Exploring the Frontier of Transient Gravitational Wave Detection

Unleashing the Power of Machine Learning

Melissa López

ISBN: 978-90-393-7812-0
doi: 10.33540/2661
Printed by: Gildeprint – www.gildeprint.nl
Cover Design: Arianna Macias Honti
Copyright: © 2024 Melissa López

This work originates as part of the research program of the Foundation for Fundamental Research on Matter (FOM), and falls as of April 1, 2017 under the responsibility of the Foundation for Nederlandse Wetenschappelijk Onderzoek Instituten (NWO-I), which is part of the Dutch Research Council (NWO).

Exploring the Frontier of Transient Gravitational Wave Detection

Unleashing the Power of Machine Learning

Het verkennen van de grenzen van detectie naar zwaartekrachtsgolven van korte duur

De kracht van machine learning ontketenen

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof.dr. H.R.B.M. Kummeling, ingevolge het besluit van het college voor promoties in het openbaar te verdedigen op

maandag 17 februari 2025 des middags te 4.15 uur

door

Melissa López
geboren op 18 januari 1995
te Buenos Aires, Argentinië

Promotor:
Prof. dr. C.F.F. Van Den Broeck

Co-promotor:
Dr. S.E. Caudill

Assessment Committee:

Dr. E. Cuoco
Dr. A. Grelli
Prof. dr. I.S. Heng
Prof. dr. T.M. Heskes
Prof. dr. T. Peitzmann

To Manel, for introducing
me to the world of physics.

Acknowledgments

As a kid, I had several passions, and one of them was writing. However, never in my life, I thought I would write a dissertation in physics. Not even in the last year of this PhD adventure.

Looking back, in these four years I had the pleasure of delving into the exciting field of gravitational waves and machine learning. Along the way, I met brilliant and inspiring people who helped me grow in my career and be a better person. Without them, this experience would not have been as unique and special.

I would like to start by thanking Sarah and Josh for giving me this amazing opportunity. During this time we faced many challenges-a global pandemic and an international moving-, but it was my pleasure to work with Sarah on all these crazy and interesting projects. She always saw the best in me, even when I could not see it myself. I also thank Chris for his interesting discussions, dedication and expert knowledge.

This adventure would have not been the same without my Utrecht team: Marc, Quirijn, Stefano, Tom and Andrew. Thank you guys for always being there for me. Some of the best memories of this PhD are thanks to you, and now I have stories to tell for a lifetime.

I was lucky to work in a great institute such as GRASP. I am grateful to the staff: Anuradha, Thomas, Monique, Maria-Thérèse, Rene, Marcel and Pannos. I especially thank Alessandro Grelli, Marta and Raymond for their invaluable help and guidance in tumultuous times. I also want to thank all the PhDs, Olaf, Tomek, Anna, Mariia, Thibeau, Luca, Bas, Johanna, Gijs, Christos, Shi, Rik, Nor and Justus; as well as the post-docs, Amit, Bhooshan and Peter.

This thesis would not have been the same without the fundamental work of many brilliant scientists from the LIGO-Virgo-KAGRA collaboration. It was my pleasure to work alongside Vincent and Greg, from Liege University, the Sapienza group, Irene, Marco Drago, Alessandro Veutro and Fulvio; Kurt and Jöel from Maastricht University, and folks from Boosting Alpha. It was my pleasure to supervise excellent students, Robin, Paloma and Ana, with bright futures ahead of themselves. I am also grateful for the help and support of Pablo, Collin, Marco Cavaglia, Rafael, Samaya, Siong and Elena Cuoco. In the collaboration, I was lucky to cross paths and become friends with Jess Irwin, Rodrigo, Alberto, Shubhagata, Siddharth and Nikhil.

Here in Utrecht, I was able to find a second home. All thanks to my Utrecht family: Natalia, Shridar, Gyöngyi, Bayan, Felix, Valeria, Lucila, Mari, Anam, Rajat, Silvia, Arianna, Harsh and Gin. We lived together so many amazing moments, all stored deeply in my heart, but please, we all need to stop moving houses. Also, my beautiful “chicas”, Ritika and Jade, share a special place in my heart. You know that “mi casa es tu casa”.

During this time abroad, I always had people back home, in Spain, welcoming me with open arms. There are too many to list here, but you know who you are. I would like to particularly thank Ana, Diana, Victoria, Juan, Alex, Jero, Pepe, Elena Isabel, Vicky, Adrián, Yaiza, Barbara, David, Lucas and Manuel for all these years of friendship and many more to come. I also thank Marisa, Antonio, Amparo and their families for their love and support.

I am deeply grateful to Carmen for always giving a bit of sense to all my chaos, but also bringing much chaos to my peace. I am thankful to my grandmother for passing on her passion for the humanities to me, to my father for giving me an appreciation for Nature and its mysteries, and to my mother for teaching me to persevere.

Last but not least, I am extremely grateful to Antoni, for all his unconditional love and support through thick and thin. Thank you for being the best travel partner I could ever ask for, and coming with me even to the end of the world.

Contents

* * *

List of Figures **v**

List of Tables **xiii**

List of Terms **xv**

Preface **xvii**

I Foundations of Gravitational Wave Astronomy **1**

1 Brief Introduction to Gravitational Wave Formalism **3**
- 1.1 Linearized Einstein equations 4
- 1.2 The traverse-traceless gauge 6
- 1.3 Interaction with test masses 7
- 1.4 Energy and momentum 10
- 1.5 Generation of gravitational waves 13

2 Gravitational Waves: Sources, Detectors and Analysis **17**
- 2.1 Gravitational-wave sources 18
  - 2.1.1 Compact binary coalescence 18
  - 2.1.2 Transient bursts sources 18
  - 2.1.3 Continuous wave sources 19
  - 2.1.4 Stochastic background 19
- 2.2 Detectors through history 19
- 2.3 Antenna pattern of interferometers 21
- 2.4 Sources of noise 23
  - 2.4.1 Power spectral density 23
  - 2.4.2 Noise lines 24
  - 2.4.3 Glitches 25
- 2.5 Data conditioning 30
  - 2.5.1 Sampling of continuous-time signals 30
  - 2.5.2 Low-pass, high-pass and band-pass filters 31
  - 2.5.3 Resampling 31
  - 2.5.4 Whitening 32
- 2.6 Data quality and vetoes 33

3 Transient Gravitational-Wave Searches **35**
- 3.1 The search formalism 35
- 3.2 Modelled searches 36
  - 3.2.1 Matched-filtering 36
  - 3.2.2 Modelled compact binary coalesence waveforms 38
  - 3.2.3 Template bank 40
  - 3.2.4 GstLAL 41
- 3.3 Unmodelled searches 44

3.3.1 Excess power . . . 44
3.3.2 A short introduction to wavelet analysis . . . 45
3.3.3 Coherent WaveBurst . . . 47
3.4 Background estimation . . . 49
3.4.1 Poisson process . . . 49
3.4.2 Time-shift analysis and false alarm rate . . . 50

II Developing a Machine Learning-Based Detection Algorithm 53

4 Machine Learning Foundations 55
4.1 On the learning landscapes . . . 56
4.2 The realm of deep learning . . . 58
4.2.1 Simplifying the human brain . . . 58
4.2.2 Convolutional neural networks . . . 62
4.3 Review of gravitational-wave applications . . . 64

III Towards the Next Generation of Detections 67

5 From Searches to Machine Learning 69
5.1 A search pipeline as a classification task . . . 69
5.2 The importance of data representation . . . 71
5.3 Mitigation of a glitch background . . . 72
5.4 Towards the next-generation of searches . . . 73

6 Simulating Transient Noise Bursts 75
6.1 Introduction . . . 75
6.2 Data . . . 75
6.2.1 Identification and classification . . . 75
6.2.2 Blip glitches . . . 76
6.2.3 Pre-processing . . . 76
6.3 Methodology . . . 78
6.3.1 Generative adversarial networks . . . 79
6.3.2 Network architecture . . . 81
6.3.3 CT-GAN training procedure . . . 82
6.4 Results . . . 83
6.4.1 Blip generation . . . 83
6.4.2 Assessing performance . . . 83
6.4.3 Assessing poor generations . . . 86
6.4.4 Limitations . . . 89
6.4.5 Applications . . . 90
6.4.6 `gengli` package . . . 94
6.5 Discussion . . . 100
6.6 Conclusion . . . 100

7 Detection of Anomalous Transient Noise Bursts 103
7.1 Introduction . . . 103
7.2 Data . . . 104
7.2.1 Characterization via auxiliary channels . . . 104
7.2.2 Fractal dimension . . . 104

7.2.3 Selected glitches 105
7.2.4 Auxiliary channels encoded in fractal dimension 107
7.3 Methodology 110
7.3.1 Tackling lack of ground truth: convolutional autoencoder 110
7.3.2 Tackling lack of absolute ordering: periodic convolutions 111
7.3.3 t-distributed stochastic neighbour embedding 112
7.4 Results 112
7.5 Discussion 118
7.6 Conclusion 120

8 Enhancing Unmodelled Core-Collapse Supernova Searches 123
8.1 Introduction 123
8.2 Data 124
8.2.1 Phenomenologial waveforms 125
8.2.2 Image generation 127
8.2.3 Previous set 128
8.2.4 Training set 128
8.2.5 Blind set 129
8.2.6 Test set 129
8.3 Methodology 129
8.3.1 Inception blocks and ResNet 129
8.3.2 Mini architectures 131
8.3.3 Optimized architecture of Mini Inception-Resnet 132
8.3.4 Training methodology 133
8.4 Results 134
8.4.1 Comparison of mini architectures in Gaussian noise 134
8.4.2 Comparison with previous results in Gaussian noise 135
8.4.3 Training and validation in real O2 noise 136
8.4.4 Testing in real O2 noise: blind set and test set 138
8.5 Discussion 139
8.6 Conclusion 141

9 Enhancing Modelled Intermediate-Mass Binary Black Hole Searches 143
9.1 Introduction 143
9.2 Data 144
9.2.1 Injections and glitches 144
9.2.2 Clustering 145
9.2.3 Trigger tracks 146
9.2.4 Time shift and false alarm rate 146
9.3 Methodology 147
9.3.1 Feature vector 148
9.3.2 Tackling class imbalance 148
9.3.3 Training model 149
9.3.4 Time coincident tracks 149
9.4 Results 150
9.4.1 Selecting a time window 150
9.4.2 Training with *known data set* of O3a 152
9.4.3 Diving into the *known data*: machine learning performance 152
9.4.4 Diving into the *known data*: significance of P_{inj} statistic 155
9.5 Discussion 158

9.6 Conclusion 158

10 Conclusions and prospects **161**

A Additional Resources **165**

A.1 Dominant sources of noise 165
    A.1.1 Thermal noise 165
    A.1.2 Quantum noise 166
    A.1.3 Seismic noise 167
    A.1.4 Gravity gradient noise 168
A.2 Wilson-Daubechies-Meyer formalism 169
A.3 Welch's method 170
A.4 `gengli`: results for *Blip* distribution from LIGO Livingston 170
A.5 IMBH search: results of LIGO Livingston and Virgo 173
    A.5.1 Selecting a time window 173
    A.5.2 Training with *known data set* of O3a 174
    A.5.3 Diving in to the *known data*: performance evalutation 175

Public Summary **177**

Openbare Samenvatting **183**

Resumen público **190**

Curriculum Vitae **197**

Bibliography **199**

List of Figures

1-1 A schematic illustration of the deformation of a ring of test masses for different phase values due to the + and × polarization of a passing GW. 10

2-1 The different types of GW sources can be differentiated by their duration (short-/long) and their associated astronomical models (modelled/unmodelled). Further details of the sources are described in the main text. Credits: Shanika Galaudage. 17
2-2 Temporal evolution of a binary system with component masses $m_1 = m_2 = 20M_\odot$. We colour in blue the inspiral, in orange the merger, and in green the ringdown. 18
2-3 A schematic setup of a Michelson-Morley interferometer retrieved from [30]. . . . 19
2-4 *(Left)* LISA orbiting around the Sun. *(Right)* Einstein Telescope underground. Artists impressions. 21
2-5 Location of the current network of second-generation interferometric detectors. . 21
2-6 Variation of $F = \sqrt{F_+^2 + F_\times^2}$ for $\psi = 0$ as a function of the longitude ϕ and the latitude θ for LIGO Livingston *(left)* and Virgo *(right)*. 22
2-7 *(Left)* PSDs of LIGO Livingston during the third observing run (O3) (blue), LIGO A+ (orange), LIGO A$^\sharp$ (green), Einstein Telescope (pink) and Cosmic Explorer (red). *(Right)* Design sensitivity of Advanced LIGO (black), and noise sources from seismic noise (brown), suspension thermal noise (blue), coating thermal noise (red), gravity gradient (green) and quantum noise (purple), retrieved from [42]. 23
2-8 Time-frequency representations of different glitch classes in LIGO during O3. . . 29
2-9 Schematic low-pass *(left)*, high-pass *(middle)* and band-pass filters *(right)*. . . . 31
2-10 *(First panel)* Raw data from the detector $s(t)$, dominated by low and high frequencies. *(Second panel)* Whitened data, *(Third panel)* Whitened and band-passed data. *(Third panel)* PSDs of coloured data (pink), white data (blue), white and band-passed data (green). 33

3-1 Waveform of a highly asymmetric system ($m_1 = 40\mathrm{M}_\odot, m_2 = 10\mathrm{M}_\odot$) using IMRPhenomXPHM [101]. *(Top panel)* Circular system with aligned spins and no higher order modes. *(Bottom panel)* Precessing system ($\chi_1 = [-0.2, 0.4, 0.1], \chi_2 = [-0.5, 0.2, -0.4]$) with higher-order modes ($\ell = 2, m = 1,$). 40
3-2 Template bank as a function of the component masses employed by GstLAL pipeline during O2 and O3, retrieved from [106]. Every point represents a template, either from BNS (green), NSBH (red), BNS (blue). Points in magenta and black were added to improve the background estimation for the scarcely populated region of the bank. 40
3-3 IMBH template bank from O3 (grey) as a function of the progenitors' masses measured in $M_\odot$ and logarithmic scale. We colour in blue the templates that generated a trigger with their maximum SNR ρ due to the presence of GW190521. *(Top panel)* Hanford detector. *(Bottom panel)* Livingston detector. 43

3-4 *(Left):* Fourier transform with a fixed window in the time-frequency plane. The same rectangular window is used to analyze high and low frequencies. *(Right):* Wavelet transform with a sketch of Daubechies wavelets in the time-frequency plane. The different window sizes allow us to see small and gross features. Reproduced from [131]. 45
3-5 GW150914 in LIGO Hanford. *Left:* cWB spectrogram, first time-frequency representation of a GW signal. *Right:* Omicron spectrogram retrieved from [73]. Note the reduction on the spectral leakage of cWB spectrogram with respect to the Omicron spectrogram. 47
3-6 Sketch of data in a network of detectors. 51

4-1 Sketch of the different types of ML, namely, supervised (green), unsupervised (orange) and reinforcement (blue). 57
4-2 Representation of a galaxy by DALL·E 2 [155], a DL model. 58
4-3 *(Left:)* Cartoon of a biological neuron, the basic computational unit of the human brain. *(Right:)* Sketch of its mathematical model. 59
4-4 Comparison of sigmoid *(left)* and ReLU *(right)* activation functions. Note the difference in activation range. 60
4-5 *(Left:)* Sketch of a kernel of size $k = 3$ over an input $6 \times 6 \times 1$, a dilation $d = 1$, a stride $s = 2$ and a padding $p = 1$. *(Right:)* Sketch of a kernel of size $k = 3$ over an input $7 \times 7 \times 1$, a dilation $d = 2$, a stride $s = 1$ and a padding $p = 0$, retrieved from [166]. 63
4-6 Cumulative count of gravitational wave events detected by LIGO and Virgo, split by observing run. Note that O3 has been divided between O3a and O3b. Credits to LIGO-Virgo-KAGRA. 64

5-1 Flamingos and pelicans as seen by a ML model, retrieved from [240]. 71

6-1 Time-frequency reperesentation of a *Blip* glitch retrieved from `Gravity Spy` [65] *(left)* and a GW signal with total mass $106.6^{+13.5}_{-14.8} M_\odot$ *(right)*. 76
6-2 *(Top left) Blip* glitch reconstructed with BW (grey), where we colour the characteristic *Blip* peak (blue) and the undesired high-frequency contribution (light blue). *(Top right) Blip* glitch reconstructed with BW (grey) and denoised with $\lambda = 0.5$ (dashed orange). We colour in green the denoised characteristic *Blip* peak. *(Bottom)* Resulting ASD for the reconstructed *Blip* with BW (grey) and its denoised version with $\lambda = 0.5$ (dashed orange). We also show the ASD of the characteristic peak with (blue) and without denoising (green), as well as the high-frequency contribution (light blue). 77
6-3 Comparison between the reconstructed and the denoised population of *Blip* glitches for L1. For the reconstructed set $c^2_{GS} = 0.892 \pm 0.003$ and for the denoised set $c^3_{GS} = 0.874 \pm 0.004$ at 95% confidence level. 78
6-4 Typical GAN architecture retrieved from [257]. 79
6-5 Generator structure including NN upsampling, convolution layers and LeakyReLU activation. 81
6-6 Discriminator architecture showing strided convolutions, dropout layers and LeakyReLU activations. 81
6-7 Graph representing the discriminator loss (blue), generator loss (pink), CT (green) and GP (orange) penalisation as a function of the epochs. 82
6-8 *(Top)* Peak frequency for *Tomte* (pink) and *Blip* (green) from L1 retrieved from `Gravity Spy`[65], measured with Omicron spectrograms [73]. In blue we plot the peak frequency of the artificial *Blips* from L1. 82

6-9 *(Top row)* Generated *Blip* of L1 plotted as a function of time. In red, we represent the rescaled whitened *Blip* and in blue we plot its injection, both in the time domain. *(Bottom row)* We show the time-frequency representation of the generated injected glitches. 83
6-10 Histogram of predicted `Gravity Spy` classes for 10^3 generated *Blips* from H1. . . 85
6-11 *(Top)* We represent the joint and marginal distributions of $W_1(B_R, B_F)$ and $W_1(B_F, B_F)$ in logarithmic scale for L1 (blue) and H1 (orange) and their best fit. *(Bottom)* We represent the joint and marginal distributions of the pairs $[M_f(B_R, B_F), M_f(B_F, B_F)]$ and $[k(B_R, B_F), k(B_F, B_F)]$ for L1 (blue and pink) and H1 (orange and green), as well as their best fit. The coloured regions in the marginal distributions represent the confidence interval at 6 standard deviations. 87
6-12 Joint and marginal distribution of `Gravity Spy` confidence c_{GS} at $\rho_{opt} = 18.46$ against different metrics for different glitch classes for H1: *Blip*, *Repeating_Blips* and *No_Glitch*. We mark in the marginal distributions selected generated glitches A (solid blue), B (dotted pink), C (dashed green) and D (dash-dotted blue). 88
6-13 Time series representation (top row) and Q-scan representation of selected glitches from H1 89
6-14 PCA representation of fake *Blip* population of H1, clustered with Gaussian Mixture. The 5% most anomalous *Blips* according to the distance $W_1(B_F, B_F)$ are marked with a star. 90
6-15 The singular values (σ) are obtained from a set of 10^3 whitened glitches using SVD [278], normalized by the maximum singular values ($\sigma_{\max}$). The glitches are generated from CT-GAN. The spectrum of singular values is seen to fall sharply, implying only a few singular values (e.g., $\ell = 10$), and corresponding basis vectors are sufficient to represent the glitches. See Fig. 6-16 in which the relative reconstruction error for these glitches has been shown based on $\ell = 1, 5, 10$. For performing SVD based matched filtering for glitch templates, we followed the framework presented in [123]. 91
6-16 The plot shows the distribution of relative errors for the reconstruction of the 10^3 whitened glitches generated using CT-GAN. The relative error (ϵ_α) is calculated for each case. 93
6-17 This figure shows the histogram based on the fractional SNR loss ($\frac{\delta\rho_\alpha}{\rho_\alpha}$) for a set of glitches (10^3). For each glitch template, the SNR time-series were obtained based on (a) Standard matched-filter scheme and (b) SVD based matched filtering framework presented in [123] by varying the top-basis numbers as $\ell = 1, 5, 10$ respectively. 93
6-18 Example of glitch injection. The left image shows the input time-frequency map while the right panel shows the output of `ALBUS`. Retrieved from [217]. 94
6-19 A whitened glitch added to the white noise of Einstein Telescope, retrieved from [216]. 95
6-20 Probability distribution of distances (d_W, d_M, d_{cc}) for a benchmark population of 10^3. 96
6-21 Average `Gravity Spy` confidence c_{GS} for 10^3 artificial *Blip* glitches, generated with `gengli`, as a function of SNR ρ. 97
6-22 `Gravity Spy` classes for 10^3 generated *Blips* from H1. 97
6-23 Joint and marginal distribution of `Gravity Spy` confidence c_{GS} at $\rho_{opt} = 18.46$ against mis-match d_M for different glitch classes for H1: *No_Glitch* (blue), *Blip* (orange), *Repeating_Blips* (brown), *Blip_Low_Frequency* (grey), *Chirp*(pink), *Koi_Fish* (green). 98

6-24 Selected `gengli` glitches. *(Top panel):* confidence c_{GS} as a function of SNR. *(Middle panel):* Raw glitch signal output from `gengli`. *(Bottom panel):* spectrogram of glitch injected in real detector noise from H1 at $\rho = 18.32$. 99

7-1 Fractal dimension over a two-minute period of L1 data for the `L1:LSC-PRCL_OUT_DQ` auxiliary channel. Each point represents the fractal dimension for one second of data, and the red regions indicate the time period containing `Whistle` glitches. The grey region indicates the confidence interval of the fractal dimension at one standard deviation. 104
7-2 Comparison of the fractal dimension computing algorithms for varying number of data points. Benchmarks done on a Intel® Xeon® Processor E5-2630 v4 CPU @ 2.20GHz. 107
7-3 From left to right: Q-transform of `No_Glitch`, `Whistle`, `Tomte` and `Scattered_Light` retrieved from `Gravity Spy` [65]. 107
7-4 A sample encoded in fractal dimension with 347 safe auxiliary channels and 56 time bins. Notably, the same event is replicated four times with a different time window: time bins $\in [0, 31)$ have a time window $\mathcal{W}(t) = 0.25\,\mathrm{s}$, time bins $\in [31, 46)$ have $\mathcal{W}(t) = 0.5\,\mathrm{s}$, time bins $\in [46, 53)$ have $\mathcal{W}(t) = 1\,\mathrm{s}$, time bins $\in [53, 56]$ have $\mathcal{W}(t) = 2\,\mathrm{s}$. 109
7-5 Outline of the autoencoder with periodic convolutions implementation. The input data which is concatenated yields a shape $(t, 2c - 1)$ is presented in green. Downsampling convolutions are coloured in dark orange while upsampling convolutions are coloured in dark blue. Each convolution is followed by a Rectified Linear Unit (ReLU) activation function coloured in orange. This architecture yields a reconstruction of shape (t, c). 110
7-6 Example of periodic convolution where input of shape $(2c - 1, t)$ is convolved with a kernel $k = c = 4$. All cyclic channel permutations are convolved, so there is no absolute ordering. 111
7-7 Histogram with the reconstruction errors ϵ_R for each glitch class in logarithmic scale. 113
7-8 Joint and marginal distributions of t-SNE projections of the data. Legends correspond to `Gravity Spy` classification with confidence $> 90\%$ 114
7-9 **Region 1:** Example of anomalous glitches in the embedded space. 116
7-10 **Region 2:** Example of anomalous glitches in the embedded space. 117
7-11 **Region 3:** Example of anomalous glitches in the embedded space. 118
7-12 **Region 4:** Example of anomalous glitches in the embedded space. 119
7-13 **Region 5:** Example of anomalous glitches in the embedded space. 120

8-1 Number of simulations with a given g-mode root mean square (rms) strain at 10 kpc (per logarithmic interval) for 2-dimensional (blue bars) and 3-dimensional (red bars) simulations in the waveform calibration set. 125
8-2 Example of a phenomenological waveform generated by the waveform generator. The upper panel shows the strain as a function of time and the lower panel the corresponding spectrogram. The white-dotted curve shows the time-frequency dependency used to generate the waveform. White circles represent the pairs (t_i, ν_i) used to generate it. 126
8-3 From the top; the WDM spectrogram of LIGO Hanford, LIGO, Livingston and Virgo shown in red, green and blue, respectively. At the bottom: the image obtained by stacking the previous three spectrograms. In this case, the signal is present just in Hanford and Livingston so that the combined signal is yellow. . 127
8-4 Inception module with dimensionality reduction, adapted from [340]. 130

8-5 Reduced or “mini” versions of Resnet, Inception v4 and Inception-Resnet v1 for the best performing architectures with minimal fine-tunning. 131
8-6 The overall schema of the Mini Inception-Resnet network. 132
8-7 The schema for Inception-ResNet-A, adapted from [343]. 133
8-8 Schematic blocks from original Inception-Resnet, adapted from [343] 133
8-9 η_{CNN} (solid lines) and FAR_{CNN} (dashed lines) as functions of SNR computed during the validation process for architectures presented in 8.3.2 134
8-10 η_{CNN} (solid lines) and FAR_{CNN} (dashed lines) for different SNRs computed during the validation process for $w = 1.0$ and different θ^* thresholds. 135
8-11 η_{CNN} (solid lines) and FAR_{CNN} (dashed lines) for different SNRs computed during the validation process for $w = 2.0$ and different θ^* thresholds. 136
8-12 figure 137
8-13 η_{CNN} (solid line) and FAR_{CNN} (dashed line) for SNR in range $[1, 232]$, for $w = 2.0$ and $\theta^* = 65\%$. These results are obtained from validating 25% of the data that we have not trained on. 137
8-14 Histogram of real detector noise and injections in real-time as a function of the probabilities predicted by Mini Inception Resnet. The vertical line represents the chosen decision threshold $\theta^* = 65\%$. Given the counts of the ith bin c_i and its width b_i, we define the probability density as $c_i/(\sum_i^N c_i \times b_i)$, where N is the total number of bins of the histogram. 138
8-15 Performance of our neural network for the *blind set* and the *test set* for $\{w, N, \theta^*\} = \{2, 30000, 65\%\}$. AUC is presented in the legend of the plot. 138
8-16 Performance evaluation computed during the testing process for $\{w, N, \theta^*\} = \{2, 30000, 65\%\}$. 139
8-17 Catalog waveforms (top row) with their optimized phenological waveform from ccphen (bottom row) represented in time and time-frequency domain. 142

9-1 A *Fast Scattering* glitch labelled by `Gravity Spy`. *(Top)* SNR of GstLAL tracks as a function of time, where each point represents a template. t_{min} (orange) and t_{max} (pink) mark the beginning and end of the cluster, respectively, while $t_{cluster}$ (red) indicates its centroid given by `Gravity Spy`. *(Bottom)* Q-transform of the time series containing the *Whistle* labelled by `Gravity Spy`. 145
9-2 O3 IMBH template bank as a function of the progenitor masses in logarithmic scale, where every grey point represents a template from the bank. Triggers associated with an *Injection* or a *Blip* glitch are coloured according to the maximum SNR, ρ. 147
9-3 Illustration of K-fold cross-validation. 149
9-4 ROC curve for different time windows t_w in H1, i.e. TPR as a function of FPR in logarithmic scale. The positive class is *Injection* class, while the negative class is any of the other classes. *(Top)* Testing in the *known data set* of O3a. *(Bottom)* Testing in the *known data set* of O3b. Note that the dashed line indicates a random guess. 151
9-5 A *Whistle* glitch labelled by `Gravity Spy`. *(Top)* SNR of GstLAL tracks as a function of time, where each point represents a template. t_{min} (orange) and t_{max} (pink) mark the beginning and end of the cluster, respectively, while $t_{cluster}$ (red) indicates its centroid given by `Gravity Spy`. *(Bottom)* Q-transform of the time series containing the *Whistle* labelled by `Gravity Spy`. While `Gravity Spy` labels a single glitch, GstLAL identifies both. 151

9-6 Comparison between training and validating with 9-fold cross-validation for H1 during O3a. *(Top)* Mean accuracy at 3 standard deviations (shaded region) as a function of the epochs during training and testing. *(Bottom)* Average 9-fold cross-validation loss as a function of the epochs during training and testing. . . . 152

9-7 Relative values of the confusion matrix for the test set for H1. The y-axis represents the groud-truth, i.e. the true class, and the x-axis represents the prediction of the model, i.e. the predicted class. 153

9-8 *(Top panel)* Single detector TPR as a function of the average SNR $\bar{\rho}$ in the *known data set* of O3b for H1, L1 and V1 . *(Bottom panel)* SNR ρ probability density of different classes in the *known data set* of O3b. 154

9-9 Probability density of being *Injection*, using the *known* test set of O3b for H1, in blue, L1, in red, and their time coincidence, in purple. *(Top)* Probability density of elements in the *Injection* class in logarithmic scale. *(Bottom)* Probability density of elements in all the other classes, i.e. glitches, in logarithmic scale. Given the counts of the ith bin c_i and its width b_i, we define the probability density as $c_i/(\sum_i^N c_i \times b_i)$, where N is the total number of bins of the histogram. 155

9-10 ROC curve of double and triple detector time (H1L1V1) for Eq. 9.5 (dashed lines) and Eq. 9.6. Here, TPR is the rate of correctly classified GW injections from the zero lag, while FPR is the rate of incorrectly classified time slides background tracks. 155

9-11 Inverse false alarm rate (iFAR) as a function of the ranking statistic in triple time (H1L1V1) for different detector combinations. The solid lines show the time-shifted background with real GW signal tracks, while dashed lines show the time-shifted background without them. 157

9-12 TPR, i.e., the rate of correctly classified GW injections, at iFAR $= 0.1$ (equivalent to one per 10 years) for different detector combinations in triple time (H1L1V1). Error bars represent the standard error at 3 standard deviations. 157

A-1 Schematic illustration of a fluctuating gravitational force F_g on a suspended test mass, caused by the propagation of a surface wave through the ground. 168

A-2 Histogram of predicted `Gravity Spy` classes for 10^3 *Blips* from L1. 171

A-3 Joint and marginal distribution of `Gravity Spy` confidence c_{GS} at $\rho_{opt} = 18.46$ against different metrics for different glitch classes for L1: *Blip*, *Repeating_Blips* and *No_Glitch*. We mark in the marginal distributions selected glitches A (solid blue), B (dotted pink), C (dashed green) and D (dash-dotted blue). 172

A-4 Time series representation (top row) and Q-scan representation of selected glitches from L1. 173

A-5 ROC curve for different time windows t_w in L1, i.e. TPR as a function of False Positive Rate FPR in logarithmic scale. *(Left)* Testing in the *known data set* of O3a. *(Right)* Testing in the *known data set* of O3b. Note that the dashed line indicates a random guess. 173

A-6 ROC curve for different time windows t_w in V1, i.e. TPR as a function of False Positive Rate FPR in logarithmic scale. *(Left)* Testing in the *known data set* of O3a. *(Right)* Testing in the *known data set* of O3b. Note that the dashed line indicates a random guess. 174

A-7 Comparison between training and validating with 9-fold cross-validation during training and testing. *(Top row)* Mean accuracy at 3 standard deviations as a function of the epochs during. *(Bottom row)* Average 9-fold cross-validation loss as a function of the epochs. 174

A-8 Relative values of the confusion matrix for the test set for L1. 175

A-9 Relative values of the confusion matrix for the test set for V1. 175
A-10 Probability density of being *Injection* (P_{inj}), using the *known* test set of O3b. *(Top row)* Probability density of elements in the *Injection* class in logarithmic scale. *(Bottom row)* Probability density of elements in all the other classes, i.e. glitches, in logarithmic scale. Given the counts of the ith bin c_i and its width b_i, we define the probability density as $c_i/(\sum_i^N c_i \times b_i)$, where N is the total number of bins of the histogram. 176
S1.1 James Webb Space Telescope image with a large number of lensed galaxies. Credits: NASA, ESA, CSA, and STScI. 177
S1.2 Hubble Space Telescope image of the Crab Nebula. Credits: NASA and STScI. . 178
S1.3 Interpretation of the cosmic auditorium. Credits:DALL · E 2 179
S1.4 Sagittarius A*, the supermassive black hole at the center of the Milky Way. Credits: EHT. 180
S2.1 James Webb Space Telescope afbeelding met een groot aantal sterrenstelsels. Credits: NASA, ESA, CSA en STScI. 183
S2.2 Hubble Space Telescope afbeelding van de Krabnevel. Credits: NASA en STScI. 184
S2.3 Interpretatie van het kosmisch auditorium. Credits: DALL · E 2 185
S2.4 Sagittarius A*, het supermassieve zwarte gat in het centrum van de Melkweg. Credits: EHT. 186
S3.1 Imagen del telescopio espacial James Webb con un gran número de galaxias lentes. Créditos: NASA, ESA, CSA y STScI. 190
S3.2 Imagen del telescopio espacial Hubble de la nebulosa del Cangrejo. Créditos: NASA y STScI. 191
S3.3 Interpretación del auditorio cósmico. Créditos: DALL · E 2 192
S3.4 Sagitario A*, el agujero negro supermasivo en el centro de la Vía Láctea. Créditos: EHT. 193

List of Tables

5.1 Confusion matrix of GW + noise (positive class) against only noise (negative class) 70

6.1 Size of the *Blip* set for each detector in the different phases of the pre-processing: selection, reconstruction and evaluation. 76
6.2 Pearson coefficient for different metrics and detectors. 86

8.1 Parameter space of the phenomenological templates. The second, third and fourth columns indicate the range minimum and maximum, respectively) for each parameter and the spacing used in the sampling of the parameter space. For Q and D, we show the actual values instead. All times are post-bounce times. 126
8.2 List of models of the test set used in the injections. M_{ZAMS} corresponds to the progenitor mass at zero-age in the main sequence (ZAMS). Unless commented, all progenitors have solar metallicity, result in explosions and their GW signal do not show signatures of the standing-shock accretion instability (SASI). 128

9.1 Total time of search (T_s) for different detector combinations, measured in years. 146
9.2 Original data set size before sampling 148
9.3 Area under the curve (AUC) from ROC curves in Fig. 9-10. 156

List of Terms

ASD Amplitude Spectral Density

AUC Area Under the Curve

BBH Binary Black Holes

BNS Binary Neutron Star

BW BayesWave

CBC Compact Binary Coalesence

CCSN Core-Collapse Supernova

CNN Convolutional Neural Networks

CT Consistency Term

CT-GAN Consistency Term Generative Adversarial Network

cWB coherent WaveBurst

DL Deep Learning

EOB Effective One Body

FAR False Alarm Rate

FD Fractal Dimension

FN False Negative

FP False Positive

FPR False Positive Rate

GAN Generative Adversarial Networks

GP Gradient Penalty

GPU Graphics Processing Unit

GW Gravitational Waves

GWOSC Gravitational-Wave Open Science Center

H1 LIGO Hanford

iFAR Inverse False Alarm Rate

IMBH Intermediate-Mass Black Hole

IMR Inspiral, Merger and Ringdown

KAGRA Kamioka Gravitational Wave Detector

L1 LIGO Livingston

LIGO Laser Interferometer Gravitational-wave Observatory

LISA Laser Interferometer Space Antenna

LSC Length Sensing Control

MCMC Markov Chain Monte Carlo

ML Machine Learning

MLP Multi-Layer Perceptron

NN Nearest-Neighbour

NSBH Neutron Start - Black Hole

O1 First Observing Run

O2 Second Observing Run

O3 Third Observing Run

O3a First Half of the Third Observing Run

O3b Second Half of the Third Observing Run

O4 Fourth Observing Run

PD Photo Detector

PRCL Power Recycling Cavity Length

PSD Power Spectral Density

ReLU Rectified Linear Unit

rms Root Mean Square

ROC Receiver Operating Characteristic

rROF Rudin-Osher-Fatemi

SASI Standing-Shock Accretion Instability

SMBH Super-Massive Black Holes

SNEWS SuperNova Early Warning System

SNR Signal-to-Noise-Ratio

STFT Short-Time Fourier Transform

SVD Singular Value Decomposition

t-SNE t-distributed stochastic neighbour embedding

TN True Negative

TP True Positive

TPR True Positive Rate

TT Transverse-Traceless

V1 Virgo

WDM Wilson-Daubechies-Meyer

WGAN Wasserstein Generative Adversarial Networks

ZAMS Zero-Age in the Main Sequence

Preface

Overview

In 2015, the LIGO-Virgo collaboration confirmed the first direct observation of gravitational waves resulting from the merger of two black holes. However, the journey to detect these waves, predicted by Albert Einstein nearly a century earlier, has been arduous. It required decades of technological innovation, persistent scientific investigation, and global collaboration to finally enable us to probe the densest and most energetic regions of the universe.

Over the past three observing runs, the second-generation detectors, Advanced LIGO and Advanced Virgo, have confirmed over 90 gravitational wave signals, with many more expected during the ongoing fourth observing run. As the sensitivity of these second-generation detectors continues to improve and third-generation interferometers are added to the network, the number of detections is expected to increase exponentially. Nevertheless, as the number of transient gravitational wave detections grows, new computational challenges will emerge, and novel frontiers will need exploration.

Nowadays, machine learning is a booming enterprise. In the past few years, they have gained significant interest due to their success in various tasks and domains, such as their ability to uncover intricate patterns and perform rapid, accurate inference. In this regard, the field of gravitational waves is no exception. In the context of an exploding field of both gravitational wave discoveries and machine learning applications, in this thesis we unleash the power of machine learning to complement state-of-the-art gravitational wave searches and even enhance their performance, fostering a synergistic relationship between the two disciplines.

Outline

This thesis is divided into three different parts. Part I lays the foundation of gravitational wave astronomy, covering from theory to experiment and providing an overview of data analysis techniques. Part II introduces the core principles of machine learning. Finally, Part III demonstrates how current search algorithms can be enhanced by leveraging machine learning methods.

Part I. Foundations of Gravitational Wave Astronomy

Chapter 1 offers a concise introduction to gravitational wave theory. Chapter 2 discusses the various types of gravitational wave sources, provides an overview of the detectors and their noise sources, and explains the fundamentals of data analysis. Chapter 3 reviews both modelled and unmodelled searches, detailing how the significance of gravitational wave candidates is measured.

Part II. Developing a Machine Learning-Based Detection Pipeline

Chapter 4 introduces machine learning methods and reviews their applications in the field of gravitational wave research.

Part III. Towards the Next Generation of Gravitational Wave Detection

Chapter 5 bridges the field of machine learning with gravitational wave searches, highlighting the primary challenges: background mitigation and limited sensitivity on certain transient sources. Chapter 6 simulates transient noise bursts within the background distribution. Chapter 7 investigates data-driven methods for identifying transient noise bursts. Chapter 8 enhances unmodelled gravitational wave search algorithms with a focus on supernovae explosions. Chapter 9 enhances modelled gravitational wave search algorithms, with a focus on black hole mergers.

External contributions

The work described in this thesis is the result of collaborative efforts. In Chapter 6, Kerwin Buijsman provided the clean signals of transient burst noise, Vincent Boudart developed the initial machine learning model, and Stefano Schmidt assisted in releasing the Python package `gengli`. In Chapter 7, my master's student, Robin van der Laag, optimized the pre-processing to run in near real-time, while another master's student, Paloma Laguarta, developed the machine learning model. In Chapter 8, Pablo Cerdá-Durán provided the phenomenological waveforms, and Marco Drago constructed the datasets. In Chapter 9, Stefano Schmidt assisted with part of the data pre-processing, while master's students Ana Martins, Jonno Schoppink, and Wouter van Straalen developed the machine learning model.

Part I

Foundations of Gravitational Wave Astronomy

Chapter 1

Brief Introduction to Gravitational Wave Formalism

Four known forces govern physical interactions of the universe, namely, strong, weak, electromagnetic, and gravitational. Gravity, the weakest of all forces, is commonly experienced daily among earthlings, and its study has puzzled scientists' bright minds since ancient times. However, our understanding of this force is still limited.

In 1687, Sir Isaac Newton published his book *Philosophiae Naturalis Principia Mathematica*, where the laws of motion and universal gravitation are described [1]. Undoubtedly, Newton's *Principia* was a scientific revolution and established the field of classical mechanics. Nonetheless, Newton's theory *describes* gravitational attraction, but there was a need to *explain* the nature of gravity. In 1915, Albert Einstein introduced the theory of general relativity, redefining Newton's law of universal gravitation and providing a unified description of gravity as a geometric property of space and time [2].

In 1916, Albert Einstein found that the linearized weak-field equations of general relativity had wave-like solutions, predicting the existence of gravitational waves (GW) whose amplitude would be insignificantly small [3]. During the 20th century, there was extensive debate about the physical reality of GW. It was not until 1974 that the GW emission from a binary pulsar system was indirectly measured by its orbital changes, leading to a Nobel Prize in 1993 and to the recognition that direct detection of such emission would open a window to probe the physics of our violent universe [4]. A century after the fundamental predictions of Einstein, the Laser Interferometer Gravitational-Wave Observatory (LIGO) and Virgo detected the first GW emission and the first binary black hole merger, known as GW150914, leading to a Nobel Prize in 2017 [5, 6, 7].

In the following Sections, we introduce the theoretical GW formalism: how GW propagate and interact with the test masses of the detectors [8, 9, 10]. Before jumping to the GW formalism, we specify the notation and convention employed in this chapter. We will use natural units for which the velocity of light c and Newton's constant G are unity. We consider a space-time with its coordinates of a global inertial reference frame

$$x^{\mu} = (t, x^1, x^2, x^3) \text{ for } \mu \in \{0, 1, 2, 3\}, \tag{1.1}$$

the Minkowski metric,

$$g_{\mu\nu} = \eta_{\mu\nu} \quad , \text{ where } \eta_{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1), \tag{1.2}$$

and its invariant line element,

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu. \tag{1.3}$$

Given the Christoffel symbols,

$$\Gamma^\rho_{\mu\nu} = \frac{1}{2} g^{\sigma\rho} (\partial_\nu g_{\sigma\mu} + \partial_\mu g_{\sigma\nu} - \partial_\sigma g_{\mu\nu}), \tag{1.4}$$

we define the Riemann tensor as,

$$R^\rho_{\sigma\mu\nu} = \partial_\mu \Gamma^\rho_{\nu\sigma} - \partial_\nu \Gamma^\rho_{\mu\sigma} + \Gamma^\rho_{\mu\lambda}\Gamma^\lambda_{\nu\sigma} - \Gamma^\rho_{\nu\lambda}\Gamma^\lambda_{\mu\sigma}. \tag{1.5}$$

We can contract the Riemann tensor to obtain the Ricci tensor,

$$R_{\mu\nu} = g^{\rho\sigma} R_{\sigma\mu\rho\nu} = R^\rho_{\mu\rho\nu}, \tag{1.6}$$

and contracting one more time, we arrive at the Ricci scalar:

$$R = g^{\mu\nu} R_{\mu\nu}. \tag{1.7}$$

1.1 Linearized Einstein equations

The Einstein field equations are a set of ten coupled differential equations that describe how the curvature of space-time influences the motion of matter, and how the presence of matter at a point affects the average space-time curvature in its neighbourhood. Mathematically,

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = 8\pi T_{\mu\nu}. \tag{1.8}$$

On the left-hand side, we have defined $G_{\mu\nu}$, commonly known as the Einstein tensor, which depends on the description of the geometry of space-time. On the right-hand side, $T_{\mu\nu}$ represents the stress-energy tensor, encoding all energy, momentum and stresses associated with matter.

The geometric description of general relativity allows us to freely choose the coordinate system since it is invariant under the group of all possible coordinate transformations,

$$x^\mu \to x'^\mu(x), \tag{1.9}$$

where $x'^\mu(x)$ is an arbitrary diffeomorphism[1]. Under Eq. 1.9 the metric transforms as,

$$g_{\mu\nu}(x) \to g'_{\mu\nu}(x') = \frac{\partial x^\rho}{\partial x'^\mu} \frac{\partial x^\sigma}{\partial x'^\nu} g_{\rho\sigma}(x). \tag{1.10}$$

We now consider the generation of gravitational waves in the context of linearized theory. This means that we assume that the gravitational field generated by the source is sufficiently weak, so the background space-time can be taken as flat. Therefore,

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}, \quad |h_{\mu\nu}| \ll 1, \tag{1.11}$$

expanding the equations of motion to linear order in $h_{\mu\nu}$, a small perturbation, and its derivatives with respect to coordinates. Since the numerical values of the components of a tensor depend on the reference frame, we are interested in a reference frame where Eq. 1.11 holds. Nonetheless, choosing a reference frame breaks the invariance of general relativity under coordinate transformation, but a residual gauge symmetry remains,

---

[1] A continuously differentiable and invertible mapping between two differentiable manifolds.

$$x^\mu \to x'^\mu = x^\mu + \xi^\mu(x), \qquad \text{where } |\partial_\mu \xi_\nu| = \mathcal{O}(|h_{\mu\nu}|). \tag{1.12}$$

Using the transformation law of the metric Eq. 1.10 and keeping linear terms in $h_{\mu\nu}$ and $\partial_\mu \xi_{nu}$,

$$h_{\mu\nu}(x) \to h'_{\mu\nu}(x') = h_{\mu\nu}(x) - (\partial_\mu \xi_\nu + \partial_\nu \xi_\mu). \tag{1.13}$$

If $|\partial_\mu \xi_\nu| = \mathcal{O}(|h_{\mu\nu}|)$ then $|h_{\mu\nu}| \ll 1$ is preserved, so these slowly varying diffeomorphisms are a symmetry of linearized theory. We can also perform finite global Lorentz transformations

$$x^\mu \to \Lambda^\mu{}_\nu x^\nu, \qquad \text{satisfying } \Lambda_\mu{}^\rho \Lambda_\nu{}^\sigma \eta_{\rho\sigma} = \eta_{\mu\nu}. \tag{1.14}$$

Applying the previous equation under a Lorentz transformation,

$$\begin{aligned} g_{\mu\nu}(x) \to g'_{\mu\nu}(x') &= \Lambda_\mu{}^\rho \Lambda_\nu{}^\sigma g_{\rho\sigma}(x) \\ &= \eta_{\mu\nu} + \Lambda_\mu{}^\rho \Lambda_\nu{}^\sigma h_{\rho\sigma}(x). \end{aligned} \tag{1.15}$$

Therefore,

$$g'_{\mu\nu}(x') = \eta_{\mu\nu} + h'_{\mu\nu}(x'), \qquad \text{with } h'_{\mu\nu}(x') = \Lambda_\mu{}^\rho \Lambda_\nu{}^\sigma h_{\rho\sigma}(x) \tag{1.16}$$

showing that $h_{\mu\nu}$ is a tensor under Lorentz transformations. While rotations never undermine the condition $|h_{\mu\nu}| \ll 1$, boosts must be limited to those fulfilling it. The Riemann tensor in linear order in derivatives of $h_{\mu\nu}$ becomes

$$R_{\mu\nu\rho\sigma} = \frac{1}{2}(\partial_\nu \partial_\rho h_{\mu\sigma} + \partial_\mu \partial_\sigma h_{\nu\rho} - \partial_\mu \partial_\rho h_{\nu\sigma} - \partial_\nu \partial_\sigma h_{\mu\rho}), \tag{1.17}$$

which is invariant under the residual gauge transformation in Eq. 1.12. Thus, the linearized Einstein field equations can be written as

$$\begin{aligned} G_{\mu\nu} &= R_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu} R \\ &= \frac{1}{2}\left(\partial_\sigma \partial_\mu h^\sigma{}_\nu + \partial_\sigma \partial_\nu h^\sigma{}_\mu - \partial_\mu \partial_\nu h - \Box h_{\mu\nu} + \eta_{\mu\nu}(\partial_\rho \partial_\lambda h^{\rho\lambda} + \Box h)\right) = 8\pi T_{\mu\nu}, \end{aligned} \tag{1.18}$$

where $\Box = \eta^{\mu\nu}\partial_\mu \partial_\nu$ is the d'Alembert operator and $h \equiv \eta^{\alpha\beta} h_{\alpha\beta}$. To simplify Eq. 1.18 we introduce the trace-reversed metric perturbation $\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu} h$, so

$$\Box \bar{h}_{\mu\nu} + \eta_{\mu\nu} \partial^\rho \partial^\sigma \bar{h}_{\rho\sigma} - \partial^\rho \partial_\nu \bar{h}_{\mu\rho} - \partial^\rho \partial_\mu \bar{h}_{\nu\rho} = -16\pi T_{\mu\nu}. \tag{1.19}$$

We now want to find a gauge, i.e. a choice of the coordinate system in which the Einstein tensor takes a simplified form. This would be achieved if,

$$\partial^\mu \bar{h}_{\mu\nu} = 0, \tag{1.20}$$

known as *harmonic gauge*. To prove it we use Eq. 1.13 to impose Eq. 1.20:

$$\bar{h}_{\mu\nu} \to \bar{h}'_{\mu\nu} = \bar{h}_{\mu\nu} - (\partial_\mu \xi_\nu + \partial_\nu \xi_\mu - \eta_{\mu\nu} \partial_\rho \xi^\rho) \Rightarrow \partial^\mu \bar{h}_{\mu\nu} \to (\partial^\mu \bar{h}_{\mu\nu})' = \partial^\mu \bar{h}_{\mu\nu} - \Box \xi_\mu \tag{1.21}$$

Thus, if the initial configuration $h_{\mu\nu}$ is such that $\partial^\nu \bar{h}_{\mu\nu} = f_\mu(x)$, where $f_\mu(x)$ is some function, then to obtain $(\partial^\mu \bar{h}_{\mu\nu})' = 0$ we must choose,

$$\Box\xi_\mu = f_\mu(x). \tag{1.22}$$

As the d'Alembertian operator $\Box$ is invertible, it always admits solutions of the form

$$\xi_\mu = \int d^4y G(x-y) f_\mu(y), \tag{1.23}$$

utilizing Green's function. In this gauge, the Einstein field equations become:

$$\Box\bar{h}_{\mu\nu} = -16\pi T_{\mu\nu}. \tag{1.24}$$

While the original representation of Einstein field equations had ten independent components, by introducing four constraints with the harmonic gauge, we reduce $h_{\mu\nu}$ to six independent components. Further specifying the gaugereduces further $h_{\mu\nu}$ to two independent components [9]. Simplifying for consistency Eq. 1.20 and Eq. 1.24, we find the conservation of energy-momentum in the linearized theory,

$$\partial^\mu T_{\mu\nu} = 0. \tag{1.25}$$

1.2 The traverse-traceless gauge

To study the propagation of GW and their interaction with the test masses of a GW detector, we are interested in Eq. 1.24 outside of the source, where

$$\Box\bar{h}_{\mu\nu} = 0, \quad \text{where } T_{\mu\nu} = 0 \text{ (outside the source)}. \tag{1.26}$$

As discussed in the previous Section, Eq. 1.20 does not fix the gauge completely, so we can impose $\Box\xi_\mu = 0$ to satisfy the condition of Eq. 1.20 under coordinate transformation, as defined in Eq. 1.12. A direct consequence of $\Box\xi_\mu = 0$ is that $\Box\xi_{\mu\nu} = 0$, where

$$\Box\xi_{\mu\nu} \equiv \partial_\mu\xi_\nu + \partial_\nu\xi_\mu - \eta_{\mu\nu}\partial_\rho\xi^\rho, \tag{1.27}$$

since the flat space d'Alembertian $\Box$ commutes with ∂_μ. Hence, from Eq. 1.21 we can substract the functions $\xi_{\mu\nu}$ that satisfy $\Box\xi_{\mu\nu} = 0$, and depend on four independent arbitrary functions ξ_μ. Therefore, we can choose ξ^0 such that $\bar{h} = 0$, implying that $\bar{h}_{\mu\nu} = h_{\mu\nu}$, and the three functions $\xi^i(x)$ are chosen so that $h^{0i}(x) = 0$, so

$$\partial^0 h_{00} + \partial^i h_{0i} = 0 \Rightarrow \partial^0 h_{00} = 0. \tag{1.28}$$

A time-independent h_{00} corresponds to the static part of the gravitational interaction, i.e. to the Newtonian potential of the source which generated the GW. Therefore, setting $h_{0\mu} = 0$ leaves only the spatial components of h_{ij}, so we can define the *transverse-traceless* (TT) *gauge*[2]:

$$h^{0\mu} = 0 \textit{ (spatial components)}, \; h^i_i = 0 \textit{ (vanishing trace)}, \; \partial^i h_{ij} = 0 \textit{ (Lorentz gauge)}. \tag{1.29}$$

In this gauge, Eq. 1.26 has plane wave solutions:

$$h^{TT}_{ij}(x) = e_{ij}(\mathbf{k})e^{ikx}, \quad \text{where } k^\mu = (\omega, \mathbf{k}) \text{ and } \omega = |\mathbf{k}|, \tag{1.30}$$

where $e_{ij}(\mathbf{k})$ is the polarization tensor. For a single plane wave and a given wave-vector $\mathbf{k}$, we see from Eq. 1.29 the non-zero components of h^{TT}_{ij} are in the plane transverse to $\hat{\mathbf{n}}$. We chose

---

[2] Quantities in this gauge are denoted by the superscript TT.

$\hat{\mathbf{n}}$ along the z axis, imposing that h_{ij} is symmetric and traceless,

$$h_{ij}^{TT}(t,z) = \begin{pmatrix} h_+ & h_\times & 0 \\ h_\times & -h_+ & 0 \\ 0 & 0 & 0 \end{pmatrix}_{ij} \cos\left[\omega(t-z)\right], \tag{1.31}$$

where h_+ and $h_\times$ are the amplitudes of the "plus" and "cross" polarization of the wave, defined with respect to a given choice of axes in the transverse plane. In terms of the interval ds^2,

$$\begin{aligned} ds^2 = -c^2dt^2 + dz^2 + &\{1 + h_+ \cos\left[\omega(t-z)\right]\}dx^2 \\ &+ \{1 - h_+ \cos\left[\omega(t-z)\right]\}dy^2 + 2h_\times \cos\left[\omega(t-z)\right]dxdy \end{aligned} \tag{1.32}$$

1.3 Interaction with test masses

Having described in the previous Section GW and their propagation, we discuss their interaction with an idealized set of test masses from the GW detector. As GW have a special simple form in the TT gauge, it is relevant to understand which reference frame corresponds to this gauge.

A time-like curve is defined by the condition that $ds^2 < 0$. From this condition, we can define the proper time τ, carried by the clock along the trajectory as $c^2d\tau^2 = -ds^2 = -g_{\mu\nu}dx^\mu dx^\nu$. Thus, we can parametrize the trajectory $x^\mu = x^\mu(\tau)$. Given a fixed boundary condition $x^\mu(\tau_A) = x^\mu_A$ and $x^\mu(\tau_B) = x^\mu_B$, the classical trajectory of a point-like test mass m is obtained by extremizing the action

$$S = -m\int_{\tau_B}^{\tau_A} d\tau, \tag{1.33}$$

yielding the *geodesic equation*,

$$\frac{d^2x^\mu}{d\tau^2} + \Gamma^\mu_{\nu\rho}(x)\frac{dx^\nu}{d\tau}\frac{dx^\rho}{d\tau} = 0. \tag{1.34}$$

Considering two nearby geodesics parametrized by $x^\mu(\tau)$ and $x^\mu(\tau) + \xi^\mu(\tau)$. If $|\xi^\mu| \ll$ than the average variation of the gravitational field, then taking the difference between the two nearby geodesics and expanding to first order in ξ, yields the *geodesic deviation*

$$\frac{d^2\xi^\mu}{d\tau^2} + 2\Gamma^\mu_{\nu\rho}(x)\frac{dx^\nu}{d\tau}\frac{d\xi^\rho}{d\tau} + \xi^\sigma\partial_\sigma\Gamma^\mu_{\nu\rho}(x)\frac{dx^\nu}{d\tau}\frac{d\xi^\rho}{d\tau} = 0, \tag{1.35}$$

simplified as

$$\frac{D^2\xi^\mu}{D\tau^2} = -R^\mu{}_{\nu\rho\sigma}\xi^\rho\frac{dx^\mu}{d\tau}\frac{dx^\sigma}{d\tau}, \tag{1.36}$$

by introducing the covariant derivative of a vector field $V^\mu(x)$ along the curve $x^\mu(\tau)$,

$$\frac{DV^\mu}{D\tau} \equiv \frac{dV^\mu}{d\tau} + \Gamma^\mu_{\nu\rho}V^\nu\frac{dx^\rho}{d\tau}. \tag{1.37}$$

Eq. 1.36, governed by the Riemann tensor, represents the tidal gravitational force experimented by two nearby time-like geodesics. Writing this equation in the reference frame of interest will show the test masses' behaviour for the corresponding observer.

We ask what it means physically to be in the TT gauge. Given a test mass at rest at $\tau = 0$ its geodesic equation is,

$$\left.\frac{d^2x^i}{d\tau^2}\right|_{\tau=0} = -\left[\Gamma^i_{\nu\rho}(x)\frac{dx^\nu}{d\tau}\frac{dx^\rho}{d\tau}\right]_{\tau=0} = -\left[\Gamma^i_{00}\left(\frac{dx^0}{d\tau}\right)^2\right]_{\tau=0}, \quad \text{where } \frac{dx^i}{d\tau} = 0 \tag{1.38}$$

Evaluating the Christoffel symbol, Eq. 1.4, in the TT gauge

$$\Gamma^{i,TT}_{00} = \frac{1}{2}(2\partial_0 h_{0i} - \partial_i h_{00}) = 0, \tag{1.39}$$

due to the TT gauge conditions. This shows that the coordinates of the TT stretch themselves, such that the position of free test masses at rest does not change after the arrival of the wave. Since the coordinate distance of test masses at rest remains constant, also their coordinate separation must remain constant for any arbitrary finite separation. Considering two test masses initially at rest at $\tau = 0$, their geodesic deviation is

$$\left.\frac{d^2\xi^i}{d\tau^2}\right|_{\tau=0} = -\left[2\Gamma^i_{0j}(x)\frac{d\xi^i}{d\tau}\right]_{\tau=0}, \tag{1.40}$$

since the TT gauge both Γ^i_{00} and $\Gamma^i_{0\rho}$ with ρ non-spatial, vanish. Hence, at $\tau = 0$ we have $d\xi^i/d\tau = 0$, so $d^2\xi^i/d\tau^2 = 0$, and the geodesic separation ξ^i remains constant.

The TT gauge coordinate system was chosen such that the position of test masses is unchanged as a GW passes by. Still, physical effects can instead be measured with proper distances or proper times. For illustration, we consider two events at $(t, x_1, 0, 0)$ and at $(t, x_2, 0, 0)$. In the TT gauge the coordinate distance $x_2 - x_1 = L$ remains constant despite the propagation of a GW along the z axis, but from Eq. 1.32 the proper distance s is

$$s = (x_2 - x_1)\left[1 + h_+ \cos(\omega t)\right]^{1/2} \simeq L\left[1 + \frac{1}{2}h_+ \cos(\omega t)\right], \tag{1.41}$$

preserving only linear terms in h_+. Hence, the changes in the proper distance are periodical due GW. Generally, if the spatial separation between two events is given by L and the spatial separation between two test masses is given by L_i, then the proper distance is $s^2 = L^2 + h_{ij}L_iL_j$.

In a laboratory, we do not expect to have free-falling particles, but rather a test mass which is free to move. Such test mass will be displaced by GW with respect to its original position. The simplest experiment to analyze is a free fall laboratory, where if we focus on a sufficiently small region of space, we can choose coordinates $(t, \mathbf{x})$ so that the metric is flat. Expanding at second order in x^i yields the metric of the *proper detector frame*,

$$ds^2 \simeq -c^2dt^2 + \delta_{ij}dx^i dxj + \mathcal{O}\left(\frac{x^ix^j}{L_B}\right), \tag{1.42}$$

where L_B is the typical variation scale of the metric. Now, from Eq. 1.35, the geodesic deviation of two test masses moving non-relativistically in the proper detector frame is

$$\frac{d^2\xi^i}{d\tau^2} + \xi^\sigma\partial_\sigma\Gamma^i_{00}\left(\frac{dx^0}{d\tau}\right)^2 = 0, \tag{1.43}$$

using the fact that $\Gamma^\mu_{\nu\rho}$ is negligible compared to the terms in the free-falling frame, and $dx^i/d\tau$ being neglected with respect to $dx^0/d\tau$. Due to the metric, non-zero contributions come from spatial derivatives that act in x^ix^j, so $\xi^\sigma\partial_\sigma\Gamma^i_{00} = \xi^j\partial_j\Gamma^i_{00}$ are evaluated at the expansion point P. Moreover, at point P, we have $R^i_{0j0} = \partial_j\Gamma^i_{00} - \partial_0\Gamma^i_{0j} = \partial_j\Gamma^i_{00}$ and Eq. 1.43 becomes

$$\frac{d^2\xi^i}{d\tau^2} + R^i{}_{0j0}\xi^j\left(\frac{dx^0}{d\tau}\right)^2 = 0 \tag{1.44}$$

We limit ourselves to linear order in h, so

$$\frac{d^2\xi^i}{dt^2} = -R^i{}_{0j0}\xi^j, \qquad \text{where } t \approx \tau, \tag{1.45}$$

where t is the time coordinate of the proper detector frame. Finally, we compute the

Riemann tensor ${R^i}_{0j0}\xi^j$, which is invariant in the linearized theory (see Eq. 1.12), so we can compute it in the frame of our preference. We employ the TT frame, since in this frame GW have the simple form. From the Riemann tensor, Eq. 1.17, we find

$$ {R^i}_{0j0} = R_{i0j0} = -\frac{1}{2}\frac{d^2 h_{ij}^{TT}}{dt^2}. \tag{1.46} $$

Thus, the equation of the geodesic deviation in the proper detector frame is

$$ \frac{d^2\xi^i}{dt^2} = -\frac{1}{2}\frac{d^2 h_{ij}^{TT}}{dt^2}\xi^j \tag{1.47} $$

which resembles the Newtonian tidal force, allowing a more intuitive description of the detector. One must note that to derive the equation for the geodesic deviation, we have expanded the Christoffel symbols to first order in ξ, under the condition that $\xi^i \ll$ than the average variation of the gravitational field (see Eq. 1.35). For a GW, this length-scale is the reduced wavelength $\lambdabar = \lambda/2\pi$. Hence, if a detector has a characteristic linear size L, we can discuss its interaction with GW using the geodesic deviation, Eq. 1.47, if and only if $L \ll \lambdabar$.

While this condition holds for bar detectors and ground-based interferometers, it does not for space-based detectors, so a full general relativity treatment is needed to study the influence of GW on test masses.

Coming back to our laboratory, to study the effect of a GW on test masses, we consider a ring of test masses located in the (x, y) plane, whose origin lies in the centre of the ring. Initially, the ring is at rest in the proper detector frame. We also consider a GW propagating along the z-axis, such that from Eq. 1.47, the displacement will be confined to the (x, y) plane. As $+$ and $\times$ polarization from Eq. 1.31 are independent, we can consider them separately, so taking for instance $+$ polarization,

$$ h_{ab}^{TT} = h_+ \sin\omega t \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{1.48} $$

at $z = 0$, where $a, b = 1, 2$ are the indices in the transverse plane. The location of a test mass is $\xi_a(t) = (x_0 + \delta x(t), y_0 + \delta y(t))$, where (x_0, y_0) are the unperturbed positions of the test masses and $\delta x(t), \delta y(t)$ are the displacements due to the GW. Thus,

$$ \begin{aligned} \begin{pmatrix} d^2(\delta x)/dt^2 \\ d^2(\delta y)/dt^2 \end{pmatrix}_+ &= -\frac{h_+}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} x_0 + \delta x \\ y_0 + \delta y \end{pmatrix}\omega^2\cos\omega t \\ &\approx -\frac{h_+}{2}\begin{pmatrix} x_0 \\ -y_0 \end{pmatrix}\omega^2\cos\omega t, \end{aligned} \tag{1.49} $$

where we assume that $\delta x, \delta y$ is $\mathcal{O}(h_+)$, so they can be neglected with respect to the constant parts x_0, y_0. Solving Eq. 1.49 for $+$ polarization, and following a similar computation for $\times$ polarization yields,

$$ \begin{pmatrix} \delta x \\ \delta y \end{pmatrix}_+ = \frac{h_+}{2}\begin{pmatrix} x_0 \\ -y_0 \end{pmatrix}\sin\omega t, \quad \begin{pmatrix} \delta x \\ \delta y \end{pmatrix}_\times = \frac{h_\times}{2}\begin{pmatrix} y_0 \\ x_0 \end{pmatrix}\sin\omega t \tag{1.50} $$

In Fig. 1-1 we show the resulting deformation of a ring of test masses in the (x, y) plane.

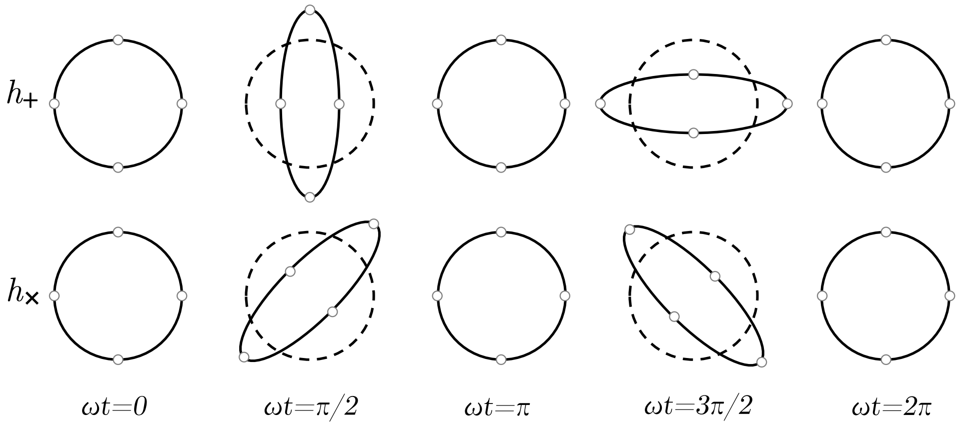


Figure 1-1: A schematic illustration of the deformation of a ring of test masses for different phase values due to the + and × polarization of a passing GW.

1.4 Energy and momentum

In Einstein's linearized theory, the space-time is flat and GW are small fluctuations around it, where, in a suitable gauge, $h_{\mu\nu}$ satisfies the wave equation. Nonetheless, this approach does not describe the energy and momentum associated with GW, since, according to general relativity, any form of energy contributes to the curvature of space-time. Instead, we would like to define GW perturbation over some curved, dynamical, background metric $\bar{g}_{\mu\nu}(x)$, so

$$g_{\mu\nu}(x) = \bar{g}_{\mu\nu}(x) + h_{\mu\nu}(x), \qquad \text{where } |h_{\mu\nu}| \ll 1. \tag{1.51}$$

While in the linearized theory, the background metric was chosen to be the constant flat-space metric $\eta_{\mu\nu}$, in this dynamical metric $g_{\mu\nu}(x)$ we need to separate the background metric $\bar{g}_{\mu\nu}(x)$ from the fluctuations $h_{\mu\nu}$. In the most general setting, there is no unambiguous way to perform this separation. This situation is similar to the waves in the sea: there is no unambiguous way to state which part of the movement of the surface of the water belongs to a given wave, and which part belongs to a "background" originated by the incoherent superposition of perturbations of varied origin.

To understand how the perturbation of $h_{\mu\nu}$ propagates and how it affects the background of space-time, we begin by expanding Einstein equations around the background metric $\bar{g}_{\mu\nu}$

$$g_{\mu\nu}(x) = \bar{g}_{\mu\nu}(x) + h^{(1)}_{\mu\nu}(x) + h^{(2)}_{\mu\nu}(x) + \dots, \tag{1.52}$$

where $h^{(1)}_{\mu\nu}(x)$ and $h^{(2)}_{\mu\nu}(x)$ are first and second order perturbations describing GW. We cast the Einstein equations in the form

$$R_{\mu\nu} = 8\pi G(T_{\mu\nu} - \frac{1}{2}g_{\mu\nu}T), \tag{1.53}$$

where $T_{\mu\nu}$ is the energy-momentum tensor of matter and T its trace. Expanding the Ricci tensor,

$$R_{\mu\nu} = \bar{R}_{\mu\nu} + R^{(1)}_{\mu\nu} + R^{(2)}_{\mu\nu} + \dots, \tag{1.54}$$

where $\bar{R}_{\mu\nu}$ is constructed with $\bar{g}_{\mu\nu}$, containing only low frequency modes. $R^{(1)}_{\mu\nu}$ is linear in $h_{\mu\nu}$, so it contains only high-frequency modes. $R^{(2)}_{\mu\nu}$ is quadratic in $h_{\mu\nu}$ and therefore contains bot high and low frequencies. Therefore, the Einstein equations can be split into equations for low and high frequencies, yielding respectively

$$\bar{R}_{\mu\nu} = -\left[R^{(2)}_{\mu\nu}\right]^{\text{Low}} + 8\pi G(T_{\mu\nu} - \frac{1}{2}g_{\mu\nu}T)^{\text{Low}} \tag{1.55}$$

$$R^{(1)}_{\mu\nu} = -\left[R^{(2)}_{\mu\nu}\right]^{\text{High}} + 8\pi G(T_{\mu\nu} - \frac{1}{2}g_{\mu\nu}T)^{\text{High}}, \tag{1.56}$$

Defining $\bar{\nabla}_\mu$ as the covariant derivative with respect to the background metric $\bar{g}_{\mu\nu}$,

$$R^{(1)}_{\mu\nu} = \frac{1}{2}(\bar{\nabla}^\alpha\bar{\nabla}_\mu h_{\nu\alpha} + \bar{\nabla}^\alpha\bar{\nabla}_\nu h_{\mu\alpha} - \bar{\nabla}^\alpha\bar{\nabla}_\alpha h_{\mu\nu} - \bar{\nabla}_\nu\bar{\nabla}_\mu h), \tag{1.57}$$

where $h = g^{\alpha\beta}h_{\alpha\beta}$ is the trace of $h_{\mu\nu}$ with respect to the background metric $\bar{g}_{\mu\nu}$. At quadratic order we find,

$$\begin{aligned} R^{(2)}_{\mu\nu} = \frac{1}{2}\bar{g}_{\rho\sigma}\bar{g}^{\alpha\beta}\Big[&\frac{1}{2}\bar{\nabla}_\mu h_{\rho\alpha}\bar{\nabla}_\nu h_{\sigma\beta} + (\bar{\nabla}_\rho h_{\nu\alpha})(\bar{\nabla}_\sigma h_{\mu\beta} - \bar{\nabla}_\beta h_{\mu\sigma}) \\ &+ h_{\rho\alpha}(\bar{\nabla}_\nu\bar{\nabla}_\mu h_{\sigma\beta} + \bar{\nabla}_\beta\bar{\nabla}_\sigma h_{\mu\nu} - \bar{\nabla}_\beta\bar{\nabla}_\nu h_{\mu\sigma} - \bar{\nabla}_\beta\bar{\nabla}_\mu h_{\nu\sigma}) \\ &+ (\frac{1}{2}\bar{\nabla}_\alpha h_{\rho\sigma} - \bar{\nabla}_\rho h_{\alpha\sigma})(\bar{\nabla}_\nu h_{\mu\beta} + \bar{\nabla}_\mu h_{\nu\beta} - \bar{\nabla}_\beta h_{\mu\nu})\Big] \end{aligned} \tag{1.58}$$

Setting $T_{\mu\nu} = 0$, we see that in Eq. 1.55 $\bar{R}_{\mu\nu}$ is governed by $[R^{(2)}_{\mu\nu}]^{\text{Low}}$. From Eq. 1.58, we observe that $R^{(2)}_{\mu\nu}$ is a sum of terms of order $(\partial h_{\alpha\beta})^2$ and $(h_{\alpha\beta}\partial h_{\gamma\delta})^2$, such that, in the low modes these two terms give contributions of the same order of magnitude. Thus, in order of magnitude $\bar{R}_{\mu\nu} \sim (\partial h_{\alpha\beta})^2$, which implies that the derivatives of the perturbation $h_{\mu\nu}$ affect the curvature of the background metric $\bar{g}_{\mu\nu}$.

The variation scale of $\bar{g}_{\mu\nu}$ and h are L_B and $\bar{\lambda}$, respectively. In order of magnitude, $\partial\bar{g}_{\mu\nu} \sim 1/L_B$ and $\partial h_{\mu\nu} \sim h_{\mu\nu}/\bar{\lambda}$. From the relation between the background curvature $\bar{R}_{\mu\nu}$ with, on one hand, the second derivative of the background metric, and on the other hand, the derivatives of the perturbation,

$$\bar{R}_{\mu\nu} \sim \partial^2\bar{g}_{\mu\nu} \sim \frac{1}{L_B^2}, \quad \bar{R}_{\mu\nu} \sim (\partial h_{\mu\nu})^2 \Rightarrow \frac{1}{L_B^2} \sim \left(\frac{h_{\mu\nu}}{\bar{\lambda}}\right)^2 \Rightarrow h_{\mu\nu} \sim \frac{\bar{\lambda}}{L_B}, \tag{1.59}$$

which is the curvature determined by GW. Considering a non-vanishing $T_{\mu\nu}$, the GW contribution to the background curvature is negligible compared to the contribution of matter sources. Thus,

$$\frac{1}{L_B^2} \sim \left(\frac{h_{\mu\nu}}{\bar{\lambda}}\right)^2 + \text{(matter contribution)} \gg \frac{h^2_{\mu\nu}}{\bar{\lambda}^2} \Rightarrow h_{\mu\nu} \ll \frac{\bar{\lambda}}{L_B}, \tag{1.60}$$

which is the curvature determined by matter. Suppose we force the background metric to be $\eta_{\mu\nu}$. In that case, we force strictly $1/L_B = 0$, so any arbitrarily small but finite value of $h_{\mu\nu}$ violates the condition in Eq. 1.60. Consequently, to compute higher-order corrections we cannot insist on a flat background metric. From Eq. 1.59 and Eq. 1.60 we also found that GW are only well defined for $h_{\mu\nu} \ll 1$. If $h_{\mu\nu} \sim 1$ the separation between GW and background vanishes.

We introduce a scale $\bar{l}$, such that $\bar{\lambda} \ll \bar{l} \ll L_B$. We average over a spatial volume with side $\bar{l}$, denoted $\left\langle \dots \right\rangle_{\bar{l}}$. In this average, modes on the scale of order L_B remain constant, while modes on a scale of order $\bar{\lambda}$ average to zero. The basic idea of this procedure is to “integrate out” the fluctuations that take place on a length-scale $< \bar{l}$, to obtain an effective theory that describes the physics at the length-scale $\bar{l}$. We can rewrite Eq. 1.55 as

$$\bar{R}_{\mu\nu} = -\left\langle R^{(2)}_{\mu\nu} \right\rangle_l + 8\pi \left\langle T_{\mu\nu} - \frac{1}{2} g_{\mu\nu} T \right\rangle_l, \tag{1.61}$$

where condition $ƛ \ll L_B$. We define an effective energy-momentum tensor of matter $\bar{T}^{\mu\nu}$,

$$\left\langle T_{\mu\nu} - \frac{1}{2} g_{\mu\nu} T \right\rangle_l = \bar{T}^{\mu\nu} - \frac{1}{2} \bar{g}_{\mu\nu} \bar{T}, \tag{1.62}$$

where $\bar{T} = \bar{g}_{\mu\nu} \bar{T}^{\mu\nu}$ is the trace. By definition, $\bar{T}^{\mu\nu}$ is a low-frequency quantity and smoothed form of the matter energy-momentum tensor $T_{\mu\nu}$. We can visualize it as a "macroscopic" version (with respect to the scale $ƛ$), while $T_{\mu\nu}$ is the "microscopic" quantity. We also define the stress-energy tensor of GW as

$$t_{\mu\nu} = -\frac{1}{8\pi} \left\langle R^{(2)}_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R^{(2)} \right\rangle_l, \text{ where } R^{(2)} = \bar{g}^{\mu\nu} R^{(2)}_{\mu\nu}, \tag{1.63}$$

and its trace

$$t = \bar{g}^{\mu\nu} t_{\mu\nu} = +\frac{1}{8\pi} \left\langle R^{(2)} \right\rangle_l. \tag{1.64}$$

Inserting Eq. 1.63 in Eq. 1.64 yields,

$$-\left\langle R^{(2)}_{\mu\nu} \right\rangle_l = 8\pi \left(t_{\mu\nu} - \frac{1}{2} \bar{g}_{\mu\nu} t \right). \tag{1.65}$$

So, we can rewrite Eq. 1.61 as

$$\bar{R}_{\mu\nu} - \frac{1}{2} \bar{g}_{\mu\nu} \bar{R} = 8\pi (\bar{T}_{\mu\nu} + t_{\mu\nu}), \tag{1.66}$$

known as the "coarse-grained" form of the Einstein equations, which determine the dynamics of the low-frequency part of the metric $\bar{g}_{\mu\nu}$ in terms of the matter energy-momentum tensor $\bar{T}_{\mu\nu}$, and the tensor $t_{\mu\nu}$, which depends only on the gravitational field.

According to the definition of $t_{\mu\nu}$ in Eq. 1.63, it has both physical contributions and coordinate-dependent contributions. It is relevant to distinguish both contributions, as the first one will be associated with the energy-momentum tensor of GW, while the second one will be associated with the coordinate system, and can vanish with an appropriate gauge choice.

The most straightforward way to get only the physical contribution is to use the harmonic gauge condition from Eq. 1.20. We can rewrite Eq. 1.58 since we are now interested in the energy and momentum carried by GW at large distances from the source, where we can approximate the background space-time as flat, such that we can replace $\bar{\nabla}^\mu \to \partial_\mu$. We can integrate the space-time derivative ∂_μ, neglecting the boundary term. Using the gauge conditions $\partial^\mu h_{\mu\nu} = 0$ and $\eta^{\mu\nu} h_{\mu\nu} = 0$, and from the equation of motion $\Box h_{\mu\nu}$, we see that most terms collapse to zero, yielding

$$\left\langle R^{(2)}_{\mu\nu} \right\rangle_l = -\frac{1}{4} \left\langle \partial_\mu h_{\alpha\beta} \partial_\nu h^{\alpha\beta} \right\rangle_l. \tag{1.67}$$

$\left\langle R^{(2)} \right\rangle_l = 0$ upon integration by parts, so from Eq. 1.63 we have

$$t_{\mu\nu} = \frac{1}{32\pi} \left\langle \partial_\mu h_{\alpha\beta} \partial_\nu h^{\alpha\beta} \right\rangle_l, \tag{1.68}$$

which is invariant under Eq. 1.12 and only depends on the physical modes h^{TT}_{ij}. Finally, from Eq. 1.66 we observe that the left-hand side is covariantly conserved with respect to $\bar{\nabla}^\mu$, so $\bar{\nabla}^\mu (\bar{R}_{\mu\nu} - \frac{1}{2} \bar{g}_{\mu\nu} \bar{R}) = 0$, because of the so-called Bianchi identity [9]. Thus,

$$\bar{\nabla}^{\mu}(\bar{T}_{\mu\nu}+t_{\mu\nu}), \tag{1.69}$$

which reflects that there is in general exchange of energy and momentum between the matter sources and GW. As before, far from the sources $\bar{\nabla}^{\mu} \to \partial^{\mu}$ and outside the source $\bar{T}_{\mu\nu}=0$, such that $\partial^{\mu}t_{\mu\nu}=0$, which is the conservation of the energy-momentum tensor. Under the conditions of the TT gauge at a large distance r from the source, we find the energy given by

$$\frac{dE}{dt}=\frac{r^2}{32\pi}\int d\Omega\Big\langle\frac{\partial h_{ij}^{TT}}{\partial t}\frac{\partial h_{ij}^{TT}}{\partial t}\Big\rangle_{l}, \tag{1.70}$$

and the momentum is given by

$$\frac{dP^k}{dt}=-\frac{r^2}{32\pi}\int d\Omega\Big\langle\frac{\partial h_{ij}^{TT}}{\partial t}\partial^k\left(\frac{\partial h_{ij}^{TT}}{\partial t}\right)\Big\rangle_{l}. \tag{1.71}$$

1.5 Generation of gravitational waves

From the linearized theory, the Einstein field equation in the harmonic gauge (Eq. 1.20) is,

$$\Box\bar{h}_{\mu\nu}=\text{-}16\pi T_{\mu\nu}, \tag{1.72}$$

where $T_{\mu\nu}$ is the energy-momentum tensor of matter, which satisfies the flat-space conservation law $\partial^{\mu}T_{\mu\nu}=0$. Since Eq. 1.72 is linear in $h_{\mu\nu}$ it can be solved with Green's function, such that its solution is

$$\bar{h}_{\mu\nu}(x)=-16\pi\int d^4x' G(x-x')T_{\mu\nu}(x'). \tag{1.73}$$

Using the appropriate boundary conditions we find that for a radiation problem, the appropriate solution is the retarded Green's function,

$$G(x-x')=-\frac{1}{4\pi|\mathbf{x}-\mathbf{x}'|}\delta(x^0_{ret}-x'^0), \quad \text{where } x'^0=t',\ x^0_{ret}=t_{ret},\ t_{ret}=t-|\mathbf{x}-\mathbf{x}'|, \tag{1.74}$$

where t_{ret} is the retarded time. Hence, since $\mathbf{x}'$ is restricted to the inside part of the source, the solution is

$$\bar{h}_{\mu\nu}(t,\mathbf{x})=4\int d^3x'\frac{1}{|\mathbf{x}-\mathbf{x}'|}T_{\mu\nu}(t-|\mathbf{x}-\mathbf{x}'|,\mathbf{x}'). \tag{1.75}$$

We can project this solution in the TT gauge introducing the symmetric and transverse tensor $P_{ij}(\hat{\mathbf{n}})=\delta_{ij}-n_in_j$, where $\hat{\mathbf{n}}$ is the direction of propagation of the GW. Thus, we define

$$\Lambda_{ij,kl}(\hat{\mathbf{n}})=P_{ik}P_{jl}-\frac{1}{2}P_{ij}P_{kl} \tag{1.76}$$

which is a projector since $\Lambda_{ij,kl}\Lambda_{kl,mn}=\Lambda_{ij,mn}$. Moreover, it is transverse on all indices $n^i\Lambda_{ij,kl}=0,\ldots$, and traceless with respect to (i,j) and (k,l) indices, $\Lambda_{ii,kl}=\Lambda_{ij,kk}=0$. Therefore, outside of the source

$$h_{ij}^{TT}(t,\mathbf{x})=\Lambda_{ij,kl}(\hat{\mathbf{n}})\bar{h}_{kl}=4\Lambda_{ij,kl}(\hat{\mathbf{n}})\int d^3x'\frac{1}{|\mathbf{x}-\mathbf{x}'|}T_{kl}(t-|\mathbf{x}-\mathbf{x}'|,\mathbf{x}'), \tag{1.77}$$

h_{ij}^{TT} depends only on the integrals of the spatial components T_{kl}. We can eliminate T_{0k} and T_{00} since they are related to T_{kl} by the conservation of the energy-momentum tensor. We denote $d \ll r$ as the typical radius of the source, and we can expand

$$|\mathbf{x}-\mathbf{x}'| = r - \mathbf{x}'\cdot(\hat{\mathbf{n}}) + \mathcal{O}\left(\frac{d^2}{r}\right). \tag{1.78}$$

We are interested in h_{ij}^{TT} at large distances from the source, where the detector is located, such that $r \to \infty$ at fixed t. At large distances,

$$h_{ij}^{TT}(t,\mathbf{x}) = \frac{4}{r}\Lambda_{ij,kl}(\hat{\mathbf{n}})\int d^3x' \int \frac{d\omega}{2\pi}\frac{dk}{(2\pi)^3}\tilde{T}_{kl}e^{-i\omega(t-r)+(\mathbf{k}-\omega\hat{\mathbf{n}})\cdot x'} \tag{1.79}$$

where $\tilde{T}_{kl}$ is the Fourier transform of T_{kl} and we neglect terms $(1/r^2)$. This equation is valid both for relativistic and for non-relativistic sources, as long as the linearized theory applies and r is sufficiently large.

The equations for the generation of radiation are greatly simplified if the typical velocities inside the source are small compared to the speed of light. We define ω_s as the typical frequency of motion inside the source, such that the typical velocities inside the source are $v \sim \omega_s d$. The frequency of the radiation will be $\omega \sim \omega_s$. In terms of $\lambdabar = c/\omega$

$$\lambdabar \sim \frac{c}{v}d \tag{1.80}$$

As before, $\lambdabar$ represents the reduced wavelength and d the characteristic size of the source. In a non-relativistic system $v \ll c$, such that $\lambdabar \gg d$. When $\lambdabar$ is much bigger than the size of the system, we only need to know the coarse features of the system, so the emission of radiation is governed by the lowest multipole. Hence, we perform a multipole expansion for gravitational radiation, starting from h_{ij}^{TT} at spatial infinity in terms of Fourier transform, given by Eq. 1.79.

For a non-relativistic source $\hat{T}_{kl}(\omega,\mathbf{k})$ is peaked around a typical frequency ω_s with $\omega_s \ll c$. The energy-momentum tensor is non-vanishing only inside the source, so $|x'| \leq d$. Then, the dominant contribution to h_{ij}^{TT} comes from frequencies ω that satisfy $\omega x'\cdot\hat{\mathbf{n}} \lesssim \omega_s d \ll 1$. Hence, we can expand the exponential in Eq. 1.79

$$e^{-i\omega(t-r+x'\cdot\hat{\mathbf{n}}))} = e^{-i\omega(t-r)} \times \left[1 - i\omega x'^i n^i + \frac{1}{2}(-i\omega)^2 x'^i x'^j n^i n^j + \dots\right] \tag{1.81}$$

Now, we define the moments of the stress tensor T^{ij},

$$S^{ij}(t) = \int d^3x T^{ij}(t,\mathbf{x}), \tag{1.82}$$

$$S^{ij,k}(t) = \int d^3x T^{ij}(t,\mathbf{x})x^k, \tag{1.83}$$

$$S^{ij,kl}(t) = \int d^3x T^{ij}(t,\mathbf{x})x^k x^l, \tag{1.84}$$

and similarly for all higher-order moments. In this notation, a comma separates the spatial indices which originate from T^{ij} from the indices due to the expansion. The energy-momentum tensor of matter is symmetric, $T^{ij} = T^{ji}$. Then its moments are symmetric in the first type of indices and the second type, e.g. $S^{ij,k} = S^{ji,k}$, $S^{ij,kl} = S^{ij,lk}$, $S^{ij,k} \neq S^{ik,j}$. Inserting Eq. 1.81 in Eq. 1.79 yields

$$h_{ij}^{TT}(t,\mathbf{x}) = \frac{4}{r}\Lambda_{ij,kl}(\hat{\mathbf{n}}) \times \left[S^{kl} + n_m\frac{d}{dt}S^{kl,m} + \frac{1}{2}n_m n_p\frac{d}{dt^2}S^{kl,mp} + \dots\right]_{t-r}, \tag{1.85}$$

evaluated at the retarded time $t-r$. From Eq. 1.82 − 1.84, we can see that every higher term has an additional factor $x^m \sim \mathcal{O}(d)$, while each time derivative adds a factor $\mathcal{O}(\omega_s)$. Thus, in Eq. 1.85, every successive term has an additional factor $\mathcal{O}(\omega_s d)$ which implies a $\mathcal{O}(v/c)$

correction with respect to the previous term.

The moments of the energy density M and the moments of the momentum density P are

$$\begin{aligned} M &= \int d^3x T^{00}(t,\mathbf{x}), & P^i &= \int d^3x T^{0i}(t,\mathbf{x}) \\ M^i &= \int d^3x T^{00}(t,\mathbf{x}) x^i, & P^{i,j} &= \int d^3x T^{0i}(t,\mathbf{x}) x^j \\ M^{ij} &= \int d^3x T^{00}(t,\mathbf{x}) x^i x^j, & P^{i,jk} &= \int d^3x T^{0i}(t,\mathbf{x}) x^j x^k \end{aligned} \tag{1.86}$$

The time derivatives of these quantities and the moments T^{ij} satisfy the energy-momentum conservation $\partial_\mu T^{\mu\nu} = 0,$ with non-linearities being neglected, as we are working within the linearized theory. This means that we are also neglecting the back-action of the GW on the source.

$$\begin{aligned} \frac{d}{dt}M = 0 \ \textit{(mass conservation)}\ , &\quad \frac{d}{dt}P^i = 0 \ \textit{(total momentum conservation)} \\ \frac{d}{dt}M^i = P^i, &\quad \frac{d}{dt}P^{i,j} = S^{i,j} \\ \frac{d}{dt}M^{ij} = P^{i,jk} + P^{j,ki} + P^{k,ij}, &\quad \frac{d}{dt}P^{i,jk} = S^{ij,k} + S^{ik,j} \end{aligned} \tag{1.87}$$

While a physical system that radiates GW loses mass, in this approximation, the mass M of the radiating body is conserved due to neglecting the back action of the source dynamics because of the energy carried away by GW. With these equations we can express the multipole expansion, Eq. 1.85, in terms of $\{M, M^i, M^{ij}, \ldots\}$ and $\{P^i, P^{i,j}, \ldots\}$, which have a more immediate physical interpretation. Combining $dM^i/dt = P^i$ and $dP^{i,j}/dt = S^{i,j}$, we can rewrite the leading-order contribution to Eq. 1.85 as

$$\left[h_{ij}^{TT}(t,\mathbf{x})\right] = \frac{2}{r}\Lambda_{ij,kl}(\hat{\mathbf{n}})\frac{dM^{kl}}{dt^2}(t-r) \tag{1.88}$$

Introducing the reduced quadrupole moment as

$$Q^{ij} \equiv M^{ij} - \frac{1}{3}\delta^{ij}M_{kk} = \int d^3x \rho(t,\mathbf{x})(x^i x^j = \frac{1}{3}r^2\delta^{ij}), \tag{1.89}$$

where M_{kk} is the trace of M^{ij}. Using the fact that $\Lambda_{ij,kl}$ vanishes when contracted with δ^{ij}, so $\Lambda_{ij,kl}M_{kl} = \Lambda_{ij,kl}Q_{kl}$. Hence, Eq. 1.88 in terms of the reduced quadrupole yields

$$\left[h_{ij}^{TT}(t,\mathbf{x})\right] = \frac{2}{r}\Lambda_{ij,kl}(\hat{\mathbf{n}})\frac{dQ_{kl}}{dt^2}(t-r) \equiv \frac{2}{r}\frac{dQ_{ij}^{TT}}{dt^2}(t-r) \tag{1.90}$$

When the direction of propagation $\hat{\mathbf{n}} = z$, we evaluate Eq. 1.90 and compare to Eq. 1.48

$$h_+ = \frac{1}{r}\left(\frac{dM_{11}}{dt^2} - \frac{dM_{22}}{dt^2}\right), \tag{1.91}$$

$$h_+ = \frac{2}{r}\left(\frac{dM_{12}}{dt^2}\right). \tag{1.92}$$

Chapter 2

Gravitational Waves: Sources, Detectors and Analysis

As we have seen in the previous chapter, GW are ripples in space-time, and due to the fall-off of their amplitude with distance, they exhibit minuscule amplitudes. After a century since the formulation of GW formalism, and nearly four decades of technological advancements dedicated to constructing GW detectors, scientists successfully measured this space-time deformation in 2015. Nowadays, GW detectors have discovered over 90 GW signals, probing down to the densest and most energetic regions of cosmic objects, which were hidden from astronomers' sight up until now [11, 12, 13]. Furthermore, novel astronomical detections are expected with the upgrade of second-generation detectors, as well as the construction of third-generation detectors, such as the Laser Interferometer Space Antenna (LISA), Einstein Telescope and Cosmic Explorer [14, 15, 16].

In this chapter, we present an overview of GW sources detectable by current and/or future ground-based detectors. We also introduce the current state-of-the-art of GW detectors and their noise sources, as well as introducing basic data analysis techniques.

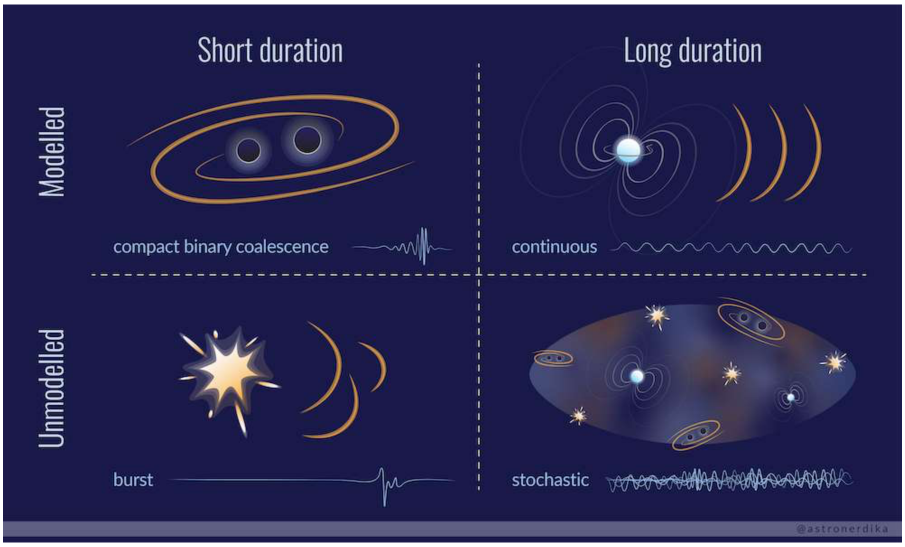


FIGURE 2-1: *The different types of GW sources can be differentiated by their duration (short/long) and their associated astronomical models (modelled/unmodelled). Further details of the sources are described in the main text. Credits: Shanika Galaudage.*

2.1 Gravitational-wave sources

In Fig. 2-1 we can observe an overview of GW sources classified according to their duration (short/long), and if astrophysical models are associated with the source (modelled/unmodelled). In the next subsections, we provide an overview of sources of interest for ground-based detectors.

2.1.1 Compact binary coalescence

The main type of short duration and modelled GW sources are known as Compact Binary Coalescence (CBC). These coalescing systems are composed of two compact bodies, either binary black holes (BBH), binary neutron stars (BNS) or one neutron star and one black hole (NSBH). In this context, "compactness" is the ratio between the mass and the radius of the object, which is related to the strength of the GW emission, and as a consequence, it provides insights into the dynamics and astrophysical properties of the binary system.

