

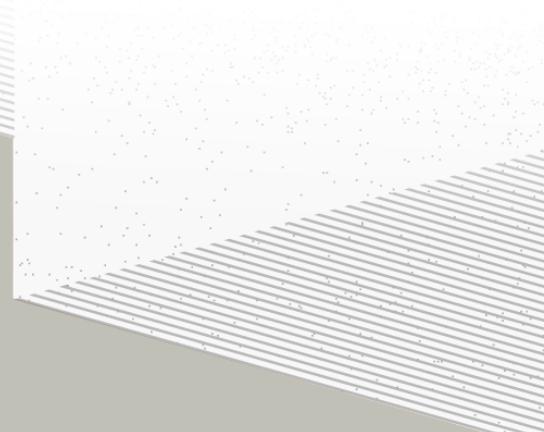
1

2

THÈSE DE DOCTORAT DE

NANTES UNIVERSITÉ

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



Par

Léonard Imbert

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

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

Rapporteurs avant soutenance :

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

Composition du Jury :

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

³ Contents

⁴	Contents	¹
⁵	Remerciements	⁵
⁶	Introduction	⁷
⁷	1 Neutrino physics	⁹
⁸	1.1 Standard model	⁹
⁹	1.1.1 Limits of the standard model	⁹
¹⁰	1.2 Historic of the neutrino	⁹
¹¹	1.3 Oscillation	⁹
¹²	1.3.1 Phenomologies	⁹
¹³	1.4 Open questions	⁹
¹⁴	2 The JUNO experiment	¹¹
¹⁵	2.1 Reactor Neutrinos physics in JUNO	¹²
¹⁶	2.1.1 Antineutrino spectrum measured in JUNO	¹²
¹⁷	2.1.2 Background spectra	¹⁴
¹⁸	2.1.3 Statistical analysis and sensitivities	¹⁵
¹⁹	2.2 Other physics	¹⁶
²⁰	2.3 The JUNO detector	¹⁸
²¹	2.3.1 Detection principle	¹⁹
²²	2.3.2 Central Detector (CD)	¹⁹
²³	2.3.3 Veto detector	²³
²⁴	2.4 Calibration strategy	²⁴
²⁵	2.4.1 Energy scale calibration	²⁴
²⁶	2.4.2 Calibration system	²⁵
²⁷	2.4.3 Instrumental non-linearity calibration	²⁶
²⁸	2.5 Satellite detectors	²⁷
²⁹	2.5.1 TAO	²⁷
³⁰	2.5.2 OSIRIS	²⁸
³¹	2.6 Software	²⁸
³²	2.7 State of the art of the Offline IBD reconstruction in JUNO	²⁹
³³	2.7.1 Interaction vertex reconstruction	²⁹
³⁴	2.7.2 Energy reconstruction	³⁴

35	2.7.3 Machine learning for reconstruction	37
36	2.7.4 Physics results	39
37	2.8 Summary	39
38	3 Machine learning: Introduction to the methods and algorithms used in this thesis	41
39	3.1 Core concepts in machine learning and neural networks	42
40	3.1.1 Boosted Decision Tree (BDT)	42
41	3.1.2 Artificial Neural Network (NN)	42
42	3.1.3 Training procedure	44
43	3.1.4 Potential pitfalls	47
44	3.2 Neural networks architectures	50
45	3.2.1 Fully Connected Deep Neural Network (FCDNN)	50
46	3.2.2 Convolutional Neural Network (CNN)	50
47	3.2.3 Graph Neural Network (GNN)	52
48	3.2.4 Adversarial Neural Network (ANN)	54
49	4 Image recognition for IBD reconstruction with the SPMT system	55
50	4.1 Method and model	56
51	4.1.1 Model	57
52	4.1.2 Data representation	58
53	4.1.3 Dataset	60
54	4.1.4 Data characteristics	61
55	4.2 Training	63
56	4.3 Results	63
57	4.3.1 J21 results	64
58	4.3.2 J21 Combination of classic and ML estimator	66
59	4.3.3 J23 results	68
60	4.4 Conclusion and prospect	70
61	5 Graph representation of JUNO for IBD reconstruction	73
62	5.1 Data representation	74
63	5.2 Message passing algorithm	76
64	5.3 Data	78
65	5.4 Model	80
66	5.5 Training	80
67	5.6 Optimization	82
68	5.6.1 Software optimization	82
69	5.6.2 Hyperparameters optimization	83
70	5.7 performance of the final version	83
71	5.8 Conclusion	87
72	6 Reliability of machine learning methods	91
73	6.1 Method	92
74	6.2 Architecture	92

6.2.1	Back-propagation problematic	93
6.2.2	Reconstruction Network	94
6.2.3	Adversarial Neural Network	94
6.2.4	Training	95
6.3	Results	95
6.3.1	Back to identity	95
6.3.2	Breaking of the reconstruction	95
6.4	Conclusion and prospect	95
7	Dual calorimetric analysis for Precision Measurement	97
7.1	Motivations	99
7.1.1	Discrepancies between the SPMT and LPMT results	99
7.1.2	Charge Non-Linearity (QNL)	100
7.2	Approach	101
7.2.1	Data production	102
7.2.2	Individual fits	103
7.2.3	Joint fit	104
7.2.4	Data and theoretical spectrum generation	105
7.2.5	Limitations	106
7.3	Fit software	106
7.3.1	IBD generator	107
7.3.2	Fit	108
7.4	Technical challenges and development	109
7.5	Results	109
7.5.1	Validation	109
7.5.2	Covariance matrix	113
7.5.3	Statistical tests	118
7.6	Conclusion and perspectives	120
8	Conclusion	125
A	Calculation of optimal α for estimator combination	127
A.1	Unbiased estimator	127
A.2	Optimal variance estimator	127
B	Charge spherical harmonics analysis	129
C	Additional spectrum smearing	137
D	Correction of E_{vis} bias	139
List of Tables		141
List of Figures		150
List of Abbreviations		151

¹¹³ Remerciements

¹¹⁴ **Introduction**

¹¹⁵ **Chapter 1**

¹¹⁶ **Neutrino physics**

¹¹⁷ *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.*

¹¹⁸ **Contents**

¹¹⁹ 1.1 Standard model	⁹
¹²⁰ 1.1.1 Limits of the standard model	⁹
¹²¹ 1.2 Historic of the neutrino	⁹
¹²² 1.3 Oscillation	⁹
¹²³ 1.3.1 Phenomologies	⁹
¹²⁴ 1.4 Open questions	⁹

¹²⁹ **1.1 Standard model**

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

¹³⁰ **1.1.1 Limits of the standard model**

¹³¹ **1.2 Historic of the neutrino**

¹³² **First theories**

¹³³ **Discovery**

¹³⁴ **Milestones and anomalies**

¹³⁵ **1.3 Oscillation**

¹³⁶ **1.3.1 Phenomologies**

¹³⁷ **1.4 Open questions**

¹³⁸ **Chapter 2**

¹³⁹ **The JUNO experiment**

¹⁴⁰

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

¹⁴¹

Contents

¹⁴² 2.1 Reactor Neutrinos physics in JUNO	¹⁴³	¹⁴⁴ 12
2.1.1 Antineutrino spectrum measured in JUNO	12
2.1.2 Background spectra	14
2.1.3 Statistical analysis and sensitivities	15
2.2 Other physics	16
2.3 The JUNO detector	18
2.3.1 Detection principle	19
2.3.2 Central Detector (CD)	19
2.3.3 Veto detector	23
2.4 Calibration strategy	24
2.4.1 Energy scale calibration	24
2.4.2 Calibration system	25
2.4.3 Instrumental non-linearity calibration	26
2.5 Satellite detectors	27
2.5.1 TAO	27
2.5.2 OSIRIS	28
2.6 Software	28
2.7 State of the art of the Offline IBD reconstruction in JUNO	29
2.7.1 Interaction vertex reconstruction	29
2.7.2 Energy reconstruction	34
2.7.3 Machine learning for reconstruction	37
2.7.4 Physics results	39
2.8 Summary	39

¹⁶⁹

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_{32}^2 . From this idea is born the Jiangmen Underground Neutrino Observatory (JUNO) experiment.

¹⁷⁴

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



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

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

2.1 Reactor Neutrinos physics in JUNO

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

2.1.1 Antineutrino spectrum measured in JUNO

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

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

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_{32}^2 and Δm_{31}^2 causing a phase shift of the spectrum as we can see in the figure 2.2.

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

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

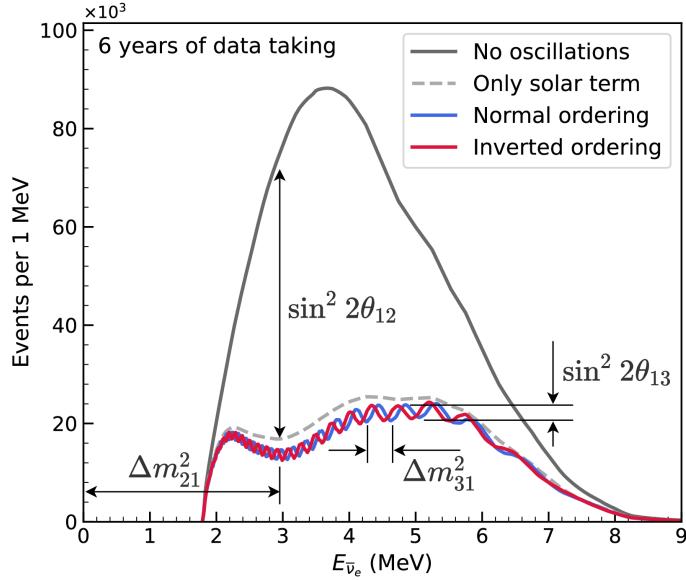


FIGURE 2.2 – Expected number of neutrinos event per MeV in JUNO after 6 years of data taking. The black curve shows the flux if there was no oscillation. The light gray curve shows the oscillation if only the solar terms are taken in account (θ_{12} , Δm_{21}^2). The blue and red curve shows the spectrum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

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

209 $\bar{\nu}_e$ flux coming from nuclear power plants

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

214 The $\bar{\nu}_e$ coming from reactors are emitted from β -decay of unstable fission fragments. The Taishan
215 and Yangjiang reactors are Pressurised Water Reactor (PWR), the same type as Daya Bay. In those
216 type of reactor more the 99.7 % and $\bar{\nu}_e$ are produced by the fissions of four fuel isotopes ^{235}U , ^{238}U ,
217 ^{239}Pu and ^{241}Pu . The neutrino flux per fission of each isotope is determined by the inversion of the
218 measured β spectra of fission product [4–8] or by calculation using the nuclear databases [9, 10].

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

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

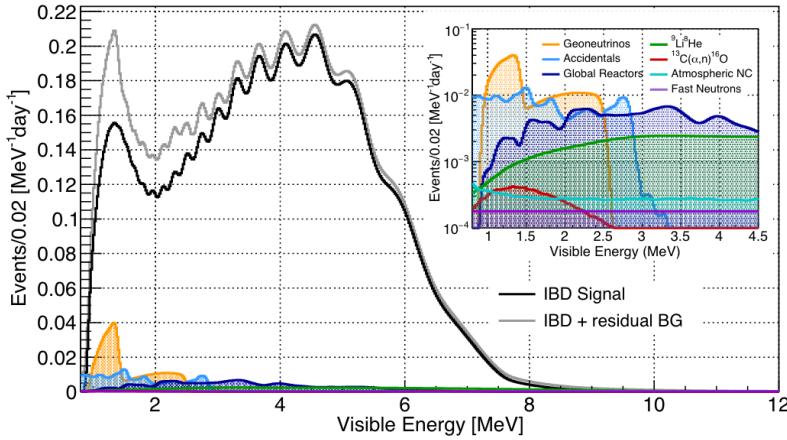


FIGURE 2.3 – Expected visible energy spectrum measured with the LPMT system with (grey) and without (black) backgrounds. The background amount for about 7% of the IBD candidate and are mostly localized below 3 MeV [3]

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

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

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

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

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

2.1.2 Background spectra

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

- The geoneutrinos producing background in the 0.511 ~ 2.7 MeV region.

- The neutrinos coming from the other nuclear reactors around Earth.
 In addition to all those physics signal, non-neutrinos signal that would mimic an IBD will also be present. It is composed of:
 — The signal coming from radioactive decay (α , γ , β) from natural radioactive isotopes in the material of the detector.
 — Cosmogenic event such as fast neutrons and activated isotopes induced by muons passing through the detector, most notably the spallation on ^{12}C .
 All those events represent a non-negligable part of the spectrum as shown in figure 2.3.

2.1.3 Statistical analysis and sensitivities

	Central Value	PDG 2020	100 days	6 years	20 years
$\Delta m_{31}^2 (\times 10^{-3}\text{eV}^2)$	2.5283	± 0.034 (1.3%)	± 0.021 (0.8%)	± 0.0047 (0.2%)	± 0.0029 (0.1%)
$\Delta m_{21}^2 (\times 10^{-3}\text{eV}^2)$	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 [14]) is compared with 100 days, 6 years and 20 years of JUNO data taking.

- The measurement of the reactor antineutrino oscillation at JUNO will be based on a spectrum reconstructed with the LPMT system. We describe briefly the fit procedure here.
 In Chapter 7, we will perform fits to the spectrum reconstructed with the SPMT system, as well as joint fits of the two kind of spectra. What's decribed below remains valid.

2.47 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 interest θ_{13} , θ_{12} , Δm_{21}^2 , Δm_{31}^2 . Noise parameters can be included in the parameters spectrum such as the earth density ρ between the power plants and JUNO.

2.64 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

²⁶⁷ defined as

$$\chi^2 = \sum_i \frac{(N_{th}^i - N_{data}^i)^2}{\sigma_i^2} \quad (2.2)$$

²⁶⁸ where N_{th}^i is the number event in the i th bin of the theoretical spectrum, N_{data}^i is the number of event
²⁶⁹ in the i th bin of the measured spectrum and σ_i is the uncertainty of this bin. Two classic statistic test
²⁷⁰ exist Pearson and Neyman where the difference is the estimation of σ_i parameters.

²⁷¹ This σ_i is composed of the systematics uncertainties discussed above but also from the statistic
²⁷² uncertainty of the spectrum. Considering a Poisson process, the statistic uncertainty is estimated
²⁷³ as $\sigma_{stat}^i = \sqrt{N^i}$. In a Pearson test, $N^i \equiv N_{th}^i$ whereas in a Neyman test $N^i \equiv N_{data}^i$. Under the
²⁷⁴ assumption that the content of each bin follow a Gaussian distribution (a Poisson with high enough
²⁷⁵ statistic), the two test are equivalent. But studies on Monte Carlo spectrum showed that the Pearson
²⁷⁶ and Neyman statistic are biased in opposite direction. It is easily visible where, for the same data,
²⁷⁷ Pearson will prefer a higher N_{th}^i to reduce the ratio $\frac{1}{N_{th}^i}$ whereas Neyman will prefer a lower N_{th}^i to
²⁷⁸ reduce the $(N_{th}^i - N_{data}^i)$ term.

²⁷⁹ This problematic can be circumvented by summing the two test, yielding the CNP statistic test
²⁸⁰ and/or by adding a term

$$\chi^2 = \sum_i \frac{(N_{th}^i - N_{data}^i)^2}{\sigma_i^2} - \ln |\mathbf{V}| \quad (2.3)$$

²⁸¹ where V is the covariance matrix of the theoretical spectrum yielding the PearsonV and CNPV
²⁸² statistic test.

²⁸³ The χ^2 is minimized by exploring the parameter phase space via gradient descent.

²⁸⁴ 2.2 Other physics

²⁸⁵ While the design of JUNO is tailored to measure $\bar{\nu}_e$ coming from nuclear reactor, JUNO will be able
²⁸⁶ to detect neutrinos coming from other sources thus allowing for a wide range of physics studies as
²⁸⁷ 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
DSNB (w/o PSD)	2300 elastic scattering		
Solar neutrino	2–4 IBDs/year	10–40 MeV	Atmospheric ν
Atmospheric neutrino	hundreds per year for ${}^{8}\text{B}$	0–16 MeV	Radioactivity
Geoneutrino	hundreds per year	0.1–100 GeV	Negligible
	≈ 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

²⁸⁸ Geoneutrinos

²⁸⁹ Geoneutrinos designate the antineutrinos coming from the decay of long-lived radioactive elements
²⁹⁰ inside the Earth. The 1.8 MeV threshold necessary for the IBD makes it possible to measure geoneu-
²⁹¹ trinos from ${}^{238}\text{U}$ and ${}^{232}\text{Th}$ decay chains. The studies of geoneutrinos can help refine the Earth
²⁹² crust models but is also necessary to characterise their signal, as they are a background to the mass
²⁹³ ordering and oscillations parameters studies.

294 **Atmospheric neutrinos**

295 Atmospheric neutrinos are neutrinos originating from the decay of π and K particles that are pro-
 296 duced in extensive air showers initiated by the interactions of cosmic rays with the Earth atmosphere.
 297 Earth is mostly transparent to neutrinos below the PeV energy, thus JUNO will be able to see neu-
 298 trinos coming from all directions. Their baseline range is large (15km \sim 13000km), they can have
 299 energy between 0.1 GeV and 10 TeV and will contain all neutrino and antineutrinos flavour. Their
 300 studies is complementary to the reactor antineutrinos and can help refine the constraints on the NMO
 301 [2].

302 **Supernovae burst neutrinos**

303 Neutrinos are crucial component during all stages of stellar collapse and explosion. Detection of
 304 neutrinos coming for core collapse supernovae will provide us important informations on the mech-
 305 anisms at play in those events. Thanks to its 20 kt sensible volume, JUNO has excellent capabilities
 306 to detect all flavour of the $\mathcal{O}(10 \text{ MeV})$ postshock neutrinos, and using neutrinos of the $\mathcal{O}(1 \text{ MeV})$
 307 will give informations about the pre-supernovae neutrinos. All those informations will allow to
 308 disentangle between the multiple hydro-dynamic models that are currently used to describe the
 309 different stage of core-collapse supernovae.

310 **Diffuse supernovae neutrinos background**

311 Core-collapse supernovae in our galaxy are rare events, but they frequently occur throughout the
 312 visible Universe sending burst of neutrinos in direction of the Earth. All those events contributes to
 313 a low background flux of low-energy neutrinos called the Diffuse Supernovae Neutrino Background
 314 (DSNB). Its flux and spectrum contains informations about the red-shift dependent supernovae rate,
 315 the average supernovae neutrino energy and the fraction of black-hole formation in core-collapse su-
 316 pernovae. Depending of the DSNB model, we can expect 2-4 IBD events per year in the energy range
 317 above the reactor $\bar{\nu}_e$ signal, which is competitive with the current Super-Kamiokande+Gadolinium
 318 phase [15].

319 **Beyond standard model neutrinos interactions**

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

328 **Proton decay**

329 Proton decay is a potential unobserved event where the proton decay by violating the baryon num-
 330 ber. This violation is necessary to explain the baryon asymmetry in the universe and is predicted
 331 by multiple Grand Unified Theories which unify the strong, weak and electromagnetic interactions.
 332 Thanks to its large active volume, JUNO will be able to take measurement of the potential proton
 333 decay channel $p \rightarrow \bar{\nu}K^+$ [16] thanks to the timing resolution of the SPMT system. Studies show
 334 that JUNO should be competitive with the current best limit at 5.9×10^{33} years from Super-K. This

³³⁵ studies show that JUNO, considering no proton decay events observed, would be able to rule a
³³⁶ limit of 9.6×10^{33} years at 90 % C.L.

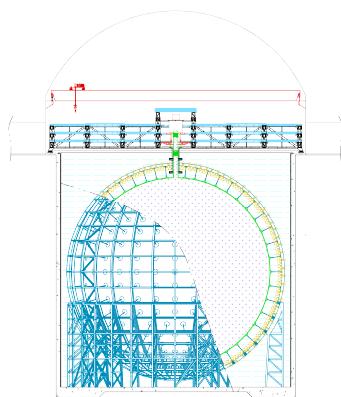
³³⁷ 2.3 The JUNO detector

³³⁸ The JUNO detector is a scintillator detector buried 693.35 meters under the ground (1800 meters
³³⁹ water equivalent). It consists of Central Detector (CD), a water pool and a Top Tracker (TT) as shown in figure 2.4a. The CD is an acrylic vessel containing the 20 ktons of Liquid Scintillator (LS). It is
³⁴⁰ supported by a stainless steel structure and is immersed in that water pool that is used as shielding
³⁴¹ from external radiation and as a Cherenkov detector for the background. The top of the experiment
³⁴² is partially covered by the Top Tracker (TT), a plastic scintillator detector which is used to detect the
³⁴³ atmospheric muons background and is acting as a veto detector.
³⁴⁴

³⁴⁵ The top of the experiment also hosts 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 systems of
³⁴⁶ Photo-Multiplier Tubes (PMT). They are attached to the steel structure and their electronic readout
³⁴⁷ is submerged near them. A third system of PMT is also installed on the structure but are facing
³⁴⁸ outward of the CD, instrumenting the water to be Cherenkov detector. The CD and the Cherenkov
³⁴⁹ detector are optically separated by Tyvek sheet. A chimney for LS filling and purification and for
³⁵⁰ calibration operations connects the CD to the experimental hall from the top.
³⁵¹

³⁵² The CD has been dimensioned to meet the requirements presented in section 2.1.1:

- ³⁵³ — 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 allows for a full containment of most of the events, preventing the energy loss in non-instrumented parts that would arise from a segmented detector.
- ³⁵⁴ — Its large overburden shield it from most of the atmospheric background that would pollute the signal.
- ³⁵⁵ — The localization of the experiment, chosen to maximize the disappearance with a 53 km baseline and in a region that allows two nuclear power plants to be used as sources.



(A) Schematic view of the JUNO detector.



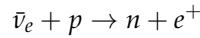
(B) Top down view of the JUNO detector under construction

FIGURE 2.4

³⁶¹ This section covers in details the different components of the detector and the detection systems.

³⁶² **2.3.1 Detection principle**

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



³⁶³ Kinematics calculation shows that this interaction has an energy threshold for the $\bar{\nu}_e$ of $(m_n + m_e -$
³⁶⁴ $m_p) \approx 1.806$ MeV [17]. This threshold make the experiment blind to very low energy neutrinos.
³⁶⁵ The residual energy $E_\nu - 1.806$ MeV is be distributed as kinetic energy between the positron and the
³⁶⁶ neutron. The energy of the emitted positron E_e is given by [17]

$$E_e = \frac{(E_\nu - \delta)(1 + \epsilon_\nu) + \epsilon_\nu \cos \theta \sqrt{(E_\nu - \delta)^2 + \kappa m_e^2}}{\kappa} \quad (2.4)$$

³⁶⁷ 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
³⁶⁸ equation that the positron energy is strongly correlated to the neutrino energy.

³⁶⁹ The positron and the neutron will then propagate in the detection medium, the Liquid Scintillator
³⁷⁰ (LS), loosing their kinetic energy by exciting the molecule of the LS (more details in section 2.3.2).
³⁷¹ Once stopped, the positron will annihilate with an electron from the medium producing two 511
³⁷² KeV gamma. Those gamma will themselves interact with the LS, exciting it before being absorbed
³⁷³ by photoelectrical effect. The neutron will be captured by an hydrogen, emitting a 2.2 MeV gamma
³⁷⁴ in the process. This gamma will also deposit its energy before being absorbed by the LS.

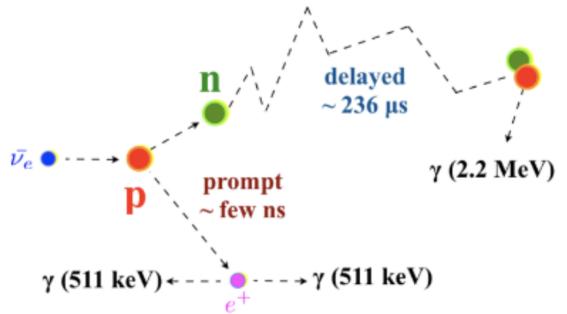


FIGURE 2.5 – Schematics of an IBD interaction in the central detector of JUNO

³⁷⁵ The scintillation photons have frequency in the UV and will propagate in the LS, being re-absorbed
³⁷⁶ and re-emitted by compton effect before finally be captured by PMTs instrumenting the acrylic
³⁷⁷ sphere. The analog signal of the PMTs digitized by the electronic is the signal of our experiment.
³⁷⁸ The signal produced by the positron is subsequently called the prompt signal, and the signal coming
³⁷⁹ from the neutron the delayed signal. This naming convention come from the fact that the positron
³⁸⁰ will deposit its energy rather quickly (few ns) where the neutron will take a bit more time (~ 236 μs).

³⁸¹ **2.3.2 Central Detector (CD)**

³⁸² The central detector, composed of 20 ktons of Liquid Scintillator (LS), is the main part of JUNO. The
³⁸³ LS is contained in a spherical acrylic vessel supported by a stainless steel structure. The CD and
³⁸⁴ its structural support are submerged in a cylindrical water pool of 43.5m diameter and 44m height.
³⁸⁵ We're confident that the water pool provide sufficient buffer protection in every direction against the
³⁸⁶ rock radioactivity.

387 **Acrylic vessel**

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

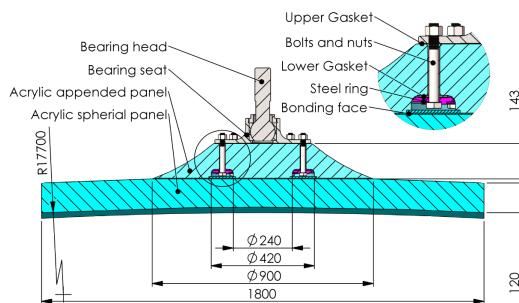


FIGURE 2.6 – Schematics of the supporting node for the acrylic vessel

398 **Liquid scintillator**

399 The Liquid Scintillator (LS) has a similar recipe as the one used in Daya Bay [18] but without gadolinium
 400 doping. It is made of three components, necessary to shift the wavelength of emitted photons to
 401 prevent their reabsorption and to shift their wavelength to the PMT sensitivity region as illustrated
 402 in figure 2.7:

- 403 1. The detection medium, the *linear alkylbenzene* (LAB). Selected because of its excellent trans-
 404 parency, high flash point, low chemical reactivity and good light yield. Accounting for ~
 405 98% of the LS, it is the main component with which ionizing particles and gamma interact.
 406 Charged particles will collide with its electronic cloud transferring energy to the molecules,
 407 gamma will interact via compton effect with the electronic cloud before finally be absorbed
 408 via photoelectric effect.
- 409 2. The second component of the LS is the *2,5-diphenyloxazole* (PPO). A fraction of the excitation
 410 energy of the LAB is transferred to the PPO, mainly via non radiative process [19]. The
 411 PPO molecules de-excites in the same way, transferring their energy to the bis-MSB. The PPO
 412 makes for 1.5 % of the LS.
- 413 3. The last component is the *p-bis(o-methylstyryl)-benzene* (bis-MSB). Once excited by the PPO, it
 414 will emit photon with an average wavelength of ~ 430 nm (full spectrum in figure 2.7) that
 415 can thus be detected by our photo-multipliers systems. It amount for ~ 0.5% of the LS.

416 This formula has been optimized using dedicated studies with a Daya Bay detector [18, 21] to reach
 417 the requirements for the JUNO experiment:

- 418 — A light yield / MeV of the amount of 10^4 photons to maximize the statistic in the energy
 419 measurement.
- 420 — An attenuation length comparable to the size of the detector to prevent losing photons during
 421 their propagation in the LS. The final attenuation length is 25.8m [22] to compare with the CD
 422 diameter of 35.4m.

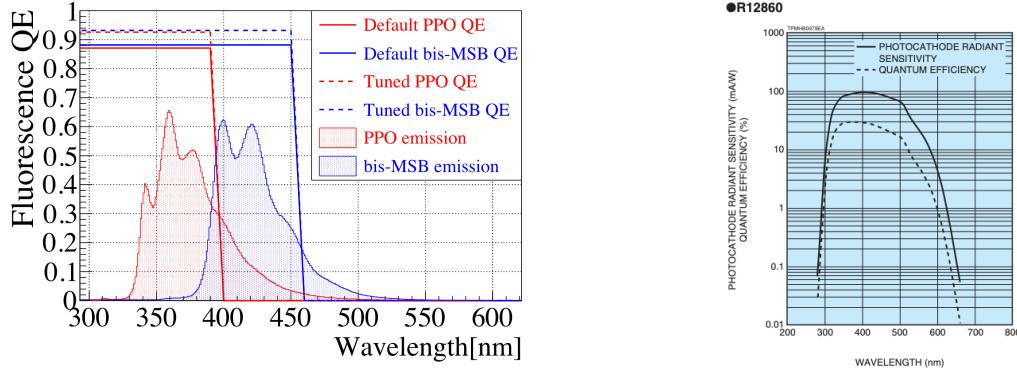


FIGURE 2.7 – On the left: Quantum efficiency (QE) and emission spectrum of the LAB and the bis-MSB [18]. On the right: Sensitivity of the Hamamatsu LPMT depending on the wavelength of the incident photons [20].

- 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) [23] to ensure that the requirements are kept during the lifetime of the experiment, more details to be found in section 2.5.2.

429 Large Photo-Multipliers Tubes (PMTs)

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

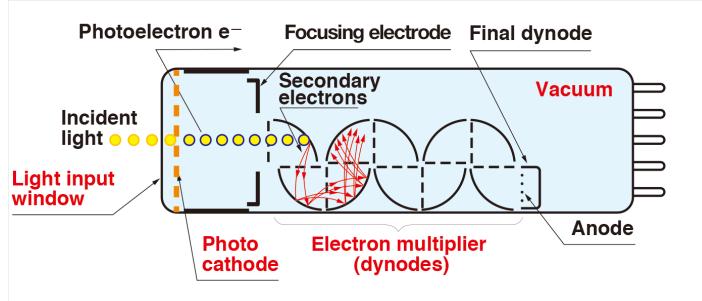


FIGURE 2.8 – Schematic of a PMT

- 437 The Large Photo-Multipliers Tubes (LPMT), used in the central detector and in the water pool, are
 438 20-inch (50.8 cm) radius PMTs. ~ 5000 dynode-PMTs [20] were produced by the Hamamatsu[®]
 439 company and ~ 15000 Micro-Channel Plate (MCP) [24] by the NNVT[®] company. This system is
 440 the one responsible for the energy measurement with a energy resolution of $3\%/\sqrt{E}$, resolution
 441 necessary for the mass ordering measurement. To reach this precision, the system is composed of
 442 17612 PMTs quasi uniformly distributed over the detector for a coverage of 75.2% reaching ~ 1800

443 PE/MeV or $\sim 2.3\%$ resolution due to statistic, leaving $\sim 0.7\%$ for the systematic uncertainties. They
 444 are located outside the acrylic sphere in the water pool facing the center of the detector. To maintain
 445 the resolution over the lifetime of the experiment, JUNO require a failure rate $< 1\%$ over 6 years.

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

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

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

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

- 457 — Three high voltage units, each one powering a PMT.
- 458 — 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... This FPGA also act as a data buffer when needed by the DAQ and trigger system.
- 460 — 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).

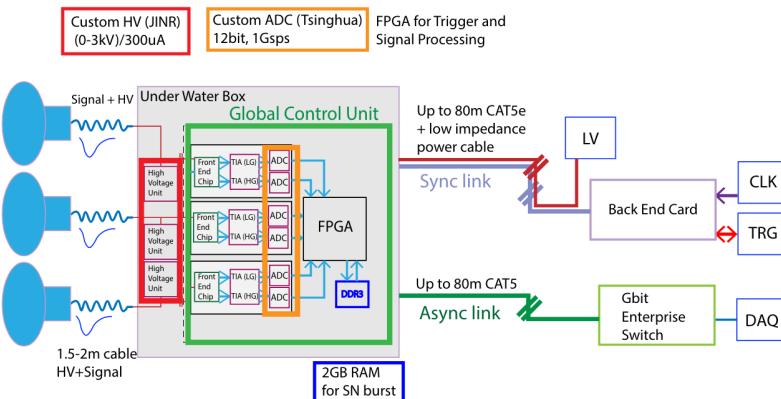


FIGURE 2.9 – The LPMT electronics scheme. It is composed of two part, the *wet* electronics on the left, located underwater and the *dry* electronics on the right. They are connected by Ethernet cable for data transmission and a dedicated low impedance cable for power distribution

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

467 Small Photo-Multipliers Tubes (SPMTs)

468 The Small PMT (SPMTs) system is made of 3-inch (7.62 cm) PMTs. They will be used in the CD
 469 as a secondary detection system. Those 25600 SPMTs will observe the same events as the LPMTs,
 470 thus sharing the physics and detector systematics up until the photon conversion. With a detector

coverage of 2.7%, this system will collect ~ 43 PE/MeV for a final energy resolution of $\sim 17\%$. This resolution is not enough to measure the NMO, θ_{13} , Δm_{31}^2 but will be sufficient to independently measure θ_{12} and Δm_{21}^2 .

The benefit of this second system is to be able to perform another, independent measure of the same events as the LPMTs, constituting the Dual Calorimetry useful for calibration 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 intrinsic effect (see section 2.4). Using this property, the intrinsic charge non linearity of the LPMTs can be measured by comparing the PE count in the SPMTs and LPMTs [25]. 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 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 transmission of the data to the DAQ.

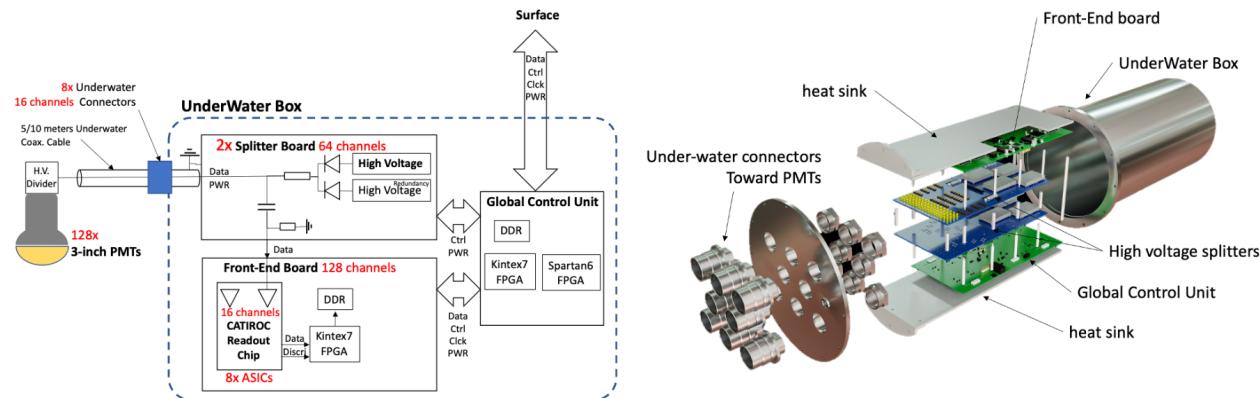


FIGURE 2.10 – Schematic of the JUNO SPMT electronic system (left), and exploded view of the main component of the UWB (right)

2.3.3 Veto detector

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

Cherenkov in water pool

The Water Cherenkov Detector (WCD) is the instrumentation of the water buffer around the CD. When high speed charged particles will pass through the water, they will produce cherenkov photons. The light will be collected by 2400 MCP LPMTs installed on the outer surface of the CD

498 structure. The muons veto strategy is based on a PMT multiplicity condition. WCD PMTs are
 499 grouped in ten zones: 5 in the top, 5 in the bottom. A veto is raised either when more than 19
 500 PMTs are triggered in one zone or when two adjacent zones simultaneously trigger more than 13
 501 PMTs. Using this trigger, we expect to reach a muon detection efficiency of 99.5% while keeping the
 502 noise at reasonable level.

503 Top tracker

504 The JUNO Top Tracker (TT) is a plastic scintillator detector located on the top of the experiment (see
 505 figure 2.11). Made from plastic scintillator from OPERA [26] layered horizontally in 3 layers on the
 506 top of the detector, the TT will be able to detect incoming atmospheric muons. With its coverage,
 507 about 1/3 of the of all atmospheric muons that passing through the CD will also pass through the 3
 508 layer of the detector. While it does not cover the majority of the CD, the TT is particularly effective
 509 to detect muons coming through the filling chimney region which might present difficulties from the
 other subsystems in some classes of events.

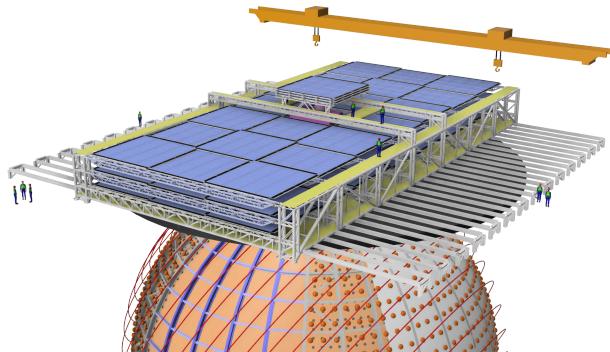


FIGURE 2.11 – The JUNO top tracker

510

511 2.4 Calibration strategy

512 The calibration is a crucial part of the JUNO experiment. The detector will continuously bath in
 513 neutrinos coming from the close nuclear power plant, from other sources such as geo neutrinos,
 514 the sun and will be exposed to background noise coming from atmospheric muons and natural
 515 radioactivity. Because of this continuous rate, low frequency signal event, we need high frequency,
 516 recognisable sources in the energy range of interest : [0-12] MeV for the positron signal and 2.2 MeV
 517 for the neutron capture. It is expected that the CD response will be different depending on the type
 518 of particle, due to the interaction with LS, the position on the event and the optical response of the
 519 acrylic sphere (see section 2.7). We also expect a non-linear energy response of the CD due to the LS
 520 properties [18] but also due to the reponse of the LPMTs system when collecting a large amount of
 521 PE [25].

522 2.4.1 Energy scale calibration

523 While electrons and positrons sources would be ideal, for a large LS detector thin-walled electrons
 524 or positrons sources could lead to leakage of radionucleides causing radioactive contamination.

525 Instead, we consider gamma sources in the range of the prompt energy of IBDs. The sources are
 526 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
$^{241}\text{Am-Be}$	n, γ	neutron + 4.43 MeV (12C*)
$^{241}\text{Am-}^{13}\text{C}$	n, γ	neutron + 6.13 MeV (16O*)
$(n, \gamma)p$	γ	2.22 MeV
$(n, \gamma)^{12}\text{C}$	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

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

531 From this calibration we call E_{vis} the "visible energy" that is reconstructed by our current algorithms
 532 and we compare it to the true energy deposited by the calibration source. The results shown in figure
 533 2.12 show the expected response of the detector from calibration sources. The non-linearity is clearly
 534 visible from the $E_{\text{vis}}/E_{\text{true}}$ shape. See [27] for more details.

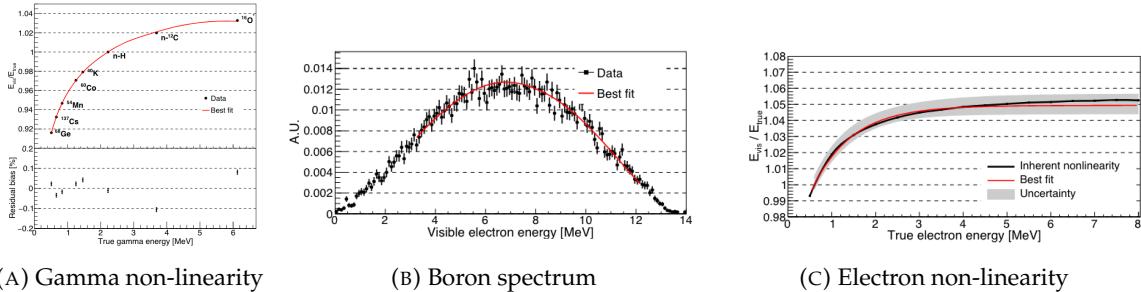


FIGURE 2.12 – Fitted and simulated non linearity of gamma, electron sources and from the ^{12}B spectrum. Black points are simulated data. Red curves are the best fits. Figures taken from [27].

535 2.4.2 Calibration system

536 The non-uniformity due to the event position in the detector (more details in section 2.7) will be
 537 studied using multiples systems that are schematized in figure 2.13. They allow to position sources
 538 at different location in the CD.

- 539 — For a one-dimension vertical calibration, the Automatic Calibration Unit (ACU) will be able
 540 to deploy multiple radioactive sources or a pulse laser diffuser ball along the central axis of
 541 the CD through the top chimney. The source position precision is less than 1cm.
- 542 — For off-axis calibration, a calibration source attached to a Cable Loop System (CLS) can be
 543 moved on a vertical half-plane by adjusting the length of two connection cable. Two set of
 544 CSL will be deployed to provide a 79% effective coverage of a vertical plane.

- 545 — A Guiding Tube (GT) will surround the CD to calibrate the non-uniformity of the response at
 546 the edge of the detector
 547 — A Remotely Operated under-LS Vehicle (ROV) can be deployed to desired location inside LS
 548 for a more precise and comprehensive calibration. The ROV will also be equipped with a
 549 camera for inspection of the CD.

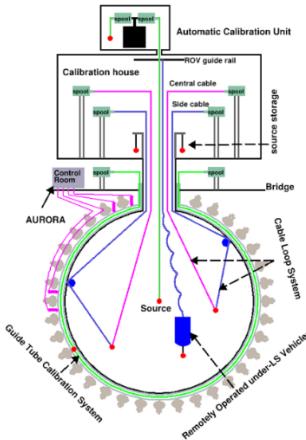


FIGURE 2.13 – Overview of the calibration system

550 The preliminary calibration program is depicted in table 2.5.

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

551 2.4.3 Instrumental non-linearity calibration

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

554 In short, during a typical IBD event, between 0 and 100 PEs can be produced in a given LPMT
 555 (depending on the position of the interaction and the positron energy). This is a large dynamic range.
 556 When the number of PEs is high, the reconstruction of the LPMT charge can become inaccurate,
 557 underestimating the actual number of PEs as illustrated in figure 2.14. This QNL is difficult to

558 separate from other non linearities (like the non linearity in the LS photon yield as a function of
 559 the deposit energy). In chapter 5 and 6 of this thesis [25], a calibration method that constitutes the
 560 core of dual calorimetry are described. They are based on the comparisons between signals seen in
 561 LPMTs and signals seen in SPMTs. In the latter system, due to its small angular coverage, individual
 562 SPMT rarely see more than 1 PE per event, and therefore are essentially immune against QNL. The
 563 method described in [25] uses a tunable light source covering the range of 0 to 100 PE perLPMT
 564 channel

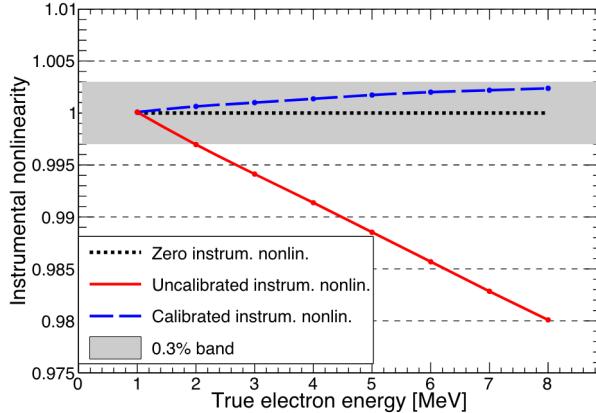


FIGURE 2.14 – Event-level instrumental non-linearity, defined as the ratio of the total measured LPMT charge to the true charge for events at the center of the detector. The solid red line represents event-level non-linearity without the channel-level correction in an extreme hypothetical scenario of 50% non-linearity over 100 PEs for the LPMTs. The dashed blue line represents that after the channel-level correction. The gray band shows the residual uncertainty of 0.3%, after the channel-level correction. Figure taken from [27].

565 2.5 Satellite detectors

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

570 2.5.1 TAO

571 The Taishan Antineutrino Observatory (TAO) [13, 28] is a ton-level gadolinium doped liquid scin-
 572 tillator detector that will be located near the Taishan-1 reactor. It aim to measure the $\bar{\nu}_e$ spectrum at
 573 very low distance (44m) from the reactor to measure a quasi-unoscillated spectrum. TAO also aim to
 574 provide a major contribution to the so-called reactor anomaly [12]. Its requirement are to the level of
 575 2 % energy resolution at 1 MeV.

576 Detector

577 The TAO detector is close, in concept, to the CD of JUNO. It is composed of an acrylic vessel
 578 containing 2.8 tons of gadolinium-loaded LS instrumented by an array of silicon photomultipliers

(SiPM) reaching a 95% coverage. To efficiently reduce the dark count of those sensors, the detector is cooled to -50 °C. The $\bar{\nu}_e$ will interact with the LS via IBD, producing scintillation light, that will be detected by the SiPMs. From this signal the $\bar{\nu}_e$ energy and the full spectrum reconstructed. This spectrum will then be used by JUNO to calibrate the unoscillated spectrum, most notably the fission product fraction that impact the rate and shape of the spectrum. A schema of the detector is presented in figure 2.15a.

2.5.2 OSIRIS

The Online Scintillator Internal Radioactivity Investigation System (OSIRIS) [23] is an ultralow background, 20 m³ 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 $^{214}\text{Bi} - ^{214}\text{Po}$ and $^{212}\text{Bi} - ^{212}\text{Po}$, 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.

