppression flips for the neutrinos and antineutrinos respectively.

1.7.1 Mass hierarchy

Measurements of the oscillation probabilities allow for the determination of the mass squared differences. From atmospheric neutrino oscillation measurements the value of Δm_{31} has been determined, whereas the value of Δm_{21} has been determined from solar neutrino measurements. Other experiments have been able to place limits on the magnitudes of the individual mass terms. However two possible orderings, or hierarchies, of the mass states remain possible. The

first hierarchy is named the Normal Ordering (NO) where the third mass state is placed above the other two and is thus considered to be the heaviest. The second is the Inverted Ordering (IO) where the third mass term would be the lightest and be placed below the other two state. Both can be seen in Figure 1.8. This figure also illustrates the flavour contents of the 3 neutrino mass states.

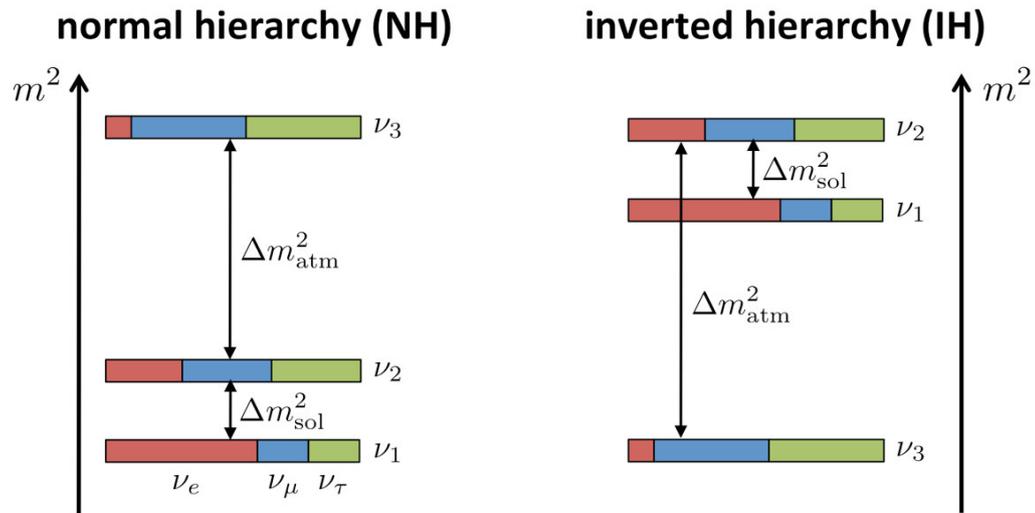


Figure 1.8: The two possible orderings (hierarchies) of the neutrino mass states. The normal ordering is shown on the left, the inverted ordering on the right. Courtesy of [23].

Chapter 2

Experimental status of neutrino oscillation measurements

In section 1.1 a very brief overview was given of the history of neutrino discovery measurements. This history is marked by the achievements of numerous experiments and the physicists, engineers and supporting staff behind them. While a full walk-through of this rich history would certainly not be a waste of time, it goes beyond the scope of this thesis. Instead a more comprehensive examination of the current experimental status of the neutrino oscillation measurements and experiments will be presented here. Neutrino oscillation experiments can be split into several different sectors namely the solar, atmospheric, reactor and accelerator sectors. The distinctions between these sectors are based on the source of neutrinos, their respective energy ranges and their baseline lengths. Experiments in the different sectors are typically sensitive to only some of the oscillation parameters due to the unique operational conditions which form these distinctions.

2.1 Solar neutrinos

The fusion reactions which power the Sun effectively turn it into a neutrino source. The energy ranges associated with neutrinos produced in these reactions is of the order MeV. The Standard Solar Model (SSM) [24] describes the processes and reactions which are capable of contributing to the overall neutrino

flux produced by the Sun. From Figure 2.1 it is easy to see that not all reactions contribute equally to the overall flux. Furthermore the shapes of the spectra indicate that neutrino energy sensitivity is the key factor in determining how much of the total neutrino flux an experiment can measure and which specific reactions can be probed via a measurement of the solar neutrino flux. There are two chains of reactions which have constituent reactions capable of producing neutrinos. Both of these convert hydrogen to helium and are known as the proton-proton chain reaction (pp chain) and the carbon-nitrogen-oxygen cycle (CNO). In Figure 2.1 contributions to the total flux are listed by their source reaction. For example the neutrino originating from a ${}^8\text{B}$ undergoing beta decay would be referenced as a ${}^8\text{B}$ neutrino.

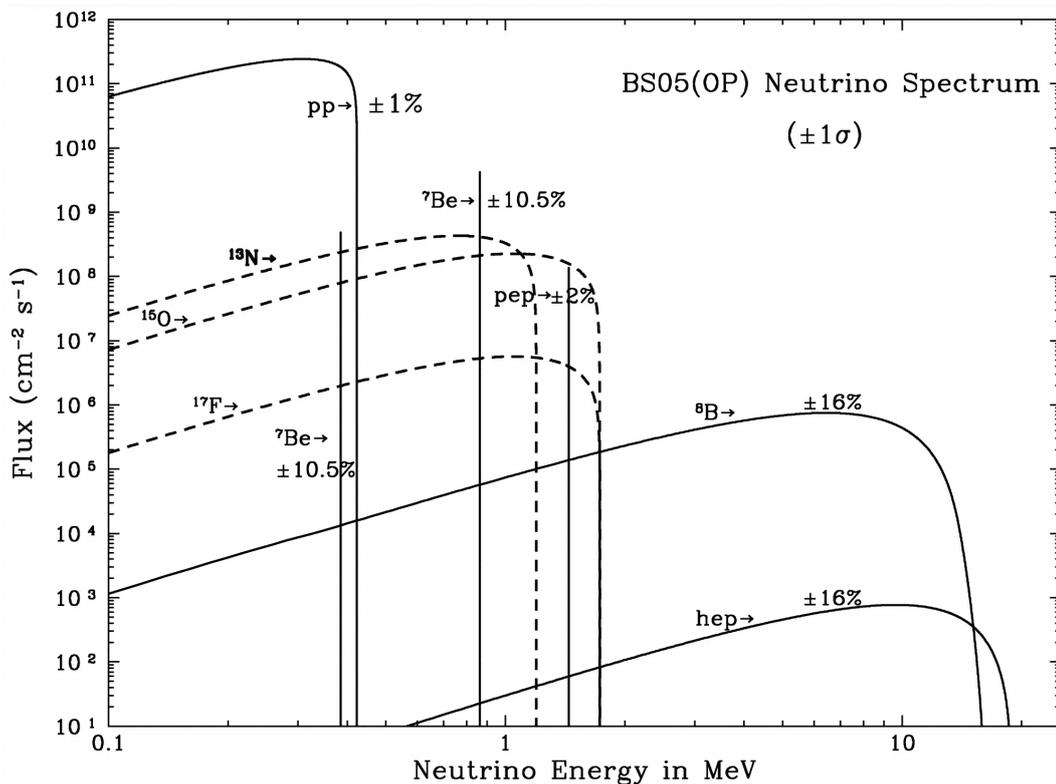


Figure 2.1: The solar neutrino flux decomposed by individual contribution of the various reactions in the SSM as a function of neutrino energy. This plot was taken from [24] and employs a radiative opacity correction to the SSM. The dashed (solid) lines indicates contributions from reactions in the CNO cycle (pp chain).

2.1.1 Solar neutrino experiments

The Brookhaven Solar Neutrino Experiment [25] was the pioneering measurement of the solar neutrino flux. Located in Lead, South Dakota at the Homestake Gold Mine it is therefore colloquially referred to as the Homestake experiment. The experiment consisted of a 380 cubic meter tank filled with perchloroethylene placed at considerable depth (1478 m). Placing experiments underground is commonly done when there is a depth-reducible background present at the surface. In the case of the Homestake experiment the cosmic ray background required this form of shielding. Neutrinos present in the solar neutrino flux could penetrate to this depth and upon reaching the chlorine 37 atoms present in the perchloroethylene they could interact. If the neutrinos energy is above 0.814 MeV it can be captured by the chlorine and transform the chlorine into argon 37. The neutrino capture interaction is given by,



Collection of the produced amount of argon in this interaction resulted in the neutrino flux measurement.

This measurement formed the basis of the solar neutrino problem as the measured flux was only a third of its predicted size. The final measurement of the neutrino flux was $2.56 \pm 0.16 \pm 0.16 \cdot 10^{-36} \frac{\text{capture}}{\text{s} \cdot \text{atom}}$ [26] whereas the theoretical prediction is $8.46_{-0.88}^{+0.87} \cdot 10^{-36} \frac{\text{capture}}{\text{s} \cdot \text{atom}}$ [27]. While unknown at the time, neutrino oscillations account for this difference.

This experiment was unable to confirm major segments of the SSM. This was primarily due to the experiment only being sensitive to a single flavour of neutrinos, the electron neutrinos, and its inability to measure the neutrino flux components below the 0.814 MeV threshold.

Several other experiments operating on this radiochemical principle have been executed to further study the solar neutrino flux. Lowering the threshold of the neutrino capture would allow for the probing of the high pp flux which has one of the lower uncertainties. When chlorine is substituted for gallium

the threshold is lowered to 0.233 MeV,



In this reaction the gallium is transformed into germanium. The Soviet-American Gallium Experiment (SAGE) used approximately 50-57 tons of liquid gallium located 2100 meters underground at the Baksan Neutrino Observatory in Russia. The Gallium Experiment also known as GALLEX uses 101 tons of GaCl_3 which is equivalent to 30.3 tons of gallium. After an interruption GALLEX continued under a different name, Gallium Neutrino Observatory (GNO). It is located at the Laboratori Nazionali del Gran Sasso in Italy at a rock depth equivalent of 3200 meters of water. The final measurements for these experiments came to $65.4^{+3.1+2.6}_{-3.0-2.8} \cdot 10^{-36} \frac{\text{capture}}{\text{s-atom}}$ for SAGE [28] and $69.3 \pm 4.1 \pm 3.6 \cdot 10^{-36} \frac{\text{capture}}{\text{s-atom}}$ for GALLEX+GNO [29]. The SSM prediction for these experiments is $127.9^{+8.1}_{-8.2} \cdot 10^{-36} \frac{\text{capture}}{\text{s-atom}}$ [27].

Experiments relying on radiochemical reactions run for a period of time before the transformed elements are collected and analysed. An alternative mechanism to measure the solar neutrino flux is possible. This was demonstrated in the Kamiokande experiment. It consisted of a 3 kt water-Cherenkov detector located in the Mozumi mine of the Kamioka Mining and Smelting Co. LTD. in the Gifu Prefecture of Japan. Approximately a thousand 50 cm PhotoMultiplier Tubes (PMT) lined the inner wall of the tank. The original purpose of the experiment was the investigation of proton decays. At that time it was called the Kamioka Nucleon Decay Experiment (KamiokaNDE). The threshold required for this measurement was well above that required for solar neutrino measurements. Modifications were made which reduced the background and thus lowered the threshold to within the range required for solar neutrino measurements. As a neutrino from the Sun scatters on an electron present in the active volume of the detector the recoil of the electron

produces Cherenkov light,

$$\begin{aligned} \nu_x + e^- &\rightarrow \nu_x + e^- \quad (\text{NC}) \\ \nu_e + e^- &\rightarrow \nu_e + e^- \quad (\text{CC}) \end{aligned} \tag{2.3}$$

This light propagates through the water and is captured by the PMT's located on the detector walls. As this is an elastic scattering event measurements of the neutrino energy spectrum are possible. Furthermore measurement of the incoming neutrino direction is possible through the directional correlation between the incident neutrino and recoiling electron. The Cherenkov light is produced by the recoiling electron in a cone along the direction it is travelling in. This provides a very distinctive experimental signature which can be used to discriminate between signal and background events. The scattering occurs via both charged current and neutral current interactions which means that this measurement of this interaction is sensitive to all neutrino flavours. The charged current interaction can only occur with the electron neutrinos which means that the cross-section for the electron neutrinos is larger than that of the other flavours in this experiment. A larger version of this experiment was made, called Super-Kamiokande (SK). It operates on the same principle but has a much larger volume of water, approximately 50 kt. 32 kt of that being active volume lined by 11 thousand PMT's. Both experiments found smaller numbers of observed solar neutrino events compared to the expected number from the SSM. For example the ${}^8\text{B}$ neutrino flux, to which both experiments are sensitive, was measured to be $2.345 \pm 0.014 \pm 0.036 \cdot 10^6 \frac{1}{\text{cm}^2\text{s}}$ [30] whereas $5.46 \pm 0.66 \cdot 10^6 \frac{1}{\text{cm}^2\text{s}}$ [31] is predicted by the SSM.

Another experiment employing a similar Cherenkov based measurement method was the Sudbury Neutrino Observatory (SNO). SNO is located 2100 meters underground at Vale's Creighton Mine in Sudbury, Canada. It consisted of a 1000 ton spherical detector contained in a submerged mine cavity. The detector was submerged in water aiding the structural integrity of the detector when filled as well as providing an additional method of radiation

shielding. The detector wall is lined with 9600 PMT's encapsulating the active region containing 1000 tons of deuterium oxide (heavy water, D_2O) plus a small proportion of regular water. The acrylic vessel containing the heavy water has a radius of 6 meters, whereas the PMT's are located at a radius of 8.5 meters. The choice of heavy water as the active material facilitates interactions of different types. The first is the elastic electron scattering described earlier in equation (2.3). The second is the electron neutrino charged current interaction,

$$\nu_e + d \rightarrow e^- + p + p \quad (2.4)$$

The third is the neutral current interaction,

$$\nu_x + d \rightarrow \nu_x + n + p \quad (2.5)$$

In the neutral current interaction all flavours of neutrinos participate equally. In the elastic electron scattering all neutrino flavours can participate, but they do not do so equally. In the charged current interaction only the electron neutrinos can participate. Solar neutrinos are too low in energy for the other lepton flavour CC interactions. Combining these leads to a measurement of both the ν_e and non ν_e components of the solar neutrino flux. SNO is sensitive to the 8B neutrinos and through the NC channel measured their flux to be $5.25 \pm 0.16_{-0.13}^{+0.11} \cdot 10^6 \frac{1}{\text{cm}^2\text{s}}$ [32] which is fully consistent with the SSM predicted value.

To further study the neutrino flux that falls below the energy threshold of SNO and SK, Borexino was constructed. It consists of approximately 300 tonnes of liquid scintillator in a radio-pure environment. It is located deep underground at the Gran Sasso National Laboratory. Due to the high spatial and energy resolution of the detector as well as the low background environment its energy threshold is 0.19 MeV [33]. This has allowed for measurements of the 7B , 8B , pep , pp and CNO neutrino fluxes.

An overview of the experiments contributing to the measurements of the

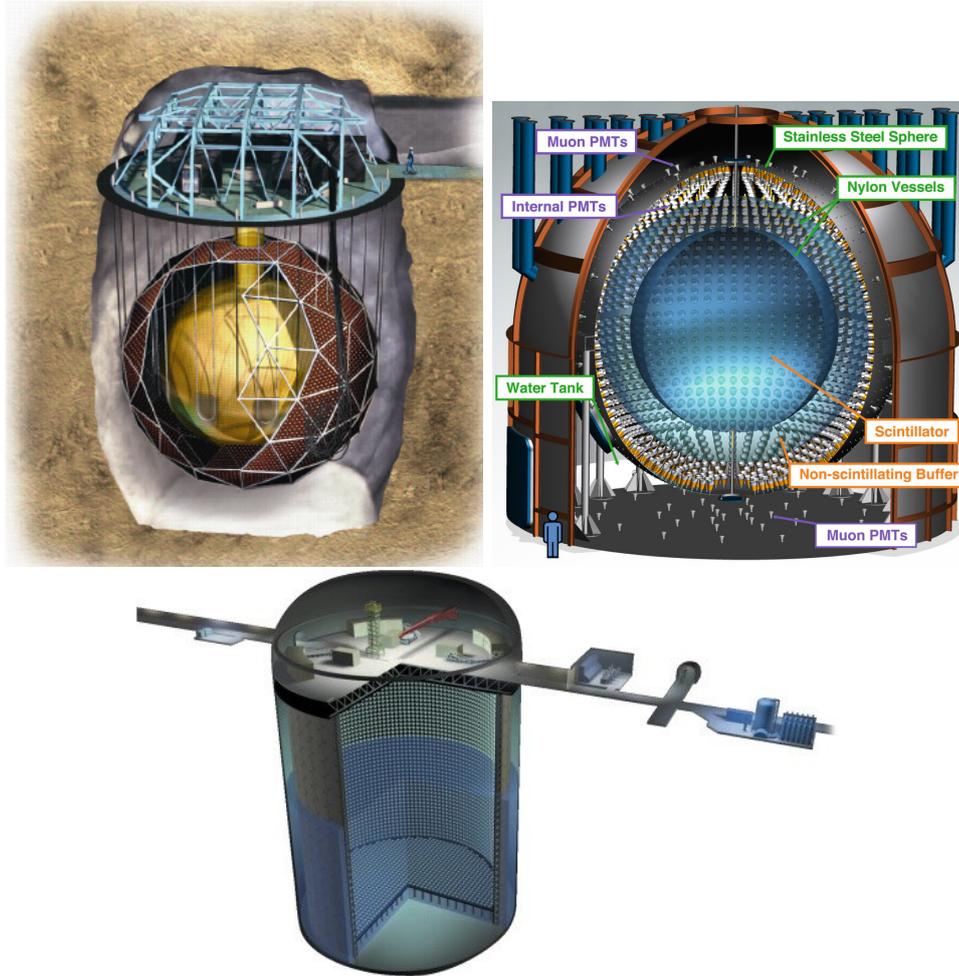


Figure 2.2: Schematic overview of several solar neutrino experiments. The SNO+ detector is shown on the left, the Borexino detector is shown on the right and the partially filled Super-Kamiokande detector is shown on the bottom. Images courtesy of [34], [35], [36].

solar neutrino flux are given in table 2.1. A collection of detectors employed in a subset of these experiments is shown in Figure 2.2.

2.2 Atmospheric neutrinos

The Sun is not the only source of neutrino flux measurable on Earth. When high energy cosmic nucleons impact the Earth's atmosphere, they interact with the particles already present. These interactions lead to the production of mesons whose subsequent decays produce neutrinos. This process is shown in Figure 2.3. As the incoming nucleons have a broad energy spectrum there are many different paths of interaction chains which can contribute to the overall

Name	Target material	Energy threshold (MeV)	Mass($tonnes$)	Years
Homestake	C_2Cl_4	0.814	615	1970 – 1994
SAGE	Ga	0.233	50	1989–
GALLEX	$GaCl_3$	0.233	100 [30.3 for Ga]	1991 – 1997
GNO	$GaCl_3$	0.233	100 [30.3 for Ga]	1998 – 2003
Kamiokande	H_2O	6.5	3,000	1987 – 1995
Super-Kamiokande	H_2O	3.5	50,000	1996–
SNO	D_2O	3.5	1,000	1999 – 2006
KamLAND	Liquid scintillator	0.5/5.5	1,000	2001–
Borexino	Liquid scintillator	0.19	300	2007–

Table 2.1: Overview of solar neutrino experiments. Recreated from [37].

atmospheric neutrino flux. Additionally due to the large mean free path length of the neutrinos it is possible for neutrinos to travel all the way through the Earth before interacting. This path length ranges from approximately 10 km straight above the measurement point (cosine of the zenith angle equal to 1) all the way to the opposite side of the Earth with a path length of approximately 10^4 km (cosine of the zenith angle equal to -1). This results in a total neutrino flux having an angular dependence on the path length. Given the variation in the energy spectrum and the number of interaction modes the total neutrino flux contains all possible flavours of neutrinos and antineutrinos. The primary mode is the charged pion decay chain. This chain occurs when a charged pion is produced in the atmosphere and decays to a muon and a muon neutrino. Depending on the interaction location and energy of the muon it can further decay into a positron, electron neutrino and muon antineutrino before hitting the ground,

$$\begin{aligned}
 \pi^+ &\rightarrow \mu^+ + \nu_\mu, & \mu^+ &\rightarrow e^+ + \nu_e + \bar{\nu}_\mu \\
 \pi^- &\rightarrow \mu^- + \bar{\nu}_\mu, & \mu^- &\rightarrow e^- + \bar{\nu}_e + \nu_\mu
 \end{aligned}
 \tag{2.6}$$

The charge conjugated chain is possible and has been shown in equation (2.6) in parenthesis.

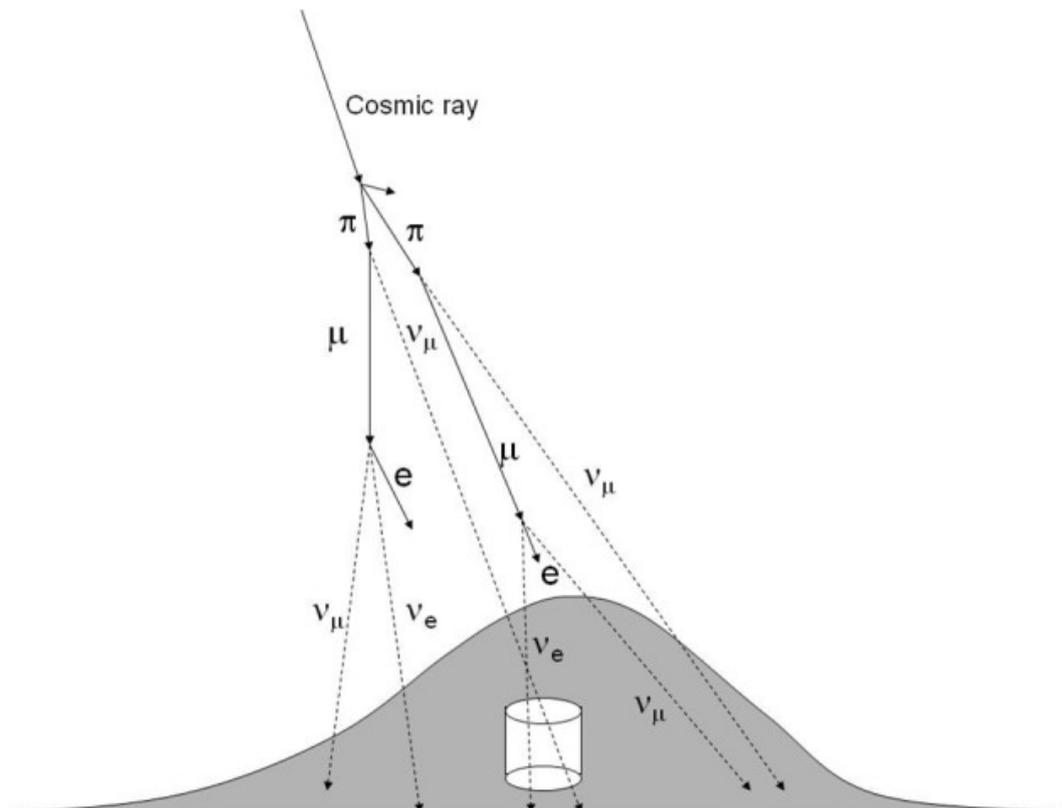


Figure 2.3: Neutrino production originating from cosmic-ray interactions with air nuclei in the Earth's atmosphere. Courtesy of [38].

Theoretical efforts to calculate the atmospheric neutrino flux have focused on modelling of the primary cosmic rays and particle compositions and their subsequent interactions in the atmosphere. Uncertainty on the total flux is approximately 15% whereas ratios of the different flux components allow for much lower uncertainties of approximately 5% due to cancellations in the errors [39].

2.2.1 Atmospheric neutrino experiments

In the 1960's two experiments detected atmospheric neutrinos. There is some contention as to which of these two experiments was first, the Case-Witwatersrand-Irvine (CWI) liquid scintillator detector used in the East Rand Proprietary Mines (ERPM) experiment [40] detected the neutrinos two months before the collaboration (Bombay-Osaka-Durham) at the Kolar Gold Fields (KGF) [41]. The KGF detector consisted of plastic scintillator planes, neon

flash tubes, photomultipliers and lead sheets using coincidence triggers to make the measurements. The KGF experiment published their results two weeks before the ERPM experiment and thus the first detection has been attributed to them. The KGF experiment is located in India whereas the ERPM experiment occurred in South Africa.

Several subsequent experiments were capable of detecting atmospheric neutrinos. Measurements of atmospheric neutrino flavours were made by these experiments through detection of flavour-specific nucleon decays. Kamiokande and the Irvine-Michigan-Brookhaven (IMB) experiment employed water-Cherenkov detectors. Iron tracking calorimeters were also used for this purpose, for example by the NUSEX and Frejus experiments. The flavour discerning capabilities of both of these detector types were limited to the electron and muon flavours. Similarly to the theoretical treatment of the atmospheric neutrino flux measuring the ratios between the different flavours greatly reduces uncertainty on the measurement. This ratio is defined as the total flux of muon neutrinos and antineutrinos over the total flux of electron neutrinos and antineutrinos. The measured ratios of both NUSEX [42] and Frejus [43] were in agreement with the theoretical predictions of the atmospheric neutrino flux without neutrino oscillations.

The Kamiokande [44] and IMB [45] experimental measurements of these same ratios discovered a discrepancy. Significantly fewer muon neutrinos and antineutrinos were found. Further analysis of this measurement by the Kamiokande experiment with additional data found a direct correlation of muon disappearance probability with the distribution of zenith angle (corresponding to differing path lengths). The Kamiokande measurements were statistics limited. Its follow-up experiment Super-Kamiokande would provide the resolution. Super-Kamiokande made a comparative measurement of the upward- (long path length) and downward- (short path length) going atmospheric neutrino distributions. A significant deficit of upward going muon neutrinos was measured [46]. This deficit can be seen in Figure 2.4. A similar

measurement of the electron neutrinos showed no excess nor deficit. Further analysis of the Super-Kamiokande data would result in a measurement of the muon neutrino survival probability. This measurement was consistent with the theoretical prediction including neutrino oscillations, specifically $\nu_\mu \rightarrow \nu_\tau$ oscillations. Super-Kamiokande has shown evidence for the occurrence of these oscillations with a charged current cross section measurement of the ν_τ neutrino [47], which is shown in Figure 2.5.

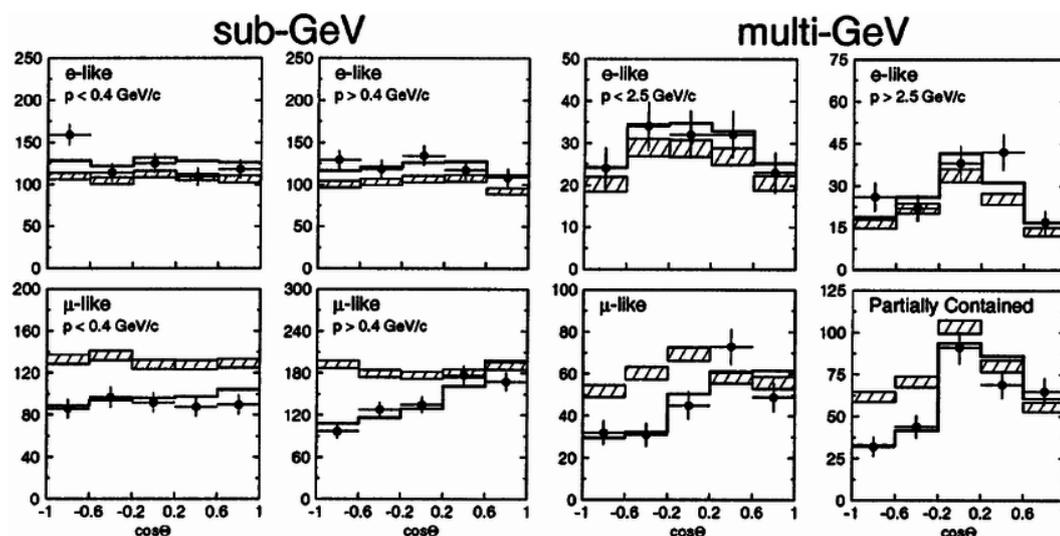


Figure 2.4: SK measurement of upwards ($\cos \theta < 0$) and downwards ($\cos \theta > 0$) going neutrino events for various energy ranges. The hatched region displays the number of expected events without oscillations, the black line shows the best fit for $\nu_\mu \leftrightarrow \nu_\tau$ oscillations. Courtesy of [46].

Other experiments have confirmed Super-Kamiokande's ratio measurements of neutrino oscillation such as the MACRO and Soudan2 experiments. Additionally ν_τ appearance has been independently observed and measured by the OPERA and IceCube experiments. The IceCube experiment consists of a large number of PMT's distributed across a cubic kilometre of Antarctic ice. Similarly to IceCube other neutrino telescopes are sensitive to atmospheric neutrino oscillations and thus can be used to produce additional measurements.

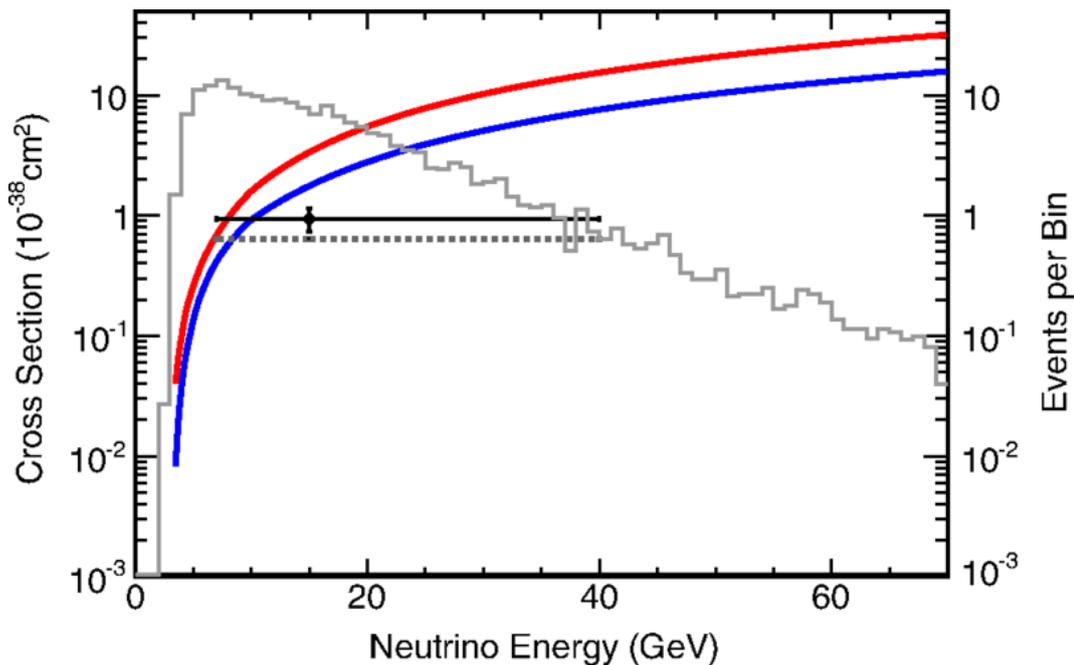


Figure 2.5: SK measurement of flux-averaged charged-current tau neutrino cross section (black) shown with the theoretical differential cross sections (red, blue for ν_τ and $\bar{\nu}_\tau$ respectively) and flux averaged cross section (dashed gray). Courtesy of [47].

2.3 Reactor neutrinos

Conventional fission nuclear reactors produce a large number of electron antineutrinos. The principal isotopes undergoing fission are ^{235}U , ^{238}U , ^{239}Pu and ^{241}Pu . Most of their daughter nuclei possess a surplus of neutrons. Any number of these neutrons can undergo beta decay and transform into a proton, electron and electron antineutrino. The proton and electron remain within the core of the nuclear reactor while the antineutrino escapes due to the low interaction cross-section with the surrounding matter. A theoretical flux calculation is difficult due to the numerous decay channels which all contribute to the overall flux as well as the unique conditions specific to each nuclear reactor, such as the precise ratio of the various fissile isotopes. The current methodology relies on separate measurements of the beta decay spectra of all the possible decay products [48] [49] [50]. Combining this data within a model the total antineutrino flux can be constructed.

This method of calculating the total antineutrino flux leaves a relative

uncertainty in the range of 10 – 20% mainly due to missing data in the nuclear databases from which the beta spectra are pulled and other systematic effects. More recent calculations have combined information from additional nuclear databases and used higher order corrections to improve upon the previous predictions. This resulted in similarly shaped spectra and uncertainties however the overall normalization factor was found to be 3% higher for reanalysis of data recorded at the Institut Laue-Langevin [51].

2.3.1 Reactor neutrino experiments

The energy range of the reactor antineutrino spectrum is typically limited to 10 MeV [52]. This energy range is therefore below that which is required to perform direct charged current measurements of the $\bar{\nu}_e \rightarrow \bar{\nu}_\mu$ (approx. 107 MeV) or $\bar{\nu}_e \rightarrow \bar{\nu}_\tau$ (approx. 3.5 GeV) oscillation channels. This leaves $\bar{\nu}_e \rightarrow \bar{\nu}_e$ neutrino disappearance as the only possible neutrino oscillation channel which can be measured from a nuclear reactor source. The inverse beta decay,

$$\bar{\nu}_e + p \rightarrow n + e^+ \tag{2.7}$$

has a threshold of approximately 1.81 MeV and is thus suitable for measuring the reactor antineutrino flux. The experimental signature of this inverse beta decay is unique as after the antineutrino is captured the positron can be identified and after a delay the subsequent neutron is captured which releases a distinct gamma ray. The design of detectors specifically built for antineutrino measurements therefore consists of a detector volume filled with protons and scintillator. Additionally, to increase the efficiency of neutron captures, doping of the scintillator with gadolinium is possible. Gadolinium has a high capture cross-section for neutrons and the associated gamma ray emission has a higher energy than neutron capture on hydrogen.

The KamLAND experiment deployed a 1,000 tonne pure liquid scintillator detector in the Kamiokande mine. The energy spectrum shape of the observed electron antineutrinos is consistent with electron antineutrino disappearance

through neutrino oscillations [53] and inconsistent with decay and decoherence models [54]. This is shown in Figure 2.6.

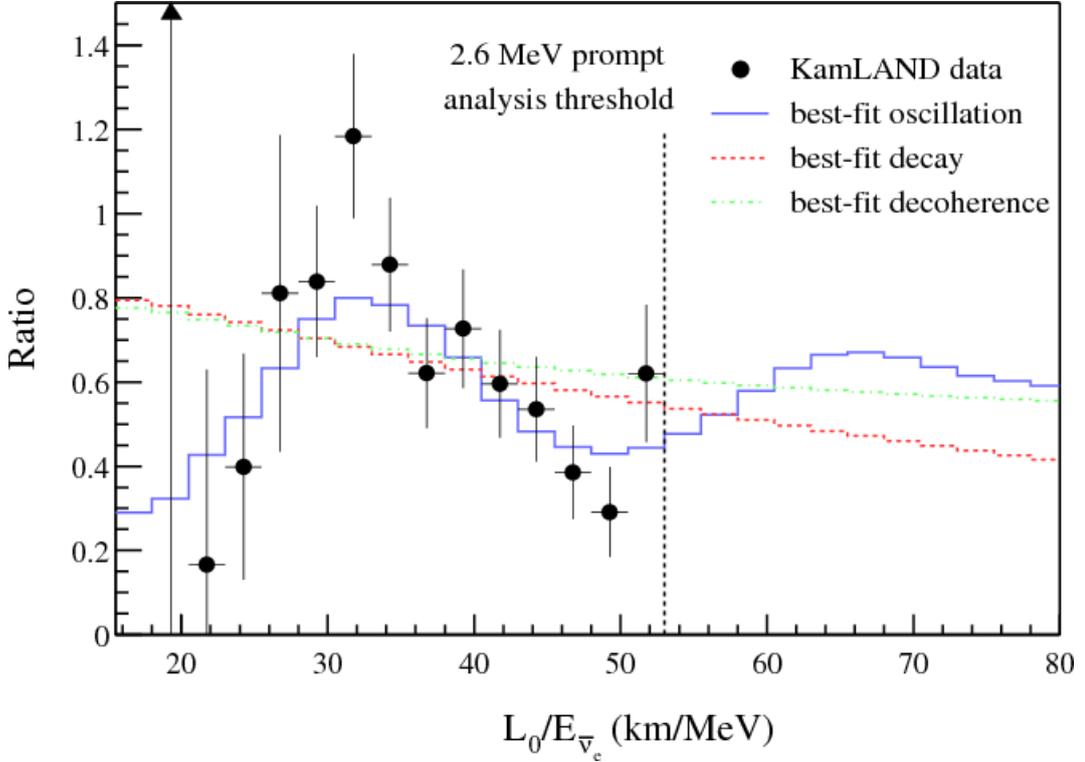


Figure 2.6: Ratio of observed $\bar{\nu}_e$ spectrum to the expectation for no oscillations as a function of L_0/E in the KamLAND experiment. The data is plotted with $L_0 = 180$ km, as if exposed by a single reactor at this distance. Courtesy of [54].

To enable a more precise measurement of the mixing angle associated with the electron antineutrino disappearance channel three experiments were set up. The detector design of these three experiments is very similar. They consist of a clear acrylic cylindrical tank filled with gadolinium-doped scintillator which serves as the target. This is surrounded by liquid scintillator to contain the produced gamma rays from the neutron captures. This is surrounded by clear mineral oil which allows light to pass through into the PMT's which line the outer steel wall. To reduce backgrounds this outer steel vessel is submerged in liquid scintillator or water which when instrumented allows it to be used as a veto. The Double Chooz experiment used liquid scintillator whereas Daya Bay and RENO used water. Additionally these experiments used two

or more detectors placed with different path lengths to the reactor source(s). This allows for the measurement of the antineutrino flux at different baselines. Comparative analysis can greatly reduce systematic uncertainties present in the overall measurement.

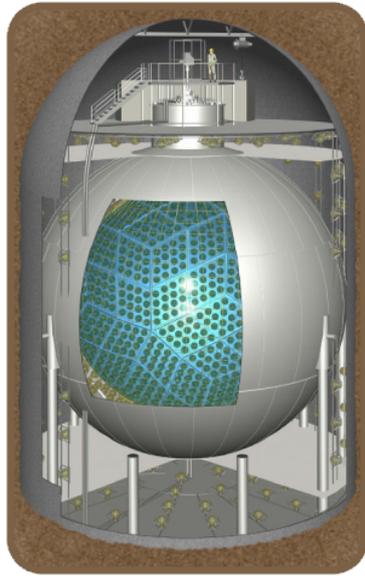
In 2012 these three experiments published their first measurements, independently of each other. The measured antineutrino spectra of each of the experiments showed the characteristic shape and size associated with electron antineutrino disappearance consistent with neutrino oscillations. An overview of relevant information regarding these experiments and the KamLAND experiment is given in table 2.2 and schematic representations of the various detectors is shown in Figure 2.7.

Name	Reactor power (GW_{th})	Baseline (km)	Detector mass ($tonnes$)	Years
KamLAND	Variable	avg.180	1,000	2001–
Double Chooz	4.25×2	1.05	8.3	2011 – 2018
Daya Bay	2.9×6	1.65	20×4	2011 – 2020
RENO	2.8×6	1.38	16	2011–

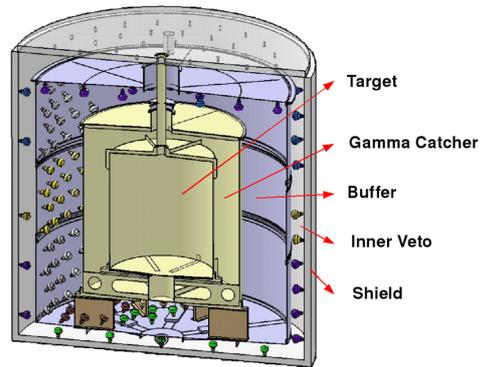
Table 2.2: Overview of reactor neutrino experiments. Recreated from [37].

2.4 Accelerator neutrinos

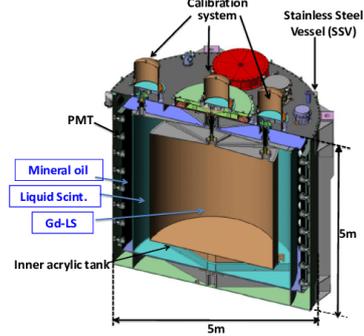
Besides neutrinos as by-products of interactions occurring in the Sun, atmosphere or in nuclear reactors other more intentional sources have been created. This is the case for neutrino beams. Typically protons are accelerated to high energies before being allowed to collide with a target. These collisions produce primarily pions and kaons. After collision with the target, magnetic fields are used to select pions and kaons with either positive or negative charge. This selection is done through the orientation of the direction of the magnetic fields, focussing one type and deflecting the other. A set of magnetic horns are employed to generate and control these magnetic fields. The produced particles then enter into a long decay pipe/tunnel. The length of this tunnel is optimized to produce a beam with high purity. Additional decays of the muons produced



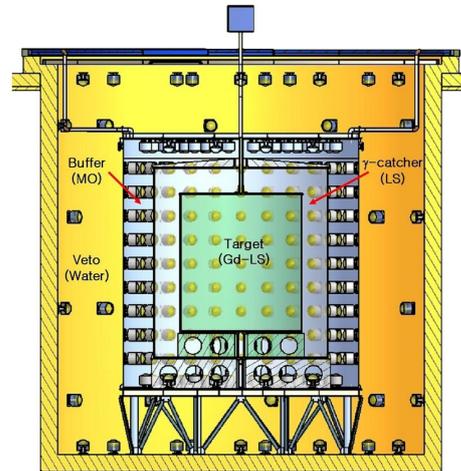
(a) The KamLAND detector.
Adapted from [55]



(b) The Double Chooz detector.
Courtesy of [56].



(c) The Daya Bay detector.
Courtesy of [57].



(d) The RENO detector.
Adapted from [58]

Figure 2.7: Schematic overview of several reactor neutrino experiments.

in the pion decays form a source of wrong sign neutrino contamination. The kaon has multiple decay paths some of which ($\approx 5\%$) yield contamination as well. Finally any remaining non-neutrino particles in the beam are stopped in material.

A more detailed description of the neutrino beam used in the NOvA experiment is given in section 3.1.

2.4.1 Long baseline experiments

The oscillation probability depends on the ratio of the path length and the neutrino energy. As neutrinos are produced in a beam with a specific range of energies the placement of any detectors is the determining factor for the measurement of neutrino oscillations. For energies in the GeV range baselines of hundreds up to a thousand kilometers are required. This category of experiments are referred to as long-baseline experiments. Typically a near-far detector setup is used. This is possible due to the neutrino source being a beam. The near detector can be used to make detailed measurements of the neutrino flux and cross-sections. Furthermore predictions for the data collected at the far detector can be made using the measurement from the near detector.

The first long baseline experiment (K2K) used a multi-detector setup placed in the neutrino beam produced at the 12 GeV proton synchrotron located at the KEK laboratory. The near detector consisted primarily of a 1 kt water-Cherenkov detector. This was located ≈ 300 m from the end of the beam target. The design of the near detector was chosen to be a scaled down version of the water-Cherenkov detector at Kamioka Observatory. This experiment made the first measurement of neutrino oscillation in a neutrino beam environment [59]. It confirmed the measurements of atmospheric neutrinos by Super-Kamiokande.

The Main Injector Neutrino Oscillation Search (MINOS) consisted of a near and far detector. The far detector was located in the Soudan mine at a distance of 735 km from Fermilab. The neutrino beam is generated from 120 GeV protons coming from the Main Injector. The design of the near and far detectors is similar, both of them being magnetised iron scintillator tracking

calorimeters. The mass of the near detector is 0.98 kt and the mass of the far detector 5.4 kt . After changes were made to the positioning of the beam target and horns the energy spectrum of the measured neutrinos changed. MINOS+ is the same experiment running in the higher energy mode of the beam. Analysis of the MINOS and MINOS+ data has led to precision measurements for muon neutrino disappearance and electron neutrino appearance [60] in addition to placing constraints on the potential values the parameters of three flavour neutrino oscillations can have [61].

The Super Proton Synchrotron particle accelerator at CERN was also used for the purpose of creating a neutrino beam. This beamline is referred to as the CERN Neutrinos to Gran Sasso (CNGS). It sourced high energy protons from the Super Proton Synchrotron (SPS). With a baseline of 732 km and an average energy of 17 GeV tau neutrino can directly observed. The Oscillation Project with Emulsion-tRacking Apparatus (OPERA) located at Gran Sasso detected 5 tau neutrinos originating from the muon neutrino beam [62]. This experiment consisted of nuclear emulsion bricks layered with lead sheets. These bricks were stacked into parallel walls with plastic scintillator counters layered in between them.

The Tokai to Kamioka (T2K) experiment uses neutrinos generated from the proton synchrotron at J-PARC. With a shorter baseline of 295 km it is located off-axis from the center of the neutrino beam. This results in a shift of the neutrino energy spectrum towards the lower end. The far detector is the Super Kamiokande water-Cherenkov detector. The combination of off-axis placement coupled with the set of near detectors and large far detector has resulted in precision measurements of the electron neutrino appearance channel. Measurements performed in both the neutrino and antineutrino mode allow for the measurement of the amount of CP violation present in the neutrino sector.

The NOvA experiment also falls in the category of long-baseline experiments. Chapter 3 provides a more thorough treatment of the experiment and its most recent results. In relation to the other experiments discussed here,

NOvA's achievements to date are briefly noted here. The NOvA experiment has confirmed electron neutrino appearance in the NuMI muon neutrino beam. Furthermore electron antineutrino appearance has been measured. Finally analysis of the data collected has led to constraints and, in some combinations of parameters, exclusion of some values of the CP-violating phase. An overview of the latest measurements of the NOvA experiment is given in section 2.5.1.

Two large scale future experiments are planned, the Deep Underground Neutrino Experiment (DUNE) and Hyper-Kamiokande. Hyper-Kamiokande will be a 260 kt water-Cherenkov detector serving as the successor to Super-Kamiokande. It therefore will share the baseline as the upgrades at J-PARC will provide the neutrino beam for this experiment. The DUNE experiment will be using the upgrades to the Main Injector at Fermilab to observe neutrinos travelling 1300 km to the Sanford Underground Research Facility. The far detector will consist of 4 separate modules taking advantage of advances in liquid argon time projection chamber technologies to achieve fine grained resolution of the neutrino interactions.

An overview of long baseline neutrino oscillation experiments is provided in table 2.3

2.4.2 Short-baseline experiments

The study of neutrino oscillations with accelerator neutrinos is not limited to high energies and long baselines. Baselines in the range of a single kilometer are considered short. When coupled with neutrinos of sufficiently low energies, observations and measurements of neutrino oscillations are possible.

The 800 MeV linac located at the Los Alamos National Laboratory produced pions which would stop in the subsequent target. This resulted in negative pions being adsorbed by the material in the target whereas the positive pions would decay. Their decay products would then further decay. The resulting beam therefore consisted of muon neutrinos, muon antineutrinos and electron neutrinos. Given the branching ratios of the decays involved the contamination of electron antineutrinos should be low. The Liquid Scintillator

Name	Far detector technology	Beamline	Baseline (km)	Neutrino Energy (GeV)	Years
K2K	Water-Cherenkov	KEK-PS	250	1.3	1999-2004
MINOS	Iron-scintillator	NuMI	735	3	2005-2013
MINOS+	Iron-scintillator	NuMI	735	7	2013-2016
OPERA	Emulsion hybrid	CNGS	730	17	2008-2012
ICARUS	Liquid argon TPC	CNGS	730	17	2010-2012
T2K	Water-Cherenkov	J-PARC	295	0.6	2010-
NOvA	Liquid scint. tracking calorimeter	NuMI	810	2	2014-
DUNE	Liquid argon TPC	LBNF	1300	2-3	
Hyper-Kamiokande	Water-Cherenkov	J-PARC	295	0.6	

Table 2.3: Overview of long baseline accelerator neutrino experiments. Recreated from [37].

Neutrino Detector (LSND) measured the electron antineutrino appearance. The detector consisted of 167 tonnes of liquid scintillator located 30m from the neutrino source. An excess of events was measured [63].

The KARlsruhe Rutherford Medium Energy Neutrino (KARMEN) experiment used protons from the ISIS synchrotron at the Rutherford Appleton Laboratory. These protons would produce pions when interacting with the target. After 17 meters the produced neutrinos would be measured by the segmented liquid scintillation calorimeter. The KARMEN experiment made measurements of the electron antineutrino appearance channel [64]. Their findings were consistent with the predictions from modelling of the background and therefore did not agree with the LSND experiment.

The Mini Booster Neutrino Experiment (MiniBooNE) was a detector filled with 0.818 kt of ultra-refined methylene compounds. Lining the detector walls are 1280 PMT's. It is located in Fermilab's Booster Neutrino Beamline (BNB)

at a distance of 541 meters from the target. As this beamline can be operated in either neutrino or antineutrino mode measurements of electron neutrino and electron antineutrino appearances have been made. Excesses in both of these measurements have been found [65].

Additional experiments were constructed in the BNB to further investigate the observed excess. Collectively these experiments are known as the Short-Baseline Neutrino (SBN) program at Fermilab. The SBND detector is located 110 meters from the target, MicroBooNE is 470 meters away and finally ICARUS is at 600 meters. These three experiments use liquid argon TPC's with variable sizes. Recent analysis from MicroBooNE found no significant low energy electron neutrino excess [66] [67] [68] [69].

The observed short-baseline excesses of the ν_e and $\bar{\nu}_e$ appearance oscillation channels could be consistent with models which include one or more sterile right handed neutrinos. This evidence for the existence of sterile neutrino is widely regarded as controversial due to the incompatibility of these results with those found by other experiments, such as the KARMEN experiment, other short and long baseline ν_μ disappearance experiments and most recently the MicroBooNE experiment.

2.5 Experimental values of the oscillation parameters

The experiments discussed in detail each use different methods and technologies to make measurements of various parameters in the model of neutrino oscillations. Combination of their respective data allows for a thorough analysis of the entire model. These efforts are referred to as the three flavour ($3 - \nu$) oscillation analysis. Using the individual measurements and their errors the six parameters of the neutrino oscillations can be determined. These parameters are the two squared mass differences, the three mixing angles and the CP violating phase. Numerical analysis leads to preferred values for these parameters. These values change depending on which datasets are included in the analysis. New experimental data is continually incorporated in this

analysis. The preferred values vary depending on which hierarchy is assumed to be correct, either the normal or inverted mass hierarchies. Changing the ordering of the mass states leads to different preferred values. The combination of a set of preferred values is referred to as the best fit. In table 2.4 the contributions of the experiment types and individual experiments are listed. As not all types of neutrino sources and neutrino detector technologies are equally sensitive to measurement of all the relevant neutrino oscillation parameters, the contributions vary from experiment type and the individual experiments. The contributions are further subdivided by whether they form the primary or secondary contribution. Secondary contributions can be confirmations, less precise measurements employing a different technique, or usage of a different neutrino source. All of which can aid in reducing the overall uncertainty of the specific relevant oscillation parameter.

Experiment	Dominant	Important
Solar neutrino experiments (2.1.1)	θ_{12}	$\Delta m_{21}^2, \theta_{13}$
Reactor LBL (KamLAND)	Δm_{21}^2	θ_{12}, θ_{13}
Reactor MBL (Daya-Bay, Reno, D-Chooz)	$\theta_{13}, \Delta m_{31,32}^2 $	
Atmospheric Experiments (SK, IC-DC)	$\theta_{23}, \Delta m_{31,32}^2 $	θ_{13}, δ_{CP}
Accelerator LBL $\nu_{\mu}, \bar{\nu}_{\mu}$ Disappearance (K2K, MINOS, T2K, NOvA)	$ \Delta m_{31,32}^2 , \theta_{23}$	
Accelerator LBL $\nu_e, \bar{\nu}_e$ Appearance (MINOS, T2K, NOvA)	δ_{CP}	θ_{13}, θ_{23}

Table 2.4: Overview of experiments contributing to the current best determination of the neutrino oscillation parameters. Recreated from [37].

The NuFIT collaboration is one of the groups producing the full three flavour analysis. The recent results are summarized in table 2.5. Not all necessary information is publicly available for all the experiments which contribute to this analysis. For example the Super-Kamiokande collaboration has not released the χ^2 uncertainty maps for their latest analysis. In table 2.5 two versions of the global fit are shown, one which includes the latest SK atmospheric results without the accompanying uncertainty map. Older versions of these are available and have been re-used for this latest global fit.

		NuFIT 5.1 (2021)			
		Normal Ordering (best fit)		Inverted Ordering ($\Delta\chi^2 = 2.6$)	
		bfp $\pm 1\sigma$	3σ range	bfp $\pm 1\sigma$	3σ range
without SK atmospheric data	$\sin^2 \theta_{12}$	$0.304^{+0.013}_{-0.012}$	0.269 \rightarrow 0.343	$0.304^{+0.012}_{-0.012}$	0.269 \rightarrow 0.343
	$\theta_{12}/^\circ$	$33.44^{+0.77}_{-0.74}$	31.27 \rightarrow 35.86	$33.45^{+0.77}_{-0.74}$	31.27 \rightarrow 35.87
	$\sin^2 \theta_{23}$	$0.573^{+0.018}_{-0.023}$	0.405 \rightarrow 0.620	$0.578^{+0.017}_{-0.021}$	0.410 \rightarrow 0.623
	$\theta_{23}/^\circ$	$49.2^{+1.0}_{-1.3}$	39.5 \rightarrow 52.0	$49.5^{+1.0}_{-1.2}$	39.8 \rightarrow 52.1
	$\sin^2 \theta_{13}$	$0.02220^{+0.00068}_{-0.00062}$	0.02034 \rightarrow 0.02430	$0.02238^{+0.00064}_{-0.00062}$	0.02053 \rightarrow 0.02434
	$\theta_{13}/^\circ$	$8.57^{+0.13}_{-0.12}$	8.20 \rightarrow 8.97	$8.60^{+0.12}_{-0.12}$	8.24 \rightarrow 8.98
	$\delta_{\text{CP}}/^\circ$	194^{+52}_{-25}	105 \rightarrow 405	287^{+27}_{-32}	192 \rightarrow 361
	$\frac{\Delta m_{21}^2}{10^{-5} \text{ eV}^2}$	$7.42^{+0.21}_{-0.20}$	6.82 \rightarrow 8.04	$7.42^{+0.21}_{-0.20}$	6.82 \rightarrow 8.04
	$\frac{\Delta m_{3\ell}^2}{10^{-3} \text{ eV}^2}$	$+2.515^{+0.028}_{-0.028}$	+2.431 \rightarrow +2.599	$-2.498^{+0.028}_{-0.029}$	-2.584 \rightarrow -2.413
	with SK atmospheric data	$\sin^2 \theta_{12}$	$0.304^{+0.012}_{-0.012}$	0.269 \rightarrow 0.343	$0.304^{+0.013}_{-0.012}$
$\theta_{12}/^\circ$		$33.45^{+0.77}_{-0.75}$	31.27 \rightarrow 35.87	$33.45^{+0.78}_{-0.75}$	31.27 \rightarrow 35.87
$\sin^2 \theta_{23}$		$0.450^{+0.019}_{-0.016}$	0.408 \rightarrow 0.603	$0.570^{+0.016}_{-0.022}$	0.410 \rightarrow 0.613
$\theta_{23}/^\circ$		$42.1^{+1.1}_{-0.9}$	39.7 \rightarrow 50.9	$49.0^{+0.9}_{-1.3}$	39.8 \rightarrow 51.6
$\sin^2 \theta_{13}$		$0.02246^{+0.00062}_{-0.00062}$	0.02060 \rightarrow 0.02435	$0.02241^{+0.00074}_{-0.00062}$	0.02055 \rightarrow 0.02457
$\theta_{13}/^\circ$		$8.62^{+0.12}_{-0.12}$	8.25 \rightarrow 8.98	$8.61^{+0.14}_{-0.12}$	8.24 \rightarrow 9.02
$\delta_{\text{CP}}/^\circ$		230^{+36}_{-25}	144 \rightarrow 350	278^{+22}_{-30}	194 \rightarrow 345
$\frac{\Delta m_{21}^2}{10^{-5} \text{ eV}^2}$		$7.42^{+0.21}_{-0.20}$	6.82 \rightarrow 8.04	$7.42^{+0.21}_{-0.20}$	6.82 \rightarrow 8.04
$\frac{\Delta m_{3\ell}^2}{10^{-3} \text{ eV}^2}$		$+2.510^{+0.027}_{-0.027}$	+2.430 \rightarrow +2.593	$-2.490^{+0.026}_{-0.028}$	-2.574 \rightarrow -2.410

Table 2.5: Three flavour oscillation parameters from the NuFIT fit to global data. The best fit points (bfp) for both the normal and inverted mass hierarchies are shown. Furthermore the one and three sigma confidence intervals for the oscillation parameters are given. Two variations of the oscillation parameters are shown, differing by the inclusion of the latest Super-Kamiokande atmospheric data (SK-atm). Courtesy of [37].

2.5.1 Measurements of the NOvA experiment

The NOvA experiment observes the $\nu_\mu \rightarrow \nu_e$, $\nu_\mu \rightarrow \nu_\mu$, $\bar{\nu}_\mu \rightarrow \bar{\nu}_e$, and $\bar{\nu}_\mu \rightarrow \bar{\nu}_\mu$ oscillation channels. Using measurements of the unoscillated beam at the Near Detector, predictions for the expected number of candidate neutrino events at the Far Detector can be made. With the observed number of neutrino events at the Far Detector a joint fit can be performed to extract the set of oscillation parameters which best matches the recorded data.

Using selected neutrino events the ν_μ and ν_e neutrino energy spectra for both beam modes can be reconstructed and are shown in Figure 2.8.

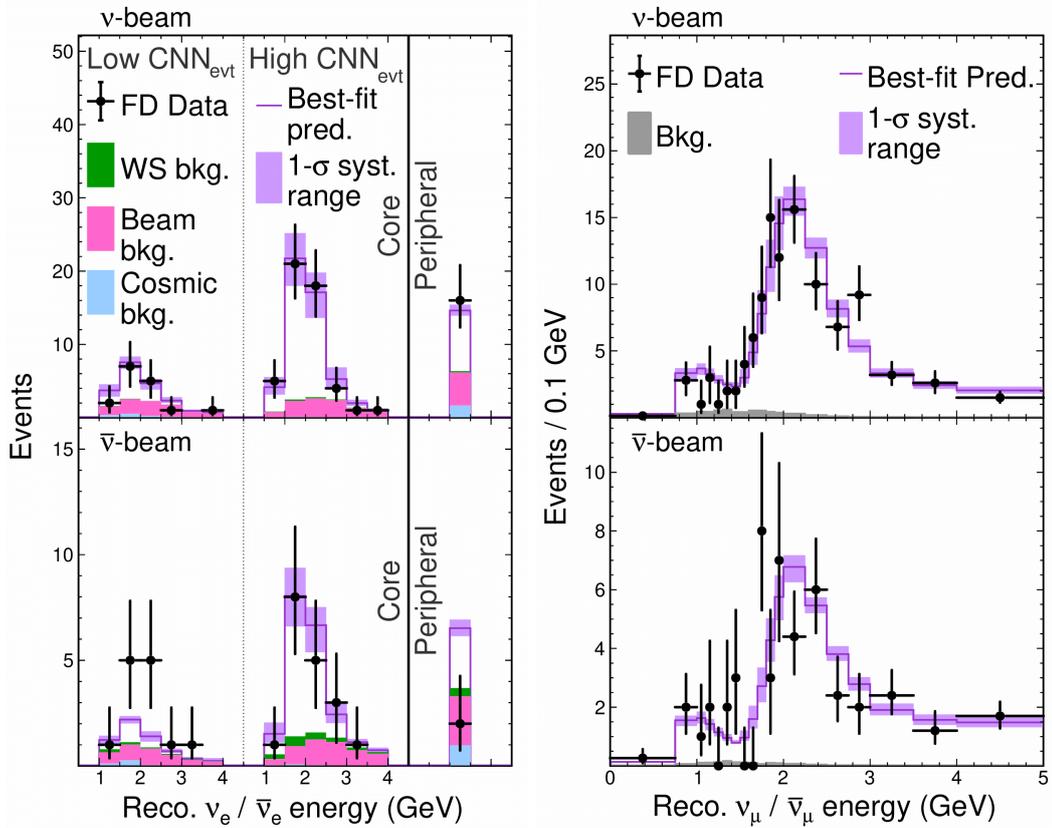


Figure 2.8: Reconstructed FD neutrino energy spectra for neutrino (top) and antineutrino (bottom) beam modes. The $\nu_e/\bar{\nu}_e$ CC samples are shown separately (left) while the combined sample is shown for the $\nu_\mu/\bar{\nu}_\mu$ sample (right).

The analysis of the combined data samples results in a best fit for the relevant oscillation parameters. A summary of these values is shown in table 2.6.

Oscillation parameter	Normal ordering		Inverted ordering	
	Upper octant ($\theta_{23} > \pi/4$)	Lower octant ($\theta_{23} < \pi/4$)	Upper octant	Lower octant
Δm_{32}^2 (10^{-3} eV^2)	$+2.41^{+0.07}_{-0.07}$	$+2.39$	-2.45	-2.44
$\sin^2 \theta_{23}$	$0.57^{+0.03}_{-0.04}$	0.46	0.56	0.46
δ_{CP} (π)	$0.82^{+0.27}_{-0.87}$	0.07	1.52	1.41
Rejection significance	-	1.1σ	0.9σ	1.1σ

Table 2.6: Best fit values of the oscillation parameters for various configurations of the mass ordering and θ_{23} octant. Recreated from [70].

Using the Feldman-Cousins unified approach [71], [72] the confidence intervals are determined for the oscillation parameters. The 90% confidence level regions for the oscillation parameters are plotted and comparisons with similar measurements of other experiments are provided in Figure 2.9.

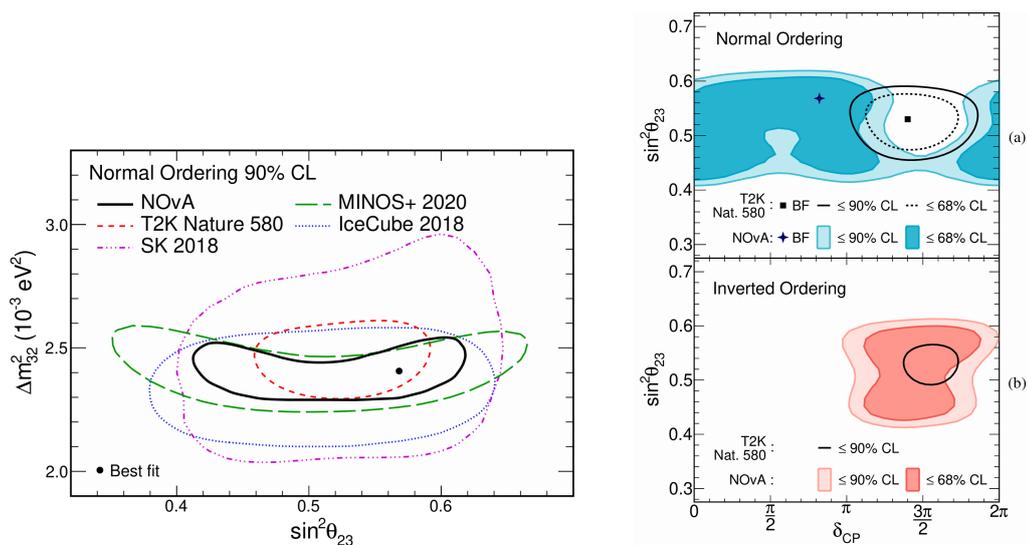


Figure 2.9: Confidence level contours for oscillation parameters. Confidence level contours from measurements of other experiments of the same oscillation parameters are included.

Further elaboration on the data samples and selection is provided in section 3.3.1.

Chapter 3

The NOvA Experiment

The NuMI Off-Axis electron Neutrino Appearance experiment (NOvA) is a long-baseline neutrino oscillation experiment. It consists of two detectors which are functionally identical to each other. Both of these detectors are placed in the NuMI neutrino beam. The near detector is located underground onsite at Fermilab at roughly a kilometer from the beam's target. The far detector is located 810 km away close to the Canadian border, on the surface at Ash River in the state of Minnesota. Both detectors are off-axis in relation to the beam by 14.6 milliradians. This results in a narrowing of the neutrino energy spectrum reducing the magnitude of the background. The baseline length has been chosen to maximize the $\nu_\mu \rightarrow \nu_e$ appearance oscillation probability. Likewise the detector design has been chosen to optimize identification of ν_e charged current events.

3.1 Fermilab and the NuMI beamline

At the Fermi National Accelerator Laboratory (Fermilab) protons are accelerated in the Main Injector to an energy of 120 GeV. They are subsequently directed onto a graphite target with an initial design power of 700 kW. Continual upgrades to the accelerator complex are resulting in higher beam powers. A power of 900 kW was reached on the 25th of April 2022.

In figure 3.2 a schematic overview of the Fermilab accelerator complex is shown. The accelerator complex begins at the ion sources. Here hydrogen gas



Figure 3.1: Map of the off-axis beamline originating at Fermilab. The near detector is located at Fermilab approximately one kilometer from the target, whereas the far detector is located 810 kilometers away at Ash River. Courtesy of [73].

is ionized to produce negative hydrogen ions. These hydrogen ions are then accelerated in stages, first to 35 KeV then to 750 KeV by radio-frequency (RF) quadrupoles. The hydrogen ions go from the ion source to the linear accelerator (linac). The linac employs a combination of drift tubes and RF-cavities to accelerate the hydrogen ions to 400 MeV. At the end of the linac a carbon foil is placed through which the hydrogen ions pass. This causes them to surrender their electrons, finally becoming protons.

The process of extracting hydrogen ions from the source and accelerating them is not a continuous process, it occurs in groupings. Each of these ion groups

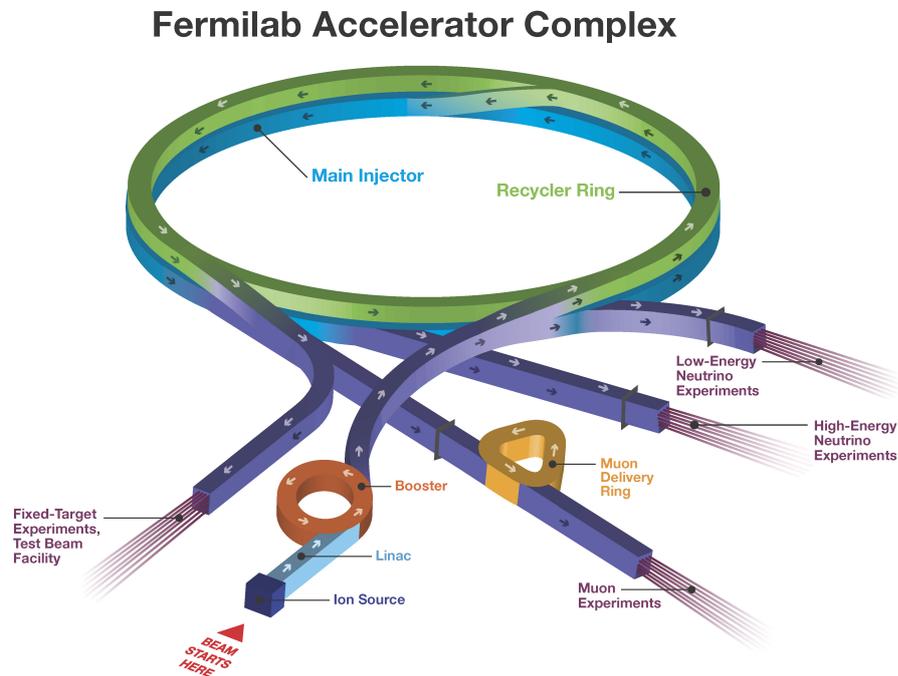


Figure 3.2: Schematic overview of the Fermilab accelerator complex. Courtesy of [74].

(and eventually proton groups) is called a bunch and contains approximately 4×10^{12} particles.

These protons then enter the Booster ring. This ring consists of magnets which bend the path of the charged particles into a circle. The radius of the Booster ring is 74.5 meters and its circumference is 468 meters. The protons which are now traveling along the Booster ring are accelerated to 8 GeV. After exiting the Booster ring the protons enter the Recycler. Originally the Recycler was used for storage and cooling of anti-protons. Now it is used for the injection of Booster ring protons into the Main Injector. This is done through a process called slipstacking which effectively allows for the doubling of the beam intensity by superimposing pairs of bunches. The final stage of acceleration is the Main Injector. Here energies of 120 GeV are reached. From the Main Injector protons are distributed to various experiments as required, for direct usage or for the

production of other particles such as neutrinos.

The Main Injector supplies the Neutrinos at the Main Injector (NuMI) beam with proton bunches. These are grouped into spills. A spill is the collection of 12 proton bunches provided in a 10 micro second window. A spill is supplied approximately once every second.

The NuMI beam starts with the collision of the 120 GeV Main Injector protons and the graphite target. The high energy protons can produce a large variety of secondary particles. Of these secondary particles the charged mesons (kaons, pions) are most abundant. Located directly behind the target a pair of magnetic horns are capable of deflecting the charged mesons. For the Forward Horn Current (FHC) mode the generated magnetic fields focus the positively charged mesons along the direction of the beamline. Inverting the applied current results in the Reverse Horn Current (RHC) mode. For this mode the negatively charged mesons are focussed. The alternately charged mesons will be deflected into the surrounding material. The charged mesons remaining in the beam enter into the decay pipe. This pipe is 675 meters long and has a diameter of 2 meters. The charged mesons decay into neutrinos, muons and electrons,

$$\begin{aligned}
 \pi^{+(-)} &\rightarrow \mu^{+(-)} + \nu_{\mu}(\bar{\nu}_{\mu}) \\
 \pi^{+(-)} &\rightarrow e^{+(-)} + \nu_e(\bar{\nu}_e) \\
 K^{+(-)} &\rightarrow \mu^{+(-)} + \nu_{\mu}(\bar{\nu}_{\mu}) \\
 K^{+(-)} &\rightarrow \pi^0 + e^{+(-)} + \nu_e(\bar{\nu}_e) \\
 \mu^{+(-)} &\rightarrow e^{+(-)} + \nu_e(\bar{\nu}_e) + \bar{\nu}_{\mu}(\nu_{\mu})
 \end{aligned} \tag{3.1}$$

The branching fractions for these decays are not equal to one another. The positive pion decay to ν_{μ} has a branching fraction 99.98% whereas the positive kaon decay to ν_{μ} has a fraction of 63.55%. Therefore in FHC mode the beam composition will be dominated by muon neutrinos prior to any oscillations occurring. For RHC mode the muon antineutrinos are the dominant flavour. There will still be contamination of other neutrinos in the beam. Detailed simulations can predict the amount of contamination present at various stages of

the beamline and subsequent detector locations. In Figure 3.3 the composition of the neutrino beam as it would be measured without oscillations present is shown.

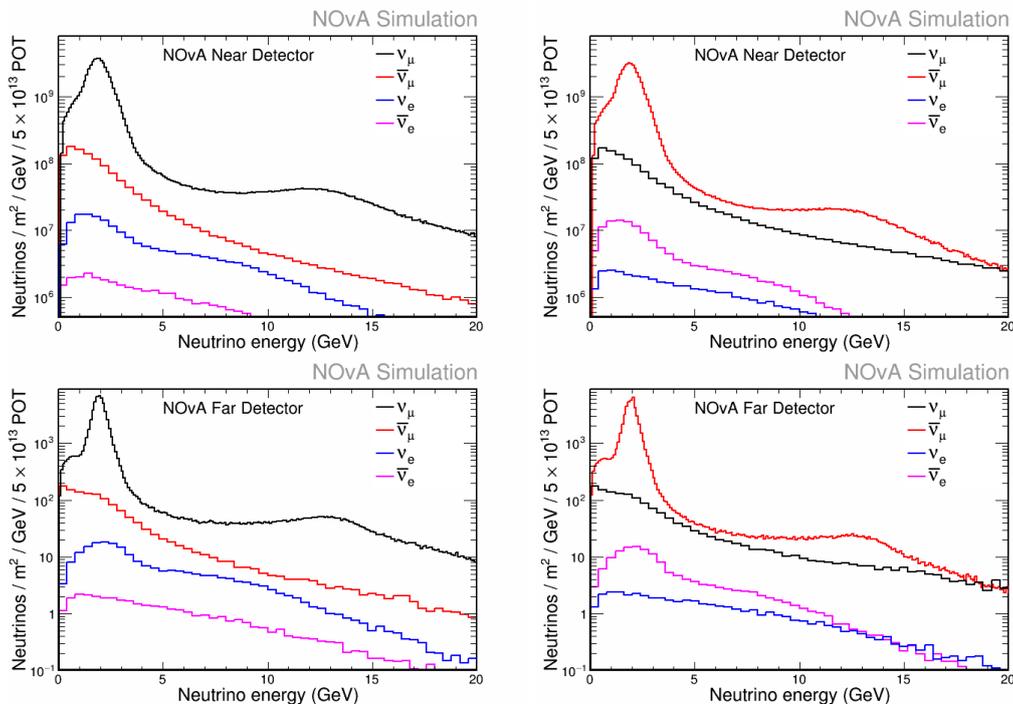


Figure 3.3: Expected number of neutrino events at both the near(top) and far(bottom) detectors for the FHC(left) and RHC(right) modes. This simulation includes flux and neutrino cross section corrections, but precludes any oscillation effects. Neutrino energies ranging from 1-5 GeV are included. Courtesy of [75].

