Background Reduction Techniques for Neutrinoless Double Beta Decay Search at CANDLES Experiment

Temuge Batpurev
Department of Physics, Graduate School of Science,
Osaka University

Doctoral Thesis

January 20, 2020

Abstract

Neutrinoless Double Beta Decay($0\nu\beta\beta$) search is a subject of intense experimental and theoretical research. The detection of this decay provides direct evidence of Majorana nature of the neutrino. Moreover, the absolute mass scale of the electron neutrino can be written in terms of the decay rate of $0\nu\beta\beta$ and thus the decay provides information on neutrino mass scales.

The $0\nu\beta\beta$ decay is a special case of the double beta decay, which emits two neutrinos and two electrons. The double beta decay is only observed in nuclei where a single decay of an electron results in higher energy state while double electron decay results in lower energy state. Only a few tens of nuclei are candidates for the double beta decay and a direct evidence of the decay was first found in 1987. Since then, the double beta decay has been observed in many other nuclei and has a half-life on the order of 10^{21} to 10^{22}. The $0\nu\beta\beta$ decay has not been observed and theoretical estimate of the half-life is on the order of 10^{27}.

The CANDLES(CAlcium fluoride for studies of Neutrino and Dark matters by Low Energy Spectrometer) experiment is looking for the $0\nu\beta\beta$ decay using ^{48}Ca as the source. The nuclei was chosen due to it having the highest $Q_{\beta\beta}$ value among the candidate nuclei at 4.3 MeV. There are few observed natural backgrounds in this energy range. However, ^{48}Ca has a very low natural abundance at 0.187%. Thus, the experimental challenge is to reduce the count of observed natural backgrounds as much as possible to detect signal from the low number of candidate ^{48}Ca isotopes.

The main background in CANDLES experiment is generated from the disintegration of ^{212}Bi, which appears in the detector due to Uranium contamination. The ^{212}Bi nuclei has an α channel and a β channel decay, and both of these channels create background events up to 5 MeV. Thus, CANDLES collaborators spend significant efforts to reduce these background.

In this study, we developed two new methods for the reduction of ^{212}Bi background events. One method is based on traditional analysis and we show that it can achieve high background rejection efficiency and low accidental rejection rate. The other method is based on machine learning, more particularly neural networks, and we show that it provides a significant improvement over previous methods. Since neural networks are not part of the standard analysis tool-set of a physicist, an introductory chapter is also provided. One of the goals of the study is to show that neural networks can be adapted for physics analysis, despite some of its shortcomings.

Contents

Chapter 1

Introduction

1.1 Double Beta Decay

The double beta decay($2\nu\beta\beta$) is a transition between two different nuclei with the same mass number A. The characteristic equation can be written as $(A, Z) \longrightarrow (A, Z+2) + 2e + 2\overline{\nu_e}$. This decay occurs if the initial nuclei is energetically forbidden to decay to the intermediate $(A, Z+1)$ nuclei, while the final $(A, Z+2)$ nuclei has lower overall energy. In the ^{48}Ca case, the β decay to ^{48}Sc is hindered, while the $2\nu\beta\beta$ decay to ^{48}Ti is energetically allowed. The first direct evidence of $2\nu\beta\beta$ was presented by S.R. Elliott and collaborators in 1987 using ^{82}Se nuclei[1]. Since then, $2\nu\beta\beta$ decay has been observed in many other nuclei[2].

The question of whether the neutrino follows Dirac statistics or Majorana statistics is still an open question. Because neutrino does not have charge and has no color change, the only distinguishing factor between the neutrino and the anti-neutrino is the lepton number, which is not required to be conserved in many extensions of the standard model [3]. The neutrinoless double beta decay($0\nu\beta\beta$) has been proposed and studied to determine the nature of the neutrino.

The $0\nu\beta\beta$ is a lepton number violating decay and the characteristic equation can be written as $(A, Z) \longrightarrow (A, Z+2) + 2e$ and it has a change in lepton number of 2. This decay is not allowed if the neutrino is a Dirac type particle because the anti-neutrino emitted by neutron does not have the right chirality to interact with the other neutrino. In other words, the neutrino and anti-neutrino are distinguishable. However, with the discovery of the neutrino oscillation[4], we have evidence that the neutrinos have mass and thus chiral symmetry can be broken. If the neutrino is a Majorana type particle, then the neutrino and the anti-neutrino are the same particles and thus weak interaction allows the $0\nu\beta\beta$ decay. This would then be a direct evidence of the Majorana nature of the neutrino.

Because neutrino is not emitted in the $0\nu\beta\beta$ decay, almost all of the

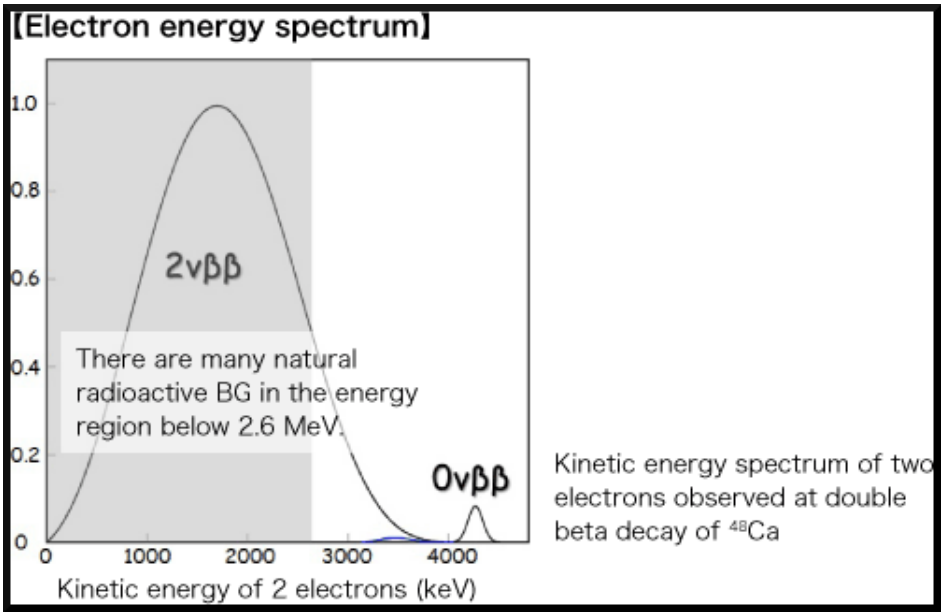


Figure 1.1: The $2\nu\beta\beta$ and the $0\nu\beta\beta$ decay energy spectrum. Because no neutrinos are emitted in $0\nu\beta\beta$, almost all energy is carried away by the β particles.

energy is carried away by the two β particles. Some energy is transferred to the parent nuclei due to recoil. Figure 1.1 shows the expected energy spectrum of the $0\nu\beta\beta$ and $2\nu\beta\beta$ decays. The $2\nu\beta\beta$ decay emits two neutrinos that carry away energy. Thus the energy spectrum of $2\nu\beta\beta$ is continuous over a large energy range. The $0\nu\beta\beta$ decay is sharply peaked near a single value. This value is called the Q value and is equivalent to the energy difference of the initial state and the final state of the $0\nu\beta\beta$ decay. Due to the extreme low decay rate of $0\nu\beta\beta$, detectors are required to be very low background. Thus, many $0\nu\beta\beta$ experiments are located underground to reduce cosmic backgrounds. It should be noted that the ratio of the half-lives of $2\nu\beta\beta$ and $0\nu\beta\beta$ is expected to be larger than 10^6. Then, the plot in Fig. 1.1 is largely exaggerated.

The decay rate of the $0\nu\beta\beta$ can be written as:

$$\frac{1}{T^{0\nu}_{1/2}} = G_{0\nu}|M^{0\nu}|^2\langle m_{\beta\beta}\rangle^2 \tag{1.1}$$

where $T^{0\nu}_{1/2}$ is the half-life of the decay, $G_{0\nu}$ the two-body phase-space factor, $M^{0\nu}$ the nuclear matrix element(NME) and $\langle m_{\beta\beta}\rangle^2$ the effective Majorana mass of the electron neutrino[5]. The mass term $m_{\beta\beta}$ can be written in

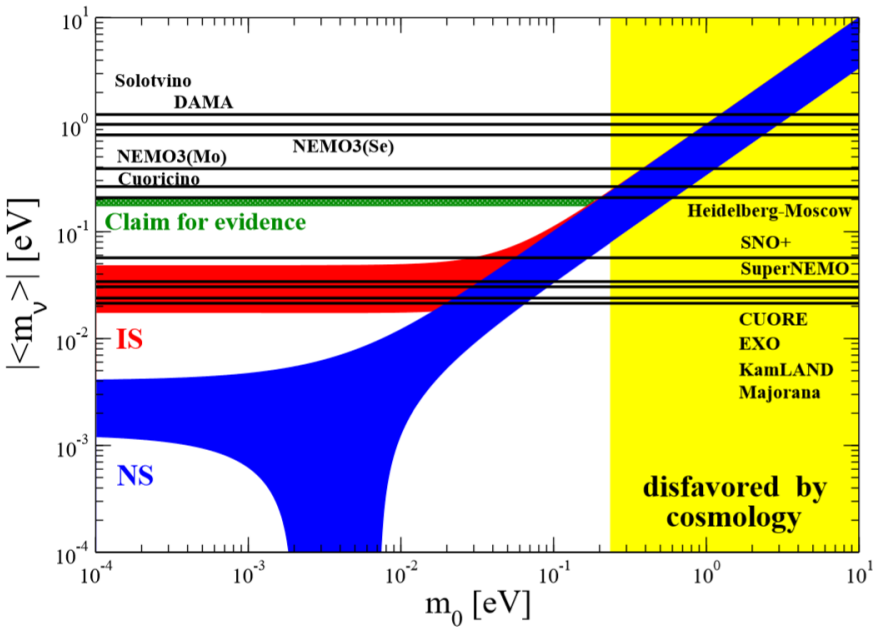


Figure 1.2: The absolute mass scale of the electron neutrino is shown as a function of the lowest mass state of the neutrino. The sensitivity of the newer experiments are able to probe the absolute mass scale in the inverted hierarchy range. *Figure source: [3]*

terms of the neutrino masses and the PMNS matrix as:

$$m_{\beta\beta} = \sum_{i=1}^{3} U_{ei}^2 m_i \tag{1.2}$$

where m_i are the neutrino masses and the U_{ei} are the elements of the Pontecorve-Maki-Nakagawa-Sakata(PMNS) matrix related to the electron neutrino. The PMNS matrix describes the neutrino mixing parameters between the mass states and the flavor states. Thus, observation of the $0\nu\beta\beta$ provides information on the absolute mass scale of the neutrinos. Figure 1.2 shows the absolute mass scale as a function of the lowest mass state. The newer detectors and experiments are now able to probe the absolute mass scale for the inverted hierarchy.

The decay rate of the $0\nu\beta\beta$ is dependent on the nuclear matrix element(NME). This is a difficult calculation because initial and final state nuclear wave function must be determined. Overview of some of the approaches to this calculation can be found in [6]. The differing approaches provide differing values of the NME and some examples are shown in Fig. 1.3. We observe that for some nuclei, the NME can change by almost an order of magnitude. Thus, observation of the $0\nu\beta\beta$ places some restrictions to some of the calculation methods.

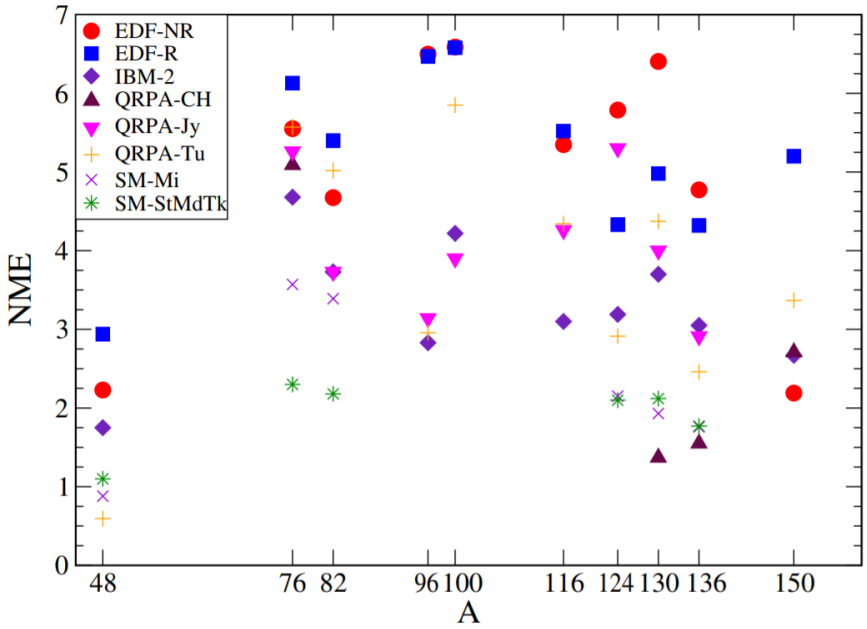


Figure 1.3: NME calculated for various $0\nu\beta\beta$ candidate nuclei are shown by different calculation methods. *Figure source: [6]*

In summary, the $0\nu\beta\beta$ decay is a subject of intense experimental and theoretical study. Observation of the decay is a direct evidence of the Majorana nature of the neutrino. It provides important information to the standard model extensions and may lead to physics beyond the standard model. Also, the decay can provide valuable information on the absolute mass scale of the neutrinos. These are the motivations for $0\nu\beta\beta$ study.

1.2 CANDLES Experiment

The CANDLES(CAlcium fluoride for studies of Neutrino and Dark matters by Low Energy Spectrometer) experiment is currently operating at the Kamioka Underground Laboratory[7]. The experiment uses the ^{48}Ca isotope for the $0\nu\beta\beta$ decay search. Among the few $0\nu\beta\beta$ decay candidate nuclei[8], the ^{48}Ca has the highest $Q_{\beta\beta}$ value at 4.27 MeV. This gives ^{48}Ca an advantage over other double beta decay candidate nuclei because the high $Q_{\beta\beta}$ value is larger than most natural background energy ranges. Table 1.1 shows other $0\nu\beta\beta$ decay candidate nuclei along with corresponding experiments. The natural abundance of ^{48}Ca is roughly 0.2%. This poses a challenge from an experimental point of view. Due to the low abundance, the detector must have a significant amount of ^{48}Ca present. The half-life of $0\nu\beta\beta$ is expected to be larger than 10^{26} yr. Since the event rate of $0\nu\beta\beta$ is extremely low and

Nulcei	$Q_{\beta\beta}$ value $[keV]$	Natural Abundance [%]	$2\nu\beta\beta$ half-life [year]	Experiment
^{48}Ca	4274	.187	$0.44 * 10^{20}$	CANDLES, NEMO-3
^{76}Ge	2039	7.8	$15 * 10^{20}$	GERDA, MAJORANA
^{82}Se	2996	9.2	$0.92 * 10^{20}$	SuperNEMO, LUCIFER
^{100}Mo	3035	9.6	$0.07 * 10^{20}$	NEMO-3, MOON, AMoRe
^{116}Cd	2809	7.6	$0.29 * 10^{20}$	COBRA
^{130}Te	2530	34.5	$9.1 * 10^{20}$	CUORE, SNO+
^{136}Xe	2458	8.9	$21 * 10^{20}$	KamLAND-Zen, EXO, NEXT
^{150}Nd	3367	5.6	$0.08 * 10^{20}$	SNO+ , DCBA/MTD

Table 1.1: List of $0\nu\beta\beta$ decay candidate nuclei.

the abundance of ^{48}Ca is also low, the number of background events must be minimized as much as possible. For this reason, the detector is located underground, at 2,700 m water equivalent depth.

The CANDLES experiment uses CaF_2 crystals containing target ^{48}Ca nuclei and these crystals are scintillators. We have 96 crystals in the detector and these contain about 365 g of ^{48}Ca nuclei. Thus the CaF_2 crystals act as the source and the detector. More detailed description of detector is given in chapter 2.

1.3 Other Experiments for $0\nu\beta\beta$ search

In this section, results of several past and currently running experiments for $0\nu\beta\beta$ search is summarized.

1.3.1 Heidelberg-Moscow Experiment

The Heidelberg-Moscow experiment used High Purity Ge(HPGe) detector with 11 kg of Ge enriched to 87% ^{76}Ge to measure the half-life of $0\nu\beta\beta$. In 2001, some members of the experiment reported a first evidence for detection of $0\nu\beta\beta$ using data from measuring period of August 1990 - May 2000 (54.98 kg y) [9]. The observed energy spectrum is shown in Fig. 1.4 and the reported $0\nu\beta\beta$ half-life of ^{76}Ge was:

$$T^{0\nu}_{1/2} = (0.8 - 18.3) \times 10^{25} y \ (95\% \ c.l.) \tag{1.3}$$

with best value of $1.5 \times 10^{25} y$. The effective neutrino mass was reported to be:

$$\langle m_{\beta\beta} \rangle = (0.11 - 0.56) \ eV \ (95\% \ c.l.) \tag{1.4}$$

with best value of 0.39 eV. However, the result of the experiment generated some criticism and has not been confirmed by other experiments.

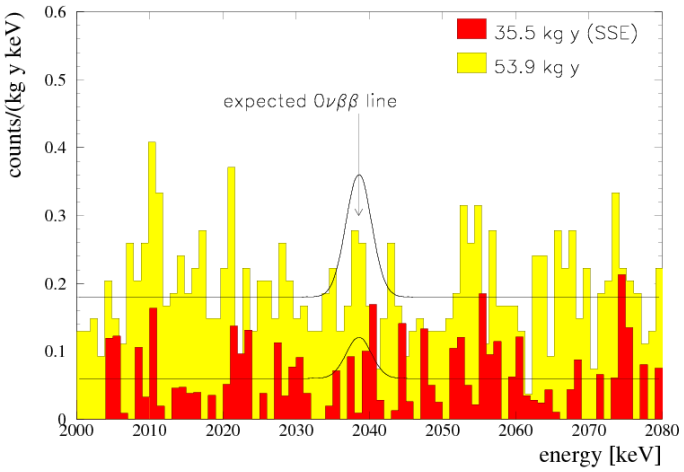


Figure 1.4: The observed energy spectrum of the Heidelberg-Moscow Experiment.

1.3.2 GERDA Experiment

The Germanium Detector Array(GERDA) experiment is currently operating at the Laboratori Nazionali del Gran Sasso in Italy. The experiment uses 86% enriched ^{76}Ge immersed in liquid argon cryostat. In phase-I, 17.8 kg of enriched coaxial Ge detectors were used and later upgraded by 3.63 kg of Broad Energy Ge detectors. Using data from 2011 to 2013, accumulating 21.6kg y of exposure, GERDA experiment refuted the results of the Heidelberg-Moscow experiment. In phase-II, the detector was upgraded to 35.8 kg of enriched ^{76}Ge. Using data from both phases, a lower limit on the half-life was reported[10]:

$$T^{0\nu}_{1/2} = 8.0 \times 10^{25}\ y\ (90\%\ c.l.) \tag{1.5}$$

1.3.3 Majorana Demonstrator Experiment

The Majorana Demonstrator(MJD) experiment is currently operating at the Sanford Underground Research Facility in South Dakota, US. MJD uses 88% enriched ^{76}Ge with a mass of 29.7 kg. The Ge detector is installed in copper vacuum cryostat, which is shielded by layers of copper, lead and polyethylene shieldings. The design principle of the detector is ultra-clean materials and process controls to reach minimum backgrounds. The latest results, using ^{76}Ge exposure of (26.0 $\pm$ 0.5) kg y, places the lower limit on half-life for ^{76}Ge to be [11]:

$$T^{0\nu}_{1/2} = 2.7 \times 10^{25}\ y\ (90\%\ c.l.) \tag{1.6}$$

Depending on the nuclear matrix element estimates, the upper limit on the effective neutrino mass is also estimated to be in the range of (200-433) meV.

1.3.4 CUORE Experiment

The Cryogenic Underground Observatory for Rare Events(CUORE) experiment is also operating at the Laboratori Nazionali del Gran Sasso in Italy. CUORE is a bolometric experiment, using 988 TeO_2 crystals, arranged into 19 cylindrical towers, with each tower containing 52 crystals. The experiment started taking data in spring of 2017. Their first reported result, using ^{130}Te exposure of 86.3 kg y, places the lower limit on the half-life of ^{130}Te to be[12]:

$$T^{0\nu}_{1/2} = 1.3 \times 10^{25}\ y\ (90\%\ c.l.) \tag{1.7}$$

The experiment is expected to reach sensitivity of $9 \times 10^{25}\ y$ in five years of data taking.

1.3.5 KamLAND-Zen Experiment

The kamLAND-Zen(Kamioka Liquid Scintillator Anti-Neutrino Detector - Zero Neutrino) experiment is operating at the Kamioka Observatory in Gifu prefecture, Japan. The target nuclei is ^{136}Xe, with 91% enrichment in the detector. In phase-I, data from October 2011 to June 2012 was used to obtain exposure of 89.5 kg y, and lower limit on the half-life was placed at $> 1.9 \times 10^{25}\ y$.

In phase-II, the Xe mass was increased from 320 kg to 380 kg and data was taken from November 2013 to October 2015. The ^{136}Xe exposure in this phase was 504 kg y. Combining the results of both phases, the new lower limit on the half-life of ^{136}Xe was set to[13]:

$$T^{0\nu}_{1/2} = 1.07 \times 10^{26}\ y\ (90\%\ c.l.) \tag{1.8}$$

The upper limit on the effective neutrino mass was estimated to be $< (61 - 165)$ meV.

1.4 Main Backgrounds of CANDLES

The ^{48}Ca isotope was chosen for CANDLES due to its high Q value of 4.3 MeV for $0\nu\beta\beta$. Most natural backgrounds deposit few events in this energy range. Figure 1.5 shows the energy spectrum of CANDLES recorded over 400 days. The events shown are β like events. Major background counts are significantly reduced above 3 MeV. However, we see a large number of events in the Q value region. These events are almost entirely from the decay of ^{212}Bi.

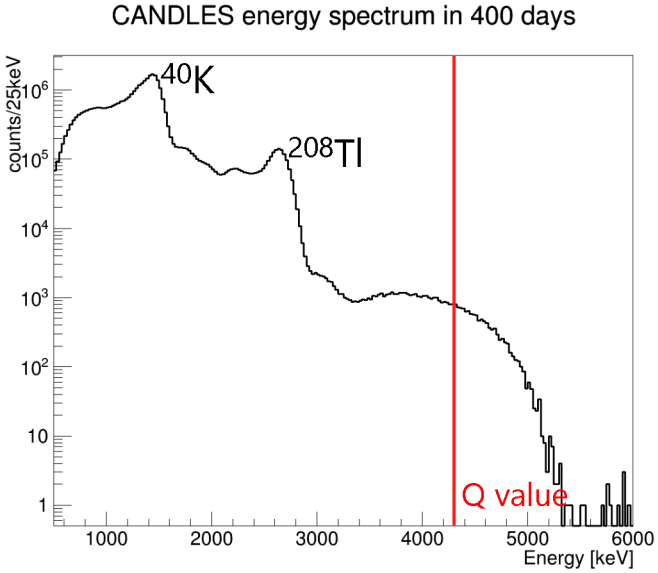


Figure 1.5: CANDLES energy spectrum is shown for β like events over 430 days. The backgrounds observed in the Q value region is almost entirely from the decay of ^{212}Bi.

The main background ^{212}Bi appears in CANDLES data due to Uranium contamination of the CaF_2 crystals. The decay scheme of ^{212}Bi is shown in Fig. 1.6. It has two decay channels but they both lead to stable ^{208}Pb. Both decay channels deposit energy in the Q value region with a maximum energy of about 5 MeV. Therefore, significant effort is spent by the CANDLES collaborators to reduce these background events.

The α decay channel has an energy of 6.21 MeV and decays to ^{208}Tl. Due to the α quenching factor, the observed energy is around 1.7 MeV. Thus, the α decay to ^{208}Tl is not a serious background in the Q value region. However, ^{208}Tl is a beta decay nuclei to ^{208}Pb and it has a Q value of 5 MeV. The energy spectrum of ^{208}Tl in CANDLES is shown in Fig. 1.7 and the decay scheme is shown in Fig. 4.1. We observe that a significant number of background events are expected in the Q value region from the disintegration of ^{208}Tl. To detect and remove these events, we check for coincidence between the candidate events and the α events decayed by the ^{212}Bi parent in the same crystal. This is known as the delayed coincidence method. Because the half-life of ^{208}Tl is 3 minutes, we can find the preceding α with relatively high efficiency.

The decay of ^{208}Tl also produces several γ particles and the largest energy among them is 2.6 MeV. These γ particles have a large chance of escaping the original crystal and depositing energy in nearby crystals. In this case, the position of the event may be reconstructed in another crystal

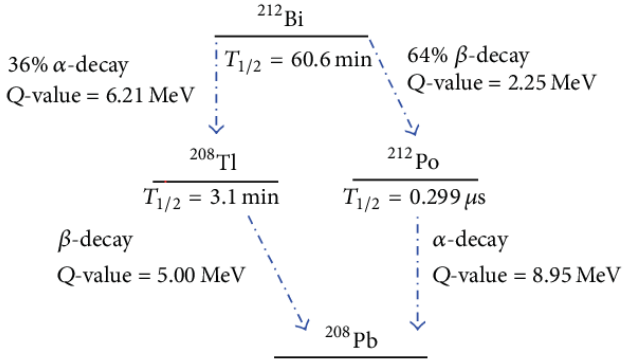


Figure 1.6: The decay scheme of ^{212}Bi is shown. It has two decay channels and both channels produce background events in the Q value region.

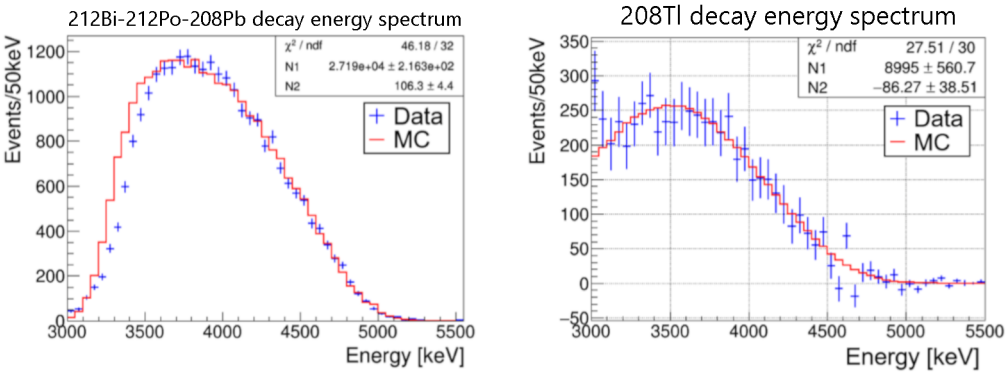


Figure 1.7: The energy spectrum of ^{212}Bi decays.

and thus the delayed coincidence method will not be able to find the α particle of the parent. An example of this event's possible mechanism is shown in Fig. 4.2. Currently, we do not have a method to detect and reject these multi-crystal events.

The β decay branch of ^{212}Bi decays to ^{212}Po which then decays to ^{208}Pb by emitting an α particle. Because the half-life of ^{212}Po is 300ns, the ^{212}Bi $\rightarrow$ ^{212}Po $\rightarrow$ ^{208}Pb chain appears as a single event in our detector. These appear as double pulse events because shortly after the initial β decay, the second α pulse appears and the event signal appears as a combination of two pulses. An example of these double pulse events is shown in Fig. 5.1.a. The energy spectrum of these double pulse events in CANDLES detector is shown in Fig. 1.7. Thus, the β branch of ^{212}Bi also deposit a large number of events in the Q value region. To detect and reject these events, we apply a reference α and β pulse combination to fit an event's signal and determine

the decay time between the two pulses. An example of this method is shown in Fig. 5.2. However, we found that this method works poorly when the decay time is less than 20ns.

1.5 Purpose of the Study

The purpose of this study is the reduction of main backgrounds due to ^{212}Bi. We developed two new methods for background reduction; one to detect and reject multi-crystal events arising from ^{208}Tl decay and the other to improve double pulse event detection below 20ns decay time. As our experiment aims to achieve zero background measurements in the Q value region, developing background reduction techniques is very important.

In this study, we used a new method based on machine learning. Machine learning, more particularly neural networks, has become a very prevalent subject in modern data sciences. However, adoption of neural network based techniques by the physics community has been slow. There are some underlying issues with neural networks that make it less appealing for physics data analysis. More details on this will be given later. In this study, we ask if neural networks can be adopted for physics data analysis if we could overcome some of its drawbacks using external information. Our neural network application provided significant improvements over previous method and the details are presented in this study.

Chapter 2

CANDLES Experiment

2.1 CANDLES Detector

The CANDLES detector is located at the Kamioka Underground Observatory in Gifu prefecture, Japan. The detector is placed in 2,700 m water equivalent depth to help reduce cosmic muon backgrounds. The detector overview is shown in Fig. 2.1. CANDLES detector is placed inside a steel tank of cylindrical shape with size (3 x 4) m. The inside and the outside of the detector is lined with 5 mm thick Boron sheets. These work as neutron shield to reduce (n, γ) reactions. The main shielding component is boron carbide (B_4C) with ratio of 40% by weight. The sides and the bottom of the detector is shielded by 10 cm to 12 cm of Pb. Due to weight concerns, the top of the detector is shielded by Pb of 7 cm thickness. With these shielding installed, external backgrounds are expected to be reduced to a few events per year[14].

The detector is filled with purified water that act as an additional layer of shielding. The inner walls are mounted with 62 photo-multiplier tubes(PMTs) that collect and record photon signals. The main tank contains a secondary acrylic tank of size (1.4 x 1.4) m. The acrylic tank is filled with a liquid scintillator and also contain all 96 CaF_2 crystals. Details of each of the components are presented below. To increase photon collection efficiency, geometric light funnels with reflective surfaces are installed to reflect photons toward the PMTs.

2.1.1 CaF_2 Crystals

CANDLES detector uses 96 CaF_2 scintillating crystals[15]. The scintillation rate is a bit over 1,000 photons emitted per 1 MeV deposited in crystal and is dependent on temperature. The CaF_2 crystals contain about 365g of our target nuclei, ^{48}Ca. Therefore, the CaF_2 crystals act as decay source and scintillating detector. The crystals are cooled to 5 degrees Celsius to increase photon yield.

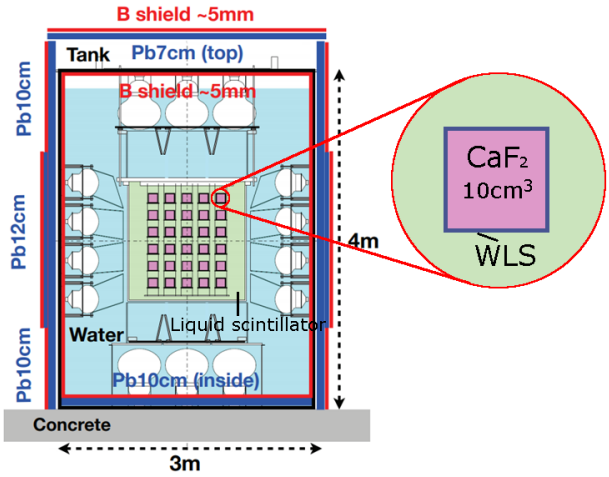


Figure 2.1: CANDLES Detector. The CANDLES detector is contained in a steel tank of 3mx4m. The inner walls of the detector is installed with 62 photo-multiplier tubes(PMTs). The detector has an inner acrylic tank that houses the 96 CaF_2 crystals. The tank is filled with purified water and the inner acrylic tank is filled with liquid scintillator. The CaF_2 crystals are housed inside a wavelength shifter(WLS).

The CaF_2 crystals scintillate in the ultra-violet range. The emission spectrum of CaF_2 is shown in Fig. 2.2.a. The PMTs used in CANDLES have an effective range in the visible spectrum, as shown in Fig. 2.2.b. To overlap the CaF_2 emission spectrum with the effective range of the PMTs, we use a wavelength shifter(WLS). The WLS surrounds the CaF_2 crystals with a thickness of 5 mm. The main component of the WLS is bis-MSB which has a concentration of 0.1 g/l. The absorption and emission spectrum of the WLS is shown in Fig. 2.2.c. Solid line is the absorption spectrum and the dashed line is the emission spectrum.

The 96 CaF_2 crystals are arranged into 6 layers, with each layer containing 16 crystals. The arrangement of the first layer is shown in Fig. 2.3. The distance between the crystals is 10 cm. By design, crystal #11 has higher contamination than the other crystals. This crystal is used in various calibrations. Due to its high background event rate, this crystal is completely ignored in the analysis of this study.

2.1.2 Photo-Multiplier Tubes(PMTs)

CANDLES experiment uses three types of PMTs. They have differing diameters at 20 inch, 13 inch and 10 inch. The characteristics of the PMTs are shown in Table 2.1. The top and the bottom of the detector are installed with the 20 inch PMTs. The inner side of the detector is lined with 13 inch

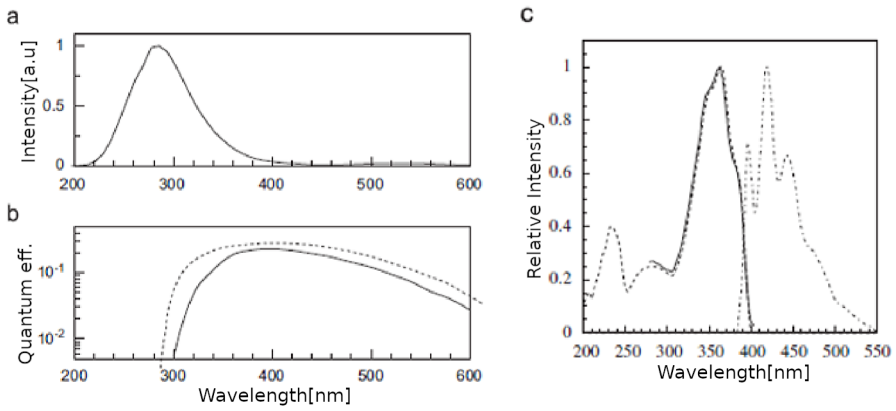


Figure 2.2: a. Emission spectrum of CaF_2 is shown. b. Quantum efficiency of the PMTs used in CANDLES is shown. Dashed line is for the 20 inch PMT. c. Absorption and emission spectrum of the WLS. The main fluorescing component is bis-MSB. Solid line shows the absorption spectrum and the dashed line shows the emission spectrum.

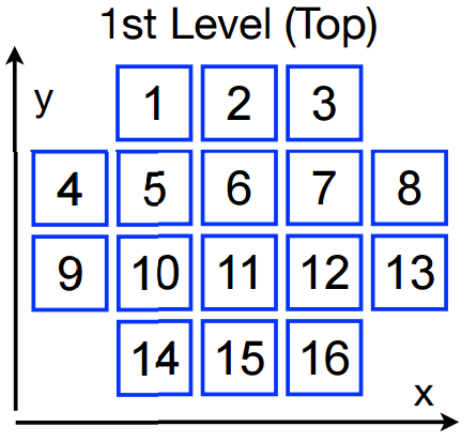


Figure 2.3: CANDLES crystals arrangement for the first 16 crystals. These crystals make the first layer of crystal. All 96 crystals are arranged in this configuration and into 6 layers. Layer 1 is at the top of the detector.

Product Name	R7250	R8055	R7081
Diameter	20inch	13inch	10inch
Photocathode material	Bialkali	Bialkali	Bialkali
Dynode	10 stage Box&Line	10 stage Box&Line	10 stage Box&Line
Spectral Response	300 to 650nm	300 to 650nm	300 to 650nm
Peak Wavelength	420nm	420nm	420nm
Gain	5 x 10^7 at 2kV	5 x 10^7 at 2kV	5 x 10^7 at 1.5kV

Table 2.1: CANDLES PMTs characteristics.

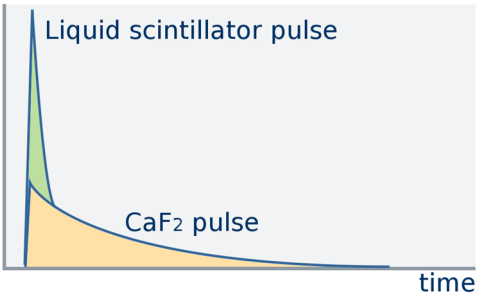


Figure 2.4: CaF_2 and LS event pulse shapes. The LS has a much shorter decay time and produce a sharply rising pulse shapes. CaF_2 crystals produce long tailed pulse shapes.

PMTs. However, the very top layer of PMTs on the detector's side wall is installed with 10 inch PMTs.

2.1.3 Liquid Scintillator

The inner acrylic tank of the detector is filled with a liquid scintillator(LS). The decay time of the liquid scintillator is much shorted than that of the CaF_2 crystal. An example curve is shown in Fig. 2.4. The decay time of the CaF_2 crystal is on the order of few μs whereas the LS decay time is a few tens of nanoseconds. If an event deposits energy in the LS, we can reject these events based on their pulse shape. Events originating from outside the CaF_2 crystals are likely to deposit energy in the LS and thus the pulse shape is used to reject external background events. CANDLES uses a hardware based rejection system for LS events and those that survive this hardware rejection are detected during analysis with pulse shape discrimination methods. Details of both of these methods will be shown in a separate section.

The scintillating and fluorescing components of the liquid scintillator are 2,5-Diphenyloxazole(PPO) and bis-MSB, which have concentrations of 1.0 g/l and 0.1 g/l, respectively. The PPO has an output spectrum that peaks

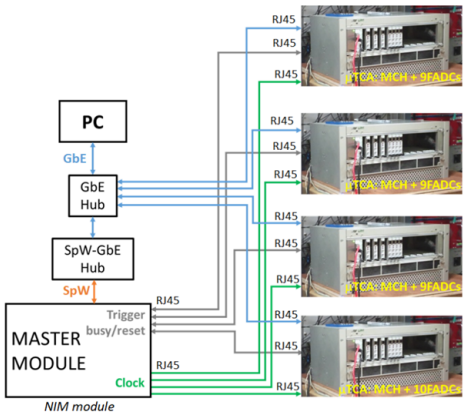


Figure 2.5: CANDLES main hardware. There are 4 μTCA modules with each module housing 9 or 10 ADC units. All modules are connected to a master module to receive trigger instructions.

at 385 nm. These components are mixed into a solvent of Paraol250(80%) and Pseudocumene(20%). The photon yield of LS is slightly less than that of CaF_2 crystals. If an event deposits energy in the LS, the extracted event energy is slightly less than that of CaF_2 events.

2.2 Detector Hardware

2.2.1 Data Acquisition System

CANDLES detector collects event information from all 62 PMTs and transfers the signal through a BNC cable to amplifier modules. The BNC cables have a length of approximately 30 m. CANDLES data acquisition system(DAQ) uses 4 μTCA modules with each module containing 9 or 10 AMC-FADC units. AMC-FADC stand for advanced mezzanine card - flash analog to digital converter. The main function of the FADC units are to convert analog PMT signals to digital signals for recording on a computer. All μTCA units are connected to a master module as shown in Fig. 2.5.

When trigger request is received from one module, the master module sends out trigger signal to all μTCA crate. The master module also connects to recorder computer through Spacewire-to-ethernet hub and can signal the start of an event. The event signal is sent from each μTCA crate to the recorder computer through a gigabit ethernet cable. Each of the μTCA crates have 2 event-parallel buffers that can be read by the computer in parallel. This gives us a trigger inefficiency of less than 10^{-9} and an acceptable trigger rate of 200 cps. The current CANDLES event rate is about 10 cps.

The CANDLES DAQ system records an event for a duration of 8.96 μs. However, we record event pulse at two different time resolutions. The first

768 ns of an event is recorded by 384 channels, with each channel recording the total event signal over a 2 ns duration. The next 8.2 μs of the event signal is recorded by 127 channels, with each channel recording the total event signal over 64 ns. An example of this changing of timing resolution in event pulse can be observed in Fig. 3.5.c. The information in each channel is stored as an 8 bit integer. The reason for splitting the event signal into these channels is due to disk space considerations. At current event rate, CANDLES records over 20 GB of data per day.

2.2.2 Dual Gate Trigger

The main trigger policy of CANDLES is the Dual Gate Trigger(DGT)[16]. It is designed to remove liquid scintillator events and only keep CaF_2 events at the hardware level. Figure 2.6.a shows possible energy depositions by various particles in CANDLES. Particles originating from outside a CaF_2 crystal are likely to deposit energy in the LS. The pulse shape of these events have a very short decay time. Most of the event's signal is contained within the first hundred nanoseconds. Some particles may also deposit energy in both the LS and the CaF_2 crystals. In these cases, the event's pulse shape has a sharply rising component as well as a long decay time component. Finally, α and β events that are generated in CaF_2 crystals completely deposit all of its energy in the CaF_2, generating a long decay time pulse. It's also possible for γ particles generated inside CaF_2 crystals to escape the crystal. These types of events generate a CaF_2 + LS pulse shapes.

To reduce events with LS pulse component, the DGT takes integrals of an event pulse at two different ranges, as shown in Fig. 2.6.b. The first gate has an integration duration of 128 ns from the start of an event. The second gate has an integration duration of 168 ns and it starts 384 ns after the start of an event. If an event deposits all of its energy in the LS, then the signal in the first gate is non-zero while the second gate has no signal. Therefore, the two gates have zero coincidence. On the other hand, if the event is purely CaF_2 crystal event, then both gates have signal and coincidence is observed. The DGT trigger only activates if it detects signal in both gates. This trigger system reduces the trigger rate to 4%. More information about CANDLES DAQ system can be found in [17].

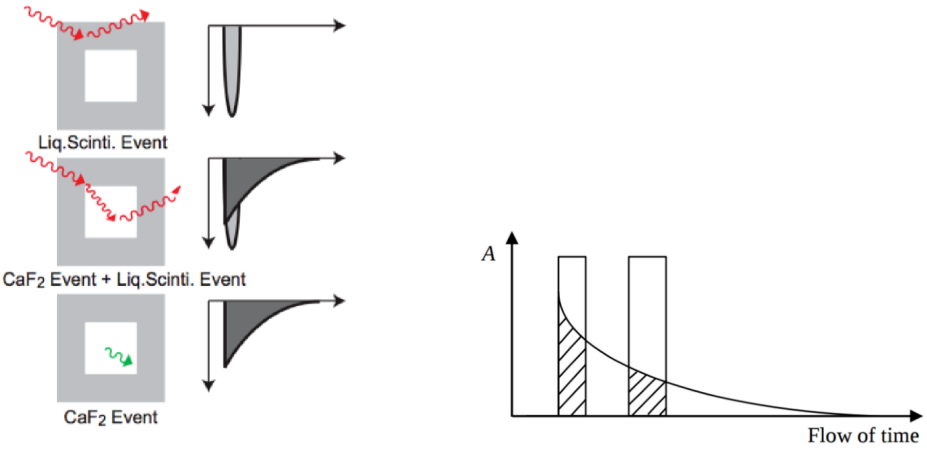


Figure 2.6: a. Possible event signals in CANDLES detector. Events outside the CaF_2 crystals deposit energy in the LS. This creates a sharp rise and sharp decay events that only range up to 200 ns. CaF_2 only events generate long tailed pulse shapes and can range to 4 μs. Some events can deposit energy in both LS and CaF_2 crystal, generating a mixed pulse shape. b. The first and second gates of the dual gate trigger system. The first gate has a duration of 128 ns from the start of the event. The second gate has a duration of 168 ns and start 384 ns from the start of the event. CaF_2 only events are selected by hardware if both gates have non-zero signal.

Chapter 3

CANDLES Data Process Methods

CANDLES detector records all PMT data individually with 500 channels(section 2.2.1). From the raw data, various parameters are extracted, such as energy, position, pulse shape indicator etc. In this section, the parameters that were used in this study are presented. These parameters are obtained as part of CANDLES data processing methods.

3.1 Timing Correction

Pulse shape discrimination(PSD) is a widely used method in physics to veto various events and backgrounds. The first step in PSD is to obtain the summed event pulse. CANDLES uses 62 PMTs to collect event signals. However, the start channels of these PMTs are not synchronized. Figure 3.1 shows a 140 channel(280 ns) region near a start of an event. Each band in the figure represents signal received at each PMT. As we see here, the start of an event can be in very different channels for different PMTs.

To apply start channel timing correction, we collect LS events at the center of the detector. Because LS has fast rise time and short decay time, majority of the photons emitted arrive within 100 channels from the start of an event. Therefore, the photon statistics is good for LS event to determine the start channels of different PMTs. We select PMT2 as the reference PMT and every other PMT's signal is synchronized with PMT2's start channel. The start channel difference between any PMT and PMT2 is collected and we fit a Gaussian function to obtain the correction offset. An estimate of correction offset for one PMT is shown in Fig. 3.2. The resulting modification to the summed pulse shape due to start channel correction is shown in Fig. 3.3.

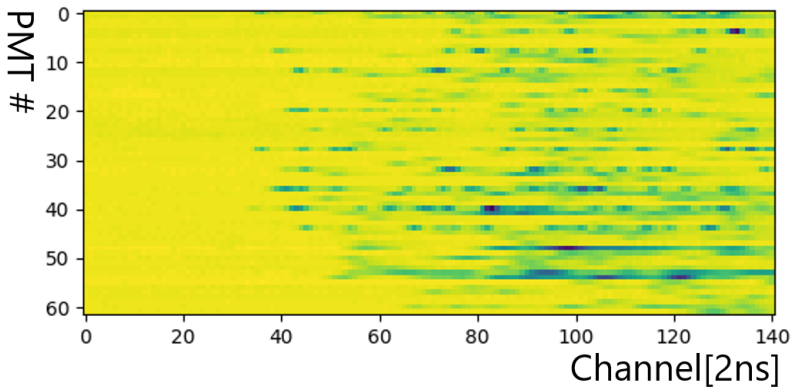


Figure 3.1: A single event signal as received by each PMT. The y axis is the PMT number and x axis is the channel number. Each channel represents a 2 ns signal. Each band in the figure represents the signal of an event at that PMT. The signal intensity received by each PMT is dependent on the location of the event. We see here that the start of the event is not synchronized between different PMTs.

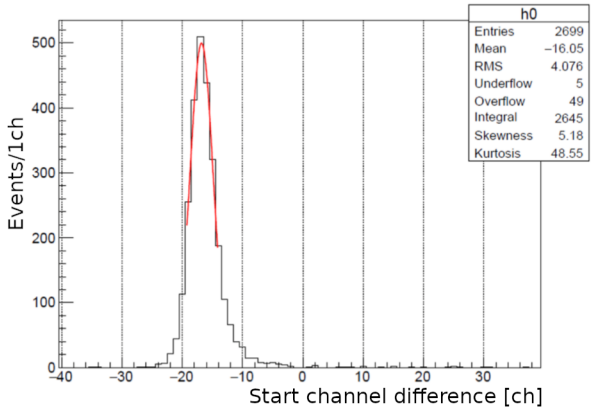


Figure 3.2: Start channel difference between PMT2 and another PMT. The start of an event is calculated independently for each PMT. Then the start channel of any other PMT is subtracted from start channel of PMT2. The resulting distribution is fitted by Gaussian and we obtain the start channel correction for that PMT with respect to PMT2.

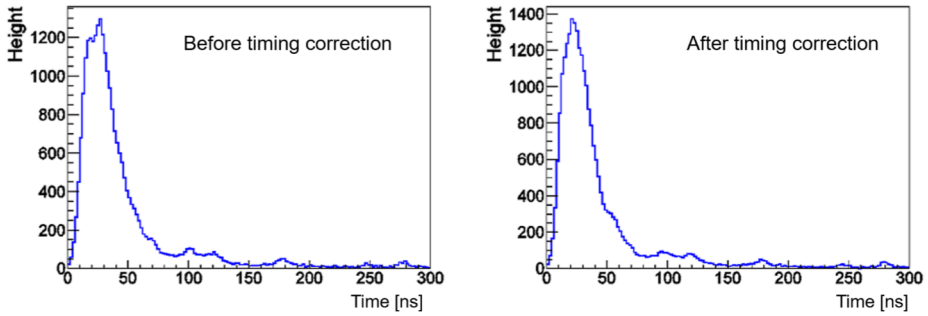


Figure 3.3: Example of applying start channel correction. The summed pulse of all PMTs of an event has a better shape after the start channels are corrected for.

3.2 Pedestal

Each PMT has a reference baseline(pedestal) against which the signal is recorded. Figure 3.4 shows the initial 382 channels of a single PMT signal. For this PMT, the pedestal value is set to 244.5, which causes the oscillation between values of 244 or 245. FADC units record values as integers. The sudden drop in analog-to-digital converter(ADC) values represent arrival of single photons at the PMT. Arrival of single photons is a discrete process. By summing the signals of each PMT, we obtain a more continuous pulse shape, as shown in Fig. 3.5.a. The large variance in the initial 382 channels is due to the different timing resolutions of each channel set by the DAQ(Sec. 2.2.1).

With the PMT signals summed, we see that the baseline value of the PMTs still remain in the summed pulse. The baseline is subtracted to obtain the event summed pulse shape, as shown in Fig. 3.5.b. To obtain the baseline value, we take the average of the first 30 channels of an event. Due to DAQ buffering, the first 60 to 70 channels do not contain event signal and we can use them to determine the event baseline level before an event. In this example, the summed pulse is of a γ decay by ^{208}Tl with energy 2.6 MeV.

