



HAL
open science

The neutrino nature through the study of the Xenon 136 double-beta decays on the PandaX-III experiment

Andrii Lobasenko

► **To cite this version:**

Andrii Lobasenko. The neutrino nature through the study of the Xenon 136 double-beta decays on the PandaX-III experiment. High Energy Physics - Experiment [hep-ex]. Université Paris-Saclay, 2024. English. NNT : 2024UPASP051 . tel-04789094

HAL Id: tel-04789094

<https://theses.hal.science/tel-04789094v1>

Submitted on 18 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The neutrino nature through the study of the Xenon 136 double-beta decays on the PandaX-III experiment

*La nature du neutrino à travers l'étude des désintégrations
double-bêta du Xénon 136 avec l'expérience PandaX-III*

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°576, particules hadrons énergie et noyau:
instrumentation, imagerie, cosmos et simulation (PHENIICS)

Spécialité de doctorat: Physique des particules

Graduate School : Physique. Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche

Département de Physique Nucléaire (Université Paris-Saclay, CEA),
sous la direction de **Damien NEYRET** (Chercheur E6 (HDR), CEA Saclay)
et le co-encadrement de **Yann BEDFER** (Chercheur E6, CEA Saclay).

Thèse soutenue à Paris-Saclay, le 10 juillet 2024, par

Andrii LOBASENKO

Composition du jury

Membres du jury avec voix délibérative

Nicole d'HOSE Chercheuse, IRFU/DPhN, CEA Université Paris-Saclay	Présidente
Denis BERNARD Directeur de recherche, Laboratoire Leprince-Ringuet, École Polytechnique, Palaiseau	Rapporteur & Examineur
Luca SCOTTO LAVINA Chargé de recherche CNRS (HDR), LPNHE, Sorbonne Université	Rapporteur & Examineur
Dominique THERS Professeur de 1 ^{ière} classe, Subatech, IMT Atlantique	Examineur
Theopisti DAFNI Professeur, Département de physique théorique, Université de Saragosse, Espagne	Examinatrice
Tobias LIAUDAT Chercheur, IRFU/DEDIP, CEA Université Paris-Saclay	Examineur

Titre: La nature du neutrino à travers l'étude des désintégrations double-bêta du Xénon 136 avec l'expérience PandaX-III

Mots clés: Physique des particules, neutrino de Majorana, désintégration double-beta sans émission de neutrino, détecteurs Micromegas Microbulk, discrimination double-beta / gamma, Réseaux neuronaux convolutifs

Résumé: La recherche de la désintégration double bêta sans neutrino ($0\nu\beta\beta$) est cruciale pour faire progresser notre compréhension de la physique et explorer la physique au-delà du modèle standard. Cependant, cette recherche est incroyablement difficile en raison de l'extrême rareté de la désintégration, qui nécessite une interprétation approfondie et une dépendance aux contraintes expérimentales et aux modèles nucléaires théoriques. L'expérience PandaX-III est dédiée à la recherche de $0\nu\beta\beta$ dans ^{136}Xe . Il s'agit d'une chambre de projection temporelle (TPC) gazeuse à haute pression équipée de détecteurs Micromegas. Ce choix a été fait pour maximiser la capacité de détection des traces de particules et minimiser les fluctuations statistiques dans la résolution en énergie. L'un des principaux défis de la recherche d'événements $0\nu\beta\beta$ est la discrimination entre le signal et les événements de bruit de fond, qui contaminent la région d'intérêt (ROI). Le système de lecture par pistes des détecteurs Micromegas (une combinaison de 52 détecteurs forme un plan de lecture) permet la reconstruction 2D précise des trajectoires d'ionisation avec les informations de charge et de temps. Cela permet d'étudier l'énergie et la topologie des trajectoires d'électrons et, en conséquence, de distinguer le signal du bruit de fond. Pour réduire la diffusion latérale et obtenir des traces reconstruites plus fines, nous ajoutons un gaz quencher de 1% de triméthylamine (TMA) au gaz ^{136}Xe enrichi à 90%. La résolution en énergie actuelle de l'expérience PandaX-III est de 3% pour l'énergie de 2457 keV de la désintégration de ^{136}Xe $0\nu\beta\beta$ et devrait être améliorée à 1%. Cependant, plusieurs facteurs peuvent dégrader cette résolution en énergie, tels que la présence de canaux morts, les inhomogénéités de gain dans les détecteurs Micromegas ou l'attachement des électrons dans la TPC. Ce travail de doctorat présente une étude de l'impact des canaux manquants sur les reconstructions d'énergie et de topologie dans l'expérience PandaX-III. Les résultats de la détermination de la charge du blob n'offrent pas la possibilité souhaitée de reconstituer la partie de son énergie qui aurait été perdue en raison des canaux manquants dans XZ à partir des projections YZ des traces d'événements reconstruites et *vice versa*. Cependant, l'étude a montré qu'il est possible d'utiliser des algorithmes d'apprentissage automatique pour atténuer l'impact des canaux manquants sur la reconstruction de l'énergie et de la topologie. Un modèle de réseau neuronal convolutif (CNN) a été développé pour prédire l'énergie réelle des électrons à partir des données simulées collectées par les Micromegas avec des canaux manquants. Les résultats finaux montrent que le modèle CNN prédit l'énergie réelle des événements enregistrés par les Micromegas avec des canaux manquants avec une grande efficacité. Nous observons une amélioration de l'efficacité de détection du signal de Monte Carlo dans la ROI, qui passe de 69% à 89% après l'application du modèle CNN, par rapport à l'approche directe consistant à additionner les amplitudes des signaux provenant des Micromegas dont les canaux sont manquants. Un autre modèle CNN a également été utilisé pour classer les événements à deux électrons des événements à un seul électron dans les données de Monte Carlo affectées par des canaux manquants. Le modèle est capable, en présence de canaux manquants, de rejeter 99% des événements de bruit de fond tout en conservant une efficacité de 26% pour les signaux $0\nu\beta\beta$ dans la ROI. Les résultats de ce travail sont prometteurs et ouvrent la voie à d'autres études visant à améliorer la résolution en énergie et le rejet du bruit de fond dans l'expérience PandaX-III.

Title: The neutrino nature through the study of the Xenon 136 double-beta decays on the PandaX-III experiment

Keywords: Particle physics, Majorana neutrino, neutrinoless double-beta decays, Microbulk Micromegas detectors, double-beta / gamma discrimination, Convolutional Neural Networks

Abstract: The search for neutrinoless double-beta decay ($0\nu\beta\beta$) is crucial for advancing our understanding of physics and exploring physics beyond the Standard Model. However, this pursuit is incredibly challenging due to the decay's extreme rarity, requiring profound interpretation and reliance on experimental constraints and theoretical nuclear models. The PandaX-III experiment is dedicated to the search for $0\nu\beta\beta$ in ^{136}Xe . It is a high-pressure gaseous Time Projection Chamber (TPC) with Micromegas detectors. This design choice is made to maximize the particle track detection and discrimination $0\nu\beta\beta$ signal *vs.* gamma background capabilities. One of the main challenges of the $0\nu\beta\beta$ search is the discrimination between the signal and background events, which contaminate the region of interest (ROI). The strip readout system of the Micromegas detectors (a combination of 52 of them form a readout plane) allows for the precise 2D reconstruction of the ionization tracks together with the charge and time information. This allows for studying the electron tracks' energy and topology and ultimately discriminating the signal from the background. To reduce the lateral diffusion and get thinner reconstructed tracks, we add a 1% trimethylamine (TMA) quencher to the ^{136}Xe gas enriched to 90%. The current energy resolution of the PandaX-III experiment is 3% for the 2457 keV energy of the ^{136}Xe $0\nu\beta\beta$ decay, envisioned to be improved to 1%. However, several factors can degrade the energy resolution, such as the presence of dead channels, gain inhomogeneities in the Micromegas detectors, or electron attachment in the TPC.

This Ph.D work presents a study on the impact of missing channels on the energy and topology reconstructions in the PandaX-III experiment. The results of the Blob charge determination do not provide the desired possibility of reconstituting the part of the blob energy that would have been lost due to missing channels in XZ from YZ projections of reconstructed event tracks and *vice versa*. However, the study gave insight into employing machine learning (ML) algorithms to mitigate the impact of missing channels on energy and topology reconstructions. A Convolutional Neural Network (CNN) model was developed to predict the true energy of the electrons from the simulated data collected by the Micromegas with missing channels. The final results show that the CNN model predicts the true energy of the events recorded by the Micromegas with missing channels with a good energy resolution. We observe an improvement in the detection efficiency of the Monte Carlo $0\nu\beta\beta$ signal in the ROI from 69% to 89% after applying the CNN model, in comparison to the direct approach of directly summing amplitudes of the signals from the Micromegas with missing channels. Another CNN model was also used to classify the two-electron events from the single-electron events in the Monte Carlo data affected by missing channels. The model is capable of rejecting 99% of the background events while maintaining a 26% efficiency for the $0\nu\beta\beta$ signal in the ROI. The results of this work are promising and pave the way for further studies to improve the energy resolution and background rejection in the PandaX-III experiment.

*To my alma mater, the Faculty of Physics and Technology of the University of Kharkiv,
which was destroyed by russian forces in March 2022.
And to all the brave men and women in Ukraine fighting against russian aggression.*

Acknowledgements

I would like to express my sincere gratitude to Denis Bernard and Luca Scotto Lavina for their invaluable contributions as reviewers of this thesis. Their thoughtful comments and suggestions have been instrumental in shaping the final work. I am equally indebted to Theopisti Dafni, Dominique Thers, and Tobias Liaudat for their willingness to be my jury members and to evaluate my work. I would like to particularly appreciate Nicole d'Hose for accepting the position of president of the jury, as well as for her support and guidance during my PhD journey.

I owe an immeasurable debt of gratefulness to my supervisor, Damien Neyret, and my co-supervisor, Yann Bedfer, whose expertise, patience, and encouragement have been instrumental in this, without any doubt, endeavor. Their directions, insights, suggestions, and feedback have helped me make this thesis and ignited my passion for experimental research.

Working in the PandaX-III collaboration was a pleasure, and I am particularly grateful to Ke Han for the support and feedback during my PhD project.

My sincere thanks go to my colleagues at the DPhN department at CEA. Particularly to Hervé Moutarde and Francesco Bossu for all the support throughout these years. For all the fruitful discussions, I am thankful to Loïc, Eric, Benjamin, Valerie, Michael, Marine, Egle, Stephan, and again, Eric. Special thanks go to Isabelle for everything. I am also glad that I shared, in the meantime, a similar experience of working on the thesis with Andrea, Borana, Pierre, and Michael. Thank you all, especially Andrea Porro, for being a great friend of mine.

Also, I would like to thank my colleagues from NPAC, namely Lera, Juan, Yuya, Vidya, Lyosha, and Polina. I am happy I met all of you, and surely our roads will guide us only a little bit away from each other. I would remember time spent in your company with a smile, especially during the harsh COVID times, when the law felt expandable thanks to you.

Separate thanks go to my Ukrainian friends, without whom this thesis would be impossible and whose influence on my life in general was immense. Thanks to Lera for all our memories and conversations through the years. The Borscht parties were a blast. Thanks to Oleksiy for all the discussions about politics, weird sports, and critically acclaimed books. Thanks to Vsevolod for being an amazing friend and sharing a similar mindset to mine (I wonder why). Thanks to Yaroslav and Anya for all the great evenings after long days at work. Thanks to Karl and Dana for being close despite the distance that separates us. And thank to everyone else who I am grateful to share time through the past years: Anya, Andrii, Alex, Olya, Matthieu, Karyna, Kyrylo, Dasha, Valentyna, Oleksandra, Borys, Ira, Viktor, Sophia, Olena, and many others.

Last but not least, I would like to express immense gratitude to my family, who supported me from start to finish of my doctoral thesis and without whom I could not imagine becoming a doctor of philosophy in physics.

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 800945 — NUMERICS — H2020-MSCA-COFUND-2017.

Contents

List of Figures	3
List of Tables	15
Résumé étendu en français	19
Introduction	25
1 Neutrinoless Double Beta Decay: An Overview	27
1.1 Introduction	27
1.2 Theoretical Background	28
1.2.1 Neutrinos	28
1.2.2 Neutrino Oscillations and Neutrino Masses	29
1.2.3 Majorana Neutrinos	31
1.2.4 Neutrinoless Double Beta Decay	32
1.2.4.1 Decay image	32
1.2.4.2 Half-life time computation and Effective Majorana Mass	34
1.3 Conclusion	38
2 Neutrinoless Double-Beta Decay Experimental Setup	39
2.1 Introduction	39
2.2 Different Experimental Approaches	40
2.2.1 Experiments related to ^{76}Ge	40
2.2.2 Experiments related to ^{130}Te	41
2.2.3 Experiments related to ^{100}Mo	42
2.2.4 Experiments related to ^{136}Xe	42
2.3 PandaX-III Experiment	44
2.3.1 Jinping Underground Laboratory	45
2.3.2 High pressure Gaseous Xenon TPC Detector	45
2.3.3 Micromegas Readout plane	47
2.3.4 Signal reading and processing	50
2.3.4.1 DAQ system	50
2.3.4.2 Data processing	50
2.4 Limitations and Problematics	52
2.5 Conclusion	55

3	Topology of the Electron Tracks (Geant4)	57
3.1	Introduction	57
3.2	Low Energy Electrons in Gas	58
3.2.1	Updated Blob Charge Identification Process	62
3.2.1.1	Primary Blob charge identification principle based on the Maximum Energy Deposition hits	62
3.2.1.2	Blob charge identification principle based on the most ener- getic sphere of the defined radius	63
3.2.1.3	Identification of the Blob charge Barycenter	66
3.3	Electron Propagation Topology	70
3.4	Conclusions	76
4	Introduction to Machine Learning and Neural Networks	77
4.1	Introduction	77
4.2	Principles of Machine Learning	78
4.2.1	Loss and Metrics in Supervised Learning	79
4.2.2	Neural Networks and Deep Neural Networks	82
4.2.2.1	Gradient Descent and Backpropagation	85
4.2.2.2	More on activation functions	87
4.2.2.3	DNN training problems and optimizations	88
4.3	Convolutional Neural Networks	89
4.4	Conclusion	92
5	Regression problems	93
5.1	Introduction	94
5.2	Definitions and preliminary study	95
5.3	Data cleaning	98
5.4	Training the toyCNN on cleaned data	100
5.4.1	Global normalization of the input data images	100
5.4.1.1	Detected energies as labels	100
5.4.1.2	Primary energies as labels	101
5.4.2	Local normalization of the input images with E_{p} as labels	103
5.5	Model adjustments and improvements, data augmentation	104
5.5.1	Missing channels introduction and expanded dataset	104
5.5.2	Two-output model	106
5.5.3	Separation of the two outputs	110
5.5.4	Skip connections. Residual learning	110
5.5.4.1	MultiResNet model evaluation results	110
5.5.5	Different loss functions	112
5.5.5.1	Huber loss try	113
5.5.5.2	LogCosH loss try	113
5.6	Final model	114
5.6.1	Primary energy prediction evaluation	116
5.6.2	Primary energy prediction evaluation on the 2e dataset	117
5.6.3	Additional Multi3ResNet training on the 1e+2e dataset	118
5.6.4	Evaluation on the 0nbb dataset	120
5.7	Z position prediction	120

5.7.1	Evaluation of the model trained on the 1e dataset	122
5.7.2	Evaluation of the model trained on the 1e+2e dataset	122
5.8	Summary	124
5.9	Conclusions	124
6	Classification Problems	127
6.1	Introduction	127
6.2	CNN approach for the classification of 1e/2e events	128
6.2.1	Data Preparation	128
6.2.2	Model Architecture description	130
6.2.3	Hyperparameter Tuning and Training Process	132
6.3	Results from the Classification Model	134
6.4	Conclusions	139
7	Summary and Conclusions	143
A	PandaX-III Data Analysis Software Framework—REST	145
A.1	Introduction	145
A.2	Application scenarios	146
A.2.1	Data processing	146
A.2.2	Data visualization	147
A.2.3	Simulation program: restG4	147
A.3	Development logic	148
A.3.1	Key classes	148
A.3.2	Data structures	149
A.3.3	Running process	150
A.3.4	Configuration file - RML	151
A.3.5	Main functions	152
A.4	Processing chain for PandaX-III experiment	153
B	Backpropagation algorithm on a simple neural network	157
	Bibliography	161

List of Figures

1.1	<i>Left:</i> Experimental setup used by Reines and Cowan to detect neutrinos. <i>Right:</i> Example of a signal event, with the detection of two back-to-back gammas from the positron annihilation and of several other gammas from the neutron capture. Adapted from [8].	29
1.2	The normal and inverted hierarchies, along with the present status of masses and their mixing as obtained from the experiments on solar, atmospheric, reactor and accelerator neutrinos [20].	31
1.3	Double beta decay route map of ^{136}Xe . Since the $Z+1$ nucleus has a greater binding energy, it can only directly decay into the $Z+2$ nucleus, that is, ^{136}Ba . Sourced from [23].	33
1.4	Left: Feynman diagram of double beta decay. Right: Feynman diagram for neutrinoless double-beta decay (light neutrino exchange). Sourced from [23].	33
1.5	Diagram illustrating the energy spectrum of neutrinoless double-beta decay. The horizontal axis represents energy of electrons in keV, while the vertical axis denotes count in arbitrary units. Neutrino double-beta decay continuum is depicted in black, and a neutrinoless monoenergetic peak is highlighted in red. Relative sizes have been adjusted for clarity. It's worth noting that this peak exhibits some broadening due to two main factors: 1. The atomic nucleus carries a small amount of recoil energy, contributing to intrinsic broadening of electron energy. 2. Practical limitations in detector energy resolution. The nuclide featured in the diagram is ^{136}Xe , showcasing a monoenergetic peak situated near the Q value of 2458 keV.	34
1.6	Predictions on $\langle m_{\beta\beta} \rangle$ as a function of the lightest neutrino mass, considering both the NH and the IH. The bands arise from our lack of knowledge on the phase parameters (see Eq. 1.10). The shaded regions represent the 3σ regions resulting from the propagation of uncertainties related to oscillation parameters. This figure is sourced from Reference [28].	36

1.7	Distribution of decay half-lives, scaled by the square of the unknown parameter $\langle m_{\beta\beta} \rangle$ for $0\nu\beta\beta$ candidates as a function of mass number A . All the plotted results are obtained according to Eq. 1.9, using various nuclear models for the matrix element. Black circles, black bars - the shell model (SM), green squares - the interacting boson model (IBM-2), red bars, orange times signs, and magenta crosses - different versions of the quasiparticle random phase approximation (QRPA), and energy density functional theory (EDF), relativistic - downside cyan triangles, non-relativistic - blue triangles. Adapted from Ref. [26]. There exists a tendency for smaller decay half-lives in favor of the isotopes ^{96}Zr , ^{100}Mo , ^{118}Cd	37
2.1	Different experimental approaches for neutrinoless double beta decay.	40
2.2	Background model and data from the EXO-200 experiment. The $0\nu\beta\beta$ signal peak (purple, visible on the Phase I plot) is covered by the background events from ^{238}U and ^{232}Th decay chains. Plot adapted from [50].	44
2.3	Comparison of the m.w.e and relative excavated volume for different underground laboratories. The Jinping Underground Laboratory is the deepest laboratory in the world with a 6720 m.w.e shielding. Plot adapted from [53].	45
2.4	PandaX-III high pressure vessel design cross-section. A lower tank and an upper flange, fabricated through welding of cylindrical body, flange discs, and ellipsoidal heads. The cylindrical body has a 12 mm wall thickness, 2000 mm inner diameter, and stands about 1100 mm tall. Ellipsoidal head has an 18 mm wall thickness and a 2000 mm inner diameter. Flange discs, connecting the lower tank and the upper flange - 2180 mm in diameter and 164 mm thick. Weighing around 5 tons, the entire container is approximately 2300 mm tall. The lower tank includes four side flanges for various purposes, while the upper flange has 12 flanges for signal readout and 2 smaller flanges for air circulation. The field cage is built out of acrylic barrel, with an inner diameter of 1.6 meters and a height of 1.2 meters. The strip-shaped flexible PCBs form shaping rings interconnected by resistors, totaling 119 $G\Omega$ resistance. Sourced from [56].	46
2.5	(a) General Micromegas readout plane design. The Micromegas module (MM) consists of a metal mesh layer, a Kapton layer, and readout strips. Within a narrow gap between the metal mesh layer and the Kapton layer, electrons undergo amplification via a powerful electric field. Embedded within this gap are readout strips, which capture the charges. Adapted from [58]. (b) Micromegas readout strips visualization. The readout strips are squared copper pads connected in strip channels, 64 in X and 64 in Y directions. . .	47
2.6	Micromegas production process. (a) MicroBulk Micromegas production process. Sourced from [60]. (b) Thermal Bonding Micromegas production process. Adapted from [62]. (c) MicroBulk Micromegas tested at CEA Saclay. Sourced from internal communication. (d) Finished Thermal Bonding Micromegas. The Thermal Bonding Micromegas are employed in the PandaX-III experiment. Sourced from internal communication.	49
2.7	52 Micromegas arrangement and design on the readout plane suspension frame. Adapted from [63].	50

2.8	Pulses collected on each triggered strip (a) and projected tracks corresponding to XZ and YZ projections (b) of a cosmic-ray muon event detected. The color scale represents the amplitudes of the pulses. Adapted from [56].	51
2.9	Simulated $0\nu\beta\beta$ signal pulses after the detector response simulation (a) and corresponding XZ and YZ track projections after track processing (b). . . .	52
2.10	Gain inhomogeneities in the amplification region in one of the TBMMs. The color scale represents the gain of the amplification region. Sourced from [62].	52
2.11	Hit map from a calibration run showing a missing channel in the Microbulk Micromegas. We can observe one channel with no signal on the X axis and four channels with no signal on the Y axis. The color scale represents the hit rate for each position on the MM.	53
2.12	Comparison of the energy spectra of the $0\nu\beta\beta$ decay events with and without missing channels. ROI defined as 3σ around the Q-value of the decay ($\sim 3\%$ FWHM): [2364, 2552] keV	54
2.13	(a) NDBD event track simulated and reconstructed in REST software without missing channels. (b) Same NDBD event track simulated and reconstructed in REST software with missing channels. The area of the missing channels is highlighted in purple dashed rectangle on both plots. We can see that the missing channels lead to the separation of the track into two parts on the XZ projection. Thus, resulting in the misleading event track reconstruction - such topology information is useless for the analysis. On the YZ projection we can also observe that the black line representing the track has a connection at the end of the track to the point roughly in the middle of the track. This is due to the fact that the REST algorithm sometimes may have flaws that need to be calibrated.	54
3.1	$0\nu\beta\beta$ decay event simulated and reconstructed in REST. One of the blobs is falling on a missing channel (highlighted in purple) in the XZ projection. The corresponding undetected energy could be reconstituted by exploiting the correlation between XZ and YZ so that the overall energy is reconstructed within the ROI.	58
3.2	Comparison of relative depth dose distributions of protons and electrons in case of a single field. 110 MeV protons Bragg peak and 6 MeV electrons in water. Adapted from [75].	59
3.3	Visualization of the Bragg peak approximation on the XZ projection, knowing the deposited energy on the YZ projection.	60
3.4	Absolute difference between the Z coordinates of the Blob charges on the XZ and YZ projections.	61
3.5	(left) Correlation map of the Z coordinates of the first Blob charge on the XZ and YZ projections. (right) Correlation map of the Z coordinates of the second Blob charge on the XZ and YZ projections.	61
3.6	Energy correlation maps for the first and second Blob charges on the XZ and YZ projections. The Blob charge radius is 15 mm. The Blob charge Z coordinates on the XZ and YZ projections are within 5 mm from each other.	62

3.7	Step-by-step illustration of the Blob charge identification process on XZ projection for the NDBD event. (a) All the track hits on the XZ projection. (b) Selection of the track hits after the threshold for the energy deposition. (c) Found the most energetic hit, which coordinates are considered to be the Blob charge coordinates. The energy depositions of the track hits, within the Blob charge radius, are accumulated. (d) The track hits are separated into two groups: Blob charge hits and the tail hits. The process is repeated sequentially for the tail hits, until the tail is empty.	64
3.8	Step-by-step illustration of the Blob charge identification process on XZ projection for the NDBD event. (a) Selected track hits on the XZ projection above the energy deposition threshold. (b) For each of the selected hits, we accumulate the energy depositions of the track hits, within the Blob charge radius. (c) Comparing the energy depositions of the spheres, we select the sphere with the maximum energy deposition and consider the coordinates of the sphere center to be the Blob charge coordinates. (d) The track hits inside the sphere are being removed from the list of the track hits and the leftout track hits are being considered as a <i>tail</i> of the track. Again the process is repeated sequentially for the tail hits, until the tail is empty.	65
3.9	Example of the event with the kink in the track. The kink is identified as the Blob charge.	66
3.10	Comparison of the energy correlation maps for the Blob charges on the XZ and YZ projections. (a) and (c) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the maximum energy deposition hits. (b) and (d) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the most energetic sphere of the defined radius. The energy threshold for the track hits is 25 keV. The Blob charge radius is 15 mm for (a) and (b) and 10 mm for (c) and (d). . . .	67
3.11	Comparison of the energy correlation maps for the Blob charges on the XZ and YZ projections. (a) and (c) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the maximum energy deposition hits. (b) and (d) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the most energetic sphere of the defined radius. The Blob charge radius is 15 mm. The energy threshold for the track hits is 25 keV for (a) and (b) and 40 keV for (c) and (d). . . .	68
3.12	Visualization of the Barycenter approach for the Blob charge identification. After calculation of the Barycenter of the track hits within the Blob charge (a), the energy depositions of the track hits, within the Blob charge radius, centered at the Barycenter, are accumulated and now those considered to be the energy deposition of the Blob charge (b).	70

3.13	Comparison of the energy correlation maps for the Blob charges on the XZ and YZ projections after applying the Barycenter calculation for the Blob charge identification for both Maximum energy deposition and Maximum energy sphere approaches. (a) and (c) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the maximum energy deposition hits. (b) and (d) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the most energetic sphere of the defined radius. The energy threshold for the track hits is 25 keV. The Blob charge radius is 15 mm for (a) and (b) and 10 mm for (c) and (d). . . .	71
3.14	Visualization of the electron propagation track in Geant4. At each interaction point, the particle kinetic energy, energy deposition, step length, θ angle, kinetic energy of the secondary particle and it's type are being collected. The distance L to the end of the electron propagation track is calculated as well.	74
3.15	Energy loss of the electron, normalized by the number of electron events, with respect to the distance to the end of the electron propagation track. The rapid increase of the energy loss of the electron is observed at the distance of 5 mm to the end of the electron propagation track.	74
3.16	Stopping power of the electron with respect to the electron kinetic energy. . .	75
3.17	Angle θ between the electron momentum vectors before and after the interaction point, with respect to the distance to the end of the electron propagation track.	75
4.1	Example of the linear regression fit.	79
4.2	Example of the ROC curve. The dashed line represents the random classifier, while solid lines represent different classifiers. The most effective classifier is the blue one, which has the largest AUC-ROC closer to 1. Adapted from [96].	81
4.3	Node definition, where x_i are the inputs, ω_{ji} are the weights, ω_{j0} is the bias term, and $y_j = \sum_i^{n=3} \omega_{ji}x_i + \omega_{j0}$ is the output from the node.	82
4.4	Linear model with a single node.	83
4.5	Linear model with a single hidden layer.	83
4.6	Linear model with a single node and an activation function.	84
4.7	Overfitting on the training/validation loss plot. In an ideal scenario, both the training and validation loss curves should go down as the model learns. The training loss should eventually flatten out, indicating the model has learned the training data. The validation loss, representing performance on new data, should also decrease but ideally stay close to the training loss, showing the model is learning generally applicable patterns and not just memorizing the training data. In this example, the model starts overtraining after the 500th epoch, where the validation loss starts increasing. Sourced from [104]. . . .	89
4.8	2D-convolution operation. A kernel is convoluted with the input image, producing a feature map. A convolutional layer usually consists of multiple kernels, each producing a separate feature map. The values in the kernel are trainable parameters. Sourced from [107]	90

4.9	Simple Feedforward CNN architecture. To the left is an example of an input image with the size of $128 \times 128 \times 1$. The image is passed through the convolutional layer (yellow) with 32 kernels, followed by the activation function (orange) and the pooling layer (red), which reduces the spatial dimensions of the input by a factor of 2. This combination of layers is called a Conv1 block and outputs 32 feature maps of size 64×64 . The next block, Conv2 , has 64 kernels and outputs 64 feature maps of size 32×32 . The flattened layer (purple) reshapes the 2D array into a 1D array, which is then passed to the fully connected layer (light green) with 128 neurons and further applications of the activation function (dark green).	91
4.10	ResNet architecture with one skip connection. A skip connection clones the output of the first convolutional block and performs an additional convolutional operation on it together with pooling. The output of the skip connection is added to the output of the third convolutional block. Further operations are the same as in the example of the CNN (Fig. 4.9).	91
5.1	Example of an input image (128×128) without missing channels. The left part of the image corresponds to the XZ projection of the track, and the right part - to the YZ projection. The amplitude of the pixels is the ADC count, divided by amplification factor. The sum of all amplitudes is the total deposited energy of the track event.	96
5.2	$E_{\mathfrak{P}}$ and $E_{\mathfrak{D}}$ distributions before, (a), and after, (b), the "one Micromegas Module selection".	97
5.3	Preliminary toyCNN architecture. The input to the network is a 2D image of the track XZ and YZ projections. No missing channels are used. The output is the $E_{\mathfrak{P}}$ of the event.	97
5.4	(a) Comparison of the true value ($E_{\mathfrak{P}}$) and predicted value ($\hat{E}_{\mathfrak{P}}$) of the Primary Energy. (b) Correlation plot of $\hat{E}_{\mathfrak{P}}$ vs. $E_{\mathfrak{P}}$	99
5.5	Residual distribution between $E_{\mathfrak{D}}$ and $E_{\mathfrak{P}}$. The 3σ range determines the events that are considered for the training.	99
5.6	$E_{\mathfrak{P}}$ and $E_{\mathfrak{D}}$ flat distribution.	100
5.7	Train and validation loss for the toyCNN training with $E_{\mathfrak{D}}$ considered as labels for the cleaned 80k dataset.	101
5.8	Correlation plot between $E_{\mathfrak{D}}$ and $\hat{E}_{\mathfrak{D}}$ from the toyCNN evaluation on the cleaned 80k dataset. Global normalization of the input images.	101
5.9	Evaluation of the toyCNN trained on the cleaned 80k dataset. Global normalization of the input images. (a) Correlation plot for the residual between $E_{\mathfrak{D}}$ and $\hat{E}_{\mathfrak{D}}$ with respect to true labels - $E_{\mathfrak{D}}$ in this training session. (b) Residual distribution between $E_{\mathfrak{D}}$ and $\hat{E}_{\mathfrak{D}}$	102
5.10	Train and validation loss for the toyCNN training with $E_{\mathfrak{P}}$ as labels from the cleaned dataset. Global normalization of the input data events is applied. . .	102
5.11	Evaluation of the toyCNN trained on the cleaned 80k dataset. Global normalization of the input images. (a) Correlation plot for the residual between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ with respect to $E_{\mathfrak{P}}$. (b) Residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$	103
5.12	Train and validation loss for the toyCNN training with $E_{\mathfrak{P}}$ as labels. Input data events are normalized independently - local normalization.	103

5.13	Evaluation of the toyCNN trained on the cleaned 80k dataset with local normalization of the input images. (a) Correlation plot for the residual between the $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ with respect to the $E_{\mathfrak{P}}$. (b) Residual distribution between the $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$	104
5.14	(a) Example of the input array with missing channels. (b) Example of the missing channels map, corresponding to the input array. (c) Concatenated map and input arrays, of a $[2, 128, 128]$ size.	105
5.15	MultiCNN architecture, with two outputs.	106
5.16	MultiCNN evaluation. Trained on the 300k dataset with global normalization of input images. (a) Correlation plot for the residual between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$. (b) Residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$	107
5.17	$\hat{E}_{\mathfrak{P}}$ distribution in the ROI from MultiCNN. The efficiency of events' prediction in ROI is 79%.	108
5.18	MultiCNN evaluation. Trained on the 300k dataset with global normalization of input images. (a) Correlation plot for the residual between $E_{\mathfrak{D}}$ and $\hat{E}_{\mathfrak{D}}$. (b) Residual distribution between $E_{\mathfrak{D}}$ and $\hat{E}_{\mathfrak{D}}$	109
5.19	MultiCNN evaluation. $\hat{E}_{\mathfrak{D}}$ distribution corresponding to $E_{\mathfrak{P}}$ in the ± 15 MeV band at $Q_{\beta\beta}$	109
5.20	Two outputs separation in the MultiCNN model.	110
5.21	From the MultiCNN model with <i>separated</i> outputs evaluation. Correlation plot for the residual between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ with respect to $E_{\mathfrak{P}}$. The presence of the artifacts is reduced in comparison to the previous model results for the primary energies (Fig. 5.16a).	111
5.22	Skip connection introduced in the MultiResNet architecture. Here an output from the first convolutional block is used as an input for the skip connection convolutional block. The output of the skip connection is then added to the output of the second convolutional block. Further propagation of the data through the model is unchanged: the summed output is flattened and passed through the fully connected layers.	111
5.23	MultiResNet model evaluation. Trained on the 300k dataset with global normalization of input images. (a) Correlation plot for the residual between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ with respect to $E_{\mathfrak{P}}$. (b) Residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$	112
5.24	Efficiency of the events' prediction in the ROI for the MultiResNet model. (a) $\hat{E}_{\mathfrak{P}}$ in ROI. The efficiency of events' prediction in the ROI is 84%. (b) Efficiency estimation for the direct sum of amplitudes from images with missing channels (Detection efficiency is 69%). In this way we can compare the improvement the model brings to the energy reconstruction.	112
5.25	Comparison of the MSE (Squared Error), MAE (Absolute Error), LogCosH loss and Huber loss for $\delta = 1&5$ functions. X-axis represents "error" of prediction, while Y-axis represents the loss value. Sourced from CS4780/CS5780: Machine Learning [FALL 2015] course at Cornell University.	114
5.26	Multi3ResNet architecture. Color scheme: yellow - convolutional layers with 64 filters each, orange - ReLu activation function, red - Average Pooling layers with 2×2 kernel (separate <i>skipPool</i> Average Pooling layer with 4×4 kernel), purple - flatten layer, light green - fully connected layers, dark green - ReLU activation function.	116

5.27	Multi3ResNet evaluation. Trained on the 300k dataset with global normalization of input images. (a) Correlation plot for the residual between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ with respect to $E_{\mathfrak{P}}$. (b) Residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$	116
5.28	Efficiency of the events' prediction in the ROI for the Multi3ResNet model. $\hat{E}_{\mathfrak{P}}$ in ROI. The efficiency of events' prediction in the ROI is 89%.	117
5.29	The energy distributions corresponding to the primary energies of each of the electrons in the dataset containing 2e events. The dataset is used only for evaluation purposes.	117
5.30	Multi3ResNet trained on the 1e dataset. Plots represent the evaluation on the 2e dataset. (a) Correlation plot for the residual between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ with respect to $E_{\mathfrak{P}}$. (b) Residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$	118
5.31	1e and 2e energy spectra in the combined 1e+2e dataset. The 1e energy spectrum is flat, while the 2e energy spectrum is a sum of two primary energies, depicted in Fig. 5.29.	119
5.32	Multi3ResNet trained on 1e+2e dataset evaluation. (a) Correlation plot for the residual between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ with respect to $E_{\mathfrak{P}}$. (b) Residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$	119
5.33	Efficiency of the events' prediction in the ROI for the Multi3ResNet trained on 1e+2e dataset. $\hat{E}_{\mathfrak{P}}$ in ROI. The efficiency of events' prediction in the ROI is 89%.	119
5.34	Residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ from the Multi3ResNet model evaluation on the $0\nu\beta\beta$ dataset. The model is trained on the 1e+2e dataset. . .	120
5.35	Efficiency of the events' prediction in the ROI for the Multi3ResNet trained on 1e+2e dataset. (a) $\hat{E}_{\mathfrak{P}}$ of the $0\nu\beta\beta$ events. Efficiency of events' prediction in the ROI is 90%. (b) Efficiency of the direct extraction from the sum of amplitudes.	121
5.36	Multi3ResNet evaluation results on the $0\nu\beta\beta$ dataset without missing channels. (a) Residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$. (b) $\hat{E}_{\mathfrak{P}}$ with the mean shift correction. The efficiency of events' prediction in the ROI is 33% and 96% for the uncorrected and corrected mean shift, respectively.	121
5.37	\hat{Z}_0 evaluation for the Multi3ResNet model trained on the 1e dataset. (a) Correlation plot for the residual between Z_0 and \hat{Z}_0 with respect to Z_0 . (b) Residual distribution between Z_0 and \hat{Z}_0	122
5.38	\hat{Z}_0 evaluation for the Multi3ResNet model trained on the 1e dataset. Evaluation on the 2e dataset. (a) Correlation plot for the residual between Z_0 coordinate and \hat{Z}_0 coordinate with respect to Z_0 . (b) Residual distribution between Z_0 and \hat{Z}_0	123
5.39	\hat{Z}_0 evaluation for the Multi3ResNet model trained on the 1e+2e dataset. Evaluation on the 1e+2e testing dataset. (a) Correlation plot for the residual between Z_0 and \hat{Z}_0 with respect to Z_0 . (b) Residual distribution between Z_0 and \hat{Z}_0	123
5.40	\hat{Z}_0 evaluation for the Multi3ResNet model trained on the 1e+2e dataset. Evaluation on the $0\nu\beta\beta$ dataset. (a) Correlation plot for the residual between Z_0 and \hat{Z}_0 with respect to Z_0 . (b) Residual distribution between Z_0 and \hat{Z}_0 . .	124

6.1	Results taken from [73] study on the event discrimination. Plots show the energy distribution of the $0\nu\beta\beta$ (a) and high-energy background (b) events before and after <i>track</i> , <i>Q</i> , <i>twist</i> , and <i>length</i> cutoffs. Such approach reduces the background contamination in the ROI by the factor of ~ 100 , while keeping the $0\nu\beta\beta$ signal efficiency at 17.5%.	128
6.2	Primary energy spectrum of the 1-electron and 2-electron events in the combined 1e+2e dataset. Distribution of primary energies is flat for both types of events and lays in the [30 keV, 4 MeV] energy range. The number of entries in the dataset is $1.2 \cdot 10^6$ equally distributed between the 1-electron and 2-electron events.	129
6.3	Comparison of the 1-electron (a) and 2-electron (b) events input images with similar primary energies ~ 2.4 MeV. The 2-electron event is more complex in terms of the track topology, visibly has a Blob charge at each end of the track, and the track length is significantly shorter than the 1-electron event.	130
6.4	Class3ResNet model visualization. The color scheme is the same as in the previous chapter with yellow representing the convolutional layers, orange - ReLU activation functions, red - average pooling layers, purple - flattening layer, green - dense layers with following ReLU activation functions marked as dark green. A skip connection is added after the first average pooling layer, which contains another convolutional block with average pooling layer. The output from the main path and the skip connection is concatenated before flattening the output and passing it through a set of dense layers. The final output layer contains two outputs for the classification task.	132
6.5	Loss curve (a) and accuracy curve (b) for the classification model during the training process. The model is trained consecutively on 10 smaller datasets of the full training dataset. Training iteration on each subset consists of 100 epochs. We observe a rapid overfitting of the model closer to the end of each training iteration, however, the model's performance improves over time. The model's weights from the third epoch of the last training iteration are selected as the final model and used for evaluation on the testing set.	134
6.6	(a) Probability distributions for the 1-electron and 2-electron events from the Class3ResNet in the 1e2e_test dataset. We observe a high contamination of the False Positives classified as 2-electron events. (b) ROC curve for the classification model. The AUC value is 0.9572.	135
6.7	(a) Signal Efficiency and Background Rejection Efficiency curves for the Class3ResNet model as a function of the probability threshold value. (b) The Background Rejection Efficiency vs Signal Efficiency curve for Class3ResNet evaluation on the 1e2e_test dataset.	136
6.8	(a) Probability distributions for the 1-electron and 2-electron events from the Class3ResNet in the 1e2e_test dataset selected in the ROI. The high contamination of the False Positives classified as 2-electron events is still visible. (b) ROC curve for the classification model. The AUC value is 0.9679.	136

6.9	(a) Signal Efficiency and Background Rejection Efficiency curves for the Class3ResNet model as a function of the probability threshold value. For events selected in the ROI. (b) The Background Rejection Efficiency vs Signal Efficiency curve for Class3ResNet evaluation on the 1e2e_test dataset for events selected in the ROI.	137
6.10	Energy distributions of the events selected in the ROI before (blue) and after (red) the model's rejection at the threshold value of 0.95 that corresponds to the Background Rejection Efficiency of 99% and Signal Efficiency of 26%. (a) Primary energy distribution for the 2-electron events. (b) Primary energy distribution for the 1-electron events. (c) Total sum of the degraded amplitudes for the 2-electron events. (d) Total sum of the degraded amplitudes for the 1-electron events.	138
6.11	Energy distributions of the events selected in the ROI before (blue) and after (red) the model's rejection at the threshold value of 0.95 that corresponds to the Background Rejection Efficiency of 99% and Signal Efficiency of 20%. (a) Primary energy distribution for the $0\nu\beta\beta$ events. (b) Total sum of the degraded amplitudes for the $0\nu\beta\beta$ events.	138
6.12	Plots represent primary energy of the weakest electron in the 2-electron events for the 1e2e_test dataset (a) and the $0\nu\beta\beta$ dataset (b). The blue line shows the primary energy before the model's discrimination, while the red histogram shows what is left after the model's rejection. Below each plot, the ratio of the accepted events with respect to the weakest electron's primary energy is shown.	140
6.13	The reconstructed energy spectra of signal and backgrounds before and after the optimal cut for the $0\nu\beta\beta$ events(left) and background events from ^{214}Bi decay(right) from the work of Hao et al. [114].	141
A.1	The entire data processing workflow, from simulation to reconstruction, in the context of studying algorithms for neutrinoless double-beta decay, sourced from [66].	146
A.2	REST browser interface. (right) Visualization of a specific instance in the waveform instance stage. (left) Visualization of the instance with added channel range plotting conditions. The images are from [66], where the data is from a run of the TREX [127] prototype detector.	147
A.3	Employing restG4 for simulating the initial energy deposition results in the detector. Left: Schematic representation of the Baby-IAXO detector's geometry [128]. Right: Depiction of energy deposition from a neutron event in the detector, with white lines illustrating a sequence of secondary particles. Image sourced from [66].	148
A.4	Schematic diagram of the contents of a REST file. Items in red font are mandatory. Image is from [66].	149

A.5	A schematic representation illustrating the flow of event data within a REST processing chain. The run object is initialized, providing access to specific metadata or event data present in the input REST file, along with any additional objects outlined through RML. Subsequently, the data undergoes processing via the implementation within the process runner object. Various event types (A, B, C, D) refer to distinct implementations for specific events. The resulting output REST file consolidates all metadata information accessible to the chain, including any pre-existing data, featuring the transformed output for the specific event and the updated analysis tree. Image is from [66].	151
A.6	REST program execution framework. The two primary executables, rest-Manager and restRoot, are highlighted in red. The rest-config shell script is highlighted in blue. Image is from [66].	153
B.1	Linear model with two hidden layers.	157

List of Tables

1.1	Current target isotopes in prominent $0\nu\beta\beta$ -decay experiments along with their respective characteristics: Daughter Nuclei, $Q_{\beta\beta}$, Natural Abundance, and $2\nu\beta\beta$ -decay half-life values. The provided $2\nu\beta\beta$ -decay half-life values represent the utmost precision found in the literature. Adapted from [1].	35
2.1	Representative experiments of each nuclide and the neutrinoless double-beta decay half-life and effective Majorana mass limits.	43
3.1	Results from the updated Blob charge identification process. The average of the mean energies of the Blob charges for the XZ and YZ projections, obtained using the Maximum Energy Deposition hits and the Most Energetic Sphere approaches (directly and after applying the Barycenter recalculation). The table shows the results for different Blob charge radii R and energy thresholds for the track hits definition.	69
5.1	Preliminary toyCNN. Attribute for convolutional layers stands for the kernel shape and the number of kernels, while for pooling layers it is kernel shape. N_p is the number of parameters in the layer. For the convolutional layers, N_p is calculated as $(F * F * C * K) + (K \text{ bias})$, where F is the kernel size, C is the number of channels in the input, and K is the number of kernels. For the dense layers, N_p is calculated as $(N * M) + (M \text{ bias})$, where N is the number of neurons in the previous layer and M is the number of neurons in the current layer. The input to the network is a 2D image of the track. The model is trained to predict the primary energy $E_{\mathfrak{P}}$	98
5.2	MultiCNN with two outputs. Attribute for convolutional layers stands for the kernel shape and the number of kernels, while for pooling layers it is kernel shape. N_p stands for the number of parameters in the layer.	107
5.3	Multi3ResNet architecture. Attribute for convolutional layers stands for the kernel shape and the number of kernels, while for pooling layers it is kernel shape. N_p stands for the number of parameters in the layer. The Skip_AveragePooling2D layer is the additional Average Pooling layer that is applied to the skip connection.	115

5.4 Comparison of the model’s performance in predicting the primary energies and. The training datasets and the test fractions for the model evaluation are provided. The last two columns correspond to the efficiency of the events’ prediction in the ROI and the standard deviation of the residual distribution between the true and predicted primary energies. The toyCNN was trained and evaluated only on the perfect readout configuration without missing channels in this study. The table also includes the evaluation of the Multi3ResNet model on the $0\nu\beta\beta$ dataset with and without missing channels. The drop in the efficiency of the events’ prediction for the $0\nu\beta\beta$ dataset without missing channels is due to the mean shift in the residual distribution between $E_{\mathfrak{N}}$ and $\hat{E}_{\mathfrak{N}}$ which has to be investigated. 125

6.1 Classification model layer description. Attribute for convolutional layers stands for the kernel shape and the number of kernels, while for pooling layers it is kernel shape. N_p is the number of parameters in the layer. For the convolutional layers, N_p is calculated as $(F * F * C * K) + (K \text{ bias})$, where F is the kernel size, C is the number of channels in the input, and K is the number of kernels. For the dense layers, N_p is calculated as $(N * M) + (M \text{ bias})$, where N is the number of neurons in the previous layer and M is the number of neurons in the current layer. 131

6.2 Best hyperparameters obtained from the hyperparameter tuning process. For the main branch of the model, the average pooling kernel is 4×4 . For the skip connection, the average pooling kernel is 16×16 to match the dimensions of the output from the main branch before concatenation. 133

6.3 Signal Efficiency and Background Rejection Efficiency for the $1e2e_test$, $1e2e_test_ROI$, and $0\nu\beta\beta$ datasets. 139

Résumé étendu en français

Les neutrinos, d'abord hypothésés par Wolfgang Pauli en 1930, restent l'une des particules les plus énigmatiques du Modèle Standard de la physique des particules. Ils sont électriquement neutres, interagissent faiblement. Ils ont une masse (très faible), comme en témoignent les oscillations de saveur des neutrinos - un phénomène où les neutrinos oscillent entre différents états de saveur (électron, muon, tau) lorsqu'ils se déplacent dans l'espace. Nous obtenons, à partir des expériences d'oscillation, des informations sur les masses des différents états de masse qui nous laissent deux possibilités pour les hiérarchies de masse: Normale (NH, $m_1 < m_2 < m_3$) et Inversée (IH, $m_3 < m_1 < m_2$). On peut noter que dans ces conditions, l'échelle de masse absolue reste indéterminée et qu'il en va de même pour l'indice de l'état de masse de plus faible masse. Une autre question qui reste ouverte est la nature du neutrino, s'ils sont des particules de Dirac ou de Majorana. Par conséquent, nous sommes incités à explorer les propriétés des neutrinos de manière approfondie.

Les neutrinos sont produits dans divers processus, comprenant, entre autres, les réactions nucléaires dans le Soleil, les réacteurs nucléaires et les interactions des rayons cosmiques avec l'atmosphère terrestre. Malgré leur abondance, les neutrinos sont difficiles à étudier en raison de leur faible interaction avec la matière. L'une des méthodes les plus prometteuses pour comprendre les propriétés des neutrinos est la détection de la double désintégration bêta sans émission de neutrinos ($0\nu\beta\beta$). Ce processus nucléaire hypothétique n'est possible que si les neutrinos sont leurs propres antiparticules (particules de Majorana). De plus, si elle est observée, la désintégration $0\nu\beta\beta$ pourrait fournir des informations sur l'échelle de masse absolue des neutrinos. Dans le Modèle Standard, la double désintégration bêta avec émission de deux neutrinos ($2\nu\beta\beta$) est autorisée et extrêmement rare, avec des demi-vies observées dépassant 10^{19} ans. La valeur de $Q_{\beta\beta}$ c'est-à-dire l'énergie libérée lors de la désintégration, est partagée entre les particules chargées et neutres. Par conséquent, le spectre énergétique mesuré est continu, similaire à celui de la désintégration β .

D'un autre côté, la $0\nu\beta\beta$ implique l'émission de seulement deux électrons et aucun neutrino, ce qui entraîne un pic à la valeur $Q_{\beta\beta}$ dans le spectre énergétique. La mise en évidence de ce pic serait une preuve directe de la nature Majorana des neutrinos. La demi-vie de la $0\nu\beta\beta$ est estimée à plus de 10^{25} ans, et pour l'observer, il serait nécessaire d'avoir une exposition très sensible (masse \times temps), une réjection efficace des bruits de fond et une bonne résolution énergétique.

Cette thèse est dédiée à l'expérience PandaX-III, qui vise à rechercher la double désintégration bêta sans émission de neutrinos ($0\nu\beta\beta$) dans le ^{136}Xe . On utilise une chambre à projection temporelle (TPC) à haute pression remplie de gaz de xénon enrichi et d'un gaz quencher de 1% de triméthylamine (TMA). Le ^{136}Xe est un gaz noble avec une grande abondance naturelle, ce qui en fait un bon candidat pour les recherches de $0\nu\beta\beta$, avec la possibilité de dimensionner une expérience à une grande masse. La valeur $Q_{\beta\beta}$ du ^{136}Xe est de 2457 keV,

ce qui est relativement faible par rapport à d'autres isotopes utilisés dans les recherches de $0\nu\beta\beta$. La région d'intérêt (ROI), que nous définissons autour de la valeur $Q_{\beta\beta}$ pour rechercher le signal de $0\nu\beta\beta$, est plus susceptible d'être contaminée par le bruit de fond gamma provenant de la radioactivité naturelle.

Pour détecter l'ensemble de l'ionisation et mesurer l'énergie des électrons, l'expérience PandaX-III utilise des modules Micromegas (MM). Les spécificités du MM pour le traçage nous permettent de détecter des traces de particules chargées afin d'utiliser la topologie de l'événement pour discriminer le bruit de fond gamma. Dans le contexte de PandaX-III, la lecture des MM est constituée de pixels avec un pas de 3 mm, interconnectés en canaux dans deux directions orthogonales. Chaque module possède 128 canaux, 64 dans les directions X et 64 dans les directions Y. Ainsi, une telle configuration de lecture fournit des projections XZ et YZ des traces. Les informations sur la topologie des traces permettent de distinguer le signal $0\nu\beta\beta$ des événements de bruit de fond. Un exemple serait la présence de deux charges Blob (correspondant au pic de Bragg de chacune des traces d'électrons) pour le signal $0\nu\beta\beta$, tandis que les événements de bruit de fond n'auraient qu'une seule charge Blob. L'expérience vise à atteindre une résolution énergétique de 3% à 2457 keV, avec des efforts pour améliorer cette résolution à 1%. Cependant, plusieurs facteurs peuvent dégrader la résolution énergétique. L'effet d'attachement des électrons dans le gaz entraîne une perte de dépôt d'énergie, tandis que les inhomogénéités du gain de lecture dans le MM peuvent entraîner une dégradation de la résolution énergétique. De plus, en raison de défauts de construction ou de courants sombres élevés, certains canaux du MM peuvent être déconnectés, ce qui peut entraîner une perte d'informations sur les traces et le dépôt d'énergie, entraînant une dégradation de la résolution énergétique et, par conséquent, de la sensibilité de détection de $0\nu\beta\beta$. Cette thèse se concentre sur l'analyse de ce dernier problème et le développement de différentes approches pour atténuer (ou être robuste face) aux effets des canaux manquants dans les modules Micromegas.

Dans le cadre de l'expérience PandaX-III, la perte de trois canaux par module Micromegas (MM) (52 MM sur le plan de lecture) est le maximum acceptable. Les modules avec plus de trois canaux manquants avant l'assemblage du plan de lecture ne sont pas utilisés dans l'expérience. Cependant, même trois canaux manquants par MM peuvent entraîner une perte de 22% du dépôt d'énergie total. Cette perte peut avoir un impact significatif sur la résolution en énergie. De plus, plusieurs canaux manquants consécutifs peuvent conduire à une perte d'informations sur la topologie des traces, ce qui est crucial pour le rejet du bruit de fond. La reconstruction de l'énergie et le rejet du bruit de fond peuvent être gravement affectés si un dépôt de charge Blob est perdu en raison des canaux manquants. Ainsi, la première partie de cette thèse est consacrée à l'étude analytique de la reconstruction des charges Blob manquantes sur l'une des projections de traces. Elle est suivie par une approche plus générale utilisant l'apprentissage automatique pour la reconstruction de l'énergie des événements dégradés par les canaux manquants. La dernière partie de la thèse est dédiée à l'approche de l'apprentissage automatique pour la discrimination entre le signal $0\nu\beta\beta$ et les événements de bruit de fond, basée sur les informations de topologie des traces en présence de canaux manquants.

Topologie des traces d'électrons et analyse des charges Blob

Cette partie de la thèse présente l'étude de la corrélation énergétique des charges Blob entre les projections XZ et YZ des traces issues de la simulation Monte-Carlo des événements

$0\nu\beta\beta$ dans l'expérience PandaX-III. La première partie de cette étude a été réalisée avec le logiciel REST, où le processus `TRestFindTrackBlobsProcess` a été utilisé pour identifier les charges Blob aux deux extrémités de chaque projection de trace. La corrélation énergétique des charges Blob s'est avérée faible et dépendante du paramètre de rayon de la charge Blob. Par conséquent, un autre processus a été développé pour mieux déterminer les charges Blob et améliorer la corrélation entre les deux projections. Ce processus se concentre sur l'identification du hit (objet dans REST représentant une amplitude à un instant donné sur chaque canal) sur les projections avec le dépôt d'énergie maximal dans la trace et l'accumulation des dépôts d'énergie des hits dans un rayon défini autour de ce hit pour déterminer la charge Blob. Cependant, cette méthode est limitée par sa dépendance au hit le plus énergétique, qui ne correspond pas toujours au véritable pic de Bragg de la trace d'électron.

Pour résoudre ce problème, la seconde approche prend en compte tous les hits de la trace et calcule le dépôt d'énergie total dans un rayon sphérique défini autour de chaque hit. Le hit dont la sphère environnante a le dépôt d'énergie maximal est sélectionné comme centre de charge Blob. Les hits utilisés pour la détermination de la charge Blob sont ensuite exclus, et le processus est répété pour les hits restants.

Les deux méthodes ont été affinées en recalculant les coordonnées de la charge Blob en utilisant le barycentre, ce qui permet d'obtenir une distribution d'énergie plus précise en considérant le centre de masse des hits à l'intérieur du rayon de la charge Blob. Cet ajustement vise à réduire la dispersion des énergies des charges Blob observée dans les méthodes précédentes.

L'étude explore également le rayon optimal de la charge Blob et le seuil de dépôt d'énergie pour les hits de trace, en reconnaissant la sensibilité de ces paramètres pour l'identification correcte des charges Blob. Une simulation détaillée utilisant l'outil Geant4 a été réalisée pour analyser la propagation des électrons dans un mélange gazeux dans les conditions de l'expérience PandaX-III. Cette simulation est utilisée pour déterminer à quel endroit le long de la trace les électrons entrent dans la phase de charge Blob, caractérisée par un dépôt d'énergie accru.

L'analyse montre que des diffusions multiples significatives affectent la trace même avant que l'électron n'atteigne le pic de Bragg, compliquant ainsi l'identification des charges Blob. Malgré cela, les résultats suggèrent que les charges Blob émergent généralement dans les 20 derniers mm de la trace de propagation de l'électron, avec des énergies attendues variant de 200 à 300 keV. L'approche directe pour reconstruire les charges Blob perdues ou les corrélérer avec la torsion de la trace est moins réussie que prévu, ce qui souligne l'importance d'étudier les techniques d'apprentissage automatique pour les travaux futurs.

Approche et Méthodologie en Apprentissage Automatique

L'objectif principal de la thèse est le développement d'une méthode basée sur l'apprentissage automatique (ML) pour atténuer les effets des canaux manquants dans les détecteurs Micromegas. Dans cette partie du travail, nous nous concentrons sur l'application des réseaux de neurones convolutifs (CNN) pour reconstruire l'énergie primaire des événements de particules dans une chambre à projection temporelle (TPC) dans des conditions où chaque module MM contient trois canaux manquants. L'objectif crucial est d'améliorer la précision de la reconstruction énergétique en utilisant la topologie de la trace et les amplitudes des signaux. L'étude explore le potentiel des CNN pour prédire l'énergie primaire à partir de

données simulées, démontrant l'efficacité de cette approche pour traiter des schémas de données complexes et des conditions difficiles.

Nous commençons par utiliser un modèle simplifié, "toyCNN", pour évaluer le potentiel des CNN à prédire l'énergie primaire à partir de données simulées. Par convention, une combinaison de couches convolutives, d'activation ReLU et de couches de regroupement est appelée un bloc convolutif. Ce modèle contient deux blocs convolutifs, suivis d'une couche entièrement connectée et d'un seul nœud de sortie pour la prédiction de l'énergie. Cette étude initiale exclut les pathologies complexes du détecteur pour simplifier le problème, en se concentrant sur des événements avec un seul électron. Les données d'entrée, représentant la topologie de la trace, sont normalisées pour faciliter un apprentissage efficace du modèle CNN. Le "toyCNN" démontre la capacité d'approximatif de l'énergie primaire, bien que limité, notamment dans les régions avec peu de données d'entraînement.

En s'appuyant sur ces résultats, nous explorons diverses techniques de normalisation et optimisations de modèle. Un modèle plus sophistiqué, "MultiCNN", est introduit, lequel gère l'apprentissage multitâche en prédisant à la fois les énergies primaire et détectée. Ce modèle incorpore des couches entièrement connectées supplémentaires pour mieux capturer les caractéristiques sous-jacentes des événements. L'étude montre des résultats prometteurs, avec le modèle réussissant à prédire l'énergie primaire sur une large gamme d'énergies, même en présence de défis de données significatifs. Le modèle MultiCNN montre une efficacité de prédiction de 79% et un RMS de 71 keV pour les résidus entre les énergies primaires prédites et réelles.

Un modèle MultiCNN, avec des couches séparées pour des sorties distinctes, a été testé mais a encore montré un décalage moyen notable dans les prédictions. Pour remédier à cela, une architecture semblable à un réseau résiduel (ResNet) a été introduite, incorporant des connexions de contournement pour maintenir le flux de gradient et améliorer l'apprentissage de la représentation. Un nouveau modèle appelé MultiResNet a été développé, comprenant trois blocs convolutifs (sans couches de regroupement incluses), une connexion de contournement entre le deuxième bloc convolutif et la couche d'aplatissement, et une couche entièrement connectée pour chaque sortie. Ce modèle MultiResNet réduit considérablement le RMS des résidus à 62 keV et améliore l'efficacité à 84%.

Pour améliorer davantage la robustesse du modèle, différentes fonctions de perte telles que Huber et LogCosH ont été explorées pour atténuer l'impact des valeurs aberrantes, qui faussent souvent les résultats. Finalement, la fonction de perte LogCosH, combinée avec l'optimiseur AdamW, a été choisie pour sa stabilité et sa sensibilité réduite aux valeurs aberrantes. Cette configuration a conduit au développement du modèle Multi3ResNet, qui comprend trois blocs convolutifs avec des couches de regroupement moyennes incluses pour mieux capturer et conserver les caractéristiques essentielles des données d'entrée. Le modèle dispose maintenant de deux couches entièrement connectées pour chaque sortie. De plus, le modèle Multi3ResNet est entraîné pour prédire la coordonnée Z du dépôt d'énergie primaire, laquelle est inconnue lors de l'acquisition des données. La prédiction de la coordonnée Z est intéressante pour le rejet de bruit de fond, car elle permet d'appliquer une réjection spatiale aux événements de bruit de fond qui ont été générés par des rayons gamma provenant par exemple du plan de lecture. L'entraînement sur un grand ensemble de données (300 000 points de données) montre la performance du modèle, avec une efficacité de prédiction de 89% pour les énergies primaires et un RMS amélioré de 54 keV. Cependant, des défis ont persisté lors de l'application du modèle à des données avec deux électrons primaires

(événements 2e), soulignant la sensibilité du modèle à la topologie des données. L'étude se termine par un entraînement supplémentaire du modèle sur un ensemble combiné 1e+2e avec une évaluation supplémentaire sur le sous-ensemble 1e+2e et l'ensemble de données $0\nu\beta\beta$, où le modèle maintient une haute efficacité et précision, marquant une amélioration significative par rapport aux méthodes précédentes et augmentant la sensibilité potentielle dans les recherches $0\nu\beta\beta$. L'efficacité de prédiction $0\nu\beta\beta$ au sein de la ROI est augmentée de 69% à 90% par rapport à la somme directe des amplitudes de signal, avec un RMS des résidus réduit à 54 keV. De plus, le modèle prédit la coordonnée Z du dépôt d'énergie primaire, maintenant un petit RMS pour le résidu entre la coordonnée Z prédite et vraie de 6,4 cm pour les événements $0\nu\beta\beta$, en maintenant une performance similaire pour l'ensemble de données 1e+2e.

Analyse de Classification des Événements

Dans la dernière partie de cette thèse, l'accent est mis sur la classification des événements de particules plutôt que sur la reconstruction de l'énergie à l'aide des modèles de réseaux de neurones convolutifs (CNN). L'objectif est désormais de mieux discriminer entre les différents types d'événements. La partie précédente a démontré l'efficacité des modèles CNN pour la reconstruction de l'énergie des particules. Cependant, l'objectif actuel est de distinguer les événements à 1 électron des événements à 2 électrons, en particulier pour traiter la contamination élevée de bruit de fond dans la région d'intérêt (ROI) de l'expérience PandaX-III.

