DOCTORAT BRETAGNE LOIRE / MATIERE MOLECULES ET MATERIAUX

UNIVERSITÉ DE NANTES

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE NANTES

ÉCOLE DOCTORALE Nº 596
Matière, Molécules, Matériaux
Spécialité : *Physique des particules*

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, LP2I Bordeaux
David Rousseau	Directeur de recherche, IJCLab

Composition du Jury :

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

Contents

Remerciements

Introduction

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.

1.1 Standard model

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

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

Chapter 2

The JUNO experiment

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

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.

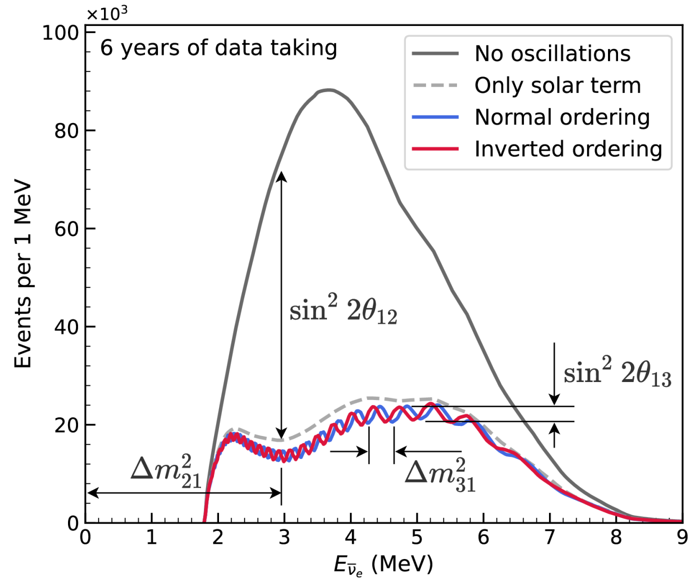


FIGURE 2.2 – Expected number of neutrinos event per MeV in JUNO after 6 years of data taking. The black curve shows the flux if there was no oscillation. The light gray curve shows the oscillation if only the solar terms are taken in account (θ_{12}, Δm^2_{21}). The blue and red curve shows the spectrum 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.

2.1 Neutrinos physics in JUNO

Even if the JUNO design detailed in section 2.2 was optimized for the measurement of the NMO, its large detection volume, excellent energy resolution and background level and understanding make it also an excellent detector to measure the flux coming from other neutrino sources. Thus the scientific program of JUNO extends way over reactor antineutrinos. The following section is an overview of the different physics topic JUNO will contribute in the coming years.

2.1.1 Reactor neutrino oscillation for NMO and precise measurements

Previous works [1, 3] shows that oscillation parameters and the NMO can be observed by looking at the $\bar{\nu}_e$ disappearance energy spectrum coming from medium baseline nuclear reactor. This disappearance probability can be expressed as [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. By carefully adjusting a theoretical spectrum to the data, one can extract the NMO and the oscillation parameters. The statistic procedure used to adjust the theoretical spectrum is reviewed in more details in the section 2.7. To reach the desired sensitivity, JUNO must meet multiple requirements but most notably:

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 precision of 1% in order to not err on the location of the oscillation pattern.
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 to limit the spectrum distortion due to statistical uncertainties.

$\bar{\nu}_e$ 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 $\bar{\nu}_e$.

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.

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

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

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. Using this method, the flux uncertainty is expected to be of an order of 2-3 % [11].

In addition to those prediction, a satellite experiment named TAO[12] 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 2.4.1. It will help identifying unknown fine structure and give more insight on the $\bar{\nu}_e$ flux coming from this reactor.

One the open issue about reactor anti-neutrinos flux is the so-called neutrino anomaly [13], an unexpected surplus of neutrino emission in the spectra around 5 MeV. Multiples scientists are trying to explain this surplus by advanced recalculation of the nuclei model during beta decay [14, 15] but no consensus on this issue has been reached yet.

Background in the neutrinos reactor spectrum

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 $\sim$ 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.

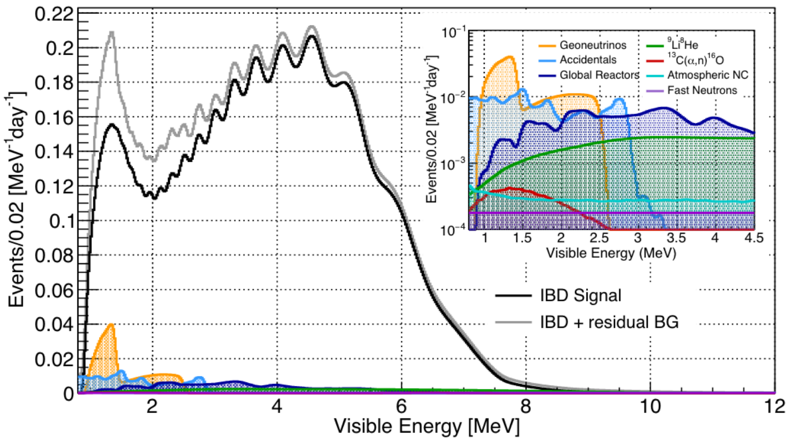


FIGURE 2.3 – Expected visible energy spectrum measured with the LPMT system with (grey) and without (black) backgrounds. The background amount for about 7% of the IBD candidate and are mostly localized below 3 MeV [11]

Identification of the mass ordering

To identify the mass ordering, we adjust the theoretical neutrino energy spectrum under the two hypothesis of NO and IO. Those give us two χ^2, respectively χ^2_{NO} and χ^2_{IO}. By computing the difference $\Delta\chi^2 = \chi^2_{NO} - \chi^2_{IO}$ we can determine the most probable mass ordering and the confidence interval: NO if $\Delta\chi^2 > 0$ and IO if $\Delta\chi^2 < 0$. Current studies shows that the expected sensitivity the mass ordering would be of 3.4σ after 6 years of data taking in nominal setup[2]. More detailed explanations about the procedure can be found in the section 2.7.

Precise measurement of the oscillations parameters

The oscillations parameters θ_{12}, θ_{13}, Δm^2_{21}, Δm^2_{31} are free parameters in the fit of the oscillation spectrum. The precision on those parameters have been estimated and are shown in table 2.2. Wee see that for θ_{12}, Δm^2_{21}, Δm^2_{31}, precision at 6 years is better than the reference precision by an order of magnitude [11]

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

TABLE 2.2 – A summary of precision levels fir the oscillation parameters. The reference value (PDG 2020 [16]) is compared with 100 days, 6 years and 20 years of JUNO data taking.

2.1.2 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 2.3 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	0–80 MeV	Negligible
	2300 elastic scattering		
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 2.3 – 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 mechanisms 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 [17].

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 $p \rightarrow \bar{\nu}K^+$ [18] 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.

2.2 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.

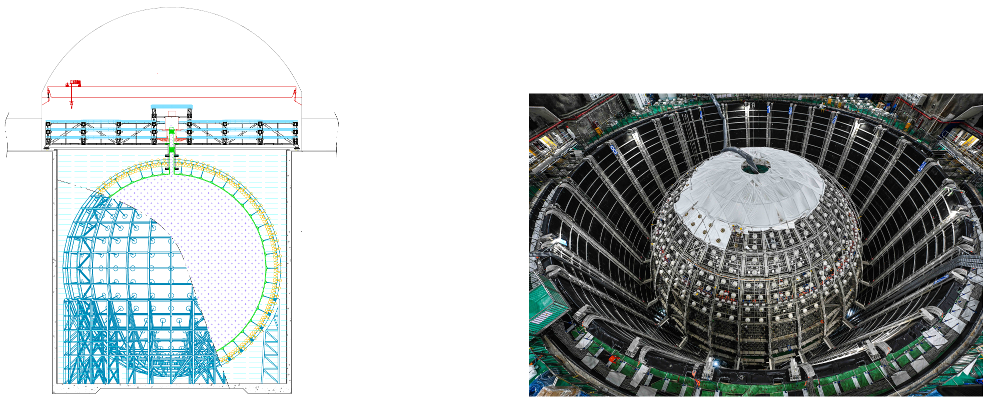
The top of the experiment also host the LS purification system, a water purification system, a ventilation system to get rid of the potential radon in the air. The CD is observed by two system of Photo-Multipliers Tubes (PMT). They are attached to the steel structure and their electronic readout is submersed 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.

2.2.1 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 [19]. 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 [19]

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

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 2.2.2). 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.

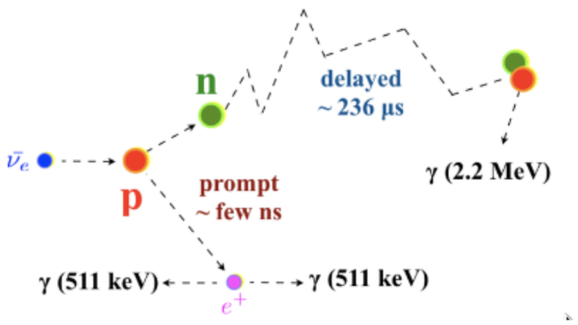


FIGURE 2.5 – Schematics of an IBD interaction in the central detector of JUNO

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 ($\sim 236\ \mu$s).

2.2.2 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.

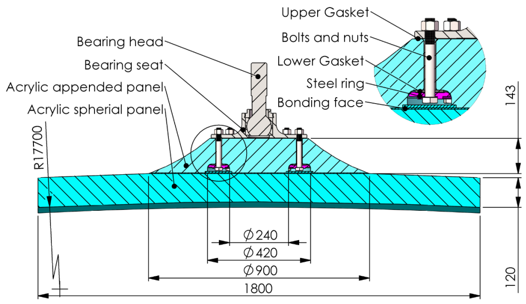


FIGURE 2.6 – Schematics of the supporting node for the acrylic vessel

Liquid scintillator

The Liquid Scintillator (LS) has a similar recipe as the one used in Daya Bay [20] 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 [21]. 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.

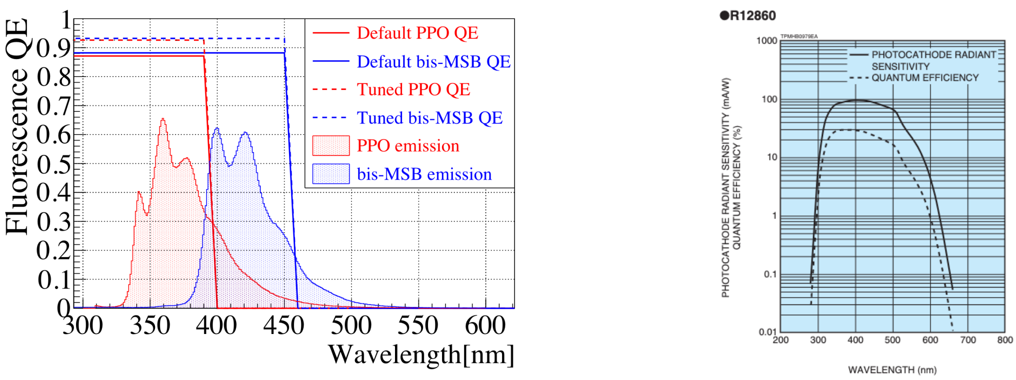


FIGURE 2.7 – **On the left:** Quantum efficiency (QE) and emission spectrum of the LAB and the bis-MSB [20]. **On the right:** Sensitivity of the Hamamatsu LPMT depending on the wavelength of the incident photons [22].

This formula has been optimized using dedicated studies with a Daya Bay detector [20, 23] 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 [24] 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) [25] to ensure that the requirements are kept during the lifetime of the experiment, more details to be found in section 2.4.2.

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.

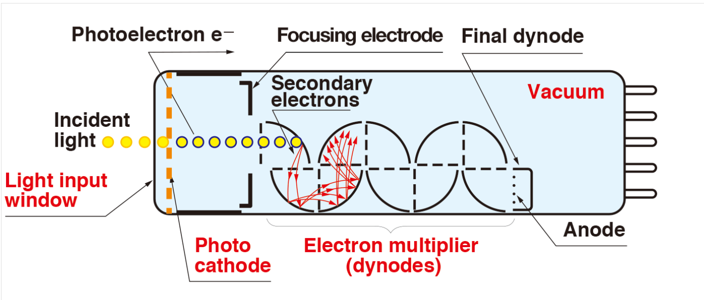


FIGURE 2.8 – Schematic of a PMT

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 [22] were produced by the Hamamatsu© company and $\sim$ 15000 Micro-Channel Plate (MCP) [26] 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 2.3 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).

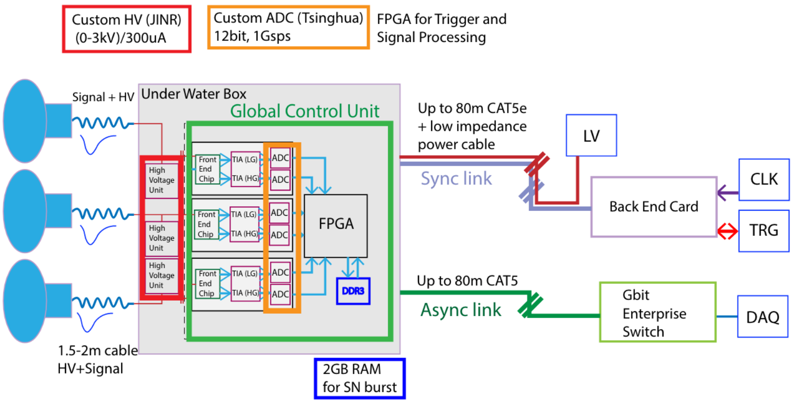


FIGURE 2.9 – The LPMT electronics scheme. It is composed of two part, the *wet* electronics on the left, located underwater and the *dry* electronics on the right. They are connected by Ethernet cable for data transmission and a dedicated low impedance cable for power distribution

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 2.3). Using this property, the intrinsic charge non linearity of the LPMTs can be measured by comparing the PE count in the SPMTs and LPMTs [27]. 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.

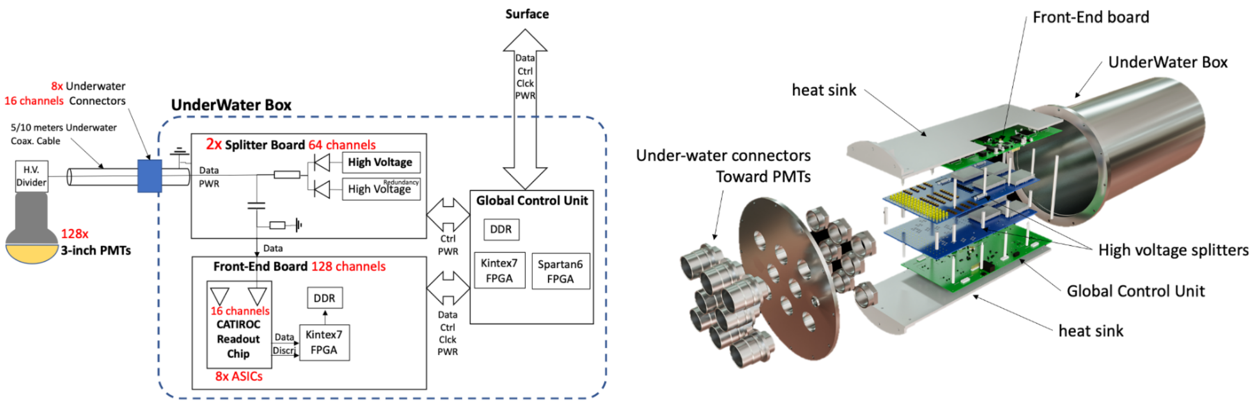


FIGURE 2.10 – Schematic of the JUNO SPMT electronic system (**left**), and exploded view of the main component of the UWB (**right**)

2.2.3 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 [28] 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.

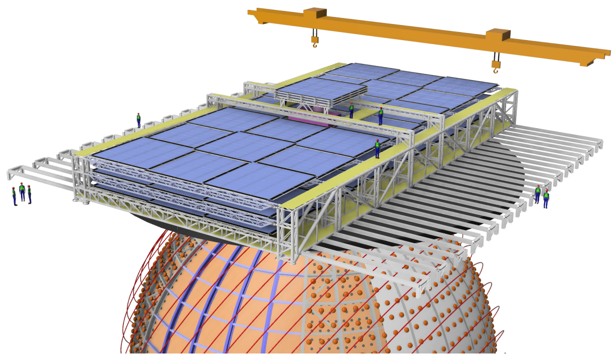


FIGURE 2.11 – The JUNO top tracker

2.3 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 2.6). We also expect a non-linear energy response of the CD due to the LS properties [20] but also due to the saturation of the LPMTs system when collecting a large amount of PE [27].

2.3.1 Energy scale calibration

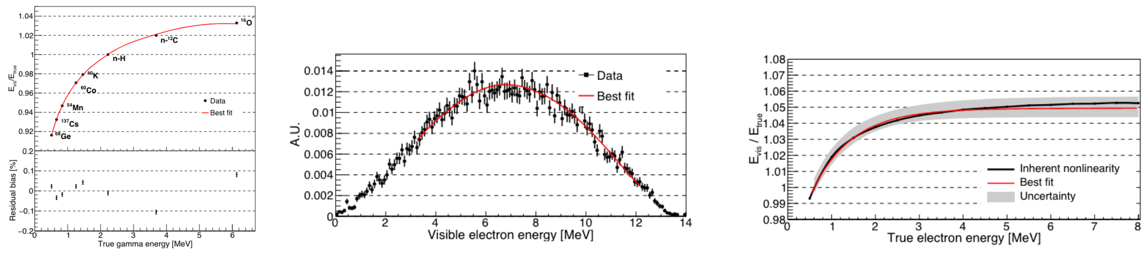
While electrons and positrons sources would be ideal, for a large LS detector thin-walled electrons or positrons sources could lead to leakage of radionucleides causing radioactive contamination. Instead, we consider gamma sources in the range of the prompt energy of IBDs. The sources are reported in table 2.4.

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 2.4 – 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 [29] 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 [29].

2.3.2 Calibration system

The non-uniformity due to the event position in the detector (more details in section 2.6) 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 2.5.

2.3.3 Instrumental non-linearity calibration

As mentioned in the introduction of this section, we expect an instrumental non-linearity due to the LPMT system saturating. This results in the LPMT underestimating the number of collected photo-electrons. This non-linearity is illustrated in figure 2.14. This non-linearity would consequently convolve with the LS non-linearity. To correct this effect, the LPMT are first calibrated to the channel level using the dual calorimetry calibration technique which consist of comparing the LPMT and SPMT calorimetry calibration using a tunable light source covering the range of 0 to 100 PE per LPMT channel.

Within such range, the SPMT serve as an approximate linear reference since SPMT operate primarily operate in photo-counting mode in this range. Using this technique, the residual non-linearity in the LPMT response due to the saturation effect is under 0.3 %.

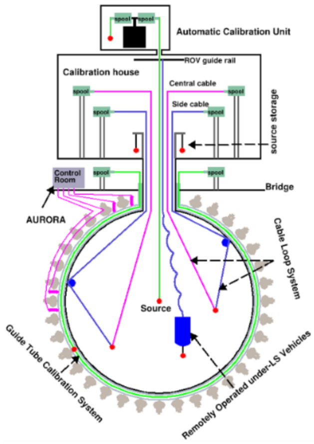


FIGURE 2.13 – Overview of the calibration system

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 2.5 – Calibration program of the JUNO experiment

2.4 Satellite detectors

As introduced in section 2.1.1 and section 2.2.2, 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.

2.4.1 TAO

The Taishan Antineutrino Observatory (TAO) [12, 30] 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 provide a major contribution to the so-called reactor anomaly [13]. Its requirement are to the level of 2 % energy resolution at 1 MeV.

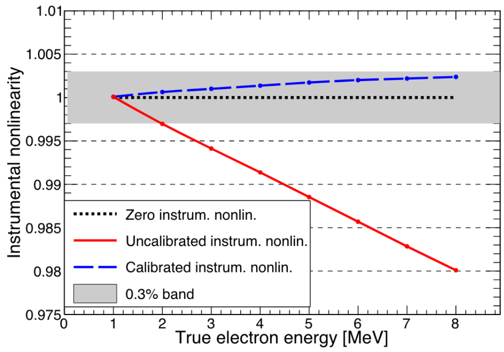


FIGURE 2.14 – Event-level instrumental non-linearity, defined as the ratio of the total measured LPMT charge to the true charge for events uniformly distributed in the detector.The solid red line represents event-level non-linearity without the channel-level correction, with position non-uniformity obtained at 1 MeV applied, 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 [29].

Detector

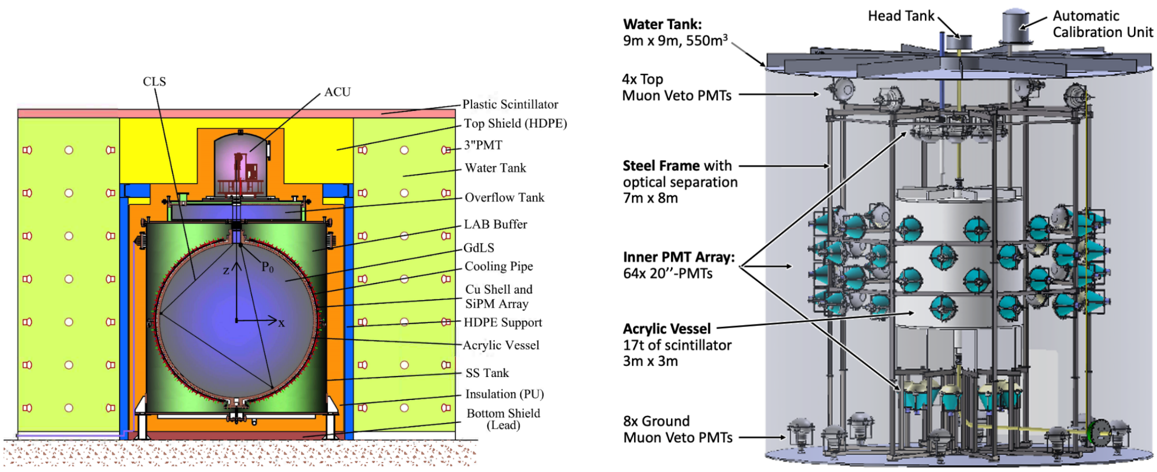
The TAO detector is close, in concept, to the CD of JUNO. It is composed of an acrylic vessel containing 2.8 tons of gadolinium-loaded LS instrumented by an array of silicon photomultipliers (SiPM) reaching a 95% coverage. To efficiently reduce the dark count of those sensors, the detector is cooled 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.

2.4.2 OSIRIS

The Online Scintillator Internal Radioactivity Investigation System (OSIRIS) [25] 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

2.5 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 [31] framework:

- — Various primary particles simulators for the different kind of events, background and calibration sources.
- — A Geant4 [32–34] 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 2.6 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.

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.

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

The main reconstruction method currently run in JUNO is a data-driven method based on a likelihood maximization [35, 36] 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.

2.6.1 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.

$$\vec{r}_{cb} = a \cdot \frac{\sum_i q_i \cdot \vec{r}_i}{\sum_i q_i} \tag{2.3}$$

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 $a \approx 1.3$ [37]. The results in figure 2.16b 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 2.16a 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.

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.

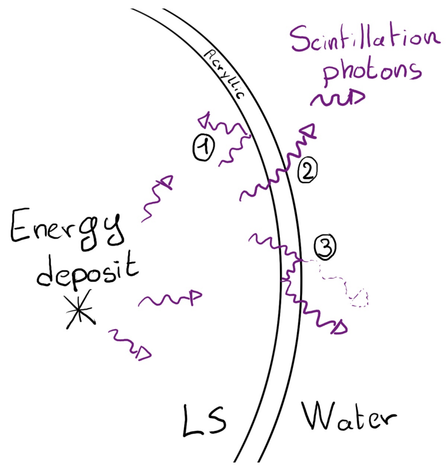
Time based algorithm

The time based algorithm use the distribution of the time of flight corrections Δt (Eq 2.4) of an event to reconstruct its vertex and t_0. It follow the following iterations:

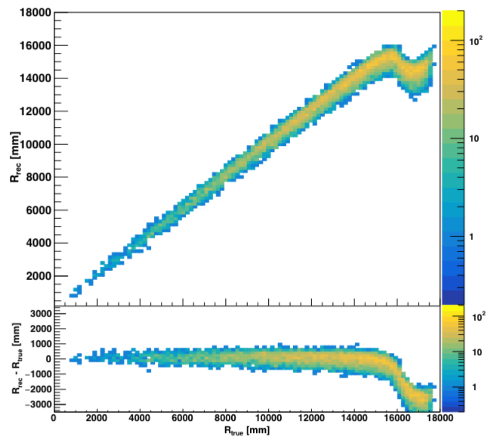
1. Use the charge based algorithm to get an initial vertex to start the iteration.
2. Calculate the time of flight correction for the ith PMT using

$$\Delta t_i(j) = t_i - \text{tof}_i(j) \tag{2.4}$$

   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 [37] for detailed description of this effective velocity). Plot the Δt distribution and label the peak position as Δt^{peak} (see fig 2.17a).



(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 2.16

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

$$\vec{\delta}[\vec{r}(j)] = \frac{\sum_i \left(\frac{\Delta t(j) - \Delta t^{\text{peak}}(j)}{\text{tof}_i(j)}\right) \cdot (\vec{r}_0(j) - \vec{r}_i)}{N^{\text{peak}}(j)} \tag{2.5}$$

   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 $\vec{\delta}[\vec{r}(j)] < 1\text{mm}$ 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 [38–40]. To reduce bias in the vertex reconstruction, the following equation is used to correct t_i into t'_i:

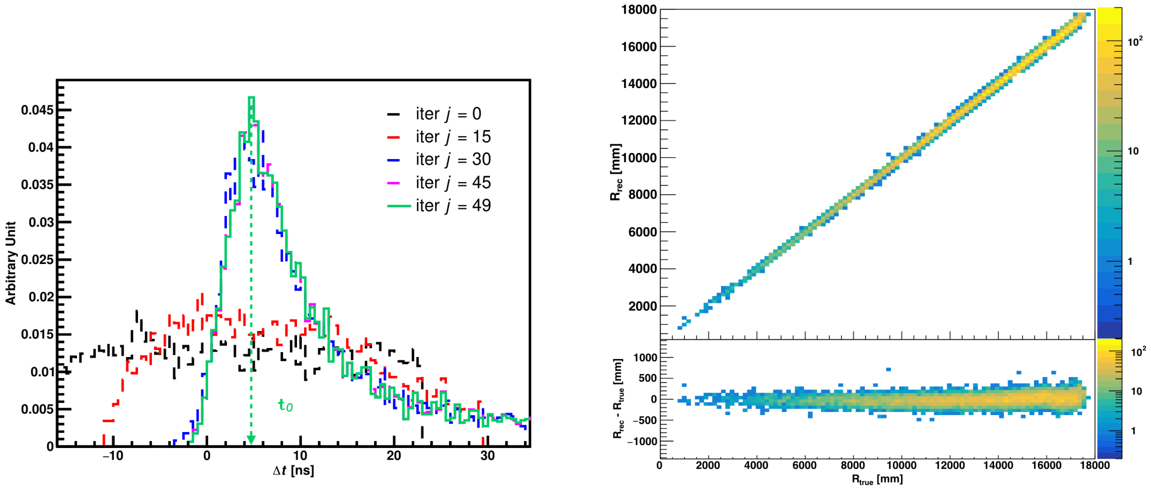
$$t'_i = t_i - p_0/\sqrt{N_i^{\text{pe}}} - p_1 - p_2/N_i^{\text{pe}} \tag{2.6}$$

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 [37]. The results presented in figure 2.17b 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

$$t^i_{\text{res}}(\vec{r}_0, t_0) = t_i - \text{tof}_i - t_0 \tag{2.7}$$



(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 2.17

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

$$f(t_{\text{res}}) = \sum_k \frac{\rho_k}{\tau_k} e^{\frac{-t_{\text{res}}}{\tau_k}}, \sum_k \rho_k = 1 \tag{2.8}$$

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

$$g(t_{\text{res}}) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(t_{\text{res}}-\nu)^2}{2\sigma^2}} \cdot f(t_{\text{res}}) \tag{2.9}$$

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

$$p(t_{\text{res}}) = (1 - \epsilon_{\text{dn}}) \cdot g(t_{\text{res}}) + \epsilon(t_{\text{res}}) \tag{2.10}$$

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

$$\mathcal{L}(\vec{r}_0, t_0) = -\ln\left(\prod_i p(t^i_{\text{res}})\right) \tag{2.11}$$

The parameter of Eq. 2.10 can be measured experimentally. The results shown in figure 2.18 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.