3.3 Conversion to Photo-Electron Count

A photon arrival information is converted to electric signal by the PTM. This electric signal is converted to digital dimensionless unit by the FADC modules. We can convert the total ADC signal to total number of photo-electrons(p.e) using a conversion factor. We first estimate the integrated ADC values of single photons. To obtain single photon(1p.e) signals, we trigger the detector on a clock at 3 Hz and check for photon signals. A

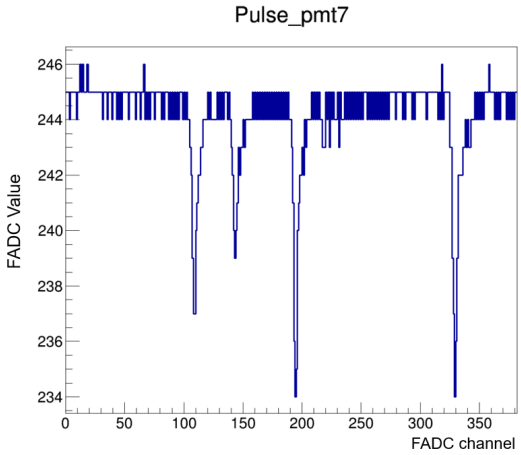


Figure 3.4: Example of the first 384 channels of a single PMT. The output of the ADC units have a baseline(pedestal) of around 245. Each sudden dip in the signal represents the arrival of a single photon.

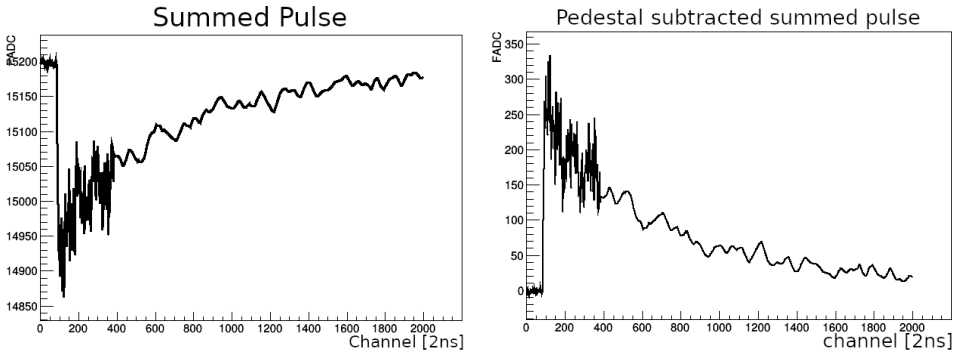


Figure 3.5: a. Example of summed pulse shape obtained by combining the signal of each PMT into a single pulse. The baseline levels of the PMTs are not subtracted. b. Only the event pulse shape is obtained by removing the PMTs pedestal levels. The change in fluctuation level at channel 384 is due to the different time resolution setting used by the DAQ.

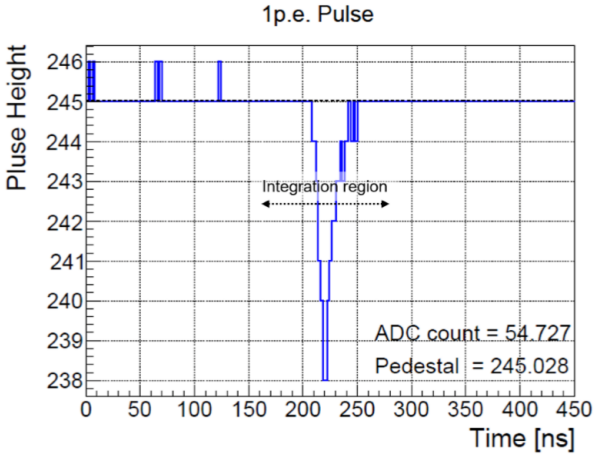


Figure 3.6: An example of a single photon signal recorded by a single PMT. Note here that the single photon signal is longer than 1 channel. The total ADC signal is obtained by integrating ±60 ns(30 ch) from the start of the pulse.

photon signal is detected if we find 2 adjacent channels that are 2 counts above the PMT baseline. An example of 1p.e signal is shown in Fig. 3.6. Once a 1p.e signal is detected, we integrate the signal around ±60 ns to obtain the total ADC count. To monitor PMT stability, we also obtain 0p.e events by integrating clock triggered events from 330 ns to 450 ns.

Figure 3.7 shows the total ADC counts of 1p.e and 0p.e events. The mean values of the 1p.e distribution is taken as the ADC to p.e conversion factor. The conversion factor is estimated for each PMT. With this, the number of photons generated by an event is obtained by applying the conversion factor to an event's total ADC signal.

3.4 Event Position Reconstruction

The 96 CaF_2 crystals are positioned into 6 layers and each layer has the structure as in Fig. 2.3. We can reconstruct the position of an event by the weighted average method given by the formula:

$$\vec{r} = \frac{\sum_{i=1}^{62} NPE(i) \times \overrightarrow{PMT(i)}}{NPE_{total}} \tag{3.1}$$

Here, $\overrightarrow{PMT(i)}$ is the position of the ith PMT and $NPE(i)$ is the total number of photo-electrons received at the ith PMT. This allows us to reconstruct the position of events from each crystal within the detector. We use ^{208}Tl events to obtain the positions of these events and produce Fig. 3.8. Around each cluster, we fit a Gaussian function along each axis to determine the

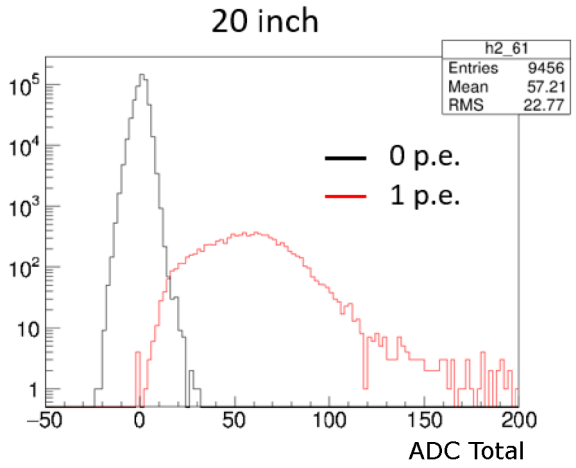


Figure 3.7: The distribution of 1p.e and 0p.e signals. The mean value of the 1p.e distribution is taken as the ADC signal to photon count conversion factor. The distribution is made for each individual PMT.

x,y,z positions of each crystal and also obtain the standard deviation of the Gaussians, as shown in Fig. 3.9.

Based on the crystal positions and their standard deviations, we can apply a *crystal cut* to select events from only a single crystal. For a clean selection, we often use a cut of 1.5σ or less. Figure 3.10 shows the standard deviation dependence of each crystal, calculated by ^{40}K γ events. The standard deviation values are energy dependent and must be estimated for different energies. For the deviation at higher energies, we used neutron calibration source to obtain data in the 5 MeV to 10 MeV. The details of the neutron calibration method is beyond the scope of this study and more information can be found in [18]. The results of the fitting is shown in Fig.

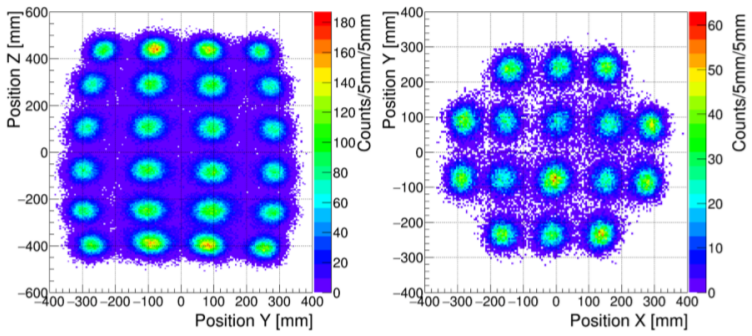


Figure 3.8: ^{208}Tl events position reconstructions by the weighted average method. Each of the clusters represent the position of a crystal.

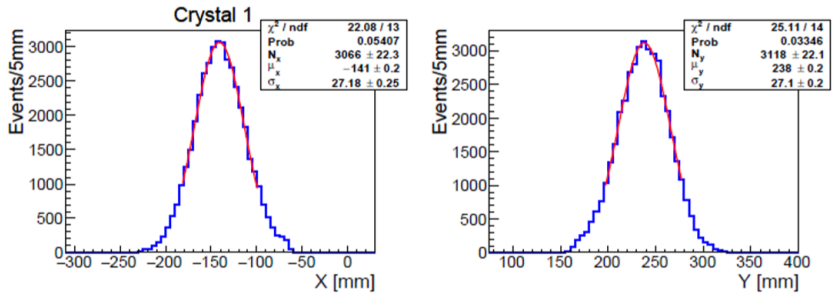


Figure 3.9: Gaussian fitting for crystal positions. The x and y distributions of events from crystal 1 are shown. By fitting a Gaussian, the position of each crystal can be obtained. The standard deviation is event energy dependent and must be determined based on the event energy.

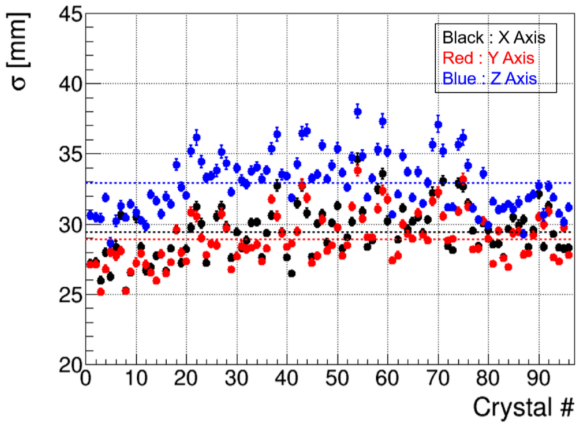


Figure 3.10: The standard deviations of x,y,z positions of each crystal as estimated by ^{40}K events. The standard deviation is higher for crystals towards the center of the detector.

3.11. Using the fitted deviation values, we are able to apply crystal cut at any event energy.

3.5 Event Energy Reconstruction

Total number of photons generated by an event can be obtained using the 1p.e conversion factor(section 3.3). Because the photon yield is different for each crystal, photon count to energy conversion factor must be calculated per crystal. Thus, the energy of an event can be calculated with the following

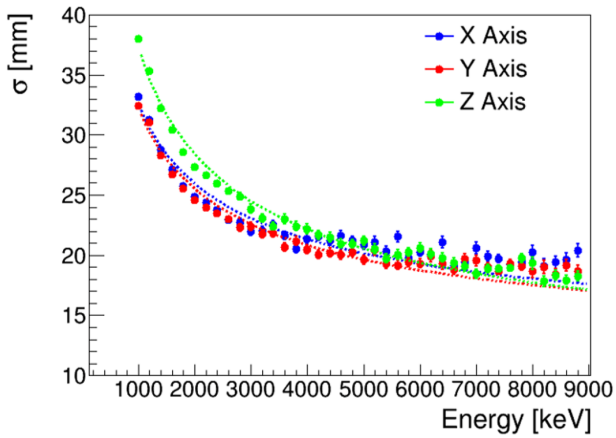


Figure 3.11: Position standard deviation obtained by fitting. For any event energy, we can apply a cut based on the deviation from each crystal center. Typically, we use a cut of 1.5σ to select events in one of the CaF_2 crystals.

equation:

$$Energy = F(i) \sum_{pmt=1}^{62} NPE(pmt) \tag{3.2}$$

where $NPE(pmt)$ is the number photons counted in a that PMT and $F(i)$ is the energy conversion factor of crystal i. The conversion factor is calculated for each crystal using an ^{88}Y source. The decay scheme of ^{88}Y is shown in Fig. 3.12. It is a γ source that emits primarily at 900 keV and 1.8 MeV.

The ^{88}Y is an external source that we introduce to the detector during calibration. It is placed inside a small capsule and lowered by cable into the detector through calibration ports. There are six ports that we use to position the source. The source is initially lowered to between the top two crystal layers and we take calibration data. The source position relative to the layers is shown in Fig. 3.13.a. We then lower it further to between the 3rd and 4th layer, followed by 5th and 6th layers. Once we take data for all layers, the source is inserted into the next port. The position of the ports relative to the crystals is shown in Fig. 3.13.b as red dots. The 1.8 MeV γ events absorbed by the CaF_2 crystals are used to calculate the photon count to energy conversion factor.

Figure 3.14.a shows a typical photon count of events in a single crystal taken during ^{88}Y calibration run. By fitting for the 1.8 MeV peak of the calibration source, we can generate the conversion factors. The conversion factor is calculated for each crystal and the mean values are shown in Fig. 3.14.b. As we see here, the conversion factor has crystal dependency. We can also estimate the standard deviation from the ^{88}Y calibration data but the deviation is also energy dependent. It must be estimated by referencing

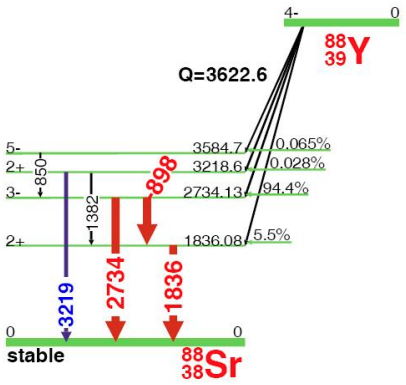


Figure 3.12: ^{88}Y decay scheme. It emits a nearly simultaneous 900 keV and 1.8 MeV γ particles. We use the 1.8 MeV γ as an energy calibration source.

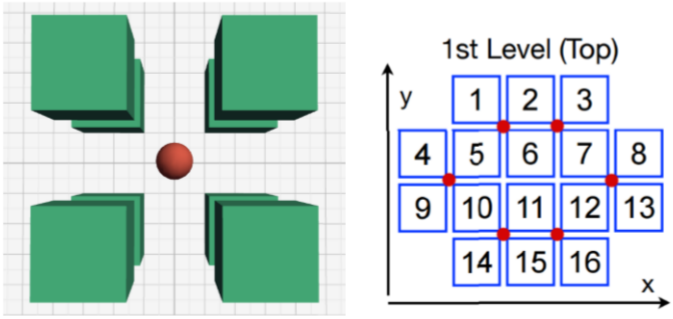


Figure 3.13: a. The ^{88}Y source is suspended by cable between layers of crystals. At each position, the source has 4 closest crystals that receive majority of events. b. The port locations relative to the crystal geometry is shown as dots. There are 6 ports and depending on the port location, each crystal in the detector receive significant number of ^{88}Y γ events.

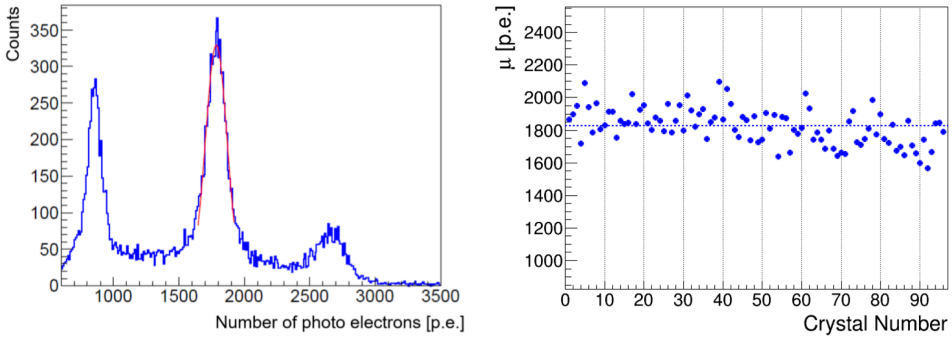


Figure 3.14: a. The photon count spectrum is shown for ^{88}Y calibration data. The 1.8 MeV peak is fitted with Gaussian to obtain the energy conversion factor. b. The fluctuation of the mean value of 1.8 MeV ^{88}Y events is shown for each crystal. The mean value fluctuates due to the differing photon yield of each crystal.

other events at different energies.

We define the energy resolution of our detector as:

$$resolution = \frac{\sigma}{\mu_{energy}}, \tag{3.3}$$

where μ_{energy} and σ are the mean value and standard deviation of the energy of an event distribution, estimated by a Gaussian fitting. The resolution is estimated using various events at different energies and the fitting is shown in Fig. 3.15. The ^{40}K and ^{208}Tl events are from natural backgrounds and are present in regular CANDLES data. The high energy events are obtained using the (n, γ) calibration method. The details can be found in [18]. From Fig. 3.15, we can see that the energy resolution improves with energy. At Q value, the CANDLES energy resolution is expected to be 2.4%.

3.6 Pulse Shape Discrimination

The scintillating properties of CaF_2 crystals have a slightly different decay times for α and β events and produce different summed pulse shapes. Figure 3.16 shows the averaged pulse shapes of β and α events. We can distinguish between α and β events by fitting reference pulses. This is called the pulse shape discrimination(PSD) method. Shown in Fig. 2.4, LS events generate a very different pulse shape. By creating an LS reference pulse, we can also reduce CaF_2 + LS type events. Purely LS events are removed from data by the dual gate trigger system.

We use ^{215}Po events to generate the reference α pulse. ^{215}Po events are present in the CaF_2 crystals as a result of contamination by ^{235}U. The decay chain ^{219}Rn $\rightarrow$ ^{215}Po $\rightarrow$ ^{211}Pb produces two α particles at energies

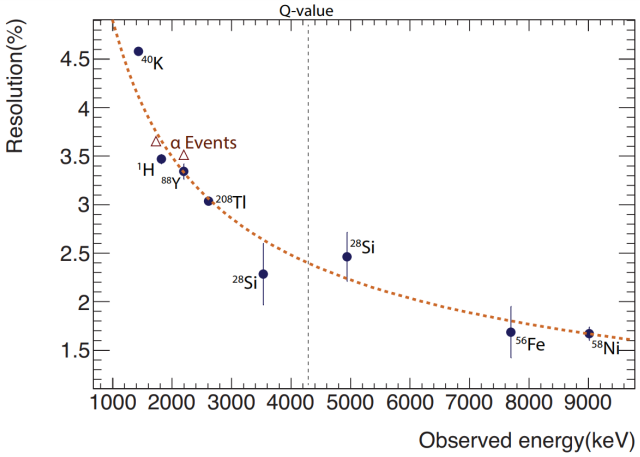


Figure 3.15: The energy resolution for CANDLES detector is obtained by fitting to various calibration events and natural background events. At Q value of ^{48}Ca, the energy resolution is expected to be 2.4%.

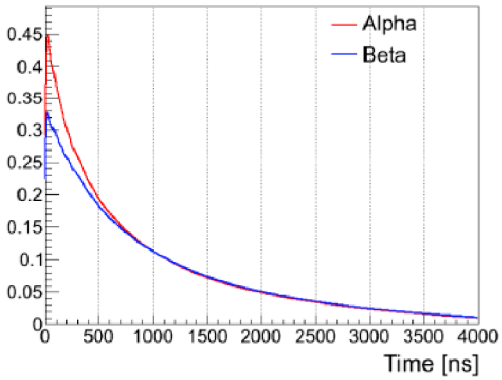


Figure 3.16: Averaged α and β event pulses are shown. The number of photons generated by α events in CaF_2 is slightly more than that of β events. The reference pulses are obtained by averaging α or β events in the detector. The reference pulses are fitted to any event in the data to estimate the likelihood of the event being either β or α event. There are no distinguishable difference between β and γ pulses.

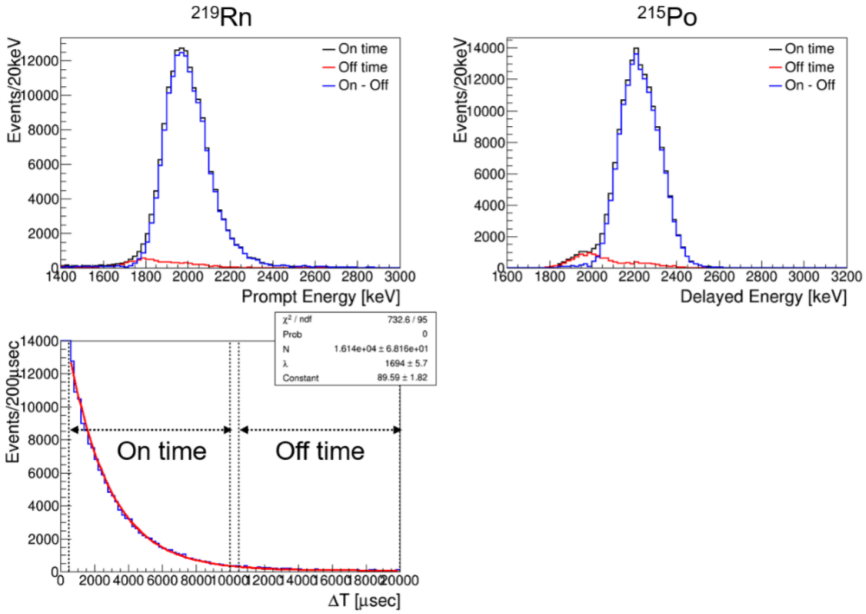


Figure 3.17: a.) Energy distribution of ^{219}Rn events selected by coincidence with α event of ^{215}Po within 20 ms in the same crystal. The energy distribution is not Gaussian because ^{219}Rn also emits γ particle, which can deposit energy in the LS. This shifts the peak due to slightly lower photon yield of LS. b.) ^{215}Po events selected by coincidence with ^{219}Rn event within 20 ms. c.) Time difference between the ^{219}Rn and ^{215}Po events. Events occurring after 10 ms are considered accidental(off-time).

of 6.8 MeV and 7.4 MeV. Due to the short half-life(1.78 ms) of ^{215}Po, we can detect these events by finding two α like events occurring within a short period. Due to quenching factor of CaF_2 crystals, the observed energies of the α particles are 2.0 MeV and 2.2 MeV.

To obtain ^{215}Po events, α event of either ^{219}Rn or ^{215}Po can be tagged first. We then collect events with expected energy in the same crystal over 20 ms. Due to the CANDLES event rate of about 10 cps, accidental coincidence with other events in this search is very low. Once both α candidates are found, we can plot the energies and their timing difference, shown in Fig. 3.17. Events occurring at more than 10 ms are considered accidental(off time) and events occurring under 10 ms are considered positive match(on time). We fit the timing difference curve for the half-life of the events, which give us a value of (1694 ± 6) μs, which is close to the known value of 1780 μs. As we see here, the resulting ^{215}Po distribution has low contamination of accidental events, thus providing a good selection of α events. These ^{215}Po events are then averaged to produce the alpha reference pulse.

To make the β reference pulse, we use ^{208}Tl events, which are γ particles

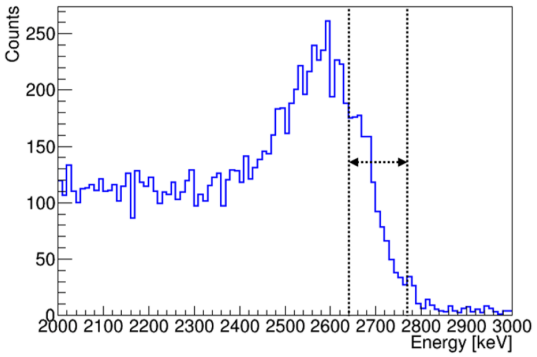


Figure 3.18: ^{208}Tl event selection for β reference pulse. These events originate outside the crystals and thus can deposit energy in the LS. Because photon yield in the LS is lower than CaF_2, the observed energy of CaF_2 + LS events is lower than the true value. To reject CaF_2 + LS events, we select events above the peak value, shown by the vertical bars.

with energy 2.6 MeV. There are no measurable differences between the γ and β pulse shapes in CANDLES. These ^{208}Tl events originate from outside the crystals and most of these events are removed from data by the dual gate trigger system. Some of these events deposit all of its energy in the CaF_2 crystals. We can use these events to obtain the β reference pulse. To avoid events that partially deposit energy in the LS, we only select candidate ^{208}Tl events above the mean value of the energy, shown in Fig. 3.18. Because the light yield of LS is less than the CaF_2 crystals, the real energy peak shifts lower due to energy deposit in LS. With this selection of events, we take the averaged pulse as the β reference pulse.

Because dual gate trigger system removes purely LS events from data, we use a different trigger strategy to make the LS reference pulse. Candidate LS events are selected within $\pm$200 mm of the center of the detector in order to collect enough number of photons in each PMT. We also select LS events above 300 keV to reduce fluctuations due to low number of photons generated. The selected candidate events are averaged to make the LS reference pulse.

The generated pulse shapes are fitted to event summed pulse to determine the event type. We found that fitting the event in the 500 ns to 4,000 ns range results in a better pulse classification, as shown in Fig. 3.19. This is due to the high fluctuation causing poor fitting in the first 250 channels(500 ns). The pulse shape is fitted in the 500 ns to 4,000 ns range and afterwards the first 500 ns of an event is compared to the extrapolated reference pulses. This comparison provides the likelihood that the event is either α or β event. The separation ability of the PSD method is shown in Fig. 3.20. PSD_β is the value obtained by fitting the β reference pulse and PSD_α is the value

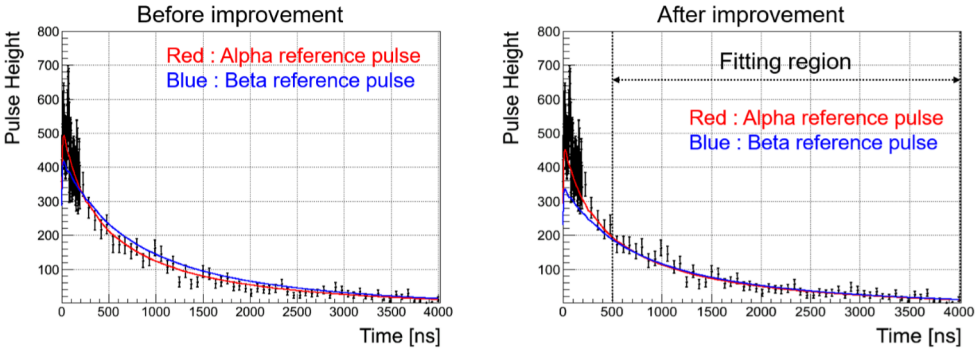


Figure 3.19: Example of reference pulses fitting. To first 500 ns has too much fluctuation and causes poor fitting, shown in left figure. The reference pulses are fitted between 500 ns and 4 μs. The initial 500 ns of the event is then compared(not fitted) to the reference pulse. This provides a better fitting result as shown in the right figure.

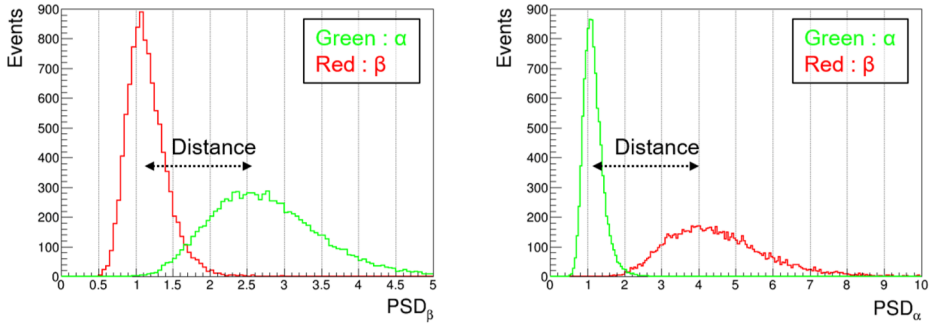


Figure 3.20: The PSD ability to separate α and β events is shown. In general, the PSD method is highly effective in separating α and β events.

obtained by fitting the α reference pulse.

3.7 Shape Indicator

To separate α and β pulse shapes, we also use Shape Indicator(SI) parameter. The SI parameter compares α and β reference pulses per channel. It can be characterized by the formula:

$$SI \propto \sum_i \frac{e_{\alpha,i} - e_{\beta,i}}{e_{\alpha,i} + e_{\beta,i}} f_i \tag{3.4}$$

where f_i is the value of the ith channel of an event's pulse shape and $e_{\alpha,i}$, $e_{\beta,i}$ are the ith channel value of the reference pulses. Figure 3.21 shows the difference in α and β events by SI value. There are no differences between

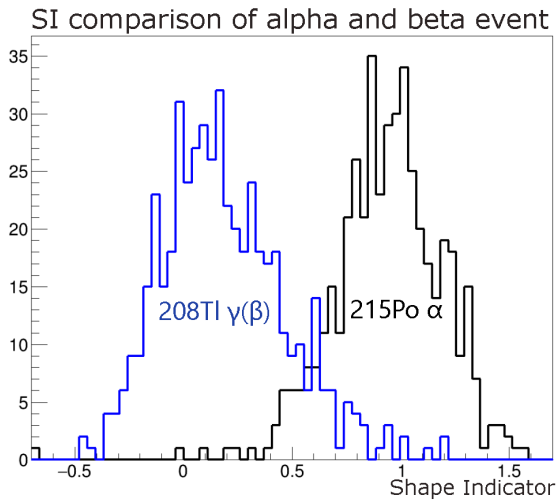


Figure 3.21: Comparison of β and α particles by SI is shown. In this case ^{208}Tl γ particles and ^{215}Po α particles are shown. There are no differences between β and γ particles. The β events are centered at 0 and α events are centered at 1.

γ and β pulse shapes and thus ^{208}Tl γ particles distribution is shown to represent β events. We scale the SI values such that the β events are centered at 0 while α particles are centered at 1. The SI values are based on the Gatti parameter [19] and more info can be found in [20].

Chapter 4

Reduction of Multi-Crystal ^{208}Tl Backgrounds

4.1 ^{208}Tl Multi-Crystal Events

^{208}Tl is a naturally occurring background and has a Q value of 4.99 MeV. It can occur within the CaF_2 crystals or from other components within the detector. The nuclei is produced by α decay from ^{212}Bi with a half-life of 3 minutes. The decay scheme of ^{208}Tl is shown in Fig. 4.1. It produces several γ particles, with the highest energy of these at 2.6 MeV. The majority of these ^{208}Tl decay events are removed by the dual gate trigger system that effectively reduces background events that deposit energy in the liquid scintillator. However, if the decay occurs inside one of the CaF_2 crystals and majority of the energy is contained within the crystal, then these events can not be reduced by the dual gate trigger system and need to be removed by some other method. Since the decay deposit events with energy near the Q value of $0\nu\beta\beta$, it is considered one of the main backgrounds of CANDLES.

The majority of the ^{208}Tl events are rejected by coincidence of ^{208}Tl candidate events with the preceding high energy α event from parent ^{212}Bi in the same crystal. This is known as the delayed coincidence method. However, due to the high energy of the γ emissions, particularly the 2.6 MeV γ that is always emitted, the probability of the γ particle escaping a crystal with significant energy is high. In such a case, an escaped particle can deposit energy in a neighboring crystal, as shown in Fig. 4.2. This poses a problem because if the escaped γ particle have a high enough energy, then the event position will be determined to be in a different crystal than the parent ^{212}Bi. We can not use the delayed coincidence method to veto these events. In this chapter we develop a method to distinguish these multi-crystal events from regular events.

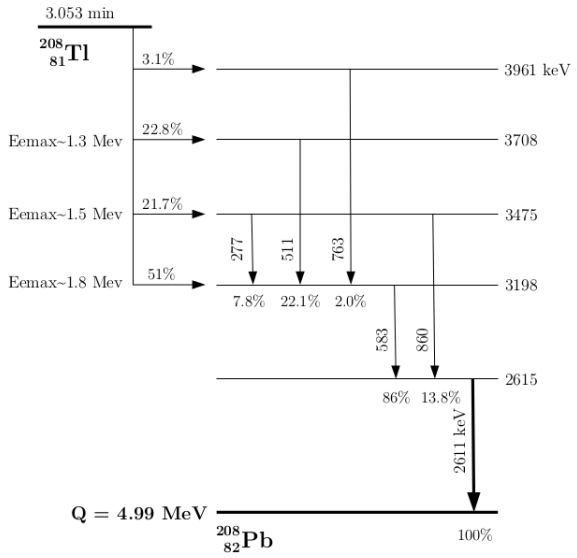


Figure 4.1: The decay scheme of ^{208}Tl is shown. ^{208}Tl has a Q value of 4.99 MeV and produces several γ particles. A 2.6 MeV γ is always produced in the decay, which has the highest energy among the decayed γ particles.

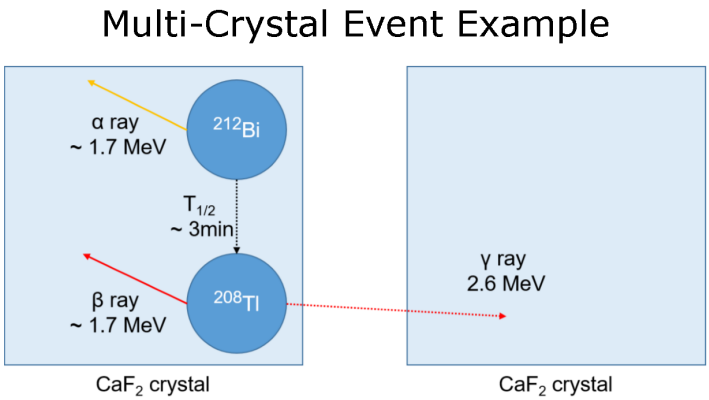


Figure 4.2: Possible way that a multi-crystal event can occur in data. The decay of ^{208}Tl produces several γ particles and a β particle. Because the γ particles emitted from ^{208}Tl have high energies, they can escape the original crystal and deposit energy in the next crystal or crystals. In this case, signal is received from multiple crystals. We can not find the preceding α event if the event position is reconstructed in the wrong crystal.

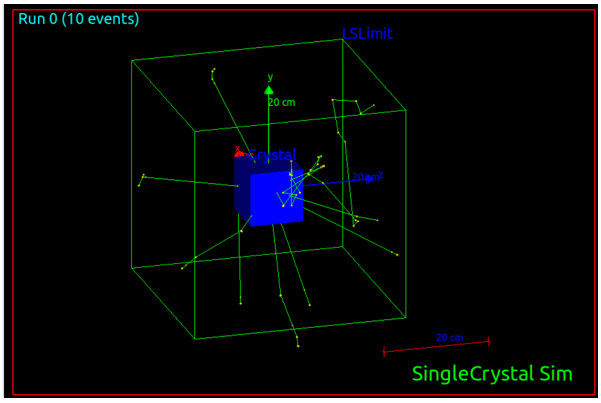


Figure 4.3: Simulation of CaF_2 crystal and surrounding LS to determine the escape probability of γ particles.

4.2 Simulations

As a first step, we carried out various simulations to understand multi-crystal event probabilities. The simulations were performed with Geant4[21]. For all simulations, we used the shielding physics list[22], which includes most high energy and low energy processes. The first simulation performed was to simulate the escape probability of the two highest energy γ emitted from ^{208}Tl. We simulated a single CaF_2 crystal placed in liquid scintillator as shown Fig. 4.3. The simulation includes the CaF_2 crystal, liquid scintillator, WLS and acrylic container with known physical parameters.

The two highest energy γ particles emitted by ^{208}Tl have 2.6 MeV and 1.094 MeV energies. For the simulation, we generated 1,000 γ particles at the center of the crystal and at every 0.5 cm from the center to estimate their escape probabilities. Figure 4.4 shows the estimated escape probabilities by simulation for the two highest energy γ particles. For each plot, three different probabilities are estimated. We examine what is the overall probability of the particle leaving the crystal with any amount of energy. This is shown by the top curve of each plot(blue curve). We also examine what the escape probability of the particle carrying 100% of its energy is. That is, the particle does not deposit any energy in the original crystal. The middle curve shows probability of escape carrying at least 75% of its energy.

We see in Fig. 4.4 that the 2.6 MeV γ have an overall 90% probability of escaping the crystal. It has more than 40% probability to escape the original crystal with all 2.6 MeV of its energy. This indicates that the ^{208}Tl events have a very high probability of creating multi-crystal events. Similarly, the 1.094 MeV γ also have a high probability of escaping the original crystal. It should be noted that although the γ particles escape with high energy, if the particles deposit any significant amount of energy in the liquid scintillator,

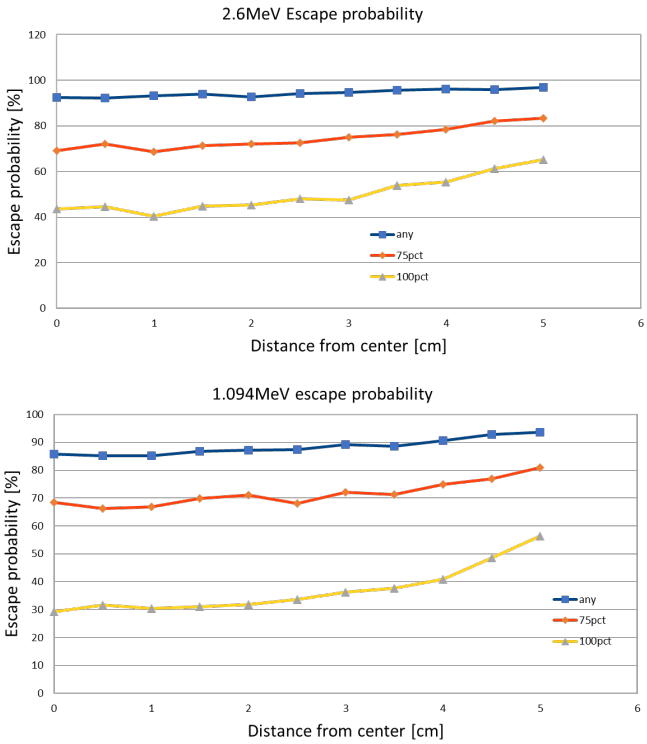


Figure 4.4: The escape probabilities of the γ particles are shown. Particles are generated at every 0.5 cm from the center of the crystal. The outgoing direction of the particles are random. For each position, 1,000 particles are generated. There 3 curves in each plot, representing probability to escape with 100% energy, 75% energy and any amount of energy. Simulation indicated a very high probability for γ particles escaping the CaF_2 crystals.

then these events would be automatically rejected by the dual gate trigger system. Therefore, we are only interested in cases where the particles do not deposit any significant energy in the liquid scintillator.

Next, we simulated a case where a crystal is completely surrounded by other crystals, as is the case for crystals toward the center of the detector. The configuration of the simulation is shown in Fig. 4.5.a. In this simulation, 5,000 γ particles were generated at the center of the central crystal with energy 2.6 MeV. We estimate how many crystals can receive energy from the γ particles. Figure 4.5.b shows the results of this simulation. The results show the number of crystals that receive energy in a single γ emission. We observe that even with the central crystal completely surrounded by other crystals, the 2.6 MeV γ still has a 25% probability to completely escape this configuration without any energy deposit. Only 45% of γ emissions produce single-crystal events. The probability of generating multi-crystal event by

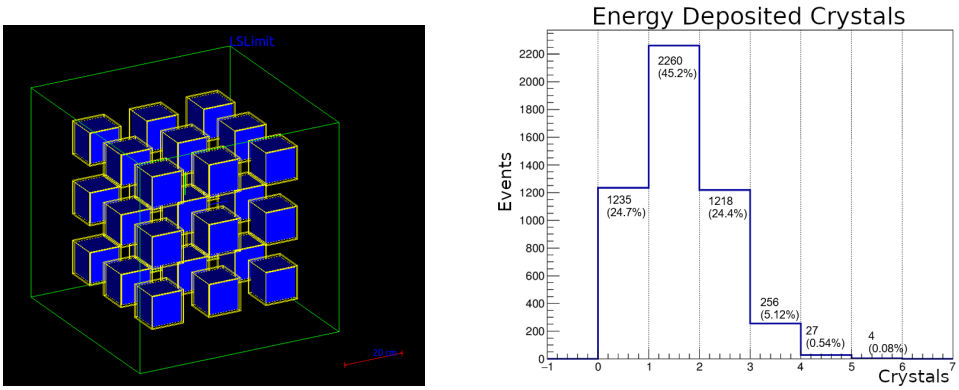


Figure 4.5: a. Configuration of the multiple crystal simulation. 2.6 MeV γ particles are generated at the center of the central crystal. We then count the number of crystals that receive energy from a single γ particle. b. Number of crystals that receive energy from a single γ particle generated in the central crystal. There is a 25% probability that no crystals receive energy. There is a 45% probability of single-crystal events. Probability of generating multi-crystal events is about 30%.

the 2.6 MeV γ is about 30% for this configuration. However, this simulation does not take into account that high energy deposit in liquid crystal events will be automatically removed. Also, different crystals configuration are not simulated. This is because in the current crystals configuration(Fig. 2.3), there are many different possible configurations of crystals that a single crystal can be surrounded by. This simulation configuration is the worst case scenario because other crystals configurations seen in CANDLES would typically have fewer surrounding crystals and therefore the escaped γ will find fewer crystals to deposit energy in. Therefore, the overall multi-crystal event probability will be less than 30%.

More extensive simulations were carried out but the results were inconclusive or unreliable. The reasons were due to the various cuts that we use to reduce backgrounds. Estimating how these cuts would affect the simulations was found to be difficult. We simulated a full CANDLES detector setup and used expected number of ^{208}Tl decays. However, difficulties arose when we tried to estimate how many events would be rejected by the dual gate trigger system, the crystals cut and the PSD cuts. Because CANDLES detector uses light pipe to funnel photons towards the PMTs, the geometry of the light pipe mirror surface must be known. The position of an event is reconstructed by the number of photons received by each PMT. This position reconstruction method can not be used without knowing how the light pipe funnels photons. Therefore, estimating the crystal cut became a very difficult issue. As a result, we moved to a more analytic approach to estimating and rejecting multi-crystal events.

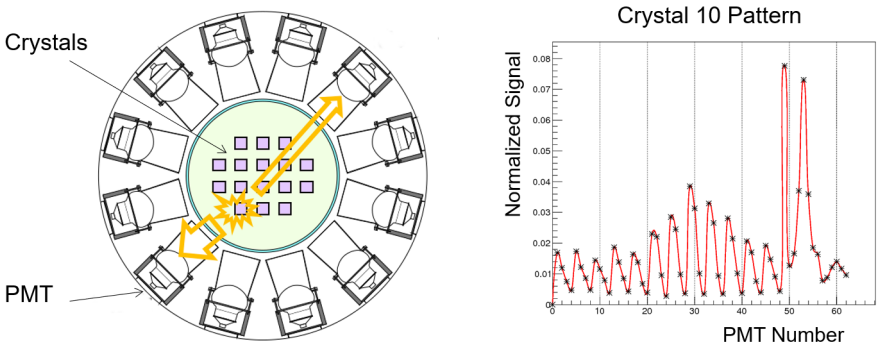


Figure 4.6: a. The total signal received at each PMT is dependent on the position of the crystal. The closest PMTs to a crystal receive maximum signal. b. This signal dependence on crystal leads to a unique PMT signature that we call hit pattern(HP).

4.3 Making Crystal Hit Patterns

CANDLES detector uses 62 PMTs completely surrounding the crystals to record event signals. The amount of signal received by each PMT is dependent on the crystal. The PMTs closest to an event crystal receives maximum signal while the PMTs furthest away receive the least signal, as shown in Fig. 4.6.a. Depending on the event crystals, unique total signal patterns are produced, as shown in Fig. 4.6.b. In this figure, the *hit pattern*(HP) of an event in crystal #10 is shown. When we change to a different event crystal, the total signal received at each PMT is different and thus a different HP is produced. By averaging over many events from the same single crystal, we can produce the HPs of all crystals. These crystal HPs can be useful because we can fit them to an event's observed signal and if a single crystal's HP has a very good match to the event's signal and every other crystal's HPs have poor match, we can say that it is a single-crystal event. Conversely, if an event's signal has similar match to multiple crystal HPs, then we can say it is a multi-crystal event. This is because if an event deposits energy in two crystal, then the event's signal will be a combination of two crystal's HPs.

To obtain the HPs of each crystal, we need to separately select known events in each crystal and average over the selected events. This is different from generating reference pulses for PSD(Sec. 3.6), although the same events are used to generate both HPs and reference pulses. In this study, we used the naturally occurring background ^{208}Tl's 2.6 MeV γ particles to make the HPs. To estimate the standard deviation, we also used the 900 keV and 1.8 MeV γ particles of our calibration source, ^{88}Y(Sec. 3.5). The selection criteria for the particles are shown in Table 4.1. The selection criteria used is

Particle	Crystal	PSD_β	Energy
^{88}Y 900 keV γ	$< 1.5\sigma$	$< 1.5\sigma$	(900 +/- 150) keV
^{88}Y 1800 keV γ	$< 1.5\sigma$	$< 1.5\sigma$	(1.83 +/- .15) MeV
^{208}Tl 2.6 MeV β	$< 1.5\sigma$	$< 1.5\sigma$	(2.6 +/- .15) MeV

Table 4.1: Event selection criteria to generate multi-crystal events.

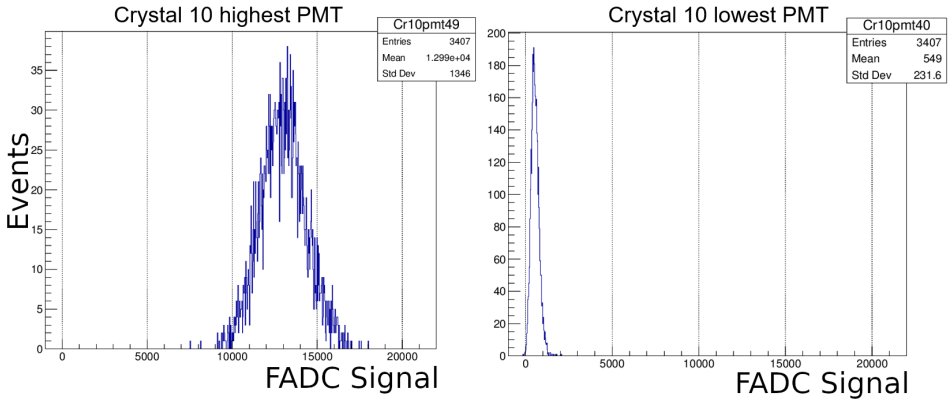


Figure 4.7: a. The signal received at PMT49 from crystal 10 events. For this crystal, PMT49 received maximum signal. b. Signal received at PMT40 from crystal 10 events. PMT40 receives least signal from crystal 10. We take the mean and deviation from each histogram for all PMTs to generate the hit pattern for that crystal.

not very strict in order to increase the number of events selected per crystal.

To make the crystal HPs, ^{208}Tl 2.6 MeV γ events were collected in each crystal and a histogram of the total signal received at each PMT were made. Figure 4.7 shows the PMTs that receive the highest and the lowest signal from crystal 10 events. In the case of crystal 10, PMT49 receives the largest portion of the total signal while PMT40 receives the least signal. From here, the statistical mean and the statistical standard deviations are extracted for each PMT. By collecting ^{208}Tl events in each crystal and creating histograms of observed signal in each PMT, the HPs of all crystals were made.

In principle, the PMTs are receiving photons from the crystals and therefore we can convert the HPs to photon counts with the ADC to 1p.e conversion factor. Since the HPs are obtained from the mean photon count at each PMT, we could estimate the standard deviation of each PMT signal in a HP by taking the statistical uncertainty, given by the square root of the mean photon count. However, this estimate of uncertainty falls much below the uncertainty observed by taking the statistical deviation of observed signal. Figure 4.8 shows the standard deviations in units of photo-electrons obtained by making HPs from the three sources shown in Table 4.1. This

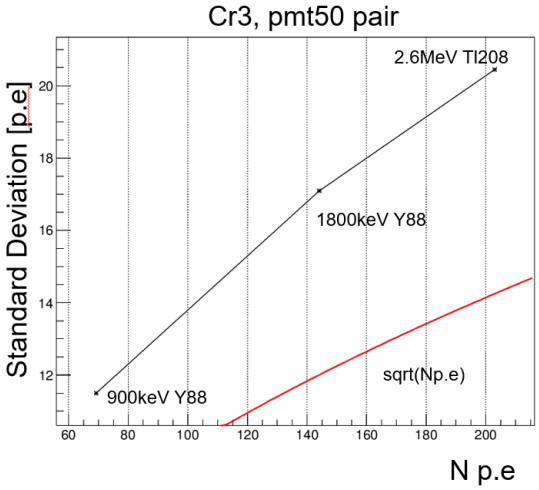


Figure 4.8: The red curve shows the square root of the number of photo-electrons received, which is the expected deviation. The black curve shows the calculated standard deviation using data. These standard deviations are calculated from making hit patterns from ^{88}Y and ^{208}Tl sources. Clearly, the observed standard deviation is higher than the statistical error.

figure was made by observing the number of photons collected by PMT50 from crystal #3 events and calculating the statistical standard deviation. The square root curve of the number of photo-electrons is also shown, which would be the expected deviation.

Clearly, the estimated deviation from data is higher than the statistical error. This is expected because although we are receiving photons from a single crystal, the events can have a large position variation within the crystal. The crystals have a size of 10 cm on each side, which causes the observed photon count at each PMT to vary significantly depending on the position of the event inside the crystal. This means that we can not use statistical error to estimate the standard deviation when we apply the HPs to events with different energy levels. Therefore, we have to estimate the standard deviations for events at different energy levels.

