Improving our ability to distinguish gravitational-wave signals from detector transient noise for the fourth LIGO-Virgo-KAGRA observing run

A dissertation in partial fulfillment of the requirements for the degree of
Physicist

María Sofía Álvarez López

Supervisor
Dr. Alejandro García
Universidad de los Andes

Supervisor
Dr. Jess McIver
The University of British Columbia

Universidad de los Andes
Facultad de Ciencias - Departamento de Física
Bogotá, Colombia
July 16, 2023

Resumen

A pesar de haber alcanzado sensibilidades capaces de detectar la amplitud extremadamente pequeña de las ondas gravitacionales (GWs), los datos de los detectores LIGO y Virgo contienen frecuentes ráfagas de ruido transitorio no Gaussiano, comúnmente conocidas como “glitches”. Los “glitches” se presentan en diversas morfologías de tiempo-frecuencia, y resultan especialmente problemáticos cuando imitan la forma de las GWs reales. Dada la mayor tasa de eventos esperada en el actual periodo de observación de LIGO-Virgo (O4), la validación de los candidatos de eventos de GWs requiere mayores niveles de automatización. Gravity Spy, una herramienta de aprendizaje automático que clasificó con éxito tipos comunes de “glitches” de LIGO y Virgo en observaciones anteriores, tiene el potencial de ser reestructurada como un clasificador de señales de GWs-vs-ruido de detector para distinguir entre “glitches” y señales de GW con precisión. Un clasificador de señales de GWs-vs-“glitches” utilizado para la automatización debe ser robusto y compatible con una amplia gama de ruido de fondo, nuevas fuentes de “glitches” y la probable aparición de “glitches” y GWs solapados en la misma ventana de tiempo. Presentamos GSpyNetTree, el *Gravity Spy Convolutional Neural Network Decision Tree*: un clasificador multi-etiqueta multi-CNN que utiliza CNNs en un árbol de decisión ordenado a través de la masa total candidata de una onda gravitacional. Integrado en el Informe de Calidad de Datos de LIGO-Virgo (DQR, por sus siglas en inglés), GSpyNetTree es una de las herramientas esenciales en la evaluación de la necesidad de mitigación de “glitches” en O4. Esta tesis presenta el desarrollo de GSpyNetTree, su construcción y resultados, desde su origen como un clasificador multi-clase a su estado actual como clasificador multi-etiqueta. Por último, se evalúa su desempeño en candidatos de ondas gravitacionales del actual periodo de observación, O4, y se proponen técnicas para mejorar su desempeño en futuras iteraciones.

Abstract

Despite achieving sensitivities capable of detecting the extremely small amplitude of gravitational waves (GWs), LIGO and Virgo detector data contain frequent bursts of non-Gaussian transient noise, commonly known as 'glitches'. Glitches come in various time-frequency morphologies, and they are particularly challenging when they mimic the form of real GWs. Given the higher expected event rate in the current observing run (O4), LIGO-Virgo GW event candidate validation requires increased levels of automation. Gravity Spy, a machine learning tool that successfully classified common types of LIGO and Virgo glitches in previous observing runs, has the potential to be restructured as a signal-vs-glitch classifier to distinguish between glitches and GW signals accurately. A signal-vs-glitch classifier used for automation must be robust and compatible with a broad array of background noise, new sources of glitches, and the likely occurrence of overlapping glitches and GWs. This dissertation presents GSpyNetTree, the Gravity Spy Convolutional Neural Network Decision Tree: a multi-CNN multi-label classifier using CNNs in a decision tree sorted via total GW candidate mass. Integrated into the LIGO-Virgo Data Quality Report, GSpyNetTree is one of the essential tools in assessing the necessity of glitch mitigation in O4. This thesis presents the development, building process, and results of GSpyNetTree: from its origin as a multi-class classifier based on Gravity Spy, to its current O4 status as a multi-label classifier. Finally, the performance of GSpyNetTree identifying data quality issues in the public O4 GW candidates published in GraceDB is evaluated, and new ways to improve the tool's classifications are suggested.

Agradecimientos

Quisiera agradecer a mi asesora Jess McIver: gracias por mostrarme el camino de las ondas gravitacionales y permitirme enamorarme de ellas. Pero, sobre todo, estaré eternamente agradecida por su apoyo, por enseñarme a ser cada vez mejor científica mientras se es un gran ser humano, por quererme ver triunfar en mi camino personal y científico y por ser, además de una gran mentora, un verdadero modelo a seguir. Algún día espero ser para científicos más jóvenes lo que ha sido para mí. También me gustaría agradecer a mi asesor Alejandro García por la infinita confianza que ha depositado en mí, por quererme ver como una científica y persona exitosa, por preocuparse por mi como persona, y por sus valiosísimos aportes científicos a esta tesis. Gracias por permitirse explorar este mundo de las ondas gravitacionales conmigo, por ser un gran mentor y por motivarme a siempre ser la mejor versión de mí. Sin su ayuda, este trabajo no hubiera sido posible. También agradezco profundamente al Dr. Raymond Ng y a la Dr. Haydemar Núñez, cuya mentoría y aportes en los aspectos computacionales de esta tesis son inmensurables. Asimismo, quisiera agradecer a todo el grupo de ondas gravitacionales de la Universidad de British Columbia, en particular a Mervyn Chan, Franz Herbst y Beverly Berger, y al grupo de caracterización de detectores de la colaboración científica de LIGO, especialmente a Derek Davis y Sidd Soni, por sus aportes e increíbles conversaciones para enriquecer este proyecto. También agradezco al Dr. Andrés Flórez por sus enseñanzas tanto en la física, como en la investigación y como persona.

Siempre estaré eternamente agradecida con Dios y con mi familia. Gracias a mis papás, José y Carmenza, por su amor incondicional y su apoyo. Gracias por enseñarme a hacer siempre mi trabajo bien hecho y por amor a Dios; ese es el mejor regalo que pudieron haberme dado. Gracias por su amor y apoyo incondicionales, que he sentido durante toda mi vida, y por el cual he crecido como una mujer y una científica feliz. Gracias por estar para mí en cada minuto de este camino que he construido y sigo construyendo, porque desde pequeña no pude haber sido más afortunada de tener a los mejores papás del mundo. Gracias por su ejemplo, por enseñarme a ser perseverante y por dar lo mejor de mí misma; pero, sobre todo, gracias por siempre permitirme ser yo y amarme así. Gracias por ser unos seres humanos maravillosos, y mis primeros modelos a seguir. También agradezco a mi hermano Juan, porque siempre me ha apoyado en cada paso de mi vida. Gracias por ser una de las primeras personas en motivarme a ser científica y por siempre escucharme y estar para mí. Desde pequeña ha sido un gran ejemplo para mí, y lo sigue siendo. A los tres, los amo con todo mi corazón. También, agradezco a mi novio Daniel por su amor y paciencia, por su confianza en mí, por escucharme y por apoyarme en todo lo que podía. Gracias por alentarme en mis momentos difíciles y por compartir conmigo los felices. Finalmente, agradezco a Mariana Jaramillo y Daniel Serrano por ser unos grandes amigos y por siempre estar para mí. Gracias a todos los que están, o estuvieron, en algún punto de este camino, y a los que siguen para lo que está por venir.

La mayor parte de los aspectos computacionales de esta tesis fueron desarrollados usando recursos computacionales proveídos por el Laboratorio LIGO y apoyados por los Grants de la National Science Foundation de Estados Unidos PHY-0757058 y PHY-0823459.

Acknowledgements

I would like to thank my advisor Jess McIver: thanks for introducing me and exploring with me this fantastic world of gravitational-waves and for allowing me to become passionate about them. But, overall, I will be forever thankful for all your support, for showing me how to be a great scientist while being a wonderful human being, for wanting me to thrive in my personal and academic path, and for being not only a great mentor but a role model. Someday I hope to be to younger scientists what you have been to me. I would also like to thank my advisor Alejandro Garcia for the infinite trust he has placed in me, for wanting to see me succeed as a scientist and as a person, for caring about me, and for his invaluable scientific contributions to this thesis. Thank you for being a great mentor, and for always motivating me to be the best possible version of myself. Without your help, this work would not have been possible. I am also deeply grateful to Dr. Raymond Ng and Dr. Haydemar Núñez, whose mentorship and contributions to the computational aspects of this thesis are immeasurable. I would also like to thank the entire gravitational wave group at the University of British Columbia, in particular Mervyn Chan and Franz Herbst, and the detector characterization group of the LIGO scientific collaboration, especially Beverly Berger and Derek Davis, for their contributions and conversations to enrich this project. I also thank Dr. Andres Florez for his lessons not only in physics and research but also as a person.

I will always be eternally grateful to God and to my family. To my parents, José and Carmenza, for their unconditional love and support. Thank you for teaching me to always do my best, and for your unconditional love and support, for which I have grown as a happy woman and scientist. Thank you for being there for me every step of the way, for your example, for teaching me to be perseverant and to always do my best, but, above all, thank you for always allowing me to be myself and for loving me that way. Thank you for being wonderful human beings and my first role models. I also thank my brother Juan for his support in every step of my life. Thank you for being one of the first people to motivate me to become a scientist and for always listening to me and being there for me. Thanks for being a great example of success. To all three of you, I love you. Also, I thank my boyfriend, Daniel, for his love, patience, trust, and for supporting me in everything he could. Thank you for encouraging me in my difficult moments and for sharing with me the happy ones. Finally, I thank Mariana Jaramillo and Daniel Serrano for being great friends and for always being there for me. Thanks to all those who are or were with me at some point along this path and to those who will for what is to come.

Most of the computational aspects of this this thesis were developed using computational resources provided by the LIGO Laboratory and supported by National Science Foundation Grants PHY-0757058 and PHY-0823459.

Contents

1 Introduction

On September 14, 2015, the two gravitational-wave (GW) detectors of the Laser Interferometer Gravitational Wave Observatory (LIGO), recorded the stretching and squeezing of space-time due to the GW generated by two coalescing binary black holes (BBHs) of 29 $M_\odot$ and 36 $M_\odot$ [1]. Besides being a further confirmation of Einstein's theory of general relativity (GR, see chapter 2) [2], LIGO started a new field of astronomy by providing a new way of discovering and understanding the universe [2, 3]. Since the initial observation during the first observing run (O1) [4], and following major upgrades [5], there have been dozens of detections of transient GW signals, in the second (O2) [6] and third (O3a, O3b) [7, 8] observing runs. After major upgrades, the fourth observing run (O4), which started on May 24th, 2023, promises even more GW events. These GWs are generated by the asymmetric acceleration of mass distributions, and their discovery has yielded valuable astrophysical insights on populations of neutron stars and BBHs not previously provided by electromagnetic (EM) observations [2] (see Chapter 3).

LIGO detects GWs using interferometers whose basic mechanism is similar to Michelson-Morley's (see Chapter 4) and whose perpendicular arms are 4 kilometers long each. Their considerable length makes them more sensitive to the small amplitude of GWs, which produce length changes on the order of 10^{-18} m in the space-time geometry [2, 3]. This is equivalent to measuring fluctuations to a thousandth the diameter of a proton.

The required detectors' sensitivity is reached by mitigating several noise sources, such as unwanted movement of the mirrors suspended at the end of the interferometer arms, thermal noise and Brownian motion [2], and Earth's seismic activity [9]. These noise sources dictate the frequency range of possible GW detections (see Chapter 5), and limit LIGO's detection range. Using powerful lasers, massive mirrors, and isolating all sensitive LIGO components from non-GW disturbances, help achieve the required sensitivity. Nevertheless, LIGO detectors are still prone to non-astrophysical noise sources that disturb the detectors [10]. Identifying, characterizing, and segregating them is crucial to achieving confident and accurate GW detections and is part of the LIGO's Detector Characterization group's tasks.

Non-Gaussian, transient noise, commonly known as *glitches*, is particularly interesting in the context of more sensitive Advanced LIGO (aLIGO) detectors. Glitches result from instrumental, environmental, or unknown sources [11, 12] and come in various time-frequency morphologies (see Chapter 6). They are particularly challenging when they mimic the form of actual GW events, generating false-positive candidates [3], corrupting data, and biasing astrophysical parameter estimation. Along with the vast amount of data that LIGO and Virgo generate, developing robust tools and methods to identify and characterize these glitches is crucial for extracting GW signals from this data.

A common tool used for glitch classification is Gravity Spy [3], which uses a Convolutional Neural Network (CNN) image classifier to distinguish 23 different glitch classes and 1 Chirp[1] class via time-frequency visualizations, also called spectrograms. After successfully classifying

---

[1] Chirps are Hardware injections that move LIGO's mirrors to emulate real GW signals.

glitches in previous observing runs, Gravity Spy has the potential to be used as a signal-vs-glitch classifier. However, to do so, several changes need to be applied to the original architecture. This thesis builds up on the recommendations put forward by Jarov et al. [13] and Alvarez-Lopez et al. [14] to build such classifier. The classifier developed in this thesis is called GSpyNetTree, the Gravity Spy Convolutional Neural Network Decision Tree, and is a decision tree of Convolutional Neural Networks (CNNs) sorted via GW candidate mass, that detects glitches responsible for or in the presence of GW candidate events.

Before explaining how GSpyNetTree works and its performance on recently detected GW candidates in the current LIGO-Virgo-KAGRA observing run, the evolution of GSpyNetTree is detailed, starting with the proposal of Jarov et al. [13] that sets the basis for this thesis. Following this review, the architecture and performance in discerning GW candidate events from glitches of the first version (also known as the pre-O4 version) of GSpyNetTree are explained. This pre-O4 GSpyNetTree is based on a decision tree of multi-class classifiers that leverage the InceptionV3 [15] architecture, one of the state-of-the-art CNN architectures for image classification [14].

After studying the performance of the first version of GSpyNetTree on expected O4 scenarios, including the possibility of overlapping GWs and glitches, the occurrence of new types of glitches, and a different background noise, further ways to improve the classifier are presented, giving rise to the O4-era version of GSpyNetTree. The generation of the training set (including GW simulation and glitch selection), as well as new Machine Learning techniques used to increase the robustness of the tool to new glitches, a broad array of background noise, and the possibility of glitches occurring in the proximity of GWs, are described.

Before evaluating the results of the O4-era version of GSpyNetTree, the integration of the tool to the O4 Data Quality Report [16, 17], which is one of the most important tools in the LIGO-Virgo event validation pipeline, is explained. At this stage, the way in which GSpyNetTree assesses the validation of a candidate event, confirming its terrestrial or astrophysical origin, is detailed. Finally, the results of the O4 era version are explained. This includes an overview of GSpyNetTree's (outstanding) classifications in a subset of the superevents that have been registered since the start of the fourth LIGO-Virgo-KAGRA observing run. This thesis concludes with ways to further improve GSpyNetTree's classifications.

2 General Relativity and gravitational-waves

With the first detection of gravitational-waves (GWs), LIGO and Virgo opened a new window in the understanding of the Universe. It is important to study how GWs surge from Einstein's theory of General Relativity (GR). This chapter examines how GWs arise naturally from GR and analyzes some of their properties relevant to interferometric detection, following Saulson's [18, 19], Maggiore's [20], and Hughes' and Flanagan's [21] approach.

Developed by Einstein in 1915, one of the main principles of GR is that the curvature of spacetime dictates the motion of bodies within it. As these bodies move in spacetime, its curvature and geometry change. In other words, spacetime dictates how matter moves, and matter dictates how spacetime curves [22]. By curving spacetime, matter generates a gravitational field. From classical electromagnetism (EM), we know that EM waves (e.g. light and radio waves) are propagating oscillations of the EM field, emitted by accelerated charged particles. In a similar fashion, GWs are propagating oscillations of the gravitational field, generated by accelerated masses that create these reverberations ("ripples") in spacetime [21, 23].

To study the way in which GWs surge from GR, we start with the standard, flat Minkowski spacetime metric in a vacuum, $\eta_{\mu\nu}$, which is given in Cartesian coordinates by:

$$\eta_{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1) = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.1}$$

Throughout this chapter, I will use the Greek alphabet for spacetime indices (ranging from 0 to 3 (or t, x, y, z, respectively)), and the Latin alphabet for spatial components only. We now consider a small perturbation to this metric, $h_{\mu\nu}$, such that $||h_{\mu\nu}|| \ll 1$. The new, perturbed metric is given by:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}. \tag{2.2}$$

The condition imposed on the perturbation implies that the gravitational field must be weak and that the coordinate system is constrained to be approximately Cartesian [21]; and, thus, linear. This is why it is often stated that GWs surge as solutions to linearized versions of the Einstein equations in the weak-field limit [19]. Using an appropriate gauge choice, known as the transverse traceless gauge (TT gauge), Einstein's field equation becomes a wave equation. To build the TT gauge, it is useful to consider a change in notation by introducing the trace-reversed perturbation[1], instead of $h_{\mu\nu}$, which is given by:

$$\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h^{\mu}_{\mu}, \tag{2.3}$$

where h^{μ}_{μ} is the trace of the metric perturbation. As in EM, we start by applying the Lorentz

---

[1] The trace-reversed metric simplifies the calculation of the Riemann tensor in the field equations, however this is out of the scope of this dissertation. These calculations can be found in Hughes and Flanagan [21].

gauge condition (commonly used in radiation studies) to the trace-reversed metric, such that:

$$\partial^\mu \bar{h}_{\mu\nu} = 0. \tag{2.4}$$

Note the similarity with the EM Lorentz gauge, $\partial^\mu A_\mu = 0$, where A_μ is the vector potential. We can now further specialize this gauge such that the perturbation is merely spatial

$$h_{00} = h_{0i} = 0, \quad i \in \{1, 2, 3\}, \tag{2.5}$$

and traceless

$$h^i_i = 0. \tag{2.6}$$

These leaves us only with the spatial components, for which the Lorentz gauge now reads

$$\partial^j h_{ij} = 0, \tag{2.7}$$

such that the spatial metric perturbation is transverse. These three conditions define the TT gauge, in which Einstein's equations simplify to a wave equation:

$$\Box\, \bar{h}_{\mu\nu} = 0. \tag{2.8}$$

Note that, under the TT gauge, the metric perturbation becomes traceless and, thus, $\bar{h}_{\mu\nu} = h_{\mu\nu}$. In this way, we can rewrite equation 2.8 as:

$$\Box\, h_{\mu\nu} = \left(\nabla^2 - \frac{1}{c^2}\partial_t^2\right) h_{\mu\nu} = 0. \tag{2.9}$$

There are two major insights that we can get from this equation. First, GWs travel at the speed of light, c. This means that gravity is causal: all changes that affect gravitating sources are communicated to distant observers at the speed of light [20, 21]. Second, just as in EM, equation 2.9 has plane wave solutions,

$$h_{ij}(x) = \mathrm{Re}\ \epsilon_{ij}(\mathbf{k})\ e^{ik_\mu x^\mu}, \tag{2.10}$$

where $x^\mu = (x^0, \mathbf{x}) = (ct, \mathbf{x})$, $k_\mu = (-2\pi f/c, \mathbf{k}) = (-|\mathbf{k}|, \mathbf{k})$ is the 4-vector wave-vector and $\epsilon_{ij}(\mathbf{k})$ is the polarization tensor [20]. One of the best features of the TT gauge is that it shows the fact that GWs have two polarization components.

To visualize this, suppose a single plane wave propagating in the $\hat{z}$ direction. The Lorentz condition implies that $h_{3j} = 0$. Further applying the spatial condition (equation 2.5), we get that the only non-zero components of h_{ij} are h_{11}, h_{12}, h_{21} and h_{22}. By symmetry,

$$h_{12} = h_{21} \equiv h_\times, \tag{2.11}$$

and because of the traceless condition (equation 2.6),

$$h_{11} = -h_{22} \equiv h_+, \tag{2.12}$$

where h_+ and $h_\times$ are the amplitudes of the "plus" and "cross" orthogonal polarizations of the wave. In this way,

$$h_{\mu\nu}(t,z) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_\times & 0 \\ 0 & h_\times & -h_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cos\left[2\pi f\left(t - \frac{z}{c}\right)\right] = (h_+\epsilon^+_{\mu\nu} + h_\times\epsilon^\times_{\mu\nu}) \cos\left[2\pi f\left(t - \frac{z}{c}\right)\right], \quad (2.13)$$

with

$$\epsilon^+_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \epsilon^\times_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (2.14)$$

a basis of the polarization tensor. A GW might have an arbitrary polarization given by a linear combination of the "plus" and "cross" polarizations. However, it is important to consider the case of perfectly "plus" and "cross" polarized waves.

A perfectly "plus" polarized wave will have an equal and opposite effect along the $\hat{x}$ and $\hat{y}$ axes,

$$h^+_{\mu\nu}(t,z) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & 0 & 0 \\ 0 & 0 & -h_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cos\left[2\pi f\left(t - \frac{z}{c}\right)\right], \quad (2.15)$$

whereas "cross" polarized waves simultaneously affect both directions,

$$h^\times_{\mu\nu}(t,z) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & h_\times & 0 \\ 0 & h_\times & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cos\left[2\pi f\left(t - \frac{z}{c}\right)\right]. \quad (2.16)$$

These polarizations receive the names of "plus" and "cross" because of the way in which polarized GWs interact with matter and the shape of the force fields that they produce [23]. If a wave is plus-polarized, such that it is described by equation 2.15, it will produce the strain effect on matter shown in the top panel of Figure 2.1. Namely, the GW will first stretch spacetime in the $\hat{x}$ direction and compress it in the $\hat{y}$ direction. After returning to its original shape, the GW will now stretch spacetime in the $\hat{y}$ direction and compress it in the $\hat{x}$ direction. The GW will finally return to its original shape to complete a period. As the wave propagates, spacetime will be alternately stretched and compressed in each direction. On the other hand, a cross-polarized wave (equation 2.16) interacting with matter will simultaneously affect both the $\hat{x}$ and $\hat{y}$ directions, as shown in the bottom panel of Figure 2.1.

With the h_+ and $h_\times$ amplitudes we can calculate the *strain* produced by a GW, or the fractional change in distance in a given direction due to the stretching/compression of space. This *strain* is one of the fundamental measurable quantities in GW detectors (such as LIGO, Virgo, and KAGRA). In fact, it is the interaction of GWs with matter that motivates their

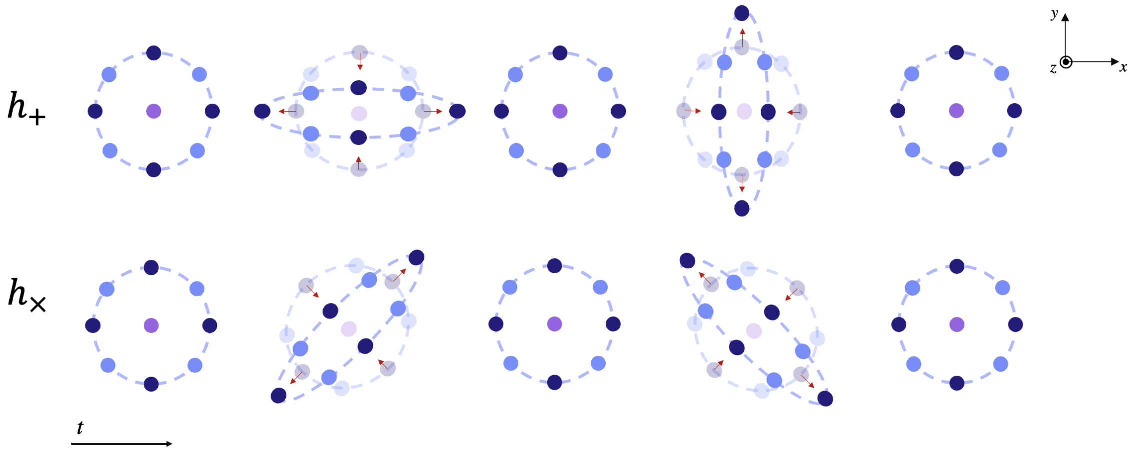


Figure 2.1: Effect of “plus” (top) and “cross” (bottom) polarized GWs, propagating in the $\hat{z}$ direction, on a circular ring of matter during one GW period. The lighter circles represent the original positions of the test particles and the arrows show the stretching/compression of spacetime in the $\hat{x}$ and $\hat{y}$ directions. Note how spacetime stretches and compresses in the $\hat{x}$ and $\hat{y}$ directions differently, depending on the polarization of the GW.

interferometric design. Chapter 4 describes these detectors and their GW detection mechanism.

2.1. How are gravitational-waves generated?

Up until now, we have studied how the solutions to the linearized version of Einstein’s equations in a vacuum (described in equations (2.15) and (2.16)) result in the propagation of GWs. We now study how this gravitational radiation is generated. Just as in EM, the emission of GWs can be expressed in terms of a retarded potential [22]. However, for situations in which the size of the source is very small compared to the wavelength ($r_{\text{source}}/\lambda \ll 1$), the multipole expansion is a more convenient approximation.

Recall that in EM, charge conservation forbids monopole radiation. The conservation of energy plays an analogous role for gravitational radiation, such that there is no monopole term either [18]. In EM, the dominant contribution to the radiated field **E** comes from the second time derivative of the electric dipole moment, **p**, such that

$$\mathbf{E} = \frac{1}{rc^2}[\hat{\mathbf{r}} \times (\hat{\mathbf{r}} \times \ddot{\mathbf{p}})], \tag{2.17}$$

where c is the speed of light in vacuum, r is the distance from the source to the field point, and **p** is defined by:

$$\mathbf{p} \equiv \int \mathbf{r}'\rho(\mathbf{r}')\, \mathrm{d}^3r', \tag{2.18}$$

where $\rho(\mathbf{r}')$ is the charge density, and the primed coordinates are associated to the position of the source, following Griffiths’ convention [24].

In contrast to EM, there is no dipole contribution for gravitational radiation. A changing dipole moment results from the motion of the center of density - charge density for EM radiation, energy density in the case of gravitation. While the center of charge of an object is free to oscillate, the oscillation of the center of mass of an isolated system violates conservation of momentum [25]. Thus, the laws of conservation of momentum require the time derivative of the dipole moment to remain constant for any isolated system.

It is with higher moments of mass distributions that GWs are generated. For situations where the source's gravitational field is small (i.e., the motion of the source is slow compared with the speed of light), the leading contribution to the generation of GWs is given by the quadrupole moment, $I_{\mu\nu}$. The gravitational analog of equation (2.17) allows us to understand the amplitude (or strain, $h_{\mu\nu}$) of the GWs generated at the source, such that:

$$h_{ij} = \frac{2G}{rc^4}\ddot{I}_{ij}^{TT}(t_r), \tag{2.19}$$

where h_{ij} is the spatial part of the perturbation metric (i.e. the GW), G the gravitational constant, c the speed of light in vacuum, r the distance to the source, $t_r = t - r/c$ the retarded time[2], and $\ddot{I}_{\mu\nu}^{TT}$ the second time derivative of the quadrupole moment of the source in the TT gauge. The specific components of this tensor depend on the properties of the source [20].

Additionally, just as their EM counterpart, GWs also carry energy and momentum in the direction of propagation. Due to conservation laws, these are dictated by the gravitational luminosity of the source, or the energy emitted per unit of time:

$$\mathcal{L} = \frac{\mathrm{d}E}{\mathrm{d}t} = \frac{G}{5c^5}\left(\sum_{i,j}\frac{\partial^3 I_{ij}^T}{\partial t^3}\right)^2, \tag{2.20}$$

where I_{ij}^T corresponds to the traceless quadrupole moment. Note that it is no longer transverse as the total luminosity is not related to a specific direction of propagation. This simplification allows the introduction of the reduced quadrupole moment,

$$I_{ij}^T = \int \rho(\mathbf{x})\left(x_i x_j - \frac{1}{3}x^2\delta_{ij}\right)\,\mathrm{d}^3x, \tag{2.21}$$

in which $\mathbf{x}$ is a position vector within the source, $\rho(\mathbf{x})$ the density of the source, and δ_{ij} the Kronecker delta.

In order to have $\ddot{I}_{\mu\nu} \neq 0$, it is necessary to have an asymmetric mass distribution. Spherically and axially symmetric systems have $\ddot{I}_{\mu\nu} = 0$; and, therefore, do not emit GWs [2]. Some examples of sources of gravitational radiation include compact objects orbiting each other, or rotating neutron stars with asymmetries on their surfaces. A more detailed description of these and other sources of GWs is given in Chapter 3.

---

[2] Just as in EM, this means that the current configuration of the system depends on its configuration at a time r/c in the past.

3 Types and sources of gravitational-waves

Sources of GWs are large, fast-rotating (usually moving at relativistic speeds), and asymmetric mass distributions. From equation (2.19), it is required that the second derivative of the quadrupole moment is large enough to compensate for the constant $G/c^4 \approx 10^{-45}\,\mathrm{s\,kg^{-1}\,m^{-1}}$ and the distant location of GW sources (recall that $h_{ij} \propto 1/r$) [2]. Therefore, sources need to be massive, and usually relativistic, in order to be detected. This means that detectable sources are namely astrophysical, and they cannot be generated on Earth.

There are several sources of GWs (some of which have already been detected with LIGO and Virgo, such as Compact Binary Coalescenses - CBCs). However, the first solid evidence for the existence of GWs was the Hulse-Taylor binary pulsar, PSR1913+16 [26], discovered in 1974 by Hulse and Taylor using the Arecibo telescope in Puerto Rico. Pulsars are rotating neutron stars that emit bursts of radio waves once or twice in every rotation period [18].

Examining the evolution of the orbit, Hulse and Taylor noted a gradual decrease in the time for one full 7.75-hour orbit [2]. This increase in the system orbiting rate resulted from the neutron stars getting closer to each other, which in turn was caused by the loss of orbital energy from the system. Hulse and Taylor noted that the rate at which this system was losing energy was the same rate GR predicted for the emission of GWs. The reduction of the period of the orbit was a direct consequence of the source generating GWs. Figure 3.1 shows the decrease of the orbital period for 30 years of observations. Even though the reduction is small (40 s), it has been very well measured to agree with GR. This observation is regarded as the first indirect proof of the existence of GWs.

We now know that many processes and objects in the Universe generate different types of GWs, namely: Continuous Gravitational-waves (CWs), Stochastic GWs, Burst GWs, and Compact Binary Coalescences (CBCs). The latter are the only types of signals that we have discovered with aLIGO and AdVirgo up to date. The following subsections describe the aforementioned types of GWs and the sources that generate them, focusing on CBCs - the ones used for the simulations in this thesis.

3.1. Compact Binary Coalescences

Compact Binary Coalescenses (CBCs) are the prototypical aLIGO and AdVirgo GW signal sources [2], and they are generated by a variety of astrophysical objects, such as black holes, binary neutron stars, and binary white dwarfs. As a matter of fact, the first GW signal ever discovered, GW150914, was a CBC of two stellar-mass black holes [1]. Just as the Hulse-Taylor pulsar showed, a pair of compact, *coalescing* astrophysical bodies (i.e., orbiting each other) radiate energy in the form of GWs. The energy loss of the system results in the shrinkage of the orbital period, which reduces the distance between the two orbiting compact objects and increases their speed. This process is known as the *inspiral* phase of a CBC. When they are very close together, the two objects combine into a single one during the *merger* phase. The

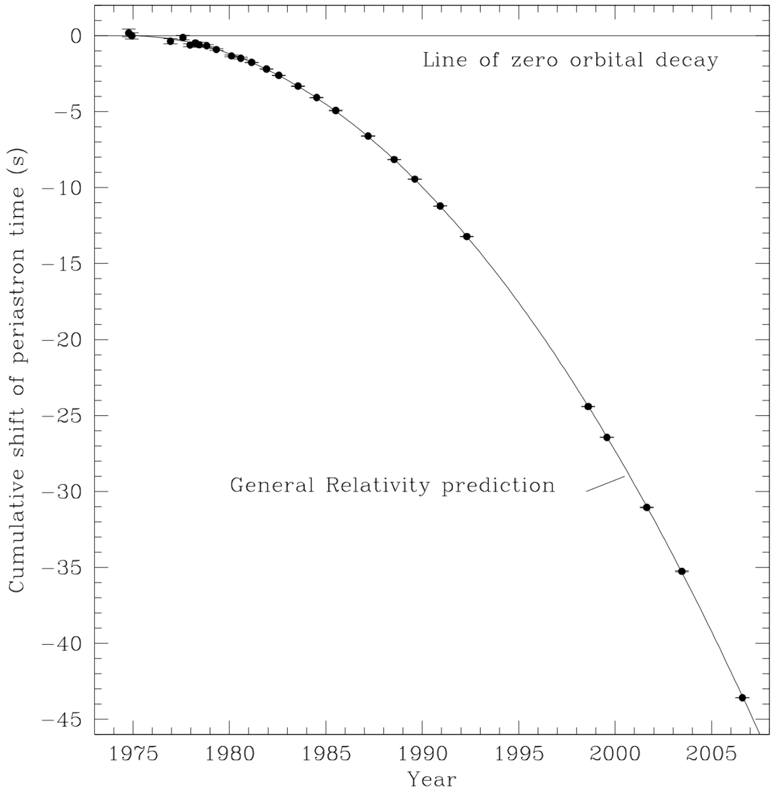


Figure 3.1: Orbital decay caused by the loss of energy by gravitational radiation which results in the emission of GWs by the Hulse-Taylor binary pulsar, PSR1913+16. (Reproduced from [27]).

following sections briefly describe these stages, as well as the sources that generate these GWs.

3.1.1. Inspiral Phase

The inspiral phase of a CBC begins with two compact objects far apart, orbiting each other in a Keplerian orbit, as shown at $t = -2$ s in Figure 3.2 for a pair of coalescing black holes. The inspiral phase of a binary neutron star system is similar, and is shown in Figure 3.8 of subsection 3.1.4.

If we model our sources as point masses m_1 and m_2, the strain of the system will evolve as [29]:

$$h_+(t) = 2h(t)\frac{1+\cos^2(i)}{2}\cos(\Phi + \phi_c), \tag{3.1}$$

for a plus-polarized GW, and

$$h_\times(t) = 4h(t)\cos(i)\sin(\Phi + \phi_c), \tag{3.2}$$

for a cross-polarized GW. In equations (3.1) and (3.2), i is the inclination angle of the orbital angular momentum of the source with respect to the observer, Φ the phase of the gravitational-wave, ϕ_c the phase of the GW at the time of coalescence t_c, and $h(t)$ is the amplitude of the GW, given by:

$$h(t) = \frac{1}{r}\mathcal{M}^{5/3}(\pi f_{\mathrm{GW}})^{2/3}. \tag{3.3}$$

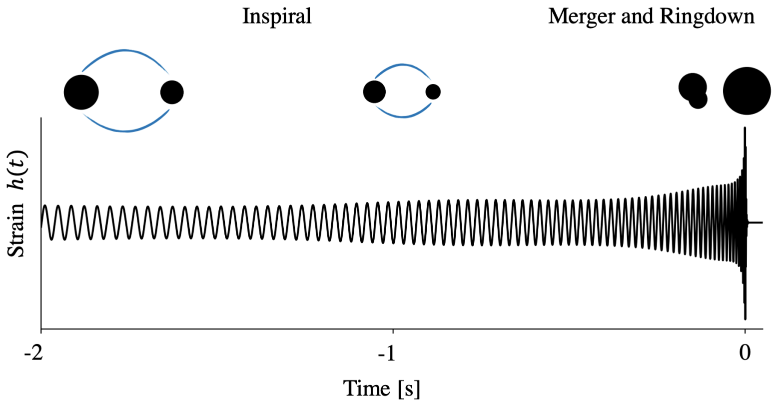


Figure 3.2: Gravitational waveform of the last two seconds of a simulated CBC signal for a system of binary black holes (BBH), along with diagrams of the two compact objects showing the evolution of the system. A pair of coalescing binary black holes (BBH) get closer to each other, as a result of a reduction in their orbital period due to the emission of gravitational radiation during the inspiral phase. The amplitude modulation that occurs in this phase is due to spin-induced precession of the orbital plane. The two compact objects finally merge together at $t = 0$ s. The resulting black hole enters a ringdown phase, in which it emits GWs until reaching equilibrium. (Adapted from [28]).

Here, r is the distance to the source and f_{GW} is the frequency of the gravitational radiation (which is twice the frequency of the orbital system). We have introduced $\mathcal{M}$, the 'chirp' mass,

$$\mathcal{M} = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}, \tag{3.4}$$

named after the evolution of the signal as it 'chirps' or increases in frequency during the inspiral phase [2]. This 'chirping' is shown in Figures 3.4 and 3.9 for GW150914 and GW170817, the first ever recorded Binary Black Hole (BBH) and Binary Neutron Star (BNS) GW events, respectively. Note that signals with bigger $\mathcal{M}$ have larger amplitudes. This is why a pair of coalescing black holes emit stronger gravitational radiation than a system of binary neutron stars.

During the early inspiral, the phase Φ and frequency f_{GW} of the GW evolve as [29]:

$$\Phi_{\mathrm{GW}} = -2\left(\frac{t_c - t}{5\mathcal{M}}\right)^{5/8}, \tag{3.5}$$

$$f_{\mathrm{GW}} = \frac{1}{2\pi}\frac{\partial \Phi}{\partial t} = \frac{1}{\pi \mathcal{M}}\left(\frac{5}{256}\frac{\mathcal{M}}{t_c - t}\right)^{3/8}. \tag{3.6}$$

In time, as the objects get closer to each other due to the shrinkage of the system's orbit caused by the emission of gravitational radiation, the GW frequency (3.6) and amplitude (3.3) increase. Note that systems with higher 'chirp' mass result in lower frequencies, which is one of the main motivations for discriminating signals by the mass of the sources that originated them in GSpyNetTree.

Before merging but very advanced in the inspiral phase, the binary system enters a post-Newtonian (PN) regime, for which the previous approximations no longer hold. This happens between $t = -0.5$ s and nearly before $t = 0$ s for the binary black hole system shown in Figure 3.2.

The PN approximation provides corrections to the late inspiral phase that effectively describe the nearly-relativistic binary system at this stage[1] [30].

3.1.2. Merger Phase

When the two compact objects coalesce, the system becomes relativistic and highly dynamical, such that the approximations used for the inspiral phase are no longer valid. In this case, numerical relativity (NR) is needed to understand how the system merges[2]. Merger occurs at $t = 0$ s both for the BBH system shown in Figure 3.2 and for the BNS system in Figure 3.8.

Following the merger phase, the state of the system depends on the CBC sources. If these were a pair of black holes, the merged object will now start to radiate gravitational radiation during the **ringdown** phase, until reaching equilibrium. The frequency and damping time of the radiation are determined by the black holes' mass and spin. Figure 3.3 shows the state of a BBH system ±10 ms around the merger time.

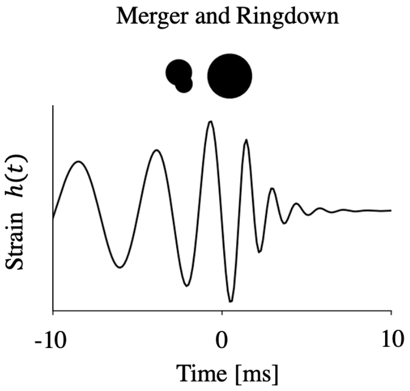


Figure 3.3: Gravitational waveform of the simulated CBC signal of Figure 3.2, 10 ms around the merger time at $t = 0$ s. The two objects are very close to each other before $t = 0$ s and they finally merge into a single, bigger object at $t = 0$ s, as shown in the two diagrams on top of the waveform. Following the merger, the new black hole rings down emitting GWs until reaching a state of equilibrium. (Adapted from [28]).

If the system was a BNS (Binary Neutron Star system) or BHNS (Black Hole - Neutron Star system), the resulting merged object depends on the masses of the original sources, as explained in subsections 3.1.4 and 3.1.5, respectively. The following subsections describe the most relevant types of CBCs, namely stellar remnants such as systems of binary black holes (BBH), binary neutron stars (BNS), binary systems of a neutron star and a black hole (NSBH), and white dwarfs (for lower frequency GWs).

3.1.3. Binary Black Hole Systems

Up to date, binary black hole (BBH) signals are the most common GW events in all LIGO-Virgo observing runs [4, 6–8]. This is because of their small radii and large masses, which make their strain $|h| \sim 10^{-21}$ detectable by our current GW interferometers [1]. Even though BBHs

---

[1] The post-Newtonian approximation is out of the scope of this thesis. In their review, Blanchet [30] provides a description of these corrections for the late inspiral phase.

[2] The NR treatment can be followed in Baker et al. [31].

are not visible in the electromagnetic spectrum, they are interesting sources of gravitational radiation because they are some of the fastest and most dense objects in our Universe. Indeed, black holes are the most compact objects known, and they are the expected remnant of a Main Sequence star of mass $m > 25\ M_\odot$ [32].

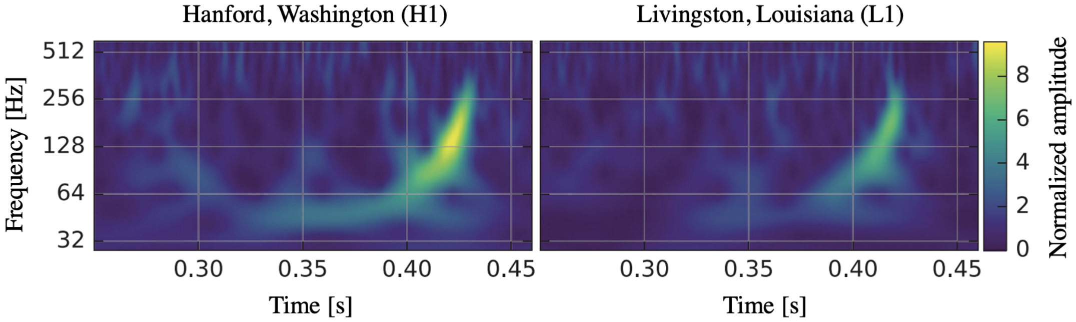


Figure 3.4: Time-frequency visualizations of GW150914, as detected in the Hanford (left) and Livingston (right) observatories. Note how the signal 'chirps' up in frequency near the merger time. (Adapted from [1]).

The most famous BBH event is GW150914, the first ever recorder signal of GWs [1]. Figure 3.5 shows the strain $h(t)$ recorded by the LIGO Hanford (H1) and LIGO Livingston (L1) detectors, along with waveform reconstructions and Numerical Relativity predictions consistent with the parameters recovered from the event.

Using parameter estimation and other techniques, it was determined that this signal had a chirp mass of $\mathcal{M} \approx 30\ M_\odot$, with $m_1 = 35^{+5}_{-3}\ M_\odot$ and $m_2 = 30^{+3}_{-4}\ M_\odot$ (in the source frame), resulting in a merger of $M = 62^{+4}_{-3}\ M_\odot$ [33]. It was determined that this system was a BBH because of its calculated mass and orbital frequency (75 Hz). While a pair of neutron stars are compact enough, they could not possibly have had the mass calculated for GW150914. Similarly, a black hole-neutron star binary would have had a larger total mass than that calculated for GW150914, and would thus have merged at a much lower frequency [1]. Moreover, the decay of the waveform after the merger time is consistent with the damped oscillations of a black hole during the ringdown phase, confirming GW150914 as the gravitational radiation produced by a BBH coalescence.

Besides the strain $h(t)$ shown in Figure 3.5, a very useful way of visualizing gravitational-wave data is via time-frequency visualizations, also called spectrograms. Figure 3.4 shows these representations for GW150914 in the Hanford and Livingston observatories. Note how the signal 'chirps' up in frequency during the inspiral phase and approaching the merger time. Spectrograms are the types of visualizations used to generate the samples of GSpyNetTree.

Following GW150914, LIGO and Virgo have detected several GWs from BBH systems. Having a plethora of different astrophysical parameters (e.g., sky location and mass), these signals are morphologically different from each other. A remarkable case is GW190521: the most massive BBH merger detected up to date. With component masses of $85^{+21}_{-14}\ M_\odot$ and $66^{+17}_{-18}\ M_\odot$, its remnant mass was calculated to be $142^{+28}_{-16}\ M_\odot$ [34].

The more massive a merger is, the shorter the signal and the lower its frequency. Figure 3.6 shows the spectrogram visualizations of GW190521 in the LIGO Hanford, LIGO Livingston, and Virgo observatories. Note how this signal differs considerably in morphology compared

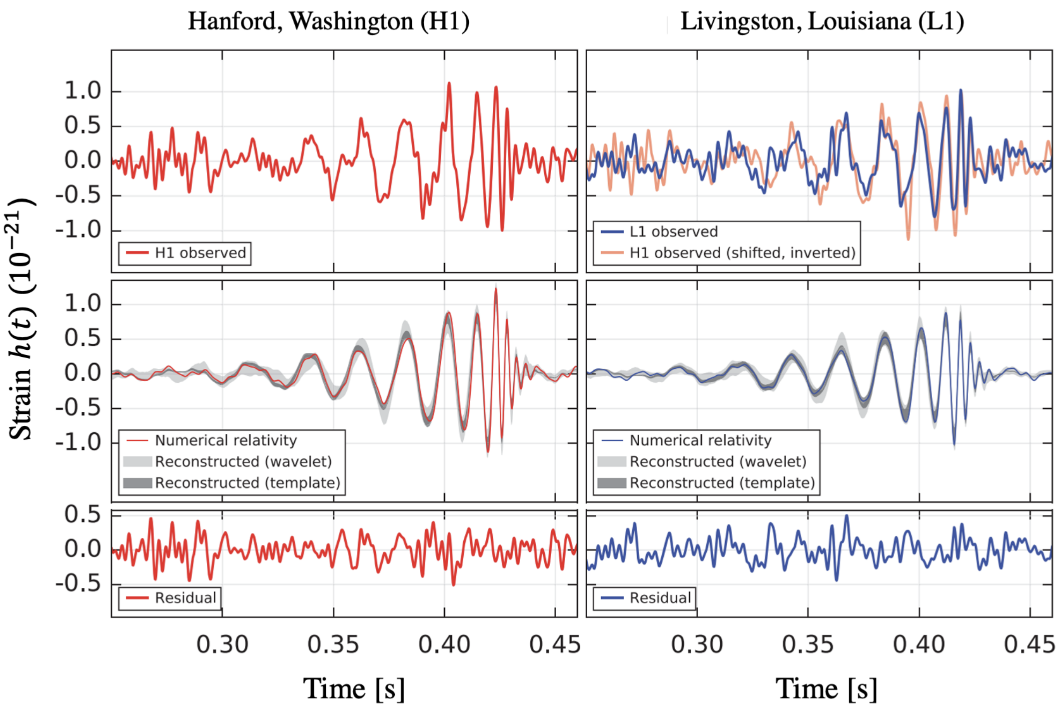


Figure 3.5: *First row:* Time series showing the measured strain for the gravitational-wave event GW150914 observed by the LIGO Hanford (left column) and LIGO Livingston (right column) observatories. *Second row:* A concordant NR waveform (in red) for a system with parameters consistent with those recovered from GW150914, along with 90% credible regions for two independent waveform reconstructions from the signal, shaded in gray. Note how the measured data is concordant with the NR simulations and waveform reconstructions, though the early inspiral phase is noisier in the time series data due to instrumental and noise constraints that do not allow the reconstruction of the early inspiral. *Third row:* Residuals after subtracting the measured time series from the NR simulation, demonstrating the significance of the event. (Adapted from [1]).

to GW150914, shown in Figure 3.5. This is a result of the latter being a lower mass merger. Considering these differences in GW morphology is not only fundamental for their searches; but also makes some signals more similar to particular kinds of transient noise (i.e., glitches, explained in Chapter 6) than others. For this reason, GSpyNetTree considers different mass ranges for GWs, along morphologically similar glitches, for each of its classifiers.

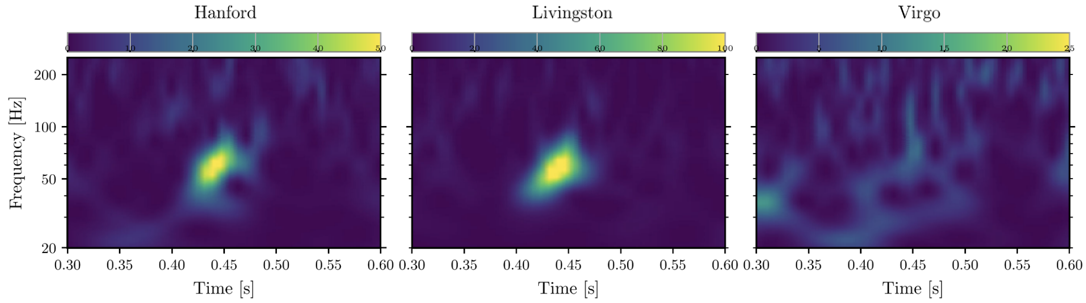


Figure 3.6: Time-frequency visualizations of GW190521, the most massive gravitational-wave event detected up to date. Compared to lower mass events, like GW150914, it is a shorter, and lower-frequency event. This is characteristic of higher mass GW events. (Adapted from [34]).

GW190521 was the first intermediate mass black hole (IMBH) ever detected by LIGO and Virgo [34]. An IMBH is a class of black hole with mass $100\ M_\odot \leq M \leq 10^4\ M_\odot$; much less massive than stellar black holes, with masses ranging from $5\ M_\odot$ to several tens of solar masses [23, 35]. More massive black holes - Supermassive black holes (SMBHs), with masses above $10^5\ M_\odot$, are usually found in the centers of galaxies, like Sagittarius A* in the Milky Way. Because of their mass, binaries of SMBHs have much lower frequencies and larger amplitudes than IMBHs and stellar black holes, and cannot be detected with ground-based detectors. This is where space-based interferometers, like LISA (explained in Subsection 5.2.2), will complement our understanding of the Universe by detecting sources with frequencies in the order of milliHertz [2].

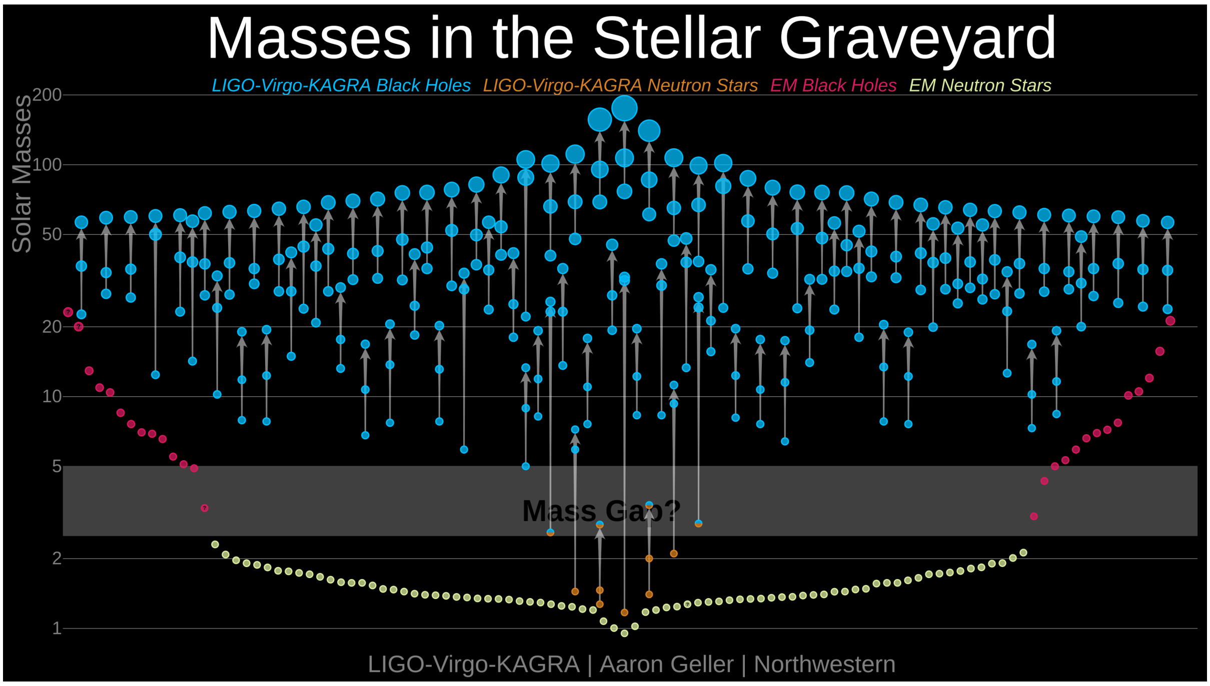


Figure 3.7: Catalog of gravitational-wave events observed in the first, second, and third Observing Runs of LIGO-Virgo, alongside neutron stars and black holes from electromagnetic observations. The arrows that connect pairs of black holes, neutron stars, and a black hole and a neutron star, represent binary mergers of each type of these types observed by LIGO and Virgo. (Reproduced from [34]).

Most of the BBH GW events detected by LIGO-Virgo until now, like GW150914, are stellar mass black hole binaries. The catalog of all GW events detected up until the third Observing Run with LIGO-Virgo, along with black holes and neutron stars from electromagnetic observations, is shown in Figure 3.7. Note how most of the detected signals are, indeed, stellar mass black holes, with just a few (including GW190521) above $100\ M_\odot$. This Figure also shows the population of neutron star binaries (explained in the following Subsection) and black hole - neutron star systems (explained in Subsection 3.1.5) detected up to date.

3.1.4. Binary Neutron Star Systems

After black holes, neutron stars are the most compact objects in the Universe. They are also stellar remnants, and are the collapsed cores of massive supergiant stars, which had a total mass between 10 $M_\odot$ and 25 $M_\odot$ when they were on the Main Sequence [32]. Neutron stars get their name from the fact that they are mainly composed of neutrons, and are held up primarily by neutron degeneracy pressure and repulsive nuclear forces.

Unlike black holes, neutron star binaries have been previously observed via electromagnetic methods, as explained before for the Hulse-Taylor binary pulsar. Therefore, many of their parameters are well known. Their mass, for instance, is expected to be around 1.4 $M_\odot$. Their radius, though estimated to be around 10 km, is less well measured. This is because it depends on the equation of state of matter at densities above that of the nucleus of an atom, which is not known yet, and has many theoretical difficulties associated with its derivation.