Un modèle CNN est utilisé pour cette tâche de classification, en tirant parti des caractéristiques topologiques des traces de particules. Cette approche contraste avec une méthode précédente de Javier Galan, qui utilisait une extraction de caractéristiques analytiques pour réduire significativement la contamination de fond, bien que avec une efficacité de signal limitée. L'approche de Hao Qiao utilisait un CNN pour distinguer entre les événements de $0\nu\beta\beta$ et les événements de fond à haute énergie, obtenant des mesures de performance notables. Le travail actuel vise à améliorer ces méthodes en affinant l'architecture du modèle et la préparation des données.

La préparation des données pour le modèle de classification reflète celle du modèle de reconstruction de l'énergie, avec une attention particulière à l'équilibrage du jeu de données entre les événements à 1 électron et à 2 électrons. Cela est crucial car les caractéristiques topologiques des traces, qui sont liées aux énergies des particules, doivent être représentées avec précision pour éviter les biais. Les données simulées provenant de REST sont utilisées, avec des spectres d'énergie allant de 30 keV à 4 MeV. Le jeu de données est équilibré avec environ 50% de chaque type d'événement, y compris les événements à faible énergie, pour garantir un apprentissage complet.

Le modèle de classification, nommé Class3ResNet, utilise une architecture modifiée par rapport au modèle de reconstruction de l'énergie précédent. Il comprend trois blocs convolutifs, une connexion de contournement intégrant des blocs convolutifs supplémentaires, et trois couches entièrement connectées optimisées par l'Optimisation Bayésienne. L'entraînement du modèle implique un grand jeu de données (1,2 million de points de données) divisé en plus petits lots pour gérer les contraintes de mémoire GPU, et le réglage des hyperparamètres est effectué à l'aide de la bibliothèque Hyperopt.

Un accomplissement significatif est la capacité du modèle CNN à classer les événements à deux électrons par rapport aux événements à un électron avec une grande précision. Le

modèle, Class3ResNet, atteint une précision de classification de 92%, avec une Efficacité du Signal de 31,7% et une Efficacité de Rejet du Fond de 99% pour distinguer entre les événements à 1 électron et à 2 électrons. Le modèle montre une dégradation de l'Efficacité du Signal pour les événements dans la ROI sur le jeu de données $1e+2e$, atteignant 26% pour la même Efficacité de Rejet du Fond. De plus, on observe que le modèle donne une Efficacité du Signal de 20% pour les événements $0\nu\beta\beta$. Certaines limitations sont observées, en particulier avec les événements impliquant des électrons à faible énergie, ce qui rend leur classification précise plus difficile.

Les résultats suggèrent que bien que le modèle soit performant dans l'ensemble, il existe des nuances dans la classification des événements liées aux niveaux d'énergie, ce qui impacte l'efficacité de détection du signal pour différents types d'événements. Cela souligne la nécessité d'un raffinement continu pour équilibrer la détection des signaux avec le rejet du fond dans les applications futures.

Perspectives et conclusion

La thèse se conclut par une discussion des implications des résultats et des pistes de travail futures. Bien que prometteurs, les modèles de ML actuels nécessitent une validation supplémentaire avec des données réelles provenant de l'expérience PandaX-III. Les recherches futures pourraient intégrer des données provenant de plusieurs modules Microegas et traiter d'autres sources de dégradation des données, telles que les inhomogénéités du gain de lecture. L'affinement des modèles CNN avec un ensemble de données d'entraînement plus diversifié pourrait améliorer la robustesse et la précision des reconstructions d'énergie et des classifications d'événements.

Nous démontrons la faisabilité et le potentiel de l'utilisation de techniques avancées de ML, spécifiquement des CNN, pour aborder des problèmes complexes en physique des particules expérimentale. Les méthodes développées améliorent significativement la reconstruction d'énergie et la classification des événements pour les recherches de $0\nu\beta\beta$ dans l'expérience PandaX-III, ouvrant la voie à des mesures plus sensibles et plus précises dans les expériences futures. Le travail souligne l'importance des approches interdisciplinaires, combinant la physique et l'apprentissage automatique, pour relever les défis des configurations expérimentales modernes.

Introduction

Neutrinos, whose existence was first postulated by Wolfgang Pauli in 1930, are one of the most mysterious particles in the Standard Model of particle physics. They are electrically neutral, weakly interacting, and have a very small mass. We know that neutrinos have mass from the observation of neutrino flavor oscillations. This phenomenon occurs when neutrinos produced in one flavor state (electron, muon, or tau) change into another flavor state as they propagate through space. The problem is that the Standard Model of particle physics assumes that neutrinos are massless particles. Therefore, the study of neutrino properties is crucial for understanding the fundamental laws of nature, expanding our knowledge of the universe, and answering some of the most profound questions in physics.

Neutrinos are produced in a variety of processes, such as nuclear reactions in the Sun, nuclear reactors, and cosmic rays interacting with the Earth's atmosphere. Neutrinos are also produced in the Earth's interior and emitted by radioactive elements decay. It makes them an omnipresent particle in the universe, and they are the most abundant particles after photons. However, due to their properties, it is not easy to characterize them. A lot of fundamental questions about neutrinos remain unanswered, such as the absolute neutrino mass scale, the nature of neutrinos (whether they are Dirac or Majorana particles), the origin of the matter-antimatter asymmetry in the universe, etc. To answer the former questions, we need to measure the neutrino mass scale, and the most sensitive method to do this is the study of neutrinoless double beta decay ($0\nu\beta\beta$). This process is a hypothetical nuclear decay in which two neutrons in a nucleus simultaneously decay into two protons, two electrons without neutrino emission. A two neutrino double beta decay ($2\nu\beta\beta$) is an allowed process in the Standard Model, and it is the rarest nuclear decay ever observed with a half-life greater than 10^{19} years. To observe $0\nu\beta\beta$, whose half-life is expected to be greater than 10^{25} years, we need to conduct highly precise measurements using increasingly large experimental setups and more effective background noise rejection techniques.

This thesis is dedicated to studying the $0\nu\beta\beta$ in the PANDAX-III experiment. The aim of this experiment is to search for $0\nu\beta\beta$ in ^{136}Xe using a high-pressure gaseous time projection chamber (TPC) filled with enriched Xenon gas. The $0\nu\beta\beta$ is identified by detecting the energy deposited by the electrons emitted in the TPC. Micromegas detectors, which are capable of measuring the position and energy of the electron tracks, are used in the experiment. The PandaX-III experiment is expected to have a 3% energy resolution for 2457 keV energy of the ^{136}Xe $0\nu\beta\beta$ decay, with an ongoing effort to improve it to 1%. However, the energy resolution can be degraded by several factors, either pertaining to the micromegas detectors, such as disconnected channels or gain inhomogeneities, or to the working of the TPC itself, such as electron attachment. The main goal of this thesis is to develop a method, based on Machine Learning (ML), to hinder these factors. In the work presented here, I will focus on missing channels. Nevertheless, the method developed can be generalized to also

tackle any of the aforementioned degradation factors. Furthermore, I will extend my ML approach to address the problem of rejecting background events mimicking the signature of the $0\nu\beta\beta$ signal.

In the first chapter, the neutrinoless double beta decay is introduced, and the importance of the study of this process is discussed. The second chapter presents different experimental techniques used to search for $0\nu\beta\beta$, and the PANDAX-III experiment is described in detail. The effect of missing channels on the energy and topology reconstructions is also discussed. The third chapter is dedicated to the analysis of the electron track topology in the PandaX-III experiment context from Monte Carlo simulations. In this chapter, attention is paid to the determination of the Blob charge of an electron track - one of the most essential characteristics of the track. The Blob charge is related to the Bragg peak of the electron track and is used in the discrimination of the background events from the $0\nu\beta\beta$ signal. The Blob charge may not be correctly measured if there are missing channels. Thus, knowing the Blob charge is crucial for energy reconstruction and particle identification. In the fourth chapter, the Machine Learning algorithms are introduced with a focus on the Convolutional Neural Networks (CNN) that are used in this work. The fifth chapter presents the method developed to reconstruct the true energy of the electrons from the simulated data collected by the Micromegas with missing channels. A detailed methodology of CNN implementation is presented, and the results of the model predictions are discussed. Finally, in the last chapter, a CNN model is used to classify the two-electron events vs the single-electron events from the Monte Carlo data affected by the missing channels.

Chapter 1

Neutrinoless Double Beta Decay: An Overview

... I shall face a battle I know not,
I shall ride a road I know not...
— *The Epic of Gilgamesh*
~2000 BC

1.1	Introduction	27
1.2	Theoretical Background	28
1.2.1	Neutrinos	28
1.2.2	Neutrino Oscillations and Neutrino Masses	29
1.2.3	Majorana Neutrinos	31
1.2.4	Neutrinoless Double Beta Decay	32
1.3	Conclusion	38

1.1 Introduction

Neutrinoless double beta decay ($0\nu\beta\beta$) represents a theoretical decay mechanism in which two neutrons within a nucleus transform into two protons, emitting two electrons without neutrinos. This process, if observed, would provide direct evidence that neutrinos are Majorana particles, *i.e.* fermions that are their own antiparticles. Importantly, $0\nu\beta\beta$ also defies the conservation of lepton numbers, as the emitted electrons are unaccompanied by their corresponding electron anti-neutrinos.

The existence of Majorana neutrinos has the potential to unlock key insights into explaining the intriguing matter-antimatter asymmetry observed in our universe. However, it's important to note that the physical phenomenon of $0\nu\beta\beta$ lies beyond the scope of the Standard Model (SM) of particle physics, which currently governs the behavior of subatomic particles.

While double beta decay processes involving the emission of two electron anti-neutrinos have been observed in several isotopes, the elusive $0\nu\beta\beta$ process remains a subject of on-

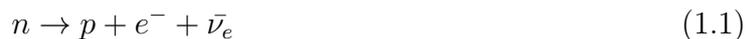
going investigation. To delve into the most recent and comprehensive compilation of both theoretical and experimental struggles in this field, readers are encouraged to explore review articles, such as [1].

1.2 Theoretical Background

1.2.1 Neutrinos

Where it all began?

In 1930, Wolfgang Pauli introduced the concept of neutrinos as a mean to explain the continuous energy distribution of electron observed in beta decay. In the process of decay, alongside the emission of electrons, the atomic nucleus releases an electrically neutral particle - difficult to detect - that carries away a random portion of the energy. At first Pauli called this particle a neutron, but later Enrico Fermi proposed to name this particle "neutrino" (*Italian: Little neutral one*) at the Solvay Physics Conference in 1933 [2]. Therefore, the basic model of beta decay was established:



In 1937, Ettore Majorana, a student of Fermi, postulated the intriguing idea that neutrinos could potentially be their own antiparticles [3], which would be referred to as Majorana fermions. In 1939, based on Majorana's theory, W. Furry proposed the hypothesis of neutrinoless double-beta decay [4]. Double-beta decay, a second-order process of beta decay, entails the simultaneous emission of two electrons and two anti-electron neutrinos. If the neutrino indeed behaves as its own antiparticle, there exists a probability that the two neutrinos annihilate each other during the decay process. Such scenario results in the absence of emitted neutrinos, which is referred to as neutrinoless double-beta decay.

An observation of muons in cosmic rays in 1936 by Carl Anderson and Seth Neddermeyer revealed the existence of a new generation of leptons [5]. The discovery of muon decay followed in 1941, with the initial measurement of its decay spectrum occurring in 1948. It was discovered that the energy spectrum of the decay product electrons was also a continuous spectrum, prompting the introduction of the second-generation neutrinos known as muon neutrinos.

In terms of experiments, from 1942 onward, people mastered the technology of manufacturing reactors capable of generating substantial quantities of neutrinos (antielectron neutrinos) with MeV energies. Such technological progress became an important prerequisite for the direct detection of neutrinos. Bruno Pontecorvo was the first to propose the idea of detecting neutrinos in 1946 involving the absorption of neutrinos by ^{37}Cl , leading to the production of ^{37}Ar , which decays producing Auger electrons. The notation of this process is as follows: $\nu_e + ^{37}\text{Cl} \rightarrow ^{37}\text{Ar} + e^{-}$. Generally, in the early 1950s, researchers conducted a series of experiments based on this idea to try to find neutrinos, yet they were unsuccessful [6]. The problem was that the neutrinos emitted by the reactors were antielectron neutrinos, while the ^{37}Cl experiment could only detect electron neutrinos. At the time, there was no distinction between the two types of neutrinos.

Despite the initial setbacks, the idea of detecting neutrinos was not abandoned. Clyde Cowan and Frederick Reines proposed a new experiment, called "Project Poltergeist" which

employs anti-beta decay to detect antielectron neutrinos emitted from the reactor [7]. The anti-beta decay process $\bar{\nu}_e + p \rightarrow n + e^+$ involves a proton capturing an antielectron neutrino, resulting in the production of a neutron and a positron. The positron annihilates with an electron, producing two gamma rays with an energy of 0.511 MeV each that work as a marker of the neutrino capture. The experimental setup featured a structure consisting of two tanks of 100 liters of water each, sandwiched between three layers of scintillator, as one can see on Fig. 1.1(left). The setup was placed near the reactor core at a distance of around 20 meters, which simplified the detection process. It must be noted that cadmium chloride was added to the water in order to absorb neutrons. The Cadmium-108 in the solution absorbs neutrons, transforming into Cadmium-109, which decays by emitting gamma rays, detected by scintillators: $^{108}\text{Cd} + n \rightarrow ^{109m}\text{Cd} \rightarrow ^{109}\text{Cd} + 2\gamma$ as seen on Fig. 1.1(right). Detecting

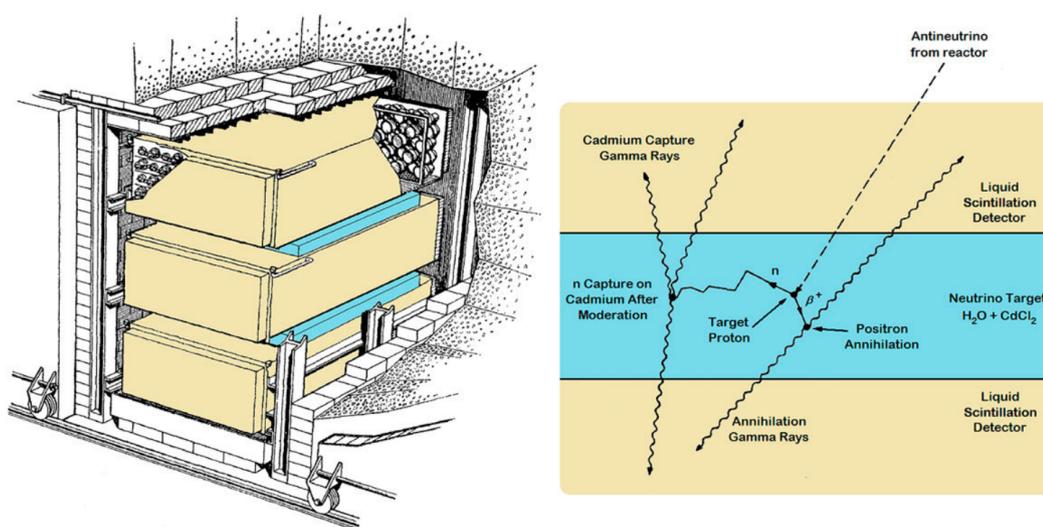


Figure 1.1: *Left:* Experimental setup used by Reines and Cowan to detect neutrinos. *Right:* Example of a signal event, with the detection of two back-to-back gammas from the positron annihilation and of several other gammas from the neutron capture. Adapted from [8].

the neutron absorption signal, which is delayed by several microseconds with respect to the positron annihilation, further helps the identification of the neutrino. Finally, they observed different signal rates under the operating conditions of turning the reactor on/off, with a ratio of more than 5 times, proving the existence of neutrinos.

1.2.2 Neutrino Oscillations and Neutrino Masses

In 1957, Bruno Pontecorvo proposed the idea of neutrino oscillations [9], which was later developed by Ziro Maki, Masami Nakagawa and Shoichi Sakata in 1962 [10]. The theory of neutrino oscillations is based on the assumption that neutrinos have non-zero masses. This proposition by Pontecorvo later provided a basis to explain the solar neutrino problem.

The Homestake experiment, conducted by Raymond Davis Jr. in 1968, was the first experiment to measure the flux of electron neutrinos originating from the Sun's core. The experiment was based on the detection of neutrinos via the inverse beta decay process: $\nu_e + ^{37}\text{Cl} \rightarrow ^{37}\text{Ar} + e^-$. Surprisingly, the measured value fell between one third and one half of the anticipated value by the Standard Solar Model (SSM) [11]. This discrepancy between

the measured and predicted values gave rise to a persistent challenge known as the "Solar Neutrino Problem". This enigma persisted for approximately three decades, during which thorough investigations scrutinized both the experiment and the solar model for potential issues, yet none were uncovered. Eventually, it was discerned that both the experiment and the model were fundamentally accurate, and that the discrepancy arose from the fact that neutrinos exhibit greater complexity than initially assumed.

In the case that the three neutrino flavors (ν_e , ν_μ , ν_τ) have non-zero masses, different from each other, the neutrino flavor oscillation is allowed. Therefore, neutrinos produced in the Sun's core as electron neutrinos can oscillate into undetectable flavors during their passage to Earth. With this in mind, many experiments were conducted to measure the flux of neutrinos from the Sun, including the Kamiokande experiment in 1987 [12], the GALLEX experiment in 1991 [13], the Super-Kamiokande experiment in 1996 [14] and the SNO experiment in 2000 [15]. Takaaki Kajita of Super-Kamiokande and A.B. McDonald of SNO were awarded the Nobel Prize in Physics in 2015 for the discovery of neutrino oscillations [16].

Neutrino flavor oscillation requires that the mass eigenstates of neutrinos are not flavor eigenstates. Since a neutrino is always produced in a flavor eigenstate, this flavor eigenstate wave function will be a mixture of the 3 mass eigenstates. The mixing of eigenstates is described by the Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix [10, 17], denoted as U . In the absence of CP violation and no Majorana neutrinos, the transformation relationship between the two eigenstates is expressed as follows:

$$\begin{aligned} \begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} &= U \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} = \begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu1} & U_{\mu2} & U_{\mu3} \\ U_{\tau1} & U_{\tau2} & U_{\tau3} \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} = \\ &= \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13} & c_{12}c_{23} - s_{12}s_{23}s_{13} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13} & -c_{12}s_{23} - s_{12}c_{23}s_{13} & c_{23}c_{13} \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} \end{aligned} \quad (1.2)$$

$$(1.3)$$

where c and s are the abbreviations of $\cos \Theta$ and $\sin \Theta$ respectively; 12, 13 and 23 respectively represent three mixing angles: Θ_{12} , Θ_{13} and Θ_{23} .

Let's assume that neutrinos propagate in vacuum with a propagation distance L and energy E . As it propagates, it decomposes into three mass eigenstates, each oscillating in the form of a plane wave. Therefore, the transition probability of the neutrino flavor α to the neutrino flavor β is expressed as follows:

$$P(\nu_\alpha \rightarrow \nu_\beta) = \left| \sum_{i=1}^3 U_{\alpha i}^* U_{\beta i} e^{-i \frac{m_i^2 L}{2E}} \right|^2 \quad (1.4)$$

where $U_{\alpha i}$ is the element of the PMNS matrix, m_i is the mass of the i -th mass eigenstate. The absolute values of mass eigenstates are currently unknown; however, the mass-squared differences between them are known. The difference between m_1 and m_2 can be derived from the solar neutrino oscillation experiments, yielding a minuscule value. Similarly, the difference between m_2 and m_3 is obtained from the atmospheric neutrino oscillation experiments, resulting in a significantly larger value [18].

The mass-squared differences between the three mass eigenstates are expressed as follows:

$$\Delta m_{12}^2 = m_2^2 - m_1^2 = (7.53 \pm 0.18) \times 10^{-5} \text{ eV}^2 \quad (1.5)$$

$$\Delta |m_{32}^2| \equiv |m_3^2 - m_2^2| \approx |m_{31}^2| = (2.44 \pm 0.06) \times 10^{-3} \text{ eV}^2 \quad (1.6)$$

Two potential configurations exist for the three mass eigenstates: $(m_1 < m_2 \ll m_3)$, called Normal Hierarchy (NH) ordering; and $(m_3 \ll m_1 < m_2)$, referred to as the Inverted Hierarchy (IH) ordering, as illustrated in Fig. 1.2. In the NH, the two lightest mass eigenstates have a small mass difference, around 10 meV, with the third eigenstate being approximately 50 meV heavier. In the IH the lightest mass eigenstate is followed by a pair of higher mass eigenstates that are about 50 meV heavier, with the mass difference within the pair being roughly 10 meV. The specific mass ordering of neutrinos remains undetermined, although recent experiments lean toward a Normal Hierarchy [19].

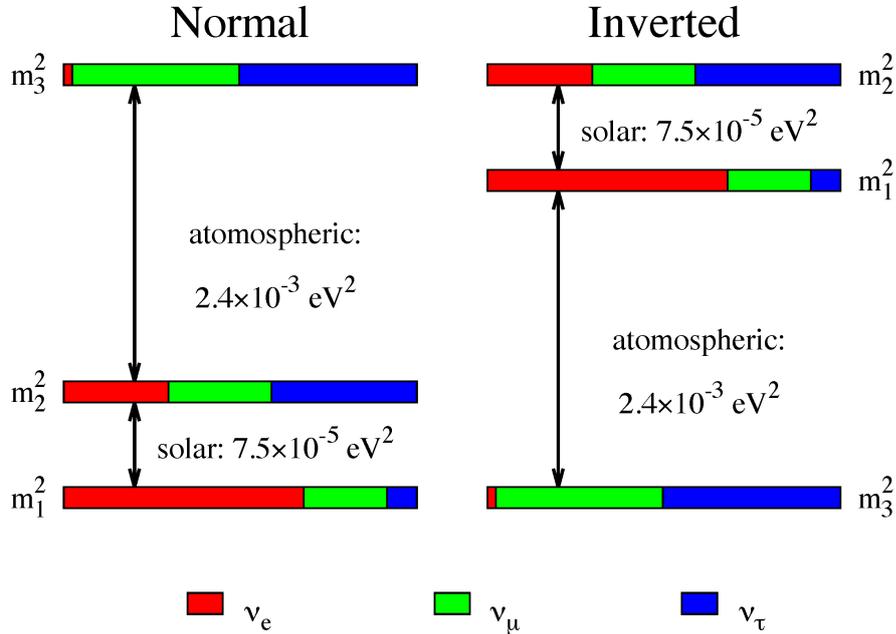


Figure 1.2: The normal and inverted hierarchies, along with the present status of masses and their mixing as obtained from the experiments on solar, atmospheric, reactor and accelerator neutrinos [20].

1.2.3 Majorana Neutrinos

To explain the origin of neutrino masses, the Standard Model (SM) of particle physics must be extended. One method is to introduce a new right-handed neutrino field through the Higgs mechanism, which is referred to as the seesaw mechanism, allowing the neutrino to a Dirac mass. Unfortunately, this approach has flaws. On one hand, we never observed right-handed neutrinos, and on the other hand, it is hard to explain the remarkably small mass of neutrinos as well as why the coupling constant is small (on the order of 10^{-12}) [21] compared to other elementary particles that interact with the Higgs field. An extension of the theory involving the right-handed neutrino field is the incorporation of the Majorana property. Assuming that both the left and right-handed neutrino fields adhere to the

Majorana equation, they can couple with their respective charge conjugate fields, leading to the acquisition of Majorana masses. Notably, this Majorana mass is distinct from the earlier discussed Dirac mass. The actual neutrino mass results from the superposition of the Majorana mass and the Dirac mass.

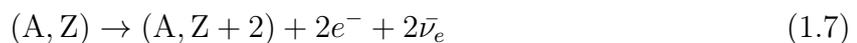
It's noteworthy that the assertion "neutrinos are their own antiparticles" doesn't contradict the earlier statement that "the Cl-37 experiment and the anti-beta decay experiment can only detect neutrinos and antineutrinos, respectively." What these two experiments actually differentiate is the left-handed helicity state and the right-handed helicity state of neutrinos. The default two-component theory of the Standard Model categorizes them as positive and negative neutrinos. However, introducing Majorana's model allows these two helicity states to originate from the same particle. In reality, a small fraction of neutrinos of opposite helicity state may be entangled in the neutrino beam [22]. Assuming a reactor releases a batch of neutrinos, the mixing ratio of their left and right-hand fields, as per the Dirac equation under the small mass approximation, is m/E , where m is the neutrino mass (below the eV level) and the energy E is in the MeV range. The proportion of anomalous helical states can be computed as m/E^2 , roughly on the order of 10^{-12} [23]. If the anomalous helical state of reactor neutrinos is observed through experiments like ^{37}Cl to confirm that neutrinos are Majorana particles, the occurrence of an event might take nearly 10 million years, making it nearly impossible [24].

1.2.4 Neutrinoless Double Beta Decay

1.2.4.1 Decay image

The most effective method for evidencing Majorana neutrinos is through the exploration of neutrinoless double-beta decay. Neutrinoless double-beta decay represents a unique instance within the broader category of double-beta decay. Double-beta decay is a second-order process stemming from general beta decay. In beta decay, a neutron within the nucleus transforms into a proton, resulting in the transformation of the nucleus from (A, Z) to $(A, Z+1)$. This reaction necessitates the release of energy, making it imperative for the binding energy of the final-state atomic nucleus to be lower and more stable than the initial state, thus facilitating beta decay. The imbalance between the binding energies of the initial and final nuclei is referred to as the Q value.

Within the nucleus, the intrinsic spin of protons and neutrons causes them to form pairs with opposite spin directions. Consequently, nuclei comprising an even number of neutrons or protons exhibit greater stability compared to those with an odd number. The quintessence of stability is an atomic nucleus featuring an even number of neutrons and protons, with all nucleons paired. For such even-even nuclei, general beta decay would transform them into odd-odd nuclei, leading to an increase in binding energy, rendering it an improbable process, as depicted in Fig. 1.3. In reality, these nuclei can only undergo double-beta decay, transitioning two nuclear charges at a time, namely:



Double-beta decay was first discovered on ^{82}Se nuclei [25]. Discovered in over a dozen even-even nuclei, neutrinoless double-beta decay exhibits a half-life in the range of 10^{19} to 10^{21} years. If neutrinos are Majorana particles, there exists the possibility of neutrinoless

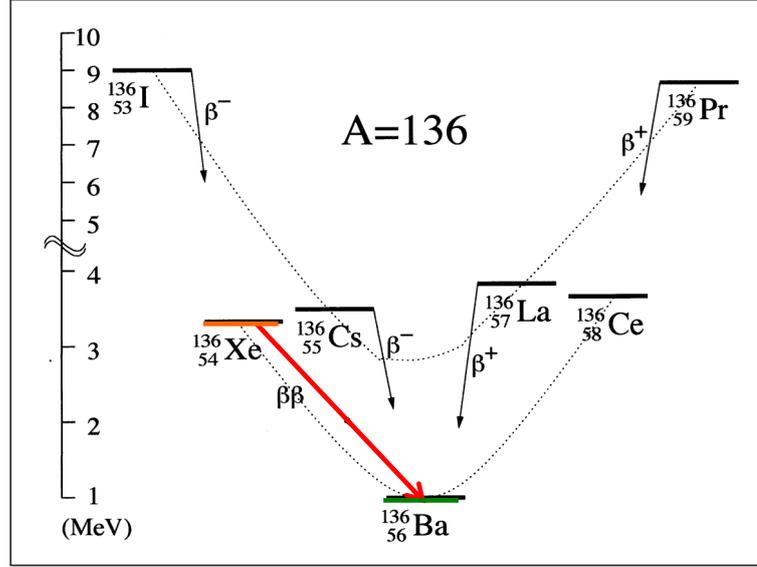


Figure 1.3: Double beta decay route map of ^{136}Xe . Since the $Z+1$ nucleus has a greater binding energy, it can only directly decay into the $Z+2$ nucleus, that is, ^{136}Ba . Sourced from [23].

double-beta decay based on the process of double beta decay, as illustrated in Fig. 1.4. In this scenario, the two neutrinos annihilate each other, resulting in no neutrino emission in the final state, represented by:

$$(A, Z) \rightarrow (A, Z + 2) + 2e^- \quad (1.8)$$

In this scenario, as there are no neutrinos to carry away energy, the two electrons will

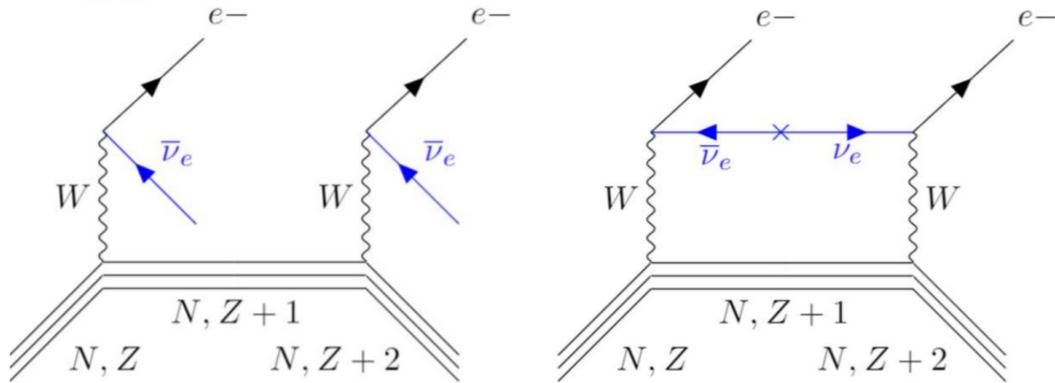


Figure 1.4: Left: Feynman diagram of double beta decay. Right: Feynman diagram for neutrinoless double-beta decay (light neutrino exchange). Sourced from [23].

bear almost all the decay energy (given the much greater mass of the daughter nucleus in comparison to two electrons, the nuclear recoil energy is negligible, i.e., for ^{136}Xe the nuclear recoil energy of ^{136}Ba daughter nucleus is of the factor of dozens of eV), generating monoenergetic peaks near the Q value in the energy spectrum. Given this characteristic, experiments can be designed to detect double-beta decay in various nuclides, searching for single-energy peaks at the end of the continuum spectrum. This approach allows the

identification of cases involving neutrinoless double-beta decay, as depicted in Fig. 1.5. In comparison to double-beta decay involving neutrinos, neutrinoless double-beta decay

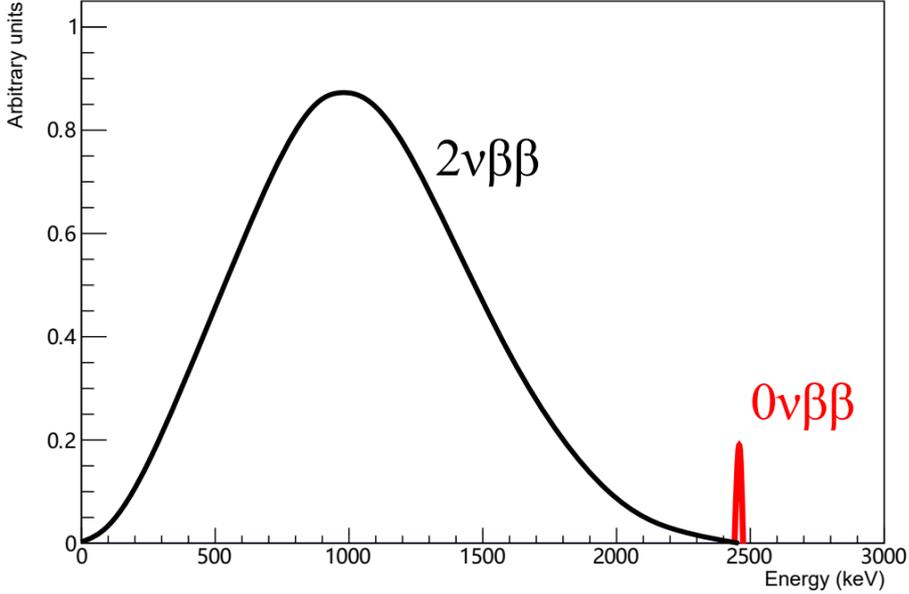


Figure 1.5: Diagram illustrating the energy spectrum of neutrinoless double-beta decay. The horizontal axis represents energy of electrons in keV, while the vertical axis denotes count in arbitrary units. Neutrino double-beta decay continuum is depicted in black, and a neutrinoless monoenergetic peak is highlighted in red. Relative sizes have been adjusted for clarity. It's worth noting that this peak exhibits some broadening due to two main factors: 1. The atomic nucleus carries a small amount of recoil energy, contributing to intrinsic broadening of electron energy. 2. Practical limitations in detector energy resolution. The nuclide featured in the diagram is ^{136}Xe , showcasing a monoenergetic peak situated near the Q value of 2458 keV.

features fewer final particles, resulting in a larger phase space factor. This characteristic helps compensate for the low probability associated with the m/E term. When Wendell Furry initially introduced the concept of neutrinoless double-beta decay in 1939, he posited that the majority of double beta decays were neutrinoless. However, we now understand that the decay rate is suppressed due to the necessity of matching the helicity of the neutrino [23]. While numerous double-beta decay cases involving neutrinos have been observed, there is still no evidence of neutrinoless double-beta decays.

1.2.4.2 Half-life time computation and Effective Majorana Mass

A notable characteristic of the $0\nu\beta\beta$ process lies in its unique ability to probe the absolute scale of neutrino masses. If we neglect the contributions from lepton-number violating mechanisms other than the $\nu_e - \bar{\nu}_e$ exchange of Fig. 1.4, the $0\nu\beta\beta$ rate is intricately linked to a parameter known as the squared effective Majorana neutrino mass, denoted as $\langle m_{\beta\beta} \rangle$. This relationship is expressed as follows [26]:

$$\frac{1}{T_{1/2}^{0\nu}} = G_{0\nu} |M_{0\nu}|^2 \langle m_{\beta\beta} \rangle^2 \quad (1.9)$$

In this context, $T_{1/2}^{0\nu}$ is the decay half-life, $G_{0\nu}$ is the phase space factor and $M_{0\nu}$ is the nuclear matrix element. These latter factors are derived from nuclear theory and exhibit notable variations depending on the specific theoretical models. The effective neutrino mass $\langle m_{\beta\beta} \rangle$ is a combination of the neutrino mass eigenvalues, m_k , with the neutrino mixing matrix, $U_{\alpha k}$:

$$\begin{aligned} \langle m_{\beta\beta} \rangle &= \left| \sum_k U_{ek}^2 m_k \right| \\ &= \left| m_1 c_{12}^2 c_{13}^2 + m_2 s_{12}^2 c_{13}^2 e^{2i\lambda_2} + m_3 s_{13}^2 e^{2i(\lambda_3 - \delta)} \right| \end{aligned} \quad (1.10)$$

Here, $c_{ij} = \cos \theta_{ij}$ and $s_{ij} = \sin \theta_{ij}$, where θ_{ij} are the mixing angles (or oscillation parameters), λ_i are the Majorana phases, and δ is the Dirac phase. We should mention that the matrix used differs from that of Eq. 1.3 by the phase δ , which accounts for CP violation, and the phases λ_i required by the Majorana nature of the neutrino.

Therefore, the value of $\langle m_{\beta\beta} \rangle$ is subject to variation based on the two potential scenarios for the neutrino mass hierarchy: the NH and the IH illustrated in Fig. 1.2. Fig. 1.6, shows the permissible ranges of $\langle m_{\beta\beta} \rangle$ plotted against the lightest neutrino mass in these two scenarios.

Cosmological considerations provide reasonable constraints, suggesting that the mass of the lightest neutrino, denoted as m_{lightest} , should not exceed 50 meV. Consequently, in the IH case $\langle m_{\beta\beta} \rangle$ is between 15 and 55 meV and in the NH case $\langle m_{\beta\beta} \rangle$ remains below 15 meV. However, some authors (*e.g.* [27]) disregard cosmological constraints, on the ground that they are model-dependent and affected by large uncertainties.

Exploring the domain of $\langle m_{\beta\beta} \rangle$ down to 15 meV requires a sensitivity to half-lifetime exceeding approximately 10^{27} years. Fig. 1.7 presents an evaluation of the requirements in various models, illustrating the dependence on the mass number A of the isotope [26].

Isotope	Daughter Nuclei	$Q_{\beta\beta}$ (keV)	Natural Ab. (%)	$T_{1/2}^{2\nu\beta\beta}$ (yr)
^{48}Ca	^{48}Ti	4267.98(32)	0.187(21)	$(6.4_{-0.6}^{+0.7}(st)_{-0.9}^{+1.2}(sys)) \times 10^{19}$
^{76}Ge	^{76}Se	2039.061(7)	7.75(12)	$(1.926 \pm 0.94) \times 10^{21}$
^{82}Se	^{82}Kr	2997.9(3)	8.82(15)	$(8.60 \pm 0.03(st)_{-0.13}^{+0.19}(sys)) \times 10^{19}$
^{96}Zr	^{96}Mo	3356.097(86)	2.80(2)	$(2.35 \pm 0.14(st) \pm 0.16(sys)) \times 10^{19}$
^{100}Mo	^{100}Ru	3034.40(17)	9.744(65)	$(7.12_{-0.14}^{+0.18}(st) \pm 0.10(sys)) \times 10^{18}$
^{116}Cd	^{116}Sn	2813.50(13)	7.512(54)	$(2.63_{-0.12}^{+0.11}) \times 10^{19}$
^{130}Te	^{130}Xe	2527.518(13)	34.08(62)	$(7.71_{-0.06}^{+0.08}(st)_{-0.15}^{+0.12}(sys)) \times 10^{20}$
^{136}Xe	^{136}Ba	2457.83(37)	8.857(72)	$(2.165 \pm 0.016(st) \pm 0.059(sys)) \times 10^{21}$
^{150}Nd	^{150}Sm	3371.38(20)	5.638(28)	$(9.34 \pm 0.22(st)_{-0.60}^{+0.62}(sys)) \times 10^{18}$

Table 1.1: Current target isotopes in prominent $0\nu\beta\beta$ -decay experiments along with their respective characteristics: Daughter Nuclei, $Q_{\beta\beta}$, Natural Abundance, and $2\nu\beta\beta$ -decay half-life values. The provided $2\nu\beta\beta$ -decay half-life values represent the utmost precision found in the literature. Adapted from [1].

While it is conventional to express the physics capabilities of $0\nu\beta\beta$ searches in terms of $\langle m_{\beta\beta} \rangle$, it is crucial to understand that this approach only reflects part of the $0\nu\beta\beta$ landscape.

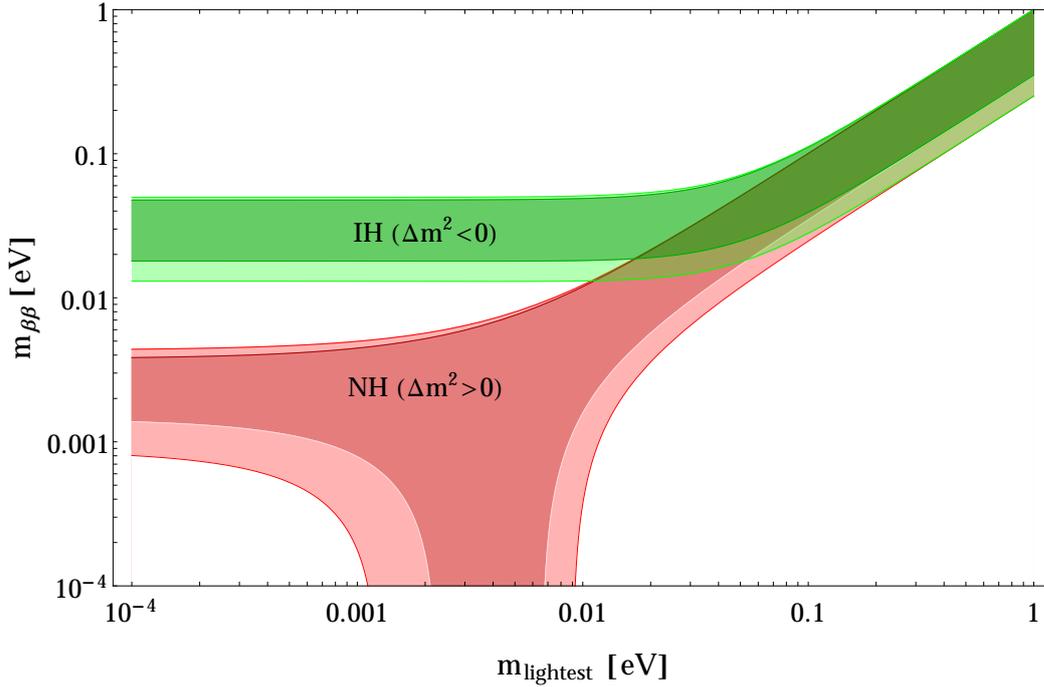


Figure 1.6: Predictions on $\langle m_{\beta\beta} \rangle$ as a function of the lightest neutrino mass, considering both the NH and the IH. The bands arise from our lack of knowledge on the phase parameters (see Eq. 1.10). The shaded regions represent the 3σ regions resulting from the propagation of uncertainties related to oscillation parameters. This figure is sourced from Reference [28].

There are other lepton-number violating processes that can be envisioned once the principle of lepton-number violation is acknowledged [1]. In the case that some of these turn out to be sizable, a detection of $0\nu\beta\beta$ decay would no longer be proportional to $\langle m_{\beta\beta} \rangle$. Yet, in any case, it remains that a detection immediately would imply that neutrinos are Majorana particles [29].

Double beta decay is present in 35 isotopes, but only a few of them are considered for $0\nu\beta\beta$ studies, taking into account factors such as endpoint ($Q_{\beta\beta}$ -value ≥ 2 MeV), natural abundance, and potential enrichment. Some isotopes are particularly advantageous because they can serve as both source and detector for $0\nu\beta\beta$ decay, i.e., ^{76}Ge , ^{130}Te , and ^{136}Xe . This dual role enhances the sensitivity of the experiment by maximizing the interaction probability and simplifying the experimental setup. Fig. 1.7 lists several prominent isotopes used in $0\nu\beta\beta$ decay searches. In $0\nu\beta\beta$ decay, the combined energy of the two emitted electrons is fixed at a specific value known as the $Q_{\beta\beta}$ -value, representing the mass difference between the mother and daughter nuclei (Tab. 1.1). The experimental identification of a distinct peak at the $2\nu\beta\beta$ endpoint serves as the signature for $0\nu\beta\beta$, as illustrated in Fig. 1.5. The region centered around the $Q_{\beta\beta}$ -value is termed the Region Of Interest (ROI). Achieving excellent energy resolution and minimizing background noise are crucial. For an experiment characterized by exposure time (t), detector target mass (M), isotopic abundance (a) of the relevant isotope in the detector target mass, atomic number (A), detection efficiency (ε), energy window of interest (ΔE) defined by the full width at half maximum ($FWHM$) resolution of the detector, and background level (b) normalized by time, mass, and energy, the expected number of background events over the experiment's

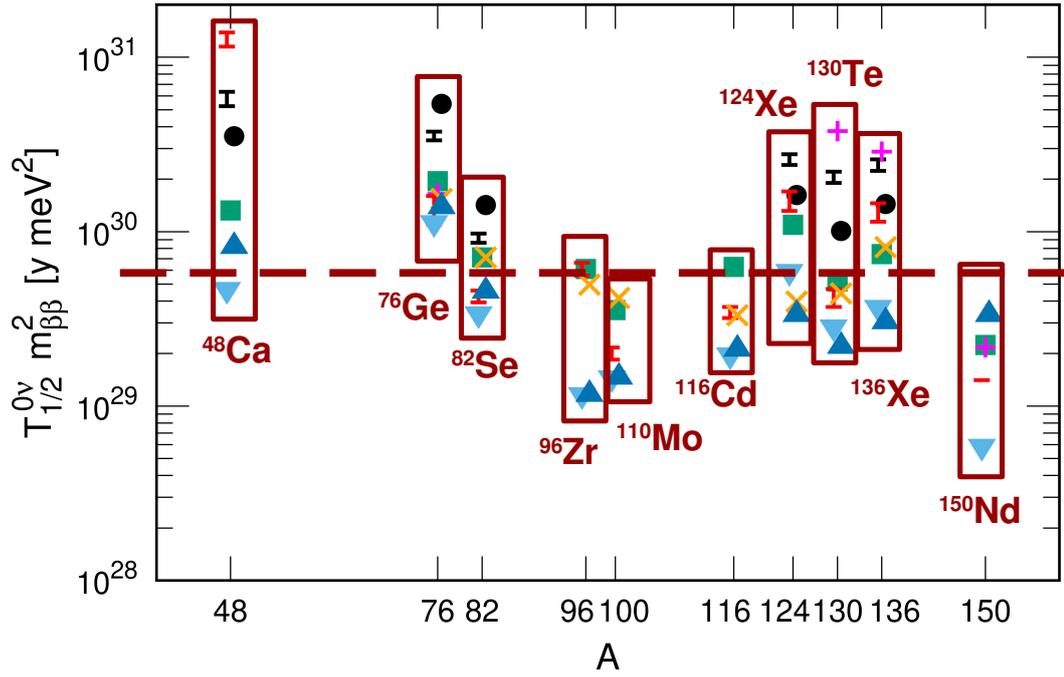


Figure 1.7: Distribution of decay half-lives, scaled by the square of the unknown parameter $\langle m_{\beta\beta} \rangle$ for $0\nu\beta\beta$ candidates as a function of mass number A . All the plotted results are obtained according to Eq. 1.9, using various nuclear models for the matrix element. Black circles, black bars - the shell model (SM), green squares - the interacting boson model (IBM-2), red bars, orange times signs, and magenta crosses - different versions of the quasiparticle random phase approximation (QRPA), and energy density functional theory (EDF), relativistic - downside cyan triangles, non-relativistic - blue triangles. Adapted from Ref. [26]. There exists a tendency for smaller decay half-lives in favor of the isotopes ^{96}Zr , ^{100}Mo , ^{118}Cd .

lifetime is given by $N_b = b \cdot \Delta E \cdot M \cdot t$. One must note that depending on the background level, the experiment's sensitivity to $0\nu\beta\beta$ decay may differ. Thus, two cases are then possible for the experiment's sensitivity to $0\nu\beta\beta$ half-lifetime:

$$\begin{aligned} N_B < 1 \text{ 'zero background'} &\Rightarrow T_{1/2}^{0\nu} \propto \frac{a \cdot \varepsilon}{A} M \cdot t \\ N_B \gg 1 &\Rightarrow T_{1/2}^{0\nu} \propto \frac{a \cdot \varepsilon}{A} \sqrt{\frac{M \cdot t}{b \cdot \Delta E}} \end{aligned} \tag{1.11}$$

1.3 Conclusion

The discovery of neutrinoless double-beta decay would represent a significant milestone in the field of neutrino physics, providing a direct confirmation of neutrinos as Majorana particles. The observation of this decay would also provide valuable insights into the absolute scale of neutrino masses, complementing the information obtained from neutrino oscillation experiments. The search for neutrinoless double-beta decay is a challenging task, requiring the development of sophisticated experimental techniques to achieve the necessary sensitivity. The successful detection of neutrinoless double-beta decay would have profound implications for our understanding of the fundamental properties of neutrinos and the nature of the universe.

Chapter 2

Neutrinoless Double-Beta Decay Experimental Setup

Якщо ти втомився, пам'ятай заради чого.
(Ukrainian: *If you are tired, remember what purpose it is for.*)
— Pavlo Petrychenko

2.1	Introduction	39
2.2	Different Experimental Approaches	40
2.2.1	Experiments related to ^{76}Ge	40
2.2.2	Experiments related to ^{130}Te	41
2.2.3	Experiments related to ^{100}Mo	42
2.2.4	Experiments related to ^{136}Xe	42
2.3	PandaX-III Experiment	44
2.3.1	Jinping Underground Laboratory	45
2.3.2	High pressure Gaseous Xenon TPC Detector	45
2.3.3	Micromegas Readout plane	47
2.3.4	Signal reading and processing	50
2.4	Limitations and Problematics	52
2.5	Conclusion	55

2.1 Introduction

Searching neutrinoless double-beta decay is a tricky task requiring a sophisticated experimental setup. To detect such rare events with half-lives expected to exceed 10^{25} years, the experimental setup must be highly efficient, with low background noise, preferably zero background, and high energy resolution.

There are a lot of experiments worldwide that are trying to tackle this problem using different isotopes and detection methods. This chapter introduces different experimental approaches, giving particular attention to the use of ^{136}Xe as a source and the detection medium for double-beta decay. ^{136}Xe is used in many experiments due to its high natural

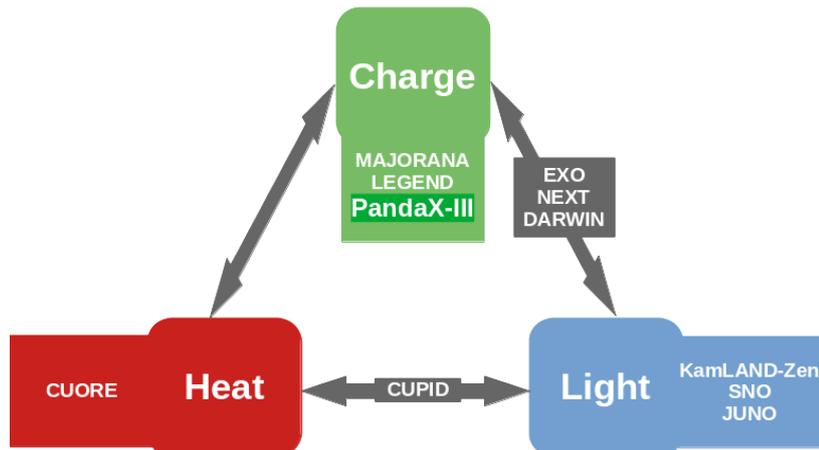


Figure 2.1: Different experimental approaches for neutrinoless double beta decay.

abundance, Q-value, and possible use in gas and liquid detectors, providing easy scalability. In PandaX collaboration, xenon is used mainly for the dark matter search [30][31][32], and the PandaX-III experiment is designed to explore the Majorana nature of neutrinos by searching for neutrinoless double-beta decay [33].

The PandaX-III collaboration is composed of several institutes such as the Shanghai Jiao Tong University (SJTU), the University of Science and Technology of China (USTC), the University of Zaragoza, and the *Commissariat à l’Energie Atomique et aux Energies Alternatives* (CEA) in France through the laboratory of the Institute of Research on the Fundamental Laws of the Universe (IRFU). The PandaX-III experiment is a high-pressure gaseous ^{136}Xe time projection chamber (TPC) [34] detector with Micromegas readout within the Chinese Jinping Underground Laboratory (CJPL). This section will explain the readout system, data acquisition, and data preprocessing, followed by a discussion of the experiment’s limitations and problematics. It is important to understand the challenges of the experimental setup that will be tackled in the latter chapters.

2.2 Different Experimental Approaches

In general, most rare event experiments (*e.g.* dark matter searches, neutrinoless double beta decay, etc.) use methods to detect light signals, electrical signals, or heat signals, individually or in combination. Globally recognized neutrinoless double-beta decay experiments employing these three detection methods are marked in Fig.2.1. As mentioned in the previous chapter, the beneficial isotopes that are commonly used for these kinds of experiments are ^{76}Ge in semiconductor detectors, ^{130}Te and ^{100}Mo in crystal detectors, and ^{136}Xe in gas/liquid detectors.

Let us briefly introduce the experiments related to these isotopes.

2.2.1 Experiments related to ^{76}Ge

Because of its excellent properties, germanium is widely used in the semiconductor industry to make detectors. It has high energy resolution and reliability and responds quickly. The industry has effective methods to purify ^{76}Ge , so detectors made from it have very low

background noise. It makes ^{76}Ge a promising material for neutrinoless double beta decay experiments. However, one drawback is that the phase space factor for ^{76}Ge is relatively small. This means experiments using it may need to measure a higher half-life. Also, it is challenging to scale up individual detectors because of how ^{76}Ge crystals grow, so they are often arranged in arrays, which limits how much the experiment can grow.

Globally leading experiments involving ^{76}Ge include GERDA and the Majorana Demonstrator, led by Germany and the U.S., respectively. These experiments have collaborated to form LEGEND, a global initiative to advance the next generation of Ge-based neutrinoless double beta decay detectors. What follows is a brief introduction detailing the peculiarities of these experiments.

- *GERDA* [35]: Located at the Gran Sasso Underground Laboratory (LNGS), GERDA experiment [36] uses high-purity Ge detectors to explore neutrinoless double beta decay. Starting in 2015 with 17.7 kg of ^{76}Ge the experiment expanded to 44.2 kg by 2019, accumulating an exposure of 127.2 kg-year. Utilizing effective waveform-event selection and boasting an outstanding energy resolution, GERDA observed zero counts in the region of interest, establishing a half-life limit of 1.8×10^{26} years.
- *Majorana Demonstrator* [37]: stationary at the Sanford Underground Research Facility (SURF), it employs high-purity Ge detectors arranged in a tombstone configuration. Starting data acquisition in 2015, it achieved an exposure of 51.7 kg-year. Leveraging waveform-event analysis, the experiment set a lower limit on the half-life of 2.7×10^{25} years [38].
- *LEGEND* [39]: Formed in 2017, it aims to develop the next generation of Ge-based neutrinoless double beta decay experiments. The LEGEND-200 experiment, initiated with an exposure of 200 kg-year, began commissioning in 2022. With plans to operate for five years, achieving an exposure of 1 tonne-year, LEGEND-200 could potentially reach a sensitivity level of 10^{27} years. The ultimate goal is the LEGEND-1000 experiment, targeting an exposure of 10 tonne-year with the potential to achieve a sensitivity level of 10^{28} years. As of now, LEGEND-1000 is in the preliminary research phase, with operations dependent on external conditions.

2.2.2 Experiments related to ^{130}Te

The Tellurium-130 (^{130}Te) isotope, with its natural abundance of 34.08%, emerges as a favorable candidate for neutrinoless double-beta decay experiments due to its abundant availability, eliminating the need for enrichment and lowering experimental costs. However, it emits gamma rays during decay, posing challenges in shielding to maintain low background noise, which is critical for achieving the desired sensitivity. Two prominent experiments exploring ^{130}Te are SNO+ and CUORE.

- *SNO+* [40][41]: Located in Canada's SNOLAB underground laboratory, SNO+ extends the original SNO solar neutrino detector's capabilities to investigate neutrinoless double-beta decay in ^{130}Te . The experiment repurposes existing infrastructure, including a 12-meter diameter acrylic sphere, a large water tank, and over 9000 photomultiplier tubes. By submerging the acrylic sphere filled with liquid scintillator containing 3.9 tonnes of ^{130}Te into the water tank, it aims to capture scintillation

light signals produced by neutrinoless double-beta decay events. With an anticipated sensitivity of 2.1×10^{26} years over a five-year exposure period, SNO+ aims to boost this sensitivity to 1×10^{27} years in future phases.

- *CUORE* [42]: Operating at the Gran Sasso Underground Laboratory (LNGS), CUORE employs ^{130}Te within tellurium dioxide crystals held at extremely low temperatures for detecting neutrinoless double-beta decay. Each crystal, cut and polished into $5 \times 5 \times 5 \times \text{cm}^3$ cubes, houses highly sensitive Neutron Transmutation Doped (NTD) semiconductor temperature sensors. CUORE has evolved from two prototype phases, CUORICINO and CUORE-0, to a scale of 204 kg of effective nucleons. Despite active data collection since 2017 and an accumulated exposure of 1 tonne-year, CUORE has yet to observe any signal of neutrinoless double-beta decay, setting a stringent lower limit on the half-life at 2.2×10^{25} years [43]. Its next-generation experiment, CUPID, aims to achieve "zero background" by integrating light signal collection for particle identification based on low-temperature crystal calorimeters.

2.2.3 Experiments related to ^{100}Mo

A descendant of the CUORE experiment, CUPID-Mo (CUORE Upgrade with Particle Identification in Molybdenum) [44] is a neutrinoless double-beta decay experiment that consists of Li_2MoO_4 scintillating bolometers containing the nucleus of interest, ^{100}Mo . Each bolometric crystal has a Germanium light detector to improve particle identification capabilities. During the period between 2019 and 2020, CUPID-Mo collected a total exposure of 2.71 kg-year of Li_2MoO_4 , yielding significant physics and technical outcomes. Notably, it achieved the most stringent worldwide constraint on the neutrinoless double beta decay half-life of Mo-100, setting a limit of 1.8×10^{24} years. This translates to limits on the effective Majorana neutrino mass ranging from 280 to 490 meV. Ongoing analysis may also unveil other important insights into various hypothetical exotic processes. Advancements in background suppression techniques, cryogenic readout, and physics searches beyond $0\nu\beta\beta$ are moving toward the next phase of the CUPID-1T experiment, which aims to achieve a sensitivity of 10^{27} years on the half-life of Mo-100 [45].

2.2.4 Experiments related to ^{136}Xe

The ^{136}Xe isotope is a versatile candidate for neutrinoless double-beta decay experiments. It can be used in gas and liquid detectors, providing easy scalability. The isotope's relatively high natural abundance of 8.85% and large Q-value of 2457.83 keV make it an attractive choice for neutrinoless double-beta decay experiments. A forefront candidates in the search for neutrinoless double-beta decay using ^{136}Xe are the KamLAND-Zen, EXO, NEXT, and PandaX-III experiments.

- *KamLAND-Zen 800* [46]: The KamLAND-Zen 800 experiment, located in the Kamioka Observatory, Japan, employs 2-type of liquid scintillator: 1000 tons of pure liquid scintillator and 745 kg Xe-loaded liquid scintillator with 91% enrichment. The most recent results from the KamLAND-Zen 800 experiment have set a lower limit for the $0\nu\beta\beta$ half-life of 2.3×10^{26} years at 90% confidence level, corresponding to upper limits on the effective Majorana mass of 36-156 meV.

- *NEXT* [47]: NEXT stands for Neutrino Experiment with a Xenon TPC, housed at the Canfranc Underground Laboratory (LSC) in Spain. It operates, as the name suggests, a high-pressure ^{136}Xe gas time projection chamber (TPC) to explore neutrinoless double-beta decay to with a pixelized tracking plane and a calorimetric plane. The NEXT-White detector (a predecessor of the NEXT-100) with 3.5 kg of enriched ^{136}Xe has showed an energy resolution below 1% FWHM at the Q-value of the decay, and allowed to set a lower half-life limit of 1.3×10^{24} years at 90% confidence level. This corresponds to an upper limit on the effective Majorana mass of 48-207 meV. The NEXT-100 experiment with 100kg of ^{136}Xe serves as a stepping stone towards larger-scale experiments, aiding in technical validation and background model verification. Given the elusive nature of neutrinoless double-beta decay, NEXT explores two approaches for enhanced sensitivity: NEXT-HD, which scales up NEXT-100's mass and design changes, and NEXT-Bold, incorporating a barium tagging technique. Barium tagging in NEXT-Bold involves detecting the presence of a single barium ion produced during neutrinoless double-beta decay events. This technique aims to confirm the occurrence of such events with high specificity, significantly reducing background noise and enhancing the experiment's sensitivity [48].
- *EXO* [49, 50]: A U.S.-based experiment, EXO-200, was designed to explore neutrinoless double-beta decay using a TPC with 80% enriched 200 kg liquid ^{136}Xe . A wire grid collected 2D location data together with the energy deposition of the events. The scintillating light on the other hand provided an aid in reconstructing the event's 3D location and enhancing energy measurement. The final results from the whole EXO-200 dataset that was collected between 2011 and 2014 in the first phase of the experiment, and between 2016 and 2018 in the second phase, conclude with a half-life limit of 3.5×10^{25} years and an effective Majorana mass limit of 93-286 meV. A tonne scale experiment, nEXO, is currently under extensive R&D and design phase, aiming to achieve a sensitivity beyond 10^{28} years on the half-life of ^{136}Xe [51].

In the Tab.2.1, we summarize the experiments described above with the expected neutrinoless double-beta decay half-life and effective Majorana mass limits. The PandaX-III

Nuclide	Experiment	Update year	Half-life (years)	$\langle m_{\beta\beta} \rangle$ (meV)
^{76}Ge	GERDA	2020[35]	$> 1.8 \times 10^{26}$	$< (79 - 180)$
^{76}Ge	Majorana Demonstrator	2022[38]	$> 2.7 \times 10^{25}$	$< (70 - 207)$
^{76}Ge	LEGEND-200	2022[39]	potentially $> 10^{27}$	-
^{130}Te	CUORE	2022[43]	$> 2.2 \times 10^{25}$	$< (90 - 305)$
^{130}Te	SNO+	2021[41]	$> 2.1 \times 10^{26}$	$< (50 - 200)$
^{100}Mo	CUPID-Mo	2022[44]	$> 1.8 \times 10^{24}$	$< (280 - 490)$
^{136}Xe	KamLAND-Zen	2022[46]	$> 2.3 \times 10^{26}$	$< (36 - 156)$
^{136}Xe	EXO-200	2019[50]	$> 3.5 \times 10^{25}$	$< (93 - 286)$
^{136}Xe	NEXT	2023[47]	$> 1.3 \times 10^{24}$	$< (48 - 207)$

Table 2.1: Representative experiments of each nuclide and the neutrinoless double-beta decay half-life and effective Majorana mass limits.

experiment employs ^{136}Xe as the source and detection medium for neutrinoless double-beta

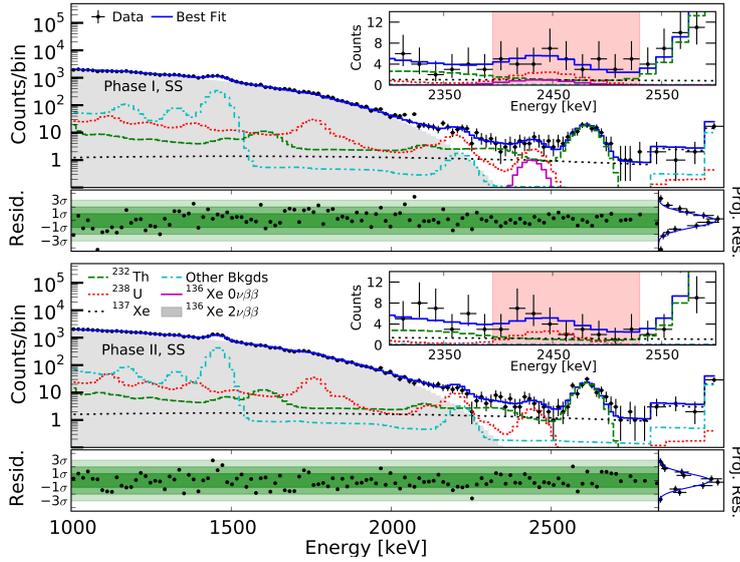


Figure 2.2: Background model and data from the EXO-200 experiment. The $0\nu\beta\beta$ signal peak (purple, visible on the Phase I plot) is covered by the background events from ^{238}U and ^{232}Th decay chains. Plot adapted from [50].

decay. With the current constraints and the half-life limits of different experiments in mind, the next section provides a detailed explanation of the PandaX-III experiment.