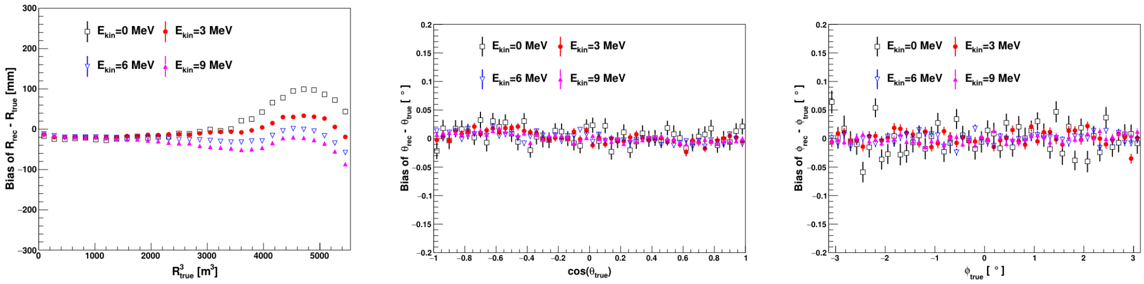


FIGURE 2.18 – Bias of the reconstructed radius R (left), θ (middle) and ϕ (right) for multiple energies by the time likelihood algorithm

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

$$P(\vec{r}_0, E) = \prod_j P^j_{nohit}(\vec{r}_0, E) \cdot \prod_i P^i_{hit}(\vec{r}_0, E) \tag{2.12}$$

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

$$\mathcal{L}(\vec{r}_0, E) = -\ln(P(\vec{r}_0, E)) \tag{2.13}$$

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

$$\mu_i(\vec{r}_0, E) = Y \cdot \frac{\Omega(\vec{r}_0, r_i)}{4\pi} \cdot \epsilon_i \cdot f(\theta_i) \cdot e^{-\sum_m \frac{d_m}{\zeta_m}} \cdot E + \delta_i \tag{2.14}$$

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. 2.14 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 2.3. In the results presented in figures 2.19, the PDF was produced using MC simulation and 29 specific calibrations position [37] along the Z-axis of the detector. We see that the charge likelihood algorithm show little bias in the TR area and a better resolution than the time likelihood. The figure 2.20 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.

The charge based likelihood algorithms already give use some information on the energy as Eq. 2.13 is minimized but the energy can be further refined as shown in the next section.

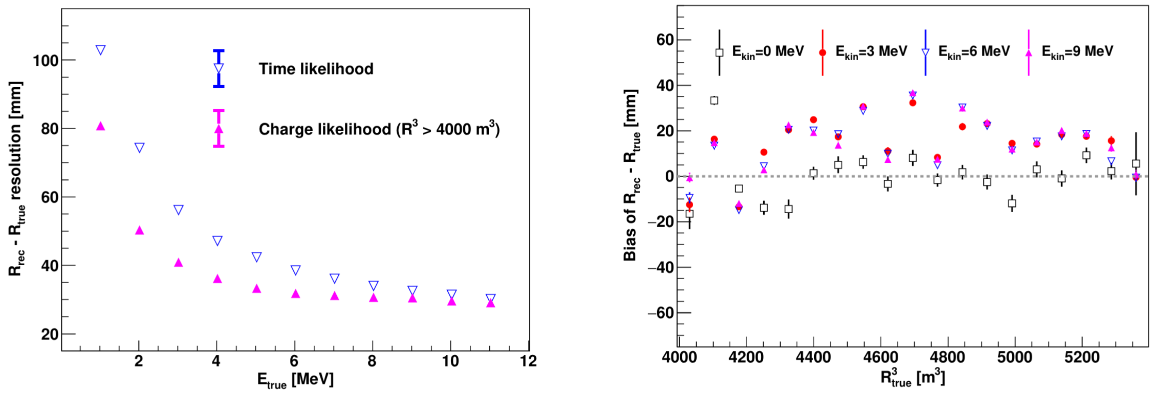


FIGURE 2.19 – **On the left:** Resolution of the reconstructed R as a function of the energy in the TR area ($R^3 > 4000\mathrm{m}^3 \equiv R > 16m$) by the charge and time likelihood algorithms. **On the right:** Bias of the reconstructed R in the TR area for different energies by the charge likelihood algorithm

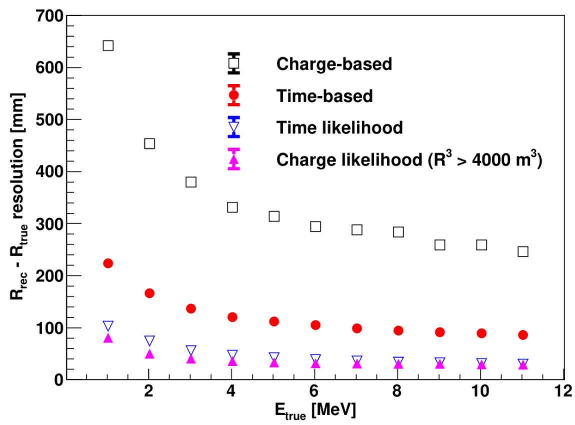


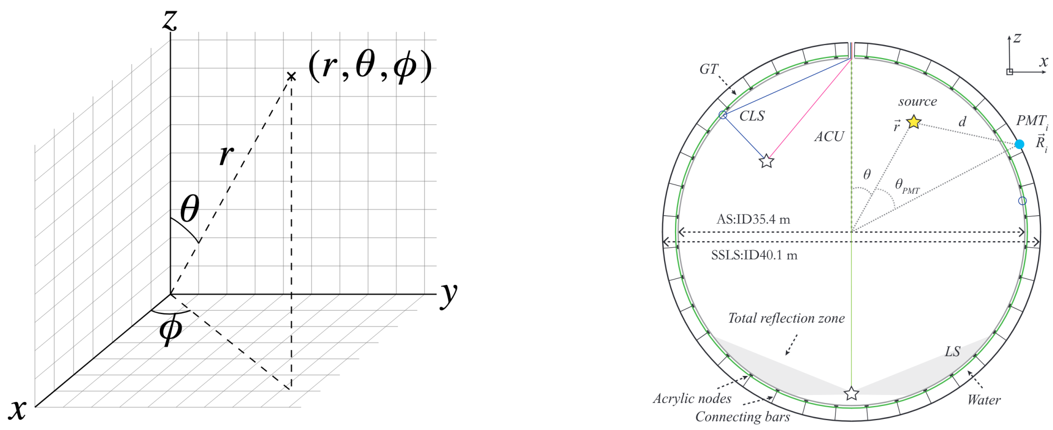
FIGURE 2.20 – Radial resolution of the different vertex reconstruction algorithms as a function of the energy

2.6.2 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 [41].

Charge estimation

The most important element in the energy reconstruction is $\mu_i(\vec{r}_0, E)$ described in Eq. 2.14. 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



(A) Spherical coordinate system used in JUNO for reconstruction

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

FIGURE 2.21

using $(r, \theta, \theta_{pmt})$ as defined in figure 2.21b.

$$\hat{\mu}(r, \theta, \theta_{pmt}, E_{vis}) = \frac{1}{E_{vis}} \frac{1}{M} \sum_{i}^{M} \frac{\frac{\bar{q}_i}{\hat{Q}_i} - \mu_i^D}{\mathrm{DE}_i}, \ \mu_i^D = \mathrm{DNR}_i \cdot L \tag{2.15}$$

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 [41]. $\bar{q}_i$ 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 [36].

Time estimation

The second important observable is the hit time of photons that was previously defined in Eq. 2.7. It is here refined as

$$t_r = t_h - \mathrm{tof} - t_0 = t_{LS} + t_{TT} \tag{2.16}$$

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

$$t_{TT} = \mathcal{N}(\overline{\mu_{TT} + t_d}, \sigma_{TT}) \tag{2.17}$$

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 [41]. 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

$$F(t, r, d) = \int_t^L f(t', r, d)dt' \tag{2.18}$$

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

$$P_T^l(t|k = n) = I_n^l[f_l(t)F_l^{n-1}(t)] \tag{2.19}$$

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

$$f_d(t) = 1/L,\ F_d(t) = 1 - \frac{t}{L} \tag{2.20}$$

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

$$P_T(t|\mu_l, \mu_d, k = 1) = I_1[P(1, \mu_l)P(0, \mu_d)f_l(t) + P(0, \mu_l)P(1, \mu_d)f_d(t)] \tag{2.21}$$

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:

$$\mathcal{L}(q_1, q_2, ..., q_N|\vec{r}, E_{vis}) = \prod_{j\in\text{unfired}} e^{-\mu_j} \prod_{i\in\text{fired}} \left(\sum_{k=1} P_Q(q_i|k) \cdot P(k, \mu_i) \right) \tag{2.22}$$

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:

$$\mathcal{L}(t_{1,r}, t_{2,r}, ..., t_{N,r}|\vec{r}, t_0) = \prod_{i\in\text{hit}} \frac{\sum_{k=1}^K P_T(t_{i,r}|r, d, \mu_i^l, \mu_i^d, k) \cdot P(k, \mu_i^l + \mu_i^d)}{\sum_{k=1}^K P(k, \mu_i^l + \mu_i^d)} \tag{2.23}$$

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

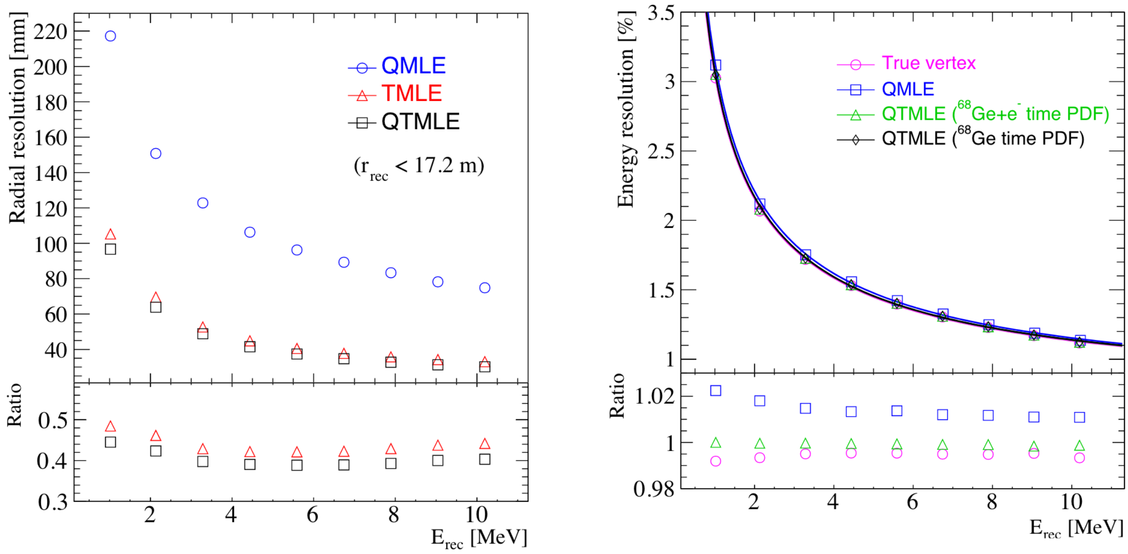
$$\mathcal{L}(q_1, q_2, ..., q_N; t_{1,r}, t_{2,r}, ..., t_{N,r}|\vec{r}, t_0, E_{vis}) = \mathcal{L}(q_1, q_2, ..., q_N|\vec{r}, E_{vis}) \cdot \mathcal{L}(t_{1,r}, t_{2,r}, ..., t_{N,r}|\vec{r}, t_0) \tag{2.24}$$

The radial and energy resolutions of the different likelihood are presented in figure 2.22 (from [41]). 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.

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 the next section, we'll see another type of data-driven method based on machine learning.

2.6.3 Machine learning for reconstruction

Machine learning (ML) is family of data-driven algorithms that are inferring behavior and results from a training dataset. A overview of methods and detailed explanation of the Neural Network (NN) subfamily can be found in Chapter 3.



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

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

FIGURE 2.22

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:

$$\mathbb{R}^{5\times N_{pmt}} \longmapsto \mathbb{R}^4 \tag{2.25}$$

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 2.5 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 (see chapter 3.1). They have been explored for vertex reconstruction [42] et for energy reconstruction [42, 43].

For vertex and energy reconstruction a BDT was developed using the aggregated informations presented in 2.6.

Its reconstruction performances are presented in figure 2.24.

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 2.6 – Features used by the BDT for vertex reconstruction

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 2.7 – 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

A second and more advanced BDT, subsequently named BDTE, that only reconstruct energy use a different set of features [43]. They are presented in the table 2.7

Neural Network (NN)

The physics have shown a rising for Neural Network (NN) in the past years for event reconstruction, notably in the neutrino community [44–47]. Three type of neural networks have explored for event reconstruction in JUNO Deep Neural Network (DNN), Convolutional Neural Network (CNN) and Graph Network (GNN). More explanation about those neural network can be found in chapter 3.

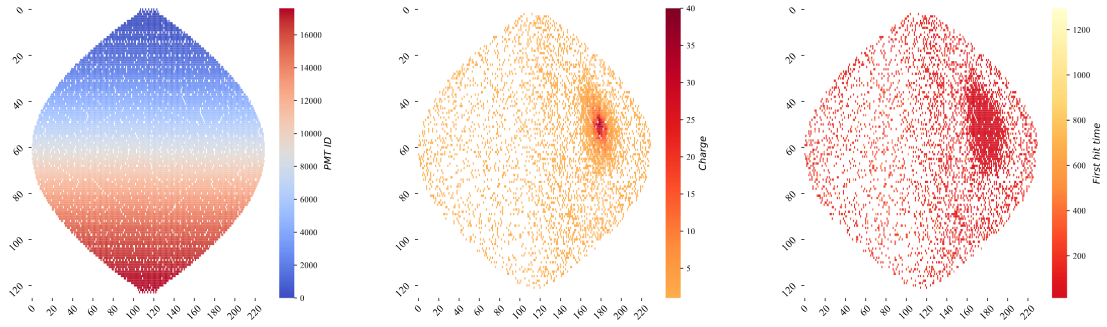
The CNN are using 2D projection of the detector representing it as an image with two channel, one for the charge Q and one for the time t. The position of the PMTs is structurally encoded in the pixel containing the information of this PMT. In [42], the pixel is chosen based on a transformation of θ 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 2.23. The performances of the CNN can be found in figure 2.24.

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 [42] the nodes, are geometrical region of the detector as defined by the HealPix [48]. 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 [49]. The performances of the GNN are presented in figure 2.24.



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

FIGURE 2.23 – 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

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.

2.7 JUNO sensitivity to NMO and precise measurements

Now that the event have been reconstructed, selected and that the non-IBD background have been rejected, we have access to the measured energy flux from JUNO. We consider two spectra, the one measured by the LPMT system and the one measured by the SPMT system. This give rise to three possible analysis: A LPMT only analysis, a SPMT only analysis and a joint analysis. This joint analysis is the subject of the chapter 7 of this thesis.

The following details about JUNO measurement is common to the three analysis. The details and specific of the joint analysis are detailed in chapter 7.

2.7.1 Theoretical spectrum

To extract the oscillation parameters and the NMO from the measured spectrum, it is compared to a theoretical spectrum. This theoretical spectrum is produced based on the theory of the three flavour oscillation (see section 1.3), the measurements produced by the calibration, the input from TAO and adjusted Monte Carlo simulations:

- The absolute flux and the fission product fraction yield calibrated by TAO.
- The estimation of the neutrinos flux from other sources, such as the geoneutrinos, by theoretical model.
- The computed cross-section of $\bar{\nu}_e$ and the LS.
- The estimation of mislabelled event, such as fast neutron events from cosmic muons, using Monte Carlo simulation.
- The measured bias and resolution of the LPMT and SPMT system by the calibration.
- The time dependent reactor parameters (age of fuel, instantaneous power of the reactors, etc...)

These systematics parameters come with their uncertainties that need to be taken into account by the fitting framework. This theoretical spectrum will, in the end, depend of the oscillation parameters of

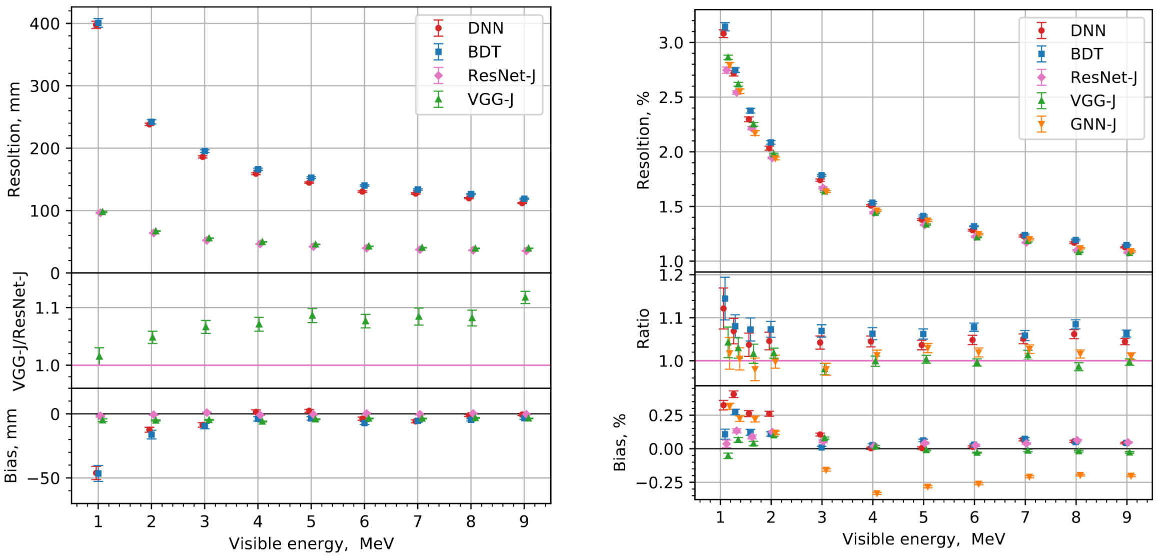


FIGURE 2.24 – Radial (left) and energy (right) resolutions of different ML algorithms. The results presented here are from [42]. DNN is a deep neural network, BDT is a BDT, ResNet-J and VGG-J are CNN and GNN-J is a GNN.

interest $\theta_{13}, \theta_{12}, \Delta m^2_{21}, \Delta m^2_{31}$. Noise parameters can be included in the parameters spectrum such as the earth density ρ between the power plants and JUNO.

2.7.2 Fitting procedure

The theoretical and measured spectra are represented as two histograms depending on the energy. The theoretical spectrum is adjusted with the data using a χ^2 minimization where χ^2 is naively defined as

$$\chi^2 = \sum_i \frac{(N^i_{th} - N^i_{data})^2}{\sigma_i^2} \tag{2.26}$$

where N^i_{th} is the number event in the ith bin of the theoretical spectrum, N^i_{data} is the number of event in the ith bin of the measured spectrum and σ_i is the uncertainty of this bin. Two classic statistic test exist Pearson and Neyman where the difference is the estimation of σ_i parameters.

This σ_i is composed of the systematics uncertainties discussed above but also from the statistic uncertainty of the spectrum. Considering a Poisson process, the statistic uncertainty is estimated as $\sigma^i_{stat} = \sqrt{N^i}$. In a Pearson test, $N^i \equiv N^i_{th}$ whereas in a Neyman test $N^i \equiv N^i_{data}$. Under the assumption that the content of each bin follow a Gaussian distribution (a Poisson with high enough statistic), the two test are equivalent. But studies on Monte Carlo spectrum showed that the Pearson and Neyman statistic are biased in opposite direction. It is easily visible where, for the same data, Pearson will prefer a higher N^i_{th} to reduce the ration $\frac{1}{N^i_{th}}$ whereas Neyman will prefer a lower N^i_{th} to reduce the $(N^i_{th} - N^i_{data})$ term.

This problematic can be circumvented by summing the two test, yielding the CNP statistic test and/or by adding a term

$$\chi^2 = \sum_i \frac{(N^i_{th} - N^i_{data})^2}{\sigma_i^2} - \ln|\mathbf{V}| \tag{2.27}$$

where V is the covariance matrix of the theoretical spectrum yielding the PearsonV and CNPV statistic test.

The χ^2 is minimized by exploring the parameter phase space via gradient descent.

2.7.3 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 [11]. The expected sensitivity to mass ordering is 3σ after 6.5 years [50].

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

Machine learning and Artificial Neural Network

> *"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*

Machine Learning (ML) and more specifically Neural Network (NN) are families of data-driven algorithms. They are used to model complex distributions from a finite dataset to extract a generalist behavior. They learn, adapt their intrinsic parameters, interactively by computing their performances or *loss* on those datasets. They take advantage of simple microscopic operations such as *if condition* or non-continuous but differentiable function like *ReLU* in heavy numbers to model macroscopic complex and precise behaviours.

They are now widely used in a wide variety of domain including natural language processing, computer vision, speech recognition and, the subject of this thesis, scientific studies.

We found them in particle physics, either as the main algorithm or as secondary algorithm, for event reconstruction, event classification, waveform reconstruction, etc..., domains where the underlying physic and detector processes are complex and highly dimensional. Physicists have traditionally been forced to use simplifications or assumptions to ease the developmment of algorithms or equations (a good example is the algorithm presented in section 2.6) where machine learning could refine and take into account those effects, provided that they have enough data and computing power.

This chapter present an overview of the different kind of machine learning methods and neural networks that will be discussed in this thesis.

3.1 Boosted Decision Tree (BDT)

One of the most classic machine learning algorithm used in particle physics is Boosted Decision Tree (BDT) [51] (or more recently Gradient Boosting Machine [52]). The principle of a BDT is fairly simple : based on a set of observables, a serie of decisions, represented as node in a tree, are taken by the algorithm. 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 follow a simple score reward procedure. During the training phase the prediction of the BDT is compared to a known truth about the data. The score is then used to

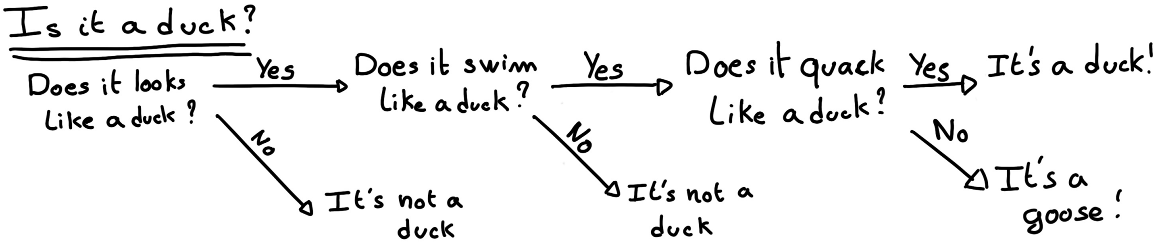


FIGURE 3.1 – Example of a BDT that determine if the given object is a duck

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.

3.2 Artificial Neural Network (NN)

One other big family of machine learning algorithm is the artificial Neural Networks (NN). The idea of developing automates which component mimic, in a simplistic way, the behavior of biological neurons emerge in 1959 with the paper "*What the Frog's Eye Tells the Frog's Brain*" [53]. They develop an automate where each component possess an *activation function*. Each one of those components then transmit its information to the other following a certain efficiency or *weight*. Those works influenced scientist and notably Frank Rosenblatt who published in 1958 what is considered the first neural network model the Perceptron [54].

Modern neural network still nowadays use the neuron metaphor to represent neural network, but approach them as a graph where the nodes are neurons possessing an activation function and edges holding the weights, or *parameters* in modern literature, between those nodes. Most of the modern neural network work with the principle of neurons layers. Each neurons belong to a layer and takes input from the preceding layer and forward it result to next layer. For example the most basic couple of layers is the fully connected layers where each neurons of the input layer is connected to every other neurons of ouput layer. All the neurons posses the same activation function F. The connection between the two layers is expressed as a tensor T_j^i where i is the index of the precedent layer and j the index of the next layer. The propagation from the layer I to J is then described as

$$J_j = F_J(T_j^i I_i + B_j) \tag{3.1}$$

where the learning parameters are the tensor T_j^i and the bias tensor B_j. This is the fundamental component of the Fully Connected Deep NN (FCDNN) family presented in section 3.2.1. Most of the modern neural networks use gradient descent to optimize their parameters, i.e. the gradient of the parameter θ in respect of the loss function $\mathcal{L}$ is subtracted each optimisation step

$$\theta_{i+1} = \theta_i - \frac{\partial \mathcal{L}}{\partial \theta} \tag{3.2}$$

i being the training step index. This induce $\mathcal{L}$ needs to be differentiable with respect to θ, 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 [55] (more details in section 3.2.5).

This description of neural networks as layers introduced 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*.

The loss $\mathcal{L}$ described above is a score representing how well the NN is doing. As seen above, it needs to be differentiable with respect to the parameter of the NN. 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, a common loss is the Mean Square Error (MSE). Let i be our dataset, y_i be the target scalar, x_i the input data and $f(x_i,)$ the result of the network. The network here is modelled by f, and its parameter by the set

$$\mathcal{L} :- MSE = \frac{1}{N}\sum_i^N (y_i - f(x_i))^2 \tag{3.3}$$

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

$$\mathcal{L} :- MAE = \frac{1}{N}\sum_i^N |y_i - f(x_i)| \tag{3.4}$$

3.2.1 Fully Connected Deep Neural Network (FCDNN)

The Fully Connected Deep Neural Network (FCDNN) architecture is the natural evolution of the Perceptron. The input data is represented as a first order tensor I_j and then fed forward to multiple fully connected layers (Eq 3.1) as presented in the figure 3.2a. Most of the time, the classic ReLU function

$$\mathrm{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

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

$$\mathrm{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{3.6}$$

$$\mathrm{PReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{otherwise} \end{cases} \tag{3.7}$$

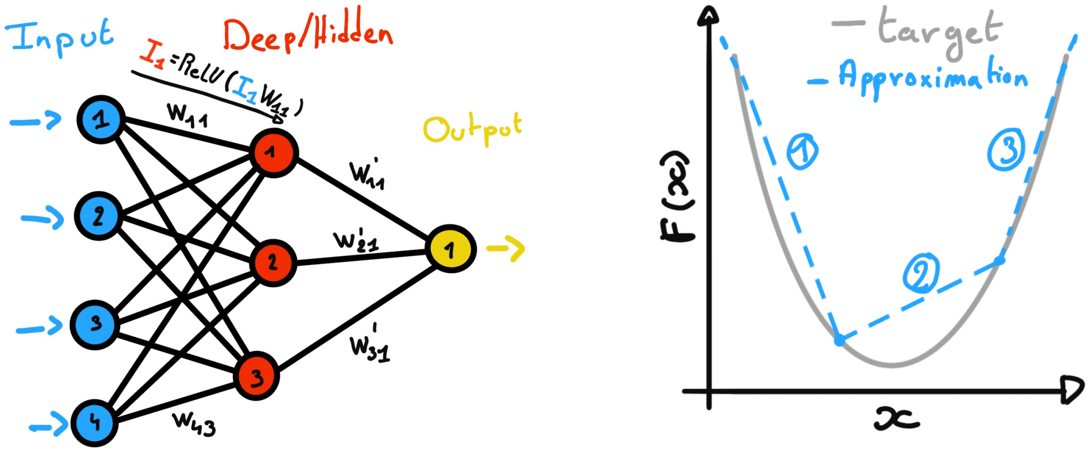
The reasoning behind ReLU and PReLU is that with enough of them, you can mimic any continuous function as illustrated in figure 3.2b. Sigmoid is more used in case of classification, its behavior going hand in hand with the Cross Entropy loss function used in classification problems.

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)

Convolutional Neural Networks are a family of neural networks that use discrete convolution filters, as illustrated in an example in figure 3.3, to process the input data, often images. They have the advantage to be translation invariant by construction, this mean that they are capable of detecting oriented features independently of their location on the image. The learning parameters are located in the filters, the network thus learn the optimal filters to extract the desired features. 2D CNN, where the filters are second order tensors that span over third order tensors, are commonly used in image recognition [56] for classification or regression problematics.

The convolution layers are commonly chained [57], 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. 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



(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 3.2

filters is commonly then flattened and feed to a smaller FCDNN which will process the filters results to yield the desired output.

