COLLEGE DOCTORAL PAYS DE LA LOIRE
ED MATIERE, MOLECULES MATERIAUX ET GEOSCIENCES

Nantes Université

1

2

THÈSE DE DOCTORAT DE

NANTES UNIVERSITÉ

ÉCOLE DOCTORALE N° 596
Matière, Molécules, Matériaux
Spécialité : *Physique Subatomique et Instrumentation Nucléaire*

Par
Léonard Imbert

Deep learning methods and Dual Calorimetric analysis for high precision neutrino oscillation measurements at JUNO

Thèse présentée et soutenue à Nantes, le 2 Decembre 2024
Unité de recherche : Laboratoire SUBATECH, UMR 6457

Rapporteurs avant soutenance :

Christine Marquet	Directrice de recherche au CNRS, LP2I Bordeaux
David Rousseau	Directeur de recherche au CNRS, IJCLab

Composition du Jury :

Président :	Barbara Erazmus	Directrice de recherche au CNRS, Subatech
Examinateurs :	Juan Pedro Ochoa-Ricoux	Full Professor, Universty of California, Irvine
	Yasmine Amhis	Directrice de recherche au CNRS, IJCLab
	Christine Marquet	Directrice de recherche au CNRS, LP2I Bordeaux
	David Rousseau	Directeur de recherche au CNRS, IJCLab
Dir. de thèse :	Frédéric Yermia	Professeur des universités, Nantes Université
Co-dir. de thèse :	Benoit Viaud	Chargé de recherche au CNRS, Subatech

Contents

Remerciements

Introduction

The Standard Model of particle physics (SM) has been remarkably successful at accounting for, or predicting experimental observations in the laboratory. However, it is the subject of several limitations. For instance, it provides a mechanism to explain the existence of mass but can't predict the peculiar pattern followed by fermion masses. The same applies to CP violation. The SM predicts its existence but not the amplitude necessary to explain the baryonic asymmetry of the Universe. For such reasons, one can assume the SM is the manifestation of a more fundamental physics, Beyond the Standard Model (BSM).

Neutrino physics is a window on BSM. Indeed, the mass of known neutrinos is at least 5 order of magnitudes below that of the lightest fermion, which further deepens the issue of fermion mass generation. Some solutions have implication on the nature of neutrinos – dirac or majorana fermions ? – which one of the big unknowns in this domain. Additional neutrinos beyond the three presently known shall also be considered. The way neutrinos mix flavor to make neutrino oscillation possible is also unexplained. This is one of the tasks of BSM models to answer such questions. Before that, a good part of the World experimental program in the 10 coming years is to complete the exploration of 3-neutrino physics by answering mainly two questions : does CP violation exist the lepton system ? What is the Neutrino Mass ordering (NMO) ? An introduction to neutrino physics will be given in Chapter 1.

The Jiangmen Underground Neutrino Observatory (JUNO), currently under construction in China, aims to address these questions, particularly the determination of the NMO. JUNO's approach is to study reactor antineutrinos emitted from nearby nuclear power plants. By precisely measuring the energy spectrum of these antineutrinos after oscillation, JUNO seeks to detect the subtle interference patterns in the spectrum that are sensitive to the NMO. The ability to achieve this requires unprecedented precision in both the energy resolution and the calibration of the detector's response to neutrino events. JUNO is expected to start data collection in 2025, with the goal of determining the NMO at a significance level of 3-4σ after six years of data taking. At the heart of JUNO's experimental design is its dual calorimetry system, comprising two separate sets of photomultipliers—large (LPMT) and small (SPMT) PMTs—that allow for independent energy measurements of the same events. This dual system is not only essential for improving energy resolution but also for providing cross-checks that ensure systematic uncertainties are well-understood and minimized. Achieving JUNO's goals depends on this dual calorimetry system, as it will enable precise reconstruction of the energy spectrum and the identification of potential discrepancies between the two systems.

Another emerging area of importance in particle physics experiments is the application of machine learning (ML) techniques. Over the past decade, ML methods, particularly deep learning, have been increasingly used to tackle complex problems in event classification, reconstruction, and even data generation like the High luminosity LHC Upgraded experiments. Performant online reconstruction, critical for the trigger systems of such experiments, is another example. The complexity of the data and the required precision in experiments such as JUNO make ML an attractive tool. In particular, Neural Networks (NNs) and other advanced ML models have shown potential for improving the accuracy of energy reconstruction and other key analysis tasks. However, for the results obtained

using ML methods to be trusted by the scientific community, the reliability of these methods must be rigorously demonstrated. An introduction to ML, and in particular Neural Network (NN) is given in Chapter 3.

This thesis was performed in the framework of the Neutrino group at Subatech, since October 2021. The exploratory works reported in this manuscript addresses the subjects mentioned above, in the particular context of the measurement by JUNO of the reactor antineutrino oscillation to determine the NMO. Before the start of this thesis, several ML energy reconstruction algorithms – Boosted Decision Trees (BDT), Fully Connected Neural Networks (FCNN), Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs) – had already been developed within the collaboration. Their performance seems to match that of the classical algorithm but not to do convincingly better. We have explored a possibility to do better by developing a GNN with an innovative architecture tailored to the JUNO experiment. Before that, we developed a CNN for the reconstruction of the anti-neutrino energy using only JUNO's small PMTs system. This CNN is useful in particular in Chapter 7 as there is official SPMT only reconstruction in the collaboration yet. These algorithms are described in Chapters 4 and 5.

We have been the first in JUNO to address the issue of ML reliability. We explore in this thesis the feasibility of an Adversarial Neural Network (ANN) to generate (and therefore identify) scenarios of discrepancies between raw data in the real detector and in the detector's simulation. The focus here is on discrepancies that could alter JUNO's results on NMO, but are too subtle to be detected via usual data/MC comparisons in control samples. This is presented in Chapter 6.

We have already mentioned earlier it is crucial for JUNO to understand its energy scale with a good precision. This is the raison d'être of the existence of two calorimetric readout systems : the large (LPMT) and small (SPMT) photomultipliers systems. It allows Dual Calorimetry techniques to constrain our understanding of the reconstruction. The last subject of this thesis explores for the first time one of them : the Dual Calorimetry with neutrino oscillation, which leverages potential discrepancies between the oscillation analyses performed with each system. Our work on this is described in Chapter 7. It was also the occasion of technical developments on the analysis framework used at Subatech. These improvements will be very useful for future analyses of the group, beyond Dual calorimetry.

Chapter 1

Neutrino physics

The neutrino, or ν for the close friends, a fascinating and invisible particle. Some will say that dark matter also have those property but at least we are pretty confident that neutrinos exists.

Contents

1.1 Standard model

1.1.1 Limits of the standard model

1.2 Historic of the neutrino

First theories

Discovery

Milestones and anomalies

1.3 Oscillation

1.3.1 Phenomologies

1.4 Open questions

Decrire le m
Regarder th
Kochebina
Limite du r
Interessant,
les neutrino
CP ? Pb des

Chapter 2

The JUNO experiment

"Ave Juno, rosae rosam, et spiritus rex". It means nothing but I found it in tone.

Contents

The first idea of a medium baseline (∼52 km) experiment, was explored in 2008 [1] where it was demonstrated that the Neutrino Mass Ordering (NMO) could be determined by a medium baseline experiment if $\sin^2(2\theta_{13}) > 0.005$ without the requirements of accurate knowledge of the reactor antineutrino spectra and the value of Δm^2_{32}. From this idea is born the Jiangmen Underground Neutrino Observatory (JUNO) experiment.

JUNO is a neutrino detection experiment under construction located in China, in Guangdong proving, near the city of Kaiping. Its main objectives are the determination of the mass ordering at the 3-4σ level in 6 years of data taking and the measurement at the sub-percent precision of the oscillation parameters Δm^2_{21}, $\sin^2\theta_{12}$, Δm^2_{32} and with less precision $\sin^2\theta_{13}$[2].

FIGURE 2.1 – **On the left:** Location of the JUNO experiment and its reactor sources in southern china. **On the right:** Aerial view of the experimental site

For this JUNO will measure the electronic anti-neutrinos ($\bar{\nu}_e$) flux coming from the nuclear reactors of Taishan, Yangjiang, for a total power of 26.6 GW_{th}, and the Daya Bay power plant to a lesser extent. All of those cores are the second-generation pressurized water reactors CPR1000, which is a derivative of Framatome M310. Details about the power plants characteristics and their expected flux of $\bar{\nu}_e$ can be found in the table 2.1. The distance of 53 km has been specifically chosen to maximize the disappearance probability of the $\bar{\nu}_e$. The data taking is scheduled to start early 2025.

2.1 Reactor Neutrinos physics in JUNO

JUNO will try to determine the NMO and to bring at the few per mille level our knowledge of Δm^2_{31}, Δm^2_{21} and $\sin^2(2\theta_{12})$ via the precision analysis of the spectrum of the visible energy left by reactor antineutrinos in its detector.

2.1.1 Antineutrino spectrum measured in JUNO

To some extent, this analysis is equivalent to to extracting from this spectrum the oscillation probability [2] :

$$P(\bar{\nu}_e \to \bar{\nu}_e) = 1 - \sin^2 2\theta_{12} c_{13}^4 \sin^2 \frac{\Delta m^2_{21} L}{4E} - \sin^2 2\theta_{13} \left[c_{12}^2 \sin^2 \frac{\Delta m^2_{31} L}{4E} + s_{12}^2 \sin^2 \frac{\Delta m^2_{32} L}{4E} \right]$$

Where $s_{ij} = \sin\theta_{ij}$, $c_{ij} = \cos\theta_{ij}$, E is the $\bar{\nu}_e$ energy and L is the baseline. We can see the sensitivity to the NMO in the dependency to Δm^2_{32} and Δm^2_{31} causing a phase shift of the spectrum as we can see in the Figure 2.2.

In practice, a fit to the grey distribution of Figure 2.3 will be performed. It is the sum of two components :signal (black) and bacgrounds (colored). Reactor antineutrinos are detected by JUNO via Inverse Beta Decays (IBD) : $\bar{n}u_e + p - > e^+ + n$. The energy spectrum under investigation is therefore that of the reconstructed e^+ visible energy. The black signal spectrum is therefore the sum of the antineutrino differential fluxes from all reactors and reaching the detecteur, weighted by the oscillation probability of Eq 2.1.1 and the IBD differential cross section and convoluted with detection effects. These various ingredients are theoretically modelled in order to provide the probability density function (PDF) to be used in the fit.

To reach JUNO's goals, it takes that this experimental spectrum still bears sizeable traces of the very small phase shift mentioned above. Most notably, the following requirements must be fulfilled :

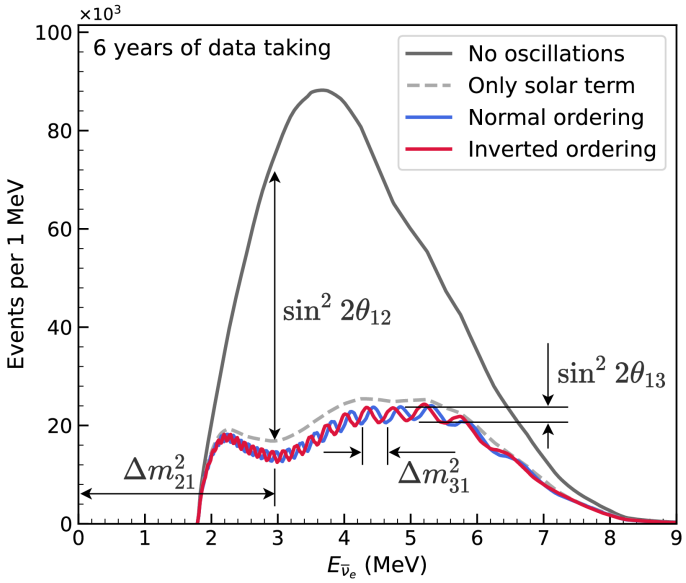


FIGURE 2.2 – Expected number of neutrinos event per MeV in JUNO after 6 years of data taking. The black curve shows the flux if there was no oscillation. The light gray curve shows the oscillation if only the solar terms are taken in account (θ_{12}, Δm^2_{21}). The blue and red curve shows the spectrum in the case of, respectively, NO and IO. The dependency of the oscillation to the different parameters are schematized by the double sided arrows. We can see the NMO sensitivity by looking at the fine phase shift between the red and the blue curve.

1. An energy resolution of $3\%/\sqrt{E(\text{MeV})}$ to be able to distinguish the fine structure of the fast oscillation.
2. An energy scale known at the better than the 1% level.
3. A baseline between 40 and 65 km to maximise the $\bar{\nu}_e$ oscillation probability. The optimal baseline would be 58 km and JUNO baseline is 53 km.
4. At least $\approx$ 100,000 events.This is the necessary statistics to reach JUNO's canonical sensitivity after 6 years of data taking.

$\bar{\nu}_e$ flux coming from nuclear power plants

To get such high measurements precision, it is necessary to have a very good understanding of the sources characteristics. For its NMO and precise measurement studies, JUNO will observe the energy spectrum of neutrinos coming from the nuclear power plants Taishan and Yangjiang's cores, located at 53 km of the detector to maximise the disappearance probability of the $\bar{\nu}_e$.

The $\bar{\nu}_e$ coming from reactors are emitted from β-decay of unstable fission fragments. The Taishan and Yangjiang reactors are Pressurised Water Reactor (PWR), the same type as Daya Bay. In those type of reactor more the 99.7 % and $\bar{\nu}_e$ are produced by the fissions of four fuel isotopes ^{235}U, ^{238}U, ^{239}Pu and ^{241}Pu. The neutrino flux per fission of each isotope is determined by the inversion of the measured β spectra of fission product [4–8] or by calculation using the nuclear databases [9, 10].

The neutrino flux coming from a reactor at a time t can be predicted using

$$\phi(E_\nu, t)_r = \frac{W_{th}(t)}{\sum_i f_i(t) e_i} \sum_i f_i(t) S_i(E_\nu) \tag{2.1}$$

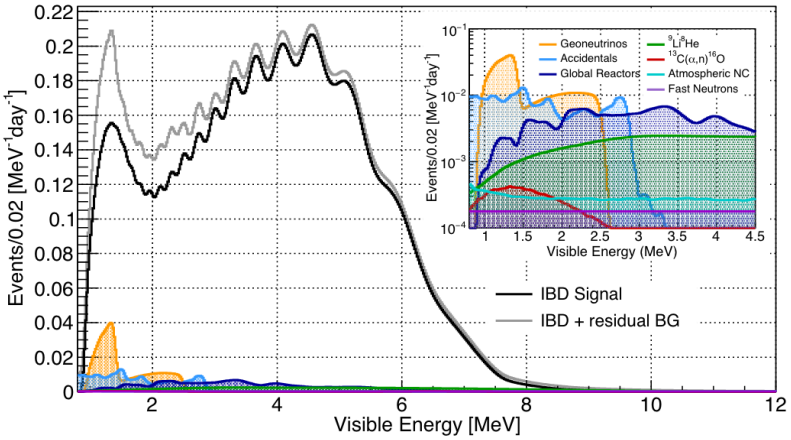


FIGURE 2.3 – Expected visible energy spectrum measured with the LPMT system with (grey) and without (black) backgrounds. The background amount for about 7% of the IBD candidate and are mostly localized below 3 MeV [3]

Reactor	Power (GW_{th})	Baseline (km)
Taishan	9.2	52.71
Core 1	4.6	52.77
Core 2	4.6	52.64
Yangjiang	17.4	52.46
Core 1	2.9	52.74
Core 2	2.9	52.82
Core 3	2.9	52.41
Core 4	2.9	52.49
Core 5	2.9	52.11
Core 6	2.9	52.19
Daya Bay	17.4	215
Huizhou	17.4	265

TABLE 2.1 – Characteristics of the nuclear power plants observed by JUNO.

where $W_{th}(t)$ is the thermal power of the reactor, $f_i(t)$ is the fraction fission of the ith isotope, e_i its thermal energy released in each fission and $S_i(e_\nu)$ the neutrino flux per fission for this isotope.

The latter flux is difficult to predict. To evaluate JUNO's sensitivity and to serve as a starting point in the spectrum PDF, the Huber-Mueller model is used [5], corrected using Daya Bays data [11] to account for a ∼5% deficit with respect to models, referred to as the reactor antineutrino anomaly [12], and for a discrepancy between models and data in the spectral shape (the so call 5 MeV bump).

In addition to those prediction, a satellite experiment named TAO[13] will be setup near the reactor core Taishan-1 to measure with an energy resolution of 2% at 1 MeV the neutrino flux coming from the core, more details can be found in Section 2.5.1. It will help identifying unknown fine structure and give more insight on the $\bar{\nu}_e$ flux coming from this reactor.

2.1.2 Background spectra

Considering the close reactor neutrinos flux as the main signal, the signals that are considered as background are:

— The geoneutrinos producing background in the 0.511 ∼ 2.7 MeV region.

— The neutrinos coming from the other nuclear reactors around Earth.

In addition to all those physics signal, non-neutrinos signal that would mimic an IBD will also be present. It is composed of:

— The signal coming from radioactive decay (α, γ, β) from natural radioactive isotopes in the material of the detector.

— Cosmogenic event such as fast neutrons and activated isotopes induced by muons passing through the detector, most notably the spallation on ^{12}C.

All those events represent a non-negligeable part of the spectrum as shown in Figure 2.3.

2.2 Other physics

While the design of JUNO is tailored to measure $\bar{\nu}_e$ coming from nuclear reactor, JUNO will be able to detect neutrinos coming from other sources thus allowing for a wide range of physics studies as detailed in the table 2.2 and in the following sub-sections.

Research	Expected signal	Energy region	Major backgrounds
Reactor antineutrino	60 IBDs/day	0–12 MeV	Radioactivity, cosmic muon
Supernova burst	5000 IBDs at 10 kpc 2300 elastic scattering	0–80 MeV	Negligible
DSNB (w/o PSD)	2–4 IBDs/year	10–40 MeV	Atmospheric ν
Solar neutrino	hundreds per year for ^{8}B	0–16 MeV	Radioactivity
Atmospheric neutrino	hundreds per year	0.1–100 GeV	Negligible
Geoneutrino	$\approx$ 400 per year	0–3 MeV	Reactor ν

TABLE 2.2 – Detectable neutrino signal in JUNO and the expected signal rates and major background sources

Geoneutrinos

Geoneutrinos designate the antineutrinos coming from the decay of long-lived radioactive elements inside the Earth. The 1.8 MeV threshold necessary for the IBD makes it possible to measure geoneutrinos from ^{238}U and ^{232}Th decay chains. The studies of geoneutrinos can help refine the Earth crust models but is also necessary to characterise their signal, as they are a background to the mass ordering and oscillations parameters studies.

Atmospheric neutrinos

Atmospheric neutrinos are neutrinos originating from the decay of π and K particles that are produced in extensive air showers initiated by the interactions of cosmic rays with the Earth atmosphere. Earth is mostly transparent to neutrinos below the PeV energy, thus JUNO will be able to see neutrinos coming from all directions. Their baseline range is large (15km $\sim$ 13000km), they can have energy between 0.1 GeV and 10 TeV and will contain all neutrino and antineutrinos flavour. Their studies is complementary to the reactor antineutrinos and can help refine the constraints on the NMO [2].

Supernovae burst neutrinos

Neutrinos are crucial component during all stages of stellar collapse and explosion. Detection of neutrinos coming for core collapse supernovae will provide us important informations on the mech-

anisms at play in those events. Thanks to its 20 kt sensible volume, JUNO has excellent capabilities to detect all flavour of the $\mathcal{O}$(10 MeV) postshock neutrinos, and using neutrinos of the $\mathcal{O}$(1 MeV) will give informations about the pre-supernovae neutrinos. All those informations will allow to disentangle between the multiple hydro-dynamic models that are currently used to describe the different stage of core-collapse supernovae.

Diffuse supernovae neutrinos background

Core-collapse supernovae in our galaxy are rare events, but they frequently occur throughout the visible Universe sending burst of neutrinos in direction of the Earth. All those events contributes to a low background flux of low-energy neutrinos called the Diffuse Supernovae Neutrino Background (DSNB). Its flux and spectrum contains informations about the red-shift dependent supernovae rate, the average supernovae neutrino energy and the fraction of black-hole formation in core-collapse supernovae. Depending of the DSNB model, we can expect 2-4 IBD events per year in the energy range above the reactor $\bar{\nu}_e$ signal, which is competitive with the current Super-Kamiokande+Gadolinium phase [14].

Beyond standard model neutrinos interactions

JUNO will also be able to probe for beyond standard model neutrinos interactions. After the main physics topics have been accomplished, JUNO could be upgraded to probe for neutrinoless beta decay ($0\nu\beta\beta$). The detection of such event would give critical informations about the nature of neutrinos, is it a majorana or a dirac particle. JUNO will also be able to probe for neutrinos that would come for the decay or annihilation of Dark Matter inside the sun and neutrinos from putative primordial black hole. Through the unitary test of the mixing matrix, JUNO will be able to search for light sterile neutrinos. Thanks to JUNO sensitivity, multiple other exotic research can be performed on neutrino related beyond standard model interactions.

Proton decay

Proton decay is a potential unobserved event where the proton decay by violating the baryon number. This violation is necessary to explain the baryon asymmetry in the universe and is predicted by multiple Grand Unified Theories which unify the strong, weak and electromagnetic interactions. Thanks to its large active volume, JUNO will be able to take measurement of the potential proton decay channel $p \rightarrow \bar{\nu}K^+$ [15] thanks to the timing resolution of the SPMT system. Studies show that JUNO should be competitive with the current best limit at 5.9×10^{33} years from Super-K. This studies show that JUNO, considering no proton decay events observed, would be be able to rules a limit of 9.6×10^{33} years at 90 % C.L.

2.3 The JUNO detector

The JUNO detector is a scintillator detector buried 693.35 meters under the ground (1800 meters water equivalent). It consist of Central Detector (CD), a water pool and a Top Tracker (TT) as showed in Figure 2.4a. The CD is an acrylic vessel containing the 20 ktons of Liquid Scintillator (LS). It is supported by a stainless steel structure and is immersed in that water pool that is used as shielding from external radiation and as a cherenkov detector for the background. The top of the experiment is partially covered by the Top Tracker (TT), a plastic scintillator detector which is use to detect the atmospheric muons background and is acting as a veto detector.

The top of the experiment also host the LS purification system, a water purification system, a ventilation system to get rid of the potential radon in the air. The CD is observed by two system of Photo-Multipliers Tubes (PMT). They are attached to the steel structure and their electronic readout is submersed near them. A third system of PMT is also installed on the structure but are facing outward of the CD, instrumenting the water to be cherenkov detector. The CD and the cherenkov detector are optically separated by Tyvek sheet. A chimney for LS filling and purification and for calibration operations connects the CD to the experimental hall from the top.

The CD has been dimensioned to meet the requirements presented in Section 2.1.1:

— Its 20 ktons monolithic LS provide a volume sizeable enough, in combination with the expected $\bar{\nu}_e$ flux, to reach the desired statistic in 6 years. Its monolithic nature also allow for a full containment of most of the events, preventing the energy loss in non-instrumented parts that would arise from a segmented detector.

— Its large overburden shield it from most of the atmospheric background that would pollute the signal.

— The localization of the experiment, chosen to maximize the disappearance with a 53km baseline and in a region that allow two nuclear power plant to be used as sources.

(A) Schematics view of the JUNO detector.

(B) Top down view of the JUNO detector under construction

FIGURE 2.4

This section cover in details the different components of the detector and the detection systems.

2.3.1 Detection principle

The CD will detect the neutrino and measure their energy mainly via an Inverse Beta Decay (IBD) interaction with proton mainly from the ^{12}C and H nucleus in the LS:

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

Kinematics calculation shows that this interaction has an energy threshold for the $\bar{\nu}_e$ of $(m_n + m_e - m_p) \approx 1.806$ MeV [16]. This threshold make the experiment blind to very low energy neutrinos. The residual energy $E_\nu - 1.806$ MeV is be distributed as kinetic energy between the positron and the neutron. The energy of the emitted positron E_e is given by [16]

$$E_e = \frac{(E_\nu - \delta)(1 + \epsilon_\nu) + \epsilon_\nu \cos\theta \sqrt{(E_\nu - \delta)^2 + \kappa m_e^2}}{\kappa} \tag{2.2}$$

where $\kappa = (1+\epsilon_\nu)^2 - \epsilon_\nu^2 \cos^2\theta \approx 1$, $\epsilon_\nu = \frac{E_\nu}{m_p} \ll 1$ and $\delta = \frac{m_n^2 - m_p^2 - m_e^2}{2m_p} \ll 1$. We can see from this equation that the positron energy is strongly correlated to the neutrino energy.

The positron and the neutron will then propagate in the detection medium, the Liquid Scintillator (LS), loosing their kinetic energy by exciting the molecule of the LS (more details in Section 2.3.2). Once stopped, the positron will annihilate with an electron from the medium producing two 511 KeV gamma. Those gamma will themselves interact with the LS, exciting it before being absorbed by photoelectrical effect. The neutron will be captured by an hydrogen, emitting a 2.2 MeV gamma in the process. This gamma will also deposit its energy before being absorbed by the LS.

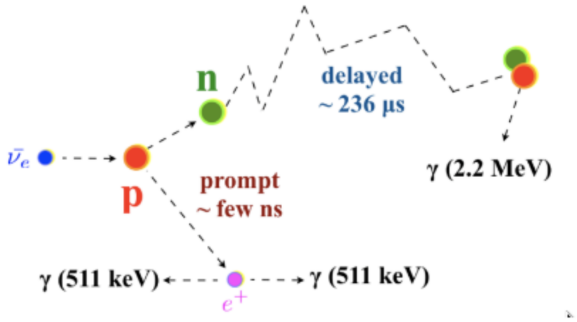


FIGURE 2.5 – Schematics of an IBD interaction in the central detector of JUNO

The scintillation photons have frequency in the UV and will propagate in the LS, being re-absorbed and re-emitted by compton effect before finally be captured by PMTs instrumenting the acrylic sphere. The analog signal of the PMTs digitized by the electronic is the signal of our experiment. The signal produced by the positron is subsequently called the prompt signal, and the signal coming from the neutron the delayed signal. This naming convention come from the fact that the positron will deposit its energy rather quickly (few ns) where the neutron will take a bit more time ($\sim$ 236 μs).

2.3.2 Central Detector (CD)

The central detector, composed of 20 ktons of Liquid Scintillator (LS), is the main part of JUNO. The LS is contained in a spherical acrylic vessel supported by a stainless steel structure. The CD and its structural support are submerged in a cylindrical water pool of 43.5m diameter and 44m height. We're confident that the water pool provide sufficient buffer protection in every direction against the rock radioactivity.

Acrylic vessel

The acrylic vessel is a spherical vessel of inner diameter of 35.4 m and a thickness of 120 mm. It is assembled from 265 acrylic panels, thermo bonded together. The acrylic recipes has been carefully tuned with extensive R&D to ensure it does not include plasticizer and anti-UV material that would stop the scintillation photons. Those panels requires to be pure of radioactive materials to not cause background. Current setup where the acrylic panels are molded in cleanrooms of class 10000, let us reach a uranium and thorium contamination of <0.5 ppt. The molding and thermoforming processes is optimized to increase the assemblage transparency in water to >96%. The acrylic vessel is supported by a stainless steel structure via supporting node (fig 2.6). The structure and the nodes are designed to be resilient to natural catastrophic events such as earthquake and can support many times the effective load of the acrylic vessel.

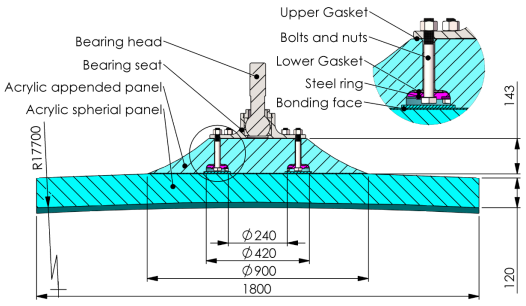


FIGURE 2.6 – Schematics of the supporting node for the acrylic vessel

Liquid scintillator

The Liquid Scintillator (LS) has a similar recipe as the one used in Daya Bay [17] but without gadolinium doping. It is made of three components, necessary to shift the wavelength of emitted photons to prevent their reabsorption and to shift their wavelength to the PMT sensitivity region as illustrated in Figure 2.7:

1. The detection medium, the *linear alkylbenzene* (LAB). Selected because of its excellent transparency, high flash point, low chemical reactivity and good light yield. Accounting for $\sim$ 98% of the LS, it is the main component with which ionizing particles and gamma interact. Charged particles will collide with its electronic cloud transferring energy to the molecules, gamma will interact via compton effect with the electronic cloud before finally be absorbed via photoelectric effect.
2. The second component of the LS is the *2,5-diphenyloxazole* (PPO). A fraction of the excitation energy of the LAB is transferred to the PPO, mainly via non radiative process [18]. The PPO molecules de-excites in the same way, transferring their energy to the bis-MSB. The PPO makes for 1.5 % of the LS.
3. The last component is the *p-bis(o-methylstyryl)-benzene* (bis-MSB). Once excited by the PPO, it will emit photon with an average wavelength of $\sim$ 430 nm (full spectrum in Figure 2.7) that can thus be detected by our photo-multipliers systems. It amount for $\sim$ 0.5% of the LS.

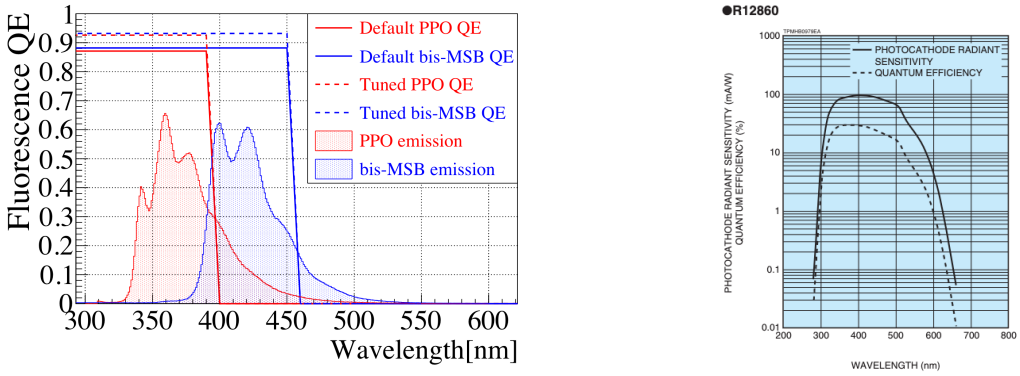


FIGURE 2.7 – **On the left:** Quantum efficiency (QE) and emission spectrum of the LAB and the bis-MSB [17]. **On the right:** Sensitivity of the Hamamatsu LPMT depending on the wavelength of the incident photons [19].

This formula has been optimized using dedicated studies with a Daya Bay detector [17, 20] to reach the requirements for the JUNO experiment:

— A light yield / MeV of the amount of 10^4 photons to maximize the statistic in the energy measurement.

— An attenuation length comparable to the size of the detector to prevent losing photons during their propagation in the LS. The final attenuation length is 25.8m [21] to compare with the CD diameter of 35.4m.
— Uranium/Thorium radiopurity to prevent background signal. The reactor neutrino program require a contamination fraction $F < 10^{-15}$ while the solar neutrino program require $F < 10^{-17}$.

The LS will frequently be purified and tested in the Online Scintillator Internal Radioactivity Investigation System (OSIRIS) [22] to ensure that the requirements are kept during the lifetime of the experiment, more details to be found in Section 2.5.2.

Large Photo-Multipliers Tubes (LPMTs)

The scintillation light produced by the LS is then collected by Photo-Multipliers Tubes (PMT) that transform the incoming photon into an electric signal. As described in Figure 2.8, the incident photons interact with the photocatode via photoelectric effect producing an electron called a Photo-Electron (PE). This PE is then focused on the dynodes where the high voltage will allow it to be multiplied. After multiple amplification the resulting charge - in coulomb [C] - is collected by the anode and the resulting electric signal can be digitalized by the readout electronics from which the charge and timing can be extracted.

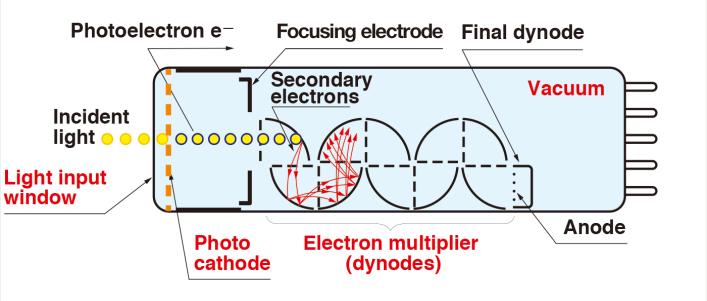


FIGURE 2.8 – Schematic of a PMT

The Large Photo-Multipliers Tubes (LPMT), used in the central detector and in the water pool, are 20-inch (50.8 cm) radius PMTs. $\sim$ 5000 dynode-PMTs [19] were produced by the Hamamatsu© company and $\sim$ 15000 Micro-Channel Plate (MCP) [23] by the NNVT© company. This system is the one responsible for the energy measurement with a energy resolution of $3\%/\sqrt{E}$, resolution necessary for the mass ordering measurement. To reach this precision, the system is composed of 17612 PMTs quasi uniformly distributed over the detector for a coverage of 75.2% reaching $\sim$ 1800 PE/MeV or $\sim$ 2.3% resolution due to statistic, leaving $\sim$ 0.7% for the systematic uncertainties. They are located outside the acrylic sphere in the water pool facing the center of the detector. To maintain the resolution over the lifetime of the experiment, JUNO require a failure rate < 1% over 6 years.

The LPMTs electronic are divided in two parts. One "near", located underwater, in proximity of the LPMT to reduce the cable length between the PMT and early electronic. A second one, outside of the detector that is responsible for higher level analysis before sending the data to the DAQ.

The light yield per MeV induce that a LPMT can collect between 1 and 1000 PE per event, a wide dynamic range, causing non linearity in the PMT response that need to be understood and calibrated, see Section 2.4 for more details.

Before performng analysis, the analog readout of the LPMT need to be amplified, digitised and packaged by the readout electronics schematized in Figure 2.9. This electronic is splitted in two parts: *wet* electronic that are located near the LPMTs, protected in an Underwater Box (UWB) and the *dry* electronics located in deicated rooms outside of the water pool.

The LPMTs are connected to the UWB by groups of three. Each UWB contains:

— Three high voltage units, each one powering a PMT.

— A global control unit, responsible for the digitization of the waveform, composed of six analog-digital units that produce digitized waveform and a Field Programmable Gate Array (FPGA) that complete the waveform with metadatas such as the local timestamp trigger, etc... Ths FPGA also act as a data buffer when needed by the DAQ and trigger system.

— Additional memory in order to temporally store the data in case of sudden burst of the input rate (such as in the case of nearby supernovae).

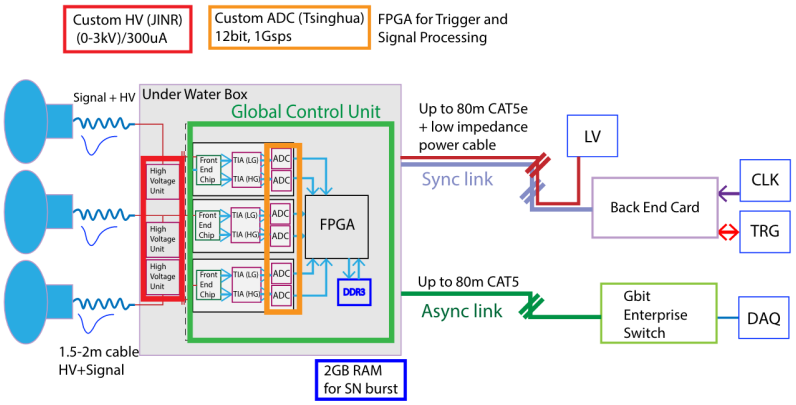


FIGURE 2.9 – The LPMT electronics scheme. It is composed of two part, the *wet* electronics on the left, located underwater and the *dry* electronics on the right. They are connected by Ethernet cable for data transmission and a dedicated low impedance cable for power distribution

The *dry* electronic synchronize the signals from the UWBs abd centralise the information of the CD LPMTs. It act as the Global Trigger by sending the UWB data to DAQ in the case if the LPMT multiplicity condition is fulfilled.

Small Photo-Multipliers Tubes (SPMTs)

The Small PMT (SPMTs) system is made of 3-inch (7.62 cm) PMTs. They will be used in the CD as a secondary detection system. Those 25600 SPMTs will observe the same events as the LPMTs, thus sharing the physics and detector systematics up until the photon conversion. With a detector coverage of 2.7%, this system will collect $\sim$ 43 PE/MeV for a final energy resolution of $\sim$ 17%. This resolution is not enough to measure the NMO, θ_{13}, Δm^2_{31} but will be sufficient to independently measure θ_{12} and Δm^2_{21}.

The benefit of this second system is to be able to perform another, independent measure of the same events as the LPMTs, constituing the Dual Calorimetry useful for calibrationa and, as it we will explore in this thesis, for physics analysis. Due to the low PE rate, SPMTs will be running in photo-counting mode in the reactor range and thus will be insensitive to LPMT intrisic effect (see Section 2.4). Using this property, the intrinsic charge non linearity of the LPMTs can be measured by comparing the PE count in the SPMTs and LPMTs [24]. Also, due to their smaller size and electronics, SPMTs have a better timing resolutions than the LPMTs. At higher energy range, like supernovae events, LPMTs will saturate where SPMTs due to their lower PE collection will to produce a reliable measure of the energy spectrum.

The SPMTs will be grouped by pack of 128 to an UWB hosting their electronics as illustrated in Figure 2.10. This underwater box host two high voltage splitter boards, each one supplying 64 SPMTs, an

ASIC Battery Card (ABC) and a global control unit.

The ABC board will readout and digitize the charge and time of the 128 SPMTs signals and a FPGA will joint the different metadata. The global control unit will handle the powering and control of the board and will be in charge of the transmition of the data to the DAQ.

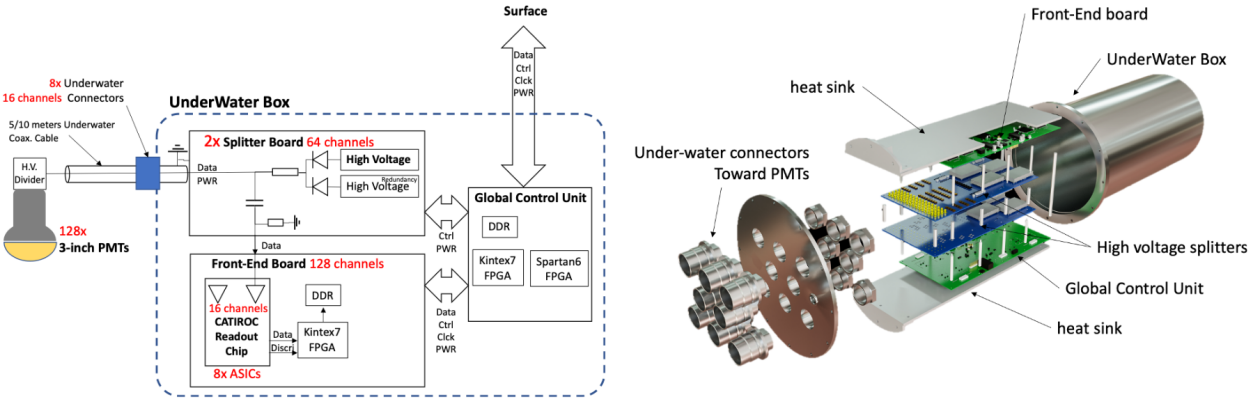


FIGURE 2.10 – Schematic of the JUNO SPMT electronic system (**left**), and exploded view of the main component of the UWB (**right**)

2.3.3 Veto detector

The CD will be bathed in constant background noise coming from numerous sources : the radioactivity from surrounding rock and its own components or from the flux of cosmic muons. This background needs to be rejected to ensure the purity of the IBD spectrum. To prevent a big part of them, JUNO use two veto detector that will tag events as background before CD analysis.

Cherenkov in water pool

The Water Cherenkov Detector (WCD) is the instrumentation of the water buffer around the CD. When high speed charged particles will pass through the water, they will produced cherenkov photons. The light will be collected by 2400 MCP LPMTs installed on the outer surface of the CD structure. The muons veto strategy is based on a PMT multiplicity condition. WCD PMTs are grouped in ten zones: 5 in the top, 5 in the bottom. A veto is raised either when more than 19 PMTs are triggered in one zone or when two adjacent zones simultaneously trigger more than 13 PMTs. Using this trigger, we expect to reach a muon detection efficiency of 99.5% while keeping the noise at reasonable level.

Top tracker

The JUNO Top Tracker (TT) is a plastic scintillator detector located on the top of the experiment (see Figure 2.11). Made from plastic scintillator from OPERA [25] layered horizontally in 3 layers on the top of the detector, the TT will be able to detect incoming atmospheric muons. With its coverage, about 1/3 of the of all atmospheric muons that passing through the CD will also pass through the 3 layer of the detector. While it does not cover the majority of the CD, the TT is particularly effective to detect muons coming through the filling chimney region which might present difficulties from the other subsystems in some classes of events.

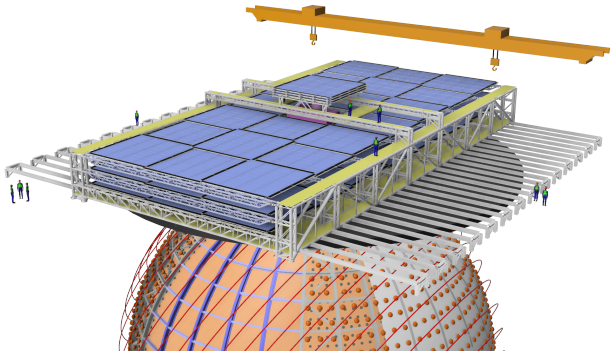


FIGURE 2.11 – The JUNO top tracker

2.4 Calibration strategy

The calibration is a crucial part of the JUNO experiment. The detector will continuously bath in neutrinos coming from the close nuclear power plant, from other sources such as geo neutrinos, the sun and will be exposed to background noise coming from atmospheric muons and natural radioactivity. Because of this continuous rate, low frequency signal event, we need high frequency, recognisable sources in the energy range of interest : [0-12] MeV for the positron signal and 2.2 MeV for the neutron capture. It is expected that the CD response will be different depending on the type of particle, due to the interaction with LS, the position on the event and the optical response of the acrylic sphere (see Section 3.3). We also expect a non-linear energy response of the CD due to the LS properties [17] but also due to the reponse of the LPMTs system when collecting a large amount of PE [24].

2.4.1 Energy scale calibration

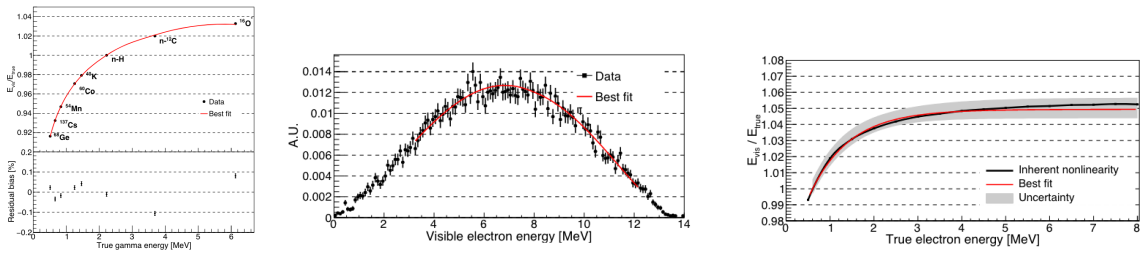
While electrons and positrons sources would be ideal, for a large LS detector thin-walled electrons or positrons sources could lead to leakage of radionucleides causing radioactive contamination. Instead, we consider gamma sources in the range of the prompt energy of IBDs. The sources are reported in table 2.3.

Sources / Processes	Type	Radiation
^{137}Cs	γ	0.0662 MeV
^{54}Mn	γ	0.835 MeV
^{60}Co	γ	1.173 + 1.333 MeV
^{40}K	γ	1.461 MeV
^{68}Ge	e^+	annihilation 0.511 + 0.511 MeV
^{241}Am-Be	n, γ	neutron + 4.43 MeV (12C*)
^{241}Am-^{13}C	n, γ	neutron + 6.13 MeV (16O*)
$(n, \gamma)p$	γ	2.22 MeV
$(n, \gamma)^{12}$C	γ	4.94 MeV or 3.68 + 1.26 MeV

TABLE 2.3 – List of sources and their process considered for the energy scale calibration

For the ^{68}Ge source, it will decay in ^{68}Ga via electron capture, which will itself β^+ decay into ^{68}Zn. The positrons will be absorbed by the enclosure so only the annihilation gamma will be released. In addition, (α, n) sources like ^{241}Am-Be and ^{241}Am-^{13}C are used to provide both high energy gamma and neutrons, which will later be captured in the LS producing the 2.2 MeV gamma.

From this calibration we call E_{vis} the "visible energy" that is reconstructed by our current algorithms and we compare it to the true energy deposited by the calibration source. The results shown in Figure 2.12 show the expected response of the detector from calibration sources. The non-linearity is clearly visible from the E_{vis}/E_{true} shape. See [26] for more details.



(A) Gamma non-linearity (B) Boron spectrum (C) Electron non-linearity

FIGURE 2.12 – Fitted and simulated non linearity of gamma, electron sources and from the ^{12}B spectrum. Black points are simulated data. Red curves are the best fits. Figures taken from [26].

2.4.2 Calibration system

The non-uniformity due to the event position in the detector (more details in Section 3.3) will be studied using multiples systems that are schematized in Figure 2.13. They allow to position sources at different location in the CD.

- For a one-dimension vertical calibration, the Automatic Calibration Unit (ACU) will be able to deploy multiple radioactive sources or a pulse laser diffuser ball along the central axis of the CD through the top chimney. The source position precision is less than 1cm.
- For off-axis calibration, a calibration source attached to a Cable Loop System (CLS) can be moved on a vertical half-plane by adjusting the length of two connection cable. Two set of CSL will be deployed to provide a 79% effective coverage of a vertical plane.
- A Guiding Tube (GT) will surround the CD to calibrate the non-uniformity of the response at the edge of the detector
- A Remotely Operated under-LS Vehicle (ROV) can be deployed to desired location inside LS for a more precise and comprehensive calibration. The ROV will also be equipped with a camera for inspection of the CD.

The preliminary calibration program is depicted in table 2.4.

2.4.3 Instrumental non-linearity calibration

One of the main interests of Dual Calorimetry is to calibrate away an instrumental effect called charge non linearity (QNL), which will be described in more detail in Chapter 7.

In short, during a typical IBD event, between 0 and 100 PEs can be produced in a given LPMT (depending on the position of the interaction and the positron energy). This is a large dynamic range. When the number of PEs is high, the reconstruction of the LPMT charge can become inaccurate, underestimating the actual number of PEs as illustrated in Figure 2.14. This QNL is difficult to separate from other non linearities (like the non linearity in the LS photon yield as a function of the deposit energy). In chapter 5 and 6 of this thesis [24], a calibration method that constitutes the core of dual calorimetry are described. They are based on the comparisons between signals seen in LPMTs and signals seen in SPMTs. In the latter system, due to its small angular coverage, individual SPMT

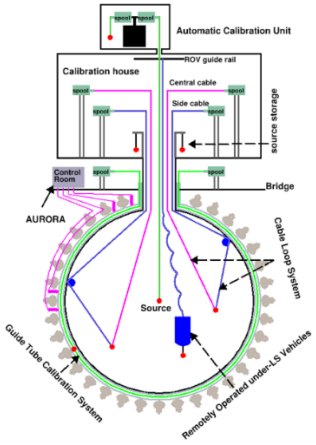


FIGURE 2.13 – Overview of the calibration system

Program	Purpose	System	Duration [min]
Weekly calibration	Neutron (Am-C)	ACU	63
	Laser	ACU	78
Monthly calibration	Neutron (Am-C)	ACU	120
	Laser	ACU	147
	Neutron (Am-C)	CLS	333
	Neutron (Am-C)	GT	73
Comprehensive calibration	Neutron (Am-C)	ACU, CLS and GT	1942
	Neutron (Am-Be)	ACU	75
	Laser	ACU	391
	^{68}Ge	ACU	75
	^{137}Cs	ACU	75
	^{54}Mn	ACU	75
	^{60}Co	ACU	75
	^{40}K	ACU	158

TABLE 2.4 – Calibration program of the JUNO experiment

rarely see more than 1 PE per event, and therefore are essentially immune against QNL. The method described in [24] uses a tunable light source covering the range of 0 to 100 PE perLPMT channel

2.5 Satellite detectors

As introduced in Section 2.1.1 and section 2.3.2, the precise knowledge and understanding of the detector condition is crucial for the measurements of the NMO and oscillation parameters. Thus two satellite detectors will be setup to monitor the experiment condition. TAO to monitor and understand the $\bar{\nu}_e$ flux and spectrum coming from the nuclear reactor and OSIRIS to monitor the LS response.

2.5.1 TAO

The Taishan Antineutrino Observatory (TAO) [13, 27] is a ton-level gadolinium doped liquid scintillator detector that will be located near the Taishan-1 reactor. It aim to measure the $\bar{\nu}_e$ spectrum at very low distance (44m) from the reactor to measure a quasi-unoscillated spectrum. TAO also aim to

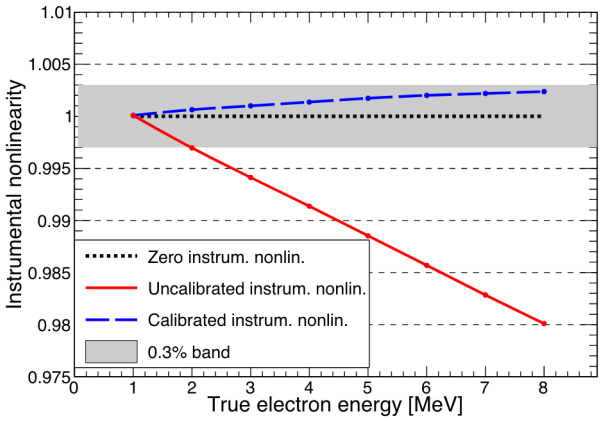


FIGURE 2.14 – Event-level instrumental non-linearity, defined as the ratio of the total measured LPMT charge to the true charge for events at the center of the detector. The solid red line represents event-level non-linearity without the channel-level correction in an extreme hypothetical scenario of 50% non-linearity over 100 PEs for the LPMTs. The dashed blue line represents that after the channel-level correction. The gray band shows the residual uncertainty of 0.3%, after the channel-level correction. Figure taken from [26].

provide a major contribution to the so-called reactor anomaly [12]. Its requirement are to the level of 2 % energy resolution at 1 MeV.

Detector

The TAO detector is close, in concept, to the CD of JUNO. It is composed of an acrylic vessel containing 2.8 tons of gadolinium-loaded LS instrumented by an array of silicon photomultipliers (SiPM) reaching a 95% coverage. To efficiently reduce the dark count of those sensors, the detector is cooled to -50 °C. The $\bar{\nu}_e$ will interact with the LS via IBD, producing scintillation light, that will be detected by the SiPMs. From this signal the $\bar{\nu}_e$ energy and the full spectrum reconstructed. This spectrum will then be used by JUNO to calibrate the unoscillated spectrum, most notably the fission product fraction that impact the rate and shape of the spectrum. A schema of the detector is presented in Figure 2.15a.

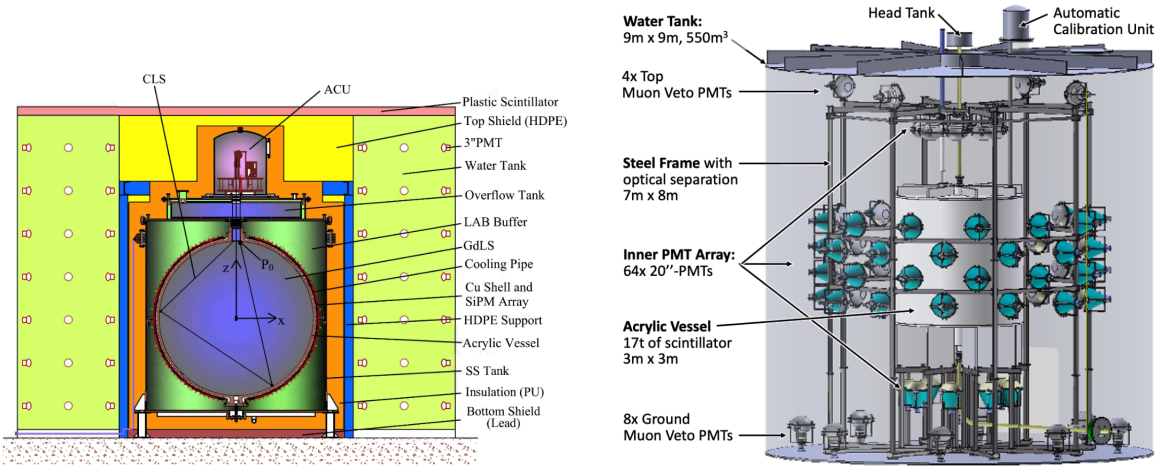
2.5.2 OSIRIS

The Online Scintillator Internal Radioactivity Investigation System (OSIRIS) [22] is an ultralow background, 20 m^3 LS detector that will be located in JUNO cavern. It aim to monitor the radioactive contamination, purity and overall response of the LS before it is injected in JUNO. OSIRIS will be located at the end of the purification chain of JUNO, monitoring that the purified LS meet the JUNO requirements. The setup is optimized to detect the fast coincidences decay of ^{214}Bi $-^{214}$ Po and ^{212}Bi $-^{212}$ Po, indicators of the decay chains of U and Th respectively.

Detector

OSIRIS is composed of an acrylic vessel that will contains 17t of LS. The LS is instrumented by a PMT array of 64 20 inch PMTs on the top and the side of the vessel. To reach the necessary

background level required by the LS purity measurements, in addition to being 700m underground in the experiment cavern, the acrylic vessel is immersed in a tank of ultra pure water. The water is itself instrumented by another array of 20 inch PMTs, acting as muon veto. A schema of the detector is presented in Figure 2.15b.



(A) Schematic of the TAO satellite detector

(B) Schematic of the OSIRIS satellite detector

FIGURE 2.15

2.6 Software

The simulation, reconstruction and analysis algorithms are all packaged in the JUNO software, subsequently called the software. It is composed of multiple components integrated in the SNiPER [28] framework:

- Various primary particles simulators for the different kind of events, background and calibration sources.
- A Geant4 [29–31] Monte Carlo (MC) simulation containing the detectors geometries, a custom optical model for the LS and the supporting structures of the detectors. The Geant4 simulation integrate all relevant physics process for JUNO, validated by the collaboration. This step of the simulation is commonly called *Detsim* and compute up to the production of photo-electrons in the PMTs. The optics properties of the different materials and detector components have been measured beforehand to be used to define the material and surfaces in the simulation.
- An electronic simulation, simulating the response waveform of the PMTs, tracking it through the digitization process, accounting for effects such as non-linearity, dark noise, Time Transit Spread (TTS), pre-pulsing, after-pulsing and ringing if the waveform. It's also the step handling the event triggers and mixing. This step is commonly referenced as *Elecsim*.
- A waveform reconstruction where the digitized waveform are filtered to remove high-frequency white noise and then deconvoluted to yield time and charge informations of the photons hits on the PMTs. This step is commonly referenced as *Calib*.
- The charge and time informations are used by reconstruction algorithms to reconstruct the interaction vertex and the deposited energy. This step is commonly reported as *Reco*. See Section 3.3 for more details on the reconstruction.
- Once the singular events are reconstructed, they go through event pairing and classification to select IBD events. This step is named Event Classification.

— The purified signal is then analysed by the analysis framework which depend of the physics topic of interest. An introduction to the reactor $\bar{\nu}_e$ is presented in Section 2.7.

The steps Reco and Event Classification are divided into two category of algorithm. Fast but less accurate algorithms that are running during the data taking designated as the *Online* algorithms. Those algorithm are used to take the decision to save the event on tape or to throw it away. More accurate algorithms that run on batch of events designated *Offline* algorithms. They are used for the physics analysis. The Offline Reco will be one of the main topic of interest for this thesis.

2.7 Reactor anti-neutrino oscillation analysis

2.7.1 IBD samples selection

The $\bar{\nu}_e$ coming from nuclear reactor will, for the most part, interact with proton, hydrogen nucleus, via Inverse Beta Decay (IBD). The first step of the oscillation analysis is to constitute a sample of IBD candidates, dominated by actual IBDs. The IBD interaction, schematised in Figure 2.5, will produce two particle, with differentiable signals.

The first signal comes from the positron slowdown and its annihilation with an electron of the LS. This is the *prompt* signal, happening a few ns after the IBD. The positron takes most of the $\bar{\nu}_e$ kinetic energy, as detailed in Section 2.3.1.

The leftover kinetic energy is taken by the neutron that, after thermalisation in the LS, will be captured by an hydrogen and produce a 2.2 MeV gamma, or by a carbon emitting a 4.9 MeV gamma. This is the *delayed* signal, happening $\sim$236 μs after the IBD. This second mono-energetic event serve as a marker for the IBD.

The IBD selection is thus based on the selection of a prompt event, with an energy between 0.8 and 12 MeV, and a delayed event with an energy in the ranges $[1.9, 2.5]$ MeV or $[4.4, 5.5]$ MeV. Those two signal needs to be in a 1 ms time window and within 1.5 m from each other. Additionally the two signal needs to be in a radius of 17.2m from the detector center (0.5 m from the edge) to protect from accidental background formed by two uncorrelated signals [32]. Those values will be further refined after once JUNO data-taking starts.

In addition, specials veto are setup to protect from cosmic muons and their aftermath. The details of those veto and selection can be found in [32].

The expected rate and selection efficiency on IBD can be found in table 2.5. After these selection, the residual background, including $\bar{\nu}_e$ coming from other sources than the reactor can be found in table 2.6.

Selection Criterion	Efficiency [%]	IBD Rate [day^{-1}]
All IBDs	100.0	57.4
Fiducial Volume	91.5	52.5
IBD Selection	98.1	51.5
Energy Range	99.8	-
Time Correlation (ΔT_{p-d})	99.0	-
Spatial Correlation (ΔR_{p-d})	99.2	-
Muon Veto (Temporal + Spatial)	91.6	47.1
Combined Selection	82.2	47.1

TABLE 2.5 – Summary of cumulative reactor antineutrino selection efficiencies. The reported IBD rates (with baselines <300 km) refer to the expected events per day after the selection criteria are progressively applied. Table taken from [32]

Backgrounds	Rate [day^{-1}]	B/S [%]
Geoneutrinos	1.2	2.5
World reactors	1.0	2.1
Accidentals	0.8	1.7
^{9}Li/^{8}He	0.8	1.7
Atmospheric neutrinos	0.16	0.34
Fast neutrons	0.1	0.21
^{13}C(α,n)^{16}O	0.05	0.01
Total backgrounds	4.11	8.7

TABLE 2.6 – Expected background rates, background to signal ratio (B/S), and rate and shape uncertainties. The B/S ratio is calculated by using the IBD signal rate of 47.1/day. Table taken from [32]

Once a sample is obtained, the oscillation analysis will consist essentially on the fit of a spectrum model to the spectrum observed in the selected sample. More specifically, the spectrum under analysis is the spectrum of the reconstructed visible energy of the positron : E^{vis}. The reconstruction is presented in detail in Section 3.3. For 6 years of data taking, it will resemble that on Figure 2.3. In the next sections, I describe the fit procedures developed in JUNO. This will be the occasion to introduce notions useful for Chapter 7. Besides, I'll also describe the versions of the fit used in this Chapter 7.

2.7.2 Synthetic overview of fit procedures developed at JUNO

Several fit procedures are being developed by JUNO collaborators (half a dozen of groups work in parallel within the collaboration). We do not have the ambition of a thorough description here. Instead, we try to introduce the main elements useful to the reader to understand JUNO's future results, and the fit procedures used Chapter 7.

In most cases, the fit is a binned fit to the histogrammed spectrum of $E^{e^+}_{vis}$, like the one in Figure 2.3. It is is based on the minimization of a χ^2 test statistics. Generically, it can be written this way :

$$\chi^2 = (\boldsymbol{T}(\boldsymbol{\theta},\boldsymbol{\eta}) - \boldsymbol{D})^T V^{-1} (\boldsymbol{T}(\boldsymbol{\theta},\boldsymbol{\eta}) - \boldsymbol{D}) + \chi^2_{nuis}(\boldsymbol{\eta}) \tag{2.3}$$

where the components of data vector $\boldsymbol{D}$ are the number of events found in individual bins of the fitted histogram, $\boldsymbol{T}(\boldsymbol{\theta},\boldsymbol{\eta})$ is the vector of the predicted number of entries in each bins. This prediction is the integration over the width of the bins of the spectrum model for a given NMO (described latter in this section).

