POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

# Modeling reactor antineutrino spectrum: machine learning correction to the summation method and mitigation of the pandemonium effect

Tesi di Laurea Magistrale in
Nuclear Engineering - Ingegneria Nucleare

Author: **Leonardo Maturi**

Student ID: 251762
Advisor: Prof. Antonio Cammi
Co-advisors: Prof. Monica Sisti, Lorenzo Loi, Dr. Matteo Borghesi
Academic Year: 2024-25

# Abstract

The accurate modeling of non-oscillated reactor antineutrino energy spectrum is a fundamental requirement for high precision neutrino oscillation experiments as JUNO, whose first aim is to the determine the neutrino mass ordering. The summation method, one of the primary tools for predicting such spectrum thanks to its modeling of the spectrum fine structure, is known to be affected by systematic uncertainties, culminating in the so-called *5 MeV bump*. A dominant source of discrepancy is the Pandemonium effect, an experimental bias coming from incomplete nuclear decay data measured with High Purity Germanium (HPGe) gamma ray detectors, leading to mis-predictions of the spectral shape, with a systematic underestimation of the low energy part. While Total Absorption Gamma-ray Spectroscopy (TAGS) measurements provide the necessary corrections, they are available for only a limited number of fission fragments.

This thesis presents a machine learning framework and methodology designed to study and correct the Pandemonium effect. A dataset was constructed, comprehensive of 52 isotopes with available TAGS measuremnts and 52 *SAFE isotopes* for which HPGe data is considered reliable. Various supervised regression models were trained to predict the target normalized spectra shape (TAGS when available or HPGe for SAFE isotopes) using only nuclear properties and features derived from the HPGe data.

Four distinct models were developed and compared: two multitask Gaussian Process models — an Intrinsic Coregionalization Model (ICM) and a Linear Model of Coregionalization (LCM) — and two Artificial Neural Networks (a shallow and a deep one). The ICM model was identified as the best performing one, both on the single isotopes spectral shapes and on the summed spectrum of reference actinides ($^{235}$U and $^{238}$U). A mitigation of Pandemonium effect is indeed obtained.

This work establishes that data-driven approaches can provide robust and generalizable corrections to the summation method, and moreover highlights the impressive role they can play in the world of physics research. A valuable direction is offered by this methodology for the reduction of systematic discrepancies. The application of analogous or different data-driven techniques seems therefore an important possibility for physics, not

to be overlooked in future years. It is very important however to keep in mind the intrinsic statistical structure of this framework. The predictions obtained with this methodology must be regarded as complementary tools to future experimental efforts, rather than a definitive substitute for them.

**Keywords:** Reactor Antineutrino Spectrum, Pandemonium Effect, Summation Method, Machine Learning, Gaussian Processes, Nuclear Data, JUNO Experiment.

# Abstract in lingua italiana

La modellazione accurata dello spettro non oscillato degli antineutrini da reattore è un requisito fondamentale per esperimenti di oscillazione dei neutrini ad alta precisione come JUNO, il cui primo obiettivo è determinare l'ordinamento delle masse dei neutrini. È noto che il summation method, uno degli strumenti principali per prevedere tale spettro grazie alla sua modellazione della struttura fine dello spettro, è influenzato da incertezze sistematiche, che culminano nel cosiddetto 5 MeV bump. Una fonte dominante di discrepanza è l'effetto Pandemonio, un bias sperimentale derivante da dati incompleti sul decadimento nucleare misurati con rivelatori di raggi gamma al germanio ad alta purezza (HPGe), che porta a previsioni errate della forma spettrale, con una sottostima sistematica della parte a bassa energia. Sebbene le misurazioni della spettroscopia di raggi gamma ad assorbimento totale (TAGS) forniscano le correzioni necessarie, sono disponibili solo per un numero limitato di frammenti di fissione. Questa tesi presenta un framework e una metodologia di apprendimento automatico progettati per studiare e correggere l'effetto Pandemonio. È stato costruito un set di dati completo di 52 isotopi con misure TAGS disponibili e 52 isotopi SAFE per i quali i dati HPGe sono considerati affidabili. Sono stati addestrati vari modelli di regressione supervisionata per prevedere la forma normalizzata degli spettri target (TAGS quando disponibile o HPGe per gli isotopi SAFE) utilizzando solo proprietà e caratteristiche nucleari derivate dai dati HPGe. Sono stati sviluppati e confrontati quattro modelli distinti: due modelli basati su processi gaussiani multitasking — un modello di coregionalizzazione intrinseca (ICM) e un modello lineare di coregionalizzazione (LCM) — e due reti neurali artificiali (una superficiale e una profonda). Il modello ICM è stato identificato come quello più performante, sia sulle forme spettrali dei singoli isotopi che sullo spettro sommato degli attinidi di riferimento ($^{235}$U e $^{238}$U). Si ottiene infatti una effettiva mitigazione dell'effetto Pandemonio. Questo lavoro dimostra che gli approcci basati sui dati possono fornire correzioni robuste e generalizzabili al metodo di sommatoria e sottolinea inoltre il ruolo impressionante che possono svolgere nel mondo della ricerca fisica. Questa metodologia offre una direzione preziosa per la riduzione delle discrepanze sistematiche. L'applicazione di tecniche analoghe o meno, basate sui dati, sembra quindi un'importante possibilità per la fisica, da non trascurare negli anni futuri.

È molto importante tuttavia tenere presente la struttura intrinsecamente statistica di questi modelli. Le previsioni ottenute devono essere considerate strumenti complementari ai futuri sforzi sperimentali, piuttosto che un loro sostituto definitivo.

**Parole chiave:** Spettro di Antineutrini da Reattore, Effetto Pandemonio, Summation Method, Apprendimento Automatico, Processi Gaussiani, Dati Nucleari, Esperimento JUNO.

# Contents

**Abstract** **i**

**Abstract in lingua italiana** **iii**

**Contents** **v**

**1 The Elusive Neutrinos** **1**
1.1 Neutrinos in the Standard Model . . . . . 2
1.2 Neutrino Flavor Oscillations . . . . . 6
1.2.1 Two-flavor Oscillations . . . . . 7
1.2.2 Three-flavor Oscillations . . . . . 10
1.2.3 Oscillations in Matter . . . . . 11
1.3 Neutrino Mass Ordering . . . . . 12

**2 The JUNO Experiment** **15**
2.1 Overview of Experimental Facilities . . . . . 15
2.2 The JUNO detector design . . . . . 17
2.2.1 Central Detector . . . . . 19
2.2.2 Veto Detectors . . . . . 20
2.2.3 JUNO Liquid Scintillator . . . . . 20
2.3 IBD Reaction and Detection . . . . . 21

**3 Reactor Antineutrino Spectrum** **25**
3.1 Antineutrino Production in Nuclear Reactor Cores . . . . . 26
3.2 Summation Method . . . . . 27
3.2.1 Uncertainties Treatment . . . . . 30
3.3 Discrepancies in Reactor Antineutrino Spectrum Modeling . . . . . 32
3.4 Pandemonium Effect . . . . . 34
3.5 Total Absorption Gamma Ray Spectroscopy . . . . . 36

3.6 Codes for the Evaluation of Reactor Antineutrino Spectrum . . . . . 37
3.6.1 RenShape . . . . . 39

**4 Machine Learning: Concepts and Methods 43**
4.1 Fundamentals of Machine Learning . . . . . 43
4.2 Supervised Learning Principles . . . . . 45
4.3 Supervised Regression Models . . . . . 52
4.3.1 Gaussian Processes . . . . . 52
4.3.2 Artificial Neural Networks . . . . . 56

**5 Approaches and Strategies to Mitigate Pandemonium Effect 63**
5.1 Description of the Problem and Proposed Approaches . . . . . 63
5.1.1 The Transition Approach . . . . . 65
5.1.2 The Shape Approach . . . . . 66
5.2 Dataset Construction . . . . . 68
5.2.1 SAFE Dataset Selection . . . . . 68
5.2.2 Training-Test Split . . . . . 70
5.2.3 Nuclear Input . . . . . 70
5.2.4 Coded Information from the HPGe Measurements . . . . . 73
5.2.5 Preprocessing of the Target $\bar{\nu}_e$ Spectrum . . . . . 75
5.2.6 Final Dataset . . . . . 78
5.3 Description of the Supervised Regression Models . . . . . 79
5.3.1 Application of Multitask Gaussian Processes . . . . . 80
5.3.2 Application of Neural Networks . . . . . 84

**6 Analysis of Predictive Results 89**
6.1 Model Performance Expectations . . . . . 89
6.2 Spectral Shapes and Integral Predictions of Single Isotopes . . . . . 90
6.3 $^{235}$U and $^{238}$U Summed Spectrum Evaluation . . . . . 105

**Conclusions 109**

**Bibliography 113**

**A Appendix A 121**
A.1 General Training Techniques for Machine Learning Models . . . . . 121

A.1.1 Training Iterations and Epochs . . . . . 121
A.1.2 Gradient Descent and Optimization . . . . . 122
A.1.3 Early Stopping e Validation . . . . . 122
A.1.4 Link with Bayesian Models . . . . . 123
A.1.5 Data Scaling . . . . . 123
A.2 Reduced-Order Modeling . . . . . 124
A.2.1 Principal Component Analysis . . . . . 124
A.2.2 Autoencoders . . . . . 128
A.3 Monte Carlo Methods . . . . . 130

**List of Figures** **131**

**List of Tables** **137**

**Acknowledgements** **139**

# 1 | The Elusive Neutrinos

Neutrinos are elementary particles of the Standard Model of Particle Physics. They have always played an important role in expanding our knowledge of Nature, and they still do. In 1930, they were first introduced by Wolfgang Pauli [1] to resolve the apparent violation of energy and momentum conservation in beta decays. In the following years, Enrico Fermi [2] incorporated them into his theory of weak interactions, establishing their role in many nuclear processes.

Their elusive nature was already clear: a neutrino is a neutral lepton[1] that interacts only weakly. Their mass was too small to be experimentally measured at the time and was hence initially assumed to be zero. Detecting such particles was a significant challenge, as they leave no electromagnetic track.

From that moment on, many experiments were set up to reveal their existence [3], with confirmation only coming in the second half of the twentieth century. The electron antineutrino was first detected in 1956 by Reines and Cowan [4]. Around the same period, major developments in particle physics had revealed the existence of negatively charged leptons other than the familiar electron: the muon [5] and the tau [6]. Each charged lepton was expected to have an associated neutrino, giving rise to the concept of leptonic flavors: $e$, $\mu$, and $\tau$. The muon neutrino was discovered in 1962 by Lederman, Schwartz, and Steinberger [7], and the tau neutrino in 2000 by the DONUT experiment at Fermilab [8]. Meanwhile, an unexpected phenomenon was observed: in 1968, Ray Davis measured a solar neutrino flux corresponding to only one-third of the Solar Standard Model prediction [9]. This discrepancy marked the beginning of decades of research that led to the discovery of neutrino flavor oscillations, providing undeniable evidence that neutrinos have a non-zero mass [10].

The study of neutrinos is now one of the most active fields in particle physics. Fundamental questions regarding mass ordering, the Dirac or Majorana nature, the absolute mass scale, and CP violation still remain open.

[1] Leptons are fermions not interacting with the strong nuclear force.

## 1.1. Neutrinos in the Standard Model

The Standard Model of Particle Physics [11] is a quantum field theory describing fundamental particles and their interactions, based on the $SU(3)_C \otimes SU(2)_L \otimes U(1)_Y$ gauge symmetry group, where $C$, $L$, and $Y$ respectively represent color charge, left-handed chirality, and hypercharge. The relevance of symmetry groups was first proposed by Yang and Mills in 1954 [12], and is detailed in standard textbooks [11, 13–15].
These gauge symmetries determine which interactions are allowed, the number of associated gauge bosons, particle multiplets, and - critically for neutrinos - why they appear massless in the original formulation of the Standard Model.
$SU(3)_C$ describes rotations in a three-dimensional color space and governs strong interactions. Quarks exist in three colors, represented as a triplet:

$$\begin{pmatrix} q_r \\ q_g \\ q_b \end{pmatrix}, \tag{1.1}$$

where $r$, $g$, $b$ denote red, green, and blue. Each $q_i$ is a Dirac spinor.
$SU(2)_L$ describes rotations in a two-dimensional space and acts only on left-handed components of fermions. Fermions form weak isospin doublets:

$$\begin{pmatrix} u_L \\ d_L \end{pmatrix}, \tag{1.2}$$

where $u_L$ and $d_L$ are the left-handed Dirac spinors representing the positive and negative isospin projections of quarks or neutrino–charged lepton pairs. Only left-handed fermions participate in weak interactions [16].
Both $SU(3)_C$ and $SU(2)_L$ are non-abelian groups. This means they do not satisfy the commutative property. As a consequence, the associated gauge bosons posses self-interacting properties. $U(1)_Y$ is a one-dimensional abelian group associated with hypercharge. Unlike non-abelian groups, its gauge boson does not self-interact, similarly to QED [17].
In the SM, there are eight gluons mediating strong interactions, and four gauge bosons mediating electroweak interactions: $W^1$, $W^2$, $W^3$ from $SU(2)_L$ and $B$ from $U(1)_Y$. Linear combinations of $W^1$ and $W^2$ generate $W^\pm$, responsible for charged-current (CC) interactions, which maximally violate parity and hence have a V-A structure. Mixing of $W^3$ and $B$ through the Weinberg angle $\theta_W$ produces $Z^0$ and $\gamma$, respectively mediating neutral-

current (NC) and electromagnetic (EM) interactions. Conversely to CC interactions, NC and EM interaction do not posses the chiral character, as a consequence of Wienberg mixing. $W^{\pm}$ and $Z^0$ acquire mass via the Higgs mechanism, while the photon remains massless.

Neutrinos are massless chiral fermions in the SM: left-handed neutrinos form weak doublets with their charged leptons, while right-handed neutrinos are absent. This prevents a Dirac mass term for neutrinos in the Higgs Lagrangian. For all other fermions indeed a Dirac mass term is prescribed and justified with the Yukawa coupling to the Higgs field. A schematic overview of the particles of the original SM is shown in Fig. 1.1.

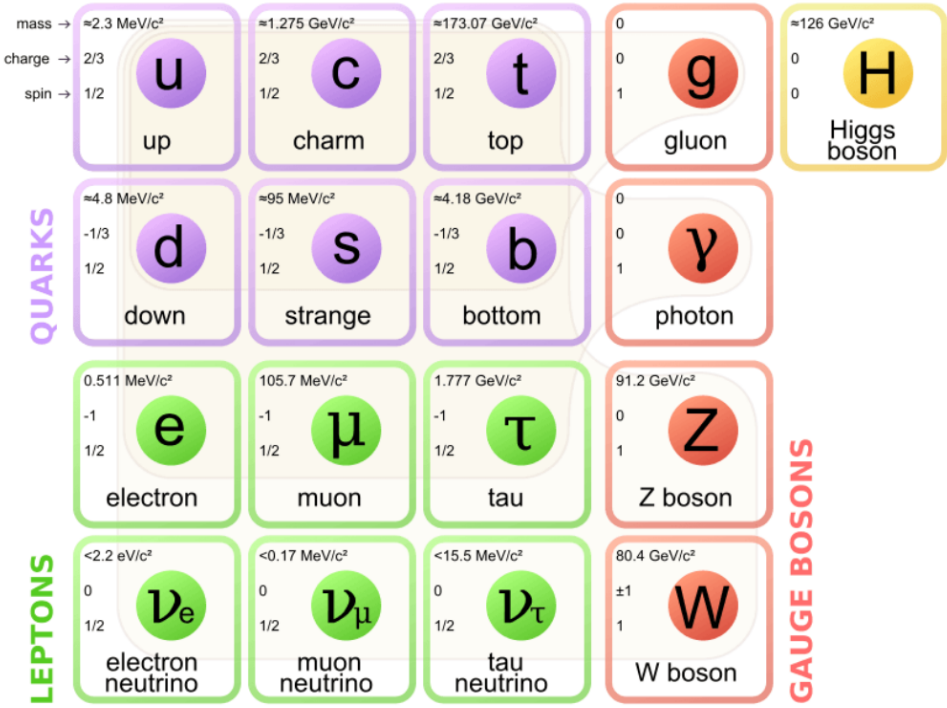



Figure 1.1: Schematic representation of the elementary particles described in the original Standard Model of Particle Physics and their fundamental properties. They are subdivided in: quarks (in purple), leptons (in green) and gauge bosons (in red). The Higgs boson (in yellow) is the particle associated to the Higgs field, responsible for the mass acquisition mechanism of the SM.

The discovery of neutrino oscillations showed that this description is incomplete. The first evidence came from the solar neutrino experiment by Ray Davis in 1968 [9]. The Solar Standard Model (SSM) predicted a flux of $\sim 6 \times 10^{10}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, dominated ($> 90\%$) by proton-proton (pp) nuclear fusion reactions (endpoint at 0.42 MeV), with smaller

contributions from $^7$B at 0.86 MeV and others, as shown in Fig.1.2. Even the ppIII chain

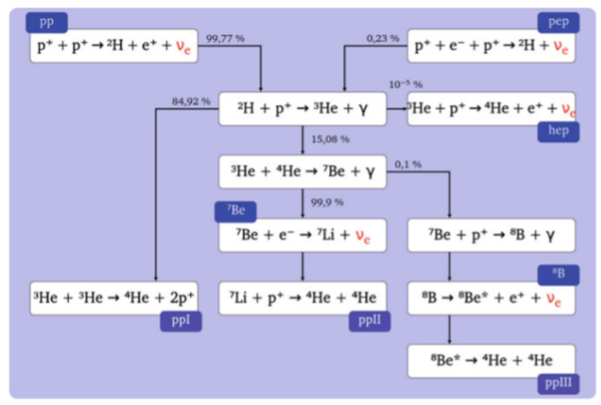



Figure 1.2: Main nuclear fusion reactions contributing to the solar neutrino flux. The figure is taken from [3].

(endpoint at 15 MeV), contributing only 0.1% of the spectrum, was crucial for Davis's experiment, which counted $^{37}$Ar atoms produced in a 615-ton tank of $C_2Cl_4$ via:

$$\nu_e + {}^{37}Cl \rightarrow {}^{37}Ar + e^-.$$

The observed rate was only about one-third of the expected value, initiating the solar neutrino problem. Following experiments, including GALLEX [18] (Gran Sasso, Italy) and SAGE [19] (Baksan, Russia), used radiochemical techniques with gallium targets to detect pp neutrinos. Water Cherenkov detectors, such as Kamiokande [20] and Super-Kamiokande [10], allowed detection of MeV energy neutrinos via Cherenkov radiation. The detectors enabled the reconstruction of their energy and direction, confirming the solar origin of neutrinos (Fig. 1.3). The total observed flux, interpreted as $\nu_e$ interactions only, remained lower than the SSM prediction. Either the Solar Standard Model was wrong, or some phenomenon was being neglected. The solution was provided by the Sudbury Neutrino Observatory (SNO) [21, 22], which used 1000 tons of heavy water ($D_2O$). Both charged-current (CC) and neutral-current (NC) interactions could be observed:

- $\nu_e\, D \rightarrow e^-\, p\, p$ (CC),
- $\nu_x\, D \rightarrow \nu_x\, n\, p$ (NC),

with $x = e, \mu, \tau$. While only $\nu_e$ contributes to CC, all flavors contribute to NC. Electrons from CC are detected via Cherenkov rings, and neutrons from NC can be captured, producing $\gamma$ photons that generate Cherenkov radiation. In the final phase, $^3$He proportional

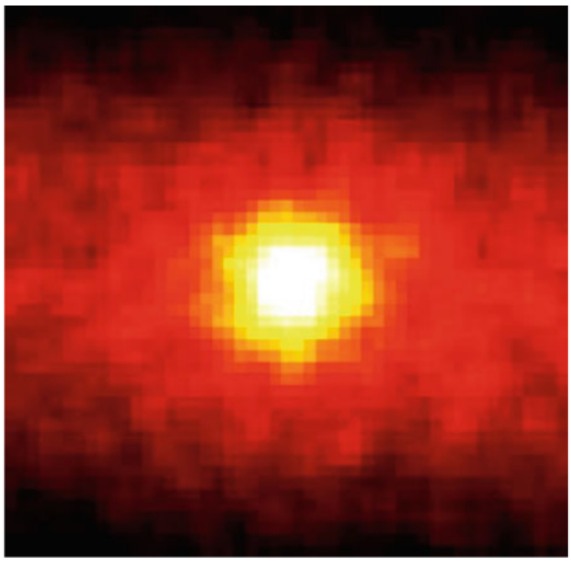


Figure 1.3: Image of the Sun obtained from a 500-day Super-Kamiokande run. The figure is taken from [3].

counters were used to independently count NC events. SNO found

$$\frac{\Phi(\nu_e)}{\Phi(\nu_x)} = 0.340 \pm 0.038 \text{ (stat. + syst.)},$$

indicating that electron neutrinos transform into other flavors on their path from the Sun. The total flux matched the SSM predictions, solving the solar neutrino problem without modifying the SSM. Neutrino flavor oscillations are possible only if the mass eigenstates (free solutions of the Dirac equation) and the flavor eigenstates (involved in the electroweak interaction Lagrangian) of neutrinos do not coincide. Conversely, the flavor eigenstates $(\nu_e, \nu_\mu, \nu_\tau)$ must arise from the mixing of the mass eigenstates $(\nu_1, \nu_2, \nu_3)$ through the lepton mixing matrix:

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = U_{PMNS} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix}, \tag{1.3}$$

where $U_{PMNS}$ is a 3x3 unitary matrix, named after Pontecorvo, Maki, Nakagawa and Sakata [23, 24].

## 1.2. Neutrino Flavor Oscillations

As previously stated, neutrinos the Standard Model assumes that neutrinos exist in three - light - flavors, namely $\nu_e$, $\nu_\mu$ and $\nu_\tau$, respectively associated to a charged lepton: the electron $e$, muon $\mu$ and tau $\tau$. The phenomenon of neutrino oscillation suggests that these neutrino flavors, which are the ones involved in the electroweak interaction Lagrangian, do not correspond to the neutrino mass eigenstates, which instead are the free-particle solution of the Dirac equation.

The mass eigenstates acquire mass through a still unknown mechanism. At this point, it is worth to introduce the distinction between Dirac and Majorana [25] particles. Dirac particles are the usual solutions of the Dirac equation. In vacuum, the Dirac equation has four independent solutions, associated to two distinct degrees of freedom: spin (up or down) and energy (particle and antiparticle). For usual fermions, such as quarks and charged leptons, this is the only possible case, since these particles carry a distinctive quantum numbers, as the electric charge. In fact, a fermion and an antifermion can be distinguished by the sign of their electric charge. For instance an electron has negative electric charge, while a positron - the electron antiparticle - has positive electric charge. For neutrinos, which have no electric charge, another possibility arises. The work of Ettore Majorana [25] highlighted the possibility that such particles could be at the same time the particle and the antiparticle. This is fully possible within the Dirac theory of fermions. To this day this is one of the most interesting, and still open, problems in physics. Many experiments are currently searching for a characteristic signal that would confirm the Majorana hypothesis: neutrino-less double $\beta$ decay [26]. This is a very rare phenomenon which is possible only if neutrinos have Majorana nature. Its detection would have huge repercussion in the world of physics. In the hypothesis that they are Dirac particles, the mass term in the Dirac Lagrangian must have the form:

$$-\mathcal{L}_D = m(\bar{\nu}_R \nu_L + \bar{\nu}_L \nu_R), \tag{1.4}$$

hence the neutrino field must exist with both chiralities, as it happens for all other fermions. However, this seems to be a rather unsatisfactory theoretical claim, founded on the assumption of the existence of right-handed neutrinos (and left-handed antineutrinos), which are coupled to no gauge field in the SM (sterile), hence with no process able to produce them. On the other hand, the Majorana hypothesis opens to possible mass acquisition mechanisms which are different from the usual Yukawa coupling. An example is the seesaw mechanism [27], where an apriori undefined number of sterile heavy neutrinos can be introduced to explain why the mass of the three light neutrinos is so small. Indepen-

dently from these questions, it is important to pay attention to the fact that if a neutrino of some flavor can oscillate into some other flavor, the conservation of the flavor lepton numbers $L_e, L_\mu, L_\tau$ is clearly not anymore valid. If one considers neutrinos to be Dirac particles, there will be a quantum number carried by $\nu_e$, $\nu_\mu$ and $\nu_\tau$ which is conserved by the interactions. This could, for example, be the total lepton number $L_e + L_\mu + L_\tau$. If instead neutrinos turn out to be Majorana particles, lepton number conservation would be violated anyway. In the following subsections, a brief theoretical introduction to the oscillation phenomenon is presented, initially restricted to a two flavor framework, and then extended to three flavors. This choice has been made in the interest of simplicity, but also because, as it will be explained, there are contexts in which neutrino oscillations can be approximately treated as if they were only involving two flavors. Moreover, no degeneracy between mass eigenstates is assumed.

### 1.2.1. Two-flavor Oscillations

Considering a two flavor scenario (suppose for the moment they are $\nu_e$, $\nu_\mu$) and two mass ($\nu_1$, $\nu_2$) eigenstates, the flavor eigenstates can be written as a function of a single real mixing angle $\theta$ as:

$$\begin{pmatrix} \nu_e \\ \nu_\mu \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix}. \tag{1.5}$$

Then, for instance, if a $\nu_e$ is produced at time $t = 0$ and position $x = 0$, the space–time evolution of this quantum state $\psi$ will be determined by the mixing of the corresponding mass eigenstates:

$$|\psi(t)\rangle = |\nu_1\rangle \cos\theta e^{-i(E_1 t - \boldsymbol{p}_1 \cdot \boldsymbol{x})} + |\nu_2\rangle \sin\theta e^{-i(E_2 t - \boldsymbol{p}_2 \cdot \boldsymbol{x})}, \tag{1.6}$$

where the exponential dependence comes from the fact that the neutrino mass eigenstates are free solutions of the Dirac equation. Moreover, $\psi(\boldsymbol{x} = 0, t = 0) = \nu_e$ as a boundary condition. The interest is in computing:

$$P(\nu_e \to \nu_\mu) = |\langle \nu_\mu | \psi(t) \rangle|^2, \tag{1.7}$$

meaning the probability that electron neutrino produced at time t = 0, oscillates into a muon neutrino after some time t. Considering the ultra-relativistic limit (UR):

$$E_j \sim p_j + \frac{m^2}{2p_j} \sim E_j + \frac{m^2}{2E_j}, \tag{1.8}$$

and along with:

$$m_1 << E_1, \tag{1.9}$$

$$m_2 << E_2, \tag{1.10}$$

the two eigenstates have approximately the same energy $E_\nu$ (detected energy) and same momentum $p_\nu$. Hence, it is possible to write:

$$E_1 \sim E_\nu + \frac{m_1^2}{2E_\nu}, \quad E_2 \sim E_\nu + \frac{m_2^2}{2E_\nu} \tag{1.11}$$

such that:

$$\Delta E = E_2 - E_1 = \frac{m_2^2 - m_1^2}{2E_\nu} = \frac{\Delta m^2}{2E_\nu}, \tag{1.12}$$

where the mass appears only as a UR correction. Since:

$$\langle \nu_\mu | = -\sin\theta \langle \nu_1 | + \cos\theta \langle \nu_2 |, \tag{1.13}$$

the scalar product of interest, takes the form:

$$\langle \nu_\mu | \psi(t) \rangle = -\sin\theta\cos\theta e^{-iE_1 t} + \sin\theta\cos\theta e^{-iE_2 t} = \sin\theta\cos\theta \left(e^{-i\Delta E t} - 1\right), \tag{1.14}$$

leading to:

$$\begin{aligned} P(\nu_e \to \nu_\mu) &= |\sin\theta\cos\theta \left(e^{-i\Delta E t} - 1\right)|^2 = \\ &= |\sin\theta\cos\theta \left(2i\sin\frac{\Delta E t}{2}\right)|^2 = \sin^2(2\theta)\sin^2\left(\frac{\Delta E t}{2}\right). \end{aligned} \tag{1.15}$$

Expressing $\Delta E = \frac{\Delta m^2}{2E_\nu}$, and realizing that for a UR particle in natural units, $L \sim t$, one can write:

$$P(\nu_e \to \nu_\mu) = \sin^2(2\theta)\sin^2\left(\frac{\Delta m^2 L}{4E_\nu}\right), \tag{1.16}$$

which can also be written in practical units as:

$$P(\nu_e \to \nu_\mu) = \sin^2(2\theta)\sin^2\left(1.27\frac{\Delta m^2(eV^2)L(km)}{E_\nu(GeV)}\right). \tag{1.17}$$

In order to have oscillation (mixing) the two masses must be different. It is possible to identify two overall quantities:

- $\sin^2(2\theta)$ as the oscillation amplitude
- $\frac{\Delta m^2 L}{4E_\nu}$ as the oscillation phase

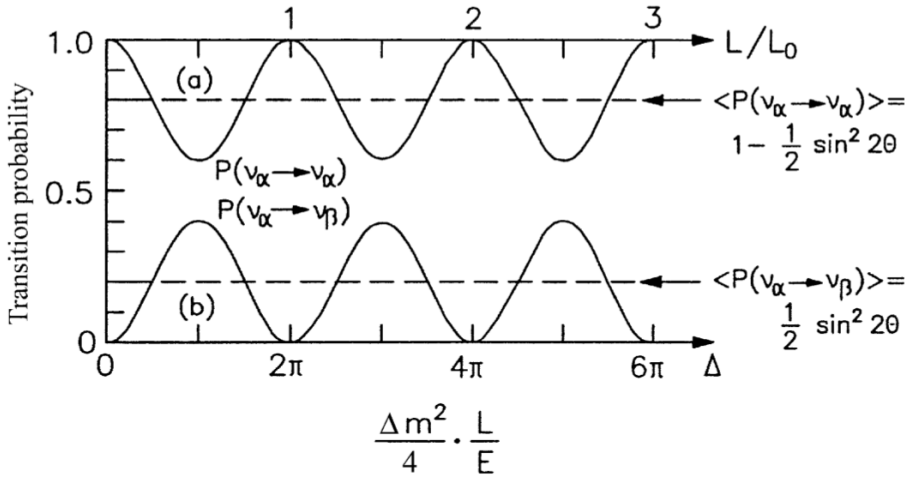



Figure 1.4: Example of neutrino oscillations in the two-flavour scheme. The dashed lines show the average oscillation probabilities. The figure is taken from [28].

A too small or too large phase makes the measurement of the oscillation parameters $(\theta, \Delta m)$ very difficult. The so-called sensitivity to a given mass splitting $|\Delta m^2|$ of an experiment is the value of $|\Delta m^2|$ for which:

$$\frac{\Delta m^2 L}{4E_\nu} \simeq 1.27 \left(\frac{\Delta m^2}{1eV^2}\right)\left(\frac{L}{1km}\right)\left(\frac{1GeV}{E_\nu}\right) \overset{!}{\simeq} 1. \tag{1.18}$$

In this way, it is possible to classify different neutrino oscillation experiments based on the average value of $L/E$, which determines their sensitivity in terms of $\Delta m^2$. It is also useful to define the quantity:

$$L_\nu = \frac{2\pi E_\nu}{\Delta m^2}, \tag{1.19}$$

which is also depending on the sign of $\Delta m^2$. In this way, the probability $P(\nu_e \to \nu_\mu)$ is maximal for $L \sim nL_\nu$.
Clearly, one can also write the survival probability of the neutrino, in our case:

$$P(\nu_e \to \nu_e) = 1 - P(\nu_e \to \nu_\mu). \tag{1.20}$$

. The disappearance and survival probability in the two flavor scenario are shown in Fig. 1.4. Experiments that search for neutrinos with a flavor different from the flavor of the initial neutrino beam are called *appearance* experiments. Experiments that measure the possible depletion of the initial neutrino beam are called *disappearance* experiments [3].
An appearance experiment is basically sensitive to a given oscillation channel $\nu_\alpha \to \nu_\eta$

with $\alpha \neq \eta$, while a disappearance experiment is sensitive to transitions to all possible different neutrino species, or to pure disappearance. This statement is true also in a three flavor scenario.
The determination of the parameters of neutrino oscillations has been one of the priorities of research during the recent years.

### 1.2.2. Three-flavor Oscillations

Assuming three weak eigenstates and three mass eigenstates, the mixing can be modeled, using the $3 \times 3$ unitary matrix introduced before, the Pontecorvo–Maki–Nakagawa–Sakata (PMNS) matrix:

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = \begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix}. \tag{1.21}$$

Taking into account the relations imposed by unitarity and the fact that several phases can be absorbed in the definition of the fields (if the neutrinos are Dirac fermions) there are only three real parameters usually chosen as the mixing angles $\theta_{12}, \theta_{13}, \theta_{23}$ and a single complex phase written in the form $e^{i\delta_{CP}}$.
The mixing phase $\delta_{CP}$ is used to measure the degree of $\boldsymbol{CP}$ violation in neutrino oscillations. Very interestingly, if $\boldsymbol{CP}$ is violated in the neutrino sector, as well as what is already known in the quark sector, there could be an explanation for the matter-antimatter asymmetry observed in the universe; indeed the degree at which $\boldsymbol{CP}$ is violated in the SM is still not enough to agree with experimental data.
The PMNS matrix can be decomposed as the product of $3\times3$ matrices:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{23} & \sin\theta_{23} \\ 0 & -\sin\theta_{23} & \cos\theta_{23} \end{pmatrix} \begin{pmatrix} \cos\theta_{13} & 0 & \sin\theta_{13} e^{-i\delta_{CP}} \\ 0 & 1 & 0 \\ -\sin\theta_{13} e^{i\delta_{CP}} & 0 & \cos\theta_{13} \end{pmatrix} \begin{pmatrix} \cos\theta_{12} & \sin\theta_{12} & 0 \\ -\sin\theta_{12} & \cos\theta_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

From this decomposition, if $\theta_{13}$ is very small, both the $\nu_e \to \nu_\mu$ and $\nu_\mu \to \nu_\tau$ can be described as approximate two-flavor oscillations, with parameters $\theta_{12}$, $\Delta m^2_{21}$ and that $\theta_{23}$, $\Delta m^2_{32}$.
An important prediction is the so-called *mass sum rule*, establishing that:

$$\Delta m^2_{21} - \Delta m^2_{31} + \Delta m^2_{32} = 0. \tag{1.22}$$

It is useful to briefly summarize the current experimental sensitivities to the neutrino oscillation parameters. Different experiments probe different aspects of the three-flavor oscillation framework, as summarized in Table 1.1.

Table 1.1: Sensitivity of the main neutrino experiments to oscillation parameters.

| Category | Examples | Main parameters |
|---|---|---|
| Solar | SNO [21, 22], Super-Kamiokande [10] | $\theta_{12}$, $\Delta m^2_{21}$ |
| Accelerator | T2K [29], NOvA [30] | $\theta_{23}$, $\lvert\Delta m^2_{3\ell}\rvert$, hints on $\delta_{CP}$ |
| Reactor | Daya Bay [31], RENO [32] | $\theta_{13}$, contribution to $\lvert\Delta m^2_{3\ell}\rvert$ |
| Atmospheric | IceCube/DeepCore [33] | $\theta_{23}$, mass ordering |

Taken together, these complementary sensitivities allow the global fits to provide a coherent picture of the three-flavor oscillation parameters, while also highlighting remaining unknowns such as the absolute mass scale and the precise value of $\delta_{CP}$.

### 1.2.3. Oscillations in Matter

In previous subsections, it was assumed that neutrinos propagate in vacuum between the source and the detector. Since neutrinos interact only weakly, it might seem that this is always an excellent approximation. Nevertheless, the effect of matter is very important to account when considering solar neutrino data, indeed an important part of their path to Earth is traveled inside the Sun [3]. Moreover, it must be taken into account when considering very high precision measurements of the oscillation parameters, such as the ones which are expected to be taken from the JUNO experiment [34–36]. In the same way that light traveling through a transparent medium can have its refractive index changed, so can a neutrino. In particular, such refractive index can be different for $\nu_e$ and $\nu_\mu$. The difference in refractive indices is determined by the difference in the real parts of the forward $\nu_e e^-$ and $\nu_\mu e^-$ elastic scattering amplitudes. The essential point is that the scattering can be coherent, with the spins and momenta of the particles remaining unchanged. In this condition (for low energy incoherent scattering can be neglected), the effect will be proportional to the density of electrons in matter, $n_e$. The scattering amplitude, instead, is proportional to $G_F$, so that a figure of merit for the effect is given by the product $G_F n_e$. The neutrino oscillations can be enhanced (or suppressed) whenever neutrinos travel through matter. In fact, while all neutrino flavors interact equally with matter through neutral currents, charged-current interactions with matter are flavor dependent (at solar neutrino energies, basically only electron neutrinos can interact). This is called the MSW effect, as it comes from works by Lincoln Wolfenstein [37], Stanislav

Mikheyev, and Alexei Smirnov [38]. Considering a constant density medium, the matter effect is translated, in a two-flavor approximation, into a modified oscillation probability:

$$P(\nu_e \to \nu_x) = \sin^2(2\theta_m)\sin^2(\frac{\pi L}{2L_\nu}F), \tag{1.23}$$

where

$$\sin(2\theta_m) = \frac{\sin(2\theta)}{F}, \tag{1.24}$$

$$F = \sqrt{\left(\cos(2\theta) - \frac{L_\nu}{L_e}\right) + \sin^2(2\theta)}, \tag{1.25}$$

$$L_e = \pm\frac{2\pi}{2\sqrt{2}G_F n_e}. \tag{1.26}$$

$L_e$, the electron neutrino interaction length, is positive for neutrinos, negative for antineutrinos. $L_\nu$, the neutrino oscillation length in vacuum, is, as defined before. The values of the mass eigenstates are also changed. The new eigenstates are given by:

$$M^2_{2,1} = \frac{1}{2}\left(m_1^2 + m_2^2 + \Delta m^2(\frac{L_\nu}{L_e} \pm F)\right). \tag{1.27}$$

Whenever $L_\nu = L_e\cos(2\theta)$ the amplitude of the oscillation is maximal - $\sin^2(2\theta_m) = 1$. Thus, for a given set of values $(E_\nu, n_e)$, resonant oscillations are possible and the oscillation probability may be strongly enhanced independently of the value of $\theta$ in vacuum.

## 1.3. Neutrino Mass Ordering

Despite significant progress in neutrino physics, several properties remain unknown, including the Dirac or Majorana nature, CP violation in the leptonic sector, and the neutrino mass ordering (MO). Oscillation experiments are sensitive only to squared mass differences $\Delta m^2_{ij}$, not absolute masses. While solar neutrino data fix $\Delta m^2_{21} > 0$, the sign of $\Delta m^2_{3\ell}$[2] is still unknown, leading to two possibilities: normal ordering (NO, $\Delta m^2_{31} > 0$) and inverted ordering (IO, $\Delta m^2_{32} < 0$). A representation of the neutrino mass ordering possibilities is shown in Fig. 1.5.

Determining the MO requires precise measurements of $\Delta m^2_{31}$ and $\Delta m^2_{32}$, achievable via long-baseline accelerator experiments (e.g., NO$\nu$A, DUNE), atmospheric neutrino experiments (e.g., PINGU), and medium-baseline reactor experiments like JUNO. Matter effects create differences in oscillation probabilities between neutrinos and antineutrinos, enhancing sensitivity to MO in appearance channels, while JUNO can resolve the ordering

[2] $\Delta m^2_{3\ell}$ stands for the mass splitting with the largest absolute value, thus $\Delta m^2_{31}$ for NO, $\Delta m^2_{32}$ for IO.

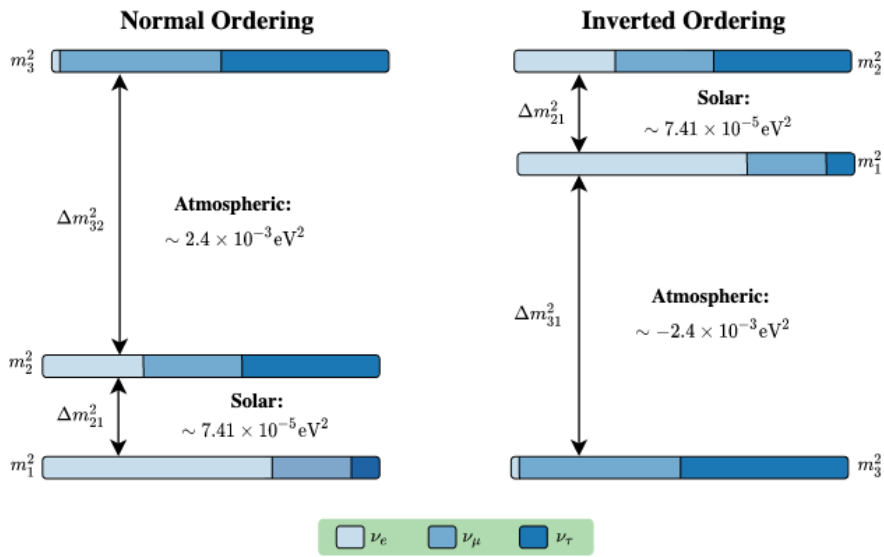



Figure 1.5: Schematic illustration of the neutrino mass ordering. The figure is taken from [36] and refers to non-updated oscillation parameters.

through high-precision spectral analysis of reactor antineutrinos [3, 30, 34]. In order to provide a clear overview of the present knowledge on neutrino oscillations, the current best-fit values oscillation parameters, as reported by the Particle Data Group (PDG), are summarized in Table 1.2.