The estimation can be made by fitting a line to the three data points in Fig. 4.8. The fitting function used has the following form:

$$\sigma = A\sqrt{Np.e} + B \tag{4.1}$$

We are simply modifying the statistical error to fit the data points since the underlying error is proportional to the square root of the number of photo-electrons received. However, for crystal and PMT pair with low number of photo-electrons, a linear function may be used as an estimate. In practice, we fitted both a linear function and Eqn. 4.1 to each crystal and PMT pair

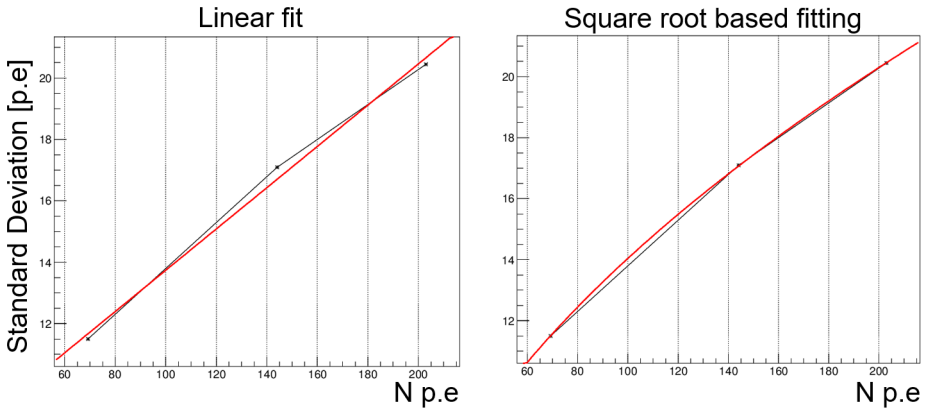


Figure 4.9: Two different fitting functions are used to estimate the deviation curve. The three data points are obtained from 900 keV and 1.8 MeV γ from calibration source ^{88}Y and 2.6 MeV γ from ^{208}Tl. For each crystal and PMT pair, the better fitting function is kept as the deviation estimate.

and kept the better fitting function as the standard deviation estimate. An example of the fitting is shown in Fig. 4.9. With fittings applied for each crystal and PMT pair, we can estimate the standard deviation of the HPs at any energy level. Ideally, we would need to use events with energy near the Q value of $0\nu\beta\beta$ to obtain a HP and to also estimate the errors but we do not have a reliable set of data at these high energies to do this. This is by design of the experiment because we used the $0\nu\beta\beta$ candidate with the highest Q value to specifically avoid any significant count of natural backgrounds. Therefore, we only used the three sources available to us.

At this stage we can perform a sanity check by applying the HPs and the fitted estimates of the standard deviations to the data we used to generate the HPs. Since the events used to generate the HPs are single-crystal events, they are expected to have very good fitting with a single crystal's HP and poor fitting with every other crystal's HP. To check this, we plot the χ^2 values of the best fitting crystal and the 2nd best fitting crystal, as shown in figure ref 4.10. The figure shows that the best fitting crystal HPs have reduced χ^2 centered around 1, as we expect. Also, since these are single-crystal events, we expect the 2nd best fitting crystal's χ^2 values to be poorer. We can see that this is indeed true with the second best crystal HP having about 3 times larger χ^2 values. We will see in the next sections that this difference between the χ^2 values are smaller for multi-crystal events.

We can also check the goodness of fit of applying the HPs to the ^{88}Y events with the extrapolated standard deviations. Figure 4.11 also shows the χ^2 values of the best fitting and the 2nd best fitting crystals of applying the HPs to ^{88}Y events. We again see that the extrapolated standard deviation estimates produce good grouping around the χ^2 value of 1 for the best fitting

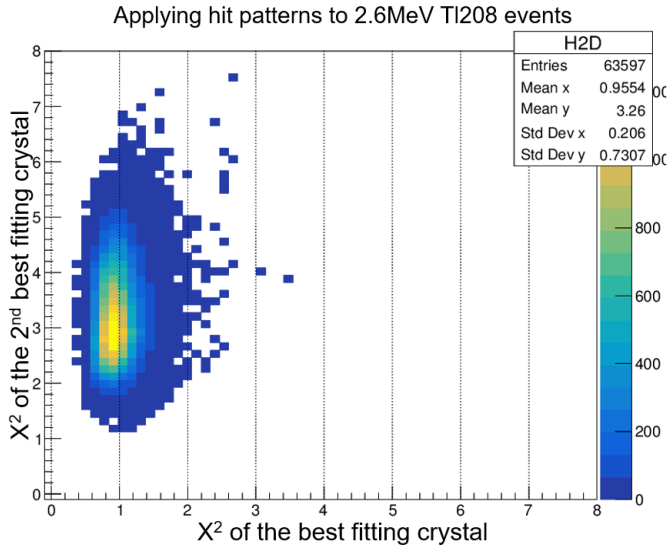


Figure 4.10: Application of the hit patterns to ^{208}Tl events. We plot the reduced χ^2 values of the best fitting and second best fitting crystals. The hit pattern correctly centers the χ^2 of the best fit crystals around 1 and the second best crystal's χ^2 is much larger than the best fit crystal's.

crystal HP. We also observe that the difference between the χ^2 values are much smaller than the ^{208}Tl case. This is especially true for the 900 keV events. This is expected because at 900 keV, the events generate less than 1,000 photons in the CaF_2 crystals and thus the statistical error is very high. As a result of large uncertainty, the difference between the χ^2 values is much smaller. We see an improvement for the 1.8 MeV events due to the increased number of photons generated. Finally, we see a tail in the ^{88}Y events that were not seen in the ^{208}Tl events. These are likely multi-crystal events because the ^{88}Y source produces a pair of γ particles nearly simultaneously and these can deposit energy in different parts of the detector.

We are yet to show the result of applying the HPs to multi-crystal events. This will be shown at the end of the chapter and application of the HPs to CANDLES recorded data will be presented in a separate chapter. The HP method is a natural way to check how much of each crystal is present in an event signal. The CANDLES crystals are positioned 10cm away from each other and each crystal has dimensions of 10cm. This means that the HP method must be able to resolve events from two neighboring crystals correctly while the separation distance is equal to the size of each crystal. Therefore, the main detriment to the HP method is likely to be the close distance between the crystals. Also, with larger distances between the crystals, particles escaping a crystal would have a larger probability of depositing energy in the liquid scintillator and thus higher probability of being

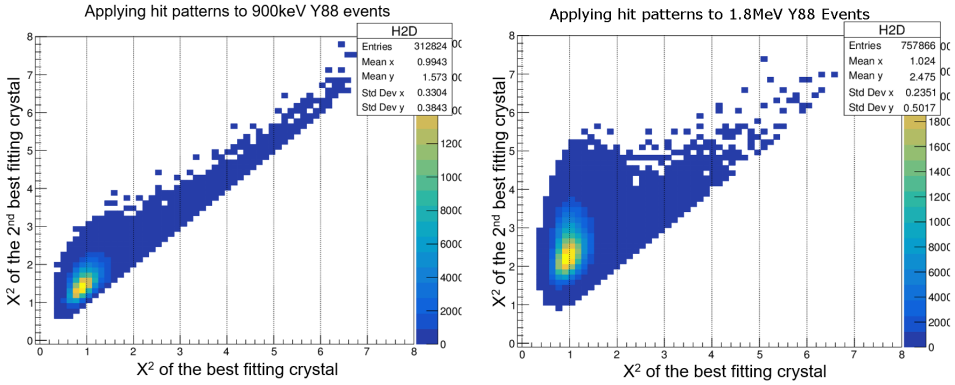


Figure 4.11: [Application of hit patterns to ^{88}Y events. The χ^2 values of the best fit crystal are again centered at 1. The difference $\Delta\chi^2 = \chi^2_{2ndbest} - \chi^2_{best}$ is much smaller than the ^{208}Tl case. This is expected due to the reduction in number of photons generated from ^{88}Y events.

rejected by the dual gate trigger system. This would decrease the number of multi-crystal events in observed data.

4.4 Making False Multi-Crystal Events

With the crystal HPs made, we need test the method on multi-crystal events. However, we do not currently have a method to reliably detect these events and thus we can not test the method on CANDLES data. But, we can generate multi-crystal events by taking events from two different crystals and summing the signal separately at each PMT. In a similar manner, we can also generate single-crystal events by taking two events in the same crystal and summing their signals at each PMT. Generating a multi-crystal event from two donor events is shown in Fig. 4.12. Here, we combined the total signal of two events, a 1.8 MeV ^{88}Y event in crystal #4 and a 2.6 MeV ^{208}Tl event in crystal #5, to generate a multi-crystal event. The donor events are selected using the same cut conditions in Table 4.1. The generated events thus has signal from two crystals. By taking two donors from the same crystal, we also generate single-crystal events.

The choice of the two donors are chosen for good reason. The total energy of the generated multi-crystal events is around 4.4 MeV. The energy distribution of multi-crystal and single-crystal events generated in this manner is shown in Fig. 4.13. These generated events cover the energy region of the Q value of CaF_2. Also, because one of the donors is the 2.6 MeV ^{208}Tl γ, this combination emulates the decay of ^{208}Tl that we wish to remove from data using the HP method. Therefore, these generated multi-crystal and single-crystal events act as a good benchmark for the HP method.

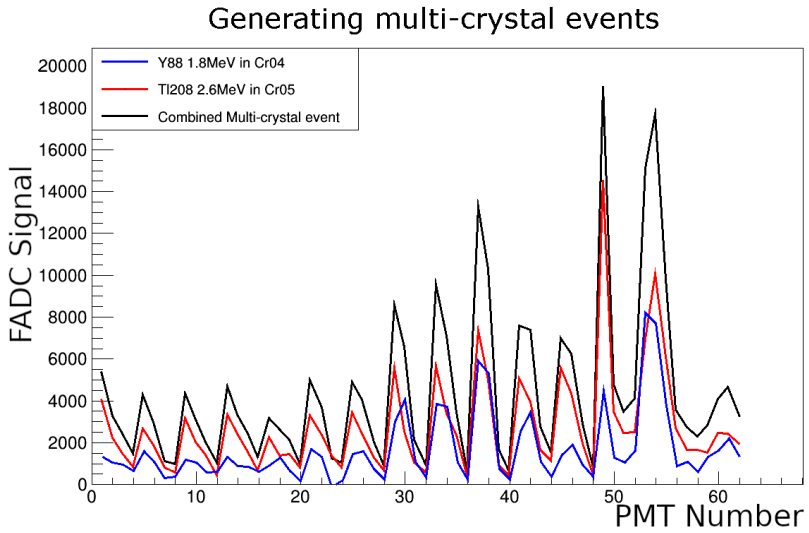


Figure 4.12: Example of generated multi-crystal event. We take the total signal of an ^{88}Y event in one crystal and add it to the signal of ^{208}Tl event in another crystal. The combined event is a 4.4 MeV multi-crystal event.

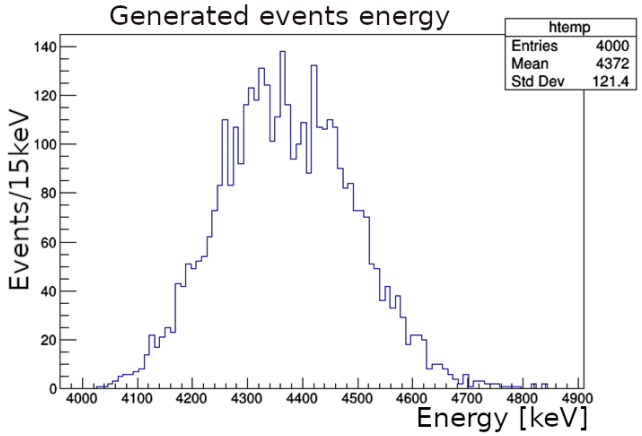


Figure 4.13: The energy spectrum of generated multi-crystal events. It covers the Q value region of ^{48}Ca and therefore is a good generated data to test the hit pattern fitting method.

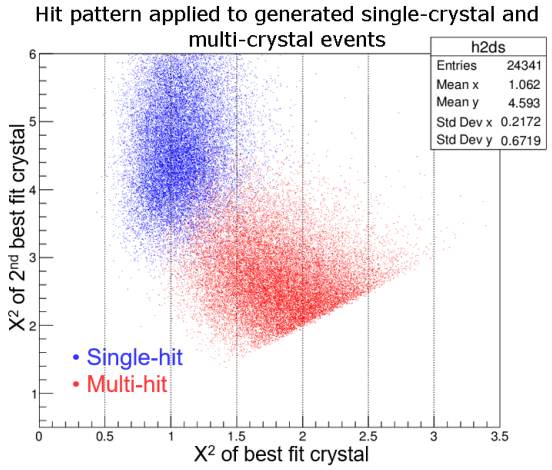


Figure 4.14: Hit pattern applied to generated single-crystal and multi-crystal events. Multi-crystal events have a small difference in $\chi^2_{2ndbest} - \chi^2_{best}$. The figure shows a good separation possibility of multi-crystal and single-crystal events by the hit pattern method.

We generated a total of about 25,000 single-crystal events and also 25,000 multi-crystal events to test the HP method. To create the multi-crystal event, the donor 1.8 MeV ^{88}Y event is first found. Next, we find the 2.6 MeV ^{208}Tl donor in one of the nearest neighbor crystals. Each crystal in CANDLES can have between 3 to 6 nearest neighbors. We generate 50 multi-crystal events for every crystal pair. When necessary, much larger number of events can be generated for any crystal.

4.5 Testing Pattern Fitting Method

With the multi-crystal and single-crystal events generated, we are in a position to test the HP fitting method. The application of the HP method to the generated events is shown in Fig. 4.14. To calculate the goodness of fit, the reduced χ^2 values are calculated using standard deviations extrapolated from fitting. We see that the extrapolated standard deviations approximate the uncertainty well, centering the single-crystal events' best fitting χ^2 around the value of 1. The difference between the χ^2 of the best fitting and second best fitting crystal HPs is also large for these events. We also observe that there is a good separation between the single-crystal and multi-crystal events. The multi-crystal events have a poor single crystal HP fitting and the difference in χ^2 values is also small.

To separate the single-crystal and the multi-crystal events, we can apply a cut based on the difference in the χ^2: $\Delta\chi^2 = \chi^2_{2ndbest} - \chi^2_{best}$. The multi-crystal and single-crystal events distribution along the $\Delta\chi^2$ axis is shown

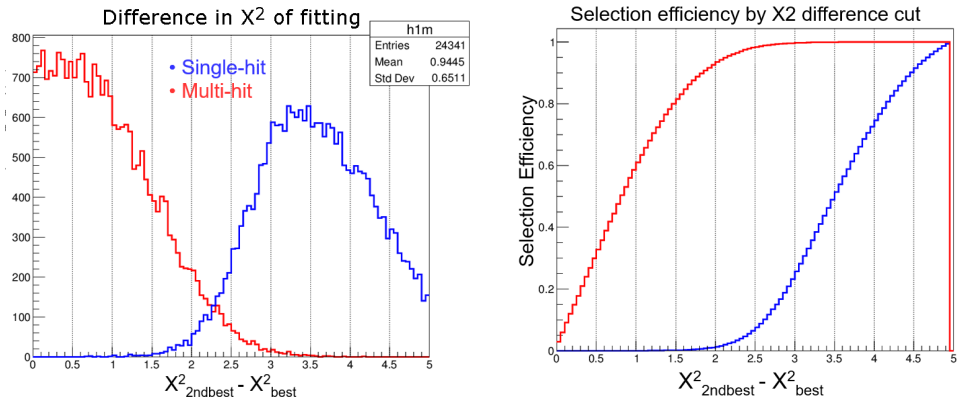


Figure 4.15: a. The multi-crystal and single-crystal events are plotted on the $\Delta\chi^2$ axis. b. Selection efficiency based $\Delta\chi^2$. A good separation is possible by applying a cut on $\Delta\chi^2$.

in Fig. 4.15.a. We see that a good separation is possible by applying a cut on $\Delta\chi^2$. The selection efficiency is shown in Fig. 4.15.b. For example, a cut of $\Delta\chi^2 > 2.5$ removes almost all multi-crystal events while reducing the single-crystal count by less than 10%.

Next, we want to estimate how many multi-crystal events survive CANDLES cuts. Because the donor events are already selected by a PSD cut, these generated events are unlikely to be reduced by PSD cuts. The major cut on multi-crystal events is the crystal cut, which removes events that are reconstructed to be outside the known position of a crystal. Because we've generated the multi-crystal and single-crystal events using real CANDLES event, we can also use CANDLES analysis tools to determine the position of the generated events. Therefore, we can also apply a crystal cut using the CANDLES analysis tools(Sec. 3.4).

The multi-crystal and single-crystal events surviving CANDLES position cut is shown in Fig. 4.16. We are selecting events within 1.5σ of a crystal center in this figure. Of the nearly 25,000 single-crystal events, only 1 single-crystal event does not survive the crystal cut. The multi-crystal events are reduced by 87.4% with the crystal cut. The crystal cut is effective because when we combine two donor events into a multi-crystal event, the position of the multi-crystal event is determined to be on some point between the positions of the original donors. Since the two donor pulses are of similar energy levels(1.8 MeV and 2.6 MeV) the position of the multi-crystal event is roughly at the mid point between the two donor event's crystals. Therefore, the crystal cut rejects a good number of multi-crystal events.

Visual inspection of Fig. 4.16 still shows a good separation between single-crystal and multi-crystal events, indicating that the HP method is potentially an improvement over using crystal cut to remove multi-crystal

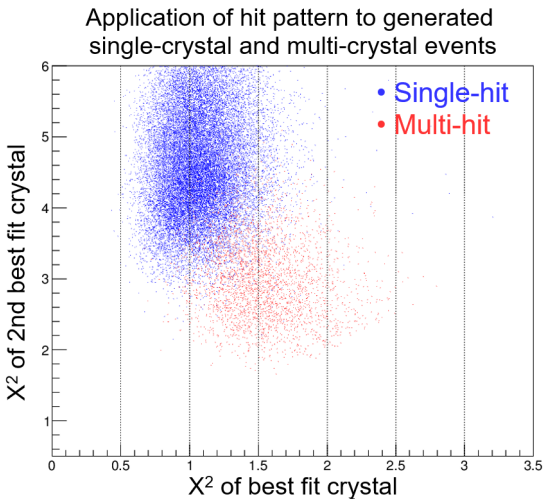


Figure 4.16: Application of hit pattern method after crystal cut on generated events. The multi-crystal events are reduced significantly by the crystal cut. Less than 15% of the multi-crystal events survive a crystal cut of 1.5σ. Compared to Fig. 4.14, the crystal cut surviving multi-crystal events are much closely centered to the single-crystal events distribution. Therefore, the separation efficiency is reduced.

events. The figure shows in general that the multi-crystal events that had larger separation from the single-crystal distribution were removed by the crystal cut. Therefore, the efficiency of cut based on $\Delta\chi^2$ will perform worse on real samples than the efficiency estimated in Fig. 4.15.b. The efficiency of the HP method can be estimated using the events surviving the crystal cut and is shown in Fig. 4.17.

Because we've use the CANDLES analysis tools to reconstruct event positions, we can also check which crystal the multi-crystal events are reconstructed in. The donor events are 1.8 MeV ^{88}Y in one crystal and 2.6 MeV ^{208}Tl in another crystal. Because the ^{208}Tl event has higher energy, we expect the reconstructed position of multi-crystal events to be in the same crystal as the ^{208}Tl donor's crystal. Sometimes, some events can be reconstructed in the lower energy ^{88}Y donor's crystal. 4.17.a separates the events into which of the donor event's crystal the multi-crystal event survives the crystal cut. As expected, majority of the events survive in the crystal that the ^{208}Tl donor was selected from. These events are also harder to separate from the single-crystal events. Some events do survive the crystal cut in the ^{88}Y donor's crystal. These events are relatively easy to separate from the single-crystal events. Moreover, in real data, if the multi-crystal event is reconstructed in the original disintegration crystal, not the crystal that 2.6 MeV γ deposits energy in, then these multi-crystal events can be rejected

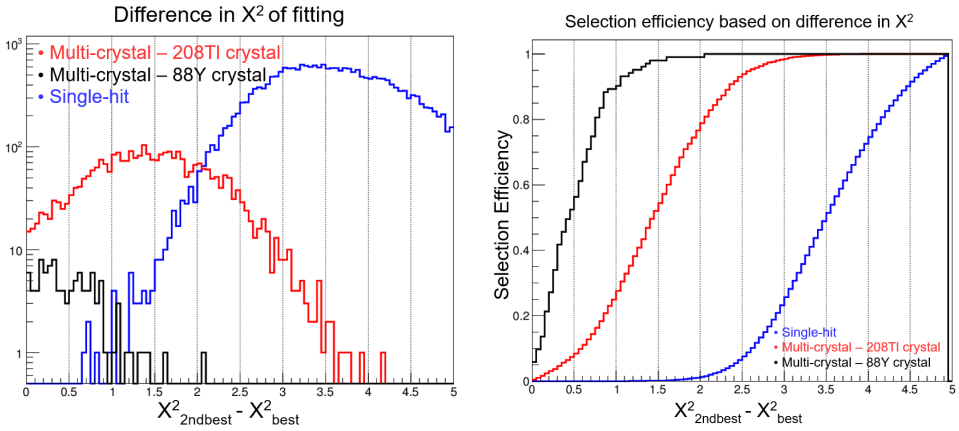


Figure 4.17: a. Difference in χ^2 of multi-crystal and single-crystal events after crystal cut. The distribution shown in red survives the crystal cut in the 208Tl donor's crystal. The distribution shown in black survive the crystal cut in the lower energy 88Y donor's crystal. b. Shows the rejection efficiency based on $\Delta\chi^2$. A good rejection efficiency is still possible with a cut based on $\Delta\chi^2$.

by the delayed coincidence method. Therefore, these events are of lesser concern.

The separation of multi-crystal events from single-crystal events are harder after crystal cut. But we see that a good efficiency can still be achieved from Fig. 4.17.b. Therefore, the HP method can be an effective tool to reduce multi-crystal events after applying crystal cut. Based on the two dimensional plot of Fig. 4.16, a more efficient cut line may be found. In practice, we used a different cut than the $\Delta\chi^2$ cut we showed in this section. The details of this cut will be presented in the CANDLES data analysis chapter.

Although original crystals of the donor events are known, this information is not used during method testing. We allow the HP method to find the best crystal of the event signal by iterating over all HPs. Since CANDLES analysis tools can determine the crystal of an event independently, we compared crystal predictions of both methods and found them to be in complete agreement. Thus HP method can be used to determine the crystal of an event.

If we compare figures 4.10, 4.11 and 4.14, we see that the mean value of the second best crystal's χ^2 for *single-crystal* events is energy dependent. When the energy is higher, $\Delta\chi^2$ is larger. This poses an issue because in CANDLES data analysis, a much larger energy region is used than the energy spectrum covered by generated multi-crystal events(Fig. 4.13). Because we do not have any good event donors outside the events used in this

section, we can not generate multi-crystal events in other energy regions. Therefore, we can not use the results of HP application on generated data directly on real CANDLES data. It is possible to modify the strength of each signal when making multi-crystal events to generate events in other energy level but this was not done because it would not simulate real data well. This is because the multi-crystal events generated here represent the special case when the 2.6 MeV γ particle deposits all energy in the neighbor crystal and no energy in the original ^{208}Tl crystal or in the liquid scintillator. In real data, this special case is not expected to always hold. Thus, the method developed here using generated multi-crystal events can only act as a general guideline and in real data application, we have to rely on a data driven approach to reject the multi-crystal events. The details of applying hit patterns to real events is presented in CANDLES data analysis chapter. But before application to data, we present the details of the convolutional neural network that was used to reduce double pulse event backgrounds in the next chapter.

Chapter 5

Reduction of Double Pulse Events with Neural Network

In this chapter, we present the details of the convolutional neural network(CNN) that was trained for detection of double pulse(DP) events in CANDLES experiment data. We begin this chapter by providing background information on DP events and previous work done to detect these events. Next, a brief overview of neural networks are provided with a more detailed introduction presented in App. A. The details of the CANDLES CNN is provided next, along with how the network was trained. Finally, some details of the various testing we performed to arrive at the final model design is presented at the end of the chapter.

5.1 Double Pulse Events

A major portion of backgrounds in CANDLES Q value region comes from the decay ^{212}Bi $\rightarrow$ ^{212}Po $\rightarrow$ ^{208}Pb. Because the half-life of the decay ^{212}Po $\rightarrow$ ^{208}Pb is 0.299μs, the event signals arrive nearly simultaneously and create DP events in CANDLES data. Examples of DP and regular single pulse(SP) events are shown in Fig. 5.1. The red bands in the figure indicate a duration of 30 channels(60 ns). The CANDLES neural network was designed to detect DP event with decay time of less than 30 channels.

After the CANDLES shielding upgrade[41], most external backgrounds are expected to be reduced to a few events per year. The remaining high energy backgrounds are from within the detector. These backgrounds are expected to be due primarily to the decay of ^{212}Bi. Therefore, a significant effort is made to reduce these backgrounds.

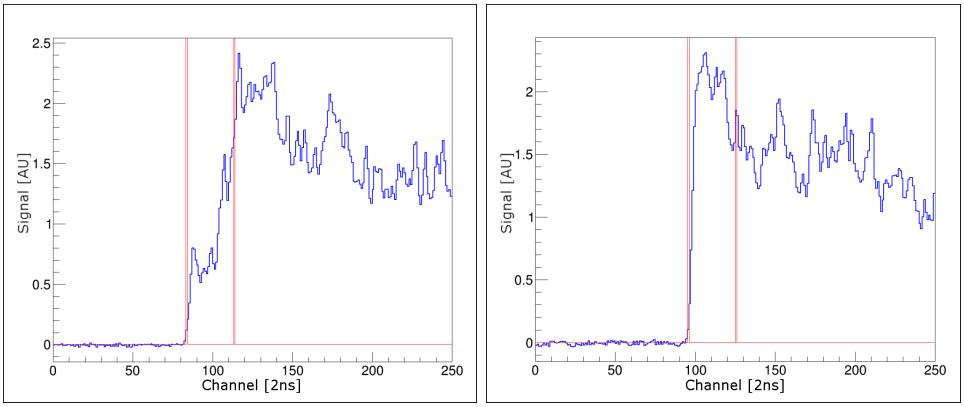


Figure 5.1: a) An example of a DP event created by the decay ^{212}Bi $\rightarrow$ ^{212}Po $\rightarrow$ ^{208}Pb. These pulses are made by combining the signals of all 62 PMTs. The red bands show the start of the event and a duration of 30 channels(60 ns). Due to photon statistics, it is often difficult to exactly determine when the second signal is received. If the time difference between the two pulses are less than 10 channels, it is very difficult to determine if the event is SP or DP. b) An example of a single pulse(SP) event.

5.2 Reference Pulse Fitting

The natural approach to detecting these DP events is to fit α and β reference pulses to the event's summed pulse as shown in Fig. 5.2. The summed pulse is obtained by combining the signal of each PMT into an overall pulse shape. The reference α and β pulses are combined and the best fitting is obtained by minimizing the χ^2. Therefore, it is a pulse shape discrimination method which we call the PSDDoubleP method.

The PSDDoubleP method has been report as having nearly 100% efficiency to detect DP events if the time delay between the two pulses is greater than 10 channels(20 ns) [42]. For DP events with less than 20 ns delay, the efficiency is reduced. These efficiencies are calculated using MC methods and also by generating false DP events.

In Fig. 5.3, the result of applying PSDDoubleP method to CANDLES data to obtain the exponential decay curve of DP events is shown. The PSDDoubleP method is applied to CANDLES data to predict the decay time of high energy events and the observed number of events in each decay time bin is shown. The half-life of ^{212}Po $\rightarrow$ ^{208}Pb decay is 299 ns. This value is used to obtain the expected exponential decay curve, shown in blue. We observe that the PSDDoubleP method predicts correct number of DP events in each bin above 30 ch. However, when the decay time of DP events is less than 30 ch, we observe discrepancy between the predicted number of DP events and the observed number of DP events. The purpose of this

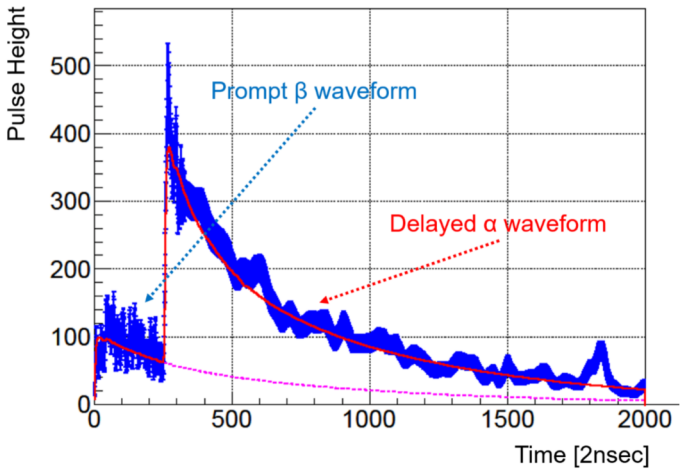


Figure 5.2: Detection of DP events is based on fitting by combining the reference α and β pulses. This method is called the PSDDoubleP method and has good efficiency to detect DP events with larger than 20 ns decay time. However, the efficiency has not been checked by an independent method.

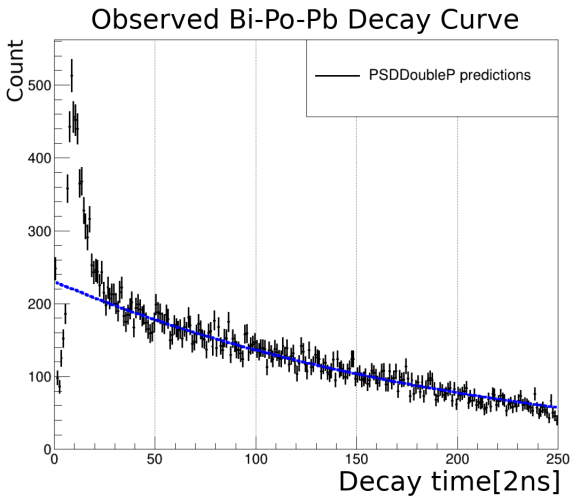


Figure 5.3: The result of applying PSDDoubleP method to CANDLES data to obtain DP events decay curve. The half-life of ^{212}Po $\rightarrow$ ^{208}Pb decay is 299ns, which is used to obtain the expected exponential decay curve shown in blue. The PSDDoubleP method obtains expected DP count in each bin above 30 ch. Below 30 ch, the observed number of DP events is inconsistent with the expected count.

study is to improve the predictions below 30 ch by using machine learning applied to the raw PMT data.

5.3 Neural Network Overview

In this section, a brief overview of neural networks is discussed. Appendix A presents an introduction to neural networks and is recommended for readers not familiar with the subject. The power of neural network is derived from its property of being able to approximate any well behaved function. We can often think of data analysis as applying a transformation function to raw data. For example, given some PMT readouts in tensor form $\mathbf{X}$, we can extract the energy of the event by applying transformation as $F(\mathbf{X}) = Energy$, where $F()$ maps the input data to energy. Generally, we can think of any output as applying some transformation function to observed data. Typically, a researcher must design every part of $F()$ to correctly map the input data to correct output. This process is time consuming and requires expert knowledge of the data.

Neural networks provide an algorithm to automatically approximate $F()$ given enough number of input and output pairs. For example, if we have enough number of event data and their corresponding energies, we would be able to approximate the transformation function from input data to energy. The application of neural network in this study can be written as:

$$F_{NN}(\mathbf{X}) = t_{decay} \tag{5.1}$$

where $\mathbf{X}$ is a combination of raw PMT data and summed pulse, t_{decay} is the decay time of a DP event and $F_{NN}()$ is a neural network that maps input to decay time. The PSDDoubleP method discussed above can be written in the same form as Eqn. 5.1, but the method only uses the summed pulse of an event as input to predict the decay time. In this study, the goal is to use neural network to approximate the transformation function that output decay time of DP events and improve accuracy over PSDDoubleP method by also inputting the raw PMT data. This discussion is expanded further in the next section.

The neural network method has some serious drawbacks. The first is that the network works as a black box and learns an implicit representation of the transformation function. This means that the uncertainty associated with the network prediction can not be determined by error propagation. This is a serious issue because often the uncertainty of the data is more important in physics than the observed value. In this study, the uncertainty of the network was approximated statistically by observing the network output. Details are presented in Chap. 6.

Another major drawback with neural networks is that it does not adapt to changing data well. For example, if the number of PMTs increased in the

input data by one, the network will not be able to handle this new input data and a new neural network must be trained. This poses an issue because during network model prototyping, various configurations and input shapes must be tried and in each case, a new model must be trained. This increases prototyping time significantly. There are other drawbacks that are discussed in App. A.

Despite its drawbacks, neural networks currently enjoy popularity among data scientists. We found that neural networks could be adapted to process CANDLES data and improve upon PSDDoubleP method for DP decay time prediction. The rest of the chapter provide details on our application.

5.4 Why Neural Network

The PSDDoubleP method works well for DP events with long decay time between the initial β and proceeding α pulses. However, we observed in Fig 5.3 that for short decay time DP events, the method performs poorly. The PSDDoubleP method attempts to fit a continuous reference pulse to the combined signals of the 62 PMTs. The event signal is not a continuous process, as it is governed by photon emission probability of the scinitllator. At roughly 1,000 photons per MeV produced by the CaF_2 scintillator, the summed pulse is not nearly smooth enough for a good fitting, as shown in Fig. 5.1. The amount of signal fluctuation per channel is often too high. This leads to poor fitting if the decay time is very short.

Moreover, for events with very short decay time between the ^{212}Po and ^{208}Pb, time of flight corrections become important. For some crystals, the time of flight of the photons can range between 3ns to 12ns to the nearest and the furthest PMTs. Currently, CANDLES data analysis does not account for the time of flight of photons. This is an issue because when a DP event occurs, each PMT will see the start of the first and the second pulses but when the signals are summed with wrong start channels, this DP timing information will be lost or reduced. As a result, we are attempting to detect DP event on information that has been diminished due to incorrect start channel synchronization.

To apply time of flight corrections to the start channels, we must create start timing correction tables for each crystal. This requires sample events with enough event statistics and energy to allow us to create timing table for each crystal. We tried to create each crystal timing table with ^{208}Tl events but found that the number of photons received at the further crystals are not enough to obtain good timing table. The CaF_2 scintillator does not produce enough photons at the start of the event to reliably determine the start channel of each PMT. We eventually dropped attempts of creating time of flight corrected timing tables.

CANDLES experiment collects and saves raw data from each of the 62

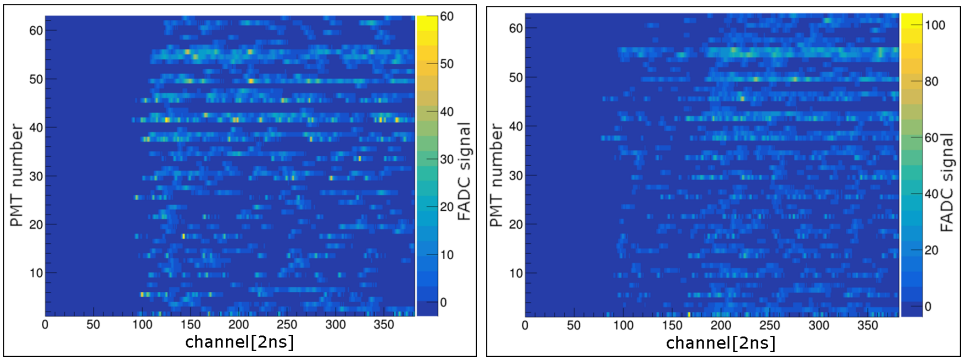


Figure 5.4: The first 384 channels of single pulse and double pulse events are shown. The y axis is the PMT number and thus each band in the figure represents the signal received at a single PMT. a.) A single pulse event is shown. b.) A double pulse event is shown. We can see visually the start of the first and the second pulses. This second pulse information contained in each PMT signal is lost or diminished if the PMT signals are combined with incorrect start channel timing.

PMTs. For DP events, the raw data potentially contains 62 pieces of information on the start time of the first and second events. The PSDDoubleP method attempts to detect the DP event after reducing the available information to a single summed pulse. Ideally, then, we need to develop a method that can classify DP event based directly on the raw data, since maximal information is available in this case. Figure 5.4 shows the signal received by each PMT in the first 384 channels for and SP and DP event. The PMT signals are stacked along the y axis; each horizontal band represents the signal received by one PMT. Figure 5.4.a shows the signal for a SP event while Figure 5.4.b shows the signal of a DP event.

There are a few points of interest here. The start channel of each PMT is not synchronized and must be done during data analysis. This can be seen in the different first signals of each PMT along the x axis. Also, both events are from the same crystal so they both show high photon statistics in the same PMTs, which are the PMTs that are closest to the event crystal. The furthest PMTs from the crystal receive fewer photons. This is why the time of flight correction is difficult, the further PMTs receive significantly less signal and also the probability of photon arriving at the PMT at the correct start channel drops. Therefore, we do not see any photons at the correct start channel in some PMTs due to photon arrival probability. But more importantly, we can visually observe that Fig. 5.4.b contains DP event. There is a clear start pulse and a second pulse recorded in each PMT signal. We need to explore the possibility of using this information to estimate the decay times of the DP events.

However, analysis of this raw data is difficult because it is a discrete data governed by photon arrival statistics. We would have difficulty "fitting" a reference pulse to each PMT signal because the shape and size of individual photons are hard to distinguish, especially in the case when multiple photons arrive at the same time. We would also need to be able to correlate the start of an event in one PMT with the start of the signal in each of the other PMTs. This brings us back to the problem of start channel synchronization, which must be done for each crystal. These are some of the difficulties with trying to apply analysis to raw data. It would be too difficult and too time consuming.

This kind of difficult to analyze data is a perfect candidate for neural network application. The network will be able to directly access the raw data and extract the most relevant features that would allow us to obtain the decay time between the initial and the second pulses. Moreover, by considering the data shown in Fig. 5.4 as an image, we can apply a convolutional neural network for this analysis. This is important because as we saw in App. A, convolutional neural networks are able to correlate spatial information. Shown in Fig. 5.4.b, DP information is highly spatial in nature. Therefore, CNNs are suitable for this type of data.

Another advantage of directly using raw data is that it mitigates uncertainties introduced during manual feature extractions. As mentioned before, creating a summed pulse loses information and also introduces uncertainty due to how the start channels were synchronized. Moreover, making the summed pulse also requires us to remove the PMT baselines. This also introduces some uncertainty due to the PMT baseline fluctuations. For detecting the time difference between the two pulses, these manual feature extraction steps may not be necessary and thus we can avoid the uncertainties introduced by each step.

Finally, neural networks are not part of the data analysis tool-set of a physicist. This is for good reason. Neural networks do not allow for a natural estimate of its uncertainties. But does this mean neural networks have no place in physics? The discussions in this section show that there are cases in which a neural network can be a good candidate for data analysis. One of the purposes of this study was to explore the viability of using neural networks for physics analysis.

5.5 The CANDLES Neural Network Model

The CANDLES neural network is a convolutional neural network with two inputs and two outputs. Figure 5.5 shows the main design of the CANDLES model. The model has two branches, one branch scanning the raw individual PMT signal, input as a 110 channel by 62 PMT matrix, as shown in the top branch of Fig. 5.5. The second branch scans the summed pulse

Branch	Input	Layer 1	Layer 2	Layer 3	Layer 4	Output
Each PMT pulse	62PMT x 110ch	Conv (15,15)x120	Conv (15,15)x62	Nodes: 100	Nodes: 50	Nodes: 31
Summed pulse	300ch	Conv (50,1)x100	Nodes: 100	Nodes: 30	-	Node: 1

Table 5.1: CANDLES MIMO model details. The network has two branches. The output of layer 3 from both branch are concatenated and input into layer 4 as one. The network has two outputs: one classifying the time delay between the initial and the proceeding pulses and the other classifying the pulse as single pulse or double pulse.

of the event, similar to the PSDDoubleP method. This is because α and β events have a slightly different pulse shapes and the network can learn to use this information for classification. We can observe a maximal difference in the pulse shape in the first 300 channels of an event(Fig. 3.16). By adding the second branch to the network, we observed nearly 10% increase in the classification accuracy. More information is provided at the end of this chapter.

The network has two outputs. The primary output has 31 nodes and predicts the decay time of DP events up to 30 channels. In other words, if the DP event has a 5 channel decay time, the network must output maximum value in node #5 and every other node should be zero. This is a classification task because we are asking the network to classify the input data into one of the 31 classes, which are the decay times between the two pulses of the DP event. The second output of the network is a single node which classify the input as DP or SP. However, this output was included only to help guide the training and was not used for analysis. Because the network has multiple inputs and multiple outputs, it was named the multi-in multi-out (MIMO) network.

Table 5.1 shows the parameters of the MIMO network. We will now look at the design of the CANDLES network in more detail. The input to the top branch in Fig. 5.5 is a matrix of received signals of each PMT. The input has shape of 62 PMT x 110 ch. We want the network to process this data and output the decay time between the initial and proceeding pulses. In the case that the event is an SP event, then it should be classified as having 0 decay time between pulses. This data can be viewed as an image where individual channel data points are viewed as image pixels and we can apply convolutional operations naturally. Therefore, the first hidden layer is a convolution using filter size of 15x15 and a total of 120 filters are applied. The 15x15 filter size was chosen after trying different filter sizes and shapes and the best performing filter size was kept. For example, a purely horizontal or purely vertical filter was used during training but we found that these perform poorly. Also, smaller and larger sized filters were tried but the smaller filters also performed poorly while there was no

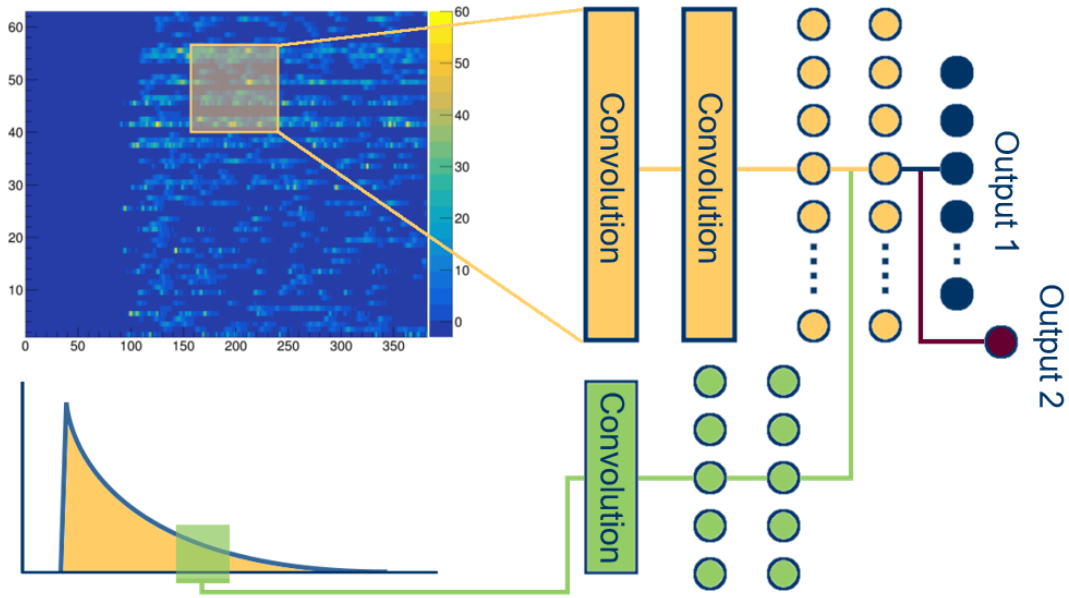


Figure 5.5: The CANDLES MIMO model. There are two inputs and two outputs. The first input sees a window of size 110 ch from 70 ch to 180 ch and signal from each individual PMT. This data can be treated like and image and we can apply a convolutional neural network. The second input is the summed pulse of the event. Because α and β events have different pulse shapes, it is an additional information that can guide the network for better classification accuracy. The type of hidden layers used are shown in the figure and the parameters of each layer is given in Table 5.1. Output 1 of the network classifies the event's decay time between the first and the second pulses. It has 31 nodes and can classify decay time from 0 channel to 30 channels(60 ns). The second output classify whether the input data is SP or DP event. However, this output was only added to help train the network. Without this output, we did not see an improvement in accuracy when we added the second input branch.

significant improvement with larger filters. Using larger filters increase the number of trainable parameters and time to train the network. After these considerations, the 15x15 filter size was chosen. For details, see Sec. 5.9.2.

Conceptually, the filter size makes sense because if the filter detects the start of an event at some pixel, it should also be able to anticipate event start signal in the nearby PMTs at around the same pixel location. This anticipation of signal in the neighboring PMTs is important to distinguish if signal is noise or an actual start of pulse. If multiple PMTs see an increase in signal at around the same pixel regions, then the filter is able to distinguish this as start of an event instead of just a single photon arriving at one PMT. In this way, the filter is able to correlate information about pulse start in the surrounding PMTs as well as learning a pseudo start channel anticipation. If the filter size is too small, then the filter's correlation power between PMTs and channels will be diminished and if we use purely horizontal or purely vertical filters, then it will not be able to correlate between PMTs or unable to correlate in the channels. On the other hand, if the filter size is too large, too many pixels of the filter are left unused. This is because when an event starts, the start channels of each PMT should fall within a few channels of each other and since this start channel is the relevant information, the filter will only need few pixel width to capture this information. Therefore, the larger sized filters do not extract significantly more information.

The input to the first convolutional layer is (62 x 110) and we used 120 filters. Each of the filters produce an activation map of size (62 x 110), therefore the output of the first layer is 62x110x120. This is already a very large data and most of the data is not very useful. That is, the relevant information is sparse in this large tensor. We should trim the data into smaller size. This is done by applying a second convolutional layer with smaller number of filters. The second convolutional layer has 62 filter of size 15x15x120. The depth of 120 in the filter size comes from having 120 activation maps from the first layer. These filters are able to see all of the output activation maps of the first layer at the same time and correlate the activation maps. This is an important step because in the first hidden layer, some filters will have good activation in some parts of the input data while other filters have activation in other parts. Therefore, only after correlation between the filters of the first hidden layer, we get a good sense of the start channels of the two pulses of the DP event. In some sense, we can think of the network as building confidence in its classification by comparing with multiple filters in the first layer.

The output of the second convolutional layer is input into 100 nodes in the next layer. Each element of the activation maps of the second layer are connected to each node in the 3rd layer. This requires over 17 million trainable parameters and the bulk of the parameters of the model is in this connection. The purpose of this layer is to compress the relevant information coming from the convolutional layers to 100 nodes, which is a single vector

of size 100.

The second branch of the network sees summed pulse of 300 channels. It was observed empirically that adding this branch as well as the second output increased the accuracy of the network by almost 10%. This is because α and β events have different pulse shapes and the network can make use of this information. The first hidden layer of the pulse branch is a 1D filter of size (50,1) and 100 filters were used. The filter size is large because it should be able to see the overall pulse shape. The output of the convolutional layer passes through two layers of nodes with 100 and 30 nodes respectively. The final 30 nodes contain all the information from the summed pulse relevant for classifying the event.

At layer four, the outputs of the two branches are combined into a single long vector before being input into 50 nodes. The two outputs of the network both have access to information coming from both input branches. The two branches have outputs of 100 and 30 nodes each at layer three. The information is combined simply by appending one vector after the other into a vector of size 130. The 4th layer extracts the most relevant information into 50 nodes and passes them into the output layers. We can think of this final 50 node layer as having compressed the (62 x 110) raw PMT data and the 300 channel pulse shape into the most relevant information to classify its decay time.

The network has two outputs. One predicts the decay time between the initial and secondary pulses. This output has 31 nodes and the node with the largest value determines start of the second second pulse. For example, if the second pulse starts 5 channels after the initial pulse, then the network must output largest value in node #5. If the event is an SP event, then node #0 must have the largest value. The second output of the network classifies the input event as SP or DP event. This output was added because we observed that adding the second input branch to the network did not improve the network accuracy but we saw an increase in accuracy when both the second branch and the second output were added. The intuition for this is that the network can not reliably use the summed pulse information to detect the start of the second pulse and so this branch is not useful without additional guidance. When the second output is added and we force the network to also predict into SP or DP classes, the filters of the second branch are trained to look for pulse shape differences and this additional information is usable to predict the start of the second pulse.

Output 1 has a multi-class(categorical) cross-entropy loss function attached(see Sec. A.2.1). Output 2 has a single-class cross-entropy loss function attached. Apart from the output nodes, all other nodes used the ReLU activation function. An overall loss function is defined by simply adding the loss of the first and second outputs. In principle, the loss functions of each output can be added with some weights but after trying several values, we found that equal weights to both output loss functions provide good network

accuracy. Therefore, the loss functions are added with equal weights.

During test phase of the neural network, we tried many different model sizes, designs and filter choices. The final design of the network is based on empirical observations and also guided by intuition. Some variations are shown at the end of this chapter. There may be a better design but searching for different model designs is time consuming due to the 2 to 3 hours required to train each model. We also do not expect to see major improvement in model accuracy because the MIMO model performs well given the uncertainty of the training data. However, with more modern network designs, the size of the model may be reduced effectively. The current CANDLES MIMO model have roughly 21.5 million trainable parameters.

5.6 Making False Double Pulse Events

Training a neural network requires a very large set of pre-classified(or labeled) data. This is not available from CANDLES data. The PSDDoubleP method can provide some DP events as training data but not enough to train the large MIMO model. This model requires tens of thousands of training data to avoid over fitting by the model. To provide the network with training data, false DP events were generated from donor events.