The amount of contamination is different for the two beam operation modes. For FHC there is approximately 2% of muon antineutrino contamination. For RHC the amount of muon neutrino contamination is considerably higher at approximately 11%. This higher contamination is a direct consequence of the smaller cross-sections of antineutrinos. Additionally other sources of contamination are present. Kaons can decay to a neutral pion, positron and electron neutrino with a branching fraction of 5.07%. The charge conjugated version of this decay can also occur, providing a source of electron antineutrinos. Muon decays also provide a small source of electron (anti) neutrinos. At the end of the decay pipe a hadron monitor is used to measure the transverse width and intensity of the hadronic parts of the beam which have not yet decayed

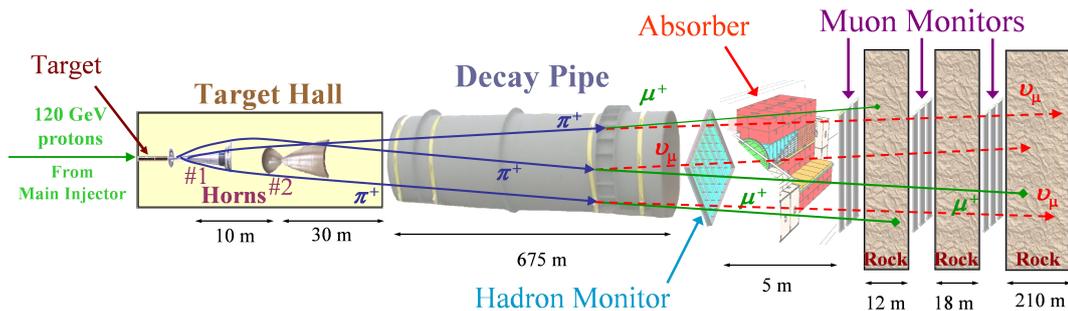


Figure 3.4: Schematic overview of the NuMI beamline. Courtesy of [76].

away. An absorber stops any of the hadronic components from penetrating to the muon monitors located behind it. These consist of three alcoves in the rock that house three independent muon monitors. These measure the residual muon flux in the beam at different locations and threshold energies. Figure 3.4 displays a schematic overview of the NuMI beamline with the relevant components detailed.

3.2 Off-axis design

The majority of neutrinos present in the NuMI beam are produced from pion and kaon decays. In their rest frame these decays produce mono-energetic neutrinos. In the lab frame of reference the kaons and pions are boosted. The energy spectrum of the boosted neutrinos has a broader distribution and is dependent on the angle with respect to the original boost direction. The energy E_ν and flux F_ν [22] of neutrinos originating from boosted pion decays can be given by,

$$\begin{aligned}
 F_\nu &= \left(\frac{2\gamma}{1 + \gamma^2 \theta^2} \right) \frac{A}{4\pi z^2} \\
 E_\nu &= \frac{0.43 E_\pi}{1 + \gamma^2 \theta^2}
 \end{aligned}
 \tag{3.2}$$

These expressions are an approximation which is valid for small angles. The area of the observing detector is given by A , z is the distance between the source and the detection point, E_π is the energy of the parent pion, $\gamma = \frac{E_\pi}{m_\pi}$, θ is the angle between the boost direction of the original pion and the produced

neutrino. For kaons the factor 0.43 becomes 0.96 in this approximation.

In Figure 3.5 on the left, the neutrino energy spectrum is plotted. As the angle increases the dependence of the neutrino energy on the pion energy decreases. This effect is non-linear and allows for the production of a neutrino energy spectrum which is narrowly peaked at 2 GeV for an angle of 14.6 mrad. This coincides with the expected location of an oscillation maximum for the 810 kilometer baseline. The effects of a change in angle on the observed neutrino energy spectrum can be seen in Figure 3.5 on the right. The medium energy tune refers to one of the various possible distances for the target and horn positions which yields the optimal performance for the NOvA experiment.

Employing off-axis detectors can have additional benefits. Several backgrounds to the ν_e appearance signal can be reduced. The first of these backgrounds is caused by neutral current events where the neutrino scatters and leaves the detector. The experimental topology of this event is similar to that of ν_e charged current interactions. Most of these neutral current background events are observed in the lower energy ranges, as the neutrino keeps travelling without depositing all of its energy and therefore only a fraction of the total neutrino energy is deposited into the detector to be observed. Reducing the high energy tail of the neutrino energy spectrum suppresses this background. Another background is the ν_e contamination present in the beam itself (intrinsic). The energy spectrum of these neutrinos is broader as they are more likely to have been produced in decays involving more produced particles. This difference in shape of the energy spectrum can be used to reject a significant fraction of these background events. The bottom plot in Figure 3.5 shows simulated energy spectra for the various signal and background components to the observed off-axis beam.

3.3 Delivered beam

The total beam exposure of experiments in the NuMI beam is measured by the number of protons colliding with the graphite target (POT). The latest

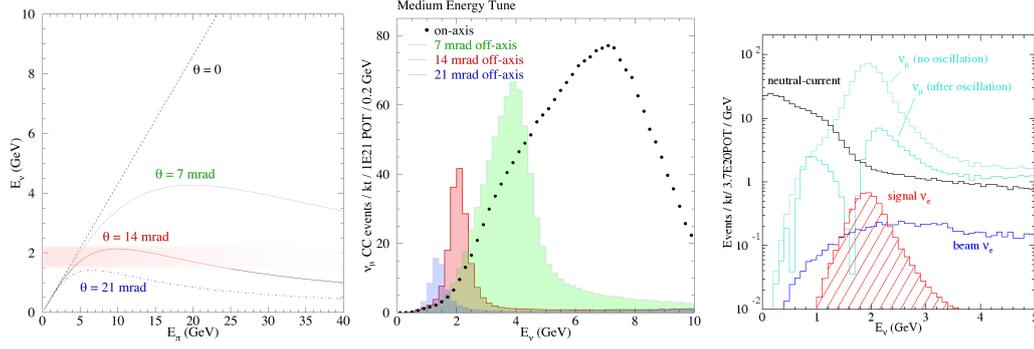


Figure 3.5: Left: The neutrino energy as a function of angle and pion energy. Contributions to the peak of the beam energy spectrum for 14.6 mrad have been highlighted in red. Middle: Expected event rates for ν_μ charged current interactions without oscillations. These rates have been calculated for detectors placed at 810 km and at various angles with respect to the beam. Right: Simulated event rates for ν_e oscillation, intrinsic beam ν_e , NC, ν_μ CC events with and without oscillations. The following oscillation parameters were assumed: $\Delta m_{32}^2 = 2.5 \times 10^{-3} eV^2$, $\sin^2(2\theta_{23}) = 1.0$, $\sin^2(2\theta_{13}) = 0.1$ at a distance of 810 km, 14.6 mrad off-axis. Figures adapted from [22].

published NOvA analysis [70] has been completed with 13.6×10^{20} POT in neutrino (FHC) mode and 12.5×10^{20} POT in antineutrino (RHC) mode. The FHC data taking period for this dataset began on the 6th of February 2014 up until the 20th of March 2020. The RHC dataset was recorded from June 29th 2016 up until February 26th 2019. During these data taking periods the average beam power was 650 kW.

In Figure 3.6 the beam exposure is shown from the start of data taking (commissioning) until the latest period included in this analysis.

3.3.1 Data selection

Each event in the dataset has to pass basic quality and containment criteria to be considered for inclusion in the analysis. A convolution neural network, Event CVN, is utilized to classify the neutrino event candidates by their interaction types, ν_μ charged current, ν_e charged current, neutral current or cosmic background. The selection criteria and this neural network are described in more detail in chapters 5 and 6. Further subdivisions of the ν_μ ($\bar{\nu}_\mu$) CC and ν_e ($\bar{\nu}_e$) CC samples based on the Event CVN scores for ν_e ($\bar{\nu}_e$) and on the

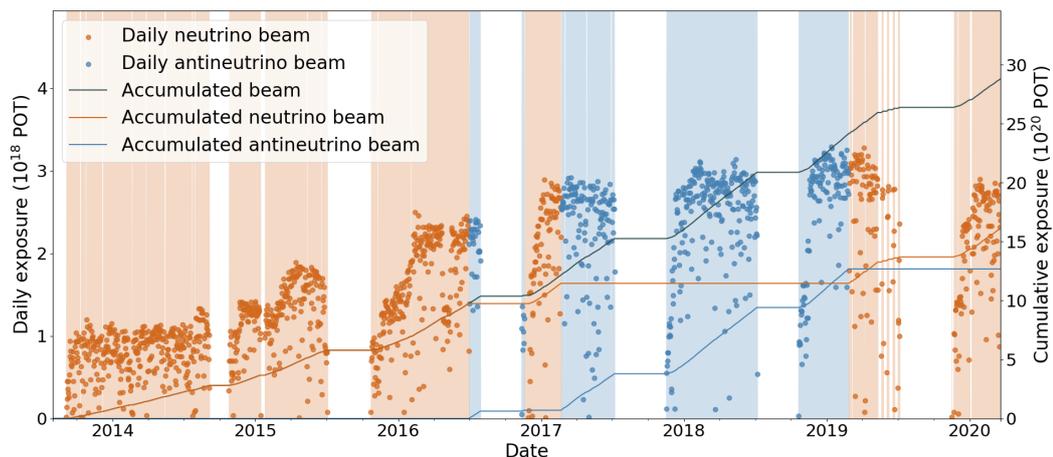


Figure 3.6: Time series of the daily (anti) neutrino beam POT up until March 2020.

hadronic energy fraction $E_{frac} = E_{had}/E_\nu$ for ν_μ ($\bar{\nu}_\mu$) are made. This binning maximizes sensitivity to the oscillation parameters in the analysis. For the ν_e ($\bar{\nu}_e$) samples, events that fail some of the initial selection criteria but pass the stricter criteria imposed by the Event CVN or cosmic background rejection methods are grouped into the peripheral sample. The events in this sample are included solely as an integrated rate in the oscillation fits. In table 3.1 the purity (see equation (7.5)) of the samples selected by the Event CVN.

Sample bins	Sample Purity (%)	
	ν_e	$\bar{\nu}_e$
Core, Low CVN score	51	36
Core, High CVN score	97	69
Peripheral	57	43
Combined	69	58
	ν_μ	$\bar{\nu}_\mu$
1 (Lowest E_{frac})	99	99
2	99	99
3	97	98
4 (Highest E_{frac})	92	95
Combined	96	98

Table 3.1: Purity of the FD selected ν_μ , $\bar{\nu}_\mu$, ν_e , and $\bar{\nu}_e$ samples. Recreated from [70].

The number of expected and observed neutrino events at the Far Detector are summarized in table 3.2.

	Neutrino beam		Antineutrino beam	
	ν_μ CC	ν_e CC	$\bar{\nu}_\mu$ CC	$\bar{\nu}_e$ CC
$\nu_\mu \rightarrow \nu_\mu$	201.1	1.7	26.0	0.2
$\bar{\nu}_\mu \rightarrow \bar{\nu}_\mu$	12.6	0.0	77.2	0.2
$\nu_\mu \rightarrow \nu_e$	0.1	59.0	0.0	2.3
$\bar{\nu}_\mu \rightarrow \bar{\nu}_e$	0.0	1.0	0.0	2.3
Beam $\nu_e + \bar{\nu}_e$	0.0	14.1	0.0	7.3
NC	2.6	6.3	0.8	2.2
Cosmic	5.0	3.1	0.9	1.6
Others	0.9	0.5	0.4	0.3
Signal	$214.1^{+14.4}_{-14.0}$	$59.0^{+2.5}_{-2.5}$	$103.4^{+7.1}_{-7.0}$	$19.2^{+0.6}_{-0.7}$
Background	$8.2^{+1.9}_{-1.7}$	$26.8^{+1.6}_{-1.7}$	$2.1^{+0.7}_{-0.7}$	$14.0^{+0.9}_{-1.0}$
Best fit	222.3	85.8	105.4	33.2
Observed	211	82	105	33

Table 3.2: Observed and predicted event counts at the NOvA Far Detector. Recreated from [70].

3.4 Detector design

Both the Near Detector (ND) and Far Detector (FD) subscribe to the same design philosophy, they are both liquid scintillator tracking detectors. The Near Detector is located at Fermilab 100 meters underground and approximately 1 km from the NuMI target. It weighs 290 tonnes. To ensure that minimum ionizing muons are contained in the detector a stack of 10 steel plates interlaced with detector planes is added at the end of the Near Detector. This is referred to as the muon catcher. As the Near Detector is located underground the background rate of cosmic muons is significantly reduced. The Far Detector is located at Ash River in Minnesota 810 kilometers away from Fermilab. It is significantly larger than the Near Detector and weighs 14 kilotonnes. This large mass is required due to the small neutrino cross-sections and small neutrino flux present at this distance from the beam compared to the flux at the ND. There is only a small overburden covering the Far Detector and therefore the

cosmic muon rate is high. This background while significant can be reduced to acceptable levels. The primary reduction comes from the NuMI spill window of 10 μ s which can be applied as a stringent timing cut. This is due to the neutrinos only being delivered from the beam to the detectors within this short window. Additional cosmic rejection cuts can be applied as the experimental signature of a downgoing cosmic muon is distinct from neutrino event signatures.

The primary purpose of the Near Detector is to measure the neutrino energy spectrum of the NuMI beam before the neutrinos have undergone oscillation. This spectrum is the product of the neutrino flux and the interaction cross-section. The Far Detector is located in the same neutrino beam and uses the same technology in its design. These factors allow for a prediction of the expected beam composition at the Far Detector to be made and reduces the magnitude of several systematic uncertainties associated with the neutrino flux and detector components.

Both detectors have been constructed using low- Z materials, with Z being the atomic number representing the number of protons present in the atoms' nucleus. The low Z number of the materials results in a longer radiation length. The radiation length is a characteristic of a material and specifies the average length for an electron to reduce its energy by a predetermined factor. A sufficiently long radiation length allows for multiple points to be observed along a particle's track due to the electromagnetic interactions of the particle with the material. The distance between subsequent points which can be observed are set by the granularity of the detector. Higher levels of granularity or longer particle tracks can improve the discrimination of different experimental signatures.

3.4.1 The NOvA detector cell

The level of granularity is tied to the basic building block of the detectors. A single basic element is called a cell. Each cell consists of a hollow extruded PVC tube. The tube conforms to a rectangular shape with rounded corners. As the detectors are free standing, the PVC cells form the skeleton of the

detector. They therefore need to be strong enough to support the full weight of the detector. To improve structural integrity the corners of the cells are rounded. The dimensions of a cell are 3.9 by 6.6 cm with a length of 3.8 m for the ND and 15.5 m for the FD.

The cells are produced by extruding PVC. This produces 16 cells in a single extrusion. Two of these extrusions are then glued together to produce a module consisting of 32 cells. Multiple modules can be laid out to form a single plane of stacked cells in a set orientation. For the ND 3 modules form a single plane, while an FD plane consists of 12 modules. Converted to cells this means that an ND plane consists of 96 cells and an FD plane consists of 384 cells.

The main body of the detectors is produced by continually stacking planes with alternating orientations. The first plane in both detectors is a plane where the cells are horizontally aligned. The detectors are longest in the z-direction which is oriented along the beam direction.



Figure 3.7: The bare PVC cells stacked in alternating planes to construct the frame of the NOvA detectors. Courtesy of [77].

The ND is constructed by placing 214 planes, one in front of the other, in

this alternating orientation. The last 22 planes are interspersed by steel plates roughly 10 cm thick. These planes consist of only 2 modules in the horizontally aligned planes. This results in the muon catcher being only two thirds as high as the rest of the ND. The primary reason for this reduced height was the reduction in cost as the steel could be reused from earlier experiments. As most muons produced in the beam neutrino interactions are travelling approximately in the direction of the beam this muon catcher is effective for the containment of these muons. In total the ND is approximately 15.9 m long and consists of 20,192 cells. The FD is significantly larger than the ND and consists of 14 diblocks. Each diblock is formed of 64 planes for a total of 896 planes in the detector as a whole. The FD therefore is approximately 60 m long and consists of 344,064 cells.

3.4.2 Liquid scintillator

The hollow PVC cells are filled with a liquid scintillator. This scintillator consists of primarily mineral oil 94.63 – 94.91% supplemented with a mixture of pseudocumene (1,2,4-trimethylbenzene) 4.98 – 5.23%, PPO (2,5-diphenyloxazole) 0.11 – 0.14%, bis-MSB (1,4-bis-(*o*-methyl-styryl)-benzene) 0.0016%, Stadis-425 0.001% and Vitamin E 0.001% [79]. This mixture has been optimized to provide the maximum amount of scintillation light yield while still being relatively safe to handle.

Each of these components serve a particular purpose in the liquid scintillator. The primary light generator is the pseudocumene. As it is excited by charged particles travelling through the liquid, photons are emitted in a narrow spectrum which peaks around the 270 to 320 nm range. The produced photon wavelength puts them in the ultra violet part of the electromagnetic spectrum. These UV photons are then adsorbed by the PPO which in turns emits photons in a spectrum ranging from 340 to 380 nm. A significant component of this spectrum extends outwards to approximately 460 nm. The photons emitted by the PPO can then be adsorbed by the bis-MSB which in turn emits photons in the range of 390 to 440 with a component extending up to 480 nm. Each step

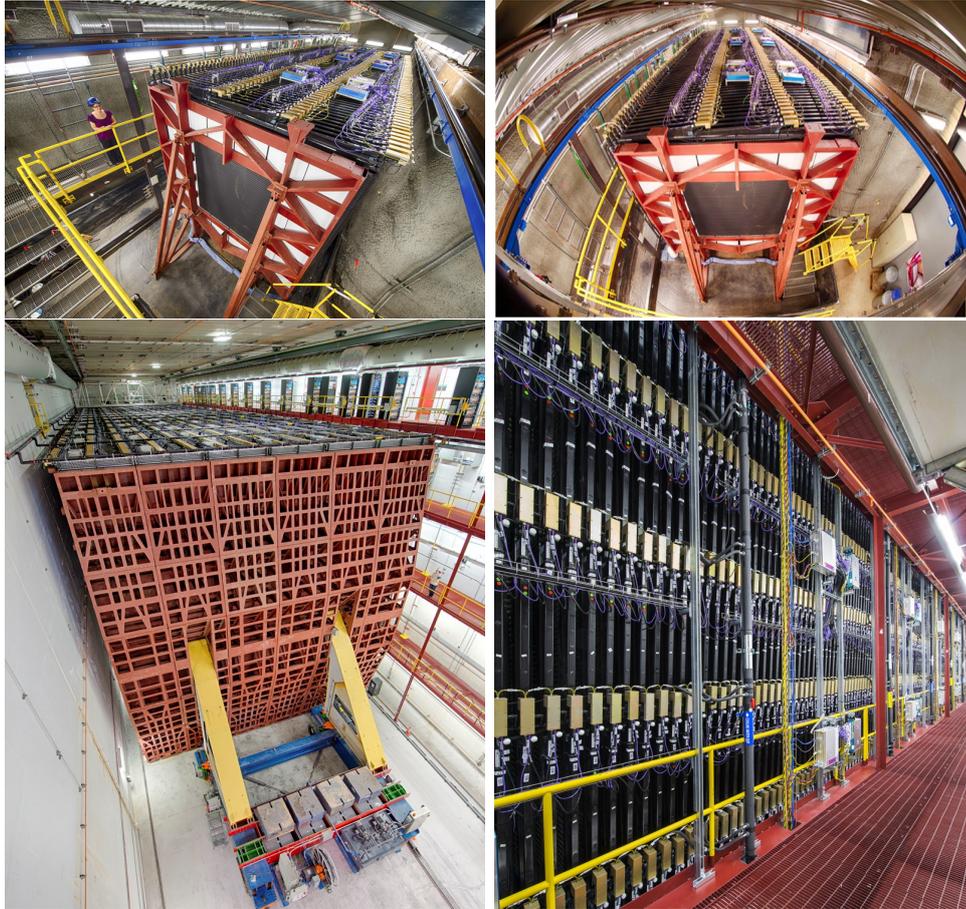


Figure 3.8: The NOvA detectors. The Near Detector is shown from two perspectives on the top two images and likewise the Far Detector is shown on the bottom two. Courtesy of [78].

in these processes reduces the energy of the scintillation photons and therefore their wavelength increases. In Figure 3.9 the emissions profiles for PPO and bis-MSB can be seen. The NOvA scintillator emission profile is almost identical to that of bis-MSB, showing the down-conversion working as expected.

The additional additives are Stadis-425 and Vitamin E. To increase fire safety Stadis-425 is added as an antistatic agent. It is a mixture of the following compounds, solvent naphtha, o-xylene, Benzenesulfonic acid, Kerosine, naphthalene, (1,2,4)-trimethylbenzene [81]. Vitamin E is added as an antioxidant to minimize the potential yellowing of polymers present in components exposed to the liquid scintillator.

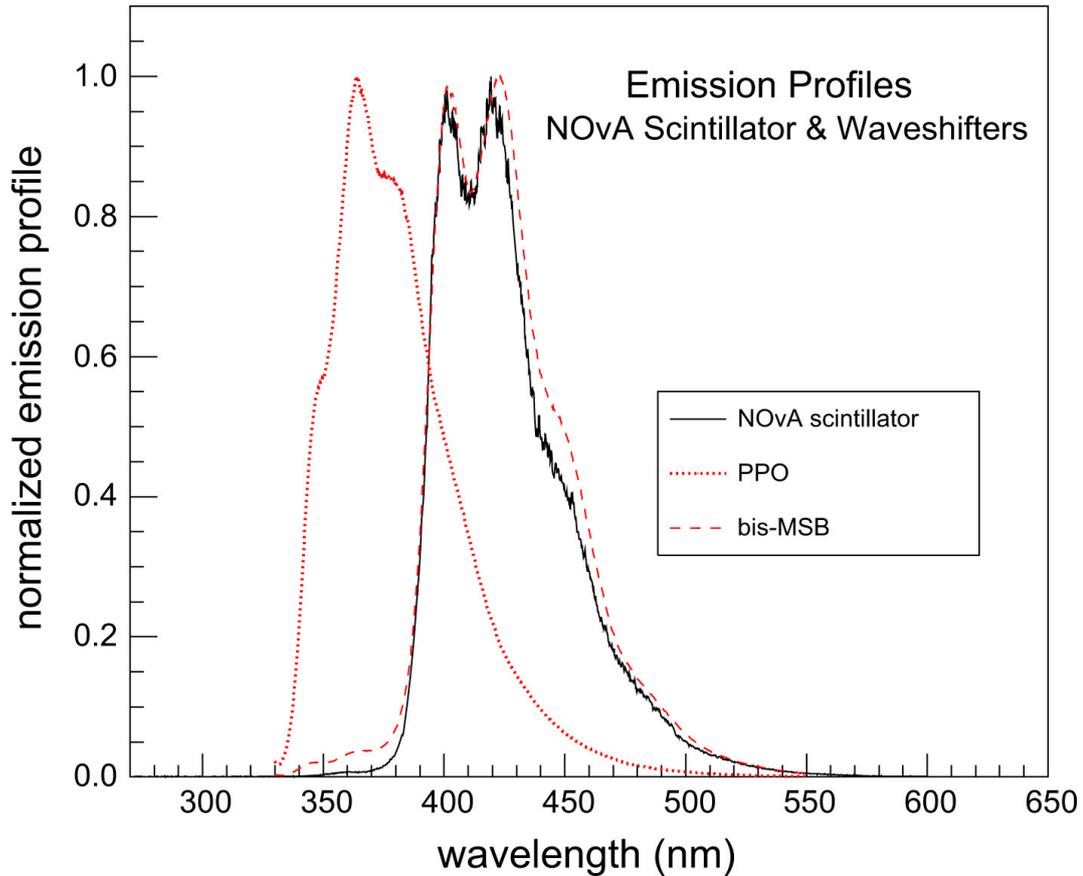


Figure 3.9: Normalized emission profiles for NOvA scintillator, PPO and bis-MSB dissolved in toluene. The peaks of the profiles are set equal to 1. Courtesy of [80].

3.4.3 Wavelength Shifting Fibre

After photons have been emitted by the pseudocumene and down-stepped to a lower wavelength they need to be collected for the purpose of readout. The clear scintillator oil allows for the photons to travel around the liquid in a cell. Unfortunately this process is not efficient enough for usage in the NOvA detectors as the physical make-up of the cell would require the light to travel for up to 15 meters to one end of the cell before readout is possible. To improve the mean pathlength of light travelling in the liquid scintillator titanium dioxide TiO_2 was added to the PVC which makes up the walls of the cells. This reduced the absorption in the cell walls. To circumvent the issue of photon re-adsorption in the liquid scintillator, wave length shifting fibers are added to the cells. A single loop of WLS fibre is placed in a cell with the

two ends at the open end of the cell. The diameter of a single fibre is 0.7 mm. With the fibre in place in the cell, scintillation light bounces around until it hits the fibre. This process has a mean pathlength of 40 cm extending out to 1.5 m until almost all scintillation photons are captured by the fibre. The specifications of the mineral oil therefore required a mean free pathlength of scintillation light beyond these distances.

The WLS fibre primarily captures light in a narrow spectrum. The spectrum extends from 400 to 450 nm. The adsorbed light is emitted in the range of 490 to 550 nm. It can then travel along the length of the fibre through internal reflections. This is not a lossless process and especially for the FD significant attenuation occurs. This final down-stepping of the scintillation light results in primarily green light surviving to the readout.

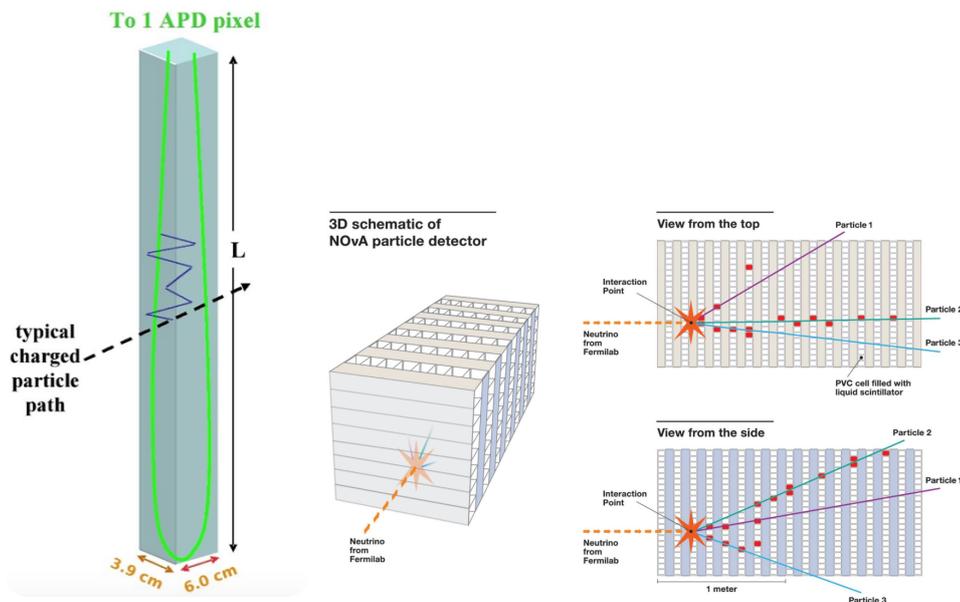


Figure 3.10: Schematic overview of the detector cell and plane structure with the typical scintillation light emitted from the passage of charged particles through the detector cells. The alternating plane structure, reminiscent of a sliced loaf of bread, produces the two distinct pixelated views of the interaction. Courtesy of [82], [83].

3.4.4 Cell optical readout

The readout of one module consists of an avalanche photodiode (APD) with 32 pixels. When a photon hits the collection region of the photodiode pixel it can

be adsorbed and produce an electron and hole. The oppositely charged hole and electron pair separate due to the electric field present in the photodiode. As the electron moves into the border area of the p-type and n-type doped semiconductors, the local strength of the electric field increases. This field strength accelerates the electron to the point where it can cause ionizing collisions. These collisions produce additional electron-hole pairs eventually resulting in an avalanche of freed charges. The magnitude of this avalanche effect is primarily determined by two factors: the local electric field strength at the p-n junction and the average pathlength between ionizing interactions of the electrons. The contributions of these two factors change with the electric field applied across the photodiode and the temperature of the photodiode itself. The APDs are operated under high voltages and low temperatures to maximize the achieved current from a incident photon and reduce the background noise levels. In the NOvA experiment the APDs are cooled and operated at -15°C with voltages resulting in interal APD gains of 100 – 150 [22]. For the light incident from the WLS fibres (500 to 550 nm) the APDs have a quantum efficiency of 85%. This high quantum efficiency, which is close to uniform across this range of the light spectrum, is particularly advantageous for usage in the NOvA detector. The APDs consist of thirty-two 1.6 mm by 1.6 mm pixels. As the fibres are looped through the detector cells the two open ends lead into the same APD pixel.

3.5 DAQ system

The NOvA Data AcQuisition (DAQ) system begins at the APDs. Once light is detected they produce an associated electrical current. This signal is led into the front end board(FEB). Each APD is connected to its own FEB module. The FEB module consists of 4 components. The APD signal is modified to increase the signal and reduce the noise. This is done by an application-specific integrated circuit(ASIC) which combines a low and high pass filter circuit. It is named the CR-RC circuit where the R and C refer to the primary



Figure 3.11: Optical connector consisting of WLS fibres from 32 cells. An unlit connector is shown on the left, on the right an opened extrusion (for demonstration purposes) is being manually lit up and the transported light illuminates the correct cell outputs on the optical connector. Left image courtesy of [84].

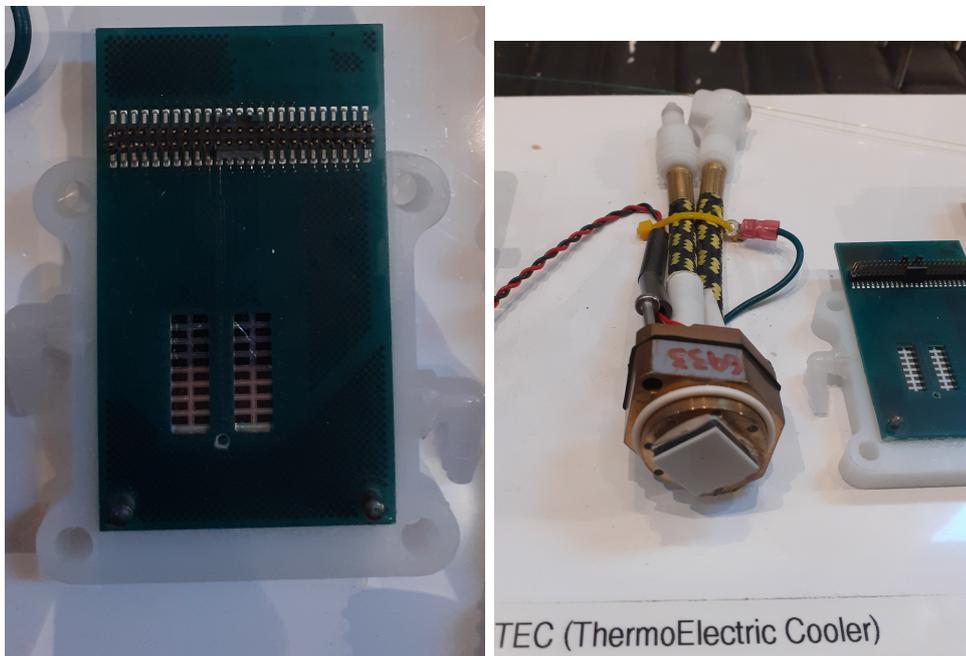


Figure 3.12: The 32 pixel avalanche photodiode and its cooler.

circuit components the resistor and capacitor. The modified signal then passes through an analog to digital converter(ADC). The ADC employs a 64 MHz clock to digitise the signal. After the ADC the signal is passed to a multiplexer (MUX). The MUX is able to combine multiple input signals to a single output datastream. The initial input signals are recoverable from this output stream. For the FD a 8 : 1 MUX is used while the ND uses a 2 : 1 MUX, both operate at 16 MHz. The twofold MUX allows the ND to handle a data rate four times as high. Finally the output signal from the MUX is directed into a Field Programmable Gate Array (FPGA). An FPGA is an integrated reprogrammable circuit primarily consisting of logic block or gates. In this use-case the FPGA boards are used to apply Dual Correlated Sampling (DCS) to the signal. This method removes lower frequency components from the signal and thus reduces the overall noise level and improves the time resolution of the signal. More specifically it compares the values of the signal for the last three clock ticks, gradual changes are filtered out while fast rises are used as indicators of real hits. The continuous output of the FPGAs is now ready to be recorded.

3.5.1 Data concentrator module

The processed signals from the FEBs are collected in Data Concentrator Modules (DCM) which are located on the detectors. These take in signals from 64 FEBs and are thus each monitoring 2048 cells. For the FD there are 168 DCMs and for the ND there are 14 DCMs. In the FD 12 DCMs service a single diblock. Each DCM collects signals from the connected APD for $50\mu\text{s}$. This is referred to as a microslice. The DCM internally stores 100 microslices and combines it into a millislice representing 5 ms worth of data. Once the DCM collects this amount of data it is transferred to the buffer farm. The buffer farm serves as a temporary storage location where 20-30 minutes worth of raw data can be stored. The exact size of this time window depends on the current activity and number of active triggers. This time window allows for decisions to be made on whether to keep or delete the captured data at an early stage. If the decision is made to keep a microslice, all coincident DCM microslices are

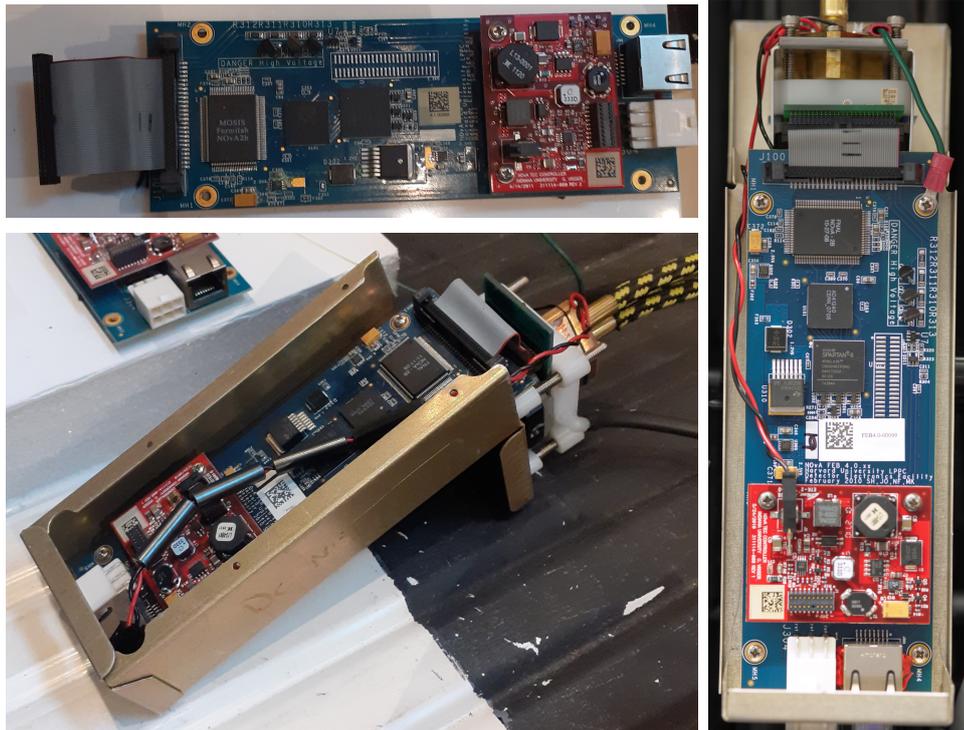


Figure 3.13: The front end board. The FEB without the APD and cooling connected is shown on the left, two views of a FEB installed in its casing with the APD and cooling connected are shown in the remaining two images. Right image courtesy of [85].

assembled into an event. This event is then moved to permanent storage. If the buffer farm was not used in this way all of the recorded data would have to be moved to permanent storage, which would overwhelm this system. Even if the permanent storage systems could handle the volume of the total recorded data, the bulk of it would not be relevant to the analysis and so would eventually be discarded at later stages of the analysis. The usage of automated triggers for early data selection guards against both of these undesirable situations.

To capture and store the interesting data from a physics perspective triggers are used in combination with the temporary storage capabilities of the buffer farm. The triggers designate which microslices are to be kept. When a trigger signal is received all microslices from the different DCMs are reorganised for permanent storage. Some triggers can also reserve microslices from previous timesteps up to the buffer maximum to be moved into permanent storage.

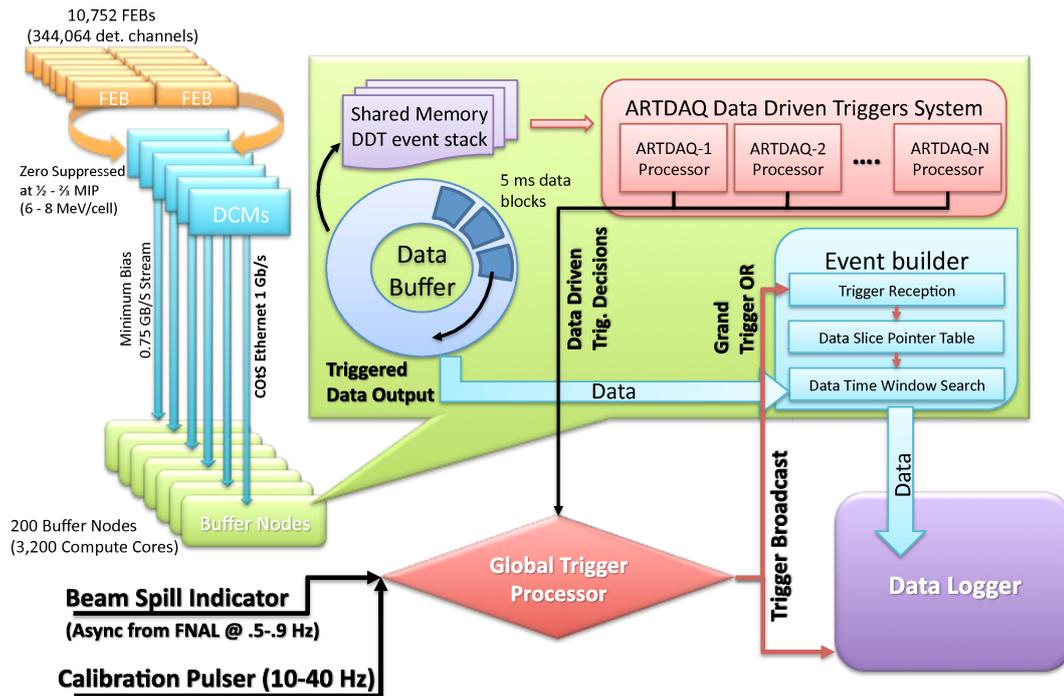


Figure 3.14: A schematic overview of the FD DAQ system.

3.5.2 Timing system

The DAQ system relies heavily on correct timing information for readout, online processing and eventual saving of raw data. In order to reliably synchronise all the different components of the DAQ system a dual timing chain system was developed for the NOvA detectors. In order to correctly process and readout for different triggers, the individual detectors need to be internally consistent with regards to timing as well as being synchronised between different detectors and Fermilab. A threshold of 10 ns is set to which all detector components of either the ND or FD need to be synced.

The timing system consists of an ensemble of Timing Distribution Units (TDUs). One of these is designated as the master unit and the others in the chain are referred to as slave units. However this designation is considered to be outdated and will be renamed to primary and secondary going forward. The primary TDU is connected to GPS which provides the primary time keeping. The secondary TDU each connect to 12 DCMs, split into 2 chains of 6 DCMs. At the end of each of these chains the timing signal is looped back onto itself

for the purposes of calibration. As the ND is located underground the primary TDU cannot be both connected to GPS and placed near the detector. It is located on the surface with a fibre-optic connection down the shaft into the underground detector area.

Time keeping starts at 00:00:00 on the 1st of January 2010 GMT. The internal clocks of most detector components run at a 64 MHz cycle and therefore 15.625 ns is the base time unit on the NOvA experiment. The primary TDU clock is synchronised to the GPS signal which runs at 10 MHz. A phase-locked loop (PLL) is used to achieve synchronization across frequencies. The timestamps are encoded in 56 bit registers which allows for a range of validity of 35.7 years. The 10 MHz clock is stable with a drift of less than 2 parts per billion per day if no external checks to the satellite signal can be performed. An additional safety check is provided at each detector in the form of a timing calibration reference (TCR) unit. The TCR unit has its own GPS antenna and receiver that can produce stable trigger outputs each second. A comparison between these trigger pulses and the primary TDU timing signal can be used to check for clock errors, desyncs or drift.

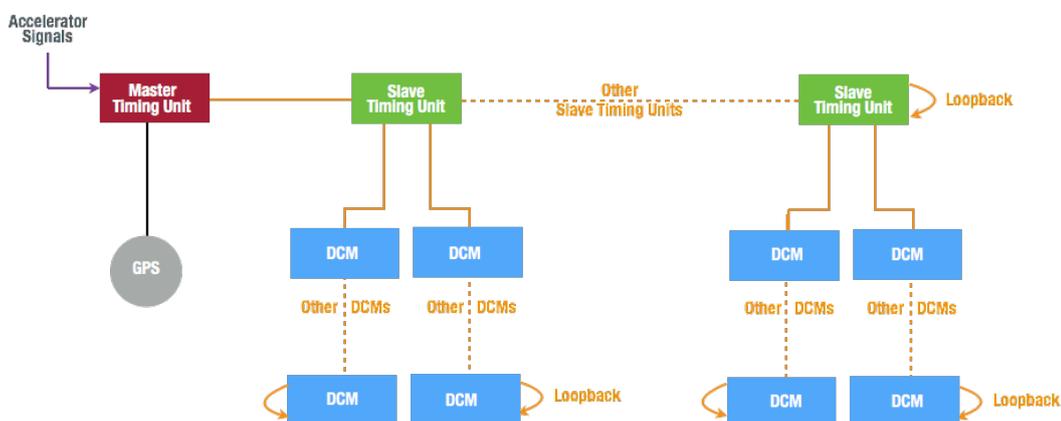


Figure 3.15: Schematic overview of the NOvA timing system.

3.5.2.1 Syncing

The individual components of the timing chains need to be synchronised. To that end the primary TDU can send out a sync signal down the chain of the secondary TDUs and connected DCMs. Using the loopbacks at the end of

the DCM chains and at the end of the TDU chain the delays of each of the copper wires connecting the individual components can be measured. These delays can be locally recorded for the TDUs but not for the DCMs. This results in an offline timing correction which is applied to the recorded data at the DCM level. FEBs are excluded from this timing delay measurement as the wires connecting them to the DCM have been measured precisely during the detector commissioning and do not exceed specific values. With the delay values measured a new future time value can be determined. Using another sync signal and the individual delay values all components can be primed and individually start their internal counters at the same new time.

3.5.2.2 Sampling and resolution

As the components of the timing chains are now in sync with each other readout of the raw data is possible. As mentioned earlier each FEB is connected to 32 cells. The FEB operates at 64 MHz but due to the MUXs the actual readout frequency is lower. For the FD this frequency is lowered to 2 MHz and for the ND it is lowered to 8 MHz. The APDs are read out with a rise time of 460 ns in the FD and 140 ns in the ND. The fall time is 7000 ns in the FD and 4500 ns in the ND. Two methods of APD signal readout can be used with the aforementioned DCS: single or multipoint timing. Before August 2014 single point timing was employed in the FD. With this method the current ADC value of a cell is compared to the previous value three ticks ago. If the current ADC value exceeds a threshold the difference in ADC values is recorded as the charge and the time of the TDC three ticks ago is recorded as the timestamp. The currently employed method is multipoint timing. Here all 4 ADC values are recorded and used to perform a pulse fit. This fit more accurately allows for the determination of the pulse magnitude and location and therefore provides a higher timing resolution.

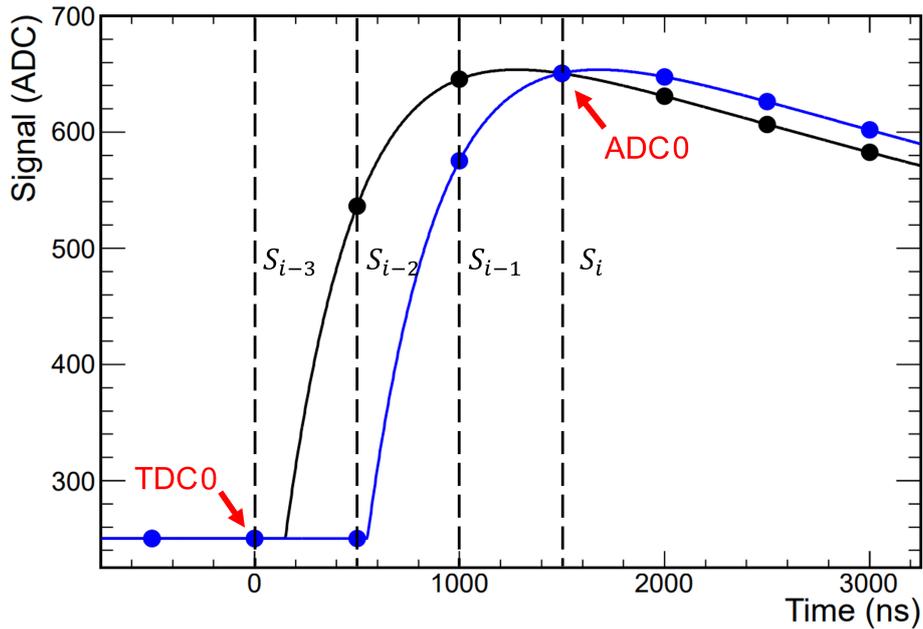


Figure 3.16: Illustration of the DCS algorithm. Multiple digitization samples (s_i, \dots, s_{i-n}) are recorded for two pulses shown in blue and black. As both pulses go over threshold with single point timing the ADC values at $s_i - s_{i-3}$ (ADC0) and the time at s_{i-3} (TDC0) would be recorded and be the equivalent for both pulses. Multipoint timing needs to be employed to enable discrimination between the two pulses. Adapted from [86].

3.5.3 Triggers

The signals that decide which data is discarded and which is saved are referred to as triggers. They trigger the data saving procedures. In NOvA there are three main types of triggers.

Signal triggers: Triggers which are based on signals external to the detectors.

Clock triggers: Triggers which are based specific recurring times.

Data driven triggers (DDT): Triggers which are based on information or activity from the detectors itself.

The following is a non exhaustive list of the triggers used in the NOvA DAQ system. These can be changed based on the needs of the experiment.

NuMI Trigger: The signal trigger which is used to record data around the beam spill of the neutrino beam originating at FNAL. The neutrinos are delivered in a $10\mu s$ window. The data is recorded $270\mu s$ before and after the beam spill window resulting in $550\mu s$ of total readout.

SNEWS Trigger: The signal trigger which reads out multiple seconds of data when a signal is received from the SuperNova Early Warning System (SNEWS)[87].

Cosmic Data Trigger: Clock based trigger which reads out data in $550\mu s$ windows at a frequency of 10 Hz in the FD and 1 Hz in the ND. This data primarily consist of cosmic muons and is used for cosmic background studies and the detector calibration.

DDEnergy: Data driven trigger which reads out a minimum of $50\mu s$ worth of data if the total deposited charge in the detector exceeds a set threshold. If this charge continuous to exceed the threshold the readout window is continually extended.

DDSupernova: Data driven trigger which continually searches for supernova neutrino candidates. This requires online reconstruction. If the number of candidates exceeds a set threshold $550\mu s$ of data will be read out.

Chapter 4

Simulation and Calibration

Physics experiments such as NOvA rely heavily on large amounts of data collected over long periods of time. To correctly understand and interpret that, experimental data simulations provide perhaps one of the most powerful tools. Simulation allows for the combined usage of current theories and models to make highly accurate predictions. In a complex experiment, such as the NOvA experiment, Monte-Carlo (MC) simulations are employed to bridge the gap between the theoretical understanding and the practical aspects of a real physics experiment. MC simulations use random sampling to produce numerical simulations. It is particularly effective in situations where there are many coupled degrees of freedom present or when the system's parameters have significant amounts of uncertainty associated with them.

4.1 NuMI Beamline simulation

The NOvA experiment simulation starts at the accelerator complex at FNAL with a simulation of the NuMI beam. The first step in the simulation chain is the modelling of the hadron production in the beam target; the FLUGG [88] and FLUKA [89] simulation packages were used for this purpose. In more recent analyses the GEANT4-based G4NuMI with the FTTP BERT and QGSP BERT hadronic simulation packages [90] are employed. The produced hadrons are then propagated through a GEANT4 [91] [92] simulation of the downstream NuMI beamline. This includes the target, the magnetic focussing horns and

the decay pipe. After this propagation is complete simulation files of the NuMI beam have been produced. Additional corrections to the output of this beamline simulation can be made to improve the prediction of the hadron production. This is done through reweighting of the neutrino flux with the Package to Predict the Flux (PPFX) [93]. This package incorporates external data from previous experiments with respect to the hadron-nucleus interactions.

The nominal neutrino flux is propagated to both the ND and FD for both horn current modes (forward and reverse). Furthermore for the FD multiple sets of flux files are produced. The nonswap files contain the nominal simulated flux present at the FD. The fluxswap files refer to files where all ν_μ and ν_e present in the beam have been exchanged for ν_e and ν_μ respectively. For the tauswap files all neutrinos in the beam have been replaced with ν_τ . Combinations of these fluxes enable the construction of any flux generated from oscillations caused by a specific set of oscillation parameters. E.g. usage of solely fluxswapped FD flux files is consistent with oscillation parameters resulting in a 100% oscillation probability for $\nu_\mu \rightarrow \nu_e$ and 0% survival probability for $\nu_\mu \rightarrow \nu_\mu$.

4.1.1 Neutrino interaction simulation

The next step of the simulation chain involves neutrino event generators. These are used to simulate how a given neutrino would interact with a target material. Knowledge of the nuclei of the materials present in the detectors and the areas surrounding them coupled with the simulated information of the neutrino fluxes at the detectors are given as input to the chosen neutrino event generator. Based on the internal event generator modelling of the physical processes that can occur when a neutrino interacts, the event generator outputs four-vectors of the outgoing particles after the primary interaction has occurred. Combining this output with the previously simulated information neutrino event rates and their interaction kinematics can be simulated.

This simulation process can be subdivided into several steps to be completed in order [94].

1. Simulation of the target nucleus. The basic requirement on this simu-

lation model is that it needs to describe the various energy-momentum distributions of the nucleons present in the target nucleus.

2. Modelling for the scattering processes which can occur at the nucleon level and modelling of the subsequent hadronization.
3. Modelling of the local nuclear medium effects on the hadrons produced in the various scattering processes.

There are a multitude of choices in parameters, models, approximations and external data to incorporate in the event generator. These choices are typically based on a consideration of validity versus model and computational complexity. An optimal decision is often unique to the circumstances and needs of a given experiment. The NOvA experiment is no exception to this consideration.

4.1.1.1 GENIE

The GENIE [95] [96] neutrino event generator is used in the NOvA experiment. The GENIE event generator is extensively used by many experiments. Most experiments hosted at FNAL employ GENIE. Its design has a focus on extensibility, modularity and flexibility. These core design elements are relied on by the NOvA experiment to select specific physics models and incorporate external datasets from previous experiments and external theory. Furthermore incorporation of ND data in the form of cross section measurements can be used to produce a NOvA specific reweighting [97] (sometimes referred to as a tune).

Version 2.12.2 of GENIE is used with the following model choices for simulations. The initial state of the neutrino interaction with the target nucleons is simulated with the Smith and Moniz global Relativistic Fermi Gas (RFG) model [98]. This model is further modified through the addition of short range nuclear correlation by inclusion of a high momentum tail for single nucleons in the Fermi momentum distribution [99]. For the QE interactions the Llewellyn Smith model is used [100]. For meson exchange current interactions

where the neutrino can interact with a pair of correlated nucleons an empirical version of the Valencia 2p2h model is used [101]. The base version of the model is tuned through reweighting. For resonant pion production and coherent charged current interactions, models produced by Rein and Sehgal are used, for RES [102] and for COH [103] enhanced with later improvements [104]. For the inelastic scattering interactions the Bodek-Yang formalism [105] is used with a custom hadronization model [106] implemented in PYTHIA [107]. The Final State Interactions (FSI) are simulated using the GENIE hN model. In addition to the choices for the various interaction types external advances in theories, approximations and experimental data is used to produce a NOvA specific tune [108].

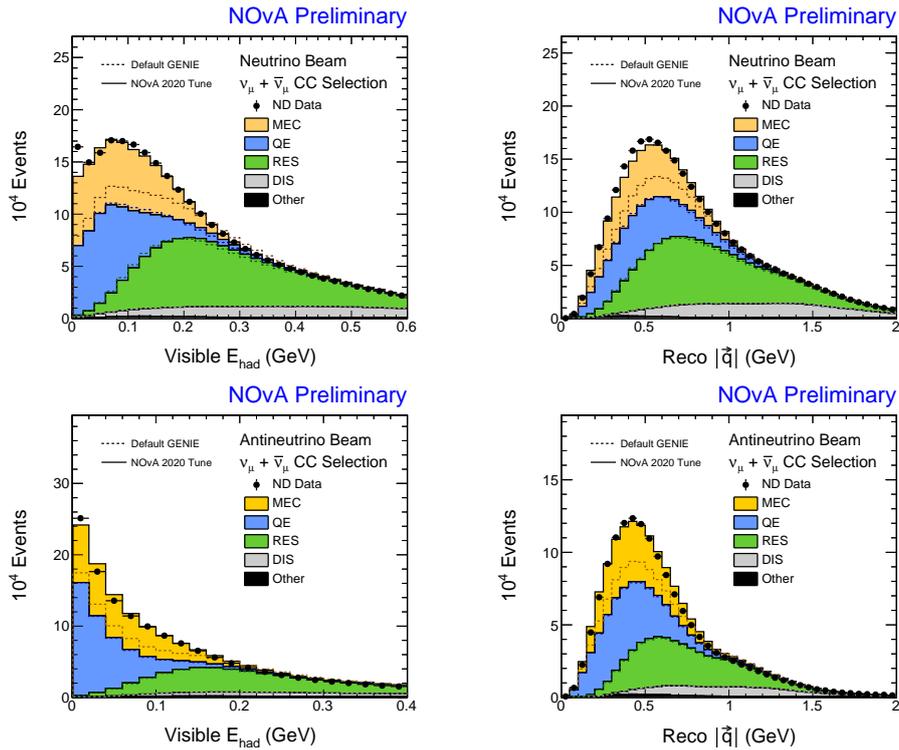


Figure 4.1: Data-MC comparisons highlighting the effects of the NOvA genie tune for the reconstructed visible hadronic energy and squared four-momentum transfer in the ND. The default genie tune is shown in the dashed lines. Events contributing to these comparisons pass the 2020 ν_μ -CC selection.

4.1.1.2 GiBUU

Some other experiments related to the NOvA experiment such as T2K are free to make different choices regarding the models, parameters and tunings they use in their simulations. There are also different frameworks which can be chosen as neutrino event generators. The Giessen Boltzmann-Uehling-Uhlenback (GiBUU) [109] event generator is one such alternative neutrino event generator. Whereas GENIE could be regarded as event generator constructed with an experimentalist's perspective, GiBUU would be the equivalent but with a theorist's approach. The GiBUU neutrino event generator doesn't fully rely on MC, instead it uses a semiclassical transport model. This model can be used to combine the effects of many different nuclear effects such as the local density approximation, the mean-field and Coulomb potentials, off-shell particle transport, 2p2h modelling and various spectral functions in the local environment. This allows a single theory-based transport model to describe a wide range of nuclear effects occurring between the transition of the initial state to the final state of the simulated interaction.

4.1.2 Cosmic ray simulation

Beyond the final state particles simulated from the neutrino interactions in the detectors, cosmic ray interactions are simulated separately. As the FD only has a small overburden to provide shielding from cosmic rays it observes a high rate of cosmic rays. While most of these can be filtered out through timing in the data taking and offline analysis techniques it still represent a significant source of background. Therefore the cosmic rays can be simulated separately and overlaid on the output of the neutrino event generator. The Cosmic-RaY shower generator (CRY) [110] is used for this purpose. The FD is assumed to be at sea level, whereas in actuality it is located 368 m above sealevel.

4.2 Detector simulation

The simulation data containing cosmic rays, neutrinos and their associated daughter particles form the input for the next step in the simulation chain.

All these particles need to be propagated through the physical material of the detectors. Detailed geometries describing the detectors have been produced with the GEANT4 package. Additionally for the ND the rock located in front of the detector is also simulated for a limited number of interactions. Muons from the neutrino interactions in this rock traverse a large rock volume which needs to be fully simulated in GEANT4. This process is demanding in terms of computational resources and thus only the neutrinos and their daughter particles depositing energy in the detector are saved. Splitting simulations with and without the rock singles allows the main simulations to be run with only the detector and detector hall in the GEANT4 geometry. The rock singles can be overlaid at a later stage.

4.2.1 GEANT4 detector geometry

Both detectors and their immediate surroundings have been reconstructed using GEANT4. This includes the shapes and locations for all the detector components and the materials from which they are constructed. For the ND the muon catcher, bookend, cavern and surrounding rock are included. The bookend refers to the supporting structure at the start and end of the detector. For the FD the pivoter and overburden are included. The pivoter is the machine which was used to tilt 31 plane blocks of the FD into place. It now acts as a bookend. The pivoter is approximated as a 15.68 m by 15.68 m by 0.049 m steel square plate. For both the ND and FD geometry imperfections resulting from construction have been included such as the tilting and staggering of the detector modules. The FD overburden is simulated as 10.2 cm of concrete and 15.2 cm of barite. The total mass of the ND is simulated to be 284.42 metric tons (includes the muon catcher) and the FD $138.43 \cdot 10^2$ metric tons [111]. The GEANT4 geometry simulations combined with the simulated particle fluxes can be used to determine the amount of energy each particle deposits in each part of the detector.

Detector	Material	Density $\left(\frac{\text{g}}{\text{cm}^3}\right)$
Far, Near	Scintillator	0.859
Far, Near	PVC	1.49
Far, Near	Glue	1.34
Far, Near	WLS Fibre	1.0457
Far (Pivoter)	Steel	7.87
Near (Muon Catcher)	Steel	7.87

Table 4.1: Materials used in the GEANT4 detector geometry simulations. Recreated from [111].

4.2.2 Photon generation and transport simulation

A component of GEANT4 can be used for the purpose of converting the deposited energy in the detectors to the equivalent amount of scintillation light. Conversion for general scintillators with a scintillation efficiency of L_0 is given through a simple linear relation for light yield (L) along the particle's path,

$$\frac{dL}{dx} = L_0 \frac{dE}{dx} \quad (4.1)$$

The organic liquid scintillator used in the NOvA detectors suffers from quenching effects which collectively can diminish the scintillation light produced. These losses increase at higher energy deposition levels. The Birks-Chou model gives correctional terms which can model these effects [112] [113],

$$\frac{dL}{dx} = \frac{L_0 \frac{dE}{dx}}{1 + k_B \frac{dE}{dx} + k_C \left(\frac{dE}{dx}\right)^2} \quad (4.2)$$

The values of the Birks' constant (k_B) and the Chou's constant (k_C) were determined from ND events containing exactly one proton and one muon track. This resulted in $k_B = 0.040 \frac{\text{g}}{\text{MeVcm}^2}$ and $k_C = -0.0005 \frac{\text{cm}^2}{\text{MeV}^2}$ for the first published analysis [111].

GEANT4 is capable of performing the next steps in the simulation chain. With the light simulated in the detector it travels to the WLS fibres and eventually to the APDs. This process occurs with similar parameters in each

of the detector cells. Repeating the individual capture and transport process in GEANT4 is computationally intensive and needlessly repetitive. For this reason templates are generated which can be used to parametrize the light collection and transport.

The photon collection by the WLS fibre is simulated using a ray tracing simulation. The input for this simulation are scintillation photons. A conversion of the charged particle's deposited energy to the scintillation light is therefore required before the photon collection simulation can be run. The initial conversion is an estimation of this conversion factor based on the material properties of the liquid scintillator. The design specification for the liquid scintillator oil is set at a light yield of at least 80% of the light yield of BC-517P scintillator. This scintillator has a light yield set at 21% of the light yield of anthracene with 2 photons per 100 eV deposited. Combining these approximate equivalences results in an initial estimate of $3360 \frac{\text{photons}}{\text{MeV}}$ for the liquid scintillator used in the NOvA detectors. This initial estimate is sufficient as a starting point to provide input to the ray tracing simulation and can be refined in subsequent iterations.

In the ray tracing simulation scintillator photons are allowed to move about the scintillator and the interior of the cell. When the path of a photon intersects with the WLS fibre, adsorption and reflectivity are used to determine if the photon gets reflected or adsorbed by the fibre. Reflectivity is determined by application of Fresnel's equations for the reflection of light at the interface of two different media. In this case the boundary is set between the mineral oil with a refractive index of 1.46 and the outer edge of the WLS fibre with a refractive index of 1.42. Furthermore a realistic scintillator spectrum is used which results in a reflectivity which varies with the average wavelength of the scintillator photons. This spectrum is weighted by the adsorption spectrum of the WLS fibre.

Incorporating this ray tracing simulation with the earlier GEANT4 simulation allows for a thorough simulation of scintillation light creation and

capture by the WLS fibre. Starting with the energy deposited by the charged particle in a specific location along the cell length, continuing by applying quenching corrections to calculate the visible energy, using the conversion factor to produce scintillation photons and finally propagating these photons into the WLS fibres. This results in a template for the mean number of captured photons as a function of position along the length of the detector cells.

Once the scintillation photons have been collected locally by the WLS fibre they can travel to the APD readout by two paths. Two possible pathways can be taken due to the loop in the fibre that is present at the far end of the detector cell. This travel is not loss-less and the average number of photons surviving is measured as a function of distance from the readout. These measurements are captured in attenuation curves. Initially bench tests were used to produce attenuation curves for the WLS fibres. While useful these proved to over-attenuate for longer distances resulting in a lower number of photons received at the readout. The calibration procedures of the first analysis were able to correct for this effect to the first order. Subsequently the quality of sets of WLS fibres has been taken into account and resulted in a number of fibre brightness groupings.

The adsorption of scintillation photons in the APDs is simulated using Poisson sampling which incorporates the APD quantum efficiency of 85%. Ray tracing from the collection point down the fibres combined with averaging over the possible paths which photons can use to travel to the readout inside the fibres can be used to determine the average travel time. Addition of the time from initial energy deposition to the scintillation photon collection by the fibre can be used to calculate the total travel time. Finally merging all these components results in the average number of photo-electron (PE) as a function of time for deposited energy in the detector cell.

4.2.3 Detector response

The final stage in the simulation chain is the simulation of electronic components resulting in the cell hit to be recorded. The number of PE s captured by the

APDs has an additional component which needs to be simulated. The APDs do not operate noiselessly and therefore have a larger variance in the number of *PEs* detected. While the distribution of the APD response which includes this larger variance has been determined, it is not easy to repeatedly sample from this distribution and so a model to approximate this distribution is generated. This model is based on the log-normal distribution. The log-normal distribution follows from the multiplicative product of many independent random positive variables. As each step in the amplification process is comprised of many random charge multiplications these can be accurately described by the log-normal distribution. The log-normal distribution employed here is defined as,

$$f(z) = e^{\ln(\mu) - \frac{\ln(1 + \frac{\sigma^2}{\mu^2})}{2} + z\sqrt{\ln(1 + \frac{\sigma^2}{\mu^2})}} \quad (4.3)$$

where the mean is given as μ , the standard deviation as σ and the samples from the unit Gaussian as z . For the modelling of the final APD response a hybrid approach is taken. For small numbers of *PEs* sampling from the theoretical distribution is feasible (< 250). For anything larger the Poisson sampled number of *PEs* is multiplied by sampling from the log-normal distribution with a μ of 1 and a σ of $\sqrt{\frac{F-1}{N_{PE}}}$ where F is the excess noise factor and N_{PE} the number of *PEs*.

The APD simulated signal from collected photons needs to be further propagated through the subsequent electronics in the FEB. This includes the ASIC, ADC and the FPGA. The ASIC is comprised primarily of a CR-RC circuit which performs pulse shaping on the signals outputted by the APD. To simulate this component the standard response of a CR-RC circuit for a unit charge pulse is used,

$$f(t) = \frac{t_F}{t_F - t_R} \left(e^{-\frac{(t-t_0)}{t_F}} - e^{-\frac{(t-t_0)}{t_R}} \right) \quad (4.4)$$

where t_R is the rise time, t_F is the fall time and t_0 is the start time of the APD *PE* collection. Noise present in the circuit is generated through addition of

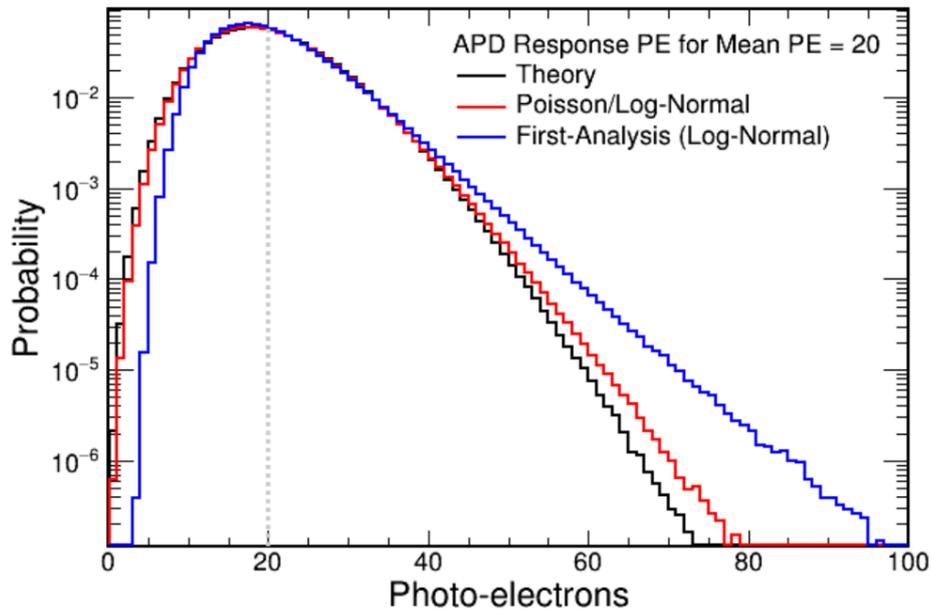


Figure 4.2: APD response distributions for three different excess noise models with an average N_{PE} .

Gaussian-Markov distributed noise to the pulse signal.

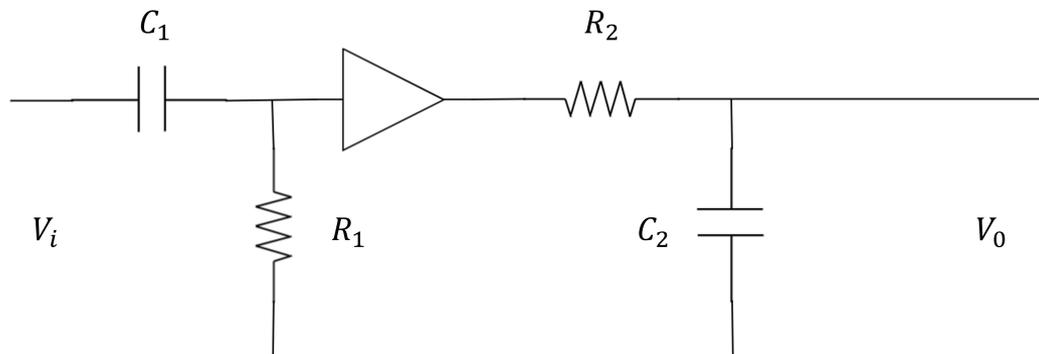


Figure 4.3: Circuit design of CR-RC circuit, recreated from [114].

For conversion of analogue PE s to digitized signals the ADC component of the FEB is simulated. Saturation of the ADC is taken into account in the conversion factor as it is defined as $\frac{ADC_{MaxPE}}{ADC_{Sat}}$. The ADC value at saturation ADC_{Sat} is given as 4095, the number of PE required to saturate the ADC

ADC_{MaxPE} can be determined. For this purpose the ADC value is defined as,

$$ADC = N_{PE} f(t) \frac{M_{APD}}{M_{FEB}} \quad (4.5)$$

where M_{APD} and M_{FEB} represent the gain factors of the APD and the FEB respectively. Rearranging equations (4.4) and (4.5) yields the following expression for the maximum number of PE s which saturate the ADC,

$$ADC_{MaxPE} = 4095 \frac{M_{FEB}}{M_{APD}} \frac{t_F - t_R}{t_F \left(\frac{t_F}{t_R}\right)^{-\frac{t_R}{t_F - t_R}} - t_F \left(\frac{t_F}{t_R}\right)^{-\frac{t_F}{t_F - t_R}}}. \quad (4.6)$$

An additional effect with regards to the APDs is also simulated. This effect is known as APD sag and stems from the physical layout of the pixels on the APD. When a sufficiently sized deposition is made on one of the APD pixels other pixels experience a simultaneous sag in the voltage level which drives the charge multiplication. All pixels across the APD share a common voltage source and hence the occurrence in a single pixel affects all connected pixels. The drop in driving voltage can be picked up by the ASIC and when the pulse shaping is applied the effects of the momentary drop can be elongated in time. As the dual correlated sampling (DCS) method relies on the previous time-steps' ADC values, the resulting slow rise back to a nominal driving voltage can be confused for a real energy deposition.

The FPGA simulation follows the same method of applying DCS as described in this section. Noise in this component of the FEB is simulated by summing the total noise from two Markov-Gaussian chains. These represent highly correlated voltage and current sources in the electronics. The distribution from which the voltage noise component is drawn is defined as,

$$V_i = \frac{t_{read}}{t_R + t_{read}} Z_i + \frac{t_R}{t_R + t_{read}} V_{i-1} \quad (4.7)$$

And the current noise components is defined as,

$$C_i = \frac{t_{read}}{t_F + t_{read}} Z_i + \frac{t_F}{t_F + t_{read}} C_{i-1} \quad (4.8)$$

Combining the two components into a total noise via,

$$FPGA_{noise} = aV_i + bC_i \quad (4.9)$$

The index i denotes the current digitization sample, DCS uses multiple ADC samples from different slices. The time between digitizations is given by t_{read} and Z_i denotes the random number drawn from a standard Gaussian distribution. The factors a, b are directly related to the calculated noise charge for the current and voltage sources. These have been approximated by the following relations,

$$ENC_V = \frac{e_n C_d}{q_e} \frac{1}{\sqrt{2t_R}} \quad (4.10)$$

$$ENC_C = \sqrt{\frac{FM_{APD} I_L (t_S - t_R)}{q_e}} \quad (4.11)$$

where e_n is the voltage noise density, C_d the detector capacitance, q_e the charge of an electron, t_S the DCS sampling time interval and I_L the leakage current. The values of I_L and e_n are tuned to average values which minimize a difference fit between simulation and real data. Further individual variation of the noise levels can be determined from real data.

Simulating this noise for all cells is costly and wasteful in terms of computing resources. When a cell records no real energy deposits noise is extracted from real data and added instead of simulating it. During one of the time windows when cosmics are recorded a significant number of cells only record real background noise as they are not in the path of the incident cosmic ray muon.

With this final addition of noise the simulation chain is now complete up to the stage of having simulated hits in the detectors and reading those hits

out through the electronics.

4.3 Calibration

The real data recorded from the detector and the simulated MC data are now stored as a collection of disconnected hits in the detector. Calibration and reconstruction can be applied to these hits to retrieve quantities relevant to the physics analyses. The calibration is focussed on the conversion of ADC values of hits to an approximation of actual energy deposited by charged particles in the detector constituting a hit. This is critical in eventually accurately measuring the energy of the measured neutrino. The shape and magnitude of the neutrino energy spectrum is utilized to probe the various relevant oscillation parameters.

In the NOvA experiment this conversion of ADC values to energy is referred to as calibration. To achieve accurate calibration of the detectors cosmic muons are exploited. There are several advantages to using cosmic muons for this purpose.

1. The cosmic muon rate is high for both detectors resulting in an abundance of energy deposits originating from cosmic muons in the detectors.
2. The experimental signature of the muon is distinct from that of most other particles and can therefore be more easily be identified.
3. The energy loss of a muon traversing a material is theoretically well understood.
4. The subset of muons which stop in the detector provide a standard candle.