Table 1.2: Current best-fit values and uncertainties of neutrino oscillation parameters (three-flavor framework). Source: PDG [39].

| Parameter | Best-fit value (NO) | Best-fit value (IO) |
|---|---|---|
| $\frac{\Delta m_{21}^2}{10^{-5}\,\mathrm{eV}^2}$ | $7.50 \pm 0.19$ | $7.50 \pm 0.19$ |
| $\frac{\Delta m_{32}^2}{10^{-3}\,\mathrm{eV}^2}$ | $2.451 \pm 0.026$ | $-2.527 \pm 0.034$ |
| $\sin^2\theta_{12}$ | $0.307 \pm 0.012$ | $0.307 \pm 0.012$ |
| $\sin^2\theta_{13}$ | $0.0216 \pm 0.0006$ | $0.0216 \pm 0.0006$ |
| $\sin^2\theta_{23}$ | $0.534^{+0.015}_{-0.019}$ | $0.537 \pm 0.020$ |
| $\delta_{CP}$ | $\sim 197°$ | $\sim 197°$ |

# 2 | The JUNO Experiment

The Jiangmen Underground Neutrino Observatory (JUNO) is a multi-purpose underground liquid scintillator detector, proposed in 2008 to measure, as a priority, the neutrino mass ordering with reactor antineutrinos ($\bar{\nu}_e$). Moreover, it will enable new measurements of oscillation parameters with previously never achieved accuracy [35]. Its construction in South China has been completed, with the beginning of data taking in august 2025. The expected large volume of the JUNO detector, excellent energy resolution and vertex reconstruction capability, offer also many opportunities to study other relevant topic to neutrino and asto-particle physics. Indeed the JUNO detector can observe not only the reactor antineutrinos, but also neutrinos/antineutrinos from terrestrial and extra-terrestrial sources, such as geoneutrinos, solar neutrinos, atmospheric neutrinos, supernova burst neutrinos and diffuse supernova neutrino background [34]. The main goal of the JUNO experiment is to reach a $3\sigma$ significance on neutrino mass ordering in about 6-7 years of data-taking. Recent studies on JUNO's sensitivity to oscillation parameters [40] shows that the experiment will be able to estimate the $\Delta m^2_{21}$, $\Delta m^2_{31}$, $\sin^2\theta_{12}$ parameters to a precision of 0.2%, 0.3%, and 0.5%, after 6 years of exposure, while reaching sub-percent precision within the first year of data taking.

## 2.1. Overview of Experimental Facilities

The JUNO detector is located at a baseline of about 52.5 km from the Taishan and Yangjiang nuclear power plants (NPPs), that will provide the reactor antineutrino spectrum to be measured [34]. The baseline was chosen to optimize the MO determination and to avoid cancellation of oscillation dephasing effect, requiring the same distance between reactor cores and the detector. The detector is placed at the first $\bar{\nu}_e$ disappearance maximum that is driven by $\Delta m^2_{21}$ while measuring simultaneously the oscillation pattern from both $\Delta m^2_{32}$ and $\Delta m^2_{21}$ [35].

Yangjiang NPP is composed of six cores of 2.9 $\mathrm{GW_{th}}$ of thermal power each. Every core is 2nd generation pressurized water reactor (PWR) CPR1000, a derivative of the Framatome M310, with improvements on safety, design and refueling. Taishan NPP was expected to

have 4 EPR reactor cores with a thermal power of 4.59 $GW_{th}$ each, however only two of them have been completed yet. The total power is expected to be 26.55 $GW_{th}$ when JUNO begins data taking [34].
The high energy resolution required for the JUNO purposes, relies on the precise knowledge of the non-oscillated antineutrino spectrum. The Taishan Antineutrino Observatory (TAO) [41], a smaller, ton level, scintillator detector, will be placed at around 42 m from one of the Taishan NPP cores with two main purposes: 1) provide a reference non-oscillated spectrum for JUNO with sub-percent energy resolution to reduce the possible model dependence in mass ordering determination, due to the antineutrino spectrum fine structure; 2) provide a benchmark useful to validate current nuclear databases, exploiting the comparison between the measurement and the prediction obtained with the summation method [42]. The theoretical prediction of the reactor antineutrino spectrum using the summation method is the main topic of this thesis.
At a distance of 215 km, the Daya Bay complex is located, where spread over 1.1 km, Daya Bay NPP, Ling Ao NPP, and Ling Ao-II NPP are found, each with two cores of 2.9 $GW_{th}$ [34].In 2018, Daya Bay provided the measurement of a reactor antineutrino spectrum, allowing model-independent predictions of reactor antineutrino spectra [31]. A schematic illustration of the location JUNO and TAO site is shown in Fig. 2.1.

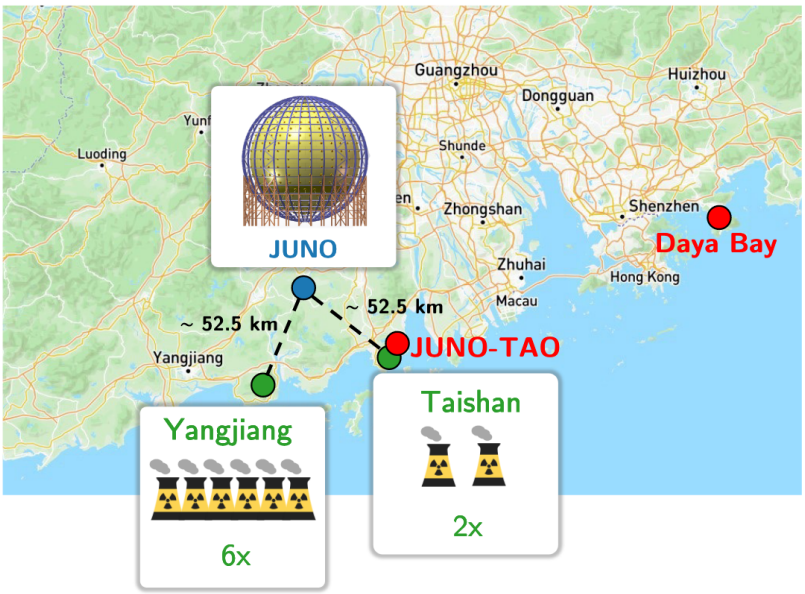



Figure 2.1: Location of the JUNO and TAO experimental site. The main 20 kton LS detector is indicated in blue [34], at a baseline of around 52.5 km from the Yangjiang NPP and the Taishan NPP. The 2.8 ton TAO detector [41], indicated in red, is located about 30 meters away from one of the Taishan reactor cores.

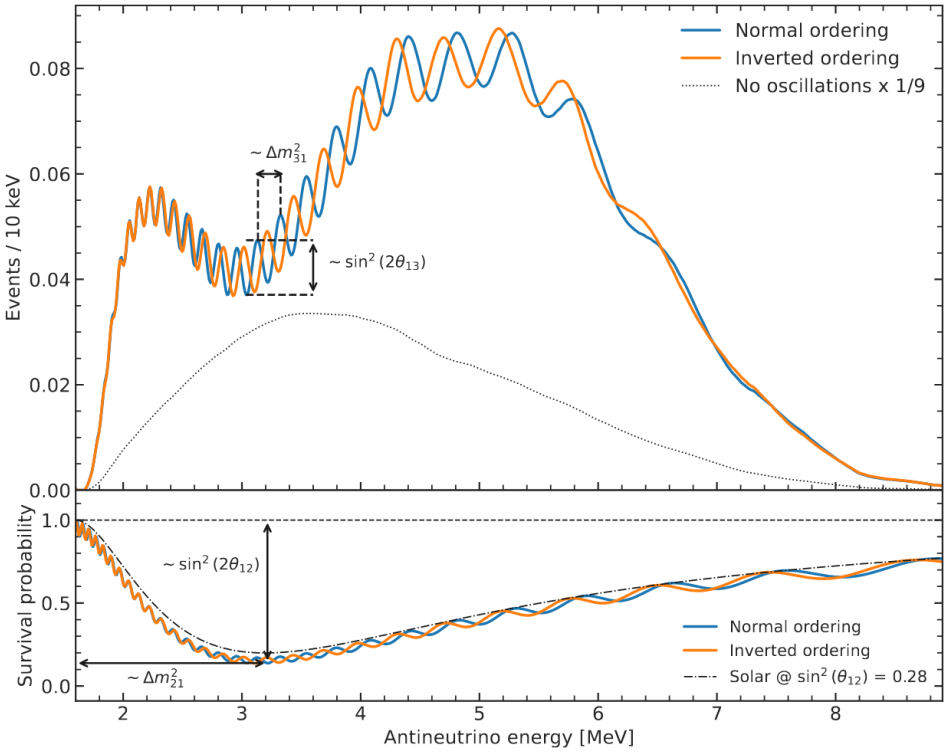



Figure 2.2: Top: Expected oscillated antineutrino spectrum at JUNO without considering detector effect. The non-oscillated spectrum is also shown, in gray. Bottom: $\bar{\nu}_e$ survival probability at a baseline of 52.5 km. The figure is taken from [36].

At JUNO's location, both a low- and high-frequency distortion of the antineutrino spectrum can be observed, respectively driven and modulated by $\Delta m^2_{21}$, $\sin^2(2\theta_{12})$ and $\Delta m^2_{31}$, $\sin^2(2\theta_{13})$. The effect is shown in Fig. 2.2. JUNO will be the first experiment to simultaneously observe neutrino oscillations from two different frequencies; moreover it will be able to observe multiple oscillation cycles due to atmospheric mass splitting $\Delta m^2_{31}$. The oscillated spectrum depends on the neutrino mass ordering, providing an important sensitivity to discriminate between normal or inverted ordering.

## 2.2. The JUNO detector design

The JUNO detector is found in an underground laboratory under approximately 690 m of rock, i.e. 1800 meters-water-equivalent (m.w.e.) and at an average baseline of 52.5 km from the two nearest NPPs. It can be subdivided in three fundamental parts: the Central Detector (CD), the Water Cherenkov Detector and the Top Tracker (TT).
The central detector consists of a 35.4 m diameter spherical acrylic vessel filled with 20 kton of scintillating liquid (LS) acting as a target. It is inserted in a cylindrical water

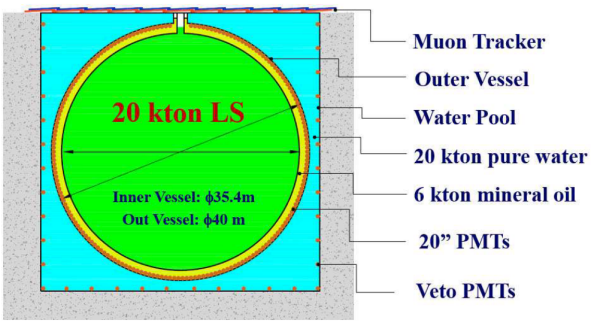



Figure 2.3: A schematic view of the JUNO detector design.

pool, resulting in a shielding of the central detector with at least 2 m of water from every direction, in order to remove the surrounding rock radioactivity. The acrylic vessel is supported by a stainless steel (SS) structure via Connecting Bars. The CD Photomultiplier Tubes (PMTs) are installed on the inner surface of the SS structure. Cherenkov light from cosmic muons, acting as a veto signal, is detected with PMTs in the water pool. On top of the water pool, another muon detector - the Top Tracker - is placed to accurately measure the muon tracks. The JUNO detector will be the largest liquid scintillator-based detector in the World for the next decade [34].
This detector is designed to have an energy resolution of $3\%/\sqrt{E(MeV)}$. Previous measurements such of reactor antineutrino spectra, as the one performed at Daya Bay, achieved a spectral uncertainty *between 1.5 and 7 MeV, it ranges from 1.0% at 3.5 MeV to 6.7% at 7 MeV, and above 7 MeV it is larger than 10%* [31]. The high resolution of JUNO is required for the mass ordering determination, since it is expected to distinguish hyperfine structure effects present in the reactor antineutrino spectrum. A schematic view of the JUNO detector is shown in Fig. 2.3. In order to reach the fixed purposes of the experiment, strict requirements have to be satisfied, including:

- large number neutrino detection events to increase statistical significance,
- high light yield from the LS to efficiently detect neutrino interactions, and achieve high energy resolution,
- control of the energy scale to enable resolution of the high frequency oscillations,
- management and minimization of background signals in order to achieve an high signal-to-noise ratio $S/N$,
- careful analysis of systematic uncertainties to avoid possible overall biases in the measurements.

In the following, these constraints are addressed with respect to the various sub-detector

systems.

### 2.2.1. Central Detector

The JUNO Central Detector consists of a 20 kton liquid scintillator contained in a spherical acrylic vessel with an inner diameter of 35.4 m and 120 mm wall thickness. The vessel is supported by a stainless steel shell of 40.1 m diameter through 590 Connecting Bars, which rests on 30 pairs of supporting legs anchored to the bottom of the water pool. The shell structure and its bearing are called the Main Structure and is shown in Fig. 2.4. The

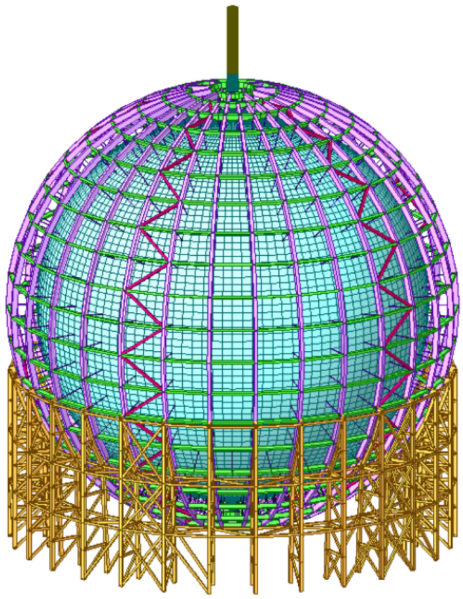


Figure 2.4: Scheme of the Central Detector (CD). The spherical Acrylic Vessel is supported by 590 Connecting Bars, constituting a spherical SS structure. The shell structure sits on a bearing consisting of 30 pairs of Supporting Legs made of stainless steel structures.

Main Structure hosts 17,612 20-inch PMTs and 25,600 3-inch PMTs pointing inward to collect scintillation light, and 2,400 20-inch PMTs facing outward for the water Cherenkov veto. The CD and the Water Cherenkov Detector are optically separated [35]. The acrylic, produced with a dedicated clean process, achieves a transparency above 96% in water and a radio-purity below 0.5 ppt U/Th, while the supporting nodes withstand loads above 100 t. The shell structure, reinforced with longitudinal and latitudinal beams, is able to sustain a buoyant force of about 3000 t and is designed to resist earthquakes up to 46.5 Gal. Since the system is expected to operate reliably for 30 years, all surfaces must be kept ultra-clean and LS-compatible to avoid radon contamination. A dedicated filling, overflow and circulation system with 50 $m^3$ tanks stabilizes the liquid level and ensures continuous purification. In addition, the main structure supports the front-end electronics, cabling,

and an anti-geomagnetic field coil to shield the PMTs from the Earth's magnetic field and preserve their performance [35].

### 2.2.2. Veto Detectors

The veto system of JUNO is composed by the Water Cherenkov Detector (WCD) and the Top Tracker (TT), working together to suppress backgrounds due to natural radioactivity and cosmic muons. The WCD is a 43.5 m diameter, 44 m high cylinder containing 35 kton of ultrapure water and instrumented with 2,400 20-inch Microchannel Plate (MCP) PMTs mounted on the Main stainless-steel Structure. Tyvek reflective foils are employed to enhance light collection, while the water provides a strong passive shielding against gamma radiation from the surrounding rock. A sophisticated purification system, producing 100 t/h at the surface and circulating underground, maintains water purity and a stable temperature of (21 ± 1 °C) around the acrylic vessel, compensating for the approximately 260 kW heat load from electronics. Radon contamination, initially at 10 Bq/m$^3$, is reduced to below 0.2 Bq/m$^3$ with a multi-stage degassing membrane system, enhanced by nitrogen micro-bubbles, achieving levels of ∼10 mBq/m$^3$. Additional protection is ensured by a 5 mm HDPE liner on the pool walls and a gas-tight cover placed 1 m above the water surface, filled with high-purity nitrogen under slight overpressure to prevent radon penetration [35]. Muon detection is optimized through the combined action of WCD and TT. The WCD achieves a tagging efficiency of 99.5%, with a local trigger logic that minimizes noise while maintaining an high efficiency. The TT, which is made of plastic scintillator strips reused from the OPERA experiment [43], covers the detector in three horizontal layers with overlapping modules, reaching an efficiency of (98.0 ± 0.5)% per module and a median angular resolution of 0.2°. Its geometry ensures effective vetoing of muons entering through the chimney region, complementing the WCD coverage. Together, the two systems provide a well-reconstructed muon sample with >99% purity, crucial for cosmogenic background rejection and calibration. Finally, to counter the geomagnetic field, which would otherwise reduce PMT efficiency by up to 60%, a system of 32 circular coils surrounds the detector. This reduces the residual field to <0.05 G in the Central Detector PMT region and <0.1 G in the WCD, ensuring optimal performance of both veto systems over the 30-year JUNO lifetime [35].

### 2.2.3. JUNO Liquid Scintillator

The JUNO liquid scintillator (LS) is composed of purified Linear Alkyl Benzene (LAB) with 2.5 g/L of 2,5-diphenyloxazole (PPO) and 3 mg/L 1,4-bis(2-methylstyryl)benzene (bis-MSB). It is optimized for both high light yield and long optical attenuation length.

The impurities of PPO and bis-MSB act as waveshitfers in the scintillator. Reactor and solar neutrino experiments generally demand a very low radioactive background; for this reason, a dedicated purification chain including alumina filtration, vacuum distillation, water extraction and gas stripping has been built to reduce the U/Th contamination at a level of $10^{-15} - 10^{-17}$ g/g. Radon and airborne impurities are suppressed through a tight sealing of all the system, which is also electro-chemically polished and constantly blanketed with high-purity nitrogen. All the strategies are able to guarantee that JUNO will maintain the optical performance required by the experiment, while minimizing backgrounds, therefore meeting the stringent sensitivity goals of the experiment [35].

## 2.3. IBD Reaction and Detection

The reactor antineutrinos will be detected by JUNO through the inverse beta decay (IBD) reaction on protons in the LS, as shown in Fig. 2.5.

The electron antineutrino ($\bar{\nu}_e$) interacts with a proton ($p$) in the LS, creating a positron

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

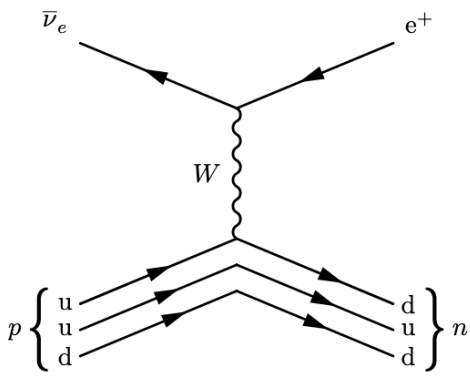



Figure 2.5: Feynman diagram of the inverse beta decay reaction on protons.

($e^+$) and a neutron ($n$). The positron retains almost all of the antineutrino energy, and quickly deposits it inside the LS. After thermalization, the positron annihilates with a surrounding electron, and two 0.511 MeV gamma rays are produced. The final result is a prompt signal proportional to the sum of the positron kinetic energy $T_{e^+}$ and the 1.022 MeV annihilation energy. The positron emission is almost isotropic, with a small preference for the backward direction [44]. On the other hand, the neutron retains most of the antineutrino momentum, resulting in a recoil direction parallel to the antineutrino incidence. The neutron then thermalizes in the scintillator, and is absorbed by a free proton or by a carbon nucleus, respectively with an approximate 99% or 1% probability. Subsequently there is the emission a 2.22 MeV or 4.95 MeV gamma ray, respectively, giving rise to a delayed signal. The time interval between the prompt and delayed signal is

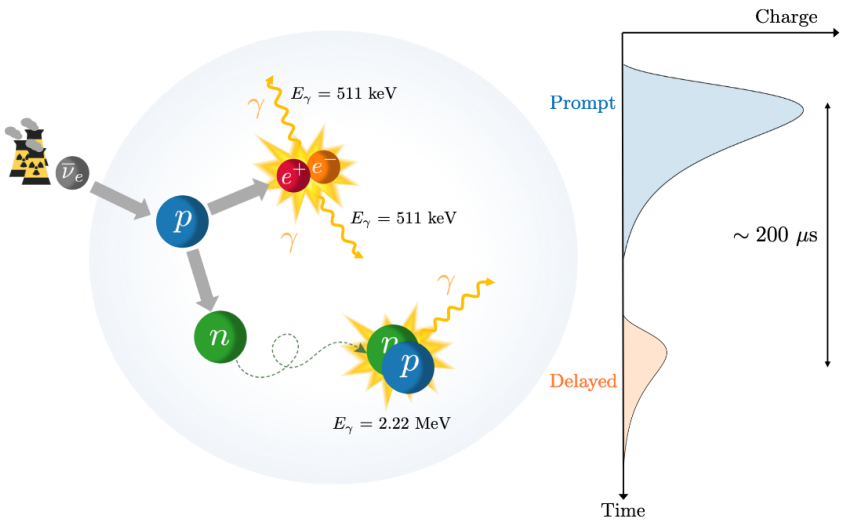



Figure 2.6: Schematic illustration of the IBD kinematics. The time-charge diagram is not in scale. The figure is taken from [36].

approximately 200 $\mu$s. The detection of IBD events in JUNO relies on the time coincidence between the prompt signal from the positron and the delayed signal from the neutron capture, allowing a clear identification of reactor antineutrinos over background signals. A scheme summarizing the reaction kinematics is proposed in Fig. 2.6.
Using the IBD reaction the electron antineutrino energy is then:

$$E_{\bar{\nu}_e} \simeq T_{e^+} + T_n + \Delta_{np} + m_{e^+}, \tag{2.1}$$

where $T_{e^+}$ and $T_n$ are the positron and neutron kinetic energies, $\Delta_{np} = m_n - m_p = 1.293$ MeV is the mass difference between neutron and proton, and $m_{e^+}$ is the positron mass. The neutron recoil energy goes from 0 to some tens of keV, thus is often neglected with respect to the MeV scale energy transferred to the positron. In this *recoil-less* approximation, the IBD kinematic threshold results to be at around 1.8 MeV for the $\bar{\nu}_e$.
Accounting for the deposition of the annihilation energy, the energy deposited $E_{dep}$ in the LS per interaction is given by:

$$E_{dep} \simeq E_{\bar{\nu}_e} - 0.782 \text{ MeV}. \tag{2.2}$$

The latter quantity is assumed to be fully absorbed by the detector, in fact, less than 1% of the IBD events is affected by energy losses due to escaping secondary particle generated either by Compton scattering or by pair production. Before the detection, the oscillation of the reactor antineutrino spectrum must be accounted. The survival probability of

electron antineutrinos is given by [1]:

$$P(\bar{\nu}_e \to \bar{\nu}_e) = 1 - \sin^2 2\theta_{12}\, c_{13}^4 \sin^2 \Delta_{21} - \sin^2 2\theta_{13} \left( c_{12}^2 \sin^2 \Delta_{31} + s_{12}^2 \sin^2 \Delta_{32} \right), \quad (2.3)$$

where $c_{ij} \equiv \cos\theta_{ij}$, $s_{ij} \equiv \sin\theta_{ij}$ and $\Delta_{ij} \equiv \frac{\Delta m_{ij}^2 L}{4E}$. The expression highlights the dependence of the survival probability on the solar and reactor mixing angles, as well as the squared mass differences, while $\theta_{23}$ and the CP-violating phase $\delta_{CP}$ do not affect it. Finally, the positron spectrum measured by the JUNO detector is directly linked to the oscillated antineutrino spectrum, shifted in energy and weighted by the IBD differential cross-section $\frac{d\sigma}{d\cos\theta}(E_{\bar{\nu}_e}, \cos\theta)$, which has a dependence both on the neutrino energy and on the scattering angle. In this kind of experiments, where IBD is used to detect electron antineutrinos, the $\bar{\nu}_e$ disappearance channel is the only one accessible at MeV energies. Higher energy channels producing muons or taus are indeed kinematically forbidden.

[1] Note that, as a consequence of the ***CPT*** theorem: $P(\bar{\nu}_e \to \bar{\nu}_e) = P(\nu_e \to \nu_e)$.

# 3 | Reactor Antineutrino Spectrum

An accurate description of the electron antineutrino ($\bar{\nu}_e$) flux emitted by nuclear reactors is fundamental for the JUNO experiment. The non-oscillated antineutrino spectrum propagates for about 52.5 km, undergoes flavor oscillation following Eq. 2.3, to be then measured by the JUNO detector, where the IBD cross section and the detector response will determine the final positron spectrum. High accuracy is required in the latter to investigate the neutrino mass ordering. Clearly, such accuracy will be affected by our knowledge of the non-oscillated reactor neutrino spectrum.

In particular, JUNO measures the convolution of the antineutrino signals coming from 8 different reactors. Therefore, a common non-oscillated spectrum is required in order to perform the fit on the oscillation parameters. From the modeling point of view, this requires to predict the spectrum of each reactor, and then evaluate their sum. The problem is that, as it will be presented in Sec. 3.3, models present various bias.

For this reason, the TAO [41] detector will be placed at around 40 m from one of the Taishan NPP nuclear reactors to measure with unprecedented energy resolution the non-oscillated $\bar{\nu}_e$ spectrum. However TAO has some limitations. The reactor antineutrino spectrum indeed depends on the operation conditions of the reactor itself, that is the actinide fission fractions. These are different from reactor to reactor.

In this context, computational tools play a relevant role in predicting the reactor antineutrino spectra. After validation with experimental measurements, like the one of Daya Bay [31] and TAO [41], such tools can be used to compute the spectra of interest relying only on the reactors operational data. [45]. It will be possible then to apply this approach to the second Taishan core and the six units of the Yangjiang plant, where near detectors are not available. In this sense, the high-resolution data from TAO would serve as a benchmark for the accuracy of nuclear databases.

This thesis deals with this last point. The question is how is it possible to improve reactor

antineutrino spectrum modeling, focusing in particular on the summation method, presented in Sec. 3.2. A positive answer to this question would be of fundamental importance not noly in the context of the JUNO experiment, but also for future ones. The validation of nuclear databases can provide ready to use models for the prediction of reactor antineutrino spectra, without the need of a near detector such as TAO.

## 3.1. Antineutrino Production in Nuclear Reactor Cores

Taishan and Yangjang NPPs host multiple pressurized water reactors (PWRs), a commercial reactor technology based on thermal fission induced by neutrons. The fission chain reaction takes place when in the overall neutronic balance of the reactor, the number of neutrons lost in absorption, fission and leakage is balanced by the number of neutrons emitted as a consequence of fission itself. At the beginning of the operational life of the fuel, the core is largely made of uranium dioxide ($UO_2$), enriched in $^{235}$U with concentrations that can range from 2 to 5 wt%. In this phase, the fission chain reaction is sustained only by uranium isotopes present in the core, namely $^{235}$U and $^{238}$U. When irradiation time increases, neutron captures by uranium isotopes and their subsequent radioactive decays progressively produce transuranic isotopes in the reactor core. In particular, $^{239}$Pu and $^{241}$Pu start to get involved in the chain of thermal neutron-induced fissions as well. The byproduct of fission are neutron-rich nuclei called fission fragments, that undergo a sequence of $\beta^-$ decays to reach stability. In $\beta^-$ decay, a neutron transforms into a proton, with the emission of two leptons, an electron and an antineutrino:

$$n \rightarrow p + e^- + \bar{\nu}_e. \tag{3.1}$$

An average of about six electron antineutrinos per fission is emitted consequently [45]. As explained above, in commercial PWRs the antineutrino flux is mainly due to the decay of the fission products of four key actinides: $^{235}$U, $^{238}$U, $^{239}$Pu and $^{241}$Pu; these nuclides account for around the 99% of the thermal power and of the total antineutrino flux relevant to JUNO, namely above the 1.8 MeV threshold [35] (see Sec. 2.3). Antineutrinos are also emitted, in small amounts, after the $\beta^-$ decay of activated nuclei, following neutron irradiation of the fuel and/or the structural materials inside the reactor core [42]. Moreover, spent fuel stored in near reactor facilities is also responsible for the emission of a small percentage of the total antineutrino flux. A schematic illustration of $\beta^-$ decay is shown in Fig. 3.1 In the following, $\beta^-$ decay will be simply referred to as $\beta$ decay.

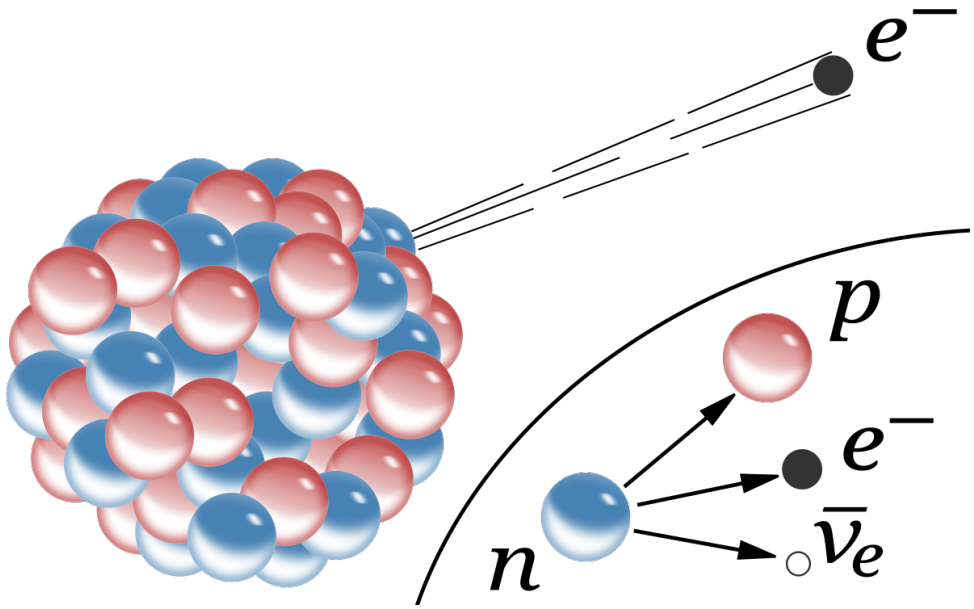



Figure 3.1: Illustration of $\beta^-$ decay in nuclei.

## 3.2. Summation Method

Up to now, the reactor antineutrino spectral modeling has been based on two main approaches: the conversion method and the summation, or *ab initio*, method. These computational approaches are both subject to limitations coming from uncertainties and incompleteness of the nuclear input data. This translates into mismatches with the experimental observations, manifested not only in the overall predicted flux, but also from the spectral shape point of view. In the context of the JUNO-TAO measurements, characterized by their extraordinary energy resolution, the summation method stands out as it preserves the so-called fine structure of the antineutrino spectrum, otherwise inaccessible to the conversion method [45].

Moreover, the summation method enables to treat each isotope in the reactor as a distinct contribute to the overall spectrum, allowing the identification of critical nuclides in the theoretical prediction. For these reason, this thesis concentrates on the use and correction of the summation method, a review of which is presented in this section.

The summation method is first defined to compute the reactor total $\beta$ spectrum and consists in summing the $\beta$ spectra $S_p(E)$ of all the beta emitters within the reactor itself. More specifically, these are the fission fragments p produced by the fission process of any

of the four actinides k considered above: $^{235}$U, $^{238}$U, $^{239}$Pu and $^{241}$Pu:

$$S_{tot}(E,t) = \sum_{k,p} A_p^k(t) S_p(E), \tag{3.2}$$

where $A_p^k(t)$ is the activity of the p$^{\text{th}}$ fission fragment due to k$^{\text{th}}$ actinide at irradiation time t and E is the electron kinetic energy. The activities of the fission fragments can be estimated using their cumulative fission yields $y_p^k$. This is an approximate treatment, founded on the assumption that all isotopes in the reactor are at equilibrium. There exist non-equilibrium models whose aim is to correct for this approximation when its validity fades. In this way, Eq. 3.2 becomes:

$$S_{tot}(E,t) = \sum_k f_k(t) \sum_p y_p^k S_p(E) = \sum_k f_k(t) S_k(E,t), \tag{3.3}$$

where $f^k(t)$ is the k$^{\text{th}}$ actinide fission rate, and $S_k(E,t)$ represents the total $\beta$ spectrum associated to one fission of the k$^{\text{th}}$ actinide, and is often referred as actinide fission spectrum [42]. In general, the $\beta$ spectrum $S_p(E)$ can be computed as the weighted sum of N$^{\text{p}}$ single $\beta$ transitions connecting either the ground state (GS) or a metastable state (MS) of the father nucleus to one of the excited levels b of the daughter nucleus:

$$S_p(E) = \sum_{b=1}^{N^p} B_b^p S_b^p(E, E_{0,b}^p), \tag{3.4}$$

where $B_b^p$ indicates the probability (branching ratio) of decay through a given transition and $E_{0,b}^p$ is the endpoint energy of the transition. The latter can be computed as:

$$E_{0,b}^p = Q_\beta^p + E_{MS}^p - E_b^{lvl}, \tag{3.5}$$

where $Q_\beta^p$ is the Q-value of the $\beta$ decay (corresponding to a GS to GS transition), $E_{MS}^p$ is the energy of the parent nucleus metastable state ($E_{MS}^p = 0$ if the parent nucleus $\beta$ decays from the ground state), and $E_b^{lvl}$ is the energy of the daughter nucleus b$^{\text{th}}$ excited state.

It is important to underline that the branching ratios of the single transitions add up to unity only if the nucleus of interest undergoes $\beta$ decay with a 100% probability; otherwise the sum of the branching ratios coincides with the probability to have $\beta$ decay. In brief, the combination of Eq. 3.3, 3.4 and 3.5 allows to express the overall reactor $\beta$ spectrum as the sum of numerous individual $\beta$-branch contributions. To accurately reproduce each branch, both a reliable formal treatment of $\beta$ decay and high-quality evaluated nuclear

data are required.
Conservation of energy enables to establish a link between the electron spectrum and the corresponding antineutrino spectrum is established at the level of each transition it is sufficient to replace the electron kinetic energy E with the antineutrino energy, given by $E_{\bar{\nu}} = E^{p}_{0,b} - E$; here, the recoil of the daughter nucleus is neglected [42], as shown in Fig. 3.2.

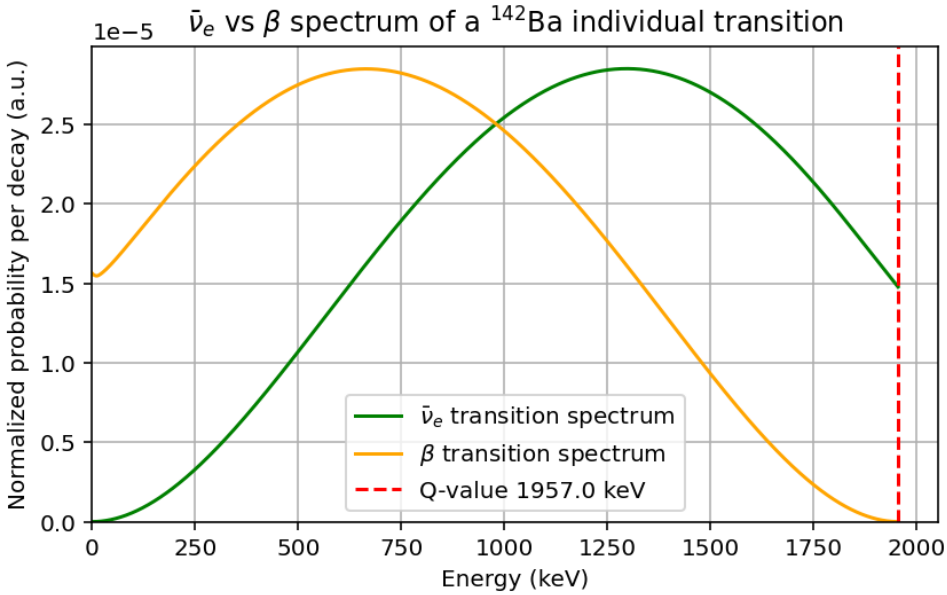



Figure 3.2: The figure shows the electron ($\beta$, in orange) and antineutrino ($\bar{\nu}_e$, in green) spectra associated to a single transition of $^{142}$Ba. This represents the crucial step in which the $\bar{\nu}_e$ spectrum is obtained from the $\beta$ one by substituting the electron energy E with the antineutrino energy $E_{\bar{\nu}} = E^{p}_{0,b} - E$, where $E^{p}_{0,b}$ is the endpoint energy of the transition. These spectra are normalized to the branching ratio of the specific $\beta$ transition to which they are associated. The weighted sum of the obtained transitions (see Eq. 3.4) then gives the overall isotope spectrum, as shown in Fig. 3.3. This is the founding characteristic of the ab-initio approach.

An example of how the $S_p(E_{\bar{\nu}})$ antineutrino spectrum is computed as the sum of individual transitions is shown in Fig. 3.3.

In the following of this thesis, only antineutrino spectra will be considered.
In summary, a complete knowledge of the individual $\beta$ transitions spectra and cumulative fission yield, alongside with validated neutronic codes for the prediction of fission fractions evolution in time are essential to obtain a detailed and in principle correct estimation of a reactor antineutrino spectra.

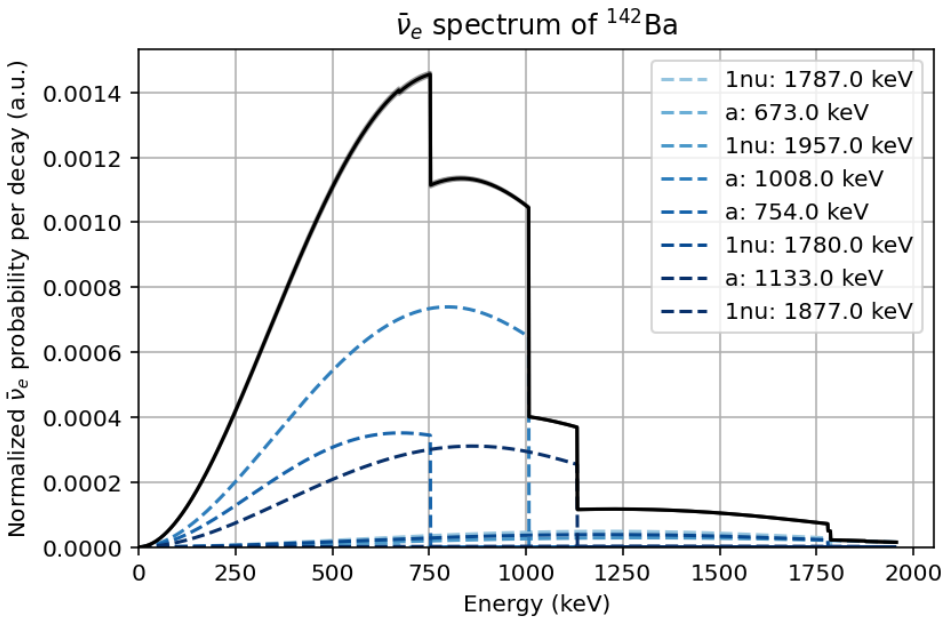



Figure 3.3: Example of computation of the total antineutrino spectrum $S_p(E_{\bar{\nu}})$ for $^{142}$Ba. The single $\beta$ transition spectra $S_b^p(E_{\bar{\nu}}, E_{0,b}^p)$ are the blue-shaded ones, while the total $S_p(E_{\bar{\nu}})$ spectrum is the black one. In the legend, the transitions type is indicated, as well as their endpoint energy: $a$ stands for allowed, while $1nu$ stands for first forbidden non-unique.

### 3.2.1. Uncertainties Treatment

It is crucial to treat the propagation of the uncertainties associated to all of the quantities involved. Many sources of uncertainty may have to be accounted [42]. Generally speaking, the uncertainties are treated through the covariance matrix formalism, a square $(n \times n)$ matrix giving the covariance between each pair of elements of a given random vector. Consider a set random vectors $\boldsymbol{X}_h = (x_1, ..., x_k, ..., x_m)$ with $h = 1, ..., n$, where the scalar components are themselves random variables. The covariance matrix is defined as:

$$\sigma_{hl}^2 = \frac{1}{m} \sum_{k=1}^{m} (x_{kh} - \mu_h)(x_{kl} - \mu_l), \tag{3.6}$$

where $\mu_h$ is the expectation value of the random vector $\boldsymbol{X}_h$. The elements on the diagonal are the variances $\sigma_{hh}^2$ of the vector $\boldsymbol{X}_h$, hence always non negative. Every diagonal element with $h \neq l$, is the covariance between $\boldsymbol{X}_h$ and $\boldsymbol{X}_l$. If this quantity is positive, it means that if one of the two grows, the other grows too. The covariance is null if the two random vectors are independent.

It is possible to consider various sources of uncertainty in reactor antineutrino modeling

through summation method. The most important uncertainty sources are:

- cumulative fission yield,
- fission fraction,
- q-value,
- branching ratio of individual $\beta$ transitions,
- individual transition normalized spectrum.