2.3 PandaX-III Experiment

The experiment uses a high-pressure gaseous TPC with a Micromegas readout. One of the main differences between the PandaX-III design and the experiments introduced above is a neglect of the scintillation light signal. In fact, the scintillation is suppressed by the quencher gas, and the experiment only reads out the electrical signal from the Micromegas array on top of the TPC. This design choice is made to maximize the particle track detection capability and minimize the statistical fluctuations in the energy resolution. The PandaX-III detector design is expected to have a 3% energy resolution at the Q-value of the $0\nu\beta\beta$ decay with an R&D goal of 1% energy resolution. The experiment is currently in finalization of the R&D and in the building phases and aims to achieve a sensitivity of 10^{26} years on the half-life of ^{136}Xe . In using ^{136}Xe as the source and detection medium for $0\nu\beta\beta$ detection, we must understand that $Q_{\beta\beta}$ of ^{136}Xe lies in the range of natural background gamma rays. Therefore, the ROI for the $0\nu\beta\beta$ search, defined by the detector's energy resolution, will have a high probability of being contaminated by background events, coming from the ^{238}U and ^{232}Th decay chains, which are the most abundant isotopes in the detector materials. Fig. 2.2 shows the data and the best background model fit from the EXO-200 full dataset analysis. The expected peak for the $0\nu\beta\beta$ signal is covered by the background events from the ^{214}Bi and ^{208}Tl decays.

A topological signature of the events is needed to suppress the background events. In the case of the PandaX-III experiment, the Micromegas array plays a crucial role in providing this information.

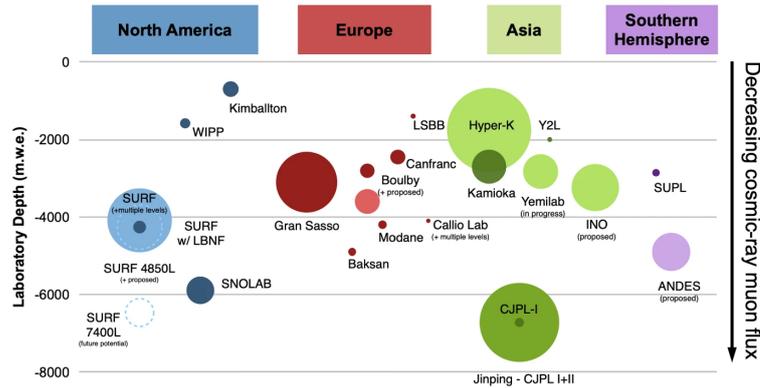


Figure 2.3: Comparison of the m.w.e. and relative excavated volume for different underground laboratories. The Jinping Underground Laboratory is the deepest laboratory in the world with a 6720 m.w.e shielding. Plot adapted from [53].

2.3.1 Jinping Underground Laboratory

The selection of the laboratory is of paramount importance in conducting low-noise experiments. This choice has significant implications for the ambient background noise to which the experiment will be subjected. The PandaX-III experiment is located in the Jinping Underground Laboratory (CJPL). It is situated in Sichuan Province, China, alongside a hydroelectric dam carved into the Jinping Mountains, which utilizes the Yalong River. This laboratory has advantages for conducting a low-background experiment, located 2400 meters beneath the Jinping mountains. The laboratory configuration allows it to achieve a cosmic radiation rate of 1 muon/week/m² thanks to the natural attenuation of the overlying rock [52]. This makes the CJPL laboratory the deepest laboratory in the world with a 6720-meter water equivalent (m.w.e) shielding [33][53]. The comparison of m.w.e. and relative excavated volume for different underground laboratories is shown in Fig. 2.3.

Furthermore, the laboratory is situated in a layer of marble, a rock naturally lower in radionuclides such as ²³²Th and ²³⁸U, whose decay chains are problematic in managing experiment background noise. These characteristics make the Jinping Underground Laboratory one of the best-shielded laboratories in the world against external radiation, making it ideal for conducting a low-background experiment like PandaX-III.

2.3.2 High pressure Gaseous Xenon TPC Detector

The PandaX-III setup, depicted in Fig. 2.4, features a high-pressure gaseous TPC as its central component. This TPC includes a charge readout plane positioned at the top, a cathode at the bottom, and an electric field shaping cage between them. An electric field of 1 kV/cm is applied across the TPC, ensuring the drift of ionized electrons towards the charge readout plane. A high-pressure stainless steel container with a volume of 6.3 cubic meters featuring a lower tank and an upper flange. The whole vessel weighs around 5 tons and stands at a height of approximately 2300 mm and a diameter of 2100 mm. Suspended from the top flat flange of a stainless steel pressure vessel, the charge readout plane operates within a maximum pressure of 10 bar, while the vessel can withstand a pressure of 13 bar. The PandaX-III collaboration aims towards a 1-ton scale experiment, with the current setup serving as a prototype for future large-scale experiments requiring five of these

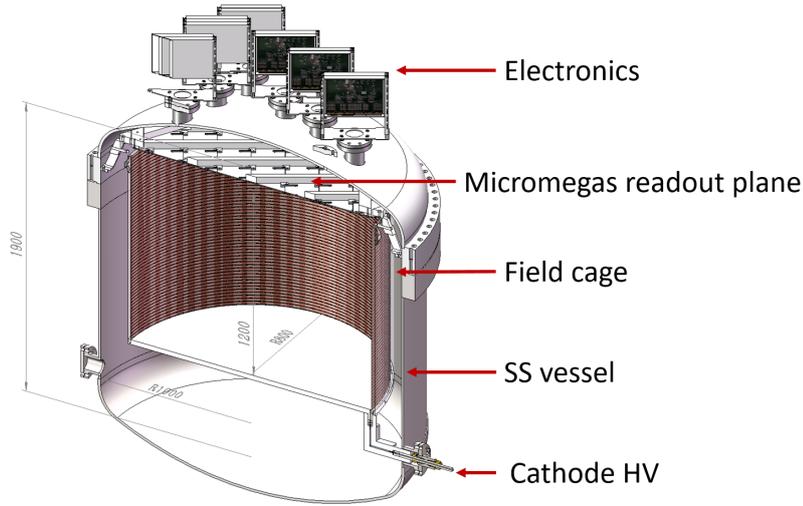


Figure 2.4: PandaX-III high pressure vessel design cross-section. A lower tank and an upper flange, fabricated through welding of cylindrical body, flange discs, and ellipsoidal heads. The cylindrical body has a 12 mm wall thickness, 2000 mm inner diameter, and stands about 1100 mm tall. Ellipsoidal head has an 18 mm wall thickness and a 2000 mm inner diameter. Flange discs, connecting the lower tank and the upper flange - 2180 mm in diameter and 164 mm thick. Weighing around 5 tons, the entire container is approximately 2300 mm tall. The lower tank includes four side flanges for various purposes, while the upper flange has 12 flanges for signal readout and 2 smaller flanges for air circulation. The field cage is built out of acrylic barrel, with an inner diameter of 1.6 meters and a height of 1.2 meters. The strip-shaped flexible PCBs form shaping rings interconnected by resistors, totaling $119\text{ G}\Omega$ resistance. Sourced from [56].

TPCs. Currently, the TPC contains 140 kg of 90% enriched ^{136}Xe gas in the active volume inside the electric field cage. Introducing a trimethylamine (TMA) quencher we increase the ionization ratio of the medium, reducing the light yield [54]. In PandaX-III, this enhances the intrinsic energy resolution by providing more primary electrons and minimizing statistical fluctuations while also improving track reconstruction accuracy by reducing electron diffusion during drift [55]. Therefore, the total mixture is a 10 bar Xenon(99%):TMA(1%) [52][33]. Due to TMA's corrosive nature, the detector's design avoids materials susceptible to corrosion, like rubber. Optical signal output is significantly reduced by quenchers, leading to the decision in PandaX-III only to read out the electrical signal using the Micromegas array atop the TPC to maximize track detection capability.

The PandaX-III detector's field cage, shown in Fig. 2.4, uses acrylic structures and flexible PCBs to maintain a uniform electric field. This design is cost-effective and helps control background levels while providing strong high-voltage performance. The cylindrical acrylic structure has flexible PCBs on the inner wall, acting as shaping rings to create a uniform electric field across an active volume of 1370 mm in length and 1696 mm in diameter. The flexible PCB, made of Kapton (polyimide) and thin copper layers, helps manage background due to its clean nature and low copper content. The Kapton film's insulation and acrylic's properties prevent sharp discharges, improving high-voltage stability.

The PandaX-III detector remains at SJTU for testing and operation and has not yet been relocated to the Jinping underground facility for low-background experiments.

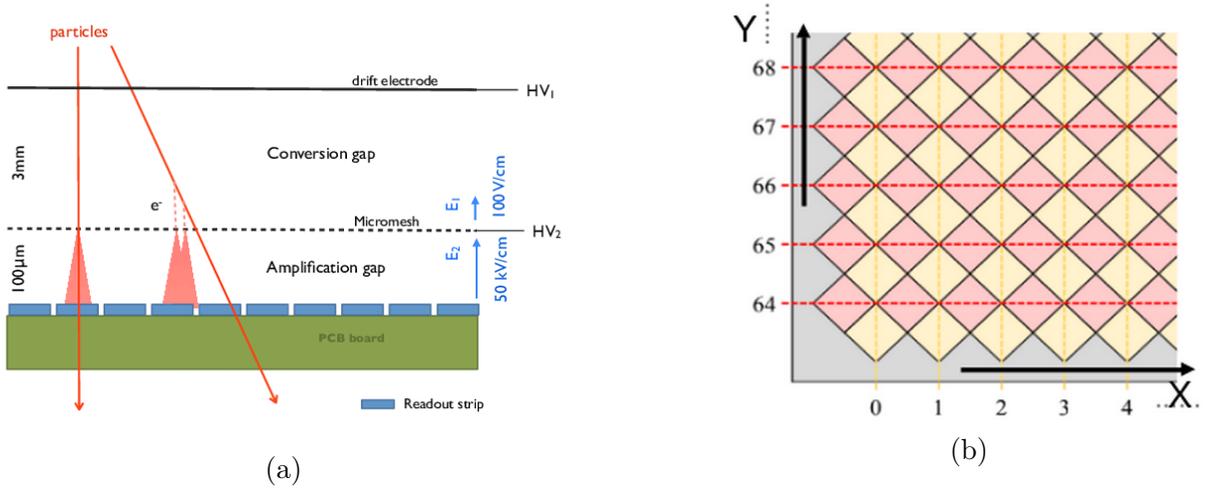


Figure 2.5: (a) General Micromegas readout plane design. The Micromegas module (MM) consists of a metal mesh layer, a Kapton layer, and readout strips. Within a narrow gap between the metal mesh layer and the Kapton layer, electrons undergo amplification via a powerful electric field. Embedded within this gap are readout strips, which capture the charges. Adapted from [58]. (b) Micromegas readout strips visualization. The readout strips are squared copper pads connected in strip channels, 64 in X and 64 in Y directions.

2.3.3 Micromegas Readout plane

In PandaX-III, the Micromegas (Micro-Mesh Gaseous Structure) technique, a proven method for Micro Pattern Gas Detection (MPGD) [57], is utilized for charge measurement. A Micromegas module (MM) comprises a metal mesh layer, a Kapton layer, and readout strips (Fig. 2.5a). In our case, the readout strips are composed of square copper pads on a PCB. Each pad is around 2 mm × 2 mm in size, resulting in a 3 mm pitch. The pads are connected in strip channels, 64 in X and 64 in Y directions, to form a strip readout configuration visualized in Fig. 2.5b.

A MM has a flat surface that incorporates a micromesh structure aimed at electron collection. Below this micromesh is a thin amplification zone where a robust (much stronger than the drift field, typically 50 times) electric field operates. Electrons gathered by the microgrid experience avalanche amplification within this region before being directed toward the bottom anode. Employing Micromegas for readout in gas detectors offers superior granularity and energy resolution compared to conventional multi-wire chambers. Each MM employed covers an active area of approximately 200 mm × 200 mm, featuring a total of 64 readout strips in both horizontal and vertical orientations.

Micromegas detectors, specifically Bulk Micromegas, were developed to strengthen the amplification areas of detectors and make large-scale production easier [59, 60]. They combined the anode, tracks, and microgrid into one structure. The metal microgrid is stretched over a frame above the readout tracks, maintaining a thin space for amplification. Earlier models faced issues with the grid's alignment, which was solved by integrating the grid into resin posts for stability.

Bulk Micromegas detectors can be made using standard circuit printing methods, making large-scale production cost-effective. However, they sometimes have inconsistencies in amplification space, affecting energy resolution. Microbulk Micromegas [60], an improvement

over Bulk Micromegas, aims for better homogeneity in the amplification space while using less material. These detectors are constructed using a fixed-thickness polyimide layer onto which two $5\ \mu\text{m}$ copper layers are deposited. The lower copper layer is the detector's anode, while the upper layer forms the microgrid. The final manufacturing step involves chemically etching micrometer-sized holes, approximately $70\ \mu\text{m}$, which serve as the amplification mechanism. Unlike Bulk Micromegas, whose amplification space is a planar surface, Microbulk Micromegas confine the amplification space within an array of micrometer-sized holes. Precise control over layer thicknesses and the substrate's full grid support ensures theoretically excellent homogeneity in Microbulk Micromegas detectors. By selecting materials suitable for radiopure experiments, it is possible to achieve detectors less than $100\ \mu\text{m}$ thick, with low mass budgets, high radiopurity, and excellent homogeneity while retaining the primary advantages of Bulk Micromegas detectors. The schematics of the Microbulk Micromegas production process are shown in Fig. 2.6a.

Another technology employed in the PandaX-III experiment is the Thermal Bonding Micromegas (TBMM) [61, 62]. The thermal bonding process is a new technology developed by the USTC in recent years. It employs a heat-press bonding method to directly adhere flexible PCB induction electrodes to a thicker metal substrate, followed by an insulating layer, the germanium resistive layer, the amplification region, and the mesh (Fig. 2.6b). This technology possesses advantages such as stability, simplicity in processing, and environmental friendliness, making it the chosen technology for PandaX-III. However, the process requires further optimization to improve the uniformity of the amplification region and the energy resolution. It is essential to mention that compared to the Microbulk Micromegas design, the TBMMs do not possess an additional copper rim encircling the active area of the mesh. In Microbulk Micromegas, the inverted polarity is applied to this rim to deflect the ionization electrons that can drift to the rim away toward the mesh, preventing them from falling into the dead zone of the detector. Therefore, the absence of the rim in the TBMM design may lead to a degradation of the detection efficiency. Both Microbulk and TBMMs are shown in Fig. 2.6c and Fig. 2.6d respectively.

The Micromegas readout electrodes used in PandaX-III are square copper pads on a PCB. They are connected in strip channels, 64 in X and 64 in Y directions, to form a strip readout configuration. Such readout configuration allows for reconstructing particle trajectories in the XZ and YZ projections, providing a *two detectors in one* feature. Each strip is 3 mm wide, yielding a spatial resolution of $3\ \text{mm}/\sqrt{12} \approx 0.9\ \text{mm}$. The Micromegas' amplification gap measures $100\ \mu\text{m}$.

To efficiently cover the readout circular plane with a diameter of 1.6 m, MMs are arranged in a rectangular array. Each module is mounted on a specially designed aluminum support frame to ensure flatness and stability. Illustrated in Fig. 2.7, 52 MMs are organized from front to back in 8 rows, with quantities per row being 4, 6, 8, 8, 8, 8, 6, and 4 modules, respectively, corresponding to one or two DN80 flanges above for signal extraction. Based on the distance from Micromegas, each signal feedthrough flange traverses and bonds with different lengths of flexible PCB extension lines (the tails of the MMs).

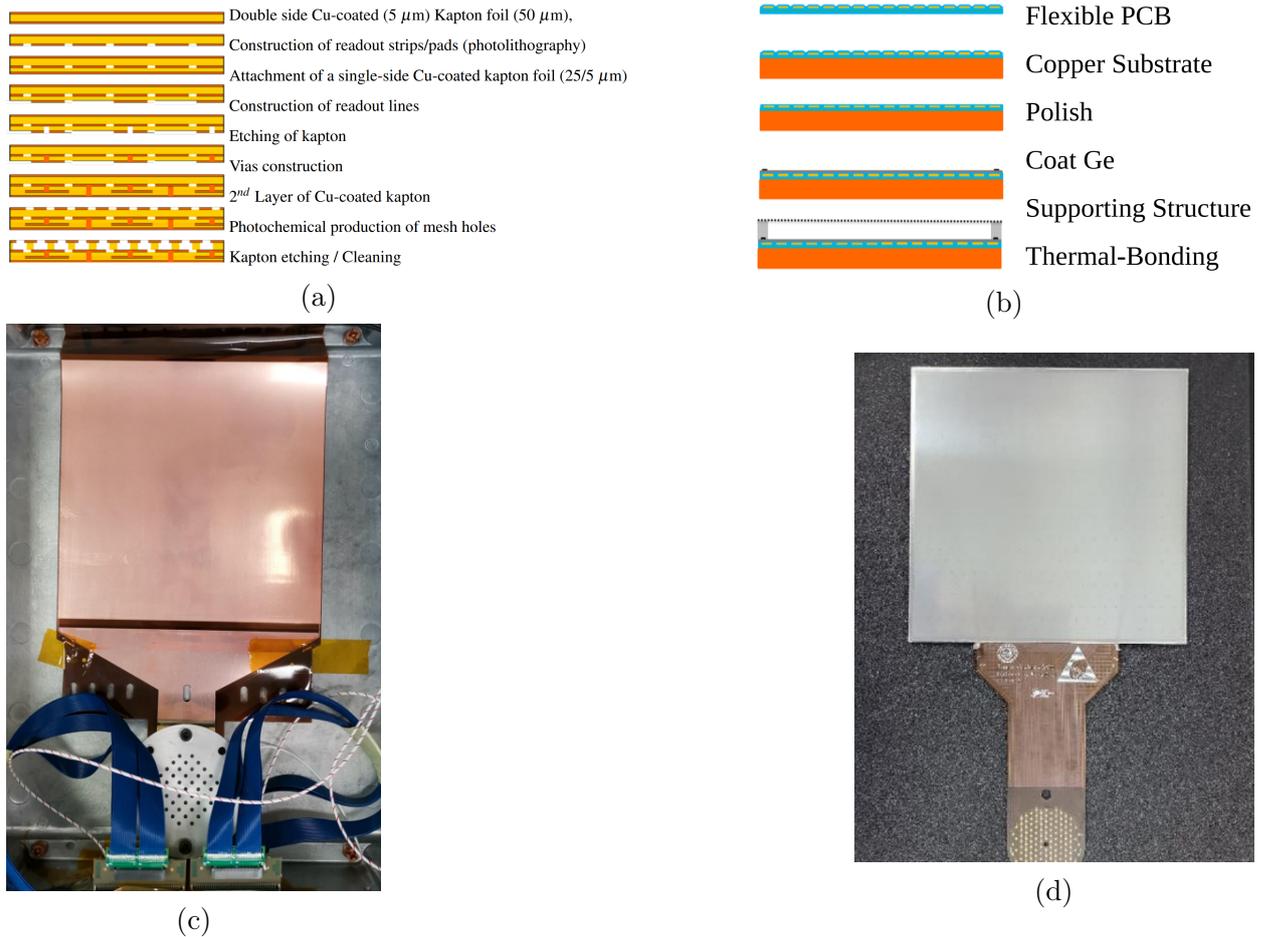


Figure 2.6: Micromegas production process. (a) MicroBulk Micromegas production process. Sourced from [60]. (b) Thermal Bonding Micromegas production process. Adapted from [62]. (c) MicroBulk Micromegas tested at CEA Saclay. Sourced from internal communication. (d) Finished Thermal Bonding Micromegas. The Thermal Bonding Micromegas are employed in the PandaX-III experiment. Sourced from internal communication.

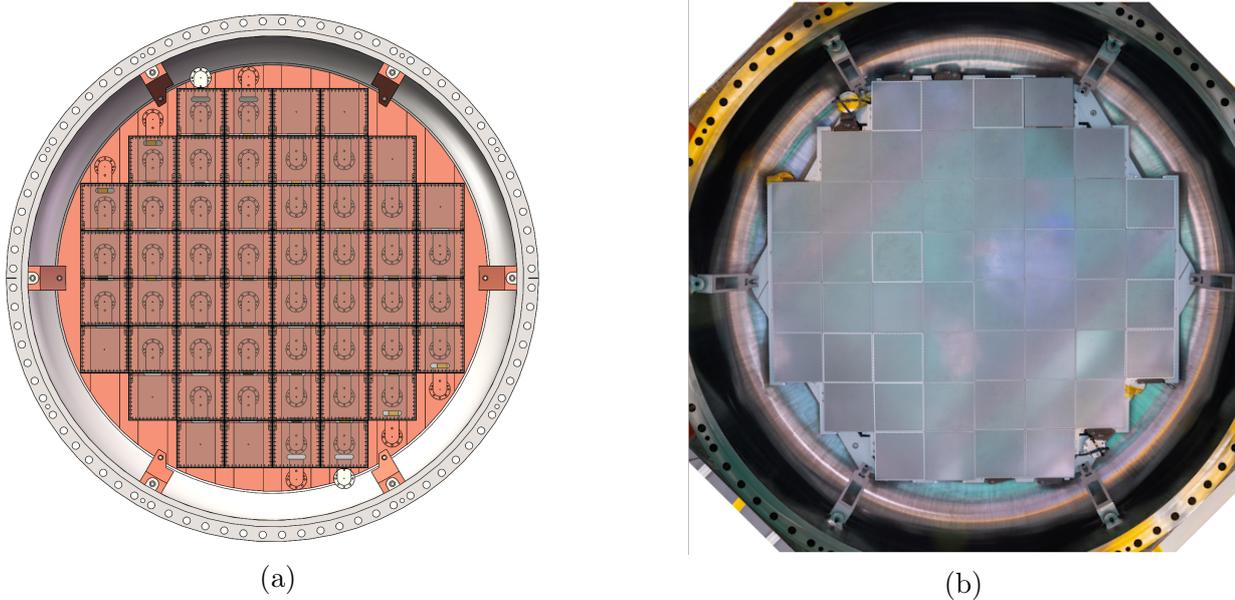


Figure 2.7: 52 Micromegas arrangement and design on the readout plane suspension frame. Adapted from [63].

2.3.4 Signal reading and processing

2.3.4.1 DAQ system

The data acquisition (DAQ) chain comprises front-end electronics, back-end electronics, and software. In PandaX-III, the front-end electronics (FEC) [64], also known as Front End Cards, utilize ASIC for Generic Electronic system for TPCs (AGET) chips [65]. Each AGET chip manages signals from 64 readout channels, offering a sampling frequency of up to 100 MHz, a dynamic range from 120 fC to 10 pC, which is adjustable for each channel, and a peaking time ranging from 50 ns to 1 μ s. Each channel primarily includes a charge-sensitive preamplifier, an analog filter (shaper), a discriminator for multiplicity building, and a 512-sample analog memory. Therefore, in the readout phase, the AGET chip digitizes the signals from each channel storing the data in a 512-sample buffer. Each FEC accommodates 4 AGET chips. Special efforts have been made to minimize the radioactivity of the FEC boards during fabrication by replacing the FR-4 epoxy laminate material with radio-pure Kapton in the PCB.

Digitized signals are transmitted via optical fibers to the back-end electronics known as TDCM (The Trigger and Data Concentrator Module), developed by IRFU (CEA) for various experiments. TDCM acts as a trigger distributor and a data aggregator for 32 FEC modules. Additionally, it provides a reference clock and handles the electronics' slow control. The DAQ software, built on the Midas framework, communicates with TDCM via Ethernet, establishing interactive interfaces for controlling and monitoring the data acquisition process.

2.3.4.2 Data processing

The REST (Rare Event Searches Toolkit) [66][67] Framework is mainly written in C++, and it is fully integrated with ROOT I/O interface. REST was born as a collaborative software

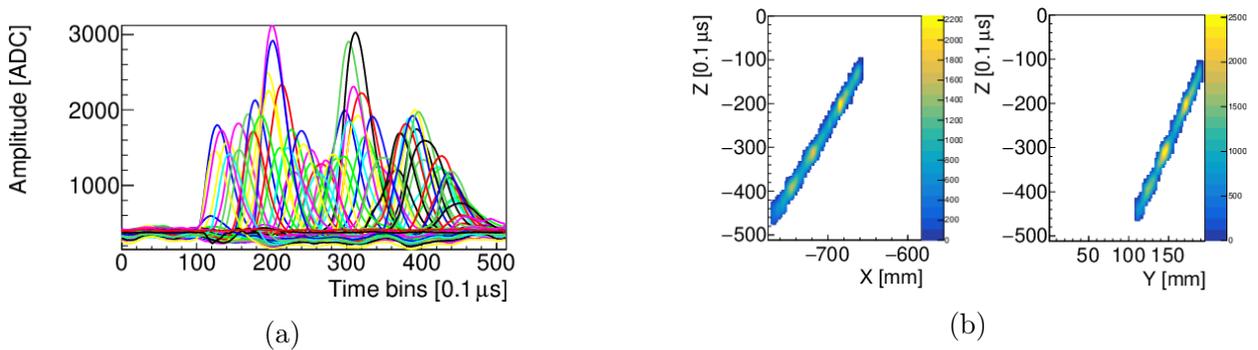


Figure 2.8: Pulses collected on each triggered strip (a) and projected tracks corresponding to XZ and YZ projections (b) of a cosmic-ray muon event detected. The color scale represents the amplitudes of the pulses. Adapted from [56].

effort to provide common tools for the acquisition, simulation, and data analysis of gaseous TPCs. The data processing pipeline in PandaX-III involves several steps, including signal processing, event reconstruction, and data analysis. The signal processing stage comprises pedestal subtraction, noise filtering, and pulse shape analysis. Pedestal subtraction is performed to eliminate the baseline noise from the signal, while noise filtering aims to reduce the noise level. Pulse shape analysis is conducted to extract the signal’s pulse height and time information. An example of the reconstructed signal of the detected cosmic-ray muon in the PandaX-III TPC prototype is shown in Fig. 2.8.

Aside from signal processing, REST offers tools for simulating events in the TPC geometry. This includes utilizing the Geant4 [68] and Garfield++ [69] packages. The simulation enables the creation of primary particles, particle propagation within the detector, ionization processes, particle diffusion, simulation of the detector’s response, gas amplification, signal shaping, digitization, *etc.* A Decay0 [70][71] extension enables us to conduct a Monte Carlo simulation for the generation of nuclear decays within the framework of double beta decay. In the context of my study, a datafile with the energy and angular distributions for electrons of the $0\nu\beta\beta$ of ^{136}Xe is generated using Decay0. This pregenerated datafile is then used as an input for Geant4 package to simulate the $0\nu\beta\beta$ events in the gas volume of the PandaX-III TPC.

The track reconstruction stage involves track reconstruction, energy calibration, clusterization of the pulses, track topology reconstruction, and analysis. Fig. 2.9a shows the simulated $0\nu\beta\beta$ signal pulses after the detector response simulation. After track processing, corresponding XZ and YZ track projections are shown in Fig. 2.9b. The black dots represent the amplitude of the pulses, corresponding to each strip at a given time bin (also called hits). The red and green circles represent clustered hits with a radius proportional to the sum of the amplitudes of the hits. The black lines represent the reconstructed tracks that can be used to perform the event selection.

Thus, REST provides a comprehensive set of tools for the data processing pipeline in PandaX-III, from signal processing to event reconstruction and analysis, allowing collaboration to perform a wide range of studies on the data collected from the experiment. A more detailed description of the REST framework can be found in Appendix A.

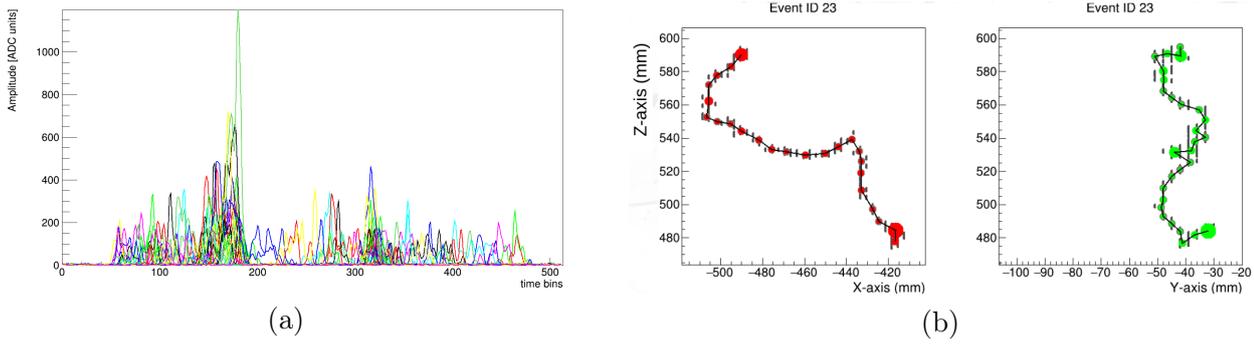


Figure 2.9: Simulated $0\nu\beta\beta$ signal pulses after the detector response simulation (a) and corresponding XZ and YZ track projections after track processing (b).

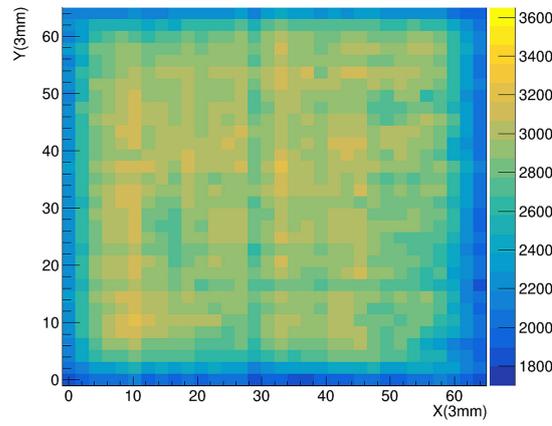


Figure 2.10: Gain inhomogeneities in the amplification region in one of the TBMMs. The color scale represents the gain of the amplification region. Sourced from [62].

2.4 Limitations and Problematics

Several factors can affect the data quality and the experiment's sensitivity to the signal of interest in a real-world experiment. Some limitations and problematics that can affect the PandaX-III experiment include gain inhomogeneities in the amplification region of the Micromegas, gaps between detectors, missing channels, *etc.*

The gain inhomogeneities in the amplification region of the Micromegas can lead to signal amplitude variations, affecting the detector's energy resolution. While the Thermal Bonding Micromegas technology shows promise in providing a stable amplification region for most of the detectors, the gain inhomogeneities are still present and will require additional calibration to correct for them (Fig. 2.10).

As mentioned in this chapter, the TBMM technology does not possess the copper rim around the active area of the mesh. With the current design, the gap between detectors (approximately 3 mm wide) can lead to a loss of the ionized electrons that can drift towards the gap, reducing the detection efficiency of the detector. This problem optimization is currently under study.

In addition, some channels in the Micromegas detectors of the PandaX-III TPC may encounter issues. These problems could arise from manufacturing flaws, handling during installation or maintenance, and the aging of the detectors, all of which are influential factors.

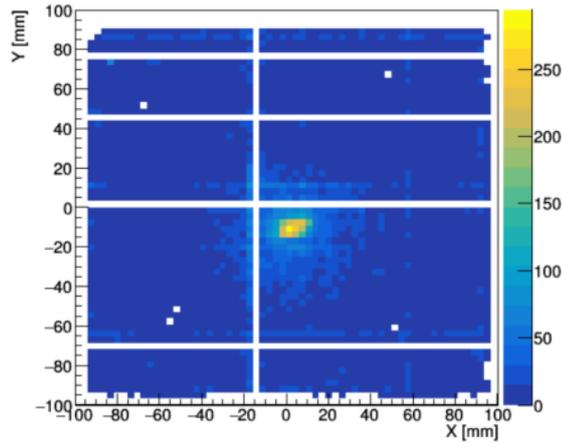


Figure 2.11: Hit map from a calibration run showing a missing channel in the Microbulk Micromegas. We can observe one channel with no signal on the X axis and four channels with no signal on the Y axis. The color scale represents the hit rate for each position on the MM.

If a channel is cut or if there is a substantial leakage current, it could render the inactive channel, causing the loss of all signals associated with it. An example of a missing channel in the Micromegas is shown on the hit map from a calibration run in Fig. 2.11.

My study will be focused on the missing channels problem in the Microbulk Micromegas detectors, as the Thermal Bonding Micromegas technology is still under development.

To understand the potential impact of missing channels on the event energy reconstruction, we simulate the decay events in the PandaX-III TPC using the REST framework. From here, the amplification gain on the micromegas is ideal and thus uniform across all channels. First, we simulate the $0\nu\beta\beta$ decay events with a readout plane configuration of 52 MMs without missing channels. Then, we simulate the same events with 1, 2, and 3 randomly missing channels at each MM. We define our ROI based on the energy resolution of the PandaX-III detector, which is $\sim 3\%$ FWHM at 2458 keV (which is the Q-value of the Double Beta decay for ^{136}Xe). The energy resolution in terms of standard deviation translates in $\sigma = 0.03 \times 2458 \text{ keV} / \sqrt{8 \ln(2)} \approx 31.3 \text{ keV}$. The ROI is defined as 3σ around the Q-value, which gives an energy range of [2364, 2552] keV. In Fig. 2.12, we compare the energy spectra of the $0\nu\beta\beta$ decay events with and without missing channels. The three missing channels is a conservative estimate of the number of channels that could be lost in the Micromegas detectors. The results show that the missing channels can affect the detector's energy resolution, leading to a broadening of the energy spectrum. At three missing channels per module, more than 20% of the events fall outside the ROI. This can lead to a loss of sensitivity to the signal of interest.

Additional attention should be paid to the condition when several channels are missing consecutively. In such a case, the more significant part of the track could be misreconstructed. This leads not only to the wrong energy reconstruction but also to the wrong topology reconstruction: the event track projection might be reconstructed as several separate tracks. Such an instance is demonstrated in Fig. 2.13. In his doctoral work [72], Benjamin Manier analyzed the impact of missing channels in the Microbulk Micromegas detectors on the energy resolution of the PandaX-III experiment. He also proposed an interpolation method to correct the energy loss due to the missing channels and the track topology reconstruction algorithm to mitigate the effect of the missing channels on the track reconstruction. The

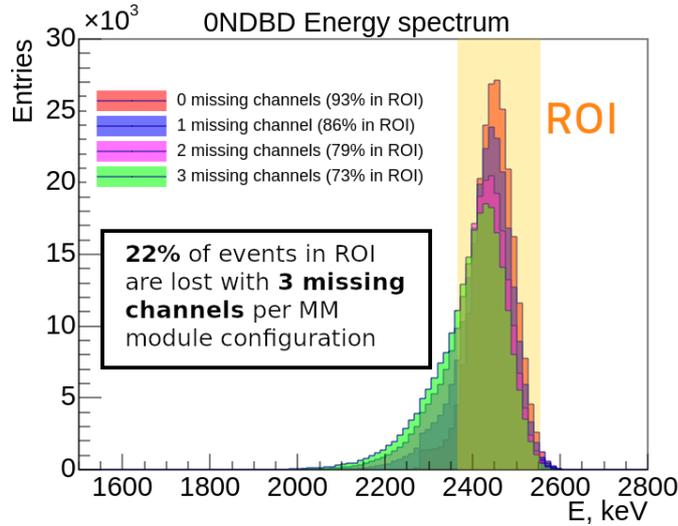


Figure 2.12: Comparison of the energy spectra of the $0\nu\beta\beta$ decay events with and without missing channels. ROI defined as 3σ around the Q-value of the decay ($\sim 3\%$ FWHM): [2364, 2552] keV

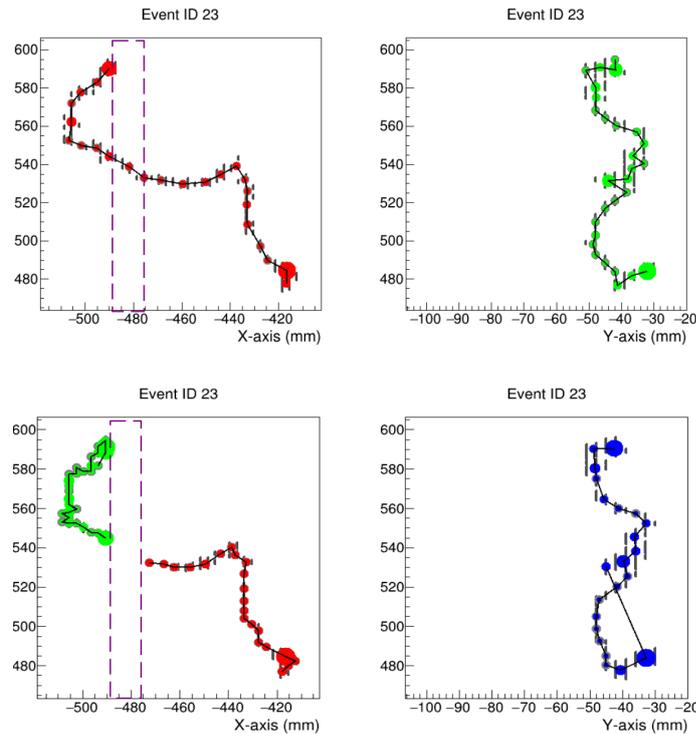


Figure 2.13: (a) NDBD event track simulated and reconstructed in REST software without missing channels. (b) Same NDBD event track simulated and reconstructed in REST software with missing channels. The area of the missing channels is highlighted in purple dashed rectangle on both plots. We can see that the missing channels lead to the separation of the track into two parts on the XZ projection. Thus, resulting in the misleading event track reconstruction - such topology information is useless for the analysis. On the YZ projection we can also observe that the black line representing the track has a connection at the end of the track to the point roughly in the middle of the track. This is due to the fact that the REST algorithm sometimes may have flaws that need to be calibrated.

interpolation method did not significantly improve the energy reconstruction of the events with missing channels. However, the track topology reconnection algorithm showed promising results. 71% of the events with up to 5 missing channels were effectively reconnected.

2.5 Conclusion

The PandaX-III experiment is a next-generation neutrinoless double beta decay experiment that aims to search for the neutrinoless double beta decay of ^{136}Xe . The experiment employs a high-pressure gaseous TPC with a Micromegas readout plane. The Micromegas technology provides a high granularity readout of the ionization signal in the TPC, offering superior energy resolution compared to conventional multi-wire chambers. The PandaX-III experiment is currently under construction at SJTU, and the TBMM development is being finalized at USTC and SJTU. The experiment is expected to be operational in the near future and should provide valuable data on the neutrinoless double beta decay of ^{136}Xe , contributing to the search for new physics beyond the Standard Model.

The missing channels in the Micromegas detectors can affect the efficiency of the experiment, leading to a loss of sensitivity to the signal of interest. The track topology reconstruction can also be affected by the missing channels, leading to the misidentification of the event and affecting the background rejection. The problem of missing channels should be addressed to ensure the optimal performance of the PandaX-III experiment. In the next chapters, I will present methods to correct for the energy loss due to the missing channels on Micromegas. While the next chapter is focused on the track topology study, the later chapters will be dedicated to the energy reconstruction study with Machine Learning algorithms employed to correct for the energy loss due to the missing channels.

Chapter 3

Topology of the Electron Tracks (Geant4)

Remember that REST is made by physicists for physicists,
who are supposed to toil and suffer till they become experts.
– *Rest-for-physics software welcome message*

3.1	Introduction	57
3.2	Low Energy Electrons in Gas	58
	3.2.1 Updated Blob Charge Identification Process	62
3.3	Electron Propagation Topology	70
3.4	Conclusions	76

3.1 Introduction

The PandaX-III experiment aims to search for $0\nu\beta\beta$ decay of ^{136}Xe . Electrons generated in the active TPC volume by gammas from the decay chains of ^{238}U and ^{232}Th with an energy equal to or above the $Q_{\beta\beta}$ value constitute the largest background source. They produce long tracks and a reconstructed energy that can fall into our ROI. Nevertheless, some of their characteristics, which can be put under the broad category of topological parameters, allow us to discriminate them against $\beta\beta$ events. The PandaX-III collaboration reported on the work done along this line in a 2020 article [73]. Several topological parameters were considered. The most effective is the so-called blob charge, *i.e.* the region of high energy density showing up at end-of-track, corresponding to the Bragg peak. The Bragg phenomenon also generates an end-of-track twist. $\beta\beta$ events have two blobs and twists, one for each of the two electrons, whereas γ events have only one, except in the rare case of pair creation. γ events also have, on average, a more significant number of secondary tracks and a longer track length.

Cuts on these parameters were optimized on a radio-purity model of the TPC. The results are presented in Section 6.1. They may seem sufficient. However, they suffer from two deficiencies. Firstly, they were obtained while independently treating topological parameters

(such as blob and twist) as well as track projections (XZ and YZ). They, hence, do not take full advantage of the information contained in the event. Secondly, they were obtained with perfectly working Micromegas. At the same time, we may have to deal with situations such as those described in Section 2.4 where some channels are missing, or the gain is inhomogeneous over the sensitive area. This former problem was addressed by Benjamin Manier in his thesis [72], with limited success. All these considerations led us to investigate the topology of electron tracks propagating in the TPC more carefully.

One line of investigation concerns correlations between blob and twist, as well as between

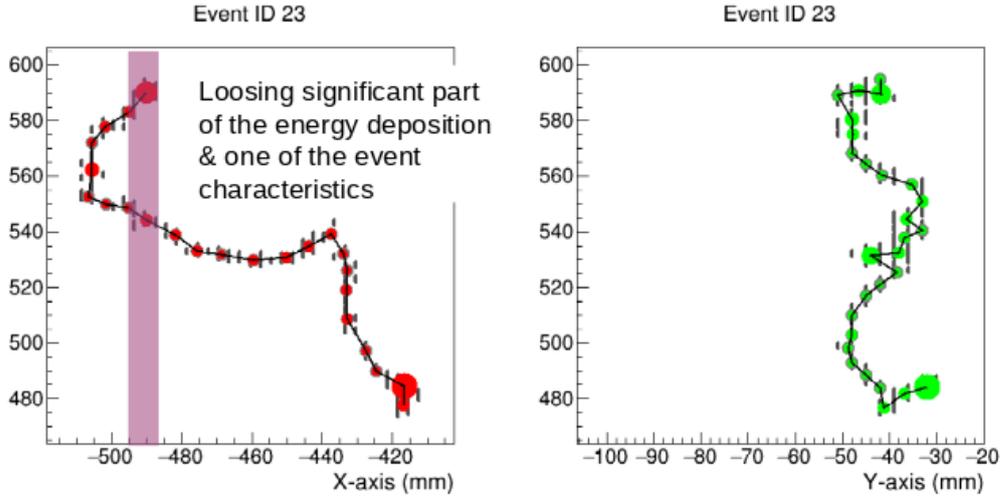


Figure 3.1: $0\nu\beta\beta$ decay event simulated and reconstructed in REST. One of the blobs is falling on a missing channel (highlighted in purple) in the XZ projection. The corresponding undetected energy could be reconstituted by exploiting the correlation between XZ and YZ so that the overall energy is reconstructed within the ROI.

XZ and YZ. It, in fact, addresses not only the first of the two deficiencies mentioned *supra* but also the second one. Indeed, if we are in a situation where channels are missing, we may want to take advantage of the correlations to reconstitute the missing information. Such a situation is illustrated in Fig. 3.1. But we will also try to characterize blob and twist *per se*.

In this chapter, to investigate the topology of the electron events in the gas, we will use, in addition to REST, the Geant4 Monte Carlo simulation toolkit.

3.2 Low Energy Electrons in Gas

First, let us consider that we have an electron and proton *beams* with energies 6 MeV and 110 MeV respectively, as we can see in the example from the depth dose curves for radiotherapy (Fig. 3.2). The ionization process dominates the proton beam energy deposition. One can clearly distinguish a proton Bragg peak position at a certain depth (on the plot, it is ~ 90 mm), which can be calculated from the Bethe-Bloch formula [74].

On the contrary, the energy deposition of the electrons is dominated by the bremsstrahlung process (particularly at energies above 10 MeV) and Multiple Coulomb scattering (MCS). The bremsstrahlung process is the photon radiation of an electron when it interacts with the Coulomb field of the nucleus. At the same time, MCS is the electron scattering on the nucleus's Coulomb field.

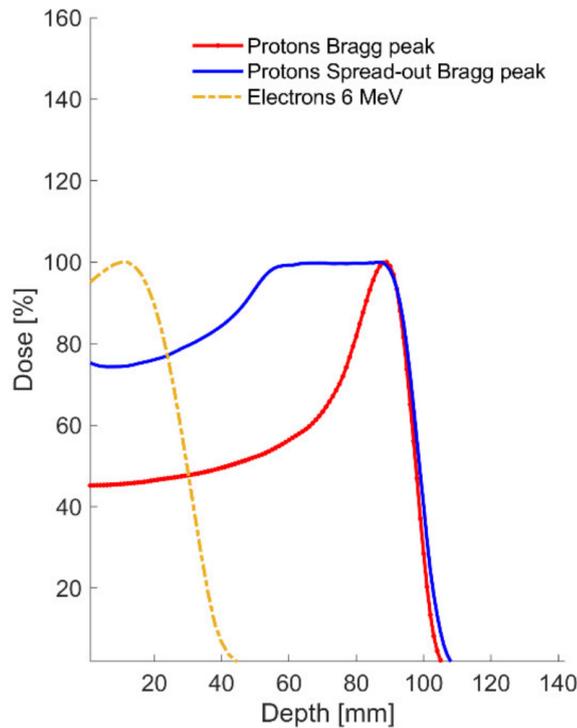


Figure 3.2: Comparison of relative depth dose distributions of protons and electrons in case of a single field. 110 MeV protons Bragg peak and 6 MeV electrons in water. Adapted from [75].

The energy deposition distribution of electron beams is generally smooth, decreasing monotonically with depth and lacking a distinct peak. Similar to protons, electrons experience a gradual energy loss primarily due to Coulomb collisions with atomic electrons. As electrons lose energy, their rate of energy deposition increases—a phenomenon crucial for the manifestation of a Bragg peak. However, this process differs significantly between electrons and protons due to the differing energy and momentum scales at which various interactions occur.

A key distinction arises from the much lighter mass of electrons compared to protons, which makes them more susceptible to the Coulomb field of the nucleus. This leads to a greater degree of scattering, which significantly alters their trajectory. For protons, which are much heavier, scattering occurs at smaller angles, allowing them to follow a relatively straight path. In contrast, electrons, due to their lighter mass, can scatter at large angles, rapidly losing memory of their initial direction. This scattering process, where the multiple scattering RMS angle is inversely proportional to the particle's momentum, leads to what is known as the "Blob" effect. This effect occurs at the same momentum scale for both electrons and protons but is more pronounced for electrons because of their lower mass and higher interaction cross-section.

Regarding energy loss, the Bethe-Bloch formula indicates that the rate of energy deposition (dE/dx) is inversely proportional to the square of the particle's velocity ($\sim 1/\beta^2$). This implies that for a non-relativistic particles, the energy loss rate increases sharply. The result is the formation of a Bragg peak, which, for protons, occurs in the sub-GeV range, and for electrons, in the sub-MeV range.

In the context of the PandaX-III experiment, however, we are primarily concerned with

lower-energy electrons, with the energy spectrum of events lying below 5 MeV and the ROI for the $0\nu\beta\beta$ decay signal around 2.5 MeV. At these energies, ionization dominates the energy deposition of electrons in the gas. Electrons "Blob" early in their trajectory due to significant scattering, and they only exhibit a Bragg peak towards the very end of their path. This behavior contrasts with protons, which typically manifest a Bragg peak before they have a chance to "Blob."

As was mentioned in the introduction to this chapter, the consecutively missing channels can lead to the loss of the Bragg peak from the track reconstruction (at least on one of the projections). Whether we can reconstruct the lost Bragg peak from the remaining information about the event arises. More precisely, the question is: *Can we approximate the Bragg peak on one of the projections (e.g XZ projection), knowing the deposited energy on another one (e.g. YZ projection), as visualized in Fig. 3.3?* To begin with, a rough

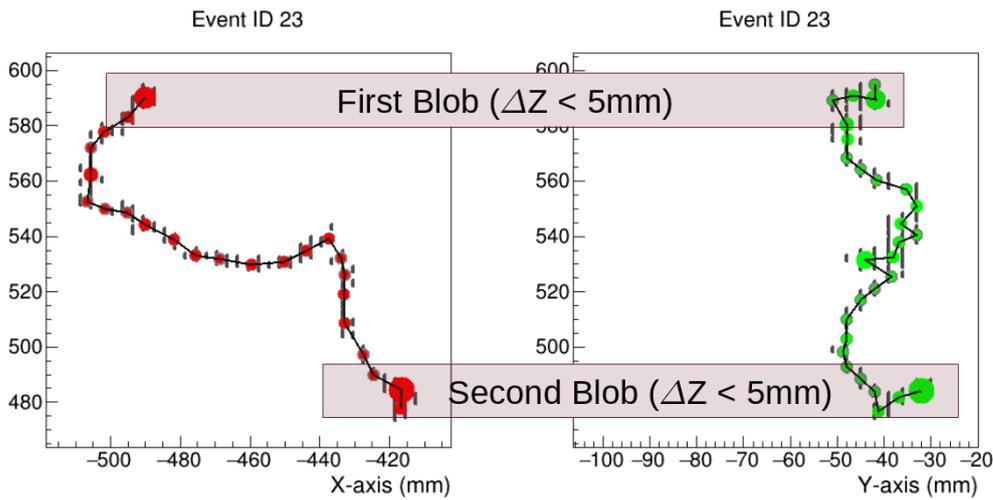


Figure 3.3: Visualization of the Bragg peak approximation on the XZ projection, knowing the deposited energy on the YZ projection.

estimation of the Blob charges (corresponding to the Bragg peaks) correlation was done using a process present in the REST software, dedicated explicitly to the Blob charge identification [66][67]. The process is executed independently after completing the track reconstruction in the two 2D projections, XZ and YZ. It searches for blobs at both ends of the reconstructed 2D tracks. Two parameters condition this process, set by the user: the fraction of track hits to consider (parameter *HitsToCheck*) and the blob charge radius (parameter *BlobRadius*). The most energetic hit within *HitsToCheck* determines the center of a circle of radius *BlobRadius* within which all hits are considered to calculate the total energy, as their sum, and the coordinates, as their barycenter, of the blob. Following the process terminology, the first Blob is the one closer to the anode (and correspondingly, the second Blob is the one further away from the anode).

With such a rough estimation of the Blob charge coordinates and energy depositions, we simulated 10^4 NDBD events inside the PandaX-III 140 kg Xe 1%TMA TPC setup in REST software. The *HitsToCheck* was chosen to be 20%, and the *Blob charge radius* was 10 mm. From the absolute difference between the Z coordinates of the Blob charges on the XZ and YZ projections, as presented in Fig. 3.4, we can observe that around 35% of the events have Blob charges with the absolute difference in Z coordinates higher than 5 mm. Furthermore,

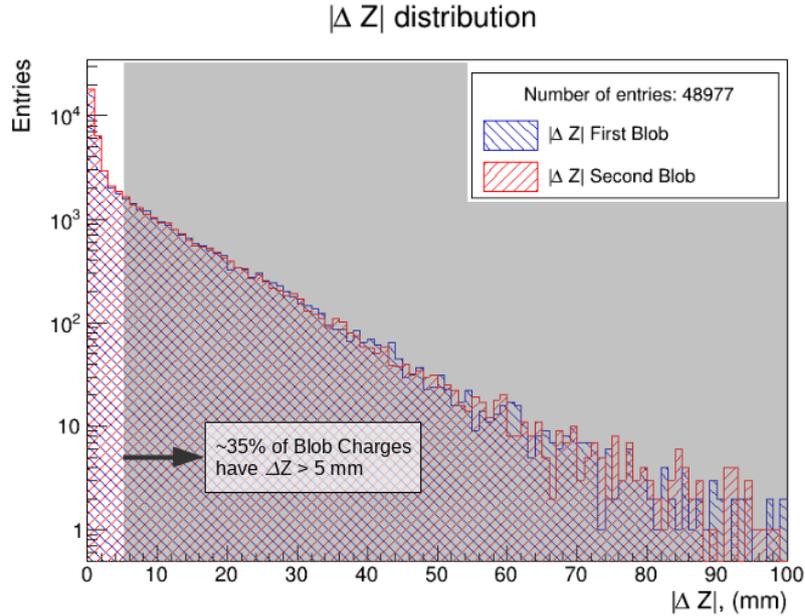


Figure 3.4: Absolute difference between the Z coordinates of the Blob charges on the XZ and YZ projections.

looking at the correlation maps of Z coordinates on the XZ and YZ projections for the first and second Blob charges, as presented in Fig. 3.5, we notice a nice correlation along the first diagonal, but also a strange structure in the shape of a *kite* in the upper right corner. This structure comes from tracks reaching the anode plane at $Z = 600$ mm before the electron comes to a stop. For these, the outcome of the blob-finding becomes meaningless: it often turns out to sit close to $Z = 600$ mm (since this is where the track ends) at times becoming arbitrary.

Let us have a look at the events for which the following condition is satisfied: The Blob

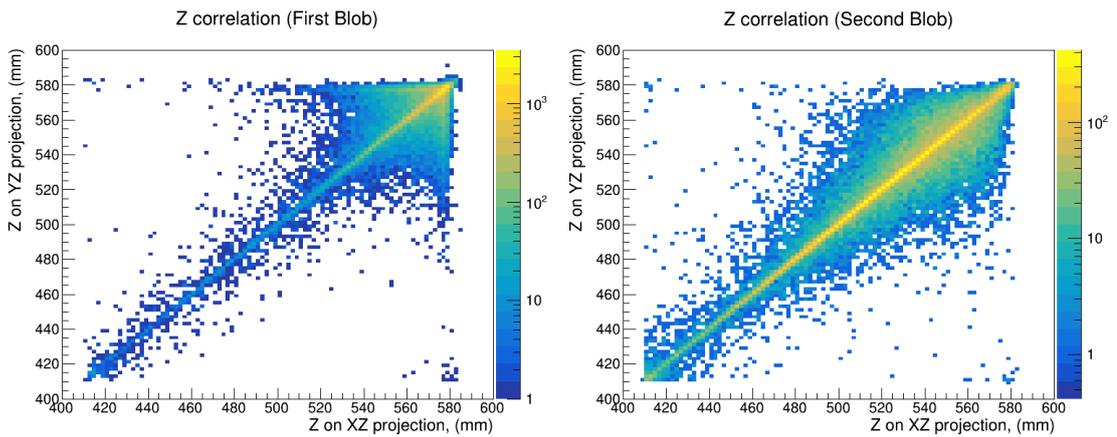


Figure 3.5: (left) Correlation map of the Z coordinates of the first Blob charge on the XZ and YZ projections. (right) Correlation map of the Z coordinates of the second Blob charge on the XZ and YZ projections.

charge Z coordinates on the XZ and YZ projections must be within 5 mm from each other.

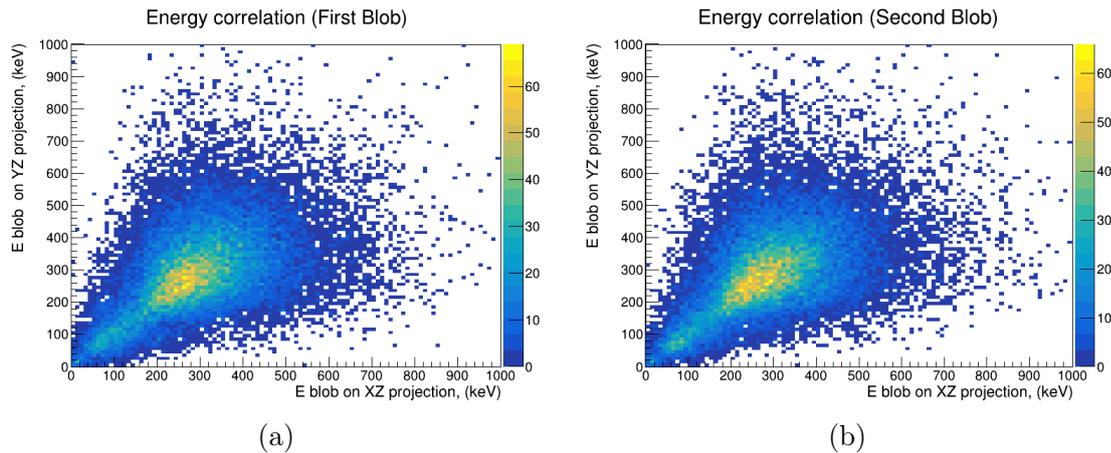


Figure 3.6: Energy correlation maps for the first and second Blob charges on the XZ and YZ projections. The Blob charge radius is 15 mm. The Blob charge Z coordinates on the XZ and YZ projections are within 5 mm from each other.

Plotting the energy correlation maps for the first and second Blob charges on the XZ and YZ projections (where the radius for the Blob charges was selected 15 mm), as presented in Fig. 3.6, even with the selection criteria, the correlation is not obvious at all. Even if we see peaks on the correlation maps around the 300 keV points for the first and second Blob charges on the XZ and YZ projections, the dispersion is large enough to prevent any conclusion.

What if we try to identify the Blob charge more precisely?

The idea is to rely not only on the first and last hits of the track, but try to locate the Blob charge coordinates on the whole track length. Therefore the Blob charge identification process needs to be modified.

3.2.1 Updated Blob Charge Identification Process

In order to identify the Blob charge coordinates more precisely, the Blob charge identification process was modified. First of all, instead of relying on the first and last hit groups of the track, now we parse through all the track hits. Remembering that the number of hits *per* raw event can reach up two thousands, in order to reduce the computation expenses, we set a threshold on energy *per* hit. Now, let's talk about the main principle behind the Blob charge identification process.

3.2.1.1 Primary Blob charge identification principle based on the Maximum Energy Deposition hits

After selecting the track hits that are being considered for the Blob charge identification, we select the hit with the maximum energy deposition. This hit is considered to be the center of the Blob charge. The energy depositions of the track hits within the Blob charge radius are accumulated and considered to be the energy deposition of the Blob charge. The track hits that were used for the Blob charge identification are being removed from the list of the track hits, and the left-out track hits are being considered as a *tail* of the track. The Blob

charge identification process is repeated for the tail until the tail is empty.

At the end of the process, we have a list of the Blob charges, with their coordinates and energy depositions for XZ and YZ projections. The Blob charges are sorted by their Z coordinates so that the first Blob charge is the one with the Z coordinate closer to the anode of the TPC (readout plane, which is located at a distance of 600 mm on the Z axis), the second Blob charge - respectively, with the Z coordinate further away from the anode. At this point, we can select the true Blob charges by applying the criteria to the Z coordinates of the Blob charges. Selecting the Blob charges with the Z coordinates within a defined range from each other, we can obtain the list of the true Blob charges.

The Blob charge identification process is illustrated in Fig. 3.7. However, with such a process, we still rely on the most energetic hit, which might not directly correspond to the Bragg peak of the electron track. Thus, another slightly different approach was introduced.

3.2.1.2 Blob charge identification principle based on the most energetic sphere of the defined radius

Similarly to the previous approach, we select the track hits that pass the energy deposition threshold. Then, for each of the selected hits, we accumulate the energy depositions of the track hits within the Blob charge radius. Comparing the energy depositions of the spheres, we select the sphere with the maximum energy deposition and consider the coordinates of the sphere center to be the Blob charge coordinates. Again, the track hits inside the sphere are being removed from the list of the track hits, and the left-out track hits are being considered as a *tail* of the track. This process is repeated for the tail until the tail is empty. The identification process is illustrated in Fig. 3.8.

Now, we can identify all the potential Blob charges on the track length with these two approaches. One must consider that we may encounter events with unequal numbers of the Blob charges on the XZ and YZ projections, thus containing the *False Blob charges*. This may happen when the track has a kink; therefore, on one of the projections, the kink is identified correctly, while on the other, the kink is identified as the Blob charge. An example of such an event is presented in Fig. 3.9.

To avoid such cases, we apply the criteria on the Z coordinates of the Blob charges coordinates for both projections. The Blob charges with the Z coordinates within the defined range from each other are considered to be the true Blob charges. However, now we became sensitive to the Blob charge radius (Fig. 3.10), and the threshold for the energy deposition of the track hits (Fig. 3.11). In addition to that, we can observe the presence of a tail/artifact on the Energy correlation maps for Blob charges on the XZ and YZ projections passed the selection criteria (Fig. 3.10, Fig. 3.11), that lays in between 0 keV point and the most intense correlation point on the map.

Now, comparing the energy correlation maps for the Blob charges, we notice that by increasing the energy threshold for the track hits and the Blob charge radius, we can reduce the number of Blob charges in the *tail* on the energy correlation maps for both approaches. However, one thing stays unchanged: the broad distribution of the Blob charge energies. In an attempt to reduce the dispersion of the Blob charge energies, each of the Blob charge identification approaches was adjusted.