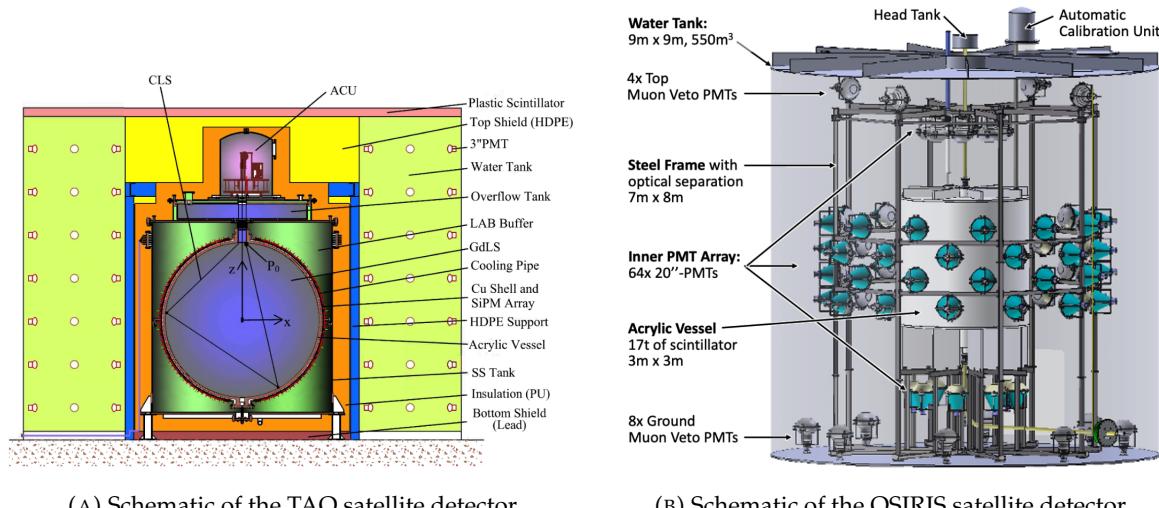


FIGURE 2.15

2.6 Software

The simulation, reconstruction and analysis algorithms are all packaged in the JUNO software, subsequently called the software. It is composed of multiple components integrated in the SNiPER [29] framework:

- 603 — Various primary particles simulators for the different kind of events, background and calibration
sources.
- 604 — A Geant4 [30–32] Monte Carlo (MC) simulation containing the detectors geometries, a custom
605 optical model for the LS and the supporting structures of the detectors. The Geant4 simulation
606 integrate all relevant physics process for JUNO, validated by the collaboration. This step of the
607 simulation is commonly called *Detsim* and compute up to the production of photo-electrons
608 in the PMTs. The optics properties of the different materials and detector components have
609 been measured beforehand to be used to define the material and surfaces in the simulation.
- 610 — An electronic simulation, simulating the response waveform of the PMTs, tracking it through
611 the digitization process, accounting for effects such as non-linearity, dark noise, Time Transit
612 Spread (TTS), pre-pulsing, after-pulsing and ringing if the waveform. It's also the step
613 handling the event triggers and mixing. This step is commonly referenced as *Elecsim*.
- 614 — A waveform reconstruction where the digitized waveform are filtered to remove high-frequency
615 white noise and then deconvoluted to yield time and charge informations of the photons hits
616 on the PMTs. This step is commonly referenced as *Calib*.
- 617 — The charge and time informations are used by reconstruction algorithms to reconstruct the
618 interaction vertex and the deposited energy. This step is commonly reported as *Reco*. See
619 section 2.7 for more details on the reconstruction.
- 620 — Once the singular events are reconstructed, they go through event pairing and classification
621 to select IBD events. This step is named Event Classification.
- 622 — The purified signal is then analysed by the analysis framework which depend of the physics
623 topic of interest.
- 624

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

630 2.7 State of the art of the Offline IBD reconstruction in JUNO

631 The main reconstruction method currently run in JUNO is a data-driven method based on a like-
632 lihood maximization [33, 34] using only the LPMTs. The first step is to reconstruct the interaction
633 vertex from which the energy reconstruction is dependent. It is also necessary for event pairing and
634 classification.

635 2.7.1 Interaction vertex reconstruction

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

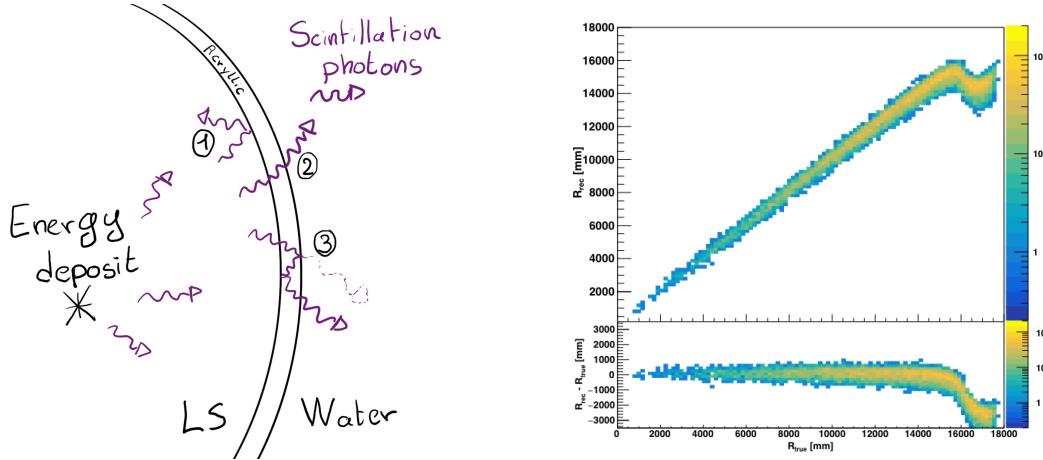
638 Charge based algorithm

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

$$\vec{r}_{cb} = a \cdot \frac{\sum_i q_i \cdot \vec{r}_i}{\sum_i q_i} \quad (2.5)$$

Where q_i is the reconstructed charge of the pulse of the i th 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 $a \approx 1.3$ [35]. 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 to be located closer to the center than the event really is.



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

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

FIGURE 2.16

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 2.6) 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 i th PMT using

$$\Delta t_i(j) = t_i - \text{tof}_i(j) \quad (2.6)$$

where j is the iteration step, t_i is the timing of the i th 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 [35] for detailed description of this effective velocity). Plot the Δt distribution and label the peak position as Δt^{peak} (see fig 2.17a).

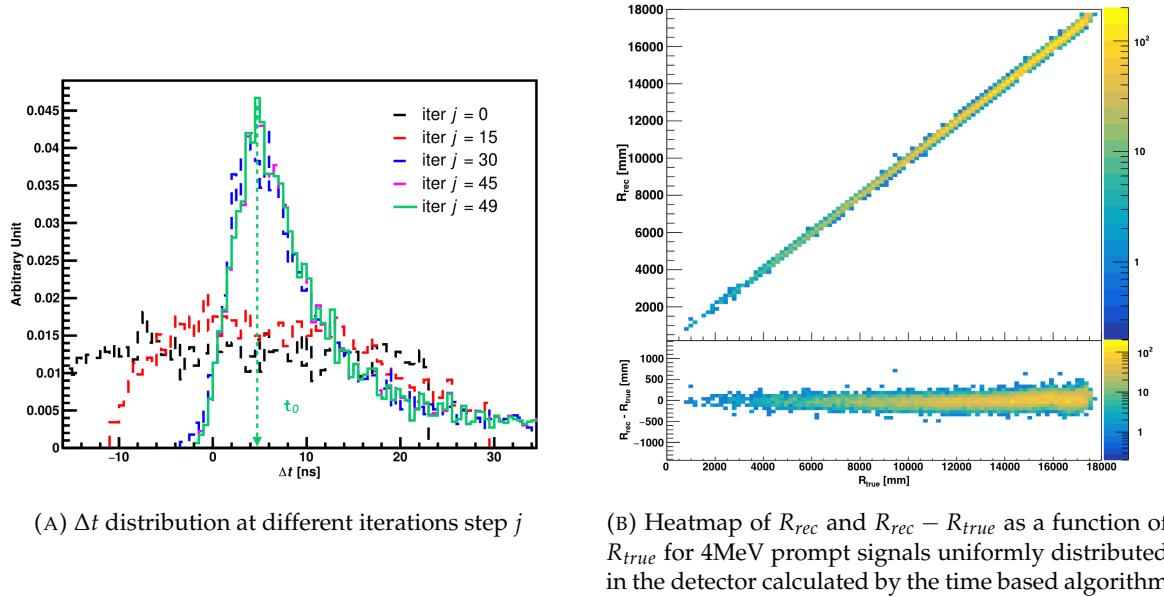
(A) Δt distribution at different iterations step j (B) Heatmap of R_{rec} and $R_{rec} - R_{true}$ as a function of R_{true} for 4MeV prompt signals uniformly distributed in the detector calculated by the time based algorithm

FIGURE 2.17

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

$$\vec{\delta}[\vec{r}(j)] = \frac{\sum_i \left(\frac{\Delta t(j) - \Delta t^{\text{peak}}(j)}{\text{tof}_i(j)} \right) \cdot (\vec{r}_0(j) - \vec{r}_i)}{N^{\text{peak}}(j)} \quad (2.7)$$

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

667 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
668 step 2.

669 However because the earliest arrival time is used, t_i is related to the number photoelectrons N_i^{pe}
670 detected by the PMT [36–38]. To reduce bias in the vertex reconstruction, the following equation is
671 used to correct t_i into t'_i :

$$t'_i = t_i - p_0 / \sqrt{N_i^{\text{pe}}} - p_1 - p_2 / N_i^{\text{pe}} \quad (2.8)$$

672 The parameters (p_0, p_1, p_2) were optimized to (9.42, 0.74, -4.60) for Hamamatsu PMTs and (41.31,
673 -12.04, -20.02) for NNVT PMTs [35]. The results presented in figure 2.17b shows that the time based
674 algorithm provide a more accurate vertex and is unbiased even in the TR area. This results (\vec{r}_0, t_0) is
675 used as initial value for the likelihood algorithm.

676 Time likelihood algorithm

677 The time likelihood algorithm use the residual time expressed as follow

$$t_{\text{res}}^i(\vec{r}_0, t_0) = t_i - \text{tof}_i - t_0 \quad (2.9)$$

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

$$f(t_{\text{res}}) = \sum_k \frac{\rho_k}{\tau_k} e^{-\frac{t_{\text{res}}}{\tau_k}}, \sum_k \rho_k = 1 \quad (2.10)$$

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

$$g(t_{\text{res}}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t_{\text{res}}-\nu)^2}{2\sigma^2}} \cdot f(t_{\text{res}}) \quad (2.11)$$

683 where σ is the TTS of PMTs and ν is the average transit time. The dark noise is not correlated with any
 684 physical events and considered as constant rate over the time window considered T . By normalizing
 685 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

$$p(t_{\text{res}}) = (1 - \epsilon_{\text{dn}}) \cdot g(t_{\text{res}}) + \epsilon(t_{\text{res}}) \quad (2.12)$$

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

$$\mathcal{L}(\vec{r}_0, t_0) = -\ln \left(\prod_i p(t_{\text{res}}^i) \right) \quad (2.13)$$

688 The parameter of Eq. 2.12 can be measured experimentally. The results shown in figure 2.18 used
 689 PDF from monte carlo simulation. The results shows that $R_{\text{rec}} - R_{\text{true}}$ is biased depending on the
 690 energy. While this could be corrected using calibration, another algorithm based on charge likelihood
 691 was developed to correct this problem.

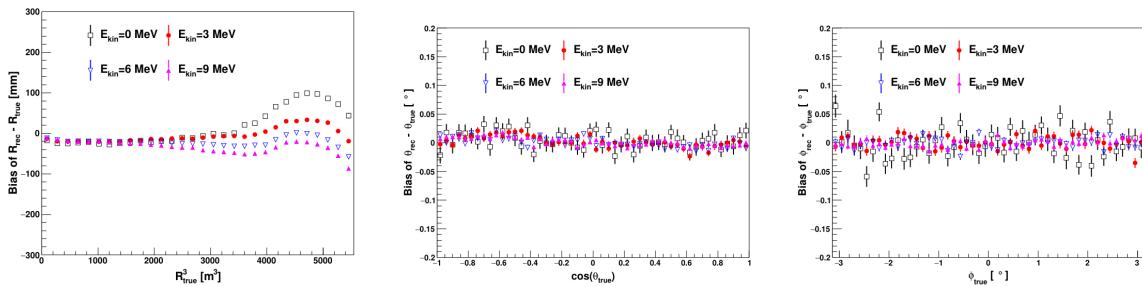


FIGURE 2.18 – Bias of the reconstructed radius R (left), θ (middle) and ϕ (right) for multiple energies by the time likelihood algorithm

692 Charge likelihood algorithm

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

- The probability to observe no hit ($N_{pe} = 0$) in the j th PMT is $P_{\text{nohit}}^j(\vec{r}_0, E) = e^{-\mu_j}$
- The probability to observe $N_{pe} \neq 0$ in the i th PMT is $P_{\text{hit}}^i(\vec{r}_0, E) = \frac{\mu^{N_{pe}^i} e^{-\mu_i}}{N_{pe}^i!}$

⁶⁹⁹ Therefore, the probability to observe a specific hit pattern can be expressed as

$$P(\vec{r}_0, E) = \prod_j P_{nohit}^j(\vec{r}_0, E) \cdot \prod_i P_{hit}^i(\vec{r}_0, E) \quad (2.14)$$

⁷⁰⁰ The best fit values of \vec{R}_0 and E can then be calculated by minimizing the negative log-likelihood

$$\mathcal{L}(\vec{r}_0, E) = -\ln(P(\vec{r}_0, E)) \quad (2.15)$$

⁷⁰¹ In principle, $\mu_i(\vec{r}_0, E)$ could be expressed

$$\mu_i(\vec{r}_0, E) = Y \cdot \frac{\Omega(\vec{r}_0, r_i)}{4\pi} \cdot \epsilon_i \cdot f(\theta_i) \cdot e^{-\sum_m \frac{d_m}{\zeta_m}} \cdot E + \delta_i \quad (2.16)$$

⁷⁰² where Y is the energy scale factor, $\Omega(\vec{r}_0, r_i)$ is the solid angle of the i th PMT, ϵ_i is its detection
⁷⁰³ efficiency, $f(\theta_i)$ its angular response, ζ_m is the attenuation length in the materials and δ_i the expected
⁷⁰⁴ number of dark noise.

⁷⁰⁵ However Eq. 2.16 assume that the scintillation light yield is linear with energy and describe poorly
⁷⁰⁶ the contribution of indirect light, shadow effect due to the supporting structure and the total reflec-
⁷⁰⁷ tion effects. The solution is to use data driven methods to produce the pdf by using the calibrations
⁷⁰⁸ sources and position described in section 2.4. In the results presented in figures 2.19, the PDF was
⁷⁰⁹ produced using MC simulation and 29 specific calibrations position [35] along the Z-axis of the
 detector. We see that the charge likelihood algorithm show little bias in the TR area and a better

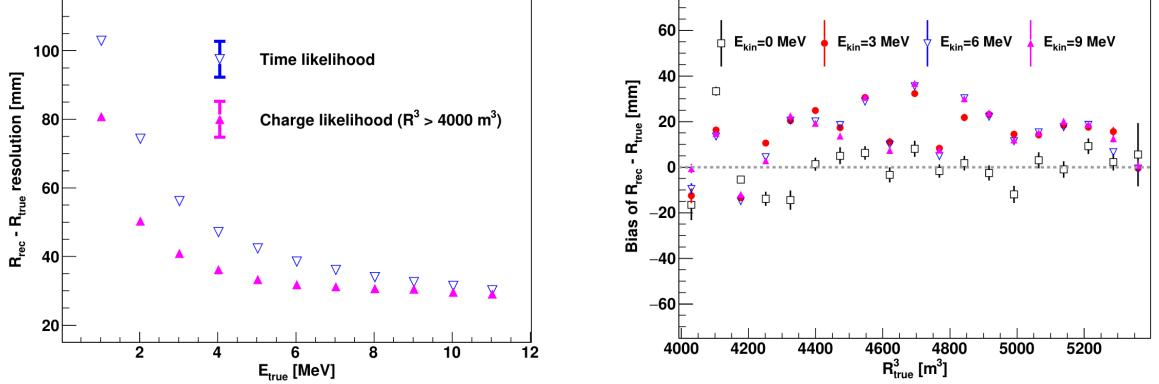


FIGURE 2.19 – On the left: Resolution of the reconstructed R as a function of the energy in the TR area ($R^3 > 4000 \text{ m}^3 \equiv R > 16 \text{ m}$) by the charge and time likelihood algorithms. On the right: Bias of the reconstructed R in the TR area for different energies by the charge likelihood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

⁷¹⁰ resolution than the time likelihood. The figure 2.20 shows the radial resolution of the different
⁷¹¹ algorithm presented for this section, we can see the refinement at each step and that the charge
⁷¹² likelihood yield the best results.

⁷¹⁴ The charge based likelihood algorithms already give use some information on the energy as Eq. 2.15
⁷¹⁵ is minimized but the energy can be further refined as shown in the next section.

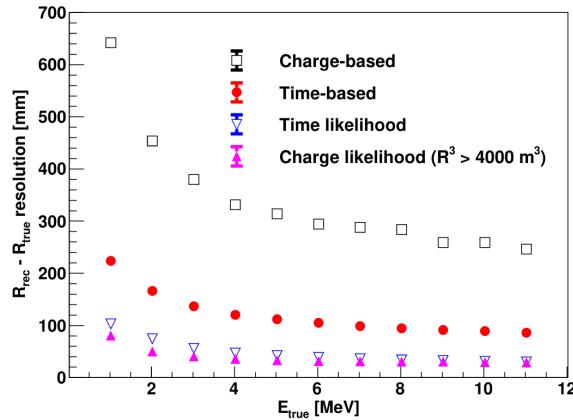
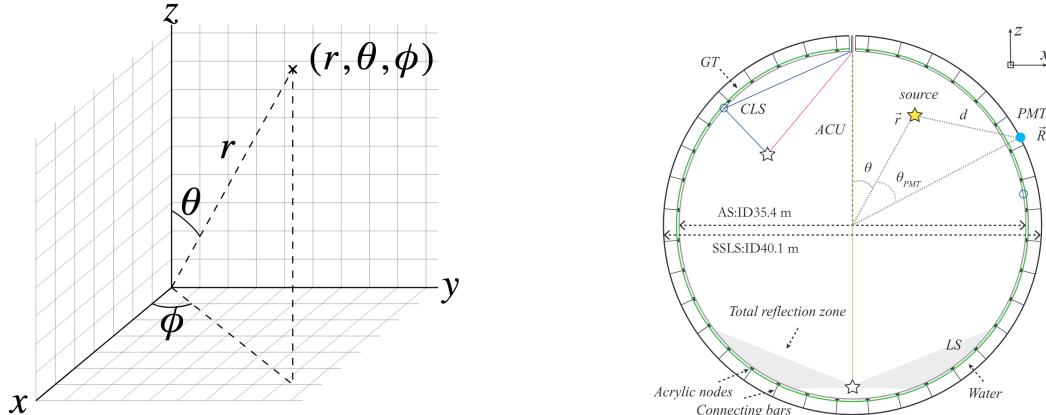


FIGURE 2.20 – Radial resolution of the different vertex reconstruction algorithms as a function of the energy

2.7.2 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 [39].



(A) Spherical coordinate system used in JUNO for reconstruction

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

FIGURE 2.21

Charge estimation

The most important element in the energy reconstruction is $\mu_i(\vec{r}_0, E)$ described in Eq. 2.16. 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

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

$$\hat{\mu}(r, \theta, \theta_{pmt}, E_{vis}) = \frac{1}{E_{vis}} \frac{1}{M} \sum_i^M \frac{\bar{Q}_i - \mu_i^D}{\text{DE}_i}, \mu_i^D = \text{DNR}_i \cdot L \quad (2.17)$$

728 where i runs over the PMTs with the same θ_{pmt} , DE_i is the detection efficiency of the i th PMT. μ_i^D
 729 is the expected number of dark noise photoelectrons in the time window L . The time window have
 730 been optimized to $L = 280$ ns [39]. \bar{Q}_i is the average recorded photoelectrons in the time window
 731 and \hat{Q}_i is the expected average charge for 1 photoelectron. The N_{pe} map is constructed following the
 732 procedure described in [34].

733 Time estimation

734 The second important observable is the hit time of photons that was previously defined in Eq. 2.9. It
 735 is here refined as

$$t_r = t_h - \text{tof} - t_0 = t_{LS} + t_{TT} \quad (2.18)$$

736 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
 737 by a gaussian

$$t_{TT} = \mathcal{N}(\bar{\mu}_{TT} + t_d, \sigma_{TT}) \quad (2.19)$$

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

741 The timing PDF $P_T(t_r | r, d, \mu_l, \mu_d, k)$ can now be generated using calibration sources [39]. This PDF
 742 describe the probability that the residual time of the first photon hit is in $[t_r, t_r + \delta]$ with r the radius
 743 of the event vertex, $d = |\vec{r} - \vec{r}_{PMT}|$ the propagation distance, μ_l and μ_d the expected number of PE
 744 and dark noise in the electronic reading window and k is the detected number of PE.

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

$$F(t, r, d) = \int_t^L f(t', r, d) dt' \quad (2.20)$$

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

$$P_T^l(t | k = n) = I_n^l [f_l(t) F_l^{n-1}(t)] \quad (2.21)$$

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

$$f_d(t) = 1/L, F_d(t) = 1 - \frac{t}{L} \quad (2.22)$$

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

$$P_T(t | \mu_l, \mu_d, k = 1) = I_1 [P(1, \mu_l) P(0, \mu_d) f_l(t) + P(0, \mu_l) P(1, \mu_d) f_d(t)] \quad (2.23)$$

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

754 Now that we have the individual timing and charge probability we can construct the charge likeli-

hood referred as QMLE:

$$\mathcal{L}(q_1, q_2, \dots, q_N | \vec{r}, E_{vis}) = \prod_{j \in \text{unfired}} e^{-\mu_j} \prod_{i \in \text{fired}} \left(\sum_{k=1}^K P_Q(q_i | k) \cdot P(k, \mu_i) \right) \quad (2.24)$$

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:

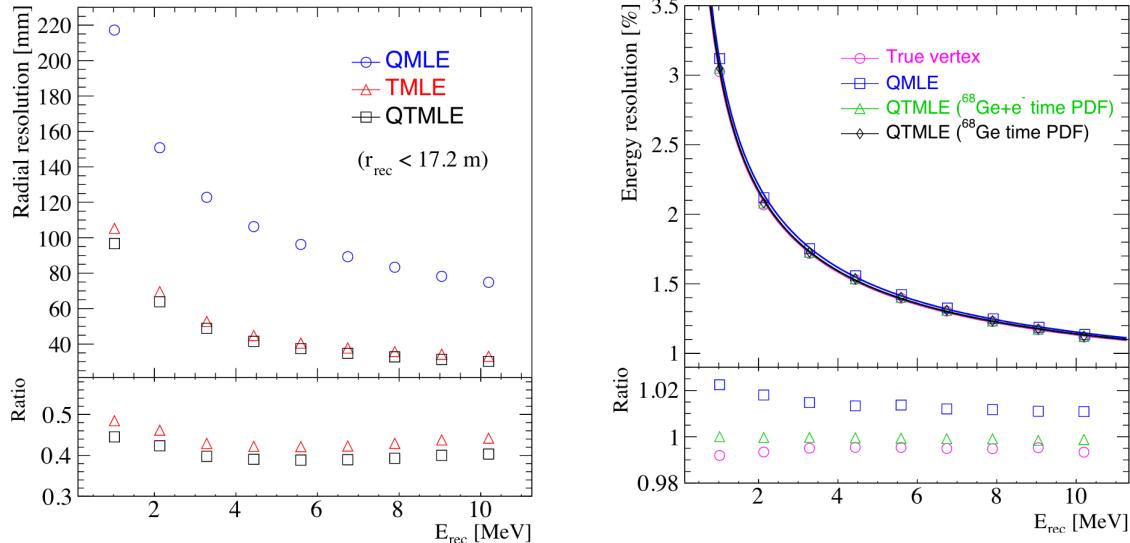
$$\mathcal{L}(t_{1,r}, t_{2,r}, \dots, t_{N,r} | \vec{r}, t_0) = \prod_{i \in \text{hit}} \frac{\sum_{k=1}^K P_T(t_{i,r} | r, d, \mu_i^l, \mu_i^d, k) \cdot P(k, \mu_i^l + \mu_i^d)}{\sum_{k=1}^K P(k, \mu_i^l + \mu_i^d)} \quad (2.25)$$

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 QTMLLE

$$\mathcal{L}(q_1, q_2, \dots, q_N; t_{1,r}, t_{2,r}, \dots, t_{N,r} | \vec{r}, t_0, E_{vis}) = \mathcal{L}(q_1, q_2, \dots, q_N | \vec{r}, E_{vis}) \cdot \mathcal{L}(t_{1,r}, t_{2,r}, \dots, t_{N,r} | \vec{r}, t_0) \quad (2.26)$$

The radial and energy resolutions of the different likelihood are presented in figure 2.22 (from [39]). We can see the improvement of adding the time information to the vertex reconstruction and that an increase in vertex precision can bring improvement in the energy resolution, especially at low energies.



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

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

FIGURE 2.22

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.

2.7.3 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.

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:

$$\mathbb{R}^{5 \times N_{pmt}} \mapsto \mathbb{R}^4 \quad (2.27)$$

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

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

Boosted Decision Tree (BDT)

On of the most classic ML method used in physics in last years is the Boosted Decision Tree (see Chapter 3.1.1). They have been explored for vertex reconstruction [40] et for energy reconstruction [40, 41].

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

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

Its reconstruction performances are presented in figure 2.24.

A second and more advanced BDT, subsequently named BDTE, that only reconstruct energy use a different set of features [41]. 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 [42–45]. 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.

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

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 [40], 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.

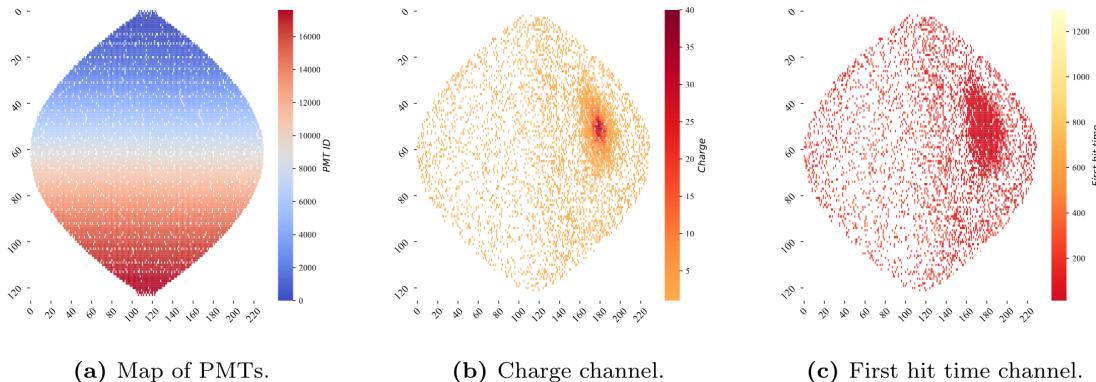


FIGURE 2.23 – Projection of the LPMTs in JUNO on a 2D plane. (a) Show the distribution of all PMTs and (b) and (c) are example of what the charge and time channel looks like respectively

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 [40] the nodes, are geometrical region of the detector as defined by the HealPix [46]. The features of the nodes are aggregated informations from the PMTs it contains. The edges contains geographic informations of the nodes relative positions.

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

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

820 of the GNN are presented in figure 2.24.

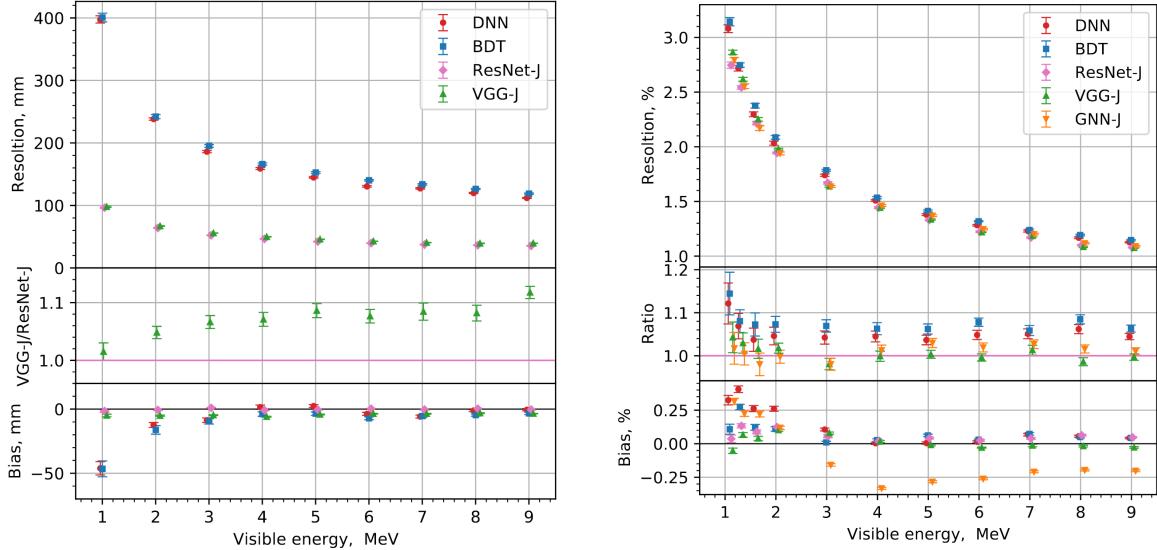


FIGURE 2.24 – Radial (left) and energy (right) resolutions of different ML algorithms. The results presented here are from [40]. DNN is a deep neural network, BDT is a BDT, ResNet-J and VGG-J are CNN and GNN-J is a GNN.

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

825 2.7.4 Physics results

826 The oscillation parameters are directly extracted from the minimization procedure and the error can
827 be estimated directly from the procedure. For the NMO, the data are fitted under the two assumption
828 of NO and IO. The difference in χ^2 give us the preferred ordering and the significance of our test.
829 Latest studies show that the precision on oscillation parameters after six year of data taking will be
830 of 0.2%, 0.3%, 0.5% and 12.1% for Δm_{31}^2 , Δm_{21}^2 , $\sin^2 \theta_{12}$ and $\sin^2 \theta_{13}$ respectively [3]. The expected
831 sensitivity to mass ordering is 3σ after 6.5 years [48].

832 2.8 Summary

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

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

⁸⁴² **Chapter 3**

⁸⁴³ **Machine learning: Introduction to the
methods and algorithms used in this
thesis**

⁸⁴⁴

⁸⁴⁵

⁸⁴⁶

"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 prostheticized 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

⁸⁴⁷

Contents

⁸⁴⁸

⁸⁴⁹

⁸⁵⁰

⁸⁵¹

⁸⁵²

⁸⁵³

⁸⁵⁴

⁸⁵⁵

⁸⁵⁶

⁸⁵⁷

⁸⁵⁸

⁸⁵⁹

⁸⁶⁰

⁸⁶¹

⁸⁶²

⁸⁶³

⁸⁶⁴

⁸⁶⁵

⁸⁶⁶

⁸⁶⁷

⁸⁶⁸

⁸⁶⁹

⁸⁷⁰

⁸⁷¹

⁸⁷²

⁸⁷³

⁸⁷⁴

⁸⁷⁵

3.1 Core concepts in machine learning and neural networks	42
3.1.1 Boosted Decision Tree (BDT)	42
3.1.2 Artificial Neural Network (NN)	42
3.1.3 Training procedure	44
3.1.4 Potential pitfalls	47
3.2 Neural networks architectures	50
3.2.1 Fully Connected Deep Neural Network (FCDNN)	50
3.2.2 Convolutional Neural Network (CNN)	50
3.2.3 Graph Neural Network (GNN)	52
3.2.4 Adversarial Neural Network (ANN)	54

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

They are used to model complex distributions from a finite dataset to extract a generalist behavior. For example, in our case, it could be an algorithm that would differentiate the nature of a particle interacting in the liquid scintillator, between a positron and an electron, based on the readout charge and time (Q, t) of the 17612 LPMT of the JUNO experiment. During a first training phase, it would learn the discriminative features between the two in the 35224-dimensional charge and time distribution, built from samples of e^+ and e^- events.

It would learn to derive from a complex, highly dimensional set of data the essential few informations characterizing the interactions: a three body energy deposition (the positron and two annihilation gammas) and the single deposit from an electron.

Ideally, the algorithm would learn to recognize those informations on its own, regardless of the input size and complexity. In practice, however, these algorithms are guided by human design through

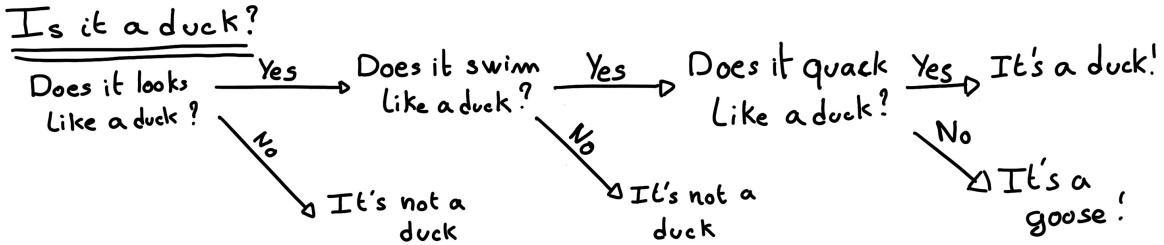


FIGURE 3.1 – Example of a BDT that determine if the given object is a duck

their architectures and training conditions. We can still hope that they can use more thoroughly the detector informations while traditional methods are often subject to assumptions or simplifications to make the task easier (see for instance the algorithm in section 2.7).

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

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

3.1 Core concepts in machine learning and neural networks

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

3.1.1 Boosted Decision Tree (BDT)

One of the most classic machine learning algorithm used in particle physics is Boosted Decision Tree (BDT) [49] (or more recently Gradient Boosting Machine [50]). 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 backpropagate corrections to the parameters of the tree. Modern BDT use gradient boosting where the gradient of the loss is calculated for each of the BDT parameters. Following the gradient descent, we can reach the, hopefully, global minima of the loss for our set of parameters.

3.1.2 Artificial Neural Network (NN)

One of the modern ML family is the Neural Network, historical name as their design was inspired by the behaviour of biological neurons in the brain. As schematized in figure 3.2, the input, output and steps inside the NN is described as neuron *layers*. The neurons of the layers take as input a

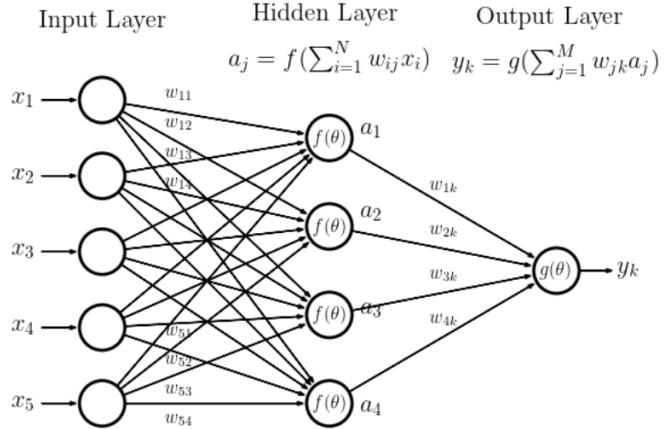


FIGURE 3.2 – Schema of a simple neural network

906 set of values from the preceding layer, here the a_i takes every informations of the x_i input layer,
 907 and aggregate those values following learnable *parameters* w_{ij} . The aggregation procedure is core of
 908 defining the architecture of the NN. The different architectures used in this thesis will be discussed
 909 in section 3.2. The process is repeated until reaching the output layer.

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

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

$$O_j = I_i + \mathbf{W}_j^i \quad (3.1)$$

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

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

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

924 This description of neural networks as layers introduce the principles of *depth* and *width*, the number
 925 of layers in the NN and the number of neurons in each layer respectively. Those quantities that not
 926 directly used for the computation of the results but describes the NN or its training are designated
 927 as *hyperparameters*.

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

932 **3.1.3 Training procedure**

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

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

$$\mathcal{L} \equiv MSE = \frac{1}{N} \sum_i^N (y_i - f(x_i, \theta))^2 \quad (3.3)$$

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

$$\mathcal{L} \equiv MAE = \frac{1}{N} \sum_i^N |y_i - f(x_i, \theta)| \quad (3.4)$$

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

943 Most of the modern neural networks use gradient descent to optimize their parameters, i.e. the
 944 gradient of the parameter w , designated in literature as θ , with respect of the loss function \mathcal{L} is
 945 subtracted each optimisation step t

$$\theta_{t+1} = \theta_t - \frac{\partial \mathcal{L}}{\partial \theta} \quad (3.5)$$

946 This induce \mathcal{L} needs to be differentiable with respect to θ , thus the layers and their activation func-
 947 tions also need to be differentiable. This simple gradient descent, designated as Stochastic Gradient
 948 Descent (SGD), can be extended with first and second order momentums like in the Adam optimizer
 949 [51]. More details about the optimizers can be found in section 3.1.3.

950 **Training lifecycle**

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

953 As illustrated in figure 3.3, the training is split into *epochs*. Each epochs is split into *step* where the
 954 NN will optimize its parameters over a *batch*, a sub-sample of the training datasets. The ideal batch
 955 size, number of event in a batch, would be the entire dataset, as the NN optimization would not be
 956 biased by the specificity of a sub-sample, but due to memory limitations the batch size is driven by
 957 technical limitations.

958 At the end of each epochs, the neural network is evaluated over a validation dataset, a dataset from
 959 which no optimisation is done. It is used as reference for the network performance as and monitor
 960 overtraining (see section 3.1.4).

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

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

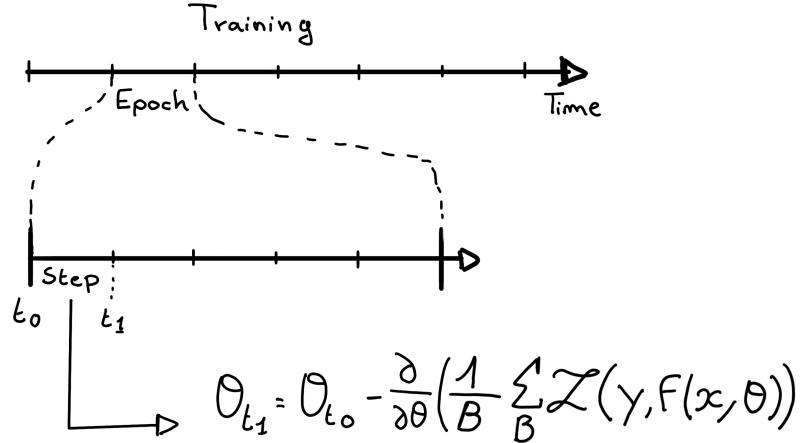


FIGURE 3.3 – Illustration of the training lifecycle

Most training are started with a fixed number of epochs, i.e. from what we've seen from precedent training, the network stop learning, the loss is constant, after N epoch so we run the training for $N + \delta$ epochs to see if the modification brings improvements to the loss profile. We can setup what's called *early stopping policies* that'll stop the training early in specific cases like loss explosion or loss stability but this require fine tuning and don't bring much in our case as we are not really limited in training time.

973 The optimizer

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

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

where θ is a parameter, i is the layer index. We see here that the gradient of the first layer is dependent of the gradient of all the following layers. Because the only value known at the start of the optimization procedure is \mathcal{L} we compute $\frac{\partial \mathcal{L}}{\partial \theta_N}$ then, $\frac{\partial \theta_N}{\partial \theta_{N-1}}$, etc... This is called the *backward propagation*.

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

1. SGD (Stochastic Gradient Descent). This is the simplest optimizer, it depend on only one hyperparameter, the learning rate λ (LR) and update the parameters θ following

$$\theta_{t+1} = \theta_t - \lambda \frac{\partial \mathcal{L}}{\partial \theta} \Big|_{\theta_t} \quad (3.7)$$

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

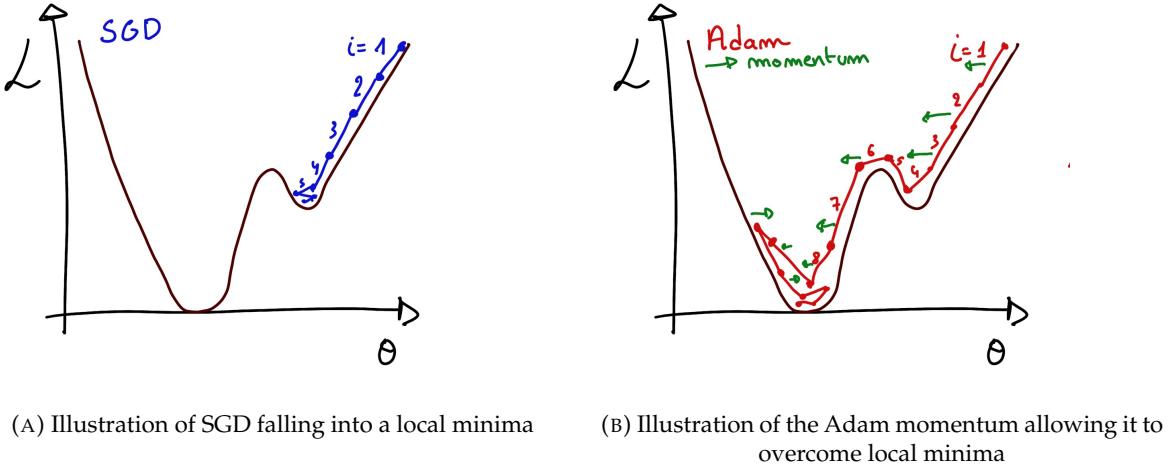


FIGURE 3.4

2. Adam [51]. 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

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

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

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

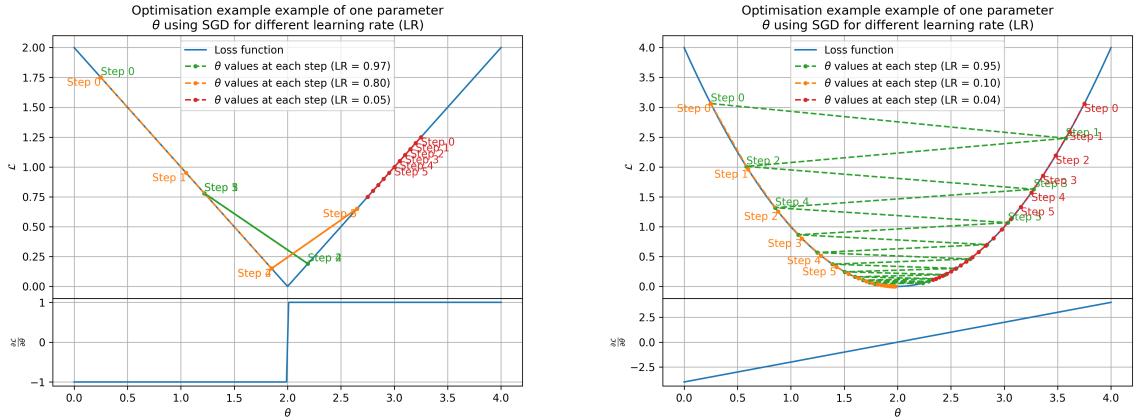
where ϵ is a small number to prevent divergence when v is close to 0. These momentums allow to overcome small local minima in the parameters phase. Imagine ball going down a slope as illustrated in 3.4a, if you ignore the stored momentum you get SGD and get stuck as on the left plot. Now if you consider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

The LR is a crucial parameter in the training of NN. You see that in case of MAE in figure 3.5a that if the LR is too high, you can end up missing the minima. Is the LR is too low, even with MSE as in figure 3.5b, you never reach the minima in the allocated number of epochs. To prevent possible issues, we setup scheduler policies.

998 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. We want to get the closest possible in early epochs before refining the training with a smaller learning rat, finer step. By reducing the learning 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



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

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

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

1006 of the model attained during the training phase.

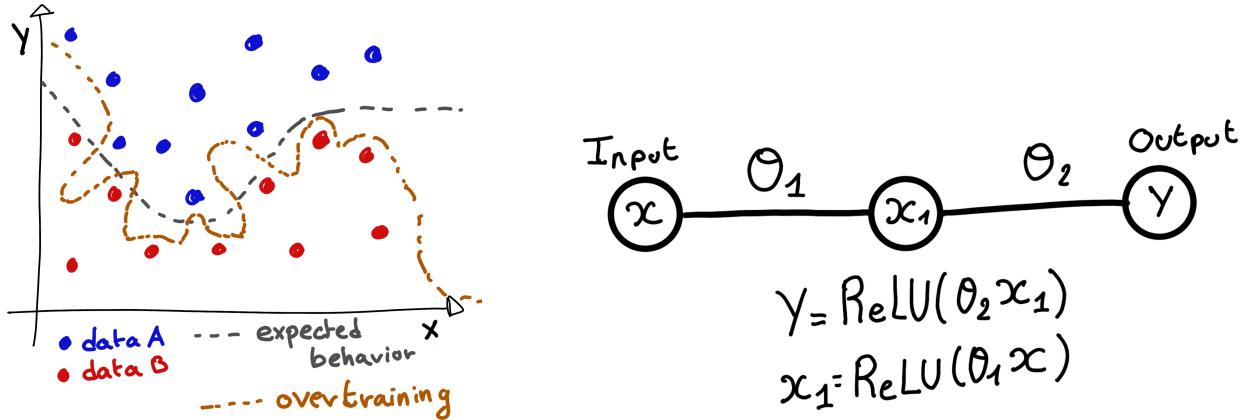
1007 3.1.4 Potential pitfalls

1008 Apart from being stuck in local minima, there is also other behaviors and effects we want to prevent
1009 during training.

1010 Overtraining

1011 This happen when the network learn the specificities of the training dataset instead of a more general
1012 representation of the underlying data distribution. This can happen if there is not enough data in
1013 comparison to the number of learning parameters, if the training data posses specific features that
1014 are not representative of the application dataset or if the NN trains for too long on the same dataset.
1015 This behavior is illustrated in figure 3.6a. Overtraining can be fought in multiple ways, for example:

- 1016 — **More data.** By having more data in the training dataset, the network will not be able the
1017 specificities of every data.
- 1018 — **Less parameters.** By reducing the number of parameters, we reduce the computing and
1019 learning capacities of the network. This will force it to fallback to generalist behaviours.
- 1020 — **Dropout.** This technique implies to randomly set some neurons to 0, i.e. cutting the relation
1021 between two neurons in a layer. By doing this, we force the network to allocate more of its
1022 parameter to the features learning, preventing those parameters to be used for overtraining.
- 1023 — **Early stopping.** During the training we monitor the network performance over a validation
1024 dataset. The network does not train on this dataset and thus cannot learn its specificities. If
1025 the loss on the training dataset diverge too much from the loss on the validation dataset, we
1026 can stop the training earlier to prevent it from overtraining.



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

(B) Illustration of a very simple NN

FIGURE 3.6

1027 Gradient vanishing

1028 Gradient vanishing is the effect of the gradient being so small for the early layers that the parameters
 1029 are barely updated after each step. This cause the network to be unable to converge to the minima.

1030 This comes from the way the gradient descent is calculated. Imagine a simple network composed of
 1031 three fully connected layers: the input layer, a intermediate layer and the output layer. Let L be the
 1032 loss, θ_1 the parameter between the input and the intermediate layer and θ_2 the parameter between
 1033 the intermediate and output layer. This network is schematized in figure 3.6b.

1034 The gradient for θ_1 will be computed using the chain rule presented in equation 3.6. Because θ_1
 1035 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
 1036 much more layer, we can see how the subsequent multiplication of small gradients would lead to
 1037 very small update of the parameters thus "vanishing gradient".

1038 Multiple actions can be taken to prevent this effect such as:

- 1039 — **Batch normalization:** In this case we apply a normalization layer that will normalize the data.
 1040 It means that we transform the input variable X into a variable D which distribution follow
 1041 $\langle D \rangle = 0$ and $\sigma_D = 1$. This helps the parameters of the network to maintain an appropriate
 1042 scale.
- 1043 — **Residual Network (ResNet)** [52]: Residual network is a technique for neural network in
 1044 which, instead of just sequentially feeding the results of each layer to the next one, you
 1045 compute a residual over the input data. This technique is illustrated in figure 3.7. The
 1046 reference [52] show empirical evidence of its relevance.