For a given fission fragment $p$, the covariance matrix associated with an uncertainty source $s$ can be expressed as:

$$V_p^s = \sum_i V_i^s + \sum_{i \neq j} V_{ij}^s, \tag{3.7}$$

where the first term sums the covariances of individual $\beta$ branches and the second term accounts for cross-correlations between branches due to shared parameters. Assuming the different uncertainty sources $s$ as independent, the total fragment covariance becomes:

$$V_p = \sum_s V_p^s. \tag{3.8}$$

The uncertainty in each energy bin is given by the square root of the corresponding diagonal element of $V_p$. At the actinide level, the covariance of a fission spectrum $S_k(E)$ can be approximated at first order using the Jacobian formalism:

$$V_k \simeq \sum_{p,q} y_p^k y_q^k V_{pq} + \sum_{p,q} V_{y_p^k y_q^k} S_p(E) S_q(E), \tag{3.9}$$

where $V_{pq}$ presents the covariance between fragment spectra $p$ and $q$, $y_p^k$ are cumulative fission yields, and $V_{y_p^k y_q^k}$ accounts for correlations among yields. One must also account for correlations between fission rates of different actinides, as a fragment may contribute to the spectra of more than one actinide. The total reactor spectrum covariance is then approximated as:

$$V_{tot} \simeq \sum_{k,l} V_{f_k f_l} S_k(E) S_l(E) + \sum_{k \neq l} \sum_{p,q} f_k f_l y_p^k y_q^l V_{pq} + \sum_k f_k^2 V_k, \tag{3.10}$$

where $f_k$ is the fission rate of the k$^{\text{th}}$ actinide and $V_{f_k f_l}$ is the covariance of the actinide fission rates.

In conclusion, a first-order estimate of the total uncertainty in the reactor antineutrino

spectrum is provided by this framework, considering and combining contributions from fragment spectra, cumulative yields, and fission rate correlations.

## 3.3. Discrepancies in Reactor Antineutrino Spectrum Modeling

In the context of reactor antineutrino spectrum modeling, a persistent discrepancy between the predicted and measured antineutrino flux has been documented. In the literature, such discrepancy is referred as Reactor Antineutrino Anomaly (RAA). It was first highlighted by in 2011 [46] as a result of the comparison between a Huber-Mueller model [47, 48] prediction and the IBD rates from 19 short-baseline experiments. As the Huber-Mueller model has been mentioned, in the interest of completeness, it is worth mentioning that it is a conversion based method, where the antineutrino spectrum is extracted from measurements of the electron spectra of fissioning isotopes. This approach relies on empirical fits of the electron data and subsequently translates them into antineutrino spectra. The discrepancy consisted in a deficit of the measured mean cross-section per fission, which is currently estimated to be at $(6.6 \pm 2.4)\%$ and significant at the level of $2.8\sigma$ [49]. This difference was then confirmed by more recent short-baseline experiments, such as Daya Bay [31], RENO [32], and Double Chooz [50], not only in the total antineutrino flux. Moreover, when the predicted IBD spectra and the measured one are normalized to the same integral, an excess of measured events is observed in the 4-6 MeV range, commonly referred to as the *5 MeV bump* [51, 52]. While the RAA is not a drastic problem anymore, the bump still needs an appropriate explanation. Short-baseline experiments provide the most precise and reliable experimental data on reactor antineutrino spectra thanks to their proximity and high statistical significance. For this reason they are also the primary instrument to investigate these discrepancies. [51, 52]. The spectral distortions measured at various short-baseline experiment are shown in Fig. 3.4.

In the previous section the summation method has been discussed extensively. From the point of view of this computational approach, the 5 MeV bump can be attributed to inaccuracies and incompleteness in the evaluated nuclear data employed in the theoretical estimation. If these are in fact present at the level of individual $\beta$ transitions, they would be automatically propagated into the total antineutrino spectrum and could potentially contribute to the observed flux deficit and spectral distortions. A revision of these nuclear input data is then of crucial interest. The conversion method, while historically relevant and widely used, is naturally insensitive to individual fission fragment contributions and therefore is not well-suited to provide details on the origins of spectral anomalies. As one

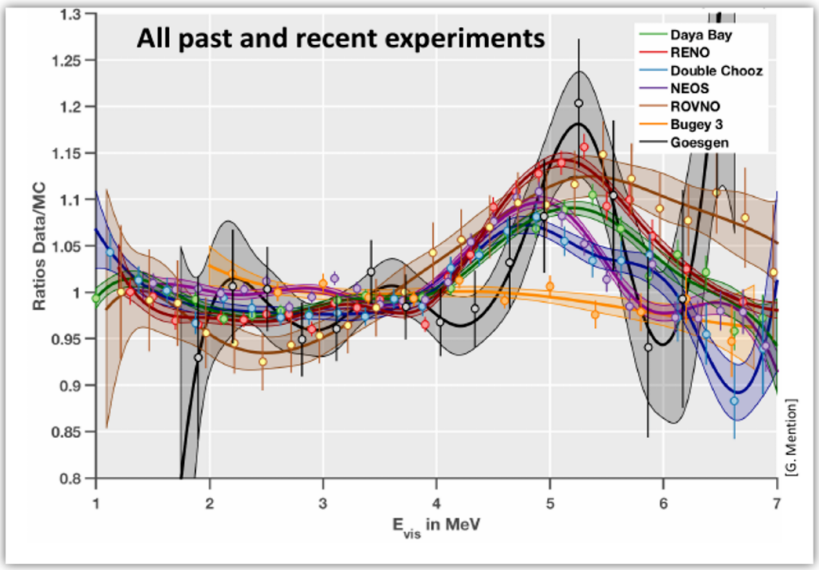



Figure 3.4: Ratio of the measured to the predicted IBD spectrum at various short-baseline experiments. The 5 MeV bump is well identified. The figure is taken from [49].

would expect, comparisons between summation- and conversion-based predictions indicate that the summation method lends itself better to identify the contributions of specific isotopes to spectral distortion, with specific reference to the 5 MeV bump [49, 51, 52]. Several hypotheses have been proposed to explain the 5 MeV bump within the context of the summation method. These include:

- **Incomplete nuclear data:** Systematic biases in the predicted flux could be the result of a non-complete characterization of the high-energy region of the spectrum for certain fission fragments. These can involve unobserved decay paths or wrongly estimated individual branching ratios, as well as errors in the determination of the cumulative fission yields.
- **Detector systematics or modeling:** residual uncertainties in detector response could affect the observed spectra, since the observed distortions are not always fully consistent.
- **Existence of new physics:** another proposal to explain the RAA was the existence of a sterile neutrino mixing with the electron flavor. Indeed, such process could, in principle, explain partially the observed deviations.

A mis-prediction in the model or an underestimated uncertainty budget remain the favored explanation to this day, as reported in [49]. These discrepancies highlight the interplay between experimental measurements and theoretical predictions, providing a

strong motivation to revise and improve both nuclear data and summation calculations, for example including more and more forbidden transitions.
Among the many possible sources of error involved in reactor antineutrino spectrum modeling, the Pandemonium effect [53] is definitely one of the most important. The aim of this thesis is to define a strategy to mitigate the Pandemonium effect and fix the 5 MeV bump. For these reasons, the next section is entirely dedicated to the presentation of the Pandemonium effect. This effect's potential role as a source of systematic bias in summation predictions underlines, once again, the importance of accurate nuclear input for antineutrino spectrum calculations.

## 3.4. Pandemonium Effect

The determination of the $\bar{\nu}_e$ spectrum of a generic isotope through the summation method requires the knowledge of the individual $\beta$ transitions spectra. These are computed using the theory of $\beta$ decay from the decay scheme of the daughter nucleus, whose energy levels are populated as a consequence of the $\beta$ transitions themselves. When a specific level is populated, after some characteristic time, which can be determined through the Heisenberg energy principle, a de-excitation gamma ($\gamma$) ray is emitted. The measurement of these gamma rays enables the reconstruction of the de-excitation levels scheme of the daughter nucleus. Assuming the father nucleus to be either in the ground state (GS) or in a metastable state (MS), one can individuate the single $\beta$ branches involved in $\beta$ of that specific isotopes. In particular, from the intensity measurements of the de-excitation $\gamma$-rays it is possible to infer the branching ratios of the individual transitions and their uncertainties, together with their associated quantum numbers, transition type and endpoint energies. These information are crucial for the determination of the spectrum of the individual $\beta$ transitions, whose weighted sum on the branching ratios gives the total $\bar{\nu}_e$ spectrum of the isotope (see Eq. 3.3). Clearly the same can be done regarding the electron spectrum.

The measurements of the de-excitation $\gamma$-rays have been carried out mostly with High Purity Germanium detectors (HPGe). These detectors are well known for their incredible energy resolution, that enables the recognition of individual $\gamma$-rays. However, due to their low atomic number, these detectors are also characterized by a low efficiency if compared to other $\gamma$-ray detectors such as the NaI(Tl) scintillators. Moreover, this detection efficiency decreases with energy. For usual applications in nuclear metrology, this is not a big problem, indeed the great energy resolution compensates for these inefficiency. On the other hand, when determining the full de-excitation scheme of a daughter isotope, the

measurement requires the full knowledge of every possible $\gamma$-ray emitted. This requirement is necessary because the GS to GS $\beta$ transition is not followed by the emission of a $\gamma$-ray. The branching ratio of this transition is computed as:

$$b_0 = 1 - \sum_{i=1}^{N} b_i, \tag{3.11}$$

where $b_0$ is the GS-GS transition branching ratio, 1 is the normalization of the branching ratios - in this example it is assumed that the isotope $\beta$-decays with a 100% probability[1], and $b_i$ are the branching ratios of all the other transitions. These latter can be inferred from the de-excitation spectrum. It is important to understand that a missed de-excitation $\gamma$-ray produces not only a missing level in the decay scheme, but also messes its internal coherence. In particular, the qualitative effect is an over-estimation of the branching ratios associated to low energy $\gamma$-rays, with a consequent under-estimation of the low energy $\bar{\nu}_e$ spectrum.

This bias was first noticed by Hardy et al. in 1977 [53]. Using a Monte-Carlo approach, they simulated the $\gamma$-de-excitation of a virtual nucleus named "Pandemonium" and its detection with a HPGe detector. A non-negligible amount of $\gamma$-rays was found to remain unobserved with HPGe detectors when compared to the simulation. This pure measurement effect, schematically illustrated in Fig. 3.5, becomes more probable as the complexity of the cascade of gamma rays increases with level density and $Q_\beta$ energy. It is called Pandemonium effect. As already mentioned, the expected qualitative effect of this bias is an under estimation of the low energy part of the antineutrino spectrum, corresponding to an overestimation of the high energy one.

As reported in [49], among the 40 top contributors to the $^{235}$U IBD flux, 19 have been identified to be subject to the Pandemonium effect. In particular, the ENSDF library [54] is one of the most affected, and not yet corrected. It is crucial then to develop experimental techniques and/or theoretical instruments to mitigate this bias, possibly responsible, as noted in Sec. 3.3 for part of the Reactor Antineutrino Anomaly. In [55] a list of potentially Pandemonium affected isotopes has been proposed. The target of this thesis is to develop a computational approach based on Machine Learning (ML) techniques to mitigate Pandemonium effect.

[1] See the discussion in Sec. 3.2.

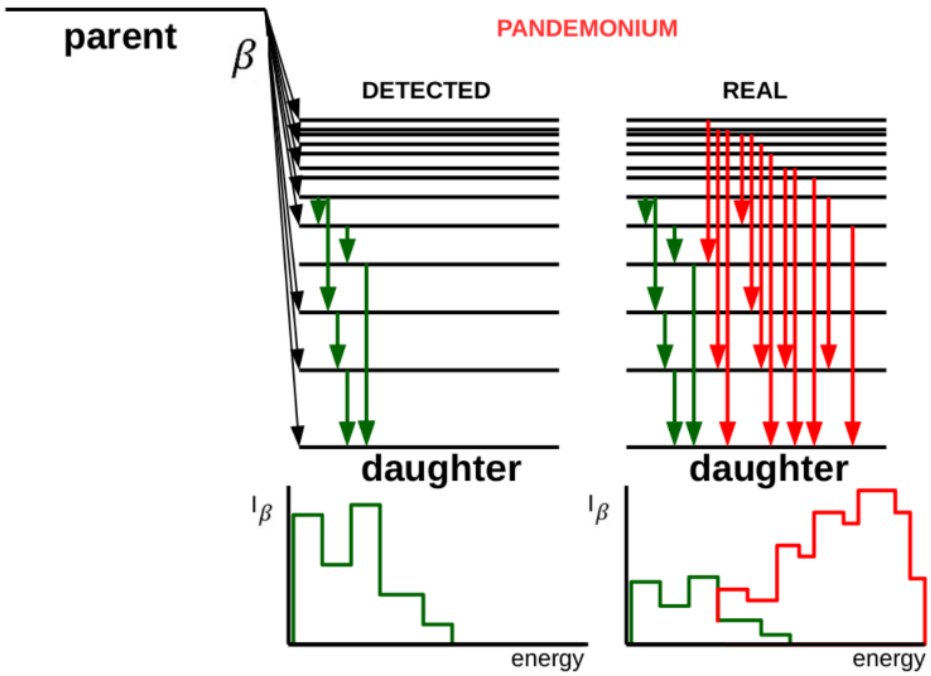



Figure 3.5: Schematic illustration of the Pandemonium effect. The gamma rays emitted from high energy levels (in red) are not detected, and result in an overestimation of the low energy levels (in green) beta feedings. The figure is taken from [49].

## 3.5. Total Absorption Gamma Ray Spectroscopy

In this section the Total Absorption Gamma Ray Spectroscopy (TAGS) technique is described. This is the experimental technique mostly used to contrast the errors induced by Pandemonium effect [56]. Differently from usual $\gamma$-ray spectroscopy, where HPGe are chosen for their very good energy resolution, TAGS measurements employ high efficiency detectors, as NaI(Tl) scintillators. These detectors have an intrinsically worse energy resolution with respect to HPGe, but are able to detect with greater confidence the high energy gamma rays unseen in Pandemonium affected measurements, thanks to thei higher atomic number. A simple resume of the properties of these detectors is shown in Table 3.1.

The aim is to detect the total energy of the $\gamma$ de-excitation cascade after $\beta$ decay, avoiding the systematic error encountered measuring individual $\gamma$-rays. The use of large scintillator crystals allows to approximately cover the whole solid angle ($4\pi$): the isotope of interest is then posed in the center of the experimental setup. As pointed out in [49], this method requires to de-convolute the experimental $\gamma$ spectrum from the detector response in order

| | **HPGe** | **NaI(Tl)** |
|---|---|---|
| Detector type | Solid state | Scintillator |
| Efficiency | Low | High |
| Energy Resolution | Very High | Moderate |
| Effect on $\bar{\nu}_e$ spectrum | Pandemonium effect | Correct measurement (TAGS) |

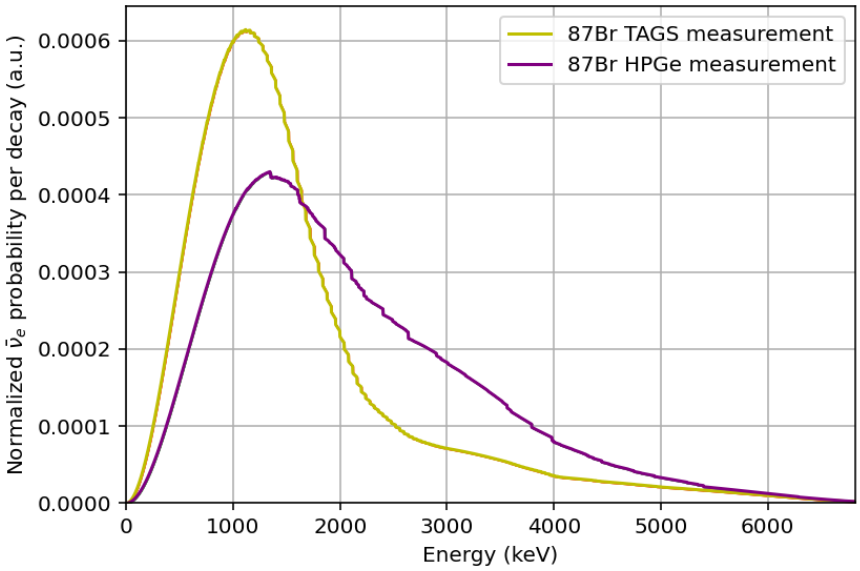
Table 3.1: In the table are shown the main characteristics of HPGe and NaI(Tl) detectors, in the context of $\gamma$-ray spectroscopy for Pandemonium effect measurements.

to obtain the $\beta$ transition intensities. Various experimental campaigns have been and still are conducted to use the TAGS technique and produce new evaluations of the $\beta$ and $\bar{\nu}_e$ spectra of the Pandemonium affected isotopes. At the time of this research, 52 TAGS measurements where at disposal of the author.

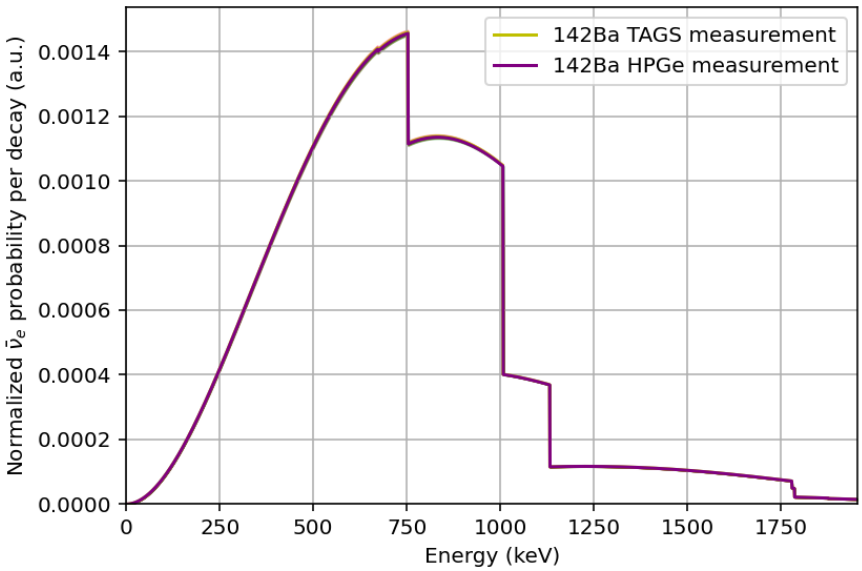
In Fig. 3.6 examples of comparison between the Pandemonium affected and TAGS measured spectra per decay, normalized to the $\beta$ decay probability of two isotopes is shown. The first case (Fig. 3.6a) shows that the HPGe measurements provides an overestimation of the high energy part of the spectrum, while underestimating in low energy, as explained in Sec. 3.4. Hence Pandemonium effect is present. This example is one of the most affected between the 52 at disposal. On the other hand, Fig. 3.6b shows that the TAGS measurement do not necessarily produce a different spectrum than HPGe measurement: in this case it is possible to conclude that no Pandemonium effect is present. This is an important distinction that will be recalled in the following of this work.
The normalized probability spectra per decay have been computed using the RenShape software [57], an open-source program developed by the INFN Milano Bicocca group, with whom the author collaborated during the research. An overview on its functioning and usefulness is presented in the next section.

## 3.6. Codes for the Evaluation of Reactor Antineutrino Spectrum

Reactor antineutrino modeling with the summation method fundamentally relies on the existence of computational tools able to estimate in the most reliable and complete way the $\bar{\nu}_e$ spectra of the fission fragments. An example of such software is BetaShape [58]. This kind of tools exploit the experimentally obtained information on the daughter decay scheme. For instance, BetaShape uses as an input the ENSDF files containing this kind




(a) Normalized probability spectrum per decay of $^{87}$Br, for different kind of measurements: TAGS (in yellow) and HPGe (in purple). The normalization is given by the $\beta$ decay probability.




(b) Normalized probability spectrum per decay of $^{142}$Ba, for different kind of measurements: TAGS (in yellow) and HPGe (in purple). The normalization is given by the $\beta$ decay probability.

Figure 3.6: Examples of comparison between HPGe and TAGS measurements. Upper figure: the comparison is shown for $^{87}$Br. The isotope shows a significative Pandemonium effect. Lower figure: the comparison is shown for 142Ba. Pandemonium effect is practically absent.

of knowledge. Branching ratios, transition type and endpoint of the transition are the crucial quantities for the prediction of the spectra. The software uses the theory of $\beta$ decay [2] and implements many corrections accounting for weak magnetism and radiative phenomena, moreover it includes the treatment of non-unique forbidden transitions, it is possible to estimate the spectra of individual $\beta$. However, BetaShape suffers some major problems: it is not an open source software. The user is therefore required to trust the prediction. Moreover, the data used in BetaShape comes exclusively from the ENSDF library. This promotes the proliferation of possible bias such as Pandemonium effect, as explained in Sec. 3.4.

### 3.6.1. RenShape

The work carried out in this thesis was largely based on the generation of antineutrino spectra via RenShape [57], an open-source, Python based, computational package developed by the INFN Milano Bicocca group. The main goal of RenShape is predicting the non-oscillated reactor antineutrino spectrum by employing the summation method, reviewed in Sec. 3.2. The software has been proposed as a verified[2] and open source alternative to other codes for the prediction of $\beta$ and $\bar{\nu}_e$ spectra already present in the scientific community. Differently from BetaShape, RenShape relies on many libraries for the spectral evaluation, as ENSDF, ENDF-B [59], JEFF [60] and others. This choice is crucial in order to have the most reliable prediction possible of the reactor antineutrino spectrum. However, it is important to underline that RenShape is not itself a spectrum calculator, but a computational framework that relies on BetaShape for the actual evaluation of $\beta$ and $\bar{\nu}_e$ spectra. Its first official release is scheduled for 2025.

The information used by RenShape is organized in a HDF5 file, whose compilation occurs following a specific workflow that will be described below. Once compiled, the *.lazy* file can be read using the RenShape reader, significantly reducing computational times, as all the necessary data is already present and does not need to be reevaluated every time. Through the reader, it is possible to access fundamental quantities for the summation method, such as cumulative fission yields (CFYs), Q-values and antineutrino spectra, both per single transition and in their total sum for each isotope. Every quantity comes with its own uncertainty.

The HDF5 file compilation begins collecting the CFYs of every isotope from the JEFF library [60]. These yields represent the probability that a fission fragment is produced from the fission of a specific actinide. Their knowledge is fundamental for the correct

[2] Verification procedure is ongoing in the research group.

evaluation of the reactor antineutrino spectrum. In particular, RenShape collects the fast fission yields of $^{235}$U, $^{238}$U, $^{239}$Pu and $^{241}$Pu, and the thermal fission yields of $^{235}$U, $^{239}$Pu and $^{241}$Pu. In this first version of the software, the cumulative fission yields of different actinides are treated as independent, even if this is not the case: uncertainty propagation in the summation method is then treated accordingly. The developers of the code aim to provide better estimations of such uncertainties in future releases.

The second step of the compilation is to retrieve the Q-values of each isotope with non-zero CFY. Q-values and their uncertainties, if present, are collected from the AME library [61]. If not present in the AME library, the Q-values are taken from the ENSDF library [54]. The other nuclear properties, such as atomic number, neutron number, half-life and $\beta$ decay probability[3] are taken from the ENSDF library directly.

The evaluation of the antineutrino spectrum consists in the reading of a pre-compiled file containing information on the individual $\beta$ branches of every isotope: specifically, the transition branching ratios, the endpoint energy and the transition type are indicated. These pieces of information are taken from the ENDF-B library [59]. When such data is available, the isotope is labeled as *discreet* and the spectral computations are passed to BetaShape. On the other hand, if these quantities are not recoverable from ENDF-B, it is possible that the pre-compiled file suggests an estimate of the isotope antineutrino spectrum based on theoretical CGM calculations. CGM stands for Cascading Gamma-ray and Multiplicities Code [62], i.e. the software used for these calculations. In this case the nuclide is labeled as *continuum* in the pre-compiled file, indeed these spectra are continuous functions of the energy, hence they must be interpolated using the RenShape binning (1 keV) as a pre-processing step. These theoretical spectra are not normalized by default, thus RenShape forces their normalization before saving them in the *.lazy* file.

After this first procedure, RenShape uses the ENSDF library as a fallback. Indeed, it is possible that some isotopes are not included in both the ENDF-B library and in CGM calculations: in these cases the antineutrino spectrum is evaluated by BetaShape using the data coming from ENSDF. If instead, the CGM calculations are available, an evaluation of the completeness of the ENSDF decay scheme is performed. If the decay scheme is considered to be approximately complete, the CGM spectrum is substituted with the ENSDF one, otherwise nothing changes. This fallback mechanism ensures that the most reliable data source is always used, prioritizing experimental evaluations over purely theoretical predictions whenever possible.

[3] Recall, from Sec. 3.2, that this quantity should be equal to the sum of the individual $\beta$ transitions branching ratios.

At the same time, the workflow ensures that RenShape prioritizes reliable sources over ENSDF data. As pointed out in Sec. 3.4, the ENSDF is probably filled with data affected from Pandemonium effect. On the other hand the ENDF library correctly contains the available TAGS corrections, resulting in a much more trustworthy estimates of the fission fragments antineutrino spectra.
A schematic representation of this workflow is displayed in Fig. 3.7.

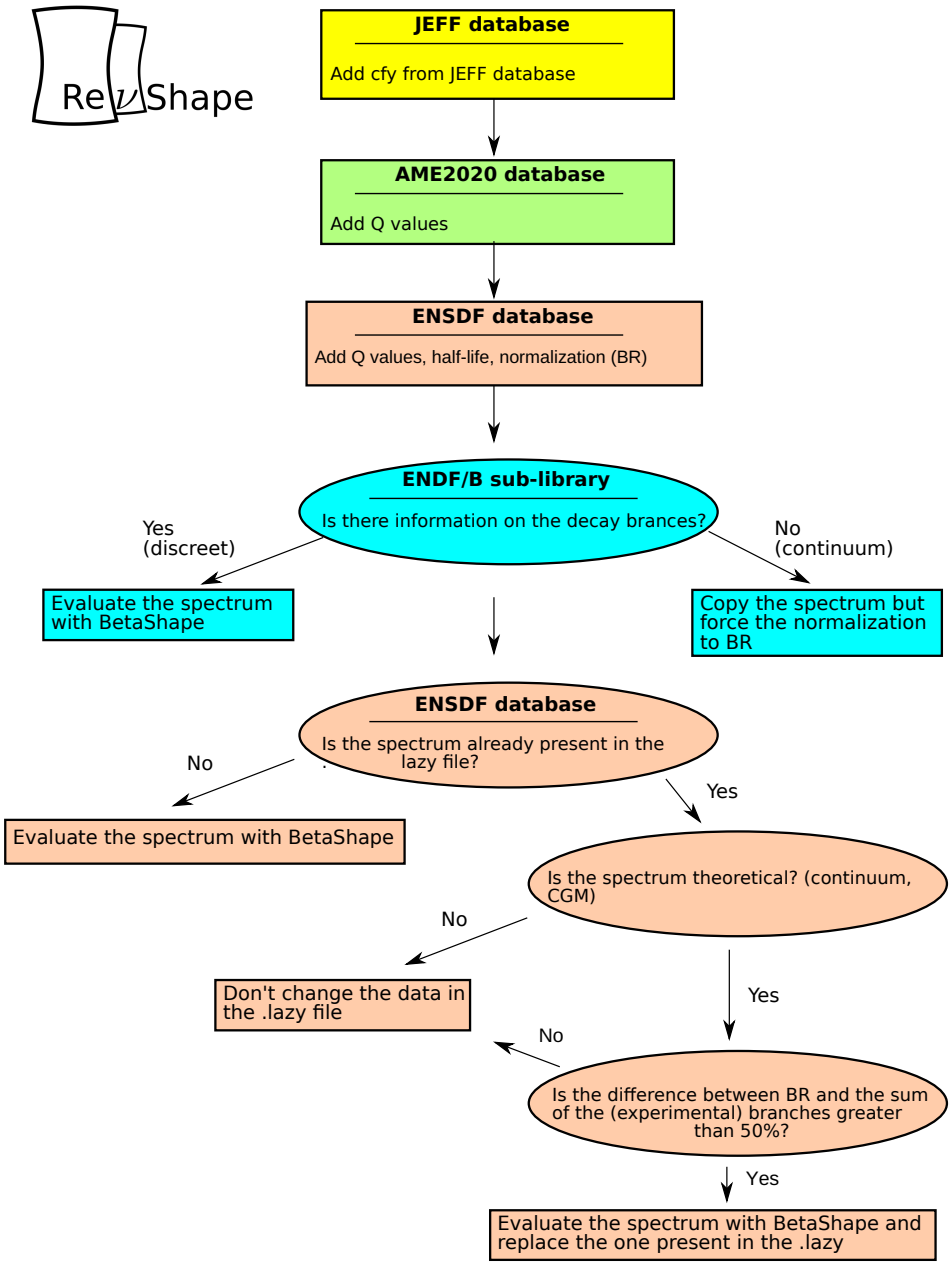



Figure 3.7: Block scheme of the RenShape workflow presented in Sec. 3.6.

# 4 | Machine Learning: Concepts and Methods

The problem of pattern recognition in data has been and still is a very important one. It has a long success story, while remaining very current. For instance, one can think about atomic spectra and their regularities, the ability to identify them has enabled the human kind to enter in a new era with quantum mechanics. The use of current computational technologies has opened the possibility to introduce computer algorithms in order to automatically discover patterns of interest, which can be then used for other purposes. This allows to consider completely new approaches to complicated problems where conventional methods fail, or become too complex. In this work, machine learning techniques have been employed to understand whether regularities could be individuated in the context of Pandemonium effect, reviewed in Sec. 3.4 and possibly to extract a computational model with the ability to predict a corrected version of the antineutrino spectrum for isotopes critical to the reactor antineutrino spectrum. This chapter is designed to give an introduction and theoretical support to the fundamental machine learning concepts and methods used in the research carried out in the thesis.

## 4.1. Fundamentals of Machine Learning

To introduce the fundamental concepts of Machine Learning (ML), the choice has been to take advantage of an example reported in the introduction of [63], together with the following discussion. Consider the possibility to recognize handwritten digits shown in Fig. 4.1. The digit corresponds to a $28\times28$ pixel image, hence it can be represented has a vector $\boldsymbol{x}$ containing 784 real numbers. The idea is to build a machine that taking as an input such vector, is able to produce the corresponding digit 0, ..., 9 output. Nonetheless the variability of handwriting is very wide. Heuristic or handcrafted rules can in principle be introduced to classify and distinguish the various patterns, but this would lead to a huge amount of rules and exceptions to them, which would lead to inevitably poor results. Much more interesting results can be obtained using a machine learning approach.

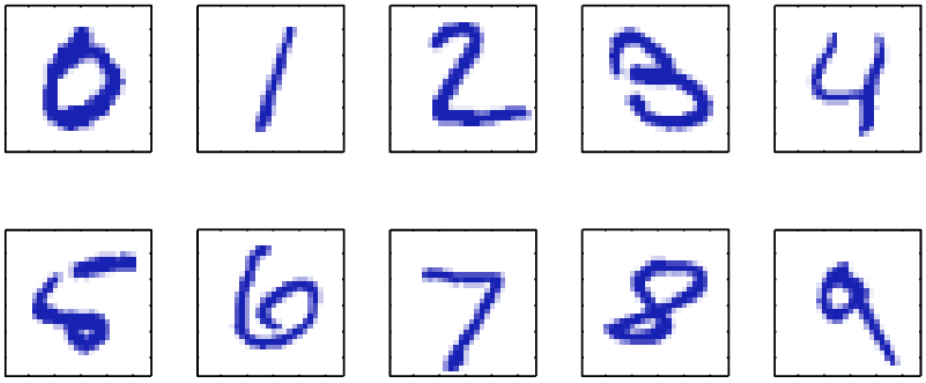


Figure 4.1: Examples of hand-written digits taken from US zip codes. The figure is taken from [63].

Suppose to have an adaptive model at disposal. One can introduce a large set of $N$ digits in the form of vectors $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ called *training set* and use it to tune the parameters of the adaptive model. Each training set vector is associated in advance to a *target vector* $\boldsymbol{t}$, which represents the identity of the digit itself, i.e. 0, 1, 2,..., 9. Generally speaking, suitable techniques can be employed to represents the various categories in form of vectors. Running the machine learning algorithm produces as a result a function $\boldsymbol{y}(\boldsymbol{x})$ which takes a digit image vector and produces an output vector with the same representation of the target vectors. The form of the function $\boldsymbol{y}(\boldsymbol{x})$ is shaped during the so-called *training* (or *learning*) phase, exploiting the training data. Once training is completed, ideally the model should be able to correctly predict the identity of new, never seen, digit image vectors. This ability is called *generalization*. The collection of these new vectors used to test the generalization capacities of the algorithm is called *test set*.
In practical applications it is very difficult to have a training set that comprises the huge variability of input vectors; most of the time it would be only a tiny fraction, hence generalization is crucial. Moreover, in many cases it is very useful to apply a *preprocessing* to the original input variables, transforming them into a new space of vectors which hopefully helps to solve the pattern recognition problem. In this example one could think to translate and scale the digit images so that each digit is centered in a box of given size. This practice reduces a lot the variability of the input, allowing the model to concentrate only on pattern relevant to the problem. Note that the test set data must be preprocessed using the same choices taken for the training ones. Another advantage of preprocessing can be to reduce the computational load of the model, resulting in a faster convergence of the algorithm. Forms of reduction of order can indeed be considered as useful preprocessing, as it will be discussed later. In this case it is important to avoid loss

of information useful to the identification of the $\boldsymbol{y}(\boldsymbol{x})$ function.
It is possible to distinguish two main applications of machine learning:

- **Supervised learning problems:** the training data comprises examples of the input vectors $\boldsymbol{x}$ along with their corresponding target vectors $\boldsymbol{t}$,
- **Unsupervised learning problems:** the training data consists of a set of input vectors $\boldsymbol{x}$ without any corresponding target values.

The example presented above belongs to a family of supervised learning problems called *classification* problems. Here the goal is to assign every input vector to one of the finite number of discrete output categories. The other big family of supervised learning problems is *regression* problems, where the desired output consist in one or more continuous variables. An example of regression problems will be the one discussed in this thesis.
Unsupervised learning problems typically have a different aim: one may be interested in highlighting groups of data with similar characteristics, known as *clustering*, or to evaluate the data distribution in the input space, known as *density estimation*, or also to project high dimensional data in two or three dimensions for the purpose of *visualization*.
Finally, outside this first classification, another possible application of ML concerns the problem of finding appropriate actions to take in a given situation in order to maximize a reward and goes under the name of *reinforcement learning*. Here no example of correct output is given, a trial and error approach is used instead. For instance, one could think to teach a neural network how to play backgammon at an high level. The great part is that it works [64]. An example of machine learning application in the JUNO experiment context can be found in [65].

## 4.2. Supervised Learning Principles

As previously stated, the central challenge of a supervised machine learning problem is that the model must perform well on new and previously unseen inputs, hence must be able to generalize. During training, one has the possibility to evaluate some error measure on the training set, called *training error*, and minimize it. So far this is nothing more than an optimization problem. Machine learning instead requires the *generalization error*, or *test error* to be as low as possible. The generalization error is defined as the expected value of the error on a new input. Statistical learning theory provides some guidance on how to affect the model performance on the test set while only observing the training set [66]. Little can be done if training and test set are collected arbitrarily. Allowing some assumptions, progress can be made.

In the context of machine learning, it is assumed that data are generated from an underlying probability distribution, called data generating distribution $p_{data}$. According to the *i.i.d. assumptions* (independence and identical distribution), each example of train and test is independently extracted from the same distribution. This allows formalizing the link between training error and generalization error: for a fixed model, the expected error on the training set coincides with that expected on the test set, since both derive from the same sampling process.
Clearly, when using a machine learning algorithm it is not useful to fix the model parameters and then sample the training and test set. Conversely, what is done is sampling the training set, adjust the model parameters to reduce the training error, and then sample the test set. With this workflow, the test error is inevitably larger or equal to the minimized training error.
A ML algorithm performance is determined by the following two abilities:

- Make the training error small,
- Make the difference between training and test error small.

These factors correspond to two crucial challenges when training a machine learning model, called $underfitting$ and $overfitting$. Underfitting occurs when the model fails to sufficiently reduce the error on the training set, while overfitting emerges when the discrepancy between training error and test error becomes too large. The behavior of the model depends on its capacity, that is, the ability to adapt to different functions: too low capacity prevents the model from learning adequately from the data, while excessive capacity can lead it to memorize irrelevant details of the training set that do not generalize about the test. The capacity of a learning algorithm can be regulated through the choice of its hypothesis space, that is, the set of functions that the algorithm can adopt as solutions. For example, linear regression uses all linear functions of inputs as hypothesis space; instead, by extending the space to polynomials, the capacity of the model increases. Machine learning algorithms achieve the best performance when their capacity is adequate for the actual complexity of the problem and the amount of data available. A model with insufficient capacity fails to solve complex tasks, while a model with excessive capacity can deal with them but risks overfitting itself, capturing superfluous details of training data. An example is shown in Fig. 4.2.

It is useful to distinguish between representational capacity and actual capacity. The first indicates which functions, in principle, a model can represent by varying its parameters. The second also depends on the training algorithm: optimization cannot always identify the ideal function within the available hypothesis, and one is often satisfied with solutions that reduce the error without minimizing it completely.

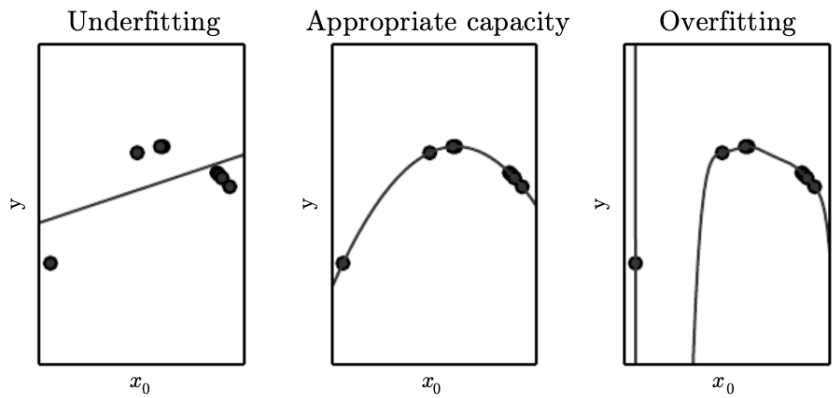



Figure 4.2: Example of underfitting, appropriate capacity and overfitting for a model aiming to fit the sample points with a function. The figure is taken from [66].

This problem of balancing simplicity and complexity recalls an ancient principle, the so-called Occam's razor [66, 67]: when multiple hypotheses are compatible with the data, it is preferable to choose the simplest. In modern statistics, such an idea is realized through tools such as statistical learning theory. A key concept of this theory is the Vapnik-Chervonenkis dimension (VC dimension), which measures the capacity of a binary classifier by indicating the maximum number of points that it can arbitrarily label. A high dimensional VC signals high capacity and, therefore, a potential risk of overfitting if the available data is not sufficient [66].
A fundamental result of this theory is that the generalization error can be limited by a trade-off between model capacity and the number of available examples: the more complex the model, the more data is needed to ensure that the error on the test set remains contained. This principle is also observed empirically by plotting error curves as complexity varies: the training error decreases monotonically, while the test error follows a characteristic U-curve, first decreasing (when underfitting is corrected) and then rising (when overfitting emerges), as shown in Fig. 4.3. Choosing the appropriate capacity of a model depends both on the structure of the model itself and on the amount of data available. In general, parametric models – such as linear regression or neural networks with a fixed number of weights – have a limited capacity, which does not increase as the data increases; on the contrary, non-parametric models can increase their complexity as a function of the size of the dataset, adapting more flexibly to the available information. A useful point of view is offered by the so-called No Free Lunch theorem, according to which, considering all possible learning problems, on average no algorithm is better than another. The best performance is only achieved when the specific regularities of real problems are exploited. This implies that there is no "universal model", and that capacity

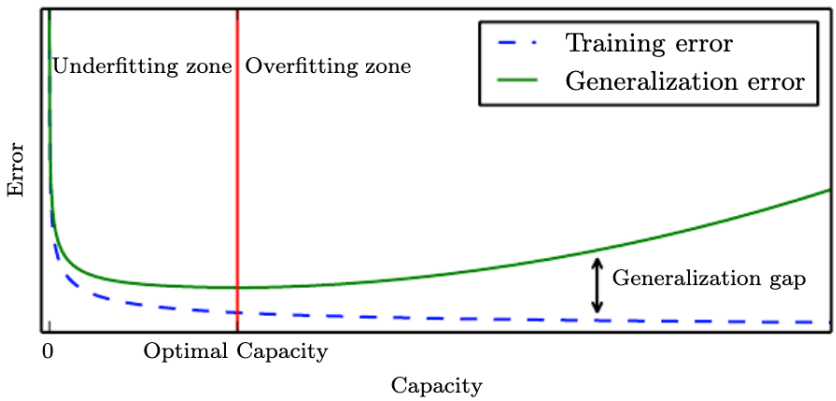



Figure 4.3: The typical relationship between model capacity and error shows that the trend of training error and test error does not coincide. When the capacity of the model is very low, both the error on training and the generalization error are high: this is the underfitting phase. As capacity increases, the training error decreases, but at the same time the gap between training error and generalization error grows. At some point, the increment of this gap outweighs the benefit of training error reduction, thus entering the overfitting phase, in which the capacity of the model is excessive in relation to the task and exceeds the optimal one. The figure is taken from [66].

must always be calibrated in relation to the nature of the task and the amount of data available. In practice, regularization strategies are used to reduce the risk of overfitting. Regularization consists in introducing constraints that limit the complexity of the model: for example, in neural networks too large weights can be penalized, or restrictions can be imposed on the form of the functions to be learned. The underlying idea is to prevent the model from over-focusing on the specific features of the training set, instead favoring more general and robust solutions. Another fundamental aspect of supervised learning concerns the choice of *hyperparameters*, that are, those external parameters that are not learned directly from the algorithm, but must be fixed apriori. The determination of these values is generally carried out through a validation set, separate from both the training and the test set, in order to estimate the generalization capacity of the model without influencing its training.