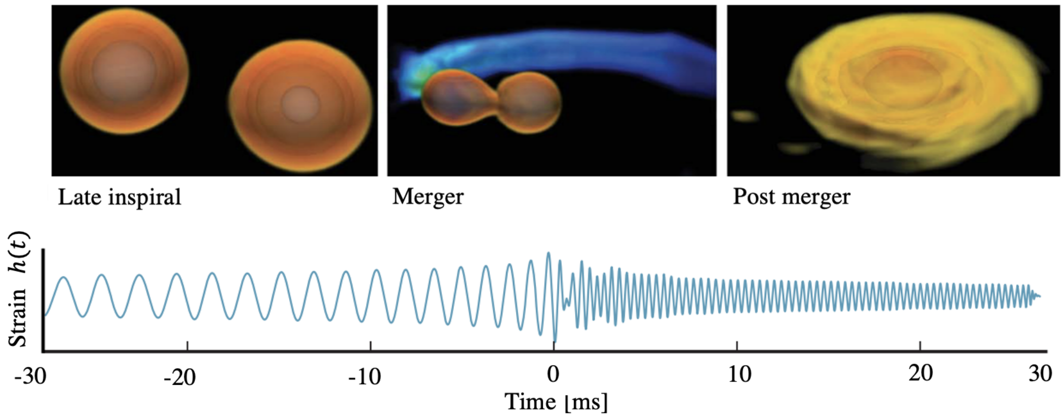


Figure 3.8: Gravitational waveform of the 30 milliseconds around merger time for a binary neutron star (BNS) system, along with simulations of the two compact objects showing the evolution of the system. The color of the objects corresponds to the density of the material, from low (blue) to high (dark red). A pair of coalescing binary neutron stars get closer to each other, as a result of a reduction in their orbital period due to the emission of gravitational radiation during the late inspiral phase. In this phase, they are tidally disrupted, after being initially tidally deformed. After merging, the new object continues emitting high-frequency GWs, and a disk of matter is formed around the new object (Adapted from [28]).

Neutron stars are often found in binary systems. Since the discovery of the Hulse-Taylor binary pulsar, radio pulsar surveys have found several BNS systems in the Milky Way [36]. Just as black holes, they are gravitationally attracted toward each other. In this process, their orbit starts to shrink due to the emission of gravitational radiation, as shown in Figure 3.8. This happens during the inspiral phase. When the two objects are very close together, they merge [32]. The resulting object will be a more massive neutron star, or a black hole, depending on whether the mass of the remnant surpasses the Tolman-Oppenheimer-Volkoff limit (i.e., the new, compact object will be a black hole if its mass is above 2.01 - 2.17 $M_\odot$ [37]).

As BNS systems are comprised of matter, electromagnetic emission is expected to occur alongside the merger. This makes them fundamentally different from BBH mergers, and provides a window for multi-messenger astronomy studies. The first neutron star merger detected with LIGO-Virgo up until now, GW170817, shows this feature [38].

GW170817 was observed on 17 August 2017, and it was the result of a BNS merger originated in NGC4993 [39], a lenticular galaxy located in the Hydra cluster. The spectrogram visualizations for this event, as observed in the LIGO Hanford, LIGO Livingston, and Virgo observatories, are shown in Figure 3.9. Note how the signal ‘chirps’ up in frequency as the system approaches the merger time, as seen in both LIGO detectors, but not in Virgo. When GW170817 was discovered, the detector horizon (maximum distance at which a detector is able to detect a BNS system with a signal-to-noise ratio of 8 (SNR=8) [38]), was 218 Mpc for LIGO-Livingston, 107 Mpc for LIGO Hanford, and 58 Mpc for Virgo [40]. The low BNS horizon for Virgo and the direction of the source with respect to the detector's antenna pattern[3] are the main reasons why the signal is not visible in the Virgo detector [38].

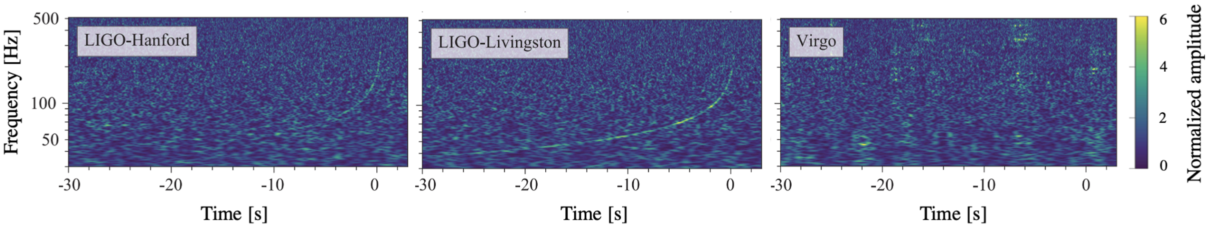


Figure 3.9: Time-frequency visualizations (spectrograms) of the binary neutron star merger GW170817, as observed in the LIGO Hanford (left), LIGO Livingston (center), and Virgo (right) observatories. The amplitude scale is normalized to each detector's noise amplitude spectral density [38]. The signal is clearly visible in both LIGO observatories, but not in Virgo, as explained in the text. Note how the BNS signal (especially in the Livingston observatory) can be traced back 30 s before the merger time at $t = 0$ s. This is a unique feature of these systems, compared to most BBH binaries, which are usually shorter in nature (as seen in Figure 3.4). (Adapted from [38]).

The BNS system that generated GW170817 had a chirp mass of $\mathcal{M} = 1.188^{+0.004}_{-0.002}\ M_\odot$ and a total mass $2.73\ M_\odot \leq M \leq 3.29\ M_\odot$, with individual components between $0.86\ M_\odot$ and $2.26\ M_\odot$ [38]. Aside from the EM observations that confirmed GW170817 as a BNS system, these measures further support the system's origin. Not only known neutron star binaries have total masses between $2.57\ M_\odot$ and $2.88\ M_\odot$ with components between $1.17\ M_\odot$ and $1.6\ M_\odot$ [41]; but also, BBH systems in our galaxy have masses substantially greater than the ones measured for GW170817. This ruled out a binary black hole system as the source of GW170817.

One of the most spectacular outcomes of GW170817 is the electromagnetic follow-up campaign that succeeded the detection. The first EM signal detected was GRB 170817A, a 2−second gamma-ray burst observed 1.74 ± 0.05 s after the merger time [42]. Over the following days and weeks after the LIGO-Virgo detection, several telescopes detected the event in other wavelengths, ranging from X-ray to radio, within the 28 deg^2 sky region to which LIGO and Virgo constrained the event. The EM observations showed that the source, which was located at 40^{+8}_{-14} pc from Earth, generated a fast-moving and rapidly cooling neutron-rich cloud, similar to the one depicted in the top-right of Figure 3.8. This is the expected remnant of a neutron star merger in its post merger phase.

Another remarkable feature of GW170817 is that it could be traced back in time further than most BBH events. Note that Figure 3.9 shows a time-scale almost 70 times larger (up until 30 s before merger) than the one in Figure 3.4, which traces the signal 0.45 s back in time. Lower mass signals cover a wider time and frequency range, which is why GWs detected from BNS systems with current detectors usually produce longer, higher-frequency signals than

---

[3] A description of the antenna pattern of a detector is given in Chapter 4.

those generated by BBHs. Notably, mass plays a substantial role in the morphology of a signal, which is one of the key considerations that GSpyNetTree takes into account. It is important to highlight, however, that during the early inspiral phase, both of these types of CBC systems emit gravitational radiation in the form of continuous waves (CWs)[4]. These GWs have not been detected yet, but as LIGO and Virgo get upgraded and other detectors (both ground and space-based) join the network, new types of GWs are expected to be detected.

3.1.5. Black Hole-Neutron Star Systems

Black Hole-Neutron Star (BHNS) systems are the last kind of events that LIGO and Virgo have detected [43]. Unlike BBH and BNS systems, BHNS binaries result from the coalescence of two different types of compact objects. Nevertheless, as they are CBCs, they share some evolutionary phases with the former.

Just as BBHs and BNSs, BHNS systems follow an inspiral phase in which their orbits shrink due to the emission of gravitational radiation in the form of GWs. When the two compact objects are very close together, they merge. The debris of the event in the post-merger phase depends on several parameters. If the black hole is very massive or if it has low spin, the neutron star is entirely swallowed by the black hole and no matter (hence, no light) is expelled. However, it is also possible that the neutron star is tidally disrupted by the black hole. In these cases, similar to BNS mergers, there is a post merger phase in which more matter is ejected or accreted into the black hole [44], as shown in Figure 3.10. These events are multi-messenger astronomy prospects, as they can be observed in different wavelengths.

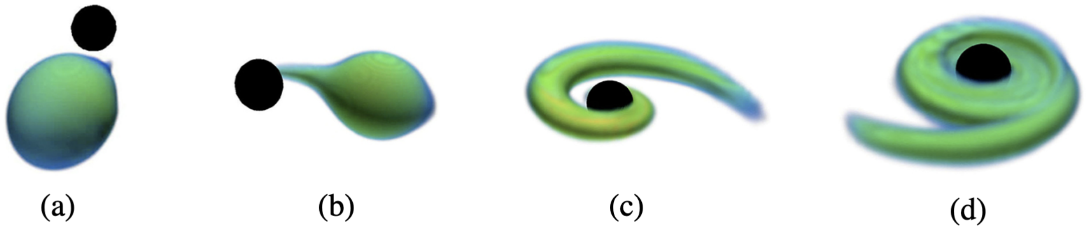


Figure 3.10: Merger and post merger phases of a tidally disrupted Black Hole-Neutron Star system. (a) Mass from the neutron star starts to be accreted onto the black hole. (b) Mass is transferred from the neutron star to the black hole in an unstable fashion. (c) The mass transferred from the neutron star forms a tidal tail. (d) The tail evolves into an accretion disk around the black hole. (Adapted from [44].)

Two confirmed BHNS events, GW200105 and GW200115, occurred only 10 days apart from each other. The first event, observed by LIGO Livingston and Virgo, had component masses $8.9^{+1.2}_{-1.5}\ M_{\odot}$ (for the black hole) and $1.9^{+0.3}_{-0.2}\ M_{\odot}$ (for the neutron star). For GW200115, observed by both LIGO detectors, the black hole and neutron star had masses of $5.7^{+1.8}_{-2.1}\ M_{\odot}$ and $1.5^{+0.7}_{-0.3}\ M_{\odot}$ respectively. All of these masses are within the 90% credible region [43]. These two events were classified as BHNS systems because, while the heavier objects in each binary have masses only known to black holes, the lighter have masses lower than those of any known black holes (and characteristic of neutron stars).

---

[4] Continuous waves are introduced in Section 3.2.

While it is possible that these two events emitted light, the electromagnetic follow-up campaign did not detect any [43]. This could be a result of the neutron star being entirely swallowed by the black hole. However, as the location of the events was rather uncertain (they were localized within a credible sky region of 7200 deg^2 and 600 deg^2 at 280^{+110}_{-110} Mpc and 300^{+150}_{-100} Mpc for GW200105 and GW200115, respectively [43]), it is also possible that the telescopes that followed-up the systems did not find any light, as it could have faded away before it was detected.

GW detections of BHNS signals help us understand the formation and evolution of these systems. Up until now, two main known possibilities exist. The first corresponds to isolated binary evolution systems, in which two compact objects already orbiting each other in a binary system explode into supernovae. While one star leaves a black hole as its remnant, the other leaves a neutron star. In this case, it is expected that the neutron star orbits in the equatorial plane of the black hole (i.e., the spin of the black hole should be aligned with the binary orbit). The other possibility corresponds to dynamical interaction, which is most likely to occur in stellar dense environments such as globular clusters. In these processes, the neutron star and black hole come from separate, non-interacting stars (and, therefore, supernova explosions) and only find and attract each other gravitationally when they are already remnants. As they were not a binary system since their formation, the neutron star could have any orientation with respect to the black hole. Hence, there is no preferred spin direction [44]. As a matter of fact, for GW200115, the spin of its black hole was opposite to the direction of the binary orbit [43], which suggests it formed via the dynamical interaction process.

BHNS events are less common than BBH signals; however, we expect that the next observing run (O4) comes with more BHNS system detections, that can hopefully be followed-up electromagnetically. Further understanding of BHNSs will enrich our knowledge of CBC systems detectable with ground-based detectors. In the following subsection, we will discuss a type of CBC systems that, due to their low frequencies, cannot be detected with aLIGO and AdVirgo. These are white dwarf binaries.

3.1.6. White Dwarf Systems

White dwarfs are the stellar remnants of Main Sequence stars with masses below 10 $M_\odot$, including our Sun [32]. The closest white dwarf known is Sirius B, the smaller component of the Sirius binary star system, located 8.6 light years from Earth. These stellar remnants are the cores left after the stars consume their hydrogen and helium shells. Compared to Main Sequence stars, they are more compact. However, they are much larger than black holes and neutron stars. In fact, the typical white dwarf radii is 1000 times bigger than that of neutron stars [2]. White dwarfs are primarily held by electron degeneracy pressure. As they decrease in radius, their mass increases until they reach the Chandrasekhar limit at 1.4 $M_\odot$ and implode into supernovae, generating a neutron star [45].

White dwarfs can be found in binaries, and several electromagnetic surveys have cataloged a wide variety of these systems [46, 47]. Nevertheless, these sources cannot be detected by aLIGO detectors, as their sensitivity (approximately 10-20000 Hz [2], as explained in Chapter 4) is almost 100 times larger than the frequency of the GWs generated by white dwarf systems. To observe these signals, space-based observatories such as LISA [48] (see Subsection 5.2.2), which will be sensitive to the mHz frequency band [2], are needed. Besides detecting the merger of these systems, LISA will be able to observe them for several years before they coalesce. This will allow LISA to detect the continuous gravitational-wave radiation (CWs) they emit during the inspiral phase. CWs are explained in the following subsection.

3.2. Continuous waves

Continuous gravitational-waves (CWs) are generated by systems with a well-defined frequency. Different from CBCs, they do not ‘chirp’ up in frequency with time. Some types of CWs are the gravitational radiation emitted by BBH, BNS, and BHNS systems long before merger during the early inspiral phase, or an off-axis mass asymmetry in a neutron star that creates a time-dependent quadrupole moment [2]. Temperature variations in neutron stars can lead to asymmetric deformations in their crusts that generate GWs when the star spins [49].

LIGO aims to detect CWs from neutron stars in the Milky Way. However, tracking these signals is not easy, as they are smaller in amplitude compared to the GWs that our detectors are currently able to observe (the strain of CWs is $|h| \sim 10^{-24}$ [50]). In short periods of time, a CW from a rotating neutron star would have nearly constant frequency and amplitude. However, over a longer interval, the signal will evolve reducing its frequency due to the relative motions of Earth and the neutron star, in addition to the reduction in the spin of the latter as it loses energy via electromagnetic and gravitational radiation [50]. Figure 3.11, shows the frequency evolution of a CW in different time scales. Note that, even though the frequency reduces over time, it only diminishes around 0.04 Hz in 2000 days. This is why they are called Continuous gravitational-waves: their strain and amplitude can be approximated as constant (i.e., with very small changes) over large periods of time.

3.3. Bursts

Advanced LIGO is expected to be sensitive enough to detect short GW signals, or bursts. These may come from core-collapse supernovae (CCSN) within the Milky Way [2, 51], but may also include bursts from Magnetars[5] [52] or *brehmstrahlung* radiation emitted from parabolic encounters of black holes in globular clusters [53].

CCSNs, or Type II Supernovae, are the best candidates for GW bursts. However, few simulations of these systems exist, as emulating them involves complex and computationally expensive interactions, like electromagnetism, fluid dynamics, general relativity, and particle physics. As a result, as opposed to CBCs, a set of confident waveforms for these kind of events is still to be generated. Moreover, these rapid and violent explosions of massive stars are extremely rare in our galaxy; with only one event per century [2]. In order to be able to observe these signals, future generation ground-based detectors (like the Cosmic Explorer [54] and the Einstein Telescope, [55] explained in Subsection 5.2.1) are needed. With 10 times greater reach, bringing the M81 galaxy into range, they have twice the chances of registering a CCSN [2, 56].

---

[5] Magnetars are neutron stars with very high magnetic fields.

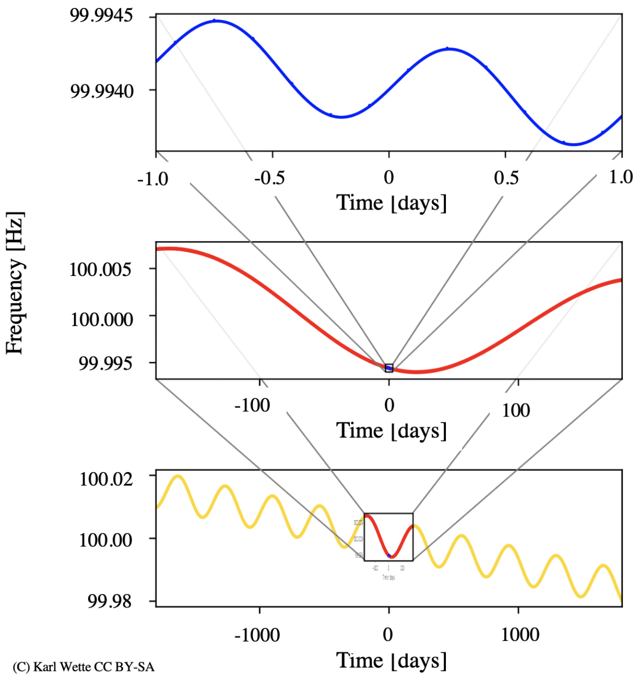


Figure 3.11: Frequency evolution of a simulated continuous gravitational-wave (CW) signal in different time-scales. The top panel shows, in the span of two days, the reduction in frequency due to Earth's rotation. The middle panel shows the frequency changing due to Earth's orbit around the Sun in one year. The bottom panel shows, over a broader time scale, the reduction in frequency as the spin of the neutron star decreases with time. (Adapted from [50].)

3.4. Stochastic gravitational-wave signals

Stochastic GW signals vary significantly from transient GW signals (CBCs and burst events) and CWs. While these signals come from specific locations in the sky, a stochastic background comes from all directions [57]. Stochastic GWs are expected to give us information about the early evolution of our Universe. Similar to the Cosmic Micro-wave Background (CMB), the left-over EM signals from the Big Bang, these GWs would have generated a cosmic gravitational-wave background, formed by the combination of a large number of random, independent events [50].

LIGO and Virgo have not detected stochastic GW signals yet. In fact, with current technologies, these GWs would appear as just noise in a single gravitational-wave detector [57]. For this reason, detecting these signals requires data analysis methods highly different from those used for CBC GWs, which are the gravitational-wave sources used to simulate GSpyNetTree's signals.

4 Gravitational-wave detectors

Following Einstein's discovery in 1918 that a linearized version of his equations (that is, in the weak-field limit) admits wave solutions [58, 59], as shown in Chapter 2, major theoretical concerns on the existence and detectability of GWs arose [60]. It was not until 1956 when these were solved by Pirani; who, in 1957, proposed that GWs should carry energy - and thus could be detected [2].

The first notion of a GW detector, developed by Joseph Weber in the 1960's, consisted of aluminum cylinders, with piezoelectric crystals attached on their surface. An incoming GW signal perpendicular to the axis of these resonant bar detectors would generate a longitudinal excitation in the metal which, in turn, would stress the piezoelectric crystals and, thus, generate a voltage [61]. The higher the voltage, the greater the strength of the detected signal, and near-simultaneous excitation of widely separated cylinders would confirm that a real signal had been detected [2]. Even though subsequent efforts (up until the 1990's) increased the sensitivity of these detectors by cooling the bars to temperatures approaching 0 K, no astrophysical signals were detected with this approach [62].

The idea of using optical interferometry to detect GWs was seen independently by various scientists as a promising alternative [63, 64]. Rainer Weiss noted that a Michelson interferometer would be highly effective for sensing the transverse traceless nature of GWs [65]. This Chapter discusses the design and operation mechanism of these detectors, with further descriptions of aLIGO and AdVirgo as the relevant interferometers for this thesis.

4.1. Detection mechanism: the Michelson interferometer

The detection mechanism pictured by Weiss involved a Michelson interferometer, like the one shown in Figure 4.1. Even though aLIGO and AdVirgo are more complex detectors because they need to measure the small amplitude of GWs, a simple Michelson interferometer suffices to understand the detection mechanism of a GW interferometer. These instruments measure the relative time that light beams take to complete a round trip in each of its perpendicular arms [18]. Incoming light from a coherent laser source enters the interferometer and is split by a beam splitter, where half of the light is sent through each arm exactly in phase. If both arms have the same length, L, and they are not perturbed by any external forces, the light will reflect in the mirrors at the end of each arm and will return to the beam splitter exactly in phase. At the beam splitter, the incoming light from each arm is split again, such that half is sent back to the input, and the other half is sent to the output photodetector, where light is finally recombined. If the incoming beams from both arms are in phase, a constructive interference occurs.

Suppose an incoming GW, propagating in the $\hat{\mathbf{z}}$ direction, such that it stretches and compresses space-time perpendicular to its propagation (that is, in the $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ directions). When reaching the interferometer, the passing of the GW causes one arm of the interferometer to

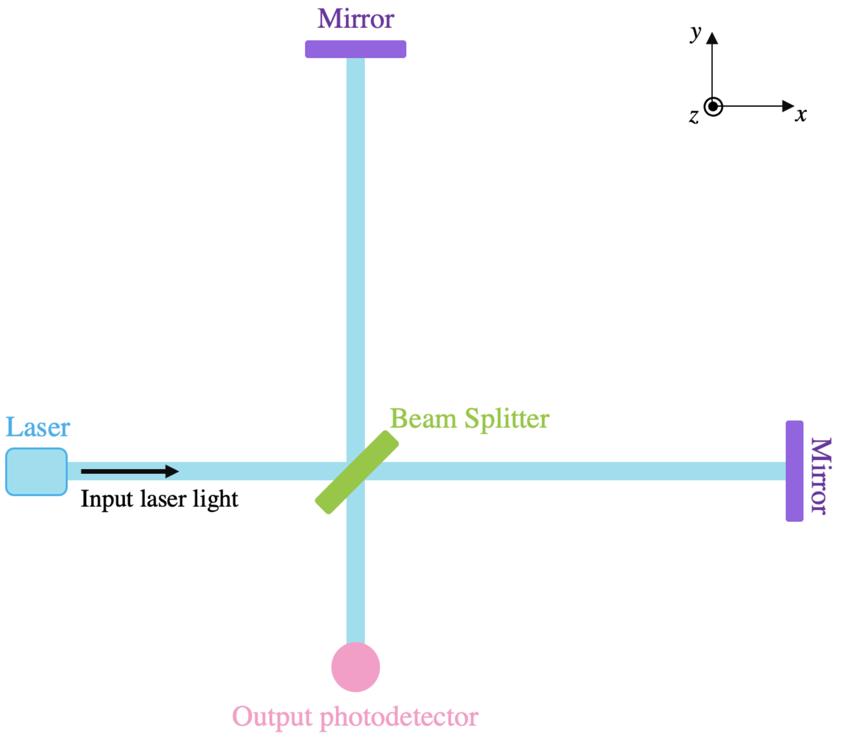


Figure 4.1: Schematic layout of a Michelson interferometer, which consists of a laser, a beam splitter, a series of mirrors, and a photodetector. A relative change in length between the arms of the interferometer is sensed as a relative change in the phase of the light, which is manifested as an interference pattern in the output photodetector.

elongate whilst the other shortens, then vice-versa, back and forth until the GW passes completely through the interferometer. This differential arm motion permits the measurement of the stretching and squeezing of space-time due to a GW, or *strain* $h(t)$, as the difference $\Delta L(t)$ in length L between the two orthogonal directions [2],

$$h(t) = \Delta L(t)/L. \tag{4.1}$$

The time it takes the light to make a round trip through the perpendicular arms of the interferometer changes with the passing of a GW. Along with the elongation/shortening of the interferometer's arms, the effective travel time of light through each perpendicular arm will change accordingly, and the light will return to the beam splitter out of phase. This relative phase difference in the $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ directions is then recombined and measured by the output photodiode, generating a particular interference pattern. The behavior, parameters, and dynamics of astrophysical GW sources, like the ones described in Chapter 3, can then be analyzed from the space-time perturbation generated by the GW and manifested in this phase shift, measured as strain.

To understand how this relative phase difference works, we take the input light from the laser, as shown in Figure 4.1, such that its electric field is given by:

$$\mathbf{E}_{\text{input}} = \mathbf{E_0}\, e^{i(2\pi f t - kx)}, \tag{4.2}$$

where E_0 is the amplitude of the electric field, k is the wave number, f is the frequency of the light beam, and t and x are varying time and position [18]. When encountering the beam splitter, 50% of the light is transmitted to the $x-$arm of the interferometer, with transmission coefficient $T = i/\sqrt{2}$, and 50% is reflected towards the $y-$arm of the detector, with reflection coefficient $R = 1/\sqrt{2}$. In this way, the electric field of the light in the $x-$arm has the form,

$$\mathbf{E}_x = \frac{i}{\sqrt{2}} \mathbf{E_0}\, e^{i(2\pi f t - k_x x)}, \tag{4.3}$$

while the electric field of the light reflected in the $y-$arm is,

$$\mathbf{E}_y = \frac{1}{\sqrt{2}} \mathbf{E_0}\, e^{i(2\pi f t - k_y y)}. \tag{4.4}$$

The wave numbers k_x, k_y depend on the effective travel time of light in the $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ directions, respectively. The beams are then reflected in the mirrors at the end of each arm, which in the context of aLIGO have nearly 100% reflectivity for the laser it uses, and they finally return to the 50/50 beam splitter. The light that is finally recombined and read by the output photodetector will have two components: a reflected component from the $x-$arm and a transmitted component from the $y-$arm, where $R = 1/\sqrt{2}$ and $T = i/\sqrt{2}$ as before. The light will have also traveled in its round trip a distance of $2L_x$ in the $x-$arm and $2L_y$ in the $y-$arm, where L_x and L_y are the lengths of the perpendicular arms of the interferometers. Note that, if the interferometer is perturbed by an external event (say, a GW, or a terrestrial source), $L_x \neq L_y$. The total electric field at the output after combining the beams is,

$$\begin{aligned}\mathbf{E}_{\text{output}} &= \frac{i}{2} \mathbf{E_0}\, e^{i(2\pi f t - 2k_x L_x)} + \frac{i}{2} \mathbf{E_0}\, e^{i(2\pi f t - 2k_y L_y)} = \frac{i}{2} \mathbf{E_0}\, e^{i(2\pi f t - 2k_x L_x - 2k_y L_y)}, \\ &= \mathbf{E_0}\, i e^{i(2\pi f t - 2k_x L_x - 2k_y L_y)} \cos{(k_x L_x - k_y L_y)}. \end{aligned} \tag{4.5}$$

The argument $k_x L_x - k_y L_y = \Delta\phi$ is the difference in phase that results from the light completing a roundtrip in both arms of the detector. In case both beams are in phase (i.e. $\Delta\phi = 0$), light with electric field of amplitude E_0 will get to the output photodetector. If, however, $\Delta\phi = n\pi/2$ with $n \in \mathbb{Z}$, no light will reach the photodetector. However, it is not the electric field what is measured at this point, but the power. Assuming the laser light has power P_{input} and given that the power of a beam of light is given by E^2,

$$P_{\text{output}} = E_0^2 \cos^2(\Delta\phi) = P_{\text{input}} \cos^2(\Delta\phi) = \frac{1}{2} P_{\text{input}} [1 + \cos{(2\Delta\phi)}]. \tag{4.6}$$

In this way, variations in the length of any arm of the interferometer result in a variation in the power measured at the photodetector. Generally speaking, the longer the detector arms, the greater the induced phase difference; and the higher the laser power (i.e. P_{input}), the more intense the signal [2]. However, in practice, the detectors must be more robust, due to the small amplitude of gravitational-wave signals, the stiffness of space-time, and the noise sources that affect GW detections.

4.1.1. How do interferometers measure gravitational-waves?

Having studied the mechanism of Michelson interferometers to measure the difference in phase using interfered laser light, we turn to study how they are used by aLIGO and AdVirgo to detect GWs and measure GW strain. Picturing the same interferometer as in Figure 4.1, we use a coordinate system compatible with the TT gauge chosen in Chapter 2. In this gauge, coordinates are marked by the world lines of free-falling test masses [18]. Even when a GW passes, the coordinates of freely falling objects do not change [20]. We may choose the interferometer mirrors to be these test masses. Even though it is true that, in Earth-based detectors like LIGO, Virgo, and KAGRA, mirrors are not freely falling (in fact, gravity is compensated by the suspensions), the forces they are subject to are static compared to the frequency of GWs. It is important to highlight that non-static forces due to the coupling with the environment (such as thermal noise) may affect the mirrors. They will be the cause of background noise and *glitches*, as described in Chapters 5 and 6.

We may set the origin of our coordinate system at the beam splitter, such that the position of the x arm mirror of length L is at $(L_x, 0)$ and the one of the y arm mirror, at $(0, L_y)$. We start our study by understanding the interaction of a "plus" polarized GW with the interferometer, incoming in the $\hat{z}$ direction. The interaction of GWs with arbitrary direction and polarization will follow.

The physical effect of a GW passing through an interferometer is that it affects the propagation of light between the fixed points defined by the mirrors at the ends of each perpendicular arm. This is a consequence of GWs being transverse traceless waves. A wave traveling in the $\hat{z}$ direction causes measurable relative motion in the $\hat{x}$ and $\hat{y}$ directions between sets of neighboring free masses. The motions in these perpendicular directions are equal and opposite [19].

In the $z = 0$ plane of the detector, our GW has the form of equation (2.15), $h_+(t) = h_0 \cos(\omega_{\mathrm{GW}} t)$, with $\omega_{\mathrm{GW}} = 2\pi f_{\mathrm{GW}}$. The spacetime interval in the TT-gauge is given by,

$$ds^2 = (\eta_{\mu\nu} + h_{\mu\nu}) dx^\mu dx^\nu = -c^2 dt^2 + [1 + h_+(t)] dx^2 + [1 - h_+(t)] dy^2 + dz^2. \tag{4.7}$$

Being massless particles, photons travel along null geodesics (i.e. $ds^2 = 0$). In this way, to first order in h_0, the light in the $x-$arm follows:

$$dx = \pm c\, dt \left[1 - \frac{1}{2} h_+(t)\right], \tag{4.8}$$

where the plus sign holds for the travel from the beam-splitter to the mirror, and the minus sign for the trip back to the beam-splitter [20]. Consider a photon that departs at a time $t = t_0$ from the beam-splitter towards the $x-$arm. When it reaches the mirror, at $x = L_x$, at $t = t_1$, we get:

$$L_x = c(t_1 - t_0) - \frac{c}{2} \int_{t_0}^{t_1} dt' h_+(t'), \tag{4.9}$$

where we have integrated using the plus sign. The photon travels back through the $x-$arm until it reaches the beam-splitter again, at $t = t_2$. Integrating equation (4.8) with the negative sign between $x = L_x$ and $x = 0$,

$$L_x = c(t_2 - t_1) - \frac{c}{2}\int_{t_1}^{t_2} dt' h_+(t') \tag{4.10}$$

Adding equations (4.9) and (4.10),

$$t_2 - t_0 = \frac{2L_x}{c} + \frac{1}{2}\int_{t_1}^{t_2} dt' h_+(t'), \tag{4.11}$$

which means that the time of arrival t_2 of light after a round trip through the $x-$arm of the interferometer is given by $t_0 + 2L_x/c$, plus a correction of order h_0. We can replace the upper limit of the integral for $t_2 = t_0 + 2L_x/c$, as the integrand is already $O(h_0)$ and we are neglecting terms $O(h^2)$ or higher [20]. This gives us[1]:

$$\tau_x \equiv t_2 - t_0 = \frac{2L_x}{c} + \frac{L_x h_0}{c}\mathrm{sinc}(\omega_{\mathrm{GW}} L_x/c)\cos\left[\omega_{\mathrm{GW}}\left(t_0 + \frac{L_x}{c}\right)\right], \tag{4.12}$$

for the $x-$arm and,

$$\tau_y \equiv t_2 - t_0 = \frac{2L_y}{c} - \frac{L_y h_0}{c}\mathrm{sinc}(\omega_{\mathrm{GW}} L_y/c)\cos\left[\omega_{\mathrm{GW}}\left(t_0 + \frac{L_y}{c}\right)\right], \tag{4.13}$$

for the $y-$arm where

$$\mathrm{sinc}(\omega_{GW} L_\alpha/c) \equiv \frac{\sin(\omega_{\mathrm{GW}} L_\alpha/c)}{\omega_{\mathrm{GW}} L_\alpha/c}, \tag{4.14}$$

with $\alpha = x, y$. One difference between equations (4.12) and (4.13) is the negative sign for the second term of τ_y, which comes from equation (4.7). However, these equations also implicitly consider $L_x \neq L_y$. In a GW interferometer, these two perpendicular arms measure almost the same (they only differ slightly because of the Schnupp asymmetry, a small asymmetry between the arms [20]), and we can replace L_x and L_y with $L \equiv (L_x + L_y)/2$.

Using L instead of L_x and L_y in equations (4.12) and (4.13), we get that the travel time difference of the light between the two perpendicular arms, $\Delta\tau \equiv \tau_x - \tau_y$, is given by:

$$\Delta\tau = \frac{2Lh_0}{c}\mathrm{sinc}(\omega_{\mathrm{GW}} L/c)\cos\left[\omega_{\mathrm{GW}}\left(t_0 + \frac{L}{c}\right)\right]. \tag{4.15}$$

This time difference between the perpendicular arms induces a phase shift that is measured by GW interferometers. We can calculate this phase shift using the travel time difference, $\Delta\tau$, and the period of oscillation of the light,

$$\Delta\phi = \frac{2\pi c}{\lambda_L}\Delta\tau = \omega_L \Delta\tau, \tag{4.16}$$

with ω_L the frequency of the light traveling through the detector's arms of length L, and λ_L its wavelength. In the case of LIGO, $\lambda_L = 1064$ nm. Using equation (4.15), the total phase induced

---

[1] The full derivation is given by Maggiore [20].

by GWs in the Michelson interferometer is,

$$\Delta\phi = \frac{2Lh_0\omega_L}{c}\text{sinc}(\omega_{\text{GW}}L/c)\cos\left[\omega_{\text{GW}}\left(t_0 + \frac{L}{c}\right)\right]. \tag{4.17}$$

It is important to highlight that, in the case of LIGO and Virgo, $L \ll \lambda_{\text{GW}}$; that is, the length of the arms of the interferometer is much shorter than the wavelength of the GW. In this way,

$$\frac{\omega_{\text{GW}}L}{c} \ll 1. \tag{4.18}$$

With this approximation, $\text{sinc}(\omega_{\text{GW}}L/c) \to 1$, and the time difference between the two perpendicular arms,

$$\Delta\tau_{\text{approx}} \approx \frac{2Lh_0}{c}\cos\left[\omega_{\text{GW}}\left(t_0 + \frac{L}{c}\right)\right]. \tag{4.19}$$

Similarly, the phase shift induced by $\Delta\tau_{\text{approx}}$ is now,

$$\Delta\phi_{\text{approx}} \approx \frac{2Lh_0\omega_L}{c}\cos\left[\omega_{\text{GW}}\left(t_0 + \frac{L}{c}\right)\right]. \tag{4.20}$$

Using equation (2.15), $\Delta\phi_{\text{approx}}$ can be rewritten as

$$\Delta\phi_{\text{approx}} \approx \frac{2L\omega_L}{c}h\left(t - \frac{L}{c}\right). \tag{4.21}$$

Up until now, we have described the response of the detector to a plus-polarized GW propagating in $\hat{\mathbf{z}}$. To understand the response of the detectors to GWs coming in an arbitrary direction, we use the antenna pattern, which describes the relative sensitivity of the detectors in different directions [66]. The antenna pattern is given in terms of the antenna pattern functions, F_+ and $F_\times$, instead of $\Delta\phi$. This is done to avoid the confusion between the induced phase in the interferometers and the direction of propagation of GWs with respect to the detector, given in terms of the azimuthal angle ϕ, polar angle θ, and polarization angle ψ, as seen in Figure 4.2.

Using the antenna pattern functions, the strain induced into an interferometer by a GW coming from an arbitrary direction is given by a linear combination of F_+ and $F_\times$ multiplied by the respective possible GW polarization, plus and cross:

$$h(t) = F_+(\theta,\phi,\psi)h_+(t) + F_\times(\theta,\phi,\psi)h_\times(t), \tag{4.22}$$

where

$$F_+ = \frac{1}{2}\left(1+\cos^2\theta\right)\cos 2\phi\cos 2\psi - \cos\theta\sin 2\phi\sin 2\psi, \tag{4.23}$$

and

$$F_\times = \frac{1}{2}\left(1+\cos^2\theta\right)\cos 2\phi\sin 2\psi + \cos\theta\sin 2\phi\cos 2\psi. \tag{4.24}$$

These functions are defined for GWs with polarization angle ψ, coming from some arbitrary direction relative to the detector's axes described by the spherical coordinates θ and ϕ [23]. Note that the antenna pattern functions relate the *sky position* of the source (i.e., the direction of gravitational-wave propagation in (θ, ϕ) with respect to the detector frame), and the polarization

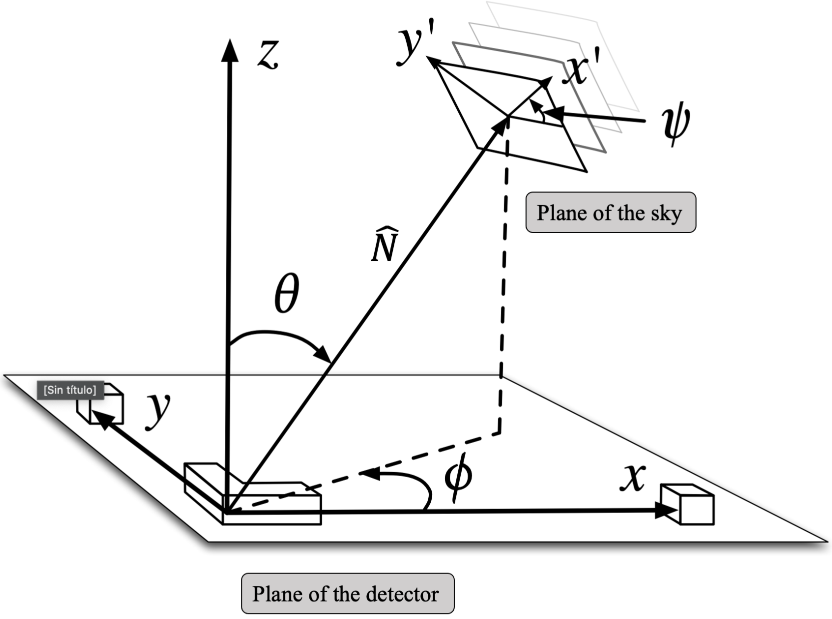


Figure 4.2: Schematic of the coordinates θ, ϕ, and ψ that describe the position and orientation of the gravitational-wave source in the plane of the sky relative to the frame of the detector. The perpendicular arms of the detector are defined as usual, oriented along the $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ axes. The GW, coming from an arbitrary direction (θ, ϕ) with respect to the detector, is rotated by the polarization angle, ϕ. This angle defines the orientation of the gravitational-wave source relative to the projection of the detector plane into the sky plane, along vector $\hat{N}$. (Adapted from [67]).

of the GW (given by the angle ψ, defined by the rotation of the coordinate frame of the source relative to the detector axes) [66].

As the strain $h(t)$ from equation (4.22) is a linear combination of the plus and cross polarizations of GWs that depends on the geometry of the interferometer and the direction of the source, a single detector cannot measure both polarizations at the same time. Moreover, as the polarization angle depends on the internal orientations of the source, its direction of arrival is usually unrelated to its polarization. For this reason, it is useful to average over the polarization angle ψ [23]. The rms response function (or antenna pattern) of the interferometer is given by:

$$\overline{F}^2(\theta, \phi) = \int \left(F_+^2 + F_\times^2\right) d\psi = \frac{1}{4}\left(1 + \cos^2\theta\right)^2 \cos^2 2\phi + \cos^2\theta \sin^2 2\phi. \tag{4.25}$$

Figure 4.3 shows the rms antenna pattern for an interferometric detector, with the detector in the plane $z = 0$ (i.e. at $\theta = \pi/2$) and its arms oriented along the x and y axes.

The sensitivity is strongest when incoming GWs are perpendicular to the plane of the detector, and there is a significant nonzero response for most directions. The only ‘blind’ regions for the GW detectors are along the bisector between the two perpendicular arms at $z = 0$ [18] (that is, at $\phi = \pi/4$ and directions differing from this by multiples of $\pi/2$) [23]. In these directions, the changes in the length of the two arms of the interferometer are always equal, and no strain is measured (i.e., from equation (4.1), $\Delta L = 0$, and $h(t) = 0$).

As we cannot assure that all GWs are coming perpendicular to the detectors, it is important

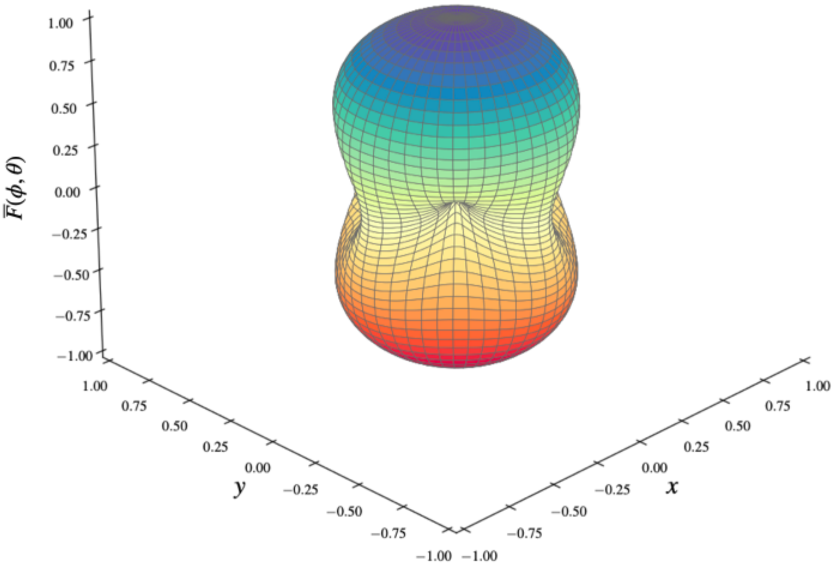


Figure 4.3: Antenna pattern (equation (4.25)) of a gravitational-wave interferometer, averaged over the polarization angle ψ, showing its sensitivity to incoming GWs from different directions in the sky. The arms of the interferometer are oriented along the x and y axes in the plane $z = 0$. The response of the detector, $\overline{F}(\theta, \phi)$, for incoming waves from a particular direction in the sky is proportional to the distance to the point on the antenna pattern in that direction [23]. Sensitivity is strongest for GWs incoming in the $\hat{\mathbf{z}}$ direction, and is zero in the $z = 0$ plane, at $\phi = \pi/4$ and its $\pi/2$ multiples.

to make our searches robust to GWs from different locations and orientations in the sky. The O4-era version of GSpyNetTree, described in Chapter 12, includes simulations of GWs coming from all directions.

Note that due to its broad sky range and significant sensitivity towards most directions of the sky, it is very unlikely that a GW event is missed, due to the direction of propagation of GWs relative to each interferometer in the global network. However, different from conventional telescopes, localizing an event in the sky in such a broad region is very challenging, which is why various interferometers (in different locations of the planet) are needed. The following section describes the GW interferometers that will be operating in the upcoming Observing Run, O4.

4.2. The global interferometer network

The global interferometer network for the upcoming Observing Run (O4), which will start in late May 2023, will have four operating interferometers, located in different parts of the world, as shown in Figure 4.4. These are LIGO Hanford, LIGO Livingston, Virgo, and KAGRA. In the past, GEO600 was part of the global network. LIGO-India, the Cosmic Explorer (in the US), and the Einstein Telescope (in Europe) will join the current GW detector network in the future.

LIGO stands for Laser Interferometer Gravitational-wave Observatory. There are two LIGO interferometers in the United States, each with two perpendicular 4-km long arms. These are located in Hanford, Washington, and Livingston, Louisiana. An in-depth description of the aLIGO interferometers is given in Section 4.3. Virgo is a French-Italian Observatory located

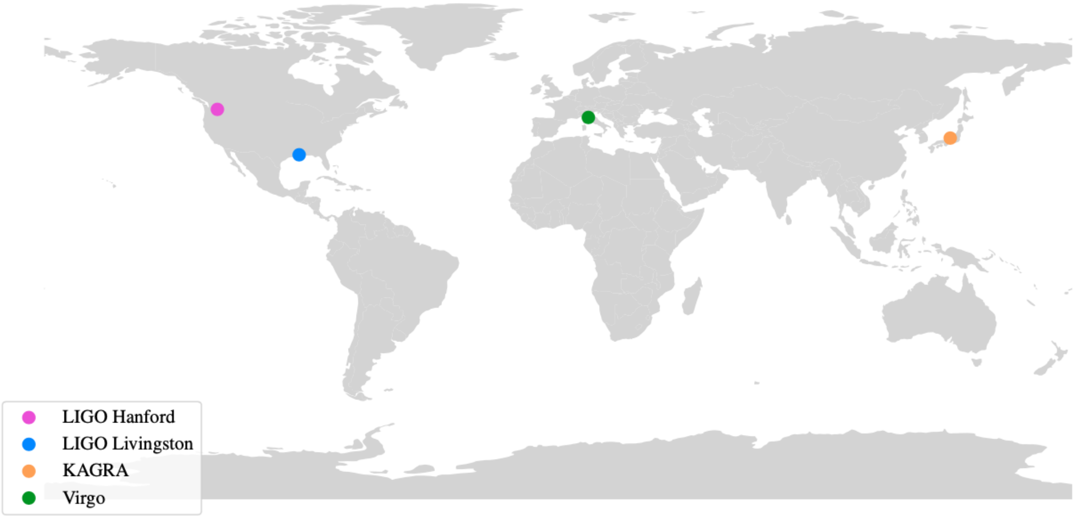


Figure 4.4: Global ground-based gravitational-wave detector network for the fourth LIGO-Virgo-KAGRA Observing Run. This network is comprised of the two US LIGO sites (located in Hanford, Washington, and Livingston, Louisiana), Virgo (in Italy), and KAGRA in Japan. (Locations of the detectors obtained from [68]).

near Pisa, Italy. Different from LIGO, it has 3-km long perpendicular arms, but with very similar instrumentation to that of LIGO, as described in Section 4.4. Unlike its three sister observatories, KAGRA, the Kamioka Gravitational-wave Detector, is an underground cryogenic interferometer located in the tunnels of the Kamioka mine in Japan. A brief description of KAGRA is given in Section 4.4.

4.3. The Advanced LIGO (aLIGO) detectors

As explained in the previous section, the aLIGO detectors are, in their most simplified version, Michelson-Morley interferometers. However, sophisticated instrumentation is needed to convert them into stable, reliable, and sensitive detectors capable of detecting gravitational-waves. This section describes the components and instrumentation of the current version of the more realistic, aLIGO detectors shown in Figure 4.5, based on [2]. A more detailed discussion of these components and subsystems is given in [69].

One of the most important components of aLIGO is the **Pre-Stabilized Laser (PSL)**, displayed in the box at the top-left of Figure 4.5. The detectors use an Nd:YAG 1064 nm infrared laser, a neodymium-doped yttrium aluminum garnet solid-state laser crystal that is capable of emitting a beam of $\sim$ 100 W power [2]. This is done to reduce the quantum photon shot noise, described in Subsection 5.1.1. Thanks to a pre-mode cleaner cavity, a 10 cm-diameter Gaussian beam (TEM_{00}) is used[2]. Keeping the beam size small is crucial for mirror fabrication purposes.

It is important to highlight that the intensity of the laser needs to be kept as constant as possible. Small fluctuations in the pressure on the mirrors vary the power read at the output

---

[2] The Transverse Electromagnetic Mode (TEM) is the mode of a waveguide that has no electric and magnetic fields in the direction of propagation. TEM_{00} is the lowest order mode.

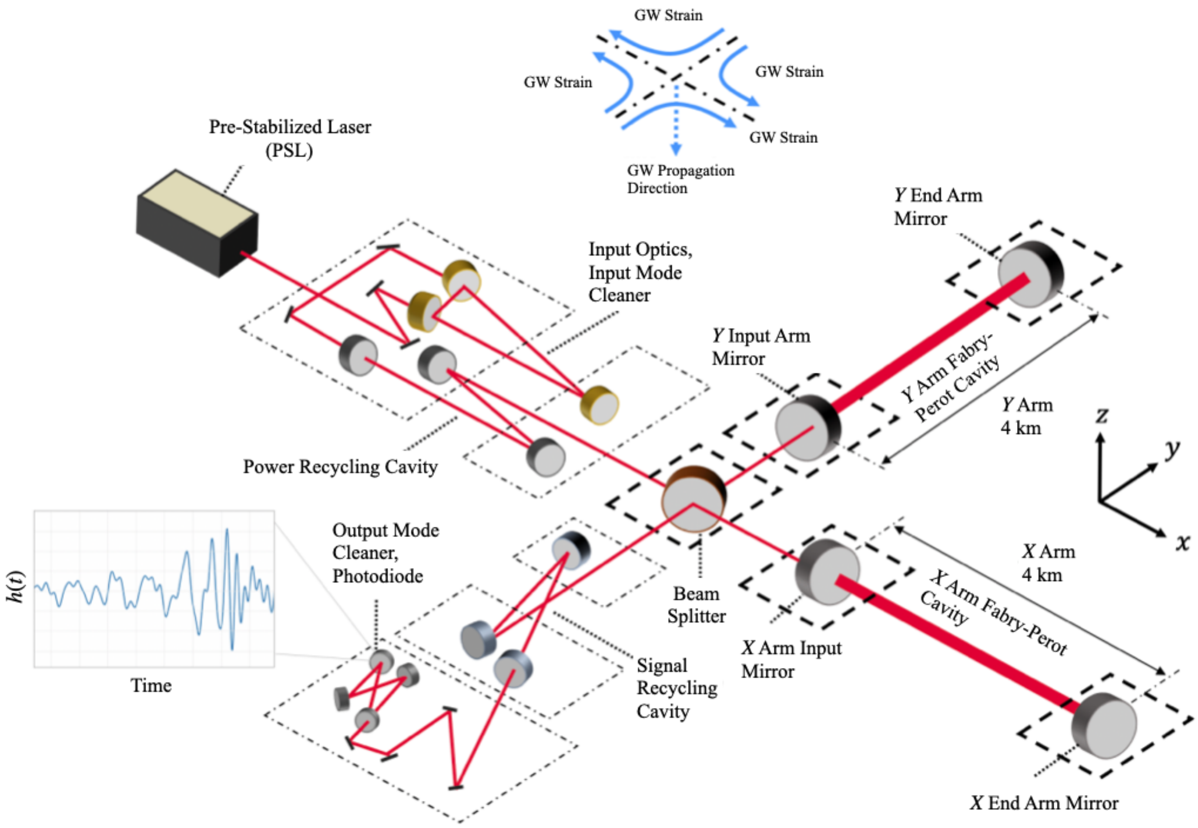


Figure 4.5: Realistic schematic layout of the aLIGO detectors. The main optical components are shown in gray, while the path of the light beam is shown in red. Note that the path of the beam is thicker within the Fabry-Perot cavities; this is a result of the increased intensity of the light in the interferometer's arms. Several components from the basic Michelson interferometer, such as the input laser, the mirrors at the end of the arms, the beam splitter, and the Output Photodiode are shown in the image, with additional instrumentation that increases aLIGO's sensitivity significantly. (Adapted from [2]).

photodiode and may mask GW signals. For this reason, a series of advanced sensors within the PSL are used to maintain laser stability.

Just after the laser beam exits the PSL, it encounters the **Input Optics**, a collection of three curved mirrors (displayed in gold in Figure 4.5) that aim to focus the beam so that it propagates through the 4 km arms without spreading out. These mirrors are also called **Input Mode Cleaners (IMC)**. Besides stabilizing the beam, the IMC removes higher-order modes in the input beam [70] and reduces noise due to variations in the beam's wavelength. It is crucial that the frequency of the laser beam is correctly matched to the natural frequency of the x and y arm cavities.

Once the laser beam passes the Input Optics, it goes to the 50/50 Beam Splitter, where half of the beam goes to the x arm and the other half, to the y arm. These arm cavities are made up of two mirrors, one near the beam splitter and the other at the end of each arm, which form a **Fabry-Perot cavity**. These optical systems are resonant for light of a particular wavelength, which is why it is important that the frequency of the laser beam is matched to that of the cavities. When this happens, the resonance of the cavity is excited, and the power of the light inside the cavity is approximately 100 times larger than that of the original beam [2]. This is the reason why the red beam in both Fabry-Perot cavities of Figure 4.5 is thicker than in other parts of the interferometer.

The **mirrors** of aLIGO are one of its fundamental parts, and they are engineered to mitigate several noise sources (e.g., thermal noise)[3]. LIGO mirrors are made of fused silica, with low hydroxide content, to mitigate infrared absorption which would otherwise heat up the mirrors and affect their morphology [71]. They also have several layers of optical coatings which make the detector more sensitive. As an example, the mirrors at the ends of the arms are almost perfectly reflective - they only absorb one out of 3.3 million photons [71]. The aLIGO mirrors are 34 cm in diameter, 20 cm thick, and weigh approximately 40 kg. They are usually called test masses, as they respond to the gravity of other bodies (i.e., the GWs) while having negligible gravity of their own [2]. These mirrors are supported by sophisticated pendulums, made up of fused-silica suspension fibers, as described in Subsection 5.1.2.

After resonating in the Fabry-Perot cavities, the light goes back to the beam splitter. Some of it is sent to the output photodiode, and another part will return to the input port. In order to benefit from this unused light and increase the intensity of the light that arrives at the beam splitter, aLIGO uses a **Power Recycling Mirror**, which forms a new cavity that is resonant to the wavelength of the beam [72]. This way, the light that is not sent to the Output Photodiode is recycled in the interferometer, increasing the detector sensitivity [2].

The light that goes toward the Output Photodiode arrives at a **Single Recycling Mirror**, which works as another resonant cavity (but now for GW information that is 'carried by' the laser beam that arrives at the Output port). This cavity boosts the intensity of the measured GW signal in a desired frequency range, increasing the detector sensitivity [2]. Before arriving at the Photodiode, the beam passes through the **Output Mode Cleaner** where higher order modes of the beam resulting from misalignment of the optic systems are removed. This is done with a cavity only resonant to the TEM_{00} mode. If these modes were not removed, excess power would be read at the **Output Photodiode**, masking true GW signals [72]. Once there, the signal is finally read, and the changes in laser beam power in time are converted to strain $h(t)$, as shown in the bottom left of Figure 4.5.

It is important to highlight that aLIGO has many more calibrators, sensors, and mechanisms (e.g., suspensions) to mitigate noise sources and improve detector sensitivity. Their usage and importance are discussed in Chapter 5.