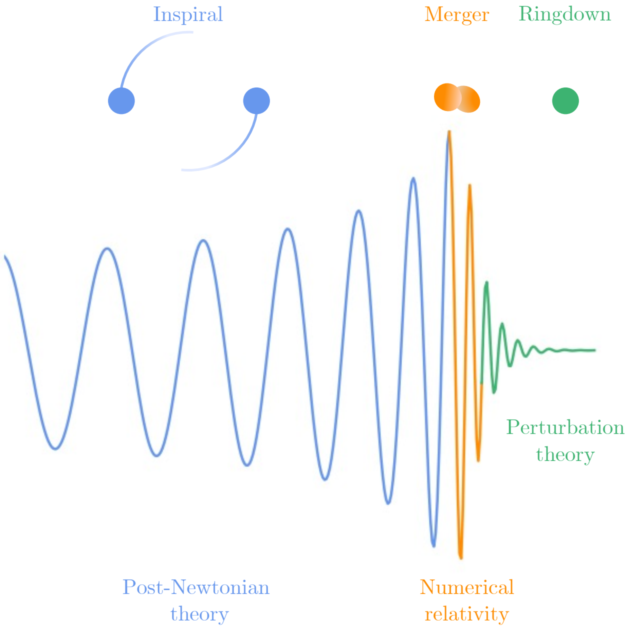


Figure 2-2: Temporal evolution of a binary system with component masses $m_1 = m_2 = 20M_\odot$. We colour in blue the inspiral, in orange the merger, and in green the ringdown.

As these two bodies orbit each other, they emit GW radiation while losing orbital energy and angular momentum, shrinking their orbit (*inspiral phase*). Eventually, an astronomical cataclysm occurs with their abrupt collision (*merger phase*). The remnant of this coalescence will quickly return to ground-state (*ringdown phase*). As we show in Fig. 2-2, the inspiral phase is modelled using post-Newtonian expansion (see Section 1.5). The merger phase is modelled with numerical relativity since in general relativity the two-body problem is not analytically solvable. The ringdown phase is modelled using perturbation theory, where the resulting compact object from the coalescence, known as *remnant* returns to ground-state "ringing" like a bell so that the resulting GW is a superposition of damped sinusoids, known as quasi-normal modes.

Since this GW sources are the most understood, we can use modelled algorithms like matched filtering techniques to detect them (see 3.2.1 for details). However, we can also use weakly modelled or model-free algorithms, as in the case of intermediate-mass black holes [17].

2.1.2 Transient bursts sources

Short and unmodelled GW sources are known as bursts. Burst can be short, up to a few seconds duration, or long, up to $\sim 10^3$ s duration. In this work, we will focus on short-duration GW transients, which include but are not limited to, core-collapse supernovae [18] and cosmic strings [19]. These sources are generally unmodelled, due to either unknown theoretical background and/or complex dynamics of the system. Since burst searches are meant to detect the unexpected, the unmodelled search algorithms employed use minimal (targeted search) or no assumptions (generic search) about the source GW signal [20, 21, 22]. As we have seen in the previous Section, model-free algorithms can also be sensitive to CBC.

2.1.3 Continuous wave sources

Persistent (long duration) and modelled GW are known as continuous waves, which are quasi-monochromatic signals with roughly constant frequency and amplitude compared to the observation time. Sources of continuous GW are single rapidly rotating non-axisymmetric massive objects, such as neutron stars with either a deformation on the surface of the star or due to some fundamental oscillation mode [23]. Continuous waves could also be produced by exotic objects, such as the annihilation of ultra-light boson clouds around spinning black holes [24, 25].

2.1.4 Stochastic background

Persistent (long duration) and unmodelled GW is known as the Stochastic background of GW. This background could be the result of the superposition of incoherent GW signals. It could arise from cosmological sources, such as the inflationary epoch, first-order phase transitions in the early universe or cosmic strings; or from astrophysical sources, such as supernovae or the inspiral and merger of CBC over the history of the universe (see [26] for a comprehensive review). LIGO and Virgo have placed upper bounds on the energy density of the stochastic background in the range $[20, 10^3]$ Hz, by calculating the cross-correlation between pairs of detectors in the search of an excess in the distribution [27].

2.2 Detectors through history

In the 1960s, Joseph Weber began experiments to detect GW with his resonant mass detectors, which measure the oscillations of a bar caused by a passing GW. Weber's detector reached a sensitivity of 10^{-16} m, achieving an important milestone towards GW detection [28]. In 1969 he claimed to have observed signals from GW, but his results remained unreproducible [29].

The concept of interferometric detectors emerged in the early 1960s and 1970s, with the fundamental design resembling that of a Michelson-Morley interferometer, which was initially conceived in 1887 to prove the existence of *luminiferous ether*: a hypothetical medium for the propagation of light waves. Their experiment measured the relative motion between Earth and such medium, not only finding null results for the existence of *luminiferous ether* but also suggesting that the speed of light is constant and independent of the observer's motion, playing a pivotal role in the development of special relativity. Furthermore, since the Michelson-Morley interferometer was designed to measure the relative length changes of two perpen-

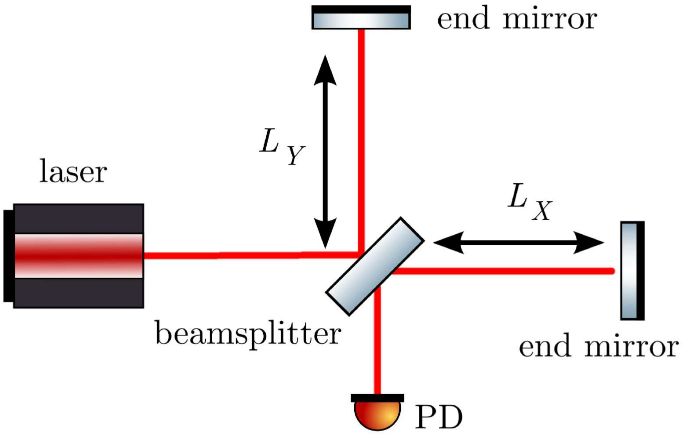


Figure 2-3: A schematic setup of a Michelson-Morley interferometer retrieved from [30].

dicular directions, it can serve as a GW detector by measuring the deformation of the test masses (see Section 1.4) [31, 30].

The individual elements composing modern GW interferometric detectors, such as lasers and mirrors, are individually well described by mostly classical physics and the complexity in the detector's behaviour arises from the combination of these elements into an optical cavity to enhance the interaction of light waves. As the description of a GW interferometer is a complex topic, the interested reader can refer to [30], but in this work, we provide a basic overview of the detector. In Fig. 2-3 we present a schematic illustration of a Michelson-Morley interferometer, composed of a laser, a beam splitter (a 50% reflecting mirror), two highly reflecting end mirrors and a photodetector (PD). The three mirrors can be considered free-falling objects in the horizontal direction, as they are suspended as pendulums, only allowed to swing freely in that direction. As we can observe in Fig. 2-3, the reflecting mirrors are placed equidistant from the beamsplitter, at a distance L_X and L_Y for the x- and y-axis respectively, but in orthogonal directions. The light beam input from the laser incides in the beam splitter, where it splits it into two beams. These two beams travel through the detector's arms until they reach the highly reflecting end mirrors, being redirected to the beamsplitter, where they are recombined and detected at the PD. When the beam recombines, it will interfere constructively or destructively if the lengths of the two arms differ by an even or odd number of wavelengths.

$$\Delta L \equiv L_X - L_Y = \begin{cases} n\lambda, & \text{(constructive interference)} \\ \left(n+\frac{1}{2}\right)\lambda, & \text{(destructive interference)} \end{cases} \tag{2.1}$$

where $n \in [0, 1, \ldots, N]$. As we saw in Section 1.3, in the proper detector frame the passage of a GW along the z-axis causes a displacement of the test masses in the $x-y$ plane from their original position. This introduces a relative temporary change in the light path on the x-axis with respect to the y-axis, that can be measured with interferometric GW detectors. As photons travel through null geodesics, which implies that the interval $ds^2 = 0$ (see Eq. 1.32), this relative change can be expressed as,

$$dx = \frac{dt}{\sqrt{1 + h_+ \cos\left[w(t-z)\right]}} \approx \left(1 - \frac{h_+ \cos\left[w(t-z)\right]}{2}\right) dt = L_x dt, \tag{2.2}$$

$$dy = \frac{dt}{\sqrt{1 - h_+ \cos\left[w(t-z)\right]}} \approx \left(1 + \frac{h_+ \cos\left[w(t-z)\right]}{2}\right) dt = L_y dt \tag{2.3}$$

where we have considered only a +-polatized GW and performed a Taylor expansion around $h = 0$. Therefore, we can compute the difference in path length between x and y arms,

$$\Delta L = L_x - L_y = h dt = Lh \implies h(t) = \frac{\Delta L}{L} \tag{2.4}$$

where $L = dt$ is the unperturbed path length. From Eq. 2.4 we can observe that a passing GW produces a fractional change in distance in the detector, generating an output called *strain*. On the other hand, Eq. 2.1 indicates such strain will cause a phase shift detectable by the PD.

Current GW interferometric detectors are modified Michelson-Morley interferometers, that form a global infrastructure for the discovery and study of GW. Nowadays, the network is formed by two Advanced LIGO [5], one located in Hanford, Washington (USA) and another one in Livingston, Louisiana (USA), Advanced Virgo [6], located in Cascina (Italy), GEO 600 [32], located in Hanover (Germany) and Kamioka Gravitational Wave Detector or KAGRA [33], located in the Gifu-prefecture (Japan). In ~~the next~~ two decades, we expect to improve the current advanced detectors at A$^\sharp$ sensitivity, as well as the addition of LIGO Aundha (India) [34, 35, 36]. Furthermore, we also expect the launch of LISA [14], as well as the construction

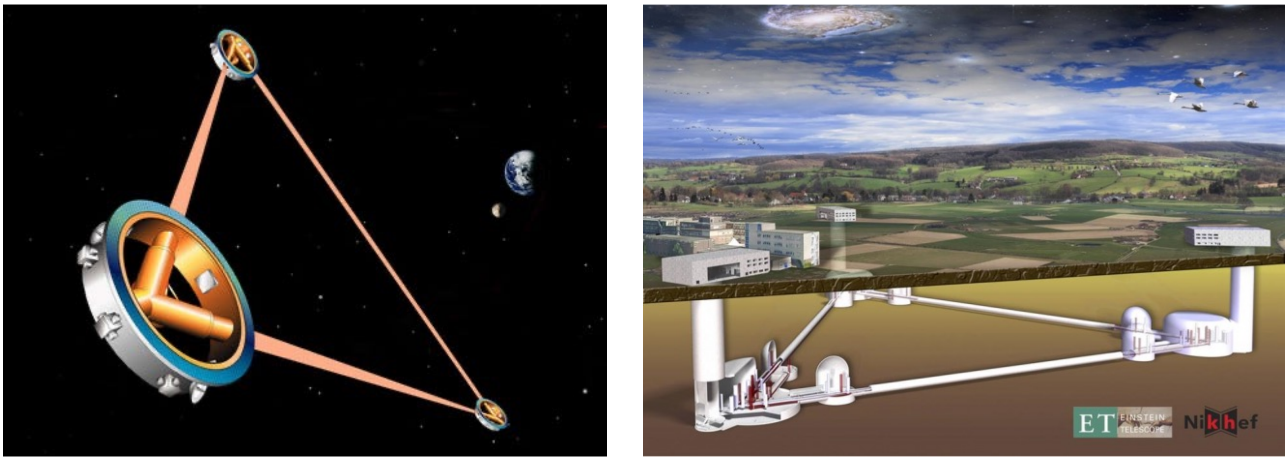


Figure 2-4: (Left) LISA orbiting around the Sun. (Right) Einstein Telescope underground. Artists impressions.

of third-generation detectors such as Einstein Telescope [15] and Cosmic Explorer [16] (see Fig. 2-4 for a visualization). In Fig. 2-5 we show the location of the second-generation of GW interferometers.

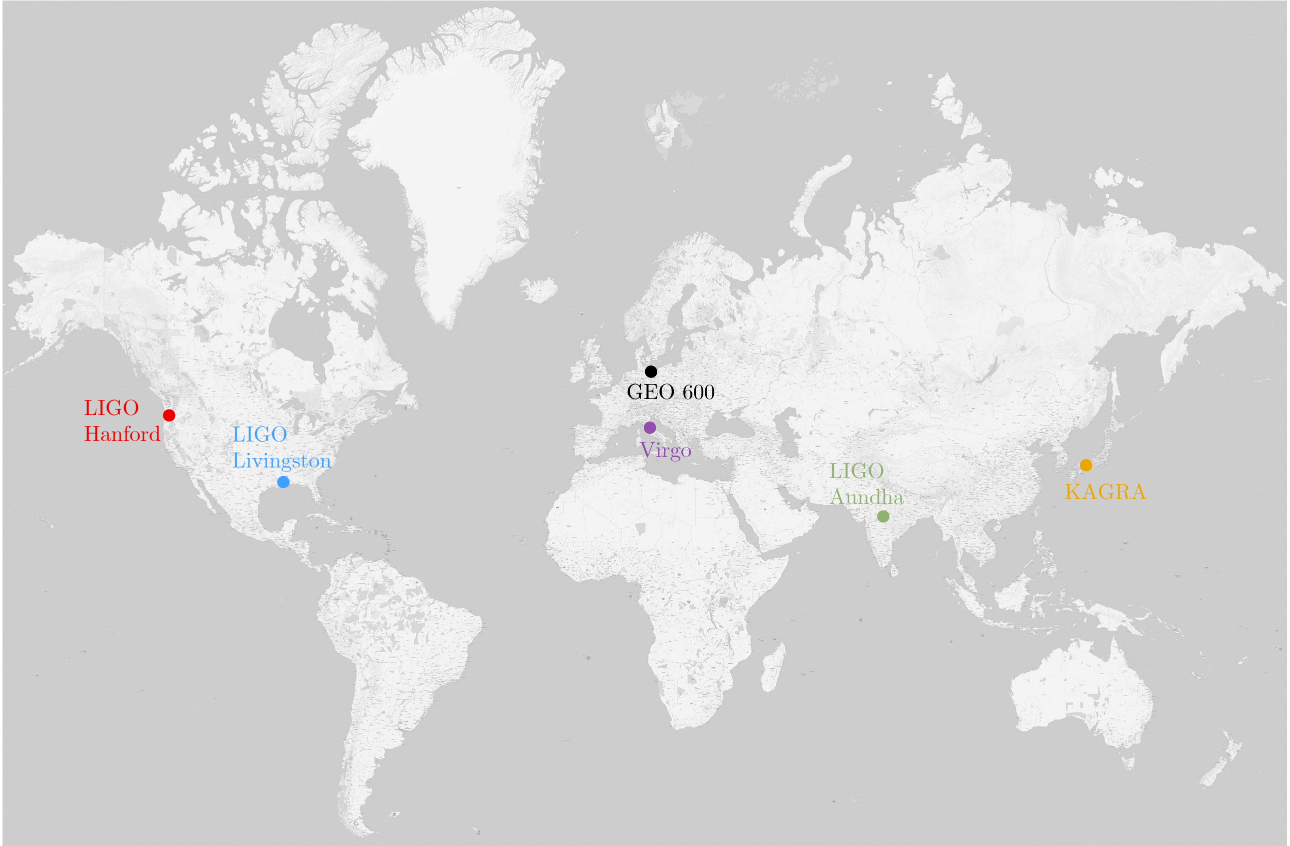


Figure 2-5: Location of the current network of second-generation interferometric detectors.

2.3 Antenna pattern of interferometers

Interferometric detectors are omnidirectional antennas and have a good sensitivity over a large fraction of the sky. From Eq. 2.4, the output of the detector will be $h(t)$, which will respond to a passing GW as $h_{ij}(t, \mathbf{x})$. The general transfer function for GW detectors is

$$h(t, \mathbf{x}) = D^{ij} h_{ij}(t, \mathbf{x}), \tag{2.5}$$

where D_{ij} is the detector tensor that depends on its geometry. For a detector which is sensitive only to GW with a reduced wavelength $\lambdabar$ much larger than its size, we can neglect the spatial dependence of the GW signal $h_{ij}(t, \mathbf{x})$, such that

$$h_{ij}(t) = \sum_{A=+,\times} e_{ij}^{A}(\hat{\mathbf{n}}) h_A(t). \tag{2.6}$$

The direction of propagation of the wave is $\hat{\mathbf{n}}$, and e_{ij}^{A} the polarization tensor defined as

$$e_{ij}^{A}(\hat{\mathbf{n}}) = \begin{cases} \hat{\mathbf{u}}_i \hat{\mathbf{u}}_j - \hat{\mathbf{v}}_i \hat{\mathbf{v}}_j & \text{for } A = + \\ \hat{\mathbf{u}}_i \hat{\mathbf{v}}_j + \hat{\mathbf{v}}_i \hat{\mathbf{u}}_j & \text{for } A = \times \end{cases} \tag{2.7}$$

where $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ are unit vectors orthogonal to $\hat{\mathbf{n}}$. Thus, Eq. 2.5 can be expressed as,

$$h(t) = \sum_{A=+,\times} D^{ij} e_{ij}^{A}(\hat{\mathbf{n}}) h_A(t) = \sum_{A=+,\times} F_A(\hat{\mathbf{n}}) h_A(t), \quad \text{where } F_A(\hat{\mathbf{n}}) = D^{ij} e_{ij}^{A}(\hat{\mathbf{n}}). \tag{2.8}$$

We have conveniently defined $F_A(\hat{\mathbf{n}})$ as the detector pattern functions, which depend on the direction of propagation of the wave $\hat{\mathbf{n}} = (\theta, \phi)$. Hence, the output of the detector yields

$$\begin{aligned} h(t) = h_+(t) F_+(\theta, \phi, \psi) + h_\times(t) F_\times(\theta, \phi, \psi), &\qquad \text{where} \\ F_+(\theta, \phi, \psi) &= \frac{1}{2}(1 + \cos^2\theta) \cos 2\phi \cos 2\psi - \cos\theta \sin 2\phi \sin 2\psi, \\ F_\times(\theta, \phi, \psi) &= \frac{1}{2}(1 + \cos^2\theta) \cos 2\phi \sin 2\psi - \cos\theta \sin 2\phi \cos 2\psi, \end{aligned} \tag{2.9}$$

where ψ is the so-called polarization angle.

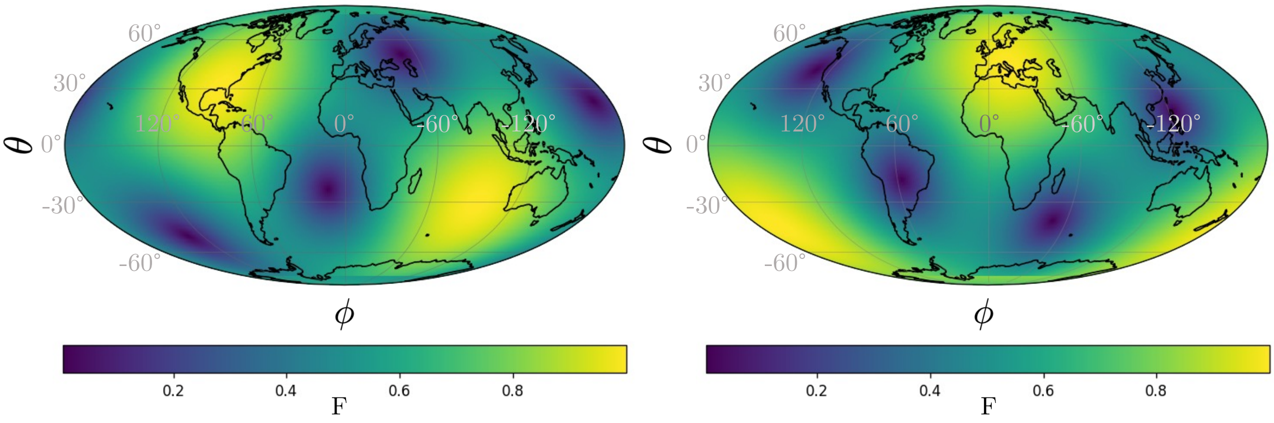


FIGURE 2-6: Variation of $F = \sqrt{F_+^2 + F_\times^2}$ for $\psi = 0$ as a function of the longitude ϕ and the latitude θ for LIGO Livingston (left) and Virgo (right).

In Fig. 2-6 we represent $F = \sqrt{F_+^2 + F_\times^2}$ as a function of the longitude ϕ and the latitude θ for LIGO Livingston in Florida (USA) and Virgo in Pisa (Italy). Large values of F imply a better sensitivity which is dependent on the orientation of the detectors. With respect to the plane defined by the arms L_X and L_Y, the most sensitive directions are orthonormal, and less sensitive directions are bisectors, such that $\cos\theta = \cos 2\phi = 0$.

2.4 Sources of noise

In the previous Section, we have seen the expression of the detector response, Eq. 2.5, in an ideal setting where there are no external perturbances. Nonetheless, the real world is full of imperfections causing undesired noise in the detector strain. Mathematically, the main strain of the detector measures

$$s(t) = n(t) + h(t), \tag{2.10}$$

where $n(t)$ represents the combination of all noise sources. Many textbooks treat the $n(t)$ as Gaussian and stationary, which is a poor approximation to the interferometers' data. The understanding and unbiased modelling of the different sources of noise is fundamental to infer the significance of GW signals and their astrophysical properties. Tasks to understand and mitigate noise sources both from the instrument and the data analysis side, also known as *detector characterization* tasks, are a significant portion of LIGO-Virgo-KAGRA collaboration's work [37, 38, 39, 40]. Following detector characterization guidelines, we can classify noise sources by dividing them into three different categories:

- *Fundamental noises:* They cannot be reduced without a major instrument upgrade, such as the installation of a new laser. An example of fundamental noise is thermal noise , associated with sources of energy dissipation, and quantum noise, related to the quantum nature of photons due to the Heisenberg uncertainty principle and quantum fluctuations.
- *Technical noises:* they arise from electronics or dust in the mirrors, and they can be reduced once identified and carefully studied.
- *Environmental noises:* include seismic-motion, acoustic and magnetic noises [41].

In the following subsections, we will provide details on the sensitivity of the detectors, while a summary of the most dominant sources of noise is provided in Appendix A.1.

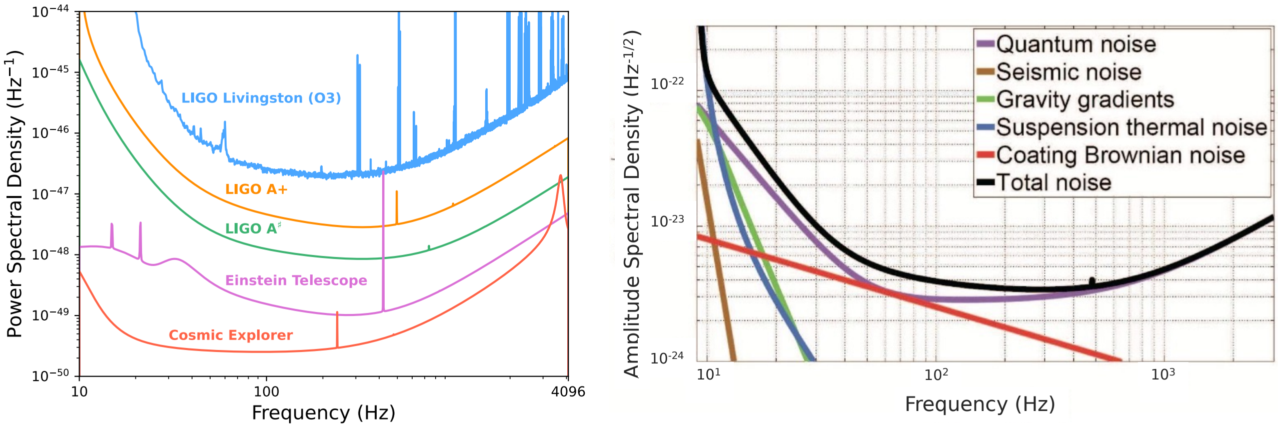


FIGURE 2-7: *(Left) PSDs of LIGO Livingston during the third observing run (O3) (blue), LIGO A+ (orange), LIGO A$^\sharp$ (green), Einstein Telescope (pink) and Cosmic Explorer (red). (Right) Design sensitivity of Advanced LIGO (black), and noise sources from seismic noise (brown), suspension thermal noise (blue), coating thermal noise (red), gravity gradient (green) and quantum noise (purple), retrieved from [42].*

2.4.1 Power spectral density

If the noise is non-stationary, then the different components of the noise are uncorrelated, and therefore the ensemble average of the Fourier components of the noise is of the form

$$\left\langle \tilde{n}^*(f)\tilde{n}(f')\right\rangle = \delta(f-f')\frac{1}{2}S_n(f), \tag{2.11}$$

where $\tilde{\cdot}$ denotes the Fourier transform, * the complex conjugate and, as in the previous chapter, $\left\langle \dots \right\rangle$ represents the average. As we assume $n(t)$ to be dimensionless, $S_n(f)$ has dimensions Hz^{-1}. We can also assume without loss of generality that $\left\langle n(t)\right\rangle = 0$. In Eq. 2.11 the right hand side diverges for $f = f'$, but in any experiment we measure $\tilde{n}(f)$ in a finite time interval T, so for $f = f'$ Eq. 2.11 takes the form:

$$\left\langle |\tilde{n}|^2\right\rangle = \frac{1}{2}S_n(f)T \;\rightarrow\; \frac{1}{2}S_n(f) = \left\langle |\tilde{n}|^2\right\rangle \Delta f \qquad \text{for } f = f', \text{ and } \Delta f = \frac{1}{T} \tag{2.12}$$

where Δf is the resolution of the measurement in frequency. Note that the factor 1/2 is conventionally inserted, such that $S_n(f)$ is obtained integrating over physical frequencies $f > 0$:

$$\left\langle |\tilde{n}|^2\right\rangle = \int_{-\infty}^{\infty} df\, df' \left\langle n^*(f)n(f')\right\rangle = \frac{1}{2}\int_{-\infty}^{\infty} df\, S_n(f) = \int_{0}^{\infty} df\, S_n(f) \tag{2.13}$$

The function $S_n(f)$ is known as the power spectral density (PSD). Moreover, the noise of the detector can also be characterized by $\sqrt{S_n(f)}$, known as amplitude spectral density ASD with dimensions $\mathrm{Hz}^{-1/2}$. In Fig. 2-7 left panel we present the PSD of various detectors PSD. While we show the average PSD of LIGO Livingston during the third observing run (O3), we also show the design sensitivity of LIGO A+ and LIGO $\mathrm{A}^\sharp$, which will be beyond the fifth observing run. Furthermore, we present the PSD of Einstein Telescope (Europe-based) and Cosmic Explorer (USA-based), which will collaborate to detect GW signals during the third-generation era. In Fig. 2-7 right panel we present the design ASD of Advanced LIGO, as well as its noise budget with different sources of noise, namely, seismic noise (brown); suspension thermal noise (blue), which is dominant at $f \lesssim 10\,\mathrm{Hz}$; coating thermal noise (red); gravity gradient (green); and quantum noise (purple), dominant at $f \gtrsim 10\,\mathrm{Hz}$ [42]. The details of these sources are elaborated in Appendix A.1, except for coating thermal noise, which interested readers can refer to [43].

Characterizing the noise of the detector with the PSD is fundamental for GW data analysis, as it provides key information about the frequency content of the data at hand. Nonetheless, the PSD is not known a priori and it needs to be properly estimated. There are two commonly used methods to compute these estimates: "off-source" and "on-source" estimate [44]. The "off-source" estimation assumes that the PSD does not vary over the duration being averaged and that there are no non-Gaussian features in the data. The "on-source" estimation, in contrast, uses the commonly adopted method of simultaneously fitting the signal and power spectral density of the detector noise. While the second method is less affected by noise artifacts, it is computationally more intensive.

The "off-source" estimation is usually preferred for GW searches. The simplest estimate is Welch's method (see Appendix A.3 for its mathematical formalism), which assumes Gaussian and stationary data. To overcome its limitations, other authors have proposed more sophisticated methods for the PSD estimation [45, 46, 47].

2.4.2 Noise lines

GW interferometers are complex experiments with many sub-systems that couple to the main detector strain $h(t)$, causing large narrow-band contributions to the PSD. As an example, we can see in Fig. 2-7 in blue the PSD of LIGO Livingston during O3 presenting several narrow-band peaks, known as *noise lines*. Most lines in the detector data are stationary, but some of them have time-varying behaviour, degrading the detector sensitivity over a larger frequency.

Such behaviour hampers continuous GW searches as these artifacts can lead to spurious outliers which require laborious follow-up.
Some noise lines occur in a "comb" pattern, and the frequency peak of their nth "teeth" follows

$$\omega_N = f_o + n\delta f, \tag{2.14}$$

where f_o is the offset and δf the spacing between the teeth. Combs are associated with linear or non-linear couplings of non-sinusoidal sources, or with non-linear couplings of sinusoidal sources. Lines and combs can have time-dependent behaviours as the configuration of the detector changes: as interferometers undergo enhancements and upgrades, and issues related to coupling with various sub-systems are resolved, but the addition of new hardware, while inevitable, introduces further challenges related to data quality [39].

To further understand the noise of the detector, its status is continuously monitored through a large set of data streams at various sampling rates, outputting $\sim 10^6$ time series from instrumental and environmental sensors. These auxiliary channels can be divided into safe (insensitive to GW) and unsafe (sensitive to GW). Some subset of these channels may serve as "witnesses" to narrow-band couplings in the detector, but they can also "witness" the production of non-gaussian transient burst noise, as we will see in the next subsection.

An example of identified narrow-band couplings is the power lines caused at 50 Hz in Virgo and 60 Hz in LIGO, as well as their respective harmonics, caused by the main power supplies. Other examples are also mechanical resonances of mirror suspension, known as "violin modes", and simulated GW signals known as "hardware injections" [48, 49].
The standard process for mitigating lines or combs is often iterative and experimental, but their main steps are:

1. Identification of noise in the main detector strain.
2. Determination of the properties of the noise, such as duration, associated frequencies and possible channel "witnesses".
3. On-site investigations or interventions.

Work on site is constrained by time availability and the risk of creating novel sources of noise. Hence, the mitigation of noise sources is usually prioritized by their strength, the number of frequency bins contaminated, and the ease of addressing their cause. Lines which are not well-understood are catalogued afterwards, helping GW searches on cleaning the data and rejecting outliers. Mitigation efforts are challenging as they can take order of days or weeks to determine if these methods have contributed significantly to data quality. Furthermore, configuration changes in the detector that lead to line generation can also take time to appear and be mitigated. We recommend that interested readers refer to [50] and references therein for an in-depth description of the methodologies developed to address these issues.

2.4.3 Glitches

Noise couplings can also cause a transient non-astrophysical burst of non-Gaussian noise, which are colloquially known as *glitches* [51, 52]. Glitches may be caused by the environment (e.g., earthquakes, wind, anthropogenic noise) or couplings with instruments (e.g., control systems, electronic components [53]), though in many cases their causes remain unknown [54]. They come in a large variety of time-frequency morphologies, have a typical duration of between sub-seconds and seconds, and have a high rate of occurrence (~ 1 per minute during the first half of the third observing run, O3a [12]).

Glitches are problematic due to their large abundance and capability of hampering GW data analysis. They can reduce the amount of analyzable data increase the noise floor, produce false positives in GW data, affect the estimation of the detector power spectral density and reduce candidate significance in searches for short- and long-lived GW signals [55, 56, 57, 27, 58]. Glitches can also bias astrophysical parameter estimation, making it difficult to determine which part of the signal corresponds to a glitch and which part to the actual GW event [59, 60, 61]. Additionally, glitches can impact line-cleaning procedures in GW searches, which rely on replacing disturbed frequency bins with artificially generated data, consistent with their neighbors [62, 50, 58]. If the surrounding data contains elevated noise floors, the efficacy of mitigation methods will be reduced.

Glitch identification and characterization is a crucial first step towards their mitigation, but due to their overwhelming amount, their characterization by hand is unfeasible. [63, 39, 64]. A promising option is then to construct machine learning (ML) algorithms for their identification. Most of the current approaches to glitch characterization with ML utilize supervised classification algorithms, where models learn to identify glitches through labelled data representations of GW strain data $h(t)$ [65, 66, 67, 68, 69, 70, 71]. In practice, glitches are visualized in time-frequency representations, which involves a modification of the standard short-time Fourier transform parameterized by a quality factor Q [72, 73, 74]. For a discussion on time-frequency representations, the interested reader can refer to Section 3.3.2.

Nonetheless, this procedure of glicht identification presents several limitations. Firstly, generating labelled data is an expensive task, since ML methods need a lot of examples for training, and experts must vet the labelling procedure. Secondly, glitch classes are highly unbalanced, biasing the models towards the most common classes. Moreover, supervised learning needs fixed class definitions that are not exhaustive nor representative of all glitch morphologies, as there could be many possible sub-classes to discover [67]. Furthermore, as GW detectors are improved, novel glitch morphologies could arise [75].

Despite these challenges, these methods have been instrumental in GW detector characterization and data analysis. In the following, we describe the different classes defined by one of the most well-known supervised ML algorithms, `GravitySpy` [65, 68], whose morphologies can be visualized in Fig. 2-8.

- *1080Lines:* these glitches manifest as brief, recurrent spikes occurring $\sim 0.1\,\mathrm{s}$ at $\sim$ 1080 Hz, and additionally accompanied by noise $< 64\,\mathrm{Hz}$. These disturbances were notably widespread in LIGO Hanford during the early stages of the second observing run (O2) but saw a reduction after that by improvements in the output mode cleaner [68].

- *1400Ripples:* they exhibit a short timespan ($\leq 0.5\,\mathrm{s}$) with a wavy morphology at $\sim$ 1400 Hz.

- *Air_Compressor:* they appears as a broad, horizontal line at $\sim 50\,\mathrm{Hz}$. Investigations in LIGO Hanford concluded that these glitches were linked to air compressor motors at the end stations. They were mitigated by replacing the vibration isolators [68].

- *Blip:* these glitches have a characteristic morphology of a symmetric “teardrop” shape in time-frequency in the range $[30, 250]$ Hz with short-durations, $\sim 0.04\,\mathrm{s}$. They appear in both LIGO Livingston and LIGO Hanford, as well as Virgo and GEO 600 [54]. Due to their abundance and form, these glitches hinder both the unmodeled burst and modelled CBC can searches, with particular emphasis on compact binaries with large total mass, highly asymmetric component masses, and spins anti-aligned with the orbital angular momentum [55, 52]. Moreover, since there is no clear correlation to the auxiliary channels, they cannot be removed from astrophysical searches yet.

- *Blip_Low_Frequency:* they have a similar shape to *Blips*, but occur at lower frequencies, with peak $\sim 10-50\,\mathrm{Hz}$. As they fall in the expected frequency band for high-mass CBC, they hinder their detection. Note that this class was added during O3 [75].

- *Chirp:* they are GW signals from CBC artificially added in the detector data via created by hardware injections, i.e. by physically displacing the detectors' test masses. Note that these signals do not accurately reflect our present understanding of CBC populations [48].

- *Extremely_Loud:* their main characteristic is an exceptionally high signal-to-noise-ratio[1] (SNR), saturating their time-frequency representation. Typically, loud glitches result from significant disruptions in the detector, adversely affecting its sensitivity [39].

- *Fast_Scattering:* they appear as short-duration arches ($\sim 0.2-0.3\,\mathrm{s}$) in the frequency range $[20-60]\,\mathrm{Hz}$. These glitches are strongly correlated with ground motion in range $[0.1-0.3]\,\mathrm{Hz}$ and $[1-6]\,\mathrm{Hz}$, which in turn is associated with thunderstorms and human activity near the detector. Note that this class was added during O3, and it is more abundant in LIGO Livingston than LIGO Hanford, due to differences in ground motion and detector sensitivity [75].

- *Helix:* these glitches are usually grouped in sets of $2-3$ separated by $\sim 0.1\,\mathrm{s}$ in the frequency band $16-512\,\mathrm{Hz}$. Investigations point out that they might be related to the auxiliary lasers used to calibrate the detectors [68].

- *Koi_Fish:* Their naming comes from their imaginative frontal resemblance to koi fishes. They are also similar to *Blips* but typically feature high-SNR, spanning the frequency range of $\sim 20-1000\,\mathrm{Hz}$.

- *Light_Modulation:* These glitches are in the frequency range $16-128\,\mathrm{Hz}$, often displaying broad-band spikes. They exhibit high-SNR and stem from fluctuations in the amplitude of the control signal for the optical sidebands, which are responsible for adjusting the length and alignment of optical cavities [68].

- *Low_Frequency_Burst:* they are short-duration glitches ($\sim 0.25\,\mathrm{s}$) in the frequency range $[10-20]$ Hz with a distinctive blob shape. These occurrences were prevalent in LIGO Livingston data during O1 and LIGO Hanford data in O3a.

- *Low_Frequency_Lines:* these glitches have a flat-line-like morphology with durations $\sim 1.5-2\,\mathrm{s}$ and usually at frequencies $< 20\,\mathrm{Hz}$.

- *Power_Line:* They are narrow lines, typically lasting $\sim 0.2-0.5\,\mathrm{s}$ near the frequency of the power grid in the United States (or harmonics of this frequency). These glitches can be attributed to various equipment dependent on this power supply.

- *Repeating_Blips:* these glitches consist on multiple *Blip* glitches, often repeating every $\sim 0.25-0.50$ s.

- *Scattered_Light:* also known as Slow Scattering, these glitches have longer duration harmonics ($\sim 2.0-4.0\,\mathrm{s}$), and in the time-frequency domain, they appear as arches often stacked on top of each other. These glitches are quite problematic since their frequency content lies in the band of interest of GW astrophysical events. In O3, they were found to be coupled with the relative motion between the optical suspension system's end test-mass chain and the reaction-mass chain [75].

---

[1] It must be noted that this definition of SNR is different to the one presented in Section 3.2.1, as it is defined in the context of Omicron [73].

- *Scratchy:* These glitches manifest as a sequence of distinct sharp peaks, primarily at intermediate frequencies in range $\sim 60 - 250\,\mathrm{Hz}$. These peaks can occur at a rate of $\sim 10 - 30\,\mathrm{s}^{-1}$, and are associated with light scattering from the Swiss cheese baffles [68].
- *Tomte:* these glitches are also short-duration ($\sim 0.25\,\mathrm{s}$) with characteristic triangular morphology. Since there is no clear correlation to the auxiliary channels, they cannot be removed from astrophysical searches.
- *Wandering_Line:* They are long-duration glitches with a distinctive undulating morphology. They can span a broad spectrum of frequencies, often displaying multiple lines simultaneously at various frequencies, but typically they spam in frequencies $> 256\,\mathrm{Hz}$.
- *Whistle:* These glitches have a characteristic V, U or W shape at higher frequencies ($\gtrsim$ 128 Hz) with typical durations $\sim 0.25\,\mathrm{s}$. They are caused when radio-frequency signals beat with the voltage-controlled oscillators [37].

As we mentioned before, persistent glitches at specific frequencies critical to data analysis constitute a problem for GW detection and parameter estimation. Hence, it is essential to eliminate the harmful influence of these glitches on the searches for GW signals. Discerning the background of glitches is indeed a challenging problem. One major difficulty is the lack of glitch simulations, making it challenging to transition from perfect simulated noise to real data. Additionally, the unknown real population of glitches complicates the evaluation and understanding of the performance of ML classifiers. Moreover, while glitches are observed in the strain data $h(t)$ they are produced in the different subsystems of the interferometers, so understanding their formation is challenging by solely utilizing $h(t)$ data.

To tackle some of these challenges, in Chapter 6, we simulate *Blip* glitches—one of the main classes of glitches that hinder transient GW searches—using ML techniques. Additionally, we propose several applications and provide a practical example to evaluate the performance of `Gravity Spy`. In Chapter 6, we found that our ML algorithm could learn anomalous morphologies due to the lack of ground truth in the real *Blip* population. To address this fundamental issue, in Chapter 7, we construct an unsupervised ML method to explore the auxiliary channel data—time series originating from the monitors of different subsystems—and learn the underlying distribution of the data, uncovering unknown glitch morphologies and overlaps.

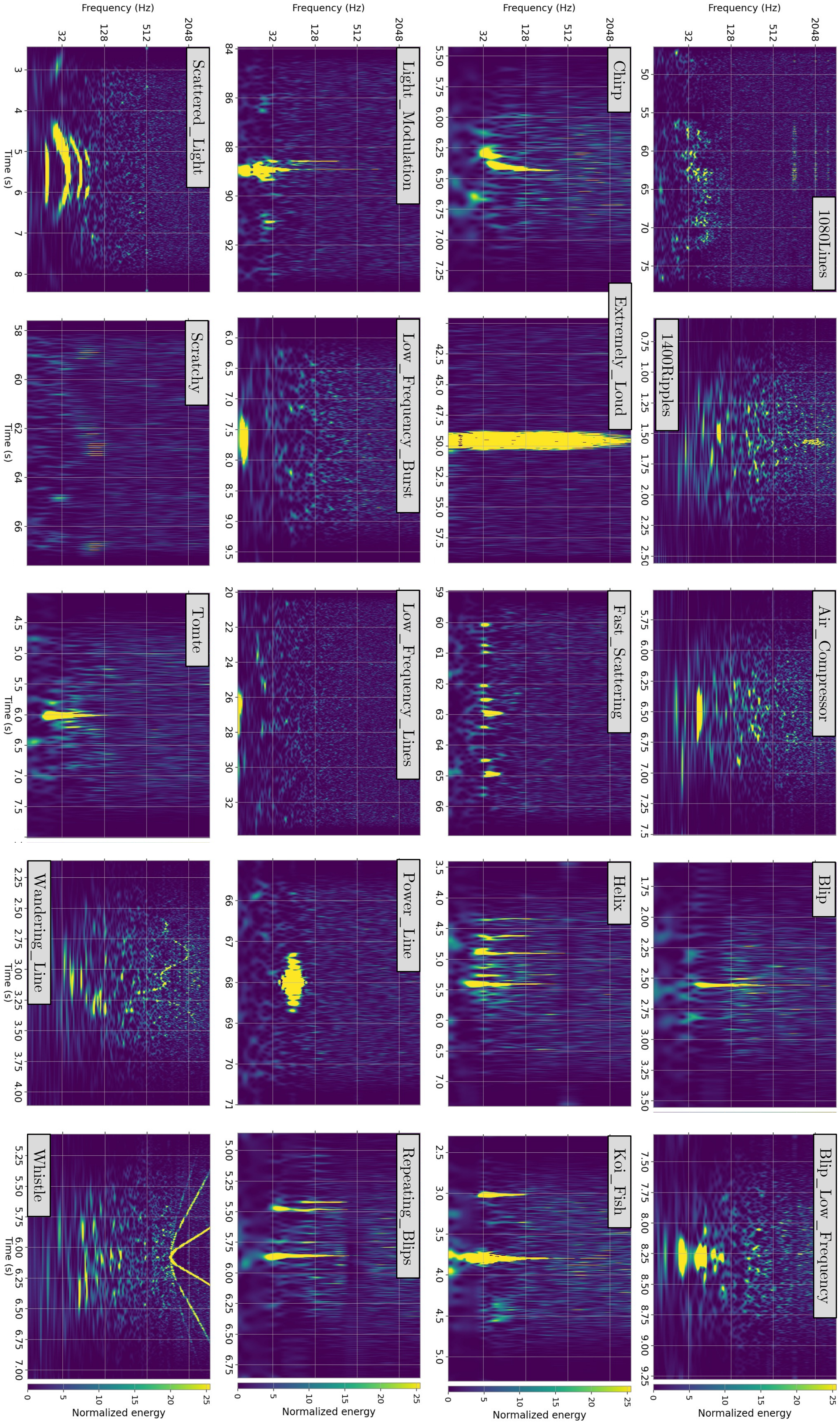


Figure 2-8: *Time-frequency representations of different glitch classes in LIGO during O3.*

2.5 Data conditioning

From the previous Sections, we have learned that the main strain of the detector $h(t)$ is impacted by fundamental noises, that can only be reduced with a major instrument upgrade, technical noises, that arise from the different subsystems within the detector, and environmental noises. As GW have minuscule amplitude and are buried in the detector noise, it is crucial to develop robust data analysis techniques that enhance these signals. In this section, we provide details about the most basic data conditioning methods that have led to GW discovery.

2.5.1 Sampling of continuous-time signals

The signal $h(t)$ is continuous, but the data measured in the detector is a representation of this sampled signal. Typically, we can obtain the discrete-time representation $x_s(t) = x[n]$ of a continuous-time signal x_c through periodic sampling as

$$x[n] = x_c(nT), \quad -\infty < n < \infty \tag{2.15}$$

where T is the sampling period, or time resolution, and $f_s = 1/T$ is the sampling frequency. It is important to note that the sampling operation is not invertible in general since many continuous-time signals can produce the same output sequence of samples. This inherent ambiguity is a fundamental issue in signal processing, but it is possible to restrict it by controlling and narrowing down the range of input signals fed into the sampling system [76]. Now, we derive the frequency domain relation between $x_c(t)$ and $x_s(t)$, so we consider that their relation is modulated by a periodic impulse train $s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$ such that

$$x_s(t) = x_c(t)s(t) = x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) = \sum_{n=-\infty}^{\infty} x_c(nT)\delta(t - nT) \tag{2.16}$$

Then, its Fourier transform, denoted as $\tilde{\ }$, will be

$$\tilde{x}_s(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \tilde{x}_c(\omega - k\omega_s), \quad \text{where } \omega_s = \frac{2\pi}{T} \tag{2.17}$$

Here, ω is the angular frequency, ω_s is the sampling angular frequency, which are continuous variables. Also, k is an integer number. From Eq. 2.17 we can observe that $x_s(t)$ consists of periodically repeated copies of $\tilde{x}_c$, which are shifted by integer multiples of ω_s and then superimposed to produce the periodic Fourier transform. According to the Nyquist sampling theorem, if $x_c(t)$ is band-limited, it is uniquely determined by

$$x[n] = x_c(nT) \quad \text{with } n \in \mathbb{Z} \quad \text{if } \omega_s = \frac{2\pi}{T} \geq \omega_N \tag{2.18}$$

where ω_N is known as Nyquist frequency. In this way the replicas do not overlap $\omega_s > 2\omega_N$ [77]. Otherwise, the frequency would components overlap, resulting in *aliasing*.

We express the discrete frequency $\Omega = \omega/T$ in radians/sample and, therefore, dimensionless. Hence, we can define the discrete Fourier transform of the sequence $x[n]$, often expressed in signal processing as $\tilde{x}(e^{i\Omega})$, is

$$\tilde{x}(e^{i\Omega}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \tilde{x}_c(\omega - k\omega_s), \tag{2.19}$$

where we have used Eq. 2.17 in the last term. Equivalently,

$$\tilde{x}(e^{i\Omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} \tilde{x}_c(\frac{\Omega}{T} - \frac{2\pi}{T}k). \tag{2.20}$$

2.5.2 Low-pass, high-pass and band-pass filters

A way to avoid aliasing is to restrict the frequencies of the signal such that $f_s > 2\omega_N$. For this, we can use a low-pass filter which allows frequencies below a given cut-off frequency. Another possibility is to employ a band-pass filter, which allows frequencies in a pre-defined range. In practice, a band-pass filter can be constructed with a low-pass filter and a filter which allows frequencies below a cut-off frequency, known as a high-pass filter. For illustration, in Fig. 2-9 we provide an example of these filters.

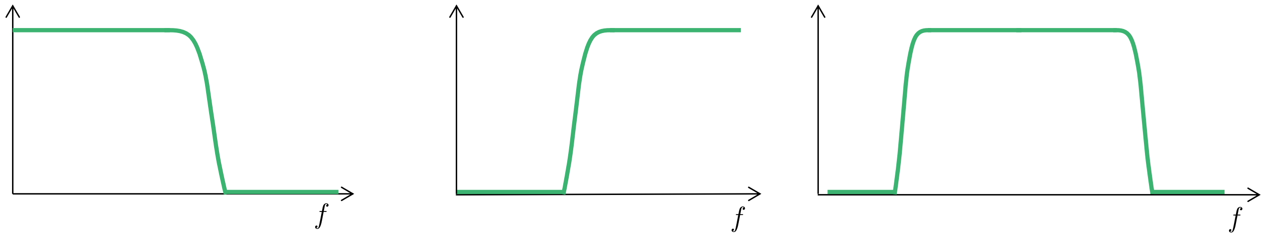


FIGURE 2-9: Schematic low-pass (left), high-pass (middle) and band-pass filters (right).

These filters are relevant in GW data analysis as the output of the detector spans from 10 Hz since the detector is not calibrated at lower frequencies, to $\sim$ 10 kHz. Limiting the frequency range of our analysis does not only avoid aliasing (see discussion in the previous section), but might simplify the data, which can lead to more accurate and reliable results. Depending on the nature of the target source, we would be inclined to perform a narrow band search, as in the cause of continuous waves with quasi-monochromatic signals (see Section 2.1.3), or even a wide range search, as in the case of transient Burst searches (see Section 2.1.2).

2.5.3 Resampling

The main strain of the detector $h(t)$ is a time series sampled at 16384 Hz for LIGO and 20 kHz for Virgo, but these data are only calibrated $>$ 10 Hz for the A+ detectors. Depending on the frequency content of the targeted signal, we might want to reduce the sampling rate of the data by defining a new time series x_d such that

$$x_d[n] = x[nM] = x_c(nMT), \tag{2.21}$$

where M is a reduction factor and T is the sampling period. $x_d[n]$ is identical to the time series that would be obtained from the continuous signal $x_c(t)$ sampled with a period $T' = MT$. The sampling rate can be reduced by a factor M without aliasing if the new time series is band-limited to $\omega < \omega_N$, and then $x_d[n]$ would be an exact representation of $x_c(t)$ if $\pi/T' = \pi/MT \geq \omega_N$. This operation is called *downsampling.*

Similarly to Eq. 2.20, the discrete-time Fourier transform of $x_d[n] = x[nM] = x_c(nT')$ is

$$\tilde{x}_d(e^{i\omega}) = \frac{1}{T'}\sum_{r=-\infty}^{\infty} \tilde{x}_c\left(\frac{\omega}{T'} - \frac{2\pi r}{T'}\right) = \frac{1}{MT}\sum_{r=-\infty}^{\infty} \tilde{x}_c\left(\frac{\omega}{MT} - \frac{2\pi r}{MT}\right). \tag{2.22}$$

Relating Eq. 2.20 and Eq. 2.22, we can express the summation index r from Eq. 2.22 as $r = m + kM$ for $k \in (-\infty, \infty)$ and $m \in [0, M-1]$. Thus, we can rewrite Eq. 2.22 as

$$\tilde{x}_d(e^{i\omega}) = \frac{1}{M}\sum_{m=0}^{M-1}\left[\frac{1}{T}\sum_{k=-\infty}^{\infty}\tilde{x}_c\left(\frac{\omega}{MT} - \frac{2\pi k}{MT} - \frac{2\pi m}{MT}\right)\right] = \frac{1}{M}\sum_{m=0}^{M-1}\tilde{x}e^{i(\omega/M-2\pi m/M)}. \tag{2.23}$$

In this way we can interpret $\tilde{x}_d(e^{i\omega})$ as composed of M copies of the periodic Fourier transform $\tilde{x}(e^{i\omega})$, frequency scaled by M and shifted by integer multiples of 2π. From this we can understand that $\tilde{x}_d(e^{i\omega})$ is periodic with period 2π and that aliasing can be avoided by ensuring that $\tilde{x}(e^{i\omega})$ is band-limited. Therefore, it is important to apply a low-pass or anti-aliasing filter before downsampling. Such procedure is known as *decimation*.

For completeness, but not utilized in this work, we could be interested in increasing the sampling rate by a factor of L, such that the new sequence will be defined as

$$x_e[n] = x[n/L] = x_c(nT/L), \qquad \text{for } n = 0, \pm L, \pm 2L, \ldots \tag{2.24}$$

This operation is known as *upsampling*, and it can be mathematically expressed as

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL], \tag{2.25}$$

where its Fourier transform

$$\tilde{x}_e(e^{i\omega}) = \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]\right)e^{-i\omega n} = \sum_{k=-\infty}^{\infty} x[k]e^{-i\omega Lk} = X(e^{i\omega L}), \tag{2.26}$$

such that the Fourier transform of the upsampled output is a frequency-scaled version of the Fourier transform of the input. To avoid artifacts, the resulting upsampled sequence needs to be low-passed.

2.5.4 Whitening

As we have seen in Section 2.4 the noise from the detector is composed of couplings of different sub-systems, as well as its surrounding environment. In GW data analysis it is key to estimate the PSD of the data to understand the sources of noise. As we can see from the top panel of Fig. 2-7, where we present the PSDs of current and future detectors, and 2-10, where we present the raw coloured time-series data, the noise of the detector is dominated by low and high frequencies. GW signals are buried within coloured detector noise, so a common practice in GW data analysis is to make the data delta-correlated or Gaussian-like with uniform variance by removing all the correlation of the noise. In practice, the resulting PSD has equal amplitude fluctuations at all frequencies, allowing for easy comparisons. The process of transforming coloured noise to Gaussian-like noise is known as *whitening* [78], and mathematically

$$\tilde{d_w}(f) = \tilde{d}(f)/S_n^{-1/2}(f) \tag{2.27}$$

where $\tilde{d}$ and $\tilde{d_w}(f)$ are the Fourier transform of the coloured data and whitened data, respectively. To obtain the time series of the whitened data we can simply apply the inverse Fourier transform in $\tilde{d_w}(f)$. It is important to note that during this process artifacts due to aliasing can occur, so the whitened data should be cropped to avoid biasing in the subsequent analysis. In the second and fourth panels of Fig. 2-10 we show the whitened time-series data and the PSD, respectively. We can see by eye the presence of GW150914, but to highlight it we band-pass between $30-250$ Hz, as we can observe in the third panel, and its respective PSD in the fourth panel of Fig. 2-10.

2.6 Data quality and vetoes

To enhance the sensitivity of GW searches and reduce the number of false alarms, multiple types of data quality products are used to indicate the state of the detector during the data analysis. One such product is *data quality flags*, which indicates the suitability of collected data for analysis [39, 40].

Category 1: These flags denote that the detector noise has been severely impacted, rendering it unsuitable for astrophysical analysis. They could indicate significant changes in the properties of the noise of the detector or incorrect calibration, which might arise from incorrect detector settings or on-site maintenance work, among others. In O2 and O3 data flagged in this category represented $< 2.0\%$ of the total data.

Category 2: These flags indicate periods of time where the data is impacted by excess noise and should be treated with caution, as investigations have demonstrated a firm correlation between auxiliary channels and $h(t)$. Search algorithms are recommended not to consider potential candidates during these times, as they are more likely to have been caused by instrumental or environmental couplings. As Category 2 flags reduce the amount of analyzable data, they can potentially jeopardize the number of detectable gravitational waves (GWs) if the amount of time removed is not minimized. Consequently, these vetoes are generally not utilized in CBC searches unless it has been demonstrated that the flagged data significantly impairs the search. While CBC searches rely on priors of the shape of the GW, unmodelled GW have minimal to no assumptions. Thus, burst searches must add further restrictions to data quality flags, increasing the removed data. On average, over O2 and O3, CBC searches removed approximately $\sim 0.20\%$ of the data using Category 2 flags, while burst searches removed approximately $\sim 0.52\%$.

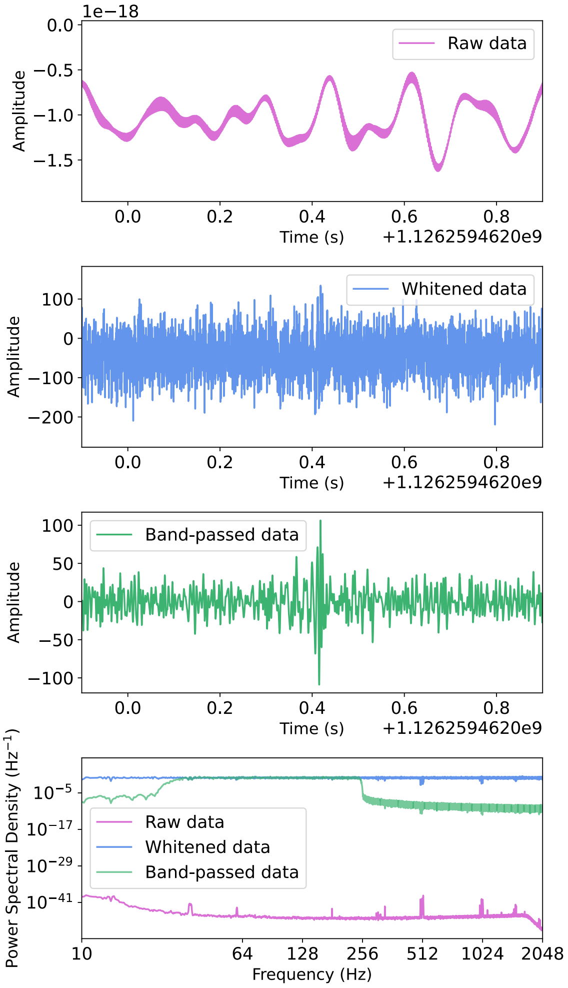


FIGURE 2-10: (First panel) Raw data from the detector $s(t)$, dominated by low and high frequencies. (Second panel) Whitened data, (Third panel) Whitened and band-passed data. (Third panel) PSDs of coloured data (pink), white data (blue), white and band-passed data (green).

Category 3: These flags indicate periods that correlate with auxiliary channels, though the exact nature of these correlations is not yet fully understood. Most of these flags are generated with the Hveto algorithm [79] by correlating safe auxiliary channels, i.e. channels insensitive to GW, with the main strain of the detector $h(t)$. Such flags remove $\sim 10\%$ of the data and are used only by some burst searches.

The data quality flags and their details are available via Gravitational-Wave Open Science Center (GWOSC) [80, 81].

Chapter 3

Transient Gravitational-Wave Searches

In the previous chapter, we described different sources of noise that affect GW searches, as well as the different efforts in data conditioning and data quality studies carried out by LIGO-Virgo-KAGRA collaboration. A key element in the search for faint GW signals is the development of exquisitely sensitive and robust algorithms. In this chapter, we briefly derive the formalism of GW searches (see Section 3.1), and describe the current state-of-the-art methods for short, or transient, modelled (Section 3.2) and unmodelled GW searches (Section 3.3). Last but not least, in Section 3.4 we describe how to construct a generic background to estimate the significance of GW searches.

3.1 The search formalism

GW search algorithms test the presence of GW signal in, for instance, a single GW detector. Thus, given the detector's time series output $s(t)$, the following hypothesis testing is solved:

Null hypothesis (H_0): the time series $s(t)$ does not contain a GW $\rightarrow s(t) = n(t)$

Alternative hypothesis (H_{a_0}): the time series $s(t)$ contains a GW $\rightarrow s(t) = h(t) + n(t)$

As before, $n(t)$ represents the noise of the detector and $h(t)$ the signal of a passing by GW. *A priori*, we do not know if $h(t)$ is present in the time series $s(t)$, so the goal of the search algorithms is to construct a statistic proportional to the probability that $h(t)$ is in $s(t)$, $P(h(t)|s(t))$. For this aim, we can use Bayes' theorem to compute the *a posteriori* probability $P(h|s)$, known as *posterior*, given the output of the detector s, where we drop (t) to simplify the notation,

$$P(h|s) = \frac{P(s|h)P(h)}{P(s)}. \tag{3.1}$$

Here, $P(s|h)$, known as *likelihood*, is the probability of obtaining the time series s if h exists in it, $P(h)$, called *prior*, is the probability of h occurring, and $P(s)$, known as *evidence*, is the probability of getting the time series s, given by

$$P(s) = P(s|0)P(0) + P(s|h)P(h), \tag{3.2}$$

since h either does or does not exist within s. Here, $P(s|0)$ is the probability of obtaining the time series s if no signal is contained, i.e. a particular realization of the noise n, and $P(0)$ is the probability of getting no signal. $P(0)$ and $P(h)$ are known as *priors*, as they represent our *a priori* belief about the existance of the signal h, irrespective of ability of our experiment to detect it. Substituting Eq. 3.2 in Eq. 3.1,

$$\begin{aligned} P(h|s) &= \frac{P(s|h)P(h)}{P(s|0)P(0) + P(s|h)P(h)} \\ &= \frac{P(s|h)}{P(s|0)[P(0)/P(h) + P(s|h)/P(s|0)]} \\ &= \frac{\Lambda}{P(0)/P(h) + \Lambda} \qquad \text{where } \Lambda = \frac{P(s|h)}{P(s|0)} \qquad (\textit{likelihood ratio}). \end{aligned} \tag{3.3}$$

Note that $P(s|h)$ is a monotonically increasing function of Λ, such that a search algorithm will aim to evaluate Λ instead of $P(s|h)$. We further note that $ln(\Lambda)$ also increases monotonically with $P(s|h)$, being commonly evaluated instead of Λ, as it varies less rapidly around the region of interest. Now, within this framework we can redefine our hypothesis testing by constructing a decision rule for the presence or absence of h, setting P_* as the decision threshold.

Null hypothesis (H_0): the time series $s(t)$ does not contain a GW $\rightarrow P(h|s) < P_*$

Alternative hypothesis (H_{a_0}): the time series $s(t)$ contains a GW $\rightarrow P(h|s) \geq P_*$

The details regarding the computation of the likelihood ratio Λ and the decision threshold P_* are dependent on the target source of GW. As we have seen in Section 2.1, we can divide the GW sources according to their duration (short/long), and if astrophysical models are associated with the source (modelled/unmodelled), so search methodologies vary according to the particularity of their targeted source. Once the particular statistics of the modelled or unmodelled algorithms have been obtained and their likelihood ratio has been computed, the posterior assessment of the performance of the algorithm and the significance of their results are common to both analyses.

3.2 Modelled searches

CBC signals, presented in Section 2.1.1, are well-understood in the GW field. Their modelling has been vastly studied employing Post-Newtonian approaches [82] and the effective-one-body formalism [83]. Thus, the essential idea of modelled searches is to use the optimal filter $K(t)$ that, given a time series $s(t) = h(t) + n(t)$, is able "filter out" $n(t)$ via cross-correlation with $s(t)$. As we have seen, such a task is challenging due to the complexity of $n(t)$, but we can simplify the problem by approximating $n(t)$ to be locally Gaussian and stationary.

3.2.1 Matched-filtering

We assume a GW signal $h(t)$ buried in stationary and Gaussian noise $n(t)$ with zero mean, such that $s(t) = h(t) + n(t)$. Given the optimal filter $K(t)$,

$$\hat{s} = \int_{-\infty}^{\infty} K(t)s(t)dt = \int_{-\infty}^{\infty} \tilde{K}(f)^* \tilde{s}(f)df, \quad \text{where } \tilde{s}(f) = \int_{-\infty}^{\infty} s(t)e^{-2\pi i f t}dt \tag{3.4}$$

where $\hat{s}$ is the filtered value of $s(t)$, * represents the complex conjugate, and $\tilde{\cdot}$ the Fourier transform. Then, the detection statistic that is maximised by the optimal filter $K(t)$ will be the SNR, defined as $\rho = S/N$. S is the expected value of $\hat{s}$ when $h(t) \neq 0$, while N is the squared root of the noise variance when $h(t) = 0$, so that

$$S = \int_{-\infty}^{\infty} K(t)\Big\langle s(t)\Big\rangle dt = \int_{-\infty}^{\infty} K(t)\Big\langle h(t)\Big\rangle dt + \int_{-\infty}^{\infty} \cancelto{0}{\Big\langle n(t)\Big\rangle} K(t)dt = \int_{-\infty}^{\infty} \tilde{K}^*(f)\tilde{h}(f), \tag{3.5}$$

given the assumption of $\Big\langle n(t)\Big\rangle = 0$, and

$$\begin{aligned} N^2 &= [\Big\langle \hat{s}^2(t)\Big\rangle - \Big\langle \hat{s}(t)\Big\rangle^2]_{h=0} = \Big\langle \hat{s}^2(t)\Big\rangle_{h_0} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{K}(f)^*\tilde{K}(f')^*\Big\langle \tilde{n}(f)\tilde{n}(f')\Big\rangle df'df \\ &= \int_{-\infty}^{\infty} \frac{1}{2}S_n(f)|\tilde{K}(f)|^2 df, \end{aligned} \tag{3.6}$$

where we have combined Eq. 2.11 and 3.4, and $S_n(f)$ represents the PSD in the frequency domain. The ratio between the expressions above yields the mathematical expression of the SNR for an optimal filter $K(t)$,

$$\rho = \frac{S}{N} = \frac{\int_{-\infty}^{\infty} \tilde{K}^*(f)\tilde{h}(f)df}{\Big[\int_{-\infty}^{\infty} 1/2S_n(f)|\tilde{K}(f)|^2 df\Big]^{1/2}}. \tag{3.7}$$

To identify the optimal filter $K(t)$ normalized by the PSD we first define the scalar product between two real functions $A(t)$ and $B(t)$, by

$$(A|B) = \mathrm{Re}\int_{-\infty}^{\infty} \frac{\tilde{A}^*(f)\tilde{B}(f)}{1/2S_n(f)}df = 4\mathrm{Re}\int_{-\infty}^{\infty} \frac{\tilde{A}^*(f)\tilde{B}(f)}{S_n(f)}df \tag{3.8}$$

where the last term holds since $A(t)$ and $B(t)$ are real functions, and $\tilde{A}(-f) = \tilde{A}^*(f)$. Following the definition of the scalar product, we can define the Wiener scalar product as,

$$(K|h) = \mathrm{Re}\int_{-\infty}^{\infty} \frac{\tilde{K}^*(f)\tilde{h}(f)}{1/2S_n(f)}df = 4\mathrm{Re}\int_{0}^{\infty} \frac{\tilde{K}^*(f)}{\sqrt{S_n(f)}}\frac{\tilde{h}(f)}{\sqrt{S_n(f)}}df = 4\mathrm{Re}\int_{0}^{\infty} \frac{\tilde{K}^*(f)}{S_n(f)}\tilde{h}(f)df, \tag{3.9}$$

where we take the real part Re since $K(t)$ and $h(t)$ are real functions. This scalar product is positive definite since $\tilde{K}(-f) = \tilde{K}^*(f)$ and $S_n(-f) = S_n(f)$ (see Section 2.4.1 for details). The third term can be interpreted as "whitening" both the optimal filter $K(t)$ and target signal $h(t)$, as they are weighted by the inverse ASD [84]. The Wiener scalar product is normalized such that $\Big\langle (n|n)\Big\rangle = 1$, which possesses the property $\Big\langle (a|n)(n|b)\Big\rangle = (a|b)$[85]. Hence, we can define the overlap between the optimal filter $K(t)$ and itself as

$$(K|K) = \Big\langle (K|n)(n|K)\Big\rangle = \int_{-\infty}^{\infty} \frac{1}{2}S_n(f)|\tilde{K}(f)|^2 df. \tag{3.10}$$

Thus, rewriting Eq. 3.7 yields,

$$\rho = \frac{S}{N} = \frac{(K|h)}{\Big\langle (K|n)(K|b)\Big\rangle^{1/2}} = \frac{(K|h)}{(K|K)^{1/2}}. \tag{3.11}$$

The maximum SNR ρ would be a "vector" of unit norm $\hat{n} = K/(K|K)^{1/2}$, such that its scalar product with the "vector" h is maximum. This is obtained choosing $\hat{n}$ parallel to h, so

$$\tilde{K}(f) \propto \frac{\tilde{h}_m(f)}{S_n(f)}, \tag{3.12}$$

where $\tilde{h}_m(f)$ is the Fourier transform of the modelled GW signal, also known as template h_m. Since the filter function is chosen to "match" the signal we are looking for, this technique is known as *matched filtering*. Therefore, we define the optimal SNR as the normalized Wiener scalar product between the template h_m and itself, and the SNR between an unknown time series $s(t)$ and the template h_m respectively as

$$\rho_{\text{opt}} = \frac{(h_m|h_m)}{(h_m|h_m)^{1/2}} = (h_m|h_m)^{1/2}, \qquad \rho = \frac{(h_m|s)}{(h_m|h_m)^{1/2}} = (\hat{h}_m|s), \tag{3.13}$$

where $\hat{h}_m = h_m/\sqrt{(h_m|h_m)}$ is the normalized template. Nonetheless, the parameter space of GW signals from CBC is vast, so to find these signals, we will need a bank of modelled GW.

3.2.2 Modelled compact binary coalesence waveforms

We consider a CBC system composed of two compact spherical objects. The morphology of the resulting GW signal of their coalescence is defined, at least, a set of 15 λ parameters. These parameters are the component masses m_1 and m_2; the component dimensionless spin vectors χ_1 and χ_2 (6 components); the luminosity distance D_L of the source; the sky location of the signal with respect to the frame of the detector (θ, ϕ); the coalescence time t_c of the signal; the inclination of the binary with respect to the line-of-sight to the system ι; the polarization angle ψ; and an orbital phase at coalescence ϕ_c. These parameters can be divided into *extrinsic* parameters λ_{ext}, which depend on the orientation of the binary with respect to an outside observer, and *intrinsic* parameters λ_{int}, which are the physical characteristics of the binary that are independent of the observer's location.

The intrinsic parameters can be used to compose other magnitudes, such as the total mass $M = m_1 + m_2$, or the chirp mass $\mathcal{M}$ and effective spin χ_{eff} defined as,

$$\mathcal{M} = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}, \qquad \chi_{\text{eff}} = \frac{m_1\chi_1 + m_2\chi_2}{m_1 + m_2}. \tag{3.14}$$

As discussed in the previous section, the production of templates is fundamental for modelled GW searches. There are various approaches for producing these templates, known as waveform *approximants*. These approximants can be analytical, numerical, or a combination of both. Different approximants can describe different physical phenomena and are restricted to a specific region of the parameter space. Below, we provide a brief description of each type of approximant. For an in-depth overview, interested readers should refer to [85].

- **Post-Newtonian:** In Section 1.1, we discussed the generation of GW without significant curvature contribution in the near-field. In realistic scenarios, binary systems do affect space-time curvature; Post-Newtonian theory assumes slow, distant orbits for weak self-gravitation, valid only for the inspiral regime [86, 87, 88, 89, 90, 91].

- **Numerical Relativity:** During the merger phase, the binary system experiences intense and highly non-linear gravitational forces, requiring the full Einstein equations to accurately describe the dynamics. Advanced numerical techniques known as Numerical Relativity are used to solve these equations [85]. Due to their complexity and computational cost, other approaches that combine Post-Newtonian theory and Numerical Relativity have been developed to produce the whole frequency regime of CBC GW: inspiral, merger, and ringdown (IMR).

- **Effective One body (EOB):** This approach improves on Post-Newtonian theory by using a resummed, non-polynomial representation of the system's velocity v, allowing for numerical solutions that extend beyond the inspiral phase into the plunge phase. For the ringdown phase, perturbation theory is employed to create a seamless IMR waveform. Although EOB is more accurate than Post-Newtonian theory, it still experiences dephasing during the merger phase, which can be corrected by calibrating EOB with Numerical Relativity simulations, resulting in "EOBNR" [92, 93, 94, 95, 96, 97].

- **Phenom:** Phenomenological or Phenom models, offer an alternative to EOB models by addressing their complexity and high computational cost, using analytical models constructed directly in the Fourier domain with closed-form expressions based on the binary system parameters. The latest iteration are the "PhenomX" models [85, 98, 99, 100, 101], used to generate the waveforms in Fig. 3-1.

To find a CBC GW we would need to maximize the SNR ρ over the λ parameters. However, this can be simplified for aligned spin searches. As discussed in Section 2.3, the observed signal $h(t)$ at the detector is the sum of the two GW polarizations, h_+ and $h_\times$, each multiplied by the detector's response function to the respective polarization F_+ and $F_\times$ (see Eq. 2.9). In the case of aligned spin searches, only the dominant $(l,|m|)=(2,2)$ modes of the waveform are considered, which allows the following simplification [102, 103],

$$\begin{aligned} h_+(t) &= \frac{1+\cos^2\iota}{2D_L} A(t-t_c;\zeta)\cos[2(\Phi(t-t_c;\zeta)+\phi_c)], \\ h_+(t) &= \frac{\cos\iota}{D_L} A(t-t_c;\zeta)\sin[2(\Phi(t-t_c;\zeta)+\phi_c)], \end{aligned} \tag{3.15}$$

where $A(t;\zeta)$ and $\Phi(t;\zeta)$ are functions of time and $\zeta=(m_1,m_2,\chi_1,\chi_2)$, with χ_1 and χ_2 denoting the constant projections of the spins in the direction of the orbital angular momentum. Combining Eq. 3.15 and Eq. 2.9 yields

$$\begin{aligned} &h(t) = \frac{A(t-t_c;\zeta)}{D_\text{eff}}\cos 2(\Phi(t-t_c;\zeta)+\phi_0), \text{ with} \\ &D_\text{eff} = \frac{D_L}{F_+^2\left(\frac{1+\cos^2\iota}{2}\right)^2 + F_\times^2\cos^2\iota}, \qquad e^{2i\phi_0} = e^{2i\phi_c}\frac{F_+(1+\cos^2\iota)/2 - iF_\times\cos\iota}{\left[F_+^2(\frac{1+cos^2\iota}{2})^2 + F_\times^2\cos^2\iota\right]^{1/2}} \end{aligned} \tag{3.16}$$

where D_eff is the effective distance and ϕ_0 is the phase of the observed waveform at coalescence. Therefore, in the aligned-spin case, as $\phi_c, D_L, \theta, \phi, \psi, \iota$ are only present at D_eff and ϕ_0, so this dependence amounts only to an overall phase and overall amplitude. Applying the stationary phase approximation [104], the Fourier transform of $h(t)$ is

$$\tilde{h} = \frac{e^{2i\phi_0}}{D_\text{eff}} A(t-t_c;\zeta)\cos[2(\Phi(t-t_c;\zeta))] = \frac{e^{2i\phi_0}}{D_\text{eff}}\tilde{h}_0(f;t_c,\zeta) = \frac{e^{2i\phi_0}}{D_\text{eff}}\tilde{h}_0(f;\zeta)e^{-2i\pi f t_c}. \tag{3.17}$$

as the coalescent time t_c parametrizes time translations of h_0 [9]. Therefore, in the aligned spin case, it is possible to quickly maximize the SNR ρ over all parameters except for the intrinsic parameters $\zeta=(m_1,m_2,\chi_1,\chi_2)$, provided subdominant modes can be neglected and that the stationary phase approximation holds. When the component spins are misaligned with the orbital angular momentum, or the waveform has higher-order modes, then the extrinsic parameters lead to a non-trivial amplitude and phase modulation [105, 106]. For illustration, in Fig. 3-1 we show the waveform of a highly asymmetric system in mass ($m_1=40\text{M}_\odot, m_2=10\text{M}_\odot$) with a circular orbit (top panel) and a highly asymmetric system in mass ($m_1=40\text{M}_\odot, m_2=10\text{M}_\odot$) with precessing orbit and higher order modes (bottom panel).

3.2.3 Template bank

The vast parameter space of GW signals $\mathcal{B}$ is continuous, but to facilitate searches with a reasonable computational cost, a dense grid of templates is selected with sufficient coverage, i.e. placed templates must have a minimal loss of SNR due to mismatch [84]. Under the assumption of a circular system which only depends on $\zeta = (m_1, m_2, \chi_1, \chi_2)$, we can define the match $\mathcal{M}$ between two templates as the maximized overlap $O(\lambda_1, \lambda_2, t)$ with respect to the time shift t, so

$$\begin{aligned} \mathcal{M}(\lambda_1, \lambda_2) &= 1 - \max_t O(\lambda_1, \lambda_2, t) \\ &= 1 - \max_t \mathrm{Re}\left\langle \hat{h}(f, \lambda_1, t) | \hat{h}(f, \lambda_2) \right\rangle \qquad (3.18) \\ &\text{where } \hat{h}(f, \lambda_1, t) \equiv h(f, \lambda_1) e^{i2\pi f t} \end{aligned}$$

The match $\mathcal{M}$ is used to define the distance between two points in $\mathcal{B}$

$$d(\lambda_1, \lambda_2) \equiv 1 - \mathcal{M}(\lambda_1, \lambda_2). \qquad (3.19)$$

An optimal template bank is composed of a minimal number of templates at a minimal SNR loss. The efficient generation of a template bank is still an active field of research and two main methods are widely being explored, namely the *stochastic method* [107, 108], which consists of including templates in the bank based on a rejection technique, and the *metric template placement*, that relies on approximating the distance between two templates with a bi-linear form known as metric [105, 109].

In Fig. 3-2 we show the template bank where each point represents a template, as a function of the progenitor masses employed by the GstLAL-based inspiral pipeline (henceforth referred to as GstLAL). The BNS region (green) covers templates with progenitor masses $m_{1,2} \in [1, 3]\,\mathrm{M}_\odot$ and dimensionless spins $|\chi_{1z,2z}| < 0.05$. The NSBH region (red) covers templates with progenitor masses $m_1 \in [3, 150]\,\mathrm{M}_\odot$, $m_2 \in [1, 3]\,\mathrm{M}_\odot$ and dimensionless spin $|\chi_{1z}| < 0.999$, $|\chi_{2z}| < 0.05$. The BBH region covers templates with progenitor masses $m_{1,2} \in [3, 400]\,\mathrm{M}_\odot$ and dimensionless spin $|\chi_{1z,2z}| < 0.999$. We can also observe that the density of the templates in the bank decreases as the total mass of the binaries ($m_T = m_1 + m_2$) increases. Typically CBC with smaller masses are longer in the sensitivity

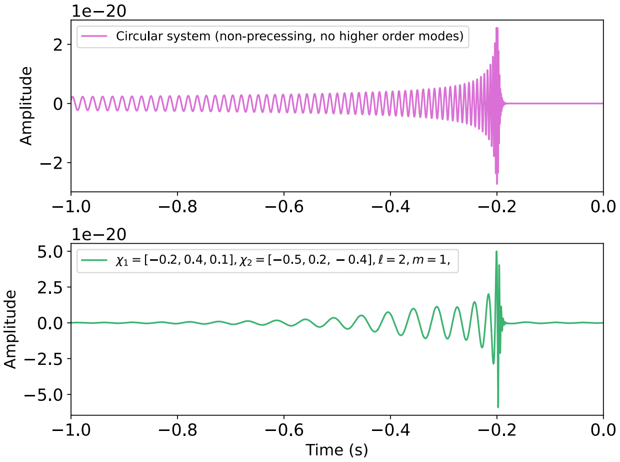


FIGURE 3-1: Waveform of a highly asymmetric system ($m_1 = 40M_\odot, m_2 = 10M_\odot$) using IMRPhenomXPHM [101]. (Top panel) Circular system with aligned spins and no higher order modes. (Bottom panel) Precessing system ($\chi_1 = [-0.2, 0.4, 0.1], \chi_2 = [-0.5, 0.2, -0.4]$) with higher-order modes ($\ell = 2, m = 1,$).

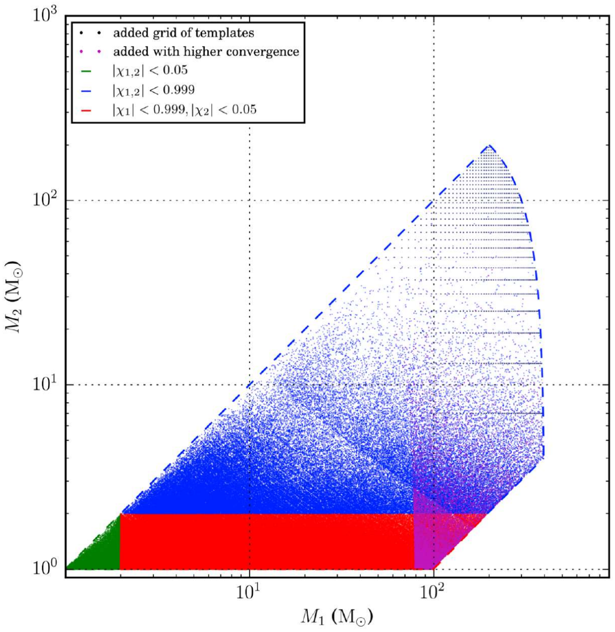


FIGURE 3-2: Template bank as a function of the component masses employed by GstLAL pipeline during O2 and O3, retrieved from [106]. Every point represents a template, either from BNS (green), NSBH (red), BNS (blue). Points in magenta and black were added to improve the background estimation for the scarcely populated region of the bank.

band of Advanced LIGO and Virgo, so these signals have more cycles than higher mass systems and their SNR loss between smaller similar templates will be larger. Thus, more templates need to cover the lower mass region.

Several current modelled state-of-the-art algorithms are matched filtering-based and need to design template banks that sufficiently cover the parameter space of CBC signals. Some examples are GstLAL [110, 106, 111, 112], PyCBC [45, 113, 114, 115, 116, 117, 118], MBTA [119, 120] and SPIIR [121]. These pipelines employ different strategies to cross-correlate template banks with the unexplored time series data from the detector, while mitigating non-Gaussian non-stationary noise, also known as glitches (see Section 2.4.3). In this particular work, we provide a high-level overview of some of the particularities of the GstLAL pipeline. For an in-depth read about the full workflow we recommend the interested reader to refer to [84, 122].

3.2.4 GstLAL

Time-domain matched filtering

We have derived the expression for the optimal SNR ρ_{opt} and the SNR ρ in Eq. 3.13. To reduce the latency of this calculation in the frequency domain, the GstLAL analysis computes the matched filtering in the time domain [122, 106, 84, 110]. To find the expression of the time-domain SNR, we first define the complex form of the SNR. The complex SNR of the ith complex template normalized $h_{c,i}(t)$ of the template bank, and the data $d(t)$ is,

$$z_i(t) = x_i(t) + iy_i(t) = 4\int_0^\infty \frac{\tilde{d}(f)\tilde{h}^*_{c,i}(f)}{S_n(f)} e^{i2\pi ft} df, \quad \text{where } 4\int_0^\infty \frac{\tilde{h}_{c,i}(f)\tilde{h}^*_{c,i}(f)}{S_n(f)} df = 1. \tag{3.20}$$

$x_i(t)$ is the matched filter response to the $+$-polarized with orbital phase ϕ_0, while $y_i(t)$ is the matched filter response to the $\times$-polarized with orbital phase $\phi_0 + \pi/2$. The output of GstLAL is the real-valued $x_i(t)$ with the real normalized template $h_i(t)$, instead of the complex-valued $z_i(t)$, which can be formulated in time domain using the convolution theorem,

$$\begin{aligned} x_i(t) &= 2\int_{-\infty}^\infty \frac{\tilde{d}(f)\tilde{h}^*_i(f)}{S_n(f)} e^{i2\pi ft}\, df = 2\int_{-\infty}^\infty \hat{d}(t+\tau)\hat{h}(\tau)\, d\tau, \\ \text{so } \hat{d}(\tau) &= \int_{-\infty}^\infty \frac{\tilde{d}(f)}{\sqrt{S_n(f)}} e^{i2\pi f\tau}\, df, \quad \hat{h}(\tau) = \int_{-\infty}^\infty \frac{\tilde{h}(f)}{\sqrt{S_n(f)}} e^{i2\pi ft}. \end{aligned} \tag{3.21}$$

where we have defined $\hat{d}(\tau)$ as the whitened data, and $\hat{h}(\tau)$ is the whitened signal. Eq. 3.21 will need to be evaluated a second time to compute the SNR response to the $\times$-polarized part, $y_i(t)$. Thus, the SNR will be

$$\rho = \max_t |x_i(t) + iy_i(t)|. \tag{3.22}$$

This is referred to as a two-phase filter, which has twice the degrees of freedom of a single-phase filter. To alleviate the duplicated computational cost GstLAL uses other methods to compute effectively [122].

Decomposition of the template bank

The high coverage of the template bank naturally leads to a high redundancy. To decompose the template bank GstLAL firstly divides it into partially overlapping sub-banks with similar frequency evolution, based on the chirp mass $\mathcal{M}$ and effective spin χ_{eff}. This process is known as "split banks" and their typical size is $\mathcal{O}(100)$ pairs of templates (one for the $+$-polarization and one for the $\times$-polarization) [122]. Afterwards, every template within a split bank is padded with zeros to have the same template length. Hence, every template can be divided into smaller pieces, with each smaller piece referred to as a time slice. Each successive time slice contains the frequency content up to a factor of two from the previous slice. This allows GstLAL to exploit the Nyquist theorem by using an appropriate sampling frequency for low- and high-frequency regions [110].

To further reduce the high dimensionality of the templates, GstLAL uses singular value decomposition, which aims to project a high dimensional input into a lower-dimensional basis. In the context of the template bank decomposition, singular value decomposition is performed in every time slice to retain only the most important basis waveform [123].

ξ^2-statistic

As we have seen in Section 2.4.3, the detector strain contains transient bursts of non-Gaussian data that can mask or mimic GW signals, producing large peaks in the SNR time series. As glitches in the detector could yield large SNRs, to mitigate them GstLAL also calculates a signal consistency check, known as ξ^2-statistic, whenever it records an SNR above a certain threshold. Such check is done by determining how similar the *SNR time series of the data* is with respect to the *expected SNR time series from the real signal* at a δt time window around the peak [110]. Mathematically, the ξ^2-statistic is defined as,

$$\xi^2 = \frac{\int_{-\delta t}^{\delta t} |z(t) - z(0)R(t)|^2 dt}{\int_{-\delta t}^{\delta t} (2 - 2|R(t)|^2) dt}, \tag{3.23}$$

where $z(t)$ is the complex SNR time series, $z(0)$ is its peak and $R(t)$ is the auto-correlation series between the complex template waveform and itself. GstLAL weights the SNR with ξ^2 defining

$$\gamma_2 = \frac{\rho}{\left(1 + \max\left(1, \xi^2\right)^3/2\right)^{1/5}}, \tag{3.24}$$

which is utilized to generate triggers [124].

Triggers

When an unknown signal is cross-correlated with the template bank, it generates SNR time series associated with the templates in the bank. When the weighted SNR, γ_2, associated with a template exceeds a certain threshold, the GstLAL pipeline records a feature vector known as a "trigger," which contains the maximum SNR, ξ^2, γ_2, masses, spins, and other parameters.

It often happens that multiple triggers from different templates match the same signal in the data (either of terrestrial or astrophysical origin). They differ in the trigger time, as well as SNR and ξ^2 values. All such triggers must be associated with the same candidate and for this reason, they need to be clustered together to form a single candidate. The GstLAL pipeline adopted a simple approach to solve this problem: the triggers with the largest SNR peak within the vicinity ($\pm 1\,$s) will be defined as the cluster centroid and utilized for further analysis[1]. While this approach has the benefits of being simple, it discards many pieces of information, such as the position of the various triggers within the template bank and their time ordering. For illustration, Fig. 3-3 shows the triggers associated with GW190521 during

---

[1] Another clustering of triggers is performed at a later stage of the pipeline. However, this has no importance for our purposes.

the intermediate-mass black hole (IMBH) search in O3 for Hanford (left panel) and Livingston (right panel). We represent the IMBH template bank (grey) as a function of the progenitor masses, and the templates that produced an SNR peak (blue). The pattern manifested across the template bank is almost identical for both detectors, implying that they formed due to the same signal of astronomical origin.

Each trigger is ranked [125, 111, 126] according to its probability of originating from an actual signal. The goal of the pipeline is then to obtain a list of triggers, ordered by their likelihood Λ to be of astrophysical origin.

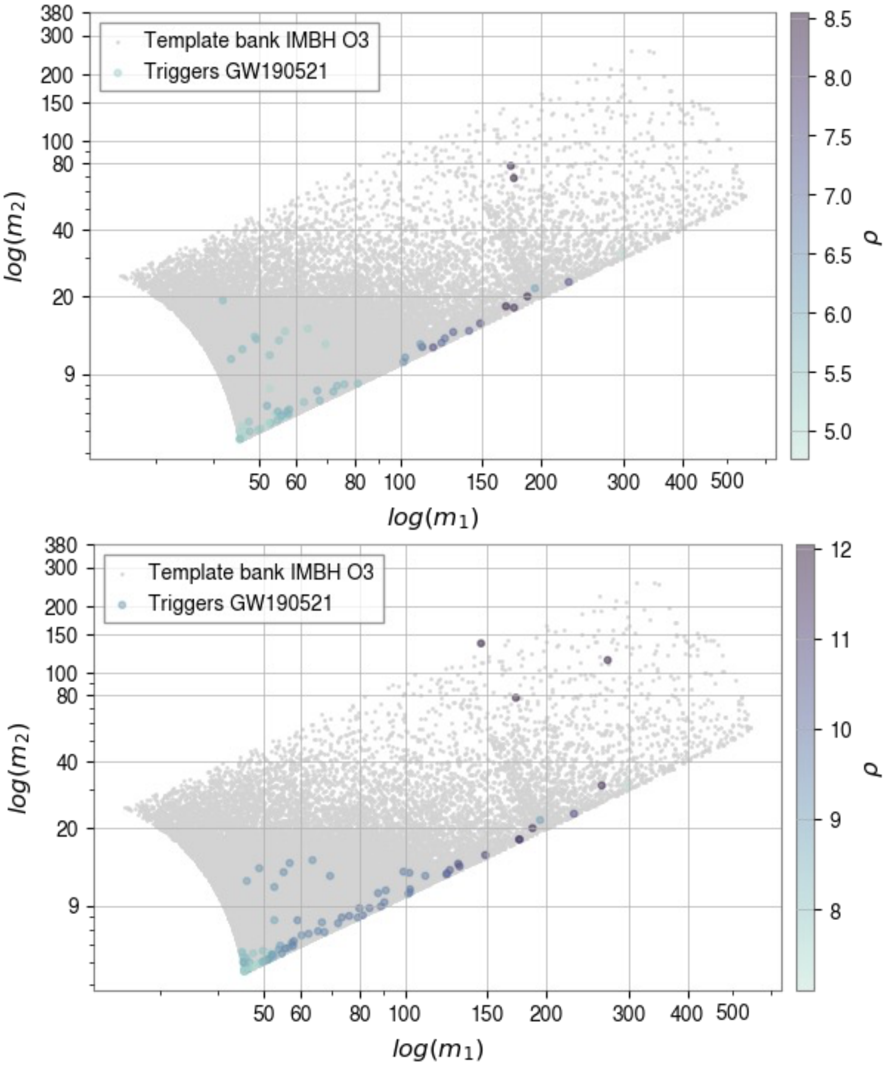


Figure 3-3: IMBH template bank from O3 (grey) as a function of the progenitors' masses measured in $M_\odot$ and logarithmic scale. We colour in blue the templates that generated a trigger with their maximum SNR ρ due to the presence of GW190521. (Top panel) Hanford detector. (Bottom panel) Livingston detector.

3.3 Unmodelled searches

While CBC signals are well-understood, other transient GW signatures arise from complex astrophysical processes that are hard to model due to their poor characterization and/or associated high computational cost. As we have seen in Section 2.1.2, this type of GW signals are known as burst, and they are characterized by having short duration ($\lesssim 1$ s) and a wide variety of frequency ranges. Because of this, GW searches need to be sensitive to a large spectrum of GW waveforms, providing the opportunity to find unexpected physical phenomena.

3.3.1 Excess power

Burst algorithms search for excess power, or energy, in time-frequency representations with minimal (targeted search) or no assumptions (generic search). Finding faint GW of unknown morphology with this method is challenging due to the presence of transient non-Gaussian and non-stationary noise, known as glitches (see Section 2.4.3). However, since glitches may be produced by the detector and/or its surroundings, it is unlikely that glitches with similar time-frequency representations occur simultaneously in detectors that are thousands of kilometres apart. Thus, burst searches exploit coincident information from the detectors in the time-frequency domain by combining the data from different detectors either *incoherently* [73, 127]—generating single detector triggers and searching for coincidences at a later stage—or *coherently* [128]—generating a unique list of triggers based on combined detector data. In the current work, we focus on the coherent case, where excess power algorithms are used to analyze network detector data using a constrained likelihood function. This function depends on the source's sky position and considers the corresponding antenna patterns of the interferometers and the time delays between interferometer pairs. Henceforth, we will refer to this method as "coherent analysis" [129].

While excess power-based algorithms cannot rely on models to provide summary statistics of GW candidates. Indeed, while CBC searches use the SNR ρ to characterize the loudness of the GW signals (Eq. 3.13), burst searches use,

$$h_{rss} = \sqrt{\int_{-\infty}^{\infty} h(t)^2 dt}, \tag{3.25}$$

which corresponds to the total GW signal energy and has units of $\mathrm{Hz}^{-1/2}$. We can also parameterize GW burst signals using their intrinsic values [130]. Assuming a well-localized burst signal $h(t)$ in time, and frequency and its Fourier transform $\tilde{h}(f)$, we can define a central time t_c, central frequency f_c, duration σ_t^2 and bandwith σ_t^2 as,

$$\begin{aligned} t_c &= \int_{-\infty}^{\infty} t\frac{|h(t)|^2}{\langle h^2\rangle}dt, & f_c &= 2\int_0^{\infty} f\frac{|\tilde{h}(f)|^2}{\langle h^2\rangle}df \\ \sigma_t^2 &= \int_{-\infty}^{\infty} (t-t_c)^2\frac{|h(t)|^2}{\langle h^2\rangle}dt, & \sigma_f^2 &= 2\int_0^{\infty} (f-f_c)^2\frac{|\tilde{h}(f)|^2}{\langle h^2\rangle}df \end{aligned} \tag{3.26}$$

where $\langle h^2\rangle = h_{rss}$. We can also define the dimensionless quality factor as $Q = \sqrt{2}f_c/\sigma_f$, which describes the characteristics of the signal in terms of its frequency localization and how it changes over time. While the exact definitions of these quantities are algorithm- and representation-dependent, it is common to use this set of parameters to express the time and frequency content of the signal.

The frequency content of the GW waveform is unknown a priori, so burst searches do not rely on Fourier transforms, which are better suited for global frequency analysis, and instead they use *wavelet transforms*.

3.3.2 A short introduction to wavelet analysis

The idea of approximating signals using the superposition of functions has existed since the early 1800s following the discovery of Fourier transforms. The main accomplishment of Fourier transforms is to analyze a time domain signal for its frequency content, as the Fourier coefficients of the transform represent the cosine and sine contribution at each frequency. However, there was a need to extend beyond Fourier analysis, as it is inadequate for handling signals with sharp transitions. Fourier series convergence and orthogonal systems gradually lead to a transition from *frequency analysis* to *scale analysis*, i.e. analyzing functions by creating mathematical structures that vary in scale, which is the fundamental idea behind wavelet analysis.