The average amount of energy a muon loses, or equivalently deposits in the material, is given by the Bethe-Bloch equation [116],

$$\left\langle -\frac{dE}{dx} \right\rangle = Kz^2 \frac{Z}{A} \frac{1}{2\beta^2} \left[\ln \left(\frac{2m_e c^2 \beta^2 \gamma^2 W_{max}}{I^2} \right) - 2\beta^2 - \delta(\gamma\beta) \right] \quad (4.12)$$

where K is equal to $4\pi N_A r_e^2 m_e c^2$ with N_A being Avogadro's number, r_e the electron radius and $m_e c^2$ the rest energy of an electron. Furthermore Z is

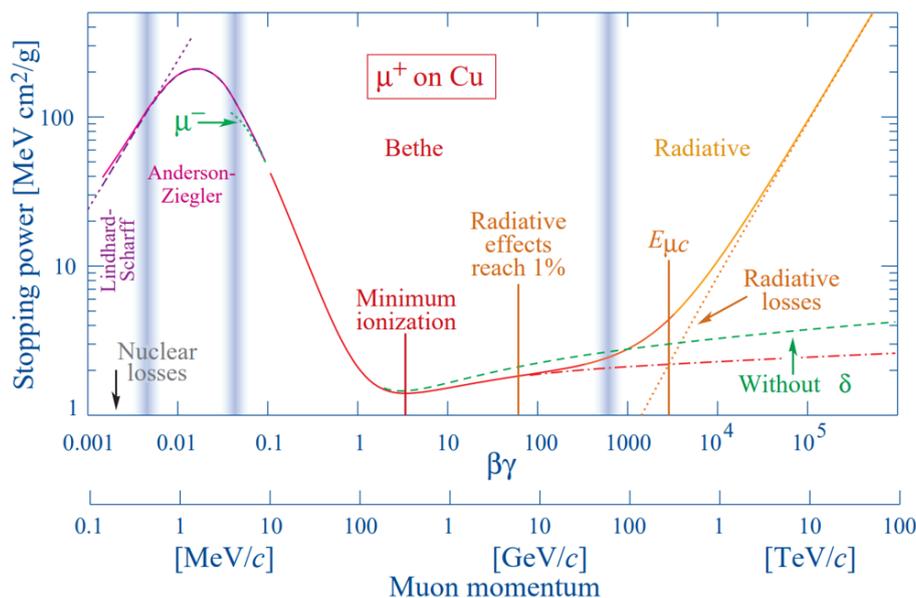


Figure 4.4: The stopping power of positive muons in terms of $\langle -\frac{dE}{dx} \rangle$ in pure copper as a function of $\beta\gamma$. Courtesy of [115].

the atomic number and A the mass number of the material. I is the mean excitation energy and δ the density effect correction to ionisation energy loss in the material. W_{max} is defined as the maximum energy transfer possible in a single collision. z, β, γ are defined as the charge, velocity and gamma factor for the muon traversing the material. Equation (4.12) is accurate to the few percent level for the range of $0.1 < \beta\gamma < 1000$. It is plotted in Figure 4.4.

In order to be used for calibration purposes the recorded muon tracks need to be of sufficient quality. To this end several cuts are placed on the recorded data to ensure the overall quality of the data, for example a containment requirement and a tri-cell cut. Additionally basic clustering and reconstruction techniques are used to find the muon tracks. These techniques are described in more detail in chapter 5. The tri-cell cut requires the muon track to traverse both the cell above and below that of the current hit. This requirement ensures that the track does not pass directly through the corner of the cell and reduces the amount of poorly reconstructed tracks in the dataset. It also allows for a trigonometric calculation of the track pathlength in a single cell. A schematic

illustration of the tri-cell criteria is provided in Figure 4.5. The position of a recorded hit within a cell can be determined from the reconstructed track and is expressed as W , the mean position along the length of a cell. W is equal to zero in the center of the cell. At the readout W attains its maximal value and at the far end of the cell it reaches its minimal value.

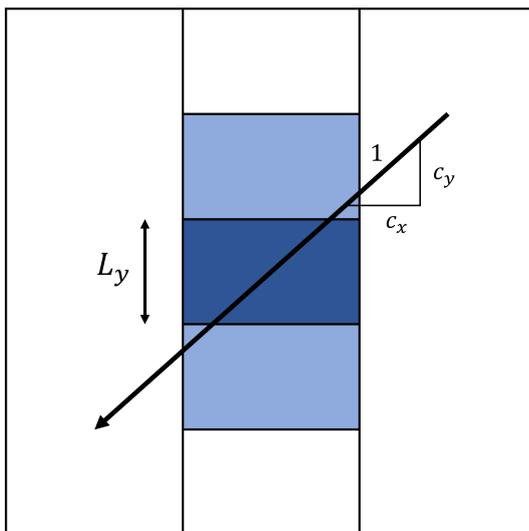


Figure 4.5: An illustration of the tri-cell calibration criteria. Hits are used in the calibration if there are energy deposits recorded in the cells directly above and below the main cell. Inspired by [117].

4.3.1 Relative calibration

The first stage of calibration is the relative calibration. Here recorded ADC values are corrected to $PECorr$, a corrected version of the PE values normalized across the detector. It is referred to as relative as its main purpose is to correct for any differences between individual detector cells. This relies on the assumption that the energy spectrum of cosmic muons is uniform across the detector. While the muon flux is approximately uniform across an empty volume equivalent to the detector volume, there are some effects which can introduce variation across the detector in terms of the energy spectrum. Nonetheless it is feasible to use cosmic muons as their energy deposited per unit distance is relatively uniform over the minimum ionizing particle energy range in which the cosmic muons fall. Separate effects introducing variations can be corrected

for individually.

In addition to the alterations of the cosmic muon spectrum there are several effects which can induce differences between channels. To correct for these effects the selected cosmic muon sample is expressed in terms of PE/cm vs W , that is approximated photo-electrons divided by the pathlength in the cell versus the hit's location in the cell. This distribution falls with distance from the readout as fewer photons survive the transport through the length of the fibre. This attenuation along the cell length is unique for each cell as the fibre placement and quality are not perfectly equivalent between cells. Additionally the light yield of the scintillator and the quantum efficiency of a given APD pixel can vary between cells.

Beyond these differences thresholding and shielding are two additional effects for which it is necessary to correct. The thresholding effect results from the WLS fibre attenuation at long lengths. Due to the large amount of attenuation present at the far end of a FD cell lower energy hits do not result in any PE s being recorded in the APDs while higher energy hits can survive to the readout. This introduces a scaling energy bias across the cell length. The boundary below which hits are unable to be recorded is referred to as the threshold. Shielding, or detector self-shielding, is a geometric effect due to the muons having traversed a larger amount of mass when they deposit energy in the lower parts of the detector. This introduces a vertical variation to the uniformity of the cosmic muon spectrum. Lower energy muons are thus more likely to be stopped by the traversed mass and therefore a bias for higher energy hits occurs in these regions. This effect is larger at the FD due to the larger pathlength difference between the top and bottom areas of the detector.

These two effects are corrected for first. This is done through a comparison between MC truth and reconstructed information,

$$T = \frac{PE}{E_{MIP}} \frac{E_{true}}{\lambda} \quad (4.13)$$

where E_{true} is the true energy deposited in the cell, E_{MIP} the expected energy

deposited by a minimum ionizing particle (the cosmic muons) based on their pathlength, PE the simulated photon electrons recorded at the readout and λ the number of simulated photons seen at the readout without these effect present. This correction is applied to every cell in the detectors individually for data, for MC it is applied for cells in a single representative plane.

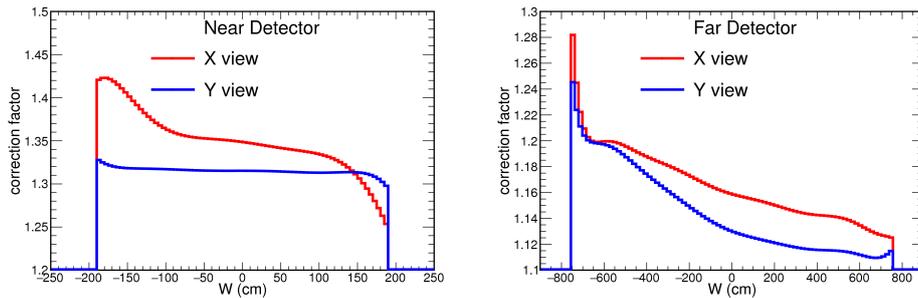


Figure 4.6: Average thresholding and shielding correction factor for the ND (left) and FD (right) as a function of the cell internal coordinate W .

In Figure 4.6 the average thresholding and shielding correction factors are shown. Thresholding contributions to the correction factor increase with distance from the readout, which is located at the positive W . The shielding contributions are a primarily vertical effect and are larger for the FD as it has considerably more mass. Its contributions are therefore expected to be largest for the lower regions of each respective detector. For the vertical cell orientation of the X-view this corresponds to an increased correction factor of the negative W region. The shielding effects on the horizontal cells of the Y-view are reduced and cancelled out due to the averaging over the vertical stacking of the cells in this view.

After the thresholding and shielding correction is applied the cosmic muon data should have a uniform energy spectrum across the detector. Figure 4.7 shows the PE/cm vs W after the thresholding and shielding correction has been applied. Cell by cell variations can then be corrected for by applying a double exponential fit to the corrected PE/cm vs W plots. The double exponential form is selected due to the two pathways the light can travel along to the WLS

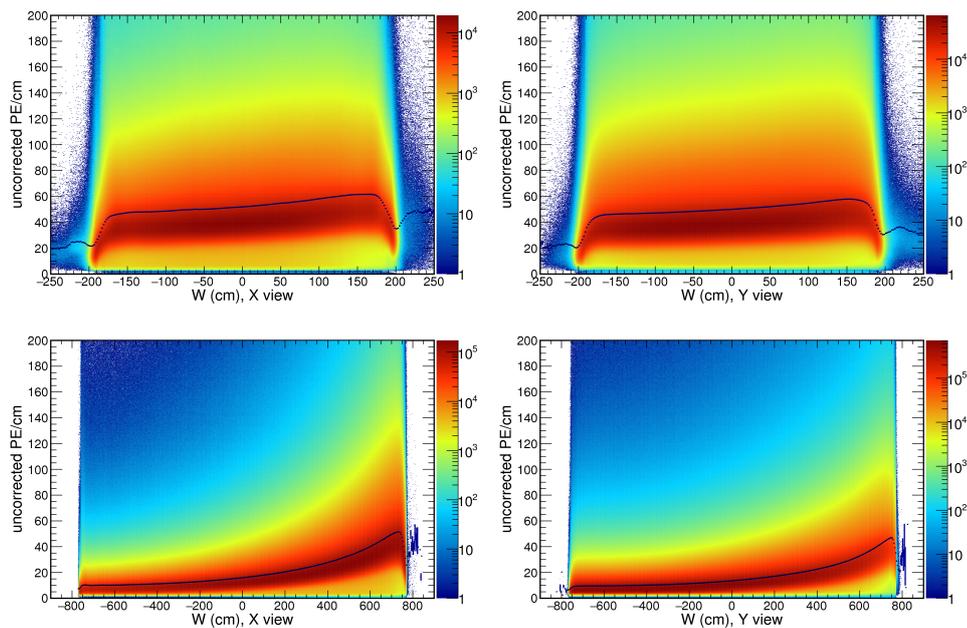


Figure 4.7: Uncorrected photo-electrons per pathlength as a function of W for the two views and detectors. The thresholding and shielding correction has been applied, the attenuation correction has not and therefore it is denoted as uncorrected. The ND is shown on top and the FD is shown on the bottom.

fibre to reach the APDs. It takes the following form,

$$y = C + A \left(e^{\frac{x}{X_A}} + e^{-\frac{\frac{3L}{2} + x}{X_A}} \right) \quad (4.14)$$

where x is the distance from the cell center ($W = 0$), X_A is the attenuation length of the fibre and L is the length of the cell itself. A, C are left as free fit parameters. The double exponential fit is confined to the central areas of the PE/cm vs W curves. This avoids any large deviations from effects at the ends of the cells impacting the overall fit. These can include deviation caused by the difference in reflectivity of the cell walls and fibre placement at the readout.

Additional deviations in the central fit area are possible. The prevailing theory of the root cause is the fibre placement in the cell itself, the two parts of the fibre can be close together or stuck to the cell wall. The different fibre positions can result in light level variations. To correct for these local effects in the attenuation fit the residuals from the double exponential fit are fit with the

LOcally WEighted Scatter plot Smoothing (LOWESS) method. This method produces a fit curve from the tri-cubed weighted mean of the deviations. The weighting is constrained to a local length scale of 30 cm and results in 20 fit points across the fit window. These points are interpolated and applied as a correction to the double exponential fit. Examples of the attenuation fit on individual cells of both detectors are shown in Figure 4.8.

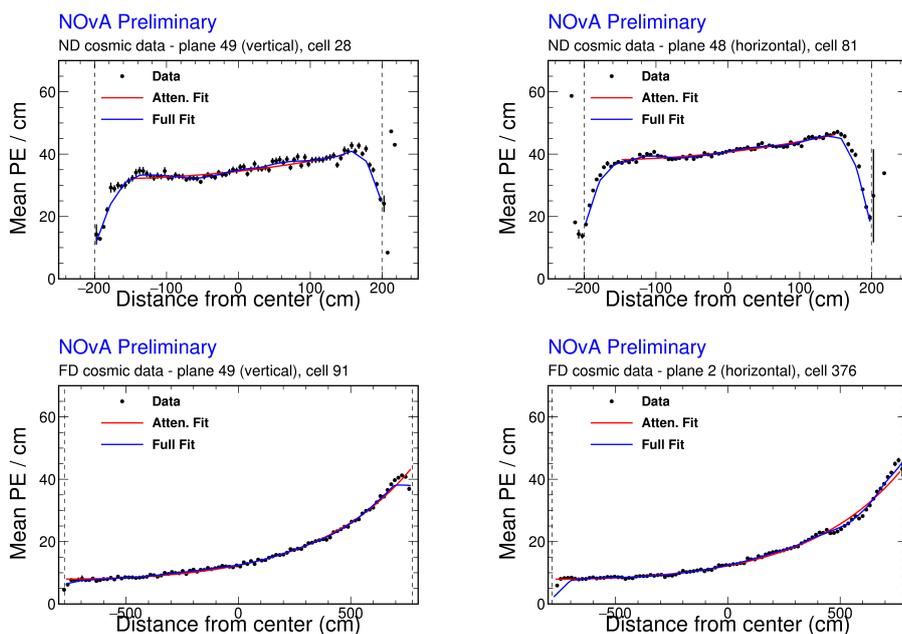


Figure 4.8: Attenuation fits for the two views and detectors. Various cells with a representative quality of fit have been selected. The red curve is the double exponential fit and the blue is the fit with the LOWESS method applied. The dashed lines highlight the fit region with the cell ends excluded.

Applying the resulting fit as a correction factor results in a new unit of corrected equivalent photo electrons called *PEC_{corr}*. These are normalized to the cell response at $W = 0$ and are uniform across the detector. The effectiveness of the relative calibration procedure is tested through the examination of the mean reconstructed to true energy ratio curves which are shown in Figure 4.9 for the second analysis. A small offset is found for the Y-view which will be resolved by the subsequent absolute part of the calibration. This concludes the relative part of the calibration procedure.

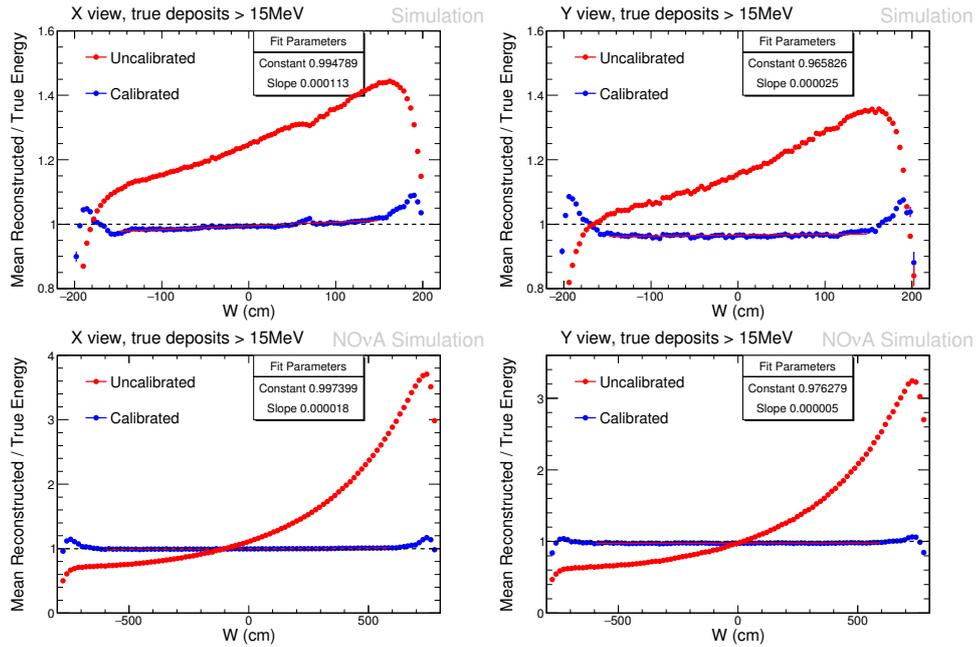


Figure 4.9: Profiles of the mean reconstructed to true energy ratios as a function of W in the two views for both detectors. The red curve shows the ratio before any calibration is applied. The blue curve shows it after the full relative calibration is applied. The data represented in these plots originates from the second NOvA analysis.

4.3.2 Absolute calibration

Now that the cell response has been normalized across the detector, the resulting *PECorr* values need to be set to an absolute energy scale. For this purpose a different cut is applied to the cosmic muon dataset. Whereas primarily through-going muons are used for the relative calibration, stopping muons are used for the absolute calibration. This choice is motivated by considerations originating in the description of the energy loss of muons traversing matter.

In Figure 4.4 the Bethe-Bloch equation (4.12) is plotted. From this plot the behaviour of a muon travelling through matter can be deduced. For a muon to fully travel through the detector its energy must be greater than the energy loss during traversal. This places those muons in the linear region to the right of the minimum ionization point. This justifies the earlier assumption made in the relative calibration with respect to the uniformity of mean energy loss across that range of the muon energy spectrum. For the stopping muons

used in the absolute calibration the mean energy loss decreases as the particle slows down until it reaches the minimum ionizing particle(MIP) point. When it slows down further and loses more of its total energy the mean energy loss increases until it fully stops and interacts. This results in a very distinct profile of energy deposition along the track length, especially the high values at the endpoint. Measuring back from the endpoint along the track profile allows for the identification of the MIP region. This region is characterized by a linear mean energy deposition. This value can be determined from equation (4.12) coupled with information on the detector materials. This results in a direct connection between the PE_{corr} recorded in cells along the stopping muon track in the MIP region and the actual deposited energy in that region.

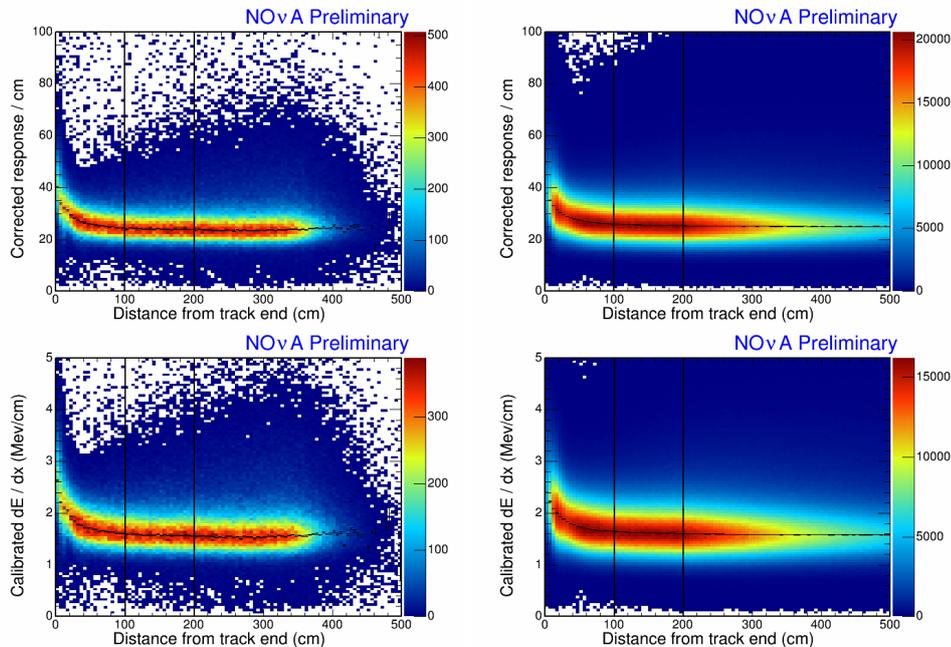


Figure 4.10: The deposited energy of stopping muons as a function of distance from the track endpoint for both detectors. The top plots show the deposited energy in terms of corrected photo-electrons as a function of pathlength post relative calibration. Hits contained within the track window 100 – 200 cm are considered to be in the MIP region and are used to connect the PE_{corr} to the correct energy scale. Applying the resulting normalization yields the reconstructed energy for cell hits as can be seen in the bottom plots.

Using the material properties of the detector components and the liquid scintillator mixture the estimated value of the average energy deposition in the

detector for MIPs is,

$$\frac{dE}{dx} = 1.7915 \pm 0.035 \text{ MeV/cm} \quad (4.15)$$

For the final conversion Muon Energy Units (MEU) are defined. These units represent both the mean detector response and the simulated energy deposition of a stopping muon hit. The ratio of these units is used to set the conversion scale between *PECCorr* and energy (GeV),

$$\text{Calorimetric energy scale} = \frac{\text{Mean}(\text{MeV/cm})_{\text{MCTruth}}}{\text{Mean}(\text{PECCorr/cm})_{\text{Reco}}} = \frac{\text{MEU}_{\text{Truth}}}{\text{MEU}_{\text{Reco}}} \quad (4.16)$$

where the mean of the MeV/cm distribution for stopping muon hits in the MIP region from MC simulations is used to compare to the means of the *PECCorr/cm* distributions in both real data and MC after reconstruction is applied (up to the *PECCorr* stage).

The full calibration procedure is repeated periodically over the course of the experiment. It is expected that the individual detector components age and degrade over time. The aging rates of various components are not equivalent. Degradation of the scintillator and WLS fibres are expected to gradually reduce the brightness of the detectors. This calibration procedure can detect these changes and correct for them.

4.3.3 Transposer

Running the calibration procedure requires a significant amount of computational resources. The frequency of the calibration procedure reruns can be balanced against this cost, resulting in a infrequent but periodic total rerun. There are two main factors contributing to the large cost of running the calibration. The calibration contains many components and calculations which have to be performed for every cell in the detectors. There are approximately 364,256 cells in both detectors combined. Secondly the data which is used as input for the calibration is stored in an event by event time ordered format. A consequence of this format is that running over a high number of input cosmic

muon data files is required to accumulate sufficient statistics for the cell by cell calibration. For reference a schematic overview of the modules required to complete just the relative calibration is provided in Figure 4.11.

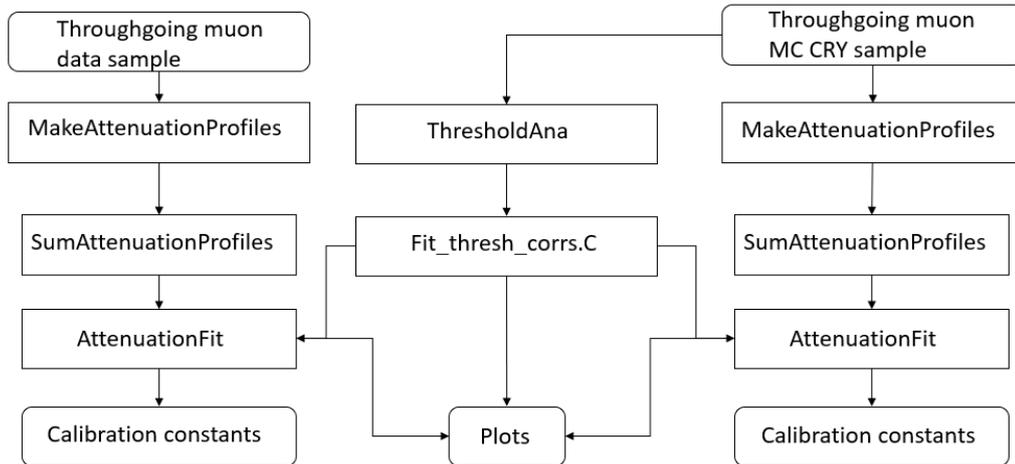


Figure 4.11: Flow diagram showing the order of modules required to be run for the completion of the relative calibration.

For smaller scale studies of calibration procedure, the detector degradation over time or partial (single cells or small groups of cells) detector calibration the balance of cost versus benefit is not equivalent to that of the full calibration. Particularly incremental studies that have to be run multiple times scale poorly due to the repeated runs over the large input datasets.

To resolve this issue the PCHitTransposer module has been created. This is an ART event processor framework module [118]. It is meant to be run before calibration or any studies occurring on Pre-Calibrated (PC) hits in the recorded or simulated events. The primary function of this module is to run over time ordered data and convert it into a geometry ordered dataset, moving the event from an ordering in (runs, subruns, events) into detector (plane, cell, event) while retaining their meta-data. Any subsequent calibration studies can directly access all events over time for the requested cells or parts of the detectors.

The module has been optimized to run on the FermiGrid with its memory footprint remaining below the threshold for a single grid slot. The module can be included in the large scale production efforts and thus only needs to be run once after the initial dataset creation. This moves the large cost of rerunning over time ordered data for each study in a non-efficient/non-optimized manner to a single upfront optimized cost.

Additionally the deployment of this module does not alter the information stored in the events nor their structure, this means that it can be seamlessly integrated in the existing codebase and does not require significant alterations to the code employed for these studies.

Chapter 5

Reconstruction

To render data hits in the ND and FD in useful units which are required for further physics analyses the calibration procedure needs to be applied. As seen in the previous chapter this procedure relies on information which is reconstructed from basic event information, such as the W coordinate of a recorded hit along its trajectory. Reconstruction information forms the basis for most physics analyses as well, where typically each analysis defines different relevant signal and background sources so accurate identification and separation of each source of activity is key to reach the sensitivity needed for the world leading measurements done by the NOvA collaboration. This is achieved through the sequential application of several reconstruction algorithms.

5.1 Clustering

The first step in this process is to separate individual hits into clusters that stem from a single source of activity such as a neutrino or cosmic ray muon event. Accurate clustering will result in identification of noise hits as well.

5.1.1 4DSlicer

For the initial analyses the 4DSlicer is employed to perform the initial clustering of hits produced by a specific event. It relies on the Density-Based Spatial Clustering for Applications with Noise (DBSCAN) clustering algorithm. DBSCAN relies on the number of neighbouring hits to define clusters and separate signal hits from noise. A scoring function is used with a threshold distance to

define whether two hits are neighbours or not. When a hit has a number of neighbouring hits above the threshold number in the previously defined region it is considered to be a core point. If a hit fails to clear the threshold on its neighbours but one of the neighbours is a core point then it is considered to be a border point. Failure to meet one or both of these conditions will result in the hit to be assigned a status of being noise. In Figure 5.1 the DBSCAN algorithm is further illustrated.

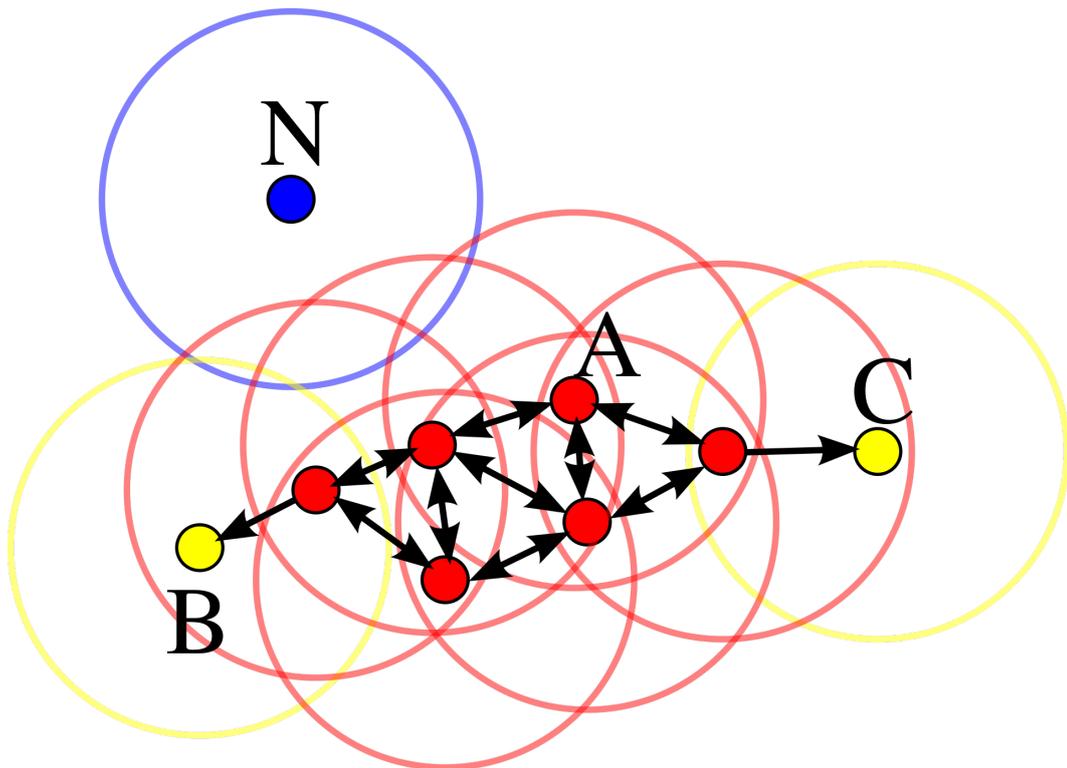


Figure 5.1: Diagram illustrating clustering with the DBSCAN algorithm. Here the minimum points are set to four and the distance ϵ for each individual point is drawn. Using these the points are grouped into core (red), border (yellow) and noise (blue) hits. Courtesy of [119].

After all hits have been assigned a status, the cluster centers are constructed from connected core points. These clusters extend until border points have been reached. The noise hits are excluded from the clusters. Additionally a post-clustering method is implemented to check for incorrectly split clusters. A border can be incorrectly drawn through a cluster when some of the neighbouring hits fall just outside of the threshold region. To correct for this

an additional threshold is defined which is slightly further and results in the definition of weak neighbours. If a border is incorrectly splitting a cluster it is expected that a sufficiently high number of weak neighbours are present. If this condition is met these smaller clusters can subsequently be merged.

To define the magnitude of the neighbouring region a score function is constructed,

$$\varepsilon = \left(\frac{\Delta T - \left| \frac{\Delta \vec{r}}{c} \right|}{T_{res}} \right)^2 + \left(\frac{\Delta Z}{D_{pen}} \right)^2 + \left(\frac{\Delta XY}{D_{pen}} \right)^2 + \left(\frac{PE_{pen}}{PE} \right)^5 \quad (5.1)$$

Events which produce hits that should be clustered together consist of relativistic particles. As their speed is typically high their hits are separated by light-like intervals in 4D space. The first term ensures that hits respect the causality conditions imposed by light-like intervals. The second and third terms penalize spatial separation of a magnitude which is considered to be too high. The final term penalizes hits which consist of a low amount of PE , as these are more likely to originate from noise. In equation (5.1) T_{res} is the timing resolution between the two hits whose distance score is being calculated. D_{pen} is a distance penalty term, PE_{pen} is the PE penalty term. In this equation PE refers not to the PE amount of a single hit but rather to the sum of the PE of the two hits being considered. The PE term is raised to a higher power to compensate for the drop-off in the noise PE spectrum. Both D_{pen} and PE_{pen} are tuned for each detector. Given that the cell direction in subsequent planes has an alternating orientation, hits can be correlated in one view or in both views. For hits present in one view $\Delta \vec{r}$ is the two dimensional distance between hits. For hits in different views $\Delta \vec{r}$ is one dimensional, ΔXY is set to zero and the D_{pen} is used with a different smaller value.

The 4DSlicer is tuned with ND data and a full breakdown of its parameters and their chosen values are motivated and evaluated in [120].

5.1.2 TDSlicer

The successor to the 4DSlicer is the TDSlicer. The change in slicing algorithm has been made to resolve some failures of the 4DSlicer. The major issue lies in the incorrect application of cluster merging. This can occur when there are hits present between large clusters that have a separate interaction origin. These could occur by statistical fluctuations resulting in a randomly higher local density of noise hits. Additionally these could be outliers of either cluster which are expected to be present. When these hits are suitably located between the clusters and are of sufficient quantity, they can act as a connection resulting in an incorrect merging of the clusters.

The TDSlicer employs a different clustering algorithm to the 4DSlicer which results in a more aggressive clustering. This reduces the number of incorrectly merged clusters at the cost of potentially incorrectly splitting clusters. In the case where a bridge could be formed between clusters and TDSlicer splits the cluster containing the bridge into two clusters, the hits which constitute the bridge will be associated to the individual clusters. Some fraction of these hits will be misgrouped by the Slicer. Overall it has been found that the cost of misgrouping these hits does not significantly impact the energy estimation and reconstruction of the event [121].

The TDSlicer starts with a search for local minima in the density of hits. Centroids are identified when these local minima are found. Clusters in dual 3D space are found based on the identified centroids. Depending on the view these clusters are located in (x, z, t) or (y, z, t) space. The 3D clusters can then be mapped to the full 4D (x, y, z, t) space.

Identification of the centroids is done through a centroid-finding algorithm [122]. This algorithm looks for maxima in the time density distributions of the hits. Initially the local point density distribution is determined,

$$\rho_i = \sum_j e^{-d_{ij}^2/\tau^2} \quad (5.2)$$

The point density distribution is calculated based on the euclidean distance d_{ij} between hits and a scale parameter τ . τ is determined independently for each detector. Given that the 3D space is defined by two spatial and one time dimension the separation distance is,

$$d_{ij} = \frac{||dt| - \frac{R}{c}|}{\tau} \quad (5.3)$$

where the separation in time is dt and in space is R .

Besides the point density an isolation score is calculated for each point,

$$\delta_i = \min_{j|\rho_j > \rho_i} (d_{ij}) \quad (5.4)$$

The isolation score evaluates the distance between a point and its closest neighbour with a higher point density. With these two values, the density and isolation scores, cluster centroids can be identified. They have high values in both scores which pass set thresholds [121]. Additionally the hits constituting the cluster are identified by applying a time-based constraint, ensuring a time separation of less than 10τ from a candidate cluster hit to the centroid hit. An example of this clustering technique is shown in Figure 5.2. The full 3D view based clusters are created through the application of Prim's algorithm [123] which constructs the cluster based on the next nearest hit to the cluster's edge. Repeated addition of close by hits continues until a threshold on the separation is reached.

Once all possible 3D (x, z, t) and (y, z, t) clusters have been constructed view matching is applied for the purpose of combination into the full 4D clusters (also referred to as slices). This view matching is achieved by comparing the average (z, t) values of the 3D clusters in the opposite views. Once an optimal match is identified the two 3D clusters are combined into the 4D slice and removed from the collection of 3D clusters. This matching is repeated until a threshold for the separation is reached after which the remaining 3D clusters are considered noise. The specific distance and threshold values are set to different

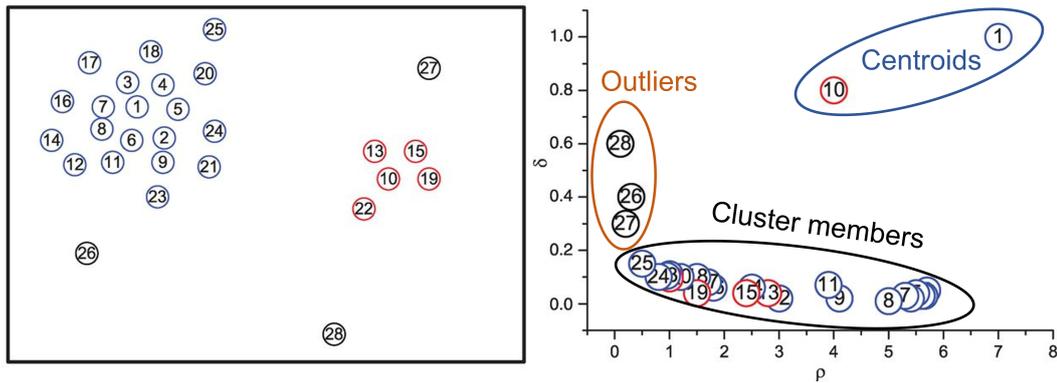


Figure 5.2: Demonstration of the centroid clustering algorithm. Distribution of points in 2D space to be clustered shown on the left. The resulting decision graph is shown on the right which is used to separate the cluster centroids from the noise hits and cluster members. Adapted from [124].

values in the two detectors [121]. The produced slices from the 4DSlicer and TDSlicer for different events can now be compared, one such example is given in Figure 5.3.

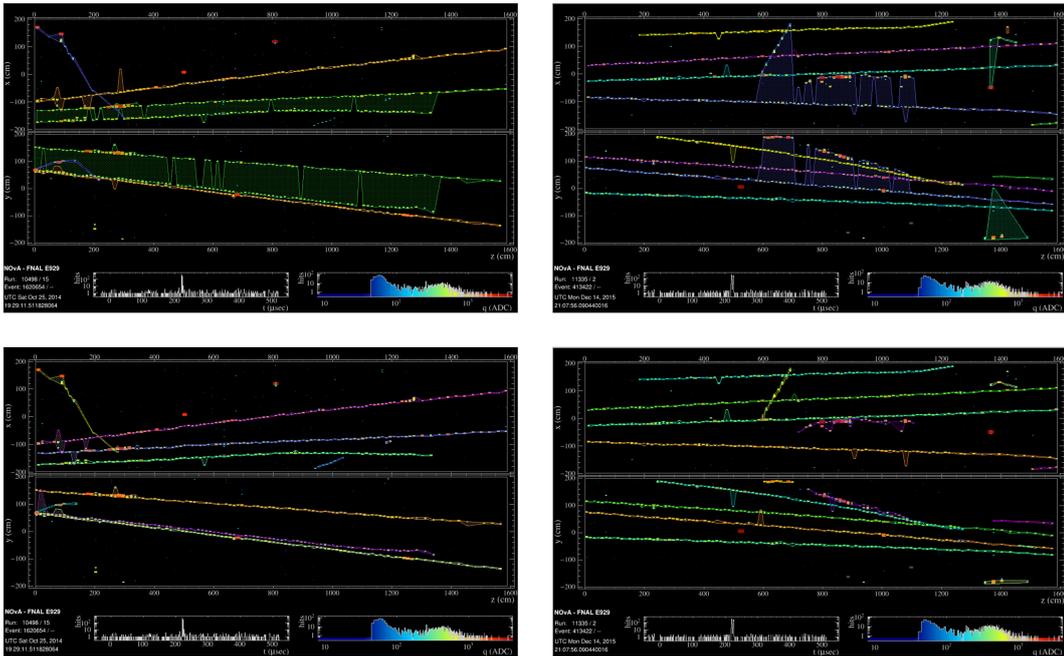


Figure 5.3: Comparison of the 4DSlicer (top) and TDSlicer (bottom). These events have been selected to highlight the described 4DSlicer failure modes. Adapted from [125].

The quality of the produced slices is measured by two metrics, the completeness (also referred to as efficiency) and the purity. These are both evaluated

on MC simulated data where the truth information of a particle traversing part of the detector is available. The completeness is then evaluated by taking the ratio of the amount of energy present in the reconstructed slice over the total energy in the event. This yields a measure for the fraction of a particle's energy being included in the slice.

$$\text{Completeness} = \frac{\text{Energy contained in the slice deposited by the particle}}{\text{Total energy deposited by the particle}} \quad (5.5)$$

The purity is evaluated by examination of the ratio of the amount of energy present in the slice originating from the event over the total energy contained in the slice. Thus yielding a measure for the amount of energy originating from other sources contained in the slice.

$$\text{Purity} = \frac{\text{Energy contained in the slice deposited by the particle}}{\text{Total energy contained in the slice}} \quad (5.6)$$

Using these metrics the performance of both the 4DSlicer and the TDSlicer can be compared. The 4DSlicer achieves a mean completeness of 0.964(0.930) and a mean purity of 0.986(0.984) for FD (ND) MC simulated data. The TDSlicer achieves a mean completeness of 0.970(0.849) and a mean purity of 0.987(0.994) for FD (ND) MC data. While the ND completeness of the TDSlicer is below that of the 4DSlicer further circumstances have to be taken into account. The number of good slices (purity > 0.9 and efficiency > 0.9) produced by the TDSlicer is higher and scales better with increasing intensity. This can be seen in Figure 5.4. A higher beam intensity results in a higher likelihood of overlapping clusters or bridging hits between clusters. These conditions result in the poorer performance and scaling behaviour of the 4DSlicer when compared to the TDSlicer.

5.2 Track reconstruction

The hits identified and clustered into slices by the TDSlicer originate from charged particles traversing parts of the detectors. Further reconstruction

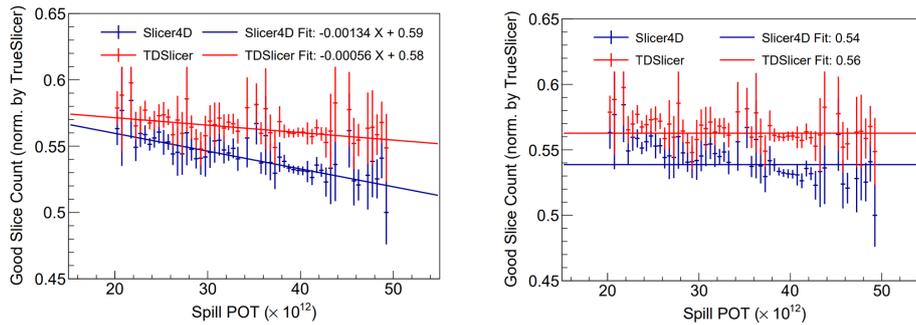


Figure 5.4: Performance comparison of the 4DSlicer and TDSlicer. The fraction of good slices produced by the two slicers are shown as a function of intensity on the left. The TDSlicer performs better in higher intensity environments where pileup is typical. Averaging over the entire range of slices returns the mean good slice count per event (right). Adapted from [126].

applied to these slices produces interpretable features of the underlying physics interactions. Determination of charged particle tracks contained in slices is the next step after the production of slices. As a charged particle moves in 4D space (x, y, z, t) two sets of tracks are expected in the lower dimensional 3D spaces representing each detector view. To reconstruct these lines a modified multipoint Hough transform is employed. The Hough transform is applied to each view independently and starts by constructing lines fit to each pair of points in the slice. These lines are parametrized in polar coordinates,

$$\rho = x \cos \phi + y \sin \phi \quad (5.7)$$

where (x, y) are Cartesian coordinates of hits in a given view of the detector (XZ or YZ), ρ is the closest distance to the origin, and ϕ is the angle between ρ and the positive x-axis. Each pair of hits can thus be parametrized by a single set of values (ρ, ϕ) in the Hough space. Populating the parameter space with values through examination of every possible combination of hits in a slice is resource intensive. Therefore limits are placed on which combination should be used for this purpose. If hits have a large separation in either XZ or YZ they are not considered. The threshold for this limit is tuned to a value of $\sqrt{15000}$ cm. Beyond reducing the computational load this also reduces the

transform's tendency to overweight far apart hit pairs as the separation results in relatively small uncertainty in ϕ . An additional limit is placed on point pairs which have similar X or Y in XZ or YZ respectively. This is done to prevent the creation of horizontal lines which do not match the data. The limit prevents the pairwise matching of hits that are closer together than $0.25\sqrt{15000}$ cm.

After the parameter space has been filled through the combination of hits with these limits in place, Gaussian smoothing is applied. The 3-by-3 weighting of the (ρ, ϕ) space can be applied with minimal computational costs due to the sparse nature of the parameter space. The improved accuracy of produced lines outweighs the computational costs associated with the Gaussian smoothing.

The lines best representing the hits in the data can now be extracted from the parameter space. This is done by searching for peaks in the binned parameter space. The peaks are identified as being above a calculated threshold value. This value is calculated from addition of the average bin height and the standard deviation of that average multiplied by a tuneable parameter. The threshold value is unique to each view. To improve the accuracy of created lines, deviation from the bin center is allowed through a correction to (ρ, ϕ) . To that end a weighted average is taken of a 7-by-7 bin square centered on the peak bin. The weights for this averaging are determined from the bin content and its separation to the peak bin.

If all bins above threshold in the parameter space are taken to be representative of lines in the data the procedure described here results in an overestimation of lines. To remedy this the Hough transform procedure is applied multiple times. After each application only the highest peak is selected and the hit pairs contributing to the peak are removed from the slice. The line constructed from these hits is added to a list. The Hough transform procedure is then applied to the remaining hits in the slice, extracting the subsequent lines. This is repeated until no more peaks are above threshold or a maximum number of lines present in the list is reached (10). Using the list of previous lines, new lines are checked to test if they are too similar to the previously

extracted lines (within 15 cm, 0.02 rad). If this is the case the hits associated to the secondary line are removed from the slice. The result of applying the multi-hough lines reconstruction on an event is shown in Figure 5.5.

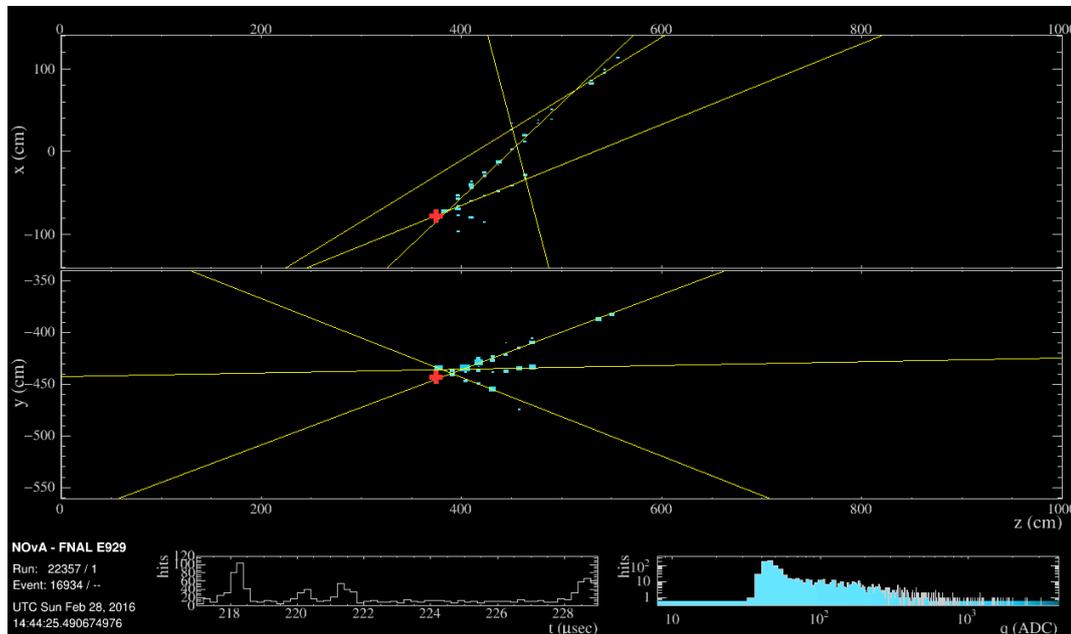


Figure 5.5: Example FD slice with elastic arms and multi-hough lines reconstruction applied. The red cross indicates the location of the elasticarms vertex and the golden lines are the reconstructed hough lines.

5.2.1 Elastic arms vertex reconstruction

Using the lines produced by the Multi-Hough transform representing the data hits a vertex of the interaction which produced them can be determined. For this purpose the Elastic Arms vertex fitter is employed. The fitter starts with an energy-cost function which is minimized for perfectly reconstructed vertexes. The seeds for the minimization are the lines provided by the application of the Multi-Hough transform. For A arms (lines) and H detector hits the energy-cost function is given as,

$$E = \sum_{h=1}^H \sum_{a=1}^A V_{ha} M_{ha} + \lambda \sum_{h=1}^H \left(\sum_{a=1}^A V_{ha} - 1 \right)^2 + \frac{2}{\lambda_v} \sum_{a=1}^A D_a \quad (5.8)$$

This function can be decomposed into individual terms which when minimized combine into an optimized vertex reconstruction. The first term penalizes any

divergence between the arm(line) and a given hit. In it M_{ha} is a measure of the distance from hit h and arm a , V_{ha} is strength of association. The second term penalizes noise hits not associated to a given arm. The third term penalizes a disconnect between the proposed vertex location and the start points of the arms. D_a is the distance between the suggested vertex location and the initial hit on an arm. Both λ terms control the relative strength of these penalization factors.

M_{ha} is determined from the shortest distance of hit h to arm a in either XZ or YZ depending on the view,

$$M_{ha} = \left(\frac{d_{ha}}{\sigma_h} \right)^2 \quad (5.9)$$

It is normalized by the spatial resolution of the detector σ_h . If a hit is located behind the start of the arm a modified form is used as an additional penalty factor,

$$M_{ha} = \begin{cases} \left(\frac{d_{hv}}{\sigma_h} \right)^2 & \text{for } \frac{d_{hv}}{\sigma_h} \leq 1 \\ \left(\frac{d_{hv}}{\sigma_h} \right)^4 & \text{for } \frac{d_{hv}}{\sigma_h} > 1 \end{cases} \quad (5.10)$$

where d_{hv} is the shortest distance between the hit h and the vertex point v .

V_{ha} is the probability that a given hit h is associated correctly to arm a . It is computed from the individual likelihoods that a hit is noise or associated to an arm,

$$V_{ha} = \frac{e^{-\beta M_{ha}}}{e^{\beta \lambda} + \sum_{b=1}^A e^{\beta M_{hb}}} \quad (5.11)$$

The association strength is normalized to the likelihood of a hit belonging to any arm and the remaining likelihood is noise. The λ term is the distance at which a hit has an equal probability to be associated to the track or be labelled as noise. The term β is representative of the influence of the arm when measured in distance to the examined hits.

The minimization of the energy-cost function starts at low values of β . The large distance of influence results in many hits being associated to the arms and

relatively low numbers of noise hits. This carries forward into the surface of the energy-cost function being smoother and less detailed. In terms of optimization the local minima are smoothed out while the global minimum largely retains its shape. Iteratively increasing β results in more realistic associations of hits to the arms versus noise and minimizes the fit parameters determining the vertex. The achieved quality of the vertex reconstruction using the elastic arms method is shown in Figure 5.6.

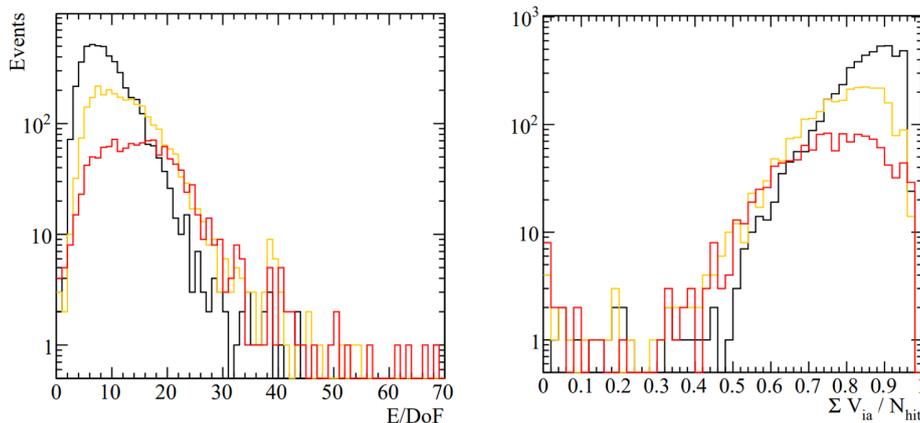


Figure 5.6: Goodness of fit for the elastic arms vertex reconstruction method. The energy cost function per degree of freedom(left) and the ratio of hits associated with the arms to the total number of hits(right). These histograms have been produced using v_{μ}^{CC} events. Three subdivision based on the quality of the reconstruction are shown, good fits in black ($x \leq 10$ cm), bad fits in orange ($10 < x \leq 50$ cm) and incorrect fits in red ($x > 50$ cm). These distances are measured from the reconstructed location to the true vertex location.

5.2.2 Fuzzy-K vertex reconstruction

The next step in the reconstruction chain uses the rough linear features extracted by the Multi-Hough transform and the vertex location produced by the Elastic Arms method. With this information a clustering algorithm can be used to produce a list of prongs associated with each vertex. A prong is a cluster of hits associated to a candidate particle which is part of the full interaction. The algorithm chosen to perform this clustering is a modified version of the Fuzzy K-Means (FKM) clustering algorithm [127]. FKM places a set number of cluster centers k such that the distance to the total number of hits is minimized.

It includes a degree of fuzziness which translated to the possibility of a hit being associated to multiple clusters. Modification of the FKM algorithm is required due to two concerns when applied to the reconstruction task at hand. The first being the number of cluster centers k , which is not known beforehand as the number of prongs can vary. Secondly the applied normalization requires association of all noise hits and outliers to one or more clusters. To resolve these two issues a Possibilistic Fuzzy Clustering Algorithm (PFCA) [128] is used. The employed variation of PFCA drops the normalization requirement contained in the classic FKM algorithm and is capable of resolving the number of clusters by itself [129].

The PFCA is not without issues, such as a large variation in final clusters based on the chosen initial seeds and a tendency to produce overlapping clusters centered at the same location. These issues can be mitigated by setting a minimum threshold on the number of clusters to be located and reparametrizing the space in which the clusters are to be found. As most hits in the slice are assumed to originate from the reconstructed vertex location the Cartesian space of a single detector view can be shifted to a circular space with its origin located at the vertex position. The hits associated to the vertex in the slice are then used to populate a density matrix which is binned along the angle θ . 360 bins span the space from $-\pi$ to π where the 0 angle is collinear with the positive Z direction (along the beampath). This results in the following expression for the density matrix,

$$w_k = \sum_{h=1}^H e^{-\left(\frac{\theta_k - \theta_h}{\sigma_h}\right)^2} \quad (5.12)$$

where θ_k is given by,

$$\theta_k = -\pi + \frac{\pi k}{180}, \quad 0 \leq k < 360 \quad (5.13)$$

Seed angles are determined from the highest bins in the density matrix.

With a list of seed angles hits can be associated to specific clusters. In the

circular space defined previously the distances between the cluster centers(a) and the individual hits(x) are calculated,

$$d_{ij} = \left(\frac{x_j - a_j}{\sigma_j} \right), \quad -\pi \leq (x_j - a_j) \leq \pi \quad (5.14)$$

With this distance the cluster membership, a measure for the strength of the association of a hit to a cluster, can be calculated,

$$\mu_{ij} = e^{-\frac{m d_{ij} \sqrt{a}}{\beta}} \quad (5.15)$$

where m refers to the amount of fuzziness and β is a normalization term for the degree of separation present in the data. In this application, β is a measure of uncertainty in the angular distribution between a cluster center and the hits. a is the number of determined clusters present. A threshold is placed on the cluster membership to prevent undefined values in further calculations due to divisions by zero. Based on the distribution of cluster membership and the positions of contained hits the cluster centers can be iteratively updated. This process continues until the shift in the cluster centers falls below a threshold of 10^{-7} radians.

Once a full list of cluster centers has been determined a check is performed to remove any possible duplicate clusters. The seed leading to the duplicate is marked as bad and a replacement cluster is found from the density matrix. If one is found the clustering is repeated, otherwise the total number of clusters is reduced by one and the unclustered hits are distributed to the remaining clusters. Finally all hits are checked to be contained in good clusters that meet the previously set threshold criteria. Hits that fall short are grouped and are used as input to a new density matrix. New seeds and eventually clusters are produced from this matrix and are checked for duplicates. This step is repeated until all hits are contained in good clusters or the maximum threshold for the cluster number has been reached.

With the number of clusters and the locations of the cluster centers

defined in terms of the angular distribution the final clustering can be done. The objective score function for FKM is modified to reflect the loss of the normalization constraint described earlier,

$$J_{PFCA}(\mu, a) = \sum_{c=1}^C \sum_{h=1}^H \mu_{ch}^m \|x_h - a_c\|^2 + \frac{\beta}{m^2 \sqrt{C}} \sum_{c=1}^C \sum_{h=1}^H (\mu_{ch}^m \log \mu_{ch}^m - \mu_{ch}^m) \quad (5.16)$$

where the C is the total number of clusters and a is the cluster center. The modification is incorporated through the addition of the second term which penalizes cluster solutions which do not assign any hits to the clusters. Equation (5.16) can be simplified through substitution of equation (5.15) resulting in,

$$J_{PFCA}(\mu, a) = -\frac{\beta}{m^2 \sqrt{C}} \sum_{c=1}^C \sum_{h=1}^H \mu_{ch}^m \quad (5.17)$$

Minimization of this score function results in the final list of 2D clusters of hits associated with a vertex.

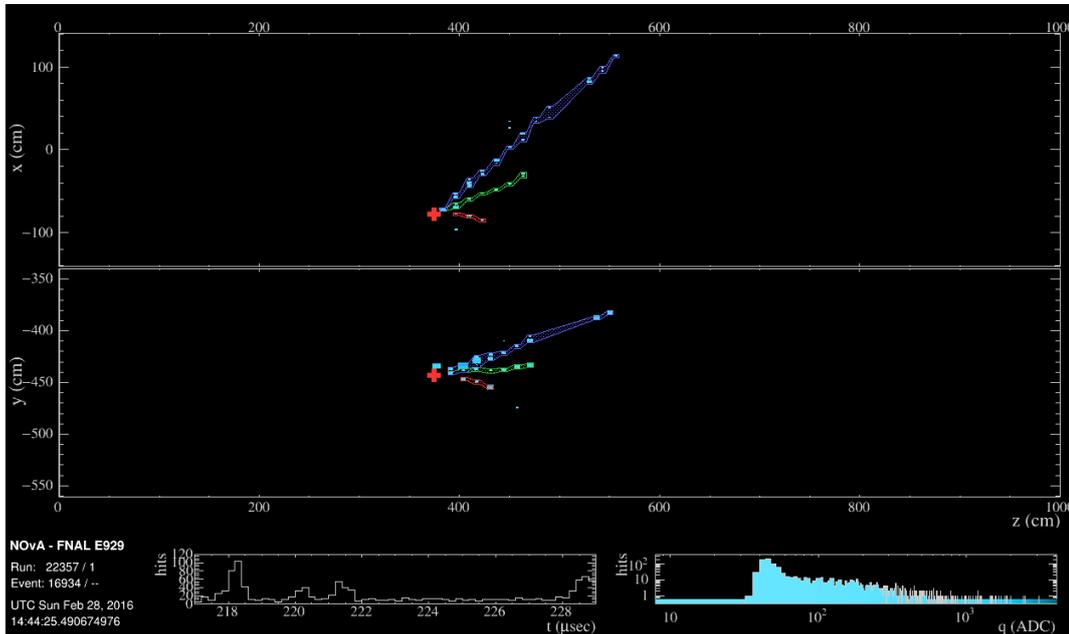


Figure 5.7: The event from 5.5 reconstructed using FuzzyK. The blue, green and red regions highlight the different regions corresponding to the individual prongs identified by FuzzyK.

The event of Figure 5.5 can be reconstructed using the FuzzyK reconstruc-

tion method and is shown in Figure 5.7. A measure of the performance of the FuzzyK method is shown in Figure 5.8.

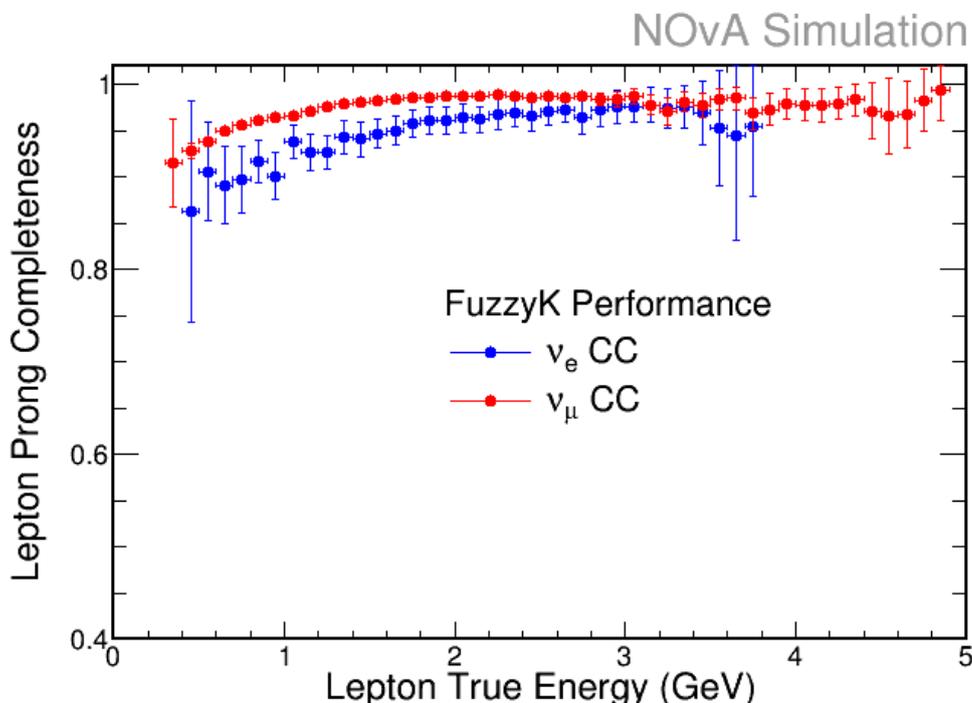


Figure 5.8: Completeness of FuzzyK produced prongs as a measure of performance. The ratio of the visible energy associated to the prong over the total true visible energy is shown for the ν_e and ν_μ FD MC events.

5.2.3 Prong matching

With the 2D prongs for each view a matching procedure is implemented to determine which two prongs should be combined together into a 3D prong. The primary criterion for this matching is the distribution of energy along the full length of the prong. These should be similar between the views as the encountered planes along the particle's trajectory alternate in their orientation. The matching begins with the production of a matrix containing all possible combinations of prongs between the two views. A combination of prongs is required to have endpoints which match to within a plane. Additionally if one of the prongs consists of hits in a single plane its paired prong can only contain hits in the planes directly adjacent to it. For each combination meeting these requirements present in the matching matrix a similarity score is calculated

based on the cumulative energy distribution at a given pathlength of the two prongs. The similarity score is calculated through the Kuiper test statistic which is examined in further detail in section 7.4.4.7. A variation in the pathlength between -12 cm and 12 cm in 0.5 cm increments is used to account for discontinuities introduced by the alternating plane structure in the detector geometry. The lowest value of the Kuiper test statistic between the two prongs across this range of pathlengths is taken and used for the matching. Once all combination in the matching matrix have been evaluated in this manner, the entry with the lowest value is matched and removed from the matrix. This is repeated until either all prongs are matched or most prongs are matched and the remainder has been marked and retained as an unmatched 2D prong. Figure 5.9 demonstrates this on a specific event.

5.3 Particle tracking

The produced 3D prongs are collections of hits originating from a charged particle traversing a part of the detector. Recorded information for each of those hits allows for the reconstruction of the most likely path the particle has taken when doing so. In NOvA multiple tracking algorithms are employed to perform this reconstruction task. The primary ones are the KalmanTrack and BreakPointFitter (BPF). Both of these are applied simultaneously as they use different methods and assumptions to solve the same reconstruction problem.

In the NOvA detectors the primary mode of energy loss for charged particles is through ionization. Due to the absence of any large curvature-inducing fields, such as a strong magnetic field¹, the general trajectory of charged particles in the detectors will be approximately linear. Small deviations to this linear track due to ionization can be approximated by multiple small angle scatterings,

$$\theta_0 = \frac{13.6 \text{ MeV}}{p\beta c} z_c \sqrt{\frac{s}{X_L}} \left(1 + 0.0038 \ln \left(\frac{s}{X_L} \right) \right) \quad (5.18)$$

where p , βc and z_c are the momentum, velocity and charge of the charged

¹Ignoring the Earth's magnetic field.

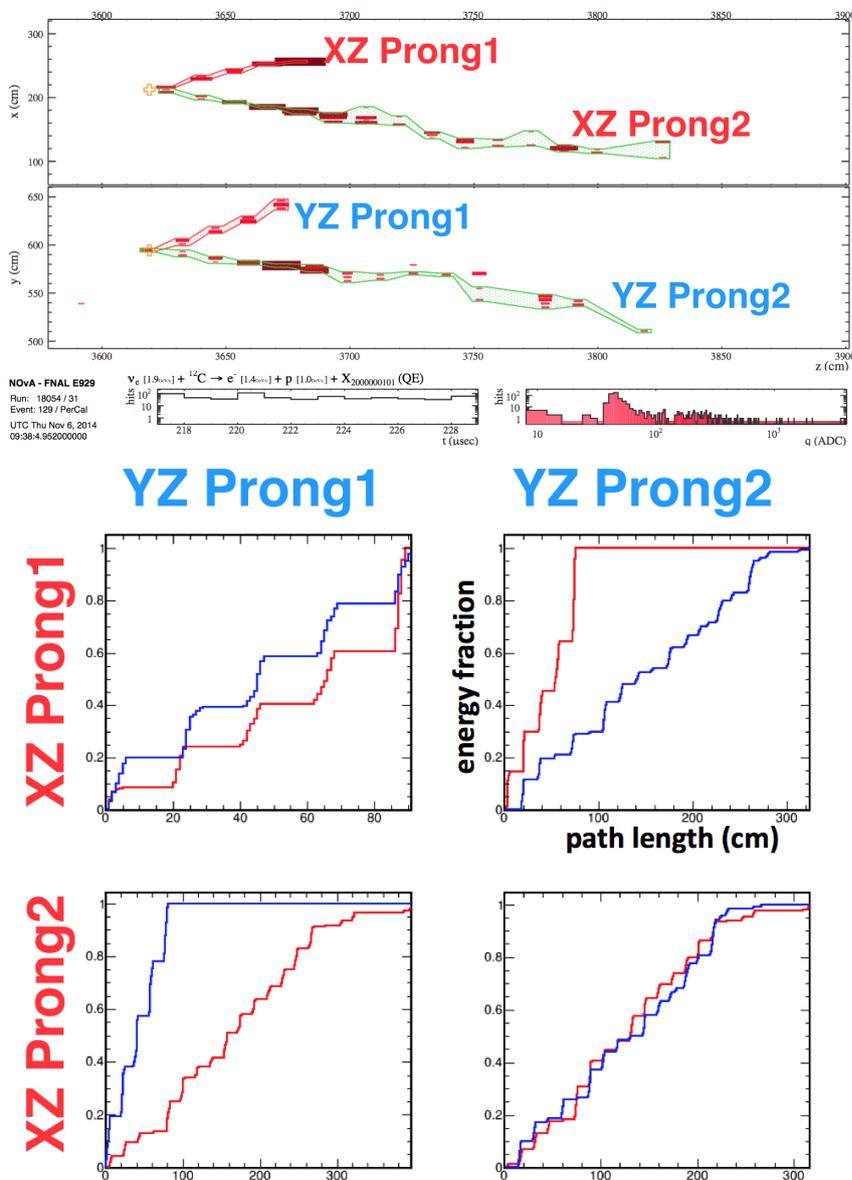


Figure 5.9: A ν_e CC QE event *run,subrun,event* = 18054,129,33 clustered into four distinct prongs (top). The four prongs are evaluated based on their similarity scores. For the Kuiper test, the closer the two curves in each of the matching profile plots(bottom) the lower their score and the higher their similarity. For this event XZ prong 1 should be matched with YZ prong 1 and XZ prong 2 with YZ prong 2.

particle. X_L is the radiation length in the local medium. The scattering angle θ_0 in the Gaussian multiple small angle approximation encompasses 98% of total scatterings. With the NOvA detector geometry and composition this yields a $\theta_0 = 5.4 \cdot 10^{-3}$ rad for 2 GeV muons. The remainder of scatterings involve larger scattering angles, primarily originating from hadronic interactions mediated via the strong force. The typical radiation length for these in the NOvA detectors is 10 planes. Combining these scattering types leads to generally linear tracks with the occasional larger scattering, continuing linearly afterwards.

Both KalmanTrack and BPF are used for reconstruction of these types of trajectories. Others tracking algorithms are used for the reconstruction of specific particle track types or features which are not easily reconciled with the assumptions and approximations underpinning both KalmanTrack and BPF, such as the typical Bremsstrahlung produced by electron showers as they stop.

5.3.1 KalmanTrack reconstruction

The KalmanTrack module starts with hits contained in the slice split by view. It explicitly does not use the previously reconstructed prongs. In a given view pairs of hits are compared based on their location. If the separation between hits is less than 4 cells they form a seed. Here seeds are candidate segments of the tracks. For each seed the position and slope of the candidate track segment is estimated. Using Kalman filters[130] with errors coming from various sources these track segments are extended to form a prediction for a larger track segment. The errors are comprised of uncertainties in hit location due to the granularity of the detectors, statistical fluctuations in the measurements and the potential scatterings occurring along the track. If a considered hit has a separation below a set threshold it is added to the track segment and the position, slope and errors are updated. This process is repeated until there are no more hits consistent with the track segments. Initial propagation of seeds and hits starts at high Z , the far end of the detector, and ends at low Z . A final propagation in the opposite direction is performed to better separate the track segments close to the interaction vertex. Hits associated to track segments

are removed from the slice and the procedure is repeated until no valid track segments can be found. Track segments which are contained in other longer segments or are less representative of the data, having larger errors on the associated hits, are removed. A schematic overview of this methodology can be seen in Figure 5.10.

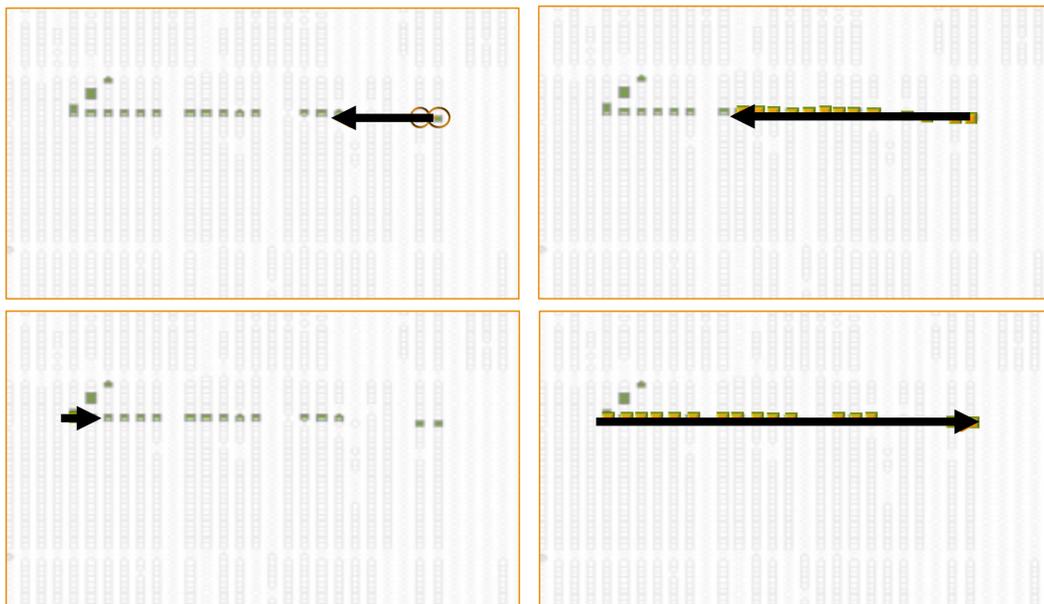


Figure 5.10: Schematic overview of the KalmanTrack reconstruction showing the progression from a track endpoint to the vertex and the subsequent backwards extrapolation. Inspired by [131].

The trajectories for each view are determined via a fit of the track segments, which due to the addition of hits can now be long enough to cover the entire trajectory. At each point along the track the location and slope are estimated following the same procedure as before. The error of all the hits before this point to the produce track prediction is calculated as well the error of the hits after this point. A weighted sum of these two errors is taken and minimized to produce the final track for this view. View matching occurs through a comparison of the Z components of each 2D track. The possible combinations of tracks are paired based on the following metric,

$$S = \frac{\Delta z_i + \Delta z_f}{Overlap} \quad (5.19)$$

where Δz_i is the difference in the start planes of the tracks in each view and Δz_f is the difference in the end planes. The overlap is the number of planes that overlap in the Z direction between the start and endpoints of the tracks. To be considered for pairing the tracks are required to have a non-zero overlap. Additionally combinations of tracks in one view can be merged if this results in a lower score when paired with a track in the opposite view. The combination of tracks with the lowest score is matched as a 3D track and removed from the list of 2D tracks. This matching procedure is repeated until no more matches are found. Figure 5.11 shows the full reconstruction method applied to an example FD event. In Figure 5.12 a measure of the performance of this reconstruction method is provided.

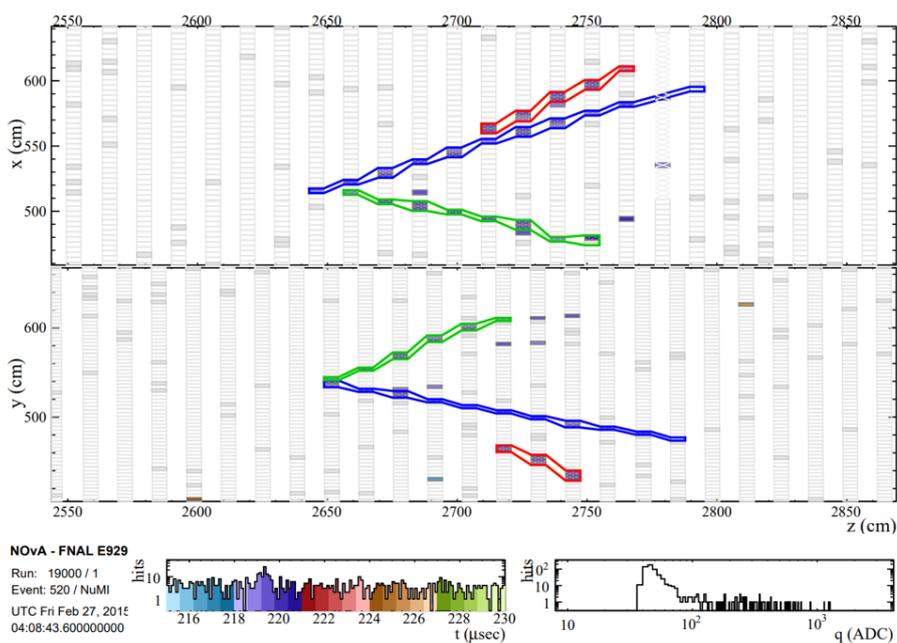


Figure 5.11: Example of KalmanTrack reconstructed tracks for a FD event.