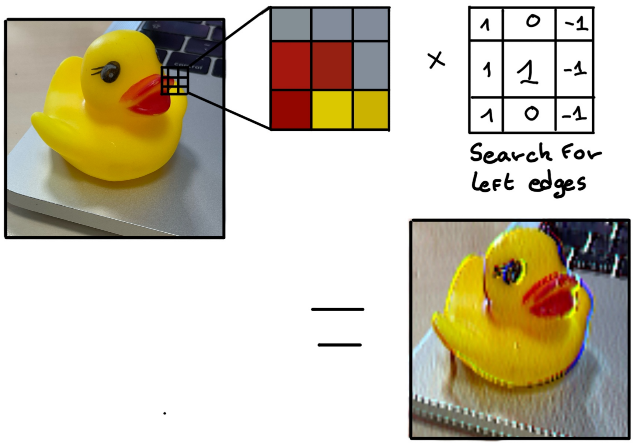


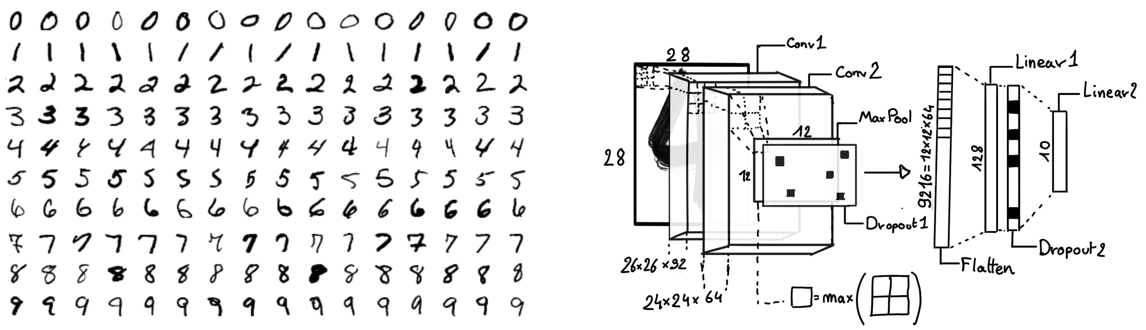
FIGURE 3.3 – Illustration of the effect of a convolution filter. Here we apply a filter with the aim do detect left edges. We see in the resulting image that the left edges of the duck are bright yellow where the right edges are dark blue indicating the contour of the object. The convolution was calculated using [58].

As an example, let's take the Pytorch [59] example for the MNIST [60], a dataset of black and white images of handwritten digits. Those images are 28×28 pixels with only one channel corresponding to the grey level of the pixel. Example of images from this dataset are presented in figure 3.4a

A schema of the CNN used in the Pytorch example is presented in figure 3.4b. 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 3.2.6 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. 3.1) 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.



(A) Example of images in the MNIST dataset

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

FIGURE 3.4

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 [61] (14 epochs) and reach an accuracy in its prediction of 99%.

3.2.3 Graph Neural Network (GNN)

Graph neural network is a family of neural network where the data is represented as a graph $G(\mathcal{N}, \mathcal{E})$ 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^{ij}_{ϵ} the adjacency tensors that hold the features $\epsilon \in [0, k_e]$ of the edge connecting the node i and j and the tensor N^i_ν that hold the features $\nu \in [0, k_n]$ of a node i.

To efficiently manipulate such object 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" [62]. In this algorithm, with each layer of message passing a new set of features is computed for each node following

$$n_i^{k+1} = \phi_u(n_i^k, \square_j \phi_m(n_i^k, n_j^k, e_{ij}^k)); \; n_j \in \mathcal{N}_i' \tag{3.8}$$

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 $e_{i,j} \to (n_i, n_j)$. k is the layer on which the message passing algorithm is applied. $\square$ 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 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 3.5. 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.

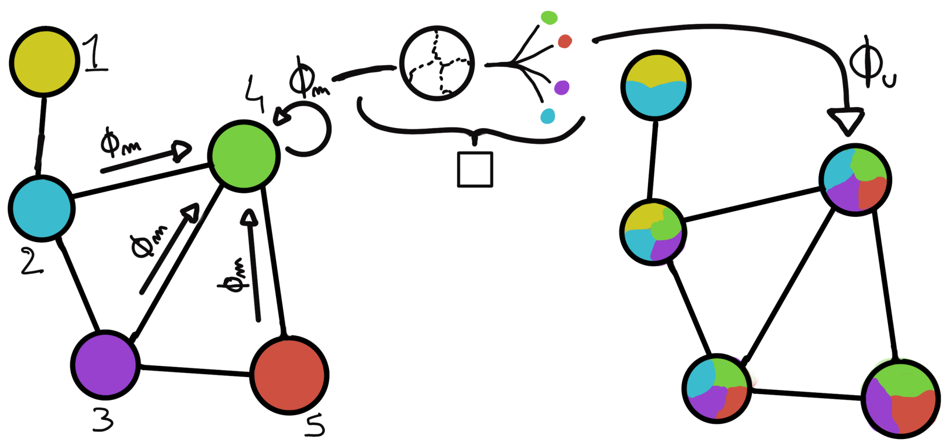


FIGURE 3.5 – Illustration of the message passing algorithm. The detailed explanation can be found in section 3.2.3

Message passing is a very generic way of describing the process of GNN and it can be specialized for convolutional filtering [49], diffusion [63] 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. [64], 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.

3.2.5 Training procedure

A neural network without the adequate training is like an empty shell. If the parameters are not optimized they are, most of the time, initialized to random number and so the output will just be random. The training is a key step in the production of a solid and reliable NN. This section aim to give an overview of the different concept and tools used in the training of our neural networks.

Training lifecycle

The training of NN does not follow strict rules, you could imagine totally different lifecycle but I will describe here the one used in this thesis, the most common one.

The training is split into *epochs* during which the NN will train on a set of subsamples called *batch*. The size of those batch is called *batch size*, a.k.a. the number of data it contains (how many images, how many events,...). Each process of a batch is called a *step*. At the end of each epochs, the neural network is evaluated over a validation dataset. This validation dataset is not used for training (no gradient of the loss is computed) and is used as reference for the network performance and monitor overtraining (see section 3.2.6). Most of the time, the parameters are updated at each step using the mean loss over the batch and the optimizer hyperparameters are updated at each epochs.

The optimizer

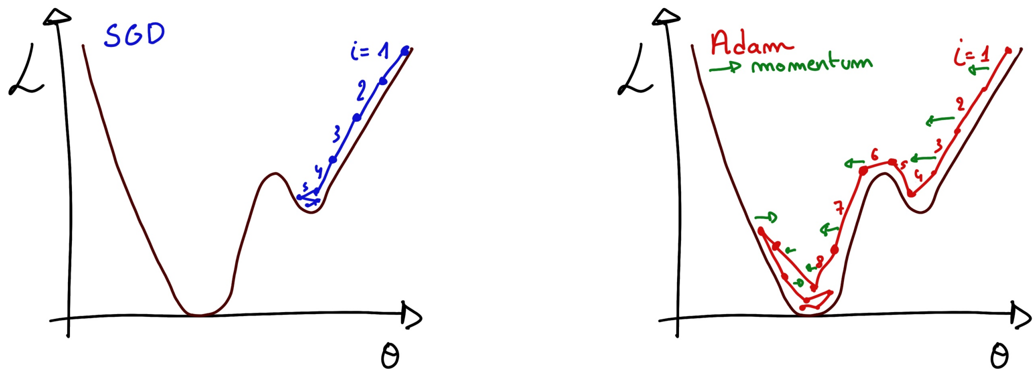
As briefly introduced section 3.2, the parameters of the neural network are optimized using the gradient descent method. We calculate the gradient of the mean loss over the batch with respect of each parameters and we update the parameters in accord to minimize the loss. The gradient is computed backward from the loss up to the first layer parameters using the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \theta_1} = \frac{\partial \theta_2}{\partial \theta_1}\frac{\partial \mathcal{L}}{\partial \theta_2} = \frac{\partial \theta_2}{\partial \theta_1}\frac{\partial \theta_3}{\partial \theta_2}\frac{\partial \mathcal{L}}{\partial \theta_3} = \frac{\partial \theta_2}{\partial \theta_1}\prod_{i=2}^{N-1}\frac{\partial \theta_{i+1}}{\partial \theta_i}\frac{\partial \mathcal{L}}{\partial \theta_N} \tag{3.9}$$

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. We thus need to compute the gradient closest to loss first before computing the gradient of the earlier layers. 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. SGD (Stochastic Gradient Descent). This is the simplest optimizer, it depend on only one



(A) Illustration of SGD falling into a local minima

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

FIGURE 3.6

hyperparameter, the learning rate λ (LR) and update the parameters θ following

$$\theta_{t+1} = \theta_t - \lambda \frac{\partial \mathcal{L}}{\partial \theta}\bigg|_{\theta_t} \tag{3.10}$$

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 3.6a.

2. Adam [55]. 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

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1)\frac{\partial \mathcal{L}}{\partial \theta} \tag{3.11}$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2)\left(\frac{\partial \mathcal{L}}{\partial \theta}\right)^2 \tag{3.12}$$

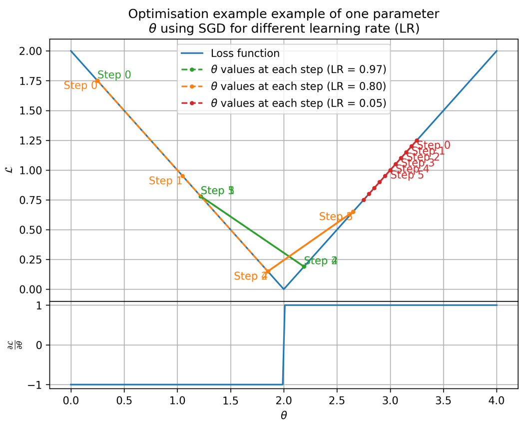
$$\theta_{t+1} = \theta i - \lambda \frac{m_{t+1}}{\sqrt{v_{t+1}} + \epsilon} \tag{3.13}$$

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 space as illustrated in figure 3.6a.

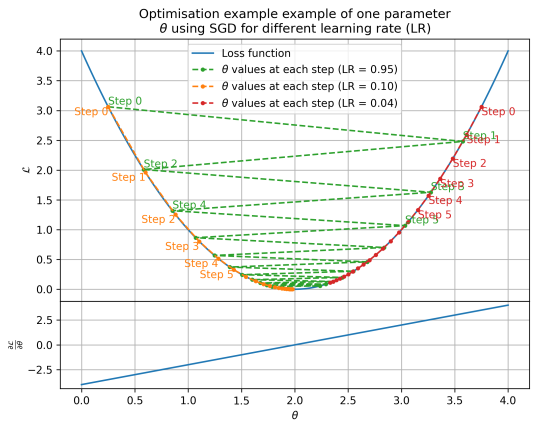
The LR is a crucial parameter in the training of NN, as illustrated in figure 3.7. To prevent possible issues, we setup scheduler policies.

Scheduler policies

Sometimes we want to update our hyperparameters or take a set of action during the training procedure. We use for this scheduler policies, for example a common policy is a decrease of the learning rate after each epochs. The reasoning is that if the learning rate is too high, the optimizer will continuously miss the minimum and oscillate around it (figure 3.7a). By reducing the learning



(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 MAE 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 3.7 – 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.

rate, we allow it to make more fine steps in the parameters phase space, hopefully converging to the true minima.

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.

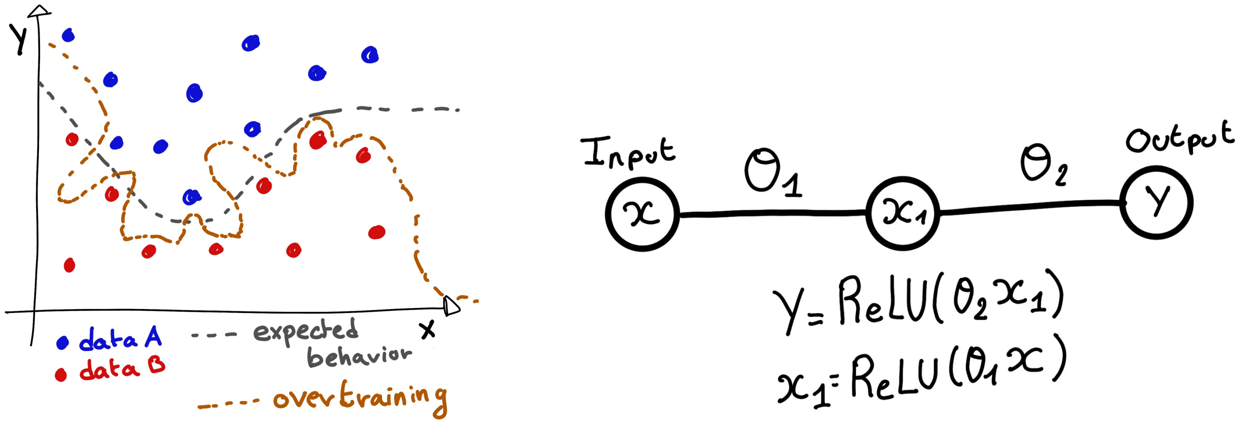
3.2.6 Potential pitfalls

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

Overtraining

This happen when the network learn the specificities of the training dataset instead of a more general representation of the underlying data distribution. This can happen if there is not enough data in comparison to the number of learning parameters, if the data contains some specific signatures specific to the training dataset or if trains for too long on the same dataset. This behavior is illustrated in figure 3.8a. 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 part of the neural network to 0. By doing this, we force the redundancy in its computing capability and, in a way, modify the data decreasing the possibility for specific learning.



(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 3.8

— **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 upper layer 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 3.8b.

The gradient for θ_1 will be computed using the chain rule presented in equation 3.9. 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 so that, let D be the data, $\langle D \rangle = 0$ and $\sigma D = 1$. This help the weight of the network to maintain an appropriate scale.

— **Residual Network (ResNet)** [65]: 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 ask each layer to calculate the residual of the input data. This technique is illustrated in figure 3.9.

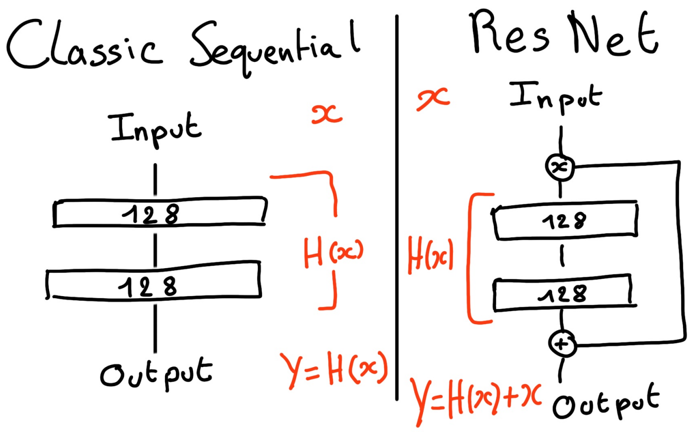


FIGURE 3.9 – Illustration of the ResNet framework

Gradient explosion

Gradient explosion happens when the consecutive multiplication of gradient cause exponential grow in the parameter value or if the training lead the network in part of the parameter space where the gradient is significantly higher than usual. 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 3.10 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 3.10 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.

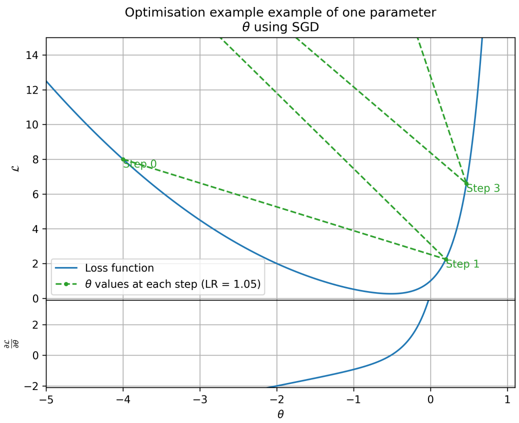


FIGURE 3.10 – Illustration of the gradient explosion. Here it can be solved with a lower learning rate but its not always the case.

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 - $\sigma_x \sigma_p \geq \frac{\hbar}{2}$

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 2.2.2), a thousands versus a few dozen, and their front-end electronics (see section 2.2.2).

They are complementary in their strengths and weaknesses and support each other, this is what we call *Dual Calorimetry*. One important point is their differences in expected resolution, the LPMT system outperform largely the SPMT system but is subject to effects such as charge non linearity [29] that could bias the reconstruction. Effects that the SPMT system is impervious to. This topic will be studied in more detail in chapter 7. Also, due to the dynamic range of the LPMT, in case of high energy and high density event such as core-collapse supernova, the LPMT system could saturate and the lower photo-coverage become a benefit.

Thus, although event reconstruction algorithm and physics analysis combines both LPMT and SPMT systems, individual approach are key studies to understand the detector and ensure their reliability. This topic will also be studied in more details in chapter 7. The subject of this chapter is to propose a machine learning algorithm for the SPMT reconstruction based on Convolutional Neural Network (CNN).

4.1 Motivations

As explained 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 arbitrary 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, while classical methods could be biased by the prior knowledge of the detector and physics processes. To study this potential phenomena, we will compare our machine algorithm to a classical reconstruction method developed for energy and vertex reconstruction [66].

We have access to a very detailed simulation of the detector (section 2.5) that will allow us to simulate arbitrary large dataset while giving access to the all the physics parameters of the event. Those parameters include the target of our reconstruction algorithms: the vertex and energy of our event. As introduced above, we hope that the ML algorithm will be able to used all the informations in the event, but that could lead that potential mismodelings in our simulation could be exploited by the algorithm. This specific subject will be studied in chapter 6.

4.2 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 on this sphere. 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 [57, 67–69] 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 [70] 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 2.6.3). The details about this planar projection will be discussed in section 4.2.2.

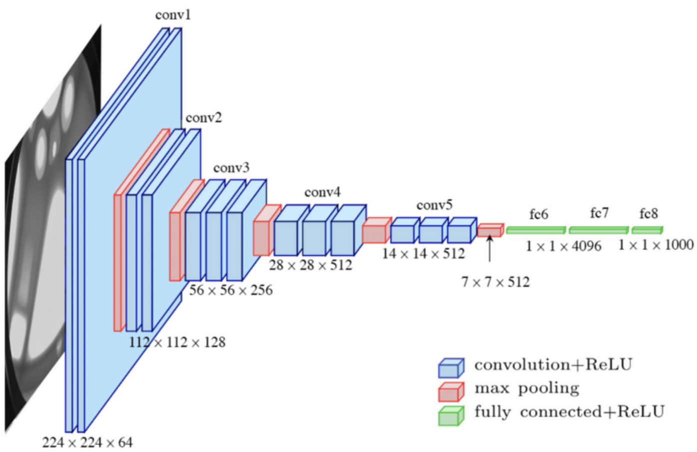


FIGURE 4.1 – Graphic representation of the VGG-16 architecture, presenting the different kind of layer composing the architecture.

4.2.1 Model

The architecture we use is derived from the VGG-16 architecture [57] 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 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 a sequence of fully connected linear layer using the PReLU activation function. For example $2 * 1024 + 2 * 512$ is the sequence of 2 layers with a width of 1024 followed by 2 other layers with a width of 512. Finally the last layer is a 4 neurons wide linear layers without activation function. Each neurons 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.

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

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

where R is the radius of the CD. With the energy in MeV and the distance in meters, we use the factor 0.85 and 10 to equilibrate 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.
- $(E_r + V_r)$ 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 $\frac{\sigma(E-E_{true})}{E_true}$ to be smaller at high energy than lower energy as illustrated in figure 2.22. 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.

Each combination of those hyperparameters (for example $(N_{blocks} = 2, N_{channels} = 32, \text{FCDNN} = (2 * 1024), \text{Loss} = (E + V))$), subsequently designated as configurations, is 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 :$-$ $E + V$
- "gen_1": $N_{blocks} = 4$, $N_{channels} = 64$, FCDNN configuration: $1024 * 2 + 512 * 2$, Loss :$-$ $E_r + V_r$
- "gen_2": $N_{blocks} = 5$, $N_{channels} = 64$, FCDNN configuration: $4096 * 2 + 1024 * 2$, Loss :$-$ $E + V$
- "gen_3": $N_{blocks} = 5$, $N_{channels} = 64$, FCDNN configuration: $4096 * 2 + 1024 * 2$, Loss :$-$ $E_r + V_r$

We cannot use the mean loss because we consider multiple loss functions, there is no guarantee that comparison of their numerical value will be meaningful. We use multiple observables to rank the performances of each configuration:

- 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.

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

— 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 [59] using NVIDIA A100 [71] and NVIDIA V100 [72] gpus. The A100 was split in two, thus the accessible gpu memory was 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 [73].

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

4.2.2 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 4.2. The x position is proportional to θ and the y position 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.

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

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

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

When two SPMTs are in the same pixel, the charges are summed and the lowest of the hit-time is chosen. 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 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 4.2 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 4.2.1, 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) [74] 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.

The second kind of pixel is pixel 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. 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.

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 4.5

4.2.3 Dataset

In this study we will discuss two datasets of one millions 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 algorithm presented in [66] was developed. This dataset is used as a reference for comparison to classical algorithm. The data in this dataset is *detsim* level (see section 2.5), where only the physic is simulated. The charge and time biases and uncertainties are implemented using toy MC adjusted using [26, 75]. The time window is not based on a selection algorithm but $t_0 :- t = 0$ is defined as the first PMT hit. The window goes up to $t_0 + 1000$ 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 2.5). Here the charge comes from the waveform integration, the time window resolution and trigger decision are all simulated inside the software. This dataset is more realistic and is used to confirm the performance of our algorithm.

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 ~ 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 monte-carlo 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.

Those events can be considered as "optimistic" as there is no pile-up with potential background or other IBD.

4.2.4 Data characteristics

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

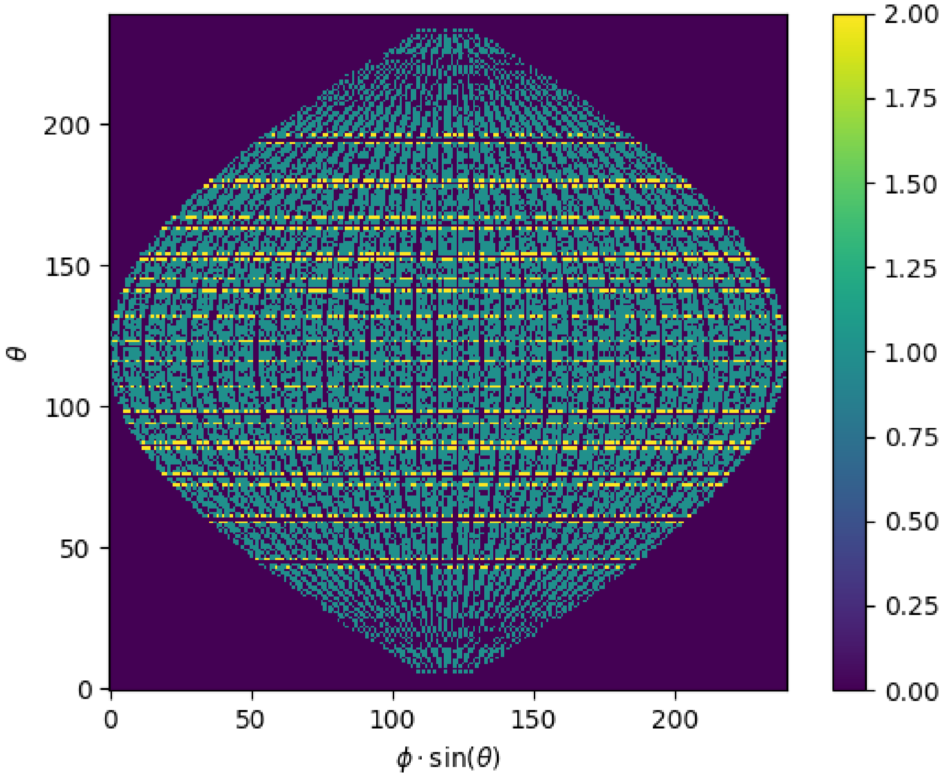


FIGURE 4.2 – Repartition of SPMTs in the image projection. The color scale is the number of SPMTs per pixel

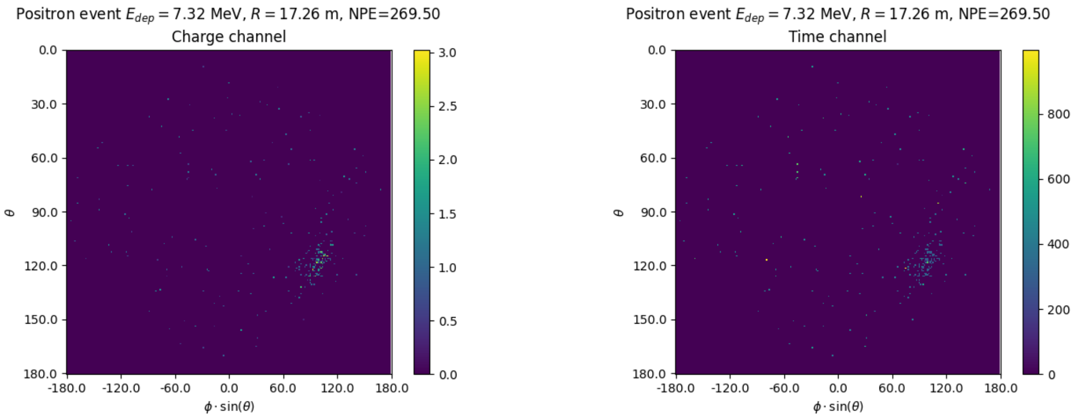


FIGURE 4.3 – Example of a high energy, radial event. We see a concentration of the charge on the bottom right of the image, clear indication of a high radius event. **On the left**: the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right**: The time channel in nanoseconds.

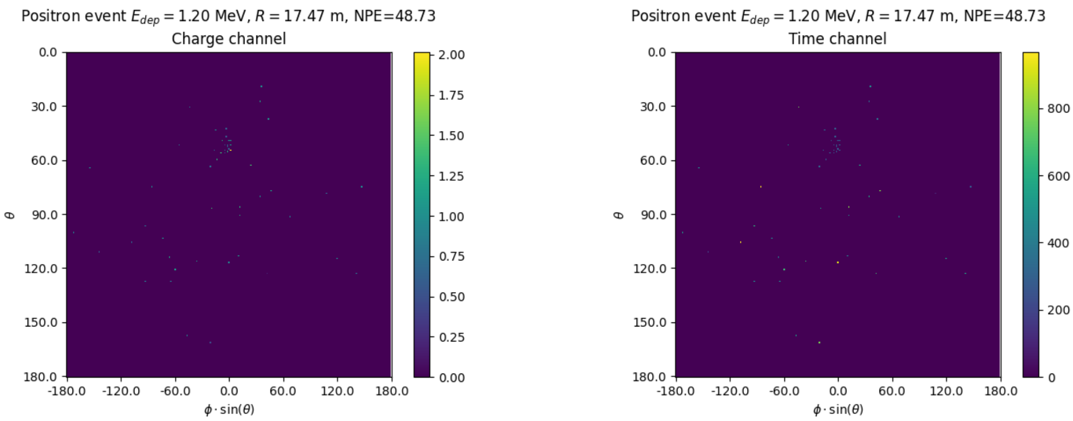


FIGURE 4.4 – Example of a low energy, radial event. The signal here is way less explicit, we can kind of guess that the event is located in the top middle of the image. **On the left**: the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right**: The time channel in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

In figures 4.3, 4.4, 4.5 and 4.6 are presented 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 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

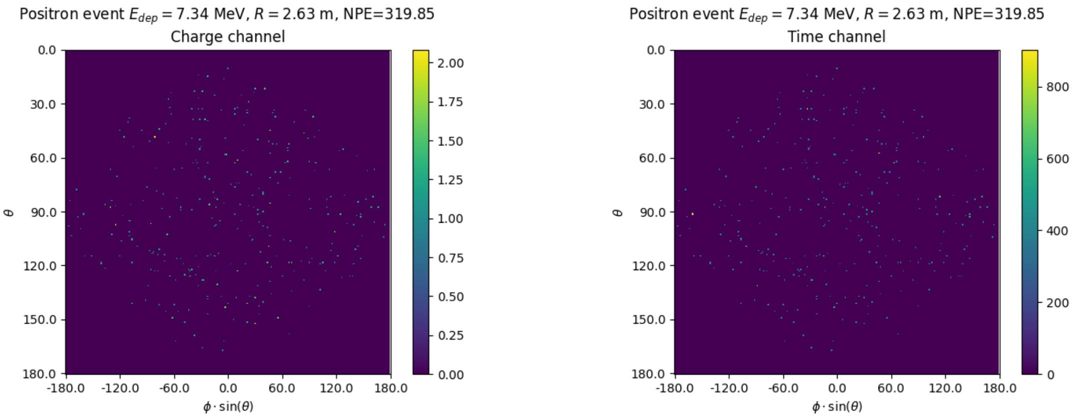


FIGURE 4.5 – Example of a high energy, central event. In this image we can see a lot of signal but uniformly spread, this is indicative of a central event. **On the left**: the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right**: The time channel in nanoseconds.