This model depends on the oscillation parameters $\boldsymbol{\theta} = (\Delta m^2_{21}, \sin^2(2\theta_{12}), \Delta m^2_{31}, \sin^2(2\theta_{13}))$, and on nuisance parameters $\boldsymbol{\eta}$ involved in the fit model and associated with systematic uncertainties. Uncertainties are treated in two ways : statistical and some of the systematic uncertainties are accounted for via the covariance matrix $V = V_{stat} + V_{syst}$; remaining systematic uncertainties are treated via the penalty term χ^2_{nuis}, which is written this way :

$$\chi^2_{nuis}(\boldsymbol{\eta}) = (\boldsymbol{\eta} - \bar{\boldsymbol{\eta}})^T \cdot V^{-1}_{\boldsymbol{\eta}}(\boldsymbol{\eta}) \cdot (\boldsymbol{\eta} - \bar{\boldsymbol{\eta}}) \tag{2.4}$$

where $\bar{\boldsymbol{\eta}}$ is the vector containing the most probable values of the nuisance parameters according to our knowledge prior to the fit, and where $V_{\boldsymbol{\eta}}$ is the covariance matrix accounting of the uncertainty on these values, and the potential correlations between them. In principles, a likelihood could be used instead of a χ^2. However, some of the systematic uncertainties are not trivial to parameterize, therefore treating them as nuisance parameters in not trivial.

An example of nuisance parameters are the A, B and C parameters of equation 7.19, which can be used to describe the resolution on the reconstructed energy. The fit model leading to $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$ indeed incorporates this resolution.

Treatment of uncertainties

Differences between various fit procedures developed within JUNO often lies in the choice of the systematic uncertainties that are treated via V or $\chi^2_{nuis}(\boldsymbol{\eta})$. Among the reasons behind these differences is the necessity to compare several approaches to ensure the robustness JUNO's oscillation analysis results. This approach was already adopted in the recent evaluations of JUNO's potential [3, 32]. Studies carried out so far at Subatech assumes a treatment entirely via V.

Other differences lies in the choice of the way to evaluate V_{stat}. Two common approaches used in χ^2 fit are the Neyman and the Pearson approaches. If the size of the fitted sample is high enough, the variation of D_i, the number of entries in bin i, around its true expectation value $\bar{D}_i$ is $\sqrt{\bar{D}_i}$. To evaluate this number, the Neyman approach uses simply the number of entries observed in the sample under analysis : $\sqrt{D_i}$. The Pearson approach uses the prediction by the fit model : $\sqrt{T(\boldsymbol{\theta}, \boldsymbol{\eta})_i}$.

Both cases are approximations which lead to biases that are not tolerable given the precision JUNO must aim at for a successful oscillation analysis. To reduce this bias, most of JUNO groups employ the "Combined Neyman Pearson" approach introduced in [33]. Schematically, it consists on combining both approaches : $(V_{stat})_{ii} = 3/\left(\frac{1}{T(\boldsymbol{\theta},\boldsymbol{\eta})_i} + \frac{2}{D_i}\right)$. Weights in this relation are chosen in order to cancel typical biases. The validity of this method is not guaranteed universally. In particular, limitations appear when a complex systematic matrix V_{syst} is added to V_{stat}.

This is the case in the approach followed at Subatech, were all sources of systematic uncertainties are treated via this matrix. Dedicated studies run at Subatech observed biases in the fitted oscillation parameters using CNP in this case. Subatech's group therefore adopted another approach (verified to be unbiased).

Originally, fitting the $E^{e^+}_{vis}$ spectrum should mean maximising a likelihood, equal to the product over all bins of the probabilities to find D_i in bin i. With a large enough samples, this product tends to a multidimensional gaussian (one dimension per bin) :

$$\mathcal{L} = 2\pi^{-\frac{N}{2}} |V|^{-\frac{1}{2}} e^{-\frac{1}{2}(\boldsymbol{D}-\boldsymbol{T}(\boldsymbol{\theta},\boldsymbol{\eta}))^T V^{-1} (\boldsymbol{D}-\boldsymbol{T}(\boldsymbol{\theta},\vec{\eta}))} \tag{2.5}$$

Replacing $\mathcal{L}$ by $-2\ln\mathcal{L}$ one obtains :

$$\chi^2_{PV} = (\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta}) - \boldsymbol{D})^T V^{-1} (\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta}) - \boldsymbol{D}) + \ln(|V|) \tag{2.6}$$

where V is the total covariance matrix with its statistical component evaluated according to the Pearson approach. The $\ln|V|$ term, often neglected in χ^2 fits, ensures that biases, essentially related to the normalisation of the fitted distribution, are avoided. This "PearsonV" χ^2 is the one that we minimize in the fits used in Chapter 7.

Another difference between the various procedures developed at JUNO is the choice of the spectrum range and binning. So far, at Subatech, we use an histogram defined between 0.8 and 9 MeV, and a regular binning involving 20 keV wide bins.

Joint fit of JUNO and TAO spectra

Another difference between the various fit procedures developed in the collaboration is the inclusion of the data collected by TAO (see Section 2.5.1). The spectrum prediction $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$ involves predictions

on the differential flux of $\bar{\nu}_e$ as a function of $E_{\bar{\nu}_e}$ produced in reactors. This is one of the main systematic uncertainties affecting the oscillation analysis. This can be constrained using the data of TAO. An efficient way to use them is via a simultaneous fit, which will constrain the part of the $\boldsymbol{\eta}$ parameters related to the reactor predictions. In this case, equation 2.3 becomes :

$$\chi^2 = \sum_d \left(\boldsymbol{T}^d(\boldsymbol{\theta}^d, \boldsymbol{\eta}) - \boldsymbol{D}^d\right)^T V^{-1} \left(\boldsymbol{T}^d(\boldsymbol{\theta}^d, \boldsymbol{\eta}) - \boldsymbol{D}^d\right) + \chi^2_{nuis}(\boldsymbol{\eta}) \tag{2.7}$$

where the d superscript stands for the spectrum measured in JUNO or TAO.

Finally, it must be noted that JUNO's sensitivity to $\sin^2(2\theta_{13})$ is too weak for a competitive measurement. In most versions of the oscillation analyses carried out within JUNO, it will be considered as a nuisance parameter. In practice, the various χ^2's presented earlier will receive an additional term :

$$\chi^2_{\sin^2(2\theta_{13})} = \frac{(\sin^2(2\theta_{13}) - \overline{\sin^2(2\theta_{13})})^2}{\sigma^2_{\overline{\sin^2(2\theta_{13})}}} \tag{2.8}$$

where $\overline{\sin^2(2\theta_{13})}$ and the denominators can be provided, for instance, by the world average on this parameter.

2.7.3 The spectrum model and sources of systematic uncertainties

The $E^{e^+}_{vis}$ spectrum observed in data (Fig 2.3) is the sum of the IBD spectrum and of the various backgrounds spectra (see table 2.6). The spectrum prediction $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$ is therefore the sum of IBD and backgrounds predictions. The latter are provided by MC simulations. The former results from the theoretical description of the series of phenomena that lead to the observed IBD spectrum. In a given bin i, it can be expressed this way :

$$T^i(\boldsymbol{\theta}, \boldsymbol{\eta}) = \sum_j C^{E_{rec}}_{ij} \int_{E^{vis}_j}^{E^{vis}_{j+1}} \mathrm{d}E^{vis} \int_{-1}^{1} \mathrm{d}\cos\theta \; \Phi(E^\nu) \frac{\mathrm{d}\sigma}{\mathrm{d}\cos\theta}(E^\nu, \cos\theta) \frac{\mathrm{d}E^\nu}{\mathrm{d}E^{dep}} \frac{\mathrm{d}E^{dep}}{\mathrm{d}E^{vis}} \tag{2.9}$$

In the above equation, 4 kinds of energies appears: following the IBD, the antineutrino energy E^ν is quasi entirely transferred to the positron, of energy E_e. It eventually annihilates, so the actual energy released in the LS is E_{dep}, which includes the mass of the annihilated electron. The production optical photons is not linear in E_{dep} (see Section 2.4), so that the visible energy (that will be reconstructed) is E_{vis}. This reconstruction comes with resolution effects, leading to E_{rec}.

Equation 2.9 describe the passage from the original differential flux (as a function of E^ν) of antineutrinos reaching the detector to the reconstructed spectrum:

- $\Phi(E^\nu)$ is the differential antineutrino flux reaching JUNO.
- $\frac{\mathrm{d}\sigma}{\mathrm{d}\cos\theta}(E^\nu, \cos\theta)$ account for the IBD cross section, which depends on the antineutrino energy and on the incidence angle.
- The last two terms of the integrand are the differential relations linking E^ν, E^{dep} and E^{vis}.
- Reconstruction effects are described via C^{rec}_{ij}'s, that make the link between the true and reconstructed visible energy. It a simple case, it is equivalent to a convolution product. The matrix formalism here prepares the fact that a realistic analysis might employ a more empirical way, based on MC.

The differential flux is expressed this way:

$$\Phi(E^\nu) = \sum_r \left(\frac{\mathcal{P}_{\bar{\nu}_e \to \bar{\nu}_e}(E^\nu, L_r)}{4\pi L_r^2} \frac{W_r}{\sum_i f_{i,r} e_i} \sum_i f_{i,r} s_i(E^\nu) \right) \tag{2.10}$$

where:

- $\mathcal{P}_{\bar{\nu}_e \to \bar{\nu}_e}(E^\nu, L_r)$ is the antineutrino survival probability at distance L_r from the production point in reactor r, dictated by the oscillation probability.
- e_i stands for the mean energy released per fission for isotope i.
- W_r is the thermal power of reactor r.
- $f_{i,r}$ is the fission fraction in reactor r of isotope i among the four.
- $s_i(E^\nu)$ is the $\bar{\nu}_e$ energy spectrum - at emission point - per fission for each isotope, as emitted by the reactor.

Sources of systematic uncertainties

The numerous quantities appearing in the spectrum model embody a good part of the systematic uncertainties. Among the leading contributions are those related to the knowledge of the reactor related quantities. Of importance are also the uncertainties related to the modelling of the non linearity of the photon emission (passage from E^{dep} to E^{vis}) and of the reconstruction resolution. The shape and rate of the backgrounds are also a leading source of systematic uncertainties. The uncertainty on IBD selection efficiency also has a notable role.

Sensitivities to NMO and oscillation parameters

JUNO will start taking data in 2025. During the months and years to come, oscillation analyses will naturally be optimized regularly. What we described here represent the state of the art mid 2024, and was used for the sensitivity studies published in [3, 32] and are presented in table 2.7

	Central Value	PDG 2020	100 days	6 years	20 years
$\Delta m^2_{31} (\times 10^{-3} \text{eV}^2)$	2.5283	$\pm$0.034 (1.3%)	$\pm$0.021 (0.8%)	$\pm$0.0047(0.2%)	$\pm$0.0029(0.1%)
$\Delta m^2_{21} (\times 10^{-3} \text{eV}^2)$	7.53	$\pm$0.18 (2.4%)	$\pm$0.074 (1.0%)	$\pm$0.024(0.3%)	$\pm$0.017(0.2%)
$\sin^2 \theta_{12}$	0.307	$\pm$0.013 (4.2%)	$\pm$0.0058 (1.9%)	$\pm$0.0016(0.5%)	$\pm$0.0010(0.3%)
$\sin^2 \theta_{13}$	0.0218	$\pm$0.0007 (3.2%)	$\pm$0.010 (47.9%)	$\pm$0.0026(12.1%)	$\pm$0.0016(7.3%)

TABLE 2.7 – A summary of precision levels for the oscillation parameters. The reference value (PDG 2020 [34]) is compared with 100 days, 6 years and 20 years of JUNO data taking.

Asimov studies

To study the behavior and performance of fit procedures with enough realism, one should perform fits to a large number of toy spectra, generated with a number events equal to what one expects in real data, for the given exposure under consideration. This allows to study the impact of realistic statistical fluctuations. This is, however, time consuming, since thousands of spectra have to be generated and fitted.

When subtle details are not crucial, another approach is possible to estimate sensitivities to the NMO and oscillation parameters, as well as (for instance) to verify the technical implementation of fitter (as we will do in Chapter 7 for the implementation of the joint fit). It consists on generating only 1

pseudo-data sample, where the content of each bin D^i is set to the predicted value T^i, computed with a reasonable choice for the values of the model parameters (for instance, with the recent PDG values for the oscillation parameters). This is equivalent to a spectrum with fluctuations. It provides valid sensitivities if the expected statistics in the real data sample is high enough in each bin to assume a gaussian behavior.

2.7.4 Versions of the fit used in this thesis

In Chapter 7, we'll study the potential of a particular application of Dual Calorimetry, call "Dual Calorimetry with neutrino oscillation." This approach require to perform fits to the E^{vis} spectrum reconstructed with the LPMT system, with the SPMT system, and a joint fit to both spectra.

In the two former cases, the PearsonV χ^2 introduced above will be used. In the latter case, it will be extended in the following way : The $\boldsymbol{D}$ data vector now possess 820 elements. Indeed, the fit is performed to a joint spectrum, where the LPMT spectrum is juxtaposed with the SPMT spectrum (see Figure 2.16).

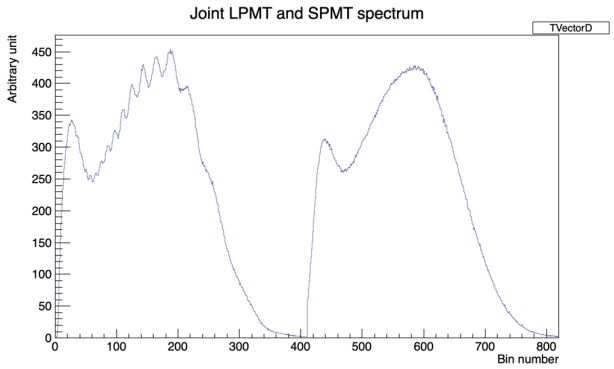


FIGURE 2.16 – Illustration of the spectrum considered when joint fitting

The prediction vector $\boldsymbol{T}\left(\boldsymbol{\theta}^d, \boldsymbol{\eta}\right)$ is naturally extended in the same way. Its components 1 to 410 predict the number of entries in the LPMT part of the LPMT+SPMT joint spectrum, while its components from 411 to 820 predict the contents of the SPMT part. Note that the list of oscillation parameters in $T_{411}(\boldsymbol{\theta}^d, \boldsymbol{\eta})$ to $T_{820}(\boldsymbol{\theta}^d, \boldsymbol{\eta})$ is the same as usual. However, $T_1(\boldsymbol{\theta}^d, \boldsymbol{\eta})$ to $T_{410}(\boldsymbol{\theta}^d, \boldsymbol{\eta})$ 2 additional parameters, $\delta \sin^2(2\theta_{12})$ and $\delta\Delta m^2_{21}$, are added to the corresponding oscillation parameters to account for a potential unexpected problem in the LPMT reconstruction or calibration.

In the case of this joint fit, the covariance matrix V is extended to a (820×820) matrix. It is a central element of this study, as will be explained in Chapter 7, since the LPMT and SPMT data spectrum are correlated, even at the statistical level. The determination of this matrix will be an important and original point.

Fits will be performed to an histogram spectrum defined over the 0.8-9 MeV range, with a flat binning (20 keV wide bins), often restricted to the 335 lowest E^{vis} bins.

In this Section 2.7, we have provided a theoretical description of the fit procedures developed at JUNO. Software frameworks are necessary to use them in practice. The framework developed at Subatech will be described in Chapter 7.

2.7.5 Physics results

The oscillation parameters are directly extracted from the minimization procedure and the error can be estimated directly from the procedure. For the NMO, the data are fitted under the two assumption of NO and IO. The difference in χ^2 give us the preferred ordering and the significance of our test. Latest studies show that the precision on oscillation parameters after six year of data taking will be of 0.2%, 0.3%, 0.5% and 12.1% for Δm^2_{31}, Δm^2_{21}, $\sin^2\theta_{12}$ and $\sin^2\theta_{13}$ respectively [3]. The expected sensitivity to mass ordering is 3σ after 6.5 years [35].

2.8 Summary

JUNO is one the biggest new generation neutrino experiment. Its goal, the measurements of oscillation parameters with unprecedented precision and an NMO preference at the 3 sigma confidence level, needs an in depth knowledge and understanding of the detector and the physics at hand. The characterisation and calibration of the detector are of the utmost importance and the understanding of the detector response in its resolution and bias is capital to be able to correctly carry the high precision physics analysis of the neutrino oscillation.

In this thesis, I explore the usage of data-driven reconstruction methods to validate and optimize the reconstruction of IBD events in JUNO in the chapters 4, 5 and 6 and the usage of the dual calorimetry in the detection of possible mis-modelisation in the theoretical spectrum 7.

Chapter 3

Introduction to the methods and algorithms used in this thesis

"I have the shape of a human being and organs equivalent to those of a human being. My organs, in fact, are identical to some of those in a prosthetized human being. I have contributed artistically, literally, and scientifically to human culture as much as any human being now alive. What more can one ask?"

Isaac Asimov, The Complete Robot

Contents

Machine Learning (ML) and more specifically Neural Network (NN) are families of data-driven algorithms. They are used in a wide variety of domains including natural language processing, computer vision, speech recognition and, the subject of this thesis, scientific studies.

Machine learning models aim to learn underlying patterns from finite datasets in order to make general predictions or classifications. For example, in our case, it could be an algorithm that would differentiate the nature of a particle interacting in the liquid scintillator, between a positron and an electron, based on the readout charge and time (Q, t) of the 17612 LPMT of the JUNO experiment. During a first training phase, it would learn the discriminative features between the two in the 35224-dimensional charge and time distribution, built from samples of e^+ and e^- events.

It extracts essential features from a highly complex and multi-dimensional dataset that describe the physical interactions: a three body energy deposition (the positron and two annihilation gammas) and the single deposit from an electron.

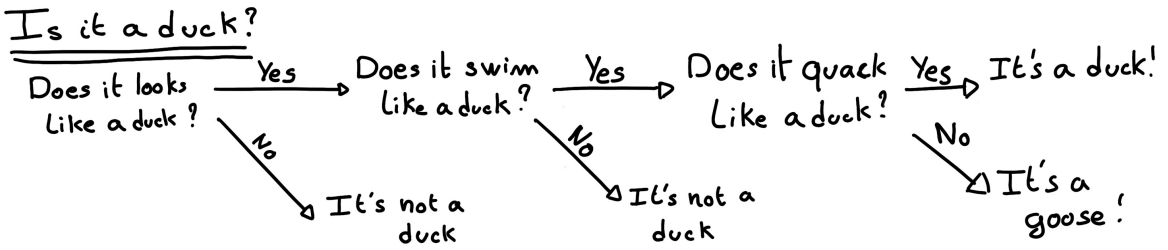


FIGURE 3.1 – Example of a BDT that determine if the given object is a duck

Ideally, the algorithm would learn to recognize those informations on its own, regardless of the input size and complexity. In practice, however, these algorithms are guided by human design through their architectures and training conditions. We can still hope that they an use more thoroughly the detector informations while traditional methods are often subject to assumptions or simplifications to make the task easier (see for instance the algorithm in Section 3.3).

The role of machine learning algorithms has expanded rapidly in the past decade, either as the main or secondary algorithm for a wide variety of tasks: event reconstruction, event classification, waveform reconstruction and so on. In particular in domains where the underlying physic and detector processes are complex and highly dimensional, and when large amount of data must be processed quickly.

This chapter present an overview of the different kind of machine learning methods and neural networks that will be discussed in this thesis, and the state of the art of the reconstructions methods in JUNO our ML algorithms will be compared to.

3.1 Core concepts in machine learning and neural networks

In this section, we discuss the core concepts in machine learning that will be used thorough this thesis. We place particular emphasis on Neural Networks, as it's the family of the algorithms described in chapters 4, 5 and 6.

3.1.1 Boosted Decision Tree (BDT)

One of the most classic machine learning algorithm used in particle physics is Boosted Decision Tree (BDT) [36] (or more recently Gradient Boosting Machine [37]).

BDTs operate by making a series of decisions based on a set of input features, with each decision represented as a node in the tree. Each decision point, or node, takes its decision based on a set of trainable parameters leading to a subtree of decisions. The process is repeated until it reach the final node, yielding the prediction. A simplistic example is given in Figure 3.1.

The training procedure followes a reward-based approach where the algorithm predictions are compared to the true outcomes. During the training phase the prediction of the BDT is compared to a known truth about the data. The score is then used to backpropagate corrections to the parameters of the tree. Modern BDT use gradient boosting where the gradient of the loss is calculated for each of the BDT parameters. Following the gradient descent, we can reach the, hopefully, global minima of the loss for our set of parameters.

3.1.2 Artificial Neural Network (NN)

One of the modern ML family is the Neural Network, historical name as their design was inspired by the behaviour of biological neurons in the brain. As schematized in Figure 3.2, the input, output

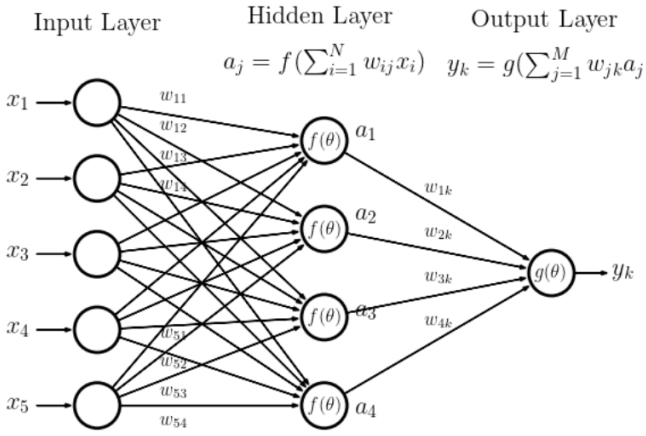


FIGURE 3.2 – Schema of a simple neural network

and steps inside the NN is described as neuron *layers*. The neurons of the layers take as input a set of values from the preceding layer, here the a_i takes every informations of the x_i input layer, and aggregate those values following learnable *parameters* w_{ij}. In the example in Figure 3.2, fully connected layers are used, meaning that each neuron in one layer is connected to every neuron in the previous layer.

The aggregation procedure is core of defining the architecture of the NN. The different architectures used in this thesis will be discussed in Section 3.2. The process is repeated until reaching the output layer.

For example, let's take the network in Figure 3.2 and say that a_1, a_2 and a_3 are the neurons of the output layer. We try to produce a vertex reconstruction algorithm that will approach the charge barycentre. Let's limit the input x_i to the charge of the ith PMT, one of the solution is to aggregate on a_1 the x coordinate of the barycenter. The network would thus adapt the w_{i1} parameters so they correspond to the x coordinates of the ith PMT. Same for the y and z coordinate on a_2 and a_3 respectively.

The layers used in the example above are designated as *Fully connected* layers, where every neurons of the layer is connected to the every neurons of the preceding layer. The layer can be expressed using the Einstein summation and in bold the learnable parameters

$$O_j = I_i + \boldsymbol{W}_j^i \tag{3.1}$$

where O_j is the output neurons vector (the a_i), I_i is the preceding layer neurons vector (the x_i) and $\boldsymbol{W}$ is the parameters, or weights, matrix (composed of the w_{ij}). In practice, this fully connected layer is often adjoined a bias $\boldsymbol{B}$ and an *activation function* F.

$$I_j = F(I_i \boldsymbol{W}_j^i + \boldsymbol{B}_j) \tag{3.2}$$

This is the fundamental component of the Fully Connected Deep NN (FCDNN) family presented in Section 3.2.1.

This description of neural networks as layers introduce the principles of *depth* and *width*, the number

of layers in the NN and the number of neurons in each layer respectively. Those quantities that not directly used for the computation of the results but describes the NN or its training are designated as *hyperparameters*.

Now we just need to adapt the parameters so that this network learn that w_{ij} are the PMT coordinate. We describe the space produced by the parameters of the network as the *parameter phase space* or *latent space*. The optimization of the network and exploration of this phase space is done through training over a *training dataset* as described in next section.

3.1.3 Training procedure

To adapt the parameters we need an object that describe how well the network perform. This is the *loss* of our neural networks $\mathcal{L}$. In our barycenter example, it could be the distance between the reconstructed and real barycenter. Using this metric we can adjust the parameters of our network.

Depending if we try to minimize or maximize it, it need to posses a minima or a maxima. For example when doing *regression*, i.e. produce a scalar result like the coordinates of a barycenter, a common loss is the Mean Square Error (MSE). Let i be our dataset, the N events considered for training, y_i be the target scalar, the barycenter positions of each events, x_i the input data, the charge vector, and $f(x_i, \boldsymbol{\theta})$ the result of the network. The network here is modelled by f, and its parameter $\boldsymbol{\theta}$

$$\mathcal{L} \equiv MSE = \frac{1}{N}\sum_{i}^{N}(y_i - f(x_i, \boldsymbol{\theta}))^2 \tag{3.3}$$

Another common loss function is the Mean Absolute Error (MAE)

$$\mathcal{L} \equiv MAE = \frac{1}{N}\sum_{i}^{N}|y_i - f(x_i, \boldsymbol{\theta})| \tag{3.4}$$

We see that those loss function possess a minima when $f(x_i, \boldsymbol{\theta}) = y_i$.

Modern neural networks typically use gradient descent to optimize their parameters by minimizing the loss function. The gradient of the parameter $\boldsymbol{w}$, designated in literature as $\boldsymbol{\theta}$, with respect of the loss function $\mathcal{L}$ is subtracted each optimisation step t

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \tag{3.5}$$

This induce $\mathcal{L}$ needs to be differentiable with respect to $\boldsymbol{\theta}$, thus the layers and their activation functions also need to be differentiable. This simple gradient descent, designated as Stochastic Gradient Descent (SGD), can be extended with first and second order momentums like in the Adam optimizer [38]. More details about the optimizers can be found in Section 3.1.3.

Training lifecycle

The training process of neural networks can vary depending on the application and dataset, but in this thesis, we follow a standard approach. As shown in Fig. 3.3, training is organized into *epochs*, each of which consists of several *steps*. During each step, the neural network optimizes its parameters using a *batch*, a subset of the entire training dataset.

The ideal batch size, meaning the number of events in each batch, would encompass the entire dataset to avoid bias introduced by sub-sample specificity. However, in large-scale experiments like JUNO, the batch size is often constrained by memory limitations due to the massive volume of data

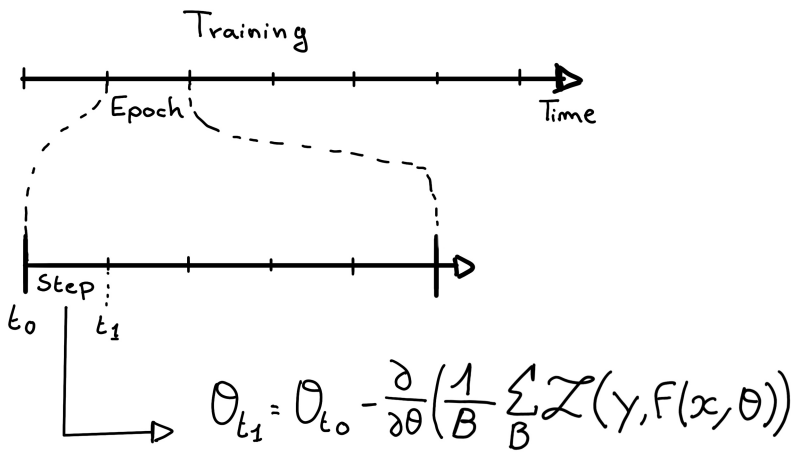


FIGURE 3.3 – Illustration of the training lifecycle

generated by the photomultiplier tubes (PMTs). Balancing batch size with memory capacity is crucial to ensure efficient and accurate training.

At the end of each epoch, the neural network is evaluated on a validation dataset, which is not used during training. This dataset serves as a reference to assess the network's performance and to monitor for signs of overfitting. In JUNO, this is critical because the model needs to generalize well to unseen experimental data and avoid overfitting to noise in the training set (see Section 3.1.4).

Hyperparameters that can be optimized during the training can be optimized at each epoch, for example the learning rate, or each step, the optimizer momentum for example.

There is not really a typical number of epochs or steps for the training. The number steps can be defined such as in one epoch, the NN see the entirety of the dataset but the number of steps and epochs are hyperparameters that are optimized over the each subsequent training. We adjust them by looking at the loss evolution profile over time.

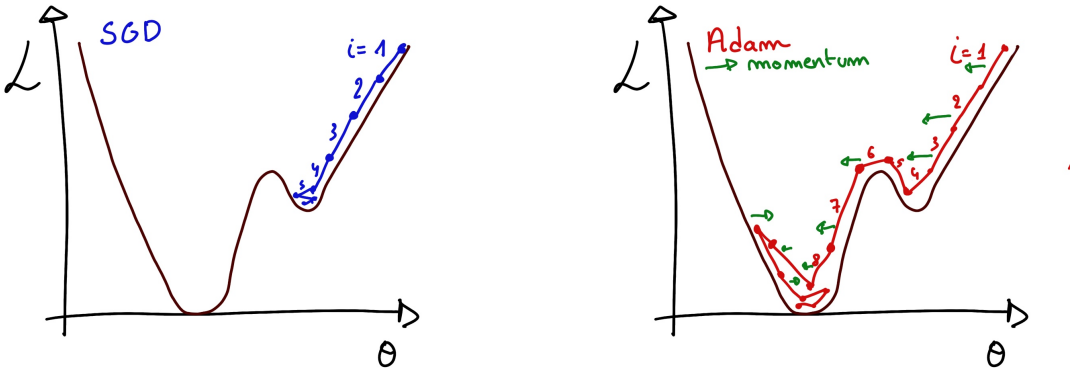
Most training are started with a fixed number of epochs, i.e. from what we've seen from precedent training, the network stop learning, the loss is constant, after N epoch so we run the training for $N + \delta$ epochs to see if the modification brings improvements to the loss profile. We can implement *early stopping policies* to halt training if certain conditions are met, such as a sudden increase in loss or when the loss plateaus. However, for the JUNO experiment, where training time is not a strict limitation, early stopping is less critical, though it may still be useful to prevent overfitting in some cases

The optimizer

As briefly introduced at the beginning of this section, the parameters of the neural network are optimized using the gradient descent method. We compute the gradient of the mean loss over the batch with respect of each parameters and we update the parameters in accord to minimize the loss. The gradient is computed backward from the loss up to the first layer parameters using the chain rule, in this case with only one parameter at each step for simplicity:

$$\frac{\partial \mathcal{L}}{\partial \theta_1} = \frac{\partial \theta_2}{\partial \theta_1}\frac{\partial \mathcal{L}}{\partial \theta_2} = \frac{\partial \theta_2}{\partial \theta_1}\frac{\partial \theta_3}{\partial \theta_2}\frac{\partial \mathcal{L}}{\partial \theta_3} = \frac{\partial \theta_2}{\partial \theta_1}\prod_{i=2}^{N-1}\frac{\partial \theta_{i+1}}{\partial \theta_i}\frac{\partial \mathcal{L}}{\partial \theta_N} \tag{3.6}$$

where θ is a parameter, i is the layer index. We see here that the gradient of the first layer is dependent of the gradient of all the following layers. Because the only value known at the start



(A) Illustration of SGD falling into a local minima

(B) Illustration of the Adam momentum allowing it to overcome local minima

FIGURE 3.4

of the optimization procedure is $\mathcal{L}$ we compute $\frac{\partial \mathcal{L}}{\partial \theta_N}$ then, $\frac{\partial \theta_N}{\partial \theta_{N-1}}$, etc... This is called the *backward propagation*.

This update of the parameters is done following an optimizer policy. Those optimizers depends on hyperparameters. The ones used in this thesis are:

1. Stochastic Gradient Descent (SGD). A simple but widely used optimizer that relies on one key hyperparameter, the learning rate (LR) / λ. It update each step the parameters θ following

$$\theta_{t+1} = \theta_t - \lambda \frac{\partial \mathcal{L}}{\partial \theta}\bigg|_{\theta_t} \tag{3.7}$$

   where t is the step index. It is a powerful optimizer but is very sensible to local minima of the loss in the parameters phase space as illustrated in Figure 3.4a.

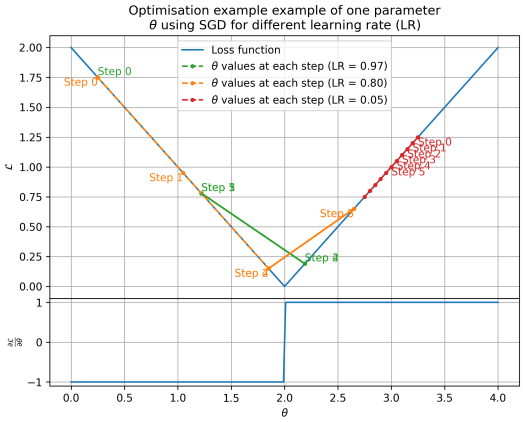
2. Adam Optimizer [38]. The concept is, in short, to have and SGD but with momentum. Adam possess two momentum $m(\beta_1)$ and $v(\beta_2)$ which are respectively proportional to $\frac{\partial \mathcal{L}}{\partial \theta}$ and $(\frac{\partial \mathcal{L}}{\partial \theta})^2$. β_1 and β_2 are hyperparameters that dictate the moment update at each optimization step. The parameters are then upgraded following

$$m_{t+1} = \beta_1 m_t + (1-\beta_1)\frac{\partial \mathcal{L}}{\partial \theta} \tag{3.8}$$

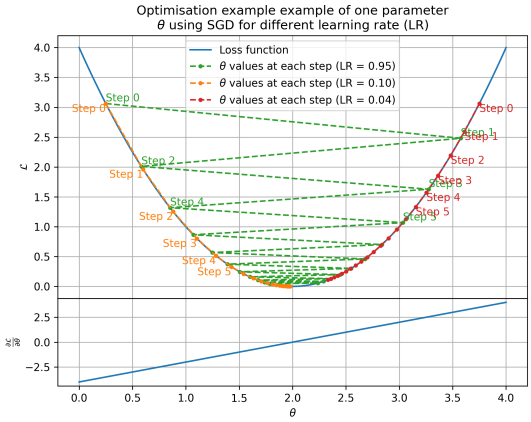
$$v_{t+1} = \beta_2 v_t + (1-\beta_2)\left(\frac{\partial \mathcal{L}}{\partial \theta}\right)^2 \tag{3.9}$$

$$\theta_{t+1} = \theta_t - \lambda \frac{m_{t+1}}{\sqrt{v_{t+1}} + \epsilon} \tag{3.10}$$

   where ϵ is a small number to prevent divergence when v is close to 0. These momentums allow to overcome small local minima in the parameters phase. Imagine ball going down a slope as illustrated in 3.4a, if you ignore the stored momentum you get SGD and get stuck as on the left plot. Now if you consider the momentum you get over the hill and end up in the global minima.



(A) Illustration of the SGD optimizer on one parameter θ on the MAE Loss. We see here that it has trouble reaching the minima due to the gradient being constant.



(B) Illustration of the SGD optimizer on one parameter θ on the MSE Loss. We see two different behavior: A smooth one (orange and red) when the LR is small enough and a more chaotic one when the LR is too high.

FIGURE 3.5 – Illustration of the SGD optimizer. In blue is the value of the loss function, orange, green and red are the path taken by the optimized parameter during the training for different LR.

Learning Rate (LR) Schedules

The learning rate plays a crucial role in determining how fast or slow the model converges. If the learning rate is too high (Fig. 3.5a), the model may skip over the optimal solution, whereas a low learning rate (Fig. 3.5b) can slow down the convergence process, leading to inefficient training. To address this, learning rate schedulers are employed.

Using a learning rate scheduler allows the optimizer to take larger steps in the early stages of training, where rapid learning is beneficial, and progressively smaller steps as the model approaches convergence. This strategy is especially useful in JUNO, where early learning from noisy data may require coarse adjustments, but fine-tuning is needed later to accurately capture subtle event characteristics.

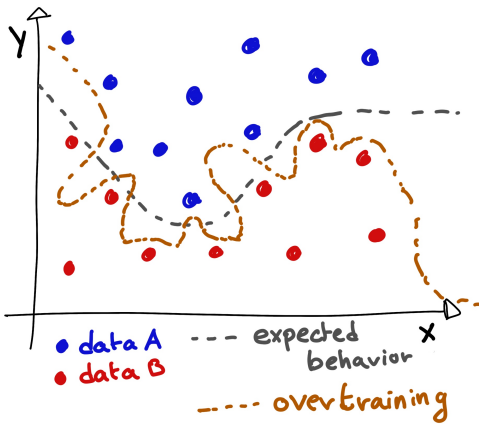
Another policy that is often use is the save of the best model. In some situation, the loss value after each epoch will strongly oscillate or can even worsen. This policy allow us to keep the best version of the model attained during the training phase.

3.1.4 Potential pitfalls

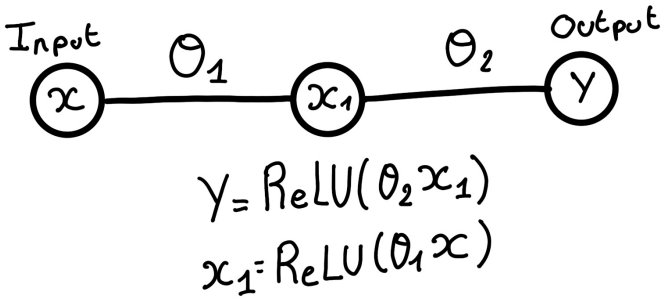
Apart from being stuck in local minima, there is also other behaviors and effects we want to prevent during training.

Overtraining

Overfitting occurs when a neural network memorizes specific details or noise from the training dataset rather than learning a general representation of the underlying data. This is common when the training dataset is small relative to the number of parameters in the network or when the dataset contains specific features that do not generalize well to unseen data. Additionally, training the



(A) Illustration of overtraining. The task at hand is to determine depending on two input variable x and y if the data belong to the dataset A or the dataset B. The expected boundary between the two dataset is represented in grey. A possible boundary learnt by overtraining is represented in brown.



(B) Illustration of a very simple NN

FIGURE 3.6

network for too many epochs can exacerbate this issue. Figure 3.6a illustrates the impact of overfitting, where the model fits the training data too closely, compromising its ability to generalize. To detect overfitting, techniques like monitoring the validation loss, early stopping, or employing cross-validation can be employed. In JUNO's context, managing overfitting is critical due to the large volume of data generated by the photomultiplier tubes (PMTs), which may include noise or other artifacts.

Overtraining can be fought in multiple ways, for example:

— **More data**. By having more data in the training dataset, the network will not be able the specificities of every data.

— **Less parameters**. By reducing the number of parameters, we reduce the computing and learning capacities of the network. This will force it to fallback to generalist behaviours.

— **Dropout**. This technique implies to randomly set some neurons to 0, i.e. cutting the relation between two neurons in a layer. By doing this, we force the network to allocate more of its parameter to the features learning, preventing those parameters to be used for overtraining.

— **Early stopping**. During the training we monitor the network performance over a validation dataset. The network does not train on this dataset and thus cannot learn its specificities. If the loss on the training dataset diverge too much from the loss on the validation dataset, we can stop the training earlier to prevent it from overtraining.

Gradient vanishing

Gradient vanishing is the effect of the gradient being so small for the early layers that the parameters are barely updated after each step. This cause the network to be unable to converge to the minima.

This comes from the way the gradient descent is calculated. Imagine a simple network composed of three fully connected layers: the input layer, a intermediate layer and the output layer. Let L be the loss, θ_1 the parameter between the input and the intermediate layer and θ_2 the parameter between the intermediate and output layer. This network is schematized in Figure 3.6b.

The gradient for θ_1 will be computed using the chain rule presented in equation 3.6. Because θ_1

depends on θ_2, if the gradient of θ_2 is small, so will be the gradient of θ_1. Now if we would have much more layer, we can see how the subsequent multiplication of small gradients would lead to very small update of the parameters thus "*vanishing gradient*".

Multiple actions can be taken to prevent this effect such as:

— **Batch normalization**: In this case we apply a normalization layer that will normalize the data. It means that we transform the input variable X into a variable D which distribution follow $\langle D \rangle = 0$ and $\sigma D = 1$. This helps the parameters of the network to maintain an appropriate scale.

— **Residual Network (ResNet)** [39]: Residual network is a technique for neural network in which, instead of just sequentially feeding the results of each layer to the next one, you compute a residual over the input data. This technique is illustrated in Figure 3.7. The reference [39] show empirical evidence of its relevance.

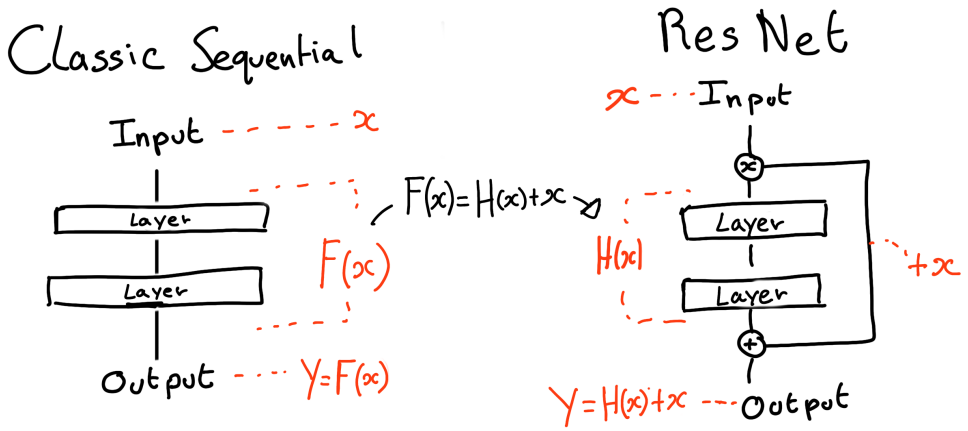


FIGURE 3.7 – Illustration of the ResNet framework

Gradient explosion

Gradient explosion occurs when gradients grow exponentially during backpropagation, causing parameter values to increase dramatically. This is particularly problematic in deep networks where the product of large gradients across layers can lead to unstable updates. In practice, gradient explosion is often caused by large learning rates, poor weight initialization, or nonlinearities in the network. For illustration, consider that the loss dependency in θ follow

$$\mathcal{L}(\theta) = \frac{\theta^2}{2} + e^{4\theta}$$
$$\frac{\partial \mathcal{L}}{\partial \theta} = \theta + 4e^{4\theta}$$

The explosion is illustrated in Figure 3.8 where we can see that the loss degrade with each step of optimization. In this illustration it is clear that reducing the learning rate suffice but this behaviour can happens in the middle of the training where the learning rate schedule does not permit reactivity.

There exist solutions to prevent this explosions:

— **Gradient clipping**: Is this case we work on the gradient so that the norm of gradient vector does not exceed a certain threshold. In our illustration in Figure 3.8 the gradient for $\theta > 0$ could be clipped at 3 for example.

— **Batch normalization**: For the same reasons as for gradient vanishing, normalizing the input data help reduce erratic behaviour.

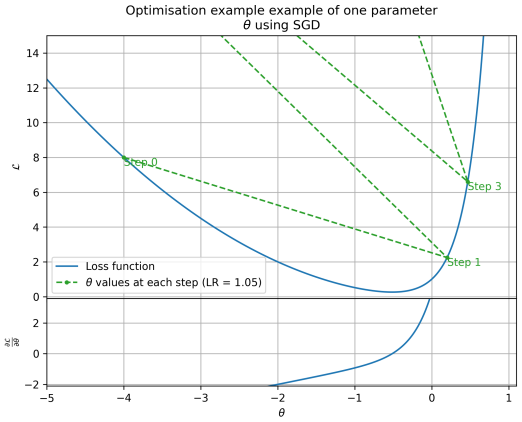


FIGURE 3.8 – Illustration of the gradient explosion. Here it can be solved with a lower learning rate but its not always the case.

3.2 Neural networks architectures

3.2.1 Fully Connected Deep Neural Network (FCDNN)

In this thesis, FCDNN serves as a baseline architecture for comparison with more specialized models like CNNs (see Section 3.2.2) and GNNs (see section 3.2.3), which are better suited to structured or graph-based data. However, FCDNNs are still useful when modeling highly abstract relationships, such as aggregating features from the JUNO PMTs. While they are powerful, their main drawback lies in their inefficiency when dealing with high-dimensional or spatially structured data, which will be addressed with convolutional architectures. This architecture is the stack of multiple fully connected layers as presented in the Figure 3.9a. Most of the time, the classic ReLU function

$$\text{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

is used as activation function. Prelu and Sigmoid are also popular choices:

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{3.12}$$

$$\text{PReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{otherwise} \end{cases} \tag{3.13}$$

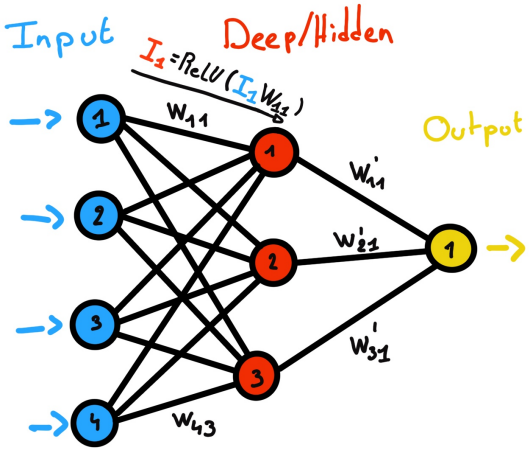
The reasoning behind ReLU and PReLU is that with enough of them, you can mimic any continuous function as illustrated in Figure 3.9b. Sigmoid is more used in case of classification, its behavior going hand in hand with the Cross Entropy loss function used in classification problems.

Due to its simplicity, FCDNN are also used as basic pieces for more complex architectures such as the CNN and GNN that will be presented in the next sections.

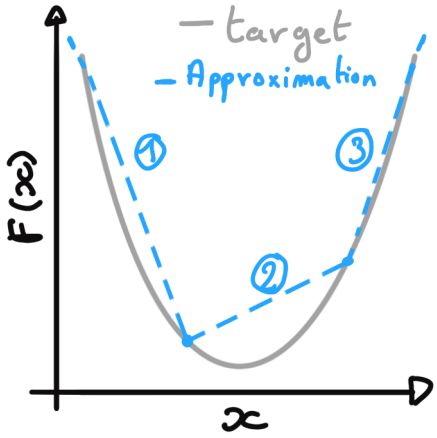
3.2.2 Convolutional Neural Network (CNN)

It's not trivial to describe in text the principles of Convolutional Neural Network (CNN) and how they works. We try a general description below followed by a step by step description of a concrete example.

Convolutional Neural Networks are a family of neural networks that use discrete convolution filters,



(A) Schema of a FCDNN



(B) Illustration of a composition of ReLU "approximating" a function. (1) No ReLU is taking effect (2) One ReLU is activating (3) Another ReLU is activating

FIGURE 3.9

as illustrated in an example in Figure 3.10, to process the input data, often images. They are commonly used in image recognition [40] for classification or regression problematics. Concretely, you multiply element-wise a portion of the input data, in the case of an image, a small part of the image, with a kernel of same dimension. In Figure 3.10, we multiply the 3×3 pixels sub-image with the 3×3 kernel.

Their filters scan the input data, highlighting patterns of interest, this scanning procedure making them translation-invariant. In the concrete case of Figure 3.10, for each pixel of the input image, we group it with the 8 neighbours pixel and produce a new pixel that correspond to the output image. For the pixel on the edges that do not have neighbours, we either create "imaginary" pixel with the value 0 or we just ignore them. If we ignore them, the output image will posses fewer pixels than the input image. We see that the operation do not care where is the pattern of interest in the images, the filter output will be *invariant* whatever *translation* is applied to the image.

This invariance mean that they are capable of detecting oriented features independently of their location on the image. These filters scan the input, highlighting important features like edges or textures, which in JUNO's case could represent spatial correlations in the timing and charge data across the detector. As the network goes deeper, it can capture more complex and abstract features, making it ideal for detecting nuanced particle interactions. Again taking 3.10 as an example, with only the 9 parameters composing the kernel, we can highlight the contour of the duck by looking at the "yellowness" of the pixels.

The learning parameters of CNNs are the kernels components, the network thus learn the optimal filters to extract the desired features.

The convolution layers are commonly chained [41], reducing the input dimension while increasing the number of filters. The idea behind is that the first layers will process local informations and the latest layers will process more global informations, as the latest convolution filters will process the results of the preceding that themself have processed local information. To try to preserve the amount of information, we tend to grow the numbers of filters for each division of the input data. The results of the convolution filters is commonly then flattened and feed to a smaller FCDNN which will process the filters results to yield the desired output.

As an example, let's take the Pytorch [43] example for the MNIST [44], a dataset of black and white images of handwritten digits. Those images are 28×28 pixels with only one channel corresponding

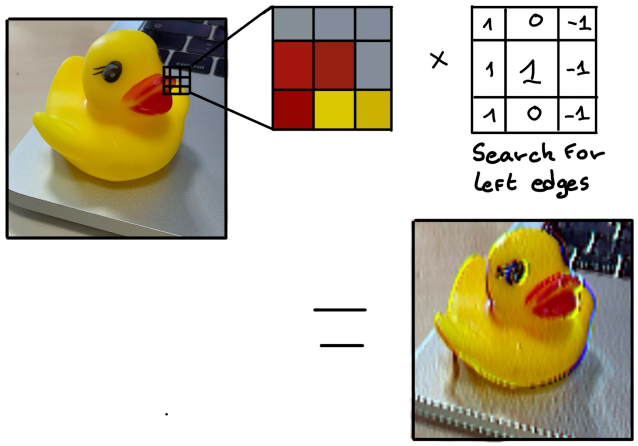


FIGURE 3.10 – Illustration of the effect of a convolution filter. Here we apply a filter with the aim do detect left edges. We see in the resulting image that the left edges of the duck are bright yellow where the right edges are dark blue indicating the contour of the object. The convolution was calculated using [42].

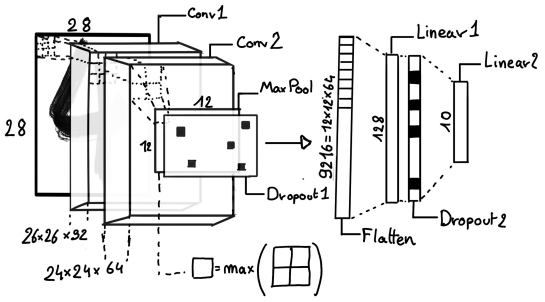
to the grey level of the pixel. Example of images from this dataset are presented in Figure 3.11a

A schema of the CNN used in the Pytorch example is presented in Figure 3.11b. Using this schema as a reference, the trained network is made of:

1. A convolutional layer of (3×3) filters yielding 32 channels. A bias parameter is applied to each channel for a total of $(32 \cdot (3 \times 3) + 32) = 320$ parameters. The resulting image is $(26 \times 26 \times 32)$ (26 per 26 pixels with 32 channels). The ReLU activation function is applied to each pixel.
2. A second convolutional layer of (3×3) filters yielding 64 channels. This channel also posses a bias parameter for a total of $(64 \cdot (3 \times 3) + 64) = 640$ parameters. Resulting image is $(24 \times 24 \times 64)$. This channel also apply a ReLU activation function.
3. Then comes a (2×2) max pool layer with a stride of 1 meaning that for each channel the max value of pixels in a (2×2) block is condensed in a single resulting pixel. The resulting image is $(12 \times 12 \times 64)$.
4. This image goes through a dropout layer which will set the pixel to 0 with a probability of 0.25. This help prevent overtraining the neural network (see Section 3.1.4 for more details).
5. The data is the flattened i.e. condensed into a vector of $(12 \times 12 \times 64) = 9216$ values.
6. Then comes a fully connected linear layer (Eq. 3.2) with a ReLU activation that output 128 feature. It needs $(9216 \cdot 128) + 128 = 1'179'776$ parameters.
7. This 128 item vector goes through another dropout layer with a probability of 0.5
8. The vector is then transformed through a linear layer with ReLU activation. It output 10 values, one for each digit class (0, 1, 2, ..., 9). It need $(128 \cdot 10) + 128 = 1408$ parameters.
9. Finally the 10 values are normalized using a log softmax function $\text{LogSoftmax}(x_i) = \log\left(\frac{\exp(x_i)}{\sum_j \exp(x_j)}\right)$. Each of those values are the probability of the input image to be a certain digit.

The final network needs 1'182'144 parameters or, if we consider each parameters to be a double precision floating point, 9.45 MB of data. To gives a order of magnitude, such neural network is considered "simple", train in a matter of minutes on T4 GPU [45] (14 epochs) and reach an accuracy in its prediction of 99%.

(A) Example of images in the MNIST dataset



(B) Schema of the CNN used in Pytorch example to process the MNIST dataset

FIGURE 3.11

3.2.3 Graph Neural Network (GNN)

In GNNs, data is represented as nodes and edges in a graph, which allows us to model the JUNO detector as a network of PMTs, where each PMT is a node and the edges represent relationships such as spatial distance or timing correlations between PMTs. This flexibility enables GNNs to capture complex interactions across the detector geometry that would be difficult to represent with a CNN. Furthermore, GNNs excel at processing non-Euclidean data, making them a natural fit for the irregular layout of the PMTs in JUNO. In this thesis, GNNs are applied to model the spatial and temporal relationships between PMTs, enabling more precise event classification and reconstruction. By leveraging the message-passing framework, the GNN can aggregate information from neighboring PMTs, allowing it to detect subtle patterns in the detector's data.

To get deeper in details, we have seen in the previous section, the CNNs are powerful for image processing, and more generally any data that can be expressed as a regular, discrete space and from which the information reside in the dispersion in this space. For an image, the edges of an object and how they assemble. A red square, straight edges with a sharp angle between them, is much less representative of a duck than an yellow sphere, round edges without sharp angles.

This "image" projection is not fitted for every problematics. The signals produced by a detector does not always have the properties of images. In the case of JUNO for example, we can create an image of two channels, one for the charge Q and one for the timing t but this image should be spheric. Furthermore JUNO is by nature inhomogeneous, using two different systems : The LPMT and the SPMT. Those two systems have different regime, and thus should be processed differently. We could imagine images with four channels, two for the LPMT and two for the SPMT, or even a branched CNN with one convolution branch for the LPMT and another one for the SPMT. Anyway, the CNN will need to combine the two systems.

To get around the restrictions of data representation imposed by CNNs, we can use the more flexible *graph* representation. A graph $G(\mathcal{N}, \mathcal{E})$ is composed of vertex or node $n \in \mathcal{N}$ and edges $e \in \mathcal{E}$. The edges are associated to two nodes $(u, v) \in \mathcal{N}^2$, "connecting" them. The node and the edges can hold features, commonly represented as vector $n \in \mathbb{R}^{k_n}$, $e \in \mathbb{R}^{k_e}$ with k_n and k_e the number of features on the nodes and edges respectively. We can thus define a graph using two tensors A_e^{ij} the adjacency tensor that hold the features $\epsilon \in [0, k_e]$ of the edge connecting the node i and j and the tensor N_ν^i that hold the features $\nu \in [0, k_n]$ of a node i.

More figuratively, using the example in Figure 3.12, we have a graph of 5 nodes with a color as feature. The edges have no features, we thus encode their existences as 0 or 1. In a realistic examples as JUNO we could represent each PMTs as nodes and the edges between them as their relation such

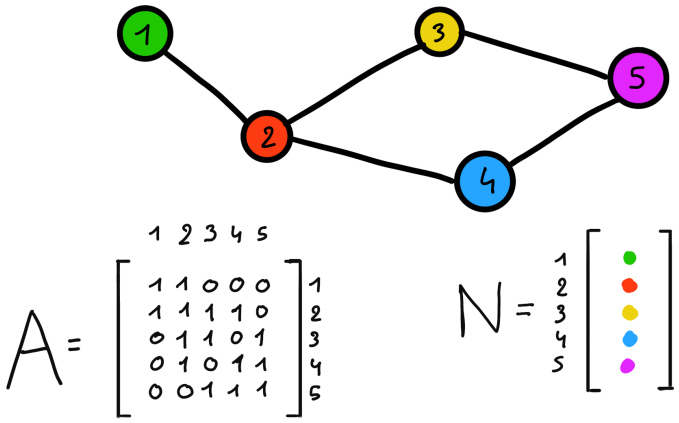


FIGURE 3.12 – Illustration of a graph and its tensor representation.

as distance, timing difference, etc... There no strict rules about what is a node or how they should be linked together. This abstraction allow us to represent virtually any type of detector of any geometry.

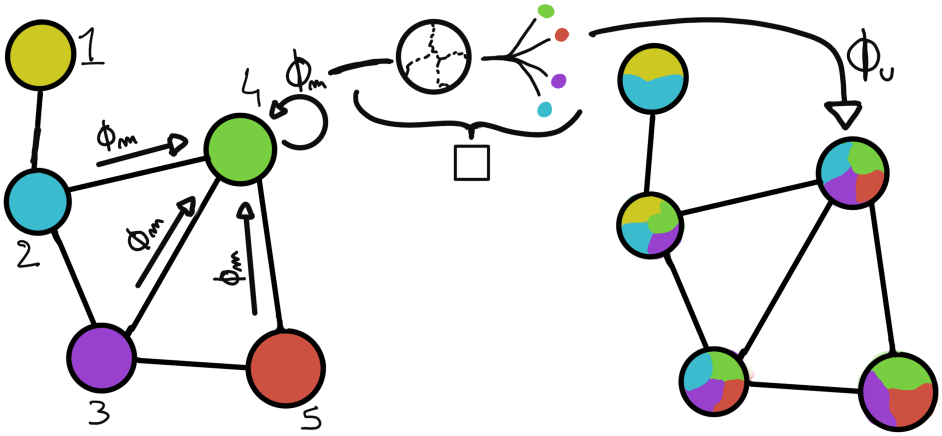


FIGURE 3.13 – Illustration of the message passing algorithm. The detailed explanation can be found in Section 3.2.3

To process such object we need specific machine learning algorithms we call Graph neural network. To efficiently manipulate graph we need to structurally encode their property in the neural network computing architecture: each node is equivalent (as opposite to ordered data in a vector), each node has a set of neighbours, ... One of this method is the message passing algorithm presented historically in "Neural Message Passing for Quantum Chemistry" [46]. In this algorithm, with each layer of message passing a new set of features is computed for each node following

$$n_i^{k+1} = \phi_u(n_i^k, \square_j \phi_m(n_i^k, n_j^k, e_{ij}^k)); \; n_j \in \mathcal{N}_i' \tag{3.14}$$

where ϕ_u is a differentiable *update* function, $\square_j$ is a differentiable *aggregation* function and ϕ_m is a differentiable *message* function. $\mathcal{N}_i' = \{n_j \in \mathcal{N} | (n_i, n_j) \in \mathcal{E}\}$ is the set of neighbours of n_i, i.e. the nodes n_j from which it exist an edge $e_{i,j} \rightarrow (n_i, n_j)$. k is the layer on which the message passing algorithm is applied. The update function need also a few other property if we want to keep the graph property, most notably the permutational invariance of its parameters (example: mean, std, sum, ...). The differents message, update and aggregation functions can really be any kind of function

if they follow the constraint presented before, even small Neural Network.

The edges features can also be updated, either by directly taking the results of ϕ_m or by using another message function ϕ_e.

To explain this process, let's take the situation presented in Figure 3.13. We start with an input graph on left, in this case the message passing algorithm is mixing the color on each nodes and produce nodes of mixed color. For simplicity, the ϕ_m and ϕ_u function are the identity, they take a color and output the same color.

Let's look at what's happening in the node 4. It has 3 neighbours and is a neighbour of itself. The four resulting ϕ_m extract the color of each nodes and then feed them to the $\square$ function. The $\square$ function just equally distribute the color in the node. Finally the ϕ_u function just update the node with the output of $\square$.

Interestingly we see that the new node 4 does not have any yellow, the color of node 1. But if we were to run the message passing algorithm again, it would get some as node 2 is now partially yellow. If color here represent information, we see that multiple step are needed so that each node is "aware" of the informations the other nodes possess.

Message passing is a very generic way of describing the process of GNN and it can be specialized for convolutional filtering [47], diffusion [48] and many other specific operation. GNN are used in a wide variety of application such as regression problematics, node classification, edge classification, node and edge prediction, ...

It is a very versatile but complex tool.