4.4. The Advanced Virgo (AdVirgo) and KAGRA detectors

As aLIGO and AdVirgo have had very similar commissioning, their instrumentation and detection mechanics are alike. Virgo's arms are also Fabry-Perot cavities, an Input Cleaner Mode is used to select a Gaussian mode, and a Power Recycling Mirror increases the intensity of the laser beam. Compared to aLIGO, there are a few key differences, such as the length of its arms (3 km) and 2 kg heavier mirrors. Before O4, just as in aLIGO, Virgo aims to implement signal recycling. In addition to that, Virgo also aims to implement the usage of

---

[3] A deeper description of the types of noise sources of LIGO, as well as the ways to mitigate them, is given in Chapter 5

frequency-dependent squeezing and a new fiber laser (an upgrade from the O3 master-slave Nd:YAG-Nd:YOV$_4$ system[4]) [68].

KAGRA, on the other hand, differs highly from aLIGO and AdVirgo in a few aspects. In order to minimize seismic noise, KAGRA is located 200 m underground (whilst its three sister observatories are all at the ground level). Additionally, KAGRA uses sapphire mirrors as test masses. These operate at cryogenic temperatures (i.e, $T \sim 20$ K), reducing thermal noise [68].

Apart from these notable differences, KAGRA uses Fabry-Perot cavities in its 3 km perpendicular arms, Power and Signal Recycling Cavities, and Input and Output Mode Cleaners just as aLIGO. After two Engineering Runs (ER) and commissioning efforts, KAGRA joined the global interferometer network in 2020 [68]. The next section explains the advantages of having several detectors in a global interferometer network.

4.5. The benefits of a global interferometer network

Having several GW detectors in different locations and with different orientations is crucial to determine the sky position of a given GW event. Even though the four detectors have different sensitivities due to their instrumental differences, they complement each other to detect GW events at varying distances in our Universe. One way to measure the sensitivity of a detector is using the BNS range[5], which is the distance at which a typical binary of two 1.4 $M_\odot$ neutron stars could be detected when averaging over all directions [69]. Figure 4.6 shows the BNS ranges for a single-detector signal-to-noise ratio (SNR) threshold of 8 for LIGO, Virgo, and KAGRA during the Observing Runs 1, 2, and 3 (O1, O2, and O3) and the expected BNS range for Observing Run 4 (O4). Note that, even though LIGO has the highest BNS range, having Virgo and KAGRA in the network is fundamental to make confident predictions and increase sensitivity.

With only the LIGO and Virgo detectors, it is already possible to constrain the likely sky location of GW events. Take GW170817, for instance. As explained in Section 3.1.4 the event was constrained to a 28 deg^2 sky region using the three detectors. At design sensitivity, LIGO and Virgo are expected to resolve the sky position of a BNS GW event with a network SNR of 12 (more on this in the next Section) to a 5 deg^2 region for 8% of all events [73]. This statistic improves considerably with the inclusion of KAGRA in the global interferometer network, as well as the planned IndIGO (LIGO-India) detector. It is also important to highlight that, as interferometers have a higher BNS range, more events become detectable. This increases the amount of GW events. As a matter of fact, the expected detection rate of O4 is approximately one GW candidate per day [74]. This higher event rate requires increased levels of automation for candidate event validation, which is where tools like GSpyNetTree are fundamental.

4.6. Searching for gravitational-waves

Up until now, we have studied how interferometers measure strain $h(t)$ and the sophisticated instrumentation used for these measurements. It is now time to understand how GW signals

---

[4] The Nd:YOV$_4$ laser stands for Neodymium-doped yttrium orthovanadate.

[5] The BNS range as a measure of sensitivity is more thoroughly explained in Chapter 5.

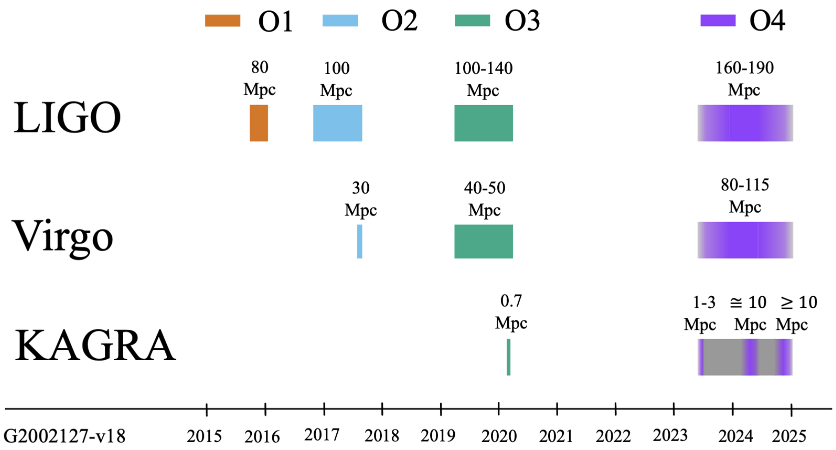


Figure 4.6: BNS range, in Mpc, for the LIGO, Virgo, and KAGRA detectors for the first four observing runs. As mentioned in the previous section, KAGRA joined the global interferometer network in 2020 during O3. While LIGO has the highest BNS range, having Virgo and KAGRA in the global interferometer network is fundamental to have more confident predictions, to localize GW events in the sky map within a small credible region, and to be sensitive to a broader distance range. (Adapted from [73]).

are found in the vast amount of (noisy) data that the interferometers generate.

GW signals interact very weakly with the interstellar medium before arriving to Earth, as opposed to the light from EM signals. This gives the possibility of using matched filtering to look for GWs in the data. Using this technique, the waveforms predicted from General Relativity can be confronted with the measurements performed by the interferometer [2]. Matched filtering allows the recovery of a known signal (the GR waveform, or template) that is embedded in the background noise. By comparing the data from the detector, $s(f)$, with the waveform template, $h(f)$, (with f the frequency) we can calculate the signal-to-noise ratio (SNR), ρ, as [75]:

$$\rho^2 = \frac{||\langle s, h\rangle||^2}{\langle h, h\rangle}, \tag{4.26}$$

where the inner product is given by:

$$\langle a, b\rangle = 4\,\mathrm{Re}\left(\int_0^\infty \frac{a(f)b^*(f)}{S_n(f)}\, e^{2\pi i f t}\, df\right). \tag{4.27}$$

Note that the product of $a(f)$ and $b^*(f)$ is normalized to the power spectral density (PSD, view Chapter 5) of the noise $S_n(f)$, and is integrated over all frequencies. Using the Fourier transform, the calculation can be translated to the time domain.

In case a GW is detected, we would get a value of $\rho \gtrsim 8$ that could not otherwise result from Gaussian noise fluctuation (LIGO's noise is mainly Gaussian; however, there might be non-Gaussian fluctuations, known as glitches, which might mask true GW signals as described in Chapter 6). CBC searches use this matched filtering technique to find signals ranging from 1 $M_\odot$ to 100 $M_\odot$.

An additional benefit of the global interferometer network is that it further permits the confirmation of a true GW event. All detectors should detect the same GW event almost simultaneously, only differing by a small difference in time of detection due to the propagation velocity of GWs, c, if the event is within their SNR range. If an event was detected by a single interferometer, it is possible that the triggered GW candidate was not astrophysical in origin, but terrestrial. This is why using the network SNR[6], ρ_{net}, for n detectors [2],

$$\rho_{net} = \sqrt{\rho_1^2 + \rho_2^2 + \cdots + \rho_n^2}, \tag{4.28}$$

is fundamental to evaluate the significance of the trigger. Comparing the network's SNR to the rate at which noise sources produce similar triggers, gives the False Alarm Rate (FAR) [2].

It is important to note that, while aLIGO and AdVirgo also perform searches for bursts (using wavelets), CWs (also with matched filtering), and stochastic gravitational-wave signals (using coherence methods), these methods differ considerably from CBC searches, and are out of the scope of this thesis.

The next chapter discusses how data quality is measured (and assured) in aLIGO, and which noise sources affect GW searches, along with a description of the mitigation of many of them.

---

[6] Section 6.1 discusses a redefinition of this metric to reduce false triggers due to glitches.

5 Assuring LIGO-Virgo data quality and sensitivity: noise sources and mitigation

In order to confidently predict gravitational-waves, aLIGO needs to be very sensitive. The sensitivity of the aLIGO and Virgo detectors is measured in two ways. The BNS range, as explained before, dictates how far away a binary neutron star merger with 1.4 $M_\odot$ component masses can be detected, averaging over all sky locations and orientations [2, 69]. Figure 4.6 shows the previous (O1, O2, and O3) and expected (O4) BNS ranges for all detectors in the current global interferometer network. This BNS range is calculated from the Power Spectral Density, PSD, which is also used to characterize the detectors. Transforming the strain data $h(t)$ (shown in Figure 3.5 and in the bottom-left inset of Figure 4.5) from the time domain to the frequency domain via the Fourier transform, the power spectrum characterizes the behavior of the detector for different frequencies [2]. A plot of the ASD (Amplitude Spectral Density, the square-root of the PSD) of the LIGO and Virgo detectors during the first half of the third Observing Run (O3a) is shown in Figure 5.1.

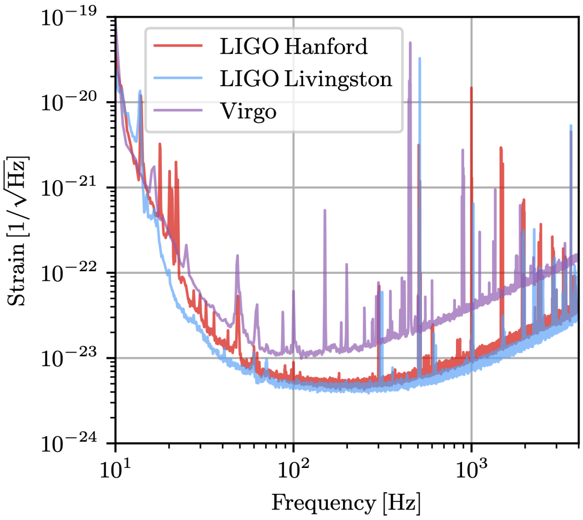


Figure 5.1: Amplitude Spectral Density (ASD), given in units of $1/\sqrt{\text{Hz}}$, of LIGO Hanford (in red), LIGO Livingston (in blue), and Virgo (in violet), during the first half of O3 (O3a). Note that the most sensitive detector is LIGO Livingston, followed by LIGO Hanford. Many of the peaks in the ASD at particular frequencies are associated to certain noise sources, as described in the text. (Reproduced from [7]).

Note that there are several peaks at determined frequencies in the ASD data for the three detectors. At such frequencies, there is a high amplitude sinusoidal component of the noise; and most of these peaks have a known origin. For instance, the lines at 500 Hz and harmonics correspond to mechanical resonances of the fused silica fibers used to suspend the mirrors in the arms of the detectors [2]. The lines at 60 Hz and harmonics (for LIGO, 50 Hz for Virgo), appear due to the AC power grid in the US and Italy, respectively. During the previous observing runs, these lines were removed in high-latency searches, while it is expected that a non-linear subtraction of these power artifacts occurs for aLIGO in low-latency for O4[1]. GSpyNetTree is robust to background noise with and without such lines.

Besides the ones described above, LIGO is subject to a plethora of other noise sources that limit detector sensitivity. The next section describes the most common noise sources in aLIGO and how they are mitigated.

5.1. Principal noise sources and mitigation

Due to the sensitivity that GW detectors are designed to have, several sources of broadband noise impact the interferometers and limit their sensitivity to astrophysical events. Many of the sources do not vary in time (i.e., are stationary) and are Gaussian in nature. However, some others do not satisfy these conditions and impact considerably the signals that can be detected. The most relevant noise sources in aLIGO are described in the following subsections.

Figure 5.2 shows the dominant noise sources in aLIGO, which will be described in the following subsections.

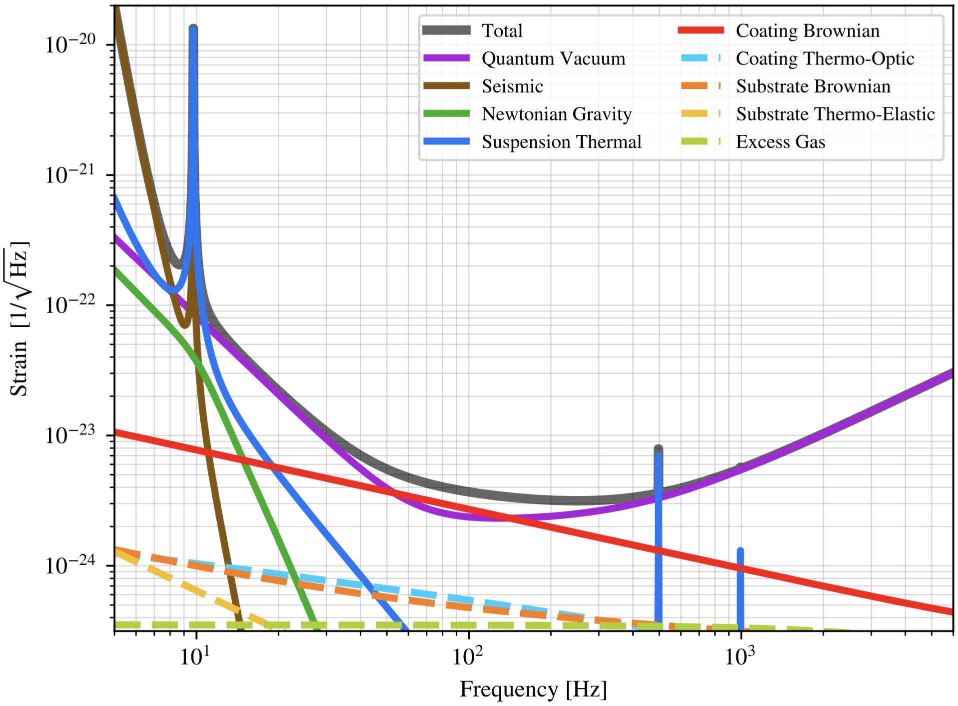


Figure 5.2: Amplitude Spectral Density (ASD) of the principal noise sources of aLIGO as a function of frequency, explained in Subsections 5.1.1 through 5.1.4. The ‘Excess gas’ noise, in pale green, is due to residual gas molecules in the arms of the interferometer, and the ‘Coating Brownian’ noise is the fundamental contribution of thermal noise [2], described in Subsection 5.1.2. At low frequencies, the ‘Newtonian’ (in green), ‘Suspension Thermal’ (in blue), and ‘Seismic’ (in brown) noise sources do not permit GW detection below 10 Hz. At higher frequencies, the detection sensitivity is limited by the ‘Quantum Vacuum’ noise, shown in purple. (Figure generated using the Gravitational Wave Interferometer Noise Calculator, `pygwinc` [76]).

---

[1] The difference between high and low-latency searches lies on the time it takes to process the data. BNS searches, for instance, always need to be done in low-latency to allow for a rapid EM response.

5.1.1. Quantum Noise

The Quantum Noise, or 'Quantum Vacuum' noise as labeled in Figure 5.2, has two main contributions: radiation pressure at low frequencies, and quantum shot noise at high frequencies. It is the contribution of the latter what limits GW detection at high frequencies.

Shot noise arises from the impossibility of measuring light intensity exactly, therefore limiting the measurement in the difference in the length of the arms, ΔL, and the strain $h(t)$. As light intensity fluctuates randomly, so does ΔL and $h(t)$. This comes from the counting error of the photons of the beam in the photodiode - which follows a Poisson distribution, and is due to the particle nature of light. This shot noise is given by [77]:

$$h_{\text{shot}}(P) = \frac{1}{NL}\sqrt{\frac{\hbar c\lambda}{2\pi\eta P}}, \tag{5.1}$$

with λ the wavelength of the light (1064 nm in the case of LIGO), η de quantum efficiency of the photodiode ($\eta \sim 1$), P the input power of the light into the interferometer, and NL the effective arm length, with L the arm length (4 km for aLIGO, 3 km for AdVirgo) and N the number of roundtrips a photon completes within the Fabry-Perot cavities of the arms.

Note that, by increasing laser power, the overall number of photons augments as well, and the fractional error on the measurement of the power, P, will decrease, lowering the relative shot noise, $h_{\text{shot}}(P)$. Nevertheless, laser power cannot be arbitrarily increased due to the **radiation pressure noise**.

Each photon arriving at the mirror of each arm exerts a force over it and transfers momentum to that mirror [2]. As the power increases, pressure fluctuations on the test masses augment, due to the number of photon collisions. This effect generates an additional motion of the test masses that potentially masks GW signals. The radiation pressure noise is given by [77]:

$$h_{\text{rad}}(f,P) = \frac{N}{mf^2L}\sqrt{\frac{2\hbar P}{\pi^3 c\lambda}}, \tag{5.2}$$

with f the frequency and m the mass of the test masses. To minimize this noise, LIGO makes its mirrors very massive (they weigh approximately 40 kg, as explained in Section 4.3). The higher the mass of the optical elements, the lower the unwanted motion due to radiation pressure. Note that this noise dominates at lower frequencies, whereas $h_{\text{shot}}(P)$ dominates for higher frequencies.

As we aim to detect GWs in a broad frequency range, it is fundamental to find the pressure P that minimizes both types of noises and maximizes the sensitivity of the detectors. This happens when $h_{\text{shot}}(P) = h_{\text{rad}}(f,P)$. Solving for P from Equations (5.1) and (5.2) we get:

$$P = \frac{\pi c\lambda m f^2}{2N^2}. \tag{5.3}$$

The Quantum Noise (or Quantum Vacuum Noise) is given by,

$$h_{\text{quantum}}(f,P) = \sqrt{h^2_{\text{shot}}(P) + h^2_{\text{rad}}(f,P)}. \tag{5.4}$$

Using the optimal power P from Equation (5.3), we find that the Quantum Noise - often known as the Standard Quantum Limit (SQL), is given by [77]:

$$h_{\text{quantum}}(f) = \frac{1}{\pi f L}\sqrt{\frac{2\hbar}{m}}. \tag{5.5}$$

Using the power recycling and signal recycling cavities, as well as other optical devices described in Chapter 4, GW detectors aim to generate the optimal power from Equation (5.3) that minimizes this quantum noise.

5.1.2. Thermal Noise

LIGO's mirrors (and their suspensions) are subject to thermal noise like all physical objects with non-zero temperatures are. As seen in Figure 5.2, there are mainly three contributors to the thermal noise, namely: 'Coating Brownian' (in red), which is the dominant LIGO noise at 100 Hz, 'Suspension Thermal' (in blue) which is one of the main limiting sources below 10 Hz (and one of the reasons why space-based interferometers are needed to detect low-frequency GW signals), 'Substrate Thermo-Elastic' (in yellow), and 'Substrate Brownian' (in orange). These last noise source, which is one of the least significant ones, comes from the thermal excitation of the mirrors themselves [2].

The 'Coating Brownian' noise, as its name suggests, results from an analogy with Brownian motion - the random walk of particles that collide with molecules that move with thermal energy [2]. This dictates the random nature of kinetic energy in all parts of a macroscopic system at non-zero temperature. In LIGO's mirrors there exists residual thermodynamic motion of the individual atoms in its surface. As these mirrors are coated with several layers of materials that help increase their reflectivity (and therefore, their sensitivity), there is a mechanical coupling which translates into mechanical friction. The coatings of mirrors are one of the main sources of friction in an interferometer [72].

The 'Suspension Thermal' noise results from the thermal excitation of the suspension fibers that hold the mirrors of the interferometer, which makes them exhibit random motion. The thermal motion in a resonant system such as this one is concentrated at frequencies close to the resonance. Therefore, at high frequencies, this noise source is significantly low compared to the coating thermal noise [2].

Mitigating Thermal Noise

Mitigating Thermal Noise in aLIGO is one of its most challenging aspects. The first approach to noise reduction is lowering the temperature of the mirrors and the suspensions. However, in order to have a considerable reduction of the noise, such temperatures need to be very cold (near the absolute zero) at all times. A more feasible solution is using materials and construction methods that minimize mechanical losses.

The fused-silica suspension fibers that hold the mirrors have a special tapered shape that minimizes any kind of thermal motion in them. They assure that the broadband suspension thermal noise is low above 10 Hz. The size, weight and cylindrical shape of the mirrors are carefully designed to make the resonance frequencies as high as possible to avoid the frequency band to which the detectors are sensitive. The lowest of them is designed to be at 6.8 kHz. Additionally, all the optical elements are assembled in such a way that the friction among them is reduced. The suspension fibers and the mirrors are joined by polishing the pieces to be flat to a fraction of an optical wavelength [2] and then 'pasted' with a liquid solution of silica powder. These pieces are then joined under pressure, so they are free of resonances and have very low friction. Lastly, the optical reflective coating of the mirrors needs to be highly reflective, have low absorption, and have the capability of limiting the amount of scattered light (Scattered light is one of the most common glitches in aLIGO). Despite being ~ 0.1 mm thick, they have very high internal friction and are the main thermal noise source for frequencies above 10 Hz. Research in new coating materials is being carried out to achieve design sensitivity levels, and mitigate scattered light.

5.1.3. Seismic Noise

At low frequencies, 'Seismic' noise (ground motion due to human activity, wind, water, and Earth's seismic activity [9]) is one of the most limiting noise sources. This is due to the fact that it falls as [77],

$$h_{\text{seismic}}(f) = \frac{A}{f^2}\frac{\text{m}}{\sqrt{\text{Hz}}}, \tag{5.6}$$

with A a constant that depends on the site where the interferometer is located. LIGO uses several techniques to mitigate seismic noise, which has made this noise negligible above 10 Hz, as described below.

Mitigating Seismic Noise

In order to reduce the possibility of gravitational-waves being masked, several systems that mitigate seismic noise (and control the position of the mirrors) are used. The mirrors of aLIGO are suspended with four glass fibers of 0.5 mm diameter and ~ 60 cm long. Pendulum suspensions are used due to their ability to protect the mirrors from seismic motion.

A pendulum suspension with resonant frequency f_r provides an isolation factor of f_r^2/f^2 above f_r, which is an isolation factor of 100 at 10 Hz for a 1 Hz pendulum [77]. By using N pendulums, the attenuation effect increases by $(f_r^2/f^2)^N$. In aLIGO, four pendulums in series are used to deliver much of the seismic noise mitigation, providing an isolation effect of $(f_r^2/f^2)^4 = (f_r/f)^8$. This 'quadrupole-pendulum system', also known as Passive suspension, is shown in Figure 5.3.

To further reduce motion, the 'quadrupole pendulum' system is hung from a servo-controlled platform, as shown in the left panel of Figure 5.4. These systems monitor and control the velocity and position of a motor based on a feedback signal. LIGO's servo controls use several low-noise seismology sensors, which track motion and rotation around the $\hat{x}, \hat{y}$, and $\hat{z}$ directions, and magnetic motors to control these movements [2]. With the usage of all these seismic isolation techniques, Seismic noise is almost negligible for all frequencies above 10 Hz.

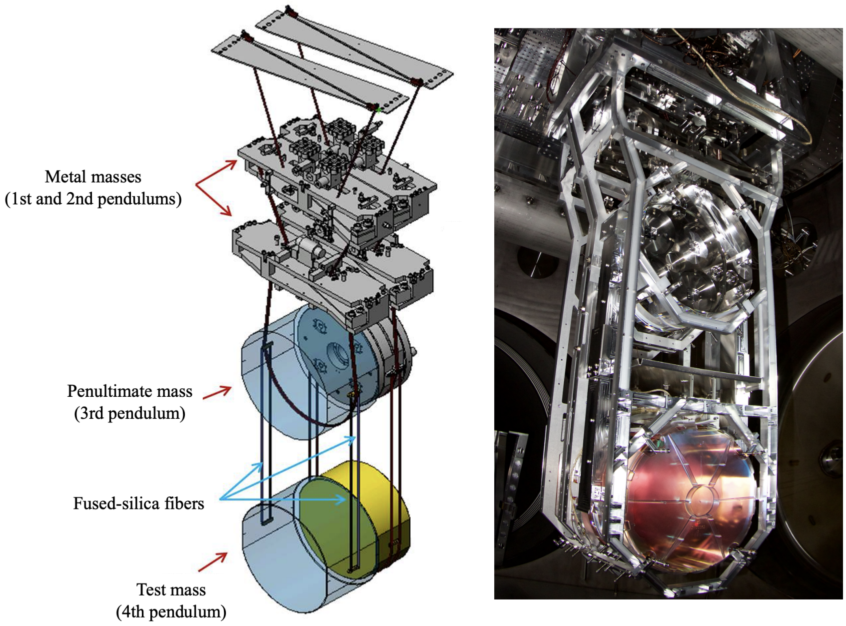


Figure 5.3: Optic suspensions of LIGO's mirrors. *Left panel:* Schematic showing the four pendulums in series used to suspend the mirror and mitigate seismic noise. The top two masses are made of steel, and the bottom two are made of pure fused silica. These last two are held with fused-silica fibers. (Figure adapted from [78]). *Right panel:* Real view of one of LIGO's test masses installed in its quadrupole suspension system. *Image credit Caltech/MIT/LIGO Lab.*

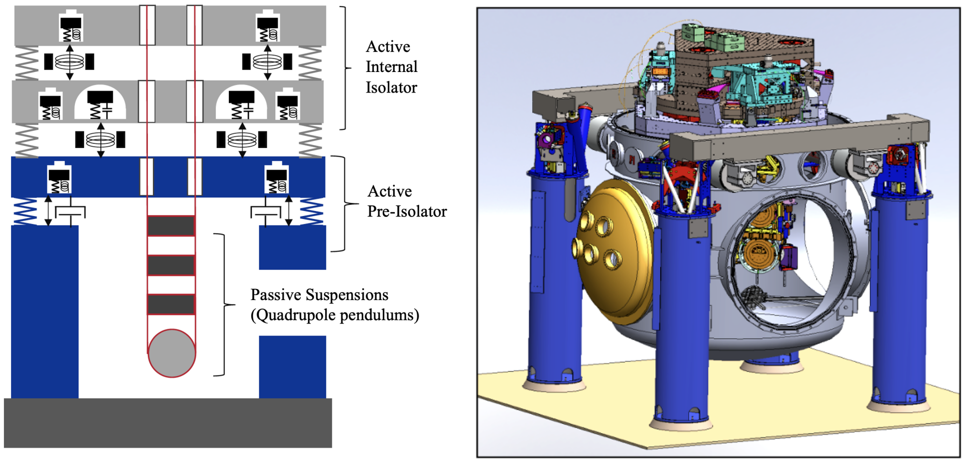


Figure 5.4: *Left:* Schematic of the passive suspensions (quadrupole pendulums) hanging from a multi-staged servo-controlled platform made up of mass-spring systems that provide additional seismic isolation. *Right:* The LIGO vacuum system, where the entire seismic isolation system from the left panel is housed. The external supports, shown in blue, provide an additional isolation layer [2]. (Figure adapted from [2]).

5.1.4. Newtonian Noise

Newtonian 'Gravity' Noise receives its name from the Newtonian gravitational attraction between the interferometer's mirrors and its surroundings. This kind of noise, which depends on the location of the detector, arises from local gravitational gradients of uneven distributions of matter around the interferometer that cause unwanted motion in the mirrors and masks GW signals [72].

There are two main causes of Newtonian noise: the movement of the buildings that house the detectors, and density fluctuations in the ground under the detectors. For the first noise source, the tilt of the optics (i.e., mirrors and suspensions) may be adjusted to reduce the relative movement [72]. However, for ground density fluctuations, no mitigation strategies to counteract movement exist yet. Most fluctuations in density are due to seismic waves (i.e., as the compression wave passes, the interferometer is pulled toward the more compressed, denser earth, moving the detector's mirrors [2]). These density fluctuations are random and non-stationary; however, they occur at ~ 7 Hz, and are not one of the dominant sources of low-frequency noise for LIGO and Virgo.

In the future, 'Newtonian' Noise will be one of the most limiting noise sources for ground-based detectors at low frequencies, as shown in dark green for the Cosmic Explorer in Figure 5.6. While placing detectors underground (like KAGRA and the Einstein Telescope in the future - see Section 5.2.1) reduces the Newtonian noise significantly, lower frequencies will only be accessible in space-based interferometers, such as LISA, described in Section 5.2.2.

5.2. Detecting gravitational-waves in a broader frequency range: next-generation GW detectors

While the current network of GW interferometers (see Section 4.5) allows us to explore our gravitational-wave universe for objects with high frequencies, there is much to explore in other frequency bands.

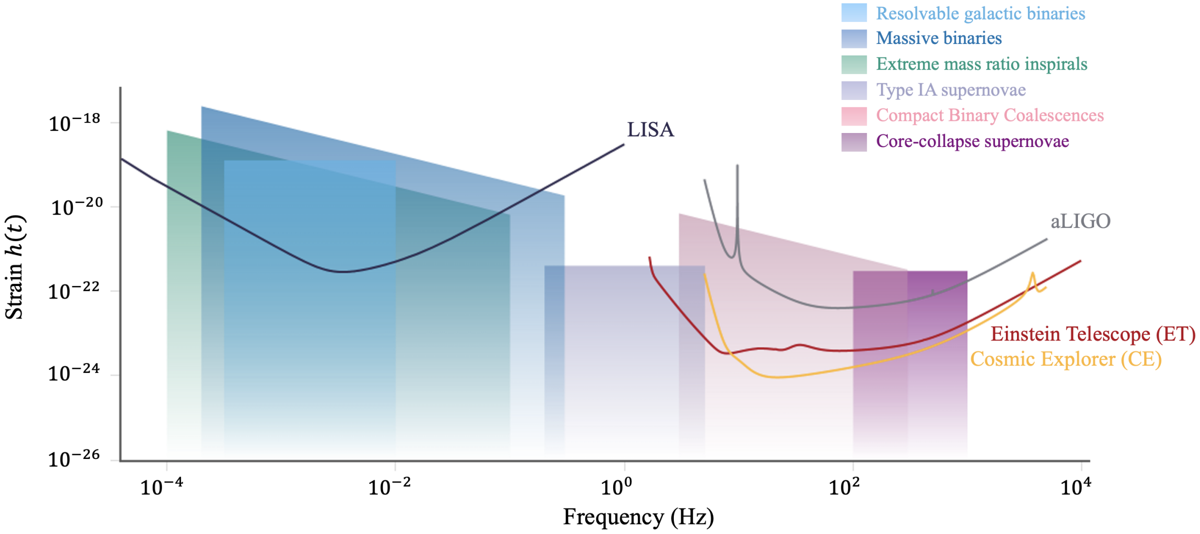


Figure 5.5: Noise curves for LISA, aLIGO, the Einstein Telescope (ET), and the Cosmic Explorer (CE) as a function of frequency, along with the strain of potential astrophysical noise sources these interferometers might detect. Low-frequency signals (namely resolvable galactic binaries, massive binaries, and extreme mass ratio inspirals) are only detectable by LISA. Higher frequency signals, like Compact Binary Coalescences (CBCs) and core-collapse supernovae (Figure adapted from a plot generated with the gravitational-wave sensitivity plotter [79]).

There are two ambitious plans for new generation GW ground-based observatories: the European Einstein Telescope (ET [55]) and the US Cosmic Explorer (CE [54]), as well as the first space-based observatory, LISA (Laser Interferometer Space Antenna [48]). Figure 5.5 shows the strain against frequency for these detectors (and aLIGO, for reference) and the detectable range of a variety of possible astrophysical sources. Note that the idea of having additional detectors is not to replace any GW interferometers, but rather to complement and augment the frequency band to which the current GW detector network is sensitive[2]. The following sections briefly describe these three proposed detectors.

5.2.1. The Cosmic Explorer and the Einstein Telescope

The Einstein Telescope (ET [55]) and the Cosmic Explorer (CE [54]) are two proposed ground-based gravitational-wave interferometer detectors planned to be built in Europe and the US in the 2030s. Instead of having 3 km and 4 km long arms, ET envisions 10 km arms while CE plans two detectors, one with 40 km long arms and the other with 20 km long arms.

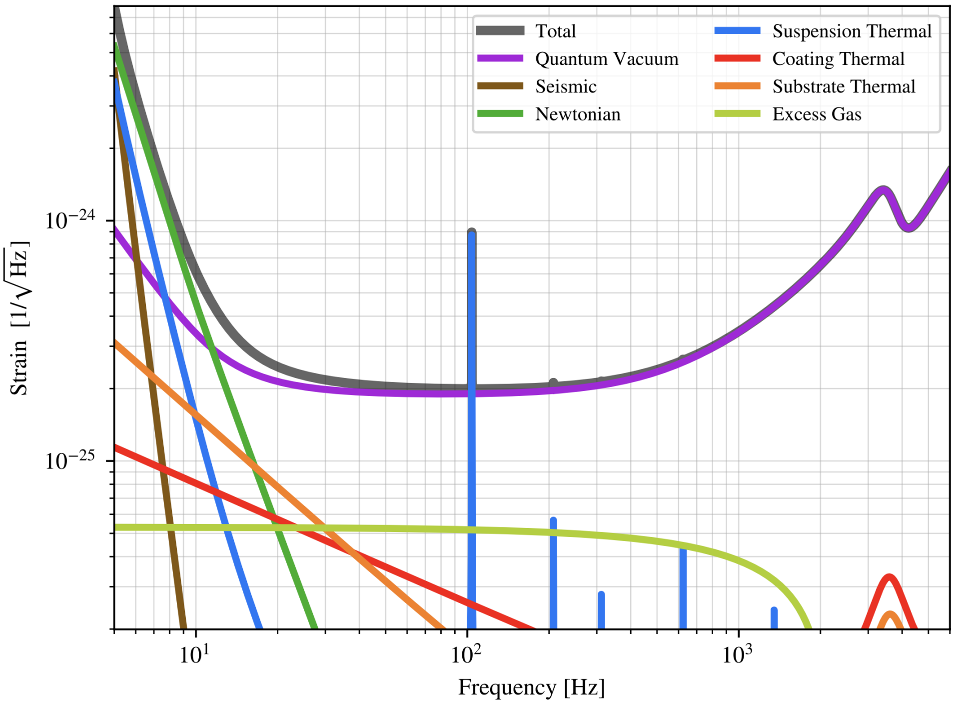


Figure 5.6: Amplitude Spectral Density (ASD) of the principal noise sources of the Cosmic Explorer (CE). Note that most of them are shared with the ones from aLIGO, but have much lower noise than the current GW interferometer. Note that, at lower frequencies, the most relevant noise source for CE will be 'Newtonian' noise, while the 'Seismic' and 'Suspension thermal' noise sources will not suppose a problem at such frequencies. Additionally, the 'Coating thermal' noise, which is the most relevant thermal noise for aLIGO, will not be a considerable noise source either. (Figure generated using the Gravitational Wave Interferometer Noise Calculator, [76]).

In order to reduce seismic noise and Newtonian noise, ET is planned to be underground; and instead of the classic Michelson-Morley 'L'-shaped interferometer design, it will have a triangular form with three detectors (just as LISA), as shown in the left panel of Figure 5.7. Each detector consists of two interferometers. This topology will permit the recovery of the polarization of a

---

[2] Other proposed instruments, such as Pulsar Timing Arrays, intend to explore GWs at much lower frequencies than LISA.

signal using a single observatory [2].

On the other hand, the CE is designed toward detecting specific GW sources, and just like LIGO, Virgo, and KAGRA it is based on the Michelson-Morley interferometer shown in Figure 4.1. The 20 km antenna is targeted toward observing the final phase of neutron-star merger signals, whereas the 40 km long one aims to observe BBH mergers at the edge of the universe [2]. Figure 5.6 shows the main noise sources that the CE will have. Note that, as mentioned in Section 5.1.4, the most limiting noise source at low frequencies will be Newtonian noise.

It is expected that both CE and ET will have ten times more sensitivity than current GW detectors. This will put mostly all coalescing binaries of $\sim 100\ M_\odot$ within the detectable range of ground-based detectors [2].

5.2.2. LISA: the Laser Interferometer Space Antenna

The Laser Interferometer Space Antenna, LISA [48], is a European Space Agency (ESA) led mission (with important contributions from NASA) that aims to build the first space-based gravitational-wave interferometer. Just like ET, LISA will be triangular-shaped: three detectors, each consisting of two interferometers, will be arranged in a triangular topology, moving along a heliocentric orbit similar to Earth's [48]. LISA's arms will be approximately 2.5 million km long; several orders of magnitude larger than those of ground-based GW interferometers. A conceptual design of LISA is shown in the right panel of Figure 5.7.

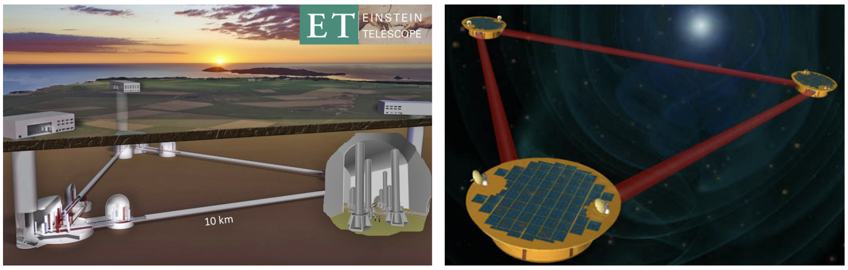


Figure 5.7: Conceptual designs of the Einstein Telescope (ET [55]) and the Laser Interferometer Space Antenna (LISA [48]). ET will be built underground whereas LISA will be the first space-based GW interferometer. Note that, different from aLIGO, Virgo, KAGRA, and the future Cosmic Explorer (CE [54]), ET and LISA consist of three detectors arranged in a triangular, equilateral topology, and each of them consists of two interferometers. *Image credit EGO (European Gravitational Observatory) and NASA-ESA, respectively.*

Without seismic and Newtonian noise, LISA will be able to detect signals ranging from $100\ \mu$Hz to 0.1 Hz, which are invisible to current GW interferometers. This will allow LISA to be sensitive to sources all ground-based detectors will not be able to discover, such as binary white dwarf systems (see Section 3.1.6), supermassive black hole (SMBH) mergers (with $10^3\ M_\odot \leq \mathcal{M} \leq 10^7\ M_\odot$), and GWs from extreme mass ratio inspirals (EMRIs), where a stellar compact object (with $M \leq 60\ M_\odot$) decays slowly around a massive black hole ($\sim 10^5\ M_\odot$). LISA will be also sensitive to detect the continuous waves (CWs) emitted by BNS and BBH systems in their inspiral phase in the Milky Way [48].

6 Transient noise: the impact of glitches on detecting gravitational-waves

Besides the noise sources described in Chapter 5, GW detectors are prone to other non-astrophysical transients, commonly known as *glitches*. These short-lived, non-Gaussian instrumental noise signals are particularly problematic when they mask and/or mimic the morphology of gravitational-waves, as they may generate false-positive candidates [12, 80, 81], corrupt data, and bias astrophysical parameter estimation (PE) [82–84].

In this sense, glitches impose an additional challenge for GW searches (see Section 4.6). If aLIGO noise was perfectly Gaussian, the matched filter SNR from Equation (4.27) alone would be enough to search for GWs. However, since LIGO and Virgo data have glitches, additional consistency tests between the data and the GR predictions are required to discriminate real GW signals from terrestrial and instrumental noise sources. The following section describes the χ^2 discriminator [85], a widely-used and more accurate method for GW signal searches. The rest of this Chapter is dedicated to understand the most frequent glitches in the LIGO detectors.

6.1. The χ^2 discriminator: the impact of detector noise on searches for gravitational-waves

An ideal GW-vs-glitch discriminator would veto all glitches and validate all astrophysical signals [75]. While such perfect test does not exist, one of the most widely used discriminators in GW searches is the χ^2 discriminator [85]. This test is employed by the PyCBC [86] matched-filter algorithm to perform GW searches in gravitational-wave detector data.

The χ^2 method divides the frequency space spanned by the waveform template into bins of equal energy, and checks if each bin of the data with a candidate event contributes the expected amount of energy [2, 75]. This discriminator is given by [85]:

$$\chi^2 = \frac{1}{2p-2}\sum_{i=0}^{p} ||\langle s, h_i\rangle - \langle h_i, h_i\rangle||^2, \tag{6.1}$$

with p the number of frequency bins, which depends on the duration of the template, h_i the waveform template in each of these bins, and s the detector data (Note that h_i and s are functions of frequency). The inner product $\langle\cdot,\cdot\rangle$ is defined in equation (4.27). If $\chi^2 > 1$, the calculated SNR is re-weighted with the χ^2 discriminator to produce a re-weighted SNR, $\tilde{\rho}$ [75]:

$$\tilde{\rho} = \begin{cases} \rho & \text{if } \chi^2 \leq 1, \\ \rho\left[\frac{1}{2}(1 + (\chi^2)^3)\right]^{-1/6} & \text{if } \chi^2 > 1. \end{cases} \tag{6.2}$$

If there is a good match between the detector data and the signal template, $\tilde{\rho} = \rho$. However, if there is a mismatch, $\tilde{\rho}$ will decrease [2], suggesting the candidate is not astrophysical in origin.

Figure 6.1 shows an example of the SNR, re-weighted SNR, and χ^2 discriminator applied to a gravitational-wave signal (GW170814) in the top panel, and to a Blip glitch (a loud and common glitch in LIGO data, see Subsection 6.2.1) in the bottom pannel. Note that, while the SNR for the glitch is even higher than for the GW signal, when it is re-weighted with the χ^2 discriminator, it reduces the astrophysical relevance of the event. On the other hand, the SNR and re-weighted SNR are almost the same for the time of GW170817.

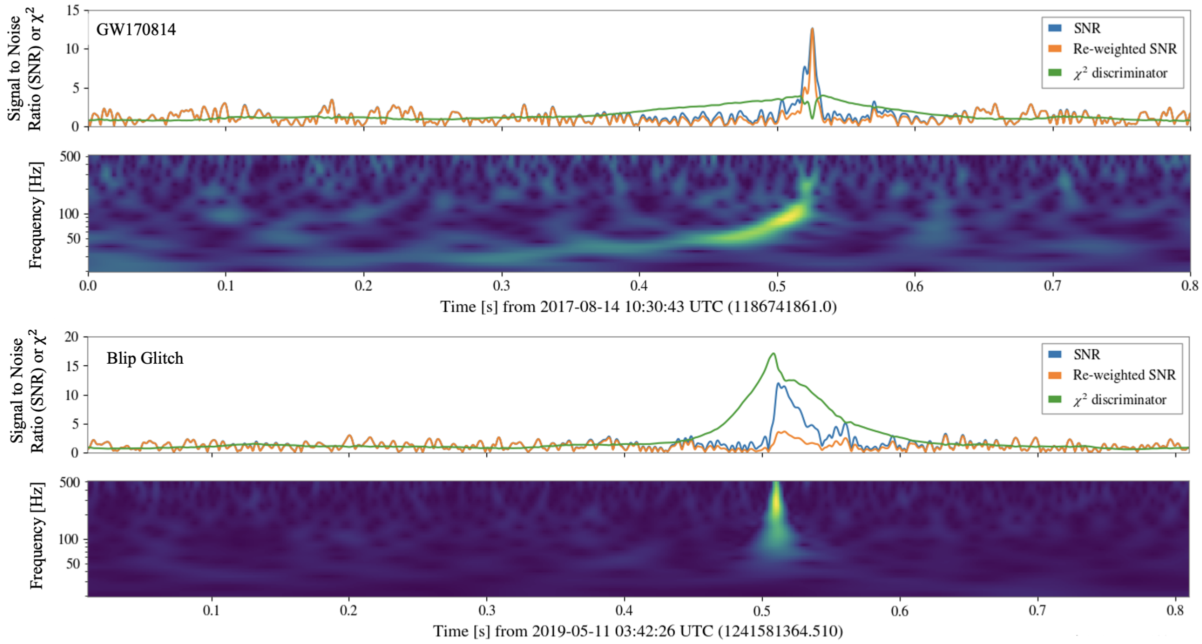


Figure 6.1: Matched-filter signal-to-noise ratio (SNR, in blue), re-weighted SNR (in orange), and χ^2 squared discriminator (in green), along with time-frequency (spectrogram) visualizations for GW170814 (top panel) and a Blip glitch (bottom panel). The matched-filter SNR and re-weighted SNR are almost equivalent for the astrophysical signal, but differ considerably for the glitch. As the time-frequency evolution of the glitch does not match the waveform template, the glitch is rejected as a GW candidate. (Image generated with PyCBC [86], based on Figure 5 of [2]).

The effectiveness of this test depends on the duration of the signal and the number of bins (p) used in the test [80]. The test is very effective for long duration signals; rejecting a variety of glitches for such candidates. However, if the noise transients are morphologically similar (in time and frequency) to gravitational-waves, the discriminator will not be an effective method to distinguish glitches from real astrophysical events. In such cases, the usage of further specialized tools that aim to detect these glitches, such as the one developed in this thesis - GSpyNetTree, are required.

6.2. Types of glitches

While there is a plethora of glitches that affect LIGO and Virgo data, the most problematic cases of transient noise are those that mimic the behavior of real GW events. Some of these glitch classes have a known cause. However, the origin of most of them is yet to be understood. Table 6.1 summarizes the characteristics of the glitches that have been found to share morphological similarities to GW candidates [13], and, thus, the most problematic ones for identifying real astrophysical events. These are the types of glitches that GSpyNetTree considers.

Glitch Type	**Frequency Range [Hz]**	**SNR**	**Duration [s]**	**Morphology**
Blip	$80 - 500$	≈ 10	$\ll 1$	Pulse
Low-frequency Blip	$10 - 100$	≈ 10	$\ll 1$	Pulse
Koi Fish	$10 - 1000$	> 100	~ 1	Loud pulse
Tomte	$10 - 100$	≈ 10	< 1	Pulse
Scratchy	$50 - 500$	< 10	~ 2	Drifting frequencies / "Comb"
Light Scattering	$20 - 200$	≈ 10	< 2	Arches
Fast Scattering	$20 - 200$	≈ 10	< 2	Energy blops in close succession
Low-frequency Lines	$10 - 20$	≈ 10	$0.1 - 2$	Lines

Table 6.1: Summary of the most important glitch classes in LIGO-Virgo data, known to be morphologically similar to gravitational-wave signals. The frequency range, SNR, duration, and morphology are described for each type of glitch. Note that the longest glitches in duration are Scratchy, Light Scattering, Fast Scattering, and Low-frequency Lines, all with different morphologies. In contrast, the shortest glitches in duration all have pulse-like morphologies. It is important to note that all these glitch classes are included in GSpyNetTree. As described in Chapter 9, GSpyNetTree considers that the morphology of each glitch makes it more similar to GW signals in specific mass ranges than others; and uses this to determine in which of its three classifiers they should be included. (Table based on [72]).

The rest of this Section is devoted to explain each of said glitches, their causes, and possible mitigation techniques (if known), based on [12]. A time-frequency visualization of each of the signals of Table 6.1, displayed in the time duration that best captures each of their morphologies[1], is shown in Figure 6.2.

6.2.1. Blip and Low-frequency Blip glitches

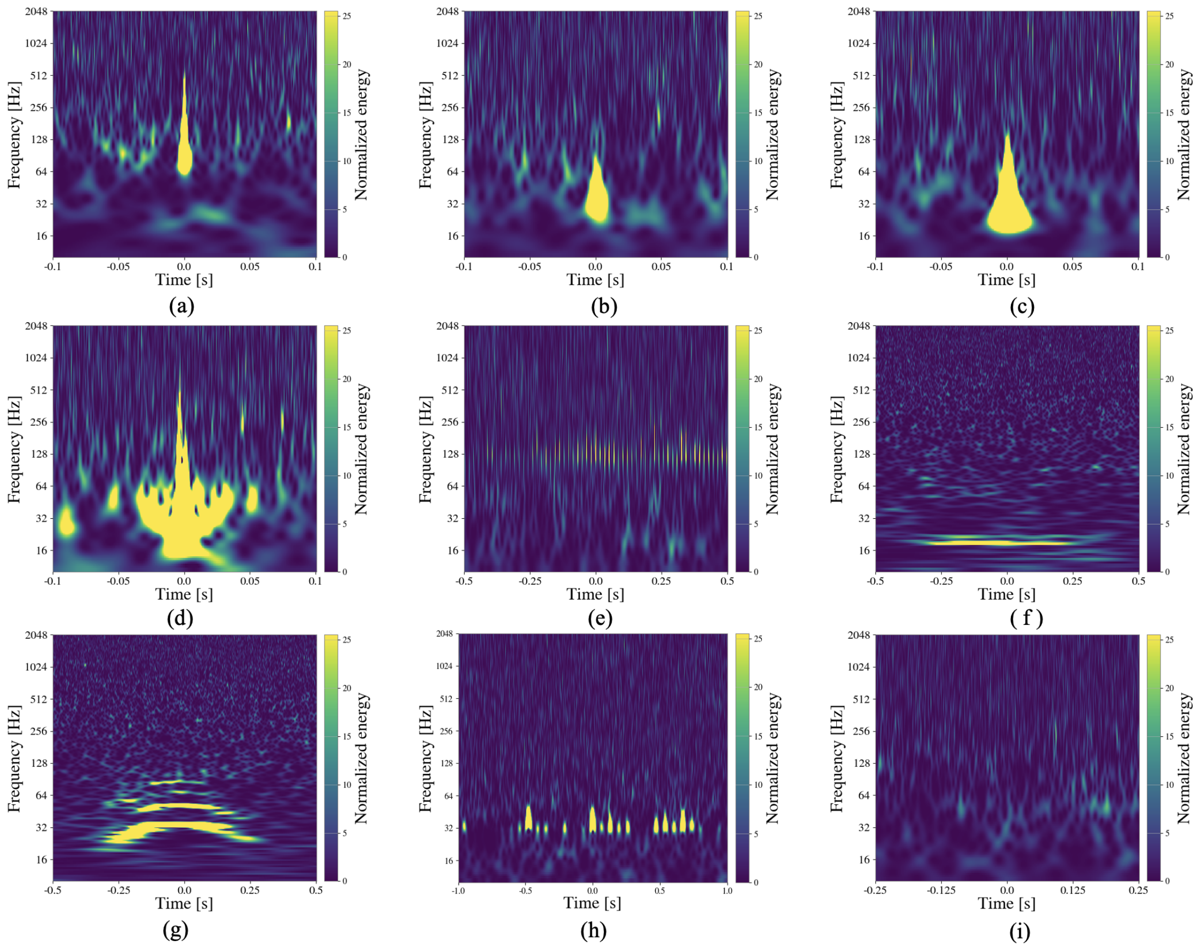
Blip transients are short duration glitches with high frequency bandwidth and no known instrumental or environmental coupling [12]. Due to its morphology, as seen in Figure 6.2(a), they may resemble a GW signal in a broad mass range. As a matter of fact, they are problematic within the entire range of masses contemplated by GSpyNetTree ($5\ M_\odot \leq M < 250\ M_\odot$). Due to their duration and large bandwidth, they overlap significantly with the shortest templates used in matched-filter searches, reducing the effectiveness of GW searches [12].

During the second observing run (O2), approximately 2 blip glitches per hour occurred in the two LIGO laboratories. This rate increased to about 4 per hour in the Livingston Observatory in O3. While these glitches have been found in all detectors (including Virgo), they do not occur in coincidence at either site, confirming they are not astrophysical in origin. As a matter of fact, the rate of blips follows a Poisson distribution [12].

One of the fundamental issues with Blip glitches is that they are typically not registered by the witness sensors that monitor the detector, impeding their systematic removal from the analyses [72]. So far, the origin of only a subset of them has been found to be correlated with instrumental issues, including computer timing errors. Many other investigations on the origin of Blip glitches have been pursued, including studies of correlations between low humidity or

---

[1] Figure 10.2 shows time-frequency visualizations of these glitches in the time-ranges considered by GSpyNetTree.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.2: Time-frequency visualizations for the most problematic glitch classes (a-h) in LIGO-Virgo data, along with a spectrogram of a clean detector background time (i) - i.e., without a glitch. Note the diversity in duration and morphology among glitch classes; this is why some of them are more similar to GWs in certain mass ranges than others - as considered by GSpyNetTree. The glitch classes shown in this Figure are: (a) Blip, (b) Low-frequency Blip, (c) Tomte, (d) Koi Fish, (e) Scratchy, (f) Low-frequency Lines, (g) Light Scattering, and (h) Fast Scattering.

high energy cosmic rays with the high rate of Blips, both of which did not find the respective phenomena responsible for these glitches [87]. Thus, the origin of the vast majority of blips remains unexplained.

Similar to Blips, Low-frequency Blips have no known origin. Due to their morphology, they are most likely to mimic short-duration GW transients, like signals from high-mass ($50\ M_\odot \leq M < 250\ M_\odot$) and extremely high-mass ($M \geq 250\ M_\odot$, as defined by GSpyNetTree) CBCs hindering detection. Low-frequency blips appeared during the third observing run (O3), they were first detected by Gravity Spy [3], and receive their name from their morphological similarity with Blips, but at much lower frequencies (namely, between 32 and 128 Hz) [88].

6.2.2. Tomte glitches

Just like Blips, Tomte glitches have SNR $\approx$ 10, as shown in Table 6.1. However, they typically impact frequencies below 128 Hz (see Figure 6.2(c)). Although they have not been mitigated yet, their origin is associated to the response in the controlling system of the mirror [72].

6.2.3. Koi Fish glitches

Koi Fishes are short, very loud glitches (SNR > 100) in LIGO and Virgo data, which saturate spectrogram visualizations almost in the entire frequency band, as shown in Figure 6.2(d). They are a problematic source of noise because they cause sudden and substantial reductions in the sensitivity of the detectors [12], besides being very frequent in detector data. Just in O3, they occurred with an average rate of 3.3/hour and 3.5/hour at the Livingston and Hanford Observatories, respectively. Similar to Blips, their rate follows a Poisson distribution, and no coupling with the detector has been found yet [12].

6.2.4. Scratchy glitches

Scratchy glitches, which appear to affect the 50 - 500 Hz band (see Figure 6.2(e)), are clusters of vertical lines with excess power in detector data. Despite only occurring once or twice per day, their long duration and variable frequencies make them a problematic source of noise, especially for low-mass gravitational-wave signals from CBCs ($5\ M_{\odot} \leq M < 50\ M_{\odot}$, based on the definition used in GSpyNetTree).

6.2.5. Low-frequency Lines

Low-frequency Lines are the type of glitches that affect the lowest frequencies in LIGO-Virgo data. They started to be more common in the third observing run, and overlapped with a significant amount of candidates, especially in Virgo. These glitches have no known origin yet.

6.2.6. Scattering glitches

As their name suggests, Scattering glitches are due to scattered light. These type of glitches surge when part of the laser light gets scattered off of the mirrors, hits a surface (i.e., the scatterer, which can be a reflective source or optic mounts, for instance) and rejoins the main beam [12], as shown in Figure 6.3. The amplitude of the glitch depends on the amount of scattered light that recombines with the main beam; and, its maximum frequency, on the strength of the relative motion between the mirror and the scatterer [12]. Scattering was one of the prevailing glitches in O3, and it came in two different morphologies: Light Scattering (or Slow Scattering, see Figure 6.2(g)) and Fast Scattering, as shown in Figure 6.2(h).