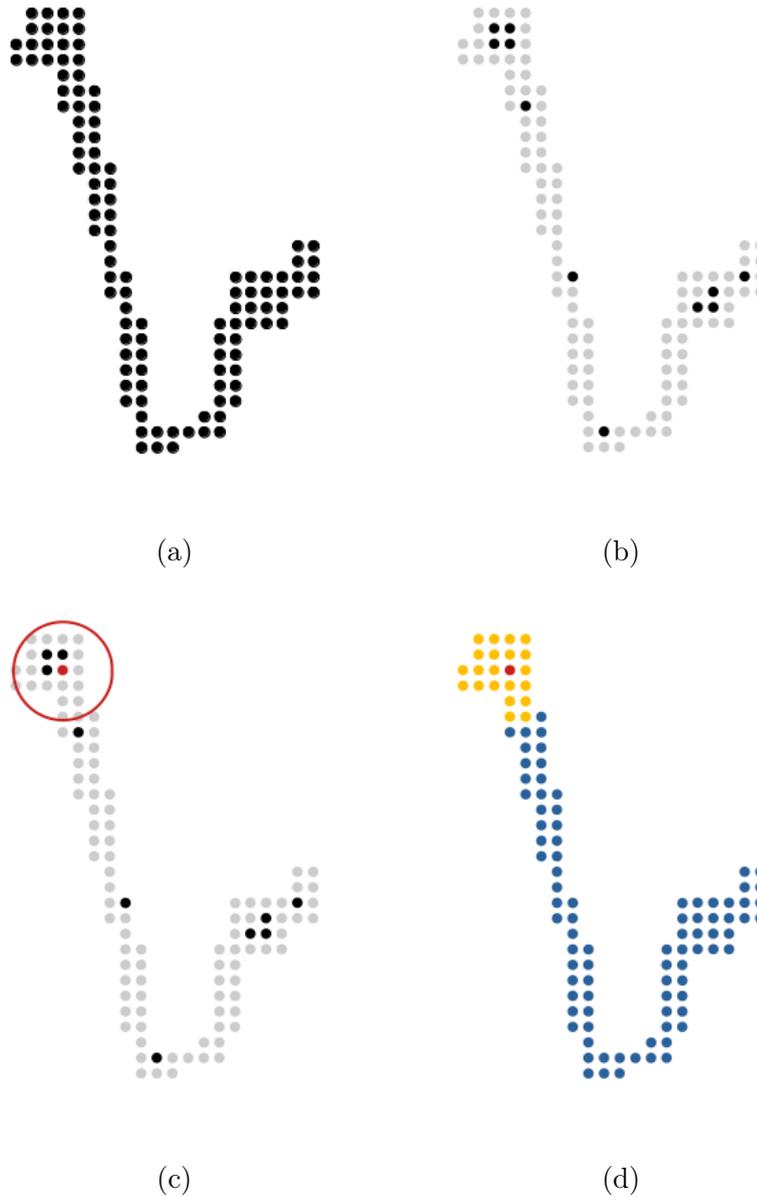


Figure 3.7: Step-by-step illustration of the Blob charge identification process on XZ projection for the NDBD event. (a) All the track hits on the XZ projection. (b) Selection of the track hits after the threshold for the energy deposition. (c) Found the most energetic hit, which coordinates are considered to be the Blob charge coordinates. The energy depositions of the track hits, within the Blob charge radius, are accumulated. (d) The track hits are separated into two groups: Blob charge hits and the tail hits.

The process is repeated sequentially for the tail hits, until the tail is empty.

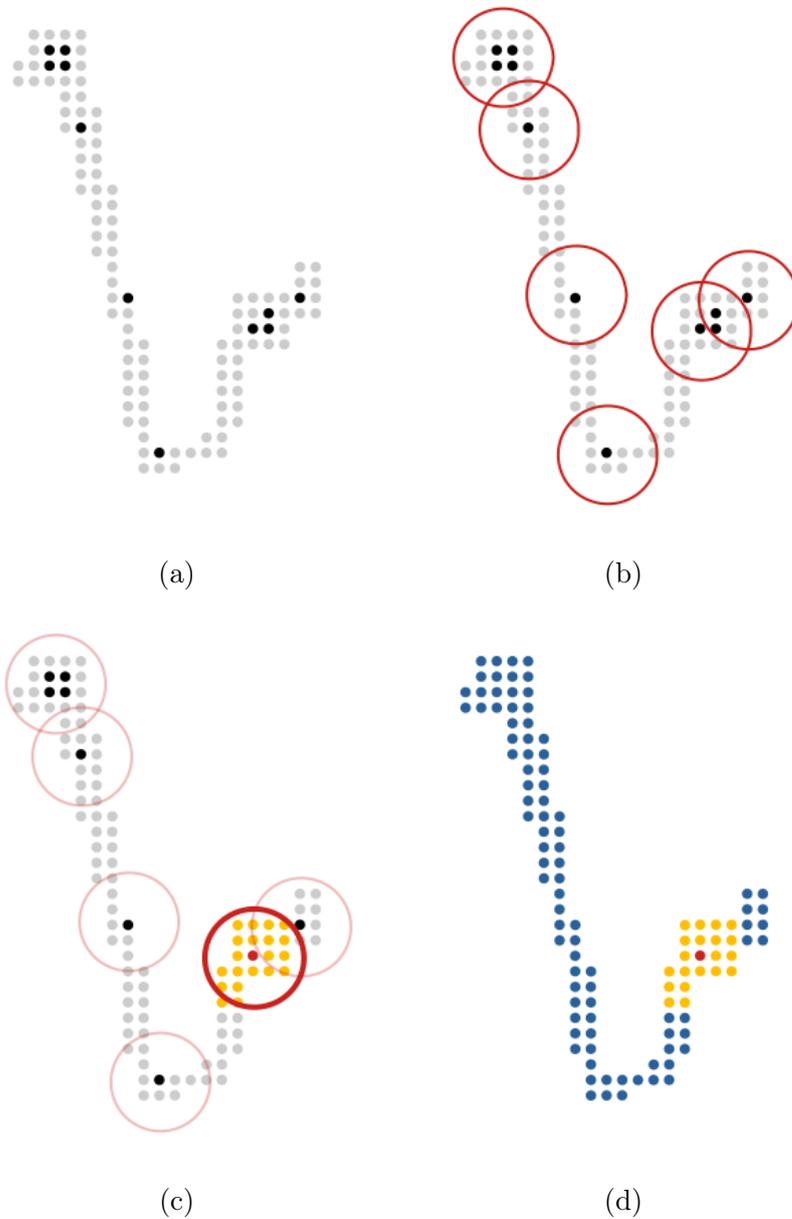


Figure 3.8: Step-by-step illustration of the Blob charge identification process on XZ projection for the NDBD event. (a) Selected track hits on the XZ projection above the energy deposition threshold. (b) For each of the selected hits, we accumulate the energy depositions of the track hits, within the Blob charge radius. (c) Comparing the energy depositions of the spheres, we select the sphere with the maximum energy deposition and consider the coordinates of the sphere center to be the Blob charge coordinates. (d) The track hits inside the sphere are being removed from the list of the track hits and the leftout track hits are being considered as a *tail* of the track. Again the process is repeated sequentially for the tail hits, until the tail is empty.

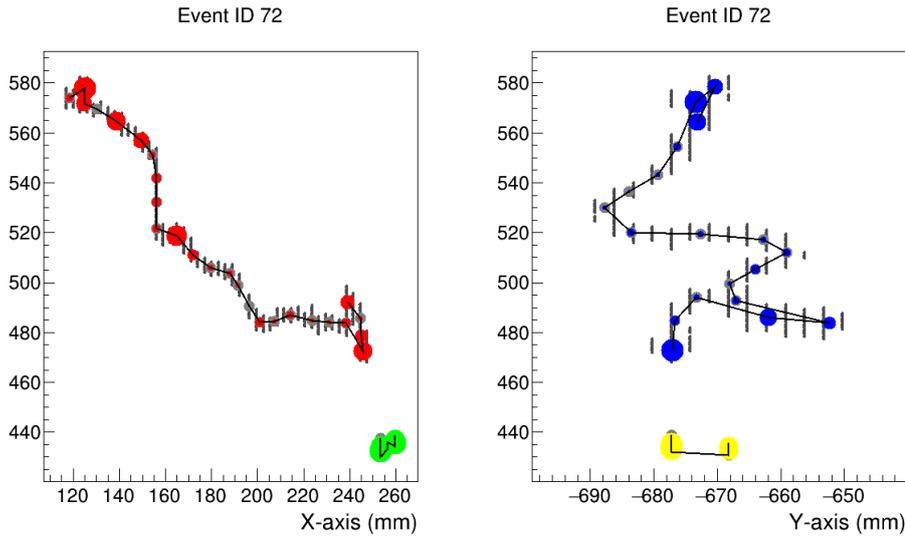


Figure 3.9: Example of the event with the kink in the track. The kink is identified as the Blob charge.

3.2.1.3 Identification of the Blob charge Barycenter

After finding each Blob charge for the Maximum Energy Deposition hits and the Most Energetic Sphere approaches, we calculate the Barycenter of all the track hits within, now considered the Blob charge coordinates. After that, we recalculate the energy deposition of the Blob charge, accumulating the energy depositions of the track hits within the Blob charge radius, centered at the Barycenter (3.12).

After applying the Blob charge coordinates and Energy recalculation using the Barycenter approach, we can observe the higher concentration of the Blob charge energies around the concentration point on the correlation maps for both approaches (Fig. 3.13). For smaller Blob charge radius ($R = 10$ mm), the difference is more significant if we compare Fig. 3.13(low row) and Fig. 3.10(low row). Nevertheless, the dispersion of the Blob charge energies is still significant and sensitive to the Blob charge radius and the energy threshold for the track hits definition.

Table 3.1 presents the results corresponding to the average of the mean energies of the Blob charges for the XZ and YZ projections, obtained using the Maximum Energy Deposition hits and the Most Energetic Sphere approaches (directly and after applying the Barycenter recalculation), for different Blob charge radii and energy thresholds.

Suppose that the threshold is technically an arbitrary parameter, and it has already been seen that increasing the threshold reduces the dispersion of the Blob charge energies. In that case, we can leave it as one degree of freedom and optimize the Blob charge radius instead.

The question now is: *What is the optimal Blob charge radius?*

To answer this question, we need to understand the topology of the electron propagation track inside the gas volume. The electron behavior inside the gas is case-sensitive, depending on the initial energy of the electron, the gas mixture, the pressure, the electric field, etc. In addition to that, the readout system, which in turn defines the spatial resolution of the track hits, detects the ionization electrons produced on the trajectory of the primary electrons that

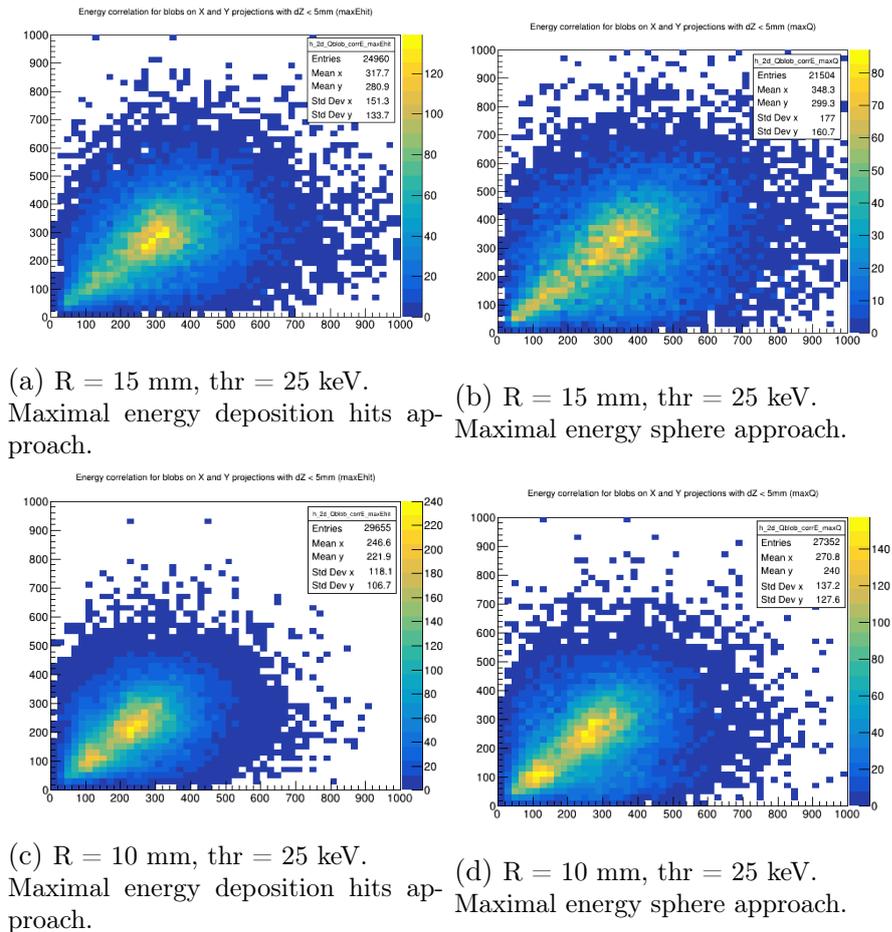


Figure 3.10: Comparison of the energy correlation maps for the Blob charges on the XZ and YZ projections. (a) and (c) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the maximum energy deposition hits. (b) and (d) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the most energetic sphere of the defined radius. The energy threshold for the track hits is 25 keV. The Blob charge radius is 15 mm for (a) and (b) and 10 mm for (c) and (d).

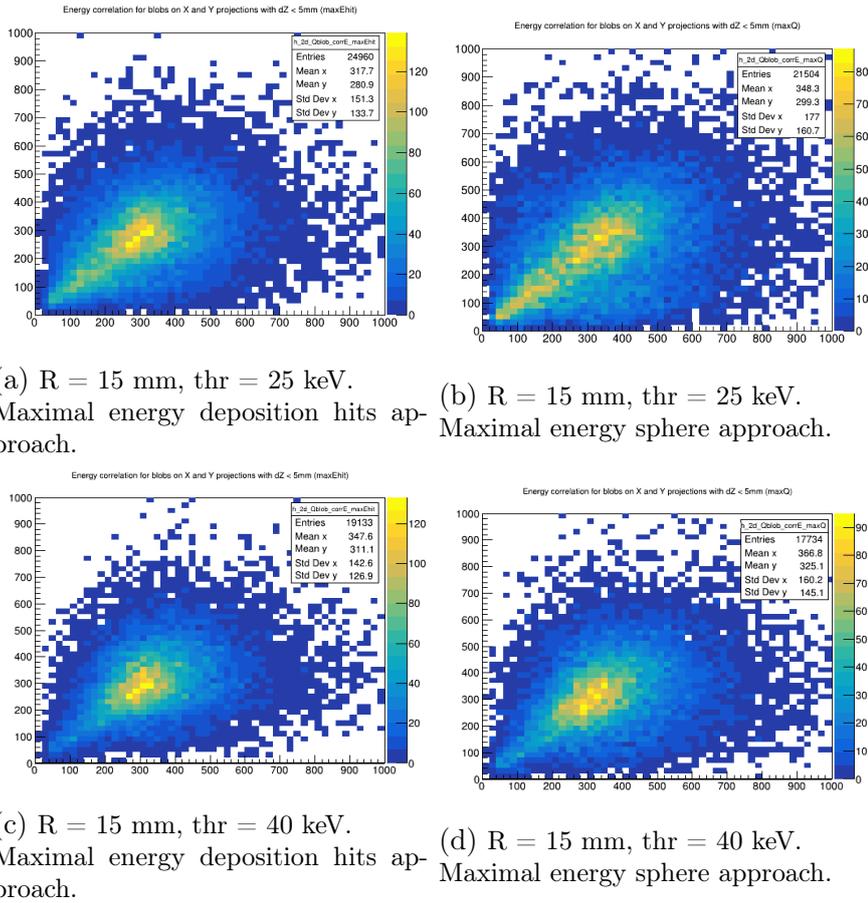


Figure 3.11: Comparison of the energy correlation maps for the Blob charges on the XZ and YZ projections. (a) and (c) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the maximum energy deposition hits. (b) and (d) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the most energetic sphere of the defined radius. The Blob charge radius is 15 mm. The energy threshold for the track hits is 25 keV for (a) and (b) and 40 keV for (c) and (d).

		max Hit, (keV)		max Sph., (keV)	
R (mm)	Thr (keV)	Direct	Bary.	Direct	Bary.
8	15	165.4	193.5	185.35	200.55
	20	185.4	212	205	218.6
	25	201.3	226.6	220.05	232.45
	30	214	237.85	231.4	243.25
	35	224	246.7	239.45	251.4
	40	231.45	253.55	245.55	257.35
10	15	196.55	230.6	220.8	240.2
	20	217.6	249.75	240.8	258.1
	25	234.25	264.6	255.4	271.7
	30	247.25	276.3	266.8	282.5
	35	257.3	285.2	274.65	290.3
	40	265.2	291.85	280.75	296.55
15	15	262.4	316.1	292.45	327.65
	20	282.7	334	310.25	342.7
	25	299.3	347.9	323.8	354.7
	30	312.1	358.75	333.65	364.15
	35	321.75	366.9	340.4	370.85
	40	329.35	373.5	345.95	376.85

Table 3.1: Results from the updated Blob charge identification process. The average of the mean energies of the Blob charges for the XZ and YZ projections, obtained using the Maximum Energy Deposition hits and the Most Energetic Sphere approaches (directly and after applying the Barycenter recalculation). The table shows the results for different Blob charge radii R and energy thresholds for the track hits definition.

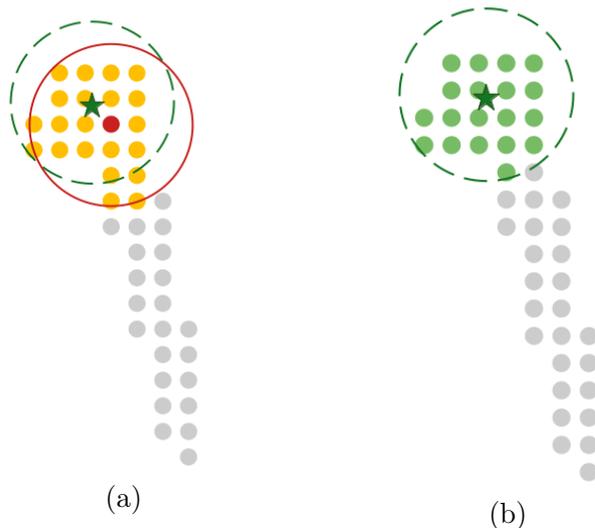


Figure 3.12: Visualization of the Barycenter approach for the Blob charge identification. After calculation of the Barycenter of the track hits within the Blob charge (a), the energy depositions of the track hits, within the Blob charge radius, centered at the Barycenter, are accumulated and now those considered to be the energy deposition of the Blob charge (b).

propagate through the gas before being registered by Micromegas. Therefore, the electron propagation topology should be analyzed in the context of the PandaX-III experiment.

3.3 Electron Propagation Topology

The electron propagation topology was studied using Monte Carlo simulation employing the Geant4 toolkit. The main idea behind the study is to understand at which point of the propagation track the electron *enters the Blob charge stage*. In other words, in which spatial boundaries does the electron experience the increase of the energy deposition and enter the Bragg peak mode, and how can it be associated with the appearance of a Blob charge?

To begin with, the study was simplified by considering only one electron propagation through the gas volume. The gas mixture was defined in accordance with the experimental design: 140 kg of Xe with 1% of TMA. The pressure was set to be 10 bars. The electric field was not included in the simulation, as the propagation of the primary electrons from ionization to the readout plane is not the main focus of the study. With this in mind, the geometry of the TPC was simplified to the gas volume of a cylindrical shape, with the dimensions 1000 mm x 1000 mm x 1000 mm. The gas volume is assigned as a *Logical Volume* of the simulation. The electron was generated at the center of the gas volume with an initial kinetic energy of 2.4 MeV. The particle source origin was restricted to the central point of the gas volume. However, the angular distribution of the electron was set to be isotropic. Such particle source configuration ensures that the electron, propagating through the gas volume, will not reach its boundaries.

To study the electron's energy deposition, we set a high *MaxStep* limit of 450 mm for the *Logical Volume*. This parameter sets the upper limit on the integration step size employed by

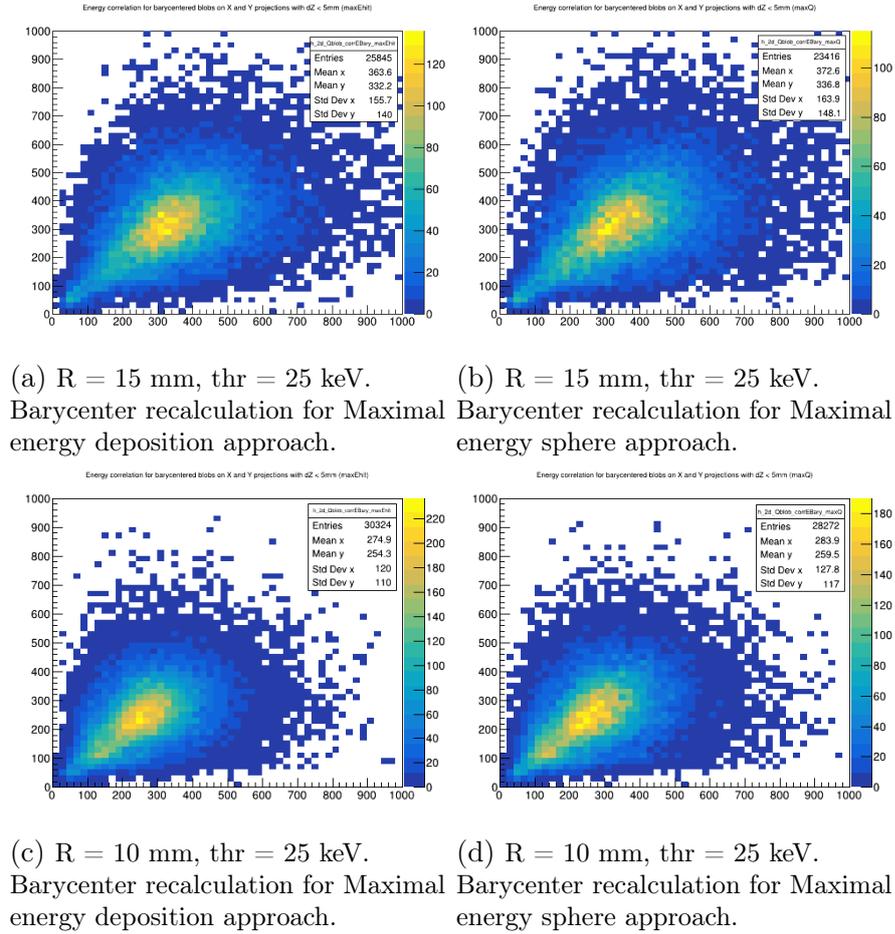


Figure 3.13: Comparison of the energy correlation maps for the Blob charges on the XZ and YZ projections after applying the Barycenter calculation for the Blob charge identification for both Maximum energy deposition and Maximum energy sphere approaches. (a) and (c) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the maximum energy deposition hits. (b) and (d) - Energy correlation maps for the Blob charges on the XZ and YZ projections, where obtained using the most energetic sphere of the defined radius. The energy threshold for the track hits is 25 keV. The Blob charge radius is 15 mm for (a) and (b) and 10 mm for (c) and (d).

Geant4 for approximating particle interactions, meaning that the energy deposition inside the *Logical Volume* is forced by Geant4 if there are no interactions with the gas. Increasing the *MaxStep* limit, we rely instead on the physics simulation limitations applied to the electron propagation. A physics list in Geant4 is responsible for defining the physical processes in the simulation. It specifies which particles are included in the simulation, as well as the models used to describe their interactions with matter. The physics list also controls the energy range over which the simulation is valid. A physics list for the simulation of ionizing electrons is the *G4EmStandardPhysics* list, which was used in the study with the following production energy cuts:

- $MinEnergy = 100 \text{ eV}$
- $MaxEnergy = 1 \text{ GeV}$

Production cuts are used to determine whether a particle can produce secondary particles in a given material. If the energy is within the production cuts range, the engine will allow the particle to produce secondary particles. Thus, we expect that secondary particles will be produced until the electron reaches the energy of 100 eV.

Additionally, specific electromagnetic processes necessitate a threshold to prevent infrared divergence, ensuring that no secondary particles are generated below a certain energy level. Gammas, electrons, and positrons necessitate a production threshold to adhere to this requirement. This threshold is specified as a distance or range cut-off and is internally converted to energy based on the characteristics of individual materials. The range cut-offs for gammas, electrons, and positrons are set to 10 μm . The problem, however, is that the *G4EmStandardPhysics* list is not designed to simulate low-energy interactions with high precision. Therefore, the *G4EmStandardPhysics* was extended with the *G4EmExtraPhysics* list that extends the capabilities of the standard electromagnetic physics list *G4EmStandardPhysics* to include additional processes that are important for simulating a wider range of physical phenomena. *G4EmExtraPhysics* is typically used for applications that require more accurate and realistic simulations of electromagnetic interactions. We made sure that atomic de-excitation effects were included in the simulation by turning on such processes as:

- *Fluorescence physical process* - the process of emission of characteristic X-rays by atoms in excited states, following the de-excitation of inner shell electrons.
- *Auger electron emission process* - the process of emission of Auger electrons by atoms in excited states, after the de-excitation of inner shell electrons.
- *PIXE process* - the process of emission of protons by atoms in excited states, after the de-excitation of inner shell electrons.

After setting the environment for the simulation, let's look at the values, which we are interested in.

At each interaction point (*step*) of the electron with the gas atoms, we collect following observables:

- E_{kin} - kinetic energy of the electron before the interaction
- E_{dep} - energy deposition of the electron at the interaction point

- E_{loss} - energy deposition of the electron at the interaction point plus the kinetic energy of the produced secondary particle
- *Step length* of the electron - the distance between the interaction points
- *Deposition point coordinates* - coordinates of the interaction point
- θ *angle* - angle between the electron momentum vectors before and after the interaction point, $\theta = \arccos \frac{\vec{p}_{\text{init}} \cdot \vec{p}_{\text{fin}}}{|\vec{p}_{\text{init}}| |\vec{p}_{\text{fin}}|}$
- *Secondary particle type* produced at the interaction point
- $E_{\text{kin secondary}}$ - kinetic energy of the secondary particle

In case the secondary particle is not *gamma*, but *e-* that has kinetic energy lower than the threshold of 100 keV, we do not consider the secondary particle as produced and sum its kinetic energy to the E_{dep} .

Knowing the starting point of the electron propagation and the coordinates of the interaction points, we can calculate the variable L_n , which is the distance between each interaction point n and the **end** of the electron propagation track. The **end** of the electron propagation track is defined as the point where the electron kinetic energy is below the threshold of 100 eV. The total number of interaction points is N , where the beginning of the electron propagation track is $n = 1$, and the end is $n = N$. The visualization of the electron propagation track is presented in Fig. 3.14.

After simulating and processing 10^6 events, the following distribution for E_{loss} , normalized by the number of events, with respect to the L was obtained (Fig. 3.15). Energy loss increases drastically at the distance L of 5 mm to the end of the electron propagation track. This is the point where the electron enters the Bragg peak mode.

On the other hand, we compute the electron **Stopping power** dE/dx numerically as the division of the E_{loss} by the *Step length* for each interaction point. The dE/dx distribution with respect to the E_{kin} shows the rapid increase of the stopping power after the electron kinetic energy reaches the value of 100 keV (see Fig. 3.16). Yet, this is only in accordance with the losses of the primary electron energy (see Fig. 3.15).

Finally, if we look at the distribution of angle θ to the L (Fig. 3.17), we can observe that for L smaller than 20 mm, the electron enters a regime where θ covers the whole angular domain. In this regime, the electron is undergoing heavy multiple scattering. This is expected, as the electron is supposed to have low kinetic energy near the end of the track. However, the multiple scattering begins much earlier than the electron enters the Bragg peak mode, distorting the definition of the Blob charge. Thus, the effect of the multiple scattering is significant and should be considered.

The study in the context of the PandaX-III experiment shows that the Blob charge identification is sensitive to the scattering angle of the electron. A much more detailed study of the electron propagation topology is required.

However, from the rough estimation of the θ angle distribution, the Blob charge could be identified in the last 20 mm of the electron propagation track. If we define the Blob charge radius to be 10 mm, we can expect that the Blob charge would contain energy in the range of 200 - 300 keV, as can be seen from the results in Table 3.1. A better determination

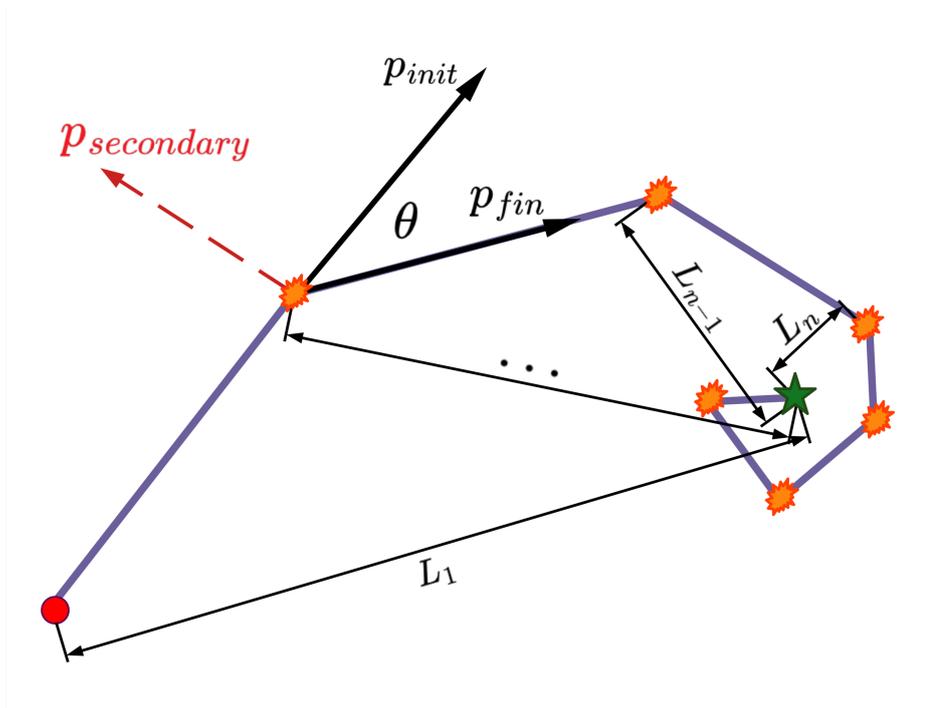


Figure 3.14: Visualization of the electron propagation track in Geant4. At each interaction point, the particle kinetic energy, energy deposition, step length, θ angle, kinetic energy of the secondary particle and its type are being collected. The distance L to the end of the electron propagation track is calculated as well.

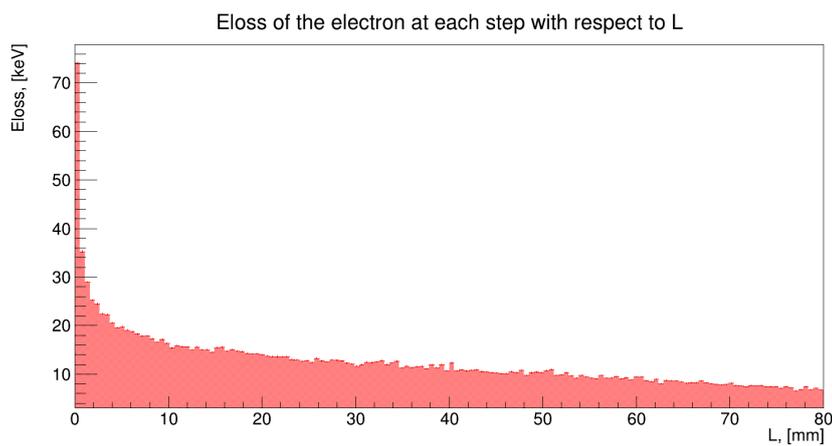


Figure 3.15: Energy loss of the electron, normalized by the number of electron events, with respect to the distance to the end of the electron propagation track. The rapid increase of the energy loss of the electron is observed at the distance of 5 mm to the end of the electron propagation track.

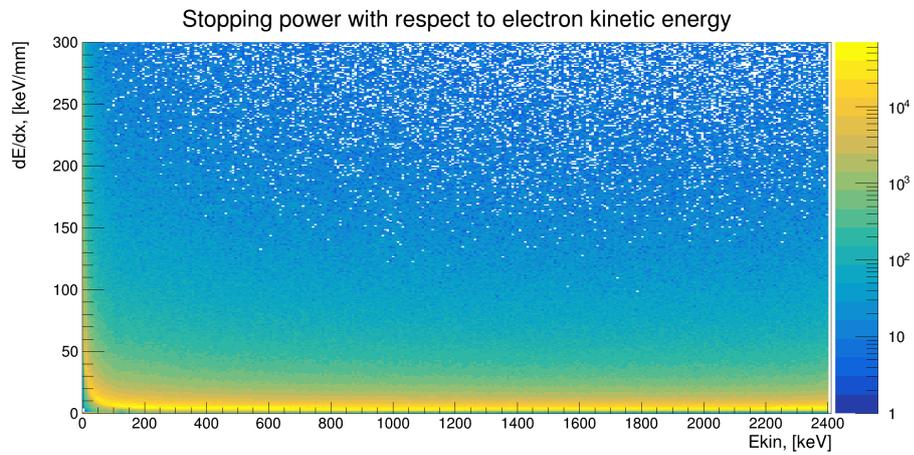


Figure 3.16: Stopping power of the electron with respect to the electron kinetic energy.

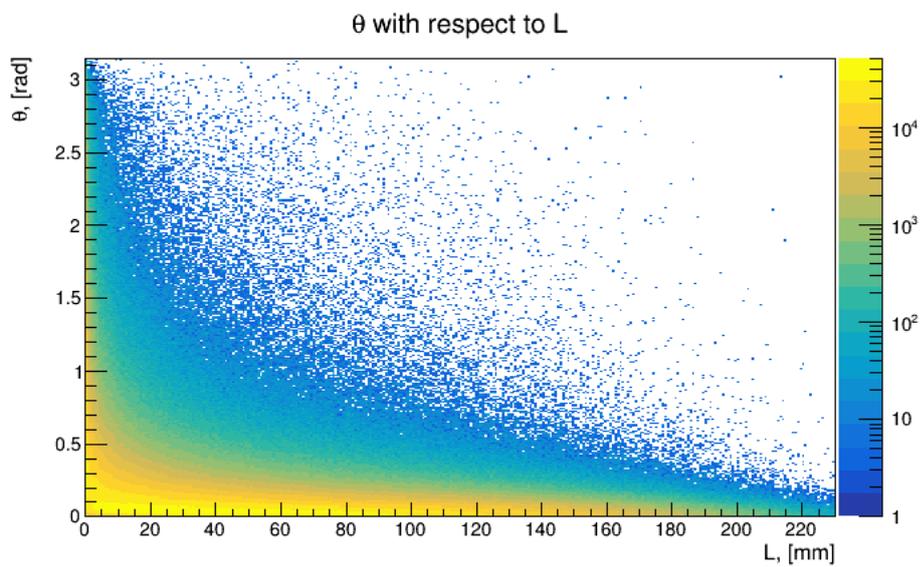


Figure 3.17: Angle θ between the electron momentum vectors before and after the interaction point, with respect to the distance to the end of the electron propagation track.

of the energy threshold for the track hits is required to reduce the dispersion of the Blob charge energies. Nevertheless, in the context of $0\nu\beta\beta$ events, if one of the Blob charges is lost due to the dead channels, this would result in the 8-12% energy loss of the event. Such events would fall from the ROI defined for the PandaX-III experiment and be considered background events.

3.4 Conclusions

The results presented in the chapter are not as conclusive as we were hoping for in the introduction. The possibility that the part of the blob energy that would have been lost due to missing channels in XZ could be directly reconstituted from YZ and *vice versa* was not confirmed. The ideas around the correlation between twist and blob did not come to fruition either.

Yet several observations have been made that are important for the last part of this thesis, dedicated to Machine Learning (ML). Firstly, the very lack of success of the direct approaches investigated here is yet another incentive to turn to ML. Secondly, we have seen that the onset of twist is much earlier than that of blob. Which means that a large fraction of the track topology contains information about the track energy content. Which is in turn promising for the ML approaches, which are to deal with the whole trajectory and not only its ends.

Last, we have understood the importance of the extra physics processes enabled by the *G4EmExtraPhysics* physics list for the correct simulation of our events.

Chapter 4

Introduction to Machine Learning and Neural Networks

Sapiens dominabitur astris
(Latin: the wise man shall master the stars)
— Claudius Ptolemy

4.1	Introduction	77
4.2	Principles of Machine Learning	78
	4.2.1 Loss and Metrics in Supervised Learning	79
	4.2.2 Neural Networks and Deep Neural Networks	82
4.3	Convolutional Neural Networks	89
4.4	Conclusion	92

4.1 Introduction

What is Intelligence?

Intelligence can be characterized as the capacity to acquire knowledge and employ effective strategies to address challenges and attain objectives, adapted to the circumstances in a dynamic and unpredictable environment.

In 1955, John McCarthy, a computer scientist and esteemed Stanford Professor, coined the term "artificial intelligence" (AI)[76]. He described AI as "the science and engineering of making intelligent machines." This introduced a new focus on machines resembling humans in their cognitive behaviors to some extent. McCarthy, widely recognized as the father of AI, organized the famous Dartmouth conference in the summer of 1956, which is widely considered to be the official birthdate of AI.

In 1959, Arthur Samuel, a pioneer in the field of AI, defined a term *Machine Learning* (ML) as a "field of study that gives computers the ability to learn without being explicitly programmed" [77]. Today, we can expand the definition of ML as a field of computer science and a part of AI that uses statistical techniques to give computer systems the ability to "learn" (i.e., progressively improve performance on a specific task) without being explicitly

programmed. From the 1950s to the 1980s, the development of ML has brought forth algorithms such as the perceptron [78], the nearest neighbor [79], and the decision trees [80]. However, the low-level single-layer algorithmic models could not solve complex problems while being computationally limited by the hardware of the time [81].

With the development of more complex architectures, including the Convolutional Neural Networks and Recurrent Neural Networks [82], application of backpropagation algorithm [83] to the training of the neural networks, increase of computational power, and the availability of vast datasets, the ML field has experienced a renaissance. It has become a powerful tool in various scientific disciplines.

Nowadays, ML is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or unfeasible; example applications include email filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR), learning to rank (LTR), and computer vision.

In the last decade, ML methods have complemented statistical analysis in physics, which experienced a data revolution. In particle physics, ML methods have been widely used for multiple purposes, such as beam dynamics studies [84], event selection [85], event processing [86], background rejection [87], particle identification [88], and many others. Indeed, it has been shown that ML methods can outperform traditional analysis techniques in many cases, and with the development of new algorithms, the gap is expected to widen.

However, applying machine learning methods requires careful consideration, as incorrect modeling choices or assumptions may lead to misleading or incorrect conclusions [89]. Considering the data limitations and potential algorithmic risks, one must interpret ML models cautiously to draw accurate and meaningful reasoning. However, let us start with the basics.

4.2 Principles of Machine Learning

The primary objective of machine learning (ML) is to recognize valuable patterns within data. In physics, ML finds application in addressing complex problems associated with vast datasets [89]. As an example, in experiments conducted at the Large Hadron Collider (LHC) at CERN, where the collisions of protons or ions at various points along the circular collider result in the generation of tens of thousands of petabytes yearly. The abundance of collisions makes distinguishing between significant and insignificant data samples essential. ML methods, in this context, play a pivotal role in the discrimination of background from the signal and contributing to the exploration of novel fundamental physics [90].

Machine learning (ML) can be categorized into three primary groups: *a) supervised learning*, *b) unsupervised learning*, and *c) reinforcement learning*. In *supervised learning*, the model learns from labeled data to make predictions or classifications [91]. *Unsupervised learning*, on the other hand, uncovers patterns and structures within unlabeled data [92]. Finally, *reinforcement learning* is centered on an agent learning optimal behaviors through interactions with an environment and receiving rewards [93].

It is the *Supervised Machine Learning* approach that is being used in my PhD project; thus, the introduction of ML techniques will be focused primarily on it.

4.2.1 Loss and Metrics in Supervised Learning

Supervised learning entails instructing a model to conduct a specific task using a provided set of examples, the inputs [92]. Initially, the input dataset is divided into two distinct sub-sets for training and testing. The training dataset is utilized to train an ML algorithm (during the so-called training phase), allowing it to discern patterns within the data. Subsequently, the trained algorithm is employed to predict new observations on the test dataset. The dataset comprises a collection of examples represented by vectors $\{x_i, y_i\}$, where $i = 1, 2, \dots, n$. Here, x_i denotes the set of attributes for the i_{th} example, and y_i is the **target variable** intended for prediction, which may be a singular value or a vector.

If the target variable y is a continuous set of values ($y \in \mathbb{R}$), the problem is called a **regression problem**. A simple example could be the prediction of the mass of a particle (target variable y_i), where the attributes x_i may be the measured momenta of its decay product and the hits in the detection system.

On the contrary, when y represents a discrete number of classes or labels ($y \in \{C_1, C_2, \dots, C_k\}$, where C_j is a class), it is called a **classification problem**. As an example, let us consider the particle identification problem. The attributes x_i may be the measured energy, momentum, and the number of hits in the detector, while the target variable y_i may be the particle type.

Machine learning algorithms strive to establish a model that can be expressed as f that effectively maps input attributes x_i to the target values y_i . This can be denoted as:

$$y \Leftrightarrow \hat{y} = f(x, \omega) + \epsilon, \quad (4.1)$$

where \hat{y} is the value predicted by the model, ω represents the model's parameters (also denoted as weights and biases), and ϵ is the random error term ($\langle \epsilon \rangle = 0$ and $\langle \epsilon_i \epsilon_j \rangle = 0$ where $i, j = \{0, 1, 2, \dots, n\}$ and n is a number of elements in the training dataset). The function denoted as $f(x, \omega)$ can be considered as the data-generating function, and a primitive example of one could be the linear function $f(x, \omega) = \omega_1 x + \omega_0$ used in the simple linear regression fit, depicted in Fig. 4.1. Here, ω_1 is the weight (the slope of the line), and ω_0 is the bias (the intercept of the line). Considering this, the primary objective of the ML algorithm is

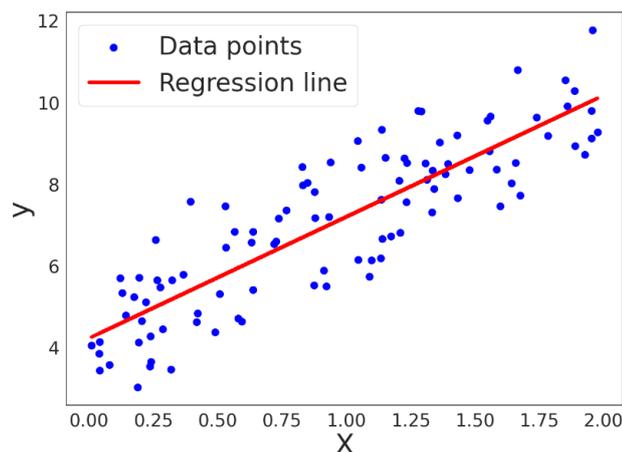


Figure 4.1: Example of the linear regression fit.

to estimate the parameter set ω . This estimation enables accurate predictions of the target

variable y for new data points to the best possible extent. In order to achieve this, one must have a mathematical expression that quantifies the model's performance and guides the fine-tuning of the model's parameters. Therefore, we can define the **Loss function** as a function that quantifies the difference between the predicted value \hat{y} and the actual (true) value y in the training dataset. We denote the loss function as $L(y, \hat{y})$, a function of the model's parameters ω . The mean loss of the predictor on the training set is referred to as the **Cost function** or **Empirical risk** in decision theory [94]:

$$C(\omega) = \frac{1}{n} \sum_{i=1}^n L(y_i, \hat{y}_i(\omega)). \quad (4.2)$$

For the regression problems, the most common loss functions are the mean squared error (MSE) and mean absolute error (MAE) [94]. The MSE is defined as:

$$C_{MSE}(\omega) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i(\omega))^2, \quad (4.3)$$

while the MAE is defined as:

$$C_{MAE}(\omega) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i(\omega)|. \quad (4.4)$$

For classification tasks, the selection of cost functions depends on the particular algorithm and objective at hand. Outputs of a classifier network are usually represented as probabilities of the input belonging to a particular class. These probabilities can be checked against some threshold for the classification to yield a yes-or-no answer. Common examples encompass the 0-1 loss function and cross-entropy loss [94]. The 0-1 loss function measures the misclassification rate within the training set. The cost function returns the fraction of misclassified training samples:

$$C_{0-1}(\omega) = \frac{1}{n} \sum_{i=1}^n I(y_i \neq \hat{y}_i(\omega)), \text{ where } I(y_i \neq \hat{y}_i(\omega)) = \begin{cases} 1, & \text{if } \hat{y}_i(\omega) \neq y_i \\ 0, & \text{otherwise.} \end{cases} \quad (4.5)$$

Similarly, the cross-entropy loss function measures the performance of a classification model whose output is a probability value between 0 and 1. It increases as the predicted probability diverges from the actual target label. The cost function for the binary classification is defined as:

$$C_{CE}(\omega) = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m y_{ij} \log \hat{y}_{ij}(\omega), \quad (4.6)$$

where m is the number of classes.

When the number of classes is only equal to two, the cross-entropy loss function is referred to as the binary cross-entropy loss function and the 4.6 is simplified to:

$$C_{CE2}(\omega) = -\frac{1}{n} \sum_{i=1}^n [y_i \log \hat{y}_i(\omega) + (1 - y_i) \log(1 - \hat{y}_i(\omega))]. \quad (4.7)$$

Apart from the loss function, which provides an insight into the model's performance, the **metrics** are also used to evaluate the model's performance. The metrics are the functions

that quantify the model’s performance while not influencing its training - they do not affect the model’s parameters. The metrics are used to evaluate the model’s performance on the validation or test dataset.

The choice of the metrics depends on the particular problem and the model’s objective. A valuable metric for classification problems is the receiver operating characteristic (ROC) curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various thresholds[95]. An ideal classifier would have a TPR of 1 and an FPR of 0, resulting in a curve near the upper left corner. The Area Under the ROC Curve (AUC-ROC) measures the model’s ability to distinguish between classes; a higher AUC indicates better performance. One can see an example of the ROC curve in Fig. 4.2.

Another common metric used in classification problems is *accuracy*, which is defined

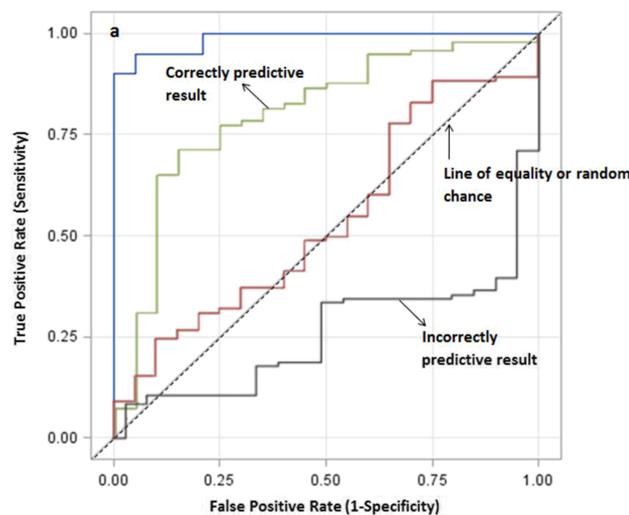


Figure 4.2: Example of the ROC curve. The dashed line represents the random classifier, while solid lines represent different classifiers. The most effective classifier is the blue one, which has the largest AUC-ROC closer to 1. Adapted from [96].

as the ratio of the number of correct predictions to the total number of predictions. The accuracy metric is calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (4.8)$$

To optimize the performance of the network, we have to minimize the cost function. This is the primary objective of the training process:

$$\omega^* = \operatorname{argmin}_{\omega} C(\omega) = \operatorname{argmin}_{\omega} \frac{1}{n} \sum_{i=1}^n L(y_i, \hat{y}_i(\omega)). \quad (4.9)$$

The training process is an iterative process that aims to find the optimal set of parameters ω^* that minimizes the cost function. Multiple algorithms can be used to achieve this. Their effectiveness depends on the particular characteristics of the problem [97]. Such algorithms rely on different concepts [92], such as probability theory (e.g., naive Bayes and logistic regression), decision theory (e.g., decision trees), information theory (e.g., neural networks), optimization theory (e.g., support vector machines) and ensemble methods (e.g., random

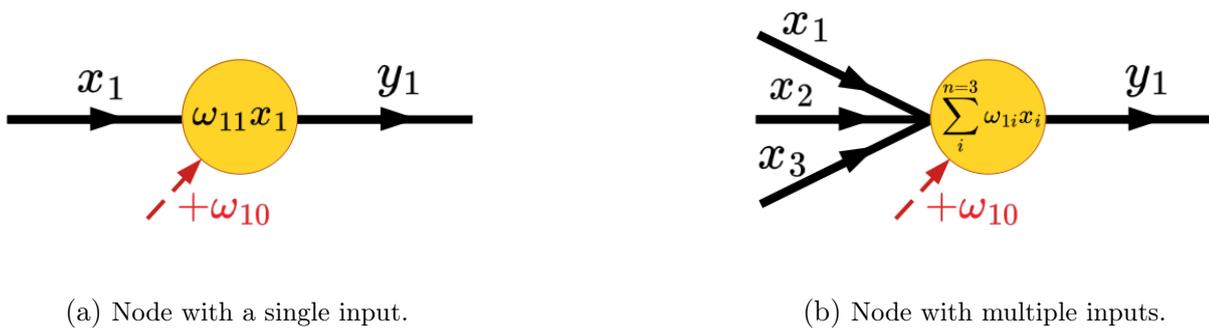


Figure 4.3: Node definition, where x_i are the inputs, ω_{ji} are the weights, ω_{j0} is the bias term, and $y_j = \sum_i^{n=3} \omega_{ji}x_i + \omega_{j0}$ is the output from the node.

forests).

To not encumber the reader with the details regarding all the algorithms, I will focus on the ones that is used in my PhD project - the *neural networks*.

4.2.2 Neural Networks and Deep Neural Networks

Neural Networks (NN) are a class of machine learning algorithms that are inspired by the biological neural networks that constitute animal brains. Such systems learn to perform tasks by considering examples without being programmed with task-specific rules. Precisely, they are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns and were found to be efficient in solving *nonlinear* problems.

The term's modern usage often refers to artificial neural networks, which are composed of artificial neurons or nodes. Thus, the term *neural network* is synonymous with *Artificial neural network* (ANN). The most common type of ANN is the *Feedforward neural network* (FNN), where the information flows in only one direction - forward - from the input nodes to the output nodes. Let us look closer at the NN description, using the FNN as an example.

The *node* is the key element of the NN. It is a mathematical function that receives one or a set of inputs, performs a *transformation* on them, and produces a single output. A transformation performed by the node might be a simple linear transformation, such as a weighted sum of inputs plus a bias term like is depicted in Fig. 4.3a for a single input and in Fig. 4.3b for three inputs. Further on, the bias term will be omitted from the node visualization for the sake of simplicity.

We can see that Fig. 4.3b represents a linear operation on the inputs. Thus, this one node already depicts a simple linear model. Fig. 4.4 encapsulates this model architecture, where the row of blue nodes represents only inputs and is called the *input layer* and the yellow node represents the *output layer*. If we add an additional layer between the input and output layers, we will get the *hidden layer*. For the hidden layer, the nodes can take the previous layer's outputs as inputs and perform a transformation on them, having unique weights and biases. Fig. 4.5 depicts a linear model with a single hidden layer, with the corresponding nodes having an orange color. At this point, it is worth mentioning that the layers in the neural network for which the nodes connected to all the nodes in the previous layer are called *fully connected layers* or *dense layers*. If we look at the output this

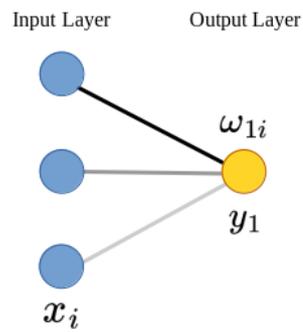


Figure 4.4: Linear model with a single node.

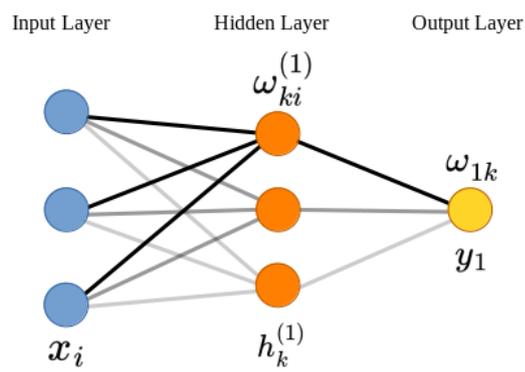


Figure 4.5: Linear model with a single hidden layer.

model architecture produces, we can see that it is still a linear function of the inputs:

$$y_1 = \sum_k \omega_{1k} h_k^{(1)} + \omega_{10} = \sum_k \omega_{1k} \left(\sum_i \omega_{ki}^{(1)} x_i + \omega_{k0}^{(1)} \right) + \omega_{10} \quad (4.10)$$

$$= \sum_{i,k} \omega_{1k} \omega_{ki}^{(1)} x_i + \sum_k \omega_{1k} \omega_{k0}^{(1)} + \omega_{10}, \quad (4.11)$$

We may add additional hidden layers to the model and change the number of nodes, but the output will still be a linear sum of the inputs. This sum won't effectively model the non-linear dependencies in the data.

So, we need to introduce the non-linearity to the model. This can be done by applying a non-linear **activation function** to the node (Fig. 4.6), after the transformation is performed on the inputs by the hidden layer's nodes. Now, as the non-linearity is introduced, it's obvious

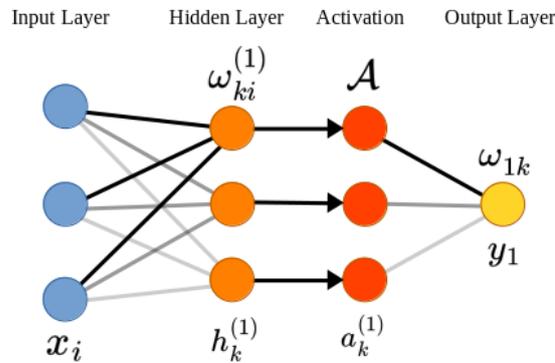


Figure 4.6: Linear model with a single node and an activation function.

that the output of the model is no longer a linear function of the inputs:

$$y_1 = \sum_k \omega_{1k} a_k^{(1)} + \omega_{10} = \sum_k \omega_{1k} \mathcal{A} h_k^{(1)} + \omega_{10} \quad (4.12)$$

$$= \sum_k \omega_{1k} \mathcal{A} \left(\sum_i \omega_{ki}^{(1)} x_i + \omega_{k0}^{(1)} \right) + \omega_{10}, \quad (4.13)$$

where \mathcal{A} is an activation function. Now, with the activation function, we can add more layers to impact the model's output more. Applying non-linearities on top of non-linearities, we can model more complex functions to map very complicated relationships between the inputs and the outputs. We can generalize the Feedforward NN with a set of layers $l = 1, 2, \dots, L$ as a combination of the linear transformation and the activation function:

$$\begin{aligned} \hat{y} = f(x, \omega) &= a^{(L)} \left(\omega^{(L)} a^{(L-1)} (\dots a^{(1)} (\omega^{(1)} x + \omega_0^{(1)}) \dots) \right) \\ &= a^{(L)} \left(h^{(L)} (\dots a^{(1)} (h^{(1)}(x)) \dots) \right). \end{aligned} \quad (4.14)$$

where indices are omitted, but $\omega^{(l)}$ represents the weights and biases $\omega_{ij}^{(l)}$ applied to input j of the neuron i in the layer l , $a^{(l)}$ is the activation function applied to the output of the layer l , and $h^{(l)}$ is the transformation performed by the layer l .

One must pay particular attention to the choice of activation function, as it may significantly impact the model's performance. However, this will be discussed later. For now, the basic building blocks of the neural network are defined, and we can proceed *deeper* into the ML types.

The main focus of the last part of my PhD project is related to the *Deep Neural Networks* (DNN). DNNs are a class of ANNs with more than one hidden layer between the input and output layers. DNNs have been widely used in such areas as computer vision, speech recognition, natural language processing, etc. [98]. These networks can learn sparse representations of data and are capable of discovering intricate structures within high-dimensional datasets.

4.2.2.1 Gradient Descent and Backpropagation

As we have already discussed, the loss function is determined by parameters ω of the model, the inputs and expected outputs, see Eq. 4.2. To find the optimal set of parameters ω^* that minimizes the cost function, as in Eq. 4.9, we may quickly run into the problem of the high-dimensional space of the parameters. The optimization problem becomes more complex as the number of parameters increases, which can rise to the order of several millions in the case of the DNNs.

However, instead of fitting the model's parameters by hand, we can use the numerical optimization algorithms through a large number of evaluations, aiming to develop a NN that performs well on unseen data. This is why the training of the NNs requires a large number of data.

The most common and, in the meantime, the most efficient optimization algorithm to solve the optimization problem is the *Gradient Descent*. The gradient descent algorithm begins by calculating the gradient of the loss function at a specific input point. During this process, the network's inputs and expected outputs remain constant while the gradient is computed for each trainable weight. As the gradient represents the direction of maximum change in the function, we can use this gradient information to adjust the weights in the direction of the steepest descent, gradually bringing the network closer to the minimum value of the loss function. Following the t -th training step, each weight can be updated as follows:

$$\begin{aligned}\omega^{(t)} &= \omega^{(t-1)} - \alpha \frac{1}{n} \nabla C(\omega^{(t-1)}) \\ &= \omega^{(t-1)} - \alpha \frac{1}{n} \frac{\partial C(\omega^{(t-1)})}{\partial \omega},\end{aligned}\tag{4.15}$$

where $\omega^{(t)}$ is the set of weights at the iteration t , α is the *learning rate* that controls the step size of the gradient descent and is a hyperparameter that needs to be tuned to decrease or increase the importance of the update following the direction opposite to the gradient. The gradient with respect to a given weight ω_{ij} at a given iteration t for a given layer l is

calculated using the chain rule, taking into account equation 4.14:

$$\begin{aligned}
 \frac{\partial C(y, \hat{y}(\omega))}{\partial \omega_{ij}^l} &= \frac{\partial C}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \omega_{ij}^l} \\
 &= \frac{\partial C}{\partial \hat{y}} \frac{da^{(L)}(h^{(L)})}{dh^{(L)}} \frac{\partial h^{(L)}}{\partial \omega_{ij}^l} \\
 &= \frac{\partial C}{\partial \hat{y}} a'^{(L)}(h^{(L)}) \omega_{ij}^L \frac{\partial h^{(L-1)}}{\partial \omega_{ij}^l} \\
 &= \frac{\partial C}{\partial \hat{y}} a'^{(L)}(h^{(L)}) \omega_{ij}^L \dots a'^{(l)}(h^{(l)}) \frac{\partial \omega_{ij}^l}{\partial \omega_{ij}^l} h^{(l-1)} \\
 &= \frac{\partial C}{\partial \hat{y}} a'^{(L)}(h^{(L)}) \omega_{ij}^L \dots a'^{(l)}(h^{(l)}) h^{(l-1)},
 \end{aligned} \tag{4.16}$$

where $a'^{(l)}(h^{(l)})$ is the derivative of the activation function with respect to the output $h^{(l)} = \omega^{(l)} a^{(l-1)} + \omega_0^{(l)}$ of the hidden layer l , and $a^{(l)}$ is the output after application of the activation function at a layer l . The layer L corresponds to the output of the network, thus $a^{(L)}(h^{(L)}) = \hat{y}$.

To not calculate the gradient for each weight in the network by hand, the **backpropagation algorithm** is used [83]. The backpropagation algorithm improves the efficiency of the gradient descent: we can calculate the outputs and the derivatives for each layer when evaluating the loss function and then store them to extract the gradient of the cost function with respect to a specific weight by using matrix multiplication:

$$\begin{aligned}
 \frac{\partial C}{\partial \omega^{(L)}} &= \frac{\partial C}{\partial \hat{y}} \frac{da^{(L)}(h^{(L)})}{dh^{(L)}} \frac{\partial h^{(L)}}{\partial \omega_{ij}^l} \\
 &= \delta^{(L)} h^{(L-1)}, \\
 \frac{\partial C}{\partial \omega^{(L-1)}} &= \delta^{(L)} \omega^{(L)} a'^{(L-1)}(h^{(L-1)}) h^{(L-2)} \\
 &= \delta^{(L-1)} h^{(L-2)}, \\
 &\dots \\
 \frac{\partial C}{\partial \omega^{(l)}} &= \delta^{(l)} h^{(l-1)}.
 \end{aligned} \tag{4.17}$$

The $\delta^{(l)}$ is called an **error term** for the layer l . We can see that first, we calculate the last layer L , then propagate the error backward through the network for each previous layer. It is worth mentioning that the backpropagation algorithm, which includes gradient-based optimizations, requires the activation and loss functions to be differentiable. A more detailed explanation of the backpropagation algorithm on the example of the simple neural network is provided in the Appendix B.

It is important to mention that the cost function is computed on the whole training dataset in the basic gradient descent algorithm (also called **batch gradient descent**). Here, the *batch* refers to the total number of training examples used in a single iteration of the gradient descent algorithm. There are other types of gradient descent algorithms, such as **Stochastic Gradient Descent** (SGD), where the cost function is calculated on a single training element; thus batch size is equal to one, and **Mini-batch Gradient Descent** (MBGD),

where the cost function is calculated on a subset of elements in the training dataset, therefore batch size is greater than one and less than the total number of training examples. Thus, MBGD compromises the two previous computationally resource and time-consuming algorithms.

Whatever gradient descent algorithm type is used, the cost function is calculated iteratively for each batch of data until all the training samples are used. This is called an *epoch*. The number of epochs is another hyperparameter that we can define. We want to stop the training process when the cost function converges to the minimum value, but in practice, achieving it is not always possible. Thus, the number of epochs is usually set to a large number, and the training process is stopped when the cost function stops decreasing.

Certain optimizers incorporate a momentum term into the weight update process, capturing information about gradients from previous training steps. One such example is the Adam optimizer [99], widely recognized for its strong performance across various training scenarios. An Adam optimizer with decoupled weight decay regularization, or AdamW [100], introduces weight decay regularization to the Adam optimizer. Weight decay regularization is important because it helps prevent overfitting by penalizing large weights in the model, thereby encouraging simpler, more generalizable solutions. Unlike traditional methods that modify the weight updates directly, AdamW decouples the weight decay from the gradient update step, applying it separately. This leads to more stable and efficient training, especially in deep learning models, by maintaining the advantages of adaptive learning rates while effectively controlling model complexity.

4.2.2.2 More on activation functions

When we employ the backpropagation algorithm to train the NN *via* Eq. 4.17, we see that the model weights will not be updated if the derivative of the activation function equals zero. In this case, the model stops learning, and the weights are frozen. It can also happen not only when the derivative of the activation function is zero but also when it is too close to zero - by using the backpropagation and multiplying the derivatives of layer l by the gradients computed for the layer $l + 1$, the gradients are exponentially decreasing. The weights for the layers closer to the input are updated by small values. This effect is called the *vanishing gradient problem*.

Contrary to that, the *exploding gradient problem* occurs when the gradients are too large, causing the model to diverge [101].

The most common activation functions are categorized into three groups: *ridge functions*, *radial functions*, and *fold functions*.

- **Ridge functions** are multivariate functions acting on the linear combinations of the inputs. The most common examples of the ridge functions are *Rectified Linear Unit (ReLU)* [102] function ($\mathcal{A}(x) = \max(0, ax + b)$) and *Logistic (sigmoid) function* ($\mathcal{A}(x) = (1 + e^{-(ax+b)})^{-1}$).

ReLU functions take as input a real number and output a number in the range $[0, \infty)$. It is continuous but not differentiable at $x = 0$ (C^0). Sigmoid functions take real numbers as input and output in the range $[0, 1]$. Apart from being continuous, they are also infinitely differentiable (C^∞). Sigmoid function (and similar ones) is usually used in classification problems to model the probability of an input belonging to a specific class.

- **Radial functions** or Radial Basis Functions (RBF) [103] constitute a distinct class of functions that are used in RBF networks, showing a particular efficiency as universal approx-

imators. Such functions are not used in my thesis project, yet they are worth mentioning, and one of the examples is the Gaussian Error Linear Unit (GELU) function ($\mathcal{A}(x) = x\Phi(x)$, where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution).

• **Fold functions** are most commonly used for *pooling* in the convolutional neural networks that will be described later on and in the output layers of multiclass classification networks. These activation functions involve aggregating over the inputs, such as computing the mean, minimum, or maximum. In multiclass classification, softmax activation is frequently employed. The softmax function takes as input a vector \mathbf{z} of K real numbers and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers:

$$\mathcal{A}(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K. \quad (4.18)$$

One should remember that an activation function \mathcal{A} is considered saturating if $\lim_{|x| \rightarrow \infty} |\nabla \mathcal{A}(x)| = 0$, if not - it is called nonsaturating function. Activation functions like ReLU, falling under the category of nonsaturating functions, may offer advantages over saturating ones, such as mitigating the vanishing gradient problem in networks with many layers.

4.2.2.3 DNN training problems and optimizations

The DNNs are powerful tools for solving complex problems but are not without challenges. One of the most common issues is the *overtraining*. The overtraining occurs from the bias-variance trade-off when training a model.

The bias of an algorithm reflects its ability to produce outputs closely aligned with the target values. At the same time, variance indicates its ability to maintain consistency in results even when the input is subject to slight variations caused by noise. This challenge is a common concern in optimization procedures, as achieving low bias and low variance simultaneously is often challenging and constrained by the available data. Typically, striking a balance involves finding a middle ground between a network capable of learning general patterns from the training dataset but achieving lower overall performance (*underfitting*) and a network that performs exceptionally well on the training data but struggles with unseen data (*overfitting*). This trade-off is influenced by the frequency with which the network encounters the training data, with networks typically experiencing underfitting at the start of training and overfitting after extensive training epochs.

We can monitor the overtraining by splitting the dataset into a training and a validation sets. The training set is used to train the model, while the validation set is used to evaluate the model periodically, *e.g.* after each epoch. This allows us to monitor when the model starts overtraining. An example of the overfitting on the training/validation loss plot is shown in Fig. 4.7. By monitoring the loss and validation loss, we can stop the training process when the validation loss starts increasing, indicating that the model is overtraining. This technique is called *early stopping*. From Fig. 4.7, we can see that the model starts overtraining after around the 15th epoch.

Additional techniques can be used to prevent overtraining instead of early stopping. Many approaches focus on constraining the complexity of the network by narrowing the range of potential weight values, preventing them from reaching excessively high values. This constraint can be implemented in various ways, such as by introducing *weight regularization*

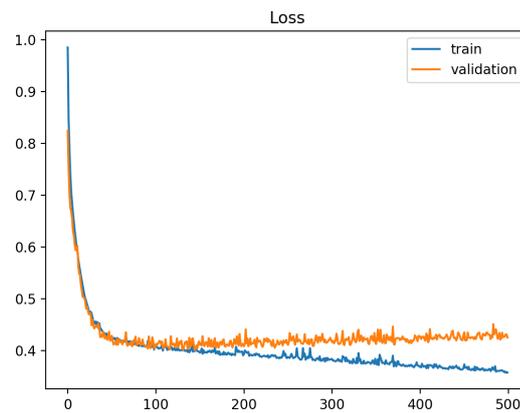


Figure 4.7: Overfitting on the training/validation loss plot. In an ideal scenario, both the training and validation loss curves should go down as the model learns. The training loss should eventually flatten out, indicating the model has learned the training data. The validation loss, representing performance on new data, should also decrease but ideally stay close to the training loss, showing the model is learning generally applicable patterns and not just memorizing the training data. In this example, the model starts overtraining after the 500th epoch, where the validation loss starts increasing. Sourced from [104].

or *dropout*. Weight regularization involves adding a penalty term to the loss function that discourages the network from assigning excessively high weights to the model parameters. This penalty term is typically proportional to the L1 (sum of the absolute values of the weights $\|\omega\|_1 = \sum_i |\omega_i|$) or L2 (sum of the squares of the weights $\|\omega\|_2 = \sum_i \omega_i^2$) norm of the weight vector. Dropout, however, involves randomly setting a fraction of the network's weights to zero during training, effectively reducing the network's complexity and preventing overtraining [105].

4.3 Convolutional Neural Networks

A subset of the DNNs called *Convolutional Neural Networks* (CNNs) aims to extract high-level insights from structured arrays of data, such as images, text, time series, and other formats. The first CNN architecture with its training using the backpropagation algorithm across all layers was proposed by LeCun et al. in 1989 [106]. Typically, CNNs are based on the alternations of the *convolutional layers*, *pooling layers*, and *fully connected layers*.

- **Convolutional layers** as one can deduce from the name, perform the convolution operation on the input data. Convolution is a linear operation, and thus, each convolutional layer is followed by the *activation function*. The neurons in the convolutional layer are connected to only a subset of adjacent neurons from the previous layer rather than all of them. This enables the layer to learn features that may appear multiple times and anywhere within the input. Each neuron in the layer shares the same set of weights, akin to a small *kernel* (also called *filter*) convoluted across patches of input and then shifted to cover the entire image, as presented in Fig. 4.8. These filters learn to extract various features, resulting in multiple *feature maps* that indicate the presence or absence of specific features within

the input. The kernel is usually initialized with random values and is updated during the training process.

- **Pooling layers** are used to capture the dominant features of each feature map while reducing the spatial dimensions of the input. It is achieved by examining patches of the input feature maps and summarizing them with a single value. Typically, this summary is either the average or the maximum value within the patch. Without pooling layers, feature maps generated by convolutional layers would closely match the input size, causing the network to focus on specific input regions to identify features. However, the relative position of features is often more critical than their absolute position. Therefore, downsampling is applied after each convolutional layer to address this issue. Pooling layers are characterized by their fixed functionality and lack of trainable parameters.

Fig. 4.9 shows the simple CNN architecture, where the convolutional layer is followed by the

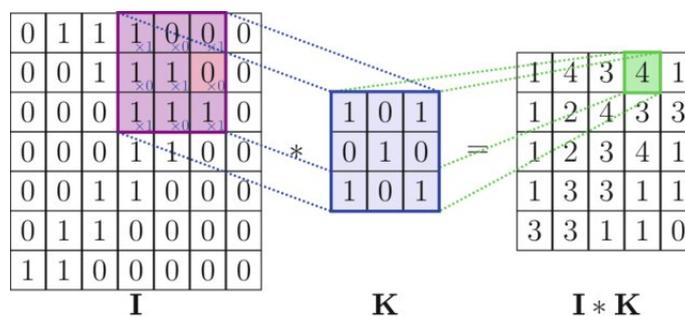


Figure 4.8: 2D-convolution operation. A kernel is convoluted with the input image, producing a feature map. A convolutional layer usually consists of multiple kernels, each producing a separate feature map. The values in the kernel are trainable parameters. Sourced from [107]

activation function, and the pooling layer is followed by the *flatten layer*, which converts the 2D-array into a 1D-array, and finally the fully connected layer. Combining convolutional and pooling layers in CNNs provides robustness against issues like vanishing and exploding gradients, thanks to weight sharing, which effectively acts as regularization. Additionally, through successive applications of convolutions, CNNs learn to extract high-level features and capture correlations between them.

Another variation of the CNNs is the *Residual Neural Networks* (ResNets) [108] [109]. ResNets are designed to address the vanishing gradient problem by introducing *skip connections* that allow the network to skip one or more layers. This enables the network to learn residual functions, which are the differences between the input and output of a layer. By learning residual functions, the network can focus on the residual information, making it easier to optimize the training process, because learning residuals is often easier than learning unreferenced functions. An example of a ResNet-like architecture with one skip connection is shown in Fig. 4.10, where the output of the first convolutional block is bypassed through a separate convolutional block and added to the output of the third convolutional block from the main path.

The CNNs are widely used in the field of particle physics for event processing [110], background rejection [111], and particle identification [112]. In the context of the PandaX-III experiment, where the collected signals may be interpreted as images and, in fact, during the post-processing of the data, converted into the 2D projections of the 3D event topology, the CNNs can be used to extract valuable information.

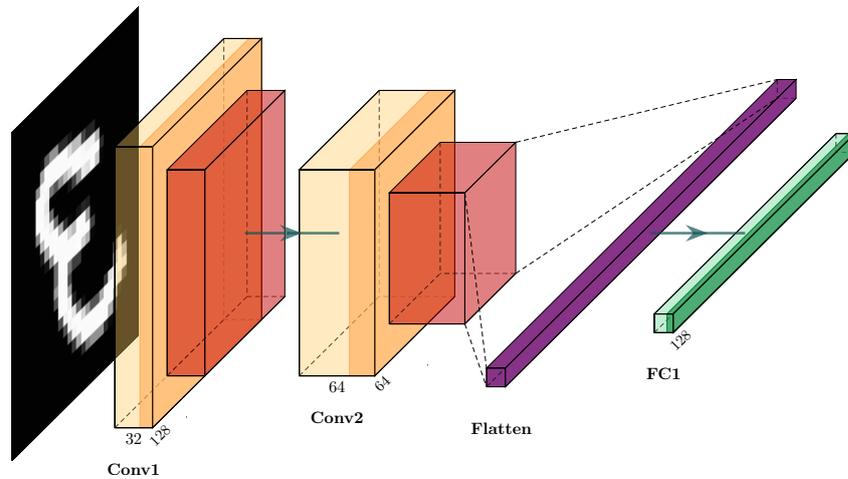


Figure 4.9: Simple Feedforward CNN architecture. To the left is an example of an input image with the size of $128 \times 128 \times 1$. The image is passed through the convolutional layer (yellow) with 32 kernels, followed by the activation function (orange) and the pooling layer (red), which reduces the spatial dimensions of the input by a factor of 2. This combination of layers is called a **Conv1** block and outputs 32 feature maps of size 64×64 . The next block, **Conv2**, has 64 kernels and outputs 64 feature maps of size 32×32 . The flattened layer (purple) reshapes the 2D array into a 1D array, which is then passed to the fully connected layer (light green) with 128 neurons and further applications of the activation function (dark green).

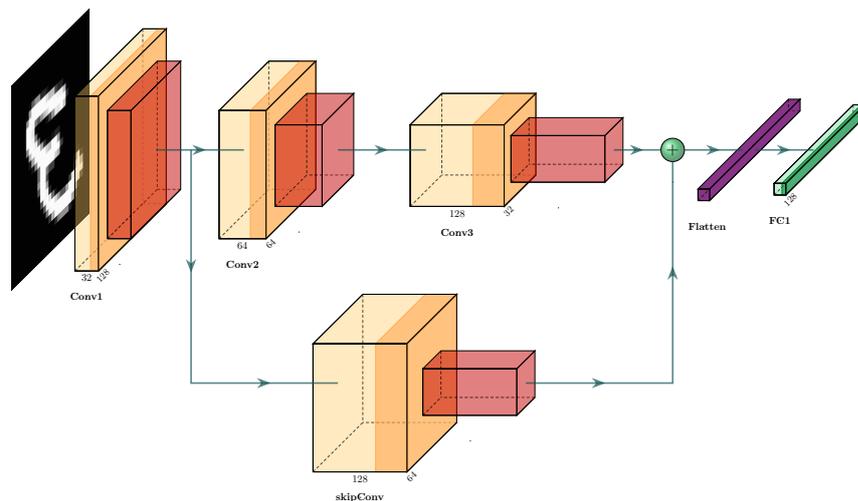


Figure 4.10: ResNet architecture with one skip connection. A skip connection clones the output of the first convolutional block and performs an additional convolutional operation on it together with pooling. The output of the skip connection is added to the output of the third convolutional block. Further operations are the same as in the example of the CNN (Fig. 4.9).

4.4 Conclusion

This chapter introduces the basics of machine learning algorithms, with the main focus on training the DNNs (CNNs in particular). With CNNs, we can extract complex information from the array of data that could be used for a vast variety of problems, including the particle physics problems. The CNNs were previously tested on the PandaX-III simulated data, for example, to reconstruct the initial Z vertex position of the event inside the TPC [113] and to discriminate between the signal and background events [114]. While showing promising results, none of the attempts addressed the problem of missing channels that may appear on the Micromegas Modules. In the next chapter, I will introduce the regression problem of the correct total energy reconstruction corresponding to the images of the 2D signal projections degraded by missing channels. Additionally, the reconstruction of the Z vertex position in parallel with the total energy reconstruction will be addressed as well.

Chapter 5

Regression problems

Я із надій будую човен, і вже немовби наяву
З тобою, ніжний, срібномовен, по морю радості пливу.
*(Ukrainian: I am building a boat out of hopes, and as if in a dream
I am with you, gentle, softly spoken, I am sailing on the sea of joy.)*
— *Vasyl Symonenko*

5.1	Introduction	94
5.2	Definitions and preliminary study	95
5.3	Data cleaning	98
5.4	Training the toyCNN on cleaned data	100
5.4.1	Global normalization of the input data images	100
5.4.2	Local normalization of the input images with $E_{\mathfrak{F}}$ as labels	103
5.5	Model adjustments and improvements, data augmentation	104
5.5.1	Missing channels introduction and expanded dataset	104
5.5.2	Two-output model	106
5.5.3	Separation of the two outputs	110
5.5.4	Skip connections. Residual learning	110
5.5.5	Different loss functions	112
5.6	Final model	114
5.6.1	Primary energy prediction evaluation	116
5.6.2	Primary energy prediction evaluation on the 2e dataset	117
5.6.3	Additional Multi3ResNet training on the 1e+2e dataset	118
5.6.4	Evaluation on the 0nbb dataset	120
5.7	Z position prediction	120
5.7.1	Evaluation of the model trained on the 1e dataset	122
5.7.2	Evaluation of the model trained on the 1e+2e dataset	122
5.8	Summary	124
5.9	Conclusions	124

5.1 Introduction

The reader is now expected to have a basic understanding of CNN, regression, and classification problems. The main goal of this chapter is the application of CNN to reconstruct the primary energy from the track topology. This approach is particularly suited to the case where the direct reconstruction, *i.e.* summing all recorded amplitudes, is impaired by missing channels. We will thus focus on this case. This regression problem can be treated in the context of Supervised Learning, see Section 4.2.1.

The energy quantity that we are interested in is the energy of the particle, or particles in the case of $0\nu\beta\beta$, at the origin of the event before they begin to lose energy in the propagation through the gas. For the $0\nu\beta\beta$, this energy should correspond to $Q_{\beta\beta}$, with which we want to show the existence of that mode of decay. A general notation for the initial energy is $E_{\mathfrak{P}}$ - primary energy. We have access to its true value in MC.

In addition to the primary energy, which remains our goal, we also try to teach our networks other characteristics of the event, *viz.* the would-be detected energy and the Z ordinate. The former is defined as the total energy reaching the anode plane that an ideal readout would record. It is an intermediate step in the conversion of the primary energy into recorded signals by the experimental apparatus. As such, we expect it to help the networks learn how to disentangle, among topology features, what is due to the primary event and its processing by the TPC from what is due to a partially deficient readout. The Z ordinate, defined *e.g.*, defined as the distance of the primary vertex to the readout plane, can play a similar role. In addition, it is an exciting quantity in itself that can be used to reject those events close to the anode plane that are affected by the radioactive contamination of the Micromegas.

In this chapter (as well as in the next one), we consider a wide range of primary energy, extending well outside the ROI (defined in Section 2.4 as [2364, 2552] keV range). Due to the limitations (finite resolution, undetected bremsstrahlung) and deficiencies (missing channels, inhomogeneities) of the experimental apparatus, interesting events (those native in the ROI) and uninteresting ones can end up with similar direct energy reconstruction. In order to restore the correct primary energy in each case, the network needs to identify features that pertain to ROI events and events at higher and lower energy. The features that we have in mind are *e.g.* length, twist, or density fluctuations. As a whole, they constitute what we will call the "track-topology". Quantitatively, the track-topology can be equated to the set of all recorded amplitudes. An event is then the union of two objects: the track-topology and the sum of amplitudes. In addition, with this wide energy range, we deport possible edge effects away from the sensitive region of the ROI.

This chapter is organized as follows. The first section is dedicated to the preliminary study of a CNN toy-model. The next section explains the data cleaning procedure, which is applied to the dataset, in order to provide the model with the most relevant data. The following section is dedicated to the architecture development of the CNN model, and the manipulations that are applied to the input data and the labels. The last section is dedicated to the final model architecture, and the results on the model's performance.

5.2 Definitions and preliminary study

As mentioned in the introduction, the primary energy is the energy of the particle, or particles, at the event's origin. For example, in the case of the $0\nu\beta\beta$, the primary energy corresponds to $Q_{\beta\beta}$. We generalize the notation of the primary energy as $E_{\mathfrak{P}}$. Likewise, the detected energy, the total energy reaching the anode plain, is denoted as $E_{\mathfrak{D}}$. Finally, the Z ordinate of the track's vertex is denoted as Z_0 .

The values predicted by the model are marked with a *hat*, *viz.* $\hat{E}_{\mathfrak{P}}$, $\hat{E}_{\mathfrak{D}}$, \hat{Z}_0 .

To begin with, we set a goal to explore the competence of CNN models to extract information from the track topology. We do this on a setup as simple as possible, *viz.* on events with only one electron. A toy model defined as *toyCNN* is used for this purpose. Therefore, no pathologies connected with the detector defects, such as the missing channels, are considered in this preliminary study.

The project's general goal is to make a model that can generalize well on the large energy range of the events and not only on the events localized in the region of interest (ROI). Of course, the events in ROI for $0\nu\beta\beta$ are of the most significant interest to us. However, we want to be able to reconstruct the correct energy of all the events that would be present during the real data acquisition. This is important, primarily because the interesting events for us, such as $0\nu\beta\beta$ events, could end up having the detected energy far outside the ROI, due to the detection deficiencies (e.g., missing channels, resolution of the detector, etc.) and vice versa, the background events could end up having the detected energy inside the ROI, due to the same reasons. With this in mind, in this exercise (as well as in the rest of the chapter), a wide range of $E_{\mathfrak{P}}$ of the electron track events is considered. We prepare the data with REST, generating electron events in the gas volume of the TPC with the angular distribution uniformly distributed over the phase space of the TPC. For this preliminary study, the energy range of [1, 3] MeV is selected.

In the current study, a simplification of the input data is made: only the events registered by one Micromegas Module are used, letting aside the cases where several (as there might be up to four) modules are involved in the track event registration. This simplification was made in order to reduce the complexity of the input data and to avoid, for the time being, the issue of the gap between the modules. This simplification was initially intended to be temporary but was never removed due to a lack of time. The issue should be addressed in future studies.

With such simplification, the conversion of the signals from the Micromegas Modules to the 2D images is relatively straightforward. Each electronics channel of the Micromegas module provides a time series of ADC counts with 512 bins. The channels with IDs from 0 to 63 correspond to the X projection and produce a 64×512 array, while the channels with IDs from 64 to 127 correspond to the Y projection and produce an equivalent array. By rebinning the time-series to 128 bins and concatenating the X and Y projections, we obtain a 128×128 array, which contains the track-topology as well the sum of amplitudes, in a 2D image that we will use as input to the CNN model.

Now, we can compare the $E_{\mathfrak{P}}$ and $E_{\mathfrak{D}}$ distributions of the simulated events before (Fig. 5.2a) and after (Fig. 5.2b) applying the "one Micromegas Module selection." The initial number of events, equal to 41k, is reduced to 20k. We can see a trend in the new energy spectrum of the newly selected events, where higher energies are less frequent, as such events are much more likely to be longer and, therefore, be registered by more than

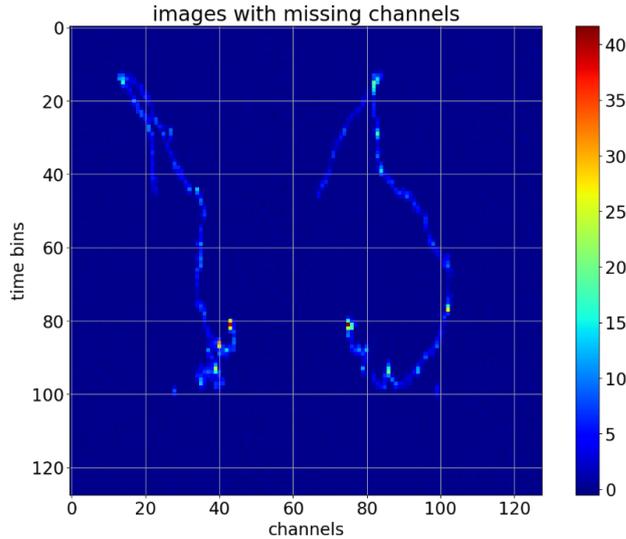


Figure 5.1: Example of an input image (128×128) without missing channels. The left part of the image corresponds to the XZ projection of the track, and the right part - to the YZ projection. The amplitude of the pixels is the ADC count, divided by amplification factor. The sum of all amplitudes is the total deposited energy of the track event.

one Micromegas Module.

As for the toyCNN, no missing channels are simulated on the readout plane, leaving the task of the CNN rather simple - to be able to properly predict the energy of the track event, given the 2D images of the track topology, which fully contain the information about the energy deposition along the track.

It is important to remember that the CNN model training relies on the gradient descent algorithm for optimization, which updates the model's weights by the magnitude of the gradient of the loss function. Therefore, the normalization of the input data (features) and the labels are crucial for the model's training in order to avoid the vanishing or exploding gradients [101]. The input data images are normalized to the range $[0,1]$ by dividing by the maximum amplitude of the input images among all inputs. The labels are normalized by the maximum energy of the events.

The toyCNN contains two convolutional blocks, each containing a convolutional layer, an activation layer, and a max pooling layer. The convolutional layers have 32 and 64 filters, respectively. The activation layers are of the ReLU type. The convolutional blocks are followed by a flattening layer and one hidden fully connected layer with 16 neurons, accompanied by a ReLU activation layer, before the output. The output layer contains one neuron, which outputs \hat{E}_{tr} for the track event. The architecture of the toyCNN is depicted in Fig. 5.3, and a detailed description is given in Table 5.1.

75% of the data are used for training, and the remaining 25% for testing. The model is trained for 50 epochs with a batch size of 256. The model predictions \hat{E}_{tr} obtained on the test dataset are shown in Fig 5.4a. We can see that the model has flaws, partially due to the lack of training data in the high-energy region. It also cannot correctly predict events with low energies. Nevertheless, looking at the correlation plot in Fig. 5.4b, we can see that the model's predictions are promising, and the model is able to predict the energy of the track event to some extent. The study is promising enough to employ CNN further for the

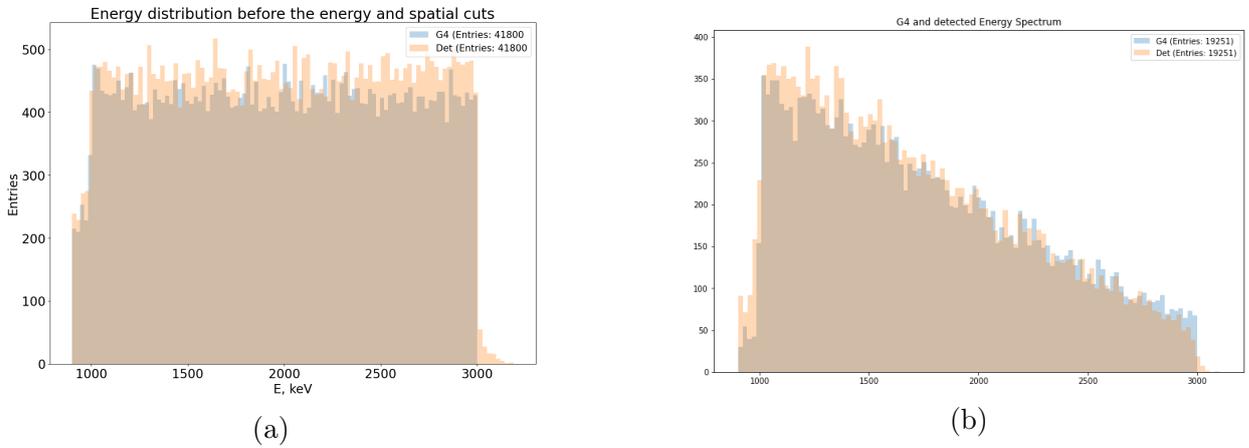


Figure 5.2: $E_{\mathcal{P}}$ and $E_{\mathcal{D}}$ distributions before, (a), and after, (b), the "one Micromegas Module selection".

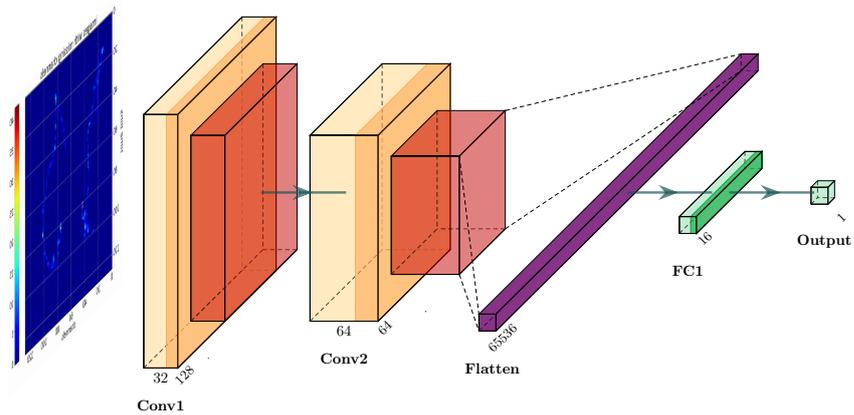


Figure 5.3: Preliminary toyCNN architecture. The input to the network is a 2D image of the track XZ and YZ projections. No missing channels are used. The output is the $E_{\mathcal{P}}$ of the event.

Layer (type)	Output Shape	Attribute	N_p
Input	(batchSize, 128, 128, 1)		0
Conv2D	(batchSize, 128, 128, 32)	$3 \times 3, 32$	320
Activation	(batchSize, 128, 128, 32)		0
MaxPooling2D	(batchSize, 64, 64, 32)	2×2	0
Conv2D	(batchSize, 64, 64, 64)	$3 \times 3, 64$	18496
Activation	(batchSize, 64, 64, 64)		0
MaxPooling2D	(batchSize, 32, 32, 64)	2×2	0
Flatten	(batchSize, 65536)		0
Dense	(batchSize, 16)		1048592
Activation	(batchSize, 16)		0
Dense	(batchSize, 1)		17

Table 5.1: Preliminary toyCNN. Attribute for convolutional layers stands for the kernel shape and the number of kernels, while for pooling layers it is kernel shape. N_p is the number of parameters in the layer. For the convolutional layers, N_p is calculated as $(F * F * C * K) + (K \text{ bias})$, where F is the kernel size, C is the number of channels in the input, and K is the number of kernels. For the dense layers, N_p is calculated as $(N * M) + (M \text{ bias})$, where N is the number of neurons in the previous layer and M is the number of neurons in the current layer. The input to the network is a 2D image of the track. The model is trained to predict the primary energy $E_{\mathfrak{P}}$.

energy reconstruction of tracks with missing channels.

5.3 Data cleaning

A possible explanation for the fact mentioned in Section 5.2 that predictions become worse at the highest edge of the energy range lies in the non-uniform distribution of $E_{\mathfrak{P}}$, see Fig. 5.2. The rationale is that features belonging to higher energies will have a lesser weight. It is reasonable to make the distribution more uniform to cure this problem. This is done by increasing the statistics of the events in the high-energy region, which are less frequent and are more likely to be registered by more than one Micromegas Module, and decreasing the statistics of the events in the low-energy region. With such data, the model is supposed to learn the energy deposition patterns along the track without being case-biased.

Additionally, some electrons can partially escape the active area of the detector. For example, they can emit a gamma via bremsstrahlung, which can travel a long distance before being converted. In principle, these cases should be taken into account. But, for the time being, we decided to simplify them. In order to do so, we consider the residual distribution between primary and detected energy and cut away all that is beyond the 3σ range of this residual distribution (Fig. 5.5). The whole dataset prepared for further trainings has 80k of events in total and the corresponding $E_{\mathfrak{P}}$ and $E_{\mathfrak{D}}$ distributions are shown in Fig. 5.6.

Now, after applying such data cleaning procedure, let's proceed to the model training.

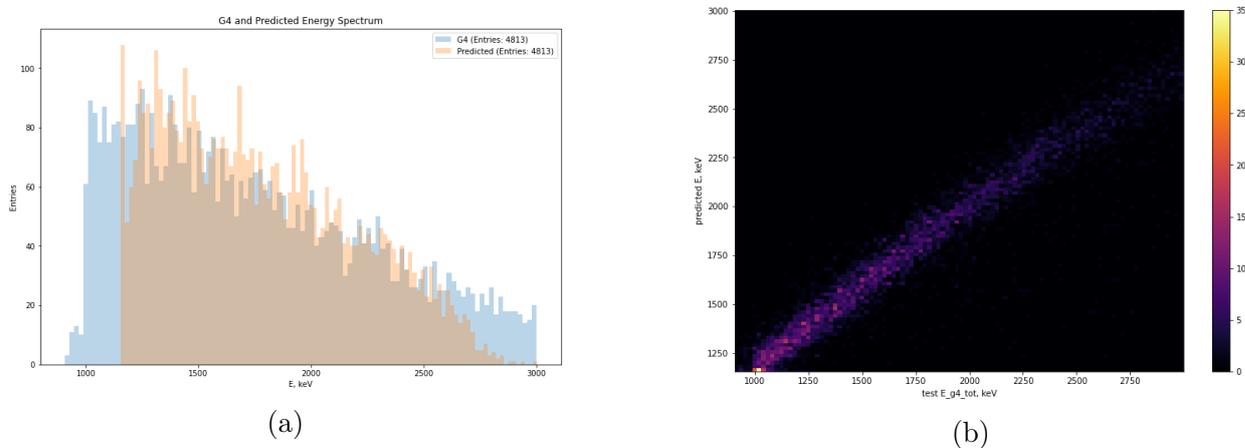


Figure 5.4: (a) Comparison of the true value ($E_{\mathcal{P}}$) and predicted value ($\hat{E}_{\mathcal{P}}$) of the Primary Energy. (b) Correlation plot of $\hat{E}_{\mathcal{P}}$ vs. $E_{\mathcal{P}}$.

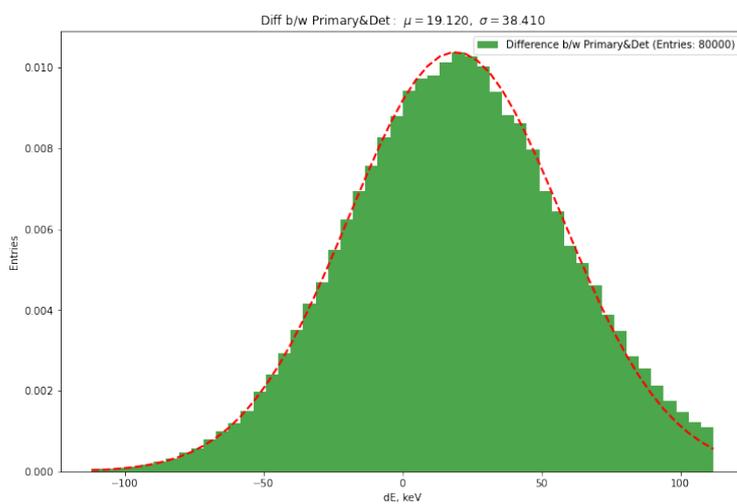
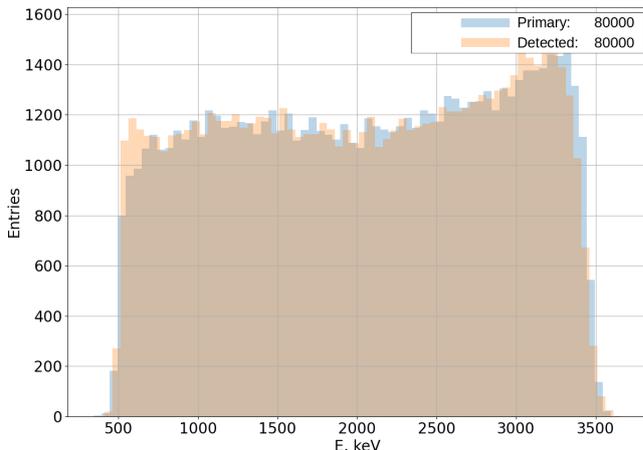


Figure 5.5: Residual distribution between $E_{\mathcal{D}}$ and $E_{\mathcal{P}}$. The 3σ range determines the events that are considered for the training.

Figure 5.6: E_p and E_D flat distribution.

5.4 Training the toyCNN on cleaned data

The model architecture does not change for the new training session. However, some adjustments have been made to the input data and the labels. First, next to the *global normalization* used so far, we introduce a *local normalization*. The global normalization is done by dividing the amplitudes of the input images by the maximum amplitude of the input images among all events. The local normalization is instead done by dividing the amplitudes of the input images by the maximum amplitude for each event. One should keep in mind that this data is simulated without missing channels on Micromegas yet. The labels are normalized by the maximum energy of all the events, as before.

To evaluate its performance on the new dataset, toyCNN is trained separately to predict E_D and E_p with global normalization. It is then also trained on E_p with local normalization. Now, the dataset is divided into 50% for the training, 25% for the validation, and 25% for the testing.

5.4.1 Global normalization of the input data images

5.4.1.1 Detected energies as labels

We start with the detected energy E_D . This quantity is not interesting in itself, but in so far as it is much simpler, it can serve as a reference for the exercise on E_D that comes next. MeanSquaredError is used as a loss function, and Stochastic Gradient Descent is used as an optimizer.

For a batch size of 256, the model is trained for 100 epochs. The training and validation loss curves are shown in Fig. 5.7. The loss decreases with the number of epochs, and from the similar trend of the training and validation loss, we can conclude that the model is indeed generalizing well on the unseen data. The validation loss does not experience any fluctuations either, which could have indicated the overfitting of the model.

If we look at the correlation plot between E_D and \hat{E}_D in Fig. 5.8, we can see the distribution is now much more homogeneous and well clustered around the diagonal. On the

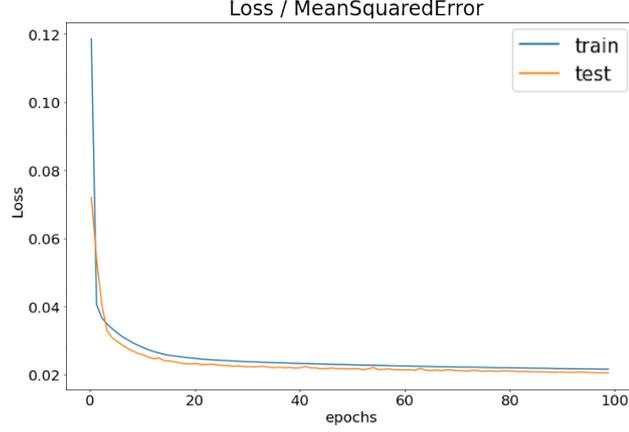


Figure 5.7: Train and validation loss for the toyCNN training with $E_{\mathcal{D}}$ considered as labels for the cleaned 80k dataset.

other hand, there is a slight deviation from the diagonal around 2000 keV. This artifact is better visible on the residual correlation plot (Fig. 5.9a). We observe an incorrect prediction of the energies lower than 2000 keV. But, we see that overall, the residuals follow a Gaussian distribution in Fig. 5.9b, with mean and RMS equal to -0.8 keV and 63.5 keV, respectively. Despite the deviations on the correlation plot, we see that the model can predict the Detected energies with a good performance.

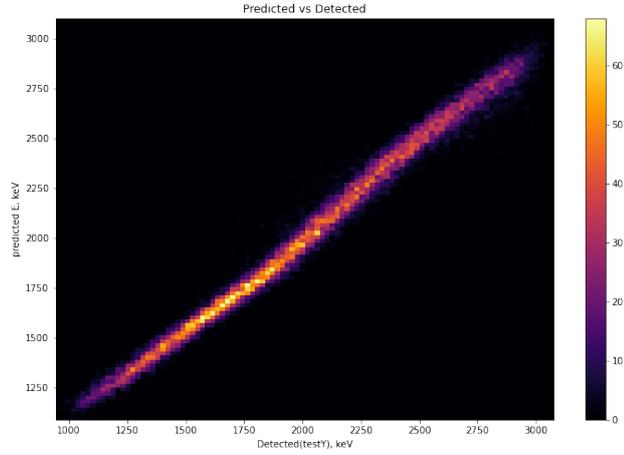


Figure 5.8: Correlation plot between $E_{\mathcal{D}}$ and $\hat{E}_{\mathcal{D}}$ from the toyCNN evaluation on the cleaned 80k dataset. Global normalization of the input images.

5.4.1.2 Primary energies as labels

Now, the model is trained to predict $E_{\mathcal{P}}$ instead. The training-validation loss plot shows (Fig. 5.10) that despite the appearance of the fluctuations on the validation loss, the model is still converging. Thus it is supposed to have a good performance on the unseen data. Nevertheless, the prediction results are similar to those of the previous training. The correlation plot between $E_{\mathcal{P}}$ and $\hat{E}_{\mathcal{P}}$ also has an artifact around the 2000 keV (Fig. 5.11a), but the residual distribution itself follows a Gaussian distribution with mean and RMS equal to

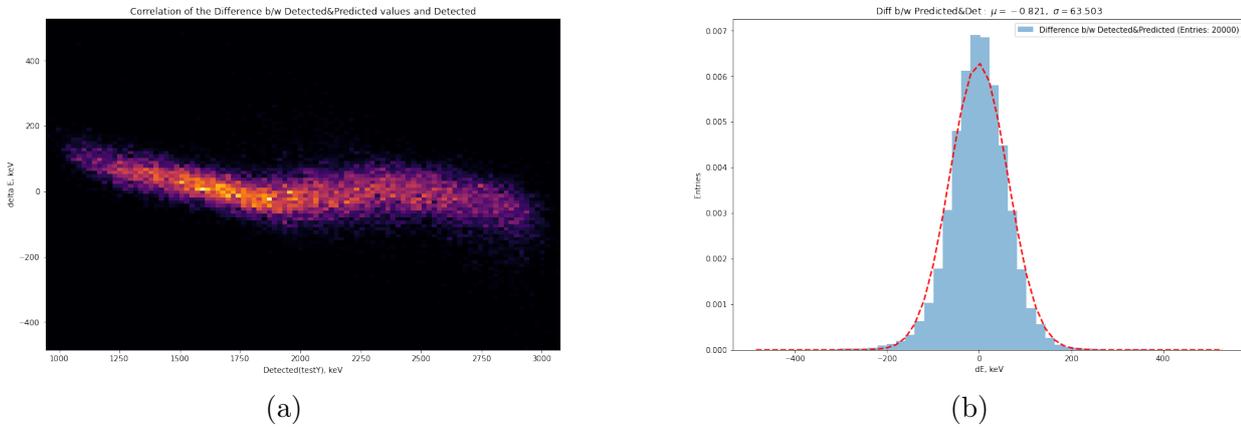


Figure 5.9: Evaluation of the toyCNN trained on the cleaned 80k dataset. Global normalization of the input images. (a) Correlation plot for the residual between $E_{\mathcal{D}}$ and $\hat{E}_{\mathcal{D}}$ with respect to true labels - $E_{\mathcal{D}}$ in this training session. (b) Residual distribution between $E_{\mathcal{D}}$ and $\hat{E}_{\mathcal{D}}$.

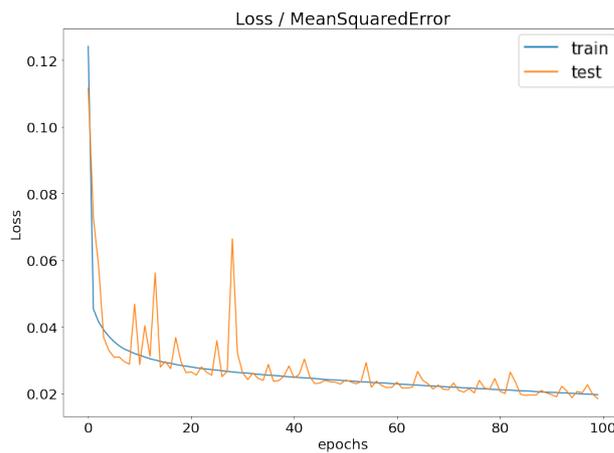


Figure 5.10: Train and validation loss for the toyCNN training with $E_{\mathcal{D}}$ as labels from the cleaned dataset. Global normalization of the input data events is applied.

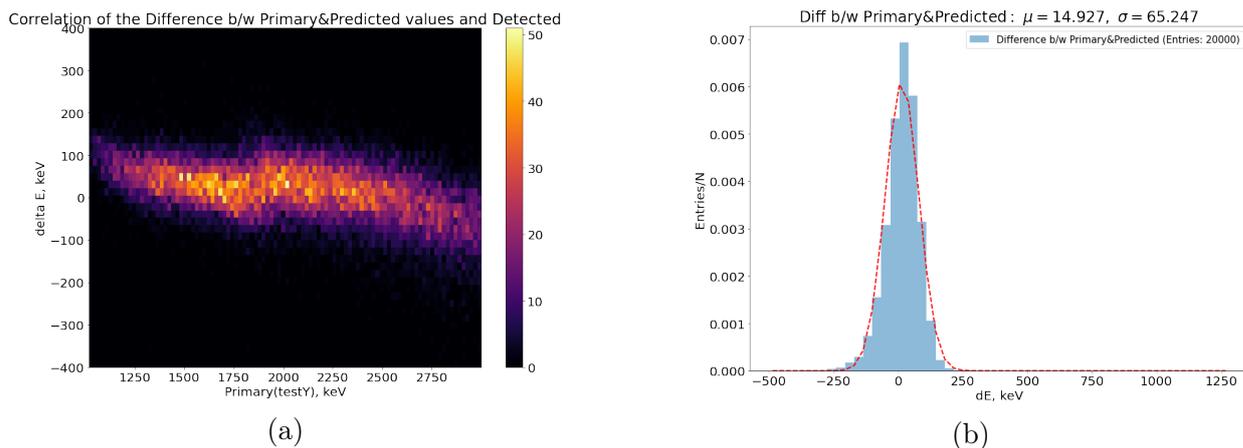


Figure 5.11: Evaluation of the toyCNN trained on the cleaned 80k dataset. Global normalization of the input images. (a) Correlation plot for the residual between $E_{\mathcal{P}}$ and $\hat{E}_{\mathcal{P}}$ with respect to $E_{\mathcal{P}}$. (b) Residual distribution between $E_{\mathcal{P}}$ and $\hat{E}_{\mathcal{P}}$.

14.9 keV and 65.2 keV, respectively as could be seen in Fig. 5.11b. And this is particularly interesting, as the toyCNN is able to predict the primary energies with as good a resolution as it does for the detected energy.

5.4.2 Local normalization of the input images with $E_{\mathcal{P}}$ as labels

In addition to the previous study in section 5.4.1, an additional test is conducted with a *local normalization* of the input images. In this way, the input images are normalized independently. Thus, topological information about the track is preserved while amplitudes are blurred. With this normalization, each input becomes independent and provides only the topological information about the energy deposition along the track.

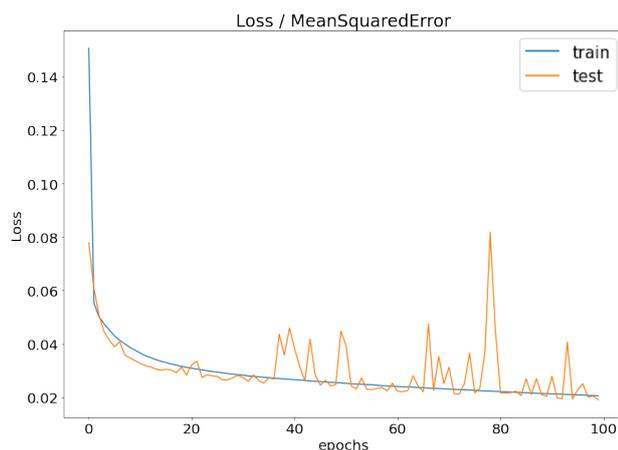


Figure 5.12: Train and validation loss for the toyCNN training with $E_{\mathcal{P}}$ as labels. Input data events are normalized independently - local normalization.

As we reduce the amount of information in the input data, it is expected that the model's performance will degrade. And indeed, the correlation plot for the residual between the $E_{\mathcal{P}}$ and $\hat{E}_{\mathcal{P}}$ has a larger deviation from the zero baseline. The residual distribution itself is

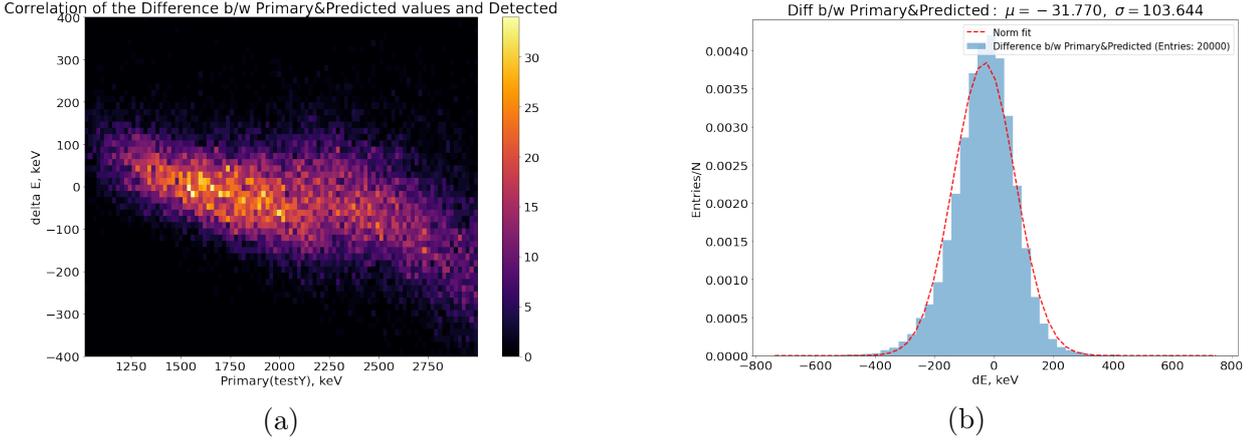


Figure 5.13: Evaluation of the toyCNN trained on the cleaned 80k dataset with local normalization of the input images. (a) Correlation plot for the residual between the $E_{\mathcal{P}}$ and $\hat{E}_{\mathcal{P}}$ with respect to the $E_{\mathcal{P}}$. (b) Residual distribution between the $E_{\mathcal{P}}$ and $\hat{E}_{\mathcal{P}}$.

wider, with mean and RMS equal to -31.8 keV and 103.6 keV, respectively as could be seen on the Fig. 5.13b and Fig. 5.13a. Despite such degradation, the model is still able to predict $E_{\mathcal{P}}$ energies with a rather good resolution. One should stress out that this prediction is obtained only from the topological information of the track. Again, this provides more evidence that the CNN application for energy reconstruction is a promising approach.

5.5 Model adjustments and improvements, data augmentation

The toy model provided a valuable insight into the CNN model’s performance on the energy reconstruction from the track topology. However, none of the results concerned the missing channels problem and overall the model’s performance obviously is asking for improvements.

5.5.1 Missing channels introduction and expanded dataset

Now, the time to introduce the missing channels on the readout plane is ripe. For the current studies, the missing channels are introduced randomly on each Micromegas Module out of 52 on the readout plane. The number of missing channels is chosen to be 3 out of 128 per module ($\sim 2\%$ of all channels), as it was estimated to be the borderline case for the module usage. If the number of the missing channels is greater than 3, the module is considered to be out of order and is not used. Therefore, 52 combinations of the missing channels are generated, and the input data images are adjusted accordingly. The input data images are of a size of 128×128 , and columns corresponding to missing channels are set to zero. In Fig. 5.14a an example of the input image with missing channels is shown. In order to explicitly provide the model with information about the missing channels, additional mask images are introduced. The mask images are the same shape as the inputs, and columns corresponding to the missing channels are set to ones (Fig. 5.14b).

The mask arrays are then concatenated with the input arrays, and the resulting $2 \times 128 \times 128$ arrays are used as inputs to the CNN model as it is visualized in Fig. 5.14c. The

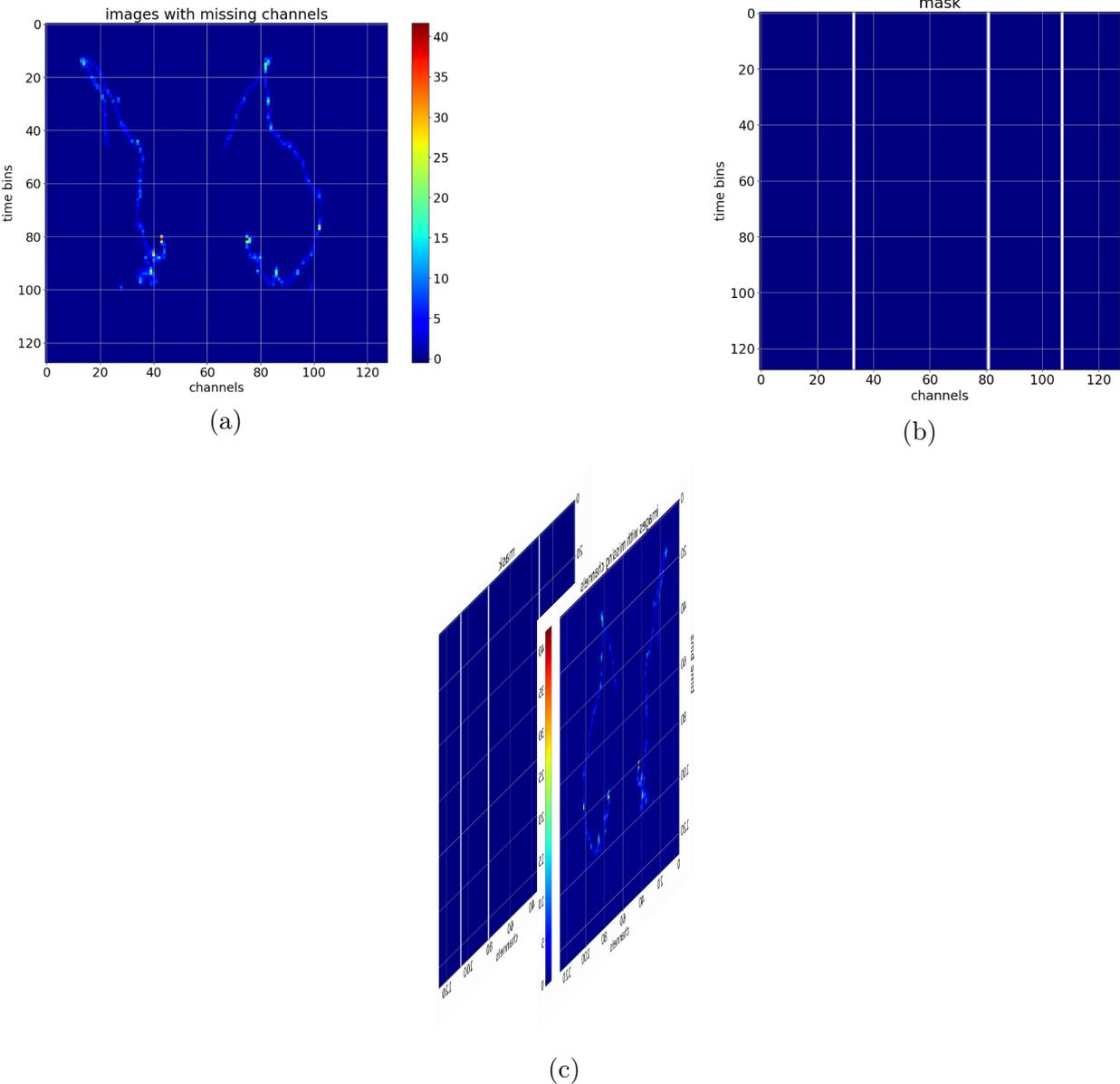


Figure 5.14: (a) Example of the input array with missing channels. (b) Example of the missing channels map, corresponding to the input array. (c) Concatenated map and input arrays, of a [2, 128, 128] size.

model is now adjusted to accept the $2 \times 128 \times 128$ arrays as inputs.

In addition, an energy range for E_{p} of the events is expanded to $[500, 3500]$ keV to introduce more low and high-energy events to the model. All the data preparation is the same as for the previous dataset. For a better generalization of the model, the dataset now contains 300k events, 50% of which are used for the training, 25% for the validation, and 25% for the testing. Ultimately, all the prediction results are evaluated on the test dataset.

5.5.2 Two-output model

A study of the toyCNN was conducted on the cleaned dataset with the missing channels introduced. The model was trained to predict only E_{D} energies with the global normalization of the input images. The results yield an RMS of 67.7 keV for the residual between E_{D} and \hat{E}_{D} and overall performance of the model shown a potential for further improvements. The results of the study were summarized and demonstrated on the 7th International Conference on Micro Pattern Gaseous Detectors 2022 [115].

Let us proceed to the model modification. Another fully connected hidden layer is added to the model, and now the two of them have 128 and 64 neurons, respectively. In addition to that, the new model has two outputs, corresponding to E_{p} and E_{D} . The justification for adding E_{D} as a prediction has been outlined in the introduction. In this case, we move to a multitask learning approach, where the model learns to predict multiple variables simultaneously. This can lead to better performance on each individual task by exploiting the shared information.

As the model is now adjusted to have two outputs, the loss function is adjusted to be a sum of the losses for the two outputs. Choosing the proper loss function becomes more critical. The updated architecture of the CNN model is depicted in Fig. 5.15, which is now called *MultiCNN*. The precise model description is given in Table 5.2.

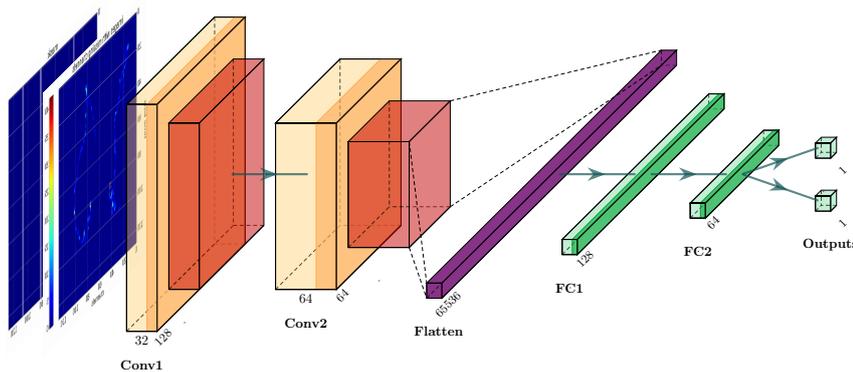


Figure 5.15: MultiCNN architecture, with two outputs.

The model is trained on the new 300k dataset, with input images normalized globally. After consecutive training sessions with the hyperparameters manual tuning, the Stochastic Gradient Descent with the learning rate of $2 \cdot 10^{-2}$ is employed. The MSE is used as a loss function. The final MultiCNN model configuration is trained for 100 epochs with a batch size of 256.

Layer (type)	Output Shape	Attribute	N_p
Input	(batchSize, 128, 128, 2)		0
Conv2D	(batchSize, 128, 128, 32)	$3 \times 3, 32$	1632
Activation	(batchSize, 128, 128, 32)		0
MaxPooling2D	(batchSize, 64, 64, 32)	2×2	0
Conv2D	(batchSize, 64, 64, 64)	$3 \times 3, 64$	18496
Activation	(batchSize, 64, 64, 64)		0
MaxPooling2D	(batchSize, 32, 32, 64)	2×2	0
Flatten	(batchSize, 65536)		0
Dense	(batchSize, 128)		8388736
Activation	(batchSize, 128)		0
Dense	(batchSize, 64)		8256
Activation	(batchSize, 64)		0
Dense	(batchSize, 1)		65
Dense	(batchSize, 1)		65

Table 5.2: MultiCNN with two outputs. Attribute for convolutional layers stands for the kernel shape and the number of kernels, while for pooling layers it is kernel shape. N_p stands for the number of parameters in the layer.

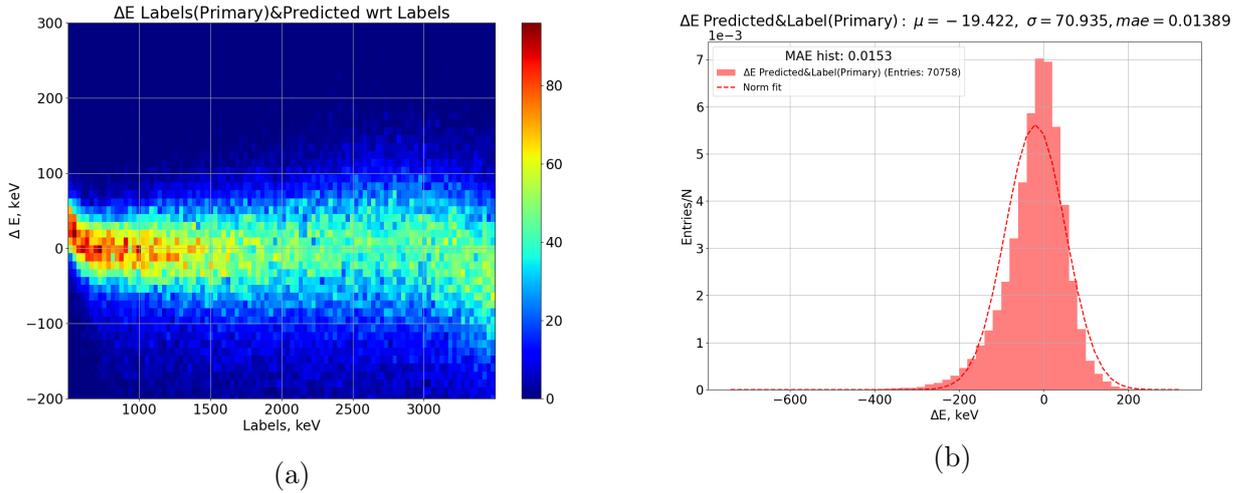


Figure 5.16: MultiCNN evaluation. Trained on the 300k dataset with global normalization of input images. (a) Correlation plot for the residual between E_{γ} and \hat{E}_{γ} . (b) Residual distribution between E_{γ} and \hat{E}_{γ} .

From now on, the model predicts two outputs: the primary and detected energies. For the former, the correlation plot is shown in Fig. 5.16b. It lies around zero. The problem is that the distribution is much more dispersed for higher energies than those lower than 1500 keV. But, we no longer observe the artifact around 2000 keV. Yet, on each end of the correlation plot, deviations from the zero baseline are visible (Fig. 5.16a). These new deviations contribute to the skewness of the residual distribution in Fig. 5.16b. Roughly fitting the distribution with a Gaussian, we obtain the mean and standard deviation equal to -19.4 keV and 70.9 keV, respectively.

Additionally, we determine an estimator of the efficiency with which our prediction is able to collect events into the ROI. For this, we select a very narrow slice of $E_{\mathfrak{P}}$ around $Q_{\beta\beta}$, of width ± 15 MeV, which emulates $0\nu\beta\beta$ events. Moreover, we determine the fraction of it staying within the ROI of ± 85 MeV. The value found for the efficiency, *viz.* 79%, is reported in Fig. 5.17, which shows the distributions of $\hat{E}_{\mathfrak{P}}$ for the selected events. Note that our efficiency estimator is only indicative here, given that it only imperfectly emulates the $0\nu\beta\beta$ events, which consist of two electrons and not one.

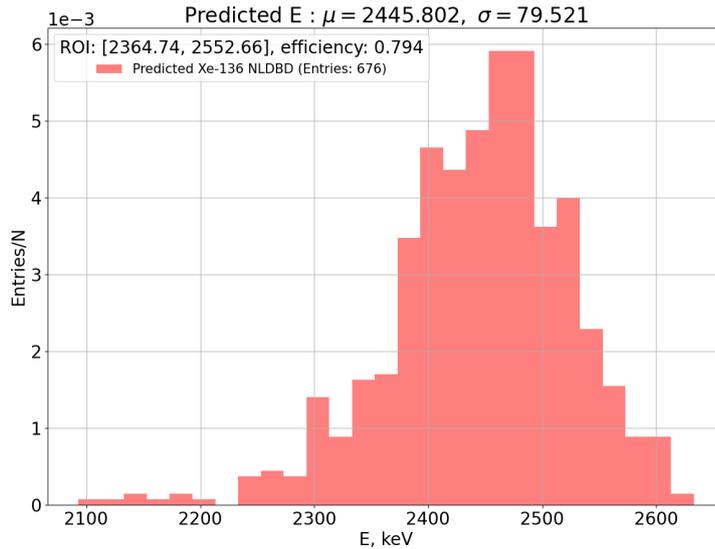


Figure 5.17: $\hat{E}_{\mathfrak{P}}$ distribution in the ROI from MultiCNN. The efficiency of events' prediction in ROI is 79%.

For comparison, the results for $E_{\mathfrak{D}}$ values are also presented. From the correlation plot, we see that the dispersion of the residual is less pronounced in comparison to $E_{\mathfrak{P}}$, although still present for higher energies (Fig. 5.18a). Similarly, the artifacts at each end of the energy range are present, which contributes to the skewness of the residual distribution (Fig. 5.18b). The standard deviation that comes from the Gaussian fit is 61.1 keV.

This model yields fairly good results, taking into account the complexity of the data and the task to reconstruct the primary energies from the tracks containing the missing channels. The results for the primary energies is still quite far from the desired resolution.

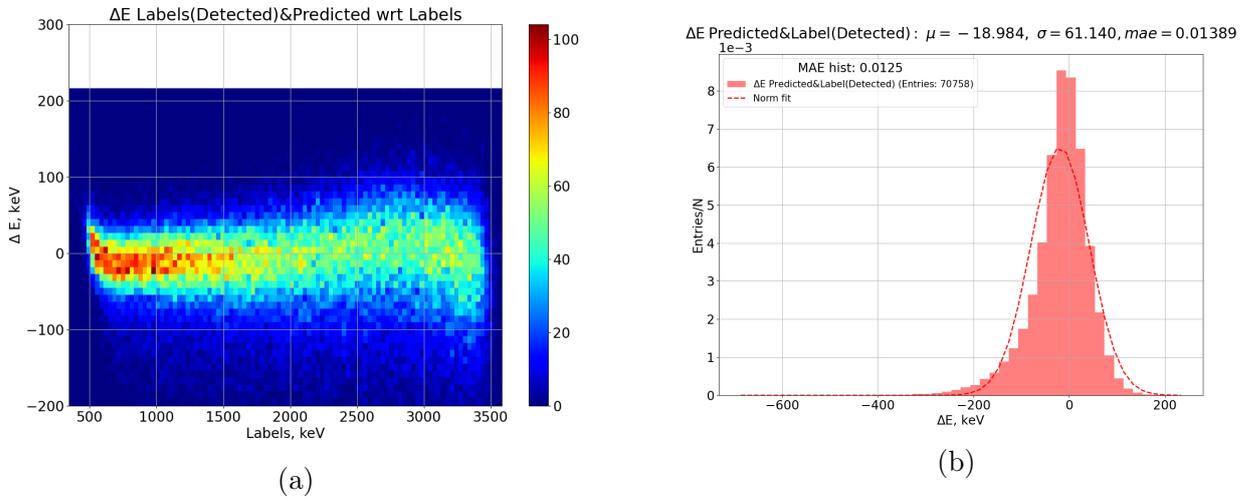


Figure 5.18: MultiCNN evaluation. Trained on the 300k dataset with global normalization of input images. (a) Correlation plot for the residual between $E_{\mathcal{D}}$ and $\hat{E}_{\mathcal{D}}$. (b) Residual distribution between $E_{\mathcal{D}}$ and $\hat{E}_{\mathcal{D}}$.

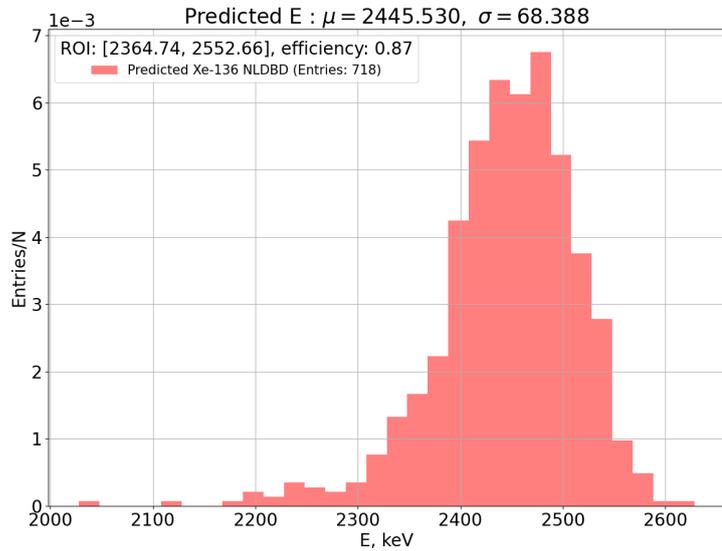


Figure 5.19: MultiCNN evaluation. $\hat{E}_{\mathcal{D}}$ distribution corresponding to $E_{\mathcal{D}}$ in the ± 15 MeV band at $Q_{\beta\beta}$.

5.5.3 Separation of the two outputs

In a trial and error approach, a CNN with separate hidden layers for the two outputs is tested as visually presented in Fig. 5.20. Now the appearance of the artifacts is reduced (Fig. 5.21), but an overall shift of the residual distribution is observed.

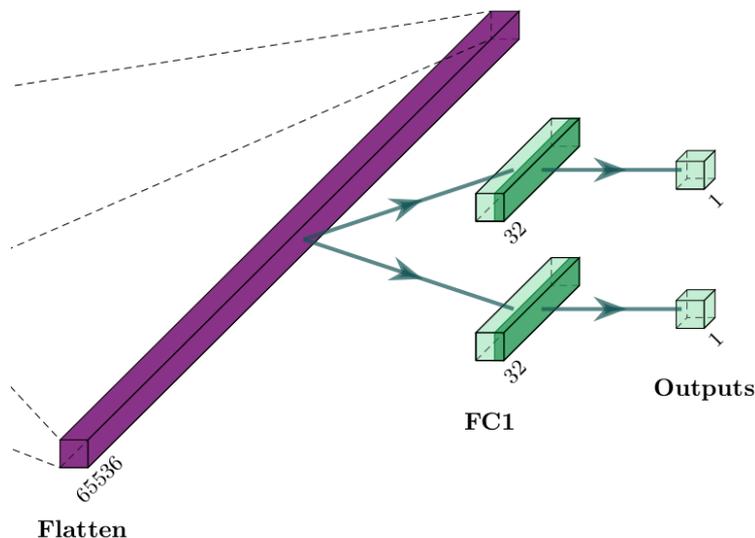


Figure 5.20: Two outputs separation in the MultiCNN model.

5.5.4 Skip connections. Residual learning

After multiple attempts to improve the MultiCNN performance, an approach using the Residual Network-like architecture (introduced in Section 4.3) is tested. We add a *skip connection* to the model, allowing the original input to be directly transmitted to downstream layers. This ensures that the gradient can flow directly through the skip connection, bypassing the complex transformations within the block. In the context of ResNets, skip connections add the element-wise sum of the input to a residual block and the output of that block. This ensures that the later layers still have access to the original information and can learn more complex representations based on it [108][109].

A modified version of the MultiCNN model with skip connection, containing a convolution block + pooling layer is present in Fig. 5.22, called *MultiResNet*. We omit the detailed description of the model, as it is similar to the previous ones, and the training process. The best results for the primary energies are presented in the next section.

5.5.4.1 MultiResNet model evaluation results

The correlation plot for the residual between true and predicted primary energies is mostly flat while being offset with respect to the zero baseline on average (Fig. 5.23a). The dispersion of the residual is less pronounced than in the previous MultiCNN model. From the Gaussian fit of the residual distribution, we see that the mean shifts by around 30 keV, while we experience a decrease in the standard deviation to around 60 keV. The E_{β} prediction is the most important for this study, and the plots for the E_{β} predictions are not provided.

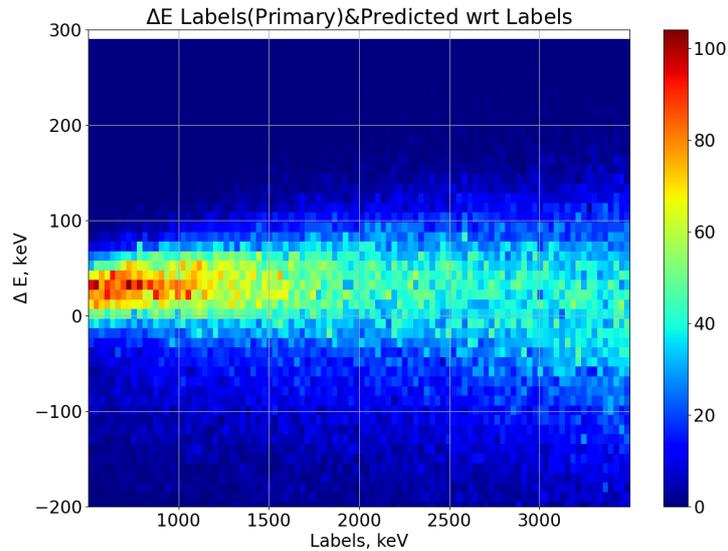


Figure 5.21: From the MultiCNN model with *separated* outputs evaluation. Correlation plot for the residual between $E_{\mathfrak{P}_3}$ and $\hat{E}_{\mathfrak{P}_3}$ with respect to $E_{\mathfrak{P}_3}$. The presence of the artifacts is reduced in comparison to the previous model results for the primary energies (Fig. 5.16a).

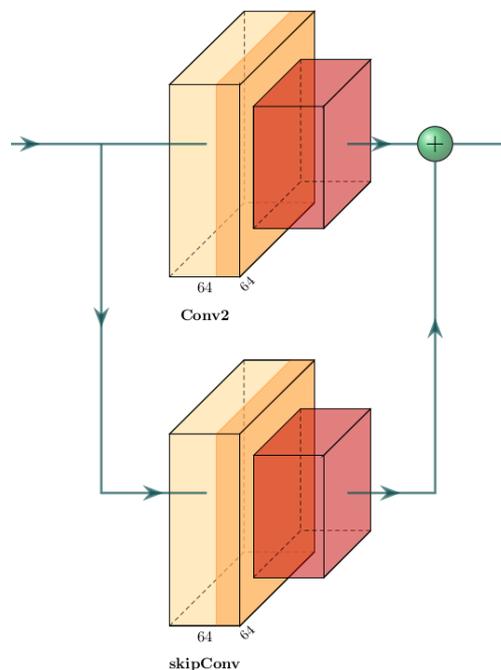


Figure 5.22: Skip connection introduced in the MultiResNet architecture. Here an output from the first convolutional block is used as an input for the skip connection convolutional block. The output of the skip connection is then added to the output of the second convolutional block. Further propagation of the data through the model is unchanged: the summed output is flattened and passed through the fully connected layers.

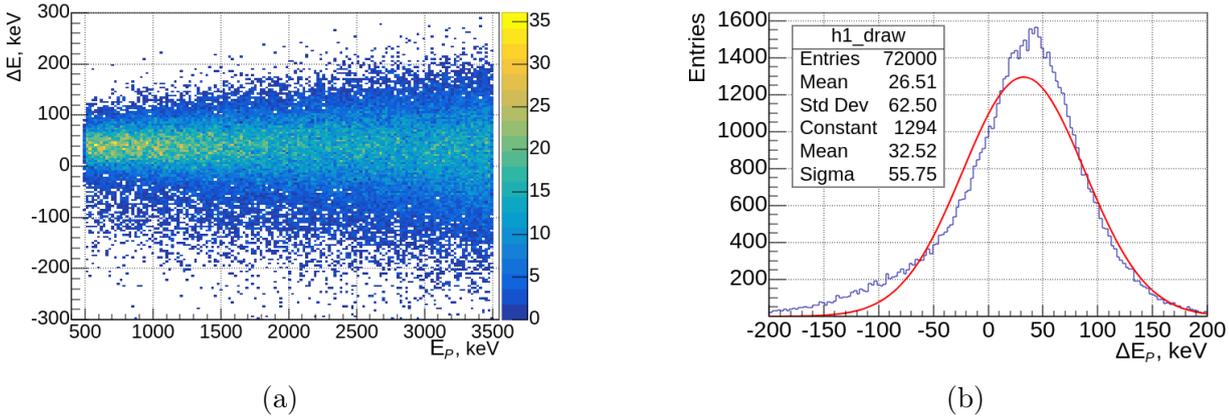


Figure 5.23: MultiResNet model evaluation. Trained on the 300k dataset with global normalization of input images. (a) Correlation plot for the residual between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ with respect to $E_{\mathfrak{P}}$. (b) Residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$.

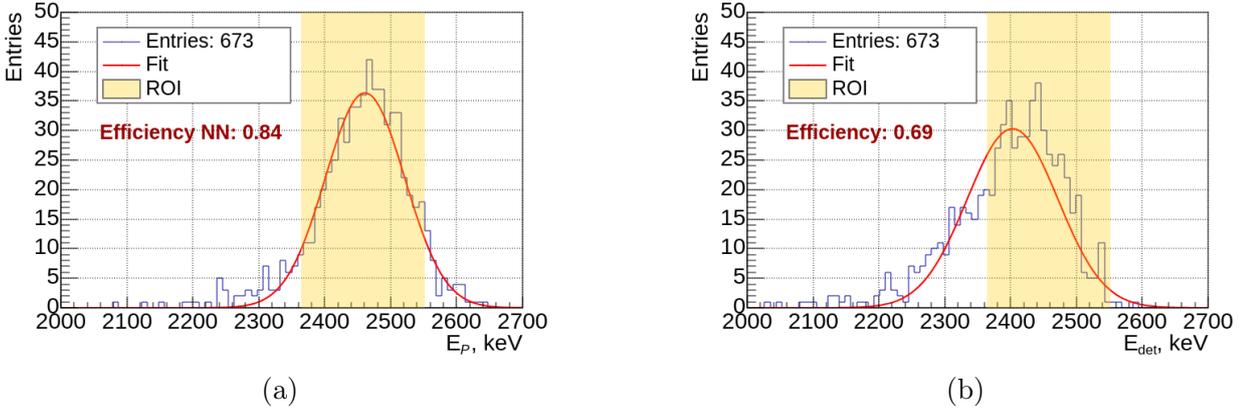


Figure 5.24: Efficiency of the events' prediction in the ROI for the MultiResNet model. (a) $\hat{E}_{\mathfrak{P}}$ in ROI. The efficiency of events' prediction in the ROI is 84%. (b) Efficiency estimation for the direct sum of amplitudes from images with missing channels (Detection efficiency is 69%). In this way we can compare the improvement the model brings to the energy reconstruction.

However, we should note that the residual between $E_{\mathfrak{D}}$ and $\hat{E}_{\mathfrak{D}}$ is also improved compared to the previous results. The sigma is around 50 keV.

Even more importantly, the efficiency of the events' prediction in the ROI is 84% as presented in Fig. 5.24a, which is a great achievement. Thus, we can conclude the effectiveness of the MultiResNet model. The problem, however, is that we still observe a mean shift in the residual distribution for $\hat{E}_{\mathfrak{P}}$, and the standard deviation is relatively high. At this stage, we decide to use another loss function to improve the model's performance.

5.5.5 Different loss functions

The loss function is a crucial part of the model training. The choice of the loss function can significantly affect the model's performance, particularly for multitask learning.

From Fig. 5.23a and Fig. 5.23b, we observe the presence of outliers in the residual distribution. The outliers are the predicted events that deviate significantly from the true values.

There are several reasons for the presence of outliers, such as the noise in the data, the model's inability to capture the complex patterns in the data, the model's overfitting, etc. We are not immune to the presence of non-trivial and non-linear patterns in the data input and the model's inability to capture them. The outliers can significantly affect the performance of the model, and the squared error loss is sensitive to the outliers, which could explain the mean shift in the residual distribution. MSE gives a much higher penalty to outliers compared to smaller errors. A single outlier can significantly skew the overall loss value and potentially mislead the optimization process during training. To address this issue, we can try different loss functions less sensitive to the outliers. The Huber loss and the LogCosH loss are two such loss functions that are more robust choices when dealing with outliers in the data [116].

5.5.5.1 Huber loss try

The Huber loss is a loss function less sensitive to outliers in the data than the squared error loss. It transitions from a quadratic function (similar to the MSE) for small errors to a linear function for larger errors. This reduces the penalty for outliers, allowing the model to focus on learning from the majority of the data that does not contain outliers. The Huber loss combines the squared error loss and the absolute error loss.

It is defined in Eq. 5.1:

$$L_{\delta}(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta \\ \delta(|y - f(x)| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \quad (5.1)$$

where y is the true value, $f(x)$ is the predicted value, and δ is a hyperparameter that determines the threshold for the absolute error loss.

There were several attempts to train the model with the Huber loss, which were far from perfect. The mean shift in the residual distribution is still present, however, varying with the δ value.

5.5.5.2 LogCosH loss try

Again, as the Huber loss, the LogCosH loss is less sensitive to outliers in the data than the squared error loss. Moreover, the LogCosH is twice differentiable everywhere [116], unlike Huber loss (differentiable only once), which is beneficial for the optimization process during training using Adam / AdamW algorithms (see Section 4.2.2.1).

The LogCosH loss is defined in Eq. 5.2:

$$L_{\text{LogCosH}}(y, f(x)) = \log(\cosh(f(x) - y)) \quad (5.2)$$

where y is the true value, and $f(x)$ is the predicted value.

The LogCosH loss is a smooth approximation of the Huber loss at $\delta = 1$ as presented in Fig. 5.25. Considering that the AdamW optimizer is employed at the later stages of the project in order to improve the model's performance, the LogCosH loss is chosen as the main loss function for the model training.

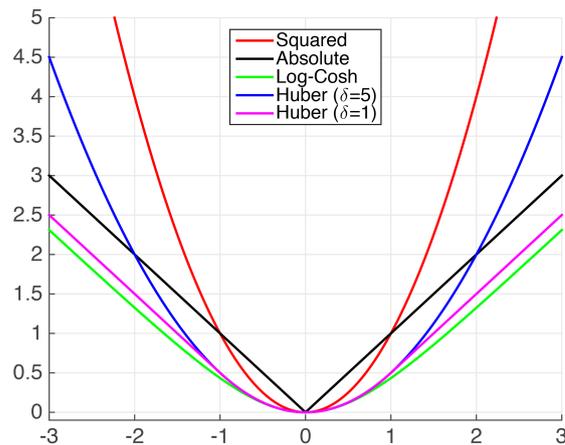


Figure 5.25: Comparison of the MSE (Squared Error), MAE (Absolute Error), LogCosH loss and Huber loss for $\delta = 1$ & 5 functions. X-axis represents "error" of prediction, while Y-axis represents the loss value. Sourced from CS4780/CS5780: Machine Learning [FALL 2015] course at Cornell University.

5.6 Final model

After the multiple optimizations and adjustments applied to the model itself and the training process in general, the final architecture and the training algorithm are chosen.

The final model, Multi3ResNet, could be described as follows: it contains three convolutional blocks with an equal amount of filters in each block, 64 to be precise. The kernel size is 3×3 for each convolutional layer. As before, the activation function is the Rectified Linear Unit (ReLU) applied on each convolutional layer. Then, the Average Pooling layer is applied with a 2×2 kernel. It is changed from the Max Pooling layer, with the idea of preserving more information from the input data during the propagation through the network. The skip connection that passes the information from the first convolutional block goes through an additional Average Pooling layer with 4×4 kernel and then is concatenated with the output of the third convolutional block. After the concatenation, the Flatten layer is applied, followed by the three separate fully connected blocks for each output. Each of these blocks contains two fully connected hidden layers with 64 and 16 neurons, respectively, which is concluded with the output layer. The ReLU activation function is applied on the first fully connected layer. The outputs from the first two fully connected blocks are the familiar $E_{\mathfrak{F}}$ and $E_{\mathfrak{D}}$, while the output from the third one is considered to be the initial Z position of the track, Z_0 . It is added to the network as it should not disturb the energy prediction, which will be discussed later in the chapter.

The optimization algorithm is the AdamW optimizer, with the learning rate of $2 \cdot 10^{-2}$, and the loss function is the LogCosH loss. The data statistics are expanded up to 300k data points for the training in order to improve the model's performance. The model is now trained for 100 epochs with a batch size of 128.

The model architecture is depicted in Fig. 5.26 followed by precise description of the layers structure in the Table 5.3.

Layer (type)	Output Shape	Attribute	N_p
Input	(batchSize, 128, 128, 2)		0
Conv2D	(batchSize, 128, 128, 64)	$5 \times 5, 64$	3264
Activation	(batchSize, 128, 128, 64)		0
AveragePooling2D	(batchSize, 64, 64, 64)	2×2	0
Skip_AveragePooling2D	(batchSize, 16, 16, 64)	4×4	0
Conv2D	(batchSize, 64, 64, 64)	$3 \times 3, 64$	18496
Activation	(batchSize, 64, 64, 64)		0
AveragePooling2D	(batchSize, 32, 32, 64)	2×2	0
Conv2D	(batchSize, 32, 32, 64)	$3 \times 3, 64$	18496
Activation	(batchSize, 32, 32, 64)		0
AveragePooling2D	(batchSize, 16, 16, 64)	2×2	0
Sum	(batchSize, 16, 16, 64)		0
Flatten	(batchSize, 16384)		0
Dense	(batchSize, 64)		8388736
Activation	(batchSize, 64)		0
Dense	(batchSize, 16)		1040
Dense	(batchSize, 1)		17
Dense	(batchSize, 64)		8388736
Activation	(batchSize, 64)		0
Dense	(batchSize, 16)		1040
Dense	(batchSize, 1)		17
Dense	(batchSize, 64)		8388736
Activation	(batchSize, 64)		0
Dense	(batchSize, 16)		1040
Dense	(batchSize, 1)		17

Table 5.3: Multi3ResNet architecture. Attribute for convolutional layers stands for the kernel shape and the number of kernels, while for pooling layers it is kernel shape. N_p stands for the number of parameters in the layer. The Skip_AveragePooling2D layer is the additional Average Pooling layer that is applied to the skip connection.

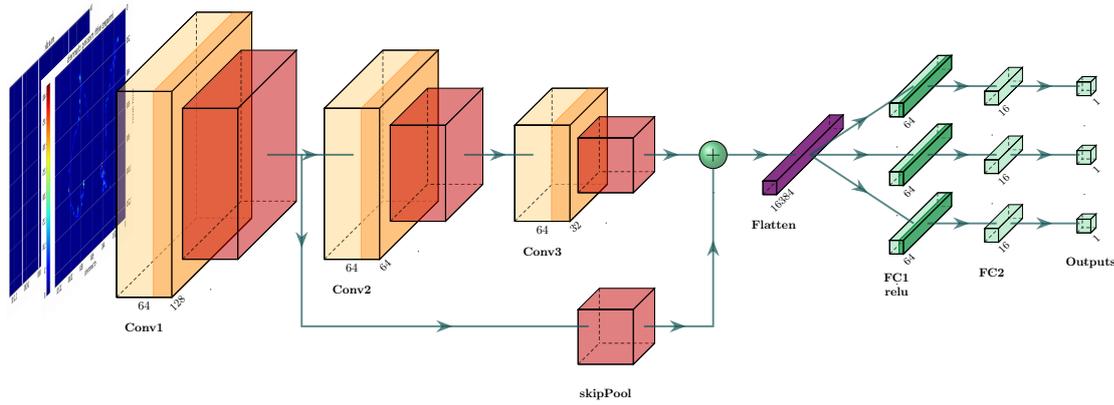


Figure 5.26: Multi3ResNet architecture. Color scheme: yellow - convolutional layers with 64 filters each, orange - ReLU activation function, red - Average Pooling layers with 2×2 kernel (separate *skipPool* Average Pooling layer with 4×4 kernel), purple - flatten layer, light green - fully connected layers, dark green - ReLU activation function.

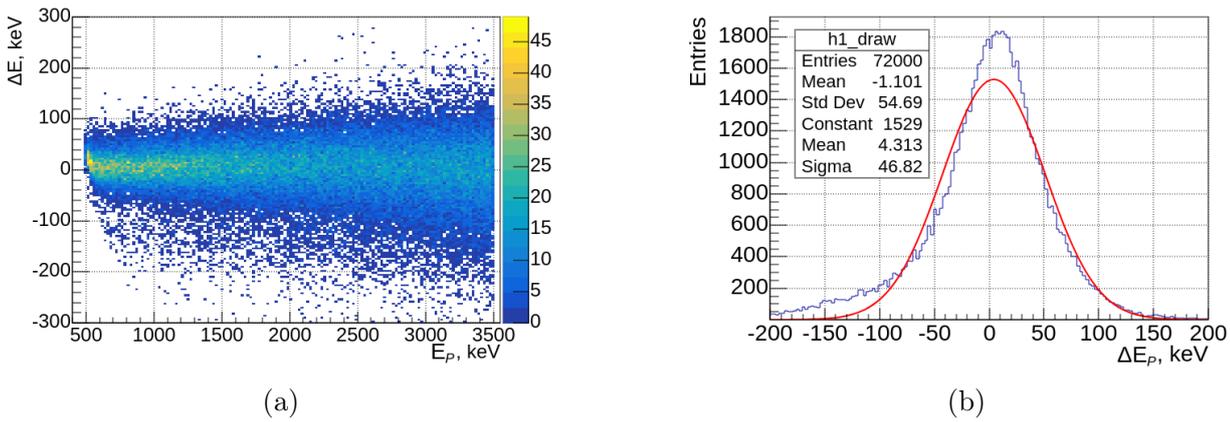


Figure 5.27: Multi3ResNet evaluation. Trained on the 300k dataset with global normalization of input images. (a) Correlation plot for the residual between E_p and \hat{E}_p with respect to E_p . (b) Residual distribution between E_p and \hat{E}_p .

5.6.1 Primary energy prediction evaluation

The model is evaluated on the 72k dataset, containing the 1e data. The results obtained for the primary energy prediction show an improvement. From the Fig. 5.27a and Fig. 5.27b we observe that the mean shift is close to negligible. The standard deviation that comes from the Gaussian fit is about 50 keV. The efficiency of the events' prediction in the ROI is 89%, as one can see in Fig. 5.28. It is the best result so far for the model trained on the 1e dataset with [500, 3500] keV range for E_p .

Now, it is necessary to evaluate the model on the two-electron data which is of our main interest in this study.

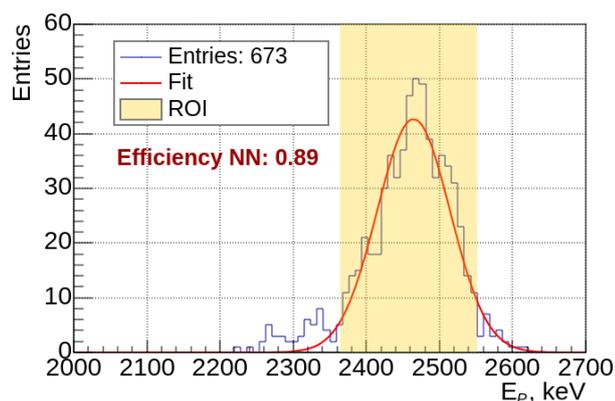


Figure 5.28: Efficiency of the events' prediction in the ROI for the Multi3ResNet model. $\hat{E}_{\mathfrak{N}}$ in ROI. The efficiency of events' prediction in the ROI is 89%.

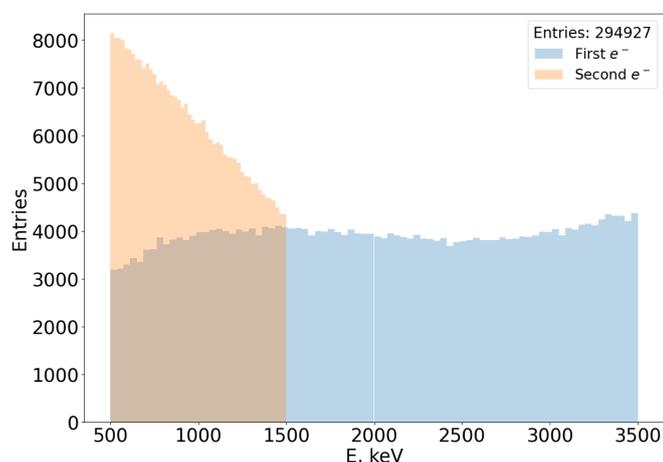


Figure 5.29: The energy distributions corresponding to the primary energies of each of the electrons in the dataset containing $2e$ events. The dataset is used only for evaluation purposes.

5.6.2 Primary energy prediction evaluation on the $2e$ dataset

A dataset containing events that emulate Double Beta events is generated to evaluate the model. The principle to generate the dataset is the same as for the $1e$ dataset but with the addition of a second electron. Both electrons are considered the primary particles with the same vertex position. The goal is to introduce the topological features corresponding to $2e$ events and to evaluate the model's performance on such data. To produce the $2e$ dataset, the 1st electron follows the same distribution as the $1e$ dataset, and the 2nd electron is added with energy in a flat range of 500-1500 keV before the "one MM module" selection. The reader could observe the distributions corresponding to the primary energies of each of the electrons after applied cuts and the total energy distribution in Fig. 5.29. We also note that the dataset contains $\sim 300k$ events in total.

From the evaluation of the Multi3ResNet on the $2e$ dataset, the results for $E_{\mathfrak{N}}$ prediction are completely different from those of the $1e$ dataset. They are shown in Fig. 5.30a and Fig. 5.30b. The correlation between $\hat{E}_{\mathfrak{N}}$ and $E_{\mathfrak{N}}$ is very bad, and consequently, the residual

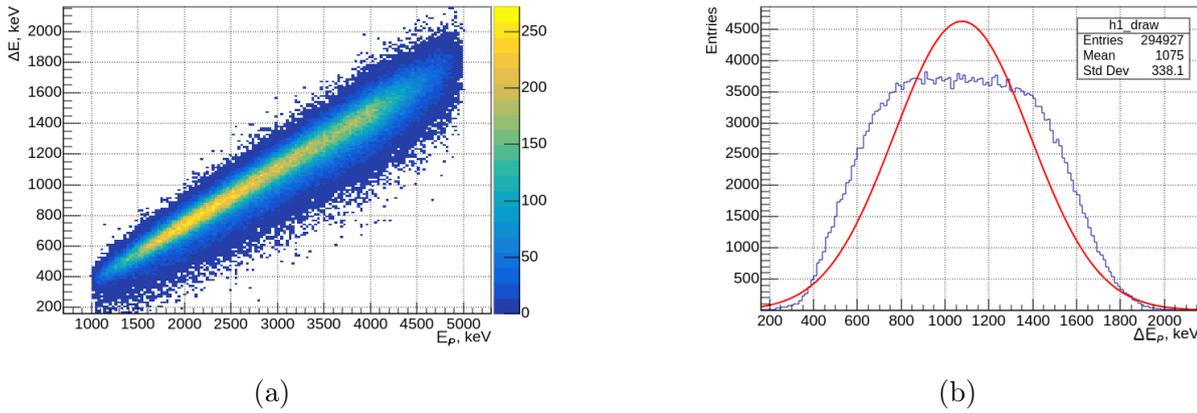


Figure 5.30: Multi3ResNet trained on the 1e dataset. Plots represent the evaluation on the 2e dataset. (a) Correlation plot for the residual between $E_{\mathcal{P}}$ and $\hat{E}_{\mathcal{P}}$ with respect to $E_{\mathcal{P}}$. (b) Residual distribution between $E_{\mathcal{P}}$ and $\hat{E}_{\mathcal{P}}$.

distribution is not Gaussian at all. However, one would likely expect such results, as the model was not trained to generalize the data containing multiple primary particles, and as previously mentioned, the CNN model is sensitive to the topological features of the data. Thus, it is mandatory to train the model on a dataset containing 1e and 2e events.

5.6.3 Additional Multi3ResNet training on the 1e+2e dataset

Finally, to address the issue of the model's performance on the 2e dataset, a dataset containing the 1e and 2e events is generated. It is essential to mention that for the current study, the 2e events energy spectrum differs from the 1e events energy spectrum. Comparing them in Fig. 5.31, one can see that the 2e energy spectrum is not flat and has a larger range of primary energies. In this dataset, the number of 1e events is almost equal to the number of 2e events, $\sim 300k$ each, totaling 600k events. The data representation must be adjusted for further studies. However, the current focus is on the model's performance, which is trained on the present dataset.

The training procedure is repeated with this 1e+2e dataset, using the same hyperparameters and the same architecture as before. Nevertheless, now 80% of the dataset is used for training, and validation and testing datasets contain 10% of the data each. After 100 training epochs, the model is evaluated on the test dataset containing the 1e and 2e events. The results for the Primary energies are shown in Fig. 5.32a and Fig. 5.32b. For the energies in the range higher than 1 MeV, there is a dispersion in the correlation between $E_{\mathcal{P}}$ and $\hat{E}_{\mathcal{P}}$. The lack of 1e events in the primary energy distribution higher than 3.5 MeV is also noticeable.

Nevertheless, even with the dispersion, the residual distribution between $E_{\mathcal{P}}$ and $\hat{E}_{\mathcal{P}}$ shows that the mean value is close to zero point from the Gaussian fit, while the standard deviation is about 60 keV (Fig. 5.32b). Such results are pretty close to the results for the 1e dataset. Moreover, the efficiency of the events' prediction in the ROI is 89%, as one can see in Fig. 5.33, which confirms the model's performance.

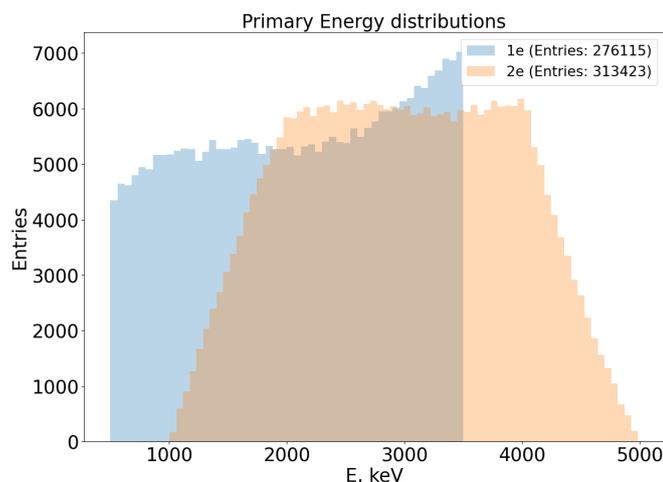


Figure 5.31: 1e and 2e energy spectra in the combined 1e+2e dataset. The 1e energy spectrum is flat, while the 2e energy spectrum is a sum of two primary energies, depicted in Fig. 5.29.

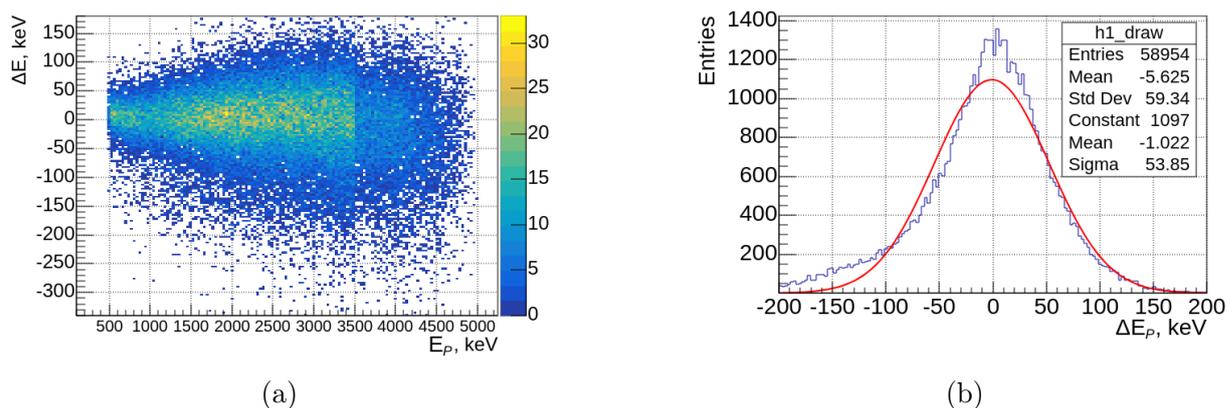


Figure 5.32: Multi3ResNet trained on 1e+2e dataset evaluation. (a) Correlation plot for the residual between E_p and \hat{E}_p with respect to E_p . (b) Residual distribution between E_p and \hat{E}_p .

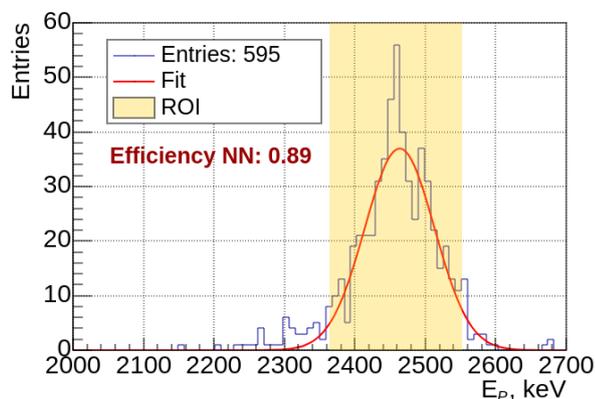


Figure 5.33: Efficiency of the events' prediction in the ROI for the Multi3ResNet trained on 1e+2e dataset. \hat{E}_p in ROI. The efficiency of events' prediction in the ROI is 89%.

5.6.4 Evaluation on the 0 $\nu\beta\beta$ dataset

After confirming the model's performance on the 1e+2e dataset, the model is evaluated on a 0 $\nu\beta\beta$ dataset. $\sim 250\text{k}$ 0 $\nu\beta\beta$ events of the Xe136 are generated within REST (as can be recalled from Section 2.3.4.2) for the evaluation of the model.

All the $E_{\mathfrak{F}}$ for the 0 $\nu\beta\beta$ events are at 2.46 MeV. Therefore, the correlation between $E_{\mathfrak{F}}$ and $\hat{E}_{\mathfrak{F}}$ is not informative. The residual distribution between $E_{\mathfrak{F}}$ and $\hat{E}_{\mathfrak{F}}$ is shifted, however. From the fit, we observe the mean shift from the zero point of ~ 30 keV, while the standard deviation is again about 60 keV (Fig. 5.34).

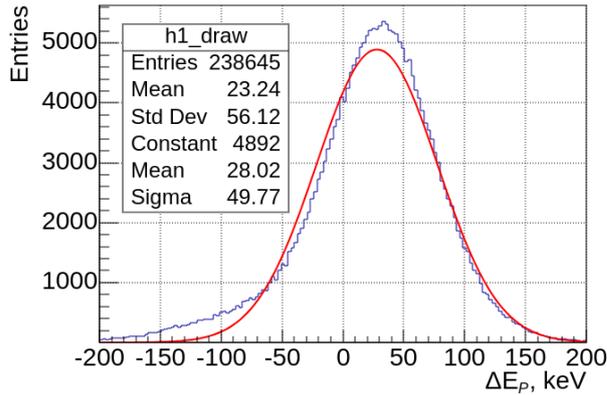


Figure 5.34: Residual distribution between $E_{\mathfrak{F}}$ and $\hat{E}_{\mathfrak{F}}$ from the Multi3ResNet model evaluation on the 0 $\nu\beta\beta$ dataset. The model is trained on the 1e+2e dataset.

The efficiency of the events' prediction in the ROI is 90% from Fig. 5.35a. By comparing the results obtained with the Multi3ResNet model with the direct extraction of event energy from the amplitudes of the signals, we can see that the efficiency of the events' prediction in the ROI increased from 69% to 90% (Fig. 5.35b). 20% increase in the prediction efficiency is a significant improvement and should improve the sensitivity of the 0 $\nu\beta\beta$ search.

Additionally, the model's performance is evaluated on the 1e+2e and 0 $\nu\beta\beta$ decay datasets, unaffected by the missing channels. Current results show that the RMS on the residual distribution between $E_{\mathfrak{F}}$ and $\hat{E}_{\mathfrak{F}}$ is reduced compared to the data evaluation with missing channels. The RMS for the 1e+2e and 0 $\nu\beta\beta$ datasets are 44 keV and 43 keV, respectively (residual distribution between $E_{\mathfrak{F}}$ and $\hat{E}_{\mathfrak{F}}$ is presented in Fig. 5.36a). However, the mean shift is increased significantly up to ~ 100 keV for both datasets. This shift is responsible for degrading the events' prediction efficiency in the ROI to 33% in both cases. In case of shift correction, the efficiency increases to 96%, which can be observed in Fig. 5.36b showcasing the $\hat{E}_{\mathfrak{F}}$ in the ROI for 0 $\nu\beta\beta$ events.

Such a significant shift in the mean value of the residual distribution for $E_{\mathfrak{F}}$ and $\hat{E}_{\mathfrak{F}}$ requires further investigation. It may be related to the model's biasing towards the data with missing channels.

5.7 Z position prediction

Determining the Z_0 is an additional task for the model, and it is based on the diffusion of charge in the detector, which correlates with the distance from the interaction point.

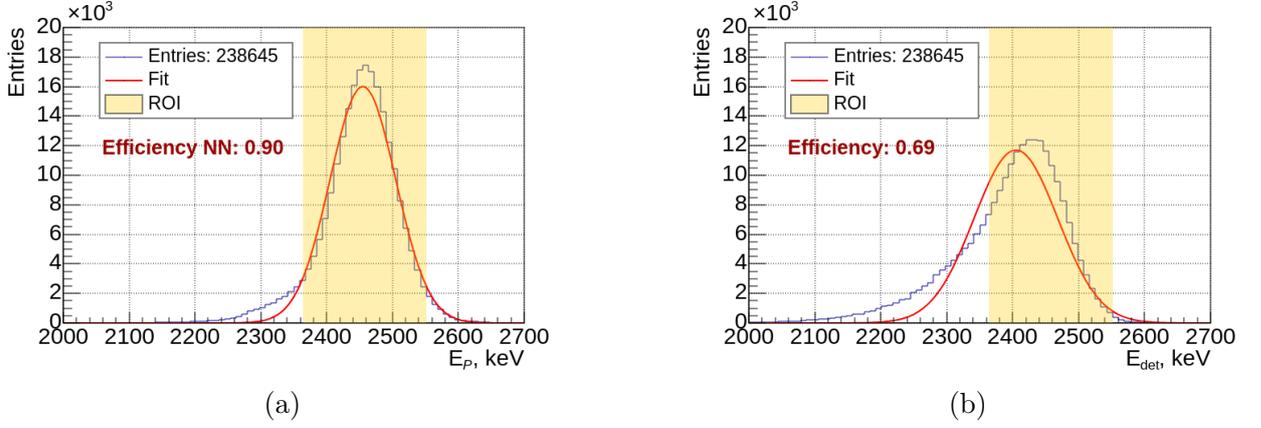


Figure 5.35: Efficiency of the events' prediction in the ROI for the Multi3ResNet trained on $1e+2e$ dataset. (a) $\hat{E}_{\mathcal{N}3}$ of the $0\nu\beta\beta$ events. Efficiency of events' prediction in the ROI is 90%. (b) Efficiency of the direct extraction from the sum of amplitudes.

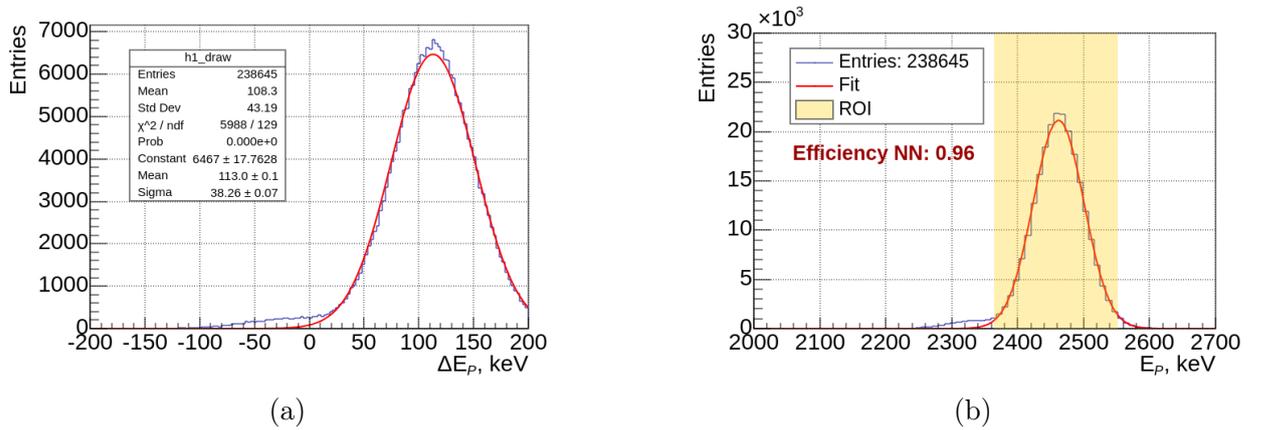


Figure 5.36: Multi3ResNet evaluation results on the $0\nu\beta\beta$ dataset without missing channels. (a) Residual distribution between $E_{\mathcal{N}3}$ and $\hat{E}_{\mathcal{N}3}$. (b) $\hat{E}_{\mathcal{N}3}$ with the mean shift correction. The efficiency of events' prediction in the ROI is 33% and 96% for the uncorrected and corrected mean shift, respectively.

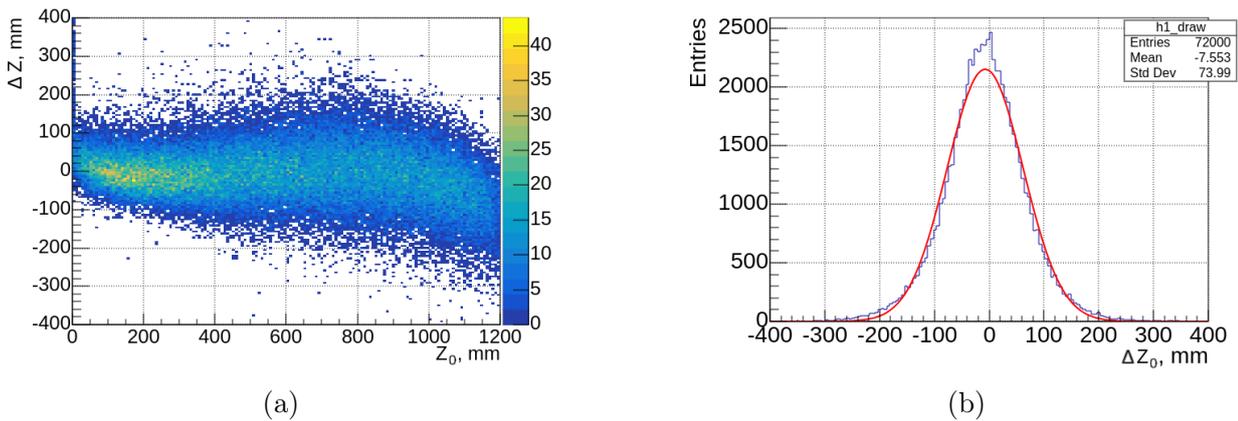


Figure 5.37: \hat{Z}_0 evaluation for the Multi3ResNet model trained on the 1e dataset. (a) Correlation plot for the residual between Z_0 and \hat{Z}_0 with respect to Z_0 . (b) Residual distribution between Z_0 and \hat{Z}_0 .

It's important to note that predicting the Z position does not impact the primary and detected energy predictions, as it depends on the track's topological characteristics and is less influenced by any missing channels.

5.7.1 Evaluation of the model trained on the 1e dataset

The Multi3ResNet model trained on the 1e data (see Section 5.6) is first evaluated on the 1e dataset (containing 72k events) and later on the 2e dataset (of ~ 300 k events). In these simulated datasets, the detector response and the gas configuration at the detector are the same and proximate to the real PandaX-III experiment conditions. For the 1e dataset evaluation, the correlation plot for the residual between Z_0 and \hat{Z}_0 coordinates shows that for the Z_0 in the range from 200 to 1000 mm, the values are mostly correlated with the Z_0 , as one can see in Fig. 5.37a. The problem is that the correlation is not perfect for the events closer to the anode (Z_0 less than 200 mm) and to the cathode (Z_0 greater than 1000 mm), and there is a structure in the residual vs. Z_0 correlation at the edges. Nevertheless, the residual distribution between Z_0 and \hat{Z}_0 shows that the mean shift is close to negligible, while the crude Gaussian fit provides a standard deviation of about 70 mm, as one can see in Fig. 5.37b.

Even though the model is trained only on the 1e data, the Z prediction on the 2e data shows the same pattern. The residual distribution for Z_0 and \hat{Z}_0 coordinates have the same mean shift and the standard deviation of 70-75 mm (Fig. 5.38b).

Such results provide evidence that the model is capable of learning the topological features of the data related to the Z position and is unaffected by the number of primary particles present in the event.

5.7.2 Evaluation of the model trained on the 1e+2e dataset

Here, results for \hat{Z}_0 evaluation for the Multi3ResNet model trained on the 1e+2e dataset are present (recalling Section 5.6.3). As for the $E_{\mathcal{P}}$ prediction, the model is evaluated first on the testing dataset containing combined 1e and 2e events and later on the $0\nu\beta\beta$ dataset.

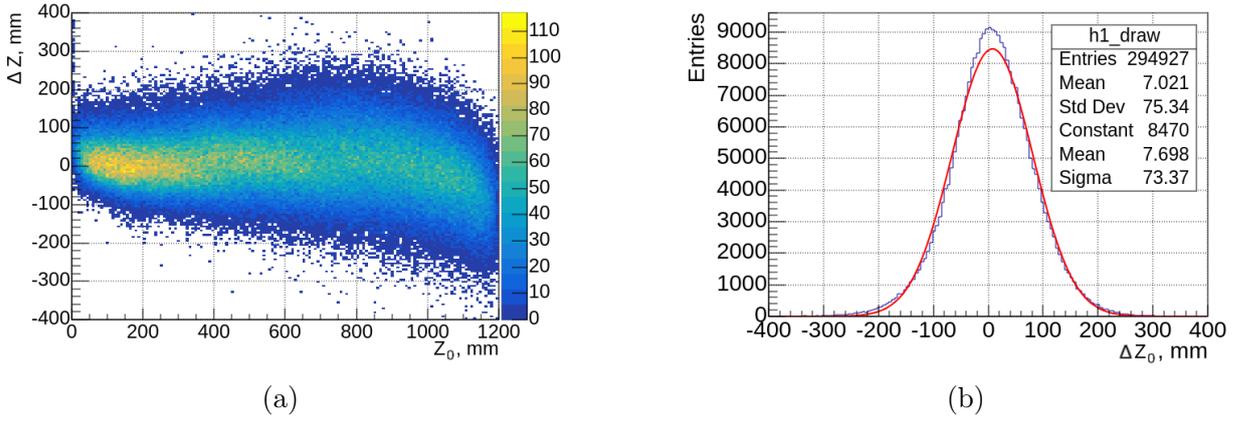


Figure 5.38: \hat{Z}_0 evaluation for the Multi3ResNet model trained on the 1e dataset. Evaluation on the 2e dataset. (a) Correlation plot for the residual between Z_0 coordinate and \hat{Z}_0 coordinate with respect to Z_0 . (b) Residual distribution between Z_0 and \hat{Z}_0 .

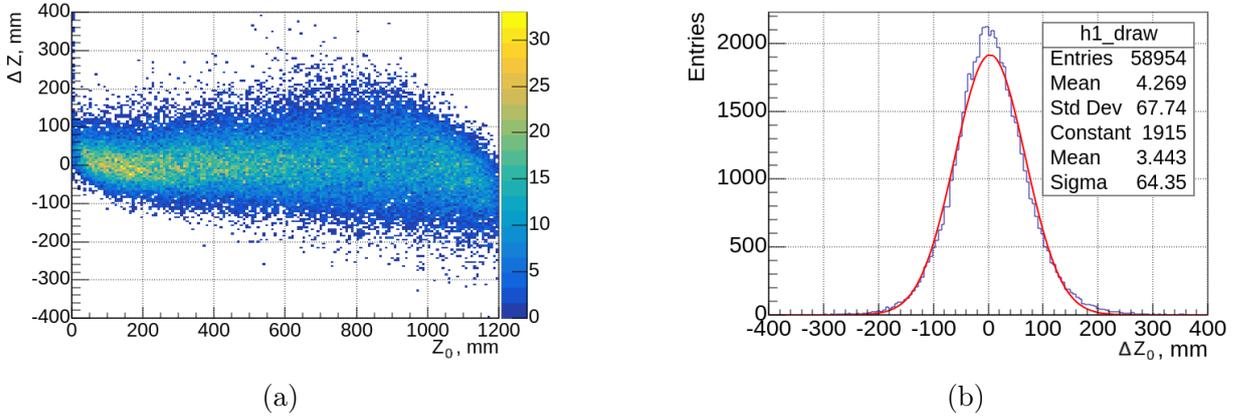


Figure 5.39: \hat{Z}_0 evaluation for the Multi3ResNet model trained on the 1e+2e dataset. Evaluation on the 1e+2e testing dataset. (a) Correlation plot for the residual between Z_0 and \hat{Z}_0 with respect to Z_0 . (b) Residual distribution between Z_0 and \hat{Z}_0 .

For the former evaluation, we do not observe a significant difference in the results for the \hat{Z}_0 compared to the results obtained for the 1e dataset. The correlation plot for the residual between Z_0 and \hat{Z}_0 in Fig. 5.39a follows the same pattern as in the case of the 1e dataset (Fig. 5.37a). Fig. 5.39b shows a slight improvement in the standard deviation of the residual distribution, which is about 65 mm.

When we do the prediction on the $0\nu\beta\beta$ data, the results are, as expected, similar to the 1e+2e prediction, as one can see in Fig. 5.40a and Fig. 5.40b. The residual distribution is shifted by the mean value of about 10 mm, while the standard deviation is still close to 65 mm.

As it was addressed in Section 5.6.4, the model's performance is also evaluated on the 1e+2e and $0\nu\beta\beta$ decay datasets without missing channels. The mean value of the residual distribution between Z_0 and \hat{Z}_0 for both datasets possesses a slight shift of ~ 30 mm, while the standard deviation stays at the level of 65 mm.

Previously, Li et al. [113] have shown that the VGGZ0net model was capable of predicting the Z position with the standard deviation of 110 mm for the Monte Carlo simulated

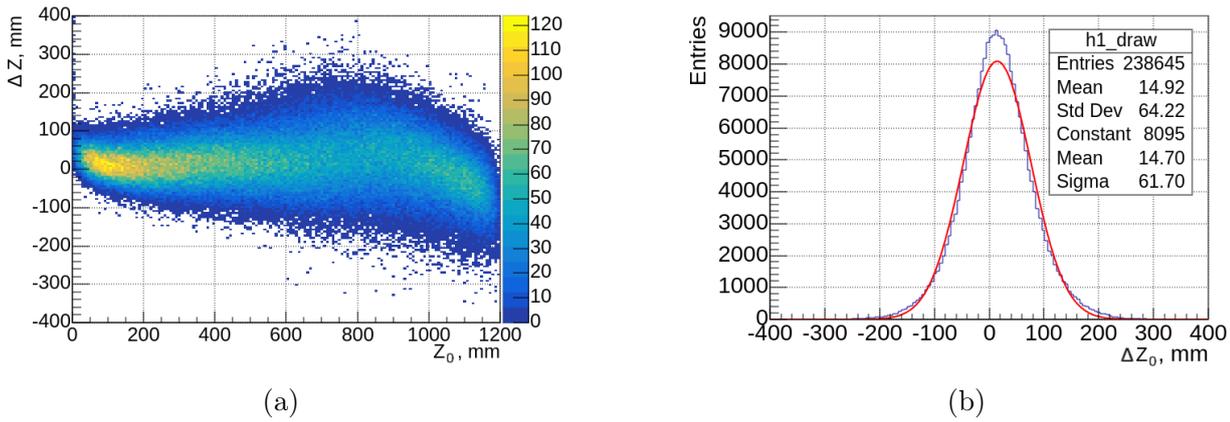


Figure 5.40: \hat{Z}_0 evaluation for the Multi3ResNet model trained on the $1e+2e$ dataset. Evaluation on the $0\nu\beta\beta$ dataset. (a) Correlation plot for the residual between Z_0 and \hat{Z}_0 with respect to Z_0 . (b) Residual distribution between Z_0 and \hat{Z}_0 .

$0\nu\beta\beta$ events in the PandaX-III experiment setup. In that study, the electron attachment effect was considered, which is different from the present study. However, the results still show promising performance of the Multi3ResNet model, given the fact that the presence of missing channels degrades input data.

5.8 Summary

The most prominent results of this study of the regression problem of the primary energies reconstruction from the images of the track topologies with missing channels are collected in Table 5.4. The table shows a gradual improvement of the CNN approach, starting from the simple toyCNN model test, through the MultiCNN and MultiResNet models for which the data with missing channels is introduced, to the final Multi3ResNet model. The latter, being trained on the $1e+2e$ dataset, is thoroughly evaluated, including on the $0\nu\beta\beta$ events, and shows very good performance with efficiencies of 89%-90%.

In addition, the Multi3ResNet model is evaluated on the Z_0 coordinates prediction. From the final results, the standard deviation of the residual distribution is around 65 mm for the $1e+2e$ and 61 mm for the $0\nu\beta\beta$ datasets. The prediction of Z_0 is not affected by the topological features of the data and the presence of multiple primary particles in the event.

5.9 Conclusions

Implementing a CNN model for the primary energy prediction (together with the Z initial coordinate) from the track images with missing channels is, in fact, possible. The Multi3ResNet model provides a 20% increase in the efficiency of the events' prediction in the ROI for the $0\nu\beta\beta$ events, compared to the direct extraction of the event energy from the sum of amplitudes. Indeed, it would be necessary to evaluate the model based on the data available from the PandaX-III experiment to confirm its performance. An example of that might be the calibration data from the detector. On the other hand, the model must be trained on the proper configuration of the missing channels of the detector readout in

Model	Full dataset	Test fraction [%]	Efficiency	σ [keV]
toyCNN (perfect MMs)	1e (80k)	25	-	65
MultiCNN	1e (300k)	20	79%	71
MultiResNet	1e (300k)	20	84%	62
Multi3ResNet	1e (300k)	20	89%	54
	1e+2e (600k)	10	89%	60
	$0\nu\beta\beta$ (250k) eval.	-	90%	56
	$0\nu\beta\beta$ w/o miss.	-	33%	43

Table 5.4: Comparison of the model’s performance in predicting the primary energies and. The training datasets and the test fractions for the model evaluation are provided. The last two columns correspond to the efficiency of the events’ prediction in the ROI and the standard deviation of the residual distribution between the true and predicted primary energies. The toyCNN was trained and evaluated only on the perfect readout configuration without missing channels in this study. The table also includes the evaluation of the Multi3ResNet model on the $0\nu\beta\beta$ dataset with and without missing channels. The drop in the efficiency of the events’ prediction for the $0\nu\beta\beta$ dataset without missing channels is due to the mean shift in the residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ which has to be investigated.

order to be relevant for the experiment.

The Multi3ResNet model evaluation on the data without missing channels shows a mean shift in the residual distribution between $E_{\mathfrak{P}}$ and $\hat{E}_{\mathfrak{P}}$ which hints at the model’s biasing towards the degraded data. This issue requires an additional study on the simulated data with the readout modules configuration where the number of missing channels ranges from 0 to 3.

In further studies, which are not covered in my work, the consideration of the events detected by the multiple Micromegas modules and the implementation of the inhomogeneity of the readout gain, in general, should be taken into account.

Regarding the Z position prediction, the Multi3ResNet model is capable of predicting the Z initial coordinates with a promising standard deviation on the residual distribution between true and predicted Z coordinates. Compared to the Li et al. [113] study, the Multi3ResNet model outperforms the VGGZ0net model in the Z position prediction. However, these results should be taken with caution, as the electron capture effect was not implemented in my study’s Monte Carlo simulation of the data.

In the next chapter, I will discuss the classification problem, which tackles the identification of the 2e events and the background 1e events. The Multi3ResNet architecture will be used as an example of the model for the classification problem.

Chapter 6

Classification Problems

... What should we have ready at hand in a situation like this?
The knowledge of what is mine and what is not mine,
what I can and cannot do.
— *Epictetus, ~108 AD*

6.1	Introduction	127
6.2	CNN approach for the classification of 1e/2e events	128
6.2.1	Data Preparation	128
6.2.2	Model Architecture description	130
6.2.3	Hyperparameter Tuning and Training Process	132
6.3	Results from the Classification Model	134
6.4	Conclusions	139

6.1 Introduction

What is the motivation for this chapter?

In the previous chapter, we discussed how CNN models can be used for the energy reconstruction task, which is a regression problem. The latest trained model showed a good performance reconstructing the energy of the detected particles, based on its topology. The core objective for the model was to reconstruct energy corresponding to the track topologies with missing channels. The model was supposed to be indifferent to the origins of the events: whether they are 1-electron background events coming from the ^{208}Tl or ^{214}Bi decays, 2-electron background events from two neutrino double beta decay of ^{136}Xe , or signal events from the $0\nu\beta\beta$ of ^{136}Xe . Now, we are interested in discrimination of the background events, from the ROI in particular, where we are expecting a high background contamination, which reduces the sensitivity of the PandaX-III experiment for the $0\nu\beta\beta$ detection.

A similar CNN architecture model is employed to resolve this classification problem of discrimination between 2-electron and 1-electron events, learning features related to the topological properties of the tracks. After data and model optimization, the best model is selected based on the performance metrics.

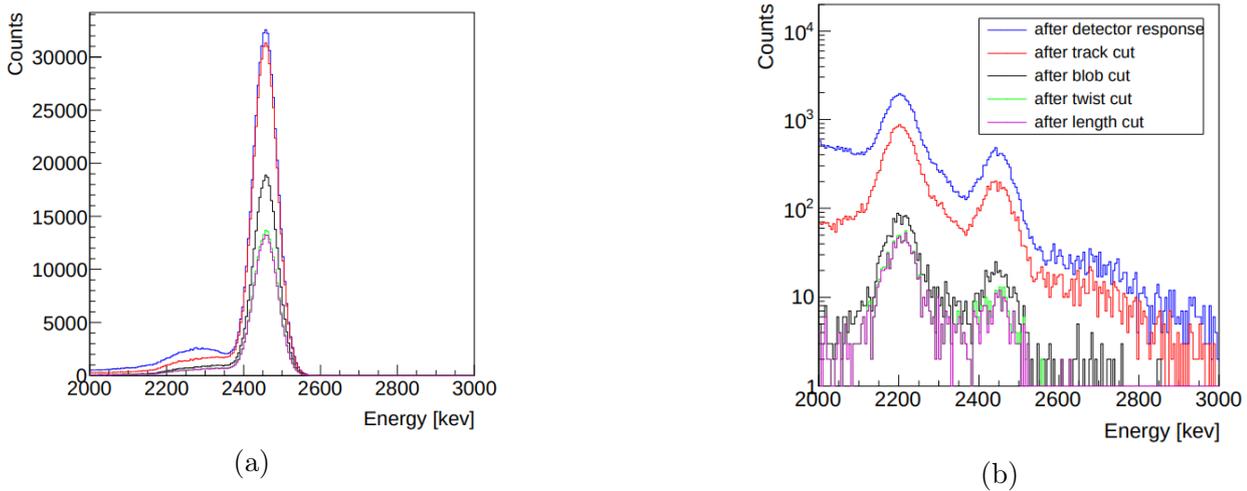


Figure 6.1: Results taken from [73] study on the event discrimination. Plots show the energy distribution of the $0\nu\beta\beta$ (a) and high-energy background (b) events before and after *track*, *Q*, *twist*, and *length* cutoffs. Such approach reduces the background contamination in the ROI by the factor of ~ 100 , while keeping the $0\nu\beta\beta$ signal efficiency at 17.5%.

Previously, Javier Galan with his collaborators proposed a method to use the topological features of the tracks to do the necessary background discrimination [73]. The track features like track energy ratio, end-track energies (Blob charge), end-track twist, and track length were extracted from the tracks, corresponding to the $0\nu\beta\beta$ and high-energy background events expected to contaminate the ROI. After applying the feature cuts, the background event contamination in the ROI was reduced by the factor of ~ 100 , while keeping the $0\nu\beta\beta$ signal efficiency at 17.5% that could be visible on Fig. 6.1.

Instead of using an analytical approach, a CNN model is employed to classify the events in my approach, that is discussed in the next section.

Previously, Hao Qiao with his collaborators used a CNN model to discriminate between $0\nu\beta\beta$ and high-energy background events in [114]. The model showed a good performance with 99.4% background rejection efficiency at 47.5% signal efficiency. The approach that is proposed in this chapter differs in the model’s architecture and the data preparation. Without further ado, let me introduce the CNN model approach for the classification of 1-electron and 2-electron events.

6.2 CNN approach for the classification of 1e/2e events

6.2.1 Data Preparation

In the previous chapter 5, we used 2D images of raw events as input to the network. Even more, the implementation is easier. It requires minimal signal processing in comparison to the analytical approach [73].

The data preparation for the classification model is exactly the same as for the energy reconstruction model, however, there is a particular attention to the 2-electron data events. The topological features of the tracks are proportional to the particle energies, and we should not neglect the potential bias which could appear in case of misrepresentation of events in

certain energy ranges, as it was in the previous chapter.

In order to obtain a balanced datasets for the 1-electron and 2-electron events, the energy spectra of both types of events must have the same energy range and the same flat distribution (Fig. 6.2). Therefore, for the 2-electron events, each of the two electrons is simulated in Geant4 from the same spatial vertex with isotropic angular distribution, and the sum of the primary energies of the two electrons follows a flat distribution. The energy range now is extended to be from 30 keV to 4 MeV, and the total number of entries in the training dataset is $1.2 \cdot 10^6$, with $\sim 50\%$ of the 1-electron events and $\sim 50\%$ of the 2-electron events. The choice to include the low energy events is motivated by the desire to expose the NN model to these events. Missing channels could carry important parts of the information of the event, thus the total sum of the signal amplitudes degrades, yet the topological features of the tracks are still present and could be learned by the model. Thus, the model must distinguish low and high energy events. Indeed, it is hard to distinguish the 1-electron and 2-electron events around 30 keV energy, but such low energy events are present in the datasets for both types of events in order not to introduce a selection bias to the model.

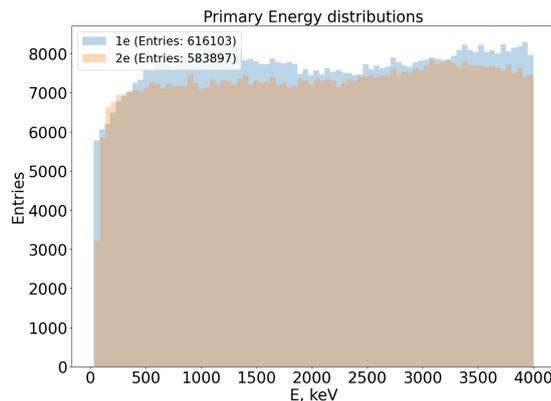


Figure 6.2: Primary energy spectrum of the 1-electron and 2-electron events in the combined 1e+2e dataset. Distribution of primary energies is flat for both types of events and lays in the [30 keV, 4 MeV] energy range. The number of entries in the dataset is $1.2 \cdot 10^6$ equally distributed between the 1-electron and 2-electron events.

As an example, one can see the comparison of the 1-electron and 2-electron events having similar primary energies ~ 2.4 MeV in Fig. 6.3.

As before, a concatenated 2D array of a track event with the map of the missing channels of a total shape of (128, 128, 2) is used as input data for the CNN model. We opted for a 2-output configuration, in order to be prepared for more complex setups where we would have had more than two possible outcomes. However, we haven't pursued the idea, for lack of time. Here, with only two possible outcomes, *viz.* signal and background, a 1-output configuration would have worked fine. For the labels, a *one-hot encoding* is used, where the 1-electron event is represented as [0, 1] and the 2-electron event as [1, 0]. The choice of such labeling is due to the loss function selection which is described in the further sections.

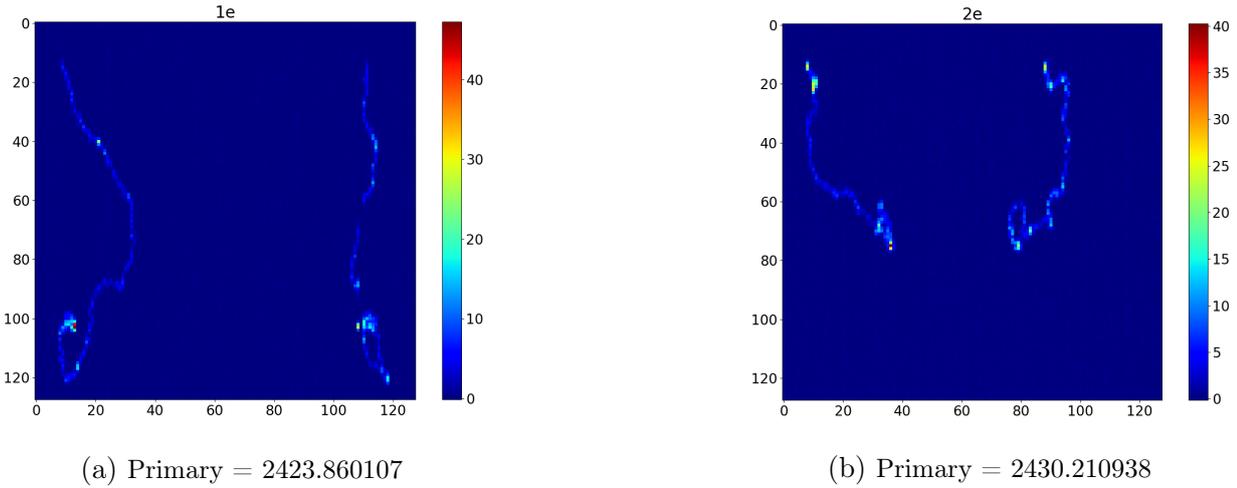


Figure 6.3: Comparison of the 1-electron (a) and 2-electron (b) events input images with similar primary energies ~ 2.4 MeV. The 2-electron event is more complex in terms of the track topology, visibly has a Blob charge at each end of the track, and the track length is significantly shorter than the 1-electron event.

6.2.2 Model Architecture description

The model's architecture used for classification has a similar structure as the final model used for the energy reconstruction task, with some modifications. After a sequence of model optimizations, the best architecture is selected, which we would call *Class3ResNet*. It consists of three consecutive convolutional blocks with an average pooling layer after each block. After the first convolutional block, a skip connection is added containing another convolutional block. The features from the main path and the skip connection are concatenated before flattening the output and passing it through a set of dense layers. Each of the convolutional layers and dense layers is followed by a ReLU activation function. The final output layer contains two outputs without an activation function, as Class3ResNet is trained using the *categorical cross-entropy loss function*. The architecture of the Class3ResNet is shown in Fig. 6.4 followed by the description of each layer in the Table 6.1.

A categorical cross-entropy loss function in a general case is used for the classification tasks, where the output is a probability distribution over the classes, as it was already introduced in Section 4.2.1. The loss function measures the difference between the predicted probability distribution and the true probability distribution:

$$L(\hat{y}, y) = \{l_1, \dots, l_N\}^T, \quad l_i = -w_{y_i} \log \frac{\exp(\hat{y}_{i,y_i})}{\sum_{c=1}^C \exp(\hat{y}_{i,c})}, \quad (6.1)$$

where \hat{y} is the predicted probability distribution, y is the true probability distribution, w_{y_i} is the weight of the class y_i , and C is the number of classes and N is the number of samples in the dataset/mini-batch.

During the training process, the value of the loss is averaged over the mini-batches, so the loss value is not directly comparable between different batch sizes:

$$L = \frac{1}{N} \sum_{i=1}^N L(\hat{y}_i, y_i). \quad (6.2)$$

Layer (type)	Output Shape	Attribute	N_p
Input	(batchSize, 128, 128, 2)		0
Conv2D	(batchSize, 128, 128, 32)	$3 \times 3, 32$	608
Activation	(batchSize, 128, 128, 32)		0
AveragePooling2D	(batchSize, 32, 32, 32)	4×4	0
Skip_Conv2D	(batchSize, 32, 32, 128)	$3 \times 3, 128$	36992
Skip_Activation	(batchSize, 32, 32, 128)		0
Skip_AveragePooling2D	(batchSize, 2, 2, 128)	16×16	0
Conv2D	(batchSize, 32, 32, 64)	$3 \times 3, 64$	18496
Activation	(batchSize, 32, 32, 64)		0
AveragePooling2D	(batchSize, 8, 8, 64)	4×4	0
Conv2D	(batchSize, 8, 8, 128)	$3 \times 3, 128$	73856
Activation	(batchSize, 8, 8, 128)		0
AveragePooling2D	(batchSize, 2, 2, 128)	4×4	0
Sum	(batchSize, 2, 2, 128)		0
Flatten	(batchSize, 512)		0
Dense	(batchSize, 128)		65664
Activation	(batchSize, 128)		0
Dense	(batchSize, 64)		8256
Activation	(batchSize, 64)		0
Dense	(batchSize, 16)		1040
Activation	(batchSize, 16)		0
Dense	(batchSize, 2)		34

Table 6.1: Classification model layer description. Attribute for convolutional layers stands for the kernel shape and the number of kernels, while for pooling layers it is kernel shape. N_p is the number of parameters in the layer. For the convolutional layers, N_p is calculated as $(F * F * C * K) + (K \text{ bias})$, where F is the kernel size, C is the number of channels in the input, and K is the number of kernels. For the dense layers, N_p is calculated as $(N * M) + (M \text{ bias})$, where N is the number of neurons in the previous layer and M is the number of neurons in the current layer.

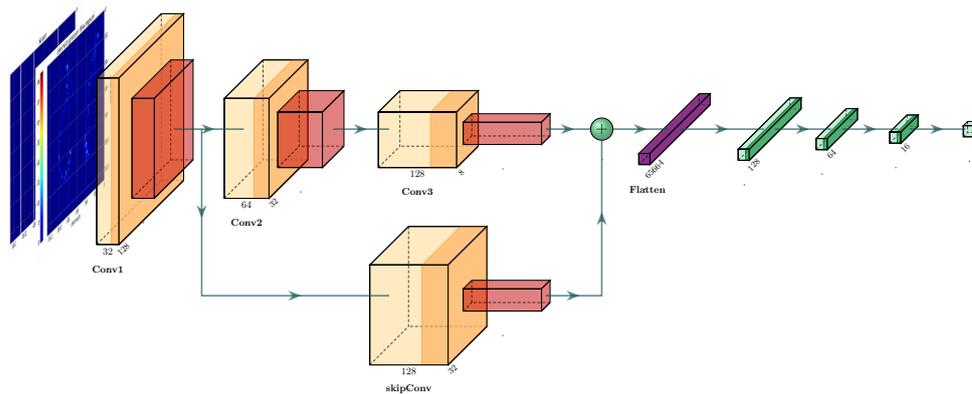


Figure 6.4: Class3ResNet model visualization. The color scheme is the same as in the previous chapter with yellow representing the convolutional layers, orange - ReLU activation functions, red - average pooling layers, purple - flattening layer, green - dense layers with following ReLU activation functions marked as dark green. A skip connection is added after the first average pooling layer, which contains another convolutional block with average pooling layer. The output from the main path and the skip connection is concatenated before flattening the output and passing it through a set of dense layers. The final output layer contains two outputs for the classification task.

Such loss function provides a way to quantify the discrepancy between the model's predicted class probabilities and the true class labels.

6.2.3 Hyperparameter Tuning and Training Process

After increasing the dataset size to $1.2 \cdot 10^6$ entries, the training of the model and the corresponding optimization of the hyperparameters required a significant amount of computational resources.

Hyperparameter tuning [117][118] is an iterative process that involves:

- *Defining a Search Space:* This specifies the range of possible values for each hyperparameter.
- *Choosing an Evaluation Metric:* This metric determines how well the model performs on unseen data. Common metrics include accuracy, precision, recall, F1-score, or loss function.
- *Selecting a Search Technique:* There are various search techniques available:
 - *Grid Search:* Evaluates all possible combinations of hyperparameter values within the defined search space. This can be time consuming for models with many hyperparameters.
 - *Random Search:* Samples random combinations from the search space, offering a more efficient alternative to grid search for large spaces.
 - *Bayesian Optimization:* Uses a probabilistic model to prioritize promising hyperparameter configurations based on past evaluations, making it particularly useful for expensive-to-evaluate models.

- *Training and Evaluating Models*: For each hyperparameter combination, the model is trained on a training set and evaluated on a validation set. The performance metric is recorded.
- *Selecting the Best Hyperparameters*: Based on the evaluation metrics, the hyperparameter combination that yields the best performance on the validation set is chosen.

Using the Bayesian Optimization technique, we obtain more subtle hyperparameters adjustments, which are not possible with the Grid Search or Random Search techniques. The Bayesian Optimization technique is particularly useful for models with a large number of hyperparameters, as it can efficiently explore the search space and find the optimal hyperparameter values.

The hyperparameter optimization process is conducted using the *Hyperopt* library in Python, which provides a flexible and efficient framework for hyperparameter tuning. The algorithm used for the optimization is the Tree-structured Parzen Estimator (TPE), which is a sequential model-based optimization algorithm that uses a probabilistic model to guide the search for the best hyperparameters. The TPE algorithm balances exploration and exploitation to efficiently search the hyperparameter space and find the optimal settings. More information about the algorithm can be found in [119].

The hyperparameters that are optimized include the learning rate, L2 regularization strength, average pooling kernel, and the number of output neurons in the first two dense layers. The hyperparameter tuning process is conducted over 20 iterations with each iteration

a budget of 100 evaluations. AdamW optimizer is used with a batch size of 128 both for the hyperparameter tuning and the later training process. The dataset for the hyperparameter tuning is reduced to 8% of the total dataset to speed up the process. The best hyperparameters are selected based on the validation accuracy metric and they are shown in Table 6.2.

As you can see, the hyperparameter tuning process is used to optimize the model’s architec-

Hyperparameter	Value
Learning Rate	0.0012
L2 Regularization Strength	0.0011
Average Pooling Kernel	4×4
Dense Layer 1 Neurons	128
Dense Layer 2 Neurons	64

Table 6.2: Best hyperparameters obtained from the hyperparameter tuning process. For the main branch of the model, the average pooling kernel is 4×4 . For the skip connection, the average pooling kernel is 16×16 to match the dimensions of the output from the main branch before concatenation.

ture as well as the learning process. The validation accuracy for the best hyperparameters is 0.90.

After the hyperparameter tuning process, the model is trained on the full dataset using the best hyperparameters. Whole dataset is split into training and validation sets with a ratio of 90:10, while separate datasets are generated for the testing. It was observed that training the model on the full dataset overflows the memory of the GPU. Therefore, the training dataset was split into 10 smaller datasets. The model is trained on each dataset separately,

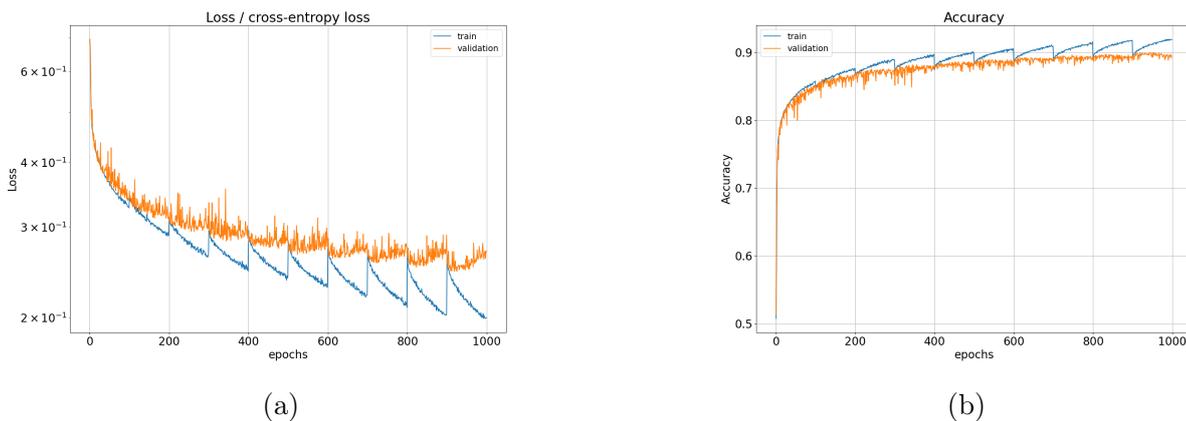


Figure 6.5: Loss curve (a) and accuracy curve (b) for the classification model during the training process. The model is trained consecutively on 10 smaller datasets of the full training dataset. Training iteration on each subset consists of 100 epochs. We observe a rapid overfitting of the model closer to the end of each training iteration, however, the model’s performance improves over time. The model’s weights from the third epoch of the last training iteration are selected as the final model and used for evaluation on the testing set.

and the weights are saved after each training iteration. The weights are then loaded into the model for the next training iteration. This process is repeated until the model has been trained on the full dataset. Each training iteration consists of 100 epochs.

The training process is monitored using the loss curve and the accuracy metric shown in Fig. 6.5a and Fig. 6.5b.

From the accuracy curve, we can see that despite the rapid overfitting of the model closer to the end of each training iteration, the model’s performance improves over time. We choose the model’s weights from the third epoch of the last training iteration for the evaluation on the testing set. For this epoch the model’s accuracy on the validation and training sets are the closest, reaching a value of 0.89.

Now, the model is ready for evaluation on the testing set.

6.3 Results from the Classification Model

To evaluate the performance of the classification model on unseen data a separate dataset is generated for the testing, containing $7.4 \cdot 10^4$ entries for equal number of 1-electron and 2-electron events. The energy range and distribution of the events in the testing dataset are the same as for the training dataset. By convention, this dataset is called the *1e2e_test* dataset.

The model with the selected best parameters on the *1e2e_test* dataset achieved an accuracy of 0.90. The sigmoid function (see Section 4.2.2.2) is used to convert the model’s output to the probability of the 2-electron event shown in Fig.6.6a. The Signal distribution in red extends well into the low probability region, so that whatever threshold we opt for to cut on probability, we have a sizable set of false negatives. The background distribution in blue extends even more massively into the high probability region, so that in order to reject false positives this time, we have to opt for a high threshold for the cut on probability.

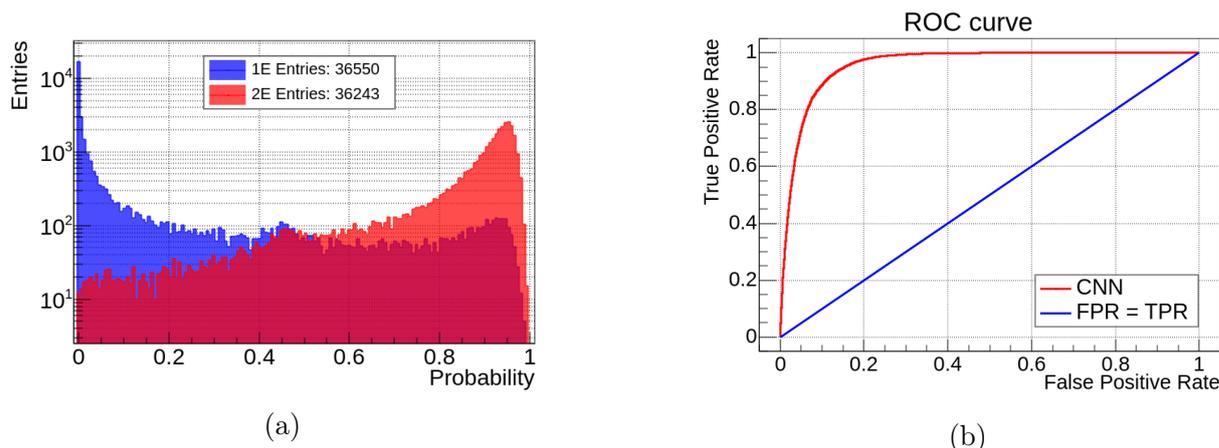


Figure 6.6: (a) Probability distributions for the 1-electron and 2-electron events from the Class3ResNet in the 1e2e_test dataset. We observe a high contamination of the False Positives classified as 2-electron events. (b) ROC curve for the classification model. The AUC value is 0.9572.

Another important metric to evaluate the model's performance is the Receiver Operating Characteristic (ROC) curve, which provides a visual representation of the trade-off between true positives and false positives (see Section 4.2.1). The ROC curve for the classification model is shown in Fig. 6.6b compared to the random classifier line. The AUC value for the model is 0.9572, indicating that the model performs reasonably well in distinguishing between the 1-electron and 2-electron events.

In addition we plot a Signal Efficiency and Background Rejection efficiency curves with respect to the threshold value. The Signal Efficiency is defined as the ratio of the number of TP to the total number of 2-electron events, while the Background Rejection Efficiency is defined as the ratio of the number of TN to the total number of 1-electron events. The Signal Efficiency and Background Rejection Efficiency curves are shown in Fig. 6.7a. The Signal Efficiency curve shows how well the model is able to identify the 2-electron events, while the Background Rejection Efficiency curve shows how well the model is able to reject the 1-electron events. The Signal Efficiency and Background Rejection Efficiency curves intersect at a threshold value of 0.6, for which we reduce the background contamination by a factor of ~ 10 , while keeping 89% of signal events. Selecting the Background rejection efficiency at 99% we can achieve the Signal Efficiency at 31.7% for the whole dataset. The Signal Efficiency vs Background Rejection Efficiency curve in Fig. 6.7b provides another view of the efficiency *vs.* purity trade-off. It is important to note that the optimal threshold value may vary depending on the specific requirements of the experiment. Our main interest is to maintain a high Signal Efficiency while reducing the background contamination by a factor of 100 in the ROI of the $0\nu\beta\beta$ signal events (defined in Section 2.4 as [2364, 2552] keV range). We select one and two electron events for which the primary energy is in this range and apply the model to classify them.

From the probability distributions of the 1-electron and 2-electron events in the ROI, we can observe a high contamination of FP events among the 1-electron events (Fig. 6.8a). The ROC curve for the events in the ROI gives similar results to the ROC curve for the whole dataset, with an AUC value of 0.9679 as one can see in Fig. 6.8b.

On the Signal Efficiency and Background Rejection Efficiency curves for the events in ROI, we can see that there is a slight degradation in the model's performance to discriminate 1-

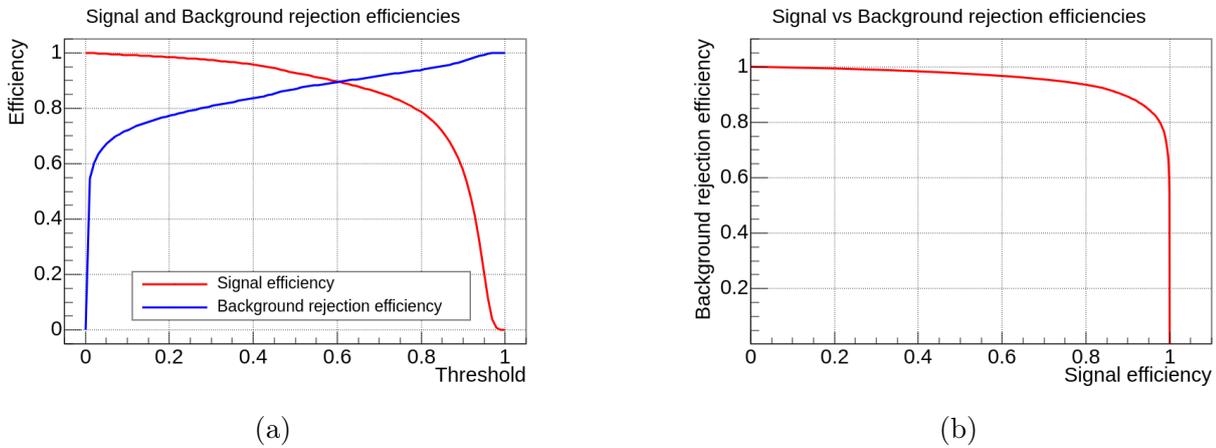


Figure 6.7: (a) Signal Efficiency and Background Rejection Efficiency curves for the Class3ResNet model as a function of the probability threshold value. (b) The Background Rejection Efficiency vs Signal Efficiency curve for Class3ResNet evaluation on the 1e2e_test dataset.

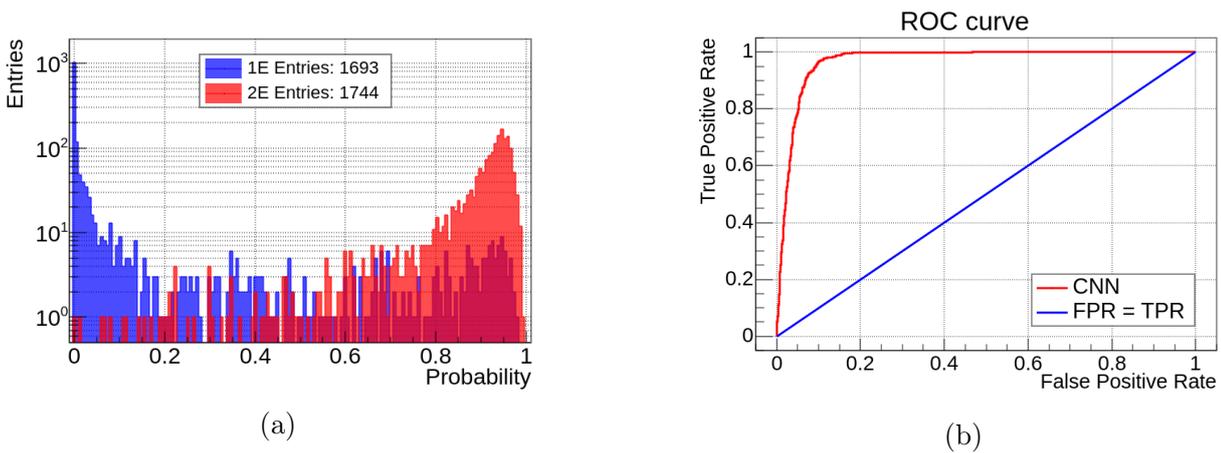


Figure 6.8: (a) Probability distributions for the 1-electron and 2-electron events from the Class3ResNet in the 1e2e_test dataset selected in the ROI. The high contamination of the False Positives classified as 2-electron events is still visible. (b) ROC curve for the classification model. The AUC value is 0.9679.

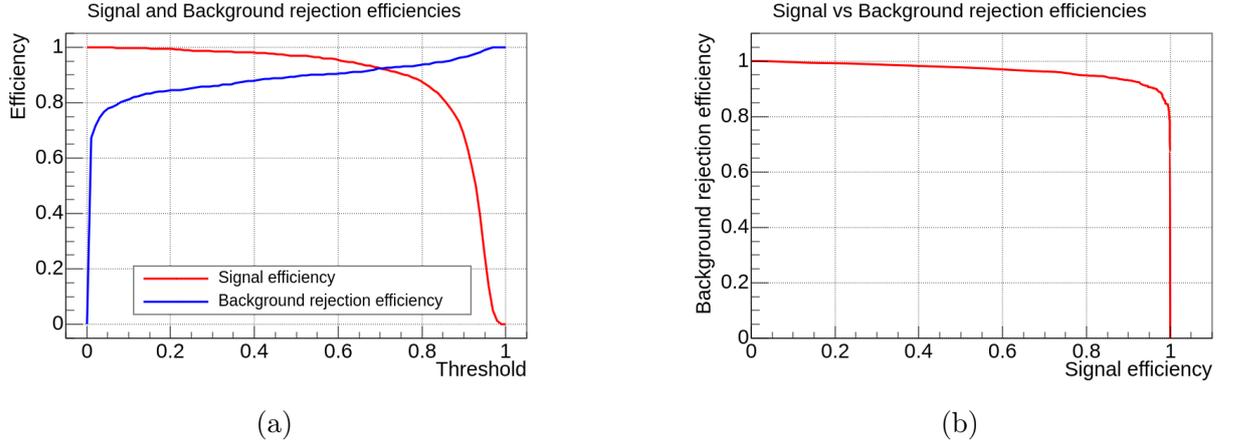


Figure 6.9: (a) Signal Efficiency and Background Rejection Efficiency curves for the Class3ResNet model as a function of the probability threshold value. For events selected in the ROI. (b) The Background Rejection Efficiency vs Signal Efficiency curve for Class3ResNet evaluation on the 1e2e_test dataset for events selected in the ROI.

electron and 2-electron events closer to the threshold value of 1 (Fig. 6.9a). A probability threshold corresponding to the background rejection efficiency of 99% is 0.95, corresponding to a signal selection efficiency of 26%. The Signal Efficiency vs Background Rejection Efficiency curve for the events in the ROI is shown in Fig. 6.9b.

The model's performance to discriminate 1-electron and 2-electron events in ROI is visualized in Fig. 6.10, where the distributions of before and after application of the model's rejection are shown for the primary energies of the 2-electron events (Fig. 6.10a) and 1-electron events (Fig. 6.10b). The plots below show the total sum of the degraded by missing channels signal amplitudes for the 2-electron and 1-electron events on Fig. 6.10c and Fig. 6.10d respectively. The sum of the degraded amplitudes correspond to the input images that are fed to the model, therefore the model is able to learn the topological features of the tracks even in the presence of missing channels.

An 80k dataset is generated for the $0\nu\beta\beta$ events to evaluate the model's Signal efficiency. Considering the same probability threshold of 0.95, the model's Signal Efficiency is reduced to 20% for the $0\nu\beta\beta$ events. The corresponding plots for the primary energy and the total sum of the degraded amplitudes distributions are shown in Fig. 6.11.

In order to understand such a behaviour of the model, we check the energy of the weakest electron, i.e. the one with the lowest energy, out of two electrons in 2-electron events dataset before and after the model's discrimination (Fig. 6.12a). For the events that contain one of the electrons with primary energy below 100 keV, the model is not able to select them as signals. This is expected as such events from the topological point of view are similar to the background events. The CNN model shows the highest Signal Efficiency of 33% for the 2-electron events with the lowest primary energy for one of the electrons at 600 keV. After that, we observe a decrease in efficiency up to $\sim 10\%$ for the 2-electron events which have the highest primary energy for one of the electrons at 2 MeV. This is the energy range where the two of the electrons have similar high primary energies therefore the topology of the whole event must possess prominent features of the 2-electron event, thus easily distinguishable from the background events. However, in the dataset, the presence of such events is dramatically low, which gave an availability bias to the model.

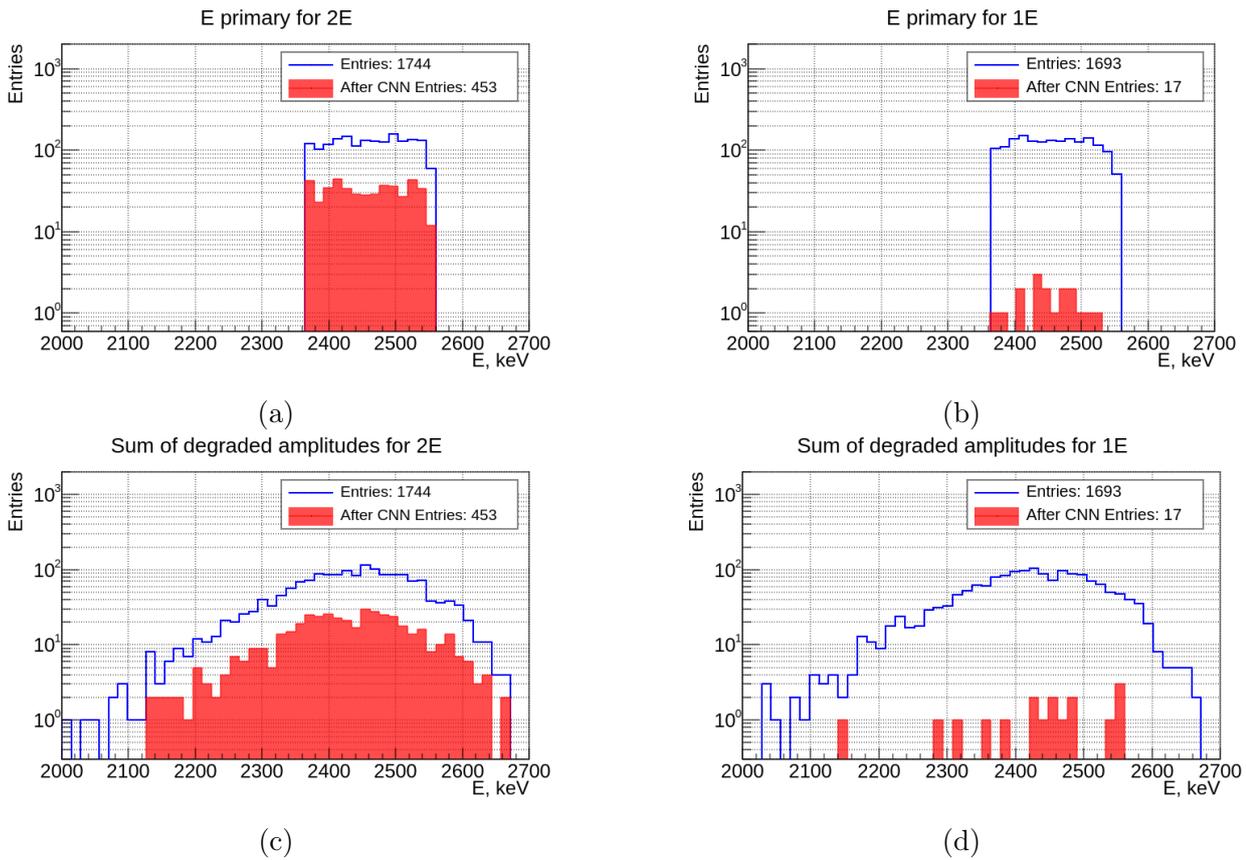


Figure 6.10: Energy distributions of the events selected in the ROI before (blue) and after (red) the model's rejection at the threshold value of 0.95 that corresponds to the Background Rejection Efficiency of 99% and Signal Efficiency of 26%. (a) Primary energy distribution for the 2-electron events. (b) Primary energy distribution for the 1-electron events. (c) Total sum of the degraded amplitudes for the 2-electron events. (d) Total sum of the degraded amplitudes for the 1-electron events.

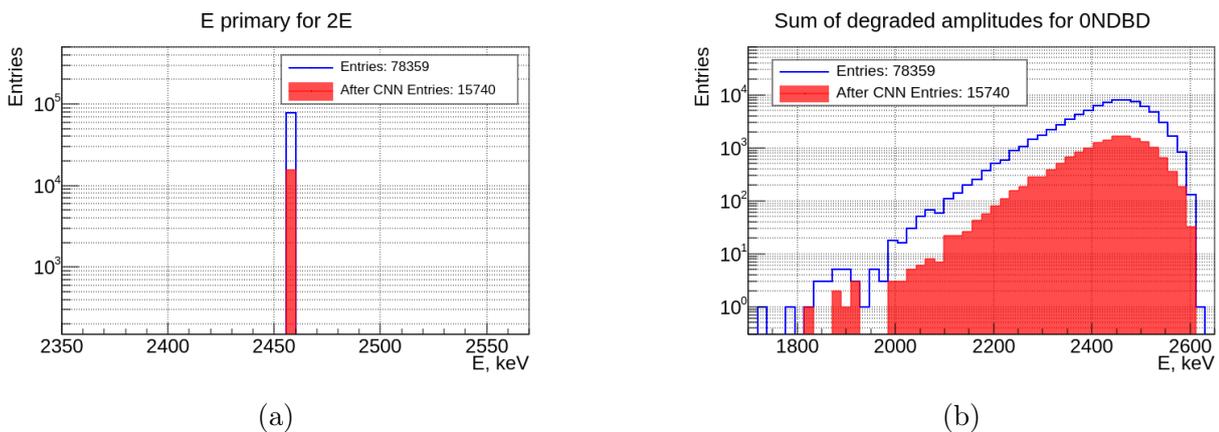


Figure 6.11: Energy distributions of the events selected in the ROI before (blue) and after (red) the model's rejection at the threshold value of 0.95 that corresponds to the Background Rejection Efficiency of 99% and Signal Efficiency of 20%. (a) Primary energy distribution for the $0\nu\beta\beta$ events. (b) Total sum of the degraded amplitudes for the $0\nu\beta\beta$ events.

The same analysis is performed for the $0\nu\beta\beta$ events dataset, which is shown in Fig. 6.12b. The $0\nu\beta\beta$ events show a similar behaviour: for the events with one of the electrons having the primary energy below 200 keV, the model doesn't recognize them as signals. Closer to higher primary energies of the electrons, the model's Signal Efficiency increases up to $\sim 25\%$ where it remains flat.

The Signal Efficiency and Background Rejection Efficiency are summarized in the Table 6.3.

Dataset	Signal Efficiency	Background Rejection Efficiency
1e2e_test	31.7%	99%
1e2e_test_ROI	26%	99%
$0\nu\beta\beta$	20%	99%

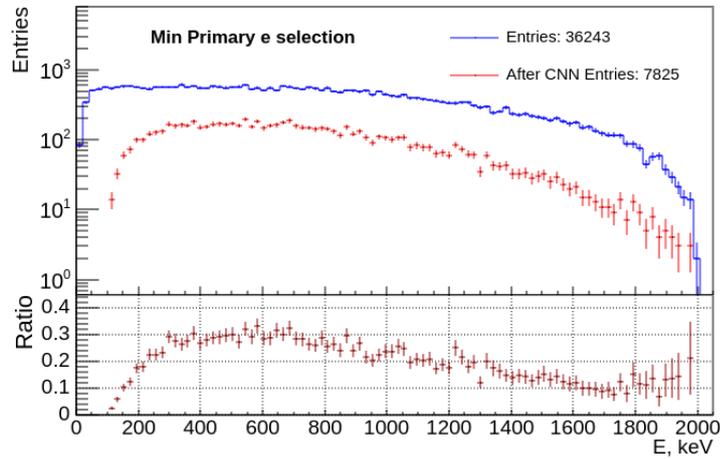
Table 6.3: Signal Efficiency and Background Rejection Efficiency for the 1e2e_test, 1e2e_test_ROI, and $0\nu\beta\beta$ datasets.

In 2018, Hao et al. [114] proposed a method to discriminate between the $0\nu\beta\beta$ events and the high-energy background events coming from the ^{238}U decay chain using the CNN model. The input data came from the postprocessing of the MC simulation of the PandaX-III detector. The readout plane is considered ideal without any missing channels. The final results have shown that the model is able to achieve a Signal Efficiency of 47.5% for the $0\nu\beta\beta$ events with a Background Rejection Efficiency of 99.4% which is shown in Fig. 6.13.

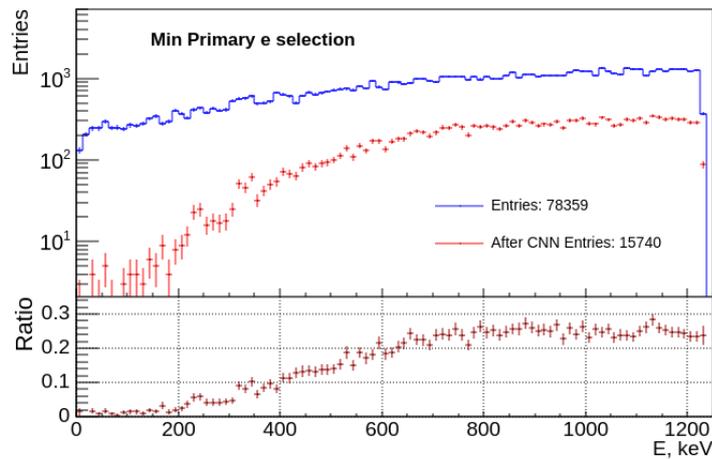
Indeed, the current model shows a lower Signal Efficiency for the $0\nu\beta\beta$ events, however, the model is trained and evaluated on the data with the missing channels, which is more realistic compared to the ideal case. The model's performance can be further improved by providing more uniform training data with the higher statistics.

6.4 Conclusions

In this chapter, we have presented a CNN model for the classification of the 1-electron and 2-electron events in the PandaX-III detector. The model is trained on the simulated data with the missing channels and is able to achieve an accuracy of 0.92 on the unseen data. The model's performance is evaluated using the ROC curve, Signal Efficiency, and Background Rejection Efficiency metrics. The model is able to distinguish between the 1-electron and 2-electron events with an AUC value of 0.9572. The Signal Efficiency and Background Rejection Efficiency curves showed that the model can achieve a Signal Efficiency of 31.7% for the 1e2e_test dataset with a Background Rejection Efficiency of 99%. For the events in the ROI, the model's performance is slightly degraded, with a Signal Efficiency of 26% and a Background Rejection Efficiency of 99%. For the $0\nu\beta\beta$ events, the model's Signal Efficiency is reduced to 20% with a Background Rejection Efficiency of 99%. The model's performance is compared to the results obtained by Hao et al. [114], who achieved a Signal Efficiency of 47.5% for the $0\nu\beta\beta$ events with a Background Rejection Efficiency of 99.4%. The current model shows a lower Signal Efficiency for the $0\nu\beta\beta$ events, however, the model is trained and evaluated on the data with the missing channels, which is more realistic compared to the ideal case. The model's performance can be further improved by providing more uniform



(a)



(b)

Figure 6.12: Plots represent primary energy of the weakest electron in the 2-electron events for the $1e2e_test$ dataset (a) and the $0\nu\beta\beta$ dataset (b). The blue line shows the primary energy before the model’s discrimination, while the red histogram shows what is left after the model’s rejection. Below each plot, the ratio of the accepted events with respect to the weakest electron’s primary energy is shown.

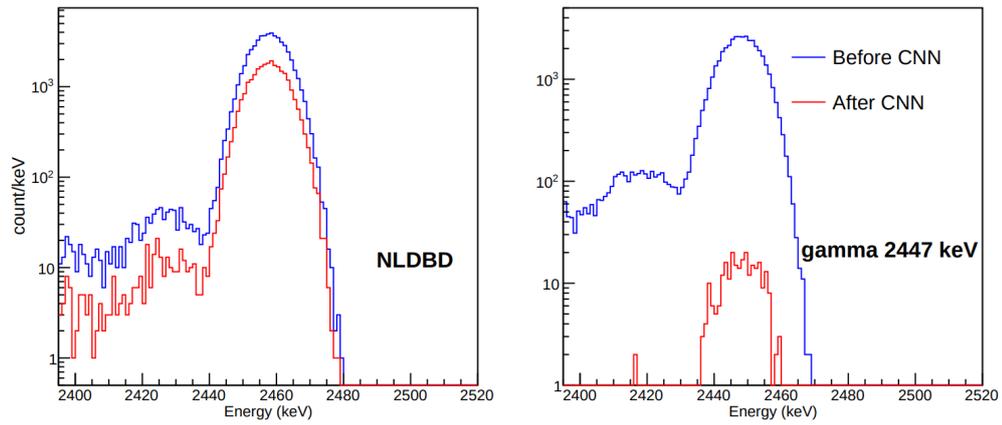


Figure 6.13: The reconstructed energy spectra of signal and backgrounds before and after the optimal cut for the $0\nu\beta\beta$ events(left) and background events from ^{214}Bi decay(right) from the work of Hao et al. [114].

training data with the higher statistics.

Chapter 7

Summary and Conclusions

This thesis presents $0\nu\beta\beta$ studies in the context of the PandaX-III experiment. The experiment's success hinges on achieving optimal performance from its Micromegas modules. Here, missing channels within these modules can significantly impact efficiency, leading to a loss of sensitivity to the $0\nu\beta\beta$ signal. Furthermore, track topology reconstruction can be compromised, hindering event identification and background rejection. In this work, results of the Blob charge determination from Monte Carlo simulations are presented, together with a Machine Learning approach, specifically employing Convolutional Neural Networks (CNNs), to rectify the energy loss induced by missing channels in the PandaX-III Micromegas detectors. Finally, an analysis of the CNN model to classify signal events from the background is presented.

Although there are no conclusive results on the Blob charge determination, each individual Blob charge contains 8-12% of the $0\nu\beta\beta$ track event from ^{136}Xe . Therefore, if the Blob charge is lost from the event reconstruction due to consecutively missing channels, the energy resolution of the event will be degraded. For now, the hypothesis that the energy lost due to missing channels in one projection (XZ) could be directly reconstructed from another projection (YZ) was not confirmed. Similarly, ideas around the correlation between twist and blob yielded different results. However, this initial exploration yielded valuable insights that paved the way for the successful application of ML methods in subsequent studies.

The results of the CNN model implementation demonstrated the remarkable potential for mitigating the effects of missing channels in the PandaX-III Micromegas detectors. The Multi3ResNet CNN model that was developed to predict the true energy of electrons from images of the tracks degraded by missing channels exhibits a remarkable increase from 69% to 90% in event prediction efficiency within the Region of Interest (ROI) for $0\nu\beta\beta$ events, compared to the conventional approach of directly summing amplitudes. This significant improvement underscores the power of CNNs in extracting complex information from intricate data patterns, ultimately leading to more accurate energy reconstructions. While these results are promising, further validation using real data from the PandaX-III experiment, such as calibration data, is crucial to confirm the model's performance under realistic conditions.

Another significant achievement is the CNN model's ability to classify two-electron events from single-electron events in Monte Carlo data affected by missing channels. The Class3ResNet CNN model, which architecture is based on the Multi3ResNet model, was

made to classify the two-electron (2e) events from the single-electron (1e) events. The 2e signature is crucial for identifying $0\nu\beta\beta$ decay, while 1e events are a significant background source. The Multi3ResNet model effectively distinguishes between these event types, achieving an accuracy of 92%. Key metrics further validate this model's performance: it has a Signal Efficiency of 31.7% and a Background Rejection Efficiency of 99% for the 1e+2e dataset. Even within the ROI, it maintains a Signal Efficiency of 26% for the same Background Rejection Efficiency. Although the model shows a lower Signal Efficiency for $0\nu\beta\beta$ events compared to previous studies by Hao et al. [114], it is important to note that this model is trained on data with missing channels, making it more representative of real-world conditions.

Future studies could incorporate events detected by multiple Micromegas modules, as in this work, a selection of events detected by a single module was used. This approach could improve the model's performance by providing more comprehensive data for training. A multi-module event selection would also allow for the study of the gaps between modules, which could also impact the energy resolution. Additionally, we need to consider the inhomogeneities in the readout gain across the detector readout plane. Future improvements could be achieved by using a more uniform training dataset with better statistical coverage.

These refinements hold the potential to push the boundaries of performance even further.

Appendix A

PandaX-III Data Analysis Software Framework—REST

A.1	Introduction	145
A.2	Application scenarios	146
A.2.1	Data processing	146
A.2.2	Data visualization	147
A.2.3	Simulation program: restG4	147
A.3	Development logic	148
A.3.1	Key classes	148
A.3.2	Data structures	149
A.3.3	Running process	150
A.3.4	Configuration file - RML	151
A.3.5	Main functions	152
A.4	Processing chain for PandaX-III experiment	153

A.1 Introduction

PandaX-III’s data analysis software relies on the Rare Event Search Toolkit (REST) framework [66]. Originally developed to meet the specific data processing requirements of the PandaX-III experiment, the software has undergone refinement in terms of stability, generality, and extensibility through extensive use and version updates. The REST repository is hosted on GitHub [67] and has been utilized in analyses for multiple experimental groups [73], [120], [121], [122].

REST is strategically designed to provide an integrated framework for low-background particle physics experiments, encompassing various software functionalities such as data reading, waveform analysis, response simulation, geometry processing, particle identification, statistical plotting, and more. It is developed as a secondary modification based on the widely used ROOT [123], [124] software framework in particle physics, integrating common simulation packages like Geant4 [125], Garfield++ [126], and database interfaces such as PostgreSQL and MySQL. Furthermore, REST itself offers electronic response simulation, commercial

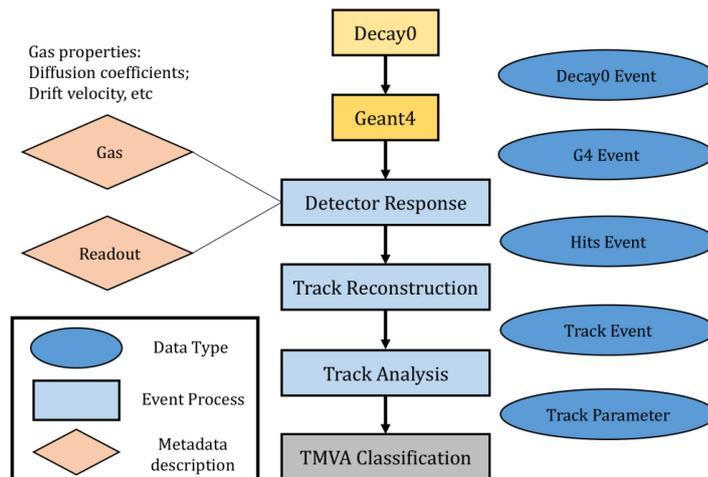


Figure A.1: The entire data processing workflow, from simulation to reconstruction, in the context of studying algorithms for neutrinoless double-beta decay, sourced from [66].

electronics unpacking tools like CoBo-AsAd, FEMINOS, USTC DCM, original waveform analysis, and particle trajectory reconstruction.

REST, employing configuration files, achieves the integration of the mentioned functionalities in a modular and procedural manner, representing the core of the framework. Each module (i.e., C++ class) implements specific functionalities and stores related configuration parameters, enhancing traceability for each run. The connectivity between modules is established through the pipeline of "events" serving as data carriers. In REST, all functionalities are regarded as the configuration process of the modules themselves or the processing of events. For instance, the functionality of database reading is executed during the initialization process of the run information module, while the electronic unpacking functionality is implemented in the event processing module for specific zero-input instances.

A.2 Application scenarios

A.2.1 Data processing

Data processing is the foundational functionality of REST, allowing for the effortless addition or removal of processing modules by reading the configuration file. Fig. A.5 illustrates the entire process, from simulation to reconstruction, in the context of studying algorithms for neutrinoless double-beta decay. The rectangular boxes represent various processing modules, all of which, with the exception of the last step, are executed within the REST framework. The elliptical boxes on the right denote the types of events circulating in the processing, while the diamond-shaped boxes on the left represent additional input metadata. If the steps for detector response were replaced with the actual steps for reading physical data, the subsequent processing methods would remain unchanged. In this manner, REST achieves seamless integration between simulated and experimental data.

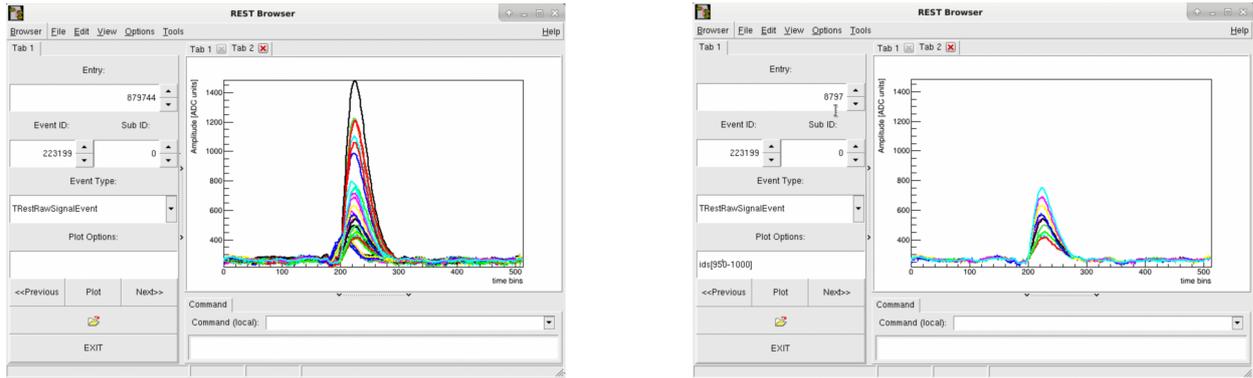


Figure A.2: REST browser interface. (right) Visualization of a specific instance in the waveform instance stage. (left) Visualization of the instance with added channel range plotting conditions. The images are from [66], where the data is from a run of the TREX [127] prototype detector.

A.2.2 Data visualization

REST offers three essential tools: the feature plotter (AnalysisPlot), metadata plotter (MetadataPlot), and event browser. The event browser, utilizing the graphical user interface (GUI) of ROOT, provides a drawing window with a control panel for visualizing individual events. Users can observe specific events by ID or storage index, encompassing all data types at each processing stage of the event. Furthermore, the event browser facilitates the input of specific plotting conditions and adjustments to the drawing range, as illustrated in Fig. A.2.

The feature plotter serves as a statistical tool for all events. By reading the corresponding configuration file, it can generate a histogram of a specific feature for all events in the file and supports input from multiple files. When plotting histograms from multiple files on the same canvas, it offers an intuitive comparison of differences between runs.

A.2.3 Simulation program: restG4

restG4 functions as a Monte Carlo simulation application within the Geant4 and REST framework, acting as the leader of the entire simulation processing pipeline. It incorporates the metadata concept, allowing for the direct interpretation of configuration files. This user-friendly approach streamlines the construction of Geant4 simulation tasks without the need for extensive coding. In restG4, geometry information is imported via GDML geometry files, while particle sources and physics lists are brought in through configuration files. The concept of sensitive volumes is introduced to record particle information within specified volumes. Additionally, restG4 provides visualization features for geometry models and particle trajectories, facilitating the examination of simulation issues. Fig. A.3 presents an example simulation.

The simulation outcomes generated by restG4 manifest as a series of events, capturing raw information about interactions between particles and the sensitive medium of the detector. This information encompasses position, time, deposited energy, particle type, parent particle type, interaction type, and more. Subsequently, these events are integrated into the ensuing data processing pipeline for further analysis.

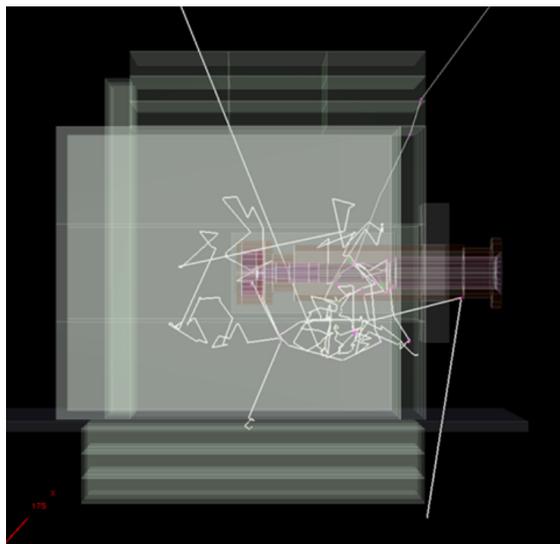
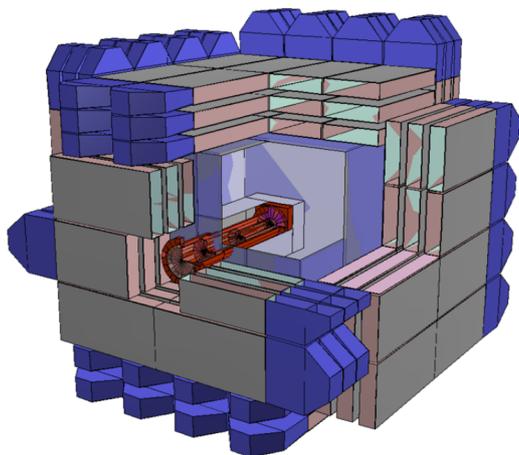


Figure A.3: Employing restG4 for simulating the initial energy deposition results in the detector. Left: Schematic representation of the Baby-IAXO detector's geometry [128]. Right: Depiction of energy deposition from a neutron event in the detector, with white lines illustrating a sequence of secondary particles. Image sourced from [66].

A.3 Development logic

REST is crafted using the C++ programming language and is constructed upon the ROOT software framework. The comprehensive software package encompasses various library files and executable files. The core library file, serving as the bedrock of the software, furnishes all the base classes of the software framework and orchestrates the calling logic between them. Other libraries' classes inherit from those in the core library, adapting their interfaces and delivering specific functionalities. External software packages like Geant4 are seamlessly incorporated into REST as functional libraries, offering users a range of choices. At the topmost level of the software, two executable files invoke the respective libraries to execute the software.

The majority of classes within REST inherit from the TObject class in ROOT, granting them direct compatibility for storage in files or retrieval from files. These classes adhere to the Taligent standard for naming conventions, commencing with "TRest." For the sake of simplicity, this text will omit the "TRest" keyword, representing "TRestEvent" as "Event."

A.3.1 Key classes

The following introduces three crucial classes within the core library, essential for later discussions on the specific logic of the software framework:

- **Event Class:** This class serves as the foundational type for storing diverse event data. It defines essential details about an event, such as timestamp, event ID, associated run ID, labels, etc. Subclasses derived from this class are tasked with storing specific event data. For instance, the RawSignalEvent class retains waveform data for each channel as member variables in corresponding arrays.



Figure A.4: Schematic diagram of the contents of a REST file. Items in red font are mandatory. Image is from [66].

- **Metadata Class:** The Metadata class is fundamental in REST for storing data beyond event data. It provides an interface for reading configuration files and outlines standard initialization processes. Various specific metadata is stored by subclasses derived from this class. Metadata classes, in conjunction with event classes, are stored in files, offering supplementary information about the detector, including readout module layout, gas information, etc. This aids users in comprehending the file contents better. Additionally, certain classes responsible for specific functions, such as managing and launching analysis scripts, possess configuration information worthy of preservation and are thus considered metadata, falling under the metadata class.
- **EventProcess Class:** The EventProcess class stands out as a distinctive type of metadata class. Subclasses derived from this class can execute data processing tasks. They accept input from specific event classes, execute designated data analysis algorithms alongside detector information from metadata classes, generate a series of features (observables) as analysis outcomes, and ultimately return the same or different event classes. Configuration information for analysis algorithms is preserved as metadata.

A.3.2 Data structures

REST files adhere to the ROOT file format with a standardized data organization. They consistently encompass a Run metadata class and an AnalysisTree (inspired by the ROOT TTree concept). The former captures details such as run ID, start and end times, and descriptive text, forming the file header. The latter systematically documents the characteristics of each instance, constituting the file content. These instances are stored in a separate tree, and if the conservation of disk space is a concern, there's an option to omit saving this segment of the data. Additionally, all metadata classes and data processing classes utilized during runtime are archived in the file. Finally, certain data processing classes generate supplementary analysis result images in fundamental ROOT formats like TH1D, TGraph, which are incorporated into the file. Fig. A.4 provides an illustration of the standardized data structure of REST.

Having the instance data alongside metadata is crucial for providing meaningful context to the instance data. For example, comprehending the significance of various points in a waveform necessitates information about the electronics' sampling rate, dynamic range,

and other specifics. REST files ensure this traceability. When saving physical results, it systematically and synchronously records various aspects of the detector's operation and program configuration as metadata. This renders the file a genuine physical record. When reading the file, it can appropriately extract this metadata, implying the logical notion of "reproducing a certain physical run." The encapsulated metadata classes are convenient for reuse. For users, the additional information in the file offers convenience for subsequent reference.

A.3.3 Running process

The REST framework offers a modular data processing pipeline known as the analysis chain. A sequence of data processing classes is connected end-to-end, facilitating the passage of instance data and structuring the pipeline. Users can customize this pipeline for diverse analysis purposes by configuring and adding/removing data processing classes in the chain through a configuration file. The input instances can be sourced from binary files acquired by the DAQ system or from REST files. The necessary metadata for processing is supplied by the Run metadata class.

The ProcessRunner class is responsible for managing and writing the output instances and features to files. Additionally, this class handles the initialization and invocation of the various data processing classes in the analysis chain. The completion of one analysis in the chain is represented by a single instance moving from input to output. Processing the entire dataset involves dealing with all instances in a run.

The analysis chain in REST follows a logically one-dimensional, one-way process. Each data processing class is designed to handle the input of one instance class and produce the output of another instance class. Subsequent data processing classes are unable to access data from earlier stages. The instance class at each stage provides comprehensive data tailored for the current stage's analysis. For instance, during the processing stage of the waveform instance class, analyses such as integration and saturation are conducted on sampled waveform information. In the hit point instance class (HitsEvent) stage, spatial clustering and drift diffusion analyses are performed on three-dimensional spatial coordinate information. Although the two instance classes correspond to the same input instance with identical instance ID information, they possess different data storage structures. Specific data processing classes can be utilized to transform between instance classes, thereby connecting analyses from distinct stages.

Analyses conducted by data processing classes, such as waveform integration, yield features referred to as characteristics. Unlike instance data, features generated at different points in the analysis chain are not passed along; instead, they accumulate. Once all data processing classes complete their tasks, the features and output instances are saved. The features are documented in a dedicated feature tree in REST, while the output instances are recorded in another tree. The feature tree, a ROOT TTree subclass, encompasses various branch variables. Users can add a value directly as a branch without the need to manually set the branch addresses, allowing for simplicity of use.

During the actual data processing run, REST employs a multi-threading engine to enhance processing speed. The main thread clones the analysis chain into multiple copies, distributing them among an equal number of subthreads. It then assigns the input instances to the subthreads for processing and subsequently gathers the processing results, including output

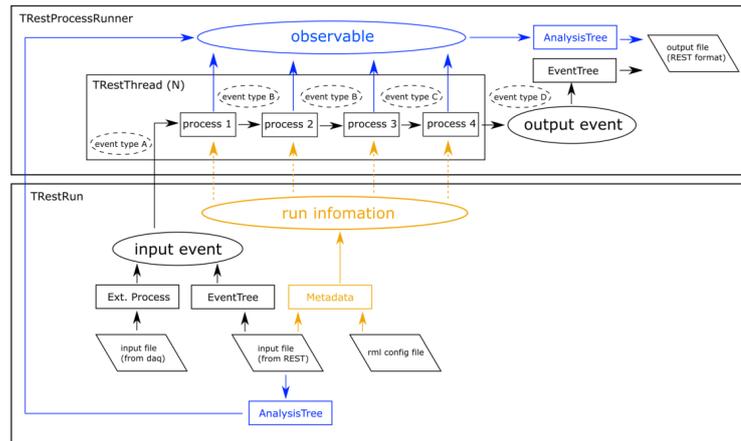


Figure A.5: A schematic representation illustrating the flow of event data within a REST processing chain. The run object is initialized, providing access to specific metadata or event data present in the input REST file, along with any additional objects outlined through RML. Subsequently, the data undergoes processing via the implementation within the process runner object. Various event types (A, B, C, D) refer to distinct implementations for specific events. The resulting output REST file consolidates all metadata information accessible to the chain, including any pre-existing data, featuring the transformed output for the specific event and the updated analysis tree. Image is from [66].

instances and corresponding features, writing them back to the file in a sequential manner. This approach achieves multi-threading. The complete data processing workflow is illustrated in Fig. A.5.

REST establishes a connection between detector data and simulated data by standardizing the instance processing flow. Detector data is encapsulated by the corresponding instance class and undergoes processes like waveform compression, position reconstruction, clustering, and particle trajectory recognition. Simultaneously, Monte Carlo-generated data, encapsulated into the same instance class using an appropriate electronic simulation program, can undergo the same physical analysis processes. This advantage facilitates the development of diverse algorithms, their debugging using simulated data, and direct application to the physical data analysis of the detector.

A.3.4 Configuration file - RML

REST's configuration files utilize the RML extension and are structured based on the extensible markup language (XML). It extends XML concepts and introduces several keywords, offering additional parsing capabilities for units and expressions, along with functionality for variable substitution. The newly introduced keywords by RML take effect during the file parsing process:

- **if:** An **if**-tagged XML element unveils its internal child elements when the specified condition is met.
- **for:** A **for**-tagged XML element iterates, revealing its internal child elements once per iteration.

- **parameter:** A **parameter**-tagged XML element is equivalent to the attribute of its parent element.
- **variable:** A **variable**-tagged XML element can substitute the corresponding XML keyword marked with **\$** and may be influenced by system environment variables.
- **constant:** A **constant**-tagged XML element can replace the corresponding XML keyword and is not affected by system environment variables.
- **globals:** A **globals**-tagged XML element reveals its internal child elements to all other elements containing child elements.

REST's architectural design establishes a direct linkage between XML elements within configuration files and metadata classes within the program. Each XML element in the file corresponds to a specific metadata class in the program, mirroring the nesting relationships of XML elements to the calling relationships among various classes in the program. For instance, the runner class takes on the responsibility of initializing and invoking diverse data processing classes within the analysis chain. In the RML format, XML elements representing various data processing classes are encapsulated within the XML element of the runner class, thus becoming its child elements. This strategic design enhances the logical flow and extensibility of the configuration file.

To streamline the initialization process, REST introduces a naming convention that aligns configuration file parameters with member variables of metadata classes, forging a direct connection between their names. Each parameter in the configuration file corresponds to a member variable in the metadata class. Capitalizing on this convention and leveraging reflective capabilities inherited from the base class, metadata classes can autonomously locate parameters corresponding to their member variables in the configuration file during initialization. This approach eliminates the need for manual coding, effectively reducing the user's workload.

In handling numerical parameters, REST adopts a standardized unit system. The configuration file parser adeptly distinguishes between the numerical and unit components of parameters. If the unit part is expressed in non-standard units, the numerical aspect is scaled to a value in standard units. For instance, if a time is specified as 5 ns in the configuration file and REST's standard unit for time is μs , the value is automatically converted to 0.005. Unlike general programming practices where variables are typically pure numbers, requiring explicit declaration of units, REST's automatic initialization and standard unit system assume default standard units for all variables. This preemptively avoids confusion arising from inconsistent units across different fields, contributing to a more seamless and error-resistant user experience.

A.3.5 Main functions

REST encompasses two distinct primary functionalities, each encapsulated within a separate executable:

- **restManager:** This executable is responsible for parsing the configuration file and executing tasks specified in its content, with a primary focus on data processing. The flexibility inherent in the configuration file allows a singular restManager instance to

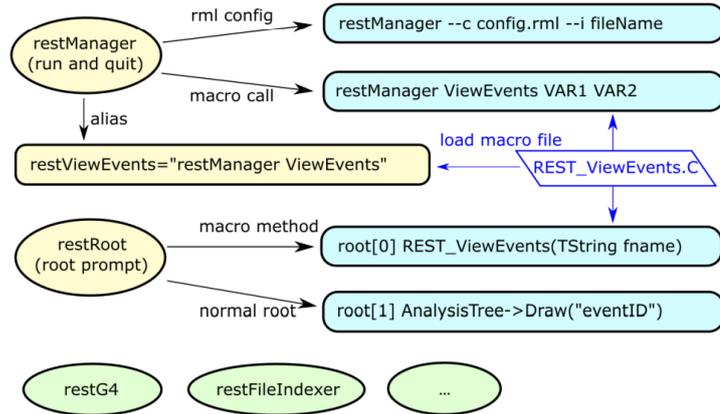


Figure A.6: REST program execution framework. The two primary executables, `restManager` and `restRoot`, are highlighted in red. The `rest-config` shell script is highlighted in blue. Image is from [66].

undertake a diverse array of tasks, ranging from data processing and event browsing to plotting and saving.

- **restRoot:** In contrast, this executable is designed to open data files, load file meta-data, and provide users with a ROOT-based command-line environment. It offers a platform for manual analysis projects, empowering users with a direct and interactive interface for their analytical endeavors.

To streamline invocation procedures, 'restManager' incorporates dedicated task aliases for common operations. Essentially, these aliases function as additional executable identifiers, though the underlying executable remains 'restManager'. As an illustration, the alias "restViewEvents" is equivalent to the command "restManagerViewEvents." Moreover, REST includes a shell script called 'rest-config', which, when supplied with specific input parameters, furnishes users with essential information about the REST program. This information encompasses details such as the installation location, the list of library files, the version, and more, facilitating user scrutiny. Fig. A.6 provides an overview of the comprehensive program execution framework.

A.4 Processing chain for PandaX-III experiment

The main construction blocks and principles of the REST framework have been explained up to this point. This section will delve into the specifics of the processing chain for the PandaX-III experiment employed in this thesis's data simulation.

The Monte Carlo simulation program `restG4`, introduced in Section A.2.3, is used to generate the initial energy deposition results in the detector's gas medium. The simulation is based on the first PandaX-III module setup, described in Chapter 2, stored in the Geometry Description Markup Language (GDML) format. The volume of the detector is filled with 140 kg of 90% enriched ^{136}Xe gas with 1% TMA. We are setting the particle production from the gas volume with isotropic angular distribution. For the 1-electron and 2-electron studies, the initial kinetic energies are set with appropriate ranges. In contrast, for the $0\nu\beta\beta$ study,

the kinetic energies for both electrons are retrieved from the kinematics produced with the Decay0 event generator [70, 71] integrated into the restG4.

The modular design of REST ensures that, throughout the manipulation process, data are encoded within a specific structure, named with the prefix `TRest` and the suffix `Event`. Simulation chains are constructed by employing processes that operate on these structures and others that transform one structure into another. This approach allows creating complex analysis chains from basic building blocks, each defined by its function and the data structure it handles. In the processing chain that simulated the detector response in the form of signals, the following structures are used:

- **`TRestDetectorHitsEvent`:** This structure is used to define an arbitrary number of energy deposits, commonly called *hits*, defined by their position in space (x, y, z) , their energy E , and their time coordinate t . It is important to note that this structure is naturally recursive because it is possible to have a `TRestDetectorHitsEvent` that contains a set of hits, which are themselves sets of hits. This feature allows for grouping hits that are close to each other.
- **`TRestGeant4Event`:** This structure is used as an extension of `TRestHitsEvent`. It adds additional information related to the Geant4 simulations that generated these hits, such as the physical process that caused the hit.
- **`TRestDetectorSignalEvent`:** This structure contains the time signal of each read-out channel of the detector plane. It is an extension of `TRestRawSignalEvent` which is a container for the time raw signal data with fixed length (e.g., the number of time samples of the DAQ system).

The output of the restG4 simulation produces `TRestGeant4Event` objects that encapsulate the energy deposition of a charged particle propagation inside the gas volume. It is passed through the processing chain to simulate the detector's response to the energy depositions to produce the signal output:

- **`TRestGeant4ToDetectorHitsProcess`:** This process is responsible for converting the `TRestGeant4Event` objects into `TRestDetectorHitsEvent` objects. It leaves only the information about the energy deposition hits in the gas volume, discarding the information about the physical process that caused the hit.
- **`TRestDetectorElectronDiffusionProcess`:** This process simulates the diffusion of electrons in the chamber, causing a temporal and spatial broadening of the signal based on the z coordinate of the signal. Here, the response of the Xenon+TMA (1%) mixture at 10 bars with a drift field of 1 kV/cm is applied.
- **`TRestDetectorHitsSmearingProcess`:** This process simulates the smearing of the signal due to the finite resolution of the detector. It allows the energy resolution of each event to be adjusted directly. For this study, the detector response is calibrated to 3% FWHM at an energy of 2457.83 keV.
- **`TRestDetectorHitsToSignalProcess`:** This process converts hits into signals by projecting these hits onto the detection plane. Each hit is assigned a readout channel using its (x, y) position by applying the channel positions contained in `TRestReadout`

metadata. The *TRestReadout* metadata contains the positions of the Micromegas modules (and corresponding channels) in the readout plane. In the PandaX-III context, the readout plane is configured with 52 Micromegas modules, each with 128 channels, described in Section 2.3.3. The z coordinate is used in conjunction with the hit's energy to create the signal. The properties of the gas where the electron drifts are utilized through the *TRestGas* class, which interfaces REST with Garfield++ [69]. Garfield++ provides the properties of the gas mixture based on calculations performed with Magboltz.

- ***TRestDetectorSignalToRawSignalProcess***: This process transforms a *TRestDetectorSignalEvent* into a *TRestRawSignalEvent*. The *TRestDetectorSignalEvent* contains signal data built with arbitrary times and their corresponding values (time, data). The data inside a *TRestRawSignal* loses precision on the time definition, and it is just a data array with a fixed number of data bins. The number of bins and the bin width depend on the DAQ system used in the experiment. In PandaX-III, the AGET system is used, which has a 512 time sample window and for this study, a 200 ns time bin width is used. In this thesis project, the first-time deposit found in the event corresponds to the bin 0 of the raw signal. An additional trigger delay shift of 50 bins is added to the raw signal to emulate the trigger delay of the AGET system.
- ***TRestRawSignalAddFullNoiseProcess***: This process is an extension of the *TRestRawSignalAddNoiseProcess* that adds white noise to the raw signal in the *TRestRawSignalEvent* and is present in the default REST framework. The noise is generated based on the noise model of the AGET system, which includes the noise of the preamplifier, the noise of the amplifier, and the noise of the ADC. The noise is added to the signal in the time domain. *TRestRawSignalAddFullNoiseProcess* extends this process by adding the noise to all the channels of the corresponding Micromegas detectors whose channels registered a signal.
- ***TRestRawSignalShapingProcess***: This process allows the convolution of the raw signal with the shaping function. In this project, we use the Gaussian shaping function with a $1 \mu\text{s}$ peaking time that corresponds to the AGET system.

The output of the processing chain is a *TRestRawSignalEvent* object that contains the signal data for each channel of the readout plane. The signal data is stored as a time series with a fixed number of time samples. The signal data is then stored in a ROOT file for further analysis.

A process called *TRestRawSignalDataCollectionProcess* was created specifically for the raw signal data collection that is used in the Machine Learning analysis present in Chapter 5 and 6. This process does not modify the raw signal data, but it collects the signal amplitudes corresponding to the time samples from each channel of the readout plane and stores them in a 2D array. The signals from the X and Y planes are stored in separate arrays. These arrays are then concatenated into a single array used as an input for the Machine Learning algorithms. The *TRestRawSignalDataCollectionProcess* process stores the *TRestRawSignalEvent* object in the ROOT file so that the raw signal data can be used by REST later on, and creates a separate .csv file with the Machine Learning input data.

The *TRestRawSignalEvent* object is then can be used to reconstruct XZ and YZ projections of the initial electron track in the detector within the REST framework. This is done by

applying the track reconstruction pipeline involving the following processes: *TRestRawZeroSuppressionProcess*, *TRestDetectorSignalToHitsProcess*, *TRestDetectorHitsToTrackFastProcess*, *TRestTrackReductionProcess*, *TRestTrackPathMinimizationProcess*, *TRestTrackReconnectionProcess*. These processes are not described in details, as the raw signal simulated data is of the main interest in this thesis project. However, the *TRestTrackBlobAnalysisProcess* is used at the final stage of the processing chain to perform the blob charge analysis on the reconstructed tracks in Chapter 3.

After the *TRestTrackReconnectionProcess* has naturally identified the track endpoints, this process records the coordinates or positions of these endpoints in the analysis tree. This process searches for the region with the highest density within the last 20% of the track length, which is then used as the blob position. The blob charge is calculated by summing the charge within a radius R around the blob position. The blob charge is then stored in the analysis tree.

Appendix B

Backpropagation algorithm on a simple neural network

The DNNs are usually trained using the *backpropagation algorithm* [83]. The backpropagation algorithm, or backward propagation of errors, is an algorithm for supervised learning of the neural networks that uses *Gradient Descent*. It is an optimization algorithm used to minimize the function. In ML for multi-layered feedforward networks it uses the chain rule to iteratively compute gradients of the cost function $C(\omega)$ with respect to the NN's weights and biases ω , as gradients indicate the direction of the steepest ascent of the cost function. At each iteration of gradient descent the weights and biases are being updated, according to the *learning rate* α :

$$\omega^{(t)} = \omega^{(t-1)} - \alpha \frac{\partial C(\omega^{t-1})}{\partial \omega}, \quad (\text{B.1})$$

where $\omega^{(t)}$ is the set of weights and biases at the iteration t and learning rate α is a hyperparameter that controls the step size of the gradient descent.

To understand the backpropagation algorithm, let's look at the model with two hidden layers with the activation functions $\mathcal{A}^{(l)}$, where l is the hidden layer index (Fig. B.1). The

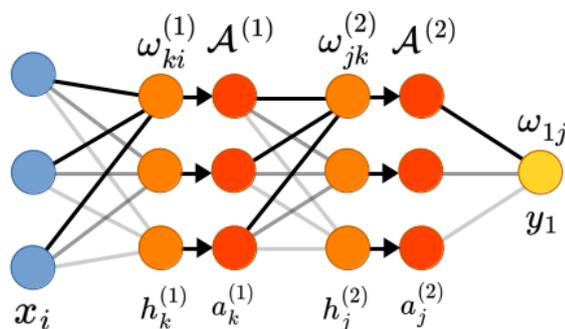


Figure B.1: Linear model with two hidden layers.

input layer denoted as vector x_i of a size of $n \times 1$, where n is the number of nodes in the input layer, which equals three. The inputs could be scalars, vectors, or even tensors. We

can write a generalized expression for the output of a hidden layer l as:

$$h_k^{(l)} = \sum_i \omega_{ki}^{(l)} a_i^{(l-1)} + \omega_{k0}^{(l)}, \quad (\text{B.2})$$

where $a_i^{(l-1)}$ is the output of the previous layer (for the first hidden layer $l = 1$ the $a_i^{(l-1)}$ is just an input layer x_i), $\omega_{ki}^{(l)}$ are the weights matrix of the hidden layer l , having size of $m \times n$, where m is the number of nodes in the hidden layer l , which in this case equals three, and n is the number of nodes in the previous layer, which is also three, and $\omega_{k0}^{(l)}$ is the bias vector of the hidden layer l , having size of $m \times 1$.

The output of the layer l with the activation function $\mathcal{A}^{(l)}$ is simply:

$$a_k^{(l)} = \mathcal{A}^{(l)} h_k^{(l)} = \mathcal{A}^{(l)} \left(\sum_i \omega_{ki}^{(l)} a_i^{(l-1)} + \omega_{k0}^{(l)} \right). \quad (\text{B.3})$$

And finally the output of the network is:

$$y_s \Big|_{s=1} = \left(\sum_k \omega_{sk} a_k^{(l)} + \omega_{s0} \right) \Big|_{s=1, l=2}, \quad (\text{B.4})$$

which is exactly the same as the Eq. B.2, if we considered the output layer as the hidden layer $l = 3$, so we can convert the Eq. B.4 in order to generalize the further calculations:

$$y_s \Big|_{s=1} = h_s^{(l)} \Big|_{s=1, l=3} = \left(\sum_k \omega_{sk} a_k^{(l-1)} + \omega_{s0} \right) \Big|_{s=1, l=3}. \quad (\text{B.5})$$

These equations above form the **forward propagation** of the neural network. After the forward propagation is performed, the output of the network y_s is compared to the target value y and the error is calculated with the cost function $C(\omega) = C(y, y_s(\omega))$. The gradient of a cost function with respect to weights $\omega_{ij}^{(l)}$ for the layer l :

$$\frac{\partial C}{\partial \omega_{ij}^{(l)}} = \frac{\partial C}{\partial h_j^{(l)}} \frac{\partial h_j^{(l)}}{\partial \omega_{ij}^{(l)}}, \quad (\text{B.6})$$

and taking into account the definition of the $h_j^{(l)}$ from the Eq. B.2:

$$\frac{\partial C}{\partial \omega_{ij}^{(l)}} = \frac{\partial C}{\partial h_j^{(l)}} a_i^{(l-1)}. \quad (\text{B.7})$$

Similarly, the gradient of a cost function with respect to biases $\omega_{0j}^{(l)}$:

$$\frac{\partial C}{\partial \omega_{0j}^{(l)}} = \frac{\partial C}{\partial h_j^{(l)}} \frac{\partial h_j^{(l)}}{\partial \omega_{0j}^{(l)}} = \frac{\partial C}{\partial h_j^{(l)}}. \quad (\text{B.8})$$

The common first term for Eq. B.7 and Eq. B.8 is the local gradient $\delta_j^{(l)}$, which is also called the **error term**:

$$\delta_j^{(l)} = \frac{\partial C}{\partial h_j^{(l)}}, \quad (\text{B.9})$$

that can be calculated knowing the cost function C itself.

Now, we can write the gradient of a cost function with respect to weights $\omega_{ij}^{(l)}$ and biases $\omega_{0j}^{(l)}$ as:

$$\frac{\partial C}{\partial \omega_{ij}^{(l)}} = \delta_j^{(l)} a_i^{(l-1)}, \quad (\text{B.10})$$

$$\frac{\partial C}{\partial \omega_{0j}^{(l)}} = \delta_j^{(l)}. \quad (\text{B.11})$$

Therefore, remembering the Eq. B.1, for the iteration $t + 1$ of the gradient descent, the weights and biases are updated as:

$$\omega_{ij}^{(l)} = \omega_{ij}^{(l)} - \alpha \delta_j^{(l)} a_i^{(l-1)}, \quad (\text{B.12})$$

$$\omega_{0j}^{(l)} = \omega_{0j}^{(l)} - \alpha \delta_j^{(l)}. \quad (\text{B.13})$$

Now, we can write the error term $\delta_j^{(l)}$ for the hidden layer l using the chain rule:

$$\delta_j^{(l)} = \frac{\partial C}{\partial h_j^{(l)}} = \sum_k \frac{\partial C}{\partial h_k^{(l+1)}} \frac{\partial h_k^{(l+1)}}{\partial h_j^{(l)}} = \sum_k \delta_k^{(l+1)} \frac{\partial h_k^{(l+1)}}{\partial h_j^{(l)}}. \quad (\text{B.14})$$

The partial derivative $\frac{\partial h_k^{(l+1)}}{\partial h_j^{(l)}}$ can be calculated using the Eq. B.2:

$$\frac{\partial h_k^{(l+1)}}{\partial h_j^{(l)}} = \frac{\partial}{\partial h_j^{(l)}} \left(\sum_i \omega_{ki}^{(l+1)} a_i^{(l)} + \omega_{k0}^{(l+1)} \right) = \omega_{kj}^{(l+1)} \frac{\partial a_j^{(l)}}{\partial h_j^{(l)}}. \quad (\text{B.15})$$

The partial derivative $\frac{\partial a_j^{(l)}}{\partial h_j^{(l)}}$ can be calculated using the Eq. B.3:

$$\frac{\partial a_j^{(l)}}{\partial h_j^{(l)}} = \frac{\partial}{\partial h_j^{(l)}} \mathcal{A}^{(l)} \left(\sum_i \omega_{ji}^{(l)} a_i^{(l-1)} + \omega_{j0}^{(l)} \right) = \mathcal{A}^{(l)'} \left(\sum_i \omega_{ji}^{(l)} a_i^{(l-1)} + \omega_{j0}^{(l)} \right). \quad (\text{B.16})$$

The partial derivative $\mathcal{A}^{(l)'}$ is the derivative of the activation function $\mathcal{A}^{(l)}$ and is called the **activation function derivative**. For the ReLU activation function $\mathcal{A}^{(l)}(x) = \max(0, x)$, the derivative is $\mathcal{A}^{(l)'}(x) = 1$ for $x > 0$ and $\mathcal{A}^{(l)'}(x) = 0$ for $x < 0$. For the sigmoid activation function $\mathcal{A}^{(l)}(x) = (1 + e^{-x})^{-1}$, the derivative is $\mathcal{A}^{(l)'}(x) = \mathcal{A}^{(l)}(x)(1 - \mathcal{A}^{(l)}(x))$.

Now, we can write the error term $\delta_j^{(l)}$ for the hidden layer l using the Eq. B.15 and Eq. B.16:

$$\delta_j^{(l)} = \sum_k \delta_k^{(l+1)} \omega_{kj}^{(l+1)} \mathcal{A}^{(l)'} \left(\sum_i \omega_{ji}^{(l)} a_i^{(l-1)} + \omega_{j0}^{(l)} \right). \quad (\text{B.17})$$

The Eq. B.17 is the **backpropagation equation**, which is used to calculate the error term $\delta_j^{(l)}$ for the hidden layer l using the error term $\delta_k^{(l+1)}$ for the hidden layer $l + 1$.

Bibliography

- [1] Matteo Agostini et al. “Toward the discovery of matter creation with neutrinoless $\beta\beta$ decay”. In: *Rev. Mod. Phys.* 95.2 (2023), p. 025002. DOI: [10.1103/RevModPhys.95.025002](https://doi.org/10.1103/RevModPhys.95.025002). arXiv: [2202.01787](https://arxiv.org/abs/2202.01787) [[hep-ex](#)] (cit. on pp. 28, 35, 36).
- [2] Enrico Fermi. “Tentativo di una teoria dell’emissione dei raggi beta”. In: *Ric. Sci.* 4 (1933), pp. 491–495 (cit. on p. 28).
- [3] Ettore Majorana. “Teoria simmetrica dell’elettrone e del positrone”. In: *Nuovo Cim.* 14 (1937), pp. 171–184. DOI: [10.1007/BF02961314](https://doi.org/10.1007/BF02961314) (cit. on p. 28).
- [4] W. H. Furry. “On transition probabilities in double beta-disintegration”. In: *Phys. Rev.* 56 (1939), pp. 1184–1193. DOI: [10.1103/PhysRev.56.1184](https://doi.org/10.1103/PhysRev.56.1184) (cit. on p. 28).
- [5] Carl D. Anderson and Seth H. Neddermeyer. “Cloud Chamber Observations of Cosmic Rays at 4300 Meters Elevation and Near Sea-Level”. In: *Phys. Rev.* 50 (1936), pp. 263–271. DOI: [10.1103/PhysRev.50.263](https://doi.org/10.1103/PhysRev.50.263) (cit. on p. 28).
- [6] Raymond Davis Jr. “Attempt to detect the antineutrinos from a nuclear reactor by the $\text{Cl}37(\text{anti-}\nu, e^-) \text{A}37$ reaction”. In: *Phys. Rev.* 97 (1955), pp. 766–769. DOI: [10.1103/PhysRev.97.766](https://doi.org/10.1103/PhysRev.97.766) (cit. on p. 28).
- [7] C. L. Cowan et al. “Detection of the free neutrino: A Confirmation”. In: *Science* 124 (1956), pp. 103–104. DOI: [10.1126/science.124.3212.103](https://doi.org/10.1126/science.124.3212.103) (cit. on p. 29).
- [8] F. Reines et al. “Detection of the free anti-neutrino”. In: *Phys. Rev.* 117 (1960), pp. 159–173. DOI: [10.1103/PhysRev.117.159](https://doi.org/10.1103/PhysRev.117.159) (cit. on p. 29).
- [9] B. Pontecorvo. “Inverse beta processes and nonconservation of lepton charge”. In: *Zh. Eksp. Teor. Fiz.* 34 (1957), p. 247 (cit. on p. 29).
- [10] Z. Maki, M. Nakagawa, and S. Sakata. “Remarks on the unified model of elementary particles”. In: *Prog. Theor. Phys.* 28 (1962), pp. 870–880 (cit. on pp. 29, 30).
- [11] Raymond Davis Jr., Don S. Harmer, and Kenneth C. Hoffman. “Search for neutrinos from the sun”. In: *Phys. Rev. Lett.* 20 (1968), pp. 1205–1209. DOI: [10.1103/PhysRevLett.20.1205](https://doi.org/10.1103/PhysRevLett.20.1205) (cit. on p. 29).
- [12] K. Hirata et al. “Observation of a Neutrino Burst from the Supernova SN 1987a”. In: *Phys. Rev. Lett.* 58 (1987). Ed. by K. C. Wali, pp. 1490–1493. DOI: [10.1103/PhysRevLett.58.1490](https://doi.org/10.1103/PhysRevLett.58.1490) (cit. on p. 30).
- [13] P. Anselmann et al. “First results from the Cr-51 neutrino source experiment with the GALLEX detector”. In: *Phys. Lett. B* 342 (1995), pp. 440–450. DOI: [10.1016/0370-2693\(94\)01586-2](https://doi.org/10.1016/0370-2693(94)01586-2) (cit. on p. 30).

- [14] Christopher W. Walter. “The Super-Kamiokande Experiment”. In: (Feb. 2008), pp. 19–43. DOI: [10.1142/9789812771971_0002](https://doi.org/10.1142/9789812771971_0002). arXiv: [0802.1041](https://arxiv.org/abs/0802.1041) [[hep-ex](#)] (cit. on p. 30).
- [15] J. Boger et al. “The Sudbury Neutrino Observatory”. In: *Nucl. Instrum. Meth. A* 449 (2000), pp. 172–207. DOI: [10.1016/S0168-9002\(99\)01469-2](https://doi.org/10.1016/S0168-9002(99)01469-2). arXiv: [9910016](https://arxiv.org/abs/9910016) [[nucl-ex](#)] (cit. on p. 30).
- [16] Nobel Prize Outreach AB. *The Nobel Prize in Physics 2015*. NobelPrize.org. Nobel Prize Outreach AB. URL: <https://www.nobelprize.org/prizes/physics/2015/summary/> (cit. on p. 30).
- [17] B. Pontecorvo. “Mesonium and anti-mesonium”. In: *Sov. Phys. JETP* 6 (1957), p. 429 (cit. on p. 30).
- [18] K. A. Olive et al. “Review of Particle Physics”. In: *Chin. Phys. C* 38 (2014), p. 090001. DOI: [10.1088/1674-1137/38/9/090001](https://doi.org/10.1088/1674-1137/38/9/090001) (cit. on p. 30).
- [19] Ivan Esteban et al. “The fate of hints: updated global analysis of three-flavor neutrino oscillations”. In: *JHEP* 09 (2020), p. 178. DOI: [10.1007/JHEP09\(2020\)178](https://doi.org/10.1007/JHEP09(2020)178). arXiv: [2007.14792](https://arxiv.org/abs/2007.14792) [[hep-ph](#)] (cit. on p. 31).
- [20] Fengpeng An et al. “Neutrino Physics with JUNO”. In: *J. Phys. G* 43.3 (2016), p. 030401. DOI: [10.1088/0954-3899/43/3/030401](https://doi.org/10.1088/0954-3899/43/3/030401). arXiv: [1507.05613](https://arxiv.org/abs/1507.05613) [[physics.ins-det](#)] (cit. on p. 31).
- [21] Rabindra N. Mohapatra and Goran Senjanovic. “Neutrino Mass and Spontaneous Parity Nonconservation”. In: *Phys. Rev. Lett.* 44 (1980), p. 912. DOI: [10.1103/PhysRevLett.44.912](https://doi.org/10.1103/PhysRevLett.44.912) (cit. on p. 31).
- [22] Samoil M. Bilenky and C. Giunti. “Lepton numbers in the framework of neutrino mixing”. In: *Int. J. Mod. Phys. A* 16 (2001), pp. 3931–3949. DOI: [10.1142/S0217751X01004967](https://doi.org/10.1142/S0217751X01004967). arXiv: [hep-ph/0102320](https://arxiv.org/abs/hep-ph/0102320) (cit. on p. 32).
- [23] Benjamin J. P. Jones. “The Physics of Neutrinoless Double Beta Decay: A Primer”. In: *Theoretical Advanced Study Institute in Elementary Particle Physics: The Obscure Universe: Neutrinos and Other Dark Matters*. Aug. 2021. arXiv: [2108.09364](https://arxiv.org/abs/2108.09364) [[nucl-ex](#)] (cit. on pp. 32–34).
- [24] Paul Langacker and Jing Wang. “Neutrino anti-neutrino transitions”. In: *Phys. Rev. D* 58 (1998), p. 093004. DOI: [10.1103/PhysRevD.58.093004](https://doi.org/10.1103/PhysRevD.58.093004). arXiv: [hep-ph/9802383](https://arxiv.org/abs/hep-ph/9802383) (cit. on p. 32).
- [25] S. R. Elliott, A. A. Hahn, and M. K. Moe. “Direct evidence for two-neutrino double-beta decay in ^{82}Se ”. In: *Phys. Rev. Lett.* 59 (18 1987), pp. 2020–2023. DOI: [10.1103/PhysRevLett.59.2020](https://doi.org/10.1103/PhysRevLett.59.2020) (cit. on p. 32).
- [26] Jonathan Engel and Javier Menéndez. “Status and Future of Nuclear Matrix Elements for Neutrinoless Double-Beta Decay: A Review”. In: *Rept. Prog. Phys.* 80.4 (2017), p. 046301. DOI: [10.1088/1361-6633/aa5bc5](https://doi.org/10.1088/1361-6633/aa5bc5). arXiv: [1610.06548](https://arxiv.org/abs/1610.06548) [[nucl-th](#)] (cit. on pp. 34, 35, 37).
- [27] Matteo Agostini, Giovanni Benato, and Jason Detwiler. “Discovery probability of next-generation neutrinoless double- β decay experiments”. In: *Phys. Rev. D* 96.5 (2017), p. 053001. DOI: [10.1103/PhysRevD.96.053001](https://doi.org/10.1103/PhysRevD.96.053001). arXiv: [1705.02996](https://arxiv.org/abs/1705.02996) [[hep-ex](#)] (cit. on p. 35).

- [28] Stefano Dell’Oro, Simone Marocco, and Francesco Vissani. “New expectations and uncertainties on neutrinoless double beta decay”. In: *Phys. Rev. D* 90.3 (2014), p. 033005. DOI: [10.1103/PhysRevD.90.033005](https://doi.org/10.1103/PhysRevD.90.033005). arXiv: [1404.2616 \[hep-ph\]](https://arxiv.org/abs/1404.2616) (cit. on p. 36).
- [29] M. Duerr et al. “On the quantitative impact of the Schechter-Valle theorem”. In: *Journal of High Energy Physics* 2011.6 (2011), pp. 1–19. DOI: [10.1007/JHEP06\(2011\)091](https://doi.org/10.1007/JHEP06(2011)091) (cit. on p. 36).
- [30] Mengjiao Xiao et al. “First dark matter search results from the PandaX-I experiment”. In: *Science China Physics, Mechanics & Astronomy* 57 (2014), pp. 2024–2030. DOI: [10.1007/s11433-014-5598-7](https://doi.org/10.1007/s11433-014-5598-7) (cit. on p. 40).
- [31] Chen Cheng et al. “Search for Light Dark Matter-Electron Scatterings in the PandaX-II Experiment”. In: *Phys. Rev. Lett.* 126.21 (2021), p. 211803. DOI: [10.1103/PhysRevLett.126.211803](https://doi.org/10.1103/PhysRevLett.126.211803). arXiv: [2101.07479 \[hep-ex\]](https://arxiv.org/abs/2101.07479) (cit. on p. 40).
- [32] Yue Meng et al. “Dark Matter Search Results from the PandaX-4T Commissioning Run”. In: *Phys. Rev. Lett.* 127.26 (2021), p. 261802. DOI: [10.1103/PhysRevLett.127.261802](https://doi.org/10.1103/PhysRevLett.127.261802). arXiv: [2107.13438 \[hep-ex\]](https://arxiv.org/abs/2107.13438) (cit. on p. 40).
- [33] Xun Chen et al. “PandaX-III: Searching for neutrinoless double beta decay with high pressure 136 Xe gas time projection chambers”. In: *Science China Physics, Mechanics & Astronomy* 60 (2017), pp. 1–40. DOI: [10.1007/s11433-017-9028-0](https://doi.org/10.1007/s11433-017-9028-0) (cit. on pp. 40, 45, 46).
- [34] H. J. Hilke. “Time projection chambers”. In: *Rept. Prog. Phys.* 73 (2010), p. 116201. DOI: [10.1088/0034-4885/73/11/116201](https://doi.org/10.1088/0034-4885/73/11/116201) (cit. on p. 40).
- [35] M. Agostini et al. “Final Results of GERDA on the Search for Neutrinoless Double- β Decay”. In: *Phys. Rev. Lett.* 125.25 (2020), p. 252502. DOI: [10.1103/PhysRevLett.125.252502](https://doi.org/10.1103/PhysRevLett.125.252502). arXiv: [2009.06079 \[nucl-ex\]](https://arxiv.org/abs/2009.06079) (cit. on pp. 41, 43).
- [36] B. Majorovits. “The search for $0\nu\beta\beta$ decay with the GERDA experiment: Status and prospects”. In: *AIP Conf. Proc.* 1672.1 (2015). Ed. by John L. Orrell, p. 110003. DOI: [10.1063/1.4928005](https://doi.org/10.1063/1.4928005). arXiv: [1506.00415 \[hep-ex\]](https://arxiv.org/abs/1506.00415) (cit. on p. 41).
- [37] N. Abgrall et al. “The Majorana Demonstrator Neutrinoless Double-Beta Decay Experiment”. In: *Adv. High Energy Phys.* 2014 (2014), p. 365432. DOI: [10.1155/2014/365432](https://doi.org/10.1155/2014/365432). arXiv: [1308.1633 \[physics.ins-det\]](https://arxiv.org/abs/1308.1633) (cit. on p. 41).
- [38] I. J. Arnquist et al. “Final Result of the Majorana Demonstrator’s Search for Neutrinoless Double- β Decay in Ge76”. In: *Phys. Rev. Lett.* 130.6 (2023), p. 062501. DOI: [10.1103/PhysRevLett.130.062501](https://doi.org/10.1103/PhysRevLett.130.062501). arXiv: [2207.07638 \[nucl-ex\]](https://arxiv.org/abs/2207.07638) (cit. on pp. 41, 43).
- [39] Jordan Myslik. “LEGEND: The Large Enriched Germanium Experiment for Neutrinoless Double-Beta Decay”. In: *13th Conference on the Intersections of Particle and Nuclear Physics*. Oct. 2018. arXiv: [1810.00849 \[physics.ins-det\]](https://arxiv.org/abs/1810.00849) (cit. on pp. 41, 43).
- [40] Benjamin Tam. “The SNO+ Experiment: Reactor & Solar ν Prospects”. In: *PoS NOW2022* (2023), p. 033. DOI: [10.22323/1.421.0033](https://doi.org/10.22323/1.421.0033). arXiv: [2211.05538 \[hep-ex\]](https://arxiv.org/abs/2211.05538) (cit. on p. 41).

- [41] Jeanne Wilson and Francesca Di Lodovico. “The SNO+ experiment”. English. In: *Journal of Instrumentation* 16.8 (Aug. 2021). ISSN: 1748-0221. DOI: [10.1088/1748-0221/16/08/P08059](https://doi.org/10.1088/1748-0221/16/08/P08059) (cit. on pp. 41, 43).
- [42] Antonio Branca. “The CUORE experiment at the LNGS”. In: *Prospects in Neutrino Physics*. Apr. 2017. arXiv: [1705.00005](https://arxiv.org/abs/1705.00005) [[physics.ins-det](#)] (cit. on p. 42).
- [43] D. Q. Adams et al. “Search for Majorana neutrinos exploiting millikelvin cryogenics with CUORE”. In: *Nature* 604.7904 (2022), pp. 53–58. DOI: [10.1038/s41586-022-04497-4](https://doi.org/10.1038/s41586-022-04497-4). arXiv: [2104.06906](https://arxiv.org/abs/2104.06906) [[nucl-ex](#)] (cit. on pp. 42, 43).
- [44] C. Augier et al. “Final results on the $0\nu\beta\beta$ decay half-life limit of ^{100}Mo from the CUPID-Mo experiment”. In: *Eur. Phys. J. C* 82.11 (2022), p. 1033. DOI: [10.1140/epjc/s10052-022-10942-5](https://doi.org/10.1140/epjc/s10052-022-10942-5). arXiv: [2202.08716](https://arxiv.org/abs/2202.08716) [[nucl-ex](#)] (cit. on pp. 42, 43).
- [45] A. Armato et al. “Toward CUPID-1T”. In: (Mar. 2022). arXiv: [2203.08386](https://arxiv.org/abs/2203.08386) [[nucl-ex](#)] (cit. on p. 42).
- [46] S. Abe et al. “Search for the Majorana Nature of Neutrinos in the Inverted Mass Ordering Region with KamLAND-Zen”. In: *Phys. Rev. Lett.* 130.5 (2023), p. 051801. DOI: [10.1103/PhysRevLett.130.051801](https://doi.org/10.1103/PhysRevLett.130.051801). arXiv: [2203.02139](https://arxiv.org/abs/2203.02139) [[hep-ex](#)] (cit. on pp. 42, 43).
- [47] P. Novella et al. “Demonstration of neutrinoless double beta decay searches in gaseous xenon with NEXT”. In: *JHEP* 09 (2023), p. 190. DOI: [10.1007/JHEP09\(2023\)190](https://doi.org/10.1007/JHEP09(2023)190). arXiv: [2305.09435](https://arxiv.org/abs/2305.09435) [[nucl-ex](#)] (cit. on p. 43).
- [48] J. J. Gomez-Cadenas. “Status and prospects of the NEXT experiment for neutrinoless double beta decay searches”. In: *54th Rencontres de Moriond on Electroweak Interactions and Unified Theories*. 2019, pp. 201–206. arXiv: [1906.01743](https://arxiv.org/abs/1906.01743) [[hep-ex](#)] (cit. on p. 43).
- [49] N. Ackerman et al. “The EXO-200 detector, part II: auxiliary systems”. In: *JINST* 17.02 (2022), P02015. DOI: [10.1088/1748-0221/17/02/P02015](https://doi.org/10.1088/1748-0221/17/02/P02015). arXiv: [2107.06007](https://arxiv.org/abs/2107.06007) [[physics.ins-det](#)] (cit. on p. 43).
- [50] G. Anton et al. “Search for Neutrinoless Double- β Decay with the Complete EXO-200 Dataset”. In: *Phys. Rev. Lett.* 123.16 (2019), p. 161802. DOI: [10.1103/PhysRevLett.123.161802](https://doi.org/10.1103/PhysRevLett.123.161802). arXiv: [1906.02723](https://arxiv.org/abs/1906.02723) [[hep-ex](#)] (cit. on pp. 43, 44).
- [51] G. Adhikari et al. “nEXO: neutrinoless double beta decay search beyond 1028 year half-life sensitivity”. In: *Journal of Physics G: Nuclear and Particle Physics* 49 (Jan. 2022), p. 015104. DOI: [10.1088/1361-6471/ac3631](https://doi.org/10.1088/1361-6471/ac3631) (cit. on p. 43).
- [52] Shaobo Wang. “The TPC detector of PandaX-III Neutrinoless Double Beta Decay experiment”. In: *JINST* 15.03 (2020). Ed. by Marzio Nessi, p. C03052. DOI: [10.1088/1748-0221/15/03/C03052](https://doi.org/10.1088/1748-0221/15/03/C03052). arXiv: [2001.01356](https://arxiv.org/abs/2001.01356) [[physics.ins-det](#)] (cit. on pp. 45, 46).
- [53] Laura Baudis et al. “Snowmass 2021 Underground Facilities & Infrastructure Frontier Report”. In: (Nov. 2022). arXiv: [2211.13450](https://arxiv.org/abs/2211.13450) [[hep-ex](#)] (cit. on p. 45).
- [54] H. Morgner. “Photo-and Penning ionization of molecules in the gas phase and in the liquid phase”. In: *Linking the Gaseous and Condensed Phases of Matter: The Behavior of Slow Electrons*. 1994, pp. 103–119. DOI: [10.1007/978-1-4615-2540-0_6](https://doi.org/10.1007/978-1-4615-2540-0_6) (cit. on p. 46).

- [55] S. Cebrian et al. “Micromegas-TPC operation at high pressure in xenon-trimethylamine mixtures”. In: *JINST* 8 (2013), P01012. DOI: [10.1088/1748-0221/10/07/E07001](https://doi.org/10.1088/1748-0221/10/07/E07001). arXiv: [1210.3287](https://arxiv.org/abs/1210.3287) [[physics.ins-det](#)] (cit. on p. 46).
- [56] Wenming Zhang et al. “Status and prospects of the PandaX-III experiment”. In: *JINST* 18.12 (2023), p. C12001. DOI: [10.1088/1748-0221/18/12/C12001](https://doi.org/10.1088/1748-0221/18/12/C12001). arXiv: [2311.13396](https://arxiv.org/abs/2311.13396) [[physics.ins-det](#)] (cit. on pp. 46, 51).
- [57] Y. Giomataris et al. “MICROMEGAS: A High granularity position sensitive gaseous detector for high particle flux environments”. In: *Nucl. Instrum. Meth. A* 376 (1996), pp. 29–35. DOI: [10.1016/0168-9002\(96\)00175-1](https://doi.org/10.1016/0168-9002(96)00175-1) (cit. on p. 47).
- [58] J. Manjarres et al. “Performances of Anode-resistive Micromegas for HL-LHC”. In: *JINST* 7 (2012). Ed. by Gregorio Bernardi, Sandro De Cecco, and Yuji Enari, p. C03040. DOI: [10.1051/epjconf/20122812071](https://doi.org/10.1051/epjconf/20122812071). arXiv: [1202.1074](https://arxiv.org/abs/1202.1074) [[physics.ins-det](#)] (cit. on p. 47).
- [59] I. Giomataris et al. “Micromegas in a bulk”. In: *Nucl. Instrum. Meth. A* 560 (2006), pp. 405–408. DOI: [10.1016/j.nima.2005.12.222](https://doi.org/10.1016/j.nima.2005.12.222). arXiv: [physics/0501003](https://arxiv.org/abs/physics/0501003) (cit. on p. 47).
- [60] S. Andriamonje et al. “Development and performance of Microbulk Micromegas detectors”. In: *JINST* 5 (2010), P02001. DOI: [10.1088/1748-0221/5/02/P02001](https://doi.org/10.1088/1748-0221/5/02/P02001) (cit. on pp. 47, 49).
- [61] Jianxin Feng et al. “A novel resistive anode using a germanium film for Micromegas detectors”. In: *Nucl. Instrum. Meth. A* 1031 (2022), p. 166595. DOI: [10.1016/j.nima.2022.166595](https://doi.org/10.1016/j.nima.2022.166595) (cit. on p. 48).
- [62] Sicheng Wen et al. “Design and fabrication of low background, high energy resolution thermal bonding Micromegas detectors for the PandaX-III experiment”. In: *Nucl. Instrum. Meth. A* 1062 (2024), p. 169206. DOI: [10.1016/j.nima.2024.169206](https://doi.org/10.1016/j.nima.2024.169206) (cit. on pp. 48, 49, 52).
- [63] Heng Lin and on behalf of the PandaX-III collaboration. “Latest progress of PandaX-III neutrinoless double beta decay experiment”. In: *Journal of Physics: Conference Series* 2502.1 (2023), p. 012007. DOI: [10.1088/1742-6596/2502/1/012007](https://doi.org/10.1088/1742-6596/2502/1/012007) (cit. on p. 50).
- [64] Shubin Liu et al. “Development of the Front-End Electronics for PandaX-III Prototype TPC”. In: *IEEE Trans. Nucl. Sci.* 66.7 (2019), pp. 1123–1129. DOI: [10.1109/TNS.2019.2907125](https://doi.org/10.1109/TNS.2019.2907125). arXiv: [1806.09257](https://arxiv.org/abs/1806.09257) [[physics.ins-det](#)] (cit. on p. 50).
- [65] Shebli Anvar et al. “AGET, the GET front-end ASIC, for the readout of the Time Projection Chambers used in nuclear physic experiments”. In: Oct. 2011. DOI: [10.1109/NSSMIC.2011.6154095](https://doi.org/10.1109/NSSMIC.2011.6154095) (cit. on p. 50).
- [66] Konrad Altenmüller et al. “REST-for-Physics, a ROOT-based framework for event oriented data analysis and combined Monte Carlo response”. In: *Computer Physics Communications* 273 (2022), p. 108281. DOI: <https://doi.org/10.1016/j.cpc.2021.108281> (cit. on pp. 50, 60, 145–149, 151, 153).
- [67] “Rest-for-physics Git Repository”. In: (2023). URL: <https://github.com/rest-for-physics/> (cit. on pp. 50, 60, 145).

- [68] S. Agostinelli et al. “GEANT4—a simulation toolkit”. In: *Nucl. Instrum. Meth. A* 506 (2003), pp. 250–303. DOI: [10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8) (cit. on p. 51).
- [69] Rob Veenhof. “GARFIELD user guide”. In: *Cern Progr. Libr. W* March (1989), pp. 1–54 (cit. on pp. 51, 155).
- [70] O. A. Ponkratenko, V. I. Tretyak, and Yu. G. Zdesenko. “The Event generator DECAY4 for simulation of double beta processes and decay of radioactive nuclei”. In: *Phys. Atom. Nucl.* 63 (2000), pp. 1282–1287. DOI: [10.1134/1.855784](https://doi.org/10.1134/1.855784). arXiv: [nucl-ex/0104018](https://arxiv.org/abs/nucl-ex/0104018) (cit. on pp. 51, 154).
- [71] Volodymyr Tretyak François Mauger. “BxDecay0 - C++ port of the legacy Decay0 FORTRAN library”. In: (2023). URL: <https://github.com/BxCppDev/bxdecay0> (cit. on pp. 51, 154).
- [72] Benjamin Manier. “Recherche d’événements double bêta sans émission de neutrinos grâce à une TPC de Xénon à haute pression lue par détection Micromegas auprès de l’expérience PandaX-III”. PhD thesis. Saclay, 2020 (cit. on pp. 53, 58).
- [73] J. Galan et al. “Topological background discrimination in the PandaX-III neutrinoless double beta decay experiment”. In: *J. Phys. G* 47.4 (2020), p. 045108. DOI: [10.1088/1361-6471/ab4dbe](https://doi.org/10.1088/1361-6471/ab4dbe). arXiv: [1903.03979](https://arxiv.org/abs/1903.03979) [[physics.ins-det](https://arxiv.org/abs/1903.03979)] (cit. on pp. 57, 128, 145).
- [74] H. Bethe and W. Heitler. “On the Stopping of fast particles and on the creation of positive electrons”. In: *Proc. Roy. Soc. Lond. A* 146 (1934), pp. 83–112. DOI: [10.1098/rspa.1934.0140](https://doi.org/10.1098/rspa.1934.0140) (cit. on p. 58).
- [75] Maria Grazia Ronga et al. “Back to the Future: Very High-Energy Electrons (VHEEs) and Their Potential Application in Radiation Therapy”. In: *Cancers* 13.19 (2021). ISSN: 2072-6694. DOI: [10.3390/cancers13194942](https://doi.org/10.3390/cancers13194942) (cit. on p. 59).
- [76] S.L. Andresen. “John McCarthy: father of AI”. In: *IEEE Intelligent Systems* 17.5 (2002), pp. 84–85. DOI: [10.1109/MIS.2002.1039837](https://doi.org/10.1109/MIS.2002.1039837) (cit. on p. 77).
- [77] Gio Wiederhold and John McCarthy. “Arthur Samuel: Pioneer in Machine Learning”. In: *IBM Journal of Research and Development* 36.3 (1992), pp. 329–331. DOI: [10.1147/rd.363.0329](https://doi.org/10.1147/rd.363.0329) (cit. on p. 77).
- [78] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386. DOI: [10.1037/h0042519](https://doi.org/10.1037/h0042519) (cit. on p. 78).
- [79] “Optimal Expected-Time Algorithms for Closest Point Problems”. In: *ACM Transactions on Mathematical Software* 6.4 (1980), pp. 563–580. DOI: [10.1145/355921.355927](https://doi.org/10.1145/355921.355927) (cit. on p. 78).
- [80] J. Ross Quinlan. “Induction of decision trees”. In: *Machine learning* 1 (1986), pp. 81–106. DOI: [10.1007/BF00116251](https://doi.org/10.1007/BF00116251) (cit. on p. 78).
- [81] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA: MIT Press, 1969 (cit. on p. 78).
- [82] Cristian Zanoci. “Exploring CNN-RNN Architectures for Multilabel Classification of the Amazon”. In: 2017. URL: <https://api.semanticscholar.org/CorpusID:53627682> (cit. on p. 78).

- [83] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323 (1986), pp. 533–536. URL: <https://api.semanticscholar.org/CorpusID:205001834> (cit. on pp. 78, 86, 157).
- [84] P. Arpaia et al. “Machine learning for beam dynamics studies at the CERN Large Hadron Collider”. In: *Nucl. Instrum. Meth. A* 985 (2021), p. 164652. DOI: [10.1016/j.nima.2020.164652](https://doi.org/10.1016/j.nima.2020.164652). arXiv: [2009.08109](https://arxiv.org/abs/2009.08109) [[physics.acc-ph](#)] (cit. on p. 78).
- [85] S. Whiteson and D. Whiteson. “Machine learning for event selection in high energy physics”. In: *Engineering Applications of Artificial Intelligence* 22.8 (2009), pp. 1203–1217. DOI: [10.1016/j.engappai.2009.05.004](https://doi.org/10.1016/j.engappai.2009.05.004) (cit. on p. 78).
- [86] Corey Adams et al. “Enhancing neutrino event reconstruction with pixel-based 3D readout for liquid argon time projection chambers”. In: *JINST* 15.04 (2020), P04009. DOI: [10.1088/1748-0221/15/04/P04009](https://doi.org/10.1088/1748-0221/15/04/P04009). arXiv: [1912.10133](https://arxiv.org/abs/1912.10133) [[physics.ins-det](#)] (cit. on p. 78).
- [87] M. Kekic et al. “Demonstration of background rejection using deep convolutional neural networks in the NEXT experiment”. In: *JHEP* 01 (2021), p. 189. DOI: [10.1007/JHEP01\(2021\)189](https://doi.org/10.1007/JHEP01(2021)189). arXiv: [2009.10783](https://arxiv.org/abs/2009.10783) [[physics.ins-det](#)] (cit. on p. 78).
- [88] Albert M Sirunyan et al. “Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques”. In: *JINST* 15.06 (2020), P06005. DOI: [10.1088/1748-0221/15/06/P06005](https://doi.org/10.1088/1748-0221/15/06/P06005). arXiv: [2004.08262](https://arxiv.org/abs/2004.08262) [[hep-ex](#)] (cit. on p. 78).
- [89] Pankaj Mehta et al. “A high-bias, low-variance introduction to Machine Learning for physicists”. In: *Physics Reports* 810 (2019). A high-bias, low-variance introduction to Machine Learning for physicists, pp. 1–124. ISSN: 0370-1573. DOI: <https://doi.org/10.1016/j.physrep.2019.03.001> (cit. on p. 78).
- [90] Georgia Karagiorgi et al. “Machine learning in the search for new fundamental physics”. English (US). In: *Nature Reviews Physics* 4.6 (June 2022), pp. 399–412. ISSN: 2522-5820. DOI: [10.1038/s42254-022-00455-1](https://doi.org/10.1038/s42254-022-00455-1) (cit. on p. 78).
- [91] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006 (cit. on p. 78).
- [92] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012 (cit. on pp. 78, 79, 81).
- [93] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cit. on p. 78).
- [94] Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022 (cit. on p. 80).
- [95] Andrew P. Bradley. “The use of the area under the ROC curve in the evaluation of machine learning algorithms”. In: *Pattern Recognition* 30.7 (1997), pp. 1145–1159. ISSN: 0031-3203. DOI: [https://doi.org/10.1016/S0031-3203\(96\)00142-2](https://doi.org/10.1016/S0031-3203(96)00142-2) (cit. on p. 81).
- [96] Jane V Carter et al. “ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves”. In: *Surgery* 159.6 (2016), pp. 1638–1645. DOI: [10.1016/j.surg.2015.12.029](https://doi.org/10.1016/j.surg.2015.12.029) (cit. on p. 81).

- [97] Ruo-Yu Sun. “Optimization for deep learning: An overview”. In: *Journal of the Operations Research Society of China* 8.2 (2020), pp. 249–294. DOI: [10.1007/s40305-020-00309-6](https://doi.org/10.1007/s40305-020-00309-6) (cit. on p. 81).
- [98] Yann LeCun, Y. Bengio, and Geoffrey Hinton. “Deep Learning”. In: *Nature* 521 (May 2015), pp. 436–44. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539) (cit. on p. 85).
- [99] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: Dec. 2014. arXiv: [1412.6980 \[cs.LG\]](https://arxiv.org/abs/1412.6980) (cit. on p. 87).
- [100] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations*. 2017. URL: <https://api.semanticscholar.org/CorpusID:53592270> (cit. on p. 87).
- [101] Sepp Hochreiter. “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6 (Apr. 1998), pp. 107–116. DOI: [10.1142/S0218488598000094](https://doi.org/10.1142/S0218488598000094) (cit. on pp. 87, 96).
- [102] Kunihiko Fukushima. “Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements”. In: *IEEE Trans. Syst. Sci. Cybern.* 5 (1969), pp. 322–333. URL: <https://api.semanticscholar.org/CorpusID:206799280> (cit. on p. 87).
- [103] M.T. Musavi et al. “On the training of radial basis function classifiers”. In: *Neural Networks* 5.4 (1992), pp. 595–603. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/S0893-6080\(05\)80038-3](https://doi.org/10.1016/S0893-6080(05)80038-3) (cit. on p. 87).
- [104] Brownlee Jason. “How to use Learning Curves to Diagnose Machine Learning Model Performance”. In: (2019). URL: <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/?ref=blog.roboflow.com> (cit. on p. 89).
- [105] Xue Ying. “An Overview of Overfitting and its Solutions”. In: *Journal of Physics: Conference Series* 1168.2 (2019), p. 022022. DOI: [10.1088/1742-6596/1168/2/022022](https://doi.org/10.1088/1742-6596/1168/2/022022) (cit. on p. 89).
- [106] Y. LeCun et al. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: [10.1162/neco.1989.1.4.541](https://doi.org/10.1162/neco.1989.1.4.541) (cit. on p. 89).
- [107] Maria Vakalopoulou et al. “Deep Learning: Basics and Convolutional Neural Networks (CNNs)”. In: *Machine Learning for Brain Disorders*. New York, NY: Springer US, 2023, pp. 77–115. ISBN: 978-1-0716-3195-9. DOI: [10.1007/978-1-0716-3195-9_3](https://doi.org/10.1007/978-1-0716-3195-9_3) (cit. on p. 90).
- [108] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: June 2016, pp. 770–778. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90) (cit. on pp. 90, 110).
- [109] Andreas Veit, Michael Wilber, and Serge Belongie. “Residual Networks Behave Like Ensembles of Relatively Shallow Networks”. In: *Advances in Neural Information Processing Systems* (May 2016), pp. 550–558. DOI: [10.5555/3157096.3157158](https://doi.org/10.5555/3157096.3157158) (cit. on pp. 90, 110).

- [110] Corey Adams et al. “Enhancing neutrino event reconstruction with pixel-based 3D readout for liquid argon time projection chambers”. In: *JINST* 15.04 (2020), P04009. DOI: [10.1088/1748-0221/15/04/P04009](https://doi.org/10.1088/1748-0221/15/04/P04009). arXiv: [1912.10133](https://arxiv.org/abs/1912.10133) [[physics.ins-det](#)] (cit. on p. 90).
- [111] Nima Lashkari. “Modular zero modes and sewing the states of QFT”. In: *JHEP* 21 (2020), p. 189. DOI: [10.1007/JHEP04\(2021\)189](https://doi.org/10.1007/JHEP04(2021)189). arXiv: [1911.11153](https://arxiv.org/abs/1911.11153) [[hep-th](#)] (cit. on p. 90).
- [112] Masahiro Yamatani et al. “New Particle Identification Approach with Convolutional Neural Networks in GAPS”. In: (Dec. 2022). DOI: [10.57350/jesa.9](https://doi.org/10.57350/jesa.9). arXiv: [2212.13454](https://arxiv.org/abs/2212.13454) [[astro-ph.IM](#)] (cit. on p. 90).
- [113] Tao Li et al. “Reconstruction of the event vertex in the PandaX-III experiment with convolution neural network”. In: *JHEP* 05 (2023), p. 200. DOI: [10.1007/JHEP05\(2023\)200](https://doi.org/10.1007/JHEP05(2023)200). arXiv: [2211.14992](https://arxiv.org/abs/2211.14992) [[hep-ex](#)] (cit. on pp. 92, 123, 125).
- [114] Hao Qiao et al. “Signal-background discrimination with convolutional neural networks in the PandaX-III experiment using MC simulation”. In: *Sci. China Phys. Mech. Astron.* 61.10 (2018), p. 101007. DOI: [10.1007/s11433-018-9233-5](https://doi.org/10.1007/s11433-018-9233-5). arXiv: [1802.03489](https://arxiv.org/abs/1802.03489) [[physics.ins-det](#)] (cit. on pp. 92, 128, 139, 141, 144).
- [115] A. Lobasenko. “Reconstruction of missing data of the PandaX-III experiment TPC using neural network”. In: *Journal of Instrumentation* 18.08 (2023), p. C08011. DOI: [10.1088/1748-0221/18/08/C08011](https://doi.org/10.1088/1748-0221/18/08/C08011) (cit. on p. 106).
- [116] Resve Saleh and A.K.Md.Ehsanes Saleh. *Statistical Properties of the log-cosh Loss Function Used in Machine Learning*. Aug. 2022. DOI: [10.48550/arXiv.2208.04564](https://doi.org/10.48550/arXiv.2208.04564) (cit. on p. 113).
- [117] A Helen Victoria and Ganesh Maragatham. “Automatic tuning of hyperparameters using Bayesian optimization”. In: *Evolving Systems* 12.1 (2021), pp. 217–223. DOI: [10.1007/s12530-020-09345-2](https://doi.org/10.1007/s12530-020-09345-2) (cit. on p. 132).
- [118] Ryan Turner et al. “Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020”. In: *Neural Information Processing Systems*. 2021. URL: <https://api.semanticscholar.org/CorpusID:233324399> (cit. on p. 132).
- [119] James Bergstra et al. “Algorithms for Hyper-Parameter Optimization”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor et al. Vol. 24. Curran Associates, Inc., 2011. URL: https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf (cit. on p. 133).
- [120] J. Castel et al. “Background assessment for the TREX Dark Matter experiment”. In: *Eur. Phys. J. C* 79.9 (2019), p. 782. DOI: [10.1140/epjc/s10052-019-7282-6](https://doi.org/10.1140/epjc/s10052-019-7282-6). arXiv: [1812.04519](https://arxiv.org/abs/1812.04519) [[astro-ph.IM](#)] (cit. on p. 145).
- [121] E. Armengaud et al. “Physics potential of the International Axion Observatory (IAXO)”. In: *Journal of Cosmology and Astroparticle Physics* 2019.06 (2019), p. 047. DOI: [10.1088/1475-7516/2019/06/047](https://doi.org/10.1088/1475-7516/2019/06/047) (cit. on p. 145).

- [122] V. Anastassopoulos et al. “New CAST Limit on the Axion-Photon Interaction”. In: *Nature Phys.* 13 (2017), pp. 584–590. DOI: [10.1038/nphys4109](https://doi.org/10.1038/nphys4109). arXiv: [1705.02290](https://arxiv.org/abs/1705.02290) [[hep-ex](#)] (cit. on p. 145).
- [123] R. Brun and F. Rademakers. “ROOT: An object oriented data analysis framework”. In: *Nucl. Instrum. Meth. A* 389 (1997). Ed. by M. Werlen and D. Perret-Gallix, pp. 81–86. DOI: [10.1016/S0168-9002\(97\)00048-X](https://doi.org/10.1016/S0168-9002(97)00048-X) (cit. on p. 145).
- [124] I. Antcheva et al. “ROOT: A C++ framework for petabyte data storage, statistical analysis and visualization”. In: *Comput. Phys. Commun.* 182 (2011), pp. 1384–1385. DOI: [10.1016/j.cpc.2011.02.008](https://doi.org/10.1016/j.cpc.2011.02.008) (cit. on p. 145).
- [125] S. Agostinelli et al. “GEANT4—a simulation toolkit”. In: *Nucl. Instrum. Meth. A* 506 (2003), pp. 250–303. DOI: [10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8) (cit. on p. 145).
- [126] Dorothea Pfeiffer et al. “Interfacing Geant4, Garfield++ and Degrad for the Simulation of Gaseous Detectors”. In: *Nucl. Instrum. Meth. A* 935 (2019), pp. 121–134. arXiv: [1806.05880](https://arxiv.org/abs/1806.05880) [[physics.ins-det](#)] (cit. on p. 145).
- [127] J. Castel et al. “The TREX-DM experiment at the Canfranc Underground Laboratory”. In: *J. Phys. Conf. Ser.* 1468.1 (2020). Ed. by Masayuki Nakahata, p. 012063. DOI: [10.1088/1742-6596/1468/1/012063](https://doi.org/10.1088/1742-6596/1468/1/012063). arXiv: [1910.13957](https://arxiv.org/abs/1910.13957) [[astro-ph.GA](#)] (cit. on p. 147).
- [128] Theopisti Dafni and Javier Galán. “Digging into Axion Physics with (Baby)IAXO”. In: *Universe* 8.1 (2022), p. 37. DOI: [10.3390/universe8010037](https://doi.org/10.3390/universe8010037). arXiv: [2112.02286](https://arxiv.org/abs/2112.02286) [[astro-ph.IM](#)] (cit. on p. 148).