5.3.2 BreakPointFitter

The BreakPointFitter uses the 3D prongs described in section 5.2.3. Relying on the information contained in the clustered hits, such as the position and energy deposited, it applies a track fitting algorithm [132] which is capable of fully capturing the multiple scattering which largely determines the overall shape

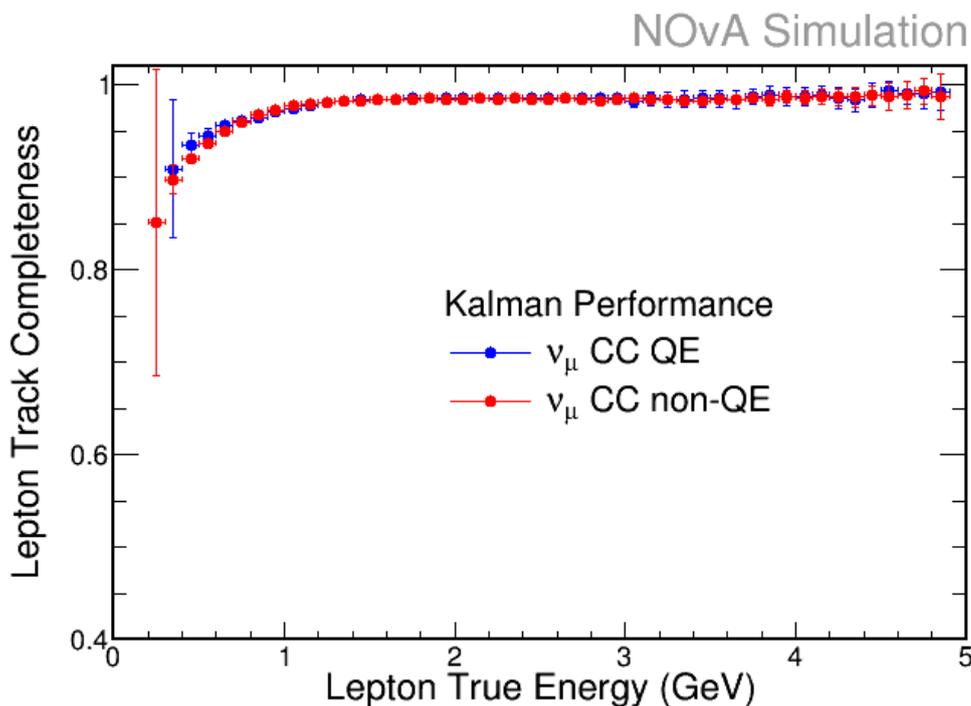


Figure 5.12: Completeness of KalmanTrack produced tracks as a measure of performance. The ratio of the visible energy associated to the track over the total true visible energy is shown for the ν_μ FD QE and non-QE MC events.

of the particle track. In each view a track is reconstructed by the repeated placement of scattering planes along the direction of travel, in the general case along the positive z -direction². In between two scattering planes the track is assumed to be approximately linear. A schematic overview is shown in Figure 5.13.

The trajectory resulting from the application of this procedure is,

$$\zeta_i = a + bz_i + \sum_{J=1}^M \alpha_J (z_i - Z_J) \Theta(z_i - Z_J) \quad (5.20)$$

where a is the vertex location determined previous, z_i the current z -coordinate, b the track angle at the vertex, α_J the deviation angle and Z_J the z -coordinate at the scattering plane J . Θ denotes the Heaviside step function which yields one when $z_i > Z_J$ and zero for $z_i \leq Z_J$. In total there are M scattering planes

²Cases with backwards travelling particles or large scattering angles are accounted for through the rotation of the prong hits into a track-based reference frame.

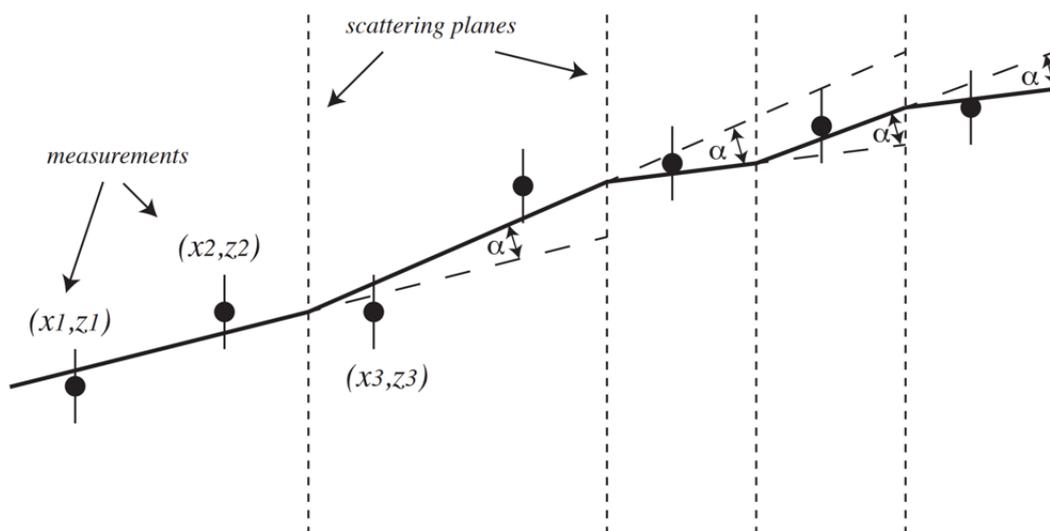


Figure 5.13: Schematic overview of the BreakPointFitter. A track is fitted along the data points through the insertion of scattering planes (dashed lines) which allow the track to deviate up to a specific scattering angle.

present, the exact number of which can range up to the total number of planes with hits in them minus one (accounting for the vertex location). To complete the fitting of the data with this model of the particle trajectory a χ^2 goodness of fit test is applied, which is defined as,

$$\chi^2 = \sum_{i=1}^n \frac{(x_i - \zeta_i)^2}{\sigma_i^2} + \sum_{J=1}^M \frac{\alpha_J^2}{\sigma_J^2} \quad (5.21)$$

where σ_i refers to error in the locations of the hits due to the finite size of the detector cells and σ_J refers to the error in the scattering angle of plane J as estimated from equation (5.18). Minimizing this χ^2 yields the set of parameters that best characterize the given hits in terms of a particle trajectory.

The placement and number of scattering planes to be used for the track reconstruction is determined based on the hits present along the potential track's path. Starting from the last recorded hit in the prong a summation is run working backwards to the vertex. All the traversed materials are accounted for and both the energy loss dE/dx as well as the number of radiation lengths X_0 traversed are determined as a function of the distance d along the track from the vertex. Using this information scattering planes are placed along the

track if:

1. d exceeds either two radiation lengths or a distance of one meter.
2. The expected root mean square of the transverse scattering $d \cdot \theta_0$ (5.18) exceeds 3 mm.

The first of these conditions places a scattering plane at least once every 7 planes, the second ensures accuracy of the linear extrapolation to the cell wall thickness which is the minimal resolvable distance in the detector.

Scattering Surface Locations

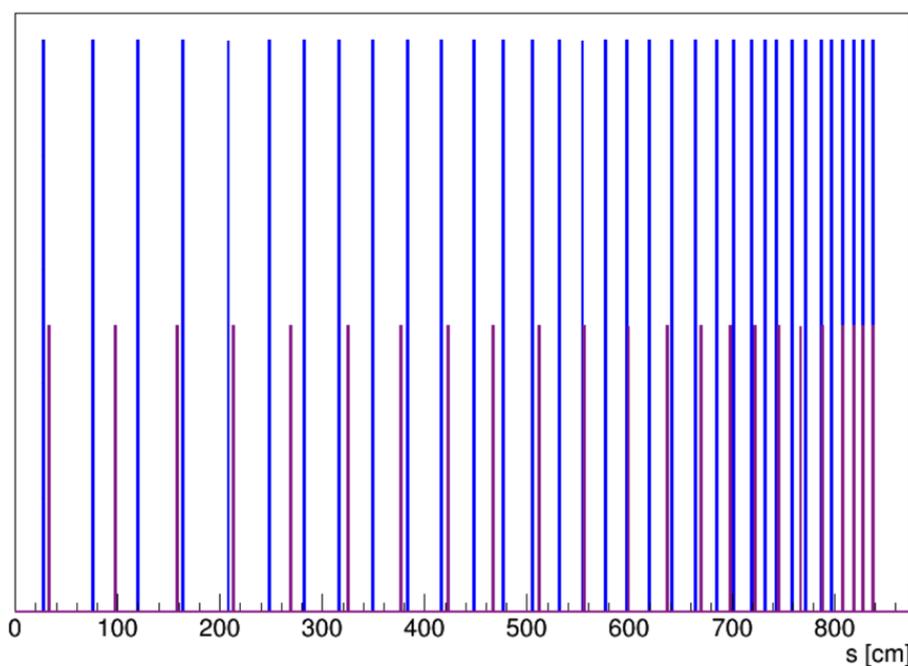


Figure 5.14: Distribution of scattering surfaces for the muon (tall, blue) and the proton (short, purple) fit assumptions along the track length s . These scattering surfaces are produced for the BPF track reconstruction of a $2 \text{ GeV}/c$ muon with a track length of $\approx 8.5 \text{ m}$.

The fit is run three times with differing sets of assumptions, namely that the particle which produced the track is either a pion, proton or a muon. These affect all parameters in the fit which carry a dependence on the characteristics of the particle, such as the mass, momentum and energy lost during traversal of the detector. For each fit a final integration is done to determine the energy

and momentum which would produce this trajectory. This information is stored along with the track and quality of the fit for later usage in the analysis. The distribution of scattering planes for the two fit assumptions is shown in Figure 5.14. Additionally an example is given in Figure 5.15 of the reconstruction of an event using the BPF method.

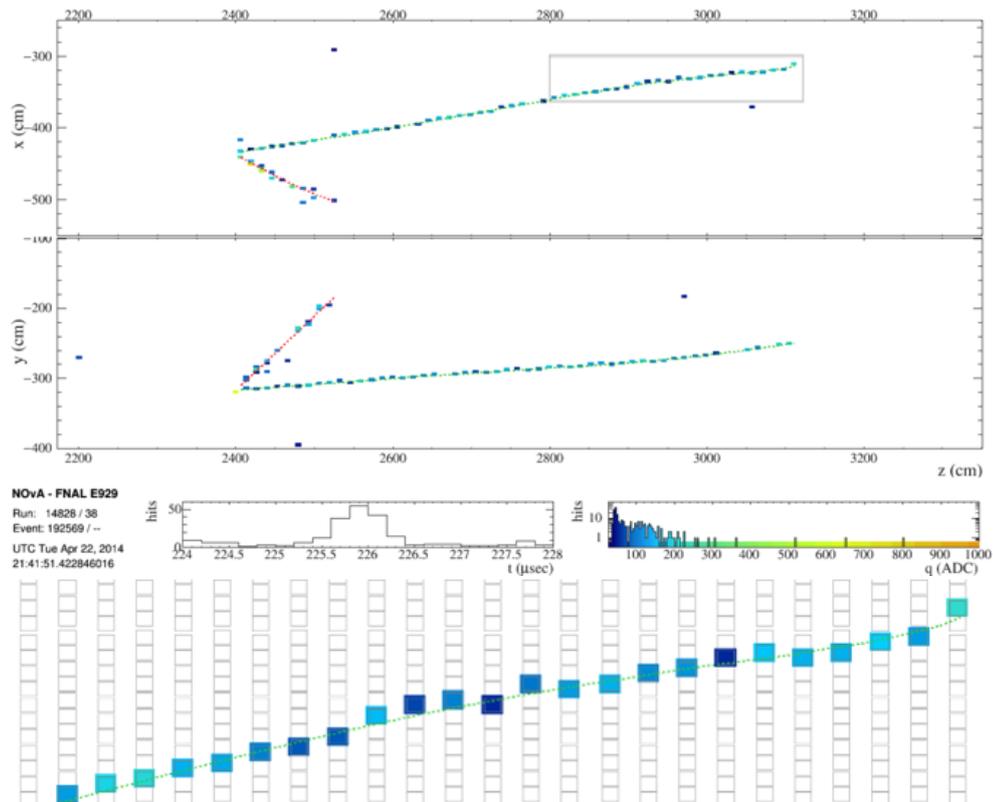


Figure 5.15: Example of a BPF reconstructed neutrino candidate event in the FD. The shown BPF produced tracks are generated under the muon fit assumption. The end of the long muon track in the X view is shown enlarged on the bottom.

Chapter 6

Machine Learning in NOvA

With the reconstruction of the vertex, particle trajectories and deposited energy the interaction can be further characterized through the calculation of additional derived variables. These could be any number of features or kinematics describing parts or the full event. Historically the reconstruction variables formed the foundation for any further physics analysis as they provide a consistent method for the separation of candidate events into signal and background. These are therefore referred to as traditional reconstruction techniques. While these are still widely used, advances in artificial intelligence and the increased availability of computational resources have enabled alternative analysis techniques. The class of computer algorithms which forms the basis for these alternative analysis techniques are categorized as Machine Learning (ML) algorithms. As these algorithms are given information to learn from they typically change, or update, their internal state to improve their performance.

Both ML algorithms and traditional reconstruction techniques have different advantages and drawbacks which determine where and how they can be leveraged within the frame of a physics analysis to maximize their effectiveness. ML algorithms excel in scenarios where large amounts of training data is available as their performances scales with it. This scaling is not infinite and will eventually a plateau of performance is reached. Additionally evaluation speed after a ML algorithm has been trained is typically faster than repeatedly performing the calculations required for traditional reconstruction variables.

The learning process of a ML algorithm can be long and can require specialized hardware to accelerate the training. The learning of ML algorithms can be difficult, if not impossible, to interpret and explain. Traditional reconstruction techniques by their construction are typically fully explainable and interpretable. With these benefits and disadvantages ML techniques enhance parts of the NOvA analyses and are used for the purposes of particle classification and reconstruction [133]. Not all ML techniques employed in the NOvA experiment will be covered in this work, the primary focus is on the CNN event classifier. In this chapter familiarity with ML concepts are not assumed. The general components of neural networks and their basic training strategies are covered in sections 6.1 and 6.2. The convolutional neural network is examined in more detail in section 6.3 and the evolution of its usage in the NOvA experiment can be found in section 6.4. Finally the general usage of datasets for the training of neural networks, domain shifts and specific adversarial techniques for the purposes of reducing negative effects stemming from these are discussed in sections 6.5 and 6.6 respectively.

6.1 Artificial neural networks

One of the most common types of ML algorithms is the Artificial Neural Network (ANN). It approximates the learning capabilities of a biological brain through simplifying the brain structure. The basic formulation of a neural network consists of neurons and connections, reminiscent of the nerve cells and synapses. The neurons receive a signal, process it and transmit it. The synapses act as wires and carry the signals from neuron to neuron.

The network is then built up from these basic components, typically structured in layers of neurons with connections to the next and previous layers. An example of a class of simple ANN's are the Multi-Layer Perceptrons(MLP). These networks are fully connected, which means that each neuron in every layer is connected to all neurons in the next layer. The general flow of information starts at the input layer then moves through a number of hidden layers into a

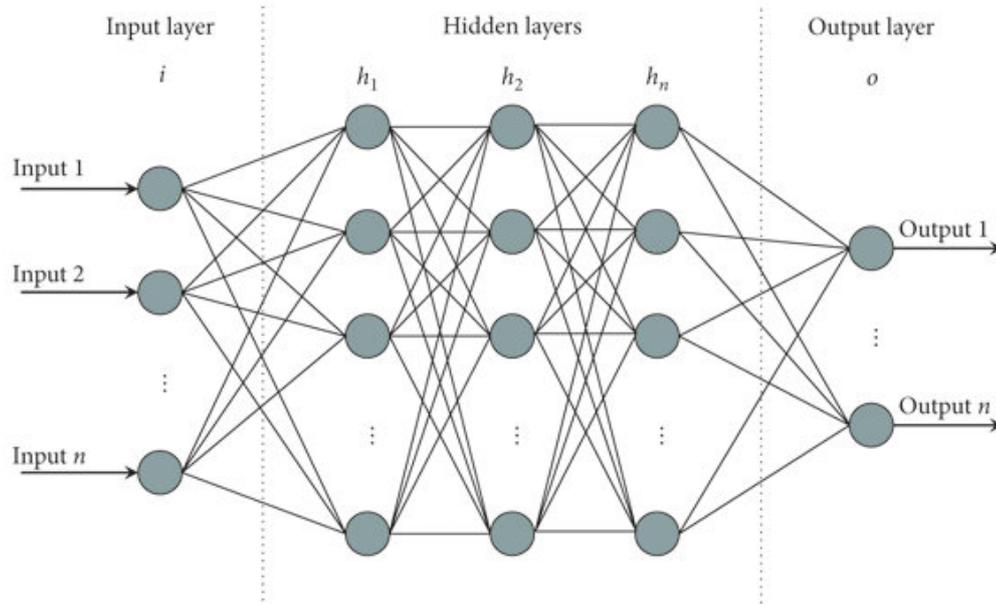


Figure 6.1: Schematic overview of simple MLP neural networks. Here the typical ordering of the input layer followed by the hidden layers feeding into the network outputs is shown. Courtesy of [134].

final output layer. This is true for feed-forward neural networks. The number of hidden layers determines whether a network is considered shallow or deep, although the boundary for this distinction is not rigorously set. A schematic overview of an example MLP neural network is shown in Figure 6.1.

6.1.1 Neurons

A simple neuron takes in a set of inputs x_i and converts these numbers into an output y . This is done by first multiplying each input with an internal changeable (learnable) weight w_i and summing all of these products with a learnable bias. This sum is then passed into an activation function σ , the output of which is the output of the neuron,

$$y = \sigma(z) = \sigma \left(\sum_{i=1}^N w_i x_i + b \right) \quad (6.1)$$

The operations contained in equation(6.1) are graphically represented in Figure 6.2. The neuron output forms the input for subsequent neurons in the

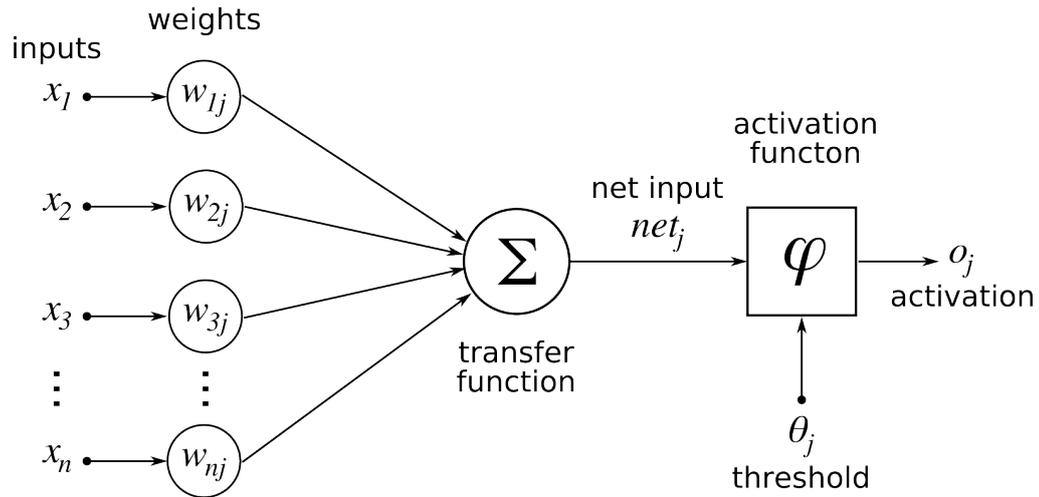


Figure 6.2: Schematic overview of a neuron. Courtesy of [135].

network. Repeatedly applying equation (6.1) allows for the expression of the network output in terms of the inputs and all the weights and biases. These calculations can be performed in parallel when considering that equation (6.1) consists of linear algebra in the form of matrix multiplication.

6.1.2 Activation function

The activation function simulates the nerve cell behaviour of firing a signal along a synapse. In the neural network this function limits what information gets propagated into the output. There are many different activation function which are commonly used for neurons in neural networks. A collection is discussed in more detail and an overview is given in Figure 6.3.

The simplest activation function is a linear one where the output is equal to the input,

$$f(x) = x \quad (6.2)$$

There are two issues with this choice. Firstly as all operations in the network are now linear, its output will be reducible to a linear function of the input. Therefore the network will be unable to learn any non-linear behaviour. Secondly the derivative of the activation function is constant and thus lacks any dependence on its input. The derivative forms an integral part of the training procedure and it having these characteristics will prevent application of the

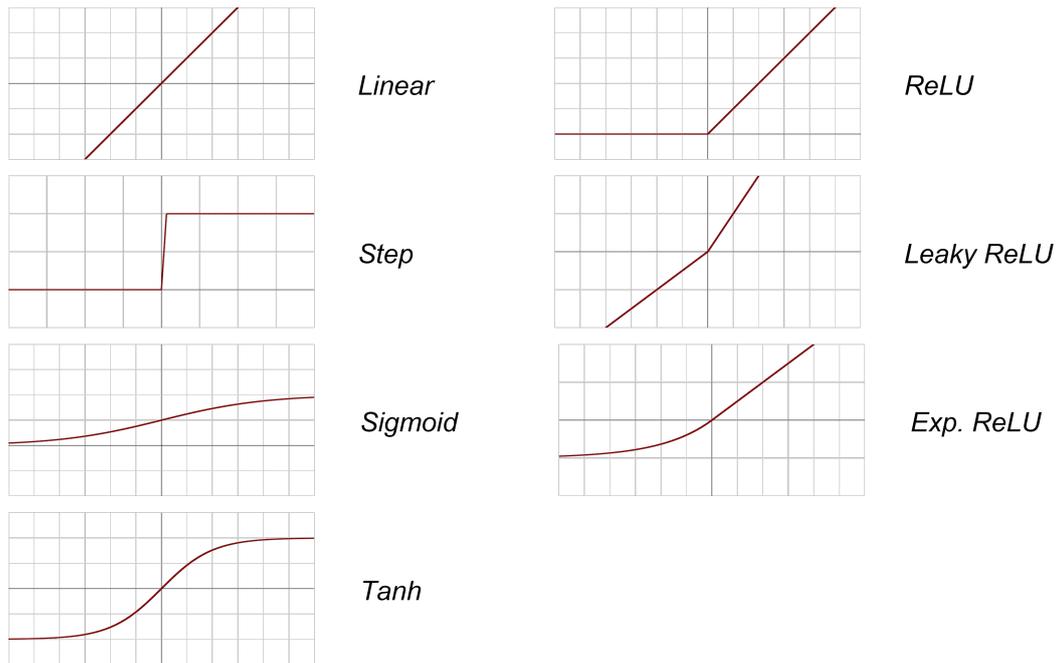


Figure 6.3: Overview of single fold activation functions which depend solely on x . Softmax depends on multiple inputs and is therefore not included.

procedure¹.

Another example of a simple activation function is the binary step function,

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases} \quad (6.3)$$

It cannot be used for multi-class classification as its output is binary and it has a derivative of zero.

To resolve the issue of linearity non-linear activation function can be used. The sigmoid function is one such functions,

$$f(x) = \frac{1}{1 + e^{-x}} \quad (6.4)$$

It is smoothly differentiable and constrains the output to the range of $[0 - 1]$. For this reason it is typically used to represent a single probability score in the networks output. When employed in the hidden layers the neuron outputs will

¹Alternatives to the default training procedure are currently being researched, but lack widespread application.

be forced to be positive, which can be troublesome for training scenarios which rely on a full range of values.

The hyperbolic tangent (\tanh) resolves this range limitation by allowing negative values,

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (6.5)$$

It is therefore better suited for usage in the hidden layers than the sigmoid function.

The Rectified Linear Unit (ReLU) approximates linearity through a piecewise definition of its output depending on the sign of its input,

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} \quad (6.6)$$

The non-differentiability at zero poses an issue of the usage as an activation function. This issue is resolved by setting the value of the derivative equal to zero or one manually for this specific point. As the ReLU function is linear for the positive input values the neural networks behaviour is approximately linear while still being able to capture non-linear characteristics present in the data, furthermore this simplifies optimization of the network [136]. Negative input values result in the activation function returning 0 and the neuron not being active in the network for this data sample. This quality greatly reduces the computational complexity involved with both the training and deployment of the network. It is worth noting that this property can also lead to the scenario where one or more neurons do not activate for any inputs, effectively dying out.

To mitigate the issue of dying ReLU activated neurons and an ill defined derivative, modifications can be made to the ReLU function. For example the leaky ReLU function adds a slope to the outputs of the negative input values,

$$f(x) = \max(ax, x) = \begin{cases} ax & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (6.7)$$

The Exponential Linear Unit (ELU) replaces the linear component with an exponential for negative input values,

$$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (6.8)$$

Its derivative is both well defined and smooth.

The softmax activation function is a modification to the sigmoid function so as to enable its usage as a probability score for a multiclass classification problem. It normalizes the output of all neurons in the layer to the sum of their exponential output values,

$$f_i(x_N) = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}} \quad (6.9)$$

where f_i is the activation of the node i out of N present in the layer.

6.2 Training

A simple neural network is constructed by defining its architecture. In the case of a simple MLP this architecture consists of an input layer, one or more multiple hidden layers and an output layer. The behaviour of the neurons in this network is described by equation (6.1). When the architecture is defined the network, also referred to as the model, is required to be initialized before it can be used to process data. Initialization determines the network parameters, such as the neuron weights and biases. Different strategies are available for this purpose [137], [138]. Examples of common initialization are the zero, random, random with a set seed, Xavier or Glorot [139] uniform and many more. Unless specified differently Glorot/Xavier uniform is chosen as the default.

With the network fully initialized a forward pass of the data is now possible. In a forward pass a single sample (or batch) of input data is passed into the input layer and propagated through the neurons of the hidden layers until finally converging into an output of the output layer. As the network has not

learned or updated its internal trainable parameters the output is expected to be random in nature. The network and its output in its current state cannot be used for any purpose beyond service as a random noise generator. This can be remedied through repeatedly applying a training procedure. This is possible as the output has a quantifiable dependency on both the input and the internal weights and biases of the network. The training method enables variation in the model parameters such that the network's output better represents the expected output. The training procedure can be repeated many times and as such only a small incremental improvement in network performance with a change to the network's weights and bias is required for this method to be effective.

6.2.1 Loss functions

With the initial output of the network and knowledge of what this output should resemble through access to truth level information (true labels) it is possible to define a cost or error function that quantifies the difference between the two. The category of cost functions used in neural network training are commonly referred to as loss functions. Regression and classification problems generally require different loss functions although some can be applied to both. An example of a loss function employed in regression problems is the squared error,

$$L(y, \hat{y}) = (y - \hat{y})^2 \quad (6.10)$$

In this loss function the difference between the network output y and the expected truth value \hat{y} is evaluated. If multiple samples are required to be evaluated, the average squared error over all the samples is taken. This is the mean squared error.

An example loss function for multiclass classification problems is the categorical cross entropy,

$$L(y, \hat{y}) = - \sum_{i=1}^C \hat{y}_i \log(y_i) \quad (6.11)$$

where there are C output classes. This loss function is typically used in combination with one-hot encoding of truth class information and softmax activation of the output layer. One-hot encoding transforms the truth information into a sparse vector containing a single positive entry of 1 for the correct class and zeros for all other classes. This naturally forms a target to match for the output of the softmax activation function.

There are a large number of loss function defined in the literature [140] all with different (dis) advantages, and even then it is not abnormal for a custom loss function to be developed when a specific network architecture or problem environment calls for it.

6.2.2 Backpropagation

The neural network learning is done through the application of a process called backpropagation. After a forward pass is completed and the loss has been calculated a backwards pass is performed. This backwards pass does not seek to reconstruct the data in terms of the input features but rather seeks to determine the derivatives or gradients of every weight and bias (trainable parameter). These derivatives dictate how the loss function varies with a change to any of the trainable parameters.

The backwards pass begins with the loss function for this batch or sample. Using equation 6.1 all nodes in the layers of the network which contribute to the loss function can be represented. This is illustrated for the final two layers of an example neural network in Figure 6.4. For each layer the weights w^L , biases b^L and output of the previous layer y^{L-1} form the output z^L which is subsequently fed into an activation function σ to produce y^L .

The gradients of the weights w_{ij}^L and biases b_i^L with respect to the loss function can be determined through the application of the chain rule,

$$\frac{\partial L(y(x, w, b), \hat{y})}{\partial w_{ij}^L} = \frac{\partial L}{\partial y_i^L} \frac{\partial y_i^L}{\partial z_i^L} \frac{\partial z_i^L}{\partial w_{ij}^L} \quad (6.12)$$

where L and i indices indicate the current layer, node (neuron) and the j^{th}

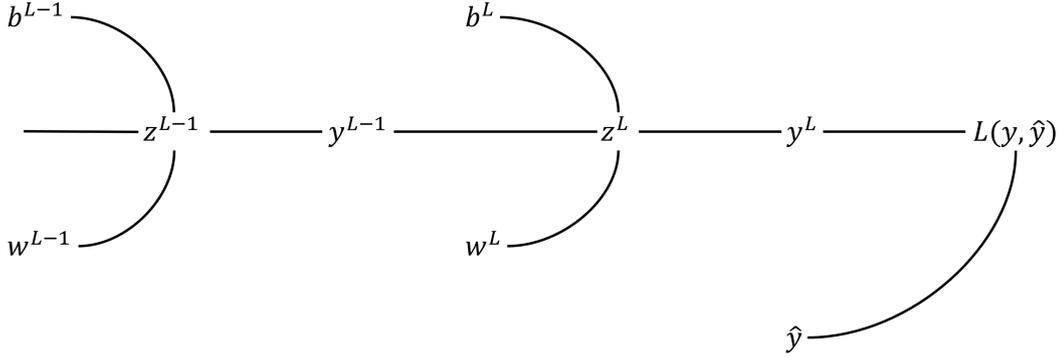


Figure 6.4: Schematic of the final two layers ($L-1, L$) of a neural network.

input to that node. The first two partial derivative can be determined from the definitions of the loss function and the activation function. The final partial derivative is equal to the output of the previous layer ($L-1$) for the weights or 1 for the biases,

$$\frac{\partial z_i^L}{\partial w_{ij}^L} = y_j^{L-1} \quad (6.13)$$

Computation of the associated derivative for layer ($L-1$) can be simplified through the usage of the chain rule coupled with derivatives that have been calculated in the previous step,

$$\frac{\partial L(y(x, w, b), \hat{y})}{\partial w_{jk}^{L-1}} = \frac{\partial L}{\partial y_i^L} \frac{\partial y_i^L}{\partial z_i^L} \frac{\partial z_i^L}{\partial y_j^{L-1}} \frac{\partial y_j^{L-1}}{\partial z_j^{L-1}} \frac{\partial z_j^{L-1}}{\partial w_{jk}^{L-1}} \quad (6.14)$$

where the third to last term and the last term are given by the weight w_{ij}^{L-1} and the output of the previous layer ($L-2$),

$$\frac{\partial z_i^L}{\partial y_j^{L-1}} = w_{ij}^L, \quad \frac{\partial z_j^{L-1}}{\partial w_{jk}^{L-1}} = y_k^{L-2} \quad (6.15)$$

The k indicating a input to the j^{th} node of the layer ($L-1$). Reusing the derivatives of the previous layers the backwards pass can be iteratively completed up to the input layer.

6.2.3 Gradient descent and optimizers

The calculated gradients are used as input for an optimizer. This refers to any algorithm which optimizes, in this case minimizes, the loss function. One of these optimizers is stochastic gradient descent (SGD) which employs an approximation of the regular gradient descent optimization strategy. The stochasticity results from the optimizer being used with only a batch or single sample of data which is randomly selected from the entire training dataset.

Gradient descent is an iterative optimization algorithm. It determines the direction of steepest descent on the surface traced out by a multi-variate function. This direction is obtained from calculating the gradient with the local values of the function variables. Depending on the number of variables which form the input to the multivariate function this surface can be embedded in a high dimensional space. Taking a step along the direction of the negative gradient results in a lower value of the function through a relatively small change in the variables,

$$a_{n+1} = a_n - \eta \nabla F(x, a) \quad (6.16)$$

where η denotes a parameter determining the size of the step to be taken. Repetition of these steps results in the locating of a local or global minimum. In Figure 6.5 this method is demonstrated on a non-convex surface.

SGD applies this methodology to the minimization of the loss function through repeatedly updating the trainable parameters of the network which form input variables to the loss function.

SGD is not the only optimizer which is widely used in neural network trainings. Different optimizers are characterized by their strategy for updating the trainable parameters of the network. The weights can for example be updated based not only on the local gradients of the current location but also on the gradients present at previous steps. This is referred to as the inclusion of momentum. It is particularly helpful in avoiding large deviations based on local differences in the loss surfaces as well as providing a more consistent traversal

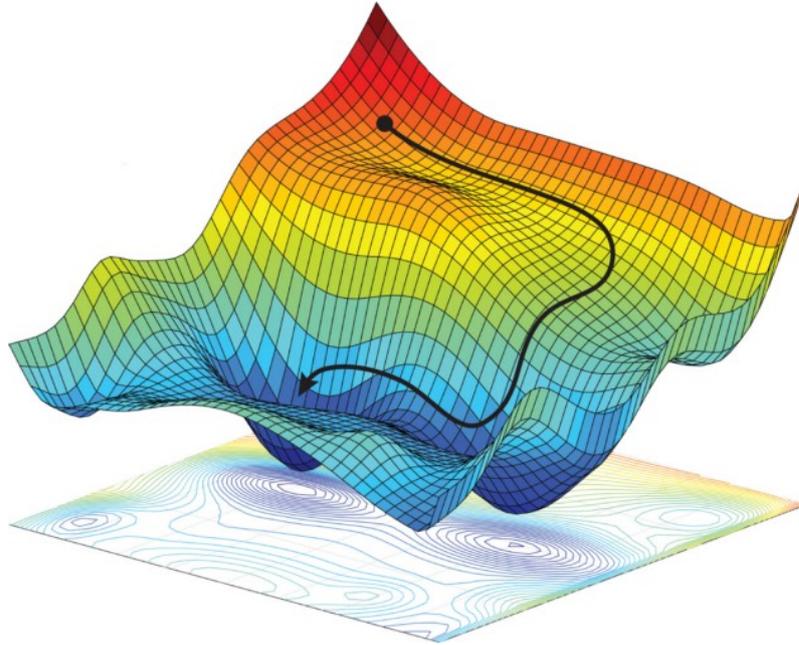


Figure 6.5: Non-convex surface representing the function on which gradient descent can be employed for the purposes of locating a local or global minimum. Courtesy of [141].

of relatively flat regions of the loss surface. These regions, such as saddle points, can yield very small or vanishing gradients in the local area resulting in small parameter updates. Other optimization strategies can introduce variations to the step sizes. SGD employs a single step size for all the parameters in the network, other optimizers assign a unique step size for each parameter and/or introduce methods for the determination and variation of the step size based on secondary statistics.

The step size used in SGD is an example of a hyperparameter. These are parameters whose values are chosen at the start of training and are used to control the training process. Other hyperparameters include but are not limited to the batchsize, which is the number of samples propagated before a parameter update occurs, and the epochs, which is the number of times the full training data is shown to the network for learning.

As the iterative training procedure through gradient descent with back-propagation relies on the gradient's magnitude and direction of all the trainable parameters in the network, any undesirable behaviour of these gradients can

negatively impact or completely stop the desired learning. When the values of parameters or their gradients become small (large) this can result in vanishing (exploding) gradients. When the gradients decrease in magnitude the updates to the trainable parameters become small and eventually stop changing altogether. For the case of increasing gradients the values of the trainable parameters or their gradients grow until the network behaviour becomes unstable and the values eventually turn into Not a Numbers (NaN's) resulting from overflow errors.

The structure of a neural network can facilitate these instabilities. For each layer that an input passes through in the network a multiplication with the weights of that layer is applied. For deeper network this results in an exponential relation between the network output and its input,

$$y \sim w^N x \quad (6.17)$$

where N is the number of layers between the input and output of the network. This relation is not exactly true as it fails to fully take the role of the biases and activation functions into account. It does however illustrate the potential issues caused by certain values of the weights and their impact on the network output and thus the gradients in the learning procedure. If the collective values of the weights are smaller than one ($w < 1$), the exponential relation will result in vanishingly small outputs. Likewise larger values will cause the output to explode.

The gradients can be further affected by not only the network output but also the repeated application of the chain rule. As seen in equations 6.12, 6.13, 6.14, 6.15, moving backwards through the layers during backpropagation results in a repeated multiplication of the activation function's gradient for those layers. If the values of that gradient become very small or large they will affect all layers beyond the current one. While activation functions are typically chosen with these issues in mind, a significant subset of activation functions are ill behaved with respect to these issues for specific ranges of input values.

For example, if the derivatives of an activation function tend towards zero for large values, which can be either positive, negative or both, it is saturating. The activation functions that constrain output to specific ranges, typically 0 to 1 or -1 to 1, tend to be saturating.

The issue of vanishing/exploding gradients has multiple possible causes and as such there are multiple strategies available to mitigate the issues. The choice of activation functions which are build into the network is unconstrained in most scenarios. It is therefore possible to construct or choose non-saturating functions which have well defined and behaved derivatives in the crucial ranges of input values. An ensemble of various different activation functions can be a viable option as well. To mitigate the effects of small or large values of the weights and biases, specific initialization methods can be followed.

Additional regularization is possible through the application of gradient clipping. This is a technique where the gradients calculated during backpropagation are clipped at a specific threshold. This results in a reduced possibility of positive feedback occurring during training. It does limit the speed at which a network can learn as the step sizes now have a strictly defined upper limit.

6.3 Convolutional neural networks

The pixelated nature of the NOvA detectors results in two highly correlated images which characterize an event, one for each view. Beyond spatial correlations, similarities arising from the underlying interactions are expected to be present. While individual variation is likely, larger shared similarities will be present for equivalent particles. Protons, pions, muons or electrons of different energies will express correlations in their presented topologies. Proton to proton, pion to pion, etc. This information can be leveraged by computer vision techniques to perform event classification in the NOvA experiment. A deep convolution neural network with a Siamese branch structure in the architecture is built to achieve high performance event classification. It is referred to as the Event Convolutional Visual Network, EventCVN or CVN. An overview of the

architecture of this network is shown in Figure 6.6.

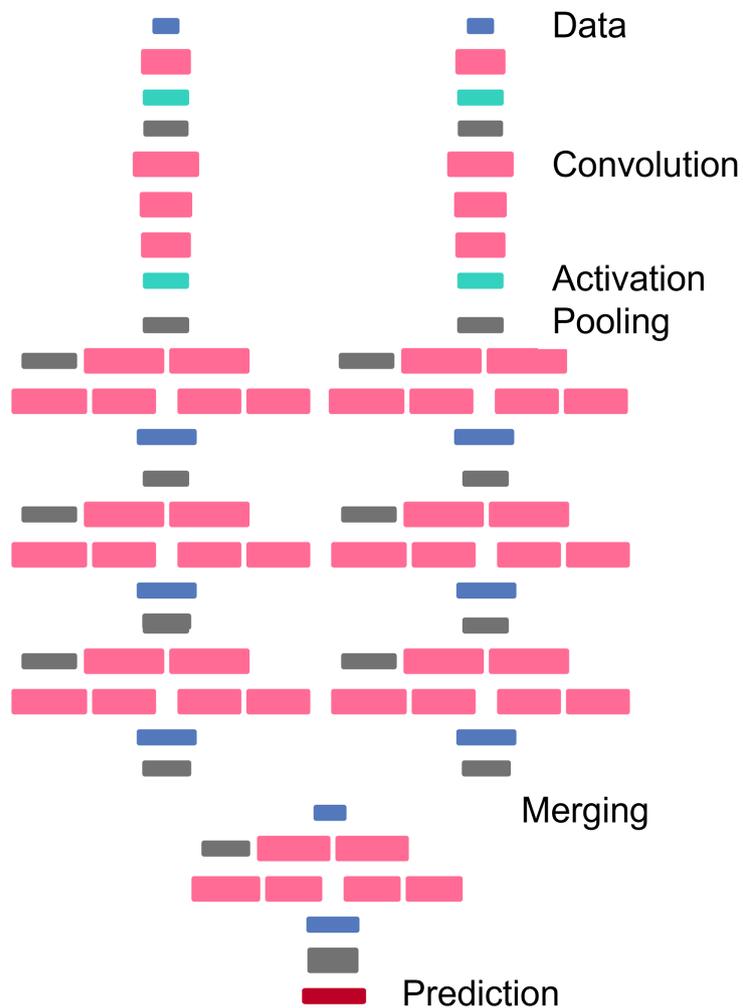


Figure 6.6: Schematic overview of the two branch Siamese network architecture with the input, convolutional, activation, pooling, merging and output layers labelled.

The CVN is a complex version of the earlier described neural networks with multiple specific augmentations which allow it to perform the classification task. The task consists of a determination which neutrino interaction type produced the two 100×80 pixel input images shown to the network as input. The input images, or pixelmaps, are rectangular windows of detectors cells surrounding the cluster of hits which constitute the event. In Figure 6.7 pixelmaps from a CVN training dataset are shown. The pixel values present in the pixelmaps

correspond to the deposited energy in the detector cells as calibrated. This energy is converted to a discrete scale from 0 to 255, this mapping preserves correlations present in the pixelmaps whilst reducing the complexity of values in the inputspace.

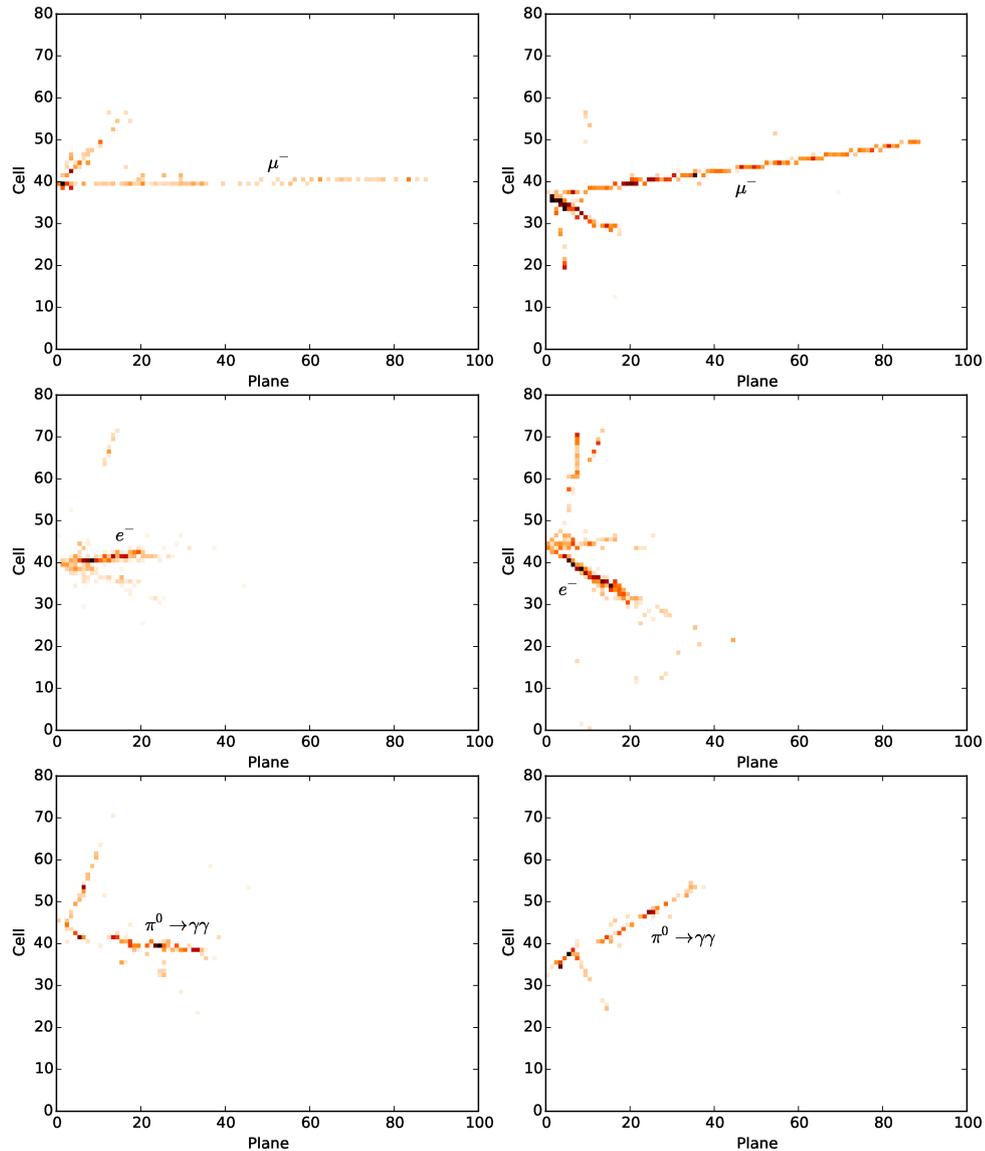


Figure 6.7: Examples of typical pixelmap images which form the CVN input. Three events are shown with both the X and Y view. A true ν_μ CC interaction (top), a true ν_e CC interaction (middle) and a true NC interaction on the bottom.

The classes of events for the purposes of classification are given as ν_μ charged current (ν_μ), ν_e charged current (ν_e), neutral current (NC) and cosmic ray muons (cosmic). Depending on the dataset used a fifth class can be

separated, v_τ charged current.

6.3.1 The convolutional layer

The CVN is build from a custom architecture which consists of two independent feature extracting branches. Each of these branches is free to learn features of the event expressed in one of the detector views. As these two branches are identical in structure they are effectively twins and hence referred to as Siamese branches. The base structure of a feature extracting branch begins with the initial convolutional layer. A convolutional layer takes in a tensor and convolves it with a kernel to produce its output. This operation is shown graphically in Figure 6.9. The trainable parameters of a convolutional layer is encoded in the values of which the kernel is constructed. The size of the kernel and the handling of edges are hyparameter choices which can be set at building of the network architecture. The output of the convolution layer is a feature map. For the initial convolutional layer to the feature extraction branch the pixelmap image is used as a 2D tensor (also known as a matrix). Its output consists of a low level feature map as the kernel can only access pixel information close to the center pixel where the kernel convolution is being applied. Repeated applications of convolutional layers can escalate local features into mid and high level features which could identify features across large swaths of the original input image. These features commonly do not reflect features identified through traditional reconstruction techniques, instead they capture abstract correlations in the feature space that hold higher levels of classification power as reflected through the network's performance. This issue is known as a failure in model explainability. The low, medium and high level feature extracting layers and their typical placement in a convolutional neural network is shown in Figure 6.8.

6.3.2 The pooling layer

The convolutional layer is not the only type of layer present in the feature extraction branch. Pooling layers are typically placed after a convolutional layer.

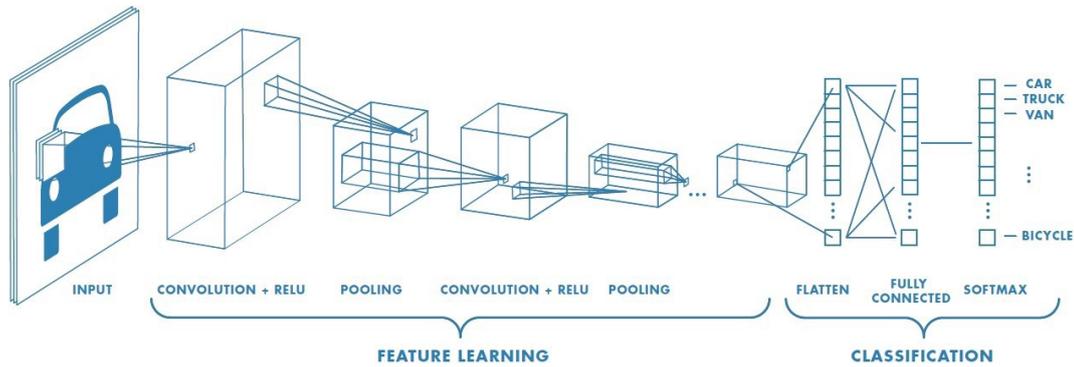


Figure 6.8: Schematic overview of a convolutional neural network consisting of several convolutional and pooling layers for the purpose of feature extraction feeding into a fully connected classifier. Courtesy of [142].

They typically act as higher dimensional filters resulting in a concentration of information through a reduction in the tensor size going from the input to the output of the pooling layer. Pooling layers can be configured in various ways, but a maximum or average pooling of a small area of the feature map is most frequently used. Besides the previously introduced convolutional operations, these two pooling operations are shown in Figure 6.9 as well. In average pooling the average of the selected inputs is taken, for maximum pooling the highest value is selected. A typical pooling layer could have a 2×2 or 3×3 filter with a 2 or 3 pixel long stride. The stride referring to the separation between applications of the filter. A more aggressive form of pooling is available through global pooling, this refers to a pooling operation where the entire feature map is condensed into a single output.

6.3.3 Batch normalization

To improve the learning performance and reduce the negative effects of internal covariate shift, batch-normalization layers are used throughout the network. Internal covariate shifts are a consequence of the sequential layered structure of neural networks as any changes to previous layers produce shifts to the input distributions of subsequent layers. Small learning rates, longer training times and conservative parameter initialization can compensate and reduce the impact of these shifts during training. Alternatively regularization of the inputs

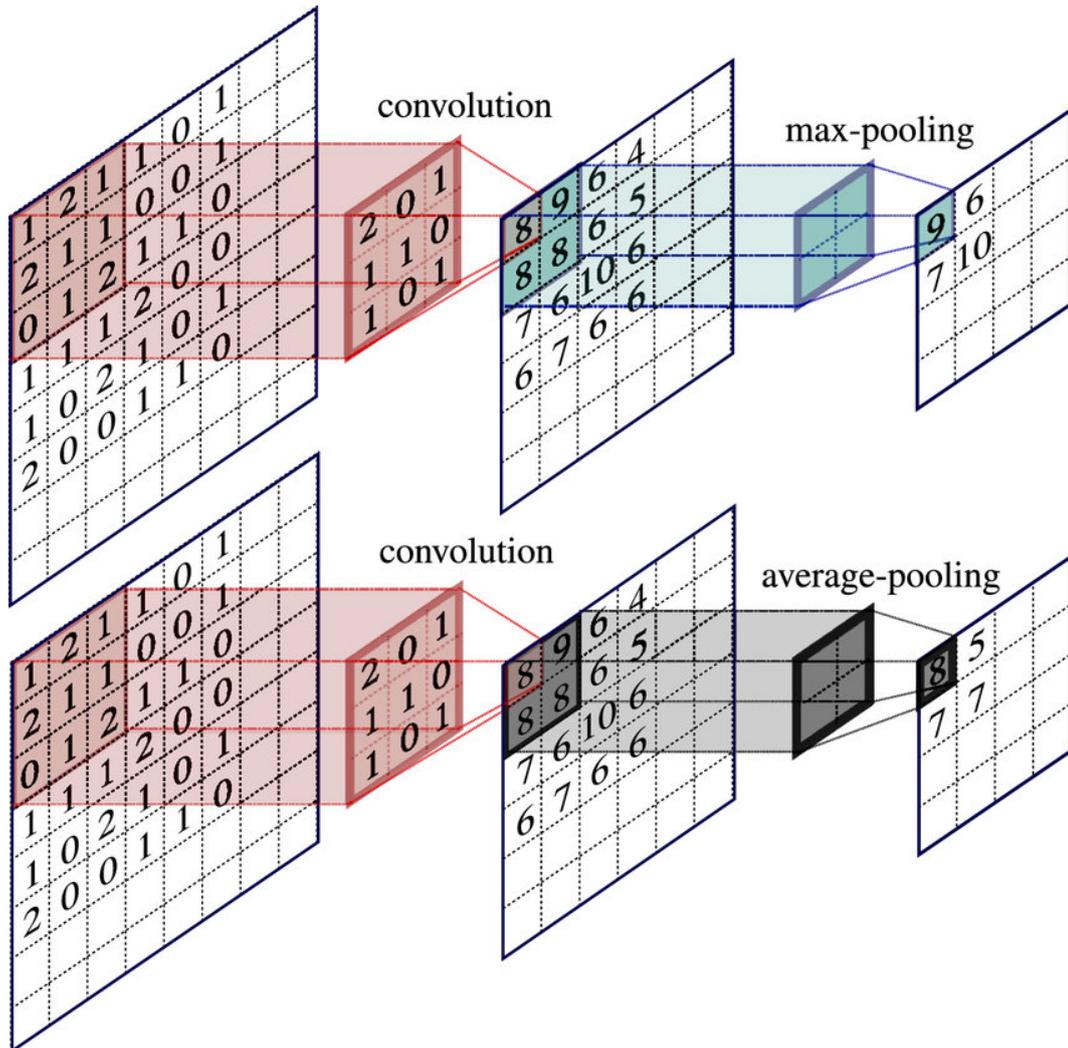


Figure 6.9: Schematic overview of the operations a convolutional and pooling layer perform in a neural network. Courtesy of [143].

or outputs of specific layers in the network can largely reduce the need for these decelerating solutions. Batch-normalization layers apply a normalization to the individual components of the supplied input, be that a tensor, matrix or vector. This normalization shifts the component to have a mean of zero and a variance of one. These statistics are estimated over the batch of samples that is currently being trained on. If desirable this normalization can be extended to larger subsets of the training dataset. To prevent a loss of non-linearity two trainable parameters for each normalizable input are introduced, a scale and shift factor. This results in the following function for the batch-normalization

layer,

$$y_i = \gamma \frac{x_i - \frac{1}{B} \sum_{i=1}^B x_i}{\sqrt{\frac{1}{B} \sum_{i=1}^B \left(x_i - \frac{1}{B} \sum_{i=1}^B x_i\right)^2 + \varepsilon}} + \beta = \gamma \frac{x_i + \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta \quad (6.18)$$

where the batch consists of B samples, ε is a small fuzz factor and γ , β are the scale and shift parameters. This function is applied repeatedly in a single batch-normalization layer for each input, with γ , β , μ_B and σ_B being unique for each input.

6.3.4 Connected classifier

The feature extracting branches feed into a single connection point. This forms the starting point of this classifier section of the network which is typically comprised of a fully connected standard MLP. Flattening the extracted features, this part of the network applies stepwise reduction on the input vectors for each layer to eventually produce a single output vector. This vector has a single component for each class present in the classification task. With the softmax activation function in the final layer this output vector represent a probability or confidence vector comprising the network's prediction score for each individual class. A value of zero corresponds to certainty that the shown sample does not belong to the reference class. A value of one provides certainty of the opposite. Typically values are produced which are located somewhere in the range of zero to one, conveying the degree of belief that the sample belongs to the reference class. The earlier discussed categorical cross entropy loss function would be used for training in this setup.

6.4 CVN architectures

The specific architecture for event CVN has evolved over time with multiple iterations. The major changes in architecture choice and the improvements which form the motivation for these changes are examined in chronological order. The general implementation remains the same, following the 2 Siamese feature extracting branches combining into a single classifier template. The

initial CVN architecture was based on the AlexNet [144]. Figure 6.10 contains a schematic overview of the AlexNet architecture. The field of computer vision is particularly active and as such advances with respect to neural network for the purpose of image recognition are frequent. These can be leveraged for the purposes of neutrino event classification, and drive continual updates to the techniques employed for reconstruction in the NOvA experiment.

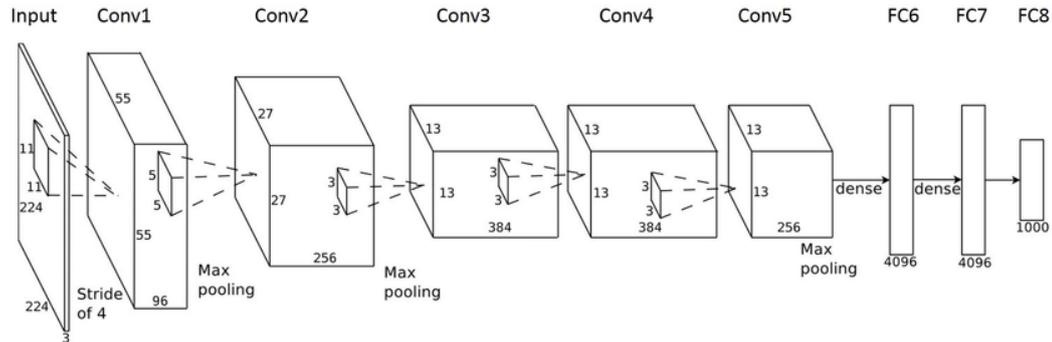


Figure 6.10: Schematic overview of the AlexNet convolutional architecture. Courtesy of [145].

Following an AlexNet inspired event CVN the next iteration was based on GoogLeNet [146]. The GoogLeNet network heavily relies on inception modules. These modules consist of several convolutional layers that are applied in parallel instead of consecutively. This effectively allows for the usage of multiple differently sized kernels to be applied in convolutional layers on the same input. The modules can be repeatedly applied to construct the full model architecture. Both the full GoogleNet architecture and the inception module are shown in Figure 6.11.

6.4.1 ResNet

Building on the earlier employment of inception modules the ResNet architecture was developed. ResNet refers to architectures which employ residual learning in some parts of the architecture. It does this through the introduction of residual learning blocks. Like the modules of GoogLeNet an input is split and passed to different layers. In the case of a simple residual learning block the input is passed into two sequential layers. The output of the second of these

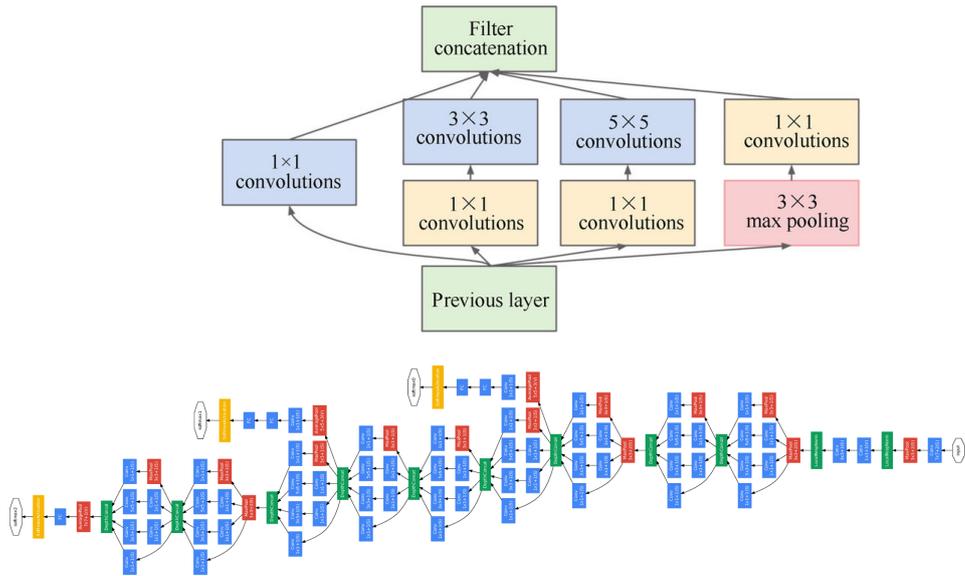


Figure 6.11: An inception module (top) [147]. The full GoogLeNet architecture with multiple output branches (bottom)[148]. The input is located on the right and the outputs on the left.

two layers is then added to the initial input. This effectively provides a path for information to bypass these two layers. Figure 6.12 shows an example of a residual learning block.

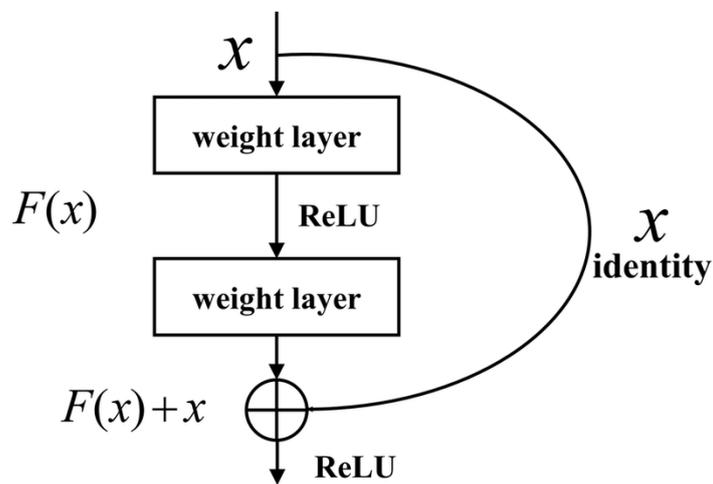


Figure 6.12: Example of a residual learning block. Weight layers can be replaced with convolutional layer and/or multi-layered structures. Courtesy of [149].

When multiple of these blocks, or bypasses, are build into the network information is provided with a simpler path through the network whilst not

losing the analysing power of the intermediate layers within the blocks. The performance improvement has been shown to be especially strong for deeper network architectures. A summary of the ResNet architectures for various depths is shown in Figure 6.13.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 6.13: Overview of various length ResNet architectures.[150]

6.4.2 Modified MobileNet

As neural network usage for the purpose of image recognition has become more widespread, so too has the need for improved network optimizations. Many industry application place tight constraints on the resource usage of the network. Moving for example from the typical parallelizable run environment of a graphics processing unit (GPU) to lower power CPUs on mobile devices. While training of a network could be completed on GPUs, deployment for wider usage would be run on lower resource environments. Even though inference is much faster and less computationally intensive than training, deployment still falls short of the aforementioned constraints. Deeper networks with higher parameter counts tend to have slower inference speeds as more computations are required to perform a single forward pass on a data sample. Additionally all the network parameters need to be stored in memory, which can prevent usage outright in lower resource environments.

MobileNets can help mitigate these issues by simplifying the convolutional operations and therefore drastically reducing the overall network size [151].

A potential slight reduction in performance is the expected trade-off. Faster training due to the reduction in parameters and operations remedies the performance reduction in many scenarios. The convolutional operations which are applied in the convolutional layers of a CNN are depthwise separable. It is this feature that MobileNets exploit to actualize the optimizations. There are two types of separability with respect to convolutional operations. The first are the spatially separable convolutions, these are not generally applicable but conceptually simpler. A convolutional layer applies matrix multiplication on a part of the input tensor with a kernel. In the case of spatially separable convolutions this kernel can be decomposed into two lower dimensional kernels. A matrix multiplication of the two kernels will reproduce the original higher dimensional kernel. The total number of associated parameters and operations to be completed for a single convolution are lower for the sequential application of the decomposed kernels than for the original. An example of a spatially separable kernel is the Prewitt kernel[152] which is employed for the purpose of edge detection in images,

$$\begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} +1 & 0 & -1 \end{bmatrix} \quad (6.19)$$

In this case the total number of parameters is reduced from nine to six which also reduces the number of elementwise multiplication operations which have to be completed.

6.4.2.1 Depthwise separable convolutions

Whether or not a convolutional operation is spatially separable depends on the decomposability of the underlying kernel. When the components of the kernel are learnable parameters spatial separability can thus no longer be used. The other type of separability does not have this limitation and can therefore be applied throughout the CNN. This type is the depthwise separability. Here the original multiplication with the kernel is replaced by two sequential multi-

plications, the first being a depthwise convolution and the second a pointwise convolution. This factorizes the original convolution into a channel-wise filtering operation and a linear combinational operation. The reduction gained by this factorization can be examined through a comparison of the computational cost.

For an input tensor I of shape $D_I \times D_I \times C_I$ to which a convolutional kernel K of size $D_K \times D_K \times M \times N$ is applied, an output tensor F of size $D_F \times D_F \times N$ is produced. To simplify the comparison the kernels are assumed to be square, the spatial sizes of the input and output tensors are taken to be equivalent ($D_I = D_F$) and convolution is applied with padding and a stride of one. Even without these simplifications the reduction is still present. The non-spatial sizes are referred to as the depth or channels of the input and output tensors. These channels are analogous to the RGB colour channels for an image, in which case the depth would be three. The computational cost of performing a standard convolution of K on I to produce F is found to be equivalent to,

$$\text{Cost}_{Conv} = D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \quad (6.20)$$

The decomposition of the full convolution into the combined depth and pointwise convolutions breaks the multiplicative scaling of the number of output channels and size of the convolutional kernel. The depthwise convolution applies a kernel to filter the input into an intermediate map for each of the channels M of the input tensor. The computational cost of this shallow convolution is,

$$\text{Cost}_{DCov} = D_K \cdot D_K \cdot M \cdot D_F \cdot D_F \quad (6.21)$$

The main reduction is gained from the operation not scaling with the number of output channels N . A pointwise convolution is then applied to components of the produced intermediate mappings. The kernel of the pointwise convolution is of size $1 \times 1 \times M$ and produces linear combinations of the depthwise convolutions output maps. Repeated application of the pointwise kernel then produces the

output tensor N . The cost of the pointwise convolution is,

$$\text{Cost}_{PCConv} = M \cdot N \cdot D_F \cdot D_F \quad (6.22)$$

Both methods for completing a convolution operation are shown in Figure 6.14.

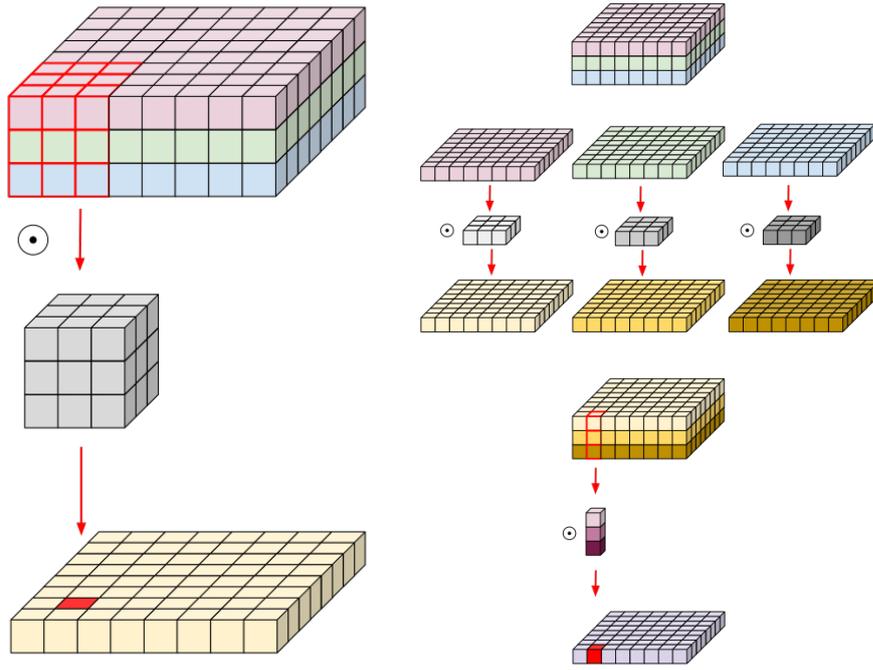


Figure 6.14: A typical convolutional operation (left) and the depthwise separable convolution (right) are shown. The input is shown on top, the convolution operation is designated by the \odot symbol and the output is shown on the bottom. The depthwise separable consists of the depthwise convolution and the pointwise convolution. The reduction in parameters is most easily visualized when the case of multiple output channels is considered ($N = 1$ to $N = 2$).

A final comparison of the reduction is then found to be,

$$\begin{aligned} \text{Fractional reduction} &= \frac{\text{Cost}_{DCConv} + \text{Cost}_{PCConv}}{\text{Cost}_{Conv}} \\ &= \frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} \\ &= \frac{1}{N} + \frac{1}{D_K^2} \end{aligned} \quad (6.23)$$

For the cases of convolutional layers with large kernel sizes and/or a large number of output channels the reduction is significant. This is the case for MobileNet and therefore all the convolutional operation within that network architecture have been implemented via this separational strategy.

6.5 ML datasets

To commence training of a neural network several datasets need to be defined. The first distinction is made between data available for training and for inference or deployment. Typically the training data is a small subset of all possible data to which the network could be exposed. For the case of (semi-) supervised learning the training data has been curated and labelled. The full training data is then split further into a training, test and validation dataset. The training dataset is comprised of data samples and labels on which the network will be training and learning. The test and validation datasets are comprised of samples from the training data which are not contained in the training dataset. The network does not learn from samples in either the test or validation datasets. Instead these datasets are used to provide an unbiased evaluation of the network's performance. The validation dataset is used to determine which values of hyperparameters are optimal for the current training environment. Once optimal hyperparameter values are determined the network's performance is evaluated on the test dataset. In the field of artificial intelligence of which machine learning is a part, the test and validation terms are often used interchangeably [153]. In the general case it is safe to assume that either refers to training data not contained in the training dataset and on which the network has not trained.

6.5.1 Generalization

The deliberate splitting of the training data is used to evaluate both the amount of under/overfitting that is occurring during the training and the degree to which the network is able to generalize. Underfitting occurs when the network is unable to identify the relations and correlations between the input and

the output of the samples in the training data. Overfitting is the opposite scenario, in that case the network memorizes individual samples and identifies more relations than necessary. Typically underfitting is encountered early during training and overfitting at later stages. For either case the network will be ill equipped to perform inference on data samples it has not previously encountered. When this is indeed the case the network fails to generalize to a sufficient degree and cannot be widely deployed with confidence. The scenarios of under, appropriate and over fitting are visualized in Figure 6.15.

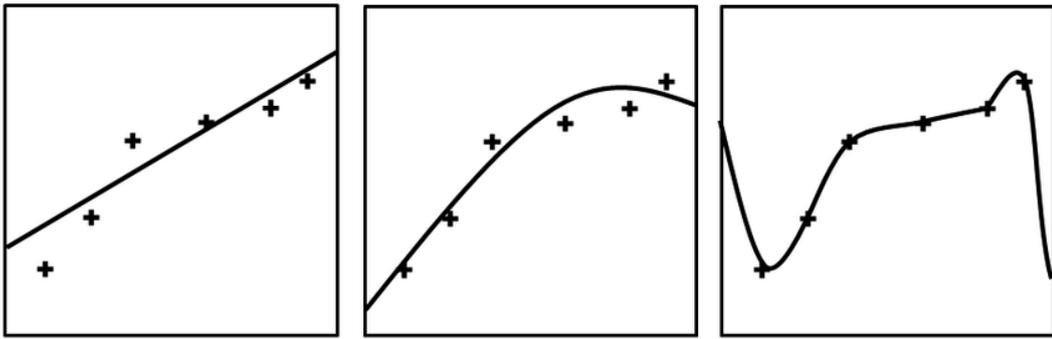


Figure 6.15: Simplified examples of under (left), over(right) and appropriate(middle) levels of fitting. Courtesy of [154].

6.6 Domain adaptation

Even for a network that does not suffer from under/overfitting a satisfying degree of generalization is not guaranteed. The training data is only a small subset of the total possible data and therefore is likely to deviate from the true statistical distribution of the total data. The training data is formed from a small sampling. This sampling is not guaranteed to be random either, selection biases can be easily introduced when constructing the training data. For example through quality cuts applied during curation or in the case where the training data is simulated to a certain degree. Further sampling to produce the training, test and validation datasets can result in different distributions for each of these datasets. As these datasets are drawn from a single source they are expected to approximate the source distribution with some small deviations. These deviations are partially exploited to evaluate the aforementioned

under/overfitting and generalization.

In an ideal training scenario the network would identify and learn from correlations present in the training data to produce internal features. If the correlations and features do not rely on any specific deviations of the training dataset distribution and instead are a result of the approximated source distribution, they should work (almost) equally well on the test/validation dataset and the wider data during deployment. Thus resulting in a well behaved network with a high degree of generalization.

This is not guaranteed to be the case for all training scenarios, the network does not have an innate preference for learning a feature based on its origin. Instead the net improvement learning has on the network's output drives the optimization procedure. Furthermore the assumption that the training data is approximately representative of the more general data is not necessarily true for all scenarios. Domain adaptation refers to scenarios where that assumption is explicitly not true.

It is helpful to first define what a domain is, as domain adaptation covers a large range of similar but different cases. In its most general form a domain refers to a dataset which is separable from another dataset through one or more specific categorizeable differences. The distinction typically refers to datasets with a difference in their statistical distributions. Domains can be either discrete or continuous depending on the nature of the identified difference. A shift between domains is possible when moving from one domain to another.

An example of domain shift is the deployment of a neural network which is trained on simulated data and deployed on real data. In this scenario the simulated data forms a domain and the real data another. The simulated data approximates the real data but includes some degree of mismodelling. The differences between the two domains and their data distributions gives rise to uncertainty with regards to the network's performance once deployed on the real data. It is reasonable to expect the network to have learned some of the mismodelling during training which will not generalize to the

real data. The subsequent loss of performance is not a priori known. The network's retention of performance across domain shifts is defined to be its robustness. While a sophisticated simulation might minimize the domain shifts magnitude and therefore potentially minimize any loss of performance there is no guarantee of similar performance on the real data. E.g. domain shifts that would be considered insignificant in most cases could result in drastically worse performance [155], deviating significantly from initial expectations. Additionally other scenarios exist where a large difference between one or multiple domains is expected, resulting in subsequent variations in network performance.