Light Scattering (or Slow Scattering) arches occur due to high ground motion in the micro-seism band (i.e., 0.1 - 0.3 Hz), and affect GW sensitivity between 20 Hz and 120 Hz. During periods of intense ground motion, the scattered light can be reflected multiple times between the mirror and the scatterer. In this cases, higher frequency harmonics of the scattering arches can be visualized in spectrogram visualizations. On the other hand, Fast Scattering glitches are correlated with ground motion between 1 Hz and 6 Hz. Some causes of these glitches are human activity and thunderstorms near the LIGO sites [12]. This type of glitch is more frequent in the

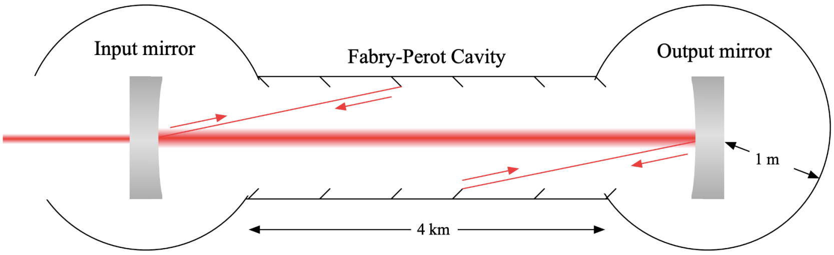


Figure 6.3: Schematic representation of the occurrence of scattering in the Fabry-Perot cavity of one of LIGO's arms. Light gets scattered in the mirrors of the interferometer, as shown with the arrows. After reflecting in a particular surface, it recombines with the main beam. (Figure adapted from [89]).

Livingston Observatory than in Hanford, and this is due to the passing of trains near the Y arm of the former [90].

6.3. Importance of glitch mitigation and subtraction

Glitches impose an additional challenge in LIGO-Virgo data analysis. While the first step in the pipeline is to detect these glitches (which is where tools like GSpyNetTree are used), gating, mitigating, and subtracting them are the steps that follow. The latter procedures become particularly important when glitches overlap with candidate signals (especially those that may trigger an electromagnetic follow-up campaign, like BNS and BHNS systems), and are necessary to accurately estimate the parameters of the sources that originated the GW signals.

A famous example of a loud glitch overlapping with a signal is that of GW170817, the first detection of a BNS system (see Section 3.1.4). This glitch happened 1.1 s before the coalescence time of the event, and overlapped in time and frequency with the final merger stage of the GW, as shown in the top panel of Figure 6.4. Using several techniques, the glitch was subtracted from the data so the parameters (e.g., sky location) of GW170817 could be estimated, and the electromagnetic search of the event could follow.

To remove this glitch (or, in general, any glitch) from the data without impacting considerably the trace of the GW signal, several methods can be applied. In the case of GW170817 it was first important to do a rapid analysis (without the glitch) that could generate a better sky localization of the signal and facilitate the EM follow-up campaign. In this process, the *gating* algorithm of the PyCBC [86] library was used. In *gating*, using the strain $h(t)$, the data around the time of the glitch is set to zero before running the matched filter algorithm. To do this process, the data is multiplied by a Tukey window function centered on the time of the peak. The usage of the Tukey window, also known as the cosine-tapered window, assures no discontinuities are induced in the input data. The window applied for this event is shown in the gray curve in the bottom panel of Figure 6.4. This process allowed a rapid reanalysis of the source and constrained GW170817 to a smaller region in space.

The process previously described may induce severe miscalculations in the parameters of the source. To accurately estimate the ones of GW170817 (and in general, any other signal), more sophisticated techniques need to be applied. In the case of GW170817 (and 15 candidates in

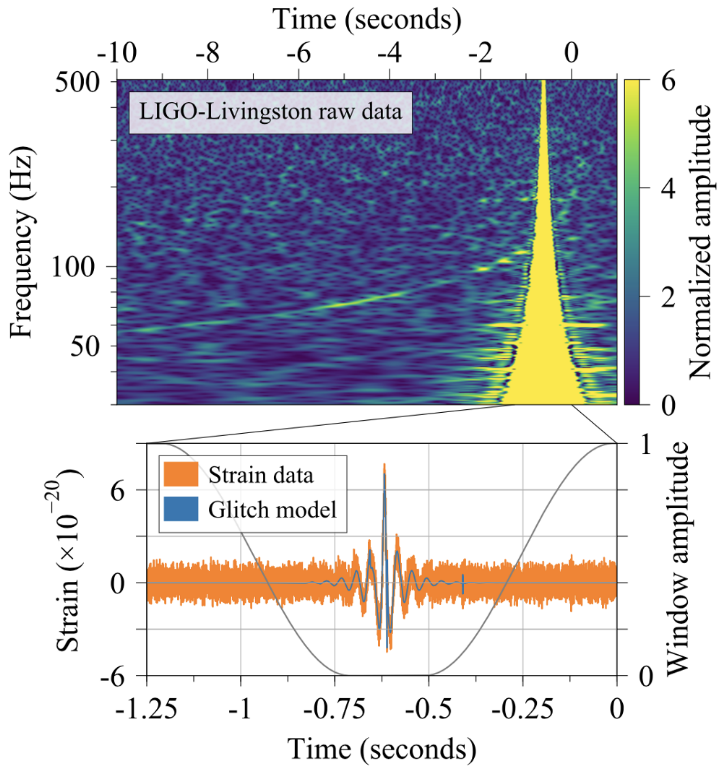


Figure 6.4: *Top panel:* Time-frequency visualization of the last 10 seconds of GW170817 as observed in the raw data of the LIGO Livingston observatory. Despite the loud glitch, which occurred 1.1 s before the coalesce time, the track of the signal is clearly visible. *Bottom panel:* Strain $h(t)$ data in the time domain of time around the glitch. To calculate the sky location of the GW, a rapid analysis used a Tukey window (in gray, right axis) to zero the data around the glitch. To further measure the source's properties, the glitch was modeled with `BayesWave` [91], an algorithm that generates a model of the glitch based on a wavelet reconstruction (shown in blue), and subtracted afterwards (Reproduced from [38]).

O3) [92], `BayesWave` [91] was used to perform glitch subtraction[2]. The following paragraphs outline the basic method `BayesWave` uses to subtract glitch data, based on [92].

The `BayesWave` algorithm assumes that in each interferometer, the timeseries data $h(t)$ from the detector is a linear combination of Gaussian noise, $n(t)$, an astrophysical signal, $s(t)$, and a glitch (or glitches), $g(t)$ [92]:

$$h(t) = n(t) + s(t) + g(t). \tag{6.3}$$

Both the signal and the transient noise (glitches) are modeled using a sum of sine-Gaussian (Morlet-Gabor) wavelets, which are given in the time domain by [91]:

$$\Psi(t; A, Q, f_0, t_0, \phi_0) = Ae^{-(t-t_0)^2/\tau^2} \cos\left(2\pi f_0(t - t_0) + \phi_0\right). \tag{6.4}$$

with $\tau \equiv Q/(2\pi f_0)$. The parameters of the wavelet are Q the quality factor, A the amplitude, ϕ_0 the phase offset, and t_0 and f_0 the central time and frequency, respectively. The number of wavelets used, and their parameters, are calculated using a transdimensional MCMC (Markov

---

[2] Other algorithms, like `gwsubtract` [93], have also been used to mitigate glitches in LIGO-Virgo data.

Chain Monte Carlo)[3] [91].

Glitches are modeled independently and separately for each detector, as they are instrumental noise. However, to model the GW event, a common set of wavelets, which have parameters describing the sky location and polarization, is used [92]. In order to mitigate a glitch, either of the following scenarios must occur: (a) the data $h(t)$ contain Gaussian noise $n(t)$ and a glitch $g(t)$, or (b) the data $h(t)$ contain Gaussian noise $n(t)$, a glitch $g(t)$, and an astrophysical signal $s(t)$. If the glitch and the signal occur sufficiently separated in time and/or frequency from each other, approach (a) is followed. On the other hand, if the glitch and signal overlap significantly, method (b) is applied. This is where the importance of having multiple detector data takes relevance. Using strain $h(t)$ from other interferometers, the coherent signal power can be separated from the glitch, assuring that no significant signal power is subtracted during glitch mitigation [92]. The modeled (and subtracted) glitch that overlapped with GW170817 in the Livingston Observatory is shown in the blue line of the bottom panel of Figure 6.4, between -0.75 s and -0.5 s. Note how the glitch removal allows the accurate reconstruction of the GW signal, as shown in the central panel of Figure 3.9.

---

[3] The transdimensional MCMC models are an extension of MCMC sampling algorithms that allow transitions between models of different dimensions.

7 LIGO-Virgo Detector Characterization and the infrastructure to understand, detect, and mitigate noise

Having studied the instrumentation of the GW interferometers (see Chapter 4), and their most relevant sources of noise (see Chapters 5 and 6), it is important to study the infrastructure and software tools used for detection, mitigation, and subtraction of noise in LIGO and Virgo. This Chapter is dedicated to understanding detector characterization in LIGO-Virgo, explaining some of its tools (namely, the ones that interact and/or are used by/with GSpyNetTree, based on [12]), and its infrastructure (which is fundamental to understand how GSpyNetTree fits within the LIGO-Virgo gravitational-wave event validation pipeline). A special emphasis is given to the Data Quality Report.

7.1. Gravitational-wave detector characterization tools

Detector Characterization (DetChar) in LIGO-Virgo is the effort of understanding, detecting, and mitigating sources of noise, both in the instruments and the data [12], and it is a fundamental component of improving the performance of the LIGO-Virgo interferometers. As the detectors are subject to a wide range of noise, which constantly evolves during observing runs due to new technologies that emerge to improve sensitivity, it is important to have robust tools that address this issue and help analyze the strain $h(t)$ data produced by the detectors. The following subsections describe some of the most important Detector Characterization tools.

7.1.1. GWpy

`GWpy` [94] is a signal processing and data visualization Python library used to study data from GW detectors. It is computationally optimized for manipulating data both in the time and frequency domain. It also has implementations of the fast Fourier transform and the multi-Q transform (see Subsection 7.1.3) for timeseries data [12]. As a matter of fact, most of the spectrograms in this thesis were generated using GWpy.

7.1.2. GW-DetChar

`GW-DetChar` is a Python extension of `GWpy` with specific applications to detector characterization. It includes tools to generate single web-pages with responsive design features, which are used to record data and interpret results. An example use case of this functionality of GW-Detchar is the Summary Pages (see Subsection 7.1.4) [12]. As a matter of fact, the output of every prediction of GSpyNetTree is a webpage that utilizes `GW-DetChar`.

7.1.3. Omega scans

Omega scans [95] are a type of spectrograms, or time-frequency visualizations, that are widely used to visualize the morphology of GW signals and glitches. Omega scans consist of a raw multi-

Q transform. This method, which is closely related to the Fourier transform, consists of multiple spectrograms that use tiles with a constant ratio of duration to bandwidth (the quality factor, Q) [12]. An optimal Q that maximizes the energy in a single time-frequency tile is chosen to perform the Q-transform, which is then interpolated to produce a high-resolution time-frequency visualization of the energy of a signal. All time-frequency visualizations presented in this thesis up until this Chapter use the Q-transform.

The Q-transform is also one of the fundamental tools used by GSpyNetTree. Indeed, all the images of glitches and signals are generated using the Q-transform, with a Q-value of 20, as explained in Chapter 12.

7.1.4. Summary Pages

The summary pages are a web service of LIGO that offers automated daily analyses of the gravitational-wave strain data and other interferometer subsystems. They are one of the main tools used to monitor the performance of LIGO detectors and overall data quality. The raw HMTL of the summary pages is generated via Python code using `GWDetChar`, and the visual layout is color-coded by interferometer [12]: blue for LIGO Livingston, red for LIGO Hanford, and orchid for Virgo.

7.1.5. LIGO-DV Web

The LIGO Data Viewer Web Service (LDVW) [96] is an online data visualization platform that provides direct access to data recorded at the LIGO and Virgo Observatories. This software is often used to generate visualizations of data and complement the analyses done with the Summary Pages [12]. Additionally, LIGO-DV web has a glitch database, in which users can filter transient noise by epoch (e.g., third observing run), type (e.g., Blip), and interferometer (e.g., Virgo), among others. These glitches correspond to Gravity Spy classifications (see Section 8.3.1), and are the fundamental data source of glitches used for building the training sets of GSpyNetTree.

7.1.6. The Data Quality Report (DQR)

The Data Quality Report (DQR) is one of the fundamental tools used for LIGO-Virgo event validation [16]. It consists of a collection of analyses and routines, including GSpyNetTree, used to enable and support the vetting of non-astrophysical GW candidates. Its infrastructure is modular so that new tools and analyses can be added as needed [12]. The DQR is integrated with the LIGO-Virgo Alert System (LVAlert) [97] and the Gravitational-wave Candidate Event Database (GraceDB) [98]. When a gravitational-wave search pipeline identifies a GW candidate, the event is recorded in GraceDB and the LVAlert broadcasts a notice to all subscribers, including the DQR. The DQR then triggers the analyses of each of the tasks it handles, like GSpyNetTree. For each of the tasks, a Data Quality issue is reported whenever the $p-$value is below a particular threshold, and each task has its own way of calculating the $p-$value. The following subsections outline the relevant, documented tasks that, along with GSpyNetTree, make up the O4 Data

Quality Report[1]. Other tasks used in O4 include: Glitch Average, Omega overlap, and GPS Numerology [17].

In parallel, an automated notice announcing the detection of a new GW candidate is sent from GraceDB to the General Coordinates Network (GCN) [99] of NASA. The DQR architecture then uploads web-based reports (generated with `GW-DetChar`) for each of its tasks to GraceDB. With these reports, a further internal review is performed to inform the decision of disseminating additional GCN Notices and Circulars or retracting[2] an announced GW candidate [12]. Figure 7.1 shows how this process works for a DQR task like GSpyNetTree.

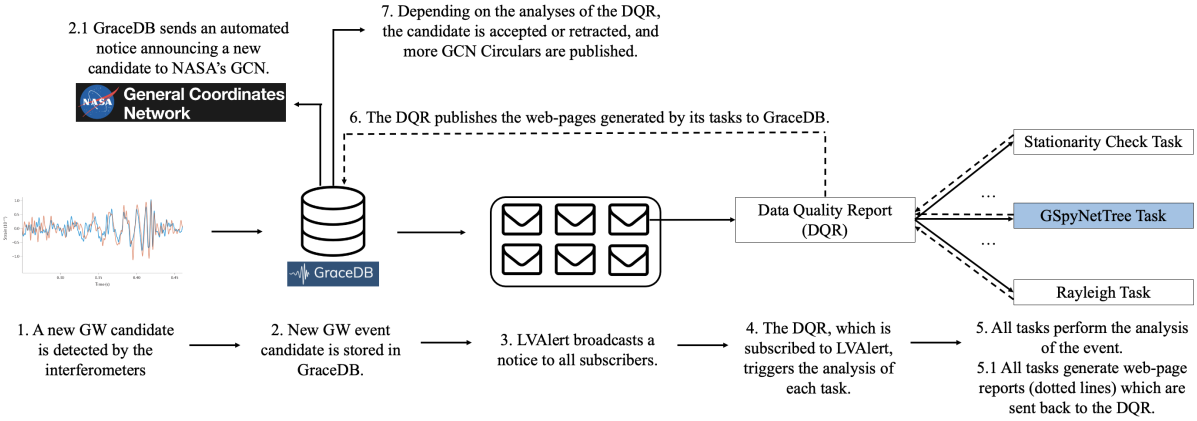


Figure 7.1: Schematic diagram of the GW event validation pipeline of LIGO-Virgo. After a GW event is stored in GraceDB, a public NASA GCN Circular is generated, and the event is broadcasted by the LVAlert system to the Data Quality Report (DQR). The DQR triggers the analyses of the event using all of its tasks, including GSpyNetTree. All DQR tasks generate web-page reports which are published to GraceDB. After an internal review of the results, more GCN Circulars are published to validate or retract the candidate event.

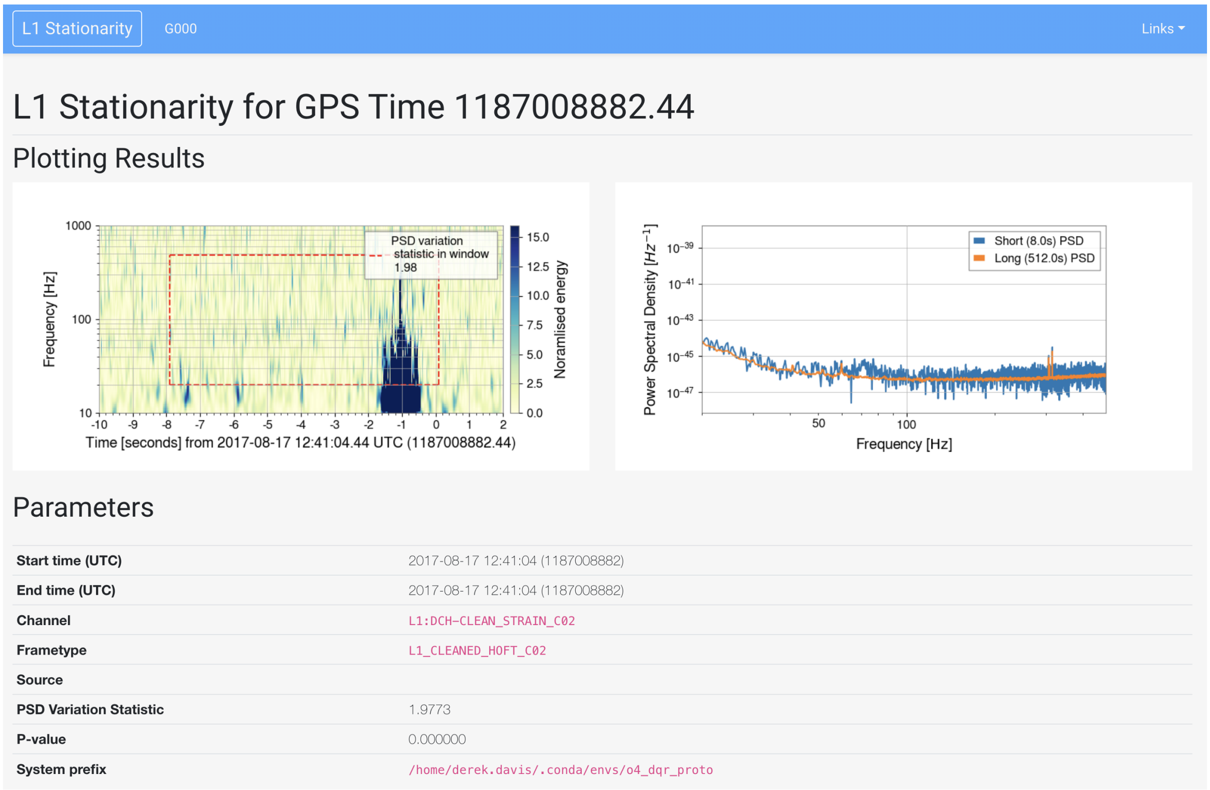
Stationarity Check task

This task is designed to identify if the noise spectrum during the time of a candidate event is elevated with respect to other nearby time periods [17]. It computes the variance of the SNR calculation from matched filtering, giving an idea of the stationarity of the detector data. The task outputs two plots: an omega scan with a box highlighting the data used for the stationarity calculation, along with the calculated PSD variation statistic, and a plot of the PSD variation near the GW candidate event. For the second plot, two PSDs are used: a short-duration PSD (often 8 seconds long), and a long-duration PSD (512 seconds long). Figure 7.2 shows an example of the resulting web-page generated for GW170817 by the Stationarity check in the Livingston Observatory. Note that the calculated $p-$value is 0, indicating a Data Quality Issue in the proximity of this candidate event.

---

[1] I only explain the three tasks (additional to GSpyNetTree) with public documentation available in `https://detchar.docs.ligo.org/dqrtasks/index.html`. The documentation of GSpyNetTree is found in Section 13.1.

[2] As a matter of fact, the first GW event of O4 was retracted: `https://gcn.nasa.gov/circulars/33871?query=S230524x`



L1 Stationarity for GPS Time 1187008882.44

Plotting Results

Parameters

Start time (UTC)	2017-08-17 12:41:04 (1187008882)
End time (UTC)	2017-08-17 12:41:04 (1187008882)
Channel	`L1:DCH-CLEAN_STRAIN_C02`
Frametype	`L1_CLEANED_HOFT_C02`
Source	
PSD Variation Statistic	1.9773
P-value	0.000000
System prefix	`/home/derek.davis/.conda/envs/o4_dqr_proto`

Figure 7.2: Result web-page of the Stationarity Check for GW170817 in the Livingston Observatory. Note that the calculated p-value is 0, which means that a Data Quality Issue was identified by this task. (Reproduced from [17]).

PEMcheck task

The PEMcheck task aims to detect the times and frequencies in which environmental noise contaminates a candidate GW signal. Given a broad array of environmental channels that monitor different instrumental/environmental aspects of the detector, the PEMcheck reports the channel(s) (if any) that might have triggered a false GW candidate. This is done using a coupling function that depends on the channel [17].

Rayleigh Statistic task

The Rayleigh task measures whether the coefficient of variation of the PSD is significantly different from nearby segments of data. The coefficient of variation (CV) is defined as the ratio of the standard deviation, σ, to the mean μ of the PSD, $c_v = \sigma/\mu$. The task uses stretches of data near the time of the GW event candidate, such that if the data preceding the event is equally non-stationary as the one containing the signal (defined by the c_v), no Data Quality issue is flagged [17].

Glitch Find task

The Glitch Find task calculates the energy of the tiles of the Q-transform and verifies if they are Gaussian (i.e., the usual noise of LIGO and Virgo) or not. If not, it selects the non-Gaussian times (that do not correspond to a GW signal) and flags them. The task generates a spectrogram of the initial event, a second spectrogram with the reconstructed glitch (if any), and the energy of the Gaussian data of the Q-tiles of the Q-transform for the initial spectrogram. Figure 7.3 shows an example of the resulting web-page generated for GW170817 by the Glitch Find task in the Livingston Observatory, with a $p-$value < 0.000001 [17].

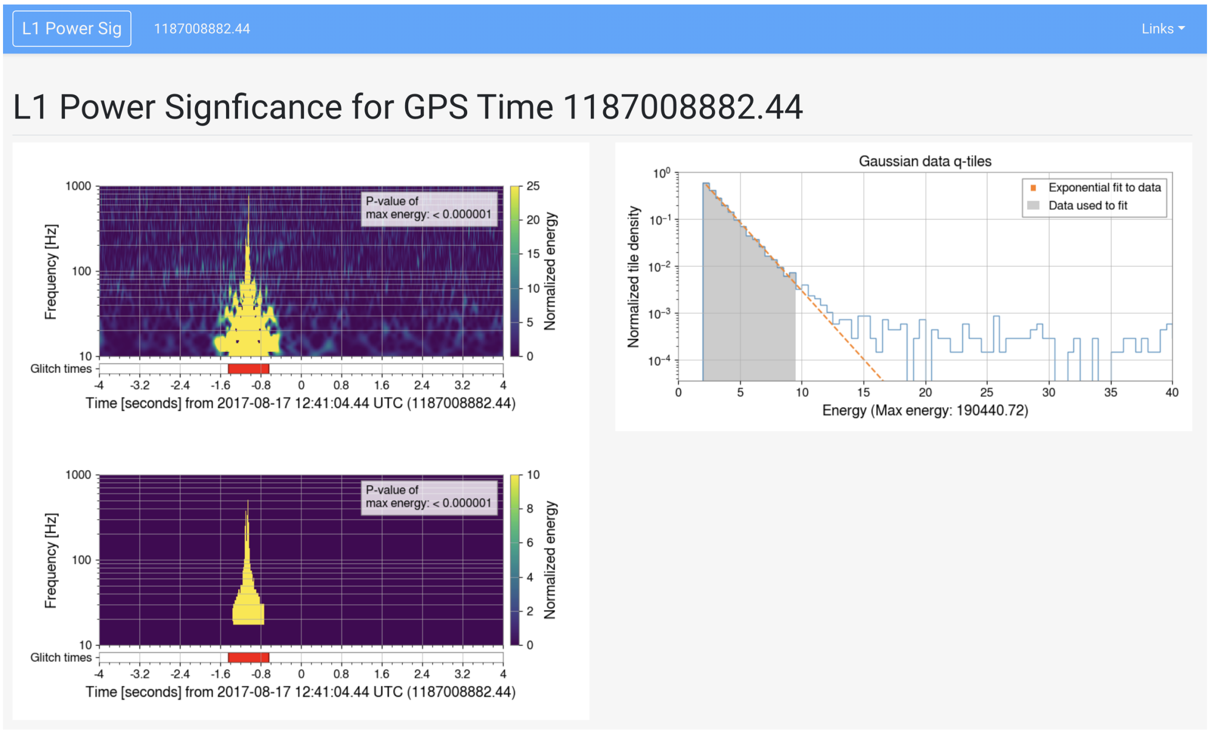

L1 Power Signficance for GPS Time 1187008882.44

Figure 7.3: Result web-page of the Glitch Find task for GW170817 in the Livingston Observatory. Note that the times corresponding to the glitch are flagged on the top spectrogram, and a new spectrogram (at the bottom) is generated with the energy of the glitch found. The GW signal of GW170817 can be faintly seen behind the glitch on the spectrogram at the top. The plot on the right shows the distribution of energy in the Q-tiles of the spectrogram on the left, used the generate the spectrogram at the bottom. (Reproduced from [17]).

8 Machine Learning: the basis of GSpyNetTree

Having studied where GSpyNetTree is hosted in the LIGO-Virgo event validation pipeline, we now turn to introduce the fundamentals of Machine Learning: the basis for building GSpyNetTree. We start by giving an introduction to Machine Learning, followed by a description of supervised methods. We end this chapter by giving a description of Gravity Spy [3], the Machine-Learning citizen-science project on which GSpyNetTree is based, and InceptionV3 [15], a state-of-the-art Machine Learning application.

Machine Learning (ML) is the science of programming computers so they can learn from data [100]. ML is particularly useful because it applies techniques that make an algorithm automatically learn to differentiate between different classes of inputs. In the case of GSpyNetTree, ML constitutes an additional validation technique to the already existing DQR tasks that identifies glitches in the proximity of a GW event candidate.

Machine Learning also has the advantage of automatically adapting to changes in data, as shown in Figure 8.1, which is particularly useful for LIGO-Virgo data. With evolving detectors using new instrumentation, background noise and glitch types change frequently. An ML model can adapt to such changes. Additionally, ML is also useful to gain insights of complex problems and large amounts of data, which is useful in LIGO-Virgo due to the vast amount of data produced by the detectors.

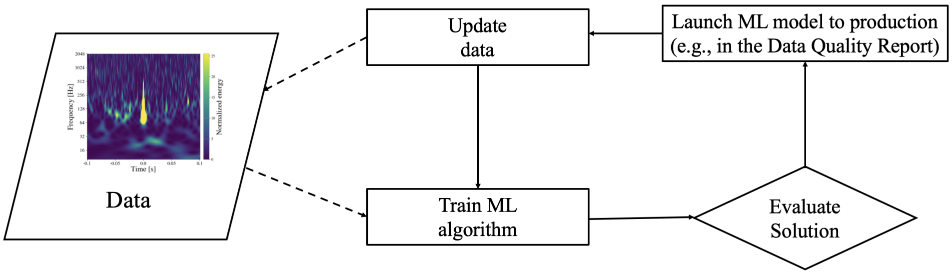


Figure 8.1: Schema of an automatically adapting to change architecture based on Machine Learning, for gravitational-wave and glitch data. Note that the process of training, evaluating, launching a new model, and updating the data can be automated. (Adapted from [100]).

There are several types of ML algorithms, which can be classified into different categories depending on several criteria. One of them is whether or not the models are trained with human supervision. If so, they are called **supervised models**; if not, they could be unsupervised, semisupervised, or based on Reinforcement Learning. [100]. The following section describes the supervised methods, which are the ones used for GSpyNeTree's classifications[1].

---

[1] For a description on the unsupervised/semisupervised learning, refer to Géron [100].

8.1. Supervised Learning

In supervised learning, the training set that is fed to the algorithm includes a label for each sample. A label is (usually) previously classified by a human and is the desired output of a given sample. One of the main types of supervised learning is classification, and this is the type of task that GSpyNetTree implements. Having learned how to classify different GWs and glitches, it must learn how to classify new events. There exist a lot of supervised learning models, such as Random Forests and Support Vector Machines (SVMs). However, some of the most powerful algorithms are Artificial Neural Networks (ANNs)[2] (see Section 8.2). A particular type of ANNs, known as Convolutional Neural Networks (CNNs) are widely used for image classification tasks. These are described in Section 8.3.

Additionally, in supervised learning, each sample is a pair consisting of an input object (the GW or glitch spectrograms, in the case of GSpyNetTree) and a desired output value (the name of such input, e.g., GW, Blip, Low-frequency Blip, or Koi Fish for GSpyNetTree). Having analyzed and learned information from the training data, supervised learning produces an inferred function (or model), which can be used for classifying new examples. In order to accurately predict the label(s) of unseen instances, the model must generalize adequately. Its ability to make good predictions highly depends on the quality and quantity of samples in the training sets, and the ML architecture used to generate the model. Classification tasks can be further split into several categories, depending on how many classes (and possible outputs per sample) each problem admits. The three fundamental types of classification tasks are binary, multi-class, and multi-label [100]. We focus on the last two as GSpyNetTree evolved from a multi-class (see Chapter 10) to a multi-label (see Chapter 12) architecture for O4.

8.1.1. Multi-class vs multi-label architectures

While binary classifiers distinguish between two different classes only, a **multi-class** classifier, as its name suggests, can distinguish between more than two classes [100]. In general, in multi-class classifiers, each label is assigned a probability between 0 and 1, such that the sum of the probabilities of all classes adds up to 1. It is important to highlight that, in a multi-class classifier, each sample can be assigned one and only one label. This way, multi-class classifiers cannot predict accurately, say, a GW candidate overlapping with a glitch. Examples of multi-class classifiers are Gravity Spy (see Section 8.3.1) and the first version of GSpyNetTree (see Chapter 10).

There are cases, however, in which it is relevant to output multiple classes for each instance (e.g., in the event of an overlapping GW signal and a glitch, we want to predict both; to detect not only the astrophysical signal but also the glitch, and be able to mitigate it). A classifier that is able to predict multiple labels is called a **multi-label** classification system [100]. An important aspect of multi-label classifiers is that they can predict zero or more classes per instance, by returning a probability ranging from 0 to 1 for each considered class. This way, the sum of the probabilities of all labels is not 1 (as occurs for multi-class classifiers, where the classes are mutually exclusive). Instead, the probability of each label can take any value from 0 to 1, and

---

[2] Note that these can be used in unsupervised problems too.

a label is said to be predicted by the classifier if its probability is greater than or equal to 0.5. In the case where no label surpasses the 50% threshold, no labels are predicted[3] [17].

8.2. Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs), or simply Neural Networks, are a type of supervised learning algorithms widely used in Machine Learning classification and regression tasks. Their name and structure are inspired by the human brain, as they were first designed to mimic how biological neurons transmit information to each other.

In general, an ANN is made up of several layers of nodes or *neurons*. Each has an input layer, one or more hidden layers, and an output layer, as shown in Figure 8.2. The depth of the ANN is given by the number of ℓ hidden layers it has. For classification problems, the number of neurons k in the output class corresponds to the number of labels in the problem. Each neuron connects to another one and has an associated weight and threshold. If the output of any individual node is above a pre-defined threshold, which depends on an activation function, the neuron is activated and it will send the information that triggered its response to the following layer of the network. Otherwise, no information is sent to the next layer by that particular neuron.

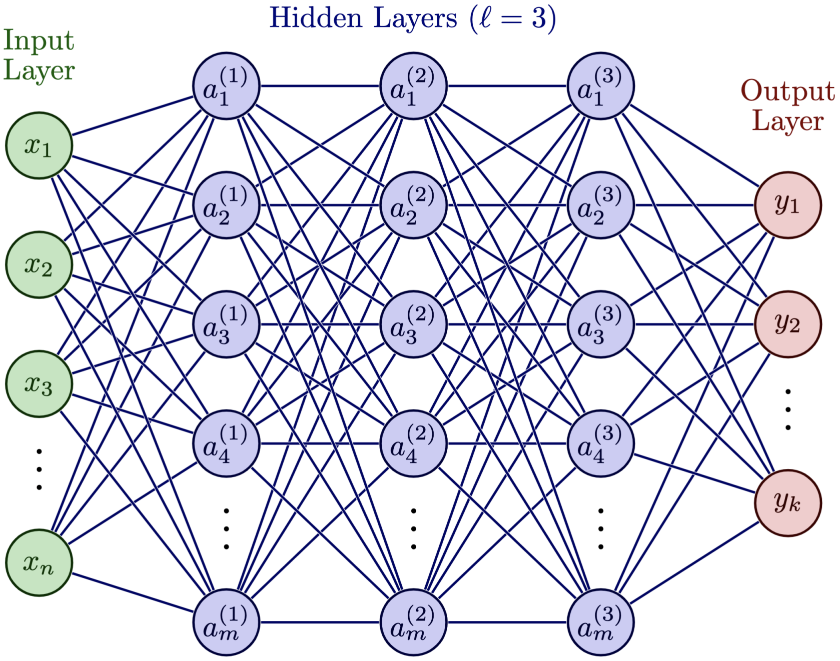


Figure 8.2: Schematic diagram of an Artificial Neural Network (ANN) consisting of an input layer (in green), three hidden layers ($\ell = 3$, in blue), and an output layer (in red). The ANN has n inputs (which is why there are n neurons in the input layer) and k output classes, and each hidden layer has m neurons. The subscript represents the $i-$th neuron in each layer, and the superscript in the hidden layers denotes the $j-$th layer, $1 \leq j \leq \ell$.

The output h_i of the $i-$th neuron of a given layer is calculated as [100]:

$$h_i(x_i) = \phi(\mathbf{x}_i^\top \mathbf{w}_i + b_i), \tag{8.1}$$

---

[3] These cases constitute an opportunity to verify the data that is being predicted by the classifier and consider other ML architectures, like unsupervised learning.

where x_i is the vector of input features, w_i is the vector of weights (which helps determine the importance of a particular variable: larger weights contribute more to the output compared with other inputs), b_i is a bias (or threshold), and ϕ is the activation function that determines whether information should be sent by the $i-$th neuron to the next layer or not. There are several activation functions, like the reLU (Rectified Linear Unit),

$$f(\mathbf{x}) = \max(0, x), \tag{8.2}$$

the sigmoid function,

$$f(\mathbf{x}) = \frac{1}{1+e^{-x}}, \tag{8.3}$$

shown in Figure 8.3, and the softmax function,

$$f(\mathbf{x}) = \frac{e^{\mathbf{x}^\top \mathbf{w}}}{\sum_{k=1}^{K} e^{\mathbf{x}^\top \mathbf{w}_k}}, \tag{8.4}$$

for K classes. The softmax function is often used as the activation function of the output layer of a multi-class ANN as it normalizes the output of a network to a probability distribution, such that the highest probability is assigned to the neuron (class) with larger weights, and the probabilities from all classes add up to one. This class will be the predicted label of a given sample. On the other hand, the reLU and sigmoid functions are used as activation functions of the neurons of the hidden layers (the most suitable function to use depends on each particular task).

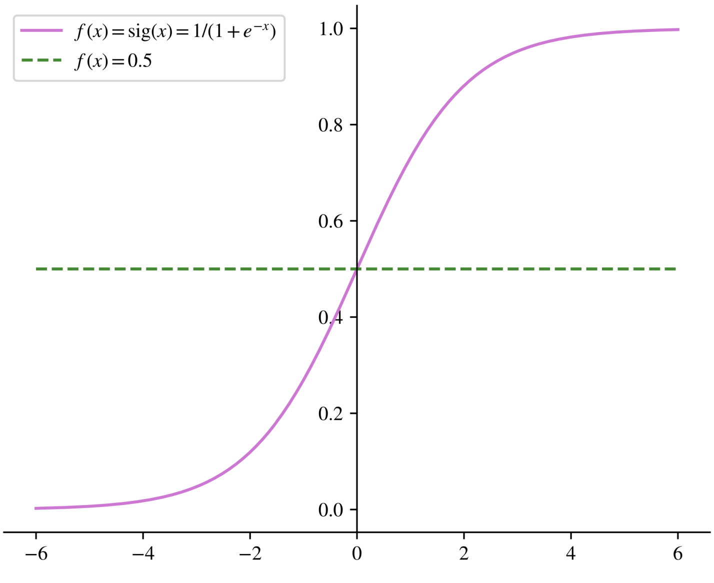


Figure 8.3: Plot of a sigmoid function, $-6 \leq x \leq 6$. Note that the range of the function is defined between 0 and 1, and the line $y = 0.5$ is included for reference.

It is important to highlight that the sigmoid function is also used as the activation function of the output layer of a multi-label ANN. This way, each neuron will have a probability $0 \leq p \leq 1$, which is rounded up to 0 if $p \leq 0.5$ or to 1 otherwise. This way, multiple neurons can end up with a value of 1 (or zero), so that several (or none) classes are predicted, instead of 1 and only 1 (as in the softmax activation function for multi-class classifiers).

8.2.1. Loss functions: optimizing a model

An ANN is trained for several epochs, in which the weights learned from the previous iteration, are the starting weights of the subsequent epoch. In each epoch, a cost (or loss) function is minimized to maximize the number of correct predictions (note that several metrics can be used for this purpose, see Subsection 8.2.2). The loss function tracks incorrect labeling of the data class(es) by a model, penalizing it if deviations in probability occur in the classification of labels. This way, low loss function values result in an increased number of correct predictions.

The cost function depends on the task. In the case of multi-class classification, the categorical cross-entropy is used [100]:

$$\text{CCE loss} = -\sum_{i=1}^{K} y_i \cdot \log \hat{y}_i, \tag{8.5}$$

with y_i and $\hat{y}_i$ the real class and predicted class of a specific sample, respectively, and K the number of classes. Note the similarity with the Shannon entropy. In the case of a binary or multi-label classifier, a binary cross-entropy function is used [100]:

$$\text{BCE loss} = -\frac{1}{K}\sum_{i=1}^{K} y_i \log \hat{y}_i + (1-y_i)\cdot\log(1-\hat{y}_i), \tag{8.6}$$

with $K = 2$ in the case of binary classifiers. Multi-label problems use this loss function as they are actually a collection of multiple binary classification subtasks. This way, if a multi-label problem consists of three non-mutually exclusive labels (say, GW, Blip glitch, and Tomte glitch), the binary cross-entropy will be calculated in a binary fashion (determining separately if the sample is a GW or not, a Blip glitch or not, and a Tomte glitch or not) and, at the end, adding all the calculated values. Note that the binary cross entropy of each label is independent of all the others.

8.2.2. Metrics: how is a model evaluated?

A cost function is used to maximize the number of correct predictions, and several metrics can be used to quantify this number. Particularly, in GSpyNetTree, we use two metrics: the accuracy and the recall, each of which is explained below.

The accuracy is a global metric, and evaluates the number of correctly classified samples, out of the total number of samples. That is, if 90 samples out of 100 were classified correctly, then the accuracy will be 90%. While this metric works perfectly to evaluate the performance of a multi-class classifier (like the pre-O4 version of GSpyNetTree), note that the definition of this metric is problematic for a multi-label classifier; as it does not consider the prediction of individual labels per sample, but the entire prediction (i.e., all labels) for that sample as a whole. This means that, if the real labels of a sample are X and Y, and the CNN only predicts X, the entire sample will be considered incorrect by the accuracy. However, this is undesirable, and results in underestimations of the real performance of the model. In the aforementioned case, the X class should be labeled as correctly classified, whereas the Y class should be penalized by the metric. To solve these issues, other metrics can be used. We explain the recall, which is the one used by GSpyNetTree in its multi-label fashion.

The recall, also known as sensitivity, measures how many samples of an actual class were correctly classified by the CNN. That is, if a sample of class X is predicted as an instance of class Y, the recall will be lower. The exact definition of recall is [100]:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}. \tag{8.7}$$

Note that the recall aims to reduce the number of instances with particular labels that are misclassified with a different set (or subset) of labels. This is the ideal metric for GSpyNetTree, as the recall will aim to minimize the number of GWs classified as glitches, and vice-versa. As a matter of fact, in the case of GSpyNetTree, it will always try to avoid misclassifying as many samples of a particular class as it can, even if that increases the false positive rate of a particular class (of course, increasing it substantially is also detrimental to GSpyNetTree as its predictions would not be confident). Nevertheless, avoiding the misclassification of as many samples as possible (even between different glitch classes) assures the minimization of the Data Quality issues that GSpyNetTree fails to flag.

8.2.3. Hyperparameter selection: Fine-tuning a model

After selecting a model, the metric to monitor it, and training it, another fundamental aspect of Machine Learning is the selection of hyperparameters. These hyperparameters are not parameters of the model (i.e., they have nothing to do with the problem and the model itself), but are adjustable arguments that need to be carefully fine-tuned or set based on ML investigations.

One of the most important hyperparameters to select is the optimizer, and its selection depends on the task [100]. An optimizer is an algorithm that aims to adjust the weights of an ML model, usually looking to minimize the loss function with each epoch. For image classification tasks, usually, the best optimizer is Stochastic Gradient Descent (SGD) [101]; however, in other tasks, the usage of an adaptive algorithm gives better results. For the SGD, there are further hyperparameters that should be fine-tuned, including the learning rate (which indicates how much model weights should be updated in each epoch) [100] and the momentum (a parameter which aims to accelerate the convergence of the SGD vectors to an optimal solution).

Finally, a fundamental hyperparameter that needs to be fine-tuned is the batch size, which corresponds to the number of samples that the model is trained on per epoch. Several studies have been performed and, while many authors agree that batch sizes should not exceed 32 [102], others believe that, depending on the task, it could be larger. In the case of GSpyNetTree, we always use a batch size of 32, as this is a standard value used in many ML applications.

8.3. Convolutional Neural Networks

Having studied the way in which ANNs work, and how they are evaluated, optimized, and fine-tuned, we now turn to study Convolutional Neural Networks (CNNs), which are a specific type of ANN widely used in image classification tasks [100].

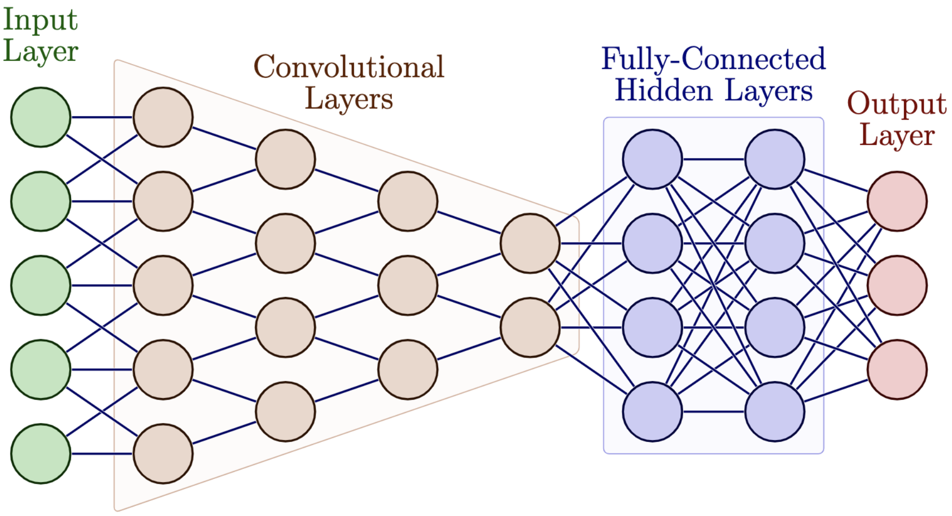


Figure 8.4: Schematic diagram of a Convolutional Neural Network (CNN), consisting of various convolutional layers, pooling layers, fully connected layers, and an output layer. While not shown here, pooling layers (usually used between convolutional layers) are often used to avoid overfitting in the models.

CNNs provide an scalable approach to image classification tasks, leveraging principles of linear algebra to identify patterns within a given sample. CNNs have five main types of layers: Input layer (usually an image), Convolutional layers (after the input layer, the initial layer of the CNN. Can be more than one), Pooling layers (between convolutional layers), Fully Connected layers, and an Output layer (the number of neurons in this layer depends of the number of classes/labels of the CNN). An example of a CNN is shown in Figure 8.4.

Each of these layers works in a different way; however, in general terms, the complexity of the CNN increases as we go deeper [100, 103]. The focus of the early layers is given to simple features, like colors or edges, which is the reason why many CNNs can be used for Transfer Learning (i.e., using the initial weights learned with a different training set in a new problem). Deeper in the network, the CNN recognizes larger elements or shapes (e.g., it starts to finally identify the morphology of a GW/glitch) until it finally learns to recognize the images it is trained on. While the CNN is being trained, it is important to avoid overfitting. Between these learning steps, pooling layers are used to reduce the parameters and complexity, and improve efficiency. Note that many pooling layers can also have a detrimental effect on the predictions of the CNN, as the information learned can be lost.

While several CNN architectures exist, we will study two of them: the Gravity Spy architecture [3], as it constitutes the base of the first version of GSpyNetTree, and the InceptionV3 architecture [15], which is the state-of-the-art version that our tool leverages.

8.3.1. Gravity Spy architecture

Gravity Spy [3, 88] is a citizen Science project that leverages a CNN architecture that accurately classifies more than 20 glitch classes[4]. It is a widely used tool within the LIGO-Virgo

---

[4] The original Gravity Spy had 20 classes [3], but now it has approximately 23 [88].

collaboration for glitch detection and has had outstanding results in the previous observing runs. Gravity Spy's architecture is shown in Figure 8.5. It consists of two sets of Convolutional + Pooling layers applied consecutively, followed by a fully connected layer and an output layer. While this is a powerful architecture, several studies have built state-of-the-art CNN architectures which leverage deeper networks that perform better in extracting CNN features. One of them is InceptionV3 [15], briefly described in the next Section.

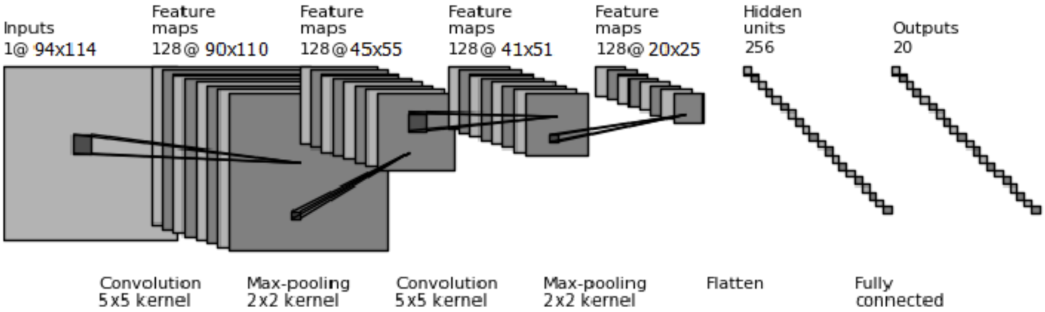


Figure 8.5: Convolutional Neural Network architecture of Gravity Spy [3], LIGO-Virgo's ML glitch classifier. Following the input layer, two sets of consecutive Convolutional layer + Pooling layer are applied. After the last pooling layer, the features are flattened and sent to a fully connected layer, which is then connected to the output of 20 neurons (20 neurons because there are 20 glitch classes). (Reproduced from [3]).

8.4. InceptionV3

InceptionV3 [15], is one of the most used state-of-the-art CNNs for image classification problems. It is a Deep architecture, consisting of 42 layers; note the difference in depth compared to the Gravity Spy architecture.

In order to leverage a powerful architecture that optimizes training time, InceptionV3 uses factorized convolutions (i.e., it splits the convolutional work among several, parallel Convolutional layers). This allows the extraction of characteristics and reduces the number of parameters of the network considerably [15]. Additionally, it also has auxiliary classifiers between layers which, during training, serve as regularizers that prevent overfitting.

A comparison of the InceptionV3 and Gravity Spy's architecture is shown in Figure 8.6. Note the robustness of InceptionV3 compared to Gravity Spy, which allows the former to perform better on image classification tasks and extract more (and/or better) features from the samples it is trained on.

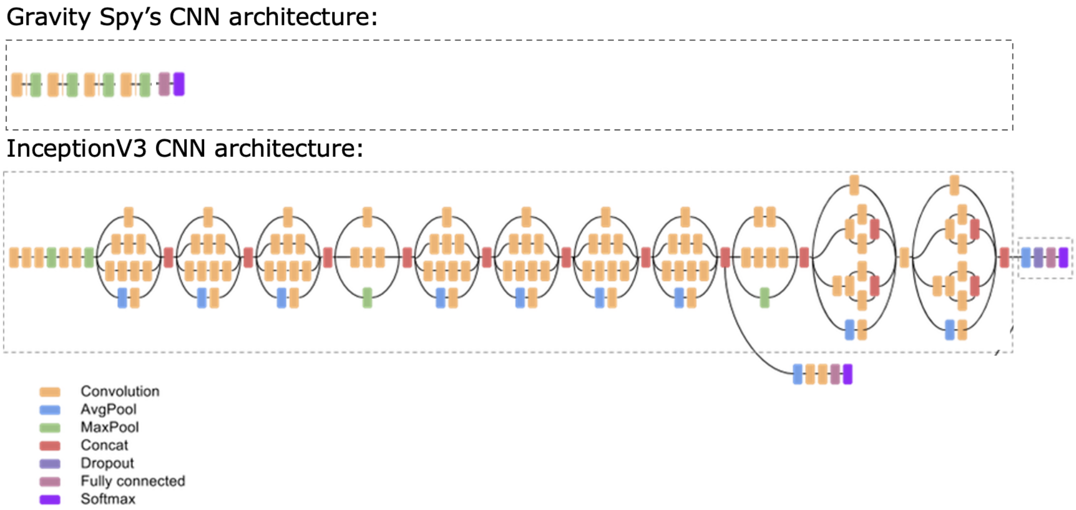


Figure 8.6: Comparison of the Convolutional Neural Network architecture of Gravity Spy (upper panel) [3] and InceptionV3’s architecture [15]. Inception is a more robust, deeper network, which makes it ideal for complex image classification tasks. (Reproduced from [14], which is in turn adapted from [15, 104]).

9 Setting the base of GSpyNetTree: building a GW signal-vs-glitch classifier based on Gravity Spy

In order to build a GW signal-vs-glitch classifier based on Gravity Spy, several changes need to be implemented to the original architecture[1]. GSpyNetTree's initial purpose was to confidently predict GW signals in a given sample. However, as the needs of the LIGO Scientific Collaboration Detector Characterization team evolved, so did GSPyNetTree.

The first version of GSpyNetTree was a decision tree of multi-class CNN classifiers, which was build upon several recommendations proposed by Jarov et al. [13] (see Chapter 10). This work outlines a new multi-classifier method to leverage prior Gravity Spy architecture to distinguish GWs from glitches. Considering previous investigations on improving Gravity Spy that have shown that its inaccuracies in glitch classification tend to be higher in poorly represented classes in the CNN's training set [105], Jarov et al.'s study recommends significant changes to Gravity Spy for the purpose of a signal-vs-glitch classifier [14].

First, it recommends augmenting the data of the Chirp class (consisting of data from hardware injections that emulate the behavior of GWs by displacing the detector's test masses [106]), which is morphologically similar to the typical GW events seen in O3 [88]. However, instead of using hardware injections, it uses GW software simulations that allow the generation of more GW samples, compared to the severely underrepresented Chirp class in the original Gravity Spy training set [3]. It also recommends deploying specialized training sets to handle different ranges of total candidate signal mass, as low and high-mass mergers have very distinct morphologies, which may be more prone to confusion with particular glitch classes (see Figure 9.1) [14].

We developed this proposed signal-vs-glitch multi-classifier architecture and analyzed its readiness for O4, as described in Chapter 10. After generating the specialized training sets based on GW and glitch morphology and incorporating the data augmentation suggested by Jarov et al. [13], we built three different classifiers, one per training set, with the same CNN architecture Gravity Spy leverages. With the three signal-vs-glitch classifiers, we made a decision tree sorted via total GW candidate mass, constituting the base for GSpyNetTree, the Gravity Spy Convolutional Neural Network Decision Tree. During the current observing run, O4, this tool will intake GW candidate events from GraceDB [98] and classify them as GWs or glitches as part of the LIGO-Virgo Data Quality Report [16] (see Section 7.1.6) [14].

After building GSpyNetTree according to the recommendations in Jarov et al. [13], we noted that the original Gravity Spy CNN architecture could be further improved (see Chapter 11). We decided to use Inception V3 [15], Google's state-of-the-art CNN for image classification tasks. We noted a vast improvement in classification accuracy, particularly for the GW class. We also considered the additional changes expected for O4, including different persistent noise subtraction for calibrated data, new potential noise sources, a high expected detection rate [74],

---

[1] Note that this is done for the purposes of our tool; Gravity Spy is a widely used tool within LIGO and Virgo for glitch classification purposes only.

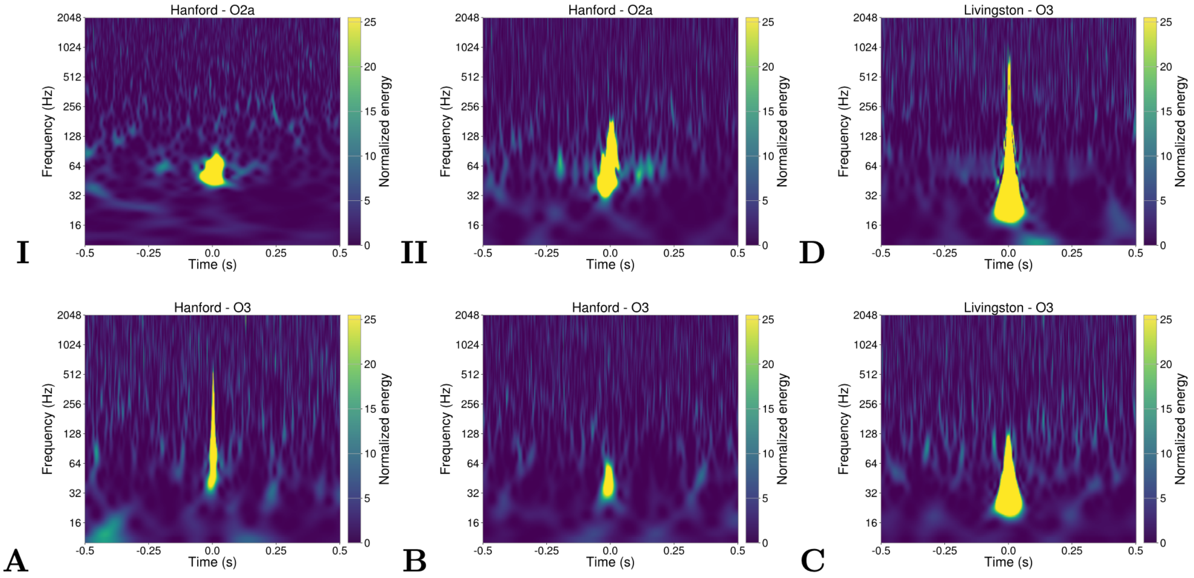


Figure 9.1: Examples of CBC GW signals (I and II) along with morphologically similar glitches (A, B, C, D) commonly confused by the glitch classifier Gravity Spy, according to [13]. A simulated GW signal with a total mass of 245 $M_\odot$ (I) is more prone to be confused with Blips (a) and Low-frequency Blips (b). In contrast, a 126 $M_\odot$ signal with a total SNR of 35 (II) shares morphological similarities with a Tomte (C) and a Koi Fish (D). (Reproduced from [13]).

and the likely occurrence of overlapping glitches and GW signals in time-frequency visualizations [14].