Finally, to ensure that a model learns meaningful patterns from the data, it is critical to clearly define what is the meaning of "good fit" or, equivalently, how to measure error. A general approach to estimate the parameters of a model is to maximize the probability of the observed data, known as maximum likelihood principle [66]. In other words, the parameters are chosen to make the collected data more likely according to the model. Consider a set of $m$ examples $\mathbb{X} = \{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}\}$ taken independently from the true

but unknown (and inaccessible) data generating distribution $p_{data}(\boldsymbol{x})$. Let $p_{model}(\boldsymbol{x};\boldsymbol{\theta})$ be a parametric family of probability distributions over the same space and indexed by $\boldsymbol{\theta}$, representing the whole set of parameters. $p_{data}(\boldsymbol{x})$ is estimated using $p_{model}(\boldsymbol{x};\boldsymbol{\theta})$ and is action on any configuration $\boldsymbol{x}$. This is a *frequentist* approach, where the true value of $\boldsymbol{\theta}$ is fixed but unknown. The point estimate $\hat{\boldsymbol{\theta}}$ can be instead seen as a random variable, indeed it is a function of the dataset. The likelihood function is defined as:

$$\mathcal{L} = p_{model}(\mathbb{X};\boldsymbol{\theta}) = \prod_{i=1}^{m} p_{model}(\boldsymbol{x}^{(i)};\boldsymbol{\theta}). \tag{4.1}$$

Note that the likelihood is not a probability density of the parameters $\boldsymbol{\theta}$, because they are not random variables; on the other hand the likelihood is formally a probability density of the input data. The maximum likelihood estimator for $\boldsymbol{\theta}$ is then defined as:

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} p_{model}(\mathbb{X};\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{model}(\boldsymbol{x}^{(i)};\boldsymbol{\theta}). \tag{4.2}$$

Here the product over all the elements of the set can be inconvenient for various reasons, like numerical underflow. An equivalent, but simpler, optimization problem is obtained if one takes the logarithm of the likelihood. Indeed the maximum value is not changed, but the product becomes a sum:

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{model}(\boldsymbol{x}^{(i)};\boldsymbol{\theta}). \tag{4.3}$$

Since the arg max does not change rescaling the log-likelihood, it is possible to divide the right hand side by $m$ and get a version of the criterion expressed as the expectation value $\mathbb{E}$ with respect to the empirical distribution $\hat{p}_{data}$, defined by the training data :

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim\hat{p}_{data}} \log p_{model}(\boldsymbol{x}^{(i)};\boldsymbol{\theta}). \tag{4.4}$$

A way to interpret the maximum likelihood estimation is that it consists in the minimization of the difference between the empirical distribution $\hat{p}_{data}$ and the model distribution. This difference can be measured using the Kullback-Leibler (KL) divergence, given by:

$$D_{KL}(\hat{p}_{data}||p_{model}) = \mathbb{E}_{\boldsymbol{x}\sim\hat{p}_{data}}[\log \hat{p}_{data}(\boldsymbol{x}) - \log p_{model}(\boldsymbol{x})]. \tag{4.5}$$

The term on the left is only a function of the data generating process, not the model. This means that when training the model to minimize the KL divergence, minimization

involves only:

$$-\mathbb{E}_{\boldsymbol{x}\sim\hat{p}_{data}} \log p_{model}(\boldsymbol{x}), \tag{4.6}$$

which is the same as maximizing Eq. 4.4. Moreover, since minimization algorithms are much more diffused than maximization ones, the problem is restated as the minimization of the negative log-likelihood (NLL). In this case it is useful to consider the maximum likelihood as a minimization of the Kullback-Leibler divergence, because the KL has a known minimum value of zero. In contrast, negative log-likelihood can take on negative values when $\boldsymbol{x}$ is a real variable [66].

The maximum likelihood estimator can easily be extended to the case where the goal is to estimate a conditional probability $P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ in order to predict $\boldsymbol{y}$ given $\boldsymbol{x}$. This is actually the most common situation, as it forms the basis of most supervised learning problems [66]. Indicating with $\boldsymbol{X}$ the set of all inputs and with $\boldsymbol{Y}$ all targets observed, then the conditional maximum likelihood estimator is:

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{Y}|\boldsymbol{X};\boldsymbol{\theta}). \tag{4.7}$$

If i.i.d. assumptions are valid one can write:

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}). \tag{4.8}$$

Machine learning applications widely use the maximum likelihood estimator because it allows to estimate the parameters of a model by adjusting them, in order to optimize compatibility with observed data. When dealing with large datasets, the approach becomes particularly useful since the estimates become more reliable and representative of the actual phenomenon.

On the other hand, if the number of data is reduced, using the maximum likelihood can lead to overfitting problems. As seen before, to reduce this risk, regularization strategies can be employed, such as weight decay in neural networks, to limit the complexity of the model and make the estimates more stable. In summary, the maximum likelihood principle provides a simple yet very powerful criterion for training supervised learning models, and enables to adapt each model architecture to encountered problem encountered.

However this is not the full picture. Up to now only the frequentist approach has been discussed. In this approach a single value of $\boldsymbol{\theta}$ is estimated, and predictions are all made based on that estimate. Another approach is to consider the whole distribution of $\boldsymbol{\theta}$ values when making a prediction. This approach goes under the name of *Bayesian* statistics [63, 66, 67]. The dataset is directly observed and so is not random, while the true parame-

ter $\boldsymbol{\theta}$ is represented as a random variable since it is unknown or uncertain. In this case, the observation of the data is preceded by the choice of a prior probability distribution $p(\boldsymbol{\theta})$, which represents the knowledge of $\boldsymbol{\theta}$. Usually, the prior is chosen to be sufficiently broad to reflect the high degree of uncertainty in the value of the parameters before observing the dataset. Consider a set of data samples $\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}\}$. It is possible to understand the effect of the data on the parameters combining the data likelihood $p(\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}|\boldsymbol{\theta})$ with the prior $p(\boldsymbol{\theta})$, using the Bayes' rule [66]:

$$p(\boldsymbol{\theta}|\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}) = \frac{p(\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)})}. \tag{4.9}$$

As already mentioned, the prior is initially chosen to be characterized by a big uncertainty, i.e. high entropy; with observation of the dataset, this distribution is progressively tuned. The prior distribution aims to focus on the most plausible parameter, based on the information contained in the training data.
While in maximum likelihood estimation the estimated value for $\boldsymbol{\theta}$ is only one, in the Bayesian approach the entire posterior distribution of parameters is considered As an example, the observation of $m$ samples enables to predict the distribution for new data as:

$$p(\boldsymbol{x}^{(m+1)}|\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}) = \int p(\boldsymbol{x}^{(m+1)}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)})d\boldsymbol{\theta}. \tag{4.10}$$

In practice, uncertainty on the parameters values is always evaluated, reflecting itself in the model predictions which comprehend the variance, not only the point estimation. This is impossible with the frequentist statistics, where the variance is estimated ex post. Clearly, the role of the prior distribution is crucial, and is often a point of critique since its choice is considered to introduce subjective human judgment in the model [66]. Bayesian methods are particularly effective with smaller amounts of data, although they often require more computational resources than classical maximum likelihood estimation.
The most rigorous approach to making prediction in the Bayesian framework is to use the full posterior distribution of parameters. However, it is still useful to have a way to make point prediction. This is due to the intrinsic complexity that emerges in tracking the full distribution while making operations with it. The Maximum A Posteriori (MAP) estimate represents one possibility. It consists in finding the value of $\theta$ that maximizes the posterior probability:

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{x}) = \arg\max_{\boldsymbol{\theta}} \big(\log p(\boldsymbol{x}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\big), \tag{4.11}$$

where $\log p(\boldsymbol{x}|\boldsymbol{\theta})$ corresponds to the usual log-likelihood, while $(\log p(\boldsymbol{\theta})$ collects the infor-

mation provided by the prior distribution. Regularization techniques cited above, such as maximum likelihood penalized with weight decay, can also be interpreted as a maximum a posteriori approximation of Bayesian statistics [66].

## 4.3. Supervised Regression Models

The regression models employed in this thesis are presented in this chapter. The common goal of these models is to learn a predictive function from the observation of the training set, and then test it on previously unseen examples of the test set. In particular the focus will be in Gaussian processes and artificial neural networks, for reasons that will be clear later.

### 4.3.1. Gaussian Processes

Gaussian Processes (GP) are non-parametric, Bayesian models that can be employed in regression machine learning algorithms, as well as in classification ones. The easiest case, often presented in the literature, consist in the modeling of a scalar output function:

$$y = f(\boldsymbol{x}) + \varepsilon, \tag{4.12}$$

where $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$. $\varepsilon$ is a random noise variable assumed to follow a Gaussian distribution, with zero mean and variance $\sigma_n^2$. In many cases this formalism is more than sufficient, however it presents limits when the aim is to model simultaneously many correlated quantities, i.e. when the output function is not a simple scalar, but a vector. One possibility can be to build one model per each dimension of the vector-valued output, but this inevitably prevents from learning possible correlations between them. In these situations then, it is natural to extend the simple Gaussian process to a *multitask* Gaussian process, with a vector as an output. The classic reference used for this subsection is [67], the standard for Gaussian processes. More information is also present in other machine learning texts, as [68] and [63].

**Definition and properties** *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution* [67].
Consider an input vector $\boldsymbol{x} \in \mathbb{R}^d$. A Gaussian process is defined by its mean function $m(\boldsymbol{x})$ and covariance function (or *kernel*) $k(\boldsymbol{x}, \boldsymbol{x}')$ of a real process $f(\boldsymbol{x})$, which is the

function to approximate. The mean and covariance functions are defined as:

$$\begin{aligned} m(\boldsymbol{x}) &= \mathbb{E}[f(\boldsymbol{x})], \\ k(\boldsymbol{x}, \boldsymbol{x}') &= \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))]. \end{aligned} \tag{4.13}$$

No explicit parameters are introduced in this formulation, which enables to reason directly in terms of functions.
A Gaussian process is commonly indicated as:

$$f(\boldsymbol{x}) \sim \mathcal{GP}\big(m(\boldsymbol{x}), k(\mathbf{x}, \mathbf{x'})\big). \tag{4.14}$$

The learning phase of these models involves the inversion of the covariance matrix, which occurs every time the training set is scanned. This poses a limit in *scalability* for GPs. The computational cost grows as $\mathcal{O}(n^3)$ as a function of the number of data $n$. Approximate solution for large dataset, such as sparse methods, are discussed in standard textbooks [63, 67].

**Gaussian regression with uncertainty prediction** Consider a training set $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^m$, where every observation is assumed to be:

$$y_i = f(\boldsymbol{x}_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_n^2), \tag{4.15}$$

where $f(\boldsymbol{x})$ is the function to model, and to which independent gaussian noise with variance $\sigma_n^2$ is added. Assume then a prior gaussian process:

$$f \sim \mathcal{GP}\big(0, k(\mathbf{x}, \mathbf{x'})\big), \tag{4.16}$$

with zero mean for simplicity. The joint vector comprising the observed values of the function and its value $f_*$ in a new (test) point $\boldsymbol{x}_*$ follows a multivariate gaussian distribution:

$$\begin{pmatrix} \boldsymbol{y} \\ f_* \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} K + \sigma_n^2 I & \boldsymbol{k}_* \\ \boldsymbol{k}_*^\top & k(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{pmatrix}\right), \tag{4.17}$$

where $K$ is the training covariance matrix with elements $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$,
$\boldsymbol{k}_* \equiv [k(\boldsymbol{x}_1, \boldsymbol{x}_*), ..., k(\boldsymbol{x}_m, \boldsymbol{x}_*)]^\top$ and $k(\boldsymbol{x}_*, \boldsymbol{x}_*)$ is the covariance matrix element associated to $\boldsymbol{x}_*$.
The predictive distribution for the test set is obtained by conditioning $f_*$ on observed

data $\boldsymbol{y}$.

$$p(f_* \mid \boldsymbol{y}) \sim \mathcal{N}(\mu_*, \sigma_*^2), \tag{4.18}$$

where

$$\mu_* = \boldsymbol{k}_*^\top (K + \sigma_n^2 I)^{-1} \boldsymbol{y}, \tag{4.19}$$

$$\sigma_*^2 = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^\top (K + \sigma_n^2 I)^{-1} \boldsymbol{k}_*. \tag{4.20}$$

Given the knowledge from the training set, the *predictive mean* $\mu_*$ represents the most probable estimate of the modeled function in a test point. It is obtained as a linear combination of the observed values $\boldsymbol{y}$, where the weights are obtained from the correlation between the $\boldsymbol{x}_*$ test and the $\boldsymbol{x}_1, ..., \boldsymbol{x}_m$ training points and from the assumptions on the kernel structure. Intuitively one can think than training points that are "closer" to the new input will have a stronger influence on the prediction.
The *predictive variance* $\sigma_*^2$ is a measure of the uncertainty the model has on the new made prediction: its value is reduced if $\boldsymbol{x}_*$ is "close" to training points while it remains high in not observed regions, as one would expect. The notion of distance involved in these interpretations is measured by the kernel function. The only restriction on the kernel function is that the total covariance matrix $K + \sigma_n^2 I$ must be positive definite. For this reason a usual practice is to add a jitter term to the covariance matrix elements.

**Kernels: choices and criticalities** The heart of the model is the choice of the kernel. It is also the intrinsic limitation of Gaussian processes: indeed a non-appropriate kernel choice makes the model rigid with respect to the data. Thus, it is a strong hypothesis of the model. Here some examples are reported:

- **RBF (Radial Basis Function) or Squared Exponential kernel:** it grants infinitely differentiable and smooth functions. The mathematical expression of this kernel is:

$$k_{\text{RBF}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left( -\frac{1}{2} \sum_{d=1}^{D} \frac{(\boldsymbol{x}_d - \boldsymbol{x}'_d)^2}{\ell_d^2} \right), \tag{4.21}$$

  where $\ell_d$ are the *characteristic lengthscale* of each dimension of the input and $\sigma_f^2$ is the variance of the signal. It is often regarded as too smooth to model noisy physical phenomena [67].

- **Matérn kernel:** the $\nu$ parameter enables to control the regularity of the basis functions, this way less smooth function with respect to RBF can be obtained. The mathematical expression of this kernel is:

$$k_{\text{matérn}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} r\right)^\nu K_\nu \left(\sqrt{2\nu} r\right), \tag{4.22}$$

where $r = \sqrt{\sum_d (\boldsymbol{x}_d - \boldsymbol{x}'_d)^2/\ell_d^2}$, $K_\nu$ is the second kind modified Bessel function, and $\Gamma(\cdot)$ is the Gamma function.

- **Linear kernel:** useful to model eventual linear relation between input and output values. The mathematical expression of this kernel is:

$$k_{\text{linear}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_b^2 + \boldsymbol{x}^\top \boldsymbol{x}', \tag{4.23}$$

  where $\sigma_b^2$ is a constant bias term that enables the kernel to model a global offset of the function with respect to the origin. It is also directly indicated as $k_{bias}$

- **Composite Kernel:** it is also possible to combine different kernels, using sum or product operations, in the tentative to model more complex relations. An example is:

$$k_{\text{composite}}(\boldsymbol{x}, \boldsymbol{x}') = k_{\text{RBF}}(\boldsymbol{x}, \boldsymbol{x}') + k_{\text{linear}}(\boldsymbol{x}, \boldsymbol{x}') + k_{\text{bias}}. \tag{4.24}$$

**Hyperparameters estimation and ARD** The hyperparameters[1] of the kernel chosen to model the data are learned during the training phase by minimizing the negative log *marginal* likelihood.
In the GP context, the marginal likelihood is defined as

$$p(\boldsymbol{y}|X, \boldsymbol{\theta}) = \int p(\boldsymbol{y}|f, \boldsymbol{\theta}) p(f|X, \boldsymbol{\theta}) df, \tag{4.25}$$

where $\boldsymbol{y}$ are the training outputs that have been observed, $f$ is the latent function, and $\theta$ are the kernel hyperparameters. It can interpreted as the likelihood of the data after marginalizing over the latent function $f$, hence, a likelihood conditioned only on the hyperparameters $\boldsymbol{\theta}$. Practical reasons then require the definition of the negative log marginal likelihood (NLML) and the use of the MAP, characteristic of Bayesian inference. A compromise between complexity of the model and adaptiveness to data is reached after a given number of scans of the training sets, which are also called *epochs*. With epochs proceeding the NLML should be a decreasing function. Moreover, it is important to avoid that the NLML gets blocked in local minima during training.
Gaussian process offer a very interesting possibility. It is called Automatic Relevance Determination (ARD) and enables the model to learn a specific lengthscale $\ell_d$ for every input dimension. As it is possible to see in the previous paragraph about kernels, the lengthscale plays a crucial role in their definition: it controls the degree of variation of the

[1] In GP literature, the parameters of the model are usually referred as hyperparameters. These should not be confused with the hyperparameters introduced earlier, since those are the external parameters fixed before training.

kernel function along a dimension $d$. A small lengthscale means that the function varies rapidly along $d$; a big one means that the function various slowly along $d$ . ARD enables to identify irrelevant inputs to the model, as one can look for high valued lengthscales which "block" the model in that specific dimension $d$.
A more refined control on variables relevance is then obtained, with the counter effect of introducing more local maxima in the log-marginal likelihood, hence making optimization harder.

**Multitask Gaussian processes** It was previously noted that classic Gaussian process are defined on scalar outputs. However there are many scientific and engineering applications in which correlated outputs are considered. An example is the problem treated in this thesis. As already mentioned, the aim is to model antineutrino spectra, which appear as arrays containing intensities in various bins. This inevitably requires an extension to multitask Gaussian processes. Each dimension of the output is treated as a task.
They extend the covariance function to the task indexes and define:

$$k\big((\boldsymbol{x}, t), (\boldsymbol{x}', t')\big) = k_x(\boldsymbol{x}, \boldsymbol{x}')\, k_t(t, t'), \tag{4.26}$$

where the $k_t$ term models the correlations between tasks. An example is the *Linear Model of Coregionalization* (LMC) [67], that enables to represent many correlated function on the basis of shared latent components. Computational costs are clearly increased at the benefit of much more representational power.
Multitask GPs are particularly effective if small amounts of data are available for each task, as they exploit statistical correlation between tasks.

Gaussian process have been extensively used in this thesis, as it will be presented in the next chapter.

### 4.3.2. Artificial Neural Networks

Artificial Neural Networks (ANN), or simply Neural Networks (NN) are a class of very diffused and versatile learning models. Historically, they are inspired to biological mechanisms of the brain, and they tent to mimic its ability to manage complexity in heterogeneous problem classes. With time, the interest in them as shifted from biological plausibility to their direct application. Indeed, they proved to have extraordinary capacity in approximating complex functions, and so became very useful instruments both in science and industry [63, 66].
This subsection is inspired by the treating that feedforward neural networks receive in

authoritative references on machine learning techniques, namely in chapter 5 of [63] and in chapter 6 of [66].

**Structure and functions** A feedforward neural network is based on an elementary unit: the *artificial neuron*. Such unit is able to receive an ensemble of numerical inputs from other units, applies to them a linear combination using weights and biases, and then applies a non linear transformation, called *activation function*. The activation function plays a crucial role: it makes the model able to learn non linear tendencies in the dataset; without it in fact, the network would be nothing else than a linear combination of inputs, independently on its depth.
A NN is indeed a collection of layers of neurons. The depth of the network is defined by the number of layers involved: the typical structure is made of a first layer corresponding to the training data input, a series of so-called *hidden layers*, and the final output layers which gives the estimation of the function.
The number of neurons involved in each hidden layer is the measure of the width of the model. The number of neurons in the input layer reflect the dimension of the input, as well as the number of neurons in the output layer reflect the dimension of the output. Therefore, if the function f to be modeled is has scalar values, there would be one neuron in the output layer.
NNs are given the name feedforward because *information flows through the function evaluated by* $\boldsymbol{x}$*, through the intermediate calculations used to define f, and finally to the output* $\boldsymbol{y}$ [66]. No feedback mechanisms is included, meaning that the outputs of the model are not fed back into itself.
Consider an input vector con $\boldsymbol{x} \in \mathbb{R}^d$. For the $l^{th}$ layer, with $l = 1, ..., L$, define:

$$\boldsymbol{a}^{(l)} = \boldsymbol{W}^{(l)}\boldsymbol{z}^{(l-1)} + \boldsymbol{b}^{(l)}, \qquad \boldsymbol{z}^{(l)} = \phi\left(\boldsymbol{a}^{(l)}\right), \tag{4.27}$$

where: $L$ is the number of hidden layers, $\boldsymbol{z}^{(0)} = \boldsymbol{x}$ is the initial input, $\boldsymbol{W}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ is the weight matrix, $\boldsymbol{b}^{(l)} \in \mathbb{R}^{n_l}$ is the bias vectors, $\phi(\cdot)$ is the chosen activation function, applied element by element. Finally, the output layer computes:

$$\boldsymbol{a}^{(L+1)} = \boldsymbol{W}^{(L+1)}\boldsymbol{z}^{(L)} + \boldsymbol{b}^{(L+1)}, \qquad \boldsymbol{z}^{(L+1)} = \phi\left(\boldsymbol{a}^{(L+1)}\right) \tag{4.28}$$

The overall function that the network implements is then given by:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{z}^{(L+1)}, \tag{4.29}$$

where $\boldsymbol{\theta} = \{\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\}_{l=1}^{L+1}$ represents the whole parameters ensemble. As pointed out in [63], *the neural network function is differentiable with respect to the network parameters*. This is an important property and will play a huge role in the network training.

**Activation functions** The role played by the activation function $\phi(\cdot)$ is fundamental. As already mentioned its presence is crucial to introduce a non-linear character in the NN model. Purely linear activation functions indeed would collapse the network in a simple linear model, independently on the number of layers included. Complex relationships in the data can be modeled thanks to the presence of the activation functions.
As for the case of the kernel choice for Gaussian processes, the choice of the activation function depends on the nature of the data and the distribution of target variables. The most common activation functions are:

- **Logistic Sigmoid ($\sigma$):**
$$\phi(\boldsymbol{a}) = \frac{1}{1 + e^{-\boldsymbol{a}}}, \tag{4.30}$$
historically very used. It has a saturative character, which creates problems in gradient-based learning, so its use is nowadays discouraged.

- **Hyperbolic Tangent (tanh):**
$$\phi(\boldsymbol{a}) = \tanh(\boldsymbol{a}), \tag{4.31}$$
it is closely related to the sigmoid and also historically used. For the same reason it is not widely used anymore

- **Rectified Linear Units (ReLU):**
$$\phi(\boldsymbol{a}) = \max(0, \boldsymbol{a}), \tag{4.32}$$
represents the nowadays standard activation function for modern neural networks. It solves the saturation problem while being extremely simple from the computational point of view.

Other activation functions do exist. An example is the Softplus function $\phi(\boldsymbol{a}) = \log(1 + e^{\boldsymbol{a}})$, introduced as a softer version of the ReLU. Indeed it is differentiable everywhere, which one might think is an advantage. The use of the Softplus is usually discouraged, since it was shown to produce worst result with respect to ReLU [66], highlighting a counterintuitive nature in activation functions performance. Activation functions for hidden units remain an active area of research and many useful types remain to be discovered.

**Probabilistic formulation and cost function** So far, NNs were seen as a generic class of parametric non linear models taking an input vector $\boldsymbol{x}$ and producing an output vector $\boldsymbol{y}$. Neural networks can also be given a probabilistic interpretation. Consider a regression problem for a single target variable t with real values. Assume that t has a Gaussian distribution with an $\boldsymbol{x}$-dependent mean, which is given by the output of the neural network

$$p(t|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(t|y(\boldsymbol{x}; \boldsymbol{\theta}), \sigma^2), \tag{4.33}$$

where $\sigma^2$ represents Gaussian noise.
Given a set of $N$ independent, identically distributed observations $\boldsymbol{X} = \boldsymbol{x}_1, ..., \boldsymbol{x}_N$ and their correspondent target values $\boldsymbol{t} = \{t_1, ..., t_N\}$, it is possible to construct the likelihood function as:

$$p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{\theta}, \sigma) = \prod_{i=1}^{N} p(t_i|\boldsymbol{x_i}, \boldsymbol{\theta}, \sigma). \tag{4.34}$$

The maximization of this likelihood is equivalent to the minimization of the sum of squares error function, which has the role of *cost function*, given by:

$$E(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{N} \|f(\boldsymbol{x_i}; \boldsymbol{\theta}) - t_i\|^2. \tag{4.35}$$

This can be easily extended to multidimensional target, as well as to non-independent observations, clearly introducing more complex minimization algorithm. Other assumptions for $p(t|\boldsymbol{x}, \boldsymbol{\theta})$ will generally lead to different cost functions.

**Backpropagation algorithm** Gradient-based descent algorithm are used to minimized the cost function $E(\boldsymbol{\theta})$ during the learning phase. The difficulty stands in the efficient computation of the cost function derivatives with respect to the huge number of parameters that can be involved in the network.
The *backpropagation* algorithm shows a computationally efficient and elegant solution to this problem. The core of this method is the chain rule for derivatives. $E(\boldsymbol{\theta})$ is defined on the training data $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^{N}$, were again $t$ is assumed to be scalar. Its mean on all observations is considered for simplicity:

$$E(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} E_i(\boldsymbol{\theta}), \tag{4.36}$$

where $E_i$ is the error for a single observation $(\boldsymbol{x_i}, t_i)$.
The application of gradient descent requires the computation of the derivatives $\nabla_{\boldsymbol{W}^{(l)}} E_n$

and $\nabla_{\boldsymbol{b}^{(l)}} E_n$. The net is made of a sequence of nested transformations, so that the gradient can be repeatedly computed using the chain rule. Doing it explicitly for every parameter would be computationally prohibitive in any case. Here is were the backpropagation algorithm becomes necessary.
First of all, an intermediate variable is defined:

$$\boldsymbol{\delta}_j^{(l)} = \frac{\partial E_n}{\partial \boldsymbol{a}_j^{(l)}}, \tag{4.37}$$

which is the error derivative with respect to the net input $\boldsymbol{a}_j^{(l)}$ of the neuron $j$ in the layer $l$. The $\boldsymbol{\delta}^{(l)}$ vector is called *local error* (or *sensitivity*) of the layer $l$. In the output layer $(L+1)$, the derivative is computed as:

$$\boldsymbol{\delta}^{(L+1)} = \nabla_{\boldsymbol{a}^{(L+1)}} E_n = \frac{\partial E_n}{\partial y} \odot \phi'(\boldsymbol{a}^{(L+1)}), \tag{4.38}$$

where $\odot$ stands for element by element product.
Once $\boldsymbol{\delta}^{(L+1)}$ is known, the error can be propagated backwards, through the hidden layers. For every layer $l = L, ..., 1$ it holds:

$$\boldsymbol{\delta}^{(l)} = \left((\boldsymbol{W}^{(l+1)})^\top \boldsymbol{\delta}^{(l+1)}\right) \odot \phi'(\boldsymbol{a}^{(l)}). \tag{4.39}$$

This is the core relationship of the backpropagation algorithm. The error of the next layer is propagated backward to the preceding one, and is modulated by the derivative $\phi'$ of the activation function.
Once every error $\boldsymbol{\delta}^{(l)}$ is known, the cost function gradients with respect to the network parameters can be found as:

$$\frac{\partial E_n}{\partial \boldsymbol{W}^{(l)}} = \boldsymbol{\delta}^{(l)} (\boldsymbol{z}^{(l-1)})^\top, \qquad \frac{\partial E_n}{\partial \boldsymbol{b}^{(l)}} = \boldsymbol{\delta}^{(l)}. \tag{4.40}$$

The training algorithm than proceeds in the following way:

1. **Forward pass**: compute every $\boldsymbol{a}^{(l)}$ e $\boldsymbol{z}^{(l)}$ for every layer, up to the output $y$.
2. **Backward pass**: compute $\boldsymbol{\delta}^{(L+1)}$ and backward propagate it.
3. **Parameters update**: the weights and bias are ,

$$\boldsymbol{W}^{(l)} \leftarrow \boldsymbol{W}^{(l)} - \eta \frac{\partial E_n}{\partial \boldsymbol{W}^{(l)}}, \qquad \boldsymbol{b}^{(l)} \leftarrow \boldsymbol{b}^{(l)} - \eta \frac{\partial E_n}{\partial \boldsymbol{b}^{(l)}}, \tag{4.41}$$

where $\eta$ is the *learning rate* of the training phase. The learning rate is an hyperpa-

rameter of the model, hence is fixed apriori (see Sec A.1.2).

The crucial aspect introduced by the backpropagation algorithm is that the computational cost grows linearly with the number of connections. This makes the computation possible even for very wide and deep networks. However, numerical stability depends on the activation function and learning rate choice. Examples of saturative activation function as the logistic sigmoid and the hyperbolic tangent, make the gradients very small, producing the so-called *vanishing gradient* phenomenon, which instead is partially mitigated by ReLU functions. The backpropagation algorithm represents one, if not the central element of modern machine learning. Thanks to it, multilayer networks can be efficiently trained. It is thus the foundation of deep learning.

**Universal approximation theorem** A theoretical result of crucial importance is the *universal approximation theorem* [69–71]. It states that a neural network with a single, sufficiently wide, hidden layer, equipped with non linear and limited activation functions, can approximate any continuous function defined on a compact of $\mathbb{R}^d$, with arbitrary precision:

$$\forall f \in C([0,1]^d),\ \forall \varepsilon > 0,\ \exists\ \boldsymbol{\theta} : \sup_{\boldsymbol{x} \in [0,1]^d} |f(\boldsymbol{x}) - f(\boldsymbol{x}; \boldsymbol{\theta})| < \varepsilon. \tag{4.42}$$

An unlimited expressive power is then contained in neural networks, at least in theory. In practice the choices made in depth and width, as well as on activation functions, influence a lot the efficiency and stability of training.

**Shallow vs deep networks** The so-called *shallow networks* are constituted of a single hidden layer. As the universal approximation theorem testifies, they have played an important role in the theoretical development of neural networks. They also conserve a practical interest in many applications. They posses the universal approximation property and are simpler to train and analyze. However, as the complexity of the data increases, it is usually more convenient to use deep networks. In this thesis, shallow networks have been used as an alternative to Gaussian processes, in order to test the validity of the proposed approach. The choice stands in the low dimensions of the dataset at disposal. A more detailed discussion will be held in the next chapter.

This concludes the chapter on Machine Learning. However, many additional knowledge are needed to understand the choices made in this work. For this reason, important pieces of information are found in App. A, including general training techniques for machine learning, reduced-order modeling and Monte Carlo methods.

# 5 | Approaches and Strategies to Mitigate Pandemonium Effect

This chapter presents the work carried out during the thesis. This chapter presents the description of the problem analyzed in this thesis, and the development of the methodology used to tackle it, with particular attention on the choice and training of ML algorithms. The theoretical and experimental background on which it is based has been presented in the previous chapters, including neutrinos properties and flavor oscillations (Ch. 1), the main characteristics of JUNO experiment (Ch. 2), the reactor antineutrino spectrum and the problems associated to its modeling (Ch. 3), and the fundamentals concepts of machine learning (Ch. 4), further extended with methodological details in App. A.
The main goal of this research is the development and application of ML techniques for the modeling of the reactor antineutrino spectrum. In particular, the specific purpose is to mitigate the so-called Pandemonium effect (Sec. 3.4), representing one of the main problematic aspects in the analytical estimation of the reactor antineutrino spectrum through the summation method. High fidelity data obtained from experiments such as Daya Bay [31] and RENO [32] have been crucial to identify this discrepancies, as described in Sec. 3.3.

## 5.1. Description of the Problem and Proposed Approaches

Sec. 3.4 described Pandemonium effect as an experimental bias affecting the gamma spectroscopy measurements required to predict the electron antineutrino (and $\beta$) spectrum of any isotope. As pointed out, this effect qualitatively consists in an underestimation of the low energy part of these spectra, as a consequence of missing transitions associated to the emission of high energy gamma rays, which due to HPGe detector inefficiencies could not be measured in experiments of the past.
Solving this issue would have many positive repercussions not only in the context of reactor antineutrino spectrum modeling (see Sec. 3.3), but also in other scientific and

engineering branches such as nuclear physics and reactor decay heat [55] evaluations.
Sec. 3.5 introduces the TAGS technique as the experimental solution to this problem. The employment of high efficiency NaI(Tl) crystals allows to collect the whole de-excitation energy, and provide a more reliable reconstruction of the $\beta$ decay scheme of daughter isotopes.
The main issue in performing a systematic TAGS experimental campaign devoted to overwrite the Pandemonium data can be found in the long time required for executing this kind of experiments as well as their costs. Also, a comprehensive list of the nuclei affected by Pandemonium is not present at the moment. It is instead available, and reported by IAEA in [55], a list of priority candidates sensitive to this bias. In total, 94 isotopes are involved. However, it is still not clear what is the degree of confidence associated to this list. From a general perspective, all the past measurements should be performed again with the TAGS technique, which is simply an impractical solution.

The solution proposed by this work is instead to use machine learning regression models, employing Gaussian processes (Sec. 4.3.1) and Artificial Neural Networks (Sec. 4.3.2), to extract possible common features that the Pandemonium effect produces to the spectra of isotopes for which TAGS measurements have already been taken.
Part of the database isotopes indeed present a couple of measurements, one performed with HPGe, likely Pandemonium affected, and another performed with the TAGS technique, Pandemonium corrected. Exploiting this information, the idea is to train a model able to predict a *Pandemonium corrected* version of the antineutrino spectrum of an isotope, on the basis of an input which does not include any information on the TAGS measurement. In this way, it would be possible to use this model to make prediction of antineutrino spectra for isotope which have not yet been observed with the TAGS technique. The fact that the input must not contain information on the *correct*, hence TAGS, version of the spectrum is a crucial one.

Among the data available for this research, up to 52 nuclides are adopted, for which NaI(Tl) detectors have been employed to correct Pandemonium effect. For all these isotopes it is possible to compare the HPGe measurements with the TAGS one.
For this reason, they are the reference data to study and correct Pandemonium effect with a ML approach. These nuclides are reported in Table 5.1 and, from now on, will be referred to as *TAGS isotopes*.

An example of comparison between the HPGe and TAGS measurements for these nuclides is shown in Fig. 3.6, in particular, there, $^{87}$Br and $^{142}$Ba are reported.

| TAGS Isotopes | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $^{86}$Br | $^{87}$Br | $^{88}$Br | $^{90}$Rb | $^{90m}$Rb | $^{91}$Rb | $^{93}$Rb | $^{94}$Rb | $^{93}$Sr |
| $^{95}$Y | $^{101}$Nb | $^{105}$Mo | $^{102}$Tc | $^{104}$Tc | $^{105}$Tc | $^{106}$Tc | $^{107}$Tc | $^{140}$Cs |
| $^{141}$Cs | $^{142}$Cs | $^{141}$Ba | $^{142}$Ba | $^{143}$Ba | $^{144}$Ba | $^{145}$Ba | $^{142}$La | $^{143}$La |
| $^{144}$La | $^{145}$La | $^{145}$Ce | $^{146}$Ce | $^{147}$Ce | $^{148}$Ce | $^{146}$Pr | $^{147}$Pr | $^{148}$Pr |
| $^{148m}$Pr | $^{149}$Pr | $^{151}$Pr | $^{149}$Nd | $^{151}$Nd | $^{153}$Nd | $^{154}$Nd | $^{155}$Nd | $^{152}$Pm |
| $^{153}$Pm | $^{155}$Pm | $^{156}$Pm | $^{157}$Pm | $^{157}$Sm | $^{158}$Sm | $^{158}$Eu | | |

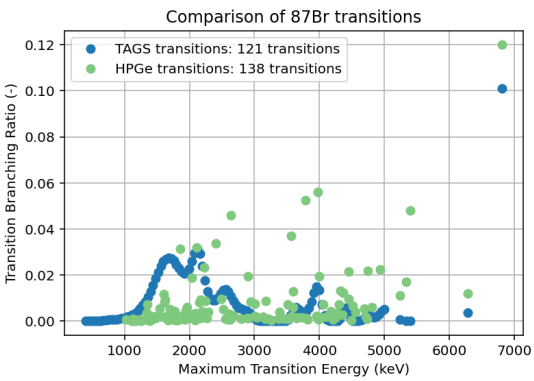
Table 5.1: The table summarizes the 52 isotopes for which a TAGS measurement has been taken, enabling comparison with HPGe measurements. These isotopes are the core element of this research.

### 5.1.1. The Transition Approach

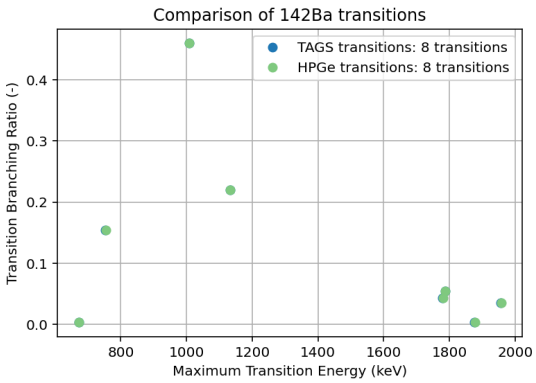
RenShape (RS), introduced in Sec. 3.6, collects many useful information about the TAGS isotopes, and many other nuclides. Of particular interest for this research are the single $\beta$ transitions spectra and their respective branching ratios, by which is possible to build the complete isotope spectrum through the summation method, with related uncertainties (see Eq. 3.4). Moreover, for the TAGS isotopes it is also possible to confront the information in RS with the same one coming from the ENSDF library, hence related to the Pandemonium affected spectra. The usage of renshape is crucial in this point of the research, being able to compare the antineutrino spectra prior to the correction (i.e., Pandemonium affected) and after (i.e., TAGS)

In principle, it is possible to study the Pandemonium effect at the level of the single beta transitions. This would be the ideal treatment since this experimental bias is very related to the phenomenology of the single transitions. For this reason, this was the first idea proposed at the beginning of this research.

However, while being extremely interesting, already with a brief analysis of the material at disposal, some practical problems emerge. As a first issue, the number of transitions involved in the de-excitation scheme of different isotope is obviously not always the same. This poses an initial problem in defining a computational structure adaptable to usual ML codes, which require rectangular matrices, where each example isotope needs to be associated to same number of input/output elements. Moreover, and most importantly, the number of transitions involved in $\beta$ decay changes between the HPGe and TAGS measurement, with different proportions from isotope to isotope. An example is shown, for $^{87}$Br and $^{142}$Ba, in Fig. 5.1. These nuclides are chosen with reference to the plots of Fig. 3.6 in Sec. 3.5.




(a) The branching ratios of the transitions of $^{87}$Br are plotted against their respective endpoint energies, for both the TAGS (in blue) and the HPGe (in green) measurements.




(b) The branching ratios of the transitions of $^{142}$Ba are plotted against their respective endpoint energies, for both the TAGS (in blue) and the HPGe (in green) measurements.

Figure 5.1: Examples of transition properties (branching ratios and endpoint energies) comparison between HPGe and TAGS measurements. Left figure: the comparison is shown for $^{87}$Br. The isotope shows a significative Pandemonium effect, indeed the transitions properties change in a visible way. Right figure: the comparison is shown for 142Ba. Pandemonium effect is practically absent, indeed the transitions properties almost exactly superpose for HPGe and TAGS.

As it is possible to see, in the case of $^{87}$Br (Fig. 5.1a) the number of transitions changes of about 10% before and after the TAGS measurement, while in the case of $^{142}$Ba (Fig. 5.1b), the number of transitions stays exactly the same, together with their properties (branching ratios and endpoint energies): this last case indeed presents no Pandemonium effect in practice. These are only two examples of a wide phenomenology. The first preliminary results showed that building a model able to predict not only the properties and spectra of these transitions, but also their number, each time different from example to example, becomes very difficult and intricate. For these reasons, this approach has been discarded. However, since the transitions play a crucial role in Pandemonium effect, it has been found a way to include them in the actual approach of this research.

### 5.1.2. The Shape Approach

The shape approach is the actual approach followed in this thesis. The main idea is to train a machine learning method to reproduce the overall shape of the TAGS antineutrino spectrum for a certain set of isotopes, based on input information such as nuclear

properties, and characteristics of the Pandemonium affected spectrum. Remind that information about the TAGS spectrum must be kept out of the input data. Instead, it has to be used only as a target to be learned. The general, simplified idea is represented in the block diagram of Fig. 5.2.