1047 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

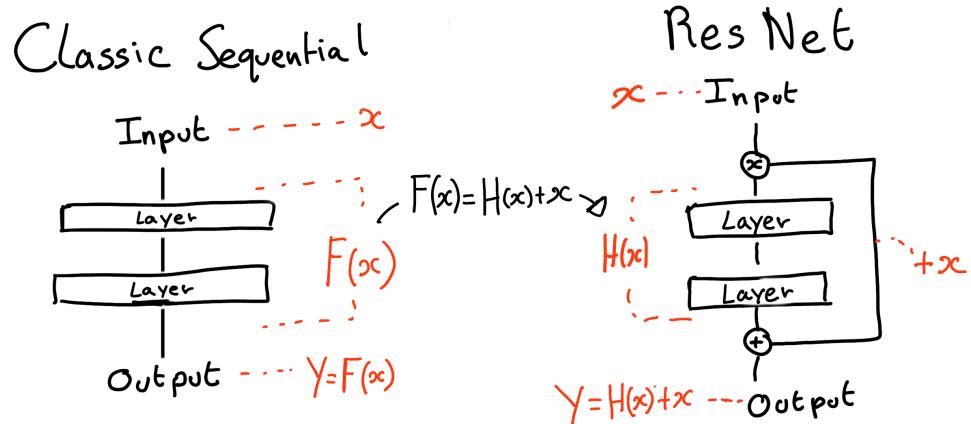


FIGURE 3.7 – Illustration of the ResNet framework

follow

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

1048 The explosion is illustrated in figure 3.8 where we can see that the loss degrades with each step of
 1049 optimization. In this illustration it is clear that reducing the learning rate suffice but this behaviour
 1050 can happens in the middle of the training where the learning rate schedule does not permit reactivity.

1051 There exist solutions to prevent this explosions:

- 1052 — **Gradient clipping:** Is this case we work on the gradient so that the norm of gradient vector
 1053 does not exceed a certain threshold. In our illustration in figure 3.8 the gradient for $\theta > 0$
 1054 could be clipped at 3 for example.
 1055 — **Batch normalization:** For the same reasons as for gradient vanishing, normalizing the input
 1056 data help reduce erratic behaviour.

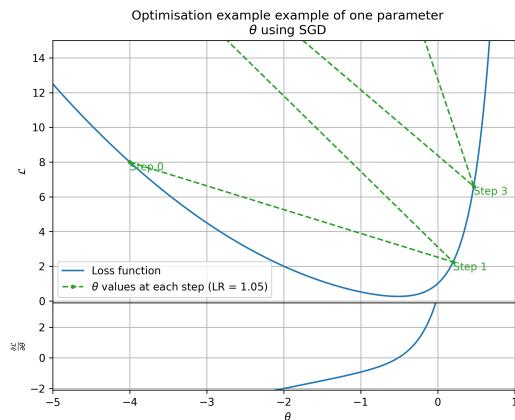


FIGURE 3.8 – Illustration of the gradient explosion. Here it can be solved with a lower learning rate but its not always the case.

1057 **3.2 Neural networks architectures**

1058 **3.2.1 Fully Connected Deep Neural Network (FCDNN)**

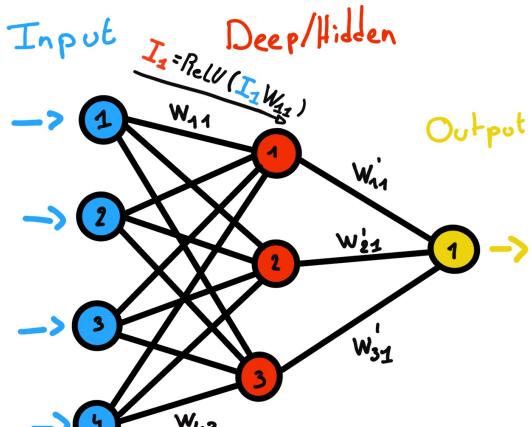
1059 The Fully Connected Deep Neural Network (FCDNN) architecture is the stack of multiple fully
 1060 connected layers as presented in the figure 3.9a. Most of the time, the classic ReLU function

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

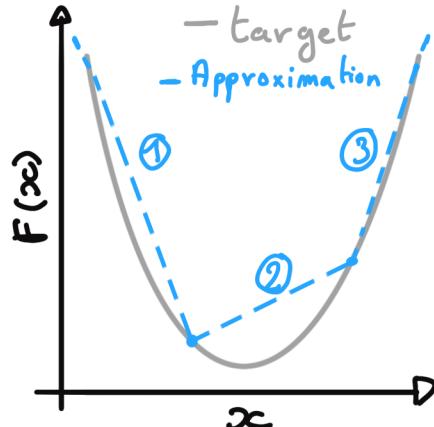
1061 is used as activation function. PReLU and Sigmoid are also popular choices:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (3.12) \quad \text{PReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{otherwise} \end{cases} \quad (3.13)$$

1063 The reasoning behind ReLU and PReLU is that with enough of them, you can mimic any continuous
 1064 function as illustrated in figure 3.9b. Sigmoid is more used in case of classification, its behavior going
 1065 hand in hand with the Cross Entropy loss function used in classification problems.



(A) Schema of a FCDNN



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

FIGURE 3.9

1066 Due to its simplicity, FCDNN are also used as basic pieces for more complex architectures such as
 1067 the CNN and GNN that will be presented in the next sections.

1068 **3.2.2 Convolutional Neural Network (CNN)**

1069 It's not trivial to describe in text the principles of Convolutional Neural Network (CNN) and how
 1070 they works. We try a general description below followed by a step by step description of a concrete
 1071 example.

1072 Convolutional Neural Networks are a family of neural networks that use discrete convolution filters,
 1073 as illustrated in an example in figure 3.10, to process the input data, often images. They are com-
 1074 monly used in image recognition [53] for classification or regression problematics. Concretely, you
 1075 multiply element-wise a portion of the input data, in the case of an image, a small part of the image,

1076 with a kernel of same dimension. In figure 3.10, we multiply the 3×3 pixels sub-image with the
 1077 3×3 kernel.

1078 Their filters scan the input data, highlighting patterns of interest, this scanning procedure making
 1079 them translation-invariant. In the concrete case of figure 3.10, for each pixel of the input image, we
 1080 group it with the 8 neighbours pixel and produce a new pixel that correspond to the output image.
 1081 For the pixel on the edges that do not have neighbours, we either create “imaginary” pixel with the
 1082 value 0 or we just ignore them. If we ignore them, the output image will posses fewer pixels than the
 1083 input image. We see that the operation do not care where is the pattern of interest in the images, the
 1084 filter output will be *invariant* whatever *translation* is applied to the image.

1085 This invariance mean that they are capable of detecting oriented features independently of their
 1086 location on the image. Again taking 3.10 as an example, with only the 9 parameters composing the
 1087 kernel, we can highlight the contour of the duck by looking at the “yellowness” of the pixels.

1088 The learning parameters of CNNs are the kernels components, the network thus learn the optimal
 1089 filters to extract the desired features.

1090 The convolution layers are commonly chained [54], reducing the input dimension while increasing
 1091 the number of filters. The idea behind is that the first layers will process local informations and
 1092 the latest layers will process more global informations, as the latest convolution filters will process
 1093 the results of the preceding that themself have processed local information. To try to preserve the
 1094 amount of information, we tend to grow the numbers of filters for each division of the input data.
 1095 The results of the convolution filters is commonly then flattened and feed to a smaller FCDNN which
 1096 will process the filters results to yield the desired output.

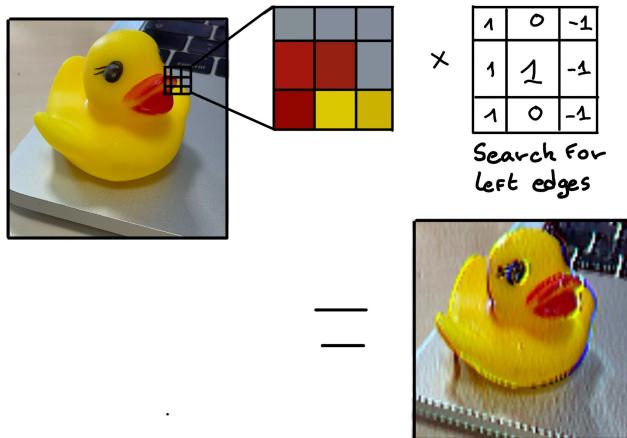


FIGURE 3.10 – Illustration of the effect of a convolution filter. Here we apply a filter with the aim do detect left edges. We see in the resulting image that the left edges of the duck are bright yellow where the right edges are dark blue indicating the contour of the object. The convolution was calculated using [55].

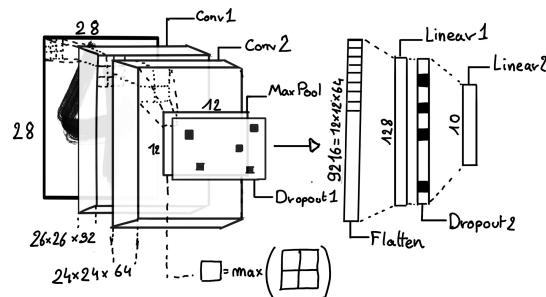
1097 As an example, let’s take the Pytorch [56] example for the MNIST [57], a dataset of black and white
 1098 images of handwritten digits. Those images are 28×28 pixels with only one channel corresponding
 1099 to the grey level of the pixel. Example of images from this dataset are presented in figure 3.11a

1100 A schema of the CNN used in the Pytorch example is presented in figure 3.11b. Using this schema
 1101 as a reference, the trained network is made of:

- 1102 1. A convolutional layer of (3×3) filters yielding 32 channels. A bias parameter is applied
 1103 to each channel for a total of $(32 \cdot (3 \times 3) + 32) = 320$ parameters. The resulting image is
 1104 $(26 \times 26 \times 32)$ (26 per 26 pixels with 32 channels). The ReLU activation function is applied to
 1105 each pixel.

- 1106 2. A second convolutional layer of (3×3) filters yielding 64 channels. This channel also posses
 1107 a bias parameter for a total of $(64 \cdot (3 \times 3) + 64) = 640$ parameters. Resulting image is $(24 \times$
 1108 $24 \times 64)$. This channel also apply a ReLU activation function.
- 1109 3. Then comes a (2×2) max pool layer with a stride of 1 meaning that for each channel the max
 1110 value of pixels in a (2×2) block is condensed in a single resulting pixel. The resulting image
 1111 is $(12 \times 12 \times 64)$.
- 1112 4. This image goes through a dropout layer which will set the pixel to 0 with a probability of
 1113 0.25. This help prevent overtraining the neural network (see section 3.1.4 for more details).
- 1114 5. The data is the flattened i.e. condensed into a vector of $(12 \times 12 \times 64) = 9216$ values.
- 1115 6. Then comes a fully connected linear layer (Eq. 3.2) with a ReLU activation that output 128
 1116 feature. It needs $(9216 \cdot 128) + 128 = 1'179'776$ parameters.
- 1117 7. This 128 item vector goes through another dropout layer with a probability of 0.5
- 1118 8. The vector is then transformed through a linear layer with ReLU activation. It output 10
 1119 values, one for each digit class $(0, 1, 2, \dots, 9)$. It need $(128 \cdot 10) + 128 = 1408$ parameters.
- 1120 9. Finally the 10 values are normalized using a log softmax function $\text{LogSoftmax}(x_i) = \log\left(\frac{\exp(x_i)}{\sum_j \exp(x_j)}\right)$.
- 1121 Each of those values are the probability of the input image to be a certain digit.

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9



(A) Example of images in the MNIST dataset

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

FIGURE 3.11

- 1122 The final network needs 1'182'144 parameters or, if we consider each parameters to be a double
 1123 precision floating point, 9.45 MB of data. To gives a order of magnitude, such neural network is
 1124 considered "simple", train in a matter of minutes on T4 GPU [58] (14 epochs) and reach an accuracy
 1125 in its prediction of 99%.

3.2.3 Graph Neural Network (GNN)

- 1127 As seen in the previous section, the CNNs are powerful for image processing, and more generally
 1128 any data that can be expressed as a regular, discrete space and from which the information reside
 1129 in the dispersion in this space. For an image, the edges of an object and how they assemble. A red
 1130 square, straight edges with a sharp angle between them, is much less representative of a duck than
 1131 an yellow sphere, round edges without sharp angles.

- 1132 This "image" projection is not fitted for every problematics. The signals produced by a detector does
 1133 not always have the properties of images. In the case of JUNO for example, we can create an image
 1134 of two channels, one for the charge Q and one for the timing t but this image should be spheric.
 1135 Furthermore JUNO is by nature inhomogeneous, using two different systems : The LPMT and the

SPMT. Those two systems have different regime, and thus should be processed differently. We could imagine images with four channels, two for the LPMT and two for the SPMT, or even a branched CNN with one convolution branch for the LPMT and another one for the SPMT. Anyway, the CNN will need to combine the two systems.

To get around the restrictions of data representation imposed by CNNs, we can use the more flexible *graph* representation. A graph $G(\mathcal{N}, \mathcal{E})$ is composed of vertex or node $n \in \mathcal{N}$ and edges $e \in \mathcal{E}$. The

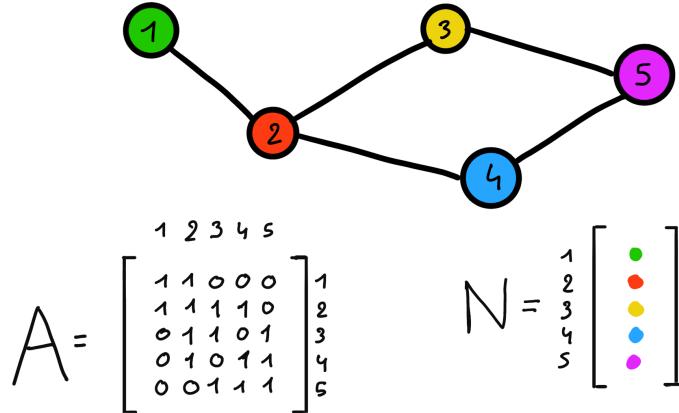


FIGURE 3.12 – Illustration of a graph and its tensor representation.

edges are associated to two nodes $(u, v) \in \mathcal{N}^2$, “connecting” them. The node and the edges can hold features, commonly represented as vector $n \in \mathbb{R}^{k_n}$, $e \in \mathbb{R}^{k_e}$ with k_n and k_e the number of features on the nodes and edges respectively. We can thus define a graph using two tensors A_e^{ij} the adjacency tensor that hold the features $e \in [0, k_e]$ of the edge connecting the node i and j and the tensor N_v^i that hold the features $v \in [0, k_n]$ of a node i .

More figuratively, using the example in figure 3.12, we have a graph of 5 nodes with a color as feature. The edges have no features, we thus encode their existences as 0 or 1. In a realistic examples as JUNO we could represent each PMTs as nodes and the edges between them as their relation such as distance, timing difference, etc... There no strict rules about what is a node or how they should be linked together. This abstraction allow us to represent virtually any type of detector of any geometry.

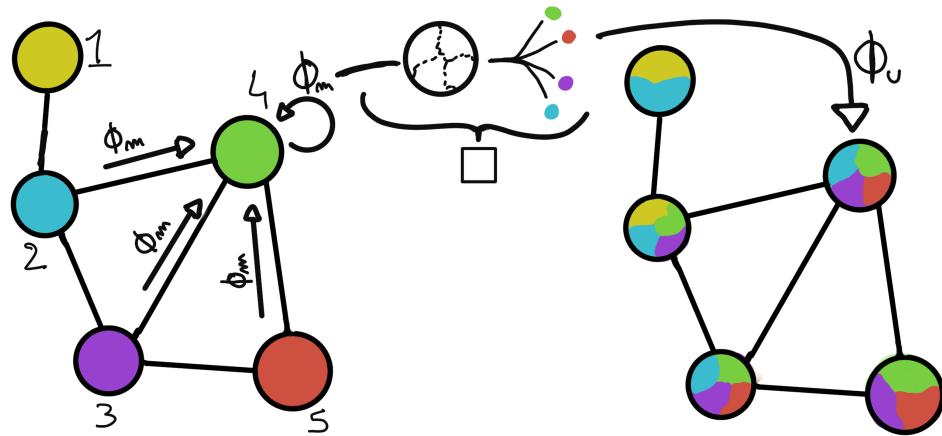


FIGURE 3.13 – Illustration of the message passing algorithm. The detailed explanation can be found in section 3.2.3

1152 To process such object we need specific machine learning algorithms we call Graph neural network.
 1153 To efficiently manipulate graph we need to structurally encode their property in the neural network
 1154 computing architecture: each node is equivalent (as opposite to ordered data in a vector), each node
 1155 has a set of neighbours, ... One of this method is the message passing algorithm presented historically
 1156 in "Neural Message Passing for Quantum Chemistry" [59]. In this algorithm, with each layer of
 1157 message passing a new set of features is computed for each node following

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

1158 where ϕ_u is a differentiable *update* function, \square_j is a differentiable *aggregation* function and ϕ_m is a
 1159 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
 1160 nodes n_j from which it exist an edge $e_{ij} \rightarrow (n_i, n_j)$. k is the layer on which the message passing
 1161 algorithm is applied. The update function need also a few other property if we want to keep the
 1162 graph property, most notably the permutational invariance of its parameters (example: mean, std,
 1163 sum, ...). The differents message, update and aggregation functions can really be any kind of function
 1164 if they follow the constraint presented before, even small Neural Network.

1165 The edges features can also be updated, either by directly taking the results of ϕ_m or by using another
 1166 message function ϕ_e .

1167 To explain this process, let's take the situation presented in figure 3.13. We start with an input graph
 1168 on left, in this case the message passing algorithm is mixing the color on each nodes and produce
 1169 nodes of mixed color. For simplicity, the ϕ_m and ϕ_u function are the identity, they take a color and
 1170 output the same color.

1171 Let's look at what's happening in the node 4. It has 3 neighbours and is a neighbour of itself. The four
 1172 resulting ϕ_m extract the color of each nodes and then feed them to the \square function. The \square function
 1173 just equally distribute the color in the node. Finally the ϕ_u function just update the node with the
 1174 output of \square .

1175 Interestingly we see that the new node 4 does not have any yellow, the color of node 1. But if we were
 1176 to run the message passing algorithm again, it would get some as node 2 is now partially yellow. If
 1177 color here represent information, we see that multiple step are needed so that each node is "aware"
 1178 of the informations the other nodes possess.

1179 Message passing is a very generic way of describing the process of GNN and it can be specialized
 1180 for convolutional filtering [47], diffusion [60] and many other specific operation. GNN are used in a
 1181 wide variety of application such as regression problematics, node classification, edge classification,
 1182 node and edge prediction, ...

1183 It is a very versatile but complex tool.

1184 3.2.4 Adversarial Neural Network (ANN)

1185 The adversarial machine learning, Adversarial Neural Networks (ANN) in the case of neural net-
 1186 work, is a family of unsupervised machine learning algorithms where the learning algorithm (gen-
 1187 erator) is competing against another algorithm (discriminator). Taking the example of Generative
 1188 Adversarial Networks, concept initially developed by Goodfellow et al. [61], the discriminator goal
 1189 is to discriminate between data coming from a reference dataset and data produced by the generator.
 1190 The generator goal, on the other hand, is to produce data that the discriminator would not be able to
 1191 differentiate from data from the reference dataset. The expression of duality between the two models
 1192 is represented in the loss where, at least a part of it, is driven by the results of the discriminator.

¹¹⁹³ **Chapter 4**

¹¹⁹⁴ **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 - $\sigma_x \sigma_p \geq \frac{\hbar}{2}$

¹¹⁹⁷ **Contents**

¹¹⁹⁸	4.1 Method and model	56
¹¹⁹⁹	4.1.1 Model	57
¹²⁰⁰	4.1.2 Data representation	58
¹²⁰¹	4.1.3 Dataset	60
¹²⁰²	4.1.4 Data characteristics	61
¹²⁰³		
¹²⁰⁴	4.2 Training	63
¹²⁰⁵	4.3 Results	63
¹²⁰⁶	4.3.1 J21 results	64
¹²⁰⁷	4.3.2 J21 Combination of classic and ML estimator	66
¹²⁰⁸	4.3.3 J23 results	68
¹²⁰⁹	4.4 Conclusion and prospect	70

¹²¹⁰ 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 ¹²¹¹ physics events inside of the same medium but they differ in their photo-coverage, respectively 75.2% ¹²¹² and 2.7%, their dynamic range (see section 2.3.2), a thousands versus a few dozen, and their front-end ¹²¹³ electronics (see section 2.3.2).

¹²¹⁴ The SPMT system is essential to the deployment of the Dual Calorimetry techniques, already mentioned in Section 2.7 and described in [25, 27, 62]. It is indeed less subject than the LPMTs to ¹²¹⁵ charge non linearity effects (QNL). This topic will be studied in more detail in Chapter 7, where the potential of one of the Dual Calorimetry techniques is explored. It consists on combined oscillation ¹²¹⁶ analyses based on two antineutrino energy spectra : one reconstructed with the LPMT system, the other one with the SPMT system. For that purpose, it is therefore necessary to have reconstruction ¹²¹⁷ tools available. Well maintained tools using the LPMT are available in the collaboration's official software. This is not the case concerning the SPMT system, where algorithms were developed more ¹²¹⁸ sporadically. This is one of the reasons why we developed the CNN described in this chapter.

1227 Our efforts on it were limited to the early months of this thesis: it was above all a way to learn about
 1228 ML and about JUNO's detector and software. We benchmarked its performance against a classical
 1229 algorithm developed in [63] but not yet implemented in JUNO's software.

1230 As discussed in Chapter 3, Machine Learning (ML) algorithms shine when modeling highly dimen-
 1231 sional data from a given dataset. In our case, we have access to complete monte-carlo simulation of
 1232 our detector to produce large datasets that could represent multiple years of data taking. Ideally ML
 1233 algorithms would be able to consider the entirety of the information in the detector and converge on
 1234 the best parameters to yield optimal results.

1235 The difference between this ideal and what can be achieved in reality is an important subject. In
 1236 particular, we wonder if an exhaustive usage of the information present in the detector could lead to
 1237 use informations that are mismodelled in our simulated training samples (or present only in these
 1238 samples) and therefore lead to biases when the algorithm is applied to real data. A simple way
 1239 to start addressing this reliability issue is to try to evaluate to which extent various reconstruction
 1240 methods use the same information. An attempt at this is presented at the end of this chapter. This is
 1241 also the subject of Chapter 6.

1242 4.1 Method and model

1243 One of simplest way to look at JUNO data is to consider the detector as an array of geometrically
 1244 distributed sensors on a sphere. Their repartition is almost homogeneous, on this sphere surface
 1245 providing an almost equal amount of information per unit surface. It is then tempting to represent
 1246 the detector as a spherical image with the PMTs in place of pixels. Two events with two different
 1247 energy or position would produce two different images.

1248 The most common approach in machine learning for image processing and image recognition is the
 1249 Convolutional Neural Network (CNN). It is widely used in research and industry [54, 64–66] due to
 1250 its strengths (see section 3.2.2) and has proven its relevance in image processing.

1251 Some CNN are developed to process spherical images [67] but for the sake of simplicity and as a
 1252 first approach we decided to go with a planar projection of the detector, approach that has proven its
 1253 efficiency using the LPMT system (see section 2.7.3). The details about this planar projection will be
 1254 discussed in section 4.1.2.

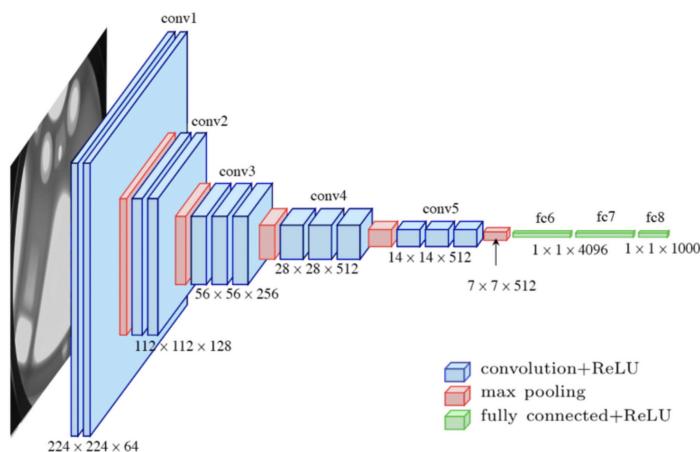


FIGURE 4.1 – Graphic representation of the VGG-16 architecture, presenting the different kind of layer composing the architecture.

4.1.1 Model

The architecture we use is derived from the VGG-16 architecture [54] 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.

- N_{blocks} : the number of convolution blocks, a block being composed of two convolutional layers with 3×3 filters using ReLU activation function, a 3×3 kernel max-pooling layer (except for the last block).
- $N_{channels}$: The number of channels in the first block. The number of channels in the subsequent blocks is computed using $N_{channels}^i = i * N_{channels}$, $i \in [1..N_{blocks}]$.
- **FCDNN configuration:** The result of the last convolution layer is flattened then fed to a FCDNN. Its configuration is expressed as the ouputs of sequenced fully connected linear layer using the PReLU activation function. For example $2 * 1024 + 2 * 512$ is the sequence of 2 layers which output is 1024 followed by 2 other layers with an output of 512. Finally the last layer is a linear layer outputting 4 features without activation function. Each feature of the last layer represent a component of the interaction vertex: Energy, X, Y, Z.
- **Loss:** The loss function. In this work we study two different loss function $(E + V)$ and $(E_r + V_r)$ detailed below.

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

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

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

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

The idea is that: due to the inherent statistic uncertainty over the number of collected Number of Photo Electrons (NPE), the absolute resolution $\sigma(E - E_{true})$ will be larger at higher energy than at low energy. But we expect the *relative* energy resolution $\frac{\sigma(E - E_{true})}{E_{true}}$ 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.

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

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

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

- Gen₀: $N_{blocks} = 4, N_{channels} = 64$, FCDNN configuration: $1024 * 2 + 512 * 2$, Loss $\equiv E + V$
- Gen₁: $N_{blocks} = 4, N_{channels} = 64$, FCDNN configuration: $1024 * 2 + 512 * 2$, Loss $\equiv E_r + V_r$
- Gen₂: $N_{blocks} = 5, N_{channels} = 64$, FCDNN configuration: $4096 * 2 + 1024 * 2$, Loss $\equiv E + V$
- Gen₃: $N_{blocks} = 5, N_{channels} = 64$, FCDNN configuration: $4096 * 2 + 1024 * 2$, Loss $\equiv E_r + V_r$

The resulting models possess between 2'041'034, for Gen₅₂ and Gen₅₃, and 5'759'839'242 parameters, for Gen₂₆ and Gen₂₇. The models of interest in this thesis, from which the results are discussed in section 4.3, possess 86'197'196 parameters for Gen₃₀ and 332'187'530 parameters for Gen₄₂. For comparison the model of CNN developed in JUNO before posses 38'352'403 parameters [40].

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

TABLE 4.1 – Sets of hyperparameters values considered in this study

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

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

The models were developped in Python using the Pytorch framework [56] using NVIDIA A100 [68] and NVIDIA V100 [69] gpus. The A100 was split in two, thus the accessible gpu memory was the same as V100, 20 Gb, making it impossible to train some of the architectures due to memory consumption.

The training was monitored in realtime by a custom tooling that was developed during this thesis, DataMo [70].

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

4.1.2 Data representation

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

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

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

1329 where H is the height of the image, W the width of the image and $(0, 0)$ the top left corner of the
 1330 image.

1331 This projection keep the SPMT position in the image proportional to their spherical coordinates while
 1332 keeping the neighbouring information. This proportionality allow us to keep the specificities of the
 1333 detector structure, the vertical bands visible in 4.2b.

1334 When two SPMTs in the same pixel are hit in the event time window, the charges are summed and
 1335 the lowest of the hit-time is chosen. The time window depends on the datasets and are detailed in
 1336 section 4.1.2. The SPMTs being located close to each other, we expect the time difference between
 1337 two successive physics signals, two photons being collected, to be small. The first hit time is chosen
 1338 because it can be considered as the relative propagation time of the photons that went the "straight-
 1339 est", i.e. that went under the less perturbation of the two. The timing is thus more representative of
 1340 the event location.

1341 The only potential problem in using this first time come from the Dark Noise (DN). Its time distribu-
 1342 tion is uniform over the signal and could come before a physics signal on the other SPMT in the pixel.
 1343 In that case, the time information in the pixel become irrelevant and we lose the timing information
 1344 for this part of the detector. As illustrated in figure 4.2b the image dimension have been optimized
 1345 so that at most two SPMTs are in the same pixel while keeping the number of empty pixels relatively
 1346 low to prevent this kind of issue.

1347 While it could be possible to use larger images (more pixel) to prevent overlapping, keeping image
 1348 small images gives multiple advantages:

- 1349 — As presented in section 4.1.1, the convolution filter we use are 3×3 convolution filter, meaning
 1350 that if SPMTs would be separated by more than one pixel, the first filter would only see one
 1351 SPMT per filter. This behavior would be kind of counterproductive as the first convolution
 1352 block would basically be a transmission layer and would just induce noise in the data.
- 1353 — It keep the network relatively small, while this do not impact the convolution layers, the
 1354 flatten operation just before the FCDNN make the number parameters in the first layer of
 1355 it dependent on the size of the image.
- 1356 — It reduce the number of empty pixel in the image.

1357
 1358 The question of empty pixel is an important question in this data representation. There is two kind
 1359 of empty pixels in the data.

1360 The first kind is pixel that contain a SPMT but the SPMT did not get hit nor registered any dark noise
 1361 during the event. In this case, the charge channel is zero, which have a physical meaning but then
 1362 come the question of the time layer. One could argue that the correct time would be infinity (or the
 1363 largest number our memory allows us) because the hit "never" happened, so extremely far from the
 1364 time of the event. This cause numerical problem as large number, in the linear operation that are
 1365 happening in the convolution layers, are more significant than smaller value. We could try to encode
 1366 this feature in another way but no number have any significance due to our time being relative to
 1367 the trigger of the experiment so -1 for example is out of question. Float and Double gives us access
 1368 to special value such as NaN (Not a Number) [71] but the behavior is to propagate the NaN which
 1369 leaves us with NaN for energy and position. We choose to keep the value 0 because it's the absorbing
 1370 element of multiplication, absorbing the "information" of the parameter it would be multiplied by.
 1371 It also can be thought as no activation in the ReLU activation function. It's important to keep in mind

1372 the fact that a part of the detector that has not been hit is also an information: There is no signal in
 1373 this part of the detector. This problematic will be explored in more details in Chapter 5.

1374 The second kind of pixels are the one that do not represent parts of the detector such as the corners
 1375 of the image. The question is basically the same, what to put in the charge and the time channel. The
 1376 decision is to set the charge and time to 0 following the above reasoning.

1377 Another problematic that happens with this representation, and this is not dependent of the chosen
 1378 projection, is the deformation in the edges of the image and the loss of the neighbouring information
 1379 in the for the SPMTs at the edge of the image $\phi \sim 180^\circ$. This deformation and neighbouring loss
 1380 could be partially circumvented as explained in section 4.4

1381 4.1.3 Dataset

1382 In this study we will discuss two datasets of one millions prompt signal of IBD events.

1383 J21

1384 The first one comes from the JUNO official MC simulation J21v1r0-Pre2 (released the 18th August
 1385 2021). This historical version is the one on which the classical SPMT reconstruction algorithm was
 1386 developed. This classical methods is based on the time likelihood presented section 2.7 for the vertex
 1387 reconstruction, and compute the energy by correcting the detector effect on the ration N_{pe}/E_{dep} . It is
 1388 detailed in Chapter 4 of [63]. This dataset is used as a reference for comparison to classical algorithm
 1389 performances. The data in this dataset is *detsim* level (see section 2.6) which includes no digitization,
 1390 no DAQ and therefore no reconstruction of PMT signals. Only the number of PEs that hit a PMT and
 1391 the hit times are provided. A fast simulation based on gaussian drawings produces charges, with
 1392 bias and variability, and the equivalent for times. The drawings parameters were adjusted based on
 1393 [24, 72]. Because there is no charge reconstruction, the timing on the event is based on the Geant4
 1394 simulation, and so $t = 0$ is the moment the positron is created in the CD. To prevent correlation
 1395 between the numerical value of the time of the first hit t_0 and the radius of the event, we offset all
 1396 time by this first hit time. Without simulation of the charge reconstruction, we cannot simulate the
 1397 event trigger, we thus add an arbitrary time cut at a $t_0 + 1000$ ns.

1398 J23

1399 The second comes from the JUNO official monte-carlo simulations J23.0.1-rc8.dc1 (released the 7th
 1400 January 2024). The data is *calib* level (see section 2.6). Here the charge comes from the waveform
 1401 integration, the time window resolution and trigger decision are all simulated inside the software.

1402 To put in perspective this amount of data, the expected IBD rate in JUNO is 47 / days. Taking into
 1403 account the calibration time, and the source reactor shutdown, it amount to $\sim 94'000$ IBD events
 1404 in 6 years. With this million of event, we are training the equivalent of ~ 10 years of data. With
 1405 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
 1406 event scenarios that could be happening in the detector.

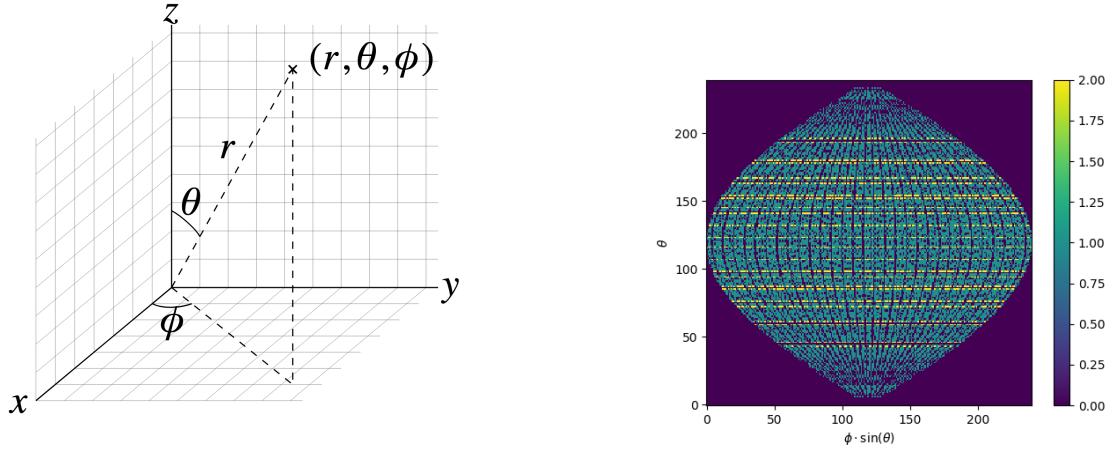
1407 While we expect and hope the MC simulation to give use a realistic representation of the detector,
 1408 there could be effect, even after the fine-tuning on calibration data, that the simulation cannot handle.
 1409 Thus, once the calibration will be available, we will need to evaluate, and if needed retrain, the
 1410 network on calibration data to establish definitive performances.

1411 The simulated data is composed of positron events, uniformly distributed in the CD volume and in
 1412 kinetic energy over $E_k \in [0; 9]$ MeV producing a deposited energy $E_{dep} \in [1.022; 10.022]$ MeV. This is
 1413 done to mimic the signal produced by the IBD prompt signal. Uniform distributions are used so that

¹⁴¹⁴ the CNN does not learn a potential energy distribution, favoring some part of the energy spectrum
¹⁴¹⁵ instead of other.

¹⁴¹⁶ 4.1.4 Data characteristics

¹⁴¹⁷ To delve a bit into the kind of data we will use, you can find in figure 4.2b the repartition of the
¹⁴¹⁸ SPMTs in the image. The color represent the number of SPMTs per pixel.



(A) Spherical coordinate system used in JUNO for reconstruction

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

FIGURE 4.2

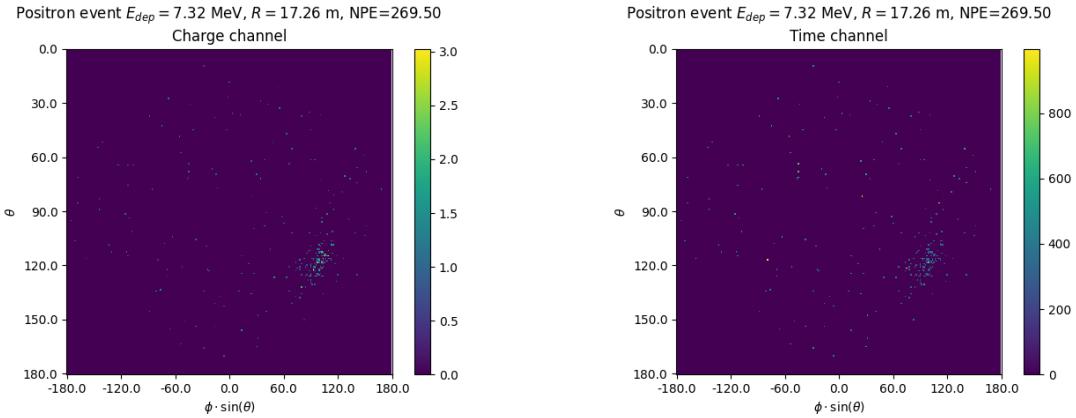


FIGURE 4.3 – Example of a high energy, radial event. We see a concentration of the charge on the bottom right of the image, clear indication of a high radius event. **On the left:** the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right:** The time channel in nanoseconds.

¹⁴¹⁹ See also figures 4.3 to 4.6 - and the explanation in their captions - which present events from J23 for
¹⁴²⁰ different positions and energies. We see some characteristics and we can instinctively understand
¹⁴²¹ how the CNN could discriminate different situations.

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

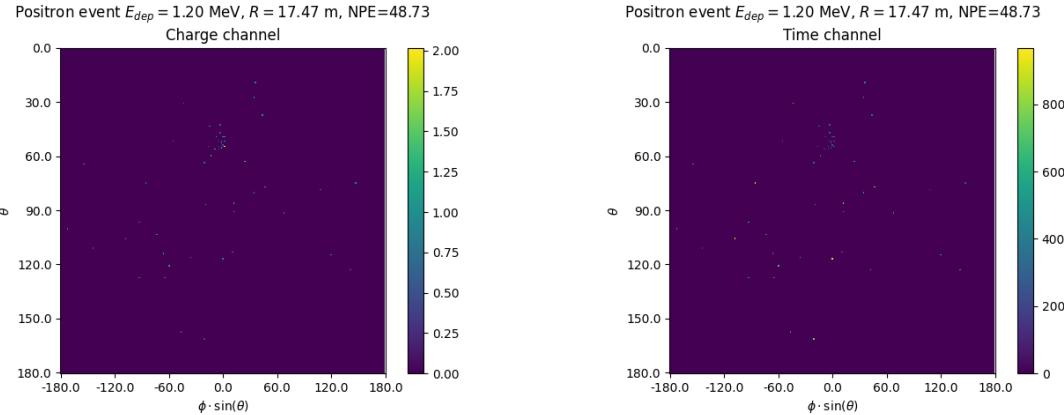


FIGURE 4.4 – Example of a low energy, radial event. The signal here is way less explicit, we can kind of guess that the event is located in the top middle of the image. **On the left:** the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right:** The time channel in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

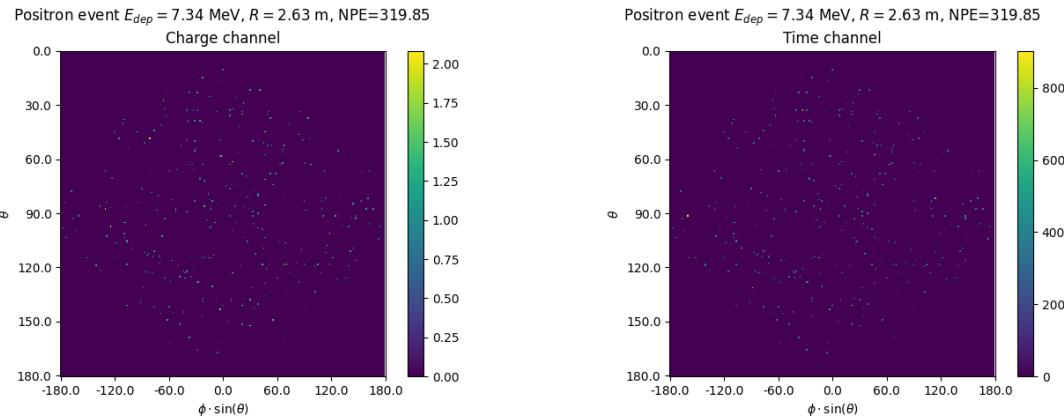


FIGURE 4.5 – Example of a high energy, central event. In this image we can see a lot of signal but uniformly spread, this is indicative of a central event. **On the left:** the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right:** The time channel in nanoseconds.

LS we can model:

$$NPE_{tot} = E_{dep} \cdot P_{mev} + D_N \quad (4.5)$$

$$\frac{NPE_{tot}}{E_{dep}} = P_{mev} + \frac{D_N}{E_{dep}} \quad (4.6)$$

¹⁴²² where NPE_{tot} is the total number of PE detected by the event, P_{mev} is the mean number of PE detected
¹⁴²³ per MeV and D_N is the dark noise contribution that is considered energy independent. In the case
¹⁴²⁴ where the readout time window is dependent of the energy the dark noise contribution become
¹⁴²⁵ energy dependant, also the LS response is realistically energy dependant but figure 4.7a shows that
¹⁴²⁶ we are heavily dominated by the stochastic behavior of light emission and detection.

¹⁴²⁷ The fit shows a light yield of 40.78 PE/MeV and a dark noise contribution of 4.29 NPE. As shown in
¹⁴²⁸ figure 4.7b, the physics makes for 90% of the signal at low energy.

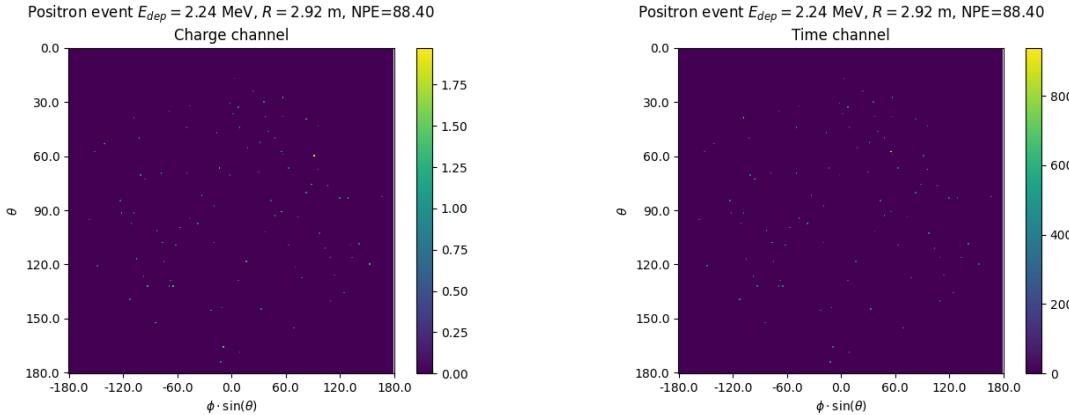


FIGURE 4.6 – Example of a low energy, central event. Here there is no clear signal, the uniformity of the distribution should make it central. **On the left:** the charge channel. The color is the charge in each pixel in NPE equivalent. **On the right:** The time channel in nanoseconds.

4.2 Training

The optimizer used for the training is the Adam [51] 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 reduced exponentially during the training at a rate of $\gamma = 0.95$, thus $\lambda_{i+1} = 0.95\lambda_i$ where i is the epoch.

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

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

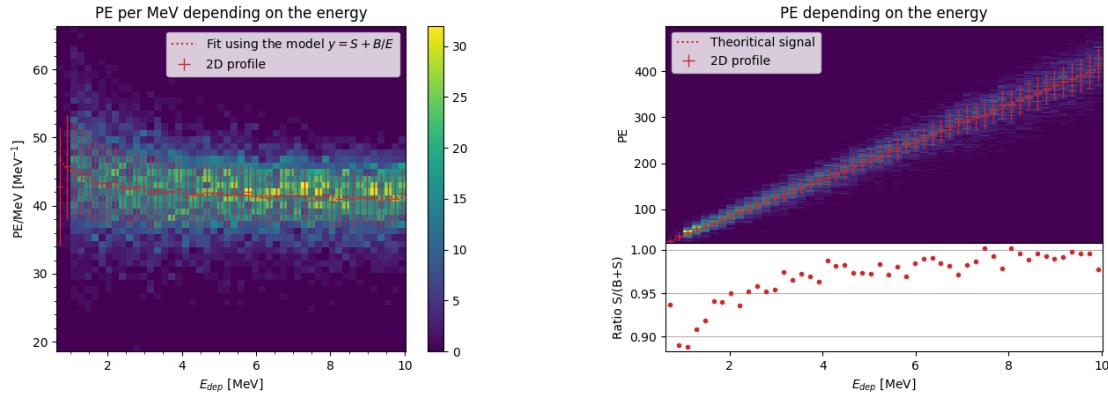
4.3 Results

Before presenting the results, let's discuss the different observables.

The events are considered point-like in this study. The target truth position, or vertex, is the mean position of the energy deposits of the positron and the two annihilation gammas. This approximation for point-like interaction is also used for the likelihood study presented in section 2.7 and in previous ML studies presented in section 2.7.3 [40].

Due to the symmetries of the detector, we mainly consider and discuss the bias and precision evolution depending on the radius R but we will still monitor the performances depending on the spherical angle θ and ϕ . From the detector construction and effect we expect dependency in radius due to the TR area effect presented in section 2.7 and the possibility for the positron or the gammas to escape from the CD for positrons interacting near the edge. We also expect dependency on θ , the top of the experiment being non-instrumented due to the filling chimney. It is also to be noted that the events in the dataset are uniformly distributed in the CD, and so are uniformly distributed in R^3 and ϕ . The θ distribution is not uniform and we will have more events for $\theta \sim 90^\circ$ than $\theta \sim 0^\circ$ or $\theta \sim 180^\circ$.

We define multiple energy in JUNO:



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

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

FIGURE 4.7

- E_ν : The energy of the neutrino.
- E_k : The kinetic energy of the resulting positron from the IBD.
- E_{dep} : The deposited energy of the positron and the two annihilation gammas.
- E_{vis} : The equivalent visible energy, so E_{dep} after the detector effect such as the LS response non-linearity.
- E_{rec} : The reconstructed energy by the reconstruction algorithm. The expected value depend on the algorithm we discuss about. For example the algorithm presented in section 2.7 reconstruct E_{vis} while the ones presented in section 2.7.3 reconstruct E_{dep} .