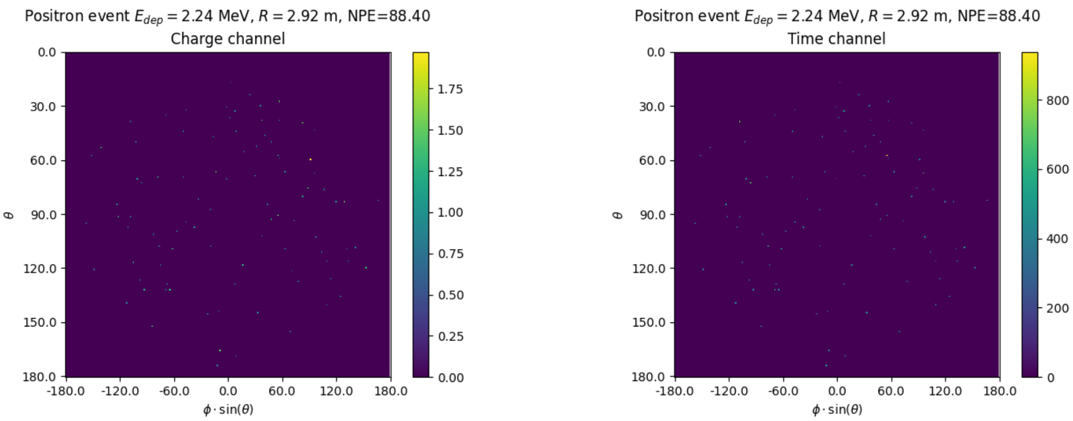


FIGURE 4.6 – Example of a low energy, central event. Here there is no clear signal, the uniformity of the distribution should make it central. **On the left**: the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right**: The time channel in nanoseconds.

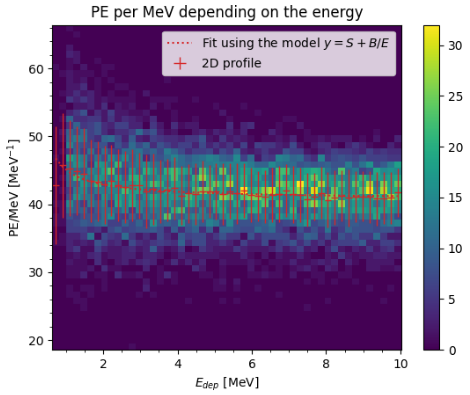
energy dependant, also the LS response is realistically energy dependant but figure 4.7a shows that we hare heavily dominated by statistical uncertainties which is why we are using this simple model.

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.

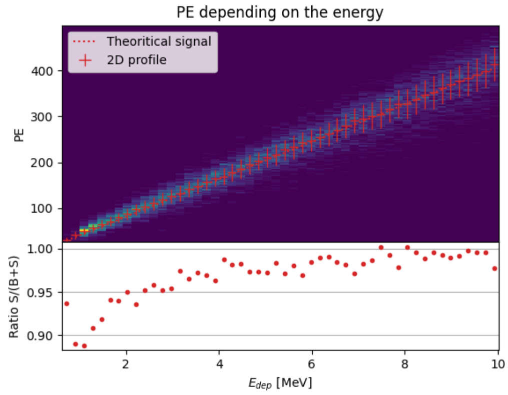
4.3 Training

The optimizer used for the training is the Adam [55] 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.

The training was composed of 30 epochs, each epoch constituted of 10k steps using a batch size of 64



(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

events. The validation was computed over a 100 steps on the validation dataset.

4.4 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. 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 2.6 and the possibility for the positron or the gammas to escape from the CD for near the edge events. We also expect dependency in 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:
- 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 absorption of scintillation photons by the LS and 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 2.6 is reconstructing E_{vis} while the ones presented in section 2.6.3 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.

4.4.1 J21 results

Those results comes from the "gen_30" model, meaning then 30th model generated using the table 4.1 or

— "gen_30": $N_{blocks} = 3$, $N_{channels} = 32$, FCDNN configuration: 2048 * 2 + 1024 * 2, Loss :$- E + V$

The performances of its reconstruction are presented in blue in figure 4.8. Superimposed in black is the performances of the classical algorithm from [66].

Energy reconstruction

By looking at the figure 4.8a and 4.8b, the CNN has similar performances in its energy resolution. Only at the end of the energy range does the resolution get a little better.

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).

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

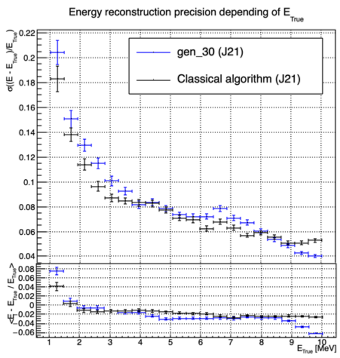
For the vertex reconstruction we do not study x, y and z independently but we use R as a proxy observable. Figure 4.9 shows the error distribution of the different vertex coordinates. We see that R errors and biases are slightly superior to the cartesian coordinates, thus R is a conservative proxy observable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

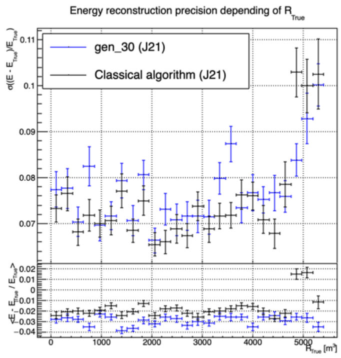
Radius reconstruction is worse than the classical algorithms in all configuration. In energy, figure 4.8c, where we see a degradation of almost 20cm over the energy range.

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.

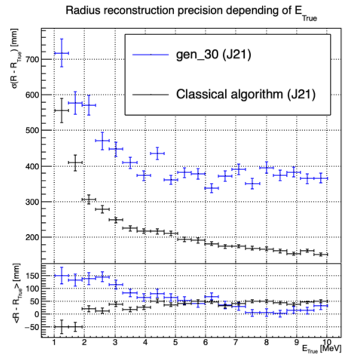
The precision also worsen when looking at the edge of the image $\theta \approx 0$, $\theta \approx 2\pi$ respectively the top and bottom of the image, and when $\phi \approx -\pi$ and $\phi \approx \pi$ respectively the left and right side of the image. This is the confirmation that the deformation of the image is problematic for the event reconstruction.



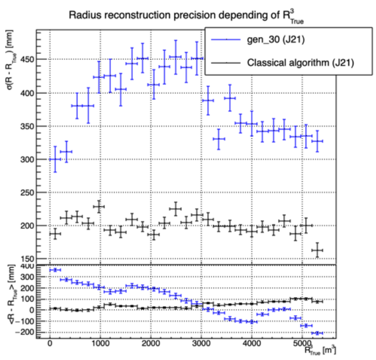
(A) Resolution and bias of energy reconstruction vs energy



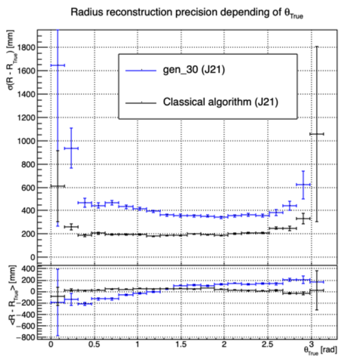
(B) Resolution and bias of energy reconstruction vs radius



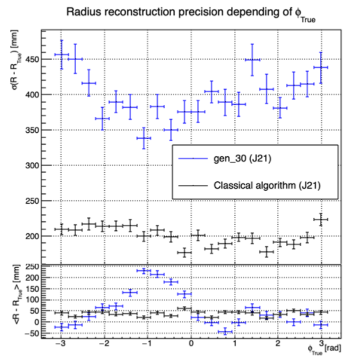
(C) Resolution and bias of radius reconstruction vs energy



(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 [66]. The top part of each plot is the resolution and the bottom part is the bias.

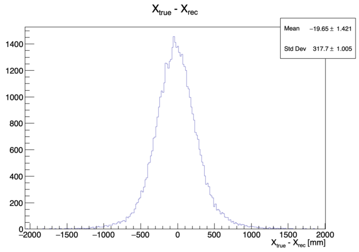
The bias in radius reconstruction is about the same order of magnitude depending of the energy but is of opposite sign. As for the energy, this behavior is studied in more details in section 4.4.2. Over radius, θ and ϕ the bias is inconsistent, sometimes event better than the classical reconstruction 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.

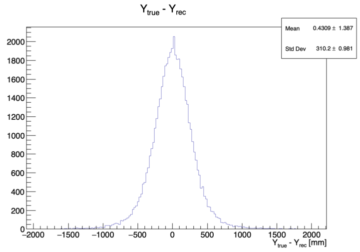
4.4.2 J21 Combination of classic and ML estimator

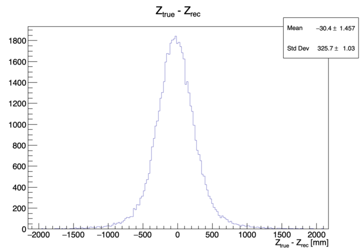
As it has been presented in previous section, there is instances where the reconstructed energy and vertex behaves differently between the neural network and the classic algorithm. For instance, if we look at figure 4.8c, we see that while the CNN tend to overestimate the radius at low energy while the 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 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.



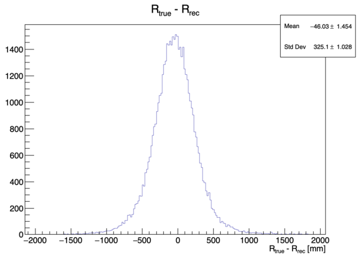


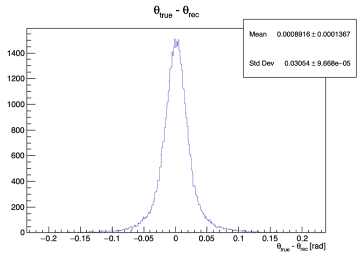


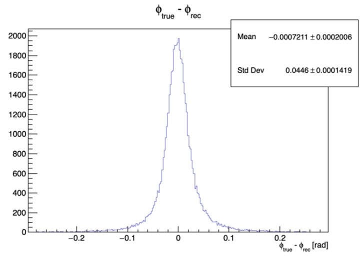
(A) Distribution of the error on reconstructed x by "gen_30"

(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 – Error 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.

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

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

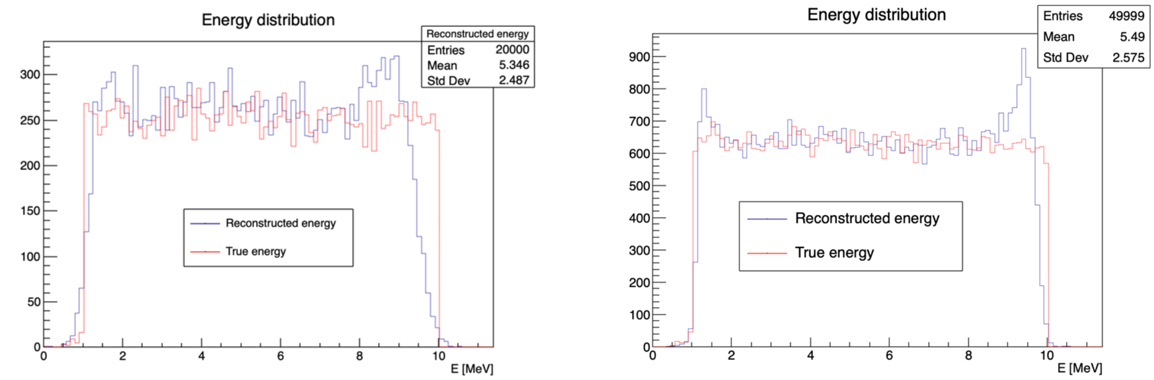
then the variance and mean would follow

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

$$\begin{aligned} \mathrm{Var}[\hat{\theta}] &= \frac{1}{4}\sigma_N^2 + \frac{1}{4}\sigma_C^2 + 2 \cdot \frac{1}{4} \cdot \sigma_{NC} && (4.13)\\ &= \frac{1}{4}\sigma_N^2 + \frac{1}{4}\sigma_C^2 + \frac{1}{2} \cdot \sigma_{NC} && (4.14)\\ &= \frac{1}{4}\sigma_N^2 + \frac{1}{4}\sigma_C^2 + \frac{1}{2} \cdot \sigma_N \sigma_C \rho_{NC} && (4.15) \end{aligned}$$

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 $\rho_{NC} \lesssim 1$ then we can gain in resolution.



(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

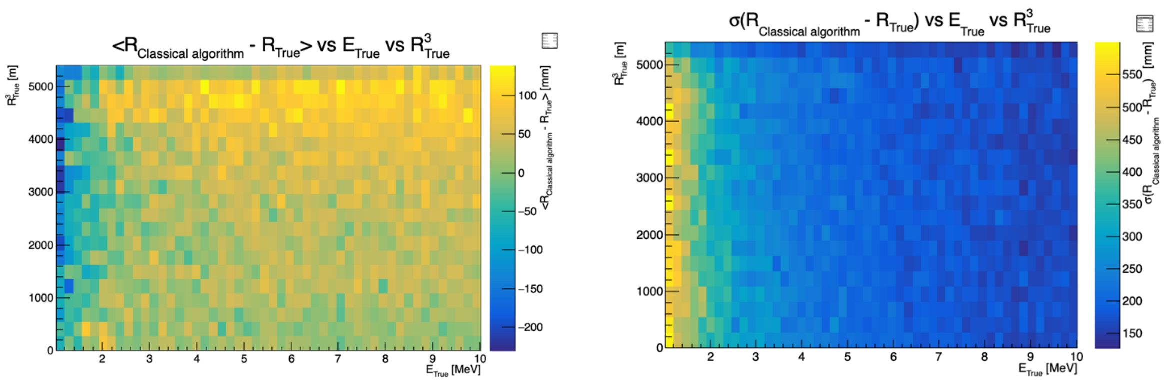


FIGURE 4.11 – Radius bias (**on the left**) and resolution(**on the right**) of the classical algorithm in a E, R^3 grid

By generalising the equation 4.9 to

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

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

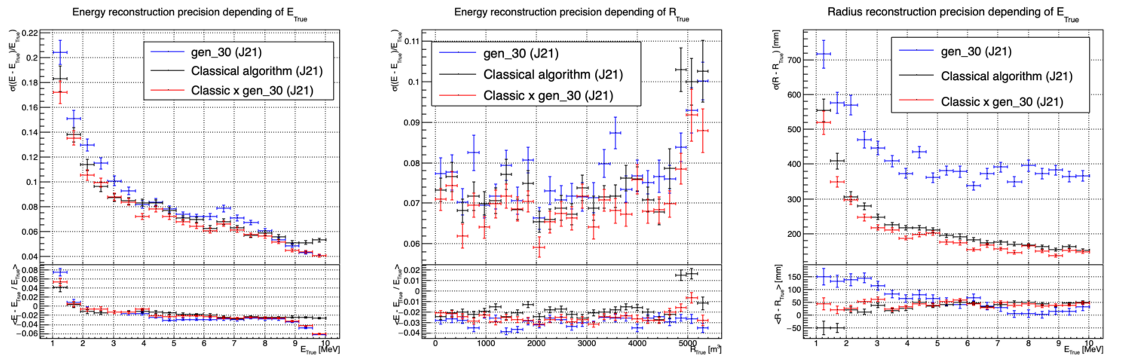
$$\alpha = \frac{\mu_C}{\mu_C - \mu_N} \tag{4.18}$$

See annex A for demonstration.

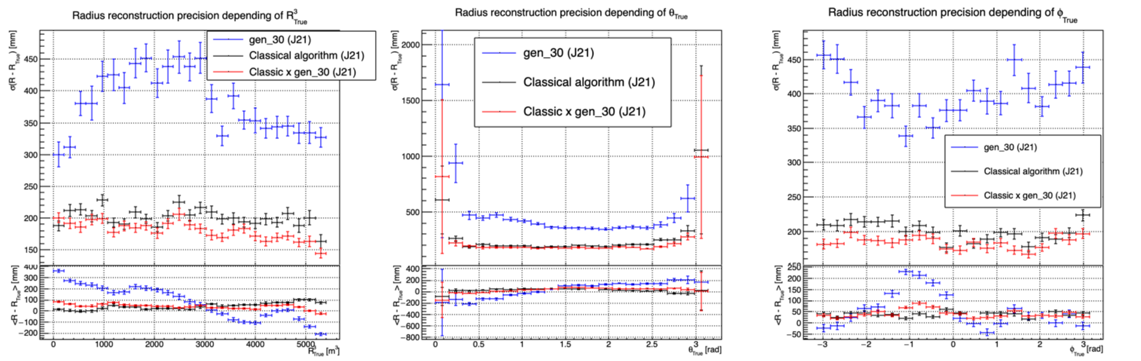
Its pretty clear from the results shown in figure 4.8 that the bias, variances and correlation are not constant across the (E, R^3) phase space. We thus compute those parameters in a grid in E and R^3 for the following results as illustrated in 4.11.

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.



(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 [66] and the combination of both using weighted mean. The top part of each plot is the resolution and the bottom part is the bias.

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 an excellent tool to indicate potential improvements to algorithms and reconstruction methods, showing with this results a potential upper limit to the reconstruction performances.

4.4.3 J23 results

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

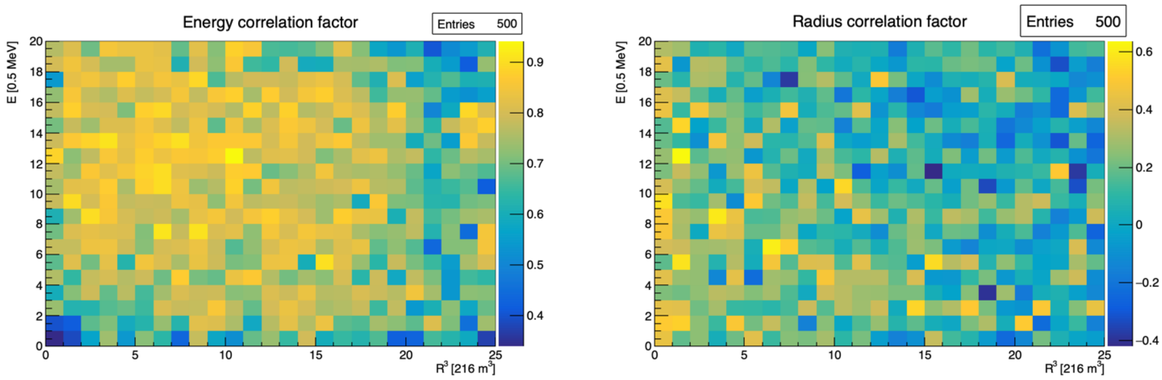


FIGURE 4.13 – Correlation between CNN and classical method reconstruction (**on the left**) for energy and (**on the right**) for radius in a E, R^3 grid

We retrained the models defined in 4.2.1 on the J23 data and used the same selection procedure. The results from the best architecture, "gen_42", are presented in figure 4.14. Following the table 4.1, "gen_42" is defined as:

— "gen_42": $N_{blocks} = 3$, $N_{channels} = 64$, FCDNN configuration: 4096 * 2, Loss :$- E + V$

Energy reconstruction

The results of the energy reconstruction are presented in figures 4.14a and 4.14b. Similarly to what we seen for J21, the resolution is close to the one of the classical algorithm with the exception of the start and end of the spectrum. This come from "gen_42" learning the shape of the distribution and pulling events from the extreme energies, like 1 and 10 MeV, to more common seen energy, like 2 and 9 MeV as illustrated in figure 4.10b. The bias disappear with the exception of low and high energy events.

Vertex reconstruction

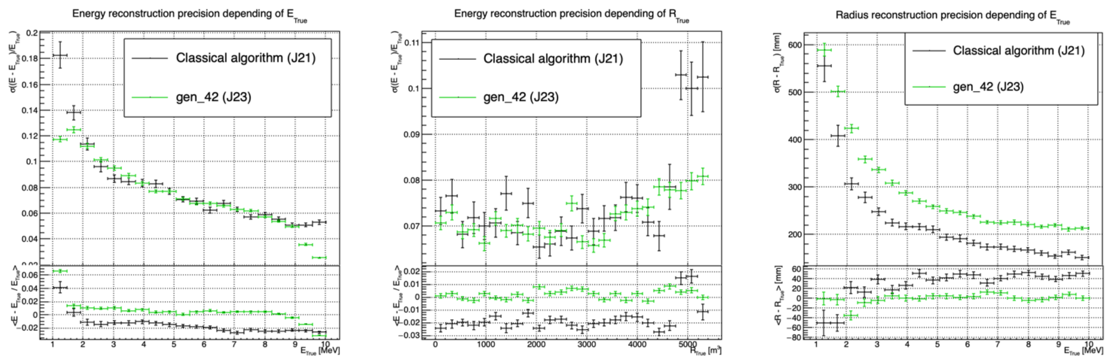
The vertex reconstruction, presented in figures 4.14c, 4.14d, 4.14e and 4.14f is not yet to the level of the classical reconstruction but the degradation is smaller than for "gen_32" being at most a difference of 15cm of resolution and closing to the performance of the classical algorithm in the most 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.

Unfortunately could not rerun the classical algorithms 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.

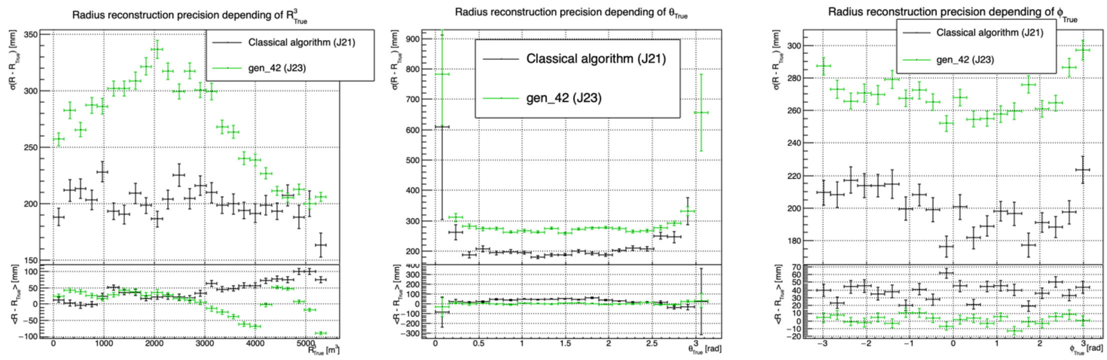
Overall the resolution improved over the transition from J21 to J23, effect probably coming from a more complete and rigorous simulation.

4.5 Conclusion and prospect

The CNN is a fine tool for event reconstruction in JUNO, and while the reconstruction performances are satisfactory, it show its limitation, the main one concerning the data representation. A lot of



(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 [66]. The top part of each plot is the resolution and the bottom part is the bias.

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 problematic of the aggregation of PMT time information and the meaning of the time channel in case of no 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.

In this thesis, we decided to solve those problem by moving away from the 2D image representation, looking into the graph representation and the Graph Neural Network (GNN). This is be the subject of the next chapter.

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

We previously showed, in chapter 4, that neural networks are relevant as reconstruction tools in JUNO. Even if they show worse performances, the combination with classical estimators could still bring improvements. We discussed the use of Convolutional Neural Network (CNN) in the previous chapter and their limitations, more specifically the limitation of the image as data representation for the experiment.

In this chapter we propose to use a Graph Neural Network (GNN), a Neural Network specialized to process graph as presented in section 3.2.3, to overcome those limitations.

5.1 Motivation

As explained in chapter 2 the JUNO sensors, the Large Photomultipliers (LPMT) and Small Photomultipliers (SPMT), are arranged on a spherical plane. When trying to represent this plane as a 2D image, due to the inherent problem of the projection, some part of the image are distorted and part of the image do not have any physical meaning (see section 4.2.2). A way to represent the data without inducing deformation is the graph, an object composed of a collection of nodes and edges representing the relation between the nodes.

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.

An approach was already proposed in JUNO by Qian et al. [42] where each nodes of the graph are like pixels, they represent geometric region of the detector and are connected with their neighbours. The LPMT informations are then aggregated on those nodes. The network then process the data using the equivalent of convolution but on graph [49].

In this work we want to take a step further in the graph representation by including the SPMT and including a maximum of raw informations.

5.2 Data representation

In an ideal world we would like to have every PMTs represented as node in the graph, each PMT being hit is an informations but the fact that PMTs were not hit is also an important information. It's by being aware of the whole of the system that we are able to give meaning to a subpart. As a reminder, in the Central Detector (CD), JUNO will posses 17612 LPMTs and 25600 SPMTs for a total of 43212 PMTs. This amount of information in itself is still manageable by modern computer if it were to be used in a neural network but when defining the relations between the nodes, it become a bit more tricky.

Excluding self relation and considering the relation to be undirected, the edge from A to B is the same from B to A, the amount of necessary edges is given by $\frac{n(n-1)}{2}$ which for 43212 PMTs amount for $933'616'866$ edges. If we encode an information with double precision (64 bits) in what we call an adjacency matrix, 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 not worth the trouble.

The option of connecting PMTs node only to their neighbours could be tempting to reduce the number of edge, but this solution does not translate well in term of internal representation in memory. Edges of sparsely connected nodes can be stored in efficient manner in a sparse matrix but the calculation in itself would often results in the concretization of the full matrix in memory, resulting in no memory gain during training.

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 densely connected to each other. We keep their number low to gain in memory consumption.
- — All the fired PMTs, that have been hit, will be represented as nodes. We call those node fired. Fired nodes are connected to the mesh they geometrically belong.
- — A final node which will hold global information about the detector and on which we will read the interaction vertex and energy. It's designated as the I/O node for input/output. This node will be connected to every mesh nodes.

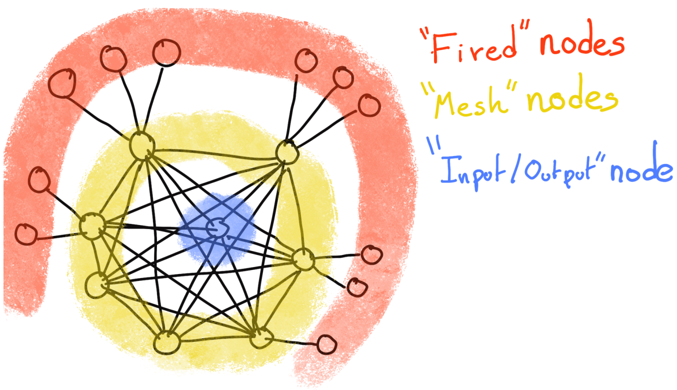
Those nodes and their relations are illustrated in figure 5.1a. From this representation, we end up with three distinct adjacency adjacency matrix

- — 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.

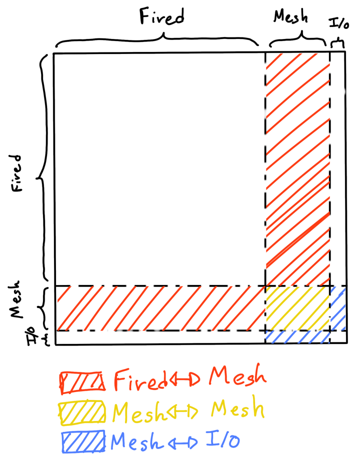
The adjacency matrix representing those relation is illustrated in figure 5.1b.

The mesh segmentation is following the Healpix segmentation [76]. 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 figure 5.3. Most of the features are low level informations such as the charge or time information but we include some high order features such as



(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

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

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 represent the likeliness of the interaction vertex to be on the segment between the center of two meshes.

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

$$\mathbb{B}_{ij} = \frac{Q_i}{Q_2} \left(\frac{l_2}{l_1}\right)^2 \tag{5.2}$$

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

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

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.

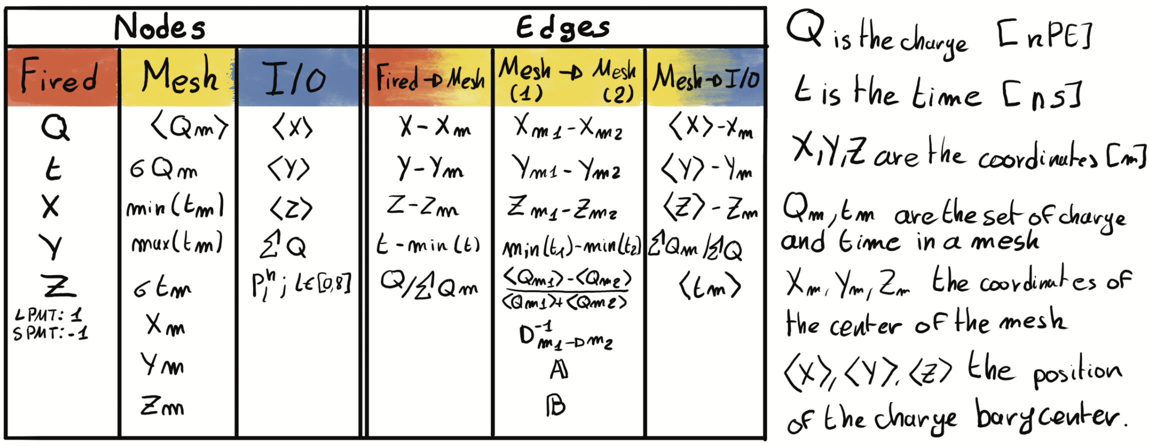


FIGURE 5.3 – Features held by the nodes and edges in the graph. $D^{-1}_{m_1 \to m_2}$ is the inverse of the distance between two mesh center. The features P_l^h, $\mathbb{A}$ and $\mathbb{B}$ are detailed in section 5.2

Because our different nodes do not have the same number of features, they live in different spaces. Most library and public algorithms available are designed with node living in the same space in mind, we thus had to develop a custom message passing algorithm.