Wavelets process data at different *scales* or *resolutions*, such that, in a sense, we can see the forest *and* the trees. If we look at a signal with a large “window”, we will notice gross features, while if we use a small “window”, we will notice smaller details. The main procedure of wavelet analysis is to adopt a wavelet prototype function, known as *mother wavelet* that will act as the basis of the analysis. Then, the temporal analysis is performed with a contracted, high-frequency version of the mother wavelet, while the frequency analysis is performed with a dilated, low-frequency version. This procedure is also less sensitive to noise as it measures the average fluctuations of the signal at different scales. The interested reader may refer to [131] for details.

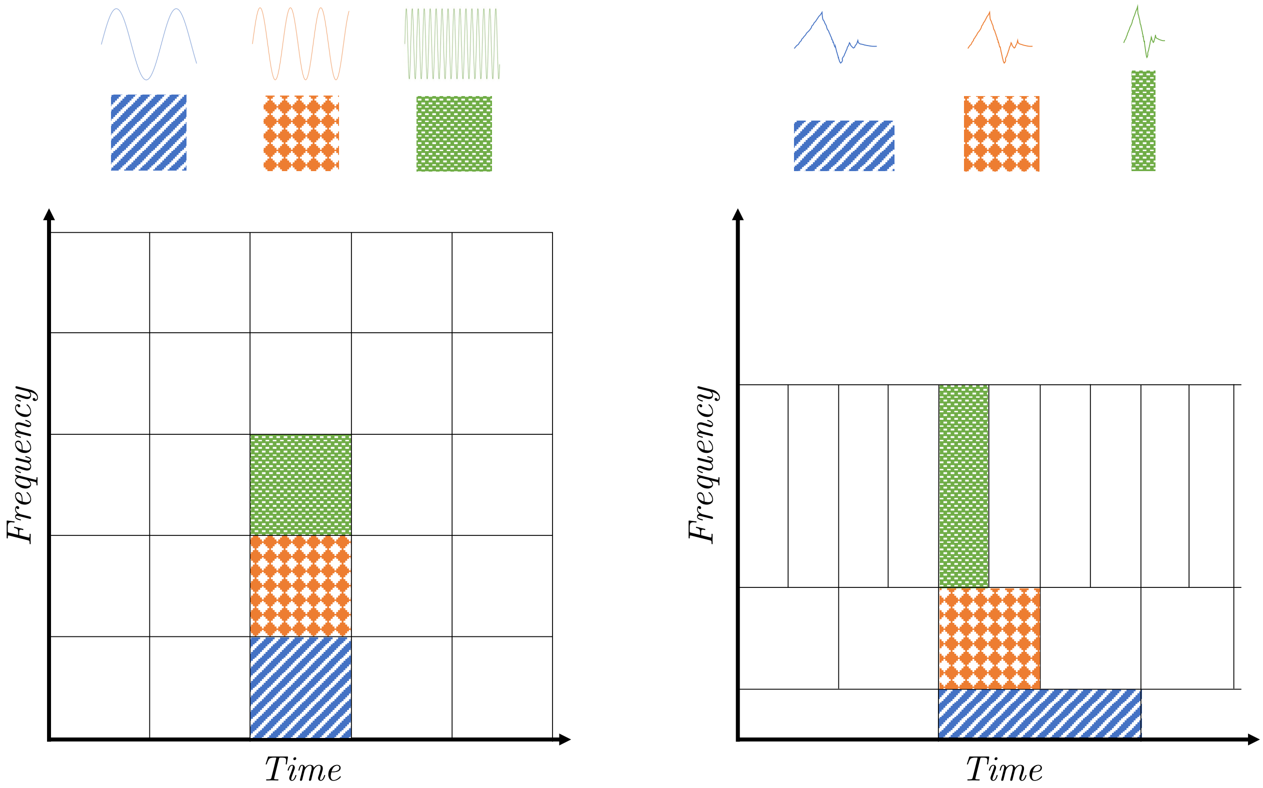


FIGURE 3-4: (Left): Fourier transform with a fixed window in the time-frequency plane. The same rectangular window is used to analyze high and low frequencies. (Right): Wavelet transform with a sketch of Daubechies wavelets in the time-frequency plane. The different window sizes allow us to see small and gross features. Reproduced from [131].

To highlight the main differences between the Fourier and wavelet transform, in the left panel of Fig. 3-4, we represent a schematic Fourier transform with a rectangular window of a certain size. Because the size of the window is constant, the resolution of the analysis is the same at all locations in the time-frequency plane. This uniform resolution can be a limitation when analyzing signals with varying frequency content, as it does not allow for better time resolution at high frequencies.

In contrast, the Short-Time Fourier Transform (STFT) allows for some flexibility by enabling the use of shorter time windows at high frequencies, thereby improving time resolution for those components. However, the STFT still lacks the inherent adaptability of wavelet transforms. In Fig. 3-4 (right panel), we illustrate the coverage of a sketch of Daubechies wavelets [132] with varying windows. Wavelets naturally adjust their window sizes based on frequency content, offering a significant advantage over both the standard Fourier analysis and STFT.

Indeed, to effectively isolate signal discontinuities, it is beneficial to have short basis functions for high frequencies and long basis functions for low frequencies. This adaptive capability of wavelets allows for a fine time resolution in the presence of high-frequency components while still providing detailed frequency analysis for low-frequency components. As a result, wavelet transforms are particularly well-suited for analyzing signals with non-stationary characteristics, where a balance between time and frequency resolution is essential.

Mathematically, the continuous wavelet transform of a continuous signal $x(t)$ with the mother wavelet ψ is defined as [130, 133],

$$X(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\psi^* \left(\frac{t-b}{a}\right) dt \tag{3.27}$$

where ψ^* is the complex conjugate of the mother wavelet ψ time-shifted by b and scaled by a. The wavelet is normalized by a factor $1/\sqrt{a}$ to ensure that each wavelet has the same energy independently of the scaling and shifting. Thus, the properties of the mother wavelet ψ at each scale determine our ability to resolve in time and frequency. For small values of a we achieve a large time resolution at the expense of uncertainty in frequency, while for large values of a we have an improved frequency resolution at the expense of time. Note that the quality factor Q is constant $\forall\, a$.

The mother wavelet $\psi(t)$ must satisfy the condition of finite energy and admissibility, which are respectively defined as,

$$E_\psi = \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty, \qquad C_\psi = \int_{-\infty}^{\infty} \frac{|\tilde{\psi}(\omega)|^2}{\omega} d\omega < \infty \tag{3.28}$$

where $\tilde{\cdot}$ represents as usual the Fourier transform and $\omega = 2\pi f$ is the angular frequency. The admissibility constant C_ψ forces $\tilde{\psi}(\omega)$ to decay faster than ω towards zero with $\tilde{\psi}(\omega) = 0$.

Wavelet transforms are more flexible than Fourier transforms, as they have infinite possible sets of basis functions. Finding an appropriate set of basis functions for wavelet applications is non-trivial and it is an active field of research also in the GW community. In the following Section, we provide details about the wavelets employed by *coherent WaveBurst* (cWB): a burst pipeline that uses a coherent excess power analysis among detectors with wavelet packet transforms, a far more versatile transform than wavelets. Wavelet packet transforms have the same time-frequency resolution at each scale, such that, in the case of cWB, the time resolution doubles while the frequency halves at every subsequent scale. For illustration we show in Fig. 3-5 the time-frequency representation of the first detection, GW150914, comparing the spectrograms of cWB, which uses wavelet packet, and Omicron, which uses a modification of the Fourier transform, known as Q-transform [73].

In the following Section, we describe the wavelet packet of cWB, their data conditioning and multi-resolution analysis. We recommend to the interested reader to refer to [129] and [134], and references therein for an in-depth description of the full pipeline.

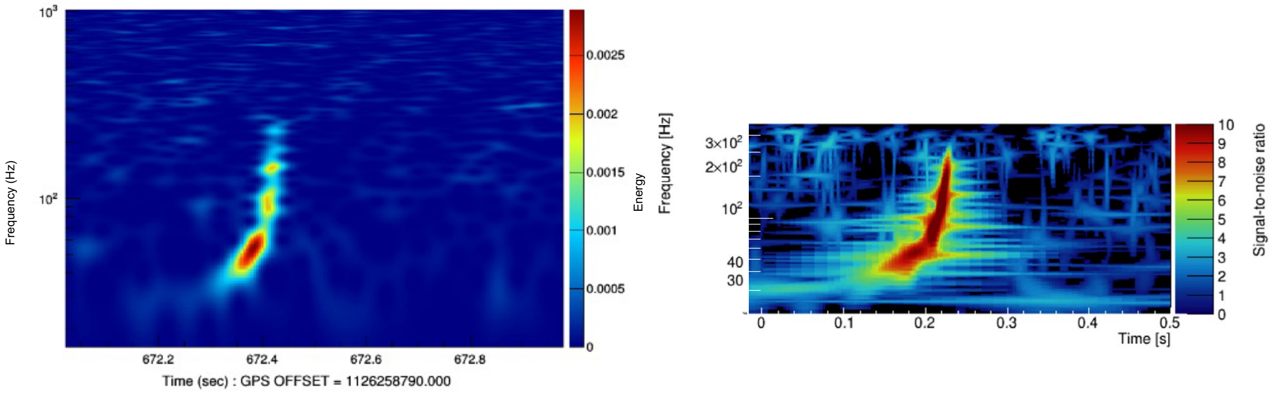


Figure 3-5: GW150914 in LIGO Hanford. Left: cWB spectrogram, first time-frequency representation of a GW signal. Right: Omicron spectrogram retrieved from [73]. Note the reduction on the spectral leakage of cWB spectrogram with respect to the Omicron spectrogram.

3.3.3 Coherent WaveBurst

Wilson-Daubechies-Meyer bases

As the appropriate selection of a wavelet basis is fundamental for detecting burst GW signals, Gabor bases and their variations are commonly employed for GW analysis due to their flexibility, analytic Fourier representation, and immediate projection onto the various detectors described by a simple phase shift [127, 135]. Still, they do not form an orthogonal basis, as the wavelets are highly correlated. Utilizing a local orthonormal basis of wavelets has several advantages: the energy is conserved, the inverse wavelet transform is well-defined, and the time-frequency data samples are statistically independent [129].

For this aim, cWB combines the conventional Wilson-Daubechies transform and the Meyer wavelet, forming the Wilson-Daubechies-Meyer (WDM) wavelet transform [136]. It is compactly supported in the frequency domain, i.e. it vanishes outside a finite interval and it decays faster than any inverse polynomial in the time domain, making it particularly convenient for transient signals. The main features of WDM transform are a fast computation via fast Fourier transform, shorter transformation filters and an improvement over spectral leakage, i.e. a better time and frequency localization, respectively, as well as a simple analytic expression for the time-delay filters. This is observed in the left panel of Fig. 3-5, which presents less spectral leakage than the right panel. For the interested readers, we derive the formalism of WDM bases in Appendix A.2.

Data conditioning

While the fast Fourier transform eases the wavelet calculation, cWB still needs to perform a coherent search among detectors. To compute these expensive calculations cWB relies on ROOT [137], a C++ library developed by CERN for fast data analysis. Furthermore, cWB is designed to be highly parallelizable with a low computational load.

As we have seen in the previous chapter, searches of GW are hampered by different sources of fundamental noises that arise from the different subsystems within the detector and its environment. In searches for the unknown, such as the one performed by cWB, a proper understanding of the background for its subsequent mitigation is crucial to claim the detection of a transient GW signal. For this aim, cWB usually analyzes time periods between 300 s and 600 s, as lowering the duration of the analyzed data would bias the evaluation of the background. The cWB methods to mitigate transient and persistent noise are as follows.

Linear prediction error: In Section 2.4.2 and 2.4.3 we have described the persistent and transient noise of the detector. These types of noise hinder the sensitivity of unmodelled algorithms, so to mitigate it authors in [130] have proposed an algorithm called linear prediction error, which assumes that the n^{th} sample of a sequence $x[n]$ is modelled by a linear combination of the previous M samples, such that the error of the predicted sequence is

$$e[n] = x[n] - \hat{x}[n] \qquad \text{where } \hat{x}[n] = \sum_{m=1}^{M} c[m]x[n-m], \tag{3.29}$$

where $\hat{x}[n]$ the predicted sequence, and $c[m]$ the coefficients which are chosen to minimize $\sigma_e = 1/N \sum_{n=1}^{N} |e[n]|^2$ over a representative training sequence of length N. Assuming $x[n]$ to be a stationary and stochastic process, we can predict the coefficients $c[m]$ using the Yule-Walker equations for auto-regressive processes with lag k,

$$\sum_{k=1}^{M} r[m-k]c[m] = r[k] \quad \text{where } r[k] = \frac{1}{N}\sum_{n=|k|}^{K} x[n]x[n-|k|] \quad \forall\, k \in [1, M]. \tag{3.30}$$

cWB implements this method to predict the behaviour of safe auxiliary channels, i.e. channels insensitive to GW, and subtract these disturbances on the main strain of the detector $h(t)$, such as noise lines and glitches on the frequency band of interest of burst searches (see [138, 139] for details). While in [130] the authors apply it on time series, cWB implemented it with the wavelet packet transform, applying it to each wavelet layer w of the WDM decomposition. In this way, the linear prediction error in the wavelet domain removes data artefacts but preserves the PSD of the noise floor [140].

Whitening: as we have seen in Section 2.5.4, whitening is a standard procedure for GW time series analysis. cWB also employs whitening procedure to highlight transient GW with WDM decomposition. First, we shall remember that cWB is a coherent analysis of d detectors (or networks), such that the discrete time series is represented in the time-frequency domain via wavelet packet transform, namely $x_d[k]$. Here, k represents the k^{th} pixel of the time-frequency representation. Assuming Gaussian noise of detector d, we can calculate the PSD $S_d[k]$ of the data $x_d[k]$ with WDM decomposition, such that the whitened data will be $w_d[k] = x_d[k]/\sqrt{S_d[k]}$.

Multi-resolution analysis

After the conditioning of the data with WDM transform, the time-frequency representations from all detectors are combined to obtain the time-frequency-energy maps $E[k] = \sum_d w_d^2[k]$, where $E[k]$ is maximized over all possible sky locations. When applying the wavelet transform, the algorithm selects the most energetic pixels (core), above a pre-defined threshold dependent on the noise floor, and their neighbours (halo). Thus, core and halo pixels are combined in a cluster, representing an event. Such procedure is repeated for different decomposition levels forming a super-cluster. Then, cWB estimates the likelihood Λ of the clusters and super-cluster being a GW signal, forming a list of candidates. For details on the likelihood calculation, the interested reader shall refer to [129] and references therein.

3.4 Background estimation

As we have described in the previous sections, in a nutshell, a search pipeline is a sequence of operations that inputs the raw data of the interferometer and produces a list of detection events or triggers, i.e. potential GW candidates, solving a hypothesis testing (see Section 3.1). Nonetheless, after forming a list of triggers we still need to compute their significance with respect to an accidental background.

3.4.1 Poisson process

Let us imagine an ideal detector where we can measure triggers caused by an astronomical source of GW. In our experiment of duration τ, we monitor $\lambda(t)$, the cumulative count of triggers as a function of time. We can repeat our experiment of duration τ N times, taking the average of $\lambda(t)$, $\bar{\lambda}(t)$. If the source we are measuring is stationary, the average $\bar{\lambda}(t)$ would be a linear function of time, but if the source is non-stationary, $\bar{\lambda}(t)$ will have a non-linear dependency. As we are interested in a duration-independent quantity, we define $\mathcal{R}(t)$ the rate density of the source

$$\mathcal{R}(t) \equiv \frac{d\lambda}{dt}. \tag{3.31}$$

Contrary to $\lambda(t)$, which mixes the intrinsic parameters of the source with an extrinsic parameter (the observation time), $\mathcal{R}(t)$ depends only on the intrinsic parameters. If we were to repeat our experiment infinite times, obtaining an infinite number of independent measurements in identical conditions over a duration τ, we would obtain the "true" rate density

$$\widetilde{\mathcal{R}}(t) \equiv \lim_{N\to\infty} \frac{\sum_{j=1}^{N} \widehat{\mathcal{R}}(t)_j}{N}, \tag{3.32}$$

where $\widehat{\mathcal{R}}$ denotes the measured rate density. From the definition in Eq. 3.31, we can calculate the mean number of events produced during the experiment.

$$\lambda = \int_0^{\lambda} d\lambda' = \int_0^{t} \widetilde{\mathcal{R}}(t')dt' \underset{\widetilde{\mathcal{R}}(t')\to\widetilde{\mathcal{R}}}{=} \widetilde{\mathcal{R}}t \tag{3.33}$$

If the measured source is stationary, the probability distribution of getting n events related to this system in a period t is given by the well-known Poisson distribution,

$$P(n|\lambda) = \frac{\lambda^n e^{-\lambda}}{n!}, \tag{3.34}$$

and the expected number of events is

$$\mathbb{E}[N] = \sum_{j=1}^{\infty} jP(j|\lambda) = \sum_{j=1}^{\infty} j\frac{(\widetilde{\mathcal{R}}t)^j e^{-\widetilde{\mathcal{R}}t}}{j!} = \widetilde{\mathcal{R}}te^{-\widetilde{\mathcal{R}}t} \sum_{j=1}^{\infty} j\frac{(\widetilde{\mathcal{R}}t)^{j-1}}{j-1!} = \widetilde{\mathcal{R}}te^{-\widetilde{\mathcal{R}}t}e^{\widetilde{\mathcal{R}}t} = \widetilde{\mathcal{R}}t \tag{3.35}$$

Therefore, in a single experiment we can measure the "true" rate density $\widetilde{\mathcal{R}}$ by dividing the number of events produced by the period. Hence, defining t_j and $\widehat{N}_j$ as the duration and the number of triggers produced by the j^{th} experiment, the average over N experiments yield,

$$\overline{\mathcal{R}} = \frac{\sum_{j=1}^{N} t_j \widehat{\mathcal{R}}_j}{\sum_{j=1}^{N} t_j} = \frac{\sum_{j=1}^{N} t_j \left(\widehat{N}_j / t_j\right)}{\sum_{j=1}^{N} t_j} = \frac{1}{T}\sum_{j=1}^{N} \widehat{N}_j, \quad \text{where } T = \sum_{j=1}^{N} t_j \tag{3.36}$$

If each experiment has duration $\approx \tau$, we can simplify the average rate density as

$$\overline{\mathcal{R}} = \frac{\sum_{j=1}^{N} \widehat{N}_j}{\tau N} = \frac{\overline{N}}{t}, \qquad \text{with variance } \delta\overline{\mathcal{R}} = \frac{1}{\tau}\sqrt{\frac{\overline{N}}{N}}, \tag{3.37}$$

where we are taking into consideration that $\overline{N}$ is produced by a Poisson process with variance $\overline{N}$. Once we have measured the average rate density $\overline{R}$ we can estimate the probability of getting n events from the measured source for any period t,

$$\begin{aligned} P(j|\overline{\mathcal{R}}t) &= \frac{(\overline{\mathcal{R}}t)^j e^{-\overline{\mathcal{R}}t}}{j!}, \\ \text{with error } \delta P &= \left|\frac{dP}{d\overline{\mathcal{R}}}\right| \delta\overline{\mathcal{R}} = \frac{t}{\tau}\sqrt{\frac{\overline{N}}{N}} e^{-\overline{N}t/\tau} \left|\frac{(\overline{N}t/\tau)^{j-1}}{(j-1)!} - \frac{(\overline{N}t/\tau)^j}{j!}\right| \end{aligned} \tag{3.38}$$

Particularly, we are interested in obtaining $j \geq 1$ triggers, such that,

$$P(j \geq 1|\overline{\mathcal{R}}t) = 1 - P(0|\overline{\mathcal{R}}t) = 1 - e^{-\overline{R}t}, \tag{3.39}$$

Note that this calculation assumes a stationary source, where the merger rate is not a function of time. If the source is non-stationary, $\widetilde{\mathcal{R}}(t)$ does not exist, since it is impossible to distinguish between variations produced by a random process and statistical fluctuations due to a changing rate density. Thus, to improve our measurement of $\widetilde{\mathcal{R}}(t)$, we must observe the source for a longer time. Nonetheless, this results in a worse measurement of $\widetilde{\mathcal{R}}(t)$, as we can only observe for a finite period at a given scale so that smaller fluctuations will be indiscernible. Since the interferometers are evolutionary machines, we need to adjust the observation time to the data quality flags, described in Section 2.6.

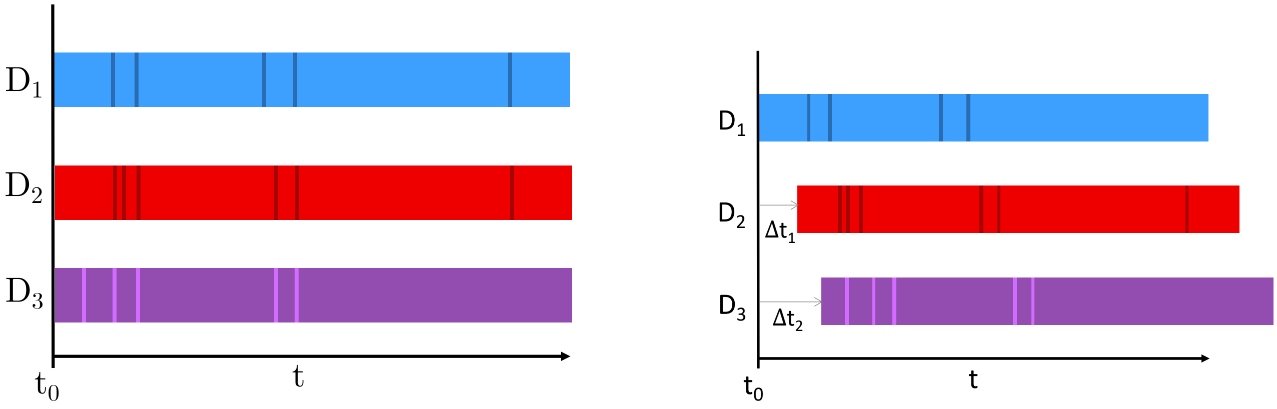
This derivation provides a robust framework for analyzing the rate of GW events detected by an ideal detector. By defining the rate density $\overline{\mathcal{R}}(t)$, we can distinguish the intrinsic properties of the astronomical source from observational parameters, allowing for a more accurate characterization of the source's behaviour. This analysis also highlights the importance of repeated experiments to obtain reliable estimates of the rate density $\overline{\mathcal{R}}(t)$ and provides tools for calculating the significance of observing events, accounting for measurement uncertainties. In the next section, we describe how to compute the significance of a GW candidate in a more realistic scenario.

3.4.2 Time-shift analysis and false alarm rate

As discussed in Section 2.4.3, the presence of transient non-Gaussian and non-stationary noise, known as glitches, hampers the detection of GW signals. Indeed, if a trigger is produced in a single detector, it is challenging to assess if it has an astronomical or terrestrial origin. Nonetheless, as the production of glitches is localized within the detector, it is unlikely that glitches with similar characteristics occur simultaneously in detectors that are thousands of kilometres apart. Similarly to Burst searches, that exploit time-coincidence between detectors (see Section 3.3), we can also use this idea to assess the significance of the triggers of our experiment, i.e. estimate the accidental background.

Let us imagine we have several GW interferometers connected as a network of I detectors. Let us also consider a trigger produced in the network of detectors with statistic Λ^* at $t_c \pm \Delta t$, where Δt includes the light travel time between detectors and statistical fluctuations. We could measure the uniqueness of our coincident trigger by comparing it to the population of triggers of the experiment, known as *foreground*. However, it is challenging to assess the origin of the foreground, so we need to construct a *background* population, where there is no correlation between detectors induced by a GW signal.

In Fig. 3-6a we can observe a network $I \equiv \{D_1, D_2, D_3\}$, where the measurement started at



(A) Foreground or zero-lag data (B) Background or time-shifted data

FIGURE 3-6: Sketch of data in a network of detectors.

t_0. The vertical lines indicate the production of foreground triggers. In this sketch, we depict single triggers, happening in individual detectors, double triggers, occurring in two detectors simultaneously, and triple triggers, appearing simultaneously in all detectors. To construct a *background* population where the triggers from different detectors are uncorrelated, we can time-shift the detectors with respect to another detector. In Fig. 3-6b we fix D_1 and time shift D_2 and D_3 relative to D_1 by Δt_1 and Δt_2, respectively. Subsequently, the coincident triggers observed in the time-shifted data will serve as the background triggers, which are accidental coincidences and not due to actual astrophysical events. Note that Δt_1 and Δt_2 need to be large enough to ensure that the slide is roughly independent, a requirement that may vary depending on the search.

To increase the significance of our foreground trigger, we need to compare it with an extensive population of background triggers. Hence, we make the process of time-sliding the data Δt_l iterative by defining the time shift $T_l \equiv l\Delta t_l$, where l is known as *lag*. When $l = 0$ we recover the foreground, which is also known as *zero-lag*. Another relevant quantity is the total observing time of the foreground or *search time* T_f, and the total observing time of the background T_b. In our setup, for the l^{th} time-shift $T_f > T_{b,l}$, since the overlap between detectors will decrease with increasing offsets. Nonetheless, it is possible to perform time shifts on a *ring*, where data that slides past the end of the segment is placed at the beginning. Note that even in this scenario, time shifts will result in different durations due to gaps in the data. To achieve a significant estimation of the background we would perform L time shifts, such that $T_f \ll T_b = \sum_{l=1}^{L} T_{b,l}$.

To assess the significance of our trigger with statistic Λ^* from the foreground, we compare it with triggers produced within the background, known as *false alarms*. We can define the *false alarm probability* P_F, as the probability of obtaining $j \geq 1$ triggers with a statistic $\Lambda \geq \Lambda^*$ within the background, during the time of the search T_f. From Eq. 3.39 this is

$$P_F(j \geq 1|\mathcal{F}(\Lambda^*), T_f) = 1 - e^{-\mathcal{F}(\Lambda^*)T_f}. \tag{3.40}$$

Here, we define $\mathcal{F}(\Lambda^*)$ as the *false alarm rate* of a trigger with statistic Λ^*, measured as [103]

$$\widetilde{\mathcal{F}}(\Lambda^*) = \frac{\sum_{l=i}^{L} \widehat{N}_l(\Lambda \geq \Lambda^*)}{T_b}, \tag{3.41}$$

where $\widehat{N}_l$ represents the false alarms of the l^{th} time-shift, i.e. the number of background triggers within the l^{th} time-shift with statistic $\Lambda \geq \Lambda^*$. As before, T_b is the total background time. It is relevant to note that for each slide the duration of $T_{b,l}$ will vary, so we define the *effective number of slides* $\widetilde{N} \equiv T_b/T_f$, so that

$$\widetilde{\mathcal{F}}(\Lambda^*) = \frac{\sum_{l=i}^{L} \widehat{N}_l(\Lambda \geq \Lambda^*)}{T_b T_f / T_f} = \frac{\widehat{N}(\Lambda \geq \Lambda^*)}{T_f \widetilde{N}} = \frac{\overline{N}(\Lambda \geq \Lambda^*)}{T_f}. \tag{3.42}$$

We define $\widehat{N}(\Lambda \geq \Lambda^*)$ as the total number of background triggers with $\Lambda \geq \Lambda^*$ in all time shifts, while $\overline{N}(\Lambda \geq \Lambda^*)$ is the average number of false alarms in a single experiment with a T_f duration. Since this is the mean of a stationary Poisson process, so from Eq. 3.37 the error is

$$\delta\widetilde{\mathcal{F}}(\Lambda^*) = \frac{1}{T_f}\sqrt{\frac{\overline{N}(\Lambda \geq \Lambda^*)}{\widetilde{N}}} = \frac{1}{T_f}\sqrt{\frac{\widehat{N}(\Lambda \geq \Lambda^*)}{\widetilde{N}^2}} = \frac{\sqrt{\widehat{N}(\Lambda \geq \Lambda^*)}}{T_b} \tag{3.43}$$

This method allows us to create an accidental background, i.e., a large population of uncorrelated background triggers, through iterative time shifts. In this way, we obtain a robust statistical framework for assessing the significance of GW candidates, enabling us to quantify the likelihood that a given trigger is due to a genuine astronomical source.

Part II

Developing a Machine Learning-Based Detection Algorithm

Chapter 4

Machine Learning Foundations

The contemporary world has been undoubtedly marked by the birth of the Internet, allowing communications between networks and devices far apart. Immediately following its development came the creation of the World-Wide Web at CERN, also called *the Web*, to make the collective knowledge of humans available to each individual using machines [141]. The Web serves as a platform to access, create, share and store information in a wide range of formats, generating rapidly enormous and complex data sets commonly referred to as Big Data, which require advanced processing techniques to extract valuable insights. Indeed, we are currently facing the era of Big Data, where we not only generate a tremendous amount of information daily via social media, business transactions or mobile devices, among others [142], but we also create gigantic experiments to unveil the mysteries of the Universe, such as the Alice [143], the James-Webb telescope [144], LIGO-Virgo-KAGRA detectors, and the future Deep Underground Neutrino Experiment [145], just to name a few.

These huge data sets are not only filling up terabytes of storage at vertiginous speeds but also creating novel jobs to analyze them in the search for useful information, be it a more efficient business strategy or new physics. In the era of Big Data, scientists are revisiting algorithms from the early 60s and 70s to exploit these data sets, where the idea of machines learning to perform a task based on previous experience was explored. Indeed, the "field of study that gives computers the ability to learn without being explicitly programmed" was already coined as *Machine Learning* by Arthur Samuel back in 1959 [146]. Nonetheless, it was only in the past decade that machines were able to learn mainly due to the following reasons:

1. *Big Data*: as machines can learn from experience, it is fundamental to have a large number of examples, which contribute to the effectiveness and generalization of these models.

2. *Hardware*: in the previous century computing resources were far more limited and inefficient than nowadays. It was not until 1999 that `Nvidia` created their first graphics processing unit (GPU), which accelerates model calculations.

3. *Software*: open-source, scientific computing frameworks played a pivotal role in the development of the field and its worldwide applications. These frameworks provide efficient tools for the computation of models, accessible to both beginners and experts.

Nowadays, Machine Learning (ML) is a booming enterprise, with billions of dollars being invested in the field, and more than 100 scientific articles being published daily. Furthermore, ML is a transversal tool at the intersection of Statistics and Computer Science, with applications ranging from the medical field [147] to economics and finance [148]. In this regard, the field of GW is not an exception, but before glancing at the current state-of-the-art, we will need to understand a few concepts. In the present chapter, we will give an overview of the different ways that machines can learn from data, how we can design our algorithm by simplifying the human brain, and how we can enhance it by exploiting the particularities of the data.

4.1 On the learning landscapes

Depending on the objective of the learning, as well as the data availability, the model will need a different strategy to learn. In ML we can divide the learning strategies between *supervised*, *unsupervised* and *reinforcement*.

Supervised learning: this type of learning requires input examples X associated with a label y, also known as the *target*, to perform future predictions based on past evidence. We denote $\mathcal{X}$ as the space of the inputs, and $\mathcal{Y}$ as the space of the targets. Thus, the algorithm's goal is to learn a mapping function $h : \mathcal{X} \to \mathcal{Y}$. We can split the learning into three stages:

1. *Training:* in this stage the algorithm sees all the pairs (X_t, y_t) for $t \in 1, \ldots, T$, where T is the size of the *training data set*. In this stage, the goal of the model is to learn the mapping h by minimizing a loss function (see Section 4.2.1). This stage is similar to the situation where a student is given lectures to learn a certain task.
2. *Validation:* in this stage the algorithm sees all the pairs (X_v, y_v) for $v \in 1, \ldots, V$, where V is the size of the *validation data set*. The algorithm is not learning to perform the task, but rather it is doing a preliminary test of its performance. The main goal of this stage is to prevent *overfitting* [149], i.e. the algorithm fits the training data perfectly, but it fails to generalize in the validation data. The validation stage also serves to improve the performance of the algorithm by adjusting it via *fine-tuning* [150]. It is useful to compare this stage to the situation where a student is given a mock exam to check what they have learned.
3. *Testing:* in this stage the algorithm sees all the pairs (X_e, y_e) for $e \in 1, \ldots, E$, where E is the size of the *testing data set*. In this phase, the algorithm is on its final version, and it is evaluated to provide an unbiased performance. It is useful to compare this stage to the situation where a student is given a final exam to test their capabilities.

It is important to note that the *training*, *validation* and *testing* input data sets come from the same initial data set. Then, it is imperative to divide the initial data set, so that the examples are not repeated in the other subsets. As ML algorithms need a lot of examples to learn, a rule of thumb to split the original data set is 80% for training, 10% for validation, and 10% for testing.

When the target is discrete, we refer to the task as *classification* as the labels are predefined classes. In turn, a classification task can be *binary*, if we want to differentiate two different classes, or *multi-class* if we want to differentiate between more than two classes. On the other hand, when the target is continuous, we refer to the task as *regression*. If our goal is to simply estimate a single variable from the input data we call it *univariate* regression, but if we want to estimate several variables that are dependent on the input data we call it *multivariate* regression. In Chapter 8 and 9 we use two different supervised classification algorithms for GW searches.

While supervised learning has several applications, these methods also have some limitations: generating labels is expensive, and usually it requires the arduous work of experts, which in turn can bias the labelling process. Another problem is that supervised algorithms are unable to handle unknown classes.

Unsupervised learning: supervised learning relies on knowing the target y, but for some applications, this information is not available due to lack of resources, or simply because the ground truth is unknown. Conversely, unsupervised learning does not require labels and only relies on the input examples X to capture the underlying distribution of the input, finding relationships or patterns in the data set. An example of unsupervised learning would be clustering, which based on unordered inputs, arranges them, for example, based on their similar characteristics. In this context, the algorithm minimizes a loss function based on the similarities or "distances" between examples. In Chapter 6 we use a supervised algorithm applied to the field of GW.

Some limitations of unsupervised approaches are that evaluating their performance and interpretation is often challenging due to the lack of ground truth. Related to the previous point, avoiding over-fitting is more difficult, as the algorithms can capture irrelevant or noisy features of the data.

Reinforcement learning: This type of learning strongly overlaps with the field of Artificial Intelligence, where the algorithm does not learn to perform a task based on input data, with or without labels, but it rather it takes decisions by interacting with the environment. In this framework, the algorithm, or *agent*, is located in an environment, or *world*, with certain rules. The agent will move in this world, passing from one *state* to another by maximizing a *reward function*, which specifies the learning task. If we give the proper "hints" to the agent, it will successfully learn what states are preferred, and what states it should avoid. Reinforcement learning is closest to the way we, humans, learn, and it is also one of the most complex. Applications of reinforcement learning are, but not limited to, robotics and control systems [151]. While in this dissertation we do not use reinforcement learning, the interested reader shall refer to [152].

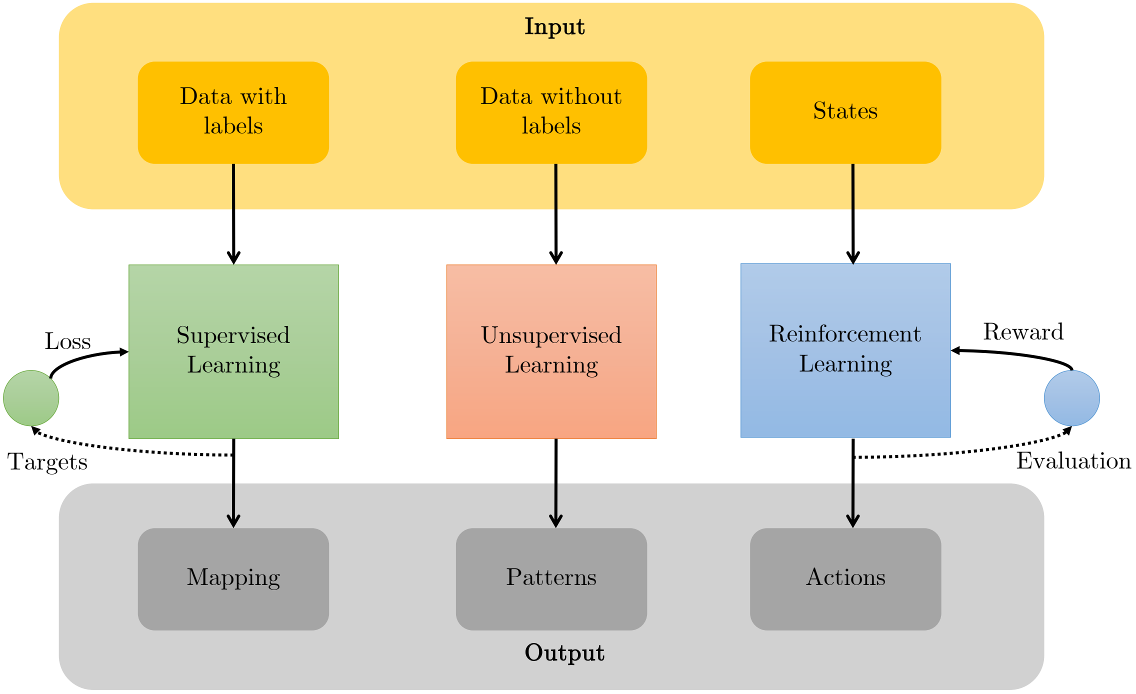


FIGURE 4-1: Sketch of the different types of ML, namely, supervised (green), unsupervised (orange) and reinforcement (blue).

In Fig. 4-1 we present a sketch summarizing the main three different types of learning. However, these learnings can be mixed, giving place to semi-supervised approaches [153] or active learning [154], among others. Now that we have seen the different ways algorithms learn, in the following we explain how we can design a machine that learns.

4.2 The realm of deep learning

The success of ML lies in creating algorithms able to learn from past experiences, and perform future predictions. While several ML algorithms are capable of generalizing, they commonly suffer from the *curse of dimensionality*, i.e. the explosive nature of high-dimensional spaces results in an exponential increase of the computational efforts to learn from the input data [150]. The curse of dimensionality does not only impact their computational cost, but also it decreases the performance of algorithms, since with a linear increase of the dimensionality, the number of examples needed to achieve a good performance increases exponentially. Nonetheless, a ML algorithm can remove irrelevant features in the data, learning useful information, and beating the curse of dimensionality, namely, artificial neural networks. Note that in this particular work, we do not provide an overview of other ML algorithms, but we recommend the interested reader to refer to [150].

FIGURE 4-2: Representation of a galaxy by DALL·E 2 [155], a DL model.

The idea of mathematically defining the human brain has been around for almost 100 years, but it was not until the birth of Big Data that these ideas became a reality. Indeed, artificial neural networks, in combination with Big Data and the improvement of hardware and software, opened the door to the realm of Deep Learning (DL). In this realm we can find applications from the autonomous car [156], to text [157] and art generation [158]. As an example, in Fig. 4-2 we show the picture of an artificial galaxy generated with DALL·E 2 [155]. In this section, we present the formalism of artificial neural networks, the building blocks of Deep Learning (DL), based on the material of [159].

4.2.1 Simplifying the human brain

The architecture of a neuron: the perceptron

The human brain is a complex organ that serves as a command centre of the nervous system, being composed of billions of neurons, which are its basic computational unit. The main tasks of the neurons are to receive information, via its *dendrites* as an electromagnetic pulse, then interpret the signal in the *nucleus* and create, or *fire*, a response signal that is carried to the next neuron via the *axon*, which connects to other neurons via *synapses*.

In 1958, psychologist Frank Rosenblatt introduced the first mathematical simplification of the human neurons, known as *perceptron* [160]. In this mathematical neuron, the information is given by a n-vector $x = [x_1, \ldots, x_n]$. Similarly to the previous case, x travels along the axon and interacts multiplicatively with the dendrites via the synapsis, or weights, $w = [w_1, \ldots, w_n]$.

The goal of the mathematical neuron is to learn the synaptic strength, which controls the strength of the influence of the previous neuron. In a simplistic model, the information of the dendrites is summed, such that if the sum is above a certain threshold, the neuron fires a signal along the axon. This *firing rate* is modelled as the activation function $f : \mathbb{R} \to \mathbb{R}$, which is non-linear.
Thus, we can mathematically define the a output, also known as activation as,

$$a = f(\sum_{i}^{n} w_i x_i + b), \tag{4.1}$$

where b is the bias term, which can be interpreted as the error of the model. In the left panel of Fig. 4-3 we present a cartoon of a biological neuron, while in the right panel, we show a sketch of its mathematical simplification related to Eq. 4.1.

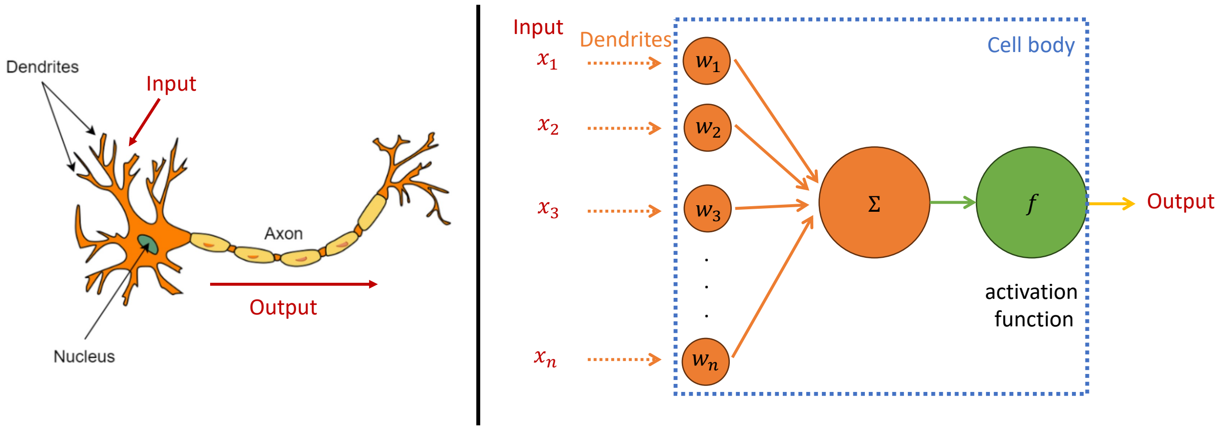


FIGURE 4-3: *(Left:) Cartoon of a biological neuron, the basic computational unit of the human brain. (Right:) Sketch of its mathematical model.*

We can then define a neural network as a stacking of neurons, where neurons are arranged in layers. Such a model is also known as *multi-layer perceptron* (MLP). Now, we consider a neural network of $n_l = 3$ layers, with hidden layers L_1 and L_2, and an output layer L_{n_l}. Each layer L_l has a matrix of weights $W_{ij}^{(l)}$ and vector of biases $b_i^{(l)}$, where the weights are the connection between the unit j of layer l and the unit i of layer $l+1$, and the bias is associated to the unit i of layer $l+1$. Then, the MLP can be mathematically described as,

$$a_1^{(2)} = f(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3 + b_1^{(1)}) \tag{4.2}$$

$$a_2^{(2)} = f(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{23}^{(1)} x_3 + b_2^{(1)}) \tag{4.3}$$

$$a_3^{(2)} = f(W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3 + b_3^{(1)}) \tag{4.4}$$

$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)} a_1^{(2)} + W_{12}^{(2)} a_2^{(2)} + W_{13}^{(2)} a_2^{(2)} + b_1^{(2)}). \tag{4.5}$$

Without initial consideration of assumption $n_l = 3$, the final output $h_{W,b}(x) = a_1^{(3)}$ is defined as the hypothesis or prediction of neural network. By defining $z_i^{l+1} = \sum_{j=1}^{2} W_{ij}^{(1)} x_j + b_l^l$, we can recast in $a_i^{(l)} = f(z_i^{(l)})$, and in turn, Eq. 4.2 – 4.4 in

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l)}, \qquad a^{(l+1)} = f(z^{(l+1)}) \tag{4.6}$$

Thus, by organizing the parameters as matrices, and using matrix-vector operations we can exploit the usage of efficient hardware, such as GPUs.

The firing of a neuron: the activation function

The activation function f is a crucial component of the neuron since it decides which neurons should be "activated" based on the input information, i.e. the weighted sum of its inputs. As the activation function introduces a non-linearity, this enables the neurons to learn complex patterns in the data.

The choice of activation function is determined by the nature of the input data, but some historically well-known activation functions are,

$$\text{sigmoid}\,(x_i) = \sigma(x_i) = \frac{1}{1+e^{-x_i}}, \tag{4.7}$$

$$\text{softmax}\,(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}, \tag{4.8}$$

$$\tanh\,(x_i) = \frac{2}{1+e^{-2x_i}} - 1. \tag{4.9}$$

Note that the sigmoid function, defined in range $[0, 1]$, is a particular case of the softmax activation function. Due to the definition of these three activation functions, large values of information within the neurons provide constant values which inhibit the information flow within the network. This implies that the activation function is "saturated" and it is commonly known as *vanishing gradients* [161]. To solve this issue we need a function that tends to infinity, such as the Rectified Linear Unit (ReLU) and similar,

$$\text{ReLU}\,(x_i) = \begin{cases} 0, & \text{if } x < 0. \\ x, & \text{if } \geq 0. \end{cases}, \qquad \text{Leaky ReLU}\,(x_i) = \begin{cases} \alpha x, & \text{if } x < 0. \\ x, & \text{if } x \geq 0. \end{cases} \tag{4.10}$$

In Fig. 4-4 we compare the sigmoid with ReLU activation function. As we can see, while for large input values the sigmoid function we get a constant value, either 0 or 1 if the input is positive or negative, ReLU does not only avoid this saturation, but it also favours positive input values, which lead to better behaviour of the information flow through the network. ReLU is widely used in the DL community, but it does not achieve the best performance for every application. Indeed, the activation function is an active field of research in DL, where scientists do not only create novel activation functions but combine current ones [162]. This is because a proper understanding of the activation function, and the flow of information through the network's architecture are key for the learning of neural networks.

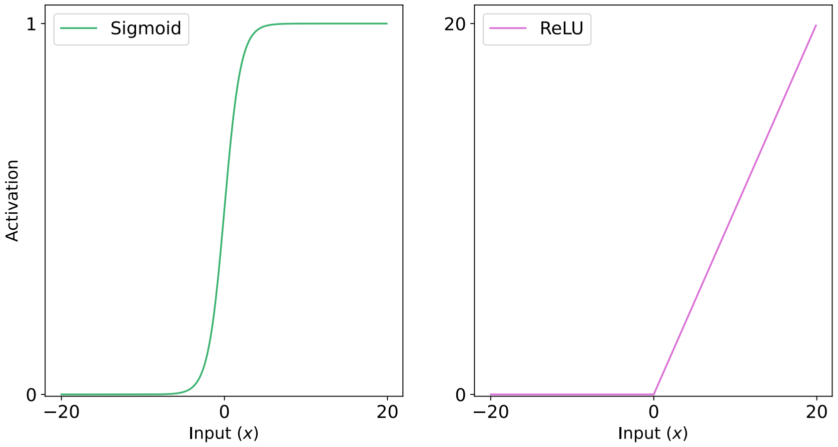


FIGURE 4-4: Comparison of sigmoid (left) and ReLU (right) activation functions. Note the difference in activation range.

The learning of a neuron: the loss function, gradient descent and backpropagation

We have talked about how we can design a neural network, but, despite mentioning it, we have not explained how the information flows within the model. Assuming that we have a supervised neural network that learns the fixed training set $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ of m training examples, we want to minimize the loss function,

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}|h_{W,b}(x) - y|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2. \tag{4.11}$$

The first term is an average sum-of-squares error between the ground-truth y and the prediction of the model $h_{W,b}$. The second term is the regularization, known as *weight decay*, controlled by the weight decay parameter λ. Note that s_l refers to the number of nodes of layer l.

The goal of the neural network is to find $W_{ij}^{(l)}$ and $b_i^{(l)}$ that minimize the loss function $J(W, b)$. Drawing an analogy, consider that we are at the top of a mountain with poor visibility due to atmospheric conditions. If our goal is to descend to the base swiftly, we naturally seek the quickest route – analogous to finding the path with the steepest negative slope from a mathematical perspective. This technique is called *gradient descent* [163], and neural networks are employed to find the optimal solution. Mathematically, one iteration of $(W_{ij}^{(l)}, b_i^{(l)})$ with gradient descent is,

$$W_{ij}^{(l)} := W_{ij}^{(l)} - \alpha\frac{\partial}{\partial W_{ij}^{(l)}}J(W,b) = W_{ij}^{(l)} - \alpha\left(\left[\frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial W_{ij}^{(l)}}J(W,b;x^{(i)},y^{(i)})\right] + \lambda W_{ij}^{(l)}\right) \tag{4.12}$$

$$b_i^{(l)} := b_i^{(l)} - \alpha\frac{\partial}{\partial b_i^{(l)}}J(W,b) = b_i^{(l)} - \frac{\alpha}{m}\sum_{i=1}^{m}\frac{\partial}{\partial b_i^{(l)}}J(W,b;x^{(i)},y^{(i)}), \tag{4.13}$$

where α is the learning rate, i.e. how wide is the step that we take to descend, which controls the learning process. It is important to note that at the beginning of the iteration $(W_{ij}^{(l)}, b_i^{(l)})$ is unknown, so we set them to be random values, rather than all zeros, so the layers of the neural network learn different functions [164]. Now we need an efficient way to compute our tuple (W_{ij}^l, b_i^l). An efficient method, widely used, is *backpropagation*, which computes the gradients through a recursive application of the chain rule.

The core of backpropagation is to first perform a "forward pass" to compute the activations throughout the network, and then to do a "backward pass" computing the amount on "error term" which measures the difference between the computed activation and the true target value. In detail, the backpropagation algorithm is:

1. "Forward pass" computing the activations of the layers
2. n_l being the output layer, and i each output unit,

$$\delta_i^{n_l} = \frac{\partial}{\partial z_i^{n_l}}\frac{1}{2}|y - h_{W,b}(x)|^2 = -(y_i - a_i^{n_l}) \cdot f'(z_i^{n_l}) \tag{4.14}$$

3. For $l = n_l - 1, \dots, 3, 2$, where i each node in layer l,

$$\delta_i^l = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)}\delta_j^{(l+1)}\right) f'(z_i^{(l)}) \tag{4.15}$$

4. Finally, we compute the desired partial derivatives,

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x, y) = a_j^{(l)} \delta_i^{(l+1)}, \qquad \frac{\partial}{\partial b_i^{(l)}} J(W, b; x, y) = \delta_i^{(l+1)} \tag{4.16}$$

We can redefine this process using matrix-vector multiplications and efficiently perform the computation with GPUs. Note that backpropagation is a special case of a technique called *automatic differentiation*, so for a deeper understanding we recommend the reader [165].

In the context of gradient descent, we define an *epoch* as a gradient descent update where the network has seen all the training data, so it updates its weights and biases accordingly. While for small data sets it is feasible to see them all at once in a single gradient descent update, this is prohibitive with larger data sets. To solve this issue, *stochastic gradient descent* [163] allows to perform gradient update in small subsets or *batches* randomly sampled. It is recommended for batches to be as big as the memory of the training device allows so that the neural network sees sufficient examples per gradient update. While this process alleviates computational complexity, it also introduces randomness in the training, which is beneficial for its convergence. Nonetheless, depending on the application it may be challenging to achieve convergence of the training process with these optimization algorithms, also known as *optimizers*, so other methods have been developed. We recommend referring to [163], and references therein for alternative optimizers. Note that we refer to pre-defined parameters of the network, such as the learning rate or the number of epochs, as *hyperparameters*, and we refer to the parameters learnt by the network, such as the weights, as simply *parameters*.

As in the case of the activation functions, the creation of novel optimization algorithms, loss functions, and neural network architectures, are active fields of research in the DL community. These choices are application-dependent, and their details are (in principle) always provided in DL papers for the sake of reproducibility.

4.2.2 Convolutional neural networks

MLP is composed of *learnable parameters*, such as weights and biases. A single layer of input n and a number of neurons m needs to learn $(n+1) \times m$ parameters. To understand how complex neural networks can be, we provide an example: we are given the task of classifying pictures of cats and dogs with a single-layer neural network of 50 neurons. The input images have a shape of (height, width, depth) $= 32 \times 32 \times 3$, where the depth is also known as the number of channels, and in this context, it represents the three colours that compose the images, namely, red, green, and blue. Now, to feed the images into the network we will have to flatten them to size $n = 32 \times 32 \times 3$. Therefore, the total number of *learnable* parameters of this neural network is

$$(n+1) \times m = (32 \times 32 \times 3 + 1) \times 50 = (3,072 + 1) \times 50 = 153,650.$$

This single-layered neural network is tremendously expensive, and it will suffer from the curse of dimensionality as we increase the resolution of the images. Moreover, these networks also break the spatial information of the input by flattening it, relating each pixel to every other pixel in the image. Indeed, in an image not all pixels are related, as it is not necessary to understand the pixels of the background to understand that a cat or a dog is present. Therefore, fully connected architectures, such as the MLP, learn irrelevant relations between pixels.

The revolution in DL came from the field of Signal Processing, where scientists had developed filters that extracted specific local characteristics of the image, such as the edges or textures. These filters are functions designed with domain knowledge and applied to the input signal using *convolutions*. Thus, the whole idea would be to *learn* them with neural networks.

In the context of image processing, an image is essentially defined as a matrix of values that correspond to its colors, and whose shape is $h \times w \times d$. Note that we are taking the convention of height × width × depth. Then, a filter, also known as *kernel*, is a squared matrix of shape $k \times k \times 1$, where $k \ll h, w$. The neurons forming the kernel are interconnected to a subset of size $k \times k \times 1$, also referred to as the *receptive field*. In this way, the kernel is simply looking at local features in this small subset of pixels, sharing the learnable parameters, known as *parameter sharing*. As before, assuming an input image of size $32 \times 32 \times 3$, and a convolutional layer of kernel 3×3, the number of parameters that it needs to learn is

$$(n + 1) \times k = (32 \times 32 \times 3 + 1) \times 3 = (3,072 + 1) \times 3 = 9,219.$$

While large kernels provide a wider receptive field, they are also more computationally intensive.

A way to increase the receptive field at a low computational cost is with *dilation* $\mathscr{d}$, where we "skip" $\mathscr{d}$ neighbours when convolving the input and the filter. As we are not only interested in a single part of the image, we need to slide the kernel through it with a given step size, defined as stride s. Moreover, sometimes the input can be too small, so we might need to increase its size by adding zeros around it, a process known as padding p. In Fig. 4-5 we show the sketch of two inputs being convolved by a pre-defined kernel.

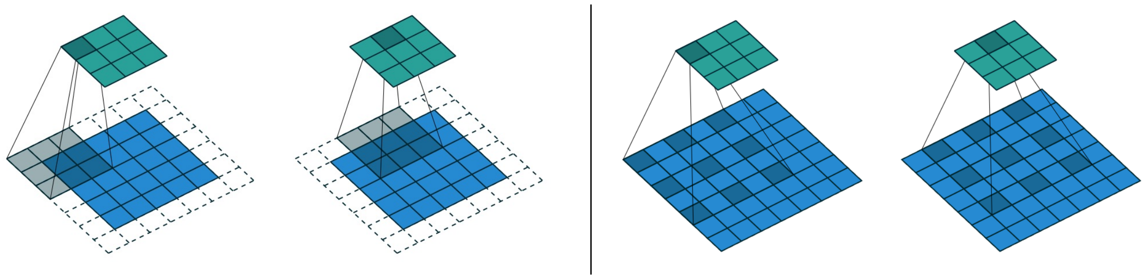


FIGURE 4-5: (Left:) Sketch of a kernel of size $k = 3$ over an input $6 \times 6 \times 1$, a dilation $\mathscr{d} = 1$, a stride $s = 2$ and a padding $p = 1$. (Right:) Sketch of a kernel of size $k = 3$ over an input $7 \times 7 \times 1$, a dilation $\mathscr{d} = 2$, a stride $s = 1$ and a padding $p = 0$, retrieved from [166].

On the left panel of Fig. 4-5 we can see that the input (blue), of original size $5 \times 5 \times 1$, was padded with a vector of zeros ($p = 1$), giving an input shape $6 \times 6 \times 1$. The kernel (green) is a squared matrix of size $3 \times 3 \times 1$, or $k = 3$, with a dilation $\mathscr{d} = 1$ and a stride $s = 2$. On the right panel of Fig. 4-5 we can see that the input (blue) has an original size $7 \times 7 \times 1$, and has no padding ($p = 0$). The kernel (green) is also a squared matrix of size $3 \times 3 \times 1$, or $k = 3$, with a stride $s = 1$, and a dilation $\mathscr{d} = 2$, which increases the receptive field by skipping neighbours.

We can think about the kernel to be a *feature map* that highlights and extracts the main features of the input to perform the task. Since the input will have more than a single relevant feature, we want to use f kernels per convolutional layer to build more expressive models. Given an input of size $h_{in} \times w_{in} \times d_{in}$ that we convolve with f kernels of sizes k, with dilation $\mathscr{d}$, stride s and padding p, we can calculate its output shape $h_{out} \times w_{out} \times d_{out}$ as,

$$\begin{pmatrix} h_{out} \\ w_{out} \\ d_{out} \end{pmatrix} = \begin{pmatrix} \left[\frac{h_{in}+2\times p-\mathscr{d}\times(k-1)-1}{s} \right] + 1 \\ \left[\frac{w_{in}+2\times p-\mathscr{d}\times(k-1)-1}{s} \right] + 1 \\ f \end{pmatrix} \tag{4.17}$$

If no padding is used, the width and the height of the images will always be $w_{out} < w_{in}$, $h_{out} < h_{in}$, but the output is conditioned by the number of kernels, or feature maps. To read a more detailed work about the different types of convolutions refer to [166].

The principal idea behind convolutional neural networks (CNN), and one of the reasons behind the success of DL, is the stacking of convolutional layers to form complex architectures. Indeed, each layer of the CNN looks at different patterns since they can *learn* different filters, depending on the information provided by the previous layers. Thus, these layers learn to recognize visual patterns by first extracting local features and subsequently combining them to obtain higher-level representations, also known as *latent* representations.

The shape of these architectures is application dependant, and the parameter space of *hyperparameters* to choose is gigantic: we do not only need to define the number of layers, the number of neurons per layer, and the way that we connect them but also there are other functions that we can add, such as *batch normalization* [167], *max-pooling* [166], *dropout* [149], among others. In this *hyperparameter* space, the challenge is not to find the *optimal* solution to our problem, but rather a solution that is *enough.* As we add neurons and layers, CNN grow in complexity, which in turn does not only bring the problem of intensive computation but the challenge of *interpretability.* Indeed, CNN, and other DL models, are complex and their inner workings opaque, so they have been granted the name of “black boxes”.

Nonetheless, great effort is being made to understand these models that are transforming our society. In the vibrant and fast-evolving field of DL, where the past is a couple of months old, scientists are arduously investigating how to interpret these models, challenging computer scientists, mathematicians, statisticians, and even, philosophers [168]. But as Albert Einstein said “if you want to know the future look at the past”, in the current work we will review some of the applications of ML and DL in the field of GW.

4.3 Review of gravitational-wave applications

In 2015, during the first observing run (O1), Advanced LIGO [5], and Advanced Virgo [6] successfully detected a BBH coalescence through the direct observation of GW [7]. After an upgrade of the detectors to increase their sensitivity, in November 2016, O2 started, followed by further significant upgrades, which initiated O3 in April 2019. While during O1 and O2, 11 candidates were observed, 74 candidates were detected over O3, and more candidates are expected in the fourth observing run (O4) [11, 12, 13]. In Fig. 4-6 we present the cumulative GW detections as a function of the observing days. Coloured areas correspond to observing runs.

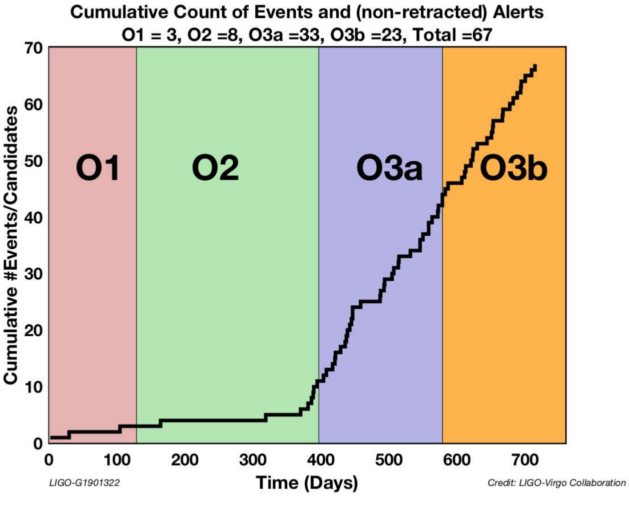


Figure 4-6: Cumulative count of gravitational wave events detected by LIGO and Virgo, split by observing run. Note that O3 has been divided between O3a and O3b. Credits to LIGO-Virgo-KAGRA.

As we have seen in Section 2.2, in the coming years, the sensitivity of the second generation of detectors will improve further. As Advanced LIGO, Advanced Virgo and KAGRA [33], approach their design sensitivity, and other interferometers, such as LIGO Aundha [34], are added into the network, the detection rate of GW observations will keep increasing. Moreover, with the construction of the third generation of detectors, such as LISA [14], Einstein Telescope [15] and Cosmic Explorer [16], the detection rate will grow steeply and GW signals will spend from hours to even days in their sensitivity band [169, 170].

Increasing the detection probability will increment the size of the search parameter space, and also add noise contribution caused by overlapping signals, known as *confusion noise* [171]. As a consequence, matched-filter search algorithms (see Section 3.2 for details) will become computationally unfeasible as the template banks grow in complexity, while the excess of power search algorithms (see Section 3.2 for details) will need more data to characterize the background and find GW signals. Parameter estimation algorithms will also be subjected to confusion noise biases [172, 173], and rapidly, current methods that rely on Markov Chain Monte Carlo (MCMC) [174] and nested sampling [175], will become computationally prohibitive, with the growing number of GW signals to analyze.

The challenges in GW research require innovative solutions, and ML has emerged as a crucial tool for addressing them. ML is an adaptive and transversal tool, with the ability to improve its performance over time as the available training data increases. This is an important characteristic within the field of GW, as the detectors are evolving machines where novel sources of noise and interference are bound to arise (see Section 2.4). Furthermore, the main advantage of ML techniques is their rapidity because most of the computations are made during the training stage. In the past few years, researchers have explored different ML applications to GW data analysis, such as feedback systems for the next generation of detectors, noise reduction, identification and modelling of non-gaussian transient burst noise, detection of modelled or unmodelled GW sources, GW signal extraction, and fast inference of intrinsic and extrinsic GW parameters.

In this review section, we will provide an overview of some of the key achievements of ML in the field of GW, but we recommend the reader to refer to [176, 177] for exhaustive reviews.

Detector characterization

In Section 2.4 we have explained different sources of noise that impact the sensitivity of the detector. The state of the detector is continuously monitored by thousands of auxiliary channels to witness the interferences caused by the main strain of the detector $h(t)$. Understanding these couplings is fundamental to mitigate persistent and transient noise that hinder GW searches.

`DeepClean` [178, 179] was built to associate auxiliary channels to predict non-linear persistent noise, using a similar idea to the linear prediction error (see Section 3.3.3), but using a fully-convolutional autoencoder. On another note, `iDQ` [180, 122] is an embedding of supervised ML classifiers that combine auxiliary channels and the main strain of the detector $h(t)$ to detect non-Gaussian transient burst noise in low latency. In the case of GW170817 [181], masked by an *Extremely_Loud* glitch in LIGO-Livingston, `iDQ` reported witness auxiliary channels in low-latency, enabling the subsequent sky-location of GW170817.

As we have seen in Section 2.4.3 glitch identification and characterization is a crucial first step towards their mitigation, but due to their overwhelming amount, their characterization by hand is unfeasible [63, 39, 64]. For this aim, researchers have built `Gravity Spy` [65, 67, 75, 68], to identify glitches in LIGO, and `GWitchHunters` [182, 70], to identify glitches in Virgo. Both algorithms combine supervised ML algorithms, in particular multi-class CNN, and citizen science to characterize glitches according to their morphologies in GW strain data $h(t)$in time-frequency representations.

However, this procedure presents several limitations. Supervised learning needs fixed class definitions that are not exhaustive nor representative of all glitch morphologies, as there could be many possible sub-classes to discover [67]. Furthermore, as GW detectors are improved, novel glitch morphologies could arise [75]. Moreover, generating these labels is an expensive task, since ML methods need a lot of examples for training, and experts must vet the labelling procedure. To overcome these limitations, authors in [183] proposed combining an unsupervised approach, providing results consistent with other supervised approaches.

Searches of GW sources

A fundamental part of modelled searches is the creation of accurate waveforms for the construction of template banks. Due to the increasing parameter space of GW searches, scientists have explored the usage of ML for faster generation of more complex and exotic CBC waveforms. For this aim a variety of ML methods have been implemented, such as Gaussian Processes [184, 185, 186], MLPs [187, 188, 189, 190, 191, 192], deep neural networks [193, 194, 195, 189], autoencoders [196, 197], transformers [198] and generative adversarial networks [199]. Moreover, we can also use ML to enable a fast sampling to construct template banks [105].

However, during the third generation of GW detectors, modelled searches that employ matched-filtering approaches will become computationally prohibitive, so researchers are also exploring ML to test the search hypothesis:

Null hypothesis (H_0): the time series $s(t)$ does not contain a GW $\rightarrow P(h|s) < P_*$

Alternative hypothesis (H_{a_0}): the time series $s(t)$ contains a GW $\rightarrow P(h|s) \geq P_*$

Indeed, several investigations tackle this problem to identify BBH [200, 201, 202] and BNS [203, 204, 205, 206] with CNN, relying on the fast prediction and generalization ability of these methods, just to name a few. These propositions intend to be an alternative to matched-filtering approaches, but they do not acquire the precision of current state-of-the-art modelled pipelines. However, ML does not need to substitute existing methods; instead, it can complement and even enhance their performance, fostering a synergistic relationship. With this idea in mind, authors in [207] translated matched-filtering to a multi-layer perceptron, and then optimized it via gradient descent, enhancing its initial performance.

In a similar line of thought, unmodelled searches that rely on the excess of power algorithms need to discriminate between glitches and Burst signals, and ML can also learn to differentiate between both classes. Indeed, that is the idea behind `XGBoost` [208, 209], a decision-tree based algorithm, and `GMM` [210], a Gaussian Mixture Model. These algorithms rely on the information provided by cWB and enhance its sensitivity towards GW signals. Moreover, ML algorithms can also learn to identify glitches based on the wavelet transform of cWB, improving the characterization of the background [211]. Conversely, several authors have created independent search algorithms with ML: some authors tackle targeted Burst searches of, for example, core-collapse-supernovae [212, 213, 214, 215], cosmic strings [216] or long burst [217, 218]; while other authors have focused on more generic searches [219, 220, 221].

Last, but not least, ML has also been used for searches of continuous waves [222, 223, 224, 225], and the stochastic background [226].

Parameter estimation

As we mentioned before, in the third-generation era, the number of GW signals to extract physics will be computationally unfeasible. For this aim, different authors have proposed to approach this problem with different ML algorithms. In the following, we present a non-exhaustive list of different applications for parameter estimation.

Some authors use standard CNN algorithms [227, 228], but variational autoencoders, such as `Vitamin` [229], seem to provide an enhanced performance. More recently, there has been a development of normalizing flow approaches that tackle the Bayes theorem [230] in different ways, such as `Dingo` [231, 232], that employs an autoregressive normalizing flow; `Nessai` [233, 234], which implements nested sampling with normalizing flows; the works in [235, 236], where the authors use a CNN to extract the main features of the data and provide this information to a normalizing flow; and the recent usage of truncated marginal neural ratio estimation [237] with `Peregrine`, for LIGO data applications, and `Saqqara` [238], for LISA data applications.

Part III

Towards the Next Generation of Detections

Chapter 5

From Searches to Machine Learning

In Chapter 4, we provided an overview of the building blocks of ML techniques. Nonetheless, due to their transversality and flexibility, ML applications tend to be problem-dependent, and users need to make educated choices, dependent not only on the task at hand but also on the particularities of the data. This inherent "know-how" of ML is not an exception in GW data analysis, where the particularities of the GW target sources (see Section 2.1), and the properties of detector noise (see Section 2.4) significantly influence the selection of statistical methods of search algorithms or "pipelines".

Therefore, in this chapter, we transition from the concept of a traditional GW search to an ML-based detection pipeline. We revisit the importance of data representation and provide an overview of the challenges ML faces in mitigating the background of glitches. Additionally, we situate this thesis within the broader field, highlighting advancements toward the next generation of GW searches.

5.1 A search pipeline as a classification task

As we described in Chapter 3, a search pipeline is a sequence of operations that inputs the raw data of the interferometer and produces a list of detection events or *triggers*, i.e. potential GW candidates, to solve the following hypothesis testing,

Null hypothesis (H_0): the time series $s(t)$ does not contain a GW $\rightarrow P(h|s) < P_*$

Alternative hypothesis (H_{a_0}): the time series $s(t)$ contains a GW $\rightarrow P(h|s) \geq P_*$

given a decision threshold P_*. While state-of-the-art search pipelines construct the likelihood ratio Λ (Eq. 3.3), we can build a supervised ML model that predicts the probability of the input data containing a GW signal based on past evidence, i.e. the model will learn the mapping $h : \mathcal{X} \rightarrow \mathcal{Y}$ between $\mathcal{X}$, the space of the detector's input X, and $\mathcal{Y}$, the space of the targets y (see Section 4.1). In ML such a supervised task can be defined as a classification problem, that can be *binary* or *multi-class*. If we perform a *binary* classification, we will distinguish between two classes: a *true* class, usually represented by $y = 1$, and a *negative class*, usually represented by $y = 0$. On the other hand, if we perform a *multi-class* classification, we will distinguish between C classes.

In the context of GW searches expressed as binary classification tasks, $y = 0$ implies that a GW signal is not present in the input data, while $y = 1$ implies that it is present. To learn such a task, the model minimizes the loss function known as *binary cross-entropy*,

$$\mathcal{L}(y,\hat{y}) = -[y\log\hat{y} + (1-y)\log(1-\hat{y})] = \begin{cases} -\log\hat{y}, & \text{if } y=1 \\ -\log(1-\hat{y}), & \text{if } y=0, \end{cases} \tag{5.1}$$

where $\hat{y}$ represents the probability predicted by the model that $y = 1$. We can generalize our binary classification to C classes by minimizing the *categorical cross-entropy* loss function,

$$\mathcal{L}(y,\hat{y}) = -\sum_{j}^{C} y_j \log\hat{y}_j, \tag{5.2}$$

where, traditionally, the true target y_j is represented as a one-hot vector. To interpret the output of the neural network as a probability $\hat{y}$, binary classification tasks use the sigmoid function, Eq. 4.7, while multi-class approaches use softmax, Eq. 4.8. These loss functions[1] have several desirable properties: they are easy to compute, differentiable and provide a probabilistic interpretation of the model's output [239]. However, they are sensitive to class imbalance, which occurs when the defined classes have significantly different sizes of samples.

In Section 3.1 we derived the equation of the likelihood ratio Λ (Eq. 3.3), and we noted that it is a monotonically increasing function of the posterior probability $P(h|s)$ of the GW signal h being present in the time series s. The main objective of the search pipelines is to use Λ as a statistical measurement, known as *ranking statistic*, to create a decision rule to detect GW signals. The fundamental difference between a state-of-the-art search algorithm and a ML-based algorithm is that this ranking statistic is not designed, but rather, *learnt*. Indeed, the prediction $\hat{y}$ of our ML model will be our ranking statistic, redefining, once again, the hypothesis testing of the GW search as,

Null hypothesis (H_0): the time series $s(t)$ does not contain a GW $\rightarrow \hat{y} < P_*$

Alternative hypothesis (H_{a_0}): the time series $s(t)$ contains a GW $\rightarrow \hat{y} \geq P_*$

TABLE 5.1: *Confusion matrix of GW + noise (positive class) against only noise (negative class)*

		Actual class	
		GW + noise	*Noise*
Predicted class	*GW + noise*	True positive (TP)	False positive (FP)
	Noise	False negative (FN)	True negative (TN)

Note that the decision threshold P_* is not dependent on the learning process of the ML algorithm, and it can be defined *a posteriori* to tune its performance.

Given this decision rule, there are two correct and two erroneous outcomes. If correctly we predict that a GW signal is or is not present in the time series s, we call this "true positive" (TP) or "true negative" (TN). Conversely, if we incorrectly predict that a GW signal is present in the data, we call this "false positive" (FP), or, most commonly in the GW field, *false alarm*. If we incorrectly predict that a GW signal is not present in the data, we call this "false negative" (FP), or *false dismissal*. In ML this information is commonly summarized in the *confusion matrix* (see Table 5.1).

From the confusion matrix, we can extract several metrics to evaluate the performance of ML algorithms, such as,

---

[1] They are particular functions of neural networks, and as such, they will be minimized via gradient descent, similarly to the example provided in Eq. 4.11. Nonetheless, other ML can be used to solve this classification problem, but they lie outside the scope of this work.

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} = \frac{\text{all correct classifications}}{\text{all classifications}} \quad \text{(accuracy)}, \tag{5.3}$$

$$TPR = \frac{TP}{TP + FN} = \frac{\text{GW signals correctly classified}}{\text{all GW signals}} \quad \text{(true positive rate)}, \tag{5.4}$$

$$FPR = \frac{FP}{TN + FP} = \frac{\text{misclassified noise}}{\text{all noise}} \quad \text{(false positive rate)}, \tag{5.5}$$

among others. In this ML context, FPR is the equivalent to the false alarm probability P_F (see Eq. 3.40). Nonetheless, it is relevant to note that the metrics derived from the confusion matrix nor the given probability $\hat{y}$ are not a measure of significance. Specifically, a comprehensive ML-based detection system must consider accidental backgrounds (see Section 3.4.2) to accurately compute the significance of predictions $\hat{y}$.

5.2 The importance of data representation

As we discussed in Chapter 4, ML is revolutionizing our contemporary world. One of the keys to ML success is not only the vast amount of information in the era of Big Data in the form of images, text, videos and audio, but also the construction of tailored ML algorithms to process these large data sets. A clear example is the design of CNN (see Section 4.2.2), capable of encoding images of flamingos or pelicans in latent representations that little have to do with our understanding of what a flamingo or a pelican *is* (see Fig. 5-1).

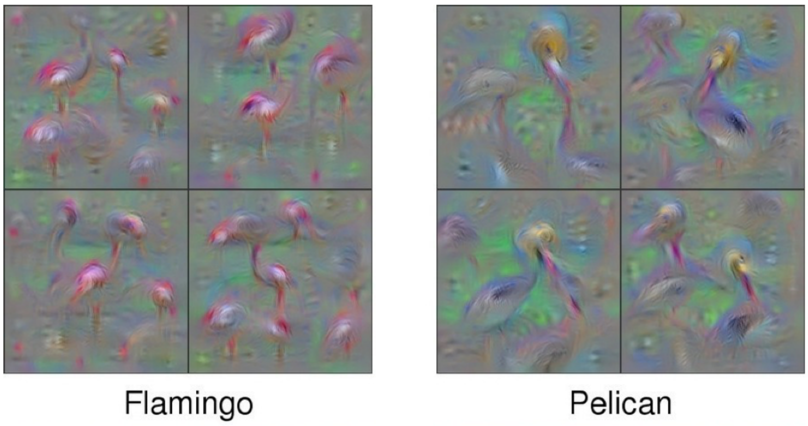


FIGURE 5-1: Flamingos and pelicans as seen by a ML model, retrieved from [240].

Due to the strong dependency of ML methods on the quality and quantity of data, it is essential to carefully consider the representation of input data. In practical applications, especially in domains sensitive to accuracy such as precise physics experiments, this becomes even more crucial, as the data must provide sufficient information to accurately represent natural phenomena. In the following, we revisit some of the data representations used in ML applications for GW searches.

- **Time series:** GW detectors produce continuous time series data, making time-domain analysis straightforward with minimal post-processing beyond noise reduction methods. This approach is particularly useful for CBC signals, which are well-localized in time within the sensitivity band of ground-based detectors, and have a linearly increasing frequency. By limiting the frequency range of interest, this analysis can effectively highlight these signals. However, interpreting time series data is challenging, especially in the presence of a complicated background of glitches with complex frequency contents.

- **Time-frequency:** This representation provides a comprehensive view of how the signal's frequency content evolves. These methods are particularly effective for analyzing non-stationary GW signals, whose frequency components change dynamically. However, they are computationally intensive, requiring additional post-processing to achieve an optimal representation and balancing the trade-off between time and frequency resolution. Nonetheless, most DL methods are designed to extract features from 2-dimensional representations, such as CNN, facilitating better integration of available models within the ML community.

- **Feature vectors from state-of-the-art pipelines:** Over the past two decades, current GW search pipelines have been fine-tuned to an exquisite level of accuracy to detect GW signals buried within the detector data. These pipelines process GW data by extracting meaningful features that have been extensively studied and validated within the GW research community. These well-defined features can be integrated with cutting-edge ML models to enhance the ability to distinguish between GW signals and glitches, and potentially extrapolate beyond current theoretical frameworks. However, strong domain knowledge is required to correctly interpret the results of this combined approach.

Having explored the various data representations crucial for effective ML applications, it is evident that distinguishing transient GW signals from a background of glitches that can mimic or obscure true GW signals (see Section 2.4.3) remains a significant challenge. In the following section, we will explore how current transient GW searches mitigate glitches and how ML can enhance these efforts.

5.3 Mitigation of a glitch background

A fundamental utility of a network of interferometers is to find coincident time triggers across independent experiments. This capability is crucial not only for locating GW signals in the sky but also for minimizing the background of glitches that hinder GW searches, as glitches are less likely to occur simultaneously in different detectors. Indeed, this principle is employed by GW searches through incoherent or coherent analysis methods: incoherent analysis generates single triggers and searches for time coincidences at a later stage, while coherent analysis generates a unique list of triggers by combining coincident detector data (see Section 3.3.1).

ML algorithms, particularly CNN, can significantly enhance both incoherent and coherent analysis methods in GW) searches. In incoherent analysis, the speed and generalization capabilities of ML methods enable rapid and efficient inference, which can then be used in subsequent steps to identify coincidences across different detectors, reducing background noise from glitches. This approach is especially beneficial for low-latency scenarios, where swift detection within seconds is crucial for initiating timely electromagnetic follow-ups.

In incoherent analysis, ML algorithms can learn to identify GW signals based on time coincidences between different detectors. CNN, in particular, can process multiple data streams like handling different colour channels in images (see Section 4.2.2). As more ground-based detectors become operational, additional data streams can be incorporated, potentially enhancing further the detection capabilities and improving the accuracy of GW searches.

A common advantage of both analysis methods is that ML algorithms can directly learn information about background noise, particularly glitches, which helps with their mitigation. However, as mentioned in Section 5.1, class imbalance—where current true GW signals are vastly outnumbered by background noise and glitches—can bias our algorithms, leading to higher rates of FP and FN. Due to a poor minimization of the loss function, some algorithms might become biased towards classifying signals as positive detections (“yes-classification”), drastically increasing the number of FP. Therefore, it is essential to carefully consider class balance and optimize loss functions to enhance the accuracy and reliability of GW detections. Furthermore, when dealing with real data, there is a risk of unknowingly contaminating the dataset with unidentified glitches, which further complicates the detection process.

Throughout this work, we address these issues using different approaches: we investigate various data representations to enhance the performance of the algorithms, explore advanced loss penalization techniques to ensure stable learning and overall robustness, and implement different methods to tackle class imbalance. These comprehensive approaches collectively contribute to overcoming the challenges posed in this and previous sections. As we refine these methods, we lay the groundwork for the next generation of GW searches.

5.4 Towards the next-generation of searches

At the end of their lives, massive stars $\sim 8 - 100 M_{\odot}$ have accumulated ≈ 1.4 M$_{\odot}$ of elements of the iron family in a compact core due to thermonuclear fusion processes. Then, the iron core cannot support its own weight and experiences a gravitational collapse. The most common core-collapse mechanism is known as the neutrino-driven mechanism [241, 242], and is expected to be responsible for the majority ($> 99\%$) of all core-collapse supernovae (CCSNe) explosions. The basic theory of CCSN explosion is consistent with SN 1987 A, which was detected via electromagnetic radiation and low energy neutrino emission [243]. Nonetheless, GW and neutrino emission, as opposed to electromagnetic radiation, provide direct and unique information about the inner dynamics of the collapse, since they are produced at its inner core.

As stellar evolution predicts, CCSN can produce remnants, either a neutron star or a black hole, but progenitors within $\gtrsim 50 - 130 M_{\odot}$ cannot produce a remnant due to pair-instability supernovae [244, 245]. Nonetheless, in nature, we also observe supermassive black holes (SMBH), such as Sagittarius A* in the Milky Way Galaxy [246, 247]. A plausible explanation for their formation is the hierarchical mergers of intermediate-mass black holes (IMBH) [248, 17]. The direct observation of IMBH populations via GW would not only strengthen the possible evolutionary link between stellar and SMBH, but unveil the details of the pair-instability mechanism and elucidate their influence in galaxy formation. Conclusive observation of IMBH remained elusive until the detection of gravitational-wave signal GW190521, which lies with high confidence in the mass gap predicted by the pair-instability mechanism.

Despite falling in the sensitivity band of second-generation detectors, modelled searches of IMBH and unmodelled searches of CCSN are hampered by the presence of non-gaussian transient burst noise, known as glitches (see Section 2.4.3). In particular, *Blip* glitches are detrimental to these searches due to their short duration (~ 0.04 s), shared frequency range ($\sim [30, 250]$ Hz) and abundance within the detector. In this thesis, we enhance the searches of these astronomical events with three different neural-network-based strategies:

Chapter 6. Simulating transient noise bursts: glitches impact GW searches since they can mimic and mask GW signals, lowering the confidence of potential candidates and biasing astrophysical parameter estimation. Because of this, there is a need for better modelling and inclusion of glitches in large-scale studies. In particular, in this work we use a DL algorithm to learn the underlying distribution of *Blip* glitches from O2 of LIGO, demonstrating that this approach can be extended to other glitch morphologies and detectors.

Chapter 7. Detection of anomalous transient noise bursts: current approaches to glitch identification use supervised models to learn their morphology in the main strain with a fixed set of classes. This type of approach has several limitations: supervised learning needs a large amount of human intervention to provide class labels; the different class definitions are not flexible nor exhaustive, as several sub-classes of glitches could exist;

and the main strain is not a witness of the physical process that generates the glitch. A challenging but interesting possibility is to encode the relevant information provided by auxiliary channels that monitor the state of the interferometers utilizing an unsupervised ML algorithm to find anomalous glitches. In this way, we learn the underlying distribution of glitches and we uncover unknown glitch morphologies, and overlaps in time between different glitches and misclassifications.

Chapter 8. Enhancing unmodelled CCSN searches: Although CCSN are among the most energetic processes in the universe, their GW signal is expected to be extremely faint. Moreover, due to the complexity and stochasticity of the waveform, generating CCSN waveforms from Numerical Relativity is challenging and computationally intensive. Thus, our primary goal in this work is twofold: on one hand, we generate an inexpensive set of phenomenological waveforms that mimic the monotonic rise of the CCSN GW signal in the time-frequency representation. On the other hand, we built a CNN to learn this monotonic rise training and validating on the phenomenological waveforms, and testing on CCSN GW signals simulated with Numerical Relativity to probe its interpolation ability.

Chapter 9: Enhancing modelled IMBH searches: Current modelled searches are matched-filtering based. Templates that closely match the unknown data generate a trigger. Under the assumption that IMBH signals and glitches have triggered different templates over time, we exploit this information to distinguish them with DL. In this way, DL provides a complementary statistic to enhance the current searches of IMBH.

Chapter 6

Simulating Transient Noise Bursts

This chapter is based on work presented in Ref. [249] and its companion, Ref. [250], where I led part of the pre-processing of the data and the construction of the model. I also led the post-processing, performance evaluation, and writing of both manuscripts, as well as the subsequent tests for review in LIGO-Virgo-KAGRA collaboration.

6.1 Introduction

While current GW search techniques for transient signals ($\lesssim 1$ minute) have been extremely successful, their sensitivity is still limited by transient glitches (see Section 2.4.3). Despite considerable efforts to mitigate the impact of glitches on GW searches [39, 40], they remain one of the major limiting factors in the detection and parameter estimation of transient GW signals. Moreover, glitches also bias ML applications in the field of GW, as their true population in real data is unknown. This necessitates better modelling and inclusion of glitches in large-scale studies, such as stress-testing pipelines.

In this chapter, we address this need by learning the underlying distribution of glitches using ML, specifically employing a generative ML method called generative adversarial networks (GAN) [251]. Additionally, we provide several examples of applications and make our findings accessible to the broader scientific community through our Python package, `gengli`.

6.2 Data

6.2.1 Identification and classification

Due to the challenges posed by glitches, such as reducing the amount of analyzable data, biasing astrophysical detection, affecting parameter estimation, and even mimicking GW signals (see Section 2.4.3 for an overview), it is essential to develop robust techniques to identify and characterize these noise sources for their possible elimination.

In previous LIGO and Virgo science runs, this classification was performed by visual inspection, which soon proved to be slow and inefficient [64]. During O2 run, the detection rate of glitches was $\approx 1\ \mathrm{min}^{-1}$; so due to the overwhelming amount of glitches present in data, identifying them by hand was unfeasible. A promising option is to construct ML algorithms to identify and classify glitches [252, 64, 253]. However, another challenge arises since a pre-labeled data set is necessary to train such algorithms.

With this goal in mind, Zevin et al. [65] developed pioneer work to classify transient noise, called `Gravity Spy`. In this work, both problems are addressed: volunteers provide large labelled data sets to train the ML algorithms through Zooniverse infrastructure, while ML algorithms learn to classify the rest of the glitches correctly, providing feedback to participants. In practice, a glitch time series that we wish to classify is fed to the algorithm that generates the Q-transform of its input (see [65] for details). Then, `Gravity Spy` classifier assigns a class and a confidence value c_{GS} to the Q-transform of the glitch, where c_{GS} represents the confidence of the label assigned. `Gravity Spy` uses a multi-class classification, and it differentiates between 23 glitch classes and the absence of glitch inside the Q-scan, known as *No_Glitch* class, in O2 [65]. This type of approach has some limitations that have been briefly discussed in Section 2.4.3.

6.2.2 Blip glitches

This work focuses on *Blip* glitches, introduced in Section 2.4.3, due to their abundance during O2 run and their simple morphology. *Blip* glitches hinder both the unmodelled Burst and modelled CBC searches [55, 52], with particular emphasis in compact binaries with large total mass, highly asymmetric component masses, and spins anti-aligned with the orbital angular momentum. For illustration, in Fig. 6-1, we see the similarities between a *Blip* an IMBH in O2 noise. Moreover, since there is no clear correlation to the auxiliary channels, they cannot be removed from astrophysical searches yet.

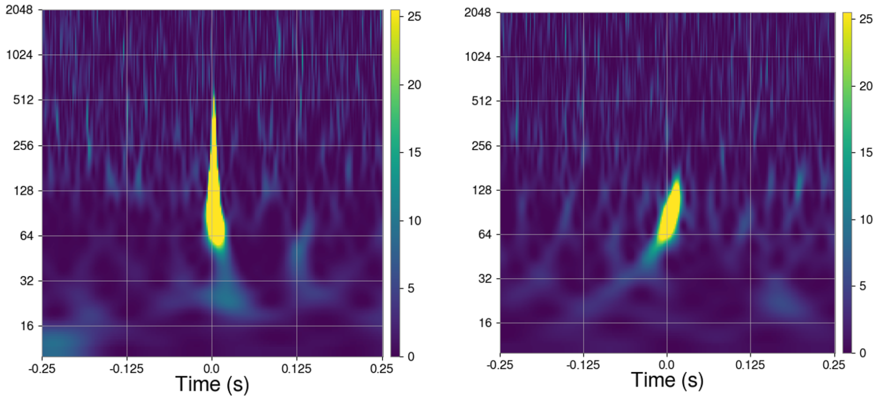


Figure 6-1: Time-frequency reperesentation of a Blip glitch retrieved from **`Gravity Spy`** *[65] (left) and a GW signal with total mass* $106.6^{+13.5}_{-14.8} M_\odot$ *(right).*

6.2.3 Pre-processing

The construction of our data set strongly relies on the confidence provided by `Gravity Spy`. Thus, to create a high confidence data set, we select the *Blip* glitches from LIGO-Livingston (L1) and LIGO-Hanford (H1) detectors of O2[1] run that have a confidence $c^1_{GS} \geq 0.9$.

Table 6.1: Size of the Blip set for each detector in the different phases of the pre-processing: selection, reconstruction and evaluation.

Pre-processing	Livingston	Hanford
Num. *Blips* $c^1_{GS} \geq 0.9$	5540	6768
BW output	5461	5612
Num. *Blips* $c^2_{GS} \geq 0.9$	3654	3407
Num. *Blips* $c^2_{GS}, c^3_{GS} \geq 0.9$	3291	2587

Glitches are surrounded by stationary and uncorrelated noise, which will hinder the learning of our ML method. Therefore, it is necessary to extract glitches from the stream data maintaining their original morphology. For this aim, we employ `BayesLine` [46] to whiten the glitches locally and `BayesWave` (BW) [135] to extract the

---

[1] Data from GWOSC https://www.gw-openscience.org/data/

glitches from the uncorrelated noise. BW uses non-orthogonal continuous Morlet-Gabot wavelets (see Section 3.3.2) to fit and reconstruct the input signal, but the selection of the set of Morlet-Gabot wavelets is made with a trans-dimensional Reversible Jump MCMC [254] that acquires a trade-off between the complexity of the model and the quality of the fit. The input signal is represented as a set of wavelets whose reconstruction is their addition.

In our particular framework, the input provided to BW is a time series containing the *Blip* glitch that is 2.0 s long. However, to avoid training the CT-GAN algorithm in irrelevant data and speed up the training phase, the samples of the final training set have 938 data points sampled at 4096 Hz, constituting 0.23 s of data.

Since the reconstruction is not perfect, we lose around 2% and 18% of the data for L1 and H1, respectively (see Table 6.1). To assess the quality of the reconstructions, we inject them in real whitened noise and evaluate it with `Gravity Spy` classifier, selecting *Blips* with a $c^2_{GS} \geq 0.9$ to generate high-quality input data. After this heavy pre-processing, the training data set is composed of around 66% and 50% of the initial data for L1 and H1, respectively.

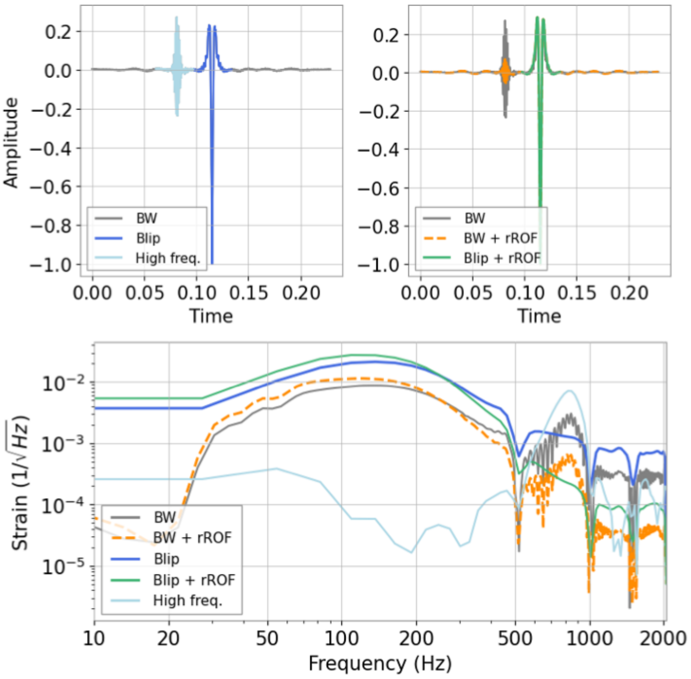


FIGURE 6-2: *(Top left) Blip glitch reconstructed with BW (grey), where we colour the characteristic Blip peak (blue) and the undesired high-frequency contribution (light blue). (Top right) Blip glitch reconstructed with BW (grey) and denoised with* $\lambda = 0.5$ *(dashed orange). We colour in green the denoised characteristic Blip peak. (Bottom) Resulting ASD for the reconstructed Blip with BW (grey) and its denoised version with* $\lambda = 0.5$ *(dashed orange). We also show the ASD of the characteristic peak with (blue) and without denoising (green), as well as the high-frequency contribution (light blue).*

Moreover, as mentioned in Section 2.4.3, *Blips* can be found in the frequency band [30, 250] Hz, but BW might introduce certain high-frequency contributions that will hinder the learning of our ML algorithm. For illustration, in Fig. 6-2 (left) we plot BW reconstruction (grey), where we coloured the characteristic *Blip* peak (blue) and the high-frequency contribution (light blue). To eliminate the high-frequency contribution, we initially set an empirical threshold to remove power excess in the surroundings of the peak. Nonetheless, some high-frequency contributions overlap with the *Blip* and cannot be removed with this method. Thus, to minimize this contribution and generate a smoother input to enhance the learning of our model, we employ regularized Rudin-Osher-Fatemi (rROF) proposed in [255].

This algorithm solves the denoising problem, $s = g + n$, where g is the smooth reconstruction of glitch and n is the noise, as a variational problem. The solution g is computed as follows:

$$g_\lambda = \arg\min_g \{\mathcal{R}(g) + \frac{\lambda}{2}\mathcal{F}(g)\}\,, \qquad (6.1)$$

where $\mathcal{R}(g)$ is the regularization term that constrains the data, which refers to the quality of the smooth reconstruction g. $\mathcal{F}(g)$ is the fidelity term, which measures the L^2-distance between the g

glitch and the observed signal s. λ regularises and controls the relative weight of both terms in the equation. It is important to note that this parameter needs to be tuned manually to achieve the desired level of denoising.

To assess the quality of the denoised *Blip* glitches, we use the `Gravity Spy` classifier for different λ parameters again, and we found $\lambda = 0.5$ to be a trade-off between preserving the structure of the glitch and removing the non-smooth high-frequency contribution. In Fig. 6-2 (right), we plot the BW reconstruction denoised with rROF (dashed orange), and the denoised characteristic *Blip* (green). In Fig. 6-2 (bottom), we show the ASD of the BW reconstruction with and without denoising (grey and dashed orange), as well as the characteristic peak with and without denoising (blue and green) and the original high-frequency contribution (light blue). We can observe that we maintain the structure of the characteristic peak by damping the power of the high-frequency contribution.