To generate false DP events, we combine the signals from two known donor pulses as shown in Fig. 5.6. Since we can choose the decay time of the false event, we are able to generate as many labeled sampled as necessary. Figure 5.7 shows an example of DP event generated with this method with a large decay time. Although the figures show the summed pulse, the DP events are generated by also adding the pulses per PMT.

SP events were also generated to train the network. These are generated by the same method but with 0 decay time between the initial and the secondary pulses. The donor events selection and method of DP and SP event generation are as follows. To generate SP events, 1.8 MeV γ donor from ^{88}Y calibration source was added to 2.6 MeV ^{208}Tl γ donor. Because the other major source of background is the decay of ^{208}Tl with several γ emissions, this combination of donors simulate an SP ^{208}Tl decay. The summed pulse has a total energy of 4.4 MeV, which is in the region of interest of CANDLES detector. To generate the DP events, 1.8 MeV γ from ^{88}Y source was added to the naturally occurring ^{215}Po or ^{214}Po α donor. This emulates the β then α emission of $^{212}\text{Bi} \rightarrow {}^{212}\text{Po} \rightarrow {}^{208}\text{Pb}$ decay, which is the source of the DP events in CANDLES. Due to the quenching factor of α events, the ^{215}Po and ^{214}Po α events have an energy of 2.2 MeV. This combination generates DP events with energy distribution centered around 4.0 MeV to 4.2 MeV.

The donor event selection criteria is shown in Table 5.2. The donor events are selected to be within 1.5σ of a crystal's center. Because ^{88}Y

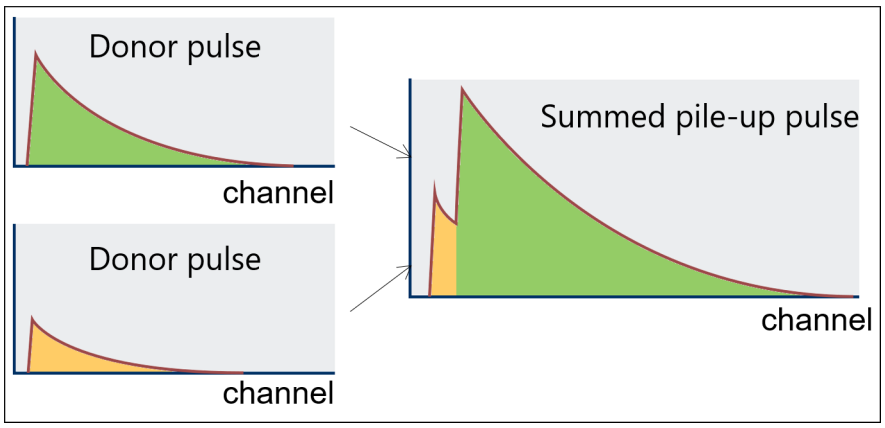


Figure 5.6: Generating false double pulse events. We can take two known donor events and combine their signals to generate false DP events. The signal is summed at each PMT level and also a summed pulse is generated, as shown in this figure. We can also control the decay time of the DP event, and thus we can generate many labeled training data.

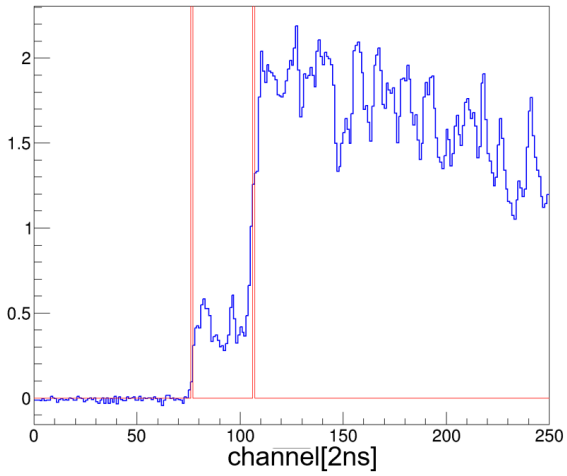


Figure 5.7: An example of generated double pulse event.

Donor	Crystal	PSD	Energy
^{88}Y γ particle	$< 1.5\sigma$	$< 1\sigma$	1.8 MeV + 75 keV
^{208}Tl $\gamma(\beta)$ particle	$< 1.5\sigma$	$< 1.5\sigma$	2.6 MeV + 75 keV

Table 5.2: Donor events selection criteria for generating double pulse and single pulse events. Strict cuts are used to reduce event selection with LS signal as much as possible.

source is external to the crystal, the emitted 1.8 MeV γ has high probability of depositing energy in the liquid scintillator. To avoid liquid scintillator signal in the donor event, a very strict PSD cut is used, at $< 1\sigma$. Also, to avoid energy deposit in the liquid scintillator, only events with reconstructed energy in 1.8 MeV to 1.8 MeV + 75 keV range are selected. Because the calibration ^{88}Y source has high intensity, we can use these strict cuts and still obtain a large number of donor events per crystal. In the case of the ^{208}Tl, the number of available events is not large enough to allow us to use a strict cut. Thus, PSD cut was loosened. However, the energy cut still only selected events above the mean value of the energy distribution. The mean value of the ^{208}Tl energy distribution is fitted for each crystal and varies per crystal. The cut selects events within 75 keV above the fitted mean of each crystal. The α events are selected based on event timing and results in a very reliable selection of α events(Sec. 3.6).

There are two main considerations when adding the signals of the donor events. The raw individual PMT data and the summed pulse has a PMT baseline that needs to be removed. For the individual PMT signal case, a constant value of 245 is subtracted, which is a typical baseline level of PMTs. Some PMTs have different baselines and some baselines fluctuate. We do not apply any special baseline subtractions for the individual PMTs signal except the constant value subtraction. This will allow the network to compensate for varying baselines and help to generalize to unseen data when real CANDLES data is input to the network. In the case of the summed pulse, the CANDLES analysis tools determined baseline is used. This baseline collects the early channels of an event and subtract from the collected signal. The DP events are generated after both of the donor event's baselines are removed.

The second consideration is to match the event start channels of both of the donor events. The start channels of events are not fixed to any certain channel and it is determine during analysis. The start channels of events can vary widely from one event to another. Figure 5.8 shows the start channels of the selected donor events used to create SP events in crystal #1. The ^{88}Y source is used during CANDLES calibration and different set of DAQ settings are used during calibration runs. As a results, the start channel of the ^{88}Y events are almost 10 channels earlier than the ^{208}Tl events. We want the MIMO network to work on events with varying start channels. Therefore, the start channel of the base donor event is not fixed to any channel. In both DP and SP cases, the ^{88}Y events act as the base donor and the generated false events keep this start channel distribution of ^{88}Y. However, for a given ^{88}Y base donor, the second donor event's start channel is shifted to match the base event's start channel. In this study, we used the start channel of each event as determined by CANDLES analysis tools. Since the start channel determination is not perfect, especially since time of flight of photons are not corrected for, a one or two channel error is

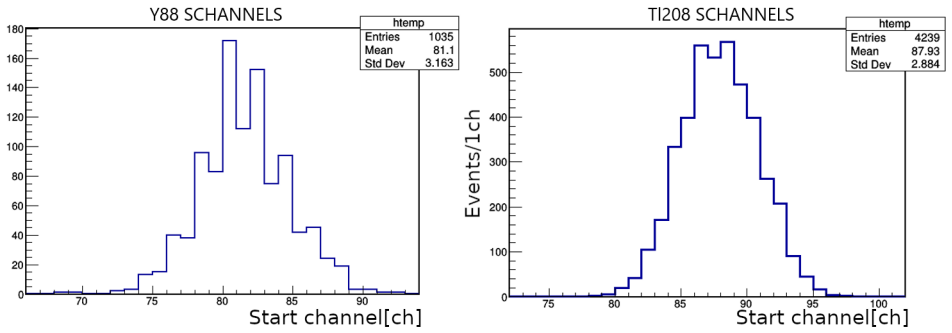


Figure 5.8: Donor events start channels for generating double and single pulse events. The start channel distribution is large for the donor events. The start channels of the ^{88}Y donors are not fixed. Thus the network trains on this start channel distribution. However, the second donor's start channel is shifted to match that of the ^{88}Y donor. When generating DP event, artificial decay time is added after both donor's start channels are overlapped.

expected in the start channel of an event. We will see later that this is the main source of error for the network. In DP generation case, the time difference between the initial pulse and the second pulse is added after start channels are overlapped.

To reduce the amount of unnecessary information being input into the network, the first 70 channel of an event is discarded. As shown in Fig. 5.8.a, most ^{88}Y event have start channel above 70 ch. Events with start channel below 70ch are automatically removed from training data. Also, the network only needs to make DP predictions in the first 30 channels from the start of the initial signal. Therefore, a large window is not necessary when we input the individual PMT signals to the network. The upper channel cutoff is placed at 180 channel. This gives us an input size of (62 PMT x 110 ch). The selection window for the MIMO network upper branch is shown in Fig. 5.9. Under the same reasoning, we also trim the summed pulse branch information. The pulse shape difference between α and β events are most visible in the first 300 channels from the start of an event. This is shown in Fig. 5.10. Therefore, the input is trimmed to be between 70 ch and 370 ch.

With this method, 120,000 training samples are generated per crystal for network training. The data set is split into 3 groups with each group containing 40,000 events. The differences between the groups are in the ratios between SP and DP events. In the first group, the ratio of SP to DP is 1:2. For the other two groups, the ratio is 1:6 and 1:30. The reasoning for this training data split is because we need the network to focus on correctly classifying SP events, since these are the main data of CANDLES. For this reason, the 1:2 split data set was created. However, we would like the network to see enough events in each classification node to accurately make

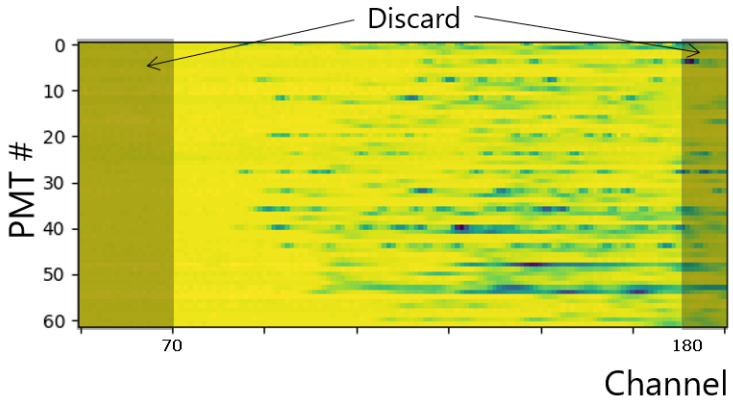


Figure 5.9: The first 70 channels of an event is mostly unused. Therefore, we discard these channels. Also, the network does not need to see very many channels to make a maximum of 30 channel decay time prediction. Thus, the upper cutoff is set at 180 channel.

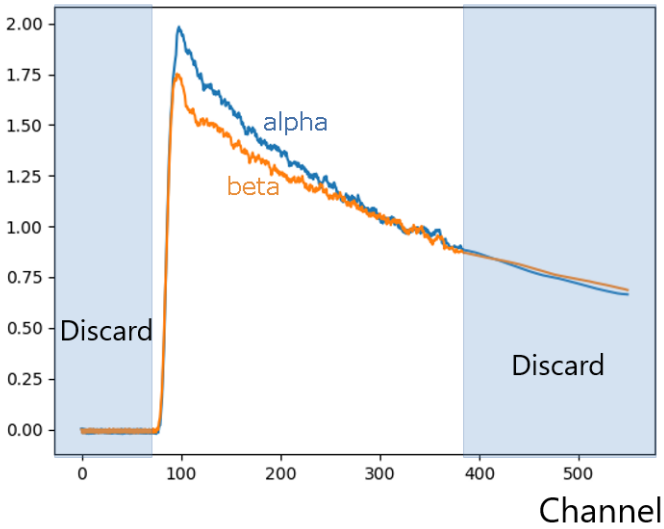


Figure 5.10: The difference between α and β pulse shapes are mostly visible in the first 300 channels from the start of an event. Thus we trim the event pulse shape from 70 ch to 370 ch.

DP event decay time predictions. The natural split between SP and DP is 1:30, which simply creates equal number of training samples per output node. However, this split does not place enough emphasis on SP events. As a result, a 1:6 ratio data set is generated to balance focus on SP events but also have enough DP events per decay time bin. During data generation, these ratios are not strictly reinforced but the probability of SP or DP generation uses these ratios. Therefore, in data, the real ratio would be slightly different.

In principle, more than 120,000 training sample can be generated per crystal but this total already presents a sizable data set. We completely filled a 500GB hard drive with only generated training data for 30 crystals. Therefore, used training data had to be deleted to make space in order to train MIMO networks for other crystals. Moreover, the CANDLES data is stored on the Research Center for Nuclear Physics(RCNP) servers, which does not have the hardware or software needed to train the MIMO model. We generated training samples on RCNP servers and transferred the data to local machine to train the network. This required quite a bit of bandwidth and disk space. As a results of these considerations, we were only able to generate 120,000 events per crystal for training.

Finally, due to an oversight, DP events with 0 decay time between the initial and secondary pulse were not created in the training data. Because the start channels of each event are not perfectly determined, we include 0 channel decay time DP in the training data by accident. But these will be labeled as being 1 channel decay time or very rarely as 2 channel decay time in the training data. Therefore, the network is trained on 0ch decay time DP events but will be told to classify these as 1ch or 2ch decay time events by accident. The effect of this is studied in App. C. The conclusion drawn is that most of these 0ch decay time DP events will be predicted as having 1ch decay time. However, about 15% is expected to be predicted as 0ch decay time events, although this result is not very reliable because the data set used to test is not a perfect representation of these events in real data.

5.7 Training Neural Network for Double Pulse Detection

In this study, we trained a MIMO network for each CaF_2 crystal. We attempted to create one model that would work on all crystals but the accuracy of the network was not as good as networks trained for each crystal. The largest network that we trained to work on all crystals was 4 times larger(81 million trainable parameters) than the single crystal models but the accuracy was about 15% lower. Several different sizes and designs were tried but none performed better than single crystal models. Therefore, we

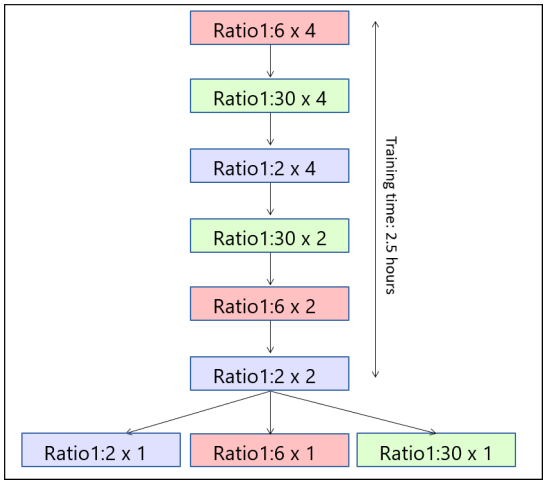


Figure 5.11: MIMO network training regime. Each data set is used 6 times for training and we split the training schedule to 6 steps. Once a single model is trained on this regime, we split the model into three model, each specializing in different ratio data set. The total training time is about 3 hours per crystal.

chose to train a MIMO network for each crystal. More details on the various models tested are presented in Sec. 5.9.

The networks are trained on the 120,000 generated DP and SP events. But these generated samples are grouped into 3 parts with each group containing 40,000 training samples. The grouping was made to emphasize more SP events in the training. It is not clear which group is the ideal data set to train the network with. Therefore, we trained 3 models per crystal and combined the outputs of the 3 models to obtain a final classification. The initial network parameters are generated randomly from a gaussian distribution centered at zero and with standard deviation of 1. All of the activation functions used in this study are ReLU (Sec. A.2.2) except for the output nodes, which uses the logistic function or softmax(Sec. A.2.3) depending on the size of the output. The subset size(batch size) of the stochastic gradient descent(Eqn. A.11) is 128.

The network training regime is shown in Fig. 5.11. Remember that training a network means minimizing the loss function(Sec. A.2.1) by applying the gradient descent algorithm iteratively. Initially, only a single network is trained, starting with the 1:6 SP to DP ratio data set. The network is trained on this data set for 4 times, meaning each sample in the 40,000 training samples is used 4 times by the network. After the training is complete, the next data set is loaded and the network continues training. The network training follows the regime shown in Fig. 5.11 and it is split into 3 models at the final stage. The 3 models complete training regime

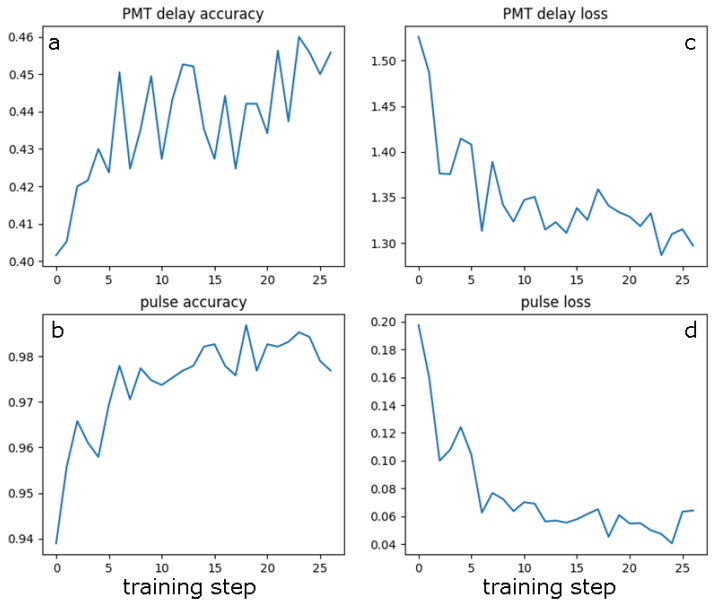


Figure 5.12: a. Increase in output 1 prediction accuracy after each training step. We see that the accuracy is plateauing, indicating that network is approaching maximal accuracy. b. Accuracy of output 2 also increases and plateaus at the final stage of training. c. Minimization of the loss function is shown for output 1. The loss function is starting to plateau, indicating the minimum of the loss function. This indicates that the network has reached the possible minimum of the loss function. d. Minimization of the loss function of output 2.

by training once on each of the data set group. In a sense, each model is specializing in data with the grouping ratios. However, this specializations are not used and we simple take the average of the 3 models as output when making predictions on new data. The training time for this regime is about 3 hours total.

Figure 5.12 shows the training curves of the final stage of network training(last stage in Fig. 5.11) for 1:30 ratio trained network. At the beginning of the training stage, 2,000 events are removed from the training data to use for testing network accuracy. This data set is not used to train the network and thus act as an independent test for network accuracy during training. The curves shown are for this testing data set. During network training, only 2,000 training events are loaded each time due to computer memory considerations. Therefore, each training step represents network training on 2,000 events. Figure 5.12.a represents the accuracy of the network to predict the decay time between first and second pulses on the testing data set. We see that at the start of the training, the network accuracy is about 40%.

After 25 training steps, this accuracy stabilizes around 45%. More importantly, this stabilization of the accuracy is the main indicator to stop the training. Because the accuracy curve is for the testing data set, any further training will over fit the training data and the accuracy of the testing set would decrease. Therefore, the network training is stopped at this stage. Figure 5.12.c shows the loss function of output #1 at each step of training. We also see here that the loss function stabilizes, indicating a minimum for the loss function.

Figure 5.12.b and 5.12.d shows the accuracy of output 2 and its loss function. The network has 98% accuracy to predict if the input event is SP or DP. But as mentioned before, this output was included only to guide the training and the output is not used for data prediction.

For each crystal, 3 models are trained. The training time for 1 crystal is about 3 hours. This does not include time required to generate the training data and time required to copy the generated data to local machine. Every step could not be automated due to disk size considerations. As a result, the total time required to train all models for all crystals was about 4 weeks. Since this was our first attempt in using neural networks, the training time is longer than what can be achieved by an experienced programmer or computer scientist. However, we believe the accuracy of the network can not be improved significantly further on this training data set. This is because the training data set has inherent errors due to incorrect start channel determine by analysis tools and the network is unable to make correct classifications on this data set. However, we will see in the next section that the inherent data set error is not a serious issue in network classification accuracy.

5.8 Results on False Double Pulse Events

Here we present the results of the trained network on the generated false DP and SP events. Application to real CANDLES data is shown in Chap. 6. At this stage, it was not known if the method would work on real data, especially given that the generated DP and SP data have fixed energy distributions centered at 4.0 MeV and 4.4 MeV. This was one of the main questions that this study explored: whether we can use false generated DP and SP events to develop a method that would work on real data. More importantly, since the training data are not real data, we can not use these data to estimate model uncertainties. For this reason, a detailed study of the model error is not carried out on the generated data. Estimating model error based on real data will be a topic of Chap. 6.

An example of network prediction on a single input event is shown in Fig. 5.13. The top figure is a DP event pulse created by combining two donor events. In this case, it is possible to determine the start channel of the second pulse visually. The bottom figure is the network prediction of the

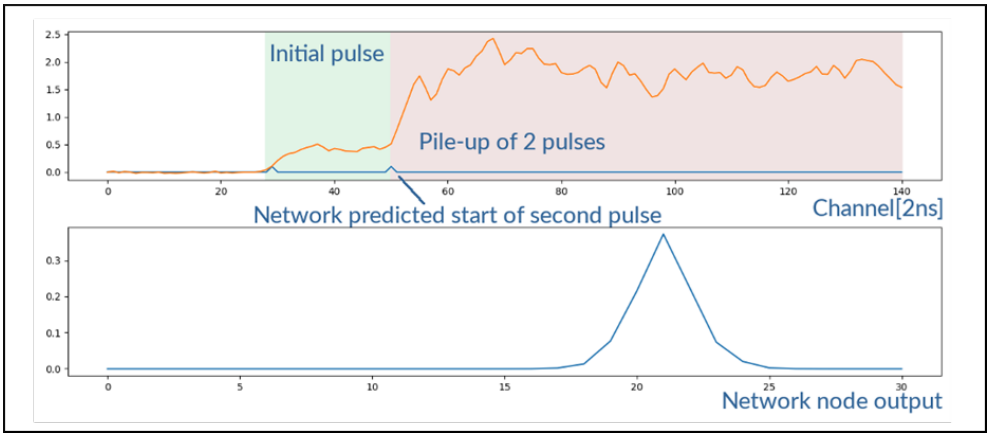


Figure 5.13: Shows a DP event generated from donor events. We can visually distinguish the start of the second pulse. The network output is shown in the bottom figure. Node #22 of the network has maximal output value, indicating a decay time of 22 channels. This is shown in the top figure and we can confirm visually that it is correct.

decay time of the event, given by the output values of the 31 nodes in the output layer. We can see that the network outputs maximum value at node #22, predicting a 22 channel delay for the second pulse. This 22 channel delay is drawn on the top figure. We also see that the network outputs non-zero values for the neighboring nodes. This is an important sanity check for the network because it does not predict wildly varying decay time for the second pulse. It keeps the prediction within the region of the correct decay time.

We might ask why the network accuracy is not better. The main reason is that when we generate DP or SP events, we first match the start channels of the two donor pulses. But the start channel of each event is not completely accurately determined. This is due to photon arrival probability and also due to not correcting for time of flight. As a result, when the false events are generated, they inherently have an error in start channel of 1 or 2 channels. This is an issue for network training because we can tell the network that the decay time for the second pulse is some value during training but in reality, the generated pulse has an error of 1 channel. Trained on this data, the network learns to output a Gaussian like prediction, which closely resembles input data with some error in the training label. In this respect, the network is smarter than what we trained it to do; it has learned to output the only sensible predictions without any supervision, which is a Gaussian type prediction output. There is another source of error in the training data which also exist in real data. We can imagine an event occurring in a crystal but due to photon flight time or due to equipment timing, one PMT sees a

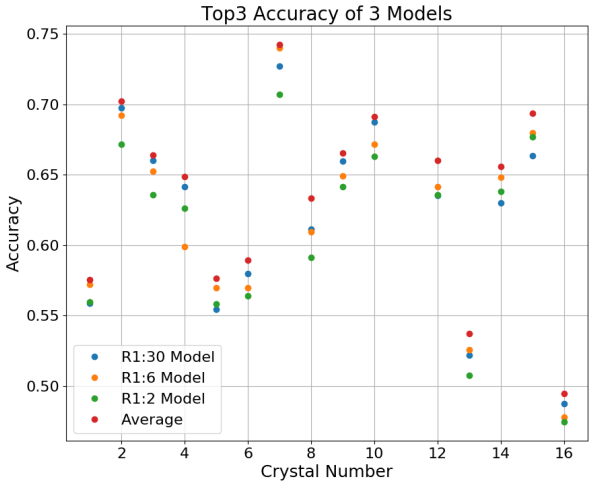


Figure 5.14: Top1 accuracy for first layer crystals. The accuracy shown here are obtained from additional 6,000 events generated with ratio 1:30. The accuracy of the three models of each crystal are consistent with each other. We also observe that the average of the three models usually perform slightly better than the individual model predictions.

start channel of some value but due to rounding up or down, another PMT places the start channel of the event into the next channel. This can occur naturally and both PMTs have the correct start channel, given the width of the channel. In this case, deciding on the correct start channel is difficult and we can expect the network to also have difficulty placing the correct decay time into the prediction nodes. As a result, the network predictions widen into a Gaussian shape.

Figure 5.14 shows the accuracy of the MIMO networks applied to the first layer of CaF_2 crystals. To calculate these accuracies, 6,000 additional data containing SP and DP events were generated for each crystal. The accuracy is calculated by taking the largest value node of output #1 as the prediction and comparing to the decay time applied during data generation. This is the top1 accuracy, meaning we only consider the largest value node as the prediction. For each crystal, 3 networks were trained, as detailed in the last section. In Fig. 5.14, the accuracy of each of these models are shown along with the average of the three predictions. We can see that in general, the networks have over 50% accuracy for most crystals in predicting the correct decay time of the second pulse. Moreover, the average of the 3 models have a slightly better accuracy than individual model predictions. Therefore, for real data predictions, we always take the average of the three models as final prediction.

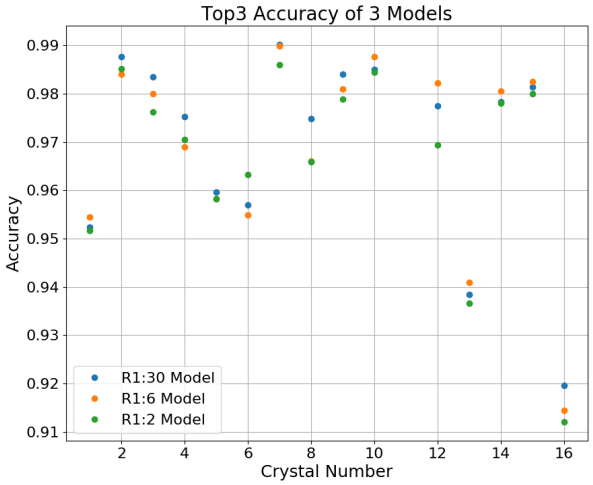


Figure 5.15: If we allow ±1ch error for the network prediction, then the network accuracy is more than 95%. This indicates that the network has reached maximal accuracy possible and the ±1ch error is likely due to error inherent in the training data.

Figure 5.15 shows the top3 accuracy of the same models. The top3 accuracy is defined as the highest value output node and its two neighboring nodes, one on each side. For example, if the network predicts the decay time of the second pulse as 5 channels, but the real value is 4 channels or 6 channels, we consider this as a correct classification. In this case, we can see that the MIMO network accuracy is over 95% for most crystals. As mentioned before, the true start channel and predicted start channel of an event can have 1 channel error and if we allow the network this error, the accuracy of the network is very high. However, this accuracy is calculated on fabricated DP and SP events and we have no guarantee of it working on real data. Therefore, we can not use this accuracy as the network accuracy on real data. The accuracy on real data has to be estimated using CANDLES data.

Figures 5.16 and 5.17 show the top1 and top3 accuracies for all crystals. As before, 6,000 events were generated for each crystal to obtain these accuracies. We can observe that network accuracies increase for higher numbered crystals. The reason for this is currently unknown. The network training follows the same training regimen for all crystals and the same model architecture is used for all models. Therefore, the reason for this trend must be contained in the data. Detector-wise, the top layer of the side wall PMTs are of different size than the rest of the PMTs. This may be the cause for this trend.

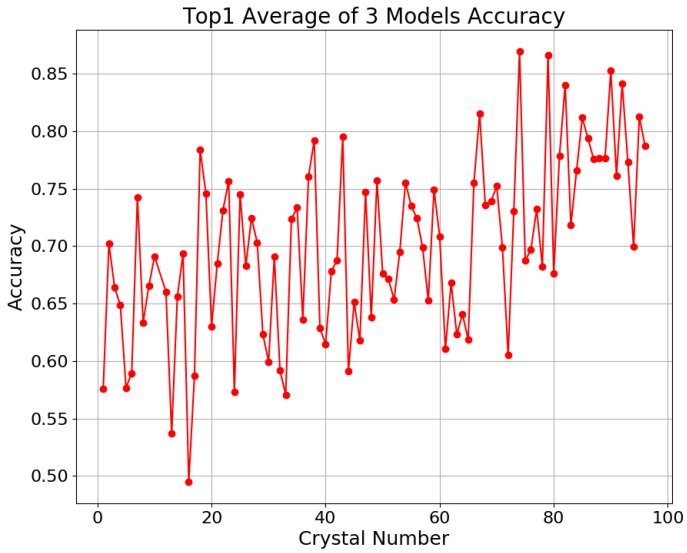


Figure 5.16: The top1 accuracy of the networks improve for lower layer crystals. The reason is not currently understood but is likely due to layer 1 crystals being close to the 10 inch PMTs.

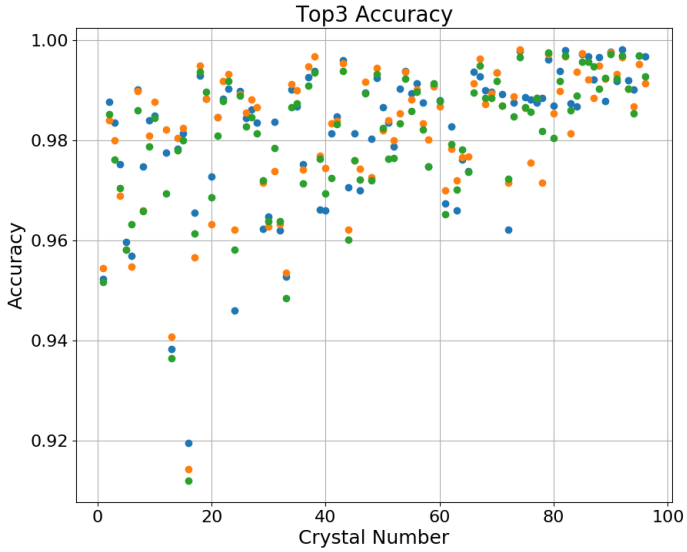


Figure 5.17: Top3 accuracy for all crystal networks. We see that the networks achieve a very high top3 accuracy. However, this accuracy is obtained for generated double pulse and single pulse events.

Finally, we can see that the top3 accuracy for all crystals are consistently very high in Fig. 5.17. With considerations to the error of the training data, the network performance is as good as we might expect. The top1 inaccuracy is likely due to noise in the training data. Considering the difficulty of the task, high top3 accuracy is good enough for our application. Different model sizes and designs are not expected to improve the model accuracy significantly. We may try to reduce the model size in the future to reduce compute time but as our first attempt in using neural networks, these models are performing above our expectations and likely very close to maximum accuracy, given the training data.

5.9 Model Designs and Parameters Search

In this section, details on how we arrived at the MIMO network design are presented. Since this was our first attempt at integrating a new technique for physics analysis, we used a trial and error approach until we came to a design that performed to our satisfaction. Thus, we present here what we learned along the way and what mistakes were made. This section can be skipped without any loss of study continuity.

5.9.1 Pulse Only Model

In our first application of neural networks, we tested if it could be used to detect DP events in general. The first network was a simple node only network(usually called the fully connected network) with 20x20x10 hidden nodes and a single output as shown in Fig. 5.18. The input to the network was the first 382 channels of the summed pulse of an event. The network had binary output and only predicted if the event was DP or SP. At this stage, we did not attempt to classify the event by decay time as in MIMO network. Because this was a fully connected network, the 382 channel input is connected to every node in the first hidden layer. Thus, the input to the first hidden layer has 382x20 weights plus 20 bias parameters. This is not a very large network and only served as a testing model.

The training samples were generated in the same way as before; we combined 1.8 MeV event from ^{88}Y with 2.2 MeV event from ^{215}Po with varying delay between initial and secondary pulses. Unlike the MIMO training data, only the summed pulse shape is generated and also at larger decay times, going up to 600 ns between the two pulses. For this application, SP events were not generated and we simply scaled known SP events to the same normalization as that of the generated DP events. For the MIMO case, SP events were generated by combining two known SP events with zero decay time.

The network was trained on 40,000 generated samples. We then applied the trained network to CANDLES experiment data. Some of its predictions

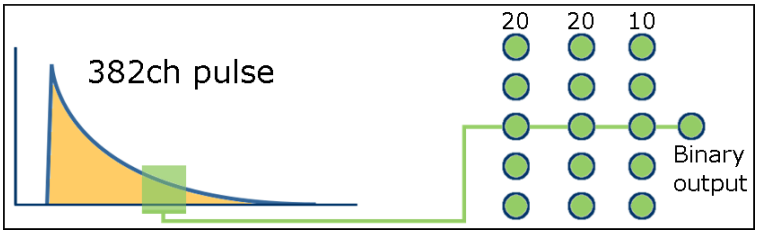


Figure 5.18: The design of the first neural network model applied for CANDLES data is shown. The input to the network was the first 382 channels of an event. The input was connected to hidden layers with 20x20x10 nodes. The output of the network is binary, which was used to predict if the input was a double pulse event or single pulse event.

on pulse shapes are shown in Fig. 5.19, where DP predictions are colored in red and SP predictions are colored in blue. The channels up to the start channel of the event are removed before prediction. Thus, the pulse shapes end and have value of 0 before 382 channel. We can observe visually that the network makes mostly correct predictions. There are no obvious DP events predicted as a SP event. In general, the SP rise time is faster than the DP predicted events, as expected.

The results of this network showed that neural networks could be adapted to make predictions on physics data. More importantly, this test showed that a network trained on generated DP events can make correct predictions on real data. Since training a network requires a large number of training data, we could satisfy this requirement by generating our own training data.

At this stage, we did not separately train a network for each crystal but the results looked promising. However, we knew that the PSDDoubleP method could achieve good results down to 20 ns decay time. In order to improve the accuracy below 20 ns, we knew that TOF corrections must be made for each crystal. However, due to the low number of photons received at some PMTs, poor fittings were observed that led to poor TOF corrections and ultimately we stopped our attempt for TOF correction.

5.9.2 PMT Data Only Model

We attempted to get around the TOF problem by applying a neural network directly on the signal received at each PMT. As discussed in Sec. 5.4, working directly on raw PMT data has the advantage of not losing any information when making summed pulse and also avoids introducing systemic errors due to incorrect start channel corrections. With these considerations, a CNN model was designed as shown in Fig. 5.20, which can be recognized as the PMT branch of the MIMO network in Fig. 5.5.

The input to this network is 140 channels by 62PMTs, representing the

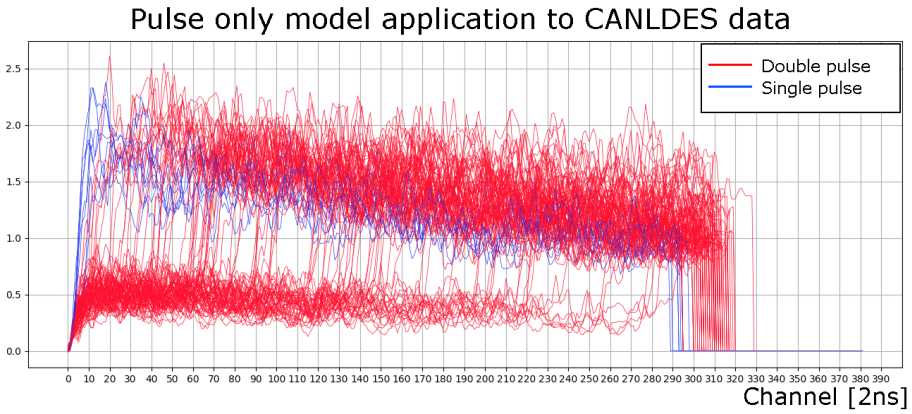


Figure 5.19: Pulse only model predictions of CANDLES data is shown. The input is the first 382 channels of an event. However, channels before the start of an event are removed and the remaining pulse is shifted. Thus, the pulse shapes end before 382 channel. Double pulse predicted events are shown in red and single pulse predicted events are shown in blue. We can observe that in general, the network makes correct predictions.

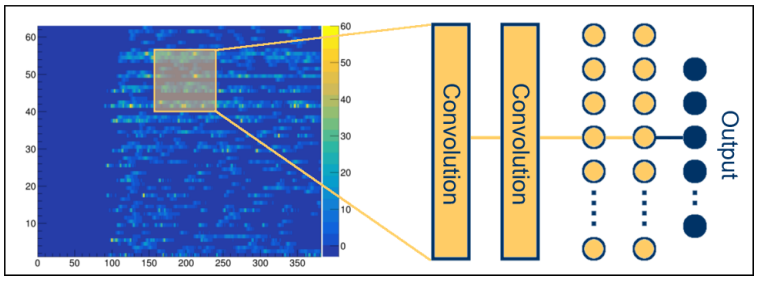


Figure 5.20: The first PMT data only model design is shown. The purpose of this model was to check if the network could learn to predict DP events from raw PMT data only. The results encouraged us to try various new models to improve its accuracy.

raw signal received at each PMT between channels 50 and 190. The channel size was later reduced for MIMO network to a duration of 110 channels from 70 ch to 180 ch with no significant loss in accuracy. The network has 31 nodes as output, with each node predicting the decay time of the DP event up to 30 channels. As before, the training data for this network was generated by adding ^{88}Y, ^{208}Tl and $^{214/215}$Po events with various time differences between the initial and secondary pulses. The training data set was generated in three parts, with SP to DP ratios of 1:2, 1:6 and 1:30, as described in Sec. 5.6.

The main feature of this network was the two convolutional layers. The intuition for this was that the first convolutional layer would scan the input data with filters designed to detect the start of the first and second pulses and the second convolutional layer would correlate the output of the first layer filters. The second convolutional layer has a second function, which is to reduce the number of parameters. Because the first convolutional layer has 120 filters, if we connect this layer to the third hidden layer, which has 100 nodes, then the number of parameters would be 120 filters x 62 PMTs x 140 channels x 100 nodes, which gives us over 100 million trainable parameters. With the addition of the second convolutional layer with 62 filters, the number goes down to 57 million trainable parameters. Thus, a significant reduction of trainable parameters is achieved as well as reduction in training time.

The design of a network requires setting numbers such as number of filters, filter sizes, number of nodes, number of layers etc. There are no general guidelines on how to set these and thus these must be set on a trail and error basis. To select the filter size of the first convolutional layer, several different filter sizes were tried, as shown in Fig. 5.21. This figure is obtained by varying the filter size and training the network on the same data set and same training schedule. In this case, the network was trained for 30 steps, where each step is a training on 2,000x3 training samples, taken equally from each of the three different SP to DP ratio data sets. However, because the initial parameters of the network are set randomly, the network can find different local minima of the loss function and thus the overall accuracy is fluctuating at different filter sizes.

From Fig. 5.21, we can observe that in general, the bigger the filter size, the more accurate the network becomes. But the accuracy does not scale well with the total number of parameters of the filter. We can observe that 15x15 or 20x20 filter size is a good compromise between the number of parameters and the overall accuracy of the network. In a similar manner, appropriate numbers for the network parameters can be searched. For the MIMO network, various filter sizes, number of filters and the number of nodes in the hidden layers were tried. Unfortunately, an extensive network parameters search could not be completed because each training regime takes one hour or more. Thus, the strategy employed was to make the network large and

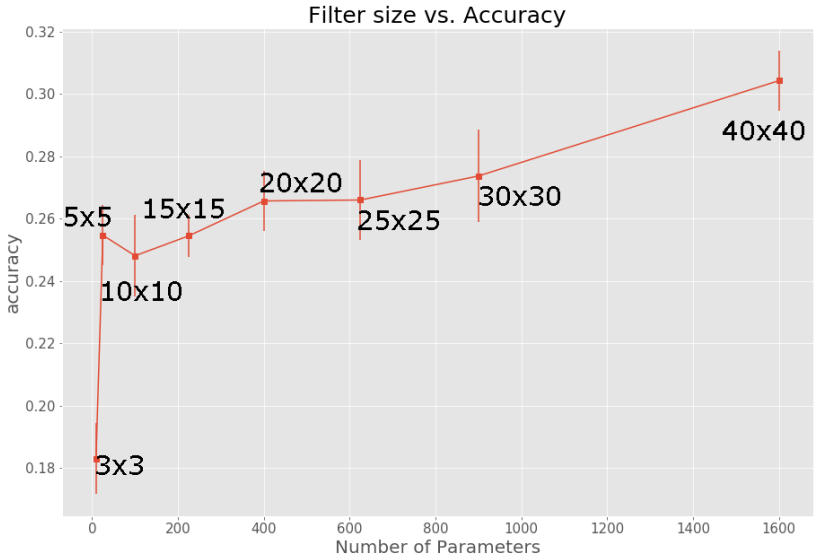


Figure 5.21: Network accuracy dependence on the filter size of the first convolutional layer is shown. 3x3 size filters were observed to be least accurate. We also observed that increasing the filter size tended to improve network accuracy but not significantly. 15x15 or 20x20 filter sizes were a good compromise between network accuracy and number of trainable parameters.

compensate for the large number of trainable parameters by generating a large number of training examples. This is necessary because the large network is prone to over-fitting the data. The final network parameters were set as listed in top row of Table 5.1. The accuracy of these PMT data only model was about 40% depending on the crystal.

5.9.3 MIMO Design

An additional information that could be used to improve the accuracy of the network is the summed pulse shape of the event. This is because α and β events have different pulse shapes and DP events are mostly α like events. Moreover, for DP events with more than 10 ch decay time, the summed pulse has good separation between the initial and secondary pulses, which the network can use to raise its accuracy. To see if the network can learn this information, the pulse only and the PMT data only models were combined.

Figure 5.22 shows the initial iterations of the combined model. The summed pulse branch is added to the PMT data network as an added information. In the first iteration, the second output was not added to the network. Although the extra branch was added, we did not observe any improvement in network accuracy. The accuracy of the various iterations of the combined model is listed in Table 5.3. The accuracy of the combined

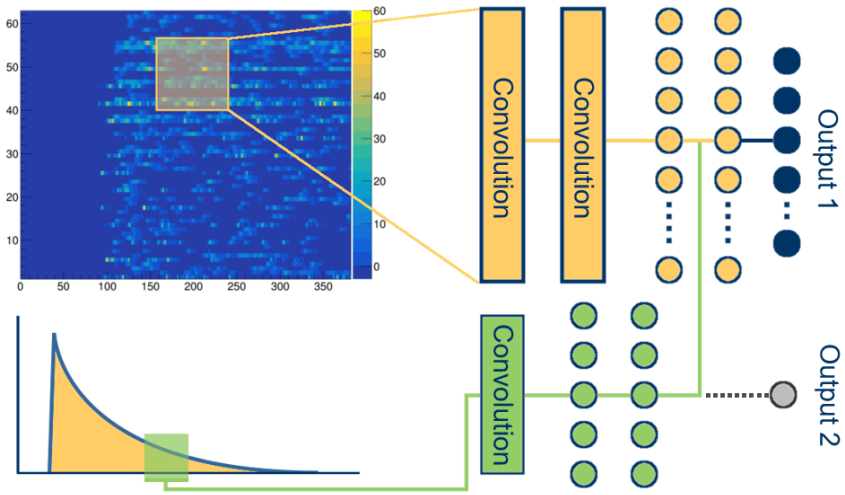


Figure 5.22: MIMO network predecessor design is shown. Adding pulse branch did not improve the network accuracy. Also adding output #2 near the pulse branch did not improve the network accuracy significantly. However, placing output #2 in parallel with output #1 improved the network accuracy significantly.

Network Design	Output 1 Accuracy	Output 2 Accuracy
PMT + Pulse	38.6%	-
PMT + Pulse + Output 2	39.6%	98.5%
MIMO Final	47.9%	99.75%

Table 5.3: Accuracy improvements of MIMO network by adding pulse branch. The position of output #2 was observed to be important to teach the network to extract useful information and improve accuracy.

model without the second output was 38.6%. Forcing the network to learn the difference between α and β pulses is apparently non-trivial.

To force the network to learn the difference, we added the secondary output, which is a single node designed to predict if the input is SP or DP. Note that the second output is not in the same position as the final MIMO design. The expectation was that the final hidden layer of the pulse branch would contain enough information to make the distinction between SP or DP. To train a network with multiple outputs, the loss function of each output is separately calculated and then combined to create an overall loss function. Additional control is allowed by defining weights when adding the two loss functions, but in this study, the loss functions were treated equally and thus the weights were set equal. The network trained in this way did not improve the accuracy of output #1, which predicts the decay time of the

two pulses. As shown in Table 5.3, output #2 accuracy is quite good with 98.5%. The reason why network accuracy does not improve is likely because the information necessary to distinguish SP and DP is not very large and thus not much information is stored in the final hidden layer of the pulse branch. This information is still enough to give good accuracy to output #2 but not enough to give improvement to output #1.

Finally, output #2 was moved to be in parallel with output #1, which is the final MIMO network design as shown in Fig. 5.5 with the network parameters listed in Table 5.1. With this configuration, the network accuracy improved to 47.9% for output #1. When output #2 is moved to the end of the network, it no longer has direct access to the information coming from the pulse branch. Thus, it must disentangle the pulse type information from the combined information of both branches. It appears that this process of disentangling the pulse type information also allows output #1 to learn the difference between α and β pulses, which results in an improved network accuracy.

The output #1 accuracy of 47.9% reported here is for testings done on crystal #1 events. An additional 5% to 10% accuracy was achieved on the generated DP and SP data by gradually reducing the learning rate(α parameter in Eqn. A.9) to a very small number. If the step size is large, then the gradient descent algorithm overshoots the minimum of the loss function and the weight parameter simply oscillates back and forth around the minimum. Thus, by allowing the learning rate to reduce to a very small number(0.0001 in this case), the parameters of the network can get closer to the minimum of the loss function. With this strategy, the final accuracy of the MIMO networks are those shown in Fig. 5.14 and Fig. 5.17.

5.9.4 MIMO Network For All Crystal

The MIMO network originally had 35 million trainable parameters. This number was reduced to about 20 million parameters by trimming the inputs as shown in Fig. 5.9 and Fig. 5.10. We attempted to create a single network that is capable of predicting DP and SP events from all crystals. The details are provided here. However, we found that a network trained for each crystal performs better than a much larger network trained on events from all crystal. This led us to train a single network for each crystal in this study.

Square convolutional filters(i.e. 15x15) are assumed to be learning pseudo TOF corrections when applied to events from a single crystal. When a network is trained on events from all crystals, the TOF to PMTs and start channels of these events are changing and thus square filters may become unable to adapt to these differing event conditions. Thus, we tried to separate the start channel information and the PMT signal information by using 1D filters along each axis, as shown in Fig. 5.23. There are two convolutional

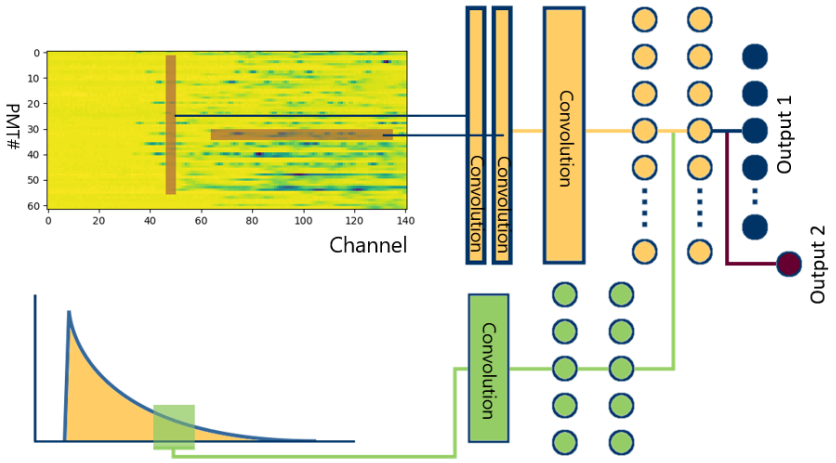


Figure 5.23: Horizontal and vertical filter CNN model design is shown. The reasoning behind separating the filters into horizontal and vertical was that the network would learn correlation between neighboring PMT signals with the vertical filters and learn each PMT rise time information with the horizontal filters.

layers in the PMT data branch that scan in the horizontal and the vertical axes. The idea is that the horizontal filter would scan for changes in signal in each PMT while the vertical filter would try to scan for start channel and TOF information. The outputs of these filters are simply stacked together and compared in the second convolutional layer.

We tried various filter sizes and number of filters with this model. The largest of these networks had 85 million trainable parameters, which is about 4 times larger than the MIMO network. However, due to training time being more than 3 hours per model design, an extensive model parameters search was not possible. Thus, we tried to make the network as large as possible, within the limitations of the hardware. This would give us an estimate of its performance and later we could reduce the size if necessary.

The network was trained on events from all crystals. We trained the network until we observed that the loss function reached its plateau point. Then we compared the large network with the MIMO network trained for crystal #1. That is, we applied the large network to the generated DP and SP events from crystal #1 and compare the accuracy to that of the MIMO model trained specifically for crystal #1. Table 5.4 shows the accuracy of the different models.

Aside from the large model with vertical and horizontal filters, we also trained two other models with horizontal only and vertical only filters. These networks were also trained on events from all crystals and tested on events from crystal #1. Table 5.4 shows that all of these additional models do

Network Type	MIMO	Large	Horizontal	Vertical
Number of Parameters	20M	85M	45M	20M
Accuracy on Cr#1	46.9%	33.2%	29.0%	29.5%

Table 5.4: Accuracies of various testing models are shown. The MIMO network is trained for crystal #1 and tested on generated DP and SP events in crystal #1. The other models are trained on generated events from all crystals but tested on crystal #1 events to compare to MIMO network results.

not perform better than the MIMO model trained specifically for a single crystal. Because of hardware limitations, we were not be able to train any larger model. Thus, our decision was to train a MIMO network for each crystal. In practice, as discussed in the last section, 3 MIMO models are trained per crystal and we take the average of these models as the final prediction.

5.10 Neural Network Framework and Hardware

The neural networks in this study were trained using Keras[37] with TensorFlow backend. TensorFlow is a machine learning framework developed by Google and has support for GPU acceleration. Keras is an extension of TensorFlow that aims to make TensorFlow easier to use. It is also maintained by Google. The TensorFlow version used in this study is 1.12. For more information on Tensorflow, please see [38]. To support GPU acceleration for network training, TensorFlow uses cuDNN and CUDA[39] libraries developed by NVIDIA. cuDNN and CUDA are C extensions that allow access to GPU hardware developed by NVIDIA. The versions used in this study were cuDNN 7.5 and CUDA 9.1.

The hardware used in this study are Intel i7 4770K CPU with 16GB of DDR3 RAM. The GPU used was a GeForce GTX1060 6GB made by NVIDIA. The computer runs Ubuntu Linux 16.04LTS. The total time required to train all CANDLES models was about 4 weeks. This includes time required to generate training data and copy the data to training computer. A significant effort was made to optimize various code to reduce the training time to 4 weeks, which we estimated to be about 4 months initially. However, additional optimization is possible, especially with parallel programming to read training data in one thread and train the network on a separate thread.

5.11 Summary

In summary, MIMO networks were trained for CaF_2 crystals to predict the decay time of DP events up to 30 channels. The networks were trained on

SP and DP events generated by combining donor events. The donor events were selected from ^{208}Tl and $^{214/215}$Po events from regular CANDLES runs and ^{88}Y events from calibrations runs. For each crystal, 120,000 training SP and DP events were generated, separated into 3 data sets based on the ratio of SP to DP events. For each crystal, 3 MIMO networks were trained and during application of the networks for data predictions, the average of the 3 network predictions is taken as the final prediction.

We observed over 50% top1 accuracy and over 95% top3 accuracy for almost all crystals. The accuracies were obtained on more generated DP and SP events and thus does not carry over to real data. Nevertheless, we observed good performance by the networks. The poor top1 accuracy is likely due to the error in the training data. The start channel of an event is not always accurately determined and thus generated DP events can have incorrect training label. Thus, the network predictions are good enough given the training data. The real network accuracies are determined using real CANDLES data in the next chapter.

Several other models were trained and tested during MIMO network development. We found that adding the pulse branch of the MIMO network as well as output #2 significantly increases network accuracy. Also, we trained a large single network to make predictions on all crystals. However, we found that a single network trained for each crystal performs significantly better than the large network for all crystals. Thus, for this application, networks were trained for each crystal.

Chapter 6

Application to CANDLES Data

The purpose of the previous chapters were to develop methods to detect and reject multi-crystal events and double pulse events. Multi-crystal events are generated when γ particles from the decay of ^{208}Tl escape the original crystal and deposit energy in nearby crystals. The delayed coincidence method may not work in this case if the event position is incorrectly determined in a different crystal. To detect these multi-crystal events, we made hit patterns(HP) in Chap. 4. In this chapter, we will apply the hit patterns to CANDLES data to detect and reject multi-crystal events.

The neural network based method was developed to detect and reject double pulse(DP) events arising from the chain decay of ^{212}Bi $\rightarrow$ ^{212}Po $\rightarrow$ ^{208}Pb. The method was developed using generated DP and single pulse(SP) events. In this chapter, we will apply the neural network method to check if it can generalize to real CANDLES data. Finally, the purpose of this study is to develop methods for reduction of CANDLES backgrounds. To this end, we report the efficiency of the two methods to reduce main backgrounds of the experiment.

6.1 CANDLES Data Selection

In this study, we use nearly 430 days of data collected by CANDLES experiment. Figure 6.1 shows the energy spectrum of this data set. We apply several cut conditions to reduce this data and each successive cut is shown in the figure. We first remove all events that are reconstructed to be outside 2σ range of a crystal(Sec. 3.4). Next, we only keep β like events by applying a PSD$_\beta$ cut(Sec. 3.6) of less than 1.5σ. This cut significantly reduces the number of events in almost all energy ranges. In the high energy region, the PSD$_\beta$ cut removes ^{208}Tl events with LS signal as well as some DP events due to it having alpha-like pulse shape. Next, we remove DP

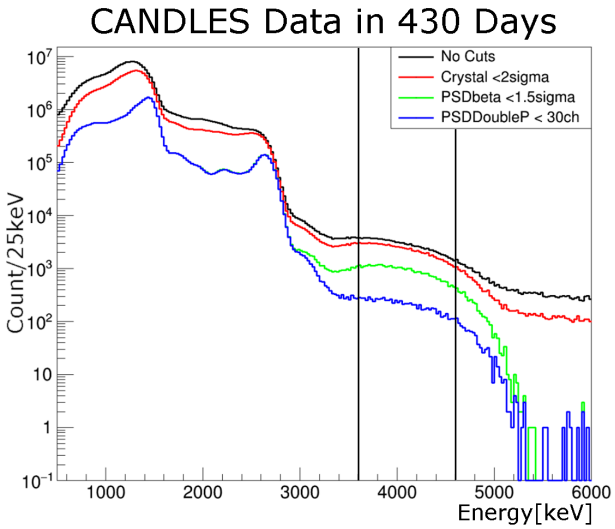


Figure 6.1: CANDLES experiment data for nearly 430 days is shown. The black spectrum shows the data without any cuts applied. We apply several cuts to reduce the size of data used in this study. The first cut is a crystal cut of less than 2σ from center of a crystal. Next, we select β-like events by applying a $\text{PSD}_\beta < 1.5\sigma$ cut. Double pulse events with decay time greater than 30 ch is removed with the PSDDoubleP cut. Finally, we restrict energy region to 3.6 MeV to 4.6 MeV due to network trained energy range considerations.

events with decay time greater than 30 ch(60 ns) using the PSDDoubleP method discussed in Sec. 5.2. The CANDLES MIMO network was designed to label DP events with decay time of less than 30 channels.

With these cuts applied, the remaining events are shown(blue spectrum) in Fig. 6.1. This final selection contains mostly DP events with decay time of less than 30 ch, and single crystal and multi-crystal events from decay of ^{208}Tl. Potential $0\nu\beta\beta$ events are also included in this selection along with β-like backgrounds from any other sources. Because the MIMO network is trained on generated DP and SP events with energy distributions centered around 4.0 MeV and 4.4 MeV(Sec. 5.6), we restrict the data to a range of 3.6 MeV and 4.6 MeV. Beyond this range, there is no guarantee that the network will work because it was not trained on a large energy range. The energy selection range is shown as vertical bars in Fig. 6.1.

The event selection with these cuts contains 8,360 events. The number of events per crystal is shown in Fig. 6.2. We see that fewer events are observed in central numbered crystals. This is by design because we place low contamination crystals at the center and use the higher contamination crystals as an additional layer of shielding. Therefore, middle layer crystals are those that have less activity. We also removed events from crystal #11

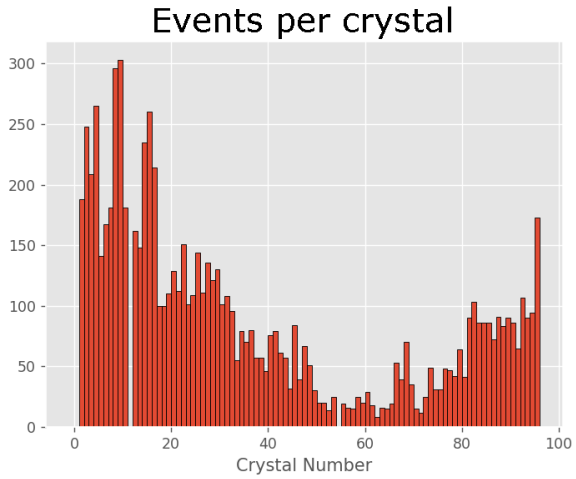


Figure 6.2: Events per crystal surviving the various cut conditions. In this study, we ignore analysis on crystal #11 due to its very high contamination, and crystal #54 due to its inconsistent quenching factor.

and crystal #54 from this analysis. Crystal #11 has very high contamination and is used for calibration purposes. Crystal #54 has inconsistent quenching factor compared to the rest of the crystals.

6.2 MIMO Application to CANDLES Data

In this section, we present the results of applying the MIMO network to CANDLES data. First, we show the main result, which is relatively self-explanatory. The proceeding sections provide a systematic estimate of the network efficiencies. The two efficiencies of interest are SP selection efficiency and DP rejection efficiency.

6.2.1 Overall Result

As a reminder, the MIMO network has 31 output nodes and each node predicts the decay time of DP events, in units of 1 channel(equal to 2 ns). When we make histograms of the network predictions, 31 bins are generated, with bin#0 counting the number of SP events while bins #1 to #30 are counts of DP events with varying decay times. DP events with decay time of less than 30 channel is selected by the PSDDoubleP method, which totals 8360 events from 430 days of data, as shown in the previous section. However, we can also select DP events with decay time greater than 30 channels using the PSDDoubleP method. This provides us with the exponential decay curve of the ^{212}Po $\rightarrow$ ^{208}Pb event, which has a half-life of 299 ns. We apply a fitting to the exponential decay curve and then extend the fitting

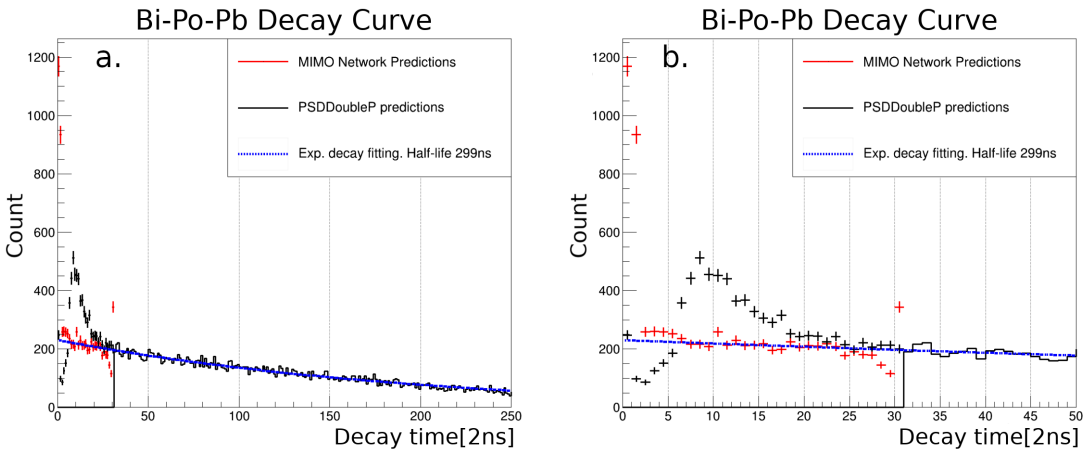


Figure 6.3: PSDDoubleP method predictions of CANDLES events up to 250 ch. The exponential decay curve of ^{212}Po $\rightarrow$ ^{208}Pb events is well predicted by the method above 30 ch. a.) An exponential decay function is fitted in the 31 ch to 200 ch range. The decay constant is set to the known value of 299 ns. Only normalization and additive constant values are fitted. The χ^2 values of the fitting is 150.2/169. The exponential decay curve is then extended to below 30 ch. This gives the expected number of events per bin. b.) The same figure is shown in the 0 ch to 50 ch range. The MIMO network is applied to events in bin#30 and under and the results are drawn in red crosses. We can observe that the PSDDoubleP predictions are not consistent with the expected curve while the MIMO network predictions follow the expected curve closely. The excess events in bin#0 and #1 are due to non DP events in CANDLES data. These new predictions show the improvement brought on by the MIMO network over PSDDoubleP method.

to below 30 channel region to obtain the expected DP decay curve. Thus, the main result of the study for DP event reduction can be summarized in Fig. 6.3.

In Fig. 6.3.a, DP events selected by PSDDoubleP method with decay time of up to 250 channels are shown in black. The DP events with decay time of greater than 30 channels are shown in black colored bars while the events with less than 30 channel decay time are shown in black crosses. The MIMO network is applied to the events with decay time less than 30ch and new predictions are made. These are the 8360 events selected in the previous section and are drawn as red crosses. An exponential decay plus a constant function is fitted to the DP events from bins #31 to #200. Because the decay has known half-life, we fix the decay constant to the known value of 299 ns. The normalization and the added constant is allowed to vary in this fitting. The constant term is added to compensate for the DP events

rejected by the crystal cut and the PSD_β cut. The χ^2 value of the fitting is 150.2 in 169 bins. We observe that the PSDDoubleP method predicts DP events very well above 30 channels.

The fitted exponential curve is then extended to below 30 ch, as shown in Fig. 6.3.b. We observe that the PSDDoubleP method predictions diverge from the expected line significantly below 20 ch bin. This was a known issue which motivated this study. From the figure, we observe that the MIMO network predictions are very consistent with the expected curve. There is an excess of events in bin#30 which we can understand to be events that should be predicted into bins #27, 28 and 29. However, this network error is not very important because our overall goal is to reject these DP events. From bin#27 down to bin#5, we can observe very close match between the MIMO predicted events and the expected curve. The χ^2 value of the observed counts and the expected curve in bins#5 to #27 is 22.64 in 23 bins. This χ^2 value is obtained by comparison and not fitted value. Thus, the network predictions are very consistent with the expected curve. However, bins #2,3,4,5 does seem to contain some excess of events due to MIMO network error. We will confirm this shortly and estimate the network error. Finally, we observe the most number of events in bins#0 and #1. This is expected because the data set contains significant number of SP events aside from the DP events. Majority of these event are from ^{208}Tl, as we will confirm in the next section.

Due to network inaccuracy, we observe a larger number of events in bin#1 than the number from the expected curve. The network appears to make errors by placing SP events into bin#1. Because of the programming error mentioned in Chap. 5, DP events with 0 decay time was not specifically generated. Thus, the network was not trained to place DP events into bin#0, although these events are present in data. We confirm in App. C that the network has learned to place 0 decay time DP events into bin#1. Moreover, in App. D, we show that almost all 0 decay time DP events are in bin#1. Thus, the excess number of events in bin#1 are partially due to 0 decay time DP events being predicted into bin #1.

Figure 6.3.b shows that the MIMO network makes predictions that are consistent with the expected number of events. It appears that the network predictions are accurate enough to be used down to bin#2. The proceeding sections are aimed at confirming the accuracy of the network and estimating the efficiency. Comparisons with PSDDoubleP method is made to show the improvement provided by the network over the existing method.

The results of both methods applied to the 8,360 events are shown in Fig. 6.4. The x axis shows the PSDDoubleP predicted decay times of the events. The y axis shows the MIMO network predicted decay times. Output of 0 indicates that the event is an SP event. We can immediately see that both methods have very good agreement on events with decay time of more than 10 channels. This is an important result because it shows that the neural

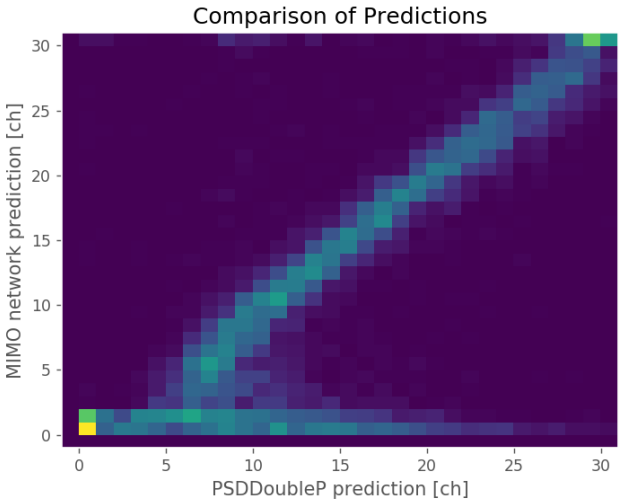


Figure 6.4: Comparison of MIMO network and PSDDoubleP method predictions. The x axis represents the decay time predictions by the PSDDoubleP method. The y axis represent the decay time predictions of the same events by the MIMO network. We observe good linearity between both methods for events with decay time of 10 ch or more.

network, trained on generated false DP and SP events, can generalize to real CANDLES data well.

We next look at the predictions that the methods disagree on. If both methods worked perfectly, then we expect to see a completely diagonal line of predictions. However, we observe that the PSDDoubleP predictions break this symmetry by "smearing" the predictions into higher bins. In other words, the PSDDoubleP method is predicting higher decay times for the DP events than the actual decay time of these events. More importantly, the figure indicates that the PSDDoubleP method is incorrectly predicting SP events as DP events. To obtain good separation of SP and DP events by PSDDoubleP method, additional information must be used in conjunction to keep the SP events predicted into higher bins. These figures show the overall improvement brought by the MIMO network over the previous method. In the next two sections, we estimate the network accuracy based on external information.

6.2.2 SP Efficiency

To estimate SP prediction accuracy by the MIMO network and its errors, we need a selection of known SP events. The cut conditions used to select events shown in Fig. 6.1 do not remove ^{208}Tl events and these events are present among the selected 8360 events. These ^{208}Tl events are SP events

and produce a β particle and several γ particles(decay scheme is shown in Fig. 4.1). Thus, by selecting these events, we are able to observe how they are predicted by the MIMO network and estimate the network SP accuracy.

^{208}Tl nuclei is produced from an α decay of ^{212}Bi and it decays to ^{208}Pb with a half-life of 3 minutes. We can effectively select ^{208}Tl events by observing the parent's α event in the same crystal. This is known as the delayed coincidence method. Figure 6.5.a shows the energy spectrum of events preceding and proceeding the analysis selected 8,360 events. This plot is obtained by creating histograms of all events occurring 720s before and after each one of the candidate SP or DP event. Because we are looking for the α particle of the parent nuclei to select ^{208}Tl events, we observe these α events in the preceding spectrum. The proceeding spectrum has no excess α events and thus provide good estimation of the expected number of backgrounds.

The search duration of 720 seconds is equivalent to 4 half-lives of ^{208}Tl. The excess events at 1.7 MeV in the preceding spectrum are the α events of the parent ^{212}Bi. We obtain the mean energy and the deviation by taking the difference of the preceding and proceeding spectrums and fitting a Gaussian. Figure 6.5.b shows the Gaussian fitting. The mean value of the Gaussian is 1.69 MeV with standard deviation of 58.7 keV. In this study, we select α events with two standard deviations from the mean. By using the delayed coincidence method, we can identify ^{208}Tl events among the selected 8,360 events. Since these are SP events, we expect to see these events appear in bin#0 of both DP detection methods.

Before we move on, we first estimated the rate of accidental α coincidence. This rate needs to be estimated under similar event selection conditions. Thus, we use the same cut conditions used in Fig. 6.1 but instead select events with decay time of 40 ch to 70 ch. That is, crystal cut $< 2\sigma$, PSD$_\beta$ $< 1.5\sigma$, PSDDoubleP between 40 ch and 70 ch and energy range of 3.6 MeV to 4.6 MeV. Figure 6.6 shows the preceding spectrum of the α coincident events and non-alpha coincident events, normalized by the number of events. The α energy range is shown in gray vertical bars. The accidental α coincidence rate is calculated as the ratio of alpha-coincident-events over non-alpha-coincident-events and this has a value of (0.107 ± 0.005).

We can also confirm that these selected α events are indeed from the parent ^{212}Bi by calculating the half-life of the decay. This can be done by fitting an exponential decay to the delta time that the α events appeared in. Figure 6.7 shows the fitting for the half-life. The x axis represents the time searched into the preceding energy spectrum for the α event. With this fitting, we obtain a half-life of (209.4 ± 21.4) s, which is consistent with the literature value of 183.18 seconds. Thus, we can accurately obtain ^{208}Tl events by finding the preceding α event.

Figure 6.8 shows the MIMO network and PSDDoubleP predictions separated by α coincident events and non-alpha coincident events. Since the α

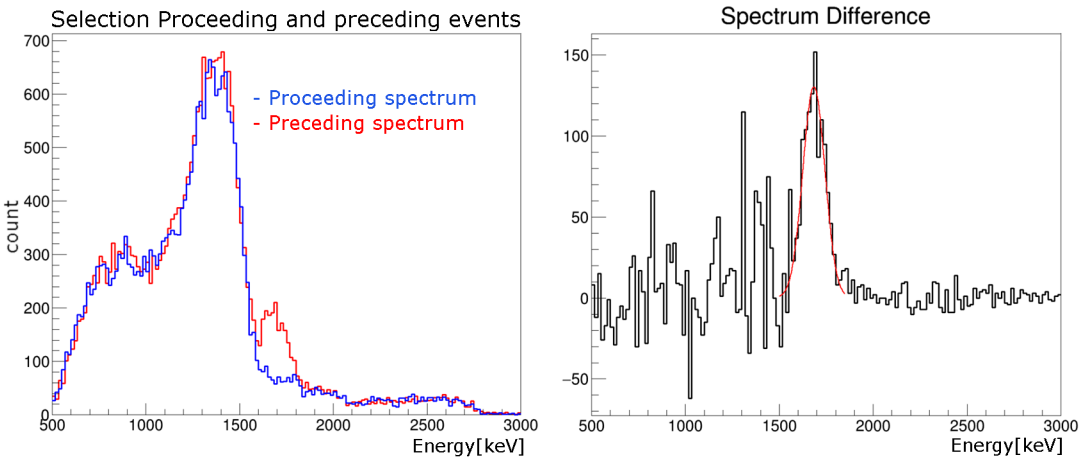


Figure 6.5: The energy spectrum of events preceding and proceeding the analysis selected events is shown(a). Because ^{208}Tl events are present in large numbers in the selection, we can observe the preceding α events. We determine the mean and deviation of the α peak by Gaussian fitting on the difference in spectrums(b).

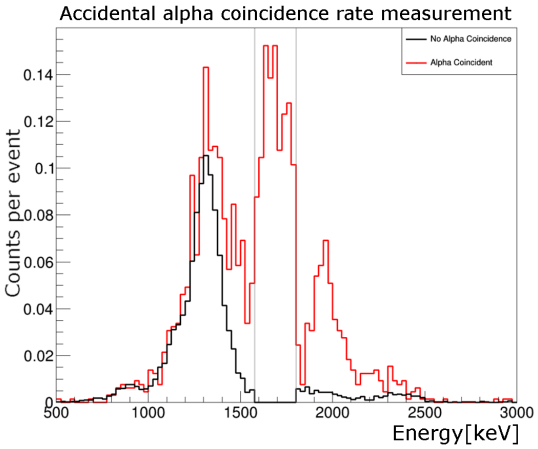


Figure 6.6: Energy spectrum of α coincident events and non-alpha coincident events. The energy spectrums are normalized by the number of events. The peak centered around 1.7 MeV corresponds to the α events of the ^{212}Bi parent.

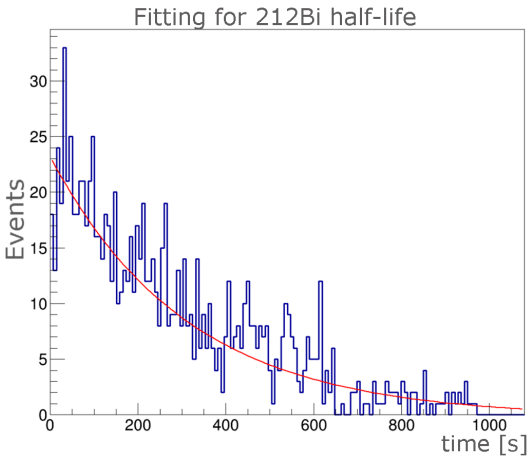


Figure 6.7: Half-life of ^{212}Bi is checked by fitting exponential decay function to the α appearance times. The obtained half-life is (209.4s $\pm$ 21.4) s. The literature value is 183.2s.

coincident events are known SP events from ^{208}Tl, it allows us to estimate the SP accuracy. We observe that the MIMO network is predicting most number of SP events in bin#0 and bin#1, as expected. There are α coincident events in all bins but these are due to accidental coincidence. We will confirm this momentarily. On the other hand, the PSDDoubleP method is not predicting nearly enough SP events in bin#0.

To confirm that the α coincident events in higher numbered bins are due to accidental coincidence, we take the ratio alpha-coincident-events over non-alpha-coincident-events in each bin. This is shown in Fig. 6.9 for both methods. Previously, we estimated that the rate of accidental α coincidence is (0.107 $\pm$ 0.005) using an independent data set with similar cut conditions. This accidental rate is shown in the figure with red dashed line. We observe that the ratio of MIMO network predictions conforms to this line very well above bin#5, indicating that the α coincident events in higher numbered bins are due entirely to accidental coincidence. We estimated the accidental coincidence rate from bin#8 to bin#30 of the MIMO network predictions and obtained a value of (0.113 $\pm$ 0.005), which is consistent with the independently estimated accidental coincidence rate.

Figure 6.9 shows that the network predicts excess α coincident events up to bin#5. There are also excess α coincident events in bins#6,7 but these appear to be within statistical errors. However, in this study, we include these bins as part of network error bins. That is, the network is assumed to incorrectly place SP events up to bin#7 but not in bins above that. On the other hand, the ratio produced by PSDDoubleP method is not consistent with the expected accidental α coincidence rate, shown in Fig. 6.9.b. This indicates that the PSDDoubleP method is incorrectly

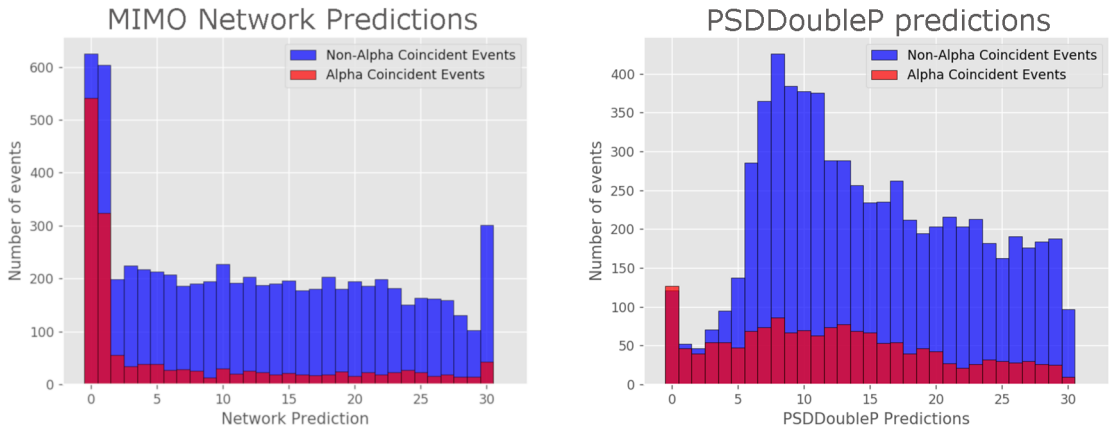


Figure 6.8: The MIMO network(a.) and PSDDoubleP method(b.) predictions are shown separated by coincidence with α event. These distributions are obtained on the same data set and thus represent the difference in methods predictions only. The α coincident events are primarily known SP events from the decay of ^{208}Tl.

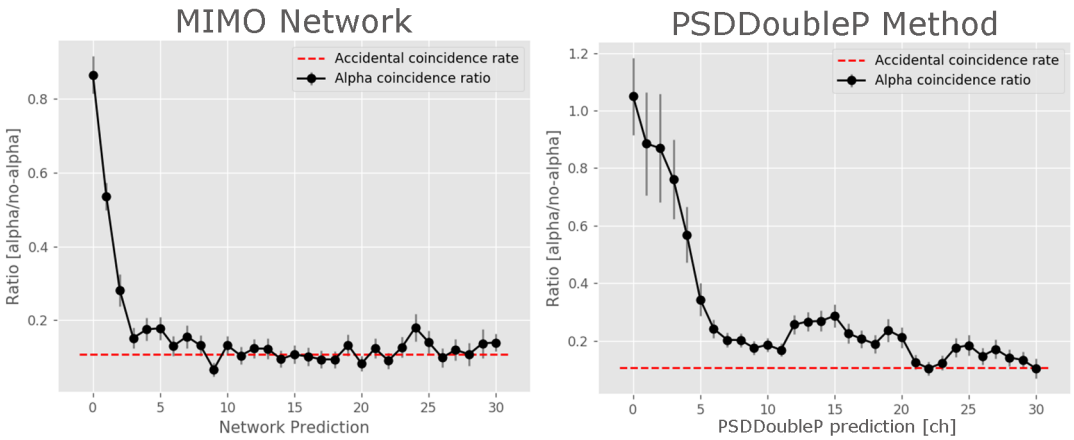


Figure 6.9: The ratio of α coincident over non-alpha coincident events in each prediction bin is shown. The ratio is high for MIMO network bins#0 and #1, indicating that the network is correctly classifying SP events with some errors. The ratio is consistent with accidental coincidence rate for MIMO network for bin#5 and above. Conversely, the ratio produced by PSDDoubleP method is higher than accidental coincidence rate in most bins.

Bin#	0	1	2	3	4	5	6	7	8	9	10
MIMO Network	0.58	0.90	0.94	0.95	0.97	0.99	0.99	1	1	1	1
PSDDoubleP	0.13	0.18	0.22	0.28	0.33	0.37	0.41	0.45	0.50	0.53	0.56
Bin#	11	12	13	14	15	16	17	18	19	20	21
MIMO Network	1	1	1	1	1	1	1	1	1	1	1
PSDDoubleP	0.59	0.64	0.69	0.74	0.79	0.82	0.85	0.87	0.90	0.93	0.93
Bin#	22	23	24	25	26	27	28	29	30		
MIMO Network	1	1	1	1	1	1	1	1	1		
PSDDoubleP	0.93	0.94	0.95	0.97	0.97	0.99	0.99	1	1		

Table 6.1: Single pulse selection efficiency of both methods by bin number. The MIMO network achieves about 60% SP selection accuracy in bin#0 and 90% accuracy by bin#1. This improvement in SP selection accuracy allows us to reject more DP events with higher confidence than the PSDDoubleP method.

placing SP events into bins for DP events. To keep these SP events in data, external information must be used. This suggests that the MIMO network can achieve higher efficiency without using external information. Thus, the MIMO network is able to provide improvement in DP and SP rejection and selection efficiencies.

Given the independently estimated accidental α coincidence rate of (0.107 $\pm$ 0.005) and the number of non-alpha coincident events in each prediction bin, the number of accidental α coincident events in each bin is estimated. We then subtract these accidental α coincident event counts from the observed α coincident events to obtain the count of only ^{208}Tl events. This is shown in Fig. 6.10. For the MIMO network case shown in Fig. 6.10.b, after the accidental α coincident events count are subtracted, we see that most SP events are correctly predicted in bin#0. Due to network inaccuracy, we also observe significant count of SP events in bin#1. Bins#2,3,4,5 also contain some number of SP events but beyond bin#5, we see that the excess α event count is within statistical fluctuations. In Fig. 6.10c,d, the distribution of SP events predicted by the PSDDoubleP method is shown. The PSDDoubleP method is observed to be predicting SP events as DP events in significant quantities up to bin#20. This shows that the MIMO network predictions can be used more reliably for events with short decay times.

From the ^{208}Tl counts in Fig. 6.10.b, the MIMO network SP selection efficiency is estimated. Figure 6.11 shows the SP selection efficiency of both methods and Table 6.1 lists the values of each bin. This is one of the main results of this study. We see that the network SP selection accuracy in bin#0 is nearly 60%. The accuracy increases to 90% if we also include bin#1 predictions. The MIMO network could be applied to reject DP events down to and including bin#2, which would see a 10% reduction of SP events. This

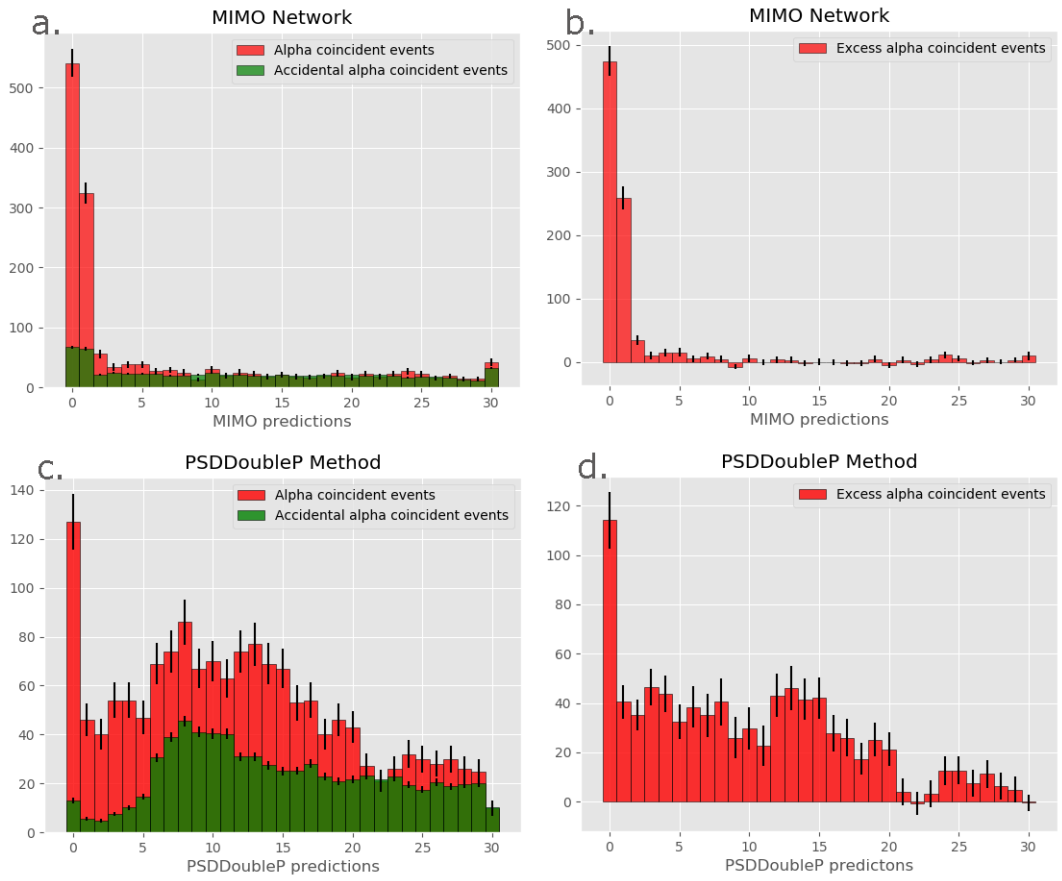


Figure 6.10: The accidental α coincident event counts are removed to obtain ^{208}Tl only counts. Since ^{208}Tl events are SP events, it allows us to estimate network accuracy. a.) Number of α coincident events are shown in red. Green bars represent the expected number of α coincident events in each bin due to accidental coincidence. These numbers are estimated by the independently obtained accidental coincidence rate of (0.107 ± 0.005) and the number of events in each bin in the non-alpha coincident spectrum. b.) ^{208}Tl event only counts are shown per bin. This is obtained by subtracting the accidental α coincident events from the observed number of α coincident events. c.) The same events predicted by the PSDDoubleP method. Accidental α coincident events count is also obtained using the accidental coincidence rate. d.) ^{208}Tl only events count per bin predicted by the PSDDoubleP method.

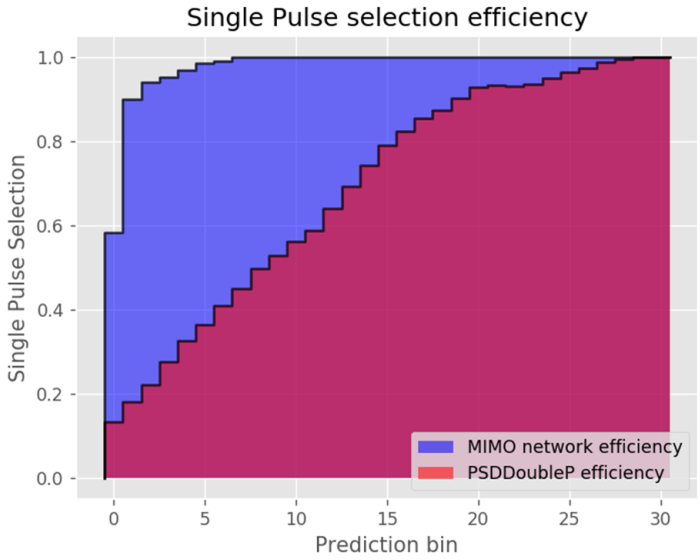


Figure 6.11: Single pulse event selection efficiency is shown. The network can effectively select 90% of single pulse events by only keeping bins#0 and #1. The PSDDoubleP method needs to keep up to first 10 bins or 20 bins to match the single pulse event selection efficiency of the network in bin#0 or bin#0 plus bin#1.

improvement in SP selection accuracy allows us to remove much more DP events with more confidence than the PSDDoubleP method.

As a summary, in this section we estimated the SP selection efficiency of the MIMO network. Two main external information was used. The first was the selection of known SP events in the form of ^{208}Tl backgrounds. These events are selected by the delayed coincidence method. The accidental α coincidence rate was also estimated using an independent data set. Next, we showed that the observed α coincident event counts were consistent with expected accidental α coincident counts in DP bins. Next, the accidental α coincident event counts were subtracted from the observed α coincident counts to obtain only ^{208}Tl events. Since these are SP events, we were able to use the observed ^{208}Tl counts to estimate the MIMO network SP efficiency. Because an event's coincidence with an α event is an independent and strong evidence, the use of these events to estimate network efficiency is justified. Moreover, shape indicator(SI) values were used as an additional confirmation in App. D to show that the above results are consistent with SP predictions made by SI. We also confirmed that there were no significant number of SP events in the DP bins through additional α coincidence check in App. E. The conclusion of this section is that the MIMO network places SP events in bins#0 and #1 with 90% accuracy.

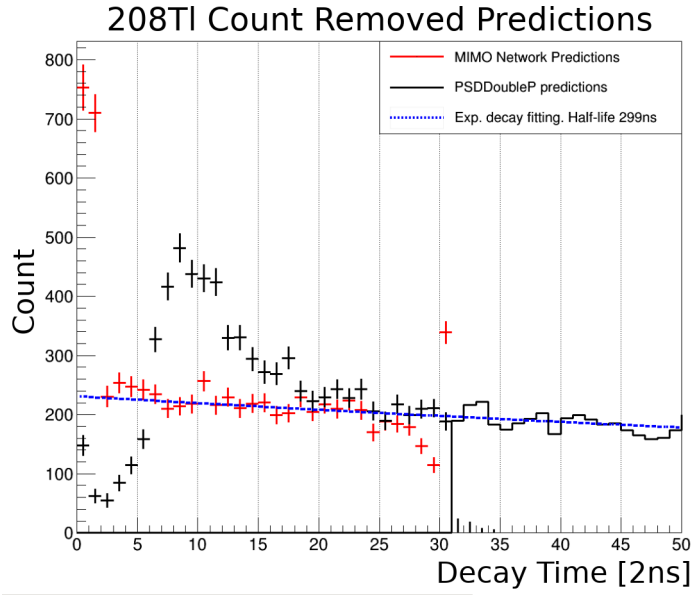


Figure 6.12: The observed DP predictions are shown after removing ^{208}Tl counts from each bin. The MIMO network predictions are shown in red crosses and is observed to be in good agreement with the expected curve. The χ^2 value of the comparison of the expected curve to the observed counts by MIMO network is 25.11/26 in bin#2 to #27.

6.2.3 DP Rejection Efficiency

From SP efficiency, we observed most of the SP events in bin#0 and bin#1. The events in bins#2 to #30 are almost all DP events. In Fig. 6.3.b, the MIMO network predictions were shown with the expected DP decay curve. We also obtained the ^{208}Tl only counts in Fig. 6.10.b. The ^{208}Tl counts are subtracted from the selected 8360 events and the resulting event predictions are shown in Fig. 6.12. This figure is obtained by bin-wise subtracting ^{208}Tl counts in figures 6.10.bd from the total prediction distribution in Fig. 6.3.b. As expected, the remaining DP predictions in each bin is more consistent with the expected curve than before. The χ^2 value of the comparison of the expected curve to the observed counts is 25.11/26 in bin#2 to #27. Thus, if we can confirm that the number of SP events in bins#2 and above are consistent with the SP efficiency curve, then we can conclude that the network predict number of DP events consistent with the expected curve down to decay time of 2 channels.

There are SP events remaining in the data, observed from the excess number of events in bins#0 and #1. The events in these bins are a mix of DP events and SP events that were not rejected for by the delayed coincidence method. From the SP efficiency observed in Table 6.1, about 10% of these

events are expected in bins above #2. This is confirmed by the shape SI analysis in App. D. From SI analysis, we observed (80.2 ± 15.0) SP events remaining in bins#2 and above, which is consistent with the SP efficiency. Moreover, we counted (772.8 ± 37.4) DP events in bins #2,3,4,5 which is consistent with the (809.0 ± 28.2) DP events expected in these bins from to the exponential decay curve. Because the observed SP and DP counts are consistent with the SP efficiency and the expected number of DP events from the exponential decay curve, we conclude that the network is consistent with the expected decay curve from bin #2 to #30. For detailed analysis, please see App. D.

Note that because accidental α coincident events can not be separated from α coincident ^{208}Tl events, in further analysis and in the analysis results above, we removed all α coincident events. This reduces observed DP counts in each bin by the accidental coincidence rate of (0.107 ± 0.005). The expected exponential decay curve is scaled to compensate and the uncertainty ranges are increased for the exponential curve as a result.

In this study, the recommended cut point to reject DP events is set to bin#2 and above. From SP efficiency, this cut point rejects 10% of the SP events. But, the expected DP rejection efficiency 93.2%, obtained from the expected exponential decay curve. Because the network was not specifically trained to classify 0 decay time DP events into bin#0, we saw that the network places these events into bin#1 in App. C. However, about 10% of the 0 decay time DP events were still expected in bin#0. This was not confirmed by SI analysis in App. D, which showed no significant count of DP events in bin#0 and also showed all DP events to be in bin#1. Because this discrepancy in DP count in bins#0 and #1 could not be resolved, the recommended cut point is set to bin#2 and above for DP events. Although SI information is more reliable and independent than the prediction checks done in App. C, the number of events in bins#0 and #1 are not enough to obtain reliable SI fitting. This inconsistency could be resolved in the future with more data. However, we confirmed that the DP counts in bins#2,3,4,5 were consistent with the expected curve, and thus we can place a cut point at bin#2.

Next, the DP rejection efficiency is estimated in Fig. 6.13. The expected number of DP events in each bin is obtained from the exponential decay curve. The observed curve is obtained from the MIMO network predictions. The events in bin#0 and #1 are estimated from SI analysis but could not be confirmed with another method. The total number of DP events in these bins is consistent with the expected number in bin#0 and #1. But the actual distribution of DP events in these bins is not confirmed. However, bin#2 is the suggested cut point and events in bin#2 and above are confirmed to be consistent with the expected decay curve. Due to the good agreement with the expected curve, this curve can be used to estimate the network's DP rejection efficiency. In Table 6.2, the DP selection efficiency per bin is

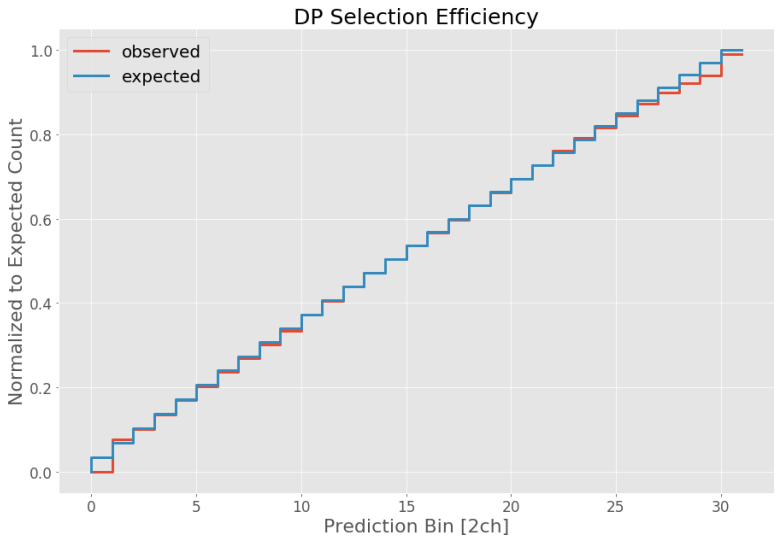


Figure 6.13: The DP event selection efficiency by MIMO network is shown in red curve. The expected DP count obtained from the exponential decay curve is shown in blue curve. The observed count is normalized by the expected curve. The observed events in bin#0 and #1 are estimated from SI analysis. However, the actual distribution of DP events in these two bins is not confirmed.

shown for the observed count and the expected curve.

Our goal is to reject as many DP events as possible without removing too many SP events. For this reason, bin#2 is a good cut point. We observe that only about 7% of DP events remain in bins#0 and #1. Also, due to the good agreement with the expected exponential decay curve, the cut point can be changed and the remaining DP events can be estimated easily. Thus, the MIMO network provides a good improvement on PSDDoubleP method by allowing rejection of DP events down to bin#2 without sacrificing a large number of SP events.

In summary, the DP efficiency is estimated in this section. From the good agreement between the network predictions and the expected curve, the expected curve was used to estimate the expected DP rejection efficiency. The number of DP and SP events in each bins is confirmed to be consistent with expected numbers by using multiple independent information. The SP count in each bin is confirmed using known SP ^{208}Tl backgrounds, accidental α coincidence rate, SI analysis in App. D and additional α coincidence study in App. E. By removing the ^{208}Tl event counts, the remaining events distribution was obtained, which are mostly DP events and SP events other than ^{208}Tl. The remaining DP events are consistent with the expected decay curve, with a χ^2 value of 25.11/26 for bins#2 to #27. We also checked if

Bin#	0	1	2	3	4	5	6	7	8	9	10
Observed	0	0.078	0.10	0.14	0.17	0.20	0.24	0.27	0.30	0.33	0.37
Expected	0.03	0.069	0.10	0.14	0.17	0.21	0.24	0.27	0.31	0.24	0.37
Bin#	11	12	13	14	15	16	17	18	19	20	21
Observed	0.40	0.44	0.47	0.50	0.54	0.57	0.60	0.63	0.66	0.70	0.73
Expected	0.41	0.44	0.47	0.50	0.54	0.57	0.60	0.63	0.66	0.69	0.73
Bin#	22	23	24	25	26	27	28	29	30		
Observed	0.76	0.79	0.82	0.84	0.87	0.90	0.92	0.94	.99		
Expected	0.76	0.79	0.82	0.85	0.88	0.91	0.94	0.97	1		

Table 6.2: The portion of double pulse events appearing in each bin is shown for the MIMO network and the PSDDoubleP method.

there were any SP events remaining in the DP bins by SI analysis and additional α coincidence study. We found that about 10% of the remaining SP events are present in bins#2 and above, which is consistent with the expected number from the SP efficiency. Thus, we conclude that the MIMO network predictions follow the expected curve and used the expected curve as the efficiency predictor.

6.3 Rejection of Multi-Crystal Events

In Chap. 4, we created hit patterns(HPs) of all crystals for the purpose of detecting and rejecting multi-crystal(MC) events. In this section, we apply the HPs to the remaining single pulse events from the previous section. We will also estimate MC event rejection efficiency using available data. As a reminder, we created HPs of each crystal and we fit the HPs to new events and obtain the χ^2 of the fittings. If the event is a single-crystal(SC) event, then it will have good fit to a single HP and the second best fitting HP will have a large χ^2 value. On the other hand, if the event is MC, then two HPs will have relatively good χ^2 values and their difference will be small.

In Fig. 6.14 we see the χ^2 value of the best fitting crystal versus the χ^2 value of the second best fitting crystal distribution of the non-alpha coincident events in various bins. A line of slope 2 is draw as a reference in all figures. We can see a significant number of events below the reference line for events in bin#0 and #1. These are MC candidate events. The MC candidate events occurring in bin#1 are incorrectly classified SP events as we saw in the last section. We see that these events rarely occur among events in bins#2 to #30. This is because events in these bins are DP events from the ^{212}Bi $\rightarrow$ ^{212}Po $\rightarrow$ ^{208}Pb decay. These are known SC events and thus we expect to see them above the reference line. However, ^{212}Bi $\rightarrow$ ^{212}Po decay also emits low energy γ particles and it has a chance to generate MC events. The events below the reference line among the DP bin events may

be due to this type of MC events. But the majority of the MC candidate events occur in bins#0 and #1, which is expected. Thus, we can visually identify the MC events among the remaining SP events.

In Fig. 6.15, we see the same kind of histograms for the α coincident events. These plots are shown for completeness. We do not expect to see much MC candidate events in any bin because the α coincident events are known SC and SP ^{208}Tl events or DP events. Some MC candidate events may appear due to accidental α coincidence. Also, MC ^{208}Tl events with accidental α coincidence will also appear among these events. We can see from Fig. 6.15 that there is a lack of any significant number of MC candidate events in these bins. This further confirms that the events occurring below the reference line in Fig. 6.14 are indeed MC events.

Visual inspection of Fig. 6.14 shows that the reference line is a relatively good separator of MC and SC events. In Chap. 4, we showed how to separate MC and SC events based on the difference in the χ^2 values. We will show later in this section that this reference line produces a better separation efficiency. The first step is to calculate the perpendicular distance from the line to each event. The results is shown in Fig. 6.16 for bin#0 events and bins#5 to #30 events. Positive values indicate that the events are above the line and negative values indicate that the events are below the line. Comparing Fig. 6.16.a and Fig. 6.16.b, we see the MC events occurring as a peak around the value of -0.3. We will apply fittings to these distributions to obtain the number of MC and SC events remaining in bins#0 and #1, and also determine the most effective cut point on this new axis.

Because bin#5 to #30 events are known single-crystal DP events, we can use these events to obtain the fitting function parameters for SC events distribution. This will allows us to constrain the fitting parameters when we apply fittings to bin#0 events. The fitting to bin#5 to #30 events are shown in Fig. 6.17.a. The functions applied to the distribution are the Crystal-Ball function[43], skewed Gaussian, Landau function and exponentially modified Gaussian function. The best fitting function by χ^2 value is the skewed Gaussian(SG) function. The fitting by SG is shown in Fig. 6.17.b. Given the standard Gaussian distribution by:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{6.1}$$

and the cumulative distribution function given by:

$$\Phi(x) = \int_{-\infty}^{x} \phi(t)dt = \frac{1}{2}[1 + err(\frac{x}{\sqrt{2}})] \tag{6.2}$$

where err is the error function, the SG function is defined by:

$$f(x) = 2\phi(x)\Phi(\alpha x) \tag{6.3}$$

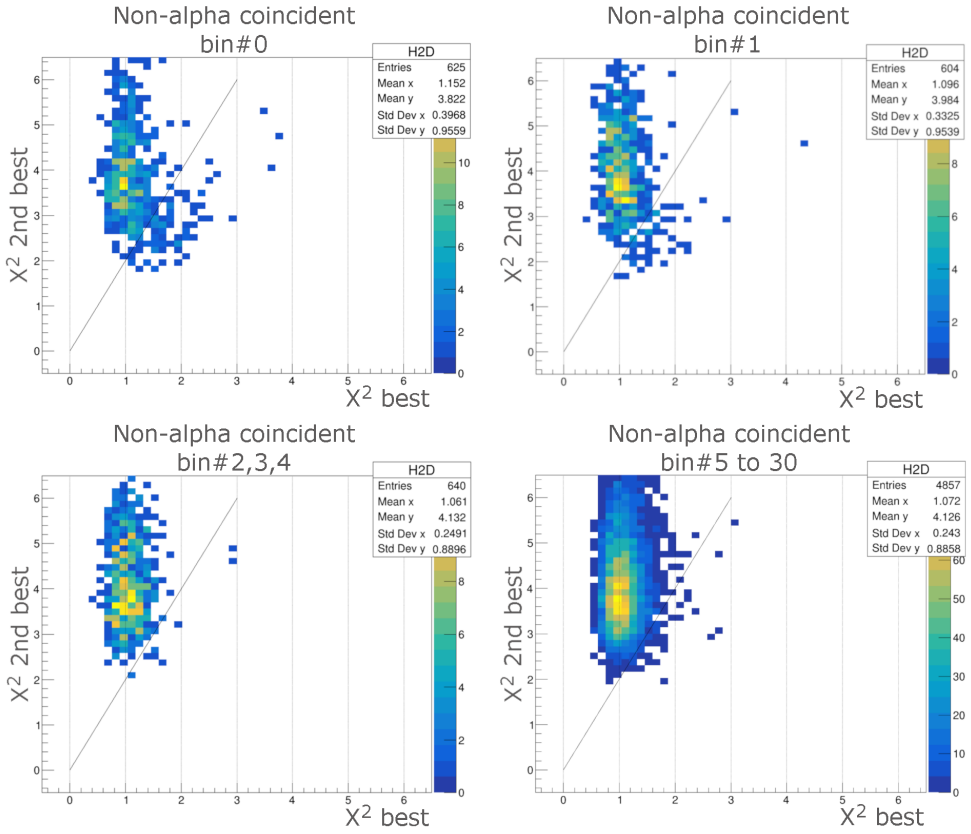


Figure 6.14: Hit pattern fitting results for non-alpha coincident events in various MIMO network output bins. A line of slope 2 is drawn as reference. We observe in bin#0 events that a significant number of events occur below the reference line. These are multi-crystal event candidates. Bin#1 also contains some number of events below the reference line. These occur due to network inaccuracy. We see few multi-crystal event candidates among events in bins#2,3,4. These events contain some single-pulse events due to network inaccuracy. For events in bins#5 to #30, we see very few multi-crystal candidates. This is because these are known double-pulse events, which are also single-crystal events. We will use these events to obtain the single-crystal event distribution parameters.

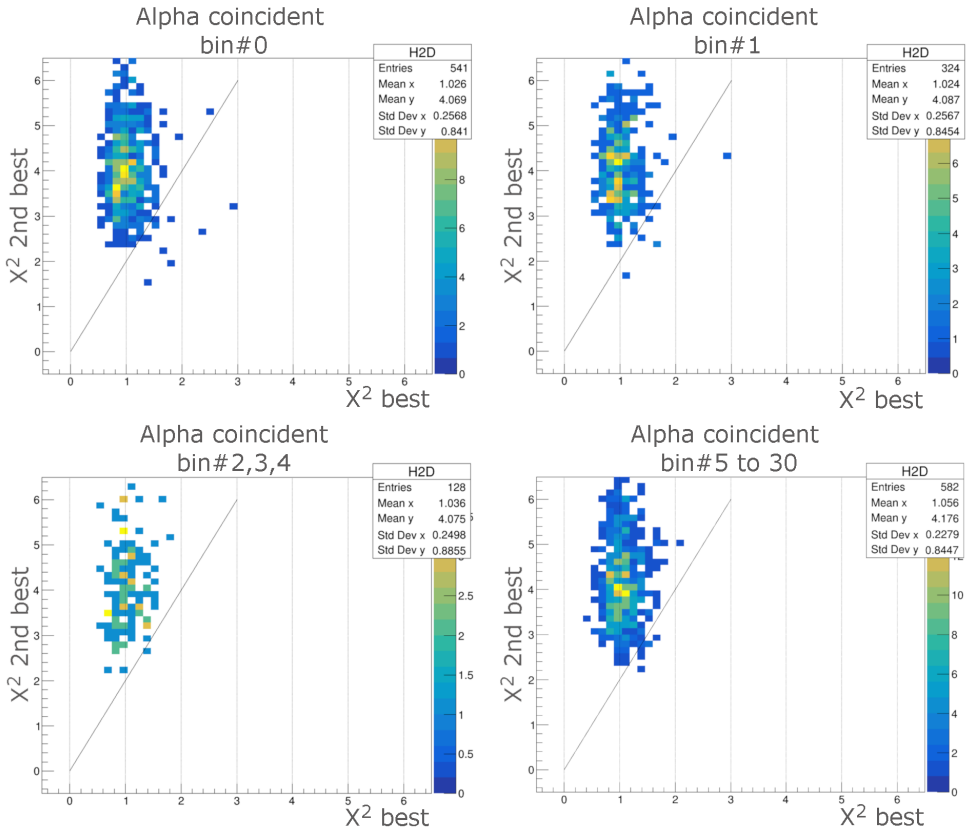


Figure 6.15: For completeness, we apply the HPs to the α coincident events. We do not see significant number of multi-crystal candidate events because the α coincident events are single-crystal ^{208}Tl events or accidental α coincident DP events. The few multi-crystal event candidates in bins#0 and #1 are likely due to accidental α coincidence or actual multi-crystal events where the preceding α was correctly determined in the same crystal.

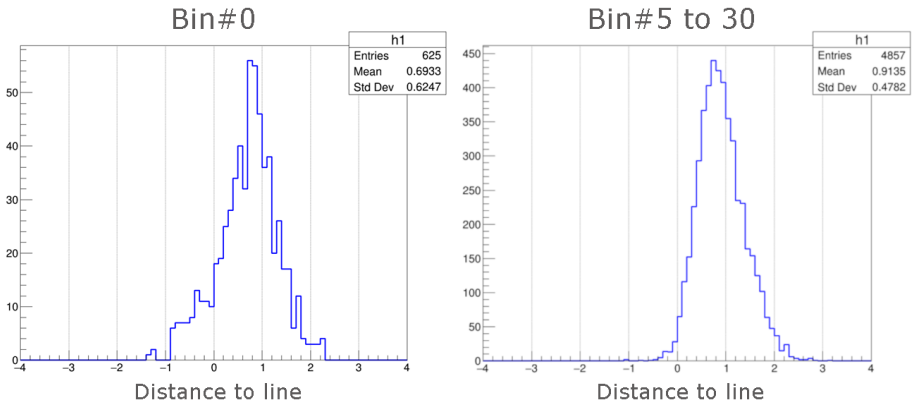


Figure 6.16: Events distances to the reference line are shown as a distribution. Values less than 0 are events below the reference line and values above 0 are above the reference line. a.) Bin#0 events distributed on the reference line. We can observe two distributions, representing the SC and MC events. b.) Bins#5 to #30 events are shown distributed on the reference line axis. Because these are known SC and DP events, we only see a single peak representing the SC events distribution.

The SG function is shown with varying α parameters in Fig. 6.18. Notice that when α is zero, the SG function reduces to the standard Gaussian distribution. To insert position and scale into the SG, we make the modification: $x \rightarrow \frac{x-\xi}{\omega}$.

To confirm the effectiveness of this reference line, we apply it to generated false MC and SC events. The false events are generated as in Sec. 4.4; SC events are generated by adding ^{88}Y events to ^{208}Tl events in the same crystal and MC events are generated by adding donor events from different crystals. We allow CANDLES analysis tools to reduce MC event count by applying crystal cut of $< 1.5\sigma$. The application of the HPs to the remaining events is shown in Fig. 6.19.a. As before, the crystal cut reduces a significant number of MC events and the surviving MC events are relatively difficult to separate from the SC events.

In Fig. 6.19.b, we calculate the distance to the reference line for the generated false events and also fit SG plus Gaussian function. Because the events are generated and pre-classified, we can completely separate the MC and SC distributions. These are shown as black and brown crossed points in Fig. 6.19.b. After the fitting is complete, the individual SG and Gaussian functions are also drawn as dashed lines. We can see that the individual functions approximate the actual distributions very well. Although not shown here, the generated SC only distribution was fitted with SG function to confirm that this function is indeed the appropriate shape for this distribution. Also, we fitted SG function to the multi-crystal only distribution but the

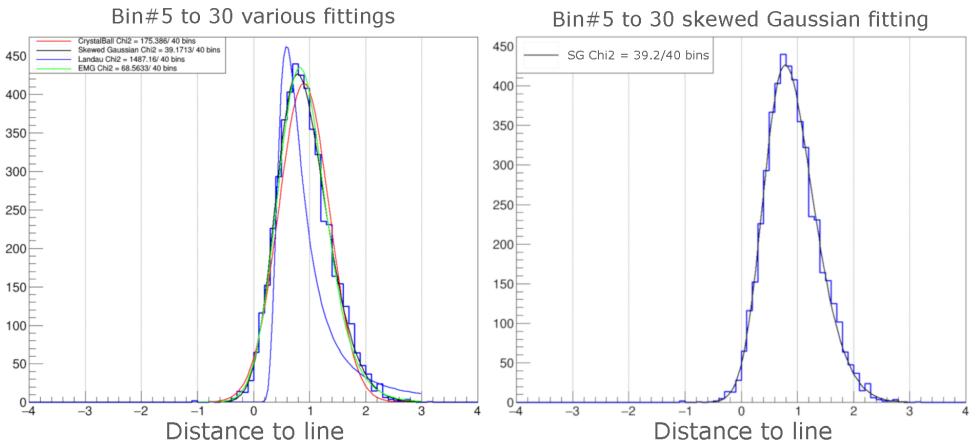


Figure 6.17: Various fittings are applied to obtain the single-crystal event distribution. Bin#5 to #30 events are known single-crystal double pulse events. a.) The fitting functions used are the CrystalBall function, skewed Gaussian, exponentially modified Gaussian and Landau function. b.) The best fitting function is the skewed Gaussian and is shown. The χ^2 of the fitting is 39.2/40bins.

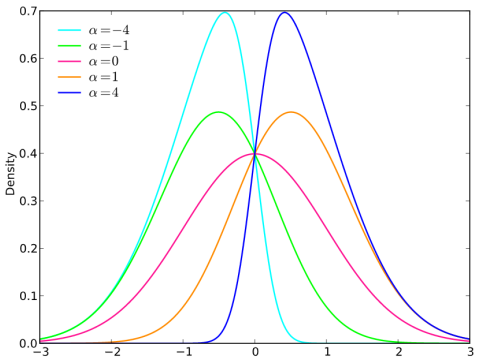


Figure 6.18: Examples of the SG function is shown with differing α parameters. Note that when α is 0, the function reduces to standard Gaussian. *image source: wikipedia commons*

fitting reverted the SG function into regular Gaussian function. Thus, SG plus Gaussian function is confirmed to be the appropriate fitting function along this new axis.

To confirm that this new axis produces better results than the $\Delta\chi^2$ based cut we discussed in Chap. 4, we also plot the multi-crystal and single-crystal distributions in Fig. 6.19.cd. These distributions are obtained from the same events that made the distributions in Fig. 6.19.b. As before, the actual MC and SC distributions are drawn in brown and black crossed points. To estimate individual distributions, we fitted SG plus Gaussian in Fig. 6.19.c and SG plus SG in 6.19.d. We observe that although the combined fitting is relatively good, the individual SG and Gaussian functions produce poor approximates of the actual distributions. Thus, the fitting produced by this new axis produces better approximations of the actual MC and SC distributions and we chose to use this axis in this study.

In chapter 4, we saw that the distribution along the χ^2 of the 2nd best fit crystal is dependent on energy. This means that the fittings in Fig. 6.19.b is also energy dependent and we can not use the parameters of the fittings on real events. This is why we estimated the fitting parameters of the SC events distribution by fitting to the events in bins#5 to #30 in Fig. 6.17.b. Given these fitting parameters, we can apply a constrained fitting to the events in bin#0 to estimate the MC and SC distributions, shown in Fig. 6.20.a. Here we constrain the parameters of the SG function representing the SC distribution to its estimated errors and only allow the normalization parameter to be freely fitted. The Gaussian function representing the MC distribution is not constrained. To also allow more freedom in the fitting, we tried SG plus SG fitting to bin#0 events but the MC distribution SG function reverted to a regular Gaussian in the fitting. Due to the few number of events available for the fitting, we also performed the same fittings to events in both bin#0 and bin#1 to increase statistics. The result of this fitting is shown in Fig. 6.20.b.

Based on the fitting results, we can estimate the MC rejection efficiencies. On top of the constrained fittings on bin#0 events and bins#0,1 events, we also applied fixed fitting and unconstrained fittings to bin#0 events. For the fixed fitting case, we completely fix the parameters of the SG function representing the SC distribution and only allow the normalization to vary. For the unconstrained fitting case, none of the parameters are constrained and we allow complete free fitting. These four fittings create differing distributions and to obtain the efficiency, we average over these four fittings and report the standard deviation as the error. The efficiency curve is shown in Fig. 6.21 and the values of some of the bins are shown in Table 6.3. In Fig. 6.21, the bold lines represent the mean values and the thin lines represent the efficiency obtained from each of the four fittings.

From Table 6.3, we can select cut point to reduce MC events. The parameter is labeled HPPara and represents the distance from the reference

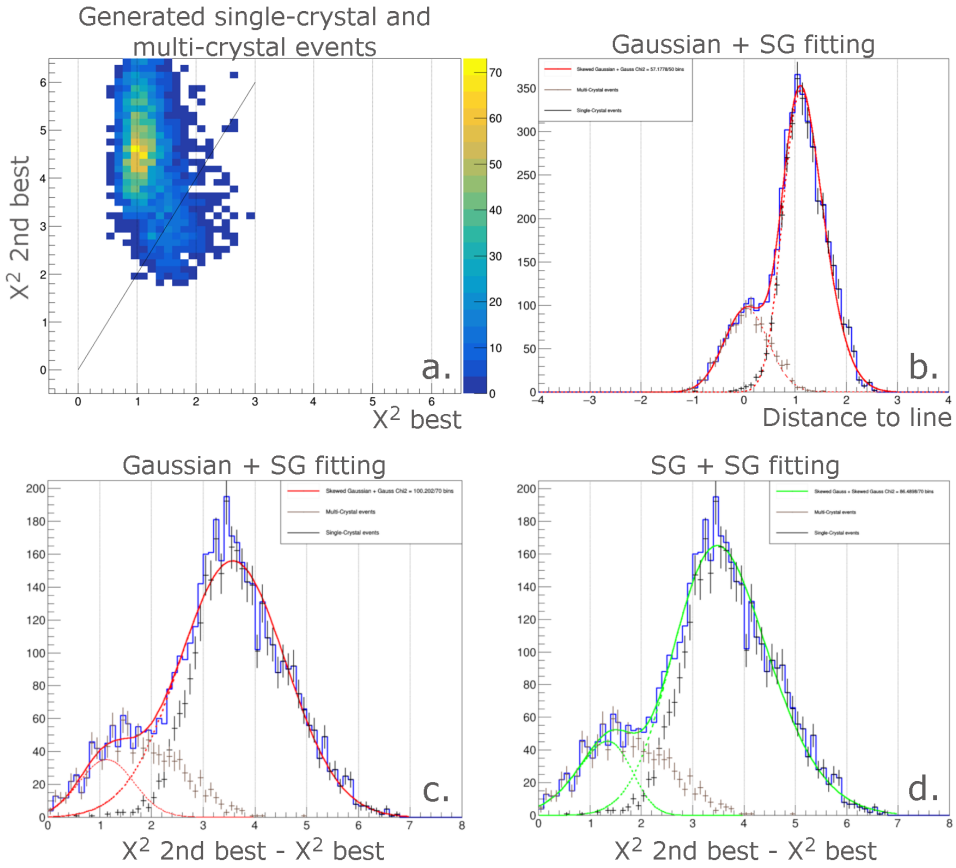


Figure 6.19: Comparison of fittings to generated multi-crystal and single-crystal events is shown. a.) The HP fitting distributions are shown for the generated single-crystal and multi-crystal events. The multi-crystal events are generated in large numbers and only the events surviving a less than 1.5 σ crystal cut are shown. b.) The distribution of the generated events are shown on the new reference line distance axis. The individual single-crystal and multi-crystal distributions are shown as dots and crosses. The distribution is fitted with Gaussian plus SG function and the individual functions model the real distributions well. c.) Fitting Gaussian plus SG on the difference in χ^2 axis is shown. d.) Fitting SG plus SG on the difference in χ^2 axis. The summed fitting function is relatively good. However, the individual functions are very poorly modeling the real distributions of the single-crystal and multi-crystal events.

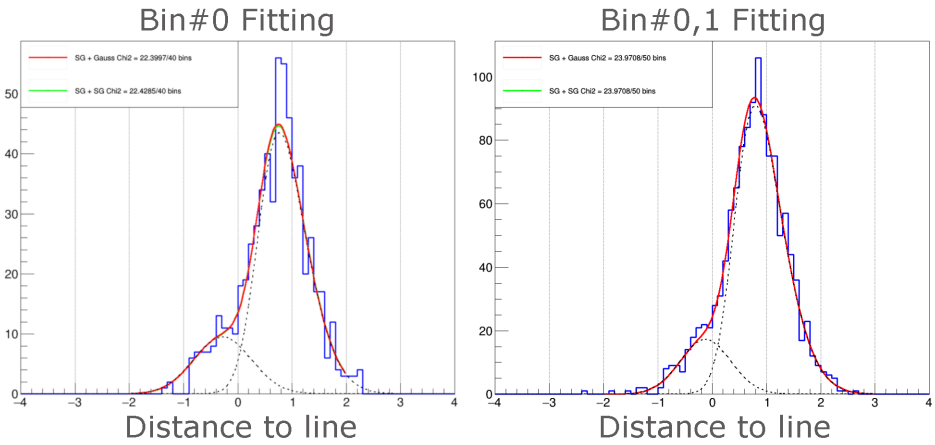


Figure 6.20: Gaussian plus SG function is fitted to observed the single-crystal and multi-crystal distributions. The parameters of the SG function representing the single-crystal distribution is constrained by the results of the previous fittings and is allowed to vary within its fitting errors. The Gaussian function is not constrained. a.) Bin#0 events are fitted with Gaussian plus SG and SG plus SG fitting functions. One of the SG function reverts to a normal Gaussian. The fitting χ^2 value is 22.4/40bins. b.) To increase statistics, bin#1 events are also included before fitting.

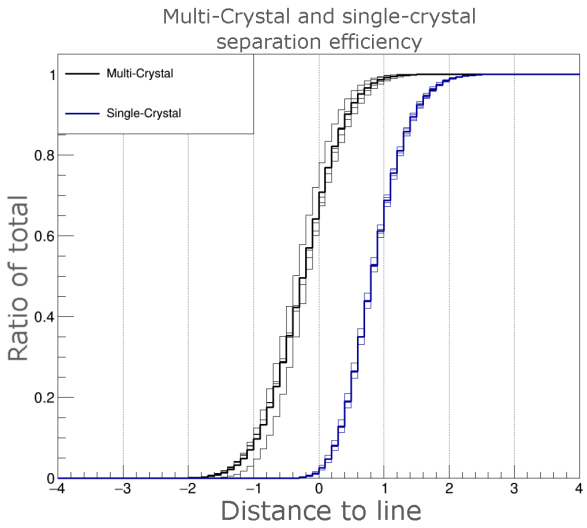


Figure 6.21: Multi-crystal event rejection efficiency curve is shown. The bold lines represent the mean curve obtained from averaging the four different fittings performed. The thin lines represent each of the fittings performed. The statistical standard deviation is obtained at each bin and reported as the error.

HPPara	-0.5	-0.4	-0.3	-0.2	-0.1	0
Multi-Crystal	0.29 ± 0.06	0.35 ± 0.06	0.42 ± 0.06	0.50 ± 0.06	0.57 ± 0.06	0.64 ± 0.05
Single-Crystal	-	-	-	-	0.01 ± 0.01	0.01 ± 0.01
HPPara	0.1	0.2	0.3	0.4	0.5	0.6
Multi-Crystal	0.71 ± 0.05	0.77 ± 0.04	0.82 ± 0.04	0.87 ± 0.03	0.90 ± 0.03	0.93 ± 0.02
Single-Crystal	0.03 ± 0.01	0.05 ± 0.01	0.08 ± 0.01	0.13 ± 0.01	0.19 ± 0.01	0.27 ± 0.02
HPPara	0.7	0.8	0.9	1.0	1.1	1.2
Multi-Crystal	0.95 ± 0.02	0.97 ± 0.02	0.98 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
Single-Crystal	0.35 ± 0.02	0.44 ± 0.02	0.53 ± 0.01	0.61 ± 0.01	0.69 ± 0.01	0.75 ± 0.01

Table 6.3: Multi-crystal and single-crystal event rejection efficiency is tabled, parameterized by the distance to the reference line(HPPara). A good multi-crystal rejection cut point is at 0.4 or 0.5, which reduces about 90% of the multi-crystal events and keeps 80% or more single-crystal events.

line to an event on the χ^2 value of the best fitted and second best fitted HP plot. For example, a cut of HPPara < 0.4 would reduce about 85% of MC events while keeping about 85% of SC events. The number of MC events estimate by the constrained fitting is (171.5 ± 20.8) events in bin#0 and #1.

6.4 Summary

The goal of this chapter is to apply the hit patterns and MIMO network to reject multi-crystal and double pulse events. In this chapter, roughly 430 days of CANDLES data was used to analyze the effectiveness of both methods. After several event selection cuts, the remaining data in the 3.6 MeV to 4.6 MeV range was 8360 events. We applied the MIMO network first and obtained new predictions for DP events with decay time less than 30ch, shown in Fig. 6.3 and compared to PSDDoubleP. The network predictions were observed to be very consistent with the expected exponential decay curve. The single pulse(SP) event selection efficiency was obtained from the distribution of ^{208}Tl events and the results were summarized in Fig. 6.11 and Table 6.1. We observed that the SP efficiency is 90% in bin#0 and #1. After subtracting the ^{208}Tl counts, we found that the DP event counts remaining in the data was consistent with the expected curve. The DP rejection efficiency is summarized in Fig. 6.13 and Table 6.2. We found that the observed DP count closely matched the expected count, thus the expected curve was used as the efficiency estimator.

In Sec. 6.3, we applied the hit patterns(HPs) to observed non-alpha coincident events in bin#0 and #1. Due to larger multi-crystal and single-crystal(SC) events ratio, the analysis focused on events in bin#0 of the MIMO predictions. A skewed Gaussian plus Gaussian fitting functions were

Rejection Method	Cut Point	Signal efficiency	Rejection efficiency	Count after rejection	Remaining background
Crystal Cut	$<2\ \sigma$	95%	-	-	-
PSDβ	$<1.5\ \sigma$	91.9%	-	-	-
PSDDoubleP	<30ch	100%	100%	8360	6621 $\pm$ 38.9
^{208}Tl cut	α coincidence	89.3%	73.1%	6739	298 $\pm$ 63.5
MIMO Network	bin#0 and #1	89.7%	93.1%	1235	410.8 $\pm$ 19.7
Multi-Crystal Cut	HPPara <0.4	87.2%	86.5%	932	27.1 $\pm$ 3.0
Q$\beta\beta$ cut	$-1\sigma + 2\sigma$	64.5%			

Table 6.4: Overall efficiency estimates are summarized.

used to obtain the MC and SC distributions on a new axis, defined as the distance to reference line and labeled "HPPara". The MC rejection efficiency was estimated using the fitted distributions and summarized in Fig. 6.21 and Table 6.3. We observed a good MC and SC separation by HPPara and this new method allows rejection of MC events, which were not possible before this study.

Finally, we estimated the overall detector efficiency after applying the two new background rejection methods. The results are summarized in Table 6.4. The first cut used is the crystal cut, which rejects most events not originating inside a crystal. The second cut is the PSD$_\beta$ cut, which rejects most events without β like pulse shape. Next, the PSDDoubleP method is used to reject DP events with decay time greater than 30 channels. The expected signal efficiency and DP rejection efficiency is 100%[42] for this cut. With these cuts, the remaining event count was 8360. From the expected decay curve, the number of remaining DP events was estimated as (6621 $\pm$ 38.9). The MIMO network and the HP methods were used on these remaining events.

The delayed coincidence method was used to select ^{208}Tl events and these events were used to estimate the network SP efficiency. Because these are also background events, these events are removed from the data. The accidental α coincidence rate is independently obtained from a separate data set and was found to be (0.107 $\pm$ 0.005), which results in signal efficiency of 89.3%. From the additional α coincidence study in App. E, the remaining ^{208}Tl events in the data is estimated to be (298 $\pm$ 63.5). This allows us to estimate the delayed coincidence method's rejection efficiency for ^{208}Tl backgrounds and the estimate is 73.1%. After this cut, the number of CANDLES events remaining is 6739.

Next, the MIMO network was applied and events in bins#2 and above were rejected as DP events. From the SP and DP efficiency estimates in Table 6.1 and Table 6.2, this cut results in 89.7% signal efficiency and 93.1% DP events rejection. Note that the DP rejection efficiency is calculated as

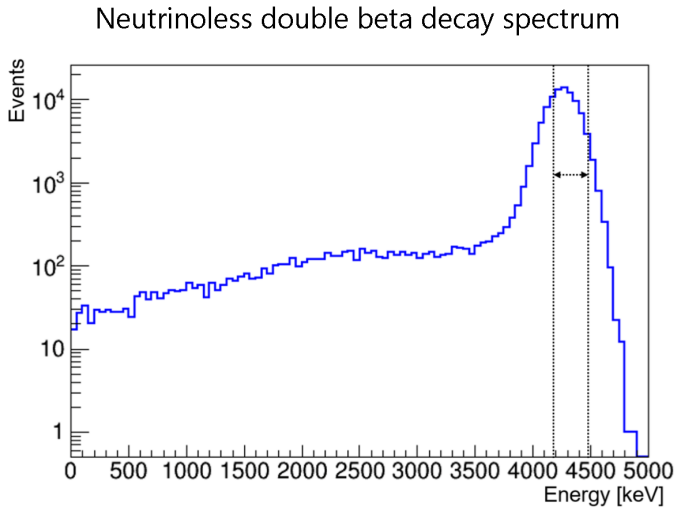


Figure 6.22: Expected energy spectrum of $0\nu\beta\beta$ obtained from MC simulation. The selected energy region is -1σ +2σ around $Q_{\beta\beta}$ value.

a ratio in 30 channels and not the full observed decay curve. From the expected decay curve, the number of remaining DP events is estimated to be (410.8 ± 19.7). Including this estimate of remaining DP events, the number of CANDLES events remaining in bins#0 and #1 is 1235 events.

The HP fitting method was developed to reduce multi-crystal(MC) events. A new parameter was defined, called the HPPara. MC rejection efficiency is defined in values of HPPara. A good cut point is HPPara less than 0.4, which result in signal efficiency of 87.2% and MC rejection efficiency of 86.5%. After this cut, 932 CANDLES data events remain with estimated (27.1 ± 3.0) MC events surviving the cut.

Although not used in this study, the $Q\beta\beta$ energy range cut of -1σ + 2σ is used in CANDLES. The efficiency is estimated from simulation, shown in Fig. 6.22. The observed tail in the energy spectrum is due to energy deposit in non scintillating part of the CaF_2 crystal housing. The estimated signal efficiency in this energy range is 64.5%. Thus the final detector efficiency is estimated to be 60.98% after all data rejection cuts and 39.39% for $0\nu\beta\beta$ signal due to $Q\beta\beta$ energy range cut.

After background rejection cuts, the initial 8360 events are reduced to 932 events. Before MC cut, (298 ± 63.5) ^{208}Tl events, (410 ± 19.7) DP events were expected in the remaining events. However, MC has a signal efficiency of 87.2%, which reduces the other background counts as well. Thus, the final background estimates are (259.9 ± 55.4) ^{208}Tl events, (358.2 ± 17.2) DP events and (27.1 ± 3.0) MC events. Around 300 events remain that are from unknown sources.

Chapter 7

Considerations

In Sec. 6.4, the number of remaining CANDLES events was 932 after various data rejection cuts. Of these, about 300 events are from unknown sources and could not be accounted for from the observed backgrounds in this study. Some possible sources are ^{208}Tl events where the parent ^{212}Bi decayed through a different α channel. ^{212}Bi has about 3% probability to decay to a different α channel, which would result in about 20 to 40 events in the remaining data. Another source of background that has not been properly studied is the $2\nu\beta\beta$ decay, which is expected to contribute a significant number of events in the 430 days of data used in this study. To estimate the number of $2\nu\beta\beta$ events in the remaining data, the energy spectrum of the decay must be known. This work is beyond the scope of this study but may explain a significant portion of the remaining data. Finally, a proper study of the cosmic backgrounds and other external background counts should be done. Because CANDLES aims for background free measurement, these additional studies are very important to account for the remaining events.

The number of remaining ^{208}Tl events in the data was estimated to be (259.9 $\pm$ 55.4). This number is estimated from additional α coincidence study in App. E and has a large uncertainty. The uncertainty rises from the low signal to noise ration. The uncertainty may be improved by using different search parameters for α coincidence, such as different energy range or different search duration. Also, tuning the initial delayed coincidence method parameters may improve the uncertainty. Finally, uncertainty may be improved by first estimating the ^{208}Tl counts before applying the ^{208}Tl cut by delayed coincidence method.

In estimating the MC rejection efficiency by HPPara cut, the systematic uncertainty was estimated from various fittings in Table 6.3. The large systematic uncertainty is due to the low signal to noise ratio of the MC events and SC events. The low number of events in bin#0 and #1 used for MC rejection analysis results in uncertain fittings. This could be improved once more CANDLES data becomes available for analysis. Therefore, the

MC rejection efficiency could be applied with more confidence in the future.

The new reference line used in the HPPara definition was simply chosen to have slope 2 and 0 intercept. However, a better reference line could be found that could separate the MC and SC events with more confidence. One way to search for a better reference line is to try an iterative process in which the slope of the line and intercept is continuously change and at each iteration, the peak to peak separation of the MC and SC distributions could be checked. The new reference line could then be define as the line that maximizes the peak to peak separation of the distributions. Since the number of events is still too few to reliably perform this check, it was not done in this study.

In this study, the DP cut point was selected to be bin#2 and above. This was chosen because it provides a good DP rejection and SP selection efficiency. But also the distribution of DP events in bins#0 and #1 were not confirmed. This means a cut point at bin#1 can not be used because we do not know the number of DP events contained in this bin. From SI analysis in App. D, it was shown that almost all 0 decay time DP events were in bin#1. Although most of these events were expected in bin#1, a small number these events were also expected in bin#0, as shown in App. C. SI values are independent and external information that could be used with more confidence than the prediction checks in App. C, but due to the low number of events, SI distribution fittings in bins#0 and #1 resulted in poor fittings. As more CANDLES data becomes available, the number of events in these bins would increase and more reliable fittings could be done. Thus, it may be possible to obtain better estimates of DP and SP events in these bins in the future.

Additionally, the unknown distribution of DP events in bin#0 and #1 was caused due to programming error, where 0 decay time DP events were not specifically generated for network training. We may observe then that generating these events for network training may result in expected number of DP events in both bins. This work was not done in this study because networks must be trained for each crystal to obtain enough CANDLES data predictions to perform statistical analysis. Training all networks take significant effort in terms of generating data, transferring data and training the networks. If deemed necessary, this work could be done in the future.

The network size is currently fixed and no modification to network design is planned. This is due again to time requirement in training network models. However, the network size and computation time is not a significant issue because making prediction using a model does not take much time. It is only in the network training that requires time. Thus, further improvements to the network is planned through continued network training. The donor ^{208}Tl and $^{214/215}$Po events for generating training DP events is continuously collected through CANDLES data runs. However, the ^{88}Y calibration events are only collected during calibration runs. Thus,

additional training DP events are generated only after new calibration runs are performed. Additional training for the network is scheduled once new calibration data becomes available.

Currently there are two major limitations of the MIMO network. The network predictions are restricted to events in energy range of 3.6 MeV and 4.6 MeV. This limitation is due to the mono energy training DP events, which are generated from mono energy donor events. Currently, there are no other viable donor events that can be used to generate DP events at other energies. Extending the energy range of the MIMO network to 5 MeV or more would allow us to see the end point of the ^{208}Tl or DP energy spectrums. Thus, additional information would become available to check the network predictions and also help estimate remaining backgrounds. One possible way to increase the energy range of the network is to artificially modify the energy of the donor events when generating training data. The pulse shape of the donor events could be scaled by some value but the PMT data information can not be scaled. This is because the PMT data is a photon count data. The number of observed photons must be scaled. Increasing the photon count of an observed donor event is difficult in practice. One possible solution is to scale the pulse shape of the α donor event and instead of using the α event's PMT data, replace it with a higher energy ^{208}Tl event's PMT data. This would mean that there will be no α event information in the PMT data but would result in generated DP event at a higher energy. The effect of generating events in this manner on the network accuracy is unknown. But it may lead to a larger energy range for network predictions.

Another issue with the network is that the generated DP events are not very clean. Because the start channel of each event is not always correctly determined, when we combine donor events to make DP events, the actual decay time of the training DP event is occasionally different from what we tell the network to classify it as. This can be observed in the network accuracy. In top1 accuracy, we observed that the network results in about 50% to 60% accuracy while the top3 accuracy is consistently above 95%. It may be possible to improve the top1 accuracy, and thus the network predictions on real CANDLES data, if we could determine the start channels of the donor events with more confidence. The current algorithm for determining the start channel of an event is a simple threshold on the summed pulse of an event. Since the summed pulse start channel information suffers from various uncertainties, it may be possible to change start channel determination to only a few high photon count PMT data. With more accurately generated DP events, the network accuracy may be improved.

Chapter 8

Conclusion

The main backgrounds of CANDLES comes from the decay of ^{212}Bi, which are due to contamination of the CaF_2 crystals. It has two decay channels, a β decay to ^{212}Po and an α decay to ^{208}Tl. Due to the short half-life of ^{212}Po, double pulse(DP) events are observed in data. The MIMO network was designed and trained to predict the decay time of DP events under 30 channels. The ^{208}Tl decays to stable ^{208}Pb, which has a significant probability to generate multi-crystal(MC) events. The hit-pattern(HP) fitting method was introduced to detect and reject these MC events.

The detection of MC events is based on fitting HPs. The HPs are the average signal received at each PMT from each crystal. A new parameter was defined, called HPPara, and MC rejection efficiency was estimated in terms of the HPPara values. The efficiency estimates are summarized in Table 6.3. We observed good separation of MC and single-crystal events by using the HPs. In this study, HPPara cut of less than 0.4 was used and the estimated signal efficiency was 87.2% and the estimated MC rejection efficiency was 86.5%.

The detection and rejection of DP events is based on a convolutional neural network(CNN) approach, called the multi-in multi-out(MIMO) network. To improve the results of previous approaches for DP detection, raw PMT signal information was used. Due to the difficulty of analysis of raw PMT signals, the CNN based approach was suggested and tested. A significant improvement over previous method was observed. The network's DP rejection efficiency and SP selection efficiency was estimated using various external information. It was found that the network prediction is consistent with the expected number of DP events down to 2 channel decay time. Thus, a DP rejection cut point of 2 channel and above was suggested with an expected DP rejection efficiency of 93.1% and an estimated SP selection efficiency of 89.7%.

Using the new background rejection methods, the number of remaining backgrounds was estimated. About 300 events remain in 430 days of CAN-

DLES data that could not be attributed to the backgrounds considered in this study. Additional background studies are recommended for $0\nu\beta\beta$ analysis.

In recent years, neural network based machine learning has gained applications in various sciences. During the analysis of this study, we were not aware of any similar application of machine learning for physics analysis. Thus, in this study, we applied CNN to see if the method could be adapted for physics data analysis. The main issue with neural network based methods is that a large number of training data must be available. Moreover, the neural network methods do not allow a natural estimates of its uncertainties. In this study, external information was used to estimate the network errors. However, despite its shortcomings, we found that the network can provide significant improvements to physics analysis. In the case of CANDLES, due to the difficulty of analyzing raw PMT data, we apply DP detection method on summed pulse shape, which is generated by losing information. We found that by using CNN method on the raw data, some improvement could be made in prediction accuracy. Thus, under certain conditions, neural networks were shown to be a viable analysis tool for physics analysis.

Appendix A

Introduction To Neural Networks

A.1 Neural Network as a Function Approximator

In this study, we developed a convolutional neural network model to reduce ^{212}Bi $\rightarrow$ ^{212}Po $\rightarrow$ ^{208}Pb decay backgrounds. Since physicists are not familiar with the neural network approach to data analysis, we present here an introduction to neural networks and convolutional neural networks. The contents are written for physicists and the depth is enough to understand the model used for this study. However, for a more in-depth introduction to neural networks, we suggest [23] or [24]. The contents of this chapter do not explore general machine learning topics, which are usually statistics based algorithms. For a more general introduction to machine learning, [25] and [26] are good introductory books.

A.1.1 Data and Data Analysis

In data analysis, we are often interested in classifying data based on what was observed or recorded by detectors. For example, we may want to classify if an event is α or β event etc. However, raw data usually comes in forms that we can not directly use for classification. CANDLES experiment uses 62 PMTs to collect event signals and the raw PMT readouts can not be used directly to classify the data. Multiple steps of information extraction from raw data is needed until the data can be used. In machine learning terms, this is called *feature extraction* and the features could be things such as energy, position or any other parameter of interest. We can think of feature extraction as a transform from one data space to another: $\mathbf{Y} = f(\mathbf{X})$, where $\mathbf{X}$ is the raw data and $\mathbf{Y}$ are the extracted features. Often multiple transformations can be chained together to obtain a final feature. For example, to convert raw PMTs data to energy, we must first subtract the

PMT baseline, then integrate the signal over the duration of the event and finally multiply by a conversion factor. This could be written as $Energy = f(g(h(\mathbf{X})))$, a three stage mapping. Mapping in this context means taking the initial data and mapping it into another feature space. Therefore, we will use mapping and feature extraction interchangeably in this study.

Suppose we find ourselves in a situation where we are unable to determine the kind of mapping that should be applied to data to obtain good features. For a long time, this was the situation that computer vision scientist found themselves in. Given an image, what kind of mapping and feature extraction steps are necessary to say that the image is that of a dog or a cat. One approach that is guaranteed to work but is impossible in practice is to transform the data with every possible mapping function. These mapping functions would be able to learn the correlation between the data points and thus can extract the absolute best features for classification, even if we do not understand what those features represent. Each of the feature extraction steps we performed as part of CANDLES data analysis tools in chapter 3 can be replicated with this search.

However, we face the problem of searching for a good mapping function in an infinite space of functions. In practice, this would require infinite time. To complete this search partially, a disciplined search algorithm was proposed, called the back-propagation algorithm, to search for good mapping functions. These mapping functions are "good" in the sense that it provides good classification of data but it is not guaranteed to be the "best" mapping function. Finding the best would require trying over every possible function. This cannot be done and only a subset is searched in practice.

Simply, a neural network is an algorithm that is able to classify data by starting with a random set of mapping functions and, by using the back-propagation algorithm, transforms them into a good set of mappings that allows for accurate data classifications. These mapping functions are automatically selecting the most relevant features from the original data that result in the most accurate classification. There are multiple drawbacks and also advantages to using the neural network method. The main reason that neural networks have become so popular today is that it does not require expert knowledge of the data to be effective. Traditionally, the functions used to extract features are designed and applied by experts in the field, e.g. how to extract energy from PMT readouts. However, with neural networks, anyone can now simply apply the algorithm and obtain a data classifier. This is acceptable to some fields, such as image processing and language processing, where the type of features extracted are of no importance. One of the purposes of this study was to determine whether the neural network algorithm could be used effectively in physics data analysis.

The other reason, and also the reason why neural network was used in this study, is that it allows feature extraction directly from raw data. In the case of CANDLES, analysis of each raw PMT signal is difficult and designing

and applying transformation functions separately for each PMT signal is an impossibly difficult task. With the application of neural networks, we were able to extract features directly from raw data that allow for more accurate data classifications. The details of neural network applied to CANDLES data will be presented in the next chapter.

There are two main drawbacks to using neural networks. The first being that we do not know what kind of features are being extracted from the data. This means that we are also unable to determine the uncertainties and error of each extracted feature. As a result, uncertainty on the final classification output is also unknown. There have been many studies that tried to determine what kind of features were extracted from data but these have mostly been in the field of image processing, as in [27] or [28]. In this study, the uncertainty on the classifications were estimated using external information. The other major drawback is that a large number of pre-classified data must be available during the search for mapping functions. With a small set of data, the network is able to completely over fit the data and provide false accuracy. There are several methods that try to reduce over fitting and they will be mentioned in relevant sections.

A.1.2 Universal Approximation Theorem

As part of a search for good set of mapping functions that extract relevant information from data, we would need to try many different functions to transform the data. One way to do this systematically is to start with an orthonormal set of functions with random initial variables and change these variables until a good mapping function is found. Neural networks adopt this approach but uses a different set of expansion functions. These expansion functions originate from the universal approximation(UAT) theorem. The statement of the theorem is given in Thm. 1 and the proof can be viewed in [29] or [30]. The UAT states that we can approximate any continuous function by summing over a sufficient number of terms in the expansion. Thus, neural networks attempt to search for good mapping functions by finding the expansion coefficients in Eqn. A.1. In principle, if a feature extraction function exists that can extract the best features for a classification, the neural network will be able to find an approximate of this function.

Theorem 1 (Universal Approximation Theorem) *Let $\sigma()$ be a monotonically increasing, nonconstant, and continuous function. Let I_n denote the n-dimensional unit cube, $[0,1]^n$, and $C(I_n)$ denote the space of continuous functions of I_n. Then, there exists an integer N, real numbers $v_i, b_i \in \mathbb{R}$ and real vectors $\mathbf{w}_i \in \mathbb{R}^m$ such that for any $\epsilon > 0$ and $F \in C(I_n)$, a function $f(\mathbf{x})$ defined as:*

$$f(\mathbf{x}) = \sum_{i}^{N} v_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i) \tag{A.1}$$

can be found to have measure of:

$$|F(\mathbf{x}) - f(\mathbf{x})| < \epsilon \tag{A.2}$$

for all $\mathbf{x} \in I_n$. *That is to say,* $f(\mathbf{x})$ *are dense in* $C(I_n)$.

Suppose $F(\mathbf{x})$ is the "good" feature extraction function and $f(\mathbf{x})$ is its approximate. The UAT states that we can write this approximate function as Eqn. A.1. Then, the job of the neural network algorithm is to determine the correct values of each of the constants of the expansions: $\mathbf{w}_i$, v_i and b_i. The function $\sigma()$ is called the **activation function** and it does not have very strict restrictions. Therefore, a wide variety of functions were tried by computer scientists and the popular few will be shown in a later section.

A.1.3 The Basic Neural Network Model

Neural network discussions often comes with a node graph showing the construction of the model. Figure A.1 shows a basic neural network model with 4 "hidden" layers, 1 input layer and 1 output layer. The input layer contains $n + 1$ nodes, with n being the number of dimensions of the input data. For example, CANDLES uses 62PMTs and so $n = 62$ if we wish to use the PMT data in a network. An extra node is also included in the input layer, although this node is often omitted in neural network model drawings. This node has value of 1 and act as a bias term, which will become apparent soon.

Every node in the input layer connects to every node in the next layer, as shown by the lines connecting the nodes. For example, node h_{11} in the first hidden layer is connected to every node in the input layer. When the node sees an input from the previous layer, it is "activated" by the activation function. That is, the value of node h_{11} is calculated as $h_{11} = \sigma(\mathbf{w}_{11} \cdot \mathbf{x} + b_{11})$, where $\mathbf{x}$ is the input data and $\mathbf{w}_{11}$ and b_{11} are the constants appearing in a single term in the summation of Eqn. A.1. The $\mathbf{w}_{11}$ constants are called weights and every input node from the previous layer is multiplied by a scalar weight. They are represented by each of the lines appearing between nodes. The weights are not shared between nodes and therefore $\mathbf{w}_{11}$ and $\mathbf{w}_{12}$ are not the same between h_{11} and h_{21} nodes. The extra node with value of 1 is multiplied by a weight and becomes the "bias" term b_i in Eqn. A.1. Some authors consider this as obvious and choose not to draw them in the graph.

We then see that each node in the first hidden layer is calculating a single term in the summation of equation A.1. The normalization terms

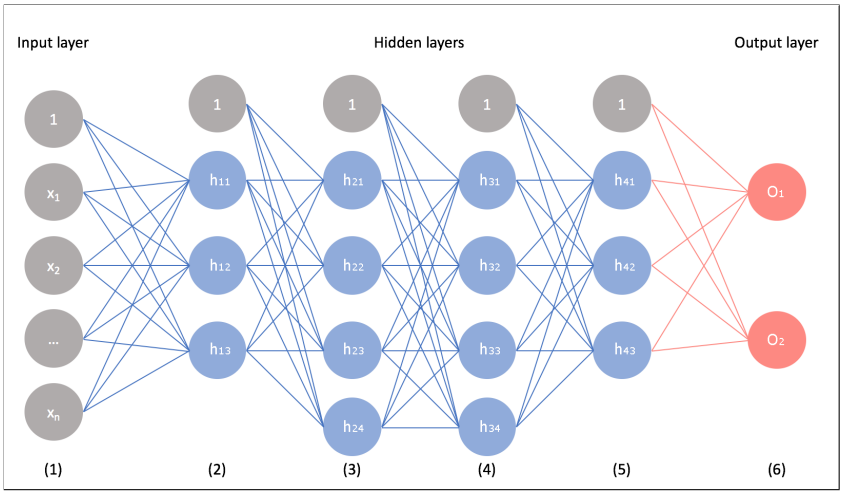


Figure A.1: A basic neural network model is shown with 1 input layer, 4 "hidden" layers and 1 output layer. The input to the network is an n dimensional vector with values $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2 \ldots \mathrm{x}_n)$. Every entry in the input data is represented by a single node in the input layer. The input layer adds an extra dimension with value equal to 1. Each node in the input layer connects to every node in the next layer, with the connections drawn by the blue lines. To calculate the value of node h_{11}, we first multiply the values of each of the input layer nodes by a set of scalar weights: $\mathbf{w}_{11} \cdot \mathbf{x} + b_{11}$, where b_{11} is the extra node added to the input layer multiplied by a weight. Next, the value of node h_{11} is calculated by evaluating the activation function: $h_{11} = \sigma(\mathbf{w}_{11} \cdot \mathbf{x} + b_{11})$. Thus, the output of node h_{11} is a scalar calculated from the input data. This "activation" operation is performed for every node in the layer and the weights connecting the nodes are different for each node. Once all of the activations are calculated for the first hidden layer, the process repeats into the next layer. This is true for the output layer as well, although the activation function of the output layer are usually different from the other nodes. More on this will come later. We see that the neural network is passing along information from the input layer to the output layer. Each layer in the hidden layers is representing a feature extraction function approximated as in Eqn. A.1. Thus, we see that the neural network is performing a set of feature extraction steps on the original data.

v_i are not included in this calculation because they are absorbed into the weights connecting to the next layer. Therefore, the first hidden layer of the model shown in Fig. A.1 is approximating a feature extraction function $F(\mathbf{x})$ using three terms. The nodes h_{11}, h_{12} and h_{13} are simply numbers after calculations and therefore, we see that the network has compressed the n dimensional input data into 3 features.

The connection between input data and overall feature extraction function $F(\mathbf{x})$ may not be apparent, especially with only three nodes in the hidden layer. Usually, hidden layers contain tens to hundreds of nodes, allowing for an approximation of more complex functions. However, an important feature of neural networks is that it can use subsets of nodes to approximate multiple separate feature extraction functions in a single layer. Consider the task of extracting the energy of an event in CANDLES and the input is the total signal received at each PMT. In this case, the input is a 62 dimensional data, representing each PMT. How many nodes are required to obtain the energy of an event from this input? This can be done with a single node. To obtain the energy, we must sum the input data to obtain the overall signal, subtract the baseline and finally multiply by a conversion factor. From equation A.1, it is apparent that a single node can accomplish this. Apparently, a single node can accomplish quite a bit of work by itself. Given tens of nodes in a single layer, several nodes can work together to extract more complex features and in each layer, multiple different features can be extracted. As a result, we often think of a layer as extracting multiple different features as well as compressing the original data into a smaller set of features. The downside is that we do not know what kind of features are being extracted by the nodes, as multiple nodes could be working in conjunction and their exact connection is unknown. This is the reason why these layers are called "hidden" layers. How the input data is transformed is not known at any of the hidden layers.

Once the values of h_{11}, h_{12} and h_{13} are calculated, they are input into the next layer and go through the same process in the second hidden layer. The second hidden layer may extract more features based on the extracted features of the first hidden layer or make more correlated features by looking at once at all of the extracted features of the first hidden layer. Since the first hidden layer compressed the input data into 3 dimensions, it does not make intuitive sense to use more than three nodes in the second hidden layer. In fact, neural network designs usually reduce in number of nodes as the number of hidden layers increase. The intuitive understanding is that at each step of the hidden layer, information is compressed and only the important features are passed to the next layer. However, there are no guiding principles in designing neural networks and users try many different variations in number of layers and number of nodes. If the network is too large, the computation time increases exponentially and the risk of over fitting increase. If the network is too small, compression of data loses too

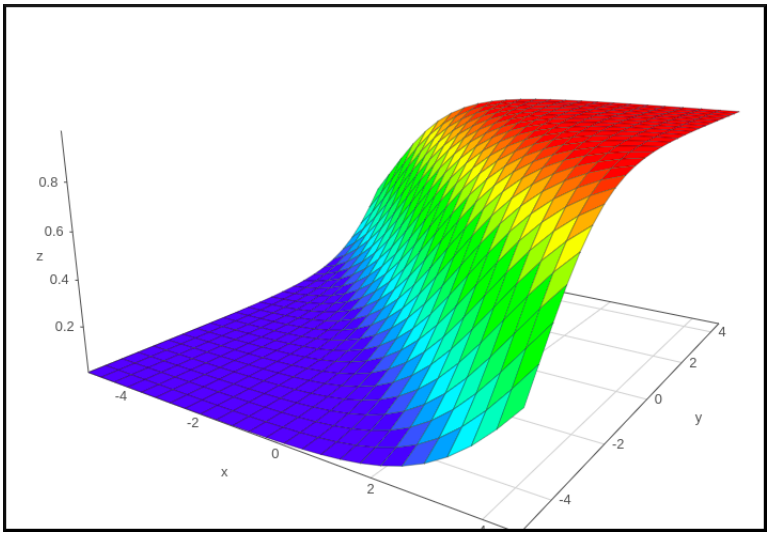


Figure A.2: Logistic(or sigmoid) function drawn for the two dimensional case. It has outputs ranging from 0 to 1, making it ideal for binary classification applications. It is used instead of the step function because of differentiability.

much information and the accuracy of the network drops. As a result, multiple different network sizes are tried until a good sized network is found for a given application.

The output nodes of the network are calculated in the same way but they are treated differently from the hidden layers. If the output layer contains only a single node, then this is a binary classification network. In this type of network, we want to classify the input data into one of two possible classes. For example, on or off, 0 or 1 etc. The output of the node is still calculated as $O = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$, where $\mathbf{x}$ is now the information flowing in from the previous layer and $\sigma()$ is an undecided activation function. We can imagine then that the classification depends on the value of $\mathbf{w} \cdot \mathbf{x} + b$ being large or small. For example, we may use a step function and classify the data into one class based on the value of $\mathbf{w} \cdot \mathbf{x}$ and the threshold being controlled by b. A popular binary classification function used as the activation function is the *sigmoid* or *logistic* function, given by:

$$\sigma(\mathbf{w} \cdot \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w} \cdot \mathbf{x} + b)}} \tag{A.3}$$

It is a smoothed step function and can be thought of as giving the probability of being in one of the classes. Example is shown in Fig. A.2. This is one of the activation functions used in this study.

In summary, a neural network is a function approximator and the functions being approximated are the feature extraction functions. The number

of hidden layers of a network represent the number of feature extraction steps and can be written as function compositions, e.g. $F(G(H(\mathbf{x})))$, and each of the functions in the composition may be multi-dimensional and could be extracting very different features. The number of nodes in each layer represent the number of expansion terms in Eqn. A.1, although the network may choose to use a small subset to approximate various different mapping functions. Depending on the size and design of the network, information is being transformed from input data and compressed through the network until features that are very important for classification are passed onto the final output layer. The output layer applies a classification function to put the input data into one of several classes. The "training" stage of the network is simply finding the best constants of the expansions of each node, $\mathbf{w}_i$ and b_i, that results in the best classification accuracy. The rest of the chapter will provide detail on how to find these constants.

A.2 Determining the Constants

A.2.1 The Loss Function

The first step in finding a good set of constants is to estimate how inaccurate our current network is at making classifications. This is achieved by defining a **loss function**. It is used to estimate the error as well as guide the training of the network. Suppose the ith training data is written as $\mathbf{x}^i$ and its corresponding class is labeled as y^i. Classes are labeled numerically; if there are only two classes, then $y^i = 0$ or 1. Then, a simple loss function, also known as *mean squared error* (MSE), can be written as:

$$J(\mathbf{W}, \mathbf{b}, \mathbf{x}) = \frac{1}{2N} \sum_{i}^{N} (h(\mathbf{x}^i) - y^i)^2 \tag{A.4}$$

Here, $h(\mathbf{x}^i)$ is the final output of the network and its value is the predicted class number. For simplicity, let us assume there are only two classes, with $y^i = 0$ or 1. The output node of the network outputs $h(\mathbf{x})^i$, and it has values between 0 and 1. If the network is completely accurate, then the network prediction and the class label are equal. That is, $h(\mathbf{x}^i) = y^i$ and so the cost function is equal to zero after summing over all available data. In practice, the loss function is not equal to zero and the purpose of training the network is to minimize this loss function. Depending on the type of classification function used on the output layer, the network can output a range of values between 0 or 1, or integer values of 0 or 1 strictly. Since the activation functions, $\sigma()$, of each node is fixed, the only variables that change the output of the network are the constant weights and biases between all the nodes of the network. They are marked as $\mathbf{W}$ and $\mathbf{b}$ collectively in Eqn. A.4. We can then frame the problem as that of minimizing the loss function

with respect to $\mathbf{W}$ and $\mathbf{b}$. Since the network can have up to millions of constants, an exact solution to the minimization problem is not attempted and an iterative solution is used. We will see how this works in the next several sections.

The issue with using the MSE cost function in practice is that the derivative of Eqn. A.4 is dependent on the value of the network output, $h(\mathbf{x}^i)$. If $h(\mathbf{x}^i)$ and y^i are very close, then $(h(\mathbf{x}^i) - y^i) \approxeq 0$ and results in a vanishing derivative. This is problematic because to minimize the cost function, we must take the derivative of the cost function and a vanishing derivative slows the network learning progress. A better loss function used in practice is called the *cross-entropy loss* and it is borrowed from information theory. For a two class classification network, the cross-entropy loss can be written as:

$$J(\mathbf{W}, \mathbf{b}, \mathbf{x}) = \sum_i -[y^i \log h(\mathbf{x}^i) + (1 - y^i) \log (1 - h(\mathbf{x}^i))] \tag{A.5}$$

As before, y^i is the class label of the ith data sample and $h(\mathbf{x}^i)$ is the network prediction for the ith input data sample. Since there are only two classes y^i can be either 0 or 1. Suppose that for a given input data, $y^i = 1$. Then, the second term in the sum of Eqn. A.5 drops out and the remaining contribution to the sum reduces to $-\log h(\mathbf{x}^i)$ for the ith sample. If $y^i = 0$, then the first term in the sum drops out and the remaining term reduces to a single logarithmic function. Comparisons of the cross-entropy and the MSE loss functions are shown for the case of a single sample with $y^i = 1$ in Fig. A.3a.

We can see that when the network outputs the correct class label, the values of the loss functions are both zero. However, if the network output is very wrong, then the cross-entropy loss function penalizes the network with a much larger value. Figure A.3b. shows the inverted first derivatives of the loss functions. The MSE derivative becomes smaller as the network output approaches the correct value and therefore, the iterative training can never reach the correct output value. This is an undesirable trait. On the other hand, the cross-entropy loss function has a derivative of 1 when the network output is equal to the correct class. Furthermore, when the network output is very wrong, then the derivative is very large for the cross-entropy loss, which forces the network to make very large steps to correct the error. Therefore, the cross-entropy loss is a much better loss function during network training.

In the case of more than 2 possible classes, the cross-entropy loss can be naturally extended to:

$$J(\mathbf{W}, \mathbf{b}, \mathbf{x}) = \sum_i \sum_{c=1}^{M} -y_c^i \log h_c(\mathbf{x}^i) \tag{A.6}$$

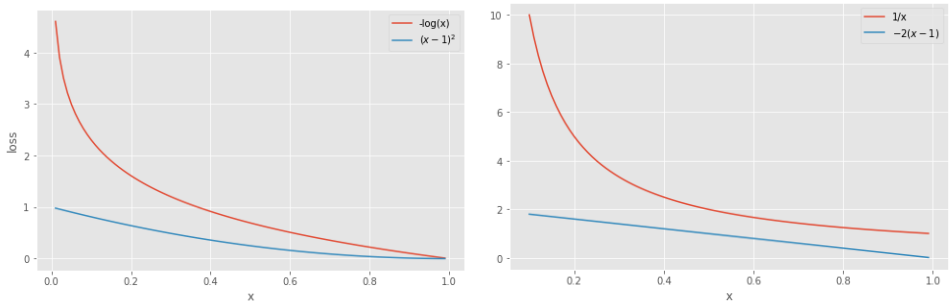


Figure A.3: a. MSE and cross-entropy loss curves shown. Assume that the true label of a sample is equal to 1 and x is the output of the network. We see here that if the network output is very wrong from the true sample label, then the cross-entropy assigns a much higher loss penalty. b. The derivatives of the MSE and cross-entropy are shown. If the network output and true label are very close, then the derivative vanishes for the MSE. The cross entropy derivative never vanishes. Moreover, if the network output is very wrong, the derivative is very large for the cross-entropy. This means the network can take very large step to correct the error because the step correction is related to the derivative of the loss function. More on this will be shown later.

Here, M is the number of possible classes and c is the index of one of these classes. y_c^i is the label of the ith input data for class c. For example, if there are 5 possible classes and the ith data belong to class 1, then $y_c^i = 1$ for $c = 1$ and $y_c^i = 0$ for $c \neq 1$. The network then outputs a class value for each possible class, given by $h_c(\mathbf{x}^i)$. In this study, both the binary and the multi-class cross-entropy losses were used. In general, any differentiable function that can provide some form of accuracy metric can be used as a loss function. It is common to see various custom loss functions for very specific purposes in advanced neural network applications. For a comparison of some of the more popular loss functions, please see [31].

A.2.2 Activation Functions

In the previous section, we saw that the universal approximation theorem allows a wide variety of functions to be used as the activation function. In this section, we will look at the different activation functions that were proposed and used in the past. Mishkin et al. published a study comparing the effectiveness of different activation functions that are popular in the context of image classification [32].

Although a variety of functions are allowed, training the network through loss function minimization means the functions should be differentiable. Also, functions with easily calculable derivatives are preferred. Examples

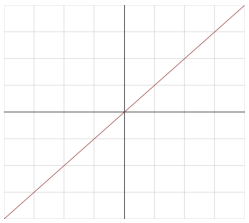
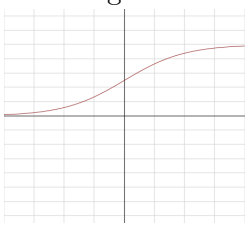
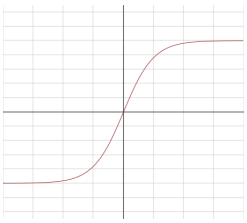
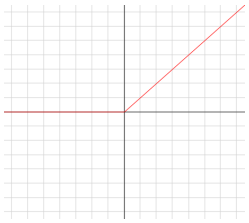
Linear	Logistic	TanH	ReLU
			
$f(x) = x$	$f(x) = \frac{1}{(1+e^{-x})}$	$f(x) = tanh(x)$	$f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$
$f'(x) = 1$	$f'(x) = f(x)(1 - f(x))$	$f'(x) = 1 - f(x)^2$	$f'(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$

Table A.1: Sample activation functions used by neural networks. All of the functions are piecewise increasing and continuous, as required by the universal approximation theorem.

of some of these functions are presented in Table A.1. In general, there are no methods that can be used to determine the best activation function for a given neural network application. The paper by Mishkin et al.[32] applied these functions to image classification problem and compared the results. Although there are four functions shown here, the paper tried several more variants of the ReLU activation function.

Historically, the logistic function, Eqn. A.3 was the preferred activation function. It first appeared in context of neural networks in the 1980s. However, the logistic function also suffers from the vanishing derivative problem, which causes the network training to slow down. As a result, the logistic and tanh activation functions were abandoned. Empirically, the ReLU and its variants were shown to perform better than the logistic or similar functions, as in [32]. However, the logistic function is still used widely in the output layer of neural networks for classification problems because of its $(0, 1)$ range.

Mishkin's results showed that the linear activation function performed the worst when compared to any of the other activation functions. The best performing class of activation functions are the ReLU(Rectified Linear Unit) and its variants. These ReLU variant functions are nice to work with because taking derivatives is not necessary, since the derivatives are known and constant. The paper recommends the ELU activation function, which is a variant of the ReLU with a piecewise exponential function for $x < 0$. If batch normalization is used, then the recommended activation function is the ReLU. Batch normalization is a simple operation that normalizes the output of each node of a layer by the total of that layer. This keeps the output of each node close to zero and because network weights and biases are

usually randomly generated from a gaussian centered at zero with deviation of one, this works to keep the scale of the weights and the information from layers on the same level.

In general, the current recommended and popular activation function is the ReLU. This recommendation is based on empirical evidence and the exact reason why it performs better than others is not known. There have been more work done on generalizing the ReLU, for example the maxout networks [33], but the choice of activation functions have not been standardized and is left entirely up to the programmer. In this study, the ReLU was used on all nodes except the output layer.

A.2.3 The Output Layer

The hidden layers of a network are extracting features from the input data and passing the features along to the next layer. For simplicity, all of the nodes in the hidden layer usually have the same activation function and are treated the same. The difference between the different hidden layers are only in the number of nodes per layer. However, the output layer of the network is viewed differently. Mathematically, the operation done by the output layer is the same as the hidden layers, except with a different activation function. Conceptually, the output layer is drawing boundary lines on the input features to classify each input data into a distinct class. Referring to the basic neural network model shown in Fig. A.1, we see that the output layer only sees the output of the last hidden layer multiplied by constant weights. Each node of the output layer also calculates an activation of the form $O_1 = \sigma(\mathbf{w}_1 \cdot \mathbf{h} + b_1)$, where $\mathbf{w}_1$ are the weights for output node O_1 and $\mathbf{h}$ are the values of the nodes of the last hidden layer. The output nodes do not share weights, i.e $\mathbf{w}_1 \neq \mathbf{w}_2$. Since we want the network to output class scores, we want to force the output of the network to be in the range $[0, 1]$. This can be achieved in many ways.

For a binary classification network, only a single node is required in the output layer with values of 0 or 1. In this study, the logistic function was used for one of the outputs, shown in Table A.1. Written explicitly, the output of a binary classification network using logistic activation function is: $O = \frac{1}{1+e^{-(\mathbf{w}\cdot\mathbf{h}+b)}}$. The next step is then to calculate the loss function using the MSE or cross-entropy loss functions, given by Eqn. A.4 or Eqn. A.5. For every training data sample and its corresponding class score, the network outputs a value calculated by the logistic function and this value is used to calculate the contribution to the loss function for the sample. During training stages, the weights and biases of the hidden layers and the output layer are varied to minimize the loss function.

For classification of more than two classes, *softmax* activation is often used on the output layer. For this study, a 30 class classification network with 30 output nodes was trained using the softmax activation. Explicitly,

the softmax activation can be written as:

$$\sigma(\mathbf{w}_i \cdot \mathbf{h} + b)_i = \frac{e^{\mathbf{w}_i \cdot \mathbf{h} + b_i}}{\sum_{j=1}^{K} e^{\mathbf{w_j} \cdot \mathbf{h} + b_j}} \tag{A.7}$$

Here, $\sigma(\mathbf{w}_i \cdot \mathbf{h} + b)_i$ is the output of the ith node of the output layer. Equation A.7 shows that for each node in the output layer, the input to the node is exponentiated first to range $[0, \infty]$ and then normalized by the total of the output layer. Therefore, each node in the output layer has value in the range $[0, 1]$ and the total sum of the outputs of the output layer is equal to 1. The softmax activation is often thought of as giving probability of the data being in one of the classes.

For the basic model shown in Fig. A.1, the output layer has 2 nodes. We can write the output of one of the nodes explicitly as:

$$\sigma(\mathbf{z})_1 = \frac{e^{\mathbf{w}_1 \cdot \mathbf{h} + b_1}}{e^{\mathbf{w}_1 \cdot \mathbf{h} + b_1} + e^{\mathbf{w}_2 \cdot \mathbf{h} + b_2}} \tag{A.8}$$

Clearly, this output is in the range $[0, 1]$ and $\sigma(\mathbf{z})_1 + \sigma(\mathbf{z})_2 = 1$. Given the probabilistic interpretation of the softmax activation, the cross-entropy loss is usually applied to estimate the network inaccuracy. For this study, $K = 30$ for the number of classes and the cross-entropy loss was used to calculate the loss function of the network.

A.2.4 Gradient Descent and the Back Propagation

We now move to the problem of determining the weights and biases connecting every node in the network. This is achieved by an iterative approach because searching for an analytic solution would require too much compute time. The target is to minimize the loss function with respect to the weights and biases. Minimization can be achieved by the *gradient descent* algorithm. The principle behind the algorithm is very simple. For a given function, at each point we calculate the direction of greatest descent of the function and move incrementally in that direction. The direction of greatest descent is calculated easily by taking the negative of the gradient of the function. An example of the gradient descent algorithm is shown in Fig. A.4. Here, we have a function $z(x, y)$ that we wish to minimize by determining the values of (x, y) that minimize the function. In this example, we start at the top of the "hill" and wish to get to the "bottom" of the graph. To achieve this, we calculate the gradient of the function at the current position and step in the opposite direction. This will move the current position down the hill. Once we reach a new point, we repeat the gradient descent until we reach the bottom. At this point, we will have found the values of (x, y) that minimize $z(x, y)$.

In the previous section, we saw two common loss functions known as MSE and cross-entropy given by Eqn. A.4 and A.5. Here, we show how

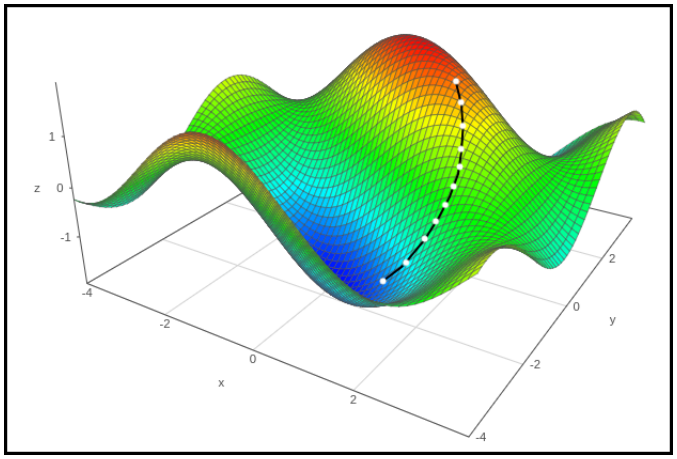


Figure A.4: An example of the gradient descent algorithm. We wish to find the values of (x, y) that minimize the function $z(x, y)$. This is achieved iterativaly by taking the gradient at the current point (x, y) and stepping in the direction opposite the gradient. Algorithmically it can be written as: $(x, y) \rightarrow (x - \alpha\frac{\partial z}{\partial x}, y - \alpha\frac{\partial z}{\partial y})$, where the parameter α controls the stepping size. In this example, we start at the top of the "hill" and iteratively take steps until we are at the "bottom" of the graph.

to find the best weights and bias constants $\mathbf{\Theta}$ that minimizes the chosen loss function. For simplicity, weights and biases are denoted by a single parameter, $\mathbf{\Theta}$. The rule to update each parameter can be written as:

$$\Theta_i \rightarrow \Theta_i - \alpha\nabla_{\Theta_i} J(\mathbf{\Theta}) \tag{A.9}$$

Each parameter Θ_i, representing a single weight or bias in some layer of the network, is updated by the negative of the gradient times α. The parameter α is called a **hyperparameter** and controls how much to step in the direction of the greatest descent. This parameter is usually set by the programmer to be a small constant. At the beginning of the network training, it can be large and as we approach the loss function minimum, the step size is decreased. For example the gradient of the MSE loss function given by Eqn. A.4 can be calculated with respect to the i'th parameter as:

$$\nabla_{\Theta_i} J(\mathbf{\Theta}) = \frac{\partial}{\partial\Theta_i}(\frac{1}{2N}\sum_j (h(\mathbf{x}^j) - y^j)^2) = \sum_j \frac{1}{N}(h(\mathbf{x}^j) - y^j)\frac{\partial h(\mathbf{x}^j)}{\partial\Theta_i} \tag{A.10}$$

Remember that $h(\mathbf{x}_j)$ is the output of the network which could have the form of the logistic function(Table A.1). In any case, they are chosen to be easily differentiable. The update of the parameters is also directly dependent on the derivative of the loss function and thus we try to avoid loss functions with vanishing derivatives. The index j is used to sum over all training examples

while index i is used to indicate the index of the parameters of $\mathbf{\Theta}$ with which we are taking the gradient with respect to. Θ_i could be some weight in an early layer of the network but it affects the final loss of the network, so it must be updated to minimize the loss. Since the network is a very simple mathematical construct, a series of nested activation functions leading to the final output, the derivative of the loss function with respect to any weight or bias term can be found by repeated application of the chain rule. Appendix B shows how to apply the chain rule on simple neural networks to obtain the gradient with respect to any parameter of the network.

The formulation above is used to update a single parameter(Θ_i) of $\mathbf{\Theta}$ and after each update step, the value of $h(\mathbf{x}^j)$ changes and so it must be recalculated at each step. By updating the parameters this way, we can move towards the minimum of the loss function. Conceptually, the network is learning to extract more and more relevant features in each of the hidden layers to obtain more and more accurate classification. Because we are taking fixed size steps in the direction of the gradient descent, we will not land on the exact values that minimize the loss function. Therefore, the learning rate α is slowly changed during training to reduce the step size taken after each iteration.

To apply the gradient descent algorithm to neural networks, the first step is to input the training data and allow the information to flow through the network and obtain the final network outputs. Using the network outputs for all available training data, the loss function is calculated. Next, we update the weights and biases of the network to minimize the loss function. To do this, starting with the loss function, the gradient of the loss function is calculated for each weights and biases of the network, going from the output layer to the input layer. This step is called the *back propagation*. After calculating the gradients, the weights and biases are updated using the formula given in Eqn. A.9. Once the parameters are updated, the process repeats until a satisfactory minimization of loss function is achieved.

This concludes the discussion on the basic version of the gradient descent algorithm. Once the gradient descent algorithm ends and the values that minimize the loss function are found, we can say that the machine has completed its learning process. However, there are several conceptual and implementation problems with this algorithm that we will discuss further in this section. The immediate concern with this algorithm is that it does not guarantee global minimum of the loss function. There are some strategies that help mitigate this problem but finding the global minimum is not guaranteed by the gradient descent algorithm. Given the millions of trainable parameters, finding the global minimum might not be possible with finite compute time.

One implementation problem comes from the form of the gradient of the loss function(equationA.10). Calculating the gradient requires summation over every training data, which could become expensive to compute if the

number of training data is too many. For the model designed for CANDLES, the model has 20 million trainable weights and biases and the GPU used to train the network could only hold parameters for about 100 training samples in memory at a time. The first modification to the gradient descent algorithm is the **Stochastic Gradient Descent**, where instead of summing over all training examples, only a small subset of the training examples are used as a batch. We can write the gradient of the loss function as:

$$\nabla_{\Theta_i} J(\mathbf{\Theta}) \approx \frac{\partial}{\partial \Theta_i}(\frac{1}{2}\sum_k (h(\mathbf{x}^k) - y^k)^2) \tag{A.11}$$

where k is a subset of the total number of events. This is a hyperparameter that is usually specified by the programmer. There are no exact methods to set this parameter and is often dictated by how much memory is available to hold the parameters during training.

Another implementation problem comes from the learning rate α that appears in equation A.9. This parameter controls the size of the step at each iteration of the gradient descent algorithm. If the step size is too large, we may completely miss the minimum of the function or simply jump over it back and forth. If the step size is too small, the number of training steps required can increase and so training time also increases. Intuitively, we may expect to see large step sizes at the beginning of the training and expect to see the step size become smaller as we approach the minimum. To emulate this, the stochastic gradient descent algorithm is modified so that $\alpha \to \alpha(t)$, where t is usually the number of training steps and the α parameter becomes smaller as more and more training steps are taken. This is called learning rate decay in neural network terminology.

Previously we mentioned that the gradient descent algorithm does not guarantee the global minimum of the loss function. If there exists a local minimum nearby, the gradient descent algorithm moves towards this local minimum. One way to search for a better minimum is to start several different training models and randomly initialize the parameters at different points in the parameter space and allow the parameters to descend to a minimum. If lucky, one of the initial parameters may end up at the global minimum. Of course this method also does not guarantee finding the global minimum. Moreover, this method could be difficult to implement because if each training regime takes several days to weeks, then starting several different initial parameters for separate training is not feasible.

One approach that is used to help mitigate the local minimum problem is to use a momentum based gradient descent algorithm. Conceptually, we are releasing a ball at some point in the loss function space and allow it to move to a minimum on its own. If a local minimum is found, the ball may have enough momentum to escape the local minimum and approach a better minimum of the function. This method is implemented using a bit

more memory and minor modification to the basic gradient descent. The gradient descent algorithm updates the parameters using Eqn. A.9. We modify this equation by calculating a momentum term before making the parameter update. The gradient descent parameter update is modified to be:

$$v_i = \rho v_{i,prev} + \nabla_{\Theta_i} J(\mathbf{\Theta}) \tag{A.12}$$

$$\Theta_i \rightarrow \Theta_i - \alpha v_i \tag{A.13}$$

where v_i is a new momentum term and $v_{i,prev}$ is the momentum of the previous step. The momentum of the previous step is stored in memory at the end of each step. At a new step, a new momentum term is calculated first by adding the previous update momentum to the current gradient of the loss function before making the update. The additional parameter ρ is an extra control parameter that allows the momentum to decay and has a value less than one. At the beginning of training regime, the previous momentum is simply initialized as 0 and momentum is added at each step. By making this single line change to the algorithm, the stochastic gradient descent method is able to overcome shallow local minima.

There are several other gradient descent methods that modify the basic gradient descent given by Eqn. A.9. These are not covered here because they were not used in this study. The popular among them are the Nesterov gradient descent and the adaGrad algorithms. An overview of different gradient descent algorithms can be found at [34]. These descent algorithms all aim to fix some shortcomings of the basic gradient descent algorithm or improve training time. But the choice of one optimization algorithm over another does not seem to make a large difference, at least with respect to the application in this study. Several different gradient descent algorithms were tested and they all achieve similar levels of accuracy.

A.3 Convolutional Neural Network

As we saw in the basic neural network model, the hidden layers of the network are extracting features from the input data. However, the number of trainable parameters increases to unfeasible size very quickly with a small number of additional nodes in the network. To combat this issue, convolutional neural networks(CNN) are used. The operation of *convolution* is intuitive and has a more literal interpretation of extracting features from input data. However, they are mostly applicable to image type data, which it found popularity in.

Suppose we have an image data that we need to classify. The image may have dimensions of 100x100 pixels and 3 color channels. If we connect a single node to this image, we have to connect to every pixel in the image, which means we have to save a weight parameter for each pixel in the image.

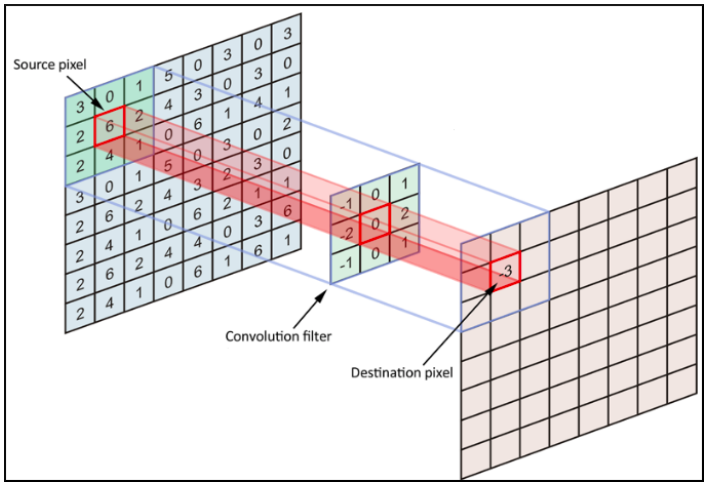


Figure A.5: Convolution operation is shown. In this example, a filter of 3x3 is used to scan the image. The center of the filter is places on a pixel in the image and a comparison is made between the filter and the image section. The comparison is made by multiplying each pixel in the image section and the filter element-wise and summing the result. Like before, an activation function of the form: $O = \sigma(\mathbf{w} \cdot \mathbf{M}_c + b)$ is used where $\mathbf{w}$ are the pixel values of the filter and $\mathbf{M}_c$ are the pixels of the image crop that the filter is placed on. The output of the convolution is saved on a matrix that preserves the location of the comparison pixel on the image. Once comparison is made, the filter is moved to the next pixel to make comparisons. Because the filter is compared to every pixel in the input image, the output of the convolution is a matrix with the same size as the input image. *image source: https://towardsdatascience.com/image-classifier-cats-vs-dogs-with-convolutional-neural-networks-cnns-and-google-colabs-4e9af21ae7a8*

For a single image and node pair, we have to save 30,000 values. On a modern computer, this is not a large value. However, if we double the image size, the number of parameters would increase exponentially. If we connect a few tens of node to the image, we see that the number of parameters quickly reach millions. Connecting nodes to an image is a computationally and memory expensive process.

Instead of connecting directly to nodes, a small sized filter(sometimes called kernel) is used to scan the image. Figure A.5 shows the convolution operation in action. The filter is overlaid on a same sized area on the image and compared to the filter. The comparison is done by multiplying the filter and the image region element-wise and summing. An activation function is applied to the comparison of image and filter at each position of the filter. That is, when the filter is overlaid on some part of the input image, the

activation is calculated by the formula:

$$O = \sigma(\mathbf{w} \cdot \mathbf{M}_c + b) \tag{A.14}$$

Here, $\mathbf{w}$ are the pixels of the filter and $\mathbf{M}_c$ is the crop of the image that the filter is overlaid on. Both $\mathbf{w}$ and $\mathbf{M}_c$ are flattened into a vector. This is the same operation used to calculate the output of a neural network node. But the number of trainable parameters are now reduced to the size of the filter. The output of the convolutional activation is saved in a matrix, as shown in Fig. A.5. The filter is overlaid on the input image by shifting the position by a single pixel and therefore, the output of the convolution is a matrix that has the same size as the input image. If portion of the filter falls outside the image border, then the input is simply considered 0 for those regions. This way, the input image and output matrix sizes are preserved.

Conceptually, the convolution operation is taking a filter and scanning the image to see if there is a match. This is a literal feature extraction operation where the extracted feature is the similarity of the section of the image to the filter. An important property of the convolution operations is that it preserves spatial information of the image. In other words, if the filter has a good match in two regions of the input image, those similar regions are also conveyed in the output matrix by the same distance as in the original image. This means that convolutional neural networks are able to convey correlation information in different parts of the image.

In practice, few tens to a few hundreds of filters are applied to the image in the first layer. Figure A.6 shows a CNN very similar to the one used in this study. If a single filter is used on the whole image, a single output matrix is generated, called an *activation map*. As observed in the figure, when multiple filters are used, multiple activation maps are generated. Each of these activation maps contain spatial information on which part of the image the filter was most similar to. These are the features that are extracted by the first convolutional layer. These activation maps are input into the next convolutional layer and the process repeats. Since the output of the first convolutional layer is 3 dimensional, the filters of the second convolutional layer are also 3 dimensional. The size of the filters are given by (N, M, D) where N and M are the (x,y) size of the filter and D is the depth, which is equal to the number of input activation maps. Therefore, the second convolutional layer filters "see" all of the activation maps at once as it scans the activation maps of the first layer. This has double significance. The first is that the second convolutional layer can learn to correlate spatial information contained in each activation map, which contain different information based on the filter. Second, it can correlate the outputs of different activation maps in certain regions of the image. For example, by seeing all of the activation maps at the same time, the second layer can see what kind of geometric shapes and what kind of color changes are present in that part of

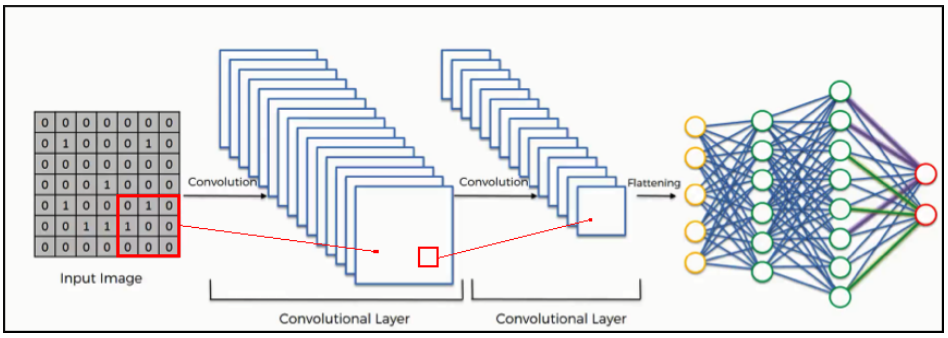


Figure A.6: Convolutional neural network example. When a single filter is used on the input image, it generates a single output matrix called the activation map. If many filters are used on the input image, it generates that many activation maps. These activation maps represent the features extracted by the filters and is input into the next layer. The next layer is also a convolutional layer. Each filter in the second convolution layer sees all of the activation maps at once. For example, it may have a size 10x10x64, where 64 is the number of activation maps. This means that the second convolutional layer is learning correlation between the filters of the first layer. Also, if the size of the filters of the second layer is large, it is able to learn more spacial correlations as well. Finally, the activation maps of the second convolutional layer are input into regular node network. This is done by connecting each entry in the activation maps to a single node. This step generated the most number of parameters in the network. *source: https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-summary/*

the image. These kinds of correlated features are extracted by the filters of the second convolutional layer.

In Fig. A.6, the second convolutional layer is connected to a regular node based network. This is achieved by connecting each pixel of the activation maps of the second layer to each node in the third layer. This is known as flattening. For each connection, a weight parameter is generated and thus the bulk of the parameters of the network are in this connection between convolutional layer and node layer. There are multiple ways to reduce the number of parameters in this connection but these are beyond the scope of this study and were not used in this study. One way to achieve this is to reduce the size of the output of convolutional layers, as discussed in [35]. There have also been models that do not use any node, as in [36].

The ultimate goal of the convolutional network is to classify the image. Therefore, we need to determine filters that result in best the classification accuracy or the minimum of the loss function. The first convolutional layer is scanning the input image with filters to check for similarity with the filters. These similarity features are passed on to the second layer. The second convolutional layer is scanning all of the activation maps from the first layer to create correlation between activation maps and spatial correlations. These features are then passed to the node based part of the neural network. The remainder of the network will extract further features and classify the image. Mathematically, the complexity of the CNN is not much different from a basic network. Therefore, the gradient descent algorithm with back propagation can train the network and the parameters of the filters.

With the gradient descent, the parameters(pixels) of the filters are learned that provide best classification accuracy. We could ask what kind of filters are being learned by the network. A visualization of the filters learned for image classification is shown for the first convolutional layer of a CNN in Fig. A.7. These filters were trained to classify images of day to day objects and scenes. We see that some filters are looking for changes in contrast with certain angles. These filters may be looking for edges in images. Other filters are looking for changes in colors or the overall color of the section of the image. Each of these filters are used to the scan the whole image and so the output of the activation map would give which part of the image has similarity to the filters. Once all filters are used to scan the images, the activation maps are passed to the next layer. The next layer can then check the correlation of the activation maps of the image. These second layer filters are high dimensional and highly complex. Therefore, it is difficult to visualize the filters and also difficult to see what the filters are looking for.

In conclusion, convolutional networks use more intuitive feature extraction operations to classify input data. These operations preserve the spatial locations of the interesting parts in the input data. Also, very large networks can be trained using convolutional networks because they use much less computer memory compared to conventional neural networks. These

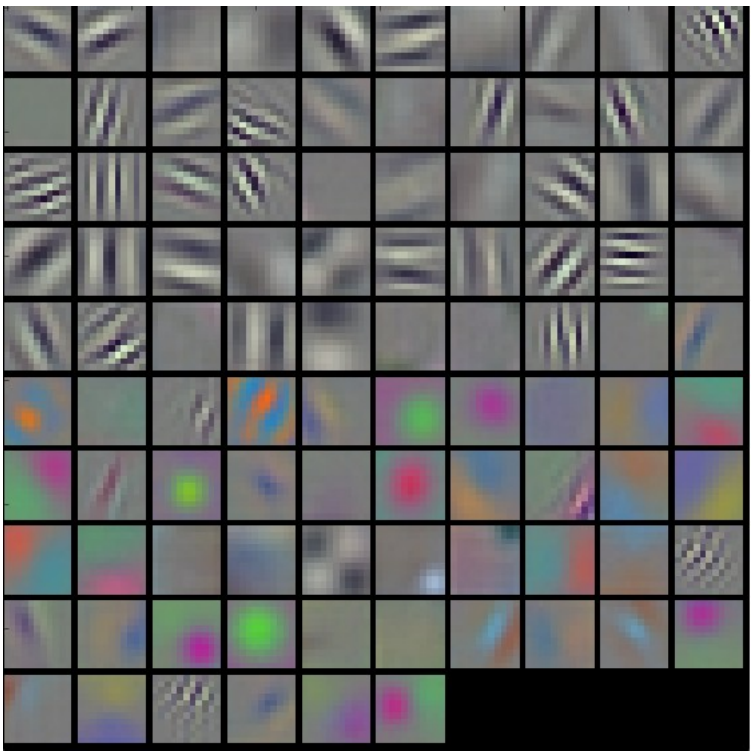


Figure A.7: The 96 filters of the first convolutional layer from an image classification network is shown. We see here that the filters are looking for very general features such as change in contrast, change in color, corners etc. When each of these filters are used to scan an image, we get an activation map that indicates where in the image these filters were highly activated. The second convolutional layer then looks at all the output activation maps and generates filters that make highly complex correlations between the filters of the first layer. We see then that the convolutional neural networks use very intuitive feature extraction steps. *source: http://cs231n.github.io/understanding-cnn/*

are the reasons why convolutional neural networks see popularity in image classification tasks. For CANDLES, this property of understanding spatial correlation in input data was put to use for classification of data. The details of the CANDLES CNN model is presented in Chap. 5.

A.4 Summary

Whenever we have some data available, we are often interested in classifying the data. This is traditionally done by extracting useful features from the data and based on these features, we attempt to classify it. Physicists are often interested in features such as energy, charge, position etc. These features can be extracted from data by applying mapping functions that take in raw data as input and output the relevant features. Design of these feature extracting functions require expertise in the field and is a time intensive process. Also, some data is highly complex or have such high dimensionality that it is difficult to design an appropriate mapping function for good features. In these cases, could use a computation based search for a good feature extracting function. This can be achieved by using a neural network.

Neural networks are capable of approximating any well behaved function, as stated by the universal approximation theorem. We can make use of this property by defining an appropriate loss function and asking the network to minimize the loss function. The network uses the gradient descent algorithm with back propagation to search the parameter space and finds the weights and biases that minimize the loss function. Conceptually, the network is extracting the most important features from the input data that achieves the best classification accuracy. The universal approximation theorem guarantees us that if a feature extraction function exist, then a neural network will be able to approximate it. The construct of the neural network is not very complex. It is simply made of a series of nested activation functions and the activation functions themselves are also very simple functions. Working together, the network is a very computationally intensive search for best feature extraction functions. Given a large enough data set, the network achieves very good data classification, and its ability to do this automatically has led to its recent popularity.

Convolutional neural networks are memory efficient feature extraction operations. They look for very specific features defined by the filters and the spatial information of the features are preserved in the convolution operation. This property was one of the basis of using a CNN for CANDLES data analysis.

There are many drawbacks to using neural networks. Neural networks require a very large set of training data to avoid over fitting. These data need to be pre-classified because we have to compare the class of each training

data to the network output. In practice, creating labeled data is a time intensive process. Training a network also requires a very powerful computer to train in any reasonable amount of time. The CANDLES models were trained over a period of about 4 weeks on an NVIDIA Geforce GTX1060 GPU. Another drawback to using neural networks is that we do not know what kind of features are extracted by the network. Therefore, if we want to use the network to classify data that has some differences with the type of data it was trained on, then we can not simply modify some part of the network to suit the new data. We have to train large parts or the whole network to work on different set of data. However, the main drawback of using neural networks in physics analysis is that we cannot naturally estimate the uncertainty or error of the network. For physics analysis, it is often the uncertainty on the classification of the data that is more relevant than the classification itself. Therefore, it would seem that neural networks are ill suited for physics analysis. In this study, we ask if neural networks can still be useful to physics analysis with an external estimate of its uncertainty.

Finally, the design and the reasons behind the design of the model used for CANDLES data analysis is presented in Chap. 5. The content of this chapter is enough to follow the rest of this study. However, neural network research is now a very large and very active field with many new concepts published every week. These are beyond the scope of this study and we would refer to [40] for some recent topics on the subject.

Appendix B

Gradient Descent by Back Propagation

The objective of neural network training is to minimize the loss function through gradient descent. We consider the simple mean squared error(MSE) loss function given by:

$$J(\mathbf{W}, \mathbf{b}, \mathbf{x}) = \frac{1}{2N} \sum_{i}^{N} (h_{out}(\mathbf{x}^i) - y^i)^2 \tag{B.1}$$

where $h_{out}(\mathbf{x}^i)$ is the network prediction for the $\mathbf{x}^i$ sample and y^i is the true class label of the same sample. When the network is completely accurate, $h_{out}(\mathbf{x}^i)$ and y^i are equal for all samples, giving zero for the loss function. Our goal is to minimize the loss function with respect to the weights $\mathbf{W}$ and bias $\mathbf{b}$ when the loss function is non-zero.

To minimize the loss function, iterative gradient descent algorithm is used. The algorithm states that for any trainable parameter in the network, we take the gradient of the loss function and step in the direction of steepest descent. This can be written as:

$$w_i \to w_i - \alpha \nabla_{w_i} J(\mathbf{W}, \mathbf{b}, \mathbf{x}) \tag{B.2}$$

where w_i is any trainable parameter in the network and α is the step size called the learning rate. In this section, we explicitly derive the update for some w in a simple neural network.

Figure B.1 shows a simple neural network with a single output node. We wish to minimize the loss function with respect to weight w_1. The influence of w_1 through the remainder of the network is shown by the directional arrows. The loss function is defined as an average over all of the training samples. For a single training sample, the contribution to the loss function is proportional to $j_i = \frac{1}{2}(h_{out}(\mathbf{x}^i) - y^i)^2$. Assuming that the other w_i weights

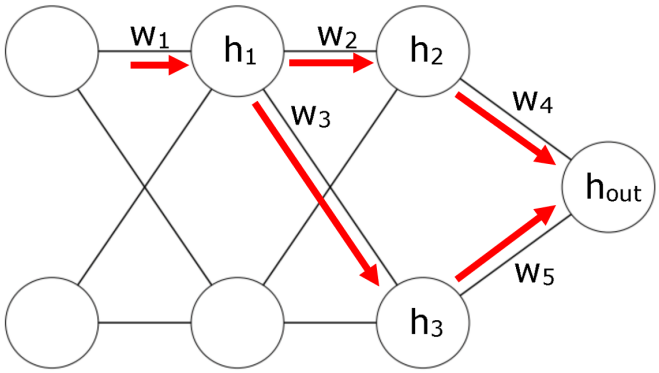


Figure B.1: A simple neural network with a single output node is shown. In neural network training, the loss function must be minimized with respect to the trainable parameters(w_i) for the given training sample. Each w_i can affect the final output by passing through several other nodes. The influence of w_1 on the remainder of the network is shown in red arrows.

are constant for the partial derivative, the partial derivative of the loss function for a single training sample can be taken with respect to w_1 as:

$$\frac{\partial j_i}{\partial w_1} = \frac{\partial j_i}{\partial h_{out}(\mathbf{x}^i)} \frac{\partial h_{out}(\mathbf{x}^i)}{\partial w_1} = (h_{out}(\mathbf{x}^i) - y^i) \frac{\partial h_{out}(\mathbf{x}^i)}{\partial w_1} \tag{B.3}$$

where $h_{out}(\mathbf{x}^i)$ is the activation of the output node. Some activation functions are shown in Table A.1 and in this example, it is a function of the nodes h_2 and h_3. Thus we can write:

$$\frac{\partial j_i}{\partial w_1} = \frac{\partial j_i}{\partial h_{out}(\mathbf{x}^i)} \left[\frac{\partial h_{out}(\mathbf{x}^i)}{\partial h_2} \frac{\partial h_2}{\partial w_1} + \frac{\partial h_{out}(\mathbf{x}^i)}{\partial h_3} \frac{\partial h_3}{\partial w_1} \right] \tag{B.4}$$

The h_2 and h_3 nodes are dependent on w_1 through h_1. Thus:

$$\frac{\partial j_i}{\partial w_1} = \frac{\partial j_i}{\partial h_{out}(\mathbf{x}^i)} \left[\frac{\partial h_{out}(\mathbf{x}^i)}{\partial h_2} \frac{\partial h_2}{\partial h_1} + \frac{\partial h_{out}(\mathbf{x}^i)}{\partial h_3} \frac{\partial h_3}{\partial h_1} \right] \frac{\partial h_1}{\partial w_1} \tag{B.5}$$

gives the change in j_i with respect to w_1. The partial terms appearing in the equation are scalar numbers for a given training sample $\mathbf{x}^i$. For example, if we assume a linear activation function for all the hidden layer nodes, then the partial terms given by $\frac{\partial h_i}{\partial h_j}$ are simply the weights w_k that connect the nodes. The advantage of using simple activation functions is that actual partial derivative calculations are not necessary if it can be written down in terms of the weights of the network.

The parameters update rule can be written more explicitly as:

$$w_i \to w_i - \alpha \nabla_{w_i} J(\mathbf{W}, \mathbf{b}, \mathbf{x}) = w_i - \alpha \left(\frac{1}{N} \sum_{k}^{N} \frac{\partial j_k}{\partial w_i} \right) \tag{B.6}$$

In deriving the equations above, we started with the loss function and applied the chain rule from the end of the network to the parameter we wish to optimize the loss function with. This backwards application of the chain rule is called the back propagation algorithm. From the example above, we can see that the method can be simply generalized to any size network. On modern neural network training frameworks, this step is done automatically at the beginning of the network definition. However, if a custom loss function is defined, then the derivative $\frac{\partial j_i}{\partial h_{out}(\mathbf{x}^i)}$ may need to be defined. For a more in depth look at how machine learning libraries handle differentiations, please refer to [44].

Appendix C

Network Prediction Checks

Due to programming error, double-pulse events with 0 decay time(0DP) were not specifically made for network training. However, due to uncertainty in defining the start channel of an event, when false DP events are made for network training, some 0DP events are accidentally made. Here, we check how 0DP events are predicted by the network. Moreover, the network prediction of DP events with negative decay time is checked.

We first check 0DP events. For testing, two different data sets were created. In the first, 0DP events were generated in the same way that the network training data was generated(Sec. 5.6). We combine event signal of ^{208}Tl with event signal of $^{214/215}$Po with zero time difference between the start of the two events. For the test data set, we generated 5,000 0DP events by combining events from crystal#1 and used MIMO models trained for crystal#1 to obtain predictions. The result is shown in Fig. C.1.a. We observe that most of these 0DP events are predicted as 1ch decay time DP events while only about 15% are predicted as 0DP events. This is expected behavior because the network was trained to put only SP events in bin#0. Accidentally occurring 0DP events are trained as 1ch decay time DP events.

The time resolution of each prediction bin is 2 ns. Thus, DP events with less than 2 ns decay time would be considered 0DP events. Depending on the exact decay time of an event, some DP information can occur in the photon count data of each PMT. However, in general, most of the usable information that can separate DP and SP events is contained in the different pulse shapes of the α and β events. We need to check if the network is indeed using pulse shape information to make SP and DP classification when decay time is very small.

Our assumption is that the network understands the difference between the β and α pulse shapes. To check this, we selected DP events predicted into bin#1 by MIMO network and with SI value greater than 0.4 and replaced their PMT data information with SP event's PMT data information predicted into bin#0 and SI < 0.2. Thus, this dataset has events with DP

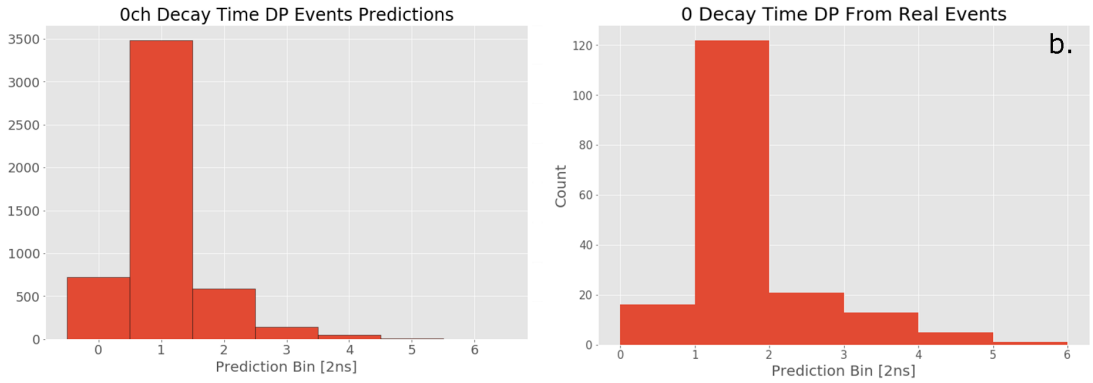


Figure C.1: MIMO network predictions of 0 decay time DP events. a.) The 0DP events were generated by combing donor ^{208}Tl and $^{214/215}$Po events. The generated events were predicted by crystal#1 MIMO network. b.) The 0DP events were generated by taking real DP event with 1ch decay time and replacing the PMT data with that of a known SP event. In both cases, the 0DP events were mostly predicted into bin#1, confirming that the network understands the difference between α and β pulse shapes.

pulse shape information and SP individual PMT data information. This data set simulates 0DP events with no usable information in the PMT data branch of the network. If indeed the network understand β and α pulse shapes, then these events should be predicted as 1 channel decay time DP events, based only on the pulse shape information. The result of applying the MIMO network to these events is shown in Fig. C.1.b.

We again observe that 0DP events generated from real data are also predicted as 1ch decay time DP events. This confirms that the network is able to distinguish between SP and DP events based only on the pulse shapes. Because the candidate events necessary to generate these events are few, only a few hundred events were generated and tested by the network. Nevertheless, we observe about 15% DP events in bin#0, which is consistent with the result obtained from generated 0DP events.

Next, we check DP events with negative decay time(ntDP). These events are not possible in principle but can occur in the data due to detector response and event position. The test events are also generated from donor ^{208}Tl and $^{214/215}$Po events. For the ntDP case, the alpha events occur 1 channel before the β events. As before, 5,000 ntDP events were generated for crystal#1 MIMO network and the resulting predictions are shown in Fig. C.2. We observe that most ntDP events are predicted in bin#1. Because the network uses pulse shape information to classify DP and SP events, even if the observed PMT data has the α and β signals reversed, the network is able to classify DP events.

Finally, we check DP events with 1 channel decay time(1DP). As before,

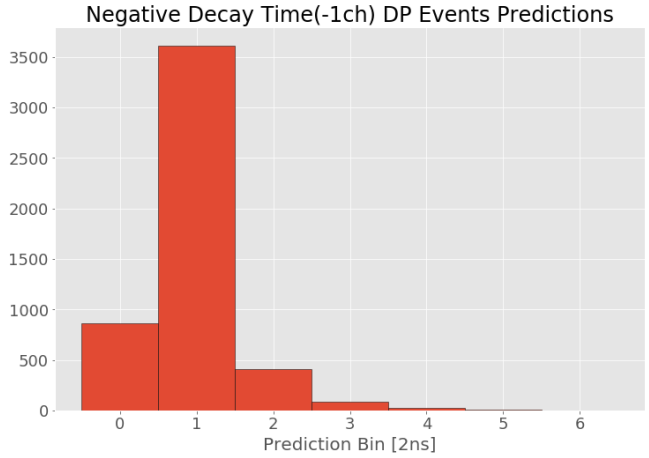


Figure C.2: Negative decay time DP events predicted by MIMO network. Because the network uses pulse shape information to classify between SP and DP events, the network places negative decay time DP events into bin#1.

these events are generated from donor events and 5,000 1DP event were generated for crystal#1. The resulting predictions are shown in Fig. C.3. We can observe that most 1DP events are correctly classified in bin#1. However, we observe significant counts in both bin#0 and bin#2.

There are multiple reasons that can cause the network to make incorrect predictions. The obvious first reason is that the network has an inherent uncertainty due to its limited training that causes it to make errors. This uncertainty could be improved by training the network further on more data. Another reason why the network would make errors is due to the pulse shape resembling a β pulse more than α pulse. These events are likely being predicted into bin#0. Finally, the generated DP events used for testing are not representative of real data. This is because when two donor events are combined to make DP event, the start channel of the two events are not completely known. As a result, DP events with differing decay times could be made. When 1DP events are generated, we are also generating number of other decay time events, which the network is predicting into other bins. This is partially the reason why we see significant number of DP events in bin#2 and above.

Moreover, the donor events are low energy events in the 2.0 MeV to 2.6 MeV range. The detector resolution improves with increasing energy. Thus, combining two low energy events with higher uncertainty results in a combined event with higher uncertainty than a real event of the same energy. With these considerations, the test data used in this section to check network predictions is understood to be a slightly poor representation of real data. The testing results do not carry over directly to real data. We

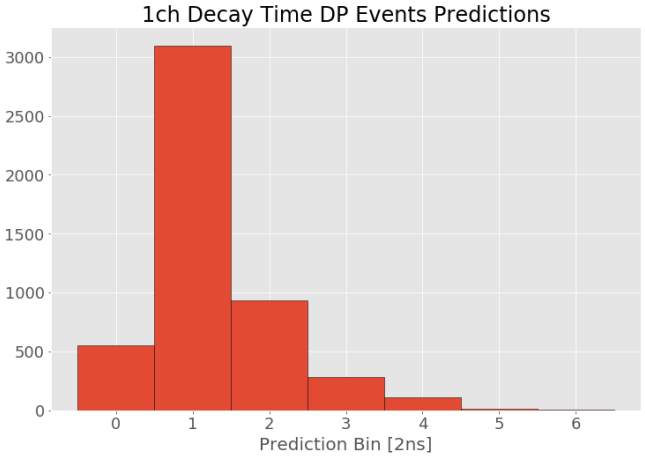


Figure C.3: 1 ch decay time DP events predicted by MIMO network. The extra events observed in bin#2 and above are likely due to the uncertainty in the actual decay time of the generated DP events.

should expect to see slightly improved results on real data.

Appendix D

Shape Indicator Fitting

Shape indicator(SI) is a parameter used to distinguish between α and β pulse shapes(Sec. 3.7). An SI value of 0 indicates a β pulse and a value of 1 indicates an α pulse. Double-pulse(DP) events have both α and β signatures, which causes the SI values to vary. The SI values can be used as an external check on the network predictions. Here, we show the analysis of SI values. The main result shows that the MIMO network predictions are consistent with the exponential decay curve of the DP events.

Figure D.1 shows the SI values of CANDLES data predicted into bins by the MIMO network and the PSDDoubleP method. Alpha coincident events are mostly known SP ^{208}Tl events with about 10% accidentally α coincident event counts. The non-alpha coincident events are primarily DP events and the events in bins#0 and #1 include SP events from other sources. The data set is split into α coincident(SP) and non-alpha coincident(mostly DP) events.

In Fig. D.1.a, we see the SI distribution of α coincident events predicted by the MIMO network. Since these are ^{208}Tl events with no α emission, we expect SI values of 0. Moreover, because these are SP events, we expect to see them in bin#0. In the figure, we observe most of the events in bin#0 and bin#1, as expected. The events observed above bin#1 are due to accidental α coincidence. Events in bin#0 are correctly centered around SI value of 0. However, the events in bin#1 are not center around 0. We saw in App. C that the network is able to understand the difference between α and β pulse shapes. It is likely that the reason why SI values are different between bin#0 and bin#1 events is because the pulse shape of SP events in bin#1 are more α like. In any case, SI distribution is consistent with expected distribution for SP events.

Figure D.1.b shows the SI distribution of non-alpha coincident events predicted by the MIMO network. We can clearly see the separation between the DP events and the SP events. Events above bin#1 are distinguishable from events in bins#0 and #1. DP events with longer decay times have more

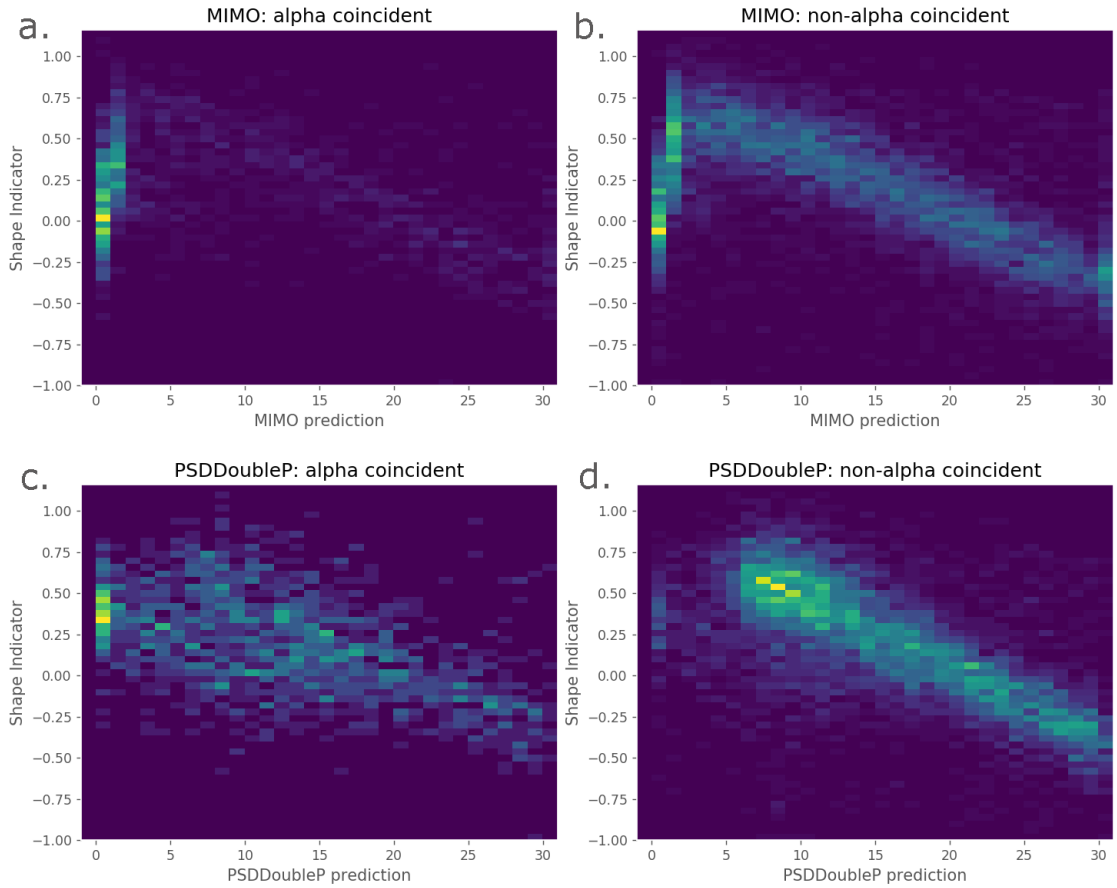


Figure D.1: Shape indicator values are shown by α coincidence and DP detection method. a) Alpha coincident events and prediction by MIMO network. Since these are mostly ^{208}Tl events, these should have SI value near 0. b.) Non-alpha coincident events and predicted by MIMO network. These contain DP events and SP events from sources other than ^{208}Tl. We can observe the separation of DP events and SP events by the jump in the SI values near bin#1. c.) Alpha coincident events and predicted by PSDDoubleP method. The SI value of these events are not cleanly centered at 0. d.) Non-alpha coincident events and predicted by PSDDoubleP. The remaining SP events are not well separated.

β like signal at the beginning of an event. Because SI values are calculated by comparing each channel of an event with that of the reference α and β pulses, DP events create a diagonal SI pattern due to the β signal early in the event's pulse shape. Thus we observe no obvious inclusion of DP events in bin#0 and some DP events in bin#1.

From α coincidence study(Sec. 6.2.2), we observed that most SP events are contained in bins#0 and #1. This is also confirmed by observing the lack of β like events in bins#2 to #30. We will look at various bins in more detail momentarily. The result of the proceeding analysis will show that through SI values acting as external evidence, the MIMO network predictions are accurate down to bin#2.

In figures D.1.cd, the SI value distributions obtained from PSDDoubleP predictions are shown. The expected distribution is not obtained in the α coincident events case. The SI values are not centered around 0 for bin#0 events. Moreover, the PSDDoubleP method places too many SP events in higher numbered bins. For the non-alpha coincident events case, the PSDDoubleP method reproduces the expected diagonal pattern of SI values. However, the method is unable to distinguish DP events under 10ch decay time very well and we see very few events in lower numbered bins. These distributions show the improvement made by the MIMO network over the PSDDoubleP method.

We now look at relevant bins in more detail. Figure D.2.a shows the SI distribution of events predicted into bins#2,3,4,5 by MIMO network. The distributions are separated by α coincident(blue) and non-alpha coincident(red) events. In the non-alpha coincident events case, we can observe an α like events distribution representing the DP events. From α coincidence analysis(Table 6.1), the MIMO network incorrectly places about 11% of SP events in these bins. These SP events are also observed by the small distribution centered near 0. For the α coincident events case, we see two small distributions, representing the incorrectly predicted SP events and accidentally α coincident DP events.

In Fig. D.2.b, the SI distribution of events predicted into bin#1 are shown. The α coincident and non-alpha coincident events distributions are nearly identical with the exception of more α like events, which are the expected DP events. An important observation is that the SP events in bin#1 are not centered near 0. As we saw in App. C, the network understand the difference between β and α pulse shapes. Because the SI values of events in bin#1 are not entirely β like(SI not equal to 0), we can conclude that the network is placing these SP events in bin#1 due to their α like pulse shapes.

Finally, in Figure D.2.c, the SI distributions of events in bin#0 are shown. We observe that the distributions are identical. Moreover, there are only a handful of events in both distributions with SI value above 0.5, indicating no significant number of DP candidate events. The α coincident events are known SP events from ^{208}Tl background. There are about 11%

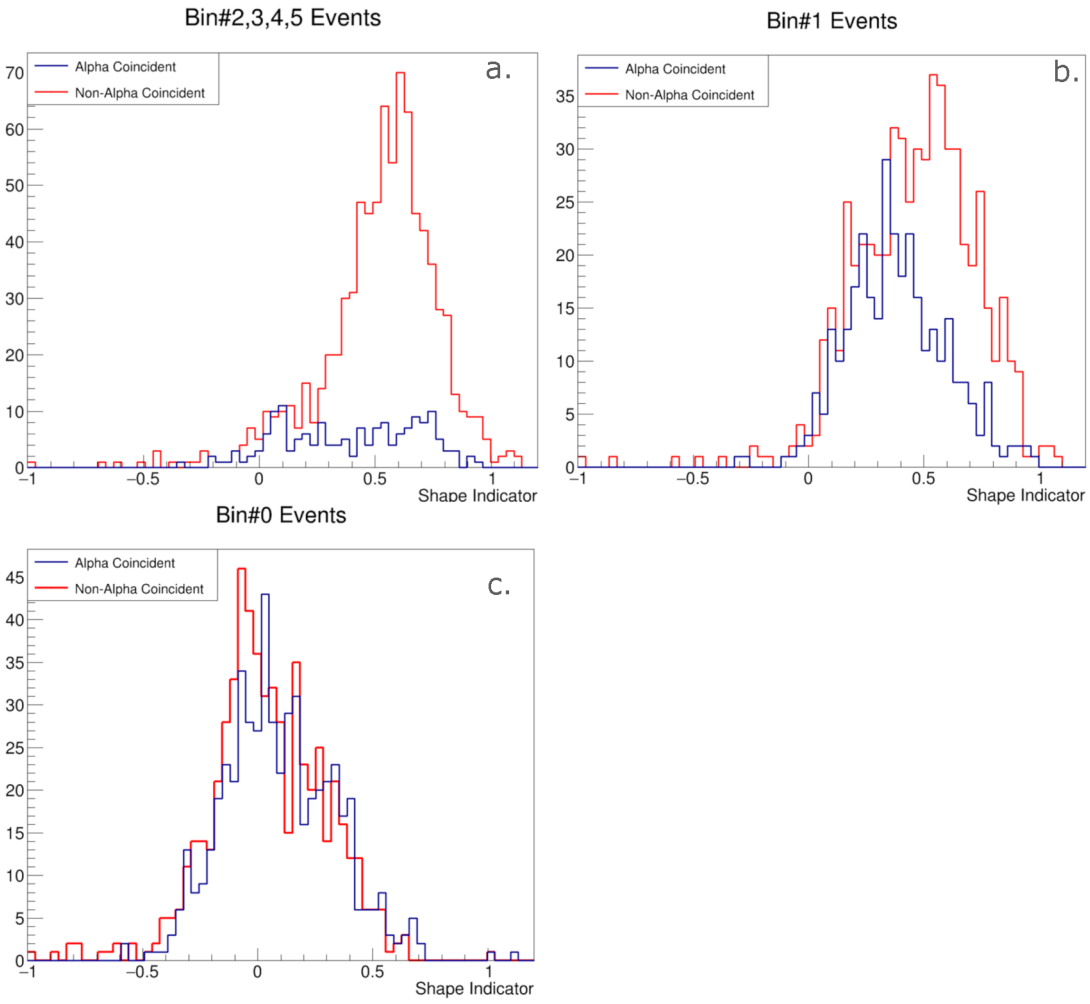


Figure D.2: The SI values of important bins are shown for events predicted by the MIMO network. a.) SI values of α and non-alpha coincident events are shown for bins#2,3,4,5. The non-alpha coincident events are primarily DP events with a mean SI value of 0.58. There are also incorrectly classified SP events in these bins and they are visible as a small peak around the value of 0.1. For the α coincident events, we see a roughly equal number of DP and SP events. b.) SI values of events in bin#1. Both α coincident and non-alpha coincident events contain significant number of SP and DP events. The non-alpha coincident events have a much larger number DP events. The DP events appearing in the α coincident events count is due to accidental coincidence with α event. c.) SI values of events in bin#0. This bin is for SP events. We see that the shape of the distributions is identical, which indicates the absence of DP events in these bins.

accidental α coincident events. However, these accidentally α coincident events are also from bin#0, taken from the non-alpha coincident spectrum, which are also SP predicted events. Thus, we conclude that there are no significant number of DP events among the α coincident events in bin#0. Due to the SI spectrums being identical, we can also conclude that there are no significant number of DP events among the non-alpha coincident events in bin#0.

We can estimate the number of SP and DP events in each bin by fitting the SI distributions. To obtain the DP events distribution parameters, we apply Gaussian plus Gaussian fitting to non-alpha coincident events spectrum in bins#2,3,4,5, as shown in Fig. D.5.a. The χ^2 value of the fitting is 35.8/41. The fitted mean SI value of the DP distribution is (0.58 $\pm$ 0.17).

The number of SP events estimated from the smaller Gaussian curve is (80.2 $\pm$ 15.0) events. The expected number of SP events in these bins can be calculated from the SP efficiency in Table 6.1 and observed number of events in bins#0 and #1. The non-alpha coincident event count in bins#0 and #1 is (1235 $\pm$ 35.1). From the expected DP exponential curve, the estimated number of DP events in bins#0 and #1 is (410.8 $\pm$ 19.2). Thus, the number of SP events in bins#0 and #1 is (825.2 $\pm$ 36.5). From the SP efficiency in Table 6.1, we see that 9.1% of SP events are expected in bins#2,3,4,5. This gives us an expected SP count of (75.1 $\pm$ 3.3). We conclude that the expected number of SP events(75.1 $\pm$ 3.3) is consistent with the observed count of SP events (80.2 $\pm$ 15.0) in bins#2,3,4,5. Thus, the SP efficiency curve is consistent with observed data.

The DP only count in bins#2,3,4,5 after reduction of SP count is (772.8 $\pm$ 37.4). The expected number of events from the exponential decay curve is (809.0 $\pm$ 28.2) events. The observed and expected DP counts are within uncertainty ranges. Combined with the SP expected and observed counts, we conclude that the observed SP and DP events in bins#2,3,4,5 are consistent with the expected number of events. Figure D.4 shows the SI distribution of events in bins#6,7,8. We observe no obvious SP distribution. From the SP efficiency estimates, less than 1% SP events(less than 10 events) are expected in these bins. Thus, we conclude that the network is consistently predicting DP events down to bin#2.

We next look at the SI distribution of events in bin#1 and its fitting in Fig. D.3.b. The DP distribution parameters were previously obtained from the fittings to events in bins#2,3,4,5(Fig. D.3.a). For events in bin#1, the DP distribution mean and deviations are constrained to the uncertainties of the previous fitting. The SP distribution parameters and the normalization of the DP distribution are not constrained. The χ^2 of the fitting is 41.5/41. However, visual observation shows that the fitting suffers from low number of statistics per bin.

The DP count estimated from the fitting is (399.8 $\pm$ 23.9) events. The expected number of DP events in bin#1 is (204.9 $\pm$ 14.1). However, in App.

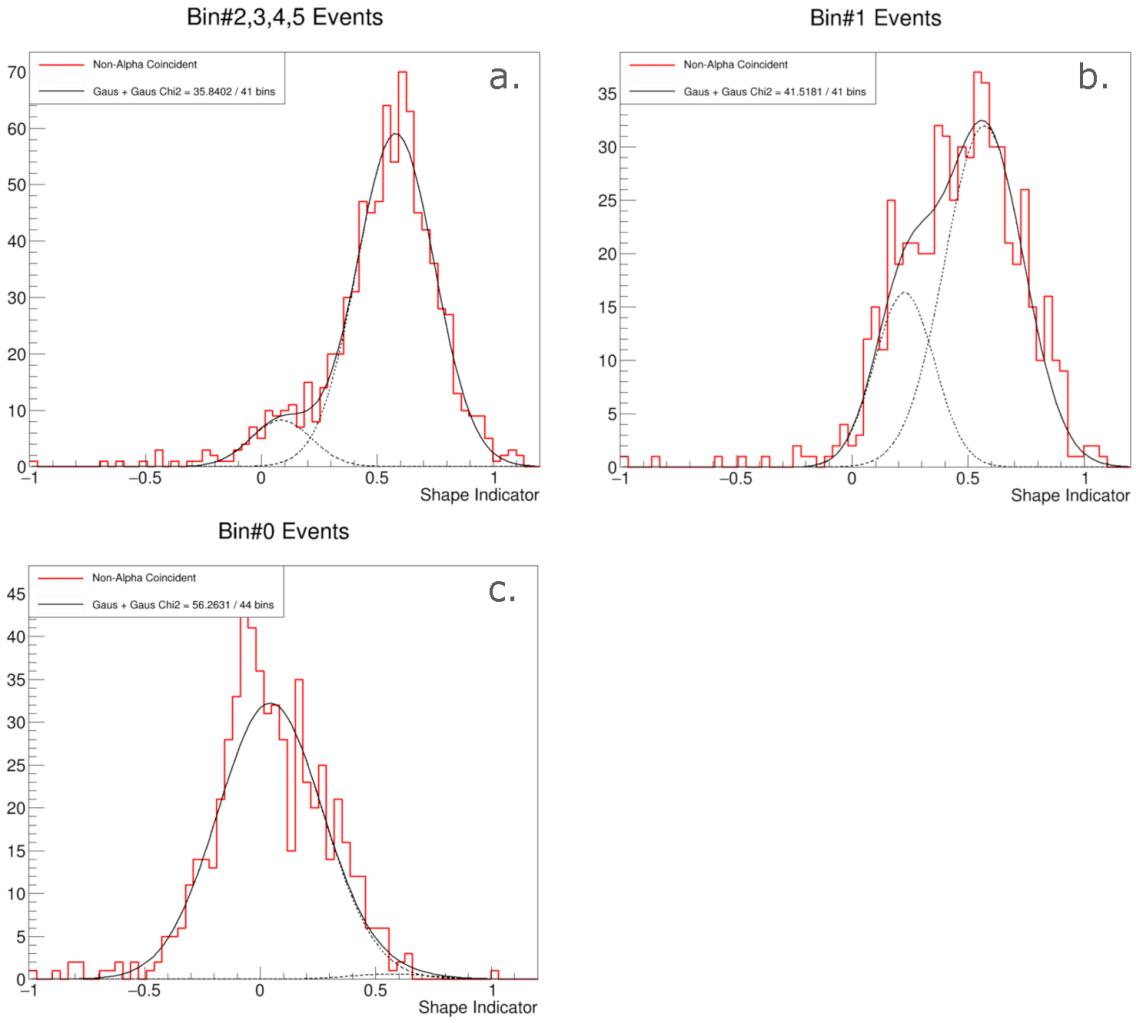


Figure D.3: Various fittings on the SI values are shown. a.) Sum of Gaussians is fitted to bin#2,3,4,5 events to obtain the mean and deviation of the DP peak. b.) Using the mean and deviation of the previous fitting, we apply a constrained fitting to events in bin#1 to obtain the count of SP and DP events. c.) The constrained fitting on bin#0 events fail. This indicates the absence of DP events in this bin.

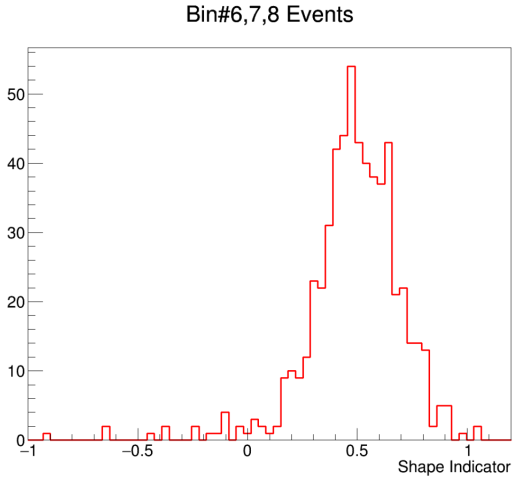


Figure D.4: SI distribution of non-alpha coincident events in bins#6,7,8 are shown. We observe no obvious SP distribution around 0. From SP efficiency estimates, less than 10 SP events are expected in these bins.

C, we showed that most DP events with 0 decay time will also be predicted into bin#1. The expected number of DP events in bins#0 and #1 is (410.8 ± 19.2), which is obtained from the exponential decay curve. The DP count observed in bin#1 is consistent with the total number of DP events expected in bins#0 and #1. This suggests that almost all DP events are in bin#1.

In Fig. D.3.c, we apply Gaussian plus Gaussian fitting to events in bin#0. As before, the DP events distribution parameters are constrained to the uncertainties. We can observe that the constrained fitting fails. Moreover, no obvious DP event candidates are observed in the SI distribution. If there are no DP events, then our previous observation that all DP events are in bin#1 is consistent. However, we saw in App. C, that some DP events are expected in bin#0, although we also commented that the result is not entirely reliable and fewer DP events may be observed in real data.

We also observed previously(App. C) that the MIMO network is able to distinguish between β and α pulse shapes. If some DP events have more β like pulse shape, then we should expect the network to predict these events into bin#0. In this case, SI analysis would not be able to separate SP and DP events in bin#0 because these DP events look like β events. However, because the SI distributions of α coincident and non-alpha coincident events in bin#0 are identical, and the fact that α coincidence is a much stronger and independent evidence for SP events, we could conclude that there are no DP events in non-alpha coincident bin#0. Since these issues could not be resolved by using SI as external evidence, in this study, the recommended cut-off point is chosen to be bin#2. That is, we can safely reject events in bins#2 and above as DP and keep events in bins#0 and #1.

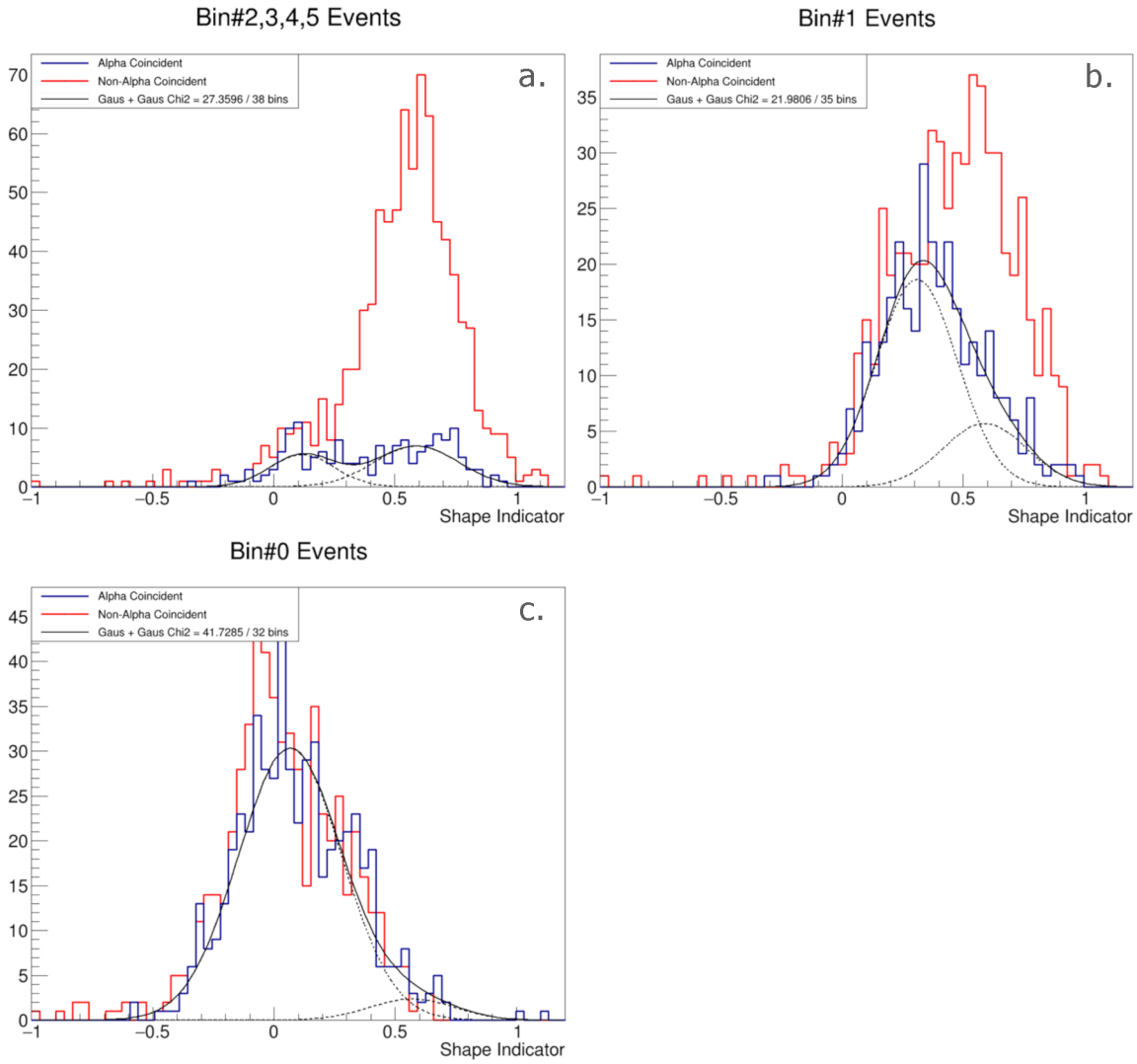


Figure D.5: Fitting the SI values for the α coincident events. The non-alpha coincident events distribution is also shown as reference. The DP distribution is constrained by the values obtained from previous fittings. a.) Bins#2,3,4,5 events are fitted. These events contain roughly equal number of DP and SP events. b.) Bin#1 events are fitted. c.) Bin#0 events are fitted. Again, poor fitting is obtained for the DP distribution.

For completeness, Fig. D.5 shows the SI distribution fittings of α coincident events in various bins. Similar SP and DP distribution fittings are observed. However, due to the low number of events, reliable conclusions could not be drawn. As before, Gaussian plus Gaussian fitting fails on events in bin#0. This is expected because we do not expect any DP events in bin#0.

In summary, we used SI values as an external evidence to confirm the MIMO network predictions. No obvious number of SP events were observed in MIMO bins above bin#6. Some number of SP events were observed in bins#2,3,4,5 and the observed count was shown to be consistent with the expected number of SP events in these bins. Moreover, the remaining DP count in these bins were also consistent with the expected number of events from the exponential decay curve. Thus, we concluded that the MIMO network can be reliably used down to bin#2. We showed that the number of DP events observed in bin#1 is consistent with the expected number of events in bins#0 and #1. As shown in App. C, the MIMO network places most 0 decay time DP events in bin#1. Moreover, we observed no significant count of DP events in bin#0. Thus, the observed number of DP events in bin#0 is not surprising. However, we also observed previously that the network is distinguishing SP and DP events by their pulse shapes. This means that the DP events predicted into bin#0 likely have β like pulse shape, in which case, SI would not be able to verify them as DP. On the other hand, the α coincident and non-alpha coincident SI distributions are identical, indicating that there are no DP events in bin#0. This would be consistent with the observation that all the DP events are in bin#1. However, since this could not be confirmed conclusively, in this study, we set the cut-off point to be bin#2 and above.

Appendix E

SP Check by Alpha Coincidence

The ^{208}Tl background events are detected by coincidence with the α decay event of the parent ^{212}Bi. The parameters used for this method are designed to remove maximum number of ^{208}Tl backgrounds without reducing too many signal events. As a result, many ^{208}Tl events remain in data after reduction. These events are expected in the non-alpha coincident predictions of the MIMO network. We can use these events to confirm the MIMO network predictions. However, due to the low number of events, the conclusion drawn is less reliable than the shape indicator analysis(App. D) or the α coincidence analysis(Sec. 6.2.2).

Figure E.1 shows the preceding and proceeding events' energy spectrums of non-alpha coincident events predicted into various bins by the MIMO network. The preceding spectrum is obtained by collecting all events occurring 1080s before the candidate high energy event that is predicted by the MIMO network into various bins. Similarly, the proceeding energy spectrum is obtained by collecting all events occurring 1080s after the high energy event. The proceeding spectrum is collected to obtain expected number of events in each bin. The 1080s search range is equal to 6 half-lives of the parent ^{212}Bi. Only crystal cut is used to make these spectrums. If ^{208}Tl events exist in the data, we would be able to see the α decay particle of the parent ^{212}Bi in the preceding energy spectrum. The expected energy region($\pm 2\sigma$) of the α decay event is indicated by the vertical bars.

In Fig. E.1.a, the preceding and proceeding events' energy spectrum for events predicted into bin#0 are shown. The excess events observed in the preceding spectrum are the α events of the parent ^{212}Bi. The number of excess events can be counted by taking the difference of the preceding and the proceeding spectrums. For the bin#0 events, the difference is shown in Fig. E.2. Then, the excess α count can be obtained by summing bins in the $\pm 2\sigma$ energy range of the α energy peak. Table E.1 shows the observed

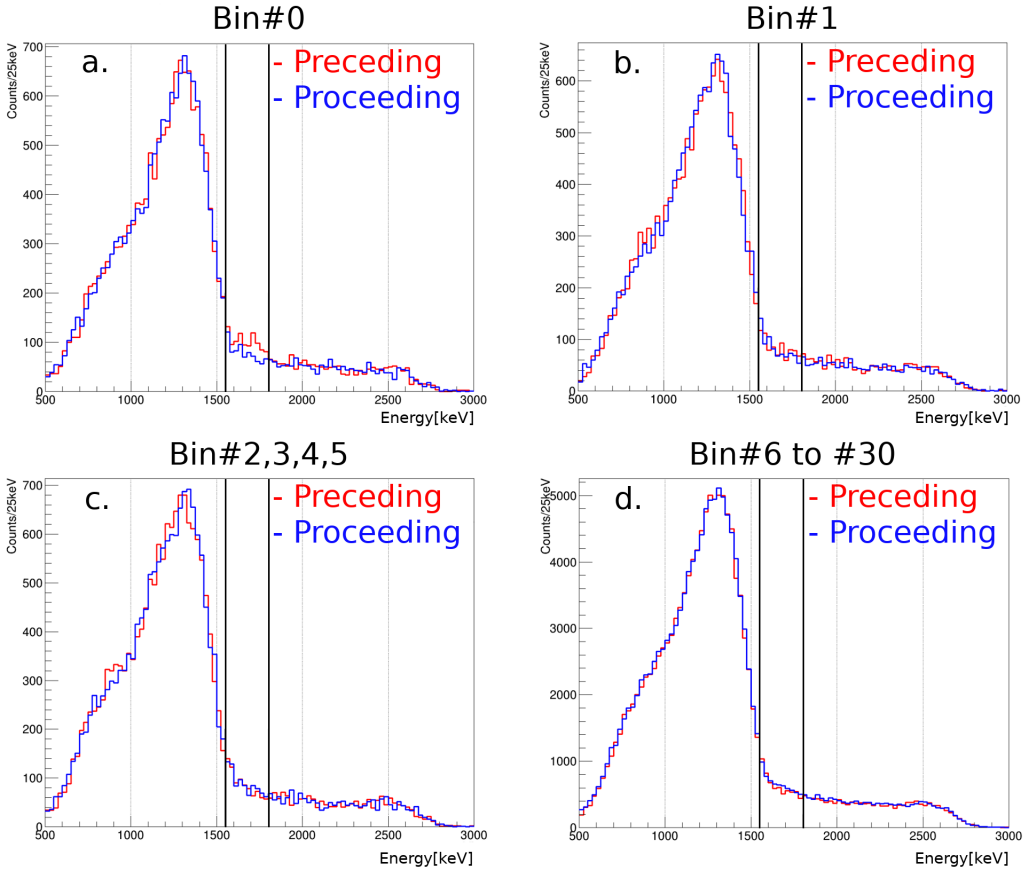


Figure E.1: Count of excess α event from ^{212}Bi observed in various MIMO bins and in non-alpha coincident events. The observation of α events indicate ^{208}Tl background events in the remaining data. The α cut used in this study were not able to identify these excess α events. a.) The comparison spectrums are shown for bin#0 events. Excess number of α events are clearly visible. b.) Comparison spectrums of bin#1 events. c.) Comparison spectrums of bins#2,3,4,5 events. d.) Comparison spectrums of bins#6 to #30.

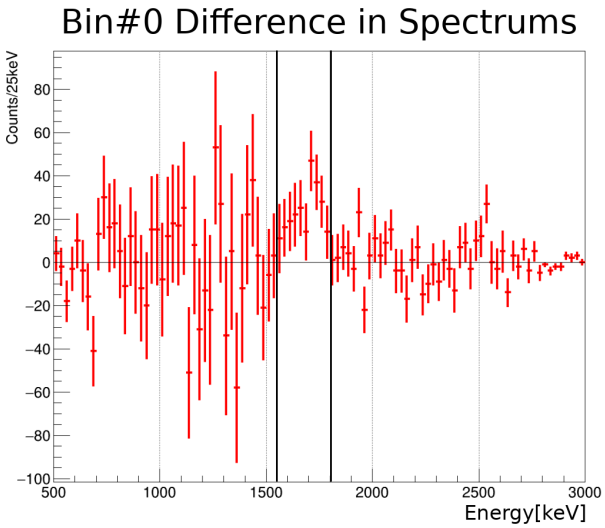


Figure E.2: Difference in preceding and proceeding energy spectrums of events in bin#0. The excess α events can be observed in the $\pm 2\sigma$ energy region of the expected α peak.

Bin	Count
bin#0	240 ± 45.7
bin#1	58 ± 44.1
bin#2,3,4,5	-19.1 ± 45.2
bins#6 to 30	-294 ± 122.6

Table E.1: Counted number of events within the $\pm 2\sigma$ α energy peak in various bins.

count of excess events within this energy range.

Referencing Fig. E.1 and Table E.1, we observe that there are clear excess of α events in bin#0. In bin#1, some excess α events are observed but due to the high uncertainty, the observed count is not reliable. In bin#2 and above, the uncertainty becomes very large and the observed α counts are also not reliable. However, we can conclude that there are no clear observation of excess α counts in bins#2 and above. This result is consistent with SI analysis and α coincidence analysis.

Bibliography

[1] S. R. Elliott, A. A. Hahn, and M. K. Moe, "Direct evidence for two-neutrino double-beta decay in ^{82}Se," *Phys. Rev. Lett.*, vol. 59, pp. 2020–2023, Nov 1987.

[2] H. Ejiri, "Double beta decays and neutrino masses," *Journal of the Physical Society of Japan*, vol. 74, no. 8, pp. 2101–2127, 2005.

[3] J. Vergados, H. Ejiri, and F. Simkovic, "Theory of neutrinoless double beta decay," *Reports on progress in physics. Physical Society (Great Britain)*, vol. 75, p. 106301, 09 2012.

[4] Y. Fukuda, T. Hayakawa, E. Ichihara, K. Inoue, K. Ishihara, H. Ishino, Y. Itow, T. Kajita, *et al.*, "Evidence for oscillation of atmospheric neutrinos," *Phys. Rev. Lett.*, vol. 81, pp. 1562–1567, Aug 1998.

[5] G. Benato, "Effective majorana mass and neutrinoless double beta decay," *The European Physical Journal C*, vol. 75, p. 563, Nov 2015.

[6] M. J. Dolinski, A. W. Poon, and W. Rodejohann, "Neutrinoless double-beta decay: Status and prospects," *Annual Review of Nuclear and Particle Science*, vol. 69, Aug 2019.

[7] T. Iida, T. Kishimoto, M. Nomachi, S. Ajimura, S. Umehara, K. Nakajima, K. Ichimura, S. Yoshida, and k. . others, "The candles experiment for the study of ca-48 double beta decay," *Nuclear and Particle Physics Proceedings*, vol. 273-275, pp. 2633 – 2635, 2016. 37th International Conference on High Energy Physics (ICHEP).

[8] V. I. TRETYAK and Y. G. ZDESENKO, "Tables of double beta decay data—an update," *Atomic Data and Nuclear Data Tables*, vol. 80, no. 1, pp. 83 – 116, 2002.

[9] H. V. Klapdor-Kleingrothaus, A. Dietz, H. L. Harney, and I. V. Krivosheina, "Evidence for neutrinoless double beta decay," *Modern Physics Letters A*, vol. 16, no. 37, pp. 2409–2420, 2001.

[10] M. Agostini, A. M. Bakalyarov, M. Balata, *et al.*, "Improved limit on neutrinoless double-β decay of ^{76}Ge from gerda phase ii," *Phys. Rev. Lett.*, vol. 120, p. 132503, Mar 2018.

[11] S. I. Alvis, I. J. Arnquist, F. T. Avignone, *et al.*, "Search for neutrinoless double-β decay in ^{76}Ge with 26 kg yr of exposure from the majorana demonstrator," *Phys. Rev. C*, vol. 100, p. 025501, Aug 2019.

[12] A. Caminata, D. Adams, C. Alduino, *et al.*, "Results from the cuore experiment †," *Universe*, vol. 5, no. 1, 2019.

[13] J. S. and, "Results and future plans for the KamLAND-zen experiment," *Journal of Physics: Conference Series*, vol. 888, p. 012031, sep 2017.

[14] H. Kakubata, *Study of Backgrounds in CANDLES to Search for Double Beta Decays of 48Ca.* PhD thesis, Osaka University, 2015.

[15] V. Mikhailik, H. Kraus, J. Imber, and D. Wahl, "Scintillation properties of pure caf2," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 566, no. 2, pp. 522 – 525, 2006.

[16] K. Suzuki, S. Ajimura, M. Nomachi, I. Ogawa, and M. Yoshizawa, "New daq system for the candles experiment," in *2014 19th IEEE-NPSS Real Time Conference*, pp. 1–3, May 2014.

[17] T. Maeda, S. Ajimura, W. M. Chan, *et al.*, "The candles trigger system for the study of double beta decay of ^{48}ca," *IEEE Transactions on Nuclear Science*, vol. 62, pp. 1128–1134, June 2015.

[18] K. Nakajima, T. Iida, K. Akutagawa, and k. . others, "Background studies of high energy gamma rays from (n,gamma) reactions in the candles experiment," *Astroparticle Physics*, vol. 100, pp. 54 – 60, 2018.

[19] E. Gatti and F. D. Martini, "A new linear method of discrimination between elementary particles in scintillation counters," *Conference on Nuclear Electronics; Belgrade, Yugoslavia (Serbia)*, pp. 265 – 276, Apr 1962.

[20] F. A. Danevich, A. S. Georgadze, V. V. Kobychev, *et al.*, "α activity of natural tungsten isotopes," *Phys. Rev. C*, vol. 67, p. 014310, Jan 2003.

[21] J. Allison, K. Amako, J. Apostolakis, P. Arce, M. Asai, T. Aso, E. Bagli, *et al.*, "Recent developments in geant4," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 835, pp. 186 – 225, 2016.

[22] J. Apostolakis, G. Folger, V. Grichine, A. Howard, V. Ivanchenko, M. Kosov, A. Ribon, V. Uzhinsky, and D. H. Wright, "Geant4 physics lists for hep," in *2008 IEEE Nuclear Science Symposium Conference Record*, pp. 833–836, Oct 2008.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[24] F. Chollet, *Deep Learning with Python*. Greenwich, CT, USA: Manning Publications Co., 1st ed., 2017.

[25] G. A. Marcoulides, "Book review of the elements of statistical learning: Data mining, inference and prediction, by trevor hastie, robert tibshirani, and jerome friedman; and generalizability theory, by robert l. brennan," *Structural Equation Modeling: A Multidisciplinary Journal*, vol. 11, no. 1, pp. 150–152, 2004.

[26] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[27] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014.

[28] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," 2013.

[29] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303–314, Dec 1989.

[30] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, pp. 251–257, Mar. 1991.

[31] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," 2017.

[32] D. Mishkin, N. Sergievskiy, and J. Matas, "Systematic evaluation of cnn advances on the imagenet," *CoRR*, vol. abs/1606.02228, 2016.

[33] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 1319–1327, PMLR, 17–19 Jun 2013.

[34] S. Ruder, "An overview of gradient descent optimization algorithms." `https://arxiv.org/pdf/1609.04747.pdf`, 2017.

[35] C. Kong and S. Lucey, "Take it in your stride: Do we need striding in cnns?," 2017.

[36] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, June 2015.

[37] F. Chollet, "keras." `https://github.com/fchollet/keras`, 2015.

[38] M. Abadi, P. Barham, J. Chen, Z. Chen, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.

[39] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with cuda," *Queue*, vol. 6, pp. 40–53, Mar. 2008.

[40] D. D. Lee, U. von Luxburg, R. Garnett, M. Sugiyama, and I. Guyon, eds., *NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems*, (USA), Curran Associates Inc., 2016.

[41] K. Nakajima *et al.*, "Performance of updated shielding system in CANDLES," *AIP Conf. Proc.*, vol. 1921, no. 1, p. 060003, 2018.

[42] T. Ohata, *Search for Neutrinoless Double Beta Decay in 48Ca with the CANDLES III experiment*. PhD thesis, Osaka University, 2018.

[43] J. E. Gaiser, *Charmonium Spectroscopy From Radiative Decays of the J/ψ and ψ'*. PhD thesis, SLAC, 1982.

[44] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, pp. 5595–5637, Jan. 2017.

List of Figures

List of Tables