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

4.3.1 J21 results

The best results comes from the Gen₃₀ model, meaning then 30th model generated using the table 4.1: Gen₃₀: $N_{blocks} = 3$, $N_{channels} = 32$, FCDNN configuration: $2048 * 2 + 1024 * 2$, Loss $\equiv E + V$.

The performances of its reconstruction are presented in blue in figure 4.8. Superimposed in black is the performances of the classical algorithm from [63].

Energy reconstruction

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

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

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

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

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

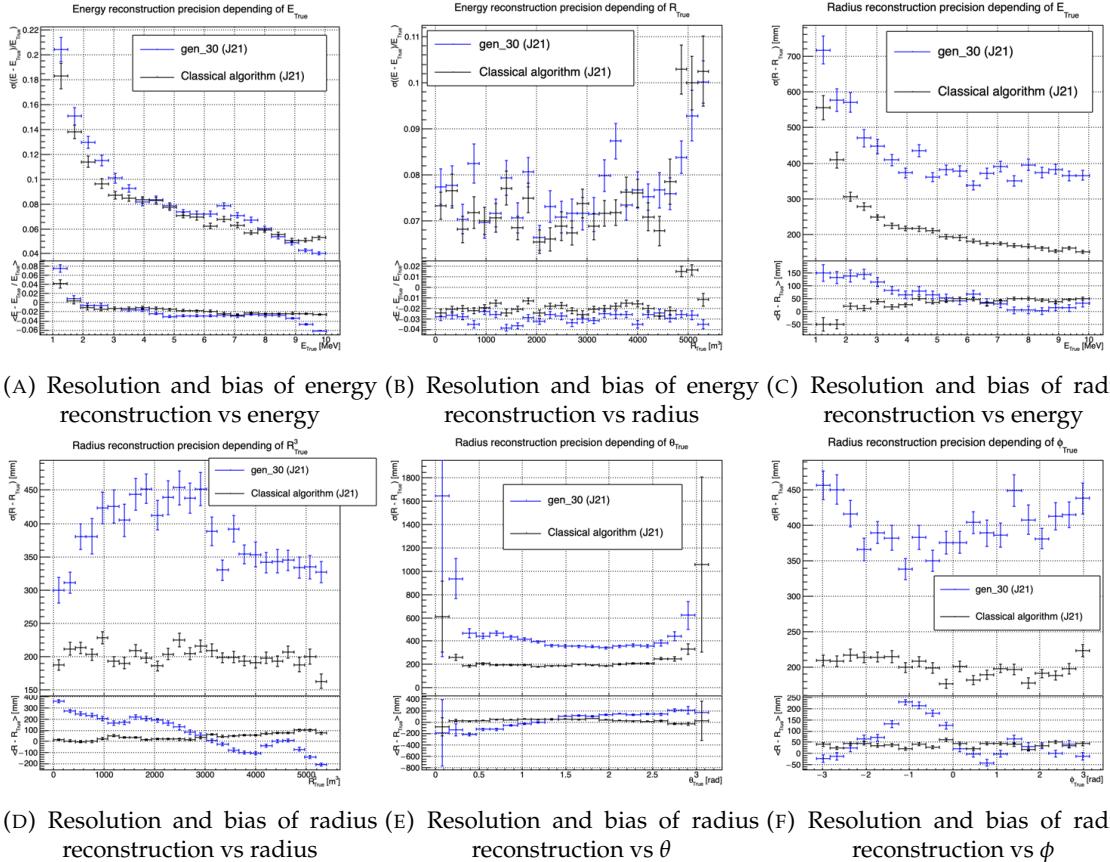


FIGURE 4.8 – Reconstruction performance of the Gen₃₀ model on J21 data and its comparison to the performances of the classic algorithm “Classical algorithm” from [63]. The top part of each plot is the resolution and the bottom part is the bias.

This behavior also explain the heavy bias at low energy in figure 4.8a. The energy bias of the CNN is 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.

4.91 Vertex reconstruction

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

The comparison of radius reconstruction between the classical algorithm and Gen₃₀ are presented in the figures 4.8c, 4.8d, 4.8e and 4.8f. The resolution obtained by the CNN is twice worse in average,

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

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

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

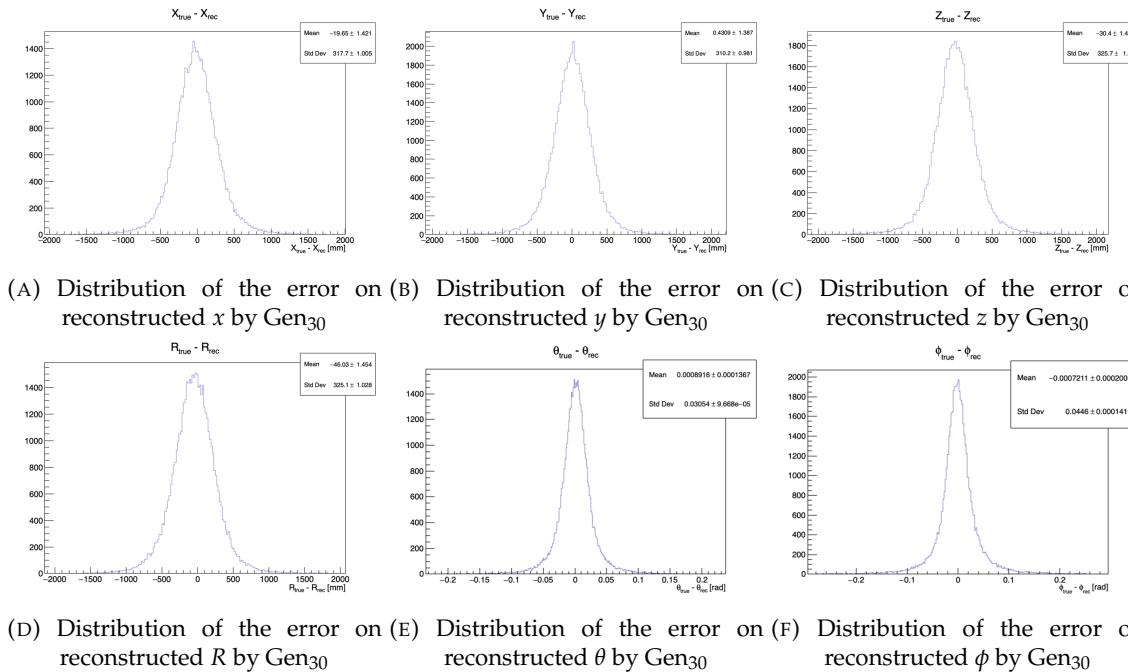
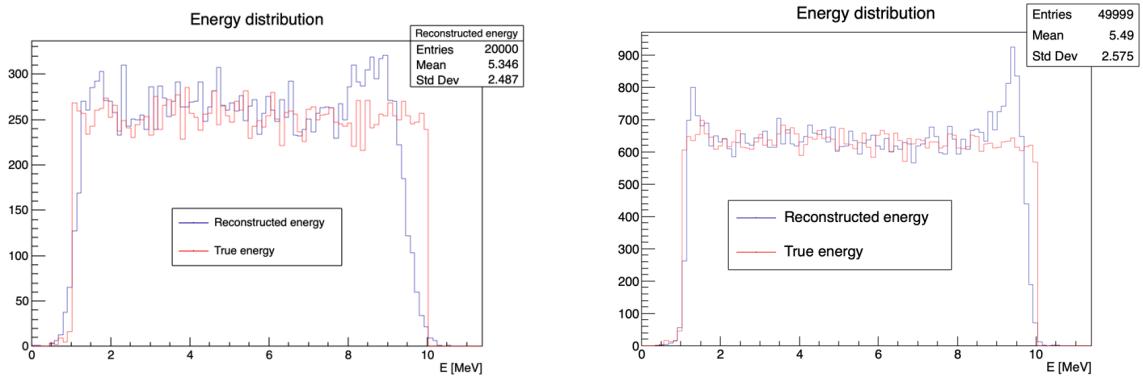


FIGURE 4.9 – Residual distribution of the different component of the vertex by Gen₃₀. The reconstructed component are x , y and z but we see similar behavior in the error of R , θ and ϕ .

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

4.3.2 J21 Combination of classic and ML estimator

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



(A) Distribution of Gen₃₀ reconstructed energy and true energy of the analysis dataset (J21)

(B) Distribution of Gen₄₂ reconstructed energy and true energy of the analysis dataset (J23)

FIGURE 4.10

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

$$E[\theta_N] = \mu_N + X; \text{Var}[\theta_N] = \sigma_N^2 \quad (4.7)$$

$$E[\theta_C] = \mu_C + X; \text{Var}[\theta_C] = \sigma_C^2 \quad (4.8)$$

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

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

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

then the variance and mean would follow

$$E[\hat{\theta}] = \frac{1}{2}E[\theta_N] + \frac{1}{2}E[\theta_C] \quad (4.10)$$

$$= \frac{1}{2}(\mu_N + X + \mu_C + X) \quad (4.11)$$

$$= \frac{1}{2}(\mu_N + \mu_C) + X \quad (4.12)$$

$$\text{Var}[\hat{\theta}] = \frac{1}{4}\sigma_N^2 + \frac{1}{4}\sigma_C^2 + 2 \cdot \frac{1}{4} \cdot \sigma_{NC} \quad (4.13)$$

$$= \frac{1}{4}\sigma_N^2 + \frac{1}{4}\sigma_C^2 + \frac{1}{2} \cdot \sigma_{NC} \quad (4.14)$$

$$= \frac{1}{4}\sigma_N^2 + \frac{1}{4}\sigma_C^2 + \frac{1}{2} \cdot \sigma_N \sigma_C \rho_{NC} \quad (4.15)$$

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

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

By generalising the equation 4.9 to

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

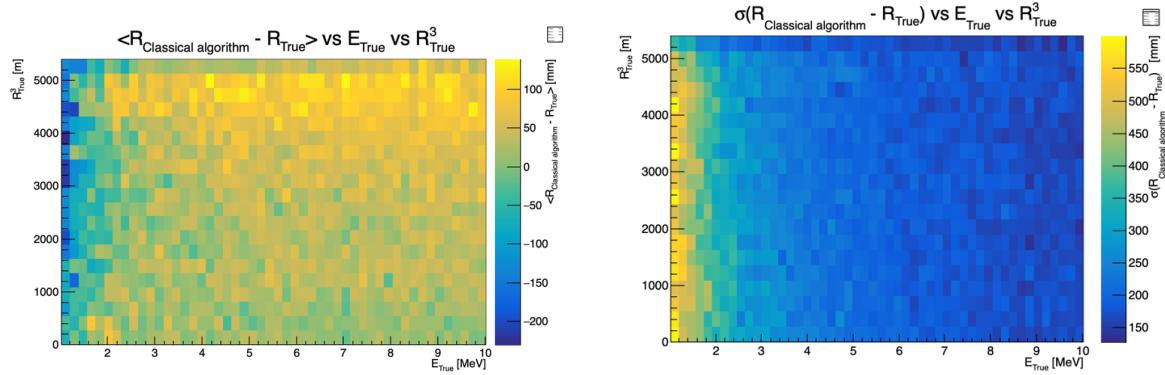


FIGURE 4.11 – Radius bias (on the left) and resolution (on the right) of the classical algorithm in a E, R^3 grid

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

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

and the estimator without bias

$$\alpha = \frac{\mu_C}{\mu_C - \mu_N} \quad (4.18)$$

See annex A for demonstration.

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

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

The map we are using are composed of 20 bins for R^3 going from 0 to 5400 m³ (17.54 m) and 50 bins in energy ranging from 1.022 to 10.022 MeV. In the case where we are outside the grid, we use the closest cell.

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

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

4.3.3 J23 results

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

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

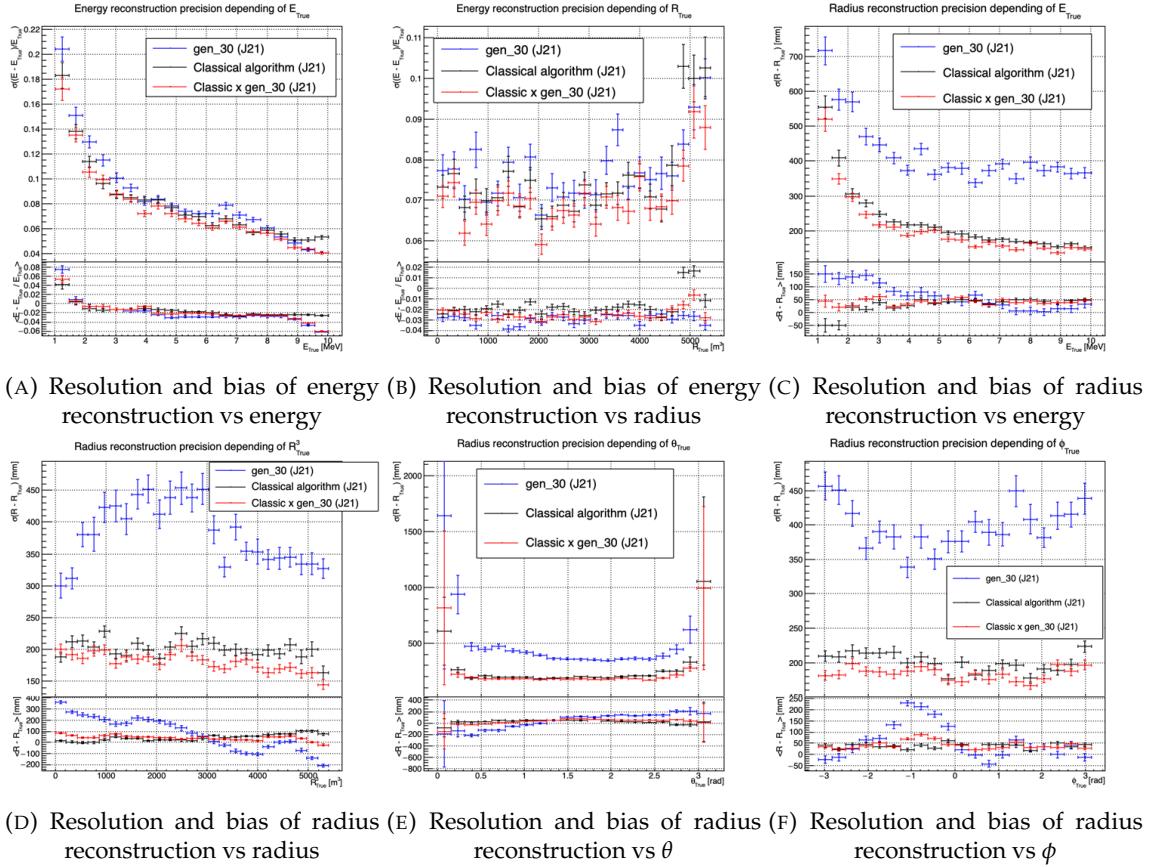


FIGURE 4.12 – Reconstruction performance of the Gen30 model on J21, the classic algorithm “Classical algorithm” from [63] and the combination of both using weighted mean. The top part of each plot is the resolution and the bottom part is the bias.

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

We retrained the models defined in 4.1.1 on the J23 data and used the same hyperparameter optimisation procedure. The results from the best architecture, Gen₄₂, are presented in figure 4.14. Following the table 4.1, Gen₄₂: $N_{blocks} = 3$, $N_{channels} = 64$, FCDNN configuration: $4096 * 2$, Loss $\equiv E + V$.

1555 Energy reconstruction

1556 The results of the energy reconstruction are presented in figures 4.14a and 4.14b. The resolution is
1557 close to the one of the classical algorithm with the exception of the start and end of the spectrum.
1558 This is the same effect that we saw with Gen₃₀, events are pulled from the edge of the distribution,
1559 resulting in smaller resolution but heavy biases.

1560 Vertex reconstruction

1561 The vertex reconstruction, presented in figures 4.14c, 4.14d, 4.14e and 4.14f is not yet to the level of
1562 the classical reconstruction but the degradation is smaller than for Gen₃₀ being at most a difference

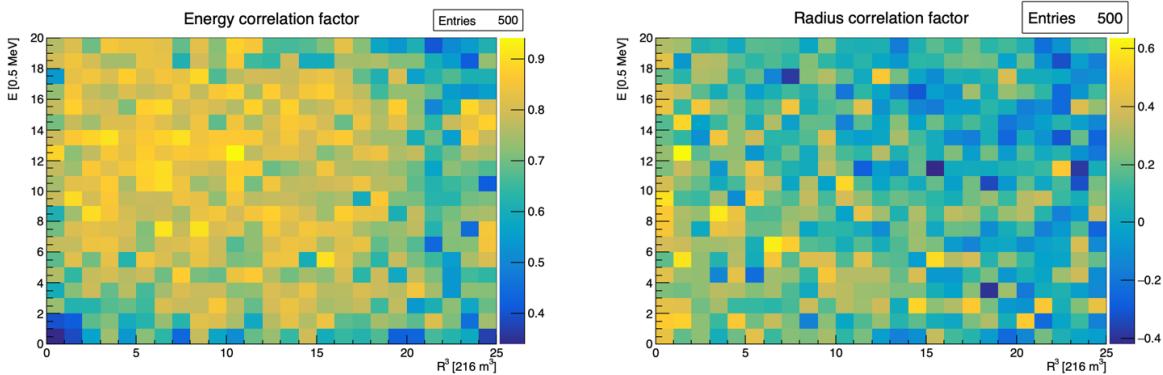


FIGURE 4.13 – Correlation between CNN and classical method reconstruction (on the left) for energy and (on the right) for radius in a E, R^3 grid

of 15cm of resolution and closing to the performance of the classical algorithm in the most favourable condition. Gen₄₂ has also very little bias in comparison with the classical method with the exception of the transition to the TR area and at the very edge of the detector.

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

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

4.4 Conclusion and prospect

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

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

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

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

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

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

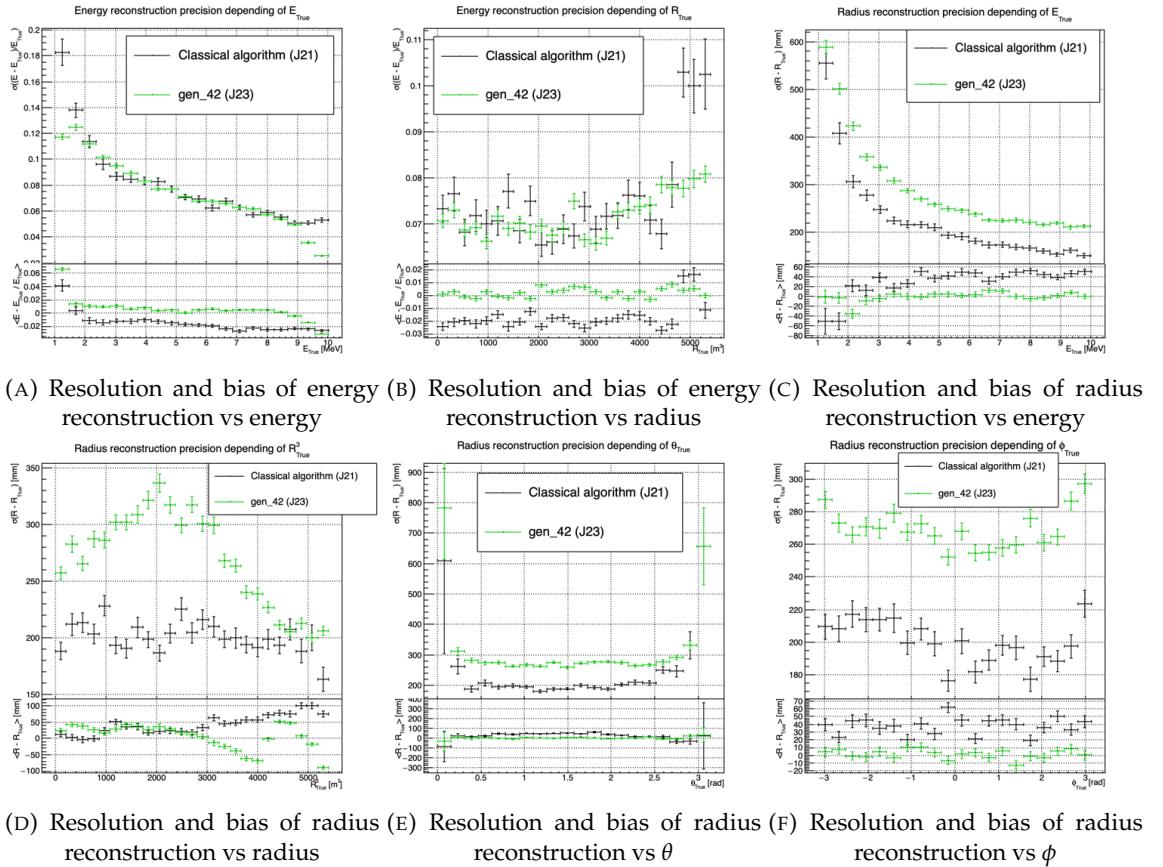


FIGURE 4.14 – Reconstruction performance of the Gen42 model on J23 data and its comparison to the performances of the classic algorithm “Classical algorithm” from [63]. The top part of each plot is the resolution and the bottom part is the bias.

¹⁵⁹² **Chapter 5**

¹⁵⁹³ **Graph representation of JUNO for
IBD reconstruction**

¹⁵⁹⁵

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

Douglas Adams, The Hitchhiker's Guide to the Galaxy

¹⁵⁹⁶

Contents

¹⁵⁹⁷	5.1 Data representation	74
¹⁵⁹⁹	5.2 Message passing algorithm	76
¹⁶⁰⁰	5.3 Data	78
¹⁶⁰¹	5.4 Model	80
¹⁶⁰²	5.5 Training	80
¹⁶⁰³	5.6 Optimization	82
¹⁶⁰⁴	5.6.1 Software optimization	82
¹⁶⁰⁵	5.6.2 Hyperparameters optimization	83
¹⁶⁰⁶	5.7 performance of the final version	83
¹⁶⁰⁷	5.8 Conclusion	87

¹⁶¹¹ In section 2.7.3, we showed that all ML methods developed before this thesis to reconstruct IBDs have
¹⁶¹² similar results, and that their performance is very similar to that of the classical, likelihood-based
¹⁶¹³ algorithm. We think these similarities can reasonably be explained by this: the input data used by
¹⁶¹⁴ all these methods to compute E or \vec{X} is the same full list of PMT integrated signals $\{(Q_i, t_i); i \in$
¹⁶¹⁵ $1, \dots, N_{PMTs}\}$, and by the high level of sophistication of the detector's description in the likelihood.
¹⁶¹⁶ It's probable that the likelihood method looses very little information.

¹⁶¹⁷ May be some was, but that the ML algorithms were not designed well enough to recover it. It's also
¹⁶¹⁸ reasonable to think that ML algorithms will make a difference when, instead of the list of (Q_i, t_i) , a
¹⁶¹⁹ rawer information will be used in input, like the full waveform. To actually be able to learn from such
¹⁶²⁰ a complex and high dimensional input, well designed architectures (that would guide the learning
¹⁶²¹ toward the solution) are necessary. In any case, it seemed welcome to us to propose an additional
¹⁶²² algorithm, with an original architecture.

¹⁶²³ For the fist stage of its development, the purpose of this part of my thesis, we considered it was
¹⁶²⁴ enough to also take the (Q_i, t_i) list as the input. In case better of equivalent performance would be
¹⁶²⁵ achieved, we could hope the architecture would make a difference when more complex inputs would
¹⁶²⁶ be used. If not, we can conclude it's probably not relevant.

¹⁶²⁷ The algorithm we propose is a GNN. It also has the advantage of addressing sphericity issues
¹⁶²⁸ described in Chapter 4. From this graph representation, we can construct a neural network that will
¹⁶²⁹ process the data while keeping some interesting properties. For example the rotational invariance,

1630 i.e. the energy and radius of the event do change by rotation our referential. For more details see
 1631 section 3.2.3. Graph representation also has the advantage to be able to encode global and higher
 1632 order informations.

1633 5.1 Data representation

1634 In section 2.7.3, we mentioned a GNN developed before the beginning of this thesis to reconstruct
 1635 IBD energies in JUNO [40]. In their approach: nodes of the graph correspond to 3072 pixels representing
 1636 geometric regions of the detector and the information of the ~ 6 LPMTs found in a pixel are then
 1637 aggregated on those nodes. The network then process the data using the equivalent of convolution
 1638 but on graph [47]. In the first layer, each node is connected only with its direct neighbours.

1639 To determine the energy released by an IBD in the LS, it is helpful to determine the position of
 1640 the main energy deposit. Therefore, relative Q and t's of PMTs all around the sphere is a useful
 1641 information. If in the first layer only neighbour nodes are linked, several layers are necessary to
 1642 access this detector-wide information. In an ideal world, we would develop a Graph NN where each
 1643 PMT is a node (even if it has not been hit in the event under consideration, since this is in itself an
 1644 information) and where each node is connected to all the other ones. This makes the detector-wide
 1645 information available as early as the first layer. This architecture might help the network to better
 1646 learn. Such an architecture can also be motivated this way: one of the strength of GNN's is their
 1647 capacity to encompass the characteristics of a detector. A node can be the representation of a detector
 1648 element, and the edge can represent its relationship with other elements. In the case of JUNO, any
 1649 measurement is collective : an interaction is seen by all the PMTs, with no a priori hierarchy in the
 1650 role of each. A fully connected GNN, in that respect, seems to make sense.

1651 Another advantage of a GNN is also that it is well adapted to inhomogenous detectors. We therefore
 1652 tried to build GNNs including both LPMTs and SPMTs.

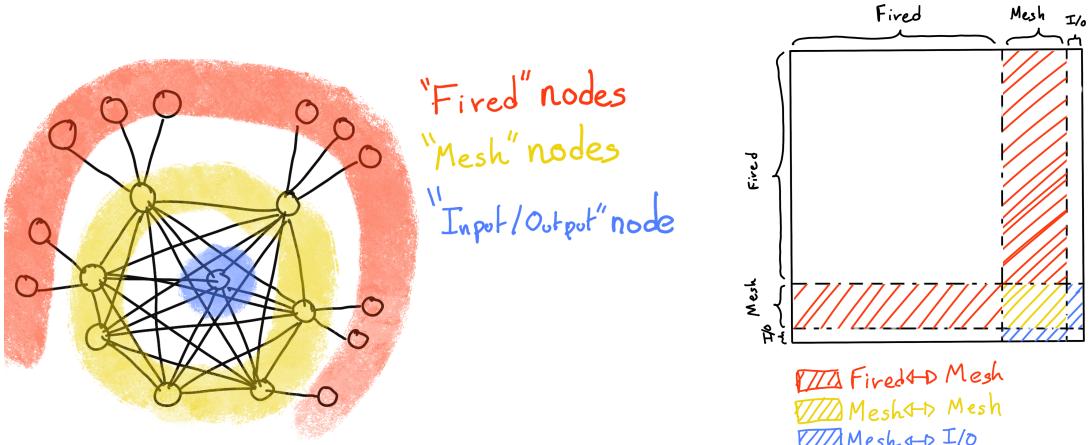
1653 With 17612 LPMTs and 25600 SPMTs, the ideal fully connected Graph mentioned above is impossible:
 1654 even excluding self relation and considering the relation to be undirected (the edge from a node A
 1655 to a node B being the same from as the one from B to A) the amount of necessary edges would be
 1656 $n(n - 1)/2$ with $n = 43212$ nodes. This amounts to 933'616'866 edges. If we encode an information
 1657 with double precision (64 bits) in what we call an adjacency matrix, illustrated in figure 3.12, each
 1658 information we want to encode in the relation would consume 4 GB of data. When adding the
 1659 overhead due to gradient computation during training, this would put us over the memory capacity
 1660 of a single V100 gpu card (20 GB of memory). We could use parallel training to distribute the training
 1661 over multiple GPU but we considered that the technical challenge to deploy this solution was too
 1662 high.

1663 We finally decided of a middle ground where we define three *families* of nodes:

- 1664 — The core of the graph is composed of nodes representing geometric regions of the detector.
 1665 We call those nodes **mesh** nodes. Those mesh nodes are all connected to each other. We keep
 1666 their number low to gain in memory consumption.
- 1667 — PMTs in which Photo-Electrons (PE) are found are represented by **fired** nodes. Fired nodes
 1668 are connected to the mesh node they geometrically belong to.
- 1669 — A final node is called the input/output node (**I/O**). It is connected to every mesh node. Its
 1670 features are combinations of signals found in the whole detector.

1671
 1672 Those nodes and their relations are illustrated in figure 5.1a. From this representation, we end up
 1673 with three distinct adjacency matrix

- 1674 — A $N_{\text{fired}} \times N_{\text{mesh}}$ adjacency matrix, representing the relations between fired and mesh. Those
 1675 relations are undirected.



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

(B) Illustration of what a dense adjacency matrix would look 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

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

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

1. P_l^h : Is the normalized power of the l th spherical harmonic. For more details about spherical

1691 harmonics in JUNO, see annex [B](#).

2. \mathbb{A} and \mathbb{B} are informations that are related the likeliness of the interaction vertex to be on the segment between the center of two meshes.

$$\mathbb{A}_{ij} = (\vec{j} - \vec{i}) \cdot \frac{\vec{l}_1}{D_{ij}} + \vec{i} \quad (5.1)$$

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

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

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

1692 where \vec{i} is the position vector of the mesh i , D_{ij} is the distance between the center of the meshes 1693 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 1694 i and n the optical index of the LS. \mathbb{A} is the vertex between center of meshes distance ratio 1695 between i and j based on the time information. For \mathbb{B} , the charge ratio evolve with the square 1696 of the distance, so the mesh couple with the smallest \mathbb{B} should be the one with the interaction 1697 vertex between its two center.

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

TABLE 5.1 – Features on the nodes of the graph. All charge are in [nPE], time in [ns] and position in [m].

Q and t are the reconstructed charge and time of the hit PMTs. (x, y, z) is the position of the PMTs and the last parameter represent the type of the PMT. It's 1 for LPMT and -1 for SPMT

Q_m and t_m is the set of charges and time of the PMT belonging the mesh m . (X_m, Y_m, Z_m) i the position of the center of the geometric region represented by the mesh m

$(\langle X \rangle, \langle Y \rangle, \langle Z \rangle)$ is the position of the charge barycenter, $\sum Q$ the sum of the collected charge in the detector and P_l^h is the relative power of the l th harmonic. See annex [B](#) for details.

1698 Since our different nodes do not have the same number of features, they exist in distinct spaces.
1699 Traditional graph neural networks only handle homogeneous graphs, where the nodes and edges
1700 have the same number of features at each layer. Therefore, the libraries and publicly available
1701 algorithms we found were not suited to our needs. As a result, we had to develop and implement a
1702 custom message-passing algorithm capable of handling our heterogeneous graph.

1703 5.2 Message passing algorithm

1704 As introduced in previous section and in the tables [5.1](#) and [5.2](#), our graphs nodes and edges will
1705 have different number of features depending on their nature, meaning that we cannot have a single

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

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

1706 message passing function. We thus need to define a message passing function for each transition
1707 inside or outside a family. Using the notation presented in section 3.2.3

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

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

$$\begin{aligned} & \phi_{u;F \rightarrow M}; \phi_{m;F \rightarrow M} \\ & \phi_{u;M \rightarrow F}; \phi_{m;M \rightarrow F} \\ & \phi_{u;M \rightarrow M}; \phi_{m;M \rightarrow M} \\ & \phi_{u;M \rightarrow IO}; \phi_{m;M \rightarrow IO} \\ & \phi_{u;IO \rightarrow M}; \phi_{m;IO \rightarrow M} \end{aligned}$$

1708 to update the nodes after each layers. Following the illustration in figure 5.3, for each transition
1709 between families or inside a family we need an aggregation, a message and an update function. For
1710 the aggregation, we use the sum. We use the same, simple, formalism for every ϕ_u :

$$\phi_u \equiv I_{i'}^{n'} = I_i^n A_{i',e}^i W_n^{e,n'} + I_i^n S_n^{n'} + B^{n'} \quad (5.6)$$

1711 using the Einstein summation notation. The second order tensor, or matrix, I_i^n is holding the nodes
1712 informations with i the node index and n the feature index. n represent the features of the previous
1713 layer and n' the features of this layer.

1714 $A_{i',e}^i$ is the adjacency tensor, discussed in the previous section, representing the edges between the
1715 node i' and the node i , each edges holding the features indexed by e . If the edge does not exist, the
1716 features are set to 0. This choice is justified by the linearity of the operation in equation 5.6 : whatever
1717 the weights, when multiplied by 0 the results is 0 and the sum result is unchanged.

1718 The learnable parameters are composed of:

- 1719 — The third order tensor $W_n^{e,n'}$ which represent the passage from the previous combined feature
1720 space between the node and the edge features $n \otimes e$, the previous layer, to the current space
1721 n' , this layer.
- 1722 — The first order tensor $B^{n'}$ which is a learnable bias on the new features n' .
- 1723 — The second order tensor $S_n^{n'}$, which can be viewed as a self loop relation where the node update
1724 itself based on the previous layer informations, going from the previous space n to the current
1725 space n' .

1726 If a node have neighbours in different families, the different IAW coming from the different families

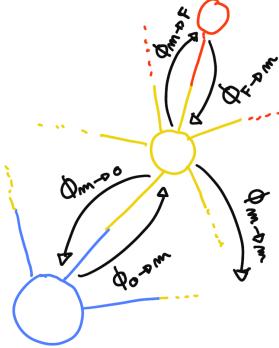


FIGURE 5.3 – Illustration of the different update function needed by our GNN

1727 are summed.

$$I' = \sum_{\mathcal{N}} [I_{\mathcal{N}} AW] + IS + B \quad (5.7)$$

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

$$I'_F = I_M A_{M \rightarrow F} W_{M \rightarrow F} + I_F S_F + B_F \quad (5.8)$$

$$I'_M = I_F A_{F \rightarrow M} W_{F \rightarrow M} + I_M A_{M \rightarrow M} W_{M \rightarrow M} + I_{IO} A_{IO \rightarrow M} W_{IO \rightarrow M} + I_M S_M + B_M \quad (5.9)$$

$$I'_{IO} = I_M A_{M \rightarrow IO} W_{IO \rightarrow M} + I_{IO} S_{IO} + B_{IO} \quad (5.10)$$

1728 We thus have a S , W and B for each of the ϕ_u function we defined above. The IAW sum can be
 1729 seen as the ϕ_m function and $IS + B$ as the second part of the ϕ_u function. Eq 5.5 gave the generic
 1730 form of message passing : to update a node i , one first combines informations from the surrounding
 1731 nodes and edges and then combine the result ($\square_j \phi_m$) with the current features of node i . Many
 1732 practical ways to combine can be tried. In our implementation of message passing (Eq. 5.6 and 5.7)
 1733 the latter combination is the simple sum of the former (IAW , the equivalent of $\square_j \phi_m$) with a linear
 1734 combination of the current features of node i ($IS + B$).

1735 Interestingly, the number of learnable weight in those layer is independent of the number of nodes
 1736 in each family and depends solely on the number of features on the nodes and the edges.

1737 The expression above only update the node features. We could update the edges, using the results of
 1738 ϕ_m for example, but for technical simplicity we only update the nodes and keep the edges constant.
 1739 Preserving the edges after each layers allow to share the adjacency matrix between all layers, saving
 1740 memory and computing time.

1741 This operation of message passing is the constituent of our message passing layers, designed in this
 1742 work as *JWGLayer*, each of them owning their own set of parameter W , S and B . To those layers, we
 1743 can adjoin an activation function such as *PReLU*

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

1744 5.3 Data

1745 For this study we will be using a 1M positrons event dataset, uniformly distributed in energy with
 1746 $E_k \in [0, 9]$ MeV and uniformly distributed in the detector. Those events come from the JUNO

1747 official simulation version J23.0.1-rc8.dc1. All the event are *calib* level, with simulation of the physics,
 1748 electronics, digitizations and triggers. 900k events will be used for the training, 50k for validation
 1749 and loss monitoring and 50k for the results analysis in section 5.7. Each events is between 2k and
 1750 12k fired PMTS, resulting in fired nodes being the largest family in our graphs in all circumstances
 1751 as illustrated in figure 5.4c.

1752 As expected, by comparing the scale between the figure 5.4a and 5.4b we see that the LPMT system
 1753 is predominant in term of informations in our data. The number of PMT hits grow with energy but
 1754 do not reach 0 for low energy event due to the dark noise contribution which seems to be around
 1755 1000 hits per event for the LPMT system (left limit of figure 5.4a) and around 15 hits per event for the
 1756 SPMT system (left limit of figure 5.4b) which is consistent with the results show in section 4.1.2.

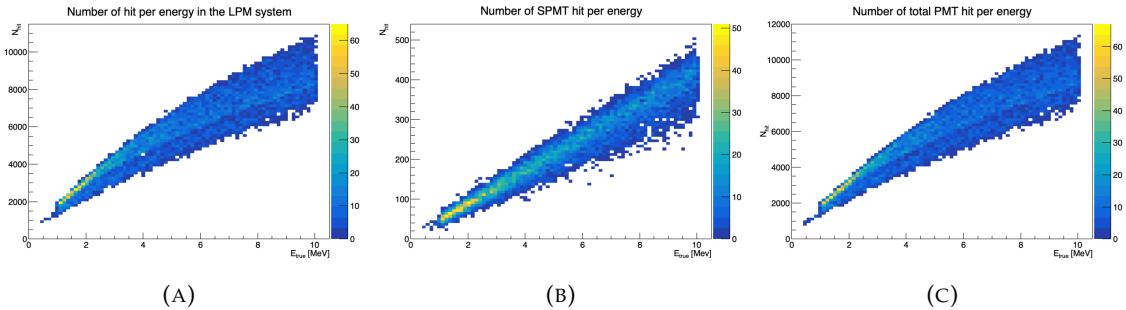


FIGURE 5.4 – Distribution of the number of hits depending on the energy. **On the right:** for the LPMT system. **In the middle :** for the SPMT system. **On the left:** For both system.

1757 The structure seen in the distribution in figure 5.4a comes from the shape of the number of hits
 1758 depending on the radius as shown in figures 5.5a and 5.5b where the number of hit decrease with
 1759 radius. It is important to understand that this is not representative of the number of PE per event
 1760 and the decrease in hits over the radius means that the PE are just more concentrated in a smaller
 1761 number of PMTs.

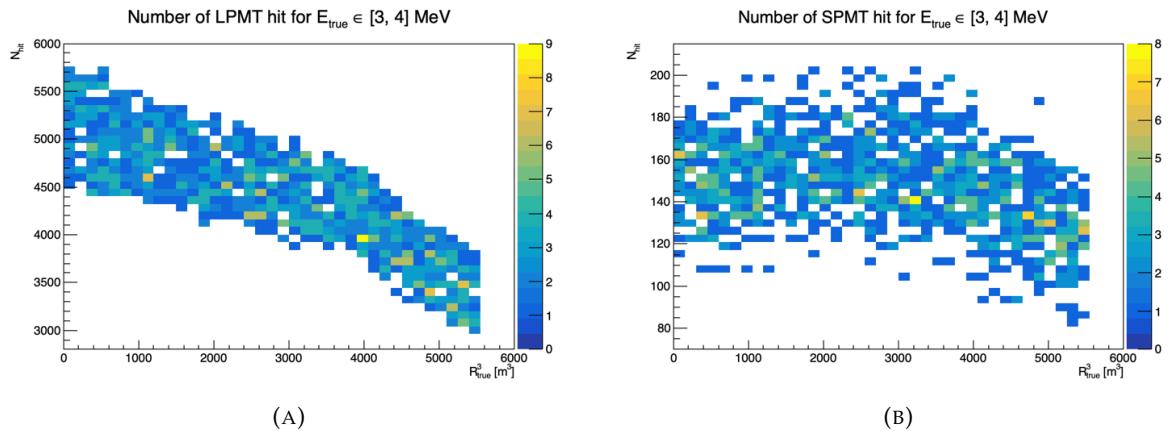


FIGURE 5.5 – Distribution of the number of hits depending on the radius. **On the right:** for the LPMT system. **On the right :** for the SPMT system. To prevent the superposition of structure of different scales we limit ourselves to the energy range $E_{true} \in [0, 9]$.

1762 No quality cut is applied here, we rely only on the trigger system. It means that event that would not
 1763 trigger are not present in the dataset but for events that triggered twice, it happens rarely, the two
 1764 trigger are considered as two separate event.

1765 5.4 Model

1766 In this section, we discuss the different layers that compose the final version of the model. The number
 1767 of layers, their dimensions, and their arrangement were fine-tuned through multiple iterations.
 1768 As mentioned earlier, each JWGLayer is defined by the number of features on the nodes and edges of
 1769 the output graph, assuming it takes as input the graph from the previous layer. For simplicity, when
 1770 discussing a graph configuration, it will be presented as follow: { N_f , N_m , N_{IO} , $N_{f \rightarrow m}$, $N_{m \rightarrow m}$, $N_{m \rightarrow f}$
 1771 } where

- 1772 — N_f is the number of feature on the fired nodes.
- 1773 — N_m is the number of features on the mesh nodes.
- 1774 — N_{IO} is the number of features on the I/O node.
- 1775 — $N_{f \rightarrow m}$ is the number of features on the edges between the fired and mesh nodes.
- 1776 — $N_{m \rightarrow m}$ is the number of features on the edges between two mesh nodes.
- 1777 — $N_{m \rightarrow f}$ is the number of features on the edges between the mesh nodes and the I/O node.

1778 Because we do not change the number of features on the edges, we can simplify the notation to { N_f ,
 1779 N_m , N_{IO} }. As an example, the input graph configuration, following the tables 5.1 and 5.2 is { 6, 8, 13,
 1780 5, 8, 5 } or, without the edge features, { 6, 8, 13 }.

1781 The final version of the model, called JWGV8.4.0 is composed of

- 1782 — An JWGLayer, converting the input graph { 6, 8, 13 } to { 64, 512, 2048 } with a PReLU activation
 function.
- 1783 — 3 resnet layers, each of them composed of
 - 1785 1. 2 JWG layers with a PReLU activation function. They do not change the dimension of the
 graph
 - 1786 2. A sum layer that sums the features in the input graph with the one computed from the
 JWG layers
- 1787 — A flatten layer that flatten the features of the I/O and mesh nodes in a vector.
- 1788 — 2 fully connected layers of 2048 neurons with a PReLU activation function.
- 1789 — 2 fully connected layers of 512 neurons with a PReLU activation function.
- 1790 — A final, fully connected layer of 4 neurons acting as the output of the network.

1793 A schematic of the model is presented in figure 5.6.

1794 We use the Mean Square Error (MSE) for the loss

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

1795 as it was the best resulting loss in Chapter 4.

1796 5.5 Training

1797 The optimizer used for training is the Adam optimizer and default hyperparameters ($\beta_1 = 0.9$,
 1798 $\beta_2 = 0.999$ and $\epsilon = 1e-8$) with a learning rate $\lambda = 1e-8$. The training last 200 epochs of 800
 1799 steps. We use a batch size of 32, the largest we can have with 40GB of GPU ram. The learning rate
 1800 is constant during the first 20 epochs then exponentially decrease with a rate of 0.99. We save two
 1801 set of parameters, the set of parameters the set that yield the lowest validation loss and the set of
 1802 parameters at the end of the training. The validation is computed over a single batch.

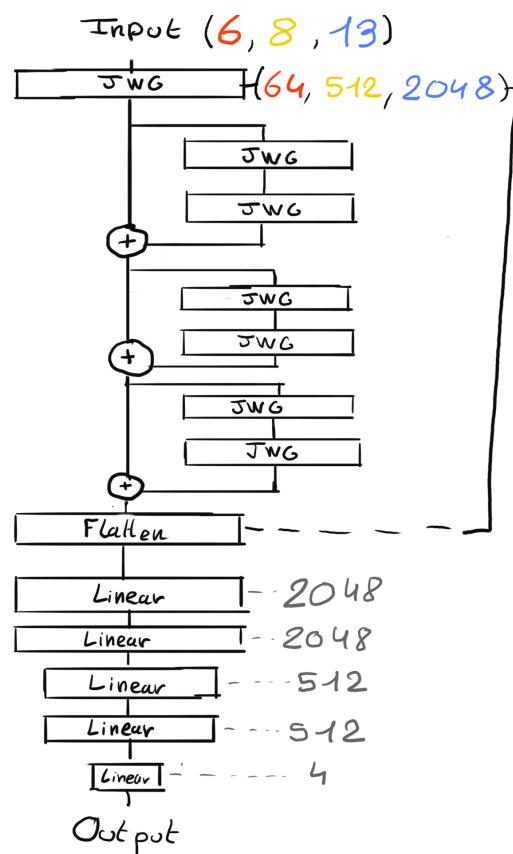


FIGURE 5.6 – Schema of the JWGv8.4.0 architecture, the colored triplet is the graph configuration after each JWG layers

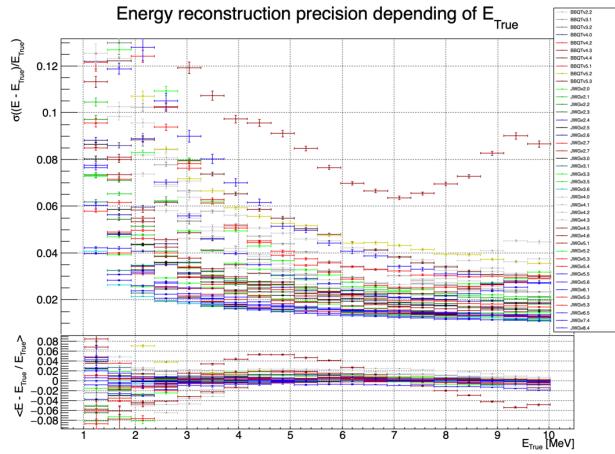


FIGURE 5.7 – Energy reconstruction depending on the true energy for samples of the different versions of the GNN

1803 5.6 Optimization

1804 The GNN model presented in previous sections is the result of a long work of optimization. Indeed,
 1805 the innovative architecture we propose left us with an infinity of possible configurations with no
 1806 guidance from prior works in literature nor in JUNO.

1807 In the end, more than 60 different configurations have been tested. This effort is illustrated on Figure
 1808 5.7¹, where the 40 configurations are compared in their ability to reconstruct the positron energy.
 1809 Although all configurations share the fundamental principles we base our innovative architecture
 1810 on (three different kinds of nodes and edges, usage of raw level features on some of them, usage of
 1811 higher level data on others, division of JUNO’s surface into regional pixels to form mesh nodes, the
 1812 very large number of edges connected to each mesh node, etc.), performances can vary a lot between
 1813 our first attempts (far beyond any acceptable energy resolution, and not even on this figure) and
 1814 recent ones. Therefore: the precise way to choose hyperparameters mattered a lot, regardless of the
 1815 relevance of the global architectural principles.