For the first version of GSpyNetTree, we performed three validation studies based on the classification challenges posed by the O4 sensitivity increase. Additionally, we propose solutions to overcome them for the O4-era version of GSpyNetTree (see Chapter 12). First, we study how the first version of GSpyNetTree responds to data with non-linear subtraction of 60 Hz AC power artifacts, as expected for low latency LIGO data in O4. We evaluate its readiness for the new background noise expected in O4 and its transferability to Virgo data, which has a different noise background from the two LIGO detectors. We also test how GSpyNetTree responds to glitches not included in its training set, as new noise sources are expected to appear during O4. We curate a variety of glitches covering several cases of interest, including frequently occurring glitches and glitches morphologically similar to others already included in the training set. We then evaluate how GSpyNetTree responds to GW candidates overlapping with glitches, which is even more likely to happen during O4 than O3, given the higher expected detection rate. In O3 only, 24% of candidates overlapped with one or more glitches [7, 8], which makes it a relevant case of interest to tackle. This first version of GSpyNetTree is based on Alvarez-Lopez[2] et. al [14].

Chapter 12 shows the O4-era version of GSpyNetTree. Implementing the suggestions of the first era version of GSpyNetTree [14], we built a new decision tree of signal-vs-glitch multi-label classifiers, including: (1) instances of Virgo glitches, (2) instances of Hanford/Livingston glitches with the 60 Hz subtraction calibration, and (3) overlapping samples of GWs and glitches. We

---

[2] This paper is currently under review in Classical and Quantum Gravity, and a preprint can be found in the arXiv: `https://arxiv.org/abs/2304.09977`.

also improve several other methods in dataset construction with respect to the first version of GSpyNetTree, and include new glitches that are very likely to happen during O4 based on Detector Characterization analyses, but not considered in the first version of our tool. In Chapter 13 we discuss the deployment of GSpyNetTree as part of the DQR in the LIGO-Virgo event validation pipeline. Finally, in Chapter 14, we evaluate our results per classifier, and perform three validation studies: on glitches not included in the training set (but on Gravity Spy), on some GWTC-3 (third gravitational-wave transient catalog) candidates [7, 8], and on a few of the public superevents detected by LIGO-Virgo-KAGRA so far in the current observing run.

10 Pre-O4 GSpyNetTree: a decision tree of multi-class signal-vs-glitch classifiers

Building upon the proof of principle described in Jarov et al. [13], GSpyNetTree leverages a decision tree of three CNN classifiers, each trained on a specialized and balanced set of GWs and morphologically similar glitches, sorted via estimated candidate mass metadata, obtained from GraceDB. The three classifiers are: the low-mass (LM) classifier (for candidates with an estimated total mass below 50 $M_\odot$), the high-mass (HM) classifier (for candidates with an estimated total mass between 50 $M_\odot$ and 250 $M_\odot$), and the extremely high-mass (EHM) classifier (for candidates with an estimated total mass above 250 $M_\odot$). Depending on the mass estimate provided via GraceDB [98], each candidate GW event is sent to the LM CNN or HM CNN to determine whether it is astrophysical or a glitch. If the candidate's mass is above 250 $M_\odot$, it is then sent from the HM classifier to the EHM classifier for more accurate classification. Figure 10.1 shows examples of simulated GW signals in each of the mass ranges, and Figure 10.2 shows the glitches they are morphologically similar to. Table 10.1 specifies the glitch classes considered for each mass range depending on morphological similarities, as considered by Jarov et al. [13].

Low-Mass (LM) Classifier		**High-Mass (HM) Classifier**		**Extremely High-Mass (EHM) Classifier**	
Class	**Samples**	**Class**	**Samples**	**Class**	**Samples**
GW (3-50 $M_\odot$)	1000	GW (50-250 $M_\odot$)	1000	GW (250-350 $M_\odot$)	1000
Blip	999	Blip	999	Blip	999
Low-Frequency Blip	1039	Low-Frequency Blip	1039	Low-Frequency Blip	1039
No Glitch	1017	No Glitch	1017	No Glitch	1017
Scratchy	1093	Koi Fish	990		
		Tomte	758		

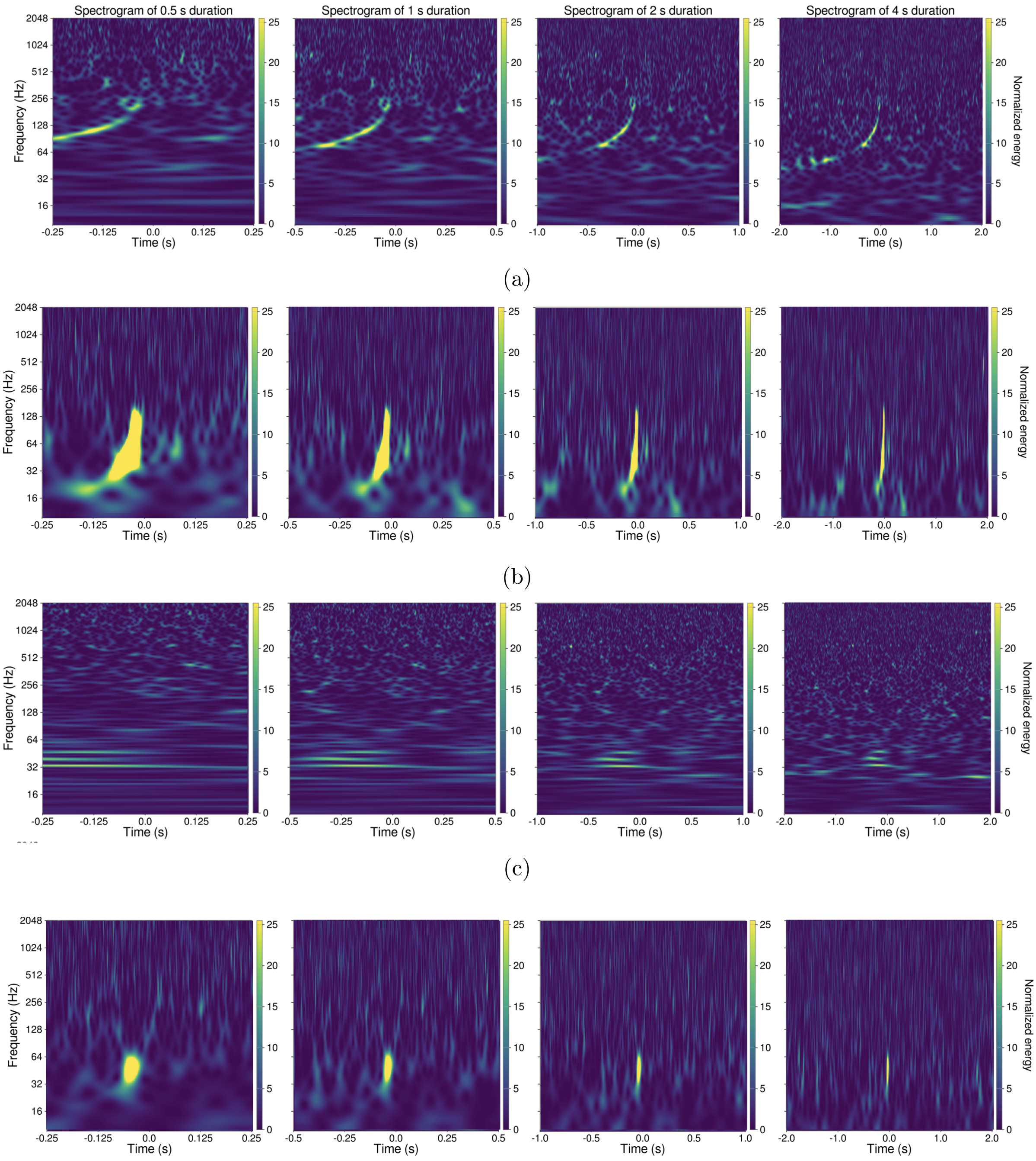
Table 10.1: Classes and number of samples (before time-offset augmentation, which add four more examples for each sample listed) per class for each of the GSpyNetTree's classifiers. The GW mass ranges listed indicate the target total mass of an event candidate.

The EHM classifier presents additional challenges that need to be addressed. Low-frequency blips share strong similarities in duration, frequency range, and morphology with EHM mergers. Indeed, the original Gravity Spy model misclassifies EHM mergers as Low-frequency blips with 99% confidence [13]. However, since the mass range of detected GWs is expected to increase in each observing run, it is essential to make GS pyNetTree robust for these possible future detections. GSpyNetTree incorporates a spectrogram scaling technique that Jarov et al. showed to be of great utility in this mass range [13]: we apply the Mercator projection, which stretches the signals vertically, scaling the image features to better segregate GW signals from Low-frequency blips[1] (see Figure 10.3).

Additionally, when dealing with CNN training sets, it is important to consider data augmentation techniques to avoid overfitting. One strategy is to generate slightly modified versions of existing samples, increasing the amount of training data. Thus, we generated four random time

---

[1] The Mercator projection is not used in the O4-era version of GSpyNetTree, as we included new glitches in the EHM Classifier and this transformation affected the performance of the CNN (see Chapter 12).



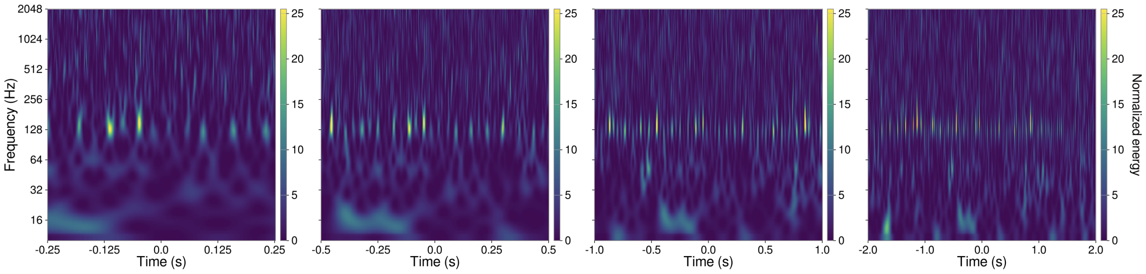
(a)

(b)

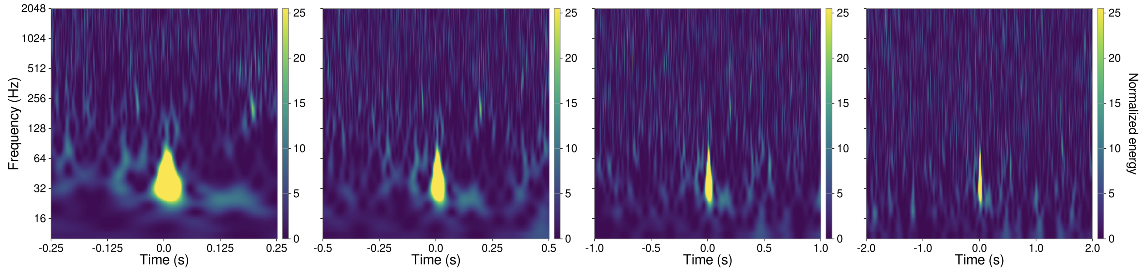
(c)

(d)

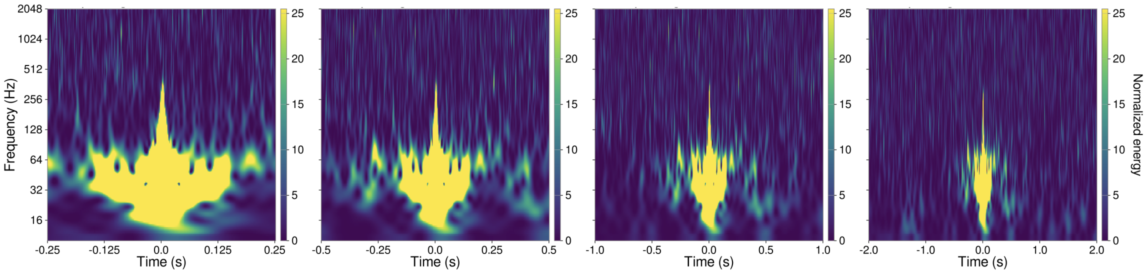
Figure 10.1: Examples of spectrograms of simulated GW signals with all four durations used in the training set of GSpyNetTree (0.5 s, 1 s, 2 s, and 4 s). (a) Software simulated GW signal in the low mass regime, with a total mass of 29.2 $M_\odot$. (b) Simulated GW signal in the high mass regime, with a total mass of 118.7 $M_\odot$. (c) Simulated GW signal in the high mass regime, with a total mass of 182 $M_\odot$, but with a significantly lower signal-to-noise ratio (SNR) than the example shown in Figure 10.1b. Signals with a low SNR may occur in any of the mass ranges, and they may be similar to the No Glitch class (Figure 10.2e). (d) Simulated GW signal in the extremely high mass regime, with a total mass of 283 $M_\odot$. These signals are morphologically similar to Low-frequency blips (Figure 10.2b).



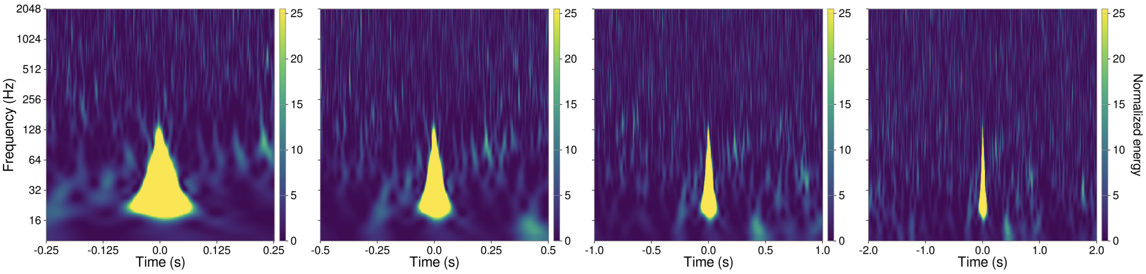
(a) Example of a Scratchy glitch.



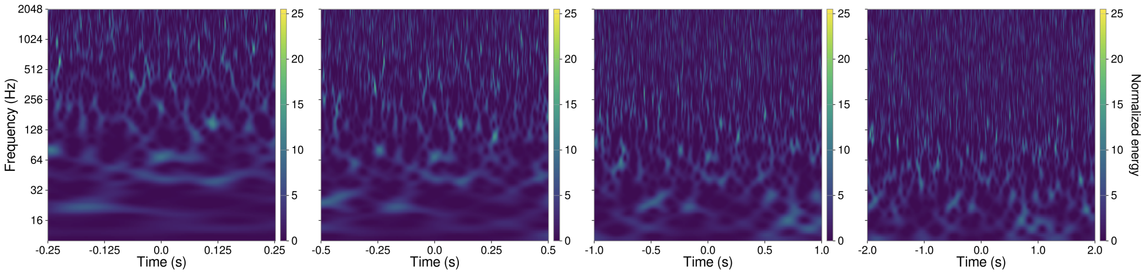
(b) Example of a Low-frequency blip glitch.



(c) Example of a Koi Fish glitch.

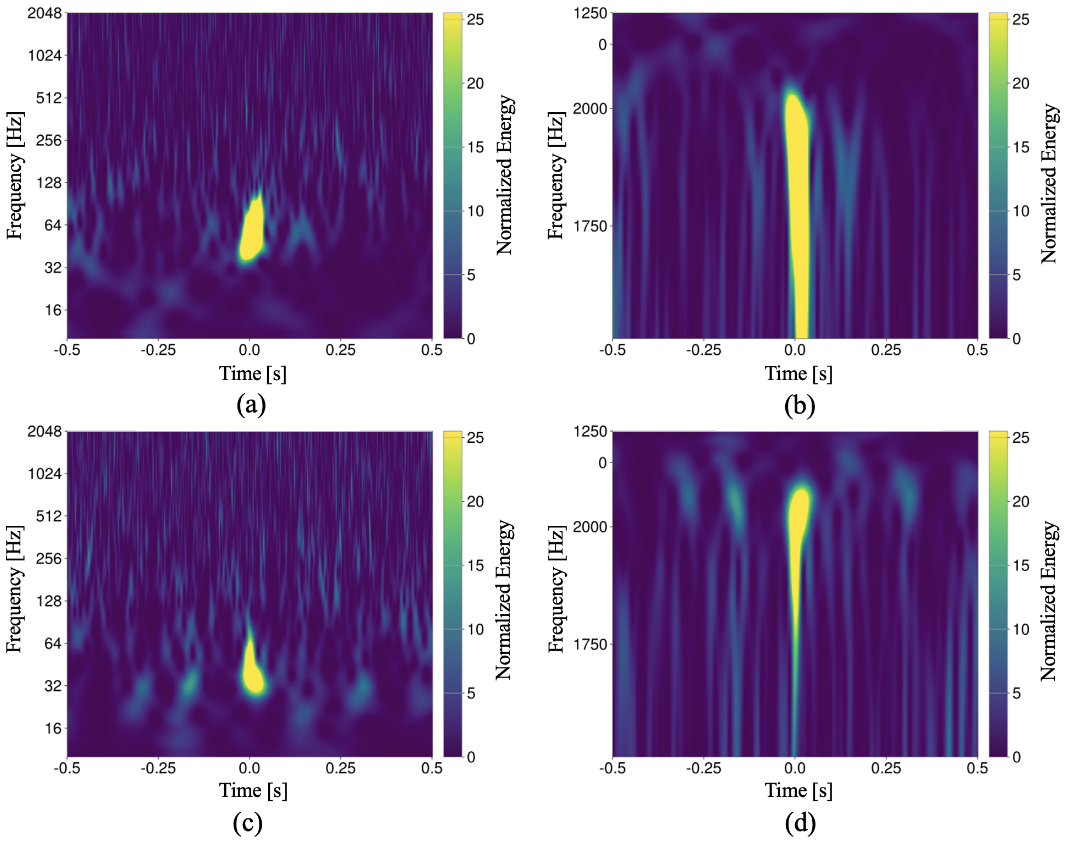


(d) Example of a Tomte glitch.



(e) Example of a No Glitch.

Figure 10.2: Examples of selected glitches with all four durations used in GSpyNetTree (0.5 s, 1 s, 2 s, and 4 s). These non-astrophysical events are morphologically similar to the GW signals in the three mass ranges, as detailed in Table 10.1. The Blip glitch is not shown here due to space constraints, but Figure 6.2(a) and Figure 10.2(A) each show an example of this type of glitch. (All glitches were obtained from LIGO-DV web [96]).



(a) (b) (c) (d)

Figure 10.3: Simulated GW signal with a total mass of 270 $M_\odot$ (a) along with a Low-frequency Blip glitch (c). Note the morphological similarity between the GW signal and the glitch. To better segregate the two samples, the Mercator projection is applied to the EHM signals, as shown in (b) and (d) for the GW and the glitch, respectively. Note the thinning of the Low-frequency Blip at lower frequencies, as opposed to the GW signal. (Adapted from [13]).

offsets within 0.1 s in the time-frequency visualizations of each GW and glitch so that samples are not always perfectly centered (see Figure 10.4). This also makes GSpyNetTree robust to small offsets in estimated candidate merger times, which is likely to happen in the GW search pipelines.

Following this approach, we built an augmented training set for each of GSpyNetTree's CNNs. First, we ensured a balanced representation of both GWs and glitches, as previous studies have shown that a higher rate of inaccuracies is related to poorly represented classes in Gravity Spy [13, 105]. Additionally, it is well known for CNN image classifiers that increasing the size of the training set improves classification performance [107]. Instead of having ~ 150 instances per class as in previous studies [13], we decided to enrich our training sets with more samples so that there were 1000 ± 300 of each per class. Moreover, we included the No Glitch class for the three classifiers to account for GWs with low signal-to-noise ratio (SNR). Table 10.1 shows the distribution of classes per classifier in the pre-O4 GSpyNetTree training sets.

We fetched all the glitches included in this training set from Gravity Spy classifications via LIGO-DV web [96] for both LIGO Hanford (LHO) and LIGO Livingston (LLO) observatories. Additionally, we manually verified all examples to discard misclassified or morphologically unconventional samples (we saved those for the validation study explained in Section 11.2). For the

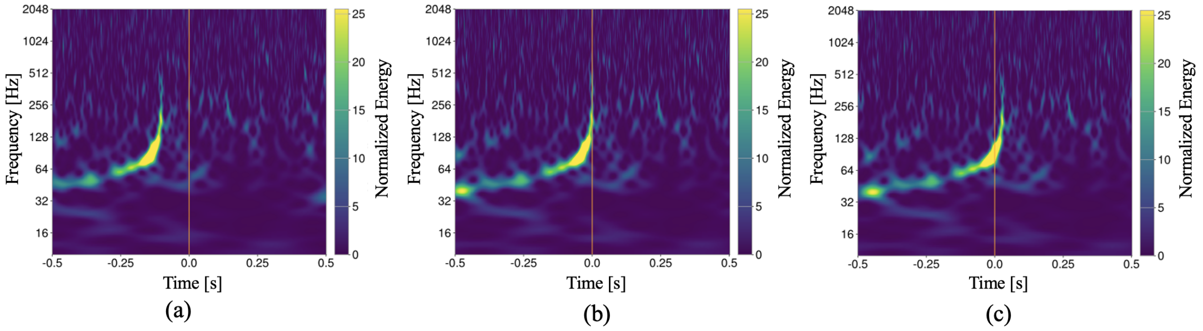


Figure 10.4: A simulated GW signal with merger time at $t = 0$ s (b), along with two copies of the signal, shifted to the left (a) and right (c). The time shifts are drawn from a uniform distribution in the range of $[-0.1, 0.1]$ s, and are included to be robust against GW candidates not centered at $t = 0$ s. Red vertical lines at $t = 0$ s are included as a visual aid. (Adapted from [13]).

GW simulations, we identified several segments of 64-second quiet detector data for both LHO and LLO during previous observing runs and injected simulated waveforms into them using the inspiral injection module of LALSuite [108], using the waveform model `IMRPhenomPv2` [109, 110]. GSpyNetTree's GW examples are uniformly drawn from a total merger mass range of 5 $M_\odot$ to 350 $M_\odot$, with individual masses ranging from 2 $M_\odot$ to 175 $M_\odot$, an SNR range of 8 to 35, and individual component spins ranging from 0.05 to 0.95. In this version of GSpyNetTree, the orientation and inclination of the interferometer with respect to the source is calculated to be optimal (that is, the GW is always coming perpendicular to the plane of the detector)[2].

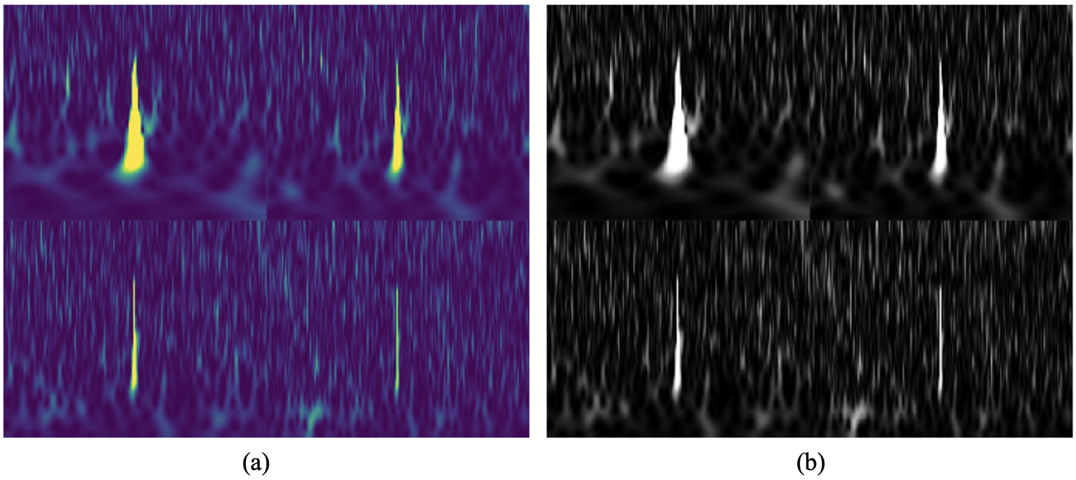


Figure 10.5: Samples generated by GSpyNetTree for a Blip glitch, in RGB (Red, Green, Blue) (a) and in black/white (b) color-schemes. The spectrograms of 0.5 (top-left), 1 (top-right), 2 (bottom-left) and 4 (bottom-right) seconds in duration are collated in a 2×2 matrix, to better capture the morphology of GWs and glitches. The samples used for GSpyNetTree are all generated in the black/white color-scheme (like Subfigure (b)) to remove the redundancy of the information codified in the three RGB channels. The colored sample in the left is included for visual reference.

We used the strain $h(t)$ timeseries from each glitch and simulated GWs to generate the time-

---

[2] This is changed in the O4-version of GSpyNetTree (see Chapter 12).

frequency features required by GSpyNetTree for each sample: four spectrograms 0.5 s, 1 s, 2 s, and 4 s in duration arranged in a 2 × 2 matrix, as in the original Gravity Spy architecture [3] (see Figure 10.5). The different spectrogram durations are used to better capture GW signals and glitches of different durations.

We fed these samples to GSpyNetTree, with architecture shown in Figure 10.6. After applying the time-offset augmentation, each sample is directed to one of GSpyNetTree's CNNs, based on its estimated total mass. If the sample's total mass is less than 50 $M_\odot$, it is directed to the LM classifier; otherwise, it is sent to the HM classifier. Events classified by the HM classifier are further directed to the EHM classifier when the following criteria are met: the total mass of the candidate is estimated to be greater than 250 $M_\odot$ and the HM classifier has classified the candidate as a Low-frequency Blip, No Glitch, or GW.

In the EHM classifier, the Mercator projection is applied to the sample before classification. Finally, each CNN returns an array of probabilities assigned to each class per sample. The pre-O4 version of GSpyNetTree (as the O4 era version) intakes GW candidate events uploaded to GraceDB [98] via the Data Quality Report [16] and classifies them as GWs or glitches with a reported probability.

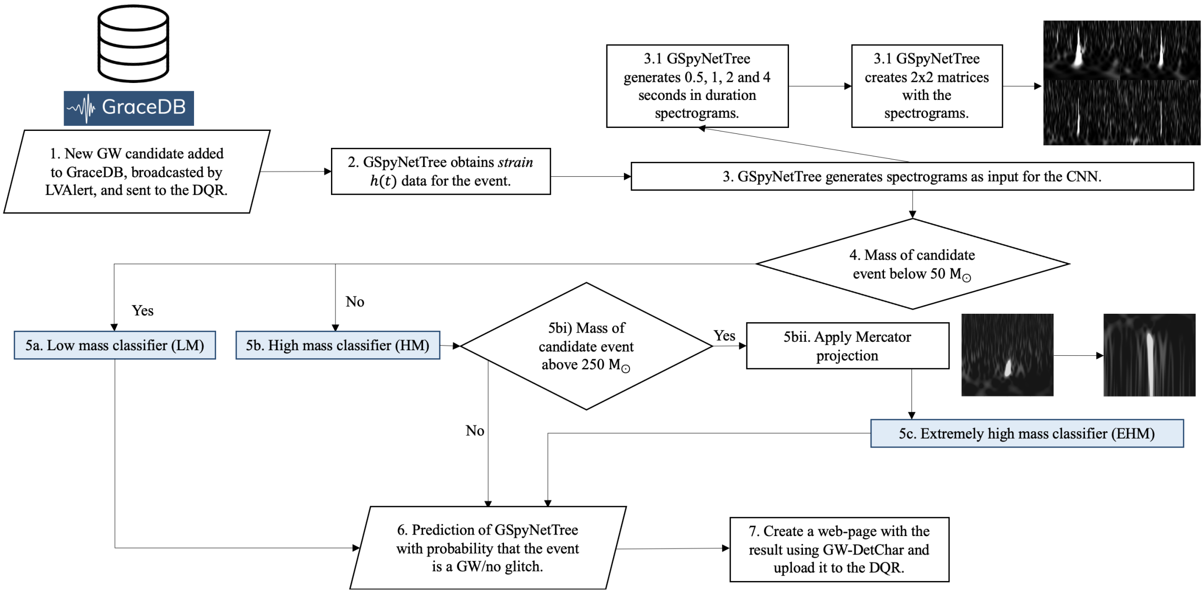


Figure 10.6: GSpyNetTree architecture: Triggered by a GraceDB superevent [98], timeseries (strain $h(t)$) data is fetched to generate spectrograms of 0.5, 1, 2, and 4 second durations. Time-frequency spectrogram visualizations are sent to the classifiers based on the estimated total candidate mass, and the Mercator transform is applied to the extremely high mass GW candidate visualizations. Each CNN outputs the probability that the input visualization contains a GW, an included class of glitch, or no glitch.

Once GSpyNetTree's training sets were complete, we trained the CNNs and evaluated their results. We used 80% of the dataset for training, and allocated the remaining 20% for testing the CNNs. Within the training set, we allocated 20% for validation. The results of the pre-O4 version of GSpyNetTree, as well as the validation studies carried out to evaluate its performance, are given in the next Chapter.

11 Pre-O4 GSpyNetTree: results

Having our augmented training sets ready for all classifiers, we first tested GSpyNetTree using the original Gravity Spy architecture [3].

The low-mass CNN had an overall accuracy of 94%, the high-mass CNN achieved 94.6%, while the extremely high-mass CNN made 96% accurate predictions. Additionally, all of them had a 92% accuracy for the GW class, with 7.2%, 4.5%, and 2.9% of GW signals misclassified as No Glitches in the LM, HM, and EHM classifiers, respectively. This was expected for low SNR signals, which may appear faint in spectrogram visualizations. Therefore, these misclassifications are not problematic for GSpyNetTree's purposes. Additionally, on average, the three classifiers accurately classified 94.2% of the glitches. While these are good results, it is not desirable to misclassify 8% of the astrophysical data per mass range, especially considering the high detection rate expected for O4 [74], and 6% of glitches (on average).

In order to further improve the accuracy of classifications, we used a new CNN architecture. It is well known that CNNs are the state-of-the-art method for complex image classification tasks [111]. Therefore, several networks specifically designed to tackle these problems have been studied and developed in the computer science realm. One of them is Inception V3: Google's state-of-the-art CNN [15]. It is made up of 42 layers, which makes it a very deep model compared to Gravity Spy's 5 layers (see Figure 8.6 in Section 8.4 for a comparison of the Gravity Spy and Inception V3 architectures). Additionally, it has shown better accuracy, less computational cost than other architectures, and a very low error (the percentage of erroneously classified samples) in various image classification tasks. Deeper neural networks require larger training sets to avoid overfitting; however, due to our drastically increased training set size over the one used in Jarov et al. [13], Inception V3 remained a viable option for our investigation.

We trained the three Inception V3 CNNs (LM, HM, and EHM) from scratch, as we had a large enough training set to do so, using the training data introduced in the previous Chapter. Figure 11.1 shows the improvement in classification with this new approach. Additionally, Appendix A shows the confusion matrices for the LM (first row), HM (second row), and EHM (third row) classifiers, using the Gravity Spy architecture (left column) and Inception V3 (right column).

Following the implementation of the Inception V3 CNNs, all classifiers reached more than 96% accuracy for GWs and all glitch classes, with 3.9%, 2.6%, and 2.3% of GWs misclassified as No Glitches, such that the GW (+ No Glitch) accuracy was 96% (+3.9%), 96% (+2.6%), and 97% (+2.3%) for the LM, HM, and EHM classifiers. This is a considerable decrease in the amount of misclassified astrophysical events. Also note that, on average, the glitch classification performance increased from 94.2% to 96.8%. The highest improvement was for the EHM classifier, from 92% to 96.3%, followed by the HM classifier, from 94% to 96.2%. The LM classifier only improved by $\sim 1\%$, but this is because its performance on glitches was already very good with the Gravity Spy architecture.

Due to the multi-class nature of the problem, the CNNs were trained using a categorical

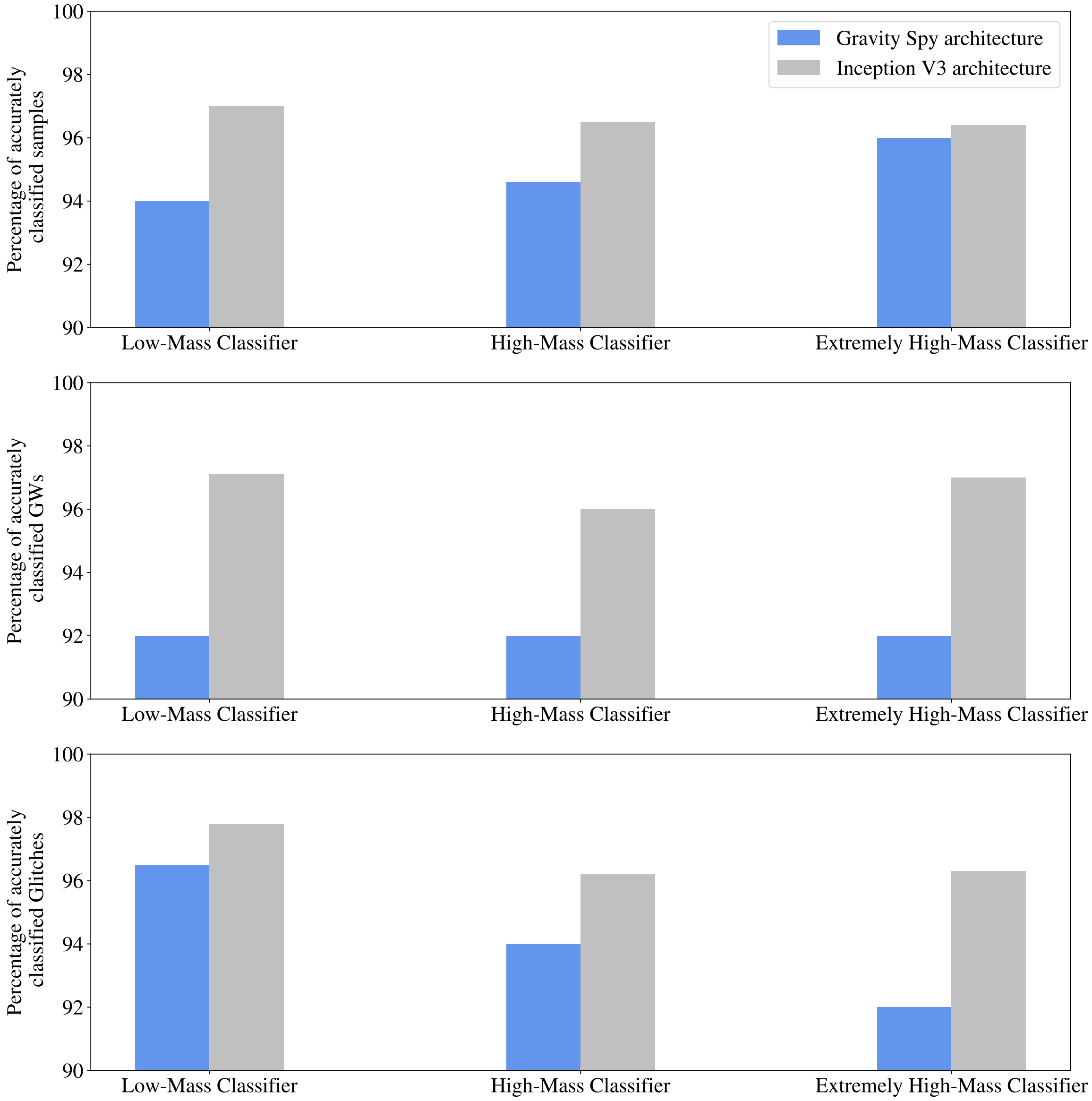


Figure 11.1: Accuracy results for the three GSpyNetTree CNNs trained using the Gravity Spy architecture (in blue) and the InceptionV3 architecture (in gray) for all GWs and glitches in the test set (upper panel), for only GWs (middle panel), and for only glitches (bottom panel). In the case of GWs, the classification accuracy improved from 92% in all three cases to 97% (LM), 96% (HM), and 97% (EHM). For glitches, the percentage of improvement from the Gravity Spy architecture to InceptionV3 was 1.3%, 2.2%, and 4.3% for the LM, HM, and EHM classifiers, respectively. Finally, overall accuracy also improved from 94% (LM), 95% (HM), and 96% (EHM) to 97% in all cases.

cross entropy loss function (see Subsection 8.2.1). To measure the performance of the classifiers, the accuracy and recall metrics were used. The latter was selected because it aims to minimize the number of false negatives in classification, which is optimal for the purpose of reducing the amount of misclassified gravitational-waves (i.e., those erroneously classified as glitches by GSpyNetTree).

Additionally, all classifiers were trained for several epochs. However, in order to avoid overfitting, several methods were implemented. First, the weights (see Chapter 7.1.6) of the best model were always saved. That is, if there was an improvement in the loss function from one epoch to the other, that model was saved and preferred over a previous one. Additionally, the model was set to early stop if there was not an improvement in the minimization of the loss function after certain number of epochs. After several experiments, this number was set to 10. Lastly, to tune the remaining hyperparameters of the model (see Subsection 8.2.3), several experiments were carried out with different optimizers and different learning rates. In particular, we experimented with the Adam Optimizer and the Stochastic Gradient Descent (SGD) optimizer. As expected, Adam had the worst results: for periods of consecutive epochs, the loss increased instead of decreasing. This is because Adam is well known to perform worse than SGD in image classification tasks [101]. For SGD, we tried using different learning rates, keeping the momentum constant (in 0.9). We ended up selecting an initial low learning rate ($\ell = 0.001$). Figure 11.2 shows the training plot of the EHM classifier using the Inception V3 architecture. Note that the loss function and validation loss function both tend to 0 (although the validation loss is above the training loss, as expected), and the accuracy and validation accuracy both tend to 1. For this classifier, the best models were obtained in epoch 15.

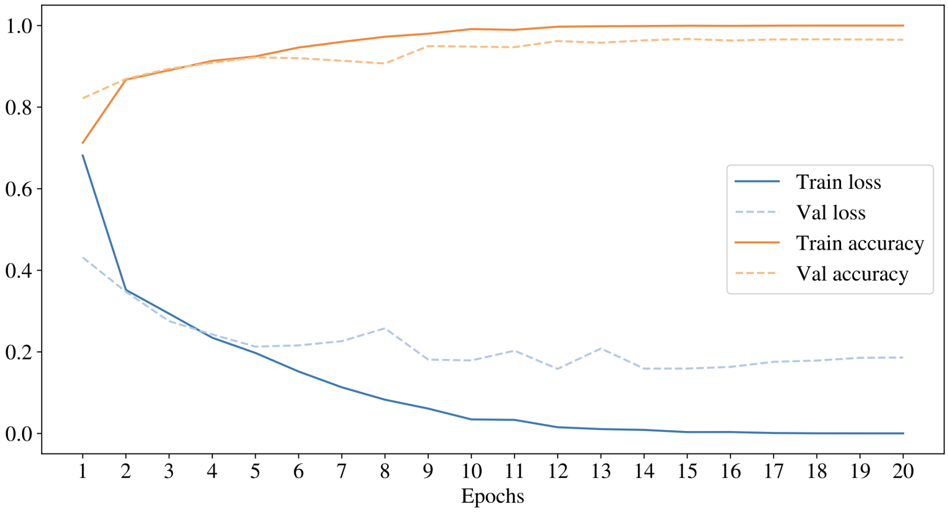


Figure 11.2: Training plot of the multi-class classifier trained with InceptionV3, with the training (solid line) and validation (dotted line) for the loss function (blue) and accuracy (orange), using an SGD optimizer with learning rate 0.001 and momentum 0.9. Note that the accuracy tends to increase, while the loss decreases. For the last epochs, a slight increase in validation loss can be seen: this is because the model has already achieved its optimum value. Indeed, the best model is found at epoch 15, where the validation loss is the lowest.

Assuring the selection of the best model with the most suitable combination of hyperparameters, and having improved the classification accuracy of the CNNs, we validated GSpyNetTree's readiness for O4 by performing three validation studies. We tested both the original Gravity Spy and the Inception V3 architecture, and the results were better with the latter for all validation studies (see [14]). The results for the InceptionV3 architecture are shown in Sections 11.1 through 11.3, based on [14].

11.1. Testing the reliance of GSpyNetTree on detector background noise

It was important to first evaluate the dependence of GSpyNetTree on detector background noise. This is important for O4 because the noise subtraction used to produce low-latency calibrated strain $h(t)$ is expected to differ from previous observing runs. O4 low-latency data is expected to use a non-linear subtraction of AC 60 Hz power artifacts (and resonances), similar to the technique used for publically released data from O3 [112]. It is also important to evaluate how transferable the GSpyNetTree model is to detect GWs for Virgo and KAGRA detectors, which have a different noise background from LIGO [14].

To test the reliance of the CNNs on background, we used 100 glitch examples per glitch class (see Table 10.1), using the non-linear 60 Hz subtracted strain channel, for each classifier. Figure 11.3 shows an example of a Blip glitch obtained from the original (left) and the clean (right) channels, respectively. Note that, to the naked eye, the differences in background are faint.

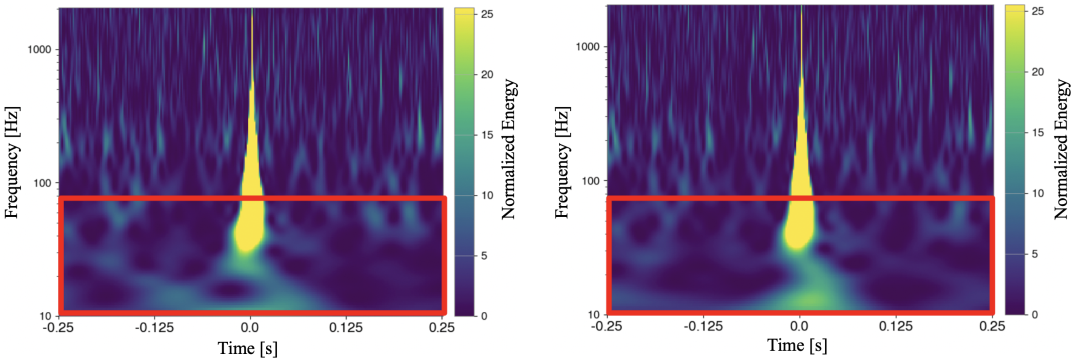


Figure 11.3: Spectrograms of an O3-era Blip from the Hanford detector, fetched from the original strain $h(t)$ channel (left) and the channel with a non-linear subtraction applied (right). A subtle difference in background can be appreciated around 60 Hz, as shown in the red boxes. The non-linear subtraction is implemented for low-latency data in O4. (Adapted from [14]).

Even though both spectrograms look quite similar to the naked eye, these small differences in background significantly impact CNN accuracy. In fact, when tested with non-linearly noise subtracted data, overall accuracy decreased to 76%, 75%, and 76% for the LM, HM, and EHM classifiers, respectively. Additionally, as shown in Figure 11.4, individual class accuracy also decreased for all of the glitches but one (Koi Fish). This likely occurs because these glitches usually cover a wide frequency and time range, so the background data applying the non-linear subtraction does not significantly impact the visualization [14].

Following this discovery, it is clear that to transfer GSpyNetTree to O4 with maximum accuracy, the CNN's dependence on background needed to be addressed with a new augmented training set. This way, the glitch dataset would be robust enough to account for changes in detector background noise. To do this, the training sets of the O4-era version of GSpyNetTree include not only data with and without the non-linear 60 Hz subtraction for LIGO Hanford and LIGO Livingston; but also Virgo glitches and simulated GWs injected in Virgo background noise [14]. This is detailed in Chapter 12.

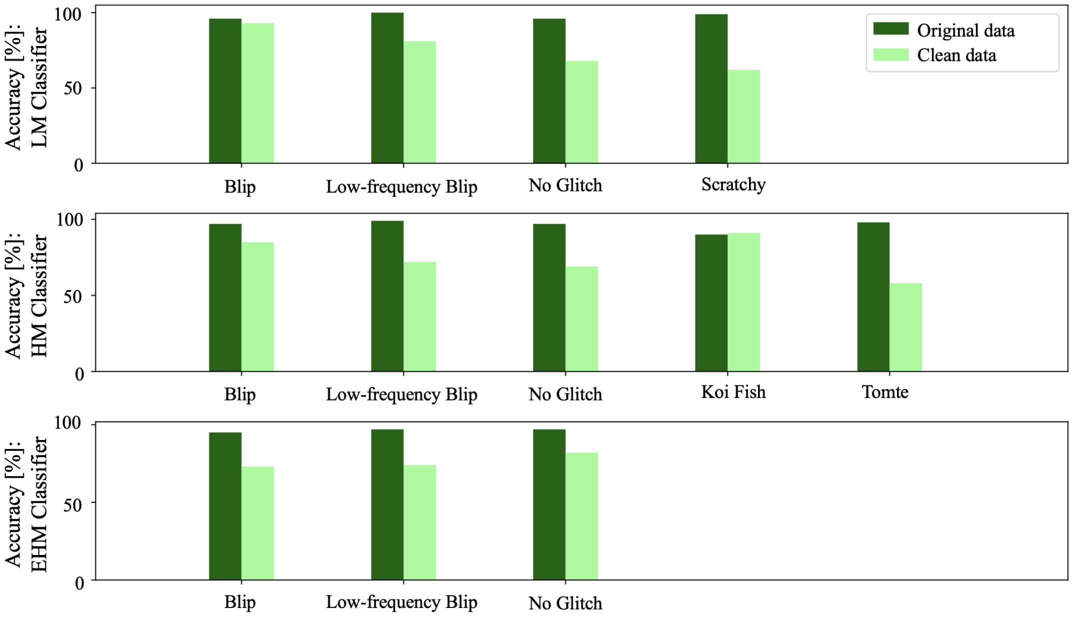


Figure 11.4: Accuracy results for the LM (top), HM (middle), and EHM (bottom) classifiers for each of the glitches in the validation set. Except for Koi Fish glitches, there is a significant decrease ($\sim 20\%$) in accuracy when the nonlinear subtraction to the data is applied. This likely happens as Koi Fish glitches are very loud and saturate the spectrogram, so the background calibrations do not affect the visualization considerably. (Adapted from [14]).

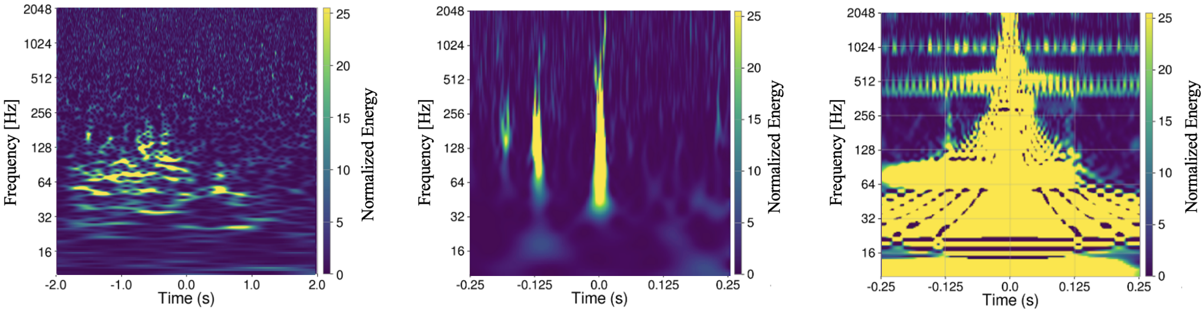
11.2. Testing the ability of GSpyNetTree to generalize on glitches not included in the original training set

To further validate the performance of GSpyNetTree toward O4, we studied the CNNs' performance on glitches not included in the original training set, which was restricted to avoid excess glitch classes that could reduce CNN performance. We selected 8 samples of Thunder glitches, all of which occurred during O3, and more than 100 samples of Extremely Loud, Repeating Blips, and Scattering glitches from previous observing runs. An example of each of the first three[1], in the duration that better captures their morphology, is shown in Figure 11.5. While the former type of glitch is not a Gravity Spy class, the three latter are part of the Gravity Spy training set [14].

Each type of glitch was carefully selected to address possible classification challenges that GSpyNetTree might face during O4, which started on May 24th, 2023. Scattering and Thunder glitches are fairly common in LIGO-Virgo data [12], so it is likely that they appear at the same time as a candidate GW. Additionally, Repeating Blips are an interesting case as Blips and Low-frequency Blips are already included in the training sets of the three classifiers. Studying the CNNs' performance in cases where these glitches repeat in the spectrograms allows a preliminary study on how the CNNs perform in cases where multiple glitch instances appear in the same visualization. Extremely Loud glitches are morphologically similar to Koi Fish glitches: they both extend in a wide frequency and time range. Including them permits the evaluation of the CNNs performance on morphologically similar glitches [14].

---

[1] An example of a Scattering glitch is shown in Figure 6.2(g).



(a) (b) (c)

Figure 11.5: Examples of three of the glitches used to test the ability of GSpyNetTree to generalize on samples not included in the original training sets: (a) Thunder glitch, (b) Repeating Blips, and (c) Extremely Loud glitch. The Thunder glitch is shown in a different time scale than the other glitches to highlight its morphology and duration. Note that, similar to a Koi Fish glitch, Extremely Loud glitches are very loud and saturate the spectrogram in (almost) the entire frequency range.

The classification results for the LM, HM, and EHM classifiers for the new glitches are shown in Figures 11.6a, 11.6c, and 11.6b, respectively. First, it is important to highlight that several glitches are often classified as GWs [14]. A lack of robustness to glitches not included in the training is problematic in O4, as new noise sources may appear with the increase in the detectors' sensitivity (see Section 4.5).

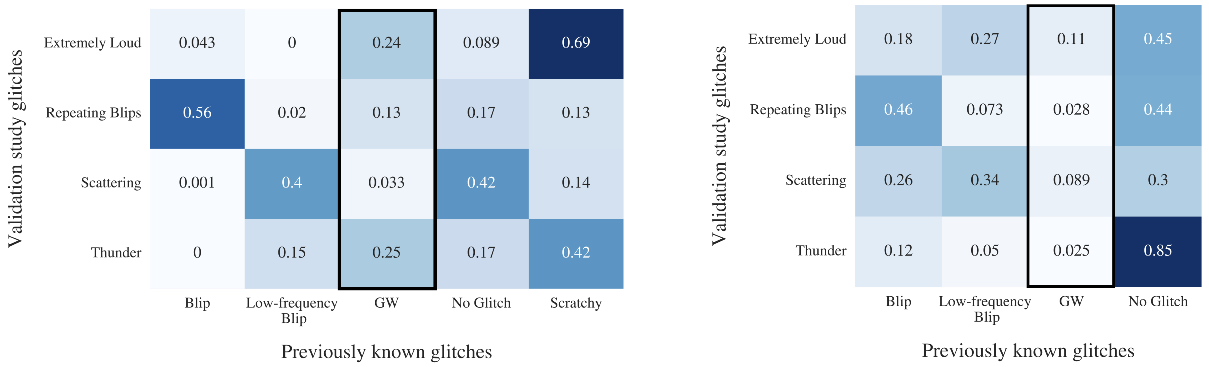
Overall, the CNNs have low confidence when classifying new glitches[2]. The classification probability is almost evenly distributed among all classes (including GWs) for most of the glitches in the three classifiers. This is, however, not the case for Repeating Blips and Scattering (in all three CNNs), and Extremely Loud glitches (in the HM classifier). Repeating Blips were classified as Blips with 56%, 52%, and 46% accuracy for the LM, HM, and EHM classifiers, respectively. Even though these results can be further improved, this is promising evidence that the CNN architecture can be tuned to better classify multiple instances of glitches in the same visualization [14].

Scattering glitches have significantly different behavior in the three classifiers. In the LM CNN, they are mistaken for No Glitches (42%) and Low-frequency Blips (40%), possibly because they evolve in a low-frequency range. However, the probability is distributed among the three non-GW classes in the EHM classifier. On the other hand, the HM CNN classifies most of the Scattering glitches (57%) as Koi Fish glitches, suggesting that it focuses on the broad (low)-frequency range the Scattering glitches cover. Since Scattering glitches are one of the most common glitches in current GW detectors [12] and having shown such different behavior among the three CNNs, this glitch class is included as part of the O4-era GSpyNetTree training set (see Chapter 12).

Moreover, morphologically similar glitches (particularly Koi Fish and Extremely Loud glitches, which both display saturated spectrograms) were classified with 99% accuracy as the class already known by the HM CNN, as shown in the red box in Figure 11.6c, and they are not easily mistaken for GWs. This is desirable as GSpyNetTree should minimize flagging non-astrophysical glitches as GW candidates. Even though not as morphologically similar as Koi Fish glitches, the

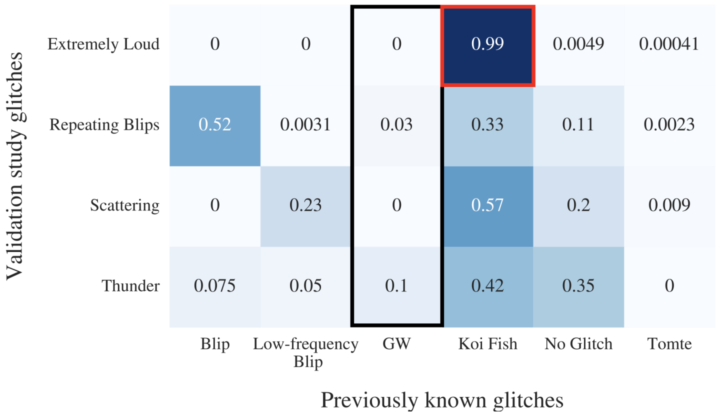
---

[2] This was expected, as supervised CNNs learn to predict what they have been exposed to during training.



(a) Confusion matrix for the LM classifier of GSpyNetTree for glitches not included in the original training set.

(b) Confusion matrix for the EHM classifier of GSpyNetTree for glitches not included in the original training set.



(c) Confusion matrix for the HM classifier of GSpyNetTree for glitches not included in the original training set.