Domain adaptation encapsulates all methods and techniques that attempt to increase robustness across domains through focussing on learning which is generalizable across domain shifts. Three related approaches of domain adaptation are examined in further detail.

6.6.1 Adversarial domain adaptation

Multiple neural networks can be simultaneously employed to achieve a complex task for which a singular neural network is insufficient. The domain adaptation is an example of a complex task where multiple goals need to be balanced to achieve an optimal solution. The desire for a high performance classifier is balanced with the need for it to be robust against specific domain shifts. The multi network approach examined here is based on a prior adaptation of general Generative Adversarial Networks [156] for the purpose of domain adaptation [157]. The approach consists of two competing networks with opposing goals and they are therefore referred to as being adversarial with respect to each other. An example of a GAN is displayed in Figure 6.16.

The first network in the ensemble is a typical classifier, named F. The output of the classifier forms the input for the adversary R. With this input the adversary attempts to predict from which domain the propagated sample originates. As the adversary improves and learns, it becomes increasingly capable of extracting and exploiting domain dependent information present in the classifier's output. Subsequently this information is used to update the

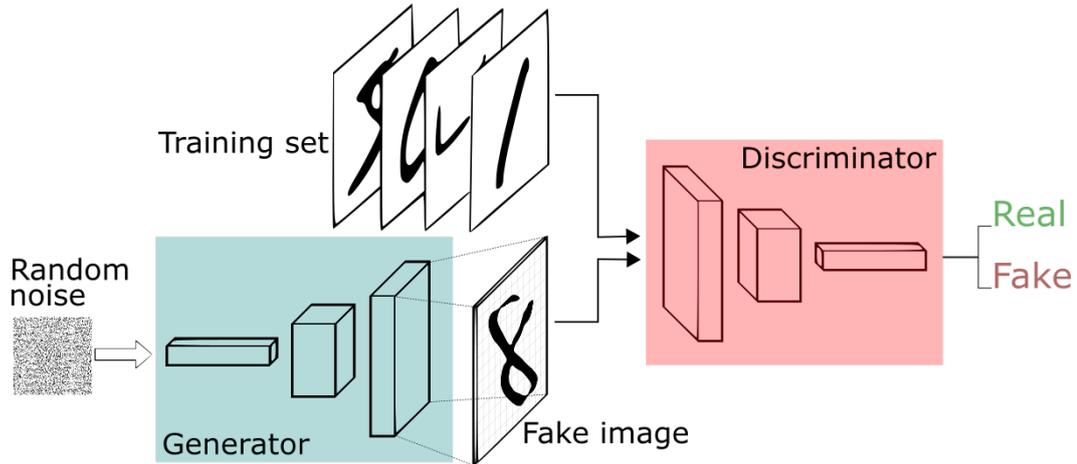


Figure 6.16: Overview of generative adversarial networks for the purpose of generating fake handwritten number images. Courtesy of [158].

classifier such that it produces outputs which contain less domain dependent information. The classifier therefore becomes more robust. Additionally this output reduces the adversary’s performance to correctly predict the domain. This is achieved through applying back propagation on the classifier with a combined loss function from both the classifier and adversary. A relative strength parameter λ is used to assign priority to the loss contributions of either the classifier or adversary.

The general training procedure is then implemented as,

1. Pretrain the classifier with loss of L_F .
2. Pretrain the adversary on the current output of the classifier with loss L_R , while the classifier is frozen².
3. Freeze the adversary in its current state, unfreeze the classifier and train the classifier with the combined loss function $L_{total} = L_F - \lambda L_R$.
4. Freeze the classifier and unfreeze the adversary. Train the adversary with loss L_R .
5. Repeat steps three and four until training reaches suitable levels of convergence and stability.

²A network or its weights are frozen when they are unable to be updated through training.

The training procedure is shown in Figure 6.17. The flow diagram shows the different stages of training.

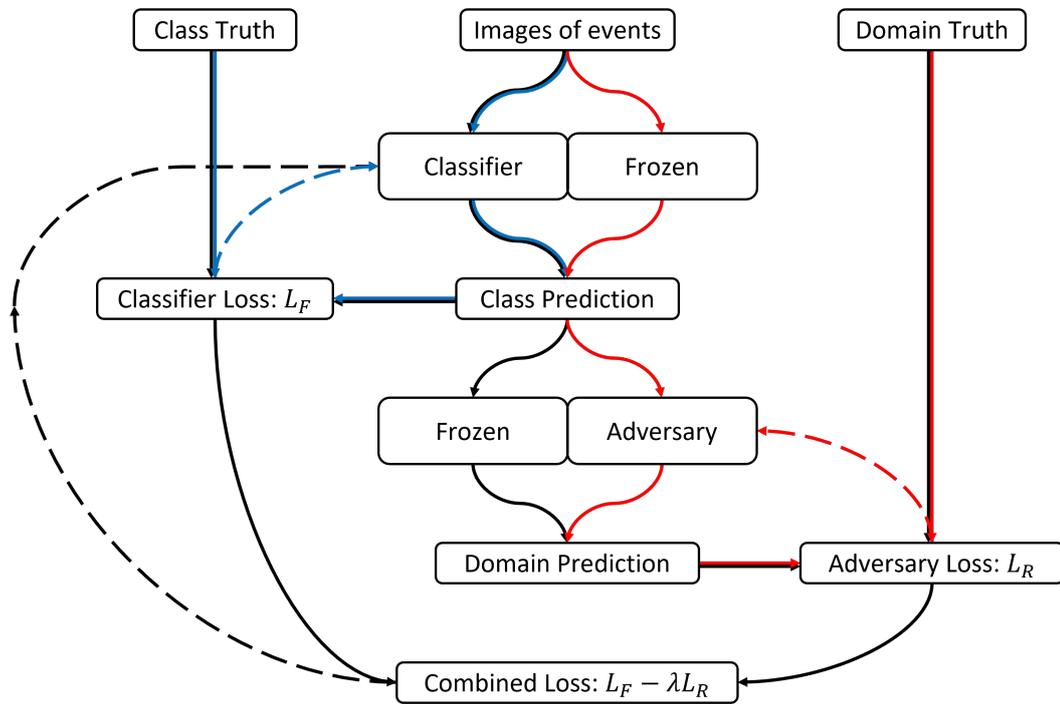


Figure 6.17: Schematic representation of the multi network adversarial domain adaptation training procedure. The stages are colour-coded. The pretraining of the classifier is shown in blue, the pretraining of the adversary (including throughput of the frozen classifier) is shown in red. The combined training of the classifier (including the frozen adversary) is shown in black. The dashed connections indicate which network is being updated through the specified loss functions.

6.6.1.1 Mode collapse

The two network approach suffers from mode collapse, similarly to other adversarial trained network ensembles such as the Generative Adversarial Networks GANs [156]. Mode collapse occurs when the delicate learning balance between the two networks is disturbed. Each network's training relies on the current state of the other network and therefore neither network can be allowed to learn too quickly. If the performance of either network changes faster than the other the balance is lost and training is likely to fail. The training scenario itself can lead to mode collapse even if the network training remains in lock-step for both networks. Each network has a different task with

a varying degree of difficulty and thus a large difference can result in a failure of training. Additionally a large variation in complexity between networks could also result in an unbalanced training. An example of these cases is the case for particle physics classifiers. Here there is typically a high performance, fast learning yet complex classifier (due to the high levels of optimization applied) coupled with the more difficult task of domain robustness. The complexity of the classifier combined with a relatively smaller domain shift can give rise to unstable training.

6.6.2 Gradient reversal

To resolve the issue of mode collapse a single network approach following the methodology of adversarial training is employed [159]. This single network approach is named gradient reversal (GR) and it moves the objective of the adversary into a separate output branch of the network. The general architecture consists of a set of shared consecutive layers for the purpose of feature extraction. These feature-extracting layers branch off into a conventional and a domain classifier. Training with this architecture results in the resolution of both the class and the domain. In between the connection point and the start of the domain classifier a gradient reversal layer is placed. This layer allows information to pass through unaltered during the forward pass. During backpropagation it flips the sign of all the gradient components and multiplies them by the factor λ . The placement of the layer results in the domain branch minimizing the domain classification error. A general example of an architecture including a gradient reversal layer is shown in Figure 6.18. During backpropagation the layers comprising the domain branch are located before the GR layer is encountered. Thus the reversal and magnitude change of the domain component of the backpropagating gradients affects the feature extracting layers. This results in those layers minimizing the classification error and maximizing the domain error. The maximization of the domain error caused by the GR results from updating the parameters in the opposite direction of the steepest descent in the loss surface for the domain classification error. This results in the network

being unable to learn any features that are not robust against a domain shift in the feature-extracting early layers and therefore the subsequent classifier becoming more robust due to not having access to those domain dependent features. The general methodology is applied across various fields in which domain shifts are present, high energy physics being amongst these [160].

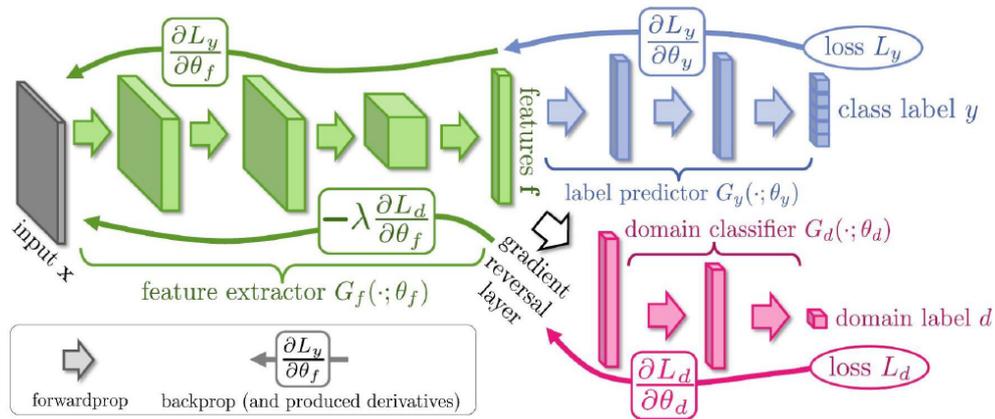


Figure 6.18: Schematic overview of adversarial domain adaptation in a single network architecture. The gradient reversal layer forces parameter updates to the previous layers which maximize the domain loss and thus are less domain dependent. This results in feature extraction layers which are robust against a domain shift. Courtesy of [161].

For the scenario where domain dependent features form the foundation for classification partial domain reduction of these features in the initial layers might be inadequate to achieve optimal robustness of the classifier. As training proceeds and the partial reduction is achieved the domain classification task becomes increasingly difficult as less domain information is progressing through the initial layers to the domain branch. This results in less reduction in domain dependence of the initial layers. This is expected to conclude into an optimal equilibrium point. Any small remaining domain dependence in the initial layers could be disproportionately beneficial for the regular classification branch. If this is the case the classifier branch will learn to give more weight to those less robust features. Backpropagation through gradient descent does not result in the learning of a specific feature based on its domain dependence, only its power to minimize the classification error (for the classification branch).

The hyperparameter λ is not a priori known and is expected to be different for each specific scenario. This difference can span several orders of magnitude (see appendix C of [159]). Alternatively to a direct influence on the magnitude of the domain gradient it can be thought of as denoting and moderating the relative importance of the classification task compared to the domain classification task.

6.6.3 Loss penalization

Instead of a separate gradient reversal layer and domain classification branch which relies on backpropagating the domain dependence reduction via the initial layers onto the classifier the same method can be applied through the loss function alone [162]. This requires no modification to the network architecture and propagates changes directly onto all trainable parameters of the network. The original loss function is augmented to include a domain penalization term which introduces a direct cost for the network output variation on a domain shift. The method derives its name from this term, loss penalization.

For the input x which can be domain shifted by Δ the network has an output of $f(x + \Delta)$. This approaches $f(x)$ when the network is robust against the domain shift. To construct a general domain penalization term in the loss function the variation of the network output under a domain shift needs to be quantifiable, even for a single sample. Additionally it needs to satisfy all requirements to be used with gradient descent and backpropagation. This includes differentiability of the function. For a batch of samples B originating from a single domain the output of the network $f(x_i + \Delta_i)$ is collected and used to fill a single histogram for each domain. For a robust network under a domain shift when allowing for statistical fluctuations these histograms should be equivalent. For the scenario of multi-class classification it is prudent to split these histograms down into a class specific domain histogram. The configuration of these histograms, such as the number of bins K is a hyperparameter similar to the batchsize. Histograms are not directly differentiable and so need to be modified before they can be used as input for the loss penalization term.

A Gaussian blurring filter is applied to each entry of the histograms. The blurring employs Gaussian functions $\mathcal{G}_k(x)$ which have a mean equal to the bin centers, a standard deviation equal to the half width of the histogram cell a maximum value equal to one $\max(\mathcal{G}_k(x)) = 1$. Summation of a single function for each bin entry b in the histograms produces a count estimation which is fully differentiable,

$$\mathcal{N}_k(f(x)) = \sum_{b=1}^B \mathcal{G}_k(f(x_b)) \quad (6.24)$$

These count estimates form the input to the loss penalization term,

$$L_{LP} = L + \lambda \Lambda(x, \Delta) \quad (6.25)$$

The form of $\Lambda(x, \Delta)$ is typically chosen to be a function which has higher values for larger separation of the count estimates. A network which is less robust with respect to a domain shift is expected to exhibit larger separations in the count estimates and therefore would produce higher loss values. The default shape of was chosen to be,

$$\Lambda(x, \Delta) = \frac{1}{K} \sum_{k=1}^K \left(\frac{\mathcal{N}_k(f(x)) - \mathcal{N}_k(f(x + \Delta))}{\mathcal{N}_k(f(x))} \right)^2 \quad (6.26)$$

Any deviations from this form will be marked.

Large bin numbers, small batchsizes, many classes or a large amount of domains invite statistical fluctuations as a result of insufficiently filled histograms. If sufficiently large these could negatively impact accuracy of the training with the inclusion of the loss penalization term.

This concludes the last domain adaptation technique covered here. The following has been discussed in this chapter, the components of neural networks and their training, usage of neural networks in the NOvA experiment through the EventCVN and its architectural evolution and finally the domain shifts and various techniques which can reduce their negative impact.

Chapter 7

Domain shifts and adaptation in NOvA

With general and NOvA specific neural network concepts as well as domain shifts and adaptation techniques clearly defined it is possible to examine domain shifts in the data collected in the NOvA experiment. To this end the data collection and selection is explored in section 7.1 with an in depth focus on the applied preselection cuts which ensure a sufficient and consistent data quality. The subsequent processing of the data and the format in which it is stored prior to, and during, usage in CVN trainings is covered in section 7.2. The datasets representing the domain shifts which are used in the trainings and robustness studies are shown in section 7.3. Additionally in section 7.4 the AdCVN framework is introduced and discussed as it covers the implementation of domain adaptation techniques within the context of CVN in addition to network training and evaluation techniques.

7.1 Data selection

The data production and collection efforts in the NOvA experiment produce vast quantities of events. Not all of these events are equally valuable to the experiment and its analyses. To include a raw data or simulated event in an eventual analysis a significant amount of computational resources and time need to be expended. Optimization of this process is therefore crucial for the maximization of the available statistics. To this end cuts are applied before the full reconstruction is run on the raw events. The cuts function analogously

to the triggers in the data selection process and are effectively filters which allows for the selection of sufficiently “good” events to be passed on to further processing. The definition of “good” is given through a collection of these predefined cuts typically referred to as preselection cuts. Individual analyses can build on these and apply further stricter cuts for a more stringent event selection, an example of which can be seen in [70].

The preselection cuts are applied in the following order.

- Data quality.
- Event quality.
- Containment.
- Cosmic rejection.

For subsequent usage in analyses, specific cuts are applied after the cosmic rejection stage. While some of the preselection cuts are specific to data collection and not data simulation, simulated events should pass these cuts regardless. Event usage for the purposes of CVN training generally does not require additional analysis specific cuts, however some processing is applied to improve the trainings. These are described thoroughly in section 7.2.

7.1.1 Data quality

There are various scenarios conceivable in which data is recorded which should be excluded from further analyses. The data quality cuts guard against these occurrences.

The first is a Bad Channel cut. If an APD is recording too many or too few hits on average for the events in a (sub)run it is considered to be cold or hot respectively. The threshold for hot is set at $10^{3.5}$ times per event and for cold at less than three times per event. All APDs used in the detectors during data taking are monitored by OnMon, an online monitoring tool which runs as a component of the DAQ system. OnMon employs hitrate maps for every cell in both detectors to determine which channels are operating outside of

the specified range. These channels are then marked as bad and committed to a separate database. Updates to this database are made from repeated validations against the current hitrate map.

The second data quality cut is the Good Runs cut. This criterion ensures that the operating conditions of the detectors at the time of data collection is consistent with the standard data taking conditions. The entire suite of criteria is documented [163] [164] and varies for the gain conditions of the detector. Examples are non-empty triggers, time-ordering checks of the first and last recorded events, validity of the event timestamps, median ADC hitrate is within expectations (gain dependent), minimum size of active analysable detector, fraction of 2D tracks below 15% and the slice production rate.

The third data quality cut is the spill cut. This cut removes any spills where the detectors or beam failed to produce a good recorded beamspill in the detectors. The spill cut includes checks on the beam center position (x,y), the beam width (x,y), the horn current, the number of delivered protons on target for the current spill and the duration of the spill. Additionally it checks for out of syncs DCMs, events with all empty DCMs, incomplete events and partial masking of the FD.

During data collection a NOvA collaborator is monitoring the DAQ systems at all times, split into three eight hour shifts. If data quality issues arise which are not picked up through the automated cuts runs of data can be manually marked as bad and would be subsequently excluded.

7.1.2 Event quality

The event quality cut ensures the slice contents meet the basic requirements to be useful for further reconstruction. An example of a common detector failures are the APD flashes [165]. These occur when a large amount of charge is deposited on an APD. The resulting saturation produces additional recorded hits spanning neighbouring channels within a time window of approximately $30\mu\text{sec}$. To remove these excess noise hits a veto is set for this saturation window. For ν_μ events in both detectors the total reconstructed energy in

a slice should not exceed 5 GeV, it should contain a reconstructed Kalman track under the muon assumption and consist of at least 20 cell hits over 4 consecutive detector planes. For ν_e events the criteria vary per detector. The FD ν_e events should be between 1 – 4 GeV, consist of at least 30 hits with a maximum of 150 hits and the longest contained prong length must be between 1 and 5 meters. The ND ν_e events should be below 4.5 GeV, consist of at least 20 hits up to a maximum of 200 hits and the prong length requirements is the same as for the ν_e FD events [166].

7.1.3 Containment

Containment cuts ensure that the recorded activity, namely the contents of the recorded slice, is primarily located within the active detector volume. Uncontained particles can carry energy out of the detector which frustrates any subsequent reconstruction efforts. Specifically the total energy of the event can no longer be correctly reconstructed. Additionally events with the primary vertex located outside of the detectors, referred to as rock events, can be rejected using the containment criteria. The specific conditions differ for the two detectors.

For the ND along the beam direction the number of planes to the start of the detector must be larger than one, likewise for the end of the detector. This results in containment along the beam direction z . For reconstructed Kalman tracks the forward extrapolation should be larger than five cells and the backward extrapolation should be larger than ten cells. These projections ensure the track positions are far enough from the detector edges. The forward extrapolation counts the distance along the reconstructed track endpoint whereas the backwards extrapolation counts from the track startpoint.

Additionally the Kalman track start position should be closer than 11 m and the end position should be closer than 12.75 m as measured from the front of the ND along the Z direction. Due to the height difference between the muon catcher and the top of the ND it is possible for a muon to be recorded which can cross the air gap between the two, these are excluded as well. For reconstructed

showers their stop and start positions should be between -1.8 m and 1.8 m for the X-view and Y-view. For the z direction they should be between 0.2 m and 15.25 m .

For the FD similar containment criteria to the ND are used, such as the single plane from the detector edge along the z direction. The forwards and backwards extrapolation condition is changed to six cells for the reconstructed Kalman tracks and is introduced for the reconstruction of the tracks using the cosmic track algorithm [167]. The cosmic backwards extrapolation is set to a seven cell minimum. For the prong start and end points a minimum distance is set for each face of the detector, the top is 60 cm , the bottom is 12 cm , the eastside along the positive z direction is 16 cm , the westside is 12 cm and the front and back are set at 18 cm . The values of the containment criteria are re-optimised for the FD to increase the number of well reconstructed ν_μ CC events that are selected and to decrease the number of cosmic events which are incorrectly included in that selection.

7.1.4 Cosmic rejection

As the FD is located on the surface there is a significant amount of cosmic activity being recorded in the detector. While timing based selection of data can remove a large amount of this activity, the coincidence with signal interactions forms a significant background. A neural network based approach (Cosmic CVN) is applied to reject cosmic events before calibration and reconstruction has been applied. This greatly reduces the computational resources expended compared to a later stage cosmic rejection cut. The network trained for this purpose is functionally similar to the CVN networks, sharing the two view siamese convolutional architecture. The overall architecture of the network is based on the ResNet18 [168] architecture. As it is meant to be deployed on pre-calibration and basic reconstruction data, the training data differs from the normal CVN training data [169]. It consists of pixelmaps for each detector view which cover the entire detector, $384\text{-by-}448$ pixels. The values of these pixelmaps are binary, 0 for no energy recorded and 1 for any amount of en-

ergy recorded. The window of time is set to sixteen microseconds with a one microsecond overlap boundary on each side. A full FD event is thus imaged by several (31) [170] partially overlapping pixelmap snapshots. Activity in each sixteen microsecond window is then classified as cosmic- or neutrino-like. The cosmic-like windows are removed before further processing. Training and testing on a dataset containing 4.48 million dual view pixelmaps originating from both FHC and RHC yielded a performance of 92.6% training accuracy and 87.8% validation accuracy [170]. Validation on FHC (RHC) data from period 2 up to production 5 demonstrated a cosmic rejection of 94.9% (95.0%) for the cosmic CVN. Cosmic rejection is equivalent to the cosmic signal efficiency. The previously employed methods, such as CosVeto which employed the Cosmic-Track reconstruction module to filter and veto cosmic like activity, achieved a cosmic rejection of 90.8% (93.6%). Furthermore the selection efficiency for various neutrino types improved compared to that of the previously employed methods, as can be seen in table 7.1.

Type	CosVeto No Cuts	CosVeto w/ Preselection	Cosmic CVN No Cuts	Cosmic CVN w/ Preselection
ν_μ FHC	93.22	n/a	99.2	99.84
ν_μ RHC	92.82	99.84	99.2	99.88
ν_e FHC	93.21	99.74	99.71	99.96
ν_e RHC	99.82	99.86	99.82	99.99

Table 7.1: Cosmic rejection selection efficiency for period 2 up to production 5 data, with and without preselection cuts applied. Recreated from [170]. The selection efficiency for ν_μ with preselections applied was not reported in the validation study.

7.2 Datafile formatting and processing

The preselection and later reconstruction is run using modules and packages developed in the C++ based ART event processor framework. This framework operates primarily on ROOT files. These files are self-descriptive and save data in a compressed binary form, which allows for efficient large data storage with standalone readability. These qualities paved the way for ROOT to become

the default analysis tool for many large scale physics experiments [171].

The ROOT data files produced by the sequential application of ART modules which constitute the reconstruction contain all relevant information and their provenance in an event-by-event format. This contained information is required as an input for any further analyses, however the current data format is not optimally configured for these next stages. To that end a reconfiguration is applied and a common standard employed in which the further analyses are constructed and run. The results of the final reconstruction stage which are ART ROOT files are reconfigured to the Common Analysis Format (CAF). Instead of the event-by-event ART ROOT structure, these ROOT files contain reconstruction variables and other information in native ROOT tree and branch format. This enables faster and efficient access to the relevant variables (and collections of events) required by later analysis steps, removing large amounts of extraneous information from the processing loops. Additionally the optimization of analysis pipeline results in a lower scaling with the number of events, which is preferable with the increasing amounts of data used to produce recent analyses.

Beyond the computational benefits from reconfiguration to the CAFs and usage of the CAFAna framework, the common standard enables the possibility of later joint analyses.

The CAF files contain the reconstruction information required for Event CVN trainings, primarily labelled pixelmaps. Nevertheless the direct usage of CAF files for the trainings and evaluations is undesirable. The pixelmaps only form a fraction of the total CAF file size and therefore the total file size is relatively large compared to the fraction of usable information contained. The repeated fetching of single event information during training and evaluation is inefficient which results in a considerable slow down and waste of computational resources. Additionally the neural network trainings are typically performed on dedicated specialized hardware which might have memory constraints and requirements for the dataset storage locations. Both of these would be hindered by unnecessarily large datasets. And finally there is no guarantee that the

neural network trainings are performed in frameworks which are efficiently compatible with the ROOT based CAF files. Machine learning is supported by ROOT through the Toolkit MultiVariate Analysis (TMVA) library and compatible with python through PyROOT. Other file formats are more efficient when utilizing python based machine learning on dedicated hardware.

The hierarchical Data Format (HDF5)[172] is designed to efficiently store, organise and compress large amounts of data in a portable self-describing format for usage in modern high performance computing environments. Conversion of CAF files into HDF5 files allows for the opportunity of a filter to even further reduce the filesize before usage. Additional compression techniques such as zero-suppression of the pixelmaps are used to further decrease the file sizes. A large number of pixelmap entries are zero which can be removed from the data files as the pixelmaps have known widths and heights. A zero entry in the pixelmap can therefore be re-classed as an empty (null) entry.

7.2.1 Data preprocessing

The trimmed HDF5 data files are streamlined for usage in the neural network training environment. Similarly to other analyses additional data preprocessing is applied before any further usage. This processing is not encoded into the data files like the earlier mentioned reconstruction, selection and reconfiguration. Rather parts of it are applied just before training or evaluation and others can be applied dynamically during the fetching of individual events.

The first preprocessing selection is a strict cut on ν_τ charged current interactions. These interactions are relatively rare due to the low oscillation probability of $\nu_\mu \rightarrow \nu_\tau$ for NOvA's specific L/E and suppressed cross-sections of charged-current τ neutrino interactions at low energies. These are suppressed primarily due to the large mass of the τ particle (1.78 GeV) [173] and result in an energy threshold of 3.5 GeV [47]. This threshold excludes a significant portion of the beam energy spectrum.

Additionally the topologies associated with produced τ particles are similar to the ν_e and ν_μ charged current interactions, due to the short lifetime of the τ

particle and the subsequent decays into electrons, muons, charged and neutral pions.

The second preprocessing selection places further restrictions on the amount of cosmics which are present in the datasets. Even with the preselection cuts the class of cosmic events is relatively abundant in the data. Neural network training with imbalanced classes can be detrimental to the network performance, imbalance during inference does not affect the network performance. The balancing is achieved through a down-sampling of the total number of cosmic events contained in the data, resulting in a 90% reduction of this class. Additionally a cut is placed on the fraction of the transverse momentum from the beam such that any events with a fraction ≥ 0.95 will be rejected. Cosmic interactions produce particles which are typically down-going and thus have a high fraction of transverse momentum. Signal events have a lower fraction of transverse momentum as they are produced by particles which travel along the beam direction and therefore have a momentum which is mostly parallel to that direction.

7.2.2 Dynamical data processing

Following the preprocessing at the start of the training or evaluation, various smaller impact data augmentations are performed dynamically on either the individual events or batches of events.

Noise addition through the random scaling of the input pixelmap values is applied during the official Event CVN trainings. A random value is drawn from a normal distribution with a standard deviation of 0.1 and a mean of 1. All the values in a pixelmap, for both views of the event, are multiplied by this value. This effectively adds uncertainty to the calibration energy scaling, discouraging the trained network from relying too heavily on the pixel values of the pixelmaps to produce predictions. This simplistic approach was inspired by initial studies into the robustness of the neural networks employed in the NOvA experiment. The main drawback of this approach for the robustness improvement is the requirement of a directly accessible encoding of the domain

variation, in this case the value of the energy scale factor.

Construction of a batch of events requires the usage of functions which load in the relevant attributes of the events. These consist of the pixelmaps for both views, the truth label information, the domain truth and the event weights. Additional optional information specifying the event can be loaded in as well, the run, subrun, cycle, event and slice information. Beyond the direct loading-in of the event information format conversions are applied. Specifically the truth labels describing the interaction type are converted from the native neutrino event generator labelling into a categorical summarization. The resulting classes are ν_μ , ν_e , NC and cosmics.

The events produced by the neutrino event generators have a weight associated with them, this is a measure of their relative importance in the larger data sample. Changes to the individual event weights through reweighting produce variations in the rates of different types of interactions, effectively altering the likelihood of certain processes to occur in the simulation. Corrections to the event weights are applied dynamically, where necessary, on an event-by-event basis during training.

7.2.2.1 Domain balancing

For training or evaluations where multiple domains are involved it is beneficial to ensure the batch content is domain balanced, which requires the events forming a batch to originate from multiple domains instead of a single one. Severly imbalanced classes in the training data generally frustrate training efforts and result in longer trainings with lower overall performances [174]. This issue extends to the domains as well, especially for the implementation of domain adaptation techniques which rely on the representations of the individual domains in each batch of sampled data. A typical solution to imbalanced classes is a reweighting of the events. This would result in events which are more abundant being assigned lower weight values reducing their importance during training. Given the higher likelihood of randomly selecting the more abundant events a balance is achieved. Alternatively, resampling

can be applied during the construction of the batch. If a batch of events does not contain a sufficient number of samples from a class, a new batch can be drawn from the dataset and merged with the original batch. This larger batch can subsequently be randomly down-sampled into a batch which is no longer deficient in any specific class.

Likewise upon the exchange of a class for a domain, a domain balanced batch can be produced.

If the imbalance is not caused by merely an abundance of events from a particular class or domain but instead by a severe deficit neither of these techniques can completely remove the negative impact of the imbalance. For the reweighting the few events from the deficient class or domain would be given very high weights, while the resampling would encourage repeated training on a small number of the same events, resulting in the encouragement of the memorization of these rare events.

For the purpose of balancing the domains in the produced batches of events the resampling method is used. Usage of either method is possible as there is no significant deficit in the sample size for the used domains, as can be seen in section 7.3. Preference for the resampling over the reweighting is due to two factors. The first being an entanglement of previous event weighting with the new balancing weights, which could frustrate efforts to isolate and investigate the effects of the original event weights. The second being the production of batches with varying numbers of effective sizes. A batch consisting of many events from an abundant class which have a lower weight assigned to them will have an overall lower number of effective events contained in it. This could frustrate the learning effort as the network will de-prioritise learning on lower weighted samples and therefore lead to longer training times.

7.3 Domain datasets

To enable an examination of the Event CVN robustness against various domain shifts several datasets have been selected. The selection is motivated primarily

by the expected impact of the domain shift on the network’s performance within the context of the broader NOvA experiment. The first domain shift is an exchange of the neutrino event generator, the second is a variation of the calibration uncertainty and the third is a variation of the light level uncertainty.

7.3.1 Neutrino event generator exchange

As mentioned in sections 4.1.1.1 and 4.1.1.2 various neutrino event generators can be used for the production of simulated data. As these event generators employ different methods for simulating the underlying physics it is expected that the simulated events consist of non-identical pixelmaps. Determination of the impact of this variation and the potential reduction of it is particularly pertinent to the efforts of both the NOvA and T2K collaborations in the pursuit of a joint analysis. The NOvA experiment primarily employs the GENIE neutrino event generator whereas the T2K collaboration employs the NEUT neutrino event generator [175]. Other choices such as the NuWro [176] are available as well. In this work the impact of exchanging the NOvA default neutrino event generator (GENIE) with the GiBUU neutrino event generator is examined. The methodology can be applied to exchanges with NEUT or NuWro generators as well.

In production campaign 4 (Prod4) ND FHC data samples were produced with both the GENIE and the GiBUU event generators. After processing through the earlier described pre-selection two datasets of HDF5 files containing the required information for training and evaluation have been constructed. The GENIE sample contains 1150 files containing events with event weights equal to 1. The total number of events contained in this data sample and a breakdown into the interaction classes is shown in Figure 7.1.

The GiBUU sample contains 2102 files with a spectrum of event weight values. The breakdown of interaction classes is shown in Figure 7.2.

During the production of the GiBUU sample the event weight calculation consists of two parts. The first is a direct usage of the event weights intrinsic to the GiBUU event generator, the second a correction for the total cross-section

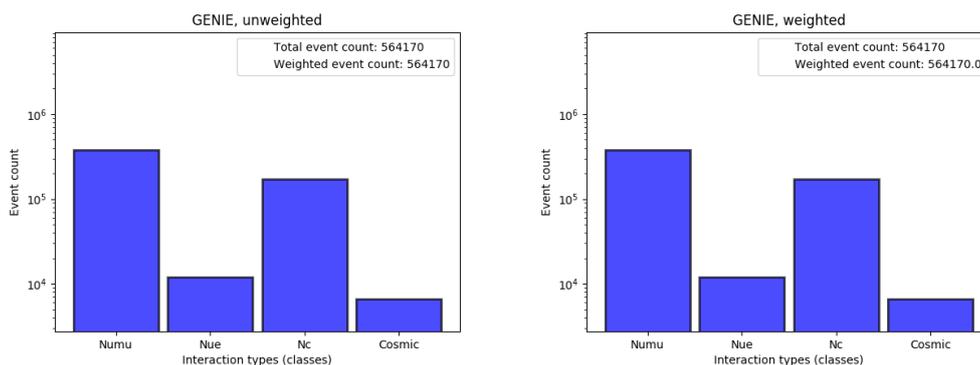


Figure 7.1: The number of events comprising the classes of the GENIE dataset. The GENIE events are weighted with a value of 1. The unweighted (left) and weighted (right) breakdowns are equivalent confirming the event weights in the data files match the expected values.

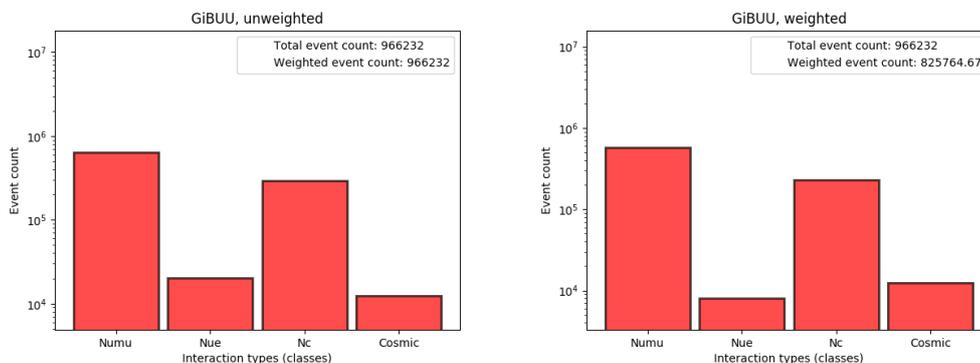


Figure 7.2: The number of events comprising the classes of the GiBUU dataset. The GiBUU events are weighted with a range of values. The unweighted (left) and weighted (right) breakdowns are not equivalent confirming the event weights in the data files are not equivalent to 1.

of the GiBUU sample to the aforementioned GENIE sample. The total cross-section correction calculation contained a bug which resulted in the incorrect usage of the neutrino fluxes, the ν_μ flux component was used for all fluxes ($\nu_\mu, \bar{\nu}_\mu, \nu_e, \bar{\nu}_e$) in the calculation. A flux correction function is included to dynamically correct the induced deviation to the GiBUU event weights during training and evaluation¹. The impact of the flux corrections as applied to the GENIE and GiBUU datasets are shown in Figure 7.3.

¹The implementation of this function is based on the FixGibuuWeight function included in CAFAna.

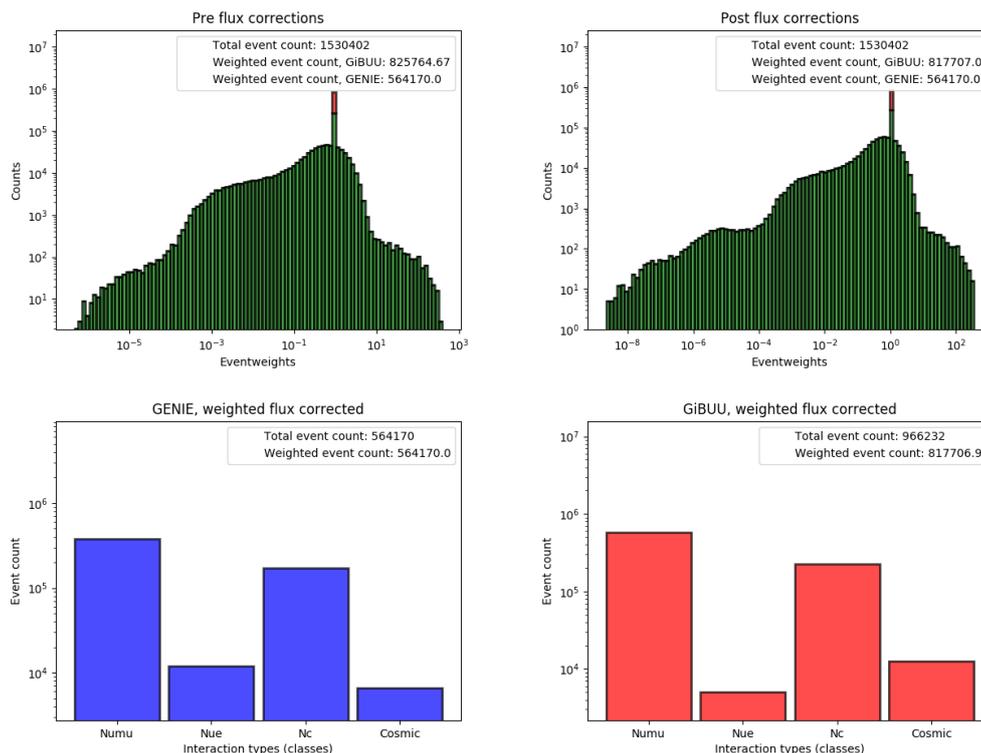


Figure 7.3: The GiBUU eventweight spectra pre and post application of the flux corrections are shown (top). The excess at eventweight 1 are due to a number of GENIE events being included directly in the GiBUU simulated dataset if the GiBUU regeneration failed. The impact of the flux correction on both the GENIE (bottom left) and GiBUU (bottom right) eventweighted class breakdown is shown. As expected only the GiBUU distribution is affected.

7.3.2 Systematic uncertainties

As the NOvA experiment seeks to make accurate measurements, both statistical and systematic uncertainties need to be taken into account. As the statistical uncertainty scales inversely with the square root of the sample size, systematic uncertainties will become dominant to the overall uncertainty in the measurements for larger recorded datasets. It is therefore important for the various analyses to be able to quantify every relevant source of systematic uncertainty. A summary of uncertainty for the 2020 analysis is shown in Figure 7.4.

To evaluate the effects of systematic uncertainties deliberate changes are introduced into the simulation chain. These changes emulate the effects of any

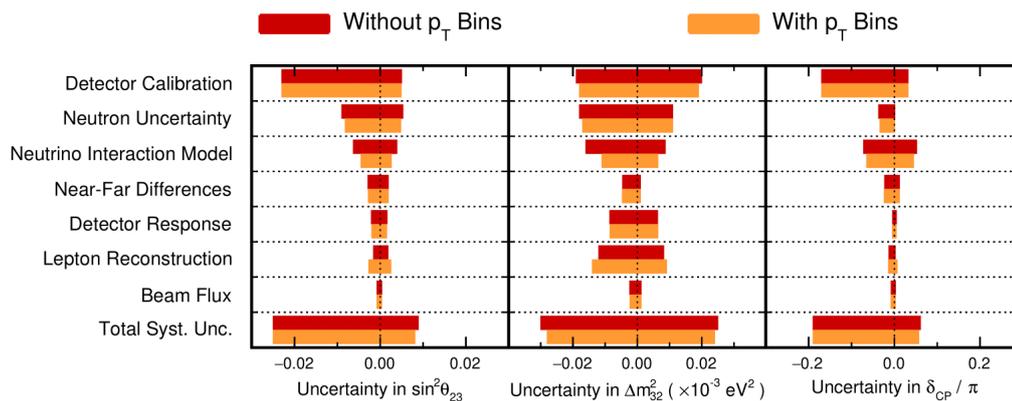


Figure 7.4: Summary of systematic uncertainty sources contributing to the uncertainty on measurements of $\sin^2 \theta_{23}$, Δm_{32}^2 and δ_{CP} in the 2020 analysis. In this analysis extrapolation for the transverse momentum bins are performed separately. [177]

mis-modelling which could give rise to larger uncertainties. The introduced variation in the simulation chain will propagate into the reconstruction and eventually the training and evaluation data samples for the neural networks.

7.3.2.1 Calibration uncertainty

Uncertainty with regards to the calibration encompasses several different effects. These are related to uncertainty in growing deviations over time as the detector components degrade with age, variations in the shapes of the calibration fits and in the determination of the absolute energy scale.

As the detector components age it is expected that their effectiveness decreases. In particular the scintillation oil and WLS fibre are thought to be the primary driving force for the observed degradation. In the FD the recorded number of hits decreases at a rate of approximately 0.24% per year [178]. With any decrease in light yield the threshold at which hits are recorded moves, this introduces variations to the recorded data. As the calibration procedure is iterative in nature the degradation effects are somewhat mitigated as the reduced light yields are incorporated in subsequent calibrations. Nonetheless the loss of a specific subset of hits and the time since the last run of the calibration needs to be accounted for. As the light yield only decreases this component of the uncertainty is applied unidirectionally. It is correlated between the two

detectors.

The attenuation fits are imperfect, particularly at the near and far ends of a cell. At the far end hits can be lost due to thresholding effects as the light has to be transported a longer length to the readout. The orientation of the loop which occurs there can induce cell by cell variations. Comparisons between the reconstructed and true energies present at the regions where edge effects occur guide the imposed magnitude of this component of the uncertainty. The edge effects may vary slightly for both the ND and FD as their cells are of different lengths, however they are correlated between the detectors as the resulting variations are similar for both the ND and FD.

With the help of independent energy scale measurements, referred to as standard candles, the magnitude in deviation of the absolute energy scale can be identified. Some of the more accurate standard candles used for the evaluation of the absolute energy scale are the stopping muons (dE/dx), the energy spectrum of Michel electrons, and the π^0 mass peak. The difference is evaluated between measurements of the standard candles on the data and MC independently. This is done for both the ND and FD, however not all samples contain sufficient statistics for an adequately precise measurement. None of the evaluated standard candles show a perfect agreement between data and MC. The variations are contained within the region of 2 to 5%. The largest recorded discrepancy, proton ND dE/dx , is used to guide the magnitude of the assigned overall uncertainty. It is further split up into a correlated and anti-correlated component. The correlated component varies the total energy scale whereas the anti-correlated component accounts for variations between the ND and FD.

Incorporation of these components into a single calibration uncertainty is achieved by repeating the simulation chain, including calibration and reconstruction. Each identified area of potential mismodeling is incorporated in the simulation chain building constructively with magnitudes set from standard candle measurements resulting in the final overall calibration uncertainty.

For usage as a representational calibration domain shift, three data sam-

ples relating to the calibration systematic are used. These data samples are produced for both FHC and RHC modes, although the FHC has been chosen for subsequent neural network trainings. Nonswapped and fluxswapped files are combined to construct a richer dataset for training and have been defined in section 4.1. In Figure 7.5 the event numbers for each class contained in the three domain samples are shown.

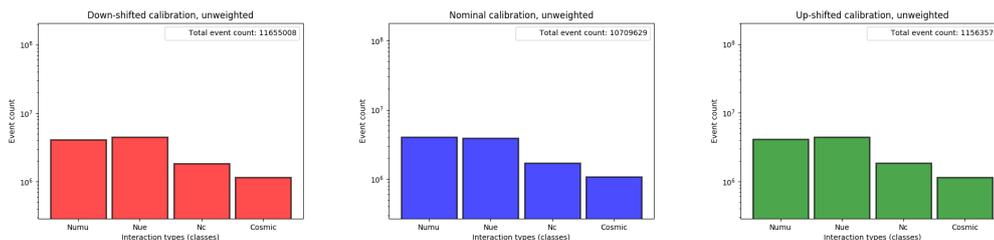


Figure 7.5: The three domains formed from the systematically shifted calibration data samples broken down by the number events for each interaction class.

7.3.2.2 Light Level uncertainty

The uncertainty associated with the light level model employed in the NOvA experiment consist of two components. The first relates to the scintillation light modelling and the second to the Cherenkov light modelling. Photons from both sources are produced in the detectors and are therefore modelled accordingly.

Some of the aforementioned discrepancies (ND proton dE/dx) could be reduced through changes in the light level modelling, indicating a possible source of mismodelling and thus uncertainty. Reduction in the factor which drives the number of Cherenkov photons produced in the simulation resulted in a significant reduction in the ND proton dE/dx discrepancy, $\approx 5\%$ to $\approx 1\%$, while leaving some of the other metrics intact. This factor is equal between the detectors and therefore the uncertainty component is correlated between them.

For the scintillation light modelling the fibre efficiency and light production modelling through equation (4.2) is evaluated for the two views and detectors independently. In the ND light production and transport is shifted by $\pm 10\%$

for both views. In the FD it is shifted by $\pm 16\%$ for the X-view and $\mp 6\%$ for the Y-view. To disentangle the light level effects from the calibration effects, the absolute calibration constant are shifted oppositely. This results in the overall energy response being preserved, while still allowing for the evaluation of thresholding effects introduced through light level shifts.

As with the calibration uncertainty separate data samples are produced to propagate the changes in the simulation. To construct a representative light level domain shift two samples, an up and down shift are used in combination with the FD FHC nominal sample described earlier. Similarly to the calibration samples both modes have been produced but the FHC sample is chosen for training. In Figure 7.6 the event numbers for each class contained in the three domain samples are shown.

Both the calibration and light level HDF5 datasets have been produced as part of Production5.

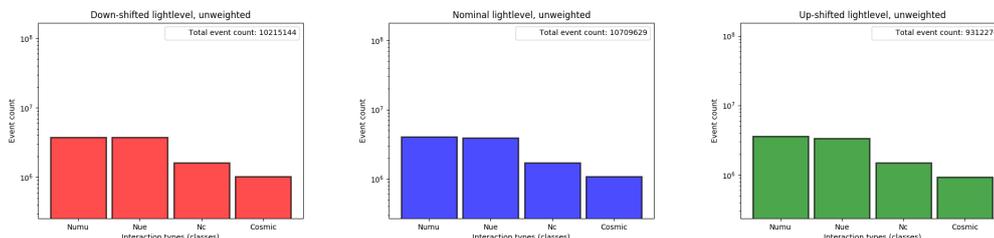


Figure 7.6: The three domains formed from the systematically shifted light level data samples broken down by the number events for each interaction class.

7.4 Computational environment

The developing field of machine learning and artificial intelligence has produced a proliferation in available software and hardware which supports these efforts. For the machine learning development work in the NOvA experiment specialized hardware is available for usage in the Fermilab Wilson Cluster Institutional Cluster (WC-IC). The current purpose of the WC-IC is to support the growing Fermilab-wide high performance computing (HPC) needs with regards to development work. The WC-IC consists of dedicated nodes connected by

low latency high speed Infiniband network fabric. Of the 135 worker nodes connected, 35 have additional hardware enabling acceleration.

- 27 worker nodes with 4 NVIDIA K40's each.
- 1 worker node with 2 NVIDIA P100's with a NVLINK connector.
- 1 worker node with 8 NVIDIA P100's.
- 4 worker nodes with 2 NVIDIA V100's each.
- 1 worker node with IBM Power9 with 4 NVIDIA V100's with a NVLINK connector.
- 1 worker node with Intel Knights Landing consisting of 512 CPU cores.

Additionally a 334TB storage space is accessible from the worker nodes and functions as the primary data storage location. A schematic overview of the current WC-IC configuration is shown in Figure 7.7.

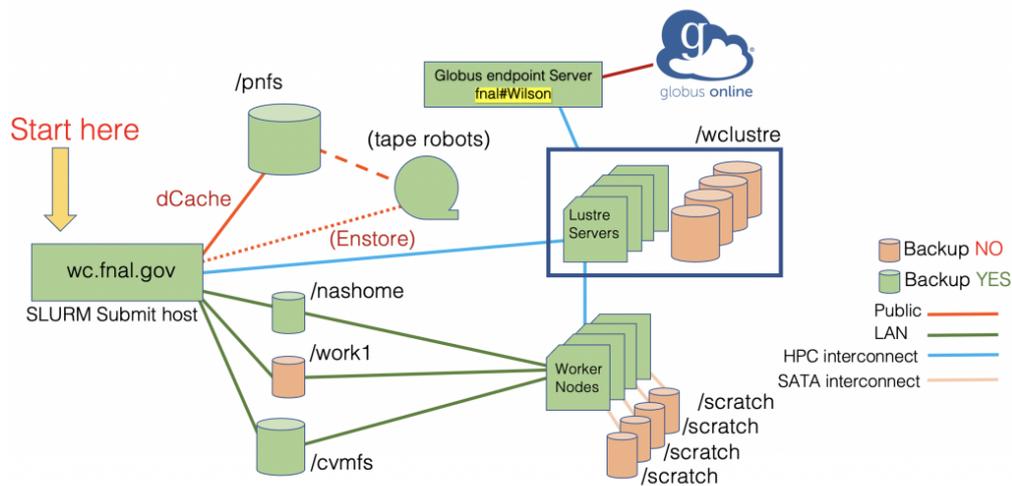


Figure 7.7: Schematic overview of the Wilson Cluster-Institutional Cluster. Courtesy of [179].

HPC software is supported and maintained on the WC-IC via the OpenHPC project and includes a large variety of different HPC software components. One typical component is container systems such as Docker.

These containers consist of all the required code, runtime, system tools, system libraries and settings which are needed to run any specified application. As they do not emulate a full virtual machine, they are smaller and more portable, while still providing a consistent environment to run software in, even on different machines.

On the WC-IC a lightweight version of the standard container technology is supported, Singularity. Singularity is optimised for usage on environments such as HPC clusters and allows for a different OS environment within the container compared to the host system. For the NOvA machine learning development work a singularity container is employed. It operates Ubuntu version 16.04, which differs from the host OS of SL7.9. The singularity container primarily contains Python packages and supporting software in specific versions and setups. The setups ensure that the packages have the required hardware acceleration enabled. Here the NVIDIA's CUDA toolkit release 9.0 is employed. The versions of the contained software are constrained to retain compatibility with integration into the wider simulation, data processing and further analyses. This includes limits originating from environments of cluster(or grids) used to perform the bulk of the experiments data processing.

Here the training and evaluation of the neural networks for the development purposes is performed in Python3, with Keras 2.2.4 employing a TensorFlow backend with version 1.12.0. Supporting functions are utilized from h5py version 2.8.0 and SciKitLearn version 0.21.3. The h5py package offers an interface capable of interacting with the HDF5 files in an efficient manner. SciKitLearn comprises a toolbox for the purpose of predictive data analysis, as such some of functions are used in the preprocessing of the data samples prior to training and evaluation.

7.4.1 Keras and TensorFlow

Keras [180] is the primary machine learning python package employed here, it is a high level Application Programming Interface (API) which enables machine learning on three different backends, TensorFlow [181], Microsoft Cognitive

Toolkit (CNTK) and Theano [182]. Due to changes and developments in the field of machine learning not all of these backends are supported today and the popularity and usages varies. At Keras version 2.3 and TensorFlow 2.0 the Keras API was moved inside TensorFlow itself, and remains accessible. Due to the previously mentioned version constraints some of the recent features and bug fixes of Keras and TensorFlow are not used in this work.

The Keras functional API forms a bridge between the human end user and the supported backend. It does this through the provisioning of high level but simple functions for the complex tasks that need to be run sequentially for machine learning. These functions are not strictly dependent on the specific backend employed, although the implementation varies for each backend. Simplification of the processes required for machine learning is a trade-off between ease of usage and understandability versus customizability and deeper access to the inner workings of both Keras and the backend. It is relatively easy to use for typical trainings and evaluations, but can require complex non-standard solutions for non-typical or custom applications.

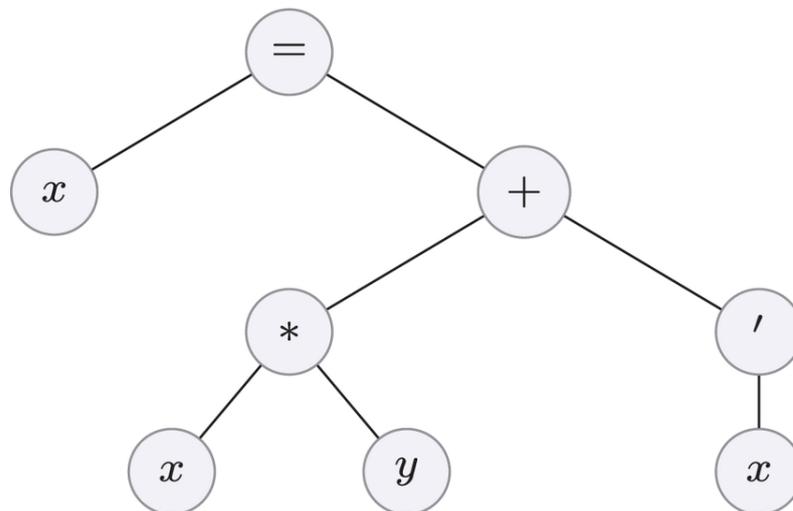


Figure 7.8: Simple expression tree representing $x = (x \cdot y) + x'$, the computational graph employed in TensorFlow shares a similar structure to the expression tree. In the TensorFlow computational graph the variables are tensors and the nodes represent the base operations which are enacted upon these tensors. Courtesy of [183].

Keras with the TensorFlow backend re-uses the base functions contained

in the TensorFlow package. TensorFlow handles data in the form of high dimensional arrays called tensors. Operations such as those comprising a forwards or backwards pass through a deep neural network can be reduced to sequential single step tensor based operations. A pass through a network layer is typically comprised of some function (e.g. addition/multiplication) of the inputs with the layer parameters. In TensorFlow both the layer parameters and every input and output are represented as tensors. A static computational graph maps all the relevant computation operations in their most simplified form into a flow chart, as can be seen in Figure 7.8. It connects the input tensors to the output tensors through the explicit definition of all intermediate operations. Although the individual values of all the tensors involved in the computation graph change during training and evaluation, their shape and connections remain the same.

7.4.2 The AdCVN framework

Keras with the TensorFlow backend alone is not sufficient to perform neural network trainings and evaluations. It provides the tools and functions for these tasks but scenario specific configuration and assembly is required before these tasks can be successfully executed. E.g. configuration of the model architecture, hyperparameter choices, data pipelines etc.

The AdCVN framework is developed to enable the execution of these task. It bundles various components which provide simplicity and consistency across the variety of trainings required for a study of domain robustness. Two components form the bulk of the framework and are contained in a primary and secondary python script. The first of these provides support for extensive argument parsing, either directly from the command line or through the use of separate configuration files. The usage of configuration files provides an additional record of runtime conditions beyond the standard output logs and records the values of the hyperparameters. It also enables a direct and simple method for restarting the training or evaluation should it be required. Beyond argument parsing the main executable functions are defined in this primary

python script, such as the main training function and evaluation methods. The eventual post evaluation processing and plotting script are separated as these are produced to be small efficient fast to run and simple to modify.

The main functions of the primary script import methods defined in the secondary script. These methods are extensive and establish most facets of the training and evaluation. Some of the components are reused in various parts of the defined methods which prevents a correctly ordered description.

7.4.2.1 Preprocessing

The data preprocessing superficially analyses the supplied dataset and applies a fast low resource trimming and indexing. It builds up a dataset containing all available events with an internal database of metadata. This metadata typically contains the parent file, parent domain, event identifier and interaction type. The interaction type is converted into a truth label: ν_μ , ν_e , NC or Cosmics. If desirable this metadata can be extended to include run, subrun, cycle, event and slice numbers. Events are then removed based on the defined cuts: p_t/p , ν_τ and cosmic downsampling. Additionally methods for the retrieval of more resource intensive event information are defined here such as the pixelmap, the event weights and their flux corrections. The total data sample is then randomly split into several smaller data samples. One for the purpose of training which the network can use to update its parameters, and the other for validating the model's generalizability. Given the sizes of the data samples used, the split is set at a threshold of 80% training to 20% validation. Evaluation of the network after training is complete, is done separately from the training/validation sample.

7.4.2.2 Generators

Beyond the meta-data not all the required information for training can be loaded into memory at once. Especially the two 100-by-80 pixel images for each event prevent the complete pre-loading of the data samples into memory. For medium to large data samples this is a common issue for neural network trainings. The solution is to dynamically load in small subsets of data. The

data-generator object is created for this purpose. It is capable of iteratively producing a batch of samples from the available meta-data and the pixelmap (and eventweight) retrieval functions. Additionally reshuffling methods included provide the capability to randomize the ordering of the current data sample. The training and validation data sample each have a corresponding and separate data generator. The reshuffling is therefore applied independently between the multiple data generators. Reshuffling is applied repeatedly during training but can be disabled if accidental repetition of events is undesirable during evaluation.

The batch based domain balancing resampling method described in section 7.2.2 is implemented in the data generators. A batchsize of 128 events is used.

7.4.2.3 The Model object

The primary training function acts as a higher level method which call upon various components prior to commencing the network training. Upon initialization a check is performed to determine whether a static network graph can be loaded in from a specified model save file. The weights and parameters are subsequently loaded in if such a model file is supplied. This functionality enables the continuation of training and the loading in of a trained network for subsequent evaluations. When no previous model is supplied a new model needs to be defined before it can be used.

A multitude of network architectures are available in AdCVN. The latest iteration is the ModifiedMobileNet. The legacy architectures include CVN Short Simple, MobileNet (v1, v2, v3) and ResNet (18, 50). The construction or build function begins with the definition of an input layer which has a single tensor shape. This single tensor is subsequently split into the two pixelmaps for each view. The subsequent layer structure is drawn from the predefined architectures up until the output layers. With the supplied classes the output classification layers are added. The full network architectures are supplied as auxiliary materials and can be accessed at a publicly accessible Github

repository².

Definition of the full network layer structure does not immediately generate the complete static network graph. The Keras Model function uses the defined layers to connect them on a layer-by-layer basis, expressing them as a function of the input layer and building up the network graph. This enables the full throughput from the input to the output of the network.

With the model definition ready various other components of the training can be initialized and defined. Both of the data-generators for the training and validation sample are initialized. The supplied choice of optimizer, e.g. Stochastic Gradient Descent or ADAM, is used to initialize the optimizer for training.

The model object is compiled. This action achieves multiple purposes. It defines what mathematical function is to be used as the loss(cost) function. The typical choices of loss functions are available from Keras or SciKitLearn. Custom loss functions can also be used if they satisfy the Keras requirements for seamless insertion. The metric functions which are monitored during training beyond the loss function are specified in the compilation as well. The default metric function is the accuracy, the percentage of correctly classified events.

Callback functions are initialized. These function run concurrently alongside the training and can be used for the purpose of monitoring or scheduling limited changes to the network or environment during training. Here several callback functions are employed during training. The ModelCheckpoint function allow continual recording of the network parameters at set points during training. This can be helpful if the training becomes unstable or significant overfitting occurs. The checkpointing enables the possibility to reset the network to the optimal point during training. Optimal is defined based on metrics such as the validation accuracy. The EarlyStopping callback stops training if the monitored metrics have not sufficiently improved during a pre-defined time period. The TerminateOnNan callback immediately stops training if any of

²https://github.com/kmulderdas/thesis_aux_files

the model parameters or outputs produce NAN's. The `LearningRateScheduler` allows for the learning rate hyperparameter to be reduced at a set time interval, it is not enabled by default here. The `ReduceLROnPlateau` combines the functionality of the `LearningRateScheduler` and the `EarlyStopping` callback functions, if the network performance plateaus the learning rate is reduced. This callback is likewise not enabled by default here.

Finally the fit function, which is the Keras training function, is called with arguments propagated from the configuration file. The training of the network then commences.

7.4.3 Custom components

To enable the various different modes of training several custom additions to the standard training procedure have to be introduced.

7.4.3.1 Eventweighting

The first addition is the usage of data samples with weights associated to them. The event weights specify the relative weight, or importance, that each event in the dataset has. During training this should be reflected in the amount of relative learning a network is capable of performing given the weighted samples. While this can be achieved in various ways, such as the repetition of events in an epoch according to the magnitude of their weight, a different approach is selected here. Given that the weight values are not necessarily integers a method which is capable of dealing with all real numbers as weight is preferable. Although not applicable here, as can be seen in Figure 7.3, negative weight numbers could be used in this method as well. The approach incorporates the event weights into the loss function itself. Direct multiplication of the event weights with the selected loss function propagates the eventweight into the magnitude of the gradients which are subsequently used for the network parameter update. This method can be used independently of the explicit form of the loss function.

The event weights are introduced as a separate input into the network

such that the current sample's eventweight as provided by the datagenerator is available to the loss function directly. Due to the version limitations of TensorFlow and Keras a direct path from the datagenerator to the loss function is not possible. The additional input layer on its own does not suffice as it is disconnected from the main graph. Connecting it into the main graph would lead to it being treated as a learnable input and subsequently would be used by the network to learn and optimize. To resolve this issue a lambda layer is employed. These layers wrap an arbitrary expression, such as a function, into a usable Keras layer. Here a function is used which contains no trainable parameters and only allows one of the inputs to the lambda layer to pass unchanged. Insertion of this layer ties the eventweight input layer into the larger graph network whilst maintaining its non-trainability and ensuring that the event weights do not influence training directly. The output of the eventweight layer can then be tied to the loss function as a separate input. This yields the desired setup for training with an eventweighted loss function as can be seen in Figure 7.9.

7.4.3.2 Gradient reversal layer implementation

For the implementation of the gradient reversal layer the keras ($< TF 2.0$) documentation for the creation of custom layers is followed. The gradient reversal itself is achieved through multiplication of the passing gradient during backpropagation with a negative constant. The value of this constant can be varied to achieve optimal results. TensorFlow itself contains a function capable of performing the elementwise negation called `negative()`. As the layer is called upon twice for each training step, once for the forwards pass and once for the backwards pass, it keeps an internal record of the number of calls it has received. Based on that information it either flips the gradients or doesn't. This ensures that the layer has no impact during the forwards pass of the training procedure. Likewise during inference the gradient flipping is disabled as the network does not need to perform any parameter updates. The gradient layer itself has no trainable parameters, the constant λ is a hyperparameter of the training.

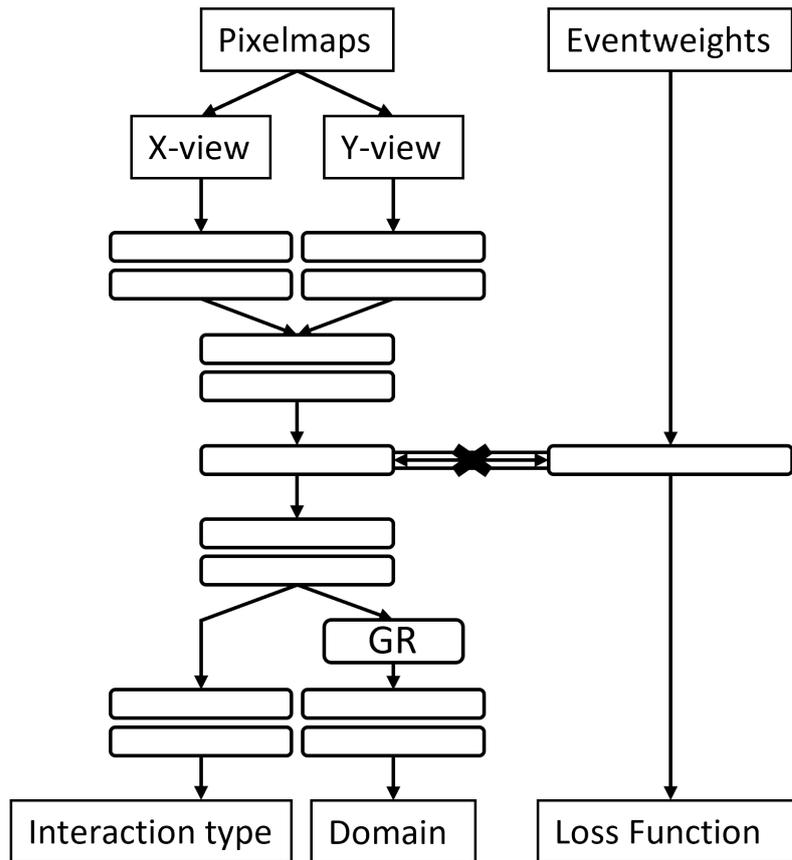


Figure 7.9: Schematic overview of a network architecture with eventweighting. The black cross denotes the inability of information to cross in the connection layer, while still allowing the event weights input layer to be included into the keras network graph.

At the point between the convolutional layers and the fully connected classifier, the gradient reversal layer is connected and feeds into its own fully connected classifier. This branch of the network classifies the domain instead of the class. The gradient reversal layer during the backwards pass updates only the previous feature-extracting layers.

7.4.3.3 Loss penalization implementation

The loss penalization method is implemented according to the description given in section 6.6.3, although minor alterations have been made. The implementation is fully contained in a custom loss function. This loss function serves as a wrapper for a traditional loss function such as the categorical cross entropy. It gathers the network outputs for the current batch, and clips their values to

the required epsilon as is the norm in Keras. Epsilon is typically set to 10^{-7} . Histograms for each domain and class are then produced. Gaussian blurring is then applied on a binwise basis for each histogram to produce the count estimates. These are then individually area normalized. Variations introduced from imbalances in the sample counts of the classes and/or domains are not of interest for the evaluation of the network’s robustness against a domain shift, while the shape variations between the network output on different domains are. Ideally the difference in behavior of the network on the same events in different domains would be evaluated, but these are not always available. A batch based average produces an approximation albeit with statistical variation.

The shape variation between the count estimates is evaluated using the following expression,

$$\Lambda(x, \Delta) = \frac{1}{K} \sum_{k=1}^K \left(\frac{(\mathcal{N}_k(f(x)) - \mathcal{N}_k(f(x + \Delta)))^2}{\mathcal{N}_k(f(x)) + \mathcal{N}_k(f(x + \Delta)) + \epsilon} \right) \quad (7.1)$$

This form was inspired by the χ^2 formula and was found to produce less extreme gradients for bins in the region where both $\mathcal{N}_k(f(x))$ and $\mathcal{N}_k(f(x + \Delta))$ have small values if a small epsilon term was added.

This loss penalization factor is then further normalized by the number of domains and multiplied by a hyperparameter which moderates the relative strength of the loss penalization term compared to the regular loss value. Simple addition then yields the total loss value for this custom loss function. Implementations with eventweighting are produced by weighting the histograms prior to the Gaussian blurring as well as multiplying the total loss value with the event weights.

7.4.3.4 Loss penalization with memory

The histogram-based calculations used in the loss penalization term function are filled on a per batch basis. A sufficiently large batchsize would ensure the constituent histograms are appropriately filled. If a larger number of classes and domains are used, the batchsize would need to be increased as well.

With large input images such as the pixelmaps the batchsizes become memory constrained. Statistical fluctuations are expected to degrade the accuracy of the loss penalization factor, as is the case for small batchsizes.

To enable operations with lower batchsizes, on the order of hundreds not thousands of samples per batch, a memory construct is introduced. This constitutes a number of function and layers which cooperate to enable the re-use of previous batch information in the current loss penalization calculation. While this information is one or several batches out of date, the network updates are typically small on a single batch. Therefore the previous network output can be used to artificially boost the statistics of the current batch loss penalization term. With the included normalizations the additional values from the memory act as a dampening to the statistical fluctuations in the histograms and count estimations. As the additional memory based values have lost their connections to the original network parameters the gradients of the current batch are only influenced by the additional shape normalization. This ensures the prevention of repeated learning on the same data samples.

Due to the version constraints the implementation of the memory is less efficient than if the latest versions were used. Additional operations are required to enable the recording and readout of the memory construction. These increase the training time, although the inflation is manageable and did not prevent usage of this implementation.

Similarly to the eventweight layer structure an additional branch consisting of several layers is created for the initial layers of the network. A connection layer with a single node is placed after the input pixelmap layer. The input pixelmap layer now has two separate outputs, the normal splitting layer and the connection layer. The connection layer is not trainable and has all its weights and parameters initialized to zero. This ensures that it serves purely as a connection point for the upcoming memory structure into the larger graph. It cannot pass any information from the input layer to the subsequent memory structure in either the forwards or backwards pass.

The memory structure is constructed to act as a storage buffer for one or more batches of data. For each batch a number of parallel layers connected to the connection layer are constructed. Each of these parallel layers contains an untrainable dense layer with weights initialized to zero and biases initialized with Glorot Uniform. This dense layer is used as temporary data storage. With weights equal to zero, inputs equal to zero and no activation function, the output of this layer will be the information contained in its bias. This form of information encoding forms the core of the memory structure. Each dense layer is employed to store a part of the batch information required for the loss penalization factor calculation. These are the batch domain truth, network output scores and if applicable the event weights. Each dense layer is followed by an untrainable reshape layer which rearranges the output tensor of the previous dense layer to the shapes required by the loss penalization calculation. The outputs of the reshape layers are combined with those of the eventweight input layer into an extended lambda layer. This layer again serves as a connection point which only allows a single input to pass. The memory structure is thus incorporated into the network graph but is isolated from directly influencing it or being influenced by it during training.

With the storage buffer constructed and initialized, the values of the buffer need to be continually overwritten and read out during training. A set of metric and callback functions is used for this purpose. These functions are created just prior to the model compilation alongside all the other metric and callback functions. The callback function initializes a set of placeholder TensorFlow variables, one for each tensor of data to be saved to the memory structure. It also retrieves the initial values of all the weights and biases of the dense layers contained in the memory structure. All of these are saved to the internal memory of the callback function. On the end of every batch during training a separate subfunction of the callback re-evaluates the current value of the saved placeholder TensorFlow variables and propagates the output of that evaluation to the values of the biases of the dense layers in the memory

structure. This effectively saves the placeholder output to the memory buffer. This is possible as tensors in TensorFlow retain their shapes but their values can change. The evaluation forces the calculation which produces the current values of the variables or tensors.

The metric functions fill the gap in the callback function's workings. They are necessary to connect the placeholder variables to the relevant information which needs to be stored. For each tensor to be stored a separate metric function is constructed. Each of these functions assigns either the batch's domain input, network output or the event weights to the associated TensorFlow variable in the callback function. As all of these are tensors, their values change for each forwards pass and are correctly propagated up to the evaluation points. The storage and updating of the memory buffer is now handled automatically during training, as both the tensors, metric and callback functions operate continually during the training itself.

The readout of the buffer and incorporation of that output into the loss function is relatively straightforward. Similarly to the eventweighting, the output of the reshape layers of the memory structure can be directly incorporated into the loss function. The additional information is concatenated into the current batch information and the final normalization updated to reflect the relatively smaller contribution of the current batch information. A schematic overview demonstrating training with the inclusion of the memory construct is shown in Figure 7.10.

7.4.4 Network evaluation

For the evaluation of the trained networks, in- or excluding event weights or adversarial methods, various different functions and scripts are included in the AdCVN framework.

7.4.4.1 Training curves

The first of these visualises the information stored in the logfiles. These files are produced and saved automatically during training on the WC-IC GPUs.

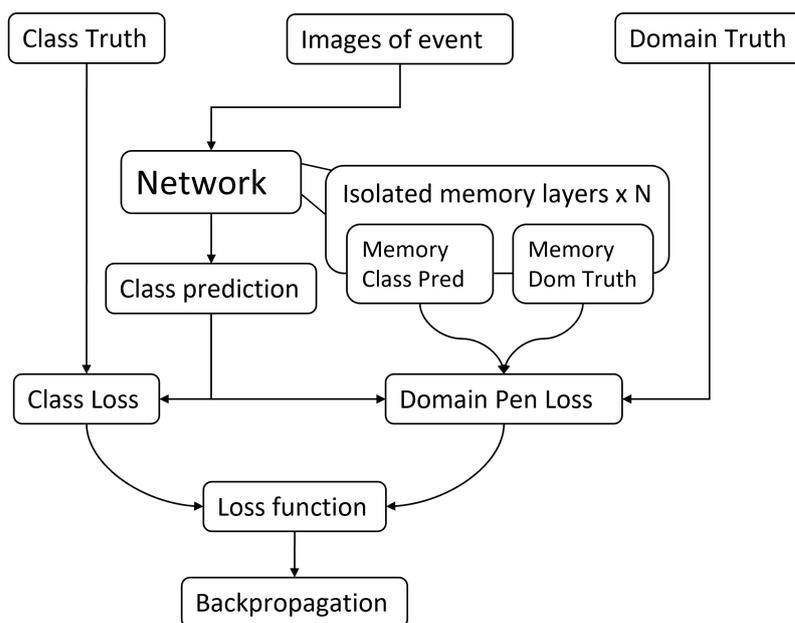


Figure 7.10: Schematic overview of a network architecture with loss penalization and memory layers. The additional memory structure allows for the feedforward of previous batches classification predictions and domain truth values into the domain loss penalization factor calculation.