1816 The spectacular improvement between early and later configurations also explains the length of this
 1817 process : for long we hoped we would finally reach the classical performance, and it was tempting
 1818 to test yet another configuration.

1819 5.6.1 Software optimization

1820 A substantial effort was devoted to the data processing workflow. Transforming JUNO simulation
 1821 outputs into graphs is a computationally expensive task. Furthermore, due to the ever-changing
 1822 nature of the graph dimensions and features during optimization, preprocessing JUNO’s files by
 1823 precalculating the graphs and then reading them from files was not viable, as it would require a
 1824 large amount of disk space to store events for each version of the graph.

1825 Therefore, the software does not rely on preprocessed data and instead computes the observables,
 1826 adjacency matrix, etc., during training. This data processing is performed in parallel on the CPU.
 1827 The raw data comes from ROOT files produced by the collaboration software, and the Event Data
 1828 Model (EDM), used internally by the collaboration [74], had to be interfaced with our software,
 1829 an interface that had to be maintained as the collaboration’s software evolved. For the harmonic

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

1830 power calculation, we migrated from the Healpix library to Ducc0 [75] for more precise control over
 1831 multithreading.

1832 5.6.2 Hyperparameters optimization

1833 The first kind of hyper-parameters that received a lot of effort concern the network's detailed archi-
 1834 tecture:

- 1835 — Message passing layers where originally not JWG layers, we started by using small FCDNN
 1836 in place of ϕ_u and ϕ_m . Due to low performances and memory consumption issues, we pivoted
 1837 to the message passing algorithm presented in section 5.2.
- 1838 — The ResNet architecture was brought after issue with the gradient vanishing.
- 1839 — The number of layers was varied between 5 and 12.
- 1840 — The number of node features after each given message passing layer (64, 512, 2048 in the final
 1841 version) was varied.
- 1842 — The Final FCDNN after the message passing layers is not present in all versions.
- 1843 — At some point, the PReLU activation function replaced the ReLU function.

1844
 1845 For some of them, software work was necessary. In any case, each configuration required a training
 1846 of about 90h. Adding the analysis time necessary to the verification of its performance and the
 1847 comparison with other versions, one understands the number of tests had to be limited.

1848 Other hyperparameters were also tested :

- 1849 — The higher level variables described in section 5.1 (powers of various spherical harmonics, \mathbb{A} ,
 1850 \mathbb{A} , $(Q_{m1} - Q_{m2})/(Q_{m1} + Q_{m2})$) were added progressively. Notice that our choice to focus
 1851 our search on this kind of variables is also due to the fact that JWGLayer involves linear
 1852 operations. It is therefore difficult for such a network to propose variables of this kind among
 1853 the node features learned layers after layers (i.e. it's difficult for the network to understand
 1854 these variables are important, or only after many layers).
- 1855 — Time allocated to training, the Learning Rate, the size of batches, etc.
- 1856 — The number of pixels (ie of mesh nodes) was varied between 192 and 768.
- 1857 — Several definitions loss functions where tried. In particular, we tried some focussed only on
 1858 the E resolution, only on the vertex resolution (R) or trying to optimize both.

1859
 1860 To make a long story short, each new configuration was the result of our reflections after having
 1861 analysed the previous configurations, or after having thought over again about JUNO's detailed
 1862 response to energy deposits – seeking for variables that could help the GNN.

1863 Another, quite common, approach was in principle possible : a random search. However, due to the
 1864 extensive training time, up to 90h per training, the heavy memory consumption of the models that
 1865 would often exceed the 20GB limit of the V100, this approach was not realistic in our case, though we
 1866 were able to extend the memory limit to 40GB thanks to a local A100 GPU card available at Subatech.

1867 5.7 performance of the final version

1868 The reconstruction performance of "JWGv8.4" are presented in figures 5.8, 5.9, 5.10 and compared to
 1869 the "Omilrec" algorithm, the official IBD reconstruction algorithm in JUNO. Omilrec is based on the
 1870 QTMLE reconstruction method that was presented in section 2.7.

1871 This comparison required to use a consistent definition of E_{true} . This is not trivial since at JUNO,
 1872 ML method reconstruct the true energy deposited by the positron+annihilation gammas (that's the

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

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

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

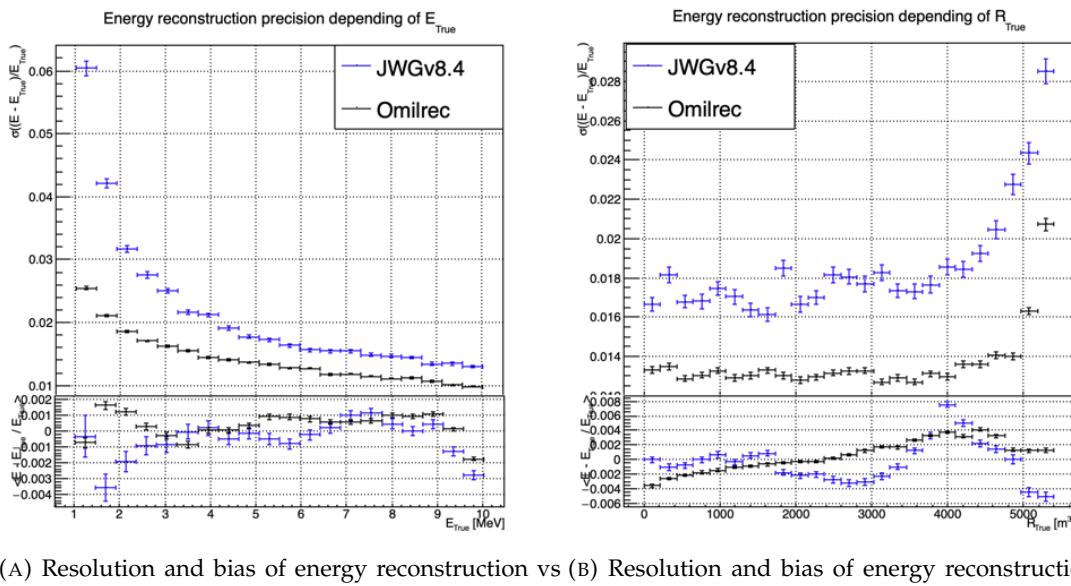


FIGURE 5.8 – Reconstruction performance of the Omilrec algorithm based on QTMLE presented in section 2.7, JWGV8.4 presented in this chapter. The top part of each plot is the resolution and the bottom part is the bias.

The first action that can be carried out in this direction was to determine if some information used by OMILREC was not used properly by JWGV8.4. For that purpose, we used again the approach presented in Chapter 4 (Sec 4.3.2 and annex A) to combine JWGV8.4 and OMILREC. We observe on figures 5.11 and 5.12 that this combination brings no sizeable improvement of the best of the two combined methods. The combination remains very close to OMILREC alone. This is an indication that JWGV8.4 does not use informations that would be overlooked by OMILREC, and that on the contrary, that's JWGV8.4 that fails to use properly important informations.

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

We tested this idea in several manners, described below.

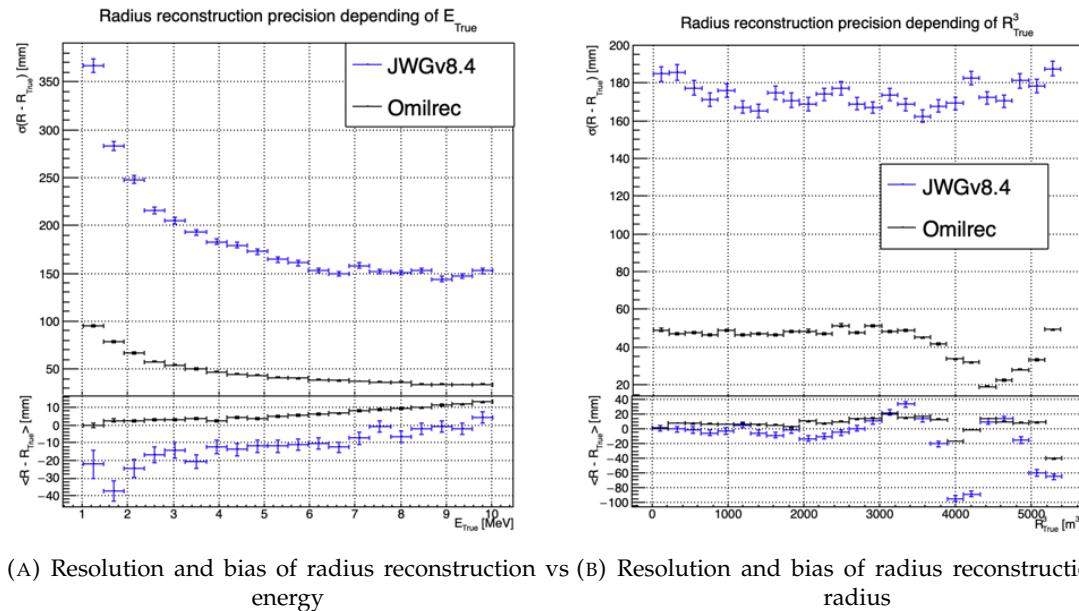


FIGURE 5.9 – Reconstruction performance of the Omilrec algorithm based on QTMLR presented in section 2.7, JWGv8.4 presented in this chapter. The top part of each plot is the resolution and the bottom part is the bias.

1895 Finer granularity

1896 We tried to recover some redundancy by increasing the number of mesh nodes from 198 to 768. The
 1897 improvement we observed was small, and did not allow to get close to OMILREC's performance.

1898 To explore further in this direction, we would ideally try 3072 pixels (the next HEALPIX rank).
 1899 However, this is not possible for our GNN due to hardware limitations, mainly the available GPU
 1900 memory. Instead, we discussed the problem with Gilles Grasseau, calculus research engineer with
 1901 whom we collaborate on the subject of ML reliability (see Chapter 6). In the framework of this ac-
 1902 tivity, Gilles needs to develop reconstruction algorithms to be "attacked" by a prototype Adversarial
 1903 NN. One of them is a pseudo-spherical CNN using oriented filters, called HCNN.

1904 To produce its input image, this algorithms split the Sphere into 3072 pixels. Each channel of this
 1905 image is an aggregation of the (Q, t) values found in all the PMTs. The charge are summed and
 1906 the lowest time is kept. The performance of this algorithm can be seen on Figures 5.13 and 5.14,
 1907 compared to OMILREC. With 3072 pixels, the performance of HCNN does not match that of OMIL-
 1908 REC, but is closer to it than our GNN. The granularity of the pixels, and the way to summarize the
 1909 individual PMTs information when going from 17000 LPMTs to only 3072 pixels indeed seems to
 1910 play a role.

1911 This is consistent with the results obtained by the first GNN tried at JUNO on reactor neutrinos
 1912 (already described in section 2.7.3). It used 3072 pixels, and also obtained an uncompetitive R
 1913 reconstruction.

1914 Information reduction, from fired to Meshes

1915 The problem described above is somehow classical. ML algorithms, ideally, would start from the full
 1916 information present in the detector, and learn to reduce it optimally.

1917 In cases where only 3072 pixels can be used instead of the complete information from 17000 PMTs,

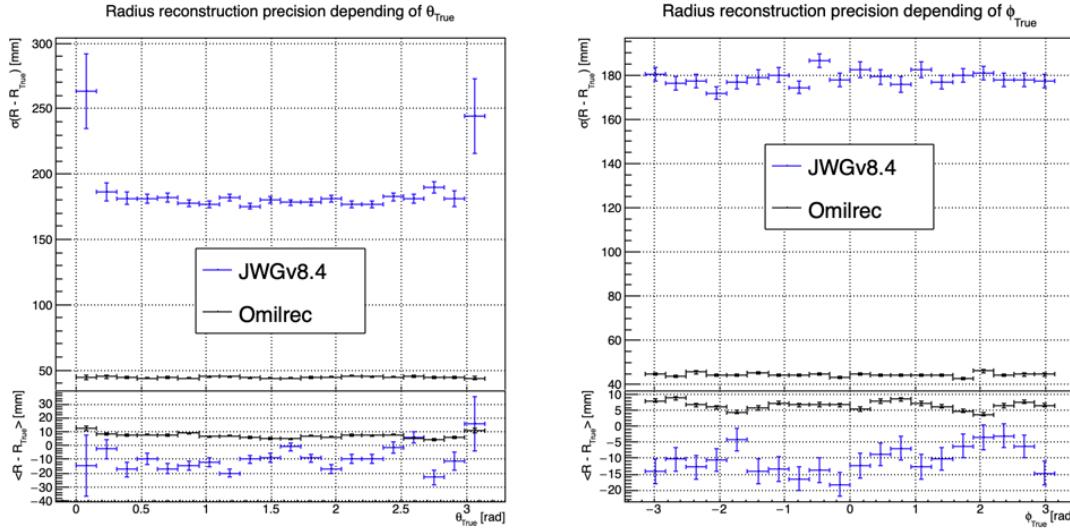


FIGURE 5.10 – Reconstruction performance of the Omilrec algorithm based on QTMLE presented in section 2.7, JWGv8.4 presented in this chapter. The top part of each plot is the resolution and the bottom part is the bias.

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

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

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

1930 Possible improvements

1931 It appears that the network is unable to aggregate the timing information correctly. While this could
 1932 be addressed by using a finer segmentation, with more mesh nodes, improvements might also arise
 1933 from refining the message-passing algorithm. The algorithm presented in this thesis is still quite
 1934 basic, relying on a simple linear combination of features. We have seen through examples in CNNs,
 1935 GNNs, and other architectures, both in research and industry, that specializing the network — for
 1936 instance, by incorporating convolutional filters — can lead to improvements that were previously
 1937 unattainable with simpler FCDNNs. Applying this approach to the message-passing algorithm, by
 1938 utilizing a GNN with a more advanced message-passing, could yield better results.

1939 Regarding the timing information, we provided high-level features, assuming this would assist the
 1940 neural network in converging to the solution. However, by offering such information upfront,
 1941 the GNN might be taking the “easy” path, settling for a local and broader minimum, rather than
 1942 extracting the features that could lead to better performance.

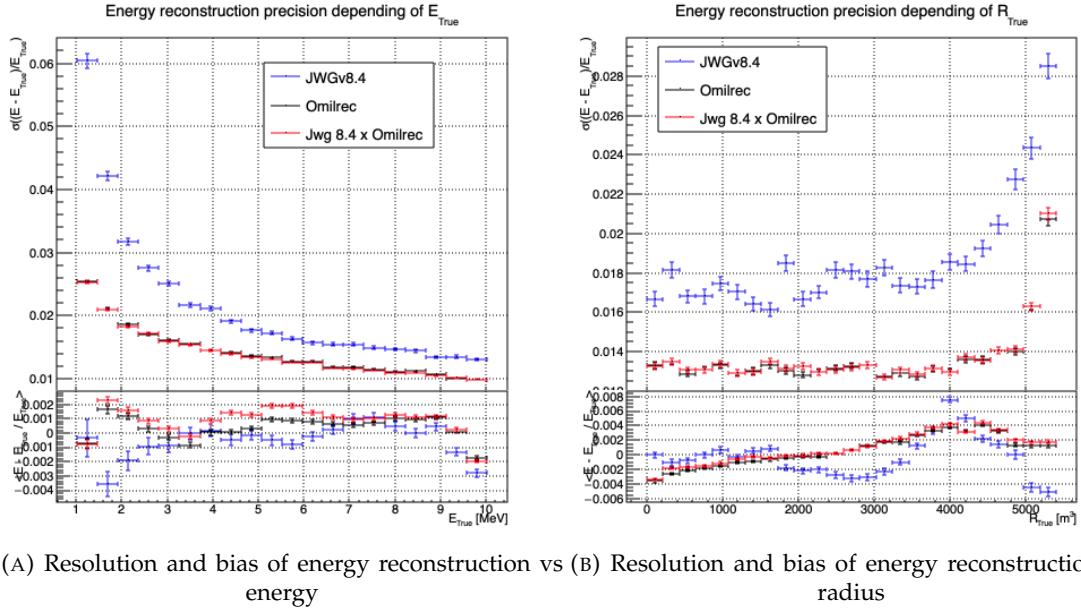


FIGURE 5.11 – Reconstruction performance of the Omilrec algorithm, JWGV8.4 and the combination between the two using the optimal variance estimator presented in annex A.2. The top part of each plot is the resolution and the bottom part is the bias.

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

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

5.8 Conclusion

To achieve its scientific goals, JUNO requires a precise and well-understood reconstruction, as it needs an energy resolution of 3% at 1 MeV. Even small, unaccounted biases could make it impossible to determine the mass ordering, as explored in Chapter 7. A likelihood-based algorithm, designed to meet JUNO's requirements and referred to as the classical algorithm, was developed and is detailed in section 2.7.

Machine learning algorithms were developed to challenge this classical approach, and they are presented in Section 2.7.3. Although they achieve the precision of the classical algorithm, they do not offer significant improvements. The GNN previously developed is a convolutional GNN where nodes correspond to pixels, connected to their neighbors based on the Healpix [73] segmentation, with the (Q, t) information aggregated onto these pixels.

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

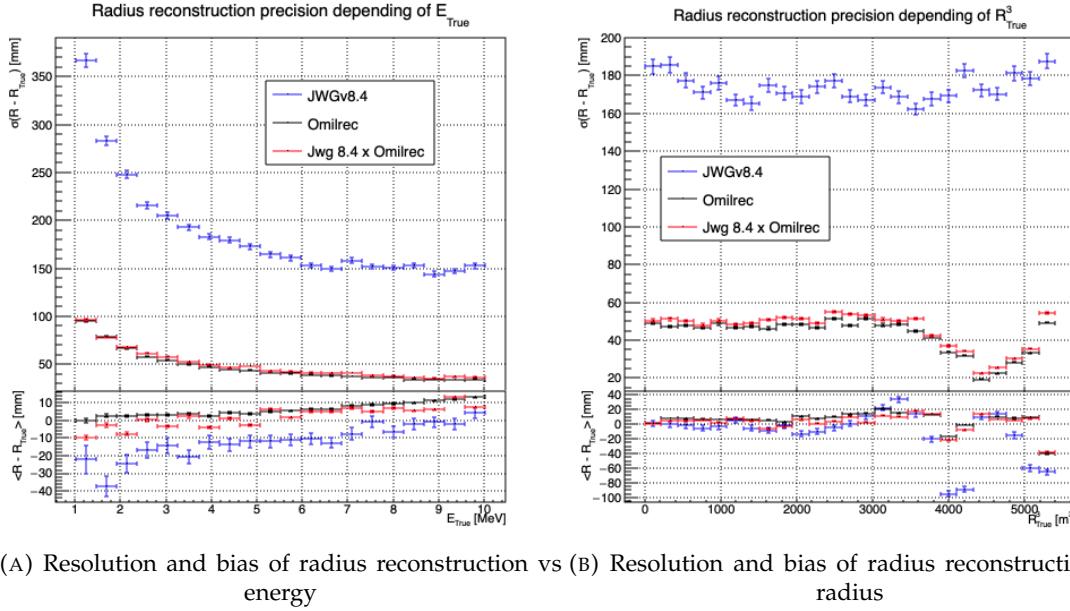
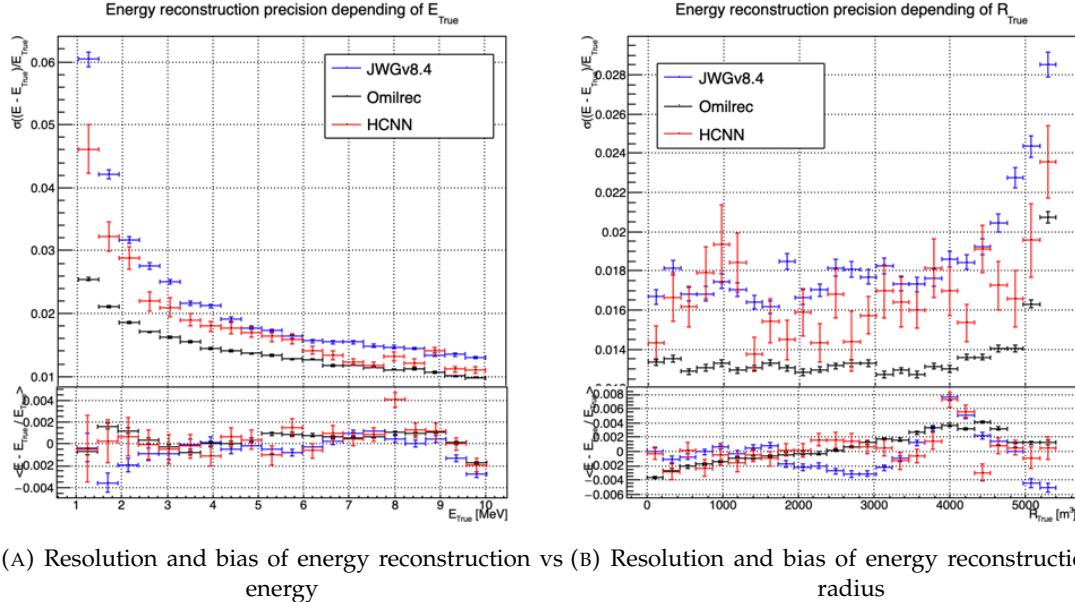


FIGURE 5.12 – Reconstruction performance of the Omilrec algorithm, JWGV8.4 and the combination between the two using the optimal variance estimator presented in annex A.2. The top part of each plot is the resolution and the bottom part is the bias.

1966 These three types, or families, of nodes do not have the same number of features, resulting in a het-
 1967 erogeneous graph. Publicly available algorithms for graph processing are designed for homogeneous
 1968 graphs, so we had to develop a custom algorithm adapted to heterogeneous graphs.

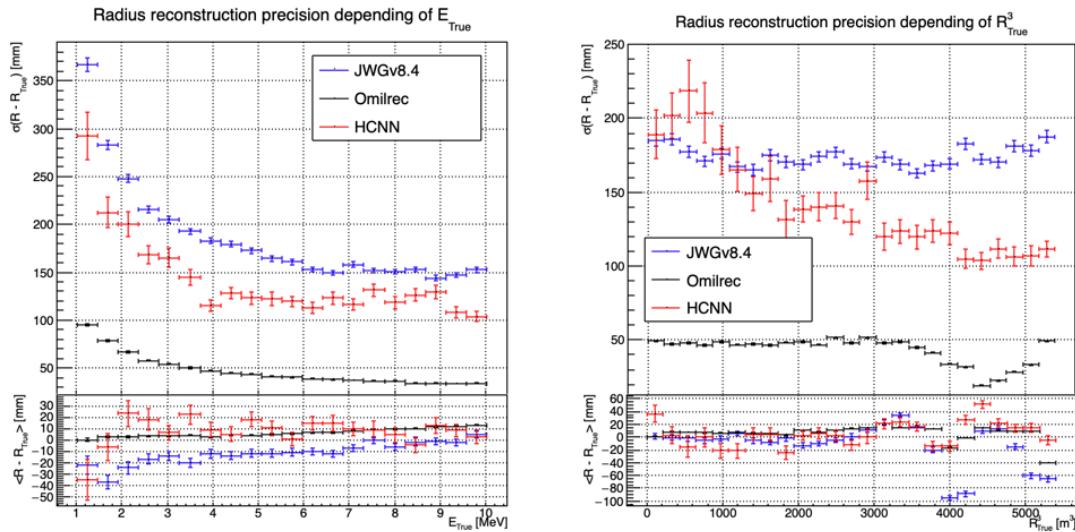
1969 This GNN required significant technical development, but the results are not at the level of the
 1970 classical algorithm. The tests we conducted suggest that the problem may lie in the aggregation
 1971 of raw information from the fired nodes onto the mesh nodes, as removing the fired nodes does
 1972 not degrade the results. Additionally, due to technical constraints, we had to reduce the number of
 1973 pixels compared to the previous GNN. Other algorithms we developed, which use a higher pixel
 1974 resolution, outperform this architecture, reinforcing our suspicion that the aggregation is the root of
 1975 the issue.

1976 Perhaps by incorporating rawer information, such as the waveform, refining the message-passing
 1977 algorithm, or adjusting the features on the different nodes, we could match the precision of the
 1978 classical algorithm. However, it is also possible that deeper, more radical changes are needed to
 1979 become competitive.



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

FIGURE 5.13 – Reconstruction performance of the Omilrec algorithm based on QTMLE presented in section 2.7, JWGv8.4 presented in this chapter and the HCNN algorithm. The top part of each plot is the resolution and the bottom part is the bias.



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

FIGURE 5.14 – Reconstruction performance of the Omilrec algorithm based on QTMLE presented in section 2.7, JWGv8.4 presented in this chapter and the HCNN algorithm. The top part of each plot is the resolution and the bottom part is the bias.

1980 **Chapter 6**

1981 **Reliability of machine learning
methods**

1983

"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

1984

Contents

19851986198719881989199019911992199319941995199619971998

6.1	Method	92
6.2	Architecture	92
6.2.1	Back-propagation problematic	93
6.2.2	Reconstruction Network	94
6.2.3	Adversarial Neural Network	94
6.2.4	Training	95
6.3	Results	95
6.3.1	Back to identity	95
6.3.2	Breaking of the reconstruction	95
6.4	Conclusion and prospect	95

1999 As explained in previous chapters, JUNO is a precision experiment where the complete understanding of the effects at hand is crucial. As it will be illustrated in Chapter 7, even small invisible biases or 2000 uncertainties could lead to the impossibility to run the measurements, or even worse, wrong our mass 2001 ordering measurements. While the liquid scintillator technology is well known and straightforward, 2002 this is the first time it is deployed to such scale, and for such precision. This novelty brings its fair 2003 share of elements, effects or assumption, that, if they were to be overlooked, could cause issue. 2004

2005 We already shown a large variety of reconstruction algorithms, OMILREC for LPMT reconstruction 2006 in section 2.7, numerous machine learning algorithms in section 2.7.3 and our own work in chapters 2007 4 and 5. Those algorithms were compared to each other based on their performance as in [40] but 2008 we are the first that looked into the correlation between the reconstruction. The combinations of 2009 algorithms shown in Chapter 4 and Chapter 5 show that some information elude the algorithms. 2010 We used this fact to try to improve our performance but this could also lead the algorithm to being 2011 vulnerable to some effect that could affect the detector and wrong the measurements.

2012 The search for such effect could be done by hand, but the process would be tedious. We propose 2013 in this thesis a machine learning method to probe for those effects. In section 6.1, I describe the 2014 method behind the algorithm. In section 6.2 I detail the architecture of our algorithm and in section 2015 6.3 the results of it. Finally, in section 6.4, I conclude and discuss about the prospect and possible 2016 improvements to bring to this work.

2017 **6.1 Method**

2018 As introduced above, JUNO needs a very good understanding of the biases and effects affecting its
 2019 reconstruction as a small bias could wrong the mass ordering measurement. To calibrate those biases
 2020 and effect, JUNO rely on multiples sources that can be located at various point in the detector. The
 2021 calibration strategy was already discussed in section 2.4 and show calibrations sources of gammas,
 2022 neutrons and positrons, with the catch that the positrons will annihilate inside the encapsulation and
 2023 only the two 511 keV gammas will be seen.

2024 None of the calibrations sources considered are positron event. While electrons and positrons events
 2025 should be pretty similar in their interaction with the electronic cloud of the LS atoms, electron events
 2026 are missing the two annihilations γ and the potential of forming a positronium [76]. The topology of
 2027 the event thus differ of the order of magnitude of our reconstruction performance: a few nanoseconds
 2028 for the energy deposit and positronium annihilation against a time transit spread between 3 and 6 ns
 2029 depending on the PMT type [77–79] and the γ will travel distances of the order of magnitude of the typical LPMT resolution of 8 cm (see section 2.7).

2031 Another natural calibration source is the ^{12}B spectrum. The ^{12}B is a cosmogenically produced isotope
 2032 through the passage of muons inside the LS. The ^{12}B decays via β^- emissions with a Q value of
 2033 13.5 MeV with more than 98% of the decay resulting in ground state ^{12}C . The ^{12}B event will be
 2034 cleanly identified by looking for delayed high energy β events after an energetic muon. Due to its
 2035 natural causes, the ^{12}B events will be uniformly distributed in the detector. The calibration strategy
 2036 consist in fitting the energy spectrum of ^{12}B with the results of the simulation to adjust the simulation
 2037 parameters. Both sources will be used to *control* the response of the detector.

2038 Unlike lasers and radioactive, from which the localization and energy will be well known, the individual truth of ^{12}B will be unknown with only the localisation loosely constrained by the muon track. Only higher order observables such as the energy distribution will be accessible.

2041 All of those considerations could hide potential unknown or undetected effect that could lead to issue in the mass ordering analysis. But, while we have idea from where the issue could come, the production by hand of event perturbations that go unseen in the calibration would be tedious. That's why we propose to use a Neural Network to produce those perturbations if they exists. A schematic of the concept is presented in figure 6.1.

2046 This network should produce physically sound perturbation, that would not be seen by the calibration but also by the visualisation of the event. If the ANN manage to produce such perturbations, we can derive systemic uncertainties from it. If it fail to find some, it is a proof of robustness for the attacked reconstruction method.

2050 For this study we consider a “physics” dataset composed of 1M positron events from J23, uniformly distributed in the Central Detector (CD) and in deposited energy between $E_{dep} \in [1.022; 10.022]$. This set represent the IBD events we want to *wrongly* reconstruct.

2053 We use a second “control” dataset of electron event also uniformly distributed in the detector and over the same energy range. They mimic the energy deposit of ^{12}B decay and are used as the sample to compute the control observables.

2056 **6.2 Architecture**

2057 We can describe the goal of the ANN by the following loss function:

$$\mathcal{L} = \mathcal{L}_{adv} + \mathcal{L}_{reg} \quad (6.1)$$

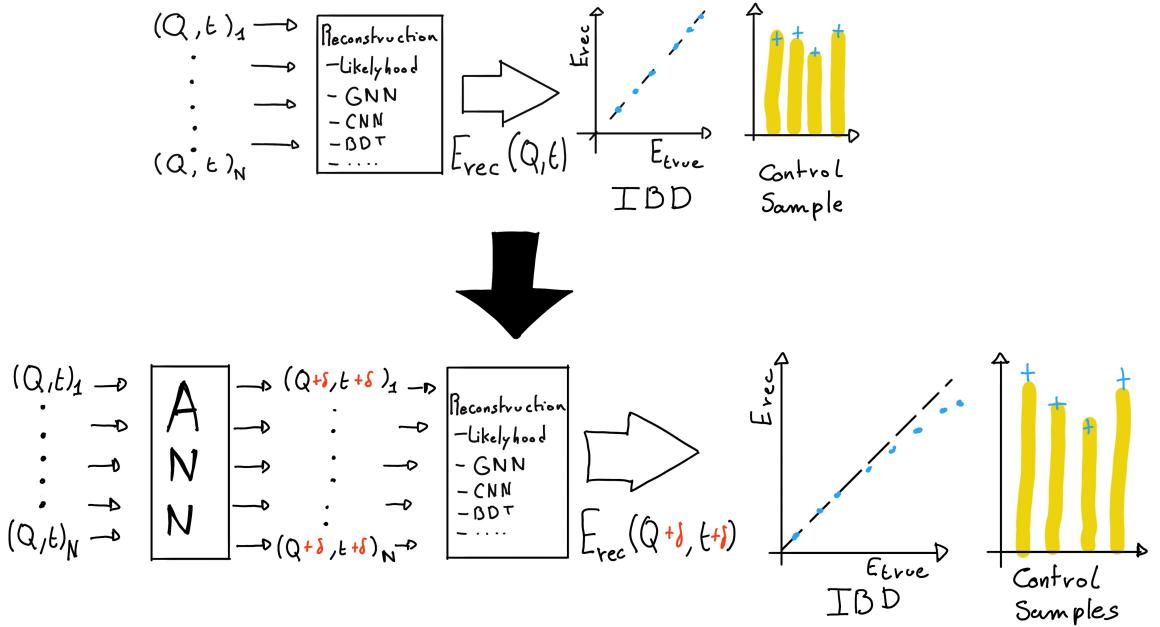


FIGURE 6.1 – Schema of the method to discover vulnerabilities in the reconstruction methods. **On the top** of the image, the standard data flow. The individual charge and times are fed to a reconstruction algorithm. From the reconstructed energies, we can produce an IBD spectrum and compute control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

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

$$\mathcal{L}_{adv} = |\text{Corr}(E_{rec}, E_{dep})| \quad (6.2)$$

which is positive or null and is minimal when the reconstructed energy is decorrelated with the deposited energy, the reconstruction is wrong.

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

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

where λ index the different control observables that will be considered in this study. It's minimal when the control observables after perturbation O_{λ}^{rec} are coherent with their expected values O_{λ}^{th} .

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

6.2.1 Back-propagation problematic

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

event X' , we can parametrize the equation 6.1

$$Y = \mathcal{F}(X); Y' = \mathcal{F}(X') = \mathcal{F}(\mathcal{G}(X)) \quad (6.4)$$

$$\mathcal{L} \equiv \mathcal{L}(\mathcal{F}(\mathcal{G}(X)), Y_t) \quad (6.5)$$

where Y_t is the reconstruction target of Y .

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

$$\frac{\partial \mathcal{L}(\mathcal{F}(\mathcal{G}(X)))}{\partial \theta} \quad (6.6)$$

which, when using the chain rule, become

$$\frac{\partial \mathcal{L}(\mathcal{F}(\mathcal{G}(X)))}{\partial \theta} = \frac{\partial \mathcal{G}}{\partial \theta} \cdot \frac{\partial \mathcal{F}}{\partial \mathcal{G}} \cdot \frac{\partial \mathcal{L}}{\partial \mathcal{F}} \quad (6.7)$$

The terms $\frac{\partial \mathcal{G}}{\partial \theta}$ and $\frac{\partial \mathcal{L}}{\partial \mathcal{F}}$ are easily computable but $\frac{\partial \mathcal{F}}{\partial \mathcal{G}}$ depends on the nature of the reconstruction algorithm. While it comes naturally when using NN algorithms, it's not so trivial for other kind of algorithms like likelihood. Solutions exists to optimize networks that work in complex, non differentiable environments, such as *Deep Reinforcement Learning* [80, 81] but as a first prototype we will restrict ourselves to neural networks for the reconstruction algorithm.

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

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

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

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

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

6.2.2 Reconstruction Network

As introduced just before, we need a NN algorithm for IBD reconstruction. We could have used the GNN exposed in Chapter 5

6.2.3 Adversarial Neural Network

— Décrire l'architecture de l'ANN

2098 6.2.4 Training

- 2099 — Presentation du dataset
2100 — 2 etapes d'entraînement
2101 — Retour à l'identité -> que l'ANN ne fasse pas n'importe quoi
2102 — Cassage de la reconstruction

2103 Hyperparameter optimization

- 2104 — Pour les même raison que l'ANN:
2105 — Phase exploratoire, architecture très changeante, random search n'est pas viable
2106 — Architecture consomme beaucoup, besoin d'entraîner sur l'A100
2107 — Possiblement que de l'optimization permettrait de faire passer sur V100, mais développement techniques nécessaires.
2108

2109 6.3 Results

- 2110 — Voir slide Gilles

2111 6.3.1 Back to identity**2112 6.3.2 Breaking of the reconstruction****2113 6.4 Conclusion and prospect**

- 2114 — Not enough
2115 — Probably guide the ANN

²¹¹⁶ **Chapter 7**

²¹¹⁷ **Dualcalorimetric analysis for Precision
Measurement**

²¹¹⁹ “We demand rigidly defined areas of doubt and uncertainty!”
Douglas Adams, The Hitchhiker’s Guide to the Galaxy

²¹²⁰ **Contents**

²¹²¹ 7.1 Motivations	²¹²²	²¹²³ 99
7.1.1 Discrepancies between the SPMT and LPMT results	99
7.1.2 Charge Non-Linearity (QNL)	100
7.2 Approach	²¹²⁵ 101
7.2.1 Data production	102
7.2.2 Individual fits	103
7.2.3 Joint fit	104
7.2.4 Data and theoretical spectrum generation	105
7.2.5 Limitations	106
7.3 Fit software	²¹³¹ 106
7.3.1 IBD generator	107
7.3.2 Fit	108
7.4 Technical challenges and development	²¹³⁴ 109
7.5 Results	²¹³⁵ 109
7.5.1 Validation	109
7.5.2 Covariance matrix	113
7.5.3 Statistical tests	118
7.6 Conclusion and perspectives	²¹³⁹ 120

²¹⁴³ JUNO is precision measurement experiment. To determine the NMO with the aimed significance,
²¹⁴⁴ JUNO must be sensitive to the tiny spectral phase shift shown on figure 7.1. Once detection effects
²¹⁴⁵ are accounted for, the difference between IO and NO spectra is further reduced, as can be seen on
²¹⁴⁶ figure 7.2.

²¹⁴⁷ Among other condition, a precise and complete understanding of the reconstruction and detector
²¹⁴⁸ effects is crucial. The challenge reside in the technology used in the detector, which, while based
²¹⁴⁹ on well known technology: scintillator observed by PMT, is being deployed on a scale never seen
²¹⁵⁰ before, in term of scintillator volume and PMT size. Understanding every effects that goes in the
²¹⁵¹ detector can become extremely complicated. Any method to help detecting problems is therefore
²¹⁵² welcome. Comparing the data and results obtained by two systems measuring the same events, but
²¹⁵³ subject to different sources of error, is therefore precious. This is the purpose of the dual calorimetry
²¹⁵⁴ techniques used in JUNO thanks to the existence of 2 PMT systems: the LPMT and SPMT systems.

The reconstruction of the IBD positron energy must be very performant: an unprecedented resolution of 3% at 1 MeV [48] is necessary to determine the NMO with the aimed significance. Moreover, it is necessary to know the energy scale with an uncertainty below 1% to correctly evaluate in our data the likelihood of the NO and IO hypotheses. Beyond that value, the risk progressively appears to exclude the NI(IO) hypothesis with the significance with which one should actually have excluded the IO(NO) if the energy scale was precisely known, as can be seen in the introduction of Chapter 4 of [25].

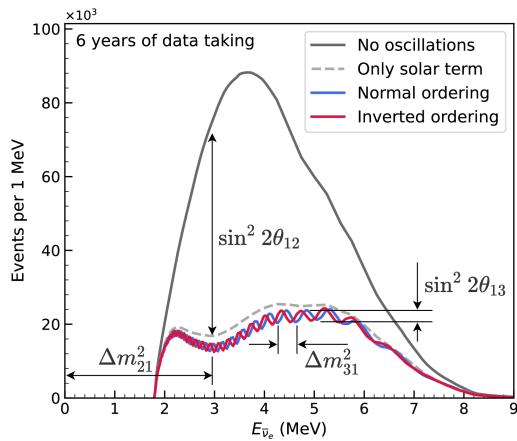


FIGURE 7.1 – Expected number of neutrinos event per MeV in JUNO after 6 years of data taking. The black curve shows the flux if there was no oscillation. The light gray curve shows the oscillation if only the solar terms are taken in account (θ_{12} , Δm_{21}^2). The blue and red curve shows the spectrum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

One of the possible source of non-linearity, which will be used as a reference in this chapter, is the charge non-linearity (QNL) that will be discussed in next section. Several dual calorimetry techniques can address this issue. Some are calibration techniques, that are also described in section 4.3 of [25]. More generally, comparing the results of the two systems will allow for the detection of potential issues on the calibration or reconstruction. This is done in this thesis by comparing directly the spectra and oscillation parameters measurements of the two PMT systems. We call this kind of dual calorimetry "Dual calorimetry with neutrino oscillation", since it is based on the visible energy spectra used by the oscillation analysis of reactor antineutrinos.

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

In the next section we will discuss the motivations behind this study. In section 7.2, I present the methods we propose to implement Dual calorimetry with neutrino oscillation, and of the way we

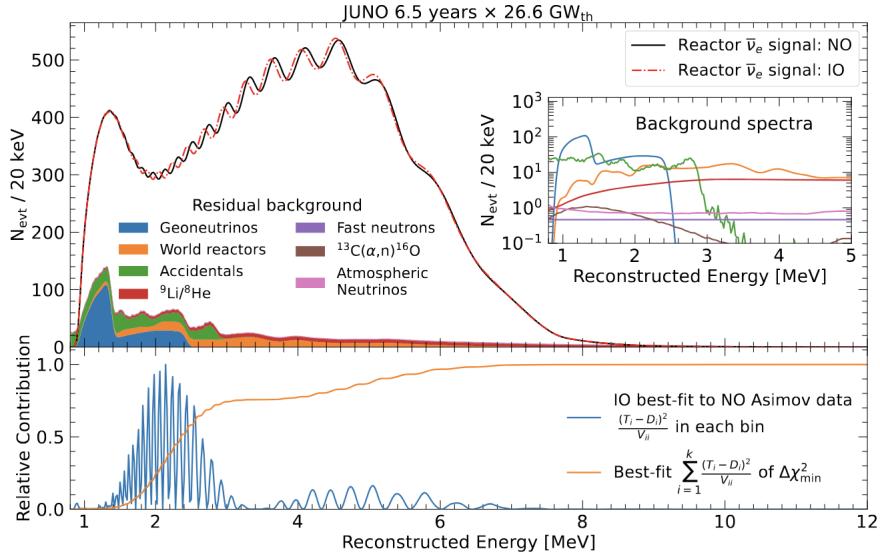


FIGURE 7.2 – Oscillated reactor $\bar{\nu}_e$ spectra for the Normal Ordering (Black) and Inverted Ordering (Red) for 6,5 years data taking and a resolution of 3% without any statistical or systematic fluctuation. Figure from [82].

2184 estimate their sensitivity. In section 7.3, I present the fit framework used, and then, in section 7.4
 2185 the technical improvement brought and the difficulties faced during the development. To end this
 2186 chapter I present the results in 7.5 and discuss the conclusions and perspectives in 7.6.

2187 7.1 Motivations

2188 7.1.1 Discrepancies between the SPMT and LPMT results

2189 As discussed in the introduction of this chapter, the SPMT and LPMT systems will observe the same
 2190 events. This mean that, after calibration, if the two system show significant differences in their results
 2191 this is the signal of potential overlook of an effect or problem. Being able to detect such differences
 2192 is thus crucial, as discussed above, even the smallest deviation from our model could lead to the
 2193 impossibility to measure the Mass Ordering (MO) or even worse, wrong our measurement.

2194 The two systems are expected to have the same sensitivity to the oscillation parameters θ_{12} and Δm_{21}^2
 2195 [3]. We will thus rely on the measurement of those two parameters to detect potential discrepancies.

2196 We could just look at the value and compare them to the estimated independent error of the two
 2197 system, but we believe and will demonstrate in this chapter that the independent study of the two
 2198 system is missing a lot of informations, and that, by taking into account the statistic and systematic
 2199 correlations between the two systems, we can produce much more powerful statistical tests.

2200 Our work in this chapter is to develop such tools, which in practice implies to define test statistics. A
 2201 first step will be to determine the distribution of these test statistics in the case when no unexpected
 2202 problem affects the LPMT nor the SPMT problem. This will give us the distribution of those statistical
 2203 test in absence of discrepancies. Later, the value of the test statistics that we will measure in real data
 2204 can be compared to these distributions to produce p-values, to judge of the potential present of an
 2205 unexpected effect.

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. Note that these tests should in principle be able to detect unexpected effects whatever their source (calibration issues, insufficient simulation tuning, etc.), provided that the distortion caused to the energy spectrum is important enough.

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 LS response is composed of physical non-linearity. Particle interactions in the LS will produce mainly scintillation light, as discussed in section 2.3.2, 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 [14]. This result in a event-wise non-linearity.

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

Precedent studies [25] suggest a model for the channel-wise QNL:

$$\frac{Q_{rec}}{Q_{true}} = \frac{-\gamma_{qnl}}{9} Q_{true} + \frac{\gamma_{qnl} + 9}{9} \quad (7.1)$$

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

Studies at previous experiments, like Daya Bay, concluded that the best reachable control of QNL in the 1-10 PEs range was $\gamma_{qnl} = 0.01$ [83]. As already mentionned in section 2.3.2, JUNO LPMTs operate in a larger range : 1-100 PEs (See also table 7.1). In such a case, a realistic value of γ_{qnl} is not known.

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

TABLE 7.1 – The charge fraction in terms of the number of PE collected at the single PMT for the reactor $\bar{\nu}_e$ IBD events. Table taken from [25]

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

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

2240 In JUNO, the visible energy is proportional to the number of emitted photons per unit energy deposit.
 2241 It includes the physical non linearities. In the equation above E_{vis}^{true} is this visible energy, while E_{vis}^{rec}
 2242 is what it becomes when the reconstructed charges found in an event are modified according to Eq.
 2243 7.1.

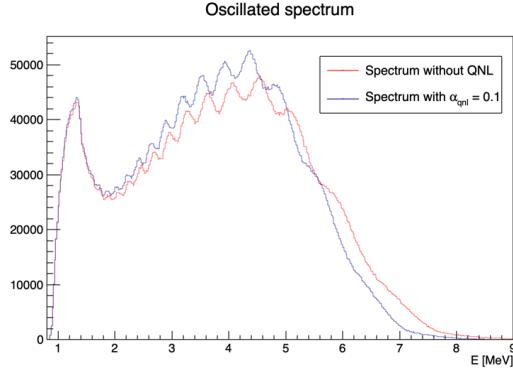


FIGURE 7.3 – Two oscillated spectra of $1e7$ event expected in JUNO. In red the spectrum without supplementary QNL. In blue the same spectrum but where an event-wise QNL $\alpha_{qnl} = 10\%$ is introduced.

2244 An example is shown on Fig. 2.14, where we show the $E_{vis}^{rec}/E_{vis}^{true}$ ratio for several samples of
 2245 uniformly distributed electron events, generated with various values of E_{vis}^{true} . Here, an extreme
 2246 value $\gamma_{qnl} = 0.05$ was assumed. On can see on Fig. 2.14 that it corresponds to a 2% effect at 8
 2247 MeV, equivalent to $\alpha_{qnl} = 0.025$.

2248 This example is from references [25], which aimed at demonstrating the potential of the dual
 2249 calorimetry calibration method mentioned in section 2.4.3. If it works as hoped, the residual event-
 2250 wise QNL effect will be below 0.3%. In this chapter, we propose methods to detect residuals higher
 2251 than this.