To verify that we can preserve the structure of *Blips* according to the current state-of-the-art, we compare in Fig. 6-3 the `Gravity Spy` confidence of reconstructed *Blips* c^2_{GS} (blue), against denoised reconstructed *Blips* c^3_{GS} (orange) from L1. As we can observe, both distributions are similar since they have similar means μ_{GS} and standard deviations σ_{GS}. Finally, we select the *Blip* glitches with $c^2_{GS} \geq 0.9$ and $c^3_{GS} \geq 0.9$, to ensure the high quality of the input data of the algorithm. Due to this heavy post-processing only $\sim$ 50% and $\sim$ 60% of the initial data from H1 and L1 are preserved, respectively

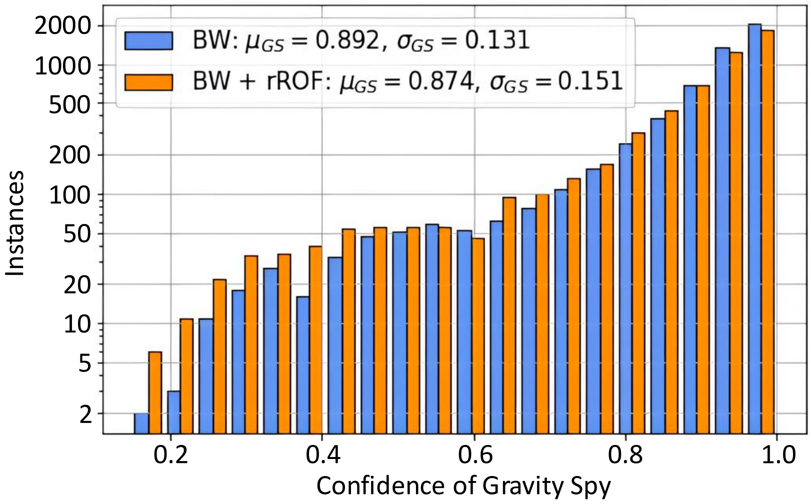


Figure 6-3: Comparison between the reconstructed and the denoised population of Blip glitches for L1. For the reconstructed set $c^2_{GS} = 0.892 \pm 0.003$ and for the denoised set $c^3_{GS} = 0.874 \pm 0.004$ at 95% confidence level.

6.3 Methodology

ML methods are not only limited to pattern recognition tasks. GAN can learn the underlying distribution of a population to produce artificial examples from Gaussian noise. With this idea in mind, the authors in [256] employed a conditional GAN to burst signals, allowing them to generate multiple classes of signals with the same algorithm and to interpolate through different classes, creating mixed signals. The powerful generation capability of GAN suggests that they can be applied to generate artificial glitches, as demonstrated in this work Indeed, this is an interesting avenue, as it is fundamental to properly understand the background of glitches to assess the performance of GW search algorithms with minimal biases. Hence, this work aims to learn the underlying distributions of glitches with GANs. In the following subsection, we provide more details about GAN methodology and the architecture of our network.

6.3.1 Generative adversarial networks

GAN [251] are a class of generative algorithms in which two neural networks, typically convolutional neural networks (CNNs), compete with each other to achieve realistic image generation. One network, known as the *generator*, is responsible for generating new images from random noise, while the other, known as the *discriminator*, tries to discriminate the generated images from the real training data. The generator progressively learns which features of the real images should be mimicked to fool the discriminator and save them into the latent space, which can be understood as a compressed representation of the input data learnt by the generator (see discussion in Section 4.2.2).

At the end of the training, new images are drawn by randomly taking a latent space vector and passing it to the generator, which has learned to translate it into a realistic image. Fig. 6-4 shows an overview of the original architecture of GAN for generating 2-dimensional data, but all the forthcoming developments still hold for 1-dimensional data. This early approach has been shown to work well under some hyperparameter configurations [258]. However, early GAN architecture [251] suffers from the significant problems of vanishing gradients (see Section 4.2.1) and meaningless loss function [259] where the minimization of the loss does not translate into improved performance.

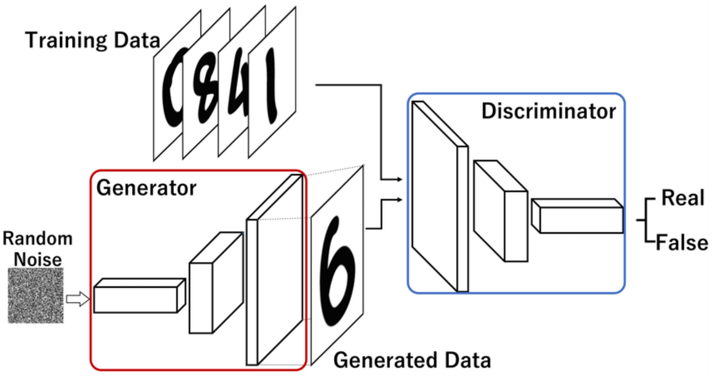


Figure 6-4: Typical GAN architecture retrieved from [257].

Wasserstein GANs [260] (WGAN) were developed to address these issues by making use of the Earth's mover distance estimator[2], or Wasserstein-1 distance (W_1) [261], which computes the similarities between two distributions. Mathematically, we can define the Wasserstein distance between the real distribution P_x and the generated distribution $P_{\tilde{x}}$ as

$$W_1(P_x, P_{\tilde{x}}) = \inf_{\gamma \in \Pi(P_x, P_{\tilde{x}})} E_{(x,y)\sim\gamma}[||x - y||] \tag{6.2}$$

where inf refers to the infimum, a set's greatest lower bound, and $\Pi(P_x, P_{\tilde{x}})$ is the set of all joint distributions $\gamma(x, y)$. $E_{(x,y)\sim\gamma}[||x - y||]$ is the expected value of the distance between x, a sample from the real distribution P_x, and y, a sample from the generated distribution $P_{\tilde{x}}$.

W_1 is evaluated through the discriminator as the training progresses and increases monotonically while never saturating, providing a meaningful loss metric even for two disjoint distributions. Since W_1 is continuous and differentiable, it yields reliable gradients, allowing us to train the discriminator till optimality to obtain high-quality generations. This change of paradigm led Arjovsky et al. [260] to reformulate the optimization problem as

$$\theta_{opt} = \arg\min_{\theta} W_1(P_x, P_{\tilde{x}}), \tag{6.3}$$

---

[2] Earth Mover's Distance metaphorically refers to the idea of shifting piles of "dirt" from one distribution to another, minimizing the total "cost" required for the transformation.

where W_1 is evaluated between the real distribution P_x and generated distribution $P_{\tilde{x}}$. Eq. 6.3 can be written as

$$\theta_{opt} = \arg\min_{\theta} \max_{\phi:\|D(x,\phi)\|_L \leq 1} L(\phi,\theta) \tag{6.4}$$

with the discriminator loss

$$L(\phi,\theta) = -E_{x\sim P_x}\Big[D(x,\phi)\Big] + E_{\tilde{x}\sim P_{\tilde{x}}}\Big[D(\tilde{x},\phi)\Big], \quad \text{where } \tilde{x} = G(z,\theta) \tag{6.5}$$

where z is a batch of the generator's la- tent vector. D and G refer to the discriminator and the generator with parameters ϕ and θ, respectively. $E_{x\sim P_x}$ indicates that the expression has been averaged over a batch of real samples x, while $E_{\tilde{x}\sim P_{\tilde{x}}}$ has been averaged over a batch of generated samples $\tilde{x}$. The new condition over ϕ in expression Eq. 6.4 imposes a constraint on the discriminator D, which must be 1-Lipschitz continuous [260]. This property limits the values of the discriminator's D gradient, resulting in a less aggressive discriminator. However, it avoids null gradients that could destabilize the learning process of the generator G. By ensuring the discriminator remains 1-Lipschitz continuous, we achieve a balance where the discriminator is effective but does not impede the generator's training through excessively flat gradients.

In practice, this can be achieved in two ways: clipping the weights of the discriminator beyond a specific value c [260], or adding a regularization term to the discriminator loss, defined in Eq. 6.5, known as gradient penalty (GP). While the first solution is a poor way to enforce the Lipschitz condition, the second solution has been widely accepted. The mathematical formulation of GP is as follows:

$$L_{tot} = L(\phi,\theta) + \lambda\, GP(\phi) \quad \text{with} \quad GP(\phi) = E_{\hat{x}\sim P_{\tilde{x}}}\left[\Big(\|\nabla_x D(\hat{x},\phi)\|_2 - 1\Big)^2\right], \tag{6.6}$$

where λ is known as the regularization parameter, $\|\cdot\|_2$ stands to the L^2-norm and $\hat{x}$ is evaluated following

$$\hat{x} = \tilde{x}\,t + x\,(1-t) \tag{6.7}$$

with t uniformly sampled $\sim [0,1]$. This method has shown impressive applications such as [262], but it is not restricted to WGANs [263, 264]. Nonetheless, unlike weight clipping, GP cannot enforce the Lipschitz condition everywhere, particularly at the beginning of the training. This can prevent the generator from converging to the optimal solution. To overcome this obstacle, Wei et al. [265] have proposed a second penalization term to add to the loss from Eq. 6.5, called consistency term (CT). They applied their new constraint to two perturbed versions of the real samples x, introducing dropout layers into the discriminator architecture. This ultimately leads to two different estimates noted $D(x',\phi)$ and $D(x'',\phi)$. CT is defined as follows

$$\begin{aligned} CT(\phi) = E_{x\sim P_x}\Big[max\Big(0,\, d(D(x',\phi), D(x'',\phi)) \\ + 0.1\, d(D\_(x',\phi), D\_(x'',\phi)) - M'\Big)\Big], \end{aligned} \tag{6.8}$$

where d(.,.) is the L^2 metric, $D\_(.,\phi)$ stands for the second-to-last layer output of the discriminator, and M' is a constant value. Wei et al. found that controlling the second-to-last layer output helps improve the performance of the WGANs. Thus, the final discriminator loss is then [265]:

$$L_{\text{tot}} = L(\phi,\theta) + \lambda_1\, GP(\phi) + \lambda_2\, CT(\phi)\,, \tag{6.9}$$

with λ_2 being the consistency parameter. This type of WGAN was called CT-GAN, which is the one that we employ in this work. Note that while the work of Weit et al. [265] was for 2-dimensional inputs, we have adapted the network to 1-dimensional inputs, as we use time series to achieve a flexible modelling.

6.3.2 Network architecture

The architecture of the networks has been inspired by the work presented in [258] but nearest-neighbour (NN) sampling layers have been preferred over strided convolution layers in the generator structure. The convolution parameters were chosen to be fixed through the generator and discriminator layers with kernel $k = 5$, no padding and stride $s = 1$. Leaky ReLU$(\cdot, \alpha = 0.2)$ has been chosen as the activation layer for both discriminator and generator, except for the output layer of the generator, which uses a $\tanh(\cdot)$ activation, allowing values $\sim [-1, 1]$, (see Section 4.2.1 for details).

In the generator structure (see Fig. 6-5), we also employ a dilation factor of 2, 4, 6, 8 and 16 for successive layers to enlarge its receptive field and, in turn, its expressivity power, at the exact computational cost [162]. Batch normalization [167] has been added to the generator architecture to make it both stable and faster to learn. The discriminator structure (see Fig. 6-6) is composed of convolutions on which spectral normalization [266] is employed to stabilize the training. Dropout layers are added, excluding the first and last layers, which is required by the consistency term (Eq. 6.8). A brief overview is provided in Section 4.2.2 and references therein.

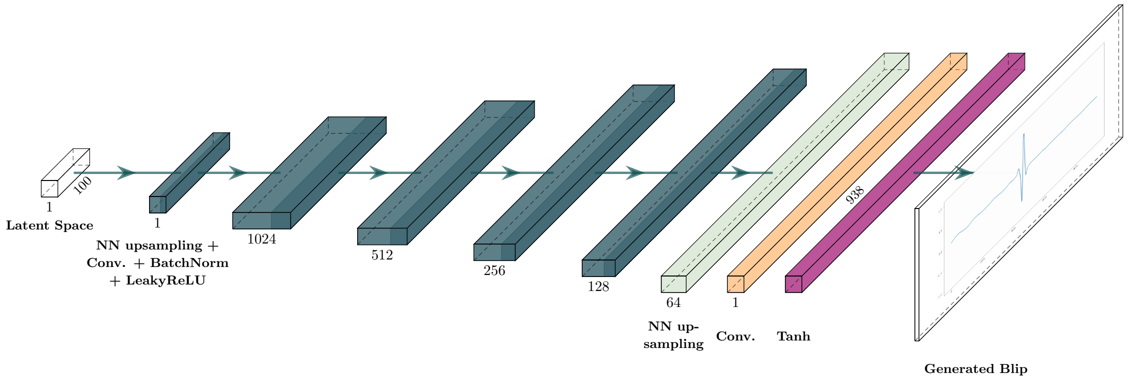


FIGURE 6-5: *Generator structure including NN upsampling, convolution layers and LeakyReLU activation.*

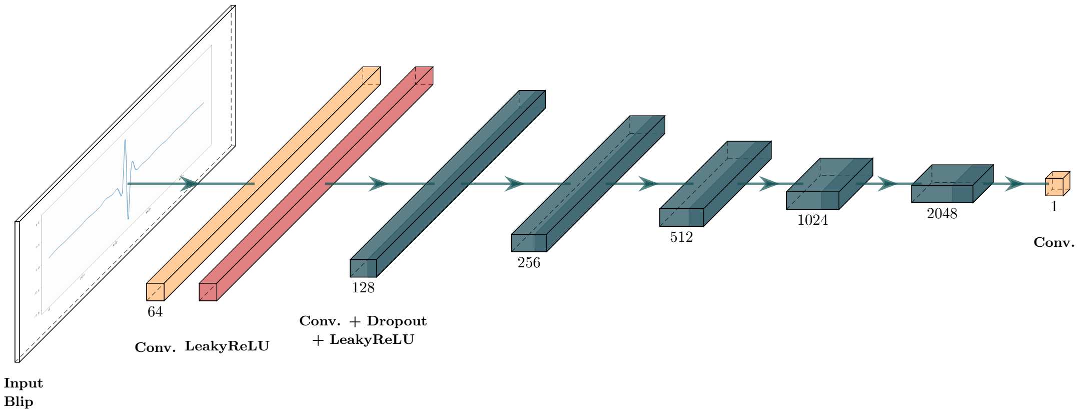


FIGURE 6-6: *Discriminator architecture showing strided convolutions, dropout layers and LeakyReLU activations.*

6.3.3 CT-GAN training procedure

During the training of the CT-GAN algorithm, both the generator and the discriminator need to be updated at similar rates to acquire stability and guarantee convergence. The task of the discriminator is more difficult since the generated samples that the discriminator intends to classify can be anywhere in the data space, and change for each new iteration [267]. Hence, to assure the stability of both networks, we update the discriminator 5 times per update of the generator, for each epoch. We employ RMSProp optimizer [268] with a learning rate $= 10^{-4}$ for both discriminator and generator, and we train the CT-GAN for 500 epochs, where we again define an epoch as the number of times the network has passed through the whole dataset. A GPU TITAN V with a memory of 96 Gb allowed us to train our model in ≈ 7.75 h.

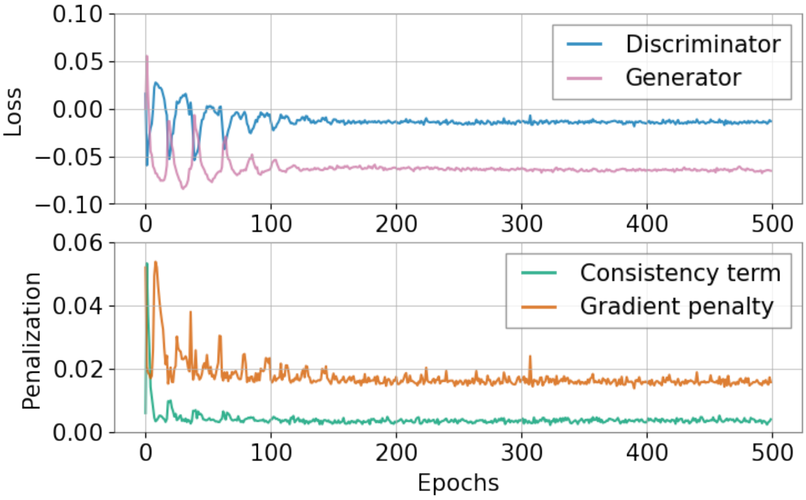


FIGURE 6-7: Graph representing the discriminator loss (blue), generator loss (pink), CT (green) and GP (orange) penalisation as a function of the epochs.

To monitor the behaviour of the CT-GAN during the training phase, we represent the generator and the discriminator loss as a function of the epochs in Fig. 6-7. We can observe that both networks stabilize around epoch 100 and continue to oscillate around values close to zero until the training is complete. After several experiments, we concluded that while CT regularised the generator, dropout regularised the discriminator and GP balanced both. This stability can also be observed in the behaviour of the CT and GP penalizations in Fig. 6-7, where both terms tend to zero as the network stabilizes. The values that helped the CT-GAN to converge were CT $= 5$ and GP $= 5$, with a dropout rate of 0.6. These values were obtained after several experiments, but in future works, it would be interesting to employ Optuna [269], which is a hyperparameter optimization framework used to automate hyperparameter searches.

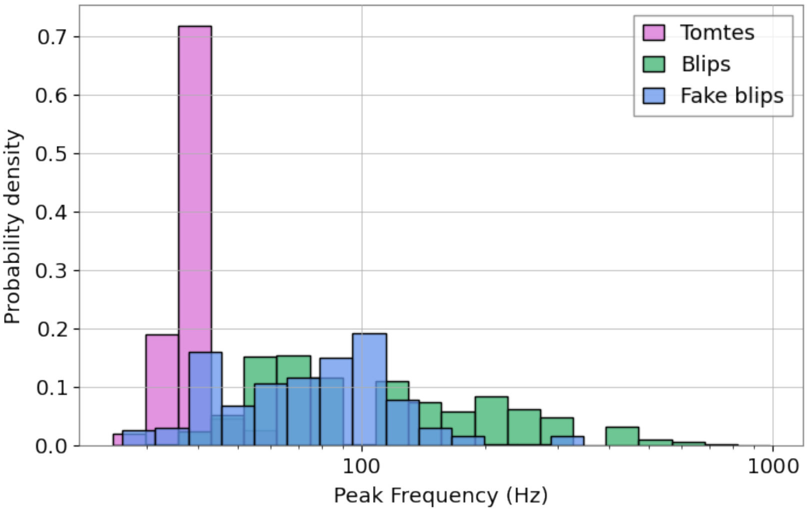


*FIGURE 6-8: (Top) Peak frequency for Tomte (pink) and Blip (green) from L1 retrieved from **`Gravity Spy`**[65], measured with Omicron spectrograms [73]. In blue we plot the peak frequency of the artificial Blips from L1.*

6.4 Results

6.4.1 Blip generation

After the training of the CT-GAN, and given a 100-dimensional vector drawn from a normally distributed latent space (as is common in other GAN-related works), we can generate 10^3 *Blips* from the input distribution of H1 and L1 in ≈ 5 s for both interferometers. It is relevant to note that each *Blip* has a length ≈ 0.23 s with an amplitude $\in [-1, 1]$, whitened and sampled at 4096 Hz. In Fig. 6-8, we compare the peak frequencies of real *Tomte* and *Blip* glitches from L1 against our artificial population, where we can observe that the bulk of the distribution of fake *Blips* is aligned with the real *Blip* population. As an example, we present in Fig. 6-9 different artificial *Blips* from L1 in the time domain, and for visualization, we also compute their time-frequency representation as in [65]. In the time-frequency representation, we can see that CT-GAN has been able to capture the distinct symmetric 'teardrop' of *Blips* in the expected frequency range $[30, 250]$ Hz. Furthermore, we can observe that in the time representation, we can reproduce different morphologies of the characteristic central peak. Even if by visual inspection it would seem that the artificial generations are closely related to the real *Blips* from O2, it is necessary to perform a statistical test to assess the performance of CT-GAN.

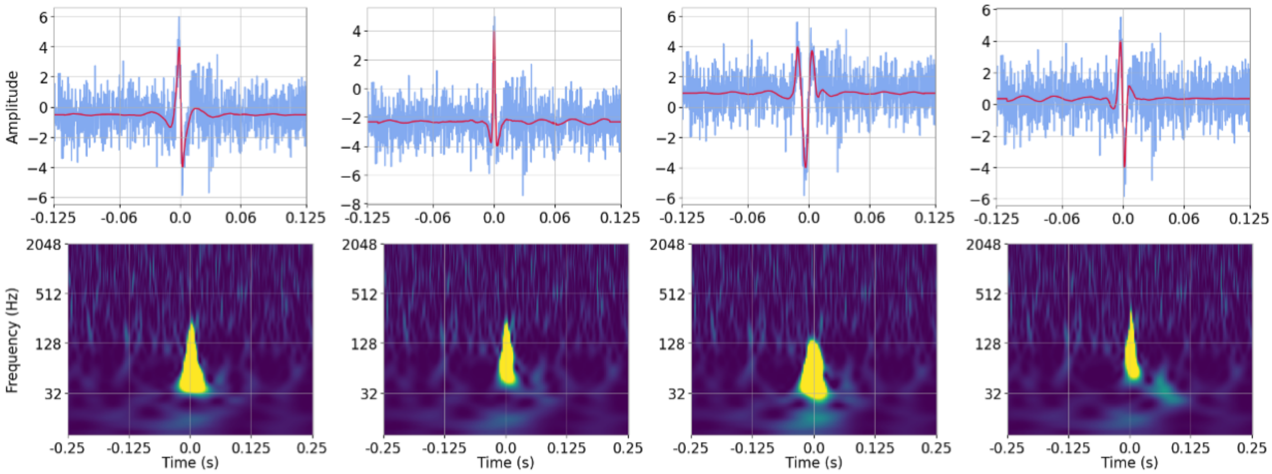


Figure 6-9: (Top row) Generated Blip of L1 plotted as a function of time. In red, we represent the rescaled whitened Blip and in blue we plot its injection, both in the time domain. (Bottom row) We show the time-frequency representation of the generated injected glitches.

6.4.2 Assessing performance

We employ four different methods to assess the quality of the population. On the one hand, we employ their Q-scan representation to evaluate our artificial population with the current state-of-the-art. On the other hand, we analyze their morphology in the time domain to take into account the phase information:

- **Gravity Spy classifier:** To assess performance using an independent ML classifier, we can inject the generated glitches in real whitened noise from O2 (see Fig. 6-9) and evaluate them with `Gravity Spy`, which will return a confidence value c_{GS} and a class label. We use the same noise strain for each generated glitch to provide the classifier with a fair comparison. Since the generated *Blip* has an amplitude $\in [-1, 1]$, we can re-scale it according to a desired optimal signal-to-noise ratio (ρ_{opt}). For this aim, we relate ρ_{opt} to the scaling parameter α by modifying the Wiener scalar product from Eq 3.9 as

$$\rho = (g|g)\Big|_{f_{min}}^{f_{max}} = 4\int_{f_{min}}^{f_{max}} \frac{|\hat{g}(f)|}{S_n(f)'}df \tag{6.10}$$

where $\tilde{g}(f)$ represents the Fourier transform of artificial *Blip* $g(t)$, and S_n is the power spectral density (PSD) of the fixed real whitened noise. With this expression we can scale the artificial glitch to achieve the given ρ_{target} using: $g' = \rho_{target} g/\rho$. One of the main drawbacks of this method is that it is computationally intensive ($\approx$ 90 s/glitch) because it is necessary to calculate the time frequency of the input time series for different time windows.

- **Wasserstein distance** (W_1): As explained in Section 6.3.1, the Wasserstein distance is continuous and never saturating, allowing us to keep track of the quality of the generated samples during the training. For further mathematical details, a formal definition can be found in [260]. This metric is then an adequate tool to compare real and generated glitches. This method is fast and efficient since the computation is performed in the time domain (≈ 0.0026 s/glitch).

- **Match function** (M_f)**:** To compute the similarity between two signals, we can also use the match function, which returns the match between both signals [270]. As we have seen in Eq. 3.18, the match can be defined as the inner product between two normalized signals $\hat{a}$ and $\hat{b}$ maximized over time (t) and phase (ϕ) [109],

$$M_f(a,b) := \max_{t,\phi} \langle \hat{a}, \hat{b} \rangle. \tag{6.11}$$

  Since the signals are noise-free, we do not employ any PSD for normalization. This calculation is performed in the frequency domain, and it is also fast and efficient ($\approx$ 0.0032 s/glitch).

- **Normalized cross-covariance** ($k(X,Y)$)**:** Assuming two random processes X and Y, their cross-covariance between time t_1 and t_2 is defined as

$$K_{X,Y}(t_1,t_2) \equiv E[(X_{t_1} - \mu(X_{t_1}))\overline{(Y_{t_1} - \mu(Y_{t_1}))}] \tag{6.12}$$

  To obtain the normalized cross-covariance coefficient, we divide the cross-covariance over the standard deviation of each random process. The maximum value of this magnitude is the metric employed to measure the similarity between two signals, as defined below

$$k = \max\Big(\frac{K_{X,Y}(t_1,t_2)}{\sigma_X \sigma_Y}\Big) \tag{6.13}$$

  This calculation, which is also in the time domain, is most efficient (≈ 0.0011 s/glitch).

`Gravity Spy`

For this procedure, we inject each generated *Blip* in real whitened detector noise and re-scale it according to Eq. 6.10 to fix ρ_{opt}. We can compute the confidence of `Gravity Spy` as a function of the optimal SNR $\rho_{opt} \in [0.1, 18.2]$. This process is conducted on 10^3 *Blip* glitches of each detector population.

In Fig. 6-10, we plot the classification labels, with maximum classification probability, for different ρ_{opt} of H1 population, while we present the results of L1 in Appendix A.4. We can observe that the dominant class is *Blip* and that the number of glitches in this class increments

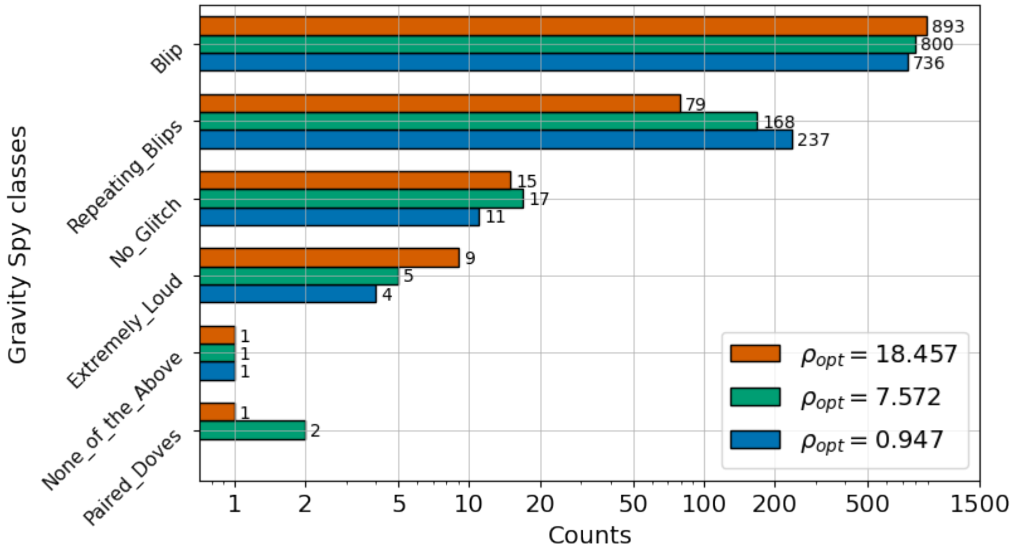


FIGURE 6-10: *Histogram of predicted* **`Gravity Spy`** *classes for* 10^3 *generated Blips from H1.*

by increasing ρ_{opt}, in opposition to other classes. Interestingly, when $\rho_{opt} = 0.947$, meaning that the artificial *Blip* is not visible by eye in the time-frequency representation, around 500 artificial glitches are labelled as *Blip*.

One could think that this type of behaviour would be expected since CNNs can "see" patterns that are invisible to the human eye, but the classifier can recognize glitches up to a certain threshold (Omicron SNR ≥ 7.5 [73]). Another reason might be that the training set of `Gravity Spy` is imbalanced, so the classifier is biased towards the larger classes such as *Blip*. Hence, it seems that `Gravity Spy` has a certain degree of miss-classification, so we employ other metrics to test the performance of our CT-GAN.

Wasserstein distance, match function and normalised cross-covariance at testing

To measure the performance of the network, we use some alternative methods, namely Wasserstein distance (W_1), match function (M_f), and normalized cross-covariance (k). These metrics are employed to calculate the similarity between two different artificial *Blips* b_1 and b_2, but we can also use them to calculate the similarity between a single artificial *Blip* b_F and the real population (B_R) or the artificial population (B_F) from each detector. Such procedure is as follows:

1. We use a certain similarity distance m to measure the distance between *Blip* b_j and a population B.
2. For each *Blip* $b_i \in B$ we compute $m_{j,i}(b_j, b_i)$, which yields a set of measurements M_j.
3. We obtain the mean and the standard error of the previous set as $\mu(M_j) \pm \epsilon(M_j)$ at 99.7% confidence interval.

The latter is the measure of similarity between the population B and b_j. Note that the numerical meaning of Wasserstein distance, match function, and normalized cross-covariance are different. For the previous example,

- If b_j is a reliable generation then $W_1(B, b_j) \approx 0$, while $M_f(B, b_j) \approx 1$ and $k(B, b_j) \approx 1$.
- If b_j is an anomalous generation then $W_1(B, b_j) \gg 0$, while $M_f(B, b_j) \ll 1$ and $k(B, b_j) \ll 1$.

Since we are dealing with real data, the real population B_R contains not only *Blips* but also certain misclassifications. If the CT-GAN had learned the underlying distribution of the data, we would expect that the real population B_R and the artificial population B_F had a similar distribution, where reliable generations would be located in the bulk of the distribution. In contrast, anomalous *Blips* would be located in the tails. Hence, under this assumption, we would expect that, given a metric m, the similarity distance between the real and artificial distribution $m(B_R, B_F)$, should be linearly related to the similarity distance of the artificial distribution against itself, $m(B_R, B_F)$.

In Fig. 6-11, we plot the joint and marginal distribution of both comparisons for different similarity distances and show the results from the least-squares estimate for each detector. Furthermore, in Table 6.2 we present the Pearson coefficient resulting from the least-squares estimate, which represents the linear correlation between both variables [271].

Table 6.2: Pearson coefficient for different metrics and detectors.

	Livingston	Hanford
Wasserstein distance	0.993	0.999
Match function	0.999	0.999
Normalised cross-covariance	0.996	0.999

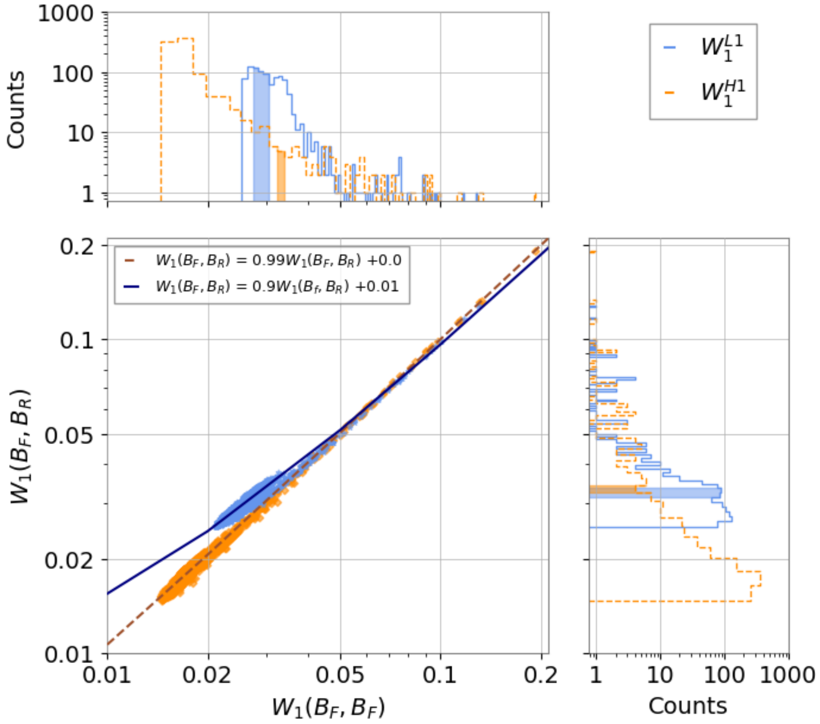
We observe that the resulting slopes (Fig. 6-11) and the Pearson coefficients (Table 6.2) for each metric and each detector are close to 1.0, meaning that both variables have a very strong linear relationship and compatibility. Thus, all similarity distances indicate that the bulk of the population is constituted by reliable blips, with the presence of some anomalous generations that can be identified by fixing an empirical threshold. Therefore, since the generated *Blips* represent the artificial and real populations, we conclude that the CT-GAN has learned the underlying distribution of *Blips* from L1 and H1.

6.4.3 Assessing poor generations

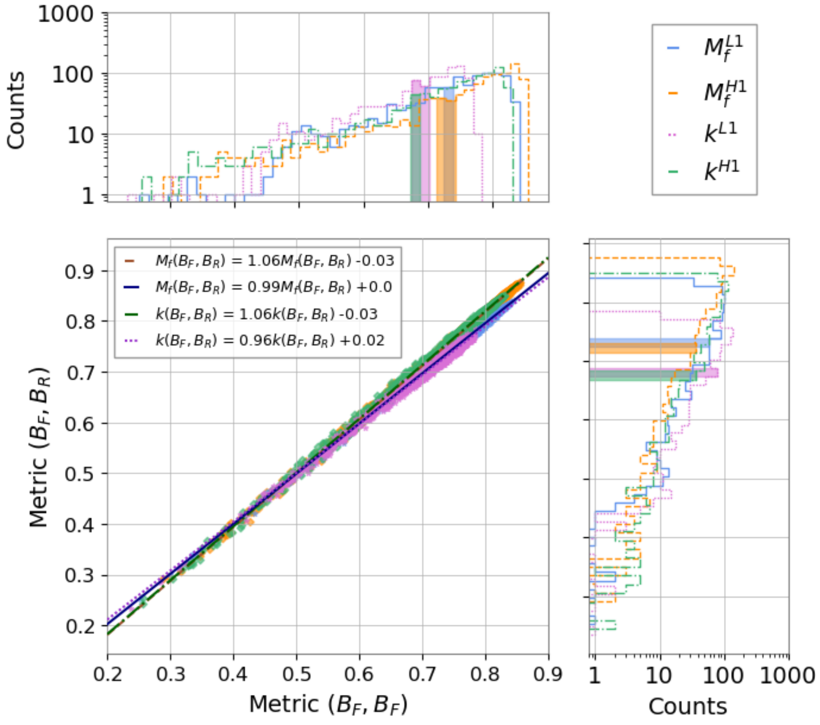
When dealing with real data, one must bear in mind that certain anomalies might be present in the data. In our particular context, our data sets might contain glitches that have a distinct morphology from the mean of the population. Such differences might not be visible in a time-frequency representation, so `Gravity Spy` might introduce certain miss-classifications that contaminate the input dataset. Since CT-GAN can learn the underlying distribution, it can also generate non-*Blip* glitches that are in the tails of the distribution. For certain studies, the presence of anomalies might be counterproductive, so differentiating reliable from anomalous generations is crucial. For this aim, we propose several metrics to identify these miss-generations.

To use `Gravity Spy` classifier, we inject the generated *Blips* in real whitened noise with a fixed optimal SNR $\rho_{opt} = 18.46$, according to Eq. 6.10. From the classification, we select the generated *Blips* that belong to the three dominant classes: *Blip*, *Repeating_Blips*, and *No_Glitch*.

In Fig. 6-12 we plot the joint and marginal distribution as probability densities of `Gravity Spy` confidence against the alternative metrics for H1 (see Appendix A.4 for details about L1). We can observe that according to `Gravity Spy` *Blip*, *Repeating_Blips*, and *No_Glitch* seem to belong to distinct probability densities. However, according to the alternative metrics, the probability densities remain centred according to a certain value for different classes. Furthermore, there seems to be no correlation between `Gravity Spy` confidence and other metrics in the joint distribution, so to further understand our results, we proceed to inspect the results by selecting examples,

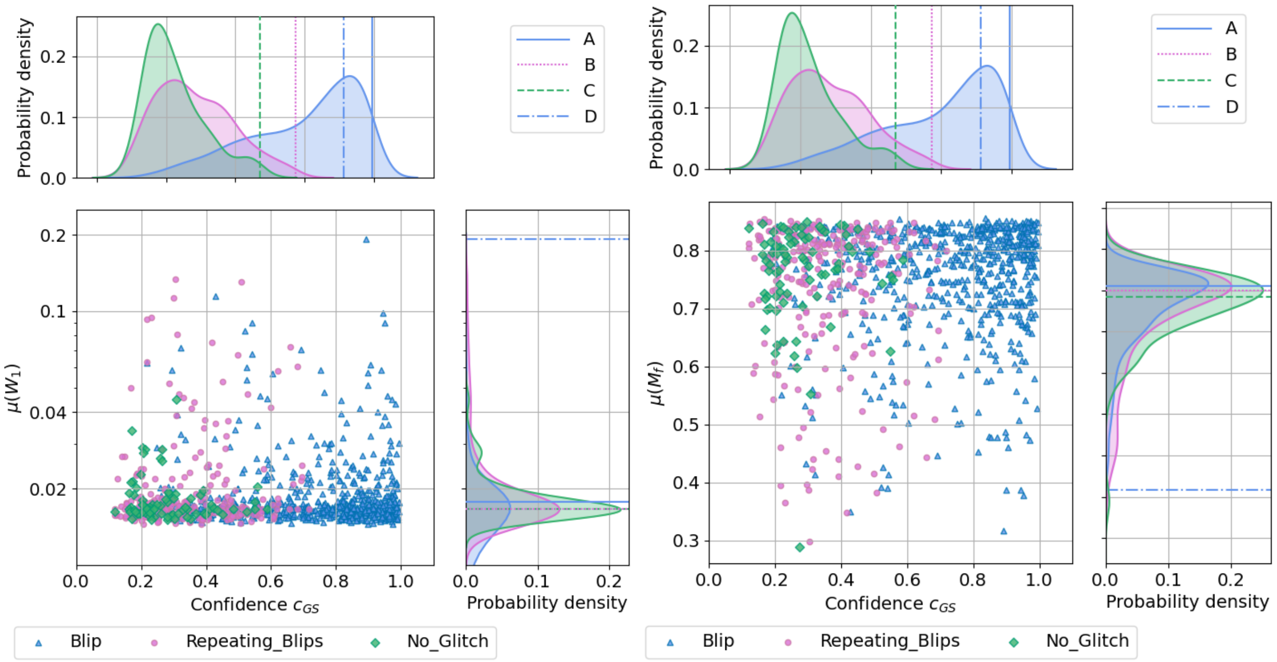


(A) Similarity distance: Wasserstein distance (W_1)

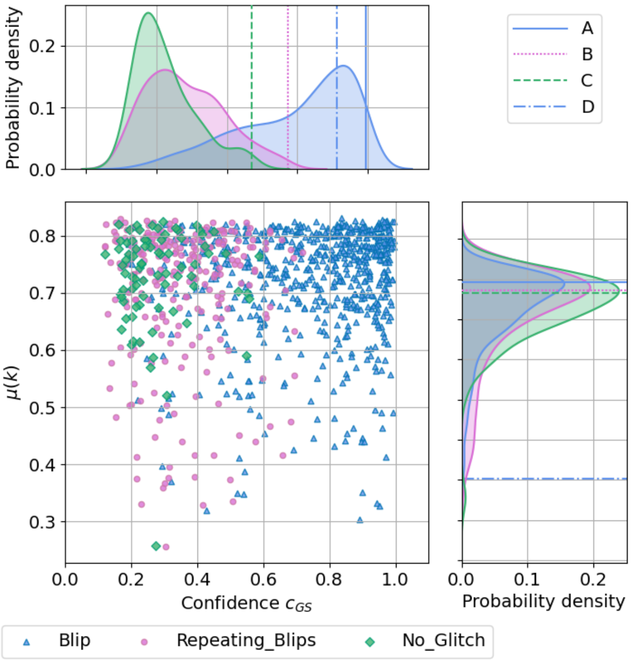


(B) Similarity distance: match function (M_f) and normalised cross-covariance (k)

Figure 6-11: (Top) We represent the joint and marginal distributions of $W_1(B_R, B_F)$ *and* $W_1(B_F, B_F)$ *in logarithmic scale for L1 (blue) and H1 (orange) and their best fit. (Bottom) We represent the joint and marginal distributions of the pairs* $[M_f(B_R, B_F), M_f(B_F, B_F)]$ *and* $[k(B_R, B_F), k(B_F, B_F)]$ *for L1 (blue and pink) and H1 (orange and green), as well as their best fit. The coloured regions in the marginal distributions represent the confidence interval at 6 standard deviations.*



(A) Wasserstein distance (W_1) (B) Match function (M_f)

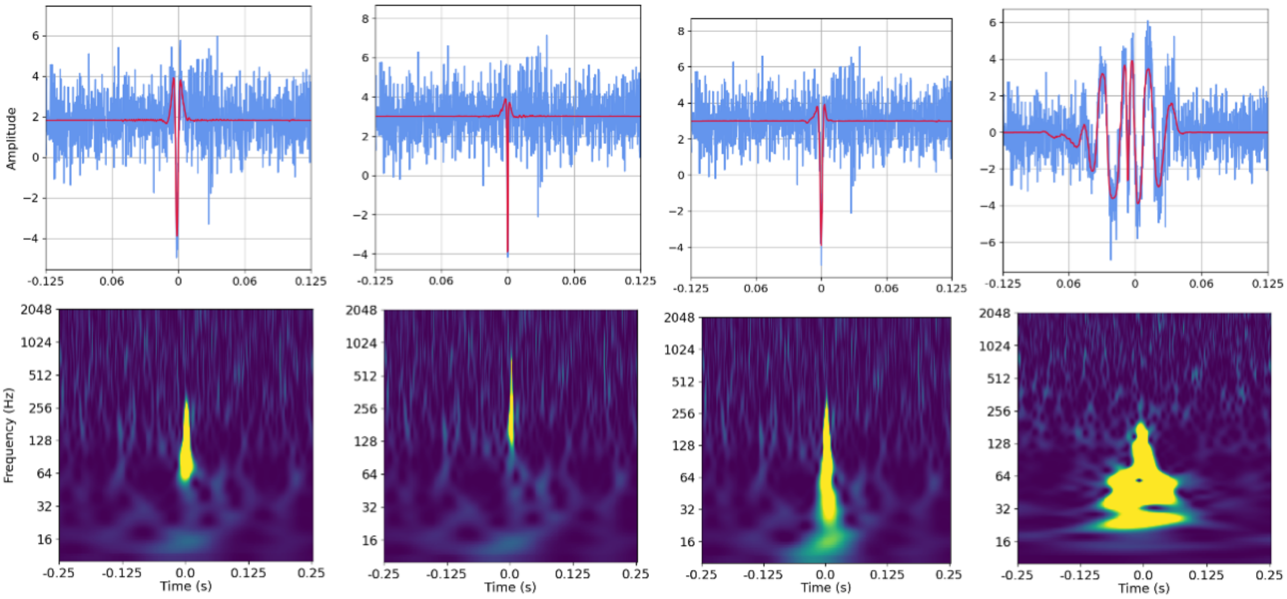


(C) Match function (M_f)

FIGURE 6-12: *Joint and marginal distribution of* `Gravity Spy` *confidence* c_{GS} *at* $\rho_{opt} = 18.46$ *against different metrics for different glitch classes for H1: Blip, Repeating_Blips and No_Glitch. We mark in the marginal distributions selected generated glitches A (solid blue), B (dotted pink), C (dashed green) and D (dash-dotted blue).*

- **Glitch A:** This glitch is labeled as a *Blip* with a high confidence according to `Gravity Spy` ($c_{GS} \approx 0.99$). Furthermore, the chosen metric has situated this glitch in the bulk of the distribution, meaning that it is a reliable *Blip* generation.
- **Glitch B:** This glitch is labeled as a *Repeating_Blips* with a confidence $c_{GS} \approx 0.72$. However, according to our metrics, it is a reliable generation.
- **Glitch C:** This glitch is labeled as a *No_Glitch* with a confidence $c_{GS} \approx 0.59$. However, according to our metrics, it is also a reliable generation.
- **Glitch D:** This glitch is labeled as a *Blip* with a high confidence according to `Gravity Spy` ($c_{GS} \approx 0.89$). Nonetheless, the chosen metric has situated this glitch in the tail of the distribution, meaning that it is an anomalous *Blip* generation.

In Fig. 6-12, we can observe that according to the alternative metrics, glitches A, B and C are situated around the centre of the probability density, while glitch D is located in the tails. Moreover, for further visualization in Fig. 6-13, we present the selected in the time domain, and we also plot their time-frequency representations. We can observe that while glitches A, B, and C seem to have a similar shape and magnitude, they differ from anomalous glitch D. Moreover, with these metrics, we are able to identify anomalous generations that deceive `Gravity Spy` classifier, and their exclusion from the generated data set can be performed by imposing a threshold.



(A) Glitch A (B) Glitch B (C) Glitch C (D) Glitch D

Figure 6-13: Time series representation (top row) and Q-scan representation of selected glitches from H1

6.4.4 Limitations

The main shortcoming that we encountered when training the CT-GAN was the limited amount of data preserved after the heavy pre-processing. CT-GAN needs a large number of samples to learn the underlying distribution, which might be a limitation when extending our methodology to other classes of glitches that are less common in the LIGO-Virgo streams. Nonetheless, some researchers are developing techniques to tackle this limitation that we will explore in future works [272].

Another relevant shortcoming of this study is the fact that the quality of our input data set strongly relies on BW reconstruction and `Gravity Spy` classification. In our particular case, *Blip* glitches have a simple morphology, but some undesired contributions were introduced by BW and some miss-classifications were introduced by Gravity Spy. Other glitches might be even harder to extract and/or classify with the current state-of-the-art due to their complex form, which in turn will hinder the performance of our CT-GAN. Moreover, longer and more complex glitches will need better architectures to be able to learn the underlying distribution of the data.

6.4.5 Applications

In the following, we provide examples of possible applications that can be explored in future works:

A. *Glitch population statistics:* Learning the distributions of glitches allows us to understand their populations further and compare their different characteristics. In this way, we can develop statistics to analyze their morphologies, populations, and production rates in more detail as was discussed in [273]. For illustration in Fig. 6-14, employing generated *Blips*, we have reduced the dimensionality of the artificial population of L1 with Principal Component Analysis (PCA) [274]. By visual inspection, we can see three main clusters that we classify with Gaussian Mixture [3] [276]. Each point represents a single fake *Blip* in PCA space coloured according to their cluster label.

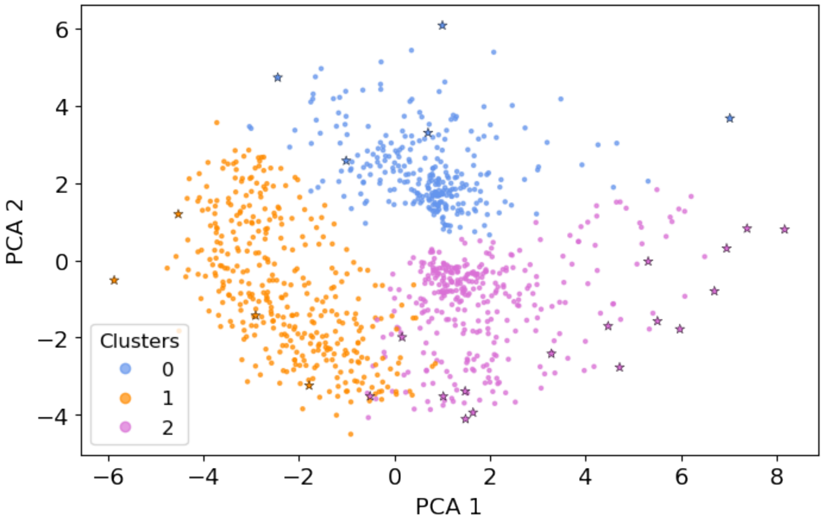


Figure 6-14: PCA representation of fake Blip population of H1, clustered with Gaussian Mixture. The 5% *most anomalous Blips according to the distance* $W_1(B_F, B_F)$ *are marked with a star.*

Furthermore, we have marked with a star 5% of the most anomalous *Blips* present in the population, according to their distance $W_1(B_F, B_F)$. It would be interesting to investigate the differences between the clusters in these distributions in future work. Another possibility would be to link the features of the *Blip* glitches with their representation in the latent space of the CT-GAN, as it was proposed in [277].

B. *Glitch template banks:* It is well-known that *Blip* glitches have a similar morphology to Intermediate-Mass Black Hole (IMBH), which hinders the detection of such events. With

---

[3] For both algorithms, we employ `scikit-learn` implementation [275]

our generator, we could create glitch templates to use matched-filtering techniques in unknown signals to compute a ranking statistic and weigh it in the likelihood function of detection pipelines. In this way, we would provide another metric to distinguish *Blip* glitches from IMBH.

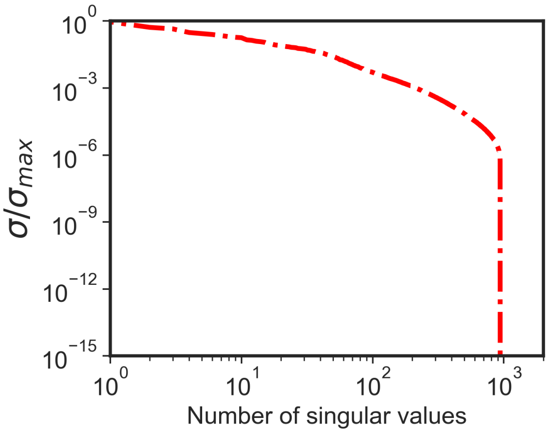


Figure 6-15: The singular values (σ) are obtained from a set of 10^3 *whitened glitches using SVD [278], normalized by the maximum singular values (σ_{max}). The glitches are generated from CT-GAN. The spectrum of singular values is seen to fall sharply, implying only a few singular values (e.g., $\ell = 10$), and corresponding basis vectors are sufficient to represent the glitches. See Fig. 6-16 in which the relative reconstruction error for these glitches has been shown based on $\ell = 1, 5, 10$. For performing SVD based matched filtering for glitch templates, we followed the framework presented in [123].*

We could use the standard matched-filter method [45] to compute the SNR time series for a specific glitch template. However, performing a matched filtering operation for a large glitch bank will be a huge task as computational time will increase drastically. We need to handle the scalability issue of the computational time of performing matched filtering with the increased number of glitch templates as we would expect to manage many glitch templates.

We can resolve this scalability issue if we adapt the matched filtering framework used in the GstLAL [112, 111] pipeline for the searches of GW signals from CBC sources. We observed that a few numbers of basis obtained using Singular Value Decomposition (SVD) [278, 123, 279, 280] can also represent the glitch templates and that basis can be used to get the matched filter output quickly. The computational time complexity of matched filtering can be reduced as the required number of basis vectors is much less than the number of glitch templates.

To show the efficacy of this framework, we generated 10^3 glitches for the L1 detector using our proposed CT-GAN-based glitch generator. We used 1 second data, sampled at 4096 Hz for this study. The data contains an injected glitch and coloured Gaussian noise with Advanced LIGO Zero Detuned High Power noise power spectral density [281]. Since the generated glitches are around 0.23 s (938 data points) sampled at 4096 Hz, we padded them with zero and made them 1 s long to generate the noisy data. The amplitudes of the injected glitch were adjusted for a target SNR of 10. Further, we used Zero Detuned High Power to whiten the data and the glitch templates. We computed the SNR time-series for each glitch template based on (a) standard matched-filter method [45] redefining Eq.

3.9 as

$$(s|g) = 4\,\mathrm{Re} \int_0^\infty \frac{\tilde{s}(f)\,\tilde{g}^*(f)}{S_n(f)}\,df\,, \tag{6.14}$$

Option (b) is the SVD based matched filter [123] in which a set of few top basis vectors have been computed from glitch-matrix first. Since each glitch template has 4096 data points, therefore the dimension of the glitch matrix is of size $10^3 \times 4096$ after stacking all the glitches together. After that, the basis vectors are matched-filtered against data, and the SNR time-series has been computed by combining coefficients of each glitch and matched filter output obtained based on basis and data.

For our example, we obtained that 10 top-basis vectors are sufficient to represent those 10^3 glitches, as it can be observed in Fig. 6-15. It shows that the singular values of a set of 10^3 glitches fall steeply, which implies a few top-basis (e.g., $10, 20$) can be used to represent those glitches. We have chosen the number of top-basis $(\ell) = 1, 5, 10$ and reconstructed the glitches in our analysis. We have computed the reconstruction error for each glitch as follows:

$$\epsilon_\alpha = \frac{\|g_\alpha - \hat{g}_\alpha\|_2}{\|g_\alpha\|_2}\,; \alpha = 1, 2, \cdots, 10^3 \tag{6.15}$$

where $\hat{g}_\alpha$ is the reconstructed whitened glitch based on $\ell = 1, 5, 10$ basis vectors respectively and $\|\|_2$ represents L_2 norm, and α is the number of total glitch templates. We also computed the fractional SNR-loss [123] for each glitch templates based on following definition:

$$\frac{\delta\rho_\alpha}{\rho_\alpha} = \frac{|\rho_\alpha| - |\hat{\rho}_\alpha|}{|\rho_\alpha|}\,; \alpha = 1, 2, \cdots, 10^3 \tag{6.16}$$

With the increasing number of bases, the relative reconstruction error should be decreased. To corroborate this statement, in Fig. 6-16, we choose three different cases with varying $\ell = 1, 5, 10$. Fig. 6-16 shows the probability density of the relative error ϵ_α for $\ell = 1, 5, 10$ respectively. The figure shows that the relative error is less for $\ell = 10$, whereas the relative error is high for $\ell = 1$. Similarly, we obtained the fraction SNR loss for all glitch templates for these three cases. Fig. 6-17 shows the construction of glitch and SNR time-series based on $\ell = 1, 5, 10$ number of basis respectively. Both plots show that $\ell = 10$ is sufficient to reconstruct the whitened glitches and represent the SNR time series. If we increase the number of bases, the reconstruction errors $(\frac{\delta\rho_\alpha}{\rho_\alpha}, \epsilon_\alpha)$ can be reduced but matched filtering cost would increase. Hence, we need to choose a minimal set of bases for which computation cost and also the reconstruction errors are low. We have chosen $l = 10$ as that minimal number for this specific example.

In a follow-up work, we will explore the possibility of the construction of a glitch bank construction, with a discussion on how to obtain ranking statistics, and signal consistency tests.

C. *Mock data challenges:* With our methodology, we can generate glitches in the time domain. The user could generate as many glitches as necessary, selecting the ones that represent best the real distribution and injecting them in real detector noise to create a realistic data challenge. Moreover, since certain anomalies are generated, those can also be selected for stress-test analysis algorithms.

As a preliminary test, the author in [217] injected some *Blip* glitches in the O3a data to evaluate how they will impact the long-duration analysis with a dedicated neural network called `ALBUS`. For visualization, we present the output in the right panel of Fig. 6-18. Since a time resolution is much larger than the glitch duration (i.e., < 0.3 s), the injected

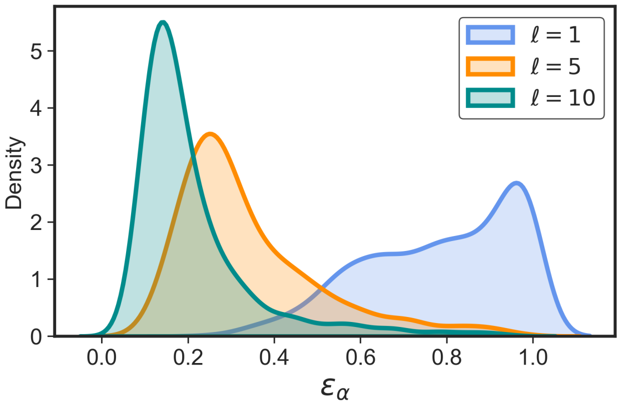


FIGURE 6-16: *The plot shows the distribution of relative errors for the reconstruction of the* 10^3 *whitened glitches generated using CT-GAN. The relative error (ϵ_α) is calculated for each case.*

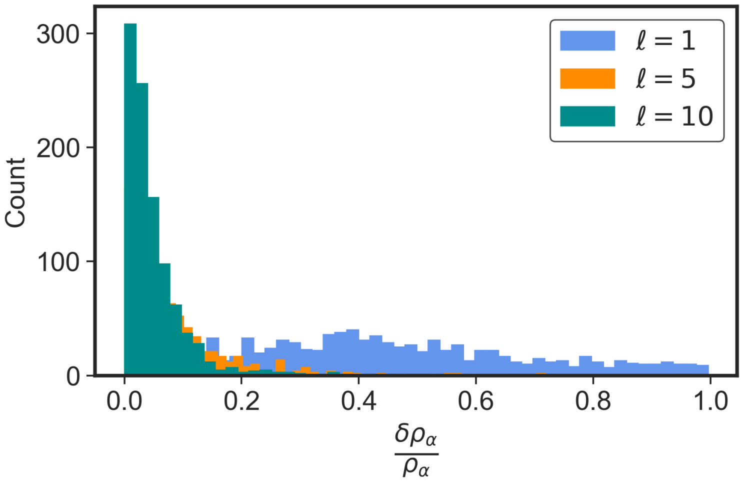


FIGURE 6-17: *This figure shows the histogram based on the fractional SNR loss ($\frac{\delta\rho_\alpha}{\rho_\alpha}$) for a set of glitches (10^3). For each glitch template, the SNR time-series were obtained based on (a) Standard matched-filter scheme and (b) SVD based matched filtering framework presented in [123] by varying the top-basis numbers as $\ell = 1, 5, 10$ respectively.*

glitch appears as a vertical line. The structure of the glitch is fully recovered and improves the detection capability of `ALBUS`. As suggested in [256], when learning different classes of glitches, we could also interpolate between them to generate hybrid classes. This hybrid dataset could be employed to discover unknown classes of glitches and improve the efficiency of detection algorithms.

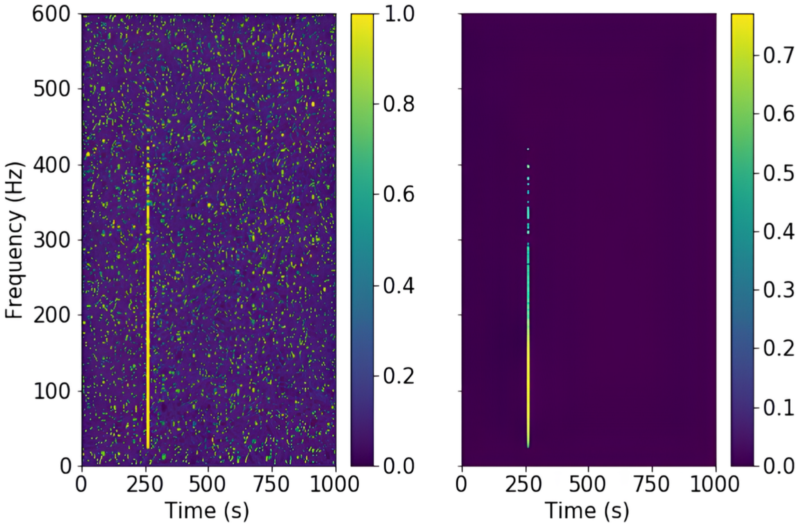


Figure 6-18: Example of glitch injection. The left image shows the input time-frequency map while the right panel shows the output of `ALBUS`*. Retrieved from [217].*

D. *New glitch detection:* Once the network has learned the underlying distribution of the data, with certain modifications, it can output how likely it is for an unknown signal to belong to the known distribution. This metric can detect anomalous generations and provide feedback to classification algorithms. For example, this information `Gravity Spy` could re-classify certain anomalies, which could imply the definition of new glitch classes and their further characterization.

E. *Improving glitch classification:* One of the main challenges of working with real data is dealing with imbalanced data sets. With our methodology, once more classes are learned, we could generate balanced data sets to improve the accuracy of classification algorithms.

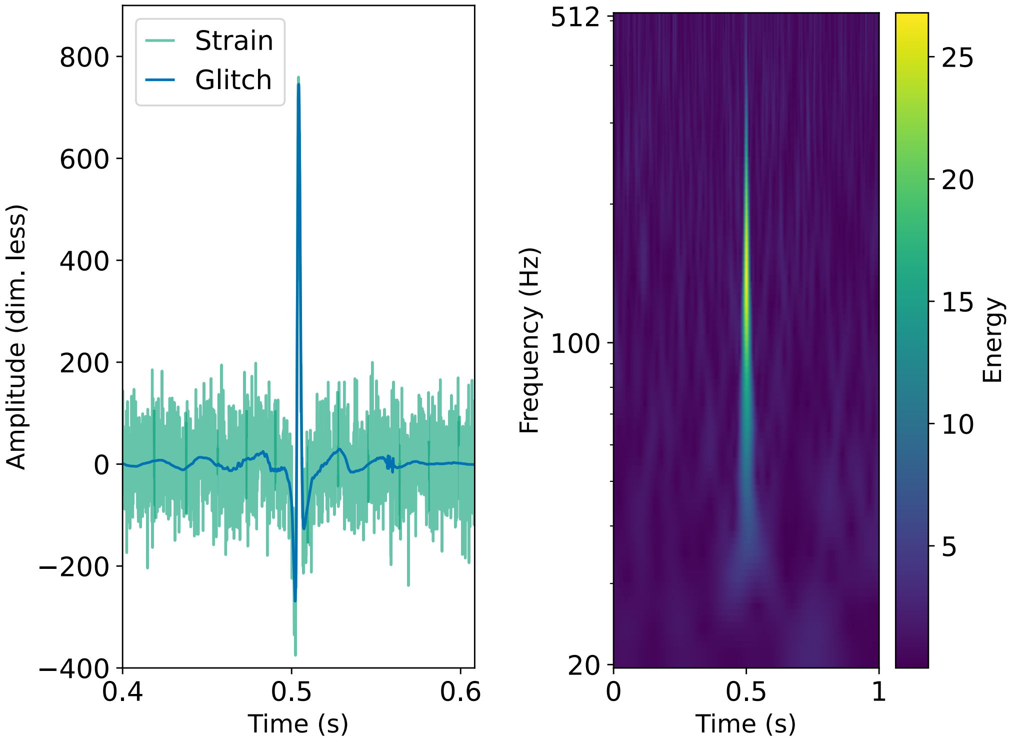
6.4.6 `gengli` package

To make this method available for a broad audience, the trained CT-GAN was made available within the package `gengli`[4], which is a flexible and user-friendly tool for glitch generation. With this tool, it is straightforward to use the generator network to produce random glitches starting from random samples from its latent space. The output of the generator, namely a raw glitch, is a *whitened* glitch evaluated on a fixed length time grid at a constant sampling rate 4096 Hz and an amplitude in the range $[-1, 1]$. We can generate a raw glitch in ~ 10 ms on a laptop.

The raw glitch generated by the GAN is often not suitable for immediate application, so several post-processing steps are required, which have been implemented in `gengli`. In

---

[4] The code is released as a Git repository: `https://git.ligo.org/melissa.lopez/gengli`, and the full documentation can be found in `https://melissa.lopez.docs.ligo.org/gengli/index.html`.



(A) Time series (B) Q-transform

Figure 6-19: A whitened glitch added to the white noise of Einstein Telescope, retrieved from [216].

the following, we provide a brief description, but the interested user should refer to the full documentation.

Post-processing

The simplest post-processing step implemented in `gengli` is *resampling*[5] the raw glitch at a desired sampling rate. As conceptually, the glitch forms part of the detector noise, to add the raw glitch to it, the noise time series needs to be whitened so that the resulting time series is: $s_w = n_w + g_w$. Nonetheless, the user might need the raw time series with the glitch, so we need to *re-color* it. Mathematically,

$$s_c = \mathcal{F}^{-1}(\sqrt{S_n(f)'}\tilde{s}_w) \tag{6.17}$$

where s_w and s_c are the whitened and coloured detector noise containing the glitch, respectively, and $S_n(f)'$ represents the PSD of the noise. Here, $\mathcal{F}^{-1}$ is the inverse Fourier transform. Furthermore, it is useful to *scale* the glitch to inject it at a desired "loudness", or SNR ρ using Eq. 6.10. In Fig. 6-19 we plot an artificial glitch (blue) scaled to a ρ_{target}, added in the white noise of Einstein Telescope, in time-series and time-frequency representation. Note these are built-in functionalities.

Selecting generations

As for any randomly distributed quantity, any randomly generated glitch comes with a different degree of "similarity" with respect to the statistical distribution. We want to make this notion

---

[5] When upsampling, we make the key assumption that there are no interesting features at frequencies higher than 4096 Hz.

mathematically precise and define an anomaly percentile for each randomly generated glitch. As before, we consider three distances between a pair of glitches:

- Wasserstein distance d_W: a standard measure of distance between distributions, commonly employed in ML and defined in [261].
- Mismatch d_M: a measure of distance based on the details of the filtering, standard in the GW field, and defined as 1−match from Eq. 6.11.
- Cross covariance d_{cc}: we employ the quantity $1 - k$, where k is the normalized cross-covariance as defined in Eq. 6.13.

We then generate a benchmark set of N_b glitches from the generator. For each of the $N_b(N_b - 1)/2$ pairs of glitches in the benchmark set, we compute the three distances above. In Fig. 6-20 we show the distribution for a population of $N_b = 1000$.

For each new glitch being generated, we compute the set of average distances (d_W, d_M, d_{cc}) between the glitch and the benchmark set and we measure the set of percentiles (p_W, p_{mm}, d_{cc}) of each of the distances with respect to the benchmark distances. The triple (p_W, p_{mm}, d_{cc}) is our novelty measure for the glitch. This allows us to filter glitches based on an anomaly score interval $[p_{min}, p_{max}]$. The code will output only glitches for which *all* the three anomaly scores lie within the given interval.

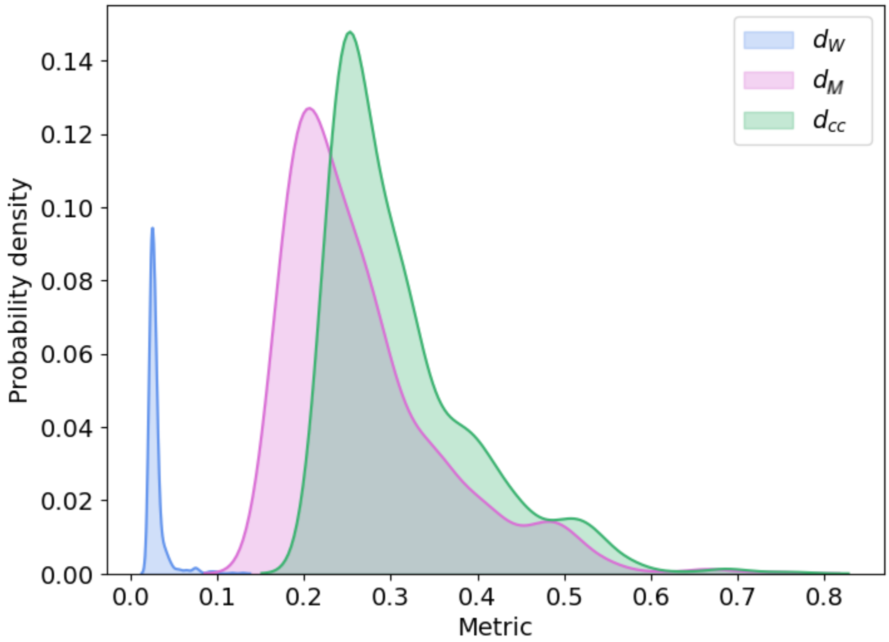


FIGURE 6-20: Probability distribution of distances (d_W, d_M, d_{cc}) for a benchmark population of 10^3.

`Gravity Spy` studies

During the review process of LIGO-Virgo-KAGRA, we realized that there was an over-writing error in `Gravity Spy` classifications of `gengli`. `Gravity Spy` stores images, with a pre-defined name, to then read them and provide a classification. When parallelizing the code, `Gravity Spy` would overwrite the images, giving the wrong classification, i.e. if image A was stored, immediately it would overwrite with image B, to then store the classification of B as the classification of A. This problem caused the Fig. 6-12 to have an erratic behaviour in the `Gravity Spy` confidence c_{GS} dimension.

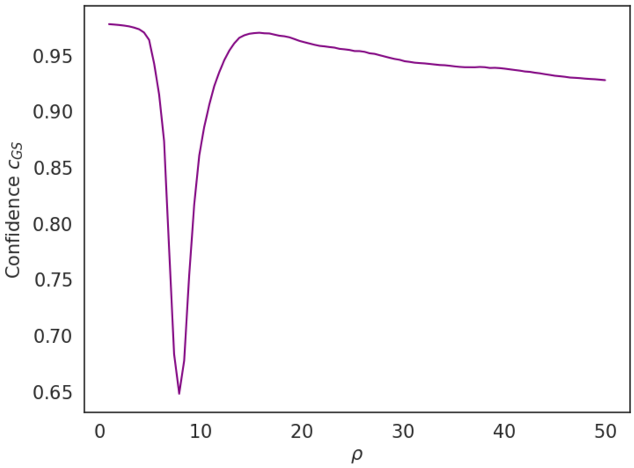


Figure 6-21: Average `Gravity Spy` *confidence c_{GS} for 10^3 artificial Blip glitches, generated with* `gengli`*, as a function of SNR ρ.*

Once this issue was solved, we repeated some of the calculations from [249] but with a larger ρ range. For this aim, we created a population of 10^3 glitches and we injected them in real whitened noise with SNR $\rho \in [1.0, 50]$, using the procedure presented in Section 6.4.2. Note that this SNR is $> \rho_{Blips} \sim 20$, as we want to test the behaviour of `Gravity Spy` for louder signals. To avoid the influence of the background we use the same real whitened noise from H1 in the range $[1262540000, 1262540040]$ GPS time for every glitch realization. Note that this time series includes sufficient data to avoid border effects in the whitening procedure.

In Fig. 6-21 we present the average `Gravity Spy` confidence c_{GS} for 10^3 artificial *Blip* glitches as a function of SNR ρ. We can observe that c_{GS} decreases as we increase ρ, with a minimum of $c_{GS} \approx 0.65$ at $\rho \sim 7.5$. As we will see in the next graphics, this minimum is because `Gravity Spy` is confident that there is no glitch in the input data, known as *No_Glitch* class, until we increase the loudness of the glitches to $\rho \sim 7.5$. As we pointed out in Section 6.4.2, this behaviour is expected, since `Gravity Spy` is trained on Omicron's triggers with $\rho > 7.5$. Furthermore, `Gravity Spy` performance decreases slowly with the increase of ρ, which might be a symptom that `Gravity Spy` is biased towards loudness. Nonetheless, further investigation is needed.

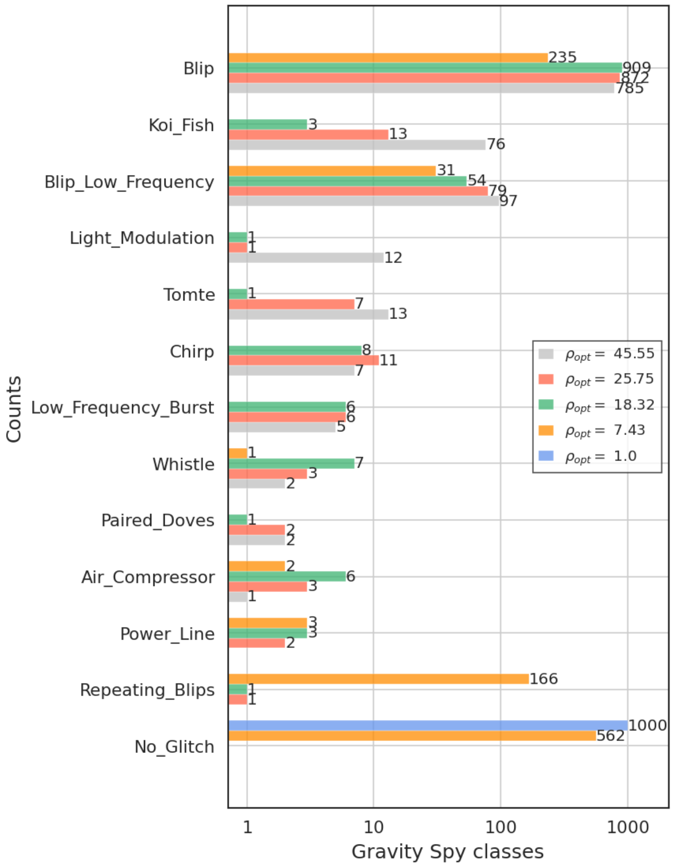
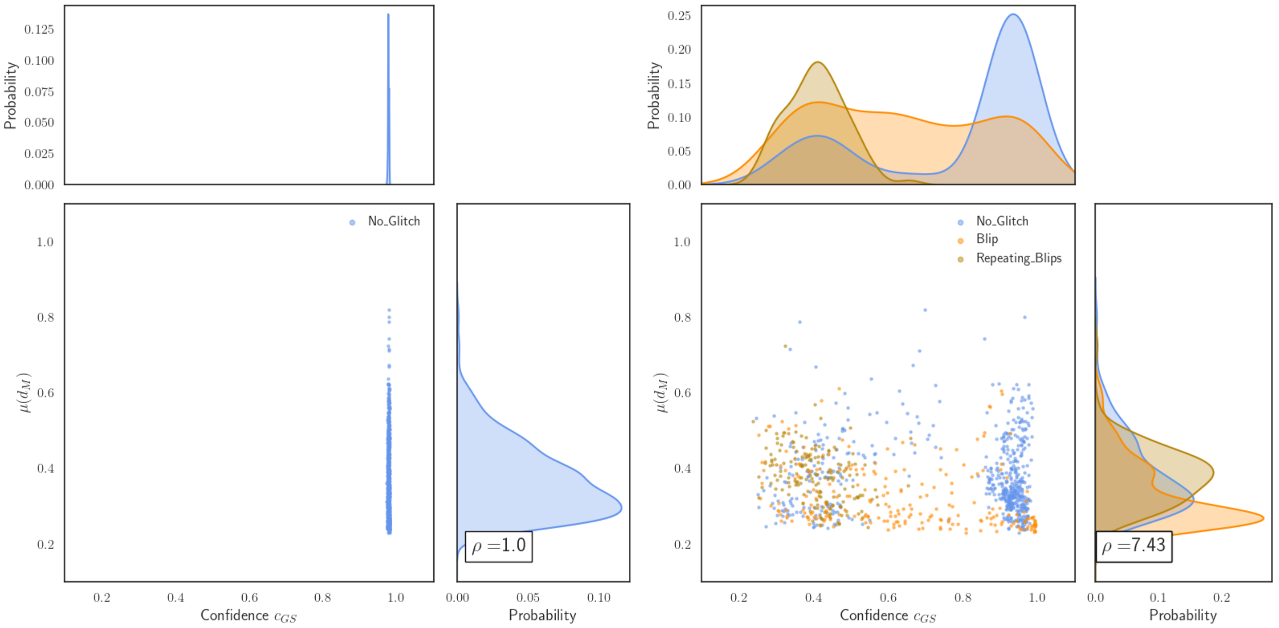
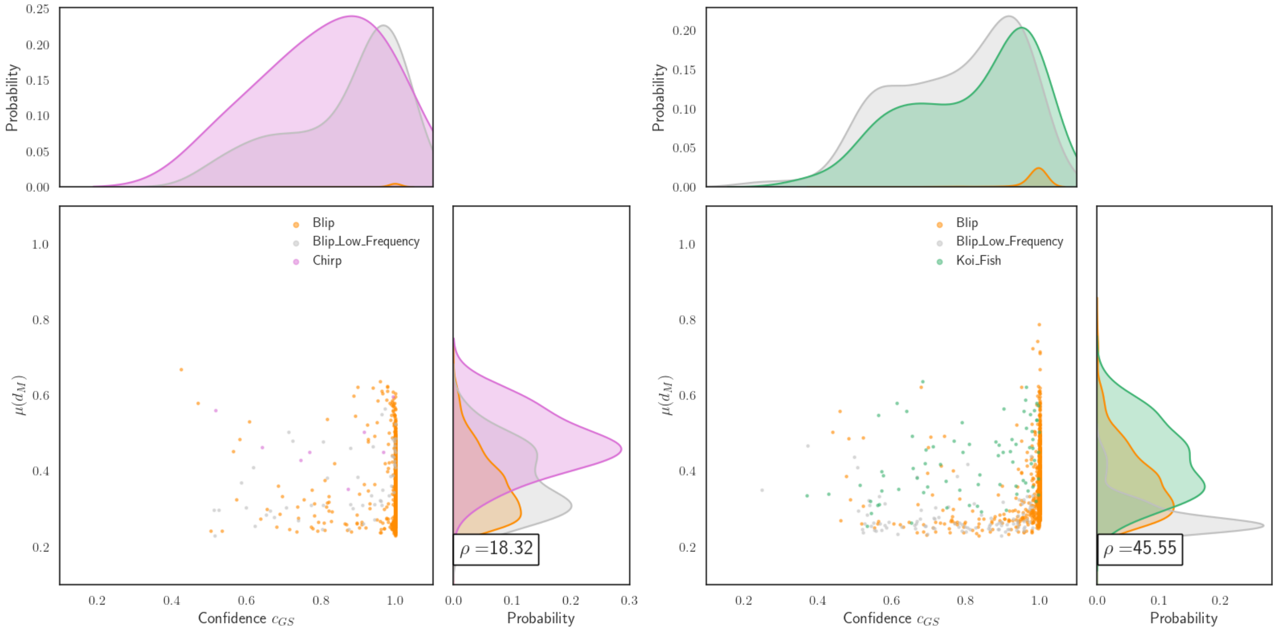


Figure 6-22: `Gravity Spy` *classes for 10^3 generated Blips from H1.*

In Fig. 6-22 we correct Fig. 6-10, where we plot the classification labels, with maximum classification probability, for different ρ_{opt} of H1 population. As we mentioned before, we can observe that at $\rho = 1.0$ every glitch is classified as *No_Glitch*, since the signal is below the bed of the noise. However, as we increase $\rho = 7.43$, `Gravity Spy` classifies some of the glitches as *Blips*, and some others as *Repeating_Blips*. At $\rho = 18.32$ most glitches are classified as *Blips*, with some misclassifications probably caused by anomalies in the data. Nevertheless, as we increase ρ, the number of *Blip* classifications degrades, increasing the number of glitches that are classified as *Koi_Fish* or *Blip_Low_Frequency*.



(A) $\rho_{opt} = 1.0$ (B) $\rho_{opt} = 7.43$

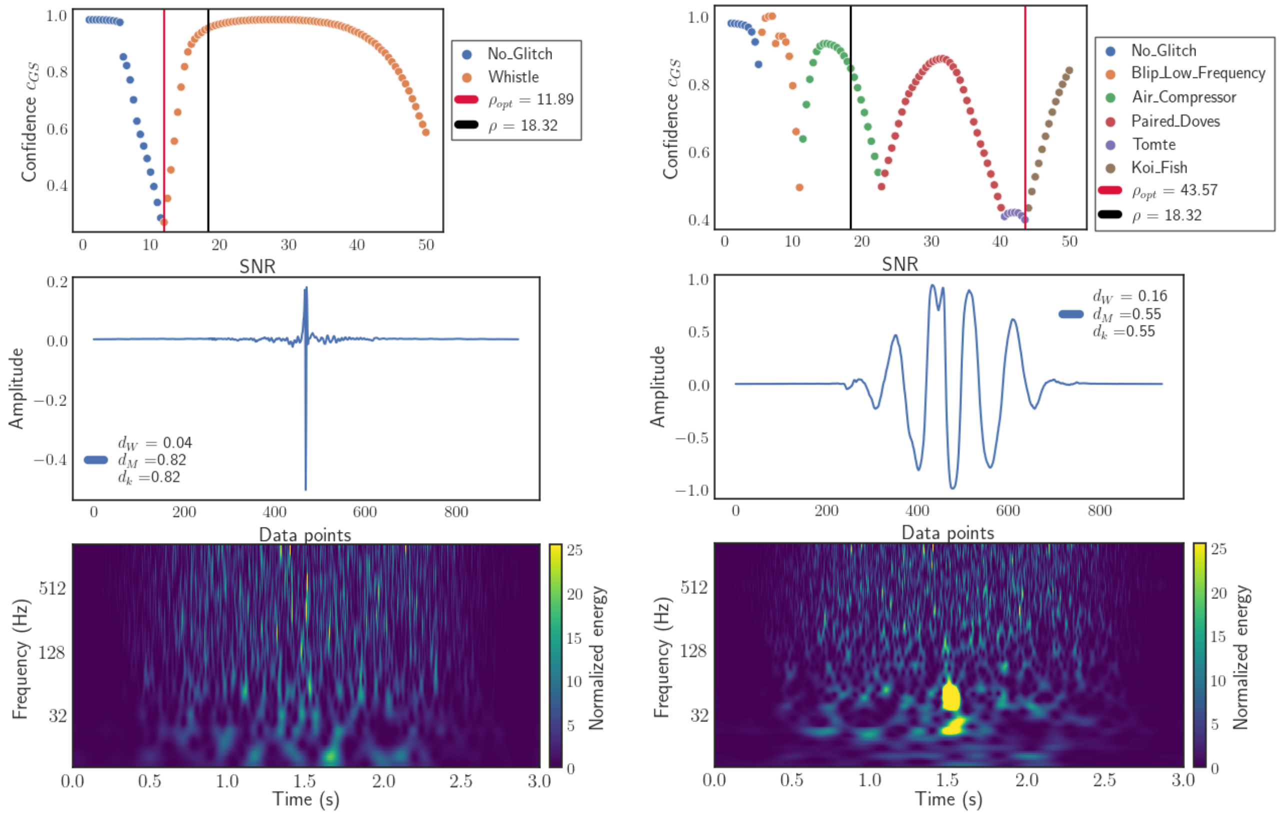


(C) $\rho_{opt} = 18.32$ (D) $\rho_{opt} = 45.55$

FIGURE 6-23: *Joint and marginal distribution of* **`Gravity Spy`** *confidence* c_{GS} *at* $\rho_{opt} = 18.46$ *against mis-match* d_M *for different glitch classes for H1: No_Glitch (blue), Blip (orange), Repeating_Blips (brown), Blip_Low_Frequency (grey), Chirp(pink), Koi_Fish (green).*

In Fig. 6-23, and to correct Fig. 6-12, we present the joint and marginal distribution of c_{GS} as a function of the mismatch d_M, as defined in Section 6.4.6. Note that the marginal distributions are expressed as probability densities. For Fig. 6-23a, where $\rho = 1.0$, we can see how every glitch is classified as *No_Glitch* with $c_{GS} \approx 1.0$. As we increase $\rho = 7.43$ (see Fig.

6-23b) `Gravity Spy` starts to change its classification towards *Blip* class, which is dominant with high c_{GS} for $\rho = 18.43$ (see Fig. 6-23c). However, for $\rho = 45.55$ and as observed in Fig. 6-23d, while still most of the glitches are classified as *Blip*, we begin to see misclassifications of *Blip_Low_Frequency* and *Koi_Fish* with $c_{GS} > 0.6$. As a final test, we selected a glitch that was anomalous according to `Gravity Spy` but not `gengli` (Fig. 6-24a), and a glitch that was anomalous for both, `Gravity Spy` and `gengli` (Fig. 6-24b).



(A) Anomalous glitch according to `Gravity Spy`, but not `gengli`.

(B) Anomalous glitch according to `Gravity Spy`, and `gengli`.

Figure 6-24: Selected `gengli` *glitches. (Top panel): confidence* c_{GS} *as a function of SNR. (Middle panel): Raw glitch signal output from* `gengli`*. (Bottom panel): spectrogram of glitch injected in real detector noise from H1 at* $\rho = 18.32$.

On one hand, in Fig. 6-24a (middle panel) we can see that the glitch has a narrow peak, but according to `gengli`'s metrics, its morphology is close to the rest of the glitch population (low d_W, high d_M and d_k). In the top panel, we can see that `Gravity Spy` is unable to see this glitch until $\rho = 11.89$, wrongly classifying it as *Whistle* with $c_{GS} \approx 1.0$ for $15 \lesssim \rho \lesssim 40$. Afterwards, c_{GS} degrades with increasing ρ. In Fig. 6-24a (bottom panel) we can see that the glitch is faint even at $\rho = 18.32$.

On the other hand, in Fig. 6-24a (middle panel) we can see that the glitch does not have a standard *Blip* morphology (high d_W, low d_M and d_k). Since it is an anomalous glitch, `Gravity Spy` is confused about its class. In Fig. 6-24a (bottom panel) we can see that the glitch at $\rho = 18.32$ does not have the characteristic "tear-drop" shape of *Blip* glitches.

6.5 Discussion

As we have seen in the previous Sections, `gengli` is a powerful tool to further understand current glitch identification algorithms, and help enhance searches of GW, as it was proposed in [217], and [216]. Nonetheless, the main limitation of `gengli` is that it can produce a single class of glitches, and according to experts in detector characterization, 23 classes exist. In Section 6.4.4, we commented that one of the main limitations of this approach is that GAN need a lot of data to learn the underlying distribution of the population, which might be challenging for glitch classes with < 100 samples. Furthermore, the pre-processing using BW is computationally intensive. Indeed, just to extract the populations of *Blip*, *Tomte* and *Koi_Fish* glitches during O3 we had to use ≈ 1.2 million CPU hours.

To overcome this issue, authors in [282] and [283] propose to use TorchGAN and ProGAN, respectively, with time-frequency representations, instead of time series, for all glitch classes. The main goal of these two approaches is to tackle the imbalanced problem of `Gravity Spy`, where some glitch classes have less than 100 examples, while other classes have more than $10,000$ examples. Nonetheless, this is also a problem for GAN-based algorithms. Thus, authors in [282] propose to over-sample glitch classes to $5,000$ examples, i.e. they sample randomly from the class distribution allowing examples to occur more than once. On the other hand, authors in [283] do not employ similar techniques as, due to the architecture of ProGAN, the underlying distribution of the data is learnt with a better generalization and less overfitting.

While both algorithms proposed in [282] and [283] show high performance when classifying artificially generated glitches with CNN, proving the added value of data-augmentation approaches, these methods are less flexible for mock data challenges as the glitch is not separated from the original background, a non-trivial task. Hence, authors in [284] and [285] continue to think about novel GAN architectures in time-series, also demonstrating that data-augmentation approaches will not only be relevant for current detectors, but also for third generation detectors such as Einstein Telescope.

6.6 Conclusion

In this chapter, we have discussed the generation of glitches with ML. In our work based on [249] and [250], we have developed a methodology to generate artificial *Blip* glitches from real data using a ML algorithm known as GAN. To be able to generate these glitches, the input *Blips* need to be processed: the signals are selected from `Gravity Spy` data to be reconstructed with BayesWave and smoothed with the rROF algorithm. Because of this heavy processing, only around 66% and 50% of the initial data from L1 and H1 is preserved.

Due to the instability of GAN algorithms, in this particular research, we trained a CT-GAN [265] modified to process 1-dimensional inputs instead of 2-dimensional. The network uses the Wasserstein distance as a loss function, which allows it to train its discriminator to optimality. It is heavily penalized to avoid training instabilities and to accurately learn the underlying distribution of *Blips*.

To assess the performance of CT-GAN, we generate a population of 10^3 *Blip* glitches for both H1 and L1. The quality measurements employed are `Gravity Spy` classifier and similarity distances. The results of these metrics indicate that the neural network was able to learn the underlying distribution of *Blip* glitches from H1 and L1, despite the presence of some anomalous generations due to imperfections of the input data set. Furthermore, it has been observed that the similarity distances can detect miss-classifications from glitch classifiers.

In this proof-of-concept investigation, we have demonstrated that it is possible to isolate *Blip* glitches from their surrounding noise and learn their underlying distribution with an ML-based method in the time domain, providing several examples of its usage. This methodology allows us to generate better quality data, and it provides us with flexibility that would be challenging to achieve with time-frequency representations. Furthermore, we also present our open-source package `gengli`: it provides an easy-to-use interface to the trained GAN output and has some additional features such as building a glitch population with or without anomalies, *resampling*, *re-colouring* and *scaling*, and further examples within the documentation. As a proof-of-concept, we use `gengli` to understand the behaviour of `Gravity Spy`.

The long-term goal of this investigation is to learn other classes of glitches in the time domain. While extracting glitches from its background is an expensive task, it could be improved via specific wavelet design for glitches, among other possibilities. Nonetheless, a fundamental issue remains: there is no guarantee that all glitches belonging to a certain class present the same standard behaviour.

Chapter 7

Detection of Anomalous Transient Noise Bursts

This chapter is based on work presented in Ref. [286] and its companion, where I supervised the data pre-processing, the construction of the models and the assessment of the performance. I also was the lead writer in [286].

7.1 Introduction

While glitch identification and characterization is a crucial first step towards their mitigation [63, 39], current glitch classifications are not flexible, nor exhaustive nor representative of all glitch morphologies, basing their classification on a combined human and ML approach using the main strain of the detector $h(t)$ (see Section 2.4.3 for a discussion). In this context, unsupervised ML methods to identify glitches could help overcome such limitations. In this chapter, we propose a novel ML algorithm that combines auxiliary channel information-where the physical mechanism of the resulting glitch is recorded-with an unsupervised anomaly detection algorithm. Due to the overwhelming amount of information, we encode the auxiliary channels from LIGO Livingston in the fractal dimension, a measure of the complexity of the time series. This representation of the data is input to a data-driven algorithm, which consists of a convolutional autoencoder with periodic convolutions that learns the underlying representation of the data, clustering glitches according to their similarity in a compressed representation. By exploiting this compressed representation for anomaly detection, we can identify glitches that strongly deviate from the general distribution of the input data, improving the understanding of glitch populations.

While unsupervised ML algorithms are agnostic, as they do not make prior assumptions regarding the data distribution, it is challenging to interpret their results. To understand the results of our algorithm and assess its performance we can compare the output of our algorithm with the findings of supervised glitch classifiers, employing them as a benchmark. In the following, we describe the benchmark used in this work, the selection of glitch populations and the FD-encoding of the data.

7.2 Data

7.2.1 Characterization via auxiliary channels

The status of GW detectors is continuously monitored through a large set of data streams at various sampling rates, outputting $\sim 10^6$ time-series from instrumental and environmental sensors. These auxiliary channels can be divided into safe (insensitive to GW) and unsafe (sensitive to GW). Depending on their origins, glitches present varied morphologies in different sets of auxiliary channels. Some subset of these channels may serve as "witnesses" of glitches and are used to create data quality flags before performing GW searches [38, 42, 79].

Despite the huge amount of auxiliary channels in a single detector, many of them do not provide useful information for noise transient investigations as they remain constant or vary with a consistent pattern, constituting a data set containing redundant and/or non-informative characteristics [287, 180, 52]. Therefore, LVK researchers have compiled a "reduced" standard list of $\sim 10^3$ auxiliary channels that are used in data quality investigations. In this work, we limit our investigation to safe auxiliary channels with sampling rates $> 512\,\text{Hz}$, yielding a set of 347 channels.

7.2.2 Fractal dimension

The first step towards characterizing glitches through safe auxiliary channels requires identifying anomalous data stretches within them [180, 73, 79]. In [288], the author proposes the measurement of fractal dimension (FD) as an additional effective tool for characterizing the instrument output in low latency. FD is an index that characterizes the self-similarity of a set and provides a measure of the complexity of the signal in the context of signal processing [289]. There are several definitions of this quantity [290, 291, 292], implying that the FD measure for a physical process can differ depending on the chosen definition. Nonetheless, we focus on the *FD variation over time* as an indicator of the evolution of the signal's complexity. As the presence of a glitch in the data affects the noise power spectrum, which in turn varies the value

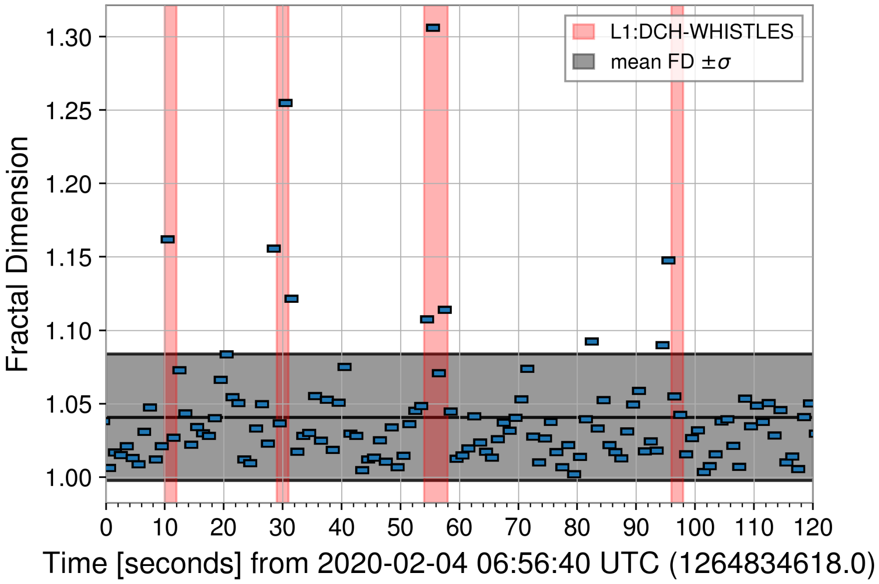


Figure 7-1: Fractal dimension over a two-minute period of L1 data for the `L1:LSC-PRCL_OUT_DQ` auxiliary channel. Each point represents the fractal dimension for one second of data, and the red regions indicate the time period containing `Whistle` glitches. The grey region indicates the confidence interval of the fractal dimension at one standard deviation.

of FD, we are only interested in the relative change which is definition independent.