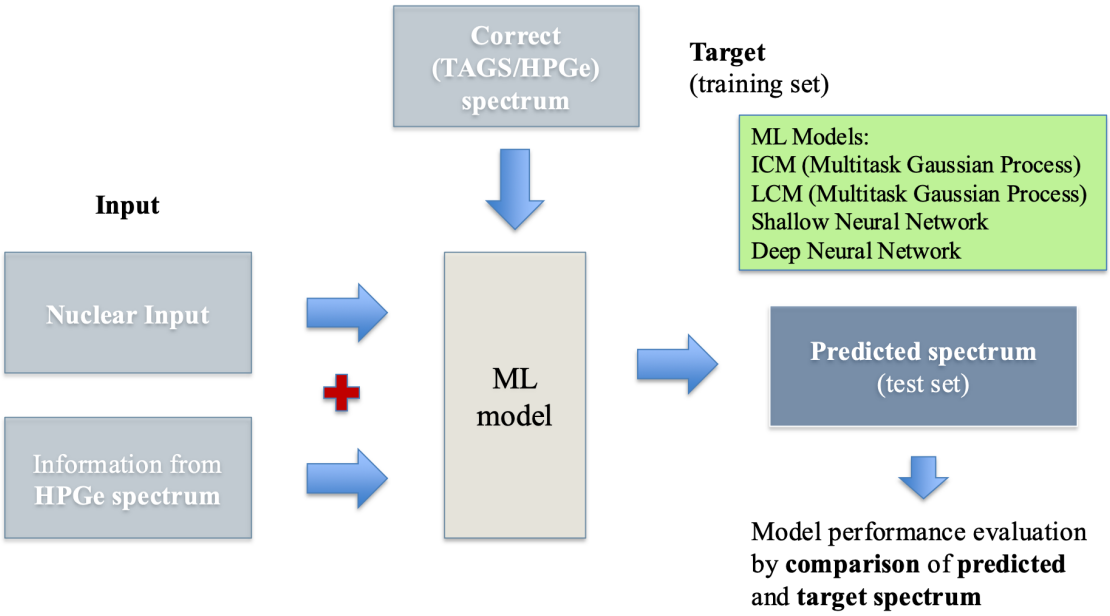



Figure 5.2: Simplified block diagram of the ML regression implemented for the purposes of this thesis.

As the name suggests, in this approach the interest is in the shape of the various spectra. For this reason, a preprocessing is required for the target data in the training set. The binned spectra of each isotope must be indeed normalized to have unitary sum. In this way, the information related to the overall $\beta$ decay branching ratio of each specific isotope, which is assumed to be reliable and not depending on Pandemonium effect, is taken out of the model. Moreover, this intermediate step permits to obtain spectral data with the same order of magnitude for every isotope.
Finally, when the training of the model is completed, the predictive capabilities of the model will produce an estimate of the shape of the antineutrino spectra of isotopes in a test set, to be compared with the actual spectrum obtained from the unitary normalization of the TAGS measurement. The performance of the model is then evaluated both on the single isotopes spectra and on the total antineutrino spectrum of some reference actinide. In this work, the choice is to focus on $^{235}$U as a thermal fission antineutrino spectrum example, and on $^{238}$U as a fast fission example. Different models have been trained to accomplish the purpose described here, and comparisons between their performance has

been evaluated. For this reason, the employed dataset is always the same.

The next sections will describe the choices taken to build the input and target dataset, and the various model explored in this work.

## 5.2. Dataset Construction

This section is dedicated to all the descriptions of the choices taken in building the dataset used in this work. The dataset is composed of the following three subparts:

- Nuclear Input $\boldsymbol{X}_1$,
- Coded information on the Pandemonium affected (HPGe) measurement $\boldsymbol{X}_2$,
- Target $\bar{\nu}_e$ spectrum $\boldsymbol{Y}$.

where $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ taken together form the input data $\boldsymbol{X}$. In practice, $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be respectively represented has $N_n \times N_X$ and $N_n \times N_Y$ matrices, where $N_n$ is the number of isotopes composing the dataset, $N_X$ is the total number of input variables used to represent an isotope, and $N_Y$ is the number of bins considered for the target spectrum.

In Sec. 3.5, Fig. 3.6 shows that Pandemonium effect is not always important or present in measurement obtained before the employment of the TAGS technique. It is then interesting to consider the following situation. The scope of the model is to predict a Pandemonium corrected version of the $\bar{\nu}_e$ spectrum. On the other hand, this is not always needed, since there are cases in which the previous information results to be already satisfactory. An additional, and powerful property of the model would then be the ability to recognize whether a correction is actually needed or not.
Accordingly, the choice has been to include a set of isotopes which are in this sense complementary to the TAGS isotopes. This set should be composed of those isotopes for which it can be assumed, with good probability, that the HPGe measurements are not affected by Pandemonium. This set of isotopes will be referred as *SAFE isotopes*. A primary concern is then to choose these nuclides.

### 5.2.1. SAFE Dataset Selection

It is not obvious to claim that a certain HPGe measurement is not affected by Pandemonium. The fundamental assumption has to be that the nuclides considered in the IAEA list of [55] are the only that suffer Pandemonium effect. With this hypothesis, it is possible to easily choose the isotopes of interest.

As it is explained in Sec. 3.6, the RenShape library consists of antineutrino spectra coming from different libraries. The ENDF-B library is the most reliable one. Therefore, the SAFE isotopes are chosen to be all those isotopes of the RenShape library that pass the following selection:

1. The isotope belongs to the subset of RenShape library coming from the ENDF-B library,
2. The isotope is not included in the IAEA Pandemonium priority list.

However, this selection results to be too wide, since it contains 338 elements. This number is much larger than the number of TAGS isotopes (52). As a consequence, the majority of the overall dataset would be composed of SAFE isotopes, leaving small space to the TAGS ones, which play the main role. Therefore, an additional selection is introduced.
Among the big number of SAFE nuclei at disposal, the ones selected to compose the SAFE dataset are the first 52 that are closer to the TAGS nuclei from the point of view of the nuclear input variables, which will be described in the next section. In this way, SAFE and TAGS isotopes are equally represented. This preprocessing is performed by computing the euclidean distances in the space of nuclear input parameters for any of the SAFE and TAGS nuclides. The first 52 SAFE candidates with the smaller distance with respect to the TAGS dataset pass the final selection and compose the actual SAFE dataset.
By means of Principal Component Analysis (PCA) (see Sec. A.2.1), it is possible to visually represent the final choice in a 2D plot in the space of the principal components, as shown in Fig 5.3.

After this selection, the final dataset includes 104 isotopes:

- **52 TAGS Isotopes:** for these nuclides, both the TAGS and HPGe measurements are available. The information on the Pandemonium affected spectrum is taken from the HPGe measurement. The target spectrum is the TAGS one.
- **52 SAFE Isotopes:** for these nuclides, the HPGe measurement is considered to be reliable. The information on the Pandemonium affected spectrum is taken from the HPGe measurement. The target spectrum is again the HPGe one.

The inclusion of the SAFE dataset enables the model to recognize the isotopes for which the information on the Pandemonium affected measurement is coherent or not with the correct target spectrum, and decide whether a correction is to be applied or not. The complete selection allows to equally represent SAFE and TAGS nuclides in the overall dataset.

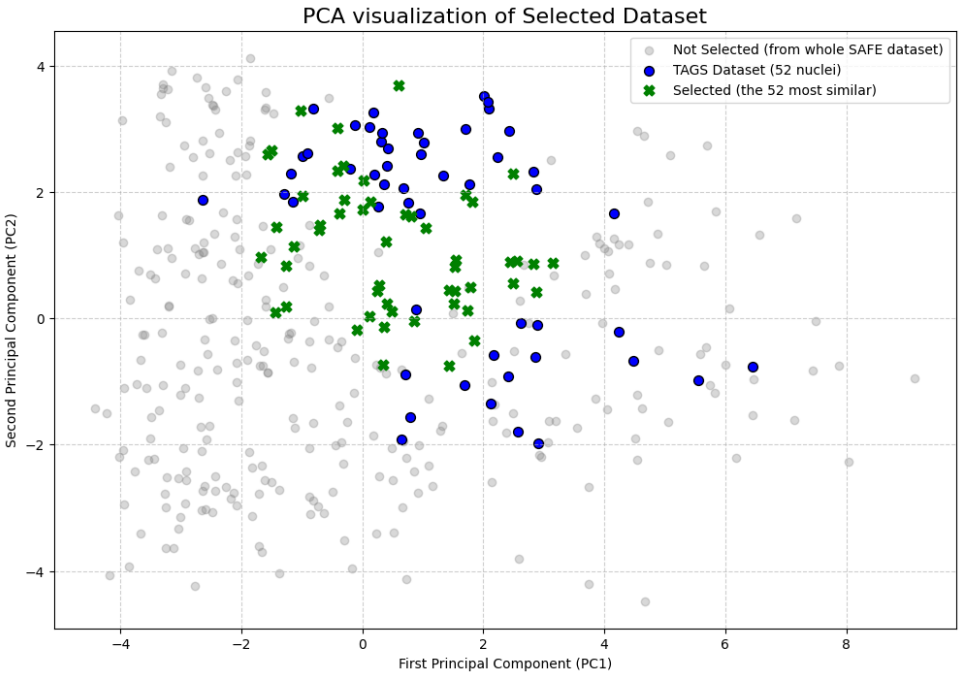



Figure 5.3: Principal component representation of the selection performed on the large number of candidates. The selected SAFE data (in green) are the ones closer to the TAGS data (in blue).

### 5.2.2. Training-Test Split

For purposes that will be clearer in Sec. 5.2.4, the training-test splitting is performed right after the definition selection of the SAFE isotopes. In particular, the splitting is performed randomly using the *test train split* function of the *sklearn* library in *Python*. The procedure is applied to TAGS and SAFE dataset separately, obtaining a TAGS training set, a TAGS test set, a SAFE training set and a SAFE test set.
The training sets are the 85% of the respective datasets, while the test ones are the remaining 15%. Then, the overall training and test sets are given as the union of the respective TAGS and SAFE sets. The training-validation splitting, introduce to monitor the performance of the models is introduced at the end of the section on dataset construction, in Sec. 5.2.6.

### 5.2.3. Nuclear Input

In Sec. 5.2.1, the SAFE dataset selection is described. The main idea is to chose, among those isotopes that passed the wide selection, the ones that are closer to the TAGS isotopes, from the point of view of the nuclear input. In this section, a review of the nuclear input values considered in the dataset of this work is performed.

Under the name nuclear input there fall two categories of quantities: the nuclear properties, related to intrinsic properties of the isotopes of interest and additional variables extracted from the knowledge of their HPGe measurement.
The nuclear properties considered here are:

- Atomic number (Z),
- Neutron number (N),
- $\beta$ decay Q-value, measured in keV,
- $\beta$ decay half-life $T^{\beta}_{\frac{1}{2}}$, measured in seconds,
- Square of the atomic number ($Z^2$): this feature emphasizes the dependence on nuclear charge and roughly scales with Coulomb effects,
- Proton fraction: ratio of protons to total nucleons, representing the proton-to-baryon fraction. It is computed as:

$$\frac{Z}{Z+N}, \tag{5.1}$$

- Isospin asymmetry parameter: it quantifies the neutron–proton imbalance normalized by the mass number. It is computed as:

$$\frac{(N-Z)^2}{A}, \quad \text{where } A = Z + N. \tag{5.2}$$

  This term is often related to the symmetry energy contribution in nuclear models.

In addition, various quantities are extracted from the transition properties (branching ratios and endpoint energies) related to HPGe measurements. Indeed, these can be used to give a strong statistical description of Pandemonium effect for a specific nucleus. Here's a list of those considered in this work.

- Number of transitions involved $n$,
- Maximum transition endpoint energy $E_0^{max}$: this quantity is included because there are cases in which it differs from the $\beta$ decay Q-value,
- Minimum transition endpoint energy $E_0^{min}$,
- Entropy of the transition branching ratios, computed as:

$$S_{br} = -\sum_i p_i \log p_i \text{ with } p_i = \frac{br_i}{\sum_j br_j}, \tag{5.3}$$

where $br_i$ is the branching ratio associated to the $i^{th}$ transition,

- Entropy of the endpoint energies, computed as:

$$S_{E_0} = -\sum_i w_i \ln(w_i) \text{ with } w_i = \frac{br_i E_{0,i}}{\sum_j br_j E_{0,j}}, \tag{5.4}$$

where $E_{0,i}$ are the transition endpoint energies associated to the $i^{th}$ transition,

- Weighted mean on the transition branching ratios of the transition endpoint energies, computed as:

$$\hat{E}_0 = \frac{\sum_i br_i E_{0,i}}{\sum_i br_i}, \tag{5.5}$$

- Mean transition endpoint energy, computed as:

$$\bar{E}_0 = \frac{1}{n}\sum_{i=1}^{n} E_{0,i}, \tag{5.6}$$

- Standard deviation of the transition endpoint energies, computed as:

$$\sigma(E_0) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(E_{0,i} - \bar{E}_0)^2}, \tag{5.7}$$

- Median transition endpoint energy, corresponding to the 50th percentile of the transition endpoint energy distribution $\tilde{E}_0$ ,
- Effective transition endpoint energy: represents the transition endpoint energy that divides the cumulative transition intensity distribution into two equal halves. It corresponds to the energy below which 50% of the total transition intensity occurs, and above which the remaining 50% lies. Formally, it is defined as:

$$\sum_{E_{0,i} \leq E_{\text{eff}}} br_i = \frac{1}{2}\sum_i br_i. \tag{5.8}$$

Recalling Sec. 5.1.1, the inclusion of these quantities is intended to inform the model on the characteristics of the $\beta$ de-excitation scheme of daughter isotopes, in the tentative to discover a transition based dependence of Pandemonium effect. They are an explicit representation of the properties of the $\beta$ transition measured with HPGe detectors.

All these input quantities can be computed for any kind (TAGS or SAFE) of isotope, they

are indeed always related to the HPGe measurement. This is crucial for the application of the model to isotopes for which TAGS measurements are not yet available.

Finally, the union of the two sets of quantities above composes the first part of the model input data, namely the nuclear input matrix $\boldsymbol{X}_1$, including 17 values for each isotope.

### 5.2.4. Coded Information from the HPGe Measurements

The information contained in $\boldsymbol{X}_1$ could be considered as a satisfactory description of the properties of the isotopes and their HPGe antineutrino spectrum. However, in this work an additional set of quantities is considered, with the aim to enforce such description.
In Sec. A.2.2, Autoencoders (AE) are presented as non linear techniques for reduced-order modeling. It is indeed possible to train an AE to extract a latent representation of a dataset of interest.
In this context, autoencoders are employed to obtain a codified representation of the branching ratios and endpoint energies. This latent, or codified representation is the second part of the model input data, referred to as $\boldsymbol{X}_2$.
The AE working principle can be summarized as:

$$\begin{pmatrix} [br_1, ..., br_n]_{HPGe} \\ [E_{0,1}, ..., E_{0,n}]_{HPGe} \end{pmatrix}_k \xrightarrow{\text{AE}} \Big([C_1, ..., C_m]\Big)_k , \tag{5.9}$$

where $n$ is the number of transitions involved in the HPGe measurement of the $k^{th}$ isotope, while $m$ is the dimension of the latent representation. As it is clear, the AE takes as an input two arrays of values for every isotope, respectively containing the branching ratios and the endpoint energies of the transitions measured with HPGe crystals.
Importantly, the AE algorithm is required to be trained exclusively on the transitions of the training set. The transitions of the test set must not be included during training to avoid data leakage in the input. Conversely, once training is completed, the AE model transforms the transitions arrays into their corresponding latent representation for all the isotopes in the training and test set.

As any ML model, the input of the AE requires some prepocessing to stabilize and enhance the model convergence during the training phase. First, since the number of transitions is different from isotope to isotope, the array containing the branching ratios and the endpoint energies are subject to a *padding* operation. In practice, the dimensions of this arrays are made homogeneous through a process which is recognized by the autoencoder and does not alter the information in them.

Secondly, a logaritmic transformation of the type:

$$\log([a_i, ..., a_n] + \varepsilon) \text{ with } \varepsilon = 10^{-10} \tag{5.10}$$

is performed on both the input arrays. This is done in order to compress the dynamic range, reducing the influence of extreme values and making distributions more symmetric. The $\varepsilon$ value is added to avoid computational errors. Finally, the resulting arrays are subject to the standardization procedure discussed in Sec. A.1.5, in order to improve numerical stability, and enhance model convergence.

In the following, the details of the encoder architecture of the autoencoder are presented, together with details on its training phase. The AE is a two input artificial neural network, built using the *torch* library in Python. One input is used to compress the branching ratios, the other to compress the endpoint energies. In both cases, this is done through two hidden layers of neurons, respectively composed of 128 and 64 neurons. The bottleneck hidden layer then further compresses the overall information towards a shared representation of dimension $m = 6$. The mean squared error (MSE) is used as a loss function, a typical choice for AEs (see Sec. A.2.2).
The performance of the AE is evaluated via the splitting of the overall training set[1] into a training and a validation subsets, using the *random split* function of the torch library in Python. The validation set size represents the 10% of the overall training set. The splitting is initialized with an initial seed equal to 5.
A validation based early stopping (see Sec. A.1.3) is used to avoid overfitting, with a patience of 50 epochs. The early stopping activates when the variation of the validation loss reaches a minimum of $10^{-5}$. The global learning rate is initialized at $10^{-4}$, but its value changes during training for every parameter of the model, thanks to the employment of the Adam optimizer (see Sec. A.1.2). Moreover, a batch size of 50 has been chosen for both training and validation subsets.
With these choices a validation loss of about 0.095 is reached, together with a training loss of about 0.032. This signals an impressive compression of the transitions information by the autoencoder. The behavior of the training and validation loss during the learning phase is shown in Fig. 5.4.

The results of this complex procedure can be represented as a matrix $\boldsymbol{X}_2$ containing 6 values for every isotope of the training and test set. These values are a codified representation of the transitions involved in the HPGe measurement.

[1]The term overall is used here to indicate the training set of the regression models employed for the spectral prediction. The AE training and validation set are a subset of the overall training set.

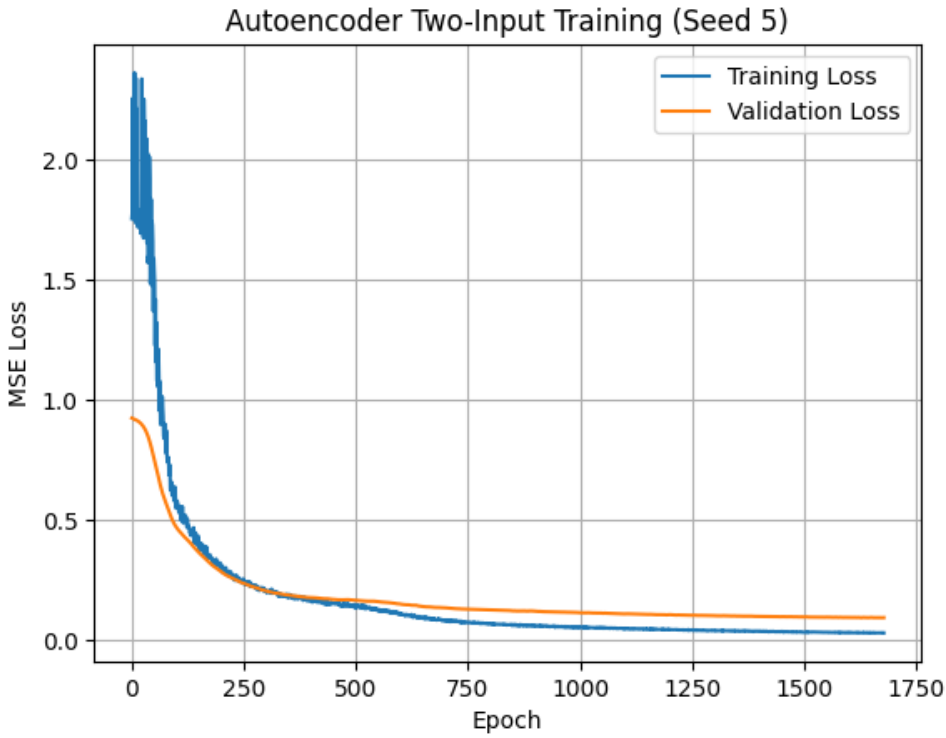



Figure 5.4: The trends of the autoencoder training (in blue) and validation (in orange) losses are plotted against the number of epochs. No overfitting appears, confirming the effectiveness of the validation based early stopping.

The union of the $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ collections of inputs quantities are put together, composing the complete input $\boldsymbol{X}$ for the ML regression models considered in this thesis. In the end, the input $\boldsymbol{X}$ is made of 23 variables per isotope.

Finally, the input $\boldsymbol{X}$ is subject to the standardization procedure, for similar reasons to the case of the autoencoder.

### 5.2.5. Preprocessing of the Target $\bar{\nu}_e$ Spectrum

The discussion of Sec. 5.2.1 clarifies the role played by the TAGS and SAFE isotopes in the dataset of this work. The target $\bar{\nu}_e$ spectrum of the TAGS isotopes is the one obtained from TAGS measurements, while for the SAFE case, the spectrum retrieved from HPGe measurements is considered as reliable, and therefore taken as a target for the regression model.

Independently on this distinction, a series of preprocessing steps are performed on the target spectra of every isotope, again supported by the argument of numerical stability and dynamic range compression.

Every isotope is indeed characterized by a different Q-value, resulting in a different number of bins for every spectra. As anticipated above, ML algorithms require instead data structures with homogeneous dimension. Therefore, every spectrum is projected onto an energy array ranging from 0 to 15 MeV, and null values of intensities are added to each array after the bin corresponding to their Q-value. In this way, every spectrum has the same number of bins, but the spectral information is kept intact. 15 MeV is an arbitrary value, larger than the maximum Q-value of the dataset. The binning of the target spectra is then set to 250 keV, in accordance with the Daya Bay measurement case. Accordingly, each spectrum can be represented as an array with $N_Y = 60$ bins.

In agreement with the discussion on the shape approach of Sec. 5.1.2, every spectra is then normalized to have unitary sum. In this way, the interest is concentrated on the shape of each spectrum, and not on their absolute contributions to the overall reactor antineutrino spectrum. The value of the normalization, which is equal to the total $\beta$ decay probability, is obviously stored, in order to correctly evaluate the antineutrino spectrum of the reference actinides after the model prediction.

Moreover, both the logarithmic transformation and the standardization are applied to the collection of target spectra. In this context, the logarithm is used also to guarantee that the final prediction of the model is always positive. Once prediction phase is completed, clearly these two operations must be inverted. The results are positive definite spectral shape, to be compared with the target spectra. An interesting question regards the ability of the model to predict spectral shapes that preserve the unitary normalization.

**PCA for Gaussian Processes** As it has been anticipated above, this work proposes the use of different types of regression models, namely neural networks and Gaussian processes. While NNs are naturally prone to work with high dimensional data, this is not the case for GPs. On the other hand, they show better capabilities when employed in low dimensional dataset. This is the reason for the use of a reduced-order modeling technique as PCA, treated in Sec. A.2.1.

The extraction of the principal components from the collection of target spectra enables to recognize intrinsic energetic patterns contained in the dataset. The principal components of the spectral dataset can indeed be seen as basis functions, whose linear combinations weighted on the coefficients are able to span the whole set of energetic shapes. The training of the model is then performed assuming as a target $\boldsymbol{Y}_{PCA}$, containing the coefficients of the PCA for every isotope. Once training is completed, extacly as for the other preprocessing, PCA must be inverted to recover the actual spectral prediction of the model.

It is important to remember once again that data leakage (see Sec. A.1.5) in preprocessing must be avoided. Any information on the test set must not leak in the training set. Therefore, the principal components are extracted only from the training set, and it is assumed that also the test set can be decomposed with those patterns. The same reasoning can be applied to the validation set, which is defined at the end of the section.

Choosing the correct number of PCs to include is a critical aspect. If the number is very reduced, their linear combination may not be able to describe the whole variance in the data, on the other hand, if the number is very high, no effective order reduction occurs, and moreover the risk is to include components that only describe noisy patterns. Accordingly, it is important to monitor the MSE of reconstruction and the explained variance fraction as a function of the number of components, as shown in Fig. 5.5.

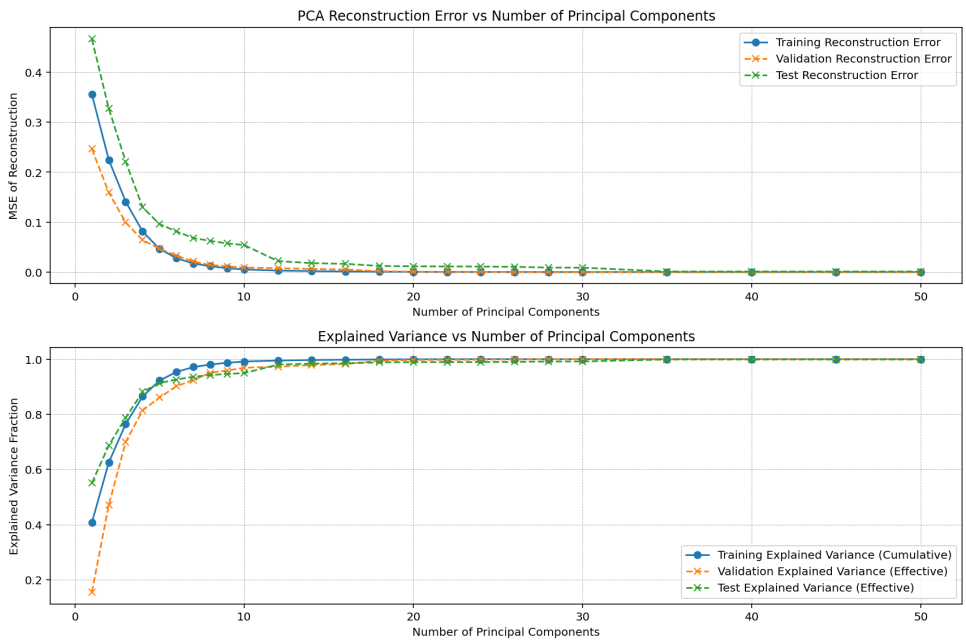



Figure 5.5: Upper figure: the mean squared error (MSE) of reconstruction is shown as a function of the number of principal components. Lower figure: the explained variance fraction is shown as a function of the number of principal components. These quantities are shown for the training (in blue), the validation (in orange) and the test (in green) sets.

With reference to Fig. 5.5, it is easy to see that with 15 number of components one can safely reconstruct practically all the information in the spectral dataset. Moreover, it is possible to recognize that the error metrics behave in a similar way for the three subsets as a function of the number of components. Indeed, the explained variance ratio is larger than 98% in all the subsets, while the MSE is always lower than 2%. This signals a strong

internal coherence in the shape patterns involved in the training and prediction phase. The first 15 principal components are shown in Fig. 5.6.

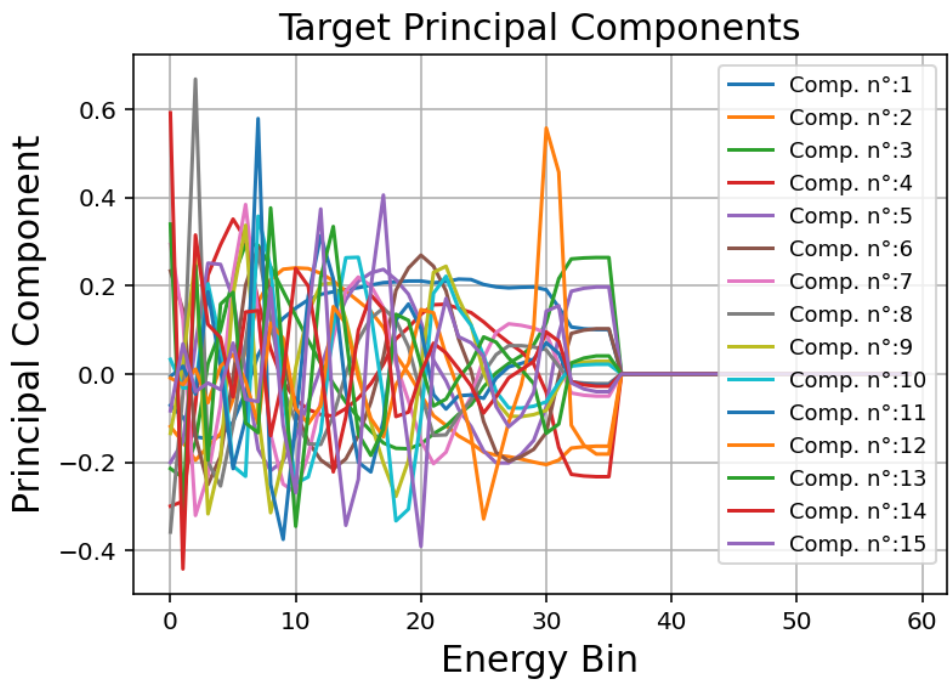



Figure 5.6: The figure shows the first 15 principal components of the spectral dataset. These represent the spectral patterns through which it is possible to reconstruct the majority of the information contained in the antineutrino spectra of the dataset. Note that, approximately after the $35^{th}$ bin, no signal is found, as a consequence of the padding operation.

Thus, PCA can be effectively used in the case of this thesis to reduce the dimensionality of the target vectors of a factor 4, importantly reducing the computational efforts of GP based models, and also with a training stabilization effect. It is instead not used when neural networks are employed.

### 5.2.6. Final Dataset

As a general conclusion to the section on the dataset construction, it is presented the training-validation set splitting. In Sec. A.1.3 validation based early stopping techniques are introduced as a simple but effective way to deal with overfitting. In Sec. 5.2.4, a training-validation splitting has already been introduced. However, such splitting was only referred to the AE training, and is not reproduced in the case of the regression models. A different splitting is instead performed in this case, previous to the application

of preprocessing to both the $\boldsymbol{X}$ and $\boldsymbol{Y}$. The train test split function from the sklearn library is used once again. The validation set dimensions are the 20% of the overall training set.

The final dataset is composed of three subsets: test, training and validation. Each set includes a number $N_n$ of isotopes and is identified by:

- an input matrix $\boldsymbol{X}_{subset}$, containing the 23 input variables for each isotope.
- a target matrix $\boldsymbol{Y}_{subset}$, containing the target spectral shape to be modeled, or the coefficients of a reduced-order representation if PCA is employed.

In the end, the three sets of samples are summarized in Table 5.2.

| | **Number of Isotopes** | **Number of variables** |
|---|---|---|
| $\boldsymbol{X}_{test}$ | $N_n = 16$ | $N_X = 23$ |
| $\boldsymbol{Y}_{test}$ | $N_n = 16$ | $N_Y = 60$ |
| $\boldsymbol{X}_{training}$ | $N_n = 70$ | $N_X = 23$ |
| $\boldsymbol{Y}_{training}$ | $N_n = 70$ | $N_Y = 60$ |
| $\boldsymbol{X}_{validation}$ | $N_n = 18$ | $N_X = 23$ |
| $\boldsymbol{Y}_{validation}$ | $N_n = 18$ | $N_Y = 60$ |

Table 5.2: The table summarizes the dimensions of the test, training and validation sets of the dataset of this work. Note that when PCA is applied: $N_Y = 60 \rightarrow N_{Y_{PCA}} = 15$.

## 5.3. Description of the Supervised Regression Models

Supervised regression models are used in this thesis to accomplish the purpose to mitigate Pandemonium effect. After the observation of the training set ($\boldsymbol{X}_{training}$, $\boldsymbol{Y}_{training}$), the generalization abilities of the model are first evaluated on the validation set ($\boldsymbol{X}_{validation}$), and then finally tested on the test set($\boldsymbol{X}_{test}$). The predictions $\hat{\boldsymbol{Y}}_{validation}$ and $\hat{\boldsymbol{Y}}_{test}$ are then compared against $\boldsymbol{Y}_{validation}$ and $\boldsymbol{Y}_{test}$. As it has been repeated extensively in this chapter, the proposal of this thesis is to evaluate the performance of various types of model. In this section, the proposed alternatives are described, both from the architecture point of view and the training choices one. In particular, different versions of Gaussian processes and artificial neural network are presented.

Recall than when GPs are used, the principal component analysis is always applied to the target spectra, hence the target vectors are composed of the PCA coefficients, not the bin intensities.

### 5.3.1. Application of Multitask Gaussian Processes

The vectorial nature of the target spectral data requires to implement a generalization of the classic Gaussian process introduced in Ch. 4. Indeed, if one avoided to do so, the only solution would be the programming of a number $N_Y$ of GPs, each one dedicated to a single PCA coefficient. This is clearly not the ideal situation, since spectral patterns arising in the spectra of interest could be correlated. Each coefficient indeed describes how much a particular principal component is present in the original spectrum. The interplay between this coefficients is crucial for the correct modeling of the spectral shape. For this reasons, Multitask Gaussian Processes (MGp) are preferred.

In such models, each PCA coefficient is treated as a different task, and an individual kernel is assigned to them. At the same time, the correlations between the different tasks is modeled by another part of the total kernel, as expressed in Eq. 4.26.

The two types of MGPs involved in this research are introduced below:

- **Linear Model of Coregionalization**:

  The Linear Model of Coregionalization (LCM), only briefly introduced in Sec. 4.3.1, is a very general formulation for the modeling of inter-task correlations. The main assumption is that each task-specific function $f_m(\mathbf{x})$ can be expressed as a linear combination of $Q$ independent latent Gaussian processes $u_q(\mathbf{x})$:

$$f_m(\mathbf{x}) = \sum_{q=1}^{Q} a_{mq}\, u_q(\mathbf{x}) + \epsilon_m, \tag{5.11}$$

  where $a_{mq}$ are task-dependent coefficients and $\epsilon_m \sim \mathcal{N}(0, \sigma_m^2)$ represents independent task-specific noise. Each latent process $u_q(\mathbf{x})$ follows a GP prior with zero mean and kernel $k_q(\mathbf{x}, \mathbf{x}')$. Mutual independence between the latent processes is assumed, therefore the cross-covariance between two tasks $m$ and $m'$ evaluated at inputs $\mathbf{x}$ and $\mathbf{x}'$ can be written:

$$\mathrm{cov}\big[f_m(\mathbf{x}), f_{m'}(\mathbf{x}')\big] = \sum_{q=1}^{Q} a_{mq}\, a_{m'q}\, k_q(\mathbf{x}, \mathbf{x}'). \tag{5.12}$$

The above formulation can be compactly written as:

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} \mathbf{B}_q \, k_q(\mathbf{x}, \mathbf{x}'), \tag{5.13}$$

where $\mathbf{B}_q = \mathbf{a}_q \mathbf{a}_q^\top$ is the so-called *coregionalization matrix*, encoding the correlations between the $M$ tasks for the $q^{th}$ latent process.

LCMs are thus able to model rich and flexible inter-task dependencies, and allow to express a different correlation structure for each latent component. Increasing the number of latent functions $Q$ enhances the expressivity of the model, at the expense of a larger number of parameters and a higher computational cost. The LCM can therefore capture complex relationships among tasks, such as partially shared information or hierarchical dependencies [72, 73].

- **Intrinsic Coregionalization Model**:

  The Intrinsic Coregionalization Model (ICM) is a constrained version of the LCM, obtained by assuming that all tasks share the same kernel function across inputs. In this case, the covariance between tasks $m$ and $m'$ is written as:

$$\mathrm{cov}\big[f_m(\mathbf{x}), f_{m'}(\mathbf{x}')\big] = \mathbf{B}_{mm'} \, k(\mathbf{x}, \mathbf{x}'), \tag{5.14}$$

  or equivalently in matrix form:

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = \mathbf{B} \otimes k(\mathbf{x}, \mathbf{x}'), \tag{5.15}$$

  where $\otimes$ denotes the Kronecker product, $\mathbf{B}$ is the positive semi-definite coregionalization matrix capturing correlations between tasks, and $k(\mathbf{x}, \mathbf{x}')$ is a shared covariance function over the input space.

  The ICM assumes that all tasks exhibit the same functional behavior with respect to the input variables, differing only in magnitude and mutual correlation. This restriction reduces the number of free parameters and typically yields a more stable and computationally efficient model. The ICM can be regarded as a special case of the LCM with $Q = 1$, i.e.

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = \mathbf{B}_1 \, k_1(\mathbf{x}, \mathbf{x}'), \tag{5.16}$$

  which explains its intrinsic simplicity. Despite its reduced flexibility, the ICM is

widely used due to its stability, interpretability, and low computational burden, especially when the tasks are strongly correlated and expected to share a common underlying physical behavior [67, 72].

In both cases, an additional *bias kernel* term, defined as $k_b(\mathbf{x}, \mathbf{x}') = \sigma_b^2$, can be added to capture global offsets or baseline differences among tasks:

$$\mathbf{K}_{\text{total}}(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} \mathbf{B}_q \, k_q(\mathbf{x}, \mathbf{x}') + \mathbf{B}_b \, k_b(\mathbf{x}, \mathbf{x}'), \tag{5.17}$$

where $\mathbf{B}_b$ may represent a diagonal or low-rank matrix. The bias term can improve the model's capacity to represent additive task-dependent effects and systematic shifts across outputs.

In summary, the main difference between LCM and ICM is that the first assumes independent kernels for every target, each one with its own hyperparameters, while the latter reduces the description of every task to a single kernel with shared hyperparameters, while preserving a modeling of the task correlations.

The training choices for these two models are reported in the following.

**Training of Intrinsic Coregionalization Model** The ICM is composed of $N_{Y_{PCA}} = 15$ tasks, sharing a common Matèrn kernel. A shared bias is modeled introducing a constant mean in the multitask model, instead of the usual zero mean. Another possibility is to introduce a constant kernel term in any of the task-specific kernels. Since no important difference between the two choices is observed, the first, simpler approach is the preferred one.
The rank of the coregionalization matrix is set to 2. This choice permits to contain overfitting and stabilize the convergence of the model. A high rank indeed allows to represent any possible correlation between tasks, introducing a high capacity in the model, often superfluous. The training is performed through the minimization of a negative marginal log likelihood based on a multitask Gaussian likelihood.
The optimization is managed by the Adam optimizer, with default parameters except for the learning rate. The learning rate is initialized to $10^{-1}$, but a scheduler is used with the *ReduceLROnPlateau* option. The effect is a reduction of the learning rate while approaching the plateau of the validation loss. This allows the model to stabilize the convergence of the minimization algorithm.
Validation based early stopping is employed to contain overfitting. The patience is set to 50, while training is stopped when variations smaller than $10^{-5}$ are observed.

Fig. 5.7 shows the training and validation loss trends during the learning phase.
The model is then put in evaluation mode, in order to estimate the loss on the various subsets. This represents the model's predictive likelihood without training-time stabilizations or jitter. The training loss reaches a minimum of -0.30 while the validation loss descends to 0.26. A final loss of 1.70 is obtained on the test set.

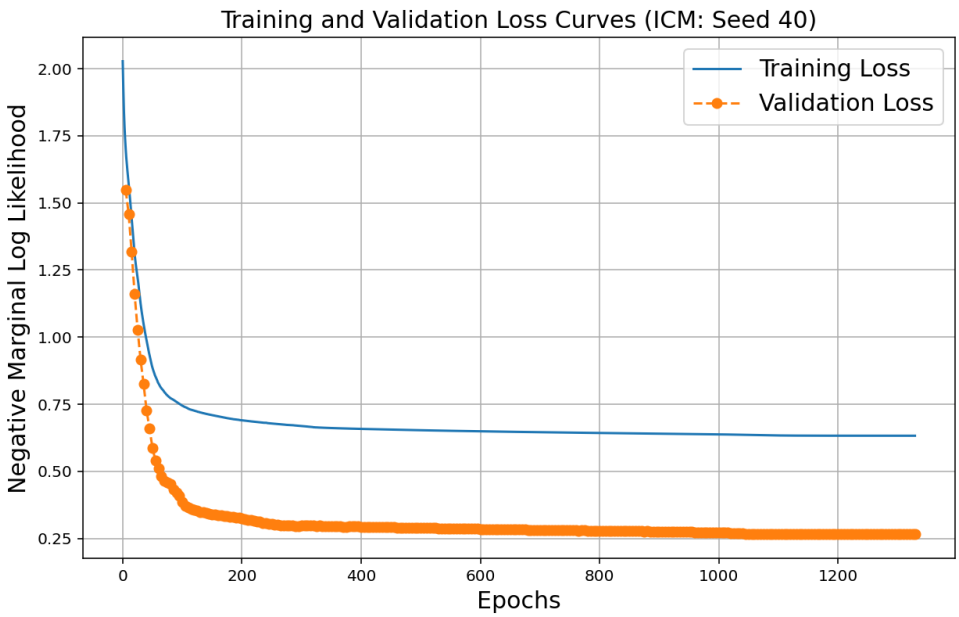



Figure 5.7: The training (in blue) and validation (in orange) losses of the ICM model are shown against the number of epochs. The training curve reaches a minimum which is higher than the final value of -0.2801. This often occurs in Gaussian process. The values obtained during learnig can indeed be influenced by the training mode.

**Training of Linear Coregionalization Model** The LCM is composed of $N_{Y_{PCA}} = 15$ tasks, each associated with a Matèrn kernel. A shared bias is again modeled introducing a constant mean in the multitask model. The rank of the coregionalization matrices is set to 2 for each kernel component, for the same reasons exposed for ICM.
The rest of the training choices is kept exactly the same as above. Fig. 5.8 shows the training and validation loss trends during the learning phase.
The training loss reaches a minimum of 0.09, while the validation loss descends to 0.50. A final loss of 1.17 is obtained on the test set.