Figure 11.6: Confusion matrices for the (a) LM, (b) EHM, and (c) HM classifiers for the test of glitches not included in the original training set: Scattering, Repeating Blips, Thunder, and Extremely Loud glitches. The x and y axes represent the predicted and true classes, respectively, and the confusion matrices are normalized by the total number of glitches of each class of the validation set. The black boxes highlight the fraction of glitches misclassified as GWs. The red box in the confusion matrix of the HM classifier highlights the fraction of Koi Fish glitches classified as Extremely Loud glitches. (Adapted from [14]).

LM CNN classified Extremely Loud glitches as the most dispersed in time and frequency glitch it knows: the Scratchy glitch. However, as opposed to the HM CNN, it misclassifies 24% of them as GWs, which is undesirable if the false positive rate is to be mitigated. On the other hand, the EHM classifier has a peculiar behavior when classifying both Thunder and Extremely Loud glitches. As shown in Figure 11.6b, 85% and 45% of the samples are misclassified, respectively, as No Glitches. Covering a wide time-frequency range with a high SNR, it is counter-intuitive that they are both classified in the No Glitch class [14].

In light of this behavior, an option to address misclassifications of new glitches is creating a new class for unknown or miscellaneous noise sources. However, this may reduce both the overall and GW classification accuracy, as glitches with substantially different morphologies would be included in the same class. Additionally, although the easiest solution is to include as many new classes in the CNNs as glitches arise, this approach will potentially reduce the performance of the CNN (as it has more classes it can get confused with) [14].

To solve this issue for the O4-era GSpyNetTree, each CNN could be used as a feature extractor in a semi/unsupervised learning task, similar to the approach followed by George et al. [113]. This way, each CNN's second-to-last layer[3] would be projected to a 2-dimensional space (or 3-dimensional space) using a clustering algorithm, such that glitches form clusters whose positions in the 2-d (or 3-d) space depend on their morphology. An example of this approach (for a 3-d space) is shown in Figure 11.7. Outliers (glitches not included in the original training set) would create a new cluster (i.e., a new glitch class) or could be separated enough to be segregated from training set classes for further Detector Characterization and glitch mitigation studies [14]. Compared to the other alternatives, this approach would successfully distinguish new glitch classes from GWs, maximizing the detection of astrophysical events with a low false alarm rate. This extension of GSpyNetTree for O4 is not developed in this thesis, and is suggested as future work.

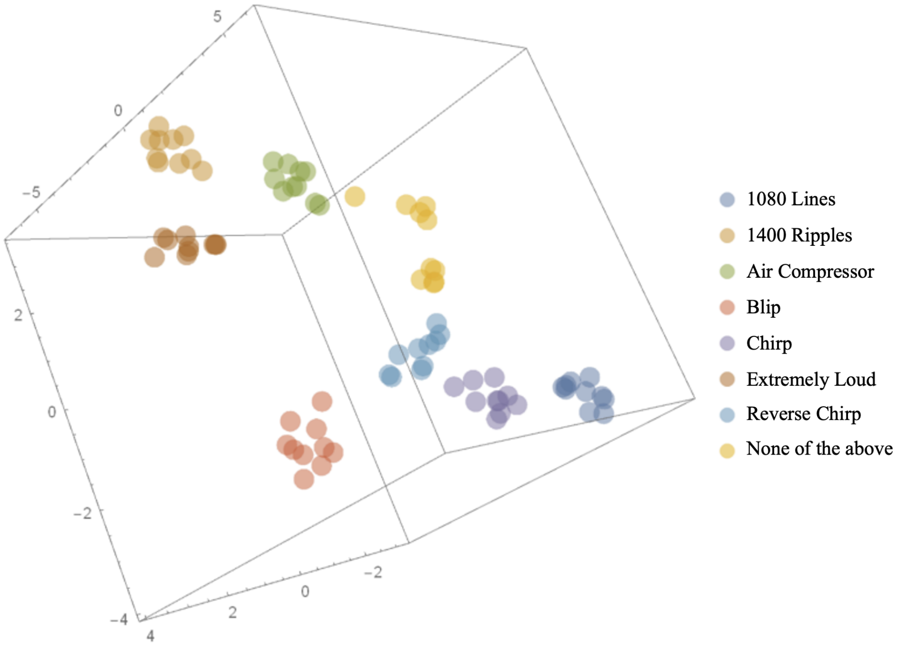


Figure 11.7: Location of different classes of glitches after applying a t-SNE (T-distributed Stochastic Neighborhood Embedding) [114] from an InceptionV3 CNN fine-tuned by George et al. [113]. The t-SNE algorithm reduces the dimensionality of the features of the CNN to a 3-d space, where different types of glitch form clusters. The relative separations of the clusters depend on the morphology of the glitches. (Adapted from [113]).

11.3. Testing the ability of GSpyNetTree to classify gravitational-waves in the presence of glitches

As the aLIGO and AdVirgo detectors become more sensitive and the rate of detected events increases, the probability of overlapping glitches and GW signals in strain $h(t)$ data also rises.

---

[3] The second-to-last layer has all the features learnt by the CNN, whereas the last layer only has the predicted output.

During the first [7] and second [8] parts of O3, it already happened with 26% and 23% of the candidates, respectively, and its therefore more likely to happen during O4. Thus, testing GSpyNetTree's performance in cases in which candidates occur in a similar time window as glitches is imperative to understand whether this hinders the classification of astrophysical signals [14].

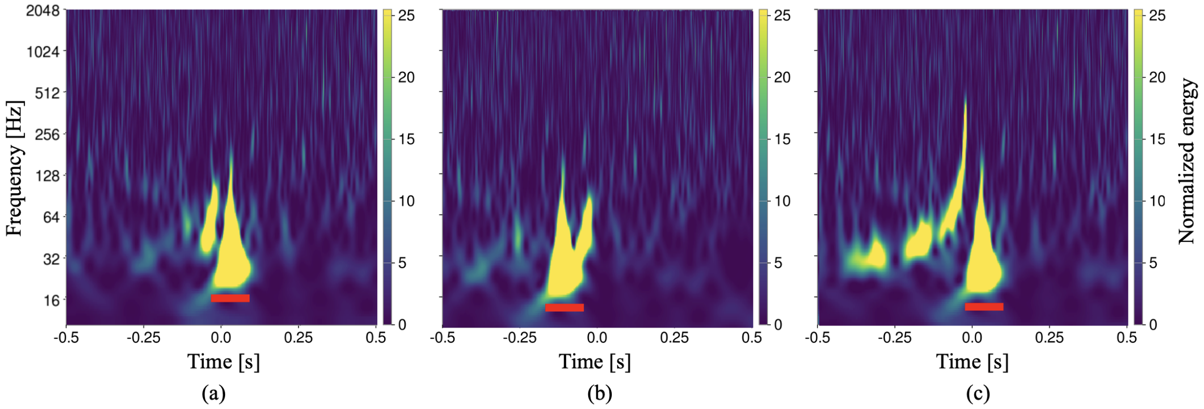


Figure 11.8: Samples of GWs injected in close proximity to a Tomte glitch (underlines in red). (a) and (b) show the same GW (with a total mass of 191.8 $M_\odot$), but the glitch has a different time-offset with respect to the GW: 0.011 s (a) and -0.130 s (b). Subfigure (c) shows the Tomte glitch with the same time offset as in (a), i.e., 0.011 s, but the GW signal has different astrophysical parameters (a total mass of 52 $M_\odot$). (Adapted from [14]).

For each classifier, 30 events per glitch class were selected. Then, 3 GWs with different astrophysical parameters (but in the same parameter space as in training) were simulated, and each of them was injected into one of the aforementioned glitches. After that, the glitch was shifted with respect to the GW, using an offset drawn from a normal distribution ($\mu = 0$, $\sigma = 0.25$ s) of time shifts. This allowed to test the performance of the CNNs in samples of GW signals (with different SNR, mass, and spin) with offset glitches occurring in the same time window of each spectrogram [14]. Figure 11.8 shows an example of a high-mass GW and a Tomte glitch [14], using different time offsets and astrophysical parameters. Similarly, Figure 11.9 shows samples of two different GWs injected into a Koi Fish (upper panel) and a Blip (lower panel) for all time durations used to generate GSpyNetTree's samples. Note that, depending on the offset, the glitch is visible in all spectrograms or just a few. In the first case, the glitch is shifted 0.47 s to the right, so it is only visible in the spectrograms of 2 and 4 seconds in duration. On the other hand, the blip is shifted 0.13 s to the right of the GW only, so it is visible in all time-frequency visualizations.

Figure 11.10 shows the results of the three classifiers for overlapping GW signals with each type of glitch (in a particular given mass range). More than 60% of the GW signals in the presence of glitches are misclassified as glitches (70% for the HM classifier). Whenever not flagged as GWs, the LM and HM CNNs classified the overlapping GW and glitch events as the longest duration or most saturated glitches they were exposed to during training: Scratchy glitches and Blips in the LM classifier, and Koi Fish and Tomte glitches in the HM classifier.

The EHM classifier tends to inaccurately classify overlapping GWs and glitches as Low-frequency blips, with almost the same probability (40%) as it classifies them as GWs (39%).

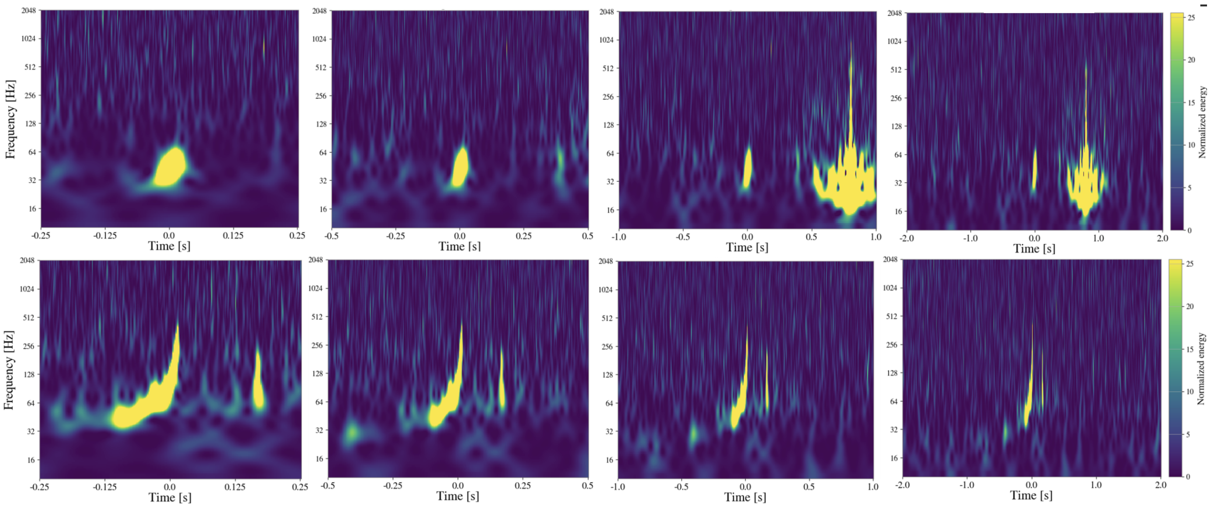


Figure 11.9: Samples of GWs injected in close proximity to a Koi Fish (top panel) and a Blip (bottom panel) in the four durations used to generate the training sets of GSpyNetTree. (*Top panel:*) The simulated GW has component masses $m_1 = 172.3\ M_\odot$ and $m_2 = 148.1\ M_\odot$ and is time shifted 0.47 s with respect to the Koi Fish glitch. (*Bottom panel:*) A GW with component masses $m_1 = 16.7\ M_\odot$ and $m_2 = 32.9\ M_\odot$ is injected 0.13 s to the left of a Blip. Note the differences in morphology between the two GWs due to their mass, and how the offset determines in which spectrograms the glitch will be visible.

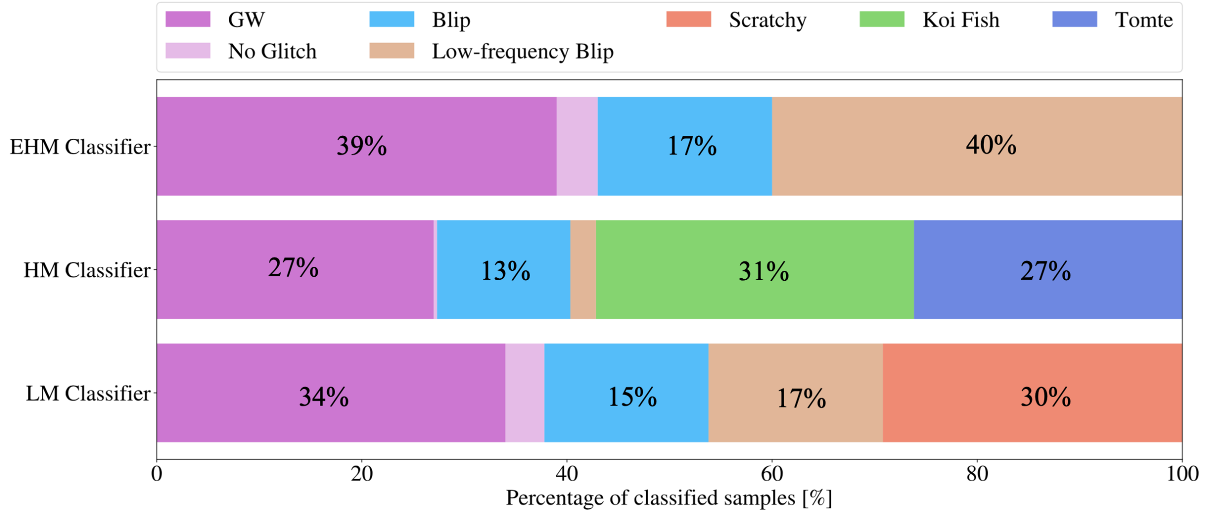


Figure 11.10: Accuracy results for the LM (top), HM (middle), and EHM (bottom) classifiers for the overlapping gravitational-waves and glitches. Percentages are shown for classes with more than 10% of the total number of instances. GWs are shown in orchid, No Glitch samples in pale orchid, Blips in light blue, Low-frequency Blips in light brown, Scratchy glitches in red, Koi Fishes in green, and Tomte glitches in dark blue. More than 55% (70% for the HM classifier) of the samples are misclassified as glitches.

In all three CNNs, having such a high fraction of candidates flagged as glitches may result in unnecessarily vetoed candidates, an issue that must be avoided for O4. This shows the importance of building an O4-era version of GSpyNetTree that is robust to these events, so it can be useful in fully-automated validation of LIGO-Virgo event candidates [14].

GSpyNetTree misclassifies most of the overlapping glitch and GW events because in its pre-

O4 era version it is a **multi-class** classifier: it outputs the probability that a given sample belongs to one and only one particular class, which is independent and mutually exclusive with the other classes. Chapter 12 describes GSpyNetTree's transition to a multi-label classifier, which would support the prediction of multiple mutually non-exclusive classes [100]. This way, the GW signal (or, equivalently, the glitch) is not misclassified, but both of them are flagged as samples of their respective labels. Figure 11.11 illustrates graphically the difference between a multi-class classifier (pre-O4 era version of GSpyNetTree) and a multi-label classifier (O4-era version of GSpyNetTree), explained in the next Chapter. An additional advantage of a multi-label classifier is that it is able to accurately classify instances of overlapping glitches (such as the Repeating Blips, see Figure 11.5), without including them as a separate class during training. In this way, multi-label GSpyNetTree would be able to accurately classify GWs in the presence of overlapping (or repeating) glitches. Section 14.5 shows the performance of the multi-label GSpyNetTree in this scenario.

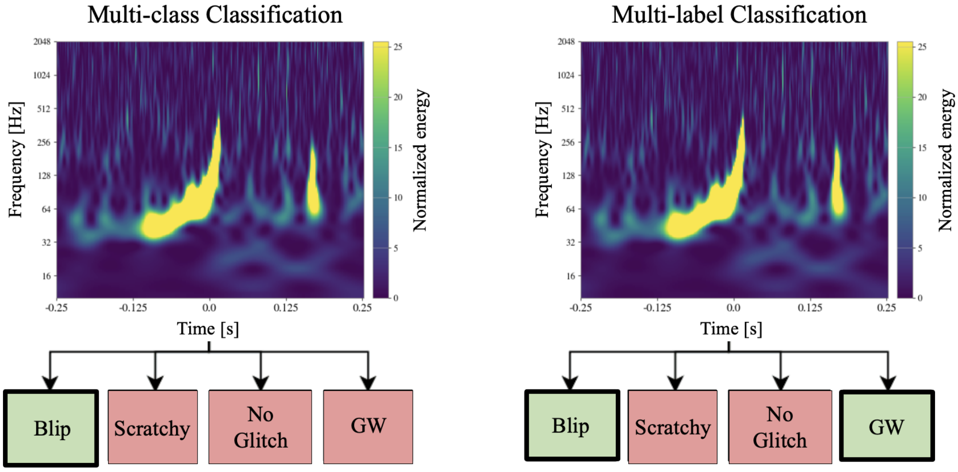


Figure 11.11: Differences between a multi-class (left) and a multi-label (right) GW signal-vs-glitch classifier. In the first case, the CNN is only able to identify one of the mutually exclusive classes it was trained on (making it prone to misclassify astrophysical data as glitches, or vice-versa). In this particular example, the classifier only detects the Blip glitch from a sample with an overlapping GW signal. The multi-label architecture, on the contrary, is able to predict several labels for a single sample. Both the GW signal and the Blip glitch are accurately classified. Chapter 12 explains the implementation of the multi-label version of GSpyNetTree.

Having explored likely O4 scenarios in which the pre-O4 multi-class version of GSpyNetTree fails, we proceeded to build a new-O4 version of our tool; based on the previous findings, and the recommendations of Jarov et al. [13] and Alvarez-Lopez et al. [14]. The next Chapter describes the O4-version of GSpyNetTree, which is currently deployed in the Data Quality Report [17] as part of the LIGO-Virgo event validation pipeline.

12 O4-era GSpyNetTree: a decision tree of multi-label signal-vs-glitch classifiers

Building up on the recommendations given by Jarov et al. [13] and Alvarez-Lopez et al. [14], we built the O4-era version of GSpyNetTree: a decision tree of multi-label GW signal-vs-glitch classifiers. Besides augmenting the training sets with Virgo glitches and with data extracted from a broad array of background noise, and adapting GSpyNetTree to a multi-label architecture, we also implemented several other changes to build a more robust training set. This Chapter outlines all the changes implemented to the pre-O4 era version of GSpyNetTree. Divided in several subsections, the generation of GW signals is tackled first, followed by the inclusion of new glitch classes. Then, the changes in the construction of the training sets are detailed; and, at the end, the transition to a multi-label classifier is addressed.

12.1. Gravitational-wave signal simulation: improvements with respect to pre-O4 GSpyNetTree

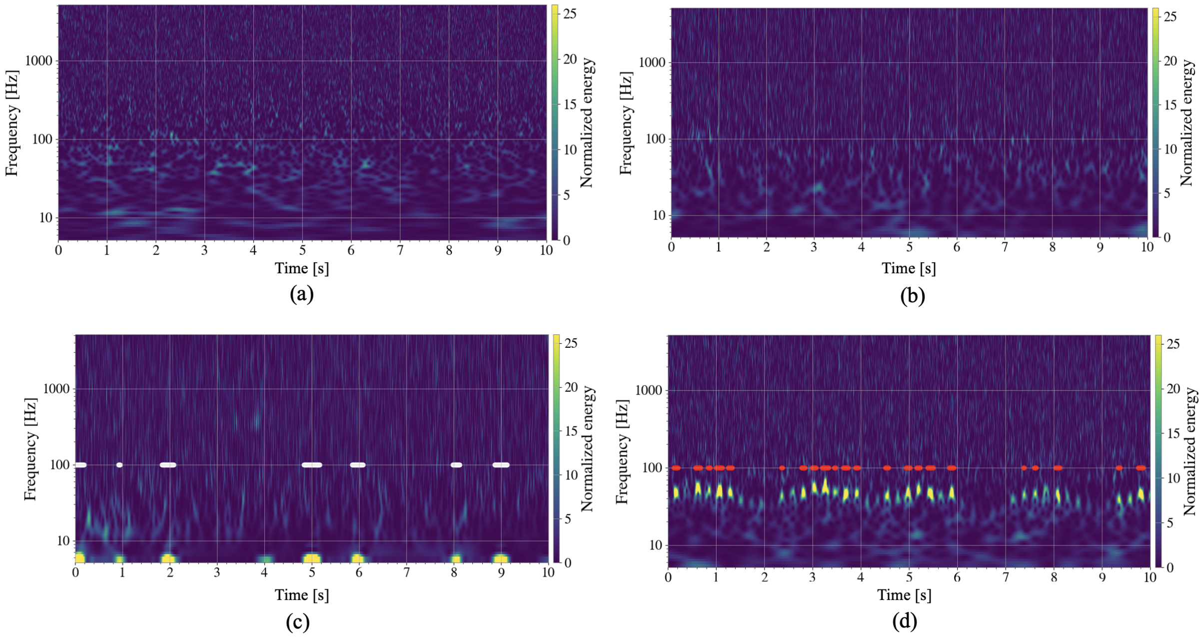
This section details the changes we implemented to GSpyNetTree regarding gravitational-wave signal simulation, making it a more robust tool with respect to its pre-O4 predecessor.

12.1.1. Making GSpyNetTree robust to a broad array of background noise

The first thing we did was to increase the automation of GSpyNetTree. Being a tool used within the LIGO-Virgo collaboration, it is important to build on the computational aspects of the project. The generation of the GW signals and their injection into background data was taking approximately 1 day, so we started to use HTCondor [115] to parallelize our computationally intensive task. This reduced the computational time of GW signal generation to around two hours.

Additionally, as described in the previous Chapter, one of the recommendations for the O4-era version of GSpyNetTree was to make it more robust to background noise. The background noise data that the pre-O4 version of GSpyNetTree used was drawn from clean detector times that occurred in a single day during O2 in the LIGO Hanford and LIGO Livingston observatories. However, the data of a single day is not representative of the noise that LIGO experiences during an entire observing run. In order to have a broader array of noise to inject the GW signals, we added to our already selected times, clean detector data from the first [7] and second [8] parts of O3, for all Hanford, Livingston, and Virgo detectors.

To do so, we studied random 10-second long time intervals of O3. As the longest spectrogram used in GSpyNetTree is 4 seconds in duration, these 10 second intervals give us 3-second windows of clean detector data before and after our longest spectrogram time interval. For these time-windows we programmatically generated spectrograms via Q-transforms for the specified time interval with a Q-value of 20 (the selected Q-value selection is justified later on). Having the data ready, we calculated the energy of each bin (or tile) of the Q-transform using a convolution over each bin. We then fitted a linear regression to the data to calculate the number of bins above the energy threshold, similar to the method employed by the Glitch Find DQR task (see Section 7.1.6). The non-clean bins were marked with white for noise below 10 Hz, as shown in Figure 12.1(c) for a time window of discarded noisy Virgo data, and with red for noise above 10 Hz (see Figure 12.1(d) for a discarded Livingston time - note that it actually corresponds to a Fast Scattering glitch). In this process, we discovered that Virgo is, in general, very noisy at lower frequencies (i.e., below 10 Hz). While this is not a limit for current GW detectors (as noise sources like seismic noise impede GW detection as such frequencies, see Chapter 5), it could make GSpyNetTree more robust to typical Virgo noise. For the O4-era version of GSpyNetTree presented in this thesis, we decided not to include such times, to assure clean detector data for GW signal injection.



(a)
(b)
(c)
(d)

Figure 12.1: The top panel shows spectrograms of two clean detector background times, (a) for LIGO Hanford and (b) for LIGO Livingston. as identified by our algorithm, which are used to inject gravitational-wave signals for GSpyNetTree's training sets. The bottom panel shows two samples of noisy detector data: (c) shows a 10-second interval of Virgo with abundant blops of noise below 10 Hz flagged in white. (d) shows a spectrogram of LIGO Livingston marked as noisy. This spectrogram shows a Fast Scattering glitch, and the time of each blip is shown in red. Both (c) and (d) times were discarded as clean backgrounds to inject gravitational-wave signals for GSpyNetTree.

We defined the normalized energy threshold to be 24, and the maximum number of bins for a time interval to be considered not clean, 4. If four or more bins (or tiles) of the Q-transform had a normalized energy above 24, the given time interval was discarded. We ran this process until we had approximately 1000 clean background times for each detector, like the ones shown in Figure 12.1(a) and Figure 12.1(b) for LIGO Hanford and LIGO Livingston, respectively. At the end, we obtained 984, 980, and 958 different clean detector data segments for LIGO Hanford, LIGO Livingston, and Virgo, respectively. Since a longer period of time (the entire duration of

O3) was considered for this background data selection, a better representation of clean detector times, compared to the pre-O4 version of GSpyNetTree, was assured.

Besides improving the method of finding clean detector background times to inject GW signals for all three GW detectors (Hanford, Livingston, and Virgo) in their respective strain $h(t)$ channels, we also generated data for the selected times applying the non-linear subtraction of 60 Hz (and resonances) power artifacts for both LIGO detectors[1] This is one of the calibrations used for low-latency strain $h(t)$ data in O4. All these techniques assure that GSpyNetTree is robust to a broad array of GW signals, independent of the detector background noise in which they are embedded.

12.1.2. Randomizing gravitational-wave signal simulation parameters

For the first version of GSpyNetTree, the longitude and latitude (equivalently, right ascension and declination) of the GW signal were calculated to be optimal with respect to the detector's orientation. However, in reality, GWs do not always come optimally oriented with respect to the detector (i.e., perpendicular to the detector plane, see Section 4.1.1). In fact, the response of the interferometer changes depending on the orientation of the GW (recall the antenna pattern of Figure 4.3). To better capture any possible location and inclination of the astrophysical sources that generated the GW with respect to the location of the interferometer on Earth, we randomized the latitude, longitude, and inclination parameters of the simulations. While this change makes GSpyNetTree more robust to GW candidates in O4, it does not have a considerable impact to the naked eye on spectrogram visualizations. The mass, SNR, and spin of the signals were already drawn from uniform (the two former) and normal (the latter) distributions; so drawing location and inclination information from a uniform distribution was the last step to randomize all relevant GW astrophysical parameters in signal simulation (within current detector limits). Note that, as well as in the pre-O4 version of GSpyNetTree, the signals were simulated using the LALSuite [108] and the `IMRPhenomPv2` [109, 110] waveforms.

12.2. Glitch selection

Once we had ready our simulated gravitational-wave signals, we focused on selecting glitches relevant for the O4-era version of GSpyNetTree, and optimizing the generation of their timeseries. We noticed that, similar to what happened with the injections, generating the timeseries for glitches was a bottle-neck of the training set generation process, so we decided to optimize it using HTCondor [115] as well. Again, parallelizing the tasks reduced the process to a couple of hours from an entire day.

Compared to its predecessor, and based on Section 11.1 [14], the O4-version of GSpyNetTree includes new Virgo glitches. These were all obtained from LIGO-dv [96] and manually verified. Including Virgo glitches makes GSpyNetTree more robust to background noise, as desired.

Based on Davis et al. [12] and the recommendations of Alvarez-Lopez et al. [14], we decided to include two additional glitch classes, Fast Scattering and Scattering, as these were the most common glitches in LIGO-Virgo data during O3, and they are likely to occur during O4. While

---

[1] Note that Virgo has different instrumentation and does not apply this calibration yet.

not morphologically similar to all kinds of GWs (although low-mass GWs might share some similarity with Scattering, and high-mass signals may be easily confused with a single Fast Scattering energy blop), it is important to include them due to their recurrent appearance in the data. This makes GSpyNetTree robust to detect GW signals in the presence of the most recurrent glitches. While not included in the original O4 training set, after exploring the performance of the O4-era version of GSpyNetTree in other types of glitches, we decided to include Low-frequency lines (a type of glitch included in Gravity Spy [3]) to make GSpyNetTree robust against noise in the low-frequency region. An example of this type of glitch is shown in Figure 12.2. Including noisy Virgo times below 10 Hz could also help address cases with very low-frequency noise in a future version of GSpyNetTree[2]. As manually verifying these samples is time consuming, only around 300 samples of Low-frequency Lines were included. Augmenting this training class is also a necessary future addendum to the current O4-version of GSpyNetTree.

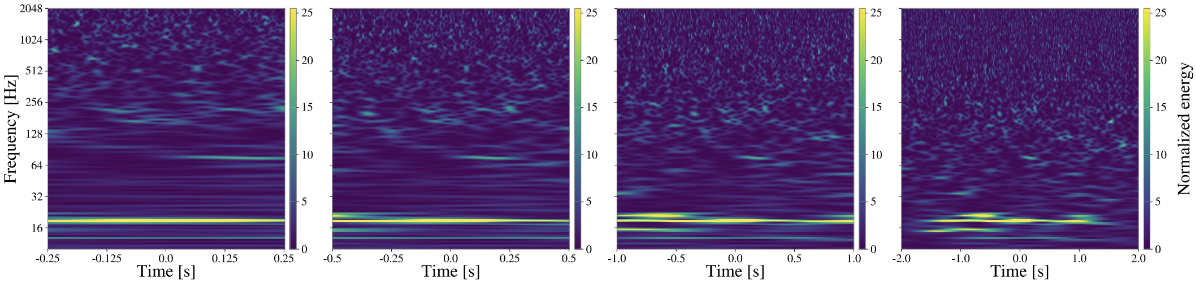


Figure 12.2: Example of the Low-Frequency Lines glitch in all four spectrograms (0.5, 1, 2, and 4 seconds in duration) used for GSpyNetTree samples. Note that this is the only type of glitch in GSpyNetTree considering noise at low frequencies.

Table 12.1 shows the distribution of glitches (including the No Glitch class) in the O4 version of GSpyNetTree. Note the importance of including more Low-frequency Lines samples in a future version of GSpyNetTree. Table 12.1 only shows the number of original samples, without considering the data augmentation techniques we use. First, it is important to highlight that LIGO Hanford and LIGO Livingston glitches are generated in two different ways (just as for clean detector times): using the usual, low-latency channel, and applying the non-linear 60Hz subtraction to the data. This way, the number of glitches (and clean detector times) for both LIGO detectors is doubled. Additionally, similar to the previous version of GSpyNetTree, four time offsets drawn from a uniform distribution between -0.5 and 0.5 seconds are used to shift the signal, so the merger time is not always centered in $t = 0$ s. We extended the range of the time offsets from $[-0.1, 0.1]$ s, to be more robust against merger times not centered at 0 by the GW search pipelines (see Section 4.6). While we aimed to have a balanced number of glitches between all detectors, there are a few types of glitches (e.g., Fast Scattering, Low-frequency Blips, and Scratchy) for which there are either no Virgo glitches (the two former) or they are very scarce (the latter). In those cases, more LIGO glitches were obtained.

---

[2] This is not implemented in this thesis, but considered as future work.

Type of Sample	**Number of Samples per Detector**		
	LIGO Hanford	**LIGO Livingston**	**Virgo**
No Glitch	984	980	958
Blip	511	488	546
Low-frequency Blip	406	594	0
Fast Scattering	459	602	0
Koi Fish	359	499	331
Light Scattering	354	217	272
Scratchy	467	533	67
Tomte	498	492	427
Low-frequency Lines[3]	90	80	140

Table 12.1: Distribution of glitches and No Glitch sample per detector for the O4-era version of GSpyNetTree. Note that the number of LIGO Hanford and LIGO Livingston samples is doubled due to the usage of the channel applying the non-linear 60 Hz power artifact subtraction, in addition to obtaining data from the original strain $h(t)$ channel. Additionally, four time-offsets, drawn from a uniform distribution $[-0.5, 0.5]$ s, are used to generate four additional samples of each glitch/No Glitch in order to augment the training set and make GSpyNetTree robust to candidates with merger time shifted from $t = 0$ s. Note that there are some types of glitches for which there are 0 or very few Virgo glitch samples. This is because either Virgo does not have these types of glitches, or they are very rare, respectively.

12.3. Generating samples of gravitational-waves in close proximity with glitches

One of the recommendations of Alvarez-Lopez et al. [14] described in Section 11.3 is the inclusion of overlapping GW signals and glitches, as this is very likely to occur during O4. In fact, it already happened with more than 25% of GW events during O3 [7, 8].

Before generating the samples, we noticed that completely overlapping glitches and GW signals (especially for very loud glitches, like Koi Fish) saturated the spectrogram visualizations in a way that the GW signal and the glitch were no longer distinguishable, as shown in Figure 12.3 for a 4-second long spectrogram.

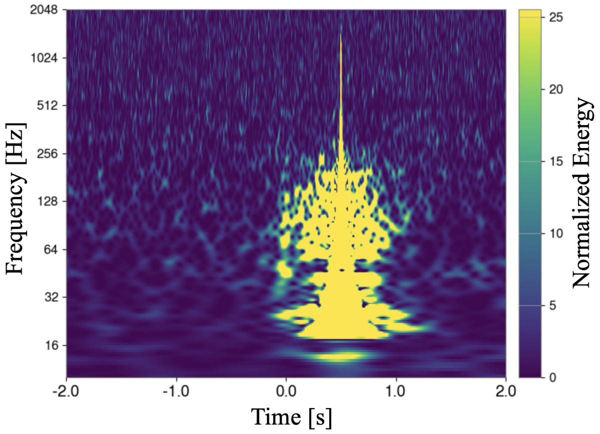


Figure 12.3: Example of a very saturated spectrogram resulting from overlapping a GW signal with a Koi Fish in very close proximity. The excess power in the time-frequency visualization results from excess energy in the tiles of the Q-transform.

To tackle this problem, we implemented two different solutions to avoid excess power in spectrograms with overlapping samples. The first thing we did was to set a standardized Q-value of 20, instead of considering the (4, 64) Q-range that was used in the pre-O4 version of GSpyNetTree. Having a lower Q-value restricts the amount of energy per tile in the Q-transform, such that excess power is less likely to saturate the spectrogram visualization. Additionally, a standardized, and relatively low Q-value also improves the visualization of low-mass GW signals, potentially increasing the number of accurately classified samples. However, it is important to experiment with other Q-values to find the one that optimizes GW signal-vs-glitch classification in the entire mass range considered by GSpyNetTree (i.e., between 5 $M_\odot$ and 350 $M_\odot$). This is not done in the current thesis but is proposed as future work for this tool.

To further avoid excess power in overlapping GW signal and glitch events, we devised a new method to inject a gravitational-wave event into a glitch. First of all, we noted that the glitches used for GSpyNetTree can be further segregated by their duration: on one category, transient glitches (shorter in duration) like Blips, Low-frequency Blips, Koi Fish, and Tomte. On the other, glitches extending in a broader time range: Scratchy, Light Scattering, Fast Scattering, and Low-frequency Lines. Therefore, random time-offsets for all types of glitch could result in a spectrogram with excess power. To generate offsets consistent with the duration of each glitch, we followed these steps:

Type of Glitch	**Average duration [s]**
Blip	0.138
Low-frequency Blip	0.137
Tomte	0.346
Koi Fish	0.899
Fast Scattering	1.657
Scratchy	1.750
Light Scattering	2.073

Table 12.2: Average duration of each glitch considered in the training sets of GSyNetTree. Note that Blips, Low-frequency Blips are shorter in nature than Koi Fish, Fast Scattering, Scratchy, and Light Scattering glitches.

1. **Calculate the average duration of each glitch:** Using the method described in Section 12.1.1, we flagged the glitchy times in a spectrogram (with either red or white, depending on if the glitch is above or below 10 Hz, respectively). Having the initial and final times of the glitch, calculate their duration. Repeat this process for 30 different samples of each glitch class, and average the results to get the average duration per glitch class. Due to the central limit theorem, the mean of the durations of the samples is representative of the duration of the entire population of glitches. Table 12.2 shows the average durations calculated for each type of glitch[4].

2. **Obtain the duration of the GW event** that is going to be injected in proximity to the

---

[4] Low-frequency Lines are not included here because, for the current version of GSpyNetTree, no overlapping samples of this glitch and GW signals were generated. Such samples will be generated when more Low-frequency Lines instances are manually verified and included in the training sets of GSpyNetTree, in a future version of the tool.

glitch.

3. **Calculate the time intervals in which the glitch can be injected without generating excess power in the spectrogram visualization.** The process varies depending on the duration of the glitch.

    - If the glitch is a short noise transient, it could be injected at any time between -2 s (the left limit of the longest spectrogram in duration considered by GSpyNetTree) and $-(\text{duration}_{\text{GW}} - 0.5 \cdot \text{duration}_{\text{glitch}})$ seconds, as shown in the top panel of Figure 12.4. To the right, the glitch can be injected anytime from $(0.5 \cdot \text{duration}_{\text{glitch}})$ seconds to 2 s, as shown in the bottom pannel of Figure 12.4. A uniform distribution is generated for these time intervals and, from each of them, one offset is drawn randomly. This way, for each simulated GW, one glitch will be injected to the left and, another one, to the right.
    - If the glitch is a longer transient, a different process is applied. Note that Scattering and Fast Scattering occur at rather low frequencies. Also, the Scratchy glitch has a very low SNR. These features suggest that, if occurring simultaneously with a GW, it is very unlikely that the spectrogram will be over-saturated. For these cases, we consider three injections per glitch: one before the merger time of the GW, one after the merger time of the GW (without the restrictions that apply for the shorter transients in either case), and one at merger time (i.e., $t = 0$ s). This last sample is included as Fast Scattering and Scattering are some of the most common glitches in LIGO-Virgo data [12], and injecting them at the same time of merger time is not likely to affect the time-frequency visualization[5].

4. Generate 1800 samples of GWs occurring in the proximity of a glitch for each of the GSpyNetTree classifiers using the appropriate glitches. This is done using the original low-latency strain $h(t)$ channel, assuring a balanced distribution of glitch samples in all detectors (i.e., ~ 600 per interferometer).

5. Generate another 1800 samples, but this time using the channel that applies the non-linear 60 Hz power artifacts subtraction, to increase robustness to a broader array of background noise. A balanced distribution of glitches in all interferometers is assured as well.

6. In the same way it is done for samples with only glitches and only GW signals, generate 4 offsets drawn from a uniform distribution in the $[-0.5, 0.5]$ s time range. This is done to increase robustness against candidate times with mergers not centered at $t = 0$ s.

When generating of these samples, we tried to maintain a balanced training set, considering the individual GWs and glitches we had already generated. Also note that each sample of the overlapping GWs and glitch instances was tagged with two labels: GW and the corresponding glitch category. As two labels may be predicted by the CNN, considerable architectural changes were applied to GSpyNetTree, as detailed in the following section.

---

[5] This same process will be applied to Low-frequency Lines when they are included in the tool.

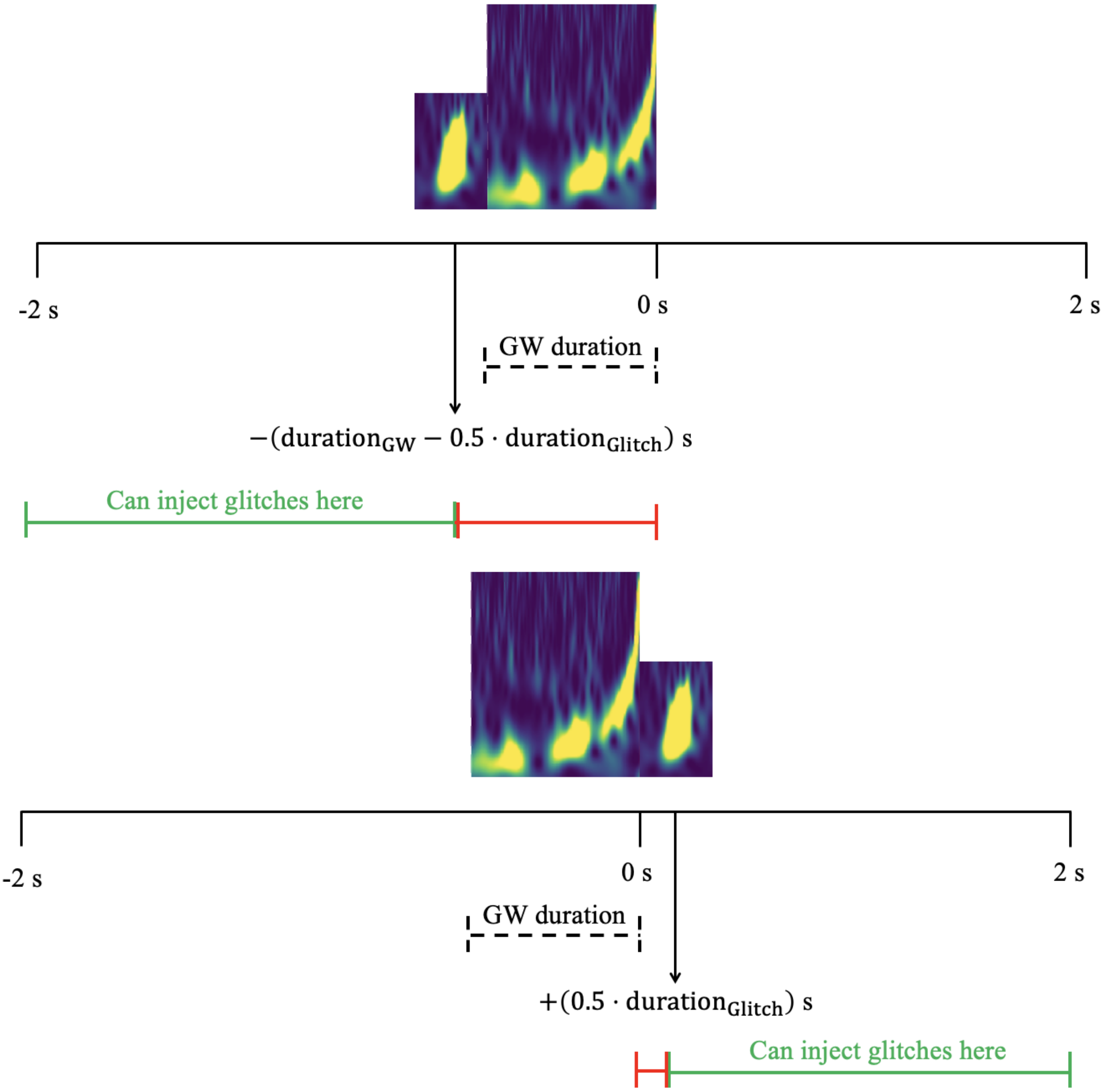


Figure 12.4: The Figure shows a schematic representation of the injection of a GW in the proximity of a Blip glitch before (top panel) and after (bottom panel) the merger time of the signal. The duration of the GW and the closest initial time at which the glitch can b injected injected are indicated in both cases. The suitable injection times are shown in green and the invalid ones, in red.

12.4. Building a multi-label architecture for GSpyNetTree

Due to the increased number of samples that the O4-version of GSpyNetTree is trained on, the first thing we had to do was to build a TensorFlow-based data generator. This was done because we noticed that our training set exceeded the RAM memory of the LIGO Livingston computing cluster we were using to train our CNNs. This way, instead of reading more than 50.000 at once, the training generator loads batches of 32 instances each time. The batch size of 32 was chosen based on various recommendations to keep this value low in Deep Neural Networks [102].

Having the training generator ready, it was important to change the CNN architecture we were using for the pre-O4 version of GSpyNetTree. There were four main changes that we performed: changing the loss function, adapting the activation function of the output layer of the CNN, using a new metric to evaluate the performance of GSpyNetTree, and changing the way in which results were interpreted.

Regarding the loss function, as described in Section 8.2.1, multi-label classifiers need a binary cross entropy loss function, instead of the categorical cross entropy that is used in multi-class classifiers. This is because a multi-label problem can be divided into several binary tasks, each of which is evaluated using this loss function. At the end, all binary results are added to obtain the loss of each epoch.

The final layer of the CNN was also changed to use a sigmoid activation function instead of a softmax activation function. This way, each neuron in the last layer has a weight ranging from 0 to 1; and a label is said to be predicted by that particular neuron if the resulting weight is above 0.5 (recall the sigmoid function of Figure 8.3). Note that, different from the multi-class classifier in which only one neuron will have the largest probability of them all (as the weights of all neurons in the last layer add up to 1), in the multi-label classifier, more than one neuron (even all of them or none of them) can have a probability higher than 50%. This way, glitches can be predicted accurately in the proximity of GWs whenever this situation happens. With the possibility of predicting either 0, 1, or more than 1 label, we had to change the way in which we interpreted GSpyNetTree's results. To do so, we noted that a single confusion matrix (like the ones in the multi-class pre-O4 version of GSpyNetTree) could not be generated for the multi-label case. Instead, one **binary** confusion matrix is generated per label. If there are N labels, each of these binary confusion matrices will have two possible outcomes per sample: predicted as the $i-$th class, and NOT predicted as the $i-$th class (or, equivalently, predicted as anything else but i), with $i \in N$.

The last change we implemented for the multi-label version of GSpyNetTree has to do with the metric monitored during training. In the pre-O4 version of GSpyNetTree, we used accuracy as the main metric. However, accuracy is not substantially useful in multi-label problems. This is because if a sample has two real labels (say, GW and Blip), and only GW is predicted, the entire sample will be marked as an incorrect classification, while the only affected class should be Blip. To avoid this issue, which underestimates the performance of the CNN, we use recall as the main metric to monitor in our CNNs. We chose this metric as its main purpose is to minimize the number of false positive samples. That is, GWs incorrectly predicted as glitches, or vice-versa. For completeness, we also include the precision metric, but we do not use it to track the performance of GSpyNetTree.

All the other aspects of the pre-O4 version of GSpyNetTree were preserved; namely, the optimizer (Stochastic Gradient Descent) with the same learning rate and momentum, and the usage of early stopping with no improvement after 15 epochs. The following section details the architecture of GSpyNetTree in its O4-era version, as well as the documentation written for the task for the DQR.

13 Deploying GSpyNetTree as part of the LIGO-Virgo Data Quality Report

Having adapted GSpyNetTree to its multi-label O4 version, we adapted the broader architecture that integrates GSpyNetTree to the DQR. The new architecture is shown in Figure 13.1.

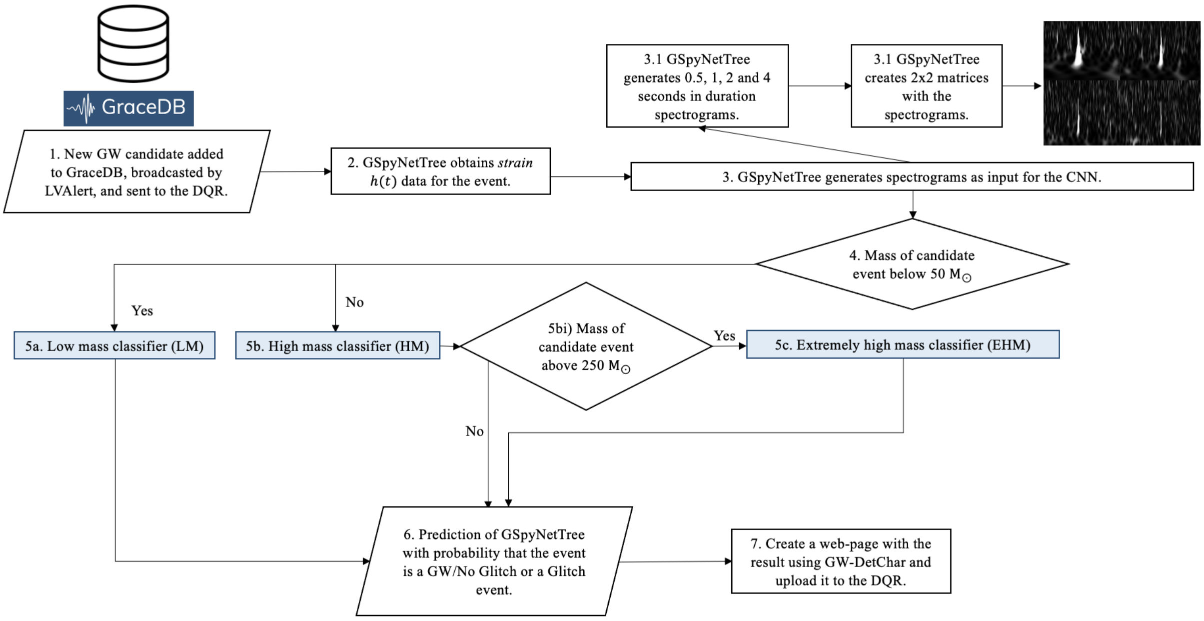


Figure 13.1: GSpyNetTree multi-label architecture: Triggered by a GraceDB superevent [98], timeseries (strain $h(t)$) data is fetched to generate spectrograms of 0.5, 1, 2, and 4 second durations. Time-frequency spectrogram visualizations are sent to the classifiers based on the estimated total candidate mass. Each CNN outputs the probability that the input visualization contains a GW and/or an included class of glitch, or no glitch. The main difference with respect to the previous architecture is the fact that no Mercator projection is applied in the EHM classifier, and GSpyNetTree outputs the probability for all of the labels it was trained on. In addition, more than one label can be predicted by GSpyNetTree in this new use case.

Note that the Mercator projection is no longer applied in the EHM classifier. However, the events are still sent from the HM classifier to the EHM classifier whenever the mass of the candidate is above 250 $M_\odot$. This means that each EHM candidate is first treated as a HM signal, so that it can be compared to Koi Fish and Tomte glitches. In case the candidate event was calculated to have a mass above 250 $M_\odot$, but it was classified as a Tomte or Koi Fish, it will not be sent to the EHM classifier. This is done to reduce the number of glitches this classifier is trained on.

In this O4-version of GSpyNetTree, we focused on developing all the necessary products to deploy our tool in the Data Quality Report. As explained in Section 7.1.6, all DQR tasks report a glitch $p-$value. If this value is below a given threshold, 0.5 in the case of GSpyNetTree, a Data Quality issue is triggered by a task. If it is above such value, no Data Quality issues are flagged and the event is said to be clean. Note that the $p-$value of all tasks is considered when making the decision of flagging or not a candidate with a DQ issue.

The following sections shows the documentation of GSpyNetTree, as seen in the DQR tasks documentation[1] [17]. Additionally, two examples of the output webpage generated by GSpyNetTree, using `GWDetchar`, are shown.

13.1. GspyNetTree documentation for the Data Quality Report

This section shows the documentation of GSpyNetTree, which is also publicly available in the DQR tasks' documentation. The most relevant section of the documentation is the explanation of the calculation of the $p-$value for GSpyNetTree. The relevant fragment of the documentation reads as follows [17]:

- "GSpyNetTree, the Gravity Spy Convolutional Neural Network Decision Tree, is a data quality report task that uses machine learning to determine whether a glitch is present at the time of a candidate event. GSpyNetTree leverages a decision tree of multi-label CNN classifiers, sorted via total estimated gravitational-wave (GW) candidate mass, and trained with morphologically similar glitches. This task is based on Alvarez-Lopez et al. [14], and a new paper on the O4-version of GSpyNetTree is in preparation.

  [. . .] GSpyNetTree leverages a multi-label architecture for its three CNNs, which means it also considers cases where a GW candidate and a glitch overlap in time (and frequency). With a multi-label architecture, GSpyNetTree is able to predict 0 or more labels for each candidate, by returning a probability ranging from 0 to 1 for each considered class. This way, the sum of the probabilities of all labels is not 1 (as occurs for multi-class classifiers, where the classes are mutually exclusive). Instead, the probability of each label can take any value from 0 to 1, and a label is said to be predicted by GSpyNetTree if its probability is greater than or equal to 0.5. In the case were no label surpasses the 50% threshold, no labels are predicted and a "human input needed" message is displayed.

  If GSpyNetTree predicts that a glitch is present (including the case where a GW and/or No Glitch label is simultaneously predicted with a glitch), GspyNetTree needs to determine if a data quality issue should be flagged. To do this, GSpyNetTree uses the glitch p-value, which ranges from 0 (data quality issue identified) to 1 (no data quality issue identified). A data quality issue is flagged whenever the p-value is below 0.05.

  The glitch p-value is calculated as $1-\max allglitchprobabilities$, such that if the probability of the glitch is very high, the p-value will be near zero and a data quality issue will be flagged. Similarly, in cases were GSpyNetTree is very confident about a GW/No Glitch prediction, the glitch probabilities are generally very low and the glitch p-value will be almost 1. Note that the GW/No Glitch probability is not used to calculate the glitch p-value".

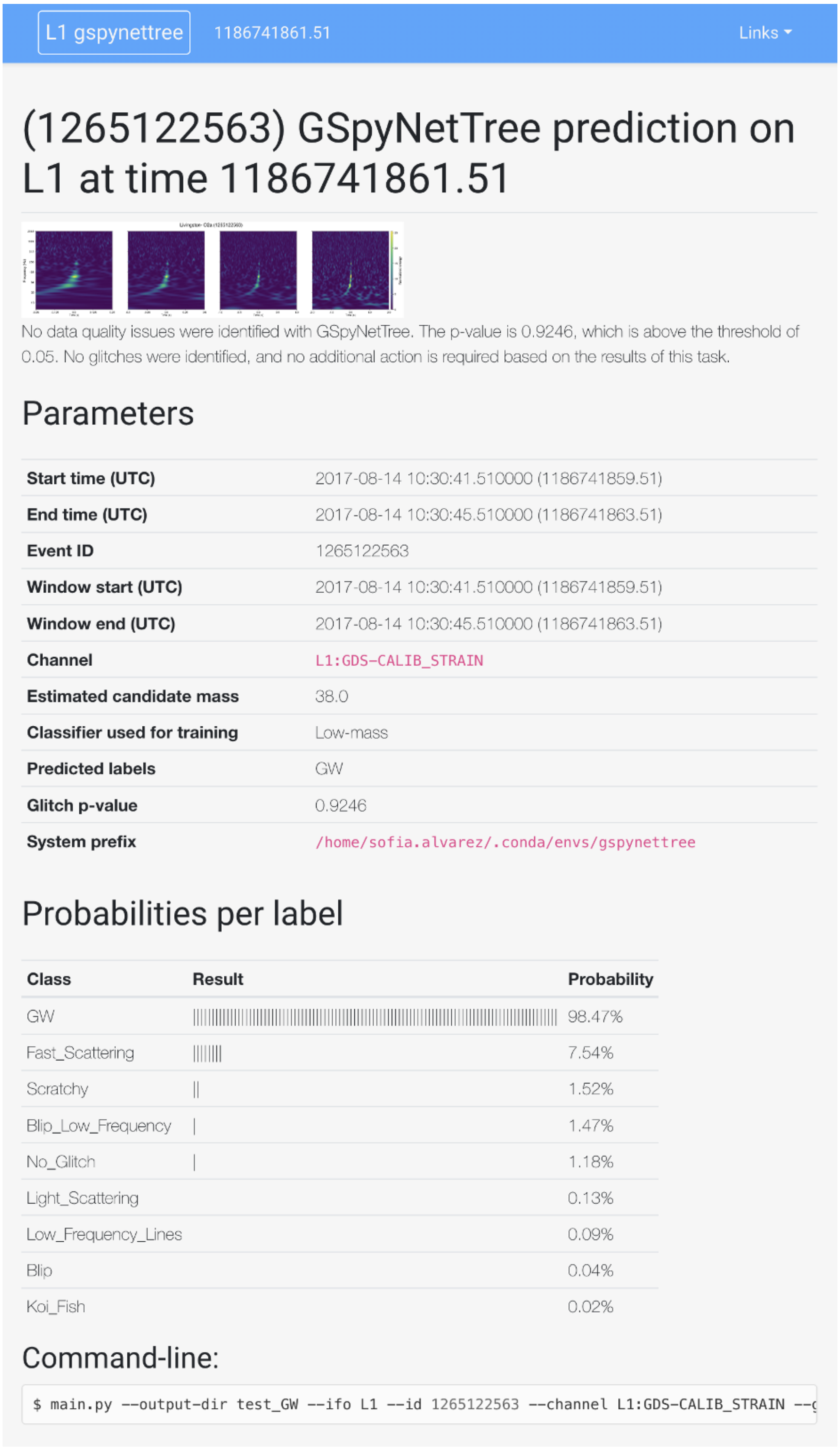
---

[1] The DQR tasks documentation for GSpyNetTree is available at: `https://detchar.docs.ligo.org/dqrtasks/gspynettree.html`.