To illustrate this, Fig. 7-1, reproduced from [288], presents the variation of FD for two minutes of data from the `L1:LSC-PRCL_OUT_DQ` auxiliary channel, which measures the Power Recycling Cavity Length (PRCL) from the Length Sensing Control (LSC) of the LIGO Livingston (L1) interferometer. The computation was performed with a time window $\mathcal{W}(t) = 1\,\mathrm{s}$, i.e. every FD value is the result of encoding 1 s of the input data. Points greater than one standard deviation σ from the mean FD correlate to the presence of `Whistle` glitches in the detector. As we can observe from Fig. 7-1, FD can be an effective tool to further understand the coupling between glitches and auxiliary channels. To extend this analysis to a larger set of safe auxiliary channels and glitch classes, we first need to speed up the FD calculation to near-real time.

Following [288] we numerically estimate the measured FD with the variation (VAR) method (see [288] for details). For a discretely-sampled set of data with N measurements $\mathcal{C} \in \mathbb{R}^N$, we can define a sliding window to compute the variation of the data with centre l and scale k,

$$\mathcal{F}_{k,l} = \left|\max\left[\mathcal{C}_{l-k}, \dots, \mathcal{C}_{l+k}\right] - \min\left[\mathcal{C}_{l-k}, \dots, \mathcal{C}_{l+k}\right]\right|. \tag{7.1}$$

Thus, the VAR estimator for a given scale k is,

$$\mathcal{A}^{\mathbf{VAR}}(k) = \frac{1}{N-2k} \sum_{l=k}^{N-1-k} \mathcal{F}_{k,l}, \tag{7.2}$$

As we can see in Algorithm 1, the implementation in [288] computes the maximum and minimum over a range of values at each iteration k, l (lines 5 and 6). The runtime of this implementation is $\mathcal{O}(N^3)$. A significant speed-up can be achieved using Algorithm 2 based on [293]. It uses the fact that at iteration $k+1$ we can compute the maximum as

$$\begin{aligned}\max[\mathcal{C}_{l-(k+1)}, \dots, \mathcal{C}_{l+(k+1)}] = \max\{&\max\left[\mathcal{C}_{(l-1)-k}, \dots, \mathcal{C}_{(l-1)+k}\right], \\ &\max\left[\mathcal{C}_{(l+1)-k}, \dots, \mathcal{C}_{(l+1)+k}\right]\},\end{aligned} \tag{7.3}$$

where the components of the right-hand side have already been computed at iteration k. This step is done on line 10 in Algorithm 2, and likewise in line 11 for the minimum. Now, the computational complexity of the FD calculation is $\mathcal{O}(N^2 \log(N))$ and the practical speed-up can be seen in Fig. 7-2, where we compute FD with both methods over data increasing in length. While this speed-up is not apparent for short stretches of data at low sampling rates, it becomes significant at sampling rates $\geq 4096\,\mathrm{Hz}$.

In practice, with Algorithm 1 with computational complexity $\mathcal{O}(N^3)$, and given 1 h of data a single auxiliary channel sampled at 16,386 Hz, we were able to FD-encode it in 1 h, but with an efficient implementation with `numba` [294] of Algorithm 2 based on [293], with computational complexity $\mathcal{O}(N^2 \log(N))$, we can now process it in 11 s. With further parallelization in a cluster, the FD computation could characterize glitches in low latency. Now that we have a fast computation of FD-value, we can construct a data set for our application.

7.2.3 Selected glitches

In the present work, we employ `Gravity Spy`- a supervised classification method that uses time-frequency representations of $h(t)$-as a benchmark, finding anomalies from its high-confidence

Algorithm 1 Implementation of the VAR method from [288].

Input: f vector of size N.

Output: A vector of size $N/2$.

```
$A[1 \dots N/2] = 0$
for $k = 1$ to $N/2$ do
    $F = \emptyset$
    for $l = k$ to $N - k$ do
        $F = F \cup \{\max\{f[l-k], \dots, f[l+k]\} - \min\{f[l-k], \dots, f[l+k]\}\}$
    end for
    $A[k] = \text{mean}(F)$
end for
return $A$
```

Algorithm 2 Improved algorithm for the VAR method.

Input: f vector of size N.

Output: A vector of size $N/2$.

```
$A[1 \dots N/2] = 0$
$u[1 \dots N-2] = 0$
$b[1 \dots N-2] = 0$
for $i = 1$ to $N - 2$ do
    $u[i] = \max\{f[i], \dots, f[i+2]\}$
    $b[i] = \min\{f[i], \dots, f[i+2]\}$
end for
$A[1] = \text{mean}(u[1 \dots N-2] - b[1 \dots N-2])$
for $i = 2$ to $N/2$ do
    for $j = 1$ to $N - 2i$ do
        $u[j] = \max\{u[j], u[j+2]\}$
        $b[j] = \min\{b[j], b[j+2]\}$
    end for
    $A[i] = \text{mean}(u[1 \dots N-2i] - b[1 \dots N-2i])$
end for
return $A$
```

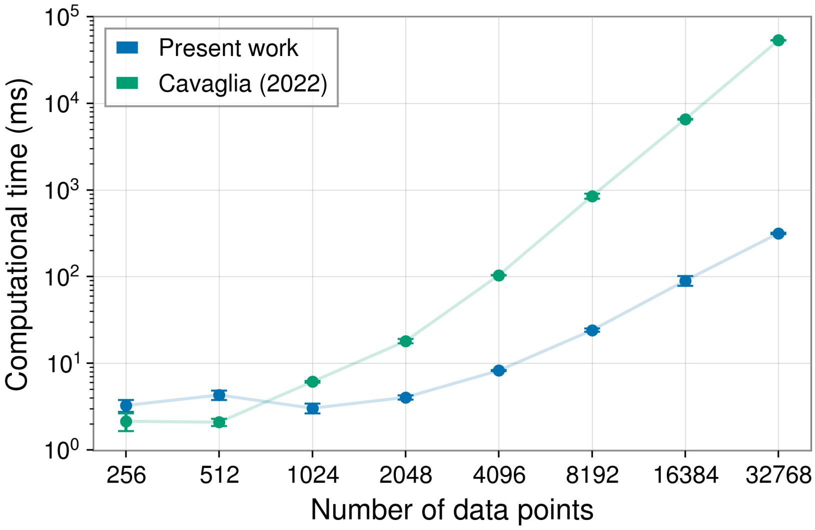
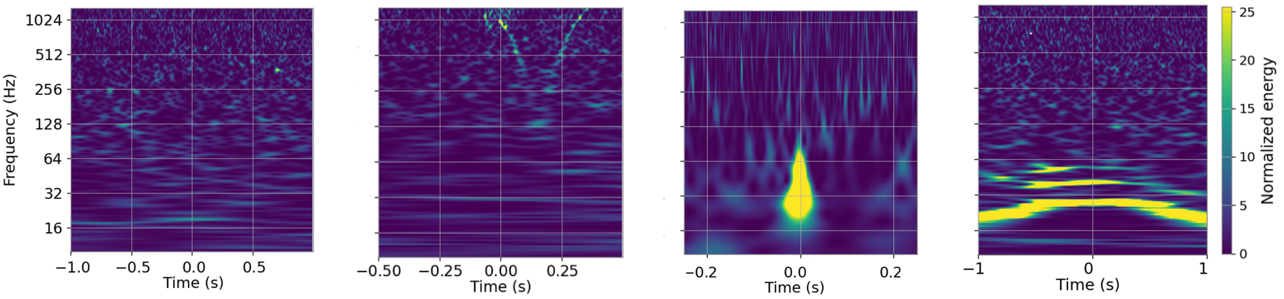


FIGURE 7-2: Comparison of the fractal dimension computing algorithms for varying number of data points. Benchmarks done on a Intel® Xeon® Processor E5-2630 v4 CPU @ 2.20GHz.



(A) `No_Glitch` class (B) `Whistle` class (C) `Tomte` class (D) `Scattered_Light` class

FIGURE 7-3: From left to right: Q-transform of `No_Glitch`, `Whistle`, `Tomte` and `Scattered_Light` retrieved from `Gravity Spy` [65].

classifications (see a description in Section 2.4.3). In this proof-of-concept work we use clean data samples and three different classes of glitches. The different classes are represented in Fig. 7-3. Our focus is on `Whistle`, `Tomte` and `Scattered_Light` glitches (see 2.4.3 for a description). Additionally, we include the `No_Glitch` class, which indicates the absence of significant excess power in the `Gravity Spy` spectrograms. This class represents a stable behaviour of the GW detector, which is reflected by non-deviant FD values. For `No_Glitch`, we select GPS times t where no glitch is present, which in $h(t)$ contain no apparent excess of power. For the three classes of glitch data, we select GPS times t of three distinct glitch morphologies in LIGO Livingston with `Gravity Spy` confidence $> 90\%$ [68]. One must note that for the glitches t represents the peak time of the Omicron alert. The three morphologies are chosen to have short and long-duration glitches that are abundant in LIGO Livingston data (> 800 samples per class), and that impact GW searches due to their wide frequency contribution. Moreover, in this study, we focus on LIGO Livingston, as the author in [288], but this investigation could be extended to LIGO Hanford and Virgo.

7.2.4 Auxiliary channels encoded in fractal dimension

Given a time t of interest, we select an array of GPS time with duration $\Delta t = 8\,\mathrm{s}$, where t is in the center. For each array of time, we retrieve 347 safe auxiliary channels with sampling rates

> 512 Hz, excluding the GW strain $h(t)$, that is then whitened and encoded in FD with time windows $\mathcal{W}(t) \in \{0.25, 0.5, 1, 2\}$ s. For each $\mathcal{W}(t)$ we have $\Delta t/\mathcal{W}(t) - 1$ time bins to ensure that t is in the center of the FD-encoded data, yielding a total of 56 time-bins per channel. Since the duration of `Scattered_Light` is $\sim 2 - 4$ s and the duration of `Tomte` is ~ 0.25 s, the length of these varying time windows ensure that any glitch morphology will be contained at least within $\mathcal{W}(t) = 2$ s [65]. Note that the sampling rate of each independent auxiliary channel varies, but we only encode safe channels with a sampling rate > 512 Hz, to have enough data points to perform a calculation of the FD, as demonstrated by the experiments in [288].

Limited by the number of `Whistle` present in LIGO Livingston, for the initial data set we select 896 GPS times for each class defined in Section 7.2.3 and presented in Fig. 7-3, yielding a balanced data set. Since each auxiliary channel monitors distinct physical processes, their average FD measurements can differ, giving priority to certain channels over others. To improve the stability of our model, we normalize in the range $[0, 1]$ the data of each auxiliary channel, as we are only interested in their relative variation. Normalizing collectively would give more importance to the channels with higher FD and dismiss the channels with lower FD.

We reduce the dimensionality of the normalized dataset, which originally has dimensions of 347 channels × 56 time bins, using a data-driven approach. Our goal is to retain the channels that capture the most relevant features of the glitches compared to the `No_Glitch` class. The procedure is as follows:

1. Defining D_{NG} as the set of `No_Glitch` FD-encoded, we compute the average of all elements in D_{NG}, $\mu(D_{\mathrm{NG}})$, to minimize extreme deviations of FD. This will be the common background when `No_Glitch` is present.

2. For a single glitch d_{C} encoded in from a certain class C, we subtract the background as $d_{\mathrm{C}} - \mu(D_{\mathrm{NG}})$. This subtraction highlights the deviations produced by the presence of a glitch in the data.

3. We identify auxiliary channels A that present a low FD deviation with respect to the background since their contribution is similar to the absence of a glitch. Thus, given a glitch d_{C} and a channel A, if $d_{\mathrm{C,A}} - \mu(D_{\mathrm{NG}})_{\mathrm{A}} \lesssim 10^{-2} \quad \forall$ glitches d_{C}, the auxiliary channel A is removed. This threshold represents a balance between data compactness and expressiveness. Too many channels can introduce irrelevant information, while too few may overlook the overall data trends.

This pre-processing reduced the dimensionality to a shape of 50 channels × 56 time bins . In Fig. 7-4 we show an example of 8 s of FD-encoded data. To train the ML model presented in the next section we will use the three glitch morphologies, with a total of 2688 samples, which contain the structure that we wish to unravel. One must note that while supervised approaches must use a subset of the data to assess the generalization ability of the model, in the present unsupervised approach we are interested in learning the details of the data at hand, such that all data instances are employed.

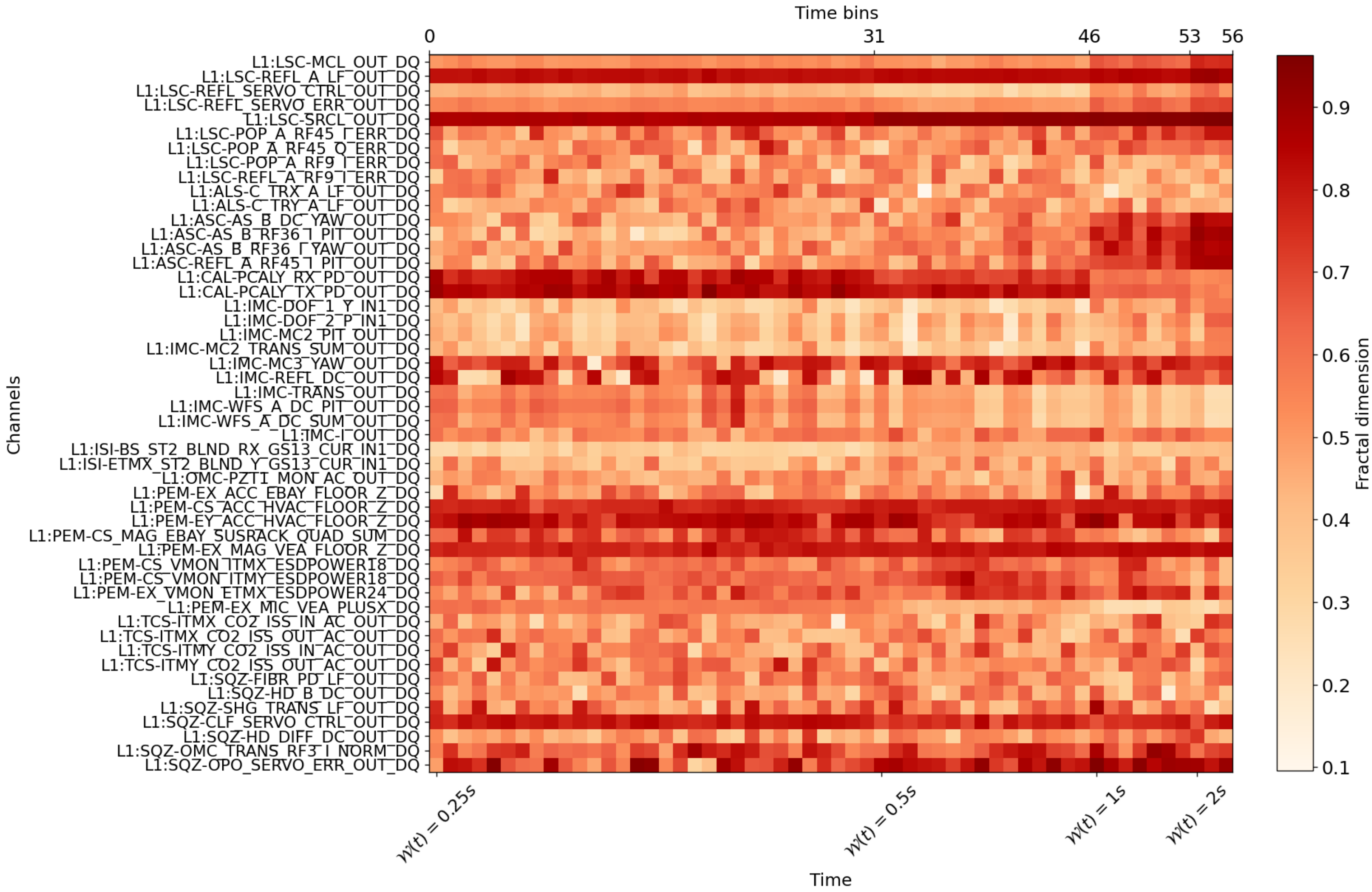


FIGURE 7-4: *A sample encoded in fractal dimension with 347 safe auxiliary channels and 56 time bins. Notably, the same event is replicated four times with a different time window: time bins* $\in [0, 31)$ *have a time window* $\mathcal{W}(t) = 0.25\,s$*, time bins* $\in [31, 46)$ *have* $\mathcal{W}(t) = 0.5\,s$*, time bins* $\in [46, 53)$ *have* $\mathcal{W}(t) = 1\,s$*, time bins* $\in [53, 56]$ *have* $\mathcal{W}(t) = 2\,s$*.*

7.3 Methodology

The novelty of this work lies in combining auxiliary channel information with ML in the context of anomaly detection. The complexity of the FD dataset implies two main challenges:

- *Lack of ground truth:* While glitch morphologies have been widely studied, there is no guarantee that all glitches belonging to a certain class present the same standard behaviour. In this work, the labels assigned by `Gravity Spy` are not considered ground truth and are used only for analysis and comparison purposes.
- *Lack of absolute ordering:* The ordering in the channels is arbitrary, as they measure different physical magnitudes. Consequently, it is not expected to find local patterns in the vertical axis with any particular channel ordering. Thus, our model needs to learn patterns beyond local correlations in an order-independent way.

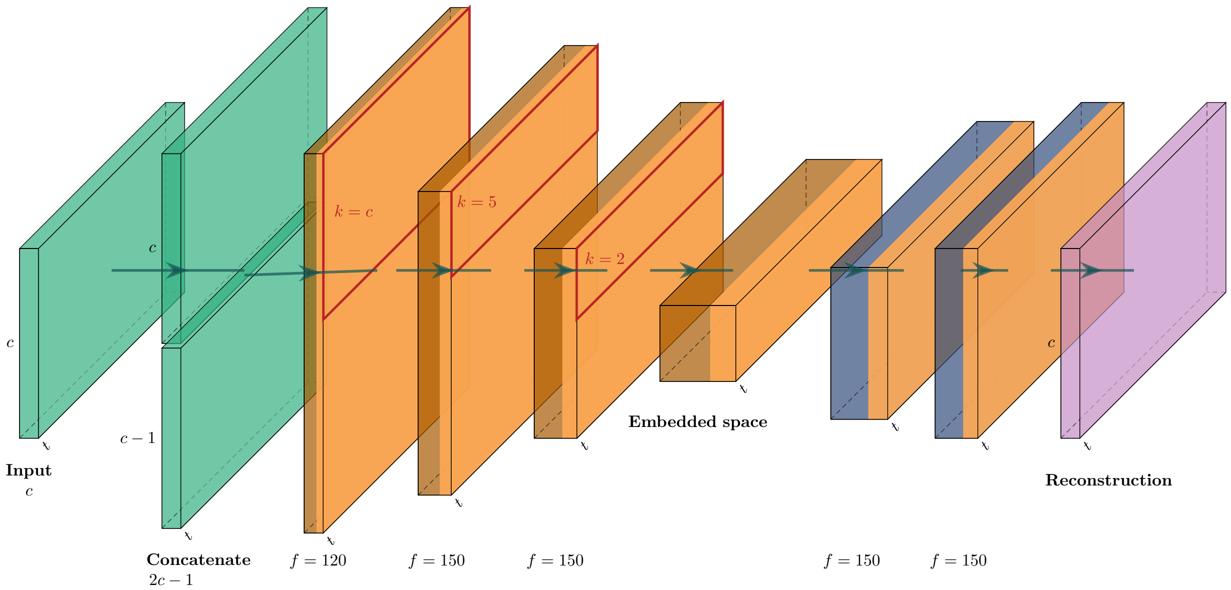


FIGURE 7-5: Outline of the autoencoder with periodic convolutions implementation. The input data which is concatenated yields a shape $(t, 2c-1)$ is presented in green. Downsampling convolutions are coloured in dark orange while upsampling convolutions are coloured in dark blue. Each convolution is followed by a Rectified Linear Unit (ReLU) activation function coloured in orange. This architecture yields a reconstruction of shape (t, c).

7.3.1 Tackling lack of ground truth: convolutional autoencoder

To address the concern of lack of ground truth, we employ an autoencoder in the context of anomaly detection. Autoencoders are a type of deep-learning algorithm known for their ability to uncover essential structures and patterns within unlabeled datasets, as well as their effectiveness in anomaly detection [295, 296]. They achieve this by compressing the data into a lower-dimensional or sparse format, known as an embedded space, maintaining the most relevant information from the dataset (encoding), and subsequently reconstructing it (decoding). The encoding is expected to employ important sub-structures that can be difficult to notice in the original representation space due to its higher dimensionality and feature redundancy [296]. As a consequence, the learned embedding serves as a reference for detecting irregularities in the glitch data, since data points that deviate significantly from their embedded representations

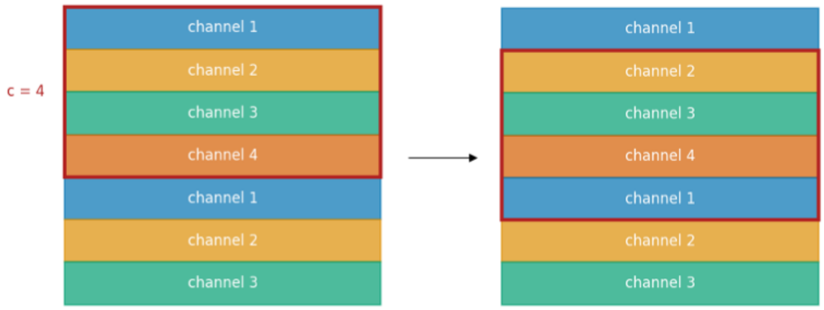


Figure 7-6: Example of periodic convolution where input of shape $(2c-1,t)$ is convolved with a kernel $k=c=4$. All cyclic channel permutations are convolved, so there is no absolute ordering.

are likely anomalous. Still, the embedded space can be hard to interpret itself, since it forms a high-dimensional space in which the data is densely packed.

When dealing with data with a natural order between the features, convolutional filters can be used to detect (hierarchically combined) local structures and patterns that allow a complex encoding model without an exponential increase of its parameters [297, 296, 298]. While our data is two-dimensional (time bins × auxiliary channels), given that we want to preserve the detail in the time dimension, we allow our model only to convolve along the channel dimension using 1-dimensional convolutions.

7.3.2 Tackling lack of absolute ordering: periodic convolutions

To address the concern of lack of absolute ordering and the possible lack of local patterns to exploit with limited range convolutional filters, we employ a periodic convolution with filters sized to cover all channels instead. We take inspiration from circular convolutions, which are used in the field of signal processing and consider the input signals as circular, or periodic, rather than finite, i.e. the end of the signal wraps around to the beginning, creating a cyclic or periodic nature [299]. In the context of this work, we use convolutional filters with a size equal to the number of channels, so that the filter has the opportunity to ignore the data's arbitrarily chosen channel ordering. The model applies these filters periodically, hence the name, so that learned filters can still be used by the model to encode structures and patterns found on different channel combinations.

The outline of the autoencoder's structure is presented in Fig. 7-5. Periodic convolutions were implemented with `tensorflow` [300] and `keras` [301] using a custom approach: given input with c auxiliary channels and t time bins, the input gets duplicated and concatenated along the channel axis, removing the last channel, as it is represented in Fig 7-6. In this way, each cyclic permutation of channels is seen only once. Thus, the dimensionality of the input fed into the model is $(c,t)+(c-1,t)=(2c-1,t)$, (see second block of Fig. 7-5 where the concatenation happens). To maintain the time dimension, we convolve along the channel dimension with a kernel size $k=c\times 1$, stride $s=1$, filters $f=120$ and no padding. This custom approach ensures that there is no specific spatial ordering in the vertical (channel) dimension and that the model can capture correlations between all the channels.

Once the first convolutional layer has captured the data patterns between arbitrary channels, the rest of the encoder architecture has the goal of constraining the data into an embedded space. Thus, it consists of two conventional downsampling convolutional layers with kernel sizes $k=5\times 1$ and $k=2\times 1$ with $f=150$, respectively, represented in Fig. 7-5 in dark orange. The decoder structure is a mirror of the encoder, but with up-sampling convolutional layers instead, coloured in blue in Fig. 7-5. The resulting embedded space has a dimensionality of 5×56 with $f = 150$, which we will use to detect anomalies.

After each convolutional layer a Rectified Linear Unit (ReLU) [302] activation function is employed to introduce non-linearity in the model (see Fig. 7-5 in orange). The ReLU activation function avoids vanishing gradients [303]. The model was trained for 500 epochs and a batch size of 168, using the Adam optimizer [304], with a learning rate $l_r = 10^{-3}$. The loss function employed is the Mean Squared Error (MSE) loss, which represents the cumulative squared error between the input and its reconstruction [305]. To assess the performance of the model, we use the reconstruction error ϵ_R which is defined as follows:

$$\epsilon_R = \sum_{k=0}^{c-1} \sum_{l=0}^{t-1} (D_{i,kl} - D_{r,kl})^2, \tag{7.4}$$

where D_i is the input to the model and D_r its reconstruction [306]. We expect that lower reconstruction errors translate to more accurate anomaly identification.

7.3.3 t-distributed stochastic neighbour embedding

As discussed in the previous section, the embedded space is still high dimensional and difficult to interpret. Hence, to further lower the data dimensionality and make the data distribution in the embedded space easier to interpret, we use the t-distributed stochastic neighbour embedding (t-SNE) method[1], which projects high-dimensional data in a low-dimensional space, preserving local relationships between data points and underlying structure of the data, but releasing global relationships between data points [307]. While the t-SNE variables do not have an interpretable physical meaning, they are a linear correlation of physical variables and they can be traced back to assess their contribution.

After obtaining the 2-dimensional projection of the embedded space with t-SNE, it is straightforward to visualize the distribution of the different data points, which correspond to the glitch instances, as we show in the next section (see Fig. 7-8). The 2-dimensional plot is expected to reveal different clusters and interesting structures in the data since data points that are distant from the main clusters of their predicted class are anomalies. By labelling each glitch with its corresponding `Gravity Spy` label with confidence $> 90\%$, outliers and new glitch morphologies are expected to be identified.

7.4 Results

To assess the performance of the autoencoder presented in the previous section, we show the achieved reconstruction errors $\epsilon_R \in [0.001, 0.014]$, as seen in Fig. 7-7. The three glitch classes present similar distributions, with the `Scattered_Light` class reaching the highest reconstruction error $\epsilon_R = 0.014$. The reconstructed input differs from its original on 1.75% and 17.5% in the best and worst reconstructed pixels, respectively. Note that the model reconstructs 98.8 % of glitches with $\epsilon_R < 0.002$, which is a low error given the range $[0.0, 1.0]$ of the input data. In general, it appears that pixels from the same auxiliary channel have similar reconstruction errors, which translates to the preservation of FD-encoded data structure within the given auxiliary channel.

In Fig. 7-8 we present joint and marginal distributions of the t-SNE projections with three different representations of the data: the original dataset with 347 auxiliary channels, the reduced dataset with 50 auxiliary channels (see Section 7.2.4), and the embedded space with shape 5×56 with $f = 150$. These t-SNE representations cluster the input data in different

---

[1] `TSNE` function from `scikit-learn` library [275] was employed.

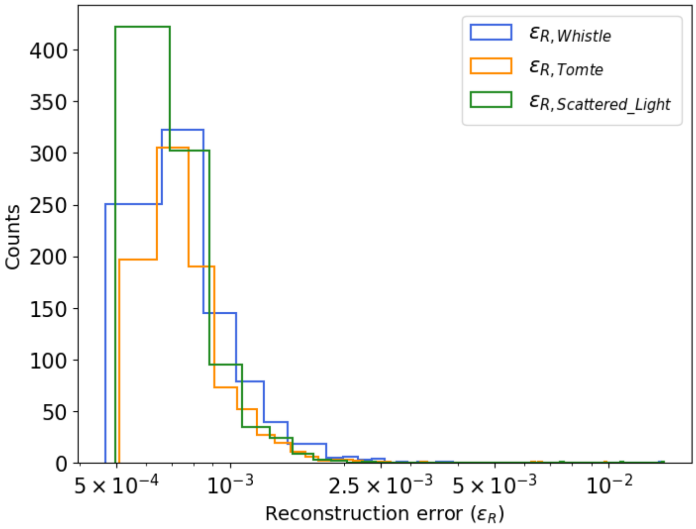


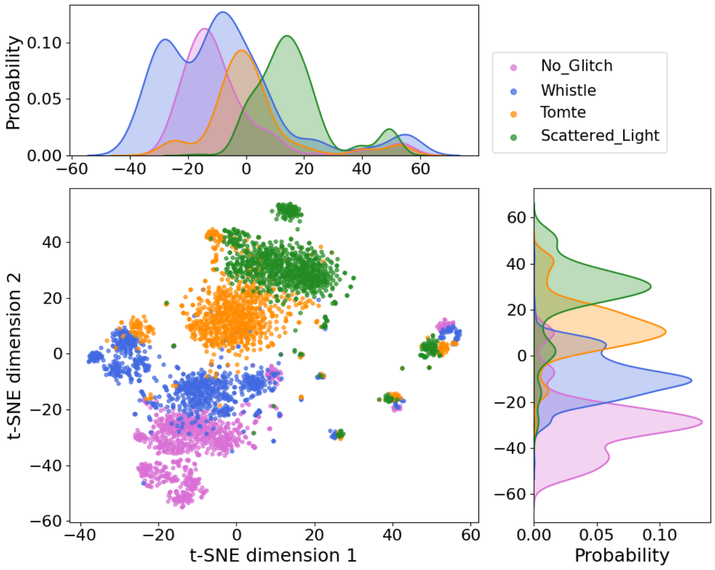
FIGURE 7-7: Histogram with the reconstruction errors ϵ_R for each glitch class in logarithmic scale.

regions of the space, such that the samples in the out-skirts will be considered anomalies. We use the labels of `Gravity Spy` to track in which regions of t-SNE space the different classes fall.

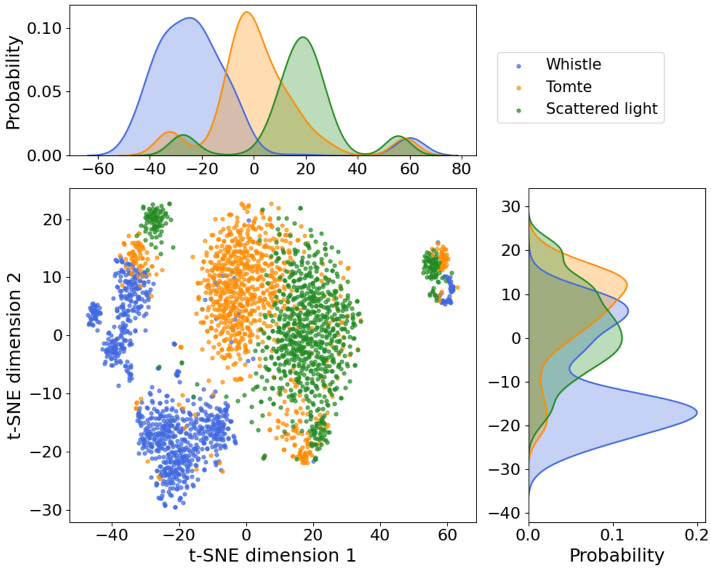
In Fig 7-8a, the t-SNE projection of the original FD-encoded data with 347 safe auxiliary channels clusters the glitch by similarity, which is consistent with `Gravity Spy`'s classification revealing some overlap between classes, especially between `Tomte` and `Scattered_Light`, as well as between `Whistle` and `No_Glitch`. A dataset with instances of dimensions 347×56 introduces an enormous complexity. Therefore, we reduce the dimensionality of the data, using the safe auxiliary channels that show the most variance in the FD-encoding, which is related to the presence of glitches (see Section 7.2.4). Such reduction in dimensionality yields a more compressed representation of the input data and less overlap between the glitch classes, as can be seen in Fig. 7-8b since there are fewer sub-clusters of each class. We use this reduced representation to train the autoencoder on `Tomte`, `Scattered_Light` and `Whistle`, as we are interested on learning the main FD-characteristics that may be related to these glitches. After training it, which yields small reconstruction errors ϵ_R, we can project its embedded space in 2 dimensions with t-SNE, as we can see in Fig. 7-8c. We can observe that it looks similar to the reduced t-SNE in Fig. 7-8b, but its marginal distributions seem smoother since the model is learning the general trend of local and global correlations of the FD-encoded data. While at each compression we are discarding some characteristics of the data, we are maintaining the general trend of the data which, as stated before, is consistent with what is observed by `Gravity Spy` in $h(t)$, the main strain of the data. We remind the reader that the labels in Fig. 7-8a, 7-8b and 7-8c are classifications of `Gravity Spy` with confidence $> 90\%$.

To explore the embedded space, in Fig. 7-8c, we have manually outlined the distinct clusters. In the following, we present the time-frequency of the strain $h(t)$ of some anomalous examples found in the outskirts of these clusters solely employing safe auxiliary channels. The reader will encounter glitch classes that have not been mentioned before in the present work, but their description can be found in [67].

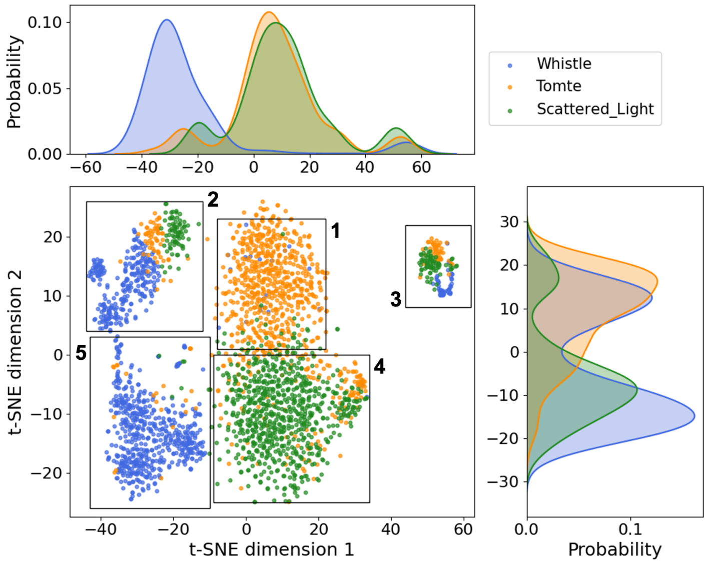
- **Region 1** corresponds to the main `Tomte` cluster. However, some `Whistle` and `Scattered_Light` labels are also present. From this region, in Fig. 7-9a we show a glitch classified as `Whistle` but with an anomalous morphology, and in Fig. 7-9c we show a misclassified Wandering Line glitch classified as a Whistle. In Fig. 7-9b we present a glitch classified



(A) *Original t-SNE:* projection of FD-encoded data with 347 auxiliary channels.



(B) *Reduced t-SNE:* projection of FD-encoded data with 50 auxiliary channels



(C) *Embedded t-SNE*: projection of the embedded space of the autoencoder showing five regions of interest.

FIGURE 7-8: *Joint and marginal distributions of t-SNE projections of the data. Legends correspond to* `Gravity Spy` *classification with confidence* $> 90\%$

as `Scattered_Light` but its morphology is similar to Scratchy, and in Fig. 7-9d we can see a `Scattered_Light` with an anomalous morphology.

- **Region 2** is a sub-cluster of `Whistle`, with `Tomte` and `Scattered_Light` overlaps. In Fig. 7-10a, we see a `Tomte` glitch that is revealed to be overlapping with a `Scratchy` glitch in a longer time window. In Fig. 7-10c we see a `Tomte` glitch overlapping with a smaller `Tomte` glitch. Fig. 7-10b shows a Fast Scattering glitch mislabelled as `Scattered_Light` and Fig. 7-10d presents a `Scattered_Light` overlapping with an unknown morphology.

- **Region 3** is a cluster with anomalous glitches from the 3 different glitch classes. Examples of anomalies are presented in Figs. 7-11a and 7-11d, which are labelled as `Whistle` but have distinct morphologies that differ from any of the 22 `Gravity Spy` classes. Another anomalous glitch from this region is the misclassified glitch shown in Fig. 7-11b, which was labelled as `Tomte` but has a morphology consistent with Fast Scattering. Another example of an anomalous glitch from this region presents as an overlap as shown in Fig. 7-11c. The glitch was labelled as `Scattered_Light` but seems to be a `Tomte` overlapping with an unknown morphology.

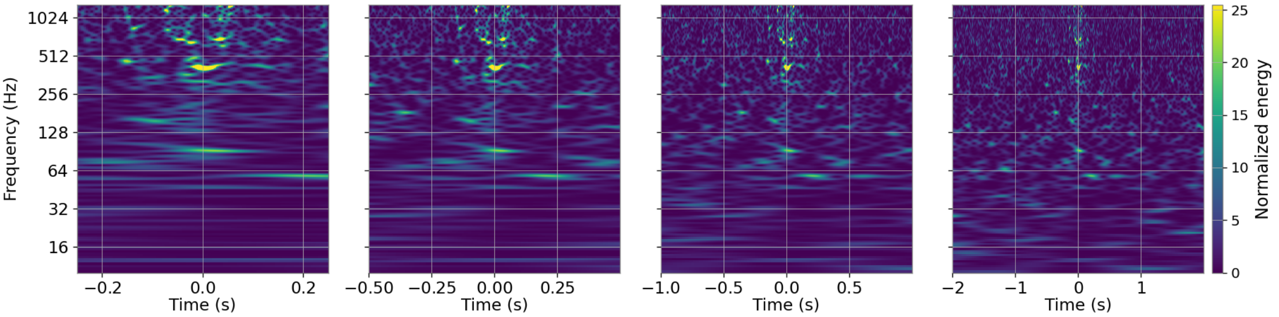
- **Region 4** corresponds to the main `Scattered_Light` cluster. In this region, there is a high presence of `Tomte` labels, which could indicate that both physical processes are related. In Fig. 7-12a, we present a glitch from this region that was labelled as `Tomte` but is consistent with the `Koi_Fish` class, while 7-12b was also labelled as a `Tomte` but seems to be an overlap between `Tomte` and Scratchy.

- **Region 5** corresponds to the main `Whistle` cluster, where we find some `Scattered_Light` labels as well as a few `Tomte` labels. In Figs. 7-13a and 7-13c we see `Tomte` glitches that appear to be overlapping with `Scratchy` glitches, in Fig. 7-13b we see a glitch labelled as `Scattered_Light` but could be a novel morphology, and in Fig. 7-13d we see a `Scratchy` glitch misclassified as `Scattered_Light`.

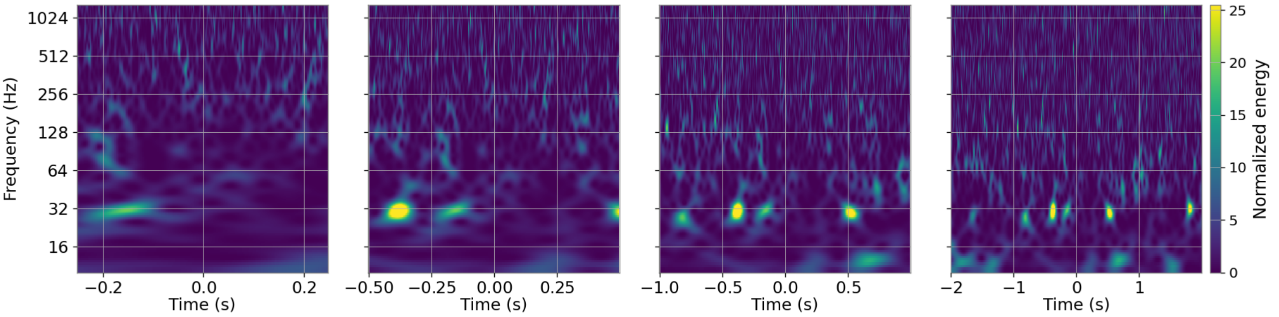
The outliers found at the outskirts of their clusters are visually and manually selected from the t-SNE representation in Fig. 7-8c, to automate the procedure in future works. After outliers have been selected, their spectrograms in $h(t)$ are visually inspected by comparing them to standard `Gravity Spy` morphologies. With this procedure, a total of 177 anomalies were found out of 2688 samples, which implies 6.6% of the data. In particular, for each class, we found:

- `Whistle`: 49 anomalies were found, 45% are unknown morphologies, 28% are `Gravity Spy` misclassifications, and 27% are glitch overlaps.
- `Tomte`: 57 anomalies were found, 32% are unknown morphologies, 21% are misclassifications, and 47% are glitch overlaps.
- `Scattered_Light`: 71 anomalies were found, 28% are unknown morphologies, 72% are misclassifications, and only one case of overlap is found.

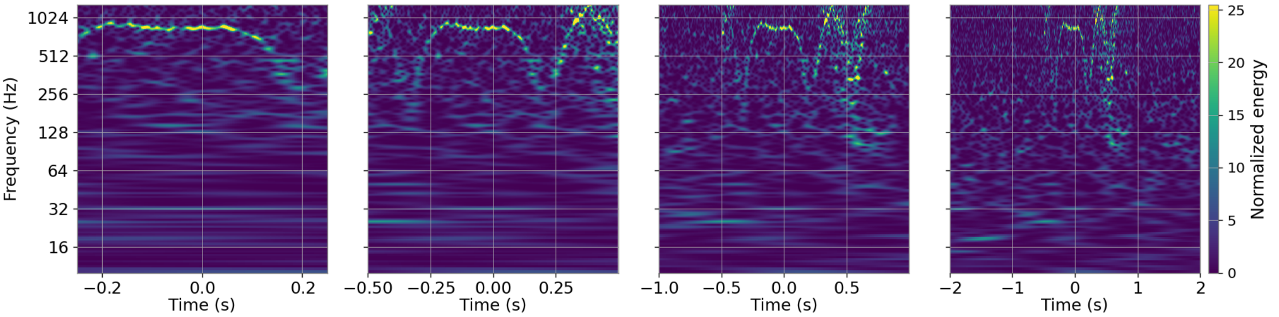
After a visual inspection, we found that for `Whistle` most outliers constitute unknown morphologies, while for `Tomte` most anomalies are due to overlaps, where it is common that two



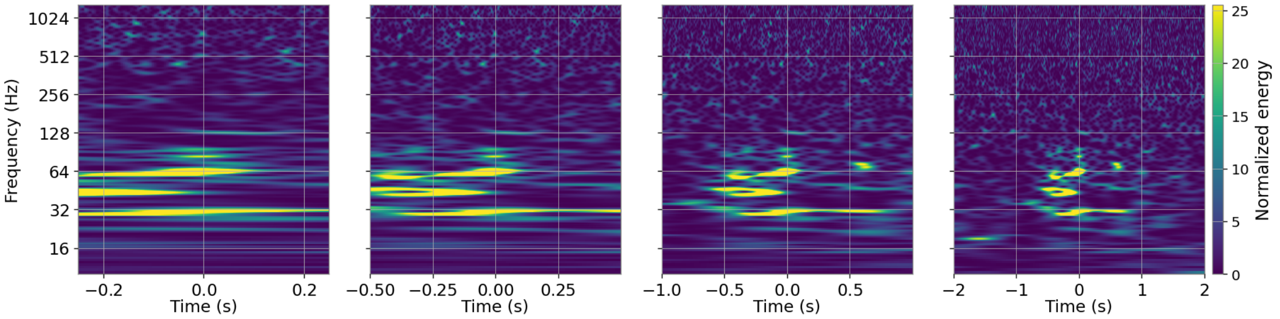
(A) Glitch labeled as `Whistle`, but appears to be an anomalous morphology.



(B) Glitch labeled as `Scattered_Light`, but its shape is consistent with the `Scratchy` class, being a misclassification.



(C) Glitch labeled as `Whistle`, but its shape is more consistent with `Wandering_Line` class, constituting a misclassification.

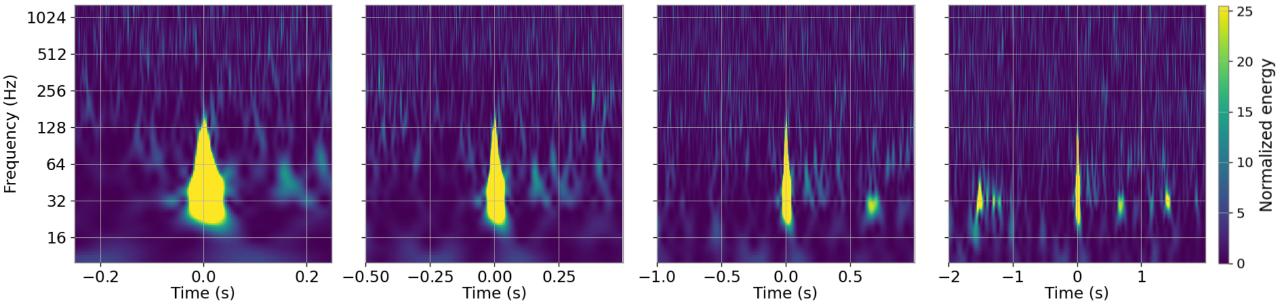


(D) Glitch labeled as `Scattered_Light`, but appears to be an anomalous morphology.

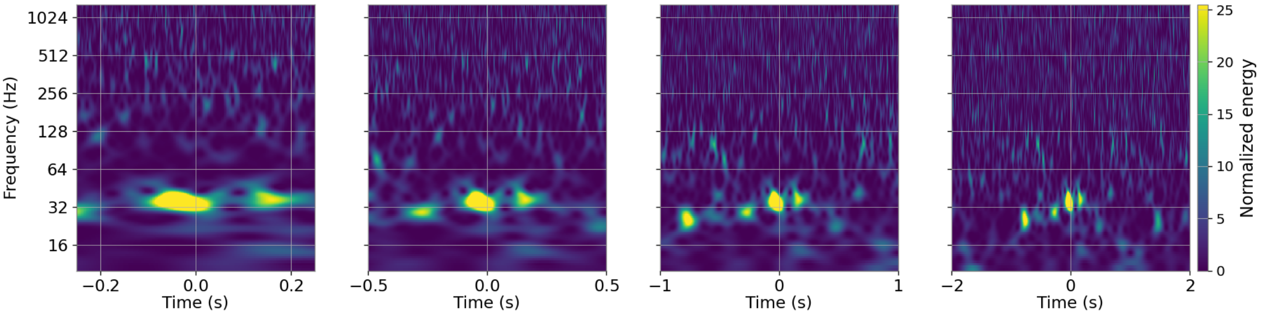
Figure 7-9: ***Region 1:*** *Example of anomalous glitches in the embedded space.*

`Tomte` happen simultaneously. For `Scattered_Light`, most outliers correspond to misclassifications, since the other seven glitch classes happen at similar frequency intervals and duration periods, namely `Low_Frequency_Burst`, `Low_Frequency_Lines`, `Power_Line`, `Scratchy`, `Air_Compressor`, `Paired_Doves` and `Fast_Scattering`.

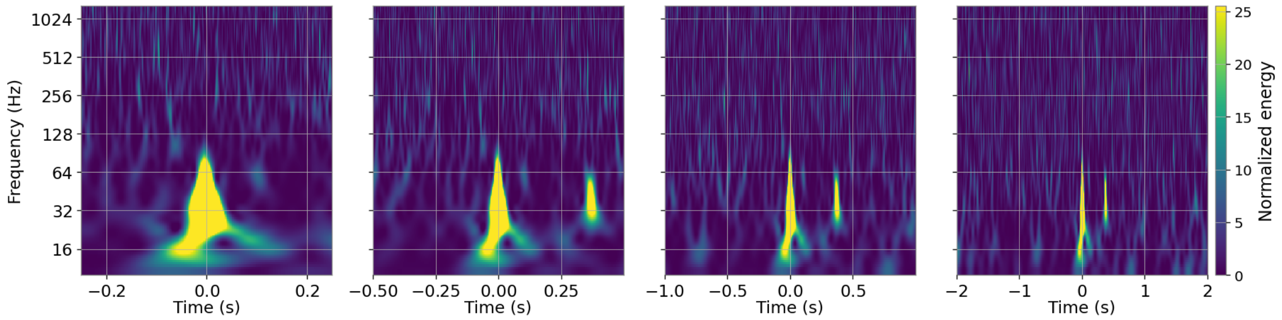
While the misclassification of glitches could be countered with the improvement of training strategies, data set construction or class definitions, the identification of anomalies arising from



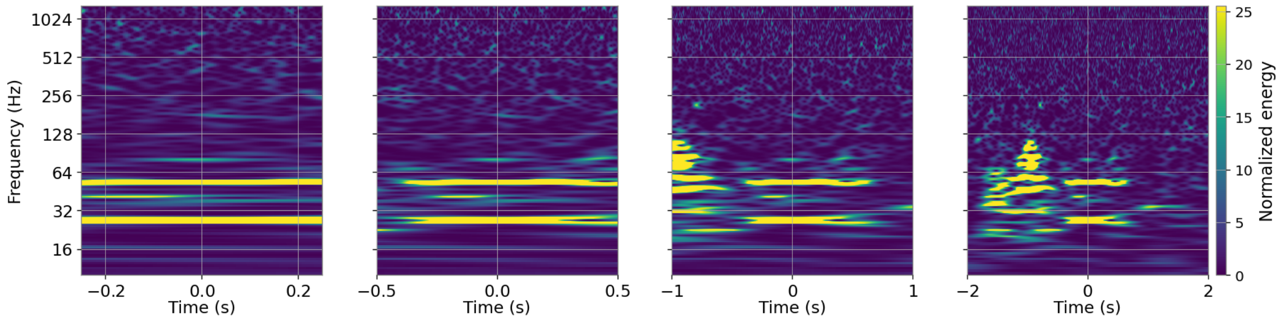
(A) Glitch labeled as `Tomte`. On the spectrogram with the longer time window, we can see an overlap between a `Tomte` and `Scratchy` glitch.



(B) Glitch labeled as `Scattered_Light`. It is consistent with the `Fast_Scattering` morphology, being a misclassification.



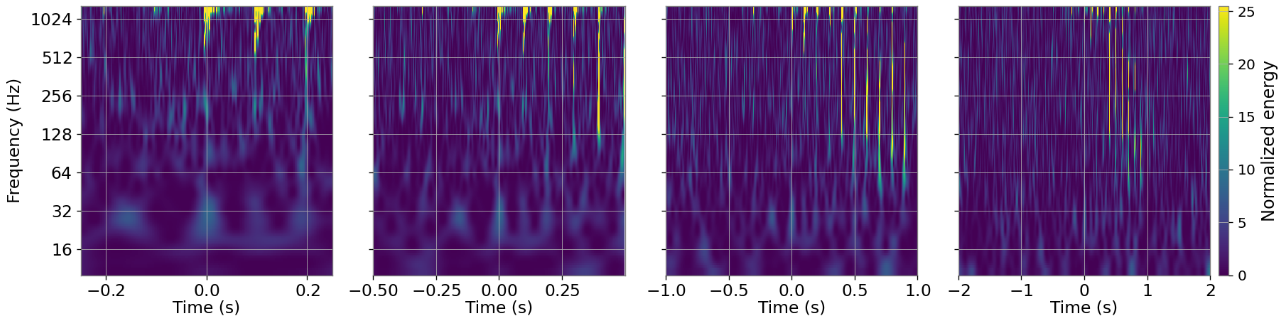
(C) Glitch labelled as `Tomte`, overlapping with a smaller `Tomte`.



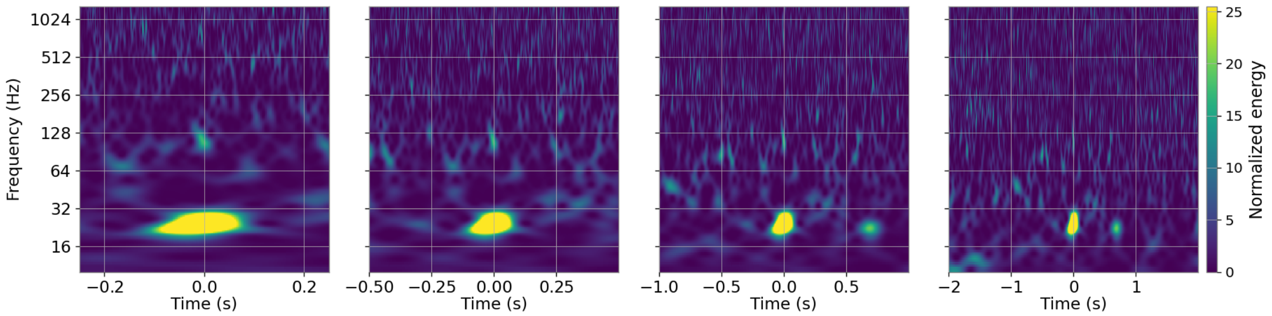
(D) Glitch labeled as `Scattered_Light`. On the left plot, an unknown morphology is overlapping at the beginning of the `Scattered_Light`.

Figure 7-10: ***Region 2:*** *Example of anomalous glitches in the embedded space.*

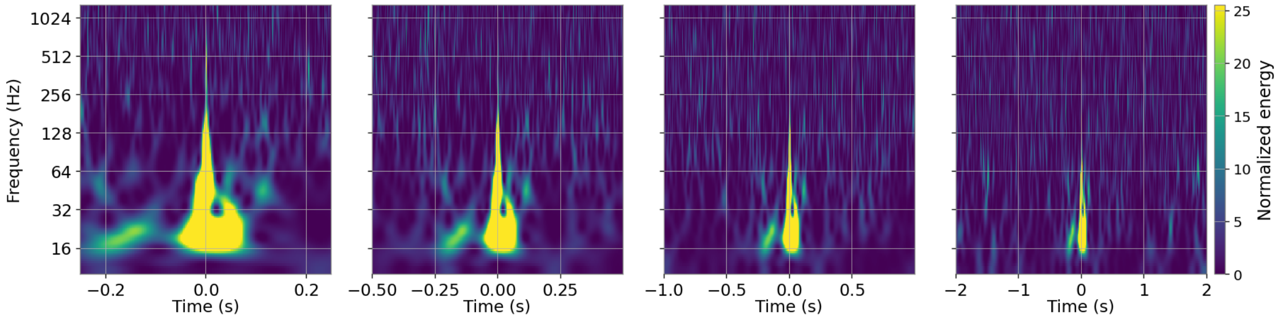
overlaps and novel morphologies would still be hampered by the strict class definitions from supervised methods. Therefore, unsupervised approaches, such as the one presented in this work, will improve the understanding of glitch populations for their subsequent mitigation.



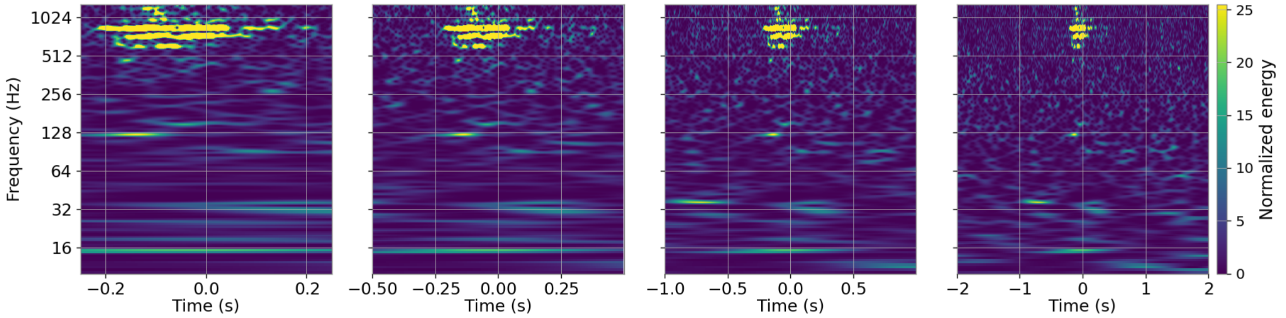
(A) Glitch labeled as `Whistle`, but it was not observed on other `Gravity Spy`'s classes, so it could be a novel morphology.



(B) Glitch labeled as `Tomte`, but its shape is consistent with `Fast_Scattering`, being a misclassification.



(C) Glitch labeled as `Scattered_Light`, but its shape is consistent an overlap between a `Tomte` and an unknown morphology.

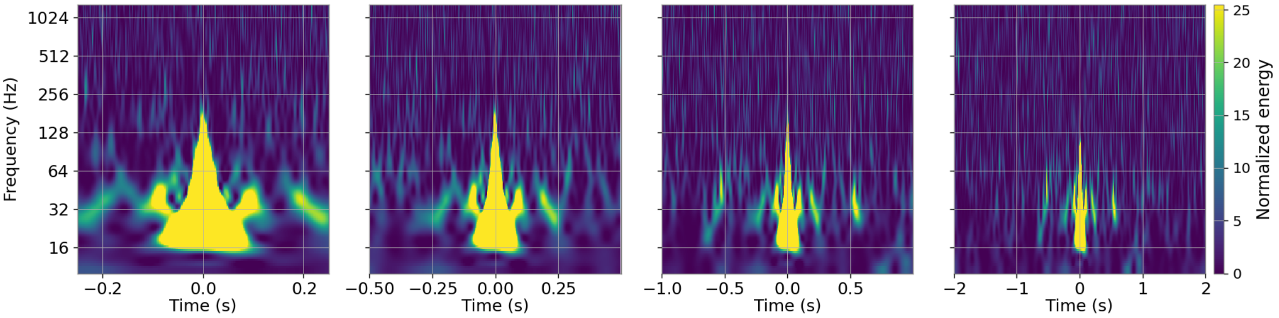


(D) Glitch labeled as `Whistle`. The observed shape is inconsistent with any `Gravity Spy` classes, constituting an anomalous glitch morphology.

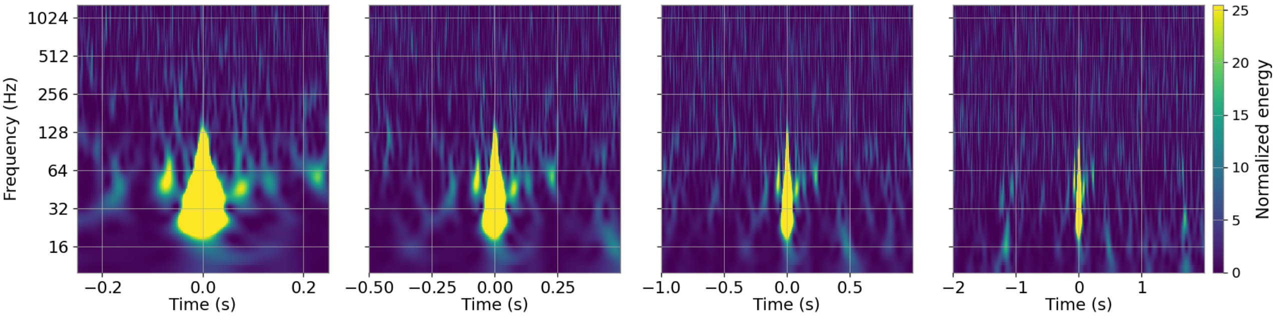
Figure 7-11: ***Region 3:*** *Example of anomalous glitches in the embedded space.*

7.5 Discussion

In this chapter we have performed an exploratory analysis of a reduced set of safe auxiliary channels from LIGO Livingston with FD-encoding in the context of anomaly detection. The focus of this work is, on one hand, to explore the potential of this data representation in the



(A) Glitch labeled as `Tomte`, but consistent with `Koi_Fish` class, being a misclassification.



(B) Glitch labeled as `Tomte`, but seems to be an overlap between a `Tomte` and a `Scratchy`, being an overlap.

Figure 7-12: ***Region 4:*** *Example of anomalous glitches in the embedded space.*

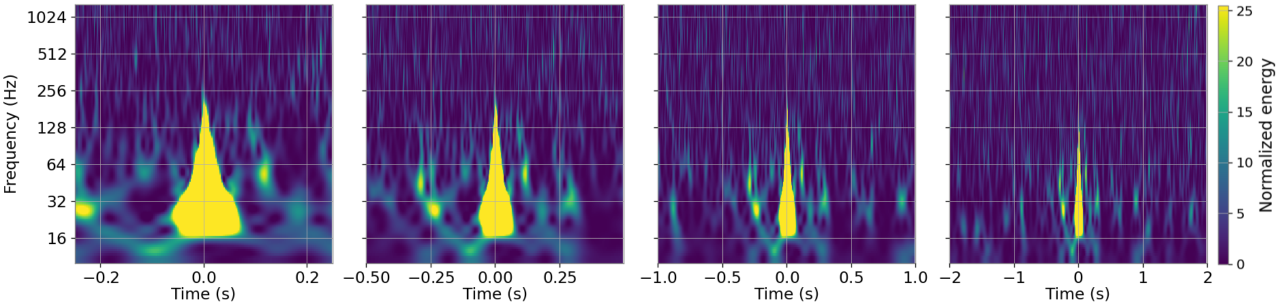
context of glitch characterization, and on the other hand, to build a data-driven model to cluster glitches in an unsupervised way with direct information from the detector, finding anomalies that deviate from the general distribution of the data.

For this aim, we first speeded up the FD calculation from a computational complexity of $\mathcal{O}(N^3)$ in [288] to $\mathcal{O}(N^2 \log N)$, constructing the FD-encoded safe auxiliary channel data set. Afterwards, we implemented a periodic convolutional autoencoder to learn the local and global structure of the data, compressed in a lower-dimensional space, known as embedded space. The reconstruction errors of the output of the autoencoder were $\sim 98.8\%$ of glitches < 0.002, implying that the autoencoder was able to learn the general trend of the data.

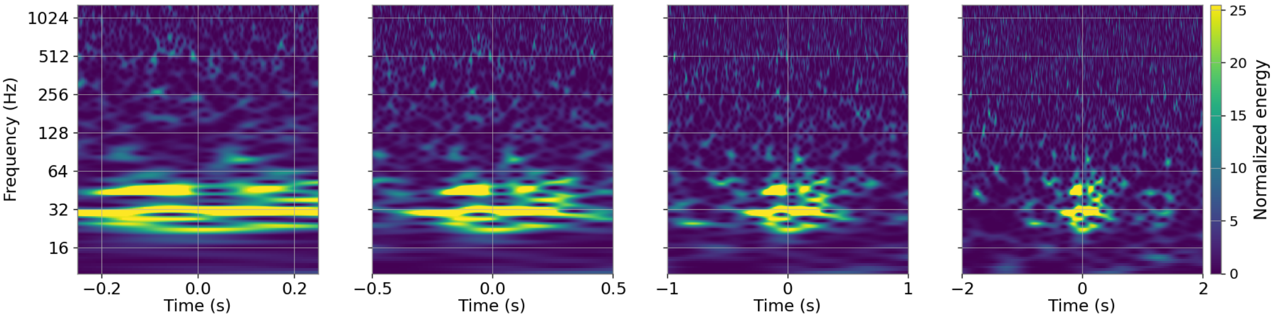
We can also observe the reliable compression of the autoencoder, using solely safe auxiliary channels, when we project the embedded space in a two-dimensional t-SNE. This t-SNE representation clusters the different classes in separate regions which are consistent with `Gravity Spy`'s observation in the main detector strain, $h(t)$. Samples that deviate significantly from their closest cluster are considered outliers. Representing these outliers in $h(t)$, we observed novel morphologies that strongly deviated from the standard definitions of `Gravity Spy`.

This methodology has shown that the safe auxiliary channel in the FD-encoding acts as a complementary representation to the visualization of $h(t)$, used to characterize the noise of the detector and to identify glitches for their subsequent mitigation. Furthermore, our algorithm is flexible and completely data-driven, capable of uncovering misclassifications, glitch overlaps and novel glitch morphologies. While our method is independent of supervised classification algorithms, we used `Gravity Spy` as a benchmark to quantify its performance: in our FD-encoded auxiliary channel data, constituted by 2688 times where glitches were present in $h(t)$, we found a 6.6% of anomalies caused by unknown morphologies labelled as their closest glitch class, similar morphologies assigned the incorrect class or glitch overlaps being overlooked.

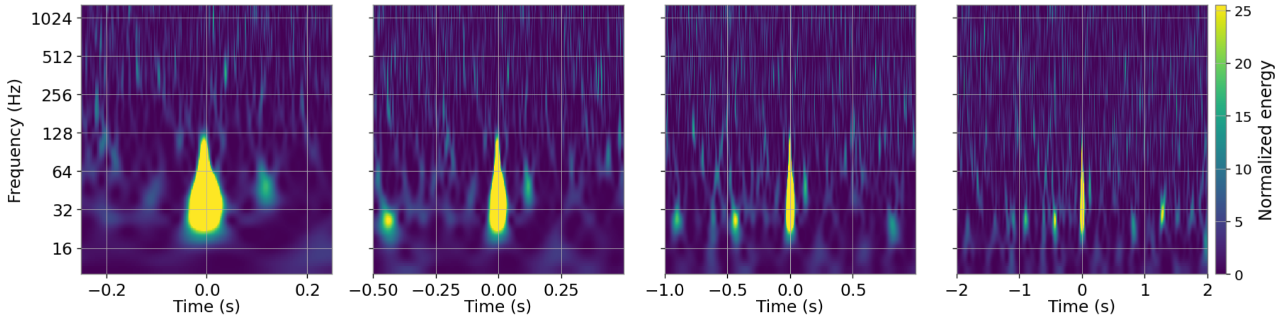
Up to the date of publication of this manuscript, we are not aware of another investigation that uses unsupervised learning with auxiliary channels.



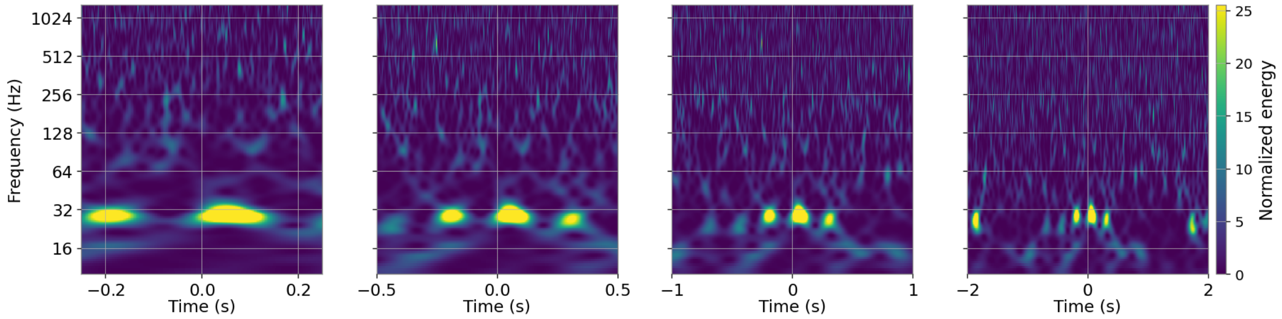
(A) Glitch labeled as `Tomte`, but it could be an overlap between `Scratchy` and `Tomte`.



(B) Glitch labeled as `Scattered_Light`, but it was not observed on other `Gravity Spy`'s classes, so it could be a novel morphology.



(C) Glitch labeled as `Tomte`, but it could be an overlap between `Scratchy` and `Tomte`.



(D) Glitch labeled as `Scattered_Light`, but its morphology is consistent with `Scratchy`, being a misclassification.

Figure 7-13: ***Region 5:*** *Example of anomalous glitches in the embedded space.*

7.6 Conclusion

As we have discussed in the previous sections the FD is a powerful tool to encode the information glitches from the auxiliary channels in a lower dimensional space. Additionally, our approach of integrating the latent space of the autoencoder with t-SNE offers an even more intuitive, lower-dimensional representation for interpreting the "glitch space". Indeed, our unsupervised approach discovers novel morphologies and glitch overlaps that are disregarded by static supervised methods.

Data-driven approaches, such as the one demonstrated in this work, can unveil anomalies present in the data and reveal relations between glitch classes, allowing us to further understand the glitch population. However, despite its strengths, our approach has certain limitations. In this proof-of-concept study, we focus on only three glitch classes out of the 22 well-known categories, which limits the generalizability of our findings. Additionally, although we selected channels based on the most significant FD variance, this selection differs from the standard set of channels typically used by LIGO experts. This discrepancy poses challenges for explainability, making it more difficult to relate our results to the underlying physical processes that LIGO experts are familiar with.

In future work, this approach will be extended to the general population of LIGO-Livingston and other interferometers to enhance the identification of glitches. Moreover, we will provide an anomaly score to assess the significance of the outliers found by our algorithm and explore a data fusion representation containing both the FD-encoded auxiliary channel data and the strain $h(t)$ in time-frequency representation, providing not only information about the physical process within the detector but also their impact on $h(t)$. Last but not least, we will investigate the correlation between safe auxiliary channels highlighted by our model and glitches appearing in $h(t)$, in the search of witness auxiliary channels to improve glitch mitigation in GW searches.

Chapter 8

Enhancing Unmodelled Core-Collapse Supernova Searches

This chapter is based on work presented in Ref. [308] and its companion, Ref. [309], where I led the construction of the models, the performance evaluation and the post-processing. I also was one of the lead writers in [308], and I was the lead writer in [309].

8.1 Introduction

High-energy cosmic neutrinos, as well as GW emission, are two novel messengers of astronomical sources. These new messengers, along with electromagnetic radiation and cosmic rays, give new insights into the most extreme energetic cosmic events. Among them, supernova explosion is one of the most challenging targets of this new astronomical approach.

While GW have been detected from mergers of binary black holes (BBH) and binary neutron stars (BNS), core-collapse supernovae (CCSN) have not yet been detected and they still represent a puzzle to solve. We had confirmation of the basic CCSN theory through the detection of MeV neutrinos from the SN1987A [243]: the collapse of a massive star's core is driven by the release of gravitational energy and the vast majority of this energy is realised in neutrinos. However, the details of the mechanism of the explosion are still an open question and the astronomical community is trying to disentangle it.

Massive stars ($M > 8M_{\odot}$) spend most of their lives burning hydrogen into helium, which settles in the core and, when temperatures increase sufficiently, burns into heavier nuclei until the iron is reached. Having reached the Chandrasekhar mass (≈ 1.4 $\mathrm{M}_{\odot}$) the iron core cannot support its weight and undergoes a gravitational collapse.

The slow contraction of the growing and ageing iron core speeds up when its central temperature ~ 1 MeV. At this stage, thermal γ photons become sufficiently energetic to partially disintegrate the iron-group nuclei to α-particles and free nucleons. At such high densities, the cross-section of the interaction of neutrinos with matter becomes large, such that neutrinos become trapped in the core. Within milliseconds, the centre reaches nuclear matter density, where the heavy nuclei dissolve in a phase transition to a uniform nuclear medium, and neutrinos can only escape by diffusing out of the neutron star rather.

As the density increases up to nuclear saturation density, heavy nuclei are disintegrated into free nucleons, producing neutrinos that become trapped. At the same time, the sharp rise of the incompressibility, due to repulsive contributions to the nuclear force between the nucleons, halts the collapse of the inner core forming a proto-neutron star. During the collapse, electrons are absorbed by nucleons through beta processes, reducing electron degeneracy pressure and accelerating the collapse. The bounce occurs when nuclear interactions between nucleons, primarily neutrons, become the dominant force. As the core rebounds, it produces a shock wave that stalls at around 100 km from the centre. Helped by the additional thermal energy deposited by the neutrinos diffusing out of the proto-neutron star, the shock may revive in timescales of hundreds of milliseconds disrupting the entire star and producing an electromagnetic signal known as supernova [310, 242].

This so-called neutrino-driven mechanism [241], is the dominant theory to explain CCSN explosions in slowly rotating progenitors. Observationally only $\sim 1\%$ of the events show signatures of fast rotation (broad-lined type Ic supernovae [311] or long gamma-ray bursts [312]), neutrino-driven explosions are likely the most common type of CCSN, and our focus of this chapter.

In a supernova explosion, GWs are generated in the inner core of the source, so that this messenger carries direct information of the inner mechanism. The feasibility of this scenario will be supported by the joint observation of neutrino and GW emission from CCSN, by assessing the correlation between neutrino emission and collapsed core motion. Although the phenomenon is among the most energetic in the universe, the amplitude of the GW impinging on a GW detector is extremely faint. For a CCSN in the centre of the Milky Way, a rare event, we could expect amplitudes of the metric tensor perturbations in $\sim [10^{-21} - 10^{-23}]$, unlike BBH or BNS. To increase the detection probability we should increase the volume of the universe to be explored, which can be achieved both by decreasing the detector noise and using better-performing algorithms.

The impossibility of using template-matching techniques in this case, due to the complexity and stochasticity of the waveform, makes it necessary to find new ways to improve the detection statistics. As we have seen in Section 3.3, current efforts to search for GW from CCSN include targeted searches for observed nearby supernovae [313, 314] and all-sky generic searches for bursts [20, 21]. For the latter two independent pipelines are used: coherent Waveburst (cWB) [315] and omicron-LIB (oLiB) [127], while BayesWave [135] is a followup of cWB GW candidate events. These searches use algorithms based on excess power to identify signals buried in the detector's noise without taking advantage of any specific feature of CCSN waveform.

In [316] it has been proposed the use of ML techniques to take advantage of the peculiarities of the CCSN GW signal to increase our detection capability with respect to current methods. In particular, the focus was on the monotonic rise of the GW signal in the time-frequency plane due to the g-mode excitation, which is the dominant feature present in the GW spectrum. A similar approach has been followed recently by [317, 212, 214, 318] and in general there has been an increasing interest in the GW community for the use of ML methods (see Section 4.3).

In this chapter, we follow a similar approach as in [316], labelled as *previous work*. The main differences are:

- the use of a more sophisticated CNN;
- the injection of simulated CCSN signals in real noise of the three advanced detectors of the LIGO-Virgo network, as measured during August 2017 during O2 ([316] only considered Gaussian noise);
- the improvement of the phenomenological templates used during the training of the CNN network to better match results from numerical simulations.

8.2 Data

In the next Section, we describe our newly improved phenomenological waveform templates that are used to train the CNN. For a comparison of performance between the previous work in [316] and the present work, we use the same data set of [316], labelled as *previous set* (see Section 8.2.3), for training and validating of the new neural network architectures. Afterwards, to tune our CNN we train the algorithm with the new phenomenological templates injected in the real noise (see Section 8.2.4). Finally, we test the network with injections of phenomenological waveforms (see Section 8.2.5) and waveforms from CCSN numerical simulations (see Section 8.2.6).

8.2.1 Phenomenologial waveforms

As in the case CBC phenomenological waveforms (see Section 3.2.2), we consider a parametric phenomenological waveform designed to match the most common features observed in the numerical models of CCSN. We focus our attention on the g-modes excitation, or gravity modes, which are induced by a buoyancy restoring force [319]. This is the most common feature of all numerical models developed so far to describe the CCSN phenomena, responsible for the bulk of the GW signal in the post-bounce evolution of the proto-neutron star. The measurement of the g-mode has been proposed as a way of inferring the properties of the proto-neutron star [see e.g. 320, 321, 322, 323].

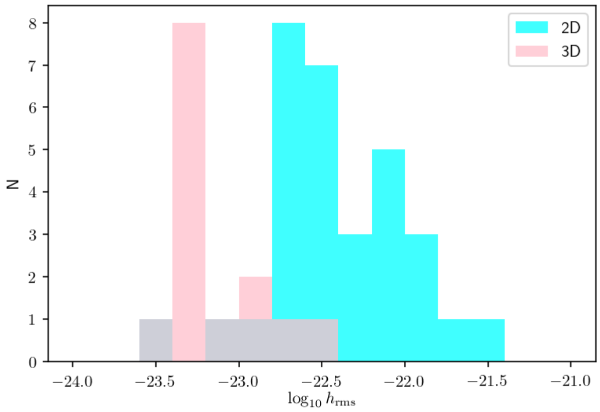


FIGURE 8-1: Number of simulations with a given g-mode root mean square (rms) strain at 10 kpc (per logarithmic interval) for 2-dimensional (blue bars) and 3-dimensional (red bars) simulations in the waveform calibration set.

Our phenomenological template aims to mimic the rising arch observed in CCSN simulations. To this end, we will consider a damped harmonic oscillator with a random forcing, in which the frequency varies with time. The phenomenological templates we used differ from the ones in [316] in two aspects: we use a new and more flexible parametrization for the frequency evolution and we use the distance as a parameter. The phenomenological templates are calibrated to mimic the features in the numerical simulations for non-rotating progenitor stars by [324, 325, 326, 327, 328, 320, 329, 330, 331], named *waveform calibration set*, hereafter.

The new parametrization describes the evolution of the frequency of the g-modes, $\nu(t)$, as a cubic spline interpolation to a series of discrete points, (t_i, ν_i), where t_i corresponds to post-bounce times. Given the relatively simple behaviour of $\nu(t)$ observed in numerical simulations, it is sufficient to use three points with $t_i = (0, 1, 1.5)$ s. ν_0, ν_1, and ν_2 are then three parameters of the template.

In [316] the amplitude of the generated waveforms has been chosen according to the SNR. In this work, we want to go one step further and use distance as a parameter for the waveform generator. To do that we relate the amplitude of the waveform with its distance using the data in the waveform calibration set. First, we need to measure the typical strain of the g-mode component for each simulation.

To this aim, we apply a high pass filter at 200 Hz, and then we use the section of the waveform containing 99% of the waveform energy to compute the root mean square (rms) value. This procedure filters out signal power at lower frequencies related to other effects different to g-modes (standing-shock accretion instabilities [332], prompt convection [333] and large-scale asymmetries due to shock propagation [334]) that are not considered in this work.

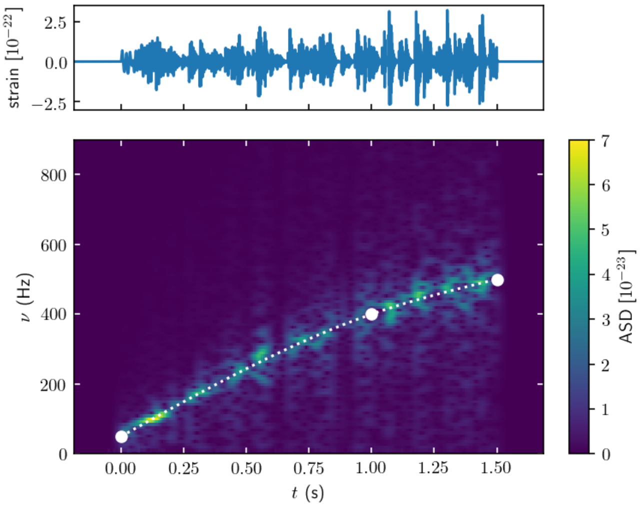


FIGURE 8-2: Example of a phenomenological waveform generated by the waveform generator. The upper panel shows the strain as a function of time and the lower panel the corresponding spectrogram. The white-dotted curve shows the time-frequency dependency used to generate the waveform. White circles represent the pairs (t_i, ν_i) used to generate it.

Fig. 8-1 shows the logarithmic distribution of the rms strain for 2-dimensional (assuming axial symmetry) and 3-dimensional (without symmetry constraints) numerical simulations at 10 kpc. There are significant differences between different simulations depending on the dimensionality. The mean and standard deviation for each distribution is

$$\log_{10} h_{\text{rms,2D}} = -22.4 \pm 0.42,$$
$$\log_{10} h_{\text{rms,3D}} = -23.1 \pm 0.29$$

for the 2-dimensional and 3-dimensional cases, respectively. Given that 3-dimensional simulations are more realistic, we use their normalization to generate our phenomenological waveforms. As a consequence, template amplitudes are about a factor of 5 smaller than typical 2-dimensional simulations. In our waveform generator, the strain of each of the waveforms is scaled to have a rms strain corresponding to a sample following a normal distribution with mean and standard deviation of our normalization, and scaled to the corresponding distance.

TABLE 8.1: Parameter space of the phenomenological templates. The second, third and fourth columns indicate the range minimum and maximum, respectively) for each parameter and the spacing used in the sampling of the parameter space. For Q and D, we show the actual values instead. All times are post-bounce times.

parameter	min.	max.	Δ	description
t_{ini} [s]	0	0.2	0.1	beginning of the waveform
t_{end} [s]	0.2	1.5	0.1	end of the waveform
ν_0 [Hz]	50	150	50	frequency at bounce
ν_1 [Hz]	1000	2000	500	frequency at 1 s
ν_2 [Hz]	1500	4500	1000	frequency at 1.5 s
ν_{driver} [Hz]	100	200	100	driver frequency
Q	$(1, 5, 10)$			quality factor
D [kpc]	$(1, 2, 5, 10, 15)$			distance to source

In synthesis, we have a waveform template that depends on a set of 8 free parameters as reported in table 8.1. In this table, the quality factor, Q, refers to the damping of the harmonic oscillator and the driver frequency, ν_{driver}, to the mean frequency of the random impulsive forcing acting on the oscillator [see 316, for details]. Additionally, for any combination of those parameters we can generate multiple realisations due to the random component in the excitation of the harmonic oscillator and on the random value of the rms strain. To represent the variety of g-mode features observed in the waveform calibration set, we provide ranges covering all the possibilities (see table 8.1). To this parameter space one has to add additional restrictions to ensure the monotonicity ($\nu_2 > \nu_1 > \nu_0$) and convexity ($(\nu_1 - \nu_0)/(t_1 - t_0) \geq (\nu_2 - \nu_1)/(t_2 - t_1)$) of $\nu(t)$, as seen in the numerical simulations. We have created the waveform template bank that contains 504 different realisations of this parameter set, for each distance, resulting in applying the restrictions above to the $9,072$ possible combinations of the parameters in table 8.1. In this way, we obtain a reasonably dense covering of the parameter space. The computational time of the generation of one phenomenological waveform is about 6 ms. This makes it possible to generate the large template banks necessary to train the CNN, which would be prohibitively expensive using multidimensional numerical simulations.

8.2.2 Image generation

In Section 4.3 we commented that almost all GW events detected by LIGO-Virgo network so far, if confirmed, are associated with CBC systems with the peculiar chirp shape of the signal. This feature is used to extract the signal from the detector noise adopting a matched filter approach (see Section 3.2.1). In the case of CCSN, data must be selected and processed using different algorithms. To enhance the detection probability and reduce the false alarm rate, the proposed method implies selecting data in a time window given by the neutrino observatories, taking advantage of common GW features predicted by CCSN numerical simulations.

To assess the robustness of our method, we selected data from O2 of the Advanced GW detectors, without relying on any neutrino information. In particular, we chose a stretch of real data even containing glitches (see Section 2.4.3), taken during August 2017, when Virgo joined the run [80]. The period includes about 15 days of coincidence time among the three detectors and we used this data set to generate about 2 years of time-shift data to train and test the neural network as a noise class.

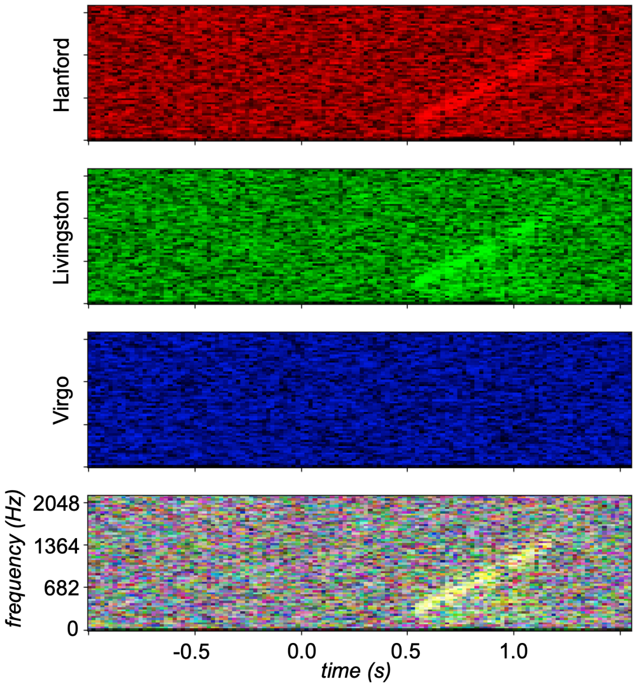


Figure 8-3: From the top; the WDM spectrogram of LIGO Hanford, LIGO, Livingston and Virgo shown in red, green and blue, respectively. At the bottom: the image obtained by stacking the previous three spectrograms. In this case, the signal is present just in Hanford and Livingston so that the combined signal is yellow.

To build images for our neural network algorithm we use the internal features of cWB algorithm[1] (see Section 3.3.3). In our work, we used this software tool to compute the WDM transform (refer to Section 3.3.3 for details), on the base of which the images of 256×64 pixels are built covering the frequency band $[0, 2048]$ Hz and a time range of 2 s. Because the GW signal is embedded in noise and difficult to extract, in [316] a technique to visually enhance the coincidences among all the interferometers of the

---

[1] cWB home page, `https://gwburst.gitlab.io/`;
public repositories, `https://gitlab.com/gwburst/public`
documentation, `https://gwburst.gitlab.io/documentation/latest/html/index.html`.

TABLE 8.2: *List of models of the test set used in the injections. M_{ZAMS} corresponds to the progenitor mass at zero-age in the main sequence (ZAMS). Unless commented, all progenitors have solar metallicity, result in explosions and their GW signal do not show signatures of the standing-shock accretion instability (SASI).*

Model name	reference	M_{ZAMS}	comments
s9	[335]	$9M_\odot$	Low mass progenitor, low GW amplitude.
s25	[335]	$25M_\odot$	Develops SASI.
s13	[335]	$13M_\odot$	Non-exploding model.
s18	[336]	$18M_\odot$	Higher GW amplitude.
he3.5	[336]	-	Ultra-stripped progenitor ($3.5M_\odot$ He core).
SFHx	[337]	$15M_\odot$	Non-exploding model. Develops SASI.
mesa20	[338]	$20M_\odot$	
mesa20_pert	[338]	$20M_\odot$	Same as mesa20, but including perturbations.
s11.2	[331]	$11.2M_\odot$	
L15	[328]	$15M_\odot$	Simplified neutrino treatment.

network has been developed. The method consists of using primary colours for the spectrograms: red (R) for LIGO-Hanford, green (G) for LIGO-Livingston and blue (B) for Virgo. A random example of the input of the CNN is shown in Fig. 8-3.

The main results in this work present some differences with respect to [316]:

- incorporation of the information of the source distance.
- coverage of larger parameter space with our phenomenological waveforms.
- as mentioned before, usage of real data from O2.
- usage of all spectrograms, instead of the ones detected by cWB.

We define the starting time of each image every 2 s, echoing the choice made in [316]. The images containing the central time of injected signals are considered as event class, instead, the ones without signals are noise class. The injected signal is expected to be about 600-700 ms in duration, drawn everywhere in the image, with a small probability of being between two consecutive images. Such images are also used for training, so the network can recognise also the partial signature of the event.

8.2.3 Previous set

In [316], phenomenological supernova signals were injected in Gaussian noise simulating the final expected sensitivity of Advanced LIGO and Virgo detectors. Signals were injected at fixed network SNR, and did not include any information about source distance. This set was constructed using the information given by cWB algorithm and, unlike in the following data sets, only using events passing the first stage of cWB analysis. This set contains about 10, 000 images with signals for 11 different SNR ranging from 8 to 40 and the same amount with only noise, 75% of the signals are used to train the network and 25% for validation.

8.2.4 Training set

The training set for CCSN signals has been constructed injecting waveforms at fixed distances: 0.2, 0.4, 1, 2 and 3 kpc. For this purpose, we have used the waveform template bank described in Section 8.2.1 injecting, for each distance, of the order of 70, 000 waveforms, with random sky localization. 75% of the set is used in the actual training while the remaining 25% is used for validation. The upper distance has been chosen according to the requested minimal SNR of the injected signals, in which the trace of the CCSN signal is distinguishable from the noise in the image passed to the neural network.

8.2.5 Blind set

In the blind set, we injected a new ensemble of about $260,000$ simulated signals, generated by the phenomenological templates described in Section 8.2.1. In this case, distance is chosen in a uniform distribution between 0.2 and 15 kpc, and positions in the sky are randomly chosen. This set is used to quantify the detection efficiency and to test the network. The set is bigger than the training set, to maintain enough statistics for all the distances.

8.2.6 Test set

For the final test, we perform injections using CCSN waveforms from numerical simulations found in the literature. In particular, we focus on 3D simulations of non-rotating progenitors representative of the neutrino-driven mechanism. The selection *test set*, hereafter, see Table 8.2, is performed based on the realism of the computed simulations in terms of neutrino transport and equation of state and on the completeness of the GW signal[2]. The selection includes models with a variety of features in the GW spectrum and coincides with the choice for ongoing supernova searches by the LIGO-Virgo-KAGRA collaboration. In particular, select models with a variety of progenitor mass M_{ZAMS} at zero-age in the main sequence (ZAMS), which refers to the mass of a star when it first begins to fuse hydrogen into helium in its core, marking the start of its main sequence phase. Furthermore, unless specified all models have solar metallicity, i.e. abundance of elements heavier than hydrogen and helium, and do not show signatures of standing-shock accretion instability SASI. Except for model L15, none of the models coincides with the models selected for the waveform calibration set used in Section 8.2.1. With this choice, the injected waveforms are in practice completely uncorrelated to any information we have used to train the CNN network. The procedure is similar to the one used for the blind set of the previous test: we injected about 65000 waveforms uniformly in distance and sky directions, from 100 pc to 15 kpc.

8.3 Methodology

8.3.1 Inception blocks and ResNet

In Section 4.3, we have seen that DL algorithms have emerged as a new tool also in the GW field. These methods can perform rapid predictions since all the intensive computation is diverted to the one-time training stage, which could make them orders of magnitude faster than the conventional matched-filtering techniques (see Section 3.2.1). In addition, there are no limitations in the size of the templates bank of GW signals, and even more, it is preferable to use large data sets to cover as deep a parameter space as possible. Due to this fact, they sparked the interest of several authors, who have built deep-learning algorithms to demonstrate their power on specific examples, including CCSN [317, 212, 214] (see Section 4.3).

As we have explained in Section 4.2.2, CNN is a specialized kind of DL algorithm to process data that has a known grid-like topology and can learn to differentiate a variety of input types due to its ability for pattern recognition [339]. With these ideas in mind, [316] provided clear evidence that, under relatively simplified conditions, deep CNN algorithms could be more

---

[2] Some of the models in the literature compute less than 100 ms after bounce or have a poor sampling rate.

efficient in extracting GW signals from CCSNe than the current methodology. Therefore, we aim to improve the neural network developed in [316], going deeper with convolutions to increase accuracy while keeping computational complexity at a reasonable cost.

The most straightforward way of improving the performance of a deep neural network is by increasing its size, which includes the number of layers and the number of neurons per layer. Nonetheless, enlarging a network implies training a larger amount of parameters and over-complicating the model, which increases dramatically the computational cost and reduces the generalization ability of the network, i.e. the network would be prone to over-fitting. A fundamental way of solving these issues would be to move from fully connected to sparsely connected architectures, as discussed in [340]. In this work, a sophisticated network topology is built, the so-called Inception network, that tries to approximate a sparse structure. The architecture is composed of blocks of convolutions, known as Inception modules.

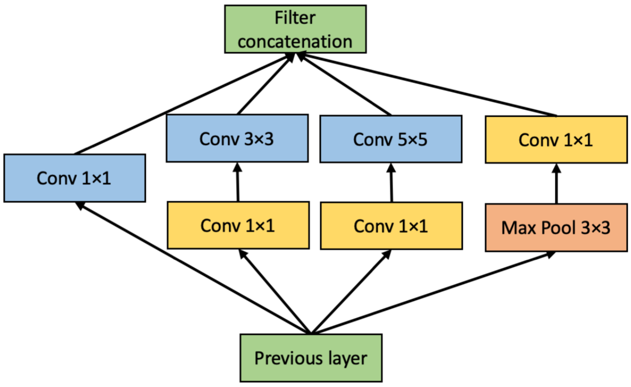


Figure 8-4: Inception module with dimensionality reduction, adapted from [340].

The input of each block is convolved in parallel by separate CNN layers with different kernels, while the outputs of all the convolutions are stacked, as we can observe in Fig. 8-4. In such a way, a sparse network is built without the necessity of choosing a particular kernel size, but computational complexity increases drastically. To prevent a high computational cost the authors introduce *dimensionality reduction*, i.e. $1 \times 1 \times 1$ convolutions that reduce the depth of the output. Note that we are taking the convention of height × width × depth as in Section 4.2.2. Thus, if we convolve our input $w_{in} \times h_{in} \times d_{in}$ with f filter $1 \times 1 \times 1$, stride $s = 1$ and padding $p = 0$, according to Eq. 4.17 the output will be $w_{in} \times h_{in} \times f$. Therefore, if $f \ll d_{in}$ the depth and the number of parameters will be greatly reduced. In later releases of the Inception network, the authors explore further the idea of *dimensionality reduction.*

In [341], they explore other ways of factorizing convolutions in various settings, especially to increase the computational efficiency of the solution without reducing the expressiveness of the block. Firstly, the authors examine the factorization into smaller convolutions, where they claim that $5 \times 5 \times 1$ convolution can be *factorized* into two $3 \times 3 \times 1$ convolutions since the final output has the same dimensions, being the main difference the number of parameters.

A $5 \times 5 \times 1$ convolution needs $(5^2 \times d_{in} + 1) \times d_{out}$ parameters to train, while for two $3 \times 3 \times 1$ convolutions it is necessary to train $2 \times (3^2 \times d_{in} + 1) \times d_{out}$ parameters, which is less computationally expensive. Secondly, they analyze the factorization into asymmetric convolutions, such that $c \times c \times 1 \longrightarrow \{c \times 1 \times 1,\, 1 \times c \times 1\}$. Again, the outputs of both processes have the same dimensionality but different amounts of parameters, $(c^2 \times d_{in} + 1) \times d_{out} > 2 \times (c \times 1 \times d_{in} + 1) \times d_{out}$. Thus, in [341] the authors *factorize* $5 \times 5 \times 1$ into $3 \times 3 \times 1$, which in turn are *factorized* by $3 \times 1 \times 1$ and $1 \times 3 \times 1$ convolutions, to lighten the calculations.

Another obstacle of deeper networks is the degradation problem, where with increasing depth, accuracy gets saturated and then degrades rapidly. In [342], this problem is approached by introducing a deep neural network, called Residual Network or ResNet. This network can learn the identity function using shortcut connections that skip one or more layers, which are also known as "skip connections". Therefore, the network is reminded every few layers how was the input a few layers before, which can be translated into learning the identity function with a simple demonstration. Furthermore, in [342] different empirical results show that the degradation problem is well addressed since accuracy gains are obtained from increasing depth.

Due to the improvements in accuracy obtained with Inception network and Resnet, in [343] authors explore the combination of these two architectures, while *factorization* is discussed in [341]. As a result, they developed, among others, an architecture called Inception-Resnet v1 which is ~ 90 layers depth. It was demonstrated that the introduction of residual connections led to a dramatic improvement in the computational speed while showing that Inception-Resnet algorithms achieved higher accuracies with fewer iterations of the training phase.