The comparison between Fig. 5.7 and Fig. 5.8 shows that the ICM training phase is more stable. Moreover, the test, training and validation losses reach a lower value in this case, signaling that the model is less prone to overfitting during learning. On the other hand, the LCM model shows a lower value of the test loss, indicating a better

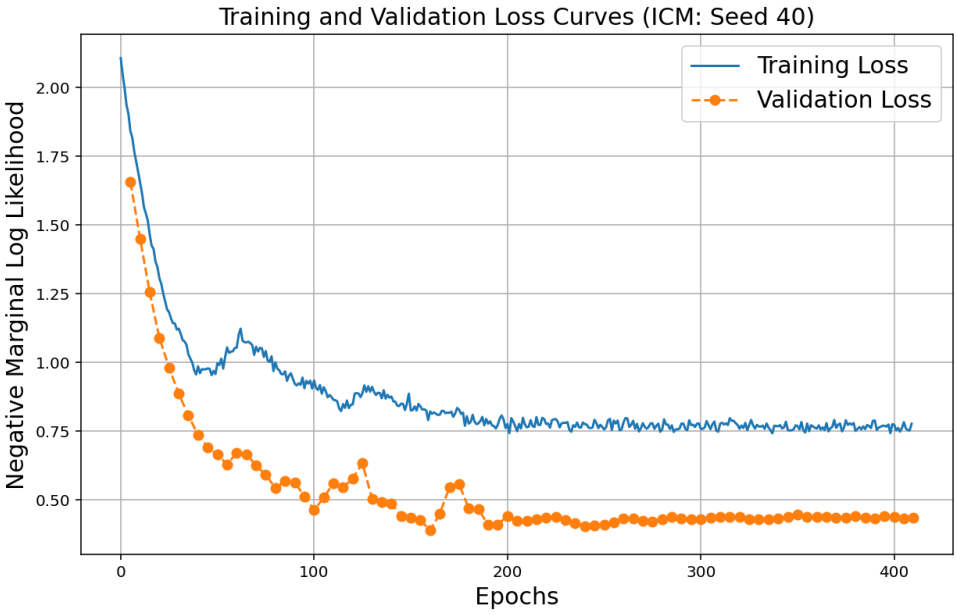



Figure 5.8: Training (in blue) and validation (in orange) losses of the LCM model against the number of epochs. As in ICM, the training curve minimum value in training mode can differ from the final evaluation loss.

generalization performance on unseen data. This suggests that the additional flexibility of multiple kernels in the LCM may allow it to capture subtle correlations between tasks that improve predictive accuracy outside the training set.

Remember that the loss values are calculated with respect to the PCA coefficients, not to the spectral shape in itself. The final judgement on the models performance is then to be evaluated looking at the comparison between the measured spectra and the final predictions of the models, obtained after the inversion of the PCA and of the other scaling procedures.

### 5.3.2. Application of Neural Networks

In the previous subsection, the need to introduce a multitask approach to tackle the problem of this thesis was highlighted when working with GPs. The application of artificial neural networks as regression models is instead directly prone to this possibility. Moreover, the much higher representational capability of NNs does not require the reduction of order obtained through the PCA, as already discussed in Sec. 5.2.5.

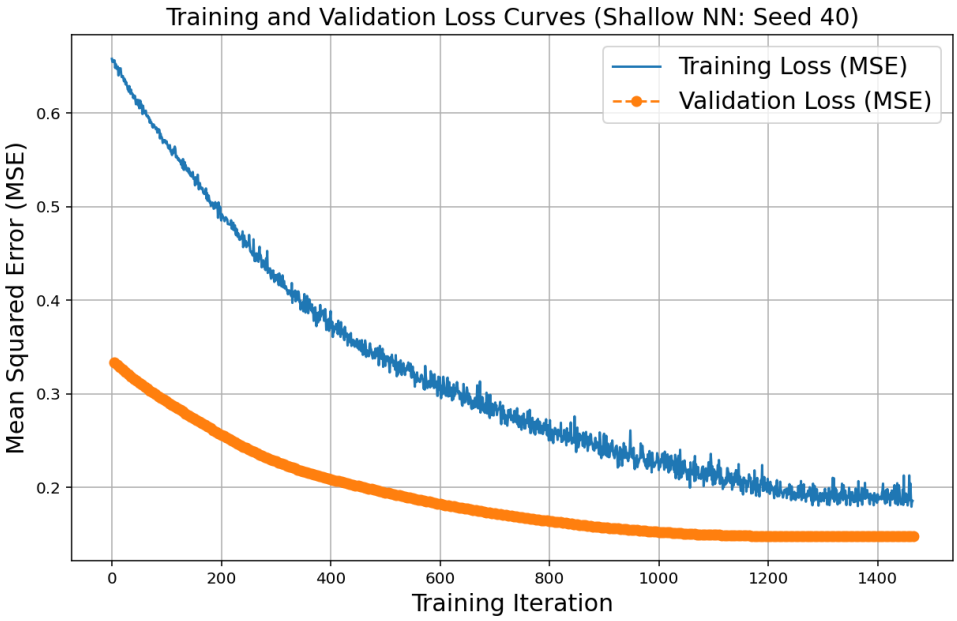
In this thesis, both a shallow and a deep neural network (see Sec. 4.3.2) have been trained, in order to evaluated the dependence of the final prediction on the depth of the models. In particular, the shallow neural network is composed of a single layer of 64 neurons,

connected to an output layer with dimension $N_Y = 60$. ReLu activation functions are employed, with a dropout of 0.1 for Monte Carlo Dropout estimation of the uncertainties (see Sec. A.3). The networks map the input features to multiple output tasks, corresponding to the energy bins of the spectra. The deep network is instead made adding another hidden layer, again made of 64 neurons. The choices during learning are shared between the two models.
The training is performed through the minimization of the mean squared error (MSE) loss between the predicted and target outputs.
Optimization is managed with the Adam optimizer, with an initial learning rate of $10^{-4}$, then adjusted through a *ReduceLROnPlateau* scheduler. The scheduler reduces the learning rate when the validation loss approaches a plateau, stabilizing the convergence. Again, validation based early stopping is employed with a patience of 50 evaluation intervals and a minimum improvement threshold of $10^{-4}$ to prevent overfitting.
Fig. 5.9 shows the training and validation loss trends over the epochs for both cases. After training, the network is set to evaluation mode to compute the loss on all datasets without the stochastic effects of dropout. The final MSE achieved by the shallow neural network (Fig. 5.9a) is 0.18 on the training set, 0.14 on the validation set, and 0.40 on the test set.
The final MSE achieved by the deep neural network (Fig. 5.9b) is 0.12 on the training set, 0.10 on the validation set, and 0.37 on the test set.

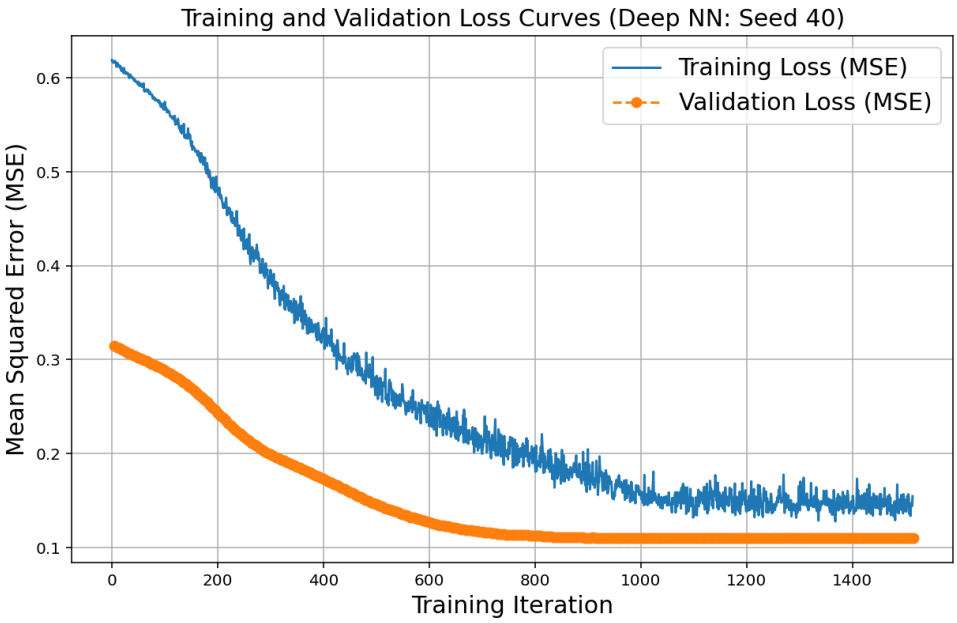
The comparison between the two models shows that the deep neural network achieves slightly lower errors on both the training and validation sets, indicating a higher capacity to fit complex patterns in the data. Moreover, the slightly lower test loss suggests an improved generalization capability compared to the shallow architecture, which appears natural due to the increase in depth.

Independently on the depth of the neural network, it may seem that lower losses are reached with respect to the Gaussian processes case. However, this comparison as no validity, since the models train on two different datasets: the MGPs work with PCA coefficients, while the NNs are trained directly with the shapes of the antineutrino spectra. A final comparison of the results of these two type of models is to be delayed to the results chapter (Ch. 6), where the actual comparison between predicted and measured shapes is reported.

As a concluding, and fundamental remark, every uncertainty on the predictions is evaluated with the Monte Carlo technique, using MC Dropout with neural networks, and usual MC sampling on the output distributions in the Gaussian processes case. Indeed,




(a) The training (in blue) and validation (in orange) losses of the shallow neural network are shown against the number of epochs.




(b) The training (in blue) and validation (in orange) losses of the deep neural network are shown against the number of epochs.

Figure 5.9: Comparison of shallow and deep neural network training phase. No major difference is observed in the convergence of the two models, even if the deep neural network is twice deeper than the shallow one.

as explained in Sec. A.3, even if MGPs are Bayesian model, the presence of the PCA pre-

processing requires an uncomfortable propagation of the uncertainties, which is avoided with the Monte Carlo technique.

# 6 | Analysis of Predictive Results

Here the results of the computations made exploiting the ML algorithms described in Ch. 5 are presented.

As explained in Sec. 5.3, four different algorithms have been used: two multitask GPs involving different kinds of coregionalization model, and two neural networks with different depth.

The results are shown for each of the presented models, with interest in identifying the one with the best performance. Such performance is evaluated using different criteria:

- Capability of the model to reproduce the spectral shape of single isotopes,
- Capability of the model to reproduce the integral of single isotope spectra,
- Capability of the model to reproduce the overall target dataset when the summation method is applied for the evaluation of the $^{235}U$ and/or $^{238}U$ antineutrino spectrum.

## 6.1. Model Performance Expectations

Before the analysis of the results themselves, a moment is taken to discuss the expected performance of the proposed ML models.

**Multitask Gaussian Processes** Multitask Gaussian processes have been employed since they are a natural generalization of Gaussian processes to multidimensional predictions. Their intrinsic Bayesian nature is of particular interest for this work, being well suited to treat problems in which a reduced number of examples is available. Moreover, the multitask characteristic allows to model the correlations between the dimensions of the observed target. This possibility would be otherwise impossible if one trained a different Gaussian process per each target dimension. Indeed this option has been investigated, with poor results.

The use of PCA with multitask Gaussian processes ensures that the predicted spectra, after inversed preprocessing, cannot be outside the space of shapes described by the

principal components extracted from the training set. Smoother energy shapes are also expected with respect to the case of neural networks.

The LCM is capable of modeling a much wider variety of patterns, indeed each principal component coefficient is modeled through a distinct kernel, each with dedicated lenghtscales for every input dimension. While the representational capabilities are stronger than the ICM case, the training of the LCM (see Fig. 5.8) is expected to be more chaotic. On the other hand, the ICM model, that uses the same model for every PC component, is expected to be characterized by a more stable training curve (see Fig. 5.7), since it treats each output dimension in the same way. From this point of view, the ICM model is expected to produce less uncertain prediction.

**Artificial Neural Networks** Artificial neural networks have also been trained in this work. Multidimensional predictions are intrinsically possible with these models. Moreover PCA is not required, thanks to their ability to treat high dimensional data. This characteristic allows to model with higher flexibility the spectral shapes of the dataset, and exploit the full non linear character of ANN.

However, conversely to Gaussian processes, neural networks work better if trained with important amounts of data, which in this work are not available. For this reason, in this specific work, the performance of the two trained networks (shallow and deep) is expected to be poorer than ICM and LCM. At the same time the computational time required, is much smaller.

The DNN is expected to be more precise than the SNN, indeed its training curve reaches a lower minimum (see Fig. 5.9a and Fig. 5.9b).

## 6.2. Spectral Shapes and Integral Predictions of Single Isotopes

In this section, the first two points will be investigated. The attention is mainly on test set, since the aim is to judge the generalization capabilities of the models. For the sake of completeness, however, and also to show in a more extensive way the spectral reproduction capacities of the models, the reconstruction of the spectral shapes of validation isotopes will be shown too.

To measure the reproduction capability on the spectral shape, the mean squared error (MSE) and the $\chi^2_{red}$ are computed for each prediction and target couple up to the Q-value

of the specific isotope [1].

To measure the integral prediction performance the ratio to the target integral is used. It will be referred to as PTIR (Prediction to Target Integral Ratio).

In the following, various figures (Fig. 6.1– 6.10) are used to show some examples of the performance of the models on single isotopes. The selected examples are chosen to put in evidence various characteristics of the models: some identify the good reconstruction capability of the various models (Fig. 6.7), some put in evidence the ability to distinguish the TAGS and HPGe measurements as in Fig. 6.3a and Fig. 6.4a, and some others show that the models not always are coherent in their predictions, as for Fig. 6.10.

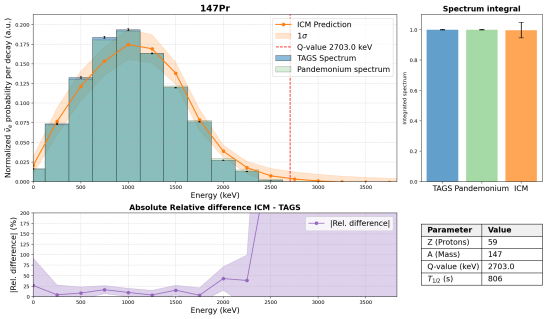
The idea is to compare the result for an isotope between each model. Therefore, each figure will contain four subfigures, one per each model.
The test set is represented in Fig. 6.1– 6.5, the validation set follows in Fig. 6.6– 6.10. In each subfigure, the upper left part illustrates the comparison between the spectral prediction and the target spectrum, always labeled as TAGS for simplicity. The spectral shape affected by Pandemonium is also shown: clearly, for the SAFE isotopes, Pandemonium and TAGS spectrum coincide. There, error bars are shown only for the target spectrum. In the upper right part the spectrum integral is shown. In the lower left part the absolute relative difference between prediction and target is shown. Finally, in the lower right part some properties of the isotope of interest are reported. In green, the spectrum used to compute the input variables for the model (HPGe). In blue, the target spectrum, hence TAGS for TAGS isotopes and HPGe for SAFE isotopes. In orange, the prediction for the test set. In red, the prediction for the validation set.

The performance evaluation plots for all the isotopes involved in the validation and test sets can be found at [74].

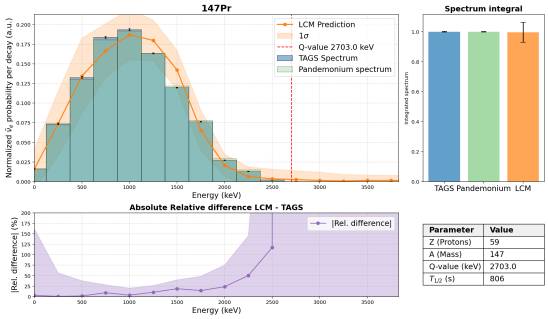
One can already see in these examples that the neural network approaches seem to have a worse performance. Fig. 6.2 show that both the SNN and the DNN show a difficulty in reproducing the whole variety of spectral shapes.

Importantly, Fig. 6.3a and Fig. 6.4a show how the ICM seems to be able to understand the important deviation that the target spectrum has with respect to the Pandemonium affected one. The case of Fig. 6.3a is very interesting. The predicted spectrum is much closer to the TAGS one than the HPGe, highlighting an important capability of the model in this sense. This is exactly what the model should be able to do. Such property seems to be present only for the ICM.

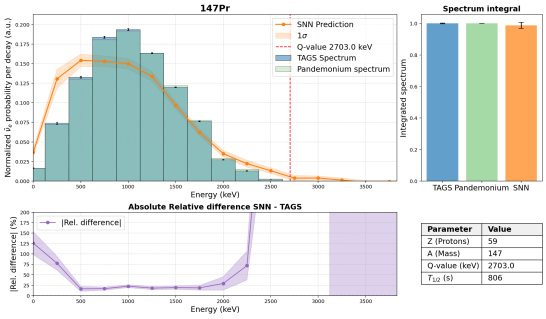
[1] Recall the padding procedure on the target spectra, described in Sec. 5.2.5



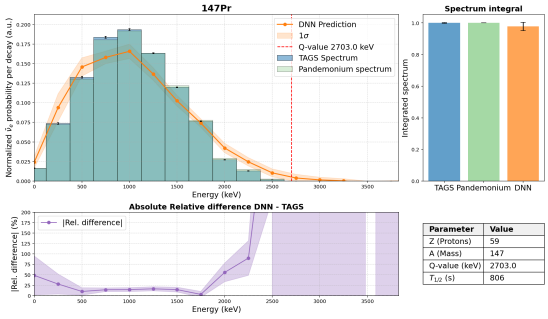

(a) The figure shows the performance of the ICM on $^{147}$Pr.




(b) The figure shows the performance of the LCM on $^{147}$Pr.

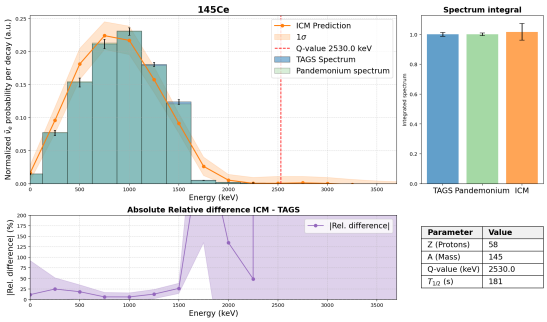



(c) The figure shows the performance of the SNN on $^{147}$Pr.

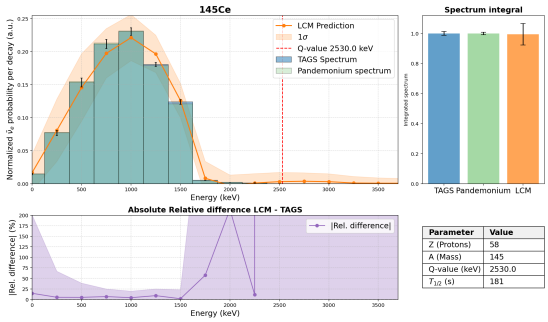



(d) The figure shows the performance of the DNN on $^{147}$Pr.

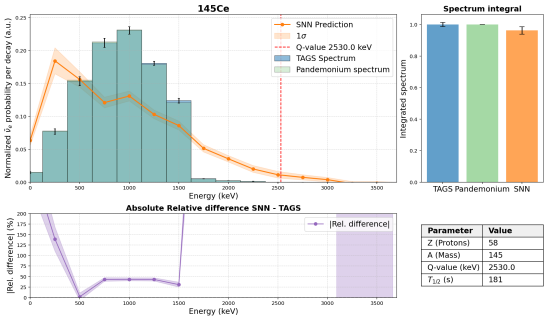
Figure 6.1: The figure shows the performance of the models considered in this work for $^{147}$Pr. Fig. 6.1a refers to the ICM, Fig. 6.1b refers to the LCM, Fig. 6.1c refers to the SNN, and Fig. 6.1d refers to the DNN.



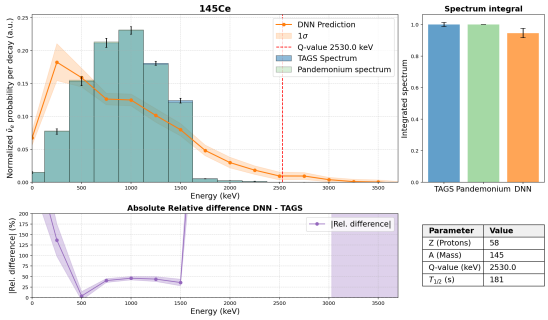

(a) The figure shows the performance of the ICM on $^{145}$Ce.




(b) The figure shows the performance of the LCM on $^{145}$Ce.

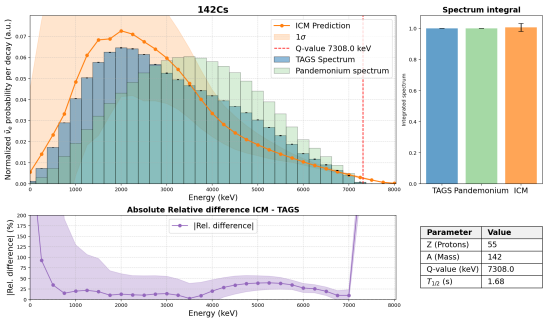



(c) The figure shows the performance of the SNN on $^{145}$Ce.

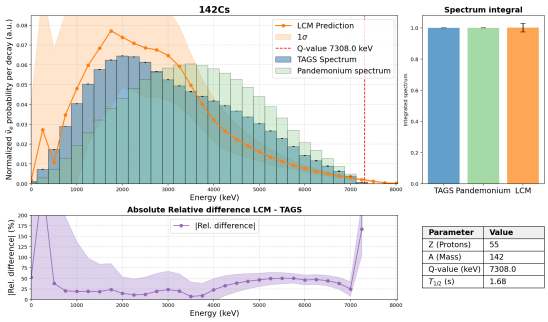



(d) The figure shows the performance of the DNN on $^{145}$Ce.

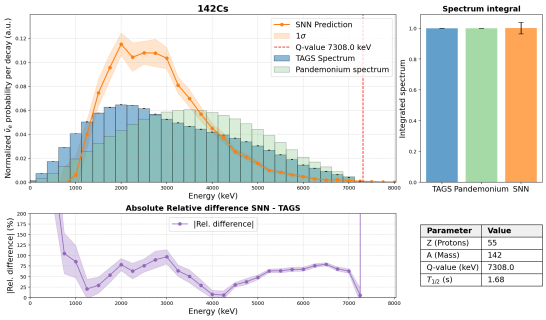
Figure 6.2: The figure shows the performance of the models considered in this work for $^{145}$Ce. Fig. 6.2a refers to the ICM, Fig. 6.2b refers to the LCM, Fig. 6.2c refers to the SNN, and Fig. 6.2d refers to the DNN.



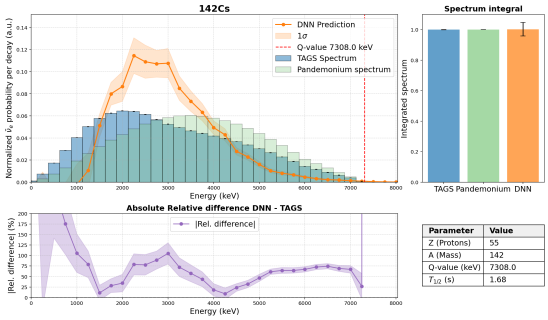

(a) The figure shows the performance of the ICM on $^{142}$Cs.




(b) The figure shows the performance of the LCM on $^{142}$Cs.

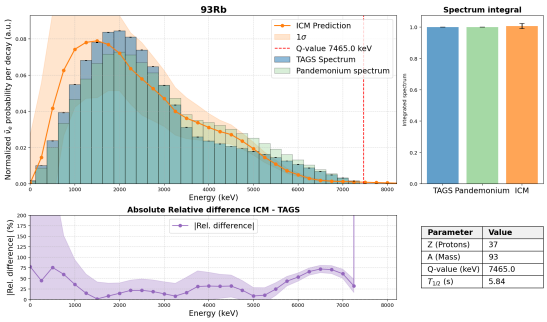



(c) The figure shows the performance of the SNN on $^{142}$Cs.

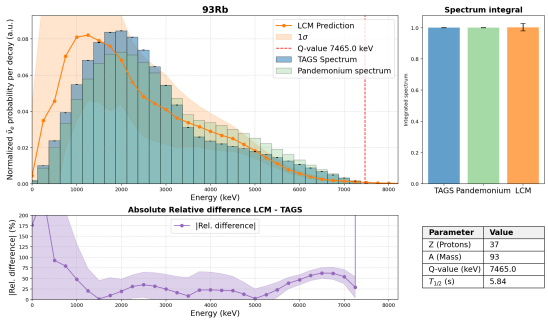



(d) The figure shows the performance of the DNN on $^{142}$Cs.

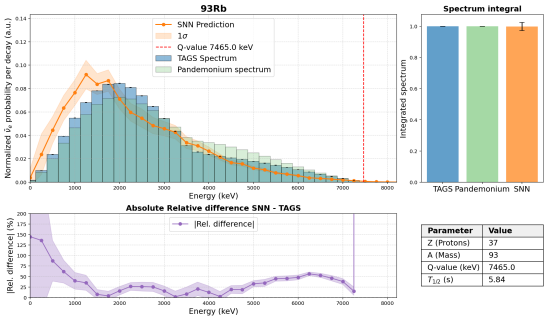
Figure 6.3: The figure shows the performance of the models considered in this work for $^{142}$Cs. Fig. 6.3a refers to the ICM, Fig. 6.3b refers to the LCM, Fig. 6.3c refers to the SNN, and Fig. 6.3d refers to the DNN.



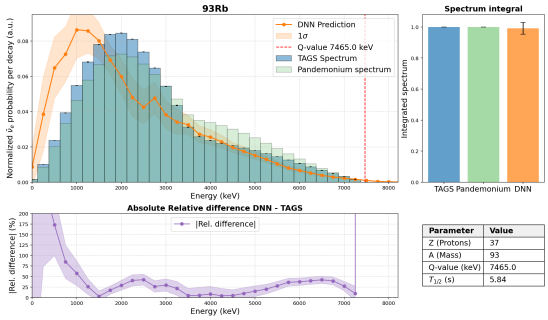

(a) The figure shows the performance of the ICM on $^{93}$Rb.




(b) The figure shows the performance of the LCM on $^{93}$Rb.

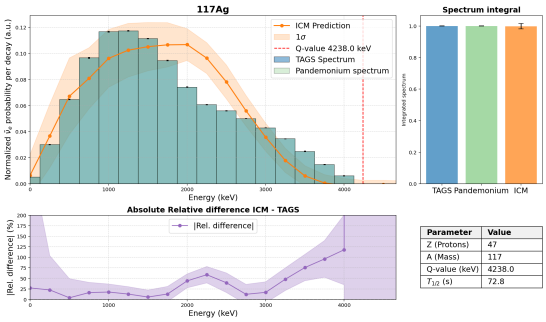



(c) The figure shows the performance of the SNN on $^{93}$Rb.

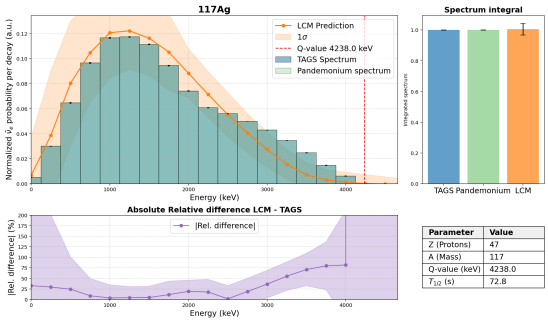



(d) The figure shows the performance of the DNN on $^{93}$Rb.

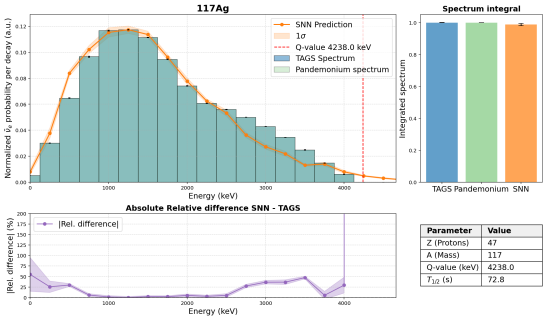
Figure 6.4: The figure shows the performance of the models considered in this work for $^{93}$Rb. Fig. 6.4a refers to the ICM, Fig. 6.4b refers to the LCM, Fig. 6.4c refers to the SNN, and Fig. 6.4d refers to the DNN.



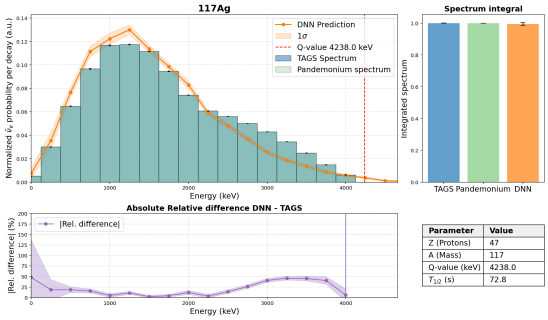

(a) The figure shows the performance of the ICM on $^{117}$Ag.




(b) The figure shows the performance of the LCM on $^{117}$Ag.

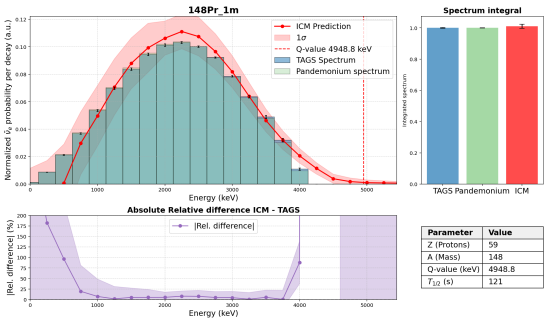



(c) The figure shows the performance of the SNN on $^{117}$Ag.

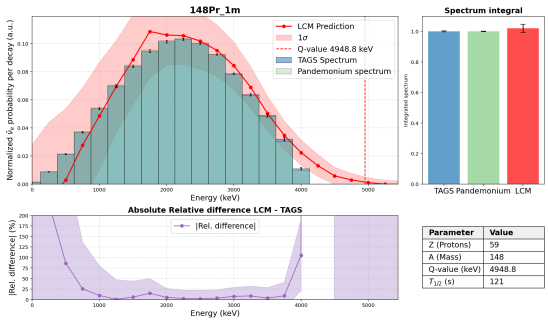



(d) The figure shows the performance of the DNN on $^{117}$Ag.

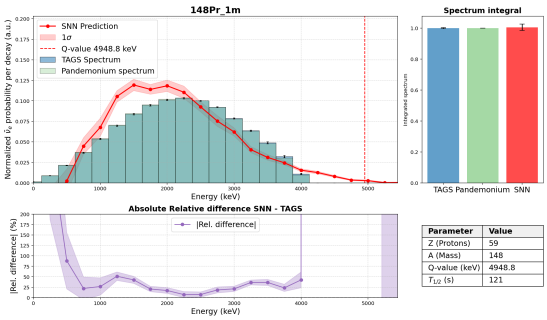
Figure 6.5: The figure shows the performance of the models considered in this work for $^{117}$Ag. Fig. 6.5a refers to the ICM, Fig. 6.5b refers to the LCM, Fig. 6.5c refers to the SNN, and Fig. 6.5d refers to the DNN.



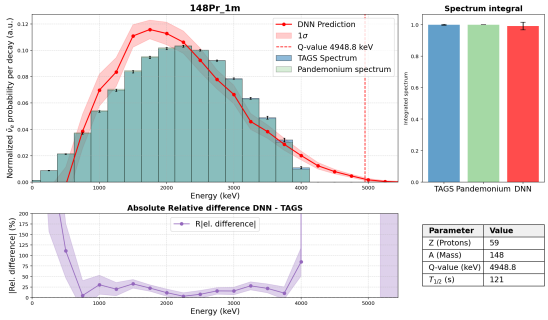

(a) The figure shows the performance of the ICM on $^{148m}$Pr.




(b) The figure shows the performance of the LCM on $^{148m}$Pr.

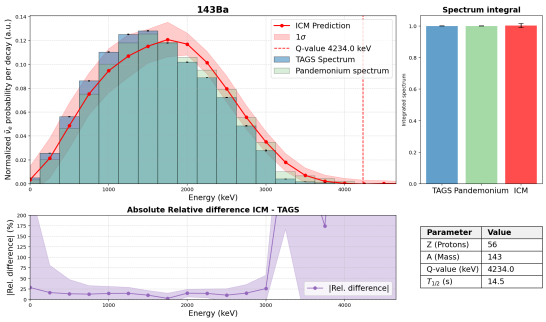



(c) The figure shows the performance of the SNN on $^{148m}$Pr.

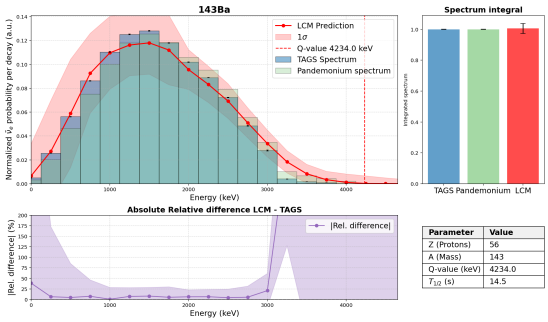



(d) The figure shows the performance of the DNN on $^{148m}$Pr.

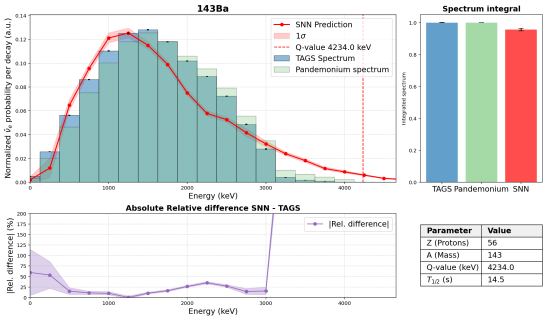
Figure 6.6: The figure shows the performance of the models considered in this work for $^{148m}$Pr. Fig. 6.6a refers to the ICM, Fig. 6.6b refers to the LCM, Fig. 6.6c refers to the SNN, and Fig. 6.6d refers to the DNN.



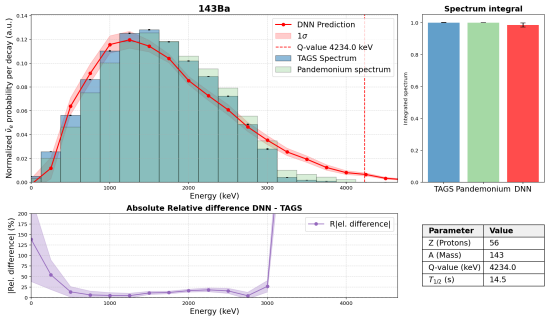

(a) The figure shows the performance of the ICM on $^{143}$Ba.




(b) The figure shows the performance of the LCM on $^{143}$Ba.

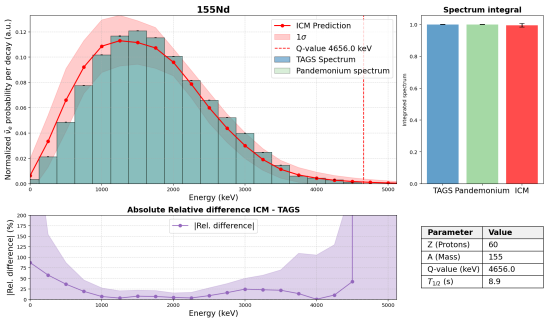



(c) The figure shows the performance of the SNN on $^{143}$Ba.

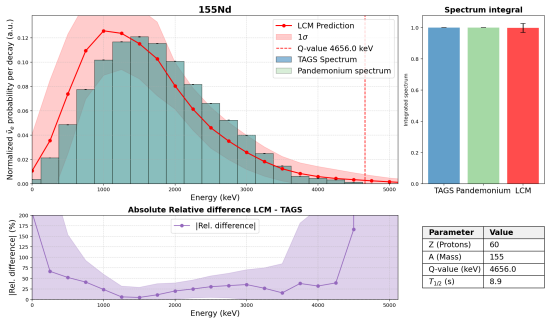



(d) The figure shows the performance of the DNN on $^{143}$Ba.

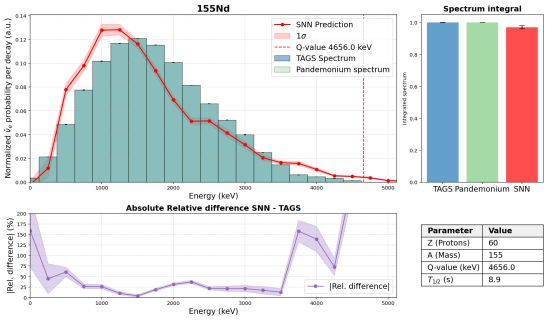
Figure 6.7: The figure shows the performance of the models considered in this work for $^{143}$Ba. Fig. 6.7a refers to the ICM, Fig. 6.7b refers to the LCM, Fig. 6.7c refers to the SNN, and Fig. 6.7d refers to the DNN.



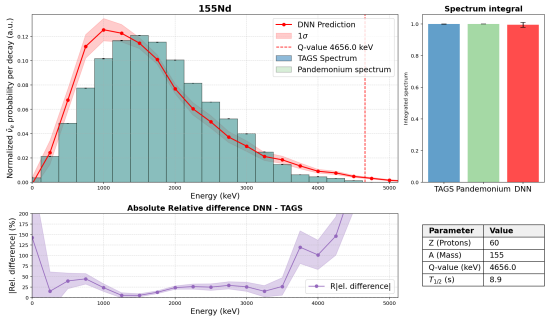

(a) The figure shows the performance of the ICM on $^{155}$Nd.




(b) The figure shows the performance of the LCM on $^{155}$Nd.

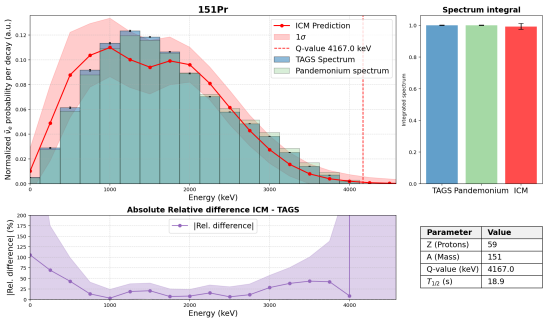



(c) The figure shows the performance of the SNN on $^{155}$Nd.

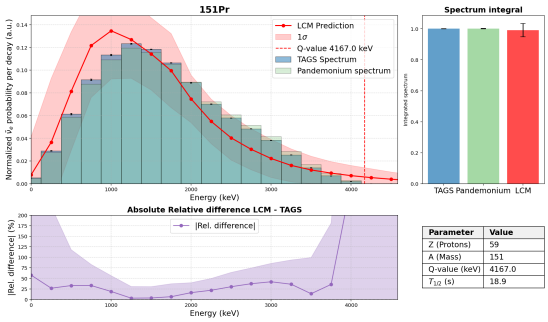



(d) The figure shows the performance of the DNN on $^{155}$Nd.

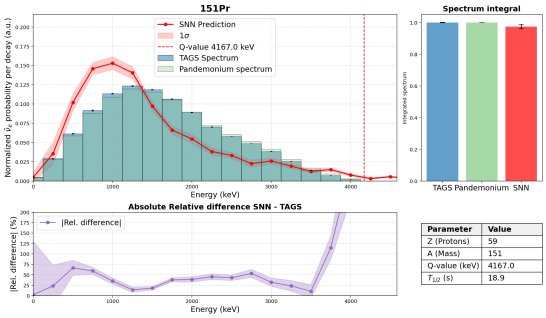
Figure 6.8: The figure shows the performance of the models considered in this work for $^{155}$Nd. Fig. 6.8a refers to the ICM, Fig. 6.8b refers to the LCM, Fig. 6.8c refers to the SNN, and Fig. 6.8d refers to the DNN.



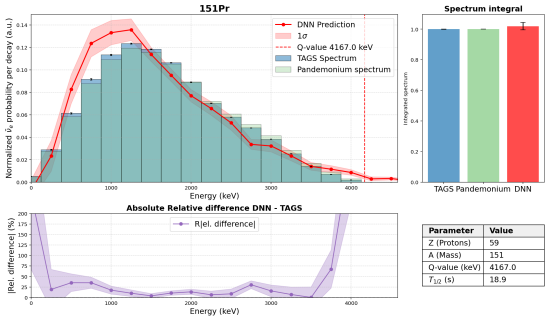

(a) The figure shows the performance of the ICM on $^{151}$Pr.




(b) The figure shows the performance of the LCM on $^{151}$Pr.

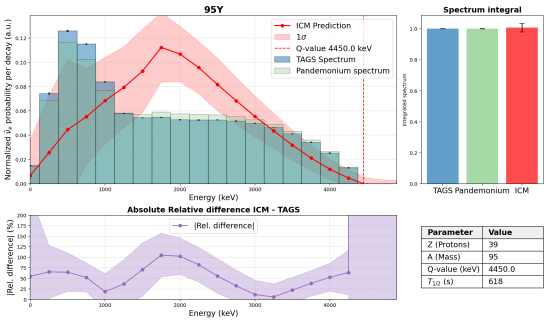



(c) The figure shows the performance of the SNN on $^{151}$Pr.