3.2.4 Adversarial Neural Network (ANN)

The adversarial machine learning, Adversarial Neural Networks (ANN) in the case of neural network, is a family of unsupervised machine learning algorithms where the learning algorithm (generator) is competing against another algorithm (discriminator). Taking the example of Generative Adversarial Networks, concept initially developed by Goodfellow et al. [49], the discriminator goal is to discriminate between data coming from a reference dataset and data produced by the generator. The generator goal, on the other hand, is to produce data that the discriminator would not be able to differentiate from data from the reference dataset. The expression of duality between the two models is represented in the loss where, at least a part of it, is driven by the results of the discriminator.

3.3 State of the art of the Offline IBD reconstruction in JUNO

The main reconstruction method currently run in JUNO is OMILREC, a data-driven method based on a likelihood maximization [50, 51] using only the LPMTs. The first step is to reconstruct the interaction vertex from which the energy reconstruction is dependent. It is also necessary for event pairing and classification.

3.3.1 Interaction vertex reconstruction

To start the likelihood maximization, a rough estimation of the vertex and of the event timing is needed. We start by estimating the vertex position using a charge based algorithm.

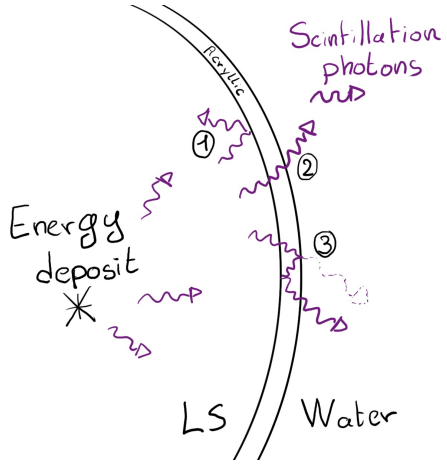
Charge based algorithm

The charge-based algorithm is basically base on the charge-weighted average of the PMT position.

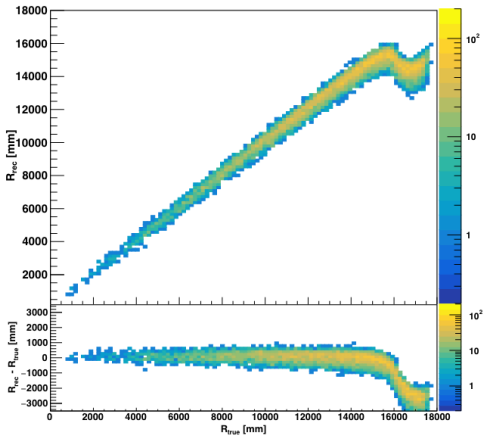
$$\vec{r}_{cb} = a \cdot \frac{\sum_i q_i \cdot \vec{r}_i}{\sum_i q_i} \tag{3.15}$$

Where q_i is the reconstructed charge of the pulse of the ith PMT and $\vec{r}_i$ is its position. $\vec{r}_0$ is the reconstructed interaction position. a is a scale factor introduced because a weighted average over a 3D sphere is inherently biased. Using calibration we can estimate $a \approx 1.3$ [52]. The results in Figure 3.14b shows that the reconstruction is biased from around 15m and further. This is due to the phenomena called "total reflection area" or TR Area.

As depicted in the Figure 3.14a the optical photons, given that they have a sufficiently large incidence angle, can be deviated of their trajectories when passing through the interfaces LS-acrylic and water-acrylic due to the optical index difference. This cause photons to be lost or to be detected by PMT further than anticipated if we consider their rectilinear trajectories. This cause the charge barycenter the be located closer to the center than the event really is.



(A) Illustration of the different optical photons reflection scenarios. **1** is the reflection of the photon at the interface LS-acrylic or acrylic-water. **2** is the transmission of the photons through the interfaces. **3** is the conduction of the photon in the acrylic.



(B) Heatmap of R_{rec} and $R_{rec} - R_{true}$ as a function of R_{true} for 4MeV prompt signals uniformly distributed in the detector calculated by the charge based algorithm

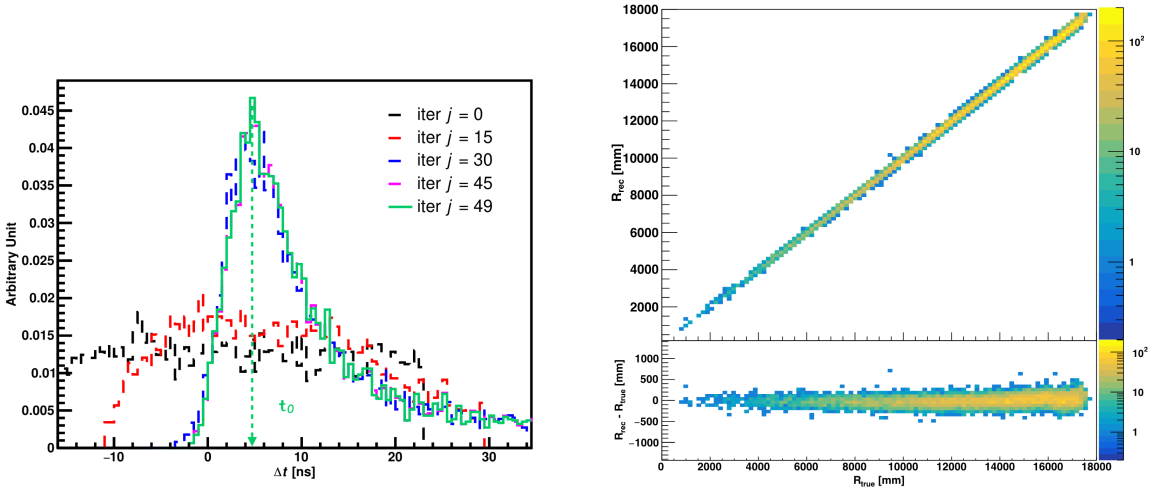
FIGURE 3.14

It is to be noted that charge based algorithm, in addition to be biased near the edge of the detector, does not provide any information about the timing of the event. Therefore, a time based algorithm needs to be introduced to provide initial values.

Time based algorithm

The time based algorithm use the distribution of the time of flight corrections Δt (Eq 3.16) of an event to reconstruct its vertex and t_0. It follow the following iterations:

1. Use the charge based algorithm to get an initial vertex to start the iteration.



(A) Δt distribution at different iterations step j

(B) Heatmap of R_{rec} and $R_{rec} - R_{true}$ as a function of R_{true} for 4MeV prompt signals uniformly distributed in the detector calculated by the time based algorithm

FIGURE 3.15

2. Calculate the time of flight correction for the ith PMT using

$$\Delta t_i(j) = t_i - \text{tof}_i(j) \tag{3.16}$$

where j is the iteration step, t_i is the timing of the ith PMT, and tof_i is the time-of-flight of the photon considering an rectilinear trajectory and an effective velocity in the LS and water (see [52] for detailed description of this effective velocity). Plot the Δt distribution and label the peak position as Δt^{peak} (see fig 3.15a).

3. Calculate a correction vector $\vec{\delta}[\vec{r}(j)]$ as

$$\vec{\delta}[\vec{r}(j)] = \frac{\sum_i \left(\frac{\Delta t(j) - \Delta t^{\text{peak}}(j)}{\text{tof}_i(j)} \right) \cdot (\vec{r}_0(j) - \vec{r}_i)}{N^{\text{peak}}(j)} \tag{3.17}$$

where $\vec{r}_0$ is the vertex position at the beginning of this iteration, $\vec{r}_i$ is the position of the ith PMT. To minimize the effect of scattering, dark noise and reflection, only the pulse happening in a time window (-10 ns, +5 ns) around Δt^{peak} are considered. N^{peak} is the number of PE collected in this time-window.

4. if $\vec{\delta}[\vec{r}(j)] < 1$mm or $j \geq 100$, stop the iteration. Otherwise $\vec{r}_0(j+1) = \vec{r}_0(j) + \vec{\delta}[\vec{r}(j)]$ and go to step 2.

However because the earliest arrival time is used, t_i is related to the number photoelectrons N_i^{pe} detected by the PMT [53–55]. To reduce bias in the vertex reconstruction, the following equation is used to correct t_i into t_i':

$$t_i' = t_i - p_0/\sqrt{N_i^{\text{pe}}} - p_1 - p_2/N_i^{\text{pe}} \tag{3.18}$$

The parameters (p_0, p_1, p_2) were optimized to (9.42, 0.74, -4.60) for Hamamatsu PMTs and (41.31, -12.04, -20.02) for NNVT PMTs [52]. The results presented in Figure 3.15b shows that the time based algorithm provide a more accurate vertex and is unbiased even in the TR area. This results $(\vec{r}_0, t_0)$ is used as initial value for the likelihood algorithm.

Time likelihood algorithm

The time likelihood algorithm use the residual time expressed as follow

$$t^i_{\text{res}}(\vec{r}_0, t_0) = t_i - \text{tof}_i - t_0 \tag{3.19}$$

In a first order approximation, the scintillator time response Probability Density Function (PDF) can be described as the emission time profile of the scintillation photons, the Time Transit Spread (TTS) and the dark noise of the PMTs. The emission time profile $f(t_{\text{res}})$ is described like

$$f(t_{\text{res}}) = \sum_k \frac{\rho_k}{\tau_k} e^{\frac{-t_{\text{res}}}{\tau_k}}, \ \sum_k \rho_k = 1 \tag{3.20}$$

as the sum of the k component that emit light in the LS each one characterised by it's decay time τ_k and intensity fraction ρ_k. The TTS component is expressed as a gaussian convolution

$$g(t_{\text{res}}) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(t_{\text{res}}-\nu)^2}{2\sigma^2}} \cdot f(t_{\text{res}}) \tag{3.21}$$

where σ is the TTS of PMTs and ν is the average transit time. The dark noise is not correlated with any physical events and considered as constant rate over the time window considered T. By normalizing the dark noise probability $\epsilon(t_{\text{res}})$ as $\int_T \epsilon(t_{\text{res}}) dt_{\text{res}} = \epsilon_{\text{dn}}$, it can be integrated in the PDF as

$$p(t_{\text{res}}) = (1 - \epsilon_{\text{dn}}) \cdot g(t_{\text{res}}) + \epsilon(t_{\text{res}}) \tag{3.22}$$

The distribution of the residual time t_{res} of an event can then be compared to $p(t_{\text{res}})$ and the best fitting vertex $\vec{r}_0$ and t_0 can be chosen by minimizing

$$\mathcal{L}(\vec{r}_0, t_0) = -\ln\left(\prod_i p(t^i_{\text{res}})\right) \tag{3.23}$$

The parameter of Eq. 3.22 can be measured experimentally. The results shown in Figure 3.16 used PDF from monte carlo simulation. The results shows that $R_{rec} - R_{true}$ is biased depending on the energy. While this could be corrected using calibration, another algorithm based on charge likelihood was developed to correct this problem.

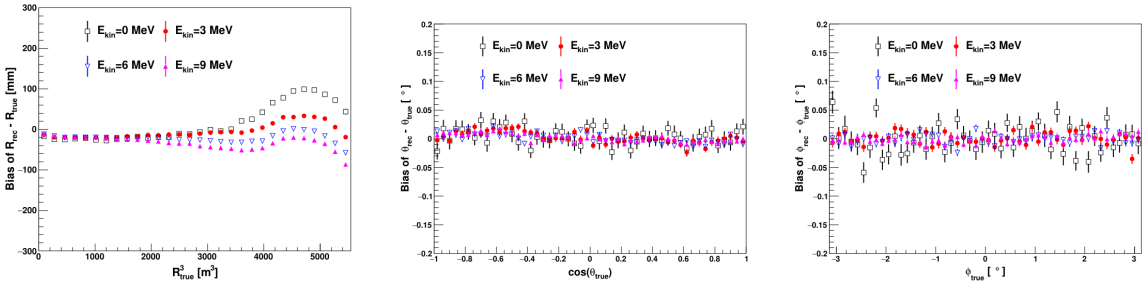


FIGURE 3.16 – Bias of the reconstructed radius R (left), θ (middle) and ϕ (right) for multiple energies by the time likelihood algorithm

Charge likelihood algorithm

Similarly to the time likelihood algorithms that use a timing PDF, the charge likelihood algorithm use a PE PDF for each PMT depending on the energy and position of the event. With $\mu(\vec{r}_0, E)$ the mean expected number of PE detected by each PMT, the probability to observe N_{pe} in a PMT follow a Poisson distribution. Thus

— The probability to observe no hit ($N_{pe} = 0$) in the jth PMT is $P^j_{nohit}(\vec{r}_0, E) = e^{-\mu_j}$

— The probability to observe $N_{pe} \neq 0$ in the ith PMT is $P^i_{hit}(\vec{r}_0, E) = \frac{\mu^{N^i_{pe}} e^{-\mu_i}}{N^i_{pe}!}$

Therefore, the probability to observe a specific hit pattern can be expressed as

$$P(\vec{r}_0, E) = \prod_j P^j_{nohit}(\vec{r}_0, E) \cdot \prod_i P^i_{hit}(\vec{r}_0, E) \tag{3.24}$$

The best fit values of $\vec{R}_0$ and E can then be calculated by minimizing the negative log-likelihood

$$\mathcal{L}(\vec{r}_0, E) = -\ln(P(\vec{r}_0, E)) \tag{3.25}$$

In principle, $\mu_i(\vec{r}_0, E)$ could be expressed

$$\mu_i(\vec{r}_0, E) = Y \cdot \frac{\Omega(\vec{r}_0, r_i)}{4\pi} \cdot \epsilon_i \cdot f(\theta_i) \cdot e^{-\sum_m \frac{d_m}{\zeta_m}} \cdot E + \delta_i \tag{3.26}$$

where Y is the energy scale factor, $\Omega(\vec{r}_0, r_i)$ is the solid angle of the ith PMT, ϵ_i is its detection efficiency, $f(\theta_i)$ its angular response, ζ_m is the attenuation length in the materials and δ_i the expected number of dark noise.

However Eq. 3.26 assume that the scintillation light yield is linear with energy and describe poorly the contribution of indirect light, shadow effect due to the supporting structure and the total reflection effects. The solution is to use data driven methods to produce the pdf by using the calibrations sources and position described in Section 2.4. In the results presented in Figures 3.17, the PDF was produced using MC simulation and 29 specific calibrations position [52] along the Z-axis of the detector. We see that the charge likelihood algorithm show little bias in the TR area and a better

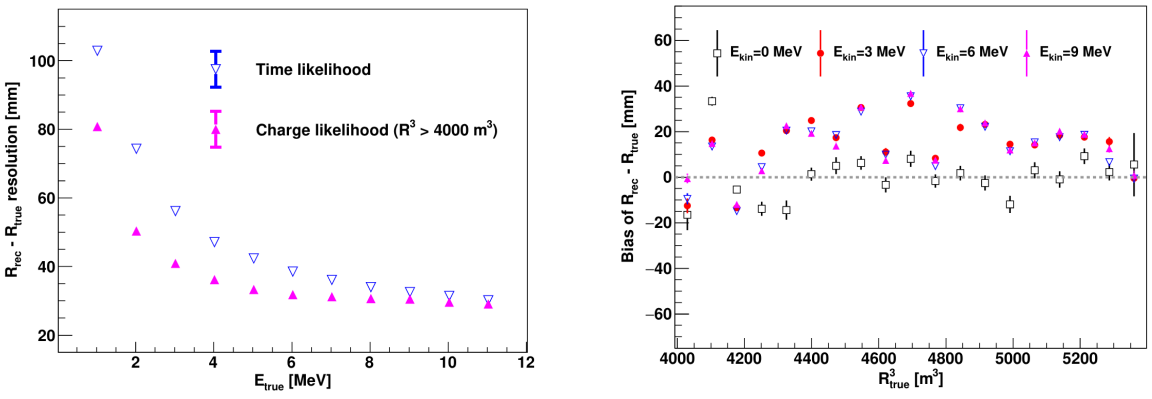


FIGURE 3.17 – **On the left:** Resolution of the reconstructed R as a function of the energy in the TR area ($R^3 > 4000\text{m}^3 \equiv R > 16m$) by the charge and time likelihood algorithms. **On the right:** Bias of the reconstructed R in the TR area for different energies by the charge likelihood algorithm

resolution than the time likelihood. The Figure 3.18 shows the radial resolution of the different

algorithm presented for this section, we can see the refinement at each step and that the charge likelihood yield the best results.

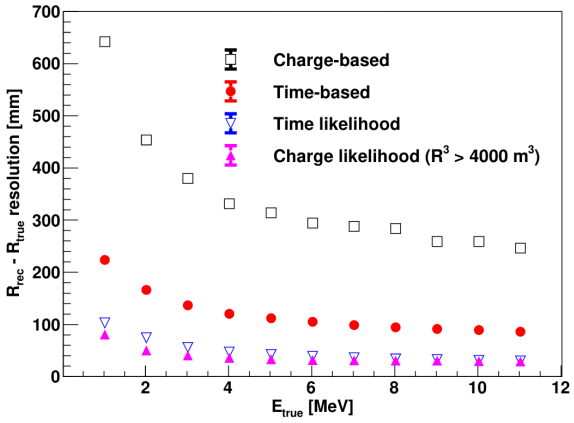
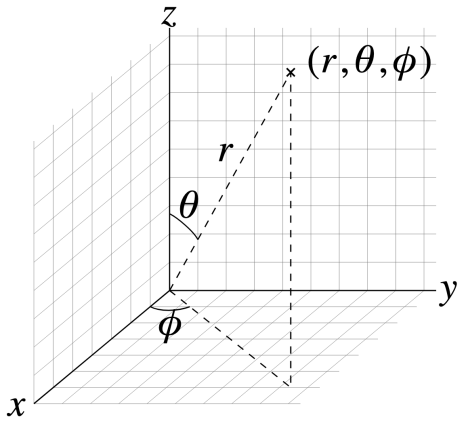


FIGURE 3.18 – Radial resolution of the different vertex reconstruction algorithms as a function of the energy

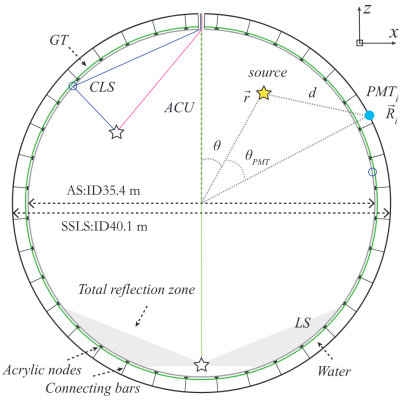
The charge based likelihood algorithms already give use some information on the energy as Eq. 3.25 is minimized but the energy can be further refined as shown in the next section.

3.3.2 Energy reconstruction

As explained in Section 2.1.1, energy resolution is crucial for the NMO and oscillation parameters measurements. Thus the energy reconstruction algorithm should take into consideration as much detector effect as possible. The following method is a data driven method based on calibration samples inspired by the charge likelihood algorithm described above [56].



(A) Spherical coordinate system used in JUNO for reconstruction



(B) Definition of the variables used in the energy reconstruction

FIGURE 3.19

Charge estimation

The most important element in the energy reconstruction is $\mu_i(\vec{r}_0, E)$ described in Eq. 3.26. For realistic cases, we also need to take into account the electronics effect that were omitted in the previous section. Those effect will cause a charge smearing due to the uncertainties in the N_{pe} reconstruction. Thus we define $\hat{\mu}^L(\vec{r}_0, E)$ which is the expected N_{pe}/E in the whole detector for an event with visible energy E_{vis} and position $\vec{r}_0$. The position of the event and PMTs are now defined using $(r, \theta, \theta_{pmt})$ as defined in Figure 3.19b.

$$\hat{\mu}(r, \theta, \theta_{pmt}, E_{vis}) = \frac{1}{E_{vis}} \frac{1}{M} \sum_i^M \frac{\frac{\bar{q}_i}{\hat{Q}_i} - \mu_i^D}{\mathrm{DE}_i}, \ \mu_i^D = \mathrm{DNR}_i \cdot L \tag{3.27}$$

where i runs over the PMTs with the same θ_{pmt}, DE_i is the detection efficiency of the ith PMT. μ_i^D is the expected number of dark noise photoelectrons in the time window L. The time window have been optimized to $L = 280$ ns [56]. $\bar{q}_i$ is the average recorded photoelectrons in the time window and $\hat{Q}_i$ is the expected average charge for 1 photoelectron. The N_{pe} map is constructed following the procedure described in [51].

Time estimation

The second important observable is the hit time of photons that was previously defined in Eq. 3.19. It is here refined as

$$t_r = t_h - \mathrm{tof} - t_0 = t_{LS} + t_{TT} \tag{3.28}$$

where t_h is the time of hit, t_{LS} is the scintillation time and t_{TT} the transit time of PMTs that is described by a gaussian

$$t_{TT} = \mathcal{N}(\overline{\mu_{TT} + t_d}, \sigma_{TT}) \tag{3.29}$$

where μ_{TT} is the mean transit time in PMTs, σ_{TT} is the Transit Time Spread (TTS) of the PMTs and t_d is the delay time in the electronics. The effective refraction index of the LS is also corrected to take into account the propagation distance in the detector.

The timing PDF $P_T(t_r|r, d, \mu_l, \mu_d, k)$ can now be generated using calibration sources [56]. This PDF describe the probability that the residual time of the first photon hit is in $[t_r, t_r + \delta]$ with r the radius of the event vertex, $d = |\vec{r} - \vec{r}_{PMT}|$ the propagation distance, μ_l and μ_d the expected number of PE and dark noise in the electronic reading window and k is the detected number of PE.

Now let denote $f(t, r, d)$ the probability density function of "photoelectron hit a time t" for an event happening at r where the photons traveled the distance d in the LS

$$F(t, r, d) = \int_t^L f(t', r, d) dt' \tag{3.30}$$

Based on the PDF for one photon $k = 1$, one can define

$$P_T^l(t|k = n) = I_n^l [f_l(t) F_l^{n-1}(t)] \tag{3.31}$$

where the indicator l means that the photons comes from the LS and I_n^l a normalisation factor. To this pdf we add the probability to have photons coming from the dark noise indicated by the indicator d using

$$f_d(t) = 1/L, \ F_d(t) = 1 - \frac{t}{L} \tag{3.32}$$

and so for the case where only one photon is detected by the PMT ($k = 1$)

$$P_T(t|\mu_l, \mu_d, k = 1) = I_1[P(1, \mu_l)P(0, \mu_d)f_l(t) + P(0, \mu_l)P(1, \mu_d)f_d(t)] \tag{3.33}$$

where $P(k_\alpha, \mu_\alpha)$ is the Poisson probability to detect k_α PE from $\alpha \in \{l, d\}$ with the condition $k_l + k_d = k$.

Now that we have the individual timing and charge probability we can construct the charge likelihood referred as QMLE:

$$\mathcal{L}(q_1, q_2, ..., q_N|\vec{r}, E_{vis}) = \prod_{j\in\text{unfired}} e^{-\mu_j} \prod_{i\in\text{fired}} \left(\sum_{k=1} P_Q(q_i|k) \cdot P(k, \mu_i) \right) \tag{3.34}$$

where $\mu_i = E_{vis}\hat{\mu_i^L} + \mu_i^D$ and $P(k, \mu_i)$ is the Poisson probability of observing k PE. $P_Q(q_i|k)$ is the charge pdf for k PE. And we can also construct the time likelihood referred as TMLE:

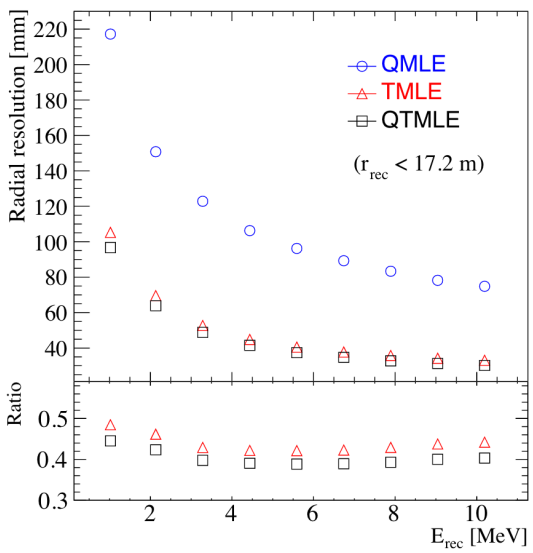
$$\mathcal{L}(t_{1,r}, t_{2,r}, ..., t_{N,r}|\vec{r}, t_0) = \prod_{i\in\text{hit}} \frac{\sum_{k=1}^{K} P_T(t_{i,r}|r, d, \mu_i^l, \mu_i^d, k) \cdot P(k, \mu_i^l + \mu_i^d)}{\sum_{k=1}^{K} P(k, \mu_i^l + \mu_i^d)} \tag{3.35}$$

where K is cut to 20 PE and hit is the set of hits satisfying $-100 < t_{i,r} < 500$ ns.

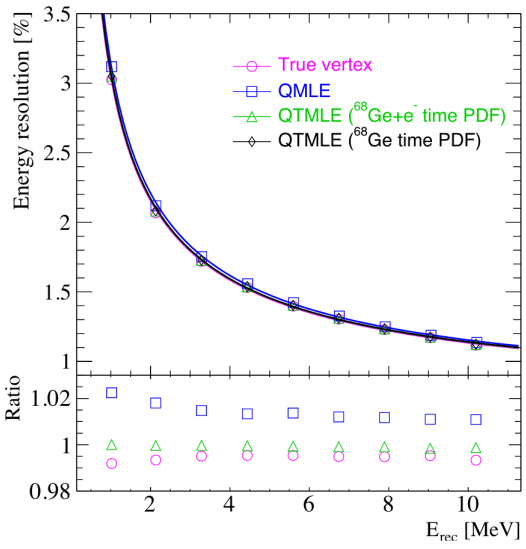
Merging those two likelihood give the charge-time likelihood QTMLE, the core algorithm of OMILREC.

$$\mathcal{L}(q_1, q_2, ..., q_N; t_{1,r}, t_{2,r}, ..., t_{N,r}|\vec{r}, t_0, E_{vis}) = \mathcal{L}(q_1, q_2, ..., q_N|\vec{r}, E_{vis}) \cdot \mathcal{L}(t_{1,r}, t_{2,r}, ..., t_{N,r}|\vec{r}, t_0) \tag{3.36}$$

The radial and energy resolutions of the different likelihood are presented in Figure 3.20 (from [56]). We can see the improvement of adding the time information to the vertex reconstruction and that an increase in vertex precision can bring improvement in the energy resolution, especially at low energies.



(A) Radial resolutions of the likelihood-based algorithm TMLE, QMLE and QTMLE



(B) Energy resolution of QMLE and QTMLE using different vertex resolutions

FIGURE 3.20

Data driven methods prove to be performant in the energy and vertex reconstruction given that we have enough calibrations sources to produce the PDF. In addition to this, member of JUNO have developed ML algorithms for reconstruction. The one focused on IBD reconstruction are presented in the next section.

3.3.3 Machine learning for reconstruction

The power of ML is the ability to model complex response to a specific problem. In JUNO the reconstruction problematic can be expressed as follow: knowing that each PMT, large or small, detected a given number of PE Q at a given time t and their position is x, y, z where did the energy was deposited and how much energy was it, modeling a function that naively goes:

$$\mathbb{R}^{5\times N_{pmt}} \longmapsto \mathbb{R}^4 \tag{3.37}$$

It is worth pointing that while this is already a lot in informations, this is not the rawest representation of the experiment. We could indeed replace the charge and time by the waveform in the time window of the event but that would lead to an input representation size that would exceed our computational limits. Also, due to those computational limits, most of the ML algorithm reduce this input phase space either by structurally encoding the information (pictures, graph), by aggregating it (mean, variance, ...) or by exploiting invariance and equivariance of the experiment (rotational invariance due to the sphericity, ...).

For machine learning to converge to performant algorithm, a large dataset exploring all the phase space of interest is needed. For the following studies, data from the monte carlo simulation presented in Section 2.6 are used for training. When the detector will be finished calibrations sources will be complementarily be used.

Boosted Decision Tree (BDT)

On of the most classic ML method used in physics in last years is the Boosted Decision Tree. They have been explored for vertex reconstruction [57] et for energy reconstruction [57, 58].

For vertex and energy reconstruction a BDT was developed using the aggregated informations presented in 3.1.

Parameter	description
$nHits$	Total number of hits
$x_{cc}, y_{cc}, z_{cc}, R_{cc}$	Coordinates of the center of charge
ht_{mean}, ht_{std}	Hit time mean and standard deviation

TABLE 3.1 – Features used by the BDT for vertex reconstruction

Its reconstruction performances are presented in Figure 3.22.

A second and more advanced BDT, subsequently named BDTE, that only reconstruct energy use a different set of features [58]. They are presented in the table 3.2

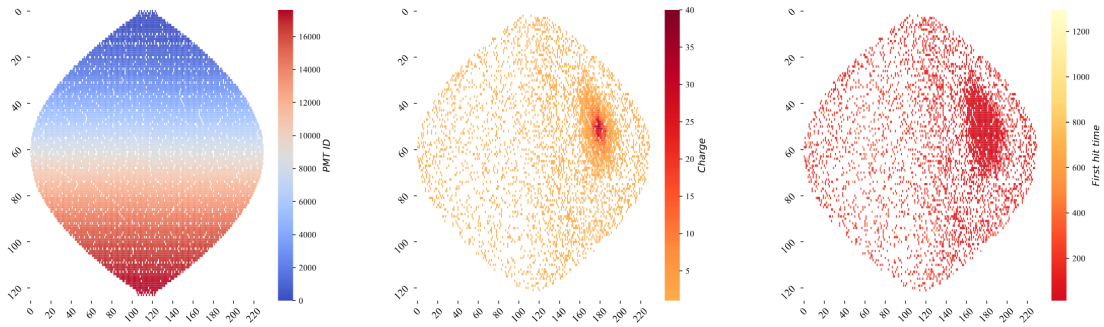
Neural Network (NN)

Three type of neural networks have explored for event reconstruction in JUNO Deep Neural Network (DNN), Convolutional Neural Network (CNN) and Graph Network (GNN).

AccumCharge	$ht_{5\%-2\%}$
R_{cht}	pe_{mean}
z_{cc}	J_{cht}
pe_{std}	ϕ_{cc}
nPMTs	$ht_{35\%-30\%}$
$ht_{kurtosis}$	$ht_{20\%-15\%}$
$ht_{25\%-20\%}$	$pe_{35\%}$
R_{cc}	$ht_{30\%-25\%}$

TABLE 3.2 – Features used by the BDTE algorithm. *pe* and *ht* reference the charge and hit-time distribution respectively and the percentages are the quantiles of those distributions. *cht* and *cc* reference the barycenters of hit time and charge respectively

The CNN are using 2D projection of the detector representing it as an image with two channel, one for the charge Q and one for the time t. The position of the PMTs is structurally encoded in the pixel containing the information of this PMT. In [57], the pixel is chosen based on a transformation of θ and ϕ coordinates to the 2D plane and rounded to the nearest pixel. A sufficiently large image has been chosen to prevent two PMT to be located in the same pixel. An example of this projection can be found in Figure 3.21. The performances of the CNN can be found in Figure 3.22.



(a) Map of PMTs. **(b)** Charge channel. **(c)** First hit time channel.

FIGURE 3.21 – Projection of the LPMTs in JUNO on a 2D plane. (a) Show the distribution of all PMTs and (b) and (c) are example of what the charge and time channel looks like respectively

Using 2D have the upside of encoding a large part of the informations structurally but loose the rotational invariance of the detector. It also give undefined information to the neural network (what is a pixel without PMT ? What should be its charge and time ?), cause deformation in the representation of the detector (sides of projection) and loose topological informations.

One of the way to present structurally the sphericity of JUNO to a NN is to use a graph: A collection of objects V called nodes and relations E called edges, each relation associated to a couple v_1, v_2 forming the graph $G(E, V)$. Nodes and edges can hold informations or features. In [57] the nodes, are geometrical region of the detector as defined by the HealPix [59]. The features of the nodes are aggregated informations from the PMTs it contains. The edges contains geographic informations of the nodes relative positions.

This data representation has the advantages to keep the topology of the detector intact. It also permit the use of rotational invariant algorithms for the NN, thus taking advantage of the symmetries of the detector.

The neural network then process the graph using Chebyshev Convolutions [47]. The performances

of the GNN are presented in Figure 3.22.

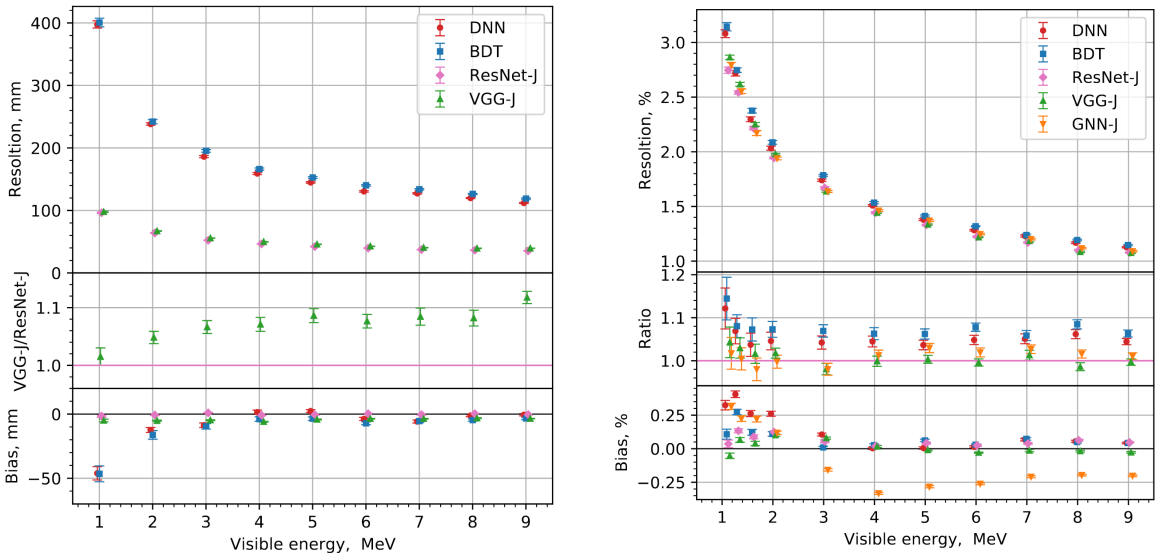


FIGURE 3.22 – Radial (left) and energy (right) resolutions of different ML algorithms. The results presented here are from [57]. DNN is a deep neural network, BDT is a BDT, ResNet-J and VGG-J are CNN and GNN-J is a GNN.

Overall ML algorithms show similar performances as classical algorithms in term of energy reconstructions with the more complex structure CNN and GNN showing better performances than BDT and DNN. For vertex reconstruction, the BDT and DNN show poor performance while CNN are on the level of the classical algorithms.

3.4 Conclusion

That these first DL algorithms tried at JUNO to reconstruct IBDs do not outperform the classical method can be explained. They constitute a first exploration of these methods potential, as do the orginal GNN we describe in Chapter 5. Indeed, the likelihood method is also based on the full list of the charges (Q) and times (t) all PMTs, and the PDF's design accounts for an advanced knowledge of the detector (with a lot of human expertise). The fact that the methods presented in this chapter can learn enough from just the Q, t list, to reach similar performance, is already an interesting result. But this is not decisive yet, in my opinion.

Actually, is there hope that one day DL methods reach better results at JUNO than classical's ? This is not a trivial question. A possibilty would be to let them start from an even rawer level (involving a number of variables which would make a likelihood intractable). This would mean, instead of Q and t, the full waveform in each PMT. With such a quantity of input information to analyse to identify patterns, even DL methods can be limited. The choice of architecture is then important, to guide the algorithm towards pertinent features. We doubt whether CNN's would be the best choice here. We bet that GNN's could be better tools, with more flexibility to hierarchise information (the choice of which PMTs to link already helps here, as well as the possible usage of higher order quantities). The first GNN developped in JUNO (decribed above, [57]) does not do that. It's still only based on (Q, t) couples and link only neighbour PMTs in its first layer. It serves essentially as a way to avoid the problems encountered by CNNs due to the planar projection of a spherical image.

In chapter 5, we tried an original GNN architecture. The goal was not yet to include a rawer

information, but to see if this architecture would perform as well as the one described above when using Q's and t's as the rawest information. If so, then there is hope that when rawer information will be included, this orginal architecture will be the one able to best use it.

Chapter 4

Image recognition for IBD reconstruction with the SPMT system

Dave - Give me the position and momentum, HAL.
HAL - I'm afraid I can't do that Dave.
Dave - What's the problem ?
HAL - I think you know what the problem is just as well as I do.
Dave - What are you talking about, HAL?
HAL - $\sigma_x \sigma_p \geq \frac{\hbar}{2}$

Contents

As explained in Chapter 2, JUNO is an experiment composed of two systems, the Large Photomultiplier (LPMT) system and the Small Photomultiplier (SPMT) system. Both of them observe the same physics events inside of the same medium but they differ in their photo-coverage, respectively 75.2% and 2.7%, their dynamic range (see Section 2.3.2), a thousands versus a few dozen, and their front-end electronics (see section 2.3.2).

The SPMT system is essential to the deployment of the Dual Calorimetry techniques, already mentioned in Section 3.3 and described in [24, 26, 60]. It is indeed less subject than the LPMTs to charge non linearity effects (QNL). This topic will be studied in more detail in Chapter 7, where the potential of one of the Dual Calorimetry techniques is explored. It consists on combined oscillation analyses based on two antineutrino energy spectra : one reconstructed with the LPMT system, the other one with the SPMT system. For that purpose, it is therefore necessary to have reconstruction tools available. Well maintained tools using the LPMT are available in the collaboration's official software. This is not the case concerning the SPMT system, where algorithms were developed more sporadically. This is one of the reasons why we developped the CNN described in this chapter.

Our efforts on it were limited to the early months of this thesis: it was above all a way to learn about ML and about JUNO's detector and software. We benchmarked its performance against a classical algorithm developped in Chapter 4 of [61] but not yet implemented in JUNO's software.

As discussed in Chapter 3, Machine Learning (ML) algorithms shine when modeling highly dimensional data from a given dataset. In our case, we have access to complete monte-carlo simulation of our detector to produce large datasets that could represent multiple years of data taking. Ideally ML algorithms would be able to consider the entirety of the information in the detector and converge on the best parameters to yield optimal results.

The difference between this ideal and what can be achieved in reality is an important subject. In particular, we wonder if an exhaustive usage of the information present in the detector could lead to use informations that are mismodelled in our simulated training samples (or present only in these samples) and therefore lead to biases when the algorithm is applied to real data. A simple way to start addressing this reliability issue is to try to evaluate to which extent various reconstruction methods use the same information. An attempt at this is presented at the end of this chapter. This is also the subject of Chapter 6.

4.1 Method and model

One of simplest way to look at JUNO data is to consider the detector as an array of geometrically distributed sensors on a sphere. Their repartition is almost homogeneous, on this sphere surface providing an almost equal amount of information per unit surface. It is then tempting to represent the detector as a spherical image with the PMTs in place of pixels. Two events with two different energy or position would produce two different images.

The most common approach in machine learning for image processing and image recognition is the Convolutional Neural Network (CNN). It is widely used in research and industry [41, 62–64] due to its strengths (see Section 3.2.2) and has proven its relevance in image processing.

Some CNN are developed to process spherical images [65] but for the sake of simplicity and as a first approach we decided to go with a planar projection of the detector, approach that has proven its efficiency using the LPMT system (see Section 3.3.3). The details about this planar projection will be discussed in section 4.1.2.

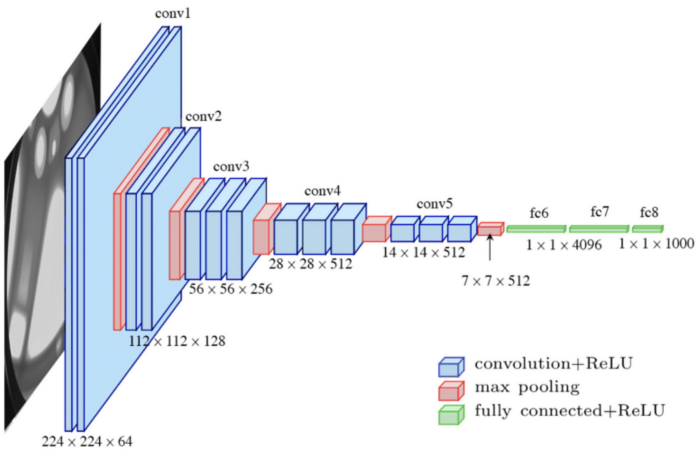


FIGURE 4.1 – Graphic representation of the VGG-16 architecture, presenting the different kind of layer composing the architecture.

4.1.1 Model

The architecture we use is derived from the VGG-16 architecture [41] illustrated in Figure 4.1. We define a set of hyperparameters that will define the size, complexity and computational power of the NN. The chose hyperparameters are detailed below and their values are presented in table 4.1.

- $\mathbf{N_{blocks}}$: the number of convolution blocks, a block being composed of two convolutional layers with 3×3 filters using ReLU activation function, a 3×3 kernel max-pooling layer (except for the last block).
- $\mathbf{N_{channels}}$: The number of channels in the first block. The number of channels in the subsequent blocks is computed using $N^i_{channels} = i * N_{channels}, i \in [1..N_{blocks}]$.
- **FCDNN configuration**: The result of the last convolution layer is flattened then fed to a FCDNN. Its configuration is expressed as the ouputs of sequenced fully connected linear layer using the PReLU activation function. For example $2 * 1024 + 2 * 512$ is the sequence of 2 layers which output is 1024 followed by 2 other layers with an output of 512. Finally the last layer is alinear layer outputing 4 features wihtout activation function. Each feature of the last layer represent a component of the interaction vertex: Energy, X, Y, Z.
- **Loss**: The loss function. In this work we study two different loss function $(E + V)$ and $(E_r + V_r)$ detailed below.

$$(E + V)(E, x, y, z) = (E - E_{dep})^2 + 0.85 \sum_{\lambda \in [x,y,z]} (\lambda - \lambda_{true})^2 \tag{4.1}$$

$$(E_r + V_r)(E, x, y, z) = \frac{(E - E_{dep})^2}{E_{dep}} + \frac{10}{R} \sum_{\lambda \in [x,y,z]} (\lambda - \lambda_{true})^2 \tag{4.2}$$

where E_{dep} is the deposited energy and R is the radius of JUNO's CD. With the energy in MeV and the distance in meters, we use the factor 0.85 and 10 to balance the two term of the loss function so they have the same magnitude.

The loss function $(E + V)$ is close to a simple Mean Squared Error (MSE). MSE is one of the most basic loss function, the derivative is simple and continuous in every point. It is a strong starting point to explore the possibility of CNNs. The loss $(E_r + V_r)$ can be seen as a relative MSE.

The idea is that: due to the inherent statistic uncertainty over the number of collected Number of Photo Electrons (NPE), the absolute resolution $\sigma(E - E_{true})$ will be larger at higher energy than at low energy. But we expect the *relative* energy resolution $\frac{\sigma(E - E_{true})}{E_t rue}$ to be smaller at high energy than lower energy as illustrated in Figure 3.20. Because of this, by using simple MSE the most important part in the loss come from the high energy part of the dataset whereas with a relative MSE, the most important part become the low energy events in the dataset. We hope that by using a relative MSE, the neural network will focus on low energy events where the reconstruction is considered the hardest.

The above losses and their parameters values results from fine-tuning after multiples runs and adjustments of the full random search.

Each combinations of those hyperparameters (for example $(N_{blocks} = 2, N_{channels} = 32, \text{FCDNN} = (2 * 1024), \text{Loss} = (E + V))$) produce models, hereinafter referred as configurations, are then tested and compared to each other over an analysis sample.

On top those generated models, we define 4 hand tailored models:

- Gen_0: $N_{blocks} = 4$, $N_{channels} = 64$, FCDNN configuration: $1024 * 2 + 512 * 2$, Loss $\equiv E + V$
- Gen_1: $N_{blocks} = 4$, $N_{channels} = 64$, FCDNN configuration: $1024 * 2 + 512 * 2$, Loss $\equiv E_r + V_r$
- Gen_2: $N_{blocks} = 5$, $N_{channels} = 64$, FCDNN configuration: $4096 * 2 + 1024 * 2$, Loss $\equiv E + V$
- Gen_3: $N_{blocks} = 5$, $N_{channels} = 64$, FCDNN configuration: $4096 * 2 + 1024 * 2$, Loss $\equiv E_r + V_r$

The resulting models possess between 2'041'034, for Gen_{52} and Gen_{53}, and 5'759'839'242 parameters, for Gen_{26} and Gen_{27}. The models of interest in this thesis, from which the results are discussed in Section 4.3, possess 86'197'196 parameters for Gen_{30} and 332'187'530 parameters for Gen_{42}. For comparison the model of CNN developed in JUNO before posses 38'352'403 parameters [57].

N_{blocks}	{2, 3, 4}
$N_{channels}$	{32, 64, 128}
FCDNN configurations	2 * 1024 2 * 2048 + 2 * 1024 3 * 2048 + 3 * 512 2 * 4096
Loss	$\{E + V, E_r + V_r\}$

TABLE 4.1 – Sets of hyperparameters values considered in this study

To rank the various configuration we cannot used directly the mean loss over the validation dataset as $(E + V)$ and $(E_r + V_r)$ are not numerically comparable. We thus use the following quantities, directly related to the reconstruction performances:

- The mean absolute energy error $\langle E \rangle = \langle |E - E_{true}| \rangle$. It is an indicator of the energy bias of our reconstruction.
- The standard deviation of the energy error $\sigma E = \sigma(E - E_{true})$. This the indicator on our precision in energy reconstruction.
- The mean distance between the reconstructed vertex and the true vertex $\langle V \rangle = \langle |\vec{V} - \vec{V}_{true}| \rangle$. This an indicator of the bias and precision of our vertex reconstruction.
- The standard deviation of the distance between the true and reconstructed vertex $\sigma V = \sigma |\vec{V} - \vec{V}_{true}|$. This is an indicator if the precision in our vertex reconstruction.

The models were developped in Python using the Pytorch framework [43] using NVIDIA A100 [66] and NVIDIA V100 [67] gpus. The A100 was split in two, thus the accessible gpu memory was the same as V100, 20 Gb, making it impossible to train some of the architectures due to memory consumption.

The training was monitored in realtime by a custom tooling that was developed during this thesis, DataMo [68].

The training of one model takes between 4h and 15h depending of its size, overall training the full 72 models takes around 500 GPU hours. Even with parallel training, this random search hyper-optimisation was time consuming.

4.1.2 Data representation

This data is represented as 240×240 images with a charge Q channel and a time t channel. The SPMTs are then projected on the plane as illustrated in Figure 4.2b using the coordinate system presented in 4.2a. The P_y coordinate, the row corresponding to the SPMT in the projection, is proportional to θ. The P_x coordinate, the column corresponding to the SPMT in the projection, is defined by $\phi \sin\theta$ in spherical coordinates. $\theta = 0$ is defined as being the top of the detector and $\phi = 0$ is defined as an arbitrary direction in the detector. In practice, $\phi = 0$ is given by the MC simulation.

$$P_y = \left\lfloor \frac{\theta \cdot H}{\pi} \right\rfloor, \ \theta \in [0, \pi] \tag{4.3}$$

$$P_x = \left\lfloor \frac{(\phi + \pi)\sin\theta \cdot W}{2\pi} \right\rfloor, \ \phi \in [-\pi, \pi], \ \theta \in [0, \pi] \tag{4.4}$$

where H is the height of the image, W the width of the image and $(0, 0)$ the top left corner of the image.

This projection keep the SPMT position in the image proportional to their spherical coordinates while keeping the neighbouring information. This proportionality allow us to keep the specificities of the detector structure, the vertical bands visible in 4.2b.

When two SPMTs in the same pixel are hit in the event time window, the charges are summed and the lowest of the hit-time is chosen. The time window depends on the datasets and are detailed in Section 4.1.2. The SPMTs being located close to each other, we expect the time difference between two successive physics signals, two photons being collected, to be small. The first hit time is chosen because it can be considered as the relative propagation time of the photons that went the "straightest", i.e. that went under the less perturbation of the two. The timing is thus more representative of the event location.

The only potential problem in using this first time come from the Dark Noise (DN). Its time distribution is uniform over the signal and could come before a physics signal on the other SPMT in the pixel. In that case, the time information in the pixel become irrelevant and we lose the timing information for this part of the detector. As illustrated in Figure 4.2b the image dimension have been optimized so that at most two SPMTs are in the same pixel while keeping the number of empty pixels relatively low to prevent this kind of issue.

While it could be possible to use larger images (more pixel) to prevent overlapping, keeping image small images gives multiple advantages:

- As presented in Section 4.1.1, the convolution filter we use are 3×3 convolution filter, meaning that if SPMTs would be separated by more that one pixel, the first filter would only see one SPMT per filter. This behavior would be kind of counterproductive as the first convolution block would basically be a transmission layer and would just induce noise in the data.
- It keep the network relatively small, while this do not impact the convolution layers, the flatten operation just before the FCDNN make the number parameters in the first layer of it dependent on the size of the image.
- It reduce the number of empty pixel in the image.

The question of empty pixel is an important question in this data representation. There is two kind of empty pixels in the data.

The first kind is pixel that contain a SPMT but the SPMT did not get hit nor registered any dark noise during the event. In this case, the charge channel is zero, which have a physical meaning but then come the question of the time layer. One could argue that the correct time would be infinity (or the largest number our memory allows us) because the hit “never” happened, so extremely far from the time of the event. This cause numerical problem as large number, in the linear operation that are happening in the convolution layers, are more significant than smaller value. We could try to encode this feature in another way but no number have any significance due to our time being relative to the trigger of the experiment so -1 for example is out of question. Float and Double gives us access to special value such as NaN (Not a Number) [69] but the behavior is to propagate the NaN which leaves us with NaN for energy and position. We choose to keep the value 0 because it’s the absorbing element of multiplication, absorbing the “information” of the parameter it would be multiplied by. It also can be though as no activation in the ReLU activation function. It’s important to keep in mind

the fact that a part of the detector that has not been hit is also an information: There is no signal in this part of the detector. This problematic will be explored in more details in Chapter 5.

The second kind of pixels are the one that do not represent parts of the detector such as the corners of the image. The question is basically the same, what to put in the charge and the time channel. The decision is to set the charge and time to 0 following the above reasoning.

Another problematic that happens with this representation, and this is not dependent of the chosen projection, is the deformation in the edges of the image and the loss of the neighbouring information in the for the SPMTs at the edge of the image $\phi \sim 180°$. This deformation and neighbouring loss could be partially circumvented as explained in Section 4.4

4.1.3 Dataset

In this study we will discuss two datasets of one millions prompt signal of IBD events.

J21

The first one comes from the JUNO official MC simulation J21v1r0-Pre2 (released the 18th August 2021). This historical version is the one on which the classical SPMT reconstruction algorithm was developed. This classical methods is based on the time likelihood presented Section 3.3 for the vertex reconstruction, and compute the energy by correcting the detector effect on the ration N_{pe}/E_{dep}. It is detailed in Chapter 4 of [61]. This dataset is used as a reference for comparison to classical algorithm performances. The data in this dataset is *detsim* level (see Section 2.6) which includes no digitization, no DAQ and therefore no reconstruction of PMT signals. Only the number of PEs that hit a PMT and the hit times are provided. A fast simulation based on gaussian drawings produces charges, with bias and variability, and the equivalent for times. The drawings parameters were adjusted based on [23, 70]. Because there is no charge reconstruction, the timing on the event is based on the Geant4 simulation, and so $t = 0$ is the moment the positron is created in the CD. To prevent correlation between the numerical value of the time of the first hit t_0 and the radius of the event, we offset all time by this first hit time. Without simulation of the charge reconstruction, we cannot simulate the event trigger, we thus add an arbitrary time cut at a $t_0 + 1000$ ns.

J23

The second comes from the JUNO official monte-carlo simulations J23.0.1-rc8.dc1 (released the 7th January 2024). The data is *calib* level (see Section 2.6). Here the charge comes from the waveform integration, the time window resolution and trigger decision are all simulated inside the software.

To put in perspective this amount of data, the expected IBD rate in JUNO is 47 / days. Taking into account the calibration time, and the source reactor shutdown, it amount to $\sim$ 94′000 IBD events in 6 years. With this million of event, we are training the equivalent of $\sim$ 10 years of data. With this amount we reach a density of $4783 \frac{\text{event}}{\text{m}^3 \cdot \text{MeV}}$, meaning our dataset is representative of the multiple event scenarios that could be happening in the detector.

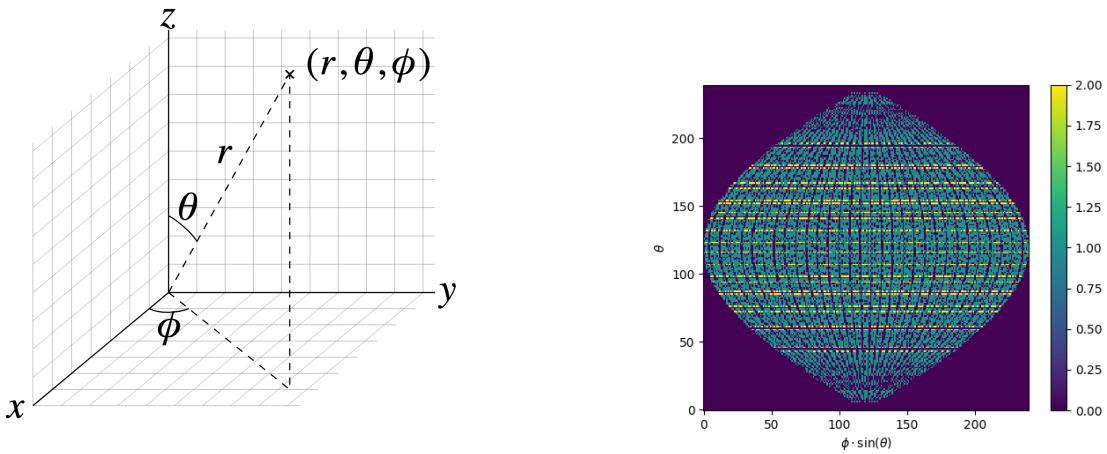
While we expect and hope the MC simulation to give use a realistic representation of the detector, there could be effect, even after the fine-tuning on calibration data, that the simulation cannot handle. Thus, once the calibration will be available, we will need to evaluate, and if needed retrain, the network on calibration data to establish definitive performances.

The simulated data is composed of positron events, uniformly distributed in the CD volume and in kinetic energy over $E_k \in [0; 9]$ MeV producing a deposited energy $E_{dep} \in [1.022; 10.022]$ MeV. This is done to mimic the signal produced by the IBD prompt signal. Uniform distributions are used so that

the CNN does not learn a potential energy distribution, favoring some part of the energy spectrum instead of other.

4.1.4 Data characteristics

To delve a bit into the kind of data we will use, you can find in Figure 4.2b the repartition of the SPMTs in the image. The color represent the number of SPMTs per pixel.



(A) Spherical coordinate system used in JUNO for reconstruction

(B) Repartition of SPMTs in the image projection. The color scale is the number of SPMTs per pixel

FIGURE 4.2

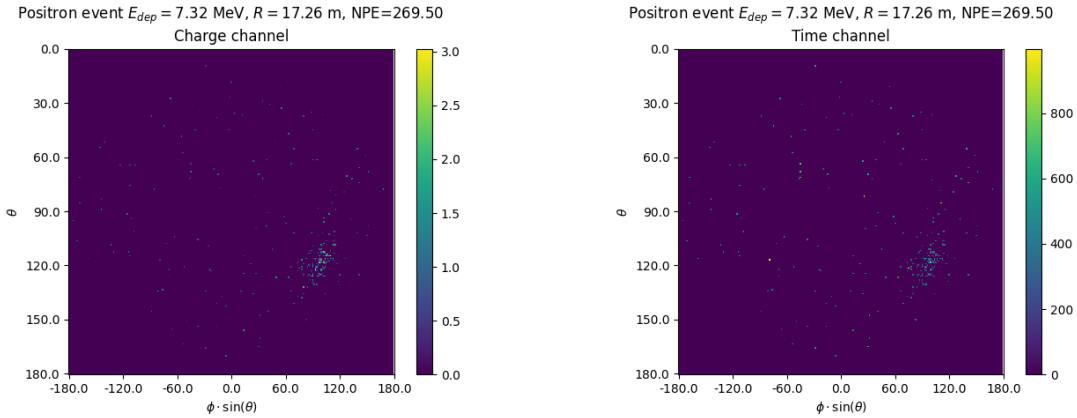


FIGURE 4.3 – Example of a high energy, radial event. We see a concentration of the charge on the bottom right of the image, clear indication of a high radius event. **On the left**: the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right**: The time channel in nanoseconds.

See also Figures 4.3 to 4.6 - and the explanation in their captions - which present events from J23 for different positions and energies. We see some characteristics and we can instinctively understand how the CNN could discriminate different situations.

To give an idea of the strength of the signal in comparison to the dark noise background, Figure 4.7a present the distribution of the ratio of NPE per deposited energy. Assuming a linear response of the

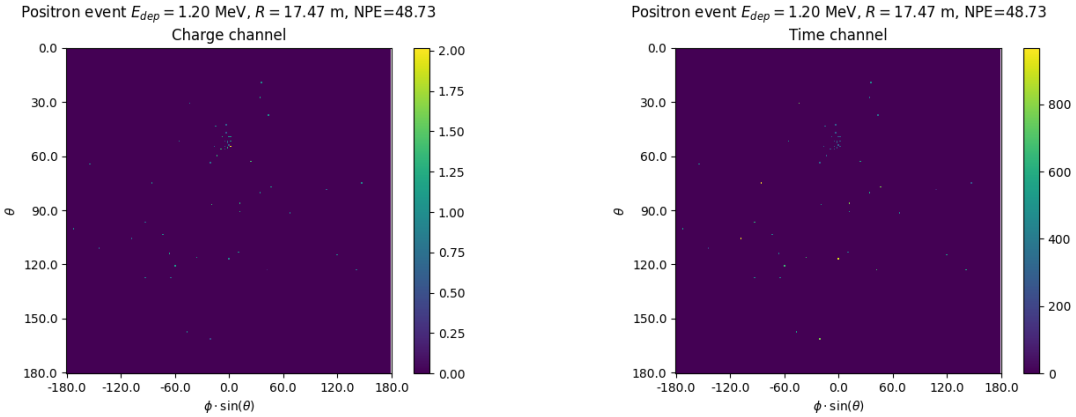


FIGURE 4.4 – Example of a low energy, radial event. The signal here is way less explicit, we can kind of guess that the event is located in the top middle of the image. **On the left**: the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right**: The time channel in nanoseconds.

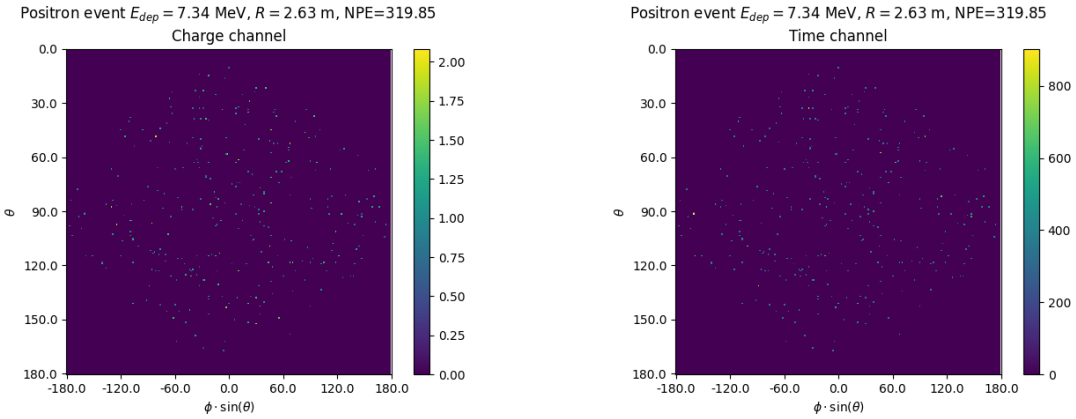


FIGURE 4.5 – Example of a high energy, central event. In this image we can see a lot of signal but uniformly spread, this is indicative of a central event. **On the left**: the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right**: The time channel in nanoseconds.

LS we can model:

$$NPE_{tot} = E_{dep} \cdot P_{mev} + D_N \tag{4.5}$$

$$\frac{NPE_{tot}}{E_{dep}} = P_{mev} + \frac{D_N}{E_{dep}} \tag{4.6}$$

where NPE_{tot} is the total number of PE detected by the event, P_{mev} is the mean number of PE detected per MeV and D_N is the dark noise contribution that is considered energy independent. In the case where the readout time window is dependent of the energy the dark noise contribution become energy dependant, also the LS response is realistically energy dependant but Figure 4.7a shows that we hare heavily dominated by the stochastic behavior of light emission and detection.