5.3 Message passing algorithm

As introduced in previous section and in figure 5.3, 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

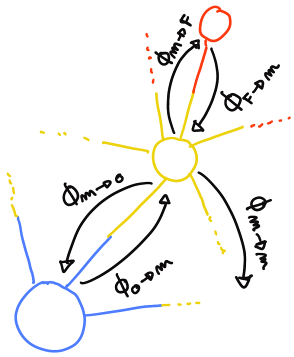


FIGURE 5.4 – Illustration of the different update function needed by our GNN

a family. Using the notation presented in section 3.2.3

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

we need to define

$$\phi_{u;f\to m}\ \phi_{m;f\to m} \tag{5.6}$$
$$\phi_{u;m\to f}\ \phi_{m;m\to f} \tag{5.7}$$
$$\phi_{u;m\to m}\ \phi_{m;m\to m} \tag{5.8}$$
$$\phi_{u;m\to io}\ \phi_{m;m\to io} \tag{5.9}$$
$$\phi_{u;io\to m}\ \phi_{m;io\to m} \tag{5.10}$$

to update the nodes after each layers as illustrated in figure 5.4. We would also need update function for the edges but for the sake of technical simplicity in this work, we will limit ourself to the nodes update. A wide variety of message passing algorithm exists, with different use cases and goal behind them. To stay generalist and to match to the best the specificity of our architecture, we implement the following algorithm:

$$\phi_u :- I_{i'}^{n'} = I_i^n A_{i',e}^i W_n^{e,n'} + I_i^n S_n^{n'} + B^{n'} \tag{5.11}$$

using the Einstein summation notation. I_i^n is the tensor 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. $A_{n,e}^{n'}$ is the adjacency tensor, discussed in the previous section, representing the connection between the node i' and the node i, each connections holding the features indexed by e . The learnable weights are composed of:

- The tensor $W_n^{e,n'}$ which represent the passage from the previous feature domain n, the previous layer, to the current domain n', this layer, knowing the relation e.
- $B^{n'}$ which is a learnable bias tensor on the new features n'.
- $S_n^{n'}$ which can be viewed as a self loop relation where the node update itself based on the previous layer informations.

If a node have neighbours in different families, the different $I_{i'}^{n'}$ coming from the different ϕ_u are summed.

$$I_{i'}^{n'} = \sum_{\mathcal{N}} \phi_{u,\mathcal{N}} \tag{5.12}$$

where $\mathcal{N}$ are the neighbouring family and $\phi_{u,\mathcal{N}}$ the update function between the target node family and the neighbour $\mathcal{N}$ family.

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. 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.

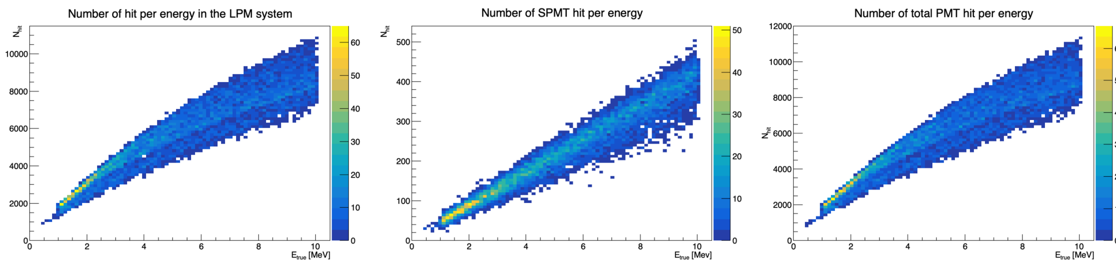
This operation of message passing is the constituent of our message passing layer, designed in this work as *JWGLayer*. To this layer, we can adjoin an activation function such as $PReLU$

$$I_{i'}^{n'} = PReLU\left(\sum_{\mathcal{N}} I_i^n A_{i',e}^{i} W_n^{e,n'} + I_i^n S_n^{n'} + B^{n'}\right) \tag{5.13}$$

5.4 Data

For this study we will be using a 1M positrons event dataset, uniformly distributed in energy with $E_k \in [0, 9]$ MeV and uniformly distributed in the detector. Those events come 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. 900k events will be used for the training, 50k for validation and loss monitoring and 50k for the results analysis in section 5.8. 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 5.5c.

As expected, by comparing the scale between the figure 5.5a and 5.5b 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 5.5a) and around 15 hits per event for the SPMT system (left limit of figure 5.5b) which is consistent with the results show in section 4.2.2.

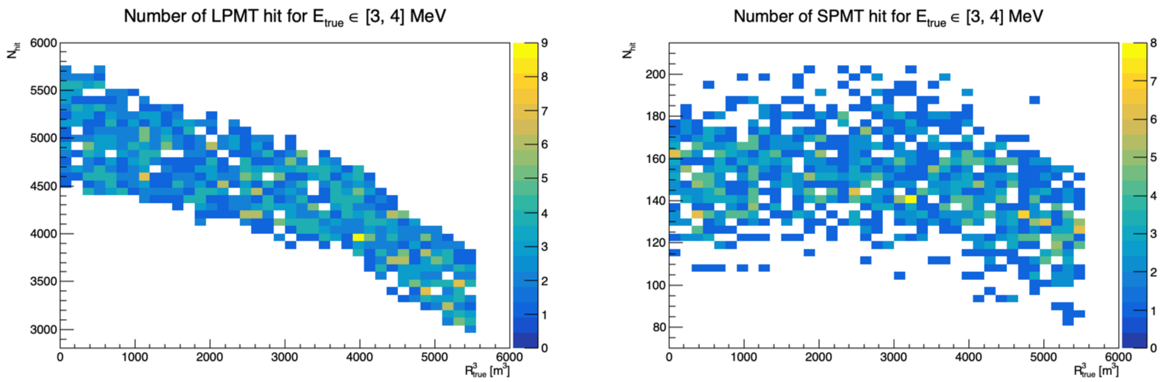


(A) (B) (C)

FIGURE 5.5 – 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 5.5a comes from the shape of the number of hits depending on the radius as shown in figures 5.6a and 5.6b 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 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.



(A) (B)

FIGURE 5.6 – 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 $E_{true} \in [0, 9]$.

5.5 Model

In this section we'll discuss the different layer composing the final version of the model. As introduced above, 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 precedent 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 figure 5.3, 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.
- A final, fully connected layer of 4 neurons acting as the output of the network.

A schematic of the model is presented in figure 5.7.

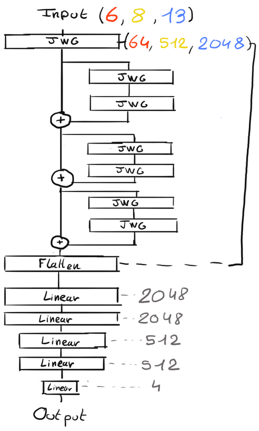


FIGURE 5.7 – Schema of the JWGv8.4.0 architecture, the colored triplet is the graph configuration after each JWG layers

5.6 Training

The optimizer used for training is the Adam optimizer and default hyperparameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$) with a learning rate λ = 1e-8. The training last 200 epochs of 800 steps. We use a batch size of 8. The learning rate is constant during the first 20 epochs then exponentially decrease with a rate of 0.99. The model saved is the model with the best validation loss during the training. The validation is computed over a single batch.

5.7 Optimization

Due to the extensive training time, up to 90h per training on the more complex architectures, and the heavy memory consumption of the models that would often exceed the 20GB limit of the V100, random search was not a realistic approach to the hyper optimisation. We were able to extend the memory limit to 40GB thanks to a local A100 GPU card available inside the laboratory.

The hyperparameters optimization was thus done "by hand", by looking at the results of the previous training and tinker hyperparameters that seems to play a role in the training. During this process, the model went into some heavy refactoring. At the start, the message passing algorithm was not the one presented above but each ϕ_u and ϕ_m function were FCDNN. Due to problems of memory consumption and gradient vanishing we pivoted to the message passing algorithm presented above.

Even the features on the graph went under investigation. With the addition of high level observables to the mesh and I/O nodes and edge, there was too much possibility to test everything. We went with the decision to keep the raw observables in the fired and for the higher order observables we tried to take the one that would be difficult for the NN to reconstruct or at least would need multiple layer to reproduce. Basically, because the operation in the JWGLayer are linear operation, any variables dependent on order > 1 of the input would be candidates. This is why we introduce standard deviation, $\mathbb{A}$, $\mathbb{B}$ and P_l^h for example.

Substantial effort went to the data processing process, transforming JUNO files into understandable graphs, before the training. Due to the volatile nature of the graph features during the optimization, the current code do not take preprocessed data and compute the observables, adjacency matrix, etc... on the fly. This data processing is carried out on the CPU, using a worker pool to allow for multiprocess. The raw data are coming from ROOT file produced by the collaboration software, the Event Data Model (EDM) used internally by the collaboration [77] had to be interfaced to our code, interface maintained through the evolution of the collaboration software. For the harmonic power calculation, we migrated from the Healpix library to Ducc0 [78] for a more fine control of the multithreading.

Over the course of the project, the model went over more than 60 different configurations to end on the one presented in this chapter.

5.8 Results

The reconstruction performance of "JWGv8.4" are presented in figure 5.9 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 2.6.

We also present the results of the optimal variance combination of the two algorithm labelled as "JWG 8.4 x Omilrec" where the reconstructed target $th\hat{e}ta$ is the weighted sum of the result of the two estimator JWGv8.4 θ_J and Omilrec θ_O.

$$\hat{\theta} = \alpha\theta_J + (1-\alpha)\theta_O;\ \alpha \in [0, 1] \tag{5.14}$$

For more details about the combination and the computation of α, refer to annex A.2.

One thing that need to be addressed before discussing results is that the Omilrec algorithm do not reconstruct the deposited energy E_{dep} but reconstruct the visible energy E_{vis}. The difference between those two different observables comes from the event-wise and channel-wise non-linearity, presented in 2.3. The multiples energy observables are already discussed in section 4.4. For the following results, the systematic bias of Omilrec that appear due, to the comparison to E_{true} instead of E_{vis} is corrected using a 5th degree polynomial

$$\frac{E_{true}}{E_{rec}} = \sum_{i=0}^{5} P_i E_{true}^i \tag{5.15}$$

The fitted distribution and the corresponding fit is presented in figure 5.8. The value fitted for this correction are presented in table 5.1.

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 5.1 – Parameters of the 5th degree polynomial used to correct Omilrec reconstructed energy.

Overall, energy and radius resolutions are not on par with Omilrec. We see from the energy dependent energy resolution in fig 5.9a that our resolution is a bit more than twice the resolution of Omilrec and the combination brings no improvements. Same observation for the energy resolution depending on the radius.

The radius resolution, presented in the figures 5.9c, 5.9d, 5.9e and 5.9f is much worse than the Omilrec one. This comes a bit as a surprise, as the energy reconstruction is dependent on the vertex reconstruction to correct for the non-uniformity and non-linearity effect. This mean that either the GNN could outperform the classical methods if the vertex was correctly reconstructed,

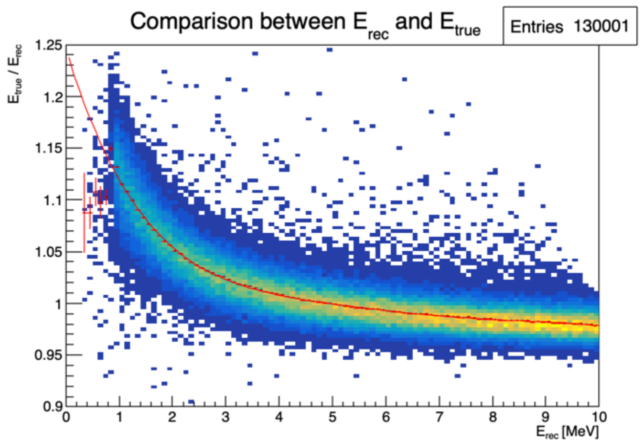


FIGURE 5.8 – Comparison between Omilrec E_{rec} and the true energy E_{true}. The profile of the distribution E_{true}/E_{rec} vs E_{rec} is fitted with a 5th degree polynomial.

or that somewhere the GNN reconstruct the vertex correctly but has trouble to formulate it in x,y,z coordinates on the latest layer.

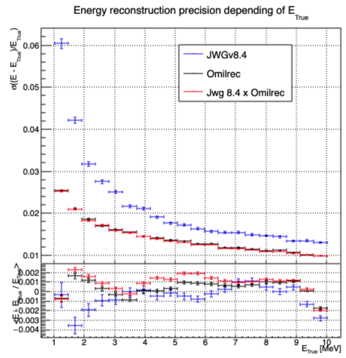
The GNN behaviours are close to Omilrec, indicating that the same information is used in the same way by both algorithms, just that the GNN seems to be less fine-tuned than Omilrec. If the precedent reasoning is true, it would mean that by adding more parameters, more layer or a higher pixelisation of the Healpix representation, the GNN could reach Omilrec performances.

5.9 Conclusion

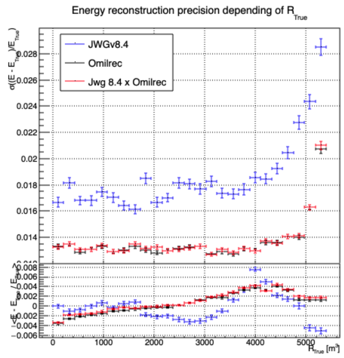
In this chapter, I present a proposition for a GNN architecture to reconstruct the energy and position of the prompt signal of an IBD interaction. The GNN is not competitive in terms of resolution with the more classical method Omilrec, which is the state of the art reconstruction method for IBD in the JUNO collaboration, but show encouraging results that could be exploited by going further in the optimisation of the hyper parameters. The message passing algorithm is still pretty naive and could probably be refined for JUNO's need.

Another possible improvement is to find a way to increase the Healpix pixelisation. Through our different work on reconstruction and by looking at the different classical methods, it seems that the time information is crucial for the vertex reconstruction, and thus for the energy reconstruction. While we are keeping every raw informations about the fired PMTs, it is possible that the aggregation on mesh nodes could cause the information loss and it has been noticed that allowing more channels to the hidden layer mesh nodes improve the resolution. This observation can be compared to the convolutional GNN presented section 2.6.3 that has similar performance with the classical method with an order 5 Healpix segmentation resulting in 3072 pixels, comforting the need of a finer pixelisation, or more parameters dedicated to aggregation through an increase of channels on the mesh nodes. Both of those improvements require some heavy memory optimisations, distributed training or more powerful hardware to address the memory consumption issue.

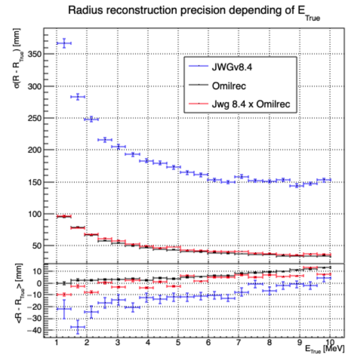
A final possible improvement would be to go further in the proximity of raw information. The charge and time used in the PMTs are extracted from a waveform, we could imagine a world where the full PMT waveform in the trigger window would be set of channels on the PMT node.



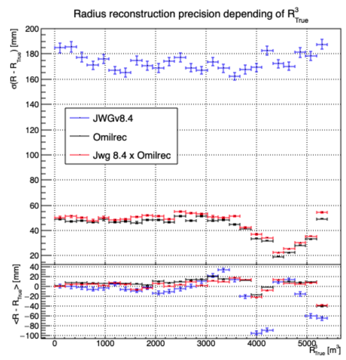
(A) Resolution and bias of energy reconstruction vs energy



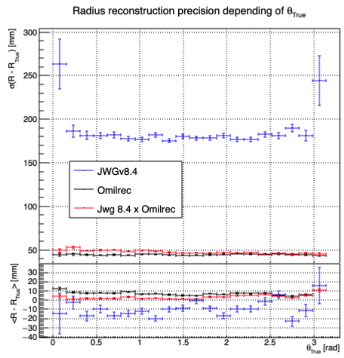
(B) Resolution and bias of energy reconstruction vs radius



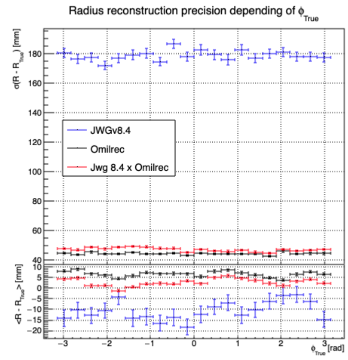
(C) Resolution and bias of radius reconstruction vs energy



(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 5.9 – Reconstruction performance of the Omilrec algorithm based on QTMLE presented in section 2.6, JWGv8.4 presented in this chapter and the combination between the two as presented in section 4.4.2. 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

Chapter 7

Joint fit between the SPMT and LPMT spectra

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

JUNO is an experiment of precise measurements, where we try to observe small fluctuation in the energy spectrum and with the goal to achieve sub-percent precision on the oscillation parameters measurement. A precise and complete understanding of the reconstruction and detector effects is thus 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. The ability to compare the results of the same experiment with two systems is thus extremely precious, this is the origin the dual calorimetry with the LPMT and SPMT system.

The resolution and bias of the reconstruction needs to be extremely well characterized: the target resolution of 3% [50] is unprecedented and is necessary to be able to distinguished between Normal Ordering (NO) and Inverse Ordering (IO). The non-linearity uncertainty needs to be constrained under 1% as exceeding this value, the risk appear to measure the wrong ordering [27].

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. The dual calorimetry can address this issue, using calibrations methods and measurements that will be employed to correct it [27].

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 systems.

The study of the independent results of the two system can provide some informations [79] but this is missing the important correlation that should be present between the two systems: they see the same events, in the same scintillator, they're bound to be correlated. We explore in this chapter a preliminary study of the impact of those correlations via multiple methods and the impact of QNL at various degrees.

In the next section we will discuss the motivations behind this study. In section 7.2, I present the approaches and assumptions in this study. 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 7.5 and discuss the conclusions and perspectives in 7.6.

7.1 Motivations

7.1.1 Discrepancies between the SPMT and LPMT results

As discussed in the introduction of this chapter, the SPMT and LPMT systems will observe the same events. This mean that, after calibration, if the two system show significant differences in their results this is the signal of potential overlook of an effect or problem. Being able to detect such differences is thus crucial, as discussed above, even the smallest deviation from our model could lead to the impossibility to measure the Mass Ordering (MO) or even worse, wrong our measurement.

The two systems are expected to have the same sensitivity to the oscillation parameters θ_{12} and Δm^2_{21} [11]. We will thus rely on the measurement of those two parameters to detect potential discrepancies.

We could just look at the value and compare them to the estimated independent error of the two system, but we believe and will demonstrate in this chapter that the independent study of the two system is missing a lot of informations, and that, by taking into account the statistic and systematic correlations between the two systems, we can produce much more powerful statistical tests.

Our work in this chapter is to develop such tools. The first step is, of course, to verify that in the case of no discrepancies, the results are coherent with the independent analysis. This will give us the distribution of those statistical test in absence of discrepancies. When we will have real data, we will be able to compare it to those distributions to compute a p-value characterizing the absence of those potential discrepancies.

To evaluate the power of our methods, we need to simulate a concrete difference between the two spectra. We have decided to study a plausible effect, the Charge Non-Linearity (QNL) that is detailed next section. But the goal of those tools is to be discrepancy agnostic, as those discrepancies could come from a variety of source (calibration issue, insufficient simulation tuning, etc...)

7.1.2 Charge Non-Linearity (QNL)

The CD energy response is subject to two kinds of non-linearity, the first one is the LS response non-linearity, where the LS photo-production is not linear with the deposited energy as illustrated in figure 2.12a. The second one is the LPMT response non-linearity where the charge read from the LPMT is not linear with respect to the number of collected Photo-Electrons (PE) (see section 2.3).

The LS non-linearity comes from physic sources. Particle interactions in the LS will produce mainly scintillation light, as discussed in section 2.2.2, but will also produce some Cherenkov light (< 10% of the collected light). Both mechanisms possess intrinsic non-linearity, for the Cherenkov emission it depends on the velocity of charged particle velocity while the scintillation photon-yield follows a so-called Birk's law with a "quenching" effect depending on the energy and type of particle [16]. This results in am event-wise QNL.

The LPMT response non-linearity can come from sheer saturation when subject to a high photon rate inducing a gain non-linearity or come from readout effects such as electronic noise, overshoot, the integration time window and even the waveform algorithm. All of these effects result in a channel-wise QNL.

Precedent studies [27] suggest a model to emulate the non-linearity response that will be used in this work. We define the channel wise non-linearity that would be applied to each LPMT readout

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

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.

We also define an event-wise non-linearity characterized by

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

where E_{vis} is the visible energy that is collected by the detector and E_{true} is the true deposited energy. An example of the effect of such event-wise QNL is presented in figure 7.1.

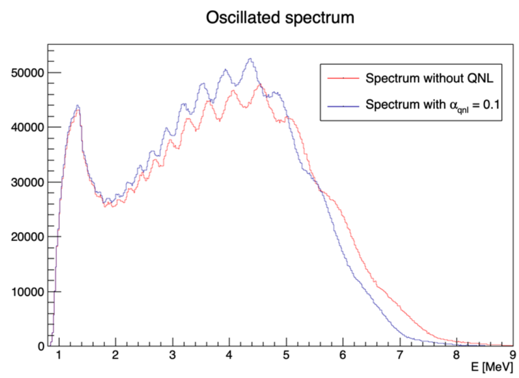


FIGURE 7.1 – Two oscillated spectra of 1e7 event expected in JUNO. In red the spectrum without supplementary QNL. In blue the same spectrum but where an event-wise QNL $\alpha_{qnl} = 10\%$ is introduced.

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.

In figure 7.2a we show the distribution of the ratio $\frac{Q_{rec}}{Q_{true}}$ for central events ($R < 4$m) and different values of γ_{qnl}. In figure 7.2a, we show the mean of this distribution as a function of the energy. We also present the effective α_{qnl} for each value of γ_{qnl}. We observe that using the event-wise QNL is equivalent to the mean behavior of using channel-wise QNL.

When using channel-wise non-linearity, we need to simulate a number of PE per LPMT, the process can be quite tedious if we want a realistic simulation. So in this study we are only using event-wise non-linearity to make the process simpler. This event-wise non-linearity will be characterized by α_{qnl} in this work.

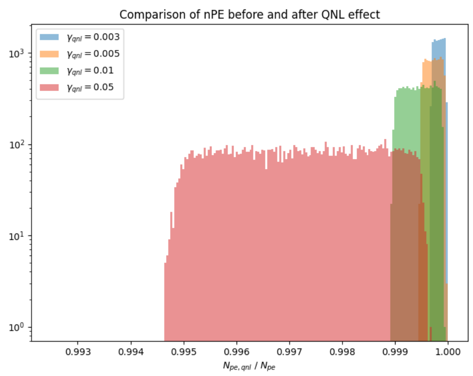
7.2 Approach

In this section, we detail the testing procedure for each of our tools.

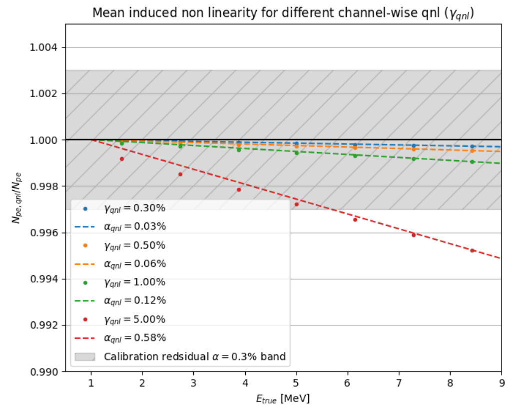
7.2.1 Data production

IBD spectra

The first step involves generating the data on which we our tools will be tested. In this study we use Monte-Carlo toys. For each toy we generate a $\bar{\nu}_e$ energy spectrum from the Taishan, Yangjiang and Dayabay nuclear power plants, the reactors used as source for the NMO analysis. The reactors



(A) Distribution of ratio of collected nPE after the additional QNL over the number of nPE that would be collected for different γ_{qnl}. We select event with an interaction radius $R < 4$m to not be affected by the non-uniformity.



(B) Ratio of collected nPE after the additional QNL over the number of nPE that would be collected at different energies. We select event with an interaction radius $R < 4$m to not be affected by the non-uniformity. The dots represent the mean of the distributions in figure 7.2a 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 [29].

FIGURE 7.2

parameters comes from JUNO official database, which shared among all physics analysis, the JUNO common inputs. This provides the initial spectra for the LPMT and SPMT systems. We then incorporate physic effects such as the LS non-linearity etc... (more details in section 7.3.1). Finally, we apply the reconstruction resolution for each system to their respective spectra, resulting in the final LPMT and SPMT spectra.

We will study the effect of exposure on our methods at different threshold: 100 days, 1 year, 2 year and finally 6 years which is the nominal data taking period for the NMO analysis.

These spectra are generated for different QNL, $\alpha_{qnl} = 0$ (no spectrum distortion) and for $\alpha_{qnl} \in \{0.01, 0.005, 0.003, 0.002, 0.001\}$. As a reminder, the calibration guarantees a residual event-wise non-linearity of $\alpha_{qnl} \leq 0.003$ [29].

The first test does not require any fitting, we are just comparing the LPMT and SPMT spectra using the expected statistical correlation matrix in the case $\alpha_{qnl} = 0$. For details about the generation of this correlation matrix, refer to section 7.5.2. This test is the spectrum χ^2 or χ^2_{spe}. In this test we compute a χ^2 representing the compatibility between the LPMT and SPMT spectra:

$$\Delta_i = h_{L,i} - h_{S,i} \tag{7.3}$$
$$U = AVA^T \tag{7.4}$$
$$\chi^2_{spe} = \vec{\Delta}^T U^{-1} \vec{\Delta} \tag{7.5}$$

Where $h_{L,i}$ and $h_{S,i}$ are the contents of the ith bin of the LPMT and SPMT spectra respectively. V is the covariance matrix of the LPMT + SPMT spectra. A is a transformation matrix defined as:

$$A_{ij} = \frac{\partial \Delta_i}{\partial h_j} = \frac{\partial (h_{L,i} - h_{S,i})}{\partial h_j} \tag{7.6}$$

Thus, $A_{ij} = 1$ if $i = j$, and $A_{ij} = -1$ if j is the SPMT bin corresponding to the i LPMT bin.

This χ^2_{spe} is minimal when the statistic between the bins of the LPMT and SPMT spectra follow the covariance matrix V. By looking at the distribution of this χ^2_{spe} when $\alpha_{qnl} = 0$ we can produce p-values for the values found when $\alpha_{qnl} \neq 0$.

Background spectra

The JUNO common inputs provide only LPMT background spectra. These background spectra are already smeared by the LPMT resolution and thus need to be regenerated to be smeared to account for the SPMT resolution. Fortunately the SPMT resolution is greater than that of the LPMT, allowing us to apply additional smearing to the spectrum using

$$S(E) = L(E) \star \frac{1}{\sqrt{|\Delta\sigma^2|}\sqrt{2\pi}} e^{-\frac{E^2}{2|\Delta\sigma^2|}};\ |\Delta\sigma^2| = \sigma_L^2 - \sigma_S^2 \tag{7.7}$$

Where $S(E)$ is the SPMT spectrum, $L(E)$ the LPMT spectrum, σ_L and σ_S the LPMT and SPMT resolution respectively. This formula is valid under the assumption that the LPMT and SPMT smearing are gaussian and that the LPMT and SPMT have the same bias. Those two assumptions are valid in the context of the IBD spectrum production as detailed in section 7.3.1. The demonstration of equation 7.7 can be found in annex C.