They contain the standard output that would be shown in the (python)terminal if the training was done interactively. As the WC-IC is located onsite at FNAL a potentially unstable internet connection could result in a disruption to the interactive training. While there are methods available to prevent this from occurring, the simplest is to employ the Simple Linux Utility Resource Manager (SLURM) [184] provided on the WC-IC to run the training as a remote job. The jobs are typically constrained to a maximum runtime of eight hours, although specific permissions can be granted to enable the submission of a 24 hour job. The SLURM job will save all the standard output to the log files. For each training the log files contain a valuable record of the exact running conditions, hardware and configuration arguments which are used to initiate the training. As an additional safeguard against bugs or human error the log data can be checked against the configuration file used in the training command, this ensures the performed training was completed with the correct arguments and hyperparameters.

After the run conditions are shown, the dataset is build up. When this

completes, information about the dataset, such as the total number of events loaded and the amount of events in the training and validation datasets, are shown as well. Now the training commences and the standard output of the Keras fit/training function is displayed. This includes the current epoch, the current batch out of the total number of batches in each epoch, the estimated time epoch completion and finally the loss(es) and specified metric values for the batches in the epoch up until the current batch. At the end of each epoch the total values for these metrics during that epoch are shown and the network is temporarily frozen, meaning that it will not update its parameters for any new samples that are pushed through the network. This is a requirement for employment of the network to perform inference. In this configuration the network is evaluated on data samples which are not contained in the training dataset, the primary purpose being evaluation of the network's ability to generalise the learning it has achieved on the training dataset. When completed the metric values on a subset of the validation sample are shown. These combined values for each step and epoch provide a thorough record of the training.

Multiple visualization scripts are included which extract information, such as the loss or accuracy, from the log files for the required intervals. Typically this is done on an epoch-by-epoch basis as individual steps can be prone to large fluctuations. The evolution of the loss and accuracy over the training time as measured in epochs can provide a method of visualising and characterizing the network's learning and performance increase during training. An example of such a training graph can be seen in Figure 7.11.

As time passes the network parameters are updated to produce output predictions which match the truth labels more closely. This learning process is reflected in both the loss and accuracy training curves. The average error as defined through the loss function decreases while the accuracy increases as the training continues. During the initial epochs of training the network quickly converges from the initialized state towards one approximating the optimized

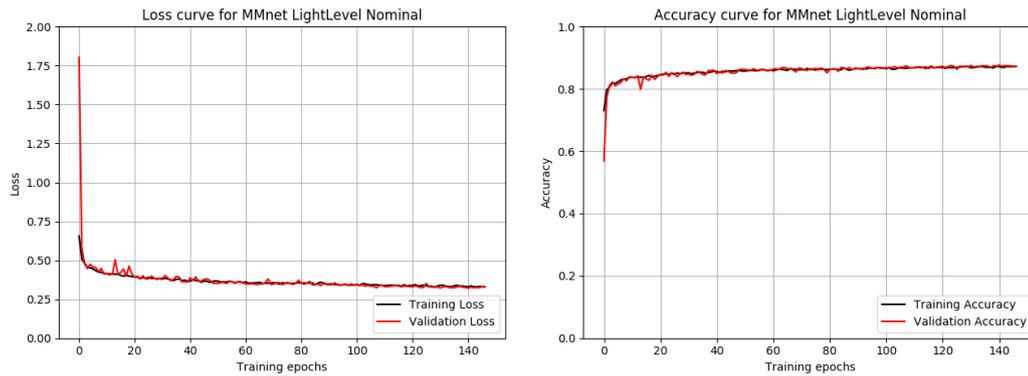


Figure 7.11: Examples of curves produced from the training of a neural network. In this case a modified mobilenet architecture was trained on the nominal domain of the light level systematic sample.

network state. During subsequent epochs the weights and biases are optimized further to approach the final state as much as possible. Due to the large epochs the initial learning can appear abrupt, if smaller epoch sizes were selected a smoother loss and accuracy curve would be produced.

7.4.4.2 Evaluation files

While the evolution of the loss and accuracy over the course of the training provides valuable information it lacks in depth information about the network's performance and its robustness against domain shifts. To remedy this a separate evaluation function is included alongside the training function. This function borrows many elements from the general training function such as the loading in of a trained network, the setup of a datagenerator and compilation of the optimizer. The previously trained model is run, in inference mode, over all the contained data samples in the test dataset exactly once. The network output, metrics, predictions and truth information for each data sample is collected and saved to a HDF5 evaluation data file. Separation of the inference and later post processing and visualization allows for fast iterations of the latter as the former is a relatively time consuming resource intensive task.

With an evaluation HDF5 file post-processing can be applied to quantify different aspects of the network performance. There are a large variety of methods and techniques available for this purpose. The selection is typically

based on a combination of factors such as the power of the technique, the frequency of usage in the wider community and interpretability of the results. Here the intersection of both the AI/ML community and that of high energy physics combined with the quantification of robustness against a domain shift form the primary motivation for the evaluation techniques applied.

The primary information input for post processing is the network output on each data sample. For each sample the network has pushed the input pixelmap images through all the layers to produce a numerical value for each class in the output layer. Due to the applied softmax activation these numerical values represent the probabilistic belief of the network that the current data sample belongs to this class. The class with highest numerical value is predicted to be correct class by default. For some application all values above a certain threshold of the numerical value of that class are predicted to be the correct class.

A comparison between the predicted class and the truth information regarding the class results in the earlier mentioned accuracy. The numerical values prior to conversion to a prediction is the network output. Additional network branches which contain an output layer, such as the domain branch in a network with a gradient reversal layer, produce numerical output values representing their own classes. For the domain branch this would be the original domain of the data sample. The secondary information for post processing is the truth information for each event, such as the true interaction type, true domain and eventweight. This information is used to subdivide and compare subsections of the total network output. The quantification of difference in network performance on these subsets and on the total forms the foundation for most evaluation methods.

7.4.4.3 Particle Identification Spectra

The prevalence of the numerical value output for each class can be plotted in a histogram to produce a spectrum. These spectra contain the probability scores for each interaction type forming a class. To simplify the name these values

or scores are referred to as PIDs or PID scores. The Particle IDentification (PID) in this case is a slight misnomer as the network is not used to identify all the individual particles participating in the interaction. Networks for this particular function are employed in the NOvA experiment. Nevertheless the terminology is applied more broadly.

With the secondary information the PID spectra can be further subdivided. The truth information can be used to split each class-based PID spectrum into two spectra for the signal and background distributions. The signal distribution contains events for which the truth information matches to the current class. Likewise for the background distribution the truth information does not match to the current class. This class-based signal and background division provides a method for reducing the multi-classification task into several binary classification tasks.

7.4.4.4 Statistical errors

The predictions of the network are produced from the values of the PID scores for each class. For the strategy which places a threshold, or cut, on the PID scores the discriminatory power of the network can be optimized through the variation of the cut location. This optimization can be performed with the truth information that is available in the evaluation data sample. For the real recorded data this truth information is not available. For the class-divided binary classification case placement of a cut on the signal and background PID spectra results in the formation of four distinct populations. The area of the signal spectrum above the cut contains a population of correctly predicted events, these are known as true positives (TP). The area of the background spectrum above the cut contains a population of incorrectly predicted events, these are the false positives (FP) or type I errors. For the two areas below the cut the same follows, the background area below the cut contains a population of correctly predicted events known as true negative (TN). The signal area under the cut contains a population of incorrectly predicted events known as false negatives (FN) or type II errors. In Figure 7.12 the formation of the various

population can be seen for a signal and background separated distribution.

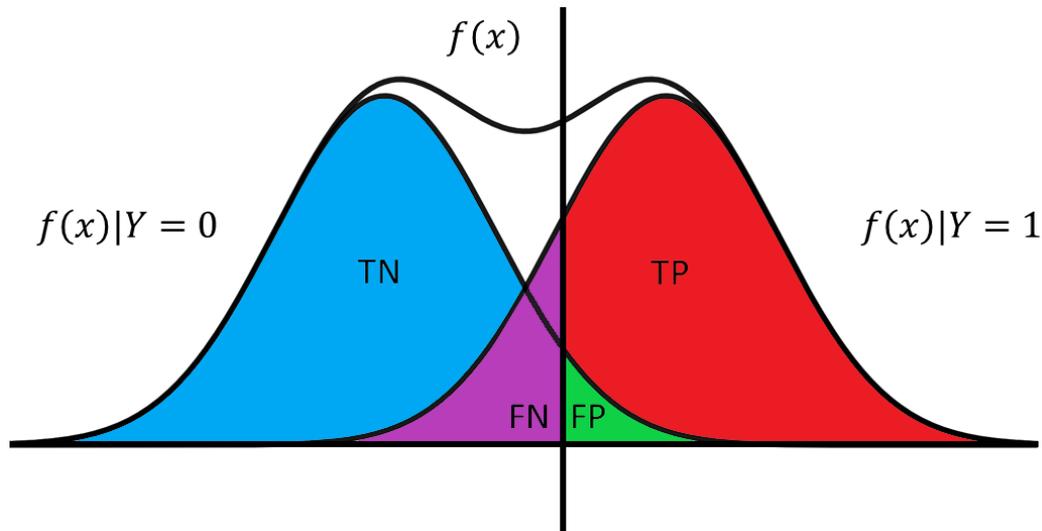


Figure 7.12: Schematic overview of a network output score $f(x)$ or PID spectrum. It is subdivided into a signal and background distribution with truth information, $Y = 1$ forms the signal and $Y = 0$ the background. With the placement of a threshold or cut, denoted by the vertical black line, the signal and background distributions are separated into the coloured TP, FP, TN and FN distributions. Note that the TN and TP distributions extend to the threshold, underneath the FN and FP here respectively.

With the definitions of the positives and negatives various statistics can be calculated and/or optimized to determine the optimal cut locations.

7.4.4.5 Confusion matrices

The confusion matrix is a visualization tool which plots the predicted and true class populations. For the binary case the matrix displays the true positives/negatives on the diagonal and the false positives/negatives on the off-diagonal elements of the matrix. For the multiclass case the TP, TN, FP and FN values are not shown directly but can be calculated from the displayed populations. With the predicted classes on one axis and the true classes on the other, the population of the matching cell (predicted is equal to true) is the true positive. The summations of the connected columns or rows yield the false positives or negatives respectively, these sums exclude the value of the connected TP diagonal.³ The true negatives can be determined from the

³This can change depending on the orientation of the two axis.

summation of the remaining populations. The matrix form of these sums is displayed in Figure 7.13. Using this method the TP, TN, FP and FN can be determined for each class, achieving the equivalent of the binary subdivision method. For that method the summations are performed at the PID spectra stage.

		Prediction class		
		$(0, \dots, n - 1)$	(n)	$(n + 1, \dots, N)$
Truth	$(n + 1, \dots, N)$	True Negative	False positive	True Negative
	(n)	False Negative	True positive	False Negative
	$(0, \dots, n - 1)$	True Negative	False positive	True Negative

Figure 7.13: Schematic overview of a confusion matrix, where for class n the matrix entries are labelled in the groups of the true positives, true negatives, false positives and false negatives.

The confusion matrix is typically normalized through area normalization of the columns or rows. Normalization by the summation of the total number of true events in each class is referred to as efficiency normalization. Oppositely the purity normalization is produced from the summation of the total number of predicted events for each class.

While the confusion matrices can be helpful in quantifying the performance

of the classifier, due to the numerical nature of the displayed information, the sheer amount of information can quickly become overwhelming for comparisons between multiple networks and multiple domains. These comparisons would require $2 \times (\text{number of domains}) \times (\text{number of networks})$ confusion matrices.

With the produced values of the TP, TN, FP and FN further derived statistics can be calculated. The accuracy metric, which is employed during training to determine the number of correctly classified samples can be expressed as follows,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7.2)$$

Other relevant statistics can be calculated in terms of the produced values as well. The true positive rate, also referred to as the efficiency or recall is the ratio of true positives to the total signal population,

$$TPR/\text{Efficiency}/\text{Recall} = \frac{TP}{TP + FN} \quad (7.3)$$

Effectively showing the fraction of events belonging to the class that are being correctly classified.

Likewise the false positive rate is the ratio of the false positives to the total background population,

$$FPR = \frac{FP}{FP + TN} \quad (7.4)$$

Effectively showing the fraction of events not belonging to the class being incorrectly classified.

The purity or precision is the ratio of the true positives to the total population of positives.

$$\text{Purity}/\text{Precision} = \frac{TP}{TP + FP} \quad (7.5)$$

This shows the fraction of events being predicted to belong to the class which

are correctly predicted.

7.4.4.6 Precision & Recall and Receiver Operator Curves

With the usage of the Precision, Recall/TPR and the FPR simple performance visualization for the classifier can be produced. These are the Precision and Recall (PR) and the Receiver Operator Characteristic (ROC) curves. As these three statistics depend on the placement of the threshold or cut, a variation of this placement can simultaneously characterize the network's performance and the optimal cut location. For each of the values in a range of cut locations a triplet of the statistics (Precision, Recall/TPR and FPR) are collected. These can then be plotted against each other. The resulting plots are the PR and ROC curves. The PR curve expresses the relation between the Precision and the Recall for the varying cut placements, whereas the ROC curve expresses the relation between the true positive rate and the false positive rate.

For the PR curve high values of both the precision and recall are desirable, and therefore the curve should approach the upper right corner of the space. The curves of higher performing networks will follow this trend. For the ROC curve a high value of the TPR for low values of the FPR is desirable. Performance indication therefore stems from the ROC curve trending towards the upper left corner of the space.

The performance of the classifier can be quantified by the Area Under the traced out Curve (AUC), as higher values (or lower for the FPR) of the statistics are required for a large area to be produced. Conventionally the AUC is calculated from the ROC curve.

These curves are typically calculated for binary classification problems. As multi-class classification can be reduced to a set of binary classifications various strategies exist for the recombination of the individual class statistics into a single summary statistic. These summary statistics can be the micro average, macro average or weighted average of other relevant statistics calculated on the individual classes. The micro average calculates a statistic over the entire evaluation data with no distinction between different classes. The macro average

calculates a statistic for each class individually and then takes the unweighted average over all these class specific values. The weighted averages is equivalent to the macro average with the substitution of the weighted average over the class specific values.

Other summary statistics can be determined from these curves such as the figure of merit (FOM), the $F1$, F_β , etc. A typical FOM which can determine the optimal location of the cut would be a simple multiplication of the efficiency and purity, although others can be defined and used as well. The $F1$ or more generally F_β are metrics which combine both the precision and recall into a single statistic. These can be useful to quantify the networks performance if the individual variations of the underlying statistics are less relevant. The $F1$ is the harmonic mean of the precision and recall whereas the F_β is the weighted average, where β denotes the weighting. The $F1$ score is defined as,

$$F_1 = 2 \frac{Pr \cdot R}{Pr + R} \quad (7.6)$$

where the precision is denoted as Pr and the recall as R . The F_β score is defined as,

$$F_\beta = (1 + \beta^2) \frac{Pr \cdot R}{(\beta^2 \cdot Pr) + R} \quad (7.7)$$

where β is the applied weight denoting the relative importance of the Precision to the Recall.

While the FOM, $F1$ and F_β are used for the evaluation of network performance during training other statistical measures are selected for the domain robustness evaluations.

7.4.4.7 Statistical measures

The statistics such as the purity and efficiency can be used beyond the purpose of quantifying a single network's performance. In the comparisons between networks trained with different architectures or hyperparameters these tools can be used to quantify the relative improvements or deteriorations. Likewise the comparative study of a network's robustness against domain shifts relies on a

similar methodology. Given that a network which is considered to be not robust would have a large variation in its output and performance when evaluated on different domains it follows that a quantification of this variation forms a suitable measure of the robustness. Similarly an optimally robust network would display an equivalent performance on each domain. If the magnitude of the variation is taken as the primary robustness metric, a reduction in the metric is then defined as an improvement to the robustness.

While the derived statistics can be utilized for the quantification of the robustness, the PID spectra from which they have been derived can be employed directly as well. Similarly to the histogram based domain comparison contained in the loss penalization calculation, the variation between PID spectra on different domains can be measured and used as a robustness measure. The PID spectra are produced in the histogram format which enables bin-wise calculations of the distance between two spectra. Various method can be utilized for this quantification.

The Kullback-Leibler (KL) divergence [185], also referred to as the relative entropy is one such measure. Its typical usage is to provide a measure of the dissimilarity between two probability distributions. The network output scores with the softmax activation function as captured in the PID spectra can be treated as discrete probability distributions, and thus the KL divergence could be used to evaluate the variation between the PID distributions. It is defined as,

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \quad (7.8)$$

where $P(x)$ and $Q(x)$ are the discrete probability distributions being compared and the summation runs over all the steps contained in those distributions. This definition of the KL divergence is not symmetric as $D_{KL}(P||Q) \neq D_{KL}(Q||P)$. This potentially frustrates further comparisons between a multitude of different PID distributions. A modified version of the KL divergence which is symmetric

can easily be constructed,

$$D_{MKL}(P, Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right) + Q(x) \log \left(\frac{Q(x)}{P(x)} \right) \quad (7.9)$$

Another measure is the Kuiper statistic [186], which is an extension of the Kolmogorov-Smirnov(KS) statistic. Both of these require a set of cumulative probability distributions as input. A binwise iterative summation where the current bin contains the full sum of all previous bins produces the cumulative probability distributions from PID spectra. The KS statistic takes the absolute difference between the two cumulative probability distributions originating from the PID spectra, and determines the maximum value between them,

$$D_{KS}(F_1, F_2) = \max |F_1 - F_2| \quad (7.10)$$

Once again a lower value of the statistic corresponds with a higher degree of similarity between the original PID spectra.

For the Kuiper statistic the positive and negative differences are considered separately and an additional summation is performed to calculate the value of the statistic,

$$D_{\text{Kuiper}}(F_1, F_2) = D^+ + D^- = \max(F_1 - F_2) + \max(F_2 - F_1) \quad (7.11)$$

The advantage of the Kuiper statistic relates to the areas of sensitivity of the KS and Kuiper statistics. The KS and Kuiper statistics are similarly sensitive to shifts of the probability distributions, however the Kuiper statistic is more sensitive to spreads in probability distributions [187]. A shift would be considered a translation whereas a spread is an increase in the magnitude and length of the probabilities tails. While the PID spectra can be normalized and contained within a limited range (0, 1) it is not uncommon for significant variations to be contained in the edge bins of the distributions. Therefore the differences in sensitivity cannot be ignored and the Kuiper statistic is selected.

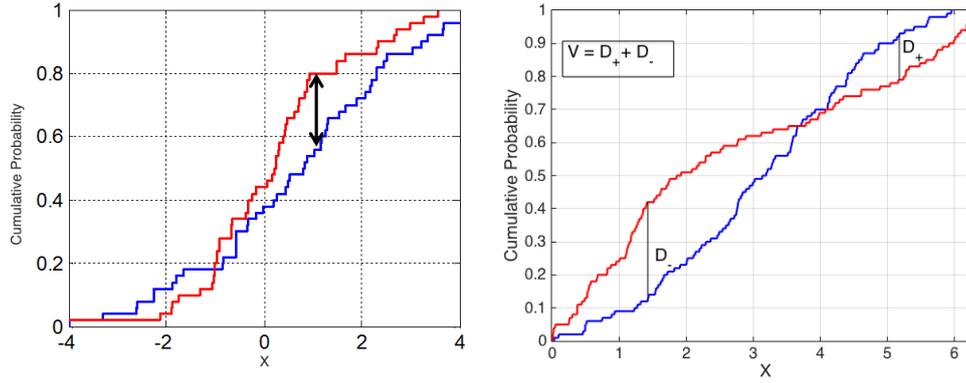


Figure 7.14: Two sample Kolmogorov-Smirnov(left) and Kuiper (right) statistics. The statistics are calculated based upon the largest distances of the cumulative probability distributions of the two samples.

Measures based on geometrical distance on a bin-wise basis can be used for the similarity quantification between the two PID histograms. These include measures such as the binwise distance, a modified χ^2 , cosine similarity, etc. The cosine similarity measures the cosine of the angle between two vectors. These vectors can be of any dimension so long as both are of the same dimension in shape. For the PID spectra each bin can be used as a separate dimension, with the full spectrum forming the vector. If the two PID spectra are more similar the angle between the vectors they trace out is smaller. The cosine similarity is calculated as,

$$S_C(A,B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (7.12)$$

The modified χ^2 measure originates from the chisquared hypothesis test. There the test statistic consists of a ratio between the squared difference of the observed and the expected values over the expected value. This statistic can be calculated for a discrete set of values such as a histogram. In the case of the PID spectra comparison a modified symmetrised version is employed,

$$\chi_{\text{Mod}}^2(A,B) = \sum_{i=1}^n \frac{(A_i - B_i)^2}{A_i + B_i} \quad (7.13)$$

With the definitions of the statistical measures which can be used to quantify the robustness of a network against a domain shift, the implementations of domain adaptation techniques in the AdCVN framework and the descriptions of multiple domain shifts present in NOvA datasets the robustness of EventCVN against these domain shifts can be determined. Additionally the benefits of applying the domain adaptation techniques can be examined.

Chapter 8

Validation of domain adaptation techniques on toy scenarios

To experimentally validate the effects of a domain shift on the neural network performance a sandbox environment is created. Additionally the effects of the previously discussed adversarial domain adaptation techniques can be demonstrated and validated in this sandbox experiment. The usage of a sandbox environment, or toy model, offers a fully controlled simplified experimental environment which is ideal for testing, development or validation purposes. Although it typically lacks the complexity of real world applications, the extensive control of all experimental variables allows the gained information to be used for further extrapolation of the achieved results. Additionally, the smaller scale correlates with a decreasing iterative training time which allows for faster testing and development. A discrete domain shift is selected for the toy scenario as it has a higher degree of similarity with the systematically shifted NOvA data samples.

A binary variation of this toy scenario is expanded on in section 8.1 with the two network domain adaptation technique. The binary classification scenario is then used to validate the effectiveness of the gradient reversal and loss penalization techniques in sections 8.2 and 8.3 respectively. A more complex multiclass variation of the toy scenario is employed in section 8.4 to validate the memory addition in a low batchsize environment.

8.1 Binary classification

With regards to domain shifts several examples of toy scenarios are employed for these purposes in the wider literature. One of these is a binary classification task. It has two variants, they differ by the specific encoding of the domain shift. For one it is continuous while for the other it is discrete. The binary classification task consists of two dimensional data points which have been drawn from one of two multivariate Gaussian distributions. The first class consists of samples drawn from a multivariate Gaussian centered at the origin with non-zero off-diagonal elements in its covariance matrix. The second consists of samples drawn from the symmetric multivariate Gaussian which has a diagonal covariance matrix. The domain is encoded in the y-coordinate of the mean location for this distribution. For the continuous case Z can take any value between -1 and 1 while for the discrete case Z is equal to either -1 , 0 or 1 . The nominal location of the mean is at $(1, 1)$. The two distributions can be summarized as follows,

$$x \sim \mathcal{N} \left((0, 0), \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \right), \quad x|(Z = z) \sim \mathcal{N} \left((1, 1 + z), \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \quad (8.1)$$

where the left distribution generates the domain unshifted signal class and the right generates the shifted background class. The standard notation for a Gaussian distribution is used, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. In Figure 8.1 a schematic overview of the cluster centers for each of these distributions is shown.

8.1.1 Data generation

The discrete binary classification toy model is implemented in python. Using the methods included in the numpy module, data samples are generated to populate the classes and domains. Each data sample is a single point with both an x and y coordinate. In total a data sample with size 125000 is generated. Half of these are generated from the signal class distribution, the rest from the background class. To encode the domains for the discrete case a set of

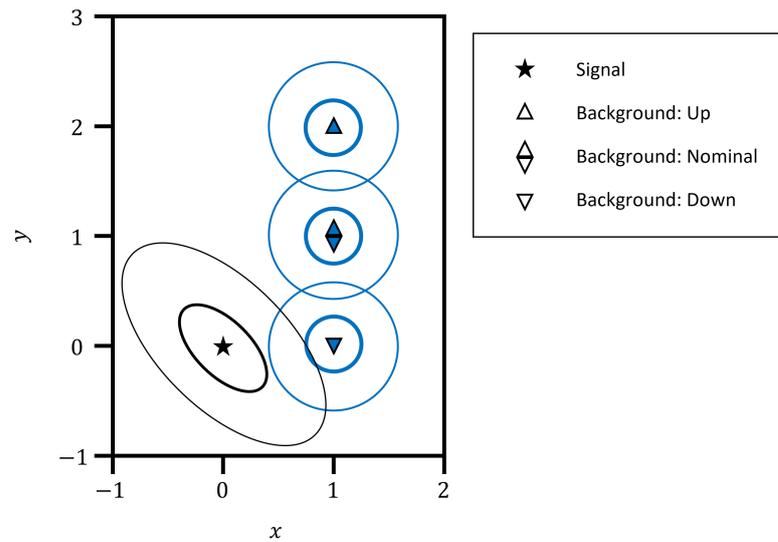


Figure 8.1: Schematic overview of the binary classification scenario. The means of the distributions from which data samples are shown for the signal and background (noted as 0 and 1 respectively). The contours indicate the shapes of the two dimensional distributions.

integers is generated from a uniform distribution. The data samples in the second class are then translated vertically by the integer amount corresponding to their domain. For the continuous case the vertical offset is drawn from a normal distribution with a mean of zero and a standard deviation of one. Once again the offset is applied only to the samples contained in the second class. Going forward the discrete case is assumed to be used. The total data sample is split 60%/40% into a training and validation dataset. Additionally the class and domain truth information are one hot encoded, which produces an array where each entry is a vector containing a single non-zero entry of value one corresponding to the truth class or domain from the numerical labelling. This recasting of the truth array shape enables an easier comparison to the network's output as their shapes match afterwards. Generally this is applied for multiclass classification with a softmax activation in the final layers.

8.1.2 Network architectures

With the data produced and preprocessed, four network architectures are defined prior to training. These are required for the purposes of examining

the effects on the network performance induced by the domain shift and the application of the adversarial domain adaptation technique. In theory only two networks are required, however known bugs in the functionality of the boolean Keras variables which govern the trainability of layers, subsection and the entire network prevent this efficient implementation. Post compilation variation of these variables should produce trainings in which the sections of a network which have been flagged as non-trainable are frozen and not updated. However simple unit testing of the relevant code demonstrates that these flags do not work as intended. Newer versions of Keras include resolutions to this issue. In this version the problem can be circumvented through the introduction of 2 more networks and the toggling of the trainable boolean variables prior to compilation.

The typical training procedure for the two network domain adaptation has been illustrated in section 6.6.1 and Figure 6.17. The multi network implementation is based on this procedure.

The first network is the discriminator. It is fully connected and consists of four layers.

- An input layer
- A 20 node tanh activated layer
- A 20 node relu activated layer
- A 2 node softmax activated classifying layer

The second network is the adversary. It too is fully connected and consists of four layers.

- An input layer consisting of the discriminator network
- A 20 node relu activated layer
- A 20 node relu activated layer
- A 3 node softmax activated domain classifying layer

It trains not on the data samples, but on the output of the discriminator network for each data sample.

The final two networks are constructed using the first two. The first of these consists a trainable discriminator followed by untrainable adversary. It has two output branches, the discriminator output and the adversary's output. Each of these branches has a separate loss function associated with it. Both losses are based on the categorical cross-entropy with a multiplication factor which determines the relative weight of the two losses. For the class/discriminator branch this weight is set to one, whereas the domain/adversary branch is given a weight of 5. The losses are summed to produce the loss value with which the network performs back propagation. Training this network will update the discriminator parameters with the influence of the adversary included.

The second network has a similar architecture but has a single output, namely the adversary's. The included version of the discriminator is set to be untrainable and the adversary is set to be trainable. A single loss function is defined as the categorical cross-entropy with strength one. This network configuration allows for training of the adversary whilst leaving the discriminator frozen.

The network configurations are represented schematically in Figure 8.2. The numbered configurations show the discriminator (pre)training (1), the adversary (pre)training (2), the combined training of the discriminator with the frozen adversary (3) and the training of adversary with the frozen discriminator (4).

8.1.3 Training and evaluation

The various networks are individually compiled directly after their definition with the correct boolean flags for trainability included. As the flags have no influence post compilation the various configurations can be set with different flags whilst maintaining the required connections to original objects (networks) they draw from. This linking is crucial for the adversarial training as the various copies of the discriminator and adversary networks need access to the same

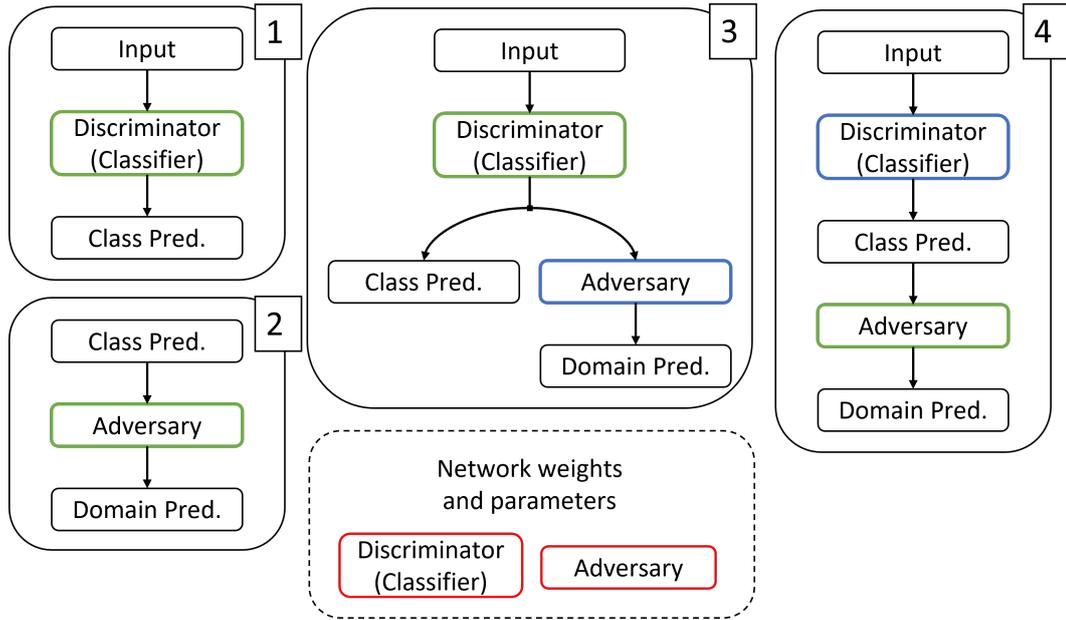


Figure 8.2: The four network configurations required for the multi network domain adaptation implementation. The All constituent networks are linked to the common weights and parameters which are shown in red. The blue networks are frozen and therefore unable to change the linked parameters. The green networks are not frozen. The network configurations are numbered as described in section 8.1.2.

weights and parameters. Training of the discriminator should carry forwards to all subsequent network configurations it is included in, even if it might not be trainable in those configurations.

The linking ensures that during evaluation of the current computational graph the original network weights and parameters are queried for their current values. Only if the flag for trainability of the current graph is set to true can the linked weights and parameters be updated based on the current evaluation.

All remaining hyperparameters are either set to their default values, or where possible chosen to be equivalent to typical values used in wider literature. The pretraining of the discriminator commences and it is trained for a period of 50 epochs. The current state post pretraining of the discriminator is evaluated. The result of this evaluation can be seen in Figure 8.3 for both the discriminator output spectrum and the mapping of the output as a function of the two dimensional input space. With the binary classification and softmax activation

the network output scores for the two classes sum to one. A low score on the decision(output) surface translates to a high predicted probability for the signal class and a high score translates to a high probability for the background class. The location and orientation of the boundary in this space suggest a good classification performance, however if the network were robust the output score across the three domains would not vary. This variation in network performance for the included domains indicates that this network is not optimally robust.

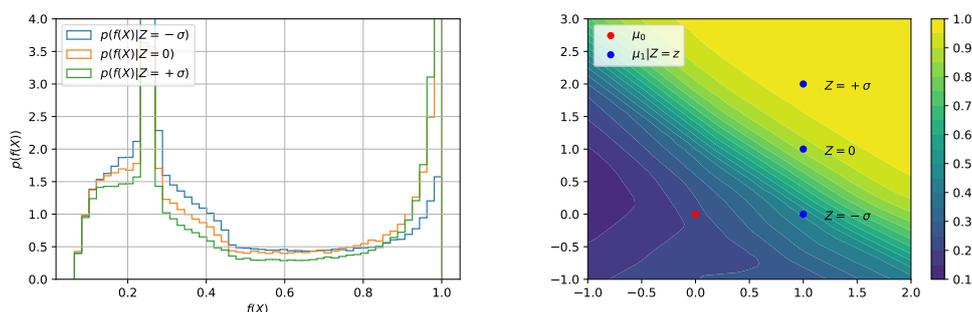


Figure 8.3: Evaluation of the pretrained state of the classifier in the domain adversarial adaptation method. The PID spectra subdivided by the domains(left) show a variation in network response for each domain. The domain with the largest distance between the means of the signal and background distributions shows the largest peak at high network scores. This is reflected in the decision surface (right) which visualizes the network output scores for each possible input.

With the discriminator pretrained the adversary can now be pretrained. This network is trained through the frozen discriminator and trainable adversary network configuration (the fourth network) for 20 epochs. With this training completed the adversarial training can commence. For each epoch in the adversarial training the discriminator is trained with the adversary's influence for an epoch. This employs the third network configuration as described previously. To complete the current epoch in the adversarial training the adversary is updated on the improved discriminator with the fourth network configuration for a single epoch. This is repeated for 200 adversarial epochs. The summary of the variations in losses during the adversarial training can be seen in Figure 8.4.

Post adversarial training the same evaluation of the discriminator is applied.

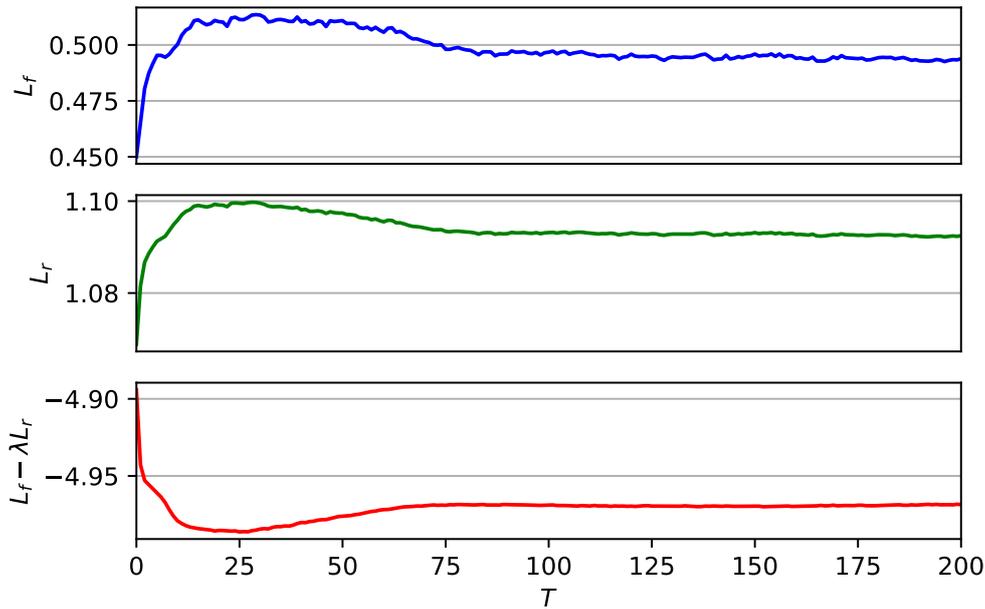


Figure 8.4: The loss curves of the domain adversarial training. The classifier loss (top) and the adversary loss (middle) show a similar behaviour, indicative of a balanced training. After pretraining ($T=0$) the inclusion of the adversary results in a decrease of the classifier performance, which in turn produces a equivalent effect in the adversary as it directly relies on the current state of the classifier. The combined training objective (bottom) decreases before reaching a steady state.

In the output spectrum the variation between the domains is still present although it has notably decreased in magnitude, for the center region around 0.4 to 0.7 the network output is nearly equivalent between the domains. A similar picture emerges when the decision surface is re-evaluated. The diagonal boundary which represents optimal classification has shifted towards a vertical boundary. This results in a more uniform response across the domains. With the prior knowledge of the domain encoding this matches the expectations for increased robustness. As the domain is encoded in the y coordinate, an optimally robust network would classify purely on the x coordinate. As this is suboptimal for classification a trade-off between the diagonal and vertical boundary is expected. Further optimization of the hyperparameters and the relative loss strength is possible. As the primary goal of the toy scenario is

achieved, namely a proof of concept, the optimization is left for future work.

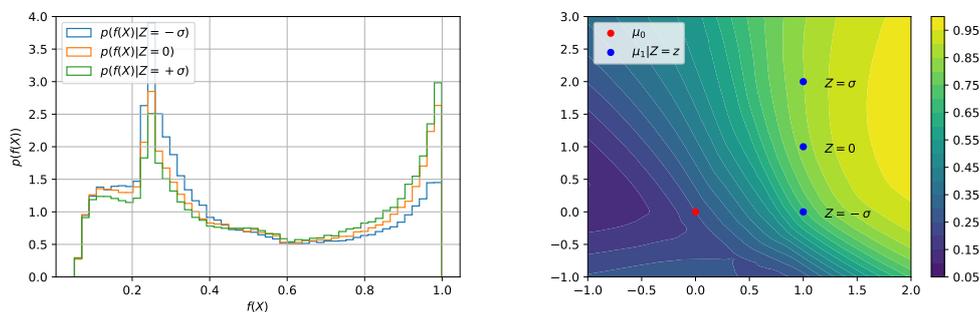


Figure 8.5: Evaluation of the adversarially trained state of the classifier in the domain adversarial adaptation method. The PID spectra subdivided by the domains(left) show a smaller variation in network response for each domain compared to 8.3, although a degree of variation between the domains remains. This is reflected in the decision surface (right) which exhibits similar network scores across the range of the domain shift.

The adversarial domain adaptation technique has been more widely applied beyond the binary classification toy example. Of particular relevance are the cases set in a high energy physics environment. An example with publicly available code and data are included in the original paper [157]. It is a binary classification problem for the discrimination of W boson jets in the Large Hadron Collider (LHC) from similar background jets produced from non W boson particle collisions and decays. The classification task is hindered by the presence of pile-up, where the degree of pile-up defines the domains. In appendix A this case and the effects of the applying this domain adaptation technique to it are examined in more detail.

8.2 Binary classification with gradient reversal

The advantages of the simple binary classification toy scenario can be leveraged for the purpose of validating other domain adaptation techniques beyond the domain adversarial training. For the evaluation of the gradient reversal technique two networks within the same toy environment as described in section 8.1 are employed. The first of these serves as the control and is a modified copy of the previous discriminator network. It is fully connected and consists of the

following layers.

- An input layer
- A 20 node tanh activated layer
- A 20 node relu activated layer
- A 20 node relu activated layer
- A 2 node softmax activated classifying layer

The architecture of the second network which includes the gradient reversal layer, has a similar depth but is wider due to the two output branches.

- An input layer, shared
- A 20 node tanh activated layer, shared
- A 20 node relu activated layer, shared. This layer forms the connection point from which the branches diverge
- A 20 node relu activated layer, classification branch
- A 2 node softmax activated output layer, classification branch
- A gradient reversal layer, domain branch
- A 20 node relu activated layer, domain branch
- A 3 node softmax activated output layer, domain branch

The gradient reversal layer has no trainable weights or parameters associated with it. It includes a hyperparameter which governs the relative strength of the inverted gradient during backpropagation. A value of 0 results in the gradients from the domain loss and branch having no impact on the shared layers, a value of 1 would result in an equal strength and higher values increase the relative impact compared to the gradient components originating from the classification branch. As the relative strength value is only applied

multiplicatively during backpropagation it does not directly impact the output values of the domain branch. Negative values are allowed as well, a value of -1 should result in normal dual branch classifier training and behavior. Inclusion of training with this value can therefore be a valuable crosscheck. For this training the relative strength is set to 10. The optimal value of the relative strength is dependent on multiple factors and is not a priori known. Contributing factors are the sizes of the losses and gradients for both branches, the size, complexity and relative simplicity of the domain shift and other environment specific variables.

Both networks are trained for a period of 100 epochs with default hyperparameters. Both networks employ the categorical crossentropy as the loss functions. For the dual branch gradient reversal network a separate weighting was introduced for the two branch losses. The domain branch was given a weight of 10 relative to the classification loss. The loss weights are separate to the relative strength of the gradient reversal layer, even though in this case their values are both equal to 10. This assignment was made to compensate for the difference in difficulty for the two separate classification problems. Training is performed on the weighted sum of the losses, which without the assigned weight results in an minimization of primarily the classification loss. The weighting restores some of the lost balance and enables the exploration of less optimal classification solutions. These are solutions which could be more robust. This weighting is implementation and scenario specific and therefore is not universally applied on all gradient reversal applications. Its specific value was determined through the comparison of the loss value fluctuations during training for each branch separately.

The loss curves are shown in Figure 8.6. The training with gradient reversal approximates the loss curve of the control network, as can be seen through a comparison of the blue and red loss curves. The influence of the gradient reversal layer can be identified through the differences between these curves. Most notably the larger degree of fluctuations during training and

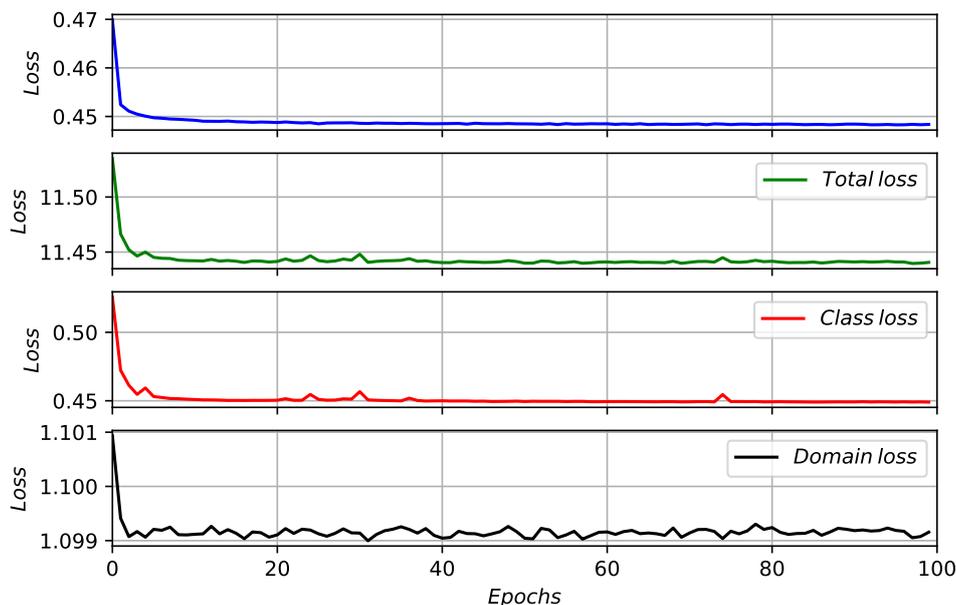


Figure 8.6: Loss curves for training with (bottom three) and without (top) the gradient reversal layer on the binary classification toy scenario. For the network with the GR layer included the total loss is the weighted sum of the classification loss and domain loss multiplied by the relative strength.

convergence to a higher loss value. Which would translate to a decrease in network performance. The loss component of the domain branch, shown in black, decreases quickly during the initial epochs as the network moves away from its random initial state and subsequently fluctuates around the same loss value. This is likely due to the conflicting optimization goals producing the parameter updates, especially for the early shared layers of the network.

The two trained networks are evaluated using similar methods as employed during the evaluation of the domain adversarially trained networks. These included the network output spectra and the decision surfaces. These can be seen in Figure 8.7 and demonstrate the effective increase in robustness when the gradient reversal technique is applied. Once more this toy scenario serves as a validation of the gradient reversal technique for the improvement of robustness against a domain shift. The network with gradient reversal applied is not fully optimised to demonstrate the optimal solution for this scenario.

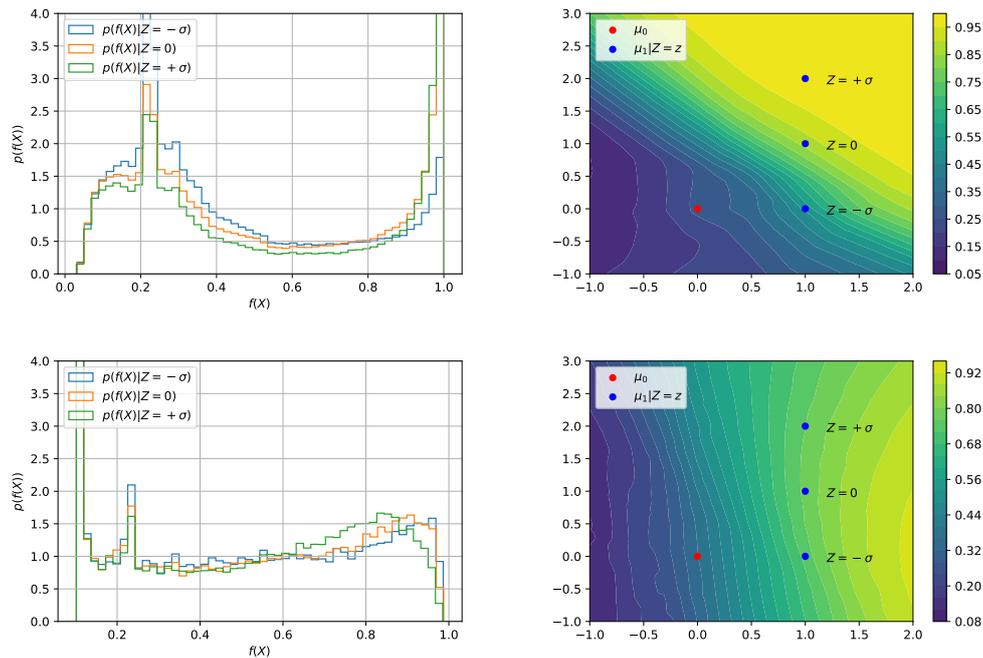


Figure 8.7: Evaluation of the impacts of the gradient reversal technique. The network trained without the gradient reversal layer included in its architecture (top) is not robust as both the PID spectra and the decision surface exhibit large variations across the domain shift. The network trained with the GR layer demonstrates minimal variation across the domain shift and is therefore more robust.

8.3 Binary classification with loss penalization

The loss penalization technique can similarly be benchmarked on the binary classification toy example. Additionally extensions to this toy scenario can be considered for further validation purposes. The binary classification task can be increased in complexity through the inclusion of an additional domain shifted class. This extension results in a multiclass classification scenario. Increases to the quantity of either the classes or domains produces internal histograms with reduced statistics which constitute a significant component in the loss penalization calculation. Likewise a reduction in the batchsize is expected to cause a similar degradation to the effectiveness of the loss penalization technique for the same reason. A comparative study of the loss penalization factor with or without the inclusion of memory is possible for this extended toy classification scenario.

Three network configurations are produced. The first network forms the control and consists of nine layers.

- An input layer for positional information (x,y)
- An input layer for the domain truth information
- A parameterless lambda layer which connects to both input layers, it solely allows the positional input to pass
- A 200 node selu activated layer
- A flatten layer, which only reshapes the incoming tensor
- A 200 node selu activated layer
- A 200 node selu activated layer
- A 200 node selu activated layer
- A 2 (3) node softmax activated classifying layer

The network size is increased compared to previously defined network configurations. The motivation for this change is the anticipated usage beyond the binary to the more difficult multiclass scenario. Furthermore the relu activation function is exchanged for the scaled exponential linear unit (SELU) activation function. This is a minor variation on the exponential linear unit activation function. A simple scaling is applied to ELU function to produce this activation function. As a larger variation in complexity of the task is expected the activation is exchanged so as not to encounter the possibility of dying relu nodes in these validation studies.

The inclusion of the additional input and lambda connecting layers is not required for the training and operation of this network. It is required for the other two networks and has therefore been included as an additional consistency measure. In this configuration it provides an additional cross check on the effectiveness of the lambda layer. If the network in this configuration

expresses robustness well beyond expectations it could indicate information leakage from the domain input layer into the wider network. Such a leak has not been observed in the development, training and evaluation of this network.

The second network configuration shares the architecture of the first. The output of the domain truth input layer still does not directly influence the subsequent network layers, but is available to be used in the custom loss function which includes the loss penalization factor calculation. This custom loss function implements the method described in section 6.6.3 from the available class truth, domain truth and network output. The loss penalization term is multiplied by the relative strength parameter and added to the usual loss function. For these three networks that loss is set to be the categorical cross entropy.

The third network configuration is of variable length. Its architecture is similar to the first and second network, but includes the previously described memory layers which act as a temporary storage buffer for the previous batch information required in the extended loss penalization calculation. Akin to the connected domain truth input layer this memory buffer is connected to the computational graph of the network, but is prevented from influencing subsequent network components. It is implemented as described in section 7.4.3.3. The output layers of the memory storage structure are included into the network output layer through the usage of the lambda layer which only lets the non-memory related information pass.

As memory layers are fully isolated the memory layer parameters cannot be updated through the typical methods applied during backpropagation. The infrastructure to enable the parameter updates for these layers is provided through the implementation of callback and metric functions, as described previously in section 7.4.3.4. This enables the storage functionality of the memory layers. Modifications to loss penalization method in the custom loss function employed by the second network allows for the readout of the memory output layers. The contained information is added to the current batch and passed on for usage in the loss penalization calculation. The depth of the

memory is set to five batches.

The relative strength parameter is set to 20 by default, although functionality for the variation of the strength is included. This enables further optimization if required. The weights and parameters of the networks are initialized to the same values for comparative studies.

The binary classification scenario dataset is generated with 100000 data samples in total. This dataset is split into a training and validation set of equal size (50%/50%). The networks are trained for a period of 50 epochs. Default values for the hyperparameters are used. The batchsizes are set to either 1000 data samples or 100 data samples. After training has completed the three network configurations, the control, the loss penalization and the loss penalization with memory, are evaluated. The produced decision surfaces which are shown in Figure 8.8 demonstrate the effectiveness of the loss penalization technique as a domain adaptation method. The decision surface for the control network exhibits the optimal classification solution in the form of a diagonal across the boundary of the two classes. The network output varies significantly across the domain shifted class. Both the networks with loss penalization and loss penalization with memory exhibit a combination between the optimal diagonal classification and the verticality of a reduced domain dependence. Application of the loss penalization technique has therefore increased the robustness and thus is effective as a domain adaptation method in this scenario.

A comparison between the two batchsizes does not demonstrate a clear and distinct difference for the inclusion of the memory in the loss penalization. A slight indication is present as the regular loss penalization network decision surface is less robust (less vertical) for the smaller batchsize. The reverse seems true for the loss penalization with the memory inclusion, it produces a more robust decision surface for a smaller batchsize.

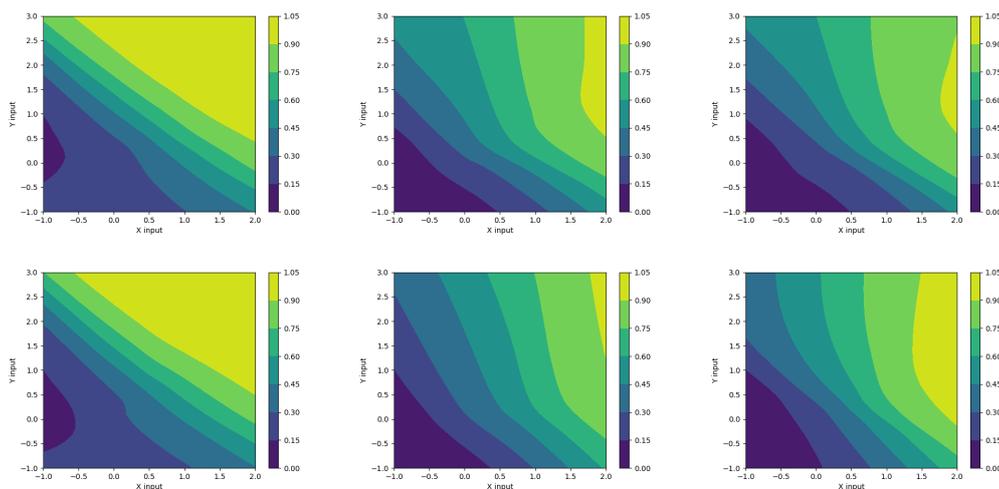


Figure 8.8: Decision surfaces of networks trained without (left), with inclusion of the loss penalization factor(middle) and with the inclusion of loss penalization and memory structures (right) for high (top) and low (bottom) batchsizes. The batchsizes are reduced from 1000 to 100.

8.4 Multi-class with loss penalization

The multiclass extension to the binary classification scenario is achieved through the introduction of a third class which is also domain shifted in the y coordinate. This class is superimposed on the two classes which constitute the binary classification scenario. It is generated from the following distribution,

$$x|(Z = z) \sim \mathcal{N} \left((0, 2 + z), \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \right) \quad (8.2)$$

where z takes the discrete values of $(-1, 0, 1)$. A schematic overview of the cluster centers generated from this and the other two distributions (see equation (8.1)) can be seen in Figure 8.9.

The training is configured with the same hyperparameters and number of data samples as for the previous binary classification scenario. Initially a training with a batchsize of a 1000 samples has been completed and evaluated. Both the PID spectra and the decision surfaces exhibit distinctive features which agree with the previous findings on the effectiveness of the loss penalization technique. For the binary classification scenario examination of a either class

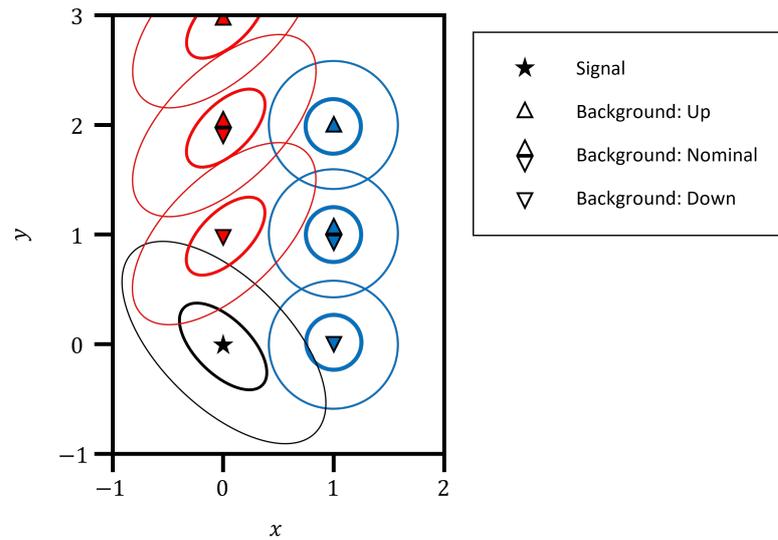


Figure 8.9: Schematic overview of the multiclass classification scenario. The means of the distributions from which data samples are shown for the signal and two background classes. The contours indicate the shapes of the two dimensional distributions.

is sufficient, as the remaining class is complementary due to its output vector summing to one with the softmax activation. The PID spectrum and the decision surface is therefore mirrored for the complementary class. Although the summation to one of the output vector is still true for the multiclass scenario, the third class introduces degeneracy. Therefore all three classes are shown individually.

In Figures 8.10 and 8.11 the PID spectra and decision surfaces are shown. For the control network the PID spectra and the decision surfaces show a classification focussed solution. The decision surfaces of the first two classes resemble those of the control network for the binary scenario. From the shapes and values of the distinct decision surfaces, the third class is the most difficult to classify correctly. This is to be expected as it has a higher degree of overlap between the two other classes. The features of the decision surfaces are reflected in the PID spectra as well. Both the first and second class show a higher degree of separation in the distributions than the third class. The PID spectra are produced from the element of the network output vector representing the

specific class. These can then be subdivided into signal, where the output class matches the truth label, and the background, where the output class does not match the truth label. For the examination of performance across the domains these distributions have been further subdivided into the separate domains. Variation between the domains can thus be easily seen, such as the case for the domain shift present in the first class. The down shift has the greatest overlap with the zeroth class (non shifted), whereas the up shift has the greatest separation in the input space. This is visible in the PID distributions, especially for the first class. The peak for the up shift is much greater than that for the down shift and this gives rise to larger degree of separation between the signal and background distributions, effectively leading to an improved classification performance on this domain.

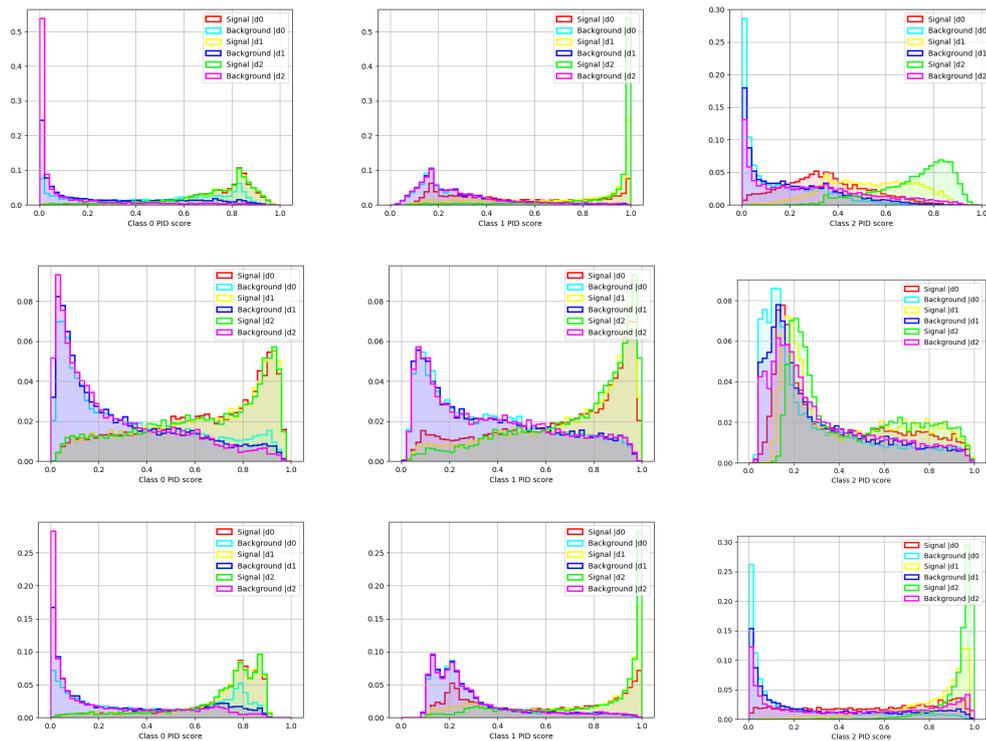


Figure 8.10: PID spectra of networks trained without (top), with inclusion of the loss penalization factor(middle) and with the inclusion of loss penalization and memory structures (bottom) for the three classes subdivided by domain. Class 0 is clustered in the bottom left hand corner of Figure 8.9, class 1 is in the top right corner and class 2 is in the top left corner.

The reverse also holds, for the PID distributions of the loss penalization

and loss penalization with memory, the variation across the domains of the signal and background distributions is much lower. Therefore the difference in performance across the domains is expected to be lower. For the inclusion of memory the multiclass scenario produces notable differences which are visible in both the PID spectra and the decision surfaces. The regular loss penalization produces decision surfaces which have sharp boundaries between the classes and show little variation across the domains (y coordinate shift of class one and two). The PID spectra reflect this with near identical signal and background distributions for the different domains. From the PID spectra it appears that the robustness gains have come at the cost of classification power, as a larger portion of the signal and background distributions concentrated in regions of overlap.

The loss penalization with memory appears less robust than the regular loss penalization but has retained a higher degree of classification power. Likewise reflected in the PID spectra and decision surfaces with the characteristic features. Nonetheless it outperforms the control network in terms of robustness.

8.4.1 Batchsize reduction

The comparatively diminished performance of the loss penalization with memory could be caused by the negative consequences of the memory inclusion. The inclusion of network output of previous batches introduces a potential delay into the learning of the network as these older batches do not reflect the current state of the network. Early in the training when the network parameters are updated significantly more than later in the training, this effect will have a larger impact. Additionally a higher value of the relative strength of the loss penalization to the regular loss should result in a higher prioritisation of the network robustness. These two changes should counteract the negative impacts of memory inclusion while retaining the benefits.

To test this hypothesis an additional test has been performed with the three network configurations and the multiclass classification scenario. Specifically the relative strength of the loss penalization network has been lowered to a

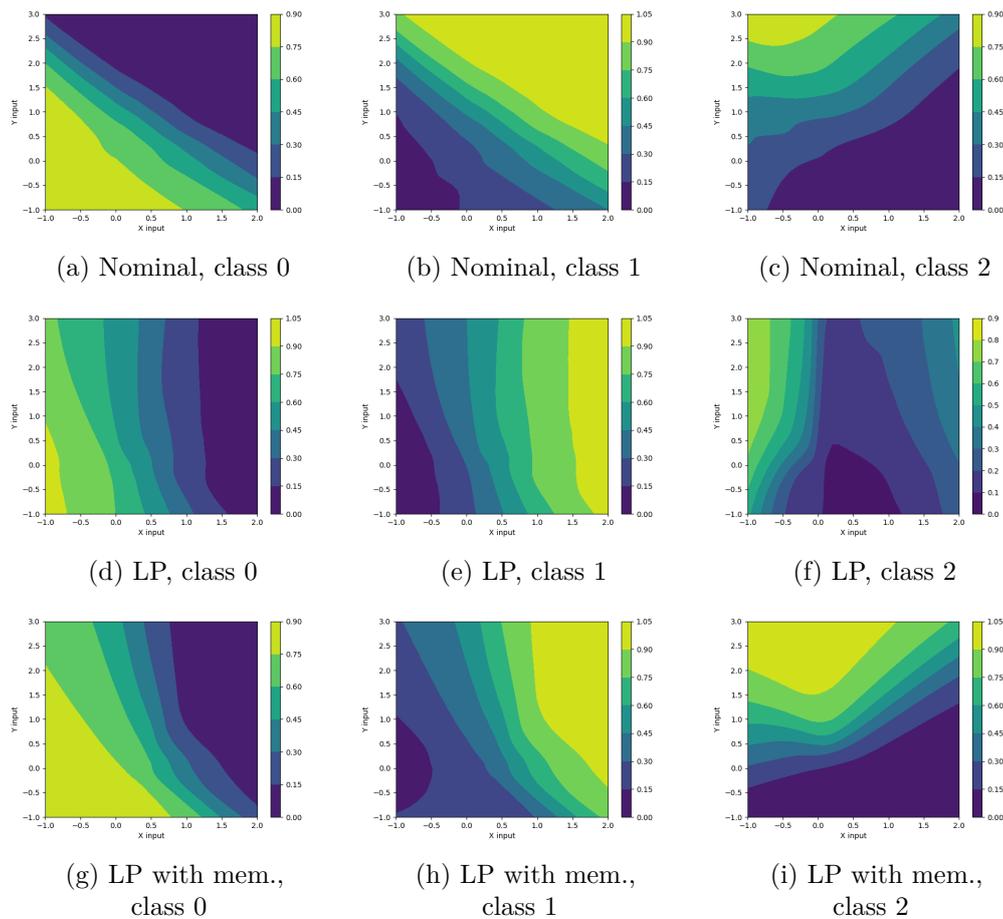


Figure 8.11: Decision surfaces of networks trained without (top), with inclusion of the loss penalization factor (middle) and with the inclusion of loss penalization and memory structures (bottom) for the three classes. The ordering of the classes (left to right) is equivalent to that of Figure 8.10.

value of 4. Additionally the batchsizes are varied between 1000 data samples and 50 datasamples to examine the variation of performance between the loss penalization and loss penalization with memory in low batchsize trainings.

The decision surfaces for the standard model training, the loss penalized training and the loss penalized training with memory are shown in Figures 8.12, 8.13 and 8.14 respectively.

The decision surfaces for the standard model training show little variation between the large and small batchsize and closely resemble the previous decision surfaces for a classification focused solution.

The network with loss penalization displays a high degree of robustness

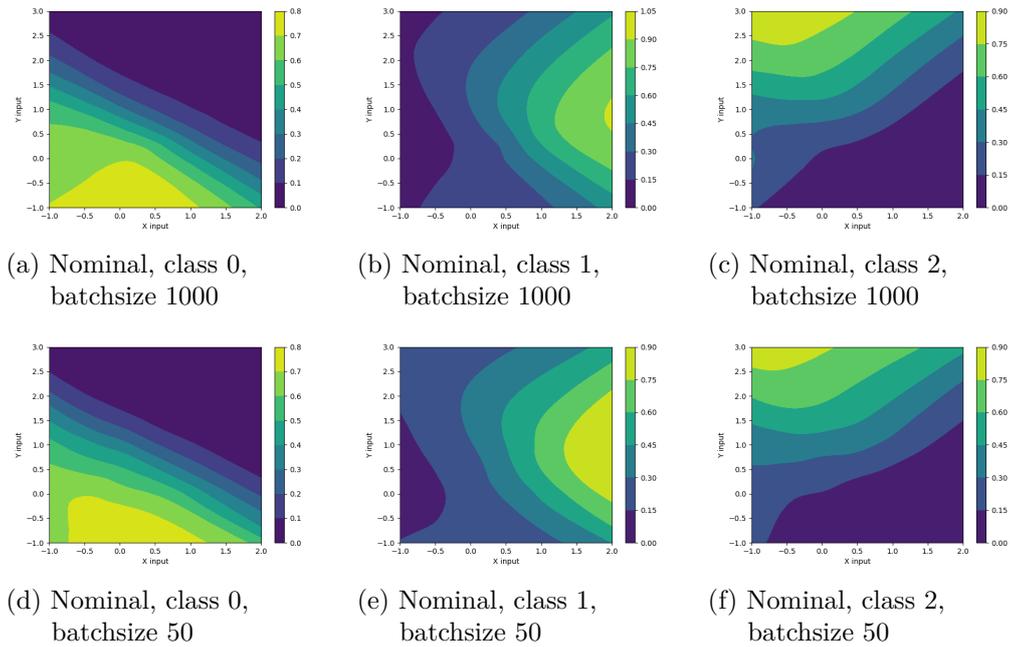


Figure 8.12: Impact of batchsize reduction on the standard model training on the multiclass classification scenario. The batchsize is reduced from 1000 (top) to 50 (bottom).

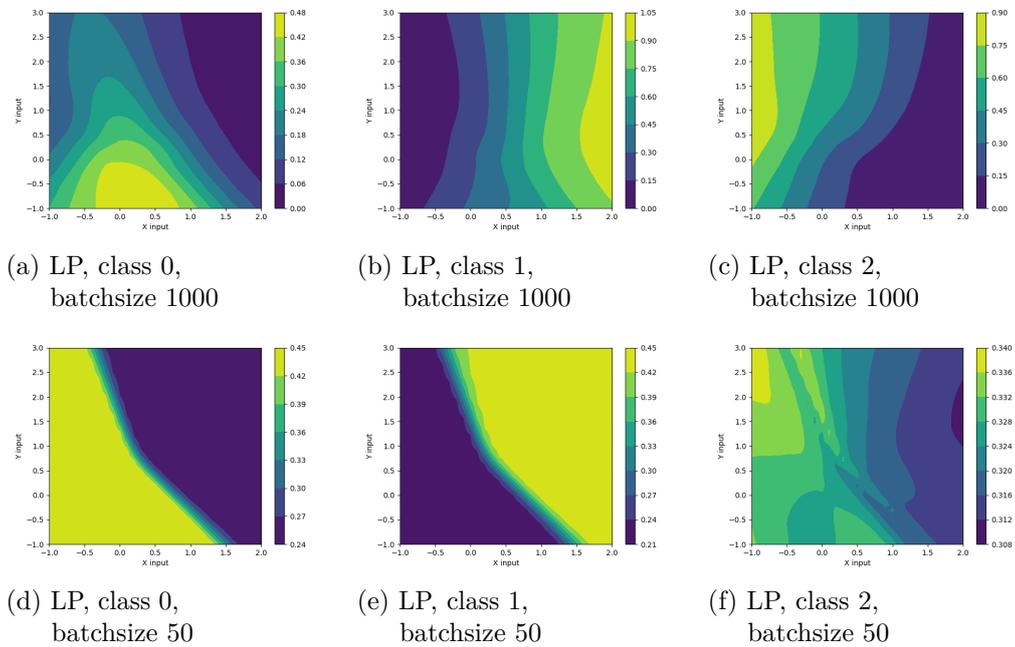


Figure 8.13: Impact of batchsize reduction on the loss penalized training on the multiclass classification scenario. The batchsize is reduced from 1000 (top) to 50 (bottom).

for the two domain shifted classes as the network output displays less variation across shifts in the y coordinate. As the separation between the classes is maintained and matches the locations of the clusters for each class the network can still perform the classification task. The decision surfaces for reduced batchsize show a sharp boundary in output scores and do not match the class cluster locations. This network is unable to perform the classification task to a sufficient degree.

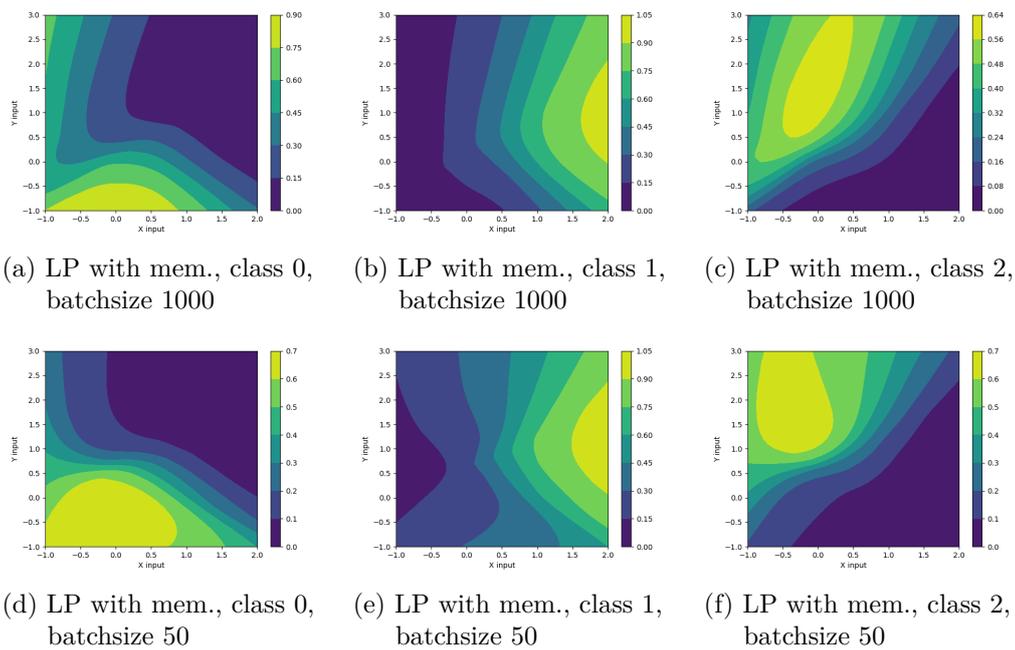


Figure 8.14: Impact of batchsize reduction on the loss penalized training with memory on the multiclass classification scenario. The batchsize is reduced from 1000 (top) to 50 (bottom).

The performance of the loss penalization network with memory appears more robust than the standard network although to a lesser degree than the regular loss penalization. The decision surfaces for the domain shifted classes show a greater variation across the y coordinate shifts than the regular loss penalization. The decision surface values in the top left corner of the input space indicates a small degree of potential misclassification. Unlike the regular loss penalization, the decision surfaces for the low batchsize more closely resemble those of the robust classifiers in the high batchsize settings. This demonstrates

that even though the performance of loss penalization with memory is slightly worse than the regular loss penalization for high batchsizes, it enables training in lower batchsizes environments.

Chapter 9

NOvA domain shifts impact and mitigation

The impact of the domain shifts on the NOvA event CVN can now be examined. This is done within the AdCVN framework through the evaluation of CVN networks trained on the included domain shifts. The general training procedure is set out in section 9.1. The magnitude of domain shift impact is isolated and examined for the CVN network trainings in section 9.2. This provides a baseline against the various domain adaptation techniques can be compared. The subsections contained in section 9.3 apply the techniques in order and demonstrate the degree to which they are effective in improving the CVN network robustness against the three examined domain shifts.

9.1 AdCVN training procedure

Training within the AdCVN framework closely approximates the training procedure [169] employed for the networks which are incorporated in the official NOvA analyses [70]. The primary focus of the official training procedure is the achievement of optimal network performance, on the limited number of networks which are required to be trained. This justifies the dedication of a significant amount of computational resources on the limited quantity of trainings. For the evaluation of robustness against domain shifts and the development of domain adaptation techniques within the NOvA neutrino

interaction environment a much larger quantity of trainings is required. As the computational resources for these studies are finite and shared for other research purposes, a need arises to streamline the training procedure. The divergences from the official training procedure are thusly motivated by the reduced training times and required computational resources. The resulting expected reduction in network performance compared to the official trainings originates solely from the variation in the training methodology. It therefore follows that this reduction applies equivalently across all networks trained with the same streamlined training procedure. This enables a fair comparative examination of the robustness of the CVN networks.