The fit shows a light yield of 40.78 PE/MeV and a dark noise contribution of 4.29 NPE. As shown in Figure 4.7b, the physics makes for 90% of the signal at low energy.

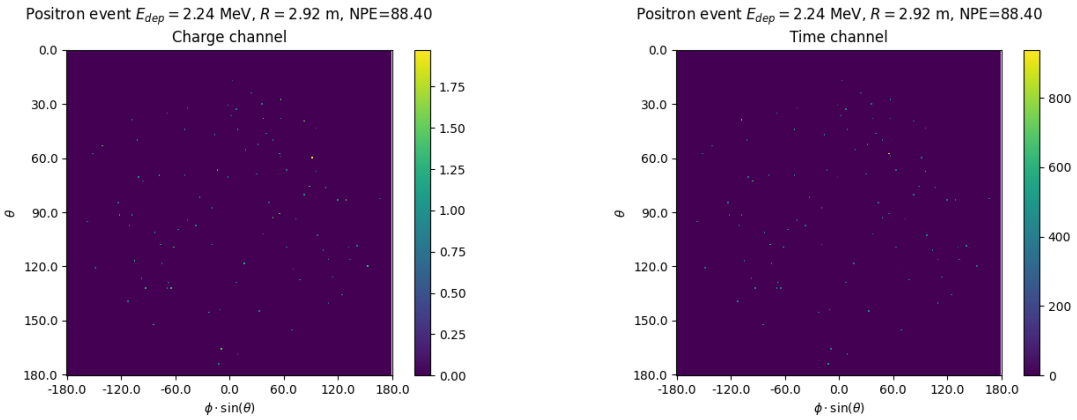


FIGURE 4.6 – Example of a low energy, central event. Here there is no clear signal, the uniformity of the distribution should make it central. **On the left**: the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right**: The time channel in nanoseconds.

4.2 Training

The optimizer used for the training is the Adam [38] optimizer, with a learning rate λ of 1e-3. The other hyperparameters were left to their default value ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e^{-8}$). The learning rate was reduce exponentially during the training at a rate of $\gamma = 0.95$, thus $\lambda_{i+1} = 0.95\lambda_i$ where i is the epoch.

Following the lifecycle presented in Section 3.1.3, the training used a batch size of 64 events meaning that, each step, the loss is computed on 64 events before updating the NN parameters. An epoch is composed of 10k steps, thus each epoch, the NN see 640k events. The training last for 30 epochs, so overall the NN go through 19.2 millions events or 19.2 times the dataset.

The number of epoch, batch size, learning rate and its decay where fine-tuned during the development of the CNN.

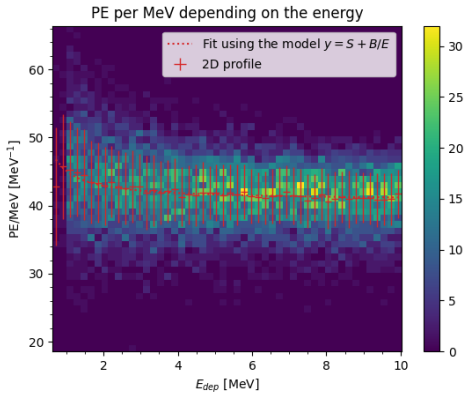
4.3 Results

Before presenting the results, lets discuss the different observables.

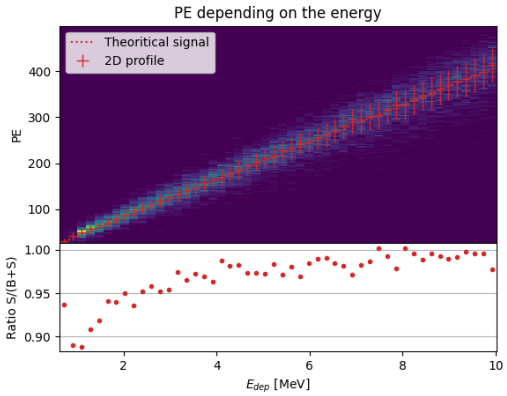
The event are considered point like in this study. The target truth position, or vertex, is the mean position of the energy deposits of the positron and the two annihilation gammas. This approximation for point like interaction is also used for the likelihood study presented in Section 3.3 and in previous ML studies presented in section 3.3.3 [57].

Due to the symmetries of the detector, we mainly consider and discuss the bias and precision evolution depending of the radius R but we will still monitor the performances depending of the spheric angle θ and ϕ. From the detector construction and effect we expect dependency in radius due to the TR area effect presented in Section 3.3 and the possibility for the positron or the gammas to escape from the CD for positrons interacting near the edge. We also expect dependency on θ, the top of the experiment being non-instrumented due to the filling chimney. It is also to be noted that the events in the dataset are uniformly distributed in the CD, and so are uniformly distributed in R^3 and ϕ. The θ distribution is not uniform and we will have more event for $\theta \sim 90°$ that $\theta \sim 0°$ or $\theta \sim 180°$.

We define multiple energy in JUNO:



(A) Distribution of PE/MeV in the J23 Dataset. This distribution is profiled and fitted using equation 4.6



(B) **On top**: Distribution of PE vs Energy. **On bottom**: Using the values extracted in 4.7a, we calculate the ration signal over background + signal

FIGURE 4.7

- E_ν: The energy of the neutrino.
- E_k: The kinetic energy of the resulting positron from the IBD.
- E_{dep}: The deposited energy of the positron and the two annihilation gammas.
- E_{vis}: The equivalent visible energy, so E_{dep} after the detector effect such as the LS response non-linearity.
- E_{rec}: The reconstructed energy by the reconstruction algorithm. The expected value depend on the algorithm we discuss about. For example the algorithm presented in Section 3.3 reconstruct E_{vis} while the ones presented in section 3.3.3 reconstruct E_{dep}.

In this study, we will set E_{dep} as our target for energy reconstruction. This choice is motivated by the ease with which we can retrieve this information in the monte-carlo data while E_{vis} is less trivial to retrieve.

4.3.1 J21 results

The best results comes from the Gen_{30} model, meaning then 30th model generated using the table 4.1: Gen_{30}: $N_{blocks} = 3$, $N_{channels} = 32$, FCDNN configuration: 2048 * 2 + 1024 * 2, Loss $\equiv E + V$.

The performances of its reconstruction are presented in blue in Figure 4.8. Superimposed in black is the performances of the classical algorithm from [61].

Energy reconstruction

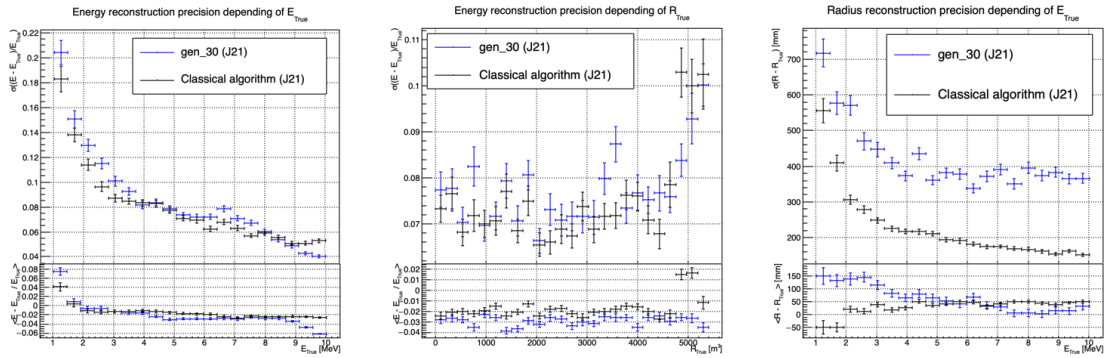
By looking at the Figure 4.8a and 4.8b, the CNN has similar performances in its energy resolution. Important biases, however, appear at low and high energy.

This is explained by looking at the true and reconstructed energy distributions in Figure 4.10a. We see that the distributions are similar for energies before 8 MeV but there is an excess of event reconstructed with energies around 9 MeV while a lack of them for 10 MeV. The neural network seems to learn the energy distribution and learn that it exist almost no event with an energy inferior to 1.022 MeV and not event with an energy superior to 10 MeV.

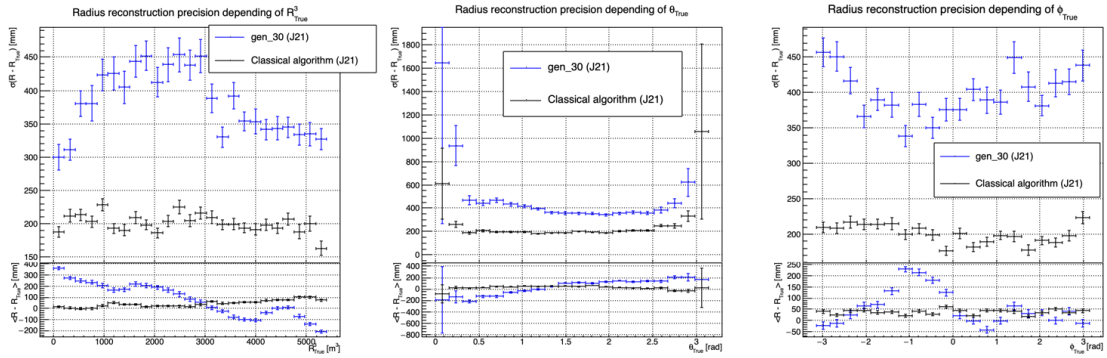
The first observation is a physics phenomena: for a positron, its minimum deposited energy is the mass energy coming from its annihilation with an electron 1.022 MeV. There is a few event with

energies inferior to 1.022 MeV, in those case the annihilation gammas or even the positron escape the detector. The deposited energy in the LS is thus only a fraction of the energy of the event.

The second observation is indeed true in this dataset but has no physical meaning, it is an arbitrary limit because the physics region of interest is mainly between 1 and 9 MeV of deposited energy (Figure 2.2). By learning the energy distribution, the CNN pull event from the border of it to more central value. That's why the energy resolution is better: the events are pulled in a small energy region , thus a small variance but the bias become very high (Figure 4.8a).



(A) Resolution and bias of energy reconstruction vs energy (B) Resolution and bias of energy reconstruction vs radius (C) Resolution and bias of radius reconstruction vs energy



(D) Resolution and bias of radius reconstruction vs radius (E) Resolution and bias of radius reconstruction vs θ (F) Resolution and bias of radius reconstruction vs ϕ

FIGURE 4.8 – Reconstruction performance of the Gen_{30} model on J21 data and it's comparison to the performances of the classic algorithm "Classical algorithm" from [61]. The top part of each plot is the resolution and the bottom part is the bias.

This behavior also explain the heavy bias at low energy in Figure 4.8a. The energy bias of the CNN if fairly constant over the energy range, it is interesting to note that the energy bias depending on the radius is a bit worse than the classical method.

Vertex reconstruction

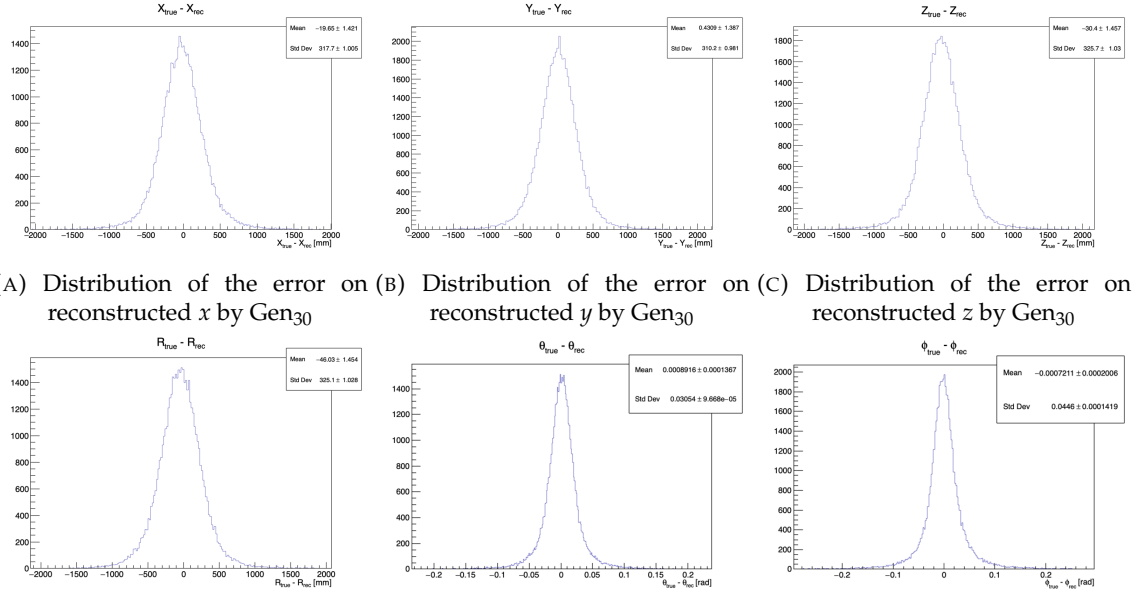
For the vertex reconstruction we do not study x, y and z independently but we use R as a proxy observable. Figure 4.9 shows the residual distribution of the different vertex coordinates. We see that R errors and biases are slightly superior to the cartesian coordinates, thus R is a conservative proxy observable to discuss the subject of vertex reconstruction.

The comparison of radius reconstruction between the classical algorithm and Gen_{30} are presented in the Figures 4.8c, 4.8d, 4.8e and 4.8f. The resolution obtained by the CNN is twice worse in average,

and worse in all studied regions. In energy, Figure 4.8c, where we see a degradation of almost 20cm over the energy range. When looking over the true event radius, Figure 4.8d, we lose between 30 and 45cm of resolution. The performances are the best for central and radial event.

The precision also worsen when looking at the edge of the image $\theta \approx 0$, $\theta \approx 2\pi$ respectively the top and bottom of the image, and when $\phi \approx -\pi$ and $\phi \approx \pi$ respectively the left and right side of the image.

The bias in radius reconstruction is about the same order of magnitude depending of the energy but is of opposite sign. As for the energy, this behavior is studied in more details in Section 4.3.2. Over radius, θ and ϕ the bias is inconsistent, sometimes event better than the classical reconstruction but can also be much worse than the classical method. This could come from the specialisation of some filters in the convolutional layers for specific part of the detector that would still work "correctly" for other parts but with much less precision.



(A) Distribution of the error on reconstructed x by Gen_{30} (B) Distribution of the error on reconstructed y by Gen_{30} (C) Distribution of the error on reconstructed z by Gen_{30}

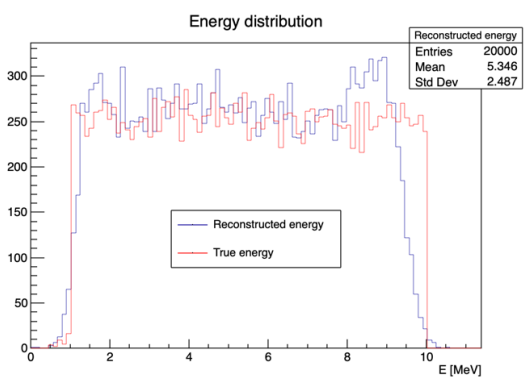
(D) Distribution of the error on reconstructed R by Gen_{30} (E) Distribution of the error on reconstructed θ by Gen_{30} (F) Distribution of the error on reconstructed ϕ by Gen_{30}

FIGURE 4.9 – Residual distribution of the different component of the vertex by Gen_{30}. The reconstructed component are x, y and z but we see similar behavior in the error of R, θ and ϕ.

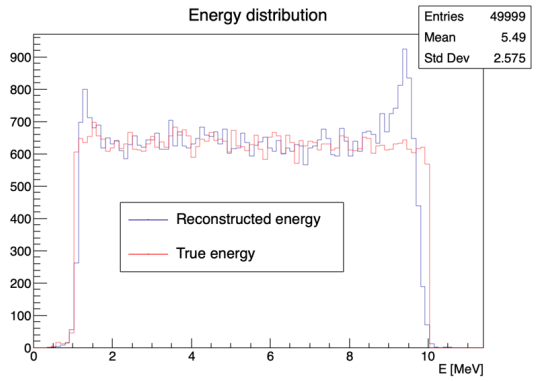
As mentioned in the introduction of this chapter, this CNN initially served as a tool for learning about machine learning and JUNO's detector and software. It eventually became necessary for use as an SPMT reconstruction tool in Chapter 7, so we made some optimizations. However, we did not invest much time in fully addressing its issues.

4.3.2 J21 Combination of classic and ML estimator

As it has been presented in previous section, there is instances where the reconstructed energy and vertex behaves differently between the neural network and the classic algorithm. For instance, if we look at Figure 4.8c, we see that while the CNN tend to overestimate the radius at low energy while the classical algorithm seems to underestimate it. Let's designate the two reconstruction algorithms as estimator of X, the truth about the event in the phase space (E, x, y, z). The CNN and the classical



(A) Distribution of Gen_{30} reconstructed energy and true energy of the analysis dataset (J21)



(B) Distribution of Gen_{42} reconstructed energy and true energy of the analysis dataset (J23)

FIGURE 4.10

algorithm are respectively designated as $\theta_N(X)$ and $\theta_C(X)$.

$$E[\theta_N] = \mu_N + X;\ \mathrm{Var}[\theta_N] = \sigma_N^2 \tag{4.7}$$
$$E[\theta_C] = \mu_C + X;\ \mathrm{Var}[\theta_C] = \sigma_C^2 \tag{4.8}$$

where μ is the bias of the estimator and σ^2 its variance.

Now if we were to combine the two esitmators using a simple mean

$$\hat{\theta}(X) = \frac{1}{2}(\theta_N(X) + \theta_C(X)) \tag{4.9}$$

then the variance and mean would follow

$$E[\hat{\theta}] = \frac{1}{2}E[\theta_N] + \frac{1}{2}E[\theta_X] \tag{4.10}$$
$$= \frac{1}{2}(\mu_N + X + \mu_C + X) \tag{4.11}$$
$$= \frac{1}{2}(\mu_N + \mu_C) + X \tag{4.12}$$

$$\mathrm{Var}[\hat{\theta}] = \frac{1}{4}\sigma_N^2 + \frac{1}{4}\sigma_C^2 + 2 \cdot \frac{1}{4} \cdot \sigma_{NC} \tag{4.13}$$
$$= \frac{1}{4}\sigma_N^2 + \frac{1}{4}\sigma_C^2 + \frac{1}{2} \cdot \sigma_{NC} \tag{4.14}$$
$$= \frac{1}{4}\sigma_N^2 + \frac{1}{4}\sigma_C^2 + \frac{1}{2} \cdot \sigma_N\sigma_C\rho_{NC} \tag{4.15}$$

Where σ_{NC} is the covariance between θ_N and θ_C and ρ_{NC} their correlation.

We see immediately that if the two estimators are of opposite bias, the bias of the resulting estimator is reduced. For the variance, it depends of ρ_{NC} but in this case if σ_C^2 is close to σ_N^2 then even for $\rho_{NC} \lneq 1$ then we can gain in resolution.

By generalising the equation 4.9 to

$$\hat{\theta}(X) = \alpha\theta_N + (1-\alpha)\theta_C;\ \alpha \in [0,1] \tag{4.16}$$

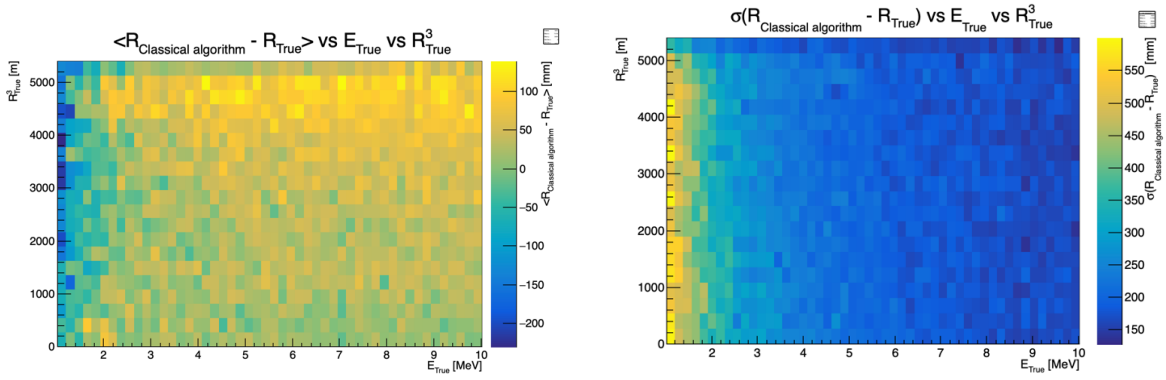


FIGURE 4.11 – Radius bias (**on the left**) and resolution(**on the right**) of the classical algorithm in a E, R^3 grid

we can determine an optimal α for two combined estimators. The estimators with the smallest variance

$$\alpha = \frac{\sigma_C^2 - \sigma_N \sigma_C \rho_{NC}}{\sigma_N^2 + \sigma_C^2 - 2\sigma_N \sigma_C \rho_{NC}} \tag{4.17}$$

and the estimator without bias

$$\alpha = \frac{\mu_C}{\mu_C - \mu_N} \tag{4.18}$$

See annex A for demonstration.

We present in this section the result of the estimator with the smallest variance.

Its pretty clear from the results shown in Figure 4.8 that the bias, variances and correlation are not constant across the (E, R^3) phase space. We thus compute those parameters in a grid in E and R^3 for the following results as illustrated in 4.11.

The map we are using are composed of 20 bins for R^3 going from 0 to 5400 m^3 (17.54 m) and 50 bins in energy ranging from 1.022 to 10.022 MeV. In the case where we are outside the grid, we use the closest cell.

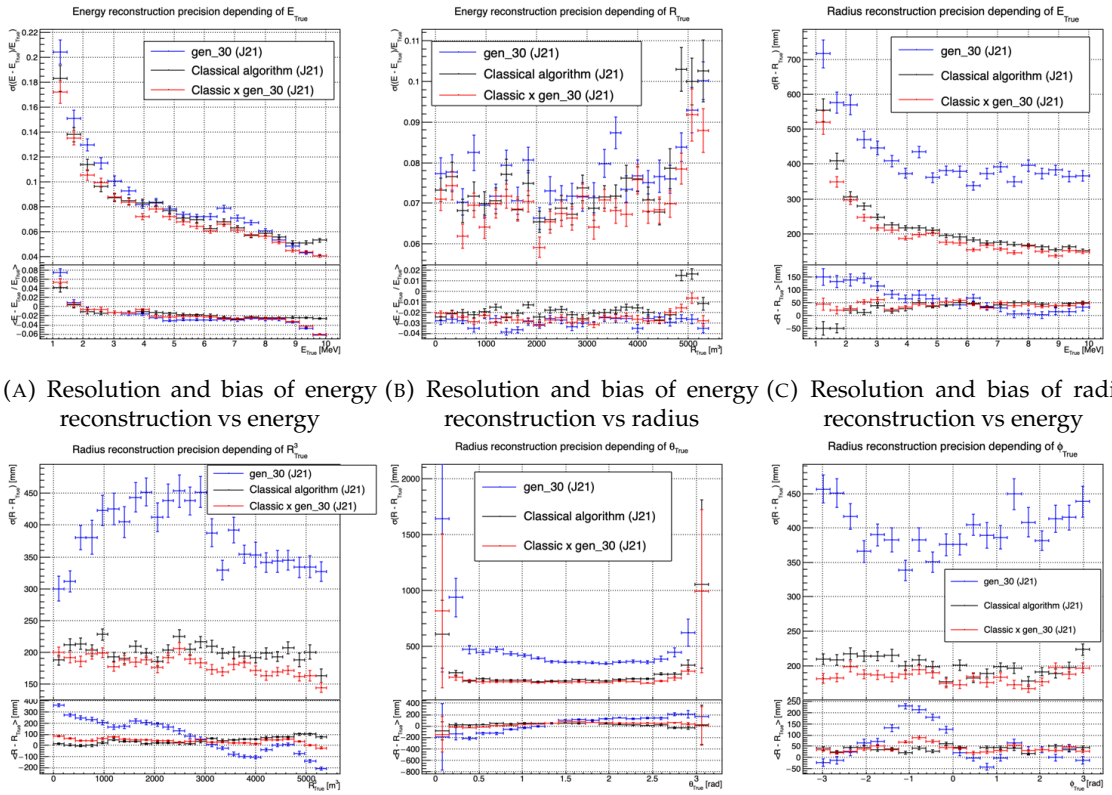
The performance of this weighted mean is presented in Figure 4.12. We can see that even when the CNN resolution is much worse than the classical algorithm, it can still bring some information thus improving the resolution. This comes from the correlation of the reconstruction error to be smaller than 1 as presented in Figure 4.13. We even see some anticorrelation in the radius reconstruction for High radius, high energy, event.

This technique is not suited for realistic reconstruction, we rely too much on the knowledge of the resolution, bias and correlation between the two methods. While this is possible to determine using simulated data or calibration sources, the real data might differ from our model and we would need to really well understand the behavior of the two system. But this is a good tool to detect that algorithms don't all use the same information, and is a first step to identify new information that could be brought to the best algorithms, to improve their performance.

4.3.3 J23 results

We needed for Chapter 7 a SPMT reconstruction tool to run the comparison with LPMT. We thus retrained the SPMT CNN on newer, more realistic data.

The J21 simulation is fairly old and newer version, such as J23, include refined measurements of the



(A) Resolution and bias of energy reconstruction vs energy (B) Resolution and bias of energy reconstruction vs radius (C) Resolution and bias of radius reconstruction vs energy

(D) Resolution and bias of radius reconstruction vs radius (E) Resolution and bias of radius reconstruction vs θ (F) Resolution and bias of radius reconstruction vs ϕ

FIGURE 4.12 – Reconstruction performance of the Gen_{30} model on J21, the classic algorithm "Classical algorithm" from [61] and the combination of both using weighted mean. The top part of each plot is the resolution and the bottom part is the bias.

light yield, reflection indices of materials of the detector, structural elements such as the connecting structure and more realistic dark noise. Additionally, the trigger, waveform integration and time window are defined using the algorithms that will ultimately be used by the collaboration to process real physics events.

We retrained the models defined in 4.1.1 on the J23 data and used the same hyperparameter optimisation procedure. The results from the best architecture, Gen_{42}, are presented in Figure 4.14. Following the table 4.1, Gen_{42}: $N_{blocks} = 3$, $N_{channels} = 64$, FCDNN configuration: 4096 * 2, Loss $\equiv E + V$.

Energy reconstruction

The results of the energy reconstruction are presented in Figures 4.14a and 4.14b. The resolution is close to the one of the classical algorithm with the exception of the start and end of the spectrum. This is the same effect that we saw with Gen_{30}, events are pulled from the edge of the distribution, resulting in smaller resolution but heavy biases.

Vertex reconstruction

The vertex reconstruction, presented in Figures 4.14c, 4.14d, 4.14e and 4.14f is not yet to the level of the classical reconstruction but the degradation is smaller than for Gen_{30} being at most a difference

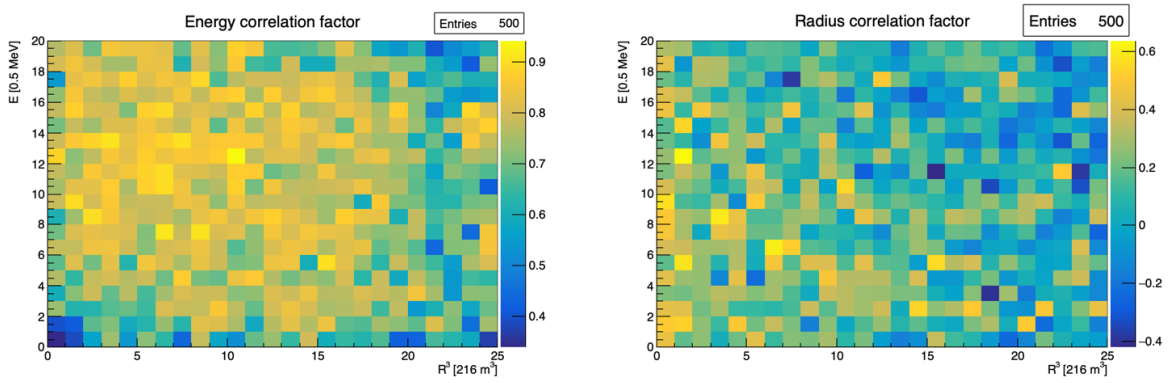


FIGURE 4.13 – Correlation between CNN and classical method reconstruction (**on the left**) for energy and (**on the right**) for radius in a E, R^3 grid

of 15cm of resolution and closing to the performance of the classical algorithm in the most favourable condition. Gen_{42} has also very little bias in comparison with the classical method with the exception of the transition to the TR area and at the very edge of the detector.

With a more realistic description of the propagation and collection of scintillation photons, of the charge and time resolutions, of the DN and of the trigger, it seems new features can be identified by the CNN.

Unfortunately could not rerun the classical algorithm over the J23 data, as the algorithm was optimised for J21 and was not included and maintained over J23. The combination method need for the two estimators to be run on the same set of event, which was impossible without the classical algorithm being maintained for J23.

4.4 Conclusion and prospect

In this chapter we have developed a CNN for the reconstruction of IBD prompt signals. This work was the opportunity to learn about machine learning and neural networks, and familiarise ourselves with JUNO's detector and software.

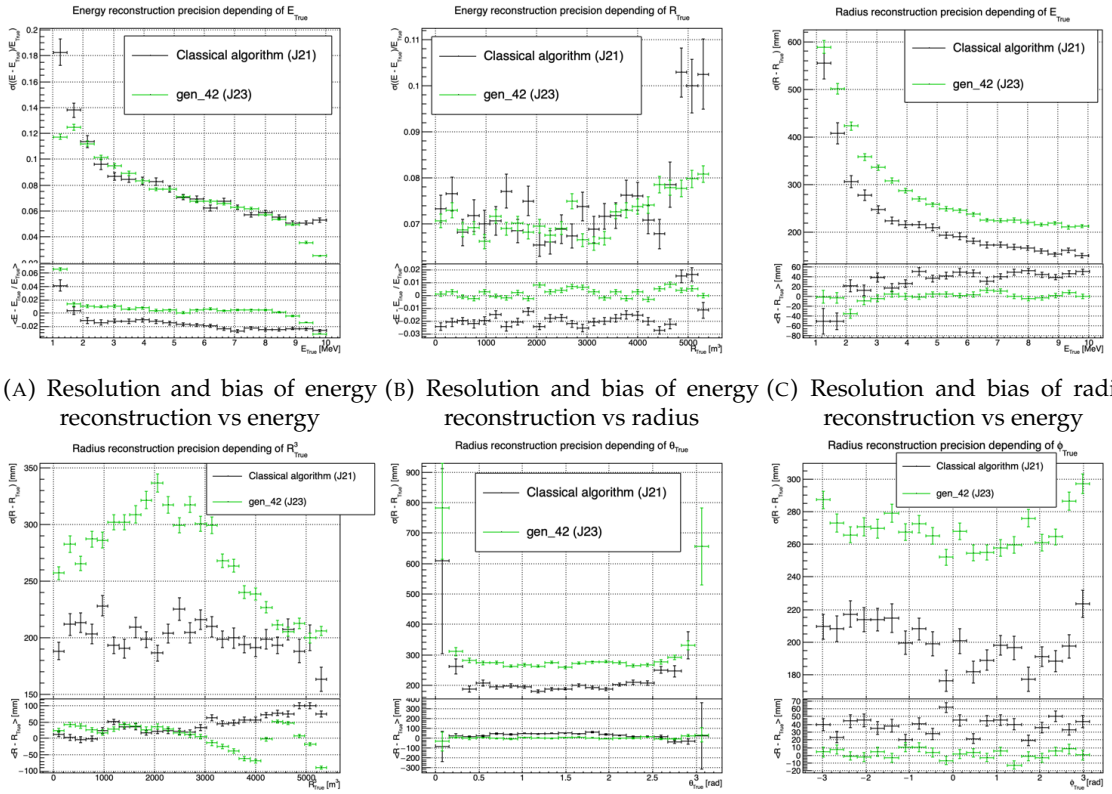
This work was revisited for the needs of Chapter 7, providing a reconstruction tools for the SPMT.

The CNN we developed suffers limitations in its performance. We think one of the reasons for this lies in the data representation. A lot of training time and resources is consumed going and optimizing over pixel with no physical meaning, the NN needs to optimized itself to take into account edges cases such as event at the edge of the image and deformation of the charge distribution.

Those problems could be circumvented, we could imagine a two part CNN where the first part reconstruct the θ and ϕ spherical coordinates and then rotate the image to locate the event in the center of the image. The second part, from this rotated image, would reconstruct the radius and energy of the event.

To overcome the time problematic, i.e. what is the time of a PMT that was never hit, we could transform this channel into a dimension. This would results in an image with multiple charge channels, each one representing the charge sum in a time interval.

Another possibility is to use a kind of algorithm that does not impose a planar projection, like a GNN. It has other advantages, as will be presented in the next chapter, where we propose a GNN to reconstruct IBD's with the LPMT system.



(A) Resolution and bias of energy reconstruction vs energy

(B) Resolution and bias of energy reconstruction vs radius

(C) Resolution and bias of radius reconstruction vs energy

(D) Resolution and bias of radius reconstruction vs radius

(E) Resolution and bias of radius reconstruction vs θ

(F) Resolution and bias of radius reconstruction vs ϕ

FIGURE 4.14 – Reconstruction performance of the Gen_{42} model on J23 data and it's comparison to the performances of the classic algorithm "Classical algorithm" from [61]. The top part of each plot is the resolution and the bottom part is the bias.

Chapter 5

Graph representation of JUNO for IBD reconstruction

"The Answer to the Great Question of Life, the Universe and Everything is Forty-two"

Douglas Adams, The Hitchhiker's Guide to the Galaxy

Contents

In Section 3.3.3, we showed that all ML methods developed before this thesis to reconstruct IBDs have similar results, and that their performance is very similar to that of the classical, likelihood-based algorithm. We think these similarities can reasonably be explained by this: the input data used by all these methods to compute E or $\vec{X}$ is the same full list of PMT integrated signals $\{(Q_i, t_i);\ i \in 1, ..., N_{PMTs}\}$, and by the high level of sophistication of the detector's description in the likelihood. It's probable that the likelihood method looses very little information.

May be some was, but that the ML algorithms were not desgined well enough to recover it. It's also reasonable to think that ML algorithms will make a difference when, instead of the list of (Q_i, t_i), a rawer information will be used in input, like the full waveform. To actually be able to learn from such a complex and high dimensional input, well designed architectures (that would guide the learning toward the solution) are necessary. In any case, it seemed welcome to us to propose an additional algorithm, with an original architecture.

For the fist stage of its development, the purpose of this part of my thesis, we considered it was enough to also take the (Q_i, t_i) list as the input. While achieving equivalent performance with simpler input might suggest that the architecture is not immediately advantageous, it remains crucial to explore the performance with more complex, rawer inputs such as full waveforms. This is where the true potential of the architecture could emerge, as it could better capture the intricacies that simpler inputs fail to represent. If performance does not improve with these richer inputs, it would then be appropriate to question the relevance of this approach.

The algorithm we propose is a GNN. It also has the advantage of addressing sphericity issues described in Chapter 4. From this graph representation, we can construct a neural network that will process the data while keeping some interesting properties. For example the rotational invariance, i.e. the energy and radius of the event do change by rotation our referential. For more details see Section 3.2.3. Graph representation also has the advantage to be able to encode global and higher order informations.

5.1 Data representation

In Section 3.3.3, we mentioned a GNN developed before the beginning of this thesis to reconstruct IBD energies in JUNO [57]. In their approach: nodes of the graph correspond to 3072 pixels representing geometric regions of the detector and the information of the $\sim$ 6 LPMTs found in a pixel are then aggregated on those nodes.This aggregation serves to simplify the data input, though at the potential cost of losing finer-grained details. The network then process the data using the equivalent of convolution but on graph [47]. In the first layer, each node is connected only with its direct neighbours.

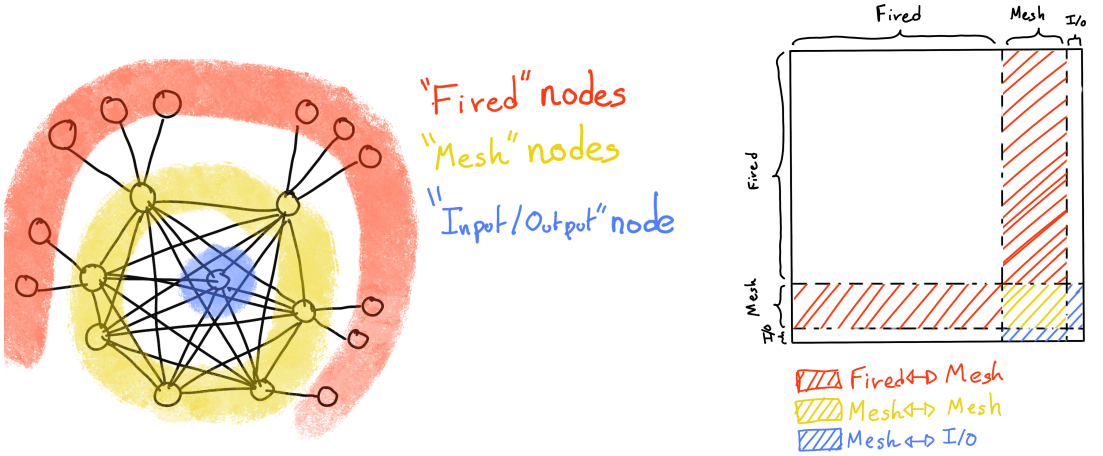
To determine the energy released by an IBD in the LS, it is helpful to determine the position of the main energy deposit. Therefore, relative Q and t's of PMTs all around the sphere is a useful information. If in the first layer only neighbour nodes are linked, several layers are necessary to access this detector-wide information. In an ideal world, we would develop a Graph NN where each PMT is a node (even if it has not been hit in the event under consideration, since this is in itself an information) and where each node is connected to all the other ones. This makes the detector-wide information avalaible as early as the first layer. This architecture might help the network to better learn. Such an architecture can also be motivated this way: one of the strength of GNN's is their capacity to encompass the characteristics of a detector. A node can be the representation of a detector element, and the edge can represent its relationship with other elements. In the case of JUNO, any measurement is collective : an interaction is seen by all the PMTs, with no a priori hierarchy in the role of each. A fully connected GNN is particularly advantageous in JUNO's case, as the lack of a priori hierarchy among the PMTs makes it important to ensure that information is shared globally from the outset. This architecture allows the network to access detector-wide information as early as the first layer, potentially improving learning efficiency. However, this comes at a significant computational cost, which necessitates careful balancing between memory usage and model performance

Another advantage of a GNN is also that it is well adapted to inhomogenous detectors. We therefore tried to build GNNs including both LPMTs ans SPMTs.

With 17612 LPMTs and 25600 SPMTs, the ideal fully connected Graph mentioned above is impossible: even excluding self relation and considering the relation to be undirected (the edge from a node A to a node B being the same from as the one from B to A) the amount of necessary edges would be $n(n-1)/2$ with $n = 43212$ nodes. This amounts to 933'616'866 edges. If we encode an information with double precision (64 bits) in what we call an adjacency matrix, illustrated in Figure 3.12, each information we want to encode in the relation would consume 4 GB of data. When adding the overhead due to gradient computation during training, this would put us over the memory capacity of a single V100 gpu card (20 GB of memory). We could use parallel training to distribute the training over multiple GPU but we considered that the technical challenge to deploy this solution was too high.

We finally decided of a middle ground where we define three *families* of nodes:

— The core of the graph is composed of nodes representing geometric regions of the detector. We call those nodes mesh nodes. Those mesh nodes are all connected to each other. We keep their number low to gain in memory consumption.



(A) Illustration of the different nodes in our graphs and their relations.

(B) Illustration of what a dense adjacency matrix would looks like and the part we are really interested in. Because Fired → Mesh and Mesh → I/O relations are undirected, we only consider in practice the top right part of the matrix for those relations.

FIGURE 5.1

— PMTs in which Photo-Electrons (PE) are found are represented by fired nodes. Fired nodes are connected to the mesh node they geometrically belong to.
— A final node is called the input/output node (I/O). It is connected to every mesh node. Its features are combinations of signals found in the whole detector.

Those nodes and their relations are illustrated in Figure 5.1a. From this representation, we end up with three distinct adjacency adjacency matrix
— A $N_{fired} \times N_{mesh}$ adjacency matrix, representing the relations between fired and mesh. Those relations are undirected.
— A $N_{mesh} \times N_{mesh}$ adjacency matrix, representing the relation between meshes. Those relation are directed.
— A $N_{mesh} \times 1$ adjacency between the mesh and I/O nodes. Those relations are undirected.
The adjacency matrix representing those relation is illustrated in Figure 5.1b.

The mesh segmentation is following the Healpix segmentation [71]. This segmentation offer the advantage that almost each mesh have the same number of direct neighbours and it guarantee that each mesh represent the same extent of the detector surface. The segmentation can be infinitely subdivided to provide smaller and smaller pixels. The number of pixel follow the order n with $N_{pix} = 12 \cdot 4^n$. This segmentation is illustrated in Figure 5.2. To keep the number of mesh small, we use the segmentation of order 2, $N_{pix} = 12 \cdot 4^2 = 192$.

We decided on having the different kind of nodes mesh (M), fired (F) and I/O have different set of features. The features used in the graph are presented in tables 5.1 and 5.2. Most of the features are low level informations such as the charge or time information but we include some high order features such as

1. P_l^h: Is the normalized power of the lth spherical harmonic. For more details about spherical harmonics in JUNO, see annex B.
2. $\mathbb{A}$ and $\mathbb{B}$ are informations that are related the likeliness of the interaction vertex to be on the

FIGURE 5.2 – Illustration of the Healpix segmentation. **On the left:** A segmentation of order 0. **On the right:** A segmentation of order 1

segment between the center of two meshes.

$$\mathbb{A}_{ij} = (\vec{j} - \vec{i}) \cdot \frac{l_1}{D_{ij}} + \vec{i} \tag{5.1}$$

$$\mathbb{B}_{ij} = \frac{Q_i}{Q_j} \left(\frac{l_2}{l_1} \right)^2 \tag{5.2}$$

$$l_1 = \frac{1}{2}(D_{ij} - \Delta t \frac{c}{n}) \tag{5.3}$$

$$l_2 = \frac{1}{2}(D_{ij} + \Delta t \frac{c}{n}) \tag{5.4}$$

where $\vec{i}$ is the position vector of the mesh i, D_{ij} is the distance between the center of the meshes i and j, Q_i the sum of charges on the mesh i, $\Delta t = t_i - t_j$ where t_i the earliest time on the mesh i and n the optical index of the LS. $\mathbb{A}$ is the vertex between center of meshes distance ratio between i and j based on the time information. For $\mathbb{B}$, the charge ratio evolve with the square of the distance, so the mesh couple with the smallest $\mathbb{B}$ should be the one with the interaction vertex between its two center.

Fired	Mesh	I/O
Q	$\langle Q_m \rangle$	$\langle X \rangle$
t	σQ_m	$\langle Y \rangle$
x	$\min(t_m)$	$\langle Z \rangle$
y	$\max(t_m)$	$\sum Q$
LPMT/SPMT: 1/-1	σt_m	$P_l^h;\ l \in [0, 8]$
	X_m	
	Y_m	
	Z_m	

TABLE 5.1 – Features on the nodes of the graph. All charge are in [nPE], time in [ns] and position in [m].
Q and t are the reconstructed charge and time of the hit PMTs. (x, y, z) is the position of the PMTs and the last parameter represent the type of the PMT. It's 1 for LPMT and -1 for SPMT
Q_m and t_m is the set of charges and time of the PMT belonging the mesh m. (X_m, Y_m, Z_m) i the position of the center of the geometric region represented by the mesh m
$(\langle X \rangle, \langle Y \rangle, \langle Z \rangle)$ is the position of the charge barycenter, $\sum Q$ the sum of the collected charge in the detector and P_l^h is the relative power of the lth harmonic. See annex B for details.

Fired → Mesh	Mesh ($m1$) → Mesh ($m2$)	Mesh → I/O
$x - X_m$	$X_{m1} - X_{m2}$	$\langle X \rangle - X_m$
$y - Y_m$	$Y_{m1} - Y_{m2}$	$\langle Y \rangle - Y_m$
$z - Z_m$	$Z_{m1} - Z_{m2}$	$\langle Z \rangle - Z_m$
$t - \min(t_m)$	$\min(t_{m1}) - \min(t_{m2})$	$\sum Q_m / \sum Q$
$Q / \sum Q_m$	$\frac{\langle Q_{m1} \rangle - \langle Q_{m2} \rangle}{\langle Q_{m1} \rangle + \langle Q_{m2} \rangle}$	$\langle t_m \rangle$
	$D^{-1}_{m1 \to m2}$	
	$\mathbb{A}$	
	$\mathbb{B}$	

TABLE 5.2 – Features on the edges on the graph. It use the same notation as in table 5.1. $D^{-1}_{m1 \to m2}$ is the inverse of the distance between the mesh $m1$ and the mesh $m2$. The features $\mathbb{A}$ and $\mathbb{B}$ are detailed in Section 5.1

Since our different nodes do not have the same number of features, they exist in distinct spaces. Traditional graph neural networks only handle homogeneous graphs, where the nodes and edges have the same number of features at each layer. Therefore, the libraries and publicly available algorithms we found were not suited to our needs. As a result, we had to develop and implement a custom message-passing algorithm capable of handling our heterogeneous graph.

5.2 Message passing algorithm

The message passing algorithm define the way the GNN will compute and update its graph. As it is detailled in Section 3.2.3, the message-passing algorithm allows each node in the graph to update its features based on information from its neighboring nodes. This update process enables the network to propagate information through the graph, allowing nodes to gradually integrate knowledge about the entire detector. This step is crucial for ensuring that each node can take into account not only its local neighborhood but also the broader context of the event.

As introduced in previous section and in the tables 5.1 and 5.2, our graphs nodes and edges will have different number of features depending on their nature, meaning that we cannot have a single message passing function. We thus need to define a message passing function for each transition inside or outside a family. Using the notation presented in Section 3.2.3:

$$n_i^{k+1} = \phi_u(n_i^k, \square_j \phi_m(n_i^k, n_j^k, e_{ij}^k));\ n_j \in \mathcal{N}_i' \tag{5.5}$$

and denoting the mesh nodes M, the fired nodes F and the I/O node IO, we need to define

$$\begin{gathered}
\phi_{u;F\to M}\ ;\ \phi_{m;F\to M} \\
\phi_{u;M\to F}\ ;\ \phi_{m;M\to F} \\
\phi_{u;M\to M}\ ;\ \phi_{m;M\to M} \\
\phi_{u;M\to IO}\ ;\ \phi_{m;M\to IO} \\
\phi_{u;IO\to M}\ ;\ \phi_{m;IO\to M}
\end{gathered}$$

to update the nodes after each layers. Following the illustration in Figure 5.3, for each transition between families or inside a family we need an aggregation, a message and an update function. For the aggregation, we use the sum. We use the same, simple, formalism for every ϕ_u:

$$\phi_u \equiv I_{i'}^{n'} = I_i^n A^i_{i',e} W_n^{e,n'} + I_i^n S_n^{n'} + B^{n'} \tag{5.6}$$

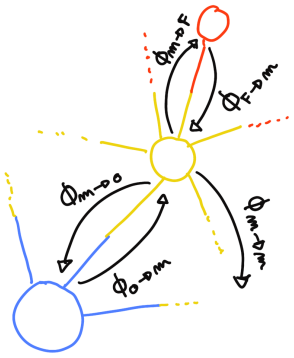


FIGURE 5.3 – Illustration of the different update function needed by our GNN

using the Einstein summation notation. The second order tensor, or matrix, I_i^n is holding the nodes informations with i the node index and n the feature index. n represent the features of the previous layer and n' the features of this layer.

$A_{i',e}^{i}$ is the adjacency tensor, discussed in the previous section, representing the edges between the node i' and the node i, each edges holding the features indexed by e. If the edge does not exist, the features are set to 0. This choice is justified by the linearity of the operation in equation 5.6 : whatever the weights, when multiplied by 0 the results is 0 and the sum result is unchanged.

The learnable parameters are composed of:

- The third order tensor $W_n^{e,n'}$ which represent the passage from the previous combined feature space between the node and the edge features $n \otimes e$, the previous layer, to the current space n', this layer.
- The first order tensor $B^{n'}$ which is a learnable bias on the new features n'.
- The second order tensor $S_n^{n'}$, which can be viewed as a self loop relation where the node update itself based on the previous layer informations, going from the previous space n to the current space n'.

If a node have neighbours in different families, the different IAW coming from the different families are summed.

$$I' = \sum_{\mathcal{N}} \left[I_{\mathcal{N}} AW \right] + IS + B \tag{5.7}$$

where $\mathcal{N}$ are the neighbouring family. In our case, dropping the tensor indices and indexing by family for readability, we get

$$I'_F = I_M A_{M\to F} W_{M\to F} + I_F S_F + B_F \tag{5.8}$$
$$I'_M = I_F A_{F\to M} W_{F\to M} + I_M A_{M\to M} W_{M\to M} + I_{IO} A_{IO\to M} W_{IO\to M} + I_M S_M + B_M \tag{5.9}$$
$$I'_{IO} = I_M A_{M\to IO} W_{IO\to M} + I_{IO} S_{IO} + B_{IO} \tag{5.10}$$

We thus have a S, W and B for each of the ϕ_u function we defined above. The IAW sum can be seen as the ϕ_m function and $IS + B$ as the second part of the ϕ_u function. Eq 5.5 gave the generic form of message passing : to update a node i, one first combines informations from the surrounding nodes and edges and then combine the result ($\square_j \phi_m$) with the current features of node i. Many practical ways to combine can be tried. In our implementation of message passing (Eq. 5.6 and 5.7) the latter combination is the simple sum of the former (IAW, the equivalent of $\square_j \phi_m$) with a linear combination of the current features of node i ($IS + B$).

Interestingly, the number on learnable weight in those layer is independent of the number of nodes in each family and depends solely on the number of features on the nodes and the edges.

The expression above only update the node features. We could update the edges, using the results of ϕ_m for example, but for technical simplicity we only update the nodes and keep the edges constant. Preserving the edges after each layers allow to share the adjacency matrix between all layers, saving memory and computing time.

This operation of message passing is the constituent of our message passing layers, designed in this work as *JWGLayer*, each of them owning there own set of parameter W, S and B. To those layers, we can adjoin an activation function such as $PReLU$

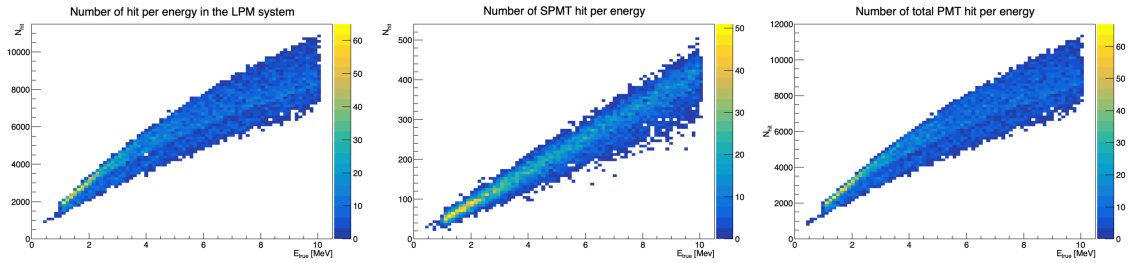
$$I' = PReLU\left(\sum_{\mathcal{N}}\left[I_{\mathcal{N}}AW\right] + IS + B\right) \tag{5.11}$$

5.3 Data

The dataset consists of 1M simulated positron events from the JUNO official simulation version J23.0.1-rc8.dc1. This version of the simulation incorporates both the physics of the detector and its electronics, ensuring that the events closely reflect real detector conditions. Importantly, this version includes advanced digitization and trigger modeling, making it suitable for testing the reconstruction capabilities of our GNN model. Those events are uniformly distributed in energy with $E_k \in [0, 9]$ MeV and distributed in the detector.

All the event are *calib* level, with simulation of the physics, electronics, digitizations and triggers. 900k events will be used for the training, 50k for validation and loss monitoring and 50k for the results analysis in Section 5.7. Each events is between 2k and 12k fired PMTS, resulting in fired nodes being the largest family in our graphs in all circumstances as illustrated in Figure 5.4c.

As expected, by comparing the scale between the Figure 5.4a and 5.4b we see that the LPMT system is predominant in term of informations in our data. The number of PMT hits grow with energy but do not reach 0 for low energy event due to the dark noise contribution which seems to be around 1000 hits per event for the LPMT system (left limit of Figure 5.4a) and around 15 hits per event for the SPMT system (left limit of Figure 5.4b) which is consistent with the results show in Section 4.1.2.

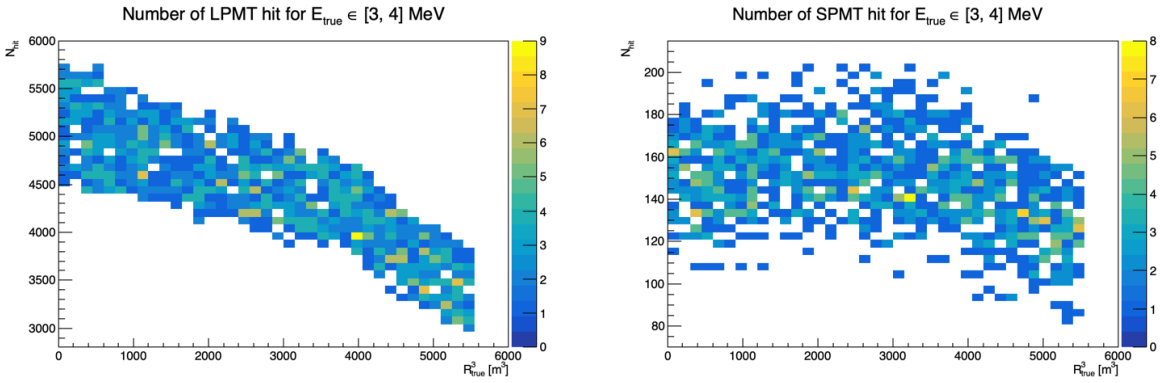


(A) (B) (C)

FIGURE 5.4 – Distribution of the number of hits depending on the energy. **On the right:** for the LPMT system. **In the middle :** for the SPMT system. **On the left:** For both system.

The structure seen in the distribution in Figure 5.4a comes from the shape of the number of hits depending on the radius as shown in Figures 5.5a and 5.5b where the number of hit decrease with radius. It is important to understand that this is not representative of the number of PE per event and the decrease in hits over the radius means that the PE are just more concentrated in a smaller number of PMTs.

No quality cut is applied here, we rely only on the trigger system. It means that event that would not



(A) (B)

FIGURE 5.5 – Distribution of the number of hits depending on the radius. **On the right:** for the LPMT system. **On the right :** for the SPMT system. To prevent the superposition of structure of different scales we limit ourselves to the energy range $E_{true} \in [0, 9]$.

trigger are not present in the dataset but for events that triggered twice, it happens rarely, the two trigger are considered as two separate event.

5.4 Model

In this section, we discuss the different layers that compose the final version of the model. The number of layers, their dimensions, and their arrangement were fine-tuned through multiple iterations. As mentioned earlier, each JWGLayer is defined by the number of features on the nodes and edges of the output graph, assuming it takes as input the graph from the previous layer. For simplicity, when discussing a graph configuration, it will be presented as follow: { N_f, N_m, N_{IO}, $N_{f\to m}$, $N_{m\to m}$, $N_{m\to f}$ } where

- N_f is the number of feature on the fired nodes.
- N_m is the number of features on the mesh nodes.
- N_{IO} is the number of features on the I/O node.
- $N_{f\to m}$ is the number of features on the edges between the fired and mesh nodes.
- $N_{m\to m}$ is the number of features on the edges between two mesh nodes.
- $N_{m\to f}$ is the number of features on the edges between the mesh nodes and the I/O node.

Because we do not change the number of features on the edges, we can simplify the notation to {N_f, N_m, N_{IO}}. As an example, the input graph configuration, following the tables 5.1 and 5.2 is {6, 8, 13, 5, 8, 5 } or, without the edge features, {6, 8, 13 }.

The final version of the model, called JWGv8.4.0 is composed of

- An JWGLayer, converting the input graph {6, 8, 13 } to { 64, 512, 2048 } with a PReLU activation function.
- 3 resnet layers, each of them composed of
  1. 2 JWG layers with a PReLU activation function. They do not change the dimension of the graph
  2. A sum layer that sums the features in the input graph with the one computed from the JWG layers
- A flatten layer that flatten the features of the I/O and mesh nodes in a vector.
- 2 fully connected layers of 2048 neurons with a PReLU activation function.
- 2 fully connected layers of 512 neurons with a PReLU activation function.

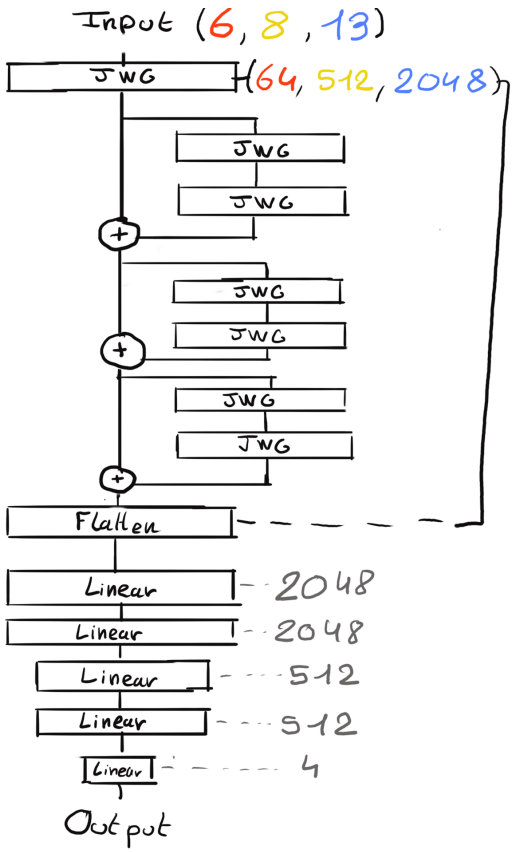


FIGURE 5.6 – Schema of the JWGv8.4.0 architecture, the colored triplet is the graph configuration after each JWG layers

— A final, fully connected layer of 4 neurons acting as the output of the network.
A schematic of the model is presented in Figure 5.6.

We use the Mean Square Error (MSE) for the loss

$$\mathcal{L} = (E_{rec} - E_{dep})^2 + (X_{rec} - X_{true})^2 + (Y_{rec} - Y_{true})^2 + (Z_{rec} - Z_{true})^2 \tag{5.12}$$

as it was the best resulting loss in Chapter 4.

5.5 Training

The optimizer used for training is the Adam optimizer (see Section 3.1.3) and default hyperparameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$) with a learning rate $\lambda =$ 1e-8. The training last 200 epochs of 800 steps. We use a batch size of 32, the largest we can have with 40GB of GPU ram. The learning rate is constant during the first 20 epochs then exponentially decrease with a rate of 0.99. We save two set of parameters, the set of parameters the set that yield the lowest validation loss and the set of parameters at the end of the training. The validation is computed over a single batch.

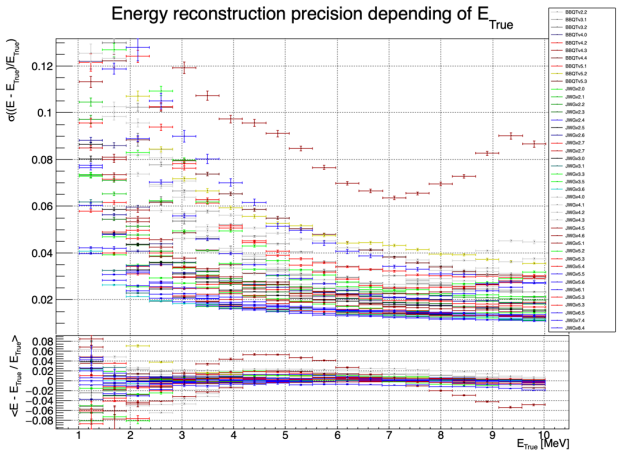


FIGURE 5.7 – Energy reconstruction depending on the true energy for samples of the different versions of the GNN

5.6 Optimization

The GNN model presented in previous sections is the result of a long work of optimization. Indeed, the innovative architecture we propose left us with an infinity of possible configurations with no guidance from prior works in literature nor in JUNO.