7.2.2 Individual fits

Each of the spectra, LPMT and SPMT, are then fitted individually with and without the presence of QNL over multiples toys. The results allow us to compute the correlation between the oscillations parameters measured by both of the systems when there is no QNL allowing us to compute a χ^2 representing the compatibility between the measurements of the systems. Because the SPMT system is not sensible to the oscillation parameters Δm^2_{31} and θ_{13}, the test is only done on the oscillation parameters θ_{12} and Δm^2_{21}. We can thus produce the individual chi square χ^2_{ind}

$$\Delta_\lambda = \lambda_L - \lambda_S \tag{7.8}$$

$$\vec{\Delta} = [\Delta_{\theta_{12}}\ \Delta_{\Delta m^2_{21}}] \tag{7.9}$$

$$U = AVA^T \tag{7.10}$$

$$\chi^2_{ind} = \vec{\Delta}^T U^{-1} \vec{\Delta} \tag{7.11}$$

where λ_L and λ_S are the measured parameters by the LPMT and SPMT systems respectively. The different λ considered are θ_{12} and Δm^2_{21}. V here is the 4×4 covariance matrix between the parameters $\theta_{12,L}$, $\Delta m^2_{21,L}$, $\theta_{12,S}$ and $\Delta m^2_{21,S}$. A is the transformation matrix that allow us to compute the covariance matrix de $\vec{\Delta}$ from V following

$$A_{ij} = \frac{\partial \Delta_i}{\partial j};\ i \in \{\theta_{12}, \Delta m^2_{21}\};\ j \in \{\theta_{12,L}, \Delta m^2_{21,L}, \theta_{12,S}, \Delta m^2_{21,S}\} \tag{7.12}$$

Same as described above, by comparing the distribution of this χ^2_{ind} when $\alpha_{qnl} = 0$ and $\alpha_{qnl} \neq 0$ we can compute the power of this test in term of p-values.

$\sin^2(2\theta_{12})$	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 7.1 – Nominal PDG2020 value [16]. All value are reported assuming Normal Ordering.

7.2.3 Joint fit

Standard joint fit

The final step is to produce a joint fit between the two spectra. In this case we adjust our model, the oscillated spectrum, over two spectra as the same time. We minimize a χ^2_{joint} defined over the two spectra, the LPMT and SPMT one

$$\Delta_i = D_i - T_i \tag{7.13}$$
$$\chi^2_{joint} = \vec{\Delta}^T V^{-1} \vec{\Delta} \tag{7.14}$$

where D_i is the content of the ith bin measured, from the data, and T_i is the theoretical number of event in this bin. V is the covariance matrix of our spectrum.

T is the fitted function and depend on multiple parameters

— The oscillation parameters θ_{12}, Δm^2_{21}, θ_{13} and Δm^2_{31}. Those parameters can be free, have a pull term or be fixed during the fit.

— We take into account in the data production the matter effect and parametrize it by the parameter ρ, the effective rock density between the reactors and the experiment. Same as the oscillation parameters, this parameter can be free, pulled or fixed.

— The exposure of the considered data which is just a normalization factor in front of the theoretical spectrum. This parameter is fixed at the start of the fit.

In the standard joint fit, the free parameters are $\sin^2(2\theta_{12})$, Δm^2_{21} and Δm^2_{31}. $\sin^2(2\theta_{13})$ is fixed to the PDG nominal value. For simplicity, we refer to $\sin^2(2\theta_{12})$ and $\sin^2(2\theta_{13})$ as θ_{12} and θ_{13} respectively.

Both of the LPMT and SPMT systems are sensitive to θ_{12} and Δm^2_{21}, thus these parameters are totally free and start at the PDG nominal value. Only the LPMT system is sensitive to Δm^2_{31}, we let it free so we can observe the effect of the deformation on it while the solar parameters θ_{12}, Δm^2_{21} are constrained by the SPMT system. To prevent Δm^2_{31} to take absurd value, we add a pull term using the PDG nominal value and errors. The PDG nominal values used in this study can be found in table 7.1.

$$\chi^2_{joint} = \vec{\Delta}^T V^{-1} \vec{\Delta} + \frac{\Delta m^2_{31} - \Delta m^2_{31,PDG}}{\sigma_{31,PDG}} \tag{7.15}$$

θ_{13} is the parameter on which we are least accurate. It's fixed to nominal value to prevent degeneracy (table 7.1).

The covariance matrix is produced from a correlation matrix C

$$V_{ij} = \sigma_i \sigma_j C_{ij} \tag{7.16}$$

where σ_i is the uncertainty on the number of event in the ith bin. We consider in this study that the content of each bin follow a Poisson statistic, thus the uncertainty is $\sigma_i = \sqrt{N_i}$ where N_i is the content of the ith bin. The bin content used for the uncertainty can come from two sources: the data and the theoretical spectra $\sigma_i = \sqrt{D_i}$ (Pearson test) and $\sigma_i = \sqrt{T_i}$ (Neyman test). Precedent studies have show that both Pearson and Neyman tests show bias at low statistic, we thus use the Pearson V test where

$$\chi^2_{joint} = \vec{\Delta}^T V^{-1} \vec{\Delta} + \frac{\Delta m^2_{31} - \Delta m^2_{31,PDG}}{\sigma_{31,PDG}} + \ln|V| \tag{7.17}$$

and the covariance matrix V is computed using the data spectrum for the uncertainty.

The estimation of the covariance is crucial in this study as the strength of this test rely on the systematic and statistical correlations between the LPMT and SPMT spectrum. The generation methods and results of this matrix is detailed in section 7.5.2.

Delta joint fit

Using the same structure we define a second joint fit, the Delta joint fit where, in addition to everything that was discussed above, we add two other parameters $\delta\theta_{12}$ and $\delta\Delta m^2_{21}$ and split the theoretical $T(\theta_{12}, \Delta m^2_{21}, ...)$ spectrum in two

$$\begin{aligned} T_{LPMT} &\equiv T(\theta_{12} + \delta\theta_{12}, \Delta m^2_{21} + \delta\Delta m^2_{21}, ...) \\ T_{SPMT} &\equiv T(\theta_{12}, \Delta m^2_{21}, ...) \end{aligned} \tag{7.18}$$

If the there is no additional distortion between the LPMT and the SPMT spectra, the fit should converge to $\delta\theta_{12} = \delta\Delta m^2_{21} = 0$. By observing the dispersion of those parameters we can define the probability $P(\alpha_{qnl} = 0|(\delta\theta_{12}, \delta\Delta m^2_{21}))$ and use the median value of $(\delta\theta_{12}, \delta\Delta m^2_{21})$ when $\alpha_{qnl} \neq 0$ to define a p-value.

The last test we explore in this thesis is to fit the same spectrum with the Standard Joint fit, that we consider as the hypothesis without distortion H_0, and the Delta Joint fit, designated as the H_1 hypothesis. By looking at the dispersion of $\chi^2_{joint,H_0} - \chi^2_{joint,H_1}$ we can extract a sensitivity to potential distortion.

7.2.4 Data and theoretical spectrum generation

To implement the joint fit, we have technically two data spectra and two theoretical spectra. The data in this study are produced using an IBD generator *IBD gen*, see section 7.3.1. The theoretical spectrum are produced the same way as data spectrum but with much higher statistics, 10^7 events to compare with the $\approx 10^5$ events for 6 years statistic. The two spectrum, that we get as a collection of events, are binned in two histograms from 0.8 to 9 MeV of reconstructed energy with bins of 0.02 MeV each, resulting in 410 bins per spectrum. An illustration of the theoretical spectrum can be found in figure 7.3. The low number of events in the tail of the spectrum can cause instability due to the low statistic, we thus cut the spectrum at 7.5 MeV / 335 bins for the fit.

All the IBD spectra presented and used in this study are produced assuming Normal Ordering using the PDG nominal value [16] for the oscillation parameters. Those values are reported in table 7.1.

7.2.5 Limitations

In this work we are only working considering the statistical errors. We can ignore systematic effects, such as effects that would affect the neutrino spectrum or the background spectrum, as they are entirely correlated between the two systems. The details of those systematic effects can be found in [11].

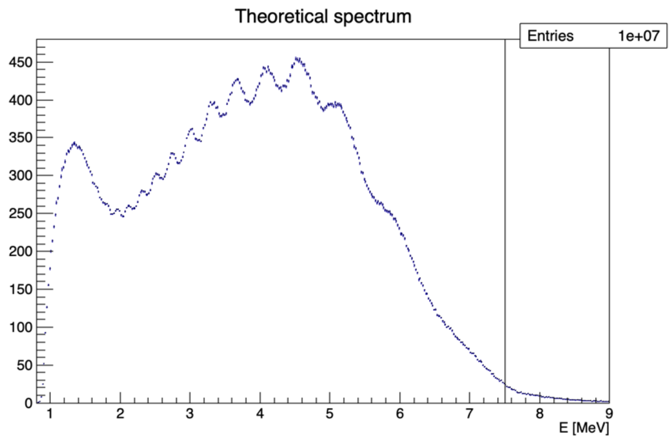


FIGURE 7.3 – Theoretical LPMT spectrum at nominal oscillation values binned using 410 bins from 0.8 to 9 MeV. It is rescaled to 6 years statistic. The black line represent the 335 bin cut

Most of our results assume decorrelated detection effects between the SPMT and LPMT systems. Their respective reconstruction effects are simulated using simple gaussian drawing on the resolution, independently from the event position. This approach was used in previous sensitivity and precision studies [11, 80]. The potential effect of those reconstruction effects and a first attempt to take them into account are explored in section 7.5.2.

Even if the goal of this work is to propose deformation agnostic tools, the QNL we use in this study is simplistic as we consider event-wise, position uniform deformation. We show in figure 7.2a and 7.2b that event-wise QNl is equivalent to the mean behaviour of channel-wise QNL but a more complete study would simulate channel-wise deformation for each event.

7.3 Fit software

In this section, I describe the ft framework that was used in this study. The software is composed of two parts as illustrated in figure 7.4: A standalone part composed of ROOT [81] macros, and the Avenue framework.

The Avenue framework is responsible for the spectrum and configuration reading, transforming the raw collection of events into spectra, managing the physics effect such as the oscillation and computing and minimizing the χ^2 with the help of the RooFit library. The macros are invoking, if necessary, the Avenue framework and are the entry point for fitting, generating the necessary inputs quantity such as the spectra and correlation matrix, analysing the fit results and managing jobs for distributed computing.

In this section we will focus on the IBD generator in section 7.3.1 and the fit macro in itself in section 7.3.2.

7.3.1 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 takes as inputs physics parameters and a collection of histograms, values and function provided by JUNO to its analysis groups, referred as the JUNO common inputs.

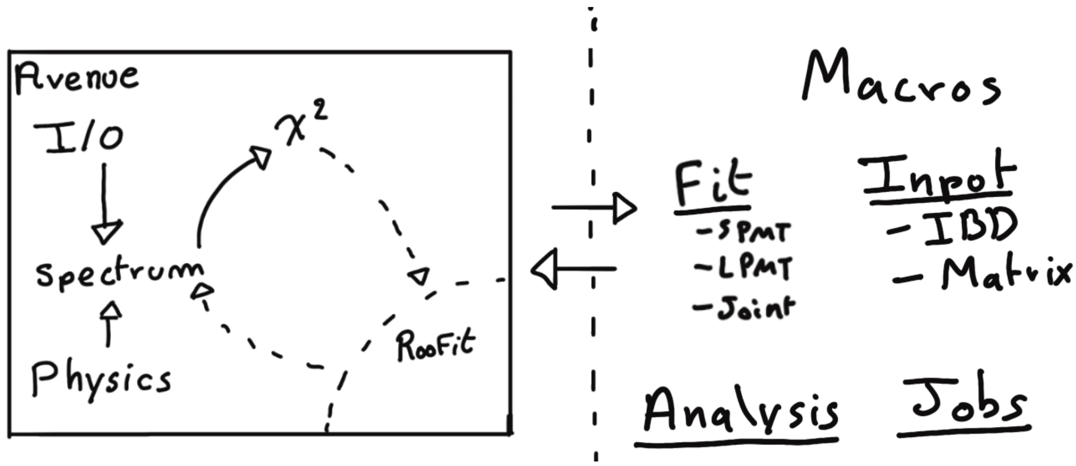


FIGURE 7.4 – Schematic description of the fit framework

Options allow to enable or disable effects such as non-uniformity and non-linearity. It finally 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 [82].
   (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 [83], 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 [29] 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

$$\frac{\sigma E_{vis}}{E_{vis}} = \sqrt{\left(\frac{A}{\sqrt{E_{vis}}}\right)^2 + b^2 + \left(\frac{c}{E_{vis}}\right)^2} \tag{7.19}$$

The relative and absolute resolutions of the LPMT and SPMT systems are illustrated in figure 7.5.

The events are stored as n-tuples and are not yet binned at the end of the generator.

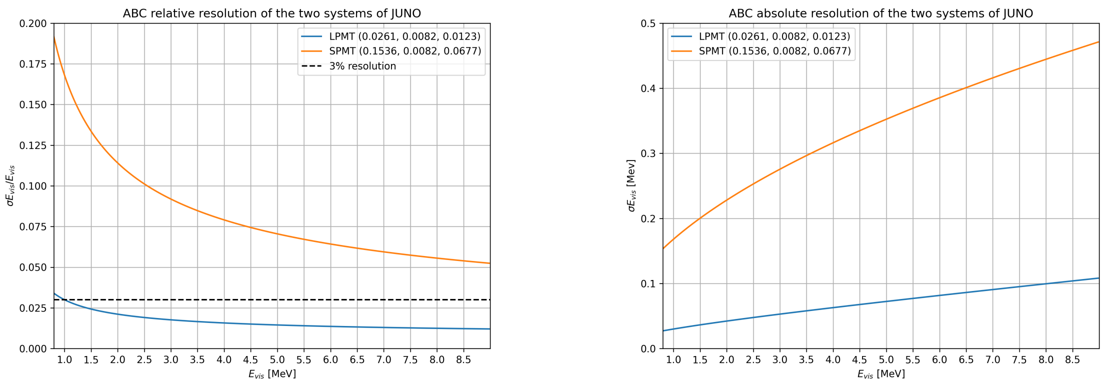


FIGURE 7.5 – Relative (**On the left**) and absolute (**On the right**) resolutions of the LPMT and SPMT systems used in this study. The number in parenthesis are the parameter A, B and C respectively for each systems.

7.3.2 Fit

The fit macro is the core of this fitting procedure. This macro is responsible for loading the fit configuration and setup the Avenue framework. Using Avenue, it will setup the data files, theoretical spectrum, choose the binning, χ^2, etc... It also have the possibility to generate toys on the fly based on the theoretical spectrum. Given this theoretical spectrum we can randomize the bin content either by:

1. Drawing the bin content in a Poisson distribution with the bin content as parameter.
2. Drawing the bin content in a Gaussian distribution with the bin content as mean and variance. The bin content is then rounded to the nearest integer.
3. Drawing the bin difference following a given covariance matrix using the Choleski decomposition. This matrix is at least the statistical covariance matrix but can also contain systematic uncertainties.

$$V = LL^T \tag{7.20}$$
$$\mathbf{R} \sim \mathcal{N}(0,1) \tag{7.21}$$
$$\tilde{\mathbf{h}} = \lceil \mathbf{h} + L\mathbf{R} \rfloor \tag{7.22}$$
$$\tag{7.23}$$

where V is covariance matrix used to produce the fluctuations, $\mathbf{R}$ is drawn in a multinomal distribution of mean 0 and variance 1, $\mathbf{h}$ the bin content of the theoretical spectrum and $\tilde{\mathbf{h}}$ the bin content of the generated toy.

The first two methods allow for the fast production of independent toys while the third allow for the production of statistical and systematical dependent toys. Unfortunately, none of those methods are fitted to produce toy with a QNL different from the theoretical spectrum. The uncertainty on the reconstructed energy σE_{rec} being dependent on E_{vis}/E_{α} makes that we would need to deconvolute the reconstruction effect from the theoretical spectrum. It is much easier to just produce those toys from the IBD generator.

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. 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 7.3.1, 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 Results

7.5.1 Validation

The first step is to confirm that the updated fit framework is able to reproduce existing results and that the joint fit behave as expected, meaning

— Without QNL, the individual (*LPMT* and *SPMT*) fit converge to the parameters nominal values and their errors are similar to the ones reported in existing analysis such as [11].

— The standard joint fit with an independent covariance matrix (*Indep Standard joint*), meaning that the covariance between the LPMT and SPMT spectra is 0, believe to have twice as much informations, and thus believe to have a grater precision than the individual fits.

— The standard joint (*Standard joint*) fit with a correlated covariance matrix has errors similar to the LPMT individual fit as the LPMT drive the precision on θ_{13} and Δm^2_{31} and that the LPMT as SPMT are expected to have close precision on θ_{12} and Δm^2_{21}.

— The delta joint (*Delta joint*) fit with covariance matrix have the same resolution as the standard joint fit. The supplementary parameter $\delta\theta_{12}$ and $\delta\Delta m^2_{21}$ should not bring supplementary precision.

The italicized name are the name used in the results reports to identify each fit. We also look into the *Indep Delta joint*, which is the Delta Joint fit but the covariance between the LPMT and SPMT spectra is 0, and the *Weighted* results where

$$\frac{1}{\sigma^2_{Weighted}} = \frac{1}{\sigma^2_{LPMT}} + \frac{1}{\sigma^2_{SPMT}} \tag{7.24}$$

We expect the weighted resolution to be similar to the *Indep Standard joint* as, in both of those test, we do not consider the correlation between the SPMT and LPMT results.

Asimov studies

We ran Asimov studies on the tests presented above on the updated framework, the results are reported in table 7.2. All those test are ran considering statistics error only, 6 years exposure with all backgrounds, Pearson χ^2 (covariance is estimated using data spectrum) and θ_{13} fixed to nominal value. For the *SPMT* fit Δm^2_{31} is fixed at nominal value as the SPMT system is net expected to be sensitive to this parameter.

	Δm^2_{21} error	$\delta\Delta m^2_{21}$ error	θ_{12} error	$\delta\theta_{12}$ error	Δm^2_{31} error	χ^2
LPMT	1.29936e-07		1.33852e-03		4.39399e-06	3.23088e-18
SPMT	1.38297e-07		1.38653e-03			2.87502e-18
Indep Standard joint	9.48731e-08		9.86765e-04		4.39212e-06	6.10592e-18
Standard joint	1.29723e-07		1.18342e-03		4.39287e-06	3.38055e-18
Weighted	9.46966e-08		9.63002e-04			
Delta joint	1.35780e-07	3.43529e-08	1.38236e-03	1.46865e-04	4.39309e-06	3.38055e-18
Indep Delta joint	1.38297e-07	1.89391e-07	1.38653e-03	1.87830e-03	4.39241e-06	6.10592e-18
Fixed Δm^2_{21} and Δm^2_{31}						
Indep Standard joint			9.33082e-04			4.82955e-26
LPMT			1.27032e-03			2.58849e-26
SMPT			1.31070e-03			2.24106e-26
Weighted			9.12193e-04			
Fixed Δm^2_{31} and θ_{12}						
Indep Standard joint	8.97117e-08					6.10617e-18
SPMT	1.30734e-07					2.87522e-18
LPMT	1.23319e-07					3.23095e-18
Weighted	8.97066e-08					

TABLE 7.2 – Results of the Asimov studies on the updated framework. All results are Asimov fit, considering 6 years exposure, θ_{13} is fixed to nominal value, χ^2 is pearson meaning that he error is estimated using the data spectrum

In every cases presented above, the fit converges to the parameters nominal value thus only the errors are presented.

We observe, as expected, that $\sigma_{Weighted} \approx \sigma_{Indep\ Standard\ joint}$ with the exception of $\sigma\theta_{12}$. This could from the slight difference 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. This deficit is partially compensated by event outside the spectrum coming back in it but we expect very few event outside the spectrum in comparison to event at the edges of it. Thus the event deficit is not totally compensated. θ_{12} being mainly driven by the amplitude of the spectrum (see illustration 2.2), that's why we think this the origin of the difference.

The second observation is that $\sigma_{Standard\ joint} \approx \sigma_{LPMT}$. Once the covariance matrix between the LPMT and SPMT is correctly introduced, the fit "understand" that it does not have supplementary information and the LPMT system, which have the best precision, dominate the resolution.

Finally for the *Delta* fit, the error on $\delta\theta_{12}$ and $\delta\Delta m^2_{21}$ are of the same order of magnitude than the errors on θ_{12} and Δm^2_{21} in the absence of the covariance matrix. As the LPMT and SPMT spectra are not connected through the covariance matrix, the delta parameters are unconstrained thus the similar errors. Once the covariance matrix is introduced, the delta are much more constrained and show errors of an order of magnitude smaller than the error on their respective parameters.

Overall, the asimov studies are satisfactory. The joint fit behave as expected and the errors on the delta parameters are significantly smaller than the error on their respective parameters, indicating great potential if they converge to value too far from 0.

Toy studies

Once we validated that the asimov study is yielding coherent results, we study the behaviour of toy studies. The above asimov study was using the Pearson χ^2 (Eq. 7.13) without pull parameter. We show in figure 7.6 the effect of using a simple Pearson χ^2. We see that $\sin^2(2\theta_{12})$ (reported as θ_{12} for simplicity) is biased of about 0.5σ and Δm^2_{21} biased of about 0.1σ. When introducing the PearsonV χ^2 (Eq. 7.17) the bias disappear as reported in figure 7.7.

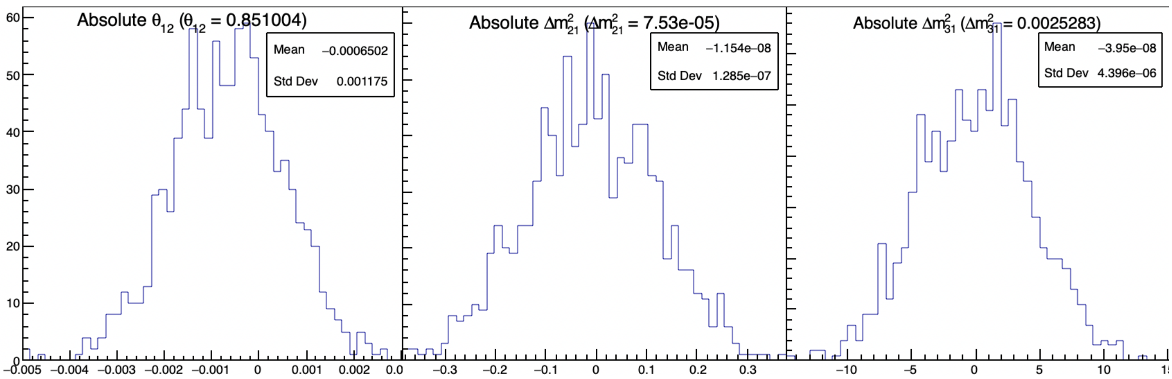


FIGURE 7.6 – Distribution of BFP - nominal value for 1000 toy Standard joint fit. 6 years exposure, all background, **Pearson** χ^2, θ_{13} fixed.

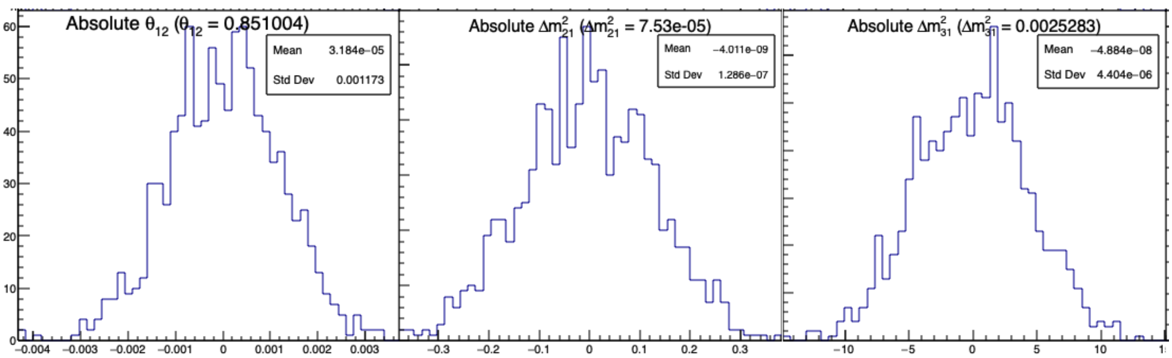


FIGURE 7.7 – Distribution of BFP - nominal value for 1000 toy Standard joint fit. 6 years exposure, all background, **PearsonV** χ^2, θ_{13} fixed.

When the supplementary parameters are introduced in the Delta Joint fit, the fit is stable as shown in the results figure 7.8. The resolutions on the oscillation parameters are slightly worse in the Delta joint fit due to the supplementary freedom. As seen in the asimov studies, the resolution of the δ parameters is an order of magnitude smaller than their respective parameters, indicating that they can be powerful tools to detect discrepancies between the SPMT and LPMT spectra.

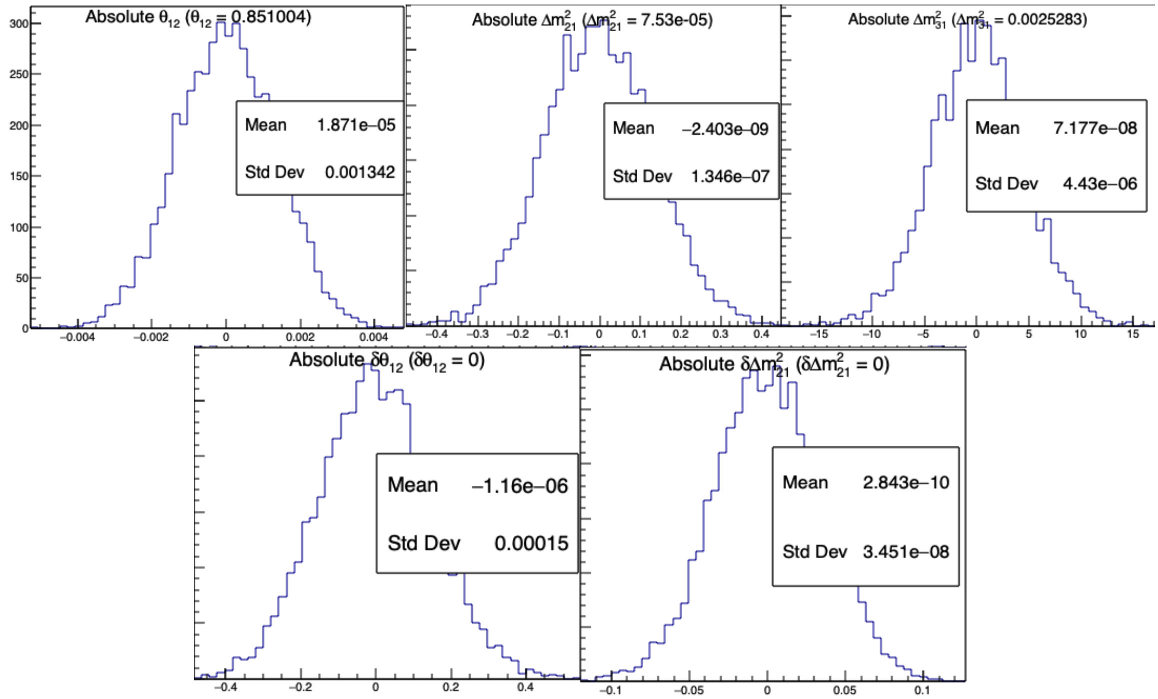


FIGURE 7.8 – Distribution of BFP - nominal value for 5000 toy Delta joint fit. 6 years exposure, all background, **PearsonV** χ^2, θ_{13} fixed.

Effect of supplementary QNL on the LPMT spectrum

Now that we know that the framework and joint fit behave correctly on unbiased data, we test the effect of introducing the QNL, as presented in Eq. 7.2, in the LPMT spectrum. To test the effect, we consider a QNL $\alpha_{qnl} = 1\%$. For reference, this is about three time the expected residual QNL after calibration ($\alpha_{qnl} = 0.3\%$ [29]). The background had to be removed as JUNO provide them already smeared, thus the introduction of supplementary QNL is not trivial, the resolution being dependent of E_{vis} which is affected by the QNL. We use a covariance matrix assuming no QNL. The effect of this QNL on the spectrum is illustrated in figure 7.9. In table 7.3 we report the results of the different scenarios.

Mean (std dev)	θ_{12} $[10^{-3}]$	Δm^2_{21} $[10^{-7}\text{eV}^2]$	Δm^2_{31} $[10^{-6}\text{eV}^2]$	$\delta\theta_{12}$ $[10^{-3}]$	$\delta\Delta m^2_{21}$ $[10^{-7}\text{eV}^2]$
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 7.3 – Results of the different fit scenarios on QNL distorted data $\alpha_{qnl} = 1\%$. The mean value are reported subtracted from their nominal value. For SPMT Δm^2_{31} is fixed at nominal value. The χ^2 is PearsonV. The correlation matrix used to fit assume no QNL in the spectrum.