Our problem is simpler than the task performed in [343], since we only need to discriminate between two classes: inputs that contain a GW CCSN signal with noise (event class) and inputs that contain only noise (noise class). Hence, the need to increase the complexity of the CNN in our case is due to the loudness of the noise power, rather than the number of the different classes. As a consequence, we developed reduced ("mini") versions of Inception v3, Resnet and Inception-Resnet v1, using the original building blocks of those networks, but adapting them to our needs and limiting the number of layers to ≤ 30 to avoid over-fitting.

8.3.2 Mini architectures

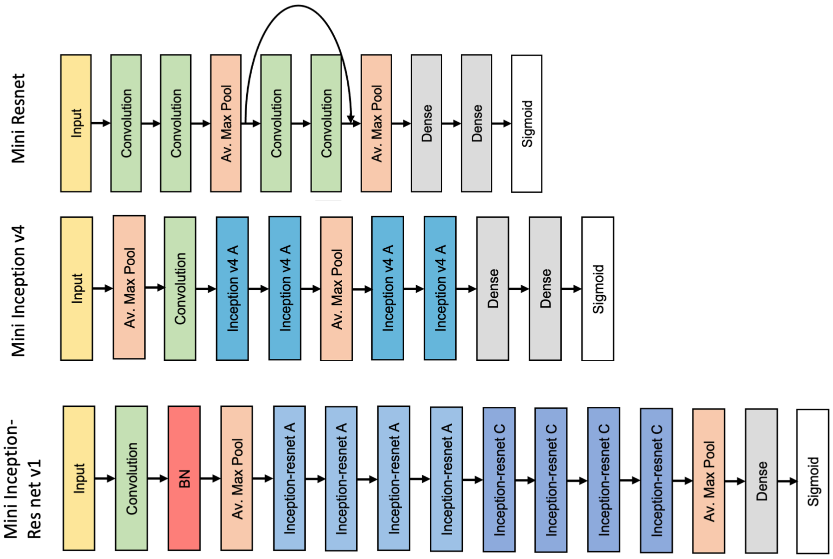


FIGURE 8-5: *Reduced or "mini" versions of Resnet, Inception v4 and Inception-Resnet v1 for the best performing architectures with minimal fine-tunning.*

In this section, we present the reduced architectures, namely Mini Resnet, Mini Inception v4 and Mini Inception-Resnet v1. Several variations of these architectures were explored and in this work, we present the best-performing ones. For the development of the networks, including the model definition, and the training and validation phases, we have used the Keras frameworks [301], based on the TensorFlow backend [344]. We employ Adamax optimizer, with a learning rate of 0.001 and we train each network for 20 epochs [345]. The activation function of all the convolutional layers is ReLU activation function, and we use *binary cross-entropy* loss function and a sigmoid activation function for the output (see Section 4.2.1).

Excessive Max pooling hinders the learning process, as it might extract crucial information that the consequent layers need. Therefore, we use a minimal amount of pooling layers, but an optimized version would require further parameter reduction. In Figs. 8-5 we provide a scheme of these networks and in the following, we present their main characteristics:

- *Mini Resnet:* it has a single "skip connection" (represented as an arch). This is because when increasing the number of layers and "skip connections", the performance of the network decreased rapidly for short architectures (≤ 30 layers). This network has a total of 381,390 parameters and a single epoch takes 31 s.

- *Mini Inception v4:* we implement the block of Inception v4 A (see [343] for details). This network has a total of 250,251 parameters and a single epoch takes 26 s.
- *Mini Inception-ResNet v1:* we implement the block of Inception-ResNet A. The modules Inception-ResNet-B and Reduction-B are the most expensive blocks since the convolutions inside them are $1 \times 7 \times 1$, $7 \times 1 \times 1$ and $7 \times 7 \times 1$, so they are discarded for this work (see [343] for details). Hence, we discard these modules to implement the reduced version of this algorithm. This is the most complex network and the most expensive to train, as it has a total of 522,346 parameters and a single epoch takes 43 s. In an optimized version, Reduction blocks should be present to enlighten the computations.

Due to its high preliminary performance we shrink the number of parameters of Mini Inception-ResNet by interspersing Inception-Resnet modules with Reduction-A blocks and fine-tuning it. In the following, we describe the full-optimized architecture of Mini Inception-ResNet.

8.3.3 Optimized architecture of Mini Inception-Resnet

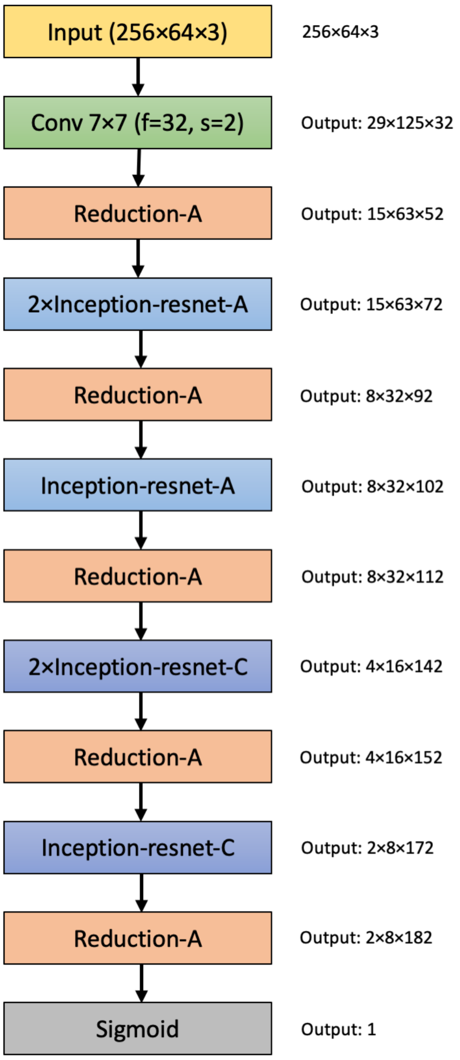


FIGURE 8-6: *The overall schema of the Mini Inception-Resnet network.*

For the development of our optimized Mini Inception-Resnet network, we employ Adam optimizer [345] with a learning rate $lr = 0.001$ and $\epsilon = 10^{-6}$ to avoid divisions by zero when computing back-propagation, and ReLU activation function. We employ a batch size of 64 because it is a good trade-off between computational complexity and performance.

Despite facing a classification problem with two classes, the approach used in [316] is to employ the categorical cross-entropy loss function with a softmax activation function in the last layer, i.e. the problem is treated as a multi-class classification problem with two classes. In this work, we simplify this approach by using a *binary cross-entropy* instead and a sigmoid activation function for the output, i.e. we address the problem as a classification problem with a positive class (event class) and a negative class (noise class). Therefore, the output of the network is a probability vector θ, which contains the probabilities of the template belonging to one class or another. The classification task is then performed according to a pre-defined threshold θ^*, i.e. the template will be classified as *event class* only if this probability overcomes θ^*.

In [343], the authors build 5 different types of blocks, namely Inception-ResNet-A, Inception-ResNet-B, Inception-ResNet-C, Reduction-A and Reduction-B. As we mentioned in the previous Section, to alleviate the computational complexity of the model we use Inception-ResNet-A and Inception-ResNet-C. We also shrink the number of parameters of our network by interspersing Inception-Resnet modules with Reduction-A blocks, as presented in Fig. 8-6.

The Inception-ResNet-A block (see Fig. 8-7) is equivalent to the Inception module shown in Fig. 8-4. It is interesting to note that the Max Pooling layer is substituted by the “shortcut connection”, and the $5 \times 5 \times 1$ convolution is *factorized* by two $3 \times 3 \times 1$ convolution layers. Moreover, the Inception-ResNet-C block (see Fig. 8-8a) is equivalent to the Inception module without the $5 \times 5 \times 1$ convolution layer. Note that the Max Pooling layer is again replaced by the "shortcut connection", and the $3 \times 3 \times 1$ convolution is *factorized* by $1 \times 3 \times 1$ and $3 \times 1 \times 1$ convolution layers. The module Reduction-A (see Fig. 8-8b) shrinks the number of parameters thanks to a $3 \times 3 \times 1$ Max Pooling layer.

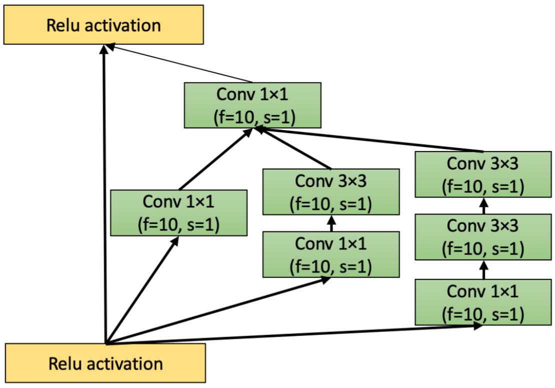
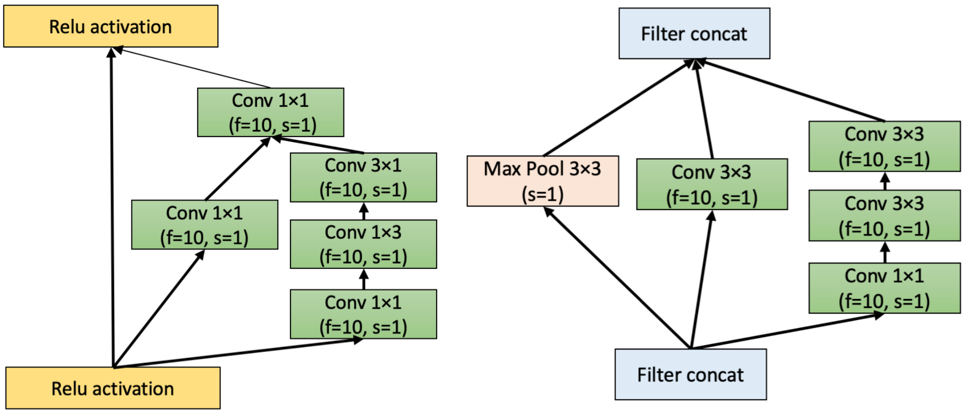


FIGURE 8-7: The schema for Inception-ResNet-A, adapted from [343].

Due to its deepness, the resulting Mini Inception-Resnet architecture is much more flexible than the one presented in [316]. As we have discussed previously, increasing the number of layers might be counterproductive and would drastically increase the computational complexity of the network. Nonetheless, these two concerns are solved with the incorporation of "shortcut connections", which allows the input not to be forgotten, and a *factorized* grid-like architecture that alleviates the computational complexity of the CNN, on top of avoiding over-fitting. Thus, Mini Inception-Resnet is ~ 30 times more complex, as the *previous network* has ≈ 6000 parameters and the optimized Mini Inception-Renset has $\approx 99,000$ parameters.



(A) The schema for Inception-ResNet-C.

(B) The schema for Reduction-A.

FIGURE 8-8: Schematic blocks from original Inception-Resnet, adapted from [343]

8.3.4 Training methodology

In this section we describe how we convert training images into categorical data for the identification of CCSN signatures in Gaussian and real noise, solving our binary classification task.

As in [316], we train the network using *curriculum learning*, where we start training with the easiest data sets, and then, gradually, the task difficulty is increased. We note that, although our set of phenomenological waveforms is constructed using a series of fixed distances, the

SNR follows a statistical distribution resulting from the random process used to generate the waveforms, for each of these distances (see Sect. 8.2.1). In practice, instead of using the distance, we define data as a set of inputs whose SNR is fixed in a pre-defined range. In this way, the difficulty of the data sets increases with decreasing SNR. It is important to note that it is key to obtain a high performance when learning easy examples at high SNR, to be able to capture the hard examples later on. The data sets are balanced so that 50% of the templates belong to the event class and 50% to the noise class. Because the present network is much larger than that in [316] where we had balanced training and validation sets, here we use 75% of the data for training and 25% for testing.

The previous work [316] measured the performance of the neural network in terms of the efficiency η_{CNN}, equivalent to the TPR (Eq. 5.4), and the false alarm rate FAR_{CNN}

$$FAR_{CNN} = \frac{\text{misclassified noise}}{\text{all classified events}} = \frac{FP}{FP+TP} \tag{8.1}$$

where these metrics are presented in Table 5.1. In this research, we also measure the performance of our network with the *receiver operating characteristic* curve (ROC curve), which is created by plotting the TPR against the FPR (Eq. 5.5).

8.4 Results

8.4.1 Comparison of mini architectures in Gaussian noise

To train and validate the networks, we use the data set described in [316], composed of waveforms ranging in the interval SNR$\in [8, 40]$. This choice allows us for a direct comparison with [316] and to make an informed decision on the choice of the architecture to optimize. In Fig. 8.3.2 we plot efficiency η_{CNN} and the false alarm rate FAR_{CNN} as a function of the SNR for the three architectures presented in 8-5. Note that these networks have not been fully optimized but are rather a proof of concept for the final design of the full architecture.

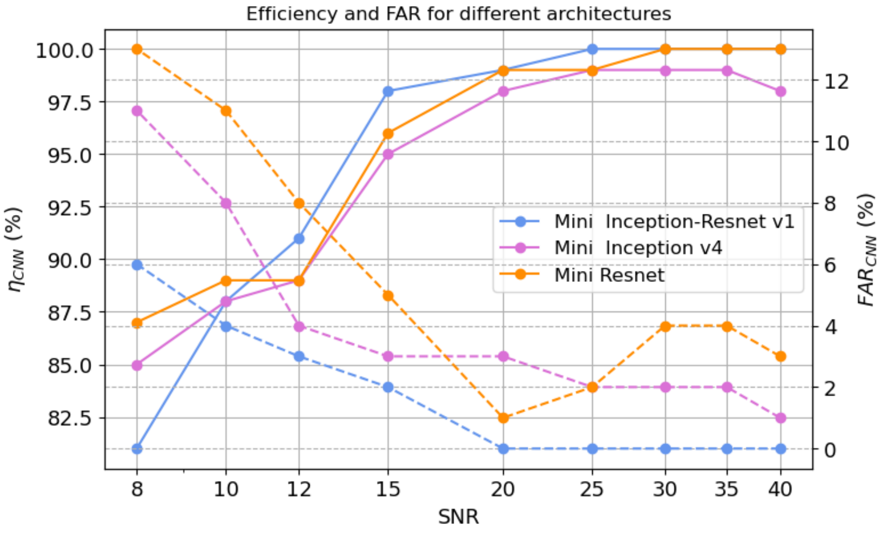


Figure 8-9: η_{CNN} (solid lines) and FAR_{CNN} (dashed lines) as functions of SNR computed during the validation process for architectures presented in 8.3.2

As we can observe, Mini Inception v4 (pink) and Mini Resnet (orange), have a similar performance in terms of efficiency η_{CNN}, but Mini Inception v4 has a lower FAR_{CNN}. Due to its complexity and generalization ability, Mini Inception-Resnet v1 acquires the lowest FAR_{CNN}

for large SNR, and it has the largest η_{CNN}, except for low SNR. Due to its better performance, we focus only on the Mini Inception-Resnet v1 network, which is optimized in the next Section and used for the different tests presented in the rest of this work.

8.4.2 Comparison with previous results in Gaussian noise

In this subsection, we will describe the experiments performed with injections in Gaussian noise. To train and validate Mini Inception-Resnet v1, we use again the data set described in Section 8.2.3, composed of waveforms ranging in the interval SNR$\in [8, 40]$. This choice allows for a direct comparison with the results in [316] and it helps to improve the present software architecture.

To improve the performance of [316] it is necessary to minimize FAR_{CNN} while maximizing η_{CNN}. Therefore, from Eq. 8.1 we wish to minimize FP instead of FN, i.e. we need to penalize the algorithm when it classifies noise class as event class. To be able to penalize the algorithm we implement *weighted binary cross-entropy*, where we assign weight w to the noise class and weight 1 to the event class. We vary this parameter between $w \in [1.0, 3.5]$, where $w = 1$ would be equivalent to a normal *binary cross-entropy* and $w = 3$ would mean that it is 3 times more important to correctly classify the noise class rather than the event class. Nonetheless, in this work, we only present the results of $w \in \{1.0, 2.0\}$ to be able to compare the best working case ($w = 2.0$), with the base case ($w = 1.0$).

Moreover, the algorithm returns the probability θ that a certain template belongs to the event class. We want this probability to be high without dramatically decreasing η_{CNN}. Therefore, we define the decision threshold θ^* in the range $[50\%, 85\%]$; when a given probability exceeds this value, we will classify the template as an event, otherwise, it is classified as noise. Therefore, we perform different experiments to tune w and θ. In figures 8-10 and 8-11, we obtain η_{CNN} and FAR_{CNN} for $w = \{1, 2\}$ and $\theta = \{50\%, 65\%, 85\%\}$.

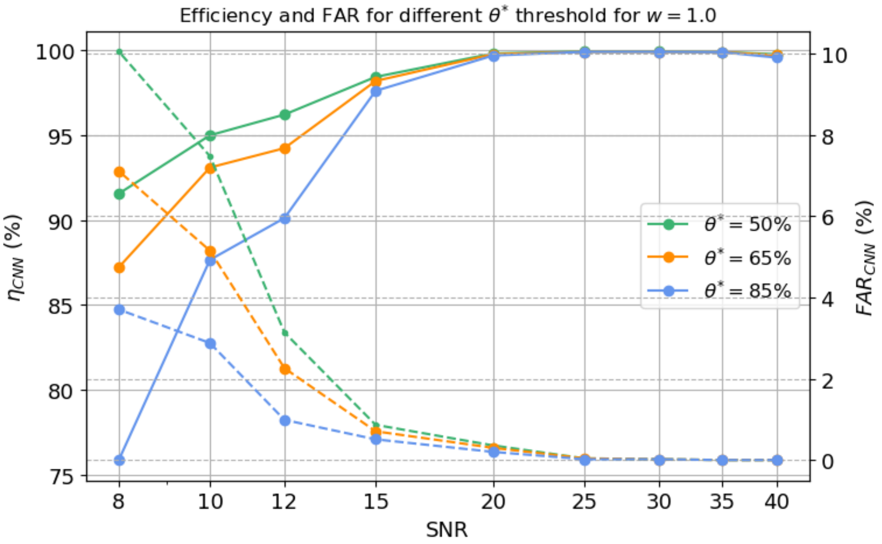


FIGURE 8-10: η_{CNN} (solid lines) and FAR_{CNN} (dashed lines) for different SNRs computed during the validation process for $w = 1.0$ and different θ^* thresholds.

In Fig. 8-10, we report the high performance of low θ in terms of η_{CNN}, paying the price in even relatively high FAR_{CNN}. The opposite behaviour occurs for high θ. To be able to improve the probability distribution θ, we will penalize the loss function with $w = 2.0$. This means that the impact of correctly classifying noise is twice higher than correctly classifying events, as we show in Fig. 8-11 where the FAR_{CNN} is minimized with respect to Fig. 8-10 with some cost in η_{CNN}.

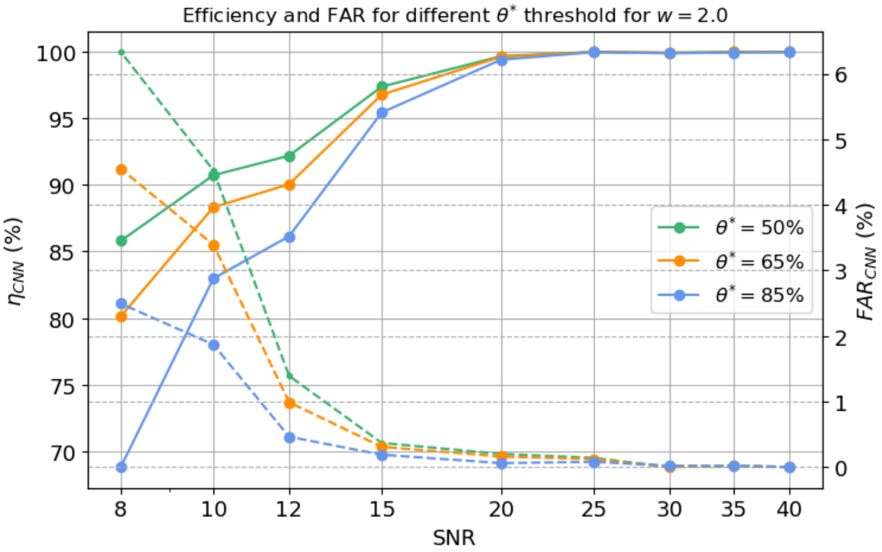


FIGURE 8-11: η_{CNN} (solid lines) and FAR_{CNN} (dashed lines) for different SNRs computed during the validation process for $w = 2.0$ and different θ^* thresholds.

Notice that w will penalize the learning, so if the network is learning correctly the results would be enhanced, but it will lead to poor results otherwise. This is evident when we compare the results shown in the figures 8-10 and 8-11: if we increase w we have less performance in terms of η_{CNN}, with little gains in FAR_{CNN}. To have a comparison between Fig. 8-10, 8-11 and the results from the previous paper [316], we plot the validation results of Mini Inception Resnet for $w = \{1, 2\}$ in Fig. 8-12. Since we want to obtain a trade-off between η_{CNN} and FAR_{CNN}, we settle $w = 2.0$ and $\theta^* = 65\%$.

The main improvement of our network with respect to [316] is the minimization of FAR_{CNN} towards $\sim 0\%$ for SNR in range $[15, 20]$, while maintaining the same η_{CNN}. We note also that the poor performance at low SNR is because this architecture is susceptible to the strong presence of Gaussian white noise, as it is pointed out in [346]. Hence, the role of the decision threshold $\theta^* = 65\%$ is two-fold. On one hand, with this decision threshold, we obtain $\max(FAR_{CNN}) \approx 4\%$ for low SNR which is the upper limit obtained by the previous paper [316]. On the other hand, $\theta^* = 65\%$ provides us with a fair trade-off between η_{CNN} and FAR_{CNN} as we have discussed before.

In terms of speed performance, in a GPU *Nvidia* Quadro P5000 it takes 1h 18 min to train, validate and test Mini Inception Resnet for this particular data set with 5 epochs for each SNR. A great part of this time is employed in training the neural network, so with bigger data sets the computational time will increase. Nonetheless, once the network is trained, the prediction is performed within seconds.

8.4.3 Training and validation in real O2 noise

In this section, we describe the experiments performed using the *training set* (section 8.2.4). This set contains injected phenomenological signals in real noise in the interval SNR$\in [1, 232]$. As before, for each data set at a given SNR, we calculate FAR_{CNN} and η_{CNN} during the validation. We also vary the penalization parameter $w \in \{1, 2\}$, and as in the previous Section we choose $w = 2$ and the decision threshold $\theta^* = 65\%$.

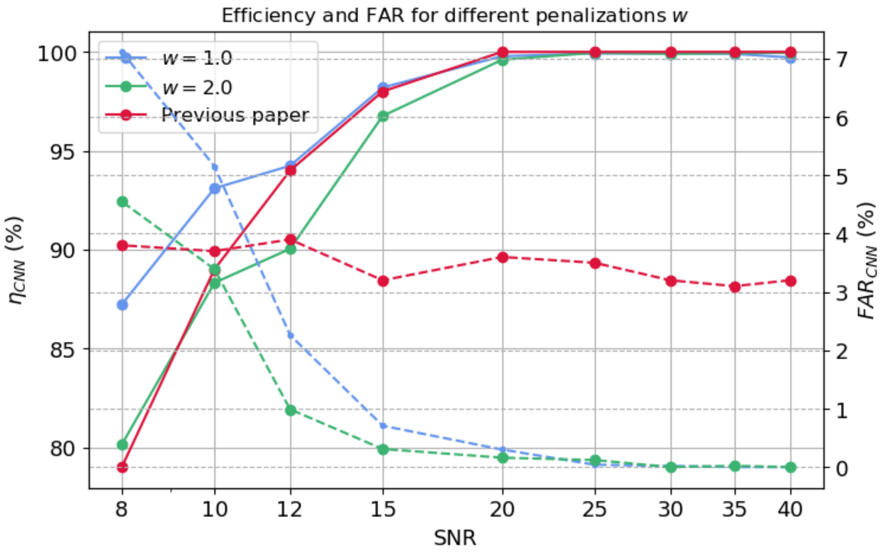


FIGURE 8-12: figure
η_{CNN} (solid lines) and FAR_{CNN} (dashed lines) as functions of SNR computed during the validation process of $w = \{1, 2\}$, with $\theta^* = 65\%$, and [316], where $\theta^* = 50\%$.

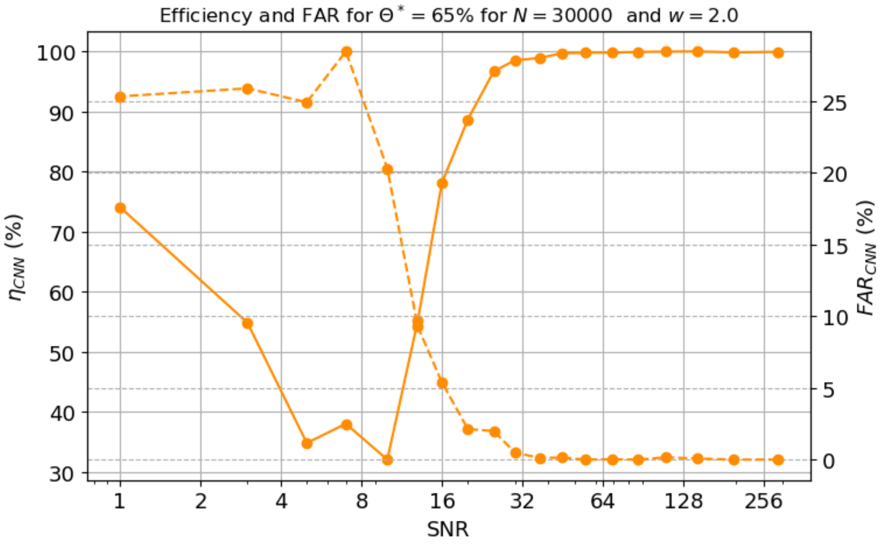


FIGURE 8-13: η_{CNN} (solid line) and FAR_{CNN} (dashed line) for SNR in range $[1, 232]$, for $w = 2.0$ and $\theta^* = 65\%$. These results are obtained from validating 25% of the data that we have not trained on.

For the network to learn correctly the input, it is crucial to perform a smooth "curriculum learning". Due to the difficulty of the data set, we separate the event templates into bins of size N and noise templates are packed accordingly. We performed the training for different N but a better trade-off between η_{CNN} and FAR_{CNN} was observed for $N = 30,000$, which provided a smoother transition between SNR bins. Therefore, in Fig. 8-13 we show the results of the validation having fixed $N = 30,000$, $\theta^* = 65\%$ and $w = 2$.

In Fig. 8-13 we note that η_{CNN} is $\sim 98\%$ above $SNR = 32$ and below this value, η_{CNN} starts decreasing. Instead, FAR_{CNN} is $\sim 0\%$ but increases for SNR < 20. For lower SNR values the method tends to show an erratic behaviour, that we foresee due to the statistical structure of the real noise.

This procedure is rather fast. In terms of speed performance, in a GPU *Nvidia* Quadro P5000 it takes 2h 21 min for Mini Inception-Resnet to train and validate for this particular data set, but only 10 min to predict the *blind set* and *test set*. The time increase in the training phase is because now we set the number of epochs to 10 instead of 5 to guarantee a better convergence of the network's trainable parameters.

8.4.4 Testing in real O2 noise: blind set and test set

In this section, we present the results obtained when we used the network trained and optimized in the previous Section on the data of the *blind set* (section 8.2.5) and the *test set* (section 8.2.6). The network has not been trained by any of the images of these two sets so they can be used for the final test of the performance of the network. The signals injected in the *blind set* correspond to waveforms generated by the same procedure used to generate the training set, while the injections in the *test set* correspond to realistic CCSN waveforms.

In Fig. 8-14, we report the histogram of the injections in the real noise. Such a plot shows the robustness of the decision threshold $\theta^* = 65\%$ even in the case of real detector noise.

In Fig. 8-15 we plot the ROC curve and we calculate the area under the curve (AUC).

We note the high performance of the *test set* (AUC=0.79) compared with that obtained for the *blind set* (AUC=0.90). Even if we only trained our network with phenomenological waveforms from the template bank described in Section 8.2.1, such waveforms mimic the behaviour of the test set described in 8.2.6, which is the main reason behind such good results.

Another interesting graph that shows the resemblance between the *blind set* and *test set* is Fig. 8-16a. Here we plot η_{CNN} as a function of the distance. As we can see, at short distances, there is a difference in efficiency between *blind set* and *test set* of ≈ 10%, but when we increase the distance, they seem to reach

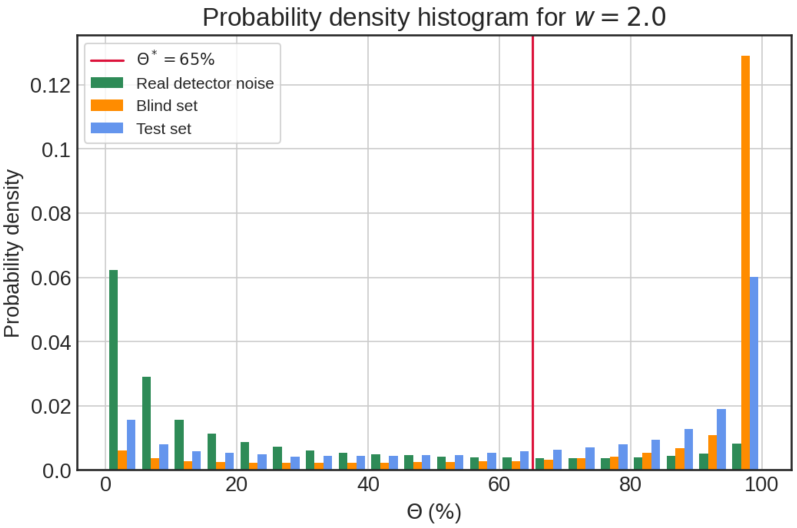


FIGURE 8-14: Histogram of real detector noise and injections in real-time as a function of the probabilities predicted by Mini Inception Resnet. The vertical line represents the chosen decision threshold $\theta^* = 65\%$. Given the counts of the ith bin c_i and its width b_i, we define the probability density as $c_i/(\sum_i^N c_i \times b_i)$, where N is the total number of bins of the histogram.

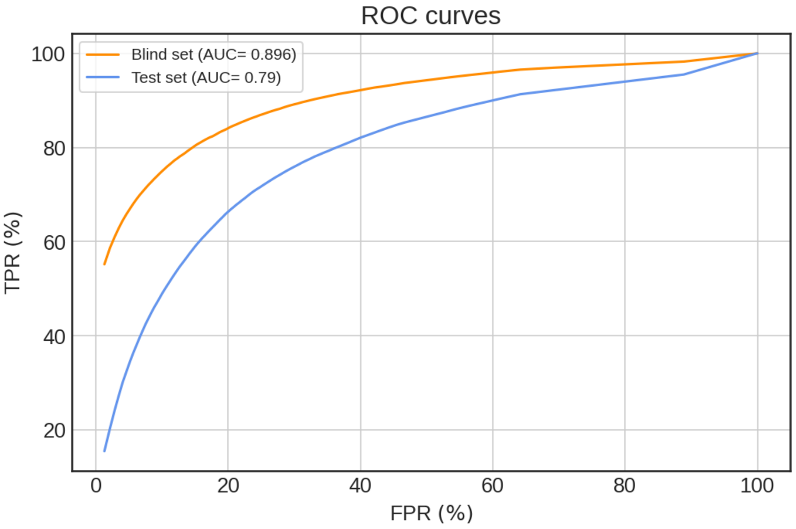
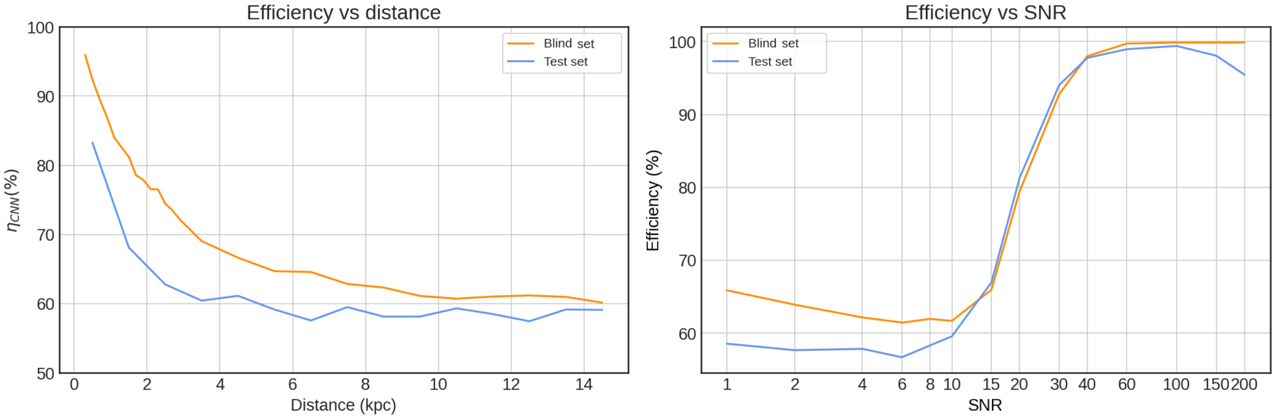


FIGURE 8-15: Performance of our neural network for the blind set and the test set for $\{w, N, \theta^*\} = \{2, 30000, 65\%\}$. AUC is presented in the legend of the plot.

a lower limit at $\eta_{CNN} \approx 60\%$. In Fig. 8-16b we also plotted η_{CNN} against SNR. For low SNR, the difference in efficiency η_{CNN} in the two cases, *blind set* and *test set*, is around 10%, while for SNR > 15 we obtain the same efficiency. These final results assess the robustness of this method to detect CCSN signals embedded in the real noise.



(A) η_{CNN} as a function of the distance.

(B) η_{CNN} as a function of SNR.

FIGURE 8-16: *Performance evaluation computed during the testing process for* $\{w, N, \theta^*\} = \{2, 30000, 65\%\}$.

8.5 Discussion

The search of CCSN signals is carried by an algorithm whose architecture trains 98,997 parameters, significantly more than the 3,210 parameters taken into account in [316]. This implies an increase in the network complexity by a factor of 30. We trained the Mini Inception-Resnet using about 26,000 images corresponding to spectrograms of phenomenological waveforms injected in real noise of the three detector network LIGO-Virgo during O2, and a similar number of images without signals. We used the curriculum learning with decreasing value of the SNR for the training. The significant differences with [316] are:

- the increase of the training images by a factor ~ 10,
- the extended variability of the injected waveforms, to mimic the behaviour of the results from the CCSN numerical simulations,
- the novel waveform parametrization for the frequency evolution,
- the use of real detector noise instead of Gaussian one,
- images are no longer built by applying a SNR threshold by cWB.

Firstly, we developed three different architectures to improve the detectability of a GW signal from CCSN: Mini ResNet, Mini Inceptio v4 and Mini Inception ResNet v1. After the first preliminary results, we decided to fine-tune Mini Inception ResNet v1 due to its better performance. Secondly, to compare the efficiency of this new method with previous results, we run the optimized version of the Mini Inception-Resnet network with the same setup as in [316]. The validation step shows that with the appropriate choice of parameters ($\theta^* = 65\%$ and $w = 2$) we minimize the FAR_{CNN} towards ~ 0 % almost maintaining the same efficiency η_{CNN} for the range of SNR$\in [15, 20]$.

Then, we tested this method by injecting signals in the noise data of the LIGO-Virgo network taken during the second observation run. We have applied this analysis method to detect two classes of signals. The first one is a blind set composed of the same phenomenological templates having the same analytical structure as those signals of the training set. The second one is based on 3-dimensional realistic numerical CCSNe simulations available in the literature.

In the validation process, carried on using the dedicated 25% of the training set where signals are uniformly distributed in distance between $[0.2, 15]$ kpc, we obtain about 80% efficiency with a false alarm rate of about 5% for SNR$= 16$, see figure 8-12.

When applying the same method trained with phenomenological templates to the case of realistic GW signals from 3-dimensional numerical simulations (*test set*), we still obtain a reliable performance. Overall, when compared to the case of the blind set, the efficiency at SNR > 15 is very similar while at lower SNR we observe a reduction of less than 10%, see figure 8-16a. The satisfying agreement is an indication that our phenomenological template generator is mimicking the main features observed in realistic CCSN and therefore, it supports the choice of this kind of template to train CNNs. Moreover, the high performance of the network for both testing data sets implies a high generalization ability, meaning that the network is not prone to over-fitting. However, the decrease in efficiency at low SNR could be an indication that some of the features of CCSN are not perfectly captured by the templates, for example, the variability of the waveform amplitude for the duration of the signal (we consider that is in average constant) or the presence of low-frequency components associated with SASI. Future work could incorporate these two features to improve the performance of the search method.

One of the advantages of the newly developed phenomenological templates is that they contain information about the distance to the source, which allows us to study the performance of the blind set with respect to the distance and to compare directly with the results of the test set. Concerning the distance, the efficiency shows in general a quick drop at ~ 2 kpc followed by a gentle decline, falling to about 60% at 15 kpc. This contrasts very much with the behaviour for the SNR that shows a steep decline at SNR ~ 15. The reason for this difference is that, at a given distance, there is some variability in the amplitude of the possible waveforms, which tends to smooth out the results over a range of distances. We expect that at larger distances the efficiency will keep decreasing towards zero, but we did not see this effect within the limited set of distances used in this work. The performance with the *realistic* test set is somewhat worse than with the blind set, but the difference in efficiency is never larger than 10%, in agreement with the results obtained as a function of SNR.

We note that these results have been obtained using realistic waveforms from 3-dimensional models, which are in general about a factor of 5 weaker than those of 2-dimensional simulations. It is also important to notice that we have used real O2 noise, so the results are expected to be better for the current detector configuration, which recently ended O3, and will improve further once the final sensitivity of LIGO, Virgo and KAGRA detectors will be achieved. These two factors make it difficult to compare our results with those obtained in other papers using injections based on 2-dimensional simulations, simulated Gaussian noise and/or ultimate detector sensitivity [317, 212, 214].

In fact, for the case of neutrino-driven explosions in [317] they use a set of waveforms from 55 numerical simulations (mixed 2-dimensional and 3-dimensional) to perform about 10^5 injections with random orientations in the sky in the range $\sim [0.2, 200]$ kpc. Using a LIGO-Virgo-KAGRA network with optimal sensitivity, they obtain an efficiency of 50% at 4 kpc with a false alarm probability of 0.1%. These results are similar to our work, however, a closer comparison is difficult since they are using an interferometer network with design sensitivity.

The work of [212] focused on using Genetic Programming algorithms to improve the significance of a single interferometer detection. For that purpose, they trained the algorithm making injections of CCSN waveforms in real detector noise from the LIGO-Virgo O1. For the case of neutrino-driven explosions, the algorithm is trained using waveforms from 2-dimensional and 3-dimensional CCSN simulations (8 in total) injected at different locations in the sky and distances in the range $\sim [1, 7.5]$ kpc (about 15,000 injections in total). Similarly to [316], they employed cWB pipeline. For waveforms from 3-dimensional simulations (not the same as ours), they get an efficiency of 86% at 3.16 kpc with 12% of false negatives. Again, the results are in the bulk of our numbers but it is difficult to compare since they are using a network with lower sensitivity than ours and the injections that are comparable to ours amount only to 4 different signals. Their results show that a detection with high significance (3-σ) for signals with an SNR as low as 10 is possible. However, it should be noted that, in their case, the same waveforms were used for training and testing.

Finally, [214] utilized a CNN trained using 5 waveforms from neutrino-driven CCSN 3-dimensional simulations injected in Gaussian noise considering the spectral sensitivity curve of Virgo during O3. The training was performed with about 25,000 random injections in the sky at distances between $\sim [0.01, 10]$ kpc. To test the robustness of the method they also accounted for short-duration detector noise transients, known as glitches, in simulated data. When using different waveforms for training and testing, they obtain an efficiency of $\sim 90\%$ of all triggers with a $\sim 10\%$ false alarms (all distances in the range). When using the same waveforms for testing and training they observe a drop in the efficiency, below 50%, for values of the SNR in the range $[11, 16]$, depending on the waveform.

Despite the differences with earlier works, overall our results seem consistent with other ML approaches. The drop of the efficiency at SNR $\sim 10 - 15$ is common for all algorithms (except for [317] that do not show this metric), which makes one wonder if there is some intrinsic limitation of ML algorithms that prevent to get closer to SNR ~ 8, a typical value for optimal template-matching algorithms. It could also be possible that more complex architecture or training sets with different pixel resolutions might improve the efficiency of this method. These are aspects that we would like to explore in the future.

8.6 Conclusion

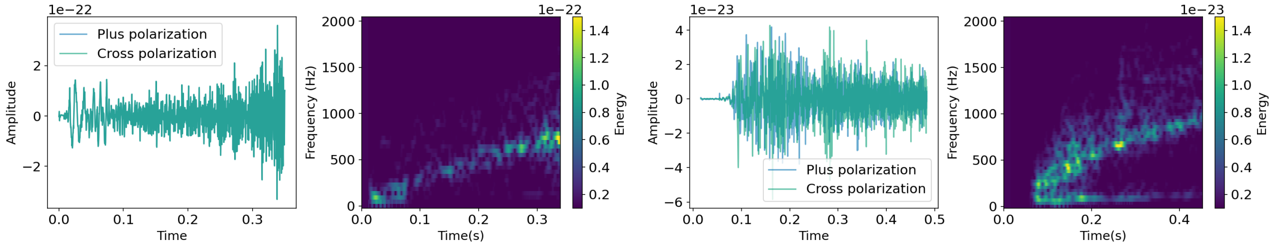
We developed a new ML algorithm to further improve the detectability of a GW signal from CCSN, following the path traced in [316]. Regarding the applicability of our method for the GW detection, we have considered a detection threshold, $\theta^* = 65\%$, that results in a FAR_{CNN} of about 5% at SNR ~ 15 (or a FPR of $\sim 10\%$ at TPR$= 50\%$). These values could be appropriate for an observation with high confidence of an event in coincidence with a neutrino signal. In those cases, the neutrino signal is expected to be bounded within 20 s during the initial SuperNova Early Warning System (SNEWS) alert [347] and very likely well within 1 s in the detailed analysis of high sensitivity neutrino detectors such as Super-K [348]. If the method were to be used in all-sky non-triggered searches, the range of values of false alarm rate needed to make a detection with high confidence could be achieved by using values of θ very close to 100%. The efficiency of the algorithm in this regime is something that could be explored in future work. Moreover, these results are promising for future detections of GWs from CCSN because the network allows us to observe more than half of the events within 15 kpc.

This work has multiple possible extensions. At present the entire data processing is rather fast: the training and validation phase, performed in the real detector noise, is done in 2 hours and 21 minutes using a GPU *Nvidia* Quadro P5000, while predicting the test set takes 3 ms for each 2 s long image. Given that we take advantage of the Keras/TensorFlow framework, widely used within the ML community, it should be easy to increase the complexity of our current CNN or to incorporate the latest developments in ML algorithms, with a reasonable increase in the computational cost of the signal search. Furthermore, we could increase the number of classes to be able to detect other GW sources with the same architecture.

In the future, this algorithm presented here will be compared under realistic conditions with the methods currently in use within the LIGO-Virgo collaboration to evaluate the real advantages of the method. While this line of research will be extended to O3, it is also important to improve the current phenomenological waveforms, as they have several limitations:

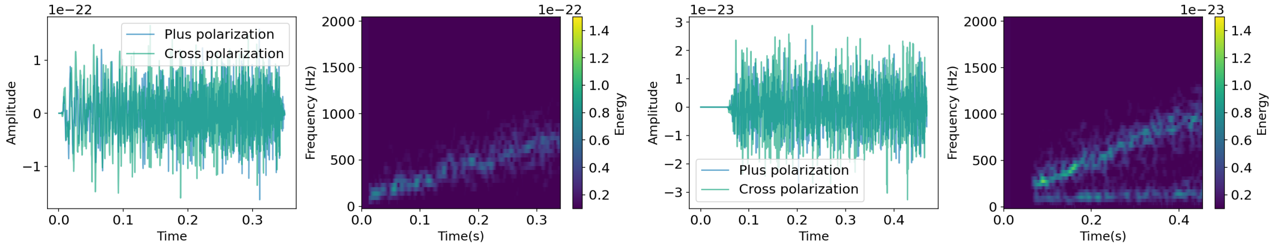
1. they do not include polarization, so they can be used as a prediction of the strain at one single detector, but cannot be used properly for networks of detectors.
2. they only contain the contribution of the g-mode to the signal but not additional modes that are expected to appear, such as SASI.
3. the parameter space and calibration used was based on a limited set of simulations and applied to a limited set of parameters being, for many parameters, just best guesses.

Therefore, we are currently extending this effort to produce phenomenological waveforms that overcome these issues, i.e. `ccphen` v4. In Fig. 8-17 we show the time domain representation and the time-frequency representation of the catalogue waveforms (top row, model TM1 on the left and model mesa20_v_LR on the right), as well as their optimized phenomenological waveforms (bottom row).



(A) Catalog waveform with g-mode: *model TM1*

(B) Catalog waveform with g-mode and SASI: *model mesa20_v_LR*



(C) `ccphen` v4 waveform with g-mode

(D) `ccphen` v4 waveform with g-mode and SASI

Figure 8-17: Catalog waveforms (top row) with their optimized phenological waveform from ccphen (bottom row) represented in time and time-frequency domain.

On one hand, the `ccphen` waveform with the g-mode in Fig. 8-17c can capture the general frequency trend of the model TM1 present in Fig. 8-17a, but is not as spread in frequency and it cannot reproduce the final peaks of the catalogue model. On the other hand, the `ccphen` waveform with g-modes and SASI in Fig. 8-17d captures the general frequency content of the g-modes and SASI of model mesa20_v_LR in Fig. 8-17b, but it lacks complexity in its frequency components and it does not match the start time of SASI of the catalogue waveform.

We expect that this improvement in the generation of phenomenological waveforms will enhance the interpolation abilities of our neural network.

Chapter 9

Enhancing Modelled Intermediate-Mass Binary Black Hole Searches

This is an unpublished chapter, where I led the pre-processing of the data, the fine-tuning of the models, the performance evaluation and the post-processing.

9.1 Introduction

The formation of supermassive black holes (SMBH) is not yet fully understood, despite their presence in nearly every galaxy, including the Milky Way [246, 247]. Although SMBHs could grow to their tremendous size over billions of years through fast accretion rates, SMBHs with masses of $10^9\,\mathrm{M}_\odot$ have been discovered in the early Universe at redshift $z > 6.5$, suggesting that other mechanisms might be involved in their rapid formation [349].

As stellar evolution predicts, stars with a helium core mass in the range $\sim 32 - 64\,\mathrm{M}_\odot$ are subject to pulsation pair instability, while stars with helium core mass in range $\gtrsim 50 - 130\,\mathrm{M}_\odot$ leave no remnant due to pair-instability supernovae [244, 245]. Because of this mass gap, a plausible explanation for the formation of SMBH is the hierarchical mergers of intermediate-mass black holes (IMBH) [248, 17]. Therefore, direct observation of IMBH populations would strengthen the possible evolutionary link between stellar mass black holes and SMBH.

The Advanced LIGO [5] and Advanced Virgo (V1) [6], detected 11 candidates during O1 and O2, none of which were IMBH candidates [11, 350, 351]. Indeed, the detection of IMBHs in GW searches remained elusive until the detection of GW190521 during O3 [12, 17]. The estimate of the individual mass components of GW190521 was $(m_1, m_2) = (85^{+11}_{-14}, 66^{+17}_{18})\,\mathrm{M}_\odot$, being the mass of the remnant $M_f = 142^{+28}_{-16}\,\mathrm{M}_\odot$, and making it the first conclusive direct observation of an IMBH [352].

Despite falling in the sensitivity band of current GW interferometers, IMBH searches are challenging as few cycles of the signal can be observed with current ground-based detectors. State-of-the-art searches employ weakly modelled algorithms, such as cWB (see Section 3.3.3) in its IMBH configuration [315, 128], or modelled algorithms using matched-filter-based techniques (see Section 3.2.1) [45]. Nonetheless, matched-filtering approaches are well-known for being computationally intensive, resorting to simplifications to reduce the computational cost, which might reduce their candidate significance [352]. Furthermore, IMBH searches are also hampered by glitches (refer to Section 2.4.3 for an overview).

As we discussed in Section 4.3, to enhance the sensitivity of current searches, a novel approach is to combine them with ML algorithms, fostering a synergistic relationship. With this idea in mind, it has been proposed to optimize matched-filtering with ML [207], and to optimize the performance of cWB providing information about the background [208, 209, 353, 210]. In a similar line of thought, in this work, we propose to enhance matched-filtering-based algorithms utilizing information provided by matched-filtering but performing a more flexible search of GW signals. Therefore, we employ the resulting *triggers* from the matched-filtering search (presented in Section 3.2.4) to train an ML algorithm and distinguish between glitches and simulated IMBH signals in O3 real single-detector data, using H1, L1 and Virgo (V1), (see Section 9.4.3). Afterwards, we construct a background of time shifts and assess the performance of the method with simulated IMBH signals (see Section 9.4.4). It is relevant to note that while this particular investigation focuses on IMBH, this method can be extended to other CBC signals.

9.2 Data

In Section 3.2.1 we provided an overview of the basic matched-filtering method, and in 3.2.4 we described how a trigger is generated by the matched-filtering algorithm used in this research, GstLAL, when a template waveform is cross-correlated with a signal. It often happens that different templates "match" an unknown signal buried within the time series $s(t)$. As a result, N triggers are "matched" or associated with such unknown signal within a finite time window Δt, which we refer to as *cluster* of triggers. After recording all such triggers, GstLAL selects the loudest SNR trigger within the cluster (clustering step), which we refer to as the *centroid*. Afterwards, these centroids are ranked according to GstLAL likelihood function Λ to claim a candidate [125].

Given the morphological time-frequency differences between glitches and GW signals, we expect the "matched" templates to exhibit a meaningful structure to differentiate them with ML methods. We refer to this inherent structure as *cluster tracks*, or simply *tracks*. In particular, we are interested in tracks generated in single-detector, right after the matched-filtering step. Hence, to maintain as much information as possible, we reproduce the IMBH search of GstLAL during O3 [17], terminating the process before the clustering step.

This work aims to utilize supervised learning (see Section 4.1) to differentiate GW from glitches. Thus, we construct two different data sets: a *controlled* data set which contains well-known glitches and simulated IMBH signals, referred to as *known data set*; and a second data set which contains real GW signals and other *a priori* unknown signals, which we use to construct an accidental background of time shifts (see Section 9.2.4 for details), referred to as *unknown data set*. While the first data set is employed to assess the performance of the method, the second data set is used to understand the significance of the ML statistic.

In the following subsection, we describe the simulated IMBH signals, and the different glitch classes present in the *known data set*. In Section 9.2.2 we describe the clustering procedure utilized for defining a track. In Section 9.2.3 we show the patterns generated by an IMBH simulation and a glitch through the template bank, and we define the feature vector associated with the triggers that will be the input to the ML model. Last but not least, in Section 9.2.4 we discuss how we construct the accidental background.

9.2.1 Injections and glitches

Since we want to distinguish between IMBH signals and glitches in a controlled environment, namely the *known data set*, we inject simulated IMBH signals in O3 real detector noise, labelled as the *Injection* class. For this aim we use `IMRPhenomD` approximant [98] with masses

$m_1 \in [50, 400]\,\mathrm{M}_\odot$ and $m_2 \in [10, 250]\,\mathrm{M}_\odot$, and dimensionless spins $\chi_1, \chi_2 \in [0, 0.99]$, uniformly sampled within the specified range with aligned spins. The distances are uniformly sampled from the range $d \in [10, 250]$ Mpc. The inclination is sampled from a uniform distribution, and the sky-localization is randomized.

Furthermore, we select six different glitch populations from the data set of `Gravity Spy` (see Section 6.2.1) during O3, namely *Blip*, *Fast_Scattering*, *Koi_Fish*, *Low_Frequency_Burst*, *Tomte* and *Whistle* (refer to Section 2.4.3 for details). Note that these glitches have been selected with a `Gravity Spy` confidence $> 90\%$, and they are chosen to have short and long durations, a wide frequency contribution, and abundant occurrences in current ground-based interferometers.

9.2.2 Clustering

As we discussed in Section 9.2 we employ the *known data set*, whose classes were defined in the previous section. In this study, we are interested in exploring the generalization power between the first and second half of the third observing run, namely O3a and O3b, respectively. For this aim, we use the *known data set* from O3a for training, validation and testing (see Section 9.4.2), while the *known data set* of O3b is used to test this generalization power (see Section 9.4.3). In the following, we describe how the cluster tracks of these data sets were defined.

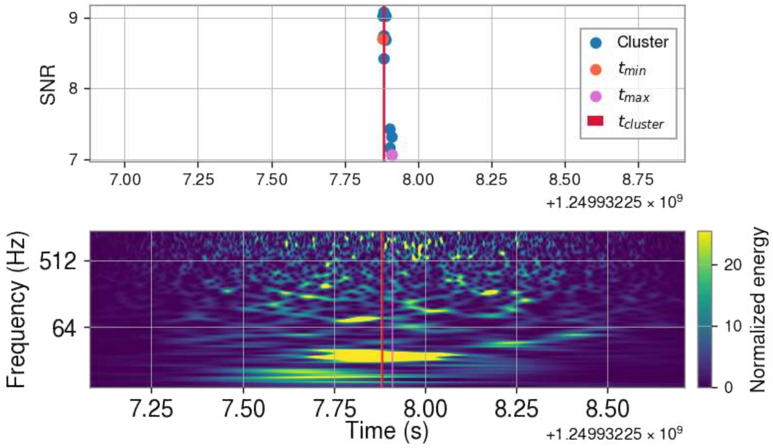


Figure 9-1: A Fast Scattering glitch labelled by `Gravity Spy`*. (Top) SNR of GstLAL tracks as a function of time, where each point represents a template. t_{min} (orange) and t_{max} (pink) mark the beginning and end of the cluster, respectively, while $t_{cluster}$ (red) indicates its centroid given by* `Gravity Spy`*. (Bottom) Q-transform of the time series containing the Whistle labelled by* `Gravity Spy`*.*

- **Known data set**: Since we know the GPS time when an *Injection* or a glitch occurred, we use it as the centroid of the cluster, $t_{cluster}$, and we select a window $t_w = \pm\,\Delta t$ around it. We have experimented with $\Delta t \in \{0.05, 0.1, 0.2, 0.5, 1\}$ s to find the optimal t_w of the cluster since IMBH signals are short. It is relevant to note that while we experiment with different t_w, once the algorithm has trained with a given t_w, this value is fixed for validation and testing. We provide a thorough discussion on the selection of optimal t_w in Section 9.4.1.

- **Unknown data set**: Usually clusters have a centroid with the highest ρ, and a set of neighbours with smaller ρ. For illustration, in the top panel of Fig. 9-1 we can observe the SNR of a cluster with this behaviour. In a realistic setting, we do not know the GPS time of the centroid of the cluster $t_{cluster}$ *a priori*, so we define them following a procedure similar to PyCBC [115]. We order all the triggers according to their GPS time.

> Afterwards, we divide the whole observing run in windows of ±1 s, starting from the beginning of the run. Within each window, we select the trigger with maximum ρ as the centroid, so that $t_{cluster} = t_{\max\rho}$. As before, we select $t_w = \pm\Delta t$ around $t_{cluster}$ to define the cluster.

While the methodology outlined above provides a systematic approach to identifying and analyzing matched-filtering clusters, it is a simplified approach with several limitations. One significant caveat is the assumption that the trigger with the highest SNR within a given window accurately represents the true centroid of a cluster, particularly in the *unknown data set*. This approach may lead to inaccuracies if multiple signals or glitches overlap within the window, potentially causing the algorithm to misidentify the centroid or miss other relevant triggers with slightly lower SNR. Additionally, the fixed window size, although optimized for specific signal characteristics like IMBH signals, might not be ideal for unexpected GW signals of unknown length, leading to suboptimal clustering in some cases. These factors could introduce biases or reduce the accuracy of the clustering, highlighting the need for further refinement and validation of the clustering algorithm in future work.

9.2.3 Trigger tracks

As we mentioned in Section 3.2.4 when the maximum SNR ρ associated with a template is over a certain threshold, it will produce a trigger with intrinsic parameters λ_{int} associated. In this particular work, we reduce the single-detector λ_{int} associated with the ith template forming the feature vector ϕ_i,

$$\phi_i = \{\rho_i, \xi_i, m_{1,i}, m_{2,i}, s_{1z,i}, s_{1z,i}\}, \tag{9.1}$$

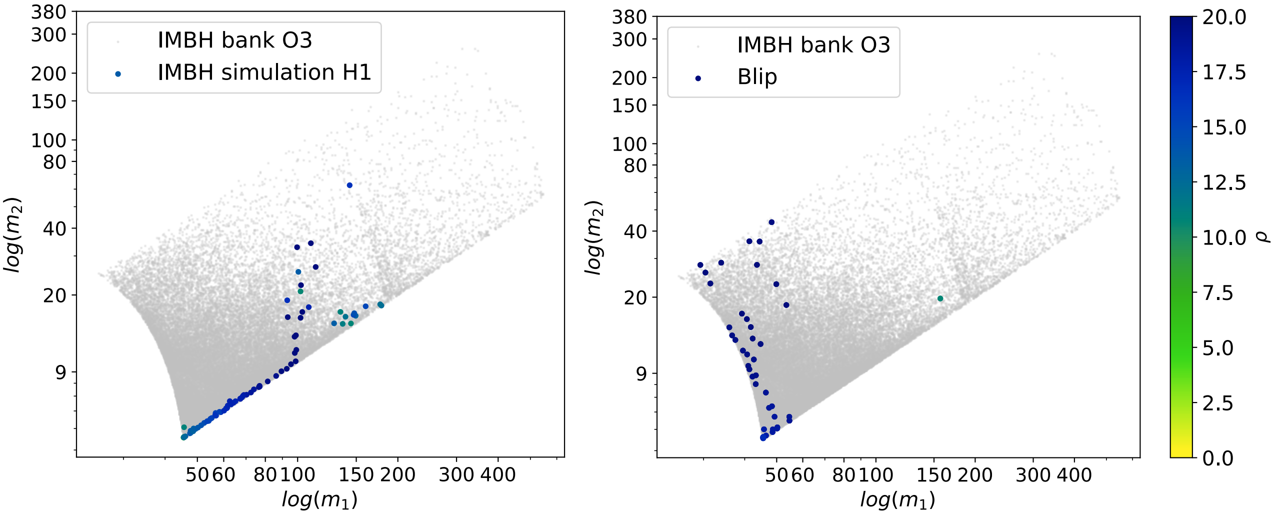
containing the SNR ρ, the consistency check ξ (see Eq. 3.23), the masses of the progenitors (m_1, m_2) and the z-component of their spins (ξ_1, ξ_2), respectively. As we mentioned before, an unknown signal $s(t)$ might have an associated *track*, so to illustrate this scenario we show in Fig. 9-2 the tracks generated by an *Injection* (Fig. 9-2a) and a *Blip* glitch (Fig. 9-2b) projected in the mass and SNR. We show the O3 IMBH template bank as a function of the progenitor masses m_1 and m_2, where every dot represents a template. We also colour the tracks, where their colour is related to the maximum SNR, ρ. As we can see in Fig. 9-2a, the *Injection* matches a concrete space in the low-mass region at $\rho \sim 10$, and a slightly sparse space in the higher-mass region at $\rho \sim 20$. On the other hand, in Fig. 9-2b the templates matching the *Blip* glitch are at $\rho \sim 12$, being sparse in the progenitor mass dimension. Assuming that the distribution of GW and glitch tracks possess distinct underlying features, we can employ ML methods to differentiate them.

9.2.4 Time shift and false alarm rate

As we discussed in Section 5.1, it is fundamental for a search algorithm to evaluate the accidental background. Unlike in Chapter 8, where we assess the number of FPs in the context of the ML algorithm, in this chapter we also construct a time-shifted background and evaluate the false alarm rate using Eq. 3.42. Since the offsets of the time shifts should be large enough that each slide can be thought of as an independent experiment, we chose to slide L1 in steps of 3 s, and V1 in steps of 6 s with respect to H1. We performed

Table 9.1: Total time of search (T_s) for different detector combinations, measured in years.

	Time of Search (T_s)
H1L1 (no V1)	0.0000382
H1V1 (no L1)	0.1117765
L1V1 (no H1)	0.1338800
H1L1V1	0.5619618



(A) Triggers of *Injection* class

(B) Triggers of *Blip* class

Figure 9-2: O3 IMBH template bank as a function of the progenitor masses in logarithmic scale, where every grey point represents a template from the bank. Triggers associated with an Injection or a Blip glitch are coloured according to the maximum SNR, ρ.

enough time shifts to produce 1,000 yr of data in triple detector coincidence. Afterwards, the time shifts are binned according to their coincidence type and their time of background.

To minimize the FPs, the types of detector time coincidences analyzed in this work are either double or triple. Using 3.42, it is necessary to properly measure the time of the search T_s and the time of the background T_b. If we are measuring the time in double-time coincidence, for example, between H1 and L1, it would imply that V1 was down at that time (no V1). Otherwise, if all detectors record data simultaneously, it is considered a triple-time coincidence. In Table 9.1 we can see an overview of the time of the search T_s. Since H1L1 (no V1) is extremely small, it is computationally intensive to generate 1000 yr of time shifts. Thus, in the interest of time and computational resources we focus on triple-coincident time as we expect a better performance.

As in a standard search, to assess if two tracks are coincident, we only consider tracks within the time coincidence window. For example, using the H1L1V1 time-coincidence, we can have tracks that occur in all detectors simultaneously and tracks that occur only in H1 and L1. Thus, we will have double coincident tracks, such as H1L1, during H1L1V1 time, as well as triple coincident tracks in H1L1V1 during H1L1V1 time.

9.3 Methodology

In Section 4.3 we have discussed how the challenges in GW research require innovative solutions, and how ML has emerged as a crucial tool for addressing them due to its adaptability and transversality. Following similar thoughts to [207], where the authors seek to enhance current state-of-the-art search algorithms with ML applications, we use the triggers generated by the IMBH search of GstLAL during O3 to train a classification algorithm for multiple classes. In this way, we output a statistic to differentiate between IMBH signals and glitches, providing direct information about the nature of these populations.

9.3.1 Feature vector

Under the assumption that different signals have different track structures, it is possible to learn them with ML. Tracks have data structures known in ML as *multi-instance representation*, where N feature vectors ϕ_i, also known as triggers, represent a given example. This implies that each track may contain a different number of triggers for each example. Consequently, the input length to our ML model changes from one example to another. This poses a limitation because standard ML techniques only accept fixed-length inputs. As a proof-of-concept to overcome this obstacle, we average all the feature vectors ϕ_i associated with a given signal by weighting them with ρ, resulting in the following input feature

$$\Phi = \{\bar{\rho}, \bar{\xi}, \bar{m}_1, \bar{m}_2, \bar{s}_{1z}, \bar{s}_{1z}\} \quad \text{where } \bar{x} = \frac{\sum_{i=1}^{N} x_i \rho_i}{\sum_{i=1}^{N} \rho_i}, \tag{9.2}$$

for $i \in 1, \ldots, N$ the triggers associated with each signal. In this way, we give more weight to the template that best "matches" a given signal. This weighted average enhanced the performance of our ML model with respect to a standard average.

9.3.2 Tackling class imbalance

Table 9.2: Original data set size before sampling

	Hanford (H1)	**Livingston (L1)**	**Virgo (V1)**
Injections	85107	101307	54436
Blip	2717	1701	1534
Fast_Scattering	114	18589	-
Koi_Fish	5147	4061	731
Low_Frequency_Burst	1523	109	3044
Tomte	537	18619	674
Whistle	1946	95	246

While we can simulate a large number of GW signals, glitches have a finite nature. Despite their occurrence rate being approximately $\sim 1\,\mathrm{min}^{-1}$, some classes of glitches are more common than others [12]. For example, in L1 there are $> 10^4$ *Tomtes*, while there are $\sim 10^3$ *Whistles*. In Table 9.2 we show the original size of our training and validation set, which is highly imbalanced. This is problematic for classification ML algorithms, as heavily imbalanced datasets tend to be biased towards the majority class to minimize their loss function. This may wrongly be interpreted as a "good" performance, while in reality, the model is only learning a single class: the largest one. Similarly to [282], to circumvent this issue we use undersampling or oversampling techniques:

- **Undersampling:** In our data set, the number of *Injections* performed is significantly higher than the largest glitch class. Thus, we randomly sample this class to match the size of the oversampled glitches.
- **Oversampling:** To oversample we use bootstrapping with replacement, which stochastically resamples the existing data set allowing the same example to appear more than once [354]. Using this method, we oversample all glitch classes to equal the size of the largest glitch class.

While this method improved the performance of our ML method, it is important to realize that bootstrapping with replacement does not generate new data nor gives new information about the classes. If the original data set does not represent the class population, it will bias the ML algorithm producing overfitting.

To avoid overfitting we use *K-fold cross-validation* [355]. This method partitions the data set in K subsets or folds (see Fig. 9-3). The model is evaluated K times, using a different fold as validation in each iteration, while training on the rest of the folds. This allows for a more robust estimate of the model's performance, as it accounts for the variability in the data and limits the potential for overfitting. In particular, we chose $K = 9$ as it is a trade-off between computational complexity and performance.

As we mentioned in Section 9.2.2 we want to test the generalization ability of the MLP (see Section 4.2.1) for O3a and O3b. Therefore, we use the *known data set* of O3a for the standard ML procedure of training, validation and testing, splitting the data set into 80% for training, 10% for validation and 10% for testing with the K-fold cross-validation method. Once we have tested in *known data set* of O3a, we use the whole *known data set* of O3b to test the generalization ability of the MLP.

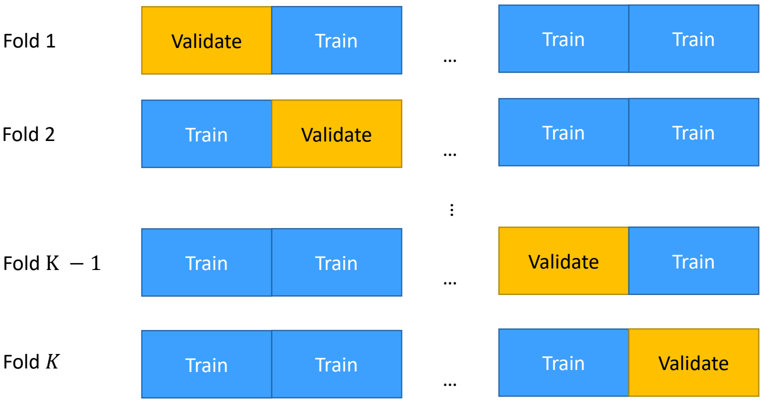


Figure 9-3: *Illustration of K-fold cross-validation.*

9.3.3 Training model

To differentiate IMBH signals from glitch classes (see Sections 2.4.3 and 9.2.1), we construct an MLP (discussed in Section 4.2.1) for each detector, that inputs the feature vector Φ (see Eq. 9.2) and outputs a probability vector that indicates the likelihood for each class. The size of this probability vector depends on the detector: we distinguish 7 classes for H1 and L1, but 6 classes for V1, as *Fast_Scattering* class is not present in that detector.

We implemented the MLP structure in `PyTorch` [356]. After several experiments, we selected the best-performing architecture from our results: it consists of 3 hidden layers of 350 each, using the ReLU activation function [302], except the output layer that uses Softmax activation function, since it is a multi-class task (see Section 4.2.1). This MLP architecture is common for each GW detector, but they have been trained separately, i.e. it is a single-detector classifier. To optimize the networks we use cross-entropy loss function, and Adam optimizer [345]. Moreover, to adjust the number of epochs and avoid overfitting we implemented an *early stopping* algorithm [357]. We define an epoch as the number of times the network has passed through the whole training and validation data set. Thus, the early stopping algorithm calculates the difference in validation accuracy $\mathcal{A}_{val}$ between the current epoch, e, and the best epoch, e^{best}. The training process finishes if

$$|\mathcal{A}_{val}^{e} - \mathcal{A}_{val}^{e^{\text{best}}}| \leq \epsilon, \tag{9.3}$$

during 150 epochs, with $\epsilon = 0.0001$. For the learning rate, we use an *adaptive learning rate* built-in the `PyTorch` function `ReduceLROnPlateau`. Setting the initial learning rate to 10^{-3}, if the validation accuracy remains constant after 100 epochs, the learning rate decreases 10%. The combination of these methods proved to increase the performance while decreasing the time needed for fine-tuning.

9.3.4 Time coincident tracks

Even though the MLP receives direct information from the *Injection* class and various glitch classes, misclassifications are still common due to the simplicity of the feature vector (see Section 9.3.1). Nonetheless, if a potential GW signal is detected in multiple detectors, the likelihood of it being of astronomical origin increases significantly. Hence, it is standard in GW searches to evaluate triggers that happened within the light time travel between detectors: 10 ms between H1 and L1, 27 ms between H1 and V1, and 26 ms between L1 and V1. An additional 5 ms is considered for statistical fluctuations [106].

For the *Injection* class of the *known data set*, we can identify coincident tracks because we have access to the ground truth. However, for the glitches of the *known data set* and the *unknown data set* itself, this information is not available *a priori*, so we need to define criteria for when two tracks are time coincident. In a standard matched-filtering search, a time coincident trigger occurs when the same template triggers in different detectors within the light travel time between them.

In this work, the approach is inherently different, as we do not deal with individual triggers but with averaged tracks. Thus, two tracks from different detectors are considered time coincident if the average time of the triggers within each track falls within the light travel time and time fluctuations. Nevertheless, even if two tracks are time coincident, there is no guarantee that they originate from the same signal. Therefore, two tracks are only considered time coincident if they have at least one common trigger. If tracks from different detectors coincide in time, we assign to the coincidence a probability equal to the harmonic mean, one of Pythagorean means [358, 359], of each trigger, relabelling our statistic for N detectors as

$$\bar{P}_{inj} = N / (\sum_{k}^{N} P_{inj,k}^{-1}) \tag{9.4}$$

where P_{inj} is the probability of being an *Injection* (see Section 9.2.1 for a description). It is important to note that the MLP performs a single-detector inference, and the time coincident step is computed independently afterwards. In future works, it would be interesting to provide the full information of the time coincident tracks to a ML algorithm. This integration could enhance the model's ability to distinguish GW signals from glitches.

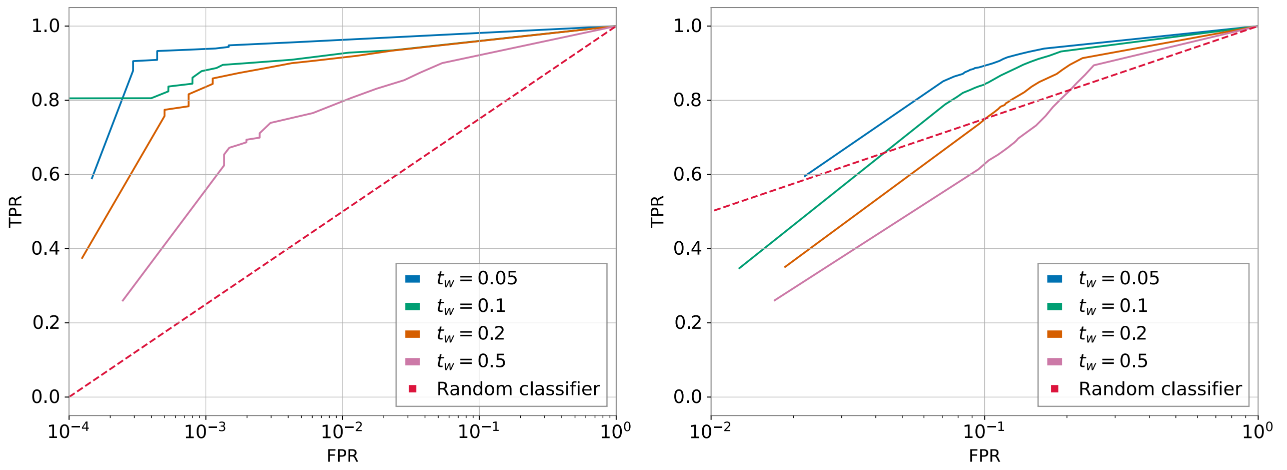
9.4 Results

In the present Section, we show the results of the performance of the MLP model with H1 data. The results for L1 and V1 can be found in the Appendix A.5. Firstly, in Section 9.4.1, we describe the selection of the time window t_w for the *known data set* of O3a, i.e. the controlled data set of O3a composed of simulated GW signals and well-known glitches. Afterwards, in Section 9.4.2, we show the results of the training and validation with the previous data set. In Section 9.4.3 we test the model with the *known data set* of O3a and O3b. In Section 9.4.4 we assess the significance of the ML inference with the accidental background of time shifts.

9.4.1 Selecting a time window

As mentioned in Section 9.2.2, we explored different time windows t_w for each GW detector, namely H1, L1 and V1. To evaluate the performance of the algorithms without fine-tuning them we use the ROC curve, which is represented as the TPR as a function of FPR [360]. A relevant point is that our task is a multi-class classification, so to compute the ROC curve, which is usually employed in binary classification, we need to reduce our problem to a pairwise comparison, i.e. *Injection* class (positive class) against all other classes (negative class).

Furthermore, it is custom in ML to calculate the area under the ROC curve as an evaluation metric, since models with a larger area under the curve have a better performance. However, in the field of GW, it is required to know the performance of the model at low FPR as we want to minimize the number of FPs when claiming detection, i.e. the number of glitches



(A) Testing on O3a with *known data set*. (B) Testing on O3b with *known data set*.

FIGURE 9-4: ROC curve for different time windows t_w in H1, i.e. TPR as a function of FPR in logarithmic scale. The positive class is Injection class, while the negative class is any of the other classes. (Top) Testing in the known data set of O3a. (Bottom) Testing in the known data set of O3b. Note that the dashed line indicates a random guess.

incorrectly classified as GW signals. For this aim, we select a grid of decision thresholds θ^* in range $[0.1, 0.9]$ with a spacing of 0.1 and in range $[0.9, 1.0]$ with a spacing of 0.01. When a given probability exceeds this value, we classify the input as a positive, i.e. *Injection* class, otherwise, it is classified as negative, i.e. any other class. As this grid of decision threshold is not usual in ML, the area under the curve has a different magnitude, so instead we select the t_w with the best performance in *known data set* of O3a and O3b.

In Fig. 9-4 we present the ROC curves of H1 for the *known data set* of O3a (Fig. 9-4a) the *known data set* of O3b (Fig. 9-4b). In Fig. 9-4a we can observe that the TPR degrades as we increase the size of t_w. In Fig. 9-4b we can see a similar behaviour but a higher FPR range.

However, at FPR $< 10^{-1}$ the performance drops dramatically below the random classifier line (dashed line), which implies that the model is making random guesses. As we increase the size of t_w, the performance of the algorithm decreases. This behaviour could be caused by the short duration of IMBH signals, meaning that if we define t_w to be large, the clusters will contain random triggers that will bias the classification task. Furthermore, this behaviour is also observed in L1 and V1 (see Appendix A.5.1 for details), so we conclude

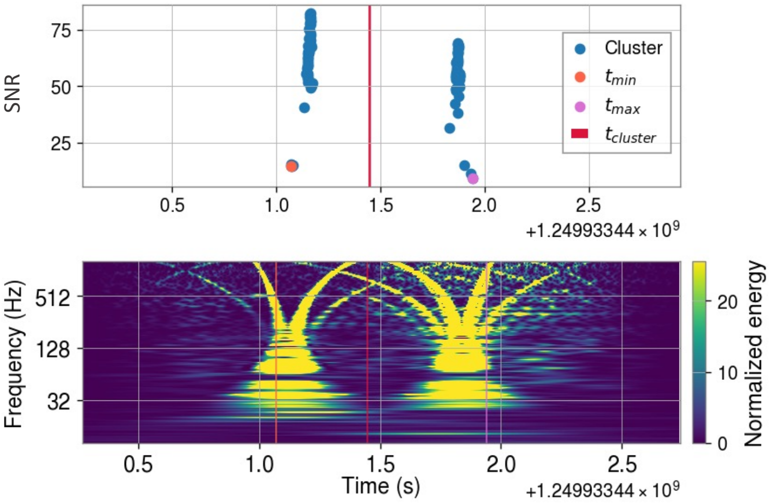


FIGURE 9-5: A Whistle glitch labelled by ***`Gravity Spy`****. (Top) SNR of GstLAL tracks as a function of time, where each point represents a template. t_{min} (orange) and t_{max} (pink) mark the beginning and end of the cluster, respectively, while $t_{cluster}$ (red) indicates its centroid given by* ***`Gravity Spy`****. (Bottom) Q-transform of the time series containing the Whistle labelled by* ***`Gravity Spy`****. While* ***`Gravity Spy`*** *labels a single glitch, GstLAL identifies both.*

that $t_w = 0.05\,\mathrm{s}$ is the best time window for our task.

It is relevant to note that one limitation of this method is the lack of ground truth when defining the centroid $t_{cluster}$ of a glitch. As an example, in Fig. 9-5 we show a *Whistle* labelled by `Gravity Spy`. The time centre of the cluster, $t_{cluster}$, is marked in red, and according to `Gravity Spy`, it is associated with a single *Whistle*. GstLAL identifies two clusters that correspond to two *Whistle* instead, as can be seen from the bottom panel. In future works, it would be relevant to study the effect of this offset on the behaviour of the model.

9.4.2 Training with *known data set* of O3a

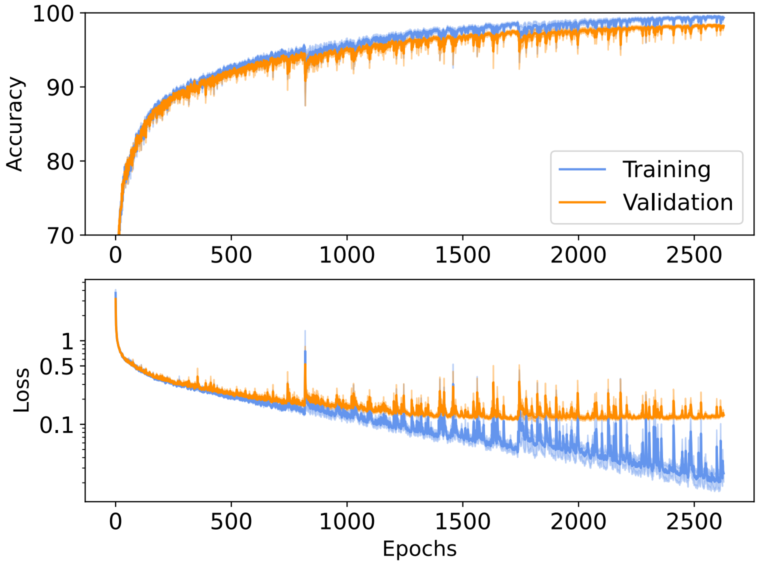
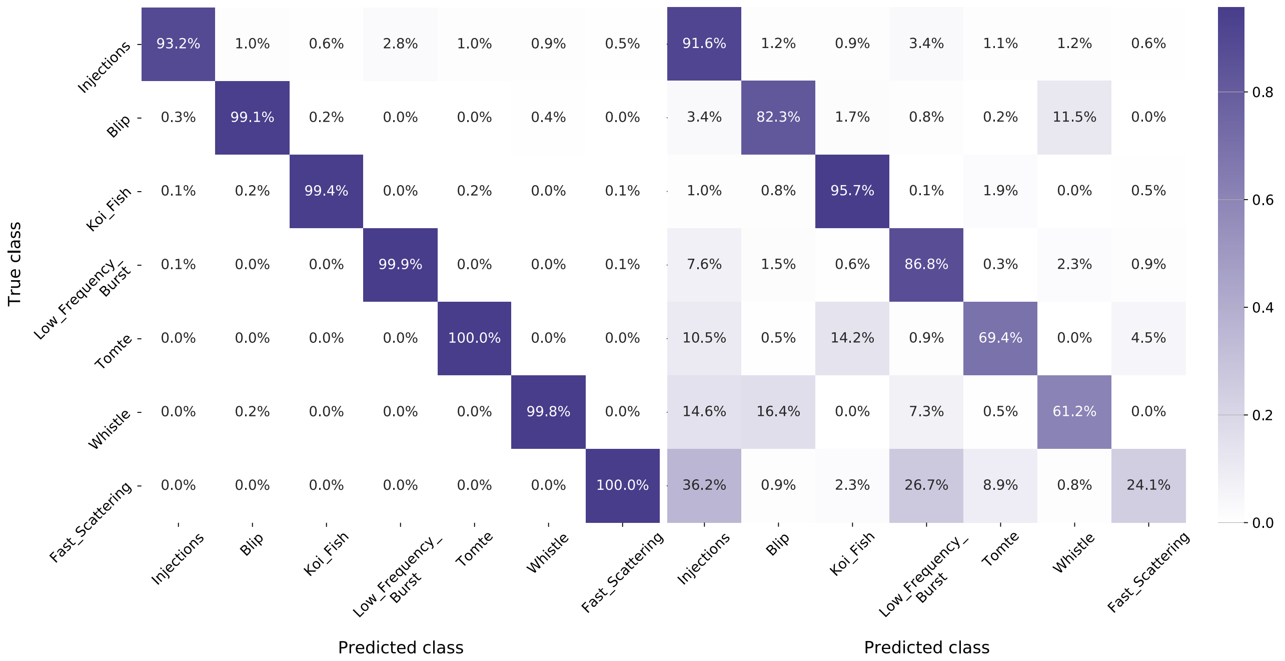


Figure 9-6: Comparison between training and validating with 9-fold cross-validation for H1 during O3a. (Top) Mean accuracy at 3 standard deviations (shaded region) as a function of the epochs during training and testing. (Bottom) Average 9-fold cross-validation loss as a function of the epochs during training and testing.

Once we have selected t_w, and after several fine-tuning experiments, we can train and validate our model with the *known data set* of O3a. As we mentioned in Section 9.3.2, we use 9-fold cross-validation to enhance the performance of the algorithm. Thus, in Fig. 9-6 we show, for training and validation, the mean accuracy (top panel) and loss (bottom panel) of the 9-fold cross-validation as a function of the epochs, where the shaded region represents ±3 standard deviations. Note that the loss is plotted in logarithmic scale so we can appreciate the difference between training and validation loss is minimal, which implies that the model is not overfitting. Another relevant point is that both mean accuracy and loss seem to have sharp peaks around epochs 900 and 1700, which could imply that the learning is unstable. However, this is a known effect of the adaptive learning rate that we described in Section 9.3.3. The described behaviour is also present in L1 and V1, whose results can be found in Appendix A.5.2.

9.4.3 Diving into the *known data*: machine learning performance

Employing GPU Tesla V100 with a memory of 16 Gb allowed us to train our model in ≈ 17.40 h for H1 data, ≈ 15.55 h for L1 data, ≈ 31.20 h for V1 data. Such a large amount of time is mainly due to the 9-fold cross-validation procedure. Nonetheless, once the models are trained we can predict a single input in $2.9 \times 10^{-6}\,\mathrm{s}$. As we also want to test the generalization power of the model between O3a and O3b *known data set*, we present their confusion matrix for H1 in Fig. 9-7 (see results for L1 and V1 in Appendix A.5.3). While in the y-axis we represent the ground truth, in the x-axis we represent the predictions of the model. Furthermore, the percentages in the rows sum 100%, and the elements in the diagonal were successfully classified.



True class	Injections	Blip	Koi_Fish	Low_Frequency_ Burst	Tomte	Whistle	Fast_Scattering
Injections	93.2%	1.0%	0.6%	2.8%	1.0%	0.9%	0.5%
Blip	0.3%	99.1%	0.2%	0.0%	0.0%	0.4%	0.0%
Koi_Fish	0.1%	0.2%	99.4%	0.0%	0.2%	0.0%	0.1%
Low_Frequency_ Burst	0.1%	0.0%	0.0%	99.9%	0.0%	0.0%	0.1%
Tomte	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%
Whistle	0.0%	0.2%	0.0%	0.0%	0.0%	99.8%	0.0%
Fast_Scattering	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%

True class	Injections	Blip	Koi_Fish	Low_Frequency_ Burst	Tomte	Whistle	Fast_Scattering
Injections	91.6%	1.2%	0.9%	3.4%	1.1%	1.2%	0.6%
Blip	3.4%	82.3%	1.7%	0.8%	0.2%	11.5%	0.0%
Koi_Fish	1.0%	0.8%	95.7%	0.1%	1.9%	0.0%	0.5%
Low_Frequency_ Burst	7.6%	1.5%	0.6%	86.8%	0.3%	2.3%	0.9%
Tomte	10.5%	0.5%	14.2%	0.9%	69.4%	0.0%	4.5%
Whistle	14.6%	16.4%	0.0%	7.3%	0.5%	61.2%	0.0%
Fast_Scattering	36.2%	0.9%	2.3%	26.7%	8.9%	0.8%	24.1%

(A) O3a *known data set*..

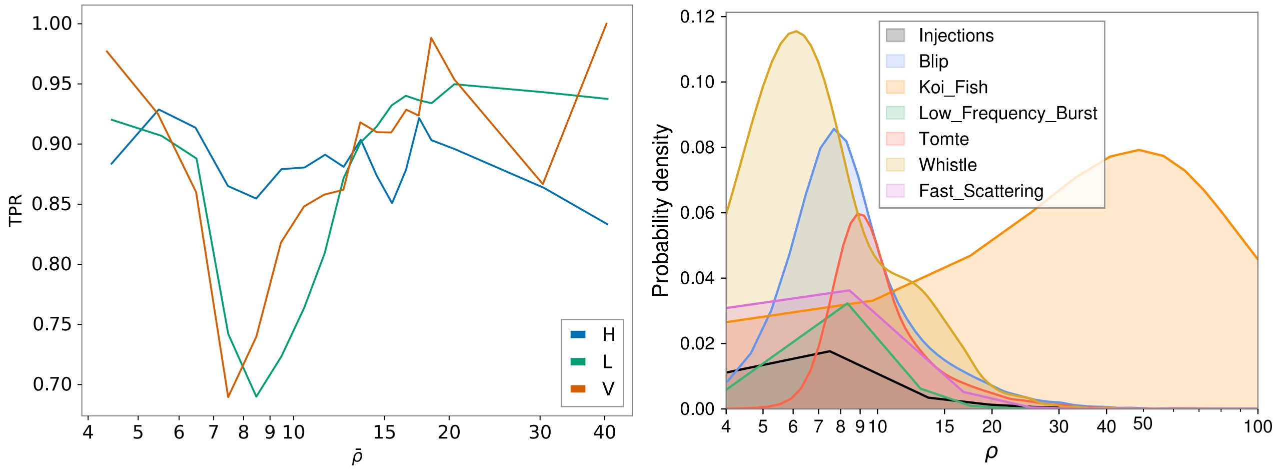
(B) O3b *known data set*.

FIGURE 9-7: Relative values of the confusion matrix for the test set for H1. The y-axis represents the groud-truth, i.e. the true class, and the x-axis represents the prediction of the model, i.e. the predicted class.

In Fig. 9-7a we show the confusion matrix of the test using the O3a *known data set* for H1, where the accuracy is 98.8%. We can see that 93.2% of *Injections* were correctly classified with little amount of misclassifications. However, for the O3b *known data set* the accuracy decreases sharply to 73.0%. We can see in Fig. 9-7b that some of the glitches are wrongly classified as *Injections* or other classes. In particular, the *Fast_Scattering* class seems to be the most problematic, as 36.2% of them seem to be classified as *Injections* and 26.7% as *Low_Frequency_Burst*. This lack of generalization between O3a and O3b is common to L1 and V1, where the accuracy in O3a is 95.8% and 99.3%, and the accuracy in O3b decreases to 67.5% and 75.9%, respectively. The misclassification of *Fast_Scattering* is also present in L1, but the misclassification of *Low_Frequency_Burst* as *Fast_Scattering* is more acute. Interestingly, in V1, the most problematic is the *Whistle* class (see Fig. A-8 and Fig. A-9 in Appendix A.5.3).

To assess the degree of misclasification during O3b with respect to the SNR distribution we compute the TPR as a function of the average SNR $\bar{\rho}$ (Fig. 9-8a). For this aim, we arrange the *Injections* in SNR bins and compute their TPR setting the decision threshold to be $P^{*}_{inj} = 0.9$. Interestingly, we can observe that the TPR is high for low $\bar{\rho}$, showing dip at $\bar{\rho} \sim 8$ of TPR ~ 0.85 for H1 , and TPR ~ 0.7 for L1 and V1. Moreover, the TPR becomes unstable at $\bar{\rho} > 15$.

To explain this behaviour we need to observe the SNR ρ distributions, computed by Omicron [73], of the different classes for the *known data set* of O3b (Fig. 9-8b). Firstly, it is important to note that `Gravity Spy` classifies glitches with Omicron SNR > 7.5. In Fig. 9-8b we show the distribution of ρ, the resulting SNR from Omicron, where we can still observe that most glitch populations are centred around $\rho \sim 8$, which explains the dip in Fig. 9-8a. Our method understands that most classes of problematic glitches, such as *Fast_Scattering*, *Tomte* and *Low_Frequency_Burst* are around this value, so it does not have trouble distinguishing *Injections* at lower ρ. Note that at $\rho < 5$ the model simply differentiates between *Whistle*, *Fast_Scattering* and *Injection* classes. Furthermore, as ρ increases, the population of *Injec-*



(A) True positive rate (TPR)

(B) SNR ρ probability density of different classes.

Figure 9-8: (Top panel) Single detector TPR as a function of the average SNR $\bar{\rho}$ in the known data set of O3b for H1, L1 and V1 . (Bottom panel) SNR ρ probability density of different classes in the known data set of O3b.

tions decreases, so due to the low statistic at $\bar{\rho} > 15$ the TPR seems unstable. We note that a different SNR distribution for the *Injection* class might lead to a different behaviour of the TPR as a function of the SNR. We will leave this exploration to future work.

A possible cause for this lack of generalization between O3a and O3b can be the fact that the interferometers are evolving systems, so the glitches produced at the beginning of the observing run can be different from the ones produced at the end. Furthermore, we remind the reader that while the *Injections* class is the actual ground truth, there is some bias in the definition of the glitch classes (e.g. Fig. 9-5). Another reason is the limitation of the model itself, as we are performing a multi-classification task with information on 6 variables (see Eq. 9.2). To lower the number of FPs, i.e. glitches that are incorrectly classified as *Injections*, in this single-detector task we can use time coincidence between detectors (see Section 9.3.4 for a definition), since it is less common for glitches to appear in several interferometers at the same time.

In Fig 9-9 we present the probability of being an *Injection* (P_{inj}) for the *known test set* of H1, in blue, L1, in red, and their time coincidence, in purple. In the top panel, we present the probability density of the *Injection* class, while in the bottom panel, we show the probability density of the glitches. In the top panel, we can observe that, for both detectors, while the MLP model classifies many *Injections* as such with large P_{inj}, there are still many *Injections* that have a low P_{inj}, which can be misclassified as glitches. Conversely, the MLP model gives many glitches a low P_{inj}, but there are still many glitches with a high P_{inj}. To increase P_{inj} of *Injections* and lower the P_{inj} of glitches, we use time coincidence. Hence, enforcing time coincidence increases P_{inj} of *Injections* (top panel), while completely discarding the glitches (bottom panel). Moreover, under the assumption that tracks with a low number of triggers are produced by detector noise we limit our analysis to tracks with 10 triggers or more.

Time coincidence, together with track reduction, discards many *Injection* tracks: while we have 21900 tracks in H1 and 43100 tracks in L1, their coincidence yields only 8967 tracks. As before, in Appendix A.5.3 we present the coincident results of the pairs L1 and V1, and H1 and V1, where an identical behaviour can be observed.

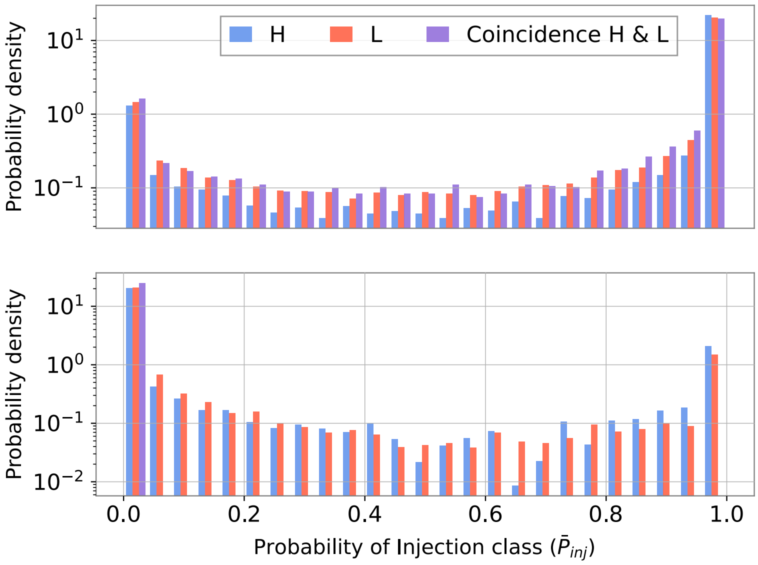


FIGURE 9-9: Probability density of being Injection, using the known test set of O3b for H1, in blue, L1, in red, and their time coincidence, in purple. (Top) Probability density of elements in the Injection class in logarithmic scale. (Bottom) Probability density of elements in all the other classes, i.e. glitches, in logarithmic scale. Given the counts of the ith bin c_i and its width b_i, we define the probability density as $c_i/(\sum_i^N c_i \times b_i)$, where N is the total number of bins of the histogram.

9.4.4 Diving into the ***known data***: significance of P_{inj} statistic

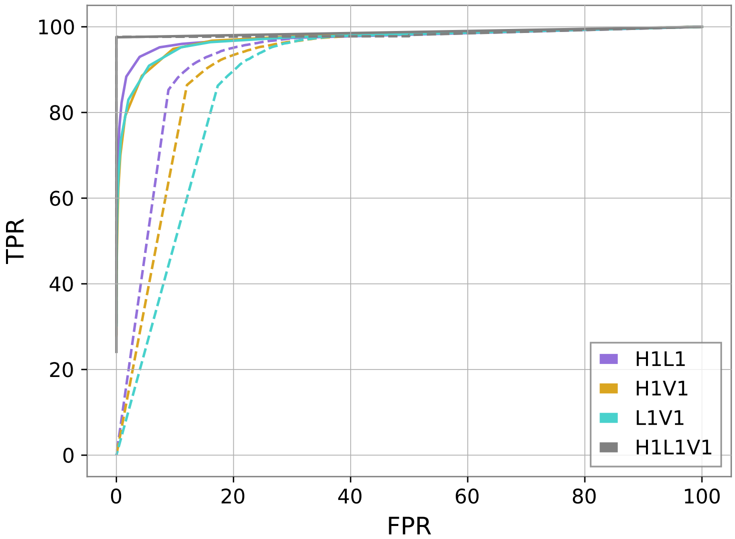


FIGURE 9-10: ROC curve of double and triple detector time (H1L1V1) for Eq. 9.5 (dashed lines) and Eq. 9.6. Here, TPR is the rate of correctly classified GW injections from the zero lag, while FPR is the rate of incorrectly classified time slides background tracks.