In the end, more than 60 different configurations have been tested. This effort is illustrated on Figure 5.7[1], where the 40 configurations are compared in their ability to reconstruct the positron energy. Although all configurations share the fundamental principles we base our innovative architecture on (three different kinds of nodes and edges, usage of raw level features on some of them, usage of higher level data on others, division of JUNO's surface into regional pixels to form mesh nodes, the very large number of edges connected to each mesh node, etc.), performances can vary a lot between our first attempts (far beyond any acceptable energy resolution, and not even on this figure) and recent ones. Therefore: the precise way to choose hyperparameters mattered a lot, regardless of the relevance of the global architectural principles.

The spectacular improvement between early and later configurations also explains the length of this process : for long we hoped we would finally reach the classical performance, and it was tempting to test yet another configuration.

5.6.1 Software optimization

A substantial effort was devoted to the data processing workflow. Transforming JUNO simulation outputs into graphs is a computationally expensive task. Furthermore, due to the ever-changing nature of the graph dimensions and features during optimization, preprocessing JUNO's files by precalculating the graphs and then reading them from files was not viable, as it would require a large amount of disk space to store events for each version of the graph.

Therefore, the software does not rely on preprocessed data and instead computes the observables, adjacency matrix, etc., during training. This data processing is performed in parallel on the CPU. The raw data comes from ROOT files produced by the collaboration software, and the Event Data Model (EDM), used internally by the collaboration [72], had to be interfaced with our software, an interface that had to be maintained as the collaboration's software evolved. For the harmonic

1. Note that this figure was prepared on idealized data with no dark noise and perfect hit time determination.

power calculation, we migrated from the Healpix library to Ducc0 [73] for more precise control over multithreading.

5.6.2 Hyperparameters optimization

The first kind of hyper-parameters that received a lot of effort concern the network's detailed architecture:

- Message passing layers where originally not JWG layers, we started by using small FCDNN in place of ϕ_u and ϕ_m. Due to low performances and memory consumption issues, we pivoted to the message passing algorithm presented in Section 5.2.
- The ResNet architecture was brought after issue with the gradient vanishing.
- The number of layers was varied between 5 and 12.
- The number of node features after each given message passing layer (64, 512, 2048 in the final version) was varied.
- The Final FCDNN after the message passing layers is not present in all versions.
- At some point, the PReLU activation function replaced the ReLU function.

For some of them, software work was necessary. In any case, each configuration required a training of about 90h. Adding the analysis time necessary to the verification of its performance and the comparison with other versions, one understands the number of tests had to be limited.

Other hyperparameters were also tested :

- The higher level variables described in Section 5.1 (powers of various spherical harmonics, $\mathbb{A}$, $\mathbb{A}$, $(Q_{m1} - Q_{m2})/(Q_{m1} + Q_{m2})$) were added progressively. Notice that our choice to focus our search on this kind of variables is also due to the fact that JWGLayer involves linear operations. It is therefore difficult for such a network to propose variables of this kind among the node features learned layers after layers (i.e. it's difficult for the network to understand these variables are important, or only after many layers).
- Time allocated to training, the Learning Rate, the size of batches, etc.
- The number of pixels (ie of mesh nodes) was varied between 192 and 768.
- Several definitions loss functions where tried. In particular, we tried some focussed only on the E resolution, only on the vertex resolution (R) or trying to optimize both.

To make a long story short, each new configuration was the result of our reflections after having analysed the previous configurations, or after having thought over again about JUNO's detailed response to energy deposits – seeking for variables that could help the GNN.

Another, quite common, approach was in principles possible : a random search. However, due to the extensive training time, up to 90h per training, the heavy memory consumption of the models that would often exceed the 20GB limit of the V100, this approach was not realistic in our case, though we were able to extend the memory limit to 40GB thanks to a local A100 GPU card available at Subatech.

5.7 performance of the final version

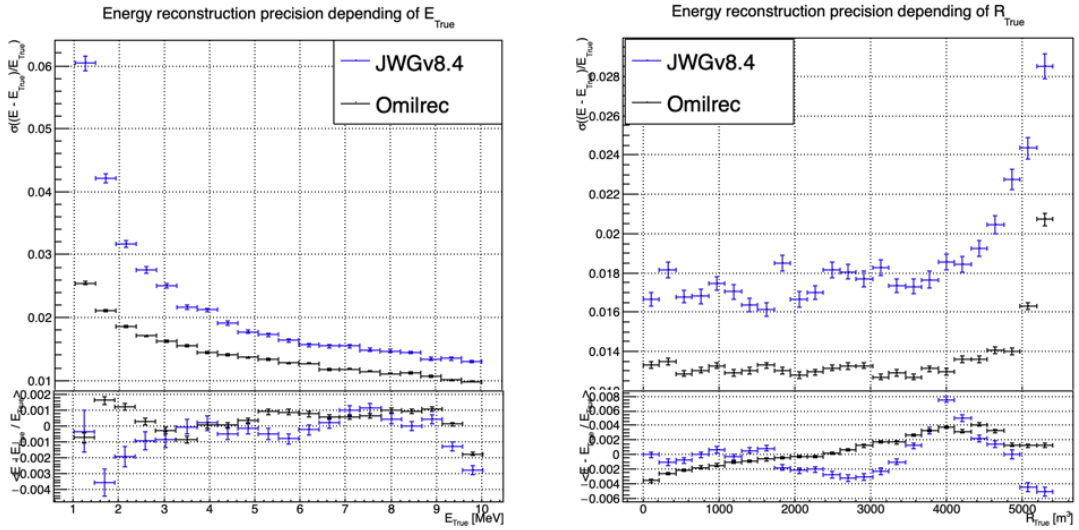
The reconstruction performance of "JWGv8.4" are presented in Figures 5.8, 5.9, 5.10 and compared to the "Omilrec" algorithm, the official IBD reconstruction algorithm in JUNO. Omilrec is based on the QTMLE reconstruction method that was presented in Section 3.3.

This comparison required to use a consistent definition of E_{true}. This is not trivial since at JUNO, ML method reconstruct the true energy deposited by the positron+annihilation gammas (that's the

target implemented in the loss function), while Omilrec, which is based on probabilities to observe a given number of PE in a given PMT, reconstruct the "visible energy". It reflects the total number of radiated and detectable scintillation or Cherenkov photons (and is subject to non linear effects like quenching).

The conversion we use to obtain comparable E_{true} is explained in Appendix C.

On Figures 5.8 to 5.10, we notice that the best GNN does not match the performance of the OMILREC algorithm. Generically, Energy resolution is 50% worse, while the resolution on R is three times worse. Reconstruction biases are not better either with the GNN. We have tried to understand the origin of this limited performance.



(A) Resolution and bias of energy reconstruction vs energy

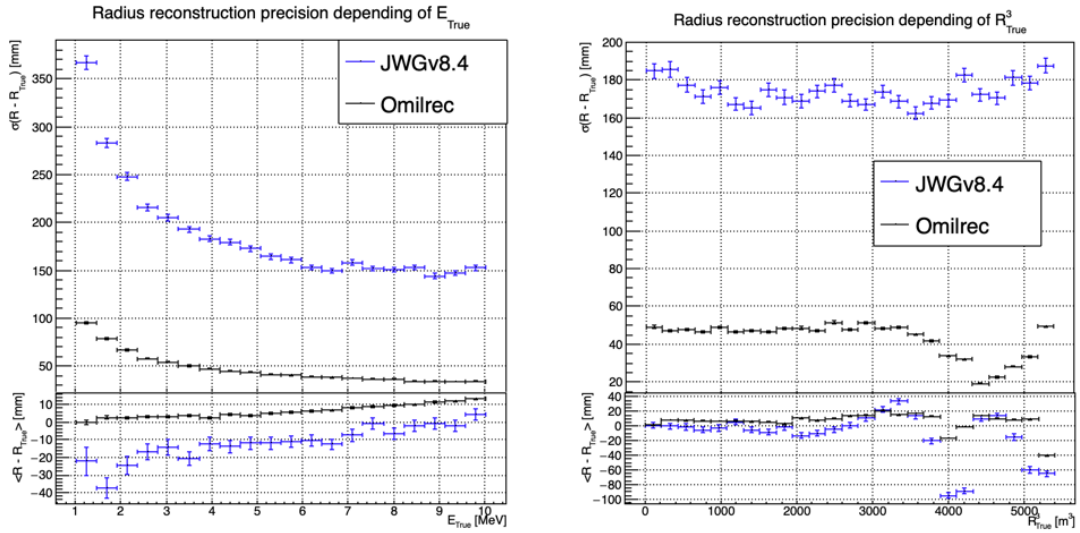
(B) Resolution and bias of energy reconstruction vs radius

FIGURE 5.8 – Reconstruction performance of the Omilrec algorithm based on QTMLE presented in Section 3.3, JWGv8.4 presented in this chapter. The top part of each plot is the resolution and the bottom part is the bias.

The first action that can be carried out in this direction was to determine if some information used by OMILREC was not used properly by JWGv8.4. For that purpose, we used again the approach presented in Chapter 4 (Sec 4.3.2 and annex A) to combine JWGv8.4 and OMILREC. We observe on Figures 5.11 and 5.12 that this combination brings no sizeable improvement of the best of the two combined methods. The combination remains very close to OMILREC alone. This is an indication that JWGv8.4 does not use informations that would be overlooked by OMILREC, and that on the contrary, that's JWGv8.4 that fails to use properly important informations.

The problem described above could be inherent to our GNN's original architecture. Discussions with JUNO's colleagues when these results were presented at the collaboration pointed to the role of PMT time information (t, in the (Q, t) pairs we use as our algorithm input features). The thousands of values found in the *fired* nodes might not be aggregated well enough when transmitted to the mesh nodes, causing a loss in the redundancy of this important information.

We tested this idea in several manners, described below.



(A) Resolution and bias of radius reconstruction vs energy

(B) Resolution and bias of radius reconstruction vs radius

FIGURE 5.9 – Reconstruction performance of the Omilrec algorithm based on QTMLE presented in Section 3.3, JWGv8.4 presented in this chapter. The top part of each plot is the resolution and the bottom part is the bias.

Finer granularity

We tried to recover some redundancy by increasing the number of mesh nodes from 198 to 768. The improvement we observed was small, and did not allow to get close to OMILREC's performance.

To explore further in this direction, we would ideally try 3072 pixels (the next HEALPIX rank). However, this is not possible for our GNN due to hardware limitations, mainly the available GPU memory. Instead, we discussed the problem with Gilles Grasseau, calculus research engineer with whom we collaborate on the subject of ML reliability (see Chapter 6). In the framework of this activity, Gilles needs to develop reconstruction algorithms to be "attacked" by a prototype Adversarial NN. One of them is a pseudo-spherical CNN using oriented filters, called HCNN.

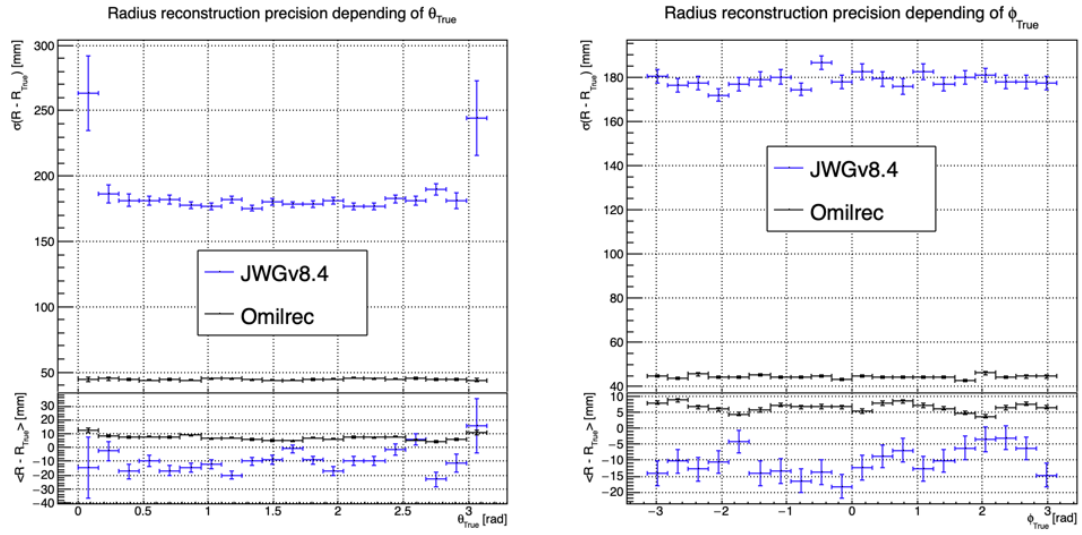
To produce its input image, this algorithms split the Sphere into 3072 pixels. Each channel of this image is an aggregation of the (Q, t) values found in all the PMTs. The charge are summed and the lowest time is kept. The performance of this algorithm can be seen on Figures 5.13 and 5.14, compared to OMILREC. With 3072 pixels, the performance of HCNN does not match that of OMILREC, but is closer to it than our GNN. The granularity of the pixels, and the way to summarize the individual PMTs information when going from 17000 LPMTs to only 3072 pixels indeed seems to play a role.

This is consistent with the results obtained by the first GNN tried at JUNO on reactor neutrinos (already described in Section 3.3.3). It used 3072 pixels, and also obtained an uncompetitive R reconstruction.

Information reduction, from fired to Meshes

The problem described above is somehow classical. ML algorithms, ideally, would start from the full information present in the detector, and learn to reduce it optimally.

In cases where only 3072 pixels can be used instead of the complete information from 17000 PMTs,



(A) Resolution and bias of radius reconstruction vs θ (B) Resolution and bias of radius reconstruction vs ϕ

FIGURE 5.10 – Reconstruction performance of the Omilrec algorithm based on QTMLE presented in Section 3.3, JWGv8.4 presented in this chapter. The top part of each plot is the resolution and the bottom part is the bias.

one needs to understand how to combine the individual from the 5 or 6 PMT found in each pixel into pixel-level features, without loosing important information.

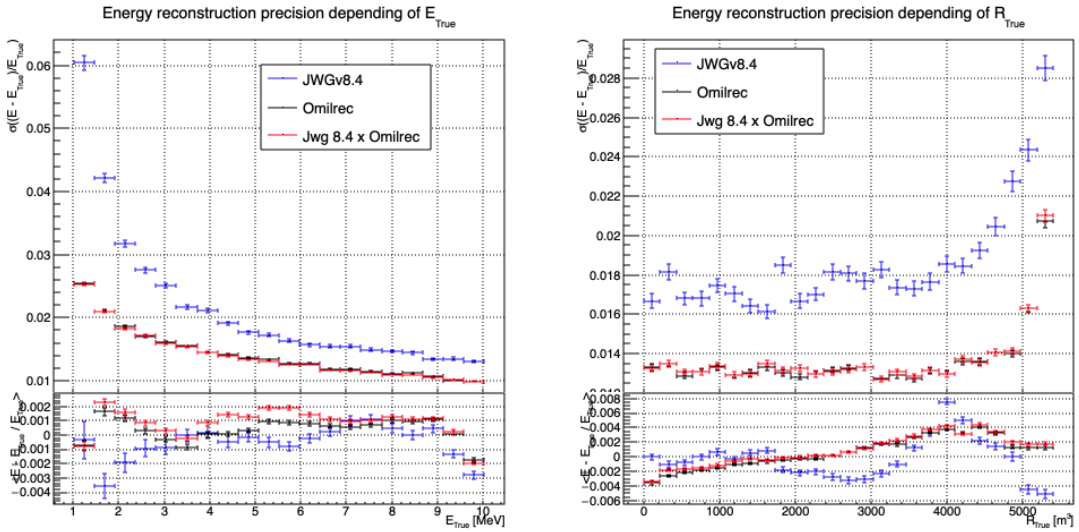
In the case of our GNN, we hoped that by connecting each mesh node to its corresponding 5 or 6 fired nodes, we could keep the full information. In reality, it seems that the message passing between fired and mesh does not work efficiently. When nodes are updated by the first (may be also by the subsequent) layer, the new mesh features might be dominated by the original features in the second column of tables 5.1, themselves a simple version of aggregation. Layer after layer, we might be limited to that level of time information, lacking time redundancy.

We have verified this by testing version of the GNN in which the link between fired and mesh was cut, or in which no time info was included among the fired nodes features. It had only a small effect which seems to confirm a problem in the way the full information, from all the individual PMTs, is used by our GNN.

Possible improvements

It appears that the network is unable to aggregate the timing information correctly. While this could be addressed by using a finer segmentation, with more mesh nodes, improvements might also arise from refining the message-passing algorithm. The algorithm presented in this thesis is still quite basic, relying on a simple linear combination of features. We have seen through examples in CNNs, GNNs, and other architectures, both in research and industry, that specializing the network — for instance, by incorporating convolutional filters — can lead to improvements that were previously unattainable with simpler FCDNNs. Applying this approach to the message-passing algorithm, by utilizing a GNN with a more advanced message-passing, could yield better results.

We could investigate alternative aggregation strategies, for example, by weighting the timing information more significantly during the message-passing phase. Additionally, testing a non-linear combination of features from fired to mesh nodes could help preserve more granular information. Another potential improvement would be to introduce attention mechanisms that dynamically as-



(A) Resolution and bias of energy reconstruction vs energy

(B) Resolution and bias of energy reconstruction vs radius

FIGURE 5.11 – Reconstruction performance of the Omilrec algorithm, JWGv8.4 and the combination between the two using the optimal variance estimator presented in annex A.2. The top part of each plot is the resolution and the bottom part is the bias.

sign more importance to relevant features in the fired nodes

Regarding the timing information, we provided high-level features, assuming this would assist the neural network in converging to the solution. However, by offering such information upfront, the GNN might be taking the "easy" path, settling for a local and broader minimum, rather than extracting the features that could lead to better performance.

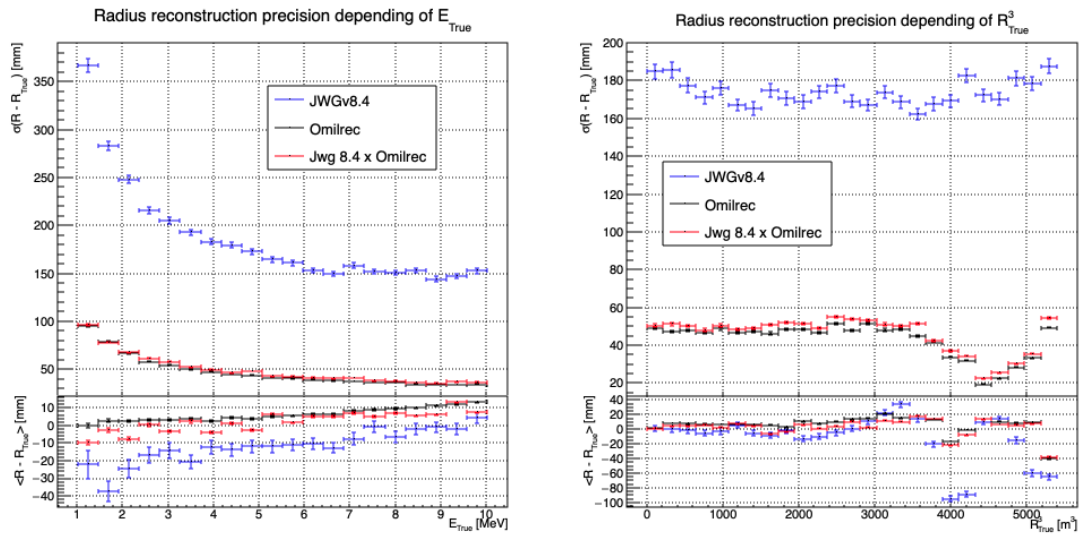
If there are difficulties in transferring information between the fired and mesh nodes, it may stem from the way we connected the fired nodes to the mesh nodes. By linking the fired nodes within the same mesh, or even connecting the fired nodes of neighboring mesh nodes, the GNN might be able to construct more meaningful information.

Finally, by providing directly the PMT waveform to the GNN, in the fired nodes, we could search for even finer precision and results. An idea would be to specialise the message function $\phi_{m;F\to M}$ to be a 1D convolutional layer over the waveform. The resulting channels would be fed to the mesh nodes for their updates.

5.8 Conclusion

To achieve its scientific goals, JUNO requires a precise and well-understood reconstruction, as it needs an energy resolution of 3% at 1 MeV. Even small, unaccounted biases could make it impossible to determine the mass ordering, as explored in Chapter 7. A likelihood-based algorithm, designed to meet JUNO's requirements and referred to as the classical algorithm, was developed and is detailed in Section 3.3.

Machine learning algorithms were developed to challenge this classical approach, and they are presented in Section 3.3.3. Although they achieve the precision of the classical algorithm, they do not offer significant improvements. The GNN previously developed is a convolutional GNN where



(A) Resolution and bias of radius reconstruction vs energy

(B) Resolution and bias of radius reconstruction vs radius

FIGURE 5.12 – Reconstruction performance of the Omilrec algorithm, JWGv8.4 and the combination between the two using the optimal variance estimator presented in annex A.2. The top part of each plot is the resolution and the bottom part is the bias.

nodes correspond to pixels, connected to their neighbors based on the Healpix [71] segmentation, with the (Q, t) information aggregated onto these pixels.

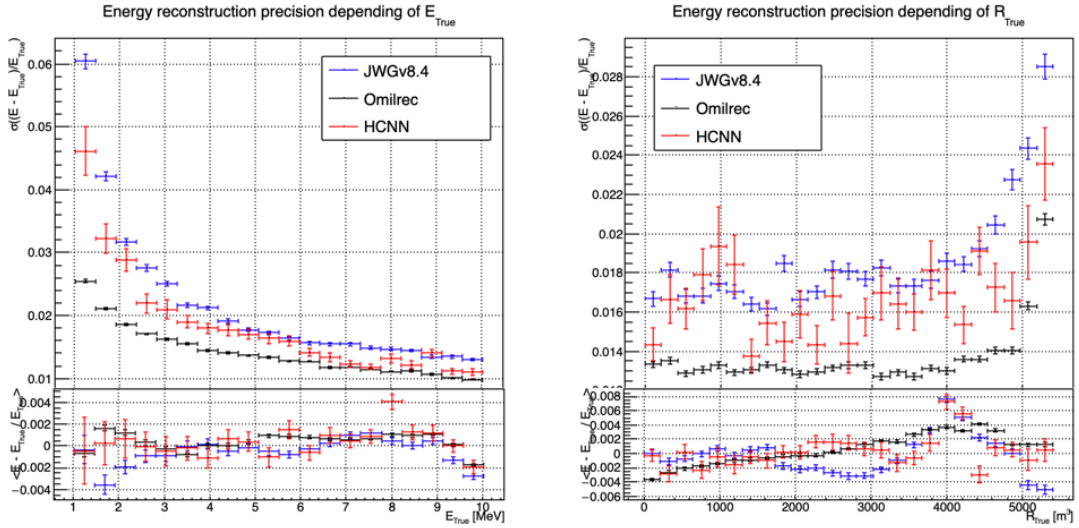
In this chapter, we introduce a novel and innovative architecture. In addition to the pixel segmentation represented by mesh nodes, we incorporate rawer information by directly representing the fired PMTs as nodes. We also fully connect the mesh nodes to each other, hoping to facilitate the transfer of information. Finally, we introduce a global node that holds global information about the detector.

These three types, or families, of nodes do not have the same number of features, resulting in a heterogeneous graph. Publicly available algorithms for graph processing are designed for homogeneous graphs, so we had to develop a custom algorithm adapted to heterogeneous graphs.

This GNN required significant technical development, but the results are not at the level of the classical algorithm. The tests we conducted suggest that the problem may lie in the aggregation of raw information from the fired nodes onto the mesh nodes, as removing the fired nodes does not degrade the results. Additionally, due to technical constraints, we had to reduce the number of pixels compared to the previous GNN. Other algorithms we developed, which use a higher pixel resolution, outperform this architecture, reinforcing our suspicion that the aggregation is the root of the issue.

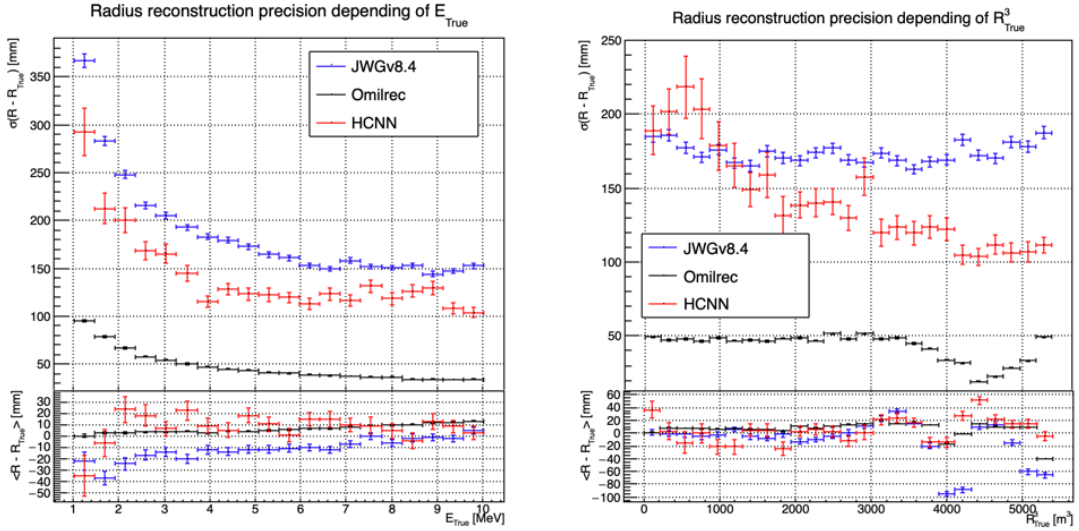
The precision required for JUNO's scientific objectives, particularly in determining mass ordering, imposes stringent constraints on reconstruction algorithms. Small biases or errors in energy resolution could significantly affect the experiment's outcomes. Future improvements may involve refining the message-passing algorithm, incorporating additional detector-specific features, and experimenting with more advanced architectures such as attention-based GNNs to further reduce reconstruction errors.

Perhaps by incorporating rawer information, such as the waveform, refining the message-passing algorithm, or adjusting the features on the different nodes, we could match the precision of the classical algorithm. However, it is also possible that deeper, more radical changes are needed to become competitive.



(A) Resolution and bias of energy reconstruction vs energy (B) Resolution and bias of energy reconstruction vs radius

FIGURE 5.13 – Reconstruction performance of the Omilrec algorithm based on QTMLE presented in Section 3.3, JWGv8.4 presented in this chapter and the HCNN algorithm. The top part of each plot is the resolution and the bottom part is the bias.



(A) Resolution and bias of radius reconstruction vs energy (B) Resolution and bias of radius reconstruction vs radius

FIGURE 5.14 – Reconstruction performance of the Omilrec algorithm based on QTMLE presented in Section 3.3, JWGv8.4 presented in this chapter and the HCNN algorithm. The top part of each plot is the resolution and the bottom part is the bias.

Chapter 6

Reliability of machine learning methods

"Psychohistory was the quintessence of sociology; it was the science of human behavior reduced to mathematical equations. The individual human being is unpredictable, but the reactions of human mobs, Seldon found, could be treated statistically"

Isaac Asimov, Second Foundation

Contents

As explained in previous chapters, JUNO is a precision experiment where the complete understanding of the effects at hand is crucial. As it will be illustrated in Chapter 7, even small invisible biases or uncertainties could lead to the imposibility to run the measurements, or even worse, wrong our mass ordering measurements. While the liquid scintillator technology is well none and straightforward, this is the first time it is deployed to such scale, and for such precision. This novelty bring its fair share of elements, effects or assumption, that, if they were to be overlooked, could cause issue.

We already shown a large variety of reconstruction algorithms, OMILREC for LPMT reconstruction in Section 3.3, numerous machine learning algorithms in section 3.3.3 and our own work in chapters 4 and 5. Those algorithms were compared to each others based on their performance as in [57] but we are the first that looked into the correlation between the reconstruction. The combinations of algorithms shown in Chapter 4 show that some information elude the algorithms. We used this fact to try to improve our performance but this could also lead the algorithm to being vulnerable to some effect that could affect the detector and wrong the measurements.

The search for such effect could be done by hand, but the process would be tedious. We propose in this thesis a machine learning method to probe for those effects. In Section 6.1, I describe the method behind the algorithm. In section 6.2 I detail the architecture of our algorithm and in section 6.4 the results of it. Finally, in section 6.5, I conclude and discuss about the prospect and possible improvements to bring to this work.

6.1 Method

As introduced above, JUNO needs a very good understanding of the biases and effects affecting its reconstruction as a small bias could wrong the mass ordering measurement. To calibrate those biases and effect, JUNO rely on multiples sources that can be located at various point in the detector. The calibration strategy was already discussed in Section 2.4 and show calibrations sources of gammas, neutrons and positrons, with the catch that the positrons will annihilate inside the encapsulation and only the two 511 keV gammas will be seen.

None of the calibrations sources considered are positron event. While electrons and positrons events should be pretty similar in their interaction with the electronic cloud of the LS atoms, electron events are missing the two annihilations γ and the potential of forming a positronium [74]. The topology of the event thus differ of the order of magnitude of our reconstruction performance. A few nanoseconds between the energy deposit and the positronium annihilation against a time transit spread between 3 and 6 ns depending on the PMT type [75–77]. The γ from the positron annihilation will travel distances of the order of magnitude of the typical LPMT resolution of 8 cm (see Section 3.3).

Another natural calibration source is the ^{12}B spectrum. The ^{12}B is a cosmogenically produced isotope through the passage of muons inside the LS. The ^{12}B decays via β^- emissions with a Q value of 13.5 MeV with more than 98% of the decay resulting in ground state ^{12}C. The ^{12}B event will be cleanly identified by looking for delayed high energy β events after an energetic muon. Due to its natural causes, the ^{12}B events will be uniformly distributed in the detector. The calibration strategy consist in fitting the energy spectrum of ^{12}B with the results of the simulation to adjust the simulation parameters. Both sources will be used to *control* the response of the detector.

Unlike lasers and radioactive, from which the localization and energy will be well known, the individual truth of ^{12}B will be unknown with only the localisation loosely constrained by the muon track. Only higher order observables such as the energy distribution will be accessible.

All of those considerations could hide potential unknown or undetected effect that could lead to issue in the mass ordering analysis. But, while we have idea from where the issue could come, the production by hand of event perturbations that go unseen in the calibration would be tedious. That's why we propose to use an Adversarial Neural Network (ANN) to produce those perturbations if they exists. A schematic of the concept is presented in Figure 6.1.

This network should produce physically sound perturbation, that would not be seen by the calibration but also by the visualisation of the event. If the ANN manage to produce such perturbations, we can derive systemic uncertainties from it. If it fail to find some, it is a proof of robustness for the attacked reconstruction method.

For this study we consider a "physics" dataset composed of 1M positron events from J23, uniformly distributed in the Central Detector (CD) and in deposited energy between $E_{dep} \in [1.022; 10.022]$. This set represent the IBD events we want to *wrongly* reconstruct.

We use a second "control" dataset of electron event also uniformly distributed in the detector and over the same energy range. They mimic the energy deposit of ^{12}B decay and are used as the sample to compute the control observables.

6.2 Architecture

We can describe the goal of the ANN by the following loss function:

$$\mathcal{L} = \mathcal{L}_{adv} + \mathcal{L}_{reg} \tag{6.1}$$

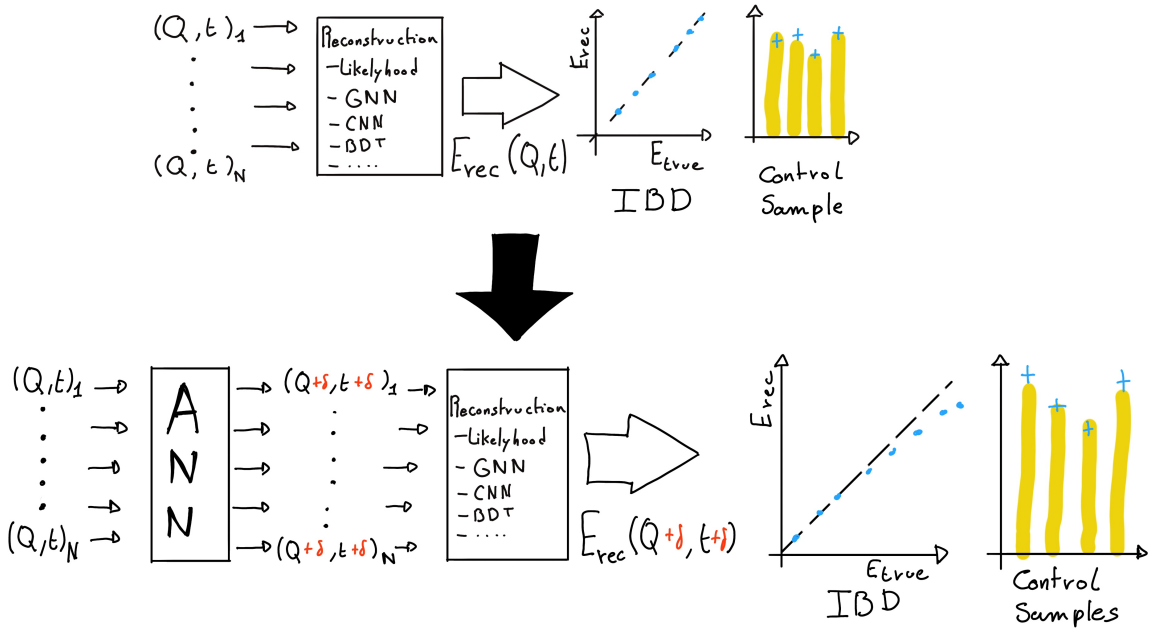


FIGURE 6.1 – Schema of the method to discover vulnerabilities in the reconstruction methods. **On the top** of the image, the standard data flow. The individual charge and times are fed to a reconstruction algorithm. From the reconstructed energies, we can produce an IBD spectrum and compute control observables from the control samples. **On the bottom**, the same data flow but we add an ANN between the input and the reconstruction. The ANN will slightly change the input charge and time so the reconstruction algorithm inaccurately reconstruct the IBD energy, but the perturbation is not visible in the control sample.

where $\mathcal{L}_{adv}$ is the adversarial loss, which is minimal when the reconstruction is "broken". We thus need to define what is a *wrong* reconstruction. We choose to define it via the correlation between the reconstructed and deposited energy

$$\mathcal{L}_{adv} = |\text{Corr}(E'_{rec}, E_{rec})| \tag{6.2}$$

where E'_{rec} and E_{rec} are the reconstructed energies after and before perturbation respectively. This loss is positive or null and is minimal when the reconstructed energy after perturbation is decorrelated with the original reconstruction.

The term $\mathcal{L}_{reg}$ is the regularisation term, which is minimal when the control variable are correctly reconstructed

$$\mathcal{L}_{reg} = \sum_{\lambda} (O^{rec}_{\lambda} - O^{th}_{\lambda})^2 \tag{6.3}$$

where λ index the different control observables that will be considered in this study. It's minimal when the control observables after perturbation O^{rec}_{λ} are coherent with their expected values O^{th}_{λ}. In this exploratory work, we choose as the control observable the difference between the reconstructed position and energy and the ground truth from the Monte Carlo simulation complemented with a penalty term P

$$\mathcal{L}_{reg} = \sum_{\lambda \in \{x,y,z,E\}} (\lambda_{rec} - \lambda_{true})^2 + P \tag{6.4}$$

This penalty P is here to prevent the ANN from producing event too different from the initial event. It will be further detailed in Section 6.2.3 .

We see that the final loss is the equilibrium between the adversarial and regularisation loss.

6.2.1 Back-propagation problematic

We would like this method to be applicable to any kind of reconstruction algorithm but this complicated considering standard training method through backward-propagation, discussed in details in Section 3.1.3. For explanation, let's define the application of the reconstruction algorithm as $\mathcal{F}$ on an event X, resulting in the prediction Y and the application of the ANN $\mathcal{G}$ on X to give a perturbed event X', we can parametrize the equation 6.1

$$Y = \mathcal{F}(X);\ Y' = \mathcal{F}(X') = \mathcal{F}(\mathcal{G}(X)) \tag{6.5}$$

$$\mathcal{L} \equiv \mathcal{L}(\mathcal{F}(\mathcal{G}(X)), Y_t) \tag{6.6}$$

where Y_t is the reconstruction target of Y.

Now if we consider a parameter $\boldsymbol{\theta}$ of the ANN on which we want to optimize $\mathcal{L}$, in the backward-propagation optimisation framework we need to compute

$$\frac{\partial \mathcal{L}(\mathcal{F}(\mathcal{G}(X)))}{\partial \boldsymbol{\theta}} \tag{6.7}$$

which, when using the chain rule, become

$$\frac{\partial \mathcal{L}(\mathcal{F}(\mathcal{G}(X)))}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{G}}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \mathcal{F}}{\partial \mathcal{G}} \cdot \frac{\partial \mathcal{L}}{\partial \mathcal{F}} \tag{6.8}$$

The terms $\frac{\partial \mathcal{G}}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \mathcal{L}}{\partial \mathcal{F}}$ are easily computable but $\frac{\partial \mathcal{F}}{\partial \mathcal{G}}$ depends on the nature of the reconstruction algorithm. While it comes naturally when using NN algorithms, it's not so trivial for other kind of algorithms like likelihood. Solutions exists to optimize networks that work in complex, non differentiable environments, such as *Deep Reinforcement Learning* [78, 79] but as a first prototype we will restrict ourselves to neural networks for the reconstruction algorithm.

The choice to use gradient descent, and therefore neural network, also allowed us to keep all technical software development wrapped in the same language and framework, PyTorch [43].

The backward-propagation introduce a second issue. At the beginning of the subsection we introduce $X' = \mathcal{G}(X)$, the event after perturbation. It's an input of the reconstruction $\mathcal{F}$, thus, let's say that the event, in its form X, is a list of tuples (id, Q, t) which are the hit on the PMT id. If $\mathcal{F}$ require the information to be formatted in a specific way (graph, images, ...) via an algorithm $\tau(X)$, it means that

$$\frac{\partial \mathcal{L}(\mathcal{F}(\tau(\mathcal{G}(X))))}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{G}}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \tau}{\partial \mathcal{G}} \cdot \frac{\partial \mathcal{F}}{\partial \tau} \cdot \frac{\partial \mathcal{L}}{\partial \mathcal{F}} \tag{6.9}$$

which also requires that $\frac{\partial \tau}{\partial \mathcal{G}}$ is differentiable.

On the other hand, if X is already formatted as the input of $\mathcal{F}$, it mean that $\mathcal{G}$ take the same format as input and we drop the requirement on τ to be differentiable. Concretely, if $\mathcal{F}$ takes an image as input, it mean that $\mathcal{G}$ will also takes an image as input and output an image. That also unfortunately mean that if some informations is loss before $\mathcal{G}$, for example during the charge and time aggregation in pixels, it cannot retrieve and modify it.

A more elegant solution would that $\mathcal{G}$ would also compute the transformation τ in addition to finding relevant perturbation, but for the simplicity of this exploratory work, we use a $\mathcal{G}$ that process transformed data.

6.2.2 Reconstruction Network (FFNN)

As introduced just before, we need a NN algorithm for IBD reconstruction. We could have used the GNN presented in Chapter 5 but we preferred a more simplistic approach to not be constrained by the memory consumption of the reconstruction neural network. This network is designated as FFNN.

This network takes as input a vector containing the results of the aggregation of charge and time on pixels. We consider JUNO composed of 3072 pixels defined by the Healpix [71] pixelisation. On each of those pixel, we sum the charges and keep the minimal time of hit, resulting in 3072 (Q, t) tuples. To those tuples, we adjoin the position of the center of those pixels, resulting in 3072 (Q, t, x, y, z) tuples. The data is finally represented as a $3072 \times 5 = 15360$ vector. In the case the charge in a pixel is 0, the time is set to 2048 ns, way after the closing of the trigger window.

The charge is expressed in N_{pe} and the time of hit in ns. The time is negative, meaning that 0 ns the first hit time and -2048 ns is the latest hit time.

The simplistic neural network is simply and Fully Connected Neural Network (FCDNN) composed of the following layer: the input layer, providing the 15360 items vector, followed by fully connected linear layers with respectively $[8192, 4096, 2048, 1024, 512, 256, 128, 64, 32]$ neurons. These layers possess a LeakyReLU activation function defined as

$$\text{LeakyReLU} = \begin{cases} x, \text{if } x > 0 \\ 10^{-2} \cdot x, \text{otherwise} \end{cases} \tag{6.10}$$

The last layer is a linear layer of 4 neurons, for (x, y, z, E) without an activation function.

The loss used is the Mean Square Error (MSE)

$$\text{MSE}(\boldsymbol{\eta}, \boldsymbol{\eta}^{true}) = \sum_i (\eta_i - \eta_i^{true})^2 \tag{6.11}$$

where η takes the values of (x, y, z, E).

The optimizer used for its training is the Stochastic Gradient Descent with momentum

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \Lambda \left(\sum_{i=0} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_{t-i}} \cdot 0.9^i \right) \tag{6.12}$$

where $\boldsymbol{\theta}_t$ is vector of learnable parameters at step t. Λ is the learning rate set at 10^{-3}. The difference with the classical SGD is the gradient term with $i > 1$. We save the gradient computed in the previous step and use them as momentum with a decaying weight. The factor 0.9 is an hyperparameter that has been selected for the training.

Additionally, to prevent over-fitting, we introduce a weight decay. Each step, we reduce the amplitude of the parameters $\boldsymbol{\theta}$ by 10^{-3}:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t * (1 - 10^{-3}) \tag{6.13}$$

Performances

The FFNN is trained independently from the ANN. The dataset is comprised of 1M positrons events uniformly distributed in the detector and in energy over $E_{dep} \in [1, 10]$ MeV. The training dataset account for 990'000 events with 10'000 events reserved for validation. The data are normalized, mean shifted to 0 and standard deviation scaled to 1, before being processed by the network.

Each epochs goes trough the entire training datasets, with a batch size of 64. The training last for 25 epochs.

Voir avec Gilles

6.2.3 Adversarial Neural Network (ANN)

The ANN aims to introduce perturbations in the event data in such a way that these perturbations are not detectable in the control dataset, while still degrading the energy reconstruction of the IBD dataset. For this, and for the reasons detailed in Section 6.2.1, the ANN operates on the inputs of the reconstruction network presented above, the FFNN. During the training, the parameters of the FFNN are *frozen*, meaning they will not be updated during the ANN training. If they were free to be optimized, they would adapt to the perturbation of the ANN, that would gao against the objective of this work.

The FFNN takes as input a vector of 5×3200 values, representing the (x, y, z, Q, t) of 3072 Healpix pixels. Those values comes from the aggregation of the PMTs belonging to those pixels.

It seems unreasonable that the ANN would modify the pixel positions, as they are derived from a mathematical construction. It could, however, perturb which PMTs are assigned to specific pixels, introducing localization errors, but the position of the PMTs is carefully monitored during JUNO's construction. Such aggregation errors would likely arise from PMTs located at the edges of the pixels, yet this scenario seems unlikely. Moreover, due to the constraints mentioned in Section 6.2.1, the ANN is required to work with the same format that the FFNN uses as input.

At the start of the project, we attempted to have it operate on both time and charge information simultaneously, but it struggled to converge. After discussions with colleagues in the collaboration, we decided that the ANN would only introduce perturbations in the charge information as most of the energy information comes from the charge.

Our ANN thus needs to output a 3200-dimensional vector, which represent the updated charges of the detector.

We decided on a Fully Connected Depp NN "bottleneck" architecture for the ANN, illustrated in Figure 6.2. This architecture put after the input a 4096 neurons wide layer, followed by smaller and smaller layers, 2040, 1024, 512, then finally 256 neurons. From this layer, the layer size grow again, 512, 1024, then 2048 neurons before the final output layer of 3072 neurons.

The idea behind this architecture is that, by reducing the number of neurons per layer, we force the network to summarize the event in 256 parameters, that it will use to regenerate an event. This architecture has also the advantage of keeping the number of learnable parameters relatively small, as the connection between small layers do not require a lot of parameters.

ANN loss

As it was mentioned in the introduction of Section 6.2, the loss of the ANN is composed of two losses, the adversarial loss $\mathcal{L}_{adv}$ and the regularisation loss $\mathcal{L}_{reg}$. To those two losses, we adjoin a penalty term that prevent the ANN from producing non-physical events.

$$\mathcal{L} = \mathcal{L}_{adv} + \mathcal{L}_{reg} + P$$

The adversarial loss $\mathcal{L}_{adv}$ is defined as the absolute value correlation between the reconstructed energy and the energy deposit (Eq. 6.2). The regularisation loss $\mathcal{L}_{reg}$ is the MSE of the true and reconstructed energy position vector (x, y, z, E) (Eq. 6.4).

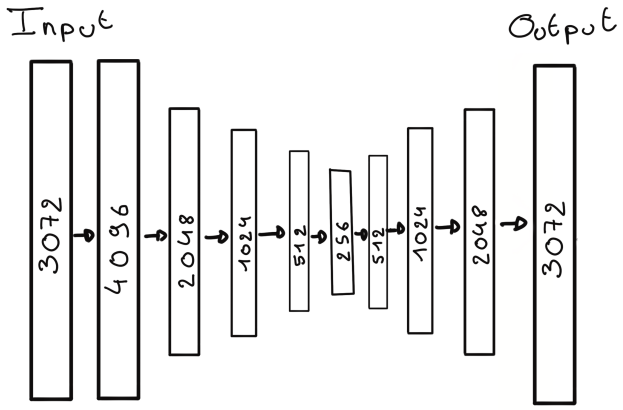


FIGURE 6.2 – Illustration of the "bottleneck" architecture of the ANN. Each block represent a fully connected layer with, on the left, the input layer and on the right the output layer. We see a first reduction of the number of neurons per layer, going from 4096 to 256, followed by an augmentation back to 4096 neurons, thus the "bottleneck"

The penalty term is here to prevent the network from generating event that are too far from the initial event. The penalty P is a function that takes the pixelated event X, its transformation after the ANN $\mathcal{G}(X)$ and a constraint ϵ

$$P(X, \mathcal{G}(X), \epsilon) = \sum_{i=1}^{3072} (ReLU(-\mathcal{G}(X)_i) + D_i) \tag{6.14}$$

with

$$D_i = \begin{cases} \frac{(X_i - \mathcal{G}(X)_i)^2}{X_i^2} & \text{if } \frac{|X_i - \mathcal{G}(X)_i|}{X_i} > \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{6.15}$$

where i index the Healpix pixels. The term $ReLU(-\mathcal{G}(X)_i$ is minimal, equal 0, when the charge after perturbation is positive. This term prevent the ANN from producing negative charge, feat impossible for the PMTs. The second term D_i is equal 0 when the relative charge between the original and perturbed pixel is less than ϵ. Otherwise, it is the square of this relative charge difference. This term penalize the ANN from producing charges too different from the original event.

When dealing with multiple losses like this, it is important tot compare them, as we do not want one term to absorb the other.

The loss $\mathcal{L}_{adv}$ range from 0 to 1 while $\mathcal{L}_{reg}$ is 0 when the vertex is perfectly reconstructed by it can theoretically go up to infinity. In practice we expect it to take value of the order of magnitude coherent with the reconstruction performances. In fact, if it would take higher value, it would mean that the reconstruction would reconstruct the event far away from the true vertex in comparison to the expected performance. This kind of issue would be immediately be detected, even with simplistic reconstructions such as the charge barycenter, which goes against the goal of producing subtle fluctuation.

We evaluate $\mathcal{L}_{reg}$ with (x, y, z) in meter and E in MeV. If the event is reconstructed with a precision of 15 cm and an energy resolution of 3% at 1 MeV, here taking the reconstruction performance of the best reconstruction algorithm OMILREC (see Sections 3.3 and 5.7), $\mathcal{L}_{reg} \approx 0.3^2 + 0.03^2 = 0.0909$. We see about an order of magnitude between $\mathcal{L}_{adv}$ and $\mathcal{L}_{reg}$. To compensate for it we weight $\mathcal{L}_{reg}$

$$\mathcal{L} = \mathcal{L}_{adv} + 60 \cdot \mathcal{L}_{reg} + P(\epsilon) \tag{6.16}$$

The amplitude of P and the value of ϵ will be further discussed in Section 6.3.

Hyperparameter optimization

All the ANN hyperparameters presented above have been optimized through the numerous iteration the architecture went through. The training is computationally expensive as we need to host both of the network on the GPU card, reaching quickly the memory limit of the GPU. The training of the ANN can takes up to 90h. The requirement of having a powerful GPU can be met locally, as Subatech possess an available A100 [66] card with 40GB of memory but we could not port over computing center as they only possess V100 [67] GPU with 20GB of memory.

Those constraint made a random search optimization impossible. It is maybe possible, through optimisation, to reduce the memory requirements to reach the threshold to run on V100 but the challenge was deemed not worth it for an exploratory work.

6.3 Training of the ANN

The training of the ANN goes through two phases. The first one consist on producing physical events, the second one into searching for perturbations. For both phases, we use the both of the datasets presented in section 6.1. We use a batch size of 64 for both datasets meaning that, for each steps, the network see 128 events.

Each epochs goes through the entirety of the training dataset.

6.3.1 First training phase: back to physics

When the ANN is created, before any training has been done, its parameters are initialized with random value. Multiple initialization methods exist. In this work, we go with a common He initialization [80], which is the default initialization in the PyTorch [43] library. If we were to ask for an event from the ANN without training first, the results would be random noise. We thus first have it to learn to produce physically sound events.

For this we go for a training of 200 epochs where the loss consist only of the penalty loss. For scaling purpose , the penalty P is scaled by 0.25.

$$\mathcal{L}_1 = 0.25 \cdot P(\epsilon = 0.01) \tag{6.17}$$

During this phase, the only objective of the network is to yield events that are the same as the original event.

The evolution of this loss $\mathcal{L}_1$ during the training for the training dataset and the validation dataset is presented in Figure 6.3. We see that the ANN converge to some stability in the loss.

The time and charge channels of two events after this training phase are presented in Figures 6.4 and 6.5. We remind that the ANN only act on the charge channel of the event.

We observe that for a localized event, Figure 6.4, the ANN reproduce correctly the event while for more a diffuse event, Figure 6.5, it produce more "uniform" charge distribution. By looking at the color scale in Figure 6.5, we observe that the ANN do not reproduce singular high number of N_{pe}. The highest pixel in the original was 12 N_{pe} whereas, after the ANN, the highest pixel is 5 N_{pe}. Furthermore, where in the original event the charge repartition, while diffuse, was still concentrated in specific pixel, the ANN spread the charges in every the pixels.

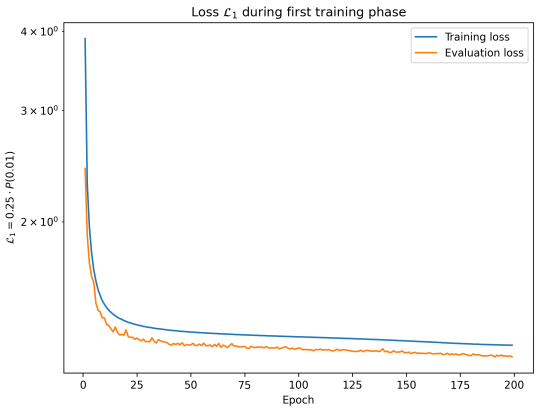


FIGURE 6.3 – Evolution of the loss $\mathcal{L}_1 = 0.25 \cdot P(0.01)$ during the first phase of the training

6.3.2 Second training phase: Breaking of the reconstruction

Once the ANN is able to reproduce physical events, we change the loss so it start to search for potential perturbations. For this we introduce the term $\mathcal{L}_{adv}$ and $\mathcal{L}_{red}$ producing a second loss $\mathcal{L}_2$. Adding those terms will significantly change the loss. The previous minima in the parameters phase space the ANN found minimizing $\mathcal{L}_1$ will not be the minima $\mathcal{L}_2$. To prevent a gradient explosion, we introduce a growing factor λ in front of the term $\mathcal{L}_{adv}$ and $\mathcal{L}_{red}$. This factor start at $\lambda = 0.01$ at epoch 201 and grow $\lambda_{i+1} = \lambda_i + 0.01$ where i index the epoch. It cap at $\lambda_{max} = 1$ at epoch 300 after what it stop growing.

Also to ease the task of the ANN, we relax the constraint in the penalty term P from $P(0.01)$ to $P(0.15)$.

The expression of the phase 2 loss $\mathcal{L}_2$ become:

$$\mathcal{L}_2 = \lambda \left(\mathcal{L}_{adv} + 60 \cdot \mathcal{L}_{reg} \right) + 0.25 \cdot P(0.15) \tag{6.18}$$

This second phase of the training last for 200 more epochs, up to epoch 400.

6.4 Results

6.5 Conclusion and prospect

— Not enough
— Probably guide the ANN

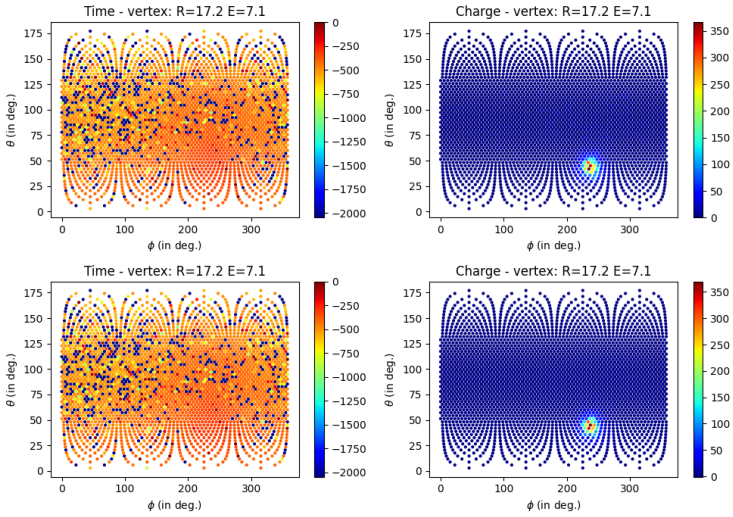


FIGURE 6.4 – Time channel (on the left) and charge channel (on the right) of a **radial, high energy event** (R = 17.2 m, E_{dep} = 7.1 MeV), **Top:** before the ANN perturbation, **Bottom:** after the ANN perturbation. The ANN have been trained for 200 epochs, just after Phase 1. Time channel in ns and charge channel in N_{pe}.

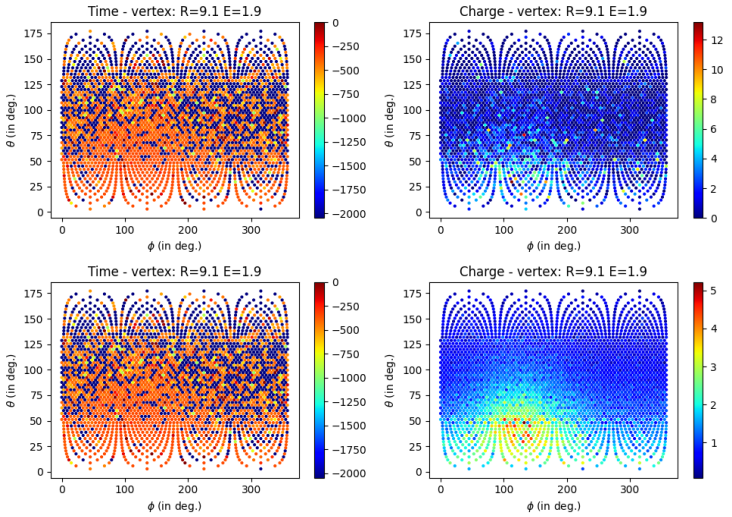


FIGURE 6.5 – Time channel (on the left) and charge channel (on the right) of a **central, low energy event** (R = 9.1 m, E_{dep} = 1.9 MeV), **Top:** before the ANN perturbation, **Bottom:** after the ANN perturbation. The ANN have been trained for 200 epochs, just after Phase 1. Time channel in ns and charge channel in N_{pe}.

Chapter 7

Dualcalorimetric analysis with neutrino oscillation for Precision Measurement

"We demand rigidly defined areas of doubt and uncertainty!"
Douglas Adams, The Hitchhiker's Guide to the Galaxy

Contents

JUNO is a high-precision neutrino oscillation experiment. To resolve the Neutrino Mass Ordering (NMO) with the required statistical significance, JUNO must be sensitive to the subtle spectral phase shift, on the order of a few percents, as illustrated in Figure 7.1. This phase shift manifests as a small

difference between the Normal Ordering (NO) and Inverted Ordering (IO) spectra, which becomes even smaller after accounting for detection effects such as energy resolution smearing, non-linear detector responses, and background contamination, as shown in Figure 7.2.

This chapter is based on simulated data due to the unavailability of real JUNO data, which will only be available in 2025. The purpose of this analysis is to validate the methods and tools developed for dual calorimetry and neutrino oscillation measurements, ensuring that they are robust and ready for future real data.

Among other condition, a precise and complete understanding of the reconstruction and detector effects is crucial. The challenge reside in the technology used in the detector, which, while based on well known technology: scintillator observed by PMT, is being deployed on a scale never seen before, in term of scintillator volume and PMT size. Understanding every effects that goes in the detector can become extremely complicated. Any method to help detecting problems is therefore welcome. Comparing the data and results obtained by two systems measuring the same events, but subject to different sources of error, is therefore precious. This is the purpose of the dual calorimetry techniques used in JUNO thanks to the existence of 2 PMT systems: the LPMT and SPMT systems.

The reconstruction of the IBD positron energy must be very performant: an unprecedented resolution of 3% at 1 MeV [35] is necessary to determine the NMO with the aimed significance.

Furthermore, an energy scale uncertainty below 1% is essential to accurately assess the likelihood of the NO and IO hypotheses. If this uncertainty exceeds 1%, systematic biases could distort the reconstructed spectra, potentially leading to the erroneous exclusion of the correct mass ordering hypothesis (NO or IO). For instance, a shift in the energy scale could mimic a phase shift between the spectra, making it possible to wrongly favor NO when IO is true, or vice versa. This effect has been studied in the introduction of Chapter 4 of [24].

Understanding all the effects influencing the detector response can be quite complex. Consequently, any methodologies that facilitate problem detection and validation of the reconstruction processes are essential for ensuring accurate results.

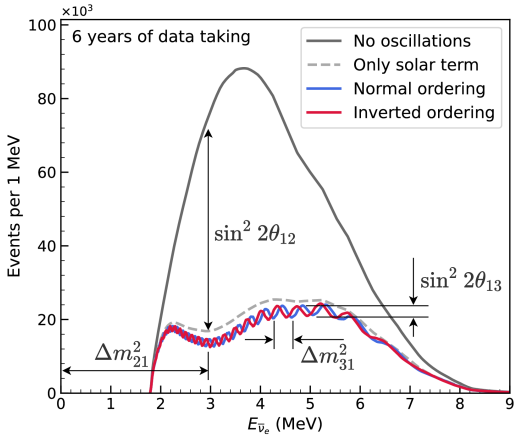


FIGURE 7.1 – Expected number of neutrinos event per MeV in JUNO after 6 years of data taking. The black curve shows the flux if there was no oscillation. The light gray curve shows the oscillation if only the solar terms are taken in account (θ_{12}, Δm^2_{21}). The blue and red curve shows the spectrum in the case of, respectively, NO and IO. The dependency of the oscillation to the different parameters are schematized by the double sided arrows. We can see the NMO sensitivity by looking at the fine phase shift between the red and the blue curve.

One detector effect to take into account is the detector non linearity. Detector non-linearity can introduce significant biases in the energy reconstruction of events, compromising the precision of

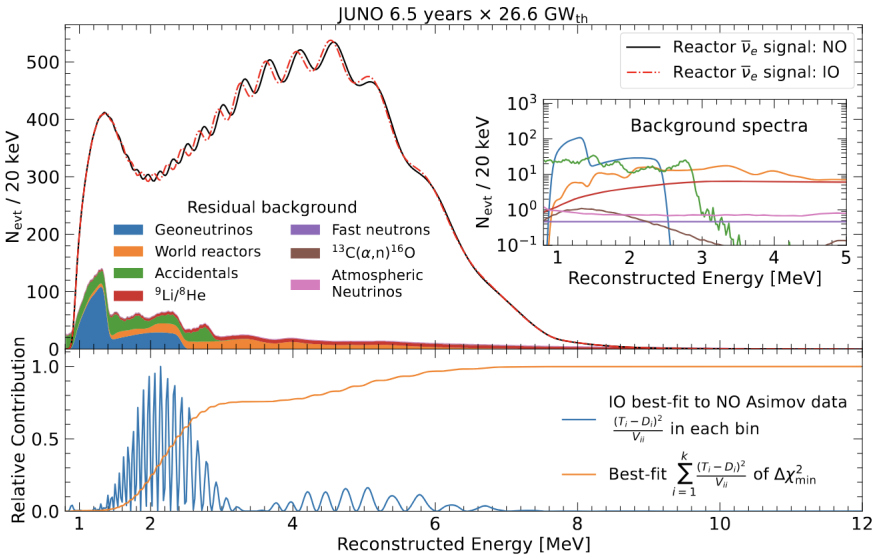


FIGURE 7.2 – Oscillated reactor $\bar{\nu}_e$ spectra for the Normal Ordering (Black) and Inverted Ordering (Red) for 6,5 years data taking and a resolution of 3% without any statistical or systematic fluctuation. Figure from [32].

neutrino oscillation measurements and increasing systematic uncertainties, which could potentially distort the determination of the neutrino mass.