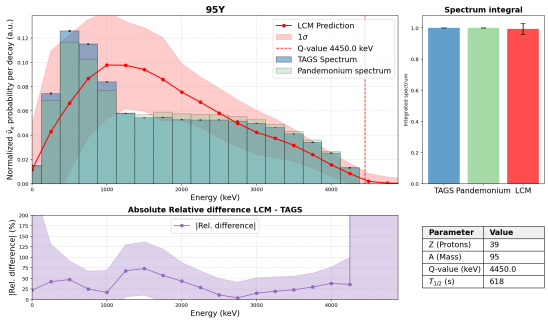



(d) The figure shows the performance of the DNN on $^{151}$Pr.

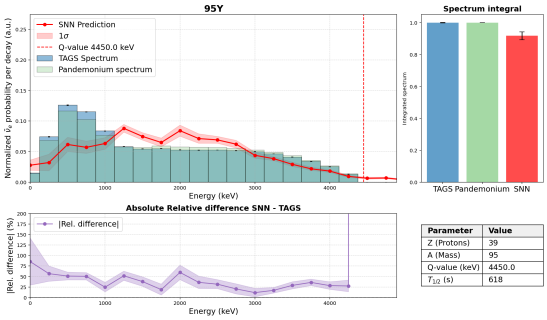
Figure 6.9: The figure shows the performance of the models considered in this work for $^{151}$Pr. Fig. 6.9a refers to the ICM, Fig. 6.9b refers to the LCM, Fig. 6.9c refers to the SNN, and Fig. 6.9d refers to the DNN.



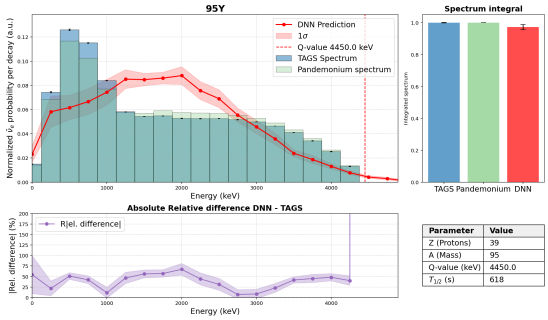

(a) The figure shows the performance of the ICM on $^{95}$Y.




(b) The figure shows the performance of the LCM on $^{95}$Y.




(c) The figure shows the performance of the SNN on $^{95}$Y.




(d) The figure shows the performance of the DNN on $^{95}$Y.

Figure 6.10: The figure shows the performance of the models considered in this work for $^{95}$Y. Fig. 6.10a refers to the ICM, Fig. 6.10b refers to the LCM, Fig. 6.10c refers to the SNN, and Fig. 6.10d refers to the DNN.

A systematic analysis is required to objectively conclude which model works best. To this purpose, a table (Table 6.1) reporting the used metrics (MSE, $\chi^2_{red}$, PTR) for this evaluation is proposed. The analysis involves only the isotopes in the test set.

Table 6.1: Comparison of test metrics for each isotope and model. The ICM, LCM, SNN and DNN models are respectively represented in red, blue, green and orange. Values for each model are reported vertically.

| Isotope | MSE | $\chi^2_{red}$ | PRIT |
|---|---|---|---|
| $^{147}$Pr | 1.6e-04 | 2.0e-01 | 1.0e+00 |
| | 1.2e-04 | 1.0e-01 | 1.0e+00 |
| | 8.0e-04 | 1.1e+00 | 9.9e-01 |
| | 3.2e-04 | 5.4e-01 | 9.8e-01 |
| $^{145}$Ce | 3.4e-04 | 2.3e-01 | 1.0e+00 |
| | 6.1e-05 | 2.3e-02 | 9.9e-01 |
| | 3.9e-03 | 4.0e+00 | 9.6e-01 |
| | 4.0e-03 | 2.8e+00 | 9.5e-01 |
| $^{154}$Pm | 5.0e-05 | 3.5e-02 | 1.0e+00 |
| | 4.1e-05 | 2.3e-02 | 1.0e+00 |
| | 3.2e-04 | 3.8e-01 | 9.9e-01 |
| | 2.5e-04 | 3.1e-01 | 9.9e-01 |
| $^{142}$Cs | 5.4e-05 | 3.3e-01 | 1.0e+00 |
| | 9.7e-05 | 5.5e-01 | 1.0e+00 |
| | 9.5e-04 | 5.0e+00 | 1.0e+00 |
| | 7.9e-04 | 3.2e+00 | 1.0e+00 |
| $^{147}$Ce | 1.1e-04 | 1.7e-01 | 1.0e+00 |
| | 8.1e-05 | 8.6e-02 | 1.0e+00 |
| | 4.5e-04 | 1.2e+00 | 9.6e-01 |
| | 4.3e-04 | 1.0e+00 | 9.8e-01 |
| $^{142}$Ba | 2.5e-03 | 2.1e+00 | 9.9e-01 |
| | 3.3e-03 | 1.1e+00 | 9.9e-01 |
| | 7.7e-03 | 1.8e+01 | 9.9e-01 |
| | 7.0e-03 | 1.1e+01 | 9.9e-01 |

*Continued on next page*

Table 6.1 – *Continued from previous page*

| Isotope | MSE | $\chi^2_{red}$ | PRIT |
|---|---|---|---|
| $^{93}$Rb | 8.3e-05 | 3.1e-01 | 1.0e+00 |
| | 1.8e-04 | 2.7e-01 | 1.0e+00 |
| | 1.3e-04 | 9.7e-01 | 1.0e+00 |
| | 2.9e-04 | 1.3e+00 | 9.9e-01 |
| $^{146}$Pr | 3.7e-04 | 3.1e-01 | 1.0e+00 |
| | 2.4e-04 | 1.6e-01 | 1.0e+00 |
| | 7.4e-05 | 1.8e+01 | 9.7e-01 |
| | 5.9e-05 | 3.2e+00 | 9.8e-01 |
| $^{118m}$Ag | 1.4e-04 | 4.8e-01 | 1.0e+00 |
| | 2.2e-05 | 5.7e-02 | 1.0e+00 |
| | 3.7e-04 | 6.1e+00 | 1.0e+00 |
| | 3.8e-04 | 1.8e+00 | 9.9e-01 |
| $^{132}$I | 1.6e-04 | 3.1e-01 | 9.9e-01 |
| | 2.1e-04 | 1.2e-01 | 9.9e-01 |
| | 9.4e-04 | 3.3e+00 | 1.0e+00 |
| | 1.1e-03 | 1.8e+00 | 9.6e-01 |
| $^{124m}$In | 3.8e-05 | 1.2e-01 | 1.0e+00 |
| | 4.6e-05 | 7.7e-02 | 1.0e+00 |
| | 6.7e-05 | 3.0e+00 | 1.0e+00 |
| | 7.9e-05 | 4.7e-01 | 9.8e-01 |
| $^{133m}$Te | 1.4e-03 | 2.3e-01 | 9.7e-01 |
| | 1.3e-03 | 2.8e-01 | 9.7e-01 |
| | 4.1e-03 | 3.9e+00 | 1.0e+00 |
| | 3.8e-03 | 2.8e+00 | 9.9e-01 |
| $^{122m}$In | 2.6e-05 | 5.8e-02 | 1.0e+00 |
| | 3.7e-05 | 3.6e-02 | 1.0e+00 |
| | 2.6e-04 | 3.0e+00 | 1.0e+00 |
| | 2.1e-04 | 1.2e+00 | 1.0e+00 |
| $^{117}$Ag | 2.8e-04 | 7.8e-01 | 1.0e+00 |
| | 1.2e-04 | 2.8e-01 | 1.0e+00 |
| | 7.2e-05 | 4.8e-01 | 9.9e-01 |
| | 9.6e-05 | 5.9e-01 | 9.9e-01 |

*Continued on next page*

Table 6.1 – *Continued from previous page*

| Isotope | MSE | $\chi^2_{red}$ | PRIT |
|---|---|---|---|
| $^{121m}$Cd | 2.2e-04 | 1.5e-01 | 1.0e+00 |
| | 5.7e-05 | 3.1e-02 | 1.0e+00 |
| | 5.4e-04 | 5.8e-01 | 9.7e-01 |
| | 4.3e-04 | 8.1e-01 | 1.0e+00 |
| $^{135}$I | 8.1e-04 | 4.1e-01 | 9.9e-01 |
| | 1.2e-03 | 3.5e-01 | 9.7e-01 |
| | 1.8e-03 | 7.5e+00 | 1.0e+00 |
| | 1.7e-03 | 2.1e+00 | 9.6e-01 |

A first clear impression is that the value of the PRIT is between 0.95 and 1.10 for every isotope and for every model, which signals an important ability in estimating the integral of the spectra by any of the employed algorithms. On the other hand some differences in performance on the spectral shapes can be found.

First, concentrating on the $\chi^2_{red}$, it is possible to observe a general tendency. The ICM and LCM (Gaussian Processes) systematically show a lower value of this metric with respect to the SNN and DNN. This is associated to the fact that the uncertainty predicted with MC sampling is larger than the one obtained with neural networks, hence naturally leads to lower values of this metrics. Computing the global $\chi^2_{red}$ on the test set one obtains the values reported in Table 6.2.

| Model | Global $\chi^2_{red}$ |
|---|---|
| ICM | 3.27e-1 |
| LCM | 1.98e-1 |
| SNN | 1.85e+0 |
| DNN | 4.32e+0 |

Table 6.2: The table shows the Global $\chi^2_{red}$ on the test set value for each model. In purple, the value closer to 1 value is obtained by ICM.

Moreover, in table 6.3 are reported the mean values of the MSE on the test set obtained by the various models.

With this values at hand, it is clear that the best performance is obtained by the ICM. Both the MSE and $\chi^2_{red}$ values across the test set testify this. Moreover, as mentioned

| Model | Mean MSE |
|---|---|
| ICM | 4.21e-4 |
| LCM | 4.46e-4 |
| SNN | 1.41e-3 |
| DNN | 1.30e-3 |

Table 6.3: The table shows the mean MSE value on the test set for each model. In brown, the lower value is obtained by ICM.

above, the ICM algorithm seems to be the only one able to guarantee a real ability in identifying those cases in which a big discrepancy between target and Pandemonium affected spectral shape, and approximately correct it.

## 6.3. $^{235}$U and $^{238}$U Summed Spectrum Evaluation

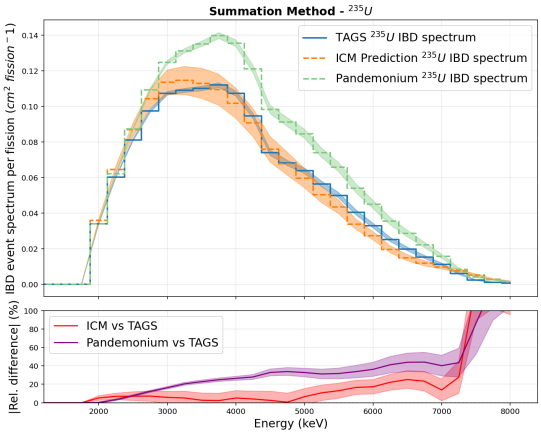
To further solidify the previous claims, in Fig. 6.11 and Fig. 6.12 are shown the estimates of the $^{235}U$ and $^{238}U$ antineutrino spectrum obtained employing the summation method. This evaluation is performed including all the isotopes of the dataset.

This analysis aim is to verify wether the performance of the model at least on the overall amount of data succeeds in its aim. Indeed there is the possibility that, even if the single spectral shapes are not well reconstructed, the overall sum of every isotope contribute may go in the direction of reducing the discrepancies of Pandemonium effect.

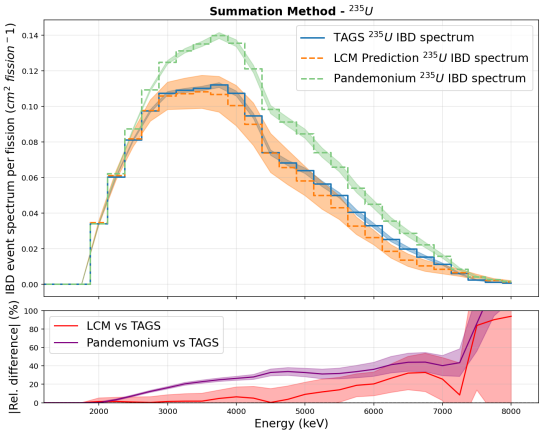
Again, the Intrinsic Coregionalization Model is the one that, for both the actinides, enables to maximally reduce the distance between the target and predicted spectra. The ICM can therefore be regarded as the best performing model.

Moreover, these last figures show an important and interesting result: apart for the case of the Linear Model of Coregionalization, where uncertainites dominate, in all the other cases, even if the spectral shape reconstruction is not always powerful the IBD event spectrum predicted is closer to the target (TAGS) than the Pandemonium affected data. This is exactly the purpose of this work.

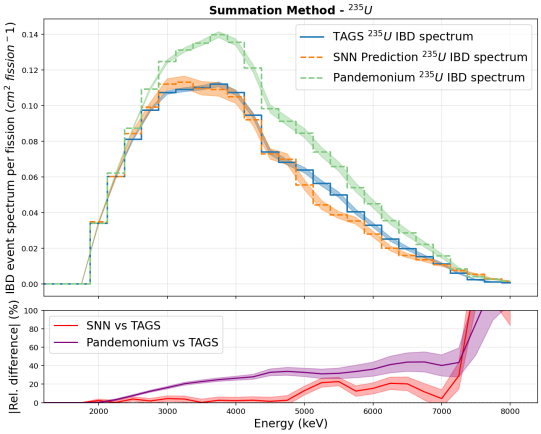
An effective mitigation of Pandemonium effect is obtained when looking at the overall dataset performance. This is very important to point out. As previously said, it is not necessarily true that even the Intrinsic Coregionalization Model is able to reconstruct correctly all the spectral shapes of the dataset. However, the result obtained in this work shows that ML can indeed be an interesting tool to treat nuclear physics problem



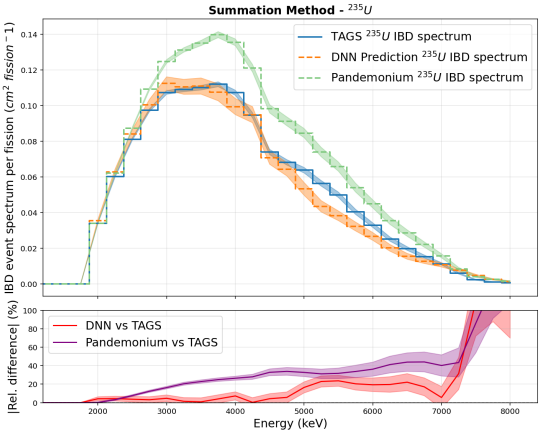

(a) The figure shows the performance of the ICM on $^{235}$U.




(b) The figure shows the performance of the LCM on $^{235}$U.



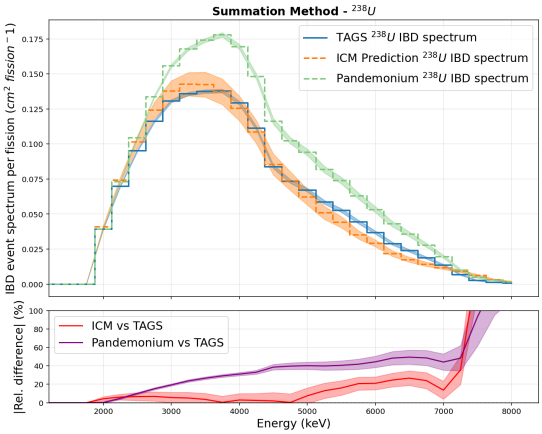

(c) The figure shows the performance of the SNN on $^{235}$U.



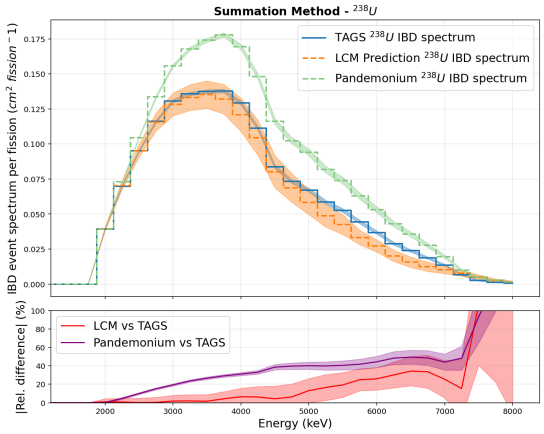

(d) The figure shows the performance of the DNN on $^{235}$U.

Figure 6.11: The figure shows a comparison between the performance of the various models when the antineutrino spectrum of $^{235}$U is obtained as the sum of the single isotope spectra predicted by the models. The antineutrino spectrum is multiplied by the IBD cross section to better put in evidence the energy interval of interest. Fig. 6.11a refers to the ICM, Fig. 6.11b refers to the LCM, Fig. 6.11c refers to the SNN, and Fig. 6.11d refers to the DNN.

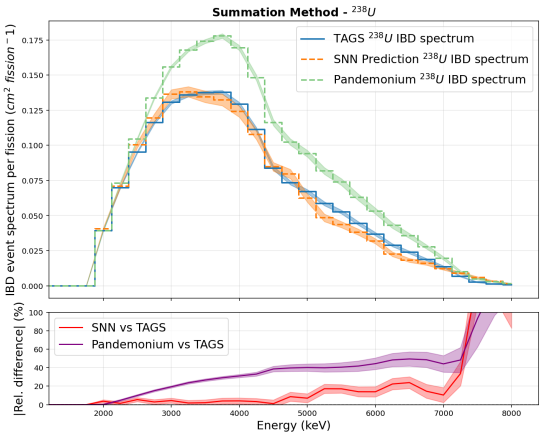
from the data point of view. As intrinsic statistical models, their performance is to be evaluated through the overall dataset. The nature of the summation method for the reactor antineutrino spectrum enables to do so even if the target object are the single isotope spectra. Therefore this thesis concludes that, even if many improvements can still be done, some validity is to be found in the approach proposed by this work. Moreover,



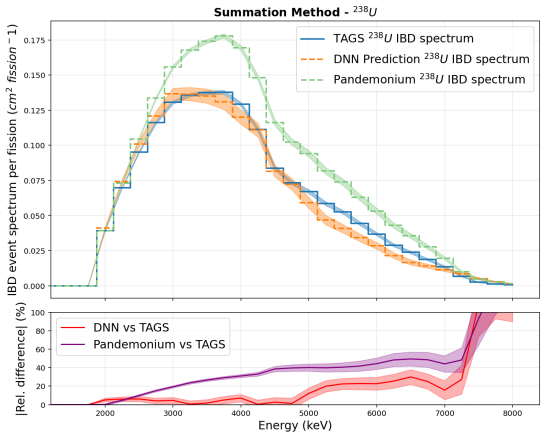

(a) The figure shows the performance of the ICM on $^{238}$U.




(b) The figure shows the performance of the LCM on $^{238}$U.




(c) The figure shows the performance of the SNN on $^{238}$U.




(d) The figure shows the performance of the DNN on $^{238}$U.

Figure 6.12: The figure shows a comparison between the performance of the various models when the antineutrino spectrum of $^{238}$U is obtained as the sum of the single isotope spectra predicted by the models. The antineutrino spectrum is multiplied by the IBD cross section to better put in evidence the energy interval of interest. Fig. 6.12a refers to the ICM, Fig. 6.12b refers to the LCM, Fig. 6.12c refers to the SNN, and Fig. 6.12d refers to the DNN.

the reduction in the absolute relative difference showed above, signals answers positively to the original question of this work, i.e. whether it was possible to model eventual common bias and patterns in Pandemonium effect phenomenology.

# Conclusions

The main topic addressed in this thesis is the accurate prediction of the reactor antineutrino spectrum. It is a fundamental requirement for high precision measurements in neutrino physics, such as the JUNO experiment. In particular, this thesis focused in developing a methodology to mitigate the Pandemonium effect, a well known experimental bias in the summation method, arising from incomplete nuclear decay data obtained with traditional HPGe detectors. The predicted spectral shape is distorted by this effect, resulting in an underestimation of the low energy part of the single isotope antineutrino spectra. The sensitivity of high precision neutrino oscillation experiments is then compromised.

To tackle this problem, an original data-driven approach based on machine learning was developed in this work. The conceptual strategy was to implement supervised regression models in order to implement a map from the biased HPGe spectral measurements to the high fidelity spectra obtained more recently with the TAGS technique. A dataset comprising 104 isotope - half with known Pandemonium correction (TAGS isotopes) and half assumed to be unaffected (SAFE isotopes) - was meticulously prepared. The model inputs consider fundamental nuclear properties as well as spectral information derived from the HPGe measurements. A particular role is played by a compressed representation of the beta transition properties derived with an autoencoder network, which successfully encoded the complex underlying physics of the decay schemes.

A summarizing block scheme representing the workflow defined in this work is shown in Fig. 6.13.

Four different models were implemented and evaluated: an Intrinsic Coregionalization Model (ICM), a Linear Model of Coregionalization (LCM), a Shallow Neural Network (SNN), and a Deep Neural Network (DNN). Their performance was analyzed on the ability to predict the spectral shape and integral of single isotopes in a test set, as well as on the resulting summed antineutrino spectrum of $^{235}$U and $^{238}$U actinides, constructed including all the isotopes in the dataset.

The multitask Gaussian processes (ICM and LCM) were expected to give more accu-

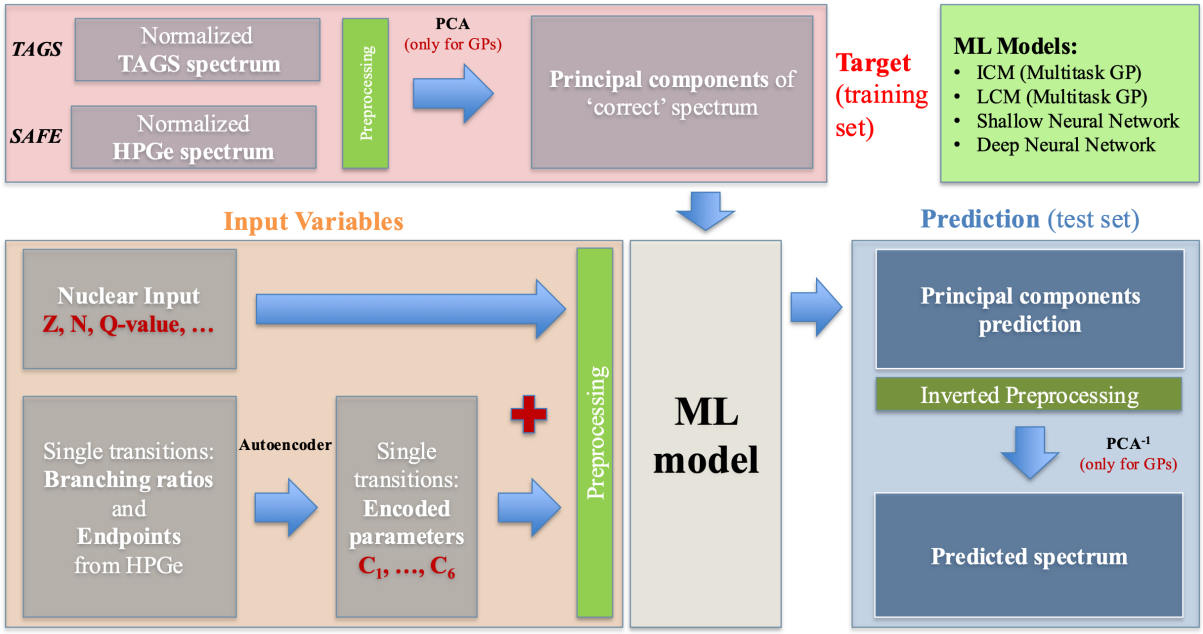



Figure 6.13: Block scheme representing the workflow defined in this work, and applied to the Pandemonium effect problem.

rate punctual prediction than the neural networks, this is due to the intrinsic ability of Bayesian models to treat low dimensional datasets as the one employed in this work. Higher computational cost and uncertainties in the prediction were foreseen at the same time. Moreover, the LCM, which treats each target dimension with distinct kernels, was expected to produce more uncertain predictions, due to an higher difficulty in training and a much wider representational capability than the ICM.
PCA was used in the case of multitask Gaussian processes, and was expected to produce smoother shapes, also more coherent with the energy patterns extracted from the dataset. On the other hand, neural network were expected to have higher flexibility, lower computational costs due to the reduced dimensions of the dataset. Their non linear predictive power was probably not fully exploited.

The key findings of this work can be summarized as follows:

1. **Superior Performance of the ICM Model:** a systematic comparison of the performance metrics, including Mean Squared Error (MSE) and the reduced chi-squared ($\chi^2_{red}$), identified the Intrinsic Coregionalization Model as the most effective. The ICM, a multitask Gaussian Process model, consistently delivered the most accurate and robust predictions for individual isotopes, showing the best balance between model complexity and generalization capability.

2. **Effective correction of spectral shapes:** the ICM model demonstrated an im-

portant ability in distinguishing between isotopes affected and unaffected by the Pandemonium effect. For affected isotopes, its predictions were significantly closer to the correct TAGS spectrum than the biased HPGe input, proving that the model had learned the underlying physics of the correction rather than simply reproducing the input.

3. **Successful Application to Actinide Spectra:** combining the predictions for all isotopes through the summation method, the resulting total antineutrino spectra for $^{235}$U and $^{238}$U showed a clear mitigation of the Pandemonium effect. The predicted spectra were closer to the target spectra (constructed using TAGS data) than the baseline spectra (constructed using HPGe data). However, important uncertainties in these predictions do affect this models and may make them useless for high precision measurement analysis.

In conclusion, this thesis has successfully designed and implemented a machine learning framework capable of providing a systematic correction to the summation method for the reactor antineutrino spectrum.

With this tool, the possibility to predict future TAGS measurements for isotopes probably affected by the Pandemonium effect is a very interesting one. Moreover, the Intrinsic Coregionalization Model could be applied to the input related to the isotopes listed by IAEA [55], hence the ones expected to suffer from Pandemonium effect. In the case the model predictions were substantially distant from the HPGe measurements, some indication could be extracted on which may be the priority isotopes to be measured with the TAGS technique.

On another note, to further validate the generalization abilities of the models discussed in this work, it would be interesting to verify their performance on different training-test splitting. Cross validation algorithms could be implemented to strongly improve the reliability of the predictions.

Finally, possible future work could focus on expanding the training dataset as new TAGS measurements become available, exploring more advanced machine learning architectures, and refining the uncertainty quantification of the final predictions.

# Bibliography


[1] W. Pauli. Letter to the group of radioactive people at the gauverein meeting in tübingen. 1930.

[2] E. Fermi. Versuch einer theorie der $\beta$-strahlen. i. *Zeitschrift für Physik*, 88:161–177, 1934. doi: 10.1007/BF01351864.

[3] A. De Angelis and M. Pimenta. *Introduction to Particle and Astroparticle Physics*. Springer, 2nd ed. edition, 2018.

[4] C. L. Cowan, F. Reines, F. B. Harrison, H. W. Kruse, and A. D. McGuire. Detection of the free neutrino: A confirmation. *Science*, 124(3212):103–104, 1956. doi: 10.1126/science.124.3212.103.

[5] Carl D. Anderson and Seth H. Neddermeyer. Cloud chamber observations of cosmic rays at 4300 meters elevation and near sea-level. *Physical Review*, 50(4):263–272, 1936. doi: 10.1103/PhysRev.50.263.

[6] Martin L. Perl, G. Abrams, A. Boyarski, M. Breidenbach, D. Briggs, W. Chinowsky, R. C. Duncan, J. O. Ehrlich, R. L. Eisele, M. S. Friedlander, J. H. Friedman, R. D. Heine, J. J. Seyler, and D. W. G. Sprott. Evidence for anomalous lepton production in $e^+e^-$ annihilation. *Physical Review Letters*, 35(18):1489–1492, 1975. doi: 10.1103/PhysRevLett.35.1489.

[7] G. Danby, J. M. Gaillard, K. Goulianos, L. M. Lederman, N. Mistry, M. Schwartz, and J. Steinberger. Observation of high-energy neutrino reactions and the existence of two kinds of neutrinos. *Phys. Rev. Lett.*, 9:36–44, 1962. doi: 10.1103/PhysRevLett.9.36.

[8] K. et al. [DONUT Collaboration] Kodama. Observation of tau neutrino interactions. *Phys. Lett. B*, 504:218–224, 2001. doi: 10.1016/S0370-2693(01)00336-0.

[9] R. Davis, D. S. Harmer, and K. C. Hoffman. Search for neutrinos from the sun. *Phys. Rev. Lett.*, 20:1205–1209, 1968. doi: 10.1103/PhysRevLett.20.1205.

[10] Y. Fukuda and Super-Kamiokande Collaboration. Evidence for oscillation of atmo-

spheric neutrinos. *Phys. Rev. Lett.*, 81:1562–1567, 1998. doi: 10.1103/PhysRevLett.81.1562.

[11] I. J. R. Aitchison and A. J. G. Hey. *Gauge Theories in Particle Physics: A Practical Introduction*, volume 2. CRC Press, 4th ed. edition, 2013.

[12] C. N. Yang and Robert L. Mills. Conservation of isotopic spin and isotopic gauge invariance. *Physical Review*, 96(1):191–195, 1954. doi: 10.1103/PhysRev.96.191.

[13] Steven Weinberg. *The Quantum Theory of Fields, Vol. 1: Foundations*. Cambridge University Press, 1995. ISBN 978-0521550017.

[14] Michael E. Peskin and Daniel V. Schroeder. *An Introduction to Quantum Field Theory*. Westview Press, 1995. ISBN 978-0201503975.

[15] Howard Georgi. *Lie Algebras in Particle Physics: From Isospin to Unified Theories*. Westview Press, 2 edition, 1999. ISBN 978-0738202334.

[16] Maurice Goldhaber, Lee Grodzins, and Andrew W. Sunyar. Helicity of neutrinos. *Physical Review*, 109(3):1015–1017, 1958. doi: 10.1103/PhysRev.109.1015. URL https://doi.org/10.1103/PhysRev.109.1015.

[17] Richard P. Feynman. Space-time approach to quantum electrodynamics. *Physical Review*, 76(6):769–789, 1949. doi: 10.1103/PhysRev.76.769.

[18] M. Altmann, F. von Feilitzsch, and U. Schanda. A parameterized pulse shape analysis method for the gallex solar neutrino experiment. *Nuclear Instruments and Methods in Physics Research A*, 366:352–361, 1996. doi: 10.1016/0168-9002(95)00987-0.

[19] J. N. Abdurashitov, V. N. Gavrin, S. Girin, and J. F. Wilkerson. Measurement of the solar neutrino capture rate with gallium metal. *Physical Review C*, 60:055801, 1999. doi: 10.1103/PhysRevC.60.055801.

[20] Particle Data Group. Review of solar neutrinos: Kamiokande and super-kamiokande. *Physics Letters B (Review section)*, 2005.

[21] SNO Collaboration. Neutrino observations from the sudbury neutrino observatory. *Physical Review C*, 2001.

[22] SNO Collaboration. Direct evidence for neutrino flavor transformation from neutral-current interactions in the sudbury neutrino observatory. *Physical Review Letters*, 2002. doi: 10.1103/PhysRevLett.89.011301.

[23] B. Pontecorvo. Mesonium and anti-mesonium. *Sov. Phys. JETP*, 6:429, 1957.

[24] Z. Maki, M. Nakagawa, and S. Sakata. Remarks on the unified model of elementary particles. *Prog. Theor. Phys.*, 28:870–880, 1962. doi: 10.1143/PTP.28.870.

[25] Ettore Majorana. Teoria simmetrica dell'elettrone e del positrone. *Il Nuovo Cimento*, 14:171–184, 1937. doi: 10.1007/BF02961314.

[26] S. Dell'Oro, S. Marcocci, M. Viel, and F. Vissani. Neutrinoless double beta decay: 2015 review. *Advances in High Energy Physics*, 2016:2162659, 2016. doi: 10.1155/2016/2162659.

[27] P. Minkowski. $\mu \to e\gamma$ at a rate of one out of 1-billion muon decays? *Phys. Lett. B*, 67:421–428, 1977. doi: 10.1016/0370-2693(77)90435-X.

[28] K. Zuber. *Neutrino Physics*. CRC Press, 3rd ed. edition, 2012.

[29] T2K Collaboration. Measurements of neutrino oscillation parameters from the t2k experiment using $3.6 \times 10^{21}$ protons on target. *European Physical Journal C*, 83(9): 782, 2023. doi: 10.1140/epjc/s10052-023-11819-x. URL https://link.springer.com/article/10.1140/epjc/s10052-023-11819-x.

[30] P. Adamson et al. First measurement of neutrino oscillation parameters using neutrinos and antineutrinos by nova. *Physical Review Letters*, 123:151803, 2020.

[31] F. P. An et al. Measurement of the reactor antineutrino flux and spectrum at daya bay. *Physical Review Letters*, 121:241805, 2018.

[32] J. H. Choi et al. Observation of energy-dependent reactor antineutrino disappearance at reno. *Physical Review Letters*, 116:211801, 2018.

[33] M. G. Aartsen et al. Measurement of atmospheric neutrino oscillations at 6–56 gev with icecube deepcore. *Physical Review Letters*, 120:071801, 2019.

[34] F. An et al. Neutrino physics with juno. *Journal of Physics G: Nuclear and Particle Physics*, 43:030401, 2016.

[35] A. Abusleme, T. Adam, S. Ahmad, R. Ahmed, S. Aiello, M. Akram, F. An, G. An, Q. An, G. Andronico, et al. Juno physics and detector. *Progress in Particle and Nuclear Physics*, 123:103927, 2022. doi: 10.1016/j.ppnp.2021.103927. URL https://doi.org/10.1016/j.ppnp.2021.103927.

[36] Vanessa Cerrone. Probing the atmospheric mass splitting with reactor antineutrino oscillations at juno. Master's thesis, Università degli Studi di Padova, Padova, Italia, 2023. URL https://thesis.unipd.it/retrieve/

59e2a997-571c-4444-87f5-9e04b1167589/Cerrone_Vanessa.pdf. Relatori: Alberto Garfagnini, Andrea Serafini.

[37] L. Wolfenstein. Neutrino oscillations in matter. *Phys. Rev. D*, 17:2369–2374, 1978. doi: 10.1103/PhysRevD.17.2369.

[38] S. P. Mikheev and A. Yu. Smirnov. Resonance amplification of oscillations in matter and spectroscopy of solar neutrinos. *Sov. J. Nucl. Phys.*, 42:913–917, 1985.

[39] Particle Data Group. Review of Particle Physics, 2024. URL https://pdg.lbl.gov.

[40] A. Abusleme et al. Sub-percent precision measurement of neutrino oscillation parameters with juno. *Chinese Physics C*, 46(12):123001, 2022. doi: 10.1088/1674-1137/ac8f02.

[41] A. Abusleme, T. Adam, S. Ahmad, S. Aiello, M. Akram, Nawab Ali, F. An, G. An, Q. An, G. Andronico, and many others (JUNO Collaboration). Tao conceptual design report: A precision measurement of the reactor antineutrino spectrum with sub-percent energy resolution. *arXiv preprint*, May 2020. URL https://arxiv.org/abs/2005.08745. arXiv:2005.08745 [physics.ins-det].

[42] Lorenzo Perissé, Anthony Onillon, Xavier Mougeot, Matthieu Vivier, Thierry Lasserre, Alain Letourneau, David Lhuillier, and Guillaume Mention. A comprehensive revision of the summation method for the prediction of reactor antineutrino fluxes and spectra. *Physical Review C*, 108(5):055501, November 2023. doi: 10.1103/PhysRevC.108.055501. arXiv:2304.14992 [nucl-ex].

[43] T. Adam, E. Baussan, K. Borer, J.-E. Campagne, N. Chon-Sen, M. Dracos, Y. Gornushkin, J.-L. Guyonnet, J. Janicsko Csathy, C. Jollet, F. Juget, A. Krasnoperov, Z. Krumstein, U. Moser, A. Nozdrin, A. Olchevski, A. Sadovski, P. Vilain, T. Wälchli, and G. (OPERA Collaboration) Wilquet. The opera experiment target tracker. *Nuclear Instruments and Methods in Physics Research Section A*, 577(3):523–539, July 2007. doi: 10.1016/j.nima.2007.04.147.

[44] P. Vogel and J. F. Beacom. The angular distribution of the reaction $\bar{\nu}_e + p \to e^+ + n$. *Physical Review D*, 60:053003, 1999. doi: 10.1103/PhysRevD.60.053003.

[45] A. Barresi, M. Borghesi, A. Cammi, D. Chiesa, L. Loi, M. Nastasi, E. Previtali, M. Sisti, S. Aiello, G. Andronico, V. Antonelli, D. Basilico, M. Beretta, A. Bergnoli, A. Brigatti, R. Brugnera, R. Bruno, A. Budano, B. Caccianiga, V. Cerrone, R. Caruso, C. Clementi, S. Dusini, A. Fabbri, G. Felici, F. Ferraro, A. Garfagnini, G. G. Giammarchi, N. Giudice, A. Gavrikov, M. Grassi, R. M.

Guizzetti, N. Guardone, B. Jelmini, C. Landini, I. Lippi, S. Loffredo, P. Lombardi, C. Lombardo, F. Mantovani, S. M. Mari, A. Martini, L. Miramonti, M. Montuschi, D. Orestano, F. Ortica, A. Paoloni, E. Percalli, F. Petrucci, G. Ranucci, A. C. Re, M. Redchuck, B. Ricci, A. Romani, P. Saggese, G. Sava, A. Serafini, C. Sirignano, L. Stanco, E. Stanescu Farilla, V. Strati, M. D. C. Torri, A. Triossi, C. Tuvè, C. Venettacci, G. Verde, and L. Votano. Analysis of reactor burnup simulation uncertainties for antineutrino spectrum prediction. *The European Physical Journal Plus*, 139(10): 952, 2024. doi: 10.1140/epjp/s13360-024-05704-z.

[46] G. Mention, M. Fechner, Th. Lasserre, Th. A. Mueller, D. Lhuillier, M. Cribier, and A. Letourneau. The reactor antineutrino anomaly. *Physical Review D*, 83(7):073006, April 2011. doi: 10.1103/PhysRevD.83.073006.

[47] Patrick Huber. Determination of antineutrino spectra from nuclear reactors. *Physical Review C*, 84(2):024617, 2011. doi: 10.1103/PhysRevC.84.024617.

[48] Th. A. Mueller, D. Lhuillier, M. Fallot, A. Letourneau, S. Cormon, M. Fechner, L. Giot, Th. Lasserre, J. Martino, G. Mention, M. Portier, J.-L. Reyss, M. Vivier, A. Derbin, Yu. Kozlov, I. Nemchenok, M. Skorokhvatov, and A. Yudin. Improved predictions of reactor antineutrino spectra. *Physical Review C*, 83(5):054615, 2011. doi: 10.1103/PhysRevC.83.054615.

[49] R. Perissé. *Modeling of Reactor Antineutrino Spectra*. PhD thesis, University of Lyon, 2023. URL https://hal.archives-ouvertes.fr/tel-03812345.

[50] Y. et al. (Double Chooz Collaboration) Abe. Measurement of the reactor antineutrino flux and spectrum at double chooz. *Journal of High Energy Physics*, 2019(1):163, 2019. doi: 10.1007/JHEP01(2019)163.

[51] A.C. Hayes and P. Vogel. Reactor neutrino spectra. *Annual Review of Nuclear and Particle Science*, 70:219–244, 2020. doi: 10.1146/annurev-nucl-021920-104702.

[52] A. Letourneau, M. Fallot, A. Tarifeño-Saldivia, et al. Origin of the reactor antineutrino anomalies in light of a new summation model with parameterized $\beta^-$ transitions. *Physical Review Letters*, 130(2):021801, 2023. doi: 10.1103/PhysRevLett.130.021801. URL https://link.aps.org/doi/10.1103/PhysRevLett.130.021801.

[53] J. C. Hardy, L. C. Carraz, B. Jonson, and P. G. Hansen. The essential decay of pandemonium: A demonstration of errors in complex $\beta$-decay schemes. *Physics Letters B*, 71(2):307–310, 1977. doi: 10.1016/0370-2693(77)90773-0.

[54] Jagdish K. Tuli. Evaluated nuclear structure data file: A manual for preparation of

data sets. Technical Report BNL-NCS-51655-Rev-98, Brookhaven National Laboratory, 2011. URL https://www.nndc.bnl.gov/ensdf/.

[55] Alan L. Nichols. Summary report of a consultants meeting, 15–17 december 2014, iaea, vienna. Technical Report IAEA-H9J1-R5QM, International Atomic Energy Agency, 2015. URL https://www-nds.iaea.org/publications/inddc/indc/indc-nds-0676/.

[56] R. C. Greenwood, R. G. Helmer, M. H. Putnam, and K. D. Watts. Measurement of $\beta^-$-decay intensity distributions of several fission-product isotopes using a total absorption $\gamma$-ray spectrometer. *Nuclear Instruments and Methods in Physics Research Section A*, 390(1-2):95–154, 1997. doi: 10.1016/S0168-9002(97)00356-2.

[57] Matteo Borghesi. Renshape. https://github.com/matteo-brg/RenShape/blob/master/doc/RS25.pdf, 2025.