Up until now, we performed the standard statistical tests in the field of ML with P_{inj} as our probability. Nonetheless, as we discussed in Section 5.1, a ML based detection algorithm requires a proper understanding of the significance of its statistics considering the accidental background. With the time-shifted background, we intend to find extreme values of $\bar{P}_{inj}$ (see Eq. 9.4), but due to the probabilistic nature of ML algorithms, our statistic saturates at 1 as we can see in Fig. 9-9. Since we are interested in the region with the highest probability we redefine our statistics as

$$-\log\left(1-\bar{P}_{inj}\right)+\epsilon, \tag{9.5}$$

where $\bar{\cdot}$ represents again the harmonic mean (see Eq. 9.4 for a definition), and $\epsilon = 10^{-20}$ is added to prevent undefined behavior when $\bar{P}_{inj} \approx 1$, ensuring numerical stability. The main advantage of using the harmonic mean in this context is that it places more emphasis on smaller values, which helps to down-rank candidates with lower statistics. To assess the performance of this statistic we compute the ROC curve, but with a different definition of TP and FP. TP are coincident GW injections of O3a present in the zero lag that has been identified by GstLAL, i.e. they produce a track. On the other hand, FP are the coincident tracks of the time shifts of O3. With this definition we can construct the ROC curve stepping on different values of our statistic $-\log\left(1-\bar{P}_{inj}\right)+\epsilon$. Note that better performing ROC would be the ones that maximize the area under the curve (AUC). It is relevant to note that, as we are working with triple time if one detector does not observe the signal, then its contribution $P_{inj,k}^{-1} = 0$ (see Eq. 9.4).

In Fig. 9-10 we present the ROC curve of $-\log\left(1-\bar{P}_{inj}\right)+\epsilon$ (dashed lines) in triple detector time (H1L1V1) for different detector combinations. We can observe that H1L1 and H1V1 have a better performance than L1V1. This implies that the L1V1 combination has more background tracks, possibly due to a larger glitch population in both detectors for different reasons. On the other hand, H1L1V1 has a better performance than double-coincidences since we only have two background tracks. We summarize their AUC in Table 9.3.

TABLE 9.3: Area under the curve (AUC) from ROC curves in Fig. 9-10.

	Time Detector	**H1L1V1**
$-\log\left(1-\bar{P}_{inj}\right)$	*H1L1*	9281.54
	H1V1	9113.69
	L1V1	8843.81
	H1L1V1	9838.81
$\overline{P_{inj}/(\xi^2/SNR^2)}$	*H1L1*	9752.80
	H1V1	9716.73
	L1V1	9712.19
	H1L1V1	9877.87

As we have seen for L1V1, it is possible that the behaviour of MLP is overly optimistic, caused by a combination of its training in a completely controlled population-GW simulations and well-known glitches-and an overly simplified track input. Because of this, we explored a combination of P_{inj} with variables used in traditional GW searches, such as SNR and ξ^2. The best-performing statistic was

$$\overline{P_{inj}/(\xi^2/\mathrm{SNR}^2)}, \tag{9.6}$$

where again $\bar{\cdot}$ represents the harmonic mean. We present in Fig. 9-10 (solid lines) its ROC curve in triple detector time (H1L1V1). We can observe that all double coincidences have improved greatly with respect to the statistic $-\log\left(1-\bar{P}_{inj}\right)+\epsilon$. Moreover, in triple coincidence, the AUC is 9877.87, while before it was 9838.81 (see Table 9.3).

In GW searches, a common magnitude to assess the performance is to use the false alarm rate (FAR) measured in $\text{yr}^{-}1$(see Eq. 3.42), constructed by employing the time shifts. In particular, it is custom to represent the inverse FAR (iFAR) as a function of the ranking statistic, $\overline{P_{inj}/(\xi^2/\mathrm{SNR}^2)}$, which we show in Fig. 9-11. This plot relates the ranking statistic with the iFAR significance, i.e. the number of false alarms per the number of analysis times measured in years. As an example, if we have $\overline{P_{inj}/(\xi^2/\mathrm{SNR}^2)} \approx 150$ for H1L1 coincidence, then we will have an iFAR ≈ 0.01, or equivalently, one false alarm per 100 years. Note that H1L1V1 is in the bottom left corner as it has two samples.

A principal characteristic of a GW ranking statistic is that it increases monotonically as the

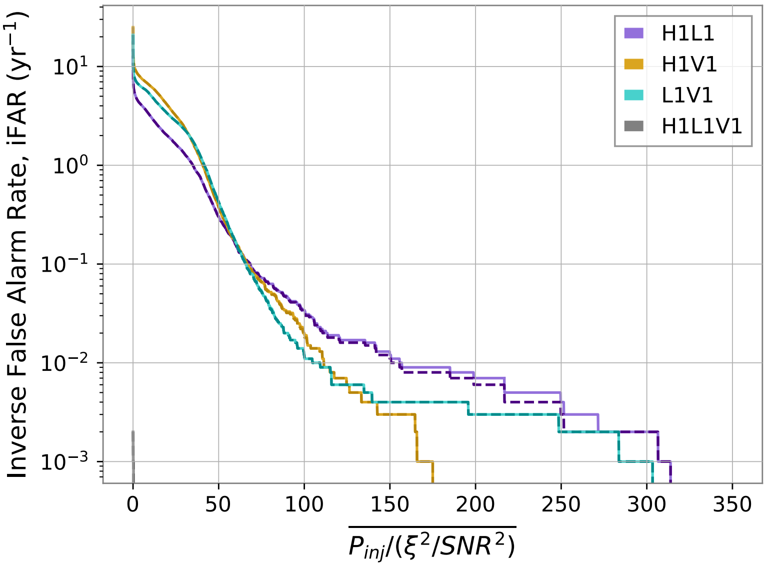
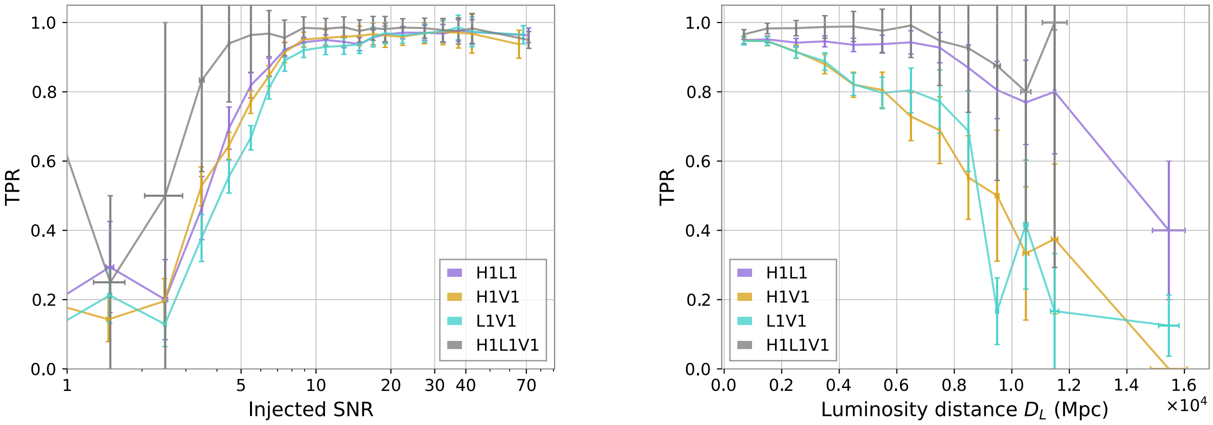


Figure 9-11: Inverse false alarm rate (iFAR) as a function of the ranking statistic in triple time (H1L1V1) for different detector combinations. The solid lines show the time-shifted background with real GW signal tracks, while dashed lines show the time-shifted background without them.

probability of finding a GW signal increases. Because of this, real GW signal tracks would lay on the far right of the iFAR distribution, increasing the false alarm. To investigate whether our ranking statistic behaves in this way we remove all tracks that correspond to real GW signals, yielding the dashed line distributions. While there is a noticeable discrepancy in the H1L1 distribution, this issue does not occur with H1V1 and L1V1. This may be due to the lower sensitivity of V1 or a limitation of our statistics.



(A) TPR as a function of injected SNR.

(B) TPR as a function of luminosity distance.

Figure 9-12: TPR, i.e., the rate of correctly classified GW injections, at iFAR = 0.1 (equivalent to one per 10 years) for different detector combinations in triple time (H1L1V1). Error bars represent the standard error at 3 standard deviations.

To further understand the performance of our statistic, we measure the recovery of GW injections from O3a. To compute the TPR, we fix iFAR at 0.1, or equivalently, one per 10 years, using the results in Fig. 9-11. In Fig. 9-12 we present TPR as a function of the injected SNR (Fig. 9-12a) and as a function of the luminosity distance D_L in Mpc (Fig. 9-12b).

In Fig. 9-12a we can observe that for all detector combinations TPR ≈ 0.95 for injected SNR $\gtrsim 8$. For SNR $\lesssim 8$ there is a drop in performance for double coincidence tracks, while triple coincident tracks maintain their TPR until ≈ 5 injected SNR. However, we must note that for an injected SNR ≈ 5 there is less statistic, as we can observe from the increase in size of the vertical error bars. In Fig. 9-12b we can observe that the TPR for H1V1 and L1V1 decreases steeply for $D_l \gtrsim 0.6\,\mathrm{Mpc}$, possibly due to the lower sensitivity of V1. Moreover, H1L1 and H1L1V1 have a similar trend, with a drop in performance at $D_l \gtrsim 1.2\,\mathrm{Mpc}$. As in the case of the injected SNR, there is less statistic for $D_l \gtrsim 1\,\mathrm{Mpc}$. In future work, we could conduct a broader injection campaign to mitigate these issues.

9.5 Discussion

In the previous Sections, we showcased the utilization of matched-filtering triggers to learn the patterns, or tracks, of different types of signals from single-GW detectors. It is relevant to highlight the reliance of the ML model on a mere 6 parameters (see Eq. 9.2) to perform the multi-classification task. This approach does not only recover successfully the test set of the population we have trained on but also accurately recovers injections of GW signals. In future works, it would be interesting to compare these results with state-of-the-art algorithms to better quantify the performance of our method.

Despite these achievements, certain limitations must be noted. The primary constraint is that we characterize the feature vector of our input with only 6 features. Before this investigation, we have not utilized a feature selection procedure for the definition of the input feature vector, as we have limited ourselves to variables that are well-understood in the GW community. Such an approach could enhance the performance of the model.

Another limitation pertains to the ranking statistic itself. In this work, we have combined the statistic of the ML model with standard measurements such as SNR and ξ^2. Due to time constraints, this exploration was limited, but it does not imply that our ranking statistic is the optimal one. In future works we will further explore enhancing our statistic with different variable combinations. However, we expect that the largest improvement would come from the inclusion of the time dimension, i.e. the behaviour of the signal through the template bank. This will most likely provide valuable insight useful to enhance the distinction between these classes.

9.6 Conclusion

In this investigation, we propose a flexible method to detect CBC signals combining the robustness of matched-filtering as an optimal filter, with the generalization power of ML algorithms. Thus, we construct an MLP model to learn from sets of the matched filtering triggers, labelled here as tracks, and perform a multi-classification task in a single detector. Specifically, we tackle the IMBH search of GstLAL during O3, but this method could be extended to other CBC signals and matched-filtering algorithms.

Since multiple templates could potentially match a given signal, meaning that the input to the model would have variable length, in this proof-of-concept work, we have reduced the dimensionality of the problem by averaging the matching templates and weighting them by SNR. Another difficulty of our task is to deal with highly unbalanced data, so to mitigate this problem, we undersample large classes and oversample small classes. Nonetheless, this could bias the model towards certain repetitive features, so to avoid overfitting we employ 9-fold-cross-validation and early-stopping algorithm. In the following, we revisit some of the major results of this method:

- *Controlled data set:* We trained our model in O3a data, and we tested its generalization ability in O3b. We used the standard ML statistics to test its robustness. We obtained accuracies of $98.8\%, 95.8\%$ and 99.3% for H1, L1 and V1 in O3a test set, while showing a sharp decrease in accuracy of $73.0\%, 67.5\%$ and 75.0% for H1, L1 and V1 in O3b test set (see Section 9.4.3). This drop in performance was caused by the influence of glitches, so we decided to implement time coincidence to lower this background.

- *Novel time coincidence:* To enhance the performance of the model, we enforce time coincidence among detectors, which greatly reduces the background of glitches. As we are dealing with clusters of triggers, instead of single triggers, we take the average time of the cluster and the light time travel between detectors to consider that tracks from different detectors coincide in time (see Section 9.3.4).

- *Computational efficiency:* This method only uses 6 variables to perform this classification (see Eq. 9.2), and while its training is intensive, we can classify a given input in 2.9×10^{-6} s. As this process is highly parallelizable, this computation essentially is inexpensive.

- *Construction of accidental background*: We constructed the time shift background to assess the significance of our ranking statistic (see Section 9.4.4) and tested its performance on simulated IMBH signals. With an $\text{iFAR} = 0.1\,\text{yr}$, we have a $\text{TPR} \gtrsim 0.8$ at $\text{SNR} \sim 5$ for all detector combination. Regarding the luminosity distance, H1L1 and H1L1V1 have a $\text{TPR} \gtrsim 0.8$, with a drop in performance at $D_L \gtrsim 1.2\,\text{Mpc}$.

In summary, this method has shown not only to have a robust performance in a controlled environment classifying IMBH injections but also shown its generalization power in finding simulated GW signals, demonstrating that it is possible to form a synergistic relationship between current state-of-the-art matched-filtering techniques and novel ML methods.

In future work, we will explore a different ML method that can process varying length inputs, as the time information might be relevant to enhance the performance of the model. For this aim, several ML algorithms could be employed, such as recurrent neural network [361] or even transformers with an attention mechanism [362]. With such a model we could perform a fair comparison with state-of-the-art pipelines to quantify the potential improvements in IMBH searches that this technology may offer. Furthermore, and as we mentioned before since this methodology is flexible and simply relies on matched filtering computation, we could extend it to other CBC signals and state-of-the-art matched-filtering algorithms.

Chapter 10

Conclusions and prospects

In 2015, the LIGO-Virgo collaboration achieved a groundbreaking milestone by detecting the first GW signal from a BBH event, namely GW150914. This monumental discovery not only confirmed the merger of two black holes but also established the field of GW astronomy. However, this achievement was the culmination of a challenging journey, requiring the collective efforts of multiple generations of scientists who meticulously designed theoretical (see Chapter 1) and experimental frameworks (see Chapter 2) essential for this detection.

Towards the next generation of gravitational wave detection

As of the completion of this thesis, over 90 transient astronomical events have been confidently detected during the past three observation runs by LIGO-Virgo collaboration and other research groups. Beyond the first detection, some of the most noteworthy GW occurrences include GW170817, which marked the pioneering multi-messenger observation involving both gravitational waves and the electromagnetic spectrum; GW190521, that challenged the mass gap due to pair-instability supernovae; or GW190814, an event characterized by significantly asymmetric, possibly indicative of a NSBH merger.

The ability to make such observations has not solely relied on the construction of GW interferometers (see Chapter 2) but the design of robust and sophisticated detection algorithms, characterized by their exquisite accuracy (see Chapter 3), and the mitigation of background noise, known as glitches (see Chapter 2). However, with the upgrade of second-generation interferometers, and the construction of third-generation interferometers, the detection rate will grow steeply, and GW signals will spend from hours to even days in their sensitivity band. This evolving scenario will introduce novel challenges to current state-of-the-art detection algorithms, as the increased detection rate will not only allow us to probe our current knowledge but will also provide an opportunity to find unexpected physical phenomena.

In recent times, ML algorithms have sparked the interest of scientists due to their success in solving various tasks in different domains, demonstrating their versatility and efficacy. The GW field is no exception, with numerous authors exploring their potential (see Chapter 4). In the context of an exploding field of both GW discoveries and ML applications, this thesis advocates for using ML to complement state-of-the-art GW searches and even enhance their performance, fostering a synergistic relationship. Hence, we establish a symbiotic relationship, focusing on enhancing both modelled and unmodelled GW searches, particularly targeting challenging signals like IMBH and CCSN, as well as modelling the noise background that poses challenges to these searches, namely *Blip* glitches.

Simulating transient noise burst

In Chapter 6, we have discussed the generation of background noise that hinders transient GW searches, with a specific focus on disruptive *Blip* glitches. These glitches impede the detection of IMBH and CCSN, as they have a short duration, a high rate of occurrence, and are in the frequency band of interest. In the present work, we showcased the production of time series of *Blip* glitches, adapting a ML algorithm for images to time series. We introduced several statistical metrics to assess the performance of the algorithm and the quality of the generations. Additionally, we have presented an open-source user-friendly interface for generating glitches, known as `gengli`, providing a wide variety of applications to enhance the performance of transient search algorithms.

Nonetheless, as highlighted in Chapter 6, one of the main limitations is the lack of ground truth in glitch populations. This issue is tackled in the next chapter as a proof-of-concept investigation. Another one of the main limitations of our work is the focus on a single class of glitches, due to the resource-intensive nature of glitch extraction for its posterior modelling. Several authors tackled this problem using less flexible approaches such as image generation, but these studies often neglect proper mitigation of background noise.

Consequently, we advocate for increased efforts within the GW community to develop flexible glitch modelling techniques, as this will greatly impact transient GW searches, enhancing the confidence of GW candidates.

Detection of anomalous transient noise bursts

In Chapter 7, we explored a selected subset of safe auxiliary channels from LIGO Livingston, employing FD-encoding to investigate its use in anomaly detection. The main goals of this study were twofold: to evaluate the effectiveness of this data representation for characterizing glitches, and to create a data-driven model that clusters glitches in an unsupervised fashion. For this aim, we constructed a model that directly inputs encoded data from the auxiliary channels of the detector to identify anomalies that stand out from the overall data pattern: misclassifications, overlaps, as well as novel morphologies.

Despite the success of this proof-of-concept investigation, several challenges remain to be addressed. In future work, we plan to extend this method to understand better the space of LIGO-Livingston glitches by increasing the number of glitch classes and auxiliary channels analyzed. Additionally, we will introduce an anomaly score to assess the significance of outliers and explore data fusion between FD-encoding and the time-frequency representation of the detector's main strain, $h(t)$. Furthermore, we aim to investigate the correlations identified by the model among auxiliary channels and explore their potential significance in the context of explainable ML.

Enhancing unmodelled core-collapse supernova searches

In Chapter 8, we discussed the CCSN phenomenon and emphasized the significance of detecting its GW to unveil the inner workings of the collapse. While CCSN are among the most energetic processes in the universe, their GW signals are expected to be extremely faint. To address this challenge, we developed a ML learning model to learn the peculiarities of the CCSN using the time-frequency representation of the current state-of-the-art unmodelled algorithm cWB. Given the computational intensity and complexity of generating CCSN waveforms, we trained the ML model on a set of phenomenological waveforms that capture key CCSN GW signal features. This not only demonstrates the model's ability to learn phenomenological waveforms but also tests its generalization power to a catalogue of CCSN GW waveforms from Numerical Relativity.

However, as highlighted in Chapter 8, to improve our work it is fundamental to refine the phenomenological waveforms to ensure a closer resemblance to the CCSN catalogue. The phenomenological set will play a crucial role in shaping the learning process of the ML model. Additionally, an important aspect is extending this methodology to O3 and comparing its performance with cWB algorithm, incorporating a proper estimation of the background noise. Nevertheless, with further development, this could be a stand-alone search algorithm tailored for targeted searches of CCSN GW signals.

Enhancing modelled intermediate-mass binary black hole searches

In Chapter 9, we discussed the importance of detecting a population of IMBH to unravel the formation of compact objects within the mass gap resulting from pair-instability supernovae. Despite falling in the sensitivity band of current ground-based detectors, IMBH are challenging to detect due to their short duration and similarity to the background noise, known as glitches.

As matched-filtering methods do not directly include information about the background and the target signal, and under the assumption that triggers generated by these methods are inherently different depending on the signal's origin to analyze, we leverage this information through a ML model. In this way, we use single-detector triggers from GstLAL to train a ML model in real O3 detector noise and simulated IMBH. Afterwards, we employ time coincidence between detectors to lower the noise background. Moreover, we construct a background of time shifts to assess its performance with a set of simulated IMBH GW signals. In future works, it would be interesting to compare these results with the current state-of-the-art search algorithms to assess the performance of our method.

Nonetheless, as highlighted in Chapter 9, the principal limitation of this proof-of-concept work is the disregard of the time dimension, which could provide valuable information to the model, since constructing an ML algorithm with varying length is a challenging task. Therefore, we advocate for continued research efforts in this direction, and the addition of these algorithms to current state-of-the-art matched filtering methods.

The era of Machine Learning

Through this thesis we have described several ML applications in GW astronomy, showing the future of GW data analysis is intertwined with the speed and generalization power of ML algorithms. While still in the early stages, I strongly believe that ML methods will empower exploration of the densest and most energetic regions of the Universe, enabling astronomical discoveries beyond existing theoretical frameworks.

To finalize, I would like to share a quote, as it reflects not only the implications of ML in the field of GW but also its broader impact on society.

> "*We live in a society exquisitely dependent on ~~science and technology~~ machine learning, in which hardly anyone knows anything about ~~science and technology~~ machine learning.*"
> — Carl Sagan (1990), modified by Melissa Lopez (2024)

Chapter A

Additional Resources

A.1 Dominant sources of noise

A.1.1 Thermal noise

Being one of the fundamental and dominant noise sources in the interferometer, as observed in Fig. 2-7 (blue), thermal noise is associated with sources of energy dissipation, coming in at least two forms: due to the Brownian motion of the mirrors, associated with the losses in the mirrors' material, and due to the suspension of the mirrors, related to losses in the wires' material [363].

To understand the motion of the mirrors, we can describe a mass m moving as a harmonic oscillator using the Langevin equation which describes Brownian motion:

$$F_{th} = m\frac{dx^2}{dt^2} + \phi\frac{dx}{dt} + kx \tag{A.1}$$

where F_{th} is the fluctuating force with white spectral density $F(\omega) = 4k_B T\phi$, ϕ is its damping coefficient, and $-kx$ is a restoring force [364]. Thus, solving this equation in the frequency domain yields the power spectral density of the position due to the Brownian motion,

$$x^2(\omega) = \frac{4k_B T\phi}{(k - m\omega^2)^2 + f^2\omega^2} \tag{A.2}$$

where k_B is the Boltzmann's constant, T the temperature of the mass, ω the angular frequency and ϕ the loss angle of the oscillator. Assuming a linear system in thermodynamic equilibrium, the fluctuations due to dissipation increase the system's noise, so that we can use the fluctuation-dissipation theorem. We can cast the thermal driving force as

$$F_{th}^2(\omega) = 4k_B T R(\omega), \tag{A.3}$$

where $R(\omega)$ is the mechanical resistance where the dissipation happens, which comes from the real part of the impedance $Z(\omega)$. The fluctuation-dissipation theorem states that the power spectrum of the force responsible for thermal fluctuations is related to the real part of the impedance $Z(\omega)$ such that

$$x^2(\omega) = \frac{4k_B T\sigma(\omega)}{\omega^2} \tag{A.4}$$

where $\sigma(\omega)$ is the mechanical conductance, the real part of the admittance $Y(\omega) \equiv Z^{-1}(\omega)$. For a simple oscillator, as described above, the impedance and admittance are

$$Z = f + i\omega + \frac{k}{i\omega}, \qquad Y = \frac{\omega^2 f + i(\omega k - m\omega^3)}{(k - m\omega^2)^2 + \omega^2 f^2} \tag{A.5}$$

By substituting the admittance in Eq. A.4 we retrieve the Langevin equation. Hence, the fluctuation-dissipation theorem can be regarded as a generalization of the Langevin equation.

A.1.2 Quantum noise

The detector's sensitivity is limited as well by quantum noise, that comes from the quantum nature of photons due to the Heisenberg uncertainty principle and quantum fluctuations [365]. The limiting quantum noise sources of the GW detectors are, on one side, the *photon shot* noise, h_{shot}, that arises due to statistical fluctuations in the number of photons at the output dominating at high frequencies; and on the other side, the *quantum radiation pressure* noise, h_{rad}, that arises due to to the fluctuations in the mirrors' positions induced by quantum radiation pressure fluctuations dominating at low frequencies. Thus total quantum noise of a Michelson interferometer in the frequency domain is:

$$\tilde{h}_{tot}(f) = \sqrt{\tilde{h}^2_{shot}(f) + \tilde{h}^2_{rad}(f)}. \tag{A.6}$$

In the following, we describe in detail the effects of the shot noise and the quantum radiation pressure on the sensitivity of the interferometer. We also recommend the reader to refer to [366] for an in-depth description of these sources.

Shot noise

In a Michelson interferometer, the average photon number per unit time $\langle n \rangle$ at the output and its standard deviation Δn in a τ time interval, are respectively

$$\langle n \rangle = \frac{\lambda}{2\pi} P_{out}, \quad \Delta n = \sqrt{\frac{\lambda P_{out} \tau}{2\pi}}, \tag{A.7}$$

where λ is the laser wavelength and P_{out} is the power of the interferometer output. We remind the reader that in this notation the speed of light c and Planck's constant $\hbar$ are taken as fundamental units. We can set the optical power fluctuations as equivalent to position fluctuations, such that the standard deviation in the displacement L, due to fluctuations in the average photon number detected, for a time interval τ, is

$$\Delta L_{shot} = \frac{\Delta n / \langle n \rangle}{\frac{1}{P_{out}} \frac{dP_{out}}{dL}} = \sqrt{\frac{\lambda}{4\pi P_{in} \tau}}, \tag{A.8}$$

where P_{in} is the input power. The strain noise caused by the photon shot noise in the frequency domain then yields,

$$\tilde{h}_{shot}(f) = \frac{1}{L} \sqrt{\frac{\lambda}{2\pi P_{in}}}. \tag{A.9}$$

Therefore, increasing the input power of the laser, P_{in}, can improve the strain associated with the shot noise, enhancing the detector sensitivity at all frequencies. Nonetheless, increasing P_{in} has its limitations from a practical point of view, as it affects h_{rad}.

Quantum radiation pressure noise

A beam of photons that strikes a mirror is reflected wielding pressure on the mirror itself. If this radiation pressure were constant, it could be compensated by a mechanism, but as the number of photons arriving on the mirror fluctuates, so does the radiation pressure. This uncertainty translates into uncertainty in the position of mirrors, which limits our ability to precisely measure the change in the length of the arms due to a passing GW.

To calculate the sensitivity due to the radiation pressure, we consider a laser beam with power P_{in} that impinges perpendicularly on a mirror. Then, the force exerted is $F_{rad} = P_{in}$, and the fluctuation of shot noise is $\Delta P_{in} = \omega\sqrt{\langle n \rangle}$, such that the fluctuations of the force in the frequency domain yields,

$$\Delta\tilde{F}_{rad}(f) = \sqrt{\frac{2\pi P_{in}}{\lambda}}. \tag{A.10}$$

Thus, the fluctuation of a mirror's position of mass m caused by the fluctuations of this force is,

$$\Delta\tilde{L}_{rad}(f) = \frac{1}{mf^2}\Delta\tilde{F}_{rad}(f) = \frac{1}{mf^2}\sqrt{\frac{2\pi P_{in}}{\lambda}}. \tag{A.11}$$

As the power fluctuations in the two arms of the detector are anti-correlated, the effect on the output is doubled. Thus, the effect of quantum radiation pressure in the strain is,

$$\tilde{h}_{rad}(f) = \frac{2\Delta\tilde{L}_{rad}(f)}{L} = \frac{1}{mf^2L}\sqrt{\frac{8\pi P_{in}}{\lambda}}. \tag{A.12}$$

Quantum-enhancement: squeezed states

From Eq. A.9 and Eq. A.12 we can observe that while h_{shot} is independent of frequency, h_{rad} scales as f^{-2}. Moreover, while increasing P_{in} will lower h_{shot}, this will come at the cost of increasing h_{rad}. Shot- and radiation pressure noise are also known as phase- and amplitude-noise, as they are related to the variance in phase and amplitude of the light, respectively.

A vacuum state of light, where the coherent amplitude is zero, has equal uncertainty in both amplitude and phase quadratures, bounded to Heisenberg's principle. To reduce the uncertainty in one variable with respect to the other we can "squeeze" the state such that one variable has less uncertainty than the other, as long as the Heisenberg inequality is satisfied. Thus, adding phase-squeezed vacuum states reduces the shot noise below the quantum limit [367, 368].

A.1.3 Seismic noise

Human, atmospheric and geological activity produce seismic noise, which is a type of environmental noise dominant at low frequencies. Seismic noise can also couple with the optical system of the detector producing an excess of noise in the main strain of the detector $h(t)$ at high frequencies [369]. The first design choice to counteract seismic noise is the location of the detectors (see Fig. 2-5).Moreover, to isolate the test masses from seismic perturbations, the detectors are equipped with sophisticated active and passive isolation systems.

- *Active systems:* They are composed of sensors that detect seismic fluctuation and apply a force to counterbalance the perturbation in a wide range of frequencies.

- *Passive systems:* They use mechanical apparatus to dissipate seismic perturbances at low frequencies. An example is the pendulum suspension of the mirrors that absorb perturbances at frequencies $f > 1\,\mathrm{Hz}$. Each stage of the suspension reduces the transfer of seismic motion by a factor of $1/f^2$.

The interested reader can refer to [370] and [371] for an in-depth description of active and passive systems in LIGO and Virgo interferometers respectively.

A.1.4 Gravity gradient noise

Another source of noise that impacts at low frequencies ($f \lesssim 20\,\mathrm{Hz}$) is the gravity gradient noise, also known as Newtonian noise. The gravitational field of Earth is non-uniform due to the fluctuations in its mass distribution. Indeed, gravity gradient noise arises in GW detectors because of the fluctuating density of the Earth beneath and near each of the interferometer's test masses. As we can observe in A-1, as a surface wave propagates in the vicinity of the test masses, it tilts them causing a disturbance [372]. Mathematically, we can describe the motion of the test masses caused by gradient gravity noise as

$$\tilde{x} = \frac{\rho\beta(f)\tilde{W}(f)}{2\pi f^2} \text{ at } f \gtrsim 3\,\mathrm{Hz}\ , \qquad \text{where } \tilde{W}(f) = \sqrt{\frac{2\tilde{X}^2(f) + \tilde{Z}^2(f)}{3}} \tag{A.13}$$

where ρ is the mass density in the vicinity of the test masses, and $\beta(f)$ a dimensionless correction known as the reduced transfer function. $\tilde{W}$ is the displacement rms-averaged over 3-dimensional directions: in the XY-plane the spectral density of the horizontal displacement is represented by $\tilde{X}$, where the seismic motions are considered to be horizontally isotropic, and the spectral density of the vertical displacement is $\tilde{Z}$ [373].

To reduce gravity gradient noise it has been proposed to measure the relevant ground motion expected to couple through local gravity and subtract it. Another approach for the reduction of this noise is to build the interferometers underground, as in the case of KAGRA, or even operate in the space, far from these density fluctuations, like LISA.

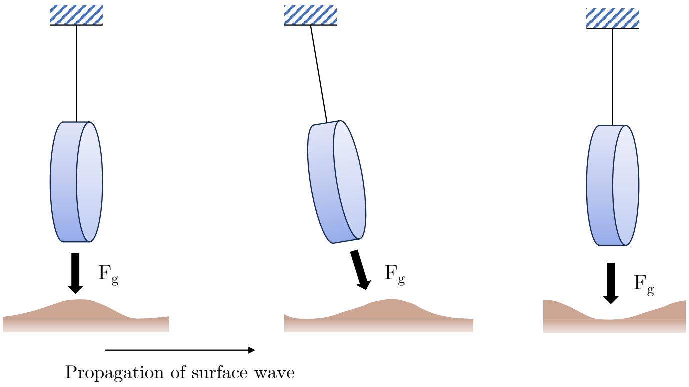


Figure A-1: Schematic illustration of a fluctuating gravitational force F_g on a suspended test mass, caused by the propagation of a surface wave through the ground.

A.2 Wilson-Daubechies-Meyer formalism

In this section, we provide the formalism of WDM discussed in section 3.3.3. Consider a discrete data $x[k]$ sampled at f_s with a maximum angular frequency $\Omega = \pi f_s$ and window function ϕ. We define WDM filters of the time-frequency representation of $x[k]$ as,

$$w_{nm} = \sum_k f_{nm}[k]x[k] \quad \text{where } n, m \in \mathbb{N} \tag{A.14}$$

The WDM basis functions in the Fourier domain are,

$$\tilde{g}_{n0}(\omega) = e^{-in\omega T}\tilde{\phi}(\omega), \qquad \tilde{g}_{nm}(\omega) = \frac{1}{\sqrt{2}}e^{-in\omega T}\tilde{\psi}_{nm}(\omega) \ \text{ for } m \leq M \tag{A.15}$$

$$\text{where } \ \tilde{\psi}_{nm}(\omega) = C^*_{m+n}\tilde{\phi}(\omega + m\Delta\Omega) + C_{m+n}\tilde{\phi}(\omega - m\Delta\Omega) \ \text{ and } \ M = \frac{\Omega}{\Delta\Omega} = \frac{T}{2\tau} \tag{A.16}$$

where $m > 0, C_{2k} = 1, C_{2k+1} = i, \Delta\Omega = 2\pi/T$ and $\tau = 1/f_s$. The generalized Meyer scaling function $\tilde{\phi}(\omega)$ in the Fourier domain is defined as

$$\tilde{\phi}(\omega) = \begin{cases} \dfrac{1}{\sqrt{\Delta\Omega}}, & |\omega| < A \\ \dfrac{1}{\sqrt{\Delta\Omega}} \cos\left[\dfrac{\pi\nu_n}{2}\left(\dfrac{|\omega| - A}{B}\right)\right], & A \leq |\omega| < A + B \end{cases} \tag{A.17}$$

$$\text{where } 2A + B = \Delta\Omega, \quad \nu_n(x) = \frac{B(x; n, n)}{B(1; n, n)}, \quad B(x; a, b) = \int_0^x t^{a-1}(1-t)^{b-1}dt,$$

being $B(x; a, b)$ the incomplete Beta function and n a parameter to control the sharpness of the edges of the scaling function $\tilde{\phi}(\omega)$. Under the condition that $M \in \mathbb{N}$ so that filters $f_{nm}[k]$ are discrete translations of $f_{0m}[k]$, we can redefine $\tilde{g}_{nM}$ to be

$$\tilde{g}_{nM} = e^{-i(2n+q)M\omega\tau}\Big[\tilde{\phi}(\omega + \Omega) + \tilde{\phi}(\omega - \Omega)\Big], \text{ for } |\omega| \leq \Omega, \tag{A.18}$$

for $q = 0$ if M is even and $q = 1$ otherwise. Thus, the new set $\tilde{g}_{nm}$ forms an orthonormal basis on $[-\Omega, \Omega]$, and its explicit WDM expansion is

$$\begin{aligned} w_{n0} &= \tau \sum_{k\in\mathbb{Z}} x[2nM + k]\phi[k] \\ w_{nm} &= \tau\sqrt{2}\text{Re}\Big[C_{m+n}\sum_{k\in\mathbb{Z}} e^{i\pi km/M}x[nM + k]\phi[k]\Big], \quad \text{for } 0 \leq m \geq M \\ w_{nM} &= \tau \sum_{k\in\mathbb{Z}} (-1)^k x[2nM + qM + k]\phi[k] \end{aligned} \tag{A.19}$$

being $\phi[k]$ the sampled values of $\phi(t)$. Employing the periodicity of the exponential factor,

$$\sum_{j=0}^{j<2M} e^{i2\pi jm/2M}X_n[j] = \sum_{k\in\mathbb{Z}} e^{i\pi km/M}x[nM + k]\phi_k \tag{A.20}$$

$$X_n[j] = \sum_{k\in\mathbb{Z}} xnM + 2kM + j\phi[2kM + j]. \tag{A.21}$$

The summation on the left side of Eq. A.20 corresponds to the discrete transform of $X_n[j]$, which has length $2M$. Thus, we can use the fast Fourier transform for a speedy calculation of $X_n[j]$ and its transform. Note that only half of the Fourier components in Eq. A.19 represent the WDM expansion, so the other half is given by the WDM expansion

$$\begin{aligned}\hat{w}_{n0} &= \tau \sum_{k\in\mathbb{Z}} x[(2n+1))M+k]\phi[k] \\ \hat{w}_{nm} &= \tau\sqrt{2}\mathrm{Im}\Big[C_{m+n}\sum_{k\in\mathbb{Z}} e^{i\pi km/M} x[nM+k]\phi[k]\Big], \quad \text{for } 0 \leq m \geq M \\ \hat{w}_{nM} &= \tau\sum_{k\in\mathbb{Z}}(-1)^k x[2nM+(1-q)M+k]\phi[k]\end{aligned} \tag{A.22}$$

in the complementary orthonormal basis constructed from the time-shifted WDM basis functions (see Eq. A.15) by $T/2$, so each WDM set (w_{nm} or $\hat{w}_{nm}$) gives a complete time-frequency representation of the data.

A.3 Welch's method

The most common method for estimating the PSD is Welch's method which divides the time signal into successive blocks, forming the periodogram for each block and averaging [374]. Mathematically, given a random process $x[n]$ we define the periodogram as

$$P_{x_m,M}[w_k] = \frac{1}{M}|\tilde{x}_m|^2 = \frac{1}{M}\Big|\sum_{n=0}^{N-1} x_m[n]e^{-i2\pi nk/N}\Big|, \text{ for } n \in [0,\ldots,M-1], m \in [0,\ldots,K-1] \tag{A.23}$$

where the windowed segment of $x[n]$ divided in M blocks is $x_m[n] = w[n]x[n+mR]$, where $w[n]$ is the window function, R the window hop size and K the number of available segments samples from $x[n]$. Periodograms represent the randomness within their block, so they need to be averaged to obtain a "stable" statistical estimate of the noise spectral envelope. Thus, the Welch estimate of the power spectral density is an average of periodograms across time,

$$S_x^W[\omega_k] = \frac{1}{K}\sum_{m=0}^{K-1} P_{x_m,M}[\omega_k]. \tag{A.24}$$

A.4 `gengli`: results for *Blip* distribution from LIGO Livingston

This appendix presents the results of *Blips* from the L1 distribution, which is compatible with the H1 population. In Fig. A-2, we present a histogram of the classes assigned by `Gravity Spy` to a population of 10^3 artificial *Blips*. As in Section 6.4, we can also observe that the three dominant classes are *Blip*, *Repeating_Blips* and *No_Glitch*, and as we increase the optimal SNR ρ_{opt}, the number of artificial glitches classified as *Blip* increases. As we stated before, `Gravity Spy` classifier seems to be biased towards *Blip* class, since at very low ρ_{opt}, the network will be unable to see the glitches. Another interesting question would be to assess the influence of the detector noise in the classification task of Gravity Spy. Similarly to Fig. 6-12, we present in Fig. A-3 the confidence of `Gravity Spy` as a function of alternative metrics for the dominant classes.

In Fig. A-3, we can also observe that there is no apparent correlation between the measurements and the confidence provided by `Gravity Spy` classifier. To inspect the results, we select certain glitches according to the definitions in 6.4.2. Note that the anomalous glitch found by Wasserstein distance (labelled as D) does not coincide with the one found by the match

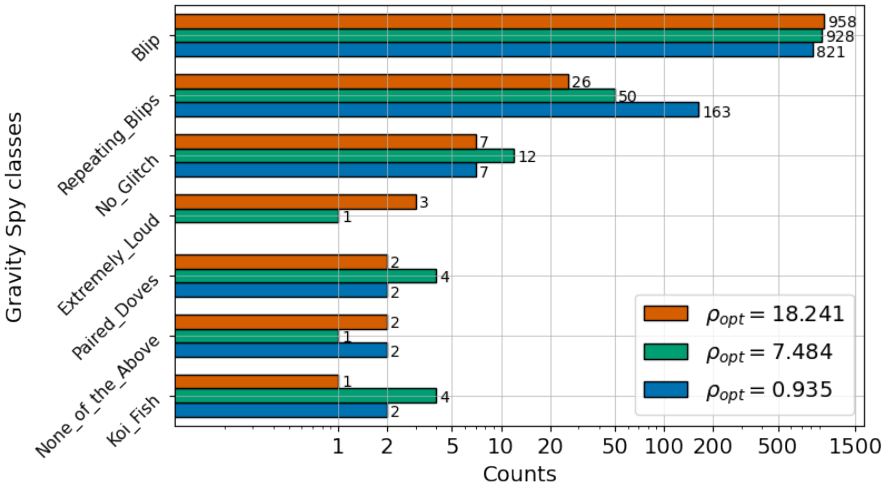
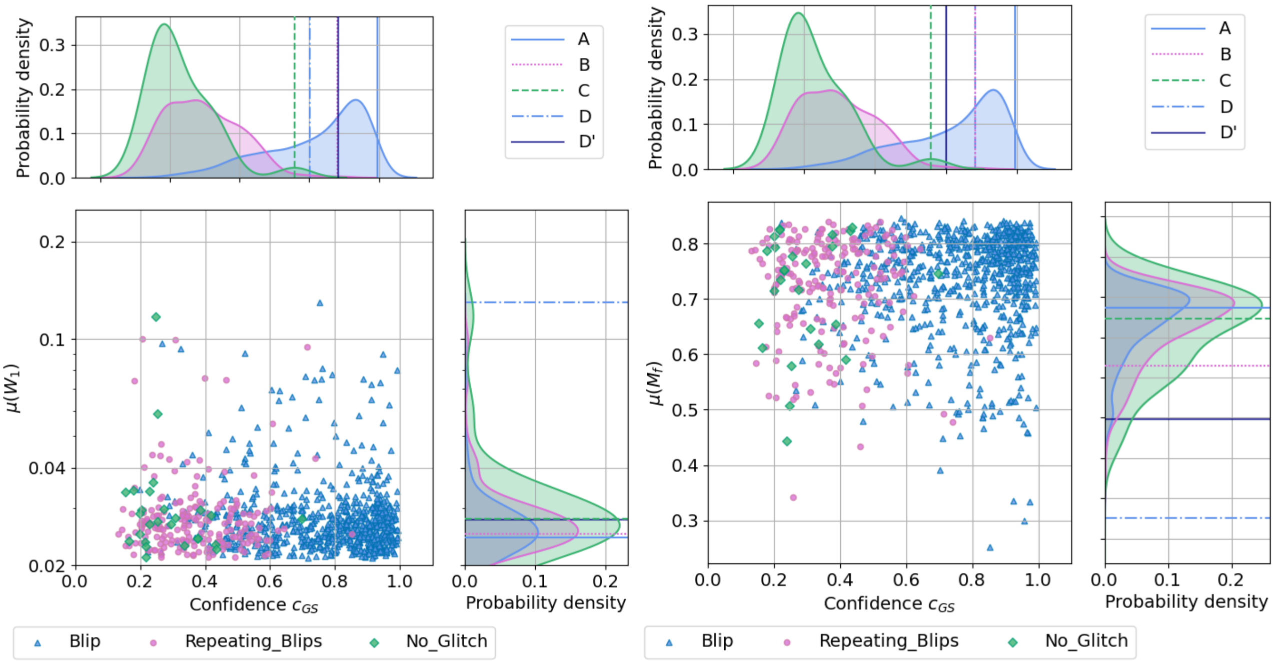


Figure A-2: Histogram of predicted **`Gravity Spy`** *classes for* 10^3 *Blips from L1.*

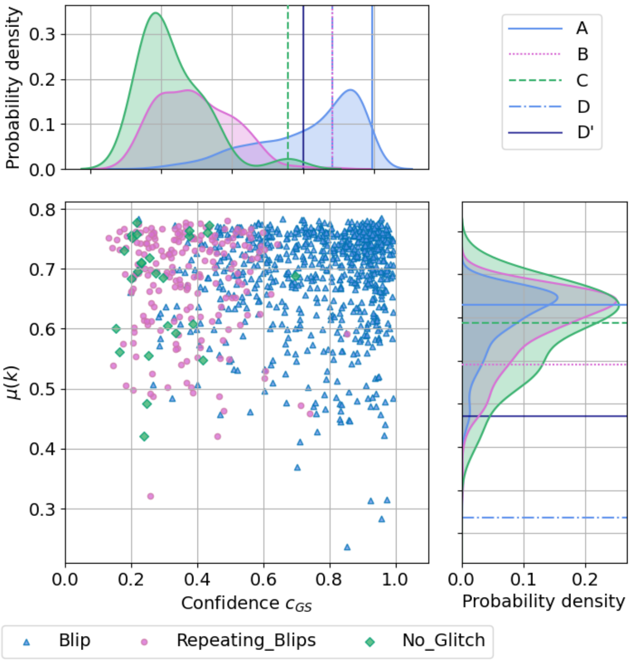
function and normalized cross-covariance (labelled as D’). `Gravity Spy` was able to correctly classify with high confidence glitches A and B, but glitches C, D, and D’ are misclassified.

For visualization and a better understanding of the results, we plot in Fig. A-4 the Q-transforms and the time series injected in real whitened noise of the selected glitches. While glitch A is classified by `Gravity Spy` as a perfect glitch, glitch C is miss-classified as *No_Glitch*, although their Q-transforms look similar. It is interesting to mention that the GAN was able to generate a *Repeating_Blip* because some repeating *Blips* are present in the input data set.

Glitches D and D’, which are misclassified by `Gravity Spy`, are situated in the tail of the distribution of the similarity distances. While glitch D has a shape very different from a standard *Blip*, glitch D’ has a very narrow peak.

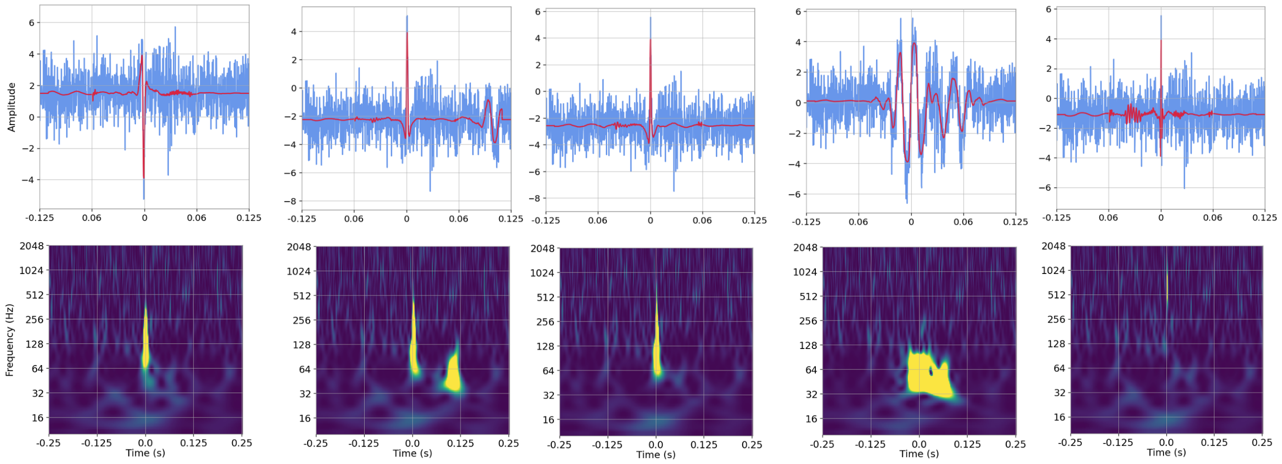


(A) Wasserstein distance (W_1) (B) Match fucntion (M_f)



(C) Normalised cross-covariance (k)

*Figure A-3: Joint and marginal distribution of **`Gravity Spy`** confidence c_{GS} at $\rho_{opt} = 18.46$ against different metrics for different glitch classes for L1: Blip, Repeating_Blips and No_Glitch. We mark in the marginal distributions selected glitches A (solid blue), B (dotted pink), C (dashed green) and D (dash-dotted blue).*

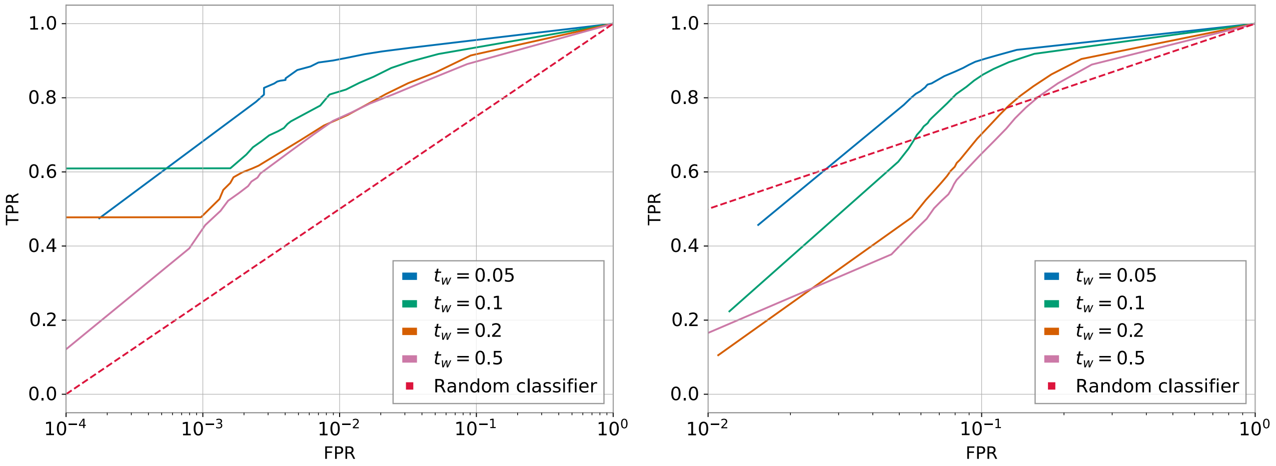


(A) Glitch A (B) Glitch B (C) Glitch C (D) Glitch D (E) Glitch D'

Figure A-4: Time series representation (top row) and Q-scan representation of selected glitches from L1.

A.5 IMBH search: results of LIGO Livingston and Virgo

A.5.1 Selecting a time window

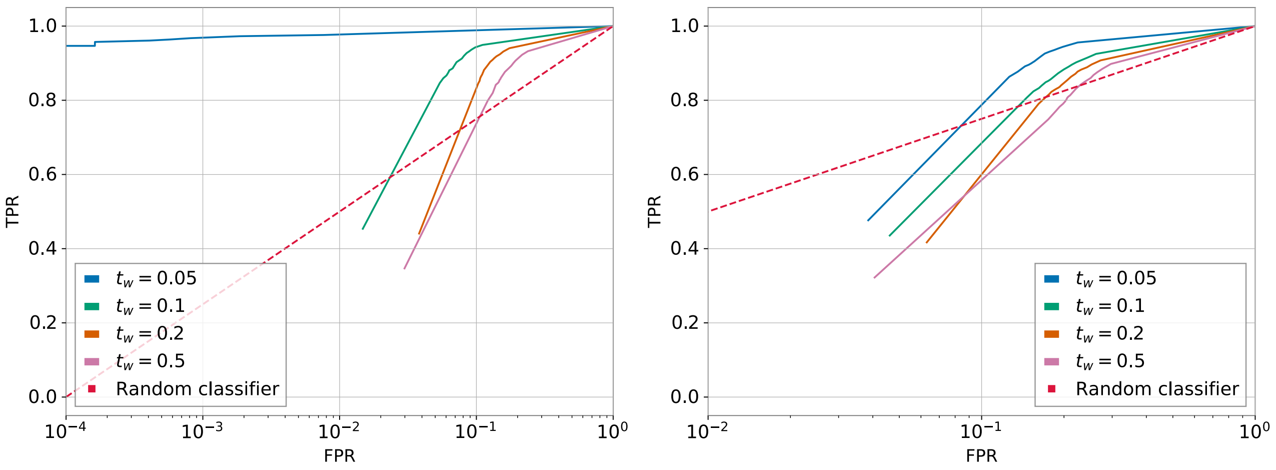


(A) Testing on O3a with *known data set*. (B) Testing on O3b with *known data set*.

Figure A-5: ROC curve for different time windows t_w in L1, i.e. TPR as a function of False Positive Rate FPR in logarithmic scale. (Left) Testing in the known data set of O3a. (Right) Testing in the known data set of O3b. Note that the dashed line indicates a random guess.

Similarly to Section 9.4.1, we show in Fig. A-5 and Fig. A-6 the ROC curves of L1 and V1, respectively. We can observe that the TPR degrades as we increase the size of t_w, being this decrease sharper for the *known data set* of O3b than for the *known data set* of O3a. Notably, for V1 (see Fig. A-6), while the ROC curve of $t_w = 0.05$ in O3a is almost constant, the ROC curve of O3b decreases even faster than in the case of O3b L1.

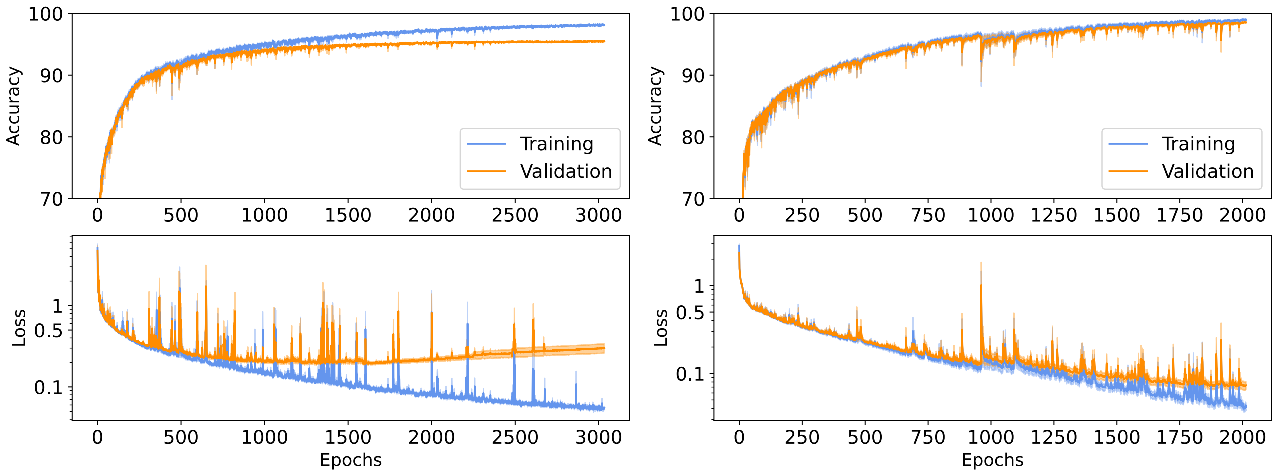
As in Fig. 9-4, in Fig. A-5 and Fig. A-6 $t_w = 0.05\,$s has a better performance than other time windows. A possible explanation for this behaviour could be that since IMBH signals are short, larger time windows would add random triggers that are unrelated to the IMBH signal itself, biasing the model. Hence, as in Section 9.4.1 we conclude that $t_w = 0.05$ is the best time window for our task.



(A) Testing on O3a with *known data set*. (B) Testing on O3b with *known data set*.

FIGURE A-6: ROC curve for different time windows t_w in V1, i.e. TPR as a function of False Positive Rate FPR in logarithmic scale. (Left) Testing in the known data set of O3a. (Right) Testing in the known data set of O3b. Note that the dashed line indicates a random guess.

A.5.2 Training with *known data set* of O3a

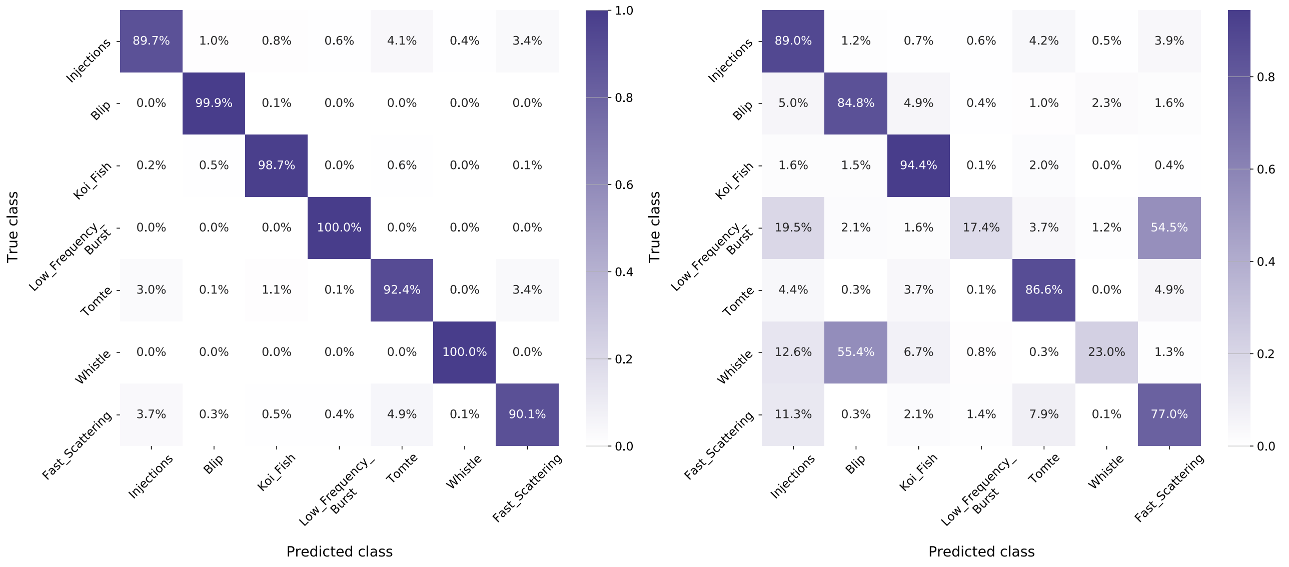


(A) Ligo Livingston. (B) Virgo.

FIGURE A-7: Comparison between training and validating with 9-fold cross-validation during training and testing. (Top row) Mean accuracy at 3 standard deviations as a function of the epochs during. (Bottom row) Average 9-fold cross-validation loss as a function of the epochs.

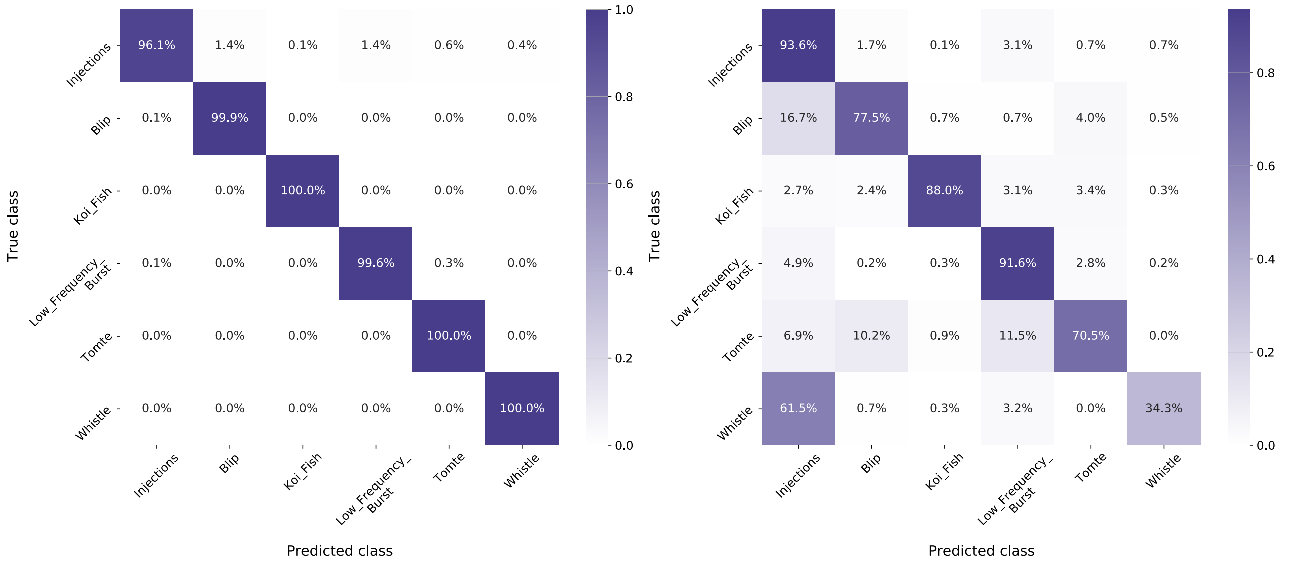
As in Section 9.4.2, in Fig. A-9 we show, for training and validation, the mean accuracy (top row) and loss (bottom row) of the 9-fold cross-validation as a function of the epochs for L1 (left column) and V1 (right column), where the shadowed region represents ±3 standard deviation. Note that the loss is plotted in logarithmic scale, so we can appreciate that there is less overfitting in V1 than in L1.

Another relevant point is the peaks in the loss functions, in particular around epoch 500 in L1 and around epoch 1000 in V1. These peaks indicate a change in the learning rate due to the adaptive learning rate scheme (see Section 9.3.3 for details).



(A) O3a *known data set.* (B) O3b *known data set.*

FIGURE A-8: *Relative values of the confusion matrix for the test set for L1.*



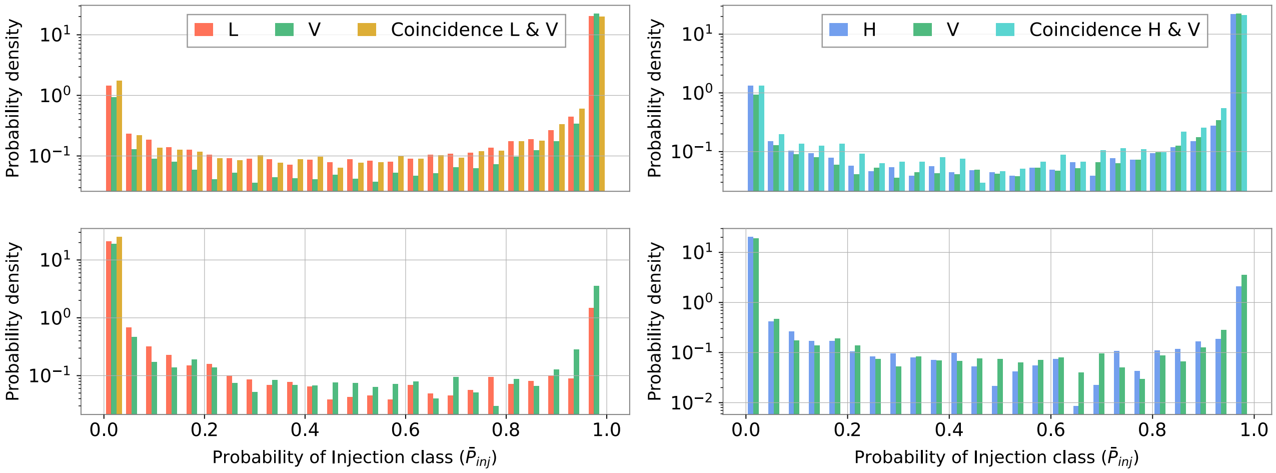
(A) O3a*known data set.* (B) O3b*known data set.*

FIGURE A-9: *Relative values of the confusion matrix for the test set for V1.*

A.5.3 Diving in to the *known data*: performance evaluation

Similarly to Section 9.4.3, where we want to test the generalization power between the *known data sets* of O3a and O3b for H1, we test the *known data sets* of O3a and O3b for L1 and V1, presenting their confusion matrices in Fig. A-8 and Fig. A-9, respectively. In Fig. A-8a and Fig. A-9a we can see that most inputs are correctly classified, yielding an accuracy of 95.81% in L1 and 99.27% in V1. Nonetheless, Fig. A-8b and Fig. A-9b we can observe an increase of false positives, decreasing the accuracy to 67.45% in L1 and 75.91% in V1.

In L1 (see Fig. A-8b), the *Injections* are mostly correctly classified, with 1.2% mislabelled as *Blip*. However, 19.5% of *Low_Frequency_Burst*, 12.6% of *Whistle* and 11.3% of *Fast_Scattering* are incorrectly classified as *Injections*. Interestingly, 54.5% of *Low_Frequency_Burst* are misclassified as *Fast_Scattering* since they share a similar frequency range. In V1 (see Fig. A-9b), the *Injections* are also mostly correctly classified, with 1.7% mislabelled as *Blip*. However, 61.5% of *Whistle* and 16.3% of *Blip* are incorrectly classified as *Injections*. It is also interesting to note that in this detector only 4.9% *Low_Frequency_Burst* are misclassified as *Injections*, while a 10.2% and 11.5% of *Tomte* are incorrectly labelled as *Blip* or *Low_Frequency_Burst*, respectively.



(A) L1 (red) and V1 (green), and their coincidence (yellow).

(B) H1 (blue) and V1 (green), and their coincidence (turquoise).

FIGURE A-10: Probability density of being Injection (P_{inj}), using the known test set of O3b. (Top row) Probability density of elements in the Injection class in logarithmic scale. (Bottom row) Probability density of elements in all the other classes, i.e. glitches, in logarithmic scale. Given the counts of the ith bin c_i and its width b_i, we define the probability density as $c_i/(\sum_i^N c_i \times b_i)$, where N is the total number of bins of the histogram.

Due to the poor generalization between O3a and O3b, we need to reduce the number of false positives. For this aim, we enforce time coincidence as in Section 9.4.3. Similarly to Fig. 9-9, in Fig. A-10 we present the probability of being *Injection* (P_{inj}) for L1 (left column) and V1 (right column). In the top row, we show the probability density of P_{inj} for the *Injection* class, and in the bottom row, we show the probability density of P_{inj} for any other class. When we enforce time coincidence, represented in yellow for L1 and V1, and in turquoise for H1 and V1, P_{inj} increases for the *Injection* class, while we completely discard all the other classes. Note that this is at the cost of reducing the number of correctly classified *Injections*.

Public summary

Gravity: The Invisible Architect of the Cosmos

Gravity is the weakest of the fundamental forces, and we experience it every day here on Earth. It's so familiar that we barely notice its effects, as we can easily overcome it by lifting a single finger. However, gravity behaves very differently on a cosmic scale. The universe is a dynamic and violent place, filled with billions of stars born in chaotic and energetic environments. These stars spend their lives producing energy through nuclear fusion—a process that powers the stars—and often end in massive explosions powerful enough to destroy entire worlds. Meanwhile, here on our small home, Earth, we are far removed from these astronomical cataclysms. We are left to wonder what other worlds are dying and what new ones are being born around those distant, shining stars.

Gravity, often referred to as the invisible architect of the cosmos, plays a vital role in shaping the universe. It dictates the orbits of planets, governs the formation of stars, and binds galaxies together. As the force that holds the vast structures of the universe in place, gravity's influence is both subtle and profound. The nature of gravity has intrigued human minds since ancient times, leading to centuries of scientific research.

In the 17th century, Sir Isaac Newton revolutionized our understanding with his law of universal gravitation, which described how gravity affects objects. However, while Newton's work explained how gravity operates, the true nature of this force remained a mystery. It wasn't until the 20th century that Albert Einstein introduced the theory of general relativity, providing a unified description of gravity and paving the way for modern physics.

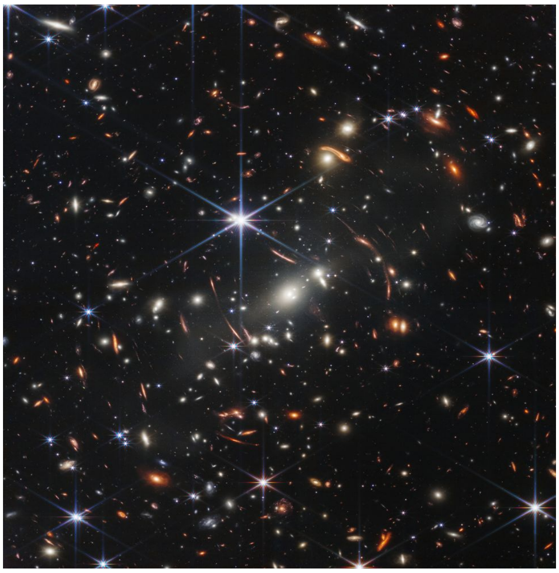


FIGURE S1.1: James Webb Space Telescope image with a large number of lensed galaxies. Credits: NASA, ESA, CSA, and STScI.

According to general relativity, gravity is a geometric property of space-time, the four-dimensional fabric that composes the universe. This fabric includes three dimensions of space and one of time. In this flexible fabric, matter causes space-time to curve, and this curvature dictates how objects move. For example, Earth orbits the Sun because it moves along the curved path generated by the Sun's mass in space-time. A fascinating consequence of these curved paths is the bending of light, known as *gravitational lensing*. This effect is illustrated in Fig. S1.1, captured by the James Webb Space Telescope, where several galaxies appear to have duplicates. In this scenario, a massive object, like a galaxy, acts as a "magnifying glass," allowing us to observe objects located behind it and causing these duplicates as a consequence.

Another consequence of general relativity is that massive astronomical objects can distort the very fabric of space-time. Imagine space-time as a lake of crystal-clear water, so transparent that you can see all the way to the bottom. When a rock is thrown into this lake, it creates ripples on the surface. In this analogy, the lake represents space-time, and the ripples are caused by astronomical cataclysm. These ripples are known as *gravitational waves*, and they travel across the universe at the speed of light. After millions of years, they can reach Earth, where we can now detect them.

Detecting ripples in space-time

Stars, much like humans, are born, age over incredibly long timescales, and eventually die. During their lifetimes, these massive clouds of compressed hydrogen continuously convert hydrogen into helium through a process called *nuclear fusion*, which generates energy. Once they exhaust their hydrogen supply, stars begin fusing heavier elements to produce energy, progressing through the periodic table until they reach iron.

At the end of their lives, very massive stars can no longer support their own weight and collapse inward. This collapse leads to a spectacular outward explosion called a supernova, one of the most powerful events in the universe. In Fig. S1.2, you can see one of the most famous remnants of such an explosion: the Crab Nebula, which was formed by a supernova observed nearly 1,000 years ago in 1054 CE. Even though supernovae are incredibly energetic, the gravitational waves they produce are very weak and hard to detect. Not all supernova explosions leave behind visible remnants like the Crab Nebula. What remains after a supernova depends on the original star's mass. Sometimes, a supernova leaves behind no remnant at all, while other times it forms a compact object like a neutron star or a black

In astrophysics, a compact object refers to a massive celestial body that is relatively small. Supernova explosions can form either a black hole or a neutron star. A neutron star is an extremely dense, compact star composed of neutrons, while a black hole is an extremely dense, compact object that deforms space-time itself, absorbing everything that comes near it, including light. For these types of objects, the description of gravity provided by Sir Isaac Newton is no longer valid due to their extreme conditions, and we need to adopt the theory of general relativity.

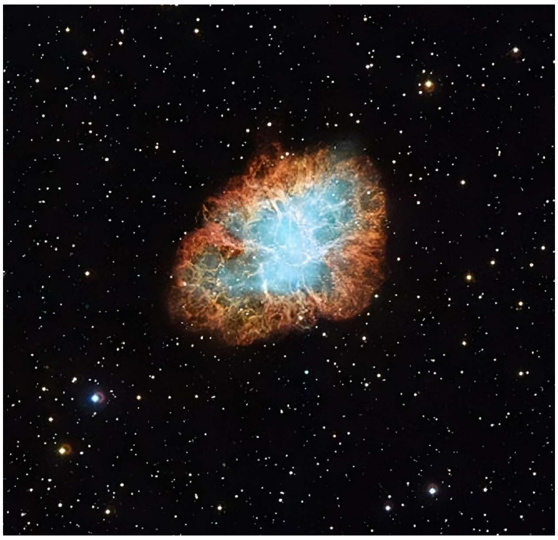


FIGURE S1.2: Hubble Space Telescope image of the Crab Nebula. Credits: NASA and STScI.

In nature, stars often form in binary systems, making it common to find pairs such as binary neutron stars, binary black holes, or a neutron star and a black hole. These celestial bodies orbit each other for billions of years, and during this time, their orbits gradually shrink due to the emission of gravitational waves, which are ripples in space-time. Eventually, the two compact objects collide in an astronomical cataclysm, merging to form a larger compact object. The gravitational waves emitted from this collision travel throughout the universe for millions of years, passing by countless galaxies. As they propagate, gravitational waves, like ripples on the surface of a lake, deform space-time, including Earth and everything on it. However, we remain unaware of their effects because the distortions they cause are extremely small.

To measure gravitational waves, scientists have built several detectors forming a network: LIGO Hanford and LIGO Livingston in the USA, Virgo in Italy, and KAGRA in Japan. These extremely complex instruments, known as interferometers, have exquisite precision to be able to detect the tiny distortions caused by gravitational waves, measuring changes in distance as small as $\sim 1/10,000$th the diameter of a proton. You can think of these machines as the most precise seismographs in the world—devices that detect and record waves—but instead of measuring earthquakes caused by tectonic plate movements, they measure ripples in space-time generated by cataclysmic events like supernova explosions and the collisions of black holes or neutron stars. Due to their incredible sensitivity, these detectors can be affected by terrestrial background noise, such as electrical malfunctions, thunderstorms, or even human activity. This background noise can produce artefacts in the data, known as *glitches*, that mimic gravitational wave signals and hinder their detection.

In 2015, the LIGO and Virgo detectors confirmed the existence of gravitational waves from a merger of two black holes, labelled as GW150914, nearly a century after Albert Einstein predicted them in his theory of general relativity. This groundbreaking discovery was awarded the Nobel Prize in Physics in 2017, and, as of this writing, over 90 gravitational wave signals have been detected since then. The immense work of the LIGO-Virgo-KAGRA collaboration has opened a new and exciting way to listen to the symphony of the cosmos and unveil its mysteries. Future research in gravitational wave astronomy promises to deepen our understanding of the universe by probing the densest and most energetic regions of cosmic objects, which were hidden from astronomers' sight up until now.

Exploring the frontier of gravitational wave detection

Just as the soft melody of a street musician's flute can be drowned out by the bustling noise of the city, gravitational waves are incredibly subtle signals often masked by various sources of detector noise. Detecting a gravitational wave signal in current ground-based detectors is like finding a needle in a haystack: we capture only a few seconds of gravitational wave signals amidst approximately $1,296,000$ seconds (two weeks) of detector noise.

Current searches for transient gravitational waves are like finding a needle in a haystack. These searches rely on two main approaches: modeled searches and unmodeled searches. Modeled searches compare theoretical models from general relativity to the detector data to determine if a given signal is of astronomical origin. Unmodeled searches, on the other hand, look for loud signals in the detector data with little to no prior information. However, both methods face challenges because gravitational wave detectors produce glitches that can mimic gravitational wave signals. To overcome this, both types of searches also rely on a key idea: if a signal is observed simultaneously in multiple detectors, it is more likely to have an astronomical origin.

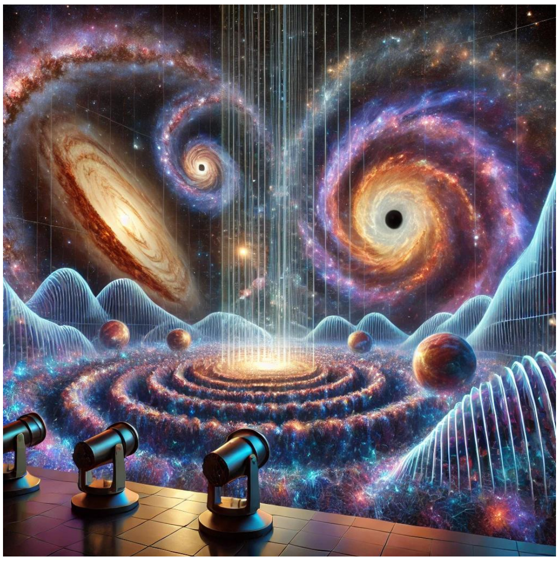


Figure S1.3: Interpretation of the cosmic auditorium. Credits:DALL · E 2

As current ground-based detectors are upgraded, they will become more sensitive to gravitational waves. The next generation of detectors, such as the Einstein Telescope, Cosmic Explorer, and LISA, is expected to be even more sensitive, allowing us to access the full symphony of the cosmos. In this scenario, we will enter the cosmic auditorium and need to disentangle the sounds of different instruments. Each gravitational wave source, such as colliding black holes or merging neutron stars, contributes its own unique "note" to the cosmic symphony, much like individual instruments in an orchestra. As we enhance our ability to detect these faint cosmic "notes," we will uncover the rich tapestry of events that shape our universe. However new detectors will pose new data analysis challenges.

Just as the creation of the internet revolutionized our society, the development of artificial intelligence (AI) is shaping it in profound ways every day. In recent years, we have advanced the ability of machines to learn and perceive the world around us in a manner similar to humans. This progress has led to remarkable milestones, including humanoid robots, real-time language translation, autonomous vehicles, and generative algorithms. These AI-driven technologies demonstrate the ability to analyze data and identify patterns at remarkable speeds, offering solutions and insights that were previously unattainable. As an example, in Fig. S1.3 we show the interpretation of the previous paragraph of DALL · E 2, an AI algorithm.

Thanks to the success of artificial intelligence (AI), especially machine learning—a branch of AI—across many fields, scientists in large physics experiments are using these algorithms to find interesting patterns in the data. Gravitational wave research is no exception. Machine learning methods are versatile and can uncover surprising patterns in the data. However, to make these methods flexible, they often become complex, making them hard to understand—a challenge known as the "black box" problem. Additionally, because machine learning algorithms learn from data, they need to be given high-quality data and meaningful features. Finally, humans are needed to evaluate how well these algorithms perform, which is not always easy, especially when discovering new patterns.

Unleashing the power of machine learning

Even though supernova explosions are among the most powerful events in the universe, the gravitational waves they produce are incredibly faint and difficult to detect. Successfully capturing these signals will give us valuable insights into the inner workings of these explosions, helping us understand how they evolve.

We know that supernovae can create black holes that are about 3 to 100 times the mass of our Sun. On the other hand, we observe supermassive black holes that can be millions of times the Sun's mass, like Sagittarius A* at the center of our Milky Way galaxy, as shown in Fig. S1.4. Scientists believe these giant black holes formed over time as smaller black holes merged together. This leaves us with a mystery: the missing link between smaller black holes created by supernovae and the enormous supermassive black holes. These are called intermediate black holes, which have masses between 100 and 1,000 times that of the Sun. Detecting more gravitational waves from intermediate black holes, like those from the event GW190521, could help us piece together the story of how supermassive black holes form and how they shape the galaxies around them.

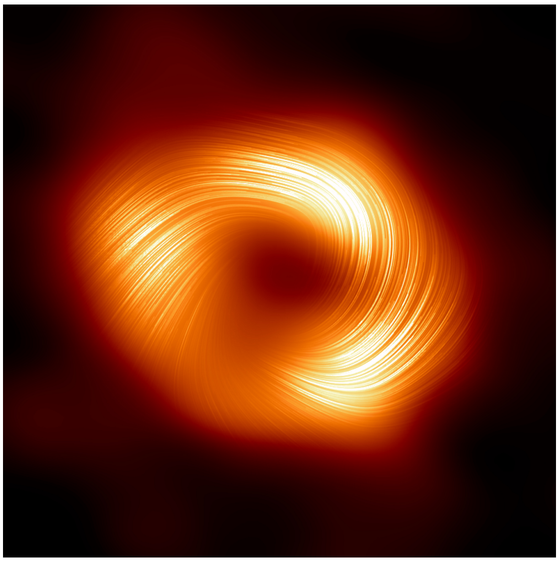


FIGURE S1.4: Sagittarius A, the supermassive black hole at the center of the Milky Way. Credits: EHT.*

Mitigating glitches

Gravitational wave detectors are incredibly complex machines made up of many optical components. These detectors have thousands of sensors that constantly monitor their condition in real time. Due to their high sensitivity, glitches lasting only a few seconds often occur in different parts of the detector and show up in the main data stream. Because we can't predict when a glitch will appear, it's crucial to understand their forms and patterns to identify them and reduce their impact on the data. However, this is a challenging task because glitches happen frequently, about once every second.

A promising way to tackle the issue of glitches in gravitational wave detectors is to categorize them based on their shapes, or "morphologies", in the main stream of the detector. By combining human expertise with machine learning algorithms that are good at finding patterns, we can better identify these glitches, which is the first step toward reducing their impact. However, there are challenges in this approach. One difficulty is determining when a glitch is absent from the data, which makes it hard and time-consuming to fully understand the results of search algorithms. Another issue is that the exact cause of a glitch is often unknown, making it challenging to find effective solutions to prevent them.

In this thesis, we address the first problem by using a generative machine learning algorithm to learn the population of one of the most abundant glitches: Blips. Blips are short lived and affect the frequencies of interest of gravitational wave searches, particularly hindering the searches of supernova and intermediate mass black holes. By creating a synthetic population of these glitches, we can improve our search algorithms in a controlled environment.

We also address the second issue using a machine learning algorithm for anomaly detection. We compress the data of auxiliary channels where these glitches are produce. By compressing this data, the algorithm can identify patterns and learn about the glitches based on their physical processes within these channels. This approach allows us to uncover unexpected patterns and gain a better understanding of the complex nature of these glitches.

Pattern recognition for intermediate-mass black holes and supernovae

Machine learning algorithms are only as effective as the data they are given. To discover interesting patterns, these algorithms require large amounts of high-quality data. So far, we have detected over 90 gravitational waves, but this number isn't sufficient to teach a machine learning algorithm what a gravitational wave looks like. Because of this limitation, we rely heavily on simulations of these astronomical events. We have a good understanding of the physics behind black hole mergers, which allows us to simulate them accurately. However, simulating a supernova explosion is much more complex. A supernova involves several forces and processes, making it difficult to model. Creating a simulation of a supernova is also extremely resource-intensive, often taking months of computation time on a supercomputer.

Machine learning algorithms are excellent at learning patterns and have a remarkable ability to generalize from them. In this thesis, we leveraged this ability by generating signals that are somewhat similar to actual supernova simulations. The key idea is that machine learning algorithms can first learn from these approximate dataset and then refine their understanding using the actual simulations. Additionally, the machine learning method learns that gravitational waves are usually detected in several detectors simultaneously.

Traditional search algorithms for intermediate-mass black holes have been refined over the last 20 years to effectively transform the stream of data from gravitational wave detectors into meaningful features. However, these search algorithms primarily rely on simulations. Machine learning algorithms, on the other hand, can learn directly from both simulations and glitches. However, without the right features, it can be very difficult for machine learning to tell the difference between real signals and noise. In this thesis, we develop a symbiotic relationship between traditional search methods and machine learning algorithms. The machine learning models learn the features generated by the traditional search algorithms, enabling them to better distinguish genuine gravitational waves from background noise.

The work in this thesis aims to enhance the detection of gravitational waves from supernovae and intermediate-mass black holes while reducing the impact of glitches. We have explored the frontiers of transient gravitational wave detection by unleashing the power of machine learning.

Openbare Samenvatting

Zwaartekracht: De Onzichtbare Architect van het Universum

Zwaartekracht is de zwakste van de fundamentele krachten, en we ervaren het elke dag hier op aarde. Het is zo vertrouwd dat we de effecten ervan nauwelijks opmerken, omdat we het gemakkelijk kunnen overwinnen door slechts één vinger op te tillen. Echter, op kosmische schaal gedraagt zwaartekracht zich heel anders. Het universum is een dynamische en woeste plek, gevuld met miljarden sterren die geboren worden in chaotische en energieke omgevingen. Deze sterren brengen hun leven door met het produceren van energie via kernfusie — een proces dat de sterren aandrijft — en eindigen vaak in massale explosies die krachtig genoeg zijn om hele werelden te vernietigen. Ondertussen zijn we hier op onze kleine thuisplaneet, de aarde, ver verwijderd van deze astronomische catastrofes. We vragen ons af welke andere werelden sterven en welke nieuwe worden geboren rond die verre, stralende sterren.

Zwaartekracht, vaak de onzichtbare architect van het universum genoemd, speelt een cruciale rol in het vormgeven van het universum. Het dicteert de banen van planeten, beheerst de vorming van sterren en houdt sterrenstelsels bijeen. Als de kracht die de grote structuren van het universum bij elkaar houdt, is de invloed van de zwaartekracht zowel subtiel als diepgaand. De aard van de zwaartekracht heeft de menselijke geest sinds de oudheid gefascineerd, wat heeft geleid tot eeuwen van wetenschappelijk onderzoek.

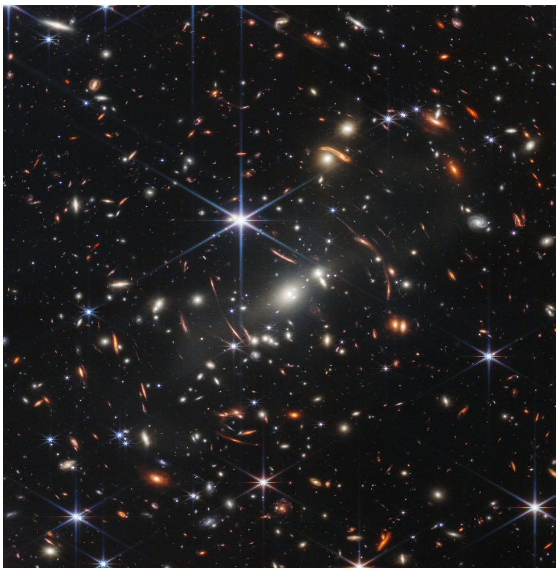


Figure S2.1: *James Webb Space Telescope afbeelding met een groot aantal sterrenstelsels. Credits: NASA, ESA, CSA en STScI.*

In de 17e eeuw revolutioneerde Sir Isaac Newton ons begrip met zijn wet van universele zwaartekracht, die beschreef hoe zwaartekracht objecten beïnvloedt. Echter, hoewel het werk van Newton uitlegde hoe zwaartekracht werkt, bleef de ware aard van deze kracht een mysterie. Pas in de 20e eeuw introduceerde Albert Einstein de theorie van de algemene relativiteitstheorie, die een verenigde beschrijving van zwaartekracht bood en de weg baande voor de moderne natuurkunde.

Volgens de algemene relativiteitstheorie is zwaartekracht een geometrische eigenschap van de ruimtetijd, het vierdimensionale weefsel dat het universum vormt. Dit weefsel omvat drie dimensies van ruimte en één van tijd. In dit flexibele weefsel zorgt materie ervoor dat ruimtetijd kromt, en deze kromming bepaalt hoe objecten bewegen. Bijvoorbeeld, de aarde draait om de zon omdat deze beweegt langs het kromme pad dat wordt veroorzaakt door de massa van de zon in de ruimtetijd. Een fascinerend gevolg van deze kromme paden is de buiging van licht, bekend als *gravitatie-lensvorming*. Dit effect wordt geïllustreerd in Fig. S2.1, vastgelegd door de James Webb Space Telescope, waar verschillende sterrenstelsels duplicaten lijken te hebben. In dit scenario fungeert een massief object, zoals een sterrenstelsel, als een “vergrootglas” en stelt ons in staat objecten te observeren die erachter liggen en veroorzaakt deze duplicaten als gevolg.

Een ander gevolg van de algemene relativiteitstheorie is dat massieve astronomische objecten het weefsel van de ruimtetijd zelf kunnen vervormen. Stel je de ruimtetijd voor als een meer van kristalhelder water, zo transparant dat je helemaal tot op de bodem kunt kijken. Wanneer een steen in dit meer wordt gegooid, creëert het rimpelingen op het oppervlak. In deze analogie vertegenwoordigt het meer de ruimtetijd, en de rimpelingen worden veroorzaakt door een astronomische catastrofe. Deze rimpelingen staan bekend als *zwaartekrachtsgolven*, en ze reizen met de snelheid van het licht door het universum. Na miljoenen jaren kunnen ze de aarde bereiken, waar we ze nu kunnen detecteren.

Het detecteren van rimpelingen in de ruimtetijd

Sterren, net als mensen, worden geboren, verouderen over ongelooflijk lange tijdschalen, en sterven uiteindelijk. Tijdens hun leven zetten deze enorme wolken van gecomprimeerd waterstof continu waterstof om in helium via een proces genaamd *kernfusie*, dat energie genereert. Zodra ze hun voorraad waterstof hebben uitgeput, beginnen sterren zwaardere elementen te fuseren om energie te produceren, voortschrijdend door het periodiek systeem totdat ze ijzer bereiken.

Aan het einde van hun leven kunnen zeer massieve sterren hun eigen gewicht niet meer dragen en storten ze in. Deze instorting leidt tot een spectaculaire explosie naar buiten, een supernova genaamd, een van de krachtigste gebeurtenissen in het universum. In Fig. S2.2 zie je een van de beroemdste overblijfselen van zo’n explosie: de Krabnevel, die werd gevormd door een supernova die bijna duizend jaar geleden in 1054 CE werd waargenomen. Hoewel supernovae ongelooflijk energiek zijn, zijn de zwaartekrachtsgolven die ze produceren zeer zwak en moeilijk te detecteren. Niet alle supernova-explosies laten zichtbare overblijfselen achter zoals de Krabnevel. Wat er na een supernova overblijft, hangt af van de massa van de oorspronkelijke ster. Soms laat een supernova helemaal geen overblijfsel achter, terwijl het andere keren een compact object vormt zoals een neutronenster of een zwart gat.

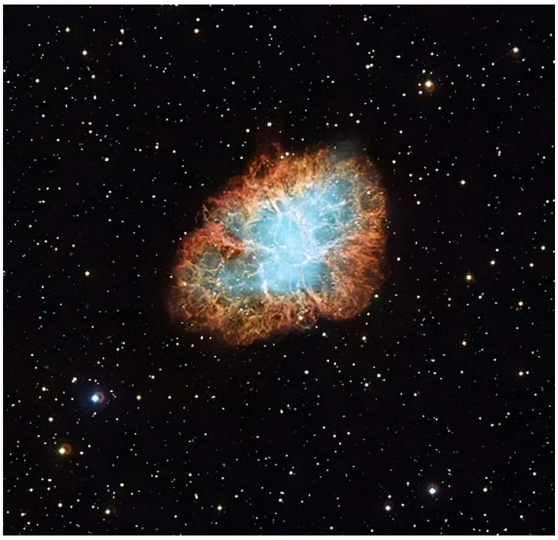


FIGURE S2.2: Hubble Space Telescope afbeelding van de Krabnevel. Credits: NASA en STScI.

In de astrofysica verwijst een compact object naar een massief hemellichaam dat relatief klein is. Supernova-explosies kunnen ofwel een zwart gat of een neutronenster vormen. Een neutronenster is een extreem dichte, compacte ster die bestaat uit neutronen, terwijl een zwart gat een extreem dichte, compacte massa is die de ruimtetijd zelf vervormt en alles absorbeert wat in de buurt komt, inclusief licht. Voor dit soort objecten is de beschrijving van de zwaartekracht zoals gegeven door Sir Isaac Newton niet langer geldig vanwege hun extreme omstandigheden, en we moeten de theorie van de algemene relativiteit toepassen.

In de natuur vormen sterren vaak binairen systemen, waardoor het gebruikelijk is om paren te vinden zoals binaire neutronensterren, binaire zwarte gaten, of een neutronenster en een zwart gat. Deze hemellichamen draaien miljarden jaren om elkaar heen, en gedurende deze tijd krimpen hun banen geleidelijk door de emissie van zwaartekrachtsgolven. Uiteindelijk botsen de twee compacte objecten in een astronomische catastrofe en smelten ze samen om een groter compact object te vormen. De zwaartekrachtsgolven die vrijkomen bij deze botsing reizen miljoenen jaren door het universum en passeren talloze sterrenstelsels. Terwijl ze zich voortplanten, vervormen zwaartekrachtsgolven, net als rimpelingen op het oppervlak van een meer, de ruimtetijd, inclusief de aarde en alles daarop. We blijven echter onbewust van hun effecten omdat de vervormingen die ze veroorzaken extreem klein zijn.

Om zwaartekrachtsgolven te meten, hebben wetenschappers verschillende detectoren gebouwd die een netwerk vormen: LIGO Hanford en LIGO Livingston in de VS, Virgo in Italië en KAGRA in Japan. Deze extreem complexe instrumenten, bekend als interferometers, hebben een verfijnde precisie om de kleine vervormingen veroorzaakt door zwaartekrachtsgolven te detecteren, en meten veranderingen in afstand die zo klein zijn als $\sim 1/10,000$ van de diameter van een proton. Je kunt deze machines zien als de meest nauwkeurige seismografen ter wereld—apparaten die golven detecteren en registreren—maar in plaats van aardbevingen veroorzaakt door bewegingen van tektonische platen te meten, meten ze rimpelingen in de ruimtetijd die worden gegenereerd door catastrofale gebeurtenissen zoals supernova-explosies en de botsingen van zwarte gaten of neutronensterren. Vanwege hun ongelooflijke gevoeligheid kunnen deze detectoren worden beïnvloed door aardse achtergrondgeluiden, zoals elektrische storingen, onweersbuien of zelfs menselijke activiteiten. Deze achtergrondgeluiden kunnen artefacten in de gegevens veroorzaken, bekend als *glitches*, die zwaartekrachtsgolfsignalen kunnen nabootsen en hun detectie bemoeilijken.

In 2015 bevestigden de LIGO en Virgo detectoren het bestaan van zwaartekrachtsgolven afkomstig van de samensmelting van twee zwarte gaten, genaamd GW150914, bijna een eeuw nadat Albert Einstein ze had voorspeld in zijn theorie van de algemene relativiteit. Deze baanbrekende ontdekking werd bekroond met de Nobelprijs voor de Natuurkunde in 2017, en, op het moment van schrijven, zijn er sindsdien meer dan 90 zwaartekrachtsgolfsignalen gedetecteerd. Het immense werk van de LIGO-Virgo-KAGRA samenwerking heeft een nieuwe en spannende manier mogelijk gemaakt om te luisteren naar de symfonie van het universum en zijn mysteries te onthullen. Toekomstig onderzoek in de zwaartekrachtsgolvenastronomie belooft ons begrip van het universum te verdiepen door de dichtste en meest energieke regio's van kosmische objecten te onderzoeken, die tot nu toe verborgen waren voor astronomen.