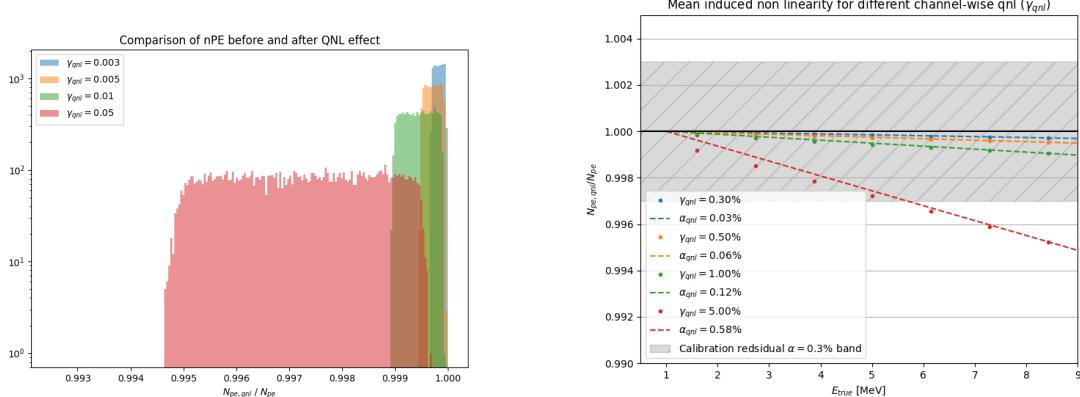
2252 Fig. 7.4b show several other examples with varying γ_{qnl} values, and the corresponding values of α_{qnl} .
 2253 Using 1M events from the JUNO official simulation J23.0.1-rc8.dc1 (released on 7th January 2024), we
 2254 simulated events up to the photon collection in LPMTs and introduced an additional channel-wise
 2255 QNL by using the equation 7.1 to modify the number of collected photons.

2256 In figure 7.4a we show the distribution of the ratio $\frac{Q_{rec}}{Q_{true}}$ for central events ($R < 4m$) and different
 2257 values of γ_{qnl} . In figure 7.4a, we show the mean of this distribution as a function of the energy. We
 2258 also present the effective α_{qnl} for each value of γ_{qnl} . We observe that using the event-wise QNL is
 2259 equivalent to the mean behavior of using channel-wise QNL.

2260 When using channel-wise non-linearity, we need to simulate a number of PE per LPMT, the process
 2261 can be quite tedious if we want a realistic simulation. So in this study we are only using event-wise
 2262 non-linearity to make the process simpler. This event-wise non-linearity will be characterized by α_{qnl}
 2263 in this work.

2264 7.2 Approach

2265 In this section, we detail the testing procedure for each of our tools.



(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 $R < 4\text{m}$ 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 $R < 4\text{m}$ to not be affected by the non-uniformity. The dots represent the mean of the distributions in figure 7.4a 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 [27].

FIGURE 7.4

2266 7.2.1 Data production

2267 IBD spectra

2268 The first step involves generating the data on which our tools will be tested. In this study we
 2269 use Monte-Carlo toys. For each toy we generate a $\bar{\nu}_e$ energy spectrum from the Taishan, Yangjiang
 2270 and Dayabay nuclear power plants, the reactors used as source for the NMO analysis. The reactors
 2271 parameters comes from JUNO official database, which shared among all physics analysis, the JUNO
 2272 common inputs. This provides the initial spectra for the LPMT and SPMT systems. We then incorpo-
 2273 rate physic effects such as the LS non-linearity etc... (more details in section 7.3.1). Finally, we apply
 2274 the reconstruction resolution for each system to their respective spectra, resulting in the final LPMT
 2275 and SPMT spectra.

2276 We will study the effect of exposure on our methods at different threshold: 100 days, 1 year, 2 year
 2277 and finally 6 years which is the nominal data taking period for the NMO analysis.

2278 These spectra are generated for different QNL, $\alpha_{qnl} = 0$ (no spectrum distortion) and for $\alpha_{qnl} \in$
 2279 $\{0.01, 0.005, 0.003, 0.002, 0.001\}$. As a reminder, the calibration guarantees a residual event-wise non-
 2280 linearity of $\alpha_{qnl} \leq 0.003$ [27].

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:

$$\Delta_i = h_{L,i} - h_{S,i} \quad (7.3)$$

$$U = AVA^T \quad (7.4)$$

$$\chi_{spe}^2 = \vec{\Delta}^T U^{-1} \vec{\Delta} \quad (7.5)$$

Where $h_{L,i}$ and $h_{S,i}$ are the contents of the i th 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:

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

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_{spe}^2 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_{spe}^2 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

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

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_{31}^2 and θ_{13} , the test is only done on the oscillation parameters θ_{12} and Δm_{21}^2 . We can thus produce the individual chi square χ_{ind}^2

$$\Delta_\lambda = \lambda_L - \lambda_S \quad (7.8)$$

$$\vec{\Delta} = [\Delta_{\theta_{12}} \Delta_{\Delta m_{21}^2}] \quad (7.9)$$

$$U = AVA^T \quad (7.10)$$

$$\chi_{ind}^2 = \vec{\Delta}^T U^{-1} \vec{\Delta} \quad (7.11)$$

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_{21}^2 . V here is the 4×4 covariance matrix between the parameters

$\sin^2(2\theta_{12})$	Δm_{21}^2	Δm_{31}^2	$\sin^2(2\theta_{13})$
$0.851^{+0.020}_{-0.018}$	$7.53 \pm 0.18 \times 10^{-5} \text{ eV}^2$	$2.5283 \pm 0.034 \times 10^{-3} \text{ eV}^2$	0.8523 ± 0.00268

TABLE 7.2 – Nominal PDG2020 value [14]. All value are reported assuming Normal Ordering.

2300 $\theta_{12,L}$, $\Delta m_{21,L}^2$, $\theta_{12,S}$ and $\Delta m_{21,S}^2$. A is the transformation matrix that allow us to compute the covariance
2301 matrix de $\vec{\Delta}$ from V following

$$A_{ij} = \frac{\partial \Delta_i}{\partial j}; i \in \{\theta_{12}, \Delta m_{21}^2\}; j \in \{\theta_{12,L}, \Delta m_{21,L}^2, \theta_{12,S}, \Delta m_{21,S}^2\} \quad (7.12)$$

2302 Same as described above, by comparing the distribution of this χ^2_{ind} when $\alpha_{qnl} = 0$ and $\alpha_{qnl} \neq 0$ we
2303 can compute the power of this test in term of p-values.

2304 7.2.3 Joint fit

2305 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 at the same time. We minimize a χ^2_{joint} defined over the two spectra, the LPMT and SPMT one

$$\Delta_i = D_i - T_i \quad (7.13)$$

$$\chi^2_{joint} = \vec{\Delta}^T V^{-1} \vec{\Delta} \quad (7.14)$$

2306 where D_i is the content of the i th bin measured, from the data, and T_i is the theoretical number of
2307 event in this bin. V is the covariance matrix of our spectrum.

2308 T is the fitted function and depend on multiple parameters

- 2309 — The oscillation parameters θ_{12} , Δm_{21}^2 , θ_{13} and Δm_{31}^2 . Those parameters can be free, have a pull
2310 term or be fixed during the fit.
- 2311 — We take into account in the data production the matter effect and parametrize it by the pa-
2312 rameter ρ , the effective rock density between the reactors and the experiment. Same as the
2313 oscillation parameters, this parameter can be free, pulled or fixed.
- 2314 — The exposure of the considered data which is just a normalization factor in front of the theo-
2315 retical spectrum. This parameter is fixed at the start of the fit.

2316 In the standard joint fit, the free parameters are $\sin^2(2\theta_{12})$, Δm_{21}^2 and Δm_{31}^2 . $\sin^2(2\theta_{13})$ is fixed to the
2317 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.

2318 Both of the LPMT and SPMT systems are sensitive to θ_{12} and Δm_{21}^2 , thus these parameters are totally
2319 free and start at the PDG nominal value. Only the LPMT system is sensitive to Δm_{31}^2 , we let it
2320 free so we can observe the effect of the deformation on it while the solar parameters θ_{12} , Δm_{21}^2 are
2321 constrained by the SPMT system. To prevent Δm_{31}^2 to take absurd value, we add a pull term using
2322 the PDG nominal value and errors. The PDG nominal values used in this study can be found in table
2323 7.2.

$$\chi^2_{joint} = \vec{\Delta}^T V^{-1} \vec{\Delta} + \frac{\Delta m_{31}^2 - \Delta m_{31,PDG}^2}{\sigma_{31,PDG}} \quad (7.15)$$

2324 θ_{13} is the parameter on which we are least accurate. It's fixed to nominal value to prevent degeneracy
2325 (table 7.2).

2326 The covariance matrix is produced from a correlation matrix C

$$V_{ij} = \sigma_i \sigma_j C_{ij} \quad (7.16)$$

2327 where σ_i is the uncertainty on the number of event in the i th bin. We consider in this study that the
2328 content of each bin follow a Poisson statistic, thus the uncertainty is $\sigma_i = \sqrt{N_i}$ where N_i is the content
2329 of the i th bin. The bin content used for the uncertainty can come from two sources: the data and the
2330 theoretical spectra $\sigma_i = \sqrt{D_i}$ (Pearson test) and $\sigma_i = \sqrt{T_i}$ (Neyman test). Precedent studies have
2331 show that both Pearson and Neyman tests show bias at low statistic, we thus use the Pearson V test
2332 where

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

2333 and the covariance matrix V is computed using the data spectrum for the uncertainty.

2334 The estimation of the covariance is crucial in this study as the strength of this test rely on the sys-
2335 tematic and statistical correlations between the LPMT and SPMT spectrum. The generation methods
2336 and results of this matrix is detailed in section 7.5.2.

2337 Delta joint fit

2338 Using the same structure we define a second joint fit, the Delta joint fit where, in addition to every-
2339 thing that was discussed above, we add two other parameters $\delta\theta_{12}$ and $\delta\Delta m_{21}^2$ and split the theoretical
2340 $T(\theta_{12}, \Delta m_{21}^2, \dots)$ spectrum in two

$$\begin{aligned} T_{LPMT} &\equiv T(\theta_{12} + \delta\theta_{12}, \Delta m_{21}^2 + \delta\Delta m_{21}^2, \dots) \\ T_{SPMT} &\equiv T(\theta_{12}, \Delta m_{21}^2, \dots) \end{aligned} \quad (7.18)$$

2341 If the there is no additional distortion between the LPMT and the SPMT spectra, the fit should
2342 converge to $\delta\theta_{12} = \delta\Delta m_{21}^2 = 0$. By observing the dispersion of those parameters we can define
2343 the probability $P(\alpha_{qnl} = 0 | (\delta\theta_{12}, \delta\Delta m_{21}^2))$ and use the median value of $(\delta\theta_{12}, \delta\Delta m_{21}^2)$ when $\alpha_{qnl} \neq 0$
2344 to define a p-value.

2345 The last test we explore in this thesis is to fit the same spectrum with the Standard Joint fit, that
2346 we consider as the hypothesis without distortion H_0 , and the Delta Joint fit, designated as the H_1
2347 hypothesis. By looking at the dispersion of $\chi^2_{joint, H_0} - \chi^2_{joint, H_1}$ we can extract a sensitivity to potential
2348 distortion.

2349 7.2.4 Data and theoretical spectrum generation

2350 To implement the joint fit, we have technically two data spectra and two theoretical spectra. The data
2351 in this study are produced using an IBD generator *IBD gen*, see section 7.3.1. The theoretical spectrum
2352 are produced the same way as data spectrum but with much higher statistics, 10^7 events to compare
2353 with the $\approx 10^5$ events for 6 years statistic. The two spectrum, that we get as a collection of events,
2354 are binned in two histograms from 0.8 to 9 MeV of reconstructed energy with bins of 0.02 MeV each,
2355 resulting in 410 bins per spectrum. An illustration of the theoretical spectrum can be found in figure
2356 7.5. The low number of events in the tail of the spectrum can cause instability due to the low statistic,
2357 we thus cut the spectrum at 7.5 MeV / 335 bins for the fit.

2358 All the IBD spectra presented and used in this study are produced assuming Normal Ordering using
2359 the PDG nominal value [14] for the oscillation parameters. Those values are reported in table 7.2.

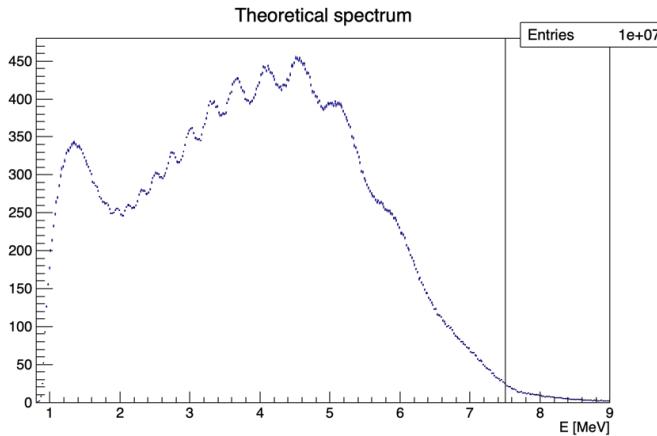


FIGURE 7.5 – Theoretical LPMT spectrum at nominal oscillation values binned using 410 bins from 0.8 to 9 MeV. It is rescaled to 6 years statistic. The black line represent the 335 bin cut

2360 7.2.5 Limitations

2361 In this work we are only working considering the statistical errors. We can ignore systematic effects,
 2362 such as effects that would affect the neutrino spectrum or the background spectrum, as they are
 2363 entirely correlated between the two systems. The details of those systematic effects can be found in
 2364 [3].

2365 Most of our results assume decorrelated detection effects between the SPMT and LPMT systems.
 2366 Their respective reconstruction effects are simulated using simple gaussian drawing on the resolution,
 2367 independently from the event position. This approach was used in previous sensitivity and
 2368 precision studies [3, 82]. The potential effect of those reconstruction effects and a first attempt to take
 2369 them into account are explored in section 7.5.2.

2370 Even if the goal of this work is to propose deformation agnostic tools, the QNL we use in this study is
 2371 simplistic as we consider event-wise, position uniform deformation. We show in figure 7.4a and 7.4b
 2372 that event-wise QNL is equivalent to the mean behaviour of channel-wise QNL but a more complete
 2373 study would simulate channel-wise deformation for each event.

2374 7.3 Fit software

2375 In this section, I describe the ft framework that was used in this study. The software is composed
 2376 of two parts as illustrated in figure 7.6: A standalone part composed of ROOT [84] macros, and the
 2377 Avenue framework.

2378 The Avenue framework is responsible for the spectrum and configuration reading, transforming
 2379 the raw collection of events into spectra, managing the physics effect such as the oscillation and
 2380 computing and minimizing the χ^2 with the help of the RooFit library. The macros are invoking, if
 2381 necessary, the Avenue framework and are the entry point for fitting, generating the necessary inputs
 2382 quantity such as the spectra and correlation matrix, analysing the fit results and managing jobs for
 2383 distributed computing.

2384 In this section we will focus on the IBD generator in section 7.3.1 and the fit macro in itself in section
 2385 7.3.2.

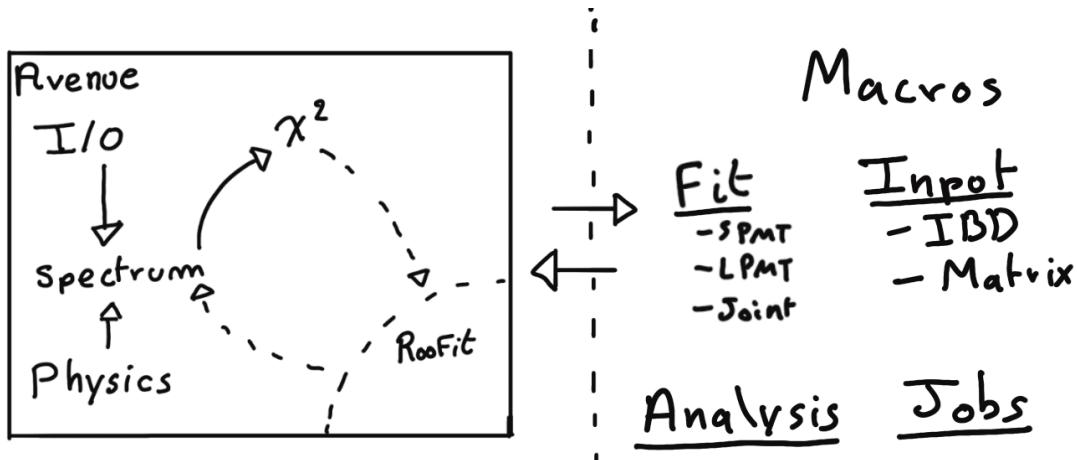


FIGURE 7.6 – Schematic description of the fit framework

2386 7.3.1 IBD generator

2387 The IBD generator is a standalone generator used to produce oscillated and non oscillated spectra
 2388 as the one seen by the JUNO experiment. It takes as inputs physics parameters and a collection
 2389 of histograms, values and function provided by JUNO to its analysis groups, referred as the JUNO
 2390 common inputs.

2391 Options allow to enable or disable effects such as non-uniformity and non-linearity. It finally take as
 2392 an argument the number of events to generate N_{evt} . Optionally, we generate an effective number of
 2393 events N by drawing in a Poisson distribution of mean N_{evt} .

2394 Then for each event we

- 2395 1. Choose randomly, following the reactor power fraction, the source reactor of the neutrino.
- 2396 2. Generate a random interaction position in the detector following a uniform distribution over
 2397 the detector volume.
- 2398 3. Draw a random neutrino energy E_ν from the expected neutrino emission spectrum of every
 2399 reactor. This spectrum is computed by:
 - 2400 (a) Computing the power spectrum of each isotopes ^{235}U , ^{238}U , ^{239}Pu , ^{241}Pu using the Huber-
 2401 Mueller model [5, 8].
 - 2402 (b) Summing the contribution of each isotopes following the respective fission fraction [0.58,
 2403 0.07, 0.30, 0.05] as reported in [85].
 - 2404 (c) The power of each reactor is then adjusted by their distances from the detector, the detector
 2405 efficiency and their mean duty cycle (11 or 12 month).
 - 2406 (d) The total spectrum is then finally adjusted by taking into account the correction of the Day
 2407 Bay bump [11], adjustment due to spent nuclear fuel and due to the non-equilibrium.
- 2408 4. (Optional) Compute the survival probability due to oscillation at nominal oscillation param-
 2409 eters value. If the neutrino does not survive, the event is rejected and the algorithm restart
 2410 from step (1).
- 2411 5. Compute the emitted positron energy E_{pos} from the mass difference. If the neutrino does not
 2412 have enough energy reject the event and start from step (1).
- 2413 6. Compute the deposited energy E_{dep} by incrementing E_{pos} by 511 keV to account for the positron
 2414 annihilation. We do not consider cases where some of the energy leak outside of the detector
 2415 (positron or annihilation gammas escaping the CD).

- 2416 7. Correct the deposited energy with the expected event-wise non-linearity from [27] to obtain
 2417 the visible energy E_{vis} .
- 2418 8. (Optional) Add a custom non-linearity as described in section 7.1.2. This non linearity is
 2419 characterized by α_{qnl} to obtain E_α .
- 2420 9. Finally, using the expected resolution of the LPMT and SPMT systems, provided in the JUNO
 2421 common inputs, we draw from a gaussian characterized by those resolution the reconstructed
 2422 energy E_{rec} or E_{lpmt} and E_{spmt} for each systems. The resolutions are provided as ABC param-
 2423 eters using

$$\frac{\sigma E_{vis}}{E_{vis}} = \sqrt{\left(\frac{A}{\sqrt{E_{vis}}}\right)^2 + B^2 + \left(\frac{C}{E_{vis}}\right)^2} \quad (7.19)$$

2424 where A is the term driven by the Poisson statistics of the total number of detected photoelec-
 2425 trons, C is dominated by the PMT dark noise, and B is dominated by the detector's spatial
 2426 non-uniformity. The relative and absolute resolutions of the LPMT and SPMT systems are
 2427 illustrated in figure 7.7.

2428 The events are stored as n-tuples and are not yet binned at the end of the generator.

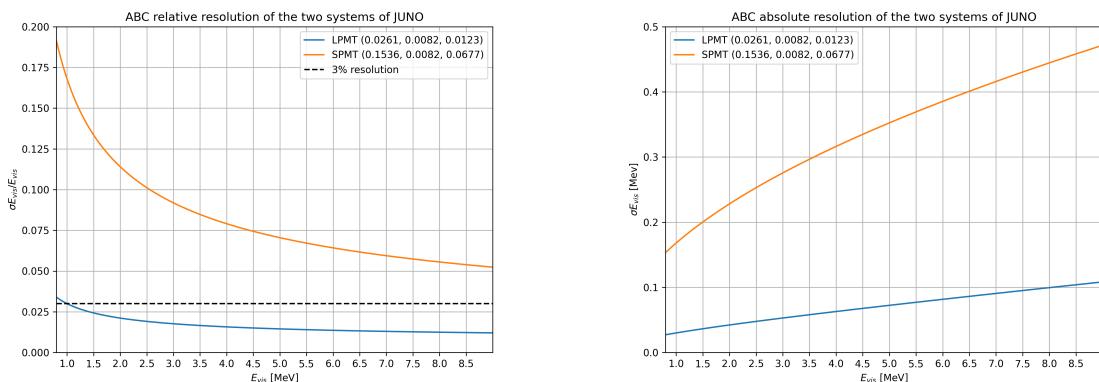


FIGURE 7.7 – Relative (On the left) and absolute (On the right) resolutions of the LPMT and SPMT systems used in this study. The number in parenthesis are the parameter A, B and C respectively for each systems.

2429 7.3.2 Fit

2430 The fit macro is the core of this fitting procedure. This macro is responsible for loading the fit
 2431 configuration and setup the Avenue framework. Using Avenue, it will setup the data files, theoretical
 2432 spectrum, choose the binning, χ^2 , etc... It also have the possibility to generate toys on the fly based
 2433 on the theoretical spectrum. Given this theoretical spectrum we can randomize the bin content either
 2434 by:

- 2435 1. Drawing the bin content in a Poisson distribution with the bin content as parameter.
 2436 2. Drawing the bin content in a Gaussian distribution with the bin content as mean and variance.
 2437 The bin content is then rounded to the nearest integer.
 2438 3. Drawing the bin difference following a given covariance matrix using the Choleski decomposi-
 2439 tion. This matrix is at least the statistical covariance matrix but can also contain systematic

uncertainties.

$$V = LL^T \quad (7.20)$$

$$\mathbf{R} \sim \mathcal{N}(0, 1) \quad (7.21)$$

$$\tilde{\mathbf{h}} = \lceil \mathbf{h} + L\mathbf{R} \rceil \quad (7.22)$$

$$(7.23)$$

where V is covariance matrix used to produce the fluctuations, \mathbf{R} is drawn in a multinomial 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_a 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 [3].

- 2473 — The standard joint fit with an independent covariance matrix (*Indep Standard joint*), meaning
 2474 that the covariance between the LPMT and SPMT spectra is 0, believe to have twice as much
 2475 informations, and thus believe to have a grater precision than the individual fits.
 2476 — The standard joint (*Standard joint*) fit with a correlated covariance matrix has errors similar to
 2477 the LPMT individual fit as the LPMT drive the precision on θ_{13} and Δm_{31}^2 and that the LPMT
 2478 as SPMT are expected to have close precision on θ_{12} and Δm_{21}^2 .
 2479 — The delta joint (*Delta joint*) fit with covariance matrix have the same resolution as the standard
 2480 joint fit. The supplementary parameter $\delta\theta_{12}$ and $\delta\Delta m_{21}^2$ should not bring supplementary
 2481 precision.

2482 The italicized name are the name used in the results reports to identify each fit. We also look into the
 2483 *Indep Delta joint*, which is the Delta Joint fit but the covariance between the LPMT and SPMT spectra
 2484 is 0, and the *Weighted* results where

$$\frac{1}{\sigma_{\text{Weighted}}^2} = \frac{1}{\sigma_{\text{LPMT}}^2} + \frac{1}{\sigma_{\text{SPMT}}^2} \quad (7.24)$$

2485 We expect the weighted resolution to be similar to the *Indep Standard joint* as, in both of those test, we
 2486 do not consider the correlation between the SPMT and LPMT results.

2487 Asimov studies

2488 We ran Asimov studies on the tests presented above on the updated framework, the results are
 2489 reported in table 7.3. All those test are ran considering statistics error only, 6 years exposure with
 2490 all backgrounds, Pearson χ^2 (covariance is estimated using data spectrum) and θ_{13} fixed to nominal
 2491 value. For the SPMT fit Δm_{31}^2 is fixed at nominal value as the SPMT system is net expected to be
 2492 sensitive to this parameter.

	Δm_{21}^2 error	$\delta\Delta m_{21}^2$ error	θ_{12} error	$\delta\theta_{12}$ error	Δm_{31}^2 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_{21}^2 and Δm_{31}^2						
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_{31}^2 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 7.3 – 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

2493 In every cases presented above, the fit converges to the parameters nominal value thus only the
 2494 errors are presented.

2495 We observe, as expected, that $\sigma_{\text{Weighted}} \approx \sigma_{\text{Indep Standard joint}}$ with the exception of $\sigma\theta_{12}$. This could
 2496 from the slight difference in statistic between the SPMT and LPMT spectra. Indeed, due to a larger
 2497 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 $\sigma_{\text{Standard joint}} \approx \sigma_{\text{LPMT}}$. 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 7.8 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 7.9.

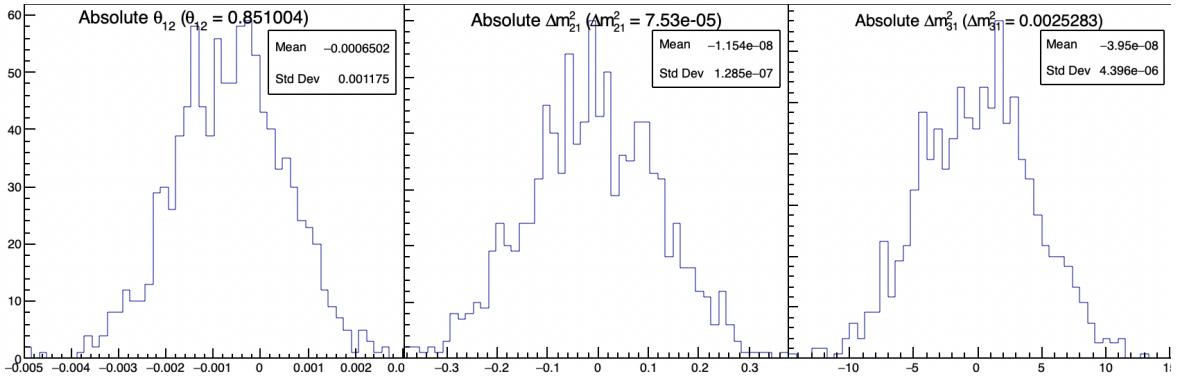


FIGURE 7.8 – Distribution of BFP - nominal value for 1000 toy Standard joint fit. 6 years exposure, all background, Pearson χ^2 , θ_{13} fixed.

When the supplementary parameters are introduced in the Delta Joint fit, the fit is stable as shown in the results figure 7.10. 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.

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

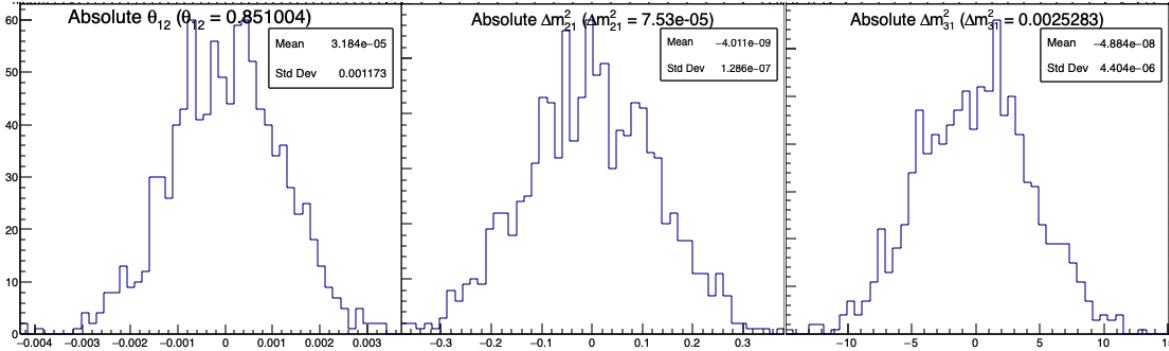


FIGURE 7.9 – Distribution of BFP - nominal value for 1000 toy Standard joint fit. 6 years exposure, all background, PearsonV χ^2 , θ_{13} fixed.

2528 calibration ($\alpha_{qnl} = 0.3\%$ [27]). The background had to be removed as JUNO provide them already
 2529 smeared, thus the introduction of supplementary QNL is not trivial, the resolution being dependent
 2530 of E_{vis} which is affected by the QNL. We use a covariance matrix assuming no QNL. The effect of this
 2531 QNL on the spectrum is illustrated in figure 7.11. In table 7.4 we report the results of the different
 2532 scenarios.

Mean (std dev)	$\theta_{12} [10^{-3}]$	$\Delta m^2_{21} [10^{-7}\text{eV}^2]$	$\Delta m^2_{31} [10^{-6}\text{eV}^2]$	$\delta\theta_{12} [10^{-3}]$	$\delta\Delta m^2_{21} [10^{-7}\text{eV}^2]$
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 7.4 – 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.

2533 The results in table 7.4 are subtracted from their nominal value, themselves reported in table 7.2.
 2534 We clearly see the bias induced by $\alpha_{qnl} = 1\%$ when comparing the SPMT and LPMT results. The
 2535 Indep Standard is, as expected, the mean value between the SPMT and LPMT: the fit having no
 2536 informations about the correlation between the spectrum think it have two uncorrelated experiments
 2537 thus report an in between value. When introducing the relationship between the LPMT and SPMT
 2538 spectra in the Standard fit, the joint fit cannot find a clean minima, it thus converge to a completely
 2539 incorrect value.

2540 Introducing the δ without the correlation in Delta Indep remove the bias and converge to the SPMT
 2541 minima, the δ absorbing the deformation of the LPMT spectra.

2542 Finally, with the δ and the covariance matrix, θ_{12} is unbiased, $\delta\theta_{12}$ absorbing the deformation. $\delta\Delta m^2_{21}$
 2543 is still heavily biased, even more than LPMT only, for the same reason than the Standard fit: the
 2544 correlation make it difficult to converge to the nominal value.

2545 Overall Δm^2_{31} bias is unchanged as the SPMT spectrum bring no information about the parameter.
 2546 The δ are significant, naively up to 7.46σ for $\delta\theta_{12}$ in the Delta fit.

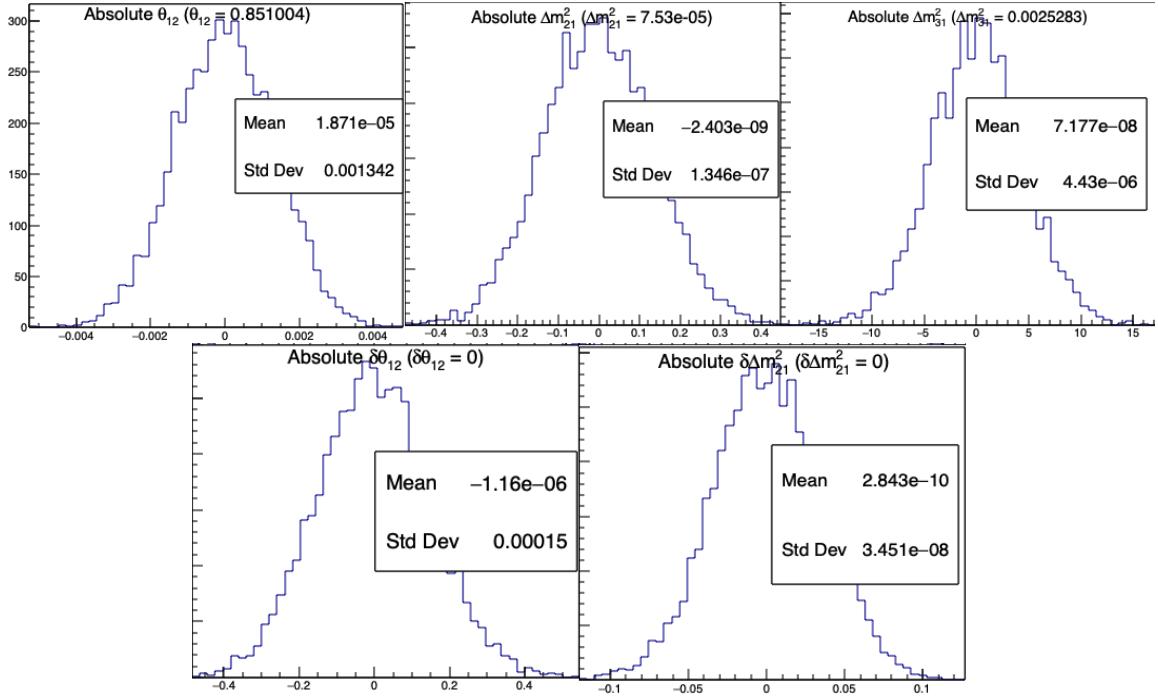


FIGURE 7.10 – Distribution of BFP - nominal value for 5000 toy Delta joint fit. 6 years exposure, all background, PearsonV χ^2 , θ_{13} fixed.

7.5.2 Covariance matrix

The covariance matrix between the LPMT and SPMT spectra is at the heart of this study as it was already mentioned in section 7.2 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 spectrum 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 [14] and considering an extended spectrum of 820 bins following the binning scheme introduced in 7.2.4, the first 410 for the LPMT and the last 410, we consider

- $\boldsymbol{\theta} = (\theta_0, \dots, \theta_n); n = 820$ the content of the spectrum bins.
- $\boldsymbol{\eta}(\boldsymbol{\theta}) = (\eta_0(\boldsymbol{\theta}), \dots, \eta_m(\boldsymbol{\theta})); m = 820$ the set of smearing functions representing the PMT resolutions.

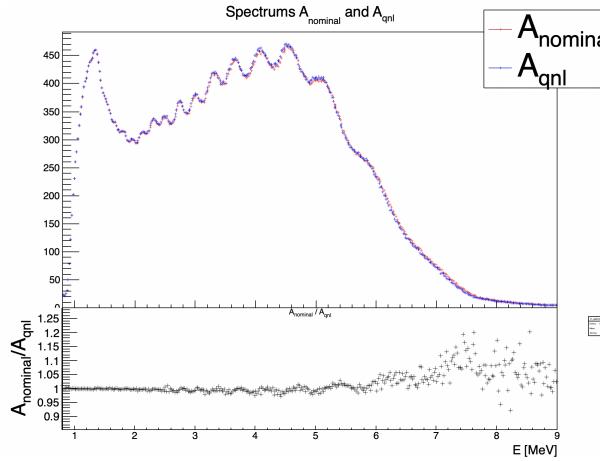


FIGURE 7.11 – **Top:** Theoretical spectrum without QNL (in red) and with $\alpha_{qnl} = 1\%$ (in blue). **Bottom:** Ratio between the theoretical spectrum with and without QNL.

²⁵⁶⁷ η_m can thus be defined as

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

²⁵⁶⁸ where $G(i, \sigma(E_i))(j)$ is the smearing function defined as

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

²⁵⁶⁹ 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 ²⁵⁷⁰ the i th bin respectively.

²⁵⁷¹ We can then construct the transfer matrix A as

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

²⁵⁷² and then compute the first part of our covariance matrix

$$U = AVA^T \quad (7.28)$$

²⁵⁷³ where V is the uncorrelated covariance matrix simply defined, under the assumption of poissonian ²⁵⁷⁴ statistic for the bin content,

$$V_{ij} = \sqrt{\theta_i \theta_j} \quad (7.29)$$

²⁵⁷⁵ Now we just need to consider the uncertainty on the smearing $\sigma \eta_i$, considering no uncertainty on ²⁵⁶⁶ the unsmeared spectrum. From Eq. 7.25, the $G(i, j) \equiv G(i, \sigma(E_i))(j)$ are considered independent ²⁵⁷⁷ from each other $\forall i, j$. This mean that this covariance matrix is diagonal, we only need $\sigma G(i, j)$. We ²⁵⁷⁸ 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 ²⁵⁷⁹ i . This is a number, we thus assume poissonian statistic so that $\sigma[G(i, j)\theta_j] = \sqrt{G(i, j)\theta_j}$.
- 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

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

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

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

By summing the two covariance matrix, we can extract a correlation matrix presented in figure 7.12. 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.

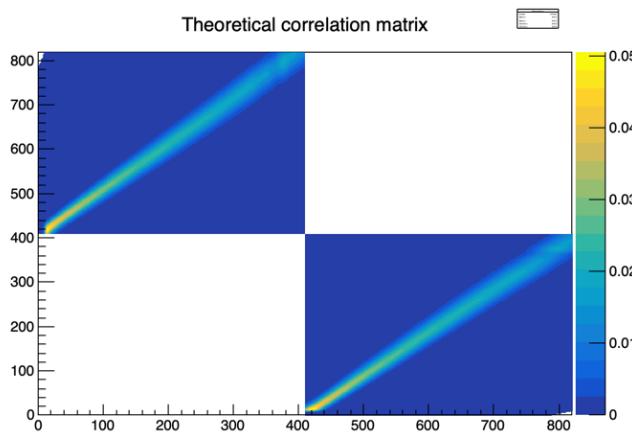


FIGURE 7.12 – Theoretical correlation matrix between the LPMT spectrum (bins 0-409) and the SPMT spectrum (410-819). The diagonal has been set to 0 (it was 1) for readability purpose.

Empiric method

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

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

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 [86] 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 7.13 the upper left corner of the estimated correlation matrix for different configurations of N and M in the limit of 10^7 total event. We see in figure 7.13a that if the toy number N is too low, the statistical noise make the correlation pattern almost completely disappear, in figure 7.13b we see clearly the same correlation pattern as in the theoretical matrix in figure 7.12. On the final matrix in figure 7.13c 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.

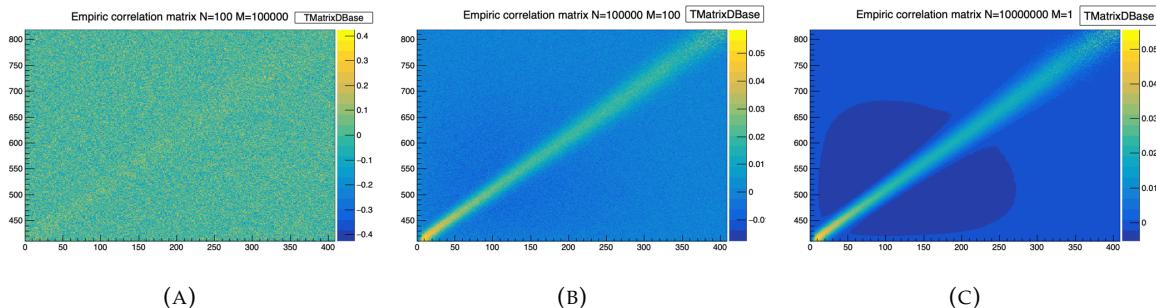


FIGURE 7.13 – Upper left corner of the estimated correlation matrix between the LPMT and SPMT spectrum for different configuration of N toy with different number of M events per toy

The difference between the element of the theoretical and the empiric correlation matrices are presented in figure 7.14a. We see 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 7.14b, 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.

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.

These empirical correlation matrices still pose an issue: These matrices needs to be invertible for χ^2 calculation. The framework use the Cholesky decomposition [87] 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.

2622 Empirical correlation matrix from fully simulated event

2623 The last study on the correlation matrix between the LPMT and SPMT spectrum consists in simulating and reconstructing full events in the official JUNO simulation framework and computing an empirical matrix based on those events.

2626 The core of the idea is that the LPMT and SPMT reconstruction errors is bound to be correlated due to systematic effects. The first and most obvious one, for example, is energy escaping from the central detector. If the positron, or one of the two annihilation gamma, escape from the detector, less energy is deposited thus both of the systems will reconstruct a lower energy that was actually deposited.

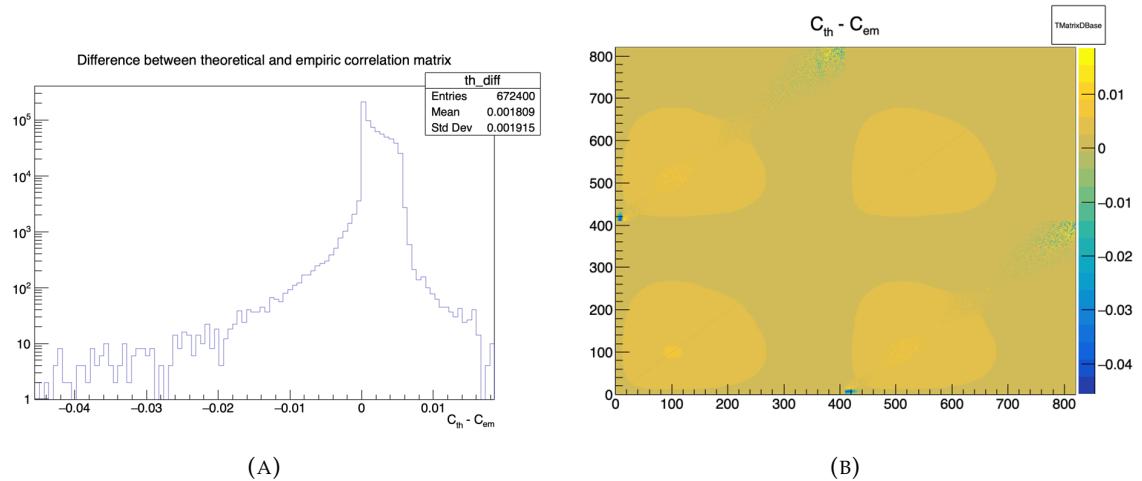


FIGURE 7.14 – Difference between the element of the theoretical and empiric correlation matrix

2630 On a more subtle scale, the randomness in the production of scintillation photons is common for the
 2631 two systems, if the liquid scintillator produces fewer scintillation photons for an event, both systems
 2632 are likely to underestimate the energy.

2633 We study those effects by computing from a dataset of IBD events, uniformly distributed in the CD,
 2634 the correlation between the reconstruction errors on the energy

$$\text{Corr}(E_{lpmt} - E_{dep}, E_{spmt} - E_{dep}) \quad (7.34)$$

2635 where E_{lpmt} and E_{spmt} are the reconstructed energies from both systems and E_{dep} is the deposited
 2636 energy in the detector.

2637 With this observable, the bias difference between the two reconstructions at fixed R and E is irrele-
 2638 vant. However, since we compute the correlation in E and R^3 bins, we need to account for the
 2639 potential spurious relationship between the errors and their respective biases. If the bias is small
 2640 relative to the resolution, it can be ignored; but if the bias variation is on the same order of magnitude
 2641 as the error, it may introduce false correlations. For this reason, based on the CNN results shown in
 2642 figure 4.8, we restrict our analysis to the $1 < E_{dep} < 9$ MeV range.

2643 The results of those correlations are presented in figure 7.15 for the single energy and radius depen-
 2644 dency and figure 7.16 for the dual energy and radius dependency.

2645 We see correlation increase with respect to the energy which can be attributed to the signal over dark
 2646 noise ratio. As more PMTs hits come from the signal, the reconstruction becomes more signal related.
 2647 Regarding the R^3 distribution, we see almost no dependency until the total reflection area. After this
 2648 point the correlation rises as the event are exposed to the optical effect of the total reflection area.

2649 By looking at figure 7.16, we can see that the rising in correlation with respect to the energy is mostly
 2650 due to the radius dependency.

2651 The exploitation of those correlations in the fit and the data production, without generating and
 2652 reconstructing full spectra from SNIPER, is a bit more complicated. As seen in section 7.3.1, we
 2653 characterize the resolution of both systems by the ABC parameters. The correlation shown here take
 2654 into account all of the ABC terms, as they are the complete correlation between the two systems, but
 2655 the generation and the modeling this correlation needs to be very well understood as, as seen before,
 2656 the mass ordering and parameters measurements are very sensitive to even small correlations.

2657 We consider the binned approach that we used here, knowing that the CNN reconstruction was

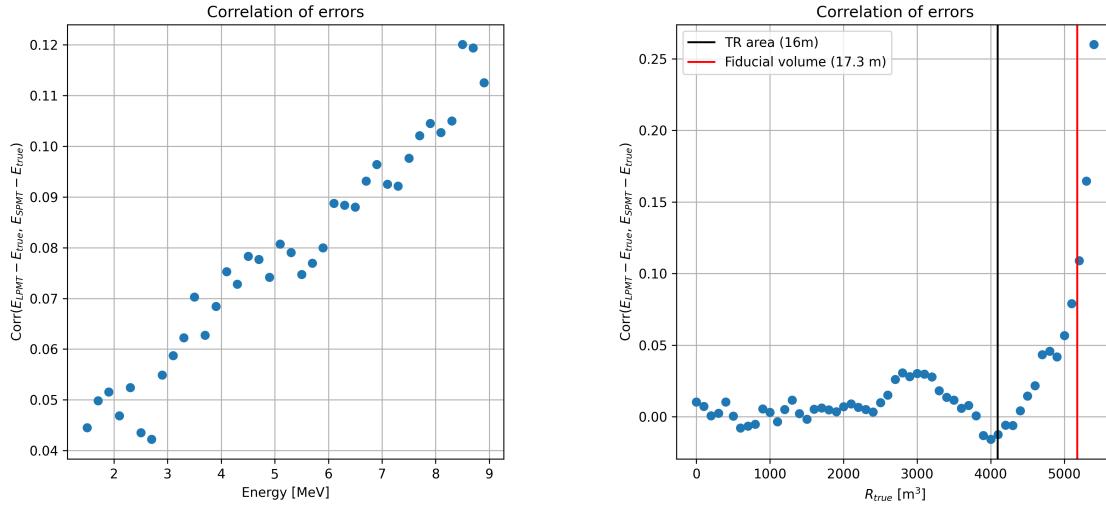


FIGURE 7.15 – Correlation on the reconstruction error between the LPMT and SPMT system as a function of (On the left) the energy, (On the right) the radius. The SPMT reconstruction comes from the NN presented in Chapter 4 and the LPMT reconstruction comes from OMILREC presented in section 2.7. To prevent effect due to the CNN bad reconstruction, we select the event with $1 < E_{dep} < 9$ MeV.

2658 deemed efficient but flawed, to be insufficient for the complete study of those effects on the fit.

2659 7.5.3 Statistical tests

2660 In this part, I present the results of the statistical tests presented in section 7.2.

2661 Test χ_{spe}^2

2662 The χ_{spe}^2 is a chi-square representing the compatibility between the LPMT ans SPMT spectra under
2663 constraints of the correlation matrix between the two.

$$\chi_{spe}^2 = \Delta h V_{spe} \Delta h^T; \Delta h = \{(h_0^L - h_0^S), \dots, (h_n^L - h_n^S)\} \quad (7.35)$$

2664 where h_i^L and h_i^S are the contents of the i th bins of the LPMT and SPMT spectra. For details about the
2665 calculation of V_{spe} , see section 7.2.