The results in table 7.3 are subtracted from their nominal value, themselves reported in table 7.1. We clearly see the bias induced by $\alpha_{qnl} = 1\%$ when comparing the SPMT and LPMT results. The Indep Standard is, as expected, the mean value between the SPMT and LPMT: the fit having no informations about the correlation between the spectrum think it have two uncorrelated experiments thus report an in between value. When introducing the relationship between the LPMT and SPMT

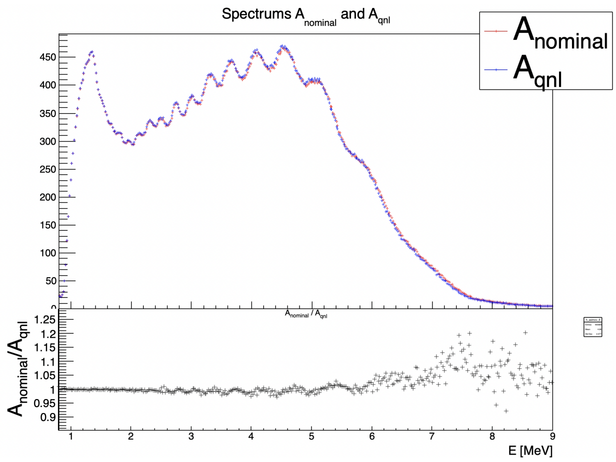


FIGURE 7.9 – **Top:** Theoretical spectrum without QNL (in red) and with $\alpha_{qnl} = 1\%$ (in blue). **Bottom:** Ratio between the theoretical spectrum with and without QNL.

spectra in the Standard fit, the joint fit cannot find a clean minima, it thus converge to a completely incorrect value.

Introducing the δ without the correlation in Delta Indep remove the bias and converge to the SPMT minima, the δ absorbing the deformation of the LPMT spectra.

Finally, with the δ and the covariance matrix, θ_{12} is unbiased, $\delta\theta_{12}$ absorbing the deformation. $\delta\Delta m^2_{21}$ is still heavily biased, even more than LPMT only, for the same reason than the Standard fit: the correlation make it difficult to converge to the nominal value.

Overall Δm^2_{31} bias is unchanged as the SPMT spectrum bring no information about the parameter. The δ are significant, naively up to 7.46σ for $\delta\theta_{12}$ in the Delta fit.

7.5.2 Covariance matrix

The covariance matrix between the LPMT and SPMT spectra is at the hearth of this study as it was already mentioned in section 7.2 and demonstrated in section 7.5.1. In this section we discuss the different approaches taken to estimate it. In this work we will mainly discuss the statistical covariance matrix between the two spectra, how the number of event in a LPMT bin influence the number of bin in the SPMT spectrim due to the resolution. We will still discuss the reconstruction effects, mostly due to non-uniformity, in on reconstruction correlation.

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 7.3.1, we consider the system resolution $\sigma(E)$ to be only dependent in energy. We do not consider the position of the event.

The first step is to compute the statistical uncertainty of the input spectrum while taking into account the smearing, considering no uncertainty on the smearing. For this, using the notation of section *39.2.5 Propagation of errors* of PDG2020 [16] and considering an extended spectrum of 820 bins following the binning scheme introduced in 7.5.3, the first 410 for the LPMT and the last 410, we consider

— $\boldsymbol{\theta} = (\theta_0, ..., \theta_n)$; $n = 820$ the content of the spectrum bins.

— $\boldsymbol{\eta}(\boldsymbol{\theta}) = (\eta_0(\boldsymbol{\theta}), ..., \eta_m(\boldsymbol{\theta}))$; $m = 820$ the set of smearing functions representing the PMT resolutions.

η_m can thus be defined as

$$\eta_i = \sum_j^n G(i, \sigma(E_i))(j)\theta_j \tag{7.25}$$

where $G(i, \sigma(E_i))(j)$ is the smearing function defined as

$$G(i, \sigma(E_i))(j) = \int_{\lfloor E_j \rfloor}^{\lceil E_j \rceil} \frac{1}{\sigma(E_i)\sqrt{2\pi}} e^{-\frac{(E_i - E)^2}{2\sigma(E_i)^2}} \, \mathrm{d}E \tag{7.26}$$

where E_i is the mean energy in the bin i and $\lfloor E_i \rfloor$ and $\lceil E_i \rceil$ are the lower and higher energy bound of the ith bin respectively.

We can then construct the transfer matrix A as

$$A_{ij} = \frac{\partial \eta_i}{\partial \theta_j} = G(i, \sigma(E_i))(j) \tag{7.27}$$

and then compute the first part of our covariance matrix

$$U = AVA^T \tag{7.28}$$

where V is the uncorrelated covariance matrix simply defined, under the assumption of poissonian statistic for the bin content,

$$V_{ij} = \sqrt{\theta_i \theta_j} \tag{7.29}$$

Now we just need to consider the uncertainty on the smearing $\sigma\eta_i$, considering no uncertainty on the unsmeared spectrum. From Eq. 7.25, the $G(i, j) \equiv G(i, \sigma(E_i))(j)$ are considered independents from each other $\forall i, j$. This mean that this covariance matrix is diagonal, we only need $\sigma G(i, j)$. We can derive this term from two equation:

— The term $G(i, j)\theta_j$ 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 $\sigma[G(i, j)\theta_j] = \sqrt{G(i, j)\theta_k}$.

— Using basic error propagation we can say that $\sigma^2[G(i, j)\theta_j] = \theta_j^2 \sigma^2 G(i, j) + G(i, j)^2 \sigma^2 \theta_j$.

Using $\sigma\theta_j = \sqrt{\theta_j}$ we derive

$$G(i, j)\theta_j = \sigma^2[G(i, j)\theta_j] = \theta_j^2 \sigma^2 G(i, j) + G(i, j)^2 \theta_j \tag{7.30}$$

$$\Rrightarrow \sigma^2 G(i, j) = \frac{G(i, j)\theta_j - G(i, j)^2 \theta_j}{\theta_j^2} \tag{7.31}$$

$$= \frac{(1 - G(i, j))G(i, j)}{\theta_j} \tag{7.32}$$

By summing the two covariance matrix, we can extract a correlation matrix presented in figure 7.10. The correlation between the SPMT and LPMT spectra is greater at the start of the spectrum, where the absolute smearing is the smallest, up to 5% correlation, and diffuse as the bins are further from each other and the absolute resolution grow.

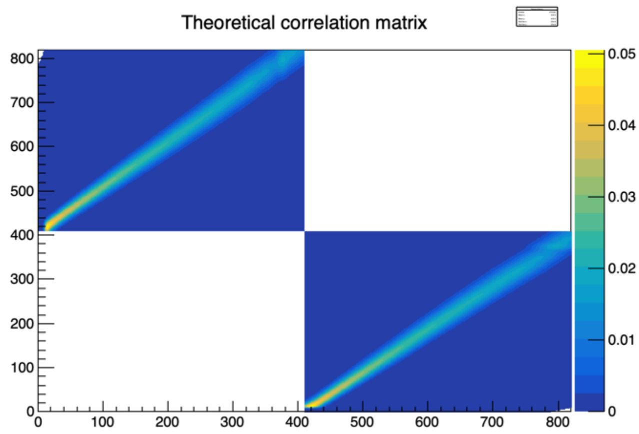


FIGURE 7.10 – Theoretical correlation matrix between the LPMT spectrum (bins 0-409) ans the SPMT spectrum (410-819). The diagonal has been set to 0 (it was 1) for readability purpose.

Empiric method

The second method is the empric way where we generate toys and just compute the empirical correlation between the bin contents.

$$\mathrm{Corr}(\theta_i, \theta_j) = \frac{\mathbb{E}[\theta_i \theta_j] - \mathbb{E}[\theta_i]\mathbb{E}[\theta_j]}{\sigma\theta_i \sigma\theta_j} \tag{7.33}$$

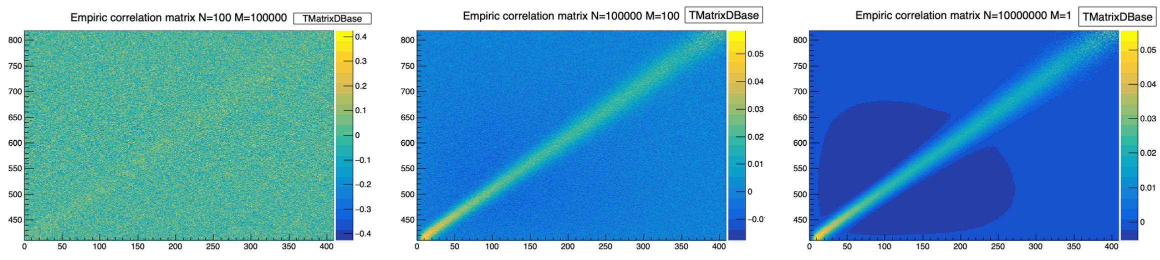
We thus generate 10^7 event using the IBD generator presented in section 7.3.1, then produce spectra from this finite set of events, meaning we must choose a number N of toy each composed of M event in order to have the best estimate.

Due to the nature of our estimator, the estimated correlation coefficient is subject to statistical fluctuation as any estimator. There is no definite formula to compute the standard deviation of the correlation coefficient as suggested in this study [84] but all cited formula depend solely on the number of samples, in our case the number of toy N, and the correlation coefficient. This indicate that maximizing the number of toy is the right decision, even if each toy posses only one sole event.

To study this rather counter intuitive observation (How can a spectrum with only one event can be representative of the experiment ?), I present in figure 7.11 the upper left corner of the estimated correlation matrix for different configurations of N and M in the limit of 10^7 total event. Wee see in figure 7.11a that if the toy number N is too low, the statistical noise make the correlation pattern almost completely disappear, in figure 7.11b we see clearly the same correlation patter as in the theoretical matrix in figure 7.10. On the final matrix in figure 7.11c the pattern is clearly visible, but we see a shade of anti-correlation around the spectrum that was not present in the theoretical correlation matrix.

The difference between the element of the theoretical and the empiric correlation matrices are presented in figure 7.12a. We that the difference between the two is very small with a bias of $1.8 \cdot 10^{-3}$ and a standard deviation of $1.9 \cdot 10^{-3}$ while the interesting correlation are of the order 10^{-2}. As presented in figure 7.12b, the most extreme differences comes from the low end of the spectrum.

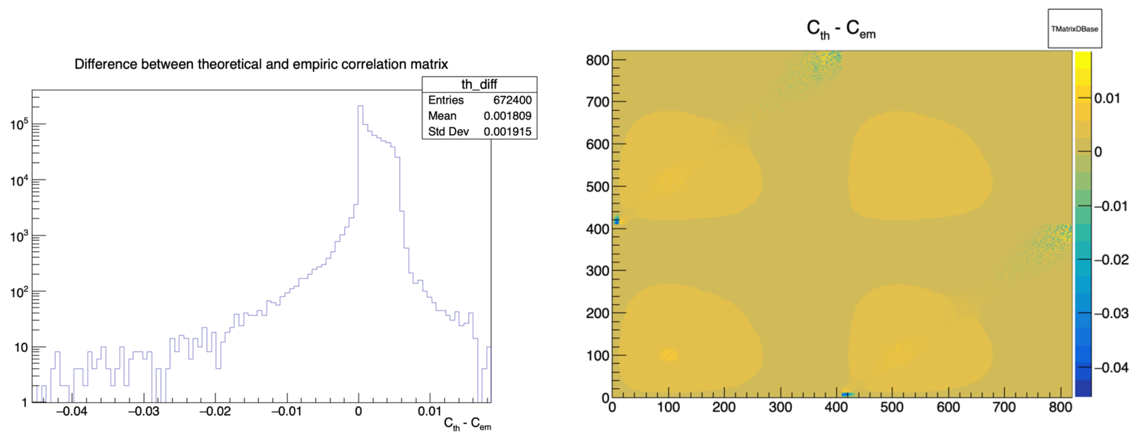
This low energy difference could be explained as the theoretical does not take into account event that would be smeared from outside the spectrum. $E < 0.8$, MeV back inside the spectrum thus missing on the potential correlations.



(A) (B) (C)

FIGURE 7.11 – 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

The second major difference between the empirical and theoretical correlation matrices is the anti-correlation of magnitude $\approx -5 \cdot 10^{-3}$ around the spectrum. In the theoretical correlation matrix, we assume that $G(i, j)$ is uncorrelated from $G(i, k)$ but this is not true in the case of a finite dataset. $G(i, j)$ represent the number of events that migrate from the bin i to j, in the case of a finite number of event to distribute between the bins, the number of event that can be distributed in the bin k is constrained by the number of event distributed in the bin j leading to the anti-correlation between this two bins.



(A) (B)

FIGURE 7.12 – Difference between the element of the theoretical and empiric correlation matrix

These empirical correlation matrices still pose an issue: These matrices needs to be invertible for χ^2 calculation. The framework use the Cholesky decomposition [85] for this, requiring the correlation matrices to be positive definite, which is not guarantee using this empirical methods. Due to this issue, the theoretical matrix is used in the studies presented in this thesis.

Empirical correlation matrix from fully simulated event

TODO

7.5.3 Statistical tests

In this part, I present the results of the statistical tests presented in section 7.2.

Test χ^2_{spe}

The χ^2_{spe} is a chi-square representing the compatibility between the LPMT ans SPMT spectra under constraints of the correlation matrix between the two.

$$\chi^2_{spe} = \mathbf{\Delta h} V_{spe} \mathbf{\Delta h}^T; \; \mathbf{\Delta h} = \{(h^L_0 - h^S_0), ..., (h^L_n - h^S_n)\} \tag{7.34}$$

where h^L_i and h^S_i are the contents of the ith bins of the LPMT and SPMT spectra. For details about the calculation of V_{spe}, see section 7.2.

The results for different exposures can be found in figure 7.13. 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 $\alpha_{qnl} = 3\%$ is 0.005 ± 0.0022.

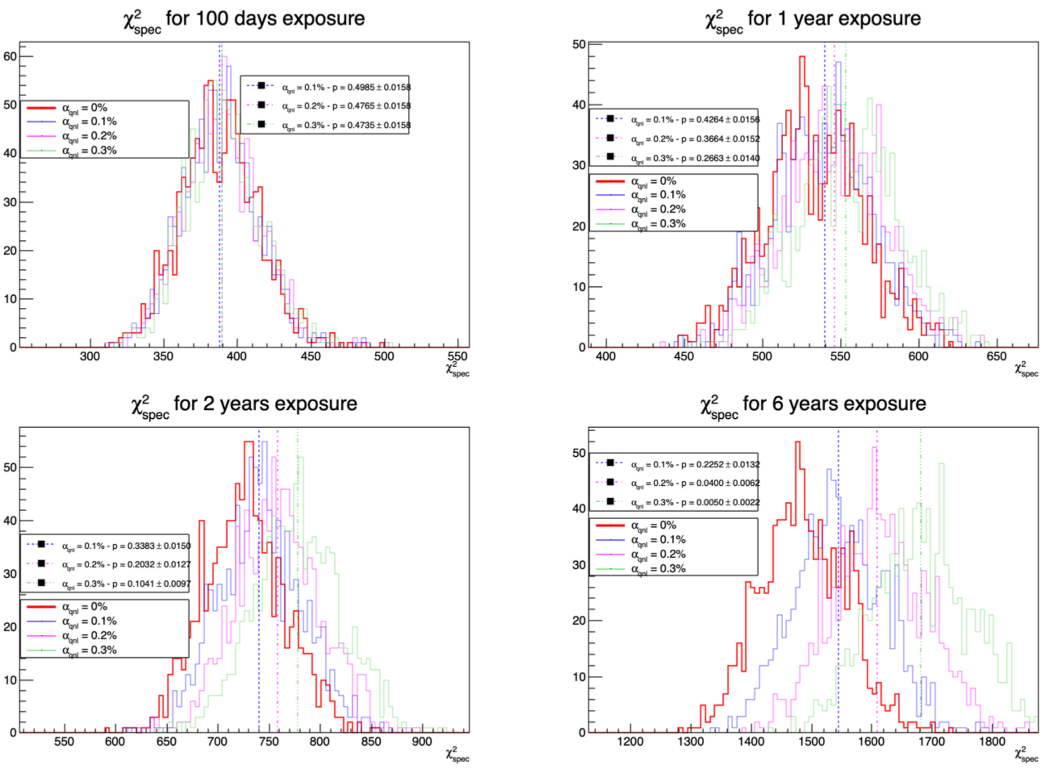


FIGURE 7.13 – Distribution of the χ^2_{spe} for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

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.

Test χ^2_{ind}

The χ^2_{ind} is the chi-square that represent the agreement between the measured oscillation parameters θ_{12} and Δm^2_{21}. This test is defined as

$$\chi^2_{ind} = \boldsymbol{\Delta\lambda} V_{ind} \boldsymbol{\Delta\lambda}^T;\ \boldsymbol{\Delta\lambda} = \{\theta^L_{12} - \theta^S_{12}, (\Delta m^2_{21})^L - (\Delta m^2_{21})^S\} \tag{7.35}$$

where θ^L_{12} and $(\Delta m^2_{21})^L$ are the oscillation parameters measured by the LPMT system. Same for θ^S_{12} and $(\Delta m^2_{21})^S$ for the SPMT system. We use V_{ind} computed for $\alpha_{qnl} = 0$. For more details about the calculation of V_{ind} see section 7.2.

The results are presented in figure 7.14. This test does not require any joint fit or covariance matrix estimation between the two spectrum, it just need the estimated covariance matrix between the four parameters. 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.

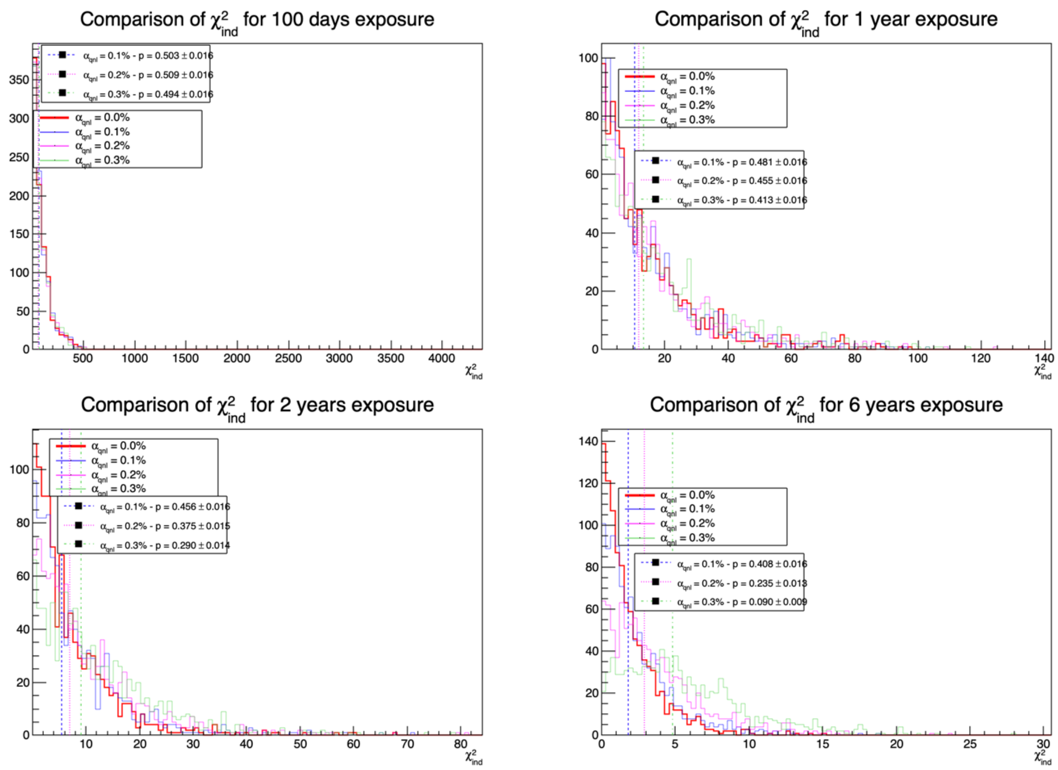
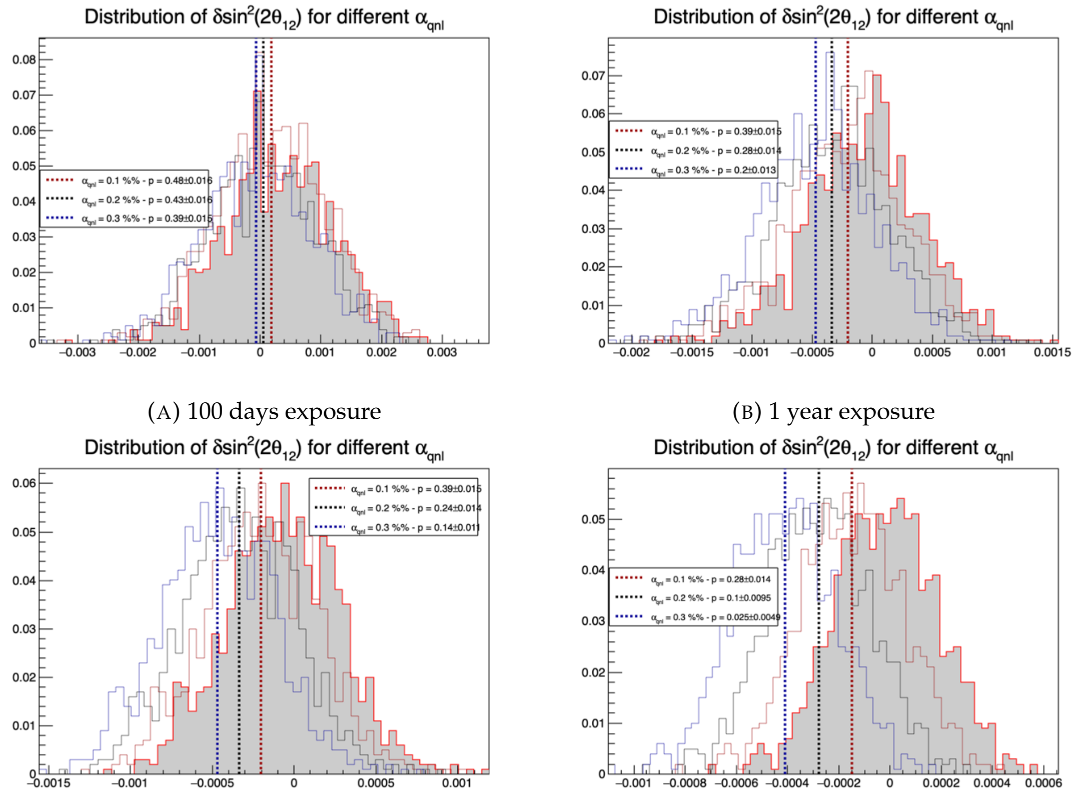


FIGURE 7.14 – Distribution of the χ^2_{ind} for 1000 toys for different exposures. The dashed lines represent the median of the distributions and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

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.

δ parameters significance

This test involves observing the values of the δ parameters in the Delta Joint fit and comparing them tho their dispersion in the case where $\alpha_{qnl} = 0$. The results are shown in figures 7.15 and 7.16.



(A) 100 days exposure

(B) 1 year exposure

(C) 2 years exposure

(D) 6 years exposure

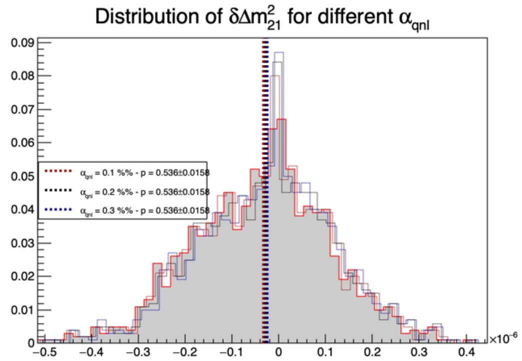
FIGURE 7.15 – Distribution of the $\delta \sin^2(2\theta_{12})$ for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

We can see that the $\delta\Delta m^2_{21}$ has a very small discriminative power (figure 7.16) 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 7.15) 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}.

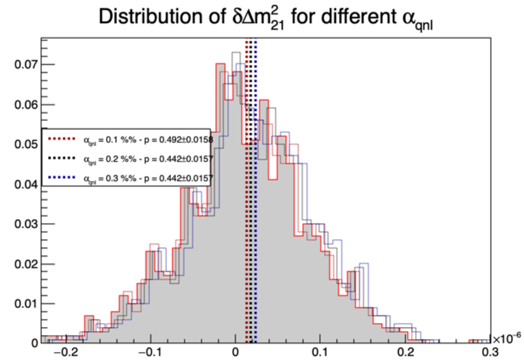
Hypothesis test

In this last test we consider the two fit Standard Joint and Delta Joint as two hypothesis. The first one, Standard Joint, is the H_0 hypothesis: we do not need supplementary parameters to describe the energy spectrum. The second one, Delta Joint, is the H_1 hypothesis: we do need those supplementary δ parameters to, if not correctly, approach the energy spectrum. If the δ parameter are unnecessary the $\chi^2_{H_0}$ should be close to $\chi^2_{H_1}$. On the other hand, if one spectrum is distorted, then those parameters are relevant and $\chi^2_{H_1} < \chi^2_{H_0}$. For this test we thus observe the $\chi^2_{H_0} - \chi^2_{H_1}$ distributions for different exposures and α_{qnl}. The results are presented in figure 7.17.

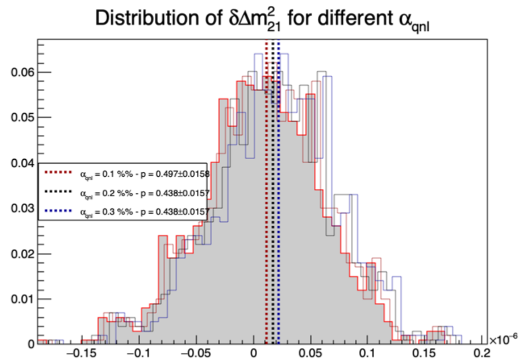
This test is the most complex, requiring two fit and the covariance matrix between the LPMT and SPMT spectra. The results are good, close to the χ^2_{spe}, one with a p-value at 6 years for $\alpha_{qnl} = 0.3\%$ of 0.01 ± 0.003.



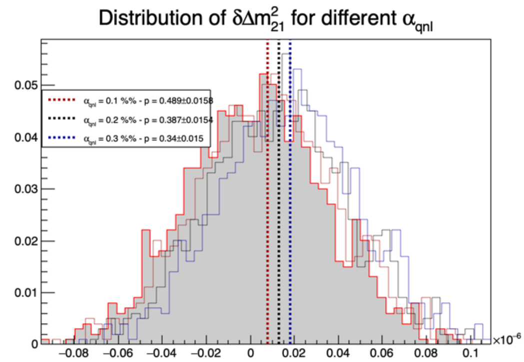
(A) 100 days exposure



(B) 1 year exposure



(C) 2 years exposure



(D) 6 years exposure

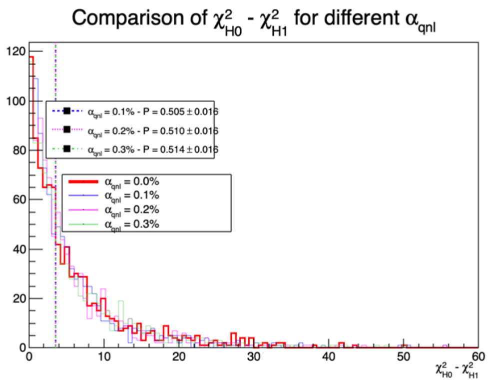
FIGURE 7.16 – Distribution of the $\delta\Delta m^2_{21}$ for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

As explained in section , the spectra used for the fit are cut at 335 bins / 7.5 MeV to prevent instability, while in χ^2_{spe} we use full 410 bins spectra. The χ^2_{spe} thus has more informations that the hypothesis test leading to this difference in power.

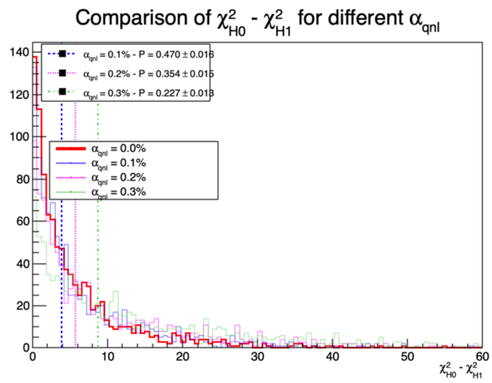
7.6 Conclusion and perspectives

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. Our results show that after 6 years of data collection, we can reject the median residual distortion with a p-value of 0.5% under the conditions outlined in this chapter.