One of the possible source of non-linearity, which will be used as a reference in this chapter, is the charge non-linearity (QNL) that will be discussed in next section. Several dual calorimetry techniques can address this issue. Some are calibration techniques, that are also described in section 4.3 of [24] . More generally, comparing the results of the two systems will allow for the detection of potential issues on the calibration or reconstruction. This is done in this thesis by comparing directly the spectra and oscillation parameters measurements of the two PMT systems. We call this kind of dual calorimetry "Dual calorimetry with neutrino oscillation", since it is based on the visible energy spectra used by the oscillation analysis of reactor antineutrinos.

In this chapter, we explore several ways to perform this comparison. One of them relies on the difference between the values of Δm^2_{21}, $\sin^2(2\theta_{12})$ measured with the LPMT and the SPMT systems. Both systems measure them with similar uncertainties. For reasonable values of the QNL, we expect these differences to be smaller than the individual uncertainties. However, the significance of these differences might still be high. Indeed, both systems reconstruct the same events, therefore the same distribution of the true positron energy, as well as the same scintillation photon emission. Therefore,the energy spectra reconstructed by the two systems share a part of their fluctuations. This translates into correlated reconstructed spectra and consequently lead to correlations between the measurements of Δm^2_{21} and $\sin^2(2\theta_{12})$. The uncertainty on the SPMT-LPMT difference is largely decreased by this correlation. Other ways to perform the comparison (see next sections) all rely the reconstructed spectra, therefore on the evaluation of the correlation between the LPMT and SPMT spectra.

In the next section we will discuss the motivations behind this study. In Section 7.2, I present the methods we propose to implement Dual calorimetry with neutrino oscillation, and of the way we estimate their sensitivity. In section 7.3, I present the fit framework used, and then, in section 7.4 the technical improvement brought and the difficulties faced during the development. To end this chapter I present the results in 7.7 and discuss the conclusions and perspectives in 7.8.

7.1 Motivations

7.1.1 Discrepancies between the SPMT and LPMT results

As mentioned earlier, the SPMT and LPMT systems are expected to detect the same events. Therefore, after proper calibration, any significant discrepancies between the two systems' results could indicate a calibration error, a systematic effect, or an unaccounted detector issue. Detecting such differences is critical, as even small deviations from the expected response could compromise the determination of the Neutrino Mass Ordering (MO) or introduce systematic biases in the oscillation parameter measurements, leading to incorrect conclusions about the true mass ordering.

Both systems are anticipated to show similar sensitivity to the oscillation parameters θ_{12} and Δm^2_{21} [3]. Therefore, any detected discrepancies will be based on these parameter measurements. A simple comparison of the values and independent uncertainties from the two systems could highlight discrepancies. However, we believe—and will demonstrate in this chapter—that an independent analysis of each system lacks critical information. By considering both statistical and systematic correlations between the two systems, we can design more robust and powerful statistical tests.

Our work in this chapter is to develop such tools, which in practice implies to define test statistics. A first step will be to determine the distribution of these test statistics in the case when no unexpected problem affects the LPMT nor the SPMT problem. This will give us the distribution of those statistical test in absence of discrepancies. Later, the value of the test statistics that we will measure in real data can be compared to these distributions to produce p-values, to judge of the potential present of an unexpected effect.

To evaluate the power of our methods, we need to simulate a concrete difference between the two spectra. We have chosen to study a specific potential effect, Charge Non-Linearity (QNL), which will be detailed in the following section. QNL affects the reconstructed energy spectrum by introducing a non-linear relationship between the true and measured charge in the PMTs. Our statistical tests are designed to detect such distortions, and they should be sensitive to unexpected effects—such as calibration errors or insufficient simulation precision—as long as the induced distortion exceeds a threshold of approximately 1-2% in the reconstructed energy spectrum.

7.1.2 Charge Non-Linearity (QNL)

The energy response of the Central Detector (CD) is influenced by two types of non-linearity. The first arises from the intrinsic properties of the Liquid Scintillator (LS), where the photon production is not linearly proportional to the deposited energy, as shown in Figure 2.12a. This non-linearity results from a combination of scintillation and Cherenkov light production. The scintillation yield is governed by Birk's law, which introduces a "quenching" effect that depends on the particle type and energy. Additionally, Cherenkov radiation, which constitutes less than 10% of the collected light, introduces a velocity-dependent non-linearity. These physical non-linearities in the LS contribute to the overall non-linearity of the energy response before any further distortions from the photomultiplier tubes (PMTs)

The second type of non-linearity comes from the LPMT charge measurements. When photons hit a PMT and give rise to PEs, a current pulse is formed. In the photon counting regime, simply exceeding a certain threshold allows to conclude that a single photon hit the PMT. When several photons hit the PMT simultaneously, one enters the photon integration regime : the pulse is sampled and integrated over a certain time window to produce a reconstructed charge Q. Calibration methods are applied to determine the relationship between the charge Q and the number of PEs (which is the quantity proportional to the energy deposit one wants to measure). Several effects impact this procedure: the signal pulse can fluctuate and be distorted between two events where the same number PEs occured; the PMT gain might not be linear as a function of the number of photons that hit the PMT;

the charge reconstruction algorithm is not supposed to be perfect, and its results are further affected by electronic noise and inter-channel cross-talk. The impact of these effects grows with the number of PEs.

Precedent studies, Section 4.2.3 of [24], suggest a model for the channel-wise QNL:

$$\frac{Q_{rec}}{Q_{true}} = \frac{-\gamma_{qnl}}{9} Q_{true} + \frac{\gamma_{qnl} + 9}{9} \tag{7.1}$$

where Q_{rec} is the reconstructed number of PE by the PMT, Q_{true} is true number of PE that hit the PMT, and γ_{qnl} is a factor representing the amplitude of the non-linearity.

Studies at previous experiments, like Daya Bay, concluded that the best reachable control of QNL in the 1-10 PEs range was $\gamma_{qnl} = 0.01$ [81]. As already mentionned in Section 2.3.2, JUNO LPMTs operate in a larger range : 1-100 PEs (See also table 7.1). In such a case, a realistic value of γ_{qnl} is not known.

	1PE	2∼5PE	5∼10PE	10∼20PE	20∼50PE	50∼100PE	>100PE
LPMT	42.56%	40.54%	8.74%	5.12%	2.80%	0.24%	0.003%
SPMT	95.19%	4.80%	0.01%	0%	0%	0%	0%

TABLE 7.1 – The charge fraction in terms of the number of PE collected at the single PMT for the reactor $\bar{\nu}_e$ IBD events. Table taken from [24]

The event-wise impact resulting from the channel-wise QNL can be parameterised this way :

$$\frac{E_{vis}^{rec}}{E_{vis}^{true}} = \frac{-\alpha_{qnl}}{9} E_{vis}^{true} + \frac{\alpha_{qnl} + 9}{9} \tag{7.2}$$

In JUNO, the visible energy is proportional to the number of emitted photons per unit energy deposit. It includes the physical non linearities. In the equation above E_{vis}^{true} is this visible energy, while E_{vis}^{rec} is what it becomes when the reconstructed charges found in an event are modified according to Eq. 7.1.

An example is shown on Fig. 2.14, where we show the $E_{vis}^{rec}/E_{true}^{vis}$ ratio for several samples of uniformly distributed electron events, generated with various values of E_{true}^{vis}. Here, an extreme value $\gamma_{qnl} = 0.05$ was assumed. On can see on Fig. 2.14 that it corresponds to a 2% effect at 8 MeV, equivalent to $\alpha_{qnl} = 0.025$. The effect of Eq 7.2 is illustrated in Figure 7.3.

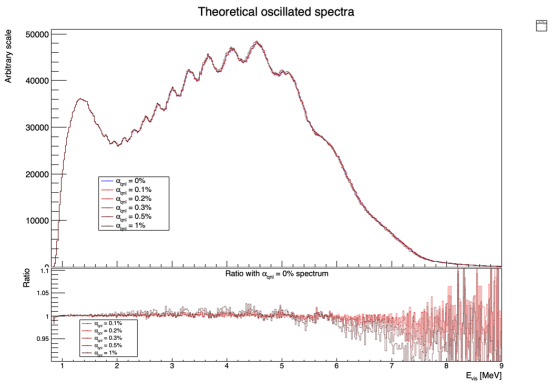


FIGURE 7.3 – **On top:** Oscillated spectra for different value of α_{qnl}. **On bottom:** Ratio of the number of event with $\alpha_{qnl} = 0\%$.

This example is from from references [24], which aimed at demonstrating the potential of the dual

calorimetry calibration method mentioned in section 2.4.3. If it works as hoped, the residual event-wise QNL effect will be below 0.3%. In this chapter, we propose methods todetect residuals higher than this.

Fig. 7.5 show several other examples with varying γ_{qnl} values, and the corresponding values of α_{qnl}. Using 1M events from the JUNO official simulation J23.0.1-rc8.dc1 (released on 7th January 2024), we simulated events up to the photon collection in LPMTs and introduced an additional channel-wise QNL by using the equation 7.1 to modify the number of collected photons.

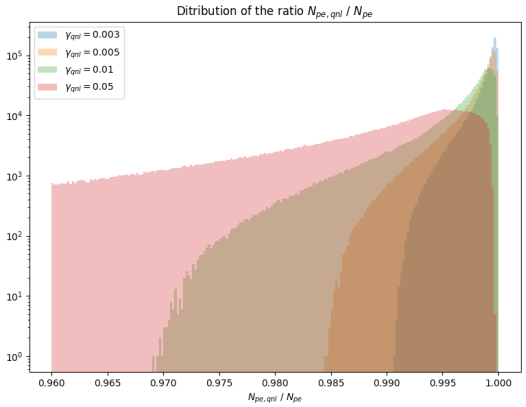


FIGURE 7.4 – Distribution the ratio reconstructed charge (in nPE equivalent) over the number of collected nPE for different value of γ_{qnl}. We use a sample of 1 million positron event uniformly distributed in the detector and in energy in the range $E_{dep} \in [1, 10]MeV$

In Figure 7.4 we show the distribution of the ratio of the reconstructed charge (in nPE equivalent) over the number of collected nPE for different values of γ_{qnl}. The right parts of those distribution, where the ratio is close to 1, are mostly central events. The charge is homogeneously distributed, the effect of the channel-wise QNL is reduced because the PMTs each collect a relatively small number of nPE. The left tail, with ratio < 1, are radial events, the photons are concentrated in a small number of PMTs, the effect of the channel wise QNL is greater.

In Figure 7.5, we show the mean of the distributions of Figure 7.4 as a function of the energy. From the 8.5 MeV data point, we compute an effective α_{qnl}. The effect of this effective α_{qnl} is represented as the dashed line. On the bottom of Fig 7.5 is presented the charge ratio difference between the effective α_{qnl} and the mean effect of a γ_{qnl}. We see that the event-wise QNL, described by Eq. 7.2, do not represent correctly the channel-wise QNL described by Eq. 7.1 at low energy. Indeed, Eq. 7.2 assume no QNL effect at 1 MeV, where in reality some of the PMTs will still suffer from QNL.

Despite this difference, the necessity to use the effective event-wise model expressed by Eq. 7.2, and consequently to find the correspondence between values of γ_{qnl} and α_{qnl}, instead of directly the channel wise model of Eq. 7.1 will be explained in Section 7.2.1.

7.2 Our approach to Dual Calorimetry with neutrino oscillation

In this section, we describe 4 statistical tests that we propose to use to detect unexpected effects in one of the PMT systems. Each test is based on a particular test statistics. In practice, the main result we want to produce in this chapter is the distributions followed by these test statistics.

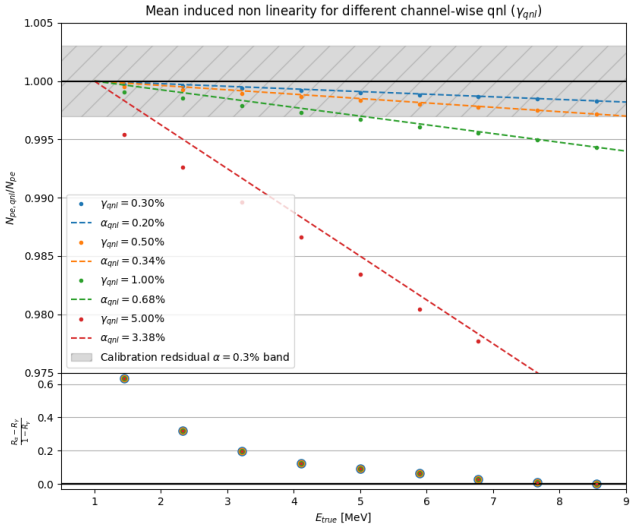


FIGURE 7.5 – **On top:** Ratio of the reconstructed charge (in nPE equivalent) over the number of collected nPE. The dots represent the mean of the distributions in Figure 7.4 and the dashed line are the equivalent event-wise non-linearity from eq 7.2. The hatched zone is the residual non-linearity expected after calibration [26]. **On bottom:** Difference between QNL induced by an event wise QNL and the mean QNL induced by a channel wise QNL. The value for α_{qnl} and γ_{qnl} follow the color code of the top figure. For a given energy, all the data point have the same value.

In this section, we propose four distinct statistical tests designed to detect unexpected discrepancies between the LPMT and SPMT systems. Each test aims to evaluate different aspects of the reconstructed energy spectra:

1. Test 1 compares the measurements of solar oscillation parameters $\sin^2 2\theta_{12}$ and Δm^2_{21} derived independently from each system.
2. Test 2 directly compares the LPMT and SPMT spectra bin by bin.
3. Test 3 involves a joint fit of the two spectra, with and without a hypothesis of discrepancy.
4. Test 4 examines the residuals in the fit of oscillation parameters $\sin^2 2\theta_{12}$ and Δm^2_{21} after the joint fit. The primary objective of this analysis is to establish the distributions of these test statistics under both the null hypothesis (no unexpected effect) and the alternative hypothesis (presence of a discrepancy).

The distributions of these test statistics cannot be analytically determined and are instead generated empirically through toy experiments. In each toy experiment, we generate two spectra of the IBD visible energy: one from the LPMT system and the other from the SPMT system. Since both systems observe the same events, their statistical fluctuations are correlated. To account for this, we compute a (820×820) covariance matrix that captures both the bin-to-bin correlations within each spectrum and the cross-correlations between the LPMT and SPMT spectra. Details of the sample generation process are provided in Section 7.3.3. Note that we use toy samples rather than samples produced by the full simulation of JUNO since the latter option would not be affordable in terms of computing time.

In the next subsection, we present the informations the reader must know about these spectra to understand the test statistics presented in the rest of the current section.

7.2.1 Toy experiments

The sensitivity of our tests depends on the sample size, which scales with the duration of exposure to the antineutrino flux: 100 days, 1 year, 2 years, and 6 years. For each exposure time, we generate 1000 toy experiments, where the number of events in the LPMT and SPMT spectra is drawn from a Poisson distribution with the expected mean value for that exposure. Since the same physical events are reconstructed by both systems, their fluctuations are not independent, and we account for the statistical correlations between the LPMT and SPMT spectra in our toy generation process. It was recently evaluated in the recent reference paper on JUNO's sensitivity [32] that that about 95000 IBDs would be selected in 6 years.

An example of pair of spectra is shown on Figure 2.16, in the form of two joint histogram of 410, 20 keV wide bins each. This is the format used in the fit performed by the present version of the reactor oscillation analysis developed at Subatech. It is important to notice that the IBD events present in the LPMT spectrum of a toy experiment are the same as those in the SPMT spectrum: the same events are just reconstructed twice, by either system. The LPMT and SPMT spectra are therefore not independent : Their respective fluctuations in the number of entries per bin are correlated. These correlations stem from what is common between the LPMT et SPMT spectra, namely :

— The statistical fluctuations of the true E_{vis} distribution (before any reconstruction).
— The fluctuation of the number of photons produced by scintillation or Cherenkov effect.

When generating toy experiment, the fluctuations drawn in each bin around the average expected number of events must account for these correlations. We therefore evaluated the (820×820) covariance matrix describing the uncertainty on the number of entries in each of the 410 bins of the 2 spectra, as well as the bin-to-bin correlations, especially those between the bins of the LPMT spectrum and those of the SPMT spectrum. This is described in Section 7.5. Here, we just want to emphasize the importance of this point, one of the original tasks to be carried out for the work presented in this chapter.

As already stated earlier, toy experiments will be used to evaluate the distributions of the four test statistics. We will first produce reference distributions: the ones that rule the possible values of the test statistics if none of the PMT systems is affected by any unexpected effect. These references are sufficient to run a test once JUNO will take data: the values of the test statistics obtained in a real data sample can be compared with the reference distributions, to evaluate to which extent the null hypothesis (no unexpected effect) is credible (p-values, or any pertinent quantities, can be computed). This is true whatever the nature of the unexpected effect.

To give an idea of the power of the method, an explicit scenario must be simulated for the unexpected effect. For that purpose, we also generate sets of toy experiments where the E_{vis} spectrum reconstructed by the LPMT is distorted using Eq. 7.2. We will test the following levels of QNL: $\alpha_{qnl} \in \{0.003, 0.002, 0.001\}$. As a reminder, the calibration guarantees a residual event-wise non-linearity of $\alpha_{qnl} \leq 0.003$ [26].

The most probable values in the distributions of the test statistics obtained in such cases will be compared with the reference distributions to derive a "median" predicted p-value. One can also compute the probability to observe in real data a p-value lower than a certain value, if the assumed QNL effect actually exists in these data.

When we initiated this work, the best test statistics to use was not obvious to us. This is why we decided to test 4 test statistics, of growing complexity. We present them in the 4 next subsections.

7.2.2 Comparing the solar parameters from individual analyses : LPMT vs SPMT

The first test statistics is probably the most natural one: it's essentially a direct comparison of the values of $\sin^2(2\theta_{12})$ and Δm^2_{21} measured by separate analyses of the LPMT and the SPMT spectra. These analyses are performed using the oscillation fit tool developed at Subatech, described in Sections 2.7 and 7.3. A fit to the LPMT spectrum provides $\sin^2(2\theta_{12})_L$ and $\Delta m^2_{21,L}$, while a separate fit to the SPMT spectrum provides $\sin^2(2\theta_{12})_S$ and $\Delta m^2_{21,S}$.

The direct comparison proceeds in practice via the differences between the fit results :

$$\Delta\theta = \sin^2(2\theta_{12})_L - \sin^2(2\theta_{12})_S \tag{7.3}$$

$$\Delta D = \Delta m^2_{21,L} - \Delta m^2_{21,S} \tag{7.4}$$

A very simple test statistics would be for instance

$$S = \frac{|\Delta\theta|}{\sigma_{\Delta\theta}} \tag{7.5}$$

directly related to the significance of the difference between the SPMT and LPMT results. This requires to determine the uncertainty $\sigma_{\Delta\theta}$. This cannot be considered as the mere quadratic sum of the uncertainties on $\sin^2(2\theta_{12})_L$ and $\sin^2(2\theta_{12})_S$ returned by the fitter. Indeed, because of the correlations, described in the previous subsection, between the LPMT and SPMT spectra, the fitted parameters are also correlated.

The calculation of $\sigma_{\Delta\theta}$ must account for it. Simple error propagation dictates :

$$\sigma^2_{\Delta\theta} = \sigma^2_{\sin^2(2\theta_{12})_L} + \sigma^2_{\sin^2(2\theta_{12})_S} - 2\sigma_{\sin^2(2\theta_{12})_L}\sigma_{\sin^2(2\theta_{12})_S}C_{L,S} \tag{7.6}$$

where $C_{L,S}$ is the correlation between the SPMT and LPMT measurements. We expect it to be high (well above 0.9, see Figures 7.6, 7.7, 7.8 and 7.9). Consequently, we expect it to considerably lower the value of $\sigma^2_{\Delta\theta}$, and increase the significance S.

This simple example can be seen as an illustration of the fact that the correlations between the LPMT and SPMT spectra boosts the sensitivity of our test statistics to unexpected effects. Indeed, with 6 years of data, and counting only the statistical uncertainties, we expect the statistical uncertainties $\sigma^2_{\sin^2(2\theta_{12})_L}$ and $\sigma^2_{\sin^2(2\theta_{12})_S}$ to both be around 0.15% [3]. A preliminary evaluation [60] of the impact of an uncorrected QNL effect with $\alpha_{qnl} = 1\%$ on the value of $\sin^2(\theta_{12})$ predicted a bias of 0.1%, therefore of 0.05% on $\sin^2(2\theta_{12})$. With no correlation, this would lead to a significance S far below 1. Accounting for the correlation allows far better.

The test statistics we actually use for this direct comparison is a generalisation of the simple one above : it includes both the results on $\sin^2(2\theta_{12})$ and Δm^2_{21} :

$$\chi^2_{ind} = \mathbf{\Delta}^T_{ind} U^{-1} \mathbf{\Delta}_{ind} \tag{7.7}$$

where $\mathbf{\Delta}_{ind}$ is a vector defined as

$$\mathbf{\Delta}_{ind} = [\Delta\theta,\ \Delta D] \tag{7.8}$$

using equations 7.3 and 7.4.

The covariance matrix U is a (2×2) matrix containing the uncertainties on the components of $\mathbf{\Delta}_{ind}$ and the correlation between them. We derive this matrix from the (4×4) covariance matrix V, which contains the uncertainties on the fitted values of $\sin^2(2\theta_{12})_L$, $\sin^2(2\theta_{12})_S$, $\Delta m^2_{21,L}$ and $\Delta m^2_{21,S}$, as well as the correlations between these quantities. For that purpose, we simply use the linear error

propagation formalism, that can be found in section 40.2.6 of the statistical review of the PDG 2020 [34] :

$$U = AVA^T \tag{7.9}$$

where the transfer matrix A is obtained this way

$$A_{ij} = \frac{\partial \Delta_i^{ind}}{\partial \lambda_j} \tag{7.10}$$

where λ_j one of the parameters $(\Delta m^2_{21,L}, \sin^2(2\theta_{12})_L, \Delta m^2_{21,S}, \sin^2(2\theta_{12})_S)$. Assuming this indexing order for j and i ordering following Eq 7.8, A is expressed

$$A = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \tag{7.11}$$

We acknowledge that linear error propagation is valid when all fluctuations or uncertainties are gaussian. However, since our results will be based on distributions of χ^2_{ind} produced with toy samples, this choice remains valid.

An important ingredient here is to determine the correlation coefficients in V. On a dedicated set of 1000 toy experiments, we perform fits to the LPMT and SPMT spectra, and compute the correlations empirically from the 1000 sets of best fit values of the solar parameters : $\sin^2(2\theta_{12})_L$ vs. $\sin^2(2\theta_{12})_S$, $\Delta m^2_{21,L}$ vs $\Delta m^2_{21,S}$, $\sin^2(2\theta_{12})_L$ vs. $\Delta m^2_{21,S}$, etc. We need the correlations corresponding to the null hypothesis and therefore use toy experiments produced with no QNL effect.

The correlations between these parameters for 100 days, 1 year, 2 years and 6 years can be found in Figures 7.6, 7.7, 7.8 and 7.9 respectively.

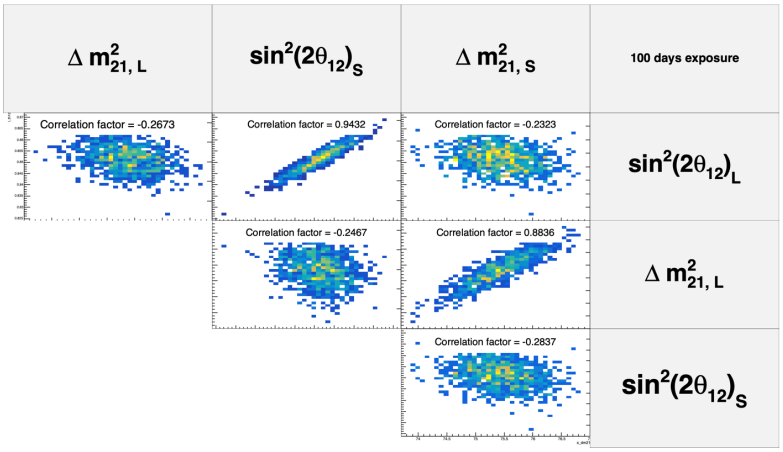


FIGURE 7.6 – Distribution and correlation between the best fit point of 1000 individual toys fit for 100 days exposure without supplementary QNL.

We observe strong correlation between the reconstructed Δm^2_{21} and $\sin^2(2\theta_{12})$ of both systems as presented in Table 7.2, row one and two. As the relative statistical uncertainty decrease with exposure, the correlations grow ranging from 0.88 to 0.95 for Δm^2_{21} and from 0.94 to 0.98 for $\sin^2(2\theta_{12})$. We observe between parameters of the same fit, a small anti-correlation of about -0.25, line 4 and 5 of Table 7.2.

Because the parameters are heavily correlated between the LPMT and SPMT fit, and that Δm^2_{21} and $\sin^2(2\theta_{12})$ are slightly anti-correlated in the same fit, the couples of different parameters from different fit, Corr($\sin^2(2\theta_{12})_L$, $\Delta m^2_{21,S}$) and Corr($\sin^2(2\theta_{12})_S$, $\Delta m^2_{21,L}$), are also anti-correlated.

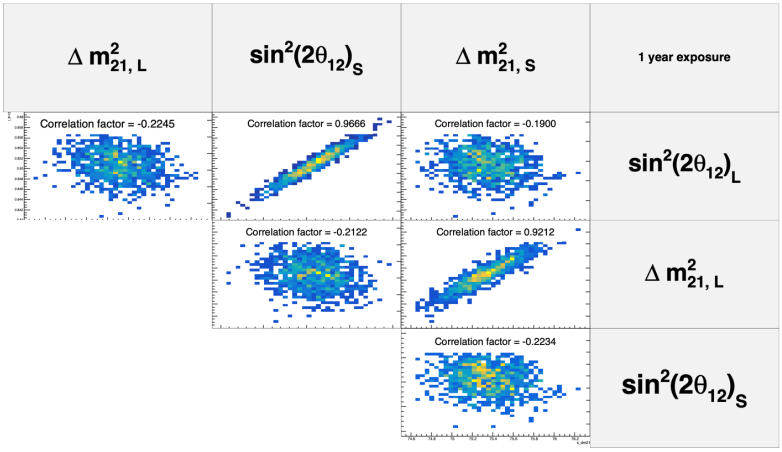


FIGURE 7.7 – Distribution and correlation between the best fit point of 1000 individual toys fit for 1 year exposure without supplementary QNL.

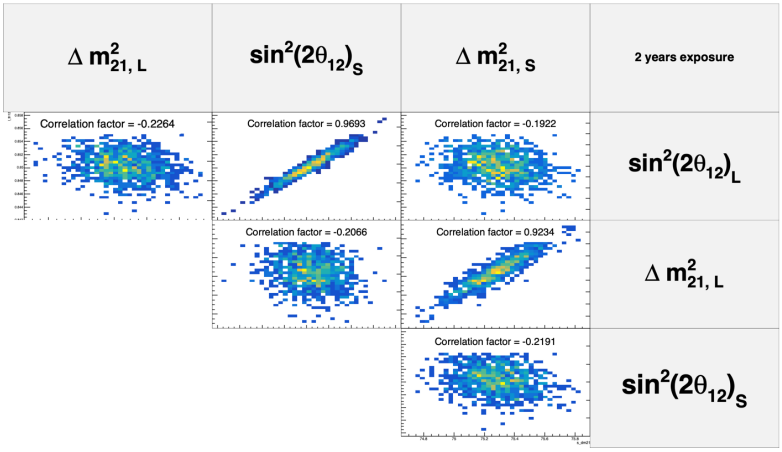


FIGURE 7.8 – Distribution and correlation between the best fit point of 1000 individual toys fit for 2 years exposure without supplementary QNL.

The distributions χ^2_{ind} will be Shown in Section 7.7.

7.2.3 Direct comparison between the SPMT and LPMT spectra

In the second test, we perform a bin-by-bin comparison of the LPMT and SPMT spectra without fitting any oscillation parameters. Again, we use here a χ^2-like statistics. We do not expect the reference distribution (for $\alpha_{qnl} = 0$) to be centered around the number of degree of freedom (i.e. the number of bins of each spectrum in our case) as should be distributed (if the spectra contain enough events in each bin to assume a gaussian behavior of the number of entries) the χ^2 comparing 2 histograms when they are consistent with each other. Indeed, even in the absence of unexpected events, the LPMT and SPMT are quite different because of the very different reconstruction resolutions. We therefore need here again to establish this reference distributions with toys. And compare them later with the distributions obtained for the various tested values of α_{qnl}.

Our test statistics is :

$$\chi^2_{spe} = \mathbf{\Delta}^T_{spe} U^{-1} \mathbf{\Delta}_{spe} \tag{7.12}$$

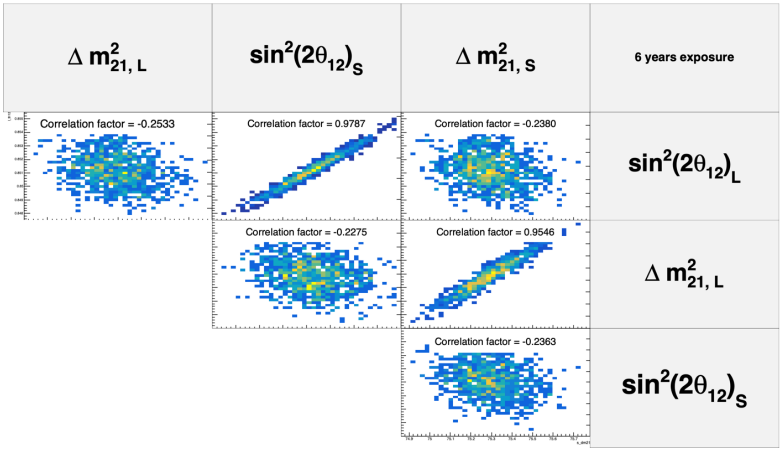


FIGURE 7.9 – Distribution and correlation between the best fit point of 1000 individual toys fit for 6 years exposure without supplementary QNL.

	100 days	1 year	2 years	6 years
$\mathrm{Corr}(\Delta m^2_{21,L}, \Delta m^2_{21,S})$	0.8836	0.9212	0.9234	0.9546
$\mathrm{Corr}(\sin^2(2\theta_{12})_L, \sin^2(2\theta_{12})_S)$	0.9432	0.9666	0.9693	0.9787
$\mathrm{Corr}(\sin^2(2\theta_{12})_L, \Delta m^2_{21,L})$	-0.2673	-0.2245	-0.2264	-0.2533
$\mathrm{Corr}(\sin^2(2\theta_{12})_S, \Delta m^2_{21,S})$	-0.2837	-0.2234	-0.2191	-0.2363
$\mathrm{Corr}(\sin^2(2\theta_{12})_L, \Delta m^2_{21,S})$	-0.2323	-0.19	-0.1922	-0.2380
$\mathrm{Corr}(\sin^2(2\theta_{12})_S, \Delta m^2_{21,L})$	-0.2467	-0.2122	-0.2066	-0.2275

TABLE 7.2 – Correlations between the parameters BFP of the individual LPMT and SPMT fits for multiple exposures using 1000 toys.

where

$$\Delta_i^{spe} = h_{L,i} - h_{S,i} \tag{7.13}$$

and

$$U = AVA^T \tag{7.14}$$

Here, i runs over the 410 bins of the individual spectra. Also, $h_{L,i}$ and $h_{S,i}$ are the contents of the ith bin of the LPMT and SPMT spectra respectively. We need to know the uncertainty on Δ_i^{spe} and the correlations with Δ_j^{spe} 's in other bins. We derive them from V, the (820×820) covariance matrix introduced at the beginning of this section, which can be seen as the covariance matrix of a 820-bin double spectrum juxtaposing the LPMT and SPMT spectra. We remind its determination will be presented in Section 7.5. To obtain U from V, we again apply the linear error propagation, with the transfer matrix :

$$A_{ij} = \frac{\partial \Delta_i^{spe}}{\partial h_j} = \frac{\partial (h_{L,i} - h_{S,i})}{\partial h_j} \tag{7.15}$$

Thus, $A_{ij} = 1$ if $i = j$, and $A_{ij} = -1$ if j is the SPMT bin corresponding to the i LPMT bin.

We expect this statistics to have a certain power since χ^2_{spe} can be increased for 2 reasons in case of unexpected problem: first, the LPMT spectrum (if the LPMT is affected) will be distorted and become less consistent with the SPMT spectrum; second, the correlations between the LPMT and SPMT might also modified. Since V present a peculiar correlation pattern (see Section 7.5), a departure from this pattern also has some valuable impact on χ^2_{spe}.

7.2.4 Joint fit of the SPMT and LPMT spectra : $\chi^2_{H_0} - \chi^2_{H_1}$

This kind of fit has already been introduced in Section 2.7. As a reminder, it involves the minimisation of

$$\chi^2_{joint} = (\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{h}) - \boldsymbol{D})^T V^{-1} (\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{h}) - \boldsymbol{D}) + \ln(|V|) \tag{7.16}$$

where $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{h})$ is the predicted joint LPMT+SPMT spectrum and $\boldsymbol{D}$ the corresponding data vector. The matrix V is the full (820×820) covariance matrix which incorporate both the statistical uncertainties and the bin-to-bin correlations between the LPMT and SPMT spectra.

In this fit, we include the usual oscillation parameters, $\sin^2(2\theta_{12})$, Δm^2_{21}, $\sin^2(2\theta_{13})$ and Δm^2_{31} along with two additional parameters, $\delta \sin^2(2\theta_{12})$ and $\delta\Delta m^2_{21}$ which allow for a potential discrepancy in the LPMT reconstruction or calibration.

Several remarks must be made here to better understand what we do precisely.

— Given JUNO's lack of sensitivity to $\sin^2(2\theta_{13})$, this parameter is fixed in the fit to the PDG value (see table 7.3). In most of JUNO's fit procedures (see Section 2.7), it's allowed to float during the minimisation, but is treated like a nuisance parameter, by adding a penalty term based on the PDG central value and uncertainty.

— The oscillation fit that we perform here does not really aim at the oscillation parameters in themselves, but is performed to detect a difference between the LPMT and SPMT spectra. JUNO is supposed to be very sensitive to Δm^2_{31} via the LPMT spectrum. However, it has been shown by studies carried out at Subatech (and confirmed since then by other groups in the Collaboration), that up to 2 years of data taking, the presence of multiple minima in Δm^2_{31} χ^2 profile can make its determination delicate. Since Δm^2_{31} is not the aim of our present study, we stabilize the fit by treating this parameter as a nuisance parameter, adding to χ^2_{joint} the following penalty term :

$$\chi^2_{\Delta m^2_{31}} = \frac{(\Delta m^2_{31} - \overline{\Delta m^2_{31}})^2}{\sigma^2_{\overline{\Delta m^2_{31}}}} \tag{7.17}$$

We define two hypothesis. The hypothesis H_0 assumes that no unexpected effect is present, meaning that $\delta \sin^2(2\theta_{12}) = 0$ and $\delta\Delta m^2_{21} = 0$, and the hypothesis H_1 where $\delta \sin^2(2\theta_{12}) \neq 0$ and $\delta\Delta m^2_{21} \neq 0$ are needed to account for any potential calibration or reconstruction bias. The test statistic is then defined as the difference between the minimized χ^2 values under H_0 and H_1:

$$\Delta\chi^2 = \chi^2_{joint,H0} - \chi^2_{joint,H1} \tag{7.18}$$

where $\chi^2_{joint,H0}$ is the result of the minimisation when the fit assumed no unexpected effect (fixing $\delta \sin^2(2\theta_{12})$ and $\delta\Delta m^2_{21}$ to 0), while $\chi^2_{joint,H1}$ assumes a possible effect, letting this parameters free to float. A large value of $\Delta\chi^2$ would indicate a significant deviation from the null hypothesis (no discrepancy), suggesting the presence of an unexpected effect in the LPMT system.

Distributions of $\chi^2_{H_0} - \chi^2_{H_1}$ in the reference case and for various values of α_{qnl} will be produced and studied in Section 7.7.

The idea behind this joint fit is that by letting the oscillation parameters and $\delta \sin^2(2\theta_{12})$ and $\delta\Delta m^2_{21}$ free to float, converging potentially to arbitrary, wrong values in the case of oscillation parameters, we add some flexibility to fully exploit the difference introduced by unexpected effects between the reference spectra and correlations.

There were other reasons to develop this joint fit. The main one was that it required an update of our software framework so it's able to perform joint fit. It was not fully ready for that. This feature will be very useful when the Subatech team will include the TAO spectrum (via a joint fit) in the oscillation studies it will perform.

$\sin^2(2\theta_{12})$	Δm^2_{21}	Δm^2_{31}	$\sin^2(2\theta_{13})$
$0.851^{+0.020}_{-0.018}$	$7.53 \pm 0.18 \times 10^{-5}$ eV^2	$2.5283 \pm 0.034 \times 10^{-3}$ eV^2	$0.0.8523 \pm 0.00268$

TABLE 7.3 – Nominal PDG2020 value [34]. All value are reported assuming Normal Ordering.

7.2.5 Joint fit of the SPMT and LPMT spectra : distribution of $\delta \sin^2(2\theta_{12})$ and $\delta\Delta m^2_{21}$

The last test statistics we will study might be complementary to $\Delta\chi^2 = \chi^2_{joint,H0} - \chi^2_{joint,H1}$.

These test statistics are simply the fitted values of $\delta \sin^2(2\theta_{12})$ and $\delta\Delta m^2_{21}$. In the reference case, when no unexpected reconstruction problem is present, we expect them to be distributed in an approximate gaussian way, centered on 0. When QNL effect will be included, they will tend to converge to higher values, to compensate the bias introduced on the fitted $\sin^2(2\theta_{12})$ and Δm^2_{21} due to the distortion of the LPMT spectra and of the correlations between the LPMT and SPMT spectra.

Again, these distributions will be studied in Section 7.7.

7.2.6 Limitations

QNL in backgrounds

The JUNO commons inputs provides background spectra that already have been smeared by the LPMT resolution. Because the resolution depends on E_{vis} (Eq. 7.19), to apply supplementary QNL we would need to de-convolute the LPMT resolution, apply the supplementary QNL then re-smear the spectra. This deconvolution is no trivial. Thus we ignore the background when produced distorted spectra.

This should not affect too much the power of our statistical tools, as the backgrounds are common to both spectra and should not have any effect on the statistical covariance matrix.

Systematics

It would be more rigorous to also include systematic uncertainties. However, in the present state of our fit framework, it would require the computation (often empirical, via the generation of thousands of toy samples) of (820×820) covariance matrices, which was judge too time consuming with respect to the time we could devote to this chapter.

Moreover, it seems reasonable to think that the sensibilities evaluated with only statistical uncertainties would not be changed much by a full treatment. Indeed, all the systematic uncertainties affect the true visible energy spectrum, before reconstruction. This spectrum is a common input to both the LPMT and SPMT reconstructions. Therefore, observed differences between the oscillation parameters measured by one or the other system should not be due to these systematics effects, and remain of the same order as if these effects were absent.

Correlation between LPMT and SPMT reconstruction

Most of our results assume uncorrelated reconstruction uncertainties between the SPMT and LPMT systems. In practice, once the E^{vis} of a toy event is generated (see Section 7.3.3), we simulate the SPMT and LPMT reconstruction by adding a δE^{rec}_{SPMT} and a δE^{rec}_{LPMT}.

The two latter increments are chosen randomly on Gaussian distribution. These two drawings are carried out independently. In reality, the reconstruction of E^{vis} is about proportional to the number of PE, therefore to the number of scintillation photons produced in the scintillator. Both the LPMT and SPMT reconstruction depend on the stochastic variation of this number event to event. Their results therefore vary in a correlated way. The correlation is kept low since it is shuffled by another source of variability, namely the sampling of photons : the SPMT indeed reconstruct only a few dozen PEs when more than 10000 photons are emitted.

This correlation is higher when the interaction takes place close to the sphere's surface (ie close to some of the PMTS), the non-uniformity effect is correlated between the two systems. To account for it, when should ideal produce the simulated samples necessary to our studies by using the full simulation. However, it would be far too CPU intensive. The impact of neglecting this correlation will be discussed in Section 7.5.

Realistic QNL

The way we implement the QNL effect in toy samples is also simplified. The size of the QNL effect in a PMT depends on the number of photons hitting it, therefore on the position of the interaction. When generating toy events, we apply QNL event-wise, only as a function of the value of E^{vis} (Eq. 7.2). As explained in Section 7.1.2, the full simulation has been used to find the average α_{qnl} for a given γ_{qnl} which is considered sufficient for this exploration.

Again, replacing toy samples with samples generated with the full simulation would yield more accurate results, but is prohibitive in terms of calculation time. For future studies, sophisticated solutions to this problem will have to be found, but are out of the scope of this thesis.

7.3 Fit software

In this section, I describe the fit framework that was used in this study. The AveNu$_e$ framework is the adaptation to JUNO of one of the frameworks, partly developed at Subatech, used by the Double Chooz [82] experiment. It is composed of two parts: the AveNu$_e$ Generators and the AveNu$_e$ Fitting Package. The Generators are a set of standalone macros, the Fitting Package is an C++ package, using the RooFit library.

Both parts of the package are interfaced with what we call the JUNO inputs. These inputs comprise all the ingredients to build a $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$ prediction, among which :

- Reactor antineutrino spectra for each isotope as predicted by Mueller [83].
- The isotopes mean releases energy.
- Reactors´s thermal powers and fission fractions.
- Various corrections to account for the contributions from the Non Equilibrium Regime and the Spent nuclear fuel.
- A correction obtained by comparing these spectrum prediction in the case of the Daya Bay experiment with actual Daya Baya data [11].
- The IBD differential cross section as function of the antineutrino energy.
- The assumed values of the oscillation and nuisance parameters at the start of the fit or for sensitivity studies.
- Parameters describing the non linearity of the photon emission as a function of the deposited energy.
- Energy reconstruction parameters (see equation 7.19 and Figure 7.10).
- The selected IBD and background expected yields per day, and the background spectra, all obtained from JUNO's full simulation and studies to design the selection.
- Uncertainties on all these quantities for the computation of covariance matrices.

We describe in the next section the role of each part of the framework.

7.3.1 AveNu$_e$ Standalone Generators

The main macro here is the "IBD generator" macro. It is used to :

- Compute $\boldsymbol{T}_{no\ osc}(\boldsymbol{\eta})$ (unoscillated theoretical spectra) predictions. It is done by toy generating a spectrum. In order to not be affected by statistical fluctuations, it generates 100 times more statistics that JUNO's expected yield after 6 years. It is provided in the form of a TTree. These predictions concern an non oscillated spectrum.
- Toy samples simulated data sets. It is essentially used to simulate data spectra altered by QNL effects (see below).
- The above productions are input to the Fitting Package, or to other macros from Standalone Generators, which compute the covariance matrices necessary to the Fitting Package. Some of the covariance matrices are computed from the $\boldsymbol{T}$'s, using linear error propagation, some other are empirical calculations based on sets of toy samples generated with varying parameters. This is also the case for one of the versions of the computation of the V_{stat} covariance matrix of the LPMT+SPMT double spectrum (see Section 7.5).

7.3.2 AveNu$_e$ Fitting Package

Its role is to perform fits to a single data samples, of to a set of toy samples. In practice :

- It loads TTrees containing the data to fit as well as the $\boldsymbol{T}_{no\ osc}(\boldsymbol{\eta})$ predictions, and create local objects representing the data spectrum and the pdf. For that purpose, $\boldsymbol{T}_{no\ osc}(\boldsymbol{\eta})$ are changed into predictions $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$ for the oscillated spectrum by weighting events in the TTree according to the oscillation probability.
- It loads the necessary covariance matrices.
- It creates from this a χ^2 object. The Pearson, Neyman, CNP and Pearson V versions are available
- It is interfaced with Minuit via RooFit classes to perform the minimisation. At each step, $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$ are re-weighted by the oscillation probability corresponding to the current value of the floating oscillation parameters.

Three kinds of data can be fitted with this Package : real data, Asimov simulated data and toy data.

When real data will be available at JUNO, we expect that the result of the IBD selection will be made available by the collaborations via TTrees.

The principles of Asimov fits were described in Section 2.7.3. In practice, our Fit Package fill the local object representing the data spectrum with $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$, assuming some values for the oscillation parameters.

The toy data samples can have two origins. Some are produced by the IBD generator macro of the AveNu$_e$ Generators. This is the case of the toy samples that we produce with QNL effects. It is also possible to generate toys directly with the Fitting Package. In that case, toy data spectra are produced by generating random fluctuations around each the values of $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$.These fluctuations must be the reflect of both statistical and systematic uncertainties. Fluctuations between bins i and j can be correlated. Such correlations are common in the case of systematic uncertainties. In general, they are 0 for the statistical uncertainties. In our case, as already explained earlier (see for instance

the Sections where the test statistics are described), bins from the SPMT part of the LPMT+SPMT spectrum are correlated to bins of the LPMT part even for the statistical part.

To generate correlated fluctuation we use, through Choleski decomposition, the covariance matrices. This way to generate toy is faster. We use it in this work in the reference case (no QNL). In the case where QNL effects are simulated, the corresponding statistical covariance matrix is not known, we therefore resort to the IBD generator.

7.3.3 Details of the IBD generator

The IBD generator is a standalone generator used to produce oscillated and non oscillated spectra as the one seen by the JUNO experiment. It is at the core of the fitting framework as it's used to generate $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$, the toy data and spectra to compute the covariances matrix.

With thus have a flexible macro with options allow to enable or disable effects such as non-uniformity and non-linearity. It take as an argument the number of events to generate N_{evt}. Optionally, we generate an effective number of events N by drawing in a Poisson distribution of mean N_{evt}.

Then for each event we:

1. Choose randomly, following the reactor power fraction, the source reactor of the neutrino.
2. Generate a random interaction position in the detector following a uniform distribution over the detector volume.
3. Draw a random neutrino energy E_ν from the expected neutrino emission spectrum of every reactor. This spectrum is computed by:
   (a) Computing the power spectrum of each isotopes ^{235}U, ^{238}U, ^{239}Pu, ^{241}Pu using the Huber-Mueller model [5, 8].
   (b) Summing the contribution of each isotopes following the respective fission fraction [0.58, 0.07, 0.30, 0.05] as reported in [84].
   (c) The power of each reactor is then adjusted by their distances from the detector, the detector efficiency and their mean duty cycle (11 of 12 month).
   (d) The total spectrum is then finally adjusted by taking into account the correction of the Day Bay bump [11], adjustment due to spent nuclear fuel and due to the non-equilibrium.
4. *(Optional)* Compute the survival probability due to oscillation at nominal oscillation parameters value. If the neutrino does not survive, the event is rejected and the algorithm restart from step (1).
5. Compute the emitted positron energy E_{pos} from the mass difference. If the neutrino does not have enough energy reject the event and start from step (1).
6. Compute the deposited energy E_{dep} by incrementing E_{pos} by 511 keV to account for the positron annihilation. We do not consider cases where some of the energy leak outside of the detector (positron or annihilation gammas escaping the CD).
7. Correct the deposited energy with the expected event-wise non-linearity from [26] to obtain the visible energy E_{vis}.
8. *(Optional)* Add a custom non-linearity as described in Section 7.1.2. This non linearity is characterized by α_{qnl} to obtain E_α.
9. Finally, using the expected resolution of the LPMT and SPMT systems, provided in the JUNO common inputs, we draw from a gaussian characterized by those resolution the reconstructed energy E_{rec} or E_{lpmt} and E_{spmt} for each systems. The resolutions are provided as ABC parameters using

$$\frac{\sigma E_{vis}}{E_{vis}} = \sqrt{\left(\frac{A}{\sqrt{E_{vis}}}\right)^2 + B^2 + \left(\frac{C}{E_{vis}}\right)^2} \tag{7.19}$$

where A is the term driven by the Poisson statistics of the total number of detected photoelectrons, C is dominated by the PMT dark noise, and B is dominated by the detector's spatial non-uniformity. The relative and absolute resolutions of the LPMT and SPMT systems are illustrated in Figure 7.10.

The events are stored as n-tuples and are not yet binned at the end of the generator.

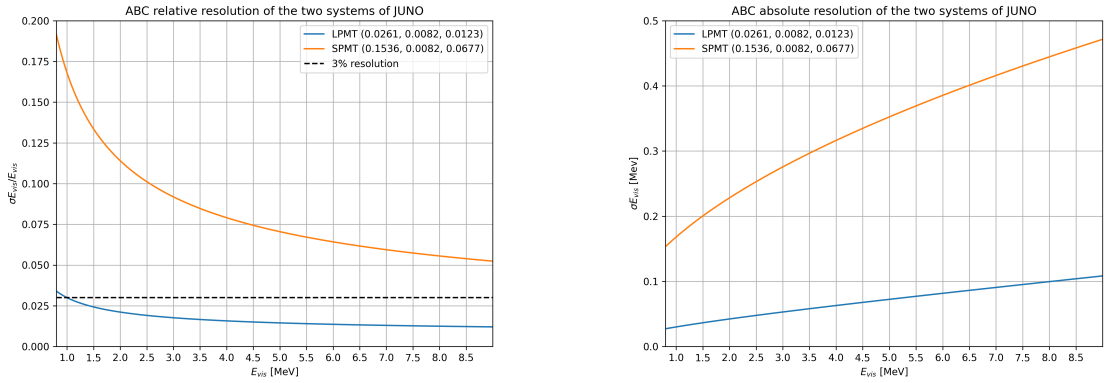


FIGURE 7.10 – Relative (**On the left**) and absolute (**On the right**) resolutions of the LPMT and SPMT systems used in this study. The number in parenthesis are the parameter A, B and C respectively for each systems.

7.4 Technical challenges and development

The fit framework Avenue was already partially developed with multispectra fitting in mind but a lot technical development was necessary to allow for a joint fit. This required a lot of work and constitute a good part of my total effort on this study. I remind that these development will be useful beyond this thesis and this subject. As already mentioned earlier, at some point, we should perform simultaneous fits of the JUNO and TAO spectra. It's also a potential starting point for combined analyses with other experiments, like long baseline experiments.

The first step was to migrate the framework from ROOT5 (last release in March 2018) to ROOT6 (v6.26.06 released in July 2022) to ensure compatibility with the data coming from the JUNO collaboration, and benefiting of the improvement and corrections that came with ROOT6. This allow us to upgrade the C++ standard from C++11 to C++17. A substantial effort has been done to modernize the code, generalizing the functions and methods via templating to help readability and using smart pointer to prevent possible memory leaks.

The Avenue framework had to be adapted, notably on the chi-square calculation and spectrum generation to correctly take into account the correlation between the SPMT and LPMT spectra. The delta joint fit requiring two more parameters over a spectrum twice as large as before with LPMT takes much more time, around 15h for 6 years exposure, than the single LPMT fit. Thus the framework and the fit macro had to be updated for distributed computing. Notably the aggregation of fit results can now be done in a single file instead of managing a file per fit. In case of numerous toy, the hard drive access time could lead to long analysis time.

While the IBD generator was already able to generate LPMT and SPMT spectrum, it was not designed for generating correlated spectrum. As detailed in Section 7.3.3, up to the reconstruction effect, the two spectrum need to share the same generation else the two spectrum would be decorrelated and it would be like we would run two different experiment.

7.5 Covariance matrix

The covariance matrix between the LPMT and SPMT spectra is at the heart of this study as it was already mentioned in Section 7.2. In this section we discuss the different approaches taken to estimate it. We remind that in this work, we consider only statistical effects and let to future works the task to include systematic uncertainties. We thus evaluate in this section the (820×820) statistical covariance matrix V of the LMPT+SPMT spectrum.

As already explained in previous Sections 7.2.6 and 7.3.3, we assume, in most of what follows, that the effect of the energy reconstruction resolution is independent between the LPMT system and the SPMT system, although this is an approximation. We therefore also briefly study the correlations between the two reconstructions.

7.5.1 Analytical method

The first method discussed is the analytical method where we propagate the resolution of the LPMT and SPMT spectra over a non-smeared spectrum. Following the approach used in the IBD generation in Section 7.3.3, we consider the system resolution $\sigma(E)$ to be only dependent in energy. We do not consider the position of the event.

Using the formalism of section *39.2.5 Propagation of errors* of PDG2020 [34] and considering an extended spectrum of 820 bins following the binning scheme introduced in 2.7.2, the first 410 for the LPMT and the last 410 for the SPMT, we consider

- $\boldsymbol{h} = (h_0, ..., h_n)$ Is the n-dimensional vector (n=820) containing the number of entries in each bin of the LPMT+SPMT true E^{vis} spectrum.
- $\boldsymbol{\zeta}(\boldsymbol{h}) = (\zeta_0(\boldsymbol{h}), ..., \zeta_n(\boldsymbol{h}))$ is the n dimensional vector containing the reconstructed E^{vis} LPMT+SPMT spectrum.

Since, like in most sensitivities studies, resolution is simulated via a gaussian smearing, $\boldsymbol{\zeta}$ can be expressed this way :

$$\zeta_i = \sum_{j=0}^{n} G(j, \sigma(E_j))(i) \cdot h_j \tag{7.20}$$

where $G(j, \sigma(E_j))(i)$ is the smearing function defined as

$$G(j, \sigma(E_j))(i) = \int_{\lfloor E_i \rfloor}^{\lceil E_i \rceil} \frac{1}{\sigma(E_j)\sqrt{2\pi}} e^{-\frac{(E_j - E)^2}{2\sigma(E_j)^2}} \, \mathrm{d}E \tag{7.21}$$

where E_j is the mean energy in the bin j and $\lfloor E_i \rfloor$ and $\lceil E_i \rceil$ are the lower and higher energy bound of the jth bin respectively.

According to 7.21, to evaluate V, the matrix describing the uncertainties on ζ_i's and the correlations between them, one has to consider uncertainties both on h_j 's and on $G(j, \sigma(E_j))(i)$'s. We use linear error propagation and split this problem in two steps : $V = V_{inputs} + V_{rec}$. The first matrix accounts for the uncertainties on the inputs from the true E^{vis} spectrum (h_i's), while the second concerns the uncertainties due to $G(j, \sigma(E_j))(i)$'s.

To evaluate V_{inputs}, we use $V_{inputs} = AUA^T$ where U is the covariance matrix of the LPMT+SPMT true E^{vis} spectrum. Since before reconstruction the LPMT and SPMT spectra are the same, this LPMT+SPMT is the juxtaposition of two 410-bin identical spectra. Moreover we are interested only in statistical uncertainties. Therefore, U has the form :

$$U = \begin{cases} \sqrt{h_i h_j} & \text{if } i = j \text{ or } |i-j| = 410 \\ 0 & \text{otherwise} \end{cases} \tag{7.22}$$

The condition $|i-j| = 410$ express the fact that one h_i of the LPMT part of the spectrum is naturally 100% correlated with the corresponding binin the SPMT spectrum.

We can then construct the transfer matrix A as

$$A_{ij} = \frac{\partial \zeta_i}{\partial h_j} = G(j, \sigma(E_j))(i) \tag{7.23}$$

and then compute the first part of our covariance matrix

$$V_{inputs} = AUA^T \tag{7.24}$$

Now we need to consider the uncertainty on the steaming from the resolution, ie to evaluate V_rec. It can be done considering no uncertainty on the true E_{vis} spectrum. The quantity $G(j,u) \equiv G(j, \sigma(E_j))(i)$ is the predicted probability for an event initially in bin j of the true Evis spectrum to be reconstructed in bin i. In pratice, the migration between these bins is a random process. Reconstructed many times the same event would not lead each time the same migrations. We need here to determine this variability. We consider that with 410 bins, migrations vary independently whatever i and j.

This allows to consider V_{rec} as diagonal, thus we only need $\sigma G(j,i)$. We can derive this term from two equation:

— The term $G(j,i) \cdot h_j$ represent the number of event smeared from the bin j that end up in the bin i. This is a number, we thus assume poissonian statistic so that $\sigma[G(j,i) \cdot h_j] = \sqrt{G(i,j) \cdot h_j}$.

— Using basic error propagation we can say that $\sigma^2[G(i,j) \cdot h_j] = h_j^2 \sigma^2 G(j,i) + G(j,i)^2 \sigma^2 h_j$.

Equating the above equations, and remembering that $\sigma h_j = \sqrt{h_j}$ since h_j is also a number of events :

$$G(j,i)h_j = \sigma^2[G(j,i)h_j] = h_j^2 \sigma^2 G(j,i) + G(j,i)^2 h_j \tag{7.25}$$

$$\Rightarrow \sigma^2 G(j,i) = \frac{G(j,i)h_j - G(j,i)^2 h_j}{h_j^2} \tag{7.26}$$

$$= \frac{(1 - G(j,i))G(j,i)}{h_j} \tag{7.27}$$

By summing the two covariance matrix V_{inputs} and V_rec, we can extract a correlation matrix presented in Figure 7.11. Typically, a bin in the SPMT part of the reconstructed spectrum is correlated up to a few percents to the corresponding bin in the LPMT spectrum and its neighbour. This might seem a small correlation. However, its concerns all bins. The global impact is therefore high. As an illustration, as seen in Section 7.2.2, the correlation between the value of $\sin^2(2\theta_{12})$ measured with the LPMT spectrum and that measured with the SPMT spectrum are correlated at more than 95%.

The correlation between the SPMT and LPMT spectra is greater at the start of the spectrum. This is expected since the absolute resolution is smaller in this region. For instance, at 1.5 MeV, the reconstruction by the SPMT re-distribute events with a sigma of more that 0.20 MeV. At 6 MeV, this is about twice more. Since the resolution reduces the initial correlations (true Evis spectra are share by both LPMT and SPMT, correlations are 100%), we therefore expect higher remaining correlations where the absolute resolution is smaller.

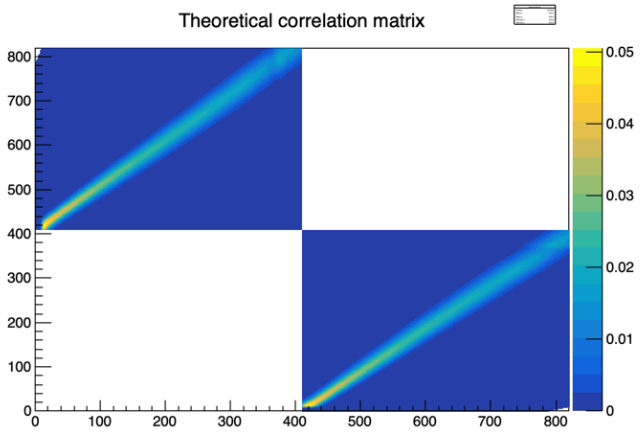


FIGURE 7.11 – Theoretical correlation matrix between the LPMT spectrum (bins 0-409) ans the SPMT spectrum (410-819). The diagonal has been set to 0 (it was 1) for readability purpose.

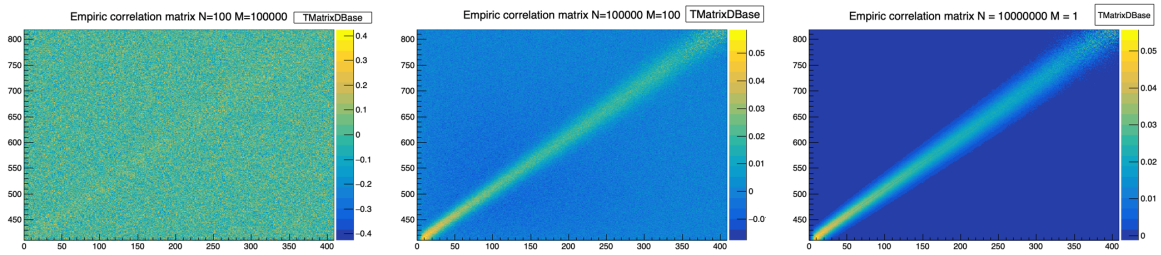
7.5.2 Empirical method

The second method is the emprical way where we generate toys and just compute the empirical correlation between the bin contents.

$$\mathrm{Corr}(h_i, h_j) = \frac{\mathbb{E}[h_i h_j] - \mathbb{E}[h_i]\mathbb{E}[h_j]}{\sigma h_i \sigma h_j} \tag{7.28}$$

We thus generate 10^7 event using the IBD generator presented in Section 7.3.3, then produce spectra from this finite set of events, meaning we must choose a number N of toy each composed of M event in order to have the best estimate.