2666 The results for different exposures can be found in figure 7.17. To give an idea of the significance of
2667 this test, we provide the median p-value for each test $\alpha_{qnl} \neq 0$. As expected, the power of this test
2668 rises as the exposure does. We see significant discrimination at 6 years for $\alpha_{qnl} \geq 0.3\%$ where the
2669 p-value for $\alpha_{qnl} = 3\%$ is 0.005 ± 0.0022 .

2670 This test relies solely on the estimated covariance matrix between the two spectra, requiring no
2671 fitting. As a result, it is a very lightweight test that can still provide valuable indications of potential
2672 unknown distortions between the two spectra.

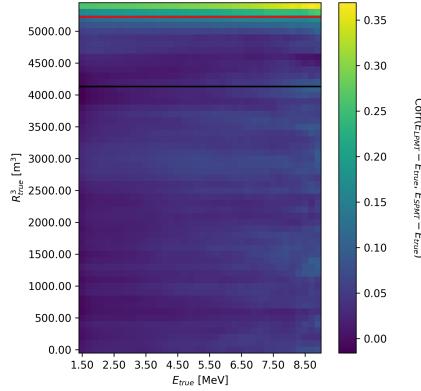


FIGURE 7.16 – Correlation on the reconstruction error between the LPMT and SPMT system as a function of the energy and the radius. The SPMT reconstruction comes from the NN presented in Chapter 4 and the LPMT reconstruction comes from OMILREC presented in section 2.7. To prevent effect due to the CNN bad reconstruction, we select the event with $1 < E_{dep} < 9$ MeV.

2673 **Test χ_{ind}^2**

2674 The χ_{ind}^2 is the chi-square that represent the agreement between the measured oscillation parameters
 2675 θ_{12} and Δm_{21}^2 . This test is defined as

$$\chi_{ind}^2 = \Delta\lambda V_{ind} \Delta\lambda^T; \Delta\lambda = \{\theta_{12}^L - \theta_{12}^S, (\Delta m_{21}^2)^L - (\Delta m_{21}^2)^S\} \quad (7.36)$$

2676 where θ_{12}^L and $(\Delta m_{21}^2)^L$ are the oscillation parameters measured by the LPMT system. Same for θ_{12}^S
 2677 and $(\Delta m_{21}^2)^S$ for the SPMT system. We use V_{ind} computed for $\alpha_{qnl} = 0$. For more details about the
 2678 calculation of V_{ind} see section 7.2.

2679 The results are presented in figure 7.18. This test does not require any joint fit or covariance matrix
 2680 estimation between the two spectrum, it just need the estimated covariance matrix between the four
 2681 parameters. We see that the p-value are much less significant than the other tests, this is because this
 2682 test possess much less information about the relation between the LPMT and SPMT systems.

2683 This test is the most straightforward as it require only the fit of the two spectra and the estimation
 2684 of the parameters covariances, but is also the less powerful with a p value for $\alpha_{qnl} = 0.3\%$ of 0.09 ± 0.009 .

2686 **δ parameters significance**

2687 This test involves observing the values of the δ parameters in the Delta Joint fit and comparing them
 2688 tho their dispersion in the case where $\alpha_{qnl} = 0$. The results are shown in figures 7.19 and 7.20.

2689 We can see that the $\delta\Delta m_{21}^2$ has a very small discriminative power (figure 7.20) even at 6 years
 2690 exposure with a p-value of 0.34 ± 0.01 for $\alpha_{qnl} = 0.3\%$. On the other hand $\delta\theta_{12}$ (figure 7.19) has
 2691 much more discriminative power with a p-value for $\alpha_{qnl} = 0.3\%$ of 0.025 ± 0.005 . This test with a
 2692 single joint fit seems to be still less powerful than the χ_{spe}^2 . This can be explained as this method
 2693 only get information through the oscillation parameters θ_{12} and Δm_{21}^2 missing potential informations
 2694 contained in Δm_{31}^2 .

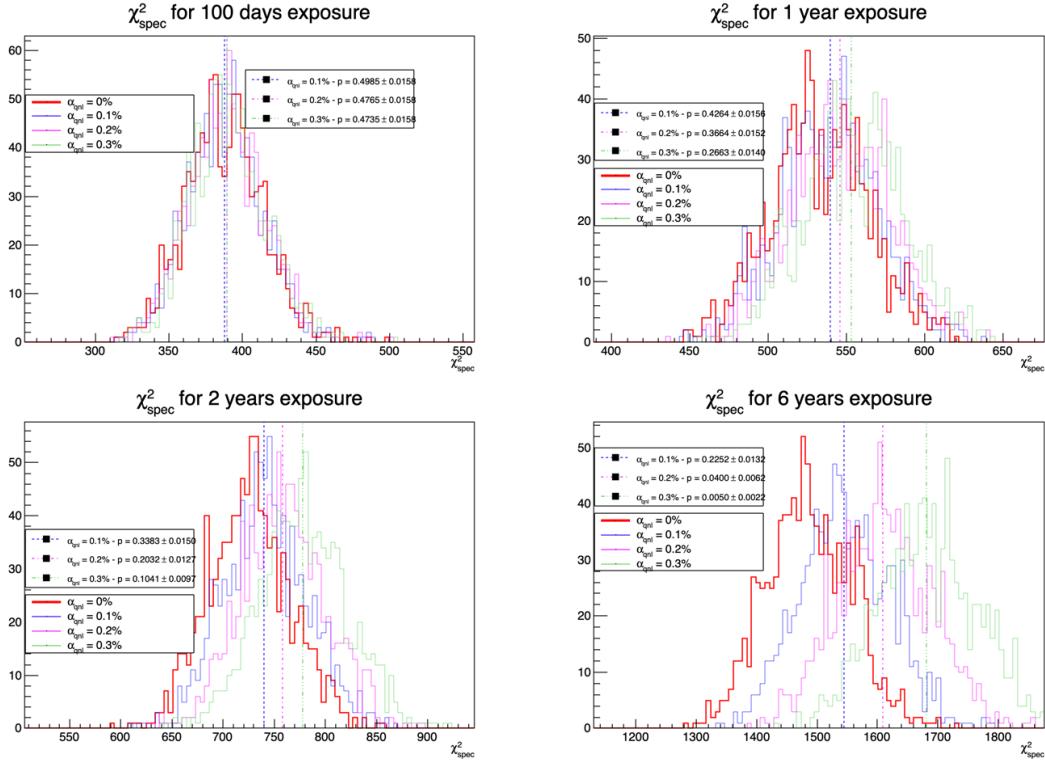


FIGURE 7.17 – Distribution of the χ^2_{spe} for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

2695 Hypothesis test

2696 In this last test we consider the two fit Standard Joint and Delta Joint as two hypothesis. The first
 2697 one, Standard Joint, is the H_0 hypothesis: we do not need supplementary parameters to describe the
 2698 energy spectrum. The second one, Delta Joint, is the H_1 hypothesis: we do need those supplementary
 2699 δ parameters to, if not correctly, approach the energy spectrum. If the δ parameter are unnecessary
 2700 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
 2701 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
 2702 exposures and α_{qnl} . The results are presented in figure 7.21.

2703 This test is the most complex, requiring two fit and the covariance matrix between the LPMT and
 2704 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
 2705 0.01 ± 0.003 .

2706 As explained in section 7.2.4, the spectra used for the fit are cut at 335 bins / 7.5 MeV to prevent
 2707 instability, while in χ^2_{spe} we use full 410 bins spectra. The χ^2_{spe} thus has more informations that the
 2708 hypothesis test leading to this difference in power.

2709 7.6 Conclusion and perspectives

2710 In this chapter, we present the development of a fit framework that allows us to fit multiple spectra
 2711 simultaneously. We also introduce a set of tools that enable us to detect potential distortions in one of
 2712 the two spectra. As an illustration of the capability of these tools, we use supplementary event-wise

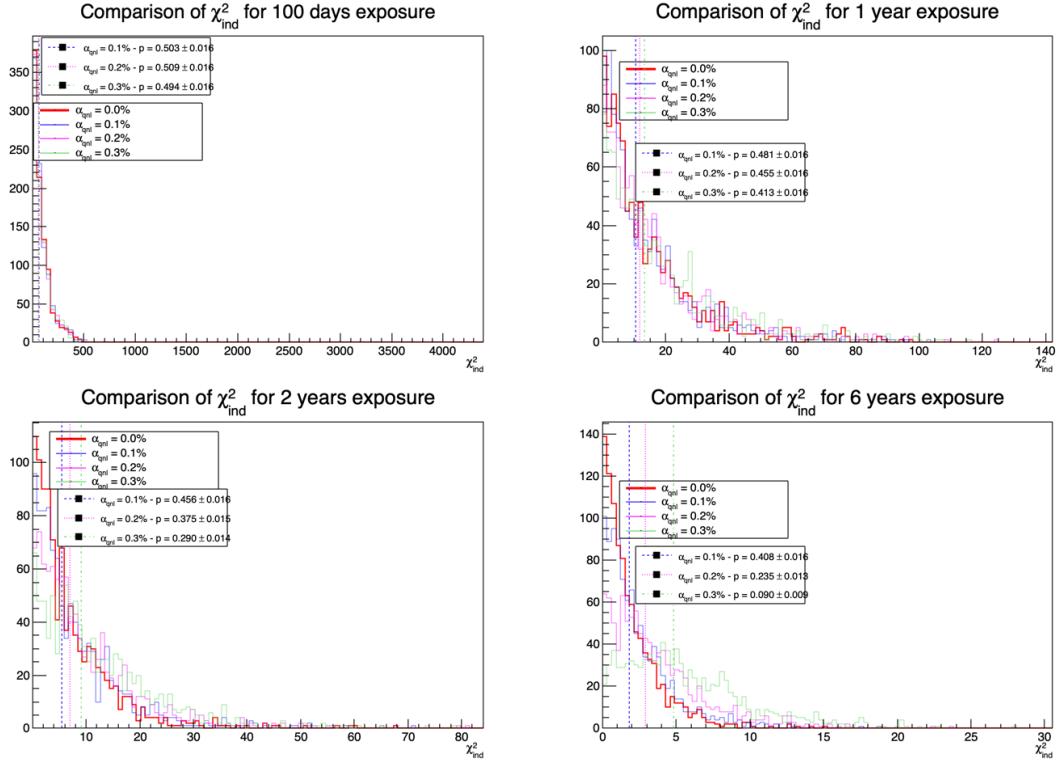


FIGURE 7.18 – Distribution of the χ^2_{Ind} for 1000 toys for different exposures. The dashed lines represent the median of the distributions and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

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.

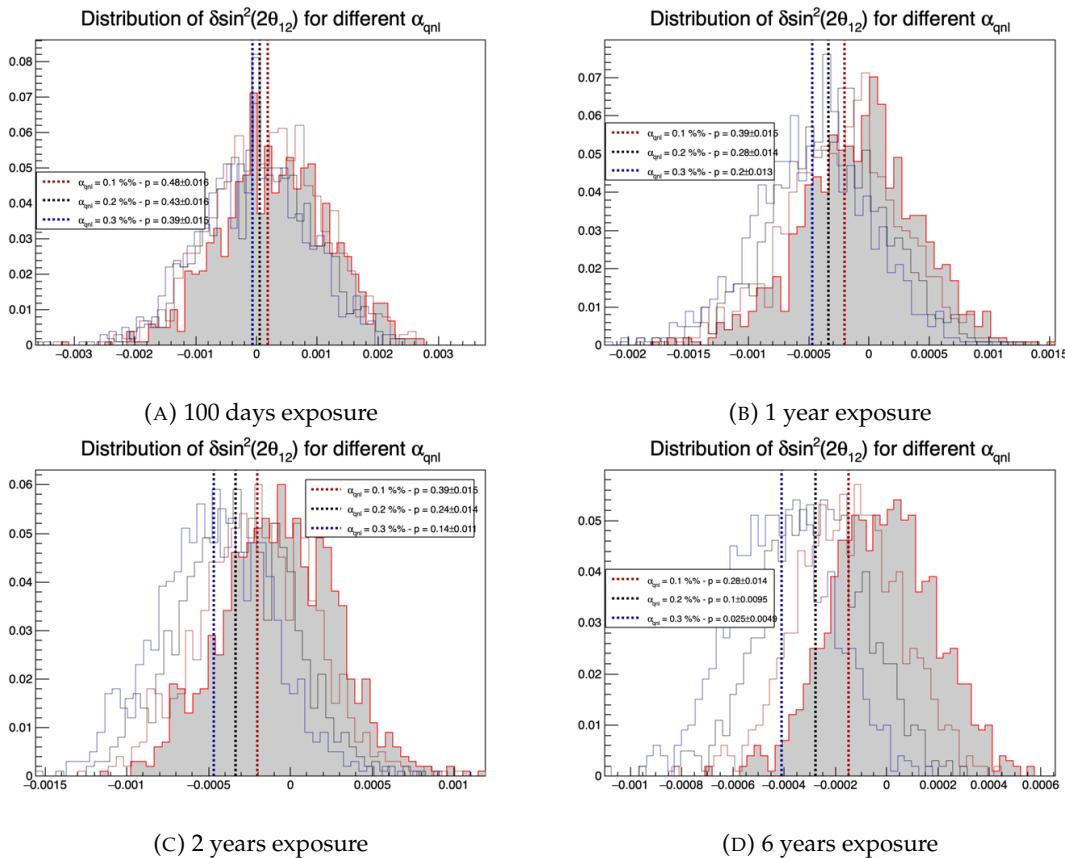


FIGURE 7.19 – Distribution of the $\delta \sin^2(2\theta_{12})$ for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

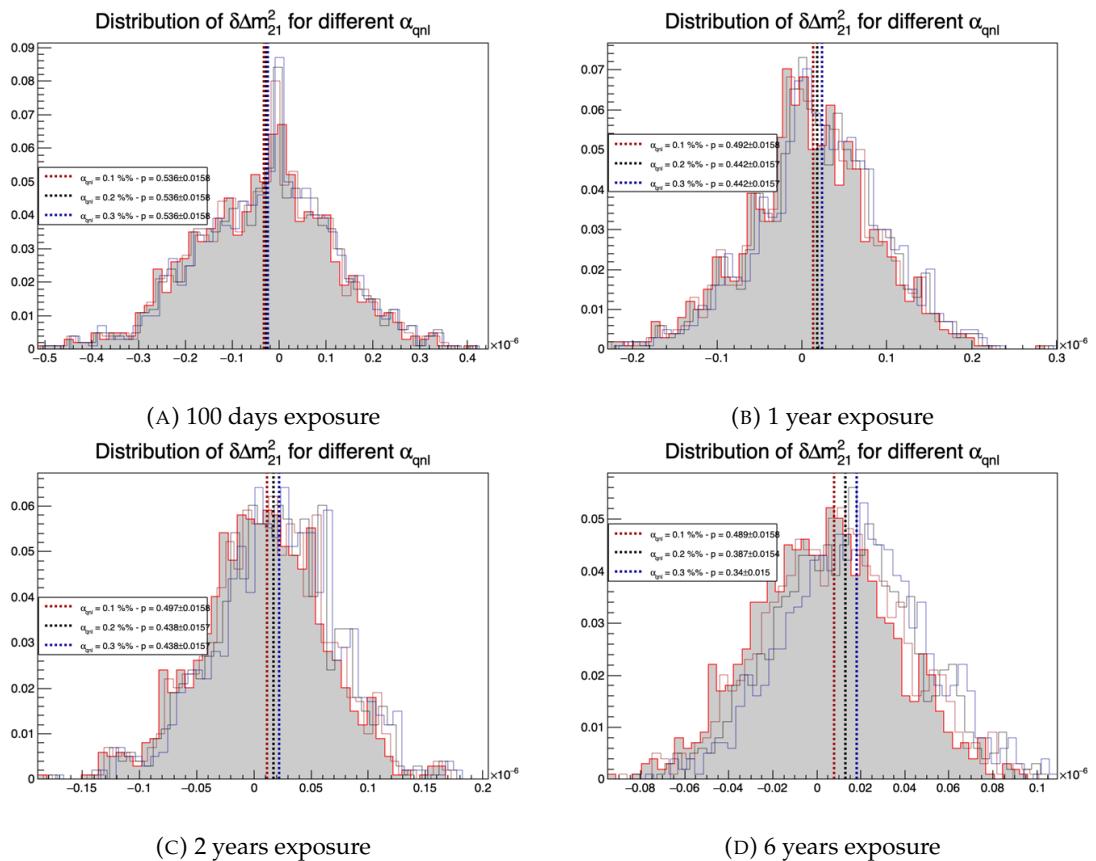


FIGURE 7.20 – Distribution of the $\delta\Delta m_{21}^2$ for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

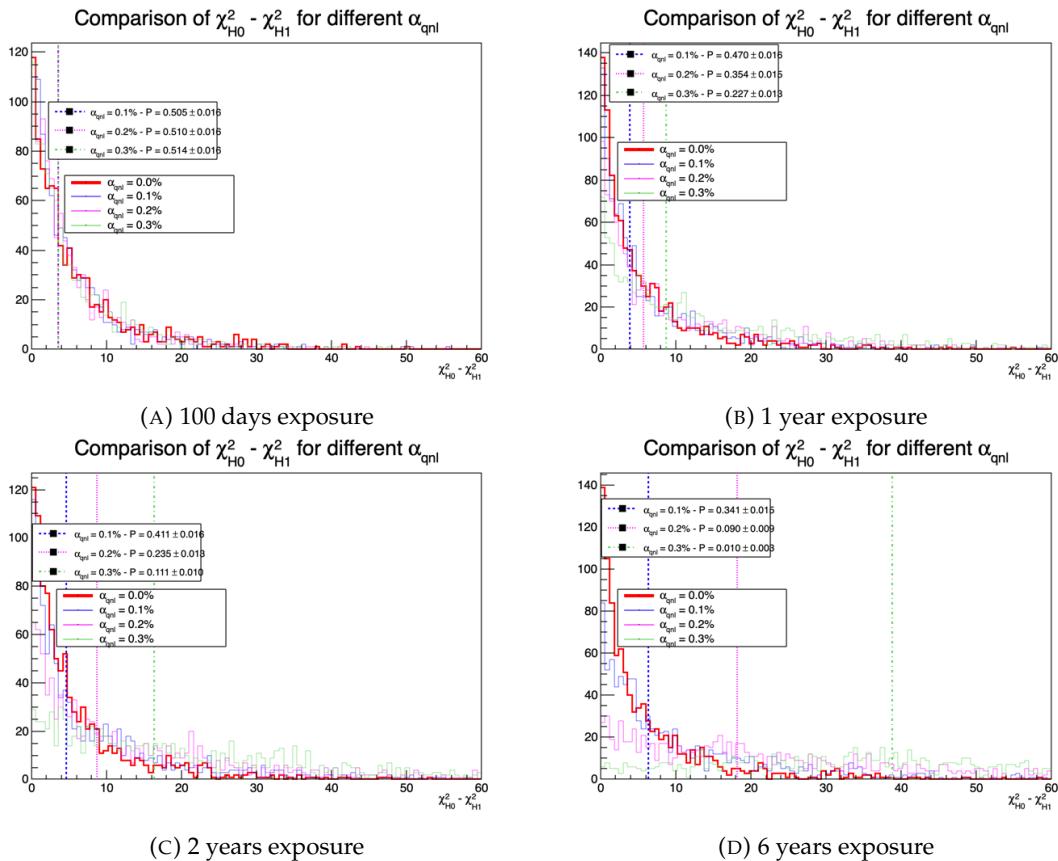


FIGURE 7.21 – Distribution of $\chi^2_{H_0} - \chi^2_{H_1}$ for 1000 toys for different exposure. The dashed line represent the median of the distribution and the p-value are the percentage of the $\alpha_{qnl} = 0$ distribution that are greater than those medians.

²⁷²² **Chapter 8**

²⁷²³ **Conclusion**

²⁷²⁴ **Appendix A**

²⁷²⁵ **Calculation of optimal α for estimator
combination**

²⁷²⁷ This annex the details of the determination of the optimal α for estimator combination presented in
²⁷²⁸ section 4.3.2.

²⁷²⁹ As a reminder, the combined estimator $\hat{\theta}$ of X is defined as

$$\hat{\theta}(X) = \alpha\theta_N + (1 - \alpha)\theta_C; \alpha \in [0; 1] \quad (\text{A.1})$$

²⁷³⁰ where θ_N and θ_C are both estimator of X .

²⁷³¹ **A.1 Unbiased estimator**

For the unbiased estimator, it is straight-forward. We search α such as $E[\hat{\theta}] = X$

$$E[\hat{\theta}] = E[\alpha\theta_N + (1 - \alpha)\theta_C] \quad (\text{A.2})$$

$$= E[\alpha\theta_N] + E[(1 - \alpha)\theta_C] \quad (\text{A.3})$$

$$= \alpha E[\theta_N] + (1 - \alpha)E[\theta_C] \quad (\text{A.4})$$

$$= \alpha(\mu_N + X) + (1 - \alpha)(\mu_C + X) \quad (\text{A.5})$$

$$X = \alpha\mu_N + \mu_C - \alpha\mu_C + X \quad (\text{A.6})$$

$$0 = \alpha(\mu_N - \mu_C) + \mu_C \quad (\text{A.7})$$

$$(A.8)$$

$$\Rightarrow \alpha = \frac{\mu_C}{\mu_C - \mu_N} \quad (\text{A.9})$$

²⁷³² **A.2 Optimal variance estimator**

The α for this estimator is a bit more tricky. By expanding the variance we get

$$\text{Var}[\hat{\theta}] = \text{Var}[\alpha\theta_N + (1 - \alpha)\theta_C] \quad (\text{A.10})$$

$$= \text{Var}[\alpha\theta_N] + \text{Var}[(1 - \alpha)\theta_C] + \text{Cov}[\alpha(1 - \alpha)\theta_N\theta_C] \quad (\text{A.11})$$

$$= \alpha^2\sigma_N^2 + (1 - \alpha)^2\sigma_C^2 + 2\alpha(1 - \alpha)\sigma_N\sigma_C\rho_{NC} \quad (\text{A.12})$$

²⁷³³ where, as a reminder, ρ_{NC} is the correlation factor between θ_C and θ_N .

Now we try to find the minima of $\text{Var}[\hat{\theta}]$ with respect to α . For this we evaluate the derivative

$$\frac{d}{d\alpha} \text{Var}[\hat{\theta}] = 2\alpha\sigma_N^2 - 2(1-\alpha)\sigma_C^2 + 2\sigma_N\sigma_C\rho_{NC}(1-2\alpha) \quad (\text{A.13})$$

$$= 2\alpha(\sigma_N^2 + \sigma_C^2 - 2\sigma_N\sigma_C\rho_{NC}) - 2\sigma_C^2 + 2\sigma_N\sigma_C\rho_{NC} \quad (\text{A.14})$$

then find the minima and maxima of this derivative by evaluating

$$\frac{d}{d\alpha} \text{Var}[\hat{\theta}] = 0 \quad (\text{A.15})$$

$$2\alpha(\sigma_N^2 + \sigma_C^2 - 2\sigma_N\sigma_C\rho_{NC}) - 2\sigma_C^2 + 2\sigma_N\sigma_C\rho_{NC} = 0 \quad (\text{A.16})$$

$$2\alpha(\sigma_N^2 + \sigma_C^2 - 2\sigma_N\sigma_C\rho_{NC}) = 2\sigma_C^2 - 2\sigma_N\sigma_C\rho_{NC} \quad (\text{A.17})$$

$$\alpha = \frac{\sigma_C^2 - \sigma_N\sigma_C\rho_{NC}}{\sigma_N^2 + \sigma_C^2 - 2\sigma_N\sigma_C\rho_{NC}} \quad (\text{A.18})$$

2734 This equation shows only one solution which is a minima. From Eq. A.18 arise two singularities:

2735 — $\sigma_N = \sigma_C = 0$. This is not a problem because as physicists we never measure with an absolute

2736 precision, neither us or our detectors are perfect.

2737 — $\sigma_N = \sigma_C$ and $\rho_{CN} = 1$. In this case θ_C and θ_N are the same estimator in term of variance thus

2738 any value for α yield the same result: an estimator with the same variance as the original ones.

²⁷³⁹ **Appendix B**

²⁷⁴⁰ **Charge spherical harmonics analysis**

²⁷⁴¹ When looking at JUNO events we can clearly see some pattern in the charge repartition based on
²⁷⁴² the event radius as illustrated in figure B.4. When dealing with identifying features and pattern on a
²⁷⁴³ spherical plane, the astrophysics community have been using, with success, the spherical harmonic
²⁷⁴⁴ decomposition. The principle is similar to a frequency analysis via Fourier transform. It comes to
²⁷⁴⁵ saying that a function $f(r, \theta, \phi)$, here our charge repartition of the spherical plane constructed by our
²⁷⁴⁶ PMTs, can be expressed

$$f(r, \theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l a_l^m r^l Y_l^m(\theta, \phi) \quad (\text{B.1})$$

²⁷⁴⁷ 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
²⁷⁴⁸ degree l and order m and P_l^m their associated Legendre Polynomials. Those harmonics are illustrated
²⁷⁴⁹ in figure B.1. By reducing the problem to the unit sphere $r = 1$, we get rid of the term r^l . The Healpix
²⁷⁵⁰ library [73] offer function to efficiently find the a_l^m factor from a given Healpix map.

²⁷⁵¹ For the above decomposition, we will define the *Power* of an harmonic as

$$S_{ff}(l) = \frac{1}{2l+1} \sum_{m=-l}^l |a_l^m|^2 \quad (\text{B.2})$$

²⁷⁵² and the *Relative Power* as:

$$P_l^h = \frac{S_{ff}(l)}{\sum_l S_{ff}(l)} \quad (\text{B.3})$$

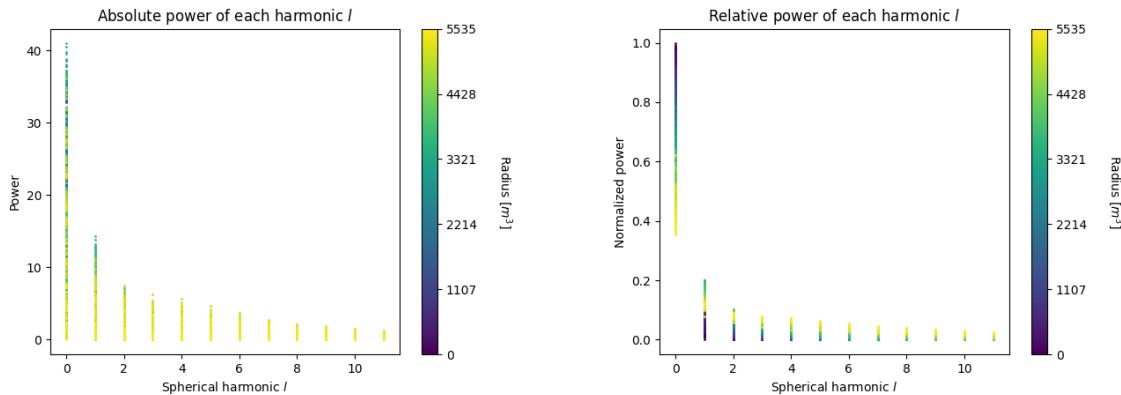
²⁷⁵³ For this study we will use 10k positron events with $E_{kin} \in [0; 9]$ MeV uniformly distributed in the
²⁷⁵⁴ CD from the JUNO official simulation version J23.0.1-rc8.dc1 (released the 7th January 2024). All the
²⁷⁵⁵ event are *calib* level, with simulation of the physics, electronics, digitizations and triggers. We first
²⁷⁵⁶ take a sub-set of 1k events and look at the power and relative power distribution depending on the
²⁷⁵⁷ radius and harmonic degree l . The results are shown in figure B.2. While don't see any pattern in
²⁷⁵⁸ absolute power, it is pretty clear that there is a correlation between the relative power of $l = 0$ and
²⁷⁵⁹ the radius of the event.

²⁷⁶⁰ When applying the same study but dependent on the energy, no clear correlation appear. The results
²⁷⁶¹ for the $l = 0$ harmonic are presented in the figure B.5. Thus, in this study we will focus on the radial
²⁷⁶² dependency of the relative power of each harmonic.

²⁷⁶³ In figures B.6 and B.7 are presented the distribution of the relative power of each harmonic for $l \in$
²⁷⁶⁴ $[0, 11]$. The relation between the radius and the relative power become even more clear, especially
²⁷⁶⁵ for the first harmonics $l \in [0, 4]$. After that for $l > 4$ their relative power is close to 0 for central event,
²⁷⁶⁶ thus loosing power. It also interesting to note the change of behavior in the TR area, clearly visible
²⁷⁶⁷ for $l = 1$ and $l = 2$.

$l:$		$P_\ell^m(\cos \theta) \cos(m\varphi)$	$P_\ell^{ m }(\cos \theta) \sin(m \varphi)$	
0	s			
1	p			
2	d			
3	f			
4	g			
5	h			
6	i			
m:	6 5 4 3 2 1 0	-1 -2 -3 -4 -5 -6		

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

$$F(R^3) \longmapsto P_l^h \quad (\text{B.4})$$

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

$$F^{-1}(P_l^h) \longmapsto R^3 \quad (\text{B.5})$$

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 $l = 0$.

2779 The minimization algorithm used are the Bent algorithm for $l = 0$ and the Bounded algorithm for
 2780 $l > 0$ provided by the Scipy library [88]. We then do the mean of the reconstructed radius from
 2781 the different harmonics. The reconstruction results are shown in figure B.3. The performance seems
 2782 correct but we see heavy fluctuation in the bias. To really be used as a reconstruction algorithm, the
 2783 method needs to be refined as discussed in the next section.

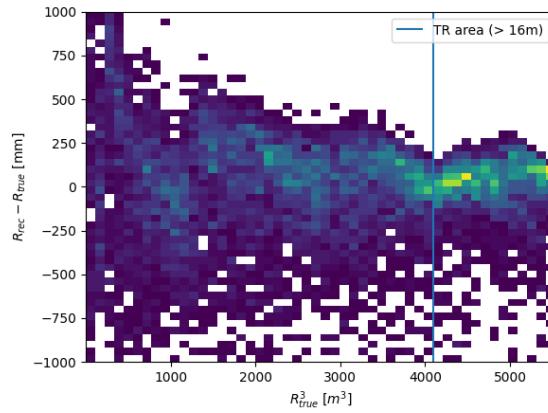


FIGURE B.3 – Error on the reconstructed radius vs the true radius by the harmonic method

Conclusion

2784 We have clearly shown in this analysis the relevance the of relative harmonic power for radius
 2785 reconstruction, and provided an erzats of a reconstruction algorithm. We will not delve further in
 2786 this thesis but if we wanted to refine this algorithm multiple paths can be explored:
 2787

- 2788 — No energy signature in the harmonics: This is surprising that there is no correlation between
 2789 the energy and the amplitude of the harmonics. We know that the energy is heavily correlated
 2790 with the total number of photoelectrons collected, it would be unintuitive that we see no
 2791 relation.
- 2792 — Localization of the event: We shown here the relation between the relative power of the har-
 2793 monic and the radius but don't get any information about the θ and ϕ spherical coordinates.
 2794 This information is probably hidden in the individual power of each order m of the degree l .
 2795 This intuition comes from the figure B.1 where in the higher degree l we see that the order m
 2796 are oriented. Intuitively, the order should be able to indicate a direction where the signal is
 2797 more powerful.
- 2798 — Combination of the degree power: Here we combined the radius reconstructed by the dif-
 2799 ferent degree via a simple mean but we shown in section 4.3.2 and annex A that this is note
 2800 the optimal way to combine estimator. A more refined algorithm probably exist to take into
 2801 account the predicting power of each order.

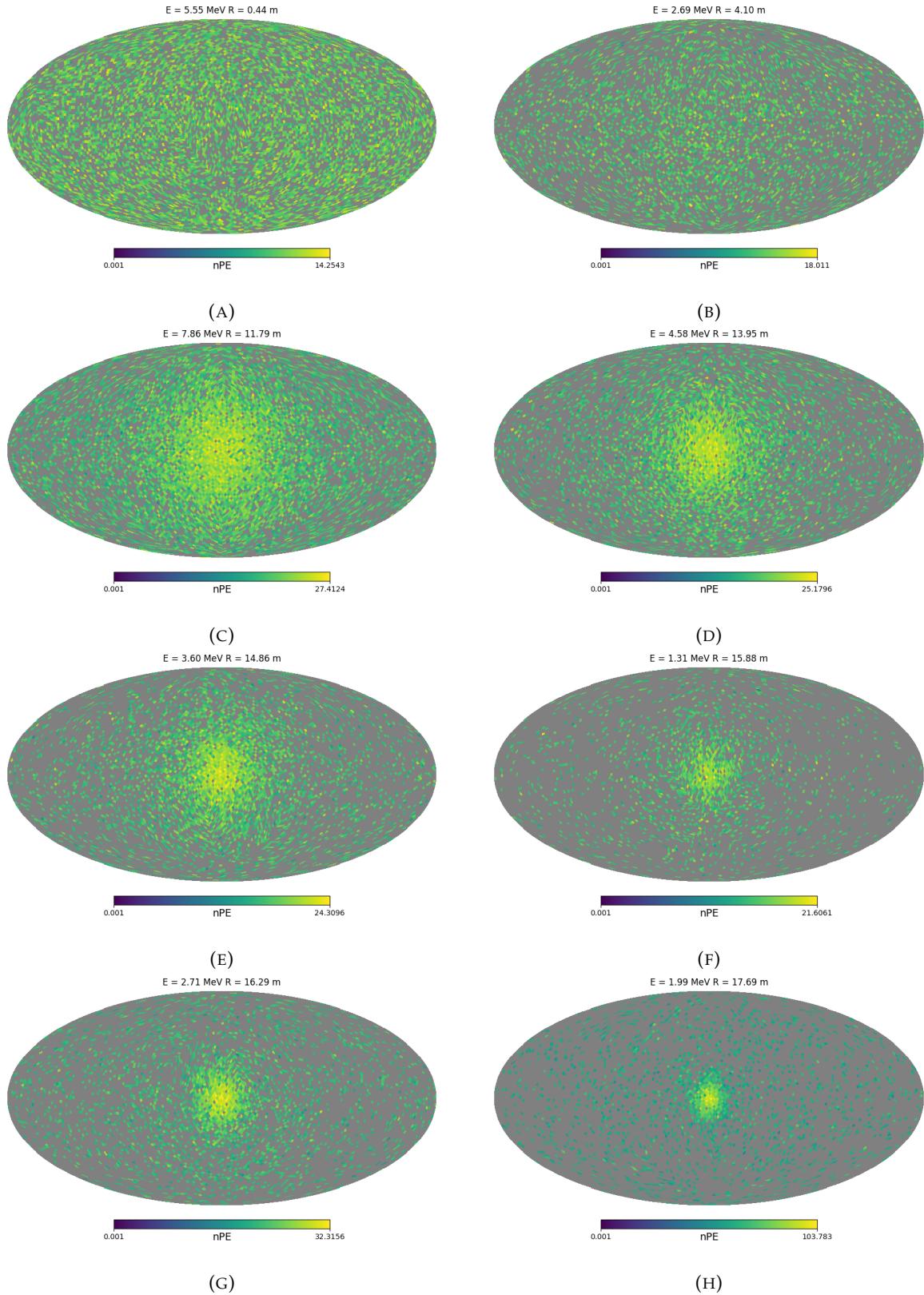


FIGURE B.4 – Charge repartition in JUNO as seen by the Healpix segmentation. Those are Healpix map of order 5 (i.e. 12288 pixels). The color represent the summed charge of the PMTs in each pixels. The color scale is logarithmic. The view have been centered to prevent event deformations.

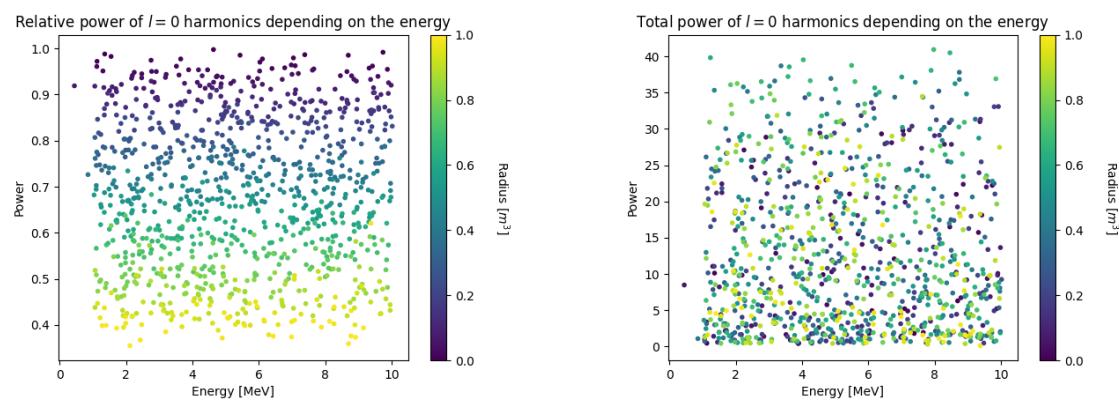


FIGURE B.5 – Scatter plot of the absolute and relative power, respectively on the left and right plot, of the $l = 0$ harmonic. The color indicate the radius of the event.

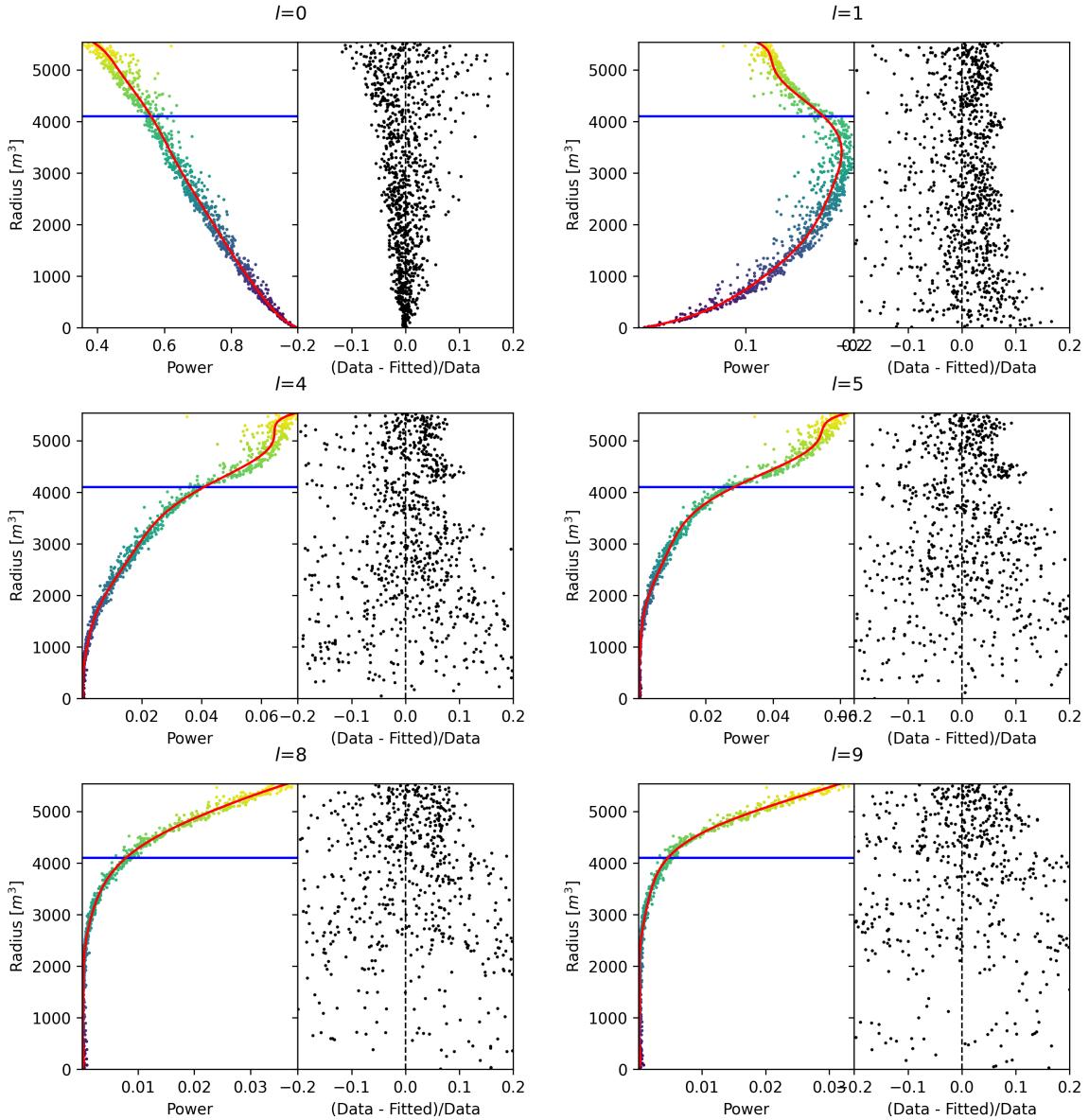


FIGURE B.6 – Plot of the distribution of the relative power of each harmonic dependent on R^3 (on the left). The Total Reflection (TR) area is represented by the horizontal blue line. The distribution are fitted using a 9th degree polynomial (red curve). The relative power error between the distribution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

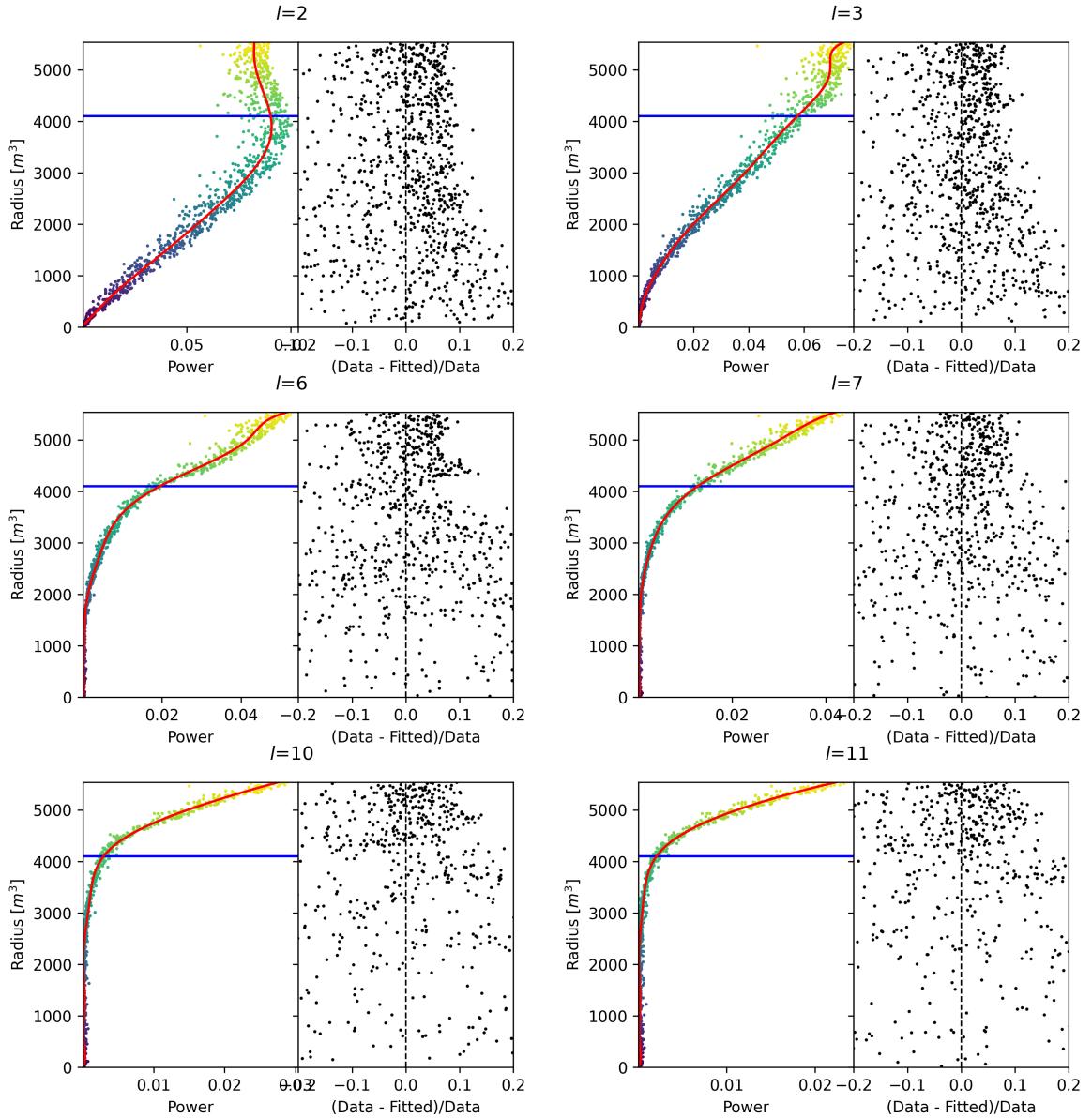


FIGURE B.7 – Plot of the distribution of the relative power of each harmonic dependent on R^3 (on the left). The Total Reflection (TR) area is represented by the horizontal blue line. The distribution are fitted using a 9th degree polynomial (red curve). The relative power error between the distribution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

²⁸⁰² **Appendix C**

²⁸⁰³ **Additional spectrum smearing**

²⁸⁰⁴ In this section we demonstrate that a spectrum S smeared by a gaussian G parametrized by its
²⁸⁰⁵ varianse σ_1^2 can be smeared by a gaussian parametrized by the variance σ_2^2 from the the smeared
²⁸⁰⁶ 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

$$K'(E, \sigma_2) = S(E) \star G(E, \sigma_1) \star G^{-1}(E, \sigma_1) \star G(E, \sigma_2) \quad (\text{C.1})$$

$$= K(E, \sigma_1) \star G^{-1}(E, \sigma_1) \star G(E, \sigma_2) \quad (\text{C.2})$$

²⁸⁰⁷ 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

$$G(E, \sigma_1) \star G^{-1}(E, \sigma_1) = \delta(E) \quad (\text{C.3})$$

$$F[G(E, \sigma_1)](\nu) \times F[G^{-1}(E, \sigma_1)](\nu) = 1 \quad (\text{C.4})$$

²⁸⁰⁸ with $F[G(E, \sigma_1)](\nu)$ the fourier transform of G

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

we have

$$F[G^{-1}(E, \sigma_1)](\nu) = (F[G(E, \sigma_1)](\nu))^{-1} = (e^{-\frac{\sigma_1^2(2\pi)^2}{2}\nu^2})^{-1} \quad (\text{C.6})$$

$$= e^{\frac{\sigma_1^2(2\pi)^2}{2}\nu^2} \quad (\text{C.7})$$

Thus we express

$$F[G^{-1}(E, \sigma_1) \star G(E, \sigma_2)] = e^{\frac{\sigma_1^2(2\pi)^2}{2}\nu^2} \times e^{-\frac{\sigma_2^2(2\pi)^2}{2}\nu^2} \quad (\text{C.8})$$

$$= e^{\frac{(2\pi)^2}{2}(\sigma_1^2 - \sigma_2^2)\nu^2} \quad (\text{C.9})$$

$$= e^{\frac{(2\pi)^2}{2}\Delta\sigma^2\nu^2}; \Delta\sigma^2 = (\sigma_1^2 - \sigma_2^2) \quad (\text{C.10})$$

²⁸⁰⁹ 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

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

²⁸¹⁰ **Appendix D**

²⁸¹¹ **Correction of E_{vis} bias**

²⁸¹² The reconstruction algorithms that are presented in this thesis in Chapters 4 and 5 do not reconstruct
²⁸¹³ the same energy as the classical algorithms presented in section 2.7. Our algorithms reconstruct the
²⁸¹⁴ deposited energy E_{dep} while the classical algorithms reconstruct a visible energy E_{vis} .