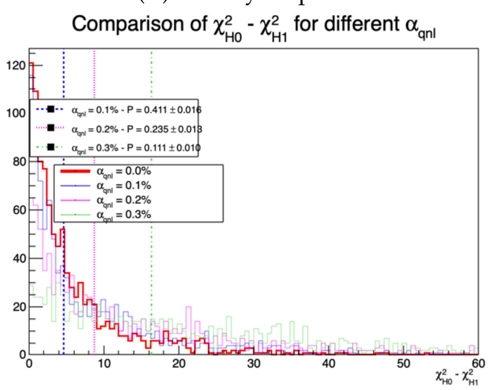
Additionally, this study is preliminary, as the background was neglected in the distortion test, and no systematic uncertainties were considered. The supplementary non-linearity was introduced event-wise but should be applied channel-wise to account for the detector's non-uniformity. The correlation matrix between the LPMT and SPMT spectra should also be further analyzed, as indicated by the discrepancies between the theoretical and empirical correlation matrices. We should also further investigate the effect of non-uniformity on the correlation matrix.



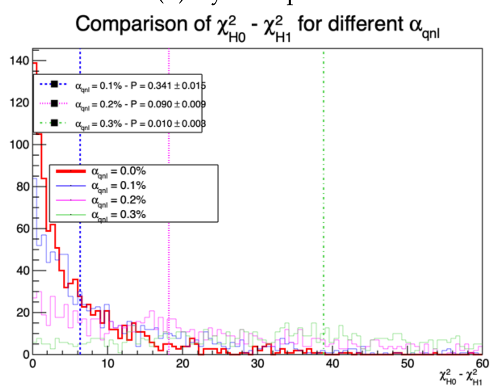
(A) 100 days exposure



(B) 1 year exposure



(C) 2 years exposure



(D) 6 years exposure

FIGURE 7.17 – 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.

Chapter 8

Conclusion

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 4.4.2.

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 [76] 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 $l = 2$.

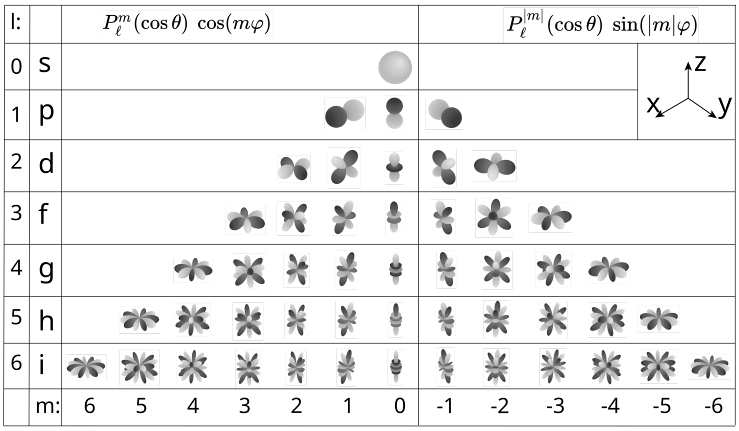


FIGURE B.1 – Illustration of the real part of the spherical harmonics

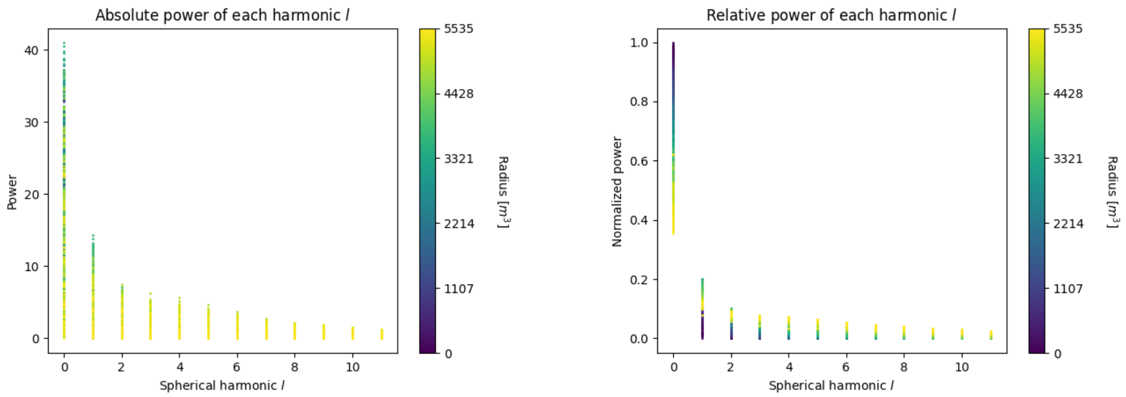


FIGURE B.2 – Scatter plot of the absolute and relative power, respectively on the left and right plot, of each harmonic degree l. The color indicate the radius of the event.

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 [86]. 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.

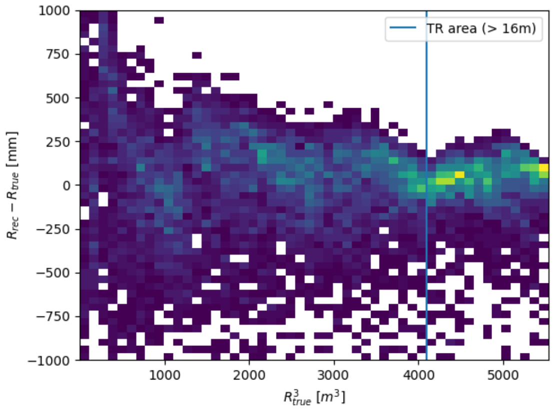
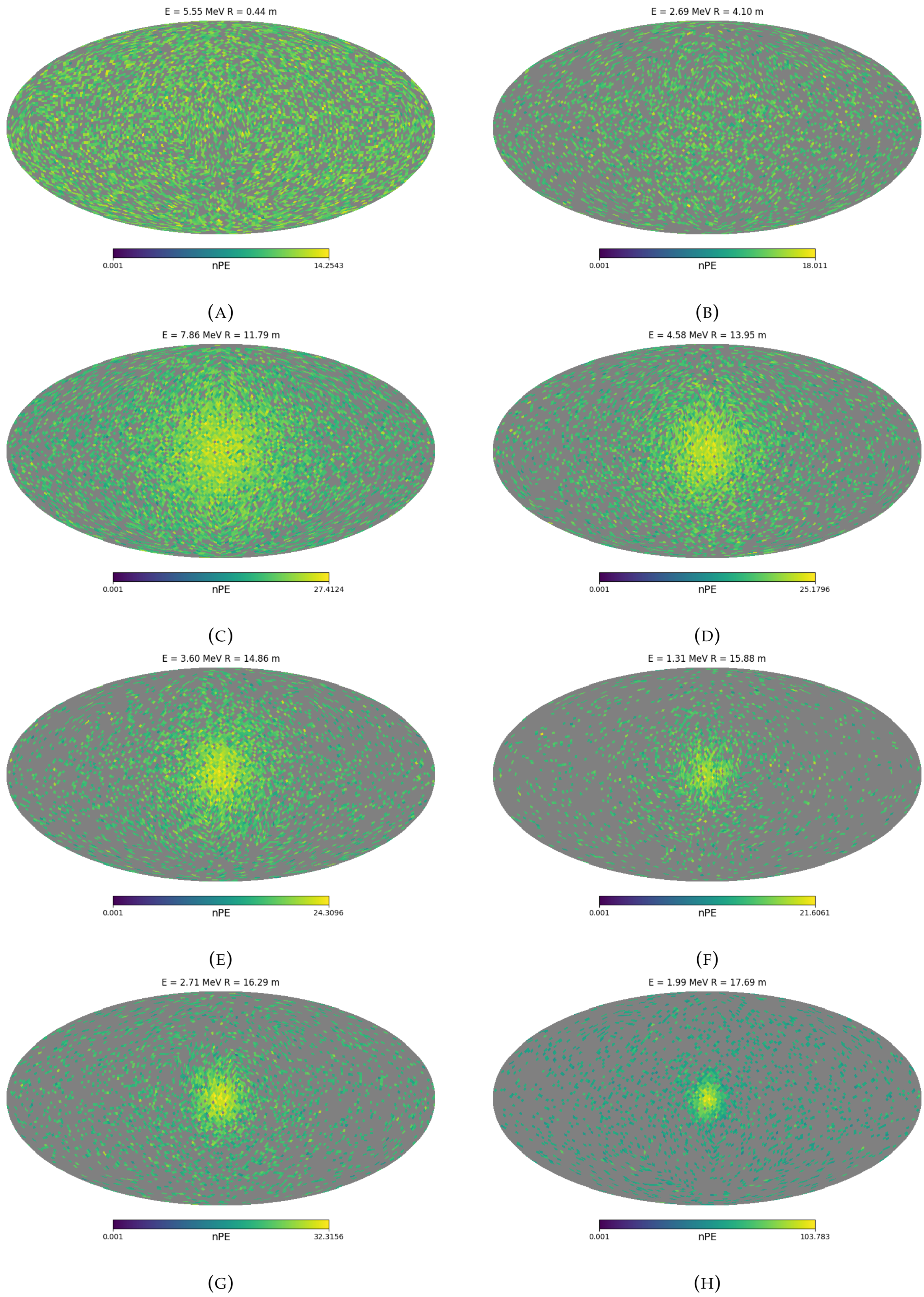


FIGURE B.3 – Error on the reconstructed radius vs the true radius by the harmonic method

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 4.4.2 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.



(A) (B) (C) (D) (E) (F) (G) (H)

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.

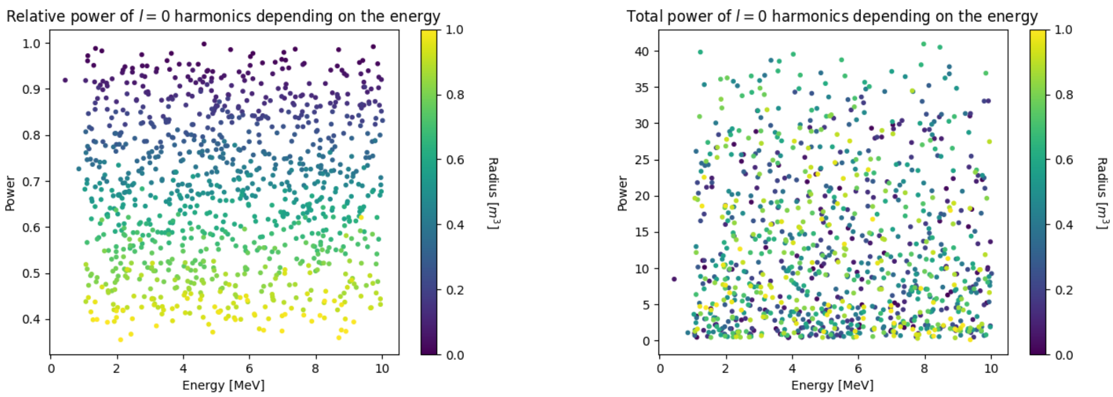


FIGURE B.5 – Scatter plot of the absolute and relative power, respectively on the left and right plot, of the $l = 0$ harmonic. The color indicate the radius of the event.

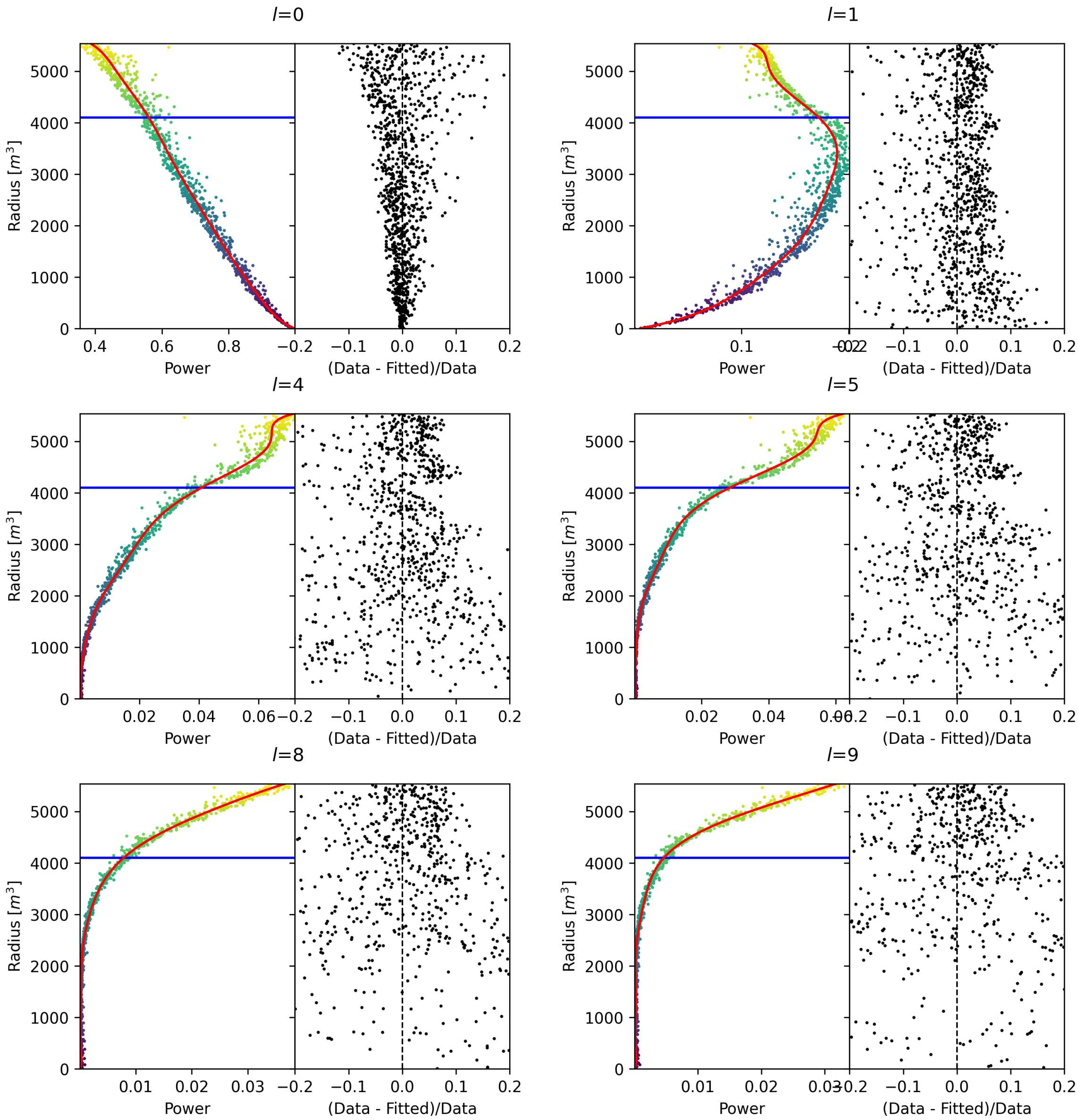


FIGURE B.6 – Plot of the distribution of the relative power of each harmonic dependent on R^3 (on the left). The Total Reflection (TR) area is represented by the horizontal blue line. The distribution are fitted using a 9th degree polynomial (red curve). The relative power error between the distribution and the fit is represented on the left. **Part 1**

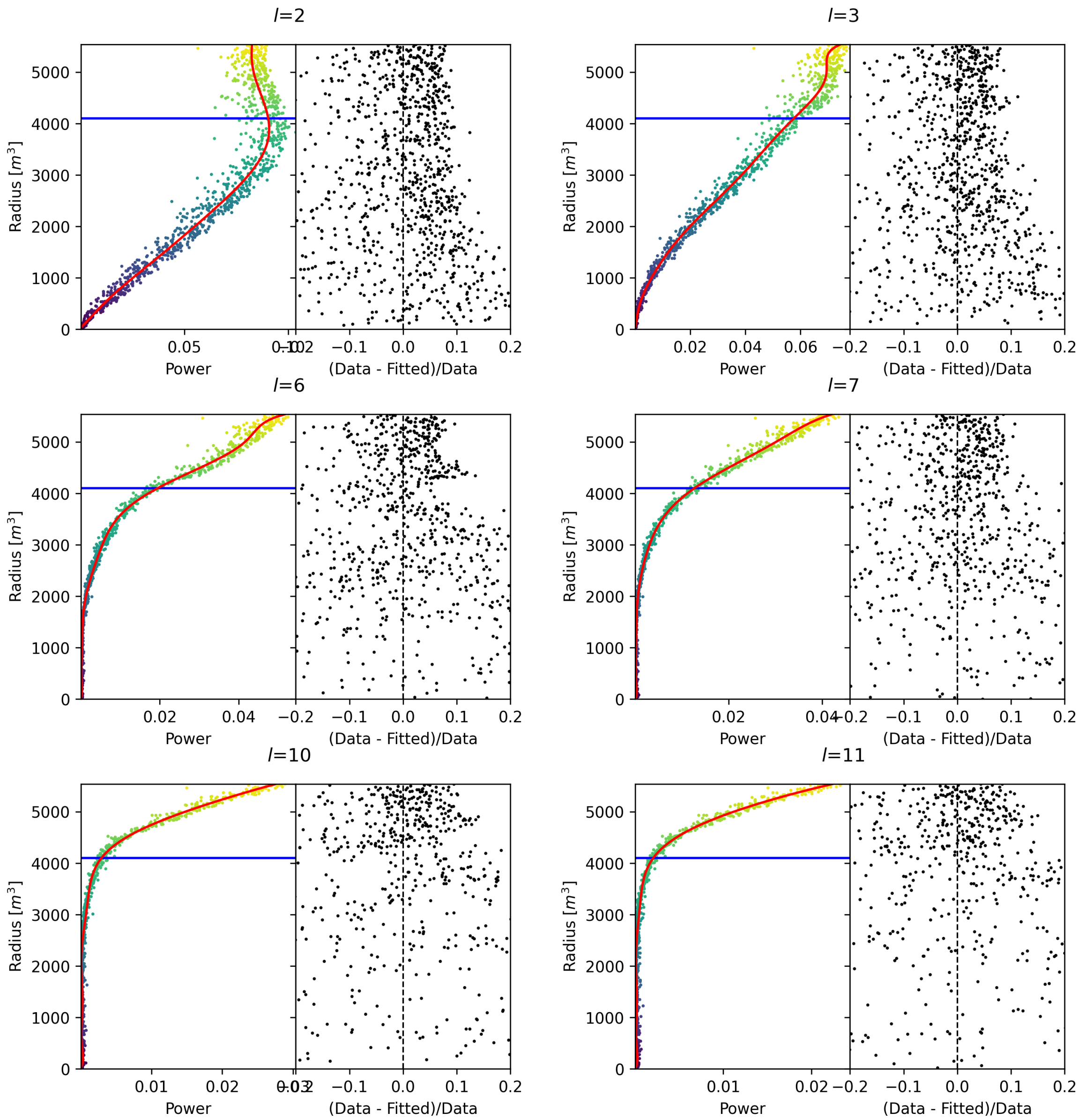


FIGURE B.7 – Plot of the distribution of the relative power of each harmonic dependent on R^3 (on the left). The Total Reflection (TR) area is represented by the horizontal blue line. The distribution are fitted using a 9th degree polynomial (red curve). The relative power error between the distribution and the fit is represented on the left. **Part 2**

Appendix C

Additional spectrum smearing

In this section we demonstrate that a spectrum S smeared by a gaussian G parametrized by its varianse σ_1^2 can be smeared by a gaussian parametrized by the variance σ_2^2 from the the smeared spectrum $K(E,\sigma_1) = S(E) \star G(E,\sigma_1)$ under the condition that $\sigma_2^2 > \sigma_1^2$.

Let $K'(E,\sigma_2) = S(E) \star G(E,\sigma_2)$ the target spectrum we can expand

$$\begin{aligned} K'(E,\sigma_2) &= S(E) \star G(E,\sigma_1) \star G^{-1}(E,\sigma_1) \star G(E,\sigma_2) && \text{(C.1)} \\ &= K(E,\sigma_1) \star G^{-1}(E,\sigma_1) \star G(E,\sigma_2) && \text{(C.2)} \end{aligned}$$

where $G^{-1}(E,\sigma_1)$ is defined as $G(E,\sigma_1) \star G^{-1}(E,\sigma_1) = \delta(E)$.

By moving into Fourier space we can express

$$\begin{aligned} G(E,\sigma_1) \star G^{-1}(E,\sigma_1) &= \delta(E) && \text{(C.3)} \\ F[G(E,\sigma_1)](\nu) \times F[G^{-1}(E,\sigma_1)](\nu) &= 1 && \text{(C.4)} \end{aligned}$$

with $F[G(E,\sigma_1)](\nu)$ the fourier transform of G

$$F[G(E,\sigma_1)](\nu) = e^{-\frac{\sigma_1^2(2\pi)^2}{2}\nu^2} \tag{C.5}$$

we have

$$\begin{aligned} F[G^{-1}(E,\sigma_1)(\nu) = \big(F[G(E,\sigma_1)](\nu)\big)^{-1} &= \big(e^{-\frac{\sigma_1^2(2\pi)^2}{2}\nu^2}\big)^{-1} && \text{(C.6)} \\ &= e^{\frac{\sigma_1^2(2\pi)^2}{2}\nu^2} && \text{(C.7)} \end{aligned}$$

Thus we express

$$\begin{aligned} F[G^{-1}(E,\sigma_1) \star G(E,\sigma_2)] &= e^{\frac{\sigma_1^2(2\pi)^2}{2}\nu^2} \times e^{-\frac{\sigma_2^2(2\pi)^2}{2}\nu^2} && \text{(C.8)} \\ &= e^{\frac{(2\pi)^2}{2}(\sigma_1^2-\sigma_2^2)\nu^2} && \text{(C.9)} \\ &= e^{\frac{(2\pi)^2}{2}\Delta\sigma^2\nu^2};\ \Delta\sigma^2 = (\sigma_1^2 - \sigma_2^2) && \text{(C.10)} \end{aligned}$$

We see that $F^{-1}[F[G^{-1}(E,\sigma_1) \star G(E,\sigma_2)]]$ is solvable if $\Delta\sigma^2 = (\sigma_1^2 - \sigma_2^2) < 0 \Rightarrow \sigma_2 > \sigma_1$. In that case

$$G^{-1}(E,\sigma_1) \star G(E,\sigma_2) = \frac{1}{\sqrt{|\Delta\sigma^2|}\sqrt{2\pi}} e^{-\frac{E^2}{2|\Delta\sigma^2|}} \tag{C.11}$$

List of Tables

List of Figures

List of Abbreviations

ACU	**A**utomatic **C**alibration **U**nit
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] Liang Zhan, Yifang Wang, Jun Cao, and Liangjian Wen. "Experimental Requirements to Determine the Neutrino Mass Hierarchy Using Reactor Neutrinos". *Physical Review D* 79.7 (Apr. 14, 2009), 073007. ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.79.073007. eprint: 0901.2976[hep-ex]. URL: http://arxiv.org/abs/0901.2976 (visited on 09/18/2023).

[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] 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).

[12] 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).

[13] 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).

[14] V. Kopeikin, M. Skorokhvatov, and O. Titov. "Reevaluating reactor antineutrino spectra with new measurements of the ratio between ^{235}U and ^{239}Pu β spectra". *Physical Review D* 104.7 (Oct. 25, 2021), L071301. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.104.L071301. eprint: 2103.01684[hep-ph,physics:nucl-ex,physics:nucl-th]. URL: http://arxiv.org/abs/2103.01684 (visited on 01/18/2024).

[15] A. Letourneau et al. "On the origin of the reactor antineutrino anomalies in light of a new summation model with parameterized β^- transitions". *Physical Review Letters* 130.2 (Jan. 10, 2023), 021801. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.130.021801. eprint: 2205.14954[hep-ex,physics:hep-ph]. URL: http://arxiv.org/abs/2205.14954 (visited on 01/16/2024).

[16] 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).

[17] 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).

[18] 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).

[19] 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).

[20] 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).

[21] 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).

[22] *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).

[23] 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).

[24] 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).

[25] 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).

[26] 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).

[27] Yang Han. "Dual Calorimetry for High Precision Neutrino Oscillation Measurement at JUNO Experiment". AstroParticule et Cosmologie, France, Paris U. VII, APC, June 2021.

[28] 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).

[29] 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).

[30] 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).

[31] 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).

[32] 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).

[33] 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).

[34] 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).

[35] 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).

[36] 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).

[37] 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).

[38] 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).

[39] 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).

[40] 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).

[41] 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).

[42] 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).

[43] 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).

[44] R. Abbasi et al. "Graph Neural Networks for low-energy event classification & reconstruction in IceCube". *Journal of Instrumentation* 17.11 (Nov. 2022). Publisher: IOP Publishing, P11003. ISSN: 1748-0221. DOI: 10.1088/1748-0221/17/11/P11003. URL: https://dx.doi.org/10.1088/1748-0221/17/11/P11003 (visited on 04/04/2024).

[45] S. Reck, D. Guderian, G. Vermariën, A. Domi, and on behalf of the KM3NeT collaboration on behalf of the. "Graph neural networks for reconstruction and classification in KM3NeT". *Journal of Instrumentation* 16.10 (Oct. 2021). Publisher: IOP Publishing, C10011. ISSN: 1748-0221. DOI: 10.1088/1748-0221/16/10/C10011. URL: https://dx.doi.org/10.1088/1748-0221/16/10/C10011 (visited on 04/04/2024).

[46] The IceCube collaboration et al. "A convolutional neural network based cascade reconstruction for the IceCube Neutrino Observatory". *Journal of Instrumentation* 16.7 (July 2021). Publisher: IOP Publishing, P07041. ISSN: 1748-0221. DOI: 10.1088/1748-0221/16/07/P07041. URL: https://dx.doi.org/10.1088/1748-0221/16/07/P07041 (visited on 04/04/2024).

[47] DUNE Collaboration et al. "Neutrino interaction classification with a convolutional neural network in the DUNE far detector". *Physical Review D* 102.9 (Nov. 9, 2020). Publisher: American Physical Society, 092003. DOI: 10.1103/PhysRevD.102.092003. URL: https://link.aps.org/doi/10.1103/PhysRevD.102.092003 (visited on 04/04/2024).

[48] 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).

[49] 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).

[50] 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).

[51] 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.

[52] 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).

[53] J. Y. Lettvin, H. R. Maturana, W. S. McCulloch, and W. H. Pitts. "What the Frog's Eye Tells the Frog's Brain". *Proceedings of the IRE* 47.11 (Nov. 1959). Conference Name: Proceedings of the IRE, 1940–1951. ISSN: 2162-6634. DOI: 10.1109/JRPROC.1959.287207. URL: https://ieeexplore.ieee.org/document/4065609 (visited on 05/06/2024).

[54] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain". *Psychological Review* 65.6 (1958). Place: US Publisher: American Psychological Association, 386–408. ISSN: 1939-1471. DOI: 10.1037/h0042519.

[55] 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).

[56] 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).

[57] 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).

[58] 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).

[59] 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).

[60] 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).

[61] *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).

[62] 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).

[63] 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).

[64] 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).

[65] 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).

[66] 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).

[67] 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).

[68] 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).

[69] 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).

[70] 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).

[71] *NVIDIA A100 GPUs Power the Modern Data Center*. NVIDIA. URL: https://www.nvidia.com/en-gb/data-center/a100/ (visited on 08/06/2024).

[72] *NVIDIA V100*. NVIDIA. URL: https://www.nvidia.com/en-gb/data-center/v100/ (visited on 08/06/2024).

[73] 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).

[74] "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).

[75] 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).

[76] 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).

[77] 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).

[78] 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).

[79] 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).

[80] Angel Abusleme et al. "Potential to Identify the Neutrino Mass Ordering with Reactor Antineutrinos in JUNO" (May 2024). eprint: 2405.18008.

[81] Rene Brun et al. *root-project/root: v6.26/06*. Version v6-26-06. Mar. 3, 2022. DOI: 10.5281/zenodo.3895860. URL: https://zenodo.org/records/3895860 (visited on 09/05/2024).

[82] 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).

[83] 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).

[84] Timo Gnambs. "A Brief Note on the Standard Error of the Pearson Correlation". *Collabra: Psychology* 9.1 (Sept. 6, 2023). Ed. by Thomas Evans, 87615. ISSN: 2474-7394. DOI: 10.1525/collabra.87615. URL: https://doi.org/10.1525/collabra.87615 (visited on 09/10/2024).

[85] "Note Sur Une Méthode de Résolution des équations Normales Provenant de L'Application de la MéThode des Moindres Carrés a un Système D'équations Linéaires en Nombre Inférieur a Celui des Inconnues. — Application de la Méthode a la Résolution D'un Système Defini D'éQuations LinéAires". *Bulletin géodésique* 2.1 (Apr. 1, 1924), 67–77. ISSN: 1432-1394. DOI: 10.1007/BF03031308. URL: https://doi.org/10.1007/BF03031308 (visited on 09/10/2024).

[86] 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).