It can be shown that empirical correlations are more precise when one maximises the number of samples, even at the price to have few events per sample. This effect is illustrated in Figure 7.12.



(A) (B) (C)

FIGURE 7.12 – Upper left corner of the estimated correlation matrix between the LPMT and SPMT spectrum for different configuration of N toy with different number of M events per toy. We observe that the statistical uncertainty, the noise effect, diminish with the number of toy considered.

The relative difference between the element of the theoretical matrix of Figure 7.11 and the empiric correlation matrix in Figure 7.12c is presented in Figure 7.13. Typically, correlations coefficient differ by 20% of their value. We have verified that differences larger than this are confined in the very low or high end of the energy spectrum, which carry no sensitivity to the solar oscillation parameters we

aim at. Therefore, for the statistical tests presented in this chapter we assume the correlations present in the theoretical version of V. This should account for the effect of correlations well enough.

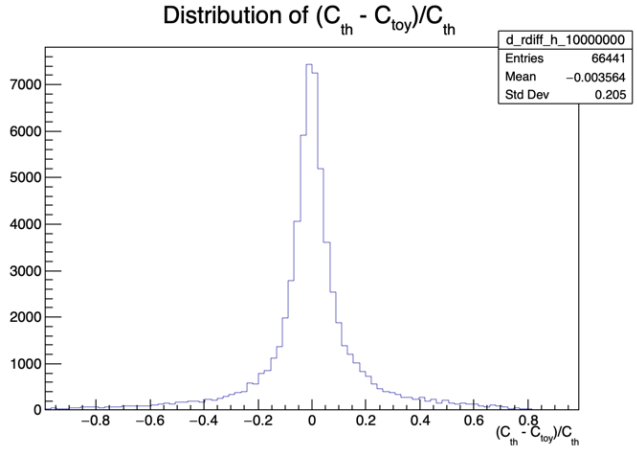


FIGURE 7.13 – Relative difference between the element of the theoretical and empiric correlation matrix

We chose to do so for practical reasons. Indeed, for the χ^2 computation and the Choleski decomposition (see Section 7.3.2) the matrix must be invertible and positive definite. The statistical uncertainty on the coefficient of the empiric matrix can prevent that, leading to complications.

7.6 Technical Validation

Standard Independent Joint Fit

We have already explained in Sections 7.2 and 7.5 that a correlation exist between the SPMT and LPMT spectra, and is accounted for in the LPMT+SPMT joint fit by the V covariance matrix, which determination is described in the previous section.

We can, however, perform a test where we ignore these correlations, setting to 0 all off-diagonal elements of V. In this case, we implicitly assume that our data contains more information, and therefore expect the uncertainties on Δm^2_{21} and $\sin^2(2\theta_{12})$ to be smaller that those obtained with individual fits to the LPMT and SPMT spectra. Assuming a gaussian behavior of the number of entries per bin, these uncertainties should be close to the weighted average of the uncertainties with the individual fits :

$$\frac{1}{\sigma^2_{Weighted}} = \frac{1}{\sigma^2_{LPMT}} + \frac{1}{\sigma^2_{SPMT}} \tag{7.29}$$

These tests are performed using an Asimov sample. Indeed, if it was done via a toy study, then generating correlated toy spectra and fitting them assuming a diagonal V matrix would have let to biases, regardless of the quality of the technical implementation. Asimov spectra, on the other hand, are generated with no fluctuations. They are supposed to return fitted values of Δm^2_{21} and $\sin^2(2\theta_{12})$ exactly equal to the values assumed during the generation. This is, together with the comparison with σ_{weight}, a strong test of the technical implementation.

Note that we fix here the δm^2_{21} and $\delta \sin^2(2\theta_{12})$ parameters to 0. Also, we assume 6 years of data taking, and the absence of unexpected instrumental effects (no supplementary QNL). A notable difference with the fit configuration used later in this chapter (and presented in Section 7.2) is that we do not treat Δm^2_{31} as a nuisance parameter. It is free to float.

	$\sigma(\Delta m^2_{21})$ [eV^2]	$\sigma(\delta\Delta m^2_{21})$ [eV^2]	$\sigma(\sin^2(2\theta_{12}))$	$\sigma(\delta\sin^2(2\theta_{12}))$	$\sigma(\Delta m^2_{31})$ [eV^2]	χ^2
LPMT	1.29×10^{-07}		1.33×10^{-03}		4.39×10^{-06}	3.23×10^{-18}
SPMT	1.38×10^{-07}		1.38×10^{-03}			2.87×10^{-18}
Indep Standard joint	9.48×10^{-08}		9.86×10^{-04}		4.39×10^{-06}	6.10×10^{-18}
Standard joint	1.29×10^{-07}		1.18×10^{-03}		4.39×10^{-06}	3.38×10^{-18}
Weighted	9.46×10^{-08}		9.63×10^{-04}			
Delta joint	1.35×10^{-07}	3.43×10^{-08}	1.38×10^{-03}	1.46×10^{-04}	4.39×10^{-06}	3.38×10^{-18}
Indep Delta joint	1.38×10^{-07}	1.89×10^{-07}	1.38×10^{-03}	1.87×10^{-03}	4.39×10^{-06}	6.10×10^{-18}

TABLE 7.4 – Uncertainties on each parameters reported by Minuit on Asimov studies. LPMT and SPMT rows are the results on the individual fit on each spectra. The Weighted row correspond to the weighted average uncertainties between the LPMT and SPMT fits following Eq. 7.29. The Indep Standard joint row is the result of the joint LPMT+SPMT fit but the off-diagonal terms are set to 0. The Indep Standard joint and Standard joint fits both are LPMT+SPMT fit but the parameters δm^2_{21} and $\delta\sin^2(2\theta_{12})$ are fixed to 0. The Delta joint and Indep Delta joint are LPMT+SPMT fit with δm^2_{21} and $\delta\sin^2(2\theta_{12})$, difference being that in the Indep version, the off-diagonal terms of the covariance matrix are set to 0.

The results are reported in Table 7.4. All those test are ran considering statistics error only, 6 years exposure with all backgrounds, $\sin^2(2\theta_{13})$ fixed to its nominal value. For the SPMT individual fit Δm^2_{31} is fixed at its nominal value as the SPMT system is not sensitive to this parameter. We use here the simple Pearson χ^2. Indeed, as explained above, an Asimov fit is supposed to find exactly the values of the parameters assumed for the generation of the spectrum, which implies a very low Pearson χ^2 (0 modulo numerical effects). This is also a strong indication that the technical implementation is correct. If we had used the usual Pearson V χ^2, the $\ln|V|$ term would have made the result more difficult to interpret.

When we performed the Standard Independent Joint Fit, as expected we observed that the fitted values of the parameters all matched the generation values. We can also see in table 7.4 that the uncertainty on Δm^2_{21} evaluated by the fit are equals the corresponding $\sigma_{weighted}$ up to 0.2%. In the case of $\sin^2(2\theta_{12})$, the agreement is up to 2.5%.

A slight difference exists in statistic between the SPMT and LPMT spectra. Indeed, due to a larger smearing in energy resolution, events that would be inside the spectrum range $[0.8, 7.5]$ MeV are smeared outside it. The $\sin^2(2\theta_{12})$ parameter being mainly driven by the amplitude of the spectrum (see illustration 7.1), it is more affected than Δm^2_{21}.

Standard Joint Fit

This case is similar to the previous one, with one difference : we now use the version of V that accounts for the correlations between the SPMT and LPMT spectra. The expected effect of this correlation is that the uncertainties on Δm^2_{21} and $\sin^2(2\theta_{12})$ should see very little improvement with respect to individual fits.

Moreover, the uncertainty on Δm^2_{31} should be very close to that obtained by the individual fit to the LPMT spectrum since only this one contains information on Δm^2_{31} (thanks to its high energy resolution). This is therefore a rather robust test.

As can be seen in Table 7.4, these expectations are observed in practice.

Delta Joint Fit

It is the same fit as above, where we let the $\delta\Delta m^2_{21}$ and $\delta \sin^2(2\theta_{12})$ parameters free to float in the fit. A test assumes no correlations (diagonal V), the other one assumes the usual V.

A first test here is that the fitter should find these parameters at 0, since no QNL is introduced in these Asimov spectra. Also, in the correlated case, we expect the uncertainties on $\delta\Delta m^2_{21}$ and $\delta \sin^2(2\theta_{12})$ to be far smaller than in the independent case. Indeed, when the χ^2 considers these two spectra are correlated, distorting only the LPMT part of the PDF without changing the SPMT part (remember: $\delta\Delta m^2_{21}$ and $\delta \sin^2(2\theta_{12})$ appear only in $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$ for the 410 first bins, see Section 7.2) leads to a quick explosion of this χ^2 when profiling values of $\delta\Delta m^2_{21}$ and $\delta \sin^2(2\theta_{12})$ away from 0.

Results in Table 7.4 are again consistent with these expectations.

Toy studies

The same tests as above have been repeated, using a set of 1000 toy samples instead of one Asimov sample. Only cases where we account for the correlations between the SPMT and LPMT spectra are carried out. The generation of the toy samples includes these correlations. We therefore also test that part.

We can see on Figures 7.14 and 7.15 the distribution of the best fit values for all the parameters of interest. The mean values and standard deviations are in all cases consistent with the results of the Asimov tests (Table 7.4). Therefore, when realistic fluctuations are simulated, even with a peculiar χ^2 computed with a complex covariance matrix and correlated data, the fit is stable and unbiased.

These distributions also confirm that the uncertainties on $\delta\Delta m^2_{21}$ and $\delta \sin^2(2\theta_{12})$ are an order of magnitude smaller than the uncertainties on Δm^2_{21} and $\sin^2(2\theta_{12})$. This is an indication of the power of the test statistics used in this chapter.

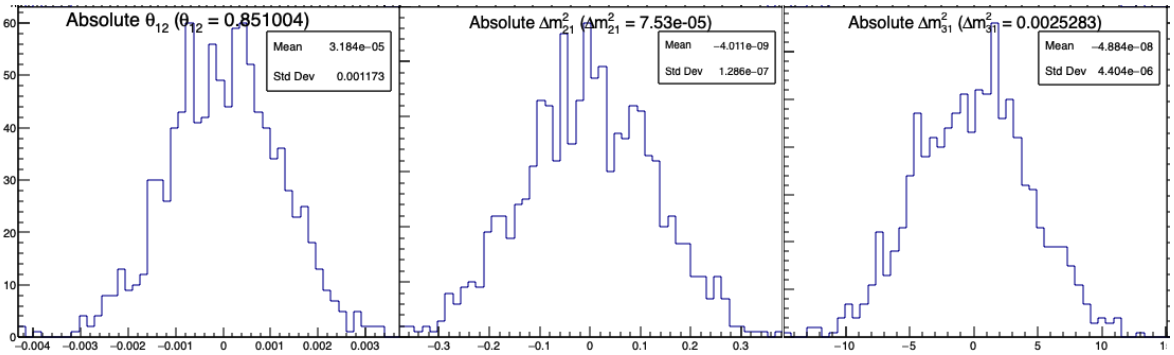


FIGURE 7.14 – Distribution of BFP - nominal value for 1000 toy Standard joint fit. 6 years exposure, all background, **PearsonV** χ^2, θ_{13} fixed. In those plots, θ_{12} stands for $\sin^2(2\theta_{12})$

Conclusion of the technical validation

All the tests carried out in this section are consistent with our expectation. We therefore conclude that the technical implementation of the tools used in this chapter is correct.

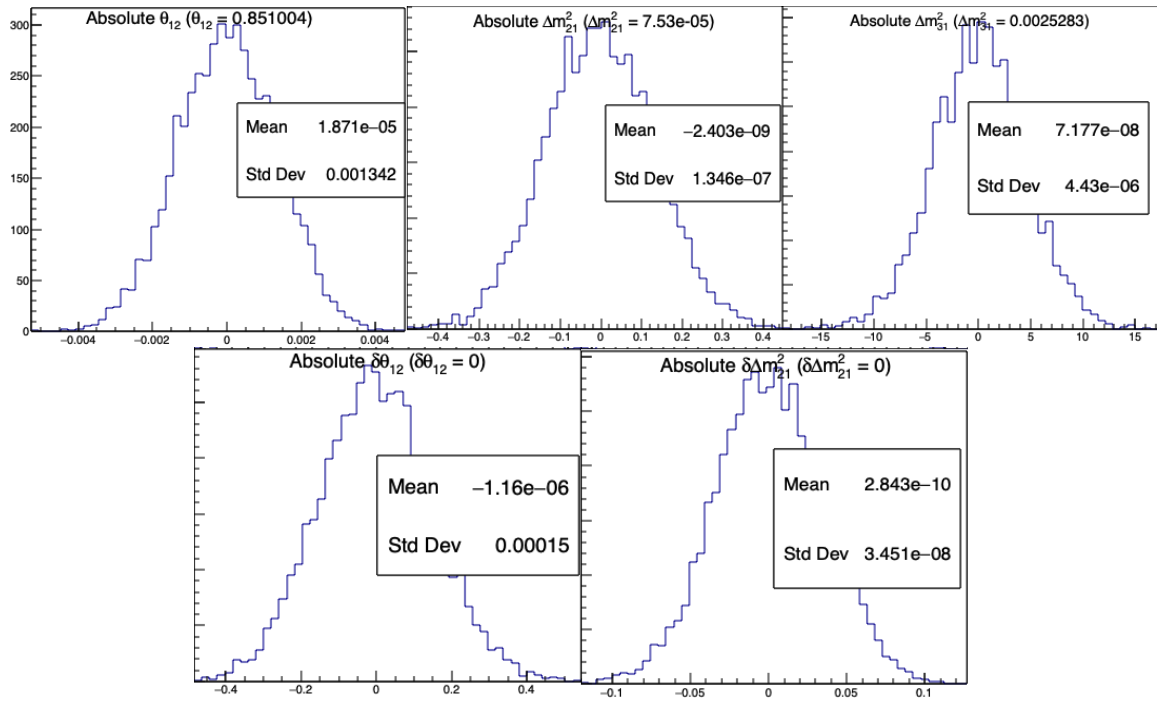


FIGURE 7.15 – Distribution of BFP - nominal value for 5000 toy Delta joint fit. 6 years exposure, all background, **PearsonV** χ^2, θ_{13} fixed. In those plots, θ_{12} stands for $\sin^2(2\theta_{12})$ and $\delta\theta_{12}$ for $\delta\sin^2(\theta_{12})$

7.7 Results

7.7.1 Effect of supplementary QNL on the LPMT spectrum

In this first part of this Section 7.7, we will present the sensitivity of various test statistics to unexpected instrumental effects affecting the SPMT and LPMT differently. The latter effects will be illustrated by generated toy samples affected by the QNL effect.

Most of the tests involve either an individual fit to the LPMT spectrum or the SPMT spectrum, or a joint fit of these two spectra. To better understand why some test statistics turn out to be more powerful than others, we study briefly in the present subsection the results of these fits and interpret the differences.

We generate toy spectra, and fit them according to the default configuration described in Section 7.2. During the generation of the LPMT spectrum, we distort it to simulated a QNL effect, with an intensity of $\alpha_{qnl} = 1\%$. For reference, this is about three time the expected residual QNL after the application of dual calorimetric calibration methods ($\alpha_{qnl} = 0.3\%$ [26]).

Backgrounds had to be ignored here: the JUNO inputs described in Section 7.3 provide a reconstructed spectrum, but not the event per event information about the true E_{vis}, which we need to apply the QNL effect (See Equation 7.19).

The effect of this QNL on the spectrum is illustrated in Figure 7.16 In Table 7.5 we report the results of the different kinds of fits.

We notice (1st line, first 3 columns) that the individual fit to the LPMT spectrum tends to find, as expected, biased value for Δm^2_{21} and $\sin^2(2\theta_{12})$ and Δm^2_{31} (biased at about -1 sigma, -1.3 sigma and - 2.2 sigmas respectively). When a joint fit is performed, with the $\delta\Delta m^2_{21}$ and $\delta\sin^2(2\theta_{12})$ fixed at 0,

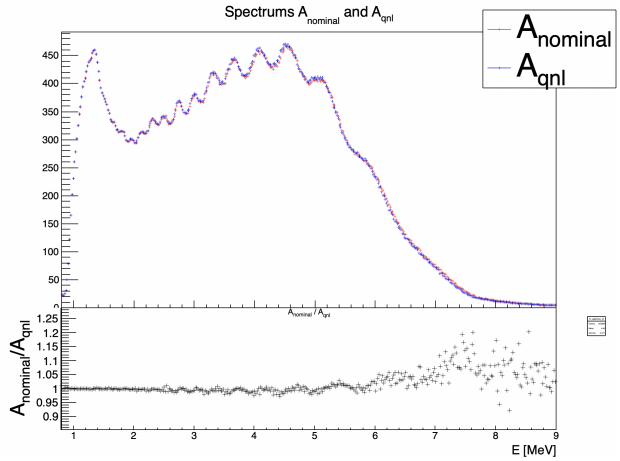


FIGURE 7.16 – **Top:** Theoretical spectrum without QNL (in red) and with $\alpha_{qnl} = 1\%$ (in blue). **Bottom:** Ratio between the theoretical spectrum with and without QNL.

and ignoring in the computation of the χ^2 the correlations between the LPMT and SPMT spectra, the biases on Δm^2_{21} and $\sin^2(2\theta_{12})$ (3rd line, first 3 columns) appear to be average of the biases seen by the individual fits to these spectra, a logical result since the individual sensitivities to these parameters are similar. The bias on Δm^2_{31} (3rd column) remains the same as with the individual fit to the LPMT spectrum, however, which is expected since the SPMT spectrum carries no sensitivity to Δm^2_{31}.

When the joint fit is performed with the nominal covariance matrix (determined in Section 7.5 assuming no QNL), biases on Δm^2_{21} and $\sin^2(2\theta_{12})$ explode: they are, respectively, about 6.5 and 2.5 times larger (4th line).

We explain it by the following mechanism : the fit tries to improve the agreement between the PDF and the data in the LPMT part of the spectrum by choosing biased values of the parameters. This in turn tends to deteriorate the agreement between the PDF and the SPMT spectrum (not distorted by QNL). In the end, a discrepancy remains between data and PDF in at least one sector (LPMT or SPMT) if not both. When the χ^2 is built with a matrix which accounts for the correlations, this discrepancy can make the χ^2 explode.

For instance, in some bins of the LPMT spectrum, we can imagine the PDF overestimates the QNL-distorted data, while the contrary happens in the corresponding bins of the SPMT spectrum. If the expected correlation is positive between these two bins, the χ^2 will reach values accounting for a larger discrepancy than if no correlation existed and if only the raw agreement between the pdf and the spectra was important.

In reality, the consistency between the two can be judged only accounting for the correlations. This is the important role of the covariance matrix in this work. In other words, the spectra predictions are not only the $\boldsymbol{T}(\boldsymbol{\theta}, \boldsymbol{\eta})$'s, but also the correlations.

Another point must be noted : the correlation matrix V is evaluated assuming no QNL. With the QNL effect added, the actual correlations between the LPMT and SPMT generated toy spectra is a bit different, adding another source of discrepancy between the data and the predictions, and further increasing the χ^2.

All in all, the minimisation of the χ^2 requires a larger scan of the oscillation parameters values than when correlations are ignored. Values can be chosen which are farther from the nominal ones, meaning larger biases.

This is actually an advantage. Indeed, we can see in table 7.5 that when $\delta\Delta m^2_{21}$ and $\delta \sin^2(2\theta_{12})$ are allowed to float in the fit, they "absorb" a large part of the bias. Notice in particular that adding the

Mean (std dev)	θ_{12} [10^{-3}]		Δm^2_{21} [10^{-7}eV^2]		Δm^2_{31} [10^{-6}eV^2]		$\delta\theta_{12}$ [10^{-3}]		$\delta\Delta m^2_{21}$ [10^{-7}eV^2]	
LPMT	-1.569	(1.171)	-0.957	(0.989)	-8.235	(3.898)	Irrelevant		Irrelevant	
SPMT	-0.164	(1.191)	-0.603	(1.054)	Not sensitive		Irrelevant		Irrelevant	
Indep Standard	-0.880	(1.174)	-0.786	(1.004)	-8.195	(3.900)	Irrelevant		Irrelevant	
Standard	-8.106	(1.423)	-2.483	(1.018)	-6.649	(4.008)	Irrelevant		Irrelevant	
Indep Delta	-0.169	(1.190)	-0.598	(1.054)	-8.234	(3.899)	-1.397	(0.259)	-0.361	(0.366)
Delta	-0.163	(1.183)	-1.532	(1.036)	-8.193	(3.934)	-1.441	(0.193)	0.654	(0.303)

TABLE 7.5 – In each column, the mean of the distribution of the 1000 best fit values found by fitting the 1000 toy samples with $\alpha_{qnl} = 1\%$ is shown, from which we subtracted the value assumed when generating the toys. A value different from 0 indicates a bias. Between bracket, the average uncertainty of the fitted value is also shown. It allows to judge of the severity of the bias. For instance, the measurement of $\sin^2(2\theta_{12})$ by fitting only the LPMT spectrum tends to be biased at the -1.569/1.171 = -1.34 sigma.

value of $\delta\Delta m^2_{21}$ to the remaining bias on Δm^2_{21} (last line, columns 1 and 4) one retrieves the bias of the individual fit to the LPMT spectrum. The same applies to $\sin^2(2\theta_{12})$. Consequently, large values of $\delta\Delta m^2_{21}$ and $\delta\sin^2(2\theta_{12})$ are expected, hence high significances to help us to detect the distortion. In this case (last line, column 4 and 5), we see the most probable values of the fitted $\delta\Delta m^2_{21}$ and $\delta\sin^2(2\theta_{12})$ parameters differ from zero at about 7.46 sigma and 2.2 sigma.

Based on the above observations, we expect the "$\chi^2_{H_0} - \chi^2_{H_1}$" and "Distributions of $\delta\Delta m^2_{21}$ and $\delta\sin^2(2\theta_{12})$" test statistics described in sections 7.2.4 and 7.2.5 to have the highest power. The "Direct comparison between the SPMT and LPMT spectra" should perform in the same ballpark. Finally, the "Comparison of individual fits" is expected to be have less power.

7.7.2 Comparison and statistical tests results

I present in this following Subsection the results from the tests and comparison detailed in section 7.2. For each distribution we compute the median p-value with respect to the distribution $\mathcal{D}(\alpha_{qnl} = 0\%)$. For this, we compute the median value of the distribution of interest $\mathcal{D}(\alpha_{qnl})$, then compute the p value

$$p = \frac{N(\mathcal{D}(0) > \text{Median}[\mathcal{D}(\alpha_{qnl})])}{N_{tot}} \tag{7.30}$$

where $N(\mathcal{D}(0) > \text{Median}[\mathcal{D}(\alpha_{qnl})])$ is the number of toy in the distribution $\mathcal{D}(\alpha_{qnl} = 0\%)$ that have a greater value than the median of the $\mathcal{D}(\alpha_{qnl})$. The p-value represent the probability for a non perturbed event to do worse that the median perturbed event.

The uncertainty on the p-value is computed using

$$\sigma p = \sqrt{\frac{p(1-p)}{N}} \tag{7.31}$$

which do not account for all uncertainties but serves as indicator.

Comparison of solar parameters from individual analysis: χ^2_{ind}

The results are presented in Figure 7.17. We see that the p-value are much less significant than the other tests, this is because this test possess much less information about the relation between the LPMT and SPMT systems.

This test is the most straightforward as it require only the fit of the two spectra and the estimation of the parameters covariances, but is also the less powerful with a p value for $\alpha_{qnl} = 0.3\%$ of 0.09 ± 0.009

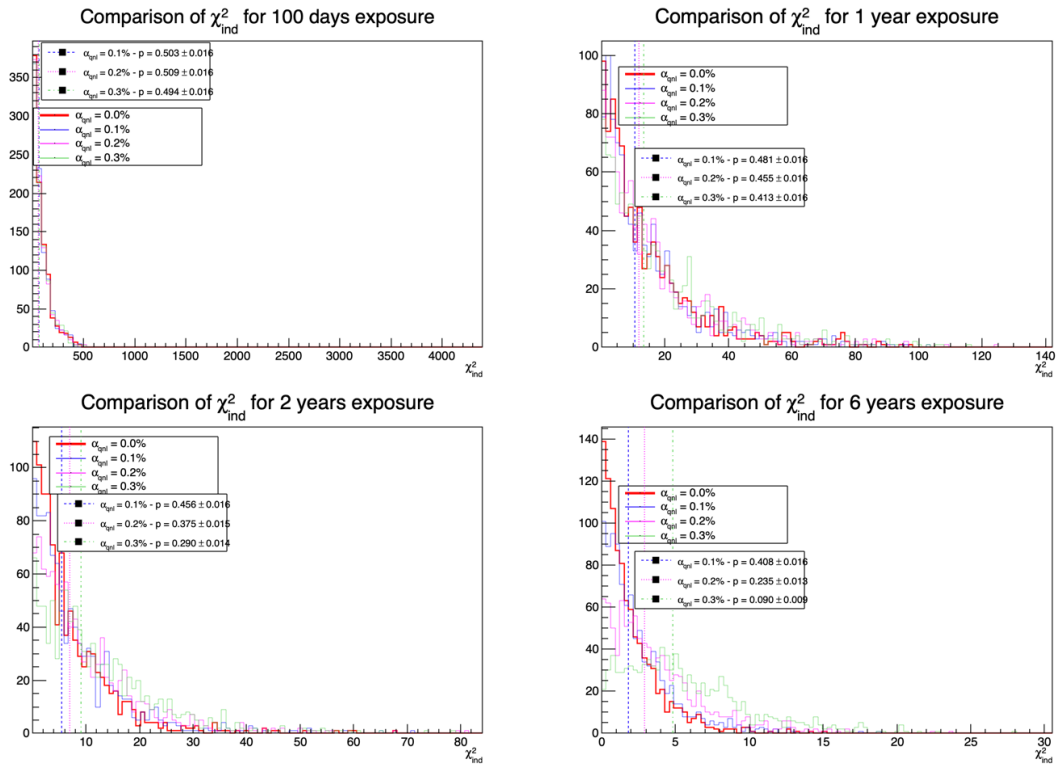


FIGURE 7.17 – Distribution of the χ^2_{ind} for 1000 toys for different exposures. The dashed lines represent the median of the distributions and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

at 6 years.

Direct comparison between the LPMT and SPMT spectra: χ^2_{spe}

The results for different exposures can be found in Figure 7.18. To give an idea of the significance of this test, we provide the median p-value for each test $\alpha_{qnl} \neq 0$. As expected, the power of this test rises as the exposure does. We see significant discrimination at 6 years for $\alpha_{qnl} \geq 0.3\%$ where the p-value for $\alpha_{qnl} = 0.3\%$ is 0.005 ± 0.0022.

This test relies solely on the estimated covariance matrix between the two spectra, requiring no fitting. As a result, it is a very lightweight test that can still provide valuable indications of potential unknown distortions between the two spectra.

Joint fit: $\chi^2_{H_0} - \chi^2_{H_1}$

This test is the most complex, requiring two fit and the covariance matrix between the LPMT and SPMT spectra. The results are presented in Figure 7.19.

The results are good, close to the χ^2_{spe}, one with a p-value at 6 years for $\alpha_{qnl} = 0.3\%$ of 0.01 ± 0.003. This sensitivity is consistent with that of χ^2_{spe}.

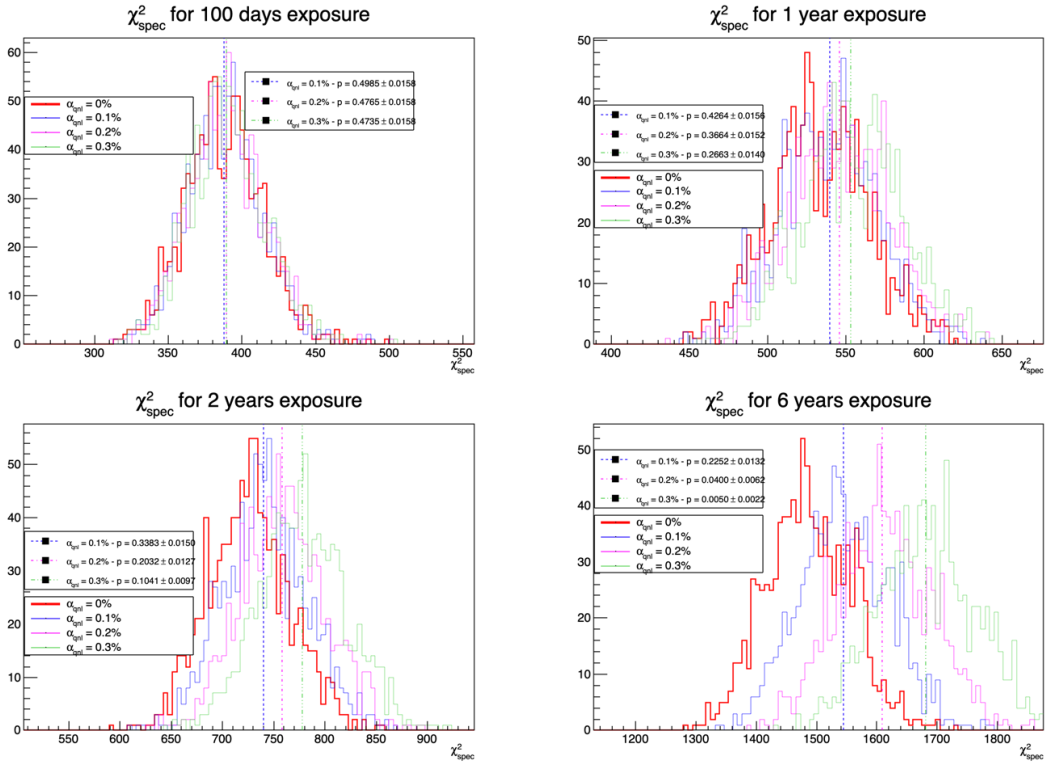


FIGURE 7.18 – Distribution of the χ^2_{spe} for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

Comparison of the parameters $\delta \sin^2(2\theta_{12})$ and $\delta\Delta m^2_{21}$

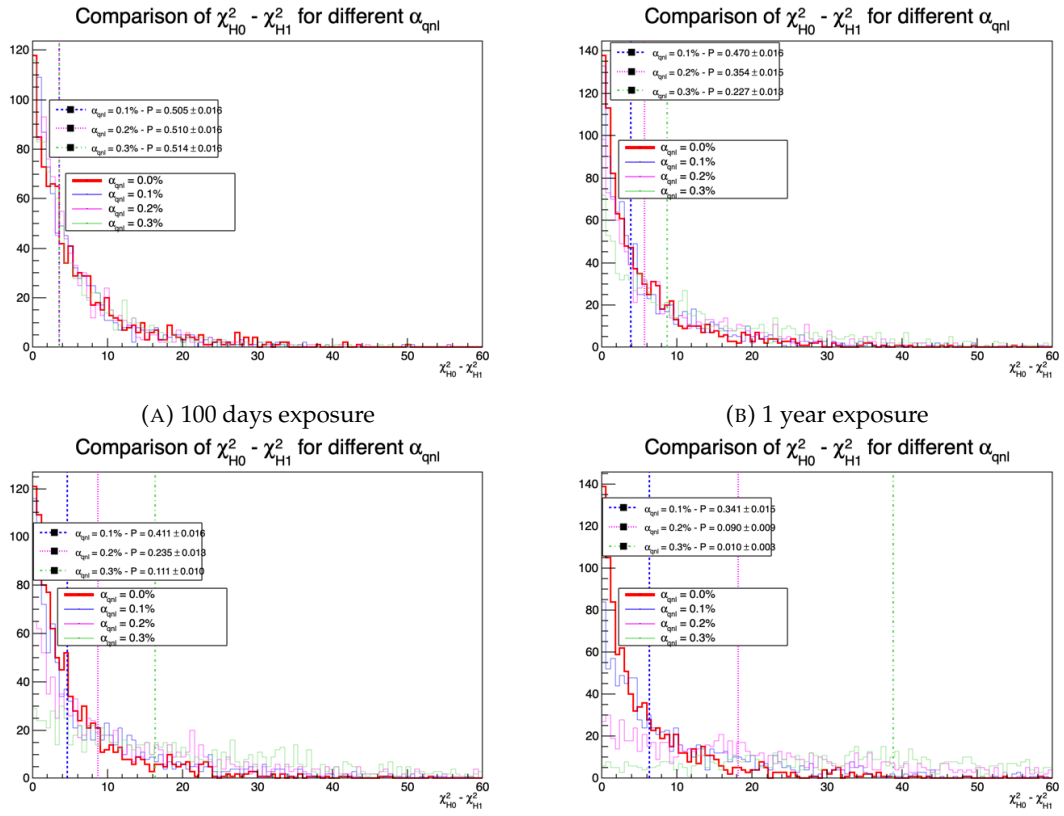
We can see that the $\delta\Delta m^2_{21}$ has a very small discriminative power (Figure 7.21) even at 6 years exposure with a p-value of 0.34 ± 0.01 for $\alpha_{qnl} = 0.3\%$. On the other hand $\delta\theta_{12}$ (Figure 7.20) has much more discriminative power with a p-value for $\alpha_{qnl} = 0.3\%$ of 0.025 ± 0.005. This test with a single joint fit seems to be still less powerful than the χ^2_{spe}. This can be explained as this method only get information through the oscillation parameters θ_{12} and Δm^2_{21} missing potential informations contained in Δm^2_{31}.

Summary

The p-values from the different test and comparison for $\alpha_{qnl} = 0.3\%$ are reported in Table 7.6.

	100 days	1 year	2 years	6 years
χ^2_{ind}	0.49	0.41	0.29	0.090
χ^2_{spec}	0.47	0.27	**0.10**	**0.005**
$\chi^2_{H_0} - \chi^2_{H_1}$	0.51	0.23	**0.11**	**0.010**
Comparison of $\delta \sin^2(2\theta_{12})$	0.39	0.2	0.14	0.025

TABLE 7.6 – Report of the p-value of the different tests and comparisons for $\alpha_{qnl} = 0.3\%$ for the different exposures.



(A) 100 days exposure

(B) 1 year exposure

(C) 2 years exposure

(D) 6 years exposure

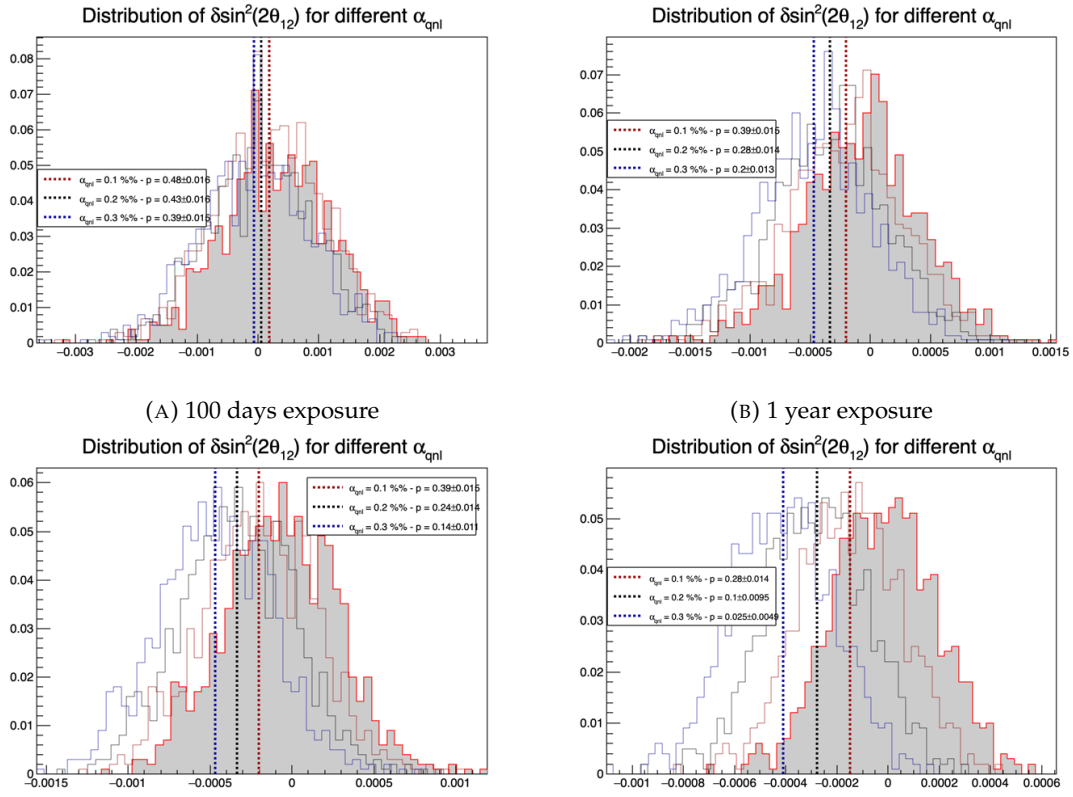
FIGURE 7.19 – Distribution of $\chi^2_{H_0} - \chi^2_{H_1}$ for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

7.8 Conclusion and perspectives

In this chapter, we present the development of a fit framework that allows us to fit multiple spectra simultaneously. We also introduce a set of tools that enable us to detect potential distortions in one of the two spectra. As an illustration of the capability of these tools, we use supplementary event-wise non-linearity and compare it to the potential residual event-wise non-linearity after calibration.

Table 7.6 gives a synthetic view of the strength of our methods. As expected, two methods that exploit the knowledge of the correlations between the SPMT and LPMT spectra obtain the best results. At high exposures, if the QNL effects are not calibrated out as well as expected ($> 0.3\%$), our best test statistics will be likely to detect them (median p-values below 10% after 2 years of data taking, about 1% after 6 years). In case of major effect (QNL or another unexpected instrumental effect) is worse , the detection will be even more likely. Below two years of data taking, only large unexpected instrumental effects can be detected.

One of JUNO most important goals is to determine the NMO independently of other experiments. This should not happen before 6 years of data taking. Our results demonstrate that dual calorimetry with neutrino oscillation can be a useful approach to help ensure the robustness of this result.



(A) 100 days exposure

(B) 1 year exposure

(C) 2 years exposure

(D) 6 years exposure

FIGURE 7.20 – Distribution of the $\delta \sin^2(2\theta_{12})$ for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

7.8.1 Empirical correlation matrix from fully simulated event

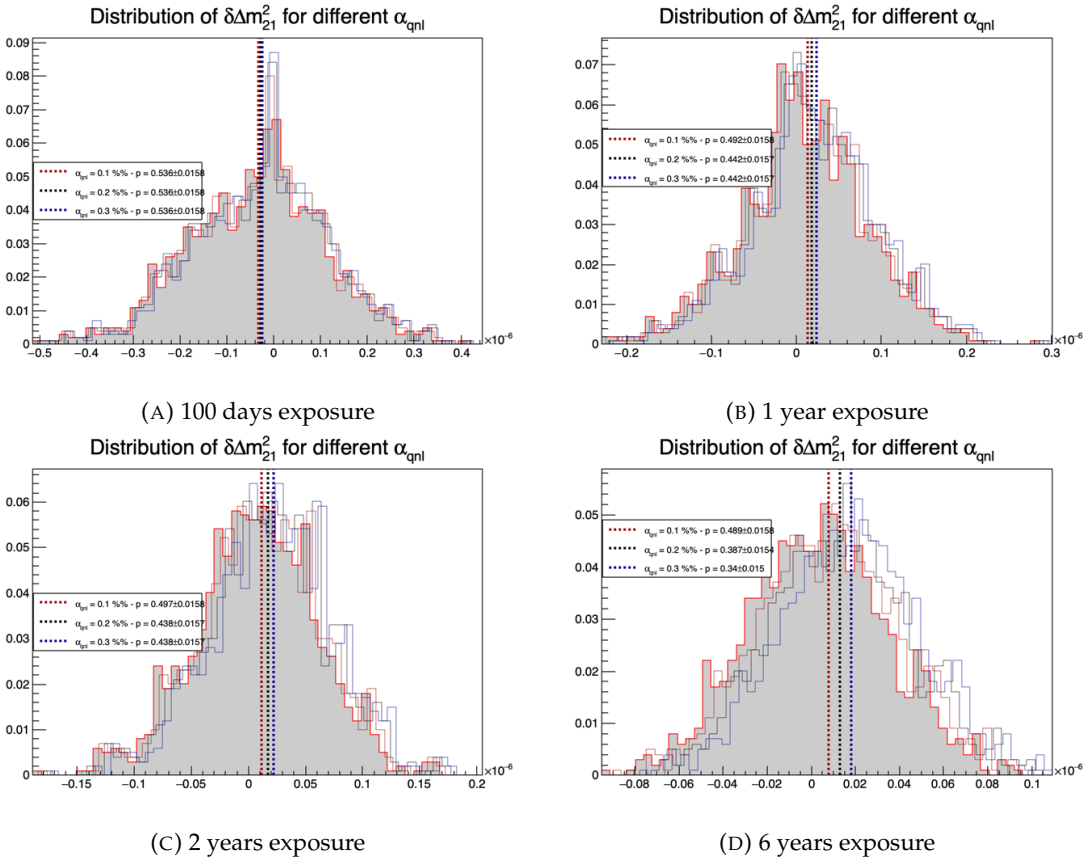
As already explained several times, one of the limitation of this work is that we assume the SPMT and LPMT energy reconstructions to be totally uncorrelated. In reality, this is not true. The V covariance matrix used in the test statistics should therefore be evaluated accounting for this. This involves complications that make the subject out of the scope of this thesis. We present here a brief study which goal is to get a rough idea of the impact of these reconstruction correlations.

The core of the idea is that the LPMT and SPMT reconstruction errors is bound to be correlated due to systematic effects. The first and most obvious one, for example, is energy escaping from the central detector. If the positron, or one of the two annihilation gamma, escape from the detector, less energy is deposited thus both of the systems will reconstruct a lower energy that was actually deposited. On a more subtle scale, the randomness in the production of scintillation photons is common for the two systems, if the liquid scintillator produces fewer scintillation photons for an event, both systems are likely to underestimate the energy.

We study those effects by computing from a dataset of IBD events, uniformly distributed in the CD, the correlation between the reconstruction errors on the energy

$$Corr(E_{rec}^{lpmt} - E_{vis}, E_{rec}^{spmt} - E_{vis}) \tag{7.32}$$

where E_{rec}^{lpmt} and E_{rec}^{spmt} are the reconstructed energies from both systems and E_{vis} the true visible



(A) 100 days exposure

(B) 1 year exposure

(C) 2 years exposure

(D) 6 years exposure

FIGURE 7.21 – Distribution of the $\delta\Delta m^2_{21}$ for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

energy. The OMILREC algorithm, presented in section 3.3, is used for the LPMT reconstruction E^{lpmt}_{rec}, and the CNN presented in Chapter 4 for the SPMT reconstruction E^{spmt}_{rec}.

The results of those correlations are presented in Figure 7.22 for the single energy and the interaction radius dependency, and Figure 7.23 for the dual energy and interaction radius dependencies.

The first observation here is that in most of the detector volume, the correlation between the SPMT and LPMT energy reconstructions does not exceed a few percents, and is in general positive.

In principle, this correlation must be dominated by the fluctuations of the photon yield produced in the scintillator, which dominates the stochastic term of the resolution (see Equation 7.19). Indeed, in a given event, both the LPMT and SPMT reconstruct the energy from the same photon yield and both are affected in the same way by a fluctuation. The correlation is reduced by the fact that SPMT system, due to its low coverage, detect only a very small fraction of the photon. This sampling is also a random phenomenon : the corresponding fluctuations hide to some extent the fluctuations of the original photon yield, and are essentially independent of the random number of photons sampled by the LPMT.

When energy is deposited at high R, close to or in the total reflection area, the proximity of the PMTs increases the number of photons detected by LPMT, and therefore reduce the sampling fluctuations. In this case, the fluctuation of the original photon yield is less shuffled by the sampling fluctuations and the resulting correlation between the LPMT and SPMT reconstruction reaches high values, up to 25% (Fig. 7.22, right).

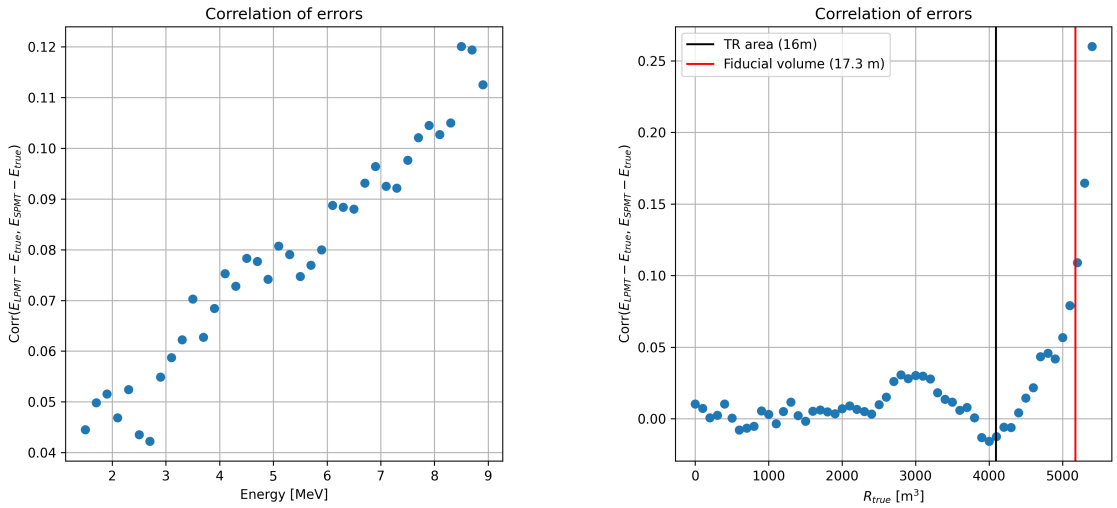


FIGURE 7.22 – Correlation on the reconstruction error between the LPMT and SPMT system as a function of **(On the left)** the energy, **(On the right)** the radius. The SPMT reconstruction comes from the NN presented in Chapter 4 and the LPMT reconstruction comes from OMILREC presented in Section 3.3. To prevent effect due to the CNN bad reconstruction, we select the event with $1 < E_{dep} < 9$ MeV.

The original photon yield grows with the visible energy. For the same reason as above, the correlation grows as well, albeit far more slowly than as a function of R^3. On Fig. 7.23, one can see that cumulating the effects of high energy and high R, correlations can reach 35%. However, in the fiducial volume and at energies below 7 MeV (ie in a part of the spectrum containing the sensitivity to Δm^2_{12} and $\sin^2(2\theta_{12})$), it never exceeds 15%.

To re-evaluate V with these reconstruction correlations accounted for, we should perform an empiric evaluation (like in Section 7.5.2). It would be based on toys generated with the IBD generator (see point 9 of Section 7.3.3), replacing the two independent random gaussian drawings by a drawing according to a 2 dimensional gaussian describing the $(E^{lpmt}_{rec} - E_{vis})$ vs. $(E^{spmt}_{rec} - E_{vis})$ distribution, and involving the correlations studied above.

A way must be found to include the variation of the correlation as a function of R and the E_{vis}. We have tried to define 2-dimensional regions in these variables, and defined each time the corresponding 2 dimensional gaussian. Then, we tuned the IBD generator to choose which of these gaussians to sample, based on the generated values of E_{vis} and R. Unfortunately, due to the limited statistics of the full simulation sample, the E_{vis} : R^3 regions were too wide. It lead to sawtooth variations of the correlation and mean values of the gaussian between neighbouring regions. The reconstructed spectra finally showed irregularities instead of a normal, smooth aspect, making it improper for any oscillation analysis.

Before a solution can be found to this problem, we limit our conclusions to this :

— The correlation between the LPMT and SPMT energy reconstruction is positive. Therefore, the SPMT and LPMT spectra should be more correlated than assumed in the statistical tests presented in this chapter. With a proper treatment, we can therefore expect a higher sensitivity to unexpected instrumental effects like QNL.

— In 80% of the detector's volume, reconstruction correlations are low, and should not impact much the sensitivities of our test statistics. If the dependence of the correlation on E_{vis} and R proved too difficult to model, one could cut IBDs reconstructed in the Total reflection area.

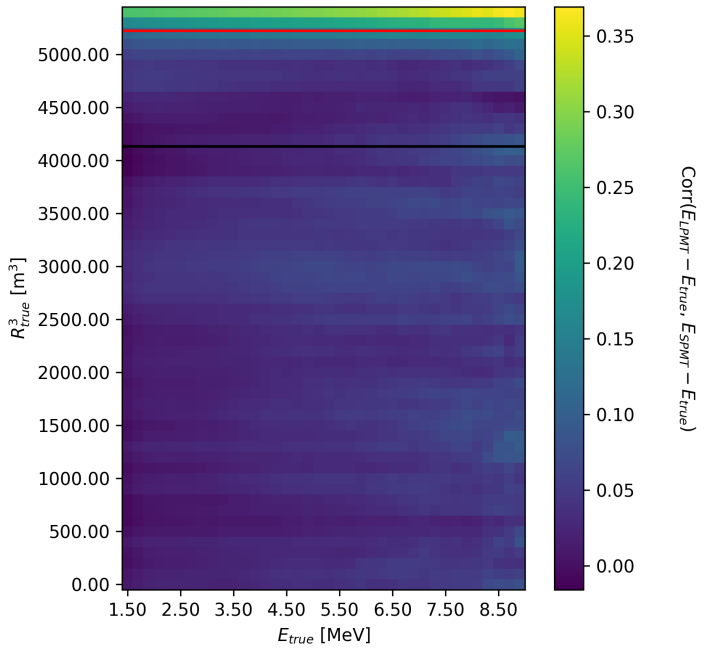


FIGURE 7.23 – Correlation on the reconstruction error between the LPMT and SPMT system as a function of the energy and the radius. The SPMT reconstruction comes from the NN presented in Chapter 4 and the LPMT reconstruction comes from OMILREC presented in Section 3.3. To prevent effect due to the CNN bad reconstruction, we select the event with $1 < E_{dep} < 9$ MeV.

The loss in statistics would be limited, as well as the impact on the sensitivities of the statistical tests.

Additionally, this study is preliminary, as the background was neglected in the distortion test, and no systematic uncertainties were considered. Those points could be easily addressed by regenerating background spectra using the same reference as used by JUNO for the common inputs and by regenerating the systematic covariances matrix with both LPMT and SPMT spectra.

The supplementary non-linearity was introduced event-wise but should be applied channel-wise to account for the detector's non-uniformity. This can be addressed via generating oscillated spectra through the JUNO official simulation. This process is very time consuming and require technical development but could be achievable given enough time.

The correlation matrix between the LPMT and SPMT spectra should also be further analyzed, as indicated by the discrepancies between the theoretical and empirical correlation matrices.

Summary and conclusion

The field of neutrino physics still has a lot of unanswered questions, namely the mass of the mass states, the Neutrino Mass Ordering (NMO), the possible existence of CP violation in the lepton sector, the unitarity of the PMNS oscillation matrix, and even the nature of the neutrino—Dirac or Majorana—is still unknown. To answer all of these questions, neutrino physics must advance into an era of precision measurements, of which JUNO will be a part.

JUNO is a 20 kton spherical liquid scintillator neutrino detector under construction that aims to measure the NMO with a confidence level of 3σ after 6.5 years of data taking. It will additionally measure the oscillation parameters θ_{12}, Δm^2_{21}, and Δm^2_{31} at the permille level. Additionally, it will run numerous other neutrino-related physics programs.

To measure the NMO and the oscillation parameters, JUNO will observe the electronic anti-neutrino spectrum from multiple nuclear power plants located 52 km away, a distance optimized to maximize the anti-neutrino disappearance. The reactor anti-neutrinos will interact with the liquid scintillator via Inverse Beta Decay. JUNO will extract the NMO and the oscillation parameters by observing the subtle interference patterns in the energy spectrum caused by the neutrino oscillation. In order to detect these interference patterns, JUNO needs an unprecedented energy resolution of $3\%/\sqrt{E\ \mathrm{MeV}}$ and a robust knowledge of the detector energy response, with the uncertainty kept under 1%.

To meet these stringent requirements, JUNO has developed sophisticated reconstruction and calibration techniques. A key element allowing for such techniques is the Dual Calorimetry, consisting of two photo-multiplier systems—large (LPMT) and small (SPMT)—each with its own characteristics. These two systems provide, almost, independent energy measurements of the same event. The presence of both systems not only enhances energy reconstruction but also provides valuable cross-checks, ensuring a thorough understanding of the systematic effects influencing JUNO.

To fully harness JUNO's capabilities and achieve the highest precision allowed by its experimental setup, we explore in this thesis the capabilities of Machine Learning (ML).

Machine learning algorithms, particularly Neural Networks (NN), have become increasingly popular among the physics community for a wide range of tasks, from event reconstruction and classification to event generation and waveform analysis. They indeed excel at extracting essential features from highly complex and multi-dimensional problems, such as the response of a physics detector.

We dedicated considerable time at the start of this thesis, through the development of a Convolutional Neural Network (CNN) presented in Chapter 4, to understanding the underlying governing mechanisms of NN. I present, in the introductory Chapter 3, a synthesis of the knowledge I gained.

Convolutional Neural Networks are a category of NN particularly efficient in processing images. This CNN was designed to reconstruct the interaction vertex and deposited energy of IBD events using solely the SPMT system. Its performance is compared with a previous reconstruction algorithm for SPMT that was developed at Subatech. This CNN shows similar performance in energy to the previously developed algorithm but worse performance in vertex reconstruction. Using an estimator combination method developed during this thesis, we have identified that there exists an algorithm that could achieve better performance than both algorithms individually.

We believe the limitations of this CNN stem from the planar representation of the spherical detector that is JUNO and the aggregation of PMT information in pixels. This representation induces deformation and information loss in the event. These problems could be circumvented either by a two-stage CNN that would first center the event in the middle of the image, reconstructing the orientation of the event, before reconstructing the radial component of the interaction vertex and the energy.

The problem of aggregation could be solved by transforming the time information, a scalar, into a supplementary dimension in the image, resulting in the stacking of successive planar projections, each representing a time slice of the event.

The limitations of CNN in JUNO prompted us to consider alternative architectures that could handle more elegantly JUNO's sphericity and keep the details of the raw information. Leveraging the knowledge gained from the development of our CNN, we decided to explore a novel and innovative Graph Neural Network architecture for IBD reconstruction.

Graph Neural Networks are networks processing graphs – a data structure composed of nodes holding features and edges representing the relations between these nodes. They work by propagating information across the graph, a.k.a message passing, which computes updated features on nodes and edges from neighboring nodes. Previous work in JUNO developed GNN where the nodes represent geometric regions of the detector. Those regions are only connected to their neighbouring regions.

In this thesis, I introduce in Chapter 5 a GNN that processes heterogeneous graphs, where the nodes are of different families. We use three families for JUNO, representing the fired PMT, geometric regions of the detector and global informations about events. This family classification allows us to fully connect the geometric regions of the detector while preserving the raw information. The ability to handle heterogeneous graphs is not provided by public frameworks, thus substantial technical development was necessary to implement our methods.

Among the global event information present in the graph are the results of a spherical harmonic analysis I developed that shows a correlation between the relative power of the harmonics and the radius of the IBD interaction.

This performance of this exploratory GNN are compared with the state of art likelihood reconstruction methods in JUNO. The results of the GNN are not on par with the performance of the likelihood methods. We explored the behavior of the GNN and identified potential problems in the propagation of information between the fired and geometric nodes. While the combination with the likelihood algorithm shows no substantial improvements, we believe that further work on the message passing algorithm and the incorporation of even more raw information such as the PMT waveform could still bring improvements to the IBD reconstruction in JUNO.

As already mentioned, JUNO's needs a robust understanding of its reconstruction, as small undetected biases could prevent us from measuring the NMO, or even worse prefer the wrong ordering. For this, wee need to trust our algorithm, and prove their reliability. This is even more important for ML algorithms, where, while we understand the global behavior is lead by their architecture, the interpretation of the detail of their optimisation is still subject to debate and research in the ML community.

We believe that the first step to ensure the reliability of the reconstruction is the comparison of a variety of algorithm. The combination method developed during this thesis allow to not only compare performance and behavior but also to probe in the difference in information used. For this, the necessity to make the reconstruction algorithm public to everyone in the collaboration is crucial. In the context of this work, in implemented ML algorithms developed in the collaboration inside JUNO official software.

A second step to ensure reliability is to probe for potential weakness in the reconstruction algorithm. In this thesis, in Chapter 6, I explore the potential of an Adversarial Neural Network (ANN) to produce physically plausible perturbation that would be undetected by the calibration system while still distort the reconstructed energy spectrum. We start in this thesis with simple neural network and while it is able to produce events, the task is too complicated to reach the desired results. More refinement of the architecture and potentially guiding the ANN in its perturbation strategy could help it.

JUNO relies on the Dual Calorimetry method to monitor and constrain our understanding of the reconstruction through calibration. In this thesis, I present in Chapter 7 the Dual Calorimetric analysis with neutrino oscillation that leverages the discrepancies between the oscillation analyses performed with each system. With this analysis, we try to detect discrepancies between the measured anti-neutrino energy spectra from the LPMT and SPMT system.

We choose to study the power of this analysis; we choose as a potential detector the Charge Non-Linearity (QNL). We show that at high exposures, if the QNL effects are not calibrated out as well as expected (greater than 0.3%), our best test statistics will be likely to detect them (median p-values below 10% after 2 years of data taking, and about 1% after 6 years). In the case of a major effect (QNL or another unexpected instrumental effect) being worse, the detection will be even more likely. Below two years' of data taking, only large unexpected instrumental effects can be detected.

During this thesis, several Neural Networks for IBD reconstruction, and the tools necessary for their understanding, have been developed. While they are not competitive with classical algorithms, they hint at potential improvements for future reconstruction algorithms. Due to the nature of the JUNO physics program and its stringent requirements, the reliability of those tools is crucial. To address this, we have explored a method based on Adversarial Neural Networks to probe for potential issues, and have pushed for the implementation of reconstruction methods in the collaboration software. To go even further in the detection of potential issues, we have developed the first Dual Calorimetric analysis with neutrino oscillation that will allow us to detect discrepancies between the LPMT and SPMT system.

Appendix A

Calculation of optimal α for estimator combination

This annex the details of the determination of the optimal α for estimator combination presented in section 4.3.2.

As a reminder, the combined estimator $\hat{\theta}$ of X is defined as

$$\hat{\theta}(X) = \alpha\theta_N + (1-\alpha)\theta_C;\ \alpha \in [0;1] \tag{A.1}$$

where θ_N and θ_C are both estimator of X.

A.1 Unbiased estimator

For the unbiased estimator, it is straight-forward. We search α such as $E[\hat{\theta}] = X$

$$\begin{aligned}
E[\hat{\theta}] &= E[\alpha\theta_N + (1-\alpha)\theta_C] && \text{(A.2)}\\
&= E[\alpha\theta_N] + E[(1-\alpha)\theta_C] && \text{(A.3)}\\
&= \alpha E[\theta_N] + (1-\alpha)E[\theta_C] && \text{(A.4)}\\
&= \alpha(\mu_N + X) + (1-\alpha)(\mu_C + X) && \text{(A.5)}\\
X &= \alpha\mu_N + \mu_C - \alpha\mu_C + X && \text{(A.6)}\\
0 &= \alpha(\mu_N - \mu_C) + \mu_C && \text{(A.7)}\\
& && \text{(A.8)}\\
\Rightarrow \alpha &= \frac{\mu_C}{\mu_C - \mu_N} && \text{(A.9)}
\end{aligned}$$

A.2 Optimal variance estimator

The α for this estimator is a bit more tricky. By expanding the variance we get

$$\begin{aligned}
\mathrm{Var}[\hat{\theta}] &= \mathrm{Var}[\alpha\theta_N + (1-\alpha)\theta_C] && \text{(A.10)}\\
&= \mathrm{Var}[\alpha\theta_N] + \mathrm{Var}[(1-\alpha)\theta_C] + \mathrm{Cov}[\alpha(1-\alpha)\theta_N\theta_C] && \text{(A.11)}\\
&= \alpha^2\sigma_N^2 + (1-\alpha)^2\sigma_C^2 + 2\alpha(1-\alpha)\sigma_N\sigma_C\rho_{NC} && \text{(A.12)}
\end{aligned}$$

where, as a reminder, ρ_{NC} is the correlation factor between θ_C and θ_N.