Het verkennen van de grens van gravitatiegolven-detectie.

Net zoals de zachte melodie van de fluit van een straatmuzikant kan worden overstemd door het drukke stadsgeruis, zijn zwaartekrachtsgolven ongelooflijk subtiele signalen die vaak worden gemaskeerd door verschillende bronnen van waarnemingsruis. Het detecteren van een zwaartekrachtsgolfsignaal in de huidige op de aarde gevestigde detectoren is als het zoeken naar een speld in een hooiberg: we vangen slechts enkele seconden zwaartekrachtsgolfsignalen op te midden van ongeveer 1.296.000 seconden (twee weken) aan waarnemingsruis.

Huidige zoektochten naar transiënte zwaartekrachtsgolven zijn als het zoeken naar een speld in een hooiberg. Deze zoektochten vertrouwen op twee hoofdbenaderingen: gemodelleerde zoektochten en ongemodelleerde zoektochten. Gemodelleerde zoektochten vergelijken theoretische modellen uit de algemene relativiteit met de detectorgegevens om te bepalen of een bepaald signaal een astronomische oorsprong heeft.

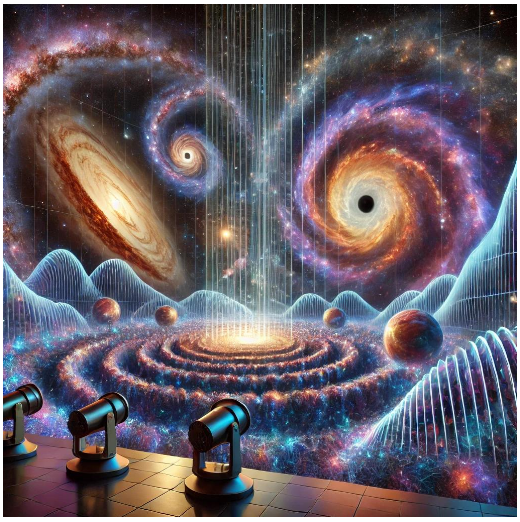


FIGURE S2.3: Interpretatie van het kosmisch auditorium. Credits: DALL · E 2

Aan de andere kant zoeken ongemodelleerde zoektochten naar harde signalen in de detectorgegevens met weinig tot geen voorkennis. Beide methoden staan echter voor uitdagingen omdat zwaartekrachtsgolfdetectoren glitches kunnen produceren die zwaartekrachtsgolfsignalen nabootsen. Om dit te omzeilen, vertrouwen beide soorten zoektochten ook op een belangrijk idee: als een signaal tegelijkertijd in meerdere detectoren wordt waargenomen, is het waarschijnlijker dat het een astronomische oorsprong heeft.

Naarmate de huidige op de aarde gevestigde detectoren worden geüpgraded, zullen ze gevoeliger worden voor zwaartekrachtsgolven. De volgende generatie detectoren, zoals de Einstein-telescoop, Cosmic Explorer en LISA, zal naar verwachting nog gevoeliger zijn, waardoor we toegang krijgen tot de volledige symfonie van het universum. In dit scenario zullen we het kosmische auditorium betreden en moeten we de geluiden van verschillende instrumenten ontcijferen. Elke bron van zwaartekrachtsgolven, zoals botsende zwarte gaten of samensmeltende neutronensterren, draagt zijn eigen unieke “noot” bij aan de kosmische symfonie, net zoals individuele instrumenten in een orkest. Naarmate we ons vermogen verbeteren om deze zwakke kosmische “noten” te detecteren, zullen we de rijke verscheidenheid aan gebeurtenissen die ons universum vormgeven, onthullen. Nieuwe detectoren zullen echter nieuwe data-analyse-uitdagingen met zich meebrengen.

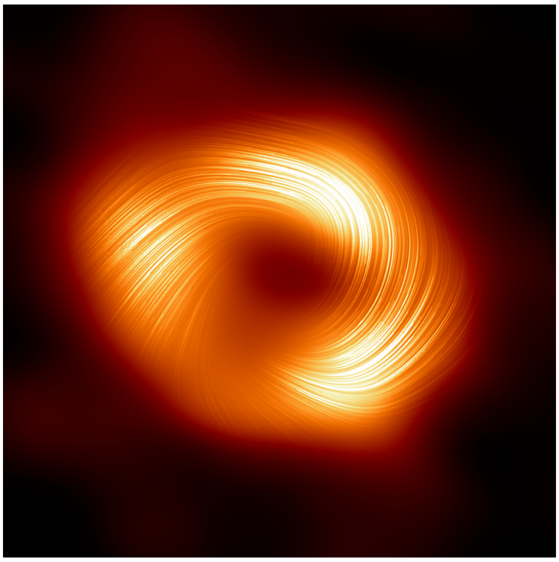


Figure S2.4: Sagittarius A, het supermassieve zwarte gat in het centrum van de Melkweg. Credits: EHT.*

Net zoals de totstandkoming van het internet onze samenleving heeft gerevolutioneerd, vormt de ontwikkeling van kunstmatige intelligentie (AI) deze elke dag op ingrijpende wijze. In de afgelopen jaren hebben we de mogelijkheid van machines om te leren en de wereld om ons heen waar te nemen op een manier die lijkt op die van mensen, verbeterd. Deze vooruitgang heeft geleid tot opmerkelijke mijlpalen, waaronder humanoïde robots, realtime taalvertaling, autonome voertuigen en generatieve algoritmen. Deze door AI aangedreven technologieën demonstreren het vermogen om gegevens te analyseren en patronen te identificeren met verbazingwekkende snelheden, wat oplossingen en inzichten biedt die voorheen onbereikbaar waren. Als voorbeeld tonen we in Fig. S2.3 de interpretatie van de vorige paragraaf van DALL · E 2, een AI-algoritme.

Dankzij het succes van kunstmatige intelligentie (AI), met name machine learning—een tak van AI—in vele velden, gebruiken wetenschappers in grote fysica-experimenten deze algoritmen om interessante patronen in de gegevens te vinden. Zwaartekrachtsgolvenonderzoek is hierop geen uitzondering. Machine learning-methoden zijn veelzijdig en kunnen verrassende patronen in de gegevens onthullen. Om deze methoden echter flexibel te maken, worden ze vaak complex, waardoor ze moeilijk te begrijpen zijn—een uitdaging die bekend staat als het "black box"-probleem. Bovendien moeten machine learning-algoritmen, omdat ze leren van gegevens, worden voorzien van hoogwaardige gegevens en betekenisvolle kenmerken. Ten slotte zijn mensen nodig om te beoordelen hoe goed deze algoritmen presteren, wat niet altijd eenvoudig is, vooral niet bij het ontdekken van nieuwe patronen.

De kracht van machine learning ontketenen

Hoewel supernova-explosies tot de krachtigste gebeurtenissen in het universum behoren, zijn de zwaartekrachtsgolven die ze produceren ongelooflijk zwak en moeilijk te detecteren. Het succesvol vastleggen van deze signalen zal ons waardevolle inzichten geven in de inwendige werking van deze explosies en ons helpen begrijpen hoe ze zich ontwikkelen.

We weten dat supernovae zwarte gaten kunnen creëren die ongeveer 3 tot 100 keer de massa van onze zon hebben. Aan de andere kant observeren we supermassieve zwarte gaten die miljoenen keren de massa van de zon kunnen zijn, zoals Sagittarius A* in het centrum van onze Melkweg, zoals getoond in Fig. S2.4. Wetenschappers geloven dat deze gigantische zwarte gaten in de loop van de tijd zijn ontstaan doordat kleinere zwarte gaten samenkwamen. We staan voor een raadsel: de ontbrekende schakel tussen kleinere zwarte gaten die door supernovae worden gecreëerd en de enorme supermassieve zwarte gaten. Deze worden Zwarte gaten met een middelmatige massa genoemd, die massa's hebben tussen 100 en 1.000 keer die van de zon. Het detecteren van meer zwaartekrachtsgolven van iZwarte gaten met een middelmatige massa, zoals die van het evenement GW190521, zou ons kunnen helpen het verhaal van hoe supermassieve zwarte gaten ontstaan en hoe ze de sterrenstelsels om hen heen vormgeven, in kaart te brengen.

Het tegengaan van Glitches

Zwaartekrachtsgolfdetectoren zijn ongelooflijk complexe machines die bestaan uit veel optische componenten. Deze detectoren hebben duizenden sensoren die voortdurend hun toestand in realtime bewaken. Vanwege hun hoge gevoeligheid komen glitches die slechts enkele seconden duren vaak voor in verschillende delen van de detector en verschijnen ze in de hoofdstroom van gegevens. Omdat we niet kunnen voorspellen wanneer een glitch zal verschijnen, is het cruciaal om hun vormen en patronen te begrijpen om ze te identificeren en hun impact op de gegevens te verminderen. Dit is echter een uitdagende taak omdat glitches vaak voorkomen, ongeveer eens per seconde.

Een veelbelovende manier om het probleem van glitches in zwaartekrachtsgolfdetectoren aan te pakken, is door ze te categoriseren op basis van hun vormen, of "morfologieën", in de hoofdstroom van de detector. Door menselijke expertise te combineren met machine learning-algoritmen die goed zijn in het vinden van patronen, kunnen we deze glitches beter identificeren, wat de eerste stap is naar het verminderen van hun impact. Er zijn echter uitdagingen bij deze aanpak. Een moeilijkheid is het bepalen wanneer een glitch afwezig is in de gegevens, wat het moeilijk en tijdrovend maakt om de resultaten van zoekalgoritmen volledig te doorgronden. Een ander probleem is dat de exacte oorzaak van een glitch vaak onbekend is, waardoor het moeilijk is om effectieve oplossingen te vinden om ze te voorkomen.

In dit proefschrift pakken we het eerste probleem aan door een generatief machine learning-algoritme te gebruiken om de populatie van een van de meest voorkomende glitches te leren: Blips. Blips zijn van korte duur en beïnvloeden de frequenties die van belang zijn voor het zoeken naar zwaartekrachtsgolven, met name het zoeken naar supernovae en zwarte gaten met een middelmatige massa. Door een kunstmatige populatie van deze glitches te creëren, kunnen we onze zoekalgoritmen in een gecontroleerde omgeving verbeteren.

We pakken ook het tweede probleem aan met behulp van een machine learning-algoritme voor anomaliedetectie. We comprimeren de gegevens van hulpkanalen waar deze glitches worden geproduceerd. Door deze gegevens te comprimeren, kan het algoritme patronen identificeren en leren over de glitches op basis van hun fysieke processen binnen deze kanalen. Deze aanpak stelt ons in staat om onverwachte patronen bloot te leggen en een beter begrip te krijgen van de complexe aard van deze glitches.

Patroonherkenning voor zwarte gaten met een middelmatige massa en supernovae

Machine learning-algoritmen zijn slechts zo effectief als de gegevens die ze krijgen. Om interessante patronen te ontdekken, hebben deze algoritmen grote hoeveelheden hoogwaardige gegevens nodig. Tot nu toe hebben we meer dan 90 zwaartekrachtsgolven gedetecteerd, maar dit aantal is niet voldoende om een machine learning-algoritme te leren hoe een zwaartekrachtsgolf eruitziet. Vanwege deze beperking vertrouwen we sterk op simulaties van deze astronomische gebeurtenissen. We hebben een goed begrip van de natuurkunde achter fusies van zwarte gaten, waardoor we ze nauwkeurig kunnen simuleren. Het simuleren van een supernova-explosie is echter veel complexer. Een supernova omvat verschillende krachten en processen, wat het moeilijk maakt om te modelleren. Het maken van een simulatie van een supernova is ook extreem intensief in termen van middelen, en kost vaak maanden aan rekentijd op een supercomputer.

Machine learning-algoritmen zijn uitstekend in het leren van patronen en hebben een opmerkelijk vermogen om van daaruit te generaliseren. In dit proefschrift hebben we gebruik gemaakt van dit vermogen door signalen te genereren die enigszins lijken op feitelijke supernovasimulaties. Het belangrijkste idee is dat machine learning-algoritmen eerst kunnen leren van deze benaderde dataset en vervolgens hun begrip kunnen verfijnen met behulp van de feitelijke simulaties. Bovendien leert de machine learning-methode dat zwaartekrachtsgolven meestal in meerdere detectoren tegelijkertijd worden gedetecteerd.

Traditionele zoekalgoritmen voor zwarte gaten met een middelmatige massa zijn in de afgelopen 20 jaar verfijnd om de datastroom van zwaartekrachtsgolfdetectoren effectief om te zetten in betekenisvolle kenmerken. Deze zoekalgoritmen vertrouwen echter voornamelijk op simulaties. Machine learning-algoritmen kunnen daarentegen direct leren van zowel simulaties als glitches. Zonder de juiste kenmerken kan het echter erg moeilijk zijn voor machine learning om het verschil te zien tussen echte signalen en ruis. In dit proefschrift ontwikkelen we een symbiotische relatie tussen traditionele zoekmethoden en machine learning-algoritmen. De machine learning-modellen leren de kenmerken die door de traditionele zoekalgoritmen worden gegenereerd, waardoor ze beter in staat zijn om echte zwaartekrachtsgolven van achtergrondruis te onderscheiden.

Het werk in deze thesis is gericht op het verbeteren van de detectie van gravitatiegolven van supernova's en zwarte gaten van middelmatige massa, terwijl de impact van storingen wordt verminderd. We hebben de grenzen van de detectie van tijdelijke gravitatiegolven verkend door de kracht van machine learning te benutten.

Resumen público

La gravedad: el arquitecto invisible del cosmos

La gravedad es la más débil de las fuerzas fundamentales y la experimentamos todos los días aquí en la Tierra. Es tan familiar que apenas notamos sus efectos, ya que podemos superarla fácilmente levantando un solo dedo. Sin embargo, la gravedad se comporta de manera muy diferente a escala cósmica. El universo es un lugar dinámico y violento, lleno de miles de millones de estrellas nacidas en entornos caóticos y energéticos. Estas estrellas pasan sus vidas produciendo energía a través de la fusión nuclear, un proceso que las alimenta, y a menudo terminan en explosiones masivas lo suficientemente poderosas como para destruir mundos enteros. Mientras tanto, aquí en nuestro pequeño hogar, la Tierra, estamos lejos de estas catástrofes astronómicas. Nos queda preguntarnos qué otros mundos están muriendo y cuáles están naciendo alrededor de esas estrellas distantes y brillantes.

La gravedad, a menudo referida como el arquitecto invisible del cosmos, juega un papel vital en la conformación del universo. Dicta las órbitas de los planetas, gobierna la formación de estrellas y mantiene unidas a las galaxias. Como la fuerza que sostiene las vastas estructuras del universo en su lugar, la influencia de la gravedad es tanto sutil como profunda. La naturaleza de la gravedad ha intrigado a la mente humana desde tiempos antiguos, lo que ha llevado a siglos de investigación científica.

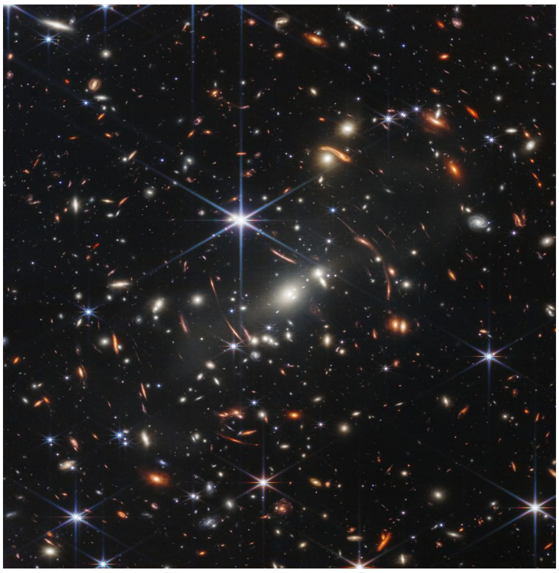


Figure S3.1: Imagen del telescopio espacial James Webb con un gran número de galaxias lentes. Créditos: NASA, ESA, CSA y STScI.

En el siglo XVII, Sir Isaac Newton revolucionó nuestro entendimiento con su ley de gravitación universal, que describía cómo la gravedad afecta a los objetos. Sin embargo, aunque el trabajo de Newton explicaba cómo opera la gravedad, la verdadera naturaleza de esta fuerza seguía siendo un misterio. No fue hasta el siglo XX que Albert Einstein introdujo la teoría de la relatividad general, proporcionando una descripción unificada de la gravedad y allanando el camino para la física moderna.

Según la relatividad general, la gravedad es una propiedad geométrica del espacio-tiempo, el tejido de cuatro dimensiones que compone el universo. Este tejido incluye tres dimensiones de espacio y una de tiempo. En este tejido flexible, la materia provoca la curvatura del espacio-tiempo, y esta curvatura dicta cómo se mueven los objetos. Por ejemplo, la Tierra orbita alrededor del Sol porque se mueve a lo largo del camino curvado generado por la masa del Sol en el espacio-tiempo. Una consecuencia fascinante de estos caminos curvados es la curvatura de la luz, conocida como *lente gravitacional*. Este efecto se ilustra en la Fig. S3.1, capturada por el telescopio espacial James Webb, donde varias galaxias parecen tener duplicados. En este escenario, un objeto masivo, como una galaxia, actúa como una "lupa", permitiéndonos observar objetos situados detrás de él y provocando estos duplicados como consecuencia.

Otra consecuencia de la relatividad general es que los objetos astronómicos masivos pueden distorsionar el mismo tejido del espacio-tiempo. Imagina el espacio-tiempo como un lago de agua cristalina, tan transparente que puedes ver hasta el fondo. Cuando una roca se lanza a este lago, crea ondas en la superficie. En esta analogía, el lago representa el espacio-tiempo y las ondas son causadas por una catástrofe astronómica. Estas ondas son conocidas como *ondas gravitacionales* y viajan a través del universo a la velocidad de la luz. Después de millones de años, pueden llegar a la Tierra, donde ahora podemos detectarlas.

Detección de ondas en el espacio-tiempo

Las estrellas, al igual que los humanos, nacen, envejecen durante escalas de tiempo increíblemente largas y eventualmente mueren. Durante sus vidas, estas enormes nubes de hidrógeno comprimido convierten continuamente hidrógeno en helio a través de un proceso llamado *fusión nuclear*, que genera energía. Una vez que agotan su suministro de hidrógeno, las estrellas comienzan a fusionar elementos más pesados para producir energía, progresando a través de la tabla periódica hasta que llegan al hierro.

Al final de sus vidas, las estrellas muy masivas ya no pueden soportar su propio peso y colapsan hacia adentro. Este colapso conduce a una espectacular explosión hacia afuera, llamada supernova, uno de los eventos más poderosos del universo. En la Fig. S3.2, puedes ver uno de los remanentes más famosos de una explosión de este tipo: la nebulosa del Cangrejo, que fue formada por una supernova observada hace casi 1,000 años en 1054 CE. A pesar de que las supernovas son increíblemente energéticas, las ondas gravitacionales que producen son muy débiles y difíciles de detectar. No todas las explosiones de supernova dejan remanentes visibles como la nebulosa del Cangrejo. Lo que queda después de una supernova depende de la masa original de la estrella. A veces, una supernova no deja ningún remanente, mientras que otras veces forma un objeto compacto como una estrella de neutrones o un agujero negro.

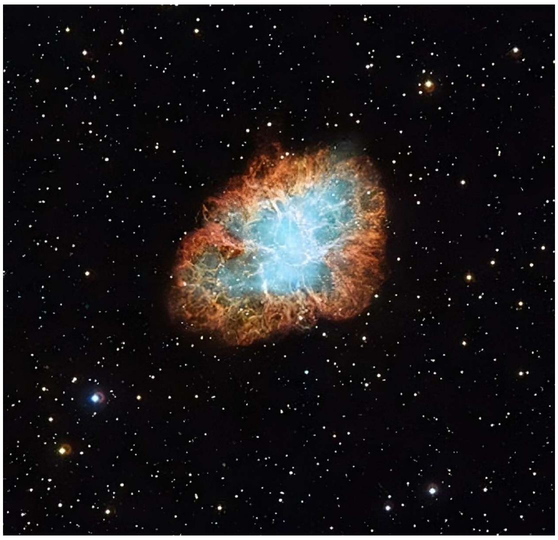


Figure S3.2: Imagen del telescopio espacial Hubble de la nebulosa del Cangrejo. Créditos: NASA y STScI.

En astrofísica, un objeto compacto se refiere a un cuerpo celeste masivo que es relativamente pequeño. Las explosiones de supernovas pueden formar un agujero negro o una estrella de neutrones. Una estrella de neutrones es una estrella extremadamente densa y compacta compuesta por neutrones, mientras que un agujero negro es un objeto extremadamente denso y compacto que deforma el espacio-tiempo en sí mismo, absorbiendo todo lo que se acerca a él, incluida la luz. Para este tipo de objetos, la descripción de la gravedad proporcionada por Sir Isaac Newton ya no es válida debido a sus condiciones extremas, y necesitamos adoptar la teoría de la relatividad general.

En la naturaleza, las estrellas a menudo se forman en sistemas binarios, por lo que es común encontrar pares como estrellas de neutrones binarias, agujeros negros binarios o una estrella de neutrones y un agujero negro. Estos cuerpos celestes orbitan entre sí durante miles de millones de años y, durante este tiempo, sus órbitas se encogen gradualmente debido a la emisión de ondas gravitacionales, que son ondas en el espacio-tiempo. Eventualmente, los dos objetos compactos colisionan en una catástrofe astronómica, fusionándose para formar un objeto compacto más grande. Las ondas gravitacionales emitidas por esta colisión viajan por todo el universo durante millones de años, pasando por innumerables galaxias. A medida que se propagan, las ondas gravitacionales, como ondas en la superficie de un lago, deforman el espacio-tiempo, incluida la Tierra y todo lo que hay en ella. Sin embargo, no somos conscientes de sus efectos porque las distorsiones que causan son extremadamente pequeñas.

Para medir las ondas gravitacionales, los científicos han construido varios detectores que forman una red: LIGO Hanford y LIGO Livingston en los EE.UU., Virgo en Italia y KAGRA en Japón. Estos instrumentos extremadamente complejos, conocidos como interferómetros, tienen una precisión exquisita para poder detectar las diminutas distorsiones causadas por las ondas gravitacionales, midiendo cambios en la distancia tan pequeños como $\sim 1/10,000$ del diámetro de un protón. Puedes pensar en estas máquinas como los sismógrafos más precisos del mundo, dispositivos que detectan y registran ondas, pero en lugar de medir terremotos causados por el movimiento de las placas tectónicas, miden ondas en el espacio-tiempo generadas por eventos catastróficos como explosiones de supernovas y colisiones de agujeros negros o estrellas de neutrones. Debido a su increíble sensibilidad, estos detectores pueden verse afectados por el ruido de fondo terrestre, como fallos eléctricos, tormentas eléctricas o incluso actividades humanas. Este ruido de fondo puede producir artefactos en los datos, conocidos como *glitches*, que pueden imitar las señales de ondas gravitacionales y dificultar su detección.

En 2015, los detectores LIGO y Virgo confirmaron la existencia de ondas gravitacionales provenientes de la fusión de dos agujeros negros, etiquetadas como GW150914, casi un siglo después de que Albert Einstein las predijera en su teoría de la relatividad general. Este descubrimiento revolucionario fue galardonado con el Premio Nobel de Física en 2017, y, hasta la fecha de este escrito, se han detectado más de 90 señales de ondas gravitacionales desde entonces. El inmenso trabajo de la colaboración LIGO-Virgo-KAGRA ha abierto una nueva y emocionante forma de escuchar la sinfonía del cosmos y revelar sus misterios. La investigación futura en astronomía de ondas gravitacionales promete profundizar nuestra comprensión del universo al sondear las regiones más densas y energéticas de los objetos cósmicos, que hasta ahora habían estado ocultas a la vista de los astrónomos.

Explorando la frontera de la detección de ondas gravitacionales.

Así como la suave melodía de una flauta de un músico callejero puede ser ahogada por el bullicioso ruido de la ciudad, las ondas gravitacionales son señales increíblemente sutiles que a menudo están enmascaradas por diversas fuentes de ruido de los detectores. Detectar una señal de onda gravitacional en los detectores terrestres actuales es como encontrar una aguja en un pajar: capturamos solo unos pocos segundos de señales de ondas gravitacionales entre aproximadamente 1.296.000 segundos (dos semanas) de ruido de detector.

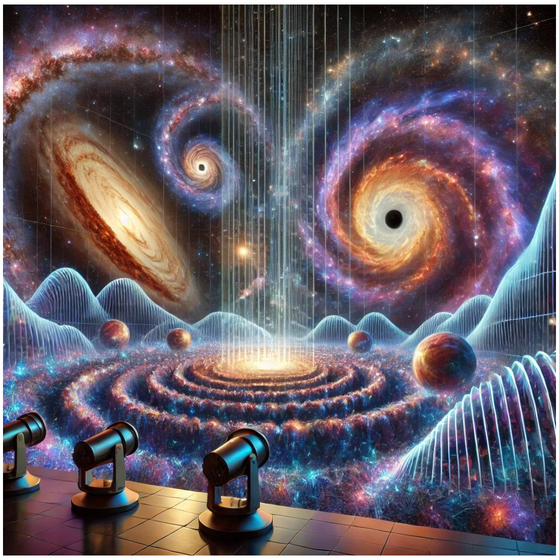


Figure S3.3: Interpretación del auditorio cósmico. Créditos: DALL · E 2

Las búsquedas actuales de ondas gravitacionales transitorias son como encontrar una aguja en un pajar. Estas búsquedas se basan en dos enfoques principales: búsquedas modeladas y búsquedas no modeladas. Las búsquedas modeladas comparan modelos teóricos de la relatividad general con los datos del detector para determinar si una señal dada tiene un origen astronómico. Las búsquedas no modeladas, por otro lado, buscan señales fuertes en los datos del detector con poca o ninguna información previa. Sin embargo, ambos métodos enfrentan desafíos porque los detectores de ondas gravitacionales producen glitches que pueden imitar las señales de ondas gravitacionales. Para superar esto, ambos tipos de búsquedas también se basan en una idea clave: si una señal se observa simultáneamente en múltiples detectores, es más probable que tenga un origen astronómico.

A medida que los detectores terrestres actuales se actualicen, serán más sensibles a las ondas gravitacionales. Se espera que la próxima generación de detectores, como el Telescopio Einstein, Cosmic Explorer y LISA, sea aún más sensible, lo que nos permitirá acceder a la sinfonía completa del cosmos. En este escenario, entraremos en el auditorio cósmico y necesitaremos desentrañar los sonidos de diferentes instrumentos. Cada fuente de onda gravitacional, como los agujeros negros en colisión o las estrellas de neutrones en fusión, contribuye con su propia "nota" única a la sinfonía cósmica, al igual que los instrumentos individuales en una orquesta. A medida que mejoramos nuestra capacidad para detectar estas débiles "notas" cósmicas, descubriremos el rico tapiz de eventos que dan forma a nuestro universo. Sin embargo, los nuevos detectores plantearán nuevos desafíos para el análisis de datos.

Así como la creación de internet revolucionó nuestra sociedad, el desarrollo de la inteligencia artificial (IA) la está moldeando de maneras profundas cada día. En los últimos años, hemos mejorado la capacidad de las máquinas para aprender y percibir el mundo que nos rodea de manera similar a los humanos. Este progreso ha llevado a hitos notables, incluidos robots humanoides, traducción de idiomas en tiempo real, vehículos autónomos y algoritmos generativos. Estas tecnologías impulsadas por IA demuestran la capacidad de analizar datos e identificar patrones a velocidades notables, ofreciendo soluciones e ideas que antes eran inalcanzables. Como ejemplo, en la Fig. S3.3 mostramos la interpretación del párrafo anterior de DALL · E 2, un algoritmo de IA.

Gracias al éxito de la inteligencia artificial (IA), especialmente del aprendizaje automático, un subcampo de la IA, en muchos campos, los científicos de grandes experimentos de física están utilizando estos algoritmos para encontrar patrones interesantes en los datos. La investigación de ondas gravitacionales no es una excepción. Los métodos de aprendizaje automático son versátiles y pueden descubrir patrones sorprendentes en los datos. Sin embargo, para hacer estos métodos flexibles, a menudo se vuelven complejos, lo que los hace difíciles de entender, un desafío conocido como el problema de la "caja negra". Además, dado que los algoritmos de aprendizaje automático aprenden de los datos, necesitan recibir datos de alta calidad y características significativas. Finalmente, se necesita la intervención humana para evaluar el rendimiento de estos algoritmos, lo cual no siempre es fácil, especialmente cuando se descubren nuevos patrones.

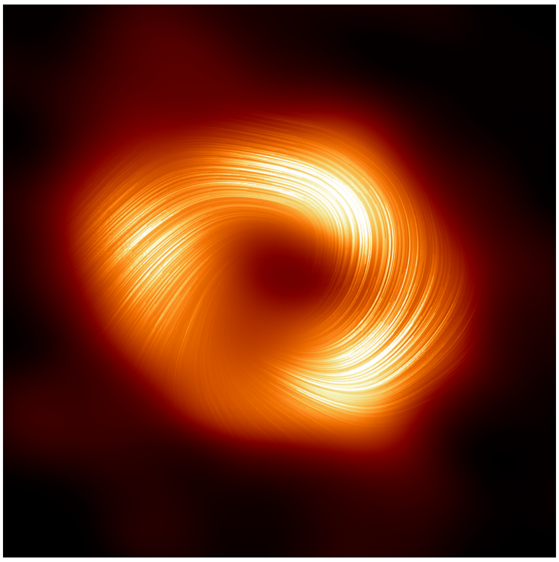


FIGURE S3.4: Sagitario A, el agujero negro supermasivo en el centro de la Vía Láctea. Créditos: EHT.*

Liberando el poder del aprendizaje automático

Aunque las explosiones de supernovas son uno de los eventos más poderosos del universo, las ondas gravitacionales que producen son increíblemente débiles y difíciles de detectar. Capturar con éxito estas señales nos brindará información valiosa sobre el funcionamiento interno de estas explosiones, ayudándonos a comprender cómo evolucionan.

Sabemos que las supernovas pueden crear agujeros negros que son aproximadamente de 3 a 100 veces la masa de nuestro Sol. Por otro lado, observamos agujeros negros supermasivos que pueden tener millones de veces la masa del Sol, como Sagitario A* en el centro de nuestra galaxia, la Vía Láctea, como se muestra en la Fig. S3.4. Los científicos creen que estos agujeros negros gigantes se formaron con el tiempo a medida que los agujeros negros más pequeños se fusionaban entre sí. Esto nos deja con un misterio: el eslabón perdido entre los agujeros negros más pequeños creados por supernovas y los enormes agujeros negros supermasivos. Estos se denominan agujeros negros intermedios, que tienen masas de entre 100 y 1,000 veces la del Sol. Detectar más ondas gravitacionales de agujeros negros intermedios, como las del evento GW190521, podría ayudarnos a reconstruir la historia de cómo se forman los agujeros negros supermasivos y cómo influyen en las galaxias que los rodean.

Mitigación de glitches

Los detectores de ondas gravitacionales son máquinas increíblemente complejas formadas por muchos componentes ópticos. Estos detectores tienen miles de sensores que monitorean constantemente su estado en tiempo real. Debido a su alta sensibilidad, a menudo ocurren glitches que duran solo unos segundos en diferentes partes del detector y aparecen en el flujo de datos principal. Debido a que no podemos predecir cuándo aparecerá un glitch, es crucial comprender sus formas y patrones para identificarlos y reducir su impacto en los datos. Sin embargo, esta es una tarea desafiante porque los glitches ocurren con frecuencia, aproximadamente una vez por segundo.

Una forma prometedora de abordar el problema de los glitches en los detectores de ondas gravitacionales es categorizarlos según sus formas o "morfologías" en el flujo principal del detector. Al combinar la experiencia humana con algoritmos de aprendizaje automático que son buenos para encontrar patrones, podemos identificar mejor estos glitches, que es el primer paso para reducir su impacto. Sin embargo, hay desafíos en este enfoque. Una dificultad es determinar cuándo un glitch está ausente en los datos, lo que hace que sea difícil y requiera mucho tiempo comprender completamente los resultados de los algoritmos de búsqueda. Otro problema es que a menudo se desconoce la causa exacta de un glitch, lo que dificulta encontrar soluciones efectivas para prevenirlos.

En esta tesis, abordamos el primer problema utilizando un algoritmo de aprendizaje automático generativo para aprender la población de uno de los glitches más abundantes: Blips. Los Blips tienen una vida corta y afectan las frecuencias de interés en las búsquedas de ondas gravitacionales, particularmente obstaculizando las búsquedas de supernovas y agujeros negros de masa intermedia. Al crear una población sintética de estos glitches, podemos mejorar nuestros algoritmos de búsqueda en un entorno controlado.

También abordamos el segundo problema utilizando un algoritmo de aprendizaje automático para la detección de anomalías. Comprimimos los datos de los canales auxiliares donde se producen estos glitches. Al comprimir estos datos, el algoritmo puede identificar patrones y aprender sobre los glitches basándose en sus procesos físicos dentro de estos canales. Este enfoque nos permite descubrir patrones inesperados y obtener una mejor comprensión de la naturaleza compleja de estos glitches.

Reconocimiento de patrones para agujeros negros de masa intermedia y supernovas

Los algoritmos de aprendizaje automático son solo tan efectivos como los datos que reciben. Para descubrir patrones interesantes, estos algoritmos requieren grandes cantidades de datos de alta calidad. Hasta ahora, hemos detectado más de 90 ondas gravitacionales, pero este número no es suficiente para enseñar a un algoritmo de aprendizaje automático cómo se ve una onda gravitacional. Debido a esta limitación, dependemos en gran medida de simulaciones de estos eventos astronómicos. Tenemos una buena comprensión de la física detrás de las fusiones de agujeros negros, lo que nos permite simularlas con precisión. Sin embargo, simular una explosión de supernova es mucho más complejo. Una supernova involucra varias fuerzas y procesos, lo que dificulta su modelado. Crear una simulación de una supernova también es extremadamente intensivo en recursos, a menudo requiere meses de tiempo de computación en una supercomputadora.

Los algoritmos de aprendizaje automático son excelentes para aprender patrones y tienen una notable capacidad para generalizar a partir de ellos. En esta tesis, aprovechamos esta capacidad generando señales que son algo similares a las simulaciones reales de supernovas. La idea clave es que los algoritmos de aprendizaje automático pueden aprender primero de estos conjuntos de datos aproximados y luego refinar su comprensión utilizando las simulaciones reales. Además, el método de aprendizaje automático aprende que las ondas gravitacionales suelen detectarse en varios detectores simultáneamente.

Los algoritmos de búsqueda tradicionales para agujeros negros de masa intermedia se han refinado durante los últimos 20 años para transformar eficazmente el flujo de datos de los detectores de ondas gravitacionales en características significativas. Sin embargo, estos algoritmos de búsqueda dependen principalmente de las simulaciones. Los algoritmos de aprendizaje automático, por otro lado, pueden aprender directamente de las simulaciones y de los glitches. Sin embargo, sin las características adecuadas, puede ser muy difícil para el aprendizaje automático distinguir entre señales reales y ruido. En esta tesis, desarrollamos una relación simbiótica entre los métodos de búsqueda tradicionales y los algoritmos de aprendizaje automático. Los modelos de aprendizaje automático aprenden las características generadas por los algoritmos de búsqueda tradicionales, lo que les permite distinguir mejor las ondas gravitacionales genuinas del ruido de fondo.

El trabajo en esta tesis tiene como objetivo mejorar la detección de ondas gravitacionales de supernovas y agujeros negros de masa intermedia, al mismo tiempo que se reduce el impacto de fallos. Hemos explorado las fronteras de la detección de ondas gravitacionales transitorias aprovechando el poder del aprendizaje automático.

Curriculum Vitae

Personal Information

Name	Melissa López
Nationality	Argentinian, Italian
Date of birth	18th January 1995
Email	m.lopez@uu.nl, lopezm@nikhef.nl

Education

2020-2024	PhD in Physics Nikhef and GRASP, Utrecht University; Dr. S. E. Caudill
2018-2020	M.Sc. in Data Science for Decision Making Maastricht University; Dr. P. Bonizzi
2014-2018	B.Sc. in Physics University of Valencia; Prof. Dr. J. A. Font Roda

Selected publications

1. T. Dooney et al. "One flexible model for multiclass gravitational wave signal and glitch generation". In: *Physical Review D* 110 (2024). doi: 10.1103/PhysRevD.110.022004.

2. P. Laguarta et al. "Detection of anomalies amongst LIGO's glitch populations with autoencoders". In: *Classical and Quantum Gravity* (2024). doi: 10.1088/1361-6382/ad1f26.

3. Q. Meijer et al. "Gravitational-Wave Searches for Comic String Cusps in Einstein Telescope Data Using Deep Learning". In: *Physical Review D* 109 (2024). doi: 10.1103/PhysRevD.109.022006.

4. M. Lopez et al. "Searching for ring-like structures in the cosmic microwave background". In: *Monthly Notices of the Royal Astronomical Society* 519.1 (2023), pp. 922-936. doi: 10.1093/mnras/stac2610.

5. G. Baltus et al. "Convolutional neural network for gravitational-wave early alert: Going down in frequency". In: *Physical Review D* 106.042002 (2022). doi: 10.1103/PhysRevD.106.042002.

6. M. Lopez et al. "Simulating transient noise burst in LIGO with gengli". In: *IAUGA Proceedings* (2022). arXiv pre-print: 2205.09204.

7. M. Lopez et al. "Simulating transient noise burst in LIGO with Generative Adversarial Networks". In: *Physical Review D* 106.023027 (2022). doi: 10.1103/PhysRevD.106.023027.

8. G. Baltus et al. “Detecting the early inspiral of a gravitational-wave signal with convolutional neural networks”. In: *CBMI Proceedings* (2021). doi: 10.1109/CBMI.2021.9461919.
9. G. Baltus et al. “Convolutional neural networks for the detection of the early inspiral of a gravitational-wave signal”. In: *Physical Review D* 103.102003 (2021). doi: 10.1103/PhysRevD.103.102003.
10. M. Lopez et al. “Deep Learning algorithms for gravitational waves core-collapse supernova detection”. In: *CBMI Proceedings* (2021). doi: 10.1109/CBMI.2021.9461885.
11. M. Lopez et al. “Deep Learning for core-collapse supernova detection”. In: *Physical Review D* 103.063011 (2021). doi: 10.1103/PhysRevD.103.063011.

Bibliography

[1] I. Newton. *Philosophiæ Naturalis Principia Mathematica*. Early English books online. Jussu Societas Regiæ ac typis Josephi Streater, prostant venales apud Sam. Smith, 1687.

[2] A. Einstein. The foundation of the general theory of relativity. *Annalen Phys.*, 49(7):769–822, 1916.

[3] A. Einstein. Näherungsweise Integration der Feldgleichungen der Gravitation. *Sitzungsberichte der K&ouml;niglich Preussischen Akademie der Wissenschaften*, pages 688–696, January 1916.

[4] R. A. Hulse and J. H. Taylor. Discovery of a pulsar in a binary system. *Astrophysical Journal Letters*, 195:L51–L53, 1975.

[5] J. Aasi et al. Advanced LIGO. *Class. Quant. Grav.*, 32:074001, 2015.

[6] F. Acernese et al. Advanced Virgo: a second-generation interferometric gravitational wave detector. *Class. Quant. Grav.*, 32(2):024001, 2015.

[7] B. P. Abbott et al. Observation of Gravitational Waves from a Binary Black Hole Merger. *Phys. Rev. Lett.*, 116(6):061102, 2016.

[8] N. Andersson. *Gravitational-Wave Astronomy: Exploring the Dark Side of the Universe*. Oxford University Press, Oxford, 2019.

[9] M. Maggiore. *Gravitational Waves. Vol. 1: Theory and Experiments*. Oxford University Press, 2007.

[10] T. G. F. Li. *Extracting Physics from Gravitational Waves: Testing the Strong-field Dynamics of General Relativity and Inferring the Large-scale Structure of the Universe*. PhD thesis, Vrije U., Amsterdam, Vrije U., Amsterdam, 2013.

[11] B. P. Abbott et al. GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs. *Phys. Rev. X*, 9(3):031040, 2019.

[12] R. Abbott et al. GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run. *Phys. Rev. X*, 11:021053, 2021.

[13] R. Abbott et al. GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo during the Second Part of the Third Observing Run. *Phys. Rev. X*, 13(4):041039, 2023.

[14] J. Baker et al. The laser interferometer space antenna: Unveiling the millihertz gravitational wave sky. *arXiv pre-print*, 2019.

[15] M. Maggiore et al. Science case for the einstein telescope. *Journal of Cosmology and Astroparticle Physics*, 2020(03):050, mar 2020.

[16] M. Evans et al. A horizon study for cosmic explorer: Science, observatories, and community. *arXiv preprint*, 2021.

[17] R. Abbott et al. Search for intermediate-mass black hole binaries in the third observing run of Advanced LIGO and Advanced Virgo. *Astron. Astrophys.*, 659:A84, 2022.

[18] B. P. Abbott et al. A First Targeted Search for Gravitational-Wave Bursts from Core-Collapse Supernovae in Data of First-Generation Laser Interferometer Detectors. *Phys. Rev. D*, 94(10):102001, 2016.

[19] R. Abbott et al. Constraints on Cosmic Strings Using Data from the Third Advanced LIGO–Virgo Observing Run. *Phys. Rev. Lett.*, 126(24):241102, 2021.

[20] B. P. Abbott et al. All-sky search for short gravitational-wave bursts in the first Advanced LIGO run. *Phys. Rev. D*, 95(4):042003, 2017.

[21] B. P. Abbott et al. All-Sky Search for Short Gravitational-Wave Bursts in the Second Advanced LIGO and Advanced Virgo Run. *Phys. Rev. D*, 100(2):024017, 2019.

[22] R. Abbott et al. All-sky search for short gravitational-wave bursts in the third Advanced LIGO and Advanced Virgo run. *Phys. Rev. D*, 104(12):122004, 2021.

[23] P. D. Lasky. Gravitational Waves from Neutron Stars: A Review. *Publ. Astron. Soc. Austral.*, 32:e034, 2015.

[24] R. Brito et al. Gravitational wave searches for ultralight bosons with LIGO and LISA. *Phys. Rev. D*, 96(6):064050, 2017.

[25] R. Tenorio, D. Keitel, and A. M. Sintes. Search Methods for Continuous Gravitational-Wave Signals from Unknown Sources in the Advanced-Detector Era. *Universe*, 7(12):474, 2021.

[26] N. Christensen. Stochastic Gravitational Wave Backgrounds. *Rept. Prog. Phys.*, 82(1):016903, 2019.

[27] R. Abbott et al. Upper limits on the isotropic gravitational-wave background from Advanced LIGO and Advanced Virgo's third observing run. *Phys. Rev. D*, 104(2):022004, 2021.

[28] J. Weber. Detection and generation of gravitational waves. *Phys. Rev.*, 117:306–313, Jan 1960.

[29] J. Weber. Evidence for discovery of gravitational radiation. *Phys. Rev. Lett.*, 22:1320–1324, Jun 1969.

[30] C. Bond et al. Interferometer techniques for gravitational-wave detection. *Living Reviews in Relativity*, 19(1), 2016.

[31] A. Franklin and R. Laymon. *The Michelson–Morley experiments of 1881 and 1887*. 2053-2571. Morgan and Claypool Publishers, 2019.

[32] B. Willke et al. The GEO 600 gravitational wave detector. *Class. Quant. Grav.*, 19:1377–1387, 2002.

[33] T. Akutsu et al. KAGRA: 2.5 Generation Interferometric Gravitational Wave Detector. *Nature Astron.*, 3(1):35–40, 2019.

[34] B. Iyer et al. Ligo-india, proposal of the consortium for indian initiative in gravitational-wave observations.

[35] C. S. Unnikrishnan. IndIGO and LIGO-India: Scope and plans for gravitational wave research and precision metrology in India. *Int. J. Mod. Phys. D*, 22:1341010, 2013.

[36] I. Gupta et al. Characterizing Gravitational Wave Detector Networks: From $A^\sharp$ to Cosmic Explorer. *arXiv pre-print*, 7 2023.

[37] L. Nuttall et al. Improving the Data Quality of Advanced LIGO Based on Early Engineering Run Results. *Class. Quant. Grav.*, 32(24):245005, 2015.

[38] B. P. Abbott et al. A guide to ligo–virgo detector noise and extraction of transient gravitational-wave signals. *Classical and Quantum Gravity*, 37(5):055002, feb 2020.

[39] D. Davis et al. LIGO detector characterization in the second and third observing runs. *Class. Quant. Grav.*, 38(13):135014, 2021.

[40] F. Acernese et al. Virgo detector characterization and data quality: results from the O3 run. *Class. Quant. Grav.*, 40(18):185006, 2023.

[41] B. P. Abbott et al. Sensitivity of the Advanced LIGO detectors at the beginning of gravitational wave astronomy. *Phys. Rev. D*, 93(11):112004, 2016. [Addendum: Phys.Rev.D 97, 059901 (2018)].

[42] M. H. Gregory et al. Advanced ligo: the next generation of gravitational wave detectors. *Classical and Quantum Gravity*, 27(8):084006, apr 2010.

[43] Alex Amato. *Low Thermal Noise Coating for New Generation Gravitational-Wave Detectors*. PhD thesis, Lyon U., 2019.

[44] C. Talbot and E. Thrane. Gravitational-wave astronomy with an uncertain noise power spectral density. *Phys. Rev. Res.*, 2(4):043298, 2020.

[45] B. Allen et al. FINDCHIRP: An Algorithm for detection of gravitational waves from inspiraling compact binaries. *Phys. Rev. D*, 85:122006, 2012.

[46] T. B. Littenberg and N. J. Cornish. Bayesian inference for spectral estimation of gravitational wave detector noise. *Phys. Rev. D*, 91(8):084034, 2015.

[47] E. Cuoco et al. On-line power spectra identification and whitening for the noise in interferometric gravitational wave detectors. *Classical and Quantum Gravity*, 18(9):1727, may 2001.

[48] C. Biwer et al. Validating gravitational-wave detections: The Advanced LIGO hardware injection system. *Phys. Rev. D*, 95(6):062002, 2017.

[49] G. Vajente et al. Machine-learning nonstationary noise out of gravitational-wave detectors. *Phys. Rev. D*, 101(4):042003, 2020.

[50] B. P. Covas et al. Identification and mitigation of narrow spectral artifacts that degrade searches for persistent gravitational waves in the first two observing runs of Advanced LIGO. *Phys. Rev. D*, 97(8):082002, 2018.

[51] L. Blackburn et al. The LSC Glitch Group: Monitoring Noise Transients during the fifth LIGO Science Run. *Class. Quant. Grav.*, 25:184004, 2008.

[52] B. P. Abbott et al. Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914. *Class. Quant. Grav.*, 33(13):134001, 2016.

[53] S. Soni et al. Reducing scattered light in LIGO's third observing run. *Class. Quant. Grav.*, 38(2):025016, 2020.

[54] M. Cabero et al. Blip glitches in Advanced LIGO data. *Class. Quant. Grav.*, 36(15):15, 2019.

[55] B P Abbott et al. Effects of data quality vetoes on a search for compact binary coalescences in Advanced LIGO's first observing run. *Class. Quant. Grav.*, 35(6):065010, 2018.

[56] R. Abbott et al. All-sky search for continuous gravitational waves from isolated neutron stars using Advanced LIGO and Advanced Virgo O3 data. *Phys. Rev. D*, 106(10):102008, 2022.

[57] B. Steltner et al. Deep Einstein@Home all-sky search for continuous gravitational waves in LIGO O3 public data. *arXiv e-prints*, 3 2023.

[58] B. Steltner, M. A. Papa, and H. B. Eggenstein. Identification and removal of non-Gaussian noise transients for gravitational-wave searches. *Phys. Rev. D*, 105(2):022005, 2022.

[59] C. Pankow et al. Mitigation of the instrumental noise transient in gravitational-wave data surrounding gw170817. *Physical Review D*, 98(8):084016, 2018.

[60] D. Davis et al. Improving the sensitivity of advanced ligo using noise subtraction. *Classical and Quantum Gravity*, 36(5):055011, 2019.

[61] J. C. Driggers et al. Improving astrophysical parameter estimation via offline noise subtraction for Advanced LIGO. *Phys. Rev. D*, 99(4):042001, 2019.

[62] J. Powell. Parameter Estimation and Model Selection of Gravitational Wave Signals Contaminated by Transient Detector Noise Glitches. *Class. Quant. Grav.*, 35(15):155017, 2018.

[63] D. Davis, L. V. White, and P. R Saulson. Utilizing aligo glitch classifications to validate gravitational-wave candidates. *Classical and Quantum Gravity*, 37(14):145001, 2020.

[64] J. Powell et al. Classification methods for noise transients in advanced gravitational-wave detectors. *Classical and Quantum Gravity*, 32(21):215012, November 2015.

[65] M. Zevin et al. Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science. *Class. Quant. Grav.*, 34(6):064003, 2017.

[66] D. George, H. Shen, and E. A. Huerta. Deep Transfer Learning: A new deep learning glitch classification method for advanced LIGO. *arXiv e-prints*, 6 2017.

[67] S. Bahaadini et al. Machine learning for gravity spy: Glitch classification and dataset. *Information Sciences*, 444:172–186, 2018.

[68] J. Glanzer et al. Data quality up to the third observing run of advanced LIGO: Gravity Spy glitch classifications. *Class. Quant. Grav.*, 40(6):065004, 2023.

[69] T. A. Ferreira and C. A. Costa. Comparison between t-SNE and cosine similarity for LIGO glitches analysis. *Class. Quant. Grav.*, 39(16):165013, 2022.

[70] M. Razzano et al. Gwitchhunters: Machine learning and citizen science to improve the performance of gravitational wave detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1048:167959, 2023.

[71] S. Alvarez-Lopez et al. Gspynettree: A signal-vs-glitch classifier for gravitational-wave event candidates. *arXiv preprint arXiv:2304.09977*, 2023.

[72] J. C. Brown. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

[73] F. Robinet et al. Omicron: a tool to characterize transient noise in gravitational-wave detectors. *SoftwareX*, 12:100620, 2020.

[74] P. J. Jung et al. Sensing and vetoing loud transient noises for the gravitational-wave detection. *J. Korean Phys. Soc.*, 73:1197, 2018.

[75] S. Soni et al. Discovering features in gravitational-wave data through detector characterization, citizen science and machine learning. *Class. Quant. Grav.*, 38(19):195016, 2021.

[76] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing.* Prentice-hall Englewood Cliffs, second edition, 1999.

[77] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, jan 1949.

[78] E. Cuoco et al. Noise parametric identification and whitening for LIGO 40-meter interferometer data. *Phys. Rev. D*, 64:122002, 2001.

[79] J. R. Smith et al. A Hierarchical method for vetoing noise transients in gravitational-wave detectors. *Class. Quant. Grav.*, 28:235005, 2011.

[80] R. Abbott et al. Open data from the first and second observing runs of Advanced LIGO and Advanced Virgo. *SoftwareX*, 13:100658, 2021.

[81] R. Abbott et al. Open Data from the Third Observing Run of LIGO, Virgo, KAGRA, and GEO. *Astrophys. J. Suppl.*, 267(2):29, 2023.

[82] L. Blanchet et al. Gravitational-radiation damping of compact binary systems to second post-newtonian order. *Phys. Rev. Lett.*, 74:3515–3518, May 1995.

[83] A. Buonanno and T. Damour. Effective one-body approach to general relativistic two-body dynamics. *Phys. Rev. D*, 59:084006, 1999.

[84] D. G. Keppel. *Signatures and dynamics of compact binary coalescences and a search in LIGO's S5 data.* PhD thesis, California Institute of Technology, January 2009.

[85] M. Mateu Lucena. *Understanding gravitational wave data with Bayesian inference and machine learning.* PhD thesis, University of Balearic Islands. Dept. of Physics., 2023.

[86] C. M. Will and A. G. Wiseman. Gravitational radiation from compact binary systems: Gravitational waveforms and energy loss to second post-newtonian order. *Phys. Rev. D*, 54(8):4813–4848, Oct 1996.

[87] M. E. Pati and C. M. Will. Post-newtonian gravitational radiation and equations of motion via direct integration of the relaxed einstein equations: Foundations. *Phys. Rev. D*, 62(12):124015, Nov 2000.

[88] M. E. Pati and C. M. Will. Post-newtonian gravitational radiation and equations of motion via direct integration of the relaxed einstein equations. ii. two-body equations of motion to second post-newtonian order, and radiation reaction to 3.5 post-newtonian order. *Phys. Rev. D*, 65(10):104008, Apr 2002.

[89] L. Blanchet. Gravitational radiation from post-newtonian sources and inspiralling compact binaries. *Living Reviews in Relativity*, 17(2):1–187, 2014.

[90] M. Levi. Effective Field Theories of Post-Newtonian Gravity: A comprehensive review. *Rept. Prog. Phys.*, 83(7):075901, 2020.

[91] G. Schäfer and P. Jaranowski. Hamiltonian formulation of general relativity and post-newtonian dynamics of compact binaries. *Living Reviews in Relativity*, 21(7):1–179, Aug 2018.

[92] A. Nagar et al. Multipolar effective one body waveform model for spin-aligned black hole binaries. *Phys. Rev. D*, 102(2):024077, 2020.

[93] S. Ossokine et al. Multipolar Effective-One-Body Waveforms for Precessing Binary Black Holes: Construction and Validation. *Phys. Rev. D*, 102(4):044055, 2020.

[94] Alessandro Nagar, Alice Bonino, and Piero Rettegno. Effective one-body multipolar waveform model for spin-aligned, quasicircular, eccentric, hyperbolic black hole binaries. *Phys. Rev. D*, 103(10):104021, 2021.

[95] A. Ramos-Buades et al. Next generation of accurate and efficient multipolar precessing-spin effective-one-body waveforms for binary black holes. *Phys. Rev. D*, 108(12):124037, 2023.

[96] A. Bohé et al. Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors. *Phys. Rev. D*, 95(4):044028, Feb 2017.

[97] R. Cotesta et al. Enriching the symphony of gravitational waves from binary black holes by tuning higher harmonics. *Phys. Rev. D*, 98(8):084028, Oct 2018.

[98] S. Khan et al. Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era. *Phys. Rev. D*, 93(4):044007, 2016.

[99] H. Estellés et al. Phenomenological time domain model for dominant quadrupole gravitational wave signal of coalescing binary black holes. *Phys. Rev. D*, 103(12):124060, 2021.

[100] H. Estellés et al. New twists in compact binary waveform modeling: A fast time-domain model for precession. *Phys. Rev. D*, 105(8):084040, 2022.

[101] G. Pratten et al. Computationally efficient models for the dominant and subdominant harmonic modes of precessing binary black holes. *Phys. Rev. D*, 103(10):104056, 2021.

[102] Duncan A. Brown. *Searching for gravitational radiation from binary black hole MACHOs in the galactic halo.* Other thesis, University of Wisconsin-Milwaukee, 12 2004.

[103] C. Capano. *Searching for Gravitational Waves from Compact Binary Coalescence Using LIGO and Virgo Data.* PhD thesis, Syracuse University, December 2011.

[104] I. Harry et al. Searching for Gravitational Waves from Compact Binaries with Precessing Spins. *Phys. Rev. D*, 94(2):024012, 2016.

[105] S. Schmidt, B. Gadre, and S. Caudill. Gravitational-wave template banks for novel compact binaries. *arXiv pre-print*, 2 2023.

[106] S. Sachdev et al. The GstLAL Search Analysis Methods for Compact Binary Mergers in Advanced LIGO's Second and Advanced Virgo's First Observing Runs. *arXiv pre-print*, 1 2019.

[107] I. W. Harry, B. Allen, and B. S. Sathyaprakash. A Stochastic template placement algorithm for gravitational wave data analysis. *Phys. Rev. D*, 80:104014, 2009.

[108] P. Ajith et al. Effectual template bank for the detection of gravitational waves from inspiralling compact binaries with generic spins. *Phys. Rev. D*, 89(8):084041, 2014.

[109] S. Roy, A. S. Sengupta, and P. Ajith. Effectual template banks for upcoming compact binary searches in Advanced-LIGO and Virgo data. *Phys. Rev. D*, 99(2):024048, 2019.

[110] C. Messick et al. Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data. *Physical Review D*, 95(4):042001, February 2017.

[111] C. Hanna et al. Fast evaluation of multidetector consistency for real-time gravitational wave searches. *Phys. Rev. D*, 101(2):022003, 2020.

[112] K. Cannon et al. Gstlal: A software framework for gravitational wave discovery. *SoftwareX*, 14:100680, 2021.

[113] B. Allen. χ^2 time-frequency discriminator for gravitational wave detection. *Physical Review D*, 71(6):062001, 2005.

[114] T. Dal Canton et al. Implementing a search for aligned-spin neutron star-black hole systems with advanced ground based gravitational wave detectors. *Physical Review D*, 90(8):082004, 2014.

[115] S. A. Usman et al. The pycbc search for gravitational waves from compact binary coalescence. *Classical and Quantum Gravity*, 33(21):215004, 2016.

[116] A. H. Nitz et al. Detecting binary compact-object mergers with gravitational waves: Understanding and improving the sensitivity of the pycbc search. *The Astrophysical Journal*, 849(2):118, 2017.

[117] G. S. Davies et al. Extending the pycbc search for gravitational waves from compact binary mergers to a global network. *Physical Review D*, 102(2):022004, 2020.

[118] A. H. Nitz et al. Rapid detection of gravitational waves from compact binary mergers with pycbc live. *Physical Review D*, 98(2):024050, 2018.

[119] T. Adams et al. Low-latency analysis pipeline for compact binary coalescences in the advanced gravitational wave detector era. *Classical and Quantum Gravity*, 33(17):175012, 2016.

[120] F. Aubin et al. The mbta pipeline for detecting compact binary coalescences in the third ligo–virgo observing run. *Classical and Quantum Gravity*, 38(9):095004, 2021.

[121] Q. Chu et al. Spiir online coherent pipeline to search for gravitational waves from compact binary coalescences. *Physical Review D*, 105(2):024023, 2022.

[122] P. Godwin. *Low-latency Statistical Data Quality in the Era of Multi-Messenger Astronomy*. PhD thesis, Penn State U., 2020.

[123] K. Cannon et al. Singular value decomposition applied to compact binary coalescence gravitational-wave signals. *Phys. Rev. D*, 82:044025, 2010.

[124] Andre Guimaraes. *Bridging the Gap Between Detector Characterization and Gravitational Wave Searches in LIGO*. Phd thesis, Louisiana State University, July 2023.

[125] K. Cannon, C. Hanna, and J. Peoples. Likelihood-Ratio Ranking Statistic for Compact Binary Coalescence Candidates with Rate Estimation. *arXiv e-prints*, 4 2015.

[126] L. Tsukada et al. Improved ranking statistics of the GstLAL inspiral search for compact binary coalescences. *Phys. Rev. D*, 108(4):043004, 2023.

[127] R. Lynch et al. Information-theoretic approach to the gravitational-wave burst detection problem. *Phys. Rev. D*, 95(10):104046, 2017.

[128] M. Drago et al. coherent waveburst, a pipeline for unmodeled gravitational-wave data analysis. *SoftwareX*, 14:100678, 2021.

[129] M. Drago. *Search for transient gravitational wave signals with unknown waveform in the LIGO-Virgo network of interferometric detectors using a fully coherent algorithm*. PhD thesis, University of Padua. Dept. of Physics., 2010.

[130] S. Chatterji et al. Multiresolution techniques for the detection of gravitational-wave bursts. *Class. Quant. Grav.*, 21:S1809–S1818, 2004.

[131] A. Graps. "an introduction to wavelets". *IEEE Comp. Sci. Engi.*, 2:50–61, 02 1995.

[132] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41:909–996, 1988.

[133] S. Roy. Nonorthogonal wavelet transformation for reconstructing gravitational wave signals. *Phys. Rev. Res.*, 4(3):033078, 2022.

[134] S. Tiwari. *Gravitational waves from eccentric binary black hole mergers: searches and astrophysical interpretations*. PhD thesis, Gran Sasso Science Institute, 2018.

[135] N. J. Cornish and T. B. Littenberg. BayesWave: Bayesian Inference for Gravitational Wave Bursts and Instrument Glitches. *Class. Quant. Grav.*, 32(13):135012, 2015.

[136] V. Necula, S. Klimenko, and G. Mitselmakher. Transient analysis with fast wilson-daubechies time-frequency transform. *Journal of Physics: Conference Series*, 363(1):012032, jun 2012.

[137] R. Brun and F. Rademakers. ROOT: An object oriented data analysis framework. *Nucl. Instrum. Meth. A*, 389:81–86, 1997.

[138] V. Tiwari et al. Regression of Environmental Noise in LIGO Data. *Class. Quant. Grav.*, 32(16):165014, 2015.

[139] S. Klimenko et al. coherent waveburst documentation. `https://gwburst.gitlab.io/documentation/latest/html/documents.html`. Accessed: 25-11-2023.

[140] I. Yakusin S. Klimenko and G. Mitselmakher. Waveburst - s5 version. `https://dcc.ligo.org/public/0027/T060112/000/T060112-00.pdf`. Accessed: 22-11-2023.

[141] T. Berners-Lee et al. The World-Wide Web Initiative, 1993.

[142] S. Kaushal and M. Dr. A review on big data: Applications and challenges. *Journal of Emerging Technologies and Innovative Research*, 01 2019.

[143] Shreyasi Acharya et al. The ALICE experiment: a journey through QCD. *Eur. Phys. J. C*, 84(8):813, 2024.

[144] J. P. Gardner et al. The james webb space telescope mission. *Publications of the Astronomical Society of the Pacific*, 135(1048):068001, jun 2023.

[145] B. Abi et al. Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume I Introduction to DUNE. *JINST*, 15(08):T08008, 2020.

[146] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44(1.2):206–226, 2000.

[147] M. Shehab et al. Machine learning in medical applications: A review of state-of-the-art methods. *Computers in Biology and Medicine*, 145:105458, 2022.

[148] P. Gogas and T. Papadimitriou. Machine learning in economics and finance. *Computational Economics*, 57, 02 2021.

[149] N. Srivastava et al. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[150] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[151] M. Soori, B. Arezoo, and R. Dastres. Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, 3:54–70, 2023.

[152] A. Y. Ng. *Shaping and polocy search in Reinforcement learning.* PhD thesis, University of California, Berkeley. Dept. of Computer Science., 2010.

[153] O. Chapelle, B. Schlkopf, and A. Zien. *Semi-Supervised Learning.* The MIT Press, 1st edition, 2010.

[154] P. Ren et al. A Survey of Deep Active Learning. *arXiv e-prints*, page arXiv:2009.00236, August 2020.

[155] R. Rassin, S. Ravfogel, and Y. Goldberg. DALLE-2 is Seeing Double: Flaws in Word-to-Concept Mapping in Text2Image Models. *arXiv e-prints*, page arXiv:2210.10606, October 2022.

[156] G. Abhishek et al. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array*, 10:100057, 2021.

[157] T. Iqbal and S. Qureshi. The survey: Text generation models in deep learning. *Journal of King Saud University - Computer and Information Sciences*, 34(6, Part A):2515–2528, 2022.

[158] E. Cetinic and J. She. Understanding and Creating Art with AI: Review and Outlook. *arXiv e-prints*, page arXiv:2102.09109, February 2021.

[159] F. Li. Cs231n: Convolutional neural networks for visual recognition, October 2018.

[160] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[161] P. Le and W. Zuidema. Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs. *arXiv e-prints*, page arXiv:1603.00423, March 2016.

[162] A. van den Oord, N.Kalchbrenner, and K. Kavukcuoglu. Pixel Recurrent Neural Networks. *arXiv e-prints*, page arXiv:1601.06759, January 2016.

[163] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv e-prints*, page arXiv:1609.04747, September 2016.

[164] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and D. Mike Titterington, editors, *AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.

[165] A. Gunes Baydin et al. Automatic differentiation in machine learning: a survey. *arXiv e-prints*, page arXiv:1502.05767, February 2015.

[166] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv e-prints*, page arXiv:1603.07285, March 2016.

[167] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[168] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1), 2021.

[169] B. Sathyaprakash et al. Scientific Objectives of Einstein Telescope. *Class. Quant. Grav.*, 29:124013, 2012. [Erratum: Class.Quant.Grav. 30, 079501 (2013)].

[170] P. Amaro-Seoane et al. Laser Interferometer Space Antenna. *arXiv e-prints*, page arXiv:1702.00786, February 2017.

[171] T. Regimbau and S. A. Hughes. Gravitational-wave confusion background from cosmological compact binaries: Implications for future terrestrial detectors. *Phys. Rev. D*, 79:062002, 2009.

[172] A. Samajdar et al. Biases in parameter estimation from overlapping gravitational-wave signals in the third-generation detector era. *Phys. Rev. D*, 104(4):044003, 2021.

[173] S. Wu and A. H. Nitz. Mock data study for next-generation ground-based detectors: The performance loss of matched filtering due to correlated confusion noise. *Phys. Rev. D*, 107(6):063022, 2023.

[174] M. van der Sluys et al. Parameter estimation of spinning binary inspirals using Markov-chain Monte Carlo. *Class. Quant. Grav.*, 25:184011, 2008.

[175] J. Veitch et al. Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library. *Phys. Rev. D*, 91(4):042003, 2015.

[176] E. Cuoco et al. Enhancing Gravitational-Wave Science with Machine Learning. *Mach. Learn. Sci. Tech.*, 2(1):011002, 2021.

[177] T. Zhao et al. Dawning of a New Era in Gravitational Wave Data Analysis: Unveiling Cosmic Mysteries via Artificial Intelligence – A Systematic Review. *arXiv preprint*, 11 2023.

[178] R. Ormiston et al. Noise Reduction in Gravitational-wave Data via Deep Learning. *Phys. Rev. Res.*, 2(3):033066, 2020.

[179] M. Saleem et al. Demonstration of Machine Learning-assisted real-time noise regression in gravitational wave detectors. *arXiv pre-print*, 6 2023.

[180] R. Essick et al. iDQ: Statistical Inference of Non-Gaussian Noise with Auxiliary Degrees of Freedom in Gravitational-Wave Detectors. *arXiv pre-print*, 5 2020.

[181] B. P. Abbott et al. GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral. *Phys. Rev. Lett.*, 119(16):161101, 2017.

[182] M. Razzano and E. Cuoco. Image-based deep learning for classification of noise transients in gravitational wave detectors. *Class. Quant. Grav.*, 35(9):095016, 2018.

[183] Y. Li, Y. Wu, and A. K. Katsaggelos. Cross-Temporal Spectrogram Autoencoder (CTSAE): Unsupervised Dimensionality Reduction for Clustering Gravitational Wave Glitches. *arXiv pre-print*, 4 2024.

[184] E. A. Huerta et al. Eccentric, nonspinning, inspiral, Gaussian-process merger approximant for the detection and characterization of eccentric binary black hole mergers. *Phys. Rev. D*, 97(2):024031, 2018.

[185] B.D. Lackey et al. Effective-one-body waveforms for binary neutron stars using surrogate models. *Phys. Rev. D*, 95(10):104036, 2017.

[186] B. D. Lackey et al. Surrogate model for an aligned-spin effective one body waveform model of binary neutron star inspirals using Gaussian process regression. *Phys. Rev. D*, 100(2):024002, 2019.

[187] S. Khan and R. Green. Gravitational-wave surrogate models powered by artificial neural networks. *Phys. Rev. D*, 103(6):064015, 2021.

[188] L. M. Thomas, G. Pratten, and P. Schmidt. Accelerating multimodal gravitational waveforms from precessing compact binaries with artificial neural networks. *arXiv pre-print*, 5 2022.

[189] S-C. Fragkouli et al. Deep Residual Error and Bag-of-Tricks Learning for Gravitational Wave Surrogate Modeling. *arXiv pre-print*, 3 2022.

[190] B. Keith, A. Khadse, and S. E. Field. Learning orbital dynamics of binary black hole systems from gravitational wave measurements. *Phys. Rev. Res.*, 3(4):043101, 2021.

[191] J. Tissino et al. Combining effective-one-body accuracy and reduced-order-quadrature speed for binary neutron star merger parameter estimation with machine learning. *Phys. Rev. D*, 107(8):084037, 2023.

[192] S. Schmidt et al. Machine Learning Gravitational Waves from Binary Black Hole Mergers. *Phys. Rev. D*, 103(4):043020, 2021.

[193] A. J. K. Chua, C. R. Galley, and M. Vallisneri. Reduced-order modeling with artificial neurons for gravitational-wave inference. *Phys. Rev. Lett.*, 122(21):211101, 2019.

[194] J. Lee et al. Deep learning model on gravitational waveforms in merging and ringdown phases of binary black hole coalescences. *Phys. Rev. D*, 103(12):123023, 2021.

[195] A. J. K. Chua et al. Rapid generation of fully relativistic extreme-mass-ratio-inspiral waveform templates for LISA data analysis. *Phys. Rev. Lett.*, 126(5):051102, 2021.

[196] P. Nousi et al. Autoencoder-driven Spiral Representation Learning for Gravitational Wave Surrogate Modelling. *arXiv pre-print*, 7 2021.

[197] C-H. Liao and F-L. Lin. Deep generative models of gravitational waveforms via conditional autoencoder. *Phys. Rev. D*, 103(12):124051, 2021.

[198] A. Khan, E. A. Huerta, and H. Zheng. Interpretable AI forecasting for numerical relativity waveforms of quasicircular, spinning, nonprecessing binary black hole mergers. *Phys. Rev. D*, 105(2):024024, 2022.

[199] F. F. Freitas et al. Generating gravitational waveform libraries of exotic compact binaries with deep learning. *arXiv pre-print*, 3 2022.

[200] D. George and E. A. Huerta. Deep Neural Networks to Enable Real-time Multimessenger Astrophysics. *Phys. Rev. D*, 97(4):044039, 2018.

[201] H. Gabbard et al. Matching matched filtering with deep networks for gravitational-wave astronomy. *Phys. Rev. Lett.*, 120(14):141103, 2018.

[202] K. Sharma, K. Chandra, and A. Pai. Fishing massive black hole binaries with THAMES. *arXiv pre-print*, 8 2022.

[203] P. G. Krastev. Real-Time Detection of Gravitational Waves from Binary Neutron Stars using Artificial Neural Networks. *Phys. Lett. B*, 803:135330, 2020.

[204] A. Menéndez-Vázquez et al. Searches for compact binary coalescence events using neural networks in the LIGO/Virgo second observation period. *Phys. Rev. D*, 103(6):062004, 2021.

[205] G. Baltus et al. Convolutional neural networks for the detection of the early inspiral of a gravitational-wave signal. *Phys. Rev. D*, 103:102003, 2021.

[206] G. Baltus et al. Convolutional neural network for gravitational-wave early alert: Going down in frequency. *Phys. Rev. D*, 106(4):042002, 2022.

[207] J. Yan et al. Generalized approach to matched filtering using neural networks. *Phys. Rev. D*, 105:043006, Feb 2022.

[208] T. Mishra et al. Optimization of model independent gravitational wave search for binary black hole mergers using machine learning. *Phys. Rev. D*, 104(2):023014, 2021.

[209] M. J. Szczepańczyk et al. Search for gravitational-wave bursts in the third Advanced LIGO-Virgo run with coherent WaveBurst enhanced by machine learning. *Phys. Rev. D*, 107(6):062002, 2023.

[210] D. Lopez et al. Utilizing Gaussian mixture models in all-sky searches for short-duration gravitational wave bursts. *Phys. Rev. D*, 105(6):063024, 2022.

[211] S. Bini et al. An autoencoder neural network integrated into gravitational-wave burst searches to improve the rejection of noise transients. *Class. Quant. Grav.*, 40(13):135008, 2023.

[212] M. Cavaglia et al. Improving the background of gravitational-wave searches for core collapse supernovae: A machine learning approach. *arXiv e-prints*, page arXiv:2002.04591, February 2020.

[213] J. M. Antelis et al. Using supervised learning algorithms as a follow-up method in the search of gravitational waves from core-collapse supernovae. *Phys. Rev. D*, 105:084054, Apr 2022.

[214] A. Iess et al. Core-Collapse Supernova Gravitational-Wave Search and Deep Learning Classification. *arXiv pre-print*, 1 2020.

[215] A. Iess et al. LSTM and CNN application for core-collapse supernova search in gravitational wave real data. *Astron. Astrophys.*, 669:A42, 2023.

[216] Q. Meijer et al. Gravitational-Wave Searches for Cosmic String Cusps in Einstein Telescope Data using Deep Learning. *arXiv pre-print*, 8 2023.

[217] V. Boudart and M. Fays. Machine learning algorithm for minute-long burst searches. *Phys. Rev. D*, 105(8):083007, 2022.

[218] V. Boudart. Convolutional neural network to distinguish glitches from minute-long gravitational wave transients. *Phys. Rev. D*, 107(2):024007, 2023.

[219] V. Skliris, M. R. K. Norman, and P. J. Sutton. Real-Time Detection of Unmodelled Gravitational-Wave Transients Using Convolutional Neural Networks. *arXiv pre-print*, 9 2020.

[220] R. Raikman et al. GWAK: Gravitational-Wave Anomalous Knowledge with Recurrent Autoencoders. *arXiv pre-print*, 9 2023.

[221] F. Morawski et al. Anomaly detection in gravitational waves data using convolutional autoencoders. *Mach. Learn. Sci. Tech.*, 2(4):045014, 2021.

[222] J. Bayley, C. Messenger, and G. Woan. Robust machine learning algorithm to search for continuous gravitational waves. *Phys. Rev. D*, 102(8):083024, 2020.

[223] L. M. Modafferi, Rodrigo R. Tenorio, and D. Keitel. Convolutional neural network search for long-duration transient gravitational waves from glitching pulsars. *Phys. Rev. D*, 108(2):023005, 2023.

[224] P. M. Joshi and R. Prix. Novel neural-network architecture for continuous gravitational waves. *Phys. Rev. D*, 108(6):063021, 2023.

[225] J. Bayley, C. Messenger, and G. Woan. Generalized application of the Viterbi algorithm to searches for continuous gravitational-wave signals. *Physical Review D*, 100(2):023006, July 2019.

[226] A. Utina et al. Deep learning searches for gravitational wave stochastic backgrounds. In *International Conference on Content-Based Multimedia Indexing*, 6 2021.

[227] O. G. Freitas et al. Comparison of neural network architectures for feature extraction from binary black hole merger waveforms. *arXiv pre-print*, 7 2023.

[228] M. Andrés-Carcasona, M. Martinez, and Ll. M. Mir. Fast Bayesian gravitational wave parameter estimation using convolutional neural networks. *Mon. Not. Roy. Astron. Soc.*, 527(2):2887–2894, 2023.

[229] H. Gabbard et al. Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy. *Nature Phys.*, 18(1):112–117, 2022.

[230] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Phil. Trans. of the Royal Soc. of London*, 53:370–418, 1763.

[231] S. R. Green, C. Simpson, and J. Gair. Gravitational-wave parameter estimation with autoregressive neural network flows. *Phys. Rev. D*, 102(10):104057, 2020.

[232] S. R. Green and J. Gair. Complete parameter inference for GW150914 using deep learning. *Mach. Learn. Sci. Tech.*, 2(3):03LT01, 2021.

[233] M. J. Williams, J. Veitch, and C. Messenger. Nested sampling with normalizing flows for gravitational-wave inference. *Phys. Rev. D*, 103(10):103006, 2021.

[234] M. J. Williams, J. Veitch, and C. Messenger. Importance nested sampling with normalising flows. *Mach. Learn. Sci. Tech.*, 4(3):035011, 2023.

[235] A. Kolmus et al. Fast sky localization of gravitational waves using deep learning seeded importance sampling. *Physical Review D*, 106, 07 2022.

[236] J. Langendorff et al. Normalizing flows as an avenue to studying overlapping gravitational wave signals. *Phys. Rev. Lett.*, 130:171402, Apr 2023.

[237] A. Cole et al. Fast and credible likelihood-free cosmology with truncated marginal neural ratio estimation. *JCAP*, 09:004, 2022.

[238] J. Alvey et al. Simulation-based inference for stochastic gravitational wave background data analysis. *arXiv pre-print*, 9 2023.

[239] J. Terven et al. Loss Functions and Metrics in Deep Learning. *arXiv e-prints*, page arXiv:2307.02694, July 2023.

[240] J. Yosinski et al. Understanding Neural Networks Through Deep Visualization. *arXiv e-prints*, page arXiv:1506.06579, June 2015.

[241] H.A. Bethe. Supernova mechanisms. *Rev. Mod. Phys.*, 62:801–866, Oct 1990.

[242] H.T. Janka. *Neutrino Emission from Supernovae.* Springer International Publishing, Cham, 2017.

[243] K. Sato and H. Suzuki. Analysis of neutrino burst from the supernova 1987a in the large magellanic cloud. *Phys. Rev. Lett.*, 58:2722–2725, Jun 1987.

[244] S. E. Woosley et al. Pulsational pair instability as an explanation for the most luminous supernovae. *Nature*, 450:390, 2007.

[245] S. E. Woosley and Alexander Heger. The Pair-Instability Mass Gap for Black Holes. *Astrophys. J. Lett.*, 912(2):L31, 2021.

[246] D. Richstone et al. Supermassive black holes and the evolution of galaxies. *Nature*, 395:A14–A19, 1998.

[247] Event Horizon Telescope Collaboration. First sagittarius a* event horizon telescope results. i. the shadow of the supermassive black hole in the center of the milky way. *The Astrophysical Journal Letters*, 930(2):L12, may 2022.

[248] J. E. Greene, J. Strader, and L. C. Ho. Intermediate-Mass Black Holes. *Annual Review of Astronomy and Astrophysics*, 58:257–312, August 2020.

[249] M. Lopez et al. Simulating transient noise bursts in LIGO with generative adversarial networks. *Phys. Rev. D*, 106(2):023027, 2022.

[250] M. Lopez et al. Simulating Transient Noise Bursts in LIGO with gengli. *arXiv pre-print*, 5 2022.

[251] I. J. Goodfellow et al. Generative adversarial nets, 2014.

[252] R. Obaid S. Mukherjee and B. Matkarimov. Classification of glitch waveforms in gravitational wave detector characterization. *Journal of Physics: Conference Series*, 243:012006, 09 2010.

[253] J. Powell et al. Classification methods for noise transients in advanced gravitational-wave detectors II: performance tests on Advanced LIGO data. *Classical and Quantum Gravity*, 34(3):034002, February 2017.

[254] J. P. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 12 1995.

[255] A. Torres-Forné et al. Total-variation-based methods for gravitational wave denoising. *Phys. Rev. D*, 90:084029, Oct 2014.

[256] J. McGinn et al. Generalised gravitational wave burst generation with generative adversarial networks. *Class. Quant. Grav.*, 38(15):155005, 2021.

[257] S. Harada et al. Biosignal generation and latent variable analysis with recurrent generative adversarial networks. *IEEE Access*, 7:144292–144302, 2019.

[258] L. Metz A. Radford and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[259] L. Weng. From gan to wgan. *ArXiv*, abs/1904.08994, 2019.

[260] M. Arjovsky et al. Wasserstein generative adversarial networks, 06–11 Aug 2017.

[261] L. V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, jul 1960.

[262] T. Karras et al. Progressive growing of GANs for improved quality, stability, and variation, 2018.

[263] T. Salimans et al. Improved techniques for training gans, 2016.

[264] A. Geiger L. Mescheder and S. Nowozin. Which training methods for GANs do actually converge?, 10–15 Jul 2018.

[265] X. Wei et al. Improving the improved training of wasserstein gans: A consistency term and its dual effect, 2018.

[266] I. Gulrajani et al. Improved training of wasserstein gans, 2017.

[267] N. Kodali et al. How to train your dragan. *ArXiv*, abs/1705.07215, 2017.

[268] T. Tieleman et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

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

[270] A. Nitz et al. gwastro/pycbc, April 2021.

[271] R. E. Walpole et al. *Probability and Statistics for Engineers and Scientists.* Pearson Education, 2007.

[272] T. Karras et al. Training generative adversarial networks with limited data. *ArXiv*, abs/2006.06676, 2020.

[273] G. Ashton et al. Parameterised population models of transient non-Gaussian noise in the LIGO gravitational-wave detectors. *arXiv e-prints*, page arXiv:2110.02689, October 2021.

[274] I. Jolliffe and J. Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374:20150202, 04 2016.

[275] F. Pedregosa et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[276] D. Reynolds. *Gaussian Mixture Models.* Springer US, Boston, MA, 2009.

[277] Y. Shen et al. Interpreting the latent space of gans for semantic face editing. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9240–9249, 2020.

[278] C. F. Van Loan and G. Golub. Matrix computations (johns hopkins studies in mathematical sciences). *Matrix Computations*, 1996.

[279] S. Kulkarni et al. Random projections in gravitational wave searches of compact binaries. *Physical Review D*, 99(10):101503, 2019.

[280] A. Reza et al. Random projections in gravitational-wave searches from compact binaries ii: efficient reconstruction of the detection statistic. *arXiv:2101.03226*, 2021.

[281] D. Shoemaker et al. Advanced ligo anticipated sensitivity curves. *LIGO Document LIGO-T0900288-v3*, 2010.

[282] J. Powell et al. Generating transient noise artefacts in gravitational-wave detector data with generative adversarial networks. *Class. Quant. Grav.*, 40(3):035006, 2023.

[283] J. Yan, A.P. Leung, and D. C. Y. Hui. On improving the performance of glitch classification for gravitational wave detection by using Generative Adversarial Networks. *Mon. Not. Roy. Astron. Soc.*, 515(3):4606–4621, 2022.

[284] T. Dooney, S. Bromuri, and L. Curier. Dvgan: Stabilize wasserstein gan training for time-domain gravitational wave physics. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 5468–5477, Los Alamitos, CA, USA, dec 2022. IEEE Computer Society.

[285] T. Dooney et al. One flexible model for multiclass gravitational wave signal and glitch generation. *Phys. Rev. D*, 110(2):022004, 2024.

[286] P. Laguarta et al. Detection of anomalies amongst LIGO's glitch populations with autoencoders. *arXiv pre-print*, 10 2023.

[287] R. E. Colgan et al. Efficient Gravitational-wave Glitch Identification from Environmental Data Through Machine Learning. *Phys. Rev. D*, 101(10):102003, 2020.

[288] M. Cavaglia. Characterization of gravitational-wave detector noise with fractals. *Class. Quant. Grav.*, 39(13):135012, 2022.

[289] J. Theiler. Estimating fractal dimension. *Journal of the Optical Society of America A*, 7(6):1055–1073, 1990.

[290] T. Gneiting, H. Ševčíková, and D. B. Percival. Estimators of fractal dimension: Assessing the roughness of time series and spatial data. *Statistical Science*, pages 247–277, 2012.

[291] C. Tricot. Two definitions of fractional dimension. *Mathematical Proceedings of the Cambridge Philosophical Society*, 91(1):57–74, 1982.

[292] M. Fernández-Martínez and M.A. Sánchez-Granero. Fractal dimension for fractal structures. *Topology and its Applications*, 163:93–111, 2014.

[293] B. Dubuc and S. Dubuc. Error bounds on the estimation of fractal dimension. *SIAM Journal on Numerical Analysis*, 33:602–626, 1996.

[294] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.

[295] D. Bank, N. Koenigstein, and R. Giryes. Autoencoders. *arXiv preprint arXiv:2003.05991*, 2020.

[296] K. O'Shea and R. Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[297] X. Guo et al. Deep clustering with convolutional autoencoders. In *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II 24*, pages 373–382. Springer, 2017.

[298] Y. Zhang. A better autoencoder for image: Convolutional autoencoder. In *ICONIP17-DCEC. Available online: `http://users.cecs.anu.edu.au/Tom.Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58.pdf`*, 2018.

[299] R. Priemer. *Introductory signal processing*, volume 6. World scientific, 1991.

[300] M. Abadi et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[301] F. Chollet et al. Keras, 2015.

[302] A. F. Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[303] R. Arora et al. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

[304] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[305] T. Kim et al. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *arXiv preprint arXiv:2105.08919*, 2021.

[306] T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae). *Geoscientific model development discussions*, 7(1):1525–1534, 2014.

[307] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[308] M. López Portilla et al. Deep learning for core-collapse supernova detection. *Phys. Rev. D*, 103(6):063011, 2021.

[309] M. López et al. Deep learning algorithms for gravitational waves core-collapse supernova detection. In *International Conference on Content-Based Multimedia Indexing*, 6 2021.

[310] H. Th. Janka. Neutrino-driven Explosions. *arXiv pre-prints*, 2 2017.

[311] W. Li et al. Nearby supernova rates from the Lick Observatory Supernova Search - II. The observed luminosity functions and fractions of supernovae in a complete sample. *Monthly Notices of the Royal Astronomical Society*, 412(3):1441–1472, April 2011.

[312] R. Chapman et al. How common are long gamma-ray bursts in the local Universe? *Monthly Notices of the Royal Astronomical Society*, 382(1):L21–L25, November 2007.

[313] B. P. Abbott et al. First targeted search for gravitational-wave bursts from core-collapse supernovae in data of first-generation laser interferometer detectors. *Physical Review D*, 94(10):102001, November 2016.

[314] B. P. Abbott et al. Optically targeted search for gravitational waves emitted by core-collapse supernovae during the first and second observing runs of advanced LIGO and advanced Virgo. *Phys. Rev. D*, 101(8):084002, 2020.

[315] S. Klimenko et al. Method for detection and reconstruction of gravitational wave transients with networks of advanced detectors. *Phys. Rev. D*, 93(4):042004, 2016.

[316] P. Astone et al. A new method to observe gravitational waves emitted by core collapse supernovae. *Physical Review D*, 98, 12 2018.

[317] I. S. Heng M. L. Chan and C. Messenger. Detection and Classification of Supernova Gravitational Waves Signals: A Deep Learning Approach. *arXiv e-prints*, page arXiv:1912.13517, December 2019.

[318] M. C. Edwards. Classifying the equation of state from rotating core collapse gravitational waves with deep learning. *Physical Review D*, 103(2):024025, January 2021.

[319] A. Torres-Forné et al. Towards asteroseismology of core-collapse supernovae with gravitational wave observations – II. Inclusion of space–time perturbations. *Mon. Not. Roy. Astron. Soc.*, 482(3):3967–3988, 2019.

[320] B. Müller, H.-T. Janka, and A. Marek. A New Multi-dimensional General Relativistic Neutrino Hydrodynamics Code of Core-collapse Supernovae. III. Gravitational Wave Signals from Supernova Explosion Models. *Astrophysical Journal*, 766:43, March 2013.

[321] A. Torres-Forné et al. Universal Relations for Gravitational-Wave Asteroseismology of Protoneutron Stars. *Physical Review Letters*, 123(5):051102, August 2019.

[322] H. Sotani and T. Takiwaki. Dimension dependence of numerical simulations on gravitational waves from protoneutron stars. *Physical Review D*, 102(2):023028, July 2020.

[323] M-A. Bizouard et al. Inference of proto-neutron star properties from gravitational-wave data in core-collapse supernovae. *arXiv e-prints*, page arXiv:2012.00846, December 2020.

[324] J. W. Murphy, C. D. Ott, and A. Burrows. A Model for Gravitational Wave Emission from Neutrino-Driven Core-Collapse Supernovae. *Astrophysical Journal*, 707:1173–1190, December 2009.

[325] A. Marek, H.-T. Janka, and E. Müller. Equation-of-state dependent features in shock-oscillation modulated neutrino and gravitational-wave signals from supernovae. *Astronomy and Astrophysics*, 496:475–494, March 2009.

[326] K. N. Yakunin et al. Gravitational waves from core collapse supernovae. *Classical and Quantum Gravity*, 27(19):194005, October 2010.

[327] S. Scheidegger, R. Käppeli, S. C. Whitehouse, T. Fischer, and M. Liebendörfer. The influence of model parameters on the prediction of gravitational wave signals from stellar core collapse. *Astronomy and Astrophysics*, 514:A51, May 2010.

[328] E. Müller, H.-Th. Janka, and A. Wongwathanarat. Parametrized 3d models of neutrino-driven supernova explosions - neutrino emission asymmetries and gravitational-wave signals. *A&A*, 537:A63, 2012.

[329] K. N. Yakunin et al. Gravitational wave signatures of ab initio two-dimensional core collapse supernova explosion models for 12 -25 $M_\odot$ stars. *Physical Review D*, 92(8):084040, October 2015.

[330] T. Kuroda et al. Correlated Signatures of Gravitational-Wave and Neutrino Emission in Three-Dimensional General-Relativistic Core-Collapse Supernova Simulations. *Astrophys. J.*, 851(1):62, 2017.

[331] H. Andresen et al. Gravitational wave signals from 3D neutrino hydrodynamics simulations of core-collapse supernovae. *Monthly Notices of the Royal Astronomical Society*, 468(2):2032–2051, 03 2017.

[332] N Ohnishi et al. Standing accretion shock instability: numerical simulations of core-collapse supernova. *Journal of Physics: Conference Series*, 112(4):042018, may 2008.

[333] S. W. Bruenn and A. Mezzacappa. Prompt Convection in Core Collapse Supernovae. *The Astrophysical Journal Letters*, 433:L45, September 1994.

[334] A. Wongwathanarat, E. Müller, and H.-Th. Janka. Three-dimensional simulations of core-collapse supernovae: from shock revival to shock breakout. *Astronomy and Astrophysics*, 577:A48, 2015.

[335] D. Radice et al. Characterizing the gravitational wave signal from core-collapse supernovae. *The Astrophysical Journal*, 876(1):L9, 4 2019.

[336] J. Powell and B. Müller. Gravitational wave emission from 3D explosion models of core-collapse supernovae with low and normal explosion energies. *Monthly Notices of the Royal Astronomical Society*, 487(1):1178–1190, 05 2019.

[337] T. Kuroda, K. Kotake, and T. Takiwaki. A new gravitational-wave signature from standing accretion shock instability in supernovae. *The Astrophysical Journal*, 829(1):L14, 9 2016.

[338] E. P. O'Connor and S. M. Couch. Exploring fundamentally three-dimensional phenomena in high-fidelity simulations of core-collapse supernovae. *The Astrophysical Journal*, 865(2):81, 9 2018.

[339] Y. Bengio I. Goodfellow and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[340] C. Szegedy et al. Going deeper with convolutions. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 06 2015.

[341] C. Szegedy et al. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[342] K. He et al. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[343] C. Szegedy et al. Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI Conference on Artificial Intelligence*, 02 2016.

[344] A. Martín et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[345] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv pre-print*, 12 2014.

[346] P. Roy et al. Effects of degradations on deep neural network architectures, 07 2018.

[347] P. Antonioli et al. SNEWS: the SuperNova Early Warning System. *New Journal of Physics*, 6:114, July 2004.

[348] S. Fukuda et al. The Super-Kamiokande detector. *Nuclear Instruments and Methods in Physics Research A*, 501(2):418–462, April 2003.

[349] K.J.Chen C.H. Lin and C.Y. Hwang. Rapid Growth of Galactic Supermassive Black Holes through Accreting Giant Molecular Clouds during Major Mergers of Their Host Galaxies. *The Astrophysical Journal*, 952(2):121, August 2023.

[350] B. P. Abbott et al. Search for intermediate mass black hole binaries in the first and second observing runs of the Advanced LIGO and Virgo network. *Phys. Rev. D*, 100(6):064064, 2019.

[351] R. Abbott et al. GWTC-2.1: Deep Extended Catalog of Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run. *arXiv e-prints*, 8 2021.

[352] R. Abbott et al. Gw190521: A binary black hole merger with a total mass of 150 $M_{\odot}$. *Phys. Rev. Lett.*, 125:101102, Sep 2020.

[353] V. Gayathri et al. Enhancing the sensitivity of transient gravitational wave searches with Gaussian mixture models. *Phys. Rev. D*, 102(10):104023, 2020.

[354] G. James et al. *An Introduction to Statistical Learning: with Applications in R.* Springer, 2013.

[355] Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *J. Mach. Learn. Res.*, 5:1089–1105, 2004.

[356] A. Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[357] Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, August 2007.

[358] H. Larry and I. Niven. Averages on the move. *Mathematics Magazine*, 58(3):151–156, 1985.

[359] J. Komić. *Harmonic Mean.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[360] K. M. Ting. *Confusion Matrix.* Springer, 2010.

[361] R. C. Staudemeyer and E. Rothstein Morris. Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv e-prints*, page arXiv:1909.09586, September 2019.

[362] A. Vaswani and toehrs. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762, June 2017.

[363] G. Gonzalez. Suspensions thermal noise in LIGO gravitational wave detector. *Class. Quant. Grav.*, 17:4409–4436, 2000.

[364] P. R. Saulson. Thermal noise in mechanical experiments. *Phys. Rev. D*, 42:2437–2445, Oct 1990.

[365] W. Heisenberg. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Zeitschrift fur Physik*, 43(3-4):172–198, March 1927.

[366] K. Goda. *Development of techniques for quantum-enhanced laser-interferometric gravitational-wave detectors.* PhD thesis, Massachusetts Institute of Technology. Dept. of Physics., 2007.

[367] L. Barsotti, J. Harms, and R. Schnabel. Squeezed vacuum states of light for gravitational wave detectors. *Reports on Progress in Physics*, 82, 2018.

[368] J. Aasi et al. Enhancing the sensitivity of the LIGO gravitational wave detector by using squeezed states of light. *Nature Photon.*, 7:613–619, 2013.

[369] D. M. MacLeod et al. Reducing the effect of seismic noise in LIGO searches by targeted veto generation. *Class. Quant. Grav.*, 29:055006, 2012.

[370] S. Soni. *Identification and Reduction of Scattered Light Noise in LIGO.* PhD thesis, Louisiana State University and Agricultural and Mechanical College. Dept. of Physics., 2021.

[371] C. Casciano. *Seismic Isolation for the Test Masses of the VIRGO Gravitational Wave Antenna.* PhD thesis, Univeristy of Pisa. Dept. of Physics., 2002.

[372] M. Pitkin et al. Gravitational Wave Detection by Interferometry (Ground and Space). *Living Rev. Rel.*, 14:5, 2011.

[373] S. A. Hughes and K. S. Thorne. Seismic gravity gradient noise in interferometric gravitational wave detectors. *Phys. Rev. D*, 58:122002, 1998.

[374] J. O. Smith. *Spectral Audio Signal Processing.* W3K, 2011.