[58] Xavier Mougeot / LNHB. BetaShape: Spectrum processing software for $\beta$ and ec decays. http://www.lnhb.fr/home/rd-activities/spectrum-processing-software/, 2023.

[59] D. A. Brown, M. B. Chadwick, R. Capote, A. C. Kahler, A. Trkov, et al. Endf/b-viii.0: The 8th major release of the nuclear reaction data library with cielo-project cross sections, new standards and thermal scattering data. *Nuclear Data Sheets*, 148: 1–142, 2018. doi: 10.1016/j.nds.2018.02.001.

[60] A. J. Plompen, O. Cabellos, C. De Saint Jean, M. Fleming, A. Algora, et al. The joint evaluated fission and fusion nuclear data library, jeff-3.3. *European Physical Journal A*, 56(181), 2020. doi: 10.1140/epja/s10050-020-00141-9.

[61] M. Wang, W. Huang, F.G. Kondev, G. Audi, and S. Naimi. The ame 2020 atomic mass evaluation (i). evaluation of input data, and adjustment procedures. *Chinese Physics C*, 45(3):030003, 2021. doi: 10.1088/1674-1137/abc4e7. Atomic Mass Evaluation (AME2020).

[62] Casey Alan Anderson, Gregg Walter McKinney, and James Robert Tutt. Neutron & gamma correlations using cgm in mcnp 6.2.0. Technical Report LA-UR-17-20353, Los Alamos National Laboratory, 2017. URL https://mcnp.lanl.gov/pdf_files/TechReport_2017_LANL_LA-UR-17-20353_AndersonMcKinneyEtAl.pdf.

[63] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006. ISBN 978-0387310732.

[64] Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, 1994. doi: 10.1162/neco.1994.6.2.215.

[65] A. Gavrikov, V. Cerrone, A. Serafini, R. Brugnera, A. Garfagnini, M. Grassi, B. Jelmini, L. Lastrucci, S. Aiello, G. Andronico, V. Antonelli, A. Barresi, D. Basilico, M. Beretta, A. Bergnoli, M. Borghesi, A. Brigatti, R. Bruno, A. Budano, B. Caccianiga, A. Cammi, R. Caruso, D. Chiesa, C. Clementi, S. Dusini, A. Fabbri, G. Felici, F. Ferraro, M. G. Giammarchi, N. Giudice, R. M. Guizzetti, N. Guardone, C. Landini, I. Lippi, S. Loffredo, L. Loi, P. Lombardi, C. Lombardo, F. Mantovani, S. M. Mari, A. Martini, L. Miramonti, M. Montuschi, M. Nastasi, D. Orestano, F. Ortica, A. Paoloni, E. Percalli, F. Petrucci, E. Previtali, G. Ranucci, A. C. Re, M. Redchuck, B. Ricci, A. Romani, P. Saggese, G. Sava, C. Sirignano, M. Sisti, L. Stanco, E. Stanescu Farilla, V. Strati, M. D. C. Torri, A. Triossi, C. Tuvè, C. Venettacci, G. Verde, and L. Votano. Interpretable machine learning approach for electron antineutrino selection in a large liquid scintillator detector. *Physics Letters B*, 860: 139141, 2025. doi: 10.1016/j.physletb.2024.139141.

[66] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016. ISBN 978-0262035613.

[67] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. ISBN 978-0262182539. URL http://www.gaussianprocess.org/gpml.

[68] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012. ISBN 978-0262018029.

[69] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989. doi: 10.1007/BF02551274.

[70] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. doi: 10.1016/0893-6080(91)90009-T.

[71] Moshe Leshno, Vladimir Y. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993. doi: 10.1016/S0893-6080(05)80131-5.

[72] Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012. doi: 10.1561/2200000016. URL https://doi.org/10.1561/2200000016.

[73] Edwin V. Bonilla, Kian M. A. Chai, and Christopher K. I. Williams. Multi-task gaussian process prediction. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems (NIPS 20)*, pages 153–160. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2007/file/30836cef983b31849e23eaa1d3e88d64-Paper.pdf.

[74] Leonardo Maturi. Supplement material: test and validation set plots for single isotopes spectral shapes and total actinide antineutrino spectra. https://github.com/leomaturi/Master_thesis_prediction_plots, 2025.

[75] Ian T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2nd edition, 2002. ISBN 978-0387954424.

[76] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2nd edition, 2004. ISBN 978-0387212395.

[77] James E. Gentle. *Computational Statistics: Monte Carlo Simulation*. Springer, New York, 2009. ISBN 978-0387981451.

[78] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1050–1059. PMLR, 2016. URL https://proceedings.mlr.press/v48/gal16.html.

# A | Appendix A

## A.1. General Training Techniques for Machine Learning Models

The main characteristics of the machine learning methods employed in this thesis have been presented in Ch. 4. However, few has been said about common practices used to train them. This section of the appendix is intended to fill this hole, describing important concepts that play an important role for model training in the present research.
The training of a machine learning model consists in the determination of the parameters of the model itself. As seen in Sec. 4.2, this aim is reached through the maximum likelihood principle. The techniques presented here are employed, as a common goal, to improve the convergence properties of the model, but also to guarantee a good generalization and to avoid overfitting phenomena.

### A.1.1. Training Iterations and Epochs

In the context machine learning, the term epoch, already introduced in Sec. 4.3.1, represents a full scan performed by the model on the training set. An epoch corresponds to the whole process in which every observation of the training set is used, at least one time, to update the parameters of the model thanks to the chosen optimization technique, such as gradient descent. The learning phase of a machine learning model is organized in many consecutive epochs. At the end of every epoch, the cost function should be decreased if confronted with the previous epoch. During each epoch, the training data can be subdivided in smaller *batches*, that can help in:

- reducing the computational load and memory occupation necessary for the gradient descent algorithm,
- introducing a degree of randomness in the parameters update, that can be useful to avoid local minima in the cost function,
- improve the generalization properties of the model.

At every epoch the model makes a sequence of iteration steps, equal to the number of batches necessary to cover the whole dataset. After every epoch, a common practice is to evaluate the performance metrics, such as the NLML of Sec. 4.3.1, both on the training and validation set. The concept of validation set was already introduce in Sec. 4.2, in the context of hyperparameters determination. Here it is again useful. The validation set is a part of the training set which is not seen during model training. In this way it is possible to monitor the model learning abilities on new data, and possibly identify overfitting phenomena. The number of epochs is an hyperparameter of the model.

### A.1.2. Gradient Descent and Optimization

Gradient descent is the standard method for the optimization of parametric models. Indeed it was already introduced in Sec. 4.3.2. Given a generic cost function $L(\boldsymbol{\theta})$, the model parameters are update at every epoch using the rule:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \, \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_t), \tag{A.1}$$

where $\eta$ is the learning rate. The choice of the learning rate is crucial: too high values can lead to numerical instabilities and divergence, while too low ones can excessively slow down convergence. One of the most used extension of gradient descent is the *Adam* (Adaptive Moment Estimation) algorithm. It introduces adaptive estimates of the first and second moment of the gradient. In this way it is particularly effective with deep neural networks or with noisy data [66]. The adaptive optimization adjusts the effective learning step for each parameter based on the first and second moments of the gradients, typically leading to larger steps at the beginning and smaller ones near convergence, improving stability when the loss surface is poorly conditioned[1]. In addition to adaptive optimizers, a learning rate scheduler can be employed to explicitly control the variation of the learning rate during training. This mechanism gradually decreases the global learning rate according to a predefined rule or in response to the model's performance (, allowing the optimization process to refine the convergence towards a local or global minimum more smoothly.

### A.1.3. Early Stopping e Validation

A crucial part of the learning phase consists in the evaluation, and control, of possible overfitting phenomena. A simple yet very effective strategy is the so called *early stopping*. It consists in interrupting the training when the performance of the model on the

[1]Loss here represents a generic way to indicate the cost function to be minimized during training.

validation set stops to decrease. During training, the *training loss* constantly decreases, while the validation loss, which is computed on previously unseen data, reaches a minimum and then starts to increase, signaling an excessive adaptation of the model to the training data. Practically, the validation loss $L_{\text{val}}(\boldsymbol{\theta}_t)$ is monitored at every epoch, and optimization is stopped when

$$L_{\text{val}}(\boldsymbol{\theta}_t) > L_{\text{val}}(\boldsymbol{\theta}_{t-p}), \tag{A.2}$$

for a previously fixed number of epochs $p$, called *patience*. As reported in [63], early stopping can be seen as an implicit form of regularization, as it reduces the overall capacity of the model to avoid excessive adaptation to training data or noise.

### A.1.4. Link with Bayesian Models

In Bayesian probabilistic models such as Gaussian processes (see Sec. 4.3.1), optimization does not involve explicit parameters - weights and bias - as in a neural network. Conversely, it involves the hyperparameters of the kernel and of the noise. As described in [67], the estimate occurs through the maximization of the marginal log likelihood:

$$\log p(y \mid \boldsymbol{X}, \boldsymbol{\theta}) = -\frac{1}{2} y^\top K_{\boldsymbol{\theta}}^{-1} y - \frac{1}{2} \log |K_{\boldsymbol{\theta}}| - \frac{n}{2} \log 2\pi, \tag{A.3}$$

where $K_{\boldsymbol{\theta}}$ is the kernel or covariance matrix. However, the choice of optimizers and stopping criteria is follows the sample principle described above, while keeping at the same time a Bayesian interpretation.

### A.1.5. Data Scaling

An aspect that is often underestimated, but fundamental to the effectiveness of training machine Learning models, is the scaling of input and target data. Most optimization algorithms, in fact, present a sensitivity to the scale of variables, as the presence of features with very different orders of magnitude can determine a non-homogeneous behavior of the gradient, slowing down convergence or leading to suboptimal local minima. Scaling techniques therefore aim to bring the different components of the dataset back to comparable scales, while preserving the relative relationships between the variables. The most common methods are normalization, that transforms every variable in order to obtain values in the [0,1] interval, and standardization, that instead centers and reshapes the data in order to get null mean and unitary variance. The latter is usually preferred in gradient descent based models, since it favors a more stable propagation of the gradient during the

training phase. In the case of probabilistic models, such as Gaussian processes, scaling also plays a conceptual role: the kernel itself can be interpreted as a measure of similarity between points in the input space, and an inadequate scale of variables can significantly alter the kernel-induced metric, compromising the quality of modeling. In general, data scaling therefore represents an essential preprocessing step to guarantee numerical stability, interpretation of parameters and optimal performance during the training phase.

As a general rule, these global operations on the dataset must be defined only on the training set. Once they are defined, they can then be applied to the test and validation set. This is done to carefully avoid data leakage into the model, hence to avoid that any information about the test set leaks in the training set. As an example, consider the case of standardization. It requires the computation of the mean and standard deviation of the dataset. If however, these quantities are computed both on the test and training set, the standardized training data would then be polluted with information on the test set, and vice versa. That's why, the mean and standard deviation are only computed on the training set, and then the required manipulations on the datasets are performed separately.

## A.2. Reduced-Order Modeling

Reduced-Order Modeling (ROM) is a computational technique that reduces the complexity of mathematical models in numerical simulations. Nowadays in fact, the large size of data and the complex models employed in a simulation can make them infeasible. ROM enables to reduce the dimensional weight of the data, projecting them into lower dimension variables, while preserving their informational content. These technique can indeed be used to evaluate latent patterns and correlations in the data. ROM can then be seen also as way to extract principal characteristics of the dataset. Another practical application in which ROM is involved is visualization: reducing a dataset into two or three dimensional variables enables to produce diagrams which may contain useful visual information. For all these reasons, ROM represents an important tool for the purposes of this thesis. This section of the appendix is dedicated to Principal Component Analysis and Autoencoders, respectively linear and non-linear techniques for model order reduction. Their application in the context of Pandemonium effect mitigation will be discussed extensively in Ch. 5.

### A.2.1. Principal Component Analysis

The fundamental principle of Principal Component Analysis (PCA) consists in the simplification of a set of data characterized by many correlations between the variables compos-

ing it. The aim is to reduce dimensionality while maintaining the data variability. This goal is reached through a linear transformation into a new space of variables, called the *principal components*. They result as independent and ordered by the quantity of data variance they can explain. In this way, the first principal components represent the most significative part of the total information [75]. Consider a vector $\boldsymbol{x}$ of $p$ random variables. Suppose the interest is in the variances of the $p$ random variables and the structure of the covariances or correlations between them. Unless the $p$ variables are small, or the data structure is very simple, it can be very unpractical to look for all the $p$ variances and the $\frac{1}{2}p(p-1)$ covariances. Another approach can be to look for a fewer ($\ll p$) number of informative variables, preserving the content of the variances and covariance or correlation matrix. In particular, PCA concentrates on the variances, while still considering covariances and correlations.
The first step requires to identify a linear function $\boldsymbol{\alpha}_1^\top \boldsymbol{x}$ of the elements of $\boldsymbol{x}$ having maximum variance. $\boldsymbol{\alpha}_1$ is a vector of $p$ constants $\alpha_{11}, \alpha_{12}, ..., \alpha_{1p}$ so that:

$$\boldsymbol{\alpha}_1^\top \boldsymbol{x} = \alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1p}x_p = \sum_{j=1}^{p} \alpha_{1j}x_j. \tag{A.4}$$

Next, look for another linear function $\boldsymbol{\alpha}_2^\top \boldsymbol{x}$, that is uncorrelated with $\boldsymbol{\alpha}_1^\top \boldsymbol{x}$ and has maximum variance. Repeating this procedure, at the $k^{th}$, a linear function $\boldsymbol{\alpha}_k^\top \boldsymbol{x}$ is found to have maximum variance and being uncorrelated to $\boldsymbol{\alpha}_1^\top \boldsymbol{x}$, $\boldsymbol{\alpha}_2^\top \boldsymbol{x}$, ... ,$\boldsymbol{\alpha}_{k-1}^\top \boldsymbol{x}$. The $k^{th}$ derived variable, $\boldsymbol{\alpha}_k^\top \boldsymbol{x}$ is the $k^{th}$ principal component (PC). One can find up to $p$ principal components, but clearly, for all the reasons mentioned above one, one hopes that the most of the variance is explained by a number $m \ll p$ of principal components.
Now that PCs are defined, one needs to understand how to find them. Suppose, for the moment, that the vector $\boldsymbol{x}$ has a known covariance matrix $\boldsymbol{\Sigma}$. The more realistic case, where $\boldsymbol{\Sigma}$ is unknown, follows by replacing $\boldsymbol{\Sigma}$ by a sample covariance matrix $\boldsymbol{S}$. It turns out that the $k^{th}$ PC given by $z_k = \boldsymbol{\alpha}_k^\top \boldsymbol{x}$, where $\boldsymbol{\alpha}_k$ is the eigenvector of $\Sigma$ corresponding to the $k^{th}$ largest eigenvalue $\lambda_k$. Moreover, if $\boldsymbol{\alpha}_k^\top \boldsymbol{\alpha}_k = 1$ is imposed to have unit length, then $var[z_k] = \lambda_k$, where $var[z_k]$ is the variance of $z_k$. The derivation follows [75].

**Principal components derivation** Consider now the first $\boldsymbol{\alpha}_1^\top \boldsymbol{x}$. The vector $\boldsymbol{\alpha}_1$ maximized $var[\boldsymbol{\alpha}_1^\top \boldsymbol{x}] = \boldsymbol{\alpha}_1^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_1$. It is clear that the maximum will not be achieved for a finite $\boldsymbol{\alpha}_1$, so some normalization constraint must be imposed. The constraint used in this derivation is $\boldsymbol{\alpha}_k^\top \boldsymbol{\alpha}_k = 1$.
The maximization of $var[\boldsymbol{\alpha}_1^\top \boldsymbol{x}] = \boldsymbol{\alpha}_1^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_1$ constrained by $\boldsymbol{\alpha}_k^\top \boldsymbol{\alpha}_k = 1$ typically occurs

through the Lagrange multipliers technique. Maximize:

$$\boldsymbol{\alpha}_1^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_1 - \lambda(\boldsymbol{\alpha}_k^\top \boldsymbol{\alpha}_k - 1), \tag{A.5}$$

where $\lambda$ is the Lagrange multiplier. Differentiating with respect to $\boldsymbol{\alpha}_1$ one obtains:

$$(\boldsymbol{\Sigma} - \lambda \boldsymbol{I}_p)\boldsymbol{\alpha}_1 = 0, \tag{A.6}$$

where $\boldsymbol{I}_p$ is the $(p \times p)$ identity matrix. Thus, $\boldsymbol{\alpha}_1$ is the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{\Sigma}$, and $var[\boldsymbol{\alpha}_1^\top \boldsymbol{x}] = \boldsymbol{\alpha}_1^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_1 = \lambda_1$ is the largest eigenvalue. The quantity to be maximized to have maximum variance is $\lambda$. In general, the $k^{th}$ PC of $\boldsymbol{x}$ is $\boldsymbol{\alpha}_k^\top \boldsymbol{x}$ and $var(\boldsymbol{\alpha}_k^\top \boldsymbol{x}) = \lambda_k$, where $\lambda_k$ is the $k^{th}$ largest eigenvalue of $\boldsymbol{\Sigma}$, and $\boldsymbol{\alpha}_k$ is the corresponding eigenvector [75]. This will now be proved for k = 2. The second principal component $\boldsymbol{\alpha}_2^\top \boldsymbol{x}$ maximizes $\boldsymbol{\alpha}_2^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_2$, while constrained to be uncorrelated to $\boldsymbol{\alpha}_1^\top \boldsymbol{x}$, hence asking $cov[\boldsymbol{\alpha}_1^\top \boldsymbol{x}, \boldsymbol{\alpha}_2^\top \boldsymbol{x}] = 0$. It holds that:

$$cov[\boldsymbol{\alpha}_1^\top \boldsymbol{x}, \boldsymbol{\alpha}_2^\top \boldsymbol{x}] = \boldsymbol{\alpha}_1^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_2 = \boldsymbol{\alpha}_2^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_2^\top \lambda_1 \boldsymbol{\alpha}_1 = \lambda_1 \boldsymbol{\alpha}_2^\top \boldsymbol{\alpha}_1 = \lambda_2 \boldsymbol{\alpha}_1^\top \boldsymbol{\alpha}_2. \tag{A.7}$$

so any of the equations:

$$\boldsymbol{\alpha}_1^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_2 = 0,\ \boldsymbol{\alpha}_2^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_1 = 0,\ \boldsymbol{\alpha}_1^\top \boldsymbol{\alpha}_2 = 0,\ \boldsymbol{\alpha}_2^\top \boldsymbol{\alpha}_1 = 0. \tag{A.8}$$

can be used to impose the constraint. In this case the choice is to impose $\boldsymbol{\alpha}_2^\top \boldsymbol{\alpha}_1 = 0$, which is completely arbitrary. Now maximize:

$$\boldsymbol{\alpha}_2^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_2 - \lambda(\boldsymbol{\alpha}_2^\top \boldsymbol{\alpha}_2 - 1) - \phi(\boldsymbol{\alpha}_2^\top \boldsymbol{\alpha}_1), \tag{A.9}$$

where $\lambda, \phi$ respectively are the normalization and no-correlation Lagrange multipliers. Differentiation with respect to $\boldsymbol{\alpha}_2$ gives:

$$\boldsymbol{\Sigma} \boldsymbol{\alpha}_2 - \lambda \boldsymbol{\alpha}_2 - \phi \boldsymbol{\alpha}_1 = 0, \tag{A.10}$$

and multiplication of this equation by $\alpha_1^\top$ on the left gives:

$$\boldsymbol{\alpha}_1^\top \boldsymbol{\Sigma} \boldsymbol{\alpha}_2 - \lambda \boldsymbol{\alpha}_1^\top \boldsymbol{\alpha}_2 - \phi \boldsymbol{\alpha}_1^\top \boldsymbol{\alpha}_1 = 0, \tag{A.11}$$

which, since the first two terms are zero and $\boldsymbol{\alpha}_1^\top \boldsymbol{\alpha}_1 = 1$, reduces to $\phi = 0$. Thus, $\boldsymbol{\Sigma}\boldsymbol{\alpha}_2 - \lambda\boldsymbol{\alpha}_2 = 0$, which is equivalent to:

$$(\boldsymbol{\Sigma} - \lambda \boldsymbol{I}_p)\boldsymbol{\alpha}_2 = 0, \tag{A.12}$$

so that $\lambda$ is an eigenvalue of $\boldsymbol{\Sigma}$ corresponding to the eigenvector $\boldsymbol{\alpha}_2$. Again $\lambda$ needs to be as large as possible. Assuming that $\boldsymbol{\Sigma}$ has no repeated eigenvalues, $\lambda$ cannot be equal to $\lambda_1$. If it did, it follows that $\boldsymbol{\alpha}_2 = \boldsymbol{\alpha}_1$, violating the constraint $\boldsymbol{\alpha}_1^\top \boldsymbol{\alpha}_2 = 0$. Hence $\lambda$ is the second largest eigenvalue of $\boldsymbol{\Sigma}$, and $\boldsymbol{\alpha}_2$ is the corresponding eigenvector.
It can be shown that for the third, fourth, . . . , $p^{th}$ PCs, the vectors of coefficients $\boldsymbol{\alpha}_3, \boldsymbol{\alpha}_4, ..., \boldsymbol{\alpha}_p$ are the eigenvectors of $\boldsymbol{\Sigma}$ corresponding to $\lambda_3, \lambda_4, ..., \lambda_p$, the third and fourth largest, . . . , and the smallest eigenvalue, respectively [75].

**Example: PCA on a 2D dataset** A simple example of PCA application is illustrated here. Consider a simple dataset of $N$ observations in $\mathbb{R}^2$:

$$\boldsymbol{X} = \begin{bmatrix} 2.5 & 2.4 \\ 0.5 & 0.7 \\ 2.2 & 2.9 \\ 1.9 & 2.2 \\ 3.1 & 3.0 \\ 2.3 & 2.7 \\ 2.0 & 1.6 \\ 1.0 & 1.1 \\ 1.5 & 1.6 \\ 1.1 & 0.9 \end{bmatrix}. \tag{A.13}$$

Centering of the data is obtained by subtraction of the mean of each column. The sample covariance matrix $\boldsymbol{S}$ is computed as:

$$\boldsymbol{S} = \frac{1}{N-1}(\boldsymbol{X} - \bar{\boldsymbol{X}})^\top(\boldsymbol{X} - \bar{\boldsymbol{X}}) = \begin{bmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{bmatrix}. \tag{A.14}$$

The eigenvalues of $\boldsymbol{S}$ are $\lambda_1 \approx 1.284$, $\lambda_2 \approx 0.049$, with the corresponding eigenvectors:

$$\boldsymbol{\alpha}_1 \approx \begin{bmatrix} 0.677 \\ 0.736 \end{bmatrix}, \qquad \boldsymbol{\alpha}_2 \approx \begin{bmatrix} -0.736 \\ 0.677 \end{bmatrix}. \tag{A.15}$$

The first PC $z_1 = \boldsymbol{\alpha}_1^\top \boldsymbol{x}$ explains the majority of the variance ($\lambda_1/(\lambda_1+\lambda_2) \approx 0.963$), while the second PC $z_2 = \boldsymbol{\alpha}_2^\top \boldsymbol{x}$ is left to explain only the residual part. Accordingly, the data projection towards $\boldsymbol{\alpha}_1$ enables to effectively reduce the dimensionality of the data from 2 to 1, and conserves more than 96% of the original variance. This illustrates the power of principal component analysis in the identification of directions of maximum variation in a dataset.

### A.2.2. Autoencoders

Autoencoders (AE) are a particular kind of neural network architecture. They are designed to learn efficient and compact representations of data. While classic neural networks can be used for supervised regression, hence for the prediction of explicit target variables, autoencoders are unsupervised learning models: their aim is to reconstruct the input itself. The model receives an input vector $\boldsymbol{x} \in \mathbb{R}^d$ and transforms it into a reduced order representation, with smaller dimension than the input, typically called *code* or *embedding*. The AE then tries to reconstruct an output $\hat{\boldsymbol{x}}$, which should be as close as possible to the original input $\boldsymbol{x}$. As PCA, the founding principle consists in learning informative internal structures of the dataset, removing noise and repetitions.

**Structure and functioning** The autoencoder architecture is composed of two main parts, called *encoder* and *decoder*. These are two neural networks working in sequence for the purposes of the AE. The encoder's role is to map the input $\boldsymbol{x}$ of dimension $d$ in a latent space with reduced dimension $k < d$:

$$\boldsymbol{h} = f_{\text{enc}}(\boldsymbol{x}; \boldsymbol{\theta}_{\text{enc}}), \tag{A.16}$$

where $\boldsymbol{h} \in \mathbb{R}^k$ represents the latent code and $\boldsymbol{\theta}_{\text{enc}}$ are the parameters of the encoder, hence its weights and bias. The decoder operates the inverse transformation. The aim is to rebuild the original input starting from the latent code $\boldsymbol{h}$:

$$\hat{\boldsymbol{x}} = f_{\text{dec}}(\boldsymbol{h}; \boldsymbol{\theta}_{\text{dec}}), \tag{A.17}$$

with $\boldsymbol{\theta}_{\text{dec}}$ parameters of the decoder. In summary, the whole AE network implements the following composite function:

$$\hat{\boldsymbol{x}} = f_{\text{dec}}(f_{\text{enc}}(\boldsymbol{x})) = f(\boldsymbol{x}; \boldsymbol{\theta}), \tag{A.18}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}}\}$. The training phase consists in the minimization of a cost function measuring the difference between the original input $\boldsymbol{x}$ and the reconstructed output $\hat{\boldsymbol{x}}$. Usually, the mean squared error (MSE) is used:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{x}^{(i)} - \hat{\boldsymbol{x}}^{(i)}\|^2, \tag{A.19}$$

but other options can also be employed.

**Dimensionality reduction and latent representation** The interesting characteristic of autoencoders consists in the so-called *bottleneck*, meaning the hidden layer that forces the model to compress information, thanks to its low dimension. The coded information $\boldsymbol{h}$ is interpreted as a latent representation of the original data, similar to the one produced with linear techniques as PCA, introduced in Sec. A.2.1. Accordingly, a linear autoencoder, built with linear activation functions and MSE, is mathematically equivalent to the principal component analysis. The non-linear character is introduced through non-linear activation functions, such as those presented in Sec. 4.3.2. AE become in this way a non-linear generalization of PCA, with the ability to capture possible non-linear, thus more complex, relationships in the data [66]. For this reason they are, if trained correctly, a very powerful instrument for order reduction, denoising, and feature extraction. This is even more true when the data are contained in non-linear manifold structures in euclidean space.

**Example: compression of digit images** A simple example describing a possible application of autoencoders is here presented. Recover the context of the first example considered in Ch. 4. A dataset containing handwritten digit, such as the one in Fig. 4.1, is made of $28 \times 28$ pixel images. Each one of them can then be displayed as a vector with dimension 784. The encoder can be trained to reduce this high dimension vector to a simpler, latent, representation, for instance of dimension 32. The decoder then learns to invert the encoding process, trying to bring the latent vector to its original higher order dimension. In such a way, the neural network implicitly learns an internal coordinate system, or representation, where similar digits are closer in the space of latent vectors.

## A.3. Monte Carlo Methods

The last problem encountered in this appendix is the uncertainty estimation of a machine learning model prediction. As described in Sec. 4.2, Bayesian models, intrinsically have this characteristic: the structure of the model directly produces a predictive distribution, hence is able to predict variances in addition to point values. On the other hand, for non Bayesian models, as neural networks, it is necessary to introduce additional methodologies for the estimation of the prediction uncertainty.
An effective strategy is the one of Monte Carlo (MC) methods [76, 77], that approximate probability distributions thanks to repeated samplings.
In the neural network context, a particularly diffused approach is the one of Monte Carlo Dropout [78]. Dropout, very used also for regularization purposes, is the founding principle. During the forward pass, some neurons in the network are randomly deactivated accordingly to some fixed probability. By making multiple forward passes on the same input with active dropout, we obtain an output distribution, the variance of which provides an estimate of the uncertainty of the network. This technique therefore allows us to derive confidence intervals on predictions without the need to build a complete Bayesian model. Differently from regularization, where dropout is active only in the learning phase, in MC Dropout it active also during the prediction one. For Gaussian processes, predictive uncertainty is already defined by the model: every prediction is a multivariate Gaussian distribution, with analytically differentiable mean and covariance. However, in multitask contexts or in the presence of intermediate transformations, such as dimensionality reduction via PCA, it may be useful to resort to Monte Carlo sampling to propagate uncertainties through nonlinear transformations and obtain reliable estimates of the final variance on observable quantities. In this case, each sample of the predictive distribution is transformed into the original space via the inverse scaling, and possibly log-transform and PCA operations. The empirical variance of the samples provides the measure of the model's uncertainty about real predictions. In summary, Monte Carlo methods represent a versatile approach for uncertainty estimation in both Bayesian and non-Bayesian models. In the case of non-Bayesian neural networks, MC Dropout allows predictive uncertainties to be obtained using an already familiar mechanism such as dropout, while in Gaussian multitask processes sampling allows uncertainty to be propagated through complex transformations of predictions, guaranteeing a consistent measure of the final variance. This approach therefore naturally fits into the training and validation workflow of the models, completing the techniques discussed in the previous paragraphs, such as regularization, loss monitoring and early stopping.

# List of Figures

1.1 Schematic representation of the elementary particles described in the original Standard Model of Particle Physics and their fundamental properties. They are subdivided in: quarks (in purple), leptons (in green) and gauge bosons (in red). The Higgs boson (in yellow) is the particle associated to the Higgs field, responsible for the mass acquisition mechanism of the SM. 3
1.2 Main nuclear fusion reactions contributing to the solar neutrino flux. The figure is taken from [3]. . . . . 4
1.3 Image of the Sun obtained from a 500-day Super-Kamiokande run. The figure is taken from [3]. . . . . 5
1.4 Example of neutrino oscillations in the two-flavour scheme. The dashed lines show the average oscillation probabilities. The figure is taken from [28]. 9
1.5 Schematic illustration of the neutrino mass ordering. The figure is taken from [36] and refers to non-updated oscillation parameters. . . . . 13

2.1 Location of the JUNO and TAO experimental site. The main 20 kton LS detector is indicated in blue [34], at a baseline of around 52.5 km from the Yangjiang NPP and the Taishan NPP. The 2.8 ton TAO detector [41], indicated in red, is located about 30 meters away from one of the Taishan reactor cores. . . . . 16
2.2 Top: Expected oscillated antineutrino spectrum at JUNO without considering detector effect. The non-oscillated spectrum is also shown, in gray. Bottom: $\bar{\nu}_e$ survival probability at a baseline of 52.5 km. The figure is taken from [36]. . . . . 17
2.3 A schematic view of the JUNO detector design. . . . . 18
2.4 Scheme of the Central Detector (CD). The spherical Acrylic Vessel is supported by 590 Connecting Bars, constituting a spherical SS structure. The shell structure sits on a bearing consisting of 30 pairs of Supporting Legs made of stainless steel structures. . . . . 19
2.5 Feynman diagram of the inverse beta decay reaction on protons. . . . . 21

2.6 Schematic illustration of the IBD kinematics. The time-charge diagram is not in scale. The figure is taken from [36]. . . . . 22

3.1 Illustration of $\beta^-$ decay in nuclei. . . . . 27

3.2 The figure shows the electron ($\beta$, in orange) and antineutrino ($\bar{\nu}_e$, in green) spectra associated to a single transition of $^{142}$Ba. This represents the crucial step in which the $\bar{\nu}_e$ spectrum is obtained from the $\beta$ one by substituting the electron energy E with the antineutrino energy $E_{\bar{\nu}} = E^p_{0,b} - E$, where $E^p_{0,b}$ is the endpoint energy of the transition. These spectra are normalized to the branching ratio of the specific $\beta$ transition to which they are associated. The weighted sum of the obtained transitions (see Eq. 3.4) then gives the overall isotope spectrum, as shown in Fig. 3.3. This is the founding characteristic of the ab-initio approach. . . . . 29

3.3 Example of computation of the total antineutrino spectrum $S_p(E_{\bar{\nu}})$ for $^{142}$Ba. The single $\beta$ transition spectra $S^p_b(E_{\bar{\nu}}, E^p_{0,b})$ are the blue-shaded ones, while the total $S_p(E_{\bar{\nu}})$ spectrum is the black one. In the legend, the transitions type is indicated, as well as their endpoint energy: $a$ stands for allowed, while $1nu$ stands for first forbidden non-unique. . . . . 30

3.4 Ratio of the measured to the predicted IBD spectrum at various short-baseline experiments. The 5 MeV bump is well identified. The figure is taken from [49]. . . . . 33

3.5 Schematic illustration of the Pandemonium effect. The gamma rays emitted from high energy levels (in red) are not detected, and result in an overestimation of the low energy levels (in green) beta feedings. The figure is taken from [49]. . . . . 36

3.6 Examples of comparison between HPGe and TAGS measurements. Upper figure: the comparison is shown for $^{87}$Br. The isotope shows a significative Pandemonium effect. Lower figure: the comparison is shown for 142Ba. Pandemonium effect is practically absent. . . . . 38

3.7 Block scheme of the RenShape workflow presented in Sec. 3.6. . . . . 41

4.1 Examples of hand-written digits taken from US zip codes. The figure is taken from [63]. . . . . 44

4.2 Example of underfitting, appropriate capacity and overfitting for a model aiming to fit the sample points with a function. The figure is taken from [66]. . . . . 47

4.3 The typical relationship between model capacity and error shows that the trend of training error and test error does not coincide. When the capacity of the model is very low, both the error on training and the generalization error are high: this is the underfitting phase. As capacity increases, the training error decreases, but at the same time the gap between training error and generalization error grows. At some point, the increment of this gap outweighs the benefit of training error reduction, thus entering the overfitting phase, in which the capacity of the model is excessive in relation to the task and exceeds the optimal one. The figure is taken from [66]. . . . . 48

5.1 Examples of transition properties (branching ratios and endpoint energies) comparison between HPGe and TAGS measurements. Left figure: the comparison is shown for $^{87}$Br. The isotope shows a significative Pandemonium effect, indeed the transitions properties change in a visible way. Right figure: the comparison is shown for 142Ba. Pandemonium effect is practically absent, indeed the transitions properties almost exactly superpose for HPGe and TAGS. . . . . 66

5.2 Simplified block diagram of the ML regression implemented for the purposes of this thesis. . . . . 67

5.3 Principal component representation of the selection performed on the large number of candidates. The selected SAFE data (in green) are the ones closer to the TAGS data (in blue). . . . . 70

5.4 The trends of the autoencoder training (in blue) and validation (in orange) losses are plotted against the number of epochs. No overfitting appears, confirming the effectiveness of the validation based early stopping. . . . . 75

5.5 Upper figure: the mean squared error (MSE) of reconstruction is shown as a function of the number of principal components. Lower figure: the explained variance fraction is shown as a function of the number of principal components. These quantities are shown for the training (in blue), the validation (in orange) and the test (in green) sets. . . . . 77

5.6 The figure shows the first 15 principal components of the spectral dataset. These represent the spectral patterns through which it is possible to reconstruct the majority of the information contained in the antineutrino spectra of the dataset. Note that, approximately after the $35^{th}$ bin, no signal is found, as a consequence of the padding operation. . . . . 78

5.7 The training (in blue) and validation (in orange) losses of the ICM model are shown against the number of epochs. The training curve reaches a minimum which is higher than the final value of -0.2801. This often occurs in Gaussian process. The values obtained during learnig can indeed be influenced by the training mode. . . . . 83
5.8 Training (in blue) and validation (in orange) losses of the LCM model against the number of epochs. As in ICM, the training curve minimum value in training mode can differ from the final evaluation loss. . . . . 84
5.9 Comparison of shallow and deep neural network training phase. No major difference is observed in the convergence of the two models, even if the deep neural network is twice deeper than the shallow one. . . . . 86

6.1 The figure shows the performance of the models considered in this work for $^{147}$Pr. Fig. 6.1a refers to the ICM, Fig. 6.1b refers to the LCM, Fig. 6.1c refers to the SNN, and Fig. 6.1d refers to the DNN. . . . . 92
6.2 The figure shows the performance of the models considered in this work for $^{145}$Ce. Fig. 6.2a refers to the ICM, Fig. 6.2b refers to the LCM, Fig. 6.2c refers to the SNN, and Fig. 6.2d refers to the DNN. . . . . 93
6.3 The figure shows the performance of the models considered in this work for $^{142}$Cs. Fig. 6.3a refers to the ICM, Fig. 6.3b refers to the LCM, Fig. 6.3c refers to the SNN, and Fig. 6.3d refers to the DNN. . . . . 94
6.4 The figure shows the performance of the models considered in this work for $^{93}$Rb. Fig. 6.4a refers to the ICM, Fig. 6.4b refers to the LCM, Fig. 6.4c refers to the SNN, and Fig. 6.4d refers to the DNN. . . . . 95
6.5 The figure shows the performance of the models considered in this work for $^{117}$Ag. Fig. 6.5a refers to the ICM, Fig. 6.5b refers to the LCM, Fig. 6.5c refers to the SNN, and Fig. 6.5d refers to the DNN. . . . . 96
6.6 The figure shows the performance of the models considered in this work for $^{148m}$Pr. Fig. 6.6a refers to the ICM, Fig. 6.6b refers to the LCM, Fig. 6.6c refers to the SNN, and Fig. 6.6d refers to the DNN. . . . . 97
6.7 The figure shows the performance of the models considered in this work for $^{143}$Ba. Fig. 6.7a refers to the ICM, Fig. 6.7b refers to the LCM, Fig. 6.7c refers to the SNN, and Fig. 6.7d refers to the DNN. . . . . 98
6.8 The figure shows the performance of the models considered in this work for $^{155}$Nd. Fig. 6.8a refers to the ICM, Fig. 6.8b refers to the LCM, Fig. 6.8c refers to the SNN, and Fig. 6.8d refers to the DNN. . . . . 99

6.9 The figure shows the performance of the models considered in this work for $^{151}$Pr. Fig. 6.9a refers to the ICM, Fig. 6.9b refers to the LCM, Fig. 6.9c refers to the SNN, and Fig. 6.9d refers to the DNN. . . . . 100
6.10 The figure shows the performance of the models considered in this work for $^{95}$Y. Fig. 6.10a refers to the ICM, Fig. 6.10b refers to the LCM, Fig. 6.10c refers to the SNN, and Fig. 6.10d refers to the DNN. . . . . 101
6.11 The figure shows a comparison between the performance of the various models when the antineutrino spectrum of $^{235}$U is obtained as the sum of the single isotope spectra predicted by the models. The antineutrino spectrum is multiplied by the IBD cross section to better put in evidence the energy interval of interest. Fig. 6.11a refers to the ICM, Fig. 6.11b refers to the LCM, Fig. 6.11c refers to the SNN, and Fig. 6.11d refers to the DNN. . . . . 106
6.12 The figure shows a comparison between the performance of the various models when the antineutrino spectrum of $^{238}$U is obtained as the sum of the single isotope spectra predicted by the models. The antineutrino spectrum is multiplied by the IBD cross section to better put in evidence the energy interval of interest. Fig. 6.12a refers to the ICM, Fig. 6.12b refers to the LCM, Fig. 6.12c refers to the SNN, and Fig. 6.12d refers to the DNN. . . . . 107
6.13 Block scheme representing the workflow defined in this work, and applied to the Pandemonium effect problem. . . . . 110

# List of Tables

1.1 Sensitivity of the main neutrino experiments to oscillation parameters. . . 11
1.2 Current best-fit values and uncertainties of neutrino oscillation parameters (three-flavor framework). Source: PDG [39]. . . . . 13

3.1 In the table are shown the main characteristics of HPGe and NaI(Tl) detectors, in the context of $\gamma$-ray spectroscopy for Pandemonium effect measurements. . . . . 37

5.1 The table summarizes the 52 isotopes for which a TAGS measurement has been taken, enabling comparison with HPGe measurements. These isotopes are the core element of this research. . . . . 65
5.2 The table summarizes the dimensions of the test, training and validation sets of the dataset of this work. Note that when PCA is applied: $N_Y = 60 \rightarrow N_{Y_{PCA}} = 15$. . . . . 79

6.1 Comparison of test metrics for each isotope and model. The ICM, LCM, SNN and DNN models are respectively represented in red, blue, green and orange. Values for each model are reported vertically. . . . . 102
6.2 The table shows the Global $\chi^2_{red}$ on the test set value for each model. In purple, the value closer to 1 value is obtained by ICM. . . . . 104
6.3 The table shows the mean MSE value on the test set for each model. In brown, the lower value is obtained by ICM. . . . . 105

# Acknowledgements

I wish to express my gratitude towards my advisor, Professor Antonio Cammi, and to Professor Monica Sisti who both gave me the opportunity to work in such an interesting and original topic, giving me their trust and autonomy as well as unspeakable help for my future in research.

A special thank you goes to Lorenzo Loi, without his guidance and tireless support, this work would not be the same. I also thank Dr. Matteo Borghesi and Stefano Riva for the valuable advice, the time they dedicated and tools they provided to me.

My heartfelt gratitude goes to my family, friends, and my loved one for the support and listening they gave throughout this journey. Your encouragement was essential.

Finally, a special thanks to Physics, for its unlimited capacity to amaze me during my life. Thanks for being the beautiful intellectual adventure I always wanted to engage.