Now we try to find the minima of $\mathrm{Var}[\hat{\theta}]$ with respect to α. For this we evaluate the derivative

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathrm{Var}[\hat{\theta}] = 2\alpha\sigma_N^2 - 2(1-\alpha)\sigma_C^2 + 2\sigma_N\sigma_C\rho_{NC}(1-2\alpha) \tag{A.13}$$

$$= 2\alpha(\sigma_N^2 + \sigma_C^2 - 2\sigma_N\sigma_C\rho_{NC}) - 2\sigma_C^2 + 2\sigma_N\sigma_C\rho_{NC} \tag{A.14}$$

then find the minima and maxima of this derivative by evaluating

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathrm{Var}[\hat{\theta}] = 0 \tag{A.15}$$

$$2\alpha(\sigma_N^2 + \sigma_C^2 - 2\sigma_N\sigma_C\rho_{NC}) - 2\sigma_C^2 + 2\sigma_N\sigma_C\rho_{NC} = 0 \tag{A.16}$$

$$2\alpha(\sigma_N^2 + \sigma_C^2 - 2\sigma_N\sigma_C\rho_{NC}) = 2\sigma_C^2 - 2\sigma_N\sigma_C\rho_{NC} \tag{A.17}$$

$$\alpha = \frac{\sigma_C^2 - \sigma_N\sigma_C\rho_{NC}}{\sigma_N^2 + \sigma_C^2 - 2\sigma_N\sigma_C\rho_{NC}} \tag{A.18}$$

This equation shows only one solution which is a minima. From Eq. A.18 arise two singularities:

- $\sigma_N = \sigma_C = 0$. This is not a problem because as physicists we never measure with an absolute precision, neither us or our detectors are perfect.
- $\sigma_N = \sigma_C$ and $\rho_{CN} = 1$. In this case θ_C and θ_N are the same estimator in term of variance thus any value for α yield the same result: an estimator with the same varianve as the original ones.

Appendix B

Charge spherical harmonics analysis

When looking at JUNO events we can clearly see some pattern in the charge repartition based on the event radius as illustrated in figure B.4. When dealing with identifying features and pattern on a spherical plane, the astrophysics community have been using, with success, the spherical harmonic decomposition. The principle is similar to a frequency analysis via Fourier transform. It comes to saying that a function $f(r, \theta, \phi)$, here our charge repartition of the spherical plane constructed by our PMTs, can be expressed

$$f(r, \theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_l^m r^l Y_l^m(\theta, \phi) \tag{B.1}$$

where a_l^m are constants complex factor, $Y_l^m(\theta, \phi) = Ne^{im\phi} P_l^m(\cos\theta)$ are the spherical harmonics of degree l and order m and P_l^m their associated Legendre Polynomials. Those harmonics are illustrated in figure B.1. By reducing the problem to the unit sphere $r = 1$, we get rid of the term r^l. The Healpix library [71] offer function to efficiently find the a_l^m factor from a given Healpix map.

For the above decomposition, we will define the *Power* of an harmonic as

$$S_{ff}(l) = \frac{1}{2l+1} \sum_{m=-l}^{l} |a_l^m|^2 \tag{B.2}$$

and the *Relative Power* as:

$$P_l^h = \frac{S_{ff}(l)}{\sum_l S_{ff}(l)} \tag{B.3}$$

For this study we will use 10k positron events with $E_{kin} \in [0; 9]$ MeV uniformly distributed in the CD from the JUNO official simulation version J23.0.1-rc8.dc1 (released the 7th January 2024). All the event are *calib* level, with simulation of the physics, electronics, digitizations and triggers. We first take a sub-set of 1k events and look at the power and relative power distribution depending on the radius and harmonic degree l. The results are shown in figure B.2. While don't see any pattern in absolute power, it is pretty clear that there is a correlation between the relative power of $l = 0$ and the radius of the event.

When applying the same study but dependent on the energy, no clear correlation appear. The results for the $l = 0$ harmonic are presented in the figure B.5. Thus, in this study we will focus on the radial dependency of the relative power of each harmonic.

In figures B.6 and B.7 are presented the distribution of the relative power of each harmonic for $l \in [0, 11]$. The relation between the radius and the relative power become even more clear, especially for the first harmonics $l \in [0, 4]$. After that for $l > 4$ their relative power is close to 0 for central event, thus loosing power. It also interesting to note the change of behavior in the TR area, clearly visible for $l = 1$ and $l = 2$.

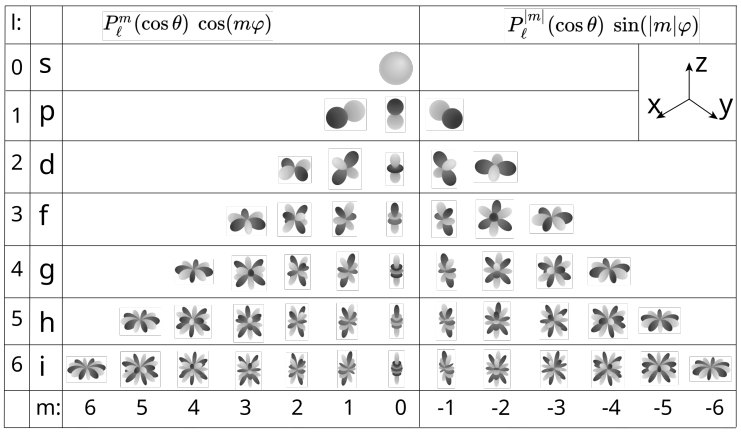


FIGURE B.1 – Illustration of the real part of the spherical harmonics

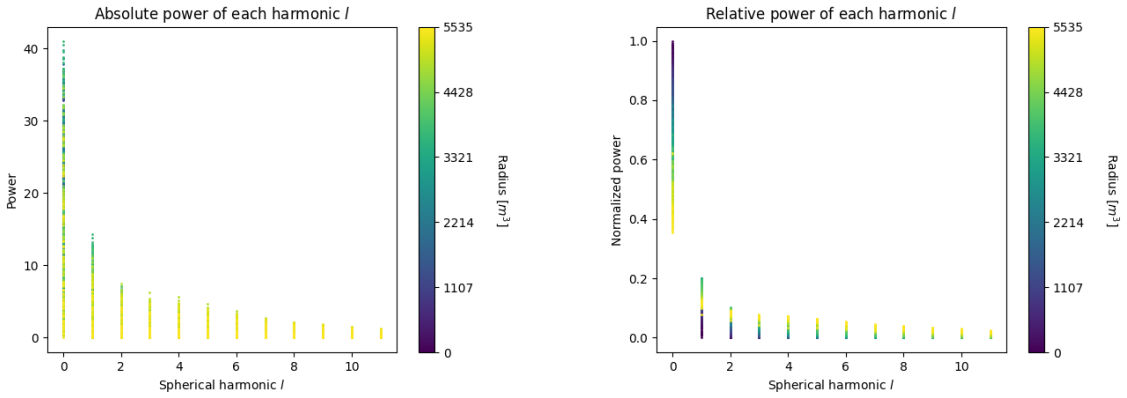


FIGURE B.2 – Scatter plot of the absolute and relative power, respectively on the left and right plot, of each harmonic degree l. The color indicate the radius of the event.

As an erzats of reconstruction algorithm, we fit each of those distribution with a 9th degree polynomial which give us the relation

$$F(R^3) \longmapsto P_l^h \tag{B.4}$$

We do it this way because some of the distribution have multiple solution for a given relative power, for example $l = 1$, while each radius give only one power. We now *just* need to find

$$F^{-1}(P_l^h) \longmapsto R^3 \tag{B.5}$$

Inverting a 9th degree polynomial is hard, if not impossible. The presence of multiple roots for the same power complexify the task even more. To circumvent this problem, we reconstruct the radius by locating the minima of $(F(R^3) - \hat{P}_l^h)^2$ where $\hat{P}_l^h$ is the measured power fraction.

To distinguish between multiple possible minima, we use as a starting point the radius given by the procedure on $l = 0$ that, by looking at the fit in figure B.6, should only present one minima. For $l > 0$ we also impose bound on the possible reconstructed R^3 as $R^3 \in [R_0^3 - 100, R_0^3 + 100]$ where R_0^3 is the reconstructed R^3 by the harmonic $l = 0$.

The minimization algorithm used are the Bent algorithm for $l = 0$ and the Bounded algorithm for $l > 0$ provided by the Scipy library [85]. We then do the mean of the reconstructed radius from the different harmonics. The reconstruction results are shown in figure B.3. The performance seems correct but we see heavy fluctuation in the bias. To really be used as a reconstruction algorithm, the method needs to be refined as discussed in the next section.

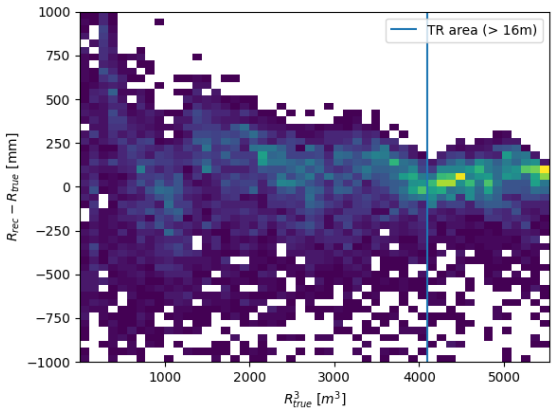


FIGURE B.3 – Error on the reconstructed radius vs the true radius by the harmonic method

Conclusion

We have clearly shown in this analysis the relevance the of relative harmonic power for radius reconstruction, and provided an erzats of a reconstruction algorithm. We will not delve further in this thesis but if we wanted to refine this algorithm multiple paths can be explored:

- No energy signature in the harmonics: This is surprising that there is no correlation between the energy and the amplitude of the harmonics. We know that the energy is heavily correlated with the total number of photoelectrons collected, it would be unintuitive that we see no relation.
- Localization of the event: We shown here the relation between the relative power of the harmonic and the radius but don't get any information about the θ and ϕ spherical coordinates. This information is probably hidden in the individual power of each order m of the degree l. This intuition comes from the figure B.1 where in the higher degree l we see that the order m are oriented. Intuitively, the order should be able to indicate a direction where the signal is more powerful.
- Combination of the degree power: Here we combined the radius reconstructed by the different degree via a simple mean but we shown in section 4.3.2 and annex A that this is note the optimal way to combine estimator. A more refined algorithm probably exist to take into account the predicting power of each order.

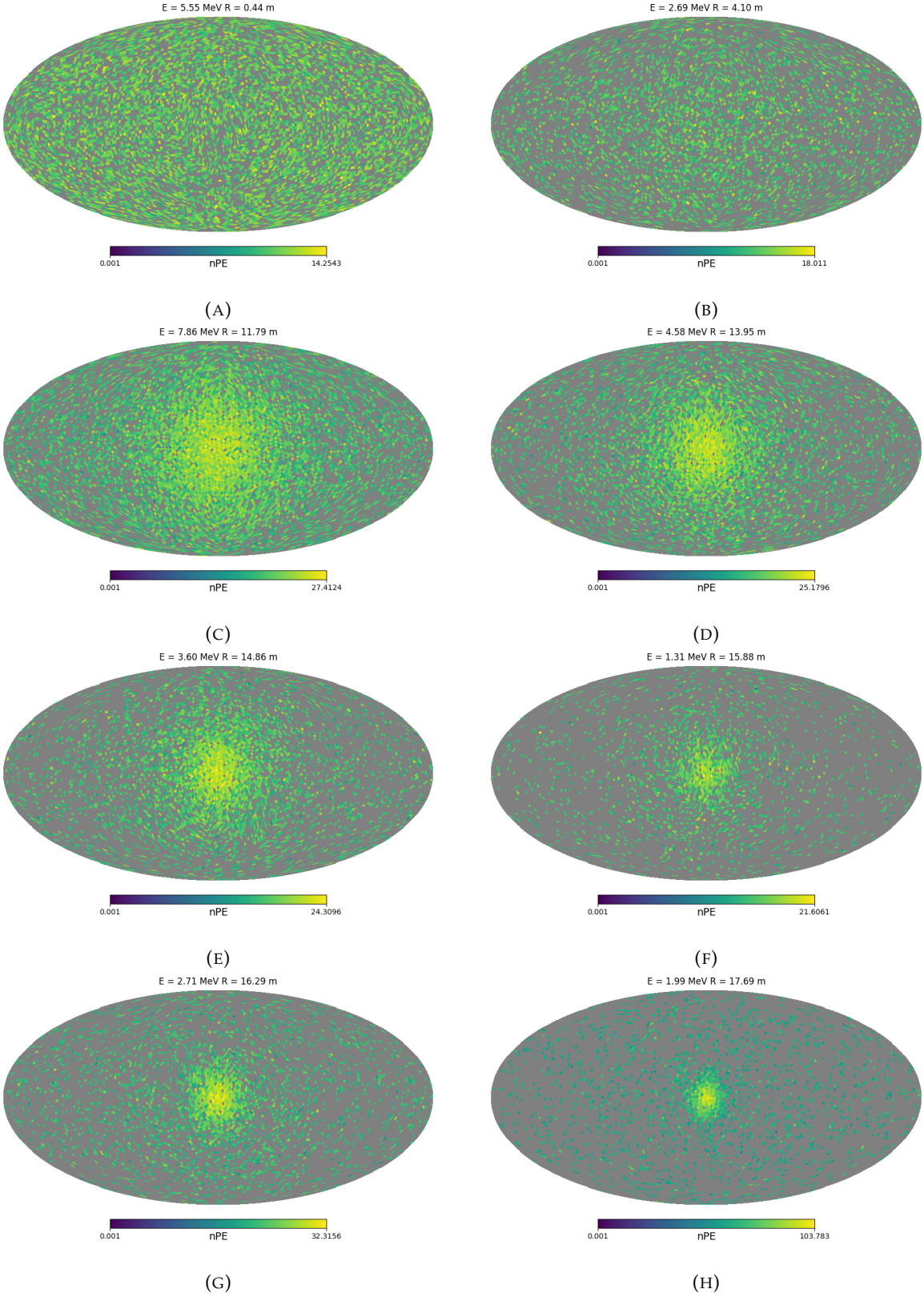


FIGURE B.4 – Charge repartition in JUNO as seen by the Healpix segmentation. Those are Healpix map of order 5 (i.e. 12288 pixels). The color represent the summed charge of the PMTs in each pixels. The color scale is logarithmic. The view have been centered to prevent event deformations.

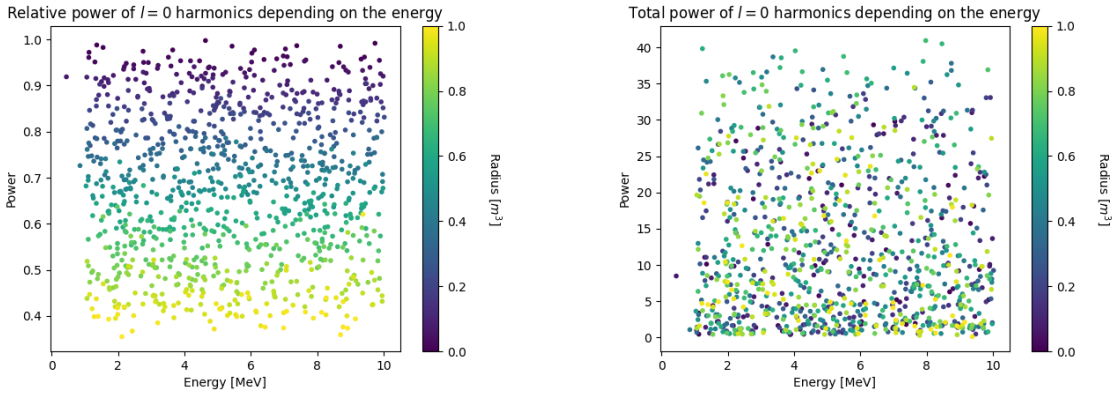


FIGURE B.5 – Scatter plot of the absolute and relative power, respectively on the left and right plot, of the $l = 0$ harmonic. The color indicate the radius of the event.

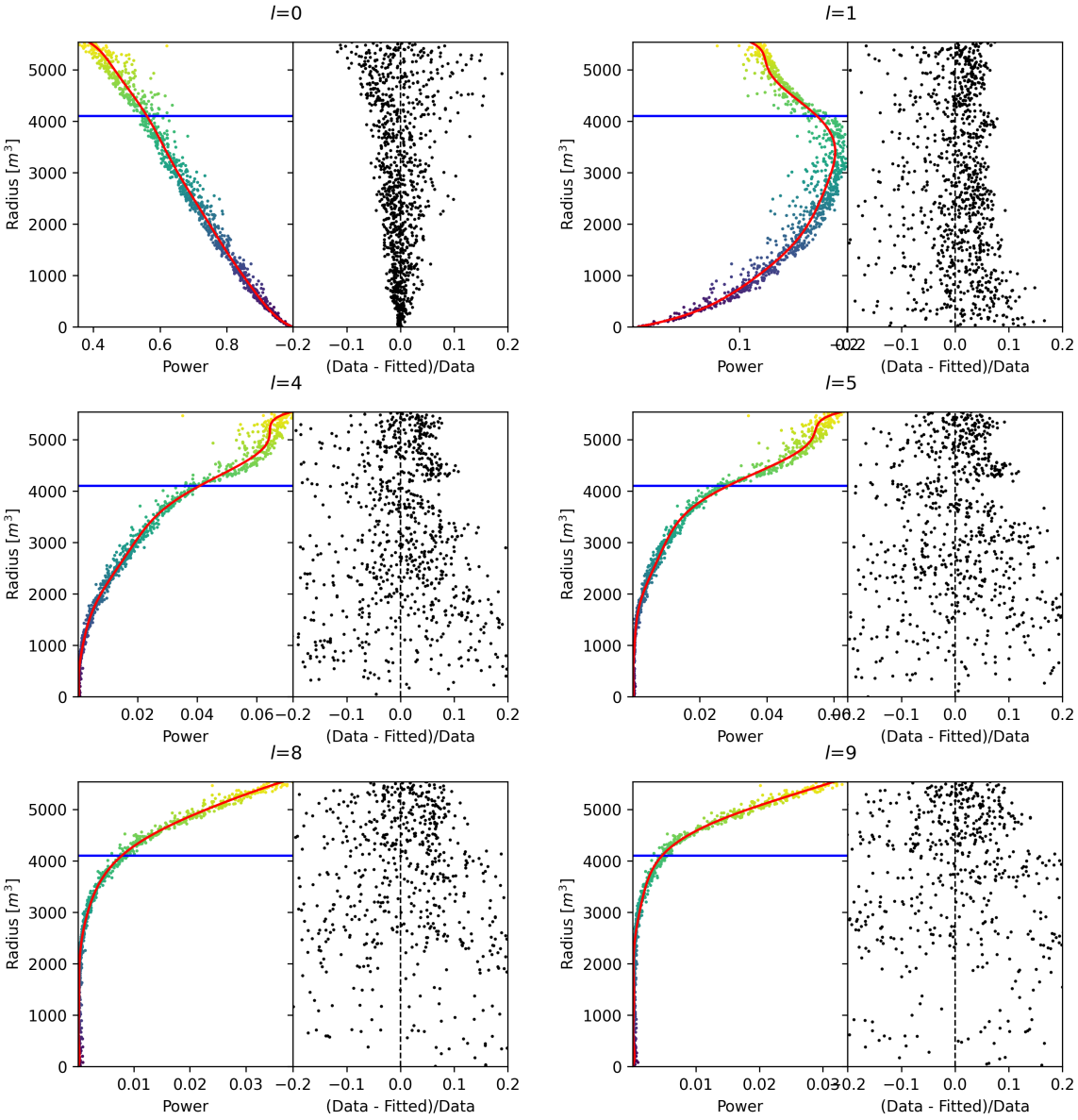


FIGURE B.6 – Plot of the distribution of the relative power of each harmonic dependent on R^3 (on the left). The Total Reflection (TR) area is represented by the horizontal blue line. The distribution are fitted using a 9th degree polynomial (red curve). The relative power error between the distribution and the fit is represented on the left. **Part 1**

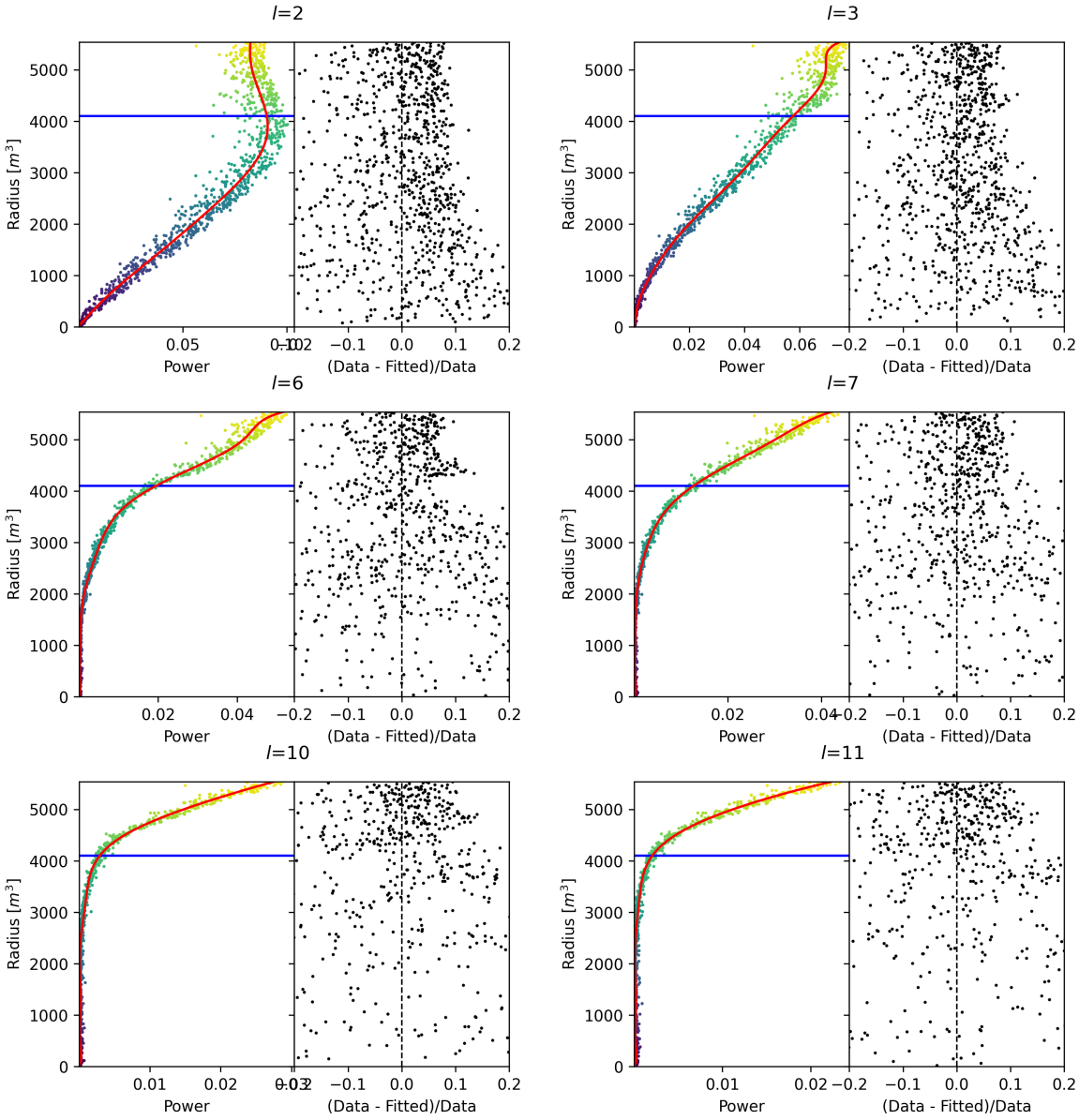


FIGURE B.7 – Plot of the distribution of the relative power of each harmonic dependent on R^3 (on the left). The Total Reflection (TR) area is represented by the horizontal blue line. The distribution are fitted using a 9th degree polynomial (red curve). The relative power error between the distribution and the fit is represented on the left. **Part 2**

Appendix C

Correction of E_{vis} bias

The reconstruction algorithms that are presented in this thesis in Chapters 4 and 5 do not reconstruct the same energy as the classical algorithms presented in section 3.3. Our algorithms reconstruct the *deposited* energy E_{dep} while the classical algorithms reconstruct a *visible* energy E_{vis}.

To understand this phenomena, let's look at the equation 3.27:

$$\hat{\mu}(r,\theta,\theta_{pmt},E_{vis}) = \frac{1}{E_{vis}}\frac{1}{M}\sum_{i}^{M}\frac{\frac{\bar{q}_i}{\bar{Q}_i}-\mu_i^D}{\mathrm{DE}_i},\ \mu_i^D = \mathrm{DNR}_i \cdot L$$

which define the expected N_{pe}/E. This define a linear relation between the number of photoelectrons and the energy. However we discussed in sections 2.3.2 and 2.4 that the number of photoelectrons collected by the LPMT system do not follow a linear relationship. Thus this visible energy is not linear with the deposited energy. This effect is corrected in physics analysis and in Chapter 7 by applying the calibrated non-linearity profile the energy spectrum.

When we need to compare our algorithm that reconstruct the deposited energy to the classical algorithms we need to correct this non-linearity. For this we fit the systematic bias of the classical algorithm using a 5th degree polynomial

$$\frac{E_{dep}}{E_{evis}} = \sum_{i=0}^{5} P_i E_{dep}^i \tag{C.1}$$

The fitted distribution and the corresponding fit is presented in figure C.1. The value fitted for this correction are presented in table C.1.

P_0	1.24541 +/- 0.00585121
P_1	-0.168079 +/- 0.00716387
P_2	0.0489947 +/- 0.00312875
P_3	-0.00747111 +/- 0.000622003
P_4	0.000570998 +/- 5.7296e-05
P_5	-1.72588e-05 +/- 1.98355e-06

TABLE C.1 – Parameters of the 5th degree polynomial used to correct Omilrec reconstructed energy.

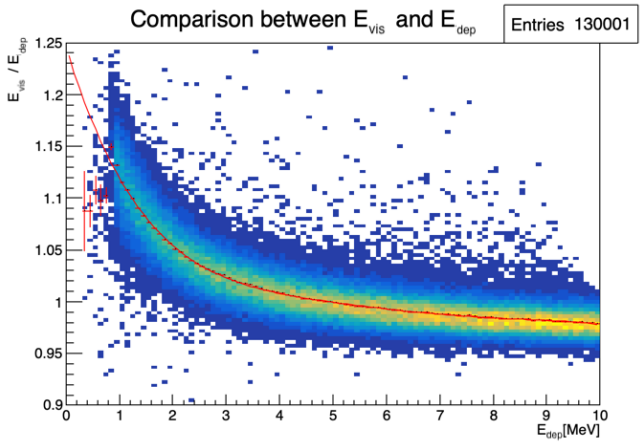


FIGURE C.1 – Comparison between Omilrec reconstructed E_{vis} and the deposited energy E_{dep}. The profile of the distribution E_{vis}/E_{dep} vs E_{dep} is fitted with a 5th degree polynomial.

List of Tables

List of Figures

List of Abbreviations

ACU	**A**utomatic **C**alibration **U**nit
ANN	**A**dversarial **N**eural **N**etwork
BDT	**B**oosted **D**ecision **T**ree
BFP	**B**est **F**it **P**oint
CD	**C**entral **D**etector
CLS	**C**able **L**oop **S**ystem
CNN	**C**onvolutional **NN**
DNN	**D**eep **NN**
DN	**D**ark **N**oise
EDM	**E**vent **D**ata **M**odel
FCDNN	**F**ully **C**onnected **D**eep **NN**
GNN	**G**raph **NN**
GT	**G**uiding **T**ube
IBD	**I**nverse **B**eta **D**ecay
IO	**I**nverse **O**rdering
JUNO	**J**iangmen **U**nderground **N**eutrino **O**bservatory
LPMT	**L**arge **PMT**
LR	**L**earning **R**ate
LS	**L**iquid **S**cintillator
MC	**M**onte **C**arlo simulation
ML	**M**achine **L**earning
MSE	**M**ean **S**quared **E**rror
NMO	**N**eutrino **M**ass **O**rdering
NN	**N**eural **N**etwork
NO	**N**ormal **O**rdering
NPE	**N**umber of **P**hoto **E**lectron
OSIRIS	**O**nline **S**cintillator **I**nternal **R**adioactivity **I**nvestigation **S**ystem
PE	**P**hoto **E**lectron
PMT	**P**hoto-**M**ultipliers **T**ubes
PReLU	**P**arametrized **Re**ctified **L**inear **U**nit
QNL	Charge (**Q**) **N**on **L**inearity
ROV	**R**emotely **O**perated under-LS **V**ehicle
ReLU	**Re**ctified **L**inear **U**nit
ResNet	**Res**idual **Net**work
SGD	**S**tochastic **G**radient **D**escent
SPMT	**S**mall **PMT**
TAO	**T**aishan **A**ntineutrino **O**servatory
TR Area	**T**otal **R**eflexion **Area**
TTS	**T**ime **T**ransit **S**pread
TT	**T**op **T**racker
UWB	**U**nder **W**ater **B**oxes
WCD	**W**ater **C**herenkov **D**etector

Bibliography

[1] Liang Zhan, Yifang Wang, Jun Cao, and Liangjian Wen. "Determination of the Neutrino Mass Hierarchy at an Intermediate Baseline". *Physical Review D* 78.11 (Dec. 10, 2008), 111103. ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.78.111103. eprint: 0807.3203[hep-ex, physics:hep-ph]. URL: http://arxiv.org/abs/0807.3203 (visited on 09/18/2023).

[2] Fengpeng An et al. "Neutrino Physics with JUNO". *Journal of Physics G: Nuclear and Particle Physics* 43.3 (Mar. 1, 2016), 030401. ISSN: 0954-3899, 1361-6471. DOI: 10.1088/0954-3899/43/3/030401. eprint: 1507.05613[hep-ex,physics:physics]. URL: http://arxiv.org/abs/1507.05613 (visited on 07/28/2023).

[3] JUNO Collaboration et al. "Sub-percent Precision Measurement of Neutrino Oscillation Parameters with JUNO". *Chinese Physics C* 46.12 (Dec. 1, 2022), 123001. ISSN: 1674-1137, 2058-6132. DOI: 10.1088/1674-1137/ac8bc9. eprint: 2204.13249[hep-ex]. URL: http://arxiv.org/abs/2204.13249 (visited on 08/11/2023).

[4] A. A. Hahn, K. Schreckenbach, W. Gelletly, F. von Feilitzsch, G. Colvin, and B. Krusche. "Antineutrino spectra from 241Pu and 239Pu thermal neutron fission products". *Physics Letters B* 218.3 (Feb. 23, 1989), 365–368. ISSN: 0370-2693. DOI: 10.1016/0370-2693(89)91598-0. URL: https://www.sciencedirect.com/science/article/pii/0370269389915980 (visited on 01/16/2024).

[5] Th A. Mueller et al. "Improved Predictions of Reactor Antineutrino Spectra". *Physical Review C* 83.5 (May 23, 2011), 054615. ISSN: 0556-2813, 1089-490X. DOI: 10.1103/PhysRevC.83.054615. eprint: 1101.2663[hep-ex,physics:nucl-ex]. URL: http://arxiv.org/abs/1101.2663 (visited on 01/16/2024).

[6] F. von Feilitzsch, A. A. Hahn, and K. Schreckenbach. "Experimental beta-spectra from 239Pu and 235U thermal neutron fission products and their correlated antineutrino spectra". *Physics Letters B* 118.1 (Dec. 2, 1982), 162–166. ISSN: 0370-2693. DOI: 10.1016/0370-2693(82)90622-0. URL: https://www.sciencedirect.com/science/article/pii/0370269382906220 (visited on 01/16/2024).

[7] K. Schreckenbach, G. Colvin, W. Gelletly, and F. Von Feilitzsch. "Determination of the antineutrino spectrum from 235U thermal neutron fission products up to 9.5 MeV". *Physics Letters B* 160.4 (Oct. 10, 1985), 325–330. ISSN: 0370-2693. DOI: 10.1016/0370-2693(85)91337-1. URL: https://www.sciencedirect.com/science/article/pii/0370269385913371 (visited on 01/16/2024).

[8] Patrick Huber. "On the determination of anti-neutrino spectra from nuclear reactors". *Physical Review C* 84.2 (Aug. 29, 2011), 024617. ISSN: 0556-2813, 1089-490X. DOI: 10.1103/PhysRevC.84.024617. eprint: 1106.0687[hep-ex,physics:hep-ph,physics:nucl-ex,physics:nucl-th]. URL: http://arxiv.org/abs/1106.0687 (visited on 01/16/2024).

[9] P. Vogel, G. K. Schenter, F. M. Mann, and R. E. Schenter. "Reactor antineutrino spectra and their application to antineutrino-induced reactions. II". *Physical Review C* 24.4 (Oct. 1, 1981). Publisher: American Physical Society, 1543–1553. DOI: 10.1103/PhysRevC.24.1543. URL: https://link.aps.org/doi/10.1103/PhysRevC.24.1543 (visited on 01/16/2024).

[10] D. A. Dwyer and T. J. Langford. "Spectral Structure of Electron Antineutrinos from Nuclear Reactors". *Physical Review Letters* 114.1 (Jan. 7, 2015), 012502. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.114.012502. eprint: 1407.1281[hep-ex,physics:nucl-ex]. URL: http://arxiv.org/abs/1407.1281 (visited on 01/16/2024).

[11] Daya Bay Collaboration et al. "Measurement of the Reactor Antineutrino Flux and Spectrum at Daya Bay". *Physical Review Letters* 116.6 (Feb. 12, 2016). Publisher: American Physical Society, 061801. DOI: 10.1103/PhysRevLett.116.061801. URL: https://link.aps.org/doi/10.1103/PhysRevLett.116.061801 (visited on 09/06/2024).

[12] G. Mention, M. Fechner, Th. Lasserre, Th. A. Mueller, D. Lhuillier, M. Cribier, and A. Letourneau. "Reactor antineutrino anomaly". *Physical Review D* 83.7 (Apr. 29, 2011). Publisher: American Physical Society, 073006. DOI: 10.1103/PhysRevD.83.073006. URL: https://link.aps.org/doi/10.1103/PhysRevD.83.073006 (visited on 03/05/2024).

[13] JUNO Collaboration et al. *TAO Conceptual Design Report: A Precision Measurement of the Reactor Antineutrino Spectrum with Sub-percent Energy Resolution*. May 18, 2020. DOI: 10.48550/arXiv.2005.08745. eprint: 2005.08745[hep-ex,physics:nucl-ex,physics:physics]. URL: http://arxiv.org/abs/2005.08745 (visited on 01/18/2024).

[14] Super-Kamiokande Collaboration et al. "Diffuse Supernova Neutrino Background Search at Super-Kamiokande". *Physical Review D* 104.12 (Dec. 10, 2021), 122002. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.104.122002. eprint: 2109.11174[astro-ph,physics:hep-ex]. URL: http://arxiv.org/abs/2109.11174 (visited on 02/28/2024).

[15] JUNO Collaboration et al. "JUNO Sensitivity on Proton Decay $p \to \bar{\nu}K^+$ Searches". *Chinese Physics C* 47.11 (Nov. 1, 2023), 113002. ISSN: 1674-1137, 2058-6132. DOI: 10.1088/1674-1137/ace9c6. eprint: 2212.08502[hep-ex,physics:hep-ph]. URL: http://arxiv.org/abs/2212.08502 (visited on 08/09/2024).

[16] Alessandro Strumia and Francesco Vissani. "Precise quasielastic neutrino/nucleon cross section". *Physics Letters B* 564.1 (July 2003), 42–54. ISSN: 03702693. DOI: 10.1016/S0370-2693(03)00616-6. eprint: astro-ph/0302055. URL: http://arxiv.org/abs/astro-ph/0302055 (visited on 01/16/2024).

[17] Daya Bay et al. *Optimization of the JUNO liquid scintillator composition using a Daya Bay antineutrino detector*. July 1, 2020. DOI: 10.48550/arXiv.2007.00314. eprint: 2007.00314[hep-ex,physics:physics]. URL: http://arxiv.org/abs/2007.00314 (visited on 07/26/2023).

[18] J. B. Birks. "CHAPTER 3 - THE SCINTILLATION PROCESS IN ORGANIC MATERIALS—I". *The Theory and Practice of Scintillation Counting*. Ed. by J. B. Birks. International Series of Monographs in Electronics and Instrumentation. Jan. 1, 1964, 39–67. ISBN: 978-0-08-010472-0. DOI: 10.1016/B978-0-08-010472-0.50008-2. URL: https://www.sciencedirect.com/science/article/pii/B9780080104720500082 (visited on 02/07/2024).

[19] *Photomultiplier tube R12860 | Hamamatsu Photonics*. URL: https://www.hamamatsu.com/eu/en/product/optical-sensors/pmt/pmt_tube-alone/head-on-type/R12860.html (visited on 02/08/2024).

[20] Yan Zhang, Ze-Yuan Yu, Xin-Ying Li, Zi-Yan Deng, and Liang-Jian Wen. "A complete optical model for liquid-scintillator detectors". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 967 (July 2020), 163860. ISSN: 01689002. DOI: 10.1016/j.nima.2020.163860. eprint: 2003.12212[physics]. URL: http://arxiv.org/abs/2003.12212 (visited on 02/07/2024).

[21] Hai-Bo Yang et al. "Light Attenuation Length of High Quality Linear Alkyl Benzene as Liquid Scintillator Solvent for the JUNO Experiment". *Journal of Instrumentation* 12.11 (Nov. 27, 2017), T11004–T11004. ISSN: 1748-0221. DOI: 10.1088/1748-0221/12/11/T11004. eprint: 1703.01867[hep-ex,physics:physics]. URL: http://arxiv.org/abs/1703.01867 (visited on 07/28/2023).

[22] JUNO Collaboration et al. *The Design and Sensitivity of JUNO's scintillator radiopurity pre-detector OSIRIS*. Mar. 31, 2021. DOI: 10.48550/arXiv.2103.16900. eprint: 2103.16900[physics]. URL: http://arxiv.org/abs/2103.16900 (visited on 02/07/2024).

[23] Angel Abusleme et al. "Mass Testing and Characterization of 20-inch PMTs for JUNO". *The European Physical Journal C* 82.12 (Dec. 24, 2022), 1168. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-022-11002-8. eprint: 2205.08629[hep-ex,physics:physics]. URL: http://arxiv.org/abs/2205.08629 (visited on 02/08/2024).

[24] Yang Han. "Dual Calorimetry for High Precision Neutrino Oscillation Measurement at JUNO Experiment". AstroParticule et Cosmologie, France, Paris U. VII, APC, June 2021.

[25] R. Acquafredda et al. "The OPERA experiment in the CERN to Gran Sasso neutrino beam". *Journal of Instrumentation* 4.4 (Apr. 2009), P04018. ISSN: 1748-0221. DOI: 10.1088/1748-0221/4/04/P04018. URL: https://dx.doi.org/10.1088/1748-0221/4/04/P04018 (visited on 02/29/2024).

[26] JUNO collaboration et al. "Calibration Strategy of the JUNO Experiment". *Journal of High Energy Physics* 2021.3 (Mar. 2021), 4. ISSN: 1029-8479. DOI: 10.1007/JHEP03(2021)004. eprint: 2011.06405[hep-ex,physics:physics]. URL: http://arxiv.org/abs/2011.06405 (visited on 08/10/2023).

[27] Hans Th J. Steiger. *TAO – The Taishan Antineutrino Observatory*. Sept. 21, 2022. DOI: 10.48550/arXiv.2209.10387. eprint: 2209.10387[physics]. URL: http://arxiv.org/abs/2209.10387 (visited on 01/16/2024).

[28] Tao Lin et al. "The Application of SNiPER to the JUNO Simulation". *Journal of Physics: Conference Series* 898.4 (Oct. 2017). Publisher: IOP Publishing, 042029. ISSN: 1742-6596. DOI: 10.1088/1742-6596/898/4/042029. URL: https://dx.doi.org/10.1088/1742-6596/898/4/042029 (visited on 02/27/2024).

[29] S. Agostinelli et al. "Geant4—a simulation toolkit". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (July 1, 2003), 250–303. ISSN: 0168-9002. DOI: 10.1016/S0168-9002(03)01368-8. URL: https://www.sciencedirect.com/science/article/pii/S0168900203013688 (visited on 02/27/2024).

[30] J. Allison et al. "Geant4 developments and applications". *IEEE Transactions on Nuclear Science* 53.1 (Feb. 2006). Conference Name: IEEE Transactions on Nuclear Science, 270–278. ISSN: 1558-1578. DOI: 10.1109/TNS.2006.869826. URL: https://ieeexplore.ieee.org/document/1610988?isnumber=33833&arnumber=1610988&count=33&index=7 (visited on 02/27/2024).

[31] J. Allison et al. "Recent developments in Geant4". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 835 (Nov. 1, 2016), 186–225. ISSN: 0168-9002. DOI: 10.1016/j.nima.2016.06.125. URL: https://www.sciencedirect.com/science/article/pii/S0168900216306957 (visited on 02/27/2024).

[32] Angel Abusleme et al. "Potential to Identify the Neutrino Mass Ordering with Reactor Antineutrinos in JUNO" (May 2024). eprint: 2405.18008.

[33] Xiangpan Ji, Wenqiang Gu, Xin Qian, Hanyu Wei, and Chao Zhang. *Combined Neyman-Pearson Chi-square: An Improved Approximation to the Poisson-likelihood Chi-square*. arXiv.org. Mar. 17, 2019. URL: https://arxiv.org/abs/1903.07185v3 (visited on 10/03/2024).

[34] Particle Data Group et al. "Review of Particle Physics". *Progress of Theoretical and Experimental Physics* 2020.8 (Aug. 14, 2020), 083C01. ISSN: 2050-3911. DOI: 10.1093/ptep/ptaa104. URL: https://doi.org/10.1093/ptep/ptaa104 (visited on 12/04/2023).

[35] JUNO Collaboration et al. "JUNO Physics and Detector". *Progress in Particle and Nuclear Physics* 123 (Mar. 2022), 103927. ISSN: 01466410. DOI: 10.1016/j.ppnp.2021.103927. eprint: 2104.02565[hep-ex]. URL: http://arxiv.org/abs/2104.02565 (visited on 09/18/2023).

[36] Leo Breiman, Jerome Friedman, R. A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. New York: Chapman and Hall/CRC, Oct. 25, 2017. 368 pp. ISBN: 978-1-315-13947-0. DOI: 10.1201/9781315139470.

[37] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." *The Annals of Statistics* 29.5 (Oct. 2001). Publisher: Institute of Mathematical Statistics, 1189–1232. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/aos/1013203451. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full (visited on 04/29/2024).

[38] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Jan. 29, 2017. DOI: 10.48550/arXiv.1412.6980. eprint: 1412.6980[cs]. URL: http://arxiv.org/abs/1412.6980 (visited on 05/13/2024).

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016

IEEE Conference on Computer Vision and Pattern Recognition (CVPR). ISSN: 1063-6919. June 2016, 770–778. DOI: 10.1109/CVPR.2016.90. URL: https://ieeexplore.ieee.org/document/7780459 (visited on 07/17/2024).

[40] Olga Russakovsky et al. *ImageNet Large Scale Visual Recognition Challenge*. Jan. 29, 2015. DOI: 10.48550/arXiv.1409.0575. eprint: 1409.0575[cs]. URL: http://arxiv.org/abs/1409.0575 (visited on 05/17/2024).

[41] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Apr. 10, 2015. DOI: 10.48550/arXiv.1409.1556. eprint: 1409.1556[cs]. URL: http://arxiv.org/abs/1409.1556 (visited on 05/17/2024).

[42] Anna Allen. *generic-github-user/Image-Convolution-Playground*. original-date: 2018-09-28T22:42:55Z. July 15, 2024. URL: https://github.com/generic-github-user/Image-Convolution-Playground (visited on 07/16/2024).

[43] Jason Ansel et al. *PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation*. Publication Title: 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24) original-date: 2016-08-13T05:26:41Z. Apr. 2024. DOI: 10.1145/3620665.3640366. URL: https://pytorch.org/assets/pytorch2-2.pdf (visited on 07/16/2024).

[44] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86.11 (Nov. 1998). Conference Name: Proceedings of the IEEE, 2278–2324. ISSN: 1558-2256. DOI: 10.1109/5.726791. URL: https://ieeexplore.ieee.org/document/726791 (visited on 07/16/2024).

[45] *NVIDIA T4 Tensor Core GPUs for Accelerating Inference*. NVIDIA. URL: https://www.nvidia.com/en-gb/data-center/tesla-t4/ (visited on 07/16/2024).

[46] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. *Neural Message Passing for Quantum Chemistry*. June 12, 2017. DOI: 10.48550/arXiv.1704.01212. eprint: 1704.01212[cs]. URL: http://arxiv.org/abs/1704.01212 (visited on 05/22/2024).

[47] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. *Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering*. Feb. 5, 2017. DOI: 10.48550/arXiv.1606.09375. eprint: 1606.09375[cs, stat]. URL: http://arxiv.org/abs/1606.09375 (visited on 04/04/2024).

[48] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. *Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting*. Feb. 22, 2018. DOI: 10.48550/arXiv.1707.01926. eprint: 1707.01926[cs, stat]. URL: http://arxiv.org/abs/1707.01926 (visited on 05/22/2024).

[49] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*. June 10, 2014. DOI: 10.48550/arXiv.1406.2661. eprint: 1406.2661[cs, stat]. URL: http://arxiv.org/abs/1406.2661 (visited on 05/29/2024).

[50] Wenjie Wu, Miao He, Xiang Zhou, and Haoxue Qiao. "A new method of energy reconstruction for large spherical liquid scintillator detectors". *Journal of Instrumentation* 14.3 (Mar. 8, 2019), P03009–P03009. ISSN: 1748-0221. DOI: 10.1088/1748-0221/14/03/P03009. eprint: 1812.01799[hep-ex, physics:physics]. URL: http://arxiv.org/abs/1812.01799 (visited on 07/28/2023).

[51] Guihong Huang et al. "Improving the energy uniformity for large liquid scintillator detectors". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1001 (June 11, 2021), 165287. ISSN: 0168-9002. DOI: 10.1016/j.nima.2021.165287. URL: https://www.sciencedirect.com/science/article/pii/S0168900221002710 (visited on 03/01/2024).

[52] Ziyuan Li et al. "Event vertex and time reconstruction in large volume liquid scintillator detector". *Nuclear Science and Techniques* 32.5 (May 2021), 49. ISSN: 1001-8042, 2210-3147. DOI: 10.1007/s41365-021-00885-z. eprint: 2101.08901[hep-ex, physics:physics]. URL: http://arxiv.org/abs/2101.08901 (visited on 07/28/2023).

[53] Gioacchino Ranucci. "An analytical approach to the evaluation of the pulse shape discrimination properties of scintillators". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 354.2 (Jan. 30, 1995), 389–399. ISSN: 0168-9002. DOI: 10.1016/0168-9002(94)00886-8. URL: https://www.sciencedirect.com/science/article/pii/0168900294008868 (visited on 03/07/2024).

[54] C. Galbiati and K. McCarty. "Time and space reconstruction in optical, non-imaging, scintillator-based particle detectors". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 568.2 (Dec. 1, 2006), 700–709. ISSN: 0168-9002. DOI: 10.1016/j.nima.2006.07.058. URL: https://www.sciencedirect.com/science/article/pii/S0168900206013519 (visited on 03/07/2024).

[55] M. Moszyński and B. Bengtson. "Status of timing with plastic scintillation detectors". *Nuclear Instruments and Methods* 158 (Jan. 1, 1979), 1–31. ISSN: 0029-554X. DOI: 10.1016/S0029-554X(79)90170-8. URL: https://www.sciencedirect.com/science/article/pii/S0029554X79901708 (visited on 03/07/2024).

[56] Gui-Hong Huang, Wei Jiang, Liang-Jian Wen, Yi-Fang Wang, and Wu-Ming Luo. "Data-driven simultaneous vertex and energy reconstruction for large liquid scintillator detectors". *Nuclear Science and Techniques* 34.6 (June 17, 2023), 83. ISSN: 2210-3147. DOI: 10.1007/s41365-023-01240-0. URL: https://doi.org/10.1007/s41365-023-01240-0 (visited on 08/17/2023).

[57] Zhen Qian et al. "Vertex and Energy Reconstruction in JUNO with Machine Learning Methods". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1010 (Sept. 2021), 165527. ISSN: 01689002. DOI: 10.1016/j.nima.2021.165527. eprint: 2101.04839[hep-ex,physics:physics]. URL: http://arxiv.org/abs/2101.04839 (visited on 07/24/2023).

[58] Arsenii Gavrikov, Yury Malyshkin, and Fedor Ratnikov. "Energy reconstruction for large liquid scintillator detectors with machine learning techniques: aggregated features approach". *The European Physical Journal C* 82.11 (Nov. 14, 2022), 1021. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-022-11004-6. eprint: 2206.09040[physics]. URL: http://arxiv.org/abs/2206.09040 (visited on 07/24/2023).

[59] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann. "HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere". *The Astrophysical Journal* 622 (Apr. 1, 2005). ADS Bibcode: 2005ApJ...622..759G, 759–771. ISSN: 0004-637X. DOI: 10.1086/427976. URL: https://ui.adsabs.harvard.edu/abs/2005ApJ...622..759G (visited on 04/04/2024).

[60] Anatael Cabrera et al. *Multi-Calorimetry in Light-based Neutrino Detectors*. Dec. 20, 2023. DOI: 10.48550/arXiv.2312.12991. eprint: 2312.12991[hep-ex,physics:physics]. URL: http://arxiv.org/abs/2312.12991 (visited on 08/19/2024).

[61] Victor Lebrin. "Towards the Detection of Core-Collapse Supernovae Burst Neutrinos with the 3-inch PMT System of the JUNO Detector". These de doctorat. Nantes Université, Sept. 5, 2022. URL: https://theses.fr/2022NANU4080 (visited on 05/22/2024).

[62] Dan Cireşan, Ueli Meier, and Juergen Schmidhuber. *Multi-column Deep Neural Networks for Image Classification*. version: 1. Feb. 13, 2012. DOI: 10.48550/arXiv.1202.2745. eprint: 1202.2745[cs]. URL: http://arxiv.org/abs/1202.2745 (visited on 06/27/2024).

[63] R. Abbasi et al. "A Convolutional Neural Network based Cascade Reconstruction for the IceCube Neutrino Observatory". *Journal of Instrumentation* 16.7 (July 1, 2021), P07041. ISSN: 1748-0221. DOI: 10.1088/1748-0221/16/07/P07041. eprint: 2101.11589[hep-ex]. URL: http://arxiv.org/abs/2101.11589 (visited on 06/27/2024).

[64] D. Maksimović, M. Nieslony, and M. Wurm. "CNNs for enhanced background discrimination in DSNB searches in large-scale water-Gd detectors". *Journal of Cosmology and Astroparticle Physics* 2021.11 (Nov. 2021). Publisher: IOP Publishing, 051. ISSN: 1475-7516. DOI: 10.1088/1475-7516/2021/11/051. URL: https://dx.doi.org/10.1088/1475-7516/2021/11/051 (visited on 06/27/2024).

[65] Taco S. Cohen, Mario Geiger, Jonas Koehler, and Max Welling. *Spherical CNNs*. Feb. 25, 2018. DOI: 10.48550/arXiv.1801.10130. eprint: 1801.10130[cs,stat]. URL: http://arxiv.org/abs/1801.10130 (visited on 07/13/2024).

[66] *NVIDIA A100 GPUs Power the Modern Data Center*. NVIDIA. URL: https://www.nvidia.com/en-gb/data-center/a100/ (visited on 08/06/2024).

[67] *NVIDIA V100*. NVIDIA. URL: https://www.nvidia.com/en-gb/data-center/v100/ (visited on 08/06/2024).

[68] Leonard Imbert. *leonard-IMBERT/datamo*. original-date: 2023-10-17T12:37:38Z. Aug. 9, 2024. URL: https://github.com/leonard-IMBERT/datamo (visited on 08/09/2024).

[69] "IEEE Standard for Floating-Point Arithmetic". *IEEE Std 754-2019 (Revision of IEEE 754-2008)* (July 2019). Conference Name: IEEE Std 754-2019 (Revision of IEEE 754-2008), 1–84. DOI: 10.1109/IEEESTD.2019.8766229. URL: https://ieeexplore.ieee.org/document/8766229 (visited on 07/03/2024).

[70] Chuanya Cao et al. "Mass production and characterization of 3-inch PMTs for the JUNO experiment". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1005 (July 2021), 165347. ISSN: 01689002. DOI: 10.1016/j.nima.2021.165347. eprint: 2102.11538[hep-ex,physics:physics]. URL: http://arxiv.org/abs/2102.11538 (visited on 02/08/2024).

[71] K. M. Gorski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelman. "HEALPix – a Framework for High Resolution Discretization, and Fast Analysis of Data Distributed on the Sphere". *The Astrophysical Journal* 622.2 (Apr. 2005), 759–771. ISSN: 0004-637X, 1538-4357. DOI: 10.1086/427976. eprint: astro-ph/0409513. URL: http://arxiv.org/abs/astro-ph/0409513 (visited on 08/10/2023).

[72] Teng Li, Xin Xia, Xing-Tao Huang, Jia-Heng Zou, Wei-Dong Li, Tao Lin, Kun Zhang, and Zi-Yan Deng. "Design and development of JUNO event data model*". *Chinese Physics C* 41.6 (June 2017). Publisher: IOP Publishing, 066201. ISSN: 1674-1137. DOI: 10.1088/1674-1137/41/6/066201. URL: https://dx.doi.org/10.1088/1674-1137/41/6/066201 (visited on 08/16/2024).

[73] Martin Reinecke. *Ducc0*. original-date: 2021-04-12T15:35:50Z. Aug. 9, 2024. URL: https://gitlab.mpcdf.mpg.de/mtr/ducc (visited on 08/16/2024).

[74] Mario Schwarz, Sabrina M. Franke, Lothar Oberauer, Miriam D. Plein, Hans Th J. Steiger, and Marc Tippmann. *Measurements of the Lifetime of Orthopositronium in the LAB-Based Liquid Scintillator of JUNO*. Apr. 25, 2018. DOI: 10.1016/j.nima.2018.12.068. eprint: 1804.09456[physics]. URL: http://arxiv.org/abs/1804.09456 (visited on 09/17/2024).

[75] Narongkiat Rodphai, Zhimin Wang, Narumon Suwonjandee, and Burin Asavapibhop. "20-inch photomultiplier tube timing study for JUNO". *Journal of Physics: Conference Series* 2145.1 (Dec. 2021). Publisher: IOP Publishing, 012017. ISSN: 1742-6596. DOI: 10.1088/1742-6596/2145/1/012017. URL: https://dx.doi.org/10.1088/1742-6596/2145/1/012017 (visited on 09/17/2024).

[76] Dong-Hao Liao et al. "Study of TTS for a 20-inch dynode PMT*". *Chinese Physics C* 41.7 (July 2017). Publisher: IOP Publishing, 076001. ISSN: 1674-1137. DOI: 10.1088/1674-1137/41/7/076001. URL: https://dx.doi.org/10.1088/1674-1137/41/7/076001 (visited on 09/17/2024).

[77] Nan Li et al. "Characterization of 3-inch photomultiplier tubes for the JUNO central detector". *Radiation Detection Technology and Methods* 3.1 (Nov. 22, 2018), 6. ISSN: 2509-9949. DOI: 10.1007/s41605-018-0085-8. URL: https://doi.org/10.1007/s41605-018-0085-8 (visited on 09/17/2024).

[78] B. Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez. *Deep Reinforcement Learning for Autonomous Driving: A Survey*. Jan. 23, 2021. eprint: 2002.00444[cs]. URL: http://arxiv.org/abs/2002.00444 (visited on 10/02/2024).

[79] Oriol Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". 575.7782 (Nov. 2019). Publisher: Nature Publishing Group, 350–354. ISSN: 1476-4687. DOI:

10.1038/s41586-019-1724-z. URL: https://www.nature.com/articles/s41586-019-1724-z (visited on 10/02/2024).

[80] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. arXiv.org. Feb. 6, 2015. URL: https://arxiv.org/abs/1502.01852v1 (visited on 10/08/2024).

[81] Daya Bay Collaboration et al. *A high precision calibration of the nonlinear energy response at Daya Bay*. arXiv.org. Feb. 21, 2019. URL: https://arxiv.org/abs/1902.08241v2 (visited on 10/01/2024).

[82] Double Chooz Collaboration et al. "The Double Chooz antineutrino detectors". *The European Physical Journal C* 82.9 (Sept. 8, 2022), 804. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-022-10726-x. eprint: 2201.13285[physics]. URL: http://arxiv.org/abs/2201.13285 (visited on 10/07/2024).

[83] Th. A. Mueller et al. "Improved predictions of reactor antineutrino spectra". *Physical Review C* 83.5 (May 23, 2011). Publisher: American Physical Society, 054615. DOI: 10.1103/PhysRevC.83.054615. URL: https://link.aps.org/doi/10.1103/PhysRevC.83.054615 (visited on 09/06/2024).

[84] X. B. Ma, W. L. Zhong, L. Z. Wang, Y. X. Chen, and J. Cao. "Improved calculation of the energy release in neutron-induced fission". *Physical Review C* 88.1 (July 12, 2013). Publisher: American Physical Society, 014605. DOI: 10.1103/PhysRevC.88.014605. URL: https://link.aps.org/doi/10.1103/PhysRevC.88.014605 (visited on 09/06/2024).

[85] Pauli Virtanen et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". *Nature Methods* 17.3 (Mar. 2020). Publisher: Nature Publishing Group, 261–272. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0686-2. URL: https://www.nature.com/articles/s41592-019-0686-2 (visited on 08/14/2024).

COLLEGE DOCTORAL PAYS DE LA LOIRE
ED MATIERE, MOLECULES MATERIAUX ET GEOSCIENCES

Nantes Université

Titre : Méthode Deep Learning and analyse Double Calorimétrique pour la mesure de haute précision des paramètres d'oscillation des neutrinos dans JUNO

Mot clés : Neutrinos; expérience JUNO; Deep Learning; reconstruction d'IBD; oscillations des neutrinos; double calorimetrie

Résumé : JUNO est un observatoire de neutrinos à scintillateur liquide, polyvalent et medium baseline (environ 52 km), situé en Chine. Ses principaux objectifs sont de mesurer les paramètres d'oscillation θ_{12}, Δm^2_{21} et Δm^2_{31} avec une précision au pour-mille et de déterminer l'ordre des masses des neutrinos avec un niveau de confiance de 3σ. Atteindre ces objectifs nécessite une résolution énergétique sans précédent de $3\%/\sqrt{\mathrm{E(MeV)}}$ avec cette technologie. Cela demande une compréhension approfondie des divers effets au sein du détecteur. Le système de double calorimetrie, composé de deux systèmes de mesure distincts observant le même événement, permet non seulement une calibration mais aussi une détection des effets du détecteur avec une grande précision, comme démontré dans cette thèse. Le Deep Learning, un outil de plus en plus utilisé en physique expérimentale, joue un rôle crucial dans cet effort. Dans cette thèse, je présente le développement, l'application et l'analyse des techniques de Deep Learning pour la reconstruction d'évèvements dans l'expérience JUNO.

Title: Deep learning methods and Dual Calorimetric analysis for high precision neutrino oscillation measurements at JUNO

Keywords: Neutrinos; JUNO experiment; Deep learning; IBD reconstruction; neutrinos Oscillation; dual Calorimetry

Abstract: JUNO is a multipurpose, medium-baseline ($\sim$52 km) liquid scintillator neutrino observatory located in China. Its primary objectives are to measure the oscillation parameters θ_{12}, Δm^2_{21}, and Δm^2_{31} with per mil precision and to determine the neutrino mass ordering at a 3σ confidence level. Achieving these goals requires an unprecedented energy resolution of $3\%/\sqrt{\mathrm{E(MeV)}}$ with this technology. This demands a comprehensive understanding of the various effects within the detector. The Dual Calorimetry system—two distinct measurement systems observing the same event-enables not only high-precision calibration but also detection of detector effects, as demonstrated in this thesis. Deep learning, an increasingly powerful tool in physics, plays a critical role in this effort. In this thesis, I present the development, application, and analysis of Deep Learning techniques for reconstruction in the JUNO experiment.

4486