Having outlined the section of the DQR documentation, it is relevant to show two samples of the output generated by GSpyNetTree. Figure 13.2 shows a candidate event accurately classified as a GW event and Figure 13.3depicts a sample classified as Fast Scattering by GSpyNetTree. Note the different messages displayed when no Data Quality issue is found (first case) and where a Data Quality issue is identified by GSpyNetTree[2].

---

[2] The first sample can be found at `https://detchar.docs.ligo.org/dqrtasks/_static/task_examples/gspynettree/GW_sample/index.html` and, the second one, at `https://detchar.docs.ligo.org/dqrtasks/_static/task_examples/gspynettree/FS_sample/index.html`



(1265122563) GSpyNetTree prediction on L1 at time 1186741861.51

No data quality issues were identified with GSpyNetTree. The p-value is 0.9246, which is above the threshold of 0.05. No glitches were identified, and no additional action is required based on the results of this task.

Parameters

Start time (UTC)	2017-08-14 10:30:41.510000 (1186741859.51)
End time (UTC)	2017-08-14 10:30:45.510000 (1186741863.51)
Event ID	1265122563
Window start (UTC)	2017-08-14 10:30:41.510000 (1186741859.51)
Window end (UTC)	2017-08-14 10:30:45.510000 (1186741863.51)
Channel	`L1:GDS-CALIB_STRAIN`
Estimated candidate mass	38.0
Classifier used for training	Low-mass
Predicted labels	GW
Glitch p-value	0.9246
System prefix	`/home/sofia.alvarez/.conda/envs/gspynettree`

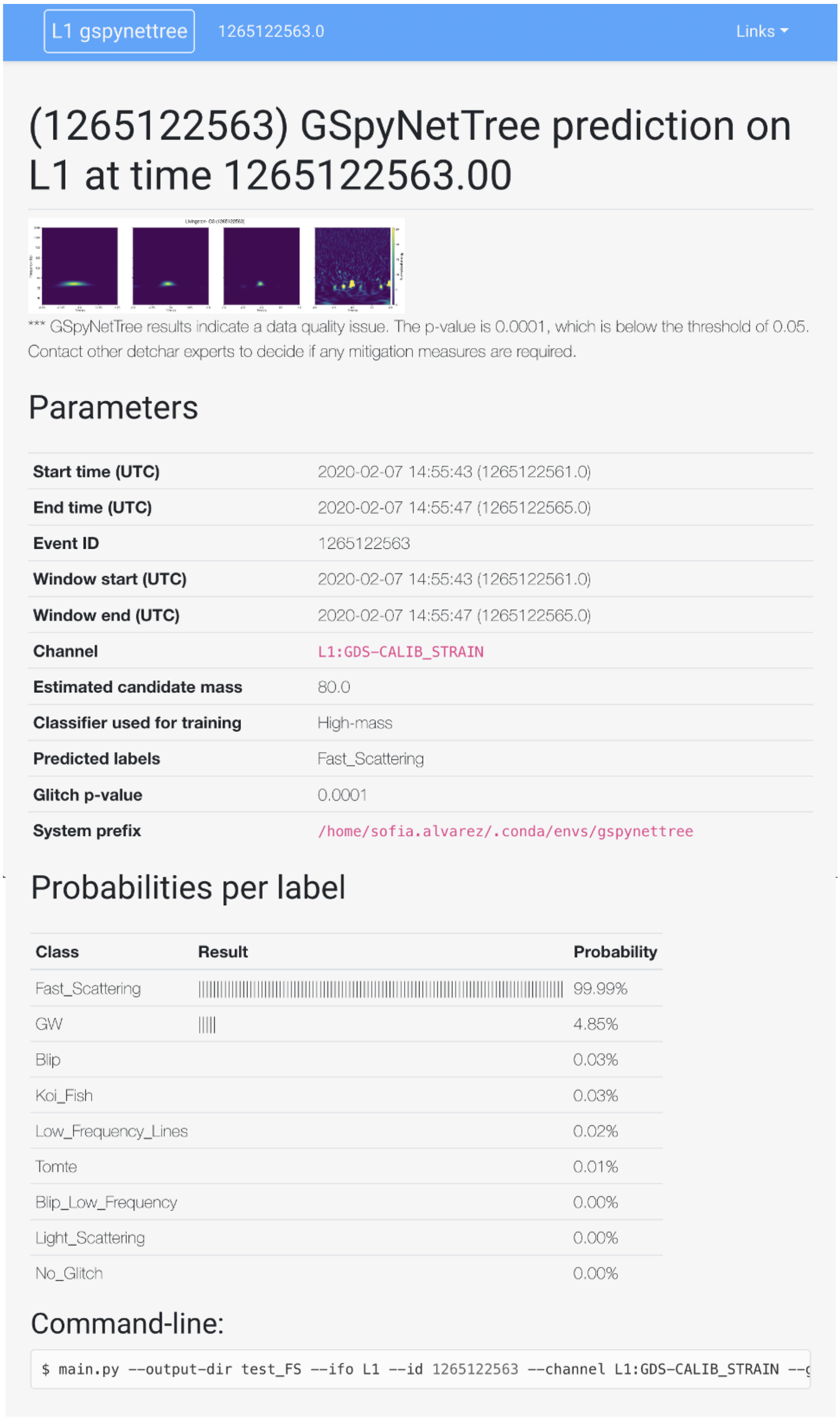
Probabilities per label

<table>
<tr><th>Class</th><th>Result</th><th>Probability</th></tr>
<tr><td>GW</td><td>||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||</td><td>98.47%</td></tr>
<tr><td>Fast_Scattering</td><td>||||||||</td><td>7.54%</td></tr>
<tr><td>Scratchy</td><td>||</td><td>1.52%</td></tr>
<tr><td>Blip_Low_Frequency</td><td>|</td><td>1.47%</td></tr>
<tr><td>No_Glitch</td><td>|</td><td>1.18%</td></tr>
<tr><td>Light_Scattering</td><td></td><td>0.13%</td></tr>
<tr><td>Low_Frequency_Lines</td><td></td><td>0.09%</td></tr>
<tr><td>Blip</td><td></td><td>0.04%</td></tr>
<tr><td>Koi_Fish</td><td></td><td>0.02%</td></tr>
</table>

Command-line:

```
$ main.py --output-dir test_GW --ifo L1 --id 1265122563 --channel L1:GDS-CALIB_STRAIN --
```

Figure 13.2: Output of GSpyNetTree when run over a GW signal of LIGO-Virgo that occurred during the second observing run. Note that the signal does not occur in the proximity of a glitch, and no Data Quality Issue is identified by this task.



(1265122563) GSpyNetTree prediction on L1 at time 1265122563.00

*** GSpyNetTree results indicate a data quality issue. The p-value is 0.0001, which is below the threshold of 0.05. Contact other detchar experts to decide if any mitigation measures are required.

Parameters

Start time (UTC)	2020-02-07 14:55:43 (1265122561.0)
End time (UTC)	2020-02-07 14:55:47 (1265122565.0)
Event ID	1265122563
Window start (UTC)	2020-02-07 14:55:43 (1265122561.0)
Window end (UTC)	2020-02-07 14:55:47 (1265122565.0)
Channel	`L1:GDS-CALIB_STRAIN`
Estimated candidate mass	80.0
Classifier used for training	High-mass
Predicted labels	Fast_Scattering
Glitch p-value	0.0001
System prefix	`/home/sofia.alvarez/.conda/envs/gspynettree`

Probabilities per label

Class	Result	Probability
Fast_Scattering		99.99%
GW		4.85%
Blip		0.03%
Koi_Fish		0.03%
Low_Frequency_Lines		0.02%
Tomte		0.01%
Blip_Low_Frequency		0.00%
Light_Scattering		0.00%
No_Glitch		0.00%

Command-line:

```
$ main.py --output-dir test_FS --ifo L1 --id 1265122563 --channel L1:GDS-CALIB_STRAIN --
```

Figure 13.3: Output of GSpyNetTree when run over a Fast Scattering glitch. Note that the noise transient os correctly classified as a glitch, and a Data Quality Issue is flagged by GSpyNetTree.

14 O4-era GSpyNetTree Results

Before detailing the results of the O4-era version of GSpyNetTree, it is important to understand a few changes that were implemented before building our last version of the classifier. In the first part of this Chapter, some issues we tackled for the first O4-era version version of GSpyNetTree are introduced.

One of the main changes for the EHM classifier in its O4-era version is that it does not implement the Mercator projection for its samples. The EHM classifier has two new glitch classes (Scattering and Fast Scattering), which are much longer in duration than blips, low-frequency blips, and GWs, so the Mercator projection is no longer useful. The advantage of the characteristic thinning of the Low-frequency Blip (see Figure 10.3) with respect to the GW that was previously useful in discerning between astrophysical signals and noise, was now affecting the predictions of other types of glitches. From the original 96% accuracy that we achieved in the pre-O4 era version of GSpyNetTree, we obtained correct classifications of glitch classes of 70% (Light Scattering), 64.5% (Fast Scattering), 75.5% (Blip), and 38.6% (Low-frequency Blip) for the EHM classifier. In contrast, the LM and HM classifiers did not reduce their performance with respect to the pre-O4 GSpyNetTree. Figure 14.1 shows the application of the Mercator projection to a Fast Scattering glitch. Note that it is very difficult to extract useful features from the samples now, and the morphology of the glitch is completely lost. Thus, we removed the application of the Mercator projection from the O4-era version of GSpyNetTree, for which the results improved considerably (see 14.1).

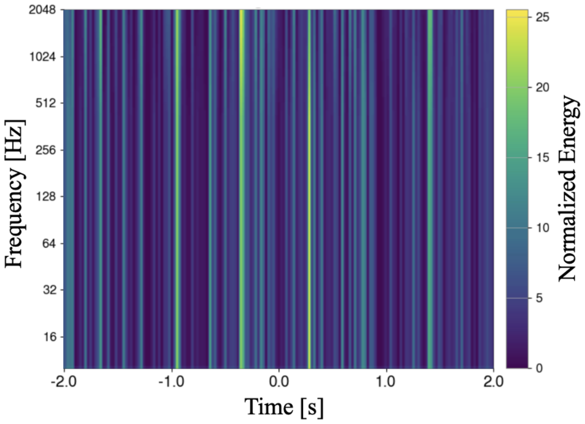


Figure 14.1: Mercator projection applied to a Fast Scattering glitch. Note that the morphology of the glitch is completely overshadowed by the usage of the Mercator projection.

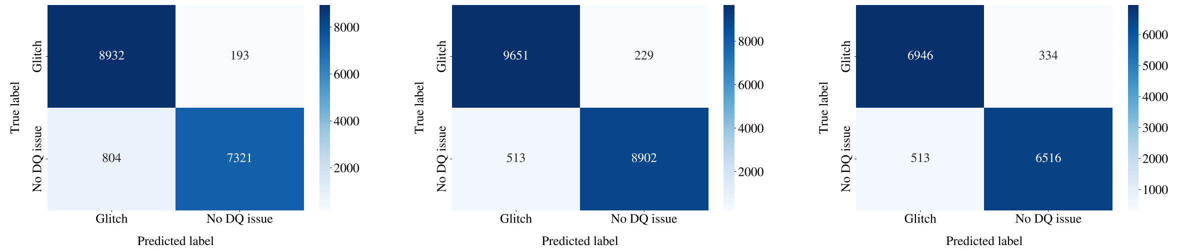
The second fundamental change that the O4-era version of GSpyNetTree has is the inclusion of Low-frequency Lines in all training sets of GSpyNetTree, and of Koi Fish glitches in the LM CNN. We noted that Low-frequency Lines (and similar Low-frequency glitches) were happening frequently in the LIGO detectors in the weeks prior to the start of O4. As this was a recent discovery, we were only able to manually verify around 300 samples of this type of glitch, and no overlapping samples of it with GWs could be generated before the start of O4. This explains the poorer results of this class compared to other GSpyNetTree glitches. A new training set including more Low-frequency Lines (as well as overlapping samples of this glitch with GWs) will be included in a future version of GSpyNetTree. In the case of Koi Fish glitches, while not morphologically similar, we realized that it is important to account for the case in which a loud and saturated glitch occurs at the moment of a GW signal (see the scenario of the glitch overlapping with GW170817 in Figure 6.4). This was an important inclusion based on preliminary results with these types of candidates.

Having outlined the major changes of GSpyNetTree, we can now study its performance in all three mass ranges for each of its CNNs.

14.1. O4 GSpyNetTree: results

Having our training sets and multi-label InceptionV3 CNNs ready, it is now time to understand the results of our classifiers in all of the mass ranges of GSpyNetTree. For visualization purposes, the binary confusion matrices per label per CNN are shown in Appendices B, C, and D, respectively.

The main purpose of the O4-era version of GSpyNetTree is to build a classifier robust to detect glitches responsible for or in the presence of GW candidates. As outlined in Chapter 13, GSpyNetTree will trigger a Data Quality Issue anytime the glitch $p-$value is below 0.05; i.e., whenever a glitch is detected. In this sense, for the results below, the case of predicting a GW or a No Glitch are indistinguishable, as either of them would pass the test imposed by the DQR. In this sense, besides data visualization purposes, we present the results of our CNNs in a binary fashion. Namely, we add the results of all glitch classes together and relabel them as "Glitch", and for the combination of GWs and No Glitch predictions, "No DQ issue identified". Ideally, in the presence of a real GW event, the GW label would be predicted instead of the No Glitch label. Figure 14.2 shows the results for the LM, HM, and EHM classifiers in this binary fashion.



(a) Confusion matrix for the LM classifier of O4-era GSpyNetTree. (b) Confusion matrix for the HM classifier of O4-era GSpyNetTree. (c) Conf. matrix for the EHM CNN of O4-era GSpyNetTree.

Figure 14.2: Binary confusion matrices for the (a) LM, (b) HM, and (c) EHM classifiers for the multi-label versions of the O4-era GSpyNetTree. The results are shown in a binary fashion for data visualization purposes, where the 'Glitch' label includes all glitch classes in each particular classifier, and the 'No DQ Issue' label covers both the GW and the No Glitch class. In these confusion matrices, the number of samples (instead of percentages) is used.

From the three classifiers, the HM classifier shows the best performance among all. It only misclassifies 5.45% of GWs/No Glitch samples as glitches, and 2.32% of glitches as GWs/No

Glitch. The results for misclassified glitches stay low for the LM (2.11%) and EHM (4.58%) classifiers. This is ideal as, responding to its original purpose, GSpyNetTree is flagging more than 95% of all glitches (and overlapping GWs and glitches) as DQ issues. Note that these percentages were calculated using the recall (see Section 8.2.2), meaning that they correspond to the amount of correctly classified samples of each class (where class is either "Glitch" or "No DQ issue".

However, the LM and EHM classifiers classify 9.89% and 7.30% of all GWs/No Glitch instances as glitches, which is a high rate compared to the HM classifier (5.45%). Particularly in the case of the LM classifier, this means that GSpyNetTree has issues classifying very low mass signals in the presence of glitches; and many superevents in the low mass range, especially those overlapping with glitches, will be misclassified as only the type of glitch they are overlapping with. The GW signal will, therefore, be ignored. While not desirable, this is not a considerable issue with GSpyNetTree, as the most important thing for it to tackle is detecting DQ issues in the presence of GWs. Indeed, if a GW candidate (even if it is in the low mass range) has already been detected by several search pipelines in various detectors, the event is most likely to be astrophysical in origin. To further understand these results, it is important to visualize which are the glitch labels most easily mistaken as GWs/No Glitch samples. Sections 14.2 through 14.4 cover these analyses, and study particularities of each classifier.

14.2. Low-mass Classifier results

The results for the incorrectly classified samples in the LM classifier, considering all glitch samples, are shown in Figure 14.3. The left panel shows the number of times GWs were misclassified as each type of glitch; while the right panel shows glitches commonly misclassified as GWs.

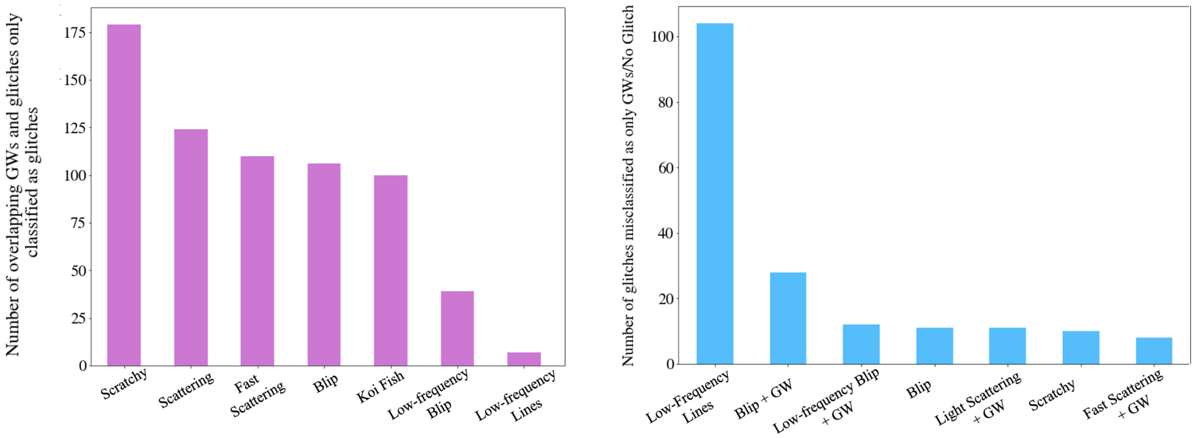


Figure 14.3: *Left:* Number of overlapping GWs and glitches only classified as glitches, per glitch type (four extra labels are not included, as only one instance of each case was predicted). Note that the Scratchy glitch is the one that most hinders low-mass GW signals, possibly due to its long duration. *Right:* Frequency of misclassifications of glitches as GWs/No Glitches. Note that the most problematic glitch class is the Low-frequency Lines.

In the case of the plot in the right, most of the misclassified glitches are Low-frequency Lines.

Indeed, this is the only label for which more than 100 samples are not accurately classified. Manually verifying these samples, we noticed that most of them are classified as No Glitch samples. This is due to the fact that Low-frequency Lines are the worst represented class in GSpyNetTree. By augmenting this class, as we intend to do in the next version of GSpyNetTree, this issue should be fixed. Additionally, we note that the next most frequent misclassified label is overlapping samples of Blips and GWs, in which only the GW is correctly classified. By visualizing a few of the samples, we noticed that in most of the misclassifications, the Blip was very far away from the signal. This way, it is important to build samples more robust to glitches occurring farther away from the signal.

For the plot on the left note that, in general, most of the overlapping GW signals and glitches classified as only glitches are Scratchy glitches; probably because the Scratchy glitch is is longer in duration and covers a wider frequency range, hence it is prone to overshadow a very faint GW signal. Followed by Scratchy glitches, the next most confused are Scattering and Fast Scattering again, probably, because of their duration. Lastly, Blips and Koi Fish are the next most likely glitch types to overshadow a GW signal. While short in duration, both of them have a very high SNR (especially the latter), possibly reducing the attention paid by the CNN to the fainter GW signal. Note that, while still good results, we can further improve the classification of GW signals in the presence of glitches by identifying the best Q-value to generate spectrogram visualizations. This is also proposed as future work on GSpyNetTree.

Another advantage of the current version of GSpyNetTree is that, being multi-label, more than two classes can be predicted; and even two glitch classes (in which the CNN was not trained on) can be identified by GSpyNetTree. As a matter of fact, the plot on the left of Figure 14.3, four predictions are not included (for visualization purposes). The labels predicted in these four cases were: Blip and Scratchy, Fast Scattering and Scratchy, Fast Scattering and Koi Fish, and Blip and Koi Fish. Note that, even though the CNNs are not trained in overlapping glitch spectrograms, the classifier sometimes predicts many glitches in a single sample. If an overlapping sample of a GW and a glitch (or alternatively an independent GW or glitch) is identified by GSpyNetTree, and an extra glitch (or GW) label is predicted, new instances non previously identified (either astrophysical in origin or not) will be flagged by GSpyNetTree. This way, our tool will be able to find GWs (or glitches in proximity with GW candidates) that were not detected by any other GW search pipeline or DQR task, respectively. This is a very interesting scenario, and a sample of a prediction of GSpyNetTree in such a case is shown in Figure 14.4. In this particular prediction, a low-mass GW signal is predicted along with a Low-frequency blip (to the left, original label of the sample) and a Blip (on the right, not originally included in the label of this particular sample). GSpyNetTree is learning the glitches it is trained on in such a way that it is capable of predicting two (and possibly even more) instances of glitches, even though it was not initially trained on these types of samples. This type of predictions can be found in the three mass ranges consudered by GSpyNetTree.

Finally, it is important to highlight that the LM classifier has a very low false alarm rate, defined as the number of actual GWs (and **only** GW samples, without overlapping glitches) that were erroneously classified as glitches. As a matter of fact, this only happens for 2.57% of the signals GSpyNetTree was tested on. This means that GSpyNetTree is very unlikely to

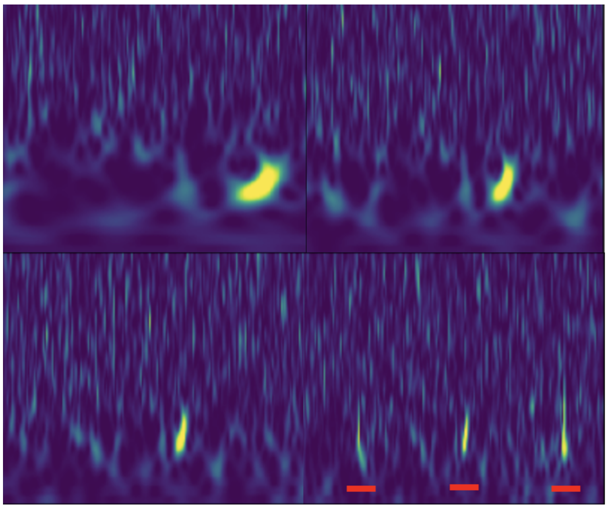


Figure 14.4: GSpyNetTree sample (colored) in which two different glitch labels (Low-frequency Blip and Blip) are predicted along a low-mass GW signal. All three instances are underlined in red in the bottom right plot. Note that, while the Low-frequency Blip is clear in all four spectrograms, in this last plot an artifact (the GW signal) can be seen at the right of Low-frequency Blip. Another artifact, this time a Blip, is visible at the left of the Low-frequency Blip. The two glitches (one of which was not originally labeled by GSpyNetTree - the Blip) and the GW signal were correctly predicted by our tool.

trigger a false DQ issue. This is a fundamental advantage of our tool, as its results flagging DQ issues can be almost always regarded as confident, meaning that the DQR and DetChar experts can rely on the results of GSpyNetTree (complemented of course, by the results of other DQR tasks) to flag a DQ issue whenever GSpyNetTree finds one. However, an improvement to our tool could be providing this rate when reporting the result generated in the webpage uploaded to the DQR report.

14.3. High-mass classifier results

A similar analysis to what was previously done for the LM classifier is shown in Figure 14.5. Just as in the LM classifier, the most problematic glitch class (meaning the one most prone to be classified as a GW/No Glitch, see bottom panel) is the Low-frequency Lines. This is the only glitch class above 80 wrong misclassifications, further confirming the need of augmenting the GSpyNetTree training sets with more samples of this glitch class, along with overlapping samples of such instances with GW signals in all mass mergers. In this sense, we note that the lack of overlapping GW samples with this particular kind of glitch affects significantly the number of correctly classified samples.

On the other hand, the type of glitch that most hinders GW classification is the Blip glitch (see top panel of Figure 14.5, closely followed by Tomte and Scattering glitches. In the case of Blips, note that the second glitch most prone to be ignored in the presence of a GW signal is the Blip glitch (see bottom panel of Figure 14.5). Besides the morphological similarities between GWs in the mass range considered by the HM classifier and the most confused glitches, it is important to note that some high-mass signals of very low SNR may be too faint to be detected

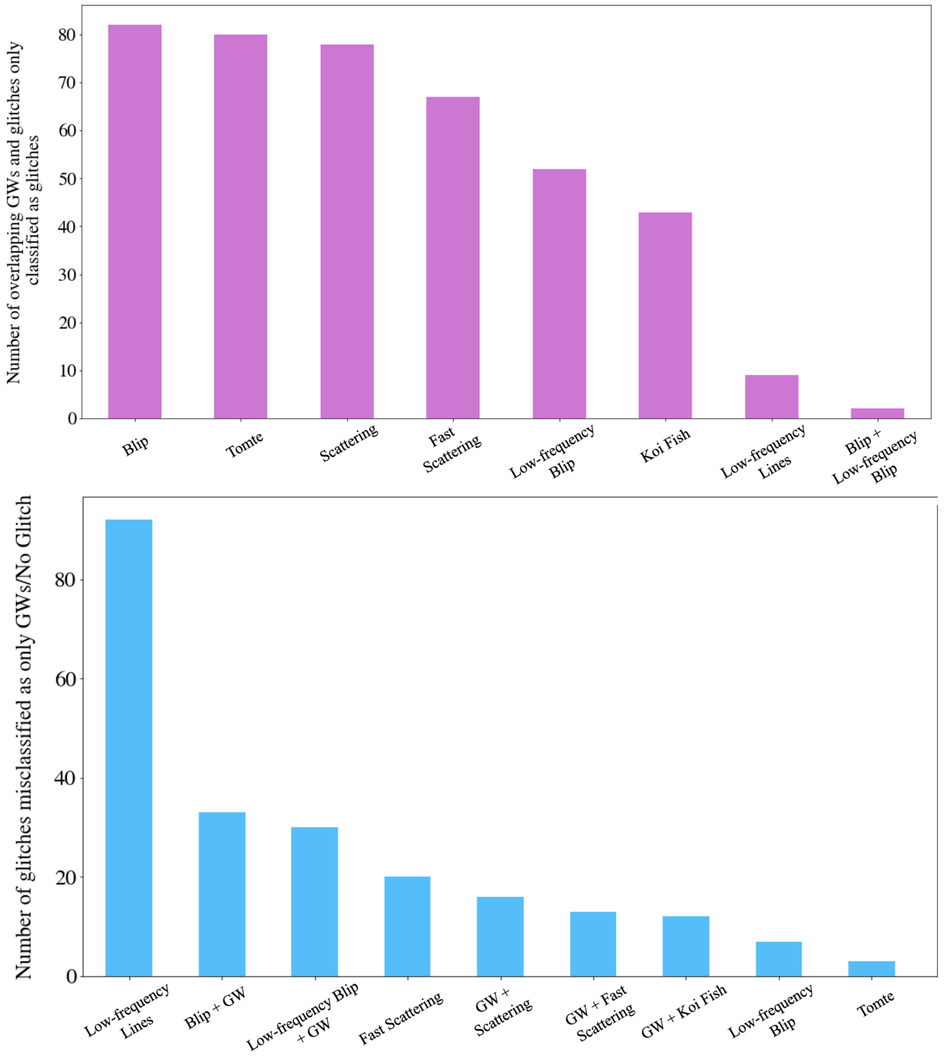


Figure 14.5: *Top:* Number of overlapping GWs and glitches only classified as glitches, per glitch type (four extra labels are not included, as only one instance of each case was predicted). Note that the Blip glitch is the one that most hinders high-mass GW signals, possibly due to the morphological similarities between both of them. *Bottom:* Frequency of misclassifications of glitches as GWs/No Glitches. Note that, as for the LM classifier, the most problematic glitch class is the Low-frequency Lines.

by GSpyNetTree. This way, a louder glitch, in close proximity to the signal, is more likely to be detected by GSpyNetTree. Nevertheless, this happens for just 5% of the candidates overlapping with glitches (the percentage of correctly classified samples is higher for GW signals alone). However, one way to further increase the accurate GW classifications is finding an optimal Q-value for each mass range that better shows the morphology of the GW signal in the presence of a glitch. Finding an optimal Q-value is proposed as future work for GSpyNetTree.

14.4. Extremely high-mass classifier results

Lastly, Figure 14.6 shows the results of GSpyNetTree on the test set of the extremely high-mass classifier. The EHM CNN shows a different behavior compared to the LM and HM classifiers. As a matter of fact note that, when overlapped with glitches, GWs tend to be overshadowed the most by Scattering glitches. Recall that Scattering glitches occur at low frequencies, so much like the EHM mergers due to their high chirp mass (see Chapter 3). In addition to Scattering being one of the longest glitches in duration, it is very likely that segments of the arches of scattering are overlapping with parts of the EHM signal, making it impossible for GSpyNetTree to detect the astrophysical candidate in such cases. To mitigate this effect, longer time offsets between EHM signals and scattering glitches must be considered in a future version of GSpyNetTree.

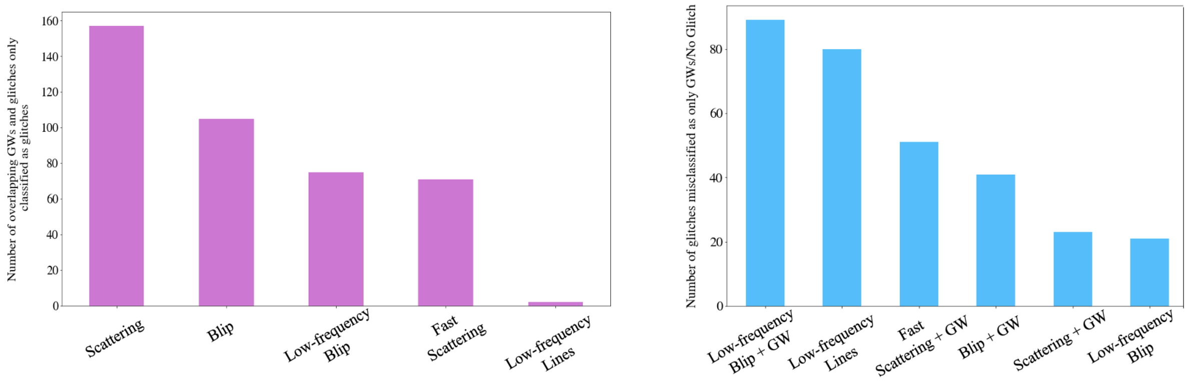


Figure 14.6: *Left:* Number of overlapping GWs and glitches only classified as glitches, per glitch type (four extra labels are not included, as only one instance of each case was predicted). Note that the Scattering glitch is the one that most hinders extremely high-mass GW signals, possibly due to the fact that both occur at low frequencies. *Right:* Frequency of misclassifications of glitches as GWs/No Glitches. Note that, different from the LM and HM classifiers, the most problematic glitch class is the GW + Low-frequency Blip prediction (followed, of course, by the Low-frequency Lines).

Additionally note that, different from the LM and HM classifiers, instead of Low-frequency Lines being the most problematic glitch class mistaken as GW signals/No glitch samples (although they are still in second place), in overlapping Low-frequency Blips and GW signals, the former is usually ignored by the CNN. While the initial approach of GSpyNetTree was to use the Mercator projection to better segregate GW signals from this particular type of glitches, we proved that such approach is no longer useful in the presence of more types of glitches. Additionally, note that the problem is not with Low-frequency Blips alone (still, about 22 of them are misclassified as GWs, although that is a very small fraction of the entire test set); but rather with the overlapping simples of this glitch and GW signals. In this case, the alternative to produce better results is not exactly on the training set, but on the overall GSpyNetTree architecture. In case a candidate event is classified as having either a Low-frequency Blip, a Blip, or both of them, the sample is further sent from the EHM CNN to a binary CNN classifier that segregates GW signals and Low-frequency blips (and overlaps of them) only. As only these two would be included in the specialized CNN, the usage of the Mercator projection to the samples included in this classifier can be evaluated again to better distinguish non-astrophysical signals. This is proposed as future work for the next version of GSpyNetTree.

Note that, while there is room to improve the CNNs, the results obtained for all of them up until now are very promising in classifying glitches responsible for GW candidates or in the presence of real astrophysical signals. As a matter of fact, it has proven its outstanding classification capabilities in several scenarios. The following sections evaluate the performance of GSpyNetTree on detecting repeating instances of glitches (particularly, repeating blips), and on two GW candidates that have been detected since the start of O4 (May 24th, 2023).

14.5. Evaluating the performance of O4-era GSpyNetTree on detecting instances of repeating glitches

One of the most important results from the study of the multi-class pre-O4 version of GSpyNetTree is that it was unable to confidently predict types of glitches it was not originally trained on. While not surprising, we noted that, after Extremely Loud glitches in the HM CNN, the next class with the most confident predictions was Repeating Blips: 56%, 46%, and 52% of all the samples of this glitch in the LM, HM, and EHM classifiers were accurately predicted as Blips. With the new multi-label architecture, we repeated this validation study to verify if the percentage of correctly classified samples increased. As the CNN is now able to predict more than one class, and some samples include a GW signal overlapping with a glitch (i.e., two energy artifacts), we suspect that the CNN will now be able to confidently predict more Repeating Blips samples as well.

Indeed, the LM classifier predicts 90% of Repeating Blips as Blips, a great improvement form the original 56%, while the HM and EHM CNN predict the former as the latter type of glitch 75% and 85% of the times, respectively. All of these are considerable improvements from the multi-class version of GSpyNetTree, suggesting that the O4-version is more robust to a broader set of glitches, including samples with repeating instances of transient noise. This makes GSpyNetTree a valuable tool for GW candidate identification, even in the presence of repeating instances of noise. Note that the unsupervised extension of GSpyNetTree will also be able to predict multiple (and new) sources of glitches, so the results obtained with this supervised multi-class version of GSpyNetTree will be further improved by this new model we are in the process of building.

14.6. Results on selected ER15 and O4 GW candidate events

To conclude the results section, we show the performance of GSpyNetTree on two selected gravitational-wave candidates from the current observing run, O4. Both of them are publicly available in the GraceDB database [98], in the Public Alerts section. The chosen candidates are a retracted GW signal due to its terrestrial origin and a BBH candidate event. Note that I only present the results of GSpyNetTree and not of the other DQR tasks, as their reports are part of the internal review of the collaboration. The spectrograms, mass of each candidate, time of occurrence, and ID are not shown either, as these are not public results yet.

14.6.1. Classification performance on example candidate event 1: retracted event

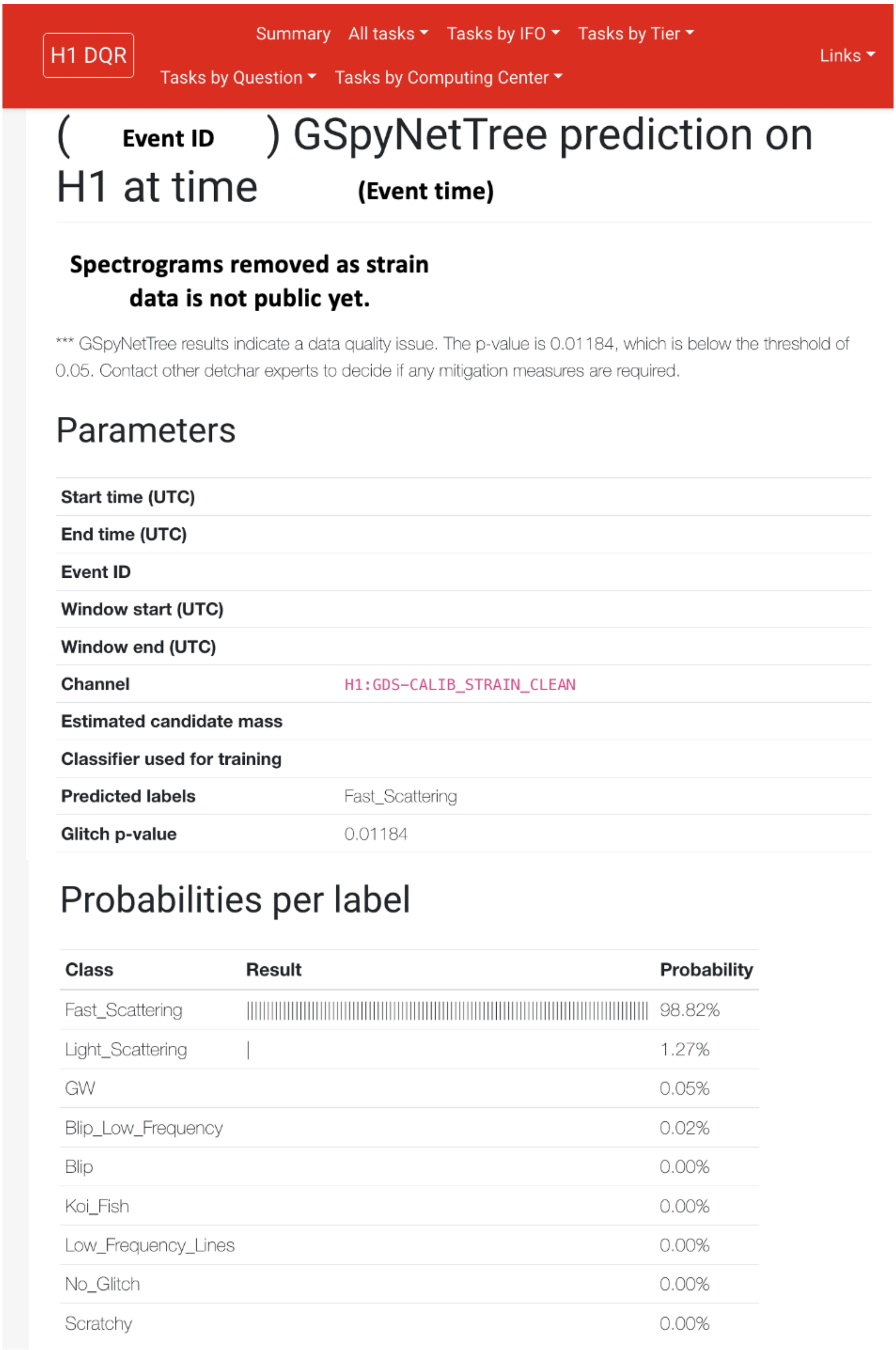
The first example candidate event of O4 is a retracted event, flagged as such because of its terrestrial origin. GSpyNetTree predicted this candidate signal as a Fast Scattering glitch in both detectors, as shown in Figures 14.7 and 14.8 for the LIGO Livingston and LIGO Hanford detectors.

14.6.2. Classification performance on example candidate event 2: a BBH candidate

This candidate event was detected in both LIGO Observatories[1]. Figures 14.9 and 14.10 show the reports generated by GSpyNetTree for this sample in the Hanford and Livingston detectors, respectively. Remarkably, there was a glitch approximately 1.5 seconds after the time of the merger of the GW in the Livingston Observatory. With an 81.62% probability, it is predicted to be a Tomte glitch. While this value did not trigger a Data Quality issue due to the fact that the glitch $p-$value is still above 0.05, the ability of GSpyNetTree to predict glitches in close proximity with GWs is proved.

---

[1] Virgo is expected to join LIGO in O4 later on 2023.



(**Event ID**) GSpyNetTree prediction on H1 at time **(Event time)**

Spectrograms removed as strain data is not public yet.

*** GSpyNetTree results indicate a data quality issue. The p-value is 0.01184, which is below the threshold of 0.05. Contact other detchar experts to decide if any mitigation measures are required.

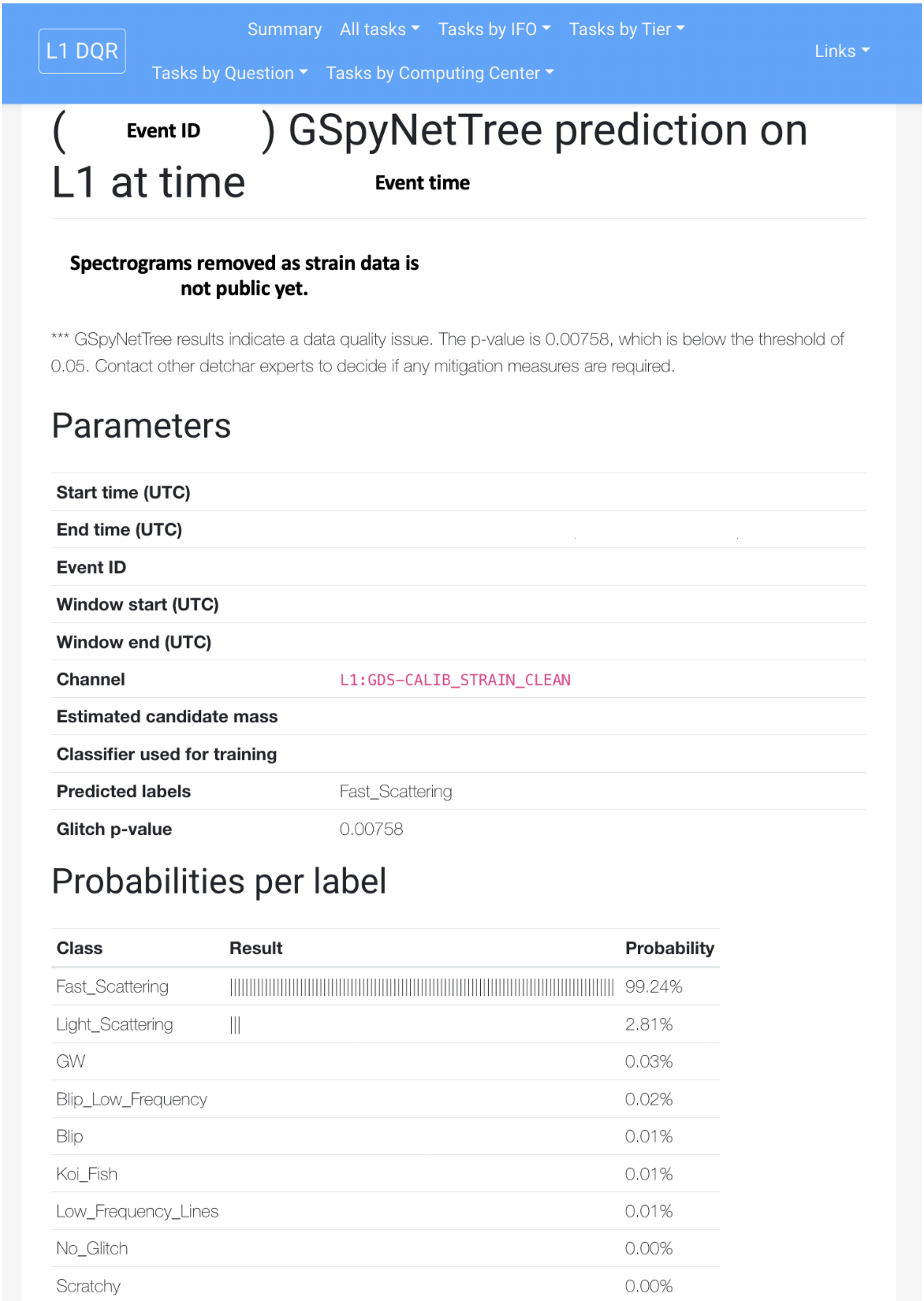
Parameters

Start time (UTC)	
End time (UTC)	
Event ID	
Window start (UTC)	
Window end (UTC)	
Channel	`H1:GDS-CALIB_STRAIN_CLEAN`
Estimated candidate mass	
Classifier used for training	
Predicted labels	Fast_Scattering
Glitch p-value	0.01184

Probabilities per label

Class	**Result**	**Probability**
Fast_Scattering	████████████████████████████████	98.82%
Light_Scattering	▏	1.27%
GW		0.05%
Blip_Low_Frequency		0.02%
Blip		0.00%
Koi_Fish		0.00%
Low_Frequency_Lines		0.00%
No_Glitch		0.00%
Scratchy		0.00%

Figure 14.7: Report webpage generated by GSpyNetTree for the retracted candidate example event 1 in the Hanford Observatory. Note that a Fast Scattering glitch is predicted, indicating a Data Quality issue.



(**Event ID**) GSpyNetTree prediction on L1 at time **Event time**

Spectrograms removed as strain data is not public yet.

*** GSpyNetTree results indicate a data quality issue. The p-value is 0.00758, which is below the threshold of 0.05. Contact other detchar experts to decide if any mitigation measures are required.

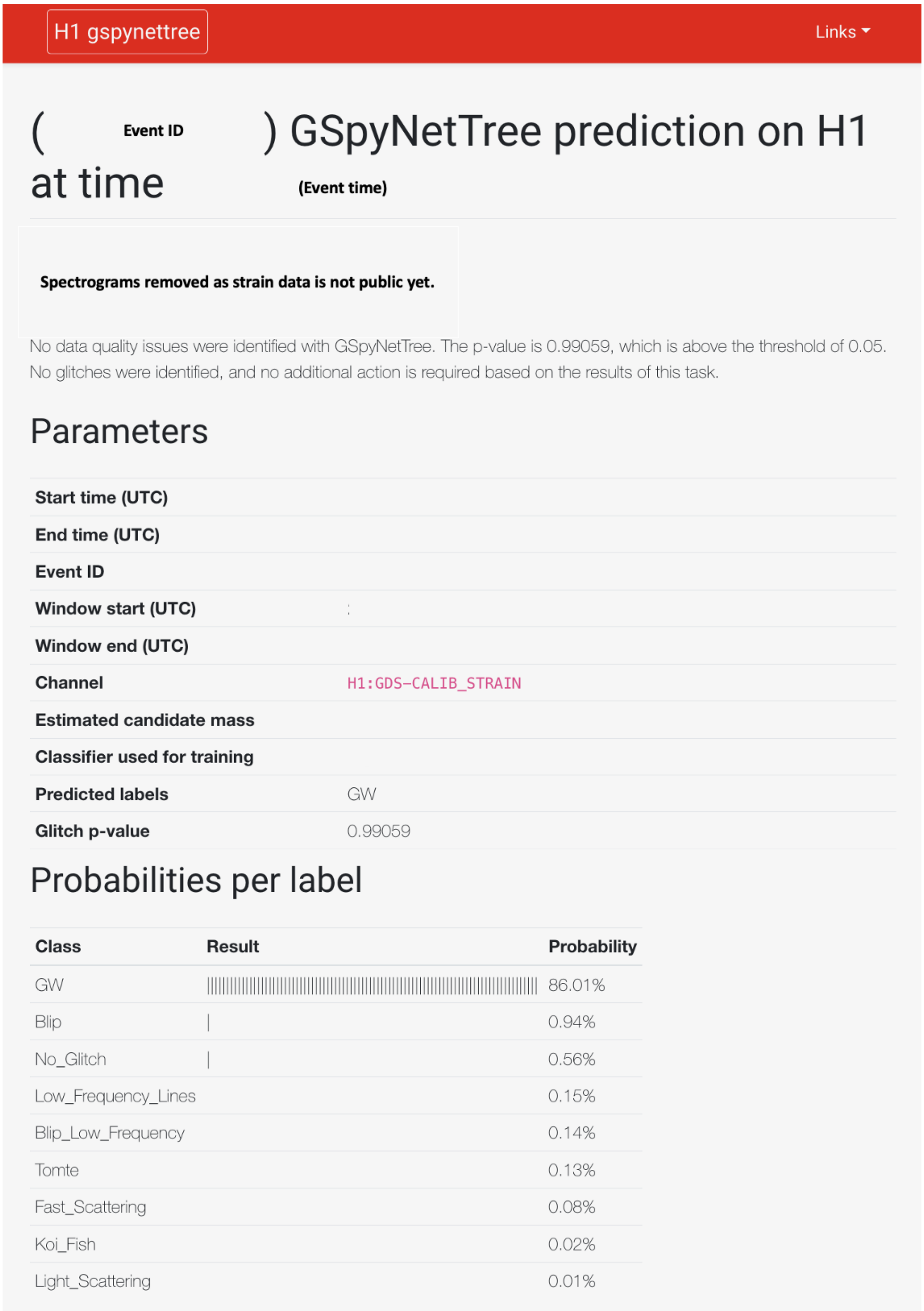
Parameters

Start time (UTC)	
End time (UTC)	
Event ID	
Window start (UTC)	
Window end (UTC)	
Channel	`L1:GDS-CALIB_STRAIN_CLEAN`
Estimated candidate mass	
Classifier used for training	
Predicted labels	Fast_Scattering
Glitch p-value	0.00758

Probabilities per label

Class	**Result**	**Probability**
Fast_Scattering		99.24%
Light_Scattering		2.81%
GW		0.03%
Blip_Low_Frequency		0.02%
Blip		0.01%
Koi_Fish		0.01%
Low_Frequency_Lines		0.01%
No_Glitch		0.00%
Scratchy		0.00%

Figure 14.8: Report webpage generated by GSpyNetTree for the retracted candidate example event 1 in the Livingston Observatory. Note that a Fast Scattering glitch is predicted, indicating a Data Quality issue.



(Event ID) GSpyNetTree prediction on H1 at time (Event time)

Spectrograms removed as strain data is not public yet.

No data quality issues were identified with GSpyNetTree. The p-value is 0.99059, which is above the threshold of 0.05. No glitches were identified, and no additional action is required based on the results of this task.

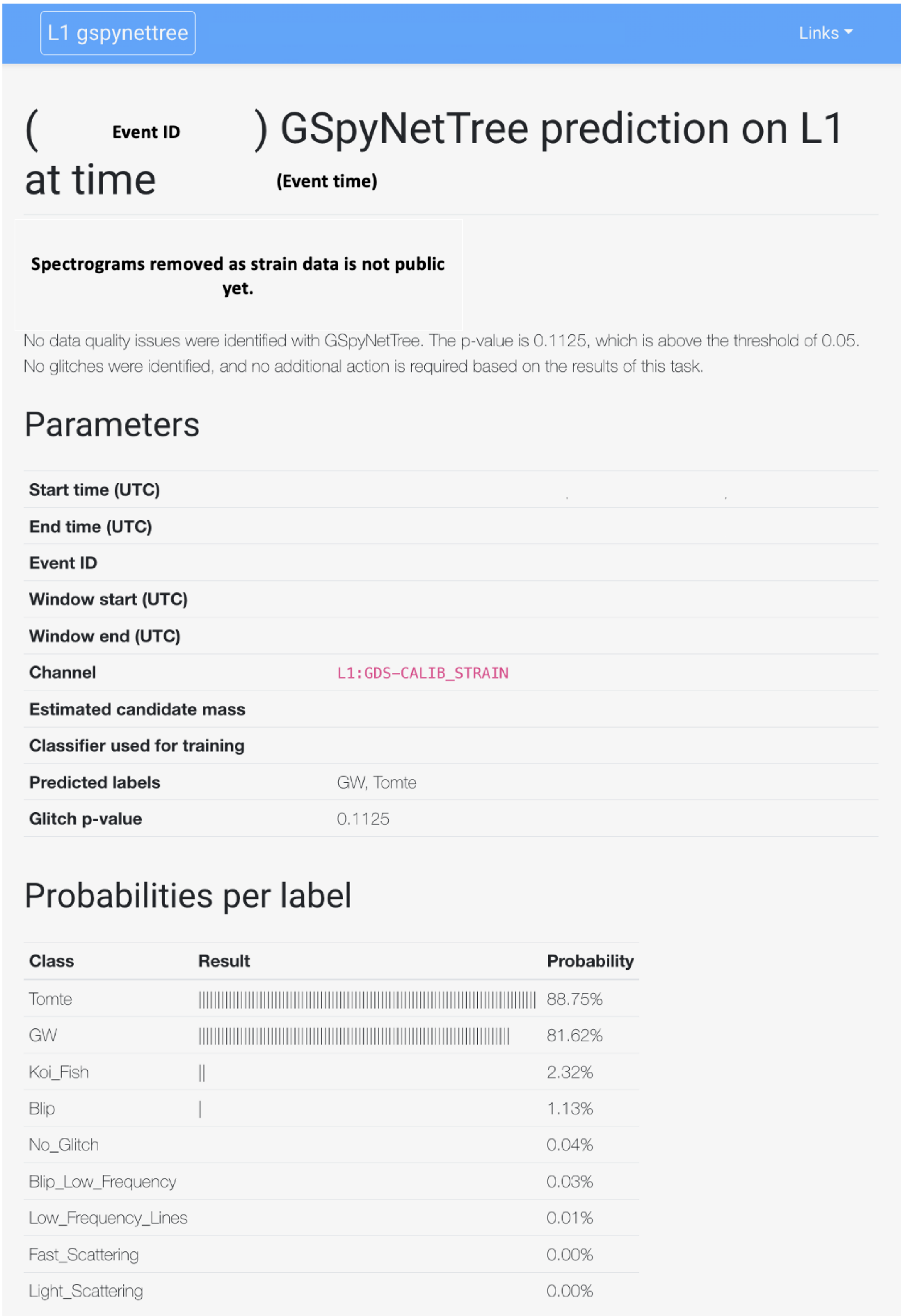
Parameters

Start time (UTC)	
End time (UTC)	
Event ID	
Window start (UTC)	:
Window end (UTC)	
Channel	`H1:GDS-CALIB_STRAIN`
Estimated candidate mass	
Classifier used for training	
Predicted labels	GW
Glitch p-value	0.99059

Probabilities per label

<table>
<tr><th>Class</th><th>Result</th><th>Probability</th></tr>
<tr><td>GW</td><td>||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||</td><td>86.01%</td></tr>
<tr><td>Blip</td><td>|</td><td>0.94%</td></tr>
<tr><td>No_Glitch</td><td>|</td><td>0.56%</td></tr>
<tr><td>Low_Frequency_Lines</td><td></td><td>0.15%</td></tr>
<tr><td>Blip_Low_Frequency</td><td></td><td>0.14%</td></tr>
<tr><td>Tomte</td><td></td><td>0.13%</td></tr>
<tr><td>Fast_Scattering</td><td></td><td>0.08%</td></tr>
<tr><td>Koi_Fish</td><td></td><td>0.02%</td></tr>
<tr><td>Light_Scattering</td><td></td><td>0.01%</td></tr>
</table>

Figure 14.9: Report webpage generated by GSpyNetTree for the second example GW candidate event in the Hanford Observatory. A GW is predicted by GSpyNetTree, with a 86% of probability that the event is a BBH signal.