To understand this phenomena, let's look at the equation 2.17:

$$\hat{\mu}(r, \theta, \theta_{pmt}, E_{vis}) = \frac{1}{E_{vis}} \frac{1}{M} \sum_i^M \frac{\frac{\bar{Q}_i}{\bar{Q}_i} - \mu_i^D}{DE_i}, \quad \mu_i^D = DNR_i \cdot L$$

²⁸¹⁵ which define the expected N_{pe}/E . This define a linear relation between the number of photoelectrons
²⁸¹⁶ and the energy. However we discussed in sections 2.3.2 and 2.4 that the number of photoelectrons
²⁸¹⁷ collected by the LPMT system do not follow a linear relationship. Thus this visible energy is not
²⁸¹⁸ linear with the deposited energy. This effect is corrected in physics analysis and in Chapter 7 by
²⁸¹⁹ applying the calibrated non-linearity profile the energy spectrum.

²⁸²⁰ When we need to compare our algorithm that reconstruct the deposited energy to the classical
²⁸²¹ algorithms we need to correct this non-linearity. For this we fit the systematic bias of the classical
²⁸²² algorithm using a 5th degree polynomial

$$\frac{E_{dep}}{E_{vis}} = \sum_{i=0}^5 P_i E_{dep}^i \quad (D.1)$$

²⁸²³ The fitted distribution and the corresponding fit is presented in figure D.1. The value fitted for this
²⁸²⁴ correction are presented in table D.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 D.1 – Parameters of the 5th degree polynomial used to correct Omilrec reconstructed energy.

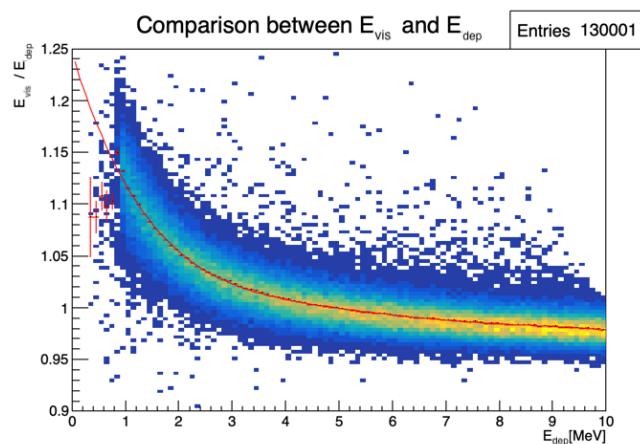


FIGURE D.1 – Comparison between Omilrec reconstructed E_{vis} and the deposited energy E_{dep} . The profile of the distribution E_{vis}/E_{dep} vs E_{dep} is fitted with a 5th degree polynomial.

List of Tables

2826	2.1	Characteristics of the nuclear power plants observed by JUNO.	14
2827	2.2	A summary of precision levels for the oscillation parameters. The reference value (PDG 2020 [14]) is compared with 100 days, 6 years and 20 years of JUNO data taking.	15
2828	2.3	Detectable neutrino signal in JUNO and the expected signal rates and major back- ground sources	16
2829	2.4	List of sources and their process considered for the energy scale calibration	25
2830	2.5	Calibration program of the JUNO experiment	26
2831	2.6	Features used by the BDT for vertex reconstruction	37
2832	2.7	Features used by the BDTE algorithm. <i>pe</i> and <i>ht</i> reference the charge and hit-time distribution respectively and the percentages are the quantiles of those distributions. <i>cht</i> and <i>cc</i> reference the barycenters of hit time and charge respectively	38
2833	4.1	Sets of hyperparameters values considered in this study	58
2834	5.1	Features on the nodes of the graph. All charge are in [nPE], time in [ns] and position in [m]. <i>Q</i> and <i>t</i> are the reconstructed charge and time of the hit PMTs. (<i>x</i> , <i>y</i> , <i>z</i>) is the position of the PMTs and the last parameter represent the type of the PMT. It's 1 for LPMT and -1 for SPMT Q_m and t_m is the set of charges and time of the PMT belonging the mesh <i>m</i> . (X_m , Y_m , Z_m) is the position of the center of the geometric region represented by the mesh <i>m</i> ($\langle X \rangle$, $\langle Y \rangle$, $\langle Z \rangle$) is the position of the charge barycenter, $\sum Q$ the sum of the collected charge in the detector and P_l^h is the relative power of the <i>l</i> th harmonic. See annex B for details.	76
2835	5.2	Features on the edges on the graph. It use the same notation as in table 5.1. $D_{m1 \rightarrow m2}^{-1}$ is the inverse of the distance between the mesh <i>m1</i> and the mesh <i>m2</i> . The features A and B are detailed in section 5.1	77
2836	7.1	The charge fraction in terms of the number of PE collected at the single PMT for the reactor $\bar{\nu}_e$ IBD events. Table taken from [25]	100
2837	7.2	Nominal PDG2020 value [14]. All value are reported assuming Normal Ordering.	104
2838	7.3	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	110
2839	7.4	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_{31}^2 is fixed at nominal value. The χ^2 is PearsonV. The correlation matrix used to fit assume no QNL in the spectrum.	112
2840	D.1	Parameters of the 5th degree polynomial used to correct Omilrec reconstructed energy. .	139

List of Figures

2860	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	12
2862	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_{21}^2). 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.	13
2863		
2864		
2865		
2866		
2867		
2868		
2869		
2870	2.3 Expected visible energy spectrum measured with the LPMT system with (grey) and without (black) backgrounds. The background amount for about 7% of the IBD candidate and are mostly localized below 3 MeV [3]	14
2871		
2872		
2873	2.4	18
2874	a Schematics view of the JUNO detector.	18
2875	b Top down view of the JUNO detector under construction	18
2876	2.5 Schematics of an IBD interaction in the central detector of JUNO	19
2877	2.6 Schematics of the supporting node for the acrylic vessel	20
2878		
2879	2.7 On the left: Quantum efficiency (QE) and emission spectrum of the LAB and the bis-MSB [18]. On the right: Sensitivity of the Hamamatsu LPMT depending on the wavelength of the incident photons [20].	21
2880		
2881	2.8 Schematic of a PMT	21
2882	2.9 The LPMT electronics scheme. It is composed of two part, the <i>wet</i> electronics on the left, located underwater and the <i>dry</i> electronics on the right. They are connected by Ethernet cable for data transmission and a dedicated low impedance cable for power distribution	22
2883		
2884		
2885		
2886	2.10 Schematic of the JUNO SPMT electronic system (left), and exploded view of the main component of the UWB (right)	23
2887		
2888	2.11 The JUNO top tracker	24
2889	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 [27].	25
2890		
2891	a Gamma non-linearity	25
2892	b Boron spectrum	25
2893	c Electron non-linearity	25
2894		
2895	2.13 Overview of the calibration system	26
2896	2.14 Event-level instrumental non-linearity, defined as the ratio of the total measured LPMT charge to the true charge for events at the center of the detector. The solid red line represents event-level non-linearity without the channel-level correction in an extreme hypothetical scenario of 50% non-linearity over 100 PEs for the LPMTs. The dashed blue line represents that after the channel-level correction. The gray band shows the residual uncertainty of 0.3%, after the channel-level correction. Figure taken from [27].	27
2897		
2898		
2899		
2900		
2901		
2902	2.15	28
2903	a Schematic of the TAO satellite detector	28

2904	b	Schematic of the OSIRIS satellite detector	28
2905	2.16	30
2906	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.	30
2907	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	30
2908	2.17	31
2909	a	Δt distribution at different iterations step j	31
2910	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	31
2911	2.18	Bias of the reconstructed radius R (left), θ (middle) and ϕ (right) for multiple energies by the time likelihood algorithm	32
2912	2.19	On the left: Resolution of the reconstructed R as a function of the energy in the TR area ($R^3 > 4000m^3 \equiv R > 16m$) 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	33
2913	2.20	Radial resolution of the different vertex reconstruction algorithms as a function of the energy	34
2914	2.21	34
2915	a	Spherical coordinate system used in JUNO for reconstruction	34
2916	b	Definition of the variables used in the energy reconstruction	34
2917	2.22	36
2918	a	Radial resolutions of the likelihood-based algorithm TMLE, QMLE and QTML	36
2919	b	Energy resolution of QMLE and QTML using different vertex resolutions	36
2920	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	38
2921	2.24	Radial (left) and energy (right) resolutions of different ML algorithms. The results presented here are from [40]. DNN is a deep neural network, BDT is a BDT, ResNet-J and VGG-J are CNN and GNN-J is a GNN.	39
2922	3.1	Example of a BDT that determine if the given object is a duck	42
2923	3.2	Schema of a simple neural network	43
2924	3.3	Illustration of the training lifecycle	45
2925	3.4	46
2926	a	Illustration of SGD falling into a local minima	46
2927	b	Illustration of the Adam momentum allowing it to overcome local minima	46
2928	3.5	Illustration of the SGD optimizer. In blue is the value of the loss function, orange, green and red are the path taken by the optimized parameter during the training for different LR.	47
2929	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.	47
2930	b	Illustration of the SGD optimizer on one parameter θ on the MSE Loss. We see two different behavior: A smooth one (orange and red) when the LR is small enough and a more chaotic one when the LR is too high.	47
2931	3.6	48
2932	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.	48
2933	b	Illustration of a very simple NN	48
2934	3.7	Illustration of the ResNet framework	49

2956 3.8	Illustration of the gradient explosion. Here it can be solved with a lower learning rate but its not always the case.	49
2957 3.9	50
2958 a	Schema of a FCDNN	50
2959 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	50
2960 3.10	Illustration of the effect of a convolution filter. Here we apply a filter with the aim do detect left edges. We see in the resulting image that the left edges of the duck are bright yellow where the right edges are dark blue indicating the contour of the object. The convolution was calculated using [55].	51
2961 3.11	52
2962 a	Example of images in the MNIST dataset	52
2963 b	Schema of the CNN used in Pytorch example to process the MNIST dataset	52
2964 3.12	Illustration of a graph and its tensor representation.	53
2965 3.13	Illustration of the message passing algorithm. The detailed explanation can be found in section 3.2.3	53
2972 4.1	Graphic representation of the VGG-16 architecture, presenting the different kind of layer composing the architecture.	56
2973 4.2	61
2974 a	Spherical coordinate system used in JUNO for reconstruction	61
2975 b	Repartition of SPMTs in the image projection. The color scale is the number of SPMTs per pixel	61
2978 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.	61
2982 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.	62
2986 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.	62
2990 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.	63
2994 4.7	64
2995 a	Distribution of PE/MeV in the J23 Dataset. This distribution is profiled and fitted using equation 4.6	64
2996 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	64
3000 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 [63]. The top part of each plot is the resolution and the bottom part is the bias.	65
3002 a	Resolution and bias of energy reconstruction vs energy	65
3003 b	Resolution and bias of energy reconstruction vs radius	65
3004 c	Resolution and bias of radius reconstruction vs energy	65
3005 d	Resolution and bias of radius reconstruction vs radius	65
3006 e	Resolution and bias of radius reconstruction vs θ	65
3007 f	Resolution and bias of radius reconstruction vs ϕ	65

3008	4.9	Residual distribution of the different component of the vertex by Gen ₃₀ . The reconstructed component are x , y and z but we see similar behavior in the error of R , θ and ϕ .	66
3009	a	Distribution of the error on reconstructed x by Gen ₃₀	66
3010	b	Distribution of the error on reconstructed y by Gen ₃₀	66
3011	c	Distribution of the error on reconstructed z by Gen ₃₀	66
3012	d	Distribution of the error on reconstructed R by Gen ₃₀	66
3013	e	Distribution of the error on reconstructed θ by Gen ₃₀	66
3014	f	Distribution of the error on reconstructed ϕ by Gen ₃₀	66
3015	4.10	67
3016	a	Distribution of Gen ₃₀ reconstructed energy and true energy of the analysis dataset (J21)	67
3017	b	Distribution of Gen ₄₂ reconstructed energy and true energy of the analysis dataset (J23)	67
3018	4.11	Radius bias (on the left) and resolution(on the right) of the classical algorithm in a E , R^3 grid	68
3019	4.12	Reconstruction performance of the Gen ₃₀ model on J21, the classic algorithm “Classical algorithm” from [63] and the combination of both using weighted mean. The top part of each plot is the resolution and the bottom part is the bias.	69
3020	a	Resolution and bias of energy reconstruction vs energy	69
3021	b	Resolution and bias of energy reconstruction vs radius	69
3022	c	Resolution and bias of radius reconstruction vs energy	69
3023	d	Resolution and bias of radius reconstruction vs radius	69
3024	e	Resolution and bias of radius reconstruction vs θ	69
3025	f	Resolution and bias of radius reconstruction vs ϕ	69
3026	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	70
3027	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 [63]. The top part of each plot is the resolution and the bottom part is the bias.	71
3028	a	Resolution and bias of energy reconstruction vs energy	71
3029	b	Resolution and bias of energy reconstruction vs radius	71
3030	c	Resolution and bias of radius reconstruction vs energy	71
3031	d	Resolution and bias of radius reconstruction vs radius	71
3032	e	Resolution and bias of radius reconstruction vs θ	71
3033	f	Resolution and bias of radius reconstruction vs ϕ	71
3034	5.1	75
3035	a	Illustration of the different nodes in our graphs and their relations.	75
3036	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.	75
3037	5.2	Illustration of the Healpix segmentation. On the left: A segmentation of order 0. On the right: A segmentation of order 1	75
3038	5.3	Illustration of the different update function needed by our GNN	78
3039	5.4	Distribution of the number of hits depending on the energy. On the right: for the LPMT system. In the middle : for the SPMT system. On the left: For both system.	79
3040	a	79
3041	b	79
3042	c	79

3058	5.5	Distribution of the number of hits depending on the radius. On the right: for the LPMT system. On the right : for the SPMT system. To prevent the superposition of structure of different scales we limit ourselves to the energy range $E_{true} \in [0, 9]$	79
3059	a	79	
3060	b	79	
3061			
3062	5.6	Schema of the JWGv8.4.0 architecture, the colored triplet is the graph configuration after each JWG layers	81
3063			
3064	5.7	Energy reconstruction depending on the true energy for samples of the different versions of the GNN	82
3065			
3066	5.8	Reconstruction performance of the Omilrec algorithm based on QTML presented in section 2.7, JWGv8.4 presented in this chapter. The top part of each plot is the resolution and the bottom part is the bias.	84
3067	a	Resolution and bias of energy reconstruction vs energy	84
3068	b	Resolution and bias of energy reconstruction vs radius	84
3069			
3070	5.9	Reconstruction performance of the Omilrec algorithm based on QTML presented in section 2.7, JWGv8.4 presented in this chapter. The top part of each plot is the resolution and the bottom part is the bias.	85
3071	a	Resolution and bias of radius reconstruction vs energy	85
3072	b	Resolution and bias of radius reconstruction vs radius	85
3073			
3074	5.10	Reconstruction performance of the Omilrec algorithm based on QTML presented in section 2.7, JWGv8.4 presented in this chapter. The top part of each plot is the resolution and the bottom part is the bias.	86
3075	a	Resolution and bias of radius reconstruction vs θ	86
3076	b	Resolution and bias of radius reconstruction vs ϕ	86
3077			
3078	5.11	Reconstruction performance of the Omilrec algorithm, JWGv8.4 and the combination between the two using the optimal variance estimator presented in annex A.2. The top part of each plot is the resolution and the bottom part is the bias.	87
3079	a	Resolution and bias of energy reconstruction vs energy	87
3080	b	Resolution and bias of energy reconstruction vs radius	87
3081			
3082	5.12	Reconstruction performance of the Omilrec algorithm, JWGv8.4 and the combination between the two using the optimal variance estimator presented in annex A.2. The top part of each plot is the resolution and the bottom part is the bias.	88
3083	a	Resolution and bias of radius reconstruction vs energy	88
3084	b	Resolution and bias of radius reconstruction vs radius	88
3085			
3086	5.13	Reconstruction performance of the Omilrec algorithm based on QTML presented in section 2.7, JWGv8.4 presented in this chapter and the HCNN algorithm. The top part of each plot is the resolution and the bottom part is the bias.	89
3087	a	Resolution and bias of energy reconstruction vs energy	89
3088	b	Resolution and bias of energy reconstruction vs radius	89
3089			
3090	5.14	Reconstruction performance of the Omilrec algorithm based on QTML presented in section 2.7, JWGv8.4 presented in this chapter and the HCNN algorithm. The top part of each plot is the resolution and the bottom part is the bias.	89
3091	a	Resolution and bias of radius reconstruction vs energy	89
3092	b	Resolution and bias of radius reconstruction vs radius	89
3093			
3094	6.1	Schema of the method to discover vulnerabilities in the reconstruction methods. On the top of the image , the standard data flow. The individual charge and times are fed to a reconstruction algorithm. From the reconstructed energies, we can produce an IBD spectrum and compute control observables from the control samples. On the bottom , the same data flow but we add an ANN between the input and the reconstruction. The ANN will slightly change the input charge and time so the reconstruction algorithm inaccurately reconstruct the IBD energy, but the perturbation is not visible in the control sample.	93
3095			
3096			
3097			
3098			
3099			
3100			
3101			
3102			
3103			
3104			
3105			
3106			
3107			
3108			
3109			

3110 7.1	Expected number of neutrinos event per MeV in JUNO after 6 years of data taking. The black curve shows the flux if there was no oscillation. The light gray curve shows the oscillation if only the solar terms are taken in account ($\theta_{12}, \Delta m_{21}^2$). 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.	98
3117 7.2	Oscillated reactor $\bar{\nu}_e$ spectra for the Normal Ordering (Black) and Inverted Ordering (Red) for 6,5 years data taking and a resolution of 3% without any statistical or sys- tematic fluctuation. Figure from [82].	99
3120 7.3	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.	101
3123 7.4	102
3124 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 $R < 4m$ to not be affected by the non-uniformity.	102
3125 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 $R < 4m$ to not be affected by the non-uniformity. The dots represent the mean of the distributions in figure 7.4a 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 [27].	102
3133 7.5	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 .	106
3135 7.6	Schematic description of the fit framework	107
3136 7.7	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.	108
3139 7.8	Distribution of BFP - nominal value for 1000 toy Standard joint fit. 6 years exposure, all background, Pearson χ^2, θ_{13} fixed.	111
3141 7.9	Distribution of BFP - nominal value for 1000 toy Standard joint fit. 6 years exposure, all background, PearsonV χ^2, θ_{13} fixed.	112
3143 7.10	Distribution of BFP - nominal value for 5000 toy Delta joint fit. 6 years exposure, all background, PearsonV χ^2, θ_{13} fixed.	113
3145 7.11	Top: Theoretical spectrum without QNL (in red) and with $\alpha_{qnl} = 1\%$ (in blue). Bottom: Ratio between the theoretical spectrum with and without QNL.	114
3147 7.12	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. .	115
3149 7.13	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	116
3151 a	116
3152 b	116
3153 c	116
3154 7.14	Difference between the element of the theoretical and empiric correlation matrix . . .	117
3155 a	117
3156 b	117
3157 7.15	Correlation on the reconstruction error between the LPMT and SPMT system as a function of (On the left) the energy, (On the right) the radius. The SPMT recon- struction comes from the NN presented in Chapter 4 and the LPMT reconstruction comes from OMILREC presented in section 2.7. To prevent effect due to the CNN bad reconstruction, we select the event with $1 < E_{dep} < 9$ MeV.	118

3162	7.16 Correlation on the reconstruction error between the LPMT and SPMT system as a function of the energy and the radius. The SPMT reconstruction comes from the NN presented in Chapter 4 and the LPMT reconstruction comes from OMILREC presented in section 2.7. To prevent effect due to the CNN bad reconstruction, we select the event with $1 < E_{dep} < 9$ MeV.	119
3167	7.17 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.	120
3170	7.18 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.	121
3173	7.19 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.	122
3176	a 100 days exposure	122
3177	b 1 year exposure	122
3178	c 2 years exposure	122
3179	d 6 years exposure	122
3180	7.20 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.	123
3183	a 100 days exposure	123
3184	b 1 year exposure	123
3185	c 2 years exposure	123
3186	d 6 years exposure	123
3187	7.21 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.	124
3190	a 100 days exposure	124
3191	b 1 year exposure	124
3192	c 2 years exposure	124
3193	d 6 years exposure	124
3194	B.1 Illustration of the real part of the spherical harmonics	130
3195	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.	130
3196	B.3 Error on the reconstructed radius vs the true radius by the harmonic method	131
3197	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.	132
3202	a	132
3203	b	132
3204	c	132
3205	d	132
3206	e	132
3207	f	132
3208	g	132
3209	h	132
3210	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.	133
3211		

3212 B.6 Plot of the distribution of the relative power of each harmonic dependent on R^3 (on
 3213 the left). The Total Reflection (TR) area is represented by the horizontal blue line. The
 3214 distribution are fitted using a 9th degree polynomial (red curve). The relative power
 3215 error between the distribution and the fit is represented on the left. **Part 1** 134

3216 B.7 Plot of the distribution of the relative power of each harmonic dependent on R^3 (on
 3217 the left). The Total Reflection (TR) area is represented by the horizontal blue line. The
 3218 distribution are fitted using a 9th degree polynomial (red curve). The relative power
 3219 error between the distribution and the fit is represented on the left. **Part 2** 135

3220 D.1 Comparison between Omilrec reconstructed E_{vis} and the deposited energy E_{dep} . The
 3221 profile of the distribution E_{vis}/E_{dep} vs E_{dep} is fitted with a 5th degree polynomial. 140

³²²² List of Abbreviations

ACU	Automatic Calibration Unit
BDT	Boosted Decision Tree
BFP	Best Fit Point
CD	Central Detector
CLS	Cable Loop System
CNN	Convolutional NN
DNN	Deep NN
DN	Dark Noise
EDM	Event Data Model
FCDNN	Fully Connected Deep NN
GNN	Graph NN
GT	Guiding Tube
IBD	Inverse Beta Decay
IO	Inverse Ordering
JUNO	Jiangmen Underground Neutrino Observatory
LPMT	Large PMT
LR	Learning Rate
LS	Liquid Scintillator
MC	Monte Carlo simulation
ML	Machine Learning
MSE	Mean Squared Error
NMO	Neutrino Mass Ordering
NN	Neural Network
NO	Normal Ordering
NPE	Number of Photo Electron
OSIRIS	Online Scintillator Internal Radioactivity Investigation System
PE	Photo Electron
PMT	Photo-Multipliers Tubes
PRelu	Parametrized Rectified Linear Unit
QNL	Charge (Q) Non Linearity
ROV	Remotely Operated under-LS Vehicle
ReLU	Rectified Linear Unit
ResNet	Residual Network
SGD	Stochastic Gradient Descent
SPMT	Small PMT
TAO	Taishan Antineutrino Oservatory
TR Area	Total Reflexion Area
TTS	Time Transit Spread
TT	Top Tracker
UWB	Under Water Boxes
WCD	Water Cherenkov Detector

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](https://doi.org/10.1103/PhysRevD.78.111103). eprint: [0807.3203\[hep-ex, physics:hep-ph\]](https://arxiv.org/abs/0807.3203). URL: [http://arxiv.org/abs/0807.3203](https://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](https://doi.org/10.1088/0954-3899/43/3/030401). eprint: [1507.05613\[hep-ex, physics:physics\]](https://arxiv.org/abs/1507.05613). URL: [http://arxiv.org/abs/1507.05613](https://arxiv.org/abs/1507.05613) (visited on 07/28/2023).
- [3] JUNO Collaboration et al. "Sub-percent Precision Measurement of Neutrino Oscillation Parameters with JUNO". *Chinese Physics C* 46.12 (Dec. 1, 2022), 123001. ISSN: 1674-1137, 2058-6132. DOI: [10.1088/1674-1137/ac8bc9](https://doi.org/10.1088/1674-1137/ac8bc9). eprint: [2204.13249\[hep-ex\]](https://arxiv.org/abs/2204.13249). URL: [http://arxiv.org/abs/2204.13249](https://arxiv.org/abs/2204.13249) (visited on 08/11/2023).
- [4] A. A. Hahn, K. Schreckenbach, W. Gelletly, F. von Feilitzsch, G. Colvin, and B. Krusche. "Antineutrino spectra from 241Pu and 239Pu thermal neutron fission products". *Physics Letters B* 218.3 (Feb. 23, 1989), 365–368. ISSN: 0370-2693. DOI: [10.1016/0370-2693\(89\)91598-0](https://doi.org/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](https://doi.org/10.1103/PhysRevC.83.054615). eprint: [1101.2663\[hep-ex, physics:nucl-ex\]](https://arxiv.org/abs/1101.2663). URL: [http://arxiv.org/abs/1101.2663](https://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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1103/PhysRevC.84.024617). eprint: [1106.0687\[hep-ex, physics:hep-ph, physics:nucl-ex, physics:nucl-th\]](https://arxiv.org/abs/1106.0687). URL: [http://arxiv.org/abs/1106.0687](https://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](https://doi.org/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](https://doi.org/10.1103/PhysRevLett.114.012502). eprint: [1407.1281\[hep-ex, physics:nucl-ex\]](https://arxiv.org/abs/1407.1281). URL: [http://arxiv.org/abs/1407.1281](https://arxiv.org/abs/1407.1281) (visited on 01/16/2024).

- [11] Daya Bay Collaboration et al. "Measurement of the Reactor Antineutrino Flux and Spectrum at Daya Bay". *Physical Review Letters* 116.6 (Feb. 12, 2016). Publisher: American Physical Society, 061801. DOI: [10.1103/PhysRevLett.116.061801](https://doi.org/10.1103/PhysRevLett.116.061801). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.116.061801> (visited on 09/06/2024).
- [12] G. Mention, M. Fechner, Th. Lasserre, Th. A. Mueller, D. Lhuillier, M. Cribier, and A. Letourneau. "Reactor antineutrino anomaly". *Physical Review D* 83.7 (Apr. 29, 2011). Publisher: American Physical Society, 073006. DOI: [10.1103/PhysRevD.83.073006](https://doi.org/10.1103/PhysRevD.83.073006). URL: <https://link.aps.org/doi/10.1103/PhysRevD.83.073006> (visited on 03/05/2024).
- [13] JUNO Collaboration et al. *TAO Conceptual Design Report: A Precision Measurement of the Reactor Antineutrino Spectrum with Sub-percent Energy Resolution*. May 18, 2020. DOI: [10.48550/arXiv.2005.08745](https://arxiv.org/abs/2005.08745). eprint: [2005.08745 \[hep-ex, physics:nucl-ex, physics:physics\]](https://arxiv.org/abs/2005.08745). URL: <http://arxiv.org/abs/2005.08745> (visited on 01/18/2024).
- [14] 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](https://doi.org/10.1093/ptep/ptaa104). URL: <https://doi.org/10.1093/ptep/ptaa104> (visited on 12/04/2023).
- [15] 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](https://doi.org/10.1103/PhysRevD.104.122002). eprint: [2109.11174 \[astro-ph, physics:hep-ex\]](https://arxiv.org/abs/2109.11174). URL: <http://arxiv.org/abs/2109.11174> (visited on 02/28/2024).
- [16] JUNO Collaboration et al. "JUNO Sensitivity on Proton Decay $p \rightarrow \bar{\nu}K^+$ Searches". *Chinese Physics C* 47.11 (Nov. 1, 2023), 113002. ISSN: 1674-1137, 2058-6132. DOI: [10.1088/1674-1137/ace9c6](https://doi.org/10.1088/1674-1137/ace9c6). eprint: [2212.08502 \[hep-ex, physics:hep-ph\]](https://arxiv.org/abs/2212.08502). URL: <http://arxiv.org/abs/2212.08502> (visited on 08/09/2024).
- [17] 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](https://doi.org/10.1016/S0370-2693(03)00616-6). eprint: [astro-ph/0302055](https://arxiv.org/abs/astro-ph/0302055). URL: <http://arxiv.org/abs/astro-ph/0302055> (visited on 01/16/2024).
- [18] 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](https://arxiv.org/abs/2007.00314). eprint: [2007.00314 \[hep-ex, physics:physics\]](https://arxiv.org/abs/2007.00314). URL: [http://arxiv.org/abs/2007.00314](https://arxiv.org/abs/2007.00314) (visited on 07/26/2023).
- [19] 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](https://doi.org/10.1016/B978-0-08-010472-0.50008-2). URL: <https://www.sciencedirect.com/science/article/pii/B9780080104720500082> (visited on 02/07/2024).
- [20] *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).
- [21] 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](https://doi.org/10.1016/j.nima.2020.163860). eprint: [2003.12212 \[physics\]](https://arxiv.org/abs/2003.12212). URL: <http://arxiv.org/abs/2003.12212> (visited on 02/07/2024).
- [22] 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](https://doi.org/10.1088/1748-0221/12/11/T11004). eprint: [1703.01867 \[hep-ex, physics:physics\]](https://arxiv.org/abs/1703.01867). URL: [http://arxiv.org/abs/1703.01867](https://arxiv.org/abs/1703.01867) (visited on 07/28/2023).
- [23] 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](https://arxiv.org/abs/2103.16900). eprint: [2103.16900 \[physics\]](https://arxiv.org/abs/2103.16900). URL: <http://arxiv.org/abs/2103.16900> (visited on 02/07/2024).
- [24] 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/10](https://doi.org/10.1140/epjc/10.1140/epjc/10)

- 3320 s10052-022-11002-8. eprint: 2205.08629 [hep-ex, physics:physics]. URL: <http://arxiv.org/abs/2205.08629> (visited on 02/08/2024).
- 3321
- 3322 [25] Yang Han. "Dual Calorimetry for High Precision Neutrino Oscillation Measurement at JUNO Experiment". *AstroParticule et Cosmologie*, France, Paris U. VII, APC, June 2021.
- 3323
- 3324 [26] 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).
- 3325
- 3326 [27] 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).
- 3327
- 3328 [28] 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).
- 3329
- 3330 [29] 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).
- 3331
- 3332 [30] 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).
- 3333
- 3334 [31] 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).
- 3335
- 3336 [32] 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).
- 3337
- 3338 [33] 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).
- 3339
- 3340 [34] 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).
- 3341
- 3342 [35] 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).
- 3343
- 3344 [36] 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).
- 3345
- 3346 [37] 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:

- 3373 0168-9002. DOI: [10.1016/j.nima.2006.07.058](https://doi.org/10.1016/j.nima.2006.07.058). URL: <https://www.sciencedirect.com/science/article/pii/S0168900206013519> (visited on 03/07/2024).
- 3374 [38] 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](https://doi.org/10.1016/S0029-554X(79)90170-8). URL: <https://www.sciencedirect.com/science/article/pii/S0029554X79901708> (visited on 03/07/2024).
- 3375 [39] 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](https://doi.org/10.1007/s41365-023-01240-0). URL: <https://doi.org/10.1007/s41365-023-01240-0> (visited on 08/17/2023).
- 3376 [40] 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](https://doi.org/10.1016/j.nima.2021.165527). eprint: [2101.04839\[hep-ex, physics:physics\]](https://arxiv.org/abs/2101.04839). URL: <http://arxiv.org/abs/2101.04839> (visited on 07/24/2023).
- 3377 [41] 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](https://doi.org/10.1140/epjc/s10052-022-11004-6). eprint: [2206.09040\[physics\]](https://arxiv.org/abs/2206.09040). URL: <http://arxiv.org/abs/2206.09040> (visited on 07/24/2023).
- 3378 [42] 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](https://doi.org/10.1088/1748-0221/17/11/P11003). URL: <https://dx.doi.org/10.1088/1748-0221/17/11/P11003> (visited on 04/04/2024).
- 3379 [43] 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](https://doi.org/10.1088/1748-0221/16/10/C10011). URL: <https://dx.doi.org/10.1088/1748-0221/16/10/C10011> (visited on 04/04/2024).
- 3380 [44] 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](https://doi.org/10.1088/1748-0221/16/07/P07041). URL: <https://dx.doi.org/10.1088/1748-0221/16/07/P07041> (visited on 04/04/2024).
- 3381 [45] 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](https://doi.org/10.1103/PhysRevD.102.092003). URL: <https://link.aps.org/doi/10.1103/PhysRevD.102.092003> (visited on 04/04/2024).
- 3382 [46] 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](https://doi.org/10.1086/427976). URL: <https://ui.adsabs.harvard.edu/abs/2005ApJ...622..759G> (visited on 04/04/2024).
- 3383 [47] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. *Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering*. Feb. 5, 2017. DOI: [10.48550/arXiv.1606.09375](https://doi.org/10.48550/arXiv.1606.09375). eprint: [1606.09375\[cs, stat\]](https://arxiv.org/abs/1606.09375). URL: [http://arxiv.org/abs/1606.09375](https://arxiv.org/abs/1606.09375) (visited on 04/04/2024).
- 3384 [48] 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](https://doi.org/10.1016/j.ppnp.2021.103927). eprint: [2104.02565\[hep-ex\]](https://arxiv.org/abs/2104.02565). URL: [http://arxiv.org/abs/2104.02565](https://arxiv.org/abs/2104.02565) (visited on 09/18/2023).
- 3385 [49] 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](https://doi.org/10.1201/9781315139470).

- [50] 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](https://doi.org/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).
- [51] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Jan. 29, 2017. DOI: [10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980). eprint: [1412.6980\[cs\]](https://arxiv.org/abs/1412.6980). URL: [http://arxiv.org/abs/1412.6980](https://arxiv.org/abs/1412.6980) (visited on 05/13/2024).
- [52] 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](https://doi.org/10.1109/CVPR.2016.90). URL: <https://ieeexplore.ieee.org/document/7780459> (visited on 07/17/2024).
- [53] Olga Russakovsky et al. *ImageNet Large Scale Visual Recognition Challenge*. Jan. 29, 2015. DOI: [10.48550/arXiv.1409.0575](https://doi.org/10.48550/arXiv.1409.0575). eprint: [1409.0575\[cs\]](https://arxiv.org/abs/1409.0575). URL: [http://arxiv.org/abs/1409.0575](https://arxiv.org/abs/1409.0575) (visited on 05/17/2024).
- [54] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Apr. 10, 2015. DOI: [10.48550/arXiv.1409.1556](https://doi.org/10.48550/arXiv.1409.1556). eprint: [1409.1556\[cs\]](https://arxiv.org/abs/1409.1556). URL: [http://arxiv.org/abs/1409.1556](https://arxiv.org/abs/1409.1556) (visited on 05/17/2024).
- [55] 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).
- [56] 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](https://doi.org/10.1145/3620665.3640366). URL: <https://pytorch.org/assets/pytorch2-2.pdf> (visited on 07/16/2024).
- [57] 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](https://doi.org/10.1109/5.726791). URL: <https://ieeexplore.ieee.org/document/726791> (visited on 07/16/2024).
- [58] 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).
- [59] 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](https://doi.org/10.48550/arXiv.1704.01212). eprint: [1704.01212\[cs\]](https://arxiv.org/abs/1704.01212). URL: [http://arxiv.org/abs/1704.01212](https://arxiv.org/abs/1704.01212) (visited on 05/22/2024).
- [60] 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](https://doi.org/10.48550/arXiv.1707.01926). eprint: [1707.01926\[cs, stat\]](https://arxiv.org/abs/1707.01926). URL: [http://arxiv.org/abs/1707.01926](https://arxiv.org/abs/1707.01926) (visited on 05/22/2024).
- [61] 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](https://doi.org/10.48550/arXiv.1406.2661). eprint: [1406.2661\[cs, stat\]](https://arxiv.org/abs/1406.2661). URL: [http://arxiv.org/abs/1406.2661](https://arxiv.org/abs/1406.2661) (visited on 05/29/2024).
- [62] Anatael Cabrera et al. *Multi-Calorimetry in Light-based Neutrino Detectors*. Dec. 20, 2023. DOI: [10.48550/arXiv.2312.12991](https://doi.org/10.48550/arXiv.2312.12991). eprint: [2312.12991\[hep-ex, physics:physics\]](https://arxiv.org/abs/2312.12991). URL: [http://arxiv.org/abs/2312.12991](https://arxiv.org/abs/2312.12991) (visited on 08/19/2024).
- [63] 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).
- [64] 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](https://doi.org/10.48550/arXiv.1202.2745). eprint: [1202.2745\[cs\]](https://arxiv.org/abs/1202.2745). URL: [http://arxiv.org/abs/1202.2745](https://arxiv.org/abs/1202.2745) (visited on 06/27/2024).

- [65] R. Abbasi et al. "A Convolutional Neural Network based Cascade Reconstruction for the Ice-Cube Neutrino Observatory". *Journal of Instrumentation* 16.7 (July 1, 2021), P07041. ISSN: 1748-0221. DOI: [10.1088/1748-0221/16/07/P07041](https://doi.org/10.1088/1748-0221/16/07/P07041). eprint: [2101.11589\[hep-ex\]](https://arxiv.org/abs/2101.11589). URL: <http://arxiv.org/abs/2101.11589> (visited on 06/27/2024).
- [66] 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](https://doi.org/10.1088/1475-7516/2021/11/051). URL: <https://dx.doi.org/10.1088/1475-7516/2021/11/051> (visited on 06/27/2024).
- [67] Taco S. Cohen, Mario Geiger, Jonas Koehler, and Max Welling. *Spherical CNNs*. Feb. 25, 2018. DOI: [10.48550/arXiv.1801.10130](https://doi.org/10.48550/arXiv.1801.10130). eprint: [1801.10130\[cs,stat\]](https://arxiv.org/abs/1801.10130). URL: <http://arxiv.org/abs/1801.10130> (visited on 07/13/2024).
- [68] NVIDIA A100 GPUs Power the Modern Data Center. NVIDIA. URL: <https://www.nvidia.com/en-gb/data-center/a100/> (visited on 08/06/2024).
- [69] NVIDIA V100. NVIDIA. URL: <https://www.nvidia.com/en-gb/data-center/v100/> (visited on 08/06/2024).
- [70] 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).
- [71] "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](https://doi.org/10.1109/IEEESTD.2019.8766229). URL: <https://ieeexplore.ieee.org/document/8766229> (visited on 07/03/2024).
- [72] 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](https://doi.org/10.1016/j.nima.2021.165347). eprint: [2102.11538\[hep-ex,physics:physics\]](https://arxiv.org/abs/2102.11538). URL: <http://arxiv.org/abs/2102.11538> (visited on 02/08/2024).
- [73] 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](https://doi.org/10.1086/427976). eprint: [astro-ph/0409513](https://arxiv.org/abs/astro-ph/0409513). URL: <http://arxiv.org/abs/astro-ph/0409513> (visited on 08/10/2023).
- [74] 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](https://doi.org/10.1088/1674-1137/41/6/066201). URL: <https://dx.doi.org/10.1088/1674-1137/41/6/066201> (visited on 08/16/2024).
- [75] 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).
- [76] Mario Schwarz, Sabrina M. Franke, Lothar Oberauer, Miriam D. Plein, Hans Th J. Steiger, and Marc Tippmann. *Measurements of the Lifetime of Orthopositronium in the LAB-Based Liquid Scintillator of JUNO*. Apr. 25, 2018. DOI: [10.1016/j.nima.2018.12.068](https://doi.org/10.1016/j.nima.2018.12.068). eprint: [1804.09456\[physics\]](https://arxiv.org/abs/1804.09456). URL: <http://arxiv.org/abs/1804.09456> (visited on 09/17/2024).
- [77] Narongkiat Rodphai, Zhimin Wang, Narumon Suwonjandee, and Burin Asavapibhop. "20-inch photomultiplier tube timing study for JUNO". *Journal of Physics: Conference Series* 2145.1 (Dec. 2021). Publisher: IOP Publishing, 012017. ISSN: 1742-6596. DOI: [10.1088/1742-6596/2145/1/012017](https://doi.org/10.1088/1742-6596/2145/1/012017). URL: <https://dx.doi.org/10.1088/1742-6596/2145/1/012017> (visited on 09/17/2024).
- [78] Dong-Hao Liao et al. "Study of TTS for a 20-inch dynode PMT*". *Chinese Physics C* 41.7 (July 2017). Publisher: IOP Publishing, 076001. ISSN: 1674-1137. DOI: [10.1088/1674-1137/41/7/076001](https://doi.org/10.1088/1674-1137/41/7/076001). URL: <https://dx.doi.org/10.1088/1674-1137/41/7/076001> (visited on 09/17/2024).

- 3530 [79] Nan Li et al. "Characterization of 3-inch photomultiplier tubes for the JUNO central detector".
 3531 *Radiation Detection Technology and Methods* 3.1 (Nov. 22, 2018), 6. ISSN: 2509-9949. DOI: [10.1007/s41605-018-0085-8](https://doi.org/10.1007/s41605-018-0085-8). URL: <https://doi.org/10.1007/s41605-018-0085-8> (visited on
 3532 09/17/2024).
- 3533 [80] B. Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil
 3534 Yogamani, and Patrick Pérez. *Deep Reinforcement Learning for Autonomous Driving: A Survey*.
 3535 Jan. 23, 2021. eprint: [2002.00444\[cs\]](https://arxiv.org/abs/2002.00444). URL: [http://arxiv.org/abs/2002.00444](https://arxiv.org/abs/2002.00444) (visited on
 3536 10/02/2024).
- 3537 [81] Oriol Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". 575.7782 (Nov. 2019). Publisher: Nature Publishing Group, 350–354. ISSN: 1476-4687. DOI:
 3538 [10.1038/s41586-019-1724-z](https://doi.org/10.1038/s41586-019-1724-z). URL: <https://www.nature.com/articles/s41586-019-1724-z> (visited on 10/02/2024).
- 3539 [82] Angel Abusleme et al. "Potential to Identify the Neutrino Mass Ordering with Reactor An-
 3540 tineutrinos in JUNO" (May 2024). eprint: 2405.18008.
- 3541 [83] Daya Bay Collaboration et al. *A high precision calibration of the nonlinear energy response at Daya*
 3542 *Bay*. arXiv.org. Feb. 21, 2019. URL: <https://arxiv.org/abs/1902.08241v2> (visited on
 3543 10/01/2024).
- 3544 [84] Rene Brun et al. *root-project/root: v6.26/06*. Version v6-26-06. Mar. 3, 2022. DOI: [10.5281/zenodo.3895860](https://doi.org/10.5281/zenodo.3895860). URL: <https://zenodo.org/records/3895860> (visited on 09/05/2024).
- 3545 [85] X. B. Ma, W. L. Zhong, L. Z. Wang, Y. X. Chen, and J. Cao. "Improved calculation of the energy
 3546 release in neutron-induced fission". *Physical Review C* 88.1 (July 12, 2013). Publisher: American
 3547 Physical Society, 014605. DOI: [10.1103/PhysRevC.88.014605](https://doi.org/10.1103/PhysRevC.88.014605). URL: <https://link.aps.org/doi/10.1103/PhysRevC.88.014605> (visited on 09/06/2024).
- 3548 [86] 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](https://doi.org/10.1525/collabra.87615). URL: <https://doi.org/10.1525/collabra.87615> (visited on 09/10/2024).
- 3549 [87] "Note Sur Une Méthode de Résolution des équations Normales Provenant de L'Application
 3550 de la MéThode des Moindres Carrés a un Système D'équations Linéaires en Nombre Inférieur
 3551 a Celui des Inconnues. — Application de la Méthode a la Résolution D'un Système Defini
 3552 D'éQuations LinéAires". *Bulletin géodésique* 2.1 (Apr. 1, 1924), 67–77. ISSN: 1432-1394. DOI: [10.1007/BF03031308](https://doi.org/10.1007/BF03031308). URL: <https://doi.org/10.1007/BF03031308> (visited on 09/10/2024).
- 3553 [88] Pauli Virtanen et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python".
 3554 *Nature Methods* 17.3 (Mar. 2020). Publisher: Nature Publishing Group, 261–272. ISSN: 1548-7105.
 3555 DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2). URL: <https://www.nature.com/articles/s41592-019-0686-2> (visited on 08/14/2024).

3565

3566

Titre : Méthode Deep Learning and analyse Double Calorimétrique pour la mesure de haute précision des paramètres d'oscillation des neutrinos dans JUNO

Mot clés : Neutrinos; expérience JUNO; Deep Learning; reconstruction d'IBD; oscillations des neutrinos; double calorimetrie

Résumé : JUNO est un observatoire de neutrinos à scintillateur liquide, polyvalent et medium baseline (environ 52 km), situé en Chine. Ses principaux objectifs sont de mesurer les paramètres d'oscillation θ_{12} , Δm_{21}^2 et Δm_{31}^2 avec une précision au pour-mille et de déterminer l'ordre des masses des neutrinos avec un niveau de confiance de 3σ . Atteindre ces objectifs nécessite une résolution énergétique sans précédent de $3\%/\sqrt{E(\text{MeV})}$ avec cette technologie. Cela demande une compréhension approfondie des divers effets au sein du détecteur.

Le système de double calorimetrie, composé de deux systèmes de mesure distincts observant le même événement, permet non seulement une calibration mais aussi une détection des effets du détecteur avec une grande précision, comme démontré dans cette thèse. Le Deep Learning, un outil de plus en plus utilisé en physique expérimentale, joue un rôle crucial dans cet effort. Dans cette thèse, je présente le développement, l'application et l'analyse des techniques de Deep Learning pour la reconstruction d'évènements dans l'expérience JUNO.

3598

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

Keywords: Neutrinos; JUNO experiment; Deep learning; IBD reconstruction; neutrinos Oscillation; dual Calorimetry

Abstract: JUNO is a multipurpose, medium baseline (~ 52 km) liquid scintillator neutrino observatory located in China. Its primary objectives are to measure the oscillation parameters θ_{12} , Δm_{21}^2 , and Δm_{31}^2 with per mil precision and to determine the neutrino mass ordering at a 3σ confidence level. Achieving these goals requires an unprecedented energy resolution of $3\%/\sqrt{E(\text{MeV})}$ with this technology. This demands a comprehensive understanding of the various effects within the

detect. The Dual Calorimetry system—two distinct measurement systems observing the same event-enables not only high-precision calibration but also detection of detector effects, as demonstrated in this thesis. Deep learning, an increasingly powerful tool in physics, plays a critical role in this effort. In this thesis, I present the development, application, and analysis of Deep Learning techniques for reconstruction in the JUNO experiment.