The primary differences between the official and streamlined trainings are the usage of hyperparameter optimization and transfer learning techniques. The Sherpa python package [188] is used for the purpose of hyperparameter optimization. This package includes various different algorithms which can be used for the efficient evaluation of different sets of hyperparameters. Performance of the optimization yields the optimal set of hyperparameters for a specific training. The transfer learning techniques are applied to effectively seed a initial point for specific trainings.

The majority of data processing and hyperparameter settings employed in the AdCVN trainings by default have been described in previous sections. The remainder include hyperparamters such as the batchsize which is set to 128 samples. The train/validation split is set at (80%/20%). The number of training steps(batches) in an epoch is set to 300 by default, although variations up to 3000 are not uncommon. The number of validation steps(batches) which are evaluated at the end of an epoch is set to 100 or 300, where the 100 steps are used if training time is a specific concern. The pixelmap value scaling is set at $\pm 10\%$, which multiplies the all the pixelmap values for an event by a random number within the interval of (90 – 110%). This measure is specifically included in the official trainings to aid in reducing the impact of the calibration and light level systematic uncertainties. A direct scaling cannot fully account

for the all the systematic effects, such as hits (dis-) appearing due to thresholds or energy and location specific variations. A study of robustness therefore requires evaluation with the systematically shifted samples. Additionally as this pixelmap scaling is included in the official training and it is expected to affect the robustness of the trained network, it needs to be similarly included in the AdCVN trainings if they are to be used for the study of the official CVN’s robustness.

Hyperparameter	Value
Batchsize	128
Fraction training sample	80%
Fraction testing sample	20%
Fraction cosmic events	10%
v_τ inclusion	False
Transverse momentum cut (ptp)	0.95
Architecture	Modified MobileNet
Training epochs	Variable(Early stopping)
Training iterations per epoch	300
Evaluation iterations per epoch	100
Pixelmap scaling	$\pm 10\%$ of pixelmap value
Learning rate	0.005
Optimizer	SGD
Learning rate decay	10%
Learning rate patience (epochs)	5
Domain batch balancing	True
Eventweights	True for GiBUUvsGENIE otherwise false
Adversarial strength (λ)	Variable (-1, 0 ranging to 1000)
Gradient reversal	Togglable
Loss penalization	Togglable
LP with memory	Togglable
Memory buffer size	5 batches

Table 9.1: Hyperparameters used in the CVN trainings.

SGD with a learning rate of 0.005 and a Nesterov accelerated momentum of 0.9 are used. The Nesterov momentum re-orders the list of operations involved with momentum in standard SGD. The standard method includes the sum of the current gradient with the diminished previously accumulated gradient.

This sum vector is then used for the subsequent update. Nesterov accelerated gradient (NAG) evaluates the current gradient at the location updated with the previously accumulated gradient. The current gradient is then used to make a correction as the proposed location from the previously accumulated gradient is not optimal. This implementation can be more efficient and tends to result in less overshoots and subsequently oscillations around the optimum [189].

During training the learning rate is reduced by 10% if the validation loss does not decrease for 5 consecutive epochs, although for specific trainings which include hyperparameter variation during the training this callback is disabled. Other callbacks employed during training provide checkpoint on each or specific epochs, which save the model state, and the early stopping condition which terminates training after a number of epochs if the network is no longer improving. The early stopping callback is only enabled if a set number of epochs to train for is not provided. An example training is shown in Figure 7.11 and an overview of the typical hyperparameters used is shown in table 9.1.

9.2 Single domain trainings

For the binary and multiclass classification scenarios the encoding of the domain shift is trivial, being a simple translation along one of the dimensions of the input space. For the proposed domain shifts against which the robustness of the CVN network is examined, the encoding of the domain shift is non-trivial. Even with the knowledge of what constitutes the introduced variation and how it is propagated into the data samples, it is difficult to pinpoint the variation as manifesting in the individual pixelmaps of the interactions. Judging the level of robustness directly from decisions surfaces with a complex domain shift is therefore impractical. Additionally the input space is significantly more complex which would further discourage the usage of this technique. Other methods, such as the statistical measures described previously, can be employed for the evaluation of robustness.

However complex the encoding of the domain shift may be, it is stored

in the individual data samples from which the network extracts information for the purpose of improved performance. If the network is capable of learning domain specific information (in the form of features), it will do so if it proves beneficial for classification performance. The behaviour of the network will thus diverge on the different domains, resulting in a low degree of robustness. If a network demonstrates high levels of robustness in the presence of a domain shift with a highly complex encoding two distinct scenarios are possible. Either the network is capable of learning from the domain shift to enhance its performance but exhibits true robustness regardless or the domain encoding is so complex that the network is incapable of learning and exploiting any possible domain dependent features of the data. To disentangle these two scenarios a different training is proposed.

For nominal trainings the primary objective is the interaction classification of the events, the presence of a domain shift is only a secondary influence. This holds even in the cases where domain adaptation techniques are applied which include some relative strength hyperparameter. To test the extent of the network's capabilities of extracting and leveraging domain specific information the training objective is altered. The interaction classification objective can be removed in favour of a domain classification objective. If the networks are incapable of correctly classifying any of the domains beyond the numbers achieved by a strategy of random guessing, it is indicative of the networks inability to learn domain dependent features of the data. If this is the case subsequent measurements of robustness may require additional scrutiny.

As the domain truth information is already present in the AdCVN framework a switch from classification to domain classification is a relatively simple exchange. The primary alteration has to be produced in the network architecture. The final classification layer needs to be updated to reflect the number of domains: two outputs for the exchange of the event generator and three for the calibration and light level systematics.

Training with the modified mobilenet in the domain classification con-

figuration is completed for the three included sets of domains. In Figure 9.1 the training and accuracy plots can be seen. These show a distinct improvement beyond random guessing, which would result in an accuracy of 50% for the two domains of Genie and GiBUU, and 33% for the triple domains of the systematics. As the performance does not outclass random guessing to a large degree, accuracies of (70 – 90%), it can be inferred that the domain classification task is not trivially easy for this network configuration. Likewise the notable instability of performance during training indicates a large degree of variation stemming from the updates the network makes during training. Typically during training the training loss is expected to be lower than the validation loss and the inverse for the accuracies. For these trainings that is not the case. There are various benign causes for this, such as differences in the application of regularization techniques between the training and validation. If the network behaviour is sensible, it can be safely disregarded.

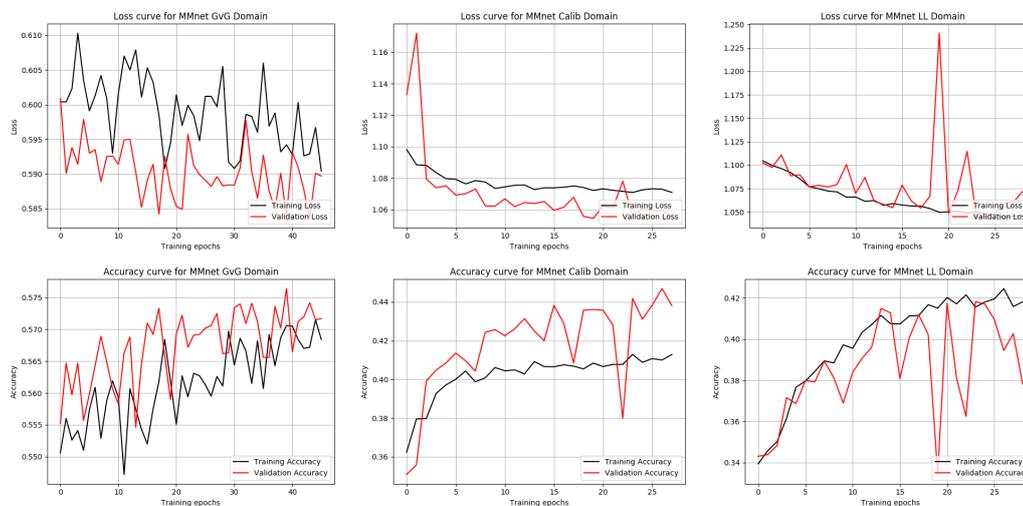


Figure 9.1: Training loss(top) and accuracy(bottom) curves for network in the domain classification configuration. The three domain shifts are trained separately, GENIE vs GiBUU(left), calibration systematic (middle) and the light level systematic (right). Although the training is unstable, performance for all trainings is increasing over the course of the training and is above the random guessing threshold.

The reporting of summary statistics at the end of an epoch, such as the loss and accuracies displayed in the training curves are evaluated at different points

in the training cycle. For the training in an epoch the network is updating continually, yet the values produced at the start of the epoch are incorporated into the reported statistic. For the validation the network is not updating and thus the reported validation statistics are reflective of the network state at the end of that epoch. The training statistics effectively show the average network state during that training epoch. Therefore large variation in network output, even if caused by small updates, will affect the reported training statistics more than the validation statistics.

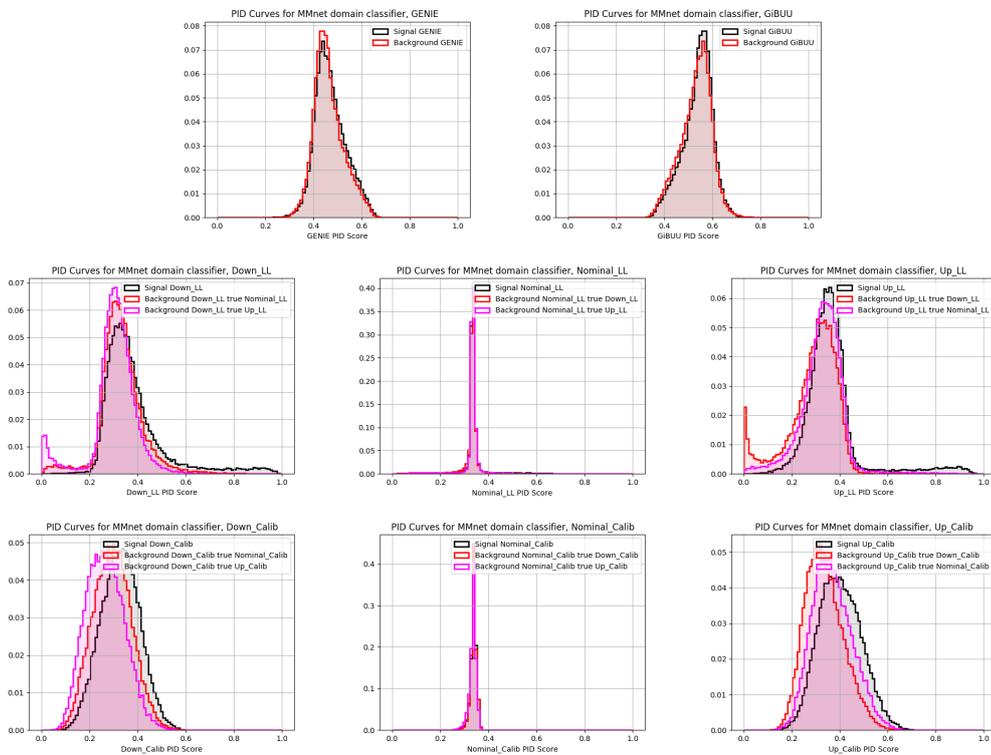


Figure 9.2: PID spectra of the networks trained in the domain classification configuration for each domain shift. The separation of the distributions indicates the ability to discriminate and classify the domains comprising the domain shift.

The reported performance displayed by the training curves of Figure 9.1 is reflected in the PID spectra as well. The PID spectra are shown in Figure 9.2. These spectra are split out into signal and background, with the background further subdivided into separate domains if applicable. The small separation between the signal and background distributions demonstrates the origin of

the domain classification above random guessing, while the large amount of overlap highlights the origin for the low overall classification performance and the general difficulty of the task.

Regardless these trainings demonstrate the networks capability to extract domain dependent information and leverage it for a domain classification task. It is therefore possible to surmise that this information is available to be leveraged by the event classification network in the pursuit of improved classification performance which is its primary objective. To what degree this actually occurs is examined in the following sections.

9.3 Domain adaptation trainings

The previously discussed domain adaptation techniques (see section 6.6) can now be applied to the domain shifted CVN trainings.

9.3.1 Domain adversarial

The first method for the evaluation of the CVN robustness against several domain shifts and the effects of employing a potential remedial measure is the employment the two network domain adversarial training technique. Implementation of the technique as described in section 6.6.1 encountered the issue of mode collapse. In this case the pretraining of the nominal CVN classifier progressed as expected. The pretraining of the adversary produced no network state capable of mapping the classifier output into a domain classification to a degree of accuracy beyond the random guessing threshold. The combined training steps, in which the classifier is updated with the inversed influence of the adversary and the adversary is updated with the improved classifier state, continually degraded the classifier without any significant change to the adversary. This training behavior is recognised as a form of mode collapse. It is caused by the imbalance in learning ability between the classifier and adversary. To enable usage of this technique different modifications to the training procedure have been tested. The primary purpose of these modifications is the rebalancing of performance between the two networks. For the classifier this

included reducing the pretraining time and reducing the aggressiveness of the learning through tuning of the associated hyperparameters (reduction to the learning rate, data samples in an epoch). This provided a range of pretrained classifier with varying levels of performance. Likewise similar modifications to bolster the performance of the pretrained adversary were made. Pretraining of the adversary, both with and without these changes, were attempted on the range of available pretrained classifiers. No usable state of the adversary network was achieved to a degree above random guessing. Variation of the relative strength between the classifier and adversary was planned but not executed as the required network states for successful combined training were not adequately present. Weighing the costs of further developmental testing against the chances of successful domain adversarial training resulted in the refocussing on other domain adaptation techniques which do not suffer from mode collapse.

9.3.2 Gradient reversal

The gradient reversal technique does not require multiple networks to operate and therefore does not suffer from the mode collapse obstacle. The implementation of the gradient reversal layer into the network architecture with the domain classifier forming a second output branch follows the description provided in section 7.4.3.2. The development, initial testing, validation of the implementation and training with the gradient reversal layer spanned several significant upgrades to both the CVN training framework and the base network architecture. Such as the exchange of the ResNet18 architecture for the modified mobilenet.

Although limited in scope the initial validation studies performed with ResNet18 architectures and the GENIE versus GiBUU domain exhibited behaviour in accordance with successful robustness-increasing domain adaptation. This initial comparative study follows a similar structure as the following larger scale examinations of robustness. A suite of networks is trained, and their performance compared based on a variety of previously discussed metrics. As

this study is limited in scope not all the metrics have been evaluated explicitly.

The suite of networks comprises a variety of networks trained on a single or multiple domains for the purposes of a comparative study. The first of the networks are trained on data samples originating from a single domain. For the case of the dual domains present with the exchange of the neutrino event generator, one of these networks is trained on only GENIE samples and one on only GiBUU samples. These two networks serve as the primary control group. In a conventional CVN training the network is trained on only the nominal domain. By evaluating the performance of these networks on all available domains the variation in performance on the various domains forms the benchmark of the CVN robustness against this domain shift. As these trainings serve as the control and are meant to closely resemble the conventional CVN training the domain adaptation techniques are explicitly not applied to them.

An additional network is trained on samples originating from both domains. This network is trained without the application of the domain adaptation techniques as well. It is an extension of the control networks and serves as a bridge to enable a fair comparison between the single domain trained networks and the networks trained with the domain adaptation technique applied. By construction, the networks with the domain adaptation techniques applied are required to be trained on multiple domains. The inclusion of this network can therefore be used to distinguish the potential benefits of letting the network learn on multiple domains to that of the inclusion of the domain adaptation technique.

The remaining networks in the suite are trained with various different hyperparameters choices for the relative strengths included in the domain adaptation technique. For this study the following strengths are included $(-1, 0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 100.0)$. The strength of -1 produces a network which is functionally identical to a normal two branched classifier. The inclusion of this network training provided an additional crosscheck, validation and

debugging method during development.

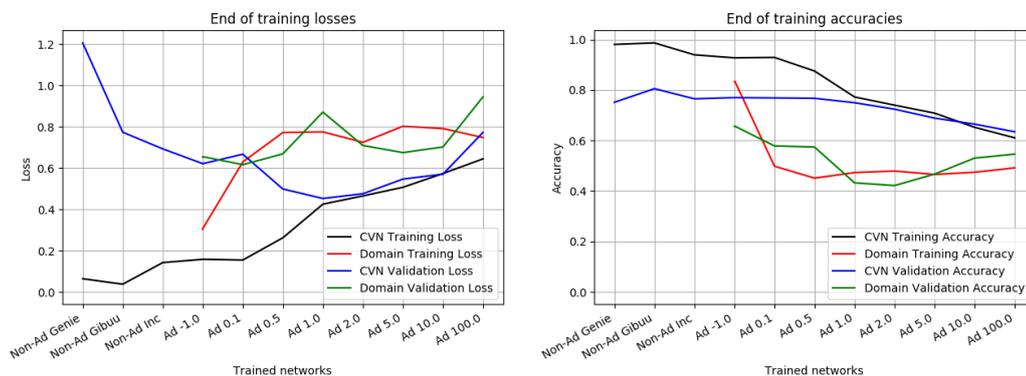


Figure 9.3: End of training losses(left) and accuracies(right) for a suite of ResNet18 networks with and without GR layers of various relative strengths.

These trainings have been completed with a training time of 25 epochs and are evaluated with $2.6 \cdot 10^5$ data samples. The recorded losses and accuracy values at the end of the trainings are shown in Figure 9.3. Given the shorter training times some statistical fluctuations are to be expected, but nonetheless some trends are visible in these figures. The networks without the gradient reversal technique applied are denoted by the Non-Ad signifier. As these networks do not possess a domain branch no domain training/validation losses and accuracies are recorded, this is reflected by the missing values in the plots. The performance of the three non adversarial networks appear relatively similar, although the network trained on the GENIE domain alone has a higher validation loss value for its final epoch. As its validation accuracy is in line with the other non adversarially trained networks the increased validation loss values can in all likelihood be attributed to one or several misclassified events producing a large contributions to the loss for that epoch. Given that these trainings for this dataset are employing the event weights this explanation fits with the reported values.

The lower strength adversarial trainings, which are referring to the networks with the gradient reversal technique applied, demonstrate similar classification performances as the non adversarial networks. For increasing strength the classification performance suffers. This is in accordance with the expectations

of a larger influence of the inverted gradient produced by the domain branch and the gradient reversal layer on the early shared feature extraction layers. For the strength of -1 the domain branch performs as an ordinary domain classifier and achieves domain classification accuracies well in excess of the random guessing threshold of 50%. For increased strengths the performance of the domain classifier suffers as the shared layers are trained to be less dependent on domain identifying information. This results in a reduction of information which the classifier network, post gradient reversal layer, can leverage to perform the domain classification task. This is reflected in the decreasing accuracy values for the increased strengths, eventually trending towards the random guessing threshold. For the higher strengths the domain classification accuracy appears to trend upwards, this could be due to the continual parameter updates with the relatively large inverted gradient contributions resulting in the network parameters values being far from an optimal solution. In this scenario the regular classification performance would suffer and due to the large and continual parameter updates it is not inconceivable that some domain relevant information leakage can occur through the shared feature extraction layers. The domain classifier could leverage any leakage for the purpose of increased domain classification performance. Alternatively leakage could be a constant factor, but stabler updates and training of the lower strength could limit the amount of leakage to low enough levels where it is unable to be leveraged successfully by the domain classifier.

9.3.3 Robustness evaluation

Within the AdCVN framework with the modified mobilenet architecture an evaluation of the CVN robustness and the impact of the gradient reversal technique is performed. It mirrors the previous comparative study but is larger in scope. It includes significantly longer trainings, all of the available domains and a more extensive evaluation.

A suite of networks is trained for each set of domains. It includes the single domain trained networks, the inclusive network and the adversarial

networks with the gradient reversal layer. The latter are trained with strengths $(-1, 1, 10, 100)$. The trainings do not have a strictly defined endpoint, instead either the early stopping callback activates and stops the training or the training times out due to the resources limitations in place on the WC-IC. The time-out for these trainings does not occur before 24 hours of training time.

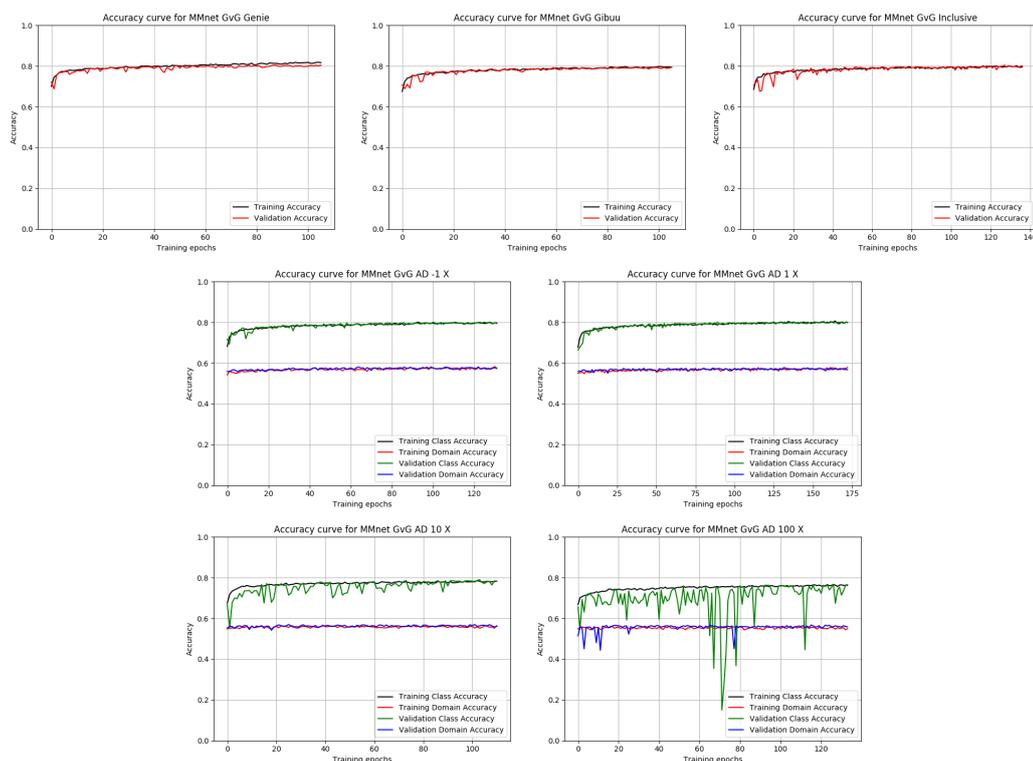


Figure 9.4: Training accuracy curves for a suite of networks trained with and without GR layers on the GENIE vs GiBUU domain shift. The class accuracy for the adversarial networks with two output branches is equivalent to the reported accuracy of the single output networks (GENIE, GiBUU, Inclusive).

The accuracy training curves are shown in Figures 9.4, 9.5 and 9.6. The loss curves display similar behaviour of the networks during training, however the interpretability of the accuracy curves is greater than that of the loss curves. For this reason only the accuracy curves are shown here. All of the trained networks included exhibit successful trainings with a comparable high degree of classification performance. Several trends across the sets of domains can be identified from these accuracy curves. The networks trained with strength -1 achieve domain validation accuracies well in excess of the random guessing

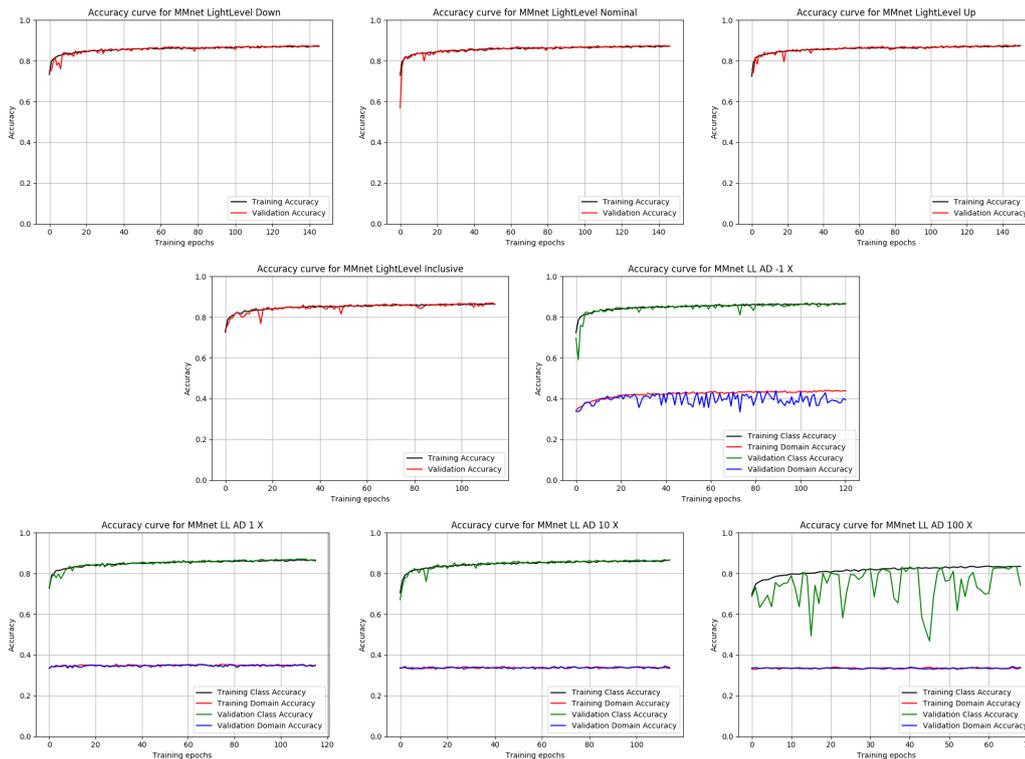


Figure 9.5: Training accuracy curves for a suite of networks trained with and without GR layers on the light level domain shift.

thresholds, which matches expectations. The domain training accuracies for these networks appear to diverge from tracking the validation accuracies to an extent. The potential reasons for a difference between training and validation loss and accuracies elucidated previously can explain this behaviour once again. The higher strength network trainings are unstable in the classification accuracy and reach notably lower values than the non adversarially trained and low strength networks. The suffering of classification performance is expected due to the increased influence of the inverted gradient components originating from the gradient reversal layer and domain classification branch. The checkpointing callback function provides the latest optimal version of the network for usage in the subsequent evaluation.

For the GENIE versus GiBUU domains the random guessing threshold in these plots is higher than the expected 50%. As the reported accuracies in these curves are unweighted the threshold for random guessing is set at

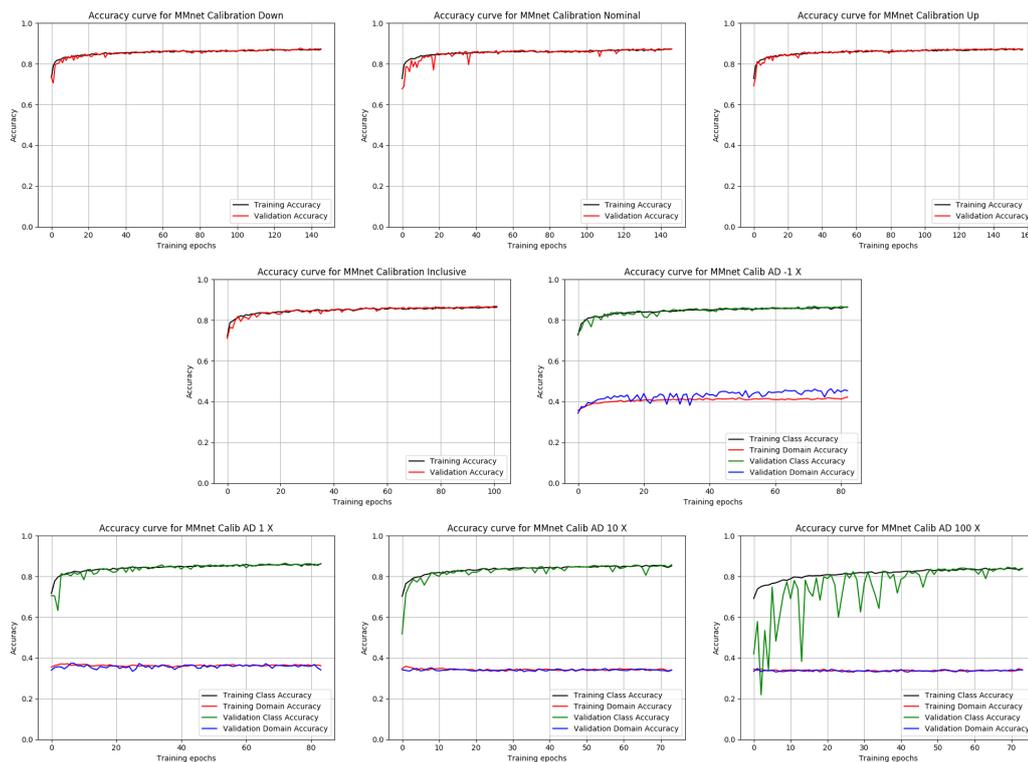


Figure 9.6: Training accuracy curves for a suite of networks trained with and without GR layers on the calibration domain shift.

approximately 56%.

Each of the trained networks is evaluated by propagating a larger number of data samples which are purposely excluded from the training dataset through the networks. The truth information of these samples is collected alongside the network output prediction scores for every available branch. These evaluations are run as a standalone job after the trainings are completed. The steps in the training, evaluation, post evaluation processing and plotting are separated as the initial steps take a relatively long time to complete compared to the latter two steps and have different computational resource requirements. Additionally this separation allows for rapid iteration of the processing and plotting scripts, which is advantageous during the development stages.

With the truth information and the network output scores the PID spectra are produced first, one for each class and domain contained in the network output vectors. Leveraging the accompanying truth information, the PID

spectra can be subdivided into a signal and background distribution. The PID spectra for the class outputs can be further subdivided per domain and the PID spectra for the domain outputs can be subdivided per class.

The distributions in the PID spectra exhibit signs of robustness and classification performance. For the class PID spectra, a subset of which is shown in Figure 9.7, visible variations in the shapes of the domain subdivided signal and background distributions are indicative of the direct negative impacts of the domain shift. A lack of variations is likewise indicative of adequate robustness to the present domain shift. From the training curves it is clear that the network performances are relatively similar, and this is reflected in the class PID spectra as these are similar across the networks as well. Although small there are some variations present, which can be further examined through the applications of the previously mentioned statistical measures.

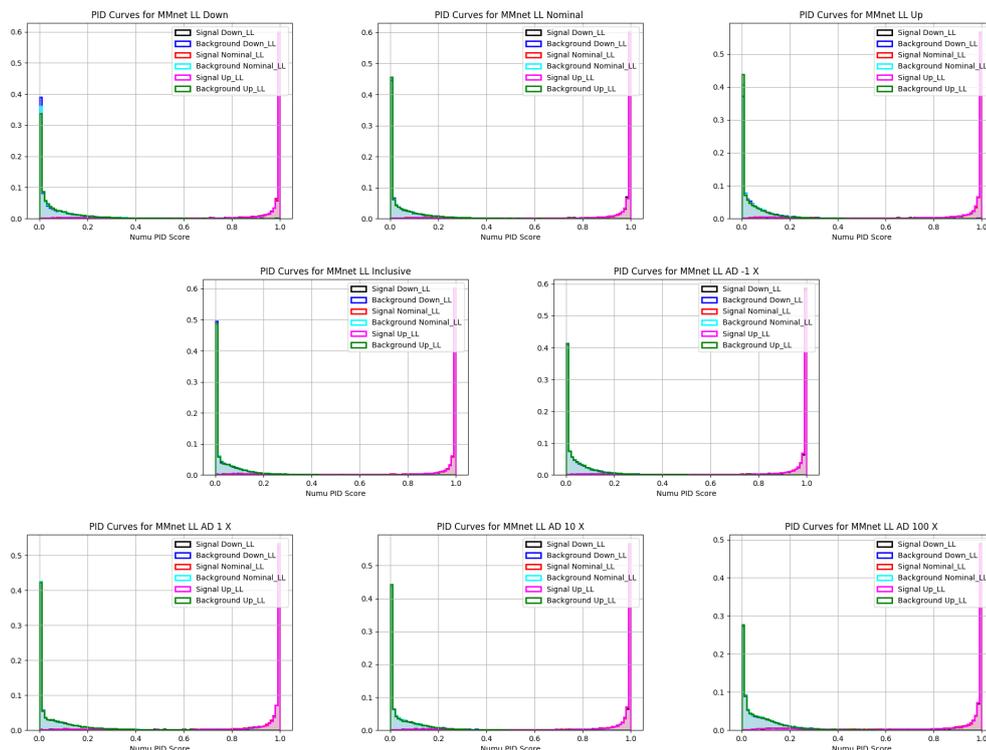


Figure 9.7: A representative subset of the class PID spectra for the networks trained on the three domain shifts. This set contains the class PID spectra of the network suite with and without the GR layer on the light level domain shift for the ν_μ class.

The differences for the domain PID spectra are much greater. As with the class PID spectra the classification performance of the domain branch is visible through the degree of separation between the signal and background distributions. Increased relative strength of the gradient reversal layer has a significant impact on the output of the domain branch. The signal and background distributions are moving closer together toward a central value for an increase of the relative strength. For higher values of the relative strengths the influence of the inverted gradient produces network updates which result in less domain dependence in the early network layers. Without this the later domain classification branch should be unable to produce domain predictions beyond random guessing. This trend is demonstrated in the domain PID spectra for increasing relative strengths visible in Figures 9.8, 9.9 and 9.10.

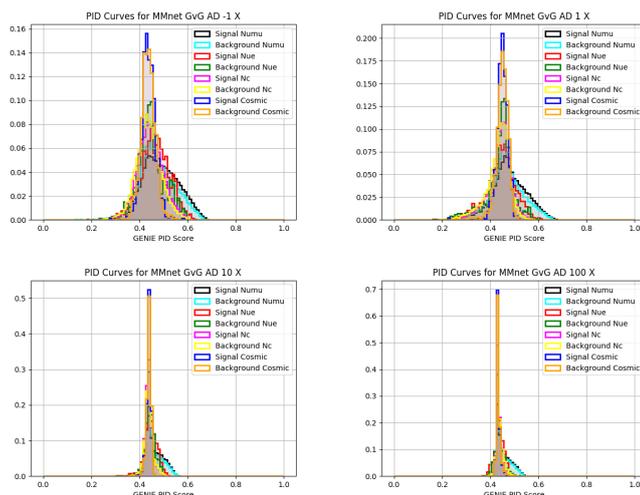


Figure 9.8: Domain PID spectra for the networks trained on the GENIE vs GiBUU domain shift. Only the GENIE domain PIDs are shown as the GiBUU domain PIDs are their reflections due to the unitary sum imposed by the softmax activation function. For increasing strength both the width and separation of the signal and background distributions decreases indicating a reduction of domain discrimination.

The cosmic class is uniquely well separated in the light level domain PID spectra, which means that the discrimination between the domains is comparatively simpler for this class. This is due to the implementation of the light level shifts not affecting the simulation of the cosmic rays. The subsequent

counter shift of the calibration energy levels is applied at a later stage and does affect the cosmic rays. This results in a larger variation for events in the cosmic class.

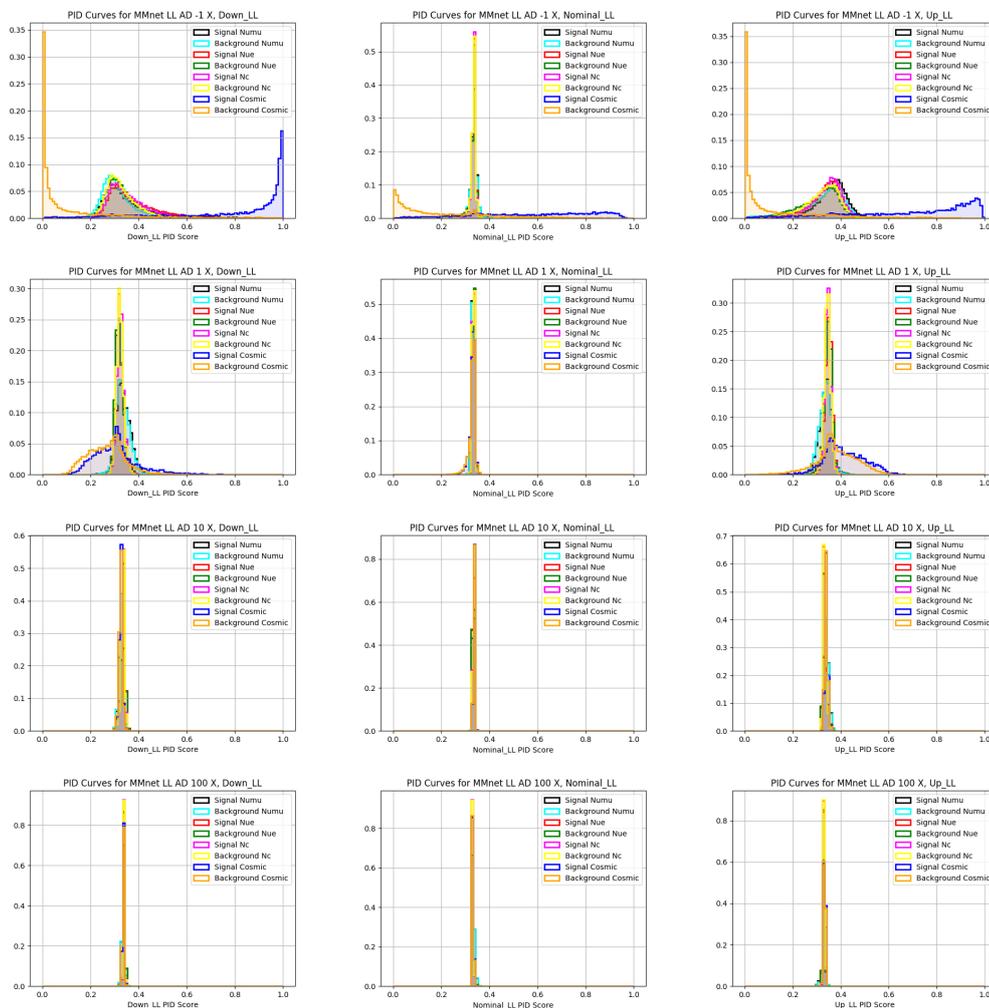


Figure 9.9: Domain PID spectra for the networks trained on the light level domain shift. The down shift(left), nominal(middle) and up(right) shifted domains are shown for increasing strengths (top to bottom). For increasing strength both the width and separation of the signal and background distributions decreases indicating a reduction of domain discrimination.

Further examination of the class PID spectra yields a measure of the networks classification performance. This is achieved through the production of the accompanying ROC and PR curves. The ROC and PR curves have been described in section 7.4.4.6. Similarly to the PID spectra these are relatively similar across the networks trained for each suite of domains. Examples of

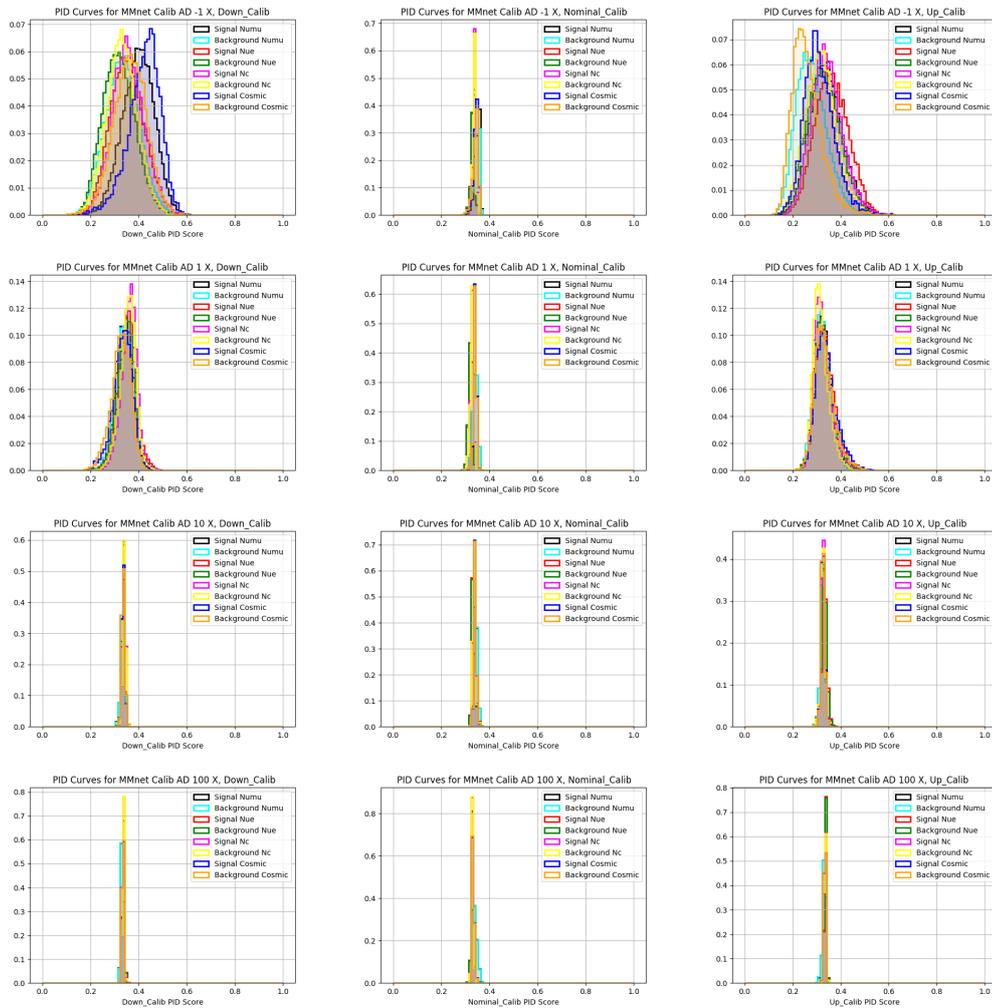


Figure 9.10: Domain PID spectra for the networks trained on the calibration domain shift. The down shift(left), nominal(middle) and up(right) shifted domains are shown for increasing strengths (top to bottom). For increasing strength both the width and separation of the signal and background distributions decreases indicating a reduction of domain discrimination.

these curves are shown in Figures 9.11 and 9.12. The two networks trained on either the up or down shifts show the largest variation in performance across the domains as can be seen by the separation between the domain separated curves. By the same evaluation method the nominal network is more robust as it has a negligible variation. The classification performance is similar between the evaluated networks and only starts to decrease for the higher strengths. The decrease is visible in the PR (ROC) curves as a movement away from the top right(left) corner.

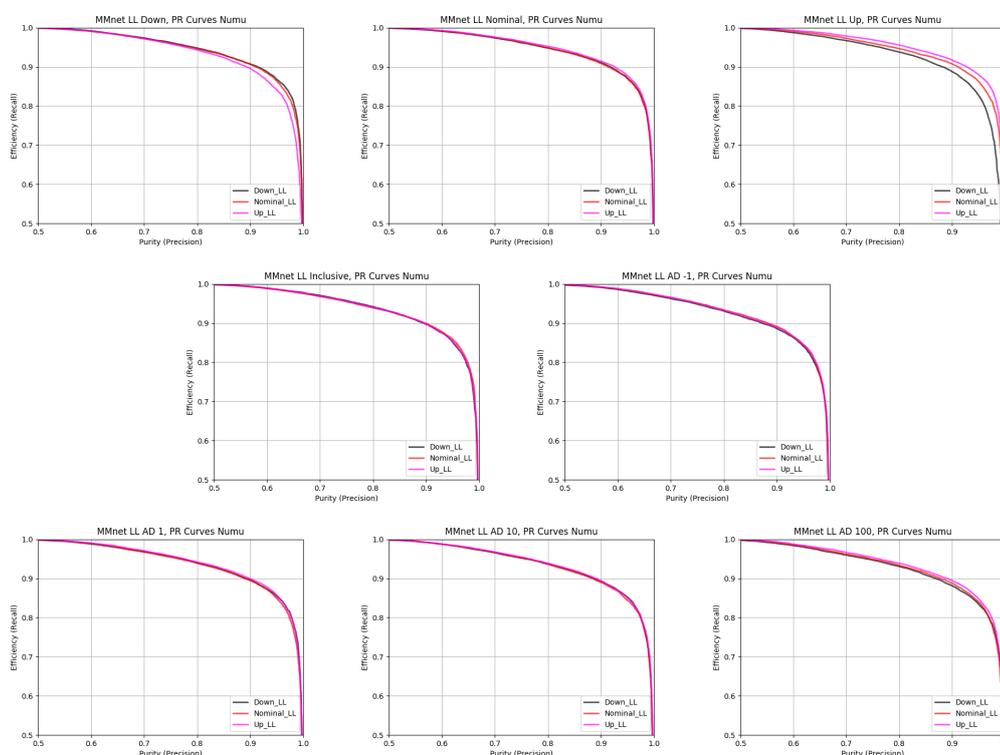


Figure 9.11: Precision and recall curves for the networks trained on the light level domain shift. These are produced for the ν_{μ} class and originate from the PID spectra shown in Figure 9.7. The similarity of the domain separated curves yields an indication of the networks robustness.

For the purposes of robustness evaluation the variations of the domain specific signal and background distributions of the class and domain PID spectra can be quantified through the usage of the previously described statistical measures. The difference between two comparable distributions are produced, for example the signal distributions between two domains for a given class.

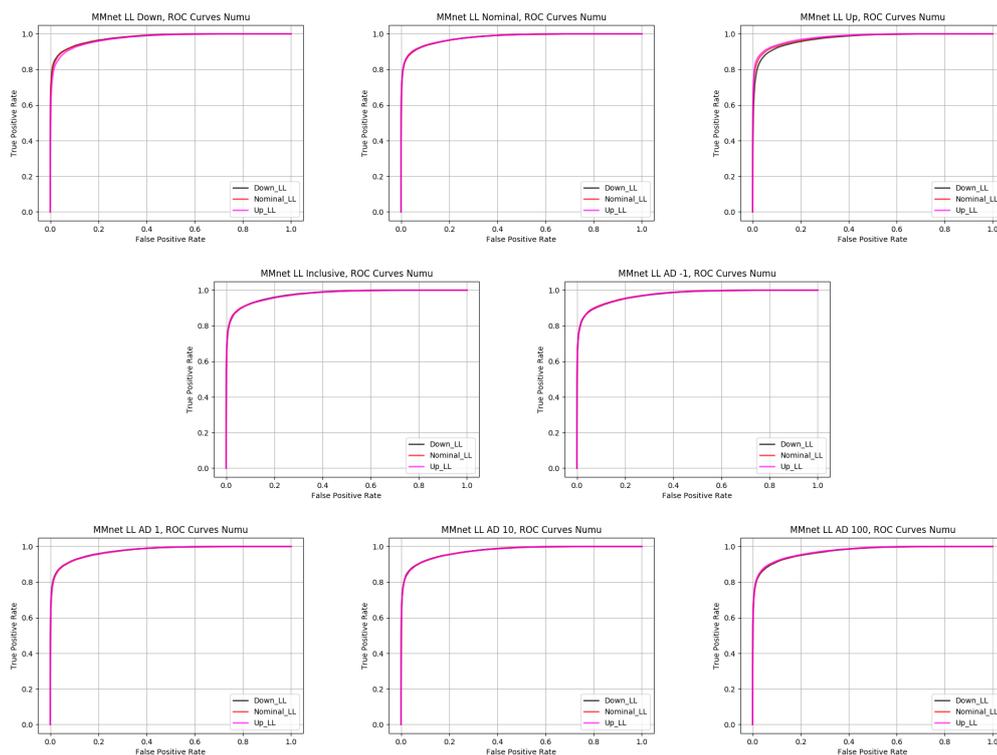


Figure 9.12: Receiver operator curves for the networks trained on the light level domain shift. These are produced for the ν_{μ} class and originate from the PID spectra shown in Figure 9.7. The similarity of the domain separated curves yields an indication of the networks robustness.

This is repeated for all combination of domains for both the signal distributions and the background distributions separately for each class. For the domain PID spectra these comparisons are produced from the comparisons between the signal and background distributions per class directly. As in the domain PID spectra the signal to background comparison is a comparison between domains.

The domain PID summary statistics across the networks for a specific domain shift exhibit the same trend identified directly in the PID distributions. For increased relative strengths the variation between the distributions across the included domains shifts decreases in value, which is equivalent to becoming more similar and therefore more robust. This trend can be seen in Figure 9.13. This is indicative of the domain branch output displaying the expected behaviour in the case of successful deployment of the gradient reversal layer as a robustness increasing technique for these domain shifts.

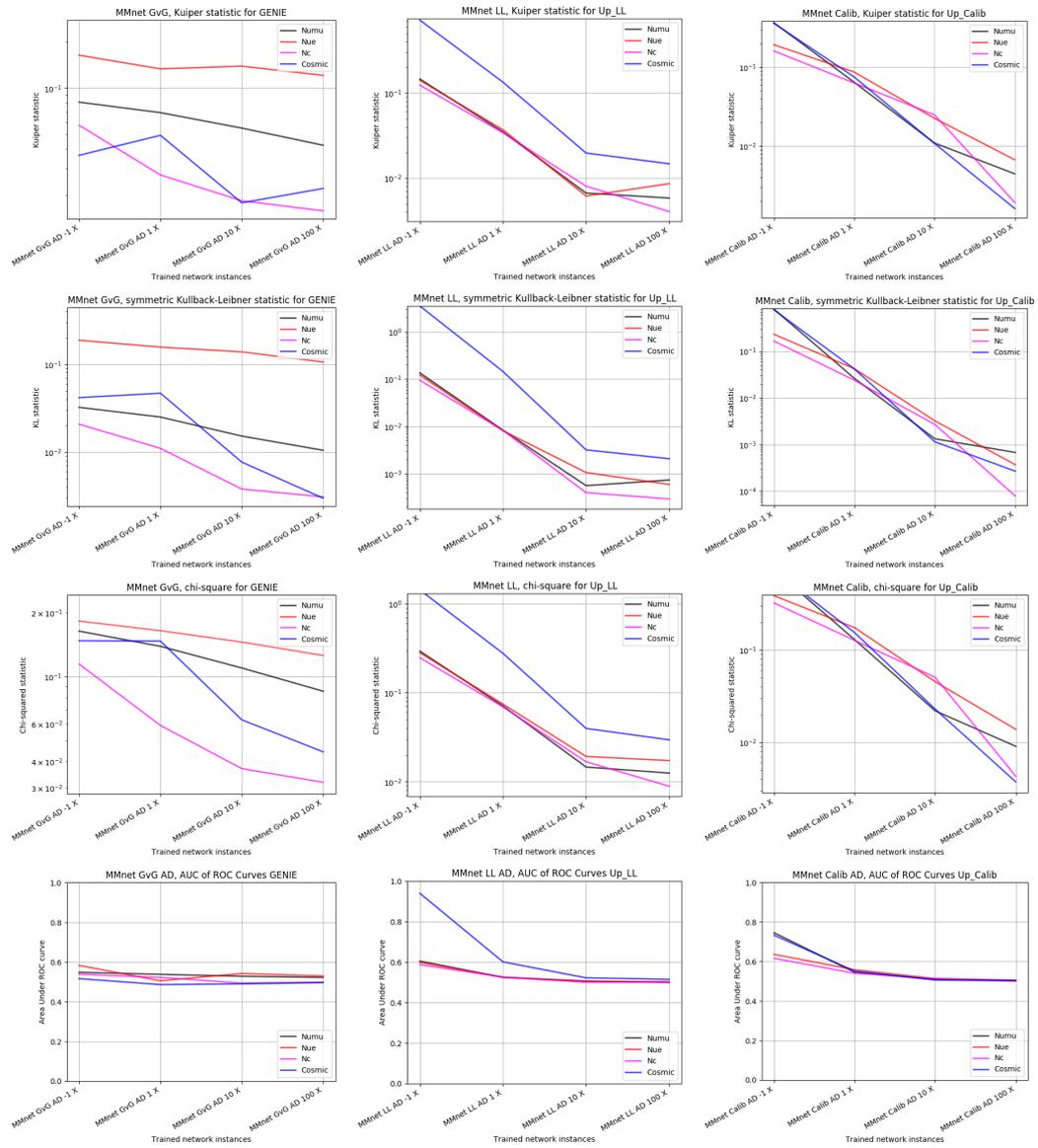


Figure 9.13: Representative subset of the summary statistics of the domain PID spectra for the networks trained on the three domain shifts, the GENIE domain (left), the upshifted light level domain (middle) and upshifted calibration domain(right). The summary statistics (top to bottom) are the Kuiper, Kullback-Leibner, χ^2 and area under the ROC curve. For increasing strength the values of the summary statistics and the variation across the domain shifts decrease which is indicative of increasing robustness.

While for the domain PID distributions and the derived statistics the domain classification is entangled with the measure of the robustness, this is explicitly not the case for the classification PID distributions and the derived statistics. Classification performance for the domain output is expressed, in part, as the separation between the signal and background distributions. This separation is the variation of the output on different domains, which is employed as the primary expression for robustness. For the classification output these two metrics are separate and distinct, as the signal and background represent different distributions in this case. The classification performance is similarly expressed as the separability between the signal and background distributions, while the robustness metric is the variation of either the signal distributions across the domains or the background distributions across the domains. Comparisons between the signal to signal or background to background distributions with the previously described measures yields a mostly performance-independent measure of robustness. This does not hold for scenarios where the network performance is degraded to the point of complete failure where the PID distributions are no longer representative of valid network output.

In Figures 9.14, 9.15 and 9.16 the classification summary statistics across the three domain shifts are shown. The variations of these do not demonstrate any clear trends towards an increased level of robustness for increases to the relative strengths of the gradient reversal layer. Even though the impact of the domain adaptation technique is clear from the earlier domain PID spectra and associated statistics, the values of the statistical metrics used for the evaluation of robustness do not decrease with a higher degree of influence from the inverted domain gradient components. The reported values for the networks trained without the gradient reversal on a single domain (LL upshift, LL downshift for the ν_μ and Cosmic classes) are higher than those of the nominal, inclusive and gradient reversed networks. This indication of an identifiable trend does not generalize to all classes and domain shifts. The expected robustness increase

should be visible as a decrease in the summary statistics with increased values of relative strength, such as in Figure 9.13 for the domain summary statistics. This trend is notably absent in Figures 9.14, 9.15 and 9.16. While an improvement to the robustness against the domain shifts is absent, the low values of the Kuiper and Kullback-Leibner statistics coupled with the high and consistent (across the domains) values of the area under ROC curves would suggest that the classification performance and robustness of these CVN trainings are high.

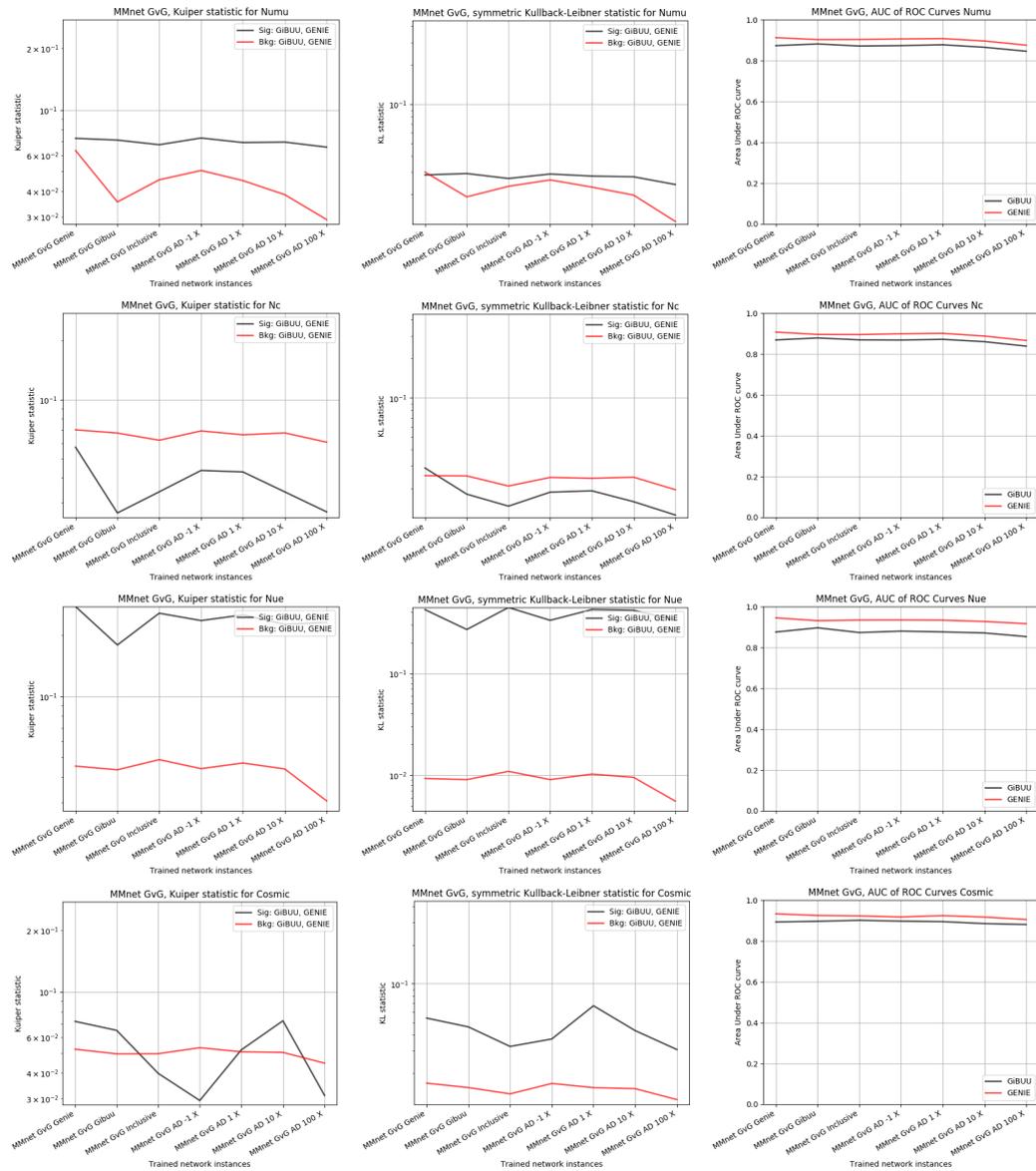


Figure 9.14: Summary statistics of the class PID spectra for the networks trained on the GENIE vs GiBUU domain shift. The four interaction classes are shown top to bottom (ν_μ , N_c , ν_e and Cosmics) and the Kuiper, Kullback-Leibner and area under the ROC curve are shown left to right. No clear trend for the robustness improvement, as defined through a decrease in the reported Kuiper and Kullback-Leibner statistics, is identifiable.

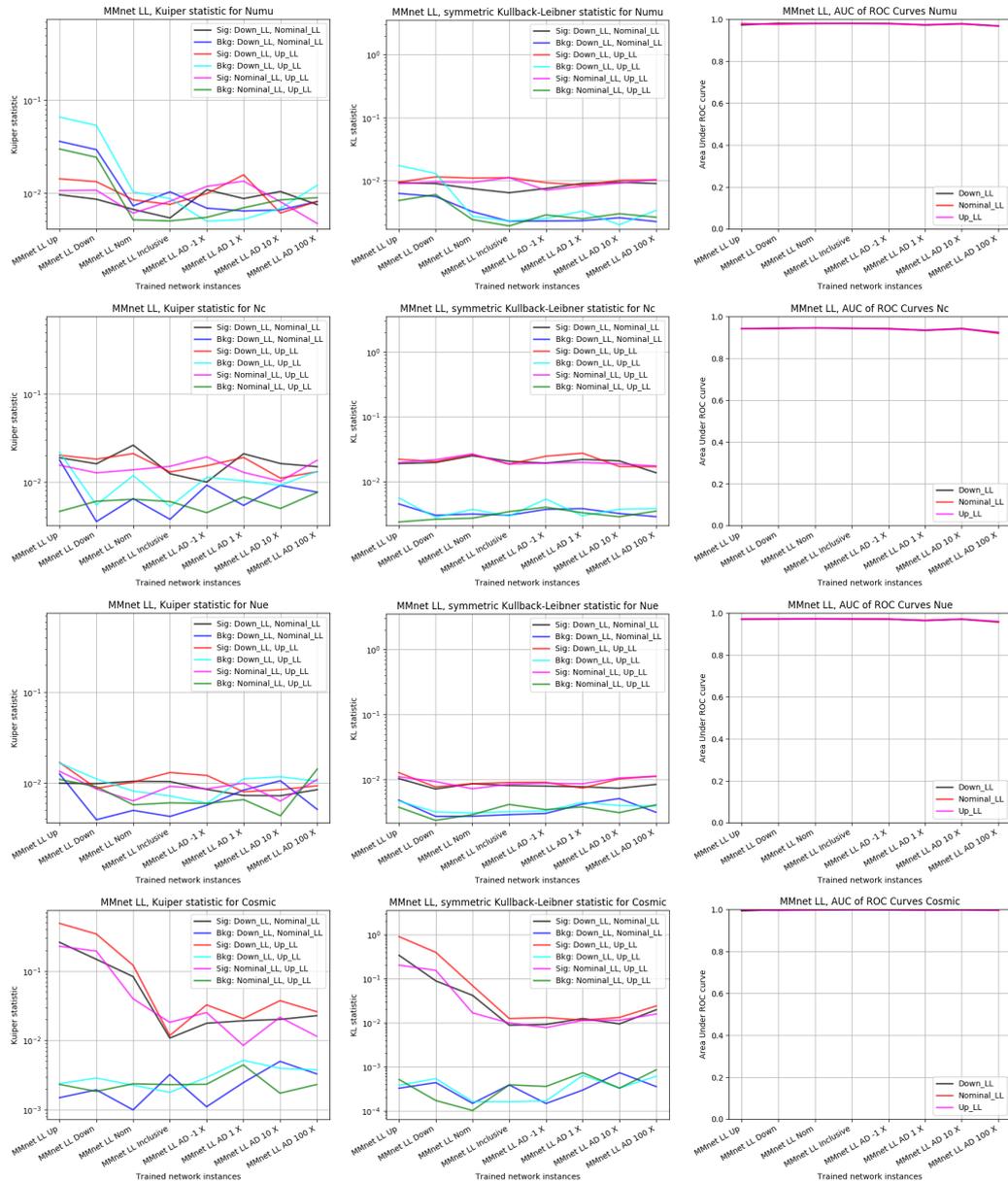


Figure 9.15: Summary statistics of the class PID spectra for the networks trained on the light level domain shift. The four interaction classes are shown top to bottom (ν_μ , NC, ν_e and Cosmic) and the Kuiper, Kullback-Leibner and area under the ROC curve are shown left to right. No clear trend for the robustness improvement, as defined through a decrease in the reported Kuiper and Kullback-Leibner statistics, is identifiable.

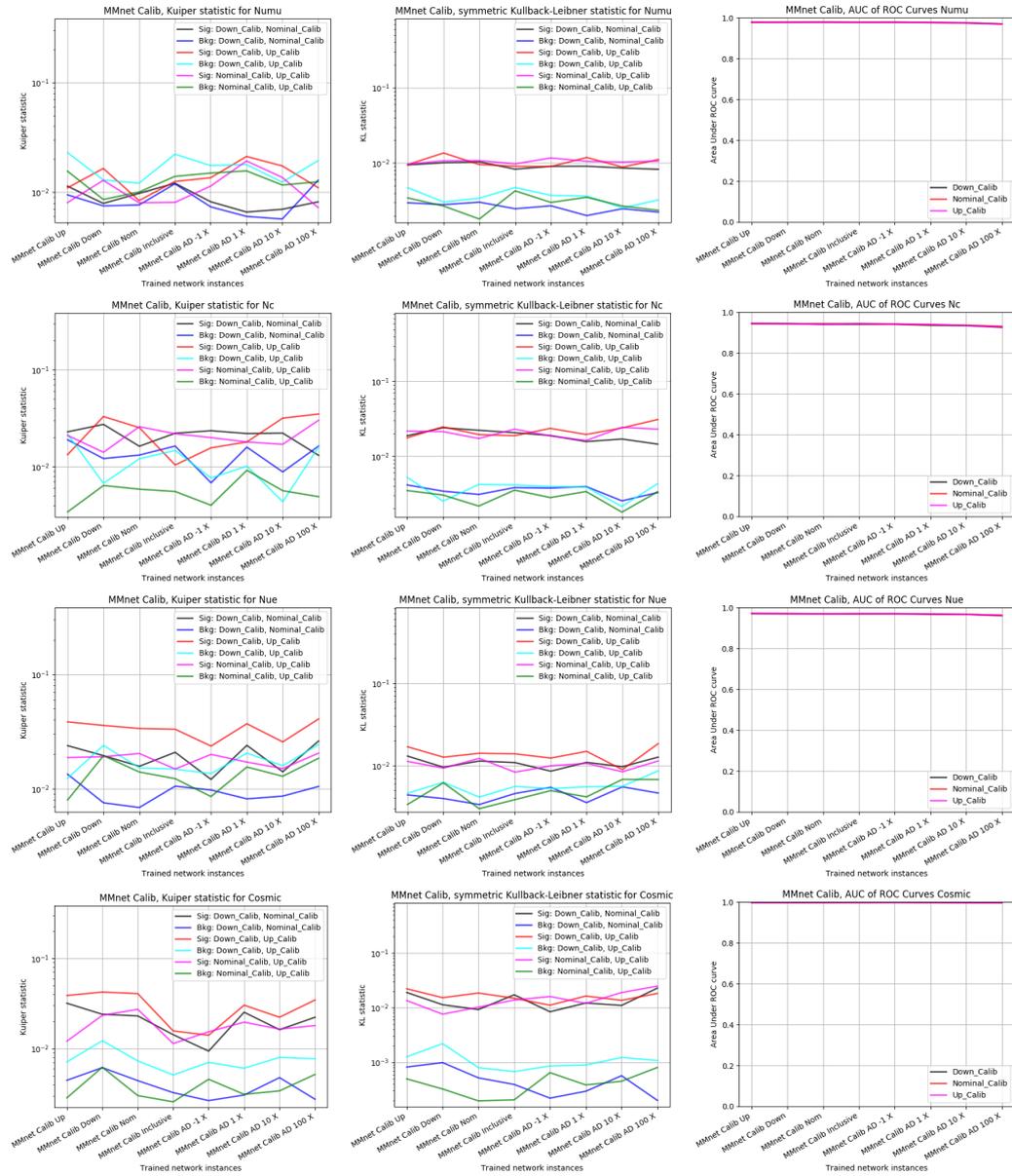


Figure 9.16: Summary statistics of the class PID spectra for the networks trained on the calibration domain shift. The four interaction classes are shown top to bottom (v_μ , NC, v_e and Cosmics) and the Kuiper, Kullback-Leibner and area under the ROC curve are shown left to right. No clear trend for the robustness improvement, as defined through a decrease in the reported Kuiper and Kullback-Leibner statistics, is identifiable.

9.3.4 Layer analysis

With the plotted outputs of the two network branches the presence of the gradient reversal layer is displayed. From the PID spectra and the associated statistics the effects are large for the domain branch while it seems to have a smaller impact on the classification branch. To improve the explainability of the internal gradient reversal effects, and how these effects propagate through the network to form the output an analysis of the internal network layers of a network trained on the light level domains is conducted. Similar to the output nodes which form the final classification layers of the network, each node of an internal layer produces an output value for each input sample that is propagated through the network. A collection of these outputs for a range of input samples can be used to create node output spectra related to the familiar PID spectra. The main difference being that these internal nodes do not represent probability scores and can have different ranges of values due to them not being activated by a softmax activation function. Nevertheless if the network is not robust the variation of the final classification layer output originates from the input samples and is propagated through the network. Likewise for a dual branched gradient reversed network, both the large variations in the domain branch and small variations in the classification branch are traceable through the network. This provides a robustness test of the early shared feature extracting layers, which are expected to be affected by the gradient reversal layer.

Akin to the summary statistic measures for the robustness with the PID spectra, the truth information can be leveraged to subdivide the individual node response distributions by domain. Due to the large number of nodes present in each layer and the high number of layers in the network the layer analysis is limited to seven layers of the networks trained with the gradient reversal layer included on the light level domain shift. It is repeated for the four relative strengths $(-1, 1, 10, 100)$. In Figure 9.17 a schematic representation of the network architecture displays the locations of the selected layers.

The KL, Kuiper and χ^2 statistics are employed to quantitatively express

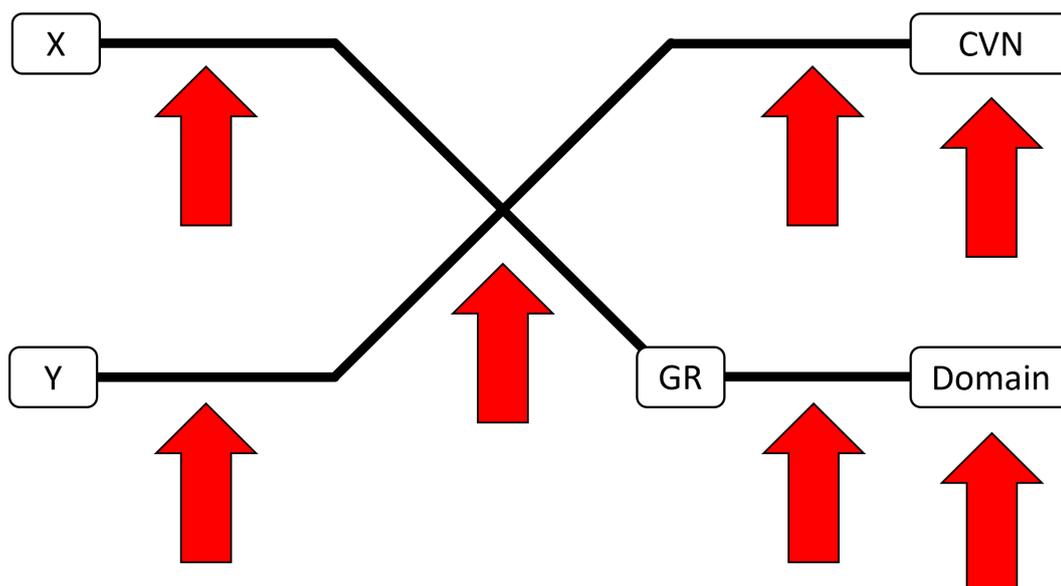


Figure 9.17: Schematic overview of the layer analysis locations in the network architecture.

the variation between distributions for each domain combination. For each node of the selected layers the calculated variations are collected. A representative subset of all these collections is shown in Figure 9.18. Lower values of the statistics indicate a higher degree of similarity between the examined distributions and thus a higher robustness against the evaluated domain variation. Likewise higher values indicate the opposite. The three examined locations show differences in the magnitudes of the statistics. The early shared layer has lower values than those of the later layers in the classification and domain branches. Additionally, the magnitudes vary to a much greater degree as a function of the relative strength for the later layers.

Not all the nodes in a trained network are equally important. From the associated weights and biases of a node in a layer of the network it is not immediately clear how the importance of a specific node should be expressed. It could be the number of activations, the eventual contributions to the output or the average of the input weights. For simplicity in the interpretability of layer analysis results an equal weight is assigned to all the included nodes, except for the nodes without any activations which are excluded. Activation on a subset of the input samples would result in a lower overall contribution of

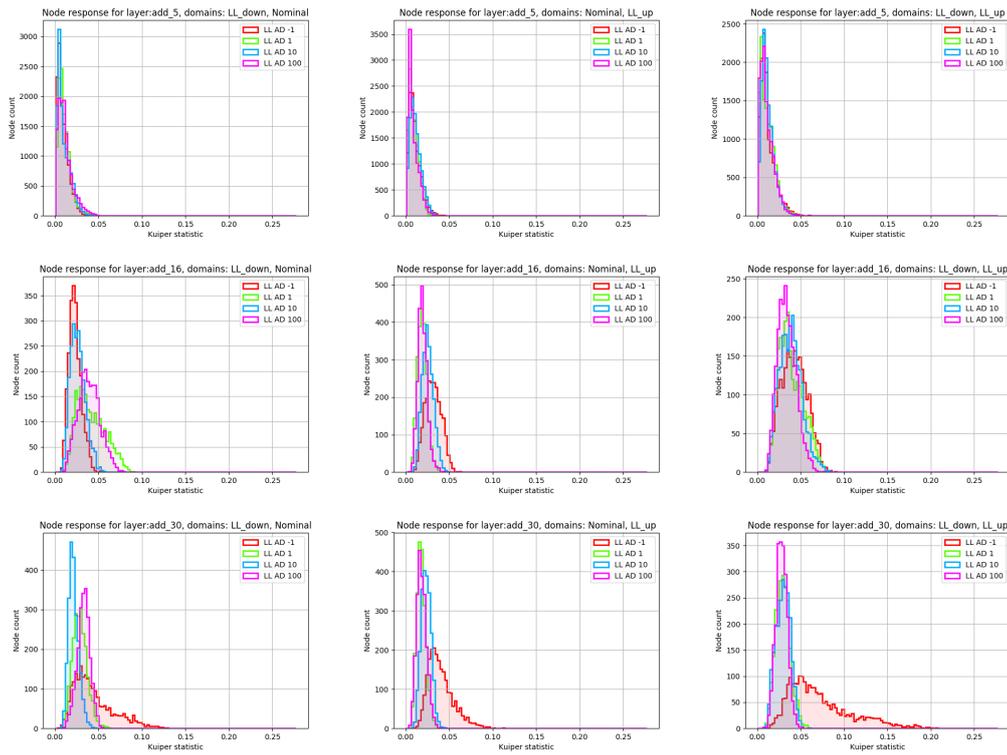


Figure 9.18: Node response distributions for all possible light level domain combinations. Three locations are shown here, halfway in the feature extracting branch of the Y-view(top), halfway down the classification branch(middle) and halfway down the domain classification branch(bottom). Four networks with gradient reversal layers are trained with increasing relative strengths, the differences in the magnitude of statistics indicates a variation in node robustness for different parts of the network.

the node as the normalization of the node output distributions are based on the size of the relevant subset of the input data samples.