(Event ID) GSpyNetTree prediction on L1 at time (Event time)

Spectrograms removed as strain data is not public yet.

No data quality issues were identified with GSpyNetTree. The p-value is 0.1125, which is above the threshold of 0.05. No glitches were identified, and no additional action is required based on the results of this task.

Parameters

Start time (UTC)	
End time (UTC)	
Event ID	
Window start (UTC)	
Window end (UTC)	
Channel	`L1:GDS-CALIB_STRAIN`
Estimated candidate mass	
Classifier used for training	
Predicted labels	GW, Tomte
Glitch p-value	0.1125

Probabilities per label

Class	Result	Probability
Tomte	||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||	88.75%
GW	|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||	81.62%
Koi_Fish	||	2.32%
Blip	|	1.13%
No_Glitch		0.04%
Blip_Low_Frequency		0.03%
Low_Frequency_Lines		0.01%
Fast_Scattering		0.00%
Light_Scattering		0.00%

Figure 14.10: Report webpage generated by GSpyNetTree for the second example GW candidate event in the Hanford Observatory. A GW is predicted by GSpyNetTree, with an 81.6% of probability that the event is a BBH signal. A Tomte glitch is also predicted to occur in close proximity to the GW, proving the ability of GSpyNetTree to detect GWs in the presence of glitches.

15 Conclusions and future work

This dissertation describes the development of GSpyNetTree: the Gravity Spy Convolutional Neural Network Decision tree, a signal-vs-glitch classifier for LIGO-Virgo GW event candidates that is part of the Data Quality Report of the fourth observing run. GSpyNetTree is part of the GW event validation pipeline, and its results have been already utilized to identify glitches responsible for or in the presence of GW candidates. The entire evolution of GSpyNetTree is presented: from its initial, pre-O4 multi-class version to its current, O4 multi-label architecture. Following a procedure of tuning the best possible models, and generating training sets robust to a broad array of background noise, glitches, and overlapping samples of GWs and glitches, we achieved more than 95% correct classifications in all glitches, both morphologically similar to GWs, and occurring in the proximity of real astrophysical events. This means that less than 5% of the Data Quality issues currently present in the LIGO-Virgo data are not identified by GSpyNetTree. As a matter of fact, in the already started observing run, GSpyNetTree has had an outstanding performance in segregating GW candidate events from glitches in the superevents published in GraceDB [98] up to date.

In addition, more than 90% of GW candidates and No Glitch samples are classified correctly among the entire mass spectrum (i.e., from 5 $M_\odot$ to 350 $M_\odot$). The misclassified GW samples usually occur for low SNR GW signals; which, if happening in close proximity with a glitch, may be completely overshadowed by the latter. This is an expected outcome, as it is difficult for GSpyNetTree to disentangle the power of a very faint signal from a very loud glitch. However, as long as the DQ issue is identified by the task and if an overlapping GW signal occurred at the same time as a non-astrophysical event, further mitigation and subtraction techniques may be applied to recover the GW signal in its entirety (as it was done for GW170817). Additionally, the tool is able to classify accurately more than 75% of repeating instances of glitches when tested over a set of Repeating Blips in the entire mass range considered by GSpyNetTree. This is a considerable increase with respect to the multi-class version of the task, and suggests that this new version of GSpyNetTree is more robust in predicting repeating instances of the morphologically similar glitches it already knows. Moreover, a very valuable contribution of GSpyNetTree to the DQR is the fact that it can predict more than one class of glitch. This way, even if multiple types of glitches occur in the proximity of GWs, GSpyNetTree will be able to detect them.

It is important to note, however, that GSpyNetTree performs poorly on instances of Low-frequency Lines; this is because they are an underrepresented class in the training set. For a future version of GSpyNetTree, these classifications will be improved (both including more manually verified samples of the glitch, and introducing instances of overlapping GWs with it). Furthermore, many GW signals are ignored because of the Q-value (of 20) used. A future version of GSpyNetTree will study a broader range of Q-values to generate the spectrograms of GSpyNetTree, aiming to find an optimized value that maximizes the visible GW signal (even for low SNR samples). Moreover, in order to further improve the classifications in the EHM classifier, which has a significant issue identifying GWs in the presence of Low-frequency blips (but not confusing astrophysical signals as these type of glitch, as in the multi-class version of GSpyNetTree), we propose the implementation of a specialized multi-label binary CNN that is triggered whenever an event is cataloged with having a GW, a Low-frequency Blip, or both. As this new classifier will only have two classes to predict, the usage of the Mercator projection can be reconsidered.

Finally, and based on the recommendations of Alvarez-Lopez et al. [14] a future version of GSpyNetTree will also include an unsupervised extension. In this new model, samples, for which GSpyNetTree's predictions are not confident, will be clustered to identify new possible sources of glitches (either very common or morphologically similar to GWs) in which GSpyNetTree was not initially trained on. This will allow, in the future, to deploy an automated GSpyNetTree system which will be updated with the new clusters (i.e., labels) that, based on the O4 data, the unsupervised architecture will generate.

Bibliography

[1] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), «Observation of Gravitational Waves from a Binary Black Hole Merger», Phys. Rev. Lett. **116**, 061102 (2016).

[2] J. McIver and D. H. Shoemaker, «Discovering gravitational waves with advanced ligo», Contemporary Physics **61**, 229–255 (2020).

[3] M. Zevin et al., «Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science», Classical and Quantum Gravity **34**, arXiv:1611.04596 [astro-ph, physics:gr-qc, physics:physics], 064003 (2017).

[4] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), «Gw150914: the advanced ligo detectors in the era of first discoveries», Phys. Rev. Lett. **116**, 131103 (2016).

[5] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), «Calibration of the Advanced LIGO detectors for the discovery of the binary black-hole merger GW150914», Physical Review D **95**, arXiv:1602.03845 [astro-ph, physics:gr-qc, physics:physics], 062003 (2017).

[6] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), «GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs», en, `10.1103/PhysRevX.9.031040`.

[7] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), «GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run», en, `10.1103/PhysRevX.11.021053`.

[8] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration and KAGRA Collaboration), «GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo During the Second Part of the Third Observing Run», en, `10.48550/arXiv.2111.03606`.

[9] J. Harms, «Terrestrial Gravity Fluctuations», Living Reviews in Relativity **22**, arXiv:1507.05850 [gr-qc], 6 (2019).

[10] The LIGO Scientific Collaboration and the Virgo Collaboration, «Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914», Classical and Quantum Gravity **33**, arXiv:1602.03844 [astro-ph, physics:gr-qc, physics:physics], 134001 (2016).

[11] B. Berger, «Identification and mitigation of advanced ligo noise sources», Journal of Physics: Conference Series **957**, 012004 (2018).

[12] D. Davis et al., «LIGO Detector Characterization in the Second and Third Observing Runs», Classical and Quantum Gravity **38**, arXiv:2101.11673 [astro-ph, physics:gr-qc], 135014 (2021).

[13] Jarov, S. et al., «A new method to distinguish gravitational-wave signals from detector glitches with Gravity Spy».

[14] S. Alvarez-Lopez et al., *GSpyNetTree: A signal-vs-glitch classifier for gravitational-wave event candidates*, 2023.

[15] C. Szegedy et al., «Rethinking the inception architecture for computer vision», in 2016 ieee conference on computer vision and pattern recognition (cvpr) (2016), pp. 2818–2826.

[16] The LIGO Scientific Collaboration and The Virgo Collaboration, «Data Quality Report user documentation», `https://docs.ligo.org/detchar/data-quality-report/` (2018).

[17] The LIGO Scientific Collaboration and The Virgo Collaboration, «Data Quality Report (DQR) tasks documentation», `https://detchar.docs.ligo.org/dqrtasks/index.html` (2023).

[18] P. R. Saulson, *Fundamentals of interferometric gravitational wave detectors* (World Scientific, 1994).

[19] P. R. Saulson, «Gravitational wave detection: principles and practice», Comptes Rendus Physique **14**, 288–305 (2013).

[20] M. Maggiore, *Gravitational waves: volume 1: theory and experiments* (OUP Oxford, 2007).

[21] E. E. Flanagan and S. A. Hughes, «The basics of gravitational wave theory», New Journal of Physics **7**, 204 (2005).

[22] K. S. Thorne, J. A. Wheeler, and C. W. Misner, *Gravitation* (Freeman San Francisco, CA, 2000).

[23] B. S. Sathyaprakash and B. F. Schutz, «Physics, astrophysics and cosmology with gravitational waves», Living reviews in relativity **12**, 1–141 (2009).

[24] D. J. Griffiths, *Introduction to Electrodynamics* (Pearson, 2013).

[25] S. Carroll, *Spacetime and geometry: an introduction to general relativity* (Benjamin Cummings, 2003).

[26] J. H. Taylor, «Pulsar timing and relativistic gravity», Classical and Quantum Gravity **10**, S167 (1993).

[27] J. M. Weisberg, D. J. Nice, and J. H. Taylor, «Timing Measurements of the Relativistic Binary Pulsar PSR B1913+16», Astrophys. J. **722**, 1030–1034 (2010).

[28] S. Vitale, «The first 5 years of gravitational-wave astrophysics», Science **372**, eabc7397 (2021).

[29] L. S. Finn and D. F. Chernoff, «Observing binary inspiral in gravitational radiation: One interferometer», Phys. Rev. D **47**, 2198–2219 (1993).

[30] L. Blanchet, «Gravitational radiation from post-newtonian sources and inspiralling compact binaries», Living Reviews in Relativity **17**, `10.12942/lrr-2014-2` (2014).

[31] J. G. Baker et al., «Gravitational-wave extraction from an inspiraling configuration of merging black holes», Phys. Rev. Lett. **96**, 111102 (2006).

[32] A. Heger et al., «How Massive Single Stars End Their Life», The Astrophysical Journal **591**, 288 (2003).

[33] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), «Improved Analysis of GW150914 Using a Fully Spin-Precessing Waveform Model», Phys. Rev. X **6**, 041014 (2016).

[34] R. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), «Gw190521: a binary black hole merger with a total mass of 150 $M_{\odot}$», Phys. Rev. Lett. **125**, 101102 (2020).

[35] S. A. Hughes, *Trust but verify: the case for astrophysical black holes*, 2005.

[36] R. N. Manchester, G. B. Hobbs, A. Teoh, and M. Hobbs, «The Australia Telescope National Facility Pulsar Catalogue», The Astronomical Journal **129**, 1993–2006 (2005).

[37] L. Rezzolla, E. R. Most, and L. R. Weih, «Using Gravitational-wave Observations and Quasi-universal Relations to Constrain the Maximum Mass of Neutron Stars», The Astrophysical Journal **852**, L25, L25 (2018).

[38] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), «GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral», Phys. Rev. Lett. **119**, 161101 (2017).

[39] B. P. Abbott et al., «Multi-messenger Observations of a Binary Neutron Star Merger», The Astrophysical Journal **848**, L12 (2017).

[40] H.-Y. Chen et al., «Distance measures in gravitational-wave astrophysics and cosmology», Classical and Quantum Gravity **38**, 055010 (2021).

[41] T. M. Tauris et al., «Formation of Double Neutron Star Systems», The Astrophysical Journal **846**, 170 (2017).

[42] B. P. Abbott et al., «Gravitational Waves and Gamma-Rays from a Binary Neutron Star Merger: GW170817 and GRB 170817A», The Astrophysical Journal **848**, L13 (2017).

[43] R. Abbott et al., «Observation of Gravitational Waves from Two Neutron Star–Black Hole Coalescences», The Astrophysical Journal Letters **915**, L5 (2021).

[44] F. Foucart, «A brief overview of black hole-neutron star mergers», Frontiers in Astronomy and Space Sciences **7**, `10.3389/fspas.2020.00046` (2020).

[45] H. Bethe, G. Brown, and C.-H. Lee, «Formation And Evolution Of Black Holes In The Galaxy», `10.1142/5142` (2003).

[46] T. R. Marsh et al., «A radio-pulsing white dwarf binary star», Nature **537**, 374–377 (2016).

[47] B. Anguiano et al., «White dwarf binaries across the h-r diagram», The Astronomical Journal **164**, 126 (2022).

[48] P. Amaro-Seoane et al., *Laser Interferometer Space Antenna*, 2017.

[49] G. Ushomirsky, C. Cutler, and L. Bildsten, «Deformations of accreting neutron star crusts and gravitational wave emission», Monthly Notices of the Royal Astronomical Society **319**, 902–932 (2002).

[50] LIGO Scientific Collaboration (LIGO Scientific Collaboration), *Introduction to LIGO and gravitational-waves.*

[51] V. Morozova, D. Radice, A. Burrows, and D. Vartanyan, «The Gravitational Wave Signal from Core-collapse Supernovae», The Astrophysical Journal **861**, 10 (2018).

[52] B. P. Abbott et al., «Search for Transient Gravitational-wave Signals Associated with Magnetar Bursts during Advanced LIGO's Second Observing Run», The Astrophysical Journal **874**, 163 (2019).

[53] B. Kocsis, M. E. Gáspár, and S. Márka, «Detection Rate Estimates of Gravity Waves Emitted during Parabolic Encounters of Stellar Black Holes in Globular Clusters», The Astrophysical Journal **648**, 411–429 (2006).

[54] D. Reitze et al., *Cosmic Explorer: The U.S. Contribution to Gravitational-Wave Astronomy beyond LIGO*, 2019.

[55] M. Maggiore et al., «Science case for the Einstein telescope», Journal of Cosmology and Astroparticle Physics **2020**, 050–050 (2020).

[56] V. Srivastava et al., «Detection prospects of core-collapse supernovae with supernova-optimized third-generation gravitational-wave detectors», Physical Review D **100**, `10.1103/physrevd.100.043026` (2019).

[57] N. Christensen, «Stochastic gravitational wave backgrounds», Reports on Progress in Physics **82**, 016903 (2018).

[58] A. Einstein, «Näherungsweise integration der feldgleichungen der gravitation. sitzungsberichte der königlich preußischen akademie der wissenschaften (berlin)», Translated as Approximative Integration of the Field Equations of Gravitation, in Alfred Engel (translator) and Engelbert Schucking (consultant), The Collected Papers of Albert Einstein **6**, 1914–1917 (1916).

[59] A. Einstein, «On Gravitational Waves», Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.), 154 (1918).

[60] D. Kennefick, «Controversies in the history of the radiation reaction problem in general relativity», arXiv preprint gr-qc/9704002 (1997).

[61] J. Weber, «Gravitational-wave-detector events», Physical Review Letters **20**, 1307 (1968).

[62] G. Pizzella, «Birth and initial developments of experiments with resonant detectors searching for gravitational waves», The European Physical Journal H **41**, 267–302 (2016).

[63] M. Gertsenshtein, «Wave resonance of light and gravitional waves», Sov Phys JETP **14**, 84–85 (1962).

[64] G. Moss, L. Miller, and R. Forward, «Photon-noise-limited laser transducer for gravitational antenna», Applied Optics **10**, 2495–2498 (1971).

[65] R. Weiss and D. Muehlner, *Electronically coupled broadband gravitational antenna* (Citeseer, 1972).

[66] B. F. Schutz, «Networks of gravitational wave detectors and three figures of merit», Classical and Quantum Gravity **28**, 125023 (2011).

[67] J. Abadie et al., «Calibration of the LIGO gravitational wave detectors in the fifth science run», Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **624**, 223–240 (2010).

[68] P. Brady, G. Losurdo, and H. Shinkai, «LIGO, VIRGO, and KAGRA as the International Gravitational Wave Network», in *Handbook of Gravitational Wave Astronomy*, edited by C. Bambi, S. Katsanevas, and K. D. Kokkotas (Springer Singapore, Singapore, 2020), pp. 1–21.

[69] J. Aasi et al., «Advanced LIGO», Classical and Quantum Gravity **32**, 074001 (2015).

[70] C. L. Mueller et al., «The advanced LIGO input optics», Review of Scientific Instruments **87**, 014502, `10.1063/1.4936974` (2016).

[71] LIGO Scientific Collaboration (LIGO Scientific Collaboration), *LIGO Optics*.

[72] D. Davis, «Improving the Sensitivity of Advanced LIGO Through Detector Characterization», PhD thesis (Syracuse University, 2019).

[73] B. P. Abbott et al., «Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA», Living Reviews in Relativity **23**, `10.1007/s41114-020-00026-9` (2020).

[74] C. Cahillane and G. Mansell, «Review of the Advanced LIGO Gravitational Wave Observatories Leading to Observing Run Four», Galaxies **10**, `10.3390/galaxies10010036` (2022).

[75] D. Davis, L. V. White, and P. R. Saulson, «Utilizing aLIGO glitch classifications to validate gravitational-wave candidates», Classical and Quantum Gravity **37**, 145001 (2020).

[76] J. G. Rollins, E. Hall, C. Wipf, and L. McCuller, *pygwinc: Gravitational Wave Interferometer Noise Calculator*, Astrophysics Source Code Library, record ascl:2007.020, July 2020.

[77] R. Poggiani, «Gravitational Wave Detectors», in *Encyclopedia of Physical Science and Technology (Third Edition)*, edited by R. A. Meyers, Third Edition (Academic Press, New York, 2003), pp. 49–65.

[78] LIGO Scientific Collaboration (LIGO Scientific Collaboration), *Vibration isolation in LIGO*.

[79] C. Moore, R. Cole, and C. Berry, *Gravitational Wave Sensitivity plotter: Gravitational Wave Detectors and Sources*.

[80] B. P. Abbott et al., «Effects of data quality vetoes on a search for compact binary coalescences in Advanced LIGO's first observing run», Classical and Quantum Gravity **35**, 065010, 065010 (2018).

[81] D. Davis et al., «Subtracting glitches from gravitational-wave detector data during the third LIGO-virgo observing run», Classical and Quantum Gravity **39**, 245013 (2022).

[82] R. Macas et al., «Impact of noise transients on low latency gravitational-wave event localization», Phys. Rev. D **105**, 103021 (2022).

[83] J. Powell, «Parameter estimation and model selection of gravitational wave signals contaminated by transient detector noise glitches», Classical and Quantum Gravity **35**, 155017 (2018).

[84] E. Payne et al., «Curious case of GW200129: Interplay between spin-precession inference and data-quality issues», Phys. Rev. D **106**, 104017 (2022).

[85] B. Allen, «χ^2 Time-frequency discriminator for gravitational wave detection», Phys. Rev. D **71**, 062001 (2005).

[86] C. M. Biwer et al., «PyCBC Inference: A Python-based Parameter Estimation Toolkit for Compact Binary Coalescence Signals», Publications of the Astronomical Society of the Pacific **131**, 024503 (2019).

[87] M. Cabero et al., «Blip glitches in Advanced LIGO data», Classical and Quantum Gravity **36**, 155010 (2019).

[88] S. Soni et al., «Discovering features in gravitational-wave data through detector characterization, citizen science, and machine learning», Classical and Quantum Gravity **38**, 195016 (2021).

[89] D. V. Martynov et al., «Sensitivity of the advanced ligo detectors at the beginning of gravitational wave astronomy», Phys. Rev. D **93**, 112004 (2016).

[90] J. Glanzer et al., *Noise in the LIGO Livingston Gravitational Wave Observatory due to Trains*, 2023.

[91] N. J. Cornish and T. B. Littenberg, «Bayeswave: Bayesian inference for gravitational wave bursts and instrument glitches», Classical and Quantum Gravity **32**, 135012 (2015).

[92] D. Davis et al., «Subtracting glitches from gravitational-wave detector data during the third LIGO-Virgo observing run», Classical and Quantum Gravity **39**, 245013 (2022).

[93] D. Davis et al., «Improving the sensitivity of Advanced LIGO using noise subtraction», Classical and Quantum Gravity **36**, 055011 (2019).

[94] D. Macleod et al., *GWpy: Python package for studying data from gravitational-wave detectors*, Astrophysics Source Code Library, record ascl:1912.016, Dec. 2019.

[95] S. Chatterji, L. Blackburn, G. Martin, and E. Katsavounidis, «Multiresolution techniques for the detection of gravitational-wave bursts», Classical and Quantum Gravity **21**, S1809–S1818 (2004).

[96] J. S. Areeda et al., *LigoDV-web: Providing easy, secure and universal access to a large distributed scientific data store for the LIGO Scientific Collaboration*, arXiv:1611.01089 [astro-ph, physics:gr-qc], Nov. 2016.

[97] The LIGO Scientific Collaboration and The Virgo Collaboration, «LIGO/Virgo Alert System (LVAlert)»,

[98] Pace A, Prestegard T, Moe B and Stephens B, «GraceDB Gravitational-Wave Candidate Event Database», `https://gracedb.ligo.org/` (2020).

[99] S. Barthelmy et al., «Introducing new GCN Kafka broker and web site for transient alerts, https://gcn.nasa.gov», GRB Coordinates Network **32419**, 1 (2022).

[100] A. Geron, *Hands-on machine learning with scikit-learn, keras, and tensorflow :* 2nd ed., https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/ (O'Reilly Media, Inc., Mumbai, 2020).

[101] A. C. Wilson et al., *The marginal value of adaptive gradient methods in machine learning*, 2018.

[102] D. Masters and C. Luschi, *Revisiting small batch training for deep neural networks*, 2018.

[103] IBM, *What are convolutional neural networks?*

[104] C. Szegedy et al., «Going deeper with convolutions», in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015), pp. 1–9.

[105] S. Bahaadini et al., «Machine learning for Gravity Spy: Glitch classification and dataset», Information Sciences **444**, 172–186 (2018).

[106] C. Biwer et al., «Validating gravitational-wave detections: the advanced LIGO hardware injection system», Physical Review D **95**, `10.1103/physrevd.95.062002` (2017).

[107] C. Luo et al., «How Does the Data set Affect CNN-based Image Classification Performance?», in 2018 5th International Conference on Systems and Informatics (ICSAI) (2018), pp. 361–366.

[108] LIGO Scientific Collaboration, *LIGO Algorithm Library - LALSuite*, free software (GPL), 2018.

[109] S. Husa et al., «Frequency-domain gravitational waves from nonprecessing black-hole binaries. I. New numerical waveforms and anatomy of the signal», Phys. Rev. D **93**, 044006 (2016).

[110] 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**, 044007 (2016).

[111] K. O'Shea and R. Nash, *An Introduction to Convolutional Neural Networks*, arXiv:1511.08458 [cs], Dec. 2015.

[112] LIGO Scientific Collaboration, Virgo Collaboration and KAGRA Collaboration, «GWTC-3 Data Release», `https://www.gw-openscience.org/GWTC-3/` (2021).

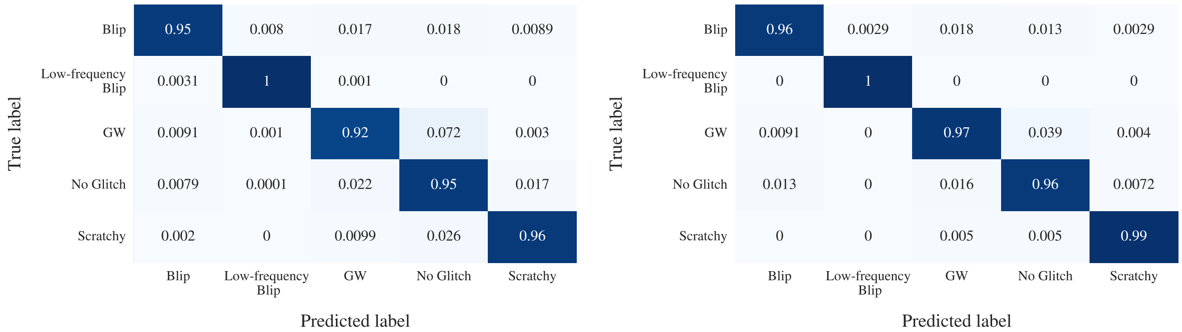
[113] D. George, H. Shen, and E. Huerta, «Glitch Classification and Clustering for LIGO with Deep Transfer Learning», in NiPS Summer School 2017 (Nov. 2017).

[114] L. Van der Maaten and G. Hinton, «Visualizing data using t-SNE.», Journal of machine learning research **9** (2008).

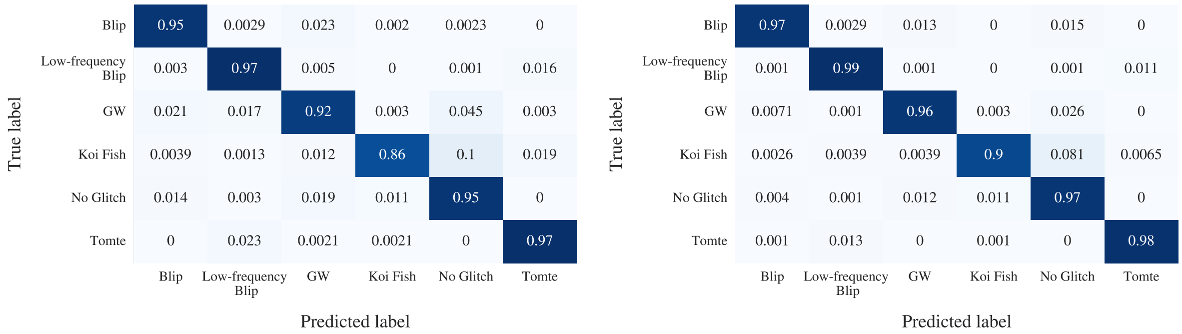
[115] D. Thain, T. Tannenbaum, and M. Livny, «Distributed computing in practice: the condor experience.», Concurrency - Practice and Experience **17**, 323–356 (2005).

A Confusion matrices for the multi-class pre-O4 version of GSpyNetTree

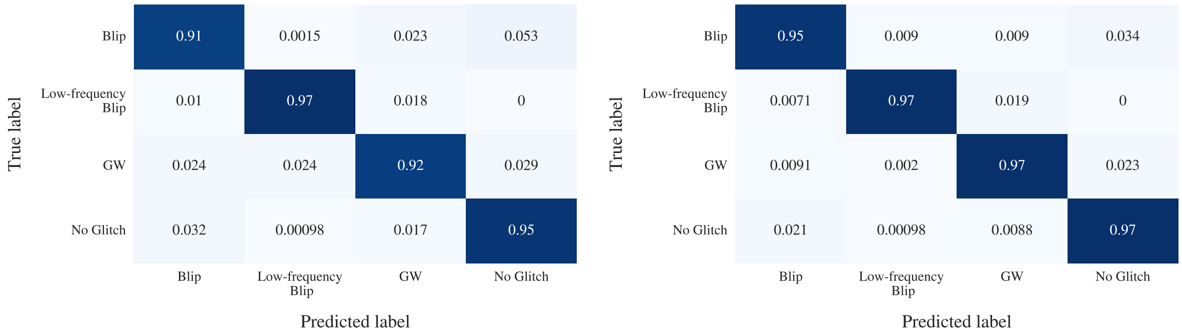
Figure A.1 shows the confusion matrices for the multi-class pre-O4 version of GSpyNetTree. Each row represents a different classifier. The left column shows the results with the Gravity Spy architecture [88], whereas the second column shows those obtained with Inception V3 [15].



(a) LM Classifier using Gravity Spy architecture. (b) LM Classifier using InceptionV3 architecture.



(c) HM Classifier using Gravity Spy architecture. (d) HM Classifier using InceptionV3 architecture.



(e) EHM Classifier using Gravity Spy architecture. (f) EHM Classifier using InceptionV3 architecture.

Figure A.1: Confusion matrices for each multi-class classifier of the pre-O4 version of GSpyNetTree. The left column shows the results using the Gravity Spy architecture [3] while the second one shows those obtained with Inception V3 [15]. The first row corresponds to the LM classifier, the second row to the HM classifier, and the third row to the EHM classifier.

B Confusion matrices for the multi-label low-mass O4 version of GSpyNetTree

Figures B.1, B.2, and B.3 show the confusion matrices for the multi-label low-mass O4 version of GSpyNetTree using the Inception V3 architecture [15].

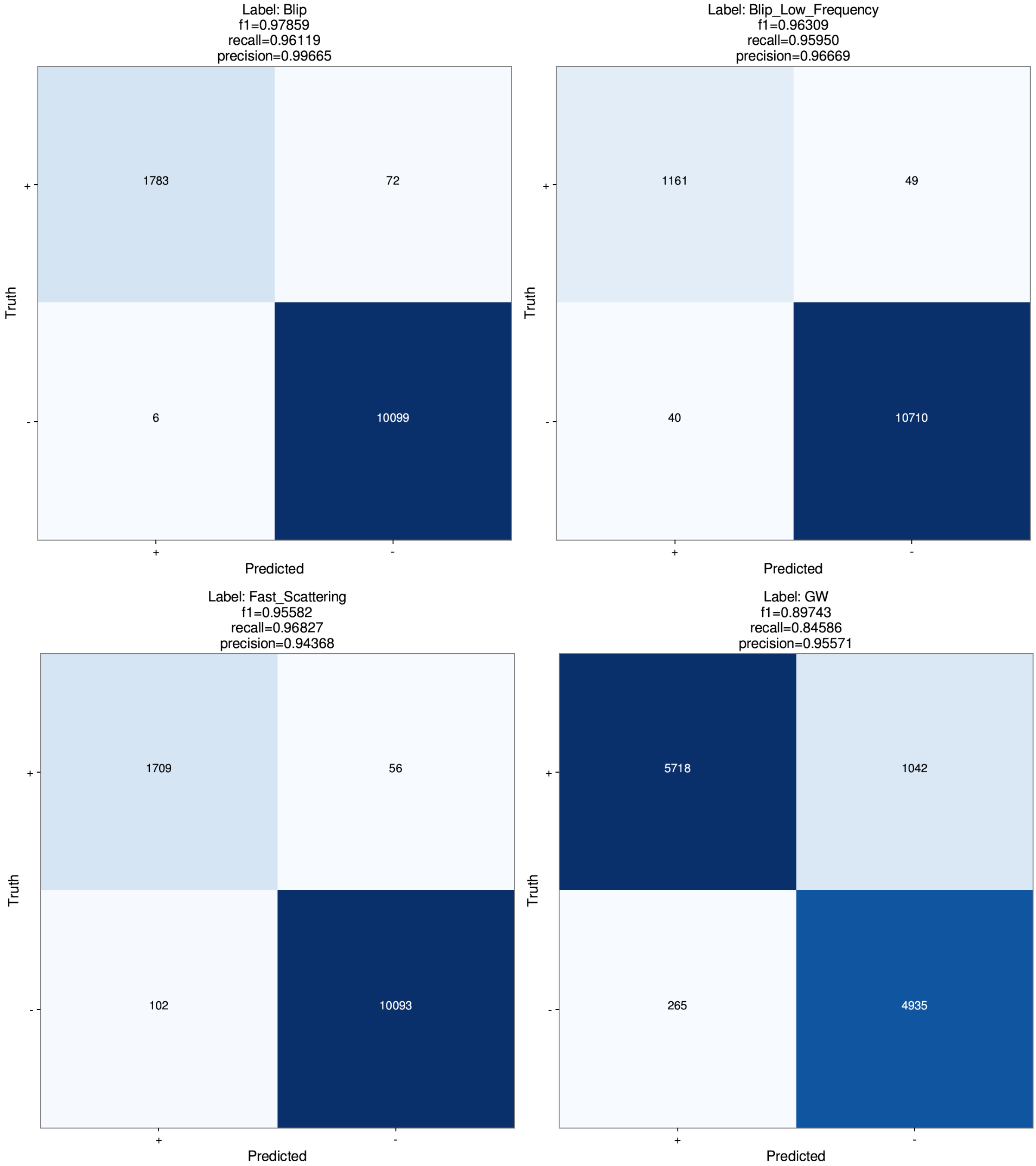


Figure B.1: Binary confusion matrices for each label of the low-mass multi-label classifier of the O4 version of GSpyNetTree.

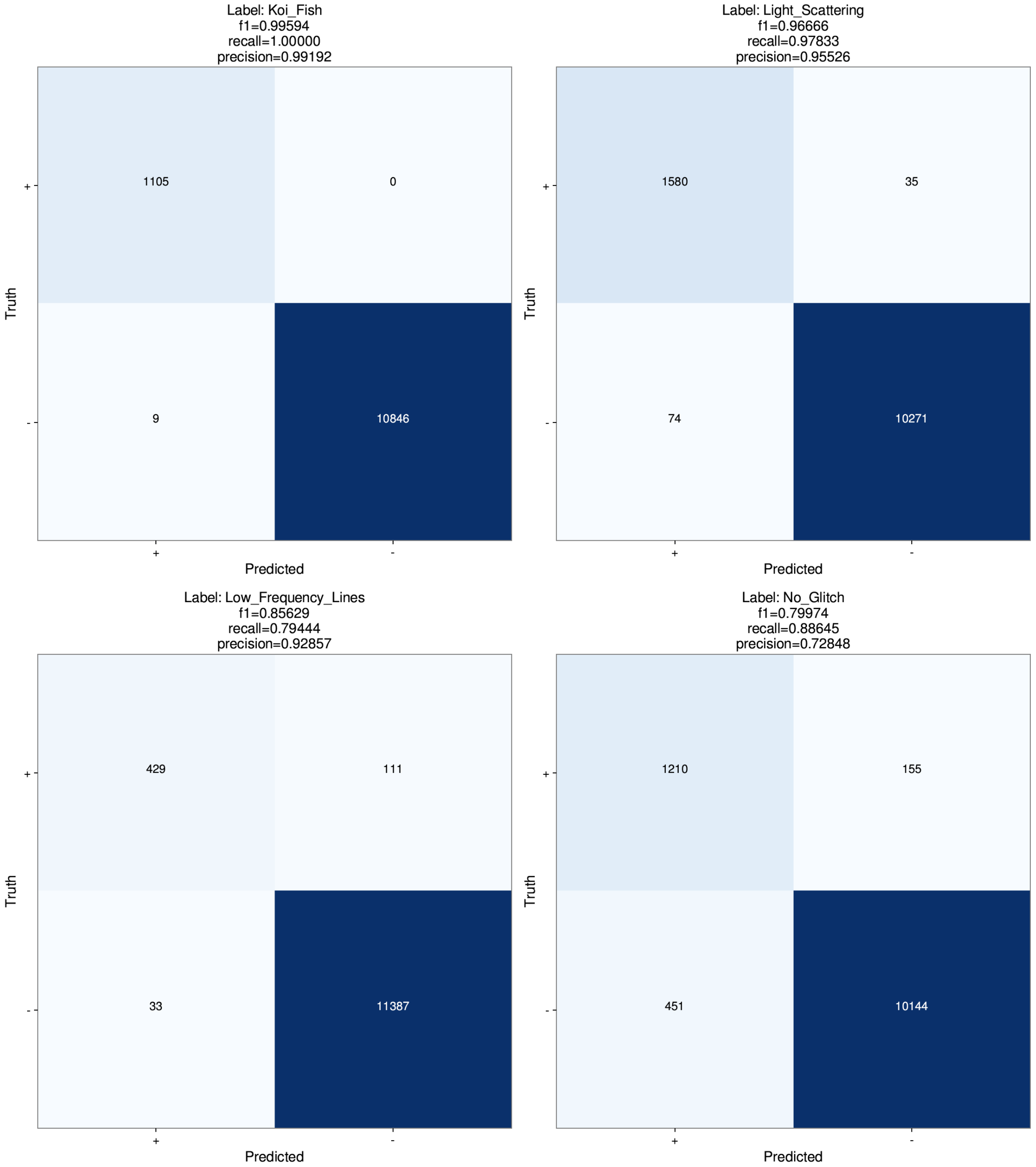


Figure B.2: Binary confusion matrices for each label of the low-mass multi-label classifier of the O4 version of GSpyNetTree.

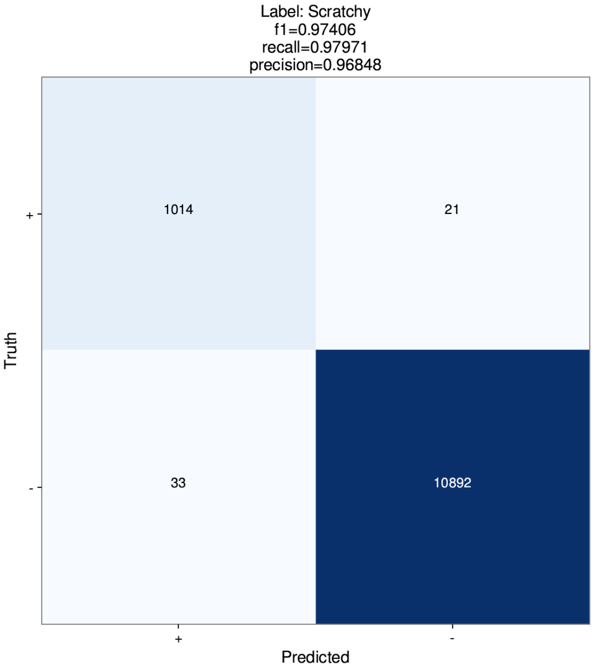


Figure B.3: Binary confusion matrices for each label of the low-mass multi-label classifier of the O4 version of GSpyNetTree.

C Confusion matrices for the multi-label high-mass O4 version of GSpyNetTree

Figures C.1, C.2, and C.3 show the confusion matrices for the multi-label high-mass O4 version of GSpyNetTree using the Inception V3 architecture [15].

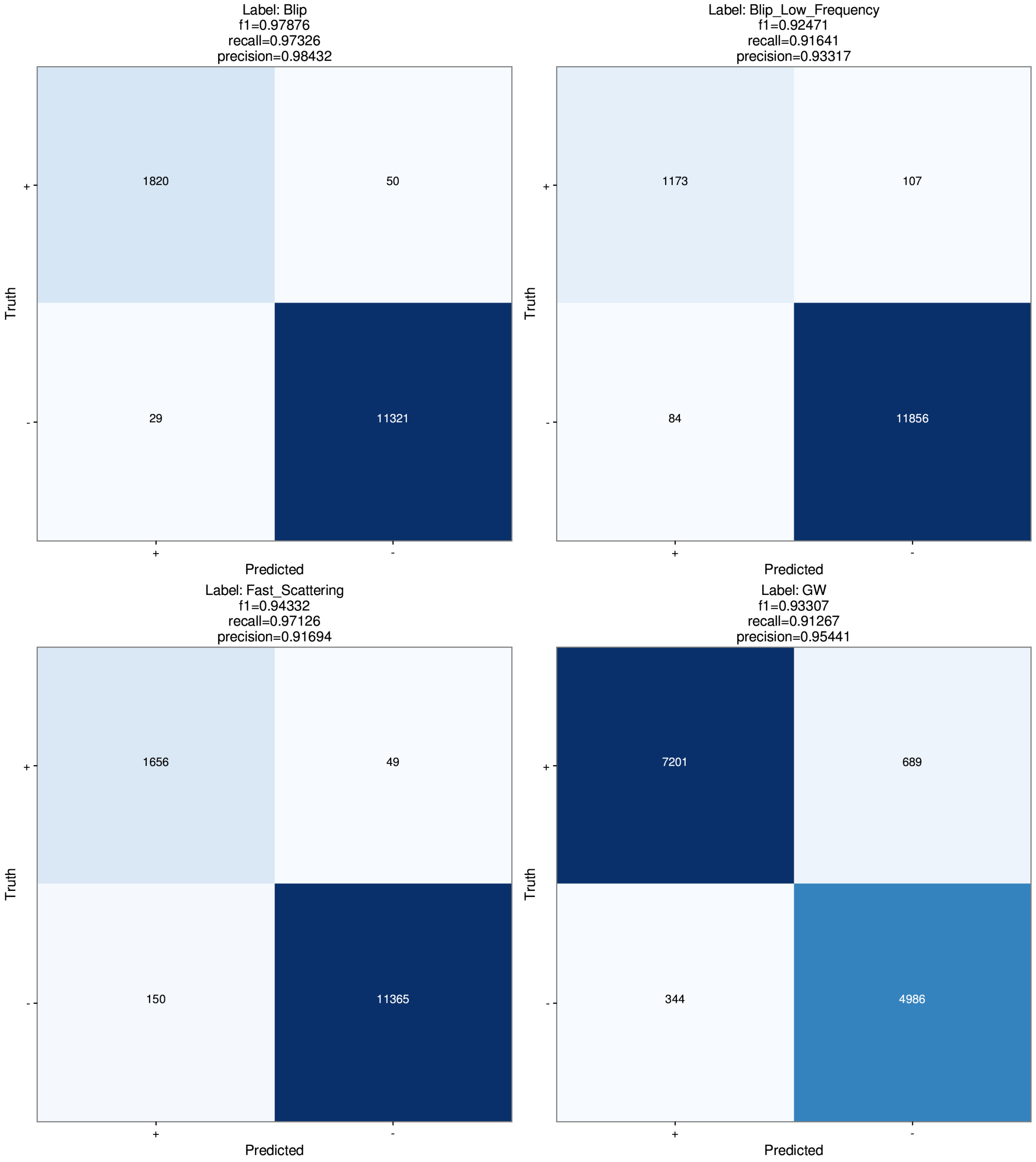


Figure C.1: Binary confusion matrices for each label of the high-mass multi-label classifier of the O4 version of GSpyNetTree.

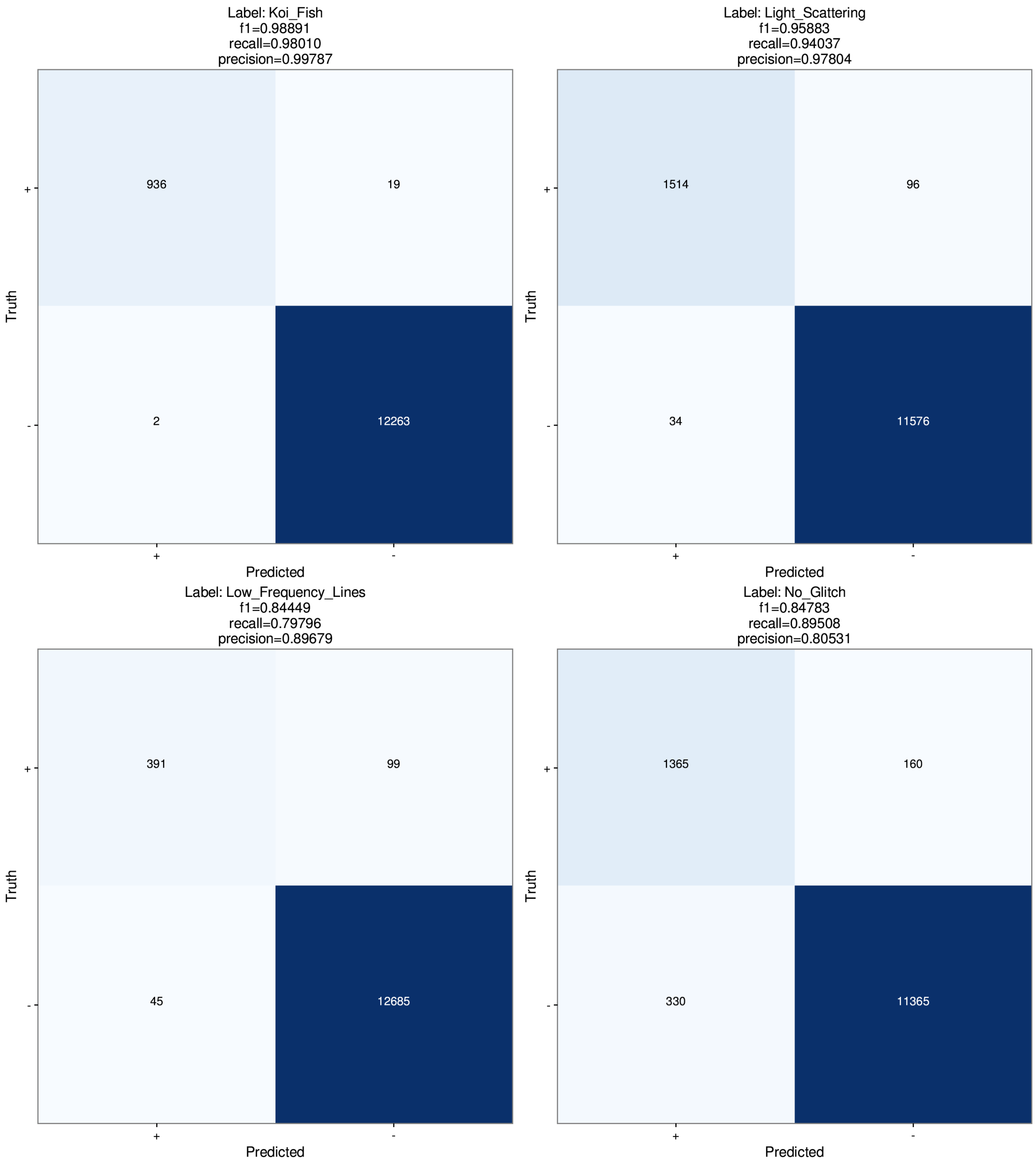


Figure C.2: Binary confusion matrices for each label of the high-mass multi-label classifier of the O4 version of GSpyNetTree.

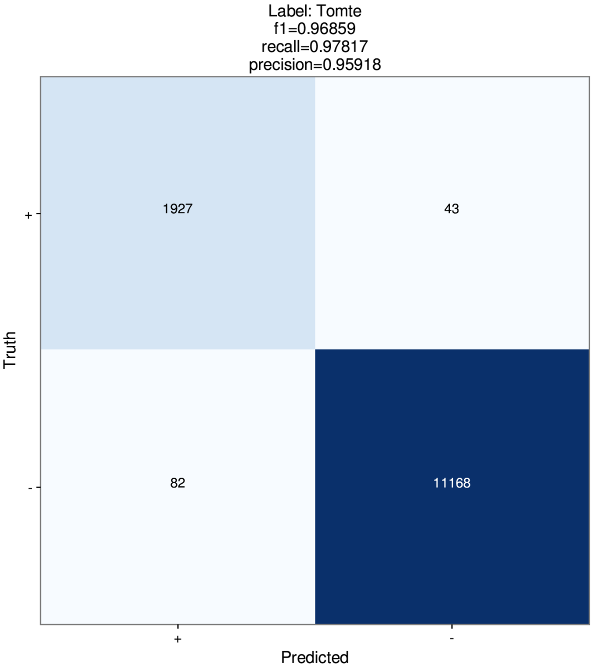


Figure C.3: Binary confusion matrices for each label of the high-mass multi-label classifier of the O4 version of GSpyNetTree.

D Confusion matrices for the multi-label extremely high-mass O4 version of GSpyNetTree

Figures D.1 and D.2 show the confusion matrices for the multi-label extremely high-mass O4 version of GSpyNetTree using the Inception V3 architecture [15].

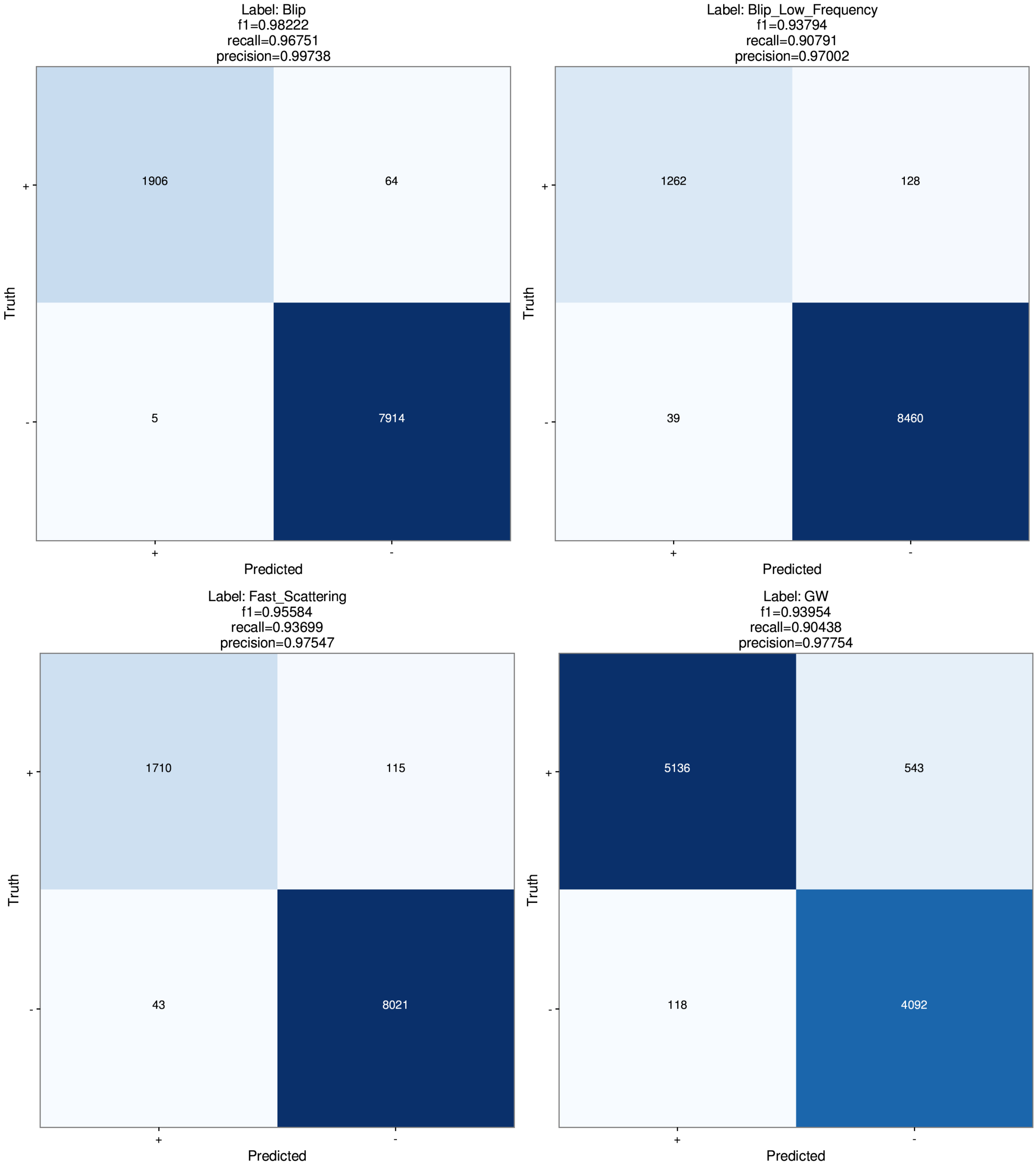


Figure D.1: Binary confusion matrices for each label of the extremely high-mass multi-label classifier of the O4 version of GSpyNetTree.

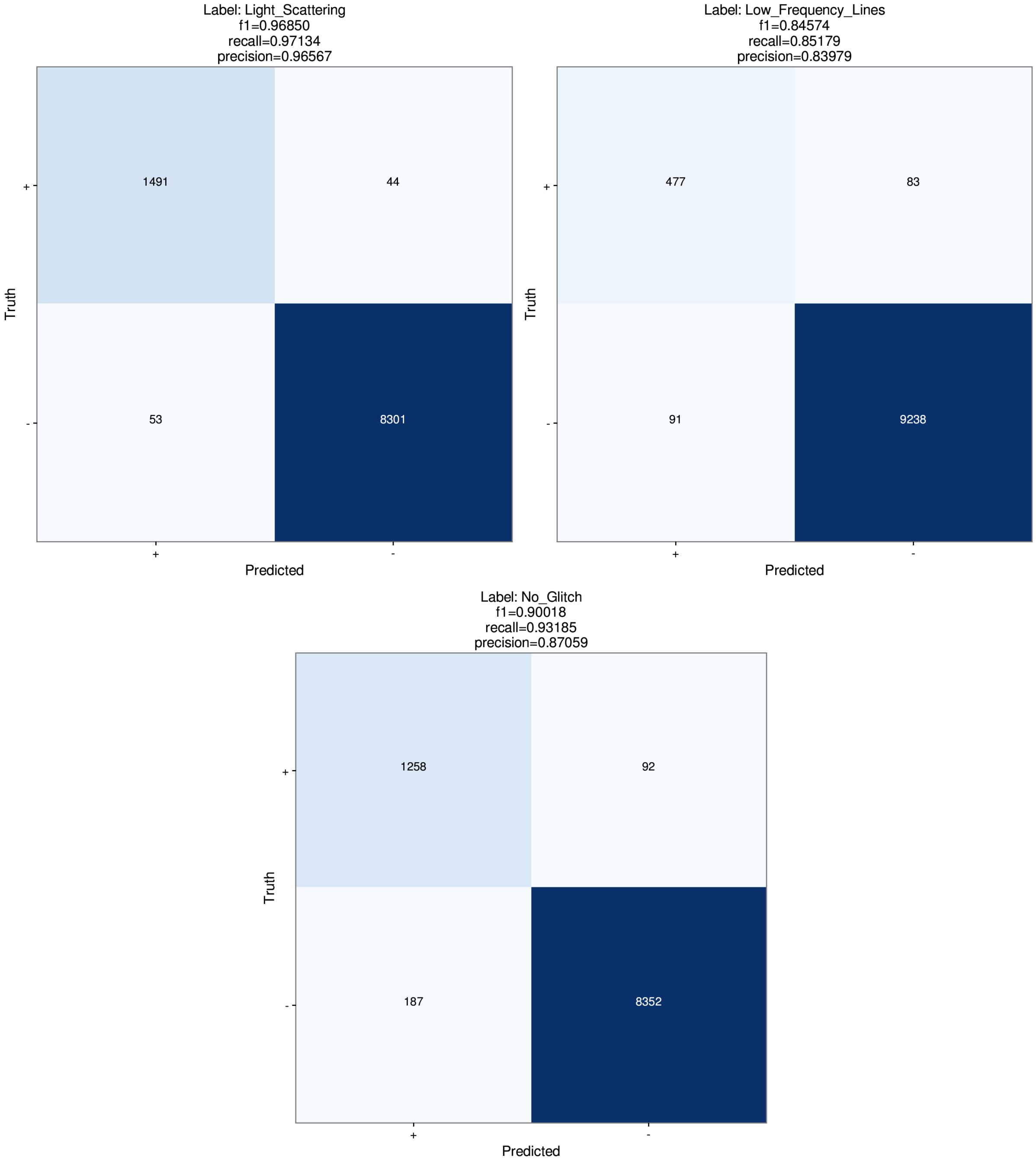


Figure D.2: Binary confusion matrices for each label of the extremely high-mass multi-label classifier of the O4 version of GSpyNetTree.