

Universidade do Minho
Escola de Ciências

Rui Lucas Oliveira Faria Ferreira

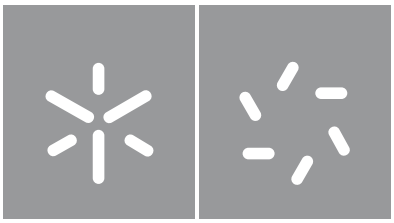
Machine-Learning Methods for Scatter Classification at LZ

Machine-Learning Methods for Scatter Classification at LZ

Rui Ferreira

UMinho | 2025

Abril de 2025



Universidade do Minho
Escola de Ciências

Rui Lucas Oliveira Faria Ferreira

Machine-Learning Methods for Scatter Classification at LZ

Dissertação de Mestrado
Mestrado em Física

Trabalho efetuado sob a orientação de
Professor Doutor Nuno Filipe da Silva Fernandes de Castro
Professor Doutor Paulo Alexandre Brinca da Costa Braz

Abril de 2025

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Atribuição

CC BY

`https://creativecommons.org/licenses/by/4.0/`

Acknowledgments

It would be remiss of me not to dedicate the first words of my acknowledgments to my parents - it was their sacrifices that allowed me to study in the field I've always loved and ultimately produce this work. I also cannot overlook the contributions of the rest of my family (my siblings and cousins would not forgive me if I did), my friends, and Cam - all of whom supported me, encouraged me and pushed me to create something I could be proud of. This work is dedicated to all of them: a small return for all they gave me, but a heartfelt one nevertheless.

I would also like to leave a few words for my advisers, Paulo Brás and Nuno Castro. I want to thank Paulo for his week-to-week guidance, his patience, and his ability to clearly share his knowledge and expertise. To Nuno, I'm grateful for his insights — both into the work itself and into the process behind its creation.

Along the path of developing the work described in this thesis, I was fortunate to receive advice and feedback from many people whose expertise far exceeded my own. I would like to thank the Machine Learning group from LZ, the LZ group from LIP-Coimbra, and Miguel Romão for taking their time to listen to me and share their invaluable feedback.

Rui Lucas Ferreira

Statement of Integrity

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Resumo

Métodos de Machine-Learning para Classificação de Espalhamentos em LZ

Esta dissertação explora a aplicação de métodos de aprendizagem automática (Machine Learning - ML) para a identificação de interações na experiência LUX-ZEPLIN (LZ), uma dos principais experiências de deteção direta de matéria escura (Dark Matter - DM). Esta tarefa exige uma análise de dados bastante precisa para permitir distinguir entre fundo e sinal. Um passo da análise de LZ é determinar o fenómeno que originou o evento - os fenómenos mais relevantes para esta tese são espalhamentos únicos e espalhamentos múltiplos. O foco principal é o desenvolvimento e avaliação de duas abordagens de ML distintas: um método modular para identificar os pulsos relevantes (proeminentes) dentro dos eventos de LZ, e um método direto baseado nas formas de onda do evento para a classificação do espalhamento.

A primeira abordagem, denominada ProFinder, utiliza uma rede neural baseada numa camada Long Short-Term Memory (LSTM) para classificar os pulsos como proeminentes ou não proeminentes com base em Quantidades Reduzidas (Reduced Quantities - RQs) derivadas do Programa de Análise de LZ (LZap). O modelo é treinado num conjunto de dados diversificados com eventos de calibração e de fundo; o seu desempenho é avaliado nos dados de treino e em dados não representados nos dados de treino, incluindo eventos de fundo e eventos de calibração com neutrões de deutério-deutério (DD).

A segunda abordagem centra-se na classificação de espalhamentos baseada diretamente na resposta dos sensores de luz de LZ ao evento, onde uma rede neural que utiliza camadas convolucionais unidimensional (one-dimensional convolutional layers - Conv1D) é utilizada para processar o evento e classificá-lo diretamente como Espalhamento Único (Single Scatter - SS) ou Não Espalhamento Único (Non-Single Scatter - NonSS). Este método visa fornecer um caminho mais independente do LZap para a classificação do espalhamento.

Ambas abordagens apresentam desafios comuns, como a baixa quantidade e qualidade de dados de treino; apresentam também desafios específicos, tal como a exigência computacional da segunda abordagem. As soluções encontradas para resolver ou contornar estes desafios estão também expostas nesta dissertação.

Palavras-Chave: Aprendizagem automática, Deteção direta de matéria escura.

Abstract

Machine-Learning Methods for Scatter Classification in LZ

This dissertation explores the application of machine learning (ML) methods for scatter identification in the LUX-ZEPLIN (LZ) experiment, a leading dark matter (DM) direct detection experiment. This task demands an accurate data analysis to allow the distinction between signal and background. A step in the LZ analysis is to determine the type of phenomena that originated the event - the most relevant phenomena for this thesis are single scatters and multiple scatters. The primary focus is on developing and evaluating two distinct ML approaches: a modular method for identifying the relevant (prominent) pulses within LZ events, and a direct signal waveform-based method for scatter classification.

The first approach, named ProFinder, leverages a Neural Network based on a Long Short-Term Memory (LSTM) layer to classify pulses as prominent or non-prominent based on Reduced Quantities (RQs) derived from the LZ Analysis Program (LZap). The model is trained on a diverse dataset of calibration and background events; its performance is evaluated on both training and stress-testing datasets, including background-only and deuterium-deuterium (DD) neutron calibration events.

The second approach focuses on direct scatter classification from the response of the light sensors in LZ to the event, where a one-dimensional convolutional neural network (CNN) is employed process the event and to classify them as Single Scatter (SS) or Non-Single Scatter (NonSS). This method aims to provide a more LZap-independent pathway for scatter classification .

Both approaches have common challenges, such as the low amount and quality of training data; both present specific challenges such as the computational demands for the second approach. The solutions that were found to solve these challenges are also presented in this dissertation.

Keywords: Dark matter direct detection, Machine-learning.

Contents

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros i

Acknowledgments ii

Statement of Integrity iii

Resumo iv

Abstract v

List of Acronyms ix

List of Figures xii

List of Tables xiv

1 The LZ Experiment 4

1.1 Dark Matter 4

1.1.1 WIMP Dark Matter 5

1.2 LZ Detector Design 7

1.2.1 Dual Phase Xenon Time-Projection Chamber 8

1.2.1.1 PMTs 11

1.2.1.2 Nuclear vs Electronic Recoils 12

1.2.2 Xenon Veto System 13

1.2.3 Dark Matter search results from the LZ experiment 15

1.3 Non Dark Matter Searches 16

1.4 Summary 17

2 The Anatomy of an LZ Event **18**
2.1 LZap - LZ Analysis Program . . . 18
2.1.1 PhotonDetection . . . 19
2.1.2 InteractionDetection . . . 22
2.2 Pulse Highlighter . . . 23
2.3 Summary . . . 25

3 Neural Networks **27**
3.1 Supervised Learning . . . 27
3.1.1 Activation function . . . 28
3.2 Feedforward Neural Networks . . . 28
3.2.1 Parameter optimization . . . 29
3.3 Recurrent Neural Networks . . . 30
3.3.1 Long Short-Term Memory . . . 31
3.4 Convolutional Neural Networks . . . 33
3.4.1 Convolution . . . 33
3.4.2 Pooling . . . 34
3.4.3 One-Dimensional Convolutional Layers . . . 35
3.5 Hyperparameter Optimization . . . 36
3.6 Summary . . . 37

4 The datasets **38**
4.1 LZ Calibration Sources . . . 39
4.1.1 Gaseous Calibration Sources . . . 40
4.1.2 Neutron Calibration Sources . . . 40
4.2 Training Dataset . . . 42
4.3 Stress Testing Datasets . . . 43
4.4 Summary . . . 44

5 ProFinder **46**
5.1 Dataset(s) . . . 46
5.2 Methodology . . . 47
5.2.1 Model Architeture and Hyperoptimization . . . 48
5.3 Training and Results . . . 49
5.3.1 Stress Testing . . . 52
5.3.1.1 BG Dataset . . . 53

5.3.1.2 DD Dataset 54
5.3.2 Scatter Type Prediction 55
5.4 Commentary and Further Work 58
5.5 Summary 59

6 Waveform-based Scatter Classification **60**
6.1 The dataset 60
6.2 Model Architecture 64
6.3 Model Training and Testing 66
6.4 Commentary and Further Work 68
6.5 Summary 69

7 Closing remarks **70**

Bibliography **72**

Appendix A Reduced Quantities Defined **79**

Appendix B ProFinder Individual Models Metrics **82**

Appendix C Cuts for the Scatter Classification Data **84**

Appendix D Hyperoptimization information **86**
D.1 ProFinder Hyperoptimization 86
D.2 Waveform Hyperoptimization 87

List of Acronyms

5S Dataset Five Sources dataset. 36, 40–44, 49, 54, 56, 77

AmLi americium-lithium. 38–40, 48, 50

AUC Area Under the Curve. 48, 63

BG background. 22

BG Dataset background-only dataset. 36, 40–43, 49, 53, 54, 65, 77, 78

BG-HE high-energy background. 39, 48, 79

BG-LE low-energy background. 39, 48

BiLSTM Bidirectional LSTM. 45

CH3T tritiated methane. 24, 39, 48

CL confidence level. 45, 47

CLth CL threshold. 47, 48, 50, 51

CNN Convolutional NN. 31–33, 35

Conv1D One-dimensional Convolutional layer. 34, 35, 59–61, 65

DD deuterium-deuterium. 13, 36, 38, 39, 41, 42, 50, 51, 53, 54, 65

DD Dataset deuterium-deuterium dataset. 36, 41–43, 50, 53, 54, 65, 77, 78

DM Dark Matter. 5–8, 16, 17

ER Electric Recoil. 10, 13, 16, 22, 38, 39, 48

GdLS Gadolinium-loaded Liquid Scintillator. 8, 9, 14

HADES Heuristics Algorithm for Discrimination of Event Substructures. 20, 21, 23

HG High Gain. 18, 20

LG Low Gain. 18, 20

LSTM Long Short-Term Memory. 30, 31, 35, 43–45, 54, 64, 65

LUX Large Underground Xenon. 5, 11

LXe Liquid Xenon. 10–12, 14, 38, 40, 41

LZ LUX-ZEPLIN. 2, 5, 7, 8, 10, 11, 14–18, 20, 25, 36–38, 41, 49, 56, 58, 66

LZap LZ Analysis Program. 18–21, 25, 36, 40, 41, 43, 44, 50–53, 56, 58, 63–66

MG Mix Gain. 18

ML Machine Learning. 26

MPE Multiple Photoelectron. 20, 21

MS Multiple Scatter. xiv, 22, 41, 51, 52, 57, 62, 79

MSE mean squared error. 28, 46, 59, 60

NN Neural Network. 26–30, 32, 64

NR Nuclear Recoil. 7, 10–14, 16, 38–40, 48, 50, 53

OD Outer Detector. 8, 14, 18, 22, 37

OS Other Scatter. xiv, 52, 62, 63, 79

PMT Photo-Multiplier Tube. 8–14, 18, 21, 25, 36, 37, 42, 56

POD Pulse Only Digitization. 18, 20, 56

PR Precision-Recall. 46, 47

ProS1 Prominent S1. 21–23, 39, 52, 53, 57, 59, 60

ProS2 Prominent S2. 21–23, 39–41, 44–46, 48, 50–53, 56–60, 65

RNN Recurrent NN. 29, 30, 35, 43, 54

ROC Receiving Operator Characteristic. 50, 51, 63

RQ Reduced Quantity. 18, 20, 21, 23, 25, 36, 43, 44, 48, 51, 56, 63, 74, 76

S1 primary scintillation. 9, 12, 13, 20–23, 25, 40, 42–44, 54, 57–59, 63, 64

S2 electroluminescence signal. 9, 12, 13, 20–25, 40–45, 48, 54, 56, 57, 59, 64

SE Single Electron. 20, 21, 57

SMPP Standard Model of Particle Physics. 6–8, 16

SPE Single Photoelectron. 20, 57

SS Single Scatter. xiv, 22, 36, 38, 40, 41, 51, 52, 57–59, 62–65, 79

SURF Sanford Underground Research Facility. 8

TPC Time-Projection Chamber. 5, 7–12, 14, 16–18, 20, 22, 37–39, 41, 56–58, 65

WIMP Weakly Interacting Massive Particle. 6–8, 13–16, 22, 36, 38

XDS Xenon Detector System. 10, 11, 14

Xe131m metastable xenon-131. 39, 48, 79

XeSkin Xenon Skin. 14

ZEPLIN ZonEd Proportional scintillation in LIquid Noble gases. 5

List of Figures

1 Effective Feynman Diagram of a DM-SM Interaction . . . 7
2 LZ Schematic . . . 8
3 LZ Event . . . 9
4 Xenon Detector System . . . 10
5 DM Interaction Integral Count Rate . . . 11
6 PMT Arrays . . . 12
7 NR/ER bands . . . 13
8 Comparison of LZ Results . . . 15

9 LZap PhotonDetection Schematic . . . 19
10 ^{131m}Xe Event . . . 20
11 AmLi Event . . . 21
12 LZap InteractionFinder Schematic . . . 22
13 *PulseHighlighter Scheme* . . . 24
14 Comparison with Largest S2 . . . 24
15 Area Ratio Test for S2 Pulses . . . 25
16 Area Ratio Test for S2 Pulses . . . 25

17 Feedforward NN Schematic . . . 29
18 RNN cell . . . 31
19 LSTM Cell Schematic . . . 32
20 Schematic for the convulotion operation in a 2D array . . . 33
21 Schematic of the Pooling of a 2D array . . . 35
22 1D Convolution . . . 36

23 Direct DD Events Spatial Distribution . . . 41

24 5S S1/S2 space . . . 42
25 ProS2 area contribution for 5S . . . 43
26 S1/S2 pulses against total number of pulses in 5S . . . 43
27 All datasets S1/S2 space . . . 44
28 ProS2 Length Against Peak Amplitude for the Datasets . . . 44

29 ProFinder Model Architecture . . . 49
30 ProFinder Validation PR Curve . . . 50
31 ProFinder Output on Test Subset . . . 51
32 ROC Curves for the Test Subset . . . 51
33 ProFinder Test Output RQ Space . . . 52
34 ProFinder Output on BG Dataset . . . 53
35 ROC Curves for the NewBG Dataset . . . 54
36 ProFinder Output on DD Subset . . . 55
37 ROC Curves for the DD Dataset . . . 55
38 DD Wrongly and Correctly Classified ProS2 Length vs PeakAmp . . . 56
39 Predicted vs Expected Number of ProS2 . . . 57

40 Time Interval Cut . . . 62
41 Pooling Effect on ProS1 . . . 63
42 Waveform Transformation . . . 63
43 ProS1/2 Length . . . 64
44 Waveform Model Architecture . . . 66
45 Waveform Training History . . . 67
46 Performance evaluation of the model through various metrics. . . . 67

47 Individual ProFinder models metrics for the 5S Dataset . . . 82
48 Individual ProFinder models metrics for the DD Dataset . . . 83
49 Individual ProFinder models metrics for the BG Dataset . . . 83

50 Cuts based on number of specific pulses in Event . . . 84

51 Characterization of the ProFinder Hypertraining . . . 87
52 Hyperparameters values for each trial. . . . 87
53 Characterization of the Waveform to Scatter Hypertraining . . . 88
54 Hyperparameters values for each trial. . . . 89

List of Tables

1 Information about the number of events in the datasets 38
2 LZ Calibration Sources 39

3 ProFinder Hyperparameters 49
4 Comparison between confident and unconfident pulse classification. 51
5 Comparison between confident and unconfident pulse classification. 54
6 Comparison between confident and unconfident pulse classification. 55
7 Scatter confusion matrix for different datasets. 57

8 Effect of the Data Cuts in the Number of Events in the 5S Dataset 61
9 Waveform Classifier Hyperparameters 66
10 Extended confusion matrix, dividing the NonSS in OS and Multiple Scatter (MS). . . . 68

11 RQ Description 81

12 Event distribution after applying various cuts. The numbers in this table do not represent the commulative cuts, i.e., some events are cut more than once. 85

Introduction

Motivation and Goals

The discrepancy between observed phenomena and theoretical predictions is the main motivation for scientific research. It was due to a such a discrepancy that the concept of Dark Matter was first proposed: a significant difference between the observed and the expected rotation velocity of the celestial bodies in Coma galaxy cluster pointed to the presence of large amounts of invisible matter distributed in the cluster. Other astronomical observations, such as the profile of galaxy rotation curves and weak gravitational lensing effects, further supported the Dark Matter hypothesis. If Dark Matter was to exist, it would have to be weakly interactive, massive and present in great quantities - five times more than the amount of non-Dark Matter present in the Universe.

These characteristics imposes restraints to the type of particles that may compose Dark Matter (if it is composed by particles). Many particles were proposed as an hypothetical dark matter particle: one of the most popular were particles known as WIMPs, weakly interactive particles with masses in the range of the weak scale. The appeal of this class of particles is that many extensions to the Standard Model of Particle Physics naturally produce a WIMP-like particle (the lightest stable particle). Additionally, if the (self) interaction of these particles is mediated by the weak nuclear force, then their thermal production in the early universe would explain the expected current cosmic abundance .

On the other hand, these characteristics also present challenges to the task of detection - the proof of existence. Since Dark Matter does not significantly interact with known matter, detectable interactions should be extremely rare. In addition, the very nature of these interactions is not known and therefore the search for Dark Matter is similar to finding a specific grain of sand in a huge beach. Nevertheless, there are strategies that may be able to accomplish this goal; one of them is direct detection of Dark Matter - a WIMP, for example, would interact with a Standard Model particle and produce a signature effect that could be detected.

LZ is an experiment for WIMP direct detection. It uses a technology known as Dual-Phase Time

Projection Chamber to create this signature signal - two consecutive pulses originated from the recoil of a Xenon nuclei caused by an interaction with an incoming WIMP. To be able to unambiguously observe this interaction, the design of LZ is directed to reduce the possible background sources (phenomena that can cover or even mimic the WIMP signal) through shielding and veto techniques. Event reconstruction is also a crucial step in WIMP search analysis: the LZ data is recorded as a time-sequence of pulses forming a waveform that is processed, characterized and classified by the data analysis strategy from LZ (the LZap algorithm). The general data flow is the following: the waveforms are calibrated and the individual pulses are identified, characterized and classified according to their own characteristics; then, the relevant pulses are reclassified using event context (the pulses are distinguished between prominent or non-prominent). Using the prominent pulses, the type of scatter that originated the event is determined and its energy and position are reconstructed.

The scatter classification is an important task to veto background events: if a WIMP interacts coherently with the xenon nucleus, it is expected that it will scatter only once and then leave the detector (due to the low WIMP-Xenon interaction cross section, a double scatter is extremely improbable). This is known as a Single Scatter. Other types of phenomena may scatter multiple times or even produce pulse sequences that are not expected from a typical nuclei recoil. The scatter classification is mostly dependent on the previous analysis, such as the identification of prominent pulses. This is done using a decision tree based on the relative area and width of the pulses in the context of the event.

The work developed and presented in this dissertation depicts two exploratory Machine-Learning approaches to prominence identification and scatter classification: the former is a modular approach, where the information about the pulses characterization and classification from LZap is used as a paneled data to determine which pulses are prominent. The expected advantages of this strategy is the inclusion of more data about the pulses and the inclusion of temporal distances between the pulses. The latter is a more direct path to scatter classification, using the waveform directly as input to the machine-learning model. The goal is to build a binary model to discriminate between Single Scatters and Non Single Scatters. Due to various limitations, from data constraints to limited computational resources, the work presented in this dissertation is in no way a ready-to-use algorithm, but an initial exploration of the possibility of integrating deep machine learning methods in the data analysis of the LZ data.

Dissertation Structure

In Chapter 1, the LZ Experiment is discussed. The discussion starts with a brief review of Dark Matter, in order to provide proper context to the experiment, followed by a description of the LZ architecture, with special focus on the center piece of the experiment - the Dual-Phase Xenon Time-Projection Chamber. The most recent dark matter search results from LZ are also described, as well as a very brief list of different

avenues for scientific exploration possible within LZ. Chapter 2 describes the LZap algorithm, including a detailed deconstruction of the *PulseHighlighter* module - the algorithm responsible for identifying which pulses are prominent.

Chapter 3 introduces the machine-learning concepts and techniques that were employed during the work that is described in this dissertation - Neural Networks. Chapter 4 describes the datasets that were utilized in this work (the training dataset is common to both approaches). All events were recorded during LZ runs and calibrations.

Chapter 5 discusses the modular prominence identifier approach - ProFinder. The methodology is described and the results in the pulse prominence classification is depicted for each dataset. The model prominent pulses are used to determine the scatter type of each event. The chapter ends with a brief synthesis of the challenged faced during the work and some commentary on the results and possible further work. Chapter 6 is analogous to the previous chapter but for the scatter classification directly from the waveforms. Since this approach is more computational demanding, the treatment of the training dataset is discussed in more depth.

Finally, some final remarks on the work developed during this dissertation are exposed in Chapter 7.

Chapter 1

The LZ Experiment

LUX-ZEPLIN (LZ) is a Dark Matter (DM) searching experiment, located in Lead, South Dakota, USA. Thirty-eight institutes, distributed across four continents participate in the LZ collaboration - the USA in America; the UK, Switzerland and Portugal in Europe; Korea in Asia; and Australia in Oceania. The main detector of LZ consists of a dual-phase Xenon Time-Projection Chamber (TPC), a common technology in other DM experiments, such as Large Underground Xenon (LUX) and ZonEd Proportional scintillation in LIquid Noble gases (ZEPLIN), the two collaborations that merged to create LZ. LZ is one of three current generation, multi-tonne Xe TPC DM experiments, alongside XENONnT (located in Italy) and PandaX (located in China). This chapter will describe the components that make up the LZ Experiment, focusing on the TPC apparatus. To properly understand the reasoning behind some design decisions, the chapter begins with a brief overview of the main scientific goal of the LZ experiment: Dark Matter detection.

1.1 Dark Matter

It was through astronomic observations that the existence of a type of (massive) matter that did not interact via electromagnetic and strong force (hence, the denomination *dark*) was first considered. The main interaction of this new type of matter is therefore gravitic: the existence of a non-luminous matter was already being discussed between 19^{th} century astronomers (mainly under the form of non-luminous stars) [1]. The 20^{th} century saw the proliferation of galaxy studies: swiss astronomer Franz Zwicky, often considered the pioneer of DM studies, studied the Coma system: he pointed out that the velocity dispersion of the cluster indicated a mass density 400 times higher than the cluster density inferred through its observation. He then extrapolated the existence of Dark Matter and that it should exist in greater quantities than luminous matter [2].

As technological and astronomical knowledge advanced, new (indirect) evidence was found, for example, in studies of the rotation speed of galaxies and of anisotropies in the Cosmic Microwave Background

radiation (CMB) [3]. The Standard Model of Cosmology (also known as Lambda Cold Dark Matter - ΛCDM) includes a free parameter related to the energy-density of dark matter in the Universe, to be determined through astronomical observations. Current estimates of this value, using measurements of the anisotropies in the CMB, point to Dark Matter accounting for 26.4% of the total mass-energy content of the Universe (the remaining is divided between baryonic matter - 5.1% - and dark energy - 68.5%) [4] justifying the scientific interest that has surrounded this hypothetical matter. If it really exists, the local density (at the Solar System level), ρ of DM is expected to be approximately 0.3 GeV cm^{-3} $\approx 5 \times 10^{-31}$kg dm^{-3} [5] .

It is important to reinforce that these observations are not direct evidence, but merely demonstrate a inconsistency with the current theories of gravity - Dark Matter provides a possible explanation to these problems while still being consistent with those theories. Other viable alternative explanations are Modified Gravity theories, which seek to, somewhat obviously, modify gravity on a cosmological scale. Extensive research has also been employed on this field [6], but conclusive evidence for any of these scenarios is yet to appear.

Admitting the existence of DM, the next question will undoubtedly be of its nature. There are a great number of hypothesis, many of them modeling Dark Matter as (non-baryonic) particles. In that case, the Standard Model of Particle Physics (SMPP), one of the most successful theories of all time, would have to be extended to account for it, since the SMPP particles cannot fully account for the expected DM energy-density. But the possible future inclusion of a new DM particle may have a bright side: some proposed extensions to the SMPP, developed to solve some problems of the theory, may naturally produce particles with the right characteristics to be a Dark Matter particle [7, 8].

1.1.1 WIMP Dark Matter

There is no reason DM should be perceived as a monolithic matter. Neutrinos are SMPP particles that could be thought as DM: they are invisible, stable and they have mass. In the same vein, there are macroscopic, baryonic, astronomic bodies that do not substantially emit light, such as brown and neutron stars or black holes. These can also be thought as a dark matter, known as MAssive Compact Halo Objects (MACHO). But they both have already been proven to be minor DM content contributors by cosmological observations [9].

Then, if a particle is to be a fundamental building block of a big portion of the Dark Matter mass-energy content, it must fulfill some conditions: it must be stable, at least on a time-scale larger than the age of Universe; it must be non-baryonic - as cosmological observations seem to indicate DM interacts weakly with baryonic matter; it must not interact (or interact very weakly) with light; and it should be cold i.e. non-relativistic.

The Weakly Interacting Massive Particle (WIMP) are a class of particles arising from different SMPP

extension proposals that fit DM criteria. The group is comprised by particles with a proposed mass in the range of $m_{\text{WIMP}} \approx m_{\text{weak}} \approx 10$ GeV $- 1$TeV and their existence is mainly motivated by three reasons: they naturally arise from some proposed extensions to the SMPP; they produce the correct relic density expected for DM (the well-known *WIMP miracle*) [10]; and their detection is, in principle, possible through direct and indirect methods. This conjugation of factors motivates WIMPs as one of the most studied DM candidates.

The relic density of the WIMPs is supposed to be caused by the freeze-out mechanism: when the Universe began, it was hot and dense, and all particles were in thermal equilibrium. Due to the rapid expansion and consequent cooling of the Universe, the processes energetic enough to create WIMPs became rarer; on the other hand, WIMPs became sparse enough to not be annihilated by another WIMPs. Eventually, the temperature fell down enough to knock the WIMPs out of equilibrium and only a small (relic) density was left in the Universe [11].

An effective theory that would allow a DM SM $\rightarrow$ DM SM scatter, represented in Figure 1, can be supposed. If WIMPs exist, they are expected to cross the Earth Solar System in bulk. Their speed relative to Earth would be time dependent as the moviment of the Earth and the Solar System move across the galaxy - nevertheless, WIMP velocity as measured in the lab frame would be expected to form a Maxwell-Boltzmann distribution (similar to an ideal gas, due to their lack of interactions) with an average value around 240 km/s [5]. A possible strategy to detect these particles and provoke the DM SM scatter would be to setup a stationary target of SM particles. The choice of the SM target should be one that maximizes the transferred energy during the coherent elastic scattering between the two particles (the energy transfer is maximized when the colliding particles have the same mass) to facilitate detection - therefore, (stable) atomic nuclei with masses around the weak scale present themselves as optimal targets for WIMPy dark matter. The interaction would cause a Nuclear Recoil (NR) that could be detected with proper technology and conditions. The expected scatter would cause an energy transfer in the order of the few dozens of KeVs, therefore, highly sensible detectors with robust background rejection mechanisms are necessary to probe into the possibility of this scatter [12, 13]. This process is an example of direct detection and it is the approach adopted by LZ and other next-generation dual phase Xenon TPC detectors.

There are alternative methods to search for DM particles. Rotating the implicit time arrow 90 degrees downwards, on Figure 1, we would obtain a DM self-annihilation interaction, creating in SMPP products. Thus, detecting the product of DM annihilation should also be a viable approach to detection - an indirect approach. A potential outcome of this self-annihilation process is the production of γ-rays. Instruments such as the Fermi Large Area Telescope can examine the inner regions of the Milky Way or entire dwarf galaxies, analyzing the γ-ray spectrum and intensity to compare them with the expected emissions from a galaxy permeated by dark matter of certain masses [14]. Another possible product of the self-annihilation are neutrinos. The signature of these neutrinos would be significantly different from the solar and Earth neutrinos that could be differentiated by sensitive neutrino experiments such as the SuperKamiokande

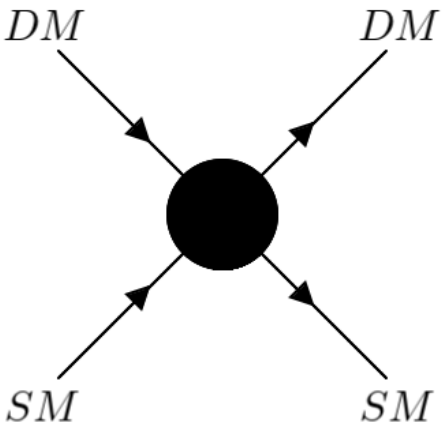


Figure 1: Effective Feynman diagram of the interaction of a DM particle with a SMPP particle.

[15, 16]

Another venue for investigation is the DM production in particle colliders - the annihilation of two SMPP particles creating DM products. The possible DM products would leave the detector without trace but other collision debris would be observed. Colliders like the LHC are set up so there is no momentum transverse to the particle beams - consequently, if the sum of the transverse momentum of the observed collision products is not zero, it is a clear indication that there are collisions products not observed by the detector, carrying the remaining transverse momentum. This missing transverse momentum is the expected signature of collider DM production [17]. A challenge for this approach are neutrinos - these particles also escape the experiment undetected and are a known source of missing transverse momentum.

But, while WIMPs are strongly motivated candidates for a that DM particle, that does not mean there aren't different, theoretically viable candidates: for example, axions are light hypothetical particles that arise naturally in the Peccei-Quinn Mechanism, a possible solution for the strong CP problem of the SM. SuperWIMPs and Hidden Dark Matter are candidates that exploit the WIMP miracle without requiring weak interactions. Every candidate presents interesting features and, until experimental exclusion, they present viable alternatives to the hypothesis of the WIMPs [10].

1.2 LZ Detector Design

As stated above, the LZ [18] is a direct detection experiment. A schematic of the experiment is shown on Figure 2. The main detector is a Time-Projection Chamber, a cylindrical (145.6 cm of height, 145.6cm of diameter) container filled with 10 tons of gaseous and liquid Xenon (Xe), and covered at the top and bottom with arrays of Photo-Multiplier Tubes (PMTs) (totalling 494 PMTs in the TPC). The

location and configuration of LZ is set up to minimize external backgrounds on the TPC: it is built 1.5 km underground - or 4300 meters of water equivalent (mwe) - at the Sanford Underground Research Facility (SURF); and the TPC is surrounded by a dual veto system to tag background events that otherwise could be mistaken for DM signal. This system is composed by the Xenon Skin Detector and the Outer Detector (OD), filled with 17.3 tonnes of Gadolinium-loaded Liquid Scintillator (GdLS), equipped with 131 and 120 PMTs, respectively. The LZ is built in a cylindrical configuration centered on the active zone of the TPC.

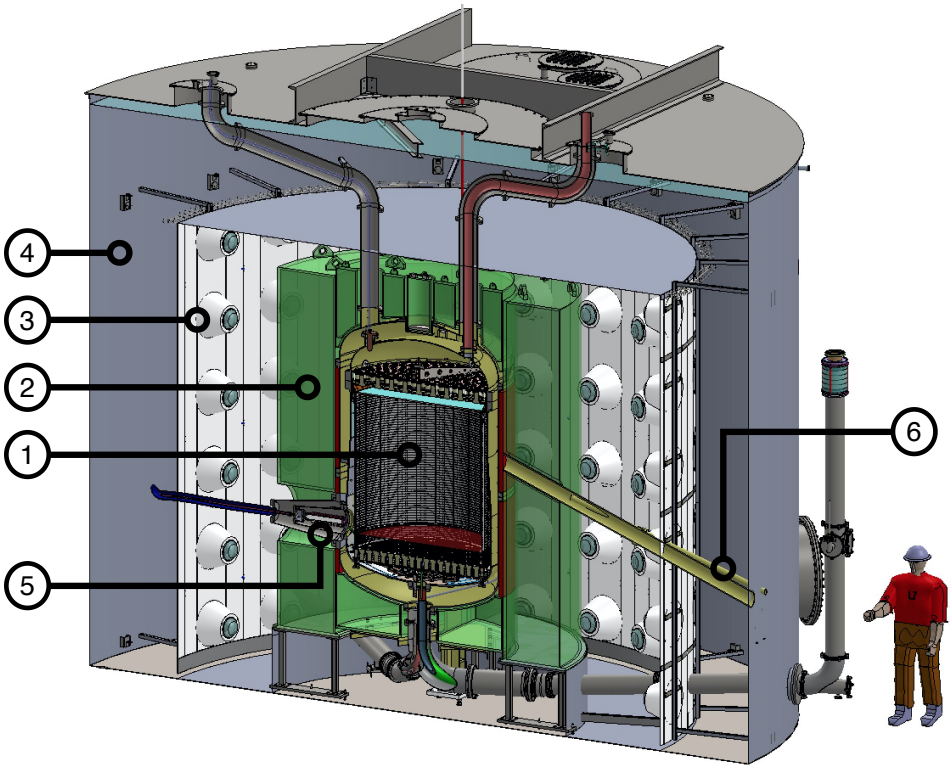


Figure 2: Cross section of the LZ experiment, exposing its structure. The main component of the detector is the the liquid xenon TPC (1), monitored by two arrays of PMTs (top and bottom). The TPC itself is inside the GdLS Outer Detector (2). A suite of 8" PMTs (3) monitors the GdLS in the water tank (4), which acts as a shield for the detector. The cathode high voltage connection is made horizontally at the lower left (5). The pitched conduit on the right (6) allows for neutron calibration sources. Image from [18].

1.2.1 Dual Phase Xenon Time-Projection Chamber

Time-Projection Chambers were idealized as a particle detector that would allow a precise reconstruction of event locations: by aligning the magnetic and electric drift fields, the ionization electrons caused by the event (for example, a decay) would not suffer significant diffusion along their path along the target. A detector positioned at the end cap of the chamber (normal to the drift field) would then be able to observe the electrons and give a precise measurement of the x-y event location. Knowing the

drift velocity of the electrons inside the chamber, the z position could be known if the time of the event was known [19, 20].

The Dual Phase TPC is an adaptation that not only allows an accurate 3D reconstruction of the event location inside the chamber, while also improving its energy reconstruction of the event. The chamber is filled with liquid (making it a denser target, improving the number of nuclei in the chamber and improving the event rate), with a layer of gas on top. Interactions with the liquid target may cause the atom to excite and subsequently to de-excite, emitting photons. This emission of photons at the time of the event is known as primary scintillation (S1). On the other hand, the interaction may cause ionization, creating free electrons in the chamber. The free electrons that do not recombine are then guided along the electric field that permeates the chamber towards the xenon liquid-gas interface. In the phase transition zone, a stronger electric field extracts the electrons from the liquid to the gas, producing secondary scintillation that can be detected as the electroluminescence signal (S2). x-y location can be inferred from the activated zone in the top PMT array; the time difference between the detection of the scintillation and the detection of the transition radiation is known as *drift time* and allows for z position reconstruction. A simple representation of this process can be seen in Figure 3.

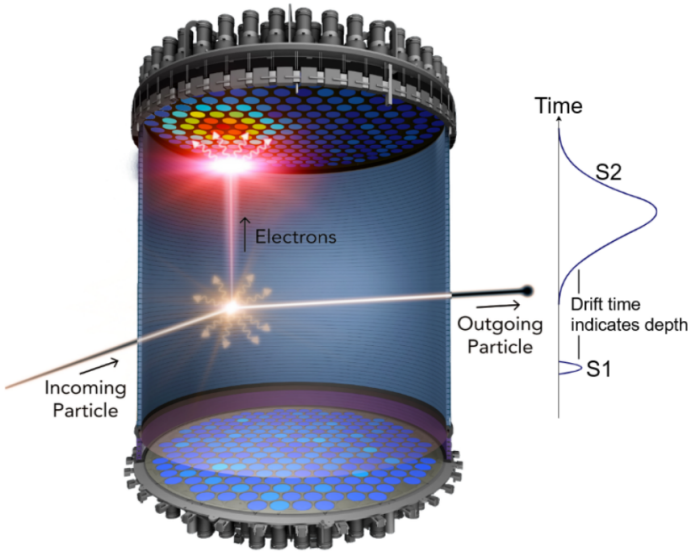


Figure 3: Representation of an LZ event. The incoming particle collides with a Xe atom (it can be a NR or a ER). The scintillation photons are detected first, at the approximate time of the event. The S2 signal is detected after the electrons from the ionization drift across the chamber. As electron drift upwards, x-y location can be inferred from the PMTs that detect the signal. The time difference between the S1 and the S2 signals allows z reconstruction, thus achieving full 3D position reconstruction.

The number of PMTs activated and the amplitude of the signal in all activated PMTs are used to reconstruct the energy of the event. The comparison between the energies of the S1 and the following S2 can be used to infer if the event is a Nuclear Recoil or from an Electric Recoil (ER) - an elastic collision

with the electronic cloud of the atom.

A schematic for the LZ Xenon Detector System (XDS) is shown in Figure 4. The LZ TPC is the main component of this system, containing 7 active tonnes (5.6 fiducial) of Liquid Xenon (LXe) that act as the target for incoming particles. The active volume occupies a cylinder of roughly 1,5 m height and diameter, and is topped by a 8 mm layer of gaseous Xenon. Another 2 tons of LXe surround the field cage, acting as protection against external radiation, forming the Xenon Skin.

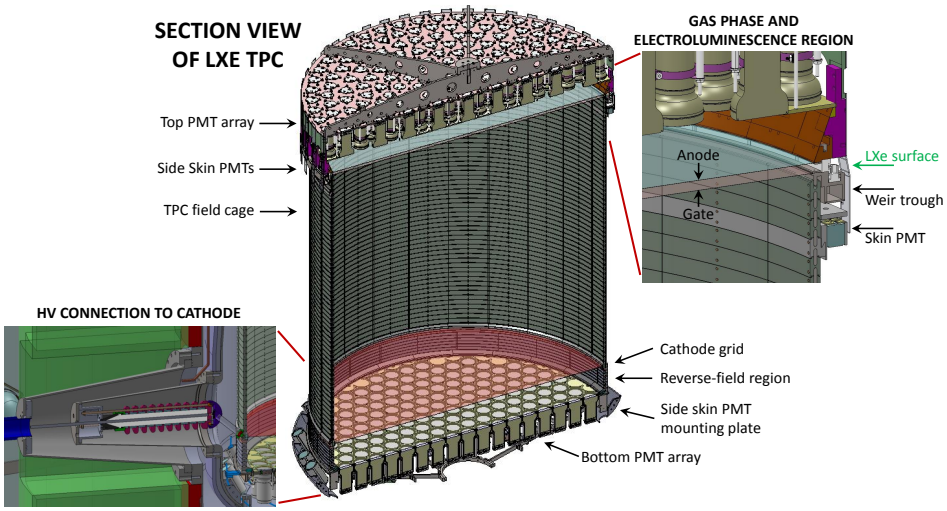
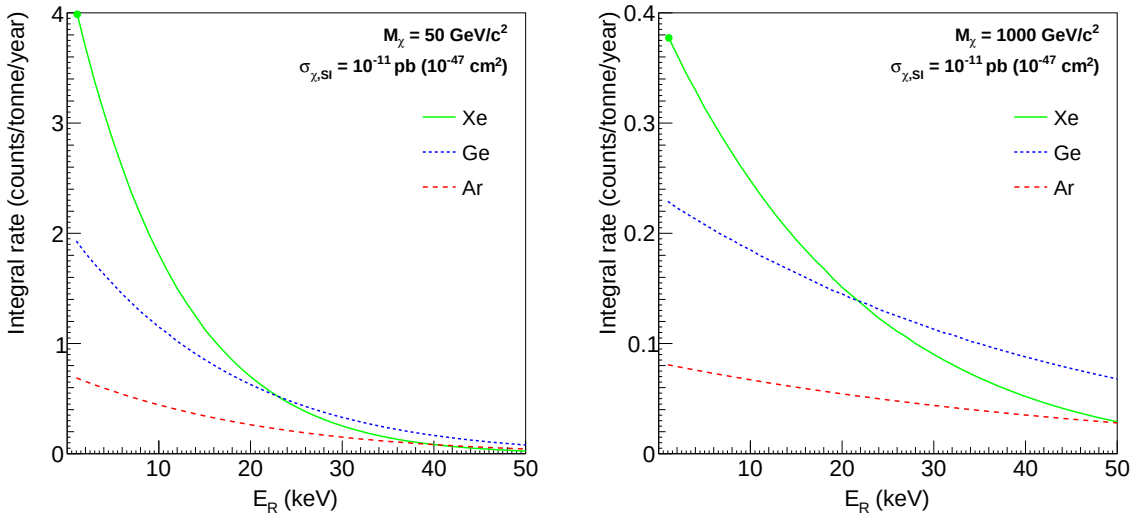


Figure 4: A schematic for the Xenon Detector System (XDS). The XDS is composed of the Two-Phase TPC and the Xe Skin Detector. The TPC itself is composed by the field cage that surrounds the active volume of xenon, the top and bottom PMT arrays, and the necessary machinery to create the electric fields (the grids and the power source connection). The field cage is surrounded by LXe, that acts as a radiation shield. This (non-active) volume is observed by the Skin PMTs, forming an anti-coincidence detector, the Xe Skin Detector. Image from [21].

Noble gases (and noble liquids) have long been used in rare event searches [22, 23]. Their scintillation properties, combined with their inertness (allowing free electrons to drift long distances) and some convenience factors (availability and relatively easy purification) makes them natural options for the TPC target. Xenon, in particular, has an high atomic number (Z=54), making it a material with high radiation stoppage power. In addition, its atomic mass is a good match to a $\approx$100 GeV/c^2 WIMP [24, 25]. For low energy events (10 to 100 KeV), the xenon scintillation yield is higher than other noble elements like argon, improving the sensitivity of the detector to events in this energy range (Figure 5).

For proper operation, the TPC is divided into 3 different regions with different electric fields, whose limits are defined by 4 electrode grids: the anode, the cathode, the gate and the bottom grid. Between the cathode and the gate there is the Drift Region, permeated by the drift field, and between the gate and the anode there is the Extraction Region, permeated by the extraction field. The voltage values can be tweaked between science runs: in the first science run, the cathode and the gate were operated to



(a) $m_{\text{WIMP}} = 50$ GeV/c^2 (b) $m_{\text{WIMP}} = 1$ TeV/c^2

Figure 5: Integrated rate above threshold per tonne per year of exposure for WIMP elastic scattering on Xe, Ge, and Ar targets for 50 GeV/c2 and 1 TeV/c2 WIMP masses and a 10^{-47} cm^2 interaction cross section per nucleon. The green dot on the upper right corner of the plots marks the 1.1 KeV NR WIMP-search threshold in LUX, one of the two LZ parent experiments. Xenon high scintillation yield allows the detector to be more sensible to low energy deposition. Plots from [21].

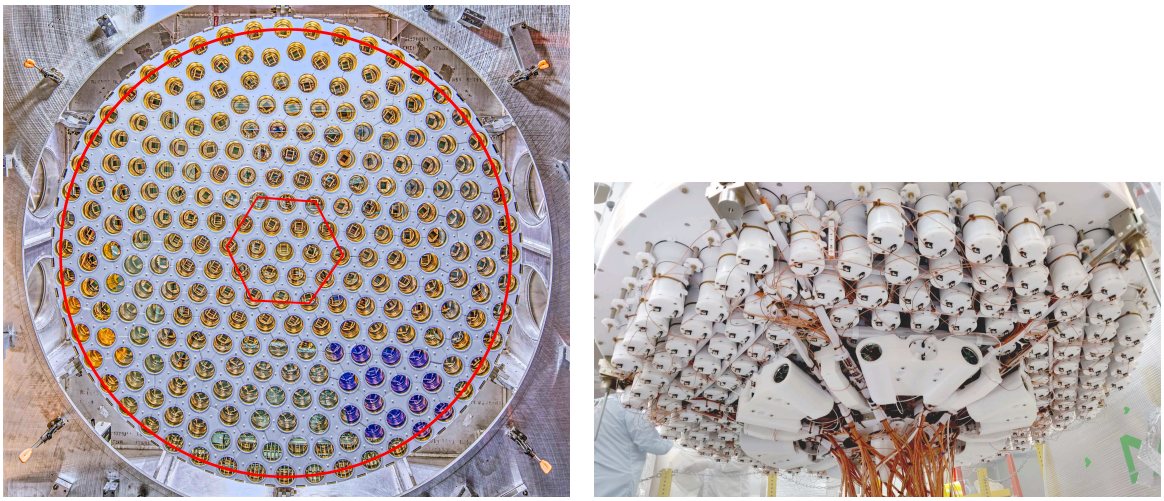
generate a 193 V/cm drift field; the anode was operated so that the extraction regions was permeated by a 7.3 kV/cm field [26]. In the third science run, the TPC was operated with a 97 V/cm drift field and a 3.4 kV/cm [27]. The cathode and the bottom grid limit the Reverse Field Zone, with an electric field in the opposite direction from the drift field. Its function is to slowly grade down the high voltage of the cathode electrode to the ground. The bottom grid protects the bottom PMT array from the 3.5 kV/cm field [28].

1.2.1.1 PMTs

The PMTs serve as the primary sensors within the detector. Photons from the S1 scintillation are predominantly collected by the bottom PMT array due to total internal reflection at the Xenon liquid-gas interface. S2 photons, on the other hand, are primarily detected by the top PMT array; the bottom array capturing a uniform fraction of approximately one-third of the S2 light. Since S2 photons are created very close to the top array, the individual PMT response in the array can be used to determine the $x - y$ of the event, while the event energy can be reconstructed using a combination of both arrays. The exception are very high energy events that can saturate the top array PMTs, in which case the energy of the event can still be reconstructed using the response of the bottom array PMTs.

3-inch Hamamatsu R11410-22 PMTs were chosen due to their high quantum efficiency for photons

in the Vacuum Ultraviolet (VUV) region (100 nm $< \lambda_\phi < 200$ nm), expected from the Xenon scintillation (a spectrum of 14 nm centered around 178 nm [29]), their good performance on the low temperatures demanded by liquid xenon, and low radiological background [30]. The arrays are shown in Figure 6.



(a) Top PMT Array (b) Bottom PMT Array

Figure 6: Arrays of R11410–22 PMTs viewing the TPC. On the left, the top PMT array photographed before assembling. To highlight the hybrid layout, a circumference in the periphery of the array and a hexagon in the center are drawn on top of the photograph. On the right, a photograph of the bottom array PMTs showcasing its hexagonal setup. The photograph also shows 18 2-inch dome PMTs, which are part of the skin veto system. For both arrays, most surfaces are covered in PTFE to maximize light collection. Images from [18].

The layout of the 253 PMTs in the array - an hexagonal pattern at the center slowly transitioning into a circumference at the perimeter, as seen in Figure 6a - minimizes inaccuracy in the position reconstruction. This is specially important for low energy events near the wall: radon deposited in the field cage wall may decay and the signal may mimic NRs. Therefore, mis-reconstructions could push the events further into the active zone, becoming a significant background.

The field cage and the surface of PMT arrays are made of high reflectivity PTFE to maximize the light reaching the PMT arrays. Due to the reflectivity of the LXe surface at the top of the TPC, the S1 light is mainly collected at the bottom array. The bottom array consists of 241 PMTs distributed in a hexagonal array (Figure 6b) and is designed to maximize photocathode coverage.

1.2.1.2 Nuclear vs Electronic Recoils

NRs occur when a particle interacts with a Xenon nucleus, emitting radiation and producing free electrons. WIMPs, but also neutrons, can cause this type of scatter. ERs are produced when a particle interacts with the electronic cloud of the atom, also emitting radiation and producing free electrons.

Then, both types of recoil can produce an S1/S2 signature. ERs are the most common of the two (they can be caused by natural radioactivity, beta decay of radioimpurities in the Xe target, etc.) and form an important background source. It is imperative to discriminate between these types of scatter - the most clear path to discriminate between those two types of events is to compare the primary scintillation with the ionization charge.

Figure 7 shows the distribution of ERs - produced by a calibration tritium source - and of NRs - produced by a deuterium-deuterium (DD) calibration source. The data shows that the S1c/S2c ratio from the events form two bands: the lower band, with higher S1c/S2c ratio formed by the NR events; and the upper band, formed by ER events, with lower S1c/S2c. Therefore, a precise energy reconstruction is essential to discriminate between both types of recoils.

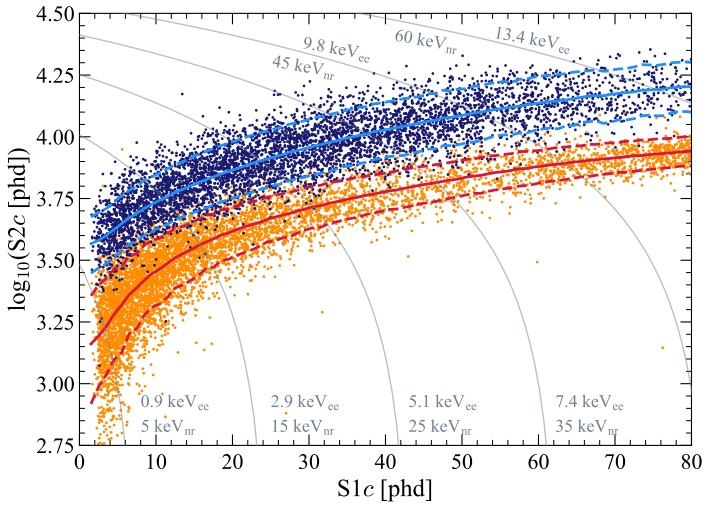


Figure 7: Data obtained from two different calibration methods: a tritium source (dark blue points, 5343 events), that produces ERs, and a deuterium-deuterium source, that produces NRs (orange points, 6324 events). Solid lines represent the median of the simulated distributions; dotted lines represent the 10% and 90% quantiles. The gray solid lines show contours of constant electron-equivalent energy (KeV_{ee}) and nuclear recoil energy (KeV_{nr}). The axis measure the area of the S1/S2 of the event. The delineating of two bands is clear, and reinforces the notion that the ratio between S1 and S2 energies is critical to discriminate between NRs and ERs. The denomination S1c and S2c indicates that the signal was normalized to the detector geometric center and to the radial center of the area the S2 was detected, respectively. Image from [26].

1.2.2 Xenon Veto System

The WIMP-Xenon nuclei scatter is a rare event; as seen in Figure 5, for liquid xenon, we would expect about 4 events per tonne per year for a 50 GeV/c^2 WIMP (even less for higher masses). Therefore, reducing the background to the maximum amount possible is critical. This can be done in two different

ways: by shielding the active volume from external events and/or by identifying those external events so they can be properly identified and removed in the data analysis.

Between the field cage and the inner cryostatic vessel sits 2 tonnes of Xenon, forming the Xenon Skin (XeSkin). The main function of this xenon volume is to act as an insulator between the electronically biased TPC and the grounded cryostatic vessel. But, given the high stoppage power of LXe, the Skin could contribute to the irreducible background of the detector: gamma radiation (or even neutrons) could cause a scatter in the active zone of the TPC and then get absorbed in the XeSkin, stopping the Outer Detector from vetoing the event. To eliminate this factor, the XeSkin is equipped with PMTs on the side skin and on the top and bottom of the TPC cylinder. This transforms the XeSkin into a veto detector, detecting scintillation in the skin region.

Neutrons are electronically neutral and their mass is comparable to that of light WIMPs. ($m_{neutron} = 0.939$ GeV/c^2). Additionaly, neutrons originated from muon interactions can reach the TPC at high velocities, depositing large amounts of energy in the Xe target. Therefore, neutrons may cause a single NRs and then escape the TPC, mimicking the expected signature of a WIMP scatter, causing an important source of NR background. Therefore, tagging neutrons is an important task to be able to identify these events.

To this effect, the XDS is surrounded by the Gadolinium-loaded Liquid Scintillator (GdLS), ten acrylic tanks filled Gadolinium (Gd) doped linear-alkylbenzene (LAB) solvent. The Gadolinium doping increases neutron tagging efficiency and reduces capture time compared to non-doped LAB - reducing the time interval where a TPC scatter can be identified with the outer detection of a neutron. When the neutrons are captured by Gd, an average of 4.7 γ are emitted (the energy depends on the Gd isotope: 7.9 MeV for ^{155}Gd and 8.5 MeV for ^{157}Gd) - this happens to 90% of captured neutrons. The other 10% are captured by hydrogen in the LAB, releasing a single 2.2 MeV γ [18]. The tanks are observed by 120 8 inch Hamamatsu R5912 PMTs.

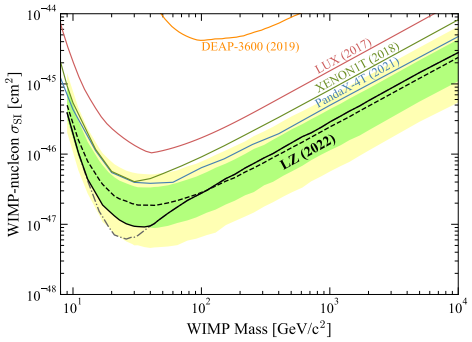
The combination of the XeSkin Detector and the OD forms a powerful veto system, focusing on reducing neutron and gamma rays background (the main sources other than neutrinos) and subsequently improving the low-background active xenon region volume. Additionally, the whole detector is submerged inside a water tank, providing an extra shielding against external sources (for example, radiation originated from the cavern walls). Every component is purified to an high degree, to minimize irreducible internal background sources coming from those impurities.

Besides the detector components, the LZ also has cryogenic and Xenon renovation and purification systems, as well as a complex electronic hardware. Their description falls outside the necessary context for this thesis, but a compact description can be found here [18].

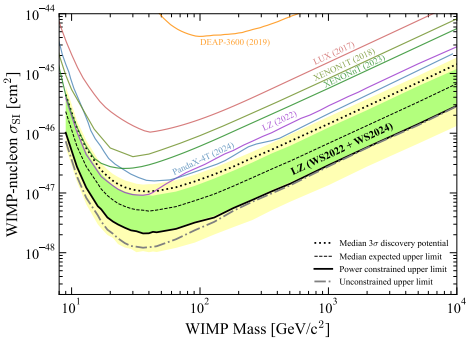
1.2.3 Dark Matter search results from the LZ experiment

LZ first science run started on December 23rd, 2021 and ended on April 18th, 2022. With a brief pause for calibrations between January 17th and 26th, 2022, totaling 109 live days. Due to the Data Acquisition dead times, periods with anomalous trigger rates and analysis quality cuts, the total live time, where a WIMP scatter could be reconstructed, was approximately 60 live days. After the run end, calibrations ran until May 11th and the first results were published in July 7th [26].

The most recent WIMP search results published by the collaboration (at the time of writing) were released in September 2024 and features data from an additional 220.0 live days - from March 27th, 2023 to April 1st, 2024 [27]. The full analysis incorporates data from the the initial 60 live days and the more recent 220 live days (with the changes in the detector parameters accounted for in the analysis). These results are compatible with the background-only hypothesis, and the expected number of WIMP scatter interactions observed in the data is zero. Nevertheless, the data was enough to improve on the upper limits of the interaction cross section previously imposed by other experiments (such as PandaX or XENONnT, mentioned in the begging of the chapter) as it can be observed on Figure 8. The additional 220 live days allowed for an even deeper probe into the parameter space, as expected.



(a) First results from LZ. Image from [26].



(b) Most recent results from LZ. Image from [27].

Figure 8: Limit on the WIMP-nucleon spin-independent (SI) interaction cross section as a function of WIMP mass. The solid lines represent 90% confidence limits for the WIMP cross section against WIMP mass for different experiments - in black, the LZ limit. The green and yellow bands are the 1σ and 2σ sensitivity bands for the LZ results, and the dotted line shows the median of the sensitivity projection.

The LZ will keep alternating between periods of data collection and periods of calibration. Calibration is essential to understand if the detector conditions are stable from the beginning to the end of a data collection period. Future data collected can be combined with this data, further improving these results and probe deeper into the parameter space, restricting WIMP theories even more - see Chapter 4 for a brief discussion of some calibration sources utilized in the LZ.

1.3 Non Dark Matter Searches

Even though the experiment is geared towards DM search, the full scope of scientific possibilities go further beyond the search for the WIMP. As stated before, the ability to discriminate well between Nuclear Recoil (NR)s and ERs is one of the most important features of the detector. Consequently, LZ is not only capable of reconstructing NRs, but ERs as well, allowing for investigation of phenomena that can be studied from this process. One possible venue for investigation comes the double beta decay, either from the SM-allowed two neutrino double beta ($2\nu\beta\beta$) decay (rare, but already observed), or from the not yet observed (and SM-forbidden) neutrinoless double beta ($0\nu\beta\beta$).

The two neutrino double beta decay is a rare decay that can occur on even-even nuclei. This process results in the emission of two electrons and two antineutrinos, clearly conserving the lepton number:

$$(A, Z) \longrightarrow (A, Z + 2) + 2e^{-} + 2\overline{\nu}_e \tag{1}$$

^{136}Xe nuclei is particularly relevant for this study - the isotope constitutes 8.9% of the natural Xenon abundance [31] and is strongly present in the TPC. This isotope double beta decays with an half-life of $2.23 \pm 0.03^{\text{(stat)}} \pm 0.07^{\text{(syst)}} \times 10^{21}$ years [32] and a Q-value of 2457.83 $\pm$ 0.37 KeV [33]. ^{134}Xe is also expected to double beta decay, but the decay is not experimentally confirmed - LZ has the potential to search for this process [34].

The study of this decay is an important tool to probe the nuclear structure of the atoms and, in particular, to find the values of nuclear matrix elements (NMEs). The decay of two isotopes can be studied to reduce the uncertainty of these values even further [35].

This process is allowed within the framework of the SMPP, conserving the electronic lepton number. The other channel for this process, on the other hand, is SM-forbidden - the neutrinoless double beta:

$$(A, Z) \longrightarrow (A, Z + 2) + 2e^{-} \tag{2}$$

The signature of this process would very similar to the $2\nu\beta\beta$ process: a mono-energetic peak with similar Q-value, but with different energy spectrums. To discriminate between these two processes, it would be necessary an highly sensible experiment, with high energy resolution and strong background rejection; these are similar requirements to the WIMP search. LZ is projected to be sensible to this process for an half-life of 1.06×10^{26} for 1000 live days of data collection in the WIMP search conditions [33], a value with similar magnitude with the most recent results [36].

The study of this process is one of the main channels to search of lepton number violation and to try to understand the nature of the neutrinos. If observed, it would imply that the neutrino is its own antiparticle, establishing the neutrino as a Majorana particle [37]. On the other hand, the simultaneous

search for phenomena with widely different energy ranges poses problems for the data analysis that have to be overcome in order to properly explore the full potential of the LZ experiment.

1.4 Summary

In this chapter, a brief introduction of DM (Section 1.1) gave way to a concise description of the LUX-ZEPLIN Experiment Design (Section 1.2), contextualizing the choices made in the construction of its components. Special attention was given to the TPC (Subsection 1.2.1), the fundamental technology that underpins the experiment. At the end of the chapter, the most recent results of the LZ experiment are shown (Subsection 1.2.3), as well as some alternative investigation venues allowed by the the design and technology that make the LZ Experiment (Section 1.3).

Chapter 2

The Anatomy of an LZ Event

The LZ data is the digitized response from all PMTs in the TPC, Skin and OD. Since the experiment is built to minimize the background of a very rare signal, it is not expected that at every moment there will be relevant data to be analysed. Therefore, in order to not continuously record electronic noise sparsed by detector events, the data is collected in a process called Pulse Only Digitization (POD) - when the PMT response crosses a certain (user-defined) threshold, the data is recorded until the response falls again below the treshold. In addition, the 32 samples before the response crosses the treshold and the 32 samples after it crosses it again are also recorded. The resulting waveform takes its name from the process that creates it: it is a POD. If the threshold is crossed again within the 32 post-samples, the two resulting PODs are merged [38]. To select which events to record, LZ has a trigger system: if it is activated, data is recorded from 2 ms before the trigger to 3 ms after the trigger (these values can also be adjusted by the user, if needed or preferable). The PODs in this time interval define the raw waveform of the LZ event.

2.1 LZap - LZ Analysis Program

The LZ Analysis Program (LZap) is the series of algorithms that takes the raw waveforms and encodes it into physics-ready quantities known as Reduced Quantities (RQs). RQs can refer to a specific pulse or an event and range from characterization quantities (for example, the amplitude and length of a pulse) to identification variables (the pulse classification or the type of scatter that originated the event).

LZ data is segmented into different categories, depending on the physical detector in which the data is generated and the gain mode of the detector. Therefore, the LZap data is divided into 6 structures - TPC (HG, LG and MG), OD (HG and LG) and Skin. The TPC Mix Gain (MG) is not obtained directly from the detector response, but is constructed from the TPCHG and TPCLG segments. The goal is to built a response with the sensibility of the high gain response while using low gain mode to correct saturation

in HG data and to adjust for the non-linearity in the PMT response.

The first step in the analysis is to identify the individual pulses in each POD and then characterize and classify them. The set of modules that accomplish these tasks is known as PhotonDetection. The next step, the InteractionFinder, takes this pulse-level information and uses it to analyze the event in its whole.

2.1.1 PhotonDetection

A schematic of the LZap *PhotonDetection* module is depicted in Figure 9.

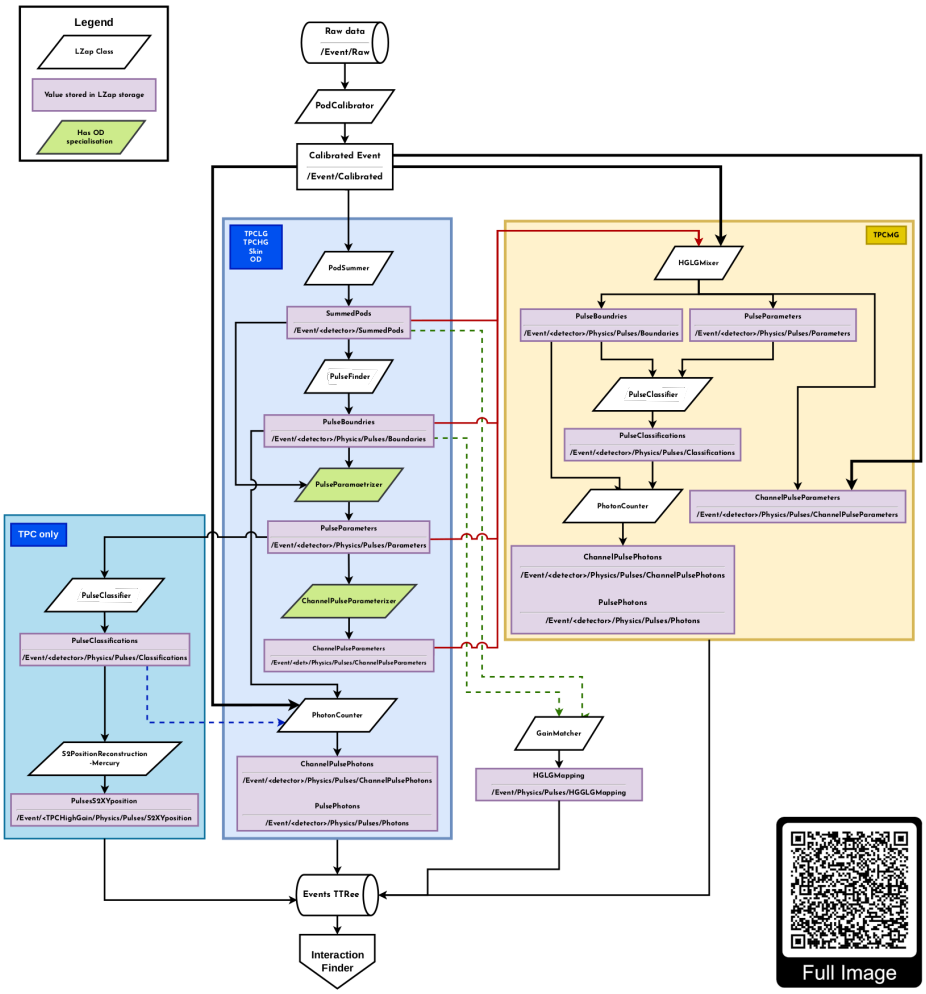


Figure 9: Simple *PhotonDetection* module schematic. The raw data is calibrated and processed in 6 different channels: three channels from the TPC data (high-gain, low-gain and mixed-gain), another two from the OD (not explicit in the schematic), and one from the XeSkin. The individual pulses that compose the PODs are identified, characterized through RQs and classified. This processed data flows to the next step of the analysis - the *InteractionFinder*.

There are nine modules (or classes) within the *PhotonDetection* module: *PODCalibrator*, *HGLGMixer*, *PODSummer*, *PulseFinder*, *PulseParametizer*, *ChannelPulseParametizer*, *PhotonCounter*, *PulseClassifier*, *GainMatcher*, and *S2PositionReconstruction*. Some of these modules are common to all LZap channels (e.g., the *PhotonCounter*), while others are channel-specific (e.g., the *HGLGMixer*).

The general data flow for the TPC channels is the following: the raw data is processed through the *PODCalibrator* - converts the digitized waveform to units of photons detected (phd), calibrates channel timing and event baseline, in addition to adjusting the event taking into account the gain factors for HG and LG; the data from all channels for every gain mode available are summed into one waveform for event (*PODSummer*) - overlapped PODs are summed into only one POD; *PulseFinder* uses two Difference of Gaussian (DoG) filters to identify single pulse structures in the PODs; these pulse structures are then characterized by the *PulseParametizer* and the *ChannelPulseParametizer* modules, by computing pulse and channel level RQs - an extensive RQ list ; finally, the pulses are classified using the *PulseClassifier* module. The end point of this process is shown in Figures 10 and 11: an event is a sequence of 6 types of pulses: S1 and S2; Single Photoelectron (SPE) and Multiple Photoelectron (MPE); Single Electron (SE); and Other.

Since LZ is built to detect and properly characterize nuclear and electronic recoil signal, discriminating between S1, S2 and these other pulses structures is extremely important - not only. In some events, such as the one represented in Figure 10, these pulses are clear and this task seems straightforward to achieve.

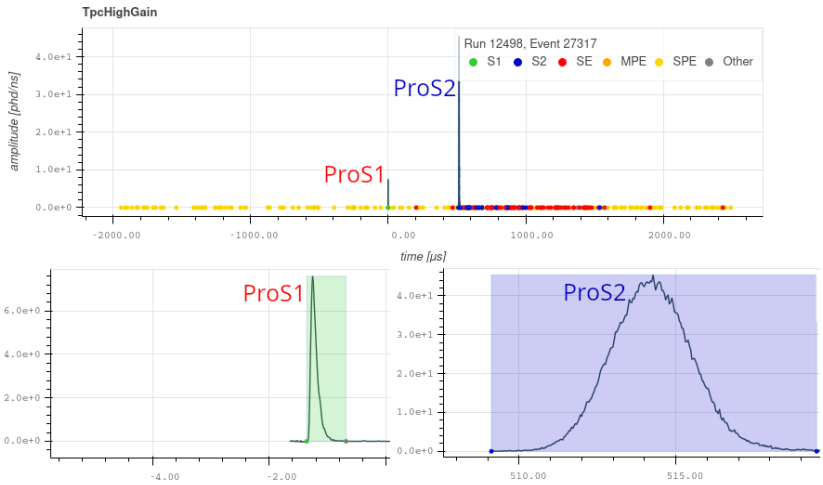


Figure 10: ^{131m}Xe calibration source event. Image from LZ Event Viewer.

In this event, the S1 pulse has a clear fast rise and an exponential decay as expected from a primary scintillation signal. Also as expected, the S2 has an almost symmetrical rise and decay. In addition, both pulses are distinctly separated from the other pulses in their PODs. On the other hand, for the event in Figure 11, the amplitude of every pulse is much smaller, and the pulse shape is less clear; in the case

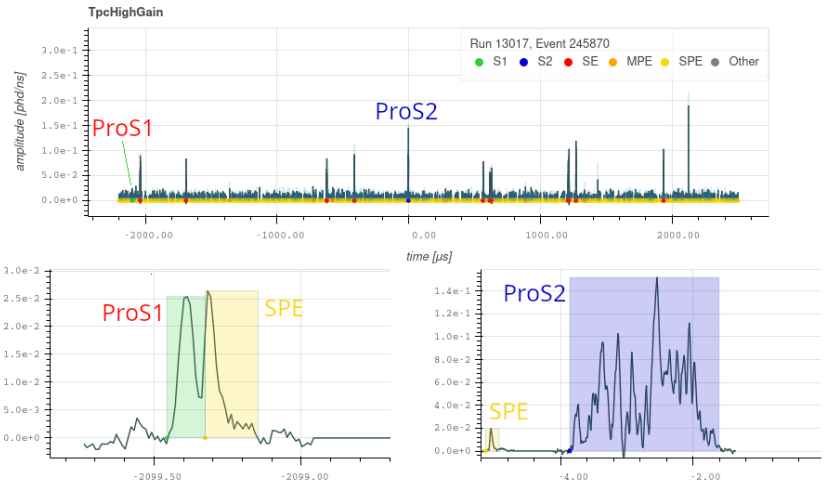


Figure 11: AmLi calibration source event. Image from LZ Event Viewer.

of the S1 pulse, the separation (and classification) is not clear. This event is an example of some of the challenges of pulse classification.

The Heuristics Algorithm for Discrimination of Event Substructures (HADES) is a decision tree algorithm based on the pulse level RQs obtained from the *PulseParametizer* module. Accurately discriminating S1 and S2 signals from unrelated electrophoton or electron signal is challenging due to overlapping features in the parameter space, especially at low energy, where light yields are reduced. The only criterium to discriminate between MPE is the number of PMTs that detected light during the duration of the pulse - a minimum of three PMT is necessary to classify as a S1 - so this distinction can be problematic at these low energy limits. A similar problem affects S2 classification, where the coincidence of multiple uncorrelated SE may be misclassified as an S2 [39].

Additionally, there are a number of cases where some structures, called pathological pulses, are wrongly classified as S1 and S2. These pulses may originate from spurious signals like PMT dark counts, afterpulsing, baseline noise, accidental coincidences, or detector artifacts such as split or merged pulses introduced during the waveform processing [39]. Nevertheless, HADES performs very well for every class of pulses. For S2 classification, the recall of the algorithm is close to 100%.

At the end of the *PhotonDetection* section of the LZap, there can be S1 and S2-like pulses that are classified as true S1 and S2 events. The distinction between these pulses and true S1 and S2 - henceforth names Prominent S1 (ProS1) and Prominent S2 (ProS2) - is a part of the following section of the analysis.

2.1.2 InteractionDetection

Following the *PhotonDetection* module, the *InteractionDetection* module receives the processed waveform and the characterized and classification of the pulses in order to obtain event-level Reduced Quantity (RQ)s. The first step is to determine which S1 and S2 pulses are prominent - there can be more than one of each. This LZap module is known as *PulseHighlighter* and it is described in more detail in Section 2.2.

Since this algorithm does not consider temporal relations between pulses, an additional selection is made: an interaction in the active region of the TPC will produce a prompt S1 signal followed by a delayed S2 signal (as seen in Section 1.2.1). Therefore, when determining the scatter topology of an event, LZap looks for ordered ProS1-ProS2, and it discards ProS2 before the first ProS1 and ProS1 after the last ProS2. After the determination of the ProS1s and ProS2s by the *PulseHighlighter*, the task of identifying the scatter topology falls on the *InteractionFinder* - a decision tree algorithm schematized in Figure 12.

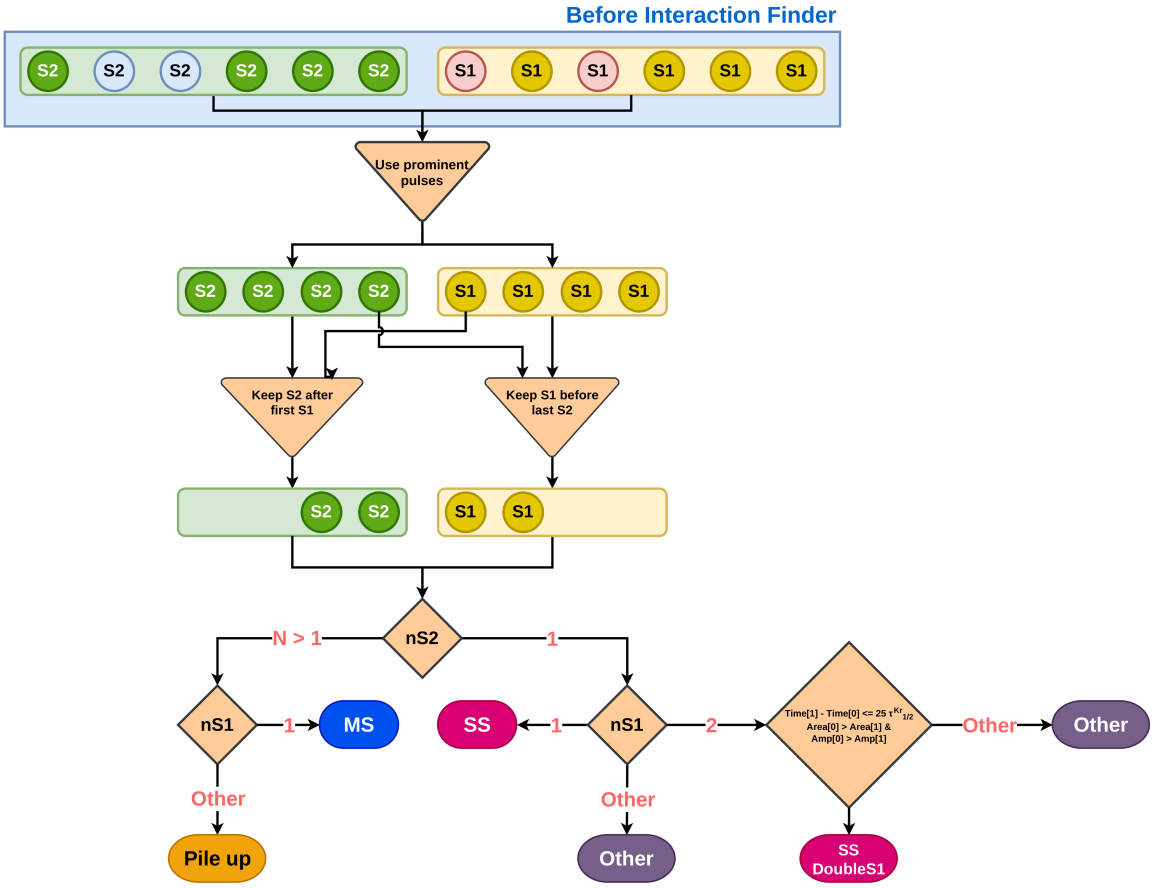


Figure 12: Simple InteractionFinder schematic.

The determination of the scatter type is essential for background rejection: a WIMP would probably only scatter one time in the TPC, producing a single ProS1 followed by a single ProS2 - this is a SS event; on the other hand, similiar energy Electric Recoil (ER) events can sometimes cause a Multiple Scatter (MS). These Multiple Scatter events are characterized by a single ProS1 followed by two or more ProS2. If an event has more than one ProS1 and more than one ProS2, the event is considered a pile-up event that can be discarded in the search for a WIMP (but are nevertheless useful in the analysis for radon tagging); events with only ProS2 and exactly two ProS1 that obey certain conditions - the first ProS1 being bigger than the second, and their temporal distance being within 25 half-time decay of the ^{85m2}Kr - are identified with the Kr decay and therefore can be excluded as background; events that do not fit these archetypes can be caused by coincidences (unrelated S1 and S2) or misclassifications at the *PulseFinder*, *PulseClassification* or *PulseHighlighter* level [40].

Data from the OD and Skin are also analyzed in order to determine of there are coincident interaction between the TPC and these "veto" detectors, and thus tag those events as BG. Parallel to this data flow, there are modules to perform the event reconstruction, positional and energetic, in order to fully characterize the event. After this process, the events are ready for the physics analysis.

2.2 Pulse Highlighter

As stated in the previous section, the *PulseHighlighter* is the module responsible for discriminating between prominent or non-prominent S1s/S2s, an important distiction that will impact the performance of the *InteractionFinder* module, responsible for scatter classification. It is an heuristic decision tree based on RQ space cuts. The scheme is depicted in Figure 13 for S2 pulses. To discriminate between NonProS1 and ProS1, the logic is very similar to the one presented in the schematic.

This module receives the pulse information from the *PhotonDetection* and selects the S2 pulses as classified by the *PulseClassifier* module. For each event, it first determines that the largest S2 in the event is prominent. The immediate consequence is that any event with only one HADES S2 has always one ProS2. If there are multiple HADES S2, they are separated based on their relation to the largest S2 in the event - in particular, the height/width ratio of the nth largest S2 against the area ratio of the nth largest $S2$ against the largest S2. The cut is shown in Figure 14.

The pulses that pass this test are not definitively ProS2; in the same vein, the S2s that fail the test are not scrapped. In both cases, the pulses are tested again. In the case of the temporarily prominent, if they pass this next test, they are definitely ProS2; in the case of the temporarily non prominent, if they fail this test, they are definitively NonProS2. The test is a comparison between the area of the largest S2 of the event and the area ratio between the pulse and the largest S2 (Figure 15).

Lastly, a third test is emplyed to the pulses that do not yet have definitive classifications, to check

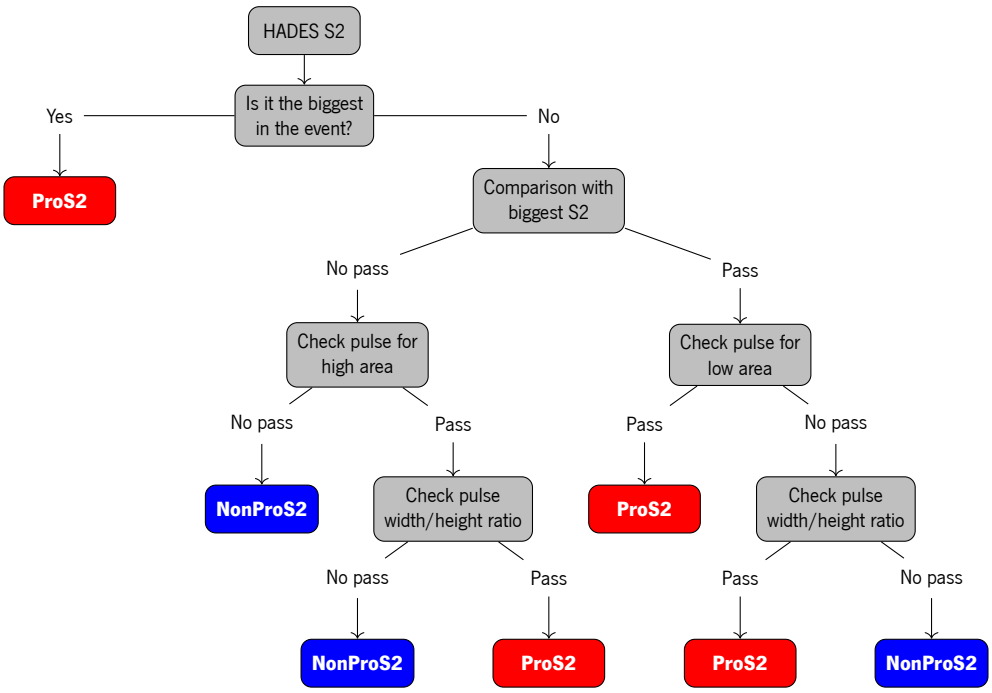


Figure 13: *PulseHighlighter* scheme for S2 pulses. The logic is similar for S1 pulses - changes occur in the speficic values for cuts.

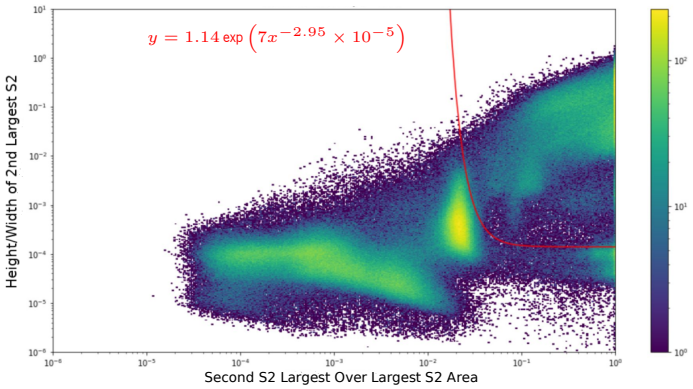
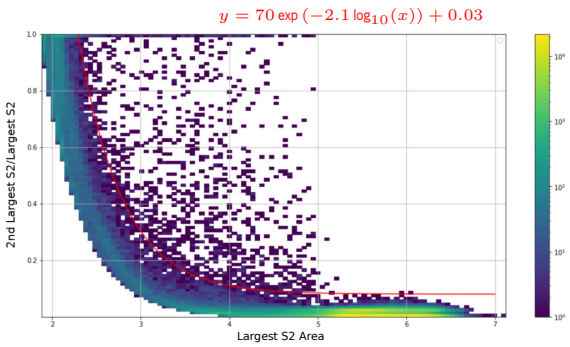


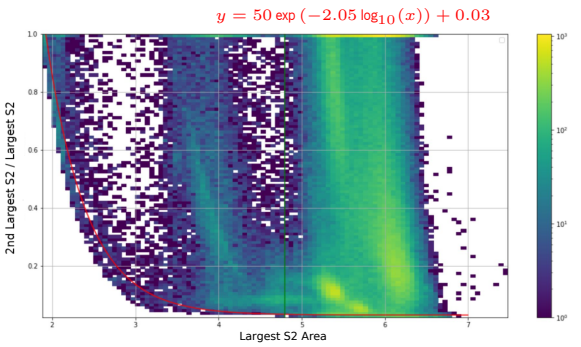
Figure 14: First parameter space discrimination in the *PulseHighlighter* - in this case, for the 2^{nd} largest pulses in their own event. This is done to nth largest S2. All S2 to the right of the red line pass this test; to the left, they fail this test. S2s from CH3T events. The red equation is the parametrization of the red curve (obtained from LZap source code).

their height to width ratio against the area of the largest S2 (Figure 16). Most pulses keep their temporary classification.

The *PulseHighlighter* algorithm utilizes all S2s in an event to try to contextualize these structures. This approach does not take into account temporal distances between S2 pulses; it also does not take into account any information about the S1 (or any other pulse structure) in the event. In addition, the

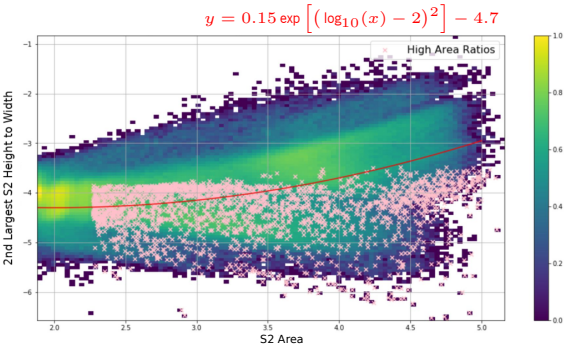


(a) Temporary NonProS2. S2s to the left of the red line are definitively NonProS2.

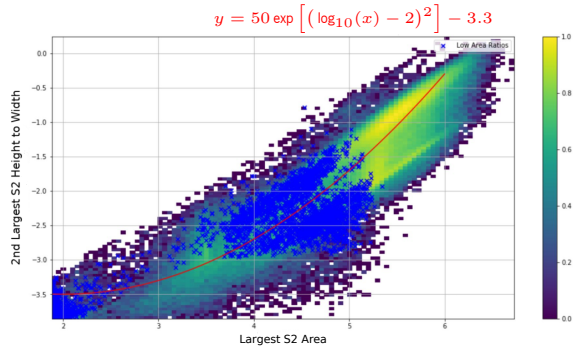


(b) Temporary ProS2. S2s to the right of the red line are definitively ProS2.

Figure 15: Second test for S2 pulses. The red equation is the parametrization of the red curve.



(a) Temporary NonProS2. S2s above the red line are definitively NonProS2.



(b) Temporary ProS2. S2s above the red line are definitively ProS2.

Figure 16: Third test for S2 pulses. The crosses represent the pulses whose classification is determined by this test. The red equation is the parametrization of the red curve.

characterization of the pulses is limited to the RQs associated with the pulse area, width and height. The goal of part of the work presented in this thesis (the work described in Chapter 5) is to study a direct alternate approach to this module, one that can take into account the temporal relations between pulses and include more information about each of them. The inclusion of that additional information would be able to hopefully achieve a better selection of prominent pulses and consequently improve scatter classification.

2.3 Summary

In this Chapter, LZap and its main components are described. LZap algorithm takes the raw information from the PMT arrays and processes it. The *PhotonDetection* (Section 2.1.1) separates, characterizes and classifies the individual pulses in the event. The *InteractionDetection* module utilizes this information

in the context of the event to find which S1 and S2 pulses are originated from scatters - this characteristic is defined as prominence. Then, the type of scatter of the event is determined, an essential step in data analysis in LZ. The algorithm that discriminates between non-prominent and prominent pulses is called *PulseHighlighter* and it is described in Section 2.2.

Chapter 3

Neural Networks

Machine Learning (ML) is a subfield of computer science, concerned with building statistical models with high predictive power. There are many different algorithms to build these models, but the general method behind these algorithms always relies into getting a dataset and building the model based on it.

Machine Learning can be broadly divided into four main groups based on the learning method: supervised, semi-supervised, unsupervised and reinforcement.

3.1 Supervised Learning

In supervised learning, the input data, x_i is paired with a label y_i, that is assumed to be related to the input by a unknown function f: $\mathbf{y} = f(\mathbf{x})$. The goal of supervised learning is to build a function h - the hypothesis or model - that approximates function f [41]. Supervised learning is mainly used for two predictive tasks: classification - predicting the class of a certain input - and regression - predicting the (real numbered) value of a quantity based on the input.

There is a large number of different possible approaches to supervised learning. The algorithm of choice to approximate f depends on various factors: the nature and dimensionality of the data, the task to complete and the bias-variance tradeoff. This last factor refers to the tradeoff between the predictive power and the generalization power of the model. Some popular choices for supervised learning approaches are decision trees and neural networks.

A Neural Network (NN) is a method inspired by the biological connection between brain neurons. The building block of NNs are units known as neurons. The simplest NN is a single neuron and is named perceptron. Perceptrons are simple binary classifiers that take an input $\mathbf{x}$ (associated with real numbered values known as weights, $\mathbf{w}$) and transforms that input as:

$$p(\mathbf{x}) = \Theta\left(\mathbf{w} \cdot \mathbf{x} + b\right) \tag{3}$$

where Θ is the Heaviside step function - it is the activation function of th neuron of the perceptron - and b is a real number associated with the neuron called bias. The perceptron training is the process of running the dataset repeatedly through the perceptron and compare the output with the event label. After each cycle - called an epoch - the weights (and bias) are updated in order to reduce the amount of mis-classifications produced by the perceptron [42].

3.1.1 Activation function

In more complex NNs, with more than one active unit, the Heaviside step function can be replaced with another activation function. Popular choices are the rectified linear unit (ReLU), the hyperbolic tangent (tanh) and the scaled exponential linear unit (SELU):

$$\text{ReLU} \equiv max(0, x), \tag{4}$$

$$\tanh \equiv \tanh(x), \tag{5}$$

$$\text{SELU} \equiv \begin{cases} \lambda\alpha\left(e^x - 1\right) & x < 0 \\ \lambda x & x \geq 0 \end{cases}, \tag{6}$$

For binary classification problems, the step function can be replaced by the sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

The sigmoid function truncates the output of the NN between 0 and 1 and the output value can be interpreted as a probability that a certain event is of class 1 instead of class 0. To transform the sigmoid truncated output into a binary classified, a threshold between 0 and 1 can be chosen.

3.2 Feedforward Neural Networks

The feedforward NN are sequences of layers. Each layer is formed by a given number of neurons. The input of each layer is the output of the previous layer (except for the first layer, the input layer, which received the data and transmits it to the NN) - information flows in one direction, from the input. Each neuron on a layer can receive the output of one or more neurons of the previous layer - when every neuron of a layer receives the output of the entire previous layer as input it is said to be fully (or densely) connected. The layers between the input and the output layer are known as hidden layers. A schematic of an arbitrary feedforward NN is shown in Figure 17.

The output, g, of a densely connected layer l, is given by, in matrix form:

$$\mathbf{g}^{(l)} \equiv \mathbf{g}^{(l)}\left(\mathbf{g}^{(l-1)}\right) = f^{(l)}\left(\left(\mathbf{W}^{(l)}\right)^T \mathbf{g}^{(l-1)} + \mathbf{b}^{(l)}\right) \tag{8}$$

where $\mathbf{W}^{(l)}$ is the weights matrix of layer l, with shape $m \times n$, where n is the number of units layer l

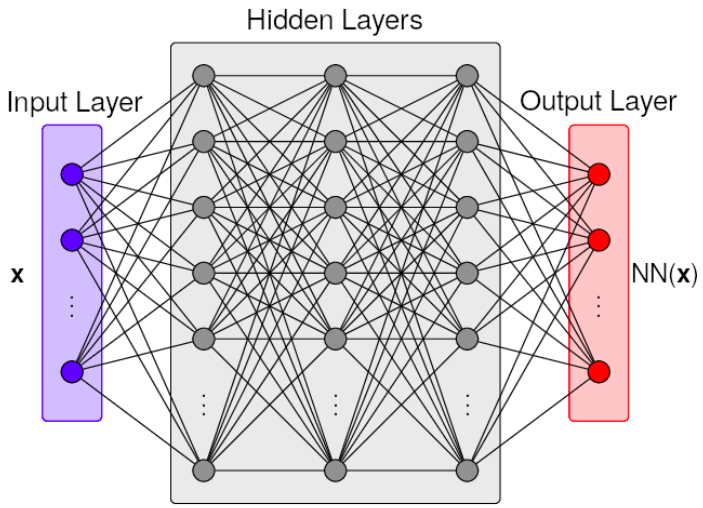


Figure 17: Schematic of an arbitrary feedforward neural network. The circles represent the nodes and the lines represent the links between them. Each column is a layer, and information flows from the left to the right. Image adapted from [43].

and m the nubmer of units in layer $l-1$; **b** are the bias of the n neurons on the current layer; and f is the activation function chosen for the layer.

The output of the previous layer can also be written as a similar function depending on the outputs of its previous layer. Therefore, a NN can be thought as a function of nested functions. Given an input **X**, the output of the NN, is given by the ouput of the last layer $\mathbf{g}^{(L)}$, which takes as object the output of the previous layer, which itself takes as object the output of the previous layer, and so on and on, as show in Equation 9.

$$NN(\boldsymbol{X}) = \mathbf{g}^{(L)}\left(\mathbf{g}^{(L-1)}\left(\mathbf{g}^{(L-2)}\left(\ldots\left(\mathbf{g}^{(1)}(\mathbf{X})\right)\right)\right)\right) \tag{9}$$

where L is the number of layers (excluding the input layer) of the feedforward NN and the superscript indicates the layer. Note that altough Figure 17 shows a network strictly composed by similar layers, feedforward layers can be and are used in conjuction with other types of layers, such as recurrent and convolutional layers, discussed below.

3.2.1 Parameter optimization

´ As in the perceptron, the training refers to the process of updating the weights and biases to improve the performance of the model. The foundation of the optimization algorithm for feedforward NN is gradient descent with backpropagation. To employ this algorithm, a (differentiable) function has to be chosen, to be able to numerically compare the NN predictions with the true classes/values - this function

is named loss (or cost) function. For regression problems, a popular choice is thmbree mean squared error (MSE) function:

$$MSE(\mathbf{y}, \mathbf{g}^{(L)}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \mathbf{g}_i^{(L)})^2 \tag{10}$$

where $\mathbf{y}_i$ are the true labels of event i, $\mathbf{g}_i^{(L)}$ is the NN output for the same event and n is the number of events evaluated. The gradient of the loss function, C, for a model with L layers can be written as:

$$\nabla C = \left(\mathbf{W}^{(1)}\right)^T f^{(1)'}(\mathbf{z}^{(1)}) \dots \left(\mathbf{W}^{(L)}\right)^T f^{(L)'}(\mathbf{z}^{(L)}) \nabla_{\mathbf{g}^{(L)}} C \tag{11}$$

The gradient depends on the weights $\mathbf{W}^{(l)}$, the derivation of the activation functions for each layer, $f^{(l)'}$ and the pre-activated output of each layer $\mathbf{z}^{(l)}$, and the gradient of the loss function with respect to the layer of the model $\nabla_{\mathbf{g}^{(L)}} C$. With these quantities, it is possible to compute the change $\boldsymbol{\delta}$ for updating the weights in layer l:

$$\boldsymbol{\delta}^j = f^{(l)'}(\mathbf{z}^{(l)}) \left(\mathbf{W}^{(l+1)}\right)^T f^{(l+1)'}(\mathbf{z}^{(l+1)}) \dots \left(\mathbf{W}^{(L)}\right)^T f^{(L)'}(\mathbf{z}^{(L)}) \nabla_{\mathbf{g}^{(L)}} C \tag{12}$$

The updated weights for layer l can be written simply as $\mathbf{W}_{new}^{(l)} = \boldsymbol{W}^l + \alpha\boldsymbol{\delta}^{(l)}$, where α is a small real valued number called learning rate, that determines how fast should the parameters should be updated in each cycle. This parameter can be constant or variable during training. The goal of the training process is to find the parameters which minimizes the loss function. The training stops after a set number of epochs (or after some condition - like loss stabilization - has been fulfilled).

There are many optimizer algorithms, such as Adam (Adaptive Moment Estimation) [44]. A key feature of these optimizers is their compatibility with mini-batch training, where updates are computed over small samplings of the training data instead of the entire dataset. This not only speeds up computations but also introduces a small factor of randomness, which can be useful to avoid some local minimums. Due to its efficiency and versatility, Adam is widely used in deep learning applications, even with different types of layers.

3.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of Neural Networks that were created for dealing with sequential data - each event is represented as an ordered lists of steps; each step (or timestep if it is a temporal sequence) is a list of values for determined variables - known as features - that characterizes that specific step - a feature vector.

Generally, a RNN unit can be thought as a feedback loop of n neurons, where n is the number of timesteps in the sequencial data, $\mathbf{X}$. The unit has a hidden state (with dimension p, the dimension of

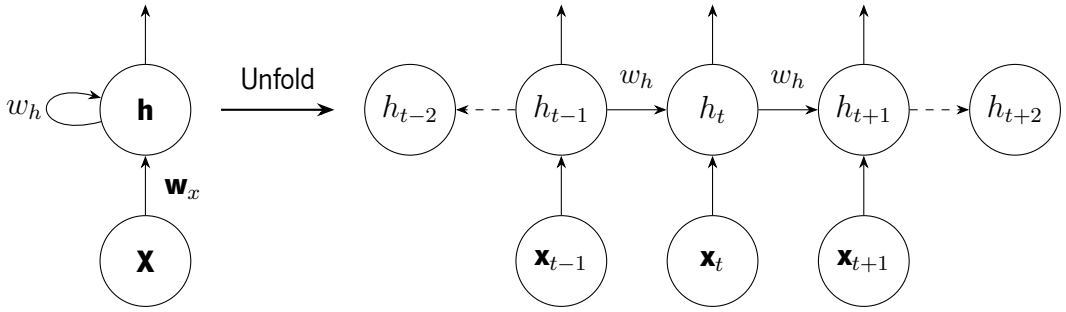


Figure 18: RNN cell scheme.

the feature vector **x**), h, that is updated for each timestep, according to the following transformation

$$h_t = f\left(\mathbf{w}_x \cdot \mathbf{X} + w_h \cdot h_{t-1} + b\right) \tag{13}$$

where $\mathbf{w}_x$ are the weights associated with the input links (as in the feedforward NNs), w_h is a weight associated with the cell, b is the bias of the cell and f is the (typically non-linear) activation function of the cell. The output of a RNN is the vector **h** with dimension n. But, given a certain task, certain inputs and outputs may be ommited (for example, for a classification task, only the last element of **h** may be considerer). RNNs applications can be broadly divided in four classes based on this choice: one-to-one (used in image classification, for example), one-to-many (used in image captioning), many-to-one (sequence classification) and many-to-many (frame labelling).

Due to their unfolding nature (where a single layer may unfold into many units), simple RNN are particularly vulnerable to the vanishing/exploding gradient problem, common in deep neural networks [45]. Another important problem is the loss of information when processing long sequences [46]. For this reason, the more popular applications of RNNs use gating mechanisms - a multiplicative interaction within the cell - to control the flow of information. The Gated Recurrent Unit (GRU) and the Long Short-Term Memory (LSTM) are some of the more popular implementations of this concept.

RNNs use a variant of the backpropagation algoritm briefly described in Section 3.2.1, named back-propagation through time (BPTT). This algorithm unrolls the Recurrent NN (RNN) and treats each loop as a layer in itself.

3.3.1 Long Short-Term Memory

Long Short-Term Memory (LSTM) [47] is a type of gated RNN where the standard RNN unit is replaced with a memory cell (Figure 19).

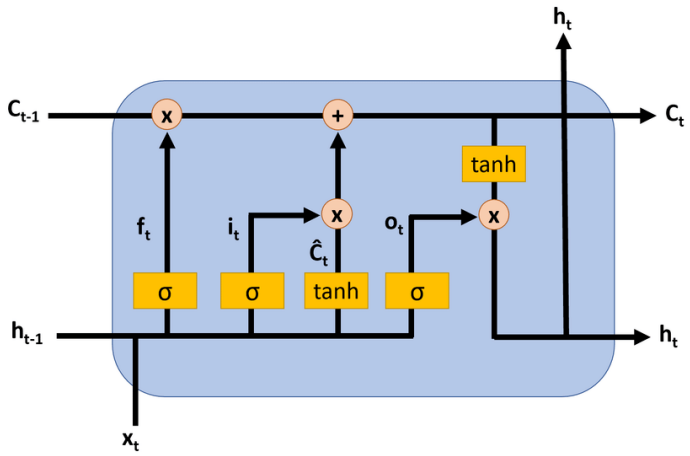


Figure 19: LSTM cell schematic. h_t is the hidden state of the cell. The two outgoing arrows for the hidden state, h_t represent both functions of the value: as the cell output at each timestep and as the input for the updating its own value on the following timestep. Image from [48].

As the name implies, the LSTM architeture is built to divide the flow of information into long-term and short-term memories. To achieve this, a new vector is introduced - the cell state **c**, associated with the long-term memory. Replacing the unit weight, four quantities, known as gates, are introduced: the input gate, $\overline{i}$; the forget gate, $\overline{f}$; the output gate, $\overline{o}$; and the new-state gate, $\overline{c}$. These gates have individual weights for the hidden state, w_h, and for the inputs, w_x; and bias, b. These quantities that can be optimized through training cycles. Each gate also has a fixed activation functions: in a typical LSTM, the input, forget and output gate utilize the sigmoid function, σ, while the new-state gate uses the hyperbolic tangent, tanh [42]:

$$\overline{f} = \sigma\left(\mathbf{w}_{xf} \cdot \mathbf{x}_t + w_{hf} \cdot h_{t-1} + b_f\right) \tag{14}$$

$$\overline{i} = \sigma\left(\mathbf{w}_{xi} \cdot \mathbf{x}_t + w_{hi} \cdot h_{t-1} + b_i\right) \tag{15}$$

$$\overline{o} = \sigma\left(\mathbf{w}_{xo} \cdot \mathbf{x}_t + w_{ho} \cdot h_{t-1} + b_o\right) \tag{16}$$

$$\overline{c} = \tanh\left(\mathbf{w}_{xc} \cdot \mathbf{x}_t + w_{hc} \cdot h_{t-1} + b_c\right) \tag{17}$$

The cell state is updated according to the following transformation:

$$c_t = \overline{f} \cdot c_{t-1} + \overline{i} \cdot \overline{c} \tag{18}$$

Since $\overline{f}$ is truncated between 0 and 1, this operation can be interpreted as an operation to decide how much long-term memory should be mantained or forgotten. In the second operation, a new value, $\overline{c}$ is created to add to the long term memory (new-state gate) and a second value, $\overline{i}$, truncated between

0 and 1, decides how much of this new state is to added to the cell state.

The hidden state is updated by:

$$h_t = \overline{o} \cdot \tanh\left(c_t\right) \tag{19}$$

The output gate, truncated between 0 and 1, determines how much of the cell state is leaked into the hidden state.

3.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are another class of neural networks, designed to process grid-like data with strong local dependencies (like image classification) [42]. For the purposes of this thesis, two-dimensional data is considered, but CNNs can also process three-dimensional data (for example an image with three color channels) - the procedures described below are identicaand Simard, P. and Frasconi, P.I.

3.4.1 Convolution

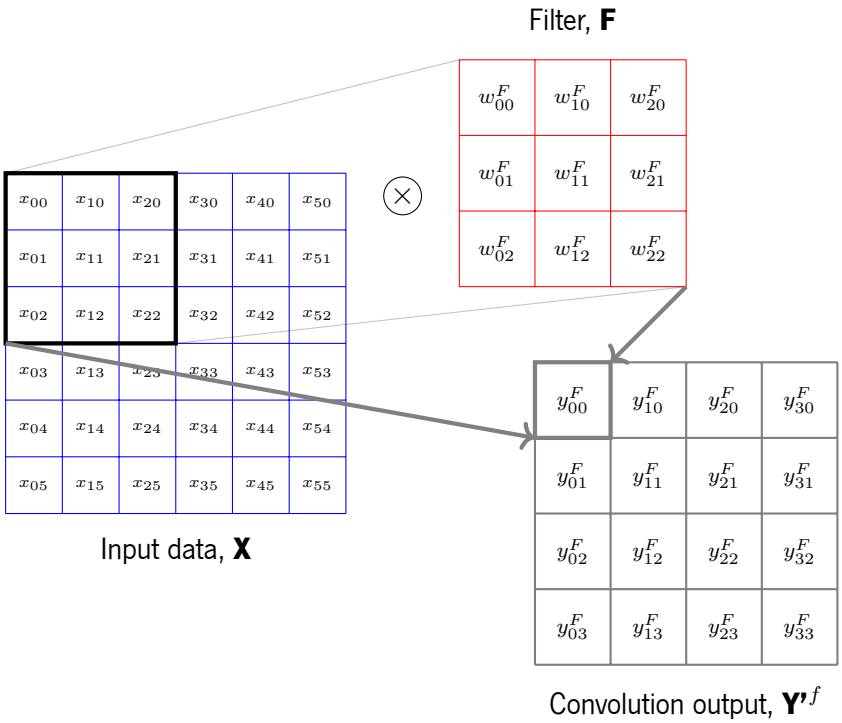


Figure 20: Schematic for the convolution operation in a convolutional layer. By sliding the filter to the right or below one step, iteratively, all the other outputs can be computed. The output of the convolution usually is put through an activation function, like ReLU, which is not depicted in the figure.

Like the name suggests, the main operation in a convolutional layer is a convolution. Consider an input **X**, with dimensions $n \times m$. A grid (with dimensions $n' \times m'$, where n' and m' are divisors of n and m respectively) with weights, w_{ij}^F for each element (and a bias, b^F, common to all), named filter. When the filter is applied to the input, it convolutes with a subset of the input with dimensions $n' \times m'$, generating an output for the neuron, y_{kl}:

$$y_{kl} = f\left(\sum_{i=k}^{k+n'} \sum_{j=l}^{l+m'} x_{ij} w_{ij}^F + b^F\right), \quad k < n - n', l < m - m' \tag{20}$$

This process is schematized in Figure 20. The output is created by choosing every possible k and l (it can be thought as the filter striding through the input, one element at the time - it is said that the filter has stride $s = 1$) Since $k \leq n - n'$ and $l \leq m - m'$, the output will have dimension $n - n' + 1 \times m - m' + 1$. In the case the stride, s, is bigger than 1, the dimensions are $\left(\frac{n-n'}{s} + 1\right) \times \left(\frac{m-m'}{s} + 1\right)$ [49].

In this class of NNs, the user usually does not choose the number of neurons, but a number of filters. The number of neurons is determined by the number of filters (for each filter there is a grid output) and by the size of the filter (that determines the shape of the output, and consequently, the number of neurons in the grid). Each filter in a layer has independent weights and bias but all have the same dimensions.

This process is efficient because allows the reduction of the trainable parameters while retaining the local dependencies of the data. While a fully connected feedforward NN has $N \times M + M$ amount of trainable paramters (for an input of N elements and an output of M elements), a convolution layer has $N_F \times (n' \times m' + 1)$ - the number of trainable parameters does not depend on the input size (it is important to note that the number of operations depends on the input size, so the CNN is still dependent on the input size).

The optimization algorithm of the convolutional layers is a variation from the backpropagation for feedforward networks - the gradients of the loss are calculated with respect to the weights in the filters by summing the contributions from all locations in the input where the filter was applied.

3.4.2 Pooling

Usually, CNNs are used to handle large data structures like images (a 100 by 100 pixels image have 10 000 data points that can be conceived as features). Convolution allows for a small number of trainable parameters to process the model, but it does not quickly reduce data dimensionality (for convolutional layers with stride equal to 1). Pooling is a strategy to downscale the data dimensionality while avoiding significant information losses [49].

The data with $N \times M$ dimensions is divided in equal regions - pools - with dimensions $n \times m$ (these regions can overlap - by comparing with convolution filters, the pool window can have different

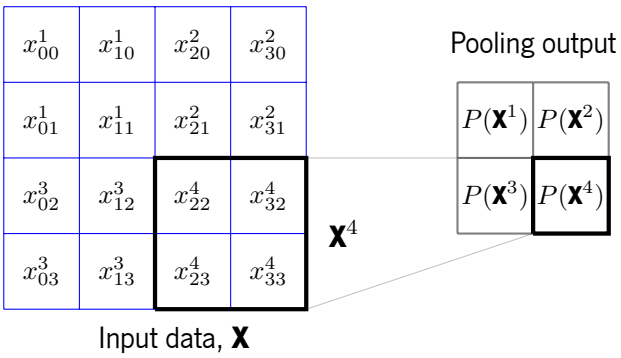


Figure 21: Pooling process in a 4×4 grid with a 2×2 pool window with stride 2. The output has dimension $\frac{4}{2} \times \frac{4}{2} = 2 \times 2$. The pooling function takes in a vector and returns a scalar. A popular choice for P is the max function - Max Pooling.

strides). Pooling is the process to select one value to represent each pool, creating a new grid, with dimensions $\frac{N}{n} \times \frac{M}{m}$ (Figure 20). There are many possible ways of choosing the representative value: a popular choice is choosing the maximum value present in the pool (Max Pooling). Other possible options is choosing the pool average (Average Pooling), or even a stochastic weighted choice (Stochastic Pooling).

Pooling layers do not have trainable parameters. Therefore, only the gradient of the layer with respect to the previous layer needs to be computed. For Max and StochasticPooling, every gradient is 0 except the one value that was chosen. For AveragePooling, every gradient is $\frac{1}{n}$, where n is the number of inputs in the pooling neuron.

3.4.3 One-Dimensional Convolutional Layers

Convolutional NN (CNN) were created to process two or three dimensional data. An option to deal with one dimensional data with these methods is to represent the data in a two dimensional structure - and many methods can be used to create this structure, (including directly reshaping the 1D structure into a 2D structure to create a *vibration image* [50]). However, these methods can be inefficient and require a large amounts of labeled data to be properly trained.

The One-dimensional Convolutional layer (Conv1D), an adaptation of regular convolutional layers, was proposed to efficiently handle one-dimensional data [51]. This convolution process is computationally simpler and tend to achieve better results than the restructuring methods shown above [52]. The parameters that define a Conv1D layer are similar to the typical convolutional layer: the number, size and stride of the filters - the difference is that the filter now has dimension k instead of $n \times m$. The output size is given by $\frac{N-k}{s} + 1$, where N is the input size, k is the filter size and s is the filter stride. The process is depicted in Figure 22.

Despite its relatively recent emergence, Conv1D quickly became popular in applications such as

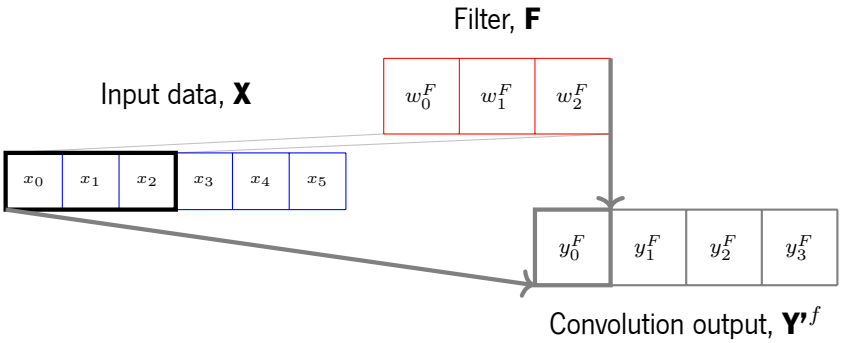


Figure 22: Schematic reprentation of a one-dimensional convolution with a filter of size 3 and stride 1 applied to a input of size 6. The output size is $\frac{N-k}{s} + 1 \rightarrow \frac{6-3}{1} + 1 = 4$.

electrocardiogram signal analysis and monitoring of high-power circuitry, where data comes in the form of timeseries [52].

3.5 Hyperparameter Optimization

While the weights and bias are the trainable parameters of neural networks, many factors can contribute to the performance of the network that can be defined by the user before training (and are fixed thereafter). These are called hyperparameters. They can be divided into two categories: model hyperparameters, related to architeture of the model (number of layers, size of the layer, activation functions); and algorithm hyperparameters (learning rate, batch size). Hyperparameters can be discrete, continuous, boolean or categorical. Hyperparameters can even be conditional (for example, the number of neurons in an optional layer is only applicable when the layer is added). The hyperparameter possibility space can be large and therefore, the choice of hyperparameter values can be arbitrary.

Hyperparameter optimization refers to the application of methods to find the optimal set of hyperparameters, based on a objective function that measures model performance (for example, minimum loss). Since the relation between the objective function and the hyperparameters may not be differentiable, the gradient-based optimization methods (such as gradient descent) may not be adequate. A possible solution to this problem is to use brute force methods such as Grid Search and Random Search. Brute force refers to the process of selection, which involves producing a large number of models with different hyperparameters and select the best values. These methods are time and computational intensive and are more adequate for models with small hyperparameter possibility spaces [53].

A possible alternative is algorithms based in Bayesian optimization. The process is the following:

1. Build the regression model and set the hyperparameter search space;
2. Randomly initialize the hyperparameters;
3. Run the model and compute objective function;

4. Build a new model with the new hyperparameters suggested by the acquisition function;
5. Repeat step 3 and 4 until an end condition is met.

The objective function and the hyperparameters of the sequentially trained models build a surrogate model - a probabilistic model - that is updated at each iteration. The acquisition function selects the next set of hyperparameters using the surrogate model and the predicted objective score on the previous set of hyperparameters. This approach allows for a more efficient resource management [54] and thus it is extremely useful for optimizing the hyperparameters of deep neural networks.

Optuna [55] is an open source library for hyperparameter optimization based on bayesian optimization. Optuna provides a flexible, modular and efficient *define-by-run* framework for defining search spaces, handling conditional hyperparameters and advanced optimization strategies. The unit of the Optuna process is the trial object - in each trial, the hyperparameters are chosen given the hypertraining history; the model is built and trained; and the objective score is found.

3.6 Summary

The chapter begins with a brief description of the concept of supervised learning (Section 3.1). Section 3.2 describes the principles behind the Feedforward Neural Network, including an overview of the backpropagation method (Section 3.2.1). Section 3.3 gives attention to Recurrent NNs and more precisely to LSTMs, while Sections 3.4 and 3.4.3 for Convolutional NN and Conv1D. Finally, hyperparameters optimization is described in Section 3.5.

Chapter 4

The datasets

To properly train and test a model, an adequate dataset has to be chosen. LZ events can vary in many ways, as we already seen above. In principle, a diverse dataset that includes most relevant topologies would be optimal. The chosen dataset contains events simulated according to 5 different sources (hereafter, the dataset will be refered by Five Sources dataset (5S Dataset)) that will be discussed and described below. The events are represented in two different forms: a paneled representation, with the ordered rows represent time-ordered pulses and the columns representing the RQs chosen to characterize the pulses; and a wafeform representation, where the event is described solely by the sum of the digitized PMT response. Two more datasets - background-only dataset (BG Dataset) and deuterium-deuterium dataset (DD Dataset) - will be used as stress testing to verify the model results, but only for the paneled data approach, due to the limitation of computational resources. One of this datasets, DD, contains data from a source not present in the training dataset.

The number of events in each dataset is shown in Table 1, including the number of events *per* scatter class. The high number of SSs can be explained by the very nature of the detector, designed to search for WIMP - that produce SS events. The exception is the DD Dataset for reasons detailed below.

Dataset	**Single Scatter**	**Multi Scatter**	**Other Scatter**	**Total**	**LZap Version**
5S	4958	643	279	5880	5.5.3/5.5.4
DD	5170	6056	3303	14529	5.8.0
BG	18706	1787	1133	21626	5.8.0

Table 1: Number of events (divided by scatter class) for each dataset. The LZap version for the 5S dataset was a development version between the 5.5.3 and 5.5.4 versions.

Every event used in this work is a real event obtained by the LZ and processed by the LZap (in the case of the paneled data). The LZap version is shown in Table 1.

4.1 LZ Calibration Sources

Rigorous calibrations are essential to ensure the proper functioning of the experiment - they are done before, after and during pauses in science runs to guarantee the quality of the data. Calibration sources vary in goal, nature and application method. LZ calibrations are listed in Table 2, divided in categories according to their nature and method of application.

	Isotope	Particle Species Used	Energy (KeV)	$t_{1/2}$	Deployment
A	^{83m}Kr (^{83}Rb)	IC, AE, x-ray, γ	32.1 and 9.4	1.83 h	Internal
	^{131m}Xe (^{131}I)	γ, x-ray	163.9	11.8 d	Internal
	^{220}Rn (^{228}Th)	α, β, γ	Various	55.6 s	Internal
B	^{3}H	β	0 – 18.6	12.3 y	Internal
	^{14}C	β	0 – 156.4	5730 y	Internal
C	^{241}AmLi	(α, n)	(5638, 0 – 1500)	433 y	External
	^{241}AmBe	(α, n)	(5638, 0 – 11×10^3)	433 y	External
	^{57}Co	γ	122	272 d	External
	^{228}Th	γ	2615	1.91 y	External
	^{22}Na	γ	511, 1275	2.60 y	External
	^{54}Mn	γ	835	312 d	External
D	^{88}YBe	(γ, n)	(1836, 152)	107 d	External
	^{88}YMg	γ	1836	107 d	External
E	DD	n	2450	–	–
	D-Reflector	n	270 – 420	–	External
	H-Reflector	n	10 – 200	–	–
F	^{241}Am	α	5638	433 y	External

Table 2: List of calibration sources grouped in 6 categories. A and B represent sources gaseous sources dispersed in the Xenon (the isotopes between parents in A represent the parent isotopes); C represents neutron and gamma sources that are inserted in the same fashion; D are very low energy sources; E are deuterium-deuterium neutron sources; and F is a calibration source for the PMTs. Table reproduced from [56].

Categories A and B are gaseous sources that are dispersed through the Xenon Circulation System; C are gamma and neutron external sources deployed close to the TPC to calibrate the TPC responde to external background; D are photoneutron sources designed to test the TPC response to low energy scatters; E are deuterium-deuterium neutron sources; and F are sources used to calibrate the OD optical

calibration system [56].

4.1.1 Gaseous Calibration Sources

The five gaseous calibration sources (divided into two categories, A and B, in Table 2, by the method of application in the TPC) are radioactive isotopes that can be fed into the liquid phase of the Xenon in the TPC through the Xenon Circulation System. This approach allows the sources to overcome the Xenon skin veto and to achieve a relatively homogeneous spacial distribution in the active volume that would be difficult to obtain with external radioisotope sources.

^{83m}Kr, ^{131m}Xe and ^{220}Rn are "generator sources": noble gases, obtained by decay of parent isotopes, injected in the Xenon circulation system before purification to avoid the accidental injection of impurities in the Xenon. ^{220}Rn is used to obtain a flow map of the LXe in the TPC, while one of its decay (^{212}Pb) can be used to calibrate the detector response to ERs; ^{83m}Kr and ^{131m}Xe are both monoenergetic sources and therefore have some overlap in purpose: for example, both can be used to determine the "electron lifetime" (the probability that a ionization electron survives the drift towards the xenon liquid-gas interface) and signal collection efficiencies. Due to the higher half-life and energetic peak, ^{131m}Xe is preferred for these calibrations: the higher lifespan allows a better homogeneity in the xenon volume and its monoenergetic γ deposition of around 163 KeV can be discriminated in data analysis during science runs. [56]. Despite the relatively high energy of the xenon decay, it will mostly produce single-site interactions (SS events).

^{3}H and ^{14}C are stored in ^{3}H-labeled CH_4 (or in ^{3}H-labeled CH_3T) and ^{14}C-labeled CH_4, respectively. These are mixed with gaseous Xe in pressure cylinders and that mix can be injected to the xenon circulation system. These sources have an high lifespan and therefore have to be removed by the purification system. ^{3}H provides a low energy range (with a 18.6 KeV maximum) used to calibrate the detector for low energy electronic recoils. [21]. ^{3}H labeled tritium-methane is used for calibrating the response of the detector for low-energy Electric Recoils.

4.1.2 Neutron Calibration Sources

Neutrons as a calibration sources are extremely important for the LZ experiment not only to test the veto systems described in Chapter 1, but also to characterize the detector response for Nuclear Recoil, the expected signature of a WIMP.

Americium-lithium (AmLi) and americium-beryllium (AmBe) are external sources that are deployed in three calibration source deployment (CSD) tubes that run down the sides of the TPC inside the vacuum space of the cryostat (category B sources). The neutron production process in these sources is similar: the americium isotope (^{241}Am) decays - $^{241}\text{Am} \rightarrow \alpha + {}^{237}\text{Np}$; the α-particle bombards the ^{7}L (or ^{9}Be)

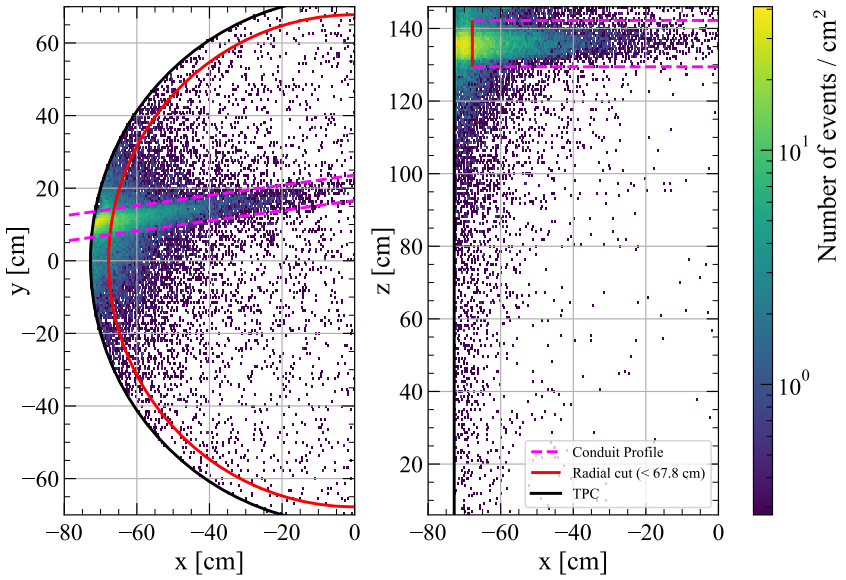


Figure 23: Spatial distribution of direct DD events in the TPC. For analysis, only events located further than 5cm from the TPC wall (solid red line) and between z=129 to z=140cm are used in DD analysis. Image from [56].

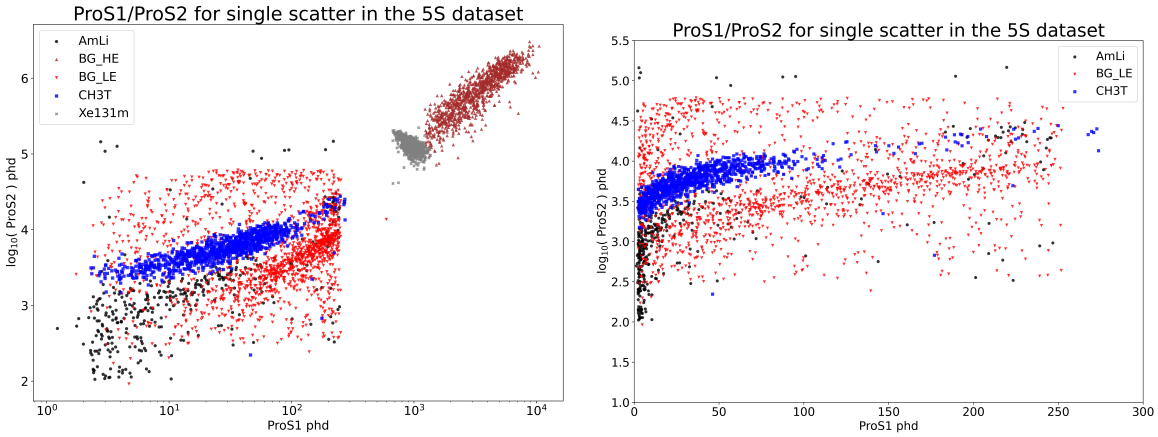
prompting a nuclear reaction: ^{7}L + $\alpha \rightarrow$ ^{10}B + n (or ^{9}Be + $\alpha \rightarrow$ ^{12}C + n) [57, 58]. The kinetic energy of the resulting neutron varies due to loss of energy of the α particle in the source - the maximum energy from an AmLi neutron is around 1.5 MeV and 11 MeV from an AmBe source, producing a maximum recoil energy in the TPC of 45 KeV and 330 KeV, respectively [56]. AmLi sources are used to calibrate for low-energy NRs around the expected energies of a WIMP recoil, while AmBe is used to calibrate for a wider, higher, range of NR energies.

Another source of neutrons for calibration is a deuterium-deuterium (DD) reaction: D + D $\rightarrow$ ^{3}He + n, where neutrons are produced monoenergetically at 2450 KeV. The neutron beam can be deployed directly throught a conduit or it can be reflected by an active scintillator at specific scattering angles to reduce the beam energy, allowing lower-energy NR characterization. Direct DD neutrons are deployed through an horizontal conduit, 3 cm below the liquid xenon surface inside the TPC - the spatial distribution of these events are shown in Figure 23; or through an angled conduit (20 degrees downward), for the reflected modes. This last conduit set up is built to maximize the length of neutron path in the xenon volume and increase the spatial separation in multiple scatter events [56].

4.2 Training Dataset

The training dataset is composed of events from 3 different calibration sources: 880 events from one neutron calibration source, AmLi, and two gaseous sources, Xe131m and CH3T, with 1000 events each. In addition, there is a number of background events in the datasets - those background events can be divided in low energy (BG-LE) and high energy (BG-HE) events - 2000 and 1000 events, respectively. These events were randomly sampled from a large background event population without applying fiducial cuts - cuts based in the position of the scatter in the TPC.

This diversity of events is the main advantage of using this dataset. Figure 24 shows the relation between the ProS1 and ProS2 for the single scatter events in the dataset. Figure 24a shows the difference between the higher energy sources like Xe131m (and the high energy background events) and the lower energy sources (AmLi and CH3T). In Figure 24b, it is possible to observe the formation of two bands. The CH3T band corresponds to low energy ERs, as expected. The other band is formed by BG-LE events, although this does not imply these events are NRs - since there are no fiducial cuts, there are a number of gas events, events in which the scatter occured in the gaseous phase of the xenon. These events are not understood as well as LXe events, and are generally removed with a minimum drift time condition.



(a) S1/S2 full space for the dataset.

(b) S1/S2 space zoom.

Figure 24: S1/S2 space for the 5S Dataset SS events.

In the work described in Chapter 5, only the LZap S1 and S2 are used. Figure 25 shows the ProS2 contribution to the total event area. As expected, events with larger ProS2s tend to have higher contributions from the ProS2s. For the events with smaller ProS2, the other pulses (NonProS2, S1, etc.) have relatively bigger contributions. This is specially true for the AmLi events, which also have the most pulses (Figure 26).

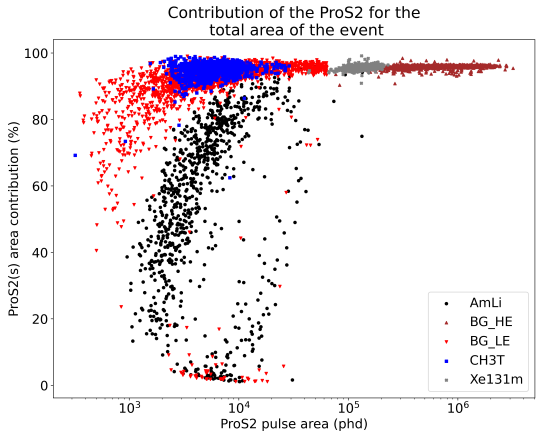


Figure 25: Contribution of the ProS2(s) for the total area of the event.

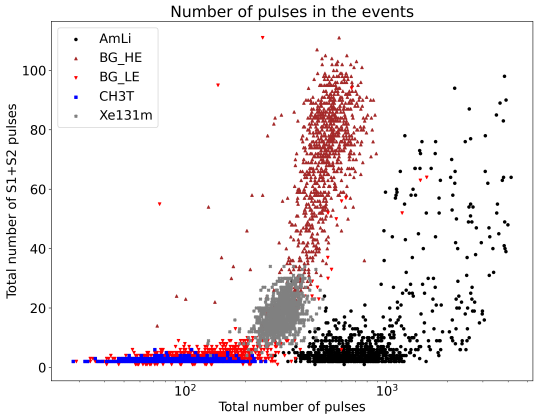


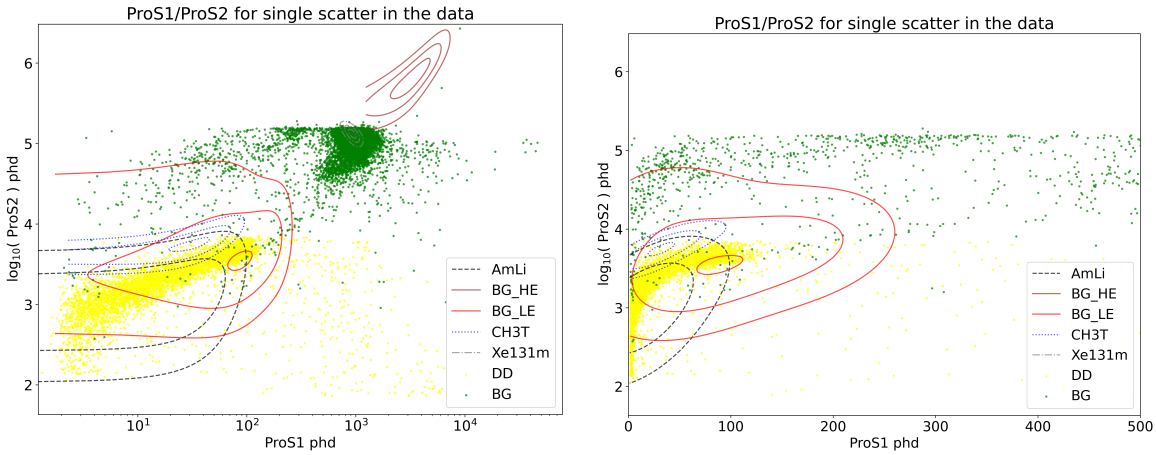
Figure 26: Number of pulses in 5S events against number of total events.

4.3 Stress Testing Datasets

The training dataset will be equally divided in three datasets: training, validation and testing. In addition, two different datasets, are used to test the limits of the model: a background-only dataset (BG Dataset) and a deuterium-deuterium dataset (DD Dataset). The latter are low energy neutron events from a deuterium-deuterium source that are not represented in the training dataset. As seen in Table 1, the DD Dataset has an higher proportion of MS than the other two datasets (it even has more MS than SS events). This is reflected in the extremely high ProS2/LZap S2 ratio in this dataset: $\approx$ 58% of the LZap S2s are ProS2s (comparing with $\approx$ 8% and $\approx$14% for the 5S Dataset and BG Dataset, respectively). The former is a dataset comprised of background events, but with a different profile than the events in the training dataset. This is shown in the S1/S2 profile for the SS events in the dataset (Figure 27).

The position of the events for the DD is also focused near the top of the TPC, due to the position of the conduit that delivers the source (see Figure 23). This affects the shape of the S2 signal - the S2 pulses are generally thinner than for other event sources, since the electrons travel, in average, less distance in the LXe. In Figure 28 it can be observed that the ProS2 from the DD Dataset are generally shorter both in amplitude and length than the ProS2 from the other two datasets.

These differences are highlighted to properly interpret the model performance on these stress testing datasets, specially given the low number of events in the training dataset - it may not be able to properly represent the full scope of the LZ events. On the other hand, the performance of the model on these datasets can be a good indicator of how well the model generalizes to unseen data. This is specially true for the DD data that is not represented in training and it has some specificities that make it an extremely challenging test to the model. In addition, it was recently found that DD data was not used to calibrate LZap and therefore the labels may not properly be as accurate as in other datasets.



(a) S1/S2 full space for the datasets.

(b) S1/S2 space zoom.

Figure 27: S1/S2 space for the SS events of all three datasets. The curves represent the subsets in the 5S dataset, and were drawn using Scipy Gaussian KDE implementation.

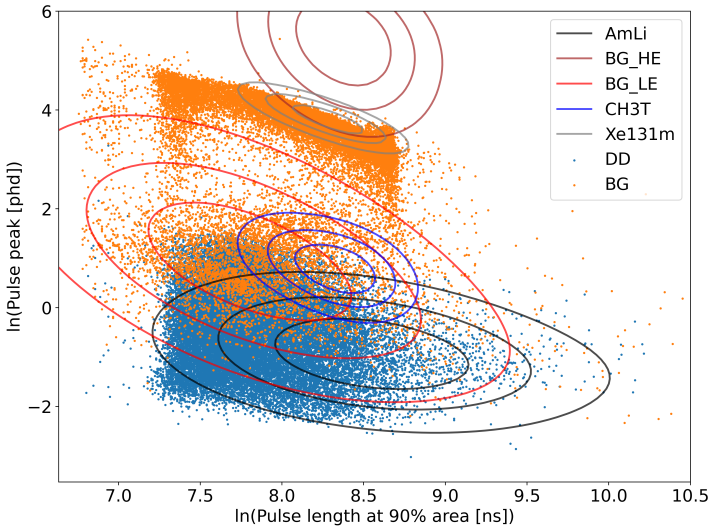


Figure 28: Pulse length at 90% area against the peak amplitude for the prominent S2s in the datasets. The focus of this plot is the DD dataset - some ProS2 from the 5S and BG datasets fell outside these boundaries of the plot.

4.4 Summary

This chapter introduces the datasets used for model training and testing, focusing on the Five Sources dataset (5S Dataset) - the training dataset -, which includes events from five calibration sources represented in paneled and waveform forms. Additional datasets, background-only dataset (BG Dataset) and deuterium-deuterium dataset (DD Dataset), used for stress testing, are also described. Section 4.1 details the calibration sources, categorized into gaseous, neutron, and PMT calibration sources. The training

dataset is described in Section 4.2 and its diversity is illustrated through the S1/S2 space, highlighting distinct energy bands. The stress-testing datasets, BG Dataset and DD Dataset, are introduced in Section 4.3, with the latter posing unique challenges due to not being represented in the 5S Dataset.

Chapter 5

ProFinder

As stated in Section 2.1, one part of the LZap process is to find the prominent S1 and S2 - those pulses that are classified as S1 and S2 by the PulseClassifier and that can be attributed to scatters. The goal of this chapter is to show the development of a machine-learning based approach to the prominence finder module, alternative to the approach employed by the LZap algorithm. This implies the events are already encoded by the LZap algorithm in pulses characterized by RQs.

Considerating the casual relations between pulses in the event, Recurrent NN (RNN), more specifically LSTMs, were chosen due to their proven capacity to detect patterns and context, as mentioned in Section 3.3.1.

The work described below was carried out using computational resources provided by the National Energy Research Scientific Computing Center (NERSC), a computing facility for the U.S.A. Department of Energy (DOE) Office of Science (SC). NERSC is a high-performance computing facility that supports scientific research across a wide range of disciplines. In particular, the *Perlmutter* system - a state-of-the-art supercomputer at NERSC designed for data-intensive and AI-driven workloads - was utilized. However, as a student, access was limited to a shared login node, which significantly constrained the computational power and storage available for the development of the models.

5.1 Dataset(s)

All datasets described in Chapter 4 are utilized in this work, either as the standard training-validation-testing dataset (5S Dataset) or as a stress testing dataset (DD Dataset and BG Dataset dataset). Only LZap S1s and S2s are used in this training. This choice was motivated by previous work on this matter and the need to reduce the computing power necessary to develop this method.

As stated in the introduction to chapter 4, an event can be thought a panel data - the ordered rows

represent the time ordered pulses[1]; each column of the paneled data represents a feature of the - a pulse RQ. In this work, 11 RQs are used:

- ln(*peakAmp*)
- *promptFraction50ns*
- ln(*pulseArea_phd*)
- ln(*coincidence*)
- ln(*pulseLength90*)
- ln(*pulseLength50*)
- *peakPos*
- ln(*peak2length*)
- *deltaTime*
- *pulseClass*
- *prominenceS2*

The definition of each RQ can be found in Appendix A except for *deltaTime* and *pulseClass*. The latter is a binary feature: 1 for LZap S2s and 0 for every other pulse. The former represents the time difference between a given pulse peak and the peak of the previous pulse (since only scintillation pulses are considered, this distance is taken in relation to the previous S1 or S2). The *deltaTime* first pulse in the event is zero. This feature is transformed according to:

$$\textit{deltaTime} \longrightarrow \ln\left(\textit{deltaTime} + 1\right) \tag{21}$$

Of these 11 RQs, all but *prominenceS2* are used as training data. The latter, on the other hand, is the training target: it has the value 0 for NonProS2 and the value 1 for ProS2.

For the training, the 5S Dataset is (equally) divided in three subsets - training, validation and testing. The data is scaled using a standard transformation :

$$\mathbf{x}_i \rightarrow \frac{\mathbf{x}_i - \overline{\mathbf{x}_i}}{\sigma_i} \tag{22}$$

where $\mathbf{x}$ is the data, i represents a given RQ and $\overline{\mathbf{x}}_i$ and σ_i represent the mean and standard deviation, respectively, for the ith RQ in the training subset. This scaling is also applied to the stress testing datasets.

To standardize the input shape of the events, all events are padded until each one has 112 rows (every event in all datasets have at most 111 scintillation pulses). The 5S Dataset is randomly divided in three different subsets: training, validation and test subsets (1960 events each).

5.2 Methodology

As stated previously in Chapter 4, the training dataset is equally divided in a training, validation and testing dataset. The training and validation subsets are used to train a model based on a BiDirectional

[1] Due to a code relic, from early approaches, the series is actually time-reversed, meaning that the first row represents the last pulse of the event. This should not affect the training, as the model is built to account for both time directions.

LSTM. Before training, the hyperparameters of the model are optimzed using Optuna. The hyperparameters of the best trial are used to build the model.

60 models, equally built, are independently trained on the training subset, and used to make an ensemble. The ensemble output for a given pulse is the average of the output of each independent model. The threshold of the model is chosen as the value that maximizes the F-Score of the validation subset ensemble output.

The use of an ensemble allows a second metric to characterize the pulse - the scores of the individual models can be used not only to determine an ensemble score but to compute a value that describes the variance of scores between individual models. To compute that quantity, which was called confidence level (CL), a straightforward bootstrapping method is employed: a new vector is created by sampling the values from the original output vector while allowing repeated selections; this process is repeated one thousand times to obtain one thousand new, boostrapped, output vectors. By taking the mean of each one of them, a distribution of bootstrapped means is created [59], which is used to compute a confidence interval for the ensemble score: the lower and upper bounds are chosen using the 2.5 and 97.5 percentiles of this distribution, respectively. The difference between the lower and upper bounds for the confidence interval are used to determine the CL for each pulse. A CL value of 0 implies that all 60 models agree exactly in the prominence score for that pulse.

Finally, the scatter type of the event is determined using a simplified version of the *InteractionFinder* algorithm described in Section 2.1.2 and the S2 pulses with a prominence score above the score threshold.

5.2.1 Model Architeture and Hyperoptimization

The architeture of the model is schematized in Figure 29. The model input layer is a Masking layer, to skip padded timesteps (pulses). Then, the data is processed by a Bidirectional LSTM (BiLSTM). This layer is in fact two LSTM layers, one processing the data without changes and the other *time-reversing* the data before processing. The output of this layer is the concatenation of the two individual LSTM layers, which are themselves a sequence of vectors with the same number of timesteps as the input.

The model ends with a Dense layer with one neuron and a sigmoid activation function - this is done to ensure a normalized output that could be interpreted as a confidence score of a certain pulse being a ProS2 or not. Between these two layers, 2 sets of consecutive Dropout - Dense layers can be added during hypertraining. After the BiLSTM layer, all timesteps are independently processed by those Dense layers.

The number of cells and neurons on the BiLSTM and Dense, respectively, are optimizable hyperparameters, as well the activation function for the Dense layers. The other hyperparameters related to each layer were left as the Keras default values (those values can be found in [60, 61])



Figure 29: ProFinder architeture scheme. The grey border indicate optional layers that can be added during the hyperparameter optimization.

The number of LSTM cell in the first layer is an optimizable hyperparameter. If a second LSTM is included, its number of cell is divisor of the number of BiLSTM cells - the factor that determines the divisor is a power of two, to be optimized during hyperparameter. The dropout rate is the same to both Dropout layers and is also an optimizable hyperparameter.

The mean squared error was chosen as the loss, and Adam was chosen as the optimizer. The learning rate, learning rate decay and the batch number are optimizable hyperparameters. A list of some relevant hyperparameters (fixed or trainable) is shown in Table 3.

Hyperparameter	**Range/Value**	**Step**	**Chosen Value**
`Optimizer`	Constant	-	Adam
`Loss`	Constant	-	MSE
`Epochs`	Constant	-	500
`Patience`	Constant	-	62
`Learning Rate`	0.0001 to 0.05	-	0.002472
`Decay` ($\times 10^3$)	0.001 to 0.1	0.001	66
`Dropout Rate`	0 to 0.4	-	0.0860
`Number of Batches`	10 to 42	4	30
`Number of Dense`	0 to 2	-	1
`BiLSTM Cells`	2 to 32	1	25
`Neurons Dense 1`	4 to 64	1	47
`Neurons Dense 2`	2 to 32	1	Not used
`Dense Activation Function`	(ReLU, SELU, tanh)	-	ReLU

Table 3: Model hyperparameters and respective search spaces, if optimizable, including step (N/A for continuous or categorical variables). In the Chosen Value column, the value chosen for each variable after the Optuna study, according to the best trial. Constant hyperparameters are fixed, and therefore do not have an Optuna range.

The models are trained during 500 epochs, with the Keras early stopping method [62] implemented with a patience of 62 epochs. The hyperparameter optimization was conducted for 600 trials, with the objective of minimizing the validation loss. The hyperparameters of the trial with lower validation loss are also shown in Table 3 and more details about the hyperoptimization can be found in Appendix B.

5.3 Training and Results

60 independent models were trained on the same dataset. No randomness control was imposed, therefore, all 60 models could show different results. The mean of the output of the 60 models is used

as a final pulse score. The metrics of those 60 individual models and comparison to the ensemble score can be found in Appendix B.

The different outputs are also used to compute a confidence score with bootstrapping: 1000 bootstrapped version of the 60 models output are created and their mean taken. The 2.5-percentile of the 1000 means is chosen as the lower bound and the 97.5-percentil is chosen as the higher bound. This method was selected to determine the confidence interval ,instead of a simpler and more direct quantity such as the standard deviation, because the output of the models will not necessarily follow a normal distribution around the ensemble score.

The threshold to discriminate between ProS2 or NonProS2 of the ensemble was defined as the threshold that maximizes the F-Score on the validation dataset. This value is 0.56937 and is shown in the PR Curve for the validation subset (Figure 30).

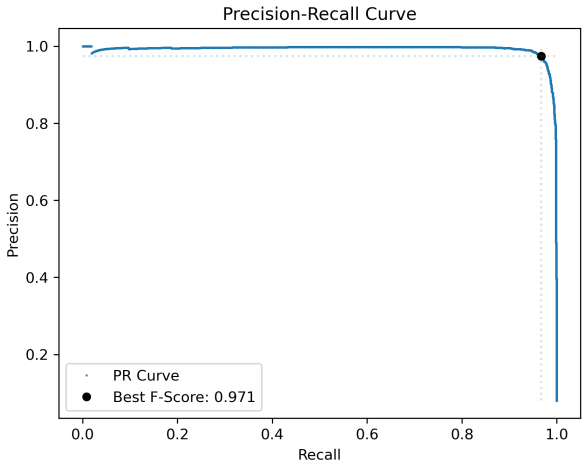
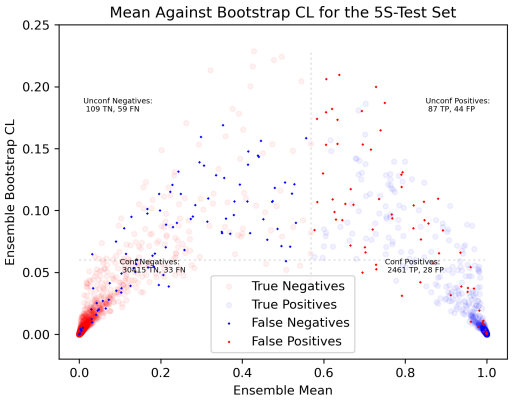


Figure 30: Validation subset Precision-Recall curve and the point with the best F-Score, at a threshold of 0.56937.

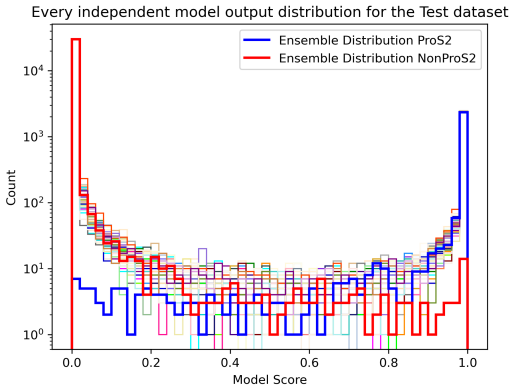
A CL threshold (CLth) is set to discriminate between high and low confidence events, based on the range of the bootstrap confidence interval - this value is defined as the 99^{th} percentile of the ranges in the validation subset and has the numeric value of 0.5983.

These two threshold create for regions in the ensemble mean - bootstrap CL space that was chosen to characterize the ensemble output. That space is shown in Figure 31a for the test subset. The output shows an high concentration of pulses scored very closely to 0 or 1, with very low ranges for the confidence interval, meaning that the independent models consistently scored those pulses as 0 or 1, respectively. This view is supported by observing the distribution of the output values for each independent models (Figure 31b), which shows that evey indepedent model has a very polarized output, with high concentration of scores close to 0 and 1 and low concentration on the intermedium values.

The two previously defined thresholds create for quadrants in the ensemble output space, dividing



(a) The ensemble output for the test subset. The dotted lines represent CL and score thresholds.



(b) Output distributions for all indepedent models.

Figure 31: ProFinder output on the test subset. On the left, it is possible to see an arch like shape, where the pulses scored with intermediate values tend to have larger confidence intervals. On the right, the distributions for each independent model output shows that every model had similarly polarized distributions.

the output in terms of score and confidence. As it can be seen in Figure 31a, above the confidence threshold there is an higher ratio of incorrectly classified pulses than below it. To compare confident and unconfident ensemble outputs, the ROC Curves are compared (Figure 32) along with Precision, Recall and AUC metrics (Table 4). It can be seen that the ensemble skill falls to near a random classifier when considering only unconfidently scored pulses, with significantly worse values for the classification metrics.

Test Subset	**Precision**	**Recall**	**AUC**
All Data	0.972	0.965	0.9995
Above CLth	0.664	0.596	0.7072
Below CLth	0.989	0.987	0.9997

Table 4: Comparison between confident and unconfident pulse classification.

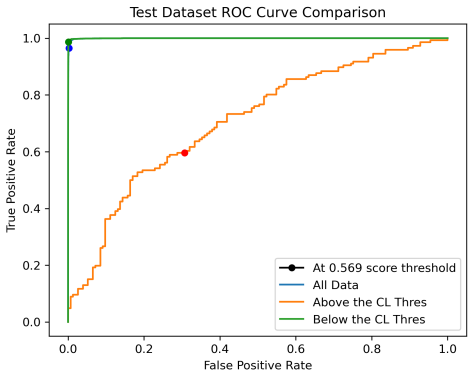


Figure 32: Comparison between the ROC Curves for the confident and unconfident classification.

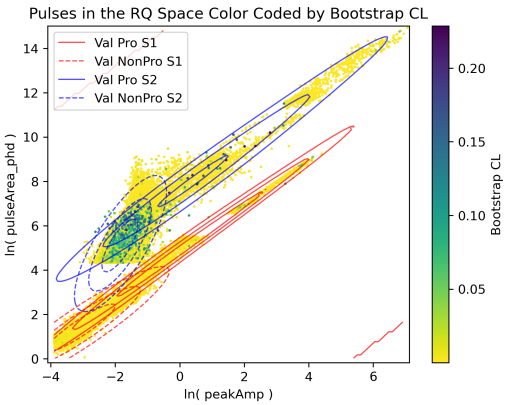
The 299 unconfidently scored pulses are distributed by 133 out of the 1960 events in the test subset

($\approx 93.2\%$) - 88 AmLi, 21 BG-LE, 24 BG-HE and 1 CH3T event. The low or null presence of CH3T or Xe131m is expected, since these events tend to have a more trivial topology, with less and more defined scintillation pulses, as shown in Chapter 4. AmLi, on the other hand, is composed by less clean events, with higher noise. AmLi is also a NR source, producing smaller S2 when compared to ER sources. This, coupled to the higher detector noise environment, makes the AmLi data extremely challenging to reconstruct.

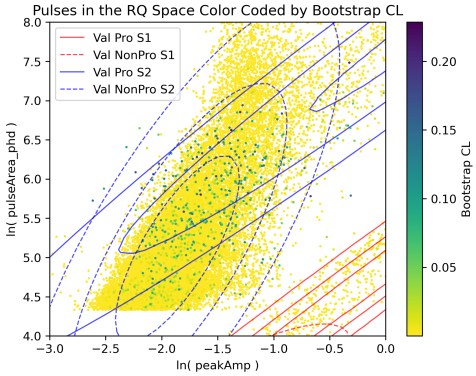
Observing the pulseArea - peakAmp slice of the RQ space (Figure 33) it can be seen that the region where the ProS2 and NonProS2 countours intercept, as expected. It is important to note that even in this region, the ensemble performs very well - for the pulses obbeying the following conditions:

$$4 < \ln(\text{pulseArea_phd}) < 8, \qquad -3 < \ln(\text{peakAmp}) < 0 \tag{23}$$

the ensemble has a precision of 0.934 and a recall of 0.902 for pulses in this region. Figure 33b shows this region of the RQ slice space, illustrating these observations.



(a) Slice of the RQ space of the test subset.



(b) Zoom in slice of the RQ space of the test subset.

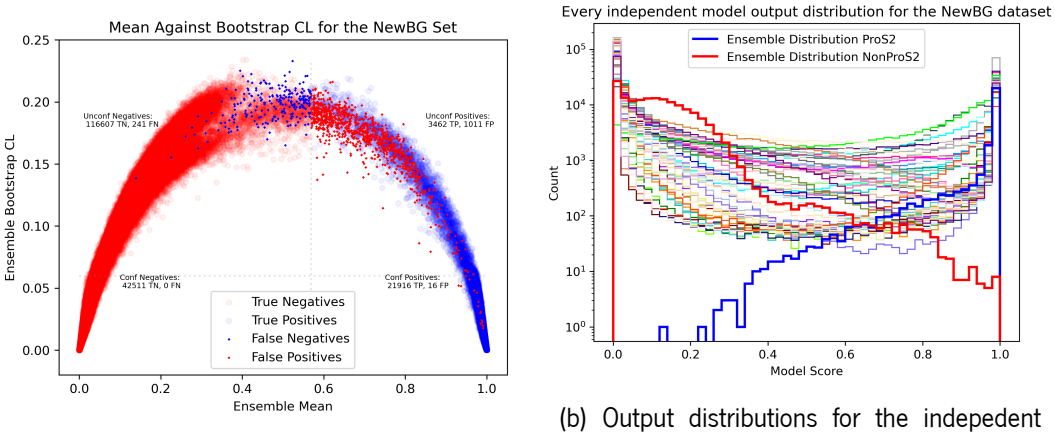
Figure 33: Slices of the RQ space for the test subset. Pulses are colormapped according to the bootstrap interval range. The countours were drawn using Scipy Gaussian KDE implementation [63] (with a kde.factor of 0.75 for non-prominent pulses and 1.5 for prominent pulses) applied to the validation dataset for each specific slice.

5.3.1 Stress Testing

The ensemble was applied to two different datasets: the BG and the DD Dataset, described in detail in Chapter 4.

5.3.1.1 BG Dataset

The BG Dataset is a collection of background events from LZ, the same event origin as the events in the BG subsets of 5S Dataset. One of the 5S Dataset shortcomings was its low amount of events, which means the background subset may not properly represent the background data range. Therefore, it should be expected a good ensemble performance, if less optimal than for the test subset due to the training dataset limitations.



(a) Ensemble output.

(b) Output distributions for the indepedent models.

Figure 34: ProFinder output on the BG dataset. The individual model output distribution (right) explain the high arch uniformity and height, as compared to the results on Figure 31.

The ensemble score-CL space for the BG data is shown in Figure 34a. When compared to the arch from the test subset output (Figure 31a), two observations can be made. The first is that the arch is taller and more uniform, meaning more pulses where less consistently scored by the individual models. This is supported by the output distributions of the independent models (Figure 34b), where the distribution is still clearly bimodal but the intermedium bins are more populated, that the models are disagreeing more regarding the prominence of the pulse.

It is important to note that the number of False Positives is much higher than the number of False Negatives - approximately 3.9% of the pulses scored as ProS2 are wrongly scored, compared to only 0.15% of the NonProS2 being misclassified. This discrepancy is relevant since losing prominent S2s (False Negatives) is worse (from an analysis standpoint) than gaining prominent S2s (False Positives). The second observation is that, as on the test subset, the pulses that were confidently scored tend to also be more correctly scored. In fact, only 14 of the 1266 wrongly scored pulses have a lower than threshold CL. Figure 35 shows the (upper left corner of) the ROC curves for the ensemble output above and below the CL threshold where the difference between the ensemble performance for the lower and higher than CL threshold pulses is clear. For the metrics shown in Table 5, the only metric that deteriorates significantly is the precision, which can be explained by the disparity between FNs and FPs mentioned

above.

BG Dataset	**Precision**	**Recall**	**AUC**
All Data	0.961	0.991	0.9997
Above CLth	0.774	0.935	0.9976
Below CLth	0.999	1	0.9999

Table 5: Comparison between confident and unconfident pulse classification.

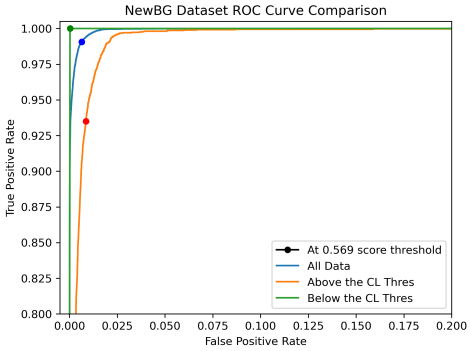


Figure 35: Comparison between the ROC Curves for the confident and unconfident classification - zoomed into the upper left corner.

From the 21626 events in the dataset, 20468 were scored in total agreement with LZap classification ($\approx 94.6\%$), which is similar to the performance on the test subset.

5.3.1.2 DD Dataset

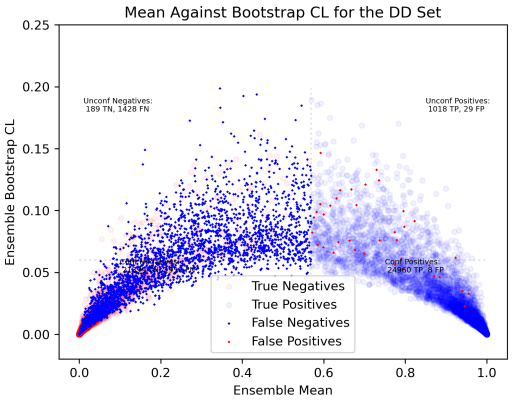
The events in the DD Dataset are obtained during periods of deuterium-deuterium calibration, described in Chapter 4. This type of data is rich in NRs with a much higher ration of multiple scatters against single scatters - and it is not represented during training. The neutron source calibration present in the train dataset is from AmLi simulations which differs significantly from the DD events. In addition, the ratio of ProS2 against NonProS2 is much higher than the train subset (see Table 1). Thus, it is expected that the ensemble performance deteriorates when comparing to the other two datasets.

The ensemble output for the DD Dataset is shown in Figure 36a. Unlike the output for both test and BG datasets, there is a very large number of wrongly scored pulses, specially False Negatives, with low CL intervals.

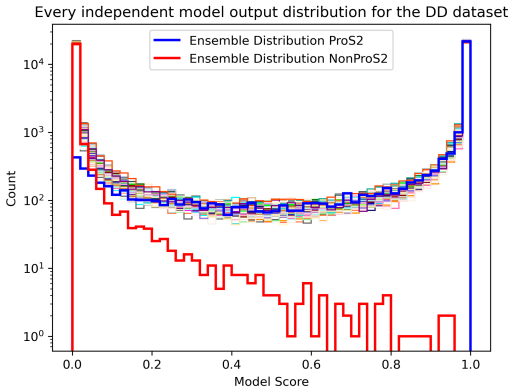
On the other hand, as for the previous datasets, the performance below the CL threshold is significantly better than the performance above the CL threshold; this fact can be observed from the ROC curves on Figure 37. The metrics on Table 6 also support this; only precision did not deteriorate much, due to the lack of FPs.

From the 14529 events in the dataset, 12186 were scored in total agreement with LZap classification ($\approx 83.4\%$). This is lower than the ratio from the previous two datasets, which is expected given the high number of FNs produced by the ensemble.

Refering back to Section 4.3, the ProS2 from DD events are generally shorter and thinner than for the other event sources. Additionally, from Table 1, this dataset also has a much higher ratio of MS to SS



(a) The ensemble output for the DD dataset.



(b) Output distributions for all indepedent models.

Figure 36: ProFinder output on the DD dataset. This output is characterized by a large amount of False Negatives.

DD Dataset	**Precision**	**Recall**	**AUC**
All Data	0.998	0.885	0.9950
Above CLth	0.972	0.416	0.7252
Below CLth	0.9997	0.927	0.9955

Table 6: Comparison between confident and unconfident pulse classification.

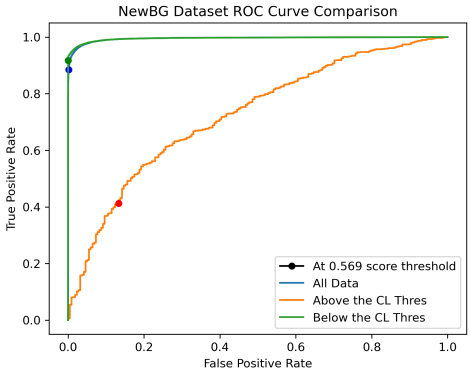


Figure 37: Comparison between the ROC Curves for the confident and unconfident classification.

events. It is likely these two reasons are factors to the high amount of misclassified ProS2 - in Figure 38, that depicts the *pulseLength90* against the *peakAmp* RQ space for the LZap ProS2, it can be observed that the lower peak size region is in fact mainly populated with misclassified ProS2s.

5.3.2 Scatter Type Prediction

The goal of finding the prominent pulses (in the case of this work, the ProS2) is to classify the scatter type of the event. A simplified version of the *InteractionFinder* algorithm is used, where there are only three possible classifications: Single Scatter, Multiple Scatter and Other Scatter, which includes all scatter types except the former two.

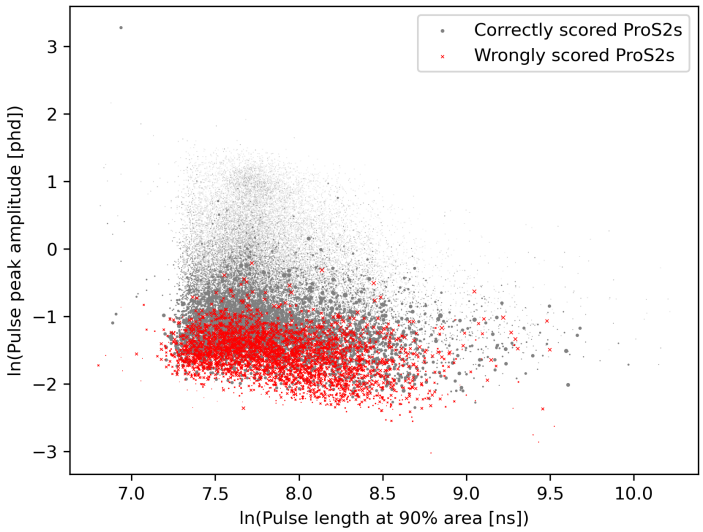


Figure 38: Pulse length at 90% area against the peak amplitude for the prominent S2s in the DD dataset. The size of the marker represents the confidence level - the bigger the marker, the higher is the uncertainty in the pulse score.

- Exclude any ProS2 before the first ProS1 and exclude any ProS1 after the last ProS2;
- Any event has more or less than one ProS1, it is classified as OS;
- Otherwise, if it has one ProS2, it is classified as SS;
- Otherwise, if it has more than one ProS2, it is classified as MS;
- Otherwise, it is classified as OS.

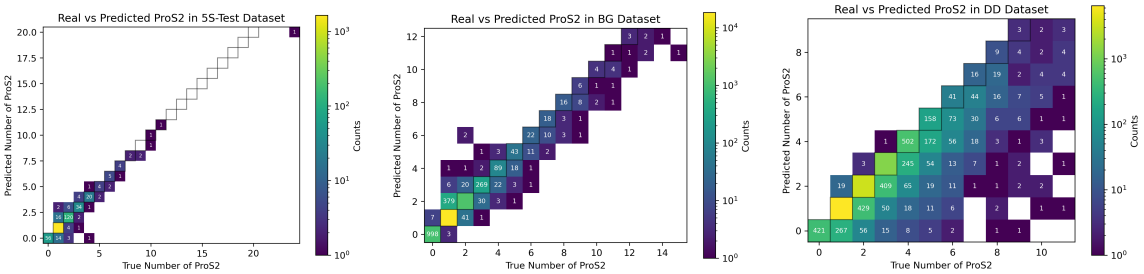
The scatter type prediction confusion matrices are shown in Table 7. It is immediate to check that there are much less wrongly classified scatter types than events in disagreement with LZap. This stems from the fact that in some cases, the loss or addition of one or more ProS2 does not change the scatter type of the event: for example, when a MS event gains one or more ProS2, it is still be classified as a MS (and even when it loses one or more ProS2, but only if there are still 2 or more remaining ProFinder ProS2 in the event).

	True SS	**True MS**	**True OS**
Test Dataset			
Pred SS	1604	4	0
Pred MS	16	195	0

	True SS	True MS	True OS
Pred OS	35	4	102
DD Dataset			
Pred SS	4884	338	2
Pred MS	14	5674	1
Pred OS	272	44	3300
BG Dataset			
Pred SS	18290	41	7
Pred MS	375	1731	0
Pred OS	41	15	1126

Table 7: Scatter confusion matrix for different datasets.

It may be more clear to look at the comparison between the predicted and expected ProS2 in an event (Figure 39). As expected, the distribution for the test subset is the closest to a diagonal line while for the BG Dataset, the distribution forms a thicker diagonal line, implying most events have similar number of true ProS2 and predicted ProS2. On the other hand, for the DD Dataset, the lower triangle below the diagonal is almost filled, showing that many events lost many ProS2 (this is in accordance to the high number of False Negatives found in the analysis of this dataset).



(a) Test subset. (b) BG dataset. (c) DD dataset.

Figure 39: Expected number of ProS2 in an event against the predicted number of ProS2 in the same event (after the first ProS1, if there is one). The square contour represents the diagonal - events with the same number of expected and predicted ProS2. It is important to note that does not imply that the event as fully correct - it may be the case that ProFinder predicts a LZap ProS2 as a NonProS2 and a LZap NonProS2 as a ProS2.

This is expected: as stated before, there is a significant (scatter) class disparity between the training dataset and the DD Dataset. Additionaly, as the DD is composed from NR events, the ProS2 tend to be smaller than a big portion of the training set.

5.4 Commentary and Further Work

The main limitation of the approach followed here was data availability: the more diverse dataset, the 5S Dataset, is also the smallest. Attempts were made to develop a model trained either on DD and BG datasets but, although those model performed well on the test subset of the data, their performance deteriorated significantly (although in different ways) when evaluated against the other datasets. A bigger, more diverse dataset (for example, including samples from deuterium-deuterium calibration) may reduce model variability (thus reducing confidence intervals) and improve overall classification performance.

Even with these limitations, the model achieved an accuracy of 99.51% for the pulses in the 5S Dataset, 99.32% for the BG Dataset and 93.29% for the DD. But, as mentioned before, the ultimate goal of the model is to identify the type of scatter, and the loss or gain of some pulses does not always affect scatter classification. For this task, in addition to the decision tree described above, the ensemble achieves an accuracy of $\approx$97% in the scatter classification for the 5S Dataset, $\approx$98% for the BG Dataset and $\approx$95% for the DD Dataset. Further analysis in simulated data may allow for a proper efficiency comparison with *PulseHighlighter*.

The performance on the DD Dataset is specially noteworthy, given the very different nature of the result, with a high presence of False Negatives. As stated above (and shown in Chapter 4), this dataset contains events of a very different nature from most of the events in the other two datasets. Therefore, one possible path to further develop the concept here described would be the development of separate models for "low energy" and "high energy" events. This distinction can be easily achieved by the total intergrated area of the event: given that S2s are the major contributor to the event area, this separation would effectively distinguish between events with large and events with small S2s. The one exception would be events with such high amounts of noise that the integrated area could be similar to events with large S2s and low noise - but these events are easily identifiable and of low importance; therefore, they could be safely excluded from analysis.

Another possible venue for exploration lies in utilizing the full event instead of only S1s and S2s, as most relevant information of an event is contained in the those two pulses. This was done, as mentioned above, due to previous expertise in the development of an LSTM based prominence identifier. To introduce these other pulses in the training, and avoid losing information, other typical RNN may be tried - the transformer [64], which gained popularity as the base for Large Language Models [65], may be a viable option for a full event training with the same panel data structure. Other time-series deep learning methods outside the Recurrent framework may also be explored, such as Temporal Convolutional Networks [66].

5.5 Summary

The problem of prominence is stated and the data treatment is described in Section 5.1. The methodology is established and the details of the models are presented (Section 5.2.1). The results are presented in Section 5.3, divided by the results on the test subset and on the ensemble performance on the unseen datasets (Section 5.3.1). The results from the ensemble are used to classify the scatter type of the events in Section 5.3.2. Finally, a brief commentary complemented by possible development venues are suggested in Section 5.4.

Chapter 6

Waveform-based Scatter Classification

As already discussed in Section 2.1, LZap takes the raw waveforms as input and returns a number of RQs. In the previous chapter, a Machine Learning approach was developed to find the Prominence of the S2 pulses in the event. The number of ProS2, and their relation to the ProS2 allowed for the classification of the event scatters as SS, MS and OS. In the grand scheme of LZap would only be a module in the *InteractionDetection* processing chain - more precisely, a possible pathway for an alternative Interaction Finder. In this chapter, a proof of concept of a more LZap-independent approach is developed. It is not totally independent since LZap pulse classification is used in some cuts (discussed below) and, more importantly, it relies on LZap scatter classification as its target.

The work developed in this chapter was accomplished utilizing the same resources as the ones described in Chapter 5. The storage and power limitations are more relevant in this case, given the memory intensive nature of the waveform representation of the data. For this reason, the methodology followed in this part of the work is simpler than the one followed in the work from Chapter 5. These limitation also forced some concessions to reduce data dimensionality, which will be described in the following discussion.

With all these limitations, the work presented below does not aim to show the creation of a complete (or close to complete) scatter classification model. Instead, the work is meant to be interpreted as a proof of concept of a direct, mostly LZap independent, method to determine the scatter type of a certain event - a innovative approach in the context of the LZ Experiment. cc

6.1 The dataset

The dataset used is the 5S Dataset. Instead of the LZap RQs, the calibrated summed waveforms from the TPC top plus bottom PMT arrays were used - in terms of the LZap data flow, the waveforms are the output of the *PODSummer* called *SummedPODs* (see Figure 9). The waveforms were extracted

in the form of timestamped PODs. The time intervals between the PODs were padded at a rate of 100 MHz, to match the POD sampling rate. The time interval between the start of the event (-2 ms) and the first POD was padded at the same rate, as well as the time interval between the end time of the last POD and the end of the event (3 ms, 0.5 ms more than the true event final). No other dataset was used due to storage limitations, therefore, there isn't stress-testing results to evaluate the model performance.

The dataset wasn't used in its entirety: to reduce the number of events used in training (due to computation power limitations), some quality cuts were applied to the data: events with more than 40 S2s (S2 cut), more than 1200 Single Photoelectron (SPE)s (SPE cut) and more than 45 SEs (SE cut) are cut. Additionally, only globally triggered events were included (trigger type - TT - cut). Out of the 5880 events, 2633 are cut: therefore, training, validation and testing will be done with 3241 events. A more in-depth analysis of the cuts is discussed in Appendix C.

To further reduce the event dimensionality, all samples past 1 ms after the trigger are excluded (this is done for each event), reducing the dimension of each event from 500 000 to 300 000. This decision is done in conjunction with the trigger type cut: a global trigger can happen for either the S1 or S2 of the event: if the data collection is triggered by an S2, this transformation does not, in principle, lose relevant information; if the data collection is triggered by an S1, the transformation still is able to capture the S2, since the maximum drift time for a TPC event is ≈ 1 ms. For non SS and MS events, it is still possible to lose a ProS2 in this transformation (as shown in Figure 40). This happens in 48 events that are excluded from the dataset (Timing Cut). The effect of each individual cut on the data are summarized on Table 8

Scatter	S2 Cut	SPE Cut	SE Cut	TT Cut	Timing Cut	Total Cut	Remaining
SS	870	62	1674	397	148	2259	2699
MS	17	27	19	217	59	244	399
OS	28	48	85	62	12	136	217
Total	915	137	1778	676	219	2687	3193

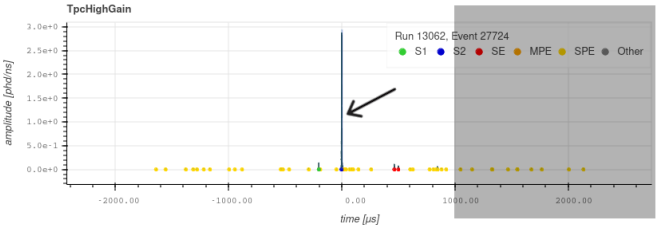
Table 8: Number of events that did not survive the cut, divided between scatter types. The total column does not represent the summation of the previous row, since some events do not pass two or more of the established cuts.

After the removal of one corrupted (SS) event, the number of events in the dataset is finally set at 3192 (83.7% SS, 11.9% MS and 4.4% OS).

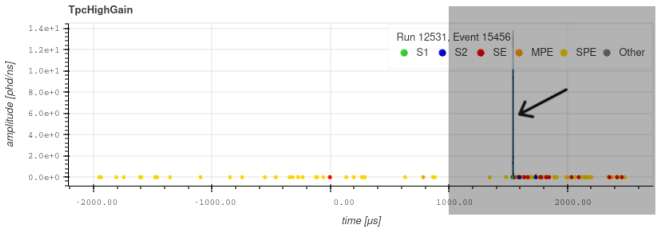
For a given event, there's two different relevant sample amplitude scales - the S2 scale, and the S1 scale. To mitigate this difference, the waveforms, wf, are transformed according to Equation 24.

$$\mathrm{wf} \rightarrow \ln\left(\mathrm{wf} + 1\right) \tag{24}$$

To reduce data dimensionality even further (in order to reduce the processing power needed to



(a) Single scatter event with a central ProS2.



(b) Single scatter event with a late ProS2.

Figure 40: Most of the events in the dataset have their ProS2(s) before the 1 ms time point (a full drift time for the LZ TPC) - similar to the upper event -, but there are 48 events in which the time interval cut excludes their ProS2(s) from the data. Arrows point to the ProS2 in both waveforms

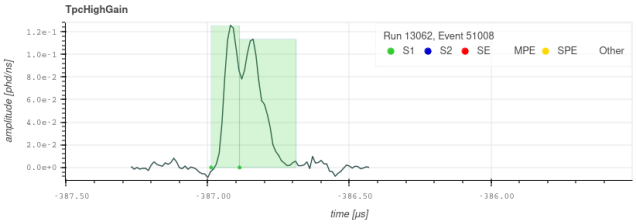
train the model), max pooling (with pool size and stride of 5) is applied to the waveforms to reduce data dimensionality - the waveform used for training, validation or testing becomes a 1D array with size $300000/5 = 60000$. This pooling operation emphasizes the general characteristics of the pulses while smoothing out less significant features - for large pulses, such as typical ProS2s, will not be strongly affected. Meanwhile, ProS1s tend to be shorter and pulse details can be removed from data. In Figure 41, a short ProS1, followed very closely by a second S1 is depicted before and after pooling. It can be seen that the pooling process may de-characterize the pulse(s) in some degree - assuming that the LZap distinction is correct.

The total effect of the transformations described above is depicted in Figure 42 for a median length and amplitude ProS2, as per LZap classification.

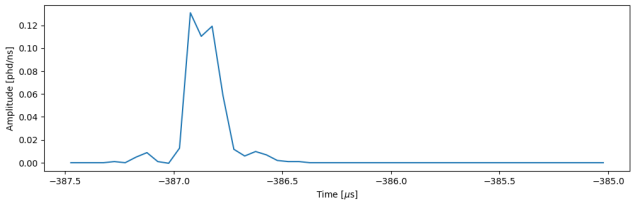
The dataset is divided in two classes: SS and NonSS. A value is attributed for each event: 1 if the event is a Single Scatter (SS), 0 otherwise. Due to class imbalance, a weight is associated to each event according to its class:

$$w_{\text{class}} = \left(\frac{1}{N_{\text{class}}}\right)\left(\frac{N}{\epsilon}\right) \Longrightarrow \begin{cases} w_{\text{SS}} = 0.298 \\ w_{\text{NonSS}} = 1.537 \end{cases} \tag{25}$$

where N is the number of events in the dataset, N_{class} is the number of events of a certain class in the dataset and ϵ is a parameter, set to $\epsilon = 4$.

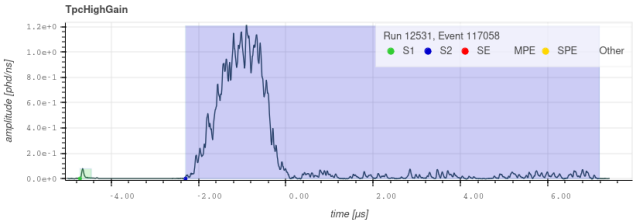


(a) Short ProS1 followed by a second S1 in LZ Event Viewer.

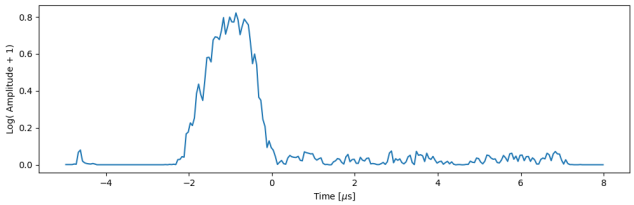


(b) Short ProS1 followed by a second S1 after pooling.

Figure 41: Effect of the pooling in a thinner ProS1 (relative to the rest of the dataset) followed by another, non-prominent, S1. In the Event Viewer (left), it can be seen that LZap divided the pod in two S1 pulses. After pooling, the shape is less clear and it approximates to the shape of only one pulse.



(a) Portion of a raw waveform on Event Viewer.



(b) Portion of the transformed waveform.

Figure 42: A small ProS1 followed by an average ProS2. The pooling clearly smooths the waveform, while the logarithmic transformation reduces the scale difference between both pulses. Note that the noise after the ProS2 is also affected by the compression.

The dataset is equally and randomly [1] divided in three subsets of 1064 events.

[1] The separation was random but there was care to ensure similar quantities of each scatter type.

6.2 Model Architecture

The model's goal is to process the waveform and return a value that can be interpreted as a confidence score for the given event being a Single Scatter (SS). The waveform is a time series, therefore the method of choice should be able to find temporal correlations; the waveform is large, so the method should be efficient at dealing with large swats of data to reduce the required computing power.

The method of choice is a mixed Conv1D and Dense layers architecture, including the typical Pooling layers from traditional convolutional models and Dropout layers to reduce overfitting. The loss of choice is the MSE loss.

Motivated by the different time (number of samples) scales between S1s and S2s, a branched architecture is used for the initial processing of the waveform, using Conv1Ds and Pooling layers whose sizes match the time scales of the two main pulse types, before concatenating the branch outputs into a main branch, composed by Dense Layers. The last layer of the model is fixed: a Dense layer with one neuron and a sigmoid activation function - the model will output one value between 0 and 1 - the model is trained to identify SS events with 1 and NonSS events with 0.

The first non-Input layers are the first layers of each branch - One-dimensional Convolutional layer (Conv1D)s with fixed filter size: 8 and 128. These values correspond to 40 and 640 sample values in the original waveform (0.4 μs and 6.4 μs in time units, respectively), and are the powers of two that approximate the mean pulse length for ProS1 and ProS2 respectively, as shown in Figure 43.

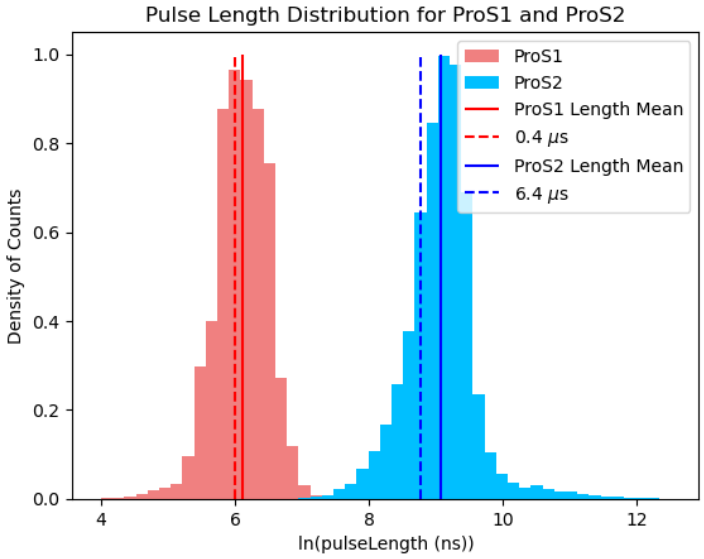


Figure 43: Pulse length distribution for ProS1s (peak on the left) and ProS2s (peak on the right). The filter size was chosen according to the power of two closer to the mean for each distribution.

The specific number of layers in each branch is an optimizable hyperparameter - with a minimum

of two Conv1D-Pooling layer pairs. The same is true for the number of Dense layers in the main branch, with a minimum of two, including the output sigmoid layer. The full list of optimizable hyperparameters is shown in Table 9, including the hyperparameter search space.

Hyperparameter	**Type/Values**	**Range/Value**	**Step**	**Chosen Value**
`Optimizer`	Constant	-	-	Adam
`Loss`	Constant	-	-	MSE
`Epochs`	Constant	-	-	200
`Patience`	Constant	-	-	20
`Learning Rate`	Float	0.00005 to 0.01	-	0.002472
`Decay` ($\times 10^2$)	Integer	1 to 100	1	66
`Dropout Rate`	Float	0.02 to 0.50	-	0.3411
`Number of Batches`	Integer	4 to 40	2	20
`S2 Branch Filter 1`	Constant	-	-	128
`S2 Branch Filter 2`	Integer	4 to 64	2	36
`S2 Branch Filter 3`	Integer	4 to 32	2	14
`S2 Branch Filter 4`	Integer	4 to 16	2	14
`S1 Branch Filter 1`	Constant	-	-	8
`S1 Branch Filter 2`	Integer	4 to 64	2	10
`S1 Branch Filter 3`	Integer	4 to 32	2	18
`S1 Branch Filter 4`	Integer	4 to 16	2	16
`Pool Size 1`	Integer	2 to 24	2	24
`Pool Size 2`	Integer	2 to 12	2	10
`Pool Size 3`	Integer	1 to 8	1	7
`Pool Size 4`	Integer	1 to 4	1	2
`Neurons 1`	Integer	16 to 256	4	200
`Neurons 2`	Integer	8 to 128	4	28
`Neurons 3`	Integer	4 to 128	4	88
`Neurons 4`	Integer	4 to 64	4	36[2]
`Activation Function`	Categorical	{relu, selu, tanh}	-	tanh
`Additional 1DCNN`	Integer	0 to 1	1	1
`Additional Dense`	Integer	1 to 3	1	2

[2] Not used due to only 3 Dense Layers

Hyperparameter	Type/Values	Range/Value	Step	Chosen Value

Table 9: Model hyperparameters and respective search spaces, if optimizable, including step (N/A for continuous or categorical variables). In the Chosen Value column, the value chosen for each variable after the Optuna study, according to the best trial. Constant hyperparameters are fixed, and therefore are not chosen by Optuna.

The Optuna study ran for 200 trials, with the goal of minimizing the validation dataset loss. After the hyper-optimization, the hyperparameters of the best trial were chosen (also shown on Table 9). Details about the hyperparameters and about the trials can be found on Appendix D.2. A diagram of the model architecture is depicted on Figure 44.

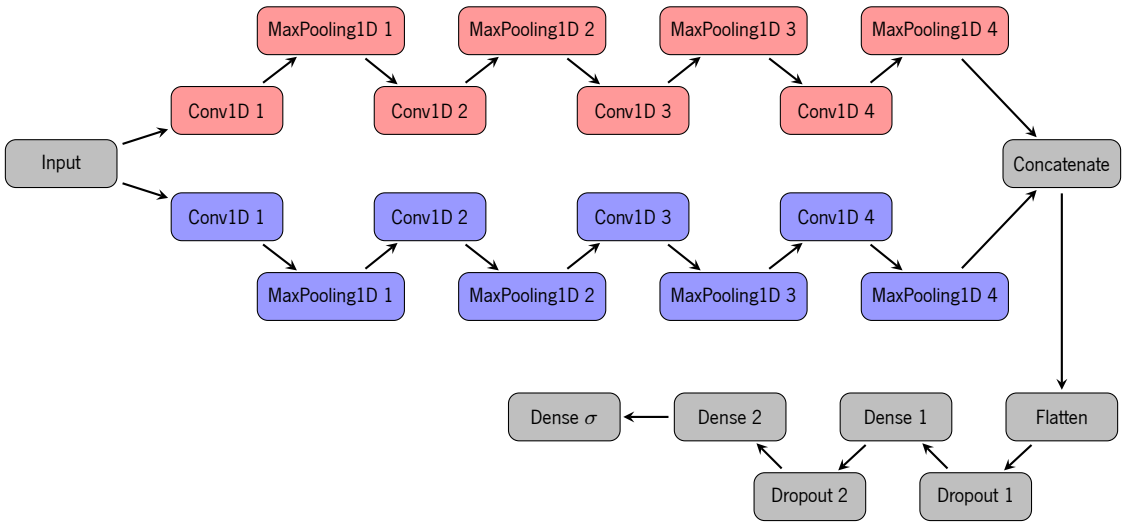


Figure 44: Waveform classification model best architecture. In red and blue, the Conv1D branches. In black, the main, unified branch.

6.3 Model Training and Testing

The ensemble approach from the ProFinder (described in Section 5.2) is not pursued due to the significantly higher computational cost associated with a raw waveform analysis. The best trial parameters (shown in Table 9) are picked, and a single model is trained on the training subset. The model was set to be trained during 200 epochs with a Early Stopping patience of 20 epochs based on the evolution of the validation loss - the model trained for 58 epochs before stopping and the training progress is shown in Figure 45 (with accuracy chosen as a simple metric to show model progression).

The chosen threshold to classify an event as a SS or a NonSS is the threshold value that results in the highest F-Score for this task, using the events from the validation dataset. For this model and

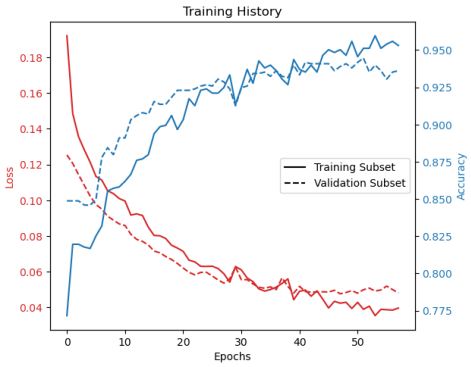
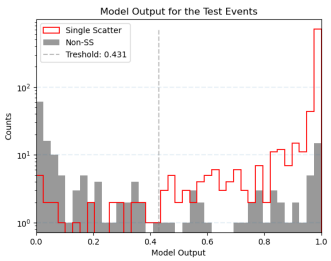


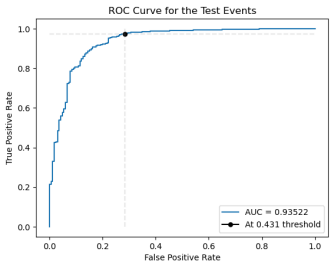
Figure 45: Training history of the model. The training loss/metric does not stabilizes. Therefore, the continued training could result in a strongly overfitted model.

the results presented below, the threshold is set at 0.431 with a validation F-Score of 0.967. Recalling the scatter classification matrix from the ProFinder model (Table 7), the F-Score for the SS classification can be computed, obtaining a value of 0.983, which is slightly better than the waveform model. This implies that even if the performance is worse, the waveform method is still comparable to the RQ-based approach of the ProFinder.

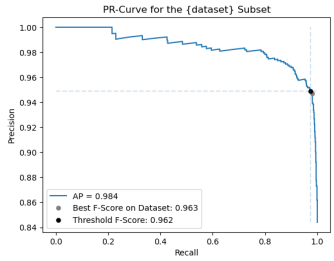
The testing subset is processed by the trained model returning a score between 0 and 1 for every event, interpreted as a classification of an event as a SS event (score is close to 1) or not (score is close to 0). The output distribution (Figure 46a) shows a very polarized distribution, indicating a high confidence model. This polarization (910 out of 1046 events within 0.1 range from the distribution limits) may be caused by the class imbalance. The ROC (Figure 46b) and the precision-recall curves (Figure 46c) of the model reflect the low number of events used in training, but their shape indicates that the model is a good discriminator. In the latter curve, it can be seen that the best threshold obtained as described above has a similar F-Score to the best threshold computed specifically for the test dataset.



(a) Model output distribution. Note that the y-axis logarithmic scale.



(b) Model's ROC Curve and AUC.



(c) Model's PR Curve, F-Score, and AP. Note the y-axis range.

Figure 46: Performance evaluation of the model through various metrics.

The confusion matrix of the model, showing Other and Multi Scatters separately, at the established

threshold is shown in Table 10. The data shows that the model tends to discriminate correctly between Single and NonSingle Scatters, except for Other Scatters, where more than half were scored above the defined threshold - as already mentioned before (Chapter 5), the OS events are a group of distinct types of scatter, with different topologies. As the aggregated class of Other Scatter is already underrepresented, the specific topologies must be even more underrepresented.

	True SS	True NonSS	
		Other	Multi
Predicted SS	875	25	22
Predicted NonSS	23	22	98

Table 10: Extended confusion matrix, dividing the NonSS in OS and Multiple Scatter (MS).

Comparing this confusion matrix with the scatter confusion matrix obtained through the ProFinder method (Table 7), the results from the waveform approach were slightly less accurate than the ProFinder approach (93.4% against 95.9%[3], respectively). This is expected because, even if the ProFinder was trained and tested on a less clean dataset, it uses higher level and compressed information (the LZap RQs), including, for the scatter classification, the prominence of S1s.

The results shown above paint a promising picture - even with time and resources constraints, the model developed is shown to be a good discriminator, with a higher than 90% accuracy. The results support the viability of a raw-waveform-based scatter classifier.

6.4 Commentary and Further Work

As shown above, some relevant concessions were made to allow the development and training of the deep model that was built to achieve this classification task: the reduction of the dataset and the reduction of the events themselves. The reduction of events in the dataset were made in a equilibrium between selecting and cutting low quality and while avoiding exacerbating the class inbalance already present in the dataset. On the other hand, the reduction of the event, while not ideal, should not impact significantly the interpretation of the model performance. These concessions were necessary due to the aforementioned computational resources limitations. Ideally, in a future development, they should be reduced or even eliminated in order to improve model performance. In addition, and as the work described in the previous chapter, a larger more diverse dataset would improve training.

The pre-processing may also be subject to improvements: the logarithmic compression was done to bring the S1 and S2 scales closer, but it has the side effect of upscaling unwanted structures: a clear

[3]To obtain this accuracy value, the events classified as Other and Multi Scatters are considered as NonSS.

example of this is the transformation of the noisy environment of electron-trains that usually follow the S2 of an event (an example of this environment is shown in Figure 42). A transient detection filter could be employed as part of a pre-processing, allowing for a LZap independent method for discriminating between S1 and some types of noise, such as the S2 tail shown in Figure 42.

Another path for improvement lies in the possible implementation of a more diverse classifier - in this work, the classifier is binary and the score can only relate to the probability of a event being a SS. A possible suggestion to improve the model is to create an ensemble of binary models, each model dedicated to classify the event as a single type of scatter - Single, Multiple or Other (remembering Other is an aggregated class of scatters, a training with a larger dataset could even split the aggregated class into the individual scatter types).

Further works can also explore different architectures and types of NN layers. The inclusion of an LSTM layer proved to be computationally expensive, but it could be experimented in a work with larger resources. A deeper exploration of the hyperparameter space could also improve the performance of the model.

6.5 Summary

The problem of scatter classification using raw waveforms is introduced, and the dataset preparation is detailed, including the application of quality cuts and dimensionality reduction techniques to address computational limitations (Section 6.1). The processing unit is a model that processes the calibrated waveforms directly, utilizing a branched 1D CNN architecture to handle temporal correlations and scale differences between S1 and S2 pulses (Section 6.2). The model's performance is evaluated on a testing subset, demonstrating promising results with high accuracy and F-Score, though slightly lower than the RQ-based ProFinder approach (Section 6.3). The results point to the viability of a waveform-based classifier, with suggestions for future improvements, such as enhanced pre-processing, diverse classification strategies, and exploration of alternative neural network architectures (Section 6.4).

Chapter 7

Closing remarks

The goal of this work was to explore possible ML applications for data analysis tasks in the LUX-ZEPLIN experiment. The specific task was the identification of the type of scatter that originates the event - an important information for effective background rejection and signal identification for a variety of physics searches. Two different approaches were followed, based on Neural Network methods, trained with a variety of datasets from distinct physics sources, from background data to calibrations. One of the approaches was also tested with a source type not included in the training as an additional test to the generalizability of the model.

The first approach, the *ProFinder*, sought to provide an alternative to the *PulseHighlighter* module in the LZap algorithm: it represents the event in a panelled data structure with time ordered rows (each row represents a pulse). A LSTM layer provided the main processing mechanism to achieve this goal - a sequence of values that indicates if an S2 is or isn't prominent. The model achieved accuracies of 99.51% for the testing subset of the training dataset, 99.32% for the BG Dataset dataset and 93.29% for the DD Dataset. After scoring each pulse, a decision tree based on the *InteractionFinder* is used to determine the scatter type of the event. For this task, the model achieves accuracies of 97%, 98% and 95% for the three datasets, respectively.

The performance deterioration observed for the DD Dataset is the most notable result: this is not unexpected due to the intrinsic differences in the nature of the DD events. Recent internal work on the LZap efficiency for these calibration events indicated that the algorithm has a tendency to lose very low energy ProS2. This is likely due to the fact that the LZap was also not calibrated with DD events, since they stray away from the average TPC response during normal operations. It is possible that this bias is being picked up and even amplified by the training, since the model is being trained against the LZap classification. This now identified bias reflects the main challenge of this task - the reliance on LZap makes it possible that the model captures systematic and unwanted, bias from it.

The second approach takes on the event immediately after the *PODSummer* module, and seeks to

directly identify if an event is or isn't a Single Scatter, bypassing the pulse identification analysis completely. This approach relies on a deep network using Conv1D layers. It is a much more computationally demanding task than the previous approach, but it does not rely on all the LZap modules that divide, characterize and classify the pulses. Even with all the constraints necessary to be able to train the model, it was able to achieve an accuracy of 93.4%. It is important to mention again the class imbalance present in the training dataset that could have affected the ability of the module to generalize properly after training. The impossibility of testing the model against unseen datasets (in the timespan of this thesis) prevents stronger conclusions on this aspect of the model.

Despite the challenges outlined above, the results presented in this work demonstrate the potential of machine learning methods for future integration into the LUX-ZEPLIN data analysis framework. Several recommendations for improvement or alternative directions have been discussed throughout the text, most of which fall into two primary categories: increasing the availability of high-quality event data, and reducing dependence on the LZap algorithm. The work conducted in this dissertation was presented and discussed with LZ collaborators specializing in machine learning applications.

To conclude this dissertation, it is important to reiterate that the goal was not to produce complete and ready models, but to explore a small set of improvements to analysis using machine-learning techniques. The datasets were not optimal, both in quantity and quality and the computational resources were limited. Nevertheless, the results are overall positive and show that these techniques (and other similar approaches) may be a viable future option for the experiment data analysis pipeline.

Bibliography

[1] G. Bertone and D. Hooper. History of dark matter. *Reviews of Modern Physics*, 90(4), October 2018. ISSN 1539-0756. doi: 10.1103/revmodphys.90.045002. URL http://dx.doi.org/10.1103/RevModPhys.90.045002.

[2] F. Zwicky trans. by H. Andernach. English and spanish translation of Zwicky's (1933), The redshift of extragalactic nebulae, 2017. URL https://arxiv.org/abs/1711.01693.

[3] G. Bertone et al. Particle dark matter: evidence, candidates and constraints. *Physics Reports*, 405 (5–6):279–390, January 2005. ISSN 0370-1573. doi: 10.1016/j.physrep.2004.08.031. URL http://dx.doi.org/10.1016/j.physrep.2004.08.031.

[4] Planck Colaboration. Planck2018 results VI. Cosmological parameters. *Astronomy & Astrophysics*, 641:A6, September 2020. ISSN 1432-0746. doi: 10.1051/0004-6361/201833910. URL http://dx.doi.org/10.1051/0004-6361/201833910.

[5] D. Baxter et al. Recommended conventions for reporting results from direct dark matter searches. *The European Physical Journal C*, 81(10), October 2021. ISSN 1434-6052. doi: 10.1140/epjc/s10052-021-09655-y. URL http:/M.Ackermannandother/dx.doi.org/10.1140/epjc/s10052-021-09655-y.

[6] T. Clifton et al. Modified gravity and cosmology. *Physics Reports*, 513(1–3):1–189, March 2012. ISSN 0370-1573. doi: 10.1016/j.physrep.2012.01.001. URL http://dx.doi.org/10.1016/j.physrep.2012.01.001.

[7] H. Davoudiasl and M. Schnubel. Bringing Peccei-Quinn mechanism down to earth, 2024. URL https://arxiv.org/abs/2406.19455.

[8] A. Bottino et al. Interpreting the recent results on direct searches for dark matter particles in terms

of relic neutralinos. *Physical Review D*, 78(8), October 2008. ISSN 1550-2368. doi: 10.1103/physrevd.78.083520. URL http://dx.doi.org/10.1103/PhysRevD.78.083520.

[9] K. Garrett and G. Duda. Dark matter: A primer. *Advances in Astronomy*, 2011:1–22, 2011. ISSN 1687-7977. doi: 10.1155/2011/968283. URL http://dx.doi.org/10.1155/2011/968283.

[10] J. Feng. Dark matter candidates from particle physics and methods of detection. *Annual Review of Astronomy and Astrophysics*, 48(1):495–545, August 2010. ISSN 1545-4282. doi: 10.1146/annurev-astro-082708-101659. URL http://dx.doi.org/10.1146/annurev-astro-082708-101659.

[11] M. Peskin. Dark matter and particle physics. *Journal of the Physical Society of Japan*, 76(11): 111017, November 2007. ISSN 1347-4073. doi: 10.1143/jpsj.76.111017. URL http://dx.doi.org/10.1143/JPSJ.76.111017.

[12] J. Feng. The WIMP paradigm: Theme and variations. *SciPost Physics Lecture Notes*, June 2023. ISSN 2590-1990. doi: 10.21468/scipostphyslectnotes.71. URL http://dx.doi.org/10.21468/SciPostPhysLectNotes.71.

[13] D. Hooper and E. Baltz. Strategies for determining the nature of dark matter. *Annual Review of Nuclear and Particle Science*, 58(1):293–314, November 2008. ISSN 1545-4134. doi: 10.1146/annurev.nucl.58.110707.171217. URL http://dx.doi.org/10.1146/annurev.nucl.58.110707.171217.

[14] The Fermi-LAT Collaboration. Searching for dark matter annihilation from milky way dwarf spheroidal galaxies with six years of Fermi Large Area Telescope data. *Phys. Rev. Lett.*, 115: 231301, Nov 2015. doi: 10.1103/PhysRevLett.115.231301. URL https://link.aps.org/doi/10.1103/PhysRevLett.115.231301.

[15] C. Koun and J. Park. New search for dark matter with neutrons at neutrino detectors, 2024. URL https://arxiv.org/abs/2409.05646.

[16] P. Baratella et al. PPPC 4 DMν: a poor particle physicist cookbook for neutrinos from dark matter annihilations in the sun. *Journal of Cosmology and Astroparticle Physics*, 2014(03):053–053, March 2014. ISSN 1475-7516. doi: 10.1088/1475-7516/2014/03/053. URL http://dx.doi.org/10.1088/1475-7516/2014/03/053.

[17] ATLAS Colaboration. Search for new particles in events with a hadronically decaying W or Z boson and large missing transverse momentum at $\sqrt{s}$ = 13 TeV using the ATLAS detector. *Journal of High*

Energy Physics, 2024(11), November 2024. ISSN 1029-8479. doi: 10.1007/jhep11(2024)126. URL http://dx.doi.org/10.1007/JHEP11(2024)126.

[18] LZ Colaboration. The LUX-ZEPLIN (LZ) experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 953: 163047, February 2020. ISSN 0168-9002. doi: 10.1016/j.nima.2019.163047. URL http://dx.doi.org/10.1016/j.nima.2019.163047.

[19] D. Nygren. Origin and development of the TPC idea. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 907:22–30, 2018. ISSN 0168-9002. doi: https://doi.org/10.1016/j.nima.2018.07.015. URL https://www.sciencedirect.com/science/article/pii/S0168900218308477. Advances in Instrumentation and Experimental Methods (Special Issue in Honour of Kai Siegbahn).

[20] D. Nygren. The time projection chamber: A new 4 pi detector for charged particles. *eConf*, C740805:58, 1974.

[21] LZ Colaboration. LUX-ZEPLIN (LZ) technical design report, 2017. URL https://arxiv.org/abs/1703.09144.

[22] V. Chepel and H. Araújo. Liquid noble gas detectors for low energy particle physics. *Journal of Instrumentation*, 8(04):R04001, apr 2013. doi: 10.1088/1748-0221/8/04/R04001. URL https://dx.doi.org/10.1088/1748-0221/8/04/R04001.

[23] E. Aprile et al. *Noble Fluids as Detector Media*, chapter 2, pages 7–32. João Wiley & Sons, Ltd, 2006. ISBN 9783527610020. doi: https://doi.org/10.1002/9783527610020.ch2. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527610020.ch2.

[24] E. Aprile and T. Doke. Liquid xenon detectors for particle physics and astrophysics. *Rev. Mod. Phys.*, 82:2053–2097, Jul 2010. doi: 10.1103/RevModPhys.82.2053. URL https://link.aps.org/doi/10.1103/RevModPhys.82.2053.

[25] L. Baudis. Dual-phase xenon time projection chambers for rare-event searches, 2024. URL https://arxiv.org/abs/2311.05320.

[26] LZ Colaboration. First dark matter search results from the LUX-ZEPLIN (LZ) experiment. *Physical Review Letters*, 131(4), July 2023. ISSN 1079-7114. doi: 10.1103/physrevlett.131.041002. URL http://dx.doi.org/10.1103/PhysRevLett.131.041002.

[27] LZ Colaboration. Dark matter search results from 4.2 tonne-years of exposure of the LUX-ZEPLIN (LZ) experiment, 2024. URL https://arxiv.org/abs/2410.17036.

[28] R. Linehan et al. Design and production of the high voltage electrode grids and electron extraction region for the LZ dual-phase xenon time projection chamber. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1031:165955, 2022. ISSN 0168-9002. doi: https://doi.org/10.1016/j.nima.2021.165955. URL https://www.sciencedirect.com/science/article/pii/S0168900221009062.

[29] A. Hitachi. Properties of liquid xenon scintillation for dark matter searches. *Astroparticle Physics*, 24(3):247–256, 2005. ISSN 0927-6505. doi: https://doi.org/10.1016/j.astropartphys.2005.07.002. URL https://www.sciencedirect.com/science/article/pii/S0927650505000964.

[30] B. Paredes et al. Response of photomultiplier tubes to xenon scintillation light. *Astroparticle Physics*, 102:56–66, 2018. ISSN 0927-6505. doi: https://doi.org/10.1016/j.astropartphys.2018.04.006. URL https://www.sciencedirect.com/science/article/pii/S0927650518300173.

[31] H. Peiser et al. Element by element review of their atomic weights. *Pure and Applied Chemistry*, 56(6):695–768, 1984. doi: doi:10.1351/pac198456060695. URL https://doi.org/10.1351/pac198456060695.

[32] A. Gando et al. Precision analysis of the two-neutrino spectrum in KamLAND-Zen and its impact on the quenching of nuclear matrix elements. *Physical Review Letters*, 122(19), May 2019. ISSN 1079-7114. doi: 10.1103/physrevlett.122.192501. URL http://dx.doi.org/10.1103/PhysRevLett.122.192501.

[33] LZ Colaboration. Projected sensitivity of the LUX-ZEPLIN experiment to the $0\nu\beta\beta$ decay of ^{136}Xe. *Phys. Rev. C*, 102:014602, Jul 2020. doi: 10.1103/PhysRevC.102.014602. URL https://link.aps.org/doi/10.1103/PhysRevC.102.014602.

[34] LZ Colaboration. Projected sensitivity of the LUX-ZEPLIN experiment to the two-neutrino and neutrinoless double β decays of ^{134}Xe. *Phys. Rev. C*, 104:065501, Dec 2021. doi: 10.1103/PhysRevC.104.065501. URL https://link.aps.org/doi/10.1103/PhysRevC.104.065501.

[35] N. Barros et al. Double beta decay searches of 134Xe, 126Xe and 124Xe with large scale Xe detectors. *Journal of Physics G: Nuclear and Particle Physics*, 41(11):115105, sep 2014. doi: 10.1088/0954-3899/41/11/115105. URL https://dx.doi.org/10.1088/0954-3899/41/11/115105.

[36] KamLAND-Zen Colaboration. Search for majorana neutrinos with the complete KamLAND-Zen dataset, 2024. URL https://arxiv.org/abs/2406.11438.

[37] C. Giuntiand and C. Kim. Massive neutrinos. In *Fundamentals of Neutrino Physics and Astrophysics*. Oxford University Press, 03 2007. ISBN 9780198508717. doi: 10.1093/acprof:oso/9780198508717.003.0006. URL https://doi.org/10.1093/acprof:oso/9780198508717.003.0006.

[38] LZ Colaboration. The data acquisition system of the LZ dark matter detector: Fadr. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1068:169712, 2024. ISSN 0168-9002. doi: https://doi.org/10.1016/j.nima.2024.169712. URL https://www.sciencedirect.com/science/article/pii/S0168900224006387.

[39] P. Braz. *Sensitivity to the $0\nu\beta\beta$ decay of ^{136}Xe and development Machine Learning tools for pulse classification for the LUX-ZEPLIN experiment*. PhD thesis, Universidade de Coimbra, Feb 2021. URL https://hdl.handle.net/10316/95274.

[40] LZ Collaboration. Lzap. https://gitlab.com/luxzeplin/lzap/LZap/, 2020. Private GitLab repository for LZap development.

[41] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach, US Edition*. Pearson Education, 4 edition, 2021. ISBN 9781292401171.

[42] C. Aggarwal. *Neural Networks and Deep Learning*. Springer, 2 edition, 2023. ISBN 978-3-031-29641-3.

[43] S. Ahmed et al. On closures for reduced order models — a spectrum of first-principle to machine-learned avenues. 06 2021. doi: 10.48550/arXiv.2106.14954.

[44] D. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[45] Y. Bengio et al. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181.

[46] S. Hochreiter et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. *A Field Guide to Dynamical Recurrent Neural Networks*, 03 2003.

[47] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[48] T. Tayeh et al. An attention-based ConvLSTM autoencoder with dynamic thresholding for unsupervised anomaly detection in multivariate time series. *Machine Learning and Knowledge Extraction*, 4:350–370, 04 2022. doi: 10.3390/make4020015.

[49]

[50] S. Kiranyaz et al. 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398, 2021. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2020.107398. URL https://www.sciencedirect.com/science/article/pii/S0888327020307846.

[51] S. Kiranyaz et al. Convolutional neural networks for patient-specific ECG classification. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2608–2611, 2015. doi: 10.1109/EMBC.2015.7318926.

[52] S. Kiranyaz et al. 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151, 04 2021. doi: 10.1016/j.ymssp.2020.107398.

[53] T. Agrawal. *Hyperparameter Optimization Using Scikit-Learn*, pages 31–52. Apress, Berkeley, CA, 2021. ISBN 978-1-4842-6579-6. doi: 10.1007/978-1-4842-6579-6_1. URL https://doi.org/10.1007/978-1-4842-6579-6_1.

[54] T. Agrawal. *Bayesian Optimization*, pages 81–108. Apress, Berkeley, CA, 2021. ISBN 978-1-4842-6579-6. doi: 10.1007/978-1-4842-6579-6_1. URL https://doi.org/10.1007/978-1-4842-6579-6_1.

[55] T. Akiba et al. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[56] LZ Colaboration. The design, implementation, and performance of the LZ calibration systems. *Journal of Instrumentation*, 19(08):P08027, August 2024. ISSN 1748-0221. doi: 10.1088/1748-0221/19/08/p08027. URL http://dx.doi.org/10.1088/1748-0221/19/08/P08027.

[57] A. Sazzad et al. Design and characterization of AmLi neutron sources for the LZ experiment. *Journal of Instrumentation*, 18(05):P05006, May 2023. ISSN 1748-0221. doi: 10.1088/1748-0221/18/05/p05006. URL http://dx.doi.org/10.1088/1748-0221/18/05/P05006.

[58] H. Ito et al. Analyzing the neutron and γ-ray emission properties of an americium–beryllium tagged neutron source. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators,*

Spectrometers, Detectors and Associated Equipment, 1057:168701, 2023. ISSN 0168-9002. doi: https://doi.org/10.1016/j.nima.2023.168701. URL https://www.sciencedirect.com/science/article/pii/S0168900223006927.

[59] B. Efron. *Bootstrap Methods: Another Look at the Jackknife*, pages 569–593. Springer New York, New York, NY, 1992. ISBN 978-1-4612-4380-9. doi: 10.1007/978-1-4612-4380-9_41. URL https://doi.org/10.1007/978-1-4612-4380-9_41.

[60] F. Chollet et al. LSTM layer, . URL https://keras.io/api/layers/recurrent_layers/lstm. Accessed on 11/10/2024.

[61] F. Chollet et al. Dense layer, . URL https://keras.io/api/layers/layers/core_layers/dense. Accessed on 11/10/2024.

[62] F. Chollet et al. EarlyStopping, . URL https://keras.io/api/callbacks/early_stopping/. Accessed on 11/10/2024.

[63] F. Chollet et al. EarlyStopping, . URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gaussian_kde.html. Accessed on 56/10/2024.

[64] A. Vaswani et al. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[65] A. Xavier et al. Transformer models: an introduction and catalog, 2024. URL https://arxiv.org/abs/2302.07730.

[66] S. Bai et al. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018. URL https://arxiv.org/abs/1803.01271.

[67] LZ Colaboration. Rqs. URL https://luxzeplin.gitlab.io/docs/softwaredocs/analysis/analysiswithrqs/rqlist.html.

Appendix A

Reduced Quantities Defined

The Reduced Quantities are values that characterize the values derived from raw waveform data from LZ that characterize the event and their signals (pulses). The type of RQs can be broadly divided in three categories: event-level RQs, that describe the event (for example, the EventHeader Tree contains the labeling of the event and the details about the data acquisition); pulse-level RQs characterize the pulses either from a specific channel or detector (for example, the pulsesTPC Tree contains the characteristics of the signal as recorded by the mixed gain mode of the TPC); and interaction-level RQs, containing the relevant information describing the interaction that originated each event (these are stored in the Scatter tree).

RQ	Unit	Description
pulsesTPC Tree		
pulseStartTime_ns	[ns]	Start time of pulse relative to event trigger time
pulseEndTime_ns	[ns]	End time of pulse relative to event trigger time
pulseLength*	[ns]	Pulse duration (given by pulseEndTime_ns - pulseStartTime_ns)
peakAmp	[phd/sample]	Max amplitude of summed pod within pulse
pulseArea_phd	[phd]	Total area of summed pod from pulse start to end

RQ	Unit	Description
areaFractionTimeN_ns (aftN)	[ns]	Time at which summed pod reaches N% of total area; ns relative to pulse start
pulseLength90*	[ns]	Pulse duration between 5% and 95% area (given by aft95 - aft5)
pulseLength50*	[ns]	Pulse duration between 25% and 75% area (given by aft75 - aft25)
peak2length*	[(phd/sample)/ns]	Relation between maxumum amplitude and length of the pulse (given by peakAmp/pulseLength90)
promptFraction50ns	[adim]	Fraction of summed pod area in 50 ns window at start of pulse relative to total pulse area; 50 ns window defined from 10 ns before to 40 ns after aft5
coincidence	[count]	Number of channels that had non-zero contribution to pulse
peakTime_ns	[ns]	Time at which summed pod reaches maximum within pulse; ns relative to pulse start
peakPos*	[adim]	Relative position of the peak within the pulse (given by peakTime_ns / pulseLength90)
pulseID	[adim]	Identification of the pulse within the event. The first pulse is labeled 0.
classification	[adim]	HADES pulse classification
prominentS2PulseIDs	[adim]	ID of the Prominent S2 pulses
prominenceS2	[adim]	Binary value: 1 for pulses with IDs in prominentS2PulseIDs, 0 for the remaining pulses
prominentS1PulseIDs	[adim]	ID of the Prominent S1 pulses
EventHeader Tree		
runID	[adim]	Label of the run where the event was originated

RQ	Unit	Description
eventID	[adim]	Label of the event within its run
lzapVersion	[adim]	Version of LZap used to process the event
Scatters.ss Tree		
s1pulseID	[adim]	pulseID of the valid Prominent S1
s2pulseID	[adim]	pulseID of the valid Prominent S2
nSingleScatters	[adim]	Binary value - 0 for nonSS and 1 for SS
Scatters.ms Tree		
s1pulseID	[adim]	pulseID of the valid Prominent S1
s2pulseID	[adim]	pulseID of the valid Prominent S2s
nSingleScatters	[adim]	Number of Scatters that originated the event

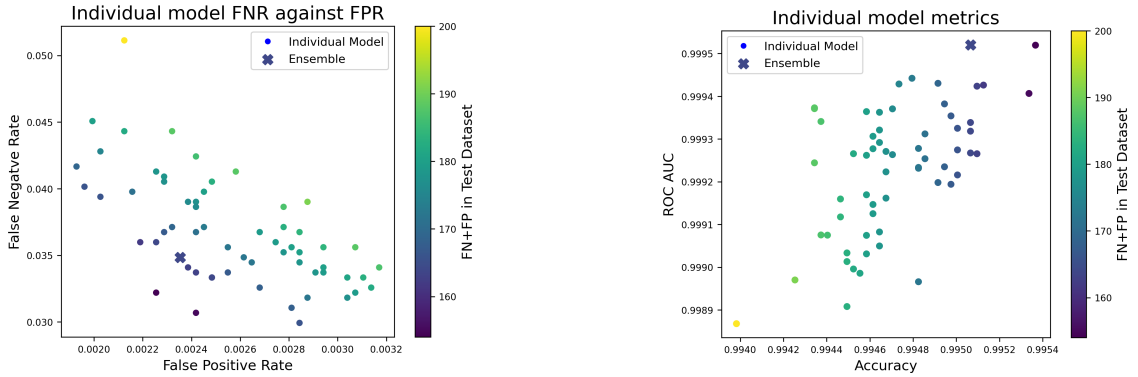
Table 11: Description of relevant RQs.This information was obtained in the LZ Documentation [67], except for rows marked with an *, which were defined using other RQs. Underlined RQs are used in training.

Appendix B

ProFinder Individual Models Metrics

The ProFinder is an ensemble composed by 60 individual and identical neural networks that were trained with random initial values. This was done to reduce model variability and improve its generability.

In Figure 47, the performance of the independent models are compared with the ensemble performance. As it is expected, there are individual models that outperform the ensemble, but the ensemble still has a better than average performance when compared to the individual models. It can also be noted that most models have similar performances.



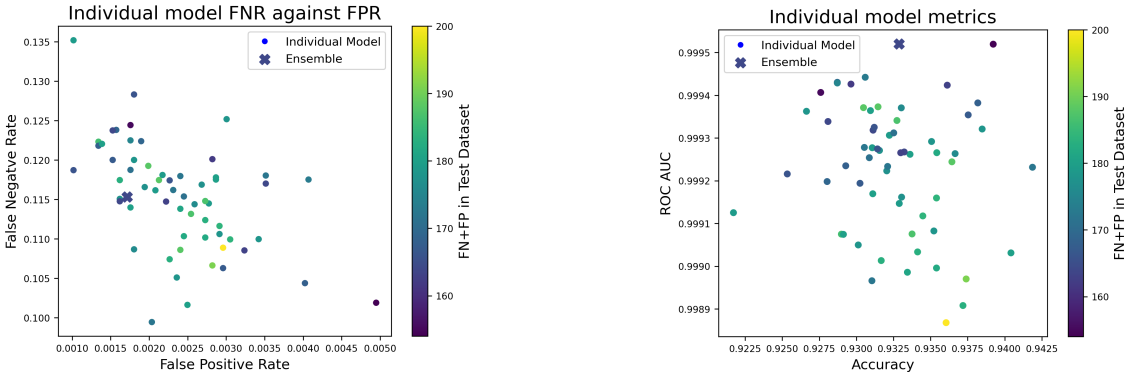
(a) FNR against FPR.

(b) Accuracy against ROC AUC.

Figure 47: Individual ProFinder models metrics for the 5S Dataset. The color of each mark represent the number of falsely classified pulses in the test subset of the 5S Dataset.

The same can be done for the DD Dataset (Figure 48). In these plots, it can be seen that there's no strong correlation between the performance of the individual models on the 5S Dataset (represented by the color of the markers) and the performance on the DD Dataset. A possible explanation would be that some models with strong performances in the training dataset overfitted and lost some generalization power. The ensemble performance is still average to better than average when compared to

the performance of the individual models in the DD Dataset. As above, most models have comparable performances (in relation to each other).

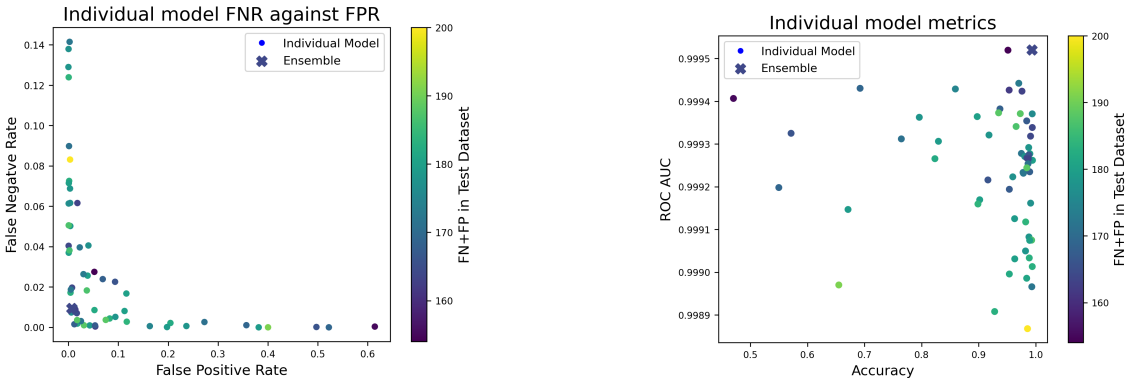


(a) FNR against FPR.

(b) Accuracy against ROC AUC.

Figure 48: Individual ProFinder models metrics for the DD Dataset. The color of each mark represent the number of falsely classified pulses in the test subset of the DD Dataset.

This is not the case for the performance of the individual models on the BG Dataset. In Figure 49, it can be seen that a few of the models have an extremely high False Positive Rate. The performance of the ensemble is, similarly as before, better than average when compared to the individual models. These two observations are compatible with the ensemble output space (Figure 34), where most pulses were correctly classified but had low confidence.



(a) FNR against FPR.

(b) Accuracy against ROC AUC.

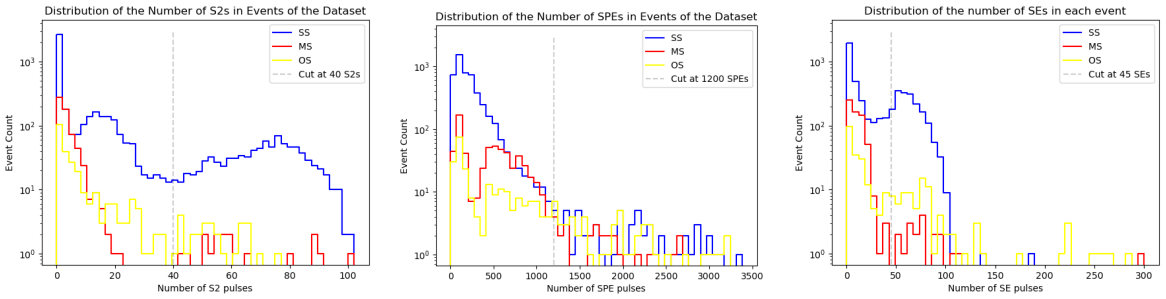
Figure 49: Individual ProFinder models metrics for the BG Dataset. The color of each mark represent the number of falsely classified pulses in the test subset of the BG Dataset.

In general, the ensemble performance was comparable to that of the best individual models, while being more consistent across datasets than those same models.

Appendix C

Cuts for the Scatter Classification Data

The values for the number of pulses cuts were chosen to achieve a balance between the remaining number of events, the equilibrium between SS, Multiple Scatter (MS) and OS and an heuristic natural choice given data distribution based on the histograms from Figure 50.



(a) Number of S2s Distribution. (b) Number of SPEs Distribution. (c) Number of SEs Distribution.

Figure 50: Cuts applied to the dataset for the waveform to scatter model training. The dashed vertical line represents the cutoff point.

In Table12, the distribution of event by event origin is shown and compared to the initial event distribution. It is notable that after the cuts, most high energy events (from the Xe131m calibration and from BG-HE) are removed from the dataset. This is not a worrisome fact, since low energy events tend to be the more challenging events to classify. It may be important to reinforce the notion that the goal of these cuts is not to improve the quality of the dataset, but to reduce the amount of events by removing low-quality events.

Number of events after	**No Cut**	**S2 Cut**	**SPE Cut**	**SE Cut**	**TT Cut**	**All Cuts**
AmLi	880	837	745	775	437	335
BG-HE	1000	1989	1998	1958	1923	1879
BG-LE	2000	1000	1000	78	968	67
Xe131m	1000	139	1000	291	919	3
CH3T	1000	1000	1000	1000	957	957

Table 12: Event distribution after applying various cuts. The numbers in this table do not represent the commulative cuts, i.e., some events are cut more than once.

Appendix D

Hyperoptimization information

As stated in Chapters 5 and 6, the hyperparameters of the models were decided by the Optuna hyperparameter optimization. Chapter 3 gives a brief description of Optuna and the bayesian optimization method applied by Optuna.

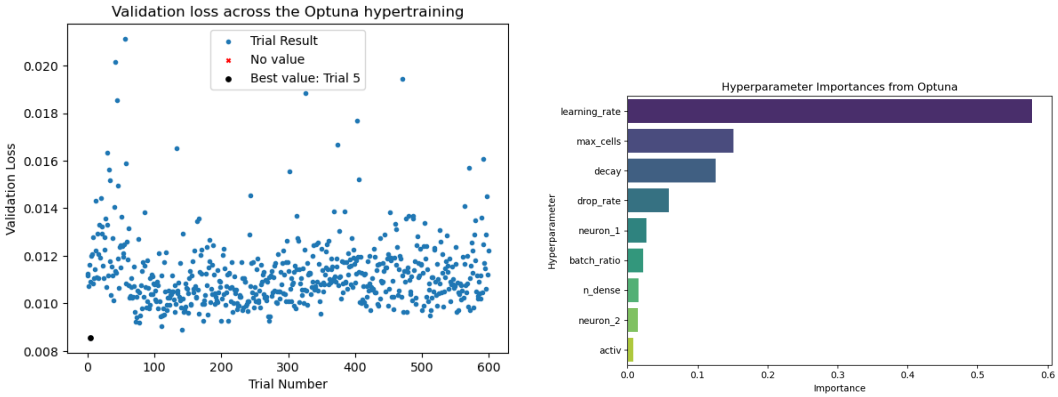
In this appendix, the hypertraining history and the importance of the hyperparameters will be discussed for both models. Hyperparameter convergence refers to the training reducing the scope of the search to a extremely reduced space where the performance of the model is expected to be nearly optimal. Convergence does not always happen, specially if the search space is large and the amount of trials is small. The importance of an hyperparameter relates to the amount of change in the hypertraining metric (in this case, the validation loss), with respect to changes in the value within the hyperparameter space. In this discussion, the importance is computed using the native Optuna method.

D.1 ProFinder Hyperoptimization

To tune nine chosen hyperparameters for the model, 600 Optuna trials were ran. In Figure 53a, the evolution of the validation loss across trials can be seen.

The best trial occurs extremely early (at the trial 5, of 600) and that the training converged around a different, higher, value. A possible explanation to this would be that the loss used in training (mean square error) is not the most adequate loss to this problem. A different explanation would be the limited amount of events in the training and validation dataset, which could cause a trial to randomly overperform on the validation dataset. In a future development, a larger dataset, a larger hyperparameter search space, a larger amount of trials and the testing of different loss functions could eliminate this problem.

Figure 53b shows the importance of the hyperparameters.. The most influential hyperparameter is the learning rate, which tends to be relatively important due to the high impact it has on the training.



(a) Trend of the validation loss evolution.

(b) Hyperparameter importance.

Figure 51: Characterization of the hypertraining for the waveform to scatter model.

Figure 52, the individual hyperparameters for each trial are shown. Note that the hyperparameters of the best trial are not in very populated regions.

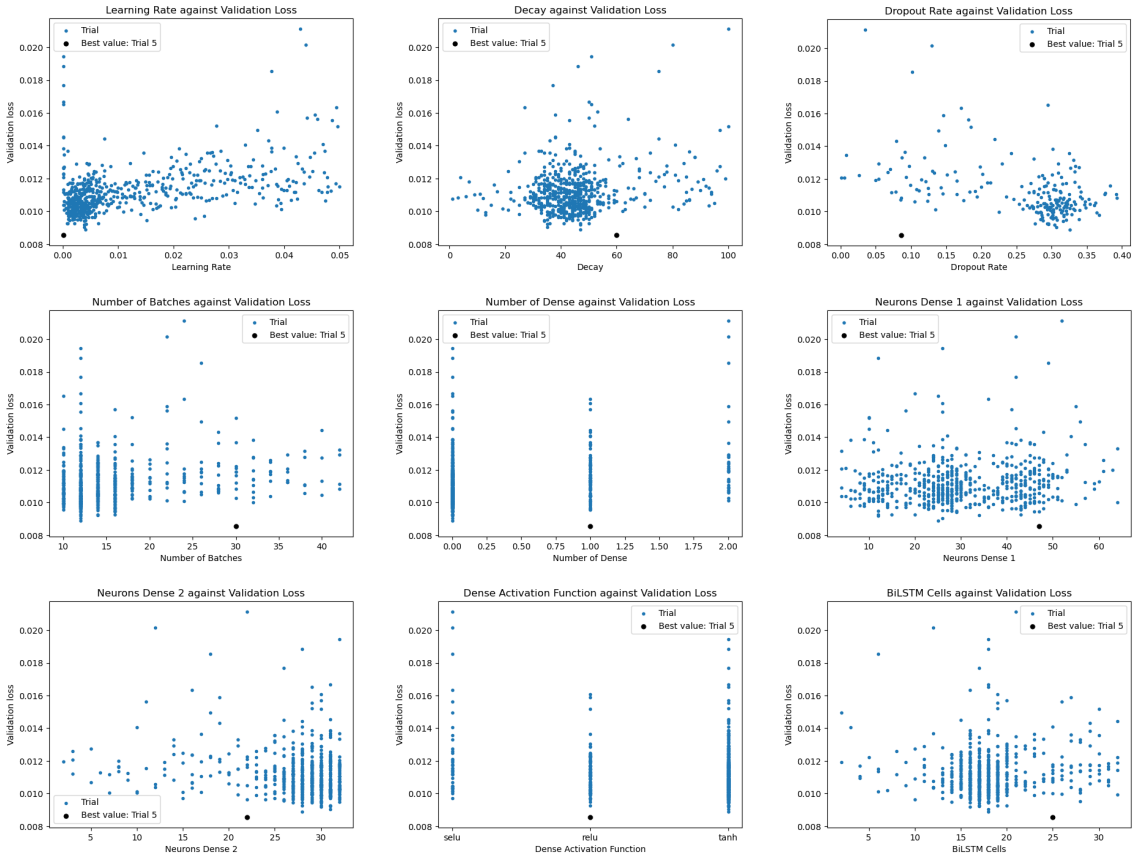
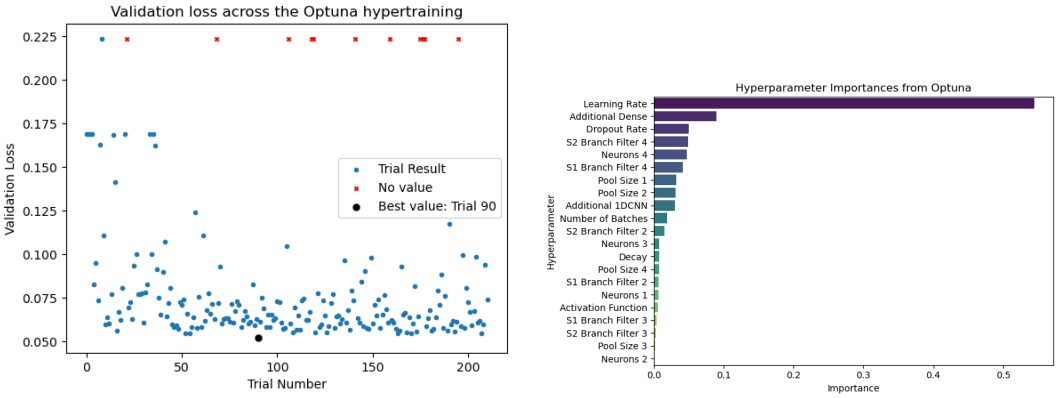


Figure 52: Hyperparameters values for each trial.

D.2 Waveform Hyperoptimization

For this model, 200 trials were ran to determine 21 hyperparameters. The higher number of parameters is easily explained by the deeper nature of the model architeture necessary to achieve this task. In Figure 53a, the evolution of the validation loss across trials can be seen. The best trial occurs relatively

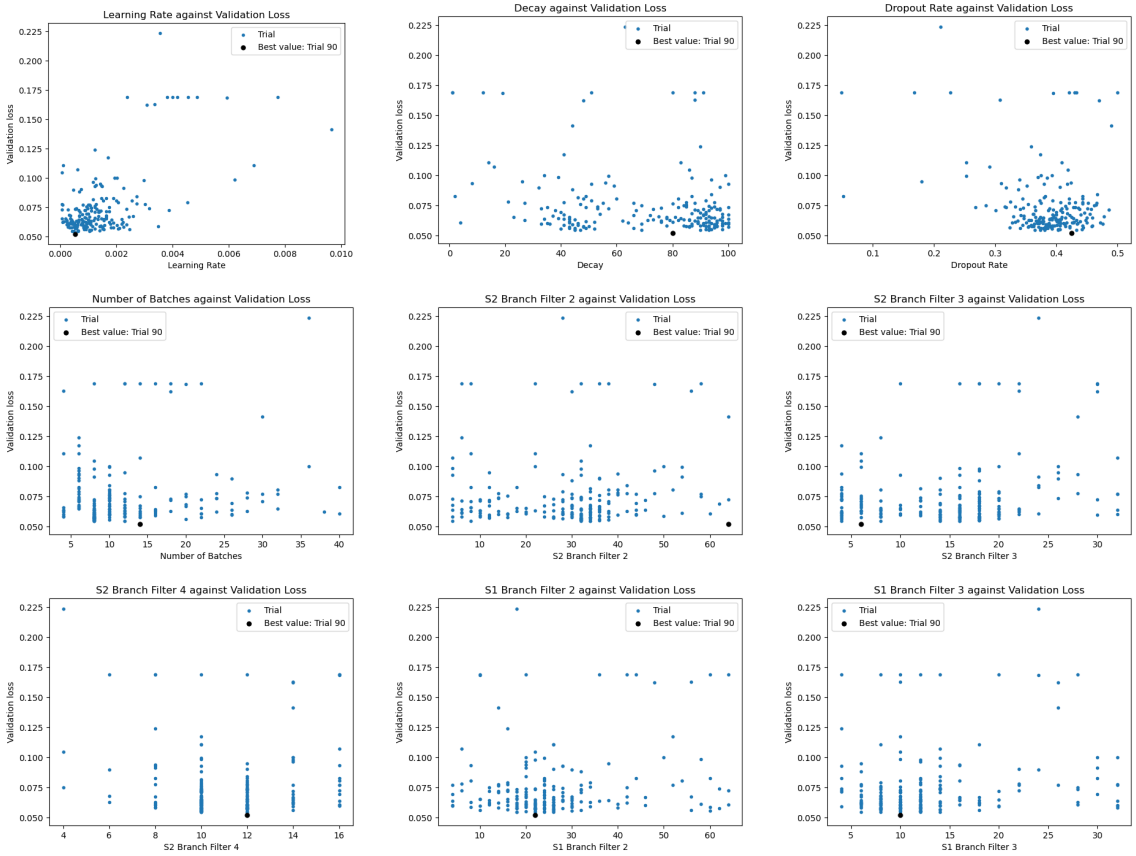


(a) Trend of the validation loss evolution.

(b) Hyperparameter importance.

Figure 53: Characterization of the hypertraining for the waveform to scatter model.

In Figure 54, the individual parameters can be seen. It can be seen that the hyperparameters of the best trial are almost always in a very populated region.



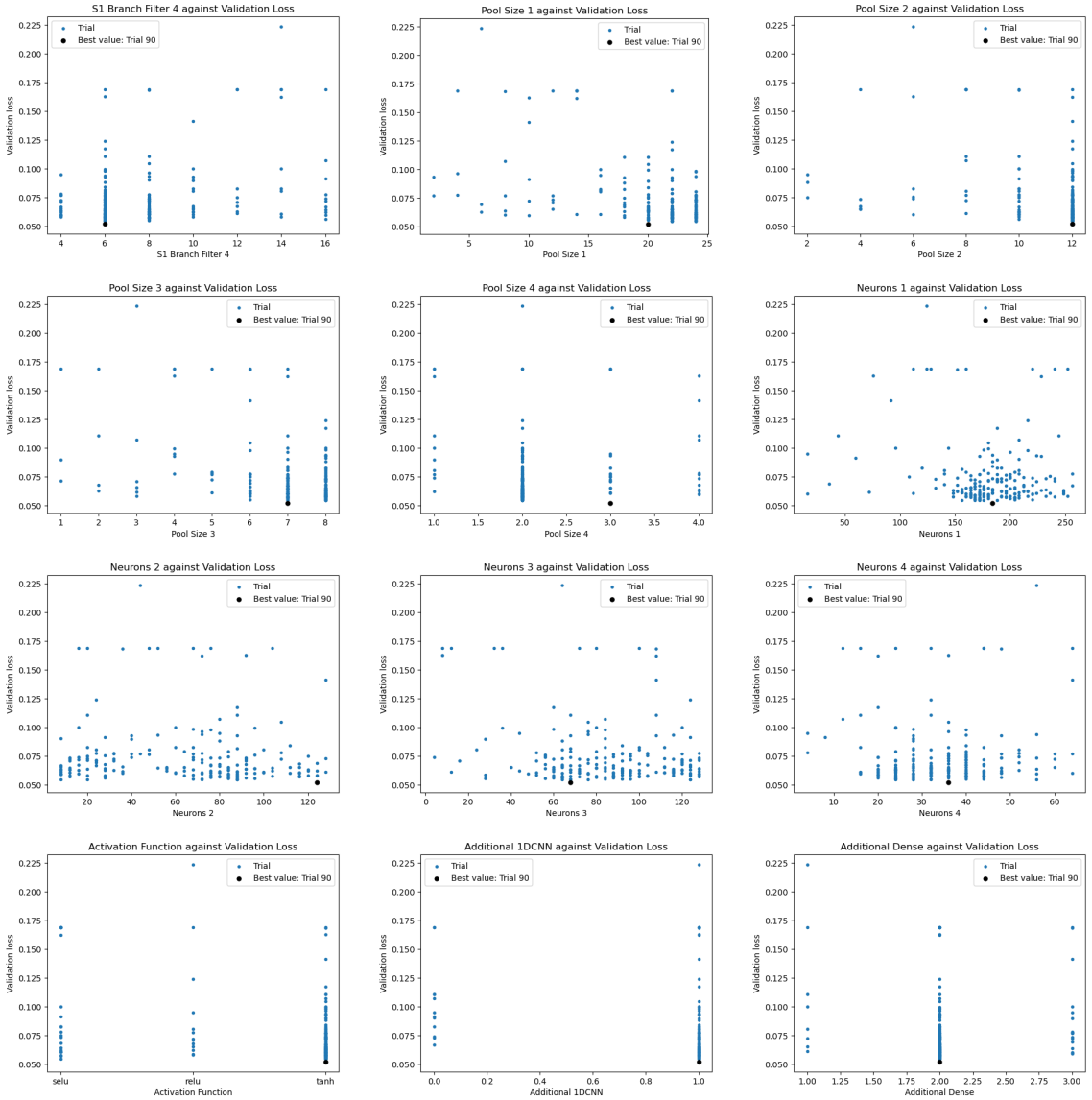


Figure 54: Hyperparameters values for each trial.