Comparisons of the unweighted means of the node distributions statistical measures as calculated over the specific domain combinations can be seen in Figure 9.19. Like the node response distributions shown in Figure 9.18, lower values represent a higher degree of robustness and higher values represent the opposite. The domain output layer displays the largest decrease for increased strength, followed by the middle layer of the domain classification branch. The connecting branch shows only a slight decrease for the increasing strength. The overall magnitude of the statistical measures is lower for this layer than those of the domain branch. The layers in the earlier feature-extracting branches of the

network show no significant variation with increased strength but have lower overall values than the connection layer. The layers in the CVN classification branch show no variation with strength, but have larger values than the early network layers and display a higher spread of values for the various domain combinations.

The comparatively low values of the statistics measuring robustness in the early layers indicates that the shared feature extracting component of the network possesses a high degree of robustness. The lack of variation between the positive and negative relative strengths of the gradient reversal layer of the robustness in these early layers suggest that this robustness is not due to the influence of the inversed gradient components. While this influence is not achieving the full desired effect, it is present in the early layers nonetheless. The updates with the inversed gradients in the early layers directly cause the reduced classification performance in the domain branch, which is demonstrated by the variation going from negative strengths to higher positive strengths. Following the increasing values of the statistical measures along the shared layers into the CVN classification branch exposes increasing exploitation and reliance on domain dependent information. As the early layers are not perfectly robust some degree of domain related information is leaking throughout the network.

9.3.5 Loss penalization with memory

By construction the loss penalization domain adaptation technique affects the network during backpropagation in its entirety. This provides a more uniform approach to the application of a robustness increasing domain adaptation technique, especially when compared against the piecewise approach of the gradient reversal layer. Within the AdCVN framework the custom components required for a comparative study of the CVN robustness and the effects of the loss penalization technique against the domain shifts are included.

To realise the evaluation of the robustness and effectiveness of the loss penalization technique a suite of networks is trained for each of the three domain



Figure 9.19: Unweighted means of the summary statistics for the node response of the domain combinations in the network layers. The seven examined layers are shown top to bottom in order of the network graph labelling, midway of the X-view feature extracting branch, midway of the Y-view branch, the connection point, midway down the classification branch, midway down the domain branch, classification output and domain output. The decrease in the displayed values as a function of relative strength demonstrates the degree to which various parts of the network are affected by the gradient reversal.

shifts. As these trainings are performed with the standard batchsize of 128 data samples per batch, the memory inclusive version of the loss penalization implementation is selected for usage. The memory structure has a depth of 5 batches for these trainings. Initial development and validation trainings performed prior to the full scale study revealed that the majority of the network learning occurs in the first 30 epochs. During these early epochs the network performance notably improves. Throughout subsequent epochs the additional improvements decrease in magnitude and the network plateaus. Figure 9.20 shows several of the training curves of the development and validation test trainings on which the location of this threshold is based.

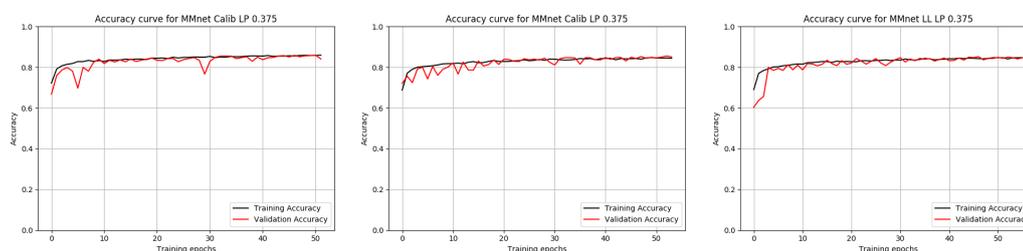


Figure 9.20: Subset of the training accuracy curves for memory enhanced loss penalized network trainings trained with a single relative strength. Here trainings on the calibration(left, middle) and light level(right) shifts are shown with strength 0.0 (left) and 0.375 (middle, right).

The plateauing of learning during the mature stages of training can be exploited to create a training schedule which requires fewer computational resources to provide an equivalent level of comparative power. Employing the callback functionality for the purposes of hyperparameter scheduling allows for the periodic variation of the relative strength included in the loss penalization calculation. This provides a method for scanning a range of values during a single training session. Inclusion of this scheduling callback function has been implemented and validated for both the gradient reversal and loss penalization domain adaptation techniques. For both cases during training a small change to the coupling strength resulted in faster convergence of the network performance to that of a network trained purely on the new value. Typically this convergence occurs within 3 – 5 epochs consisting of 300 batches. Changes for larger

values converge slower due to the general instability of the trainings for large strength values. As the network is effectively pretrained prior to the change in strength, only increased influence of the domain adaptation technique needs to be incorporated into the network before convergence is completed.

As the threshold is well beyond the typical convergence period, the network will reach a plateaued state well before the next scheduled change of the relative strength. At the end of these periodic plateaus the network state is saved for subsequent evaluation as the network state at those points of the training is equivalent to that of a network trained on the same strength without the scheduler.

To reach the threshold in a lower number of epochs the number of batches comprising an epoch are increased from three hundred to a thousand. This sets the new values for the threshold where the relative strength increases to a period of 10 epochs. To effectively scan a range of values the following recursive relation is employed to calculate the relative strength values,

$$\lambda_n = \frac{3}{2} \cdot \left(\lambda_{n-1} + \frac{1}{4} \right) \quad (9.1)$$

Application of equation (9.1) results in the relative strength values as a function of epoch number which are shown in table 9.2.

Employing this hyperparameter scheduling, extended trainings are completed for the network configurations with the loss penalization and memory structures included. This is done for each of the examined domain shifts. In Figure 9.21 the training curves are shown. As the reported loss value incorporates the contribution of the loss penalization term it is not obvious how the classification loss is changing as the training progresses. Therefore during training both the total loss and contribution of the classification loss are reported. The classification loss is the categorical cross entropy. The total loss is the sum of the classification loss and the loss penalization factor multiplied by the relative strength λ . The stepwise nature of the total loss curves is expected due to the periodic increase of the relative strength. With higher strengths the

Epochs	Relative strength (λ)
0	0.000
10	0.375
20	0.938
30	1.781
40	3.047
50	4.945
60	7.930
70	12.06
80	18.47
90	28.08
100	64.12
110	96.56
120	145.2
130	327.7
140	491.9
150	738.2

Table 9.2: Relative strengths as a function of epoch number. The range of values for the relative strengths of the loss penalization method with memory included scanned is shown.

classification loss is increasing and losing stability. While not reported during training, with both of these loss values and the current value of the relative weight the value of the unweighted loss penalization factor can be calculated. Reduction over the training time of this unweighted loss penalization factor is the result of the network prediction becoming more similar across the various domain comprising the domain shift. As the classification performance degrades while the robustness increases an optimal strength can be identified. The exact placement of this relative strength value depends on the value of performance to robustness in the trade-off. Even though the loss curves display a larger than normal degree of instability this is not reflected in the accompanying accuracy curves except for the higher strength values at which point the classification performance is known to be affected. This additional instability is due to the periodic changes of the hyperparameters during training and the additional fluctuations from the loss penalization factor calculation. The first of these should decrease quickly after the change as the convergence interval has been

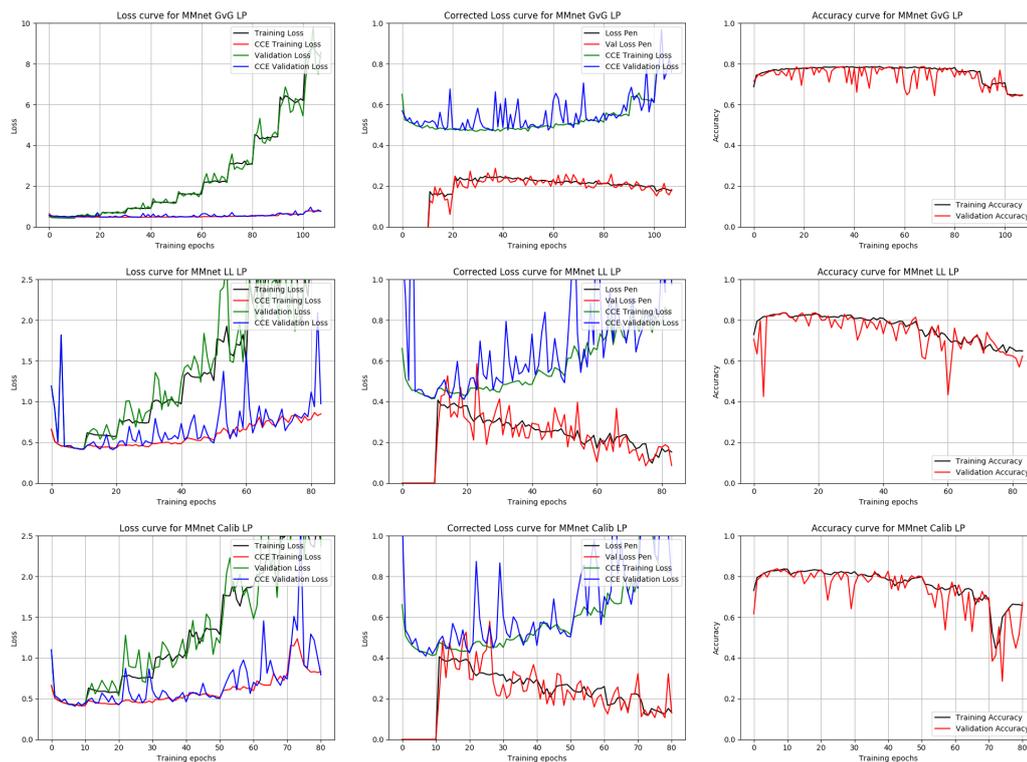


Figure 9.21: Training curves for memory enhanced loss penalized network trainings with relative strength scheduling according to equation (9.1) and table 9.2. The GENIE vs GiBUU(top), light level(middle) and calibration(bottom) domain shifts are shown. The total and classification losses are shown on the left, the individual contributions of the loss penalization factor and the classification loss to the total loss are shown in the middle and the accuracy curves are shown on the right. With increasing relative strengths both the classification performance and variation across domain shifts decrease, thus confirming an expected trade-off between network performance and robustness.

found to be much shorter than the hyperparameter variation period. The second is inherent to the calculation due to the statistical fluctuations present in it. The contributions of these fluctuations are expected to be larger with higher relative strength values as the magnitude of the fluctuations are multiplied by the relative strength prior to addition into the total loss.

The networks trained with the inclusion of the loss penalization technique and its memory components can be used in a comparison with previous network configurations trained in equivalent environments. With the periodic variation of the relative strength during the training of the loss penalization networks,

each converged network state prior to each increase of the relative strength is representative of a network trained for a longer time with that same strength.

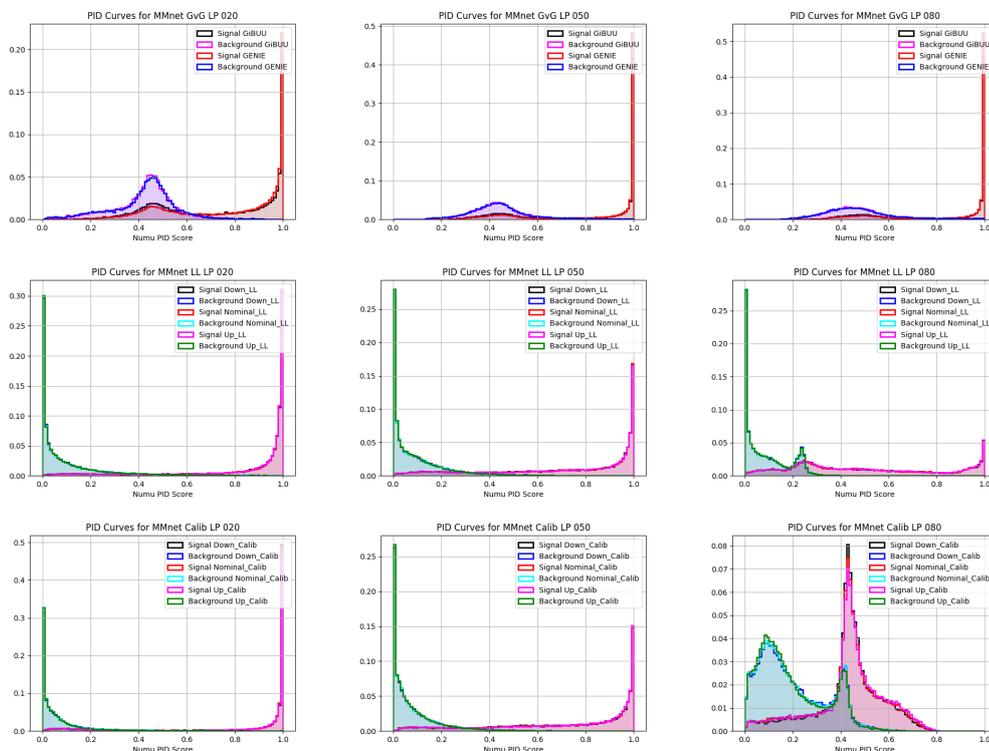


Figure 9.22: Subset of the PID spectra for the memory enhanced loss penalized network trainings with relative strength scheduling. The evaluation is repeatedly applied at the end of the training period for a given relative strength (0.94, 4.95, 18.5). The PID spectra for the v_{μ} class are shown. The reduction in separability between the signal and background distributions for increased strengths matches the decrease of performance visible in Figure 9.21. The variations across the domain shifts are relatively small and thus it is difficult to observe changes for increased strengths from the PID spectra.

Repeated evaluations for each of these saved network states are therefore required. For comparative purposes repeated evaluations are applied to the non loss penalized networks as well. In Figures 9.22, 9.23 and 9.24 a representative subset of the total evaluations performed on the suite of trained networks is shown. Both the PID spectra, the ROC and PR curves exhibit variations which increase with increasing relative strengths. The decrease in classification performance becomes clear in both the ROC and PR curves as these gradually move towards their respective random guessing thresholds. For the ROC curves

this results in movement towards the diagonal from (0, 0) to (1, 1) and for the PR curves a horizontal line from (0, 0.5) to (1, 0.5). This matches the decrease in classification performance identified in the loss and accuracy trainings curves of Figure 9.21.

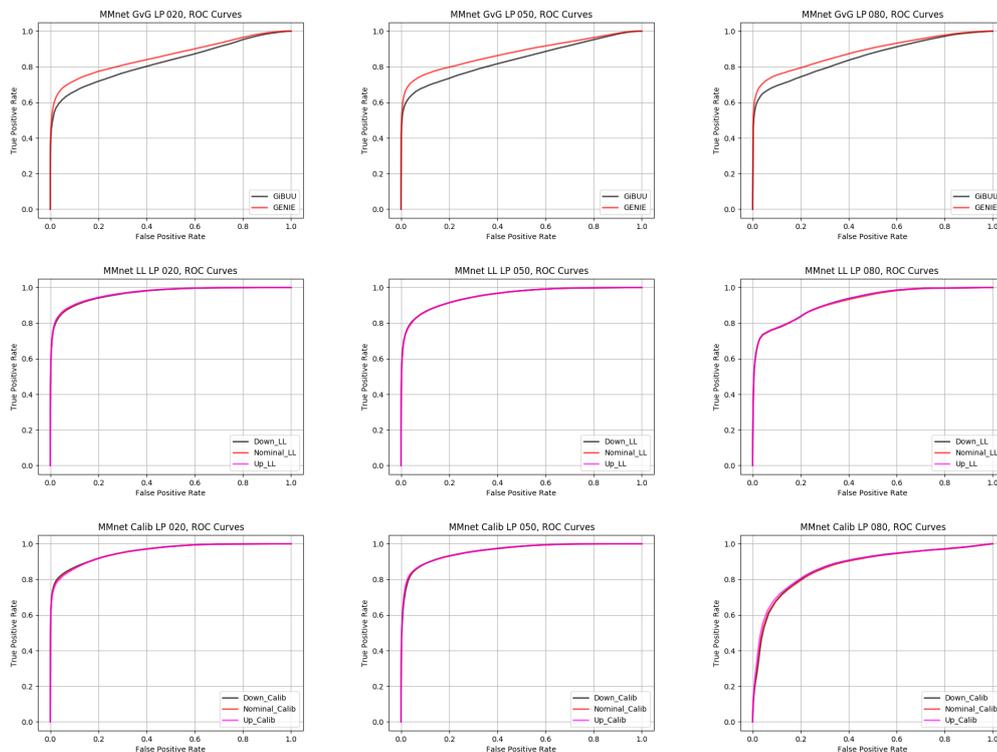


Figure 9.23: Subset of the ROC curves for the memory enhanced loss penalized network trainings with relative strength scheduling. The evaluation is repeatedly applied at the end of the training period for a given relative strength (0.94, 4.95, 18.5). The ROC curves for the v_{μ} class are shown.

The variations in the robustness are visible in the small variations between the domains in either the PID, ROC or PR curves for the light level and calibration domain shifts. As the magnitude of the domain variations are small it is difficult to identify a clear trend of decreasing variations corresponding to increasing robustness from these curves as suggested by Figure 9.21.

The ROC and PR curves for the GENIE vs GiBUU domain shift show a large difference in performance between the GENIE and GiBUU domains. The difference is present in the non loss penalization trained networks as well, which can be seen in Figure 9.25. Notably the magnitude of the difference

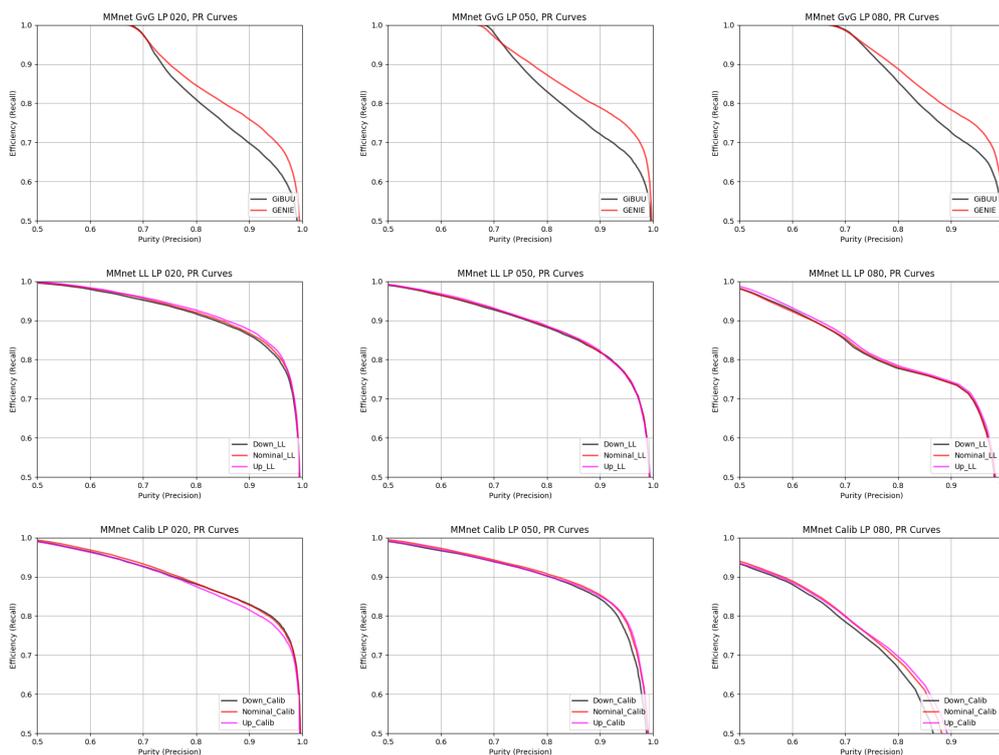


Figure 9.24: Subset of the PR curves for the memory enhanced loss penalized network trainings with relative strength scheduling. The evaluation is repeatedly applied at the end of the training period for a given relative strength (0.94, 4.95, 18.5). The PR curves for the ν_μ class are shown.

varies for the different trainings, it is largest for the network trained solely on the GENIE domain and smallest for the network trained solely on the GiBUU domain. The expected results would be for the single domain trained networks to excel on the domain closest to their training data. The observed and expected performance are consistent in the presence of separate reason resulting in a baseline performance difference between the domains. As the robustness evaluations are comparative studies, a constant offset of performance between the domains applies equally to all trainings and does not affect the individual variations between the trainings. Correcting for this inherent performance difference the behaviour of loss penalization networks trained on the GENIE vs GiBUU domain shift is consistent with that of the light level and calibration domain shifts.

To examine the robustness variation more closely the summary statistics

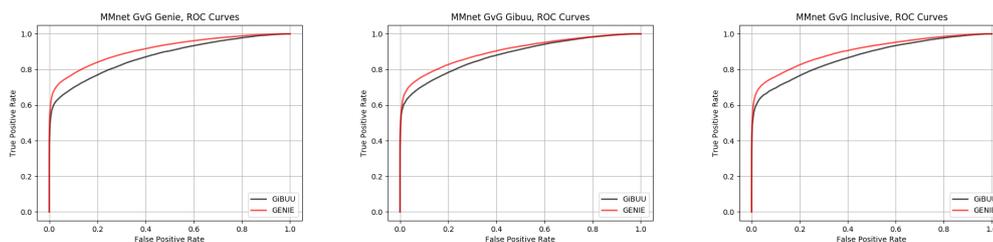


Figure 9.25: ROC curves for the v_μ class of the networks trained without loss penalization or GR on solely the GENIE domain (left), solely the GiBUU domain (middle) or on both (right).

across the domain shift are produced for the suite of the networks. These are shown in figures 9.26, 9.27 and 9.28 for the GENIE vs GiBUU, light level and calibration domain shifts respectively. The network response differences between all possible domain combinations are repeatedly evaluated with multiple metrics. For the networks trained without loss penalization this provides a measure for the degree of variation in the network performance across the domains at various points of the training. While a persistent decrease or increase is visible for a single class, network or metric no clear trend across multiple classes and metrics emerges, even for single networks. An example of such an increase can be seen in the Kuiper statistic evaluation of the non loss penalization network trained only on the GENIE domain, as for this metric the background distributions appear to diverge. This behaviour is not as well reflected in the other statistics for this case. The Kuiper statistic is more sensitive to the specific diverging variations for this case.

For the trainings which include evaluations with three domains, three distinct combinations between the domains are possible. The magnitude of the domain shift for each of these combinations is not equivalent. For the light level and calibration domain shifts the magnitude of the up and down shifts are equivalent. The combinations (downshifted, nominal) and (nominal, upshifted) therefore have an equivalently sized domain shift. The combination of (downshifted, upshifted) then has a domain shift effectively twice the size of the previously mentioned combinations. As the robustness evaluation relies on some measure of distance or similarity between the PID spectra of the

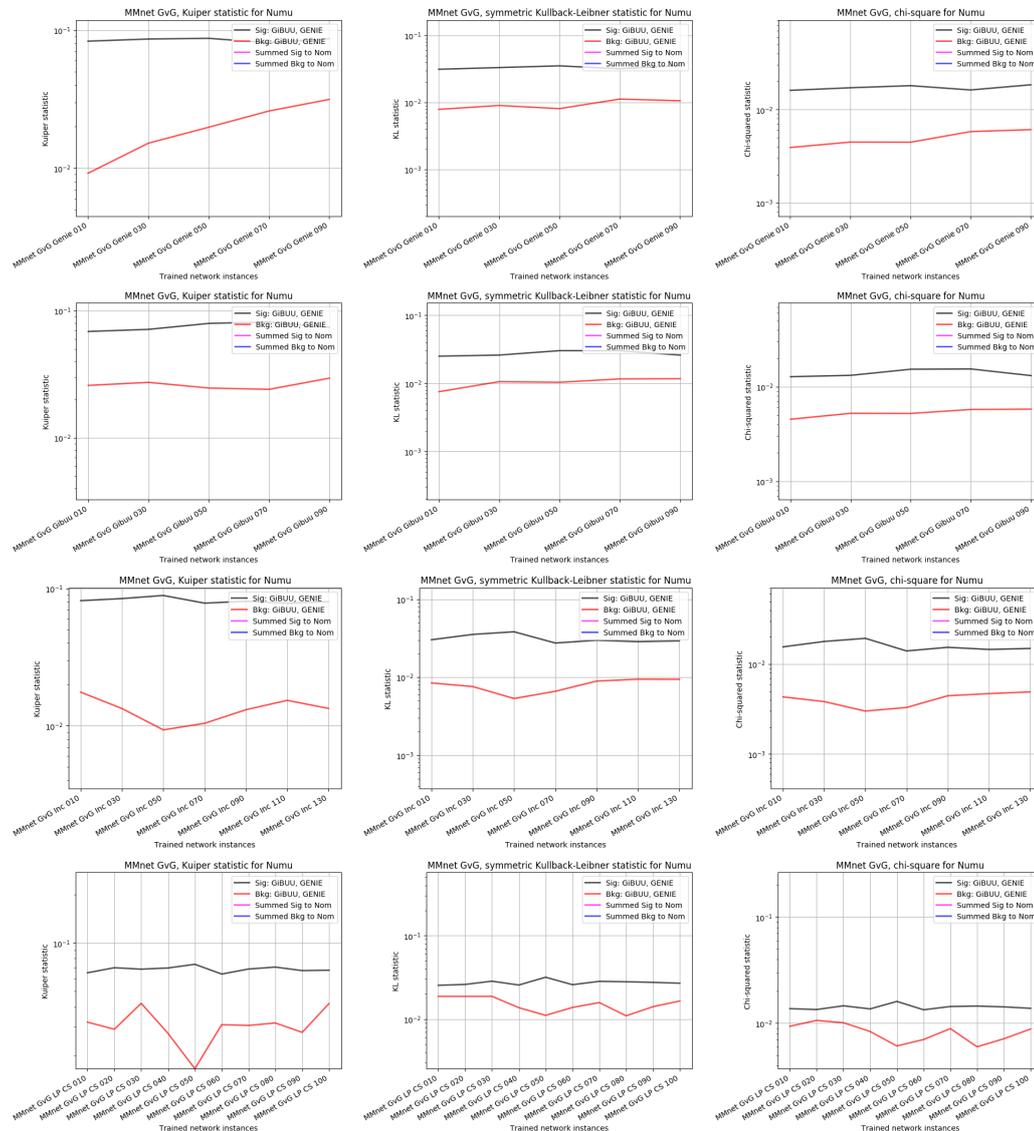


Figure 9.26: Subset of the summary statistics for the memory enhanced loss penalized network trainings with relative strength scheduling trained on the GENIE vs GiBUU domain shift. All the networks trained in the suite are included in the evaluation. In descending order, the single domain trainings without loss penalization, the multiple domain training without loss penalization and the memory inclusive loss penalization training with increasing relative strength. Only the summary statistics for the v_{μ} class are shown while all classes have been evaluated. No clear trends, such as the expected decrease of the reported values with increasing strengths which would demonstrate increasing robustness, can be identified.

network responses on different domains, this relative difference between the size of the examined domain shift for the various domain combinations could result in larger values of the summary statistics for the combination with the larger domain shift. The sum of the signal (background) distributions of the smaller domain shifts is added separately which produces an effective domain shift comparable to the combination with the larger domain shift. This enables a fair comparison between the summed and the (downshifted, upshifted) combinations. For both the light level and calibration domain shifts the two combinations with a larger domain shift have consistently larger values of the summary statistics, which corresponds with larger separation and dissimilarity in the underlying PID distributions of the network response on the different domains. The summation of the two domain combinations with smaller domain shifts is consistently producing significantly larger values than the single domain combination with the larger domain shift.

The summary statistics for the networks trained with the memory inclusive loss penalization exhibit no clear increase in robustness with increasing values of the relative strengths. This is visible in Figures 9.26, 9.27 and 9.28 as the summary statistics do not consistently decrease for higher relative strength values. The strength values are increased with the increasing number of training epochs. Even though a clear increase in robustness is not visible, from Figure 9.21 the classification accuracy decreases significantly with higher relative strengths. The memory inclusive loss penalization training is therefore capable of maintaining a consistent level of robustness for decreasing levels of network performance.

The magnitude of the values and the size of their fluctuations are similar to those present in the non loss penalization trained networks. While only a subset of the total available graphs are shown in Figures 9.26, 9.27 and 9.28 the other classes; NC, ν_e and cosmics exhibit the same behaviour. The overall magnitude of the statistical metrics coupled with the similarity of both the previously examined PID spectra, ROC and PR curves for the domains contained in

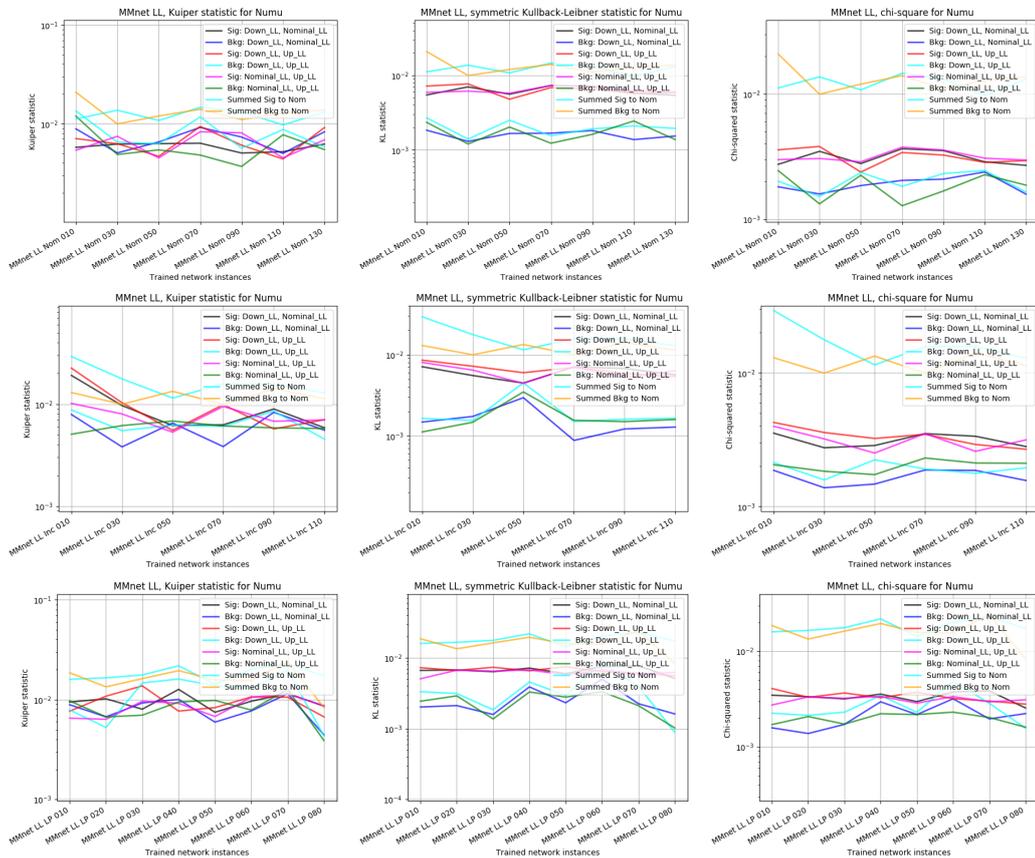


Figure 9.27: Subset of the summary statistics for the memory enhanced loss penalized network trainings with relative strength scheduling trained on the light level domain shift. All the networks trained in the suite are included in the evaluation. In descending order, the single domain training without loss penalization, the multiple domain training without loss penalization and the memory inclusive loss penalization training with increasing relative strength. Only the summary statistics for the ν_μ class are shown while all classes have been evaluated. No clear trends, such as the expected decrease of the reported values with increasing strengths which would demonstrate increasing robustness, can be identified.

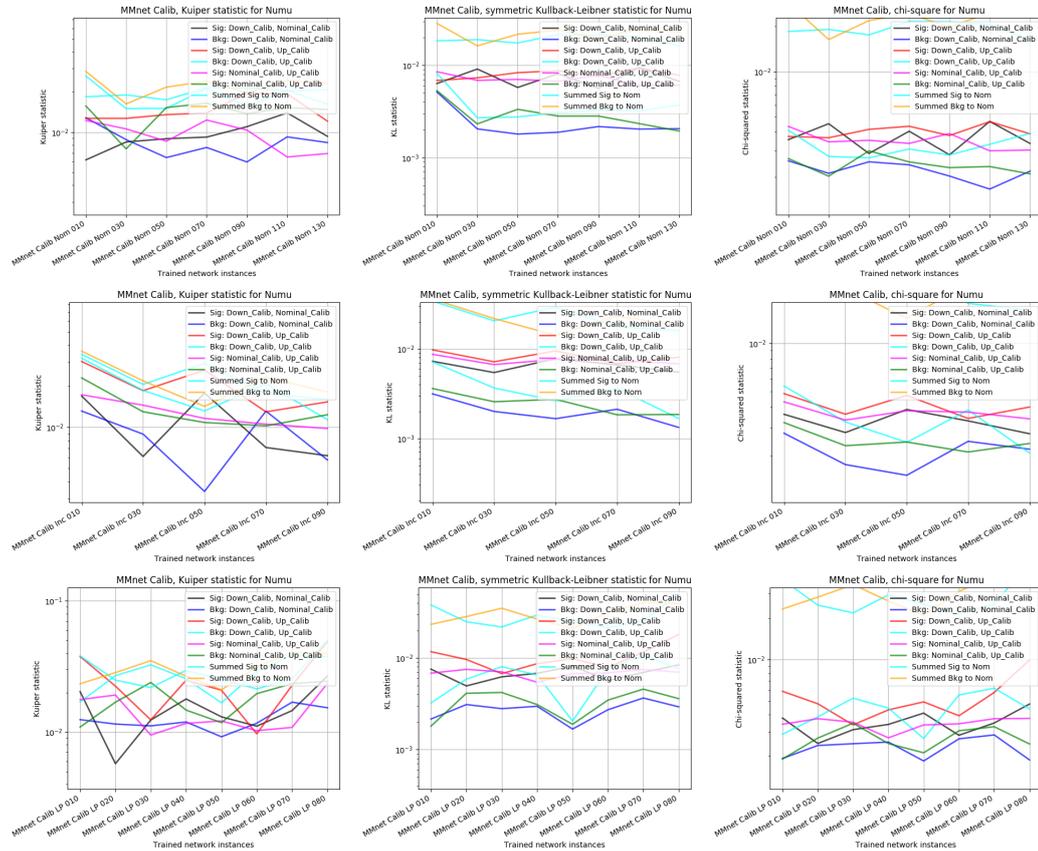


Figure 9.28: Subset of the summary statistics for the memory enhanced loss penalized network trainings with relative strength scheduling trained on the calibration domain shift. All the networks trained in the suite are included in the evaluation. In descending order, the single domain training without loss penalization, the multiple domain training without loss penalization and the memory inclusive loss penalization training with increasing relative strength. Only the summary statistics for the v_μ class are shown while all classes have been evaluated. No clear trends, such as the expected decrease of the reported values with increasing strengths which would demonstrate increasing robustness, can be identified.

the domain shifts would suggest that both the non loss penalization and loss penalization trained networks have attained a high level of robustness.

As was the case for the gradient reversal domain adaptation technique, no distinct improvement to the robustness as a result of the application of the domain adaptation techniques can be found from the examination of the summary statistics. A unambiguous decrease of the reported values as a function of increased strength could not be identified, even though the training curves displayed indications for the possibility of this trend to be present.

9.4 Discussion and future work

Evaluation of the CVN network robustness against the three domain shifts has demonstrated a degree of performance which is similar across the domains comprising the shifts. This is indicative CVN network being highly robust against the domain shifts. The single domain trainings demonstrated that this network is capable of extracting and employing domain specific information from the input samples as it achieved performances above the random guessing threshold for domain identification. This allows the possibility for the domain adaptation techniques to reduce this domain dependence and therefore further increase the robustness. Application of these techniques demonstrated a reduction in domain classification performance but no clear improvement beyond the initial robustness of the classification performance.

The template for the purpose of comparative robustness studies of neural networks presented in this work contains opportunities for both the refinement of the employed methods, the environments in which it is applied and adaptation for future applications.

One such opportunity is the application of the robustness study on data produced in the reversed horn current as opposed to the forward horn current. This effectively exchanges the neutrino beam to an antineutrino beam. Some of the interactions in matter for an antineutrino are different to those of a neutrino. This results in a subset of the total pixelmaps contained in the RHC

dataset to be different from those of the FHC dataset. The neural networks trained on either or both sets of data can vary in the levels of performance, so repeated validation studies are required to confirm optimal performance and whether a single or dual network approach is optimal. The robustness of the individual FHC or RHC networks could be evaluated separately. Alternatively a successful domain adaptation method could be leveraged to ensure equivalent performance of the networks across a change of horn current mode.

Another opportunity can be found in the application of the proposed robustness evaluation techniques on other network trainings in the NOvA experiment, such as the prong CVN's or LSTM energy estimators. With the planned inclusion of further neural networks in future NOvA analyses, the opportunities and need for both the evaluation techniques and domain adaptation methods are expected to increase significantly.

The network performance on the GENIE vs GiBUU domain shift is consistently higher for evaluation on the GENIE domain than the GiBUU domain, even for the network trained exclusively on the GiBUU domain. The cause of this constant offset has been investigated but not positively identified. The extensive validation undertaken by the producers of the simulated data from which the GENIE vs GiBUU domain shift dataset has been compiled does not reveal any potential causes due to neutrino event generators themselves. The inclusion of a subset of GENIE events in the GiBUU domain or the implementation of the eventweighting as a simple weighting in the loss function could potentially be a part of the root cause resulting in the performance difference. As the difference is correctable in the robustness evaluations, a positive identification of the cause(s) is not required for the performance of the robustness studies. Further examinations building upon the previous validation studies and these investigations could result in identification of the root cause.

Within the operations of the CVN network in the NOvA experiment, it is exposed to potential domain shifts of a limited magnitude. As seen with the evaluations of the gradient reversal and loss penalisation domain adaptational

techniques, this can result in limited outcomes if the network is already initially robust against the domain shift. With the proof of concept achieved on the various toy scenarios it is reasonable to expect stronger results on domain shifts with a greater impact on the network performance. This is the case for neural network application in the wider field beyond high energy physics as well as for some HEP experiments. In experiments such as DUNE, where the employed detector technologies can vary between detectors, the deployment of domain adaptational techniques which can ensure an equivalent network performance across these more extreme domain shifts will be invaluable.

Chapter 10

Conclusions

This work has shown that the deep convolutional neural networks which are deployed in the NOvA neutrino oscillation analyses are robust against multiple domain shifts. These consisted of an exchange of the GENIE to the GiBUU neutrino event generator and the light level and calibration systematic uncertainties. It is shown that the Event CVN is capable of learning domain dependent information for the purposes of domain classification. The variations in the neutrino interaction classification performance of the Event CVN as evaluated in the AdCVN framework across the domains constituting the domain shifts were found to be small relative to the magnitude of the performance. Multiple domain adaptation methods are validated and incorporated into the AdCVN framework with novel adaptations, such as memory structures, which enable application on the NOvA CVN training environment. Application of these techniques on the NOvA CVN trainings demonstrated a reduction in domain classification performance but no clear improvement beyond the initial robustness of the classification performance.

Appendix A

W boson jet binary classification

The adversarial domain adaptation technique of section 6.6.1 has been more widely applied beyond the binary classification toy example. The W boson jet discrimination problem in the presence of pile-up is one such case. It is a binary classification problem for the discrimination of W boson jets in the Large Hadron Collider (LHC) from similar background jets produced from non W boson particle collisions and decays. The generated data for this problem is publicly available [157] and contains multiple domains. The difference by which the various domains are identified is the presence of pile-up. Pile-up refers to the scenario where multiple proton-proton collisions occur within the data recording window, where one of these interactions could still produce the jets of interest. The number of overlapping interactions varies per data sample, but the average of the samples with pile-up is set at 50 interactions. The dataset consists of a collection of reconstructed variables describing the interactions, effectively forming a feature vector. Further information on the simulation chain and parameter choices which produced the dataset as well as the publicly available data can be found here [190].

In this scenario the training and evaluation of the discriminator, or classifier, is not as straightforward as for typical neural network applications. The primary goal is the optimization of the discovery significance of the W boson jets. This does not necessarily correlate directly with optimal network performance. For example the contribution towards discovery significance of fewer events classified

by a classifier with low false positive rate could be higher than that of a network which selects more events with a slightly worse false positive rate. When the network performance would be examined purely through accuracy or some FOM definition it is reasonable that the latter would result in a higher performance rating, even though the significance might be lower.

The classification problem between characteristic and W boson jets can be reframed into a statistical test for the discovery of W bosons from the jets. The discovery significance can then be defined as the null hypothesis rejection significance. The null hypothesis is the scenario where no W boson jets are present in the dataset and the alternative is the case where some W boson jets are present within a large sample of non W boson jets. The Approximate Mean Significance (AMS) can be used as an objective function for the purpose of determining if the number selected signal events exceed the threshold set for the hypothesis test. If so, this would lead to an rejection of the null hypothesis. The AMS has been used for a related task of Higgs boson discovery in a similar problem environment [191]. The AMS is defined in [191] equation (20) as,

$$AMS = \sqrt{2 \left((s+b) \ln \left(\frac{s+b}{b_0} \right) - (s+b-b_0) \right) + \frac{(b-b_0)^2}{\sigma_b^2}} \quad (\text{A.1})$$

where s is the sum of the weighted selected signal events(TP), b is the sum of the weighted selected background events(FP) and b_0 are given by,

$$b_0 = \frac{1}{2} (b - \sigma_b^2 + \sqrt{(b - \sigma_b^2)^2 + 4(s+b)\sigma_b^2}) \quad (\text{A.2})$$

As the AMS relates directly to the discovery significance it provides a metric which incorporates both the classifier's general performance and its robustness against pile-up. If the relative strength λ between the objectives of the classifier and adversary is varied, the trade-off between classification power and domain robustness is gradually shifted. Lower values reduce the influence of the adversary resulting in higher classification performance but lower robustness whereas higher values will produce the opposite. The AMS is

dependent on both of these and evaluation of it yields the optimum point of the discovery significance for the trade-off.

As the classification task is more complex than the binary classification toy scenario, the network architecture of the discriminator and adversary are updated to reflect the new complexity. The discriminator is fully connected with five layers, the first being the input layer, the next three consist of 64 nodes each and are activated with tanh, relu and relu respectively. For the adversary the architecture consists of five layers as well, first the input layer, the next three with 64 nodes and relu activated. The loss function are binary and categorical cross entropy respectively. With dual domains the categorical cross entropy could be swapped for binary cross entropy. The training dataset consists of 150000 data samples while the validation dataset consists of 5000000 data samples. Both of the datasets are reweighted such that the background of non W boson jet events is at least ten times more numerous than the signal events. This is required as the original simulation did not necessarily have interaction rates which were skewed to the required degree. Following the domain adversarial training procedure, the discriminator is pretrained for 10 epochs, the adversary for 10 epochs and the adversarial training is run for 40 epochs. This training is repeated for various values of the relative strength.

A comparison between the calculated AMS for the various λ values can be seen in Figure A.1. The average AMS value is reported for 5 repeated trainings per relative strength. Similarly to some of the previously described statistical evaluation tools the location of a cut, or threshold, for the selection of classified events is varied for the calculation of the AMS. The AMS is evaluated for multiple values of relative strengths and is found to be maximized for a relative strength of 10. This validates results presented in the wider literature [157], and exhibits the potential of increasing confidence in the network by reducing performance and increasing robustness. A domain adaptation technique is successfully deployed and results in increased robustness against the negative influence of pile-up.

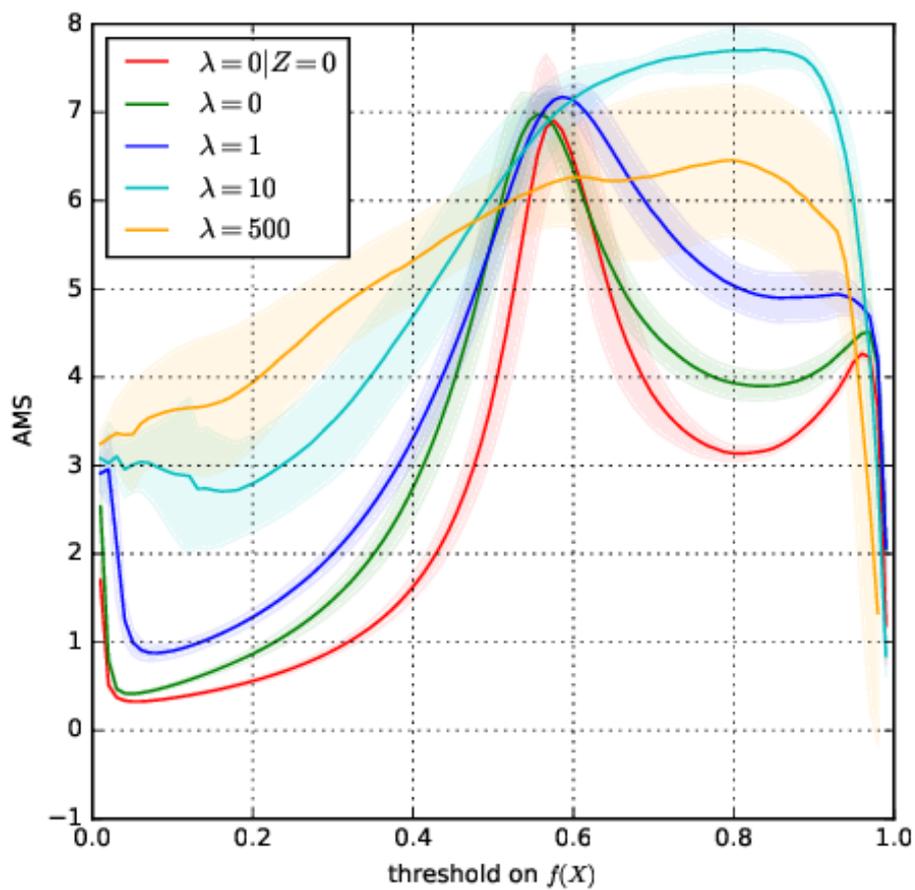


Figure A.1: The approximate median significance as a function of trainings with various relative strengths. The discovery significance, as represented by the AMS, is maximized for domain adversarial training with a relative strength of 10. Fewer locations of the threshold (100) in the AMS calculations are used than in [157], this yields a significant increase to the completion time of the AMS calculations at the cost of smoothness in the AMS curves.

Bibliography

- [1] J. Chadwick. Intensitätsverteilung im magnetischen Spektrum der β -Strahlen von Radium B+C. Druck von Friedr. Vieweg und Sohn, 1914.
- [2] HPaul. Beta spectrum of rae (radium e or ^{210}bi), Apr 2015.
- [3] W. Pauli. "Offener Brief an die Gruppe der Radioaktiven bei der Gauvereins-Tagung zu Tübingen". CERN Scientific Information Service (archive), 1930.
- [4] C.L. Cowan, F. Reines, F.B. Harrison, H.W. Kruse, and A.D. McGuire. Detection of the free neutrino: a confirmation. *Science*, 124(3212):103–104, 1956.
- [5] G. Danby, J.M. Gaillard, K. Goulios, L.M. Lederman, N. Mistry, M. Schwartz, and J. Steinberger. Observation of high-energy neutrino reactions and the existence of two kinds of neutrinos. *Phys. Rev. Lett.*, 9:36–44, Jul 1962.
- [6] K. Kodama et al. Observation of tau neutrino interactions. *Physics Letters B*, 504(3):218–224, Apr 2001.
- [7] S.M. Bilenky and J. Hošek. Glashow-weinberg-salam theory of electroweak interactions and the neutral currents. *Physics Reports*, 90(2):73–157, 1982.
- [8] E. Noether. Invariante variationsprobleme. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1918:235–257, 1918.

- [9] Fermilab Office of Sciences PBS Nova. Standard model of elementary particles, Sep 2019.
- [10] T.D. Lee and C.N. Yang. Question of parity conservation in weak interactions. *Phys. Rev.*, 104:254–258, Oct 1956.
- [11] C.S. Wu, E. Ambler, R.W. Hayward, D.D. Hoppes, and R.P. Hudson. Experimental test of parity conservation in beta decay. *Phys. Rev.*, 105:1413–1415, Feb 1957.
- [12] J.A. Formaggio and G.P. Zeller. From νe to $\bar{\nu} e \nu$: Neutrino cross sections across energy scales. *Rev. Mod. Phys.*, 84:1307–1341, Sep 2012.
- [13] T. Blank and W. Hollik. Precision observables in $\text{su}(2) \times \text{u}(1)$ models with an additional higgs triplet. *Nuclear Physics B*, 514(1–2):113–134, Mar 1998.
- [14] E. Majorana and L. Maiani. A symmetric theory of electrons and positrons. Springer, Berlin, Heidelberg, 2006, org.1937.
- [15] N. Abgrall et al. The large enriched germanium experiment for neutrinoless double beta decay (legend). *AIP Conf. Proc.*, 1894(1):020027, 2017.
- [16] C.E. Aalseth et al. Search for neutrinoless double- β decay in ^{76}Ge with the majorana demonstrator. *Phys. Rev. Lett.*, 120(13):132502, 2018.
- [17] N. Johnston et al. *Advanced Linear and Matrix Algebra*. Springer, 2021.
- [18] M. Aker and the KATRIN Collaboration. Direct neutrino-mass measurement with sub-electronvolt sensitivity. *Nature Physics*, 18(2):160–166, Feb 2022.
- [19] B. Pontecorvo. Inverse beta processes and nonconservation of lepton charge. *Zh. Eksp. Teor. Fiz.*, 34:247, 1957.

- [20] H. Nunokawa, S.J. Parke, and J.W.F. Valle. Cp violation and neutrino oscillations. *Prog. Part. Nucl. Phys.*, 60:338–402, 2008.
- [21] A.Y. Smirnov. The msw effect and matter effects in neutrino oscillations. *Physica Scripta*, T121:57–64, Jan 2005.
- [22] D.S. Ayres et al. The nova technical design report. FERMILAB-DESIGN-2007-01, 10 2007.
- [23] JUNO Collaboration. Neutrino mass hierarchies, Apr 2018.
- [24] J.N. Bahcall, A.M. Serenelli, and S. Basu. New solar opacities, abundances, helioseismology, and neutrino fluxes. *The Astrophysical Journal*, 621(1):L85–L88, jan 2005.
- [25] D. Raymond, D.S. Harmer, and K.C. Hoffman. Search for neutrinos from the sun. *Phys. Rev. Lett.*, 20:1205–1209, May 1968.
- [26] B.T. Cleveland, T. Daily, R.Jr. Davis, J.R. Distel, K. Lande, C.K. Lee, P.S. Wildenhain, and J. Ullman. Measurement of the solar electron neutrino flux with the homestake chlorine detector. *The Astrophysical Journal*, 496(1):505–526, mar 1998.
- [27] C. Pena-Garay and A. Serenelli. Solar neutrinos and the solar composition problem. *arXiv*, 11 2008.
- [28] J.N. Abdurashitov et al. Measurement of the solar neutrino capture rate with gallium metal. iii. results for the 2002–2007 data-taking period. *Phys. Rev. C*, 80:015807, Jul 2009.
- [29] M. Altmann et al. Complete results for five years of gno solar neutrino observations. *Phys. Lett. B*, 616:174–190, 2000.
- [30] K. Abe et al. Solar neutrino measurements in super-kamiokande-iv. *Phys. Rev. D*, 94(5):052010, 2016.

- [31] N. Vinyoles, A.M. Serenelli, F.L. Villante, S. Basu, J. Bergström, M.C. Gonzalez-Garcia, M. Maltoni, C. Peña Garay, and N. Song. A new generation of standard solar models. *Astrophys. J.*, 835(2):202, 2017.
- [32] B. Aharmim et al. Combined analysis of all three phases of solar neutrino data from the sudbury neutrino observatory. *Phys. Rev. C*, 88:025501, Aug 2013.
- [33] M. Agostini et al. Correlated and integrated directionality for sub-mev solar neutrinos in borexino. *Phys. Rev. D*, 105:052002, Mar 2022.
- [34] K. Kamdin. Understanding the sno+ detector, 12 2015.
- [35] M. Agostini et al. Comprehensive measurement of pp-chain solar neutrinos, 10 2018.
- [36] T. Kajita, E. Kearns, and M. Shiozawa. Establishing atmospheric neutrino oscillations with super-kamiokande, 04 2016.
- [37] P.A. Zyla et al. Review of particle physics/neutrino masses, mixing and oscillations. *PTEP*, 2020(8):083C01, 2020, 2021.
- [38] T. Kajita. Atmospheric neutrinos and discovery of neutrino oscillations, 2010.
- [39] G.D. Barr, T.K. Gaisser, S. Robbins, and T. Stanev. Uncertainties in atmospheric neutrino fluxes. *Phys. Rev. D*, 74:094009, 2006.
- [40] F. Reines, M.F. Crouch, T.L. Jenkins, W.R. Kropp, H.S. Gurr, G.R. Smith, J.P.F. Sellschop, and B. Meyer. Evidence for high-energy cosmic-ray neutrino interactions. *Phys. Rev. Lett.*, 15:429–433, Aug 1965.
- [41] C.V. Achar et al. Detection of muons produced by cosmic ray neutrinos deep underground. *Physics Letters*, 18(2):196–199, 1965.

- [42] The NUSEK Collaboration et al. Experimental study of atmospheric neutrino flux in the NUSEX experiment. *Europhysics Letters (EPL)*, 8(7):611–614, apr 1989.
- [43] K. Daum et al. Determination of the atmospheric neutrino spectra with the fréjus detector. *Zeitschrift für Physik C Particles and Fields*, 66(3):417–428, Sep 1995.
- [44] K.S. Hirata et al. Observation of a small atmospheric muon-neutrino / electron-neutrino ratio in kamiokande. *Phys. Lett. B*, 280:146–152, 1992.
- [45] D. Casper et al. Measurement of atmospheric neutrino composition with the imb-3 detector. *Phys. Rev. Lett.*, 66:2561–2564, May 1991.
- [46] Y. Fukuda et al. Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.*, 81:1562–1567, 1998.
- [47] Z. Li et al. Measurement of the tau neutrino cross section in atmospheric neutrino oscillations with super-kamiokande. *Phys. Rev. D*, 98(5):052006, 2018.
- [48] F. Von Feilitzsch, A.A. Hahn, and K. Schreckenbach. Experimental beta-spectra from ^{239}Pu and ^{235}U thermal neutron fission products and their correlated antineutrino spectra. *Physics Letters B*, 118(1):162–166, 1982.
- [49] K. Schreckenbach, G. Colvin, W. Gelletly, and F. Von Feilitzsch. Determination of the anti-neutrino spectrum from u-235 thermal neutron fission products up to 9.5-mev. *Phys. Lett. B*, 160:325–330, 1985.
- [50] A.A. Hahn, K. Schreckenbach, G. Colvin, B. Krusche, W. Gelletly, and F. Von Feilitzsch. Anti-neutrino spectra from ^{241}Pu and ^{239}Pu thermal neutron fission products. *Phys. Lett. B*, 218:365–368, 1989.
- [51] T.A. Mueller et al. Improved predictions of reactor antineutrino spectra. *Phys. Rev. C*, 83:054615, May 2011.

- [52] A. Bernstein, Y. Wang, G. Gratta, and T. West. Nuclear reactor safeguards and monitoring with anti-neutrino detectors. *J. Appl. Phys.*, 91:4672, 2002.
- [53] K. Eguchi et al. First results from kamland: Evidence for reactor antineutrino disappearance. *Phys. Rev. Lett.*, 90:021802, Jan 2003.
- [54] T. Araki et al. Measurement of neutrino oscillation with kamland: Evidence of spectral distortion. *Phys. Rev. Lett.*, 94:081801, 2005.
- [55] K. Inoue. Reactor neutrino experiments, 03 2004.
- [56] A. Cucoanes. Design studies for the double chooz trigger, 08 2022.
- [57] H. Lu. Recent results from daya bay reactor neutrino experiment, 01 2020.
- [58] H. Lee et al. Measurement of cosmogenic ^9Li and ^8He production rates at reno, 04 2022.
- [59] M.H. Ahn et al. Measurement of neutrino oscillation by the k2k experiment. *Phys. Rev. D*, 74:072003, 2006.
- [60] P. Adamson et al. Combined analysis of ν_μ disappearance and $\nu_\mu \rightarrow \nu_e$ appearance in minos using accelerator and atmospheric neutrinos. *Phys. Rev. Lett.*, 112:191801, 2014.
- [61] P. Adamson et al. Precision constraints for three-flavor neutrino oscillations from the full minos+ and minos dataset. *Phys. Rev. Lett.*, 125(13):131802, 2020.
- [62] N. Agafonova et al. Final results of the opera experiment on ν_τ appearance in the cngs neutrino beam. *Phys. Rev. Lett.*, 120(21):211801, 2018. [Erratum: *Phys.Rev.Lett.* 121, 139901 (2018)].

- [63] A. Aguilar-Arevalo et al. Evidence for neutrino oscillations from the observation of $\bar{\nu}_e$ appearance in a $\bar{\nu}_\mu$ beam. *Phys. Rev. D*, 64:112007, 2001.
- [64] B. Armbruster et al. Upper limits for neutrino oscillations muon-anti-neutrino \rightarrow electron-anti-neutrino from muon decay at rest. *Phys. Rev. D*, 65:112001, 2002.
- [65] A.A. Aguilar-Arevalo et al. Updated minibooone neutrino oscillation results with increased data and new background studies. *Phys. Rev. D*, 103(5):052002, 2021.
- [66] P. Abratenko et al. Search for an excess of electron neutrino interactions in microboone using multiple final state topologies. *Physical review letters*, 10 2021.
- [67] P. Abratenko et al. Search for an anomalous excess of inclusive charged-current ν_e interactions in the microboone experiment using wire-cell reconstruction. *Physical review letters*, 10 2021.
- [68] P. Abratenko et al. Search for an anomalous excess of charged-current ν_e interactions without pions in the final state with the microboone experiment. *Physical review letters*, 10 2021.
- [69] P. Abratenko et al. Search for an anomalous excess of charged-current quasi-elastic ν_e interactions with the microboone experiment using deep-learning-based reconstruction. *Physical review letters*, 10 2021.
- [70] M.A. Acero et al. An improved measurement of neutrino oscillation parameters by the nova experiment. *Phys. Rev. D.*, 106(3), 8 2021.
- [71] G.J. Feldman and R.D. Cousins. A unified approach to the classical statistical analysis of small signals. *Physical Review D*, 57:3873–3889, 1998.

- [72] A. Sousa, N. Buchanan, S. Calvez, P. Ding, D. Doyle, A. Himmel, B. Holzman, J. Kowalkowski, A. Norman, and T. Peterka. Implementation of feldman-cousins corrections and oscillation calculations in the hpc environment for the nova experiment. *EPJ Web of Conferences*, 214, 01 2019.
- [73] S. Tognini. Observation of multiple-muon seasonal variations in the nova near detector, 2018.
- [74] FNAL. Fnal accelerator chain graphic, Feb 2014.
- [75] L.A. Soplin. 2017-2018 beam plots, May 2018. NOvA interal document, DocDb 20843-v12.
- [76] G. Feldman, J. Hartnell, and T. Kobayashi. Long-baseline neutrino oscillation experiments, 01 2013.
- [77] Nova stacked cells backlit, 04 2010.
- [78] Nova neutrino experiment detectors, 04 2010.
- [79] S. Mufson et al. Liquid scintillator production for the nova experiment. *Nucl. Instrum. Meth. A*, 799:1–9, 2015.
- [80] S. et al Mufson. Liquid scintillator production for the nova experiment, 2015.
- [81] Innospec Fuel Specialties LLC(USA). Stadis (r) 425 safety data sheet. Innospec, 11 2015.
- [82] S. Mufson et al. Liquid scintillator production for the nova experiment, 2015.
- [83] S. Tognini. Observation of multiple-muon seasonal variations in the nova near detector, 2018.
- [84] Nova optical connector, 03 2013.

- [85] R. Hahn. Nova front end board, 05 2011.
- [86] C.J. Backhouse. Improving timing resolution, Oct 2011. NOvA internal document, DocDb 6567.
- [87] S.A. Kharusi et al. SNEWS 2.0: a next-generation supernova early warning system for multi-messenger astronomy. *New Journal of Physics*, 23(3):031201, mar 2021.
- [88] M. Campanella, A. Ferrari, P.R. Sala, and S. Vanini. First calorimeter simulation with the flugg prototype. 12 1999.
- [89] A. Fasso et al. The fluka code: present applications and future developments. 2003.
- [90] J. Allison et al. Recent developments in geant4. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 835:186–225, 2016.
- [91] S. at al. Agostinelli. Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, 2003.
- [92] J. Allison et al. Geant4 developments and applications. *IEEE Transactions on Nuclear Science*, 53(1):270–278, 2006.
- [93] L. Aliaga et al. Neutrino flux predictions for the numi beam. *Phys. Rev. D*, 94(9):092005, 2016.
- [94] H. Gallagher. Neutrino event generators. 3 2016.
- [95] C. Andreopoulos et al. The genie neutrino monte carlo generator. *Nucl. Instrum. Meth. A*, 614:87–104, 2010.
- [96] D. Doyle et al. Measurement of the double-differential muon-neutrino charged-current inclusive cross section in the nova near detector. 9 2021.

- [97] M.A. Acero et al. Adjusting neutrino interaction models and evaluating uncertainties using NOvA near detector data. *The European Physical Journal C*, 80(12), dec 2020.
- [98] R.A. Smith and E.J. Moniz. Neutrino reactions on nuclear targets. *Nucl. Phys. B*, 43:605, 1972. [Erratum: *Nucl.Phys.B* 101, 547 (1975)].
- [99] A. Bodek and J.L. Ritchie. Further studies of fermi-motion effects in lepton scattering from nuclear targets. *Phys. Rev. D*, 24:1400–1402, Sep 1981.
- [100] C.H. Llewellyn Smith. Neutrino reactions at accelerator energies. *Physics Reports*, 3(5):261–379, 1972.
- [101] K. Teppe. Cross section analyses in miniboone and sciboone experiments. *AIP Conf. Proc.*, 1663(1):020001, 2015.
- [102] D. Rein and L.M. Sehgal. Neutrino-excitation of baryon resonances and single pion production. *Annals of Physics*, 133(1):79–153, 1981.
- [103] D. Rein and L.M. Sehgal. Coherent π^0 production in neutrino reactions. *Nuclear Physics B*, 223(1):29–44, 1983.
- [104] D. Rein and L.M. Sehgal. π^0 and the deficit of forward muons in π^+ production by neutrinos. *Phys. Lett. B*, 657:207–209, 2007.
- [105] A Bodek and U.K. Yang. Higher twist, $x_1(\omega)$ scaling, and effective $1/Q^2$ pdfs for lepton scattering in the few gev region. *J. Phys. G*, 29:1899–1906, 2003.
- [106] T. Yang, C. Andreopoulos, H. Gallagher, K. Hoffmann, and P. Kehayias. A hadronization model for few-gev neutrino interactions. *Eur. Phys. J. C*, 63:1–10, 2009.
- [107] T. Sjostrand, S. Mrenna, and P.Z. Skands. Pythia 6.4 physics and manual. *JHEP*, 05:026, 2006.

- [108] M.A. Acero et al. Adjusting neutrino interaction models and evaluating uncertainties using nova near detector data. *Eur. Phys. J. C*, 80(12):1119, 2020.
- [109] O. Buss, T. Gaitanos, K. Gallmeister, H. Van Hees, M. Kaskulov, O. Lalakulich, A. B. Larionov, T. Leitner, J. Weil, and U. Mosel. Transport-theoretical description of nuclear reactions. *Phys. Rept.*, 512:1–124, 2012.
- [110] C. Hagmann, D. Lange, and D. Wright. Cosmic-ray shower generator (cry) for monte carlo transport codes. 2:1143–1146, 2007.
- [111] A. Aurisano. The nova detector simulation (interal technote). 6 2015.
- [112] J.B. Birks. Scintillations from organic crystals: Specific fluorescence and relative response to different radiations. *Proceedings of the Physical Society. Section A*, 64(10):874–877, oct 1951.
- [113] C.N. Chou. The nature of the saturation effect of fluorescent scintillators. *Phys. Rev.*, 87:904–905, Sep 1952.
- [114] A. Vold. Improving physics based electron neutrino appearance identification with a long short-term memory network, Aug 2018.
- [115] K.A. Olive et al. Review of particle physics, 2014.
- [116] K.A. Olive et al. Review of particle physics. *Chin. Phys. C*, 38:090001, 2014.
- [117] R. Nichol et al. Calibration technotes, Jun 2015. NOvA interal document, DocDb 13579-v35.
- [118] C. Green, J. Kowalkowski, M. Paterno, M. Fischler, L. Garren, and Q. Lu. The art framework. *J. Phys. Conf. Ser.*, 396, 2012.
- [119] Chire. Dbscan-illustration, Oct 2011.

- [120] M. Baird. Slicing module comparison tech note. May 2013. NOvA internal document, DocDb 9195.
- [121] J. Huang. Tdslicer technote. Mar 2018. NOvA internal document, DocDb 27689-v5.
- [122] A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.
- [123] R.C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.
- [124] A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks, 2014.
- [125] D. Pershey. Tdslicer, Oct 2016. NOvA internal document, DocDb 16407-v2.
- [126] J. Huang. Tdslicer technote, Mar 2018. NOvA internal document, DocDb 27689-v5.
- [127] J. Bezdek. Pattern Recognition With Fuzzy Objective Function Algorithms. 01 1981.
- [128] R. Krishnapuram and J.M. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2):98–110, 1993.
- [129] M.S. Yang and K.L. Wu. Unsupervised possibilistic clustering. 39(1), 2006.
- [130] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [131] N.J. Raddatz. Kalman filter tracking, Apr 2011. NOvA internal document, DocDb 5855.
- [132] G. Lutz. Optimum track fitting in the presence of multiple scattering. *Nucl. Instrum. Meth. A*, 273:349, 1988.

- [133] F. Psihas, M. Groh, C. Tunnell, and K. Warburton. A review on machine learning for neutrino experiments. *Int. J. Mod. Phys. A*, 35(33):2043005, 2020.
- [134] K.C. Yao, W.T. Huang, C.C. Wu, and T.Y. Chen. Establishing an ai model on data sensing and prediction for smart home environment control based on labview, 07 2021.
- [135] M. Tonutti. A data-driven method for real-time modelling of brain tumour deformation, 09 2016.
- [136] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. The MIT Press, Nov. 2016.
- [137] C. Sousa. An overview on weight initialization methods for feedforward neural networks. 07 2016.
- [138] P. Patel, M. Nandu, and P. Raut. Initialization of weights in neural networks. Volume 4:73 – 79, 02 2019.
- [139] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, 9:249–256, 01 2010.
- [140] K. Janocha and W. Czarnecki. On loss functions for deep neural networks in classification. *Schedae Informaticae*, 25, 02 2017.
- [141] A. Amini, A. Soleimany, S. Karaman, and D. Rus. Spatial uncertainty sampling for end-to-end control, 05 2018.
- [142] A. Alamsyah, M. Apriandito, and R. Masrury. Object detection using convolutional neural network to identify popular fashion product, 03 2019.
- [143] Y. Teo, S. Shin, H. Jeong, Y. Kim, Y.H. Kim, G. Struchalin, E. Kovlakov, S. Straupe, S. Kulik, G. Leuchs, and L. Sanchez-Soto. Benchmarking

- quantum tomography completeness and fidelity with machine learning, 10 2021.
- [144] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. 25, 2012.
- [145] S. Shehata. Using mid- and high-level visual features for surgical workflow detection in cholecystectomy procedures, 07 2016.
- [146] C. Szegedy, L. Wei, J. Yangqing, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. pages 1–9, 2015.
- [147] B. Zhao, J. Feng, X. Wu, and S. Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation, 01 2017.
- [148] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions, 06 2015.
- [149] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 06 2016.
- [150] M. Gao, J. Chen, H. Mu, and D. Qi. A transfer residual neural network based on resnet-34 for detection of wood knot defects, 02 2021.
- [151] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. ArXiv, abs/1704.04861, 2017.
- [152] J.M.S. Prewitt. Object enhancement and extraction. Picture Processing and. Psychopictorics, 1970.
- [153] B.R. Ripley. Pattern Recognition and Neural Networks. Cambridge University Press, 1st edition, Jan. 2008.

- [154] R. Sharma, A. Nori, and A. Aiken. Bias-variance tradeoffs in program analysis, 01 2014.
- [155] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. 12 2014.
- [156] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 06 2014.
- [157] G. Louppe, M. Kagan, and K. Cranmer. Learning to pivot with adversarial networks. Nov. 2016.
- [158] S. Harada, H. Hayashi, and S. Uchida. Biosignal generation and latent variable analysis with recurrent generative adversarial networks, 05 2019.
- [159] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by back-propagation. Sep 2014.
- [160] C. Englert, P. Galler, P. Harris, and M. Spannowsky. Machine learning uncertainties with adversarial neural networks. *European physical journal C.*, 79(1):4, Jan 2019.
- [161] G. Louppe, M. Kagan, and K. Cranmer. Learning to pivot with adversarial networks, Nov. 2016.
- [162] S. Wunsch, S. Jörger, R. Wolf, and G. Quast. Reducing the dependence of the neural network function to systematic uncertainties in the input space. Jun 2019.
- [163] A. Back, R.S. Bowles, M. Elkins, B Ramson, and S. Sanchez. Data quality 2020 technote. Mar 2020. NOvA interal document, DocDb 44226-v1.
- [164] R. Murphy, B. Chowdhury, and J. Coelho. Good data selection technote. Nov 2017. NOvA interal document, DocDb 13546-v6.

- [165] R. Murphy and M. Messier. Feb flashes. Jul 2014. NOvA interal document, DocDb 11663-v1.
- [166] M. Baird, L. Suter, and J. Wolcott. Summary of the 2020 fhc+rhc 3-flavor oscillation analysis. Apr 2020. NOvA interal document, DocDb 44422-v3.
- [167] B. Rebel. A window tracking algorithm for cosmic ray muons. Aug 2016. NOvA interal document, DocDb 15977-v1.
- [168] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.
- [169] M. Groh and K. Warburton. Cvn training for the 2020 analysis. Mar 2020. NOvA interal document, DocDd 42897-v2.
- [170] A. Himmel, A. Moren, and E. Niner. Early reusable cosmic rejection with a cnn. Feb 2020. NOvA interal document, DocDd 42766-v2.
- [171] R. Brun and F. Rademakers. Root — an object oriented data analysis framework. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 389(1):81–86, 1997.
- [172] The HDF Group. Hierarchical data format, version 5. 1997-2022.
- [173] M. Tanabashi et al. Review of particle physics. Phys. Rev. D, 98:030001, Aug 2018.
- [174] H. He and E.A. Garcia. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21:1263–1284, 2009.
- [175] Y. Hayato and L. Pickering. The neut neutrino interaction simulation program library. Eur. Phys. J. ST, 230(24):4469–4481, 2021.

- [176] T. Golan, J. T. Sobczyk, and J. Zmuda. Nuwro: the wroclaw monte carlo generator of neutrino interactions. *Nucl. Phys. B Proc. Suppl.*, 229-232:499–499, 2012.
- [177] M.A. Acero et al. An improved measurement of neutrino oscillation parameters by the nova experiment, 8 2021.
- [178] A.M. Hall. Light level down /fd ageing comparison. Oct 2019. NOvA interal document, DocDd 41070-v1.
- [179] FNAL (WC-IC). Screen-shot-2020-12-17-at-2.48.32-pm-1-1024x545.png, 12 2020.
- [180] F. Chollet et al. Keras. 2015.
- [181] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. Software available from tensorflow.org.
- [182] R. Al-Rfou et al. Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints, abs/1605.02688, may 2016.
- [183] J. Araújo, C. Chow, and M. Janota. Boosting isomorphic model filtering with invariants, 07 2022.
- [184] A.B. Yoo, M.A. Jette, and M. Grondona. Slurm: Simple linux utility for resource management. pages 44–60, 2003.
- [185] S. Kullback and R.A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.
- [186] N.H. Kuiper. Tests concerning random points on a circle. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Series A*, (63):38 – 47, 1960.
- [187] D.M. Mason and J.H. Schuenemeyer. A modified kolmogorov-smirnov test sensitive to tail alternatives. *The Annals of Statistics*, 11(3):933 – 946, 1983.

- [188] L. Hertel, J. Collado, P. Sadowski, J. Ott, and P. Baldi. Sherpa: Robust hyperparameter optimization for machine learning. *SoftwareX*, 12:100591, 2020.
- [189] Y. Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.
- [190] P. Baldi, K. Bauer, C. Eng, P. Sadowski, and D. Whiteson. Jet substructure classification in high-energy physics with deep neural networks. *Phys. Rev. D*, 93:094034, May 2016.
- [191] C. Adam-Bourdarios, G. Cowan, C. Germain-Renaud, I. Guyon, D. Rousseau, and B. Kegl. The higgs machine learning challenge. *Journal of Physics: Conference Series*, 664(7):072015